Скачать руководство пользователя в **pdf**

Для быстрых простых вопросов вы можете связаться с нами через **Telegram**.
Официальный канал техподдержки **@SMC5TechSupport** (https://t.me/SMC5TechSupport).
Онлайн Пн-Пт 8:00-16:00 UTC.

8SMC5-USB Руководство пользователя

1. О продукте
    1. Общие сведения
    2. Преимущества
    3. Сводная таблица характеристик
    4. Технические характеристики
2. Техника безопасности
3. Краткое руководство и начало работы
    1. Одноосная конфигурация
    2. Пример подключения простого мотора
    3. Решение возможных проблем
    4. Ручная настройка профиля
    5. Расчёт номинального тока
4. Техническое описание устройства
    1. Внешний вид и разъемы
        1. Плата контроллера
        2. Одноосная система
        3. Двухосная система
    2. Кинематика и режимы движения
        1. Движение с заданной скоростью
        2. Движение в заданную точку
        3. Смещение на заданное расстояние
        4. Движение с ускорением
        5. Компенсация люфта
        6. Реверсирование движения
        7. Рекомендации для точного движения
        8. ПИД алгоритм для управления DC двигателем
        9. Режимы остановки движения
        10. ПИД алгоритм для управления BLDC двигателем
    3. Основные возможности контроллера
        1. Поддерживаемые типы двигателей
        2. Ограничители на двигателях
        3. Концевые выключатели
        4. Автокалибровка "домашней" позиции
        5. Работа с энкодерами
        6. Датчик оборотов
        7. Обнаружение потери шагов
        8. Управление питанием мотора
        9. Критические параметры
        10. Хранение параметров во flash-памяти контроллера
        11. Пользовательские единицы координат
    4. Безопасная работа
    5. Дополнительные функции
        1. Индикация режима работы
        2. Работа с магнитным тормозом
        3. Управление с помощью джойстика
        4. Управление кнопками "вправо" и "влево"
        5. ТТЛ-синхронизация
        6. Создание многоосных систем
        7. Цифровой вход-выход общего назначения
        8. Аналоговый вход общего назначения
        9. Интерфейс управления внешним драйвером
        10. Последовательный порт
        11. Хранение позиции во FRAM-памяти контроллера
        12. Опознавание позиционеров Standa
    6. Второстепенные функции
        1. Установка нулевой позиции
        2. Установка пользовательской позиции
        3. Статус контроллера
        4. Автовосстановление USB соединения
5. Руководство по программе XILab
    1. О программе XILab
    2. Основные окна программы XILab
        1. Стартовое окно программы XILab
        2. Главное окно программы XILab в режиме управления одной осью
        3. Главное окно программы XILab в режиме управления несколькими осями
        4. Настройки программы
        5. Графики
        6. Скрипты

7. Лог XiLab
3. Настройки контроллера
    1. Настройка кинематики движения (Шаговый двигатель)
    2. Настройка диапазона движения и концевых выключателей
    3. Настройка предельных параметров контроллера
    4. Настройка параметров энергопотребления
    5. Настройка исходного положения
    6. Настройки синхронизации
    7. Настройка тормоза
    8. Контроль позиции
    9. Настройка внешних управляющих устройств
    10. Настройки UART
    11. Настройки цифрового входа-выхода общего назначения
    12. Настройка типа двигателя
    13. Настройка кинематики движения (DC мотор)
    14. Настройка контуров ПИД-регулирования
    15. О контроллере
    16. Настройка кинематики движения (BLDC мотор)
4. Настройки программы XILab
    1. Общие настройки программы XILab
    2. Настройки интерфейса абстрактного позиционера
    3. Настройки интерфейса аттенюатора
    4. Настройка режима циклического движения
    5. Настройка логирования
    6. Общие настройки отображения графиков
    7. Индивидуальные настройки отображения графиков
    8. Настройки отображения пользовательских единиц
    9. О программе
5. Характеристики позиционера
    1. Название позиционера
    2. Общие характеристики позиционера
    3. Характеристики двигателя
    4. Характеристики энкодера
    5. Характеристики датчика Холла
    6. Характеристики редуктора
    7. Характеристики вспомогательных устройств
6. Корректное завершение работы
7. Работа с сетевыми контроллерами
8. Установка XILab
    1. Установка под Windows
        1. Установка под Windows XP
        2. Установка под Windows 7
        3. Установка под Windows 8
    2. Установка под Linux
    3. Установка под MacOS
6. Программирование
    1. Руководство по программированию
        1. Работа с контроллером в среде LabView
        2. Работа с контроллером в среде Matlab
    2. Описание протокола обмена
    3. Совместимость с ПО для 8SMC1-USBhF
    4. Таймауты libximc
    5. Скрипты XILab
7. Файлы
    1. Конфигурационные файлы
        1. Линейные трансляторы
            1. 8MT160 - моторизированная линия задержки
            2. 8MT295 - моторизованный линейный транслятор большого диапазона
            3. 8MT195 - моторизованный линейный транслятор большого диапазона
            4. 8MT167 - моторизованный линейный транслятор
            5. 8MT173 - компактный моторизованный линейный транслятор
            6. 8MT173DC - компактный моторизованный линейный транслятор
            7. 8MT50 - моторизованный линейный транслятор
            8. 8MT30 - узкий моторизованный линейный транслятор
            9. 8MT175 - моторизованный линейный транслятор
            10. 8MT177 - моторизованный линейный транслятор
            11. 8MT184 - моторизованный линейный транслятор
            12. 8MT193 - моторизованный линейный транслятор
            13. 8MT200 - моторизованный линейный транслятор
            14. 8MTF - двухосный моторизованный линейный транслятор
            15. 8MTF2 - двухосный моторизованный линейный транслятор
            16. 8MTFV - моторизованный линейный транслятор
            17. 8MT60 - моторизованный линейный транслятор
        2. Ротаторы
            1. 8MR151 - моторизованный ротатор
            2. 8MR170 - моторизованный ротатор

3. 8MR174 - моторизованный ротатор
4. 8MR190 - моторизованный ротатор
5. 8MR191 - моторизованный ротатор
6. 8MRB250 - большой моторизованный ротатор
7. 8MRU - универсальный моторизованный ротатор
8. 8MPR16-1 - Моторизованный вращатель поляризатора
9. 8MRH240 - большой моторизованный ротатор

3. Вертикальные моторизованные трансляторы
    1. 8MVT100 - вертикальный моторизованный транслятор
    2. 8MVT120 - вертикальный моторизованный транслятор
    3. 8MVT188 - вертикальный моторизованный транслятор
    4. 8MVT40 - вертикальный моторизованный транслятор
    5. 8MVT70 - вертикальный моторизованный транслятор
4. Моторизованные винты
    1. 8MS00 - моторизованный винт
    2. 8CMA06 - моторизованный винт
    3. 8CMA20 - моторизованный винт
    4. 8CMA28 - моторизованный винт
    5. 8CMA16DC - моторизованный винт
5. Моторизованные гониометры
    1. 8MG00 - моторизованный гониометр
    2. 8MG99 - моторизованный гониометр
6. Моторизованные держатели зеркал
    1. 8MTOM2 - моторизованный оптический держатель
    2. 8MUP21 - моторизованный оптический держатель
    3. 8MBM24 - моторизованный держатель зеркала
    4. 8MMA60 - моторизованный держатель зеркала
    5. 8MBM57 - моторизованный держатель зеркала с большой апертурой
    6. 8MKVDOM - вертикальный моторизованный оптический держатель
7. Моторизованные аттенуаторы
    1. 10MCWA168 - моторизованный аттенуатор
    2. 10MWA168 - моторизованный аттенуатор
8. Моторизованные диафрагмы
    1. 8MID98 - моторизованная диафрагма

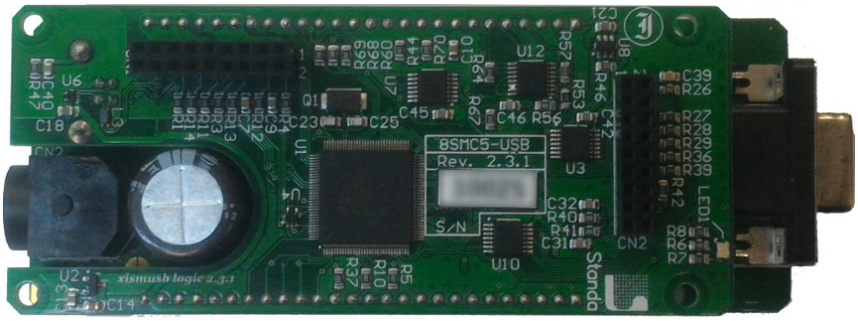
2. Программное обеспечение

8. Сопутствующие устройства
    1. Ethernet переходник
        1. Обзор
        2. Администрирование

1. О продукте

1. Общие сведения
2. Преимущества
3. Сводная таблица характеристик
4. Технические характеристики

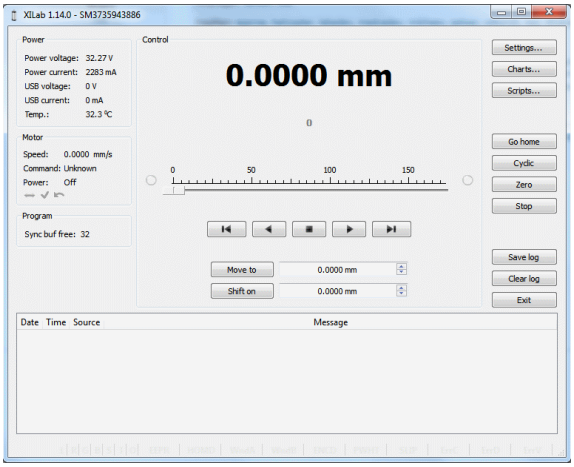
1.1. Общие сведения



Внешний вид контроллера.

Мы предлагаем недорогой ультра-компактный сервопривод с интерфейсом USB для шаговых двигателей с внешним питанием.

Забудьте о громоздких и дорогих сервоприводах! Теперь для работы вам понадобятся: шаговый двигатель, контроллер, USB кабель и практически любой внешний стабилизированный источник питания. И все! Не нужно никакого активного охлаждения. Плата контроллера по размеру не превосходит блокнот или сотовый телефон, так что вы можете положить его прямо на рабочий стол, не прибегая к монтажу. Контроллер может работать со всеми компактными шаговыми двигателями с током обмотки до 3 А, без обратной связи, а так же с шаговыми двигателями, оснащенными энкодером в цепи обратной связи, в том числе линейным энкодером на позиционере. Разъем для мотора на контроллере соответствует разъему, который использует компания Standa, и подходит для всех позиционеров Standa. USB соединение обеспечивает легкость подключения и простоту работы с компьютером. Несколько контроллеров могут быть подключены к одному компьютеру через несколько USB-портов или с помощью специальной объединительной платы, поставляемой в составе многоосных систем. Контроллер совместим практически со всеми операционными системами (Windows, Mac OS X, Linux и т. д.).



Внешний вид главного окна программы XILab.

С контроллером предоставляется все необходимое программное обеспечение и конфигурационные файлы, чтобы быстро начать работу по принципу "подключил и работает". Нами постоянно разрабатываются конфигурационные файлы для новых поддерживаемых двигателей. Так что вам нужно просто скачать конфигурационный файл для вашего позиционера из раздела Конфигурационные файлы и загрузить в XILab. Далее вам достаточно использовать только команды перемещения, все остальные настройки уже сделаны!

1.2. Преимущества

Основные преимущества

- **Компактный и мощный!** Габариты 47x37x80 мм со всеми разъемами, подходит для всех шаговых двигателей с номинальным током обмотке до 3 А.
- **Совместимость с 8SMC1-USBh** Все программы, написанные для 8SMC1-USBh, смогут работать с 8SMC5-USB после обновления MicroSMC драйвера.
- **Поддержка всех позиционеров Standa!** У вас позиционер Standa? Просто подключите, все остальное мы уже сделали за вас.
- **Помнит все!** Не нужно сохранять текущую позицию на компьютере, контроллер сделает это сам, используя собственную энергонезависимую память даже при неожиданной потере питания.
- **Умеет работать с периферией!** Поддержка квадратурного энкодера, магнитного тормоза, джойстика, концевиков, датчика нулевой позиции - все встроено, просто подключите и пользуйтесь.
- **Встроенная калибровка нуля!** Одной командой, по концевикам, датчику оборота, внешнему сигналу или их комбинациям.
- **Работает на любом компьютере!** Все программное обеспечение, поставляемое с контроллером, совместимо с Windows 8, Windows 7, Windows Vista, Windows XP SP3, Linux, Mac OS X, в том числе с 64-бит версиями.

Все преимущества

- **Действительно мощный!** Подходит для всех шаговых двигателей с номинальным током обмотке до 3 А.
- **Совместимость с 8SMC1-USBh** Все программы, написанные для 8SMC1-USBh, смогут работать с 8SMC5-USB после обновления MicroSMC драйвера.
- **Поддержка всех позиционеров Standa!** У вас позиционер Standa? Просто подключите, все остальное мы уже сделали за вас.
- **Знает своих!** Доступна встроенная функция загрузки всех конфигурационных данных непосредственно из подвижки, при поддержке такой памяти в позиционере.
- **Выбери свой интерфейс!** USB или последовательный порт, встроены и готовы к использованию.
- **По настоящему быстрый!** До 35 000 полных шагов/сек при любом делении шага!
- **Точный!** Режимы деления шага от полношагового до 1/256 шага на всех скоростях.
- **Помнит все!** Не нужно сохранять текущую позицию на компьютере, контроллер сделает это сам, используя собственную энергонезависимую память даже при неожиданной потере питания.
- **Умеет работать с периферией!** Поддержка квадратурного энкодера, магнитного тормоза, джойстика, концевиков, датчика нулевой позиции - все встроено, просто подключите и пользуйтесь. Отдельный стабилизированный выход 5В 100 мА для периферии.
- **Встроенная калибровка нуля!** Одной командой, по концевикам, датчику оборота, внешнему сигналу или их комбинациям.
- **Автономный!** Хотите работать автономно - пожалуйста. Поддерживается внешний джойстик, кнопочное управление или их комбинация.
- **Экономный!** Снижение тока через обмотку двигателя в режиме удержания. Программно, с шагом 1%.
- **Бесшумный!** Плавное движение на малых скоростях, отсутствие дополнительных шумов на больших.
- **Защищенный!** ESD защита по всем контактам внешних разъемов, защита от КЗ в цепях мотора.
- **Внимательный!** Контролирует температуры процессора, силовой части; токи и напряжения как силового питания, так и USB.
- **Современный!** Обновление микропрограммы в энергонезависимой памяти контроллера по USB.
- **Управляющий и управляемый!** Встроен вход и выход синхронизации, позволяющий начать движение в заданную позицию по внешнему сигналу и/или выдать сигнал при достижении заданной позиции. Встроен аналоговый вход общего назначения, цифровой вх/вых общего назначения.
- **Понятный!** Статусный светодиод отражает состояние контроллера и наличие электропитания. Для удобства оба сигнала и состояние концевиков также выводятся на внешние светодиоды.
- **Многоосный!** Многоосные системы создаются с помощью стандартных USB-хабов, внешних или установленных на специальную объединительную плату. До 32 осей в системе.
- **Работает на любом компьютере!** Все программное обеспечение, поставляемое с контроллером, совместимо с Windows 8, Windows 7, Windows Vista, Windows XP SP3, Linux, Mac OS X, в том числе с 64-бит версиями.
- **Примеры для всех языков!** Контроллер поставляется с кросс-платформенной библиотекой и примерами, которые позволяют быстро начать программирование на C++, C#, .NET, Delphi, Visual Basic, gcc, Xcode, Matlab, Java и LabVIEW.
- **Полнофункциональный интерфейс!** Контроллер поставляется с интерфейсом пользователя XILab, позволяющим легко управлять всеми функциями устройства без необходимости в разработке собственных программ.
- **Собственный скриптовый язык!** В XILab интегрирован скриптовый язык, позволяющий легко, без компиляции и освоения какого-либо языка программирования задавать последовательности действий, включая циклы и условные переходы.
- **Новое! Реализован новый алгоритм управления шаговым двигателем с ведущим энкодером!** На всех контроллерах 8SMC5-USB управление движением с использованием обратной связи по энкодеру делает перемещения мотора более быстрыми и более плавными. Алгоритм с ведущим энкодером сочетает в себе преимущества управления бесщеточным двигателем постоянного тока с дешевизной обычных шаговых двигателей. Никаких проскальзываний, заминок и срывов движения!

1.3. Сводная таблица характеристик

Тип питания контроллера	внешнее 12-48 В
Потребляемый контроллером ток	до 5 А от внешнего источника питания, зависит от напряжения источника питания
Ток в обмотке мотора	до 3 А
Защиты	от перегрузки по току, напряжению, короткого замыкания, горячего отключения двигателя
Режим работы	движение в направлении, движение в заданную точку, смещение на заданную дельту, поддержание заданной скорости, трапецевидный профиль скорости, режим компенсации люфта, автоматическая калибровка домашней позиции
Режимы деления шага	полношаговый, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256
Скорость движения	до 35 000 полных шагов/сек
Профиль скорости	трапецевидный
Счетчик позиции	40 бит
Интерфейс управления	USB, UART, возможность поддержки Bluetooth, Wi-Fi, Ethernet и др.
Обратная связь	квадратурный несимметричный и дифференциальный энкодер (опционально)
Концевые выключатели	2 шт.
Ширина полосы обратной связи	200 кГц для несимметричного и 5МГц для дифференциального
Датчик оборотов	поддерживается

Замечание: Рабочий диапазон напряжений питания 12-48 В. На нём контроллер работает согласно данной инструкции. Предельный диапазон напряжений питания 12-50 В. При превышении напряжения 50 В контроллер гарантированно выходит из строя. При напряжении менее 12В контроллер не способен работать.

1.4. Технические характеристики

Требования к электродвигателю

- Тип электродвигателя: биполярный шаговый электродвигатель, двигатель постоянного тока.
- Номинальный ток в обмотке: не менее 100 мА.
- Номинальное напряжение на обмотке: не менее 2 В.

Электрические характеристики контроллера

- Режимы электропитания: от внешнего источника питания.
- Ток в каждой обмотке мотора: до 3 А.
- Максимальная частота следования импульсов с энкодера: 200 кГц для несимметричного и 5МГц для дифференциального
- Стабилизированный выход 5 В (для питания энкодера и прочей внешней электроники): выходной ток не более 100 мА, стабильность выходного напряжения не хуже 5%.
- ESD защита по всем контактам внешних разъемов (D-Sub 15 pin, USB, питание).
- Защита от замыкания обмоток мотора на "землю".
- Защита от замыкания обмоток мотора между собой.
- Защита от подключения/отключения мотора к контроллеру в процессе работы.
- Защита от подачи питания неверной полярности (не более 1 сек).
- Защита от подачи напряжения питания больше допустимого (не более 1 сек).
- Ограничение тока потребляемого от USB.
- Ограничение тока потребляемого от внешнего источника питания.
- Ограничение скорости вращения мотора.
- Установка полного рабочего тока через обмотку ШД. Программно, с точностью 10 мА.
- Снижение тока через обмотку ШД в режиме удержания. Программно, с шагом 1%.

Возможности управления движением

- Режимы деления шага: полношаговый, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256.
- Бесшумное движение на малых скоростях.
- Минимальная скорость: 1/256 полного шага/сек.
- Максимальная скорость: до 35 000 полных шагов/сек при всех режимах деления шага.
- Минимальное смещение: 1/256 шага.
- Максимальное смещение: 2 147 483 647 полных шагов при всех режимах деления шага.
- Режим плавного старта/останова.
- Счетчик позиции: 40 бит (32 бит - полный шаг, 8 бит - микрошаг).
- Режимы движения: движение в направлении, движение в заданную точку, смещение на заданную дельту, поддержание заданной скорости, трапецевидный профиль скорости, режим компенсации люфта.

Дополнительные функции микропрограммы

- Режим автокалибровки "HOME" на уровне микропрограммы.
- Сохранение/загрузка настроек в энергонезависимую память контроллера.
- Обновление микропрограммы в энергонезависимой памяти контроллера по USB.
- Автоматическое сохранение позиции по счетчику шагов и по энкодеру с защитой от неожиданного отключения питания.

Дополнительные функции, реализованные через разъем подключения мотора

- Отработка сигналов с одного или двух концевиков. Конфигурируется программно.
- "Опознавание" позиционеров Standa и автоматическая загрузка всех конфигурационных данных для них (при поддержке соответствующей возможности в позиционере). Автонастройка параметров движения на основе считанных данных.
- Определение "потери шагов" и восстановление правильной позиции с помощью датчика оборотов или квадратурного энкодера (при поддержке соответствующей возможности в позиционере).
- Определение положения с помощью квадратурного энкодера. Режим x4.
- Движение шагового двигателя в режиме "ведущего энкодера", т.е. с опорой на показания квадратурного энкодера, обеспечивающее движение без потери шагов на максимальной доступной скорости реализовано в прошивке начиная с 4.1.

Дополнительные функции, реализованные через разъем backplane

- Дубликат входа USB. Используется совместно с объединительной платой.
- Последовательный порт RS-232. Доступны линии TX, RX. Параметры: скорость – 9600 - 921600 бод, ТТЛ 3.3 В. На базе последовательного порта по запросу доступны конфигурации с интерфейсом Ethernet, Bluetooth, WiFi, ZigBee и другие.
- Вход синхронизации – при подаче импульса на этот вход (срабатывание, полярность и длительность настраиваются пользователем) контроллер начинает движение в заранее заданную позицию или на заданное смещение. Параметры: ТТЛ 3.3 В.
- Выход синхронизации – выдача импульса с этого выхода контроллера (срабатывание, полярность и длительность настраиваются пользователем) производится либо по началу движения, либо завершению движения, либо через заданное перемещение (задается пользователем). Параметры: ТТЛ 3.3 В.
- Кнопки "вправо", "влево". Нажатие на кнопку приводит к движению в заданную сторону с постепенным настраиваемым ростом скорости и в соответствии с настройками ускорения и другими параметрами. Параметры: ТТЛ 3.3 В.
- Вход "джойстик". Позволяет работать с джойстиками с диапазоном выдаваемых напряжений не шире 0 - 3 В.
- Выход для управления магнитным тормозом. Позволяет управлять магнитным тормозом, установленным на ось шагового двигателя. Параметры: ТТЛ 3.3 В, 5 мА.
- Аналоговый вход общего назначения. Позволяет работать с сигналами 0-3 В. Частота считывания 1 кГц. Конфигурируется

программно.
- Цифровой вх/вых общего назначения. Частота обновления 1 кГц. Конфигурируется программно. Параметры: ТТЛ 3.3 В, 5 мА.
- Индикация концевых выключателей. TTL 3.3 В, 2 мА. Предназначены для непосредственного подключения светодиодов.
- Дублирующие статусный светодиод цифровые выходы Power и Status. TTL 3.3 В, 2 мА. Предназначены для непосредственного подключения светодиодов.
- Интерфейс управления внешним драйвером. Позволяет управлять любым внешним драйвером с помощью трех сигналов: enable, direction, clock.
- Создание многоосных систем. Многоосные системы создаются с помощью стандартных USB-хабов, внешних или установленных на специальную объединительную плату. На ПК многоосная система выглядит как множество виртуальных последовательных портов, по числу подключенных осей.

Программирование

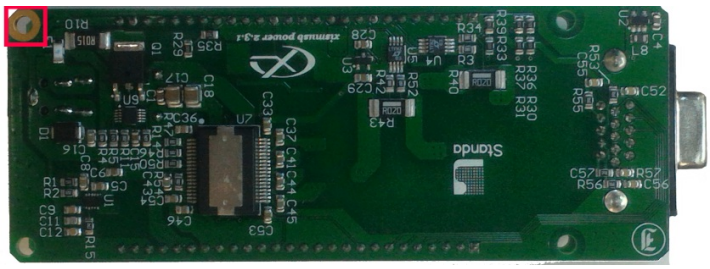
- Все поставляемое программное обеспечение, поставляемое с контроллером, совместимо с Windows 8, Windows 7, Windows Vista, Windows XP SP3, Linux, Mac OS X, в том числе с 64-бит версиями.
- Контроллер поставляется с кросс-платформенной библиотекой и примерами, которые позволяют быстро начать программирование на C++, C#, .NET, Delphi, Visual Basic, gcc, Xcode, Matlab, Java и LabVIEW.
- Контроллер поставляется с интерфейсом пользователя XILab, позволяющим легко управлять всеми функциями устройства без необходимости в разработке собственных программ.
- В XILab интегрирован скриптовый язык, диалект EcmaScript, позволяющий легко, без компиляции и освоения какого-либо языка программирования задавать последовательности действий, включая циклы и условные переходы.

2. Техника безопасности

Требования к блоку питания и заземлению. Подключение контроллера

Ниже перечислены основные требования к блоку питания для работы как с платой контроллера, так и с контроллерами в корпусе (одноосная и двухосная системы).

ВАЖНО. Блок питания должен быть заземлен через розетку 220В (трехпроводная схема подключения). Убедитесь, что используемый вами блок питания имеет заземленный минусовой выход. Если вы используете блок питания с отвязанным от земли минусовым проводом, заземляйте контроллер через **клемму заземления**, которая находится на плате контроллера!



Внешний вид платы контроллера сверху. Красным квадратом отмечена клемма заземления

Типовые схемы подключения контроллера (как коробочных версий, так и платы без корпуса):

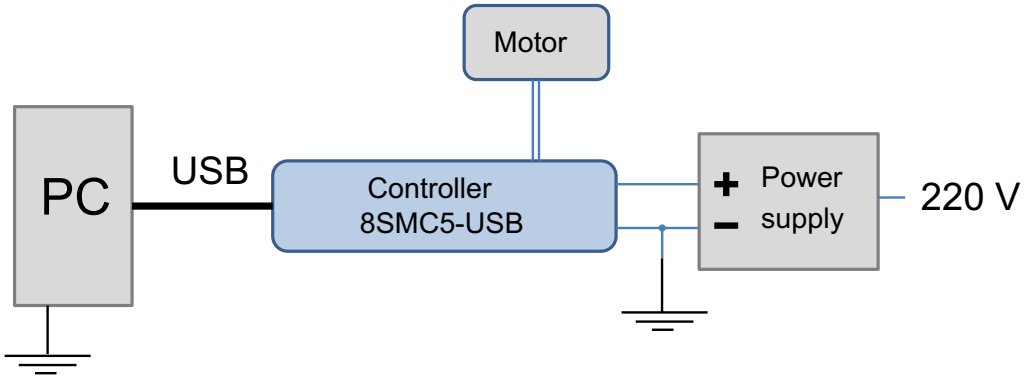


Схема подключения контроллера с заземлением через минусовый провод блока питания

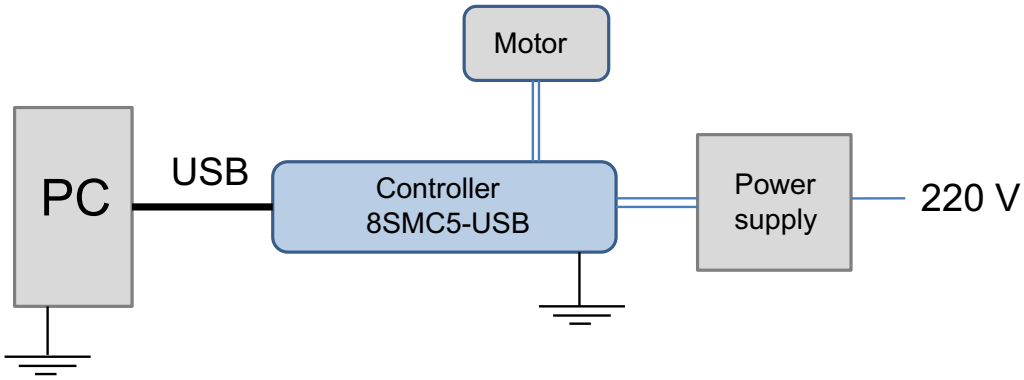


Схема подключения контроллера с заземлением через клемму заземления

Предупреждение. Блок питания должен обеспечивать необходимый ток для вращения двигателя. Абсолютным минимумом является ток, рассчитываемый по формуле $I_{power.min} = \frac{2 * I_{motor} * U_{motor}}{U_{power}}$, где $I_{power.min}$ это минимальный рабочий ток блока питания, I_{motor} это рабочий ток через обмотку двигателя, U_{power} это стабилизируемое напряжение блока питания, а U_{motor} это номинальное рабочее напряжение мотора. Рекомендуется использовать блок питания с рабочим током $I_{power} \geq 2 * I_{power.min}$. Напряжение U_{power} должно быть выше U_{motor}. Чем выше напряжение питания, тем большей скорости вращения можно достичь.

Можно использовать параметр мощности блока питания вместо рабочего тока. Абсолютным минимумом мощности блока питания будет $W_{power.min} = I_{power.min} * U_{power} = 2 * I_{motor} * U_{motor}$. Например, для мотора с рабочим током в обмотке 1А и рабочим напряжением 5В (номинальная мощность 5Вт), рабочее напряжение блока питания можно взять 20В с выдаваемой мощностью не менее 10Вт (максимальный рабочий ток блока питания не менее 0.5 А).

Плата контроллера

ВАЖНО. При работе с платой контроллера **запрещено трогать плату руками**, т.к. статический разряд может повредить компоненты на ней. Рекомендуется пользоваться антистатическими браслетами.

ВАЖНО. При работе с платой контроллера **категорически запрещено подключать плюс силового источника питания к контроллеру при не подключенном общем проводе источника питания или подключать/отключать кабель силового питания на горячую** в случае, когда источник питания находится под напряжением, контроллер подключен к компьютеру, источник питания и компьютер заземлен. **Это может привести к выходу из строя компьютера!** Данное требование является общим для любых электронных устройств с собственным электропитанием, подключенным к компьютеру по интерфейсу USB.

Предупреждение. Рекомендуется сначала подключить плату контроллера к силовому электропитанию с заземлением, либо отдельно заземлить плату с помощью специально маркированной клеммы (см. пункт выше), прежде чем подключать контроллер к мотору или к компьютеру по интерфейсу USB.

Одноосная и двухосная система в корпусе

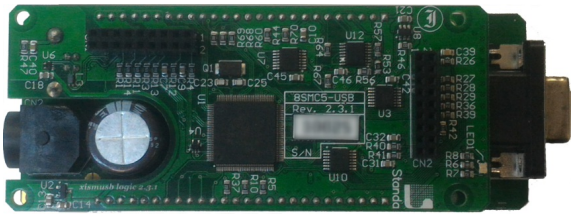
ВАЖНО. При работе с контроллером в корпусе (одноосная или двухосная системы) **запрещено превышать максимальное напряжение питания, равное 48В**, превышение этого значения более чем на 2В может немедленно привести к выходу системы из строя.

3. Краткое руководство и начало работы

Данное руководство описывает работу с контроллером многоосных и одноосных систем, настройку основных параметров и начало работы с программным обеспечением XILab для Windows 7.

1. Одноосная конфигурация - краткое описание начала работы с контроллером 8SMC5-USB для одной оси. Рассмотрена быстрая настройка XiLab, перечислено всё необходимое оборудование.
2. Пример подключения простого мотора - подключение к 8SMC5-USB шагового мотора на примере Nanotec ST5918L3008-B с энкодером CUI INC AMT112S-V. Описано, как изготовить собственный кабель, руководствуясь спецификацией на двигатель, даны пояснения характеристикам из спецификации.
3. Ручная настройка профиля - настройка рабочего профиля для XiLab. Обзор основных функциональных возможностей.
4. Расчёт номинального тока
5. Решение возможных проблем - описание и решение наиболее часто возникающих проблем при работе с 8SMC5-USB.

3.1. Одноосная конфигурация



Внешний вид платы контроллера.

3.1. Одноосная конфигурация
Введение
Системные требования
Установка ПО и первый пуск
Начало работы в ПО XILab
Проверка работоспособности
Управление из пользовательских приложений

Введение

Данное руководство описывает установку контроллера и начало работы с программным обеспечением XILab для Windows 7. Подробно с характеристиками контроллера вы можете ознакомиться в разделе Технические характеристики. Для разработки собственных приложений для контроллера рекомендуем ознакомиться с разделом Руководство по программированию и скачать пакет программ для разработки приложений в разделе Программное обеспечение.

Системные требования

> *Замечание.* В настоящем разделе требования описаны кратко. Подробно системные требования описаны в разделе Технические характеристики.

Для успешной настройки контроллера необходимо иметь:

- **Компьютер с наличием USB разъёма**



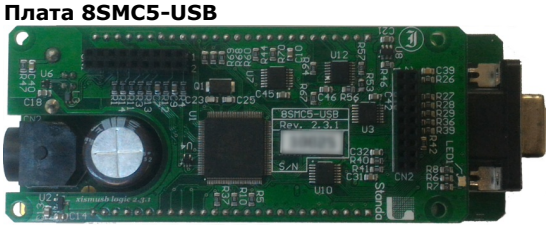
  Убедитесь, что у вас установлены все актуальные обновления системы Windows. Это позволяет поддерживать совместимость вашего компьютера с новым программным обеспечением. Посмотреть и скачать обновления, если это необходимо, можно по ссылке www.microsoft.com/updates.

- **Программное обеспечение**
  Всё необходимое ПО для работы с контроллером можно скачать по ссылке программное обеспечение.

- **USB A - mini-B кабель**

  USB-A – мини-USB-B кабель.
  Подробней о USB кабеле смотрите в разделе разъём управления платой контроллера или разъём управления 1- или 2-осной системами.

- **Плата 8SMC5-USB**



Плата контроллера двигателя.
В зависимости от комплектации и версии, внешний вид контроллера может отличаться от приведенного на рисунке. Более подробную информацию о версиях контроллеров можно найти в разделе внешний вид и разъемы.

- **Позиционер или мотор**



Позиционер на базе шагового двигателя.
На рисунке представлен используемый в работе позиционер на базе шагового двигателя, более подробно о требованиях к электромотору написано в разделе Технические характеристики. Если подразумевается использование собственных кабелей для подключения контроллера к позиционеру, пожалуйста, руководствуйтесь схемой соединения контроллера и позиционера, а также схемой расположения выводов контроллера. Для позиционеров с конечным диапазоном перемещения следует использовать два концевых выключателя: SW1 и SW2. Данные контакты используются для определения границ движения подвижки.

- **Блок питания**



Стабилизированный источник питания.

Замечания.

- Используйте стабилизированный источник питания 12 - 36 В. Слишком высокое напряжение может повредить контроллер. Подробней смотрите Техника безопасности. Рабочий ток блока питания должен быть достаточен для стабильного вращения двигателя.
- Обратите внимание, на инструкцию, прилагаемую к вашему контроллеру. В зависимости от модели контроллера, возможно более строгое ограничение на питающее напряжение. Внимательно проверьте соединение контроллера с внешним источником питания.
- Если контроллер поставляется с металлическим корпусом, то корпус требует заземления. Заземление контроллера, в случае, если он поставляется без корпуса, происходит через "землю" блока питания. Более подробно о заземлении смотри в разделе Техника безопасности.
- Убедитесь, что работающая без корпуса плата контроллера лежит на непроводящей электрический ток поверхности, а также в отсутствии посторонних частиц на плате и около нее.

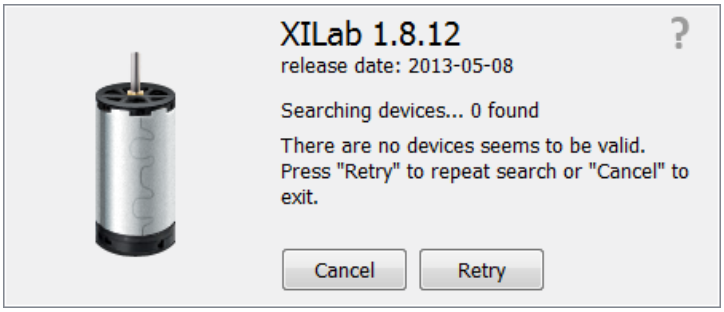
Установка ПО и первый пуск

Убедитесь, что все контроллеры отключены от компьютера.
Руководство по установке программного обеспечения находится здесь. Выберите файл "xilab-<номер версии>.exe". Инсталлятор автоматически определяет, запущен ли он на 32-битной или 64-битной системе и устанавливает соответствующую версию. Появится окно установки (версии программного обеспечения могут немного отличаться).



Первое окно установки XILab.

Нажмите кнопку “Next>” и следуйте инструкциям на экране. Все необходимое программное обеспечение, включая драйверы, пакеты и программы будут установлены автоматически. После установки по умолчанию запустится программа XiLab и появится следующее окно.



Диалоговое окно программы XILab "No devices found" (устройства не найдены).

Не нажимайте никаких кнопок.
Подключите позиционер к контроллеру. Подключите стабилизированный источник питания к контроллеру. Заземлите контроллер или блок питания. Подключите контроллер к компьютеру используя USB-A – мини-USB-B кабель.

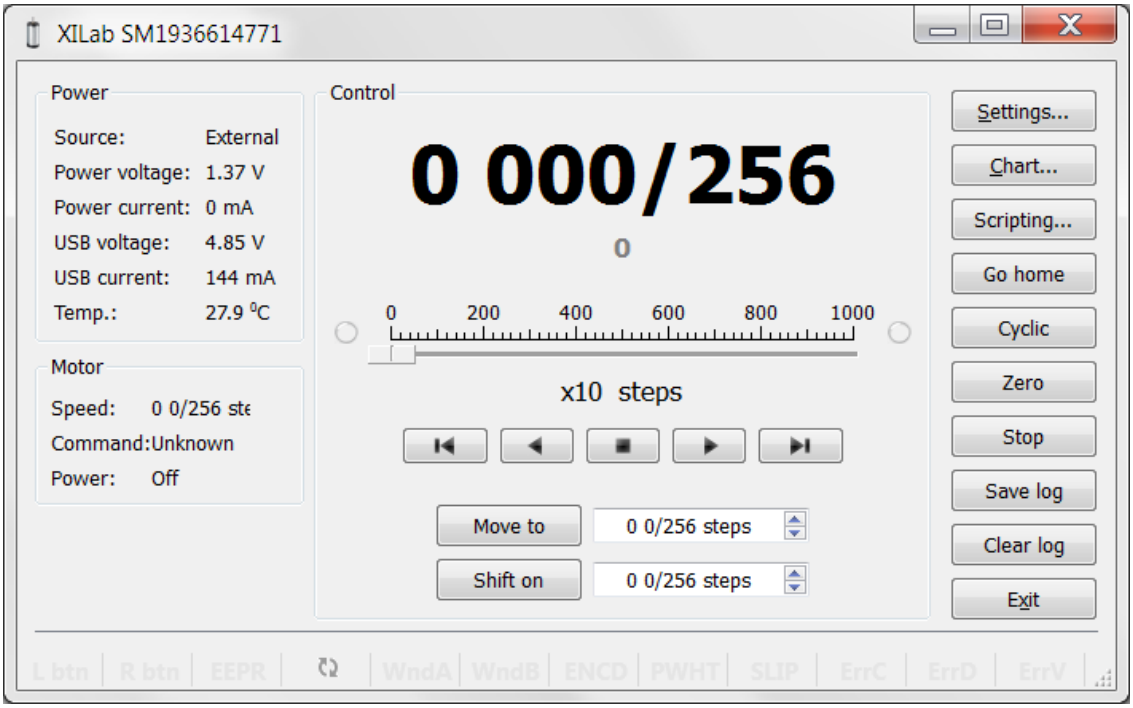
LED индикатор на плате контроллера начнет мигать. Мастер New Hardware Wizard начнет работать после первого подключения контроллера к компьютеру. Подождите пока Windows обнаружит новое устройство и установит необходимые драйверы для него.

Если драйвер автоматически не установился, то в появившемся окне выберите ”No, not this time”, затем нажмите “Next>”. В следующем окне выберите “Install from a list or specific location (Advanced)” и нажмите “Next>”. Выберите *.inf файл на диске с ПО, поставляемым с контроллером, или в директории C:\Program Files\XILab\driver\ и подождите пока установка будет завершена.

Вернитесь к диалоговому окну программы XILab "No devices found" и нажмите "Retry".
Если данное окно было закрыто, перейдите в меню “Start”, выберите Programs->XILab X.X.X->XILab и запустите. Тогда указанное окно появится.

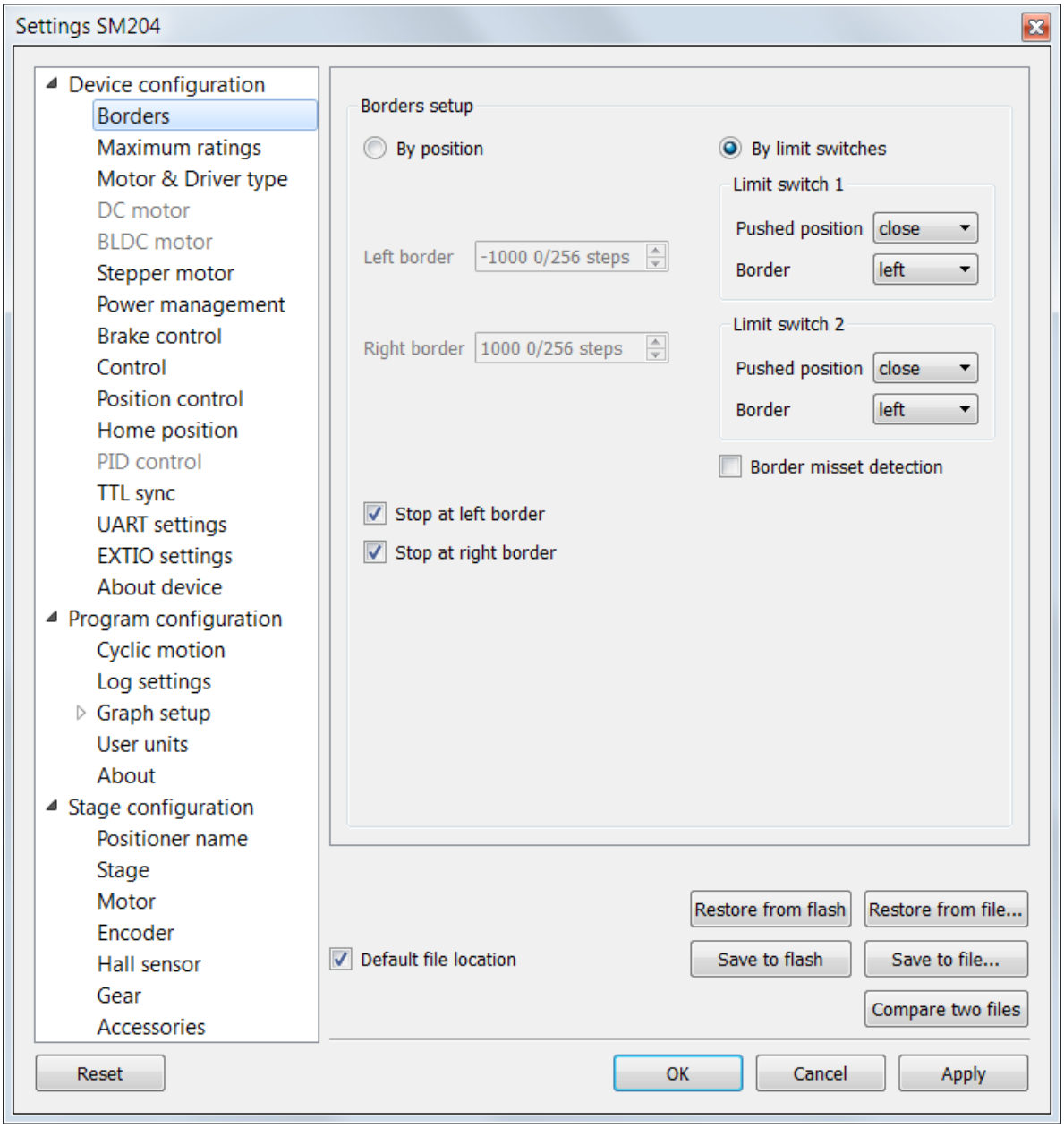
Начало работы в ПО XILab

XILab представляет собой удобный графический интерфейс пользователя для контроля работы, диагностики и настройки двигателей. Он также может быть использован для легкой установки и сохранения/загрузки параметров для любого двигателя. В этом разделе рассмотрено начало работы с XILab. Для получения полной информации о работе программы, рекомендуем Вам ознакомиться с разделом Руководство по программе XILab.



Главное окно программы XILab.

Откройте “Settings...”, нажмите на кнопку “Restore from file...” и выберите конфигурационный файл для вашего позиционера из открывшейся папки C:\Program Files\XILab\profiles\. Все поля меню “Settings...” автоматически будут заполнены значениями, подобранными для вашего позиционера. Если нужный файл не найден, посмотрите в разделе Конфигурационные файлы. Если и там вы не найдете решения, оставьте нам запрос на сайте поддержки.



XILab, меню Settings.

Предупреждение. Для работы контроллера **с шаговыми двигателями** обязательным является корректная установка:

- рабочего тока,
- границ движения и концевиков,
- критических параметров,
- ограничителей,
- режима питания мотора.

Если вы решили настроить контроллер самостоятельно, обязательно проверьте эти настройки!

Контроллер готов к работе!

Проверка работоспособности

Для проверки правильной настройки контроллера нажмите в главном окне XiLab клавишу влево или вправо в центральном ряду клавиш управления. Позиционер должен начать двигаться. Для остановки вращения используйте центральную клавишу плавной остановки. Обратите внимание на параметры питания контроллера в блоке *Power*. Там можно контролировать питающее напряжение, потребляемый ток и температуру контроллера.

Если при старте движения главное окно XiLab окрасилось в красный цвет, то это свидетельствует о сработавшей защите и попадании в режим Alarm. Это может быть вызвано неправильными настройками, неверным подключением позиционера или неисправностью контроллера. Подробней смотрите Критические параметры.

Управление из пользовательских приложений

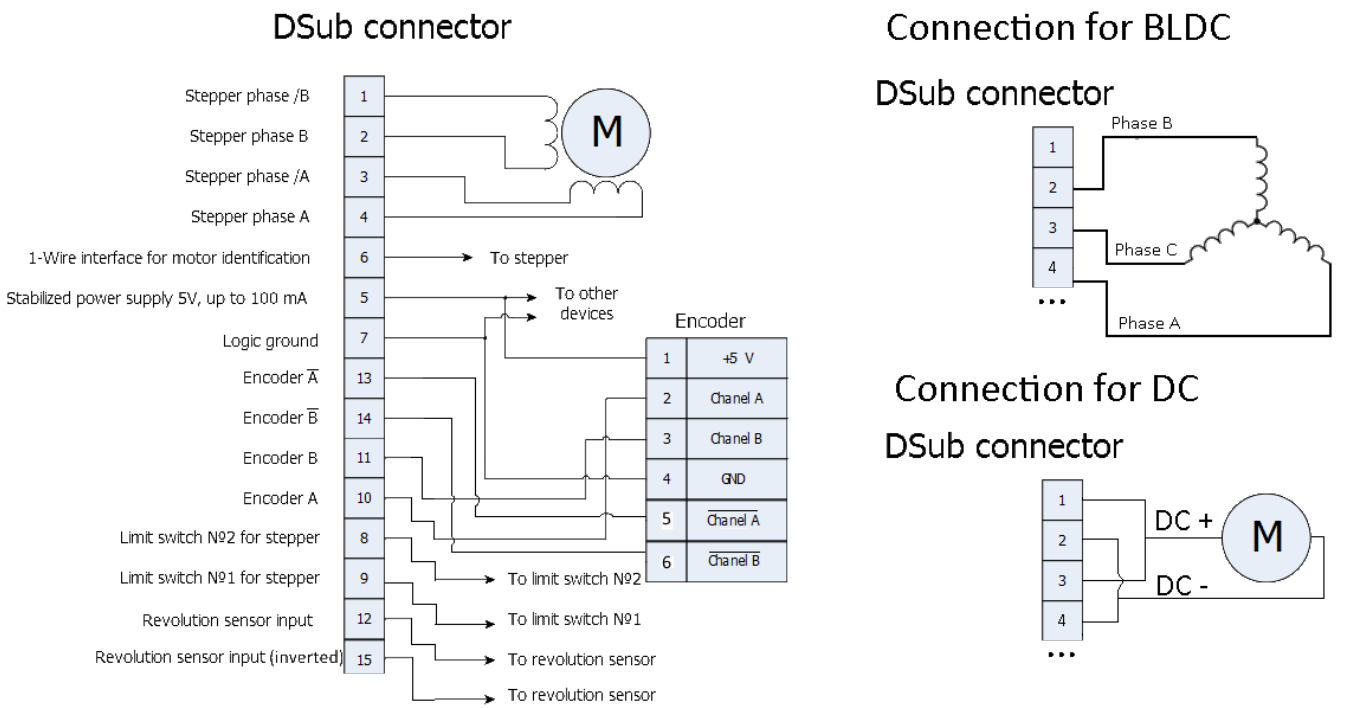
Для удобного управления контроллером Вы можете использовать программу Xilab. Однако, если у Вас есть необходимость управлять контроллером из собственных приложений, Вы легко сможете это сделать, используя функции библиотеки libximc. В комплекте разработчика библиотеки представлены примеры на различных языках программирования: C, C#, Delphi, Java, VB.net, Matlab, LabView. Если Вам необходимо автоматизировать небольшое количество действий, то возможно вместо разработки собственной программы Вы сочтете более целесообразным использовать для этого скриптовый язык программы Xilab.

3.2. Пример подключения простого мотора

Общий случай

Для того, чтобы подключить мотор к контроллеру, необходимо знать распиновку разъёма подключения позиционера, а также типовую схему подключения мотора к контроллеру:

3.2. Пример подключения простого мотора
Общий случай
Пример
Подготовка
Подключение мотора и энкодера к контроллеру



Общая схема подключения позиционера с энкодером через разъем Dsub.

> *Замечание.* Если каналы A и B энкодера работает в режиме открытого коллектора, то для обеспечения максимальной частоты передачи сигнала на высоких скоростях вращения могут потребоваться дополнительные подтяжки выходов энкодера к питанию 5 В резисторами (см. Работа с энкодерами).

> *Замечание.* Управление BLDC моторами возможно только в прошивках 4.1.0 и старше.

Пример

Рассмотрим подключение двигателя с энкодером CUI INC AMT112S-V к контроллеру 8SMC5-USB на примере двухфазного шагового мотора Nanotec ST5918L3008-B

Подготовка

Для начала работы нам понадобятся:

- Мотор;
- Энкодер;
- Распиновка D-SUB разъёма контроллера 8SMC5-USB;
- Спецификация на мотор ;
- Спецификация на энкодер ;
- Паяльное оборудование: паяльник, провода, флюс, припой, кусачки, термоусадочные трубки разных размеров;
- Винты M2.5x6 для крепления энкодера;
- D-SUB корпус + разъём (male) и провода для изготовления своего кабеля:



Подключение мотора и энкодера к контроллеру

- Прежде чем начать работу, соберите энкодер в соответствие с инструкцией по сборке, прилагаемой к нему

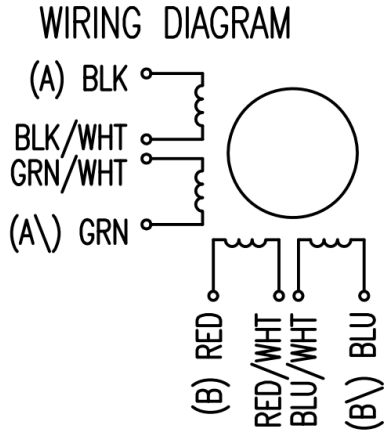


Мотор без энкодера. Обратите внимание на 2 отверстия M2.5 к которым обычно крепится энкодер



Мотор с прикреплённым энкодером

- Смотрим в спецификацию мотора и находим маркировку выводов (для Nanotec ST5918L3008-B она находится справа внизу в спецификации):



Контакты мотора

TYPE OF CONNECTION (EXTERN)				MOTOR		
UNIPOLAR	BIPOLAR 1WINDING	BIPOLAR SERIAL	BIPOLAR PARALLEL	CONNECTOR PIN NO.	LEADS	WINDING
A	A	A	A	1	BLK	A
COM	A			3	BLK/WHT	
A\		A\	A\	2	GRN/WHT	A\
B	B	B	B	4	GRN	
COM	B			5	RED	B
B\		B\	B\	7	RED/WHT	
				6	BLU/WHT	B\
				8	BLU	

Тип соединения

  - Существует последовательное и параллельное соединение обмоток, причём каждое из соединений позволяет получить свои характеристики для мотора. Мы соединим обмотки последовательно (на картинке выше обозначено красным). Для этого провода, имеющие два цвета BLK/WHT и GRN/WHT, а также RED/WHT и BLU/WHT надо соединить между собой попарно. Далее необходимо поставить в соответствие A, не A, B, не B контактам разъёма контроллера, контакты обмоток мотора ST5918L3008-B: чёрный, зелёный, красный, голубой. Одна обмотка - это соединение A - не A или B - не B. После соединения между собой двухцветных проводов, получится что одна обмотка мотора - это соединение чёрный - зелёный, а другая красный - голубой. Поэтому соответствие контактов будет

таким: чёрный- A, зелёный - не A, красный - B, голубой - не B. Это же соответствие видно на картинке *Тип соединения* выше.

- Для подключения энкодера, откройте спецификацию на энкодер и найдите на его разъёме 5 контактов: A+ (канал A), B+ (канал B, сдвинутый относительно A на 90 град), Z+ (счётчик оборотов), 5V, GND. Их надо вывести от энкодера 5 проводами и пустить вместе с проводами от мотора, т.к. далее они пойдут на один разъём. Энкодер CUI INC AMT112S-V имеет 18 пиновый вход, поэтому надо сделать кабель с таким же разъёмом на конце, чтобы вывести необходимые сигналы:



Контакты энкодера A+, B+, Z+, 5V и GND соответствуют контактам 10, 11, 12, 5, 7 D-SUB male разъёма соответственно.

Для удобства воспользуйтесь таблицами подключения к D-SUB разъёму (в скобках указан номер пина на соответствующем разъёме):

Контакты энкодера	**Контакты D-SUB**
A+ (10)	Энкодер A(10)
B+ (8)	Энкодер B(11)
Z+ (12)	Вход датчика оборотов (12)
5V (6)	Выход 5В, 100 мА (5)
GND (4)	Земля логическая (7)

Контакты мотора	**Контакты D-SUB**
A (BLK)	ШД фаза A (4)
не A (GRN)	ШД фаза не A (3)
B (RED)	ШД фаза B (2)
не B (BLU)	ШД фаза не B (1)

- Припаяйте вышеуказанные контакты к D-SUB male разъёму:



Провода от мотора и энкодера, собранные при помощи термоусадочной трубки. Обратите внимание на наличие термоусадочных трубок малого размера на проводах, идущих к обмоткам мотора (BLK, GRN, RED и BLU), а также на соединённые вместе двухцветные провода (BLK/WHT и GRN/WHT, RED/WHT и BLU/WHT). Тоненькие проводки - это контакты энкодера. Их 5 штук.



Припаянные контакты обмоток мотора



Готовый провод, идущий от мотора с D-SUB разъёмом на конце
Рекомендация: используйте термоусадочные трубки малого диаметра (2-3 мм) при пайке контактов к D-SUB разъёму и большого диаметра - для того, чтобы через них пропустить все провода, идущие от мотора и энкодера. Надевайте трубки до пайки.

- Сверху на разъём одевается защитный кожух.



- Теперь мотор можно подключить к контроллеру 8SMC5-USB.

Описание и настройки профиля дана в следующей главе Ручная настройка профиля.

3.3. Решение возможных проблем

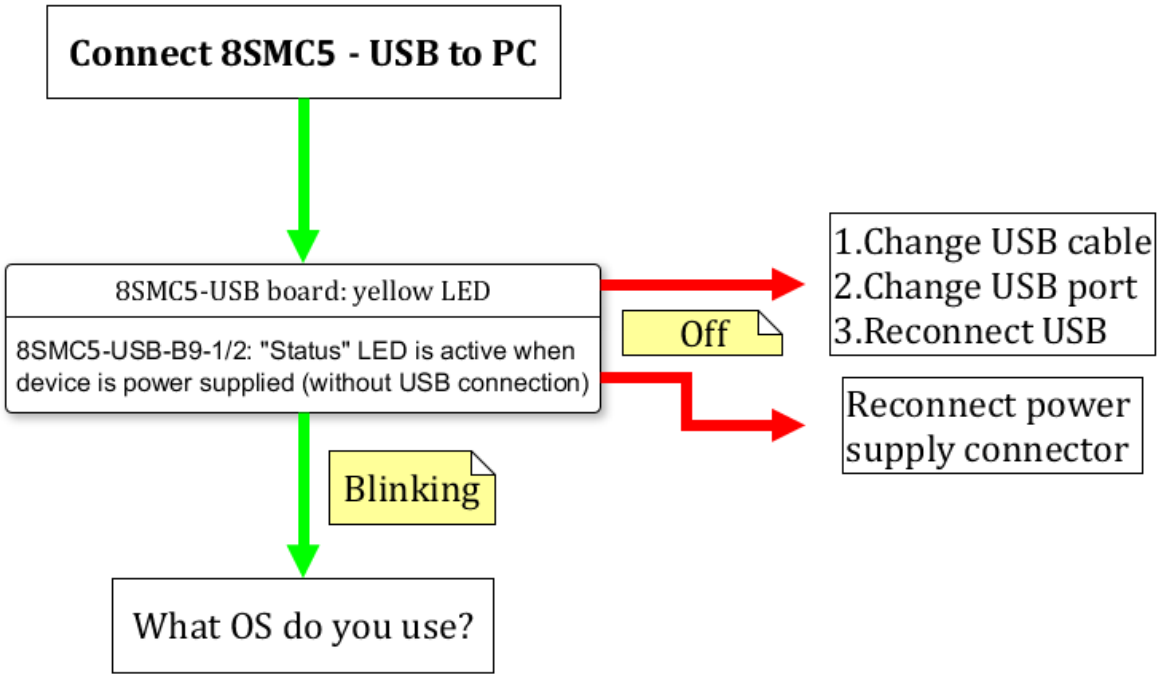
3.3. Решение возможных проблем
Контроллер не обнаруживается в XiLab
Не удаётся вращать мотором при помощи контроллера
Контроллер в состоянии Alarm
Мотор вибрирует, вращения нет
Механическое заклинивание
Зависание операционной системы при использовании библиотеки libximc и ядра Linux с версией менее 3.16

Ниже перечислен список часто встречающихся проблем и возможных решений, которые мы предлагаем пользователям наших контроллеров. Ваш случай может совпадать с нижеперечисленным, тогда вы быстро найдёте решение, в противном случае направляйте свои вопросы с подробным описанием проблемы в техподдержку:

- Задать вопрос
- Написать на почту: 8smc4@standa.lt

Контроллер не обнаруживается в XiLab

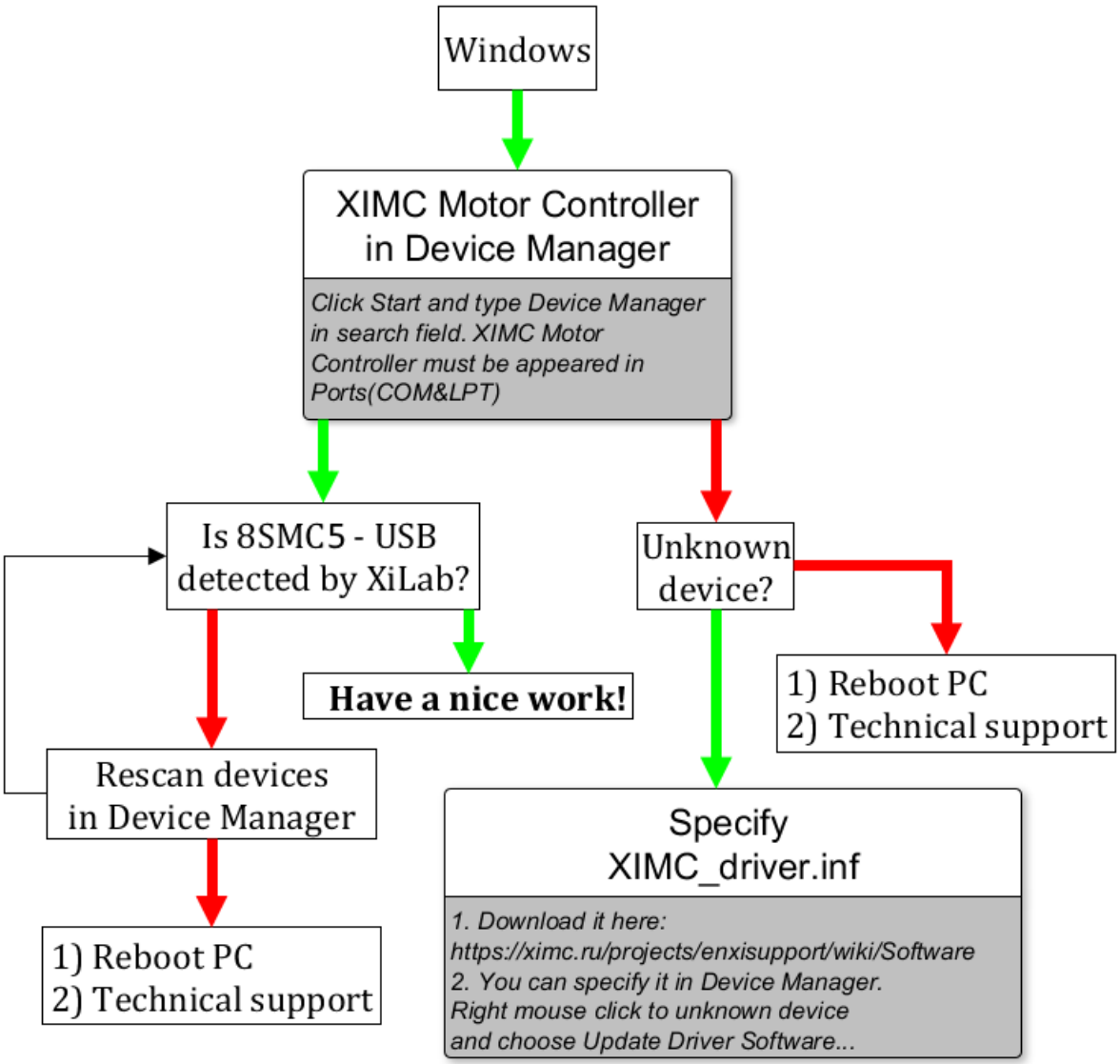
Решение:



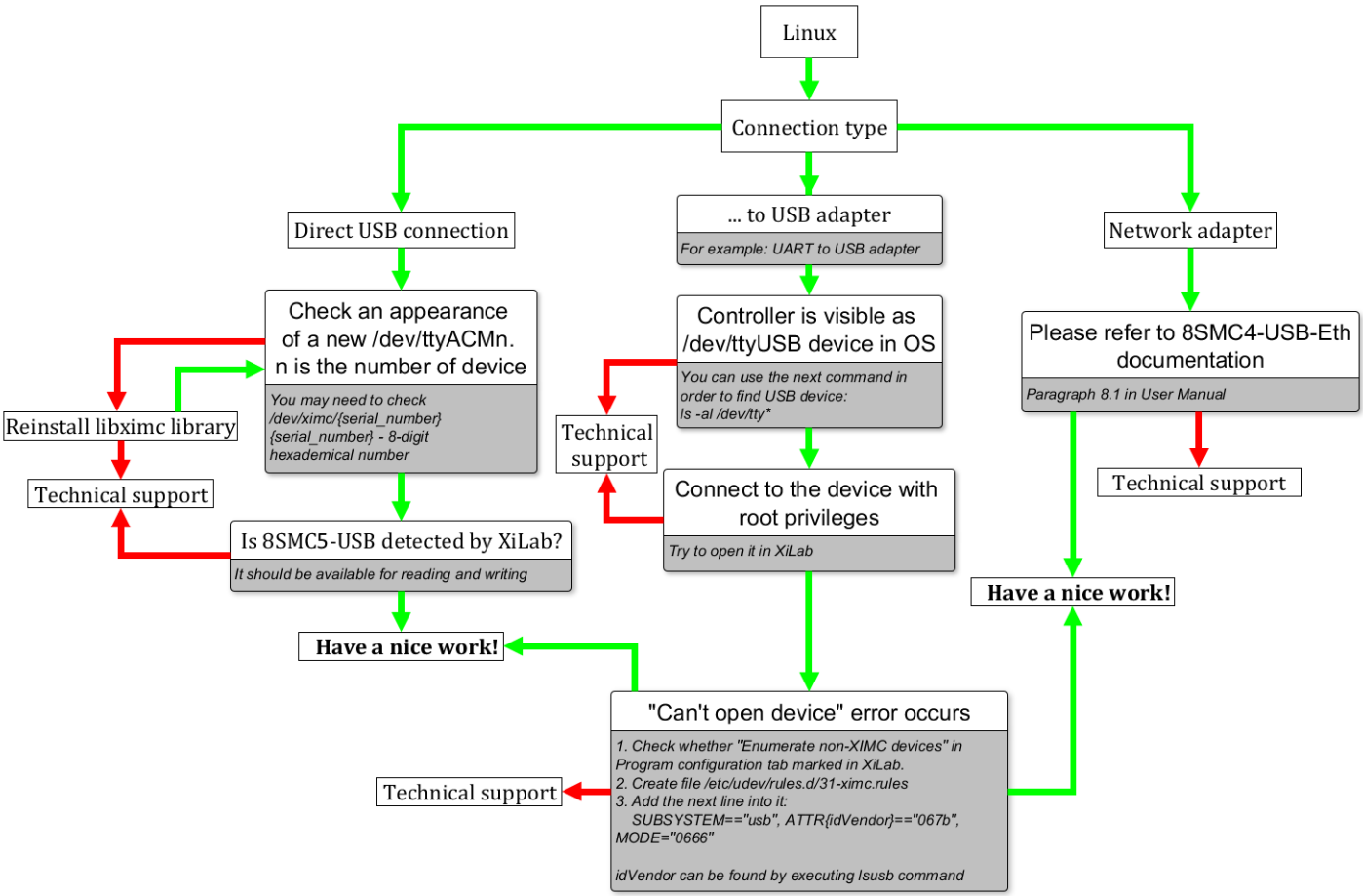
Комментарий к диаграмме:

8SMC5-USB-B9-1 - одноосная система
8SMC5-USB-B9-2 - двухосная система

Ниже изображены карты действий для разных операционных систем.
Windows:


Windows
XIMC Motor Controller in Device Manager
Click Start and type Device Manager in search field. XIMC Motor Controller must be appeared in Ports(COM&LPT)
Is 8SMC5 - USB detected by XiLab?
Unknown device?
Have a nice work!
1) Reboot PC
2) Technical support
Rescan devices in Device Manager
Specify XIMC_driver.inf
1. Download it here: https://ximc.ru/projects/enxisupport/wiki/Software
2. You can specify it in Device Manager. Right mouse click to unknown device and choose Update Driver Software...
1) Reboot PC
2) Technical support

Linux:



Комментарий к решению проблемы "Can't open device" (2 ветка):

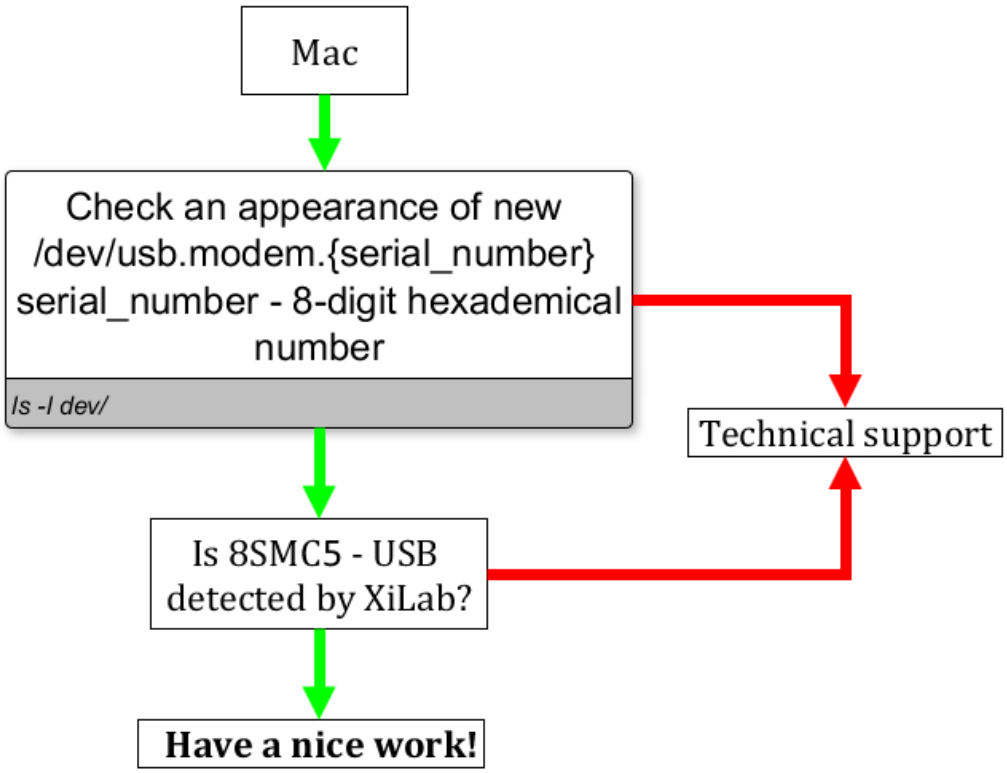
В **Linux**, при работе с контроллером через USB-... переходник, появляется устройство **/dev/ttyUSB**. XiLab отображает его в списке, но при попытке открыть возникает ошибка "can't open device" из-за отсутствия соответствующих прав доступа к устройству.

Для решения данной проблемы, создайте файл: /etc/udev/rules.d/31-ximc.rules и добавьте в него следующую строку:

```
SUBSYSTEM=="usb", ATTR{idVendor}=="067b", MODE="0666"
```

Идентификатор idVendor можно найти с помощью команды lsusb.

Mac OS:



Не удаётся вращать мотором при помощи контроллера

Контроллер в состоянии Alarm

Попробуйте нажать Stop в главном окне XiLab. Это убирает состояние Alarm. Если данный способ не помогает и Alarm появляется снова, то:

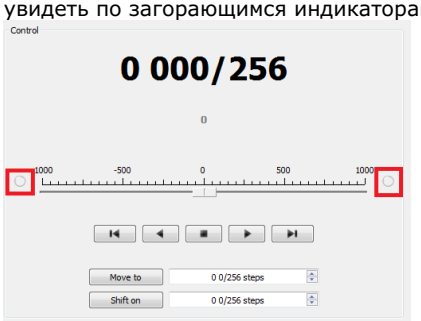
- Находясь в XiLab, перейдите во вкладку Maximum ratings
- Поставте галочку напротив опции Sticky Alarm flags. Нажмите Ок.
- Нажмите Stop в главном окне XiLab, выйдя из состояния Alarm. Повторите последовательность действий, которая привела к Alarm.
- Сделайте скриншот главного окна XiLab и отправьте его в техподдержку с подробным описание проблемы.

Мотор вибрирует, вращения нет

У данной проблемы может быть несколько причин:

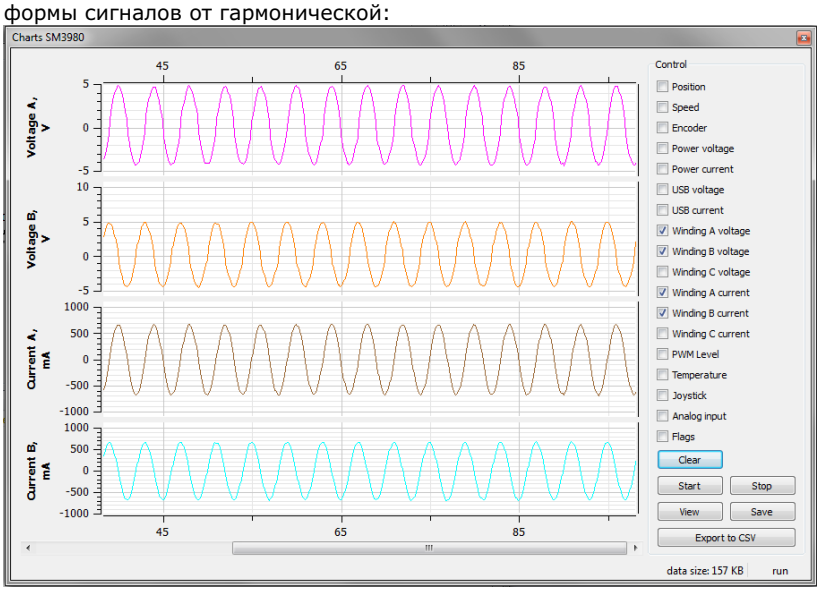
- Установлен **некорректный профиль** для вашего мотора/подвижки.
  - Поищите наиболее совпадающий по названию с используемой подвижкой профиль в папке с профилями в XiLab и на сайте ruxisupport.
  - Рекомендуется сохранить текущую конфигурацию в файл. Для этого необходимо в окне Settings программы XiLab нажать Save to file (см. настройки программы XiLab), выбрать путь, куда сохранить настройки. Затем этот файл отправить в техподдержку с описанием проблемы.

- **Неправильно настроены концевые выключатели**, в результатет чего подвижка уехала в концевик. Обычно это можно увидеть по загорающимся индикаторам в XiLab.



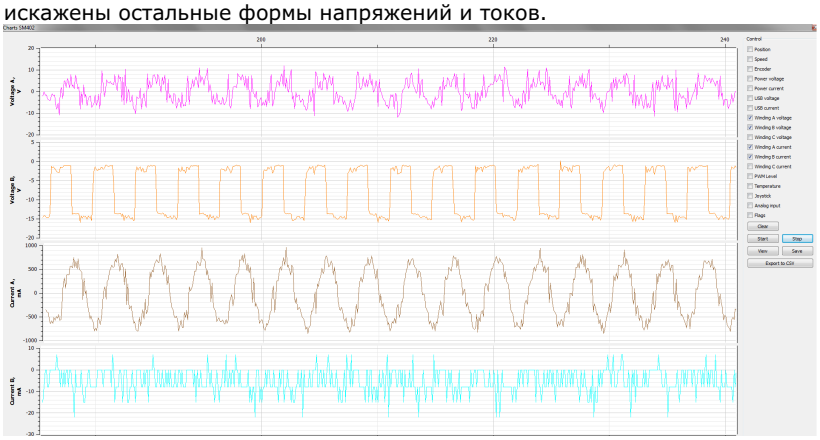
Основной причиной некорректной настройки концевиков является ошибочный профиль конфигурации для позиционера (см. предыдущий пункт). Информация по самостоятельной настройке находится в разделе ручная настройка профиля. При проблеме такого рода рекомендуется обратиться в техподдержку за дополнительной помощью.
  - Также одним из проявлений проблемы с концевиками может быть **механическое заклинивание** (см. следующий пункт).

- **Сгоревшая обмотка** мотора, проблемы с **контактом в разъёме** и т.п. Диагностировать проблемы такого рода можно самостоятельно. Для этого в XiLab есть возможность вывести графики напряжения и тока во время работы мотора. В исправном моторе ток в обмотках меняется по синусу или косинусу. В сломанном моторе будут заметны сильные отличия формы сигналов от гармонической:



Исправный случай

На графиках ниже видны проблемы. Например, отсутствует ток через обмотку B. Вероятно, в ней имеется разрыв. Также искажены остальные формы напряжений и токов.



Имеются проблемы с мотором

Для диагностики проблемы установите маленькую рабочую скорость (**1 ш/с** - оптимально) и подайте команду движения. Далее включите вывод графиков напряжений и токов в обмотках A и B в XiLab (кнопка Charts, далее отметьте галочками). Подождите некоторое время, пока графики построятся. После этого, рекомендуется отправить их (кнопка Save для сохранения) в техподдержку с подробной формулировкой проблемы. Иногда, при сгоревшей обмотке, невозможно пользоваться XiLab из-за постоянной потери устройтва и вывод графиков становится невозможен. В этом случае также обратитесь в техподдержку с подробным описанием проблемы.

Механическое заклинивание

Существует два способа справиться с заклиниванием в позиционере:

- Провернуть подвижку руками если это возможно.
- Увеличить ток в обмотке в 2-3 раза на короткое время (до 5-10 секунд) и подать команду движения в нужную сторону на невысокой скорости (**50-100 ш/с** - разумное значение). Через несколько секунд после вращения нажимать клавишу остановки (чёрный квадратик) в главом окне XiLab, до тех пор, пока не появится статус power off. Это позволит избежать перегрева мотора. **После выполнения данной операции не забудьте вернуть настройки обратно!**

Зависание операционной системы при использовании библиотеки libximc и ядра Linux с версией менее 3.16

Комментарий: проблема является следствием ошибки в драйвере последовательного порта cdc-acm. Наблюдается при частом последовательном открытии и закрытии нескольких устройств. Зависание проявляется на Debian 7 (ядро 3.2), не проявляется на Debian 8 (ядро 3.16).
Дополнительная информация о проблеме находится по следующей ссылке

Решение: обновление текущей версии Linux.

3.4. Ручная настройка профиля

Введение

3.4. Ручная настройка профиля
Введение
Подготовка к работе
Настройка рабочего тока
Настройка базовых параметров
Настройка аппаратных концевых выключателей, процедура автокалибровки
Настройка параметров энкодера
Настройка кинематических характеристик контроллера
Работа с пользовательскими единицами измерения

После подключения мотора, настраиваются необходимые для работы параметры. Рассмотрим настройку профиля на примере шагового мотора **Nanotec ST5918L3008-B**.

Подготовка к работе

- Устанавливаем и запускаем XiLab (см. раздел Краткое руководство и начало работы).
- Загружаем профиль с выставленными по умолчанию настройками. Для этого откройте Settings -> Restore from file... и выберите xilabdefault.cfg, лежащий в корне папки с XiLab.

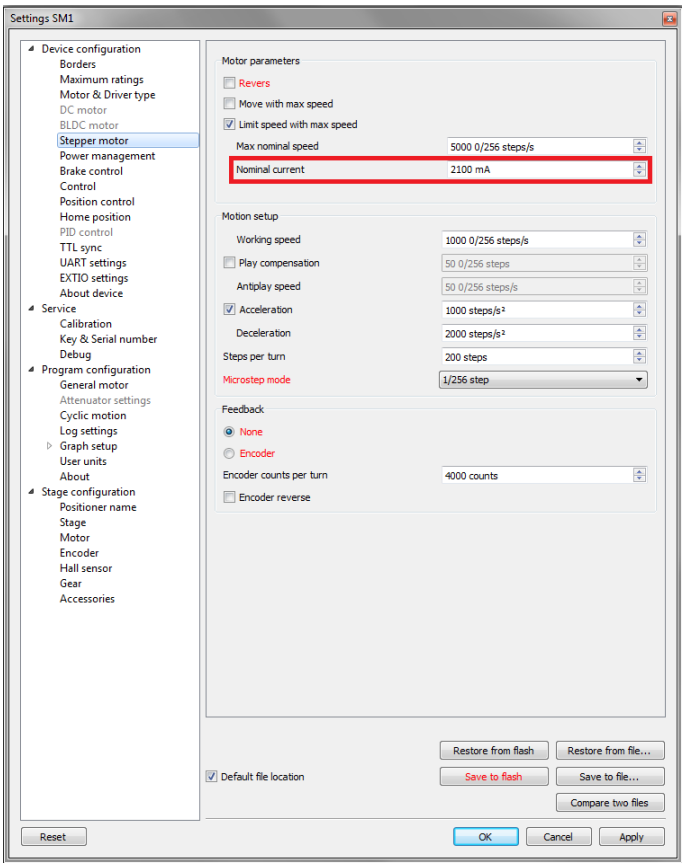
Настройка рабочего тока

Первоначально необходимо настроить правильный ток в обмотках двигателя:

- Из спецификации находим **ток фазы 2.1 А** - это максимальное значение тока для данного мотора при соединении обмоток **последовательно**:

SPECIFICATION \ CONNECTION	UNIPOLAR OR BIPOLAR–1 WINDING	BIPOLAR SERIAL	BIPOLAR PARALLEL
VOLTAGE (VDC)	3.0		
AMPS/PHASE	3.0	2.1	4.2
RESISTANCE/PHASE (Ohms)@25°C	1.0±10%	2.0±10%	0.5±10%
INDUCTANCE/PHASE (mH) @1KHz	2.2±20%	8.8±20%	2.2±20%

- Находясь в окне Settings, откройте вкладку Stepper motor. Тут задаются такие параметры, как скорость вращения, ускорение, режим движения и др (подробнее см. Настройка кинематики движения (Шаговый двигатель)). В поле Motor parameters-> Nominal current выставляется ток фазы для мотора. В это поле нужно вписать значение **не превышающее 2.1 А**:



Настройка базовых параметров

- Укажем скорость вращения в поле Working speed. Рекомендуемая величина скорости не более **1000 ш/с** при первом запуске. В том же окне укажите Max Nominal Speed (**5000 ш/с** для большинства моторов и позиционеров является разумным значением) и установите галочку напротив Limit speed with max speed. Данная настройка нужна, чтобы ограничить скорость мотора, т.к. некоторые механические системы могут быть не рассчитаны на высокие скорости и

слишком быстрое вращение может привести к сильному износу механики подвижки/мотора.

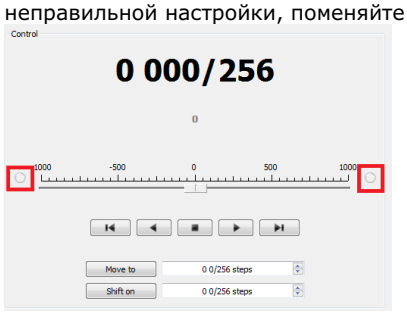
- Из спецификации мотора, найдём количество шагов на оборот. Для нашего мотора это значение равно **200 ш**. Укажем его в поле Steps per turn. Обычно в описаниях к мотору приводится величина одного шага в градусах, исходя из которой можно расчитать количество шагов на оборот, зная, что в одном обороте 360 градусов.
- Проверьте, что при старте движения вправо из главного окна XiLab подвижка физически тоже движется вправо. Если это не так, то поставьте галочку Reverse в поле Stepper motor -> Motor parameters.

Настройка аппаратных концевых выключателей, процедура автокалибровки

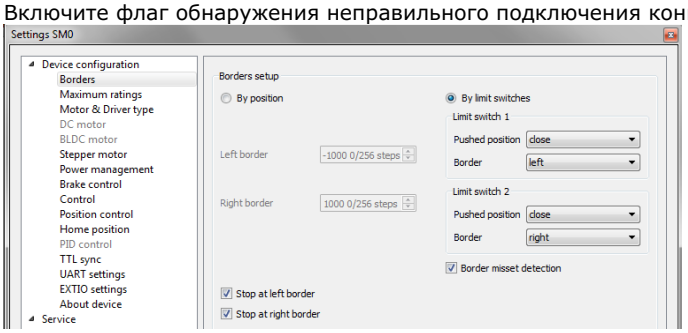
> **Замечание.** Данный пункт подразумевает использование позиционеров с аппаратными концевыми выключателями. Если в вашей системе аппаратные концевые выключатели не предусмотрены, то рекомендуется отключить остановку по концевикам в настройках. Для этого следует убрать галочки Stop at right border и Stop at left border во вкладке Borders.

Позиционеры бывают с ограниченным (трансляторы) и с неограниченным диапазоном движения (ротаторы). Ограничение диапазона перемещения может осуществляться с помощью концевых выключателей или по позиции. При работе с позиционерами первого типа при неправильной настройке концевых выключателей существует риск сломать механику, т.к. подвижная часть может попытаться выехать за пределы допустимого диапазона движения. У ротаторов такой проблемы нет. Также следует иметь ввиду, что у ротаторов может быть всего один концевик.

- Для работы с концевыми выключателями контроллеру необходимо указать какой из них будет левым, а какой - правым. Иногда это заранее неизвестно, а известно лишь, что оба концевика подключены и срабатывают каждый по достижении своей границы перемещения. Если неправильно настроить концевики, то позиционер может заклинить. Поэтому контроллер поддерживает простую функцию обнаружения неверно настроенных концевиков, останавливаясь по обоим из них. Убедитесь, что:
  - Подвижная часть находится вдали от концевиков;
  - Полярность концевиков настроена правильно (индикаторы концевиков не горят в главном окне XiLab). В случае неправильной настройки, поменяйте их полярность (Borders -> Pushed position), индикаторы должны погаснуть.



  - Включена остановка по обоим концевикам (галочки напротив Stop at right border и Stop at left border во вкладке Borders).
- Включите флаг обнаружения неправильного подключения концевиков Border misset detection во вкладке Borders.



Вкладка с настройками концевиков

- При обнаружении неверного срабатывания концевика, контроллер может перейти в режим Alarm, если включена настройка Enter Alarm state when edge misset is detected в меню Maximum ratings. Рекомендуется включить эту опцию. Начните движение в любую сторону из главного окна XiLab до остановки по концевику или перехода в режим Alarm. При возникновении Alarm нужно поменять концевики местами, изменив значения в полях Borders->Border на противоположные.

> **Предупреждение.** Защита от перепутанных концевиков не гарантирует, что о проблеме перепутанности можно забыть. Она лишь облегчает первоначальную настройку. При перепутанных концевиках нельзя начинать движение, если какой-либо концевик активен, даже при включенной функции защиты.

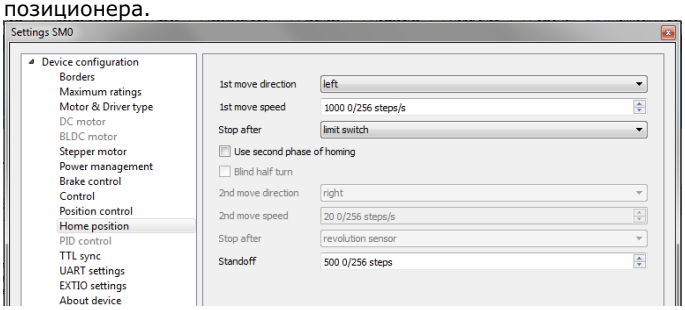
Существует ещё 2 способа определения какой из концевиков правый, а какой левый:

- Необходимо знать куда подсоединён каждый из концевиков в позиционере. При загрузке профиля с настройками по умолчанию, концевик, подсоединённый к пину 9 D-SUB разъёма контроллера, считается левым, а к пину 8 - правым. Их расположение относительно позиционера настраивается в полях Limit switch 1 и Limit switch 2 (см. скриншот выше). Запустите систему на маленькой скорости (<100 ш/с) вдали от концевиков. Если направление движения к концевику отличается от ожидаемого, измените значения в полях Borders->Border на противоположные.

- Если возможно подлезть к концевикам, то попробуйте активировать их и увидеть в главном окне XiLab загорающиеся индикаторы. Запомните, какой из индикаторов, какому концевику соответствует. Затем, вдали от концевиков, запустите систему на маленькой скорости (<100 ш/с) и убедитесь, что система движется в направлении срабатывания правильного выключателя. Соотнесите это с тем, что видите в главном окне XiLab. Если направление движения к концевику в реальной установке и в главном окне различаются, измените значения в полях Borders->Border на противоположные

Для более подробной информации о концевых выключателях см. раздел настройка диапазона движения и концевых выключателей.

- В контроллере предусмотрена полезная функция автокалибровки домашней позиции для установки начального положения позиционера.



Рассмотрим наиболее простой вариант настройки с одной фазой движения. Начать следует с установки скорости первой фазы (1st move speed) примерно в 5-10 раз меньшей, чем Working speed. Это нужно для более высокой точности процедуры автокалиборовки. В поле Stop after укажите limit switch, чтобы в процессе автокалибровки подвижка доезжала до одного из концевиков (до правого или левого - выбирается в поле 1st move direction). В поле Standoff укажите число в шагах, на которое подвижка должна отъехать от концевика. Нажмите Ok или Apply.

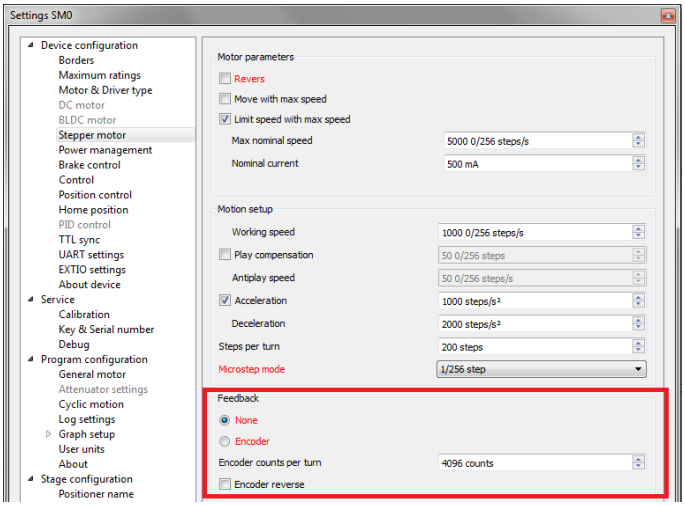
> *Замечание*. Значение в поле Standoff является знаковым. Положительное направление - право. То есть, если процедура автокалибровки настроена по правому концевику, то для того, чтобы подвижка отъехала от него влево в поле Standoff должно быть отрицательное значение.

- Запустите процедуру автокалибровки, нажав на кнопку Go home в главном окне XiLab. Результатом будет движение подвижки до указанного концевика с относительно низкой скоростью и смещение от него в сторону на значение, указанное в поле Standoff.
- После выполнения автокалибровки, нажмите Zero в главном окне XiLab для установки начала системы отсчёта.
- Повторите процедуру калибровки второй раз. Подвижка, при этом, должна снова вернуться в нулевую позицию. Обратите внимание, что могут быть небольшие отклонения от нуля, связанные с погрешностью процедуры автокалибровки.

Настройка параметров энкодера

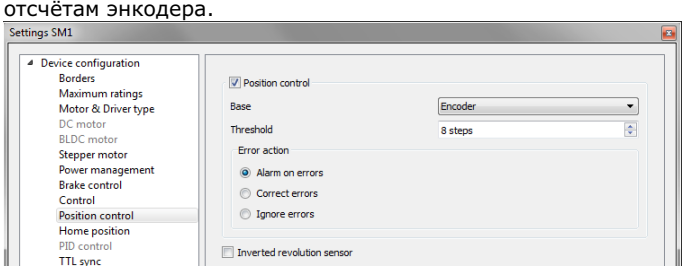
> **Замечание.** Данный пункт подразумевает использование мотора с энкодером. Если у вас двигатель без энкодера, то описанные ниже параметры можно оставить без изменений.

- Каждый энкодер имеет характеристику, указывающую **количество импульсов на оборот** (**Pulse Per Turn** - **PPT**). Для корректной работы энкодера с контроллером, необходимо в интерфейс XiLab, в поле Encoder counts per turn (вкладка Stepper motor) вписать количество отсчётов энкодера на оборот, которое равно **4xPPT**. Например, если ваш энкодер имеет **1024** импульса на оборот, то в поле Counts per turn вписывается число **4096**:



- Запустите вращение из главного окна XiLab. Если всё настроено правильно, то внизу окна загорится зелёным индикатор ENCD. В случае, если ENCD имеет жёлтый цвет, то следует поставить галочку Encoder reverse во вкладке Stepper motor, в поле с настройками энкодера. Красный цвет EDCN указывает на наличие проблем с пересчётом позиции по энкодеру.

- Существует возможность активировать контроль позиции по энкодеру. Для этого во вкладке Position control следует поставить галочку Position control и в поле Threshold указать допустимую ошибку в отсчётах энкодера. Тогда при рассогласовании позиции с отсчётами энкодера, в главном окне XiLab будет загораться индикатор SLIP и, если стоит галочка Alarm on errors, контроллер будет переходить в состояние Alarm. Установка Correct errors позволяет контроллеру работать в режиме ведущего энкодера, компенсируя разницу между реальной позицией и позицией, соответствующей отсчётам энкодера.

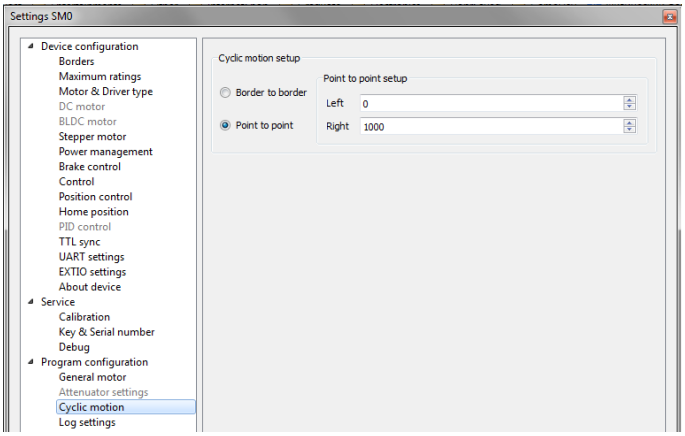


Настройка кинематических характеристик контроллера

- Во вкладке Stepper motor можно указать необходимое ускорение (Acceleration) и замедление(Deceleration) для используемого шагового двигателя. Процедура выбора оптимальных значений следующая:
  - Начиная со значений по умолчанию, делайте небольшие сдвиги мотора (старт и быстрый стоп), постепенно увеличивая Acceleration до тех пор, пока движение при этом не начнёт срываться или быть нестабильным. Примите ускорение равным примерно **половине** этого значения.
  - Замедление можно настроить примерно **в 1.5 - 2 раза больше**, чем ускорение.

- Если в вашей механической системе подъезд в желаемую позицию слева и справа неодинаков и присутствует люфт, то есть возможность устранить эту неоднозначность. Для этого поставьте галочку напротив Play compensation в Stepper motor и укажите значение, превышающее величину люфта. Знак этой настройки определяет направление подхода. Положительный знак означает подход слева, а отрицательный - справа. В поле Antiplay speed настройте скорость, с которой будет выполняться компенсационное движение. Её величина должна быть маленькой (значения **50 ш/с** достаточно), чтобы не было "заносов" во время антилюфта.

- После основной настройки позиционера/мотора, можно увеличить рабочую скорость. Процедуру настройки рабочей скорости можно провести экспериментально, подобно процедуре настройки ускорения, т.е. выбрать значение **примерно в 2 раза меньшее**, чем то, при котором наблюдается нестабильное движение. Для проверки стабильности вращения рекомендуется воспользоваться функцией Cyclic в интерфейсе главного окна, предварительно её настроив.



- В поле Microstep mode мы рекомендуем оставить значение **1/256** шага.

Работа с пользовательскими единицами измерения

Зачастую неудобно работать с шагами и микрошагами и хочется работать с более удобными единицами измерения. По этой причине в контроллере есть возможность пересчитывать координаты в привычные единицы измерения, например в миллиметры или градусы. Это делается во вкладке User units, где указывается величина шага и соответствующая ей единица измерения. Для более подробной информации обратитесь к соответствующему пункту документации.

Настройка рабочего профиля завершена.

3.5. Расчёт номинального тока

Для того чтобы шаговый двигатель выдавал максимальный вращающий момент, но при этом не перегревался, важно правильно задать такую техническую характеристику, как номинальный ток.

Чем больше ток в обмотке двигателя, тем больше вращающий момент на оси. Важно помнить, что с увеличением протекающего через обмотки тока, выделяемая тепловая мощность двигателя увеличивается. Чтобы двигатель мог работать длительное время выделяемая тепловая мощность (Закон Джоуля — Ленца) должна быть меньше мощности рассеяния. Мощность рассеяния можно рассчитать исходя из документации на двигатель.

Расчеты на базе параметров униполярного полношагового режима

Мощность рассеяния равна $P = n \cdot R_u I_u^2$, где **R_u** - сопротивление обмотки в униполярном режиме, **I_u** - ток через одну обмотку в униполярном режиме, **n** - колличество одновременно работающих обмоток.

Рассмотрим для примера ST2818M1006. Таблица в документации показывает, что в полношаговом режиме одновременно работает две обмотки (n = 2) в униполярном режиме, т.е. $P = 2R_u I_u^2$. Контроллеры моторов поддерживают только биполярный режим управления. Чтобы перейти от униполярного в биполярный режим, соединим обмотки каждой фазы последовательно, сопротивление возрастёт, $R_b = 2R_u$, где **R_b** - сопротивление последовательно соединенных обмоток для биполярного режима управления.

Алгоритм управления в контроллерах моторов работает в микрошаговом режиме и поддерживает ток так, что в одной обмотке ток меняется по функции $I_a \sin(\phi)$, в другой обмотке ток меняется по функции $I_a \cos(\phi)$, где **I_a** - амплитуда тока. Тепловая мощность выделяемая двумя обмотками в любой момент времени $P = R_b I_a^2 \sin^2(\phi) + R_b I_a^2 \cos^2(\phi) = R_b I_a^2$

Получим уравнение приравняв мощности, из которого найдём, что $I_a = I_u$

Расчеты на базе параметров биполярного полношагового режима

Мощность рассеяния равна $P = n \cdot R_b I_b^2$, где **R_b** - сопротивление обмотки в биполярном режиме, **I_b** - ток через одну обмотку в биполярном режиме, **n** - колличество одновременно работающих обмоток.

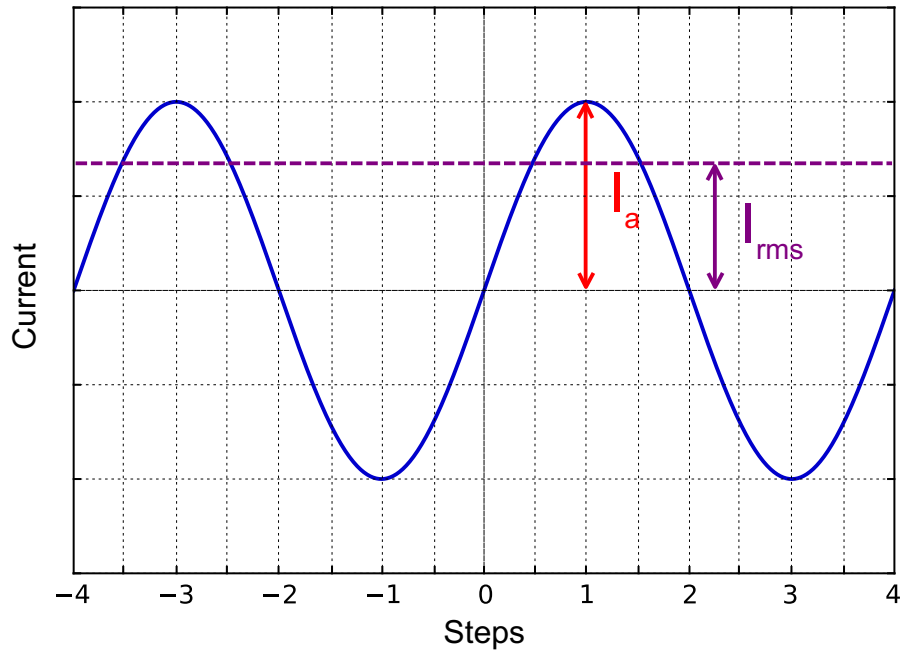
Рассмотрим для примера ST2018S0604. Таблица в документации показывает, что в полношаговом режиме одновременно работает две обмотки (n = 2) в биполярном режиме, т.е. $P = 2R_b I_b^2$

Тепловая мощность выделяемая на обмотках двигателя, управляемого контроллерами моторов, по прежнему равна $P = R_b I_a^2 \sin^2(\phi) + R_b I_a^2 \cos^2(\phi) = R_b I_a^2$

Получим уравнение приравняв мощности $2R_b I_b^2 = R_b I_a^2$. Найдем, что $I_a = \sqrt{2} \cdot I_b$

Связь со среднеквадратичным током

Переменный ток в каждой обмотке двигателя может характеризоваться своим среднеквадратичным значением за период $I_{rms} = \sqrt{\frac{1}{2\pi} \int_0^{2\pi} (I_a \sin(\phi))^2 \mathrm{d}\phi} = \frac{I_a}{\sqrt{2}}$. Тепловое выделение **одной** обмотки связано со среднеквадратичным током через неё $P_1 = R_b I_{rms}^2$. Обе обмотки идентичны $P_1 = P_2$. Общая тепловая мощность двигателя под управлением контроллера моторов $P = P_1 + P_2 = 2R_b I_{rms}^2$



Из вышеописанного найдём, что $I_{rms} = \frac{I_u}{\sqrt{2}}$, а также $I_{rms} = I_b$.

Настройка номинального тока

Контроллеры моторов способны принимать значение номинального тока в виде амплитуды тока $\mathbf{I_a}$ или в виде среднеквадратичного значения $\mathbf{I_{rms}}$. Выбор того каким способом интерпретировать входное значение номинального тока определяется отсутствием/наличием соответственно флага `ENGINE_CURRENT_AS_RMS` в поле *EngineFlags* структуры *engine_settings*. При настройке номинального тока в XILab следует правильно указывать способ интерпретации тока. Контроллеры моторов в этом случае будут обеспечивать максимальный допустимый момент, не перегревая двигатель.

Для всех моторизованных позиционеров *Standa* подготовленные конфигурационные файлы содержат номинальный ток заданный среднеквадратичным значением. Соответсвующий флаг установлен. Таким образом двигатели работают на оптимальных параметрах.

4. Техническое описание устройства

1. Внешний вид и разъемы
    1. Плата контроллера
    2. Одноосная система
    3. Двухосная система
2. Кинематика и режимы движения
    1. Движение с заданной скоростью
    2. Движение в заданную точку
    3. Смещение на заданное расстояние
    4. Движение с ускорением
    5. Компенсация люфта
    6. Реверсирование движения
    7. Рекомендации для точного движения
    8. ПИД алгоритм для управления DC двигателем
    9. Режимы остановки движения
3. Основные возможности контроллера
    1. Поддерживаемые типы двигателей
    2. Ограничители на двигателях
    3. Концевые выключатели
    4. Автокалибровка "домашней" позиции
    5. Работа с энкодерами
    6. Датчик оборотов
    7. Управление питанием мотора
    8. Критические параметры
    9. Хранение параметров во flash-памяти контроллера
    10. Пользовательские единицы координат
4. Безопасная работа
5. Дополнительные функции
    1. Индикация режима работы
    2. Работа с магнитным тормозом
    3. Управление с помощью джойстика
    4. Управление кнопками "вправо" и "влево"
    5. ТТЛ-синхронизация
    6. Создание многоосных систем
    7. Цифровой вход-выход общего назначения
    8. Аналоговый вход общего назначения
    9. Интерфейс управления внешним драйвером
    10. Последовательный порт
    11. Хранение позиции во FRAM-памяти контроллера
    12. Опознавание позиционеров Standa
6. Второстепенные функции
    1. Установка нулевой позиции
    2. Установка пользовательской позиции
    3. Статус контроллера
    4. Автовосстановление USB соединения
7. Совместимость с различным ПО
    1. Работа с MicroManager

4.1. Внешний вид и разъемы

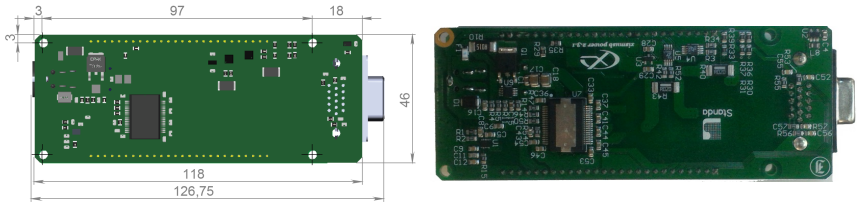
Контроллеры 8SMC5-USB выпускаются в 3 различных версиях:

1. Плата контроллера
2. Одноосная система
3. Двухосная система

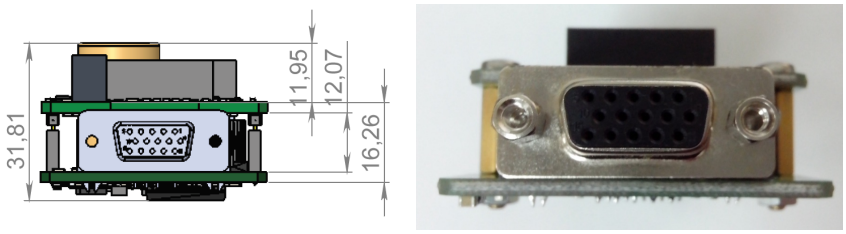
4.1.1. Плата контроллера

Геометрические размеры

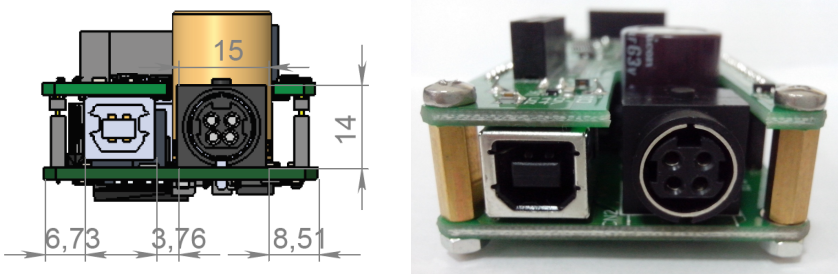
Конструктивно контроллер представляет собой две жестко соединенные печатные платы размером 46x126,75 мм. Нижняя плата содержит логический контроллер и схемы управления, верхняя - силовую часть.



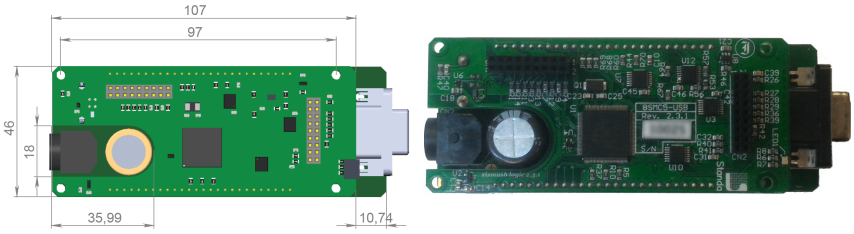
Контроллер. Вид сверху. Со стороны силового модуля и радиатора.



Контроллер. Вид спереди. Со стороны кабеля для подключения позиционера.



Контроллер. Вид сзади. Со стороны разъема для подключения питания и USB-b.



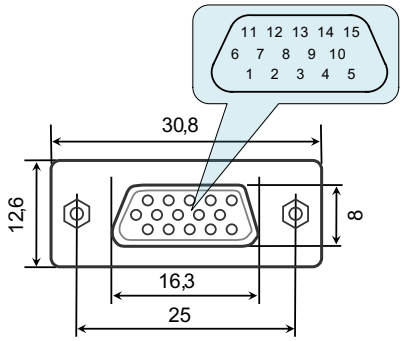
Вид снизу. Со стороны разъемов для подключения к объединительной плате.

ВАЖНО. В случае самостоятельной установки радиатора на силовой драйвер, убедитесь, что нет контакта между теплопроводящей поверхностью силового драйвера и другими электропроводящими частями установки. Если такой контакт будет присутствовать, возможно повреждение силовой схемы! Данное предупреждение относится только к контроллерам, поставляемым без внешнего корпуса.

Разъёмы

Разъем подключения позиционера

Для подключения позиционера используется разъем DSub 15 типа "мама".



Размеры и нумерация выводов в разъеме DSub, вид спереди.

Назначение выводов:
1 - ШД фаза не B или - DC-мотора
2 - ШД фаза B или + DC-мотора или фаза B BLDC мотора
3 - ШД фаза не A или - DC-мотора или фаза C BLDC мотора
4 - ШД фаза A или + DC-мотора или фаза A BLDC мотора
5 - Выход 5В, 500 мА, стабилизированное для питания энкодера
6 - Однопроводной интерфейс опознавания подвижки (работает только с позиционерами Standa)
7 - Земля логическая для концевиков, энкодера и прочего
8 - Концевик №2
9 - Концевик №1
10 - Энкодер A
11 - Энкодер B
12 - Вход датчика оборотов
13 - Инверсный канал энкодера A
14 - Инверсный канал энкодера B
15 - Инверсный вход датчика оборотов

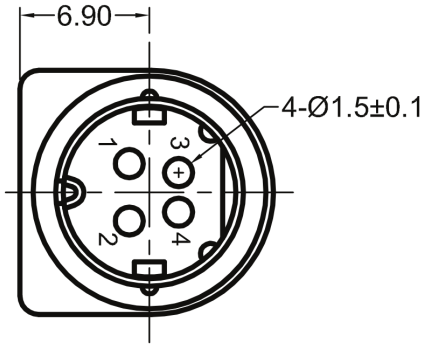
Замечание. BLDC моторы поддерживаются в прошивке версий 4.1.0 и старше.

Замечание. Для подключения DC мотора с рабочим током выше 3А нужно соединить выходы 1 и 3, а также выходы 2 и 4.

Предупреждение. Не рекомендуется присоединять к/отсоединять от контроллера двигатель пока есть питание на обмотках мотора.

Разъем силового питания для 1- и 2-осных систем

Одно- и двухосные системы, поставляемые в корпусе, используют разъём Kycon 4-pin DC power (part number по каталогу KPPX-4P, anacapa.kycon.com).





Назначение выводов:
1 - Силовое питание, "-".
2 - Силовое питание, "+". 12-48В.
3 - Силовое питание, "-".
4 - Силовое питание, "+". 12-48В.

ВАЖНО. Никогда не подавайте электропитание на контроллер и не подключайтесь к разъему силового электропитания, если вы не уверены, что параметры вашего блока питания соответствуют требуемым. Не пытайтесь подключить электропитание к контроллеру если не уверены, что разъемы на блоке питания и на корпусе контроллера совместимы друг с другом! Допустимые параметры подключения указаны в разделе техники безопасности.

ВАЖНО. Горячее присоединение/отсоединение, а также ненадежное подсоединение силового питания через разъём типа MF-4MRA может вывести из строя контроллер и/или компьютер. Подробнее смотрите в разделе техники безопасности.

Разъём управления 1- и 2-осной системами

Контроллеры в корпусе подключаются через разъём USB type-B.



Кабель USB-A - USB-B.



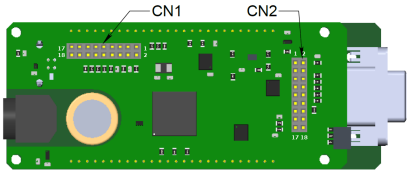
Разъем USB type B.

Таблица расположения выводов кабеля USB type-B

Номер вывода	Название	Цвет кабеля	Описание
1	VCC	Красный	+5 VDC
2	D-	Белый	Data -
3	D+	Зеленый	Data +
4	GND	Черный	Земля

Предупреждение. Используйте только проверенные и заведомо работоспособные USB-кабели! Неисправный или некачественный USB кабель может стать причиной неправильной работы контроллера, в том числе ошибок при вращении мотором или при опознавании устройства операционной системой. Для стабильного соединения используйте короткие кабели с толстыми проводниками и экранировкой.

Разъем подключения к объединительной плате



Расположение и нумерация разъемов CN1 и CN2 на плате контроллера версии 2.3.2.

Назначение выводов разъема CN1
1 - Земля питания 12-48В.
2 - Joy, аналоговый вход 0-3 В, используется для подключения внешнего джойстика.
3 - Земля питания 12-48В.
4 - Pot, аналоговый вход 0-3 В общего назначения.
5 - Земля питания 12-48В.
6 - Выход\вход альтернативной функции EXTGPIO0 (или сигнал Enable при использовании внешнего драйвера), 3.3В логика.
7 - Питание +12...+48В.
8 - U_LED, светодиод индикации режима работы контроллера (статус).
9 - Питание +12...+48В.
10 - L_LIM, светодиод индикации левого концевика.
11 - Питание +12...+48В.
12 - R_LIM, светодиод индикации правого концевика.
13 - USB_SEL, выбор USB устройства в многоосной системе.
14 - Выход\вход альтернативной функции EXTGPIO1.
15 - USB_D0_N, отрицательный выход шины USB Slave.
16 - H_USB_D0_N, отрицательный выход шины USB Master.
17 - USB_D0_P, положительный выход шины USB Slave.
18 - H_USB_D0_P, положительный выход шины USB Master.

Назначение выводов разъема CN2
1 - Зарезервировано.
2 - RX, последовательного порта, 3.3В логика.
3 - Зарезервировано.
4 - TX,последовательного порта, 3.3В логика.
5 - Зарезервировано.
6 - Сигнал Direction для управления внешним драйвером, 3.3В логика.
7 - Зарезервировано.
8 - Сигнал Clock для управления внешним драйвером, 3.3В логика.
9 - Зарезервировано.
10 - Выход синхронизации, 3.3В логика.
11 - Зарезервировано.
12 - Вход синхронизации, 3.3В логика.
13 - Выход для управления электромагнитным тормозом, 3.3В логика.
14 - Не используется.
15 - BUT_R, вход с внешней кнопки "Вправо".
16 - Выход питания 5В.
17 - BUT_L, вход с внешней кнопки "Влево".
18 - Цифровая земля для логики 3.3В и 5В.

Замечание. Неиспользуемые контакты внутреннего разъема не требуют никакого дополнительного подключения или подтяжки к земле/питанию. Просто не используйте их.

ВАЖНО. Аналоговые входы Joy, Pot рассчитаны на работу с напряжением **до** 3 В. Никогда не подавайте на эти входы большее напряжение, в том числе 3.3 В. Это может нарушить правильную работу всех аналоговых каналов контроллера и вывести из строя контроллер или двигатель.

ВАЖНО. Выводы, доступные на внутреннем разъеме не имеют защит, установка дополнительных буферов и фильтрующих цепочек целиком и полностью находится на ответственности пользователя или разработчика, создающего предназначенную для использования этих линий объединительную плату.

ВАЖНО. Внутренний разъем не допускает подачу каких-либо напряжений свыше 0.3 В относительно ножки DGND на обесточенный контроллер. Установка дополнительных защит, предотвращающих такие ситуации, если они возможны, целиком и полностью находится на ответственности пользователя или разработчика, создающего предназначенную для использования этих линий объединительную плату.

4.1.2. Одноосная система

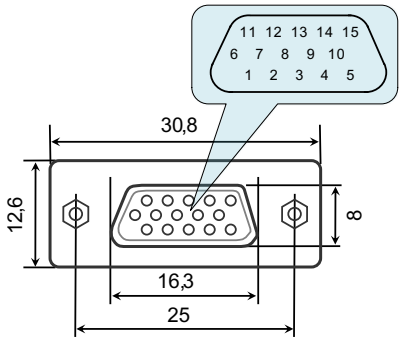
Корпус

Одноосная версия контроллера представляет собой плату контроллера в металлическом корпусе. Размеры корпуса: 66 x 46 x 114 мм.
На передней панели расположен разъем силового питания, разъём для подключения к компьютеру тип USB-B, светодиоды "стасус контроллера", "питание", "левый концевик", "правый концевик", кнопки для движения вправо и влево.
На задней панели расположен разъем подключения позиционера

Разъёмы

Разъем подключения позиционера

Для подключения позиционера используется разъем DSub 15 типа "мама".



Размеры и нумерация выводов в разъеме DSub, вид спереди.

Назначение выводов:
1 - ШД фаза не B или - DC-мотора
2 - ШД фаза B или + DC-мотора или фаза B BLDC мотора
3 - ШД фаза не A или - DC-мотора или фаза C BLDC мотора
4 - ШД фаза A или + DC-мотора или фаза A BLDC мотора
5 - Выход 5В, 500 мА, стабилизированное для питания энкодера
6 - Однопроводной интерфейс опознавания подвижки (работает только с позиционерами Standa)
7 - Земля логическая для концевиков, энкодера и прочего
8 - Концевик №2
9 - Концевик №1
10 - Энкодер A
11 - Энкодер B
12 - Вход датчика оборотов
13 - Инверсный канал энкодера A
14 - Инверсный канал энкодера B
15 - Инверсный вход датчика оборотов

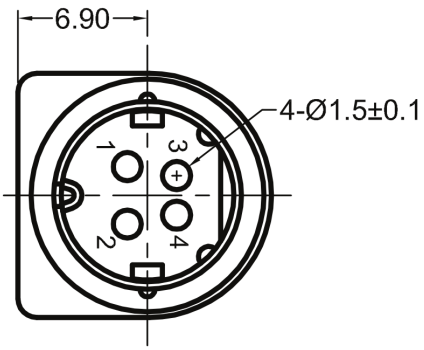
Замечание. BLDC моторы поддерживаются в прошивке версий 4.1.0 и старше.

Замечание. Для подключения DC мотора с рабочим током выше 3А нужно соединить выходы 1 и 3, а также выходы 2 и 4.

Предупреждение. Не рекомендуется присоединять к/отсоединять от контроллера двигатель пока есть питание на обмотках мотора.

Разъем силового питания для 1- и 2-осных систем

Одно- и двухосные системы, поставляемые в корпусе, используют разъём Kycon 4-pin DC power (part number по каталогу KPPX-4P, anacapa.kycon.com).





Назначение выводов:
1 - Силовое питание, "-".
2 - Силовое питание, "+". 12-48В.
3 - Силовое питание, "-".
4 - Силовое питание, "+". 12-48В.

ВАЖНО. Никогда не подавайте электропитание на контроллер и не подключайтесь к разъему силового электропитания, если вы не уверены, что параметры вашего блока питания соответствуют требуемым. Не пытайтесь подключить электропитание к контроллеру если не уверены, что разъемы на блоке питания и на корпусе контроллера совместимы друг с другом! Допустимые параметры подключения указаны в разделе техники безопасности.

ВАЖНО. Горячее присоединение/отсоединение, а также ненадежное подсоединение силового питания через разъём типа MF-4MRA может вывести из строя контроллер и/или компьютер. Подробнее смотрите в разделе техники безопасности.

Разъём управления 1- и 2-осной системами

Контроллеры в корпусе подключаются через разъём USB type-B.



Кабель USB-A - USB-B.



Разъем USB type B.

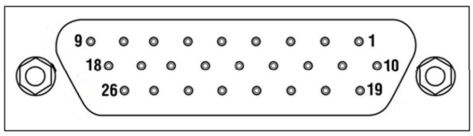
Таблица расположения выводов кабеля USB type-B

Номер вывода	Название	Цвет кабеля	Описание
1	VCC	Красный	+5 VDC
2	D-	Белый	Data -
3	D+	Зеленый	Data +
4	GND	Черный	Земля

Предупреждение. Используйте только проверенные и заведомо работоспособные USB-кабели! Неисправный или некачественный USB кабель может стать причиной неправильной работы контроллера, в том числе ошибок при вращении мотором или при опознавании устройства операционной системой. Для стабильного соединения используйте короткие кабели с толстыми проводниками и экранировкой.

Дополнительный разъем двухосных систем

На корпусе одноосной системы расположен порт HDB-26 типа "мама".



Нумерация выводов в разъеме HDB-26, вид спереди.

Назначение выводов:
1 - NC, не используется.
2 - NC, не используется.
3 - NC, не используется.
4 - GND, земля.
5 - NC, не используется.
6 - SYNC_IN_1, Вход синхронизации.
7 - SYNC_OUT_1, Выход синхронизации.
8 - RX_1, вход последовательного порта контроллера 1.
9 - NC, не используется.
10 - NC, не используется.
11 - NC, не используется.
12 - NC, не используется.
13 - DIR_1, сигнал Direction для управления внешним драйвером контроллера 1.
14 - GND, земля.
15 - +5В.
16 - BRAKE_1, выход управления тормозом контроллера 1.
17 - CLK_1, сигнал Clock для управления внешним драйвером контроллера 1.
18 - TX_1, выход последовательного порта контроллера 1.
19 - NC, не используется..
20 - NC, не используется.
21 - NC, не используется.
22 - GND, земля.
23 - NC, не используется.
24 - +5В.
25 - IO_1, порт I/O контроллера 1.
26 - POT_1, аналоговый вход общего назначения контроллера 1.

4.1.3. Двухосная система

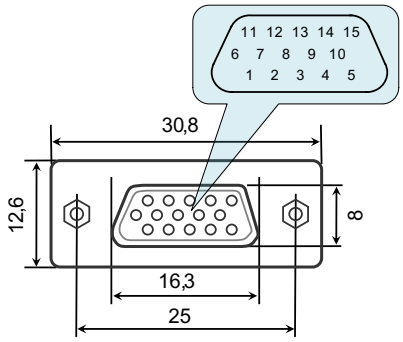
Корпус

Двухосная версия контроллера представляет собой две платы контроллера в металлическом корпусе. Размеры корпуса: 122 x 45 x 106 мм.
На передней панели расположен разъем силового питания, разъём для подключения к компьютеру тип USB-B , дополнительный разъем двухосных систем, разъём каскадирования контроллерных блоков, светодиод "питание". Разъём каскадирования контроллеров это выходной разъём USB типа A, который применяется для соединения нескольких двухосных конфигураций в цепочку, заканчивающуюся двухосным или одноосным блоком. Таким образом, требуемое количество осей может быть подсоединено к компьютеру посредством одного кабеля USB. Также на передней панели расположены светодиоды "стасус контроллера", "левый концевик", "правый концевик", кнопки для движения вправо и влево для каждого из двух контроллеров двухосной системы.
На задней панели расположены разъем подключения позиционера для каждого из двух контроллеров, а также разъем подключения джойстика и дополнительный разъём двухосных систем.

Разъёмы

Разъем подключения позиционера

Для подключения позиционера используется разъем DSub 15 типа "мама".



Размеры и нумерация выводов в разъеме DSub, вид спереди.

Назначение выводов:
1 - ШД фаза не B или - DC-мотора
2 - ШД фаза B или + DC-мотора или фаза B BLDC мотора
3 - ШД фаза не A или - DC-мотора или фаза C BLDC мотора
4 - ШД фаза A или + DC-мотора или фаза A BLDC мотора
5 - Выход 5В, 500 мА, стабилизированное для питания энкодера
6 - Однопроводной интерфейс опознавания подвижки (работает только с позиционерами Standa)
7 - Земля логическая для концевиков, энкодера и прочего
8 - Концевик №2
9 - Концевик №1
10 - Энкодер A
11 - Энкодер B
12 - Вход датчика оборотов
13 - Инверсный канал энкодера A
14 - Инверсный канал энкодера B
15 - Инверсный вход датчика оборотов

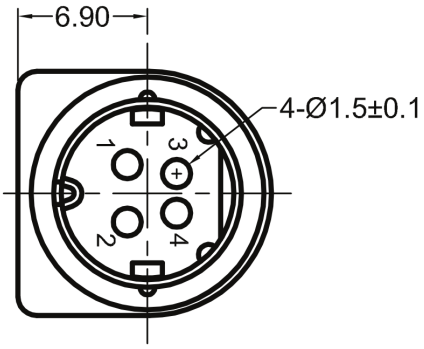
Замечание. BLDC моторы поддерживаются в прошивке версий 4.1.0 и старше.

Замечание. Для подключения DC мотора с рабочим током выше 3А нужно соединить выходы 1 и 3, а также выходы 2 и 4.

Предупреждение. Не рекомендуется присоединять к/отсоединять от контроллера двигатель пока есть питание на обмотках мотора.

Разъем силового питания для 1- и 2-осных систем

Одно- и двухосные системы, поставляемые в корпусе, используют разъём Kycon 4-pin DC power (part number по каталогу KPPX-4P, anacapa.kycon.com).





Назначение выводов:
1 - Силовое питание, "-".
2 - Силовое питание, "+". 12-48В.
3 - Силовое питание, "-".
4 - Силовое питание, "+". 12-48В.

ВАЖНО. Никогда не подавайте электропитание на контроллер и не подключайтесь к разъему силового электропитания, если вы не уверены, что параметры вашего блока питания соответствуют требуемым. Не пытайтесь подключить электропитание к контроллеру если не уверены, что разъемы на блоке питания и на корпусе контроллера совместимы друг с другом! Допустимые параметры подключения указаны в разделе техники безопасности.

ВАЖНО. Горячее присоединение/отсоединение, а также ненадежное подсоединение силового питания через разъём типа MF-4MRA может вывести из строя контроллер и/или компьютер. Подробнее смотрите в разделе техники безопасности.

Разъём управления 1- и 2-осной системами

Контроллеры в корпусе подключаются через разъём USB type-B.



Кабель USB-A - USB-B.



Разъем USB type B.

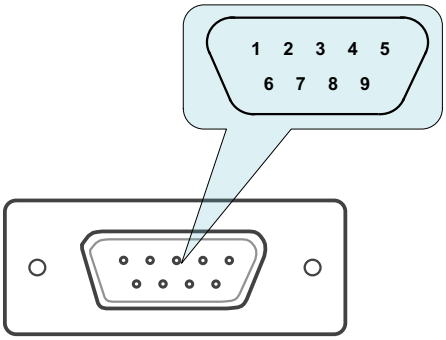
Таблица расположения выводов кабеля USB type-B

Номер вывода	Название	Цвет кабеля	Описание
1	VCC	Красный	+5 VDC
2	D-	Белый	Data -
3	D+	Зеленый	Data +
4	GND	Черный	Земля

Предупреждение. Используйте только проверенные и заведомо работоспособные USB-кабели! Неисправный или некачественный USB кабель может стать причиной неправильной работы контроллера, в том числе ошибок при вращении мотором или при опознавании устройства операционной системой. Для стабильного соединения используйте короткие кабели с толстыми проводниками и экранировкой.

Разъём подключения джойстика для 1- и 2-осной систем

Для подключения джойстика на корпусе одноосной или двухосной системы системы может быть установлен DSub 9 пин разъем типа "папа".



Размеры и нумерация выводов в разъеме DSub, вид спереди.

Назначение выводов:
1 - BUT_R_1, вход с внешней кнопки "Вправо" первой оси.
2 - Joy_2, аналоговый вход 0-3В, используется для подключения внешнего джойстика второй оси.
3 - GND, земля, общая для всех осей.
4 - BUT_R_2, вход с внешней кнопки "Вправо" второй оси.
5 - GND, земля, общая для всех осей.
6 - Joy_1, аналоговый вход 0-3В, используется для подключения внешнего джойстика первой оси.
7 - Выход 3.3В.
8 - BUT_L_1, вход с внешней кнопки "Влево" первой оси.
9 - BUT_L_2, вход с внешней кнопки "Влево" второй оси

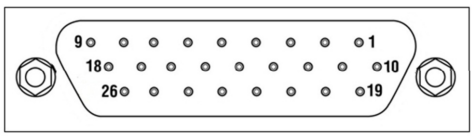
Замечание. Если разъем установлен на одноосной системе, то контакты 2, 4, 9 не используются.

Замечание. Неиспользуемые контакты внутреннего разъема не требуют никакого дополнительного подключения или подтяжки к земле/питанию. Просто не используйте их.

ВАЖНО. Аналоговые входы Joy, Pot рассчитаны на работу с напряжением ДО 3В. Никогда не подавайте на эти входы большее напряжение, в том числе 3.3В. Это может нарушить правильную работу всех аналоговых каналов контроллера и вывести из строя контроллер или двигатель.

Дополнительный разъем двухосных систем

На корпусе двухосной системы расположен порт HDB-26 типа "мама".



Нумерация выводов в разъеме HDB-26, вид спереди.

Назначение выводов:
1 - RX_2, вход последовательного порта контроллера 2.
2 - SYNC_IN_2, Вход синхронизации второй оси.
3 - TX_2, выход последовательного порта контроллера 2.
4 - GND, земля.
5 - SYNC_OUT_2, Выход синхронизации второй оси.
6 - SYNC_IN_1, Вход синхронизации оси 1.
7 - SYNC_OUT_1, Выход синхронизации оси 1.
8 - RX_1, вход последовательного порта контроллера 1.
9 - NC, не используется.
10 - CLK_2, сигнал Clock для управления внешним драйвером контроллера 2.
11 - POT_2, аналоговый вход общего назначения контроллера 2
12 - IO_2, порт I/O контроллера 2.
13 - DIR_1, сигнал Direction для управления внешним драйвером контроллера 1.
14 - GND, земля.
15 - +5В.
16 - BRAKE_1, выход управления тормозом контроллера 1.
17 - CLK_1, сигнал Clock для управления внешним драйвером контроллера 1.
18 - TX_1, выход последовательного порта контроллера 1.
19 - DIR_2, сигнал Direction для управления внешним драйвером контроллера 2.
20 - BRAKE_2, выход управления тормозом контроллера 2
21 - PBRK_2, Выход магнитного тормоза 2.
22 - GND, земля.
23 - PBRK_1, Выход магнитного тормоза 1.
24 - +5В.
25 - IO_1, порт I/O контроллера 1.
26 - POT_1, аналоговый вход общего назначения контроллера 1.

4.2. Кинематика и режимы движения

1. Движение с заданной скоростью
2. Движение в заданную точку
3. Смещение на заданное расстояние
4. Движение с ускорением
5. Компенсация люфта
6. Реверсирование движения
7. Рекомендации для точного движения
8. ПИД алгоритм для управления DC двигателем
9. Режимы остановки движения

4.2.1. Движение с заданной скоростью

Режим движения с заданной скоростью является основным режимом работы контроллера со всеми типами двигателей. Он позволяет поддерживать заданную скорость движения вдали от точки назначения, обычно применяется совместно с режимами Движение в заданную точку или Смещение на заданное расстояние. Этот режим может быть также вызван командами движения влево или вправо.

Скорость измеряется в шагах шагового двигателя или в отсчётах энкодера при его наличии (работает для всех типов двигателей) за единицу времени. Для DC-моторов постоянная скорость поддерживается даже при меняющейся внешней нагрузке.

Включение режима Движение с ускорением временно **дезактивирует** режим движения с заданной скоростью.

При получении команды для начала движения контроллер перемещает двигатель с заранее настроенной пользователем скоростью. **Скорость может быть задана** из соответствующего раздела меню "Settings..." программы XILab или с помощью вызова функции set_move_settings(), (см. раздел Руководство по программированию). **Значение скорости** при работе с шаговыми моторами можно задать в целых шагах и микрошагах в секунду, для DC-моторов скорость задается в оборотах в минуту (RPM).

Скорость специальных движений, таких как компенсация люфта или автоматическая калибровка нулевой позиции, отличается от общей скорости движения и настраивается отдельно.

Контроллер может ограничивать максимальную скорость, если соответствующая настройка сделана пользователем. В этом случае любое движение, которое бы происходило со скоростью выше максимальной, происходит с максимальной скоростью. Отдельно можно настроить контроллер, чтобы максимальная скорость использовалась для всех обычных движений, за исключением специальных, таких как компенсация люфта или автоматическая калибровка нулевой позиции. Настроить максимальную скорость и режимы её использования можно на вкладке меню "Settings..." программы XILab.

Текущую скорость вы можете увидеть в Главном окне программы XILab в поле Speed или воспользоваться графиками отображения основных параметров работы.

> *Замечание.* Если **стабильность поддержания скорости** при использовании энкодера кажется вам недостаточной, то обратитесь к рекомендациям для точного движения.

> *Замечание.* Максимально допустимая скорость, которую можно задать это **100000 шагов/сек.** или **100000 оборотов/сек.** в зависимости от типа двигателя.

4.2.2. Движение в заданную точку

Режим движения в заданную точку является основным режимом работы контроллера со всеми типами двигателей, обычно используется совместно с режимом движения с заданной скоростью. Он позволяет перемещать позиционер в заданное положение, причем координата точки назначения имеет **абсолютное значение**, в отличие от режима смещения на заданное расстояние. В режиме Компенсация люфта может производится дополнительное возвратно-поступательное движение вблизи заданной точки.

При использовании энкодера для определения текущей позиции, возможно несколько малозаметных "колебаний" прежде чем мотор остановится в заданной точке.

При получении команды для начала движения контроллер либо переходит в режим движения с ускорением, при включении соответствующей настройки, либо сразу поворачивает ось двигателя с заранее настроенной пользователем скоростью. Движение останавливается при достижении заданной точки с переходом в режим замедления, если соответствующая настройка включена. **Позиция назначения** задается в главном окне программы XiLab. **Позицию назначения** можно задать в целых шагах и микрошагах при работе с шаговыми моторами, или отсчетах энкодера для всех двигателей.

Текущую позицию вы можете увидеть в главном окне программы XILab в блоке Control или воспользоваться графиками отображения основных параметров работы.

> *Замечание.* Если **точность позиционирования** при использовании энкодера кажется вам недостаточной, то обратитесь к рекомендациям для точного движения.

4.2.3. Смещение на заданное расстояние

Режим смещение на заданное расстояние позволяет смещать позиционер на заданную величину относительно позиции 0 (если это первая команда движения после старта контроллера) или позиции, в которую пришел двигатель в результате выполнения предыдущих команд, т.е. координата точки назначения имеет **относительное значение**. Этот режим удобен в случае, когда абсолютная позиция не известна или не важна.

Режим смещение на заданное расстояние полностью аналогичен режиму Движение в заданную точку. Меняются только правила расчёта позиции назначения. Если движения в данный момент не происходит, то вызов команды смещение на заданное расстояние сдвигает мотор относительно текущей позиции. Если послать эту команду в процессе движения к позиции (*MOVE*, *MOVR*, *SSTP*, *STOP*, *LOFT*), то к смещение добавляется к позиции назначения и контроллер перенастраивается на новую позицию назначения в процессе движения. Если эта команда приходит в процессе движения по направлению, то смещение добавляется к предыдущей позиции назначения и движение автоматически перенастраивается на новую позицию.



ВАЖНО. Смещение происходит всегда относительно позиции в которую позиционер попал или должен был попасть в результате выполнения предыдущей команды, *MOVE*, *MOVR*, *SSTP*, *STOP*, *LOFT* или выполнения смещения по приходу предыдущего входного синхроимпульса, вне зависимости от того было это движение успешно выполнено или было прервано.

Замечание. Режим смещения на заданное расстояние может быть активирован как соответствующей командой, так и входным синхроимпульсом, подробнее см. раздел ТТЛ-синхронизация.

4.2.4. Движение с ускорением

Функция движения с ускорением активирована **по умолчанию**. Движение с ускорением используется для плавного начала и завершения движения без "толчков", обязательно сопровождающих мгновенный выход на заданную скорость. Кроме того, инерция ротора мотора и остальных компонентов позиционера обычно просто не позволяет мгновенно набрать высокую скорость, что приводит к потере шагов и срыву движения шагового двигателя в режиме работы без обратной связи. В режиме работы с обратной связью по энкодеру скорость будет набираться максимально быстро, как того позволяют ограничители движения. Быстрый набор скорости делает движение менее стабильным и создаёт больше шума и вибраций. Поэтому мы рекомендуем использовать движение с ускорением. Функция движения с ускорением позволяет достигать максимальных скоростей и стабильного движения даже на моторах со средним значением крутящего момента.

Режим движения с ускорением/замедлением работает следующим образом: при разгоне, когда требуемая скорость движения выше текущей по модулю, происходит постепенное ускорение движения на величину Acceleration, измеряемую в шагах на секунду в квадрате. При достижении требуемой скорости контроллер переходит в режим движения с заданной скоростью. При подходе к позиции назначения контроллер начинает снижать скорость движения так, чтобы замедление равнялось Deceleration и остановка произошла ровно в позиции назначения. Таким образом, этот режим обеспечивает трапециедальный профиль скорости. Если расстояние, на которое требуется сдвинуться мало, то ускорение может непосредственно смениться замедлением, что приведёт к треугольному профилю скорости. Включение/отключение режима движения с ускорением, а также настройку величины ускорения и замедления можно сделать в программе XiLab (см. раздел Настройка кинематики движения (Шаговый двигатель)) или командой set_move_settings(), описанной в Руководстве по программированию.

Ускорение Acceleration настраивается независимо от замедления Deceleration. Это сделано неспроста. Обычно максимальное возможное ускорения меньше чем максимальное замедление из-за трения, которое препятствует ускорению, но способствует замедлению. Поэтому для максимально быстрого отклика позиционера нужно либо воспользоваться готовыми профилями, либо экспериментально подобрать те значения ускорения и замедления, которые способен обеспечить Ваш позиционер. Для шаговых двигателей, работающих без обратной связи это те значения, при которых не происходит потери шагов. Для двигателей с обратной связью нужно проконтролировать трапециидальность скорости на графиках XiLab. Стоит брать значения ускорения/замедления в полтора-два раза ниже тех, где наблюдаются искажения профилей скорости или потери шагов.

> *Замечание.* Отключение ускорения/замедления может быть полезно для управления многоосными системами, где движение по многомерным траекториям требует постоянной проекции скорости на каждую из осей.

> *Замечание.* В Главном окне программы XILab не отображается значение ускорения.

> *Замечание.* Устанавливаемое ускорение/замедление должно быть рассчитано так, чтобы обеспечить выход на требуемую скорость или замедление с максимально возможной скрости не более чем за **5 мин.** Если установить ускорение/замедление не придерживаясь этого правила в настройках кинематики движения, то контроллер вернет ошибку связанную с попыткой задания недопустимых значений, а значение ускорения/замедления будет изменено в контроллере для попадания в допустимый диапазон.

4.2.5. Компенсация люфта

В любом механическом устройстве, например редукторе или червячной передаче, существует люфт, наличие которого приводит к тому, что при подходе к одной и той же позиции с разных сторон реальное положение позиционера будет отличаться, при том, что ось мотора находится точно в заданном положении.

Для устранения такой неоднозначности используется режим компенсации люфта, активация которого позволяет пользователю выбрать, с какой стороны нужно приближать позиционер к заданной точке. В дальнейшем при любых движениях позиционер будет подходить к точке останова только с выбранной стороны, устраняя механический люфт. Если естественное направление подхода к заданной точке не совпадает с выбранным направлением подхода, то контроллер заводит двигатель на некоторое расстояние, определяемое пользователем, за заданную точку, разворачивает двигатель и завершает подход к заданной точке с требуемой стороны.

При движении нагруженной механической системы в зоне люфта её динамические характеристики отличаются от обычного движения. Поэтому движение в зоне люфта выполняется с задаваемой пользователем скоростью.

Пользователь может настраивать следующие параметры системы компенсации люфта:

- Флаг включения/выключения компенации люфта.
- Скорость движения при выполнении компенсационного движения.
- Расстояние, на которое достаточно проехать, чтобы коомпенсировать люфт. Знак этой настройки определяет направление подхода. Положительный знак означает подход слева, а отрицательный - справа.

Контроллер сигнализирует о моментах, когда происходит отработка компенсации люфта в структуре состояния с помощью флага MOVE_STATE_ANTIPLAY. Он также выводится на главное окно XiLab.

Если нет уверенности, что текущее положение свободно от люфта, то можно совершить вынужденную компенсацию люфта с помощью команды *LOFT*. При выполнении этой команды в режиме остановки происходит сдвиг из текущей позиции на расстояние компенсации антилюфта и возвращение назад. Вызов данной команды во время движения приведет к плавной остановке двигателя. Применение этой команды имеет смысл только при включенной системе компенсации антилюфта.

> *Замечание.* Режим компенсации люфта не предусматривает никакой коррекции положения оси, он только позволяет пользователю выбрать, с какой стороны нужно приближать позиционер к заданной точке и всегда придерживаться указанного направления подхода.

Настройка компенсации люфта в программе XILab описана в настройке кинематики движения шагового двигателя. Команды включения и определения параметров компенсации люфта описаны в руководстве по программированию.

Люфт будет минимальным в том случае, если подход к заданной точке осуществляется с одинаковыми параметрами движения, поэтому оптимальными будут являться следующие значения параметров: скорость в зоне люфта должна быть равна номинальной, расстояние компенсации антилюфта должно быть таково, чтобы устройство успевало набрать номинальную скорость.
По умолчанию в профилях двигателей компенсация люфта задаётся по формуле:

$$S = \frac{U^2}{2}\left[\frac{1}{Ac} + \frac{1}{Dc}\right] + 0.2U,$$

где S - компенсация люфта, U - номинальная скорость, Ac, Dc - ускорение и замедление, 0.2 - время, в течение которого двигатель будет ехать с постоянной скоростью.

4.2.6. Реверсирование движения

Считается, что рост координаты соответствует движению вправо, а убывание координаты - движению влево. Если физически позиционер расположен так, что это правило не выполняется, либо на позиционере нанесен репер, направление возрастания которого не совпадает с ростом координаты, то необходимо реверсировать движение.

Реверс движения можно включить в меню XiLab в блоке Motor parameters. Включение этой настройки поменяет знак текущей координаты и понятия лево и право поменяются местами. Например, первое движение при калибровке нулевой позиции, будет выполняться теперь физически в другую сторону, команды влево и вправо на главном окне XiLab поменяются местами и т. п.



Предупреждение. Реверс относится к настройкам, изменение которых сказывается на всей работе контроллера. Работавшие ранее скрипты XiLab или собственные программы управления будут вести себя по иному.

В частности концевые выключатели настраиваются независимо от реверса движения. При включении или отключении реверса концевые выключатели обязательно нужно перенастроить.

4.2.7. Рекомендации для точного движения

Контроллер способен автоматически подстраиваться под необходимый режим, будь то поддержание скорости или координаты. Но скорость и качество подстройки зависят от настроек контроллера. Практически мгновенно выходить на требуемый режим способен шаговый двигатель в режиме позиционирования по шагам и микрошагам. Если шаговый двигатель физически неспособен обеспечить требуемую скорость или ускорение, то скорее всего движение вовсе остановится (сорвётся). При использовании датчика обратной связи, такого как квадратурный энкодер, в качестве опорного, движение шагового двигателя не сорвётся, но возможно, что контроллер не сможет обеспечить требуемые параметры движения.

DC-моторы требуют использования дополнительного датчика положения (энкодера). Непрямая связь между воздействием управляющей схемы и смещением положения позиционера с DC-мотором приводит к замедлению выхода на требуемую координату или на требуемую скорость. Ускорить этот процесс и сделать его более стабильным можно используя следующие рекомендации:

- в контроллер загружен и используется профиль соответствующий используемому позиционеру, если вы не уверены, что профиль верный, загрузите профиль из раздела Конфигурационные файлы;
- мотор не попадает в режим ограничения одного из рабочих параметров (ток или напряжение), см. подробнее разделы Ограничители на двигателях, Управление питанием мотора); Такое ограничение вы можете заметить по горизонтальной полоске над индикатором Current в блоке Power, Voltage или Speed в блоке Motor главного окна программы XILab в режиме управления одной осью, подробнее см. разделы Ограничители на двигателях, Управление питанием мотора).
- отсутствуют механические препятствия для движения, заклинивание оси или позиционера;
- мощность применяемого блока питания достаточна (см. Техника безопасности)

4.2.8. ПИД алгоритм для управления DC двигателем

Описание алгоритма

Управление DC двигателем осуществляется с помощью ПИД регулятора. Регулируемой величиной является координата. Для обеспечения возможности движения, сама регулируемая координата изменяется в соответствии с установленными настройками движения и поступившими командами. Изменяющуюся во времени регулируемую координату далее будем называть бегущей позицией. Управляющим сигналом регулятора является фактор заполнения ШИМ сигнала, подаваемого на обмотку двигателя.
Формула для вычисления управляющего воздействия:

$$U(t) = I + P + D = K_P \cdot E(t) + K_I \int E(t)dt + K_D \frac{dE(t)}{dt}$$, где:

$U(t)$ - управляющее воздействие

$E(t)$ - разница между бегущей координатой и текущей координатой двигателя

K_P, K_I, K_D - коэффициенты усиления пропорциональной, интегральной и дифференциальной составляющих регулятора, соответственно. Коэффициенты регулятора задаются с помощью соответствующего меню программы XiLab или с помощью вызова функции set_pid_settings(), (см. раздел Руководство по программированию).

Для того, чтобы результат работы ПИД регулятора не был зависим от двигателя, датчика обратной связи и текущего напряжения питания, производится нормировка полученного значения по следующей формуле:

$$DC(t) = \frac{U(t) \cdot U_{nom}}{U_{supp}(t) \cdot IPS}$$, где:

$DC(t)$ - фактор заполнения ШИМ

U_{nom} - номинальное(максимальное) напряжение питания двигателя, (см. раздел Ограничители на двигателях).

$U_{supp}(t)$ - текущее напряжение питания

IPS - разрешение датчика обратной связи в отсчетах/оборот

Данный подход позволяет менять двигатель, датчик обратной связи и источник питания без перенастройки ПИД регулятора.

Предупреждение. Если необходимо изменить настройки максимального напряжение питания мотора, не забудьте изменить коэффициенты ПИД регулятора в соответствии с приведенной выше формулой.

Особенности работы алгоритма

Коэффициенты ПИД регулятора

Для того, чтобы оптимальные коэффициенты ПИД регулятора не выходили из диапазона [0..65535], задаваемые пользователем значения нормируются.
Для лучшего понимания работы регулятора, рассмотрим какое влияние оказывают различные составляющие.
Будем считать, что напряжение питания $U_{supp}(t)$ постоянно и равно номинальному напряжению мотора U_{nom}. При выполнении данного предположения фактор заполнения ШИМ сигнала будет равен 1 в следующих случаях:

1. $K_P = 1, K_I = 0, K_D = 0$ - если целевая позиция превышает реальную позицию на 256 оборотов ротора
2. $K_P = 0, K_I = 1, K_D = 0$ - если интеграл, приведенный в формуле выше, равен 52,5 оборотов ·сек
3. $K_P = 0, K_I = 0, K_D = 1$ - если реальная скорость вращения ротора отличается правильной скорости на 96000 об/мин.

Попадание в целевую позицию

Попадание в целевую позицию считается успешным как только ось двигателя попадает в целевую позицию. При этом, возможно наличие некоторых переколебаний около целевой позиции. Если движение производится без ускорения, пришла команда немедленной остановки или происходит экстренная остановка по достижению концевого датчика, то мотору понадобится некоторое время до полной остановки и возвращения в правильную позицию.

Предупреждение. Если ПИД регулятор настроен неправильно, возможно возникновение длительных колебаний около целевой позиции, хотя движение и будет считаться завершённым.

Рекомендации по настройке ПИД регулятора

При настройке ПИД регулятора следует пользоваться тремя критериями качества его работы:

- Точность поддержания скорости - определяется средним отклонением скорости от требуемой. При этом, если во время движения скорость принимает не более 3 различных значений, можно считать, что режим поддержания скорости работает оптимальным образом. Более аккуратное поддержание требуемой скорости не представляется возможным из-за эффекта квантования значения скорости. Также следует минимизировать время выхода скорости на требуемое значение при выключенном ускорении/замедлении.
- Качество попадания в целевую позицию определяется четырьмя критериями:
  - Время до окончательной остановки в целевой позиции.
  - Отсутствие проскока целевой позиции при подходе к ней.
  - Отсутствие нескольких переколебаний около целевой позиции перед остановкой в ней.
  - Отсутствие самопроизвольных отклонений около целевой позиции после остановки в ней.
- Отсутствие шумов при работе. Шум увеличивается только от увеличения одного из трёх коэффициентов ПИД.

При настройке ПИД регулятора каждый может сам выбирать приоритет отдельных критериев качества в зависимости от решаемой задачи.

1. Приступая к настройке ПИД регулятора рекомендуется отключить все ограничения работы DC двигателя, в том числе ускорение.
2. Настройку регулятора лучше начинать с режима поддержания скорости.
3. Настройку ПИД регулятора следует начинать с нулевых значений интегрального и дифференциального коэффициентов.
4. Устанавливаем значение пропорционального коэффициента $K_P \leq 10$
5. Устанавливаем требуемую скорость движения и начинаем движение вправо или влево вдали от концевых выключателей.
6. Постепенно увеличиваем значение K_P и наблюдаем за графиком текущей скорости. Таким образом получаем оптимальное значение K_P при котором скорость движения поддерживается наилучшим образом и время выхода на неё перестаёт заметно уменьшаться с увеличением K_P. При этом следует обращать внимание на возможный рост шума.
7. После настройки значения K_P стоит приступить к настройке значения K_I. Интегральный коэффициент ПИД регулятора в большей степени влияет на процесс попадания в целевую позицию.
8. Устанавливаем значение интегрального коэффициента $K_I \leq 10$ и приступаем к движениям к позиции. Удобнее всего использовать команду смещения на заданное расстояние.
9. Лучше всего, чтобы при движении к позиции двигатель успевал разогнаться хотя бы до 20% от требуемой скорости.
10. Увеличение значения K_I приводит к более быстрой остановке в целевой позиции.
11. Постепенно увеличивая значение K_I добиваемся быстрого попадания в целевую позицию. Увеличение коэффициента следует прекратить, когда заметно снизится качество попадания в позицию (переколебания, осцилляции) или вырастут шумы. При этом, проскок позиции будет наблюдаться всё равно.
12. Для улучшения качества попадания в позицию понадобится изменение K_P в большую или меньшую сторону в зависимости от того где переколебаний меньше.
13. После подстройки K_P стоит проверить качество поддержания постоянной скорости и уровень шумов. При неудовлетворительных показателях качества коэффициенты подстраиваются в направлении предыдущих значений.
14. После настройки коэффициентов K_I и K_P можно приступить к настройке коэффициента K_D.
15. Настройку коэффициента K_D лучше начинать в режиме поддержания скорости, проверяя затем остальные параметры качества.
16. Коэффициент K_D увеличивается пока заметно уменьшение колебаний скорости около требуемого значения.
17. В случае наличия проскока позиции коэффициент K_D увеличивают. Однако дальнейшее увеличение приводит к колебанием около целевой позиции. Необходимо соблюсти баланс между скоростью попадания и отсутствием колебаний.

Если планируется работа с включенным ускорением, то его необходимо включить. При этом может возникнуть проскакивание целевой позиции. Для компенсации этого эффекта необходимо увеличение значения K_I.

На этом этапе первичная настройка ПИД контура завершена. Полученные коэффициенты в большинстве случаев годятся для работы. Для дальнейшей оптимизации коэффициентов они вариируются с постоянным контролем качества по выбранным критериям скорости, позиции и шума с учетом их важности в конкретной задаче.

> *Замечание.* Не рекомендуется одновременно изменять более одного коэффициента при настройке ПИД регулятора.

ВАЖНО. Крайне не рекомендуется устанавливать сразу большие значения коэффициентов ПИД регулятора или резко их изменять. Это может привести к самовозбуждению колебаний скорости на вашем двигателе и привести к поломке.

4.2.9. Режимы остановки движения

В контроллере существует два режима прекращения движения:

- экстренная остановка;
- остановка с замедлением.

Экстренная остановка

Экстренная остановка движения происходит по команде *STOP*. Контроллер старается мгновенно остановить вращение вала двигателя. Это может привести к пропуску шагов в шаговом двигателе, если не используется обратная связь. Резкое прекращение движения может пагубно влиять на оборудование, например, может произойти сдвиг образцов на предметных столиках микроскопов, потребоваться дополнительная юстировка оптической линии после резкой остановки.



Предупреждение. Когда контроллер настроен на остановку по срабатыванию левого/правого концевого выключателя, то всегда происходит экстренная остановка при достижении концевого выключателя. Этого надо избегать.

Остановка с замедлением

Остановка движения с замедлением выполняется по команде *SSTP*. В этом режиме происходит плавная остановка с замедлением Deceleration, если оно не отключено в настройках движения с ускорением.



Предупреждение. Если движение с ускорением отключено, то разницы между режимами остановки с замедлением и экстренной остановки не будет. По команде *SSTP* будет происходить резкая остановка.

4.2.10. ПИД алгоритм для управления BLDC двигателем

Описание алгоритма

Управление BLDC двигателем осуществляется с помощью ПИД регулятора. Регулируемой величиной является координата. Для обеспечения возможности движения, сама регулируемая координата изменяется в соответствии с установленными настройками движения и поступившими командами. Изменяющуюся во времени регулируемую координату далее будем называть бегущей позицией. Управляющим сигналом регулятора является модуль вектора тока, который (вектор) удерживается перпендикулярно ротору.

Формула для вычисления управляющего воздействия:

$U(t) = I + P + D = K_p \cdot E(t) + K_i \int E(t)dt + K_d \frac{dE(t)}{dt}$, где:

$U(t)$ - управляющее воздействие

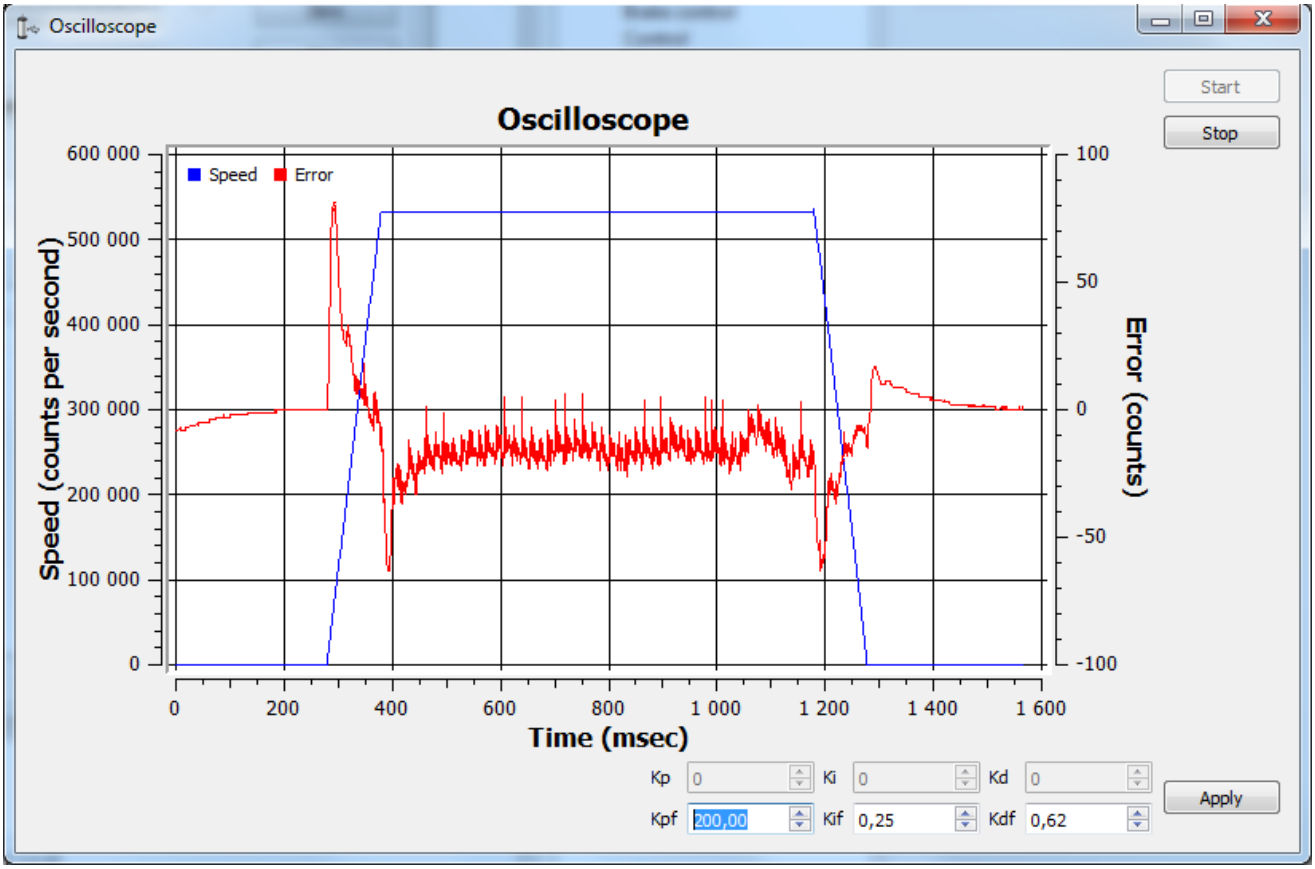
$E(t)$ - разница между бегущей координатой и текущей координатой двигателя

K_p, K_i, K_d - коэффициенты усиления пропорциональной, интегральной и дифференциальной составляющих регулятора, соответственно. Коэффициенты регулятора задаются с помощью соответствующего меню программы XiLab или с помощью вызова функции set_pid_settings(), (см. раздел Руководство по программированию).

ПИД-коэффициенты в алгоритме управления BLDC имеют тот же смысл, что и в управлении DC. См. влияние различных составляющих ПИД, рекомендации по настройке и замечания в главе ПИД алгоритм для управления DC двигателем

Ручная настройка коэффициентов ПИД-регулятора

Для тонкой настройки коэффициентов ПИД-регулятора существует специальное окно программы XiLab. В окне выводится зависимость от времени скорости BLDC-двигателя и ошибки следования соответсвующей координате, вид окна показан на скриншоте ниже.



Окно настроек коэффициентов ПИД-регулятора

Для корректной работы двигателя нужно добиться устойчивой компенсации ошибки следования.

Шаги по настройке коэффициентов:

1. Для начала нужно оценить ПИД-коэффициенты. Учитывая структуру управляемой системы, их можно вычислить по упрощенным формулам. Для этого используются параметры из документации на соответствующий двигатель и позиционер.
    - K_m - электромеханический коэффициент двигателя [Н / А] (момент создаваемый силой тока 1 А). Может быть

вычислен как отношение $K_m = \frac{F_n}{I_n}$, где F_n - номинальная (максимальное) усилие создаваемое двигателем, I_n - номинальная (максимальная) сила тока.
    - M - масса нагрузки (кг).
    - $\sigma = \frac{M}{K_m}$.
    - $K_p = 11500\sigma \cdot 1000$, $K_d = 186\sigma \cdot 1000$, $K_i = 12.2\sigma \cdot 1000$.
2. Выставить коэффициенты, рассчитанные по формулам, нажать Apply. В главном окне XiLab нажать кнопку Zero. В поле `Move to` выставить 0, отправить команду `Move to`. Двигатель должен остановиться. Попробовать сдвинуть позицию руками, убедиться, что отклик правильный - двигатель старается вернуться в нулевую позицию (реверс энкодера настроен верно).
3. В настройках движения выставить маленькую скорость, нажать Apply. В главном окне начать движение в сторону. Если начинаются вибрации и срывы, нужно увеличивать дифференциальный коэффициент (Kdf), увеличивать, пока заметно уменьшение колебаний скорости около требуемого значения.
4. Если вибрации имеют звуковые частоты (позиционер издаёт громкий звук при движении), возможно, следует уменьшить Kd коэффициент или все коэффициенты пропорционально
5. Интегральный коэффициент (Kif) отвечает за попадание в целевую позицию, для проверки удобно использовать команду `Shift on`.
6. Для более точной настройки коэффициентов используйте окно Oscilloscope, в этом окне визуализируется ошибка следования для данных параметров движения. Чтобы открыть окно нужно нажать кнопку PID tuning.

> *Замечание.* Функция более точной настройки коэффициентов PID доступна только в XiLab специальной версии. Для получения этого XiLab обратитесь в техническую поддежку.

7. После того, как коэффициенты настроены, можно их пропорционально менять, это соответствует увеличению/уменьшению массы, отклик на воздействие становится более/менее мощным. Добиться того, чтобы резкие остановки не приводили к срывам движения.

4.3. Основные возможности контроллера

1. Поддерживаемые типы двигателей
2. Ограничители на двигателях
3. Концевые выключатели
4. Автокалибровка "домашней" позиции
5. Работа с энкодерами
6. Датчик оборотов
7. Управление питанием мотора
8. Критические параметры
9. Хранение параметров во flash-памяти контроллера
10. Пользовательские единицы координат

4.3.1. Поддерживаемые типы двигателей

В настоящий момент контроллер поддерживает шаговые и DC двигатели. Характеристики поддерживаемых двигателей можно посмотреть в разделе Технические характеристики.

Шаговые двигатели

Основным параметром шагового двигателя является его номинальный ток. Номинальный ток двигателя можно установить во вкладке Настройка кинематики движения (Шаговый двигатель) .

ВАЖНО. Установка завышенного тока постепенно приведёт к перегреву двигателя и его физической поломке. Обязательно проконтролируйте, чтобы был установлен номинальный ток, соответствующий используемой подвижке. В предустановленных профилях подвижек все настройки уже сделаны правильно.

Другим важным параметром является режим деления шага. Пользователю доступны следующие режимы деления шага:

- 1 (полный) шаг
- 1/2 шага
- 1/4 шага
- 1/8 шага
- 1/16 шага
- 1/32 шага
- 1/64 шага
- 1/128 шага
- 1/256 шага

Режим микрошага устанавливается во вкладке Настройка кинематики движения (Шаговый двигатель) или командами настройки мотора, см. раздел Описание протокола обмена и описание соответствующих функций в разделе Руководство по программированию.

Замечание. Контроллер всегда использует внутреннее деление шага 1/256. При смене пользователем деления шага на более грубый, в ПО отображаются только кратные более грубому делению позиции, установка и передача становится возможна только в таких, более грубых делениях. Это сделано для поддержки устаревшего и совместимости с уже существующим ПО, работающим на делениях шага малой кратности. С другой стороны, работа на наибольшем делении шага позволяет двигаться наиболее плавно и тихо на малых скоростях.

Еще один непосредственным параметром шагового двигателя является количество полных шагов на оборот. Эта настройка не влияет на движение, но используется в блоке контроля проскальзывания или при работе с обратной связью по энкодеру.

Замечание. Контроллер поддерживает шаговые двигатели с датчиком обратной связи - энкодером. Энкодер может использоваться как основной датчик положения (подробнее) или для обнаружения проскальзывания, люфта или потери шагов (подробнее). Использование энкодера способствует стабильному прохождению резонансных скоростей без срыва движения.

DC двигатели

В отличии от шагового двигателя, для управления DC двигателем контроллеру необходимо наличие обратной связи позиции двигателя. В настоящее время, в качестве датчика обратной связи поддерживается только энкодер.
Основными параметрами двигателя являются максимальный ток и напряжение, которые устанавливаются во вкладке Настройка кинематики движения (DC мотор). Основным параметром энкодера является количество отсчетов на оборот.

ВАЖНО. Установка завышенного тока постепенно приведёт к перегреву двигателя и его физической поломке. Обязательно проконтролируйте, чтобы был установлен максимальный ток, соответствующий используемой подвижке. В предустановленных профилях подвижек все настройки уже сделаны правильно.

ВАЖНО. Установка Неправильного значения количества отсчетов энкодера на оборот приведет к тому, что установленное значение скорости не будет выдерживаться правильно. В некоторых случаях это может привести к поломке подвижки или редуктора.

Управление DC двигателем производится с помощью ПИД регулятора. Перед началом работы рекомендуется внимательно ознакомиться с разделом ПИД алгоритм для управления DC двигателем.

ВАЖНО. Неправильные настройки ПИД регулятора могут привести к поломке подвижки. В предустановленных профилях подвижек все настройки уже сделаны правильно. Без крайней необходимости не рекомендуется самостоятельно изменять данные настройки.

BLDC двигатели

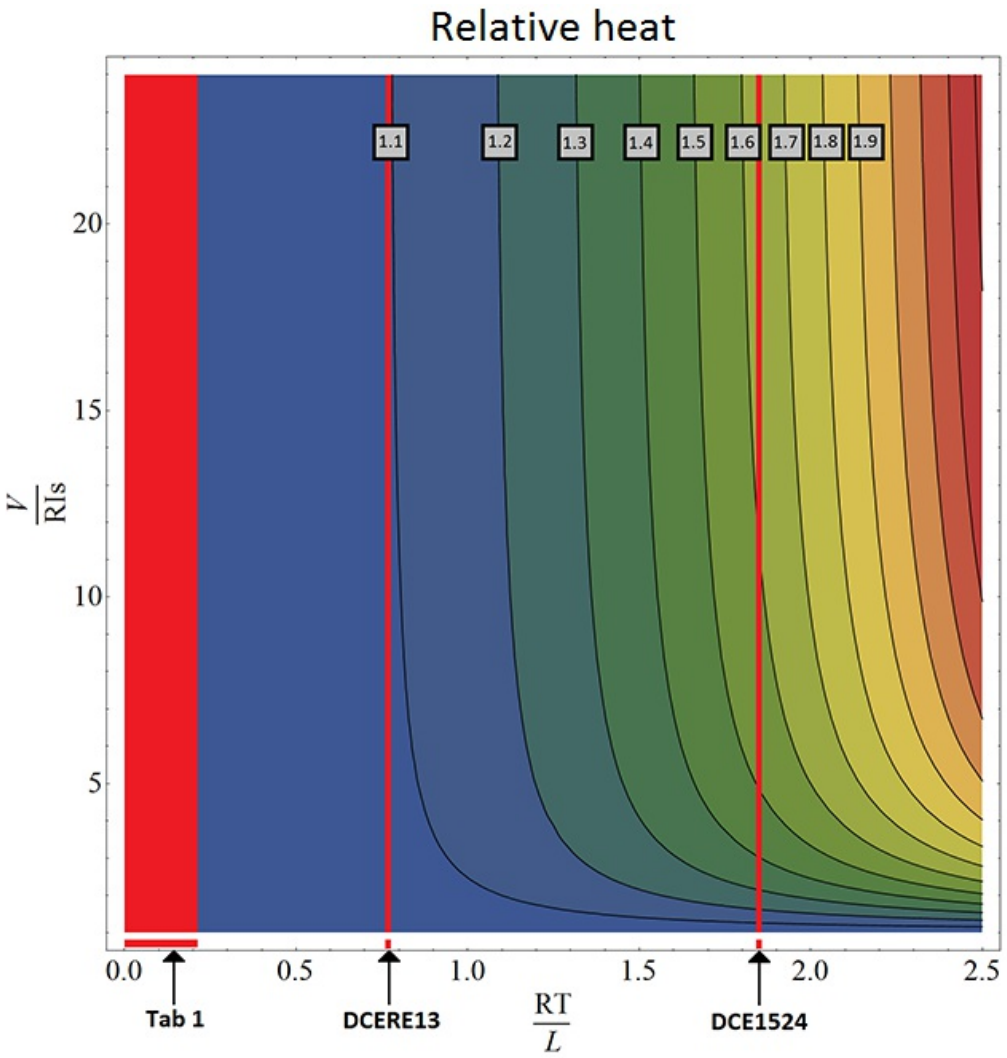
Прошивки версии 4.1.x и старше поддерживают управление BLDC. Как и для DC, для управления BLDC необходим датчик обратной связи; в настоящее время в качестве датчика обратной связи поддерживается только энкодер. Основными параметрами двигателя являются максимальный ток и количество полюсов, которые (параметры) устанавливаются во вкладке Настройка кинематики движения (BLDC мотор). Основным параметром энкодера является количество отсчетов на оборот.

Управление BLDC двигателем производится с помощью ПИД регулятора. Перед началом работы рекомендуется внимательно ознакомиться с разделом ПИД алгоритм для управления BLDC двигателем.

ВАЖНО. Как и для DC двигателя, важно установить правильное значение максимального тока, количества импульсов на оборот, ПИД-коэффициентов. Также важно установить правильное значение количества полюсов. Неправильные значения могут привести к поломке.

Критерий выбора двигателя

Для управление током в обмотках двигателях используется принцип широтно-импульсной модуляции, приводящий к колебаниям тока на частоте модуляции (так называемый "токовый риппл"). В зависимости от параметров используемого двигателя (индуктивность его обмоток, омическое сопротивление) риппл может быть разным. Такие резкие колебания тока могут приводить к тому, что мотор нагреется сильнее, чем ожидается при номинальном токе, т.е. $\frac{P_{real}}{RI_s^2}>1$, где RI_s^2 - мощность, которая ожидалась бы при прохождении постоянного тока I_s, P_{real} - действительная мощность, выделяемая в двигателе. Чтобы оценить перегрев, рекомендуем воспользоваться следующим графиком:



Tab 1

Мотор	RT/L
20	0.19576
28	0.07253
28s	0.07168
4118L1804R	0.02715
4118S1404R	0.02844
4247	0.0273
D42.3	0.0223
5618	0.0146
5618R	0.0146
5918	0.0116
5918B	0.012
VSS42	0.029
VSS43	0.0256
ZSS	0.04248
DCERE25	0.2106

Последовательность действий:

- Расчёт параметра $\frac{RT}{L}$, где R,L - сопротивление и индуктивность обмотки (см. документацию на соответствующий двигатель), T - время периода модуляции. Его следует взять равным 51.2 мкс для шаговых двигателей и 25.6 мкс для двигателей постоянного тока.

- Расчёт параметра $\frac{V}{RI_s}$, характеризующего превышение питающего напряжения над номинальным. Тут V - напряжение питания, R - сопротивление обмотки, I_s - ток стабилизации.
- Определение перегрева. Полученная точка попадёт в одну из областей на графике. Эти области соответствуют линиям равного уровня с шагом 0.1, каждая из которых показывает величину перегрева. Например, рассмотрим линию, у которой параметр перегрева равен 1.1. Слева от неё расположена область, где перегрев ниже 1.1, справа - выше 1.1. Между линиями 1.1 и 1.2 перегрев соответственно больше 1.1 и меньше 1.2 и т.д.

Пример расчёта для двигателя DCE1524 (двигатель постоянного тока):

T = 25.6 мкс R = 5.1 Ом L = 70 мкГн

$\frac{RT}{L}$ = 1.86

Далее проводим вертикальную линию, по которой можно сделать вывод какой будет перегрев при разных питающих напряжениях (эта линия проведена на графике выше). Допустим, что желаемый ток стабилизации I_s = 500 мА. Номинальное напряжение в таком случае: $R*I_s$ = 2.55 В. Тогда при превышении этого напряжения больше чем в 5, но меньше чем в 10 раз, перегрев будет между 1.5 и 1.6. А при питающем напряжении в 30 В двигатель нагреется в $\simeq$1.65 раз сильнее чем ожидается.

Для облегчения работы, все основные двигатели и их параметры были рассчитаны (см. Tab1) и нанесены на график.

4.3.2. Ограничители на двигателях

Для обеспечения безопасной работы двигателей предусмотрены ограничители по току и напряжению на обмотках двигателя, а так же ограничение максимальных оборотов оси мотора. Данные ограничения, если они активированы приводят к плавному снижению мощности и скорости вращения до значений, приводящих к снижению ограничиваемых параметров до установленных значений. Данная функция оперирует со значениями токов и напряжений непосредственно на моторе, в отличие от критических параметров, которые оперируют с токами и напряжениями на входе контроллера. Другим отличием "ограничителей" от критических параметров является то, что первые не приводят к остановке мотора и переходу в состояние **Alarm**, а лишь ограничивают рост тока, напряжения или оборотов двигателя.

Для DC моторов:

- Max voltage - номинальное напряжение питания мотора. Определяет максимальное напряжение питания на обмотках двигателя. Обычно используется для ограничения роста напряжения при заклинивании или нештатной работе подвижки. Стоит использовать только в случае, если не известно значение максимального тока на обмотках двигателя. Данный параметр используется при работе ПИД регулятора.
- Max current - определяет максимальное значение тока на обмотках двигателя. Обычно используется для ограничения роста тока при заклинивании или нештатной работе подвижки. Данное ограничение стоит выбирать исходя из того, какой ток может в течении длительного времени течь через обмотку, не вызывая повреждения двигателя (в первую очередь от перегрева).
- Max RPM - максимальная скорость вращения вала мотора. Обычно используется для ограничения скорости вращения при работе с редукторами и прочими механизмами, обладающими жесткими ограничениями на максимальную скорость вращения.

> Замечание. Не стоит путать понятия максимального тока двигателя и номинального тока. В общем случае, они могут отличаться в зависимости от охлаждения мотора и условий его эксплуатации. Также не стоит путать максимальный ток и стартовый ток, который развивается при неподвижном вале.

ВАЖНО. Изменение максимального напряжения питания мотора может привести к расстройке ПИД регулятора. Подробнее смотрите в разделе ПИД алгоритм для управления DC двигателем.

ВАЖНО. Значение максимального напряжения питания мотора может превышать значение номинального напряжения питания (обычно на 10-15%). Если вы используете двигатель с малой нагрузкой и вам нужна высокая скорость движения двигателя, то можно повысить максимальное напряжение питания мотора.

Работа ограничителя тока.
Важно помнить, что ограничитель максимального тока Max current при работе с DC двигателями работает не мгновенно. При возникновении превышения тока в обмотке двигателя, напряжение, подаваемое на двигатель, начинает постепенно уменьшаться до тех пор, пока ток через обмотку не будет меньше Max current. В случае, если во время быстрого движения произошло резкое заклинивание двигателя (самый худший случай) напряжение на обмотке двигателя может спадать в течение максимум 370мс. При правильно выбранном ограничении тока, за это время двигатель не перегреется.

> Замечание. Если поставить значение максимального тока Max current слишком маленьким, то возможно, что при большой нагрузке или высоком трении DC двигатель не сможет сдвинуться с места.

Для ШД:

- Max(nominal) Speed - максимальная скорость вращения вала мотора в шагах в секунду. Текущая скорость ШД определяется параметром Speed (см. Движение с заданной скоростью)
- Nominal current - определяет номинальное значение тока на обмотках двигателя. Это значение не превышается в силу особенностей управления шаговыми двигателями.

В программе XILab установки ограничителей описаны в разделах Настройка кинематики движения (DC мотор) и в Настройка кинематики движения (Шаговый двигатель).

4.3.3. Концевые выключатели

Задача концевых выключателей

Задача концевых выключателей - предотвратить выход позиционера за допустимые физические границы его перемещения или ограничить диапазон перемещения в соответствии с требованиями пользователя. Неправильная настройка концевых выключателей может привести к заклиниванию позиционера, если контроллер выйдет за границы допустимого диапазона.

Общие настройки

Если концевик считается активным, то в структуре состояния выставляется соответствующий флаг, а на главном окне XiLab выводится соответствующий значок (левый или правый). Контроллер способен останавливать любое движение в сторону обоих активных концевых выключателей (левого и правого), только одного (левого или правого) или не ограничивать движение. Настройка концевиков может быть выполнена в XILab (см. Настройка диапазона движения и концевых выключателей).

Программное ограничение диапазона движения

Если аппаратных ограничителей на диапазон движения нет, а позиционер требует такого ограничения, то можно использовать программные ограничители. Для этого концевики переводятся в режим ограничения по отсчётам позиции (см. Настройка диапазона движения и концевых выключателей). Используются поля левой границы и правой границы (значение правой границы должно быть больше левой). В этом режиме левый концевик считается активным если текущая позиция меньше левой границы, а правый - если текущая позиция больше правой границы движения. Срабатывание происходит за время около одной миллисекунды.

Предупреждение. Программное ограничение диапазона работает надежно только, если не происходит непосредственного задания новой позиции командами ZERO или SPOS, нет потери шагов или неисправности энкодера, при его использовании для позиционирования, а также не происходит частой потери питания во время движения. Если возникла одна из таких проблем, то программный диапазон надо перенастроить. Автоматически это можно сделать если есть подходящий опорный датчик с помощью автоматической калибровки нулевой позиции.

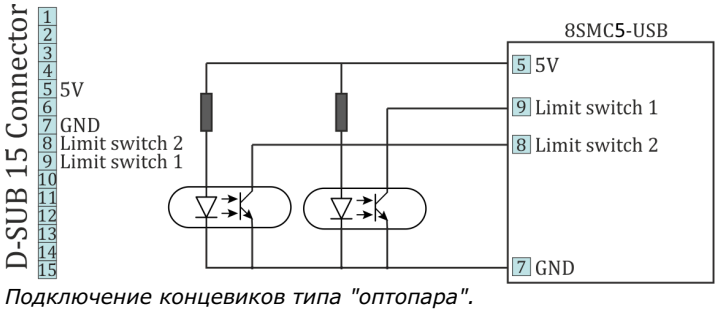
Аппаратные концевые выключатели

Контроллер может работать с концевыми выключателями на базе сухих контактов, оптопар, герконов и датчиками любых других типов, способных выдавать электрический сигнал "логическая единица" стандарта ТТЛ 5В в одном состоянии и "логический нуль" в другом. Причем каждый концевик может быть сконфигурирован независимо. Также есть возможность программно менять местами концевые выключатели и изменять их полярность.

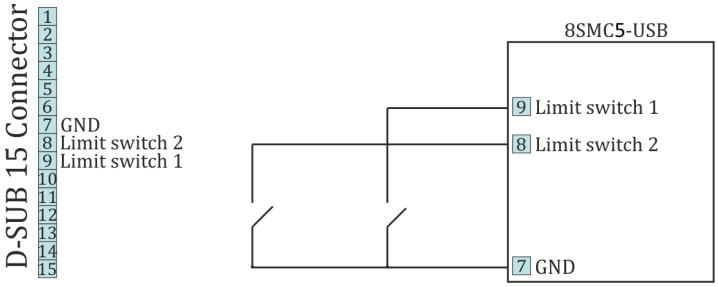
Замечание. Концевые выключатели также удобно использовать для автоматической калибровки нулевой позиции.

Подключение концевых выключателей

Концевые выключатели подключаются к выводам в разъеме DSub (см. разъём подключения позиционера), который есть во всех системах: плата контроллера, одноосная и двухосная в корпусе. Ниже приведены типовые схемы подключения:



Подключение концевиков типа "оптопара".



Подключение концевиков типа "сухой контакт".

Расположение концевых выключателей на трансляторах

Контроллеру необходимо указать какой из концевых выключателей будет левым, а какой - правым. Иногда это заранее неизвестно, а известно лишь, что оба концевика подключены и срабатывают каждый по достижении своей границы перемещения. Если неправильно настроить концевики, то позиционер может заклинить. Поэтому контроллер поддерживает простую функцию обнаружения неверно настроенных концевиков, останавливаясь по обоим из них. Убедитесь, что оба концевика не активны, их полярность настроена правильно и включена остановка по обоим концевикам. Включите флаг обнаружения неправильного подключения концевиков в соответствующем меню XiLab. Начните движение в любую сторону до остановки движения по концевику. Если движение было вправо, а левый концевик стал активным, или наоборот, то нужно поменять концевики местами (см. Настройка диапазона движения и концевых выключателей). При обнаружении неверного срабатывания концевика контроллер может перейти в режим Alarm, если включена соответствующая настройка в меню критических параметров.

Предупреждение. Защита от перепутанных концевиков не гарантирует, что о проблеме перепутанности можно забыть. Она лишь облегчает первоначальную настройку. Нельзя, в частности, начинать движение если какой-либо концевик активен, даже при включенной функции защиты.

4.3.4. Автокалибровка "домашней" позиции

Используется для определения и установки подвижки в начальную позицию. Данная функция контроллера предназначена для простого и не требующего программирования со стороны пользователя поиска "домашней" или "нулевой" позиции, положение которой может быть задано относительно одного или двух внешних датчиков и/или внешнего сигнала.
Автокалибровка может быть **настроена** пользователем в программе XILab на вкладке *Device configuration->Home position* см. раздел Настройка исходного положения, а запущена кнопкой в главном окне XiLab.

При активации этой функции контроллер вращает мотор в заданную сторону с обычной скоростью для быстрого поиска положения остановки. После нахождения граничного положения контроллер отводит мотор назад на половину оборота и начинает вновь движение в заданном направлении, но с другой настроенной скоростью. Скорость калибровки задаётся обычно более низкой, чем скорость обычного движения, для повышения точности калибровки. Окончание движения определяется одним из трех способов в зависимости от выбора пользователя:

- **движение до концевика**, то используются текущие настройки концевиков (расположение, полярность), см. раздел Настройка диапазона движения и концевых выключателей .
- **движение до поступления сигнала на вход синхронизации**, то используются текущие настройки входа синхронизации, см. раздел Настройки синхронизации, если вход синхронизации программно выключен в соответствующих настройках, то обработка сигнала со входа синхронизации никогда не произойдет.
- **движение до поступления сигнала с датчика оборотов**, то используются текущие настройки датчика оборотов, см. раздел Контроль позиции.

После успешного завершения калибровки домашней позиции в структуре состояния контроллера устанавливается флаг STATE_IS_HOMED. Если после этого позиция каким-либо образом сбилась (остановка по концевому выключателю, экстренная остановка во время движения, обнаружение потери шагов, переход в режим Alarm), то соответствующий флаг структуры состояния снимается и нужно вновь провести калибровку "домашней" позиции.

> *Замечание.* Если команда экстренной остановки движения или команда отключения питания мотора выполняются в момент, когда мотор не вращается, то это не сбивает калибровку "домашней" позиции и флаг STATE_IS_HOMED не снимается.

> *Замечание.* Если по концевому выключателю должна произойти остановка движения, то это не сбивает программу калибровки.

Точная докалибровка

После первого движения позиция контроллера уже определена, но прежде чем сделать сдвиг до домашней позиции возможно включение дополнительного движения. Это позволяет достичь точной калибровки домашней позиции с точностью достигающей на некоторых позиционерах 1/256 шага для шаговых двигателей или 1 отсчета энкодера для DC двигателя. Если установлен соответствующий флаг, контроллер вращает мотор в заданную пользователем сторону с настроенной скоростью до достижения сигнала со входа синхронизации, сигнала с датчика оборотов или до достижения концевика, в соответствии с выбором пользователя. При этом разумно использовать сигнал с датчика оборота на валу двигателя до редуктора и совершать движение на маленькой скорости. Это обеспечит максимальную точность. Так как сигналы окончания первого движения и второго движения могут совпадать, то предусмотрен флаг начала отслеживания сигнала окончания второго движения только после совершения полуоборота вала двигателя. Это позволяет избежать неоднозначной последовательности получения сигналов завершения первого и второго движений. В результате опционального второго движения калиброванная позиция уточняется.

> *Замечание.* Если используется вторая фаза движения, то первое движение можно обычно выполнять с высокой скоростью, так как оно лишь грубо калибрует позицию и точность там не требуется. Не произойдёт повышения точности если использовать для второй фазы движения второй концевик, так как его физические параметры такие же, как и у первого концевика.

Финальным действием контроллер двигает мотор в заданную сторону с настроенной скоростью на заданное расстояние Standoff. Это нужно для попадания в "домашнюю" позицию.

> *Замечание.* Получаемая в результате калибровки позиция будет немного зависеть от скорости, с которой выполнялось последнее движение до срабатывания выбранного датчика. Поэтому для точного попадания в ту же позицию не меняйте параметры скорости.

Описание функций автокалибровки домашней позиции приведено в руководстве по программированию.
Команды настройки параметров автокалибровки описаны в разделе Описание протокола обмена
Команда калибровки позиции приведена в разделе Описание протокола обмена.

Для автоматической установки настроек "домашней позиции" в комплект XILab входит скрипт set_zero. Этот скрипт изменяет настройку Standoff вкладки Home position так, что текущая позиция становится "домашней".
Использование скрипта:

- установите подвижку в желаемую позицию
- включите скрипт и дождитесь окончания его выполнения

В результате подвижка окажется в той же позиции, и все последующие вызовы функции homing будут приводить её туда. Не забудьте сохранить настройки в энергонезависимую память контроллера.

4.3.5. Работа с энкодерами

Область применения энкодеров

Энкодеры применяются для создания точной и быстродействующей обратной связи по координате со всеми типами электродвигателей. Причем обратная связь может осуществляться по положению оси мотора, по линейному положению позиционера, углу поворота моторизованного столика или по любому параметру, непосредственно связанному с положением оси мотора и измеряемому с помощью двухканального квадратурного энкодера, удовлетворяющего требованиям описанным в разделе Технические характеристики для соответствующего типа контроллера. Контроллер **8SMC5** поддерживает как дифференциальные энкодеры, так и простые (single-ended) энкодеры.

Что такое квадратурный энкодер?

Энкодер - это датчик механического движения. Квадратурный энкодер предназначен для прямого определения позиции оси. Датчик передает относительное положение оси в виде двух электрических сигналов по каналам CH A и CH B, смещенных относительно друг друга на четверть периода.



Сигналы на выходе CH A и CH B квадратурного энкодера.

Механика оптического квадратурного энкодера.

Механика оптического квадратурного энкодера представлена на рисунке. Используются две оптопары. Принцип работы оптопары: светодиод и детектор расположены напротив друг друга с разных сторон от диска. Когда "окно" диска попадает на детектор, оптопара «открыта» (выходной сигнал - логический 0). Если детектор закрыт непрозрачной частью диска, то выходной сигнал датчика – логическая 1.

Основная характеристика квадратурного энкодера – число шагов на один оборот (CPR). Стандартные значения разрешения для энкодера – от 24 до 1024 CPR. Каждый период изменения сигнала может быть расшифрован 1,2 или 4 кодами, что соответствует режимам работы X1, X2 и X4. В данном контроллере используется наиболее точный режим X4. Максимальная частота каждого из сигналов энкодера, зависит от выбранного энкодера, так для 200 кГц и режима x4 контроллер способен воспринимать 800 000 отсчётов положения по энкодеру в секунду.

Возможности контроллера

Контроллер имеет два режима работы с энкодером:

- использование энкодера как основного датчика положения (основной режим работы с двигателями постоянного тока, опционален для шаговых двигателей).
- обнаружение проскальзывания, люфта или потери шагов (рекомендуемый режим работы совместно с шаговыми двигателями, если энкодер не используется как основной датчик положения, подробнее).

Режим ведущего энкодера

В этом случае все параметры мотора, в том числе положение и скорость движения, измеряются непосредственно с помощью энкодера и имеют размерности, основанные на отсчетах энкодера. Положение отображается непосредственно в отсчетах энкодера, скорости выражаются в RPM - оборотах в минуту. Скорость движения рассчитывается контроллером на основе данных об измеренной скорости и значении количества импульсов энкодера на один полный оборот оси мотора, указанных в блоке настроек обратной связи во вкладке Настройка кинематики движения (Шаговый двигатель), (DC мотор). Отметим, что в случае использования DC мотора режимы поддержания заданной скорости, движения в заданную точку и все производные от них работают с помощью алгоритмов ПИД-регулирования и требуют соответствующих настроек. Для шаговых двигателей режим ведущего энкодера оптимизирует управление двигателем, за счет чего уменьшаются шумы при движении, стабильно проходятся резонансные скорости, достигается большая скорость вращения по сравнению с режимом работы без энкодера, без риска потери шагов, при которых сбивается координата и требуется повторная калибровка.

Новое! Новый алгоритм управления двигателем включен в последнюю версию прошивки. Алгоритм использует обратную связь с замкнутым контуром по энкодеру, подавляя колебания двигателя и акустические шумы. Этот алгоритм позволяет практически любым моторам двигаться в несколько раз быстрее без потери шагов. Все это поставляется с бесплатным обновлением прошивки и программного обеспечения. Новый алгоритм доступен с прошивкой 4.0.7+, которую можно загрузить с нашего сайта или обновить с помощью XiLab. Используйте XiLab 1.13.13+ и выставите параметр "Feedback" на "Encoder" в настройках Device configuration -> Stepper motor. Обратите внимание: значение позиции теперь отображается в отсчетах энкодера.

Подключение энкодера

Подключение энкодера к контроллеру осуществляется через разъем DSub, который есть во всех системах: плата контроллера, одноосная и двухосная в корпусе.

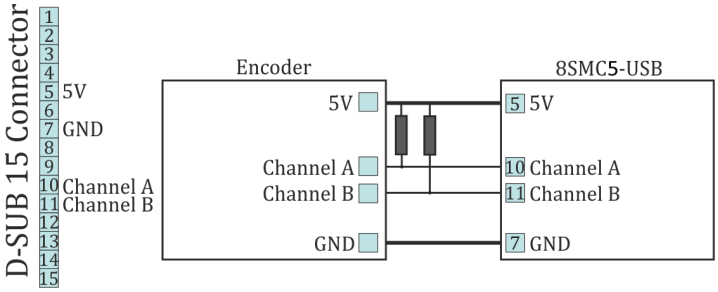


Схема подключения простого энкодера к разъему DSub.

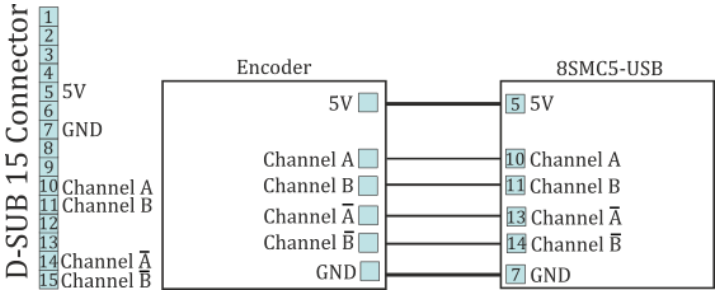


Схема подключения дифференциального энкодера к разъему DSub.

Также смотрите раздел пример подключения простого мотора.

Предупреждение. Входы энкодеров контроллера внутренне подтянуты к логической единице сопротивлением 5.1 килоОм. Обычно выходы энкодера имеют тип "открытый коллектор" с внутренним подтягивающим резистором. При передаче данных, они обеспечивают хорошие показатели перехода из высокого логического уровня в низкий. Но переход из логического 0 в логическую 1 оказывается более плавным. Он происходит через RC цепь, образованную сопротивлением подтяжки и ёмкостью кабеля. Это особенно важно для длинных кабелей. Если встроенной подтяжки недостаточно, то для улучшения показателей скорости перехода 0 - 1 можно добавить подтягивающий резистор R=1.5k Ом к +5 В на каждый выход, проверив, что открытый коллерктор энкодера способен пропускать ток 5 мА. Схема включения резисторов показана выше. Максимальной скорости работы инкрементального квадратурного энкодера можно достичь добавив к его выходу драйвер push-pull с выходным током более 10 мА, который обеспечивает резкие фронты переходов 0 - 1 и 1 - 0.

4.3.6. Датчик оборотов

Датчик оборотов предназначен для регистрации остановки (срыва) движения шагового двигателя (ШД) и более точной калибровки "домашней" позиции (см. Автокалибровка "домашней" позиции).

Контроллер может получать данные о текущей позиции с внешнего датчика оборотов, установленного на оси ШД. Датчик передает сигналы в контроллер один или несколько раз за один оборот двигателя.

Обычно датчик оборотов представляет собой маленький диск с точной шкалой деления, который устанавливается на ось ШД. С разных сторон диска, напротив друг друга, расположены источник (светодиод) и регистратор оптопары. Когда отсечка шкалы не находится между светодиодом и регистратором, датчик «открыт» (на выход оптопары подается логический ноль). Когда отсечка закрывает источник света от детектора, то датчик на выход подает логическую единицу.

Контроллер по умолчанию воспринимает низкий логический уровень как активное состояние датчика оборотов. Вход контроллера притянут к единичному логическому уровню, так что неподключенный датчик оборотов считается неактивным состоянием. При необходимости можно инвертировать вход контроллера и активным будет считаться логический уровень единицы.

Схема подключения

Выводы для подключения датчика оборотов во всех системах (плата контроллера, одноосная и двухосная в корпусе) расположены на разъёме D-SUB

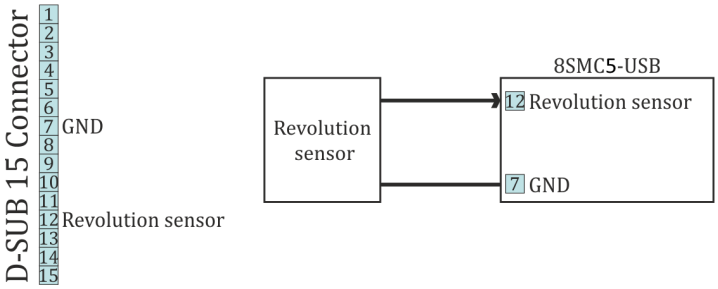


Схема подключения датчика оборотов к системе на базе 8SMC5-USB

4.3.7. Обнаружение потери шагов

Данный режим используется главным образом если производится работа с шаговым двигателем на предельных скоростях или нагрузках, где возможно застревание оси, приводящее к потере шагов. В этом случае дополнительный датчик положения (датчик оборотов или энкодер) позволяет отследить этот момент уведомить пользователя. Данная функция используется **только совместно с шаговыми двигателями** и позволяет обнаруживать потерю шагов. Все координаты и положения оси мотора измеряются в шагах и микрошагах.

При использовании энкодера в контроллере сохраняется значение количества шагов двигателя и отсчётов энкодера на оборот (см. вкладку XiLab Настройка кинематики движения). При включении функции, контроллер сохраняет текущую позицию в шагах ШД и текущую позицию по данным энкодера. Далее, в ходе движения, позиция по энкодеру преобразовывается в шаги и, если разница оказывается больше заданного значения, осуществляется индикация проскальзывания и переход в режим Alarm, при включении соответствующей настройки. Подробно использование энкодера как датчика потери шагов описано в разделе Работа с энкодерами.

При использовании датчика оборотов позиция контролируется по нему. По активному и неактивному фронтам на входе от датчика оборотов контроллер запоминает текущее положение в шагах. Далее, при каждом обороте (количество шагов на один полный оборот мотора устанавливается параметром *Steps per turn*, см. Настройка кинематики движения (Шаговый двигатель)) контроллер проверяет, на сколько шагов сместилась ось. При рассогласовании более чем на заданное значение ошибки *Minimal error* (устанавливается в настройках контроля позиции, см. Контроль позиции) осуществляется индикация проскальзывания флагом структуры состояния. Если в настройках установлен соответствующий флаг, то при выявлении ошибки контроллер переходит в режим Alarm и мотор останавливается, иначе продолжает свое движение. Если флаг индикации проскальзывания активен, то контроллер переходит в режим режим Alarm при включении соответствующего параметра в настройках.

Так же в настройках контроля позиции может быть включен режим автоматической коррекции позиции в случае потери шагов. Если эта опция включена, то при обнаружении проскальзывания контроллер останавливает вращение, корректирует шаговую позицию на основании данных энкодера и пытается запустить вращение заново.Флаг ошибки контроля позиции устанавливается, когда позиция сбивается и автоматически снимается после корректировки. В случае невозможности скорректировать позицию устанавливается флаг ошибки контроля позиции и контроллер переходит в режим Alarm. Если потеря шагов происходит в процессе движения, то статус движения не сбрасывается во время коррекции позиции. Если потеря шагов происходит в режиме удержания позиции, то для восстановления корректной позиции подаётся команда движения в позицию, в которой ось мотора находилась до потери шагов.

> *Замечание.* Для использования функции коррекции позиции нужен энкодер с разрешением не менее двух отсчётов на шаг мотора.

> *Замечание.* Для корректной работы коррекции позиции контроллеру нужно дать постоять с запитанными обмотками в течение 1 секунды для калибровки. После перехода конторллера в режим Alarm или изменения настроек требуется повторная калибровка.

> *Замечание.* При использовании автоматической коррекции позиции не рекомендуется устанавливать значение *Threshold* более 3 шагов т.к. в этом случае не любое проскальзывание будет скорректировано.

> *Замечание.* Команды резкой и плавной остановки могут быть проигнорированы контроллером во время коррекции позиции. В этом случае можно послать команду плавной остановки дважды, что приведёт к снятию питания с обмоток мотора.

> *Замечание.* Если Вы пользуетесь программными концевиками, то использовать автоматическую коррекцию позиции не рекомендуется т.к. положения программных концевиков будут изменяться в процессе коррекции позиции.

> *Замечание.* Команда резкой остановки запускает процесс перекалибровки положения датчика оборотов, причём калибровка происходит при срабатывании датчика оборотов во время движения, управляемого двигателем. Это значит, что если сразу после резкой остановки повернуть ось руками, то после начала движения проскальзывание не будет обнаружено, т.к. калибровка ещё не была произведена.

Замечание. Если датчик оборота двигателя подвержен дребезгу (механическому), то на очень малых скоростях возможны ложные срабатывания контроля по датчику оборотов.

Замечание. Контроль позиции по датчику оборотов не может обнаружить вращение оси при нулевой внутренней скорости. Т.е. если остановить двигатель и руками провернуть ось, то это не будет обнаружено.

4.3.8. Управление питанием мотора

Снижение тока потребления

Для уменьшения энергопотребления шагового двигателя в режиме ожидания контроллер позволяет задавать уровень потребляемого тока в состоянии остановки ниже номинального значения. Этот режим по умолчанию активирован. Он повсеместно используется для снижения нагрева шагового двигателя в режиме удержания при сохранении точности нахождения в заданной позиции. Уровень тока удержания задаётся в процентах от номинального уровня тока в обмотках. Также определяется время в миллисекундах, через которое ток будет снижен. Опцию снижения тока можно отключить специальным флагом. Для настройки функции снижения тока удержания смотрите функцию set_power_settings (см. раздел Руководство по программированию) или вкладку настроек XiLab - Настройка параметров энергопотребления. Установка номинального тока шагового двигателя осуществляется командой set_engine_settings (см. Руководство по программированию или раздел Настройка кинематики движения (Шаговый двигатель)).

Разумным значением уровня сниженного тока удержания является 40-70%. Это снижает энергопотребление в 2-4 раза, а сила удержания обычно остаётся достаточной. Время, через которое ток снижается разумно выбирать в диапазоне 50-500 мс. Это время окончания низкочастотных колебаний механической системы, которые могут сбить позицию удержания в некоторых системах.

Отключение питания мотора

Также для уменьшения энергопотребления шагового двигателя существует режим отключения питания мотора по таймеру. Это необходимо в основном для предотвращения траты энергии на удержание позиции, когда работа с установкой закончена и никаких движений не происходит долгое время. Этот режим по умолчанию активирован, но может быть отключен пользователем. Время от остановки до отключения настраивается в секундах. Разумным временем является 3600 секунд (один час). Для настройки функции отключения питания мотора смотрите функцию set_power_settings (см. раздел Руководство по программированию) или вкладку настроек XiLab - Настройка параметров энергопотребления.

Специфика расчёта временных задержек

Все временные задержки работают следующим образом: при каждом переходе в состояние остановки двигателя запоминается время с точностью до миллисекунды. Далее, при достижении заданных пользователем таймаутов и включенности функций PowerOff/CurrentReduce происходит отключение питания двигателя или снижение тока в обмотках. Все настройки можно менять онлайн. Например, если увеличить время таймаута PowerOff после того, как он уже случился, то обмотки запитаются и функция PowerOff сработает снова по достижению таймаута от момента остановки двигателя. То же самое касается включения и отключения флагов использования режимов PowerOff/CurrentReduce. Отсчёт таймаутов останавливается и функции PowerOff/CurrentReduce отменяются при любом начале движения.

Функция Jerk free

Иногда необходимо всегда плавно менять ток в обмотках двигателя для устранения вибраций механической системы. Для этого в контроллере предусмотрена опция Jerk free, где можно задать скорость выхода тока через обмотки с нуля на номинальное значение с точностью до миллисекунды. Опция включается соответствующим флагом. При этом все изменения тока стабилизации или отключения обмоток будут проходить с предварительным плавным набором или сбросом тока удержания. Например, если установлена скорость набора тока 100 мс и происходит событие снижение тока удержания до 50%, то он будет снижен плавно за 50 мс (а не 100 мс, ведь 100 мс нужно чтобы полностью сбросить ток до нуля). Также за 50 мс ток будет снова набран до номинального при новом движении. Для настройки функции Jerk free смотрите функцию set_power_settings (см. раздел Руководство по программированию) или вкладку настроек XiLab - Настройка параметров энергопотребления. Установка номинального тока шагового двигателя осуществляется командой set_engine_settings (см. Руководство по программированию или раздел Настройка кинематики движения (Шаговый двигатель)).

Функция плавного набора тока работает при любом изменении амплитуды тока в обмотках, например при смене номинального тока удержания. При этом скорость увеличения или уменьшения тока рассчитывается на основе максимального из введённых токов удержания: старого или нового. Если обмотки нужно отключить, то ток снижается до нуля, а только затем силовые выходные цепи контроллера обесточиваются. Если обмотки нужно запитать номинальным током, то они запитываются нулевым током и далее ток растёт до номинального.

Существуют исключения, когда ток мгновенно сбрасывается до нуля и обмотки отключаются, даже когда функция Jerk free включена. Это события опасности и попадание в состояние Alarm (см. Критические параметры), а также моменты перезагрузки контроллера для обновления программного обеспечения. Все эти события редки и не должны происходить во время работы с позиционером.

Разумным значение времени Jerk free будет 50-200 мс, так как это приведёт лишь к низкоэнергетичным вибрациям на частоте 3-10 Гц, которые будут значительно меньше вибраций от бытовых шумов (шагов, сквозняка). Установка большого времени Jerk free и функции снижения тока для экономии электроэнергии и снижения нагрева мотора приведёт к постоянным задержкам для сброса и набора тока. Поэтому время Jerk free не стоит делать большим.

4.3.9. Критические параметры

Для безопасности работы контроллера и двигателя устанавливаются максимальные и минимальные значения токов, напряжений, температур. Выход из допустимого диапазона для любого из этих параметров приводит к тому, что движение прекращается, обмотки мотора обесточиваются, контроллер переходит в состояние **Alarm**. Выход из состояния **Alarm** возможен только при устранении причины превышения критического параметра и посылки команды **STOP**. Настройки используются для всех типов двигателей.

Доступны следующие параметры:

- Low voltage off - определяет минимальное значение напряжения силового питания контроллера (измеряется десятками мВ). Включается флагом Low voltage protection. Иначе минимальный порог отключения не действует. Разумное значение 6000-8000 мВ, для рабочего диапазона питания 12-48В. Эта защита помогает обнаружить момент, когда блок питания отключился из-за срабатывания одной из его защит. Такое может произойти со стабилизированным блоком питания при превышении его рабочей мощности.
- Max current (power) - определяет максимальное значение тока силового питания контроллера (измеряется в мА). Устанавливать разумно в два раза выше, чем максимальный зарегистрированный рабочий ток потребления во время тестов. Для регистрации тока потребления используйте графики XiLab.
- Max voltage (power) - определяет максимальное значение напряжения силового питания контроллера (измеряется десятками мВ). Это ограничение разумно брать на 20% выше, чем рабочее напряжение блока питания.
- Temperature - определяет максимальное значение температуры микропроцессора (измеряется десятыми долями градуса Цельсия). Микропроцессор выдерживает рабочую температуру 75 градусов Цельсия, но не перегревается сам по себе. Повышение его температуры может косвенно свидетельствовать о перегреве силовой части платы. Значение порога перегрева разумно выбирать в диапазоне 40-75 градусов.

Флаги:

- ALARM_ON_DRIVER_OVERHEATING - Входить в состояние Alarm по превышению критической температуры драйвера (>125 градусов). Силовой драйвер сигнализирует о приближении его температуры близко к критической. Если драйвер не отключить, то при дальнейшем нагреве он отключит себя сам. Рекомендуется не доводить до принудительного отключения и установить флаг добровольного отключения.
- H_BRIDGE_ALERT - Входить в состояние Alarm при неполадках в силовом драйвере, вызванных безусловным отключение по перегреву или повреждением платы контроллера. Этот флаг должен быть установлен.
- ALARM_ON_BORDERS_SWAP_MISSET - Входить в состояние Alarm обнаружении срабатывания не того концевика, к которому осуществлялось движение (см. Концевые выключатели). Служит для более понятной индикации срабатывая подсистемы обнаружения перепутанности концевиков. Рекомендуется держать флаг включенным.
- ALARM_FLAGS_STICKING - Этот флаг настраивает "залипание" индикаторов произошедшей ошибки в статусной структуре контроллера. Иначе флаги ошибок активны только пока происходит событие, вызывающее ошибку. Если ошибка носила кратковременный характер и её причина самостоятельно исчезла, то иногда неясна причина попадания в состояние Alarm. Для этого удобно включить "залипание" и на главном окне XiLab диагностировать причину попадания в Alarm.
- USB_BREAK_RECONNECT - Этот флаг настраивает работу блока перезагрузки USB шины при потере связи. При установке этого флага данный блок начинает функционировать и отслеживать потерю связи по USB шине (к примеру, в случае удара статическим разрядом).

Установка параметров описана в меню программы XILab " Настройка предельных параметров контроллера ". Команды установки максимально допустимых значений описаны в руководстве по программированию.

4.3.10. Хранение параметров во flash-памяти контроллера

Контроллер позволяет сохранять все свои настройки в энергонезависимую память. При подаче питания на контроллер он восстанавливает настройки из этой памяти и мгновенно готов к работе. Не нужно каждое включение питания настраиваться на позиционер заново. Контроллер хранит в своих настройках своё имя, вводимое пользователем. Это удобно для его последующей идентификации.

В энергонезависимую память сохраняются все текущие рабочие параметры контроллера, относящиеся к вкладке Device configuration из меню настроек программы XILab. Это делается с помощью кнопки **Save to flash** в программе XILab или с помощью функции command_save_settings (см. Руководство по программированию).

Можно восстановить в ОЗУ контроллера все настройки из энергонезависимой памяти не только при подаче питания, но и при нажатии в XILab на кнопку **Restore from flash**, что позволяет работать с сохраненными во flash данными. Для загрузки можно использовать функцию command_read_settings, см. Руководство по программированию. Восстановленные настройки станут активны немедленно. При этом произойдёт переинициализация всех блоков контроллера.

4.3.11. Пользовательские единицы координат

Текущая координата контроллера выводится и задается в шагах шагового двигателя или отсчетах энкодера, если энкодер присутствует и включен. При работе с подвижками может быть удобно задавать позицию в миллиметрах для трансляторов, в градусах для ротаторов, или в других естественных единицах. Для этого программное обеспечение контроллера позволяет пересчитывать координаты в пользовательские единицы. Если пользователь знает какому линейному перемещению соответствует смещение шагового двигателя на определенное количество шагов, он может задать это соотношение как коэффициент пересчета и далее отдавать команды движения и наблюдать за координатой подвижки в этих единицах. Это касается и интерфейса XiLab, и использования в собственных программах и скриптах. Скорость и ускорение также задаются в единицах, производных от пользовательских (например в миллиметрах в секунду). Установка нулевой позиции делается одинаково для отсчёта в шагах или в пользовательских единицах.

В XiLab отображение пользовательских единиц можно включить на вкладке Настройки отображения пользовательских единиц. Можно установить подходящее имя пользовательских единиц.

При работе с библиотекой libximc функции, принимающие и возвращающие величины в пользовательских единицах имеют имена оканчивающиеся на _calb. Эти функции дополнительно принимают как параметр калибровочную структуру calibration_t, см. Руководство по программированию.

4.4. Безопасная работа

Несколько настроек контроллера непосредственно связаны с безопасностью работы. Неправильная их установка может повредить позиционер или контроллер. Позиционер можно повредить превышением мощности, скорости вращения и выходом за пределы допустимого диапазона движения. Обычно для безопасной работы достаточно загрузить заранее подготовленный профиль для Вашего позиционера, где все необходимые настройки уже сделаны.

Границы движения и концевики

Линейные позицинеры имеют ограниченный диапазон перемещения в отличие от круговых ротаторов. Выход такого позиционера за допустимые физические границы его перемещения - основная причина заклинивания или выхода позиционеров из строя. Для предотвращения таких поломок диапазон перемещения позиционера ограничивается в соответствии с требованиями пользователя. Для этого используются Концевые выключатели, но в части случаев, например. когда позиционер не оборудован концевиками или имеет только один концевик, границы движения могут определяться программно (см. Концевые выключатели). Часто бывает, что концевики перепутаны местами. В этом случае воспользуйтесь механизмом обнаружения перепутанности концевиков, описанным в разделе Концевые выключатели, чтобы первое же движение до границы не привело к заклиниванию позиционера. Настройка диапазона движения и концевых выключателей описана в соответствующем разделе. Команды настройки описаны в Руководстве по программированию.

Ограничители движения

Шаговый двигатель имеет основную настройку безопасности - номинальный ток в обмотках. Это основной параметр, определяющий мощность, подаваемую на двигатель. Номинальный ток должен быть установлен не выше допустимого для данного двигателя. Подробней смотрите главу Ограничители на двигателях. Для DC двигателей максимальный ток является ограничивающим и должен быть установлен согласно максимально допустимому току через DC двигатель. Если не известен максимальный ток, то может быть ограничено максимальное напряжение, подаваемое на двигатель. Это также будет препятствовать его перегреву, хотя ограничение напряжения это более грубый способ, чем ограничение тока. Подробней смотрите главу Ограничители на двигателях.

Повредить позиционер или способствовать его быстрому износу может превышение скорости вращения. Необходимо поставить флаг ограничения скорости не выше максимальной и установить правильную максимальную скорость для данного позиционера. Подробней смотрите главу Ограничители на двигателях.

Критические параметры

Контроллер отслеживает токи и напряжения, которые возникают в его цепях и способен реагировать на их подозрительные значения. Реакция обесточивает двигатель и препятствует дальнейшему движению, пока причина проблемы не устранена. Это позволяет отследить замыкания обмоток двигателя на друг друга или на землю, которое может возникнуть при повреждении кабеля позиционера или самого позиционера. Также реакция носит информативный характер, позволяя отследить некорректные значения напряжения питания или приближающийся перегрев. Поэтому прочтите главу Критические параметры и установите необходимые защиты. При возникновении опасного состояния контроллер переходит в режим Alarm и главное окно программы XiLab приобретает красный оттенок. Если такое произошло, то отследите и устраните причину опасности прежде чем отключать режим Alarm. Если Вы пользуетесь собственным приложением для управления двигателем, то обращайте повышенное внимание на флаг состояния Alarm (см. Статус контроллера).

Работа с энкодером

Если при подключении энкодера перепутать каналы датчика, то при движении двигателя в положительном направлении, энкодер будет показывать уменьшение координаты. При работе с DC двигателем, данная ситуация появится если перепутать каналы управления двигателем (DC+ и DC-). Для того, чтобы исправить эти ошибки достаточно установить флаг Encoder Reverse, указанный в блоке настроек обратной связи во вкладке Настройка кинематики движения (Шаговый двигатель) для шагового двигателя и Настройка кинематики движения (DC мотор) для DC двигателя.
Так же возможна ситуация, когда нет контакта с одним из каналов энкодера. В этом случае, при движении двигателя показания датчика будут колебаться в диапазоне [-1..1] от начальной позиции.

При работе с DC двигателем обе эти ошибки приведет к некорректной работе алгоритма управления, описанном в пункте ПИД алгоритм для управления DC двигателем. Если вы впервые подключили новый DC двигатель, то перед началом работы настоятельно рекомендуется выполнить проверку подключения энкодера. Для этого, установите следующие значения коэффициентов регулирования: $K_P = 1, K_I = 0, K_D = 0$ и попытайтесь выполнить движение вправо или влево с небольшой скоростью. После начала движения проверьте, что показания энкодера изменяются в соответствии с выбранным направлением. Если необходимо установите флаг Encoder Reverse.

4.5. Дополнительные функции

1. Индикация режима работы
2. Работа с магнитным тормозом
3. Управление с помощью джойстика
4. Управление кнопками "вправо" и "влево"
5. ТТЛ-синхронизация
6. Создание многоосных систем
7. Цифровой вход-выход общего назначения
8. Аналоговый вход общего назначения
9. Интерфейс управления внешним драйвером
10. Последовательный порт
11. Хранение позиции во FRAM-памяти контроллера
12. Опознавание позиционеров Standa

4.5.1. Индикация режима работы

Статус контроллера

В контроллере предусмотрена индикация режима работы. Для этого на плате расположен один двухцветный светодиод.
Зелёный индикатор Power показывает наличие питания 3,3 В у контроллера.
Красный индикатор Status - отображает режим работы контроллера.
Одновременное горение двух цветов выглядит как **жёлтое свечение**;

Частота мерцания, Гц	**Описание**
потушен	устройство выключено, нет питания
горит зелёным	устройство неисправно или микропрограмма не загружена
горит желтым	устройство в состоянии Alarm
0,25	устройство работает, но нет связи по USB с ПК
1	устройство работает, движение остановлено
4	устройство работает, в движении
8	устройство в процессе перепрошивки
10	устройство в процессе переподключения шины USB

Режимы работы индикатора Power/Status

Индикация концевых выключателей

В контроллере реализована индикация срабатывания концевых выключателей. Высокий логический уровень появляется на соответствующем выводе в момент активности концевого выключателя. Активное состояние определяется исходя из настроек концевых выключателей (см. Настройка диапазона движения и концевых выключателей).

Схема подключения

> *Замечание.* При использовании дополнительных светодиодов, они должны быть рассчитаны на рабочий ток 4 мА. Дополнительных резисторов, ограничивающих ток, не требуется. Для типовых светодиодов рабочий ток будет около 2 мА. Не рекомендуется использовать светодиоды синего и фиолетового цвета из-за их высокого запирающего напряжения и, как следствие, низкой яркости.

Плата контроллера

Индикаторы Power, Status продублированы выходами на многоцелевом 20 пиновом коннекторе BPC. Светодиоды подключаются к ним в соответствие со схемой ниже.

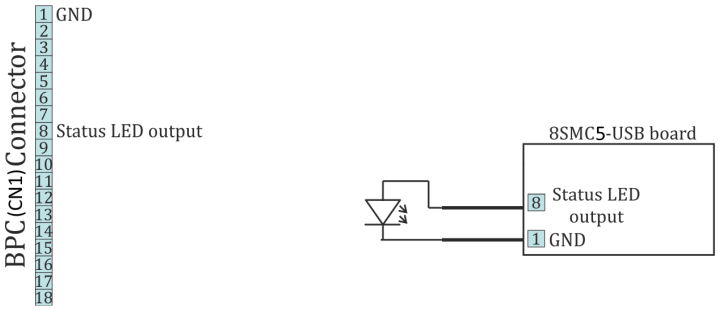


Схема подключения индикаторов Power и Status к плате контроллера

Концевые выключатели расположены на том же самом разъёме. Схема подключения указана ниже. Для индикации срабатывания концевых выключателей удобно использовать светодиоды рассчитанные на нужный ток.

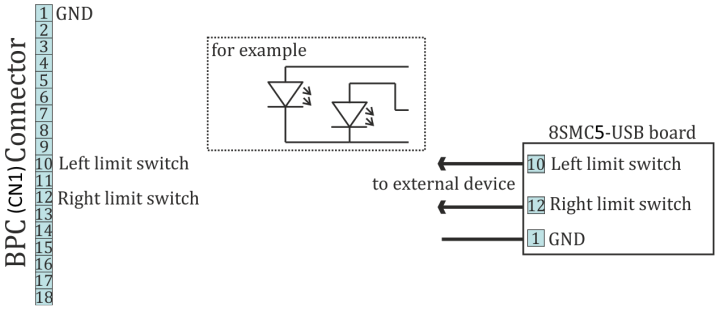


Схема подключения индикаторов концевых выключателей к плате контроллера

Одноосная и двухосная системы

В коробочных версиях контроллеров (одноосная и двухосная системы) светодиоды на передней панели представляют собой индикаторы питания, статуса и концевых выключателей, поэтому никакой схемы подключения не требуется.

4.5.2. Работа с магнитным тормозом

На разъеме BPC есть вывод для управления магнитным тормозом, установленным на ось шагового двигателя. Магнитный тормоз используется для удержания положения мотора при отсутствии питания.

Описание работы

Магнитный тормоз состоит из магнита и пружины, осуществляющей остановку оси мотора. При отсутствии напряжения на магните пружина зажимает ось в текущем состоянии, что позволяет сохранять необходимое положение мотора. После подачи напряжения на магнит, пружина освобождает ось.

Последовательность работы контроллера при отключении подвижки

Остановка мотора (время остановки запоминается в контроллере) -> Отключение магнита от питания, фиксация вала -> Отключение питания платы
При включении подвижки последовательность работы контроллера обратная.

Поскольку любое движение инерционно, для управления магнитным тормозом и процессом фиксации положения устанавливаются следующие параметры:

- Время между включением питания мотора и отключением тормоза (мс)
- Время между отключением тормоза и готовностью к движению (мс)
- Время между остановкой мотора и включением тормоза (мс)
- Время между включением тормоза и отключением питания (мс)

При отключении функции магнитного тормоза контроллер непрерывно подаёт сигнал отжатия тормоза. Это позволяет двигать мотор, оснащённый магнитным тормозом, не используя фиксацию ротора при остановках.
При отключении функции обесточивания обмоток контроллер отрабатывает только задержки между переключением тормоза и началом/остановкой движения.

Все настройки магнитного тормоза можно изменять онлайн и тормоз будет переключаться в такой режим, который был бы если бы настройка всегда имела новое значение. Например, значительное увеличение задержки срабатывания тормоза, когда тормоз уже сработал, приведёт к тому, что тормоз снова будет отведён и по достижению новой задержки от момента остановки снова сработает. Так же можно отключать и включать сам магнитный тормоз или функцию запитывания обмоток.

Тип	ТТЛ
Активное состояние (тормоз отжат)	3.3 В
Неактивное состояние (тормоз незапитан)	0 В
Рабочий ток	не более 4 мА

Электрические параметры вывода

Настройка магнитного тормоза в программе XILab описана в разделе " Настройка тормоза".

Схема подключения

Для работы с магнитным тормозом используется специальная плата, управляемая цифровым сигналом. Одноосные и двухосные системы, оборудованные такой платой и, соответственно, имеющие возможность работать с магнитным тормозом, поставляются отдельно (см. ниже).

Плата контроллера

Контакт, отвечающий за управление магнитным тормозом, расположен на разъеме BPC.

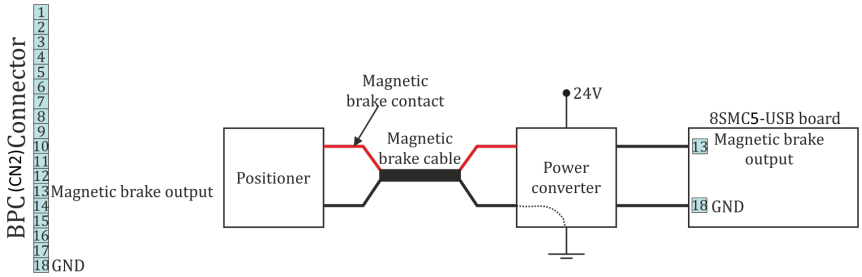


Схема подключения магнитного тормоза к плате контроллера

Power converter - это преобразователь цифровых сигналов в силовые. При высоком уровне на ножке Magnetic brake output, на контакт магнитного тормоза подвижки подаётся 24В, при низком - с него убирается напряжение. В простейшем случае это схема, построенная с использованием транзистроного ключа и диода. За более детальной информацией обращайтесь в техподдержку: 8SMC4@standa.lt.

Одноосная и двухосная система

Для того, чтобы использовать магнитный тормоз, необходимо, чтобы система с контроллером была оборудована специальной платой-преобразователем. Модели, отвечающие этому требованию, можно опознать по буквам **BR** в названии, например **8SMC5-USB-B8-1BR**.

Контакт, отвечающий за управление магнитным тормозом в коробочных версиях контроллера, расположен на 26 пиновом D-SUB разъеме. Схема подключения для двух разных систем указана ниже. Двухосные системы поставляются только с одной осью, которая может работать с магнитным тормозом.

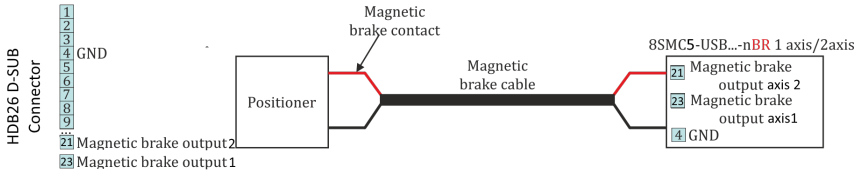


Схема подключения магнитного тормоза к одноосной или двухосной системе

4.5.3. Управление с помощью джойстика

Основная информация

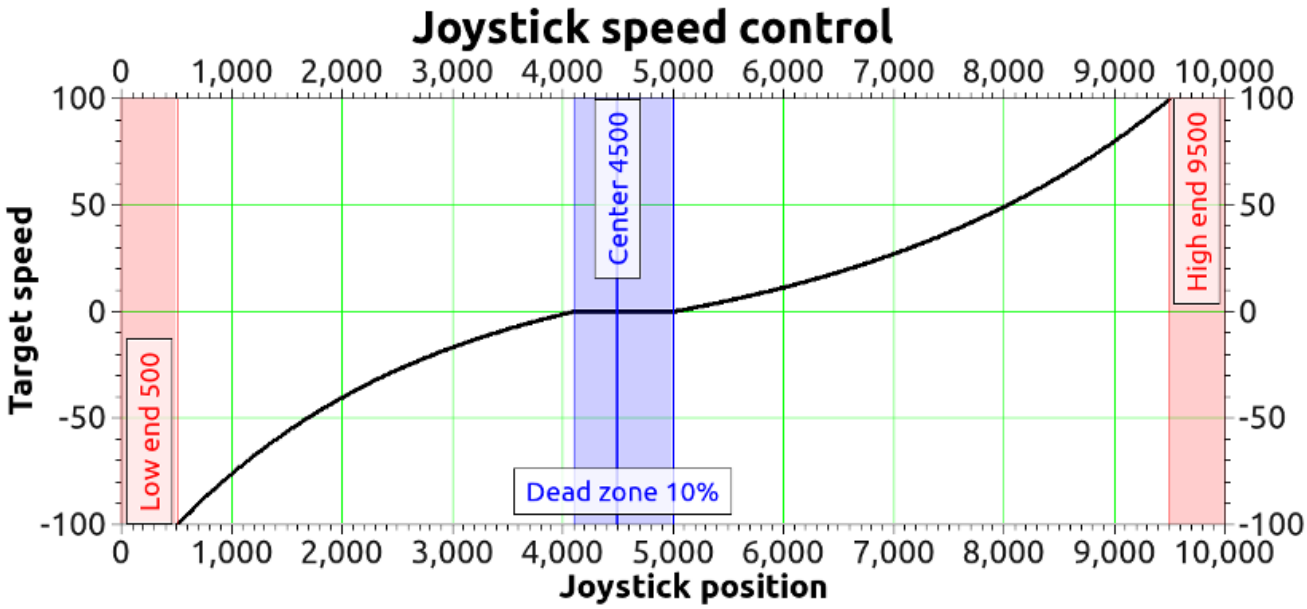
Контроллер позволяет работать с джойстиком, который выдает аналоговое напряжение в диапазоне 0-3 В. При этом напряжение в равновесном (центральном) положении, а также напряжения максимального и минимального отклонений могут быть заданы любыми в рабочем диапазоне напряжений, соблюдая условие: минимальное отклонение < центральной позиции < максимальное отклонение. Контроллер использует числовое представление напряжения на выходе джойстика: 0 В контроллер сопоставляет значению 0, а напряжению 3 В - 10000.

Для того, чтобы движение могло остановиться в центральном положении, предусмотрена мертвая зона *DeadZone*, отсчитываемая от центрального положения и измеряемая в процентах. Внутри *DeadZone* контроллер вызывает остановку движения. При отклонении джойстика от центрального положения, выводящем его из *DeadZone* начинается движение со скоростью, определяемой отклонением джойстика от границы *DeadZone* до максимального отклонения. Связью направления движения джойстика и его отклонения можно управлять с помощью флага реверса, что может быть удобно для соответствия: "отклонение вправо" - "движение вправо", - независимо от физической ориентации джойстика и подвижной части.

Скорость движения экспоненциально зависит от отклонения джойстика. Это позволяет небольшими отклонениями достигать высокой точности подводки позиции, а сильными отклонениями джойстика вызывать быстрые перемещения. Параметр нелинейности можно менять. При установке параметра нелинейности в 0 скорость движения двигателя начинает линейно зависеть от отклонения джойстика.

На графике приведён пример зависимости скорости движения от отклонения джойстика для следующих настроек:

Центральное отклонение	4500
Минимальное отклонение	500
Максимальное отклонение	9500
Мёртвая зона *DeadZone*	10%
Максимальная скорость движения	100



Пример зависимости скорости движения от отклонения джойстика.

Экспоненциального отклика джойстика, сочетающего высокую точность и скорость, может быть недостаточно. Поэтому контроллер поддерживает таблицу максимальных скоростей, между которыми можно переключаться кнопками управления "влево"/"вправо". В этом случае при нажатии на кнопку "вправо" скорость, соответствующая 100% отклонения, меняется с *MaxSpeed*[i] на *MaxSpeed*[i+1]. Если нажата кнопка "влево", максимум скорости меняется с *MaxSpeed*[i+1] на *MaxSpeed*[i]. При старте контроллера i=0. Количество скоростей в таблице - 10. Если *MaxSpeed*[x] равна нулю (целая и дробная части), то перейти на эту скорость с *MaxSpeed*[x-1] нельзя. Это сделано для возможности ограничить таблицу меньшим количеством скоростей. Попытка выйти за границы индекса таблицы скоростей (0-9) также ни к чему не приводит.

Контроллер подавляет дребезг контактов на кнопках управления. Для срабатывания кнопок длительность нажатия должна превышать 3 мс.

Если джойстик находится внутри мёртвой зоны более 5 секунд, то он не будет считаться вышедшим из неё, пока он не пробудет вне *DeadZone* более 100 мс. Это позволяет отпустить джойстик и быть уверенным, что даже случайный шум на выходе джойстика не приведет к ненужным сдвигам мотора. Пока джойстик находится внутри *DeadZone* контроллер способен принимать любые команды с компьютера, в том числе и команды движения, калибровки домашней позиции и т.п. Если при выполнении команды джойстик выводится из *DeadZone*, то команда движения отменяется и двигатель начинает подчиняться управляющему воздействию джойстика. Это позволяет включить режим управления двигателем с помощью джойстика, но не пользоваться им без надобности. А при касании джойстика он перехватит управление.

К режиму джойстика применимо все, что относится и к движению под воздействием управляющих команд: подчинение ускорению, ограничение максимальной скорости, режимы отключения обмоток при простое, работа с магнитным тормозом, компенсация люфта и т. д. Например, если резко бросить ручку джойстика внутрь *DeadZone*, то, при включении соответствующих режимов, контроллер плавно замедлит двигатель, отъедет в сторону для компенсации люфта, остановит двигатель, зафиксирует вал двигателя магнитным тормозом, плавно снизит ток и отключит питание обмоток.

Подключение кнопок "влево"/"вправо" описано в разделе Управление кнопками "вправо" и "влево"
Изменение параметров *MaxSpeed*[i], *DeadZone* описано в разделе Настройка внешних управляющих устройств

Схема подключения

ВАЖНО. Аналоговые входы для подключения джойстика расчитаны на диапазон 0-3В. Будьте внимательны и не подавайте на контакты джойстика напряжение больше 3В.

Плата контроллера

Контакты джойстика на плате контроллера расположены на разъеме BPC. Обратите внимание, что джойстику необходимо питание +3В.

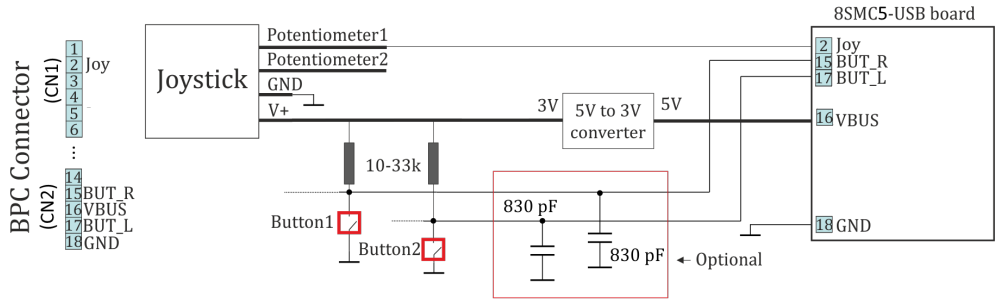


Схема подключения джойстика к плате контроллера через разъём BPC

Одноосная и двухосная система

Разъём для подключения джойстика есть в двухосной и одноосной системе. Схема подключения к нему представлена ниже.

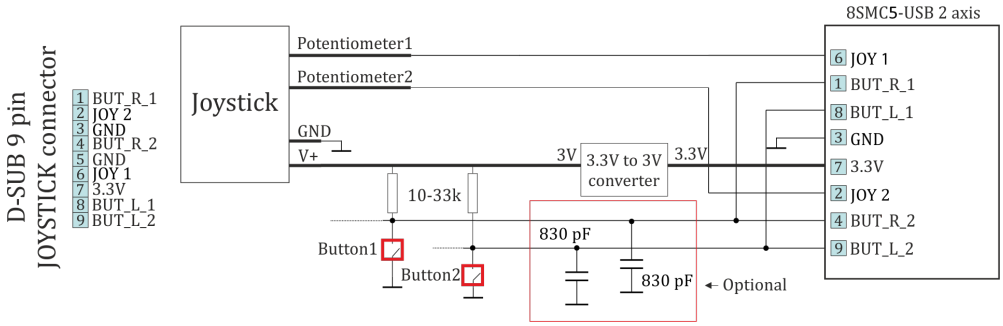


Схема подключения джойстика к двухосной системе в корпусе через разъём D-SUB9

4.5.4. Управление кнопками "вправо" и "влево"

Для каждой системы существует возможность управлять движением моторов при помощи кнопок. Контроллер поддерживает таблицу из 10 скоростей движения MaxSpeed[0-9], которые используются и для управления джойстиком, и при управлении кнопками.

Настройки кнопок передаются/считываются командами SCTL/GCTL (set_control_settings/get_control_settings).

- При кратковременном нажатии (менее *MaxClickTime*) на кнопку вправо или влево мотор сдвигается на заданное расстояние, если *DeltaPosition* и *uDeltaPosition* отличны от нуля.
- При длительном нажатии одной из кнопок, по истечении времени *MaxClickTime* контроллер запускает движение со скоростью MaxSpeed[0] и начинает отсчитываться таймаут Timeout[0]. По истечению каждого таймаута Timeout[i] скорость меняется на с MaxSpeed[i] на MaxSpeed[i+1].
- При одновременном нажатии двух кнопок контроллер совершает остановку с замедлением. Удержание двух кнопок в течении 3 секунд запускает автокалибровку "домашней" позиции.

> *Замечание.* Если вся таблица из 10 скоростей не нужна, то достаточно заполнить только её верхнюю часть. Контроллер не будет менять скорость на следующую, если она равна нулю или если таймаут, который для этого надо отсчитать, равен нулю. Например, если MaxSpeed[0] и MaxSpeed[1] ненулевые, а MaxSpeed[2] равно нулю (включая микрошаговую часть), то контроллер начнёт движение на скорости MaxSpeed[0], перейдёт на скорость MaxSpeed[1] и продолжит движенее с крайней скоростью до момента отпускания кнопки. Для той же функциональности можно сделать Timeout[1] равным нулю, величина скорости MaxSpeed[2] не будет иметь значения.
>
> Движение мотора подчиняется настройкам движения (за исключением устанавливаемой скорости). Например, при переходе от MaxSpeed[i] на MaxSpeed[i+1] мотор может ускоряться до достижения нового значения скорости или менять её скачком, если ускорение отключено.

По умолчанию состояние кнопки задаётся уровнями напряжений согласно таблице 4.5.4.1. Состояние каждой кнопки может быть программно инвертировано. При активном состоянии кнопка считается нажатой. Не имеет значения каким образом состояние становится активным (после изменения настройки инвертирования состояний или же при смене уровня напряжения при физическом воздействии на кнопку). Контроллер использует программное подавление дребезга контактов на кнопках. Кнопка считается нажатой, если активное состояние на входе кнопки длилось более 3-х миллисекунд.

Тип	ТТЛ уровень
Логический нуль (не активно)	0 В
Логическая единица (активно)	3.3 В

Таблица 4.5.4.1 – Параметры вывода

> **Предупреждение.** Если при включении контроллера или его перезагрузке на входе кнопки присутствует уровень напряжения, который считается активным, то контроллер воспримет это как сигнал нажатия кнопки и начнёт подчиняться правилам описанным выше.

Схема подключения

Плата контроллера

К плате контроллера могут быть подключены кнопки управления ("вправо", "влево") через разъём BPC. В ней не встроено подтягивающих резисторов для установки входного потенциала по-умолчанию, поэтому необходимо использовать схему с верхним (pullup) или нижним (pulldown) резисторами.

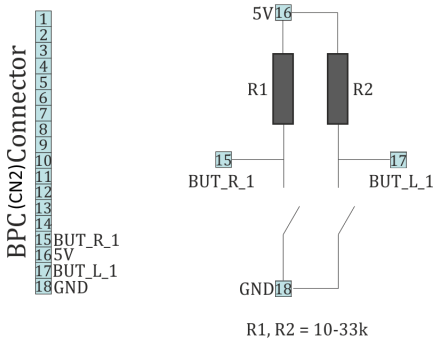


Схема подключения кнопок к разъёму BPC в плате контроллера

Одноосная или двухосная система в корпусе

Для контроллеров в корпусе кнопки уже выведены наружу. Однако есть возможность подключить свои кнопки управления к соответствующим контактам. Они находятся в D-SUB 9 разъёме и есть в двухосной и одноосной системе. Схема подключения приведена ниже.

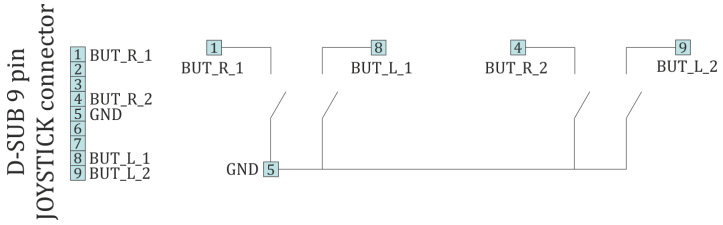


Схема подключения кнопок к разъёму двухосной системы

4.5.5. ТТЛ-синхронизация

Принцип работы

ТТЛ-синхронизация предназначена для синхронизации производимых контроллером движений с внешними устройствами и/или событиями. Например, контроллер может каждый раз при перемещении на заданное расстояние выдавать импульс синхронизации, запускающий какое-либо измерение. И наоборот, при получении импульса синхронизации от внешнего устройства, например означающего, что экспериментальная установка готова к перемещению в следующую измерительную позицию, контроллер может выполнить смещение на заранее заданное расстояние.

Для работы с входными сигналами синхронизации, генерируемыми с помощью механических контактов, предусмотрена защита от дребезга контактов. Можно установить минимальную длительность входного импульса, после которого сигнал синхронизации считается полученным. По умолчанию активным считается состояние логической единицы (см. таблицу 4.5.5.1), а запускающим фронтом - нарастающий. Если такая логика работы входа и выхода синхронизации не подходит, то её можно инвертировать.

Тип	**ТТЛ уровень**
Логический нуль (не активно)	0 В
Логическая единица (активно)	3.3 В

Таблица 4.5.5.1 – Параметры вывода

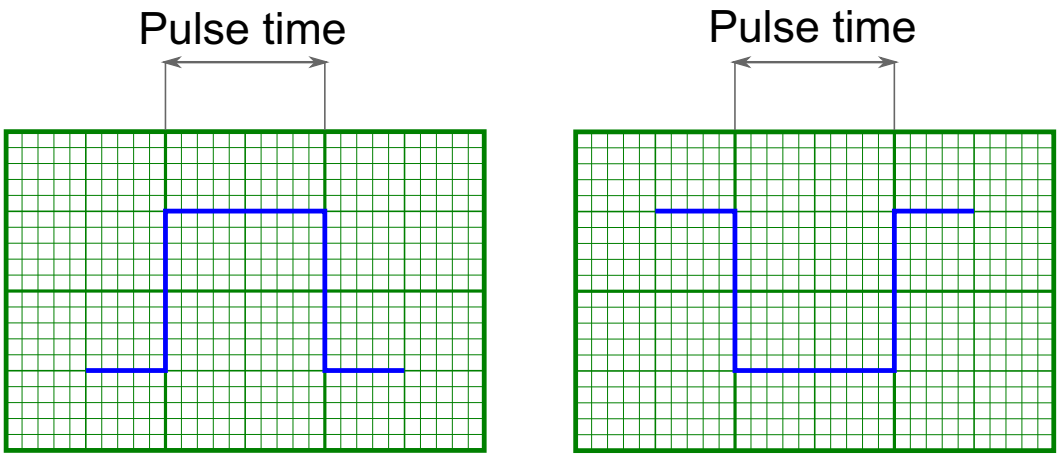


Иллюстрация инвертирования входного или выходного импульса

> *Замечание.* Для одновременности старта многоосных систем минимальная длительность входного сигнала должна быть одинаковой у всех контроллеров. Не следует использовать подавление дребезга механических контактов в системах, где такого эффекта нет, но есть короткие наводки на линию входа синхронизации. Вместо этого достаточно добавить RC цепь, фильтрующую такие фантомные входные импульсы.

Синхронизация важна при создании многоосных систем, так как позволяет нескольким осям начать движение в один и тот же момент времени, для этого все оси подготавливают к началу движения, во всех ведомых осях устанавливают режим начала движения по входному синхроимпульсу, одну ось, ведущую, настраивают на отправку импульса синхронизации, при начале движения. Выход синхронизации ведущей оси соединяют со входами ведомых. После такой настройки и подключения, движение ведущей оси вызывает мгновенный отклик и начало движения всех ведомых.

> *Замечание.* При таком соединении необходимо установить минимальную длительность входного импульса синхронизации на 0. Это отключает защиту от дребезга входного сигнала, но в описанной конфигурации механических контактов нет, следовательно нет и дребезга. Если минимальная длительность входного сигнала не равна нулю, то чтобы избежать неодновременного старта движения ведущей и ведомых осей, необходимо поставить эту длительность одинаковой у всех контроллеров, соединить выход синхронизации не только со входами ведомых контроллеров, но и со входом ведущего контроллера, а дальше подавать запускающий импульс ручным переключением состояния выхода синхронизации.

Вход и выход синхронизации полностью независимы друг от друга и иных способов управления движением. Так управление движением через XILab (см. Главное окно программы XILab в режиме управления одной осью) или пользовательскую программу, управление от джойстика (см. Управление с помощью джойстика), от кнопок ручного управления (см. Управление кнопками "вправо" и "влево"), происходит независимо от состояния входа и выхода синхронизации. Всегда выполняется принцип "приоритет у команды пришедшей позднее". То есть, например, команда движения, поданная через XILab, отменит выполняемое по импульсу входной синхронизации движение, но не повлияет на работу выходной синхронизации, а приход входного синхроимпульса, при соответствующей настройке, отменит текущее движение, инициированное пользовательской программой, заменив его на движение, определяемое настройками работы синхровхода.

Замечание. Настройки синхронизации могут быть сохранены энергонезависимой памяти контроллера, в этом случае, все, что касается работы синхронизации будет относиться и к случаю автономной работы контроллера т.е. вы можете, например, настроить смещение на заданное расстояние по приходу синхроимпульса с выдачей синхроимпульса по завершению смещения, подключить контроллер к измерительной установке, начинающей измерение по входному синхросигналу и выдающему синхроимпульс по завершению измерения и запустить такую измерительную систему без компьютера. После поступления первого импульса, измерения и смещения будут производиться автоматически без участия компьютера.

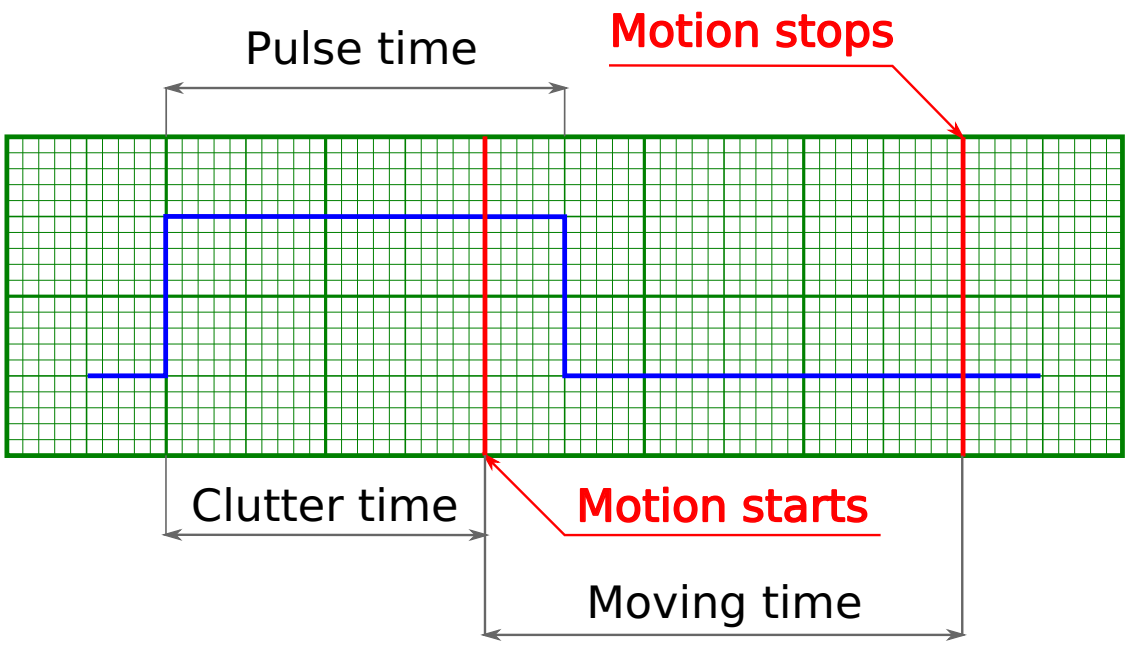
Вход синхронизации

Для входа синхронизации имеется настройка минимальной длительности входного синхроимпульса, который может быть зарегистрирован. Эта длительность задается в микросекундах. Используйте эту настройку для увеличения помехоустойчивости контроллера. Вход синхронизации может быть включен или выключен. Если он включен, то переход из не активного состояния в активное приводит к движению аналогичному выполнению команды Смещение на заданное расстояние, в которой значение смещения задается знаковым *Position* с микрошаговой частью, а скорость *Speed* с микрошаговой частью. Изменение настроек входа синхронизации во время выполнения движения не приводит к изменению параметров движения (скорости, целевой координаты). Эти параметры будут применены при приходе следующего фронта активного состояния на вход синхронизации. Это сделано специально, чтобы при выполнении согласованного многокоординатного движения можно было запрограммировать следующий сдвиг для каждой оси во время выполнения текущего сдвига. Тогда не потребуется останавливать движение осей каждый сдвиг.

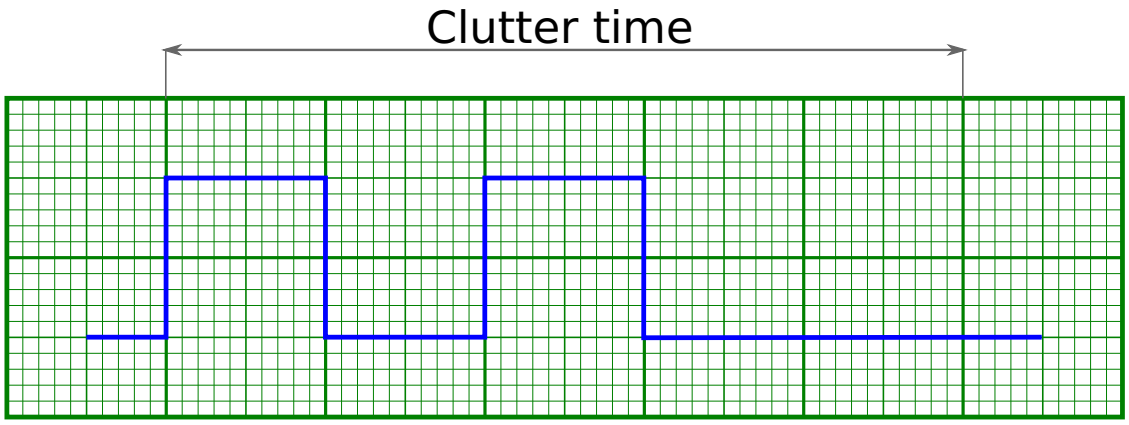
Предупреждение. Если при включении контроллера или его перезагрузке на входе синхронизации присутствует уровень напряжения, который считается активным, то контроллер воспримет это как сигнал для инициирования движения аналогичного выполнению команды Смещение на заданное расстояние.

Замечание. Position и *Speed* - отдельные переменные, которые могут быть сохранены в энергонезавиимой памяти контроллера, используются только при работе синхровхода.

Замечание. Движение по входному импульсу синхронизации подчиняется настройкам ускорения, максимальной скорости, и другим подсистемам, связанным с движением. Неправильная их настройка может мешать согласованному синхронному движению в многоосной системе.

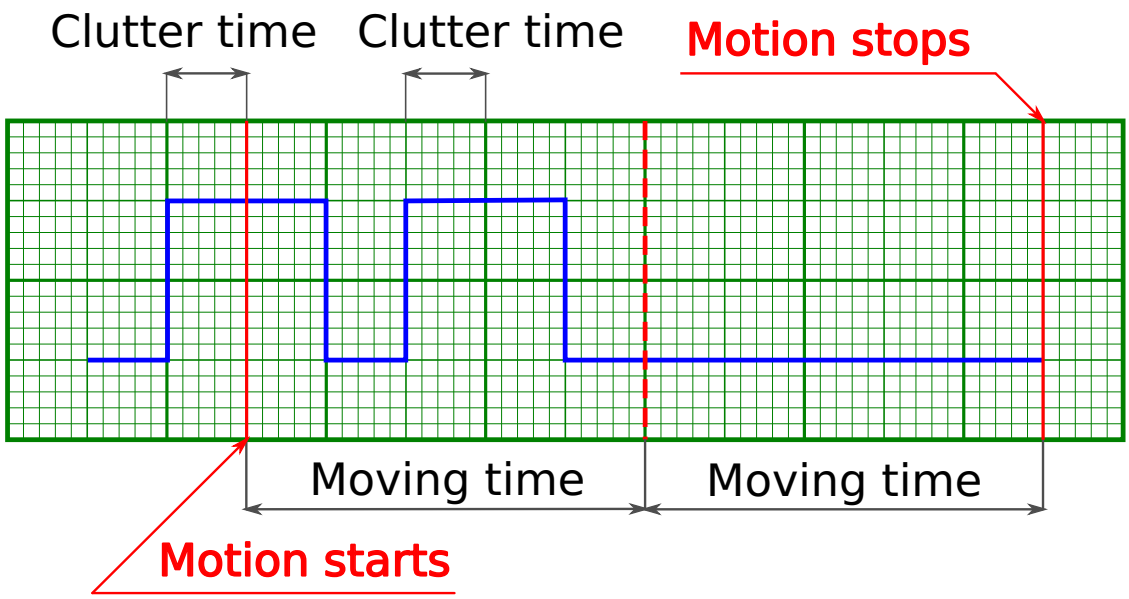


Движение начнётся раз входной импульс дольше времени подавления дребезга

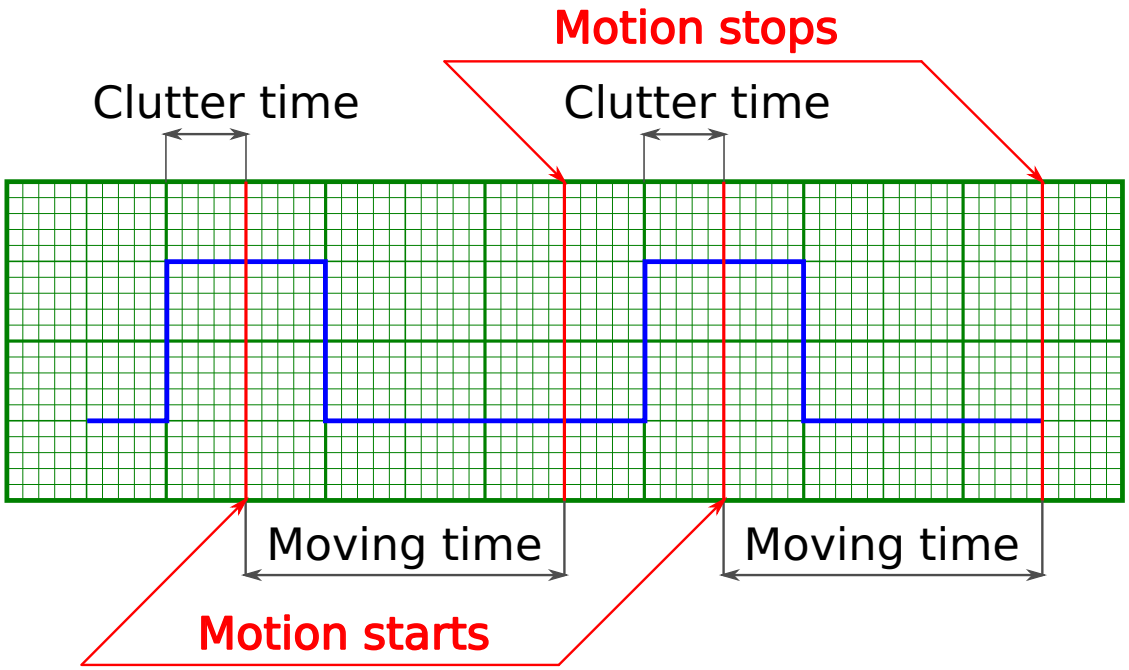


Движение не начнётся так как входные импульсы короче времени подавления дребезга

Предупреждение. Если во время исполнения сдвига пришел еще один входной синхроимпульс, то смещение будет произведено на удвоенное значение, если два - на утроенное и т.д.



Движение произойдёт один раз на двойное расстояние так как второй импльс синхронизации сработал до окончания первого движения



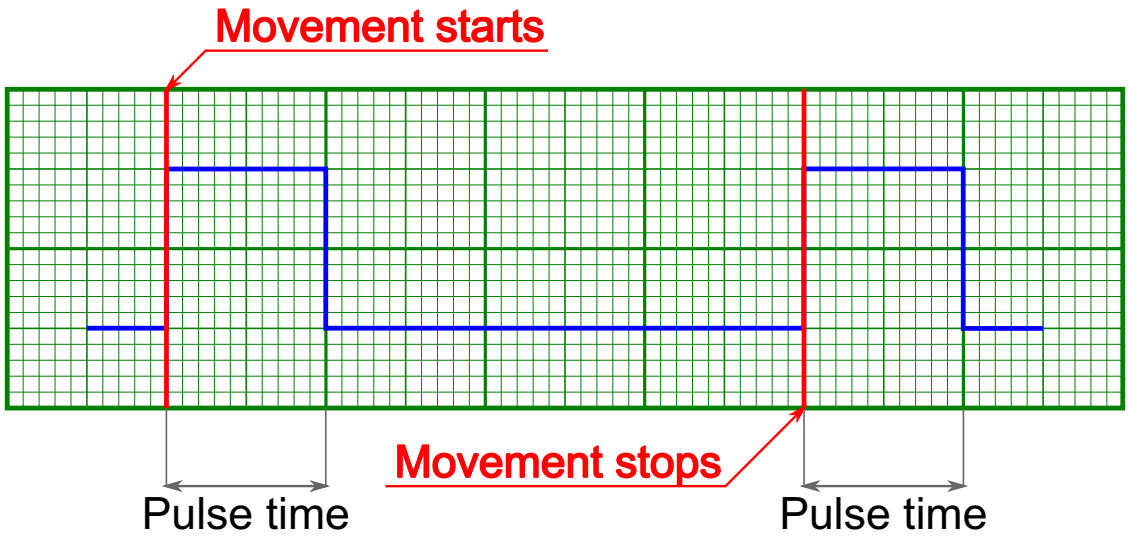
Движение произойдёт два раза с двумя стартами и двумя остановками

По умолчанию активным состоянием считается единичное состояние, а сигнал начала движения это нарастающий фронт. Вход синхронизации может быть инвертирован. При этом активным будет считаться нулевое состояние, а сигналом начала движения спадающий фронт.

> *Замечание.* Инвертирование входа синхронизации приводит к изменению понятий активного и неактивного состояний, проявляющейся, например, в статусе контроллера. Однако программное инвертирование само по себе не может являться сигналом начала движения, даже если при этом произошёл переход в активное состояние.

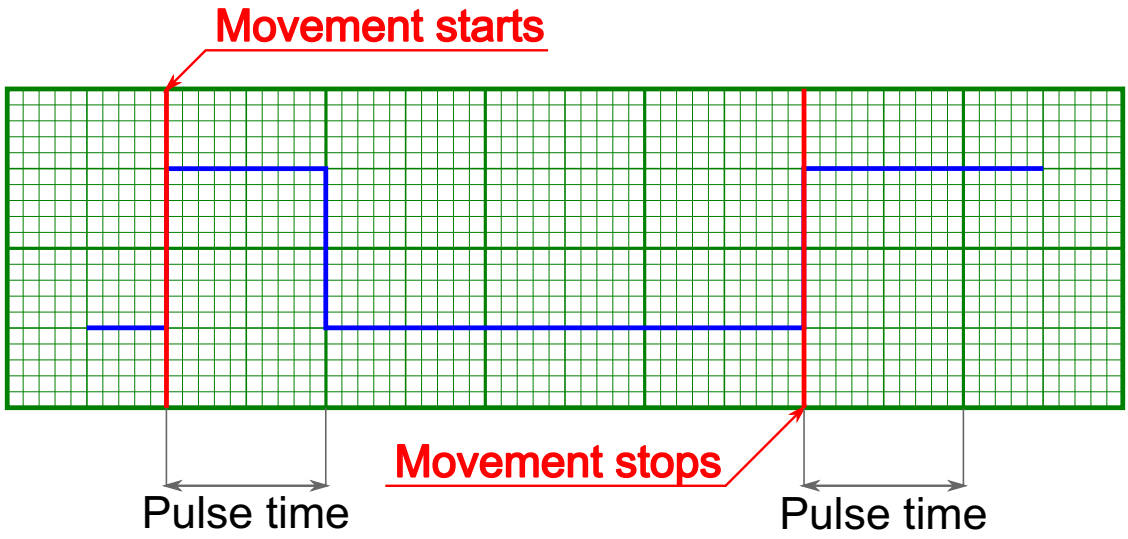
Выход синхронизации

Выходная синхронизация используется для управления внешними устройствами, привязанными к определенным событиям движения. Выходной синхроимпульс может подаваться при начале движения и/или при завершении движения, и/или каждый раз при смещении позиционера на заданное расстояние. Настройка *ImpulseTime* определяет длительность импульса синхронизации (может быть указана в микросекундах или смещении). Выход синхронизации может быть переведен в режим управляемого цифрового выхода. В этом режиме программным способом можно устанавливать единичный или нулевой логический уровень на выводе.



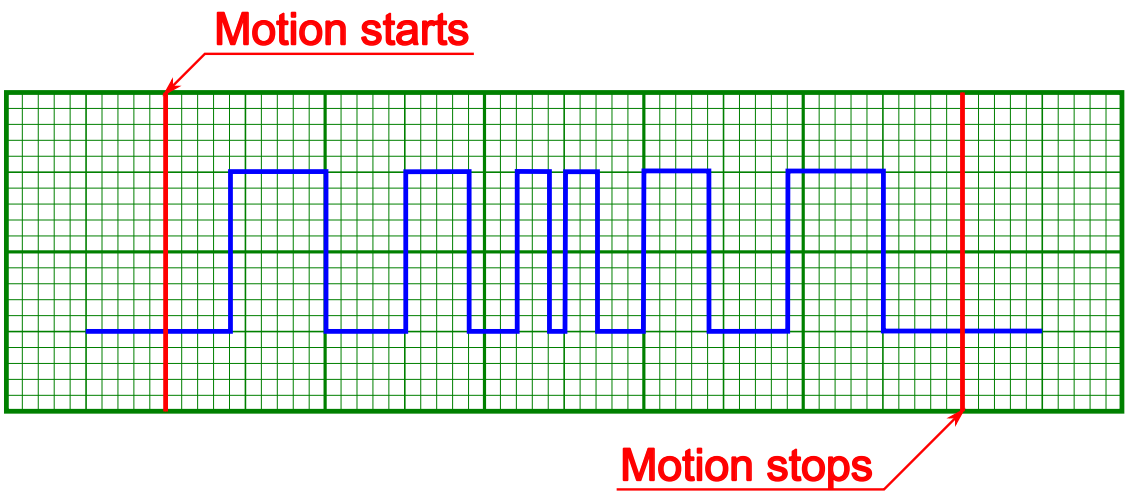
Выходные импульсы синхронизации при генерации их на старте и на остановке движения (импульс фиксированной длительности)

> *Замечание.* Если длительность синхроимпульса выражена в единицах смещения, например 10 шагов шагового двигателя, и поставлен режим "подавать синхроимпульс при завершении движения", то логический уровень на выходе синхронизации будет подан по завершению движения, но снят будет только после 10-ти шагов следующего движения.



Выходные импульсы синхронизации при генерации их на старте и на остановке движения (импульс измеряется в единицах смещения)

> *Замечание.* Если вы хотите перенастроить синхровыход и не уверены, что знаете в каком состоянии он находится, переведите его в режим выхода общего назначения и установите желаемый логический уровень.



Выходные импульсы синхронизации при движении с ускорением и генерации импульсов каждое смещение на заданное расстояние (импульс измеряется в единицах смещения)

Выходные импульсы синхронизации при движении с ускорением и генерации импульсов каждое смещение на заданное расстояние (импульс измеряется в микросекундах)

> *Замечание.* Периодическая генерация импульсов работает, как имитатор датчика полного оборота с передаточным числом. Координаты, в которых происходит генерация импульсов отсчитываются от нуля координат, а не от координаты в момент начала движения. Например, если в настройках включена генерация импульсов каждые 1000 шагов, то импульсы будут генерироваться при переходе через точки 0, 1000, 2000, 3000, и т.д. Генерация импульсов происходит при движении в обоих направлениях. Импульс генерируется в момент, когда частное от деления текущей координаты на период изменяется на единицу. Т.е. генерация импульсов при достижении координаты 1000 при движении от меньшей координаты к большей и при покидании координаты 1000 при движении от большей координаты к меньшей. Так же импульс всегда генерируется при переходе в точку 0 из какой-либо другой координаты (в том числе при обнулении координаты кнопкой ZERO).

> *Замечание.* В случае если импульсы на выходе синхронизации накладываются друг на друга, то они сливаются в один импульс.

Иллюстрация наложения импульсов синхронизации, при движении с ускорением и генерации импульсов каждое смещение на заданное расстояние, по старту и по остановке движения (импульс измеряется в микросекундах)

Настройка параметров синхронизации в **XILab** описана в разделе "Настройки синхронизации".

Схема подключения

Плата контроллера

В плате контроллера предусмотрены два ТТЛ-канала синхронизации на разъеме BPC.

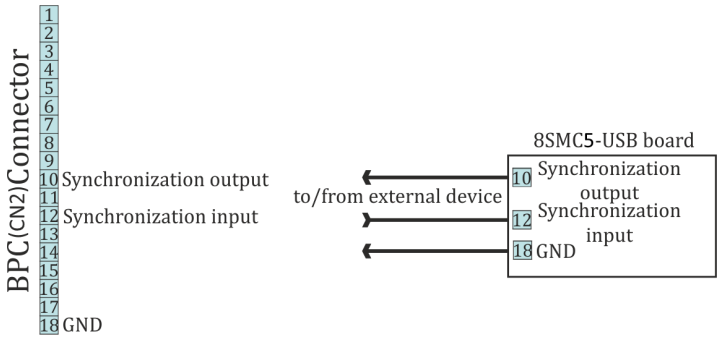


Схема подключения к каналам синхронизации на плате контроллера

Одноосная и двухосная система

Сигналы синхронизации на одноосной и двухосной системах выведены на разъём HDB26.

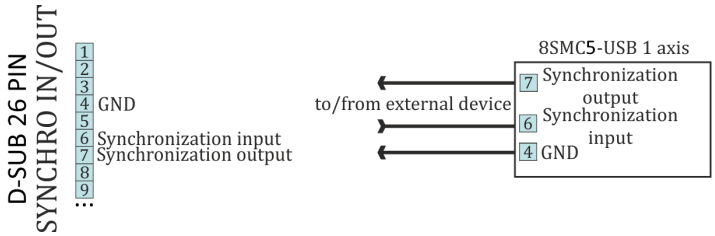


Схема подключения к каналам синхронизации на одноосной системе

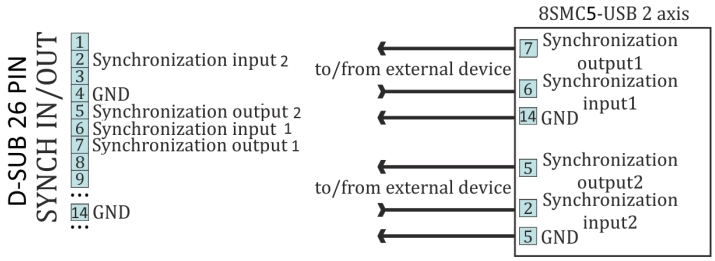


Схема подключения к каналам синхронизации на двухосной системе

4.5.6. Создание многоосных систем

Идентификация осей в составе многоосных систем осуществляется по серийному номеру контроллера. Каждый контроллер имеет свой уникальный серийный номер, который отображается в ПО XILab во вкладке О контроллере. Получение серийного номера контроллера возможно с помощью функции get_serial_number см. Руководство по программированию.

Многоосные системы на базе данного контроллера строятся с использованием активной соединительной платы на основе USB-хаба или внешнего USB-хаба.

Для правильного функционирования многоосных конфигураций плат необходимо первым действием подключить все контроллеры многоосной конфигурации друг к другу шинами питания и USB сигналов (установка контроллеров на соединительную плату).
Далее в любом порядке выполнить все следующие действия:

- Подключение электропитания к соединительной плате.
- Подключение внешних устройств.
- Подключение управляющего контроллера по USB.

Схема подключения

Схема подключения плат к многоплатной конфигурации подразумевает разводку питания (VBUS, GND) к ножкам 1 и 2 разъёма BPC, а также подвод от USB хаба независимых каналов данных D-/D+ к ножкам 15 и 17 разъёма BPC соответственно. Это делается для каждой платы. Далее подключается силовое питание:

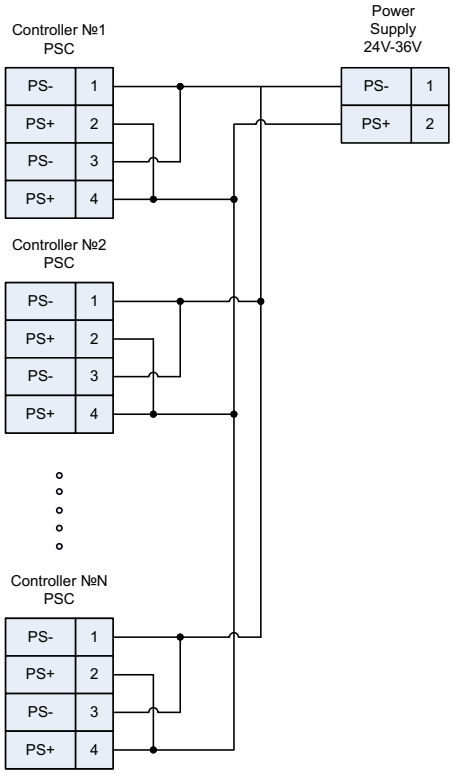


Схема подключения силового питания контроллеров в многоосной конфигурации

PSC - Power Supply Connector, разъем для силового питания контроллера
BPC - Back Panel Connector, разъем для подключения дополнительных устройств к контроллеру

4.5.7. Цифровой вход-выход общего назначения

Цифровой вход-выход общего назначения позволяет пользователю сконфигурировать его как вход или выход. По умолчанию активным считается единичный логический уровень (см. таблицу 4.5.7.1). Однако его можно инвертировать так, что активным будет считаться нулевой логический уровень.

Тип	**ТТЛ уровень**
Логический нуль (не активно)	0 В
Логическая единица (активно)	3.3 В

Таблица 4.5.7.1 – Параметры вывода

В режиме входа можно либо просто получать информацию о логическом уровне на линии (см. Статус контроллера), либо инициировать следующие действия при переходе в активное состояние:

- Выполнить команду *STOP* (быстрой остановки).
- Выполнить команду *PWOF* (отключение питания обмоток).
- Выполнить команду *MOVR* (смещение на заданное расстояние с последними использованными настройками).
- Выполнить команду *HOME* (автоматическая калибровка позиции).
- Войти в состояние *ALARM* (отключение силовых мостов и ожидание переинициализации).

Не имеет значения каким образом состояние входа становится активным (после изменения настройки инвертирования состояний или же при смене уровня напряжения). Контроллер использует программное подавление дребезга контакта на входе. Инициирование действия происходит, только если активное состояние на входе кнопки длилось более 3-х миллисекунд.

Предупреждение. Если при включении контроллера или его перезагрузке на входе присутствует уровень напряжения, который считается активным, то контроллер воспримет это как сигнал для инициирования какого-либо из действий.

Замечание. Цифровой вход имеет слабую подтяжку к земле.

В режиме вывода можно устанавливать активный или неактивный логический уровень или состояния на выбор:

- EXTIO_SETUP_MODE_OUT_MOVING - Активное состояние пока мотор находится в движении.
- EXTIO_SETUP_MODE_OUT_ALARM - Активное состояние пока мотор находится в состоянии Alarm.
- EXTIO_SETUP_MODE_OUT_MOTOR_ON - Активное состояние пока питание подано на обмотки мотора.
- EXTIO_SETUP_MODE_OUT_MOTOR_FOUND - Активное состояние пока мотор подключен.

Тип логики	ТТЛ 3.3 В
Частота обновления	1 кГц
Номинальный ток	5 мА

Таблица 4.5.7.2 – Технические характеристики вывода.

Схема подключения

Плата контроллера

Вывод расположен на разъеме BPC.

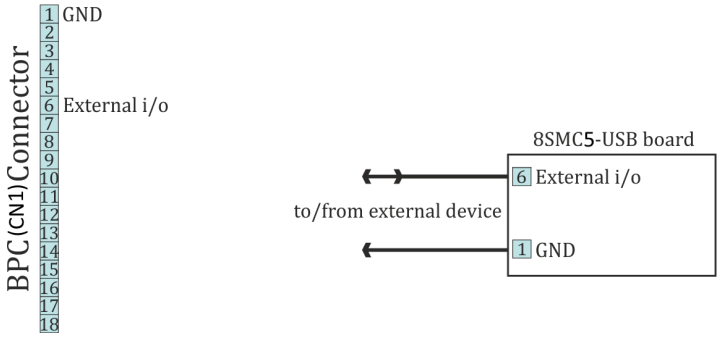


Схема подключения к цифровому входу выходу на плате контроллера

Одноосная и двухосная системы

Среди двух коробочных версий, цифровой вход-выход есть только у одной из них - двухосной. Соответствующие контакты для каждой оси выведены на HDB-26 разъём.

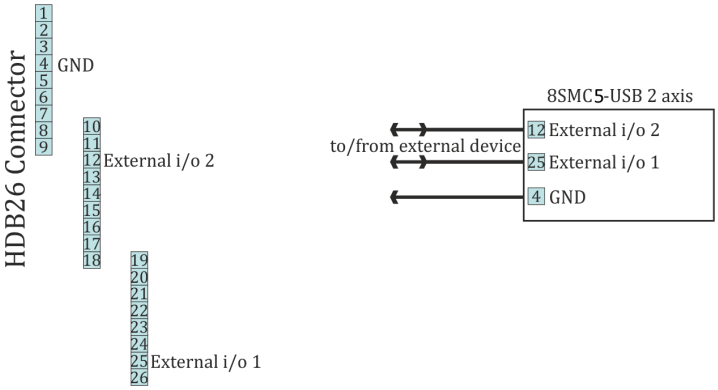


Схема подключения к цифровому входу-выходу в двухосной системе

4.5.8. Аналоговый вход общего назначения

Аналоговый вход общего назначения можно использовать для собственных нужд. Например, для измерения каких-либо внешних сигналов. Полученное значение с аналогового входа можно считать командой *GETC* или посмотреть в графиках XiLab.

Для данного контроллера диапазон аналогового входа от 0 до 10000 условных отсчетов.

ВАЖНО. Напряжение снимаемое с аналогового входа не должно выходить за пределы от 0 до 3 В. При превышении напряжения питания возможны ошибки в работе аналогового входа и работе других систем контроллера! Это может привести к выходу из строя как контроллера, так и подключенного к нему мотора.

Напряжение сигнала	0-3 В
Частота считывания	1 кГц

Таблица. Параметры входа.

Схема подключения

Плата контроллера

Контакт аналогового входа на плате контроллера расположен на разъеме BPC.

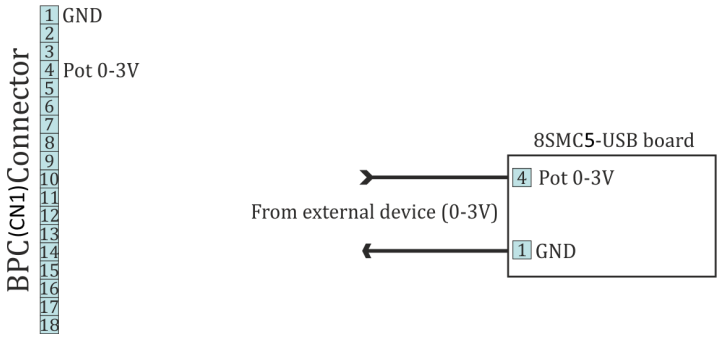


Схема подключения к аналоговому выводу на плате контроллера

Одноосная и двухосная система

Аналоговый вход есть только в двухосной системе и расположен на HDB-26 разъёме.

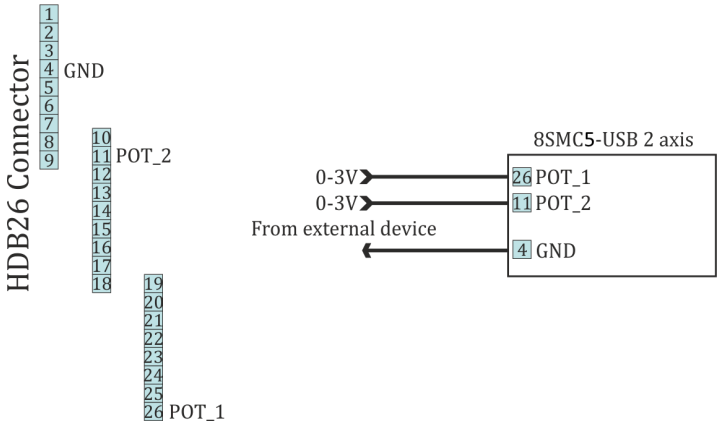


Схема подключения к аналоговому выводу в двухосной системе

4.5.9. Интерфейс управления внешним драйвером

Интерфейс позволяет управлять любым внешним драйвером с помощью трех стандартных сигналов: enable, direction, clock. Этот режим удобен, когда силового функционала контроллера недостаточно и, при этом, хочется иметь такие его возможности, как концевики, датчик оборотов, контроль позиции, управление скриптами, создание многоосных систем, управление джойстиком или кнопками, использование магнитного тормоза, и т. п. Например, для создания многоосной системы с одной мощной подъёмной осью, которая управляется внешним контроллером, и двумя менее мощными горизонтальными осями, можно использовать XiLab с трехосным интерфейсом, скриптами, а также синхронизировать движение всех трёх осей. То есть использование внешнего драйвера заменяет только силовую часть контроллера.

Сигнал Clock определяет количество сдвигов в направлении Direction (логическая единица - вправо, логический ноль - влево). Сдвиг это минимальный шаг, при текущих настройках деления шага. Для деления шага 1/32 будет подано 32 импульса на один шаг. Не забудьте настроить внешний драйвер так, чтобы он использовал то же деление шага.

Предупреждение. Частота сигнала clock в данном контроллере ограничена 78 кГц. Поэтому для достижения необходимой скорости, требуется уменьшать деление шага. Например, если необходима скорость вращения 4000 шагов в секунду, то необходимо использовать деление шага не точнее 1/8.

Тип	ТТЛ
Уровень логического нуля	0 В
Уровень логической единицы	3.3 В

Параметры выходов управления внешним драйвером

Схема подключения

Предупреждение. Выводы для управления внешним драйвером малозащищены и могут быть повреждены при неправильном использовании. Обеспечение правильного подключения и необходимых электрических защит лежит на инженере, конструирующем систему связи драйверов.

Плата контроллера

Для подключения внешнего драйвера используется три вывода на разъеме BPC.

Предупреждение. Вывод 13 является входом-выходом общего назначения (см. Цифровой вход-выход общего назначения), но он теряет функциональность, когда включен режим управления внешним драйвером.

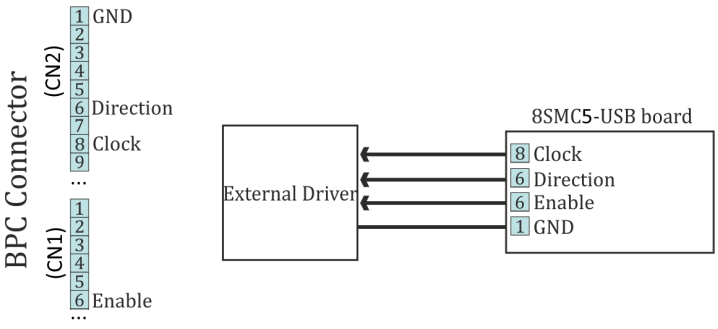
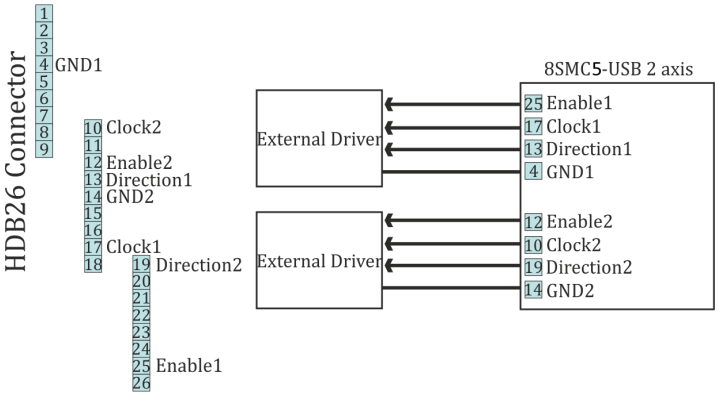


Схема подключения к внешнему драйверу

Одноосная и двухосная система

Интерфейс управления внешним драйвером есть только в двухосной системе и выведен на разъём HDB-26.

Предупреждение. Выводы 12 и 25 являются входами-выходами общего назначения (см. Цифровой вход-выход общего назначения), но они теряет свою функциональность, когда включен режим управления внешним драйвером.


HDB26 Connector
1
2
3
4 GND1
5
6
7
8
9
10 Clock2
11
12 Enable2
13 Direction1
14 GND2
15
16
17 Clock1
18
19 Direction2
20
21
22
23
24
25 Enable1
26
External Driver
External Driver
8SMC5-USB 2 axis
25 Enable1
17 Clock1
13 Direction1
4 GND1
12 Enable2
10 Clock2
19 Direction2
14 GND2

4.5.10. Последовательный порт

Контроллер допускает управление по последовательному порту UART с логикой TTL 3.3 В. За счёт высокой распространённости UART и переходников на него с USB, Bluetooth, Ethernet и других стандартных интерфейсов, контроллером можно управлять беспроводным образом (Bluetooth) или по интернету (Ethernet). Протокол передачи данных по UART такой же как и по USB. То есть потребуется только включить в XiLab или в libximc опрос нестандартных последовательных портов и устройство обнаружится если задержка ответа с него не превышает 2 секунды. Можно также управлять контроллером с помощью другого самостоятельно запрограммированного микроконтроллера, хотя это потребует поддержки протокола общения.

UART поддерживает следующие настройки:

Скорость передачи	9600-921600 бит/с
Количество бит в передаче	8
Четность	включена или выключена
Тип четности	четное, нечетное, единица, ноль - на выбор пользователя
Количество стоповых бит	1 или 2

Замечание. Чтобы установить связь с контроллером по UART нужно предварительно соединиться с ним по USB, установить необходимые настройки скорости, четности, количества стоповых бит, а затем сохранить их в энергонезависимую память контроллера. Стандартными для прилагаемого программного обеспечения являются настройки, описанные в протоколе общения. Если связь не устанавливается, то попробуйте использовать их.

Тип логики	ТТЛ 3.3 В
Максимальная скорость передачи данных	921 кбит
Номинальный ток выхода	5 мА

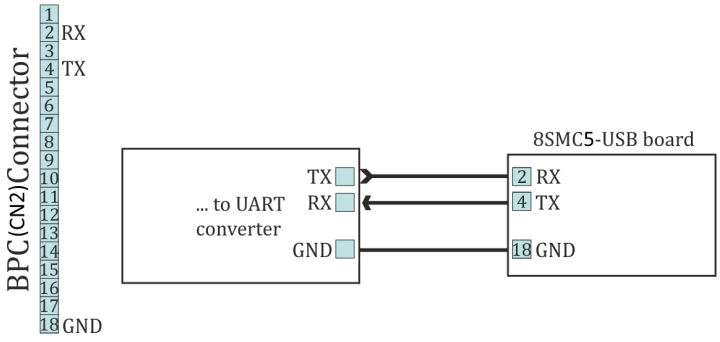
Параметры входа и выхода

Схема подключения

Предупреждение. Высокая скорость передачи данных невозможна по длинному кабелю в условиях электромагнитных наводок. Если случаются ошибки в передаче данных, то используйте фильтрующие RC цепи и снизьте скорость, чтобы характерное время цепи было как минимум в 4 раза меньше времени передачи одного бита. Характерное время RC цепи подбирается индивидуально.

Плата контроллера

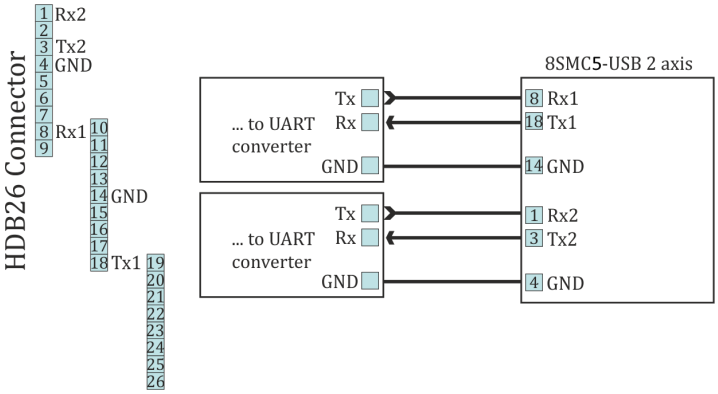
В плате контроллера выводы UART расположены на разъеме BPC.



Рекомендуемая схема подключения для платы контроллера

Одноосная и двухосная система

UART есть у двухосной и одноосной системы. Соответствующие контакты для каждой оси выведены на HDB-26 разъём.



Рекомендуемая схема подключения для двухосной системы

4.4. Хранение позиции во FRAM-памяти контроллера

Контроллер имеет функцию автоматического запоминания позиции, которая позволяет просто отключать ему питание после остановки и при следующем включении питания, контроллер будет удерживать то же положение двигателя, значение положения и инкрементального счётчика энкодера. Эта функция работает, если в момент, когда контроллер обесточен, не происходит вращения оси мотора.

> *Замечание.* Для работы этой функции необходимо подождать хотя бы 0.5 секунды после остановки вращения до выключения питания контроллера по USB. Обесточивание контроллера во время вращения также приведёт к сохранению позиции, но она будет лишь примерной и потребуется новая калибровка

4.5.12. Опознавание позиционеров Standa

В новейших позиционерах фирмы Standa (список конкретных устройств уточняйте у производителя) предусмотрено хранение настроечных и информационных параметров позиционера во встроенной в подвижку памяти. Эта память заранее прошита при изготовлении подвижки верными значениями, что позволяет не заботиться об оптимальных для конкретного позиционера настройках и начать работу сразу после сборки системы. В памяти есть также и пользовательское поле имени подвижки, которое пользователь может установить сам (см. вкладку XiLab Название позиционера).

При подключении такого позиционера к контроллеру (более подробно об электрической схеме подключения написано в разделе Пример подключения простого мотора 8SMC5 и Разъём подключения позиционера) осуществляется автоматическая загрузка информационных параметров подвижки, см. раздел Характеристики позиционера, в память контроллера. Если был установлен флаг EEPROM_PRECEDENCE (преимущество настроек считанных из памяти подвижки перед настройками во flash-памяти контроллера, см. О контроллере), то дополнительно считываются и устанавливаются все настройки системы, кроме настроек UART и имени контроллера.

При установленном флаге EEPROM_PRECEDENCE не нужно проверять и/или устанавливать настройки позиционера (к примеру, полярность и расположение концевиков, рабочий ток, параметры энкодера и магнитного тормоза). Все это будет сделано автоматически при подключении позиционера, оборудованного встроенной памятью. Однако при установленном флаге EEPROM_PRECEDENCE загрузка настроек из памяти позиционера будет происходить каждый раз при подключении подвижки, оборудованной памятью, либо при включении питания контроллера, к которому подключена такая подвижка. Поэтому если требуется изменить какие-то настройки, то необходимо снять данный флаг, изменить необходимые настройки, сохранить их во внутреннюю память контроллера.

> *Замечание.* Простое правило для использования этого флага: На ранних этапах работы этот флаг должен быть включен для удобства автоматических настроек. Со временем, когда захочется самостоятельно менять настройки, этот флаг стоит выключить, не забыв сохранить настройки во FRAM.

> *Замечание.* При отключении позиционера оборудованного памятью не происходит никаких изменений настроек.

Для разработчиков

Данные подвижки хранятся в микросхеме DS28EC20, соединяемой по интерфейсу 1-wire.

При опознавании подвижки на микросхему EEPROM периодически посылается сигнал сброса. При получении от микросхемы ответного сигнала, контроллер считывает все данные подвижки в собственную оперативную память, автоматически настраивается на подвижку и выставляет бит STATE_EEPROM_CONNECTED в статусной структуре. В XILab это отображается индикатором EEPR в главном окне. Далее проверка наличия памяти производится регулярно. В случае потери связи с EEPROM (отсутствию ответа на сигнал сброса) индикатор EEPR в XiLab гаснет.

Схема подключения

Выводы для соединения с микросхемой на всех системах (плата контроллера, одноосная и двухосная в корпусе расположены на разъёме D-SUB

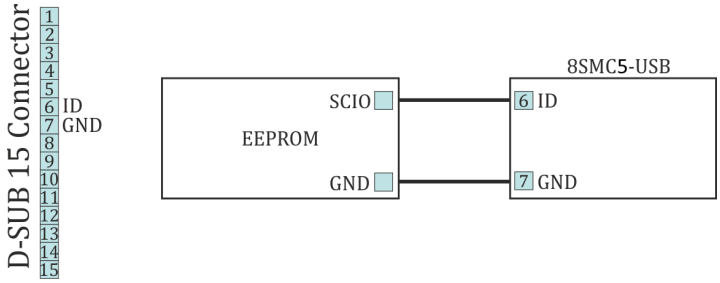


Схема соединения для тестирования внешней памяти.

4. Второстепенные функции

1. Установка нулевой позиции
2. Установка пользовательской позиции
3. Статус контроллера
4. Автовосстановление USB соединения

4.1. Установка нулевой позиции

Контроллер поддерживает установку нулевой позиции. Эту функцию стоит использовать для позиционеров маркированных репером, чтобы позиция по реперу соответствовала программной. Также эта функция удобна, когда существует одна избранная физическая позиция на диапазоне перемещения.

Для установки нулевой позиции используется специальная команда. При этом обнуляются счётчики шагов, микрошагов, энкодера. Установка нулевой позиции происходит одновременно для всех счётчиков позиции и не может привести к их рассогласованию. Влияние на текущую команду движения не оказывается. Если контроллер отрабатывал движение в некоторую физическую позицию в момент, когда текущая позиция была обнулена, то движение завершится в прежней физической позиции. Например, при движении к позиции 1000 в момент прохода 200 была послана команда обнуления позиции. Тогда счётчик позиции уменьшится на 200 и движение завершится в координате 800.

> *Замечание.* Установка нулевой позиции при работе в режиме смещения (см. Смещение на заданное расстояние) не изменит физической позиции, к которой осуществлялось движение. Следующее смещение будет происходить к той же физической позиции, что и без использования установки нулевой позиции.

4.2. Установка пользовательской позиции

Если необходимо установить позицию или счётчик энкодера не в ноль, а в некоторую пользовательскую позицию, то для этого существует команда *SPOS*. В этой команде передаются значения новых счётчиков позиции, микрошагов для шаговых двигателей и счётчика энкодера, если он является второстепенным датчиком положения. Если необходимо установить лишь одну из позиций, нужно воспользоваться флагами игнорирования части полей команды.

Отличие этой команды от обнуления позиции, в том, что не происходит обнуление последней позиции, к которой происходило смещение, нет отличия в поведении команды в момент движения и при остановке. Если команду применить в момент движения к позиции, то движение завершится в той же физической точке, в которой оно завершилось бы и без смены позиции командой *SPOS*.

4.3. Статус контроллера

Контроллер отслеживает свой статус и способен передать его в статусной структуре команды GETS. Статус контроллера включает в себя информацию о совершаемом движении, его результате, состоянии питания, энкодера, обмоток двигателя, цифровых входов и выходов, числовую информацию о позиции и питающем напряжении и токах, а также флаги ошибок.

Статус движения

MoveSts содержит:

- Флаг движения, который устанавливается когда контроллер меняет позицию двигателя.
- Флаг достижения требуемой скорости, который устанавливается если скорость равна той, с которой контроллер должен отрабатывать текущее движение.
- Флаг антилюфта, который устанавливается при подавлении люфта во время заключительной стадии движения (см. Компенсация люфта).

MvCmdSts содержит информацию о выполняемой команде. Все движения мотора вызываются командами движения к целевой позиции *MOVE*, сдвига относительно последней целевой позиции *MOVR*, движения вправо *RIGT* или влево *LEFT*, плавной *SSTP* или резкой *STOP* остановки, калибровки домашней позиции *HOME* или принудительного подавления люфта *LOFT*. Управление кнопками, джойстиком, импульсами синхронизации и т. п. тоже приводится к этим командам. Например, джойстик вызывает команды движения вправо и влево при отклонении или команду плавной остановки в центральном положении (см. Управление с помощью джойстика). В переменной MvCmdSts приведена текущая команда движения или последняя выполненная команда, а также статус команды: выпоняется она или уже выполнена. Если команда выполнена, то еще один бит показывает результат её выполнения (успешный или неуспешный). Неуспешное выполнение означает, что мы не попали к той позиции, к которой пытались двигаться или не смогли отработать люфт. Причиной может быть неожиданная остановка по концевикам, попадание в состояние Alarm. Изначальное состояние этого поля показывает неизвестную команду и статус успешного выполнения.

Статус питания мотора

PWRSts содержит информацию о питающем напряжении. Обмотки могут быть:

- Отключены (в этом случае на них не подаётся никакого напряжения).
- Запитаны сниженным током относительно номинального тока (например при использовании функции снижения тока в обмотках при остановке).
- Запитаны номинальным током.
- Запитаны недостаточным напряжением, чтобы обеспечить установленный номинальный ток.

Последний статус часто появляется при высоких скоростях вращения, ведь чем выше скорость переключения шагов, тем выше должно быть питающее напряжение, чтобы обеспечить нарастание тока в индуктивности обмоток двигетеля. Недостаточное питание не означает, что двигатель не будет вращаться, а означает, что двигатель может больше шуметь и падает крутящий момент. (См. Управление питанием мотора).

Статус энкодера

EncSts содержит информацию о подключенном энкодере если включен режим управления без обратной связи (например для шаговых двигателей). Состояния энкодера могут быть:

- Не подключен.
- Неизвестное состояние, когда недостаточно данных чтобы определить состояние энкодера.
- Подключен и исправен.
- Подключен и реверсирован (тогда нужно включить в настройках реверс энкодера).
- Подключен и неисправен.

Последнее состояние реализуется, когда на входы энкодера поступают сигналы переключения, но они не соответствуют движению ротора двигателя. Смена состояний проходит после набора достаточной статистики. Поэтому обнаружение происходит немгновенно. Также невозможно точно определить статус энкодера без движения. (См. Работа с энкодерами).

Статус обмоток двигателя

WindSts содержит информацию о состоянии обмоток. Показывается состояние для каждой из двух обмоток отдельно. Они могут быть:

- Отключены от контроллера.
- Быть подключены.
- Замкнуты накоротко.
- Их состояние можут быть неизвестно.

Замкнутым накоротко считается слишком маленькое сопротивление и индуктивность в обмотке. Отключенными обмотками считается нагрузка с слишком высоким сопротивлением.

Статус положения

В статусной структуре выводятся все данные о положении и скорости позиционера. Для этого используются поля основной позиции (CurPosition, uStep), второстепенной позиции (EncPosition), скорости (CurSpeed, uCurSpeed). Основное положение отсчитывается в шагах шагового двигателя и микрошагах, если используется управление без обратной связи. При использовании режима ведущего энкодера в CurPosition хранится счётчик положения по энкодеру а в uStep записан 0. Поле второстепенной позиции это поле энкодера если используется управление шаговым двигателем без обратной связи, счетчик шагов, если подключен шаговый двигатель в режиме ведущего энкодера, или 0, если подключен DC двигатель. Скорость выводится всегда для основного датчика позиции и измеряется в тех же единицах, что и установленная скорость движения.

Статус питания контроллера и температура

В статусной структуре выводятся:

- Ток потребления силовой части (в мА).
- Напряжение на силовой части (в десятках мВ).
- Ток потребления по USB (в мА).
- Напряжение на USB (в десятках мВ).
- Температура микропроцессора (в десятых долях градуса Цельсия).

Статусные флаги

Флаги делятся на ошибки команд управления, ошибки превышения критических параметров, общие ошибки и флаги состояния. Многие флаги не снимаются сами, пока их принудительно не снять командой *STOP*.

Ошибки команд протокола:

- errc - Неопознанная команда протокола. Ошибка возникать не должна если используется софт, совместимый с используемой в контроллере версией протокола. Флаг не снимается самостоятельно.
- errd - Код проверки целостности данных команды не сошёлся. Ошибка возникает при сбоях передачи данных. Флаг не снимается самостоятельно.
- errv - Не удалось применить одно или несколько переданных в команде значений. Возникает когда команда была принята и успешно распознана, но передаваемые в ней данные были некорректны, выходили за допустимый диапазон. Также эта ошибка может означать, что требуюмую операцию не удалось выполнить из-за аппаратного сбоя. Например, эта ошибка возникнет при установке режима деления шага, не входящего в список поддерживаемых или при установке нулевого количества шагов на оборот двигателя. Флаг не снимается самостоятельно.

Ошибки превышения критических параметров:

- Флаг, что сейчас контроллер находится в режиме Alarm.
- Флаг, что сейчас силовой драйвер сигнализирует о перегреве. Флаг снимается сам в зависимости от настроек критических параметров.
- Флаг, что температура микропроцессора вышла за допустимый диапазон. Флаг снимается сам в зависимости от настроек критических параметров.
- Флаг, что напряжение питания превысило допустимое значение. Флаг снимается сам в зависимости от настроек критических параметров.
- Флаг, что напряжение питания оказалось ниже допустимого значения. Флаг снимается сам в зависимости от настроек критических параметров.
- Флаг, что потребляемый ток из блока питания превысил допустимое значение. Флаг снимается сам в зависимости от настроек критических параметров.
- Флаг, что напряжение USB превысило допустимое значение. Флаг снимается сам в зависимости от настроек критических параметров.
- Флаг, что напряжение USB оказалось ниже допустимого значения. Флаг снимается сам в зависимости от настроек критических параметров.
- Флаг, что ток потребления по шине питания USB превысил допустимое значение. Флаг снимается сам в зависимости от настроек критических параметров.
- Флаг, что концевики перепутаны местами. Флаг не снимается самостоятельно.

Общие флаги ошибок:

- Флаг, что система контроля позиции обнаружила рассогласование позиции по счётчику шагов и датчику положения. Флаг не снимается самостоятельно (если не используется автоматическая корректировка позиции).

Флаги состояния:

- Наличие подключенной сейчас подвижки, оснащённой памятью EEPROM.
- Наличие внешнего питания. Иначе питание внутреннее. Установлен всегда.

Статус цифровых сигналов

Контроллер выводит состояние входных и выходных цифровых сигналов в виде флагов активного состояния или в виде текущего логического уровня. Активное состояние соответствует единице или нулю, в зависимости от настроек конкретного блока, например от настройки инвертирования. Флаги бывают:

- Состояние правого концевика (1 если концевик активен).
- Состояние левого концевика (1 если концевик активен).
- Состояние правой кнопки (1 если кнопка нажата).
- Состояние левой кнопки (1 если кнопка нажата).
- 1 если ножка EXTIO работает как выход. Иначе - как вход.
- Состояние ножки EXTIO (1 если активное состояние на входе или на выходе).
- Состояние датчика Холла A (1 если на входе логическая единица).
- Состояние датчика Холла B (1 если на входе логическая единица).
- Состояние датчика Холла C (1 если на входе логическая единица).
- Состояние магнитного тормоза (1 если на тормоз подано питание).
- Состояние датчика полного оборота (1 если датчик активен).
- Состояние входной ножки синхронизации (1 если ножка синхронизации в активном состоянии).
- Состояние выходной ножки синхронизации (1 если ножка синхронизации в активном состоянии).
- Состояние на входе канала энкодера A (1 если на входе логическая единица).

- Состояние на входе канала энкодера B (1 если на входе логическая единица).

4.6.4. Автовосстановление USB соединения

Данный блок предназначен для перезагрузки USB шины в случае потери связи (к примеру, это может возникнуть в случае удара статическим разрядом или отключения шины USB без отключения питания). Включение/выключение данного блока определяется флагом USB_BREAK_RECONNECT (см. Критические параметры). Если блок включен, то он отслеживает потерю связи по шине USB. В случае потери связи по шине USB через 500 мс выполняется программное переподключение к шине USB со стороны контроллера, после чего выполняется проверка состояния шины USB. Если в течение определенного времени не происходит восстановление связи (т.е. обмена даными), то выполняется повторное переподключение. Таким образом в случае не восстановления связи по USB, контроллер будет постоянно переподключаться к шине USB до тех пор, пока не произойдет восстановления связи по USB или время между переподключениями не превысит 1 минуты. В итоге в случае отключения шины USB без отключения питания контроллера (к примеру, в случае управления двигателем от джойстика или кнопок) в течение примерно 5 минут контроллер будет находиться в режиме переподключения шины USB.

> **Замечание.** Режим переподключения шины USB никаким образом не влияет на основные характеристики контроллера (к примеру, движение или удержание необходимого тока в обмотках)

Чтобы избежать синхронного переподключения к шине USB как со стороны контроллера, так и со стороны компьютера, время между переподключениями меняется по экспоненциальному закону (см. таблицу 1).

Номер перезагрузки	Время ожидания, мс
0(после потери связи)	500
1	483
2	622
3	802
4	1034
5	1333
6	1718

Визуально определить состояние блока перезагрузки USB можно по частоте мигания светодиода. В случае перехода в режим перезагрузки светодиод начнет мигать с частотой 10Гц (см. Индикация режима работы).

> **Предупреждение.** В силу особенностей строения данного программного блока, а также спецификации шины USB, блок не гарантирует 100% восстановление связи с компьютером после удара статическим разрядом.

Со стороны компьютера Xilab также производит попытки восстановления соединения с контроллером, если оно по каким-либо причинам было потеряно. При потере соединения, то есть если библиотека libximc вернула кода ошибки result_nodevice, сначала происходит ожидание 1000мс. Затем, если платформа на которой запущен Xilab принадлежит семейству Windows, средствами WINAPI опрашивается наличие устройства с соответствующим именем COM-порта в системе. Если такой порт присутствует, но библиотека libximc не может открыть его более двух раз, то вызывается функция ximc_fix_usbser_sys, которая производит ресет драйвера usbser.sys (исправление ошибки драйвера).
На платформе Linux или MacOS Xilab просто пытается повторно открыть устройство каждые 1000 мс. После успешного открытия в устройство посылаются команды чтения серийного номера, версии прошивки и некоторых настроек, необходимых для отображения интерфейса.
Библиотека libximc считает устройство потерянным (result_nodevice) когда системные функции ReadFile/WriteFile (на Windows) или read/write (на Linux/Mac) возвращают ошибку при чтении или записи данных в соответствующий USB-COM порт.

5. Руководство по программе XILab

1. О программе XILab
2. Основные окна программы XILab
    1. Стартовое окно программы XILab
    2. Главное окно программы XILab в режиме управления одной осью
    3. Главное окно программы XILab в режиме управления несколькими осями
    4. Настройки программы
    5. Графики
    6. Скрипты
    7. Лог XiLab
3. Настройки контроллера
    1. Настройка кинематики движения (Шаговый двигатель)
    2. Настройка диапазона движения и концевых выключателей
    3. Настройка предельных параметров контроллера
    4. Настройка параметров энергопотребления
    5. Настройка исходного положения
    6. Настройки синхронизации
    7. Настройка тормоза
    8. Контроль позиции
    9. Настройка внешних управляющих устройств
    10. Настройки UART
    11. Настройки цифрового входа-выхода общего назначения
    12. Настройка типа двигателя
    13. Настройка кинематики движения (DC мотор)
    14. Настройка контуров ПИД-регулирования
    15. О контроллере
4. Настройки программы XILab
    1. Общие настройки программы XILab
    2. Настройки интерфейса абстрактного позиционера
    3. Настройки интерфейса аттенюатора
    4. Настройка режима циклического движения
    5. Настройка логирования
    6. Общие настройки отображения графиков
    7. Индивидуальные настройки отображения графиков
    8. Настройки отображения пользовательских единиц
    9. О программе
5. Характеристики позиционера
    1. Название позиционера
    2. Общие характеристики позиционера
    3. Характеристики двигателя
    4. Характеристики энкодера
    5. Характеристики датчика Холла
    6. Характеристики редуктора
    7. Характеристики вспомогательных устройств
6. Корректное завершение работы
7. Работа с сетевыми контроллерами
8. Установка XILab
    1. Установка под Windows
        1. Установка под Windows XP
        2. Установка под Windows 7
        3. Установка под Windows 8
    2. Установка под Linux
    3. Установка под MacOS

5.1. О программе XILab

XILab представляет собой удобный графический интерфейс пользователя для управления позиционерами, диагностики моторов и настройки двигателей, управляемых данными контроллерами. XILab позволяет быстро настраиваться на подключенный позиционер с помощью загрузки подготовленных заранее конфигурационных файлов. Управление можно автоматизировать с помощью скриптового языка, что может использоваться непосредственно или ускорит процесс разработки собственной программы управления. XiLab поддерживает многоосевой режим и многомерные скрипты управления. Предусмотрена возможность выводить данные о состоянии контроллера и двигателя на графики и сохранять их в файл, экспортировать данные в табличном виде для обработки внешними программами. Программное обеспечение совместимо с операционными системами Windows XP SP3, Windows Vista, Windows 7, MacOS X и Linux. В зависимости от операционной системы вашего компьютера, вид некоторых окон может отличаться.

Краткое руководство по установке программы приведено здесь. В данной главе приведено подробное руководство по работе в ПО XILab.

5.2. Основные окна программы XILab

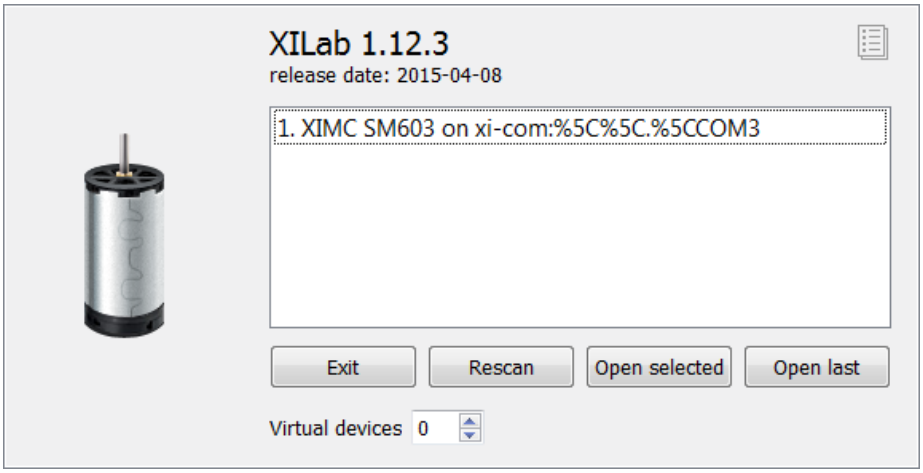
1. Стартовое окно программы XILab
2. Главное окно программы XILab в режиме управления одной осью
3. Главное окно программы XILab в режиме управления несколькими осями
4. Настройки программы
5. Графики
6. Скрипты
7. Лог XiLab

5.2.1. Стартовое окно программы XILab

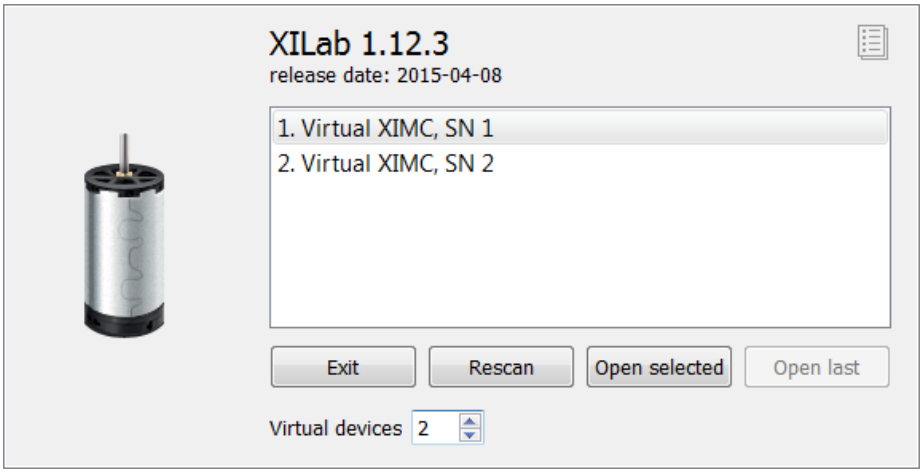
При запуске XILab открывается окно поиска контроллеров. XILab с помощью библиотеки libximc опрашивает контроллеры, подключенные к системе и выводит список найденных и успешно опознанных контроллеров на экран.



Стартовое окно XILab, найдено 0 контроллеров



Стартовое окно XILab, найден 1 контроллер



Стартовое окно XILab, найдено 2 виртуальных контроллера

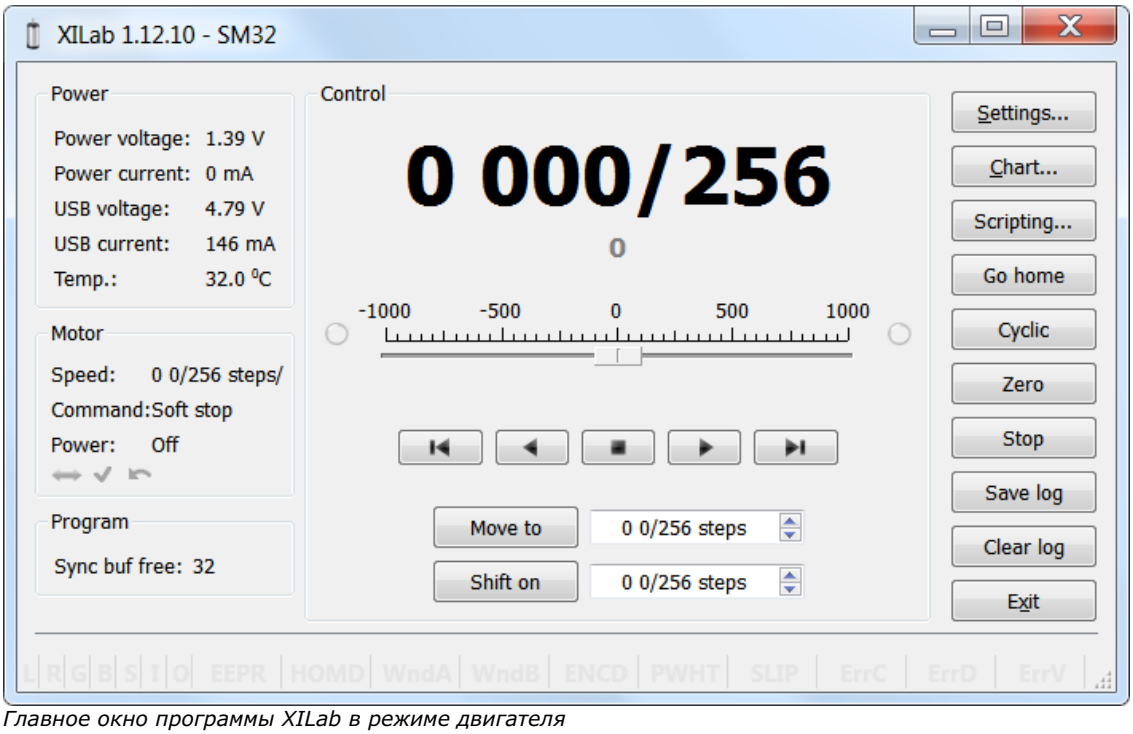
Список найденных контроллеров выводится в стартовое окно. Здесь можно выбрать один или несколько контроллеров и открыть их с помощью кнопки Open selected. Если выбран один контроллер, то будет открыто главное окно программы XILab в режиме управления одной осью, если выбрано более одного, то будет открыто главное окно программы XILab в режиме управления несколькими осями. Повторный поиск осуществляется нажатием Rescan, а выход - нажатием Exit. Если активна кнопка Open last, то это означает что найдены все контроллеры, которые были открыты при предыдущем запуске XILab, и нажатие Open last откроет эту сохраненную конфигурацию.

XILab может работать с виртуальными контроллерами XIMC, которые поддерживают протокол ответов реального контроллера. Виртуальный контроллер может быть полезен для ознакомления с интерфейсом XILab, в случае если к системе не подключены реальные контроллеры. В поле Virtual devices указано количество виртуальных контроллеров, которые будут выведены в список доступных для открытия при следующем нажатии Rescan или следующем старте XILab.

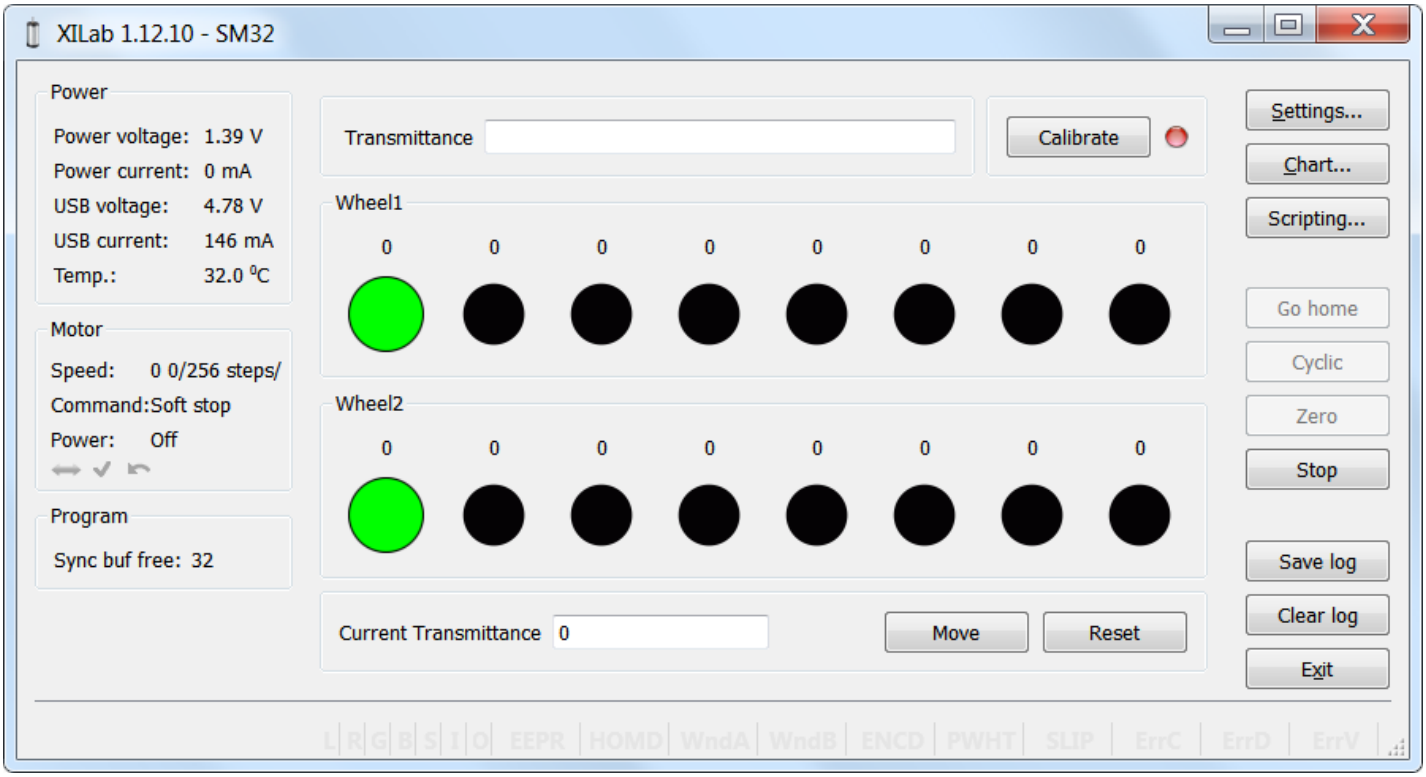
Замечание: Так как библиотека libximc открывает устройства XIMC в режиме эксклюзивного доступа, при запуске последующих копий программы XILab будут найдены и доступны для выбора только свободные контроллеры.

5.2.2. Главное окно программы XILab в режиме управления одной осью

5.2.2. Главное окно программы XILab в режиме управления одной осью
- Блок управления движением двигателя
  - Движение без точного задания конечного положения
  - Движение в заданную точку
  - Целевая позиция для команд движения
- Блок управления аттенюатором
- Состояние контроллера и мотора
  - Электропитание контроллера
  - Состояние мотора
  - Состояние программы
- Группа кнопок для управления программой
- Статусная строка.



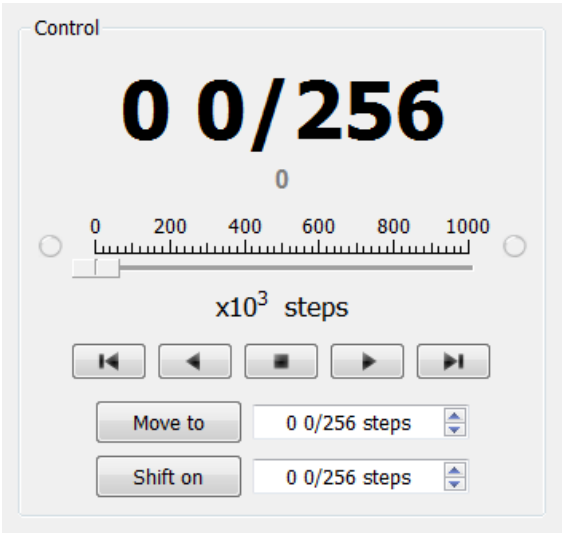
Главное окно программы XILab в режиме двигателя



Главное окно программы XILab в режиме аттенюатора

В левой части окна в группах параметров **Power** и **Motor** находятся данные о состоянии контроллера и мотора в настоящий момент. В центральной части окна расположен блок **Control**, содержащий индикаторы текущей позиции и элементы управления движением мотора. Блок **Control** в зависимости от настроек может принимать вид блока управления движением произвольного двигателя или блока управления аттенюатором. Справа расположена группа кнопок для управления программой в целом. Внизу расположен **лог**, при минимальном размере окна он скрыт. Под логом находится статусная строка. Рассмотрим эти группы более подробно.

Блок управления движением двигателя



блок Control

В центральной части блока расположен индикатор текущей позиции, под ним, если энкодер включен, располагается индикатор позиции по энкодеру. В режиме ведущего энкодера, см. раздел Работа с энкодерами главный и второстепенный индикаторы меняются местами.

Ниже расположен блок **Control**, содержащий элементы управления движением мотора. Рассмотрим их более подробно:

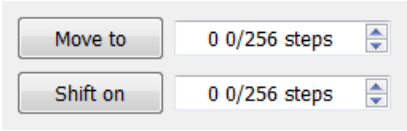
Движение без точного задания конечного положения



Кнопки управления движением

- Кнопки Влево, Стоп и Вправо запускают движение влево без указания конечной позиции, останавливают с замедлением начатое движение и запускают движение вправо без указания конечной позиции, соответственно.
- Кнопка Влево до границы заставит мотор вращаться до левой границы слайдера. Вправо до границы, соответственно, до правой границы слайдера.
- Нажатие и удержание кнопок клавиатуры Вправо, Влево при нахождении фокуса ввода в блоке слайдера начинает движение в направлении увеличения или уменьшения координаты. При отпускании кнопки движение прекращается, как будто была нажата кнопка Стоп на главном окне.

Движение в заданную точку



Управление движением в заданную точку

- Кнопка Move to запускает процесс перемещения в заданную позицию.
- Кнопка Shift on запускает процесс смещения на заданное расстояние от целевой позиции.

Целевая позиция для команд движения

Команды Move to и Shift on используют целевую позицию для расчета движения. Целевая позиция изменяется следующими командами:

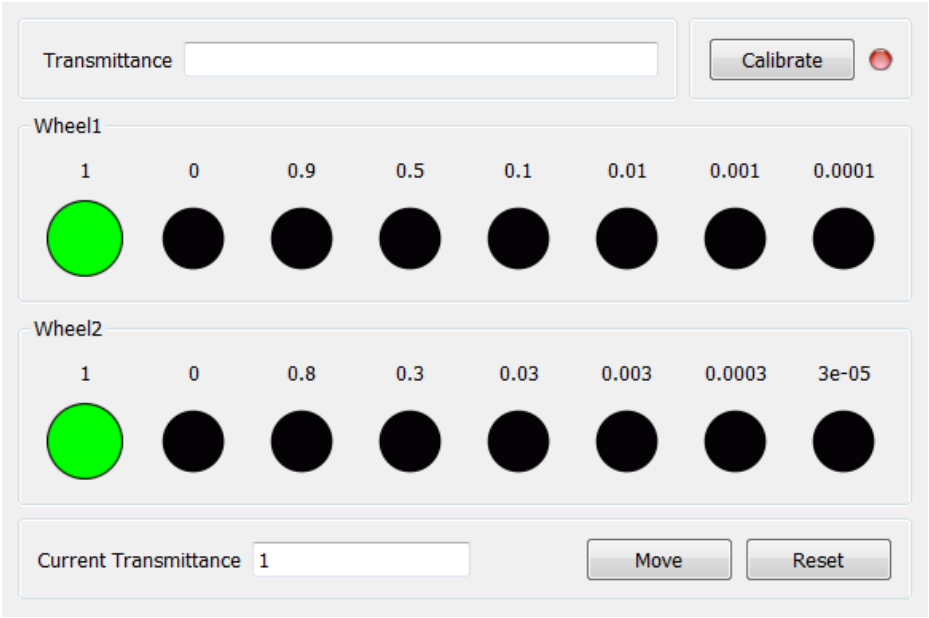
Move to <величина>
Целевая позиция = <величина>

Shift on <смещение>
Целевая позиция = целевая позиция + <смещение>

Zero (при условии отсутствия движения в момент посылки команды)
Целевая позиция = 0

Команды Stop, Влево, Вправо, Влево до границы и Вправо до границы не изменяют целевую позицию.

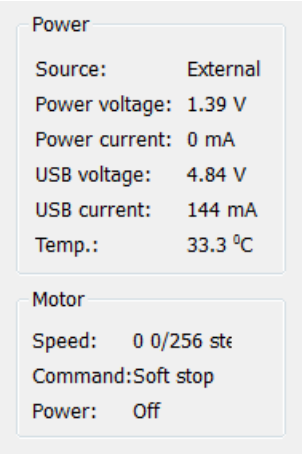
Блок управления аттенюатором



блок управления аттенюатором

В верхней части блока расположено окно Transmittance и кнопка Calibrate. Окно Transmittance предназначено для выбора желаемого коэффициента пропускания. Кнопка Calibrate предназначена для ручного поиска начальной позиции аттенюатора и при нажатии запускает сначала движение на один оборот аттенюатора с текущими настройками для однозначного определения относительного положения дисков аттенюатора, а затем функцию Автокалибровка "домашней" позиции. Нажатие Calibrate не является необходимым для движения - в случае если аттенюатор не производил калибровку или калибровка была сброшена, к примеру нажатием Cancel в процессе движения, при следующем движении калибровка будет сделана автоматически.
Аттенюатор может работать с одним или двумя дисками (у каждого диска имеется 8 фильтров), поэтому ниже будет находиться одно или два поля, соответствующие одному или двум дискам.
Далее располагается окно Current Transmittance, в котором показывается коэффициент пропускания (который группируется из коэффициентов пропускания имеющихся фильтров) наиболее близкий к желаемому.
При нажатии на кнопку Move присходит движение к тем фильтрам, которые соответствуют Current Transmittance, причем данные фильтры подсвечиваются зеленым цветом, т.е делаются активными.
Нажатие на кнопку Reset делает все фильтры неактивными (становятся серого цвета).

Состояние контроллера и мотора



Электропитание контроллера

Группа параметров **Power** содержит индикаторы:

- Source - источник электропитания контроллера. Контроллер может питаться от USB или от внешнего External источника напряжения.
- Power voltage - напряжение на силовой части.
- Power current - ток потребления силовой части.
- USB voltage - напряжение на USB.
- USB current - ток, потребляемый контроллером по USB.
- Temp. - температура процессора контроллера.

Изменение цвета индикатора Power voltage на синий или красный показывает выход за рамки диапазона допустимых значений напряжения источника питания относительно разрешенного, установленного в разделе Настройка предельных параметров контроллера. При этом контроллер переходит в состояние Alarm. Выход из состояния Alarm возможен по прекращению события, вызвавшего Alarm, если флаг Sticky Alarm не установлен и с помощью команды остановки, кнопка Stop, если флаг Sticky Alarm установлен.

Изменение цвета индикатора Power current на красный показывает превышение тока, потребляемого контроллером от источника питания, относительно разрешенного, установленного в разделе Настройка предельных параметров контроллера. При этом контроллер переходит в состояние Alarm. Выход из состояния Alarm возможен по прекращению события, вызвавшего Alarm, если флаг Sticky Alarm не установлен и с помощью команды остановки, кнопка Stop, если флаг Sticky Alarm установлен.

Изменение цвета индикатора USB voltage на синий или красный показывает выход за рамки диапазона допустимых значений напряжения USB в сторону меньшего и большего напряжения соответственно. При этом контроллер переходит в состояние Alarm. Выход из состояния Alarm возможен по прекращению события, вызвавшего Alarm, если флаг Sticky Alarm не установлен и с помощью команды остановки, кнопка Stop, если флаг Sticky Alarm установлен.

Изменение цвета индикатора USB current на красный показывает превышение тока потребления USB относительно разрешенного, установленного в разделе Настройка предельных параметров контроллера. При этом контроллер переходит в состояние Alarm. Выход из состояния Alarm возможен по прекращению события, вызвавшего Alarm, если флаг Sticky Alarm не установлен и с помощью команды остановки, кнопка Stop, если флаг Sticky Alarm установлен.

Изменение цвета индикатора Temp. на красный показывает превышение температуры на плате контроллера относительно разрешенной, установленной в разделе Настройка предельных параметров контроллера. При этом контроллер переходит в состояние Alarm. Выход из состояния Alarm возможен по прекращению события, вызвавшего Alarm, если флаг Sticky Alarm не установлен и с помощью команды остановки, кнопка Stop, если флаг Sticky Alarm установлен.

Состояние мотора

Группа параметров **Motor** содержит индикаторы:

- Speed - скорость вращения мотора.
- Command - последняя выполняемая (жирный шрифт) или выполненная (обычный шрифт) команда контроллера. Команда контроллера отображается черным цветом, если флаг ошибки движения MVCMD_ERROR не установлен, в противном случае красным. Может быть одним из следующих вариантов:
  - Move to position - перемещение в заданную позицию
  - Shift on offset - смещение на заданное расстояние
  - Move left - движение влево
  - Move right - движение вправо
  - Stop - остановка
  - Homing - нахождение начальной позиции
  - Loft - компенсация люфта
  - Soft stop - плавная остановка
  - Unknown - неизвестная команда (возможно сразу после включения контроллера)
- Power - состояние питания шагового двигателя. Может быть одним из следующих вариантов:
  - Off - обмотки мотора разомкнуты и не управляются драйвером,,

- Short - обмотки замкнуты накоротко через драйвер,
- Norm - обмотки запитаны номинальным током,
- Reduc - обмотки намеренно запитаны уменьшенным током от рабочего для снижения потребляемой мощности,
- Max - обмотки запитаны максимально доступным током, который может выдать схема при данном напряжении питания.

Состояние программы

Группа паарметров **Program** содержит индикаторы:

- Sync buf free - количество свободных ячеек в буфере команд контроллера (см. описание команды ASIA).

Группа кнопок для управления программой

- Кнопка Settings... открывает настройки контроллера, см. раздел Настройки программы.
- Кнопка Chart... открывает окно с графиками, см. раздел Графики.
- Кнопка Scripting... открывает окно работы со скриптами, см. раздел Скрипты.

- Кнопка Go home осуществляет поиск начальной позиции, см. раздел Настройка исходного положения.
- Кнопка Cyclic включает режим циклического движения, см. раздел Настройка режима циклического движения.
- Кнопка Zero обнуляет текущую позицию мотора и значение энкодера.
- Кнопка Stop посылает команду экстренной остановки, сбрасывает состояние Alarm, очищает буфер команд для синхронного движения и останавливает выполнение скрипта, если он запущен.

- Кнопка Save log сохраняет содержимое лога в файл в формате CSV (открывается диалог выбора файла для записи).
- Кнопка Clear log очищает содержимое лога.
- Кнопка Exit осуществляет корректное завершение работы, см. раздел Корректное завершение работы.

Статусная строка.

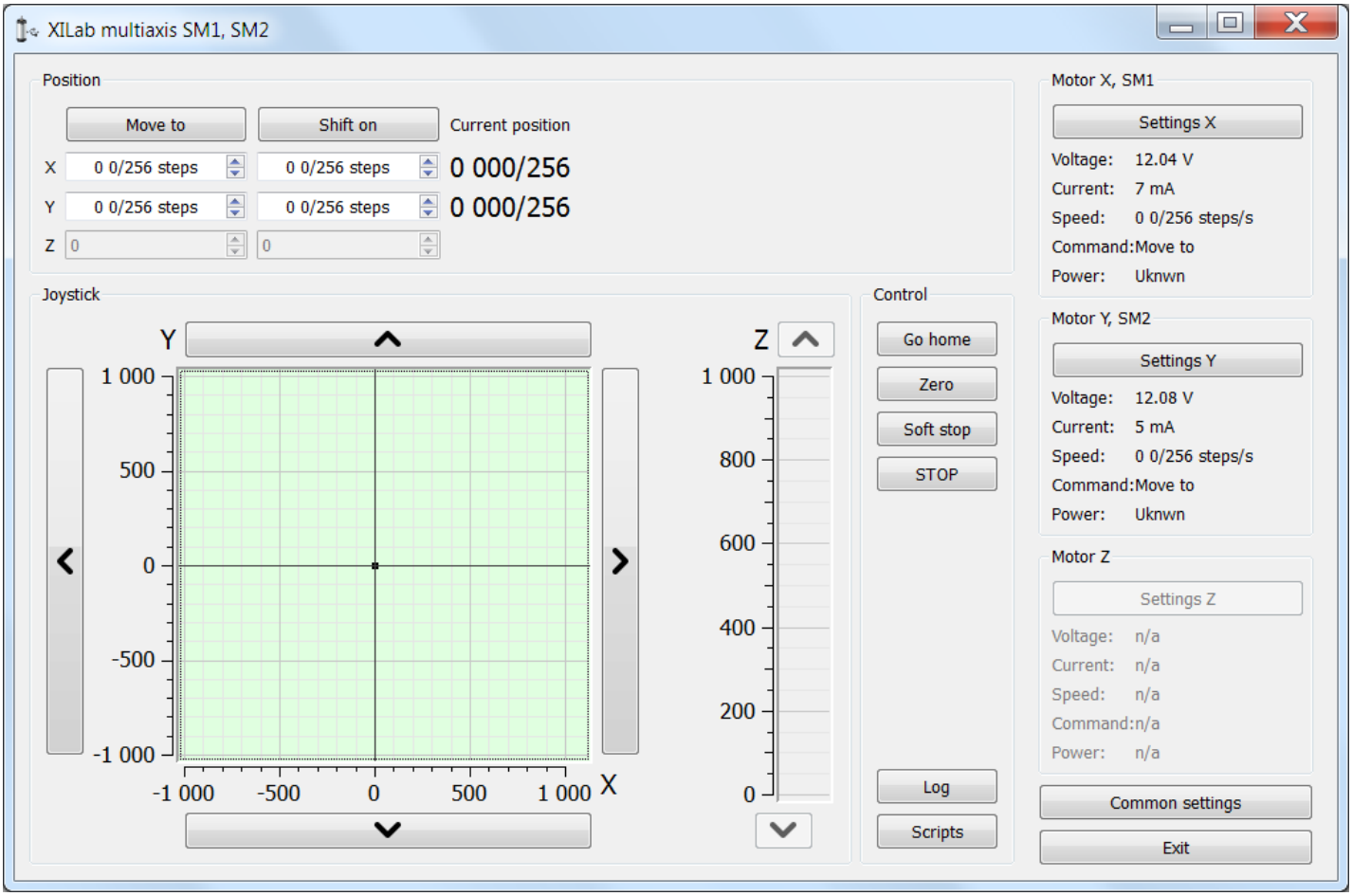
В статусной строке находятся индикаторы текущего состояния контроллера. Слева направо это блок 7 флагов,

- L - Левая кнопка нажата.
- R - Правая кнопка нажата.
- G - Вход/выход GPIO активен.
- B - Магнитный тормоз запитан.
- S - Датчик оборотов активен.
- I - Вход синхронизации активен.
- O - Выход синхронизации активен.

и отдельные индикаторы

- EEPR - Подключена память EEPROM с настройками.
- HOMD - Калибровка выполнена.
- WndA - Состояние обмотки A.
- WndB - Состояние обмотки B.
- ENCD - Состояние энкодера (нет/присутствует/инвертирован/неисправен).
- PWHT - Перегрелась силовая часть платы.
- SLIP - Обнаружено проскальзывание.
- ErrC - Недопустимая команда.
- ErrD - Нарушение целостности данных.
- ErrV - Недопустимое значение данных.

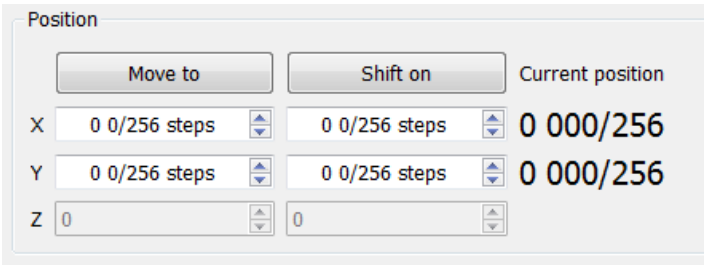
5.2.3. Главное окно программы XILab в режиме управления несколькими осями



Главное окно программы XILab

В левой верхней части окна в группе параметров **Position** находятся индикаторы текущей позиции. В левой нижней части окна расположены блоки **Joystick** и **Control**, представляющие собой графический элемент управления движением по нескольким осям и блок кнопок соответственно. В правой верхней части в блоках **Motor** находятся данные о состоянии контроллеров и подключенных к ним моторов в настоящий момент. В правой нижней части окна расположена группа кнопок для управления программой в целом. Рассмотрим эти группы более подробно.

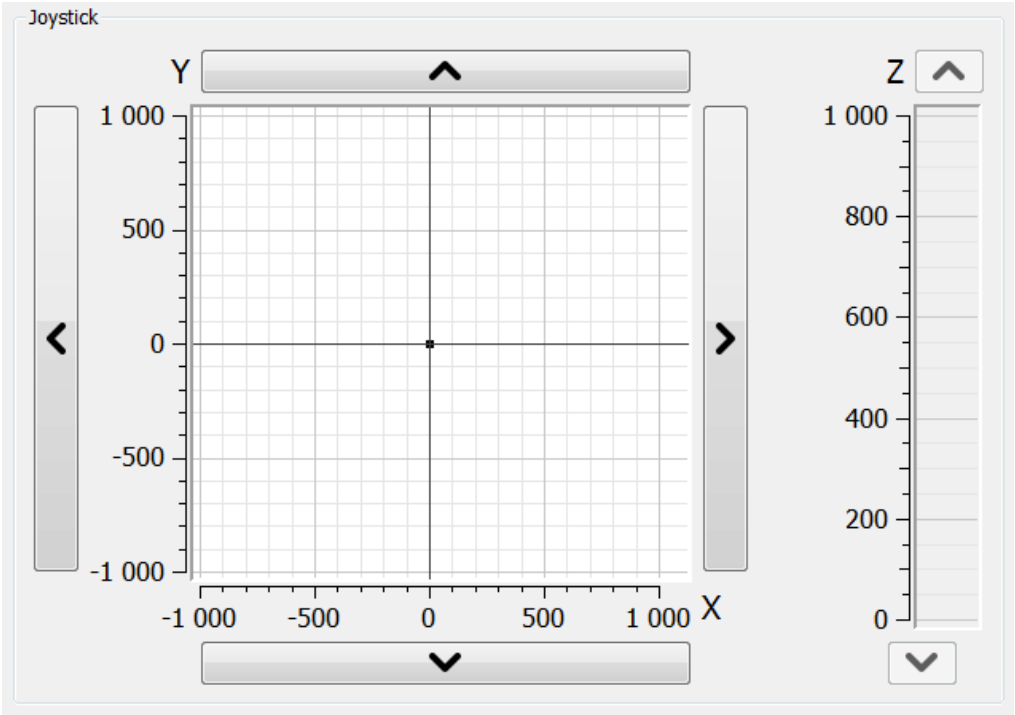
Блок позиции и движения



Блок позиции и движения

В столбце Current position расположены индикаторы текущей позиции в шагах или калиброванных единицах (см. далее) для осей X, Y и Z сверху вниз. Кнопка Move to осуществляет перемещение в координату, задаваемую элементами управления в этом столбце, а кнопка Shift on осуществляет относительное смещение на задаваемое расстояние. Если какой-то из контроллеров отсутствует или временно отключен, то соответствующая ему строка становится недоступной.

Блок виртуального джойстика



Блок виртуального джойстика

В этом блоке текущая координата контроллеров визуализируется точкой с двумя линиями на плоскости для осей X-Y и линией для оси Z. Здесь также возможно управлять движением контроллеров несколькими способами:

- при клике мышкой где-либо на плоскости X-Y или в столбце Z соответствующий контроллер или контроллеры начинают движение в выбранную координату со своими настройками движения.
- при нажатии и удержании мышкой экранных кнопок со стрелками вверх, вниз, вправо и влево соответствующая ось начинает движение в выбранном направлении. Движение прекращается с замедлением при отпускании кнопки (посылается команда *SSTP*).
- при нажатии и удержании кнопок клавиатуры вправо, влево, вверх, вниз, PageUp, PageDown при нахождении фокуса ввода в блоке джойстика ось X, Y или Z соответственно начинает движение в направлении увеличения или уменьшения координаты. Движение прекращается с замедлением при отпускании кнопки (посылается команда *SSTP*). Наличие фокуса ввода в блоке джойстика можно отследить по изменению цвета его фона с белого на светло-зеленый.

Масштаб осей задается в блоке "Slider settings" вкладки Program configuration в окне Settings индивидуально для каждого контроллера. Если включена опция пользовательских единиц, то координата по соответствующей оси отсчитывается в этих единицах. В случае если считанное из контроллера положение по какой-либо оси выходит за диапазон оси то соответствующий индикатор не отображается.

Блок управления

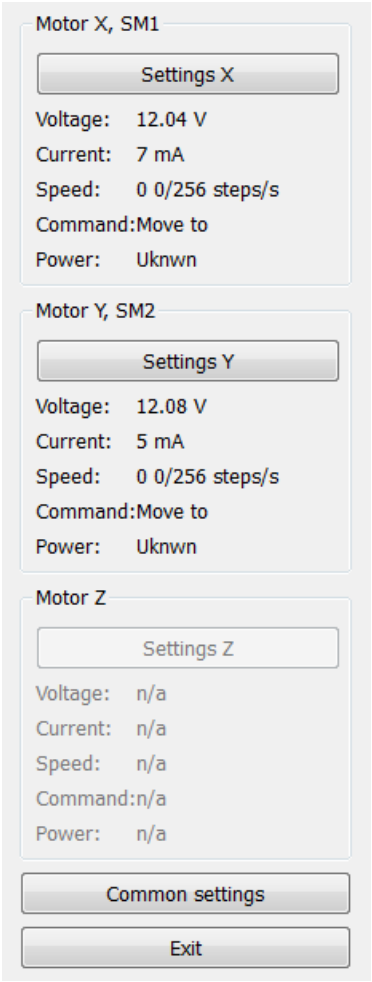


Блок управления

Кнопка Go home осуществляет поиск начальной позиции независимо для каждого контроллера, см. раздел Настройка исходного положения.
Кнопка Zero обнуляет текущую позицию мотора и значение энкодера для каждого контроллера.
Кнопка Soft stop выполняет команду плавной остановки для каждого контроллера.
Кнопка STOP посылает команду экстренной остановки каждому контроллеру, сбрасывает их состояния Alarm, очищает их буферы команд для синхронного движения и останавливает выполнение скрипта, если он запущен.

Кнопка Log открывает окно, отображающее лог программы. Сюда попадает диагностическая информация (ошибки, предупреждения, информационные сообщения) от самой программы XiLab, от библиотеки libximc, а также от исполняемых скриптов.
Кнопка Scripts открывает окно работы со скриптами, см. раздел Скрипты.

Блок индикаторов состояния контроллеров и моторов



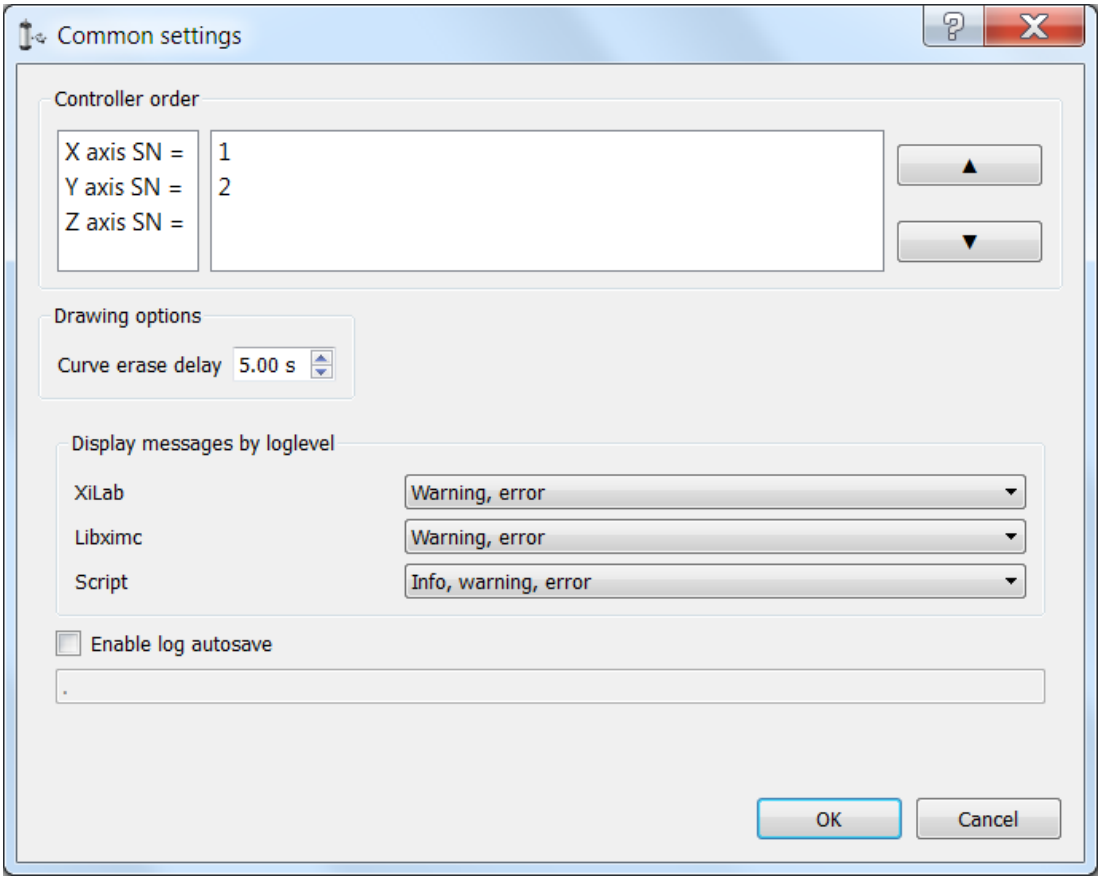
Блок индикаторов состояния контроллеров и моторов

Здесь расположены три блока индикаторов состояния контроллеров и моторов для осей X, Y и Z. В заголовке блока расположен серийный номер соответствующего контроллера. Каждый блок содержит индикаторы:

- Voltage - напряжение на силовой части мотора.
- Current - ток потребления силовой части мотора.
- Speed - текущая скорость мотора.
- Command - последняя выполненная или выполняемая команда контроллера.
- Power - состояние питания мотора.

Кнопка Settings X,Y,Z открывает настройки контроллера соответствующего этой оси, см. раздел Настройки программы.

Снизу от блоков индикаторов состояния расположены две кнопки: Common settings и Exit.
Кнопка Common settings открывает диалог с общими настройками, включающими в себя настройки логирования и соответствие серийных номеров контроллеров отображаемым осям X, Y и Z.
Кнопка Exit осуществляет корректное завершение работы, см. раздел Корректное завершение работы.



Диалоговое окно общих настроек многоосного интерфейса

В окне общих настроек расположен блок управления порядком отображения осей в интерфейсе, блок настроек отрисовки кривой движения и блок настроек логирования.

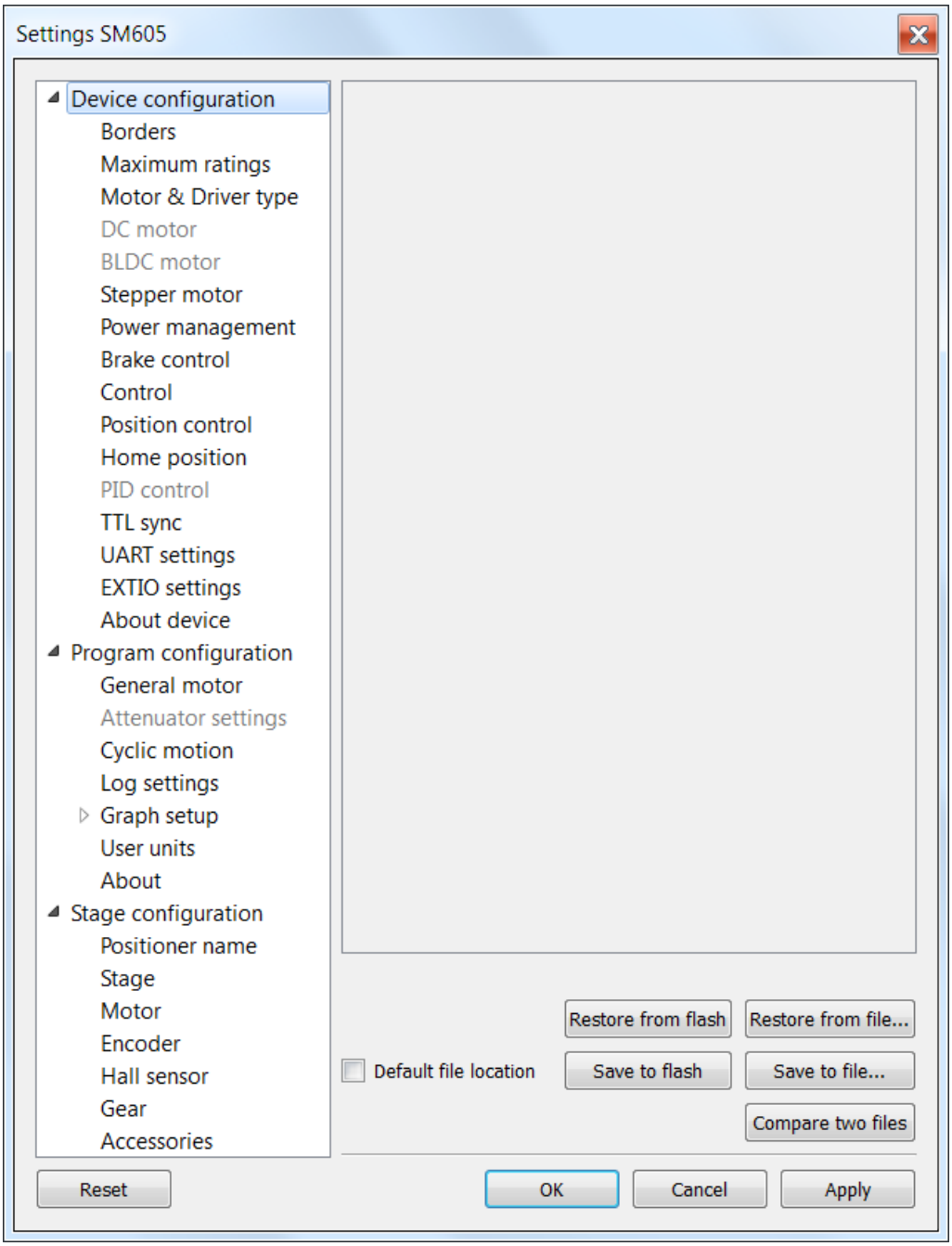
Порядок осей можно менять, выбрав серийный номер желаемой оси и нажимая кнопки "вверх" и "вниз", которые расположены справа. Это приведет к перемещению оси с выбранным серийным номером вверх или вниз в списке осей. Первая ось в списке, то есть ось с серийным номером в строке справа от надписи "X axis SN = " будет идентифицироваться как "ось X", вторая как "ось Y", третья, если она присутствует, как "ось Z".

В блоке "Drawing options" можно указать время T в секундах. Траектория движения двух контроллеров по осям X и Y за последние T секунд будет нарисована на экране виртуального джойстика как двумерная кривая. Если вы хотите отключить показ траектории, введите здесь число "0".

Блок настроек логирования полностью аналогичен одноосной версии.
В нем можно настроить уровень подробности логгирования: не выводить ничего (None), выводить только ошибки (Error), ошибки и предупреждения (Error, Warning), ошибки, предупреждения и информационные сообщения (Error, Warning, Info) для каждого из источников: программа XILab, библиотека libximc и модуль скриптов Scripts.
При включенном автосохранении запись в файл производится каждые 5 секунд. Имя файла: "xilab_log_YYYY.MM.DD.csv", где YYYY, MM и DD - текущие год, месяц и день соответственно. Формат сохраненных данных: CSV.

5.2.4. Настройки программы

Кнопка Settings из главного окна программы открывает окно настроек.



Основное окно настроек XILab

Настройки программы представлены в виде иерархического дерева и разделены на три большие группы: настройки контроллера - "Device configuration", настройки программы XILab - "Program configuration", характеристики позиционера - "Stage configuration".

В первой группе **Device configuration** находятся параметры, значения которых могут храниться непосредственно в устройстве (во флеш памяти или в ОЗУ контроллера).

Вторая группа **Program configuration** содержит настройки программы XILab, которые не записываются в контроллер, а служат для управления работой самого интерфейса XILab.

Третья группа **Stage configuration** содержит информацию о параметрах позиционера, считанную из ROM-микросхемы памяти позиционера.

Описание кнопок **Restore from flash** и **Save to flash** находится в разделе Хранение параметров во flash-памяти контроллера .

Все настройки программы из первой и второй группы настроек могут быть записаны во внешний файл при нажатии на кнопку **Save to file** .
При нажатии в XILab на кнопку **Restore from file** настройки программы загружаются в окно Settings.

При нажатии на кнопку **Compare two files** открывается диалог выбора двух файлов, а затем сравниваются все их настройки и выводится список различий. Отсутствующие в одном из файлов ключи помечаются в таблице как "<NO KEY>".

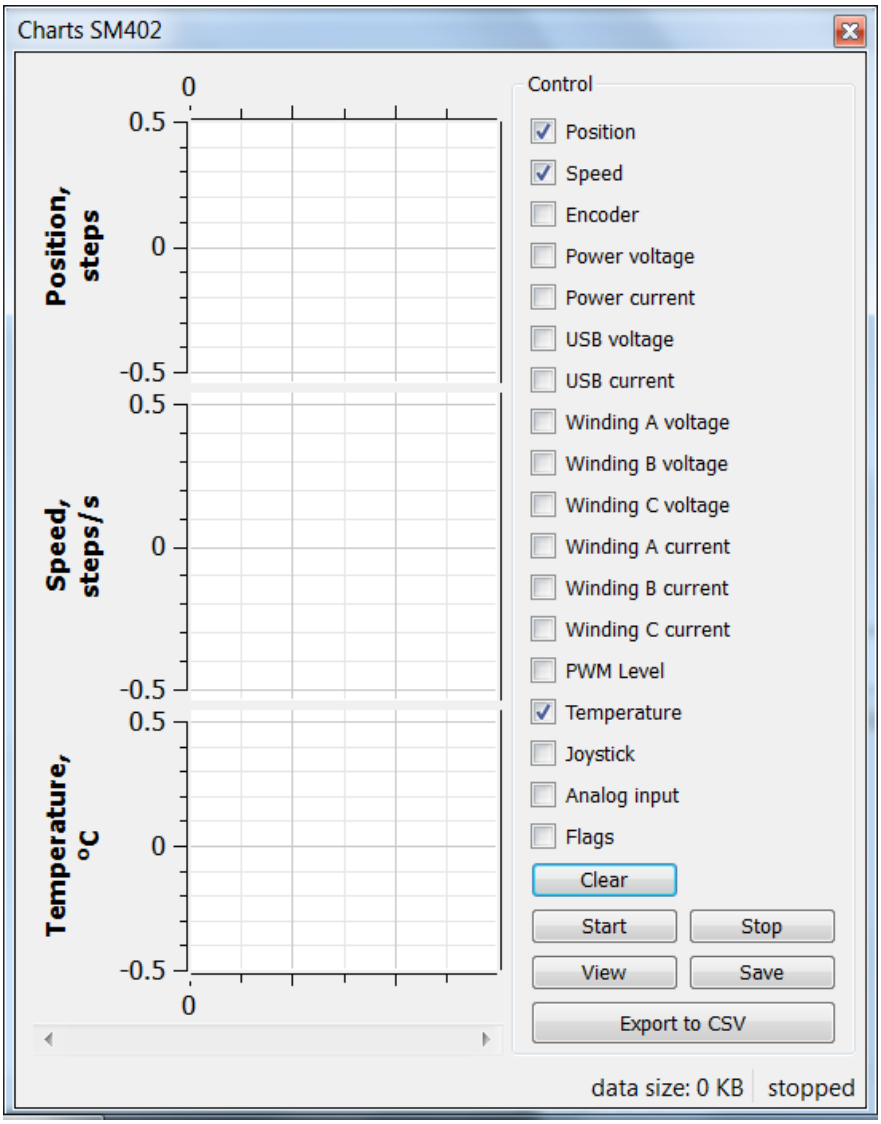
Кнопка **OK** закрывает окно Settings с сохранением всех измененных настроек в контроллер, кнопка **Cancel** закрывает окно без сохранения, кнопка **Apply** сохраняет настройки без закрытия окна.

Кнопка **Reset** сбрасывает все изменения настроек, сделанные после последнего нажатия **Apply**, или после открытия окна Settings, если кнопка **Apply** не нажималась.

> *Замечание.* При сохранении настроек в flash память контроллера туда могут быть записаны только настройки блока Device configuration.

5.2.5. Графики

Кнопка Chart из главного окна программы открывает окно для работы с графиками.



Окно графиков программы XILab

В левой части окна расположены графики величин, в правой части окна расположен блок **Control**, содержащий элементы управления графиками. Вверху блока **Control** расположены флаги включения различных графиков, внизу блока **Control** расположена группа кнопок для управления сохраненными данными графиков.

Отображаемые на графиках величины

- Position - первичное поле, в котором хранится текущая позиция, как бы ни была устроена обратная связь. В случае работы с DC-мотором в этом поле находится текущая позиция по данным с энкодера, в случае работы с ШД (шаговым двигателем) в этом поле содержится значение текущей позиции в шагах;
- Speed - текущая скорость;
- Encoder - позиция по второстепенному датчику положения;
- Power voltage - напряжение силовой части;
- Power current - ток потребления силовой части;
- USB voltage - напряжение на USB;
- USB current - ток потребления по USB;
- Winding_A_current - в случае ШД, ток в обмотке A; в случае бесщеточного, ток в первой обмотке; в случае DC в единственной;
- Winding_B_current - в случае ШД, ток в обмотке B; в случае бесщеточного, ток в второй обмотке; в случае DC не используется;
- Winding_C_current - в случае бесщеточного, ток в третьей обмотке; в случае ШД и DC не используется;
- Winding_A_voltage - в случае ШД, напряжение на обмотке A; в случае бесщеточного, напряжение на первой обмотке; в случае DC на единственной;
- Winding_B_voltage - в случае ШД, напряжение на обмотке B; в случае бесщеточного, напряжение на второй обмотке; в случае DC не используется;
- Winding_C_voltage - в случае бесщеточного, напряжение на третьей обмотке; в случае ШД и DC не используется;
- PWM_level - коэффициент заполнения ШИМ (только для двигателей постоянного тока);

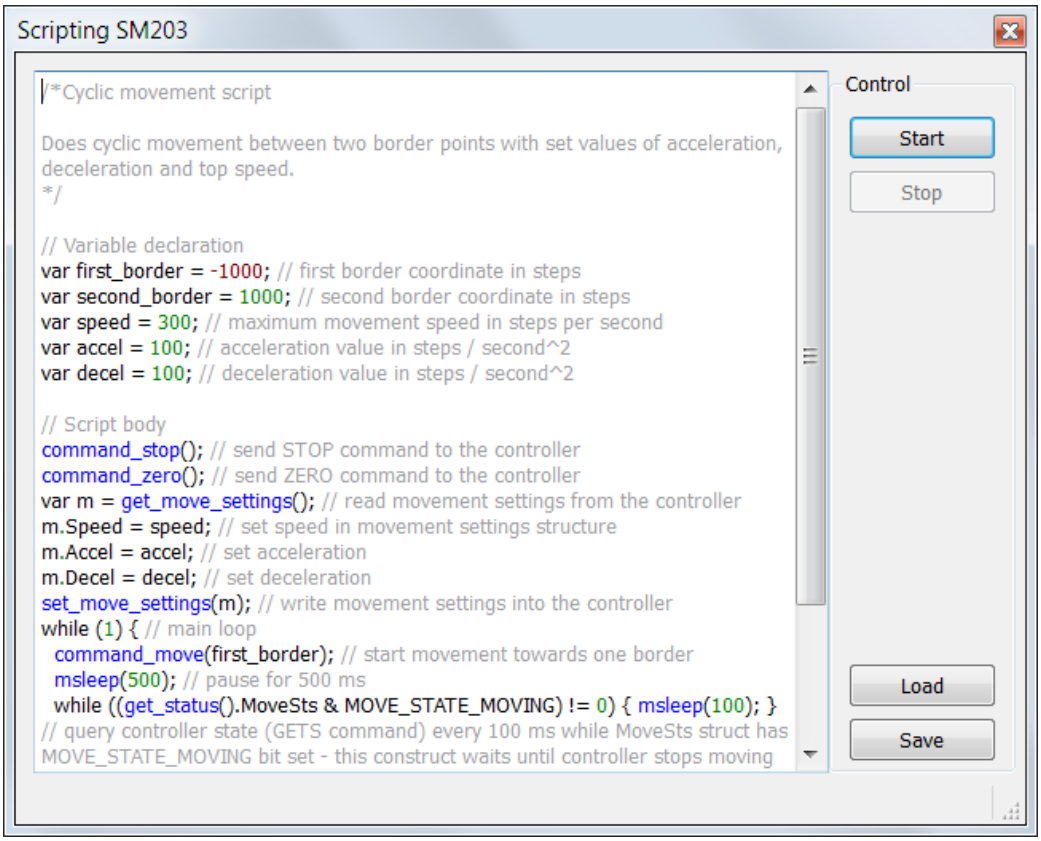
- Temperature - температура процессора контроллера;
- Joystick - значение входного сигнала от джойстика;
- Analog input - значение аналогового входа для пользовательских задач;
- Flags - состояние флагов контроллера.

Функции кнопок

- Clear - очищает сохраненные данные и окно графиков;
- Start - начинает запись данных и отображение графиков. Если включена опция "Break data update when motor stopped" в **Program configuration -> Graph setup -> Common**, то запись данных и автопрокрутка графиков при остановленном двигателе происходить не будет;
- Stop - останавливает считывание данных;
- Save - сохраняет данные графиков в файл;
- Load - загружает данные графиков из ранее сохраненного файла;
- Export to CSV - экспортирует данные графиков в файл формата CSV

5.2.6. Скрипты

Кнопка Scripting из главного окна программы открывает окно для работы со скриптами.



Окно скриптов XILab

В левой части окна расположено поле для редактирования текста скрипта, в правой части окна расположен блок Control, содержащий элементы управления скриптами.

Функции кнопок

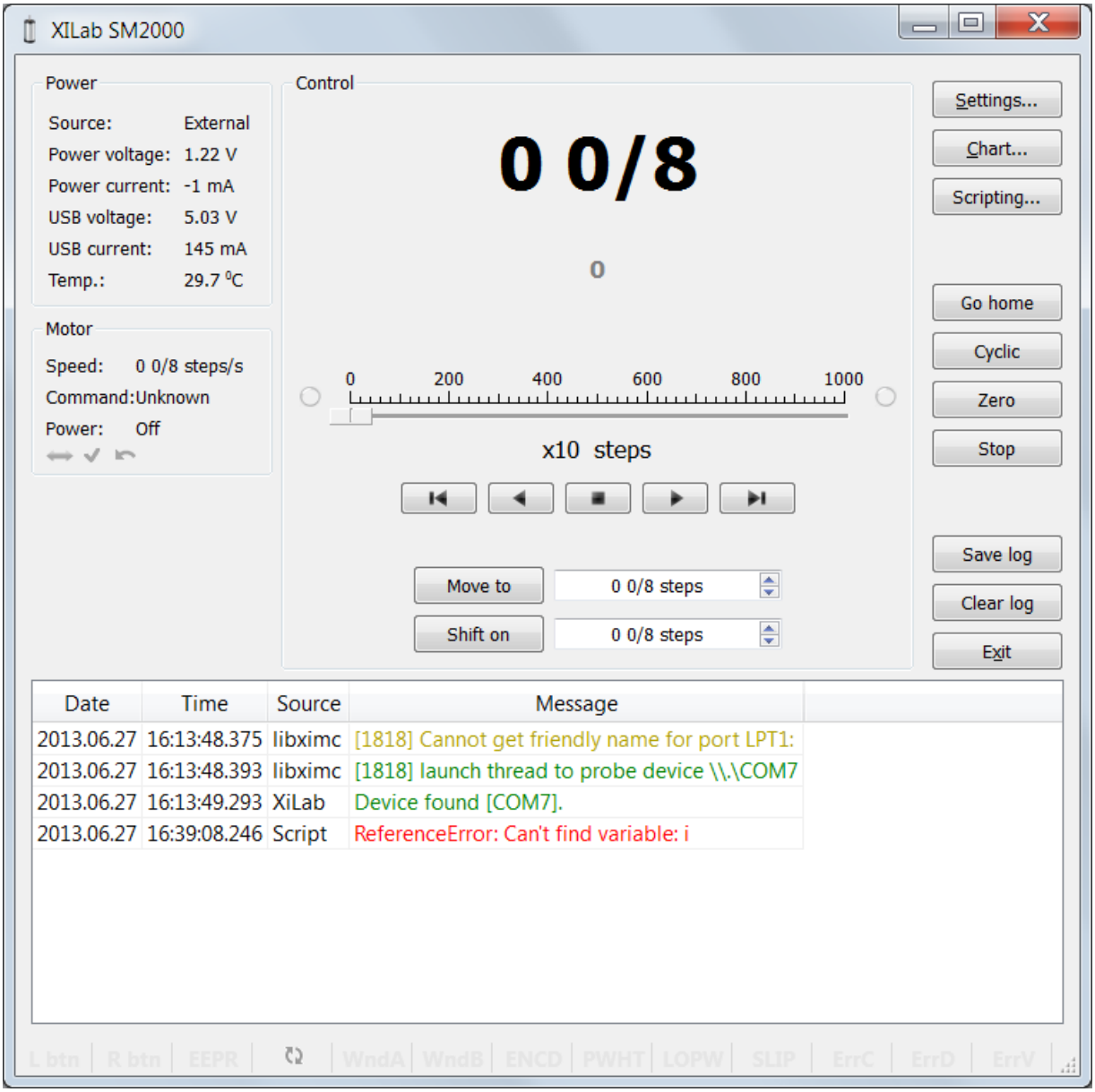
- Start - запускает выполнение скрипта. Неактивна, если скрипт уже выполняется. Сразу после нажатия кнопки и до начала интерпретации скрипта происходит автоматическое сохранение скрипта во временный файл (см. ниже).
- Stop - останавливает выполнение скрипта. Неактивна, если скрипт в данный момент не выполняется.
- Save - вызывает диалог выбора файла куда будет сохранен текущий скрипт, отображаемый в окне. Неактивна во время выполнения скрипта.
- Load - вызывает диалог выбора файла для загрузки в окно скриптов. Неактивна во время выполнения скрипта.
  Внимание, в случае загрузки все несохраненные изменения текста в окне будут потеряны!

В момент старта программы в окно "Scripting" загружается последнее сохраненное содержимое текста в окне. Автосохранение происходит перед каждым стартом скрипта, а также перед выходом из XILab, файл автосохранения находится в директории пользовательских настроек программы и имеет имя "scratch.txt".

> *Замечание.*
> Выполнение скрипта останавливается при нажатии кнопки "Stop" в главном окне программы, это сделано для экстренной остановки движения в случае необходимости.

Описание языка скриптов находится в разделе Программирование.

5.2.7. Лог XiLab



Окно лога программы XILab

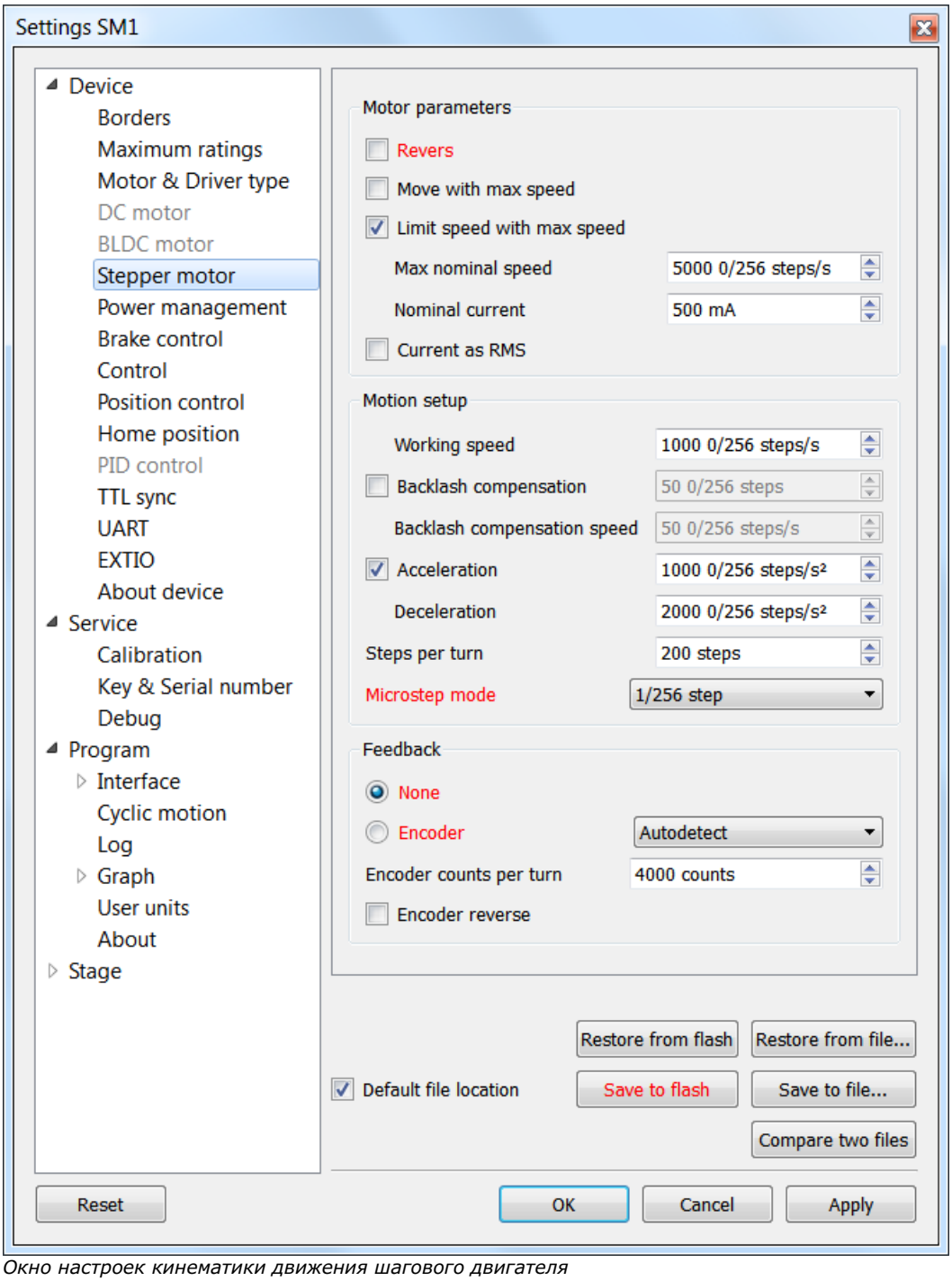
В нижней части главного окна XILab расположен лог, в который записываются сообщения от библиотеки libximc. В него также выводятся сообщения самого XILab и интерпретатора скриптов.
Лог имеет 4 колонки: дата и время возникновения записи, источник и текст сообщения.
Сообщения имеют уровень логгирования, означающий важность сообщения: ошибка, предупреждение и информационное сообщение. Сообщения ошибок выводятся красным цветом, предупреждения желтым, информационные сообщения зеленым.
Настроить тип выводимых сообщений в лог можно на вкладке Настройка логирования в окне настроек программы.

5.3. Настройки контроллера

1. Настройка кинематики движения (Шаговый двигатель)
2. Настройка диапазона движения и концевых выключателей
3. Настройка предельных параметров контроллера
4. Настройка параметров энергопотребления
5. Настройка исходного положения
6. Настройки синхронизации
7. Настройка тормоза
8. Контроль позиции
9. Настройка внешних управляющих устройств
10. Настройки UART
11. Настройки цифрового входа-выхода общего назначения
12. Настройка типа двигателя
13. Настройка кинематики движения (DC мотор)
14. Настройка контуров ПИД-регулирования
15. О контроллере

5.3.1. Настройка кинематики движения (Шаговый двигатель)

В окне настроек программы ***Device configuration -> Stepper motor***



Окно настроек кинематики движения шагового двигателя

Motor parameters - настройки, непосредственно связанные с электромотором

Revers - установка этого флага позволяет связать направление вращения мотора с направлением счета текущей позиции. Измените состояние флага, если положительное вращение мотора уменьшает счетчик позиции. Действие этого флага равносильно подключению обмотки мотора в обратной полярности.

Move with max speed - при установленном флаге мотор игнорирует заданную скорость и вращается с максимальной допустимой скоростью.

Limit speed with max speed - при установленном флаге контроллер ограничивает максимальную скорость по количеству шагов в секунду значением поля Max nominal speed. Например, если скорость превысила номинальное значение, контроллер будет снижать выходное воздействие, пока значение скорости не вернется в пределы нормы. Однако при этом контроллер останется в рабочем состоянии и будет выполнять текущую задачу.

Max nominal speed - номинальная скорость работы мотора.

Nominal current - номинальный ток через двигатель. Контроллер будет ограничивать ток этим значением.

Current as RMS - при установленном флаге задаваемое значение тока интерпретируется как среднеквадратичное значение тока, если флаг снят, то задаваемое значение тока интерпретируется как максимальная амплитуда тока.

Motion setup - настройки, связанные с кинематикой движения

Working speed - скорость движения.

Backlash compensation - компенсация люфта. Так как механика позиционера не идеальна, существует различие при подходе к заданной точке справа и слева. При включенном режиме компенсации люфта позиционер будет подходить к точке всегда с одной стороны. Установленное значение определяет количество шагов, на которое позиционер будет проходить заданную точку чтобы возвращаться к ней с одной и той же стороны. Если указанное число больше нуля, позиционер будет подходить к точке всегда справа. Если меньше нуля, то всегда слева.

Backlash compensation speed - скорость компенсации люфта. При включенном режиме компенсации люфта Backlash compensation позиционер будет подходить к точке справа или слева с установленной скоростью, определяемой количеством шагов в секунду.

Acceleration - включает режим движения с ускорением, числовое значение поля это величина ускорения движения.

Deceleration - величина замедления движения.

Steps per turn - определяет для контроллера количество шагов для совершения мотором одного полного оборота. Параметр устанавливается пользователем.

Microstep mode - режим деления шага. Доступно 9 режимов: от целого шага до 1/256 шага. Описание режимов в разделе Поддерживаемые типы двигателей.

Feedback - настройки обратной связи

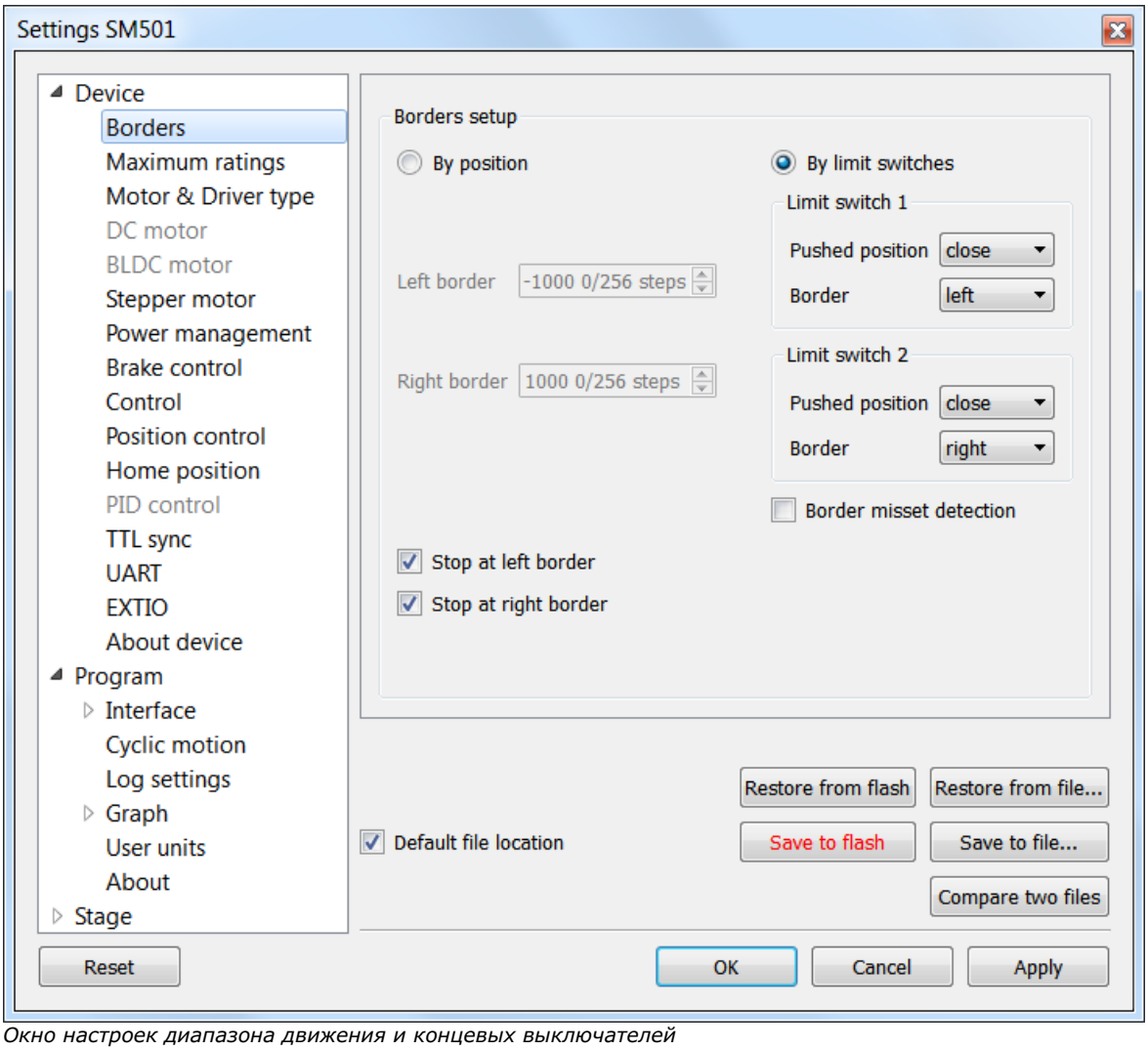
В качестве датчика обратной связи для шаговых двигателей может использоваться энкодер. При включении данной опции доступен выбор типа энкодера: недифференциальный, дифференциальный или автоматическое определение.

Encoder counts per turn - параметр определяет количество импульсов энкодера на один полный оборот оси мотора.

Encoder reverse - реверс энкодера.

5.3.2. Настройка диапазона движения и концевых выключателей

В окне настроек программы ***Device configuration -> Borders***



Окно настроек диапазона движения и концевых выключателей

Группа параметров **Borders setup** содержит параметры границ и концевых выключателей. Эти параметры позволяют предотвратить выход позиционера за допустимые физические границы его перемещения или ограничить диапазон перемещения в соответствии с требованиями пользователя. Границы могут определяться либо по положению позиционера (определяемому по внутреннему счетчику шагов контроллера), либо по концевым выключателям, установленным в крайних положениях позиционера.

Для установки границ по положению необходимо отметить пункт By position и указать значения Left border и Right border, которые соответствуют левой и правой границе соответственно.

Для установки границ по концевым выключателям необходимо выбрать пункт By limit switches и настроить работу каждого из двух концевых выключателей Limit switch 1 и Limit switch 2.
Pushed position - состояние концевика, когда он достигнут: замкнутое или разомкнутое.
Border - расположение данного концевика: слева или справа рабочего диапазона позиционера.

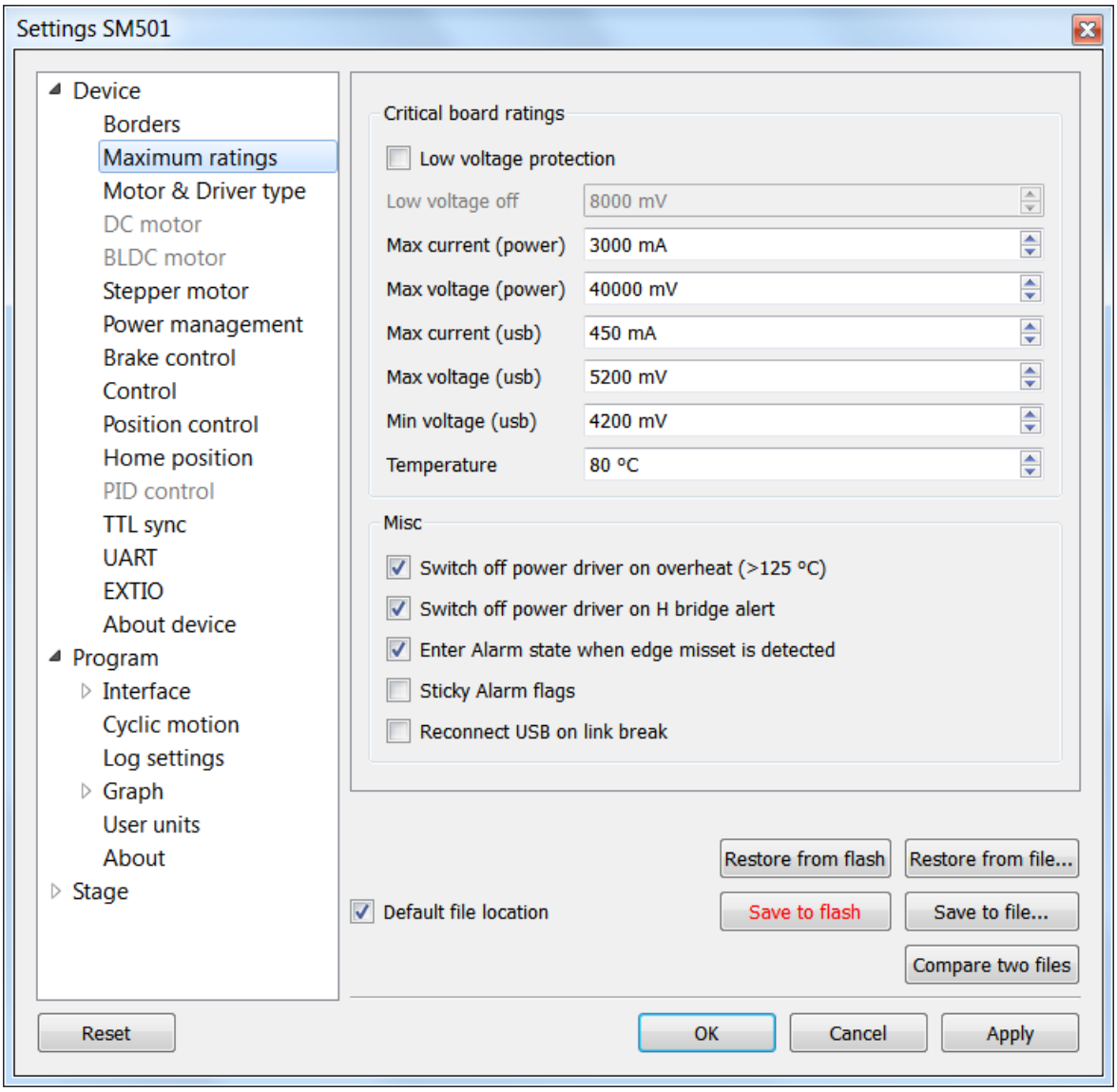
Для принудительной остановки мотора при достижении границ отметьте Stop at left border и/или Stop at right border. Тогда контроллер будет игнорировать любые команды, подразумевающие движение в сторону концевика, если соответствующий концевик уже достигнут.

При достижении граничного положения загорается соответствующий индикатор в главном окне программы.

Если флаг Border misset detection установлен, мотор останавливается при достижении обоих границ. Эта настройка нужна для предотвращения поломки двигателя при обнаружении потенциально неправильно настроенных концевиков. Обязательно прочитайте подробнее про работу контроллера в этом режиме в разделе Расположение концевых выключателей на трансляторах.

5.3.3. Настройка предельных параметров контроллера

В окне настроек программы ***Device configuration -> Maximum ratings***



Окно настроек критических параметров контроллера

Critical board ratings - эта группа параметров отвечает за максимальные значения входного тока Max current (power) и напряжения Max voltage (power) на контроллере, максимальные значения тока Max current (usb) и напряжения Max voltage (usb) на USB, минимальное значение напряжения Min voltage (usb) на USB, а также температуру Temperature платы (если измерение температуры производится у данной версии контроллера).

Если значение тока, потребляемого контроллером или величина питающего напряжения, или температура выйдут за пределы установленных здесь значений, контроллер отключает все силовые выходы и переходит в состояние Alarm. При этом в главном окне также будет информация о состоянии Alarm (фон окна сменится на красный) и у параметра вышедшего за допустимые границы значение отображается синим или красным цветом (ниже или выше порога соответственно).

При отмеченном флаге Low voltage protection включается защита от низкого напряжения питания. Low voltage off это то напряжение питания, при котором контроллер переходит в состояние Alarm.

В группу **Misc** входят все остальные настройки критических параметров.

Switch off power driver on overheat (>125 °C) - установка данного переключателя обеспечивает состояние Alarm при перегреве.

Switch off power driver on H bridge alert - установка данного переключателя обеспечивает состояние Alarm при сигнале неполадки в силовом драйвере.

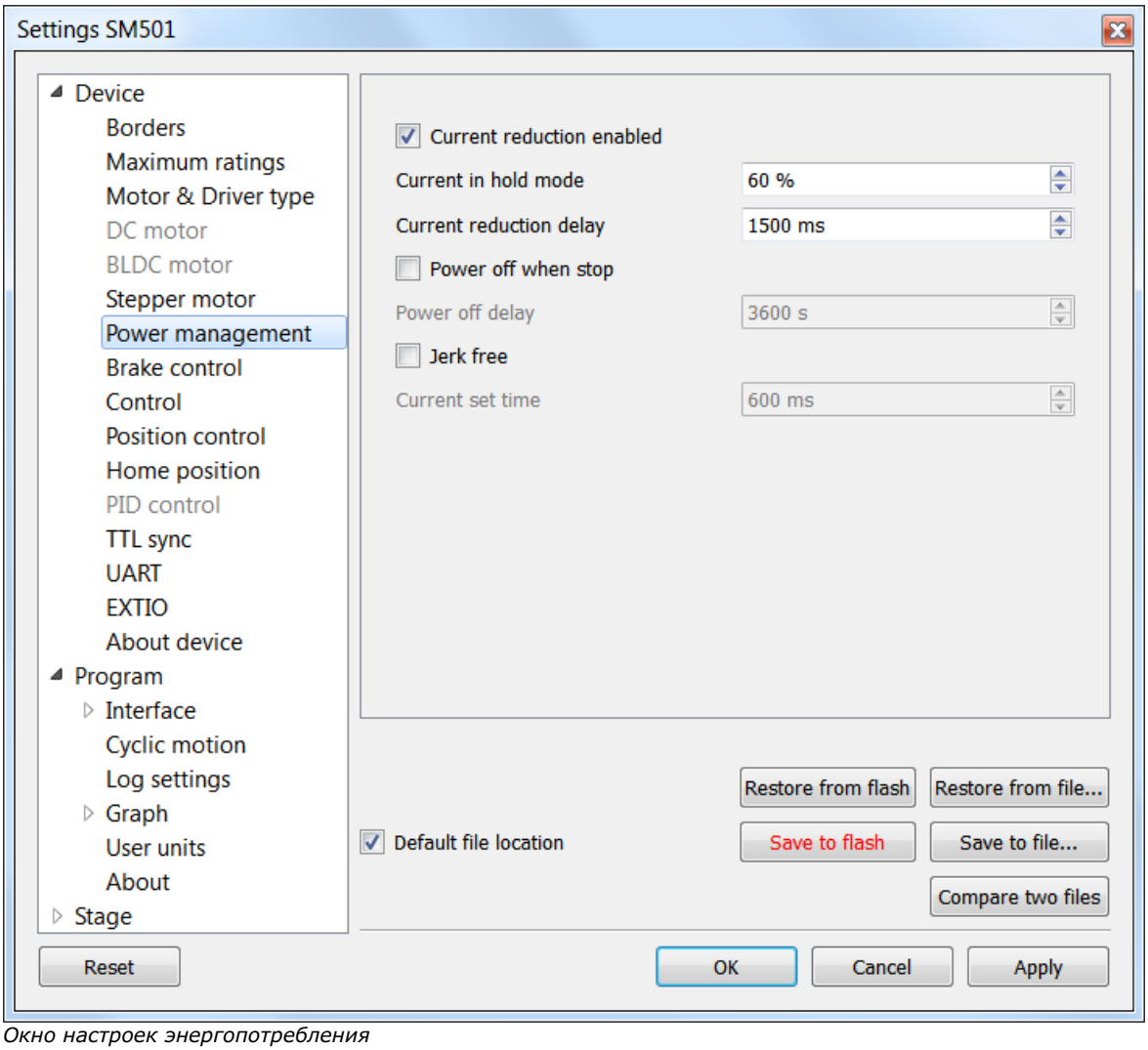
Enter Alarm state when edge misset is detected - при установке этого флага контроллер войдет в состояние Alarm при обнаружении достижения неверной границы (срабатывание правого концевика при движении влево, или наоборот).

Sticky Alarm flags - залипание состояния Alarm. При снятом флаге Sticky Alarm flags контроллер снимает флаг причины Alarm при её исчезновении (например превышение тока произошло, а дальше обмотки отключились и ток снова снизился). При установленном флаге Sticky Alarm flags флаги причины Alarm и сам режим Alarm очищаются при посылке команды Stop.

Reconnect USB on link break - при установке этого флага будет включен блок перезагрузки USB-соединения со стороны контроллера при поломке связи.

5.3.4. Настройка параметров энергопотребления

В окне настроек программы ***Device configuration -> Power Management***



Окно настроек энергопотребления

Current reduction enabled - включение функции снижения энергопотребления

- Current in hold mode - параметр определяет уровень тока в % относительно номинального значения в режиме удержания (hold mode). Диапазон значений: 0..100%.
- Current reduction delay - параметр определяет задержку от перехода в состояние STOP до уменьшения тока. Измеряется в мс. Диапазон значений: 0..65535 мс.

Power off when stop - включение функции выключения питания с обмоток мотора при переходе в состояние STOP.

- Power off delay - параметр определяет задержку в секундах от перехода в состояние STOP до полного отключения питания мотора. Диапазон значений: 0..65535 с.

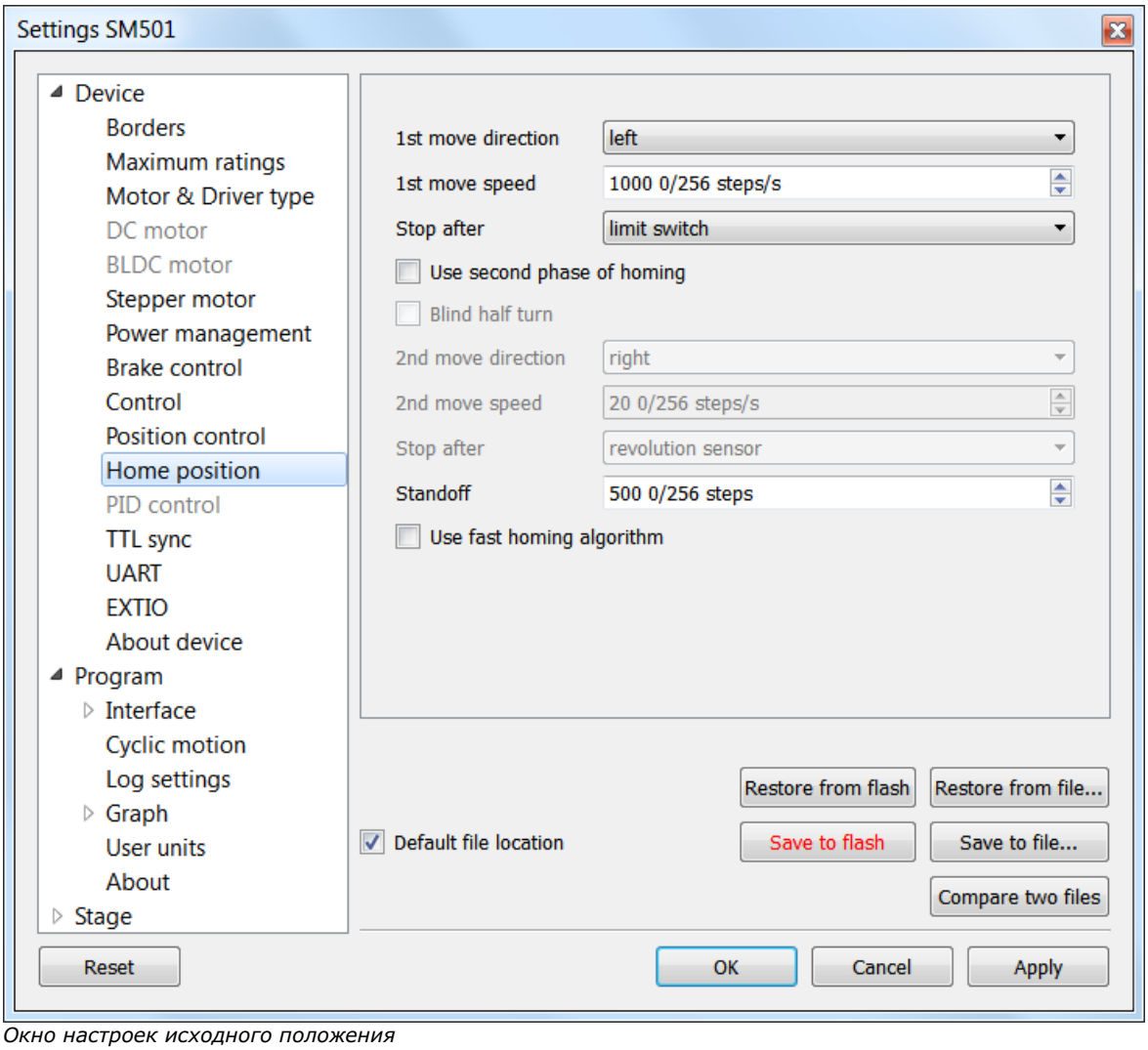
Jerk free - включение функции сглаживания тока для устранения вибраций мотора.

- Current set time - параметр определяет время установления тока в миллисекундах. Ток не может меняться быстрее, чем на 100% от номинального за это время. Диапазон значений: 0..65535 мс.

Подробное описание этих параметров находится в главе Управление питанием мотора.

5.3.5. Настройка исходного положения

В окне настроек программы ***Device configuration -> Home position***



Окно настроек исходного положения

Вкладка ***Home position*** устанавливает параметры калибровки исходного положения позиционера. Калибровка сводится к автоматическому точному обнаружению концевика, сигнала датчика оборотов или момента поступления внешнего сигнала, определяющего нулевую позицию и отстройки от нее на заданное смещение. Используется чтобы начать работу, если текущее положение позиционера не известно, но известно расположение одной реперной точки, которую мы будет называть исходным положением, относительно концевика или какого-либо другого сигнала.

Поиск исходного положения сводится к движению мотора в заданном направлении 1st move direction (вправо или влево) со скоростью Working speed до поступления сигнала от источника stop after. Затем двигатель отъезжает в противоположную сторону с этой же скоростью до тех пор пока сигнал от источника stop after в активном состоянии и продолжит движенние на половину оборота в этом же направлении. Затем начинается точная доводка в заданном направлении 1st move direction (вправо или влево) со скоростью 1st move speed. Эта скорость обычно выбирается достаточно низкой, чтобы "не пропустить" приход сигнала. Мотор останавливается в зависимости от значения stop after при поступлении сигнала со входа синхронизации, сигнала от датчика оборотов или при достижении концевого выключателя.

При установленном флаге Use second phase of homing алгоритм поиска исходного положения вращает мотор в заданном направлении 2nd move direction (вправо или влево) со скоростью 2nd move speed.

При установленном флаге Blind half turn мотор игнорирует сигнал об окончании второй фазы движения в течение половины оборота. Это сделано для того чтобы в случае, когда датчики, по которым происходит окончание первой и второй фазы движения, расположены достаточно близко друг от друга, можно было задать однозначный порядок их обнаружения.

Третьей фазой поиска исходной позиции является безусловное смещение на дистанцию standoff.

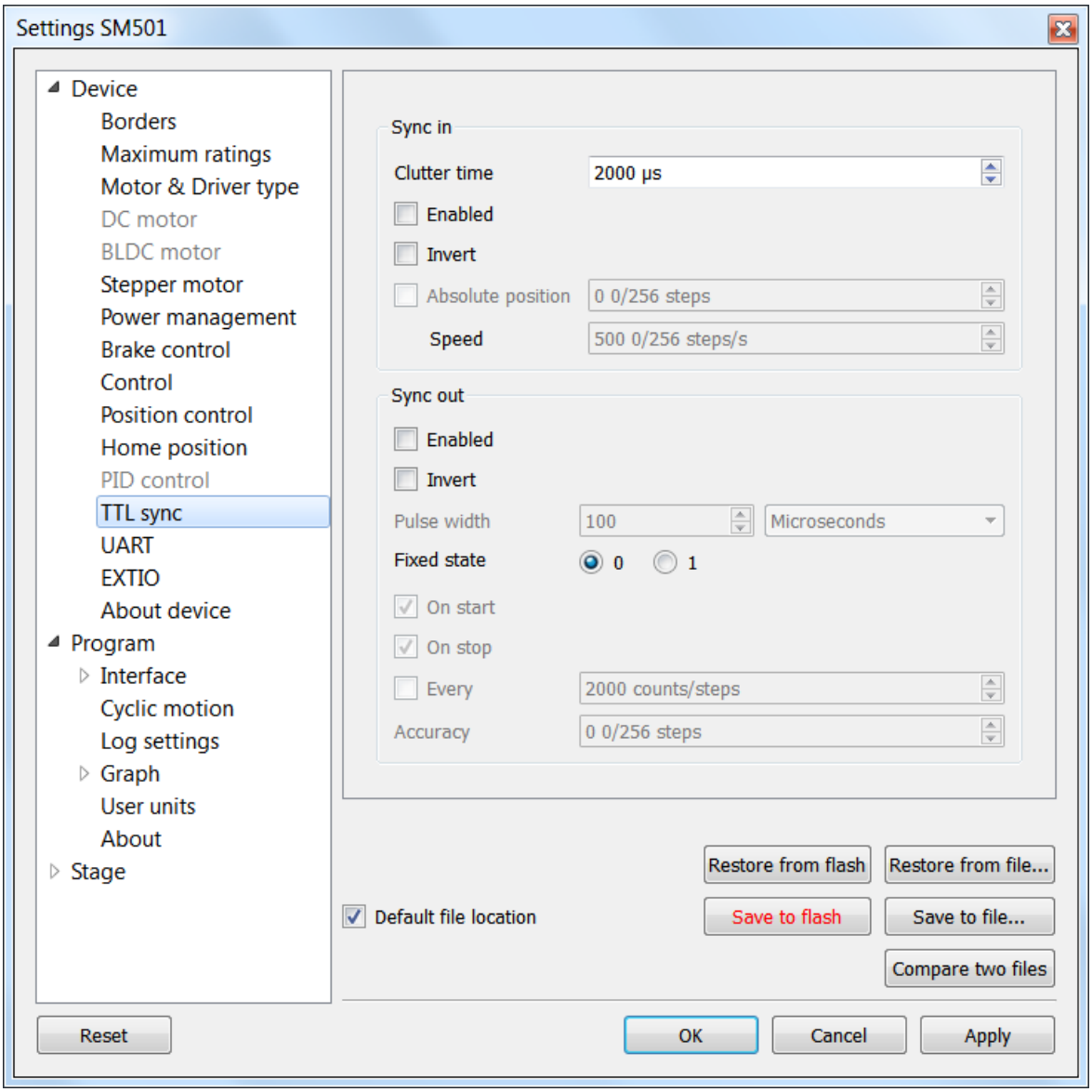
Полученная точка называется исходным положением. Важно что ее расположение на позиционере не зависит от начального положения из которого началась калибровка.

При включении опции Use fast homing algorithm первое движение к концевику происходит со скоростью Working speed (см. страницу Настройка кинематики движения (Шаговый двигатель)) для ускорения процесса поиска исходного положения.

Команды настройки описаны в разделе Описание протокола обмена.

5.3.6. Настройки синхронизации

В окне настроек программы ***Device configuration -> TTL sync***



Вкладка "Настройки синхронизации"

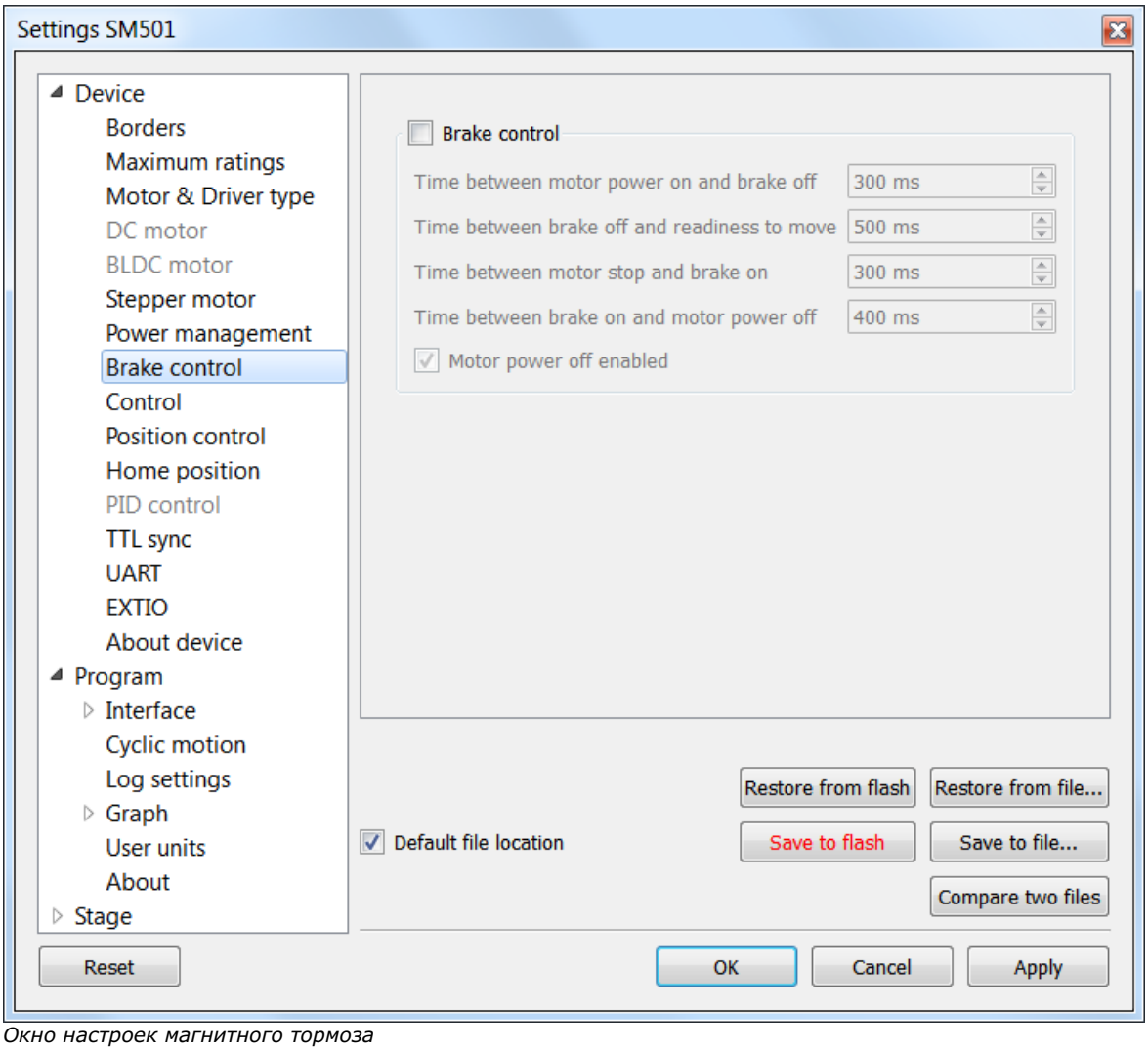
Подробно работа синхронизации описана в разделе ТТЛ-синхронизация.

Sync in
Clutter time - настройка минимальной длительности импульса синхронизации (в микросекундах). Определяет минимальную длительность входного синхроимпульса, который может быть зарегистрирован (защита от дребезга).
Enabled - включает работу в режиме входа.
Invert - установленный флаг соответствует срабатыванию по заднему фронту синхроимпульса.
Absolute position - по приходу синхроимпульса при отмеченной настройке смещается в абсолютную координату, задаваемую полем шаг/микрошаг, при снятой настройке осуществляет относительное смещение на задаваемое расстояние.
Speed - определяет скорость с которой производится движение по приходу импульса синхронизации.

Sync out
Выход синхронизации может использоваться как «выходной сигнал общего назначения».
Enabled - если флаг установлен, то синхронизация выхода работает согласно настройкам. При снятом флаге значение выхода фиксировано и равно Fixed state.
Invert - если флаг установлен, то нулевой логический уровень является активным.
Pulse width - задает длину выходного импульса в миллисекундах или шагах/импульсах энкодера.
Fixed state - устанавливает логический уровень выхода в 0 или 1 соответственно.
On start - синхронизирующий импульс генерируется в начале движения.
On stop - синхронизирующий импульс генерируется при окончании движения.
Every - синхронизирующий импульс генерируется каждые n импульсов энкодера.
Accuracy - окрестность вокруг целевой координаты, попадание в которую считается попаданием в целевую позицию и вызывает генерацию импульса по остановке.

5.3.7. Настройка тормоза

В окне настроек программы ***Device configuration -> Brake control***



Окно настроек магнитного тормоза

Для включения использования магнитного тормоза необходимо установить флаг Brake control.

Параметры:

Time between motor power on and brake off - Время между включением питания мотора и отключением тормоза (мс).

Time between brake off and readiness to move - Время между отключением тормоза и готовностью к движению (мс). Все команды движения начинают выполняться только по истечении этого времени.

Time between motor stop and brake on - Время между остановкой мотора и включением тормоза (мс).

Time between brake on and motor power off - Время между включением тормоза и отключением питания (мс).

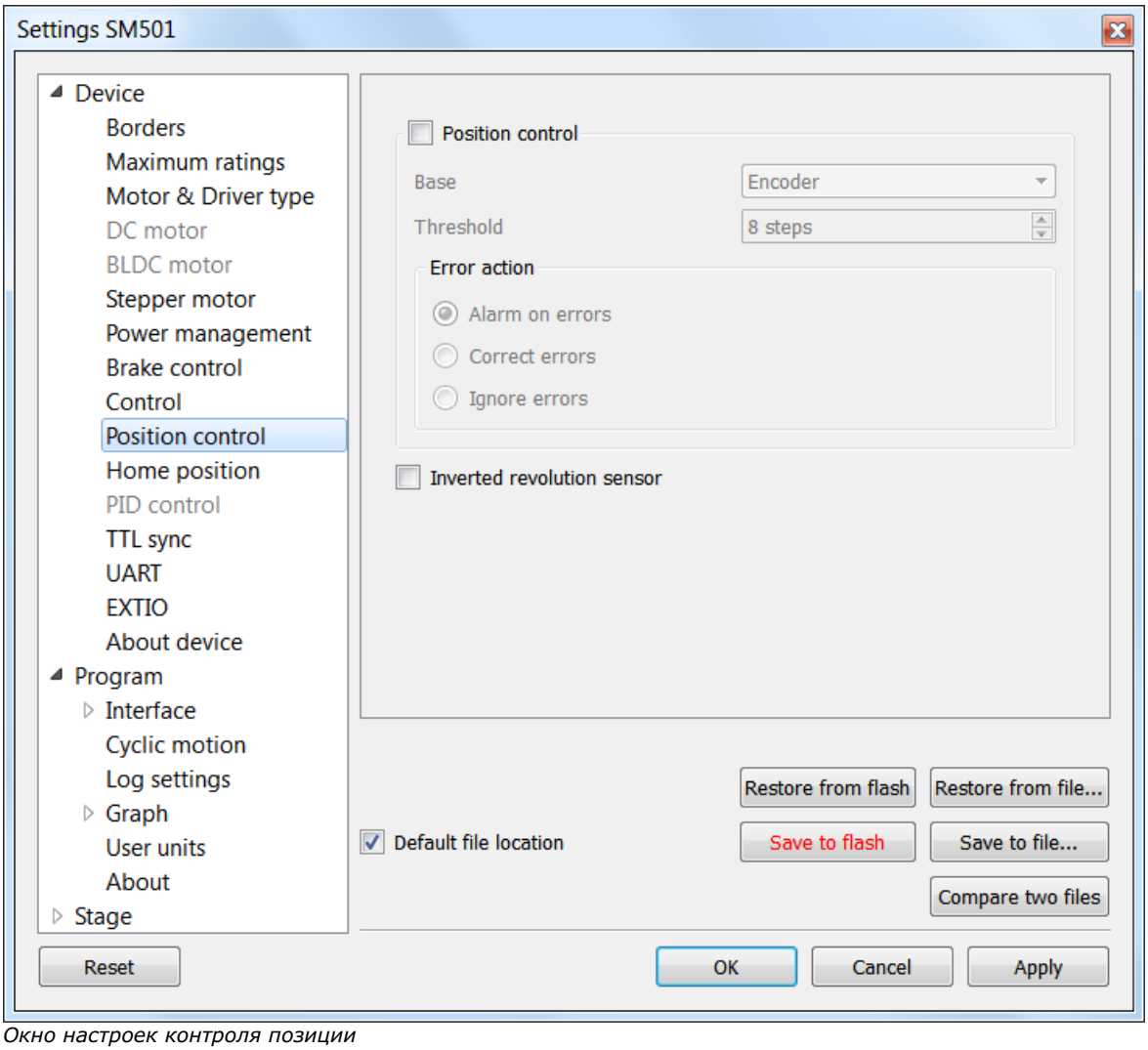
Диапазон значений: от 0 до 65535 мс.

Флаг Motor power off enabled означает, что при снятии питания с магнитного тормоза, тормоз отключает питание мотора.

Команды настройки описаны в разделе "Описание протокола обмена".

5.3.8. Контроль позиции

В окне настроек программы ***Device configuration -> Position control***



Окно настроек контроля позиции

Для активации контроля позиции необходимо установить флаг параметра Position control.

Base - выбор устройства контроля позиции. В выпадающем окне выбирается: энкодер (*Encoder*) (см. раздел "Работа с энкодерами") или датчик оборотов (*Revolution sensor*).

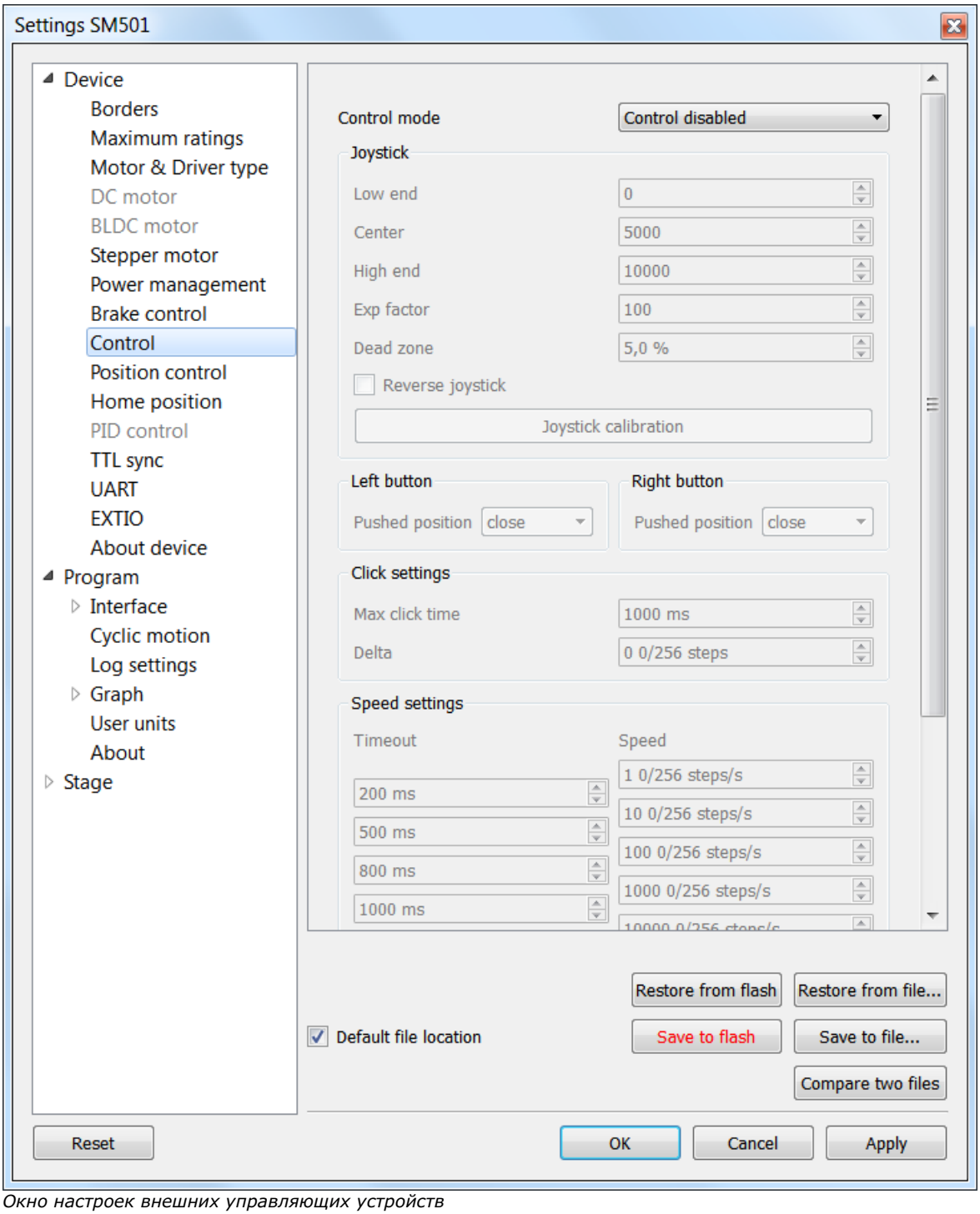
Minimal error - определяет количество потерянных шагов (0..255), которое считается ошибочным. Если количество потерь превышает заданное число шагов, то устанавливается флаг рассогласования SLIP. Дальнейшие действия зависят от настройки Error action:
Если установлена опция Alarm on errors, то контроллер перейдет в состояние Alarm.
Если установлена опция Correct errors, то контроллер попытается скорректировать ошибку проскальзывания дополнительным движением (см. раздел Обнаружение потери шагов).
Если установлена опция Ignore errors, то контроллер не будет производить никаких дополнительных действий.

Inverted revolution sensor - при отмеченном флаге датчик оборотов считается сработавшим по уровню 1, при неотмеченном действует обычная логика: 0 - это срабатывание/активация/активное состояние.

Команды настройки описаны в разделе Описание протокола обмена.

5.3.9. Настройка внешних управляющих устройств

В окне настроек программы ***Device configuration -> Control***



Окно настроек внешних управляющих устройств

Control mode - выбор внешних устройств для управления двигателем.

- Control disabled - внешние устройства не используются
- Joystick - используется джойстик
- Buttons - используются кнопки

В блоке ***Joystick*** содержатся настройки джойстика.

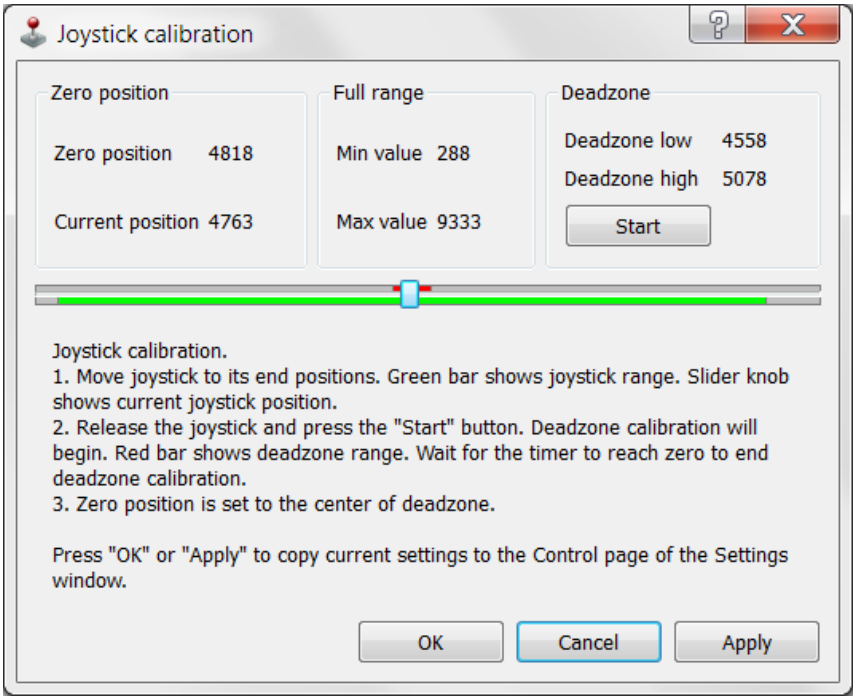
Low end, Center и High end определяют нижнюю границу, середину и верхнюю границу диапазона джойстика соответственно. То есть, нормированное значение АЦП джойстика равное или меньше Low end соответствует максимальному отклонению джойстика в сторону меньших значений.

Exp factor - параметр экспоненциальной нелинейности. См. Управление с помощью джойстика.

Dead zone - зона нечувствительности к отклонению джойстика от центрального положения. Минимальный шаг изменения: 0.1%, максимальное значение: 25.5%. Отклонению джойстика от положения Center на величину меньшую Dead zone соответствует нулевая скорость.

Reverse joystick - Реверс воздействия джойстика. Отклонение джойстика к большим значениям приводит к отрицательной скорости и наоборот.

Кнопка Joystick calibration открывает диалог калибровки джойстика.



Диалог калибровки джойстика.

Калибровка сводится к автоматическому нахождению границ и зоны нечувствительности. Она происходит следующим образом: Переместите джойстик в крайние положения - это позволит найти границы. Диапазон всех измеренных значений визуализируется зеленой линией.

Отпустите джойстик и нажмите кнопку Start - включится обнаружение зоны нечувствительности. В течение следующих 5 секунд имитируйте случайные воздействия на джойстик, которые не должны быть распознаны как смещение джойстика из нулевого положения. Диапазон зоны нечувствительности визуализируется красной линией.

Нажатие кнопки Apply передаст вычисленные значения в окно настроек, а нажатие OK передаст значения и закроет диалог калибровки.

Блоки ***Left button*** и ***Right button*** отвечают за настройку кнопок.

Pushed Position - определяет при каком состоянии (нажата кнопка или нет) подается сигнал на движение в контроллер.

- open - отжатая кнопка считается командой движения.
- close - нажатая кнопка считается командой движения.

В блоке ***Click settings*** настраивается "клик" кнопок. Нажатие кнопки на краткое время интерпретируется как клик.

Max click time - Максимальное время клика. До истечения этого времени первая скорость не включается.

Delta - Смещение (дельта) позиции. Контроллер смещается на это расстояние при каждом клике.

Блок ***Speed settings*** содержит настройки таймаутов и скоростей движения.

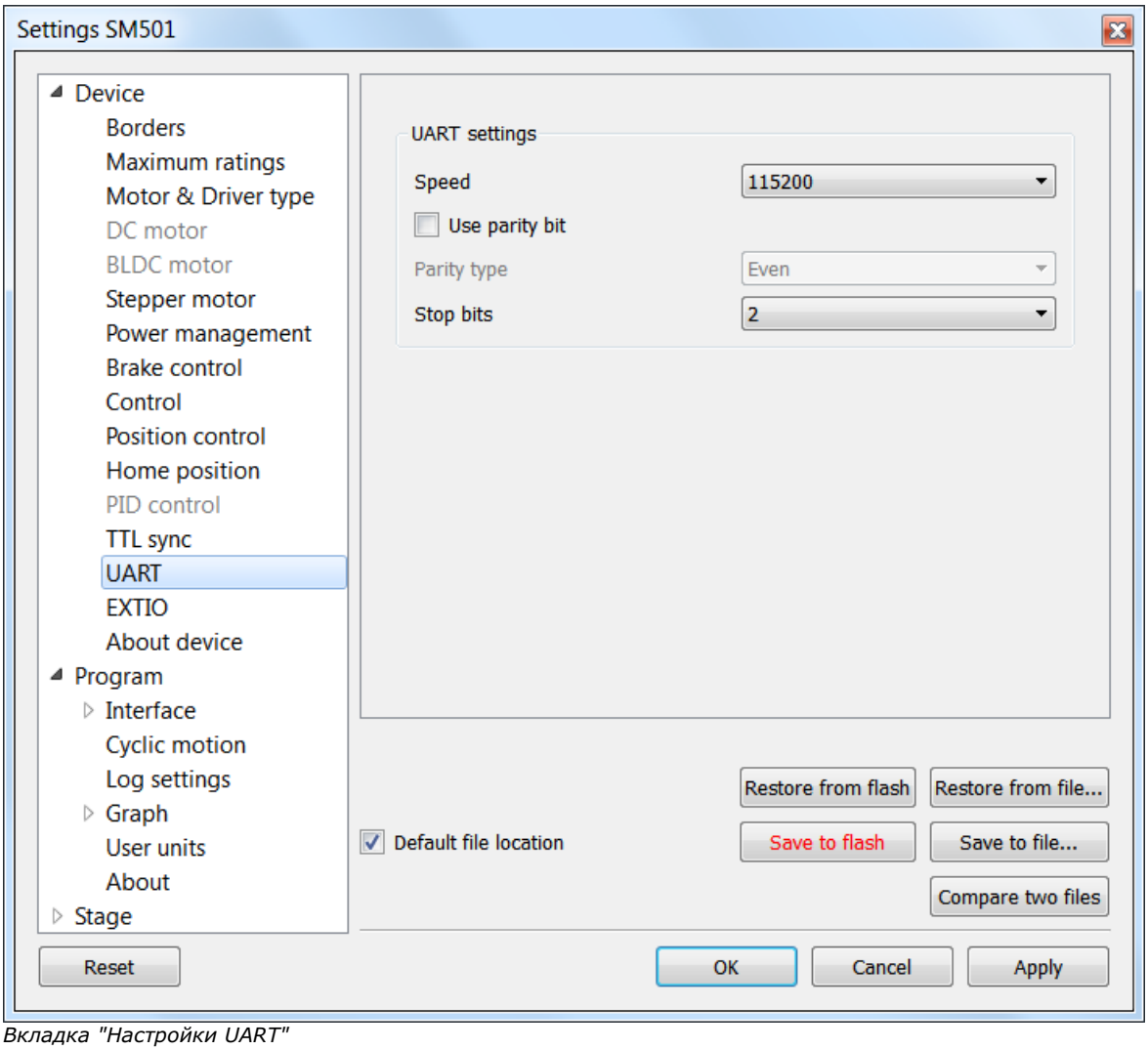
Timeout[i] - время, по истечении которого скорость переключается со Speed[i] на Speed[i+1]. Если какой-либо из Timeout[i] равен нулю, то переключение на последующие скорости происходить не будет.

Speed[i] - скорость, на которой должен работать мотор после времени Timeout[i-1]. Если какая-либо скорость равна нулю, то переключение на эту и последующие скорости происходить не будет.

Команды настройки описаны в разделе Описание протокола обмена.

5.3.10. Настройки UART

В окне настроек программы ***Device configuration -> UART***



Вкладка "Настройки UART"

Speed - выбор скорости UART из нескольких предустановленных значений в диапазоне от 9600 бит/с до 921600 бит/с.

Use parity bit - использование бита четности.

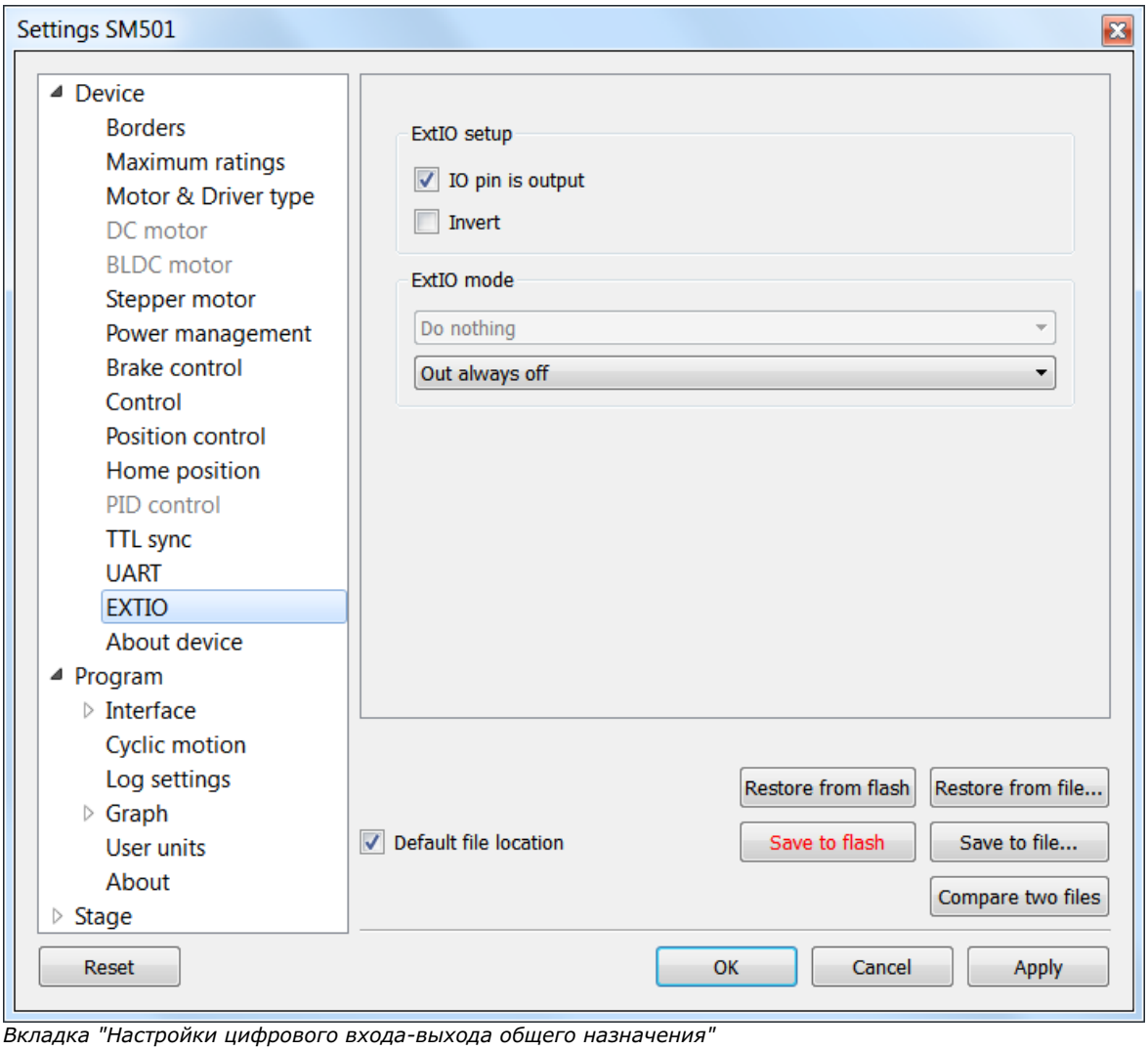
Parity type - Тип четности:

- Even - бит установлен при нечетном количестве бит
- Odd - бит установлен при четном количестве бит
- Space - бит четности всегда 0
- Mark - бит четности всегда 1

Stop bits - количество стоповых бит (1 или 2).

5.3.11. Настройки цифрового входа-выхода общего назначения

В окне настроек программы ***Device configuration -> EXTIO settings***



Вкладка "Настройки цифрового входа-выхода общего назначения"

Подробное описание в разделе Цифровой вход-выход общего назначения.

ExtIO setup
IO pin is output - если флаг установлен, то ножка ExtIO работает в режиме выхода, иначе в режиме входа.
Invert - если флаг установлен, то переход в нулевой логический уровень приводит к выполнению действия.

ExtIO mode - выбор режима работы
Если ExtIO сконфигурирован в режиме входа, то активен выбор настроек действия контроллера по входному импульсу:

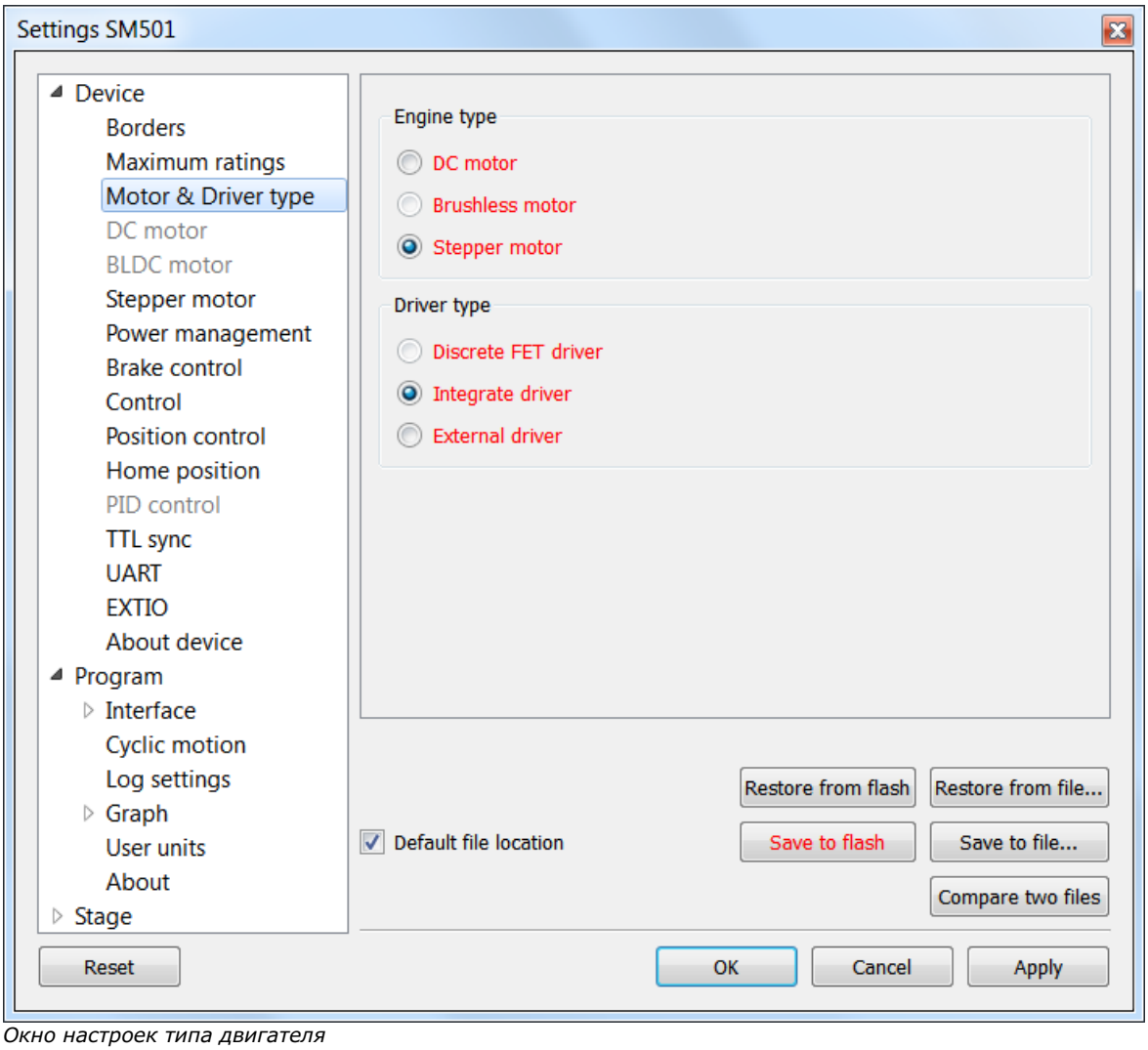
- Do nothing - ничего не делать.
- Stop on input - выполнить команду STOP.
- Power off on input - выполнить команду PWOF.
- Movr on input - выполнить команду MOVR.
- Home on input - выполнить команду HOME.
- Alarm on input - войти в состояние ALARM.

Если ExtIO сконфигурирован в режиме выхода, то активен выбор состояния выхода в зависимости от состояния контроллера:

- Out always off - всегда в неактивном состоянии.
- Out always on - всегда в активном состоянии.
- Out active when moving - в активном состоянии в процессе движения.
- Out active in Alarm - в активном состоянии, если контроллер в состоянии Alarm.
- Out active when motor is on - в активном состоянии, если запитаны обмотки мотора.
- Out active when motor is found - в активном состоянии, если подключен мотор.

5.3.12. Настройка типа двигателя

В окне настроек программы ***Device configuration -> Motor type***



Окно настроек типа двигателя

Индикация типа двигателя - шаговый Stepper motor, либо двигатель постоянного тока DC-motor. Также выбирается силовой драйвер для управления:

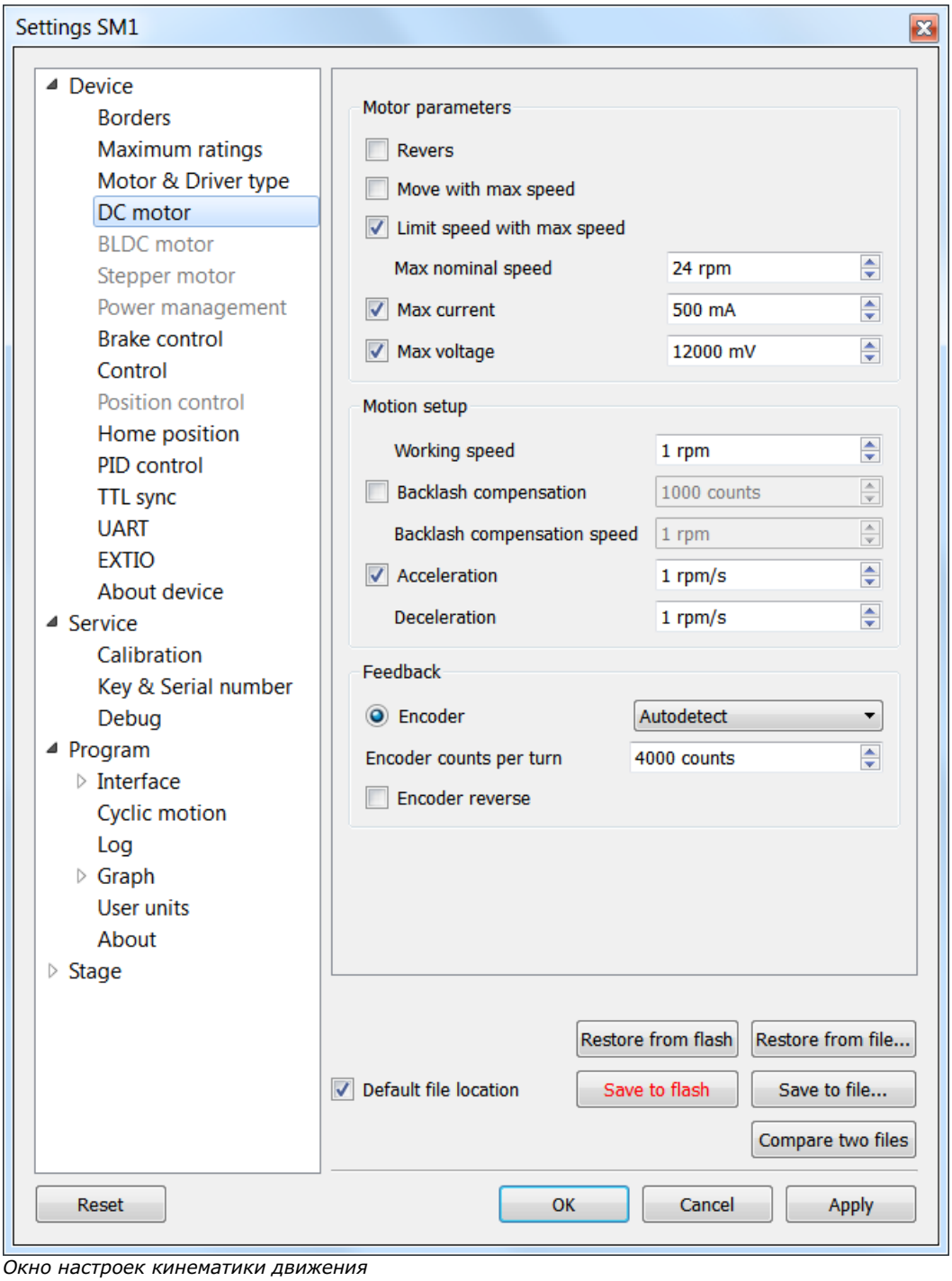
- Интегрированный. Используется в данной модификации контроллера.
- На дискретных ключах. Будет использоваться в будущих версиях.
- Внешний драйвер. Для управления шаговыми двигателями с помощью трёх стандартных сигналов (см. Интерфейс управления внешним драйвером)

Предупреждение. Смена типа драйвера или типа мотора это критическая операция, которую не следует выполнять в момент вращения двигателя. Для корректной смены нужно обесточить обмотки текущего двигателя, отключить его, изменить тип двигателя, подключить двигатель другого типа. То же самое касается и смены типа драйвера с интегрированного на внешний и наоборот.

Замечание. Доступные типы двигателей определяются используемой прошивкой. Доступные драйвера управления определяются типом платы контроллера, за исключением внешнего драйвера.

5.3.13. Настройка кинематики движения (DC мотор)

В окне настроек программы ***Device configuration -> DC Motor***



Окно настроек кинематики движения

Motor parameters - настройки электромотора

Revers - установка этого флага позволяет связать направление вращения мотора с направлением счета текущей позиции. Измените состояние флага, если положительное вращение мотора уменьшает счетчик позиции. Действие этого флага равносильно подключению обмотки мотора в обратной полярности.

Move with max speed - при установленном флаге мотор игнорирует заданную скорость и вращается с максимальной допустимой скоростью.

Limit speed with max speed - при установленном флаге контроллер ограничивает максимальную скорость по количеству оборотов в секунду значением поля Max nominal speed.

Max nominal speed, Max voltage, Max current - номинальные параметры мотора. Если они активны и применимы для данного типа двигателя, то контроллер ограничивает эти параметры в заданных рамках. Например, если скорость и напряжение на моторе превысили номинальные, контроллер будет снижать выходное воздействие, пока оба значения не будут в пределах нормы. Однако при этом контроллер останется в рабочем состоянии, будет выполнять текущую задачу.

Motion setup - настройки кинематики движения

Working speed - скорость движения.

Backlash compensation - компенсация люфта. Так как механика позиционера не идеальна, существует различие при подходе к заданной точке справа и слева. При включенном режиме компенсации люфта позиционер будет подходить к точке всегда с одной стороны. Установленное значение определяет количество шагов, на которое позиционер будет проходить заданную точку чтобы возвращаться к ней с одной и той же стороны. Если указанное число больше нуля, позиционер будет подходить к точке всегда справа. Если меньше нуля, то всегда слева.

Backlash compensation speed - скорость компенсации люфта. При включенном режиме компенсации люфта Backlash compensation позиционер будет подходить к точке справа или слева с установленной скоростью, определяемой количеством шагов в секунду.

Acceleration - включает режим движения с ускорением, числовое значение поля - величина ускорения движения.

Deceleration - величина замедления движения.

Feedback - настройки обратной связи

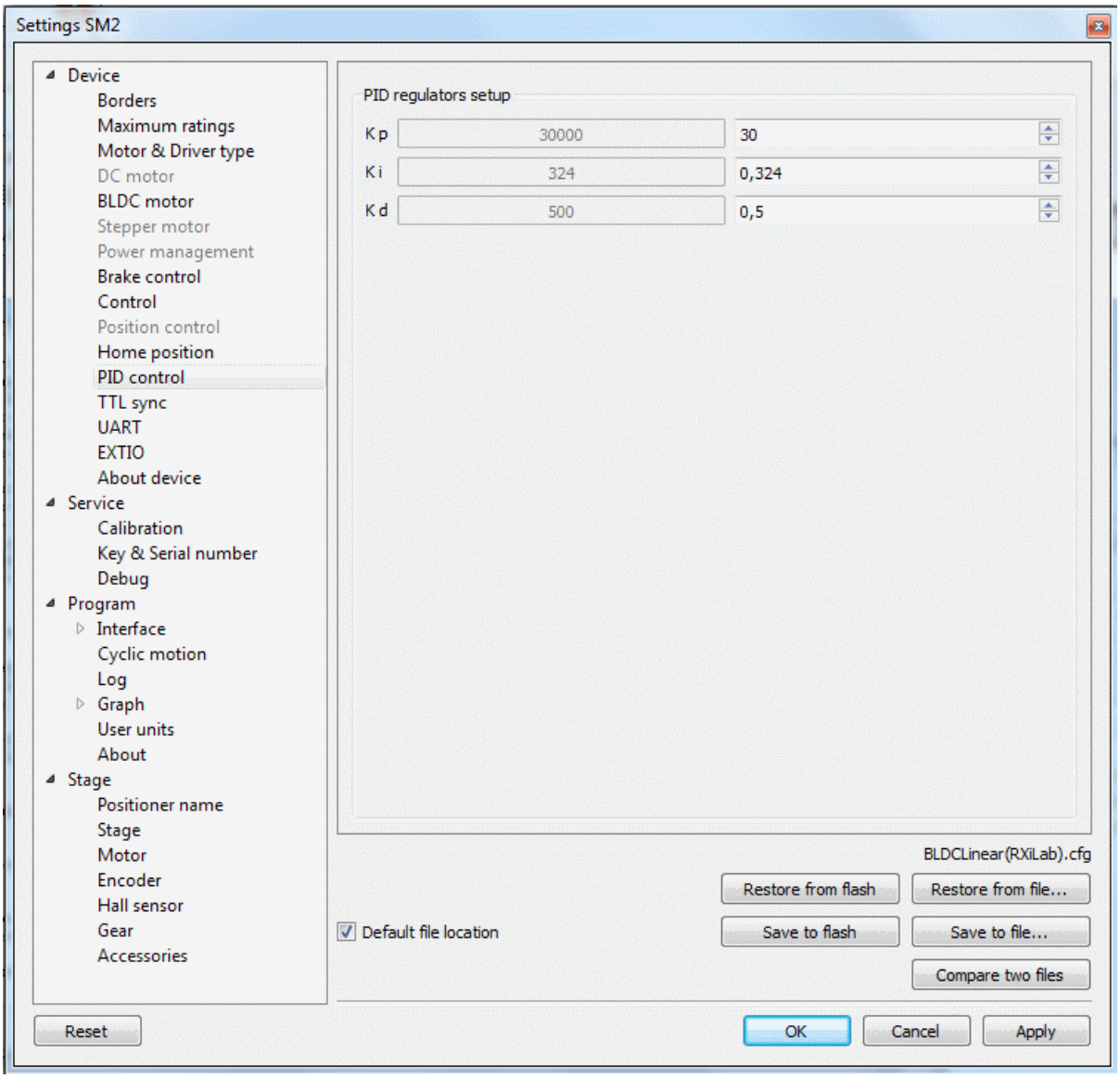
Encoder - использование энкодера в качестве датчика обратной связи. При включении данной опции доступен выбор типа энкодера: недифференциальный, дифференциальный или автоматическое определение.

Encoder counts per turn - количество импульсов энкодера на один полный оборот оси мотора.

Encoder reverse - реверс энкодера

5.3.14. Настройка контуров ПИД-регулирования

В окне настроек программы ***Device configuration -> PID control***



Окно настройки контуров ПИД-регулирования

В этом разделе вы можете изменить коэффициенты ПИД-регуляторов.
Используется регулятор по напряжению, 3 коэффициента K_p, K_i и K_d могут изменяться в диапазоне 0..65535.

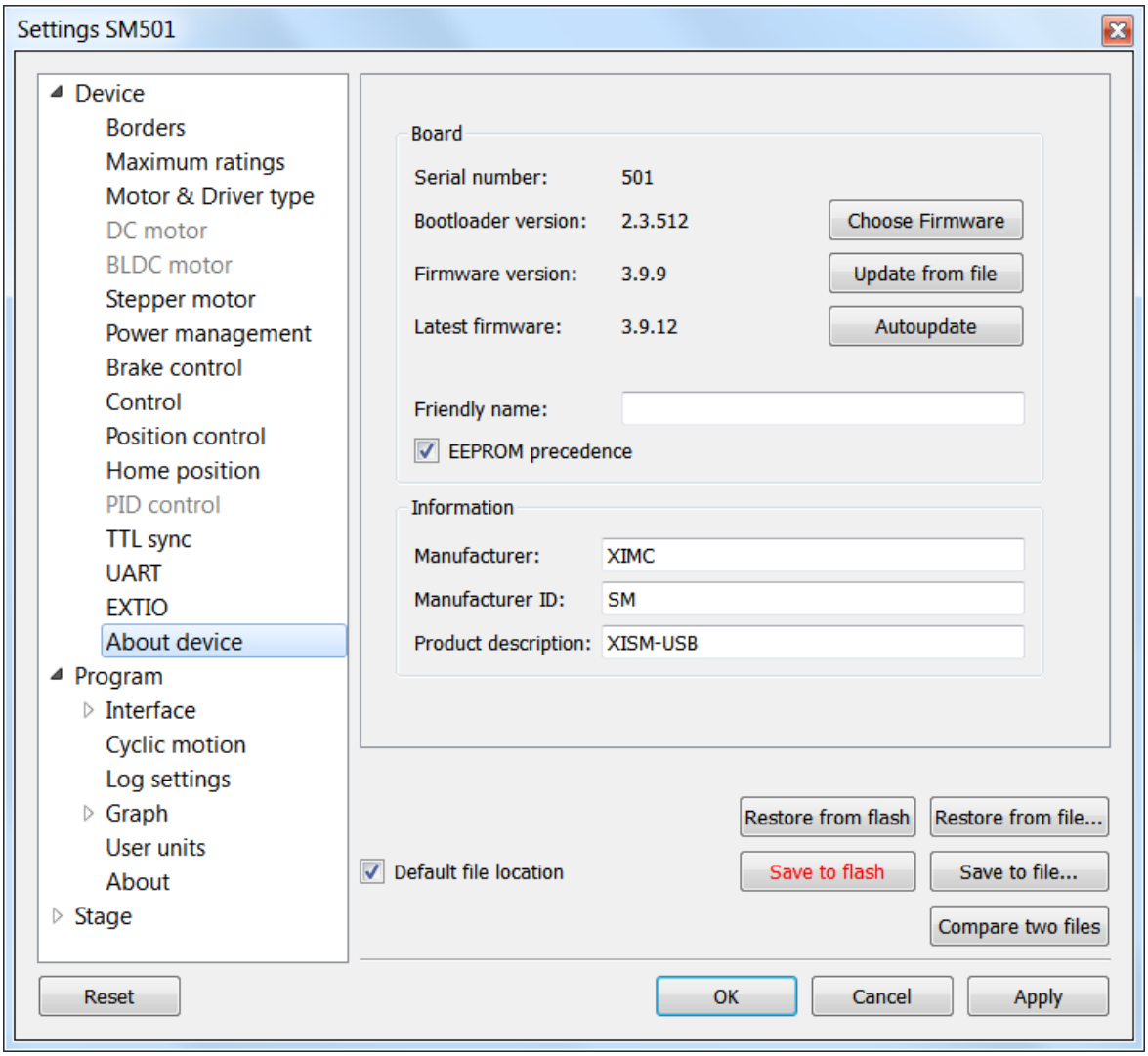
Предупреждение. Не меняйте настройки ПИД-регуляторов, если вы не уверены, что знаете, что делаете!

Поля коэффициентов ПИД с дробной частью используются только для управления BLDC двигателем (поддержка начиная с прошивки 4.1.x). В зависимости от выбранного типа двигателя (DC или BLDC) соответствующие поля становятся активными.

Команды настройки описаны в разделе Описание протокола обмена. Настройка ПИД-регуляторов описаны в разделе ПИД алгоритм для управления DC двигателем для DC и ПИД алгоритм для управления BLDC двигателем для BLDC.

5.3.15. О контроллере

В окне настроек программы ***Device configuration -> About device***



Вкладка "Об устройстве"

В разделе **Board** отображается информация о контроллере:

Serial number - серийный номер устройства.

Bootloader version - версия загрузчика.

Firmware version - версия прошивки.

Latest firmware - последняя доступная версия прошивки для данного контроллера (считывается из интернет при наличии интернет-соединения).

Кнопка Update from file открывает диалог обновления прошивки из указанного пользователем файла.
Выберите файл прошивки, имеющий расширение .cod и нажмите "Open". XILab начнет обновление прошивки, в окне появится надпись "Please wait while firmware is updating". Не выключайте питание контроллера во время обновления. По окончании обновления будет выведен диалог "Firmware updated successfully".

Кнопка Choose Firmware открывает диалог выбора версии прошивки на которую будет произведено обновление. Список версий и сами файлы прошивок скачиваются автоматически из интернет. Для работы этого типа обновления необходимо наличие активного интернет-подключения. В диалоговом окне выберите необходимую версию прошивки и нажмите "Update firmware".

Кнопка Autoupdate запускает обновление прошивки из интернет на последнюю доступную версию.

Friendly name - позволяет установить произвольное имя контроллера. Если значение имени непусто, то эта строка будет выводиться в заголовках окон вместо идентификатора и серийного номера устройства. Это имя удобно использовать если к компьютеру подключено несколько контроллеров.

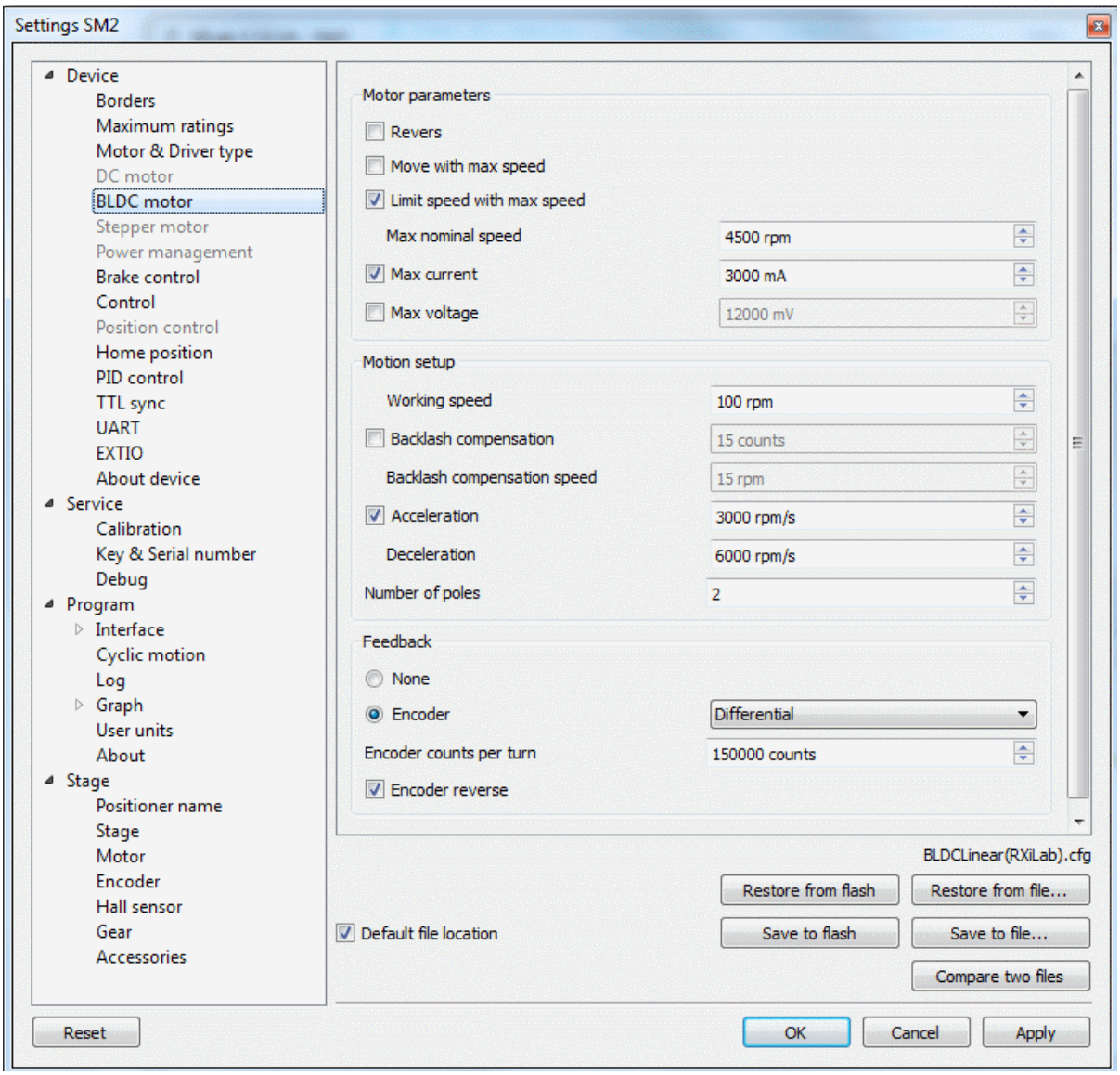
EEPROM precedence - этот флаг действует только для позиционеров с системой опознавания. При включенном флаге для контроллера становятся приоритетными настройки из внешней EEPROM-памяти позиционера и они применяются каждую перезагрузку контроллера или каждое подключение позиционера. В ином случае используются настройки из FRAM. Включение настройки окрашивает кнопку "Save to flash" в красный цвет - это предупреждает пользователя, что сохраненные во флеш-память контроллера настройки будут перезаписаны при подключении внешней памяти.

В разделе **Information** отображаются данные об устройстве: производитель, идентификатор устройства, тип устройства. Эти данные считываются из внутренней памяти контроллера.

Все эти данные сообщаются программе XILab при подключении устройства.

5.3.16. Настройка кинематики движения (BLDC мотор)

В окне настроек программы ***Device configuration -> BLDC Motor***



Окно настроек кинематики движения

ВАЖНО. Только версии прошивки 4.1.x и старше поддерживают работу с BLDC двигателем.

Motor parameters - настройки электромотора

Revers - установка этого флага позволяет связать направление вращения мотора с направлением счета текущей позиции. Измените состояние флага, если положительное вращение мотора уменьшает счетчик позиции. Действие этого флага равносильно подключению обмотки мотора в обратной полярности.

Move with max speed - при установленном флаге мотор игнорирует заданную скорость и вращается с максимальной допустимой скоростью.

Limit speed with max speed - при установленном флаге контроллер ограничивает максимальную скорость по количеству оборотов в секунду значением поля Max nominal speed.

Max nominal speed, Max voltage, Max current - номинальные параметры мотора. Если они активны и применимы для данного типа двигателя, то контроллер ограничивает эти параметры в заданных рамках. Например, если скорость и ток на моторе превысили номинальные, контроллер будет снижать выходное воздействие, пока оба значения не будут в пределах нормы. Однако при этом контроллер останется в рабочем состоянии, будет выполнять текущую задачу.

ВАЖНО. Контроллер BLDC всегда учитывает MaxCurrent и не учитывает MaxVoltage, т.к. для этих типов двигателей основной характеристикой является номинальный ток, а максимальное напряжение обычно сильно превосходит поддерживаемое контроллером двигателей.

Motion setup - настройки кинематики движения

Working speed - скорость движения.

Backlash compensation - компенсация люфта. Так как механика позиционера не идеальна, существует различие при подходе к заданной точке справа и слева. При включенном режиме компенсации люфта позиционер будет подходить к точке всегда с одной стороны. Установленное значение определяет количество пульсов энкодера, на которое позиционер будет проходить заданную точку чтобы возвращаться к ней с одной и той же стороны. Если указанное число больше нуля, позиционер будет подходить к точке всегда справа. Если меньше нуля, то всегда слева.

Backlash compensation speed - скорость компенсации люфта. При включенном режиме компенсации люфта Backlash compensation позиционер будет подходить к точке справа или слева с установленной скоростью, определяемой количеством оборотов в минуту.

Acceleration - включает режим движения с ускорением, числовое значение поля - величина ускорения движения.

Deceleration - величина замедления движения.

Number of poles - количество полюсов BLDC двигателя на один оборот

Feedback - настройки обратной связи

Encoder - использование энкодера в качестве датчика обратной связи. При включении данной опции доступен выбор типа энкодера: недифференциальный, дифференциальный или автоматическое определение.

Encoder counts per turn - количество импульсов энкодера на один полный оборот оси мотора.

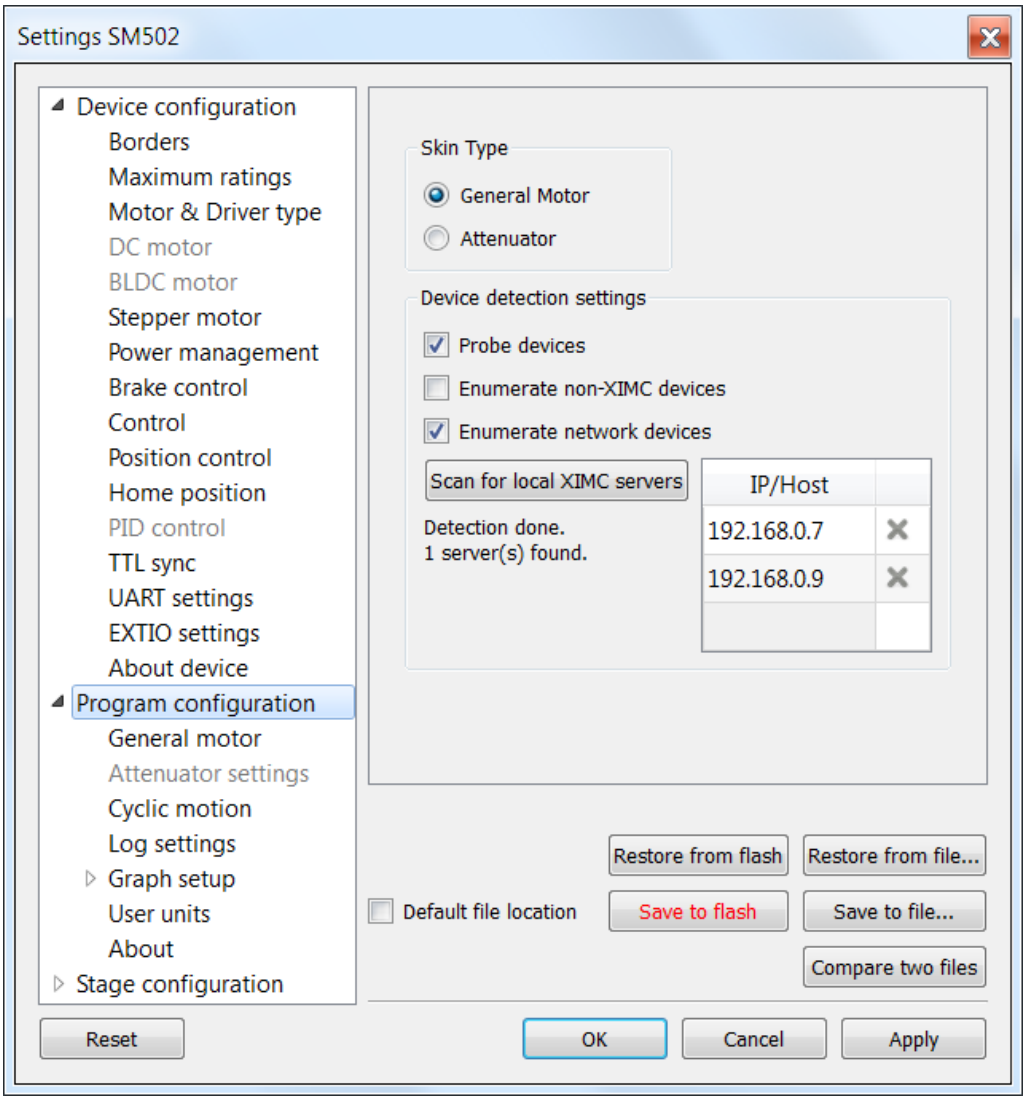
Encoder reverse - реверс энкодера

5.4. Настройки программы XILab

1. Общие настройки программы XILab
2. Настройки интерфейса абстрактного позиционера
3. Настройки интерфейса аттенюатора
4. Настройка режима циклического движения
5. Настройка логирования
6. Общие настройки отображения графиков
7. Индивидуальные настройки отображения графиков
8. Настройки отображения пользовательских единиц
9. О программе

5.4.1. Общие настройки программы XILab

В окне настроек программы ***Program configuration***



Вкладка общих настроек программы

Данная вкладка отвечает за выбор типа интерфейса Xilab и настройку обнаружения устройств XIMC.

Группа **Skin Type** содержит настройку типа интерфейса Xilab. Доступны два варианта интерфейса: "General Motor" и "Attenuator".

General Motor - включает интерфейс мотора с абстрактным позиционером. В этом режиме в главном окне текущая позиция отображается численно и графически слайдером, также доступны элементы управления движением в произвольную координату. Настройки отображения интерфейса в этом режиме находятся на вкладке General Motor.

Attenuator - включает интерфейс аттенюатора. В этом режиме в главном окне текущая позиция отображается графически, также доступны элементы управления движением в позицию с определенной прозрачностью фильтров аттенюатора. Настройки отображения интерфейса в этом режиме находятся на вкладке Attenuator settings.

Группа **Device detection settings** содержит настройки обнаружения устройств XIMC.

Probe devices - при включенной опции пытается идентифицировать контроллеры, посылая в них при открытии команды GETI и GSER.

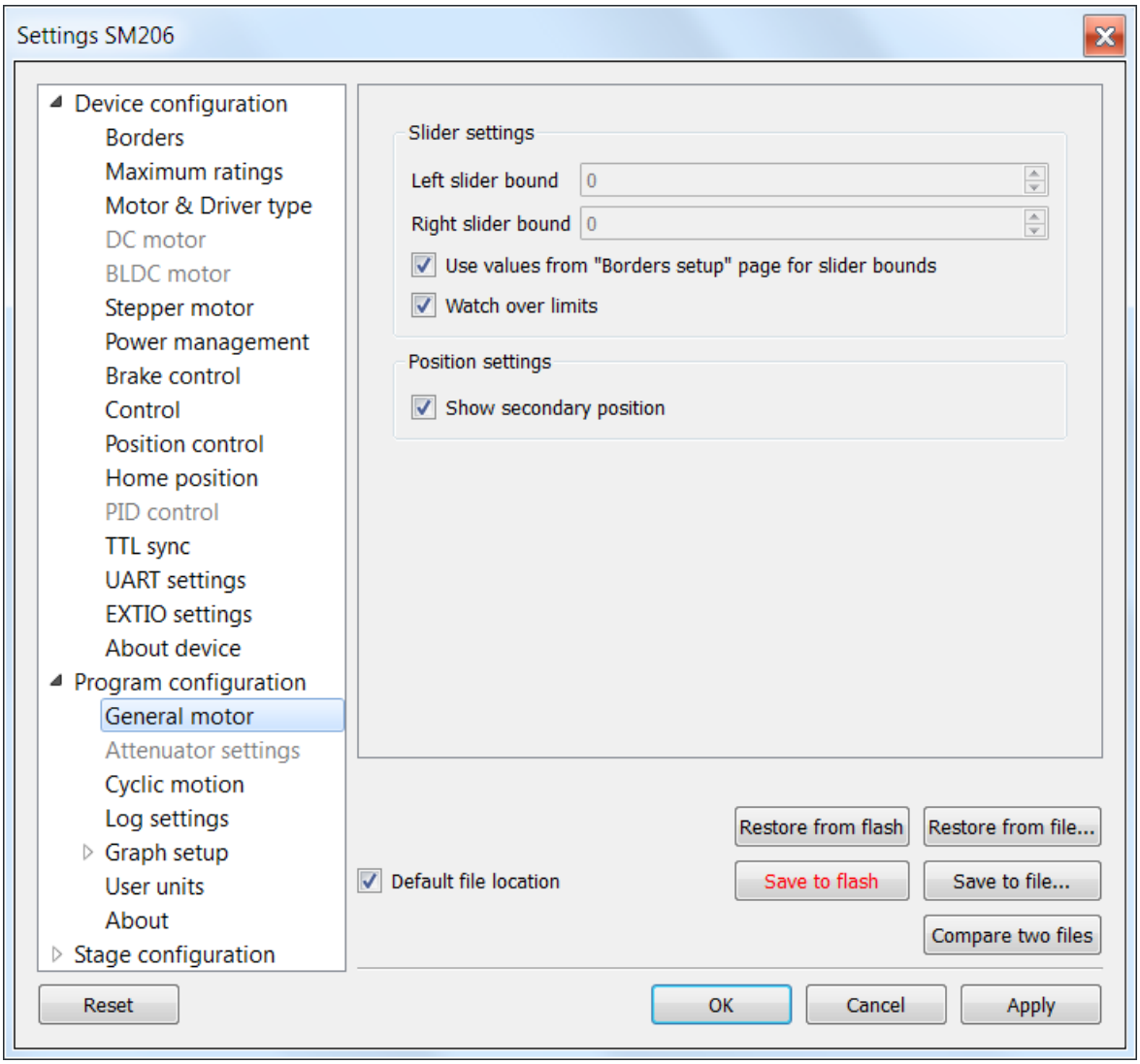
Enumerate non-XIMC devices - при включенной опции опрашивает все устройства типа COM-порт в системе. При отключенной опции опрашивает только устройства, имена которых соответствуют маске устройств XIMC ("XIMC Motor Controller" в Windows, /dev/ximc/* и /dev/ttyACM* на Linux/Mac).

Enumerate network devices - при включенной опции опрашивает сетевые устройства. Список адресов доменных имен и/или IP-адресов, на которых производится поиск устройств, находится ниже. Записи в списке можно добавлять как вручную, так и автоматически, нажав на кнопку Scan for local XIMC servers. Обратите внимание, что в случае наличия нескольких XIMC-серверов с устройствами в локальной сети будет найден случайный из них и для нахождения всех серверов потребуется несколько попыток автоматического поиска.

Предупреждение. При одновременно включенных опциях Probe devices и Enumerate non-XIMC devices XILab при старте посылает данные во все COM-порты.
При наличии в системе множества Bluetooth COM-портов, из-за особенностей работы Bluetooth опрос будет происходить последовательно, с затратами от единиц до десятков секунд на одну попытку соединения.

5.4.2. Настройки интерфейса абстрактного позиционера

В окне настроек программы ***Program configuration -> General motor***



Окно вкладки "Настройки абстрактного позиционера"

Данная вкладка предназначена для настройки интерфейса Xilab при работе с абстрактным позиционером без указания его типа. Вкладка становится активной при выборе опции "General motor" в блоке "Skin Type" в окне вкладки "Program configuration". Во вкладке настраивается отображение слайдера и второстепенной позиции.

Слайдер позиции находится в главном окне и визуально представляет текущую позицию позиционера относительно границ. Настройка сводится к установке отображаемых границ и настройке поведения границ слайдера при выходе последнего из отображаемого диапазона.



Фрагмент основного окна программы со слайдером

Группа **Slider settings** содержит настройки слайдера.

Left slider bound и Right slider bound содержат левую и правую границу слайдера соответственно.

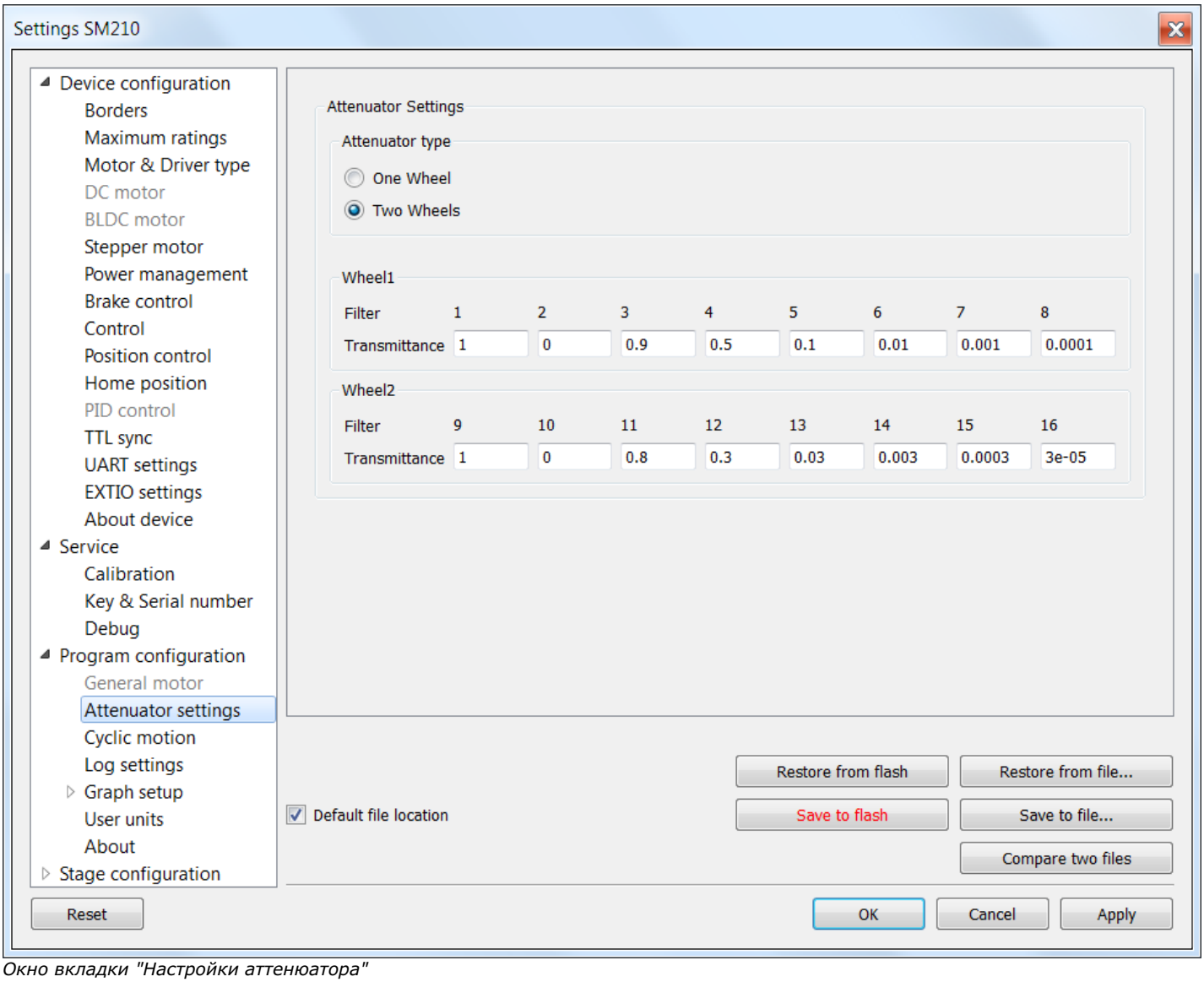
Флажок Watch over limits настраивает такое поведение границ слайдера при котором при выходе текущей позиции за диапазон слайдера, шкала начинает смещаться, чтобы отобразить текущую позицию. Общая отображаемая на слайдере дистанция при этом остается неизменной. Не используется по умолчанию. Данный флажок удобен когда вы знаете диапазон перемещения позиционера, но не знаете привязку этого положения к значениям, отображаемым в XILab, например до проведения калибровки. Часто используется совместно с настройками из вкладки Настройка исходного положения.

Группа **Position settings** содержит настройки отображения второстепенной позиции.

Флажок Show secondary position включает отображение второстепенной позиции по энкодеру в главном окне программы.

5.4.3. Настройки интерфейса аттенюатора

В окне настроек программы ***Program configuration -> Attenuator settings***



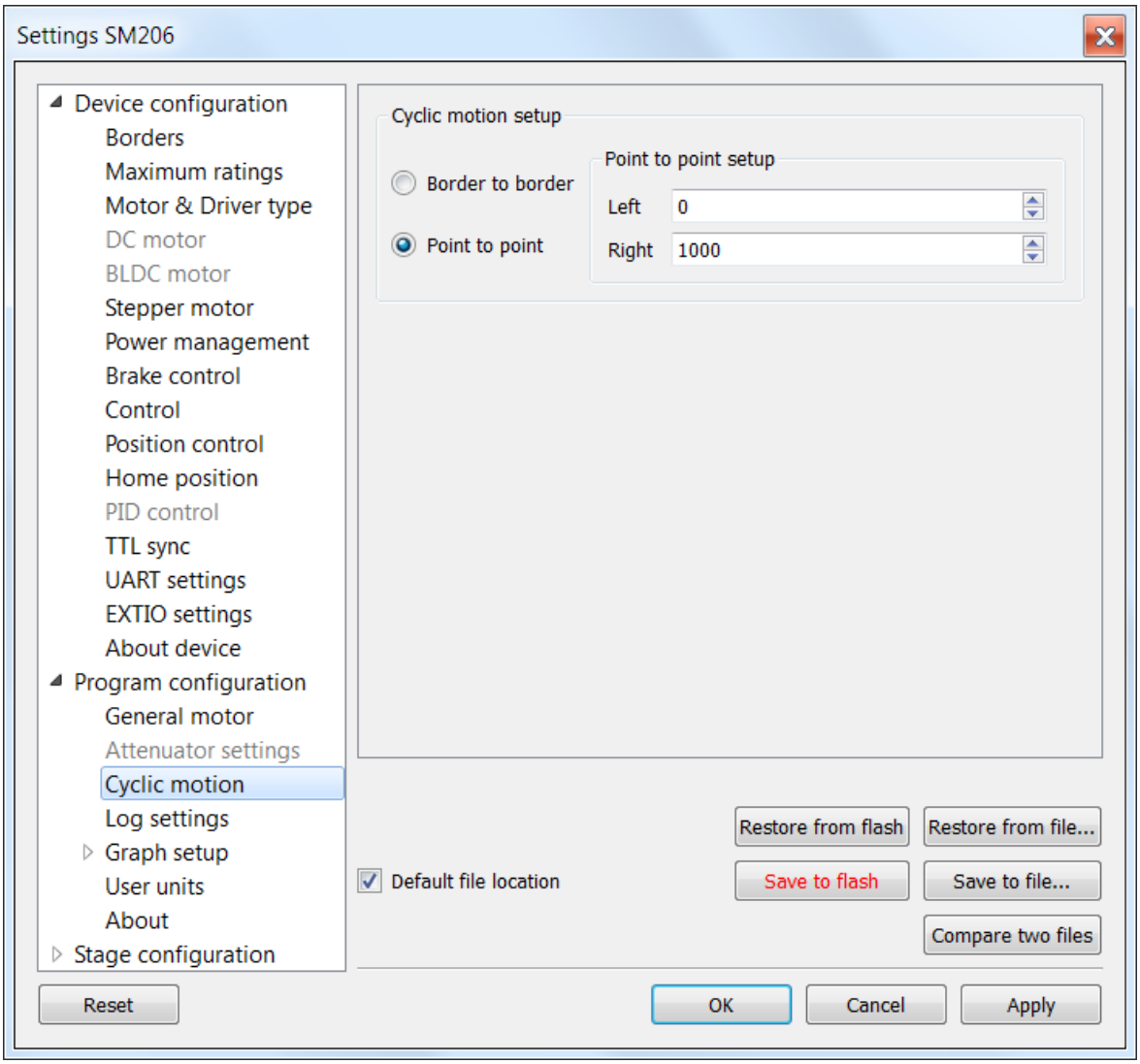
Окно вкладки "Настройки аттенюатора"

Данная вкладка предназначена для настройки интерфейса Xilab при работе с аттенюатором -- устройством с несколькими светофильтрами на вращающихся дисках, предназначенным для управления коэффициентом пропускания светового пучка. Вкладка становится активной при выборе опции "Attenuator" в блоке "Skin Type" в окне вкладки "Program configuration".

Группа **Attenuator type** содержит радиокнопки выбора аттенюатора с одним или двумя дисками. В зависимости от выбора становиится доступной один или два группы **Wheel**, в которых необходимо задать коэффициенты пропускания фильтров, расположенных на данном диске, в порядке следования.

5.4.4. Настройка режима циклического движения

В окне настроек программы ***Program configuration -> Cyclic motion***



Окно вкладки "Режим циклического движения"

Данная вкладка предназначена для настройки циклического перемещения между двумя заданными положениями. Используется главным образом в демонстрационных целях. Режим включается кнопкой Cyclic в главном окне, выключается кнопкой Stop в главном окне.

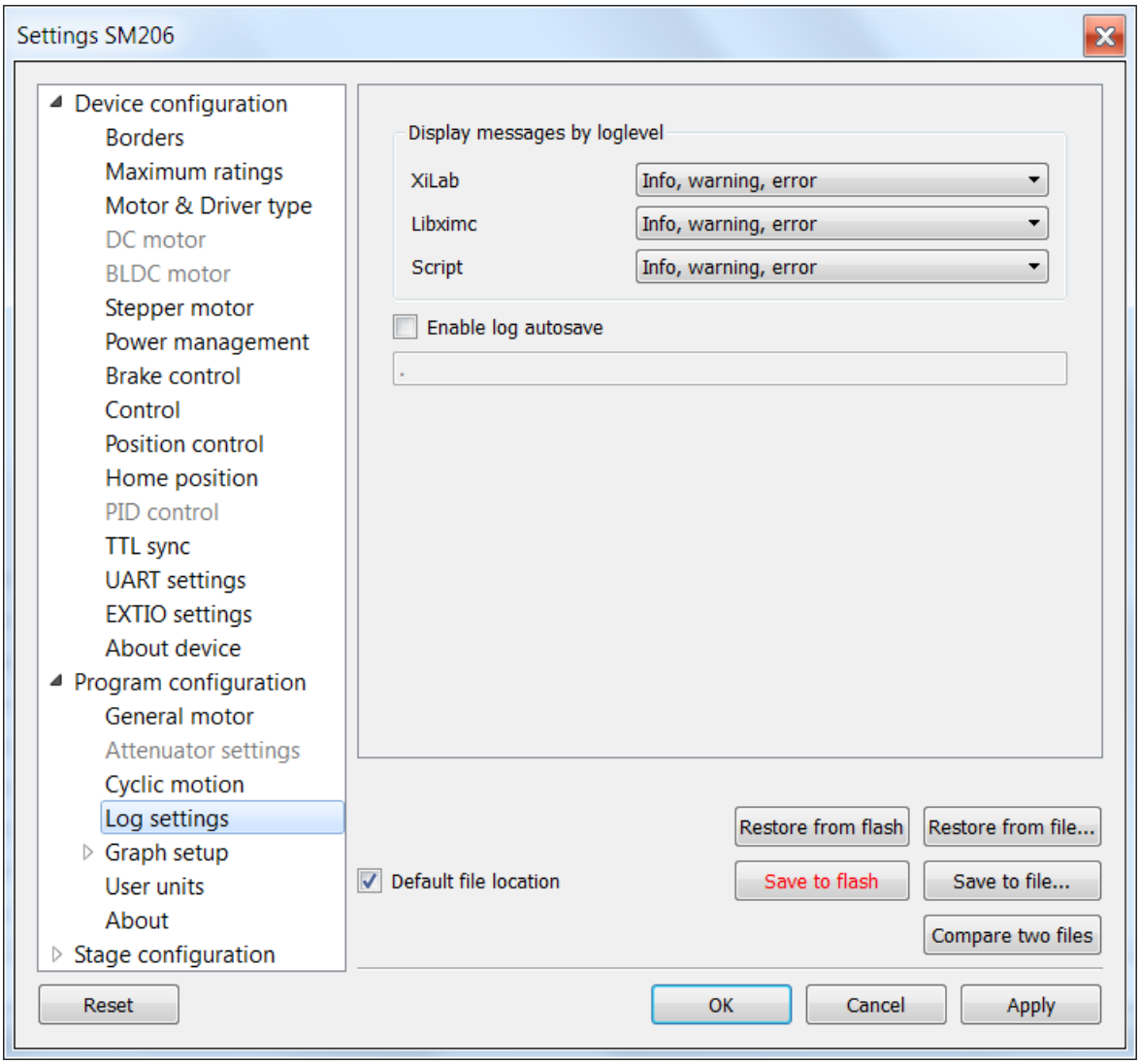
Настройки режима циклического движения:

Border to border - циклическое перемещение между границами настроенными во вкладке Настройка диапазона движения и концевых выключателей. Движение начинается к левой границе.

Point to point - циклическое перемещение между заданными в группе Point to point setup точками. Позиционер перемещается в левую точку, останавливается в ней, после этого перемещается в правую точку, останавливается, далее цикл повторяется.

5.4.5. Настройка логирования

В окне настроек программы ***Program configuration -> Log settings***



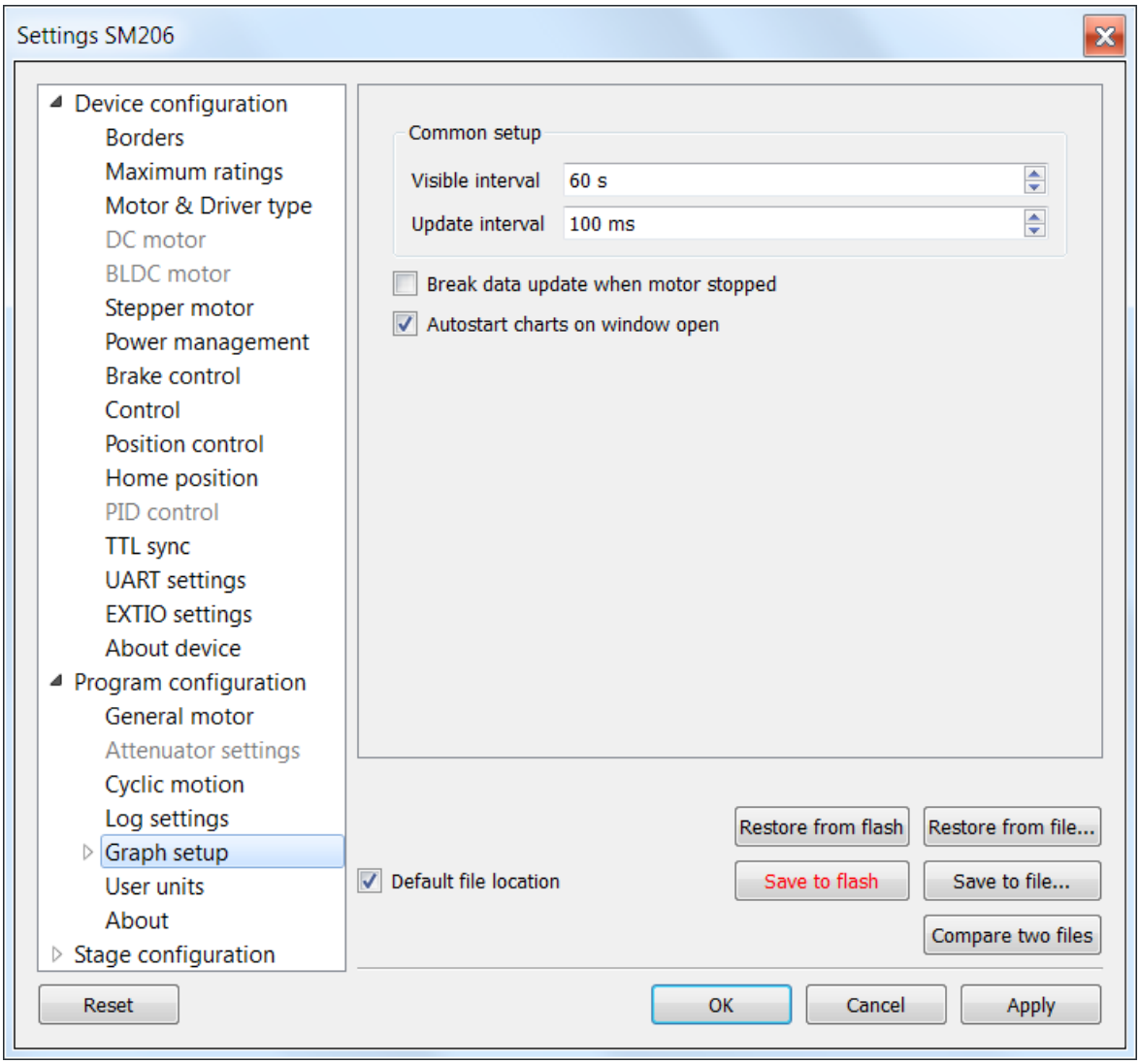
Окно настроек лога программы XILab

Данная вкладка предназначена для настройки уровня подробности логгирования. В блоке Display messages by loglevel возможные опции это не выводить ничего (None), выводить только ошибки (Error), ошибки и предупреждения (Error, Warning), ошибки, предупреждения и информационные сообщения (Error, Warning, Info) для каждого из источников: программа XILab, библиотека libximc и модуль скриптов Scripts.

При включении опции Enable log autosave включается автосохранение лога в файл, путь к директории хранения файла задается в строке ниже. Запись в файл производится каждые 5 секунд. Имя файла: "xilab_log_YYYY.MM.DD.csv", где YYYY, MM и DD - текущие год, месяц и день соответственно. Формат сохраненных данных: CSV
В файл записываются все сообщения лога независимо от уровня логирования.

5.4.6. Общие настройки отображения графиков

В окне настроек программы ***Program configuration -> Graph setup***



Вкладка общих настроек отображения графиков

Visible interval - интервал времени, отображаемый на графиках по горизонтальной оси.

Update interval - период обновления данных графиков.

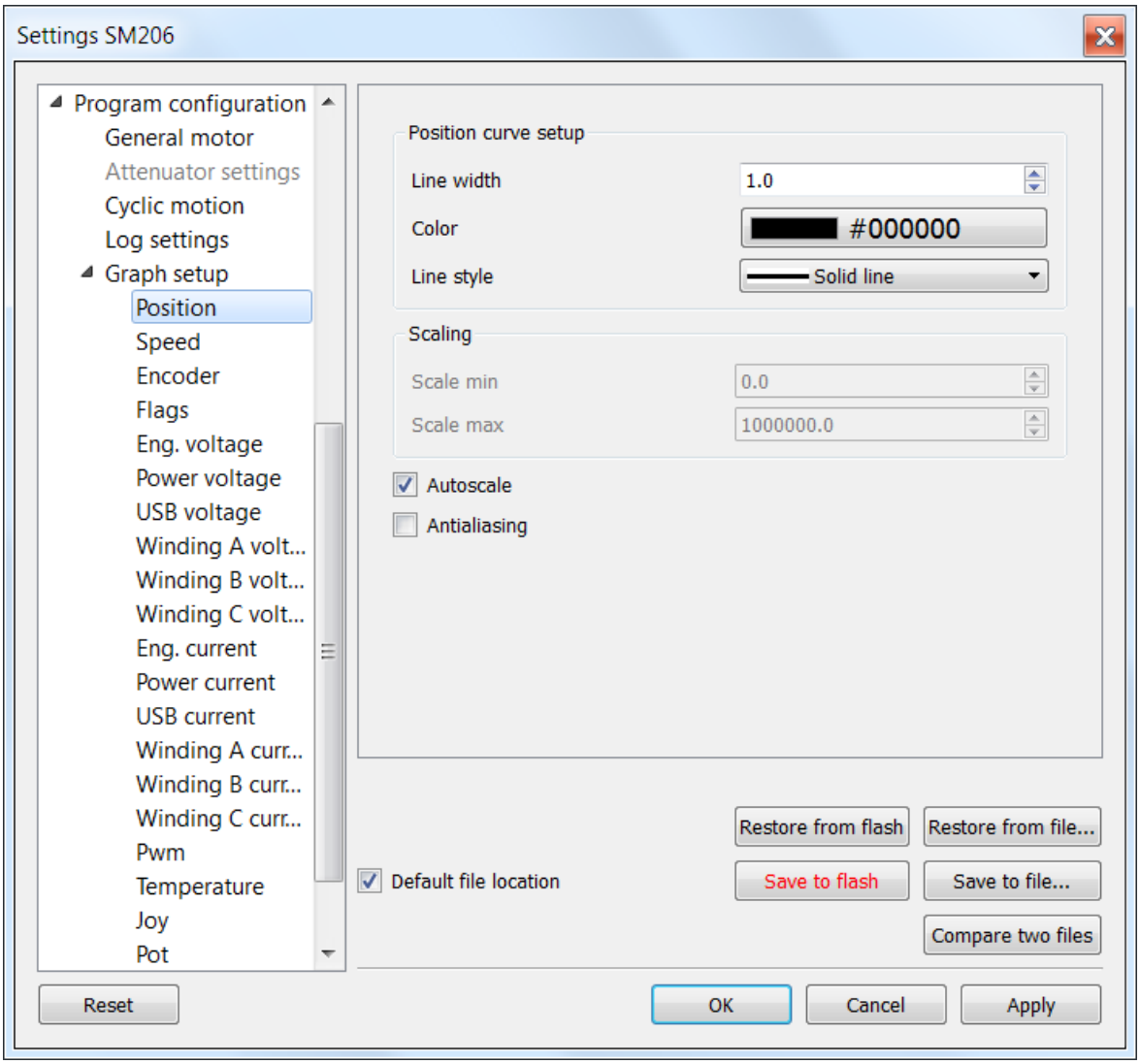
Break data update when motor stopped - остановка отрисовки графиков, когда мотор остановлен. Данный флаг позволяет более рационально использовать площадь графика, удаляя области, когда движение мотора отсутствует.

Autostart charts on window open - автоматическое начало отображения данных на графиках при открытии окна. Если вы хотите включать графики вручную, отключите эту опцию.

5.4.7. Индивидуальные настройки отображения графиков

В окне настроек программы ***Program configuration -> Graph setup -> ...***

Данный раздел в равной мере относится к индивидуальным настройкам отображения графиков скорости, напряжения, тока, скважности ШИМ, температуры и прочих параметров, отображение которых возможно на графиках в программе XILab.



Вкладка индивидуальных настроек на примере графика отображения положения

Настройка отображения графика включает в себя стиль линии и настройки масштабирования графика по вертикальной оси.

Группа параметров **Position curve setup** позволяет менять параметры кривой. Она включают в себя толщину Line width, цвет Color и тип линии Line style.

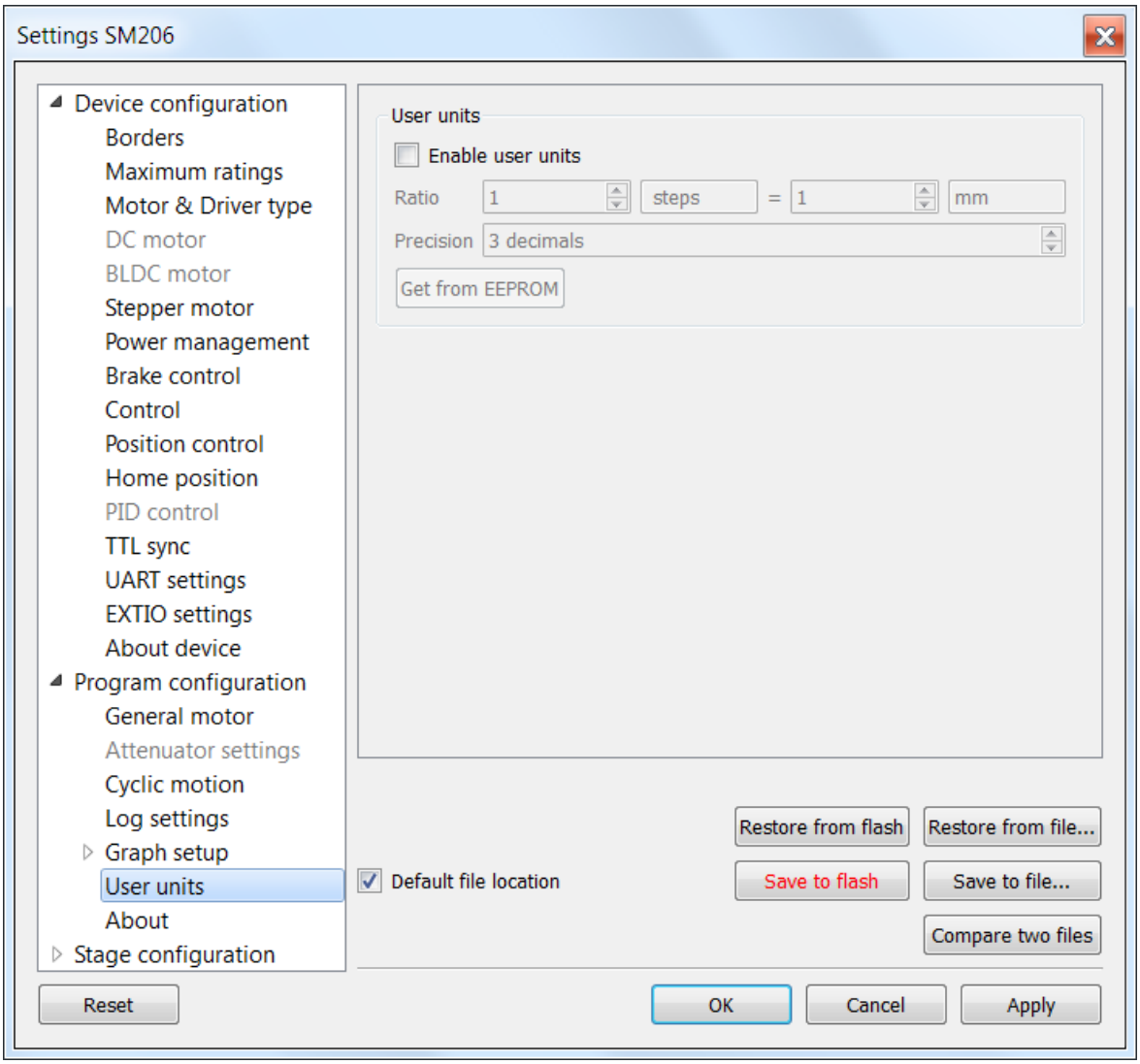
Группа параметров **Scaling** позволяет менять диапазон отображения кривой по вертикальной оси установкой значений Scale min и Scale max.

При выбранном флаге Autoscale производится автомасштабирование пределов шкалы в соответствии с пределами изменения переменной по оси Y. В этом случае параметры Scale min и Scale max игнорируются.

Флаг Antialiasing включает сглаживание линий графика, позволяющее добиться более качественного отображения, но несколько замедляющее процесс рисования графика.

5.4.8. Настройки отображения пользовательских единиц

В окне настроек программы ***Program configuration -> User units***



Окно вкладки "Настройки отображения пользовательских единиц"

Данная вкладка предназначена для настройки отображения пользовательских единиц. Используется для удобства задания и считывания координат в привычных пользователю единицах.

Настройки:

Enable user units - включает отображение пользовательских единиц вместо шагов/микрошагов (в случае шагового двигателя) или отсчетов энкодера (в случае DC). Пользовательские единицы заменяют собой шаги(отсчеты) только для отображения в главном окне программы и не влияют на настройки вкладок Settings.

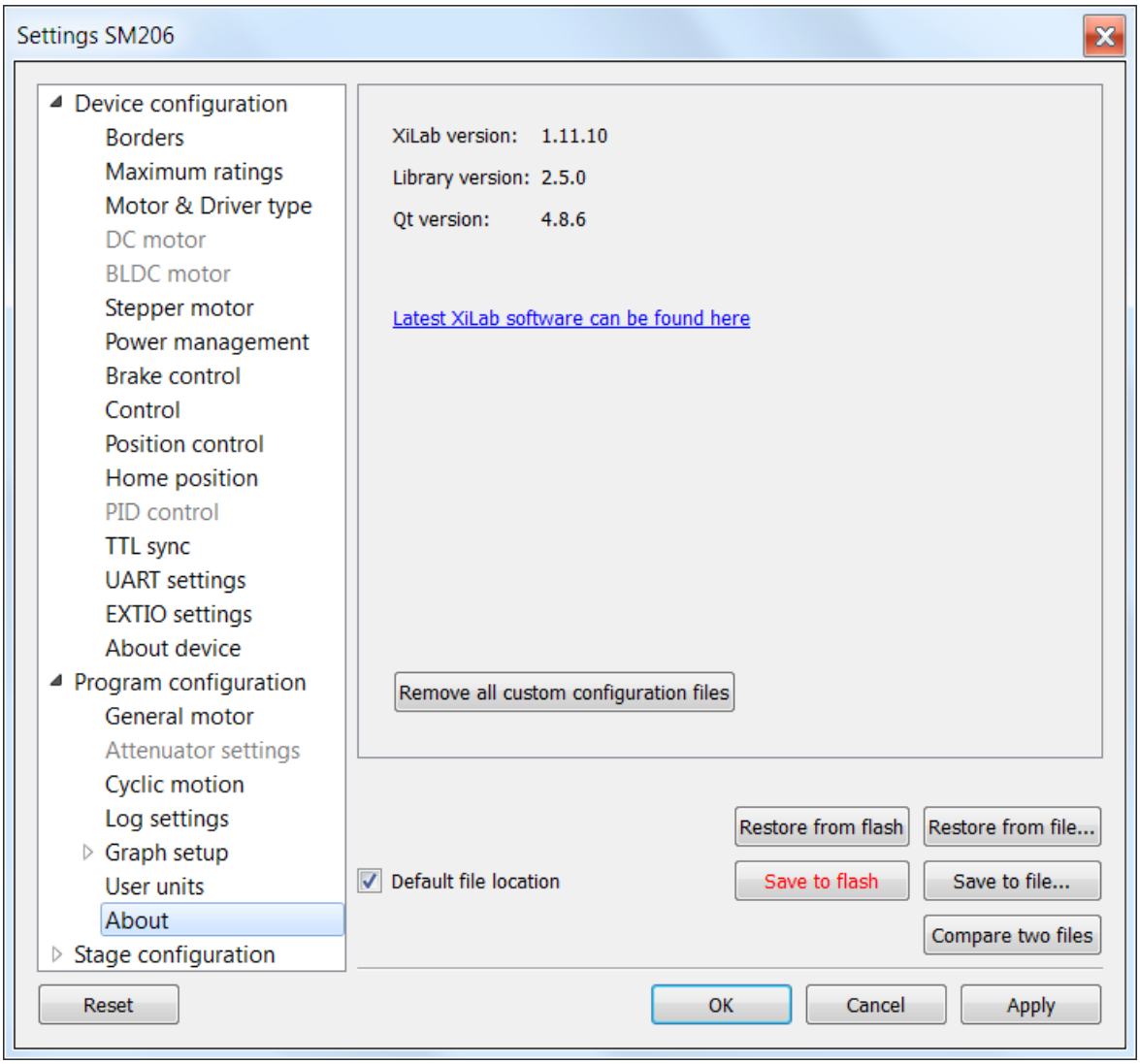
Ratio - коэффициент пересчета из шагов или отсчетов в пользовательские единицы, задается как отношение "x шагов = y единиц". Величины x, y, а также отображаемое название единицы вводятся пользователем.

Precision - точность отображения, количество знаков после запятой.

Кнопка Get from EEPROM считывает значения пользовательских единиц из памяти EEPROM.

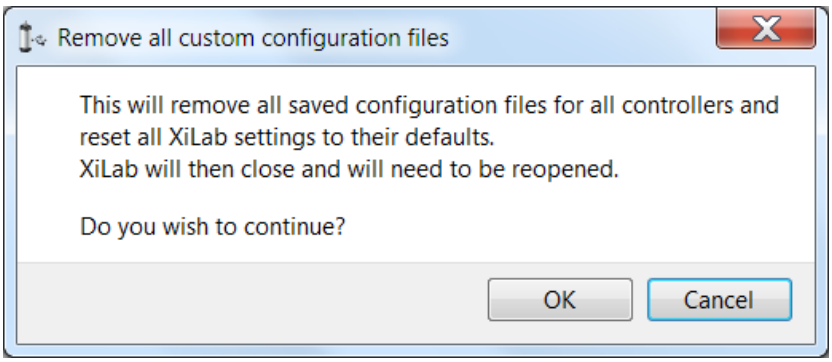
5.4.9. О программе

В окне настроек программы ***Program configuration -> About***



Вкладка "О программе"

В этом разделе отображается версия программы XILab. Также дана ссылка на страницу с последней версией программного обеспечения.



Диалог очистки пользовательских конфигураций

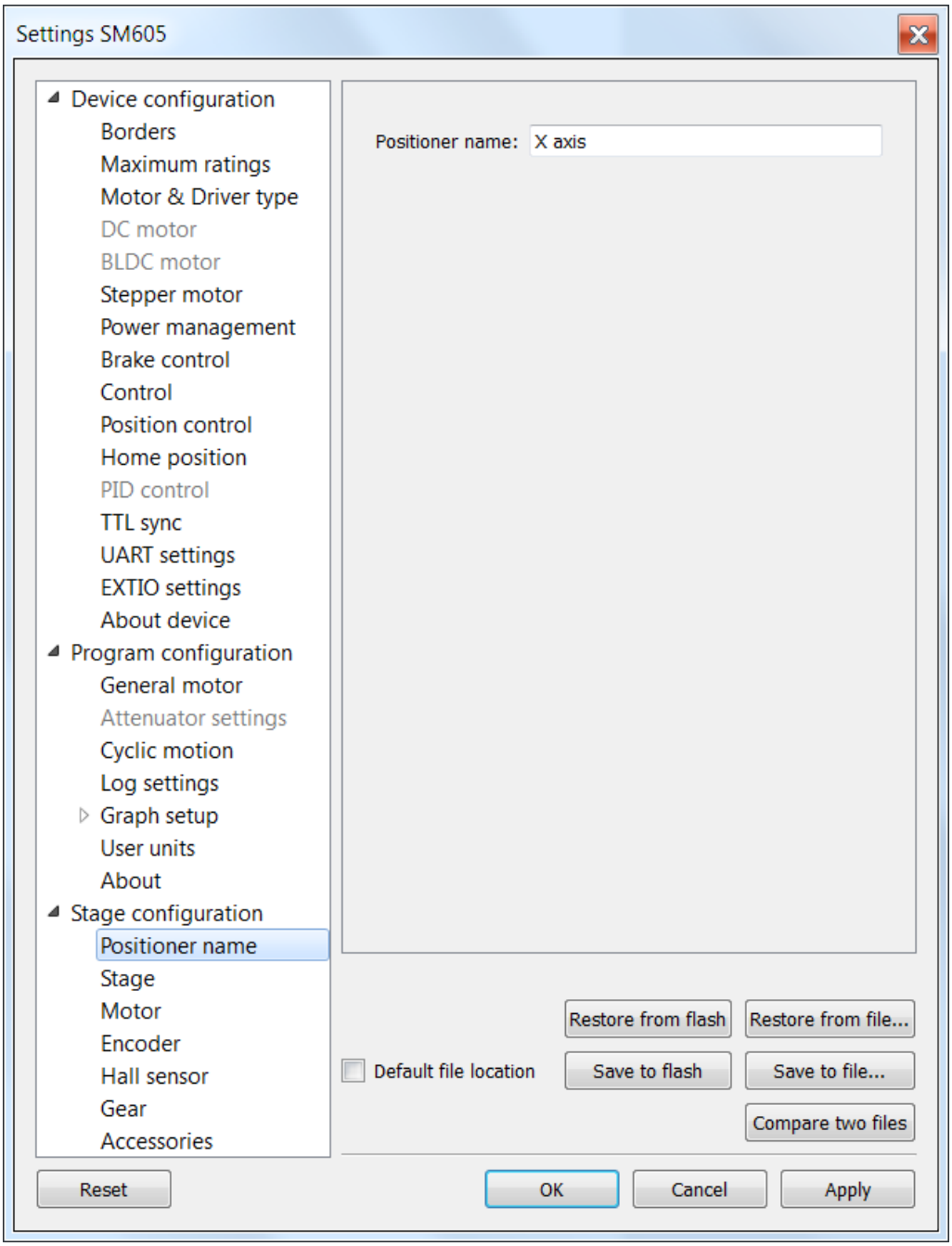
Кнопка "Remove all custom configuration files" отображает диалог с возможностью удаления всех конфигурационных файлов, созданных после инсталляции как результат запусков XiLab. Файлы, которые могут быть удалены, находятся в директории настроек XiLab: файл "settings.ini", хранящий общие настройки программы, файлы "SMnnn.cfg", хранящие индивидуальные настройки контроллеров, файлы "V_nnn", хранящие внутренние состояния виртуальных контроллеров, файл "scratch.txt", хранящий последний запущенный на выполнение скрипт (см. окно Скрипты). "nnn" здесь означает любое число. Нажатие OK в диалоге "Remove all custom configuration files" выполнит удаление файлов и закроет XiLab, нажатие Cancel отменит удаление и закроет диалог.

5.5. Характеристики позиционера

1. Название позиционера
2. Общие характеристики позиционера
3. Характеристики двигателя
4. Характеристики энкодера
5. Характеристики датчика Холла
6. Характеристики редуктора
7. Характеристики вспомогательных устройств

5.5.1. Название позиционера

В окне настроек программы ***Stage configuration -> Positioner name***

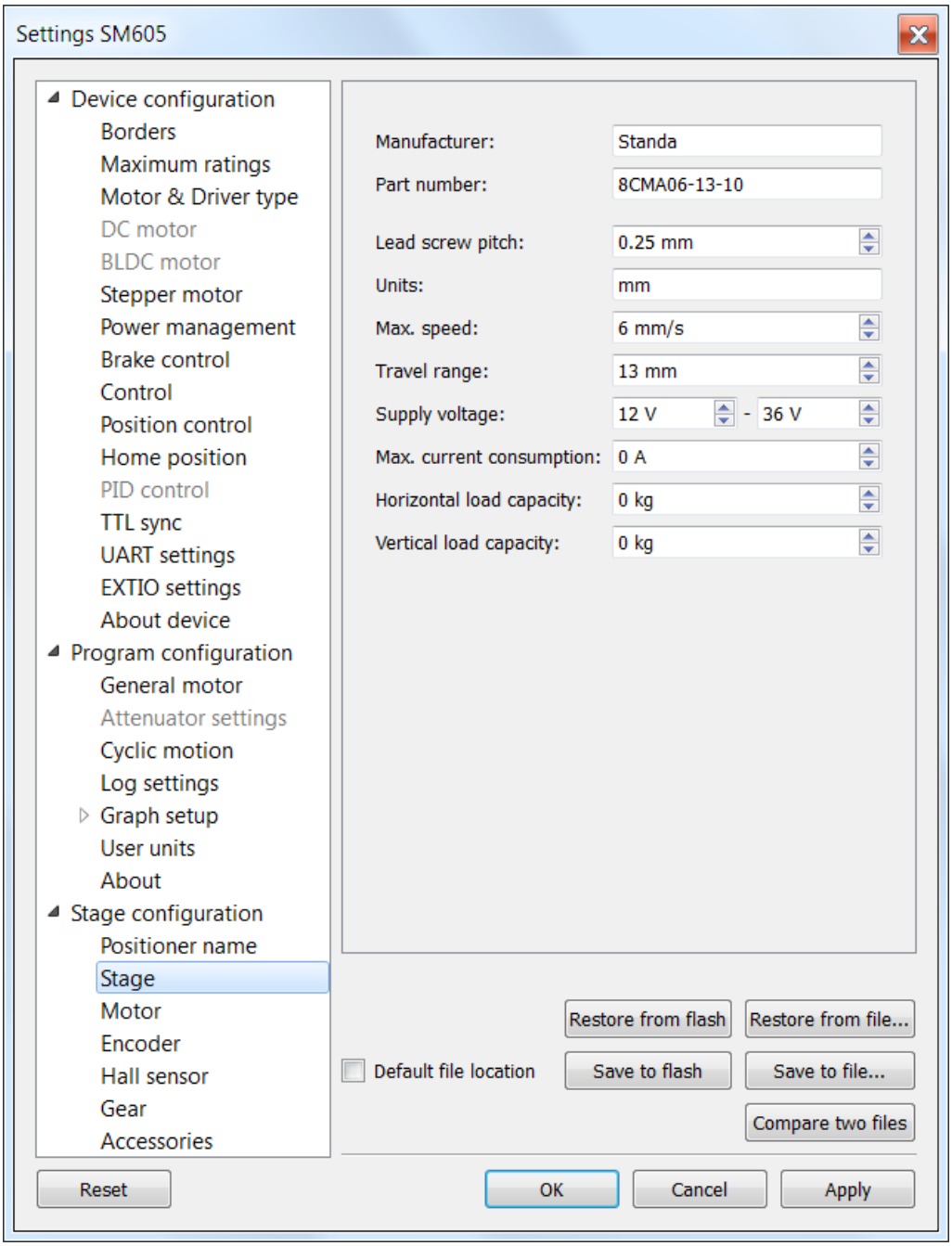


Окно с названием позиционера

Раздел содержит информацию с названием позиционера (задаётся пользователем).

5.5.2. Общие характеристики позиционера

В окне настроек программы ***Stage configuration -> Stage***



Окно с общими характеристиками позиционера

Группа параметров **stage** содержит информацию о позиционере.

Manufacturer - наименование производителя.

Part number - номер по каталогу.

Lead screw pitch - шаг резьбы ходового винта.

Units - единицы измерения перемещения в данном позиционере (мм, градусы, шаги).

Max. speed - максимальная скорость.

Travel range - диапазон перемещения.

Supply voltage - диапазон допустимых напряжений питания.

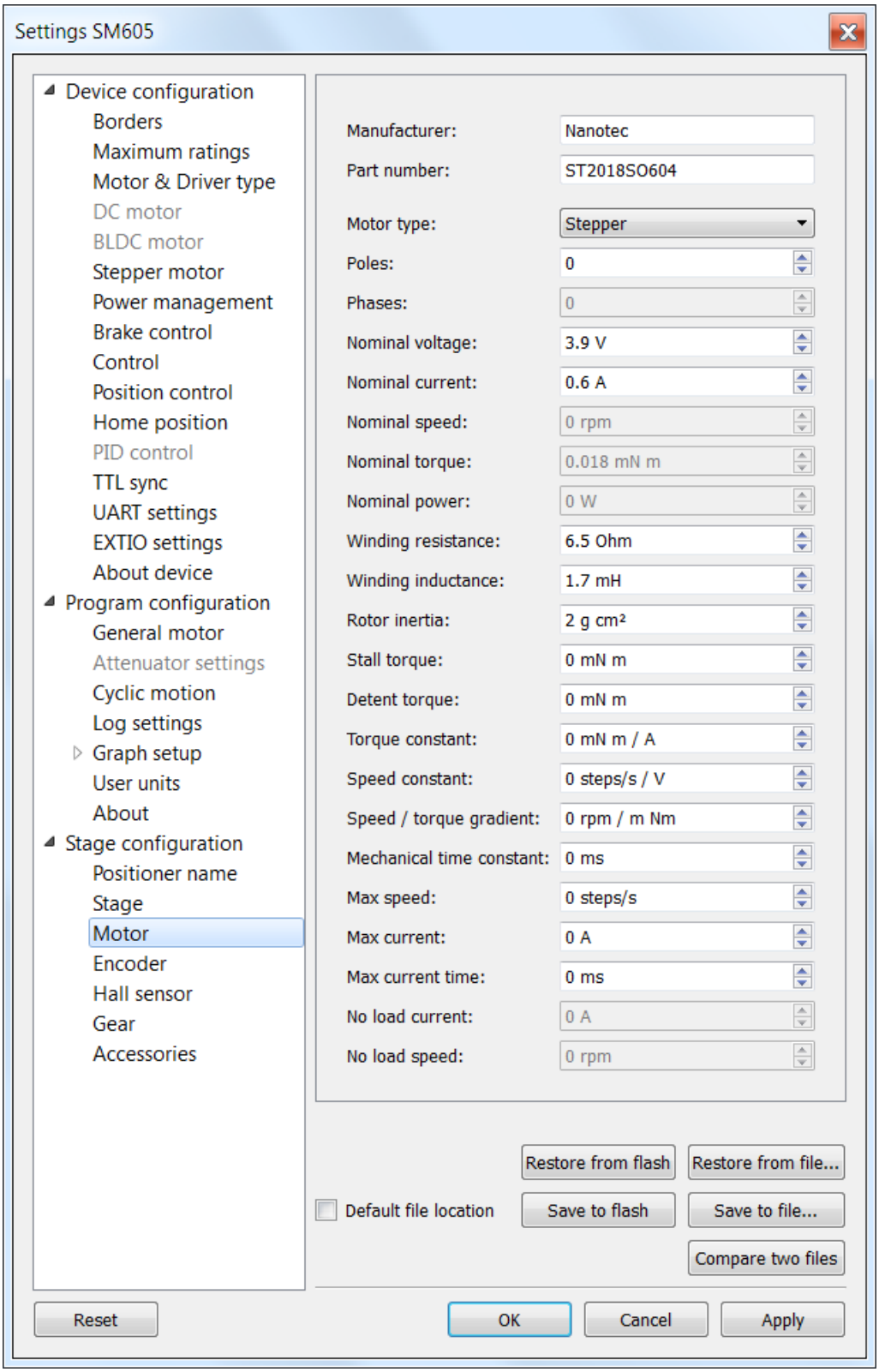
Max. current consumption - максимальное потребление тока.

Horizontal load capacity - максимальная горизонтальная нагрузка на позиционер.

Vertical load capacity - максимальная вертикальная нагрузка на позиционер.

5.5.3. Характеристики двигателя

В окне настроек программы ***Stage configuration -> Motor***



Окно с описанием характеристик двигателя

Раздел содержит информацию о двигателе.

Manufacturer - компания-производитель мотора.

Part number - номер по каталогу.

Motor type - тип мотора (шаговый, DC, BLDC)

Poles - Кол-во пар полюсов у DC или BLDC двигателя или кол-во шагов на оборот для шагового двигателя.

Phases - Кол-во фаз у BLDC двигателя.

Nominal voltage - номинальное напряжение на обмотке.

Nominal current - максимальный постоянный ток в обмотке для DC и BLDC двигателей, номинальный ток в обмотке для шаговых двигателей.

Nominal speed - номинальная скорость.

Nominal torque - номинальный крутящий момент.

Nominal power - номинальная мощность.

Winding resistance - активное сопротивление обмотки.

Winding inductance - индуктивность обмотки.

Rotor inertia - инерция ротора.

Stall torque - крутящий момент при нулевой скорости.

Detent torque - момент удержания позиции с незапитанными обмотками.

Torque constant - константа крутящего момента.

Speed constant - константа скорости.

Speed/torque gradient - константа скорость/момент.

Mechanical time constant - постоянная времени мотора.

Max speed - максимальная разрешённая скорость.

Max current - максимальный ток в обмотке.

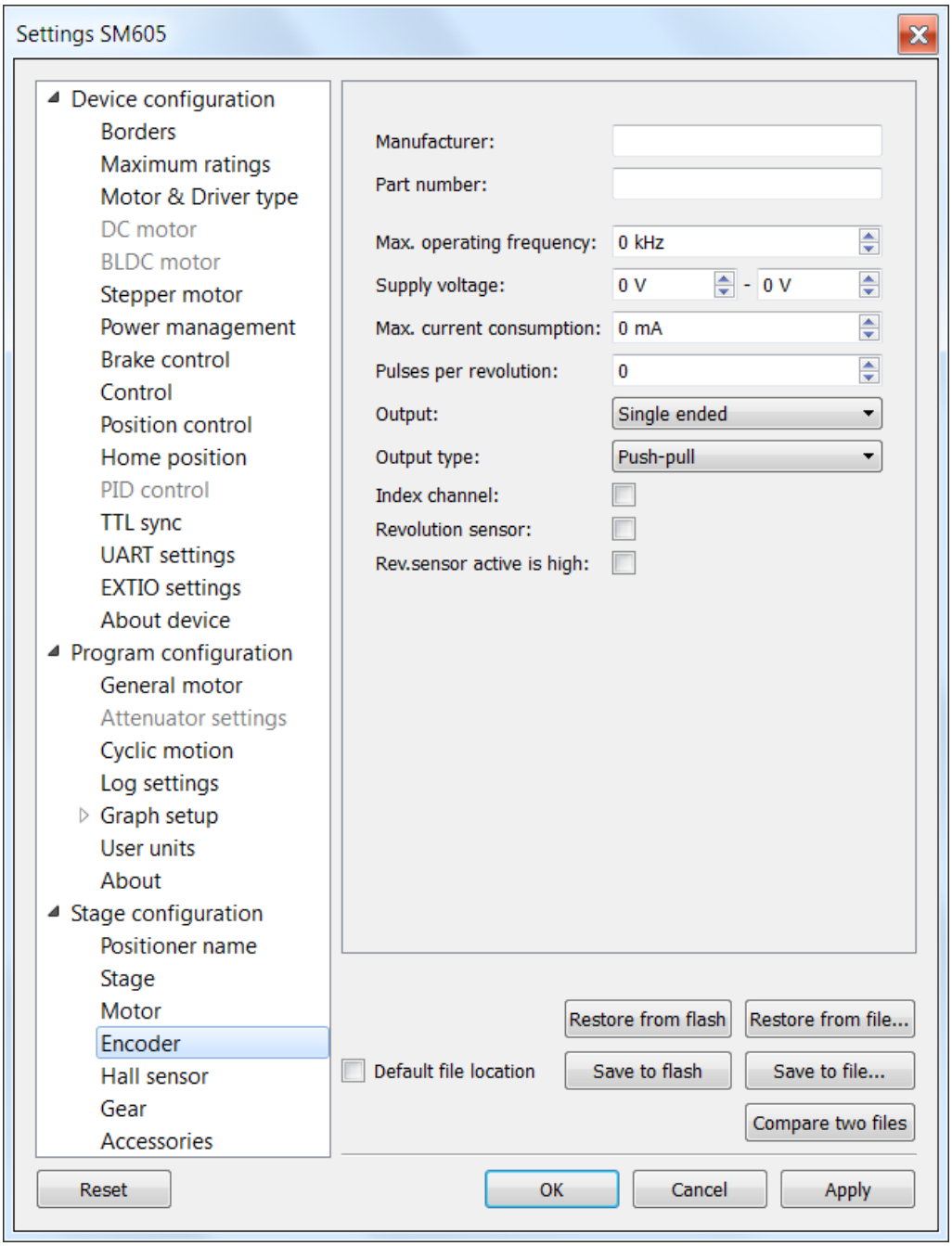
Max current time - безопасная длительность максимального тока в обмотке.

No load current - потребляемый без нагрузки ток.

No load speed - скорость на холостом ходу.

5.5.4. Характеристики энкодера

В окне настроек программы ***Stage configuration -> Encoder***



Окно с описанием характеристик энкодера

Раздел содержит информацию об энкодере:

Manufacturer - наименование компании-производителя энкодера.

Part number - номер по каталогу.

Max. operating frequency - максимальная рабочая частота.

Supply voltage - диапазон допустимых напряжений питания.

Max. current consumption - максимальное потребление тока.

Pulses per revolution - количество отсчётов на оборот.

Output - дифференциальный выход или несимметричный выход.

Output type - тип выхода (двухтактный выход, или выход с открытым коллектором).

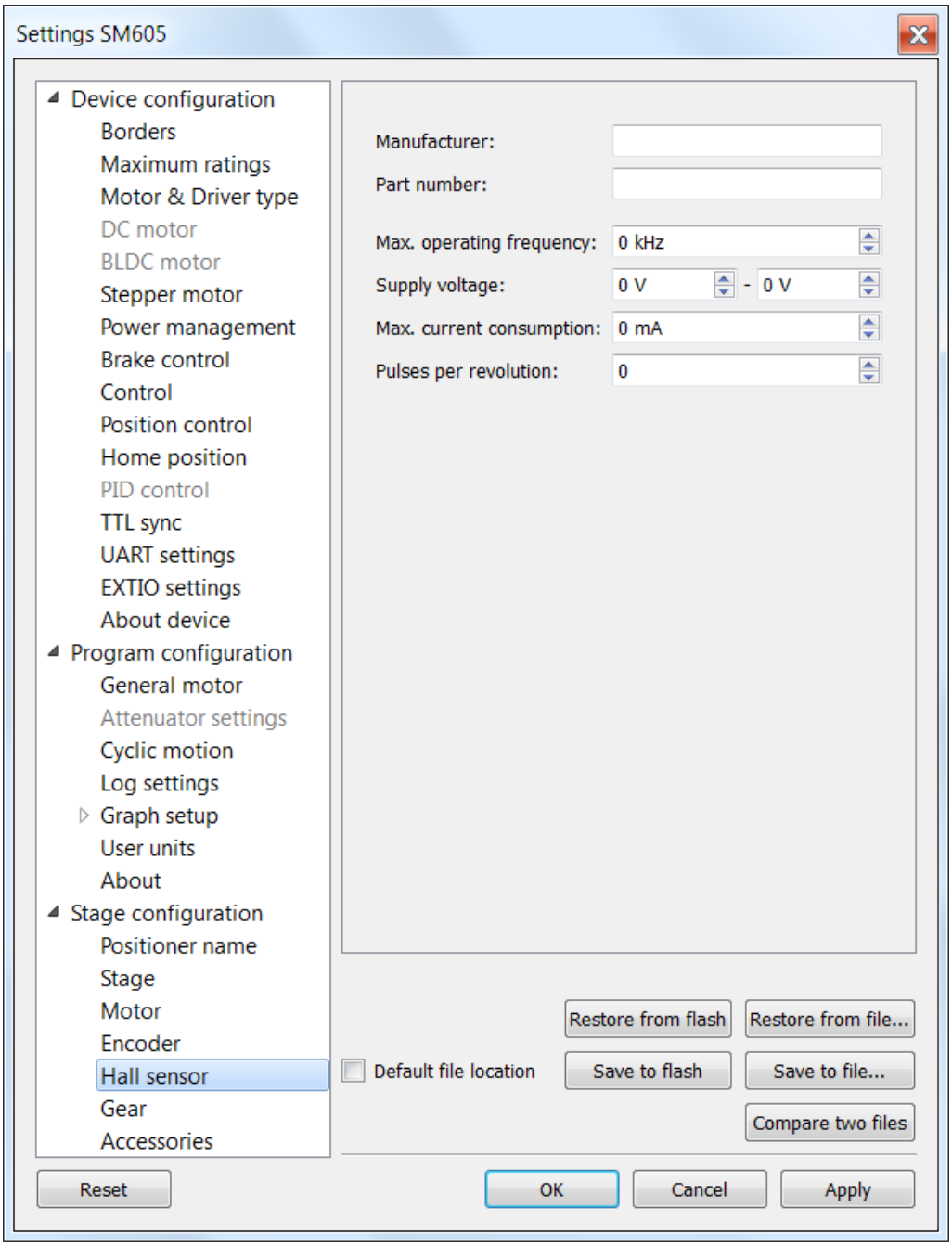
Index channel - наличие дополнительного индексного канала.

Revolution sensor - наличие датчика оборотов.

Rev.sensor active is high - присутствие данного свойства означает что активное состояние датчика оборотов соответствует логической единице, отсутствие - логическому нулю.

5.5.5. Характеристики датчика Холла

В окне настроек программы ***Stage configuration -> Hall sensor***



Окно с описанием характеристик датчика Холла

Раздел содержит информацию о датчике Холла:

Manufacturer - наименование компании-производителя датчика.

Part number - номер по каталогу.

Max. operating frequency - максимальная рабочая частота.

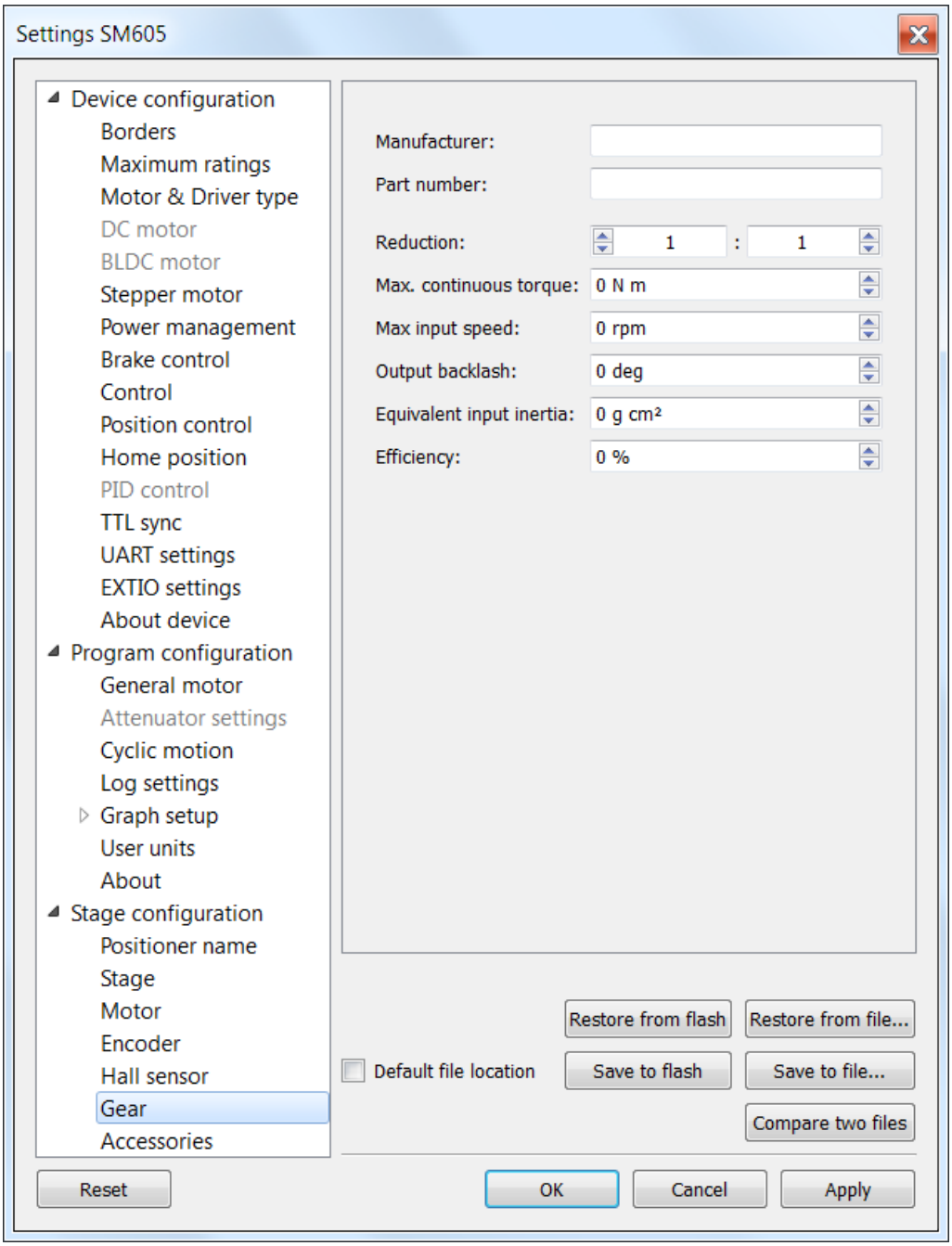
Supply voltage - диапазон допустимых напряжений питания.

Max. current consumption - максимальное потребление тока.

Pulses per revolution - количество отсчётов на оборот.

5.5.6. Характеристики редуктора

В окне настроек программы ***Stage configuration -> Gear***



Окно с описанием характеристик редуктора

Раздел содержит информацию о редукторе.

Manufacturer - наименование компании-производителя.

Part number - номер по каталогу.

Reduction - передаточный коэффициент редуктора.

Max. continuous torque - максимальный крутящий момент на входе редуктора.

Max. input speed - максимальная скорость на входе редуктора.

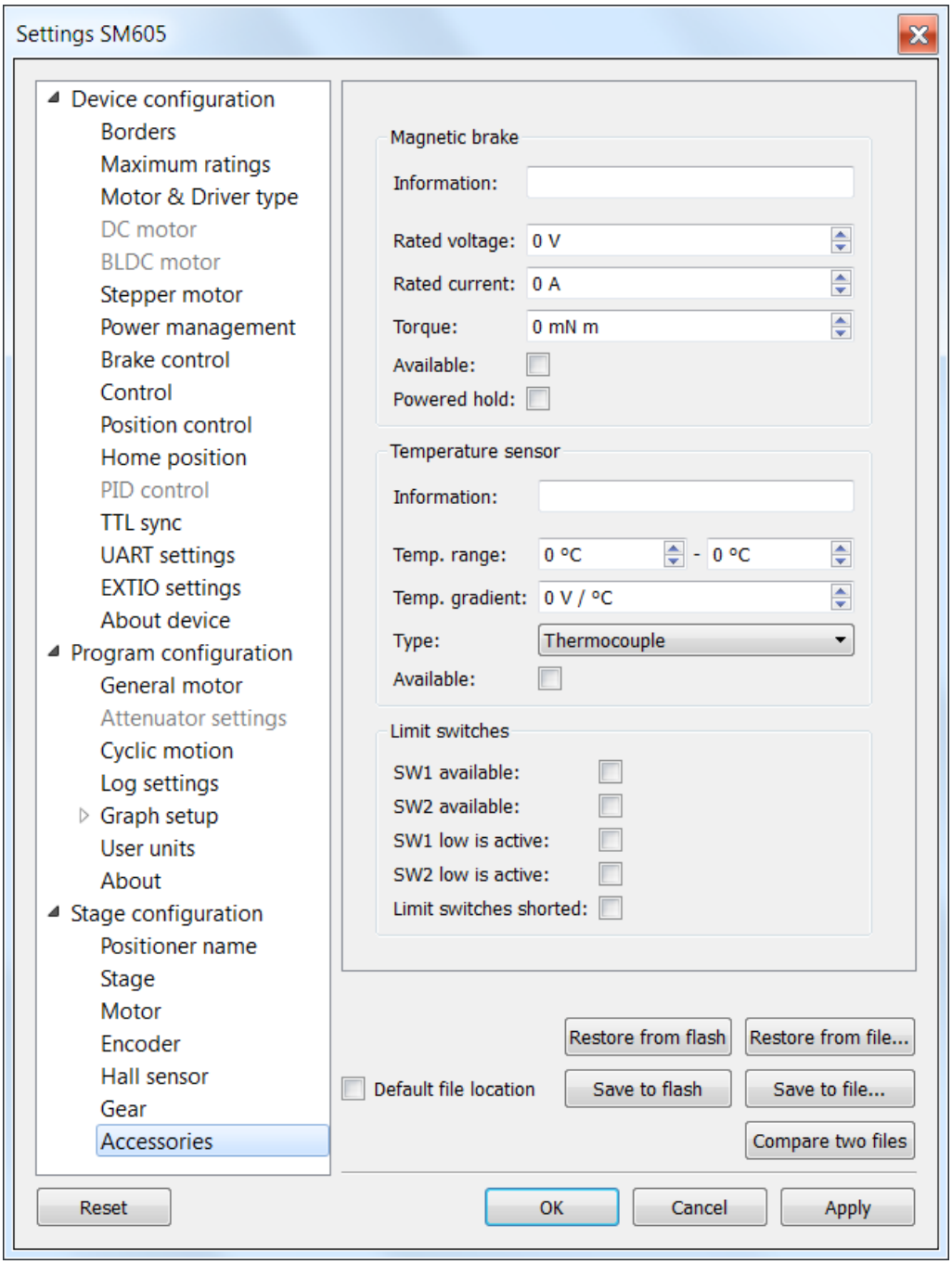
Output backlash - выходной люфт редуктора.

Equivalent input inertia - эквивалентная входная инерция редуктора.

Efficiency - КПД редуктора.

5.5.7. Характеристики вспомогательных устройств

В окне настроек программы ***Stage configuration -> Accessories***



Окно с описанием характеристик вспомогательных устройств

Раздел содержит информацию о вспомогательных устройствах.

Magnetic brake - блок магнитного тормоза.

Information - производитель и номер магнитного тормоза.

Rated voltage - номинальное напряжение для управления магнитным тормозом.

Rated current - номинальный ток для управления магнитным тормозом.

Torque - удерживающий момент.

Available - наличие тормоза.

Powered hold - присутствие данного свойства означает что магнитный тормоз находится в режиме удержания (активен) при подаче питания.

Temperature sensor - блок термодатчика.

Information - производитель и номер температурного датчика.

Temp. range - диапазон измеряемых температур.

Temp. gradient - температурный градиент.

Type - тип датчика (термопара или полупроводниковый температурный датчик).

Available - наличие датчика.

Limit switches - блок концевиков.

SW1 available - присутствие данного свойства означает что концевик, подключенный к ножке SW1, доступен.

SW2 available - присутствие данного свойства означает что концевик, подключенный к ножке SW2, доступен.

SW1 low is active - присутствие данного свойства означает что концевик, подключенный к ножке SW1, считается сработавшим по низкому уровню на контакте.

SW2 low is active - присутствие данного свойства означает что концевик, подключенный к ножке SW2, считается сработавшим по низкому уровню на контакте.

Limit switches shorted - присутствие данного свойства означает что концевики закорочены.

5.6. Корректное завершение работы

Корректное завершение работы подразумевает остановку двигателя и сохранение текущей позиции контроллером. Текущая позиция сохраняется автоматически, см. Хранение позиции во FRAM-памяти контроллера.

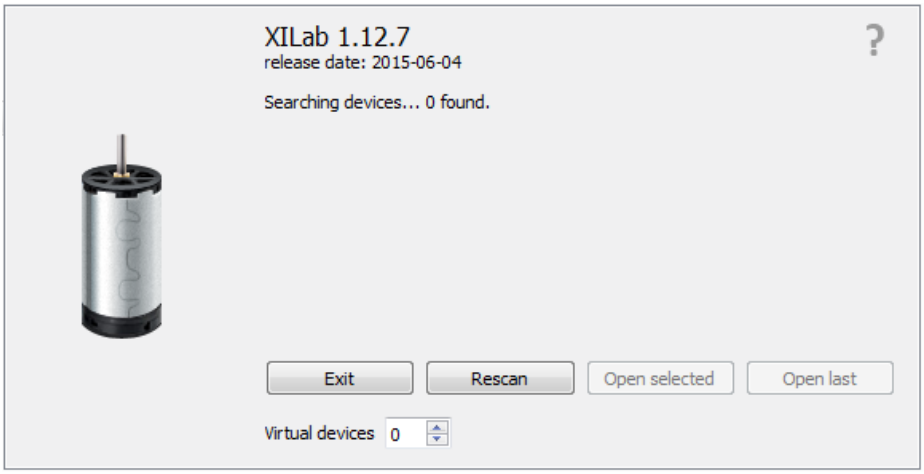
Кнопка Exit осуществляет корректное завершение работы и выход из программы. При нажатии на неё программа отдает контроллеру команду плавной остановки, а после завершения остановки команду отключения питания. Если выполнение плавной остановки было прервано каким-либо событием, например подачей команды движения джойстиком или сигналом TTL-синхронизации, или если при посылке команды плавной остановки или команды отключения питания в контроллер библиотека вернула ошибку, то выход отменяется. В этом случае необходимо проверить настройки джойстика и кнопок "вправо" и "влево" и настройки синхронизации.

5.7. Работа с сетевыми контроллерами

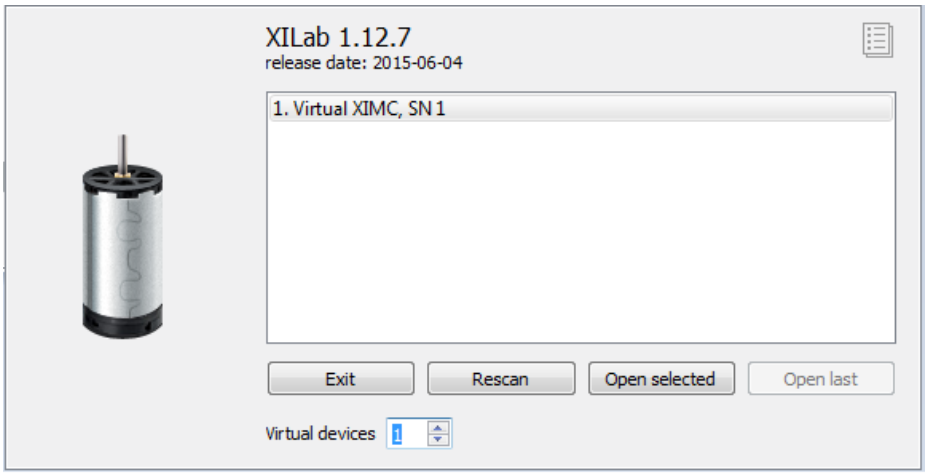
XiLab способен работать с контроллерами через Ethernet. Для этого необходим либо адаптер **8SMC4-USB-Eth1**, либо исполняемый файл сервера libximc, который вы можете запустить на любом удобном для вас устройстве с подключенными к нему контроллерами.

- В случае работы с **8SMC4-USB-Eth1** перед запуском соедините устройство с контроллерами USB кабелем. При этом, предполагается, что к контроллерам может быть подключен двигатель и подано силовое питание (подробнее см. в соответствующем разделе).
- Подключите устройство, на котором запущен сервер libximc, к той же подсети, в которой находится управляющий компьютер и DHCP сервер. В случае работы с **8SMC4-USB-Eth1** подайте на устройство питание (через разъём 5V - 2A) через блок питания, идущий в комплекте и подождите 1-2 минуты, чтобы ОС Linux на одноплатном компьютере успела загрузиться.
- Включите XiLab и проделайте следующую процедуру:

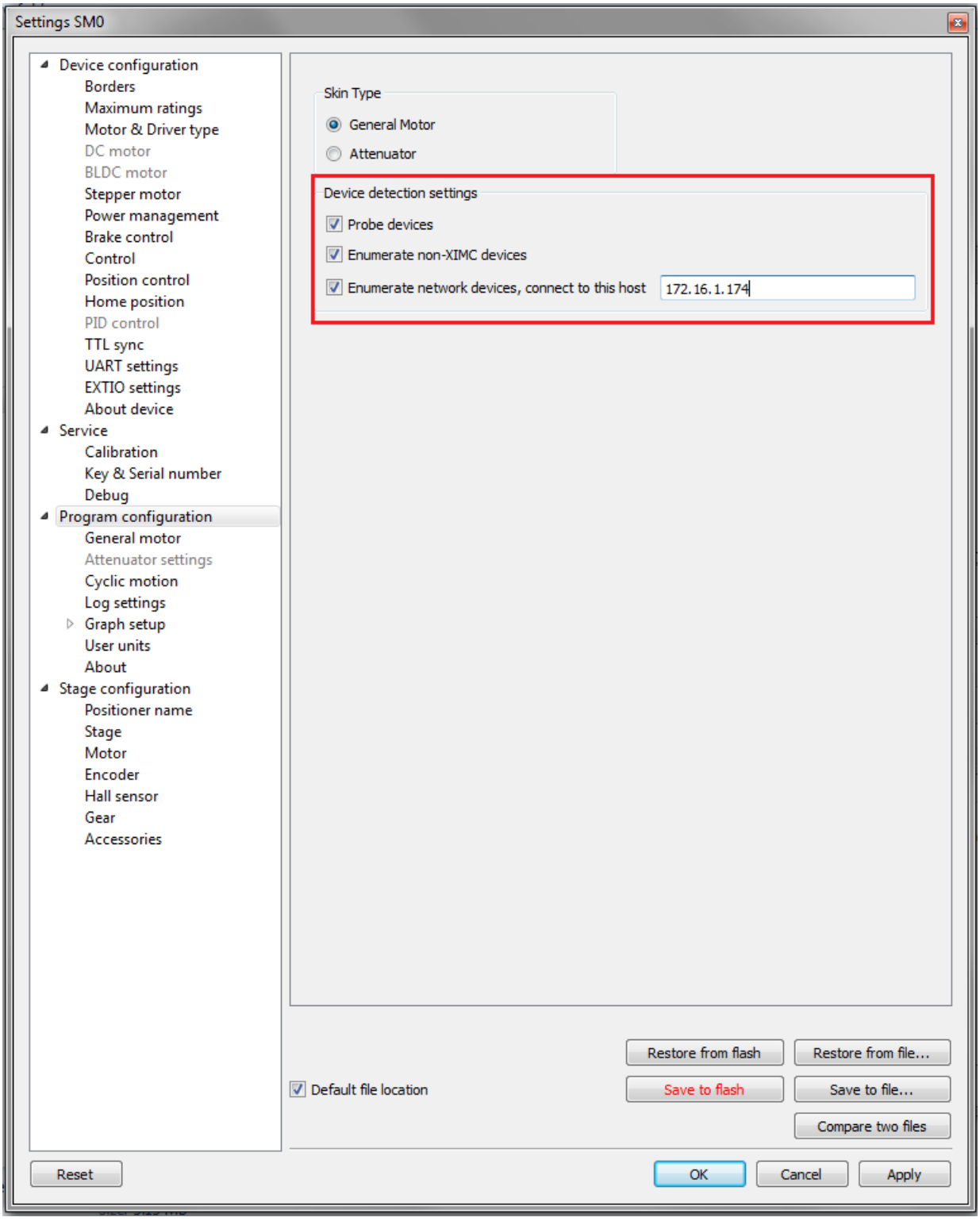
При первом запуске появится стартовое окно, в котором не будет найдено ни одного устройства.



Добавьте виртуальный контроллер. Для этого увеличьте значение в поле Virtual Devices в нижней части окна, затем нажмите Rescan. В появившемся окне выберите Virtual XIMC, SN1 и нажмите Open selected.



Перед Вами появится основное окно программы XiLab. Зайдите в Settings... и слева в списке выберите пункт Program configuration (подробнее о пунктах данного окна смотрите раздел Общие настройки программы XILab). В разделе Device detection settings поставьте галочку напротив Enumerate network devices, и нажмите кнопку "Scan for local XIMC servers". Устройство, если в данный момент на нём запущен сервер libximc, будет найден автоматически с помощью широковещательного запроса. Его IP-адрес появится в списке адресов (список адресов также можно при желании редактировать вручную).
Нажмите OK и закройте программу.



При повторном запуске XiLab найдутся все доступные адаптеру оси. В появившемся списке можно выбрать интересующую ось и управлять ей, как в случае с одним контроллером. Также можно выбрать несколько осей и управлять ими в режиме управления несколькими осями. Подробнее см. Начало работы в ПО XILab и Руководство по программе XILab.

> *Замечание.* Единожды обнаружив адрес устройства, следует иметь ввиду, что перенос устройства в другое место может привести к изменению IP.

Замечание. Работа с несколькими адаптерами может привести к тому, что при широковещательном запросе первым будет отвечать один и тот же адаптер с подключёнными к нему контроллерами. Существует два способа решения этой поблемы:

- Отключить остальные оси, найти устройство в сети, подключить всё обратно
- Многократно нажимать Scan for local XIMC servers до тех пор, пока не будет найден нужный адаптер.

5.7. Установка XILab

1. Установка под Windows
    1. Установка под Windows XP
    2. Установка под Windows 7
    3. Установка под Windows 8
2. Установка под Linux
3. Установка под MacOS

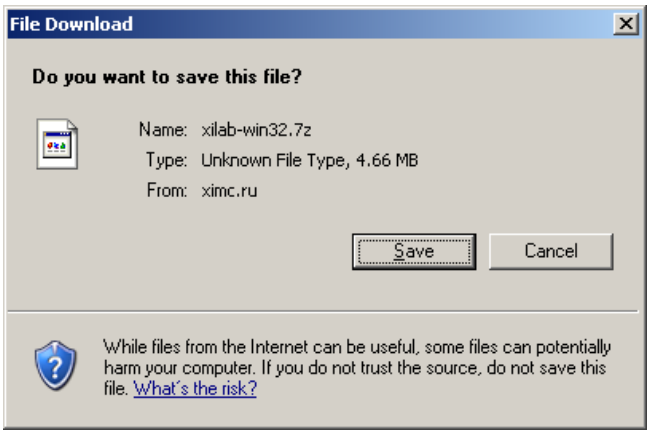
5.8.1. Установка под Windows

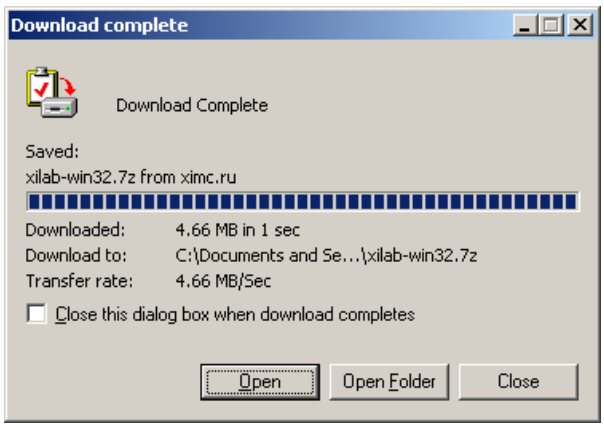
1. Установка под Windows XP
2. Установка под Windows 7
3. Установка под Windows 8

5.8.1.1. Установка под Windows XP

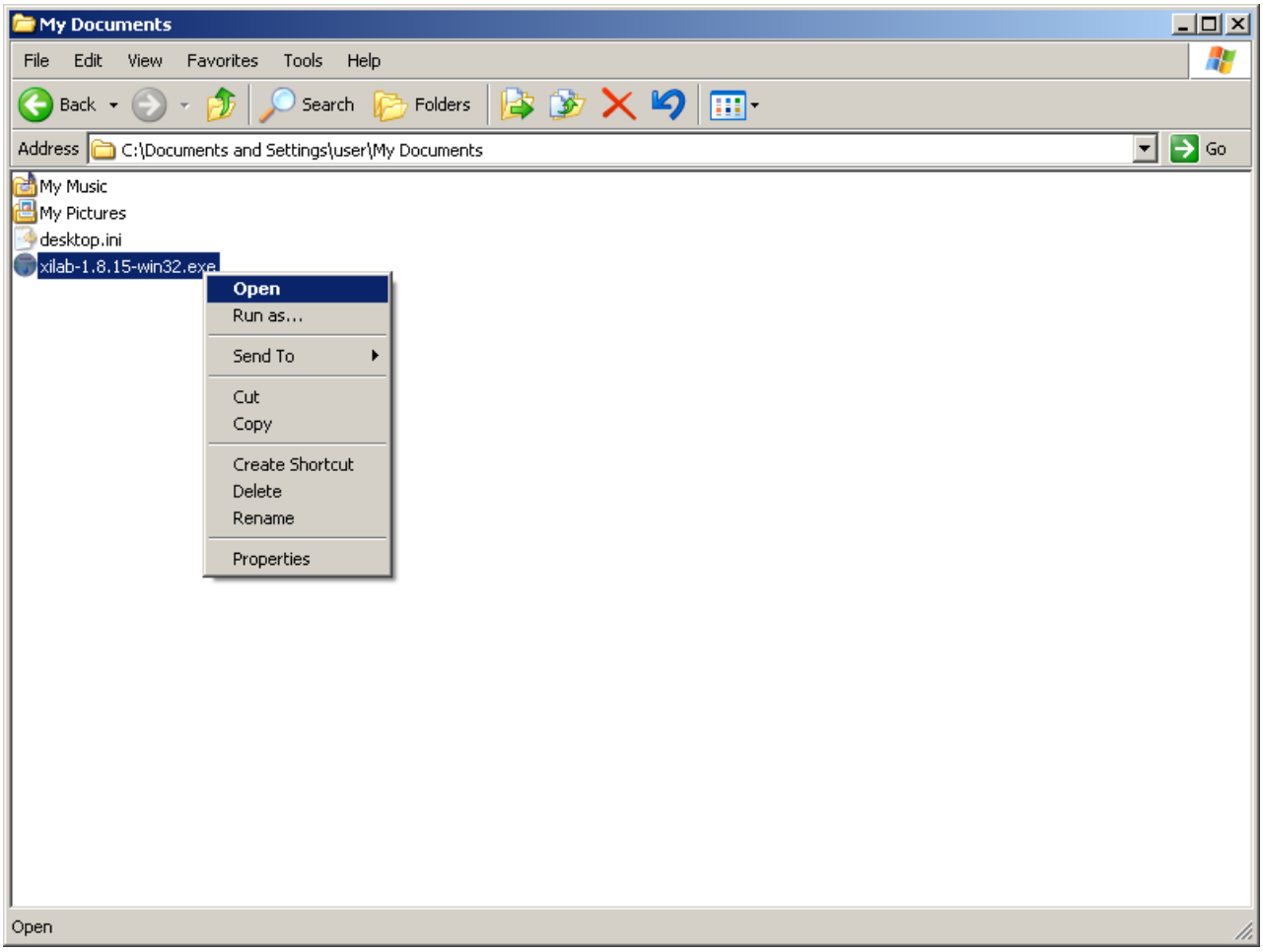
Скопируйте файл с программой установки на компьютер. Программа установки имеет название "xilab-<номер версии>.exe". Инсталлятор автоматически определяет, запущен ли он на 32-битной или 64-битной системе и устанавливает соответствующую версию.

> *Замечание.* Обратите внимание, поддерживается только Windows XP SP3. Пожалуйста обновите систему до последнего сервис пака.

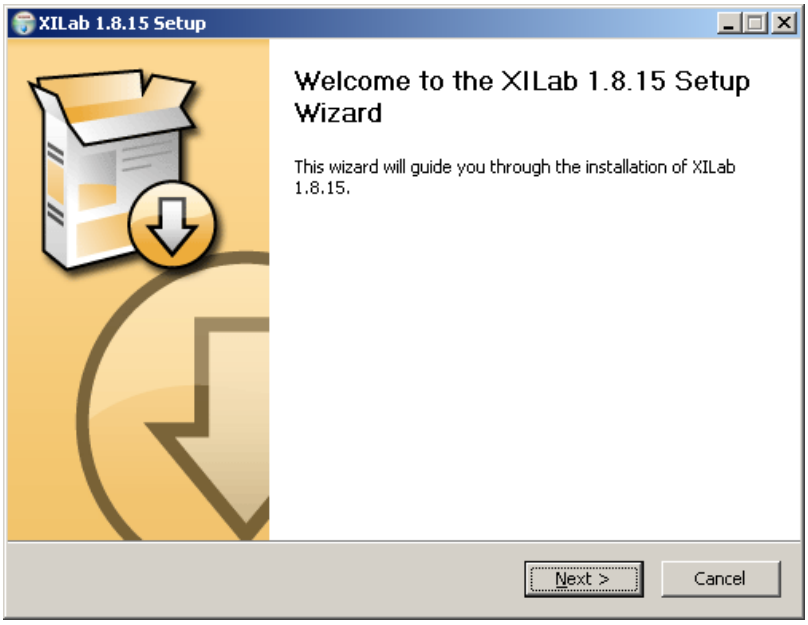




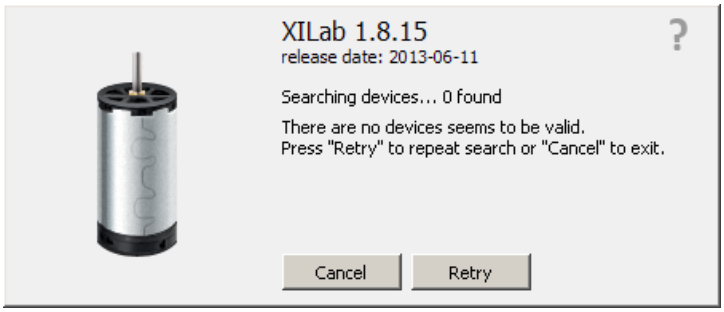
Запустите программу установки и следуйте инструкциям на экране.



Все необходимое программное обеспечение, включая драйверы, пакеты и программы будут установлены автоматически.

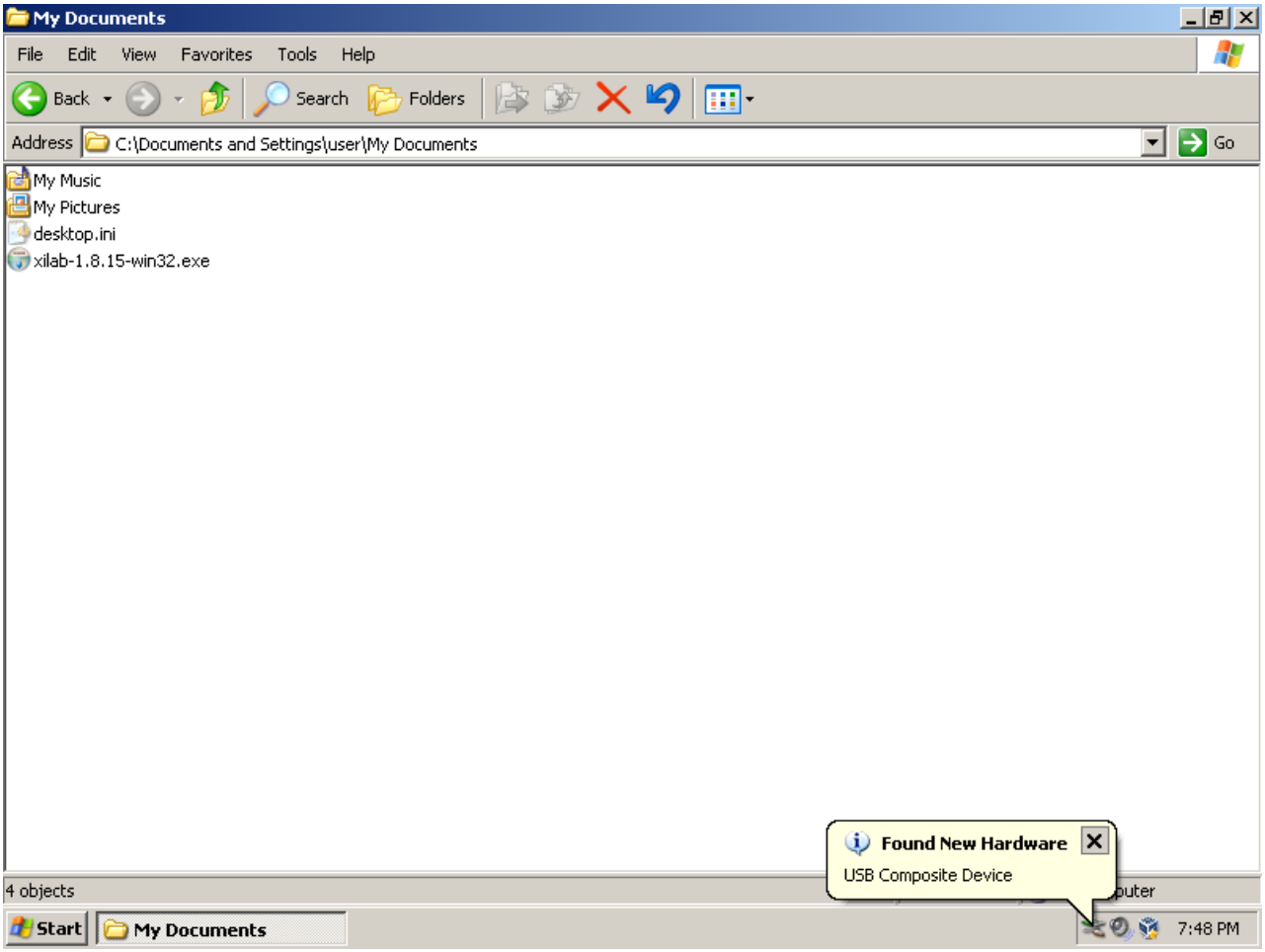


После установки по умолчанию запустится программа XiLab.



Подключите позиционер к контроллеру. Подключите стабилизированный источник питания к контроллеру. Заземлите контроллер или блок питания. Подключите контроллер к компьютеру используя USB-A – мини-USB-B кабель.

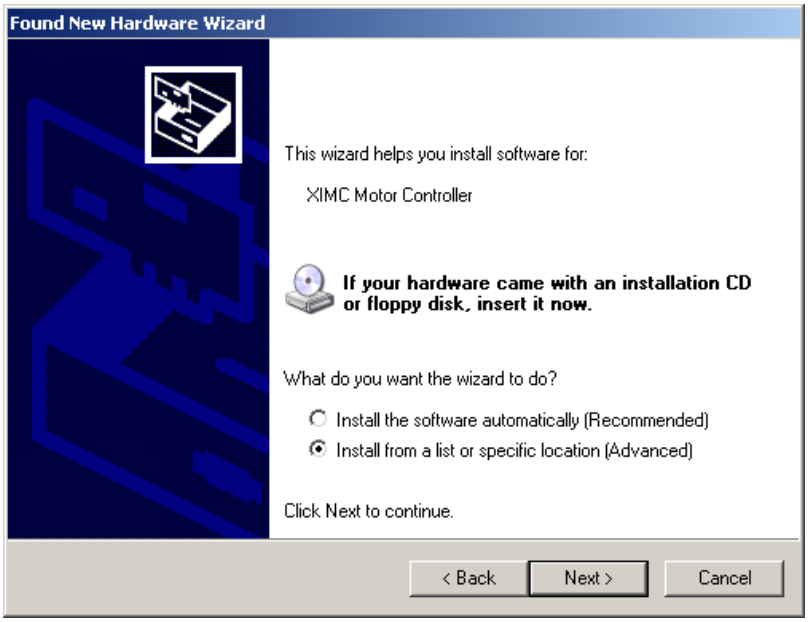
LED индикатор на плате контроллера начнет мигать. Мастер New Hardware Wizard начнет работать после первого подключения контроллера к компьютеру. Подождите пока Windows обнаружит новое устройство и установит необходимые драйверы для него.



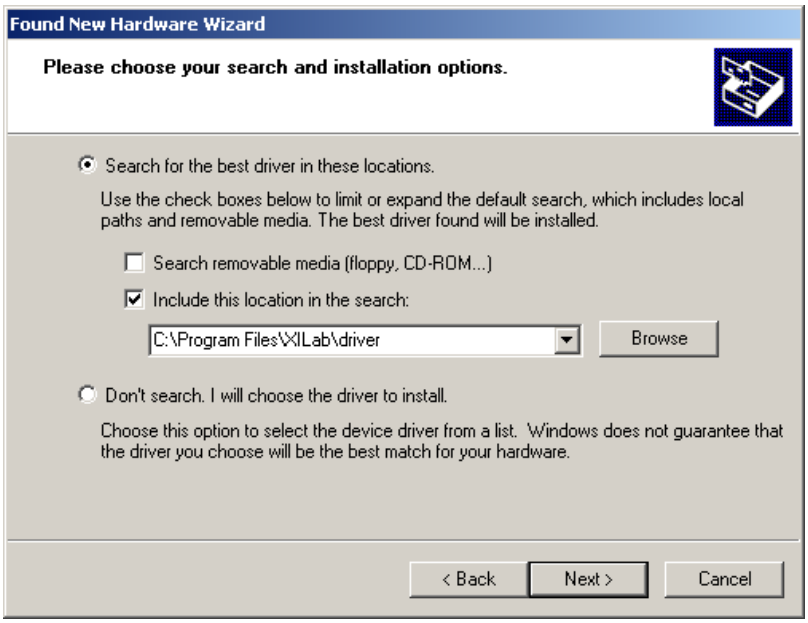
Если драйвер автоматически не установился, то в появившемся окне выберите ”No, not this time”, затем нажмите “Next >”.



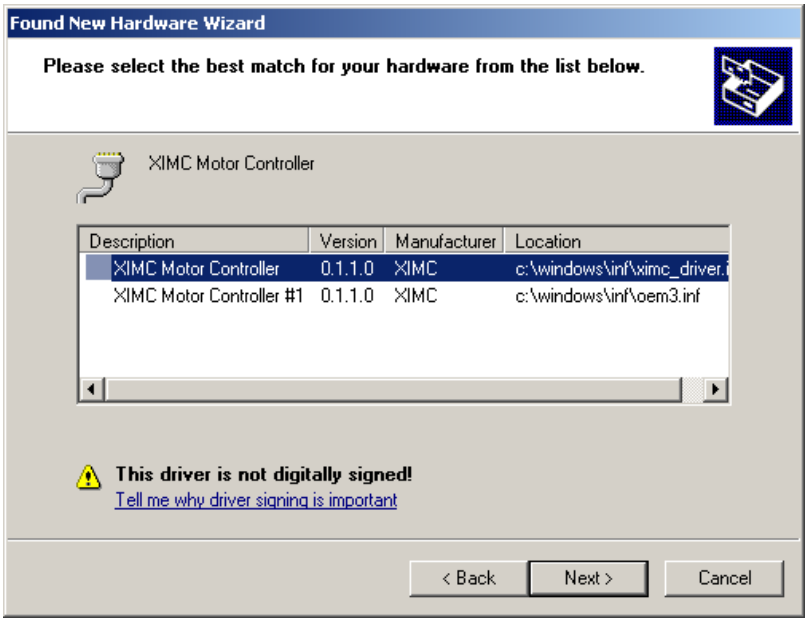
В следующем окне выберите “Install from a list or specific location (Advanced)” и нажмите “Next>”.



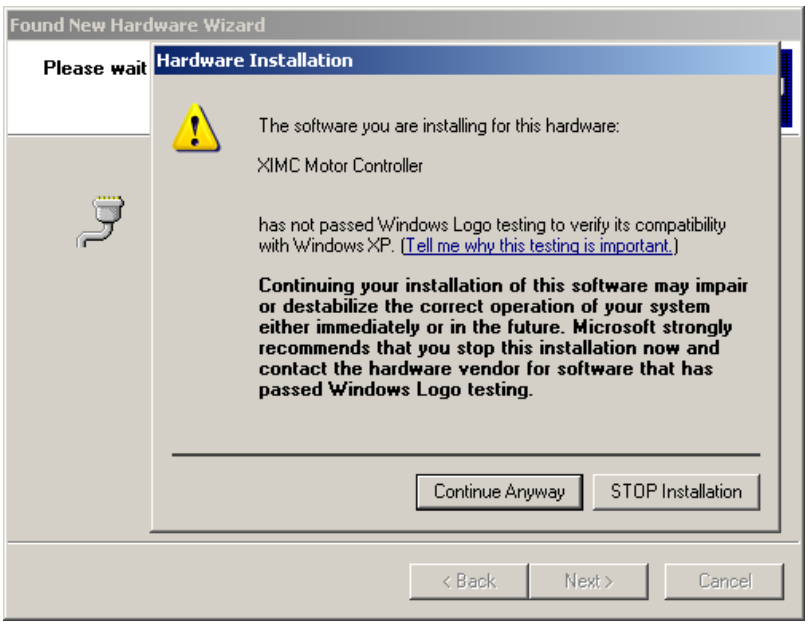
Выберите *.inf файл на диске с ПО, поставляемым с контроллером, или в директории установки программы (по умолчанию это C:\Program Files\XiLab\driver\) и нажмите "Next".



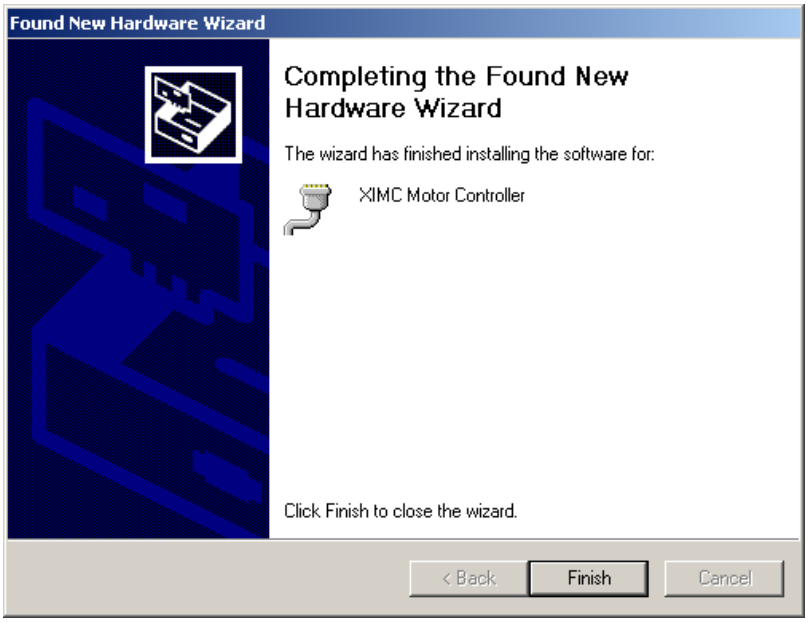
Нажмите "Next".



Нажмите "Continue anyway".

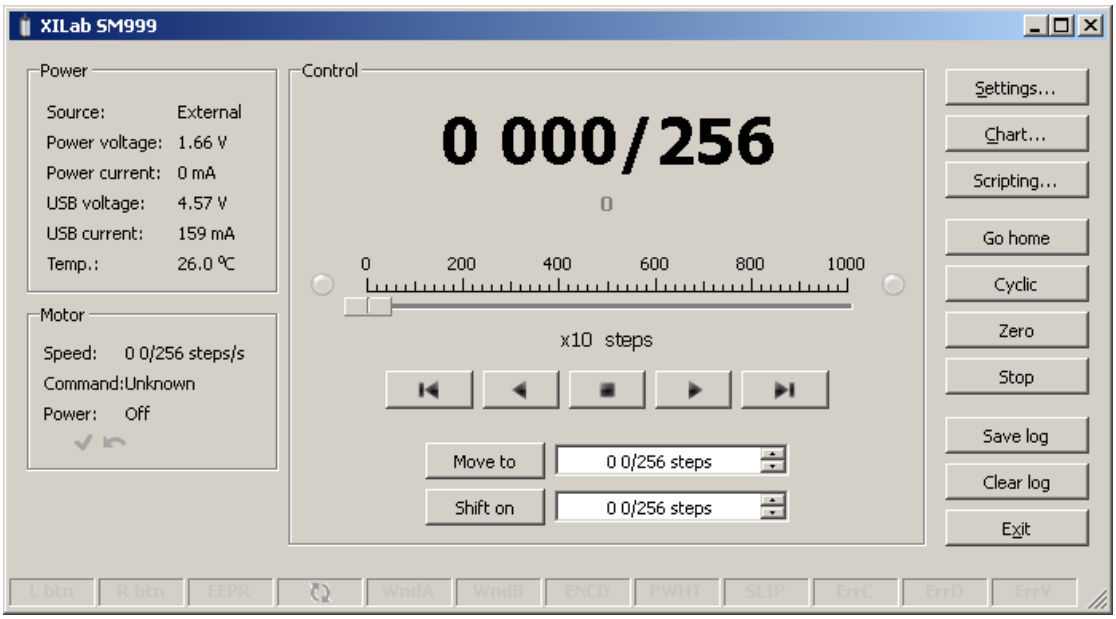


Нажмите "Finish". Установка драйвера завершена.



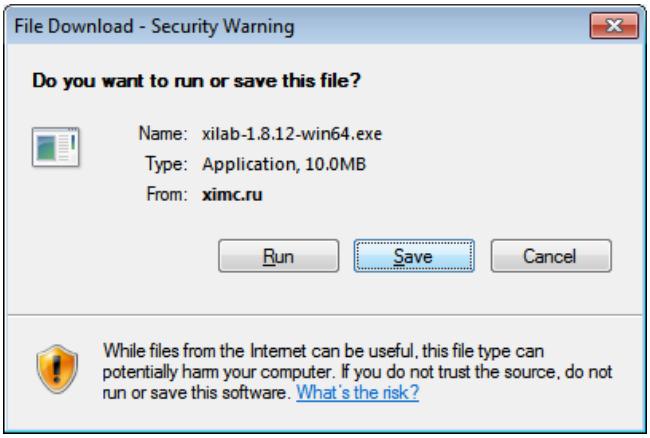
Нажмите кнопку Retry или снова запустите программу Xilab, если она была закрыта. Будет обнаружен подключенный контроллер и откроется главное окно программы.

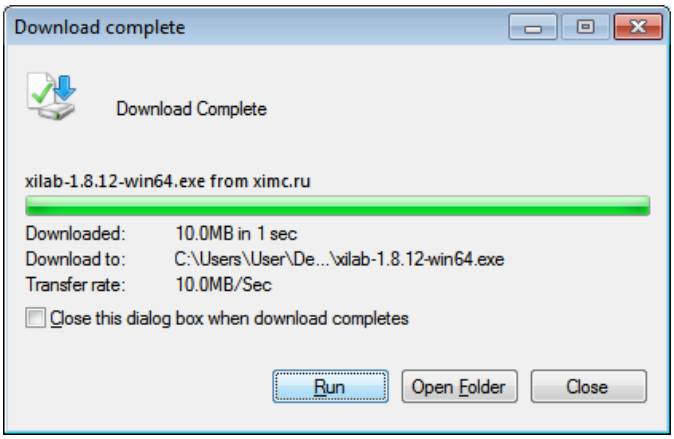




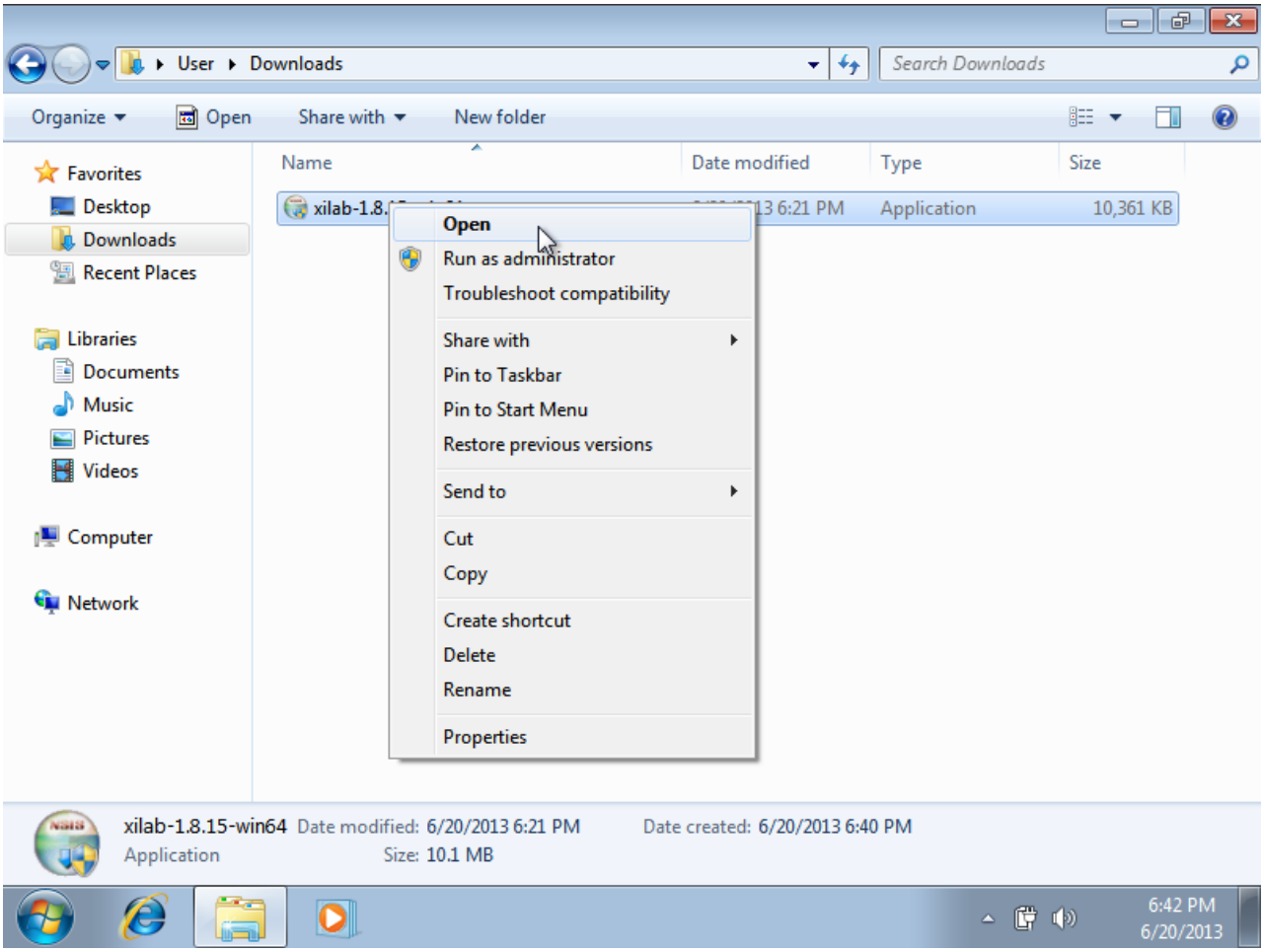
5.8.1.2. Установка под Windows 7

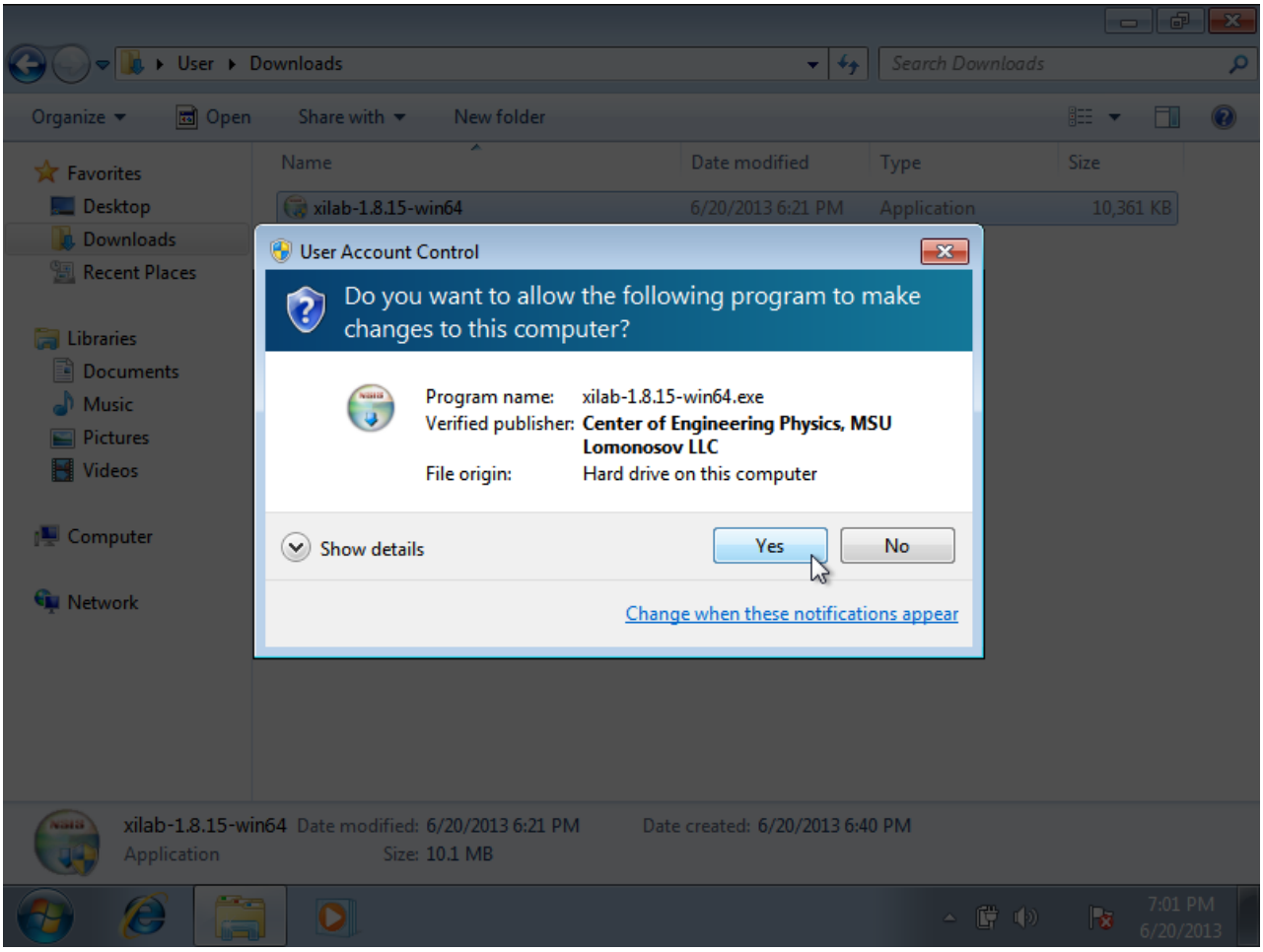
Скопируйте файл с программой установки на компьютер. Программа установки имеет название "xilab-<номер версии>.exe". Инсталлятор автоматически определяет, запущен ли он на 32-битной или 64-битной системе и устанавливает соответствующую версию.





Запустите программу установки и следуйте инструкциям на экране.


User ▸ Downloads
Search Downloads
Organize
Open
Share with
New folder
Favorites
Desktop
Downloads
Recent Places
Libraries
Documents
Music
Pictures
Videos
Computer
Network
Name
Date modified
Type
Size
xilab-1.8.
6:21 PM
Application
10,361 KB
Open
Run as administrator
Troubleshoot compatibility
Share with
Pin to Taskbar
Pin to Start Menu
Restore previous versions
Send to
Cut
Copy
Create shortcut
Delete
Rename
Properties
xilab-1.8.15-win64 Date modified: 6/20/2013 6:21 PM
Date created: 6/20/2013 6:40 PM
Application
Size: 10.1 MB
6:42 PM
6/20/2013


User ▸ Downloads
Search Downloads
Organize
Open
Share with
New folder
Favorites
Desktop
Downloads
Recent Places
Libraries
Documents
Music
Pictures
Videos
Computer
Network
Name
Date modified
Type
Size
xilab-1.8.15-win64
6/20/2013 6:21 PM
Application
10,361 KB
User Account Control
Do you want to allow the following program to make changes to this computer?
Program name: xilab-1.8.15-win64.exe
Verified publisher: Center of Engineering Physics, MSU Lomonosov LLC
File origin: Hard drive on this computer
Show details
Yes
No
Change when these notifications appear
xilab-1.8.15-win64 Date modified: 6/20/2013 6:21 PM
Date created: 6/20/2013 6:40 PM
Application
Size: 10.1 MB
7:01 PM
6/20/2013

Все необходимое программное обеспечение, включая драйверы, пакеты и программы будут установлены автоматически.

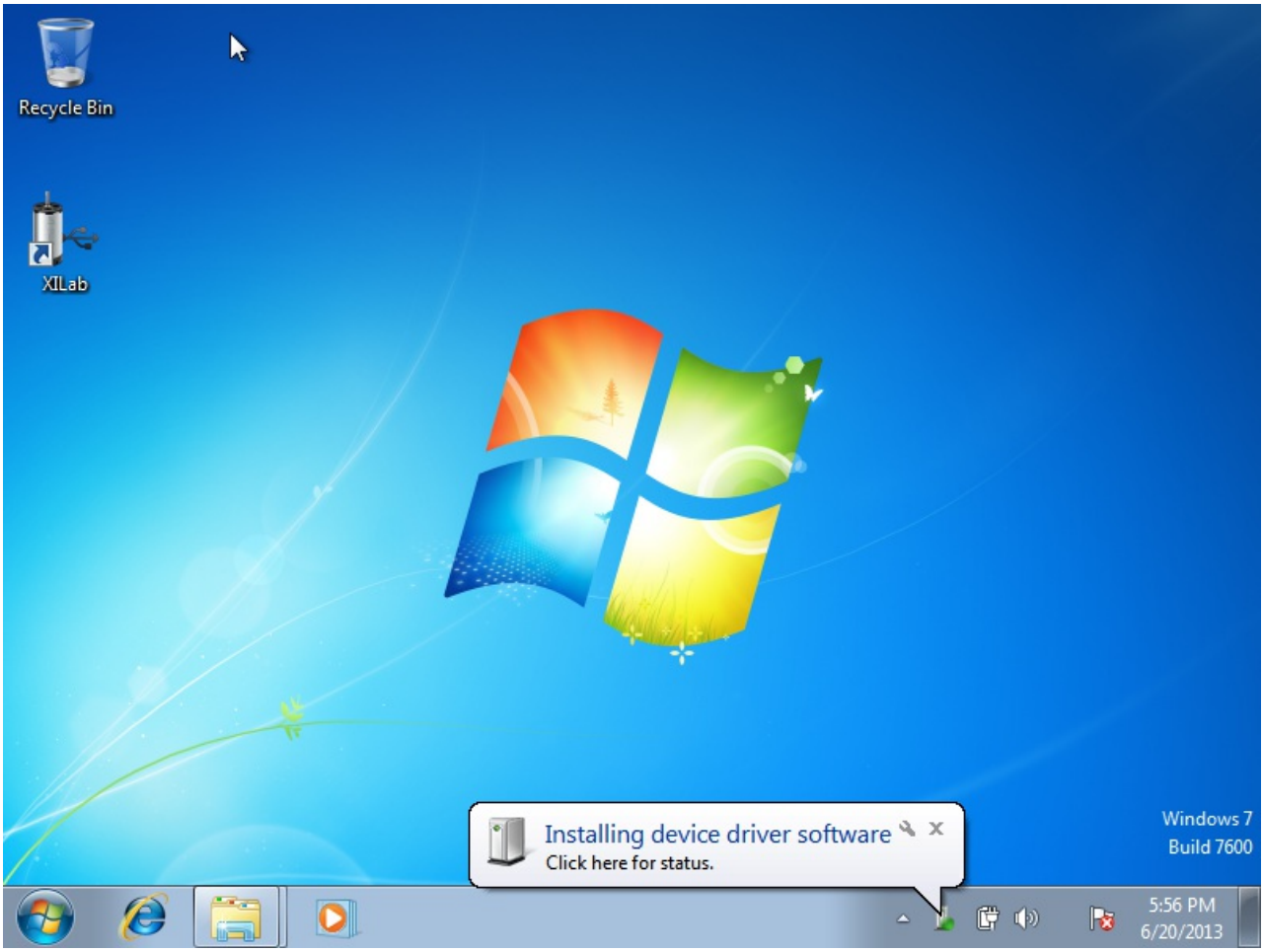


После установки по умолчанию запустится программа XiLab.



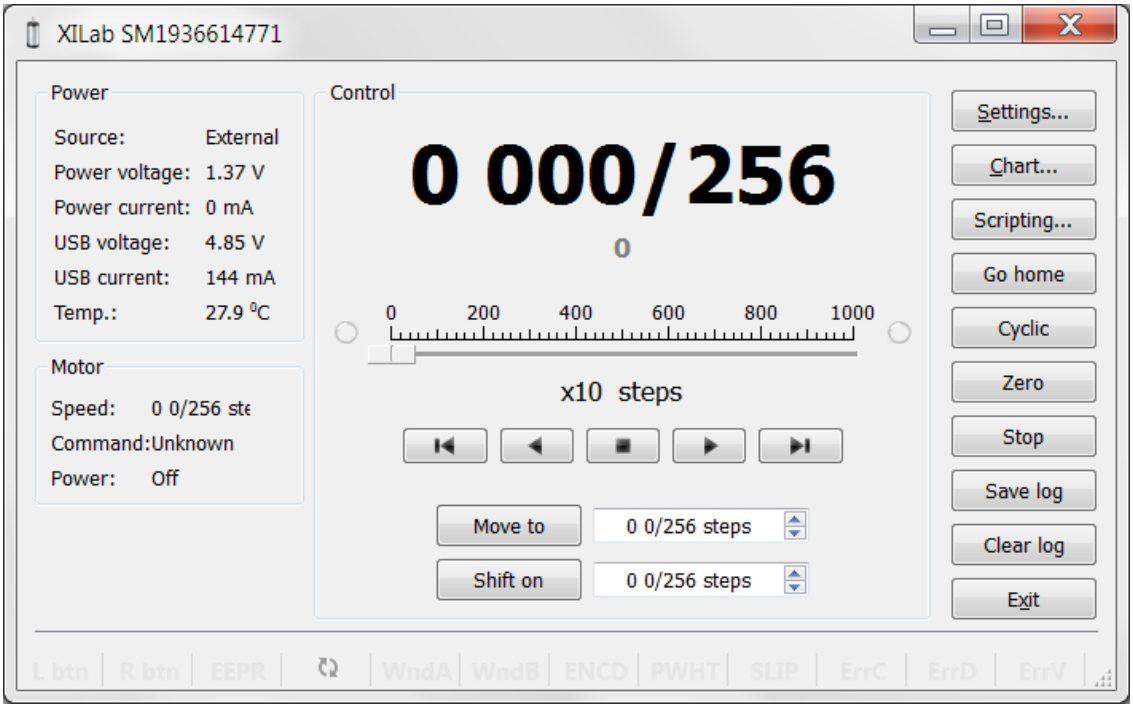
Подключите позиционер к контроллеру. Подключите стабилизированный источник питания к контроллеру. Заземлите контроллер или блок питания. Подключите контроллер к компьютеру используя USB-A – мини-USB-B кабель.

LED индикатор на плате контроллера начнет мигать. Мастер New Hardware Wizard начнет работать после первого подключения контроллера к компьютеру.



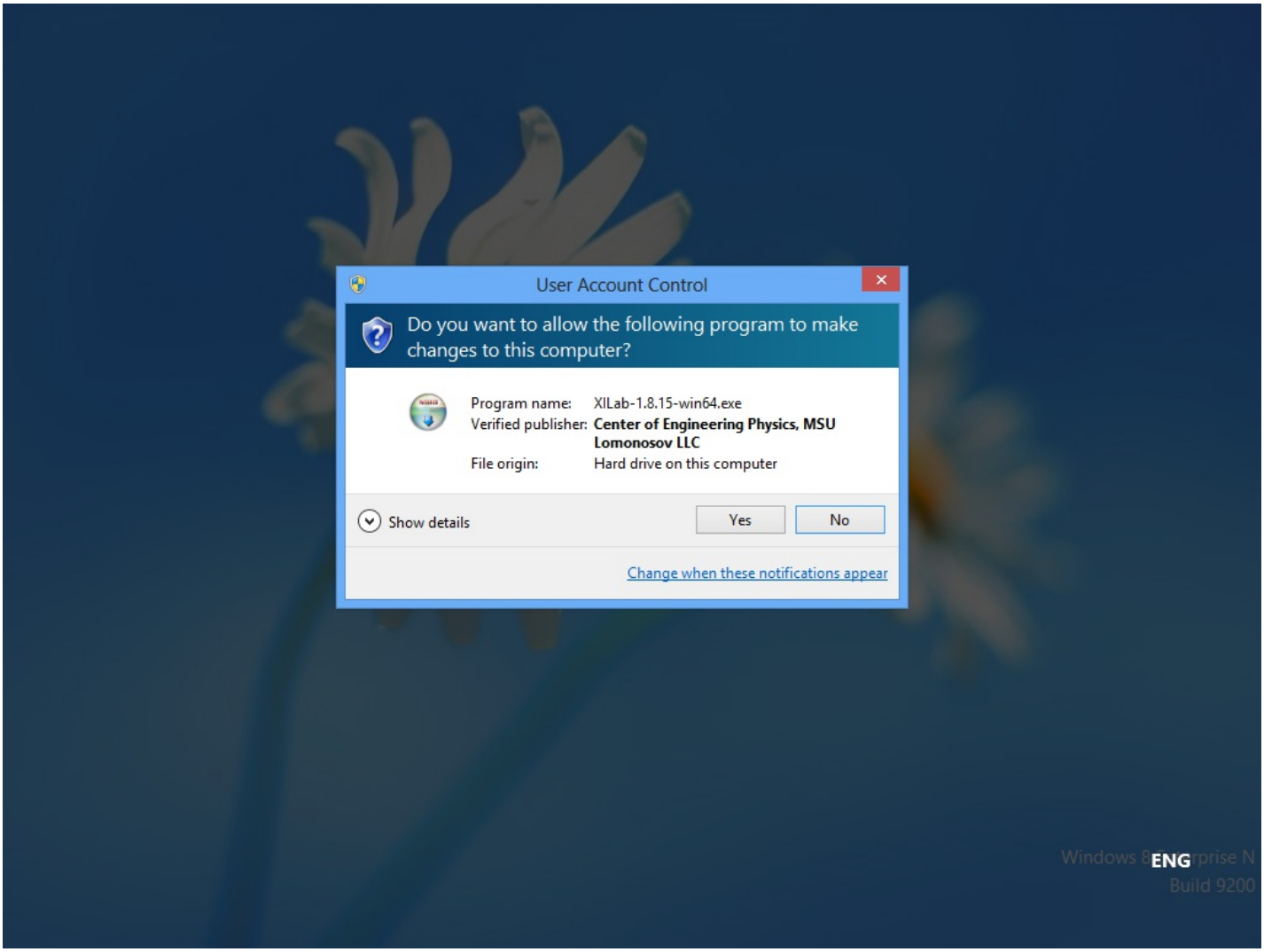
Подождите пока Windows обнаружит новое устройство и установит необходимые драйверы для него. После успешной установки драйвера нажмите кнопку Retry или снова запустите программу Xilab, если она была закрыта. Будет обнаружен подключенный контроллер и откроется главное окно программы.



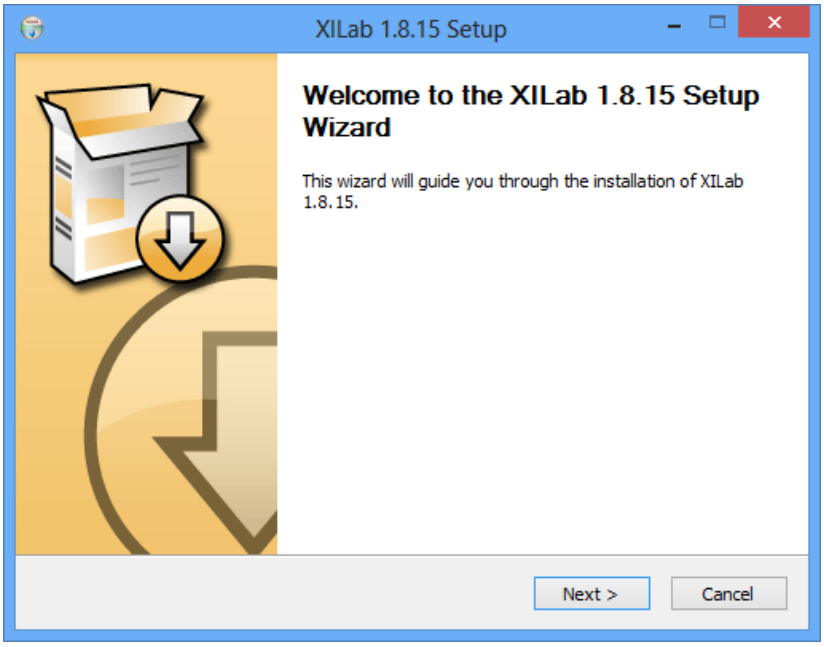

XILab SM1936614771
Power
Source: External
Power voltage: 1.37 V
Power current: 0 mA
USB voltage: 4.85 V
USB current: 144 mA
Temp.: 27.9 °C
Motor
Speed: 0 0/256 ste
Command:Unknown
Power: Off
Control
0 000/256
0
0 200 400 600 800 1000
x10 steps
Move to
0 0/256 steps
Shift on
0 0/256 steps
Settings...
Chart...
Scripting...
Go home
Cyclic
Zero
Stop
Save log
Clear log
Exit
L btn R btn EEPR WndA WndB ENCD PWHT SLIP ErrC ErrD ErrV

5.8.1.3. Установка под Windows 8

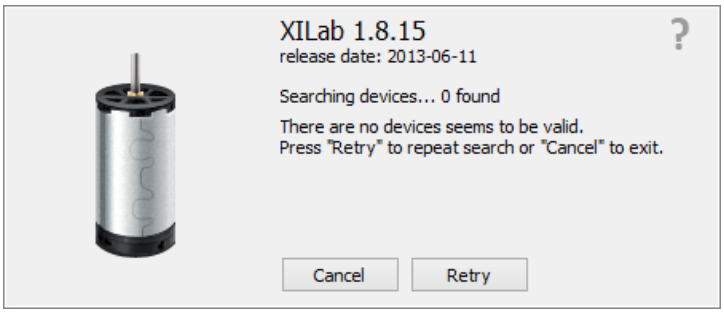
Скопируйте файл с программой установки на компьютер. Программа установки имеет название "xilab-<номер версии>.exe". Инсталлятор автоматически определяет, запущен ли он на 32-битной или 64-битной системе и устанавливает соответствующую версию.



Все необходимое программное обеспечение, включая драйверы, пакеты и программы будут установлены автоматически.



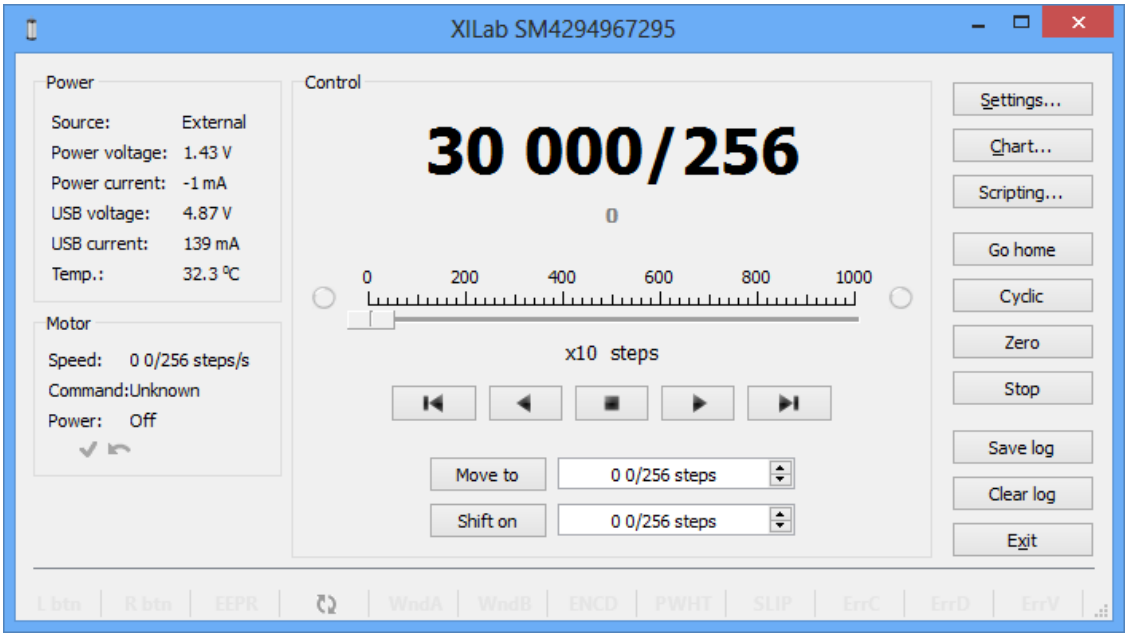
После установки по умолчанию запустится программа XiLab.



Подключите позиционер к контроллеру. Подключите стабилизированный источник питания к контроллеру. Заземлите контроллер или блок питания. Подключите контроллер к компьютеру используя USB-A – мини-USB-B кабель.

LED индикатор на плате контроллера начнет мигать. Мастер New Hardware Wizard начнет работать после первого подключения контроллера к компьютеру. Подождите пока Windows обнаружит новое устройство и установит необходимые драйверы для него. После успешной установки драйвера нажмите кнопку Retry или снова запустите программу Xilab, если она была закрыта. Будет обнаружен подключенный контроллер и откроется главное окно программы.





5.8.2. Установка под Linux

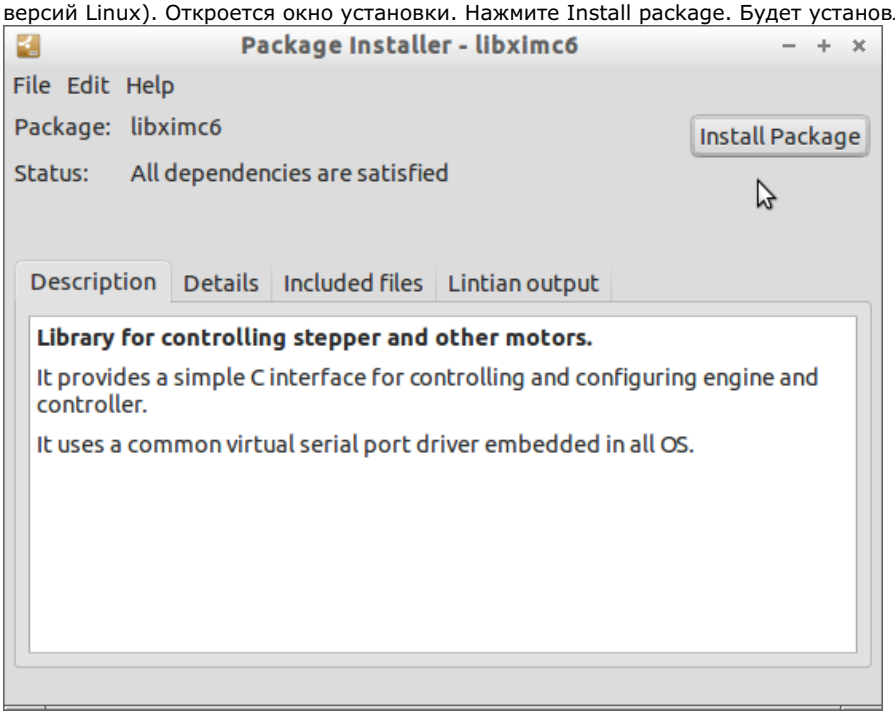
5.8.2. Установка под Linux
Debian/Ubuntu
Установка в графическом режиме
Установка в текстовом режиме
RedHat/OpenSUSE
Установка в текстовом режиме

Пакет XILab устанавливает свои файлы в следующие директории:
/usr/bin/xilab - исполняемый файл
/usr/share/icons/xilab.png - иконка
/usr/share/xilab/scripts/ - директория скриптов
/usr/share/xilab/profiles/ - директория с профилями
/usr/share/xilab/xilabdefault.cfg - файл с настройками по умолчанию

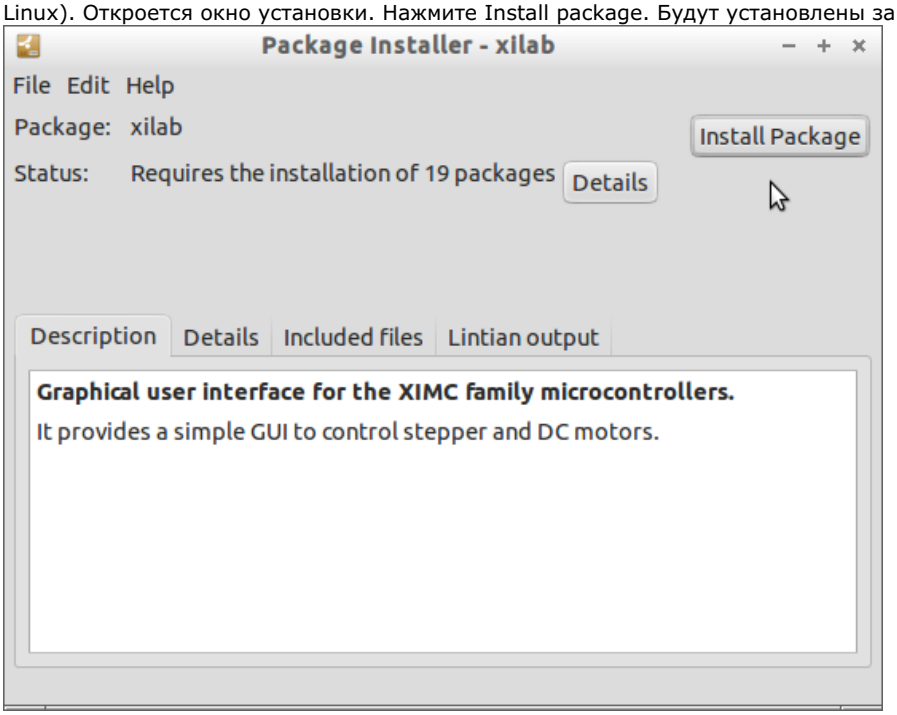
Debian/Ubuntu

Установка в графическом режиме

Кликните на файле libximc6_x.y.z-1_i386.deb (для 32-битных версий Linux) или libximc6_x.y.z-1_amd64.deb (для 64-битных версий Linux). Откроется окно установки. Нажмите Install package. Будет установлена библиотека libximc.



Кликните на файле xilab_x.y.z-1_i386.deb (для 32-битных версий Linux) или xilab_x.y.z-1_amd64.deb (для 64-битных версий Linux). Откроется окно установки. Нажмите Install package. Будут установлены зависимости и программа xilab.



Установка в текстовом режиме

Из-под суперпользователя (root) выполните следующие команды:

```
gdebi "<FILEPATH>/libximc6_<LIBVERSION>-1_<ARCH>.deb"
gdebi "<FILEPATH>/xilab-<VERSION>-1_<ARCH>.deb"
```

где <FILEPATH> это путь к файлам пакетов (например "/home/user/Downloads/"), <LIBVERSION> и <VERSION> это номер версии библиотеки libximc и программы xilab соответственно (например "2.0.2" и "1.8.12"), а <ARCH> - идентификатор архитектуры ("i386" для 32-битных систем и "amd64" для 64-битных систем).
Пример:

```
gdebi "/home/user/Downloads/libximc6-2.0.2-1_amd64.deb"
gdebi "/home/user/Downloads/xilab-1.8.12-1_amd64.deb"
```

Программа xilab требует наличия X-сервера (графического режима) для работы с ней.

RedHat/OpenSUSE

Установка в текстовом режиме

Из-под суперпользователя (root) выполните следующие команды:

```
zypper install "<FILEPATH>/libximc6-<LIBVERSION>-1.<ARCH>.rpm"
zypper install "<FILEPATH>/xilab-<VERSION>-1.<ARCH>.rpm"
```

где <FILEPATH> это путь к файлам пакетов (например "/home/user/Downloads/"), <LIBVERSION> и <VERSION> это номер версии библиотеки libximc и программы xilab соответственно (например "2.0.2" и "1.8.12"), а <ARCH> - идентификатор архитектуры ("i686" для 32-битных систем и "x86_64" для 64-битных систем).
Пример:

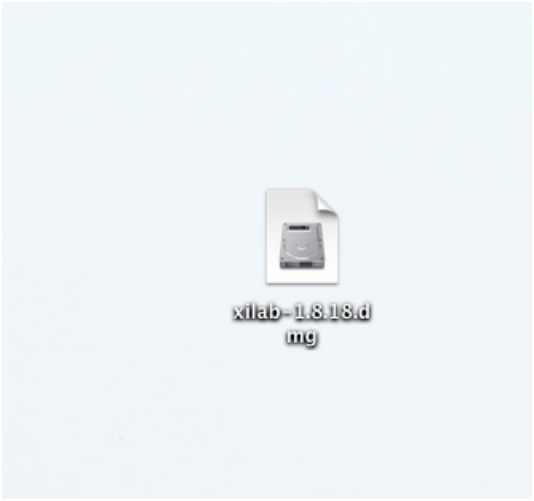
```
zypper install "/home/user/Downloads/libximc6-2.0.2-1.x86_64.rpm"
zypper install "/home/user/Downloads/xilab-1.8.12-1.x86_64.rpm"
```

Программа xilab требует наличия X-сервера (графического режима) для работы с ней.

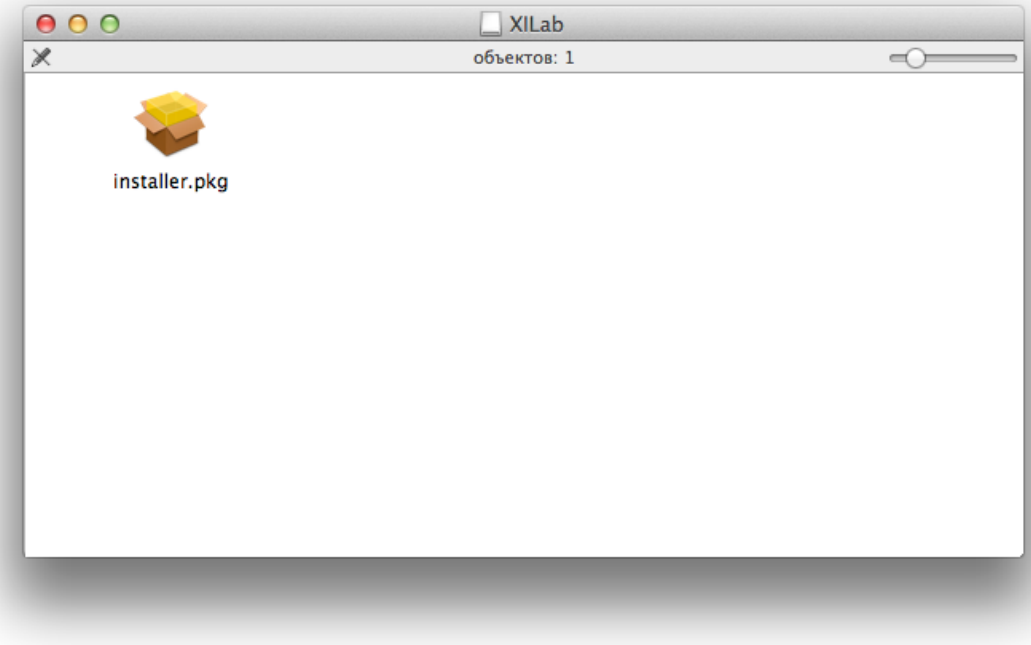
5.8.3. Установка под MacOS



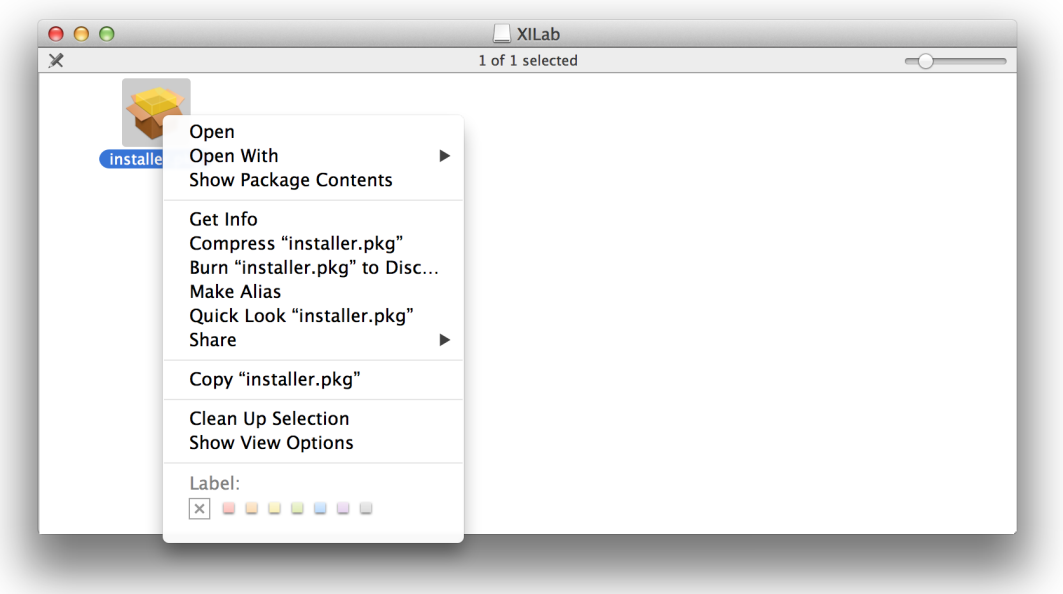
Скопируйте файл с архивом программы установки на компьютер. Архив с программой установки имеет название "xilab-<номер версии>-osx64.tar.gz".



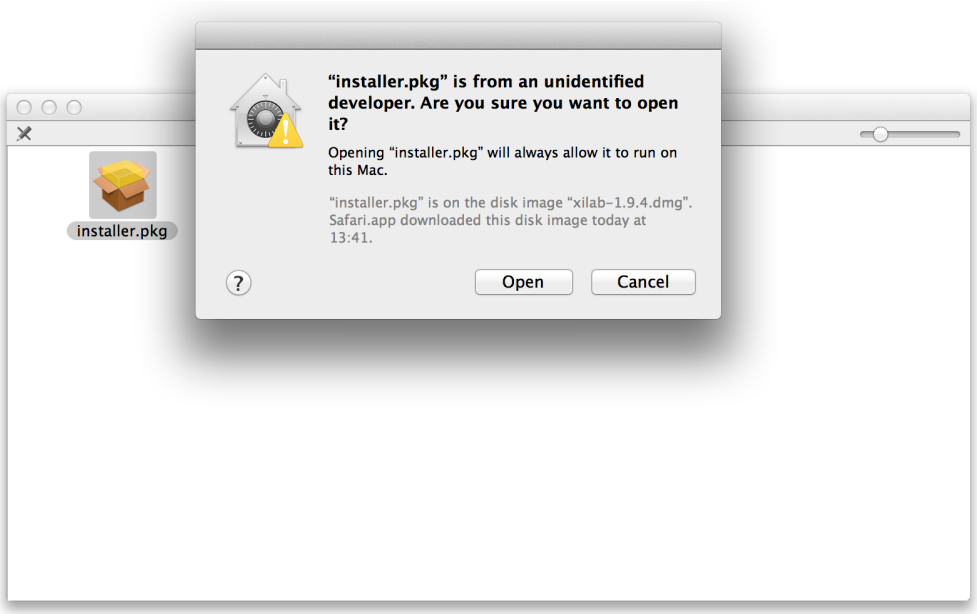
Распакуйте архив щелчком мыши.



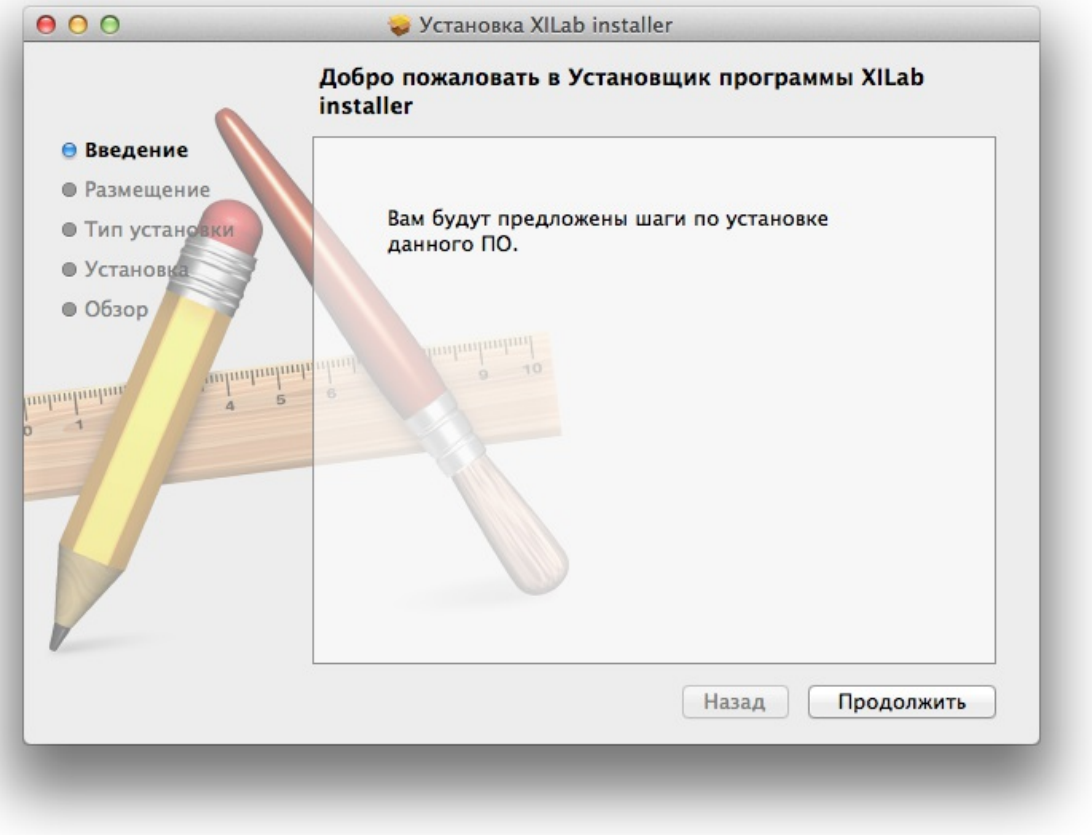
Щелкните правой кнопкой мыши на появившемся installer.pkg.



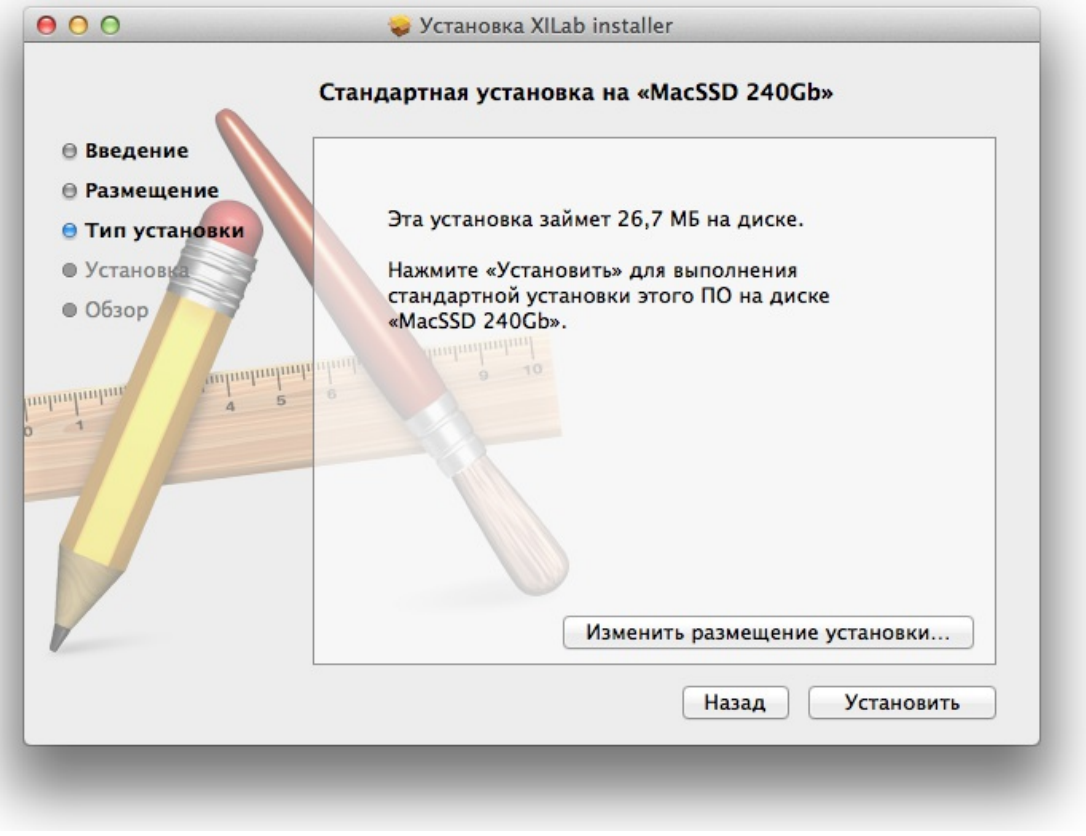
Выберите Открыть.



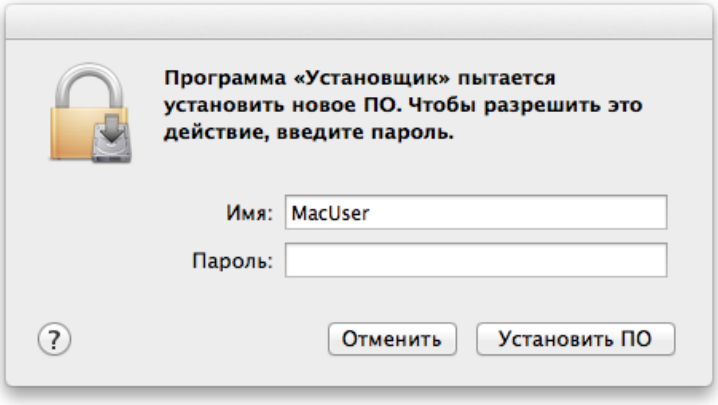
В появившемся окне так же выберите Открыть.



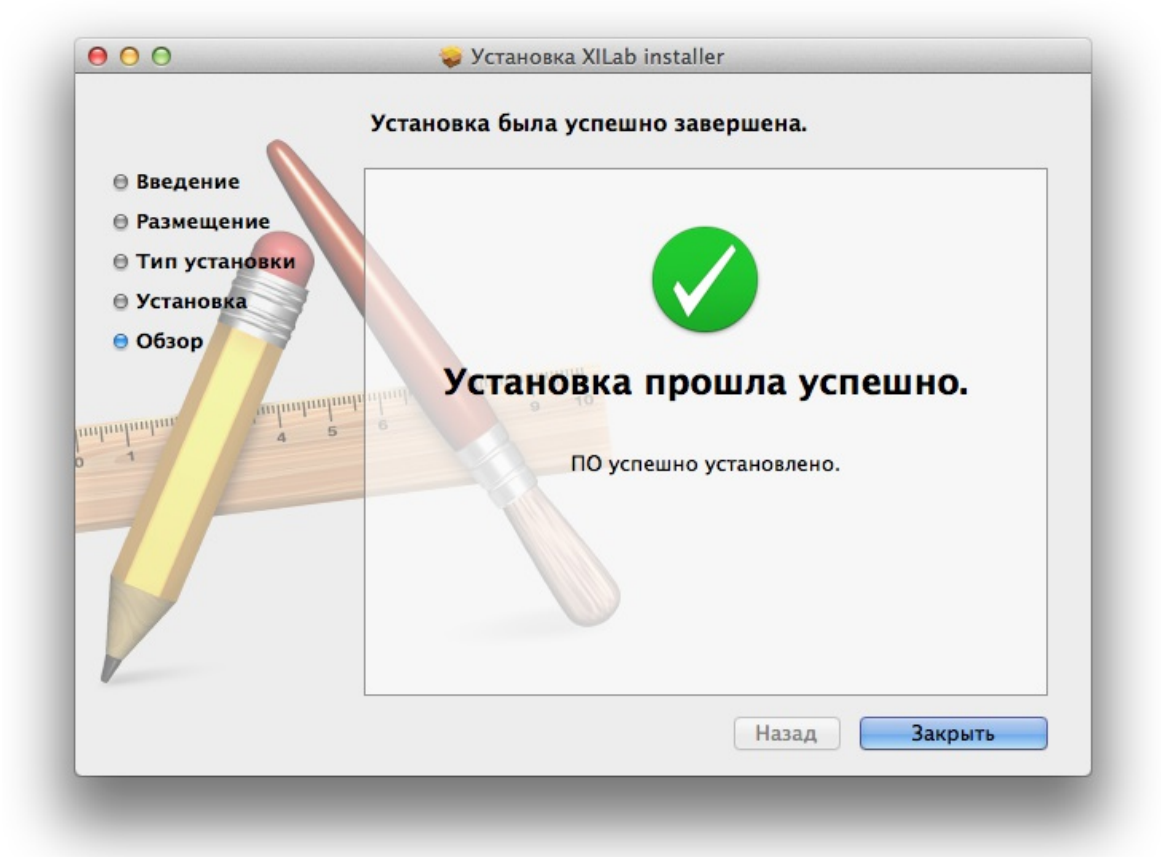
В главном окне установщика выберите "Продолжить".



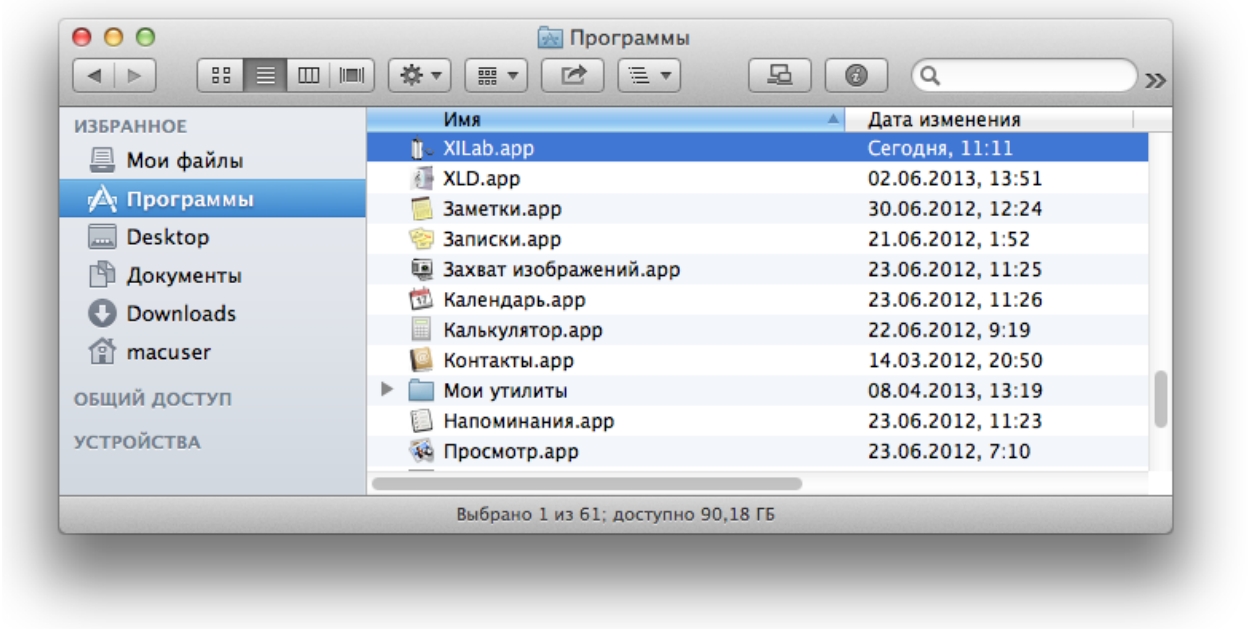
Далее выберите "Установить".



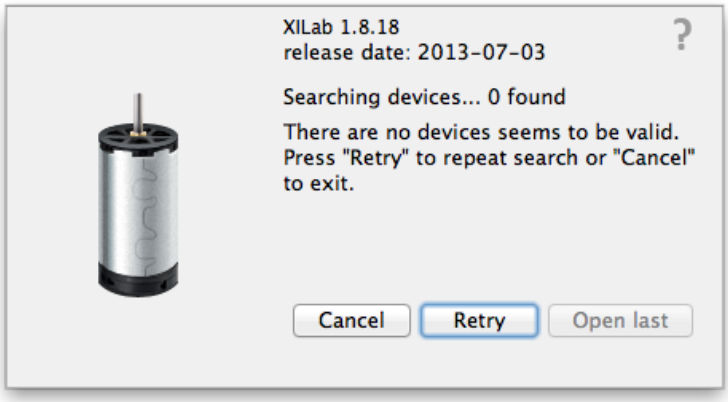
Введите пароль.



Дождитесь успешного завершения установки.



Выберите приложение XILab в разделе "Программы".



Запустите.

6. Программирование

1. Руководство по программированию
    1. Работа с контроллером в среде LabView
    2. Работа с контроллером в среде Matlab
2. Описание протокола обмена
3. Совместимость с ПО для 8SMC1-USBhF
4. Таймауты libximc
5. Скрипты XILab

6.1. Руководство по программированию

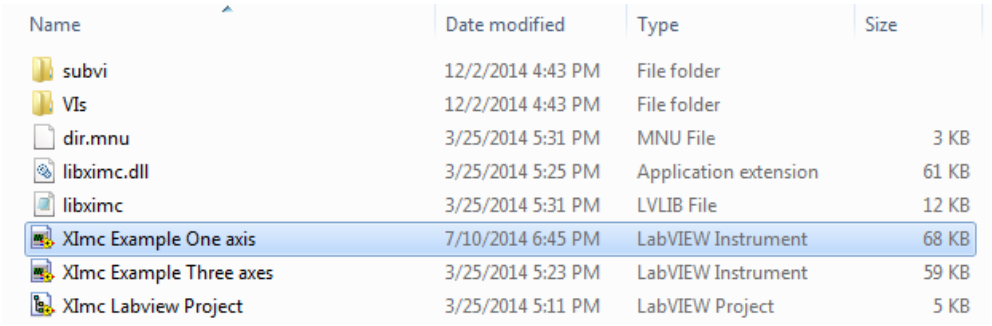
Руководство по программированию входит в комплект разработчика libximc 2.X.X, где 2.X.X - номер версии, размещено в папке /ximc-2.X.X/ximc/doc-ru/libximc7-ru.pdf. Так же оно может быть скачано по этой ссылке. Скачать комплект разработчика можно в разделе Программное обеспечение. Руководство по программированию создано в системе Doxygen.

Работа с контроллером в среде LabView

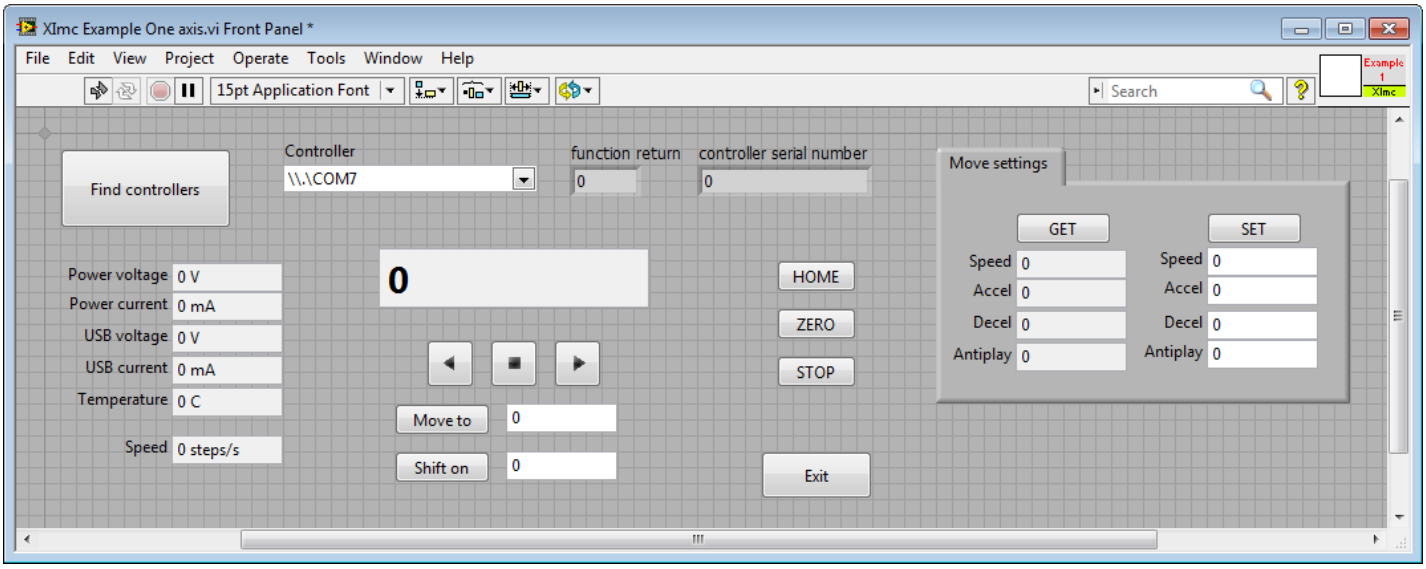
Работа с контроллером в среде Matlab

6.1.1. Работа с контроллером в среде LabView

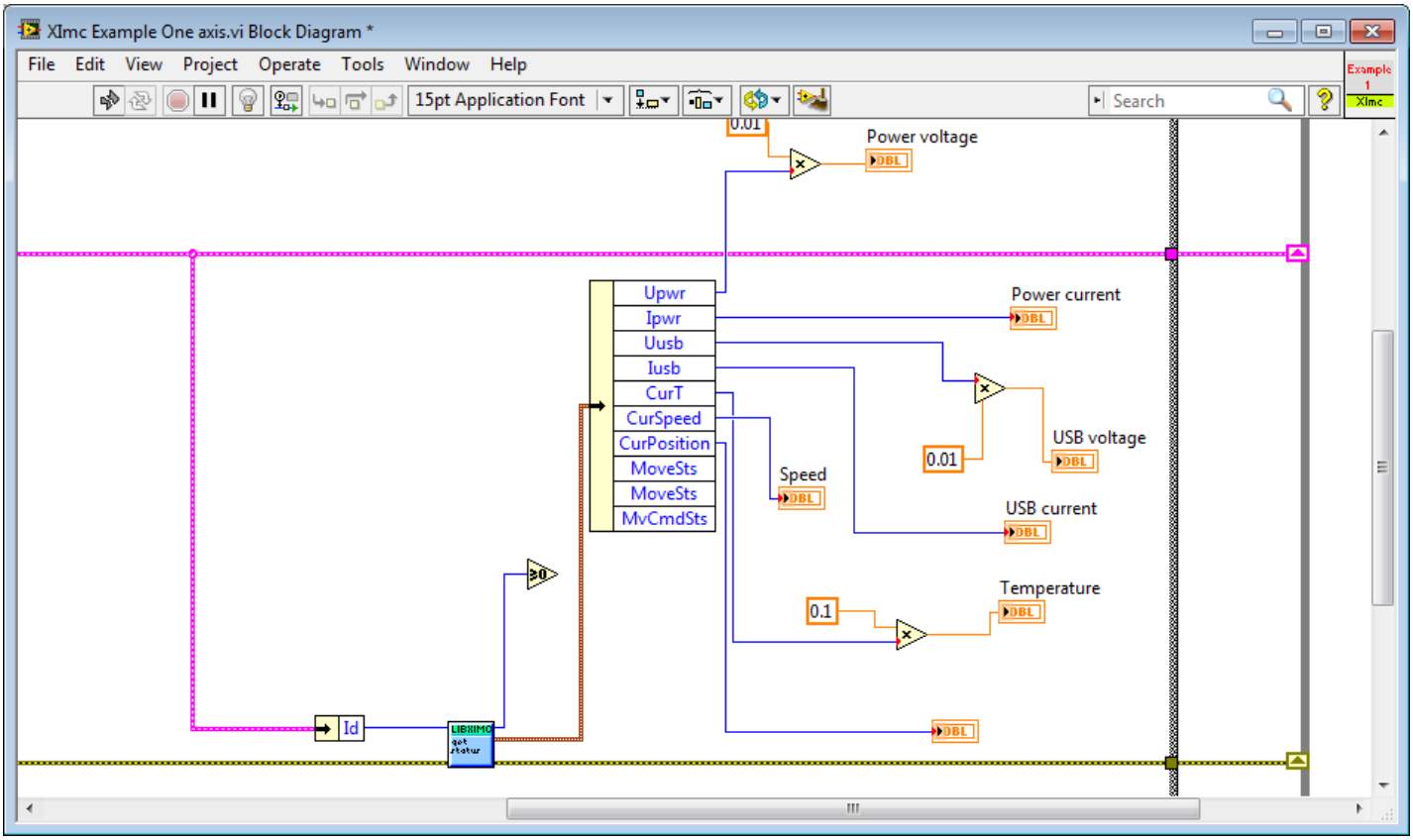
Скачайте пример программы для LabView со страницы Программное обеспечение.
Разархивируйте скачанный файл в желаемую директорию и запустите пример "XImc Example One axis".



Откроется окно среды LabView. В нем вы увидите графический интерфейс передней панели программы-примера, он представляет собой упрощенную копию интерфейса XILab.
В левой части расположена кнопка "Find controllers" для повторного опроса доступных контроллеров, поле выбора контроллера по имени последовательного порта и информационный блок текущего состояния открытого контроллера (напряжение и ток на силовой части и USB, температура, скорость движения).
В центральной части расположен блок индикации и управления. В числовое поле выводится текущая координата; доступны кнопки движения влево, вправо, остановки, движения в определенную координату ("Move to") и смещения на определенное расстояние ("Shift on").
Справа от этого блока расположены кнопки поиска домашней позиции "HOME", обнуления текущей координаты "ZERO", быстрой остановки "STOP" и под ними кнопка "Exit" для корректного завершения работы примера (см. ниже).
На вкладке справа расположен диалог "Move settings" для демонстрации загрузки и сохранения настроек. По нажатию кнопки "GET" текущие настройки движения загружаются в числовые поля под этой кнопкой, а по нажатию кнопки "SET" числа из соответствующих этой кнопке полей загружаются в контроллер.

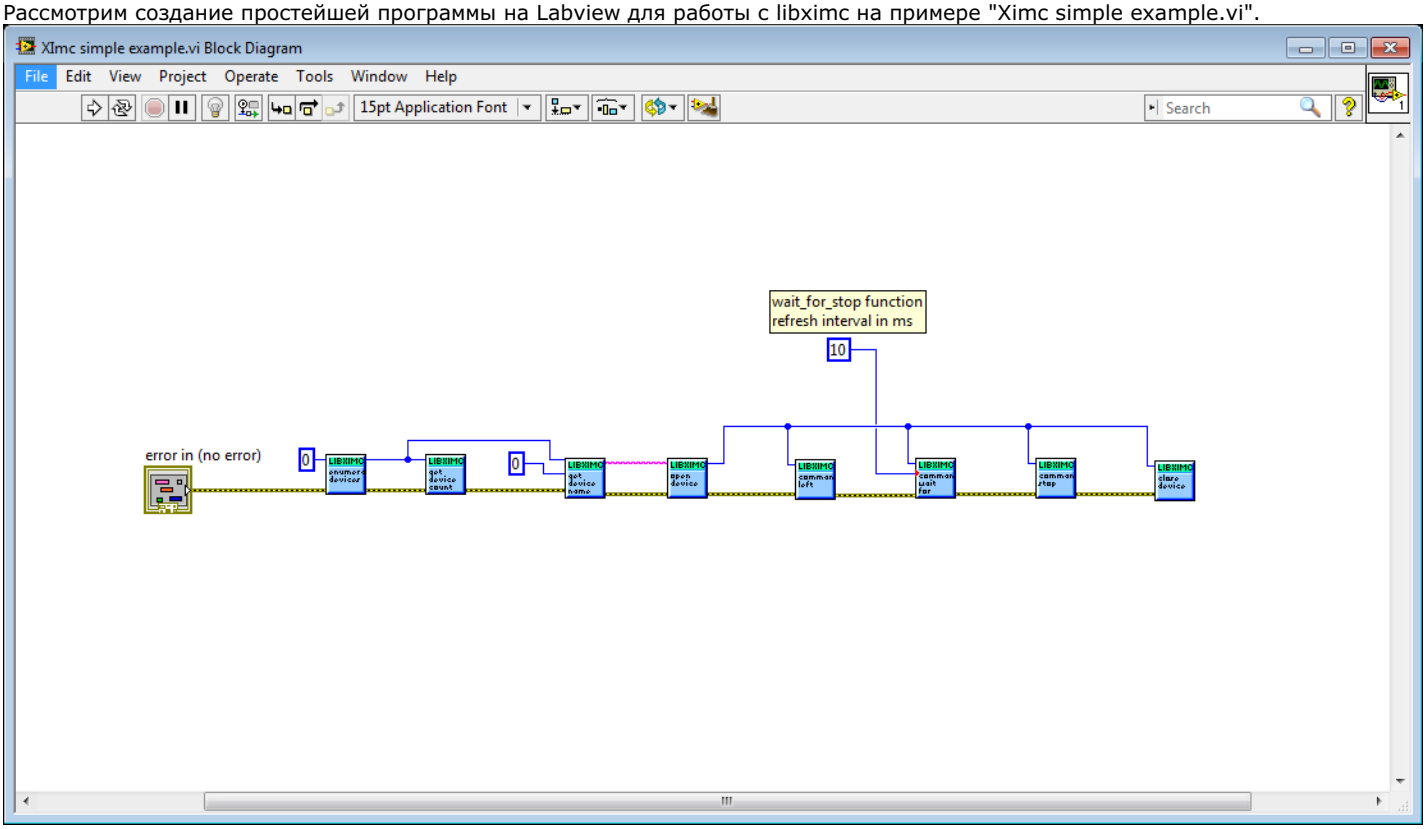


Исходный код примера можно посмотреть войдя в режим редактирования. Программа выполняет бесконечный цикл опроса состояния контроллера и вывода на экран. Если нажата какая-либо кнопка в интерфейсе, то выполняется соответствующий этой кнопке функциональный блок.



Пример использует функции библиотеки libximc. Каждой функции соответствует свой subVI модуль, имеющий входы и выходы соответствующие входным и выходным параметрам этой функции.
Для того чтобы вызывать какую-либо функцию нужно сначала выполнить опрос устройств (enumerate_devices), после этого выбрать какое-либо устройство из доступных, открыть его с помощью open_device и передать полученный идентификатор и необходимые параметры желаемой функции. После использования необходимо закрыть устройство функцией close_device.

Рассмотрим создание простейшей программы на Labview для работы с libximc на примере "Ximc simple example.vi".



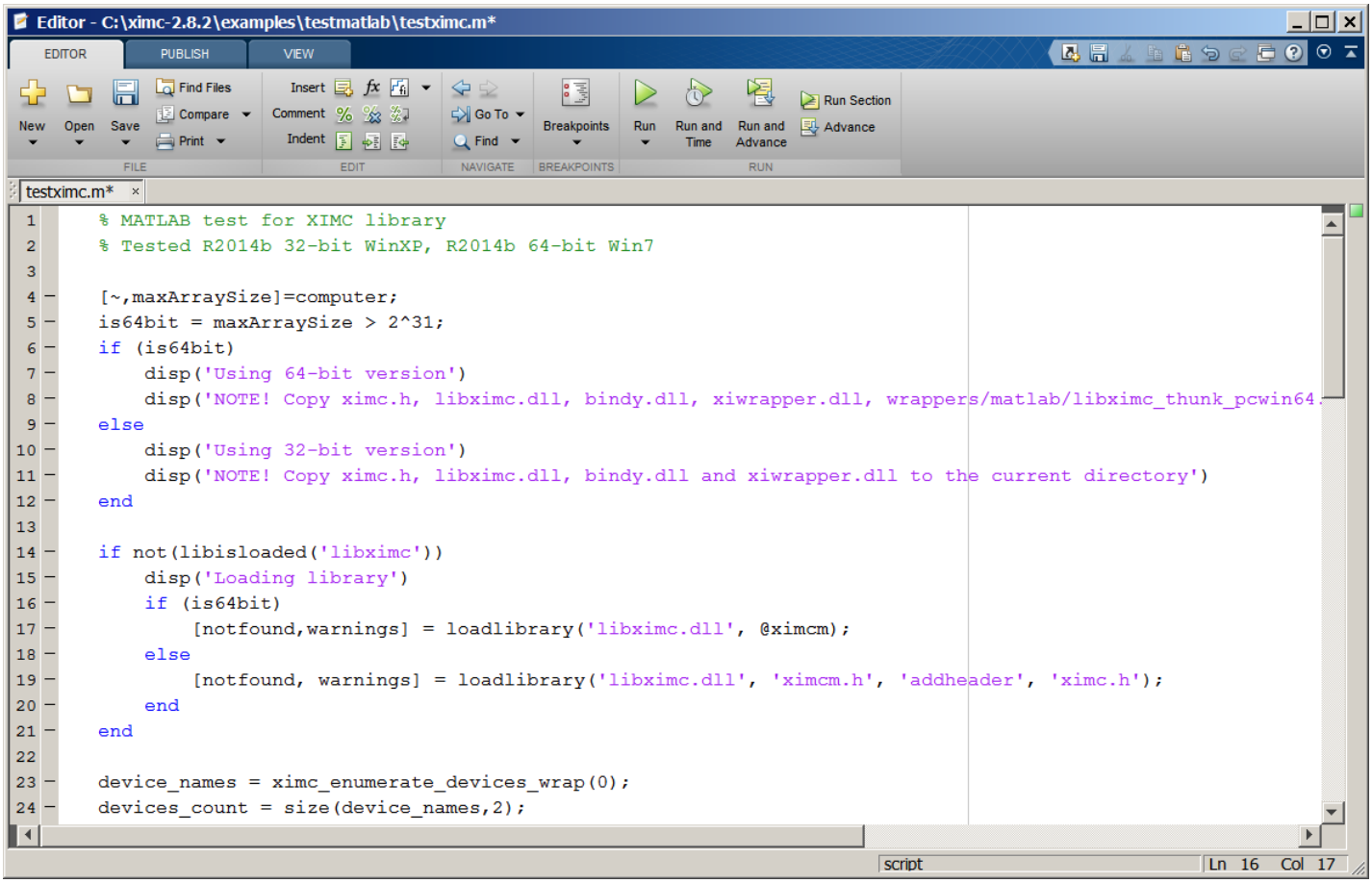
В начале программы вызывается функция enumerate_devices, которой передаются флаги открытия (подробнее см. Руководство по программированию). Результат выполнения функции enumerate_devices, являющийся закрытым указателем, передается функции get_device_name с параметром номера устройства, имя которого мы хотим получить (количество обнаруженных устройств можно получить функцией get_device_count из этого же указателя). Результат выполнения функции get_device_name передается функции open_device. Эта последовательность не является обязательной - при желании сформированную определенным образом строку с именем открываемого устройства можно передать функции open_device напрямую.
Результатом выполнения функции open_device является хендл устройства или определенная в ximc.h константа device_undefined, возвращаемое при невозможности открыть устройство. Хендл устройства передается всем функциям чтения значений из контроллера, записи значений в контроллер и подачи команд контроллеру, вместе с другими параметрами, если таковые требуются в соответствии с прототипом функции. В примере "Ximc simple example.vi" вызываются команды command_left, command_wait_for_stop и command_stop. После окончания работы с контроллером устройство необходимо закрыть, передав его хендл функции close_device, а после окончания работы с результатом функции enumerate_devices необходимо освободить память закрытого указателя функцией free_enumerate_devices (опущено в примере для краткости).

> *Замечание.* Библиотека libximc открывает контроллеры в режиме эксклюзивного доступа. Любой контроллер открытый с помощью libximc необходимо закрыть прежде чем он может быть использован другими процессами. Не останавливайте работу примера LabView или любой другой LabView программы использующей libximc с помощью кнопки "Abort execution" - это не дает возможности программе вызвать функцию close_device(), поэтому в этом случае все открытые в LabView контроллеры будут заблокированы до полного закрытия среды LabView.

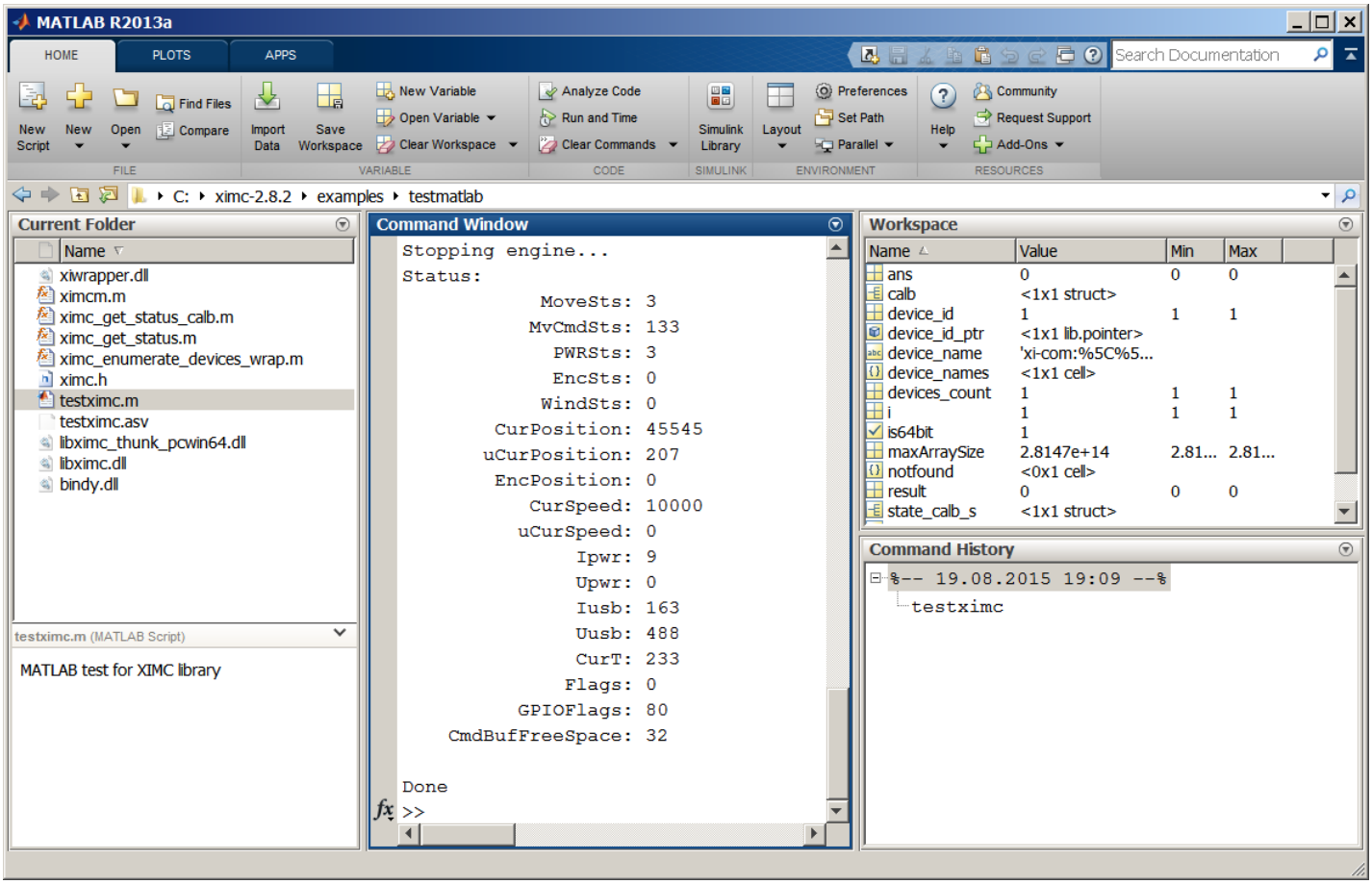
6.1.2. Работа с контроллером в среде Matlab

Библиотека libximc может использоваться для работы с контроллерами моторов в среде Matlab.

> *Замечание:* Для работы с SDK требуется Microsoft Visual C++ Redistributable Package 9.0.307291 (поставляется с SDK, файлы vcredist_x86 или vcredist_x64).



Распакуйте архив комплекта разработчика библиотеки libximc, скопируйте `ximc.h`, `win64\libximc.dll`, `win64\bindy.dll`, `win64\xiwrapper.dll`, `win64\wrappers\matlab\libximc_thunk_pcwin64.dll` и `win64\wrappers\matlab\ximc.m` в директорию `examples\testmatlab` если вы используете 64-битную среду или `ximc.h`, `win32\libximc.dll`, `win32\bindy.dll`, `win32\xiwrapper.dll` и `win32\wrappers\matlab\ximcm.h` если вы используете 32-битную. Подключите контроллер моторов к компьютеру и запустите пример `testximc.m`.



В окне выполненных команд вы увидите результат опроса контроллеров.
Вызов функций libximc из Matlab-программы производится следующим образом: определите путь к библиотекам `libximc.dll`, `bindy.dll`, `xiwrapper.dll` и заголовочному файлу `ximc.h`, а в случае 64-битной среды ещё к `libximc_thunk_pcwin64.dll` и `ximc.m`. Однократно используйте функцию Matlab `loadlibrary` чтобы загрузить библиотеку libximc и далее используйте функцию `calllib` чтобы вызвать необходимую функцию libximc. Список имен функций libximc, принимаемые и возвращаемые ими параметры вы можете найти в Руководстве по программированию.

6.2. Описание протокола обмена (v17.5)

6.2. Описание протокола обмена (v17.5)
- Описание протокола
- Исполнение команд
- Обработка ошибок на стороне контроллера
  - Неверные команды или данные
  - Расчёт CRC
  - Сбои передачи
  - Восстановление синхронизации методом таймаута
  - Восстановление синхронизации методом очистительных нулей
- Обработка ошибок на стороне библиотеки
  - Возможные значения ответа библиотеки
  - Процедура синхронизации очистительными нулями
- Группа команд настройки контроллера
  - Команда SFBS
  - Команда GFBS
  - Команда SHOM
  - Команда GHOM
  - Команда SMOV
  - Команда GMOV
  - Команда SENG
  - Команда GENG
  - Команда SENT
  - Команда GENT
  - Команда SPWR
  - Команда GPWR
  - Команда SSEC
  - Команда GSEC
  - Команда SEDS
  - Команда GEDS
  - Команда SPID
  - Команда GPID
  - Команда SSNI
  - Команда GSNI
  - Команда SSNO
  - Команда GSNO
  - Команда SEIO
  - Команда GEIO
  - Команда SBRK
  - Команда GBRK
  - Команда SCTL
  - Команда GCTL
  - Команда SJOY
  - Команда GJOY
  - Команда SCTP
  - Команда GCTP
  - Команда SURT
  - Команда GURT
  - Команда SCAL
  - Команда GCAL
  - Команда SNMF
  - Команда GNMF
  - Команда SNVM
  - Команда GNVM
- Группа команд управления движением
  - Команда STOP
  - Команда ASIA
  - Команда PWOF
  - Команда MOVE
  - Команда MOVR
  - Команда HOME
  - Команда LEFT
  - Команда RIGT
  - Команда LOFT
  - Команда SSTP
- Группа команд установки текущей позиции
  - Команда GPOS
  - Команда SPOS

Команда ZERO
Группа команд сохранения и загрузки настроек
Команда SAVE
Команда READ
Команда SARS
Команда RERS
Команда EESV
Команда EERD
Группа команд получения статуса контроллера
Команда GETS
Команда STMS
Команда GETM
Команда GETC
Команда GETI
Команда GSER
Группа команд для работы с прошивкой контроллера
Команда GFWV
Команда UPDF
Группа сервисных команд
Команда SSER
Команда RDAN
Команда DBGR
Команда DBGW
Группа команд работы с EEPROM подвижки
Команда SNME
Команда GNME
Команда SSTI
Команда GSTI
Команда SSTS
Команда GSTS
Команда SMTI
Команда GMTI
Команда SMTS
Команда GMTS
Команда SENI
Команда GENI
Команда SENS
Команда GENS
Команда SHSI
Команда GHSI
Команда SHSS
Команда GHSS
Команда SGRI
Команда GGRI
Команда SGRS
Команда GGRS
Команда SACC
Команда GACC
Команды загрузчика
Команда GBLV
Команда IRND
Команда GUID
Команда CHMT
Процедура записи прошивки
Отрицательные ответы контроллера
Отрицательный ответ ERRC
Отрицательный ответ ERRD
Отрицательный ответ ERRV

Описание протокола

Управление контроллером с ПК происходит по интерфейсу последовательного порта (COM-порт). На стороне контроллера жёстко установлены следующие параметры COM-порта:
Скорость – 115200 бод;
Длина кадра – 8 бит;
Стоп-биты – 2 бита;
Чётность – нет.
Контроль потока – нет(Xon/Xoff, CTS/RTS не используются).
Таймаут на получение, между байтами одного пакета – 400 мсек.
Порядок следования бит – LittleEndian.
Многобайтовые типы данных передаются младшим байтом вперёд.

Исполнение команд

Базовый принцип протокола - "Запрос-Ответ", причём все обмены данными инициируются ПК, т.е. ПК посылает команды в контроллер, но не наоборот. Каждая команда подразумевает получение ответа от контроллера (кроме редких случаев специальных команд), т.е. нельзя послать несколько команд подряд, без ожидания ответа на них.
Все команды делятся на сервисные, штатные управляющие и штатные информационные.
Команды выполняются сразу после их поступления в контроллер. Установленные командой SXXX параметры начинают влиять на текущее движение в течение 1 мс после установки.
Обработка команды не влияет на своевременность выполнения контроллером действий связанных с оперативным управление и контролем двигателя (работа ШИМ, взаимодействие с энкодером и т.п.).
И контроллер и ПК обладают буфером обмена. Принятые команды и данные, в случае их наличия в команде, обрабатываются один раз. То есть, после обработки эти данные удаляются из буфера и обрабатываются уже новые пришедшие байты.
Каждая команда состоит из четырёхбайтной строки, данных (если команда их предусматривает) и двухбайтного кода контроля CRC если команда содержит данные. Данные могут пересылаться как из компьютера, так и контроллером. Команда передаётся на обработку если она распознана и, в случае передачи данных, код CRC верный. После обработки пришедшей без ошибок команды контроллер посылает в компьютер четырехбайтную строку – наименование выполненной команды, затем данные, если формат команды это предусматривает, затем два байта CRC (если есть данные).

Обработка ошибок на стороне контроллера

Неверные команды или данные

Если пришедшая в контроллер команда не может быть интерпретирована, как определенная команда управления, то в компьютер посылается строка "errc", команда игнорируется, в данных текущего состояния контроллера выставляется бит "команда не распознана". Если нераспознанная команда содержала данные, то возможно неверная интерпретация принятых данных как новых команд. Необходима синхронизация.
Если пришедшая в контроллер команда интерпретирована верно, команда предусматривала данные, они пришли, но два байта CRC не соответствует полученным с ней данным, то в данных текущего состояния контроллера устанавливается флаг ошибки CRC пришедших данных, в компьютер посылается строка "errd", текущая команда игнорируется. Синхронизация приёма/передачи с компьютером не нужна.

Расчёт CRC

CRC рассчитывается для передаваемых данных. Четыре байта команды в расчёте не участвуют. Алгоритм CRC на языке Си:

```
unsigned short CRC16(INT8U *pbuf, unsigned short n)
{
  unsigned short crc, i, j, carry_flag, a;
  crc = 0xffff;
  for(i = 0; i < n; i++)
  {
    crc = crc ^ pbuf[i];
    for(j = 0; j < 8; j++)
    {
      a = crc;
      carry_flag = a & 0x0001;
      crc = crc >> 1;
      if ( carry_flag == 1 ) crc = crc ^ 0xa001;
    }
  }
  return crc;
}
```

Функция получает указатель на массив данных pbuf, длину данных в байтах n. Функция возвращает двубайтное слово - код CRC.

Сбои передачи

Наиболее вероятны следующие сбои в канале связи: исчезновение байта при приёме или передаче контроллером, возникновение лишнего байта при приёме или передаче контроллером и изменение принятого или посланного байта.
Сбои происходят при нестандартных условиях и обычно не наблюдаются вообще. Регулярные сбои возможны при некачественном, сломанном кабеле USB или соединительном кабеле между платами. Протокол не разрабатывался для штатного применения в условиях сильно нестабильной связи. В частности в таких условиях редко возможно выполнение не той команды, что была послана.

Исчезновение байта на стороне контроллера
Байт, ожидаемый, но не полученный контроллером, приводит к таймауту компьютера. Посылка команды считается компьютером неуспешной. На этот момент синхронизация передачи данных будет нарушена, но восстановится по таймауту (если таймаут контроллера меньше таймаута компьютера с учётом времени пересылки).

Исчезновение байта на стороне компьютера
Байт, не полученный компьютером, приводит к таймауту компьютера. Синхронизация не нарушена.

Возникновение байта на стороне контроллера
Лишний байт, возникший при приёме контроллером, приводит к получению компьютером одного или нескольких errc либо errd (очень редко сочетания errc и errd). Посылка команды считается неуспешной. В приёмном буфере компьютера может появиться несколько err? ответов контроллера. На этот момент синхронизация нарушена.

Возникновение байта на стороне компьютера
Байт, возникший при приёме компьютером, приводит к неверно принятой команде или неверному коду CRC. Кроме того, в приёмном буфере останется лишний байт. На этот момент синхронизация нарушена.

Изменение байта на стороне контроллера
Байт, изменившийся при приёме контроллером, приводит к получению компьютером одного или нескольких errc либо errd (очень редко сочетания errc и errd). Посылка команды считается неуспешной. В приёмном буфере компьютера может появиться несколько err? ответов контроллера. Обычно синхронизация не нарушается, но редко она может быть нарушена.

Изменение байта на стороне компьютера
Байт, изменившийся при приёме компьютером, приводит к неверно принятой команде или неверному коду CRC. На этот момент синхронизация не нарушена.

Восстановление синхронизации методом таймаута

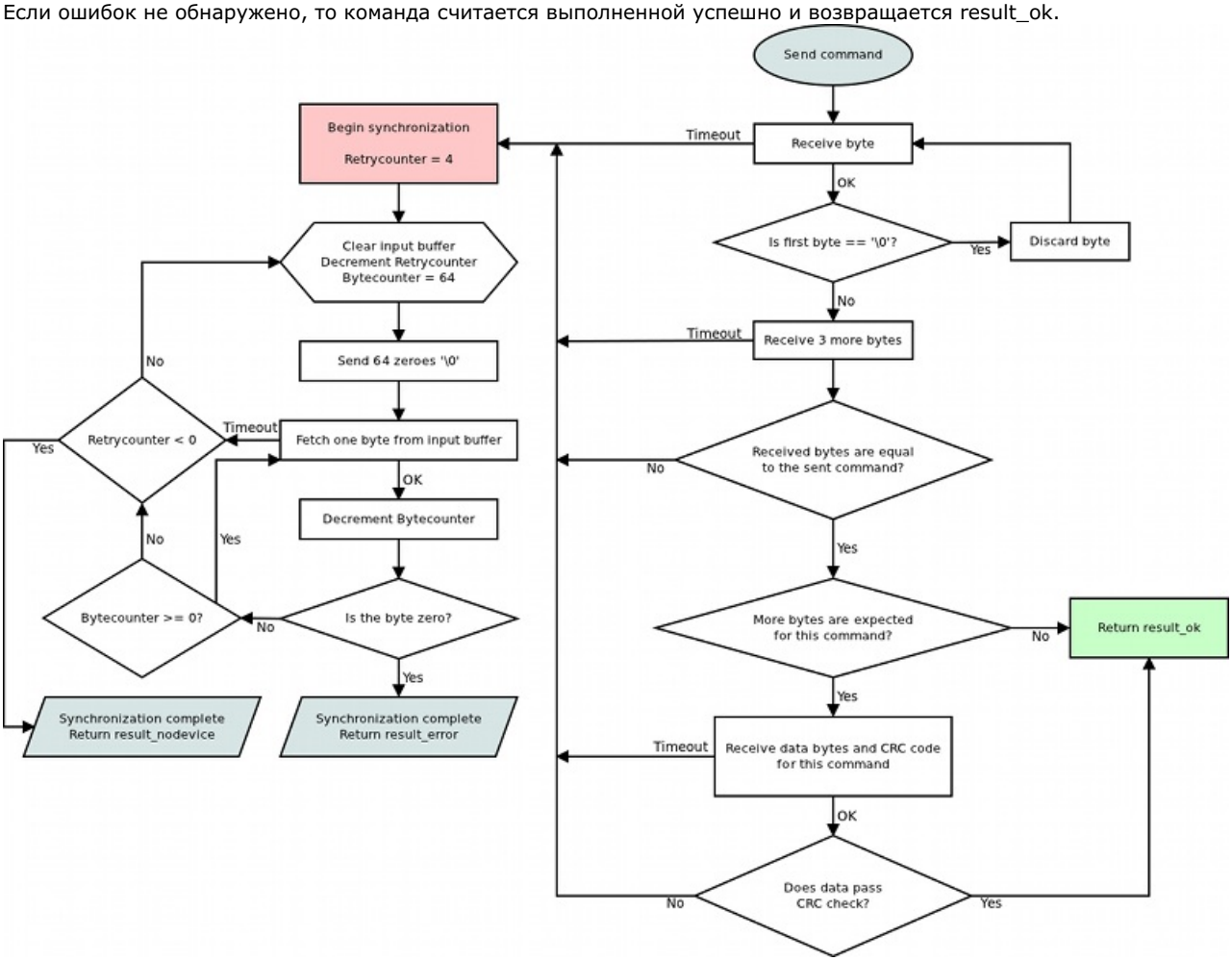
Если при получении пакета, время между получением одного или нескольких байт выходит за рамки таймаута, то полученные данные игнорируются, входной буфер очищается. Время таймаута контроллера должно быть меньше таймаута компьютера с учетом погрешности на время пересылки.

Восстановление синхронизации методом очистительных нулей

Ни одна команда не начинается нулём ('\0'). Поэтому возможен такой метод синхронизации: контроллер на каждый полученный первый байт команды равный нулю отвечает нулём, а компьютер игнорирует первые байт ответа если он равен нулю и переходит к рассмотрению следующего. Тогда в случая когда синхронизация нарушена на стороне компьютера или контроллера, но еще не прошло время таймаута контроллера, возможен следующий алгоритм:
Если компьютером в ответ на переданную команду с данными или без, получен от контроллера ответ не на ту команду, errc или errd, то с компьютера в контроллер средствами библиотеки посылается от 4 до 250 нулей (ограничение в 250 байт связано с длиной приёмного буфера и протоколом передачи данных по I2C, а передача менее 4 нулей часто не приведёт к восстановлению синхронизации). При этом происходит постоянное считывание приходящих байт от контроллера до появления первого нуля. После этого и считывание и посылка прекращаются.
Принятый нуль обычно не является частью предыдущей передачи, так как в моменты ошибок контроллер получает ответы errc/errd. В редких случаях (особое изменение байта на стороне контроллера) возможен приём лишнего мусора. Если мусор содержит нуль, то синхронизация пока не восстановилась, но снова пойдут сбои, которые повторно вызовут очистительные нули и синхронизация восстановится с некоторой попытки. Таким образом, приход первого нуля обычно означает, что приёмный буфер контроллера чист и уже не заполнится, пока не придёт первая значимая команда. Сразу после прихода первого нуля от контроллера компьютер готов передавать следующую команду. Остальные нули, находящиеся в пересылке, будут проигнорированы, так как придут до ответа контроллера.
Синхронизация завершена.

Обработка ошибок на стороне библиотеки

Практически каждая функция библиотеки возвращает статус выполнения типа `result_t`.
После посылки запроса контроллеру библиотека проверяет первые приходящие байты пока не встретит первое ненулевое значение. Все нулевые байты игнорируются. Остальные приходящие байты считаются значимыми. Библиотека ожидает первые 4 байта ответа. Далее она сравнивает их с кодом запроса и, при необходимости, ожидает остальные байты пакета данных. Если полученные 4 байта не соответствуют запросу, то запускается процедура синхронизации очистительными нулями, команда выполнена неуспешно. Если полученные первые 4 байта совпадают с кодом запроса и в ответе есть еще данные, то после их получения проверяется CRC код. Если код неверный, то запускается синхронизация очистительными нулями, выполнение команды считается неуспешным.
Если сработал таймаут при ожидании ответа контроллера, то запускается процедура синхронизации очистительными нулями, выполнение команды считается неуспешным.
Если ошибок не обнаружено, то команда считается выполненной успешно и возвращается result_ok.



Возможные значения ответа библиотеки

- `result_ok`. Ошибок нет.
- `result_error`. Общая ошибка. Может быть связана с аппаратными проблемами, отсутствием данных в буфере порта, превышением таймаутов. Также может означать сбой синхронизации, который был устранён. Такой сбой мог быть вызван помехами на линии связи с контроллером. Еще одной причиной может быть несоответствие протоколов в прошивке и в контроллере.
- `result_nodevice`. Невозможность открытия устройства, потеря связи с ним в процессе передачи данных, неудачная синхронизация. Требуется повторное открытие устройства или вмешательство пользователя.

Если функция возвращает ошибку, любые переданные в неё структуры для записи считаются неопределёнными.
Возврат кода ошибки может сопровождаться записью подробного сообщения в системный лог на unix или в stderr на windows.

Процедура синхронизации очистительными нулями

Восстановление синхронизации осуществляется посылкой нулевых байтов и считывания принимаемых байт до появления первого нулевого значения ('\0'). Опционально можно в конце синхронизации очистить буфер порта. Посылается изначально 64 нулевых байта. Если от контроллера не пришло ни одного нулевого байта за время таймаута, то 64 байта посылаются еще 3 раза. После 4 посылки и неполучения нулевого байта устройство считается потерянным и библиотека должна вернуть код ошибки result_nodevice. В случае удачной синхронизации возвращаемый код ошибки result_error.

Группа команд настройки контроллера

Функции для чтения/записи большинства настроек контроллера.

Команда SFBS

```
result_t set_feedback_settings (device_t id, const feedback_settings_t* feedback_settings)
```

Код команды (CMD): "sfbs" Или 0x73626673.

Запрос: (18 байт)

INT32U	CMD	Команда
INT16U	IPS	Количество отсчётов энкодера на оборот вала. Диапазон: 1..65535. Поле устарело, рекомендуется записывать 0 в IPS и использовать расширенное поле CountsPerTurn. Может потребоваться обновление микропрограммы контроллера до последней версии.
INT8U	FeedbackType	Тип обратной связи
		0x01 - FEEDBACK_ENCODER (Обратная связь с помощью энкодера.)
		0x03 - FEEDBACK_ENCODERHALL (Обратная связь с помощью датчика Холла.)
		0x04 - FEEDBACK_EMF (Обратная связь по ЭДС.)
		0x05 - FEEDBACK_NONE (Обратная связь отсутствует.)
INT8U	FeedbackFlags	Флаги
		0x01 - FEEDBACK_ENC_REVERSE (Обратный счет у энкодера.)
		0x02 - FEEDBACK_HALL_REVERSE (Обратный счёт позиции по датчикам Холла.)
		0xC0 - FEEDBACK_ENC_TYPE_BITS (Биты, отвечающие за тип энкодера.)
		0x00 - FEEDBACK_ENC_TYPE_AUTO (Определять тип энкодера автоматически.)
		0x40 - FEEDBACK_ENC_TYPE_SINGLE_ENDED (Недифференциальный энкодер.)
		0x80 - FEEDBACK_ENC_TYPE_DIFFERENTIAL (Дифференциальный энкодер.)
INT16U	HallSPR	Количество отсчётов датчиков Холла на оборот.
INT8S	HallShift	Фазовый сдвиг между выходным сигналом на обмотках BLDC двигателя и входным сигналом на датчиках Холла(0 - при активном только датчике холла A подается положительный потенциал на обмотку A и отрицательный потенциал на обмотку B).
INT8U	Reserved [5]	Зарезервировано (5 байт)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись настроек обратной связи.

Команда GFBS

```
result_t get_feedback_settings (device_t id, feedback_settings_t* feedback_settings)
```

Код команды (CMD): "gfbs" Или 0x73626667.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (18 байт)

INT32U	CMD	Команда (возврат)
INT16U	IPS	Количество отсчётов энкодера на оборот вала. Диапазон: 1..65535. Поле устарело, рекомендуется записывать 0 в IPS и использовать расширенное поле CountsPerTurn. Может потребоваться обновление микропрограммы контроллера до последней версии.
INT8U	FeedbackType	Тип обратной связи
		0x01 - FEEDBACK_ENCODER (Обратная связь с помощью энкодера.)
		0x03 - FEEDBACK_ENCODERHALL (Обратная связь с помощью датчика Холла.)
		0x04 - FEEDBACK_EMF (Обратная связь по ЭДС.)
		0x05 - FEEDBACK_NONE (Обратная связь отсутствует.)
INT8U	FeedbackFlags	Флаги
		0x01 - FEEDBACK_ENC_REVERSE (Обратный счет у энкодера.)
		0x02 - FEEDBACK_HALL_REVERSE (Обратный счёт позиции по датчикам Холла.)
		0xC0 - FEEDBACK_ENC_TYPE_BITS (Биты, отвечающие за тип энкодера.)
		0x00 - FEEDBACK_ENC_TYPE_AUTO (Определять тип энкодера автоматически.)
		0x40 - FEEDBACK_ENC_TYPE_SINGLE_ENDED (Недифференциальный энкодер.)
		0x80 - FEEDBACK_ENC_TYPE_DIFFERENTIAL (Дифференциальный энкодер.)
INT16U	HallSPR	Количество отсчётов датчиков Холла на оборот.
INT8S	HallShift	Фазовый сдвиг между выходным сигналом на обмотках BLDC двигателя и входным сигналом на датчиках Холла(0 - при активном только датчике холла A подается положительный потенциал на обмотку A и отрицательный потенциал на обмотку B).
INT8U	Reserved [5]	Зарезервировано (5 байт)
INT16U	CRC	Контрольная сумма

Описание:
Чтение настроек обратной связи

Команда SHOM

```
result_t set_home_settings (device_t id, const home_settings_t* home_settings)
```

Код команды (CMD): "shom" Или 0x6D6F6873.

Запрос: (33 байта)

INT32U	CMD	Команда
INT32U	FastHome	Скорость первого движения. Диапазон: 0..100000
INT8U	uFastHome	Дробная часть скорости первого движения в микрошагах(используется только с шаговым двигателем).
INT32U	SlowHome	Скорость второго движения. Диапазон: 0..100000.
INT8U	uSlowHome	Дробная часть скорости второго движения в микрошагах(используется только с шаговым двигателем).
INT32S	HomeDelta	Расстояние отхода от точки останова.
INT16S	uHomeDelta	Дробная часть расстояния отхода от точки останова в микрошагах(используется только с шаговым двигателем). Диапазон: -255..255.
INT16U	HomeFlags	Набор флагов, определяющие такие параметры, как направление и условия останова.
		0x001 - HOME_DIR_FIRST (Определяет направление первоначального движения мотора после поступления команды HOME. Если флаг установлен - вправо; иначе - влево.)
		0x002 - HOME_DIR_SECOND (Определяет направление второго движения мотора. Если флаг установлен - вправо; иначе - влево.)
		0x004 - HOME_MV_SEC_EN (Если флаг установлен, реализуется второй этап доводки в домашнюю позицию; иначе - этап пропускается.)
		0x008 - HOME_HALF_MV (Если флаг установлен, в начале второго движения первые пол оборота сигналы завершения движения игнорируются.)
		0x030 - HOME_STOP_FIRST_BITS (Биты, отвечающие за выбор сигнала завершения первого движения.)
		0x010 - HOME_STOP_FIRST_REV (Первое движение завершается по сигналу с Revolution sensor.)
		0x020 - HOME_STOP_FIRST_SYN (Первое движение завершается по сигналу со входа синхронизации.)
		0x030 - HOME_STOP_FIRST_LIM (Первое движение завершается по сигналу с концевика.)
		0x0C0 - HOME_STOP_SECOND_BITS (Биты, отвечающие за выбор сигнала завершения второго движения.)
		0x040 - HOME_STOP_SECOND_REV (Второе движение завершается по сигналу с Revolution sensor.)
		0x080 - HOME_STOP_SECOND_SYN (Второе движение завершается по сигналу со входа синхронизации.)
		0x0C0 - HOME_STOP_SECOND_LIM (Второе движение завершается по сигналу с концевика.)
		0x100 - HOME_USE_FAST (Если флаг установлен, используется быстрый поиск домашней позиции; иначе - традиционный.)
INT8U	Reserved [9]	Зарезервировано (9 байт)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Команда записи настроек для подхода в home position. Эта функция записывает структуру настроек, использующихся для калибровки позиции, в память контроллера.

Команда GHOM

```
result_t get_home_settings (device_t id, home_settings_t* home_settings)
```

Код команды (CMD): "ghom" Или 0x6D6F6867.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (33 байта)

INT32U	CMD	Команда (возврат)
INT32U	FastHome	Скорость первого движения. Диапазон: 0..100000
INT8U	uFastHome	Дробная часть скорости первого движения в микрошагах(используется только с шаговым двигателем).
INT32U	SlowHome	Скорость второго движения. Диапазон: 0..100000.
INT8U	uSlowHome	Дробная часть скорости второго движения в микрошагах(используется только с шаговым двигателем).
INT32S	HomeDelta	Расстояние отхода от точки останова.
INT16S	uHomeDelta	Дробная часть расстояния отхода от точки останова в микрошагах(используется только с шаговым двигателем). Диапазон: -255..255.
INT16U	HomeFlags	Набор флагов, определяющие такие параметры, как направление и условия останова.
		0x001 - HOME_DIR_FIRST (Определяет направление первоначального движения мотора после поступления команды HOME. Если флаг установлен - вправо; иначе - влево.)
		0x002 - HOME_DIR_SECOND (Определяет направление второго движения мотора. Если флаг установлен - вправо; иначе - влево.)
		0x004 - HOME_MV_SEC_EN (Если флаг установлен, реализуется второй этап доводки в домашнюю позицию; иначе - этап пропускается.)
		0x008 - HOME_HALF_MV (Если флаг установлен, в начале второго движения первые пол оборота сигналы завершения движения игнорируются.)
		0x030 - HOME_STOP_FIRST_BITS (Биты, отвечающие за выбор сигнала завершения первого движения.)
		0x010 - HOME_STOP_FIRST_REV (Первое движение завершается по сигналу с Revolution sensor.)
		0x020 - HOME_STOP_FIRST_SYN (Первое движение завершается по сигналу со входа синхронизации.)
		0x030 - HOME_STOP_FIRST_LIM (Первое движение завершается по сигналу с концевика.)
		0x0C0 - HOME_STOP_SECOND_BITS (Биты, отвечающие за выбор сигнала завершения второго движения.)
		0x040 - HOME_STOP_SECOND_REV (Второе движение завершается по сигналу с Revolution sensor.)
		0x080 - HOME_STOP_SECOND_SYN (Второе движение завершается по сигналу со входа синхронизации.)
		0x0C0 - HOME_STOP_SECOND_LIM (Второе движение завершается по сигналу с концевика.)
		0x100 - HOME_USE_FAST (Если флаг установлен, используется быстрый поиск домашней позиции; иначе - традиционный.)
INT8U	Reserved [9]	Зарезервировано (9 байт)
INT16U	CRC	Контрольная сумма

Описание:
Команда чтения настроек для подхода в home position. Эта функция заполняет структуру настроек, использующихся для калибровки позиции, в память контроллера.

Команда SMOV

```
result_t set_move_settings (device_t id, const move_settings_t* move_settings)
```

Код команды (CMD): "smov" Или 0x766F6D73.

Запрос: (30 байт)

INT32U	CMD	Команда
INT32U	Speed	Заданная скорость (для ШД: шагов/с, для DC: rpm). Диапазон: 0..100000.
INT8U	uSpeed	Заданная скорость в единицах деления микрошага в секунду. Используется только с шаговым мотором.
INT16U	Accel	Ускорение, заданное в шагах в секунду^2(ШД) или в оборотах в минуту за секунду(DC). Диапазон: 1..65535.
INT16U	Decel	Торможение, заданное в шагах в секунду^2(ШД) или в оборотах в минуту за секунду(DC). Диапазон: 1..65535.
INT32U	AntiplaySpeed	Скорость в режиме антилюфта, заданная в целых шагах/с(ШД) или в оборотах/с(DC). Диапазон: 0..100000.
INT8U	uAntiplaySpeed	Скорость в режиме антилюфта, выраженная в 1/256 микрошагах в секунду. Используется только с шаговым мотором.
INT8U	Reserved [10]	Зарезервировано (10 байт)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Команда записи настроек перемещения (скорость, ускорение, threshold и скорость в режиме антилюфта).

Команда GMOV

```
result_t get_move_settings (device_t id, move_settings_t* move_settings)
```

Код команды (CMD): "gmov" Или 0x766F6D67.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (30 байт)

INT32U	CMD	Команда (возврат)
INT32U	Speed	Заданная скорость (для ШД: шагов/с, для DC: rpm). Диапазон: 0..100000.
INT8U	uSpeed	Заданная скорость в единицах деления микрошага в секунду. Используется только с шаговым мотором.
INT16U	Accel	Ускорение, заданное в шагах в секунду^2(ШД) или в оборотах в минуту за секунду(DC). Диапазон: 1..65535.
INT16U	Decel	Торможение, заданное в шагах в секунду^2(ШД) или в оборотах в минуту за секунду(DC). Диапазон: 1..65535.
INT32U	AntiplaySpeed	Скорость в режиме антилюфта, заданная в целых шагах/с(ШД) или в оборотах/с(DC). Диапазон: 0..100000.
INT8U	uAntiplaySpeed	Скорость в режиме антилюфта, выраженная в 1/256 микрошагах в секунду. Используется только с шаговым мотором.
INT8U	Reserved [10]	Зарезервировано (10 байт)
INT16U	CRC	Контрольная сумма

Описание:
Команда чтения настроек перемещения (скорость, ускорение, threshold и скорость в режиме антилюфта).

Команда SENG

```
result_t set_engine_settings (device_t id, const engine_settings_t* engine_settings)
```

Код команды (CMD): "seng" Или 0x676E6573.

Запрос: (34 байта)

INT32U	CMD	Команда
INT16U	NomVoltage	Номинальное напряжение мотора в десятках мВ. Контроллер будет сохранять напряжение на моторе не выше номинального, если установлен флаг ENGINE_LIMIT_VOLT (используется только с DC двигателем).
INT16U	NomCurrent	Номинальный ток через мотор. Ток стабилизируется для шаговых и может быть ограничен для DC. Диапазон: 15..8000
INT32U	NomSpeed	Номинальная скорость (в целых шагах/с или rpm для DC и шагового двигателя в режиме ведущего энкодера). Контроллер будет сохранять скорость мотора не выше номинальной, если установлен флаг ENGINE_LIMIT_RPM. Диапазон: 1..100000.
INT8U	uNomSpeed	Микрошаговая часть номинальной скорости мотора (используется только с шаговым двигателем).
INT16U	EngineFlags	Флаги, управляющие работой мотора.
		0x01 - ENGINE_REVERSE (Флаг реверса. Связывает направление вращения мотора с направлением счета текущей позиции. При сброшенном флаге (по умолчанию) прикладываемое к мотору положительное напряжение увеличивает счетчик позиции. И наоборот, при установленном флаге счетчик позиции увеличивается, когда к мотору приложено отрицательное напряжение. Измените состояние флага, если положительное вращение мотора уменьшает счетчик позиции.)
		0x02 - ENGINE_CURRENT_AS_RMS (Флаг интерпретации значения тока. Если флаг установлен, то задаваемое значение тока интерпретируется как среднеквадратичное значение тока, если флаг снят, то задаваемое значение тока интерпретируется как максимальная амплитуда тока.)
		0x04 - ENGINE_MAX_SPEED (Флаг максимальной скорости. Если флаг установлен, движение происходит на максимальной скорости.)
		0x08 - ENGINE_ANTIPLAY (Компенсация люфта. Если флаг установлен, позиционер будет подходить к заданной точке всегда с одной стороны. Например, при подходе слева никаких дополнительных действий не совершается, а при подходе справа позиционер проходит целевую позицию на заданное расстояния и возвращается к ней опять же справа.)
		0x10 - ENGINE_ACCEL_ON (Ускорение. Если флаг установлен, движение происходит с ускорением.)
		0x20 - ENGINE_LIMIT_VOLT (Номинальное напряжение мотора. Если флаг установлен, напряжение на моторе ограничивается заданным номинальным значением(используется только с DC двигателем).)
		0x40 - ENGINE_LIMIT_CURR (Номинальный ток мотора. Если флаг установлен, ток через мотор ограничивается заданным номинальным значением(используется только с DC двигателем).)
		0x80 - ENGINE_LIMIT_RPM (Номинальная частота вращения мотора. Если флаг установлен, частота вращения ограничивается заданным номинальным значением.)
INT16S	Antiplay	Количество шагов двигателя или импульсов энкодера, на которое позиционер будет отъезжать от заданной позиции для подхода к ней с одной и той же стороны. Используется, если установлен флаг ENGINE_ANTIPLAY.
INT8U	MicrostepMode	Настройки микрошагового режима(используется только с шаговым двигателем).
		0x01 - MICROSTEP_MODE_FULL (Полношаговый режим.)
		0x02 - MICROSTEP_MODE_FRAC_2 (Деление шага 1/2.)
		0x03 - MICROSTEP_MODE_FRAC_4 (Деление шага 1/4.)
		0x04 - MICROSTEP_MODE_FRAC_8 (Деление шага 1/8.)
		0x05 - MICROSTEP_MODE_FRAC_16 (Деление шага 1/16.)
		0x06 - MICROSTEP_MODE_FRAC_32 (Деление шага 1/32.)
		0x07 - MICROSTEP_MODE_FRAC_64 (Деление шага 1/64.)
		0x08 - MICROSTEP_MODE_FRAC_128 (Деление шага 1/128.)
		0x09 - MICROSTEP_MODE_FRAC_256 (Деление шага 1/256.)
INT16U	StepsPerRev	Количество полных шагов на оборот(используется только с шаговым двигателем). Диапазон: 1..65535.
INT8U	Reserved [12]	Зарезервировано (12 байт)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись настроек мотора. Настройки определяют номинальные значения напряжения, тока, скорости мотора, характер движения и тип мотора. Пожалуйста, загружайте новые настройки когда вы меняете мотор, энкодер или позиционер. Помните, что неправильные настройки мотора могут повредить оборудование.

Команда GENG

```
result_t get_engine_settings (device_t id, engine_settings_t* engine_settings)
```

Код команды (CMD): "geng" Или 0x676E6567.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (34 байта)

INT32U	CMD	Команда (возврат)
INT16U	NomVoltage	Номинальное напряжение мотора в десятках мВ. Контроллер будет сохранять напряжение на моторе не выше номинального, если установлен флаг ENGINE_LIMIT_VOLT (используется только с DC двигателем).
INT16U	NomCurrent	Номинальный ток через мотор. Ток стабилизируется для шаговых и может быть ограничен для DC. Диапазон: 15..8000
INT32U	NomSpeed	Номинальная скорость (в целых шагах/с или rpm для DC и шагового двигателя в режиме ведущего энкодера). Контроллер будет сохранять скорость мотора не выше номинальной, если установлен флаг ENGINE_LIMIT_RPM. Диапазон: 1..100000.
INT8U	uNomSpeed	Микрошаговая часть номинальной скорости мотора (используется только с шаговым двигателем).
INT16U	EngineFlags	Флаги, управляющие работой мотора.
		0x01 - ENGINE_REVERSE (Флаг реверса. Связывает направление вращения мотора с направлением счета текущей позиции. При сброшенном флаге (по умолчанию) прикладываемое к мотору положительное напряжение увеличивает счетчик позиции. И наоборот, при установленном флаге счетчик позиции увеличивается, когда к мотору приложено отрицательное напряжение. Измените состояние флага, если положительное вращение мотора уменьшает счетчик позиции.)
		0x02 - ENGINE_CURRENT_AS_RMS (Флаг интерпретации значения тока. Если флаг установлен, то задаваемое значение тока интерпретируется как среднеквадратичное значение тока, если флаг снят, то задаваемое значение тока интерпретируется как максимальная амплитуда тока.)
		0x04 - ENGINE_MAX_SPEED (Флаг максимальной скорости. Если флаг установлен, движение происходит на максимальной скорости.)
		0x08 - ENGINE_ANTIPLAY (Компенсация люфта. Если флаг установлен, позиционер будет подходить к заданной точке всегда с одной стороны. Например, при подходе слева никаких дополнительных действий не совершается, а при подходе справа позиционер проходит целевую позицию на заданное расстояния и возвращается к ней опять же справа.)
		0x10 - ENGINE_ACCEL_ON (Ускорение. Если флаг установлен, движение происходит с ускорением.)
		0x20 - ENGINE_LIMIT_VOLT (Номинальное напряжение мотора. Если флаг установлен, напряжение на моторе ограничивается заданным номинальным значением(используется только с DC двигателем).)
		0x40 - ENGINE_LIMIT_CURR (Номинальный ток мотора. Если флаг установлен, ток через мотор ограничивается заданным номинальным значением(используется только с DC двигателем).)
		0x80 - ENGINE_LIMIT_RPM (Номинальная частота вращения мотора. Если флаг установлен, частота вращения ограничивается заданным номинальным значением.)
INT16S	Antiplay	Количество шагов двигателя или импульсов энкодера, на которое позиционер будет отъезжать от заданной позиции для подхода к ней с одной и той же стороны. Используется, если установлен флаг ENGINE_ANTIPLAY.
INT8U	MicrostepMode	Настройки микрошагового режима(используется только с шаговым двигателем).
		0x01 - MICROSTEP_MODE_FULL (Полношаговый режим.)
		0x02 - MICROSTEP_MODE_FRAC_2 (Деление шага 1/2.)
		0x03 - MICROSTEP_MODE_FRAC_4 (Деление шага 1/4.)
		0x04 - MICROSTEP_MODE_FRAC_8 (Деление шага 1/8.)
		0x05 - MICROSTEP_MODE_FRAC_16 (Деление шага 1/16.)
		0x06 - MICROSTEP_MODE_FRAC_32 (Деление шага 1/32.)
		0x07 - MICROSTEP_MODE_FRAC_64 (Деление шага 1/64.)
		0x08 - MICROSTEP_MODE_FRAC_128 (Деление шага 1/128.)
		0x09 - MICROSTEP_MODE_FRAC_256 (Деление шага 1/256.)
INT16U	StepsPerRev	Количество полных шагов на оборот(используется только с шаговым двигателем). Диапазон: 1..65535.
INT8U	Reserved [12]	Зарезервировано (12 байт)
INT16U	CRC	Контрольная сумма

Описание:
Чтение настроек мотора. Настройки определяют номинальные значения напряжения, тока, скорости мотора, характер движения и тип мотора. Пожалуйста, загружайте новые настройки когда вы меняете мотор, энкодер или позиционер. Помните, что неправильные настройки мотора могут повредить оборудование.

Команда SENT

```
result_t set_entype_settings (device_t id, const entype_settings_t* entype_settings)
```

Код команды (CMD): "sent" Или 0x746E6573.

Запрос: (14 байт)

INT32U	CMD	Команда
INT8U	EngineType	Тип мотора
		0x00 - ENGINE_TYPE_NONE (Это значение не нужно использовать.)
		0x01 - ENGINE_TYPE_DC (Мотор постоянного тока.)
		0x02 - ENGINE_TYPE_2DC (Два мотора постоянного тока, что приводит к эмуляции двух контроллеров.)
		0x03 - ENGINE_TYPE_STEP (Шаговый мотор.)
		0x04 - ENGINE_TYPE_TEST (Скважность в обмотках фиксирована. Используется только производителем.)
		0x05 - ENGINE_TYPE_BRUSHLESS (Безщеточный мотор.)
INT8U	DriverType	Тип силового драйвера
		0x01 - DRIVER_TYPE_DISCRETE_FET (Силовой драйвер на дискретных мосфет-ключах. Используется по умолчанию.)
		0x02 - DRIVER_TYPE_INTEGRATE (Силовой драйвер с использованием ключей, интегрированных в микросхему.)
		0x03 - DRIVER_TYPE_EXTERNAL (Внешний силовой драйвер.)
INT8U	Reserved [6]	Зарезервировано (6 байт)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись информации о типе мотора и типе силового драйвера.

Команда GENT

```
result_t get_entype_settings (device_t id, entype_settings_t* entype_settings)
```

Код команды (CMD): "gent" Или 0x746E6567.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (14 байт)

INT32U	CMD	Команда (возврат)
INT8U	EngineType	Тип мотора
		0x00 - ENGINE_TYPE_NONE (Это значение не нужно использовать.)
		0x01 - ENGINE_TYPE_DC (Мотор постоянного тока.)
		0x02 - ENGINE_TYPE_2DC (Два мотора постоянного тока, что приводит к эмуляции двух контроллеров.)
		0x03 - ENGINE_TYPE_STEP (Шаговый мотор.)
		0x04 - ENGINE_TYPE_TEST (Скважность в обмотках фиксирована. Используется только производителем.)
		0x05 - ENGINE_TYPE_BRUSHLESS (Безщеточный мотор.)
INT8U	DriverType	Тип силового драйвера
		0x01 - DRIVER_TYPE_DISCRETE_FET (Силовой драйвер на дискретных мосфет-ключах. Используется по умолчанию.)
		0x02 - DRIVER_TYPE_INTEGRATE (Силовой драйвер с использованием ключей, интегрированных в микросхему.)
		0x03 - DRIVER_TYPE_EXTERNAL (Внешний силовой драйвер.)
INT8U	Reserved [6]	Зарезервировано (6 байт)
INT16U	CRC	Контрольная сумма

Описание:
Возвращает информацию о типе мотора и силового драйвера.

Команда SPWR

```
result_t set_power_settings (device_t id, const power_settings_t* power_settings)
```

Код команды (CMD): "spwr" Или 0x72777073.

Запрос: (20 байт)

INT32U	CMD	Команда
INT8U	HoldCurrent	Ток мотора в режиме удержания, в процентах от номинального. Диапазон: 0..100.
INT16U	CurrReductDelay	Время в мс от перехода в состояние STOP до уменьшения тока.
INT16U	PowerOffDelay	Время в с от перехода в состояние STOP до отключения питания мотора.
INT16U	CurrentSetTime	Время в мс, требуемое для набора номинального тока от 0% до 100%.
INT8U	PowerFlags	Флаги параметров управления питанием.
		0x01 - POWER_REDUCT_ENABLED (Если флаг установлен, уменьшить ток по прошествии CurrReductDelay. Иначе - не уменьшать.)
		0x02 - POWER_OFF_ENABLED (Если флаг установлен, снять напряжение с обмоток по прошествии PowerOffDelay. Иначе - не снимать.)
		0x04 - POWER_SMOOTH_CURRENT (Если установлен, то запитывание обмоток, снятие питания или снижение/повышение тока происходят плавно со скоростью CurrentSetTime, а только потом выполняется та задача, которая вызвала это плавное изменение.)
INT8U	Reserved [6]	Зарезервировано (6 байт)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Команда записи параметров питания мотора. Используется только с шаговым двигателем.

Команда GPWR

```
result_t get_power_settings (device_t id, power_settings_t* power_settings)
```

Код команды (CMD): "gpwr" Или 0x72777067.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (20 байт)

INT32U	CMD	Команда (возврат)
INT8U	HoldCurrent	Ток мотора в режиме удержания, в процентах от номинального. Диапазон: 0..100.
INT16U	CurrReductDelay	Время в мс от перехода в состояние STOP до уменьшения тока.
INT16U	PowerOffDelay	Время в с от перехода в состояние STOP до отключения питания мотора.
INT16U	CurrentSetTime	Время в мс, требуемое для набора номинального тока от 0% до 100%.
INT8U	PowerFlags	Флаги параметров управления питанием.
		0x01 - POWER_REDUCT_ENABLED (Если флаг установлен, уменьшить ток по прошествии CurrReductDelay. Иначе - не уменьшать.)
		0x02 - POWER_OFF_ENABLED (Если флаг установлен, снять напряжение с обмоток по прошествии PowerOffDelay. Иначе - не снимать.)
		0x04 - POWER_SMOOTH_CURRENT (Если установлен, то запитывание обмоток, снятие питания или снижение/повышение тока происходят плавно со скоростью CurrentSetTime, а только потом выполняется та задача, которая вызвала это плавное изменение.)
INT8U	Reserved [6]	Зарезервировано (6 байт)
INT16U	CRC	Контрольная сумма

Описание:
Команда чтения параметров питания мотора. Используется только с шаговым двигателем. Используется только с шаговым двигателем.

Команда SSEC

```
result_t set_secure_settings (device_t id, const secure_settings_t* secure_settings)
```

Код команды (CMD): "ssec" Или 0x63657373.

Запрос: (28 байт)

INT32U	CMD	Команда
INT16U	LowUpwrOff	Нижний порог напряжения на силовой части для выключения, десятки мВ.
INT16U	CriticalIpwr	Максимальный ток силовой части, вызывающий состояние ALARM, в мА.
INT16U	CriticalUpwr	Максимальное напряжение на силовой части, вызывающее состояние ALARM, десятки мВ.
INT16U	CriticalT	Максимальная температура контроллера, вызывающая состояние ALARM, в десятых долях градуса Цельсия
INT16U	CriticalIusb	Максимальный ток USB, вызывающий состояние ALARM, в мА.
INT16U	CriticalUusb	Максимальное напряжение на USB, вызывающее состояние ALARM, десятки мВ.
INT16U	MinimumUusb	Минимальное напряжение на USB, вызывающее состояние ALARM, десятки мВ.
INT8U	Flags	Флаги критических параметров.
		0x01 - ALARM_ON_DRIVER_OVERHEATING (Если флаг установлен, то войти в состояние Alarm при получении сигнала подступающего перегрева с драйвера. Иначе - игнорировать подступающий перегрев с драйвера.)
		0x02 - LOW_UPWR_PROTECTION (Если установлен, то выключать силовую часть при напряжении меньшем LowUpwrOff.)
		0x04 - H_BRIDGE_ALERT (Если установлен, то выключать силовую часть при сигнале неполадки в одном из транзисторных мостов)
		0x08 - ALARM_ON_BORDERS_SWAP_MISSET (Если флаг установлен, то войти в состояние Alarm при получении сигнала с противоположного концевика)
		0x10 - ALARM_FLAGS_STICKING (Если флаг установлен, то только по команде STOP возможен сброс всех флагов ALARM)
		0x20 - USB_BREAK_RECONNECT (Если флаг установлен, то будет включен блок перезагрузки USB при поломке связи)
INT8U	Reserved [7]	Зарезервировано (7 байт)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Команда записи установок защит.

Команда GSEC

```
result_t get_secure_settings (device_t id, secure_settings_t* secure_settings)
```

Код команды (CMD): "gsec" Или 0x63657367.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (28 байт)

INT32U	CMD	Команда (возврат)
INT16U	LowUpwrOff	Нижний порог напряжения на силовой части для выключения, десятки мВ.
INT16U	CriticalIpwr	Максимальный ток силовой части, вызывающий состояние ALARM, в мА.
INT16U	CriticalUpwr	Максимальное напряжение на силовой части, вызывающее состояние ALARM, десятки мВ.
INT16U	CriticalT	Максимальная температура контроллера, вызывающая состояние ALARM, в десятых долях градуса Цельсия
INT16U	CriticalIusb	Максимальный ток USB, вызывающий состояние ALARM, в мА.
INT16U	CriticalUusb	Максимальное напряжение на USB, вызывающее состояние ALARM, десятки мВ.
INT16U	MinimumUusb	Минимальное напряжение на USB, вызывающее состояние ALARM, десятки мВ.
INT8U	Flags	Флаги критических параметров.
		0x01 - ALARM_ON_DRIVER_OVERHEATING (Если флаг установлен, то войти в состояние Alarm при получении сигнала подступающего перегрева с драйвера. Иначе - игнорировать подступающий перегрев с драйвера.)
		0x02 - LOW_UPWR_PROTECTION (Если установлен, то выключать силовую часть при напряжении меньшем LowUpwrOff.)
		0x04 - H_BRIDGE_ALERT (Если установлен, то выключать силовую часть при сигнале неполадки в одном из транзисторных мостов)
		0x08 - ALARM_ON_BORDERS_SWAP_MISSET (Если флаг установлен, то войти в состояние Alarm при получении сигнала с противоположного концевика)
		0x10 - ALARM_FLAGS_STICKING (Если флаг установлен, то только по команде STOP возможен сброс всех флагов ALARM)
		0x20 - USB_BREAK_RECONNECT (Если флаг установлен, то будет включен блок перезагрузки USB при поломке связи)
INT8U	Reserved [7]	Зарезервировано (7 байт)
INT16U	CRC	Контрольная сумма

Описание:
Команда записи установок защит.

Команда SEDS

```
result_t set_edges_settings (device_t id, const edges_settings_t* edges_settings)
```

Код команды (CMD): "seds" Или 0x73646573.

Запрос: (26 байт)

INT32U	CMD	Команда
INT8U	BorderFlags	Флаги, определяющие тип границ и поведение мотора при их достижении.
		0x01 - BORDER_IS_ENCODER (Если флаг установлен, границы определяются предустановленными точками на шкале позиции. Если флаг сброшен, границы определяются концевыми выключателями.)
		0x02 - BORDER_STOP_LEFT (Если флаг установлен, мотор останавливается при достижении левой границы.)
		0x04 - BORDER_STOP_RIGHT (Если флаг установлен, мотор останавливается при достижении правой границы.)
		0x08 - BORDERS_SWAP_MISSET_DETECTION (Если флаг установлен, мотор останавливается при достижении обоих границ. Нужен для предотвращения поломки двигателя при неправильных настройках концевиков)
INT8U	EnderFlags	Флаги, определяющие настройки концевых выключателей.
		0x01 - ENDER_SWAP (Если флаг установлен, первый концевой выключатель находится справа; иначе - слева.)
		0x02 - ENDER_SW1_ACTIVE_LOW (1 - Концевик, подключенный к ножке SW1, считается сработавшим по низкому уровню на контакте.)
		0x04 - ENDER_SW2_ACTIVE_LOW (1 - Концевик, подключенный к ножке SW2, считается сработавшим по низкому уровню на контакте.)
INT32S	LeftBorder	Позиция левой границы, используется если установлен флаг BORDER_IS_ENCODER.
INT16S	uLeftBorder	Позиция левой границы в 1/256 микрошагах(используется только с шаговым двигателем). Диапазон: -255..255.
INT32S	RightBorder	Позиция правой границы, используется если установлен флаг BORDER_IS_ENCODER.
INT16S	uRightBorder	Позиция правой границы в 1/256 микрошагах (используется только с шаговым двигателем). Диапазон: -255..255.
INT8U	Reserved [6]	Зарезервировано (6 байт)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись настроек границ и концевых выключателей.

Команда GEDS

```
result_t get_edges_settings (device_t id, edges_settings_t* edges_settings)
```

Код команды (CMD): "geds" Или 0x73646567.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (26 байт)

INT32U	CMD	Команда (возврат)
INT8U	BorderFlags	Флаги, определяющие тип границ и поведение мотора при их достижении.
		0x01 - BORDER_IS_ENCODER (Если флаг установлен, границы определяются предустановленными точками на шкале позиции. Если флаг сброшен, границы определяются концевыми выключателями.)
		0x02 - BORDER_STOP_LEFT (Если флаг установлен, мотор останавливается при достижении левой границы.)
		0x04 - BORDER_STOP_RIGHT (Если флаг установлен, мотор останавливается при достижении правой границы.)
		0x08 - BORDERS_SWAP_MISSET_DETECTION (Если флаг установлен, мотор останавливается при достижении обоих границ. Нужен для предотвращения поломки двигателя при неправильных настройках концевиков)
INT8U	EnderFlags	Флаги, определяющие настройки концевых выключателей.
		0x01 - ENDER_SWAP (Если флаг установлен, первый концевой выключатель находится справа; иначе - слева.)
		0x02 - ENDER_SW1_ACTIVE_LOW (1 - Концевик, подключенный к ножке SW1, считается сработавшим по низкому уровню на контакте.)
		0x04 - ENDER_SW2_ACTIVE_LOW (1 - Концевик, подключенный к ножке SW2, считается сработавшим по низкому уровню на контакте.)
INT32S	LeftBorder	Позиция левой границы, используется если установлен флаг BORDER_IS_ENCODER.
INT16S	uLeftBorder	Позиция левой границы в 1/256 микрошагах(используется только с шаговым двигателем). Диапазон: -255..255.
INT32S	RightBorder	Позиция правой границы, используется если установлен флаг BORDER_IS_ENCODER.
INT16S	uRightBorder	Позиция правой границы в 1/256 микрошагах (используется только с шаговым двигателем). Диапазон: -255..255.
INT8U	Reserved [6]	Зарезервировано (6 байт)
INT16U	CRC	Контрольная сумма

Описание:
Чтение настроек границ и концевых выключателей.

Команда SPID

```
result_t set_pid_settings (device_t id, const pid_settings_t* pid_settings)
```

Код команды (CMD): "spid" Или 0x64697073.

Запрос: (48 байт)

INT32U	CMD	Команда
INT16U	KpU	Пропорциональный коэффициент ПИД контура по напряжению
INT16U	KiU	Интегральный коэффициент ПИД контура по напряжению
INT16U	KdU	Дифференциальный коэффициент ПИД контура по напряжению
FLT32	Kpf	Пропорциональный коэффициент ПИД контура по позиции для BLDC
FLT32	Kif	Интегральный коэффициент ПИД контура по позиции для BLDC
FLT32	Kdf	Дифференциальный коэффициент ПИД контура по позиции для BLDC
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись ПИД коэффициентов. Эти коэффициенты определяют поведение позиционера. Коэффициенты различны для разных позиционеров. Пожалуйста, загружайте новые настройки, когда вы меняете мотор или позиционер.

Команда GPID

```
result_t get_pid_settings (device_t id, pid_settings_t* pid_settings)
```

Код команды (CMD): "gpid" Или 0x64697067.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (48 байт)

INT32U	CMD	Команда (возврат)
INT16U	KpU	Пропорциональный коэффициент ПИД контура по напряжению
INT16U	KiU	Интегральный коэффициент ПИД контура по напряжению
INT16U	KdU	Дифференциальный коэффициент ПИД контура по напряжению
FLT32	Kpf	Пропорциональный коэффициент ПИД контура по позиции для BLDC
FLT32	Kif	Интегральный коэффициент ПИД контура по позиции для BLDC
FLT32	Kdf	Дифференциальный коэффициент ПИД контура по позиции для BLDC
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Описание:
Чтение ПИД коэффициентов. Эти коэффициенты определяют поведение позиционера. Коэффициенты различны для разных позиционеров.

Команда SSNI

```
result_t set_sync_in_settings (device_t id, const sync_in_settings_t* sync_in_settings)
```

Код команды (CMD): "ssni" Или 0x696E7373.

Запрос: (28 байт)

INT32U	CMD	Команда
INT8U	SyncInFlags	Флаги синхронизации входа
		0x01 - SYNCIN_ENABLED (Включение необходимости импульса синхронизации для начала движения.)
		0x02 - SYNCIN_INVERT (Если установлен - срабатывает по переходу из 1 в 0. Иначе - из 0 в 1.)
		0x04 - SYNCIN_GOTOPOSITION (Если флаг установлен, то двигатель смещается к позиции, установленной в Position и uPosition, иначе двигатель смещается на Position и uPosition)
INT16U	ClutterTime	Минимальная длительность входного импульса синхронизации для защиты от дребезга (мкс).
INT32S	Position	Желаемая позиция или смещение (целая часть)
INT16S	uPosition	Дробная часть позиции или смещения в микрошагах. Используется только с шаговым двигателем. Диапазон: -255..255.
INT32U	Speed	Заданная скорость (для ШД: шагов/с, для DC: rpm). Диапазон: 0..100000.
INT8U	uSpeed	Заданная скорость в микрошагах в секунду. Используется только с шаговым мотором.
INT8U	Reserved [8]	Зарезервировано (8 байт)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись настроек для входного импульса синхронизации. Эта функция записывает структуру с настройками входного импульса синхронизации, определяющими поведение входа синхронизации, в память контроллера.

Команда GSNI

```
result_t get_sync_in_settings (device_t id, sync_in_settings_t* sync_in_settings)
```

Код команды (CMD): "gsni" Или 0x696E7367.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (28 байт)

INT32U	CMD	Команда (возврат)
INT8U	SyncInFlags	Флаги синхронизации входа
		0x01 - SYNCIN_ENABLED (Включение необходимости импульса синхронизации для начала движения.)
		0x02 - SYNCIN_INVERT (Если установлен - срабатывает по переходу из 1 в 0. Иначе - из 0 в 1.)
		0x04 - SYNCIN_GOTOPOSITION (Если флаг установлен, то двигатель смещается к позиции, установленной в Position и uPosition, иначе двигатель смещается на Position и uPosition)
INT16U	ClutterTime	Минимальная длительность входного импульса синхронизации для защиты от дребезга (мкс).
INT32S	Position	Желаемая позиция или смещение (целая часть)
INT16S	uPosition	Дробная часть позиции или смещения в микрошагах. Используется только с шаговым двигателем. Диапазон: -255..255.
INT32U	Speed	Заданная скорость (для ШД: шагов/с, для DC: rpm). Диапазон: 0..100000.
INT8U	uSpeed	Заданная скорость в микрошагах в секунду. Используется только с шаговым мотором.
INT8U	Reserved [8]	Зарезервировано (8 байт)
INT16U	CRC	Контрольная сумма

Описание:
Чтение настроек для входного импульса синхронизации. Эта функция считывает структуру с настройками синхронизации, определяющими поведение входа синхронизации, в память контроллера.

Команда SSNO

```
result_t set_sync_out_settings (device_t id, const sync_out_settings_t* sync_out_settings)
```

Код команды (CMD): "ssno" Или 0x6F6E7373.

Запрос: (16 байт)

INT32U	CMD	Команда
INT8U	SyncOutFlags	Флаги синхронизации выхода
		0x01 - SYNCOUT_ENABLED (Синхронизация выхода работает согласно настройкам, если флаг установлен. В ином случае значение выхода фиксировано и подчиняется SYNCOUT_STATE.)
		0x02 - SYNCOUT_STATE (Когда значение выхода управляется напрямую (см. флаг SYNCOUT_ENABLED), значение на выходе соответствует значению этого флага.)
		0x04 - SYNCOUT_INVERT (Нулевой логический уровень является активным, если флаг установлен, а единичный - если флаг сброшен.)
		0x08 - SYNCOUT_IN_STEPS (Если флаг установлен использовать шаги/импульсы энкодера для выходных импульсов синхронизации вместо миллисекунд.)
		0x10 - SYNCOUT_ONSTART (Генерация синхронизирующего импульса при начале движения.)
		0x20 - SYNCOUT_ONSTOP (Генерация синхронизирующего импульса при остановке.)
		0x40 - SYNCOUT_ONPERIOD (Выдать импульс синхронизации после прохождения SyncOutPeriod отсчётов.)
INT16U	SyncOutPulseSteps	Определяет длительность выходных импульсов в шагах/импульсах энкодера, когда установлен флаг SYNCOUT_IN_STEPS, или в микросекундах если флаг сброшен.
INT16U	SyncOutPeriod	Период генерации импульсов, используется при установленном флаге SYNCOUT_ONPERIOD.
INT32U	Accuracy	Это окрестность вокруг целевой координаты, попадание в которую считается попаданием в целевую позицию и генерируется импульс по остановке.
INT8U	uAccuracy	Это окрестность вокруг целевой координаты в микрошагах (используется только с шаговым двигателем).
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись настроек для выходного импульса синхронизации. Эта функция записывает структуру с настройками выходного импульса синхронизации, определяющими поведение вывода синхронизации, в память контроллера.

Команда GSNO

```
result_t get_sync_out_settings (device_t id, sync_out_settings_t* sync_out_settings)
```

Код команды (CMD): "gsno" Или 0x6F6E7367.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (16 байт)

INT32U	CMD	Команда (возврат)
INT8U	SyncOutFlags	Флаги синхронизации выхода
		0x01 - SYNCOUT_ENABLED (Синхронизация выхода работает согласно настройкам, если флаг установлен. В ином случае значение выхода фиксировано и подчиняется SYNCOUT_STATE.)
		0x02 - SYNCOUT_STATE (Когда значение выхода управляется напрямую (см. флаг SYNCOUT_ENABLED), значение на выходе соответствует значению этого флага.)
		0x04 - SYNCOUT_INVERT (Нулевой логический уровень является активным, если флаг установлен, а единичный - если флаг сброшен.)
		0x08 - SYNCOUT_IN_STEPS (Если флаг установлен использовать шаги/импульсы энкодера для выходных импульсов синхронизации вместо миллисекунд.)
		0x10 - SYNCOUT_ONSTART (Генерация синхронизирующего импульса при начале движения.)
		0x20 - SYNCOUT_ONSTOP (Генерация синхронизирующего импульса при остановке.)
		0x40 - SYNCOUT_ONPERIOD (Выдать импульс синхронизации после прохождения SyncOutPeriod отсчётов.)
INT16U	SyncOutPulseSteps	Определяет длительность выходных импульсов в шагах/импульсах энкодера, когда установлен флаг SYNCOUT_IN_STEPS, или в микросекундах если флаг сброшен.
INT16U	SyncOutPeriod	Период генерации импульсов, используется при установленном флаге SYNCOUT_ONPERIOD.
INT32U	Accuracy	Это окрестность вокруг целевой координаты, попадание в которую считается попаданием в целевую позицию и генерируется импульс по остановке.
INT8U	uAccuracy	Это окрестность вокруг целевой координаты в микрошагах (используется только с шаговым двигателем).
INT16U	CRC	Контрольная сумма

Описание:
Чтение настроек для выходного импульса синхронизации. Эта функция считывает структуру с настройками синхронизации, определяющими поведение выхода синхронизации, в память контроллера.

Команда SEIO

```
result_t set_extio_settings (device_t id, const extio_settings_t* extio_settings)
```

Код команды (CMD): "seio" Или 0x6F696573.

Запрос: (18 байт)

INT32U	CMD	Команда
INT8U	EXTIOSetupFlags	Флаги настройки работы внешнего ввода-вывода
		0x01 - EXTIO_SETUP_OUTPUT (Если флаг установлен, то ножка в состоянии вывода, иначе - ввода.)
		0x02 - EXTIO_SETUP_INVERT (Если флаг установлен, то нули считаются активным состоянием выхода, а спадающие фронты как момент подачи входного сигнала.)
INT8U	EXTIOModeFlags	Флаги настройки режимов внешнего ввода-вывода
		0x0F - EXTIO_SETUP_MODE_IN_BITS (Биты, отвечающие за поведение при переходе сигнала в активное состояние.)
		0x00 - EXTIO_SETUP_MODE_IN_NOP (Ничего не делать.)
		0x01 - EXTIO_SETUP_MODE_IN_STOP (По переднему фронту входного сигнала делается остановка двигателя (эквивалент команды STOP).)
		0x02 - EXTIO_SETUP_MODE_IN_PWOF (Выполняет команду PWOF, обесточивая обмотки двигателя.)
		0x03 - EXTIO_SETUP_MODE_IN_MOVR (Выполняется команда MOVR с последними настройками.)
		0x04 - EXTIO_SETUP_MODE_IN_HOME (Выполняется команда HOME.)
		0x05 - EXTIO_SETUP_MODE_IN_ALARM (Войти в состояние ALARM при переходе сигнала в активное состояние.)
		0xF0 - EXTIO_SETUP_MODE_OUT_BITS (Биты выбора поведения на выходе.)
		0x00 - EXTIO_SETUP_MODE_OUT_OFF (Ножка всегда в неактивном состоянии.)
		0x10 - EXTIO_SETUP_MODE_OUT_ON (Ножка всегда в активном состоянии.)
		0x20 - EXTIO_SETUP_MODE_OUT_MOVING (Ножка находится в активном состоянии при движении.)
		0x30 - EXTIO_SETUP_MODE_OUT_ALARM (Ножка находится в активном состоянии при нахождении в состоянии ALARM.)
		0x40 - EXTIO_SETUP_MODE_OUT_MOTOR_ON (Ножка находится в активном состоянии при подаче питания на обмотки.)
		0x50 - EXTIO_SETUP_MODE_OUT_MOTOR_FOUND (Ножка находится в активном состоянии при обнаружении подключенного двигателя (первой обмотки).)
INT8U	Reserved [10]	Зарезервировано (10 байт)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Команда записи параметров настройки режимов внешнего ввода/вывода. Входные события обрабатываются по фронту. Выходные состояния сигнализируются логическим состоянием. По умолчанию нарастающий фронт считается моментом подачи входного сигнала, а единичное состояние считается активным выходом.

Команда GEIO

```
result_t get_extio_settings (device_t id, extio_settings_t* extio_settings)
```

Код команды (CMD): "geio" Или 0x6F696567.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (18 байт)

INT32U	CMD	Команда (возврат)
INT8U	EXTIOSetupFlags	Флаги настройки работы внешнего ввода-вывода
		0x01 - EXTIO_SETUP_OUTPUT (Если флаг установлен, то ножка в состоянии вывода, иначе - ввода.)
		0x02 - EXTIO_SETUP_INVERT (Если флаг установлен, то нули считаются активным состоянием выхода, а спадающие фронты как момент подачи входного сигнала.)
INT8U	EXTIOModeFlags	Флаги настройки режимов внешнего ввода-вывода
		0x0F - EXTIO_SETUP_MODE_IN_BITS (Биты, отвечающие за поведение при переходе сигнала в активное состояние.)
		0x00 - EXTIO_SETUP_MODE_IN_NOP (Ничего не делать.)
		0x01 - EXTIO_SETUP_MODE_IN_STOP (По переднему фронту входного сигнала делается остановка двигателя (эквивалент команды STOP).)
		0x02 - EXTIO_SETUP_MODE_IN_PWOF (Выполняет команду PWOF, обесточивая обмотки двигателя.)
		0x03 - EXTIO_SETUP_MODE_IN_MOVR (Выполняется команда MOVR с последними настройками.)
		0x04 - EXTIO_SETUP_MODE_IN_HOME (Выполняется команда HOME.)
		0x05 - EXTIO_SETUP_MODE_IN_ALARM (Войти в состояние ALARM при переходе сигнала в активное состояние.)
		0xF0 - EXTIO_SETUP_MODE_OUT_BITS (Биты выбора поведения на выходе.)
		0x00 - EXTIO_SETUP_MODE_OUT_OFF (Ножка всегда в неактивном состоянии.)
		0x10 - EXTIO_SETUP_MODE_OUT_ON (Ножка всегда в активном состоянии.)
		0x20 - EXTIO_SETUP_MODE_OUT_MOVING (Ножка находится в активном состоянии при движении.)
		0x30 - EXTIO_SETUP_MODE_OUT_ALARM (Ножка находится в активном состоянии при нахождении в состоянии ALARM.)
		0x40 - EXTIO_SETUP_MODE_OUT_MOTOR_ON (Ножка находится в активном состоянии при подаче питания на обмотки.)
		0x50 - EXTIO_SETUP_MODE_OUT_MOTOR_FOUND (Ножка находится в активном состоянии при обнаружении подключенного двигателя (первой обмотки).)
INT8U	Reserved [10]	Зарезервировано (10 байт)
INT16U	CRC	Контрольная сумма

Описание:
Команда чтения параметров настройки режимов внешнего ввода/вывода.

Команда SBRK

```
result_t set_brake_settings (device_t id, const brake_settings_t* brake_settings)
```

Код команды (CMD): "sbrk" Или 0x6B726273.

Запрос: (25 байт)

INT32U	CMD	Команда
INT16U	t1	Время в мс между включением питания мотора и отключением тормоза.
INT16U	t2	Время в мс между отключением тормоза и готовностью к движению. Все команды движения начинают выполняться только по истечении этого времени.
INT16U	t3	Время в мс между остановкой мотора и включением тормоза.
INT16U	t4	Время в мс между включением тормоза и отключением питания мотора.
INT8U	BrakeFlags	Флаги.
		0x01 - BRAKE_ENABLED (Управление тормозом включено, если флаг установлен.)
		0x02 - BRAKE_ENG_PWROFF (Тормоз отключает питание шагового мотора, если флаг установлен.)
INT8U	Reserved [10]	Зарезервировано (10 байт)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись настроек управления тормозом.

Команда GBRK

```
result_t get_brake_settings (device_t id, brake_settings_t* brake_settings)
```

Код команды (CMD): "gbrk" Или 0x6B726267.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (25 байт)

INT32U	CMD	Команда (возврат)
INT16U	t1	Время в мс между включением питания мотора и отключением тормоза.
INT16U	t2	Время в мс между отключением тормоза и готовностью к движению. Все команды движения начинают выполняться только по истечении этого времени.
INT16U	t3	Время в мс между остановкой мотора и включением тормоза.
INT16U	t4	Время в мс между включением тормоза и отключением питания мотора.
INT8U	BrakeFlags	Флаги.
		0x01 - BRAKE_ENABLED (Управление тормозом включено, если флаг установлен.)
		0x02 - BRAKE_ENG_PWROFF (Тормоз отключает питание шагового мотора, если флаг установлен.)
INT8U	Reserved [10]	Зарезервировано (10 байт)
INT16U	CRC	Контрольная сумма

Описание:
Чтение настроек управления тормозом.

Команда SCTL

```
result_t set_control_settings (device_t id, const control_settings_t* control_settings)
```

Код команды (CMD): "sctl" Или 0x6C746373.

Запрос: (93 байта)

INT32U	CMD	Команда
INT32U	MaxSpeed [10]	Массив скоростей (в полных шагах), использующийся при управлении джойстиком или кнопками влево/вправо. Диапазон: 0..100000.
INT8U	uMaxSpeed [10]	Массив скоростей (в 1/256 микрошагах), использующийся при управлении джойстиком или кнопками влево/вправо.
INT16U	Timeout [9]	timeout[i] - время в мс, по истечении которого устанавливается скорость max_speed[i+1] (используется только при управлении кнопками).
INT16U	MaxClickTime	Максимальное время клика. До истечения этого времени первая скорость не включается.
INT16U	Flags	Флаги.
		0x03 - CONTROL_MODE_BITS (Биты управления мотором с помощью джойстика или кнопок влево/вправо.)
		0x00 - CONTROL_MODE_OFF (Управление отключено.)
		0x01 - CONTROL_MODE_JOY (Управление с помощью джойстика.)
		0x02 - CONTROL_MODE_LR (Управление с помощью кнопок left/right.)
		0x04 - CONTROL_BTN_LEFT_PUSHED_OPEN (Левая кнопка нормально разомкнутая, если флаг установлен.)
		0x08 - CONTROL_BTN_RIGHT_PUSHED_OPEN (Правая кнопка нормально разомкнутая, если флаг установлен.)
INT32S	DeltaPosition	Смещение (дельта) позиции
INT16S	uDeltaPosition	Дробная часть смещения в микрошагах. Используется только с шаговым двигателем. Диапазон: -255..255.
INT8U	Reserved [9]	Зарезервировано (9 байт)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись настроек управления мотором. При выборе CTL_MODE=1 включается управление мотором с помощью джойстика. В этом режиме при отклонении джойстика на максимум двигатель стремится двигаться со скоростью MaxSpeed [i], где i=0, если предыдущим использованием этого режима не было выбрано другое i. Кнопки переключают номер скорости i. При выборе CTL_MODE=2 включается управление мотором с помощью кнопок left/right. При нажатии на кнопки двигатель начинает двигаться в соответствующую сторону со скоростью MaxSpeed [0], по истечении времени Timeout[i] мотор двигается со скоростью MaxSpeed [i+1]. При переходе от MaxSpeed [i] на MaxSpeed [i+1] действует ускорение, как обычно.

Команда GCTL

```
result_t get_control_settings (device_t id, control_settings_t* control_settings)
```

Код команды (CMD): "gctl" Или 0x6C746367.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (93 байта)

INT32U	CMD	Команда (возврат)
INT32U	MaxSpeed [10]	Массив скоростей (в полных шагах), использующийся при управлении джойстиком или кнопками влево/вправо. Диапазон: 0..100000.
INT8U	uMaxSpeed [10]	Массив скоростей (в 1/256 микрошагах), использующийся при управлении джойстиком или кнопками влево/вправо.
INT16U	Timeout [9]	timeout[i] - время в мс, по истечении которого устанавливается скорость max_speed[i+1] (используется только при управлении кнопками).
INT16U	MaxClickTime	Максимальное время клика. До истечения этого времени первая скорость не включается.
INT16U	Flags	Флаги.
		0x03 - CONTROL_MODE_BITS (Биты управления мотором с помощью джойстика или кнопок влево/вправо.)
		0x00 - CONTROL_MODE_OFF (Управление отключено.)
		0x01 - CONTROL_MODE_JOY (Управление с помощью джойстика.)
		0x02 - CONTROL_MODE_LR (Управление с помощью кнопок left/right.)
		0x04 - CONTROL_BTN_LEFT_PUSHED_OPEN (Левая кнопка нормально разомкнутая, если флаг установлен.)
		0x08 - CONTROL_BTN_RIGHT_PUSHED_OPEN (Правая кнопка нормально разомкнутая, если флаг установлен.)
INT32S	DeltaPosition	Смещение (дельта) позиции
INT16S	uDeltaPosition	Дробная часть смещения в микрошагах. Используется только с шаговым двигателем. Диапазон: -255..255.
INT8U	Reserved [9]	Зарезервировано (9 байт)
INT16U	CRC	Контрольная сумма

Описание:
Чтение настроек управления мотором. При выборе CTL_MODE=1 включается управление мотором с помощью джойстика. В этом режиме при отклонении джойстика на максимум двигатель стремится двигаться со скоростью MaxSpeed [i], где i=0, если предыдущим использованием этого режима не было выбрано другое i. Кнопки переключают номер скорости i. При выборе CTL_MODE=2 включается управление мотором с помощью кнопок left/right. При нажатии на кнопки двигатель начинает двигаться в соответствующую сторону со скоростью MaxSpeed [0], по истечении времени Timeout[i] мотор двигается со скоростью MaxSpeed [i+1]. При переходе от MaxSpeed [i] на MaxSpeed [i+1] действует ускорение, как обычно.

Команда SJOY

```
result_t set_joystick_settings (device_t id, const joystick_settings_t* joystick_settings)
```

Код команды (CMD): "sjoy" Или 0x796F6A73.

Запрос: (22 байта)

INT32U	CMD	Команда
INT16U	JoyLowEnd	Значение в шагах джойстика, соответствующее нижней границе диапазона отклонения устройства. Должно лежать в пределах. Диапазон: 0..10000.
INT16U	JoyCenter	Значение в шагах джойстика, соответствующее неотклонённому устройству. Должно лежать в пределах. Диапазон: 0..10000.
INT16U	JoyHighEnd	Значение в шагах джойстика, соответствующее верхней границе диапазона отклонения устройства. Должно лежать в пределах. Диапазон: 0..10000.
INT8U	ExpFactor	Фактор экспоненциальной нелинейности отклика джойстика.
INT8U	DeadZone	Отклонение от среднего положения, которое не вызывает начала движения (в десятых долях процента). Максимальное мёртвое отклонение +-25.5%, что составляет половину рабочего диапазона джойстика.
INT8U	JoyFlags	Флаги управления джойстиком.
		0x01 - JOY_REVERSE (Реверс воздействия джойстика. Отклонение джойстика к большим значениям приводит к отрицательной скорости и наоборот.)
INT8U	Reserved [7]	Зарезервировано (7 байт)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись настроек джойстика. При отклонении джойстика более чем на DeadZone от центрального положения начинается движение со скоростью, определяемой отклонением джойстика от DeadZone до 100% отклонения, причем отклонению DeadZone соответствует нулевая скорость, а 100% отклонения соответствует MaxSpeed [i] (см. команду SCTL), где i=0, если предыдущим использованием этого режима не было выбрано другое i. Если следующая скорость в таблице скоростей нулевая (целая и микрошаговая части), то перехода на неё не происходит. DeadZone вычисляется в десятых долях процента отклонения от центра (JoyCenter) до правого или левого максимума. Расчёт DeadZone проиллюстрирован на графике:

Зависимость между отклонением и скоростью экспоненциальная, что позволяет без переключения режимов скорости сочетать высокую подвижность и точность. На графике ниже показан пример экспоненциальной зависимости скорости и работы мертвой зоны.

Параметр нелинейнойсти можно менять. Нулевой параметр нелинейности соответствует линейной зависимости.

Команда GJOY

```
result_t get_joystick_settings (device_t id, joystick_settings_t* joystick_settings)
```

Код команды (CMD): "gjoy" Или 0x796F6A67.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (22 байта)

INT32U	CMD	Команда (возврат)
INT16U	JoyLowEnd	Значение в шагах джойстика, соответствующее нижней границе диапазона отклонения устройства. Должно лежать в пределах. Диапазон: 0..10000.
INT16U	JoyCenter	Значение в шагах джойстика, соответствующее неотклонённому устройству. Должно лежать в пределах. Диапазон: 0..10000.
INT16U	JoyHighEnd	Значение в шагах джойстика, соответствующее верхней границе диапазона отклонения устройства. Должно лежать в пределах. Диапазон: 0..10000.
INT8U	ExpFactor	Фактор экспоненциальной нелинейности отклика джойстика.
INT8U	DeadZone	Отклонение от среднего положения, которое не вызывает начала движения (в десятых долях процента). Максимальное мёртвое отклонение +-25.5%, что составляет половину рабочего диапазона джойстика.
INT8U	JoyFlags	Флаги управления джойстиком.
		0x01 - JOY_REVERSE (Реверс воздействия джойстика. Отклонение джойстика к большим значениям приводит к отрицательной скорости и наоборот.)
INT8U	Reserved [7]	Зарезервировано (7 байт)
INT16U	CRC	Контрольная сумма

Описание:
Чтение настроек джойстика. При отклонении джойстика более чем на DeadZone от центрального положения начинается движение со скоростью, определяемой отклонением джойстика от DeadZone до 100% отклонения, причем отклонению DeadZone соответствует нулевая скорость, а 100% отклонения соответствует MaxSpeed [i] (см. команду SCTL), где i=0, если предыдущим использованием этого режима не было выбрано другое i. Если следующая скорость в таблице скоростей нулевая (целая и микрошаговая части), то перехода на неё не происходит. DeadZone вычисляется в десятых долях процента отклонения от центра (JoyCenter) до правого или левого максимума. Расчёт DeadZone проиллюстрирован на графике:

Зависимость между отклонением и скоростью экспоненциальная, что позволяет без переключения режимов скорости сочетать высокую подвижность и точность. На графике ниже показан пример экспоненциальной зависимости скорости и работы мертвой зоны.

Параметр нелинейнойсти можно менять. Нулевой параметр нелинейности соответствует линейной зависимости.

Команда SCTP

```
result_t set_ctp_settings (device_t id, const ctp_settings_t* ctp_settings)
```

Код команды (CMD): "sctp" Или 0x70746373.

Запрос: (18 байт)

INT32U	CMD	Команда
INT8U	CTPMinError	Минимальное отличие шагов ШД от положения энкодера, устанавливающее флаг STATE_RT_ERROR. Измеряется в шагах ШД.
INT8U	CTPFlags	Флаги.
		0x01 - CTP_ENABLED (Контроль позиции включен, если флаг установлен.)
		0x02 - CTP_BASE (Опорой является датчик оборотов, если флаг установлен; иначе - энкодер.)
		0x04 - CTP_ALARM_ON_ERROR (Войти в состояние ALARM при расхождении позиции, если флаг установлен.)
		0x08 - REV_SENS_INV (Сенсор считается активным, когда на нём 0, а инвертирование делает активным уровнем 1. То есть если не инвертировать, то действует обычная логика - 0 это срабатывание/активация/активное состояние.)
		0x10 - CTP_ERROR_CORRECTION (Корректировать ошибки, возникающие при проскальзывании, если флаг установлен. Работает только с энкодером. Несовместимо с флагом CTP_ALARM_ON_ERROR)
INT8U	Reserved [10]	Зарезервировано (10 байт)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись настроек контроля позиции(для шагового двигателя). При управлении ШД с энкодером (CTP_BASE 0) появляется возможность обнаруживать потерю шагов. Контроллер знает кол-во шагов на оборот (GENG::StepsPerRev) и разрешение энкодера (GFBS::IPT). При включении контроля (флаг CTP_ENABLED), контроллер запоминает текущую позицию в шагах ШД и текущую позицию энкодера. Далее, на каждом шаге позиция энкодера преобразовывается в шаги и если разница оказывается больше CTPMinError, устанавливается флаг STATE_CTP_ERROR. При управлении ШД с датчиком оборотов (CTP_BASE 1), позиция контролируется по нему. По активному фронту на входе синхронизации контроллер запоминает текущее значение шагов. Далее, при каждом обороте проверяет, на сколько шагов сместились. При рассогласовании более CTPMinError устанавливается флаг STATE_CTP_ERROR.

Команда GCTP

```
result_t get_ctp_settings (device_t id, ctp_settings_t* ctp_settings)
```

Код команды (CMD): "gctp" Или 0x70746367.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (18 байт)

INT32U	CMD	Команда (возврат)
INT8U	CTPMinError	Минимальное отличие шагов ШД от положения энкодера, устанавливающее флаг STATE_RT_ERROR. Измеряется в шагах ШД.
INT8U	CTPFlags	Флаги.
		0x01 - CTP_ENABLED (Контроль позиции включен, если флаг установлен.)
		0x02 - CTP_BASE (Опорой является датчик оборотов, если флаг установлен; иначе - энкодер.)
		0x04 - CTP_ALARM_ON_ERROR (Войти в состояние ALARM при расхождении позиции, если флаг установлен.)
		0x08 - REV_SENS_INV (Сенсор считается активным, когда на нём 0, а инвертирование делает активным уровнем 1. То есть если не инвертировать, то действует обычная логика - 0 это срабатывание/активация/активное состояние.)
		0x10 - CTP_ERROR_CORRECTION (Корректировать ошибки, возникающие при проскальзывании, если флаг установлен. Работает только с энкодером. Несовместимо с флагом CTP_ALARM_ON_ERROR)
INT8U	Reserved [10]	Зарезервировано (10 байт)
INT16U	CRC	Контрольная сумма

Описание:
Чтение настроек контроля позиции(для шагового двигателя). При управлении ШД с энкодером (CTP_BASE 0) появляется возможность обнаруживать потерю шагов. Контроллер знает кол-во шагов на оборот (GENG::StepsPerRev) и разрешение энкодера (GFBS::IPT). При включении контроля (флаг CTP_ENABLED), контроллер запоминает текущую позицию в шагах ШД и текущую позицию энкодера. Далее, на каждом шаге позиция энкодера преобразовывается в шаги и если разница оказывается больше CTPMinError, устанавливается флаг STATE_CTP_ERROR. При управлении ШД с датчиком оборотов (CTP_BASE 1), позиция контролируется по нему. По активному фронту на входе синхронизации контроллер запоминает текущее значение шагов. Далее, при каждом обороте проверяет, на сколько шагов сместились. При рассогласовании более CTPMinError устанавливается флаг STATE_CTP_ERROR.

Команда SURT

```
result_t set_uart_settings (device_t id, const uart_settings_t* uart_settings)
```

Код команды (CMD): "surt" Или 0x74727573.

Запрос: (16 байт)

INT32U	CMD	Команда
INT32U	Speed	Скорость UART
INT16U	UARTSetupFlags	Флаги настройки UART
		0x03 - UART_PARITY_BITS (Биты, отвечающие за выбор четности.)
		0x00 - UART_PARITY_BIT_EVEN (Бит 1, если чет)
		0x01 - UART_PARITY_BIT_ODD (Бит 1, если нечет)
		0x02 - UART_PARITY_BIT_SPACE (Бит четности всегда 0)
		0x03 - UART_PARITY_BIT_MARK (Бит четности всегда 1)
		0x04 - UART_PARITY_BIT_USE (Бит чётности не используется, если "0"; бит четности используется, если "1")
		0x08 - UART_STOP_BIT (Если установлен, один стоповый бит; иначе - 2 стоповых бита)
INT8U	Reserved [4]	Зарезервировано (4 байта)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Команда записи настроек UART. Эта функция записывает структуру настроек UART в память контроллера.

Команда GURT

```
result_t get_uart_settings (device_t id, uart_settings_t* uart_settings)
```

Код команды (CMD): "gurt" Или 0x74727567.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (16 байт)

INT32U	CMD	Команда (возврат)
INT32U	Speed	Скорость UART
INT16U	UARTSetupFlags	Флаги настройки UART
		0x03 - UART_PARITY_BITS (Биты, отвечающие за выбор четности.)
		0x00 - UART_PARITY_BIT_EVEN (Бит 1, если чет)
		0x01 - UART_PARITY_BIT_ODD (Бит 1, если нечет)
		0x02 - UART_PARITY_BIT_SPACE (Бит четности всегда 0)
		0x03 - UART_PARITY_BIT_MARK (Бит четности всегда 1)
		0x04 - UART_PARITY_BIT_USE (Бит чётности не используется, если "0"; бит четности используется, если "1")
		0x08 - UART_STOP_BIT (Если установлен, один стоповый бит; иначе - 2 стоповых бита)
INT8U	Reserved [4]	Зарезервировано (4 байта)
INT16U	CRC	Контрольная сумма

Описание:
Команда чтения настроек UART. Эта функция заполняет структуру настроек UART.

Команда SCAL

```
result_t set_calibration_settings (device_t id, const calibration_settings_t* calibration_settings)
```

Код команды (CMD): "scal" Или 0x6C616373.

Запрос: (118 байт)

INT32U	CMD	Команда
FLT32	CSS1_A	Коэффициент масштабирования для аналоговых измерений тока в обмотке A.
FLT32	CSS1_B	Коэффициент сдвига для аналоговых измерений тока в обмотке A.
FLT32	CSS2_A	Коэффициент масштабирования для аналоговых измерений тока в обмотке B.
FLT32	CSS2_B	Коэффициент сдвига для аналоговых измерений тока в обмотке B.
FLT32	FullCurrent_A	Коэффициент масштабирования для аналоговых измерений полного тока.
FLT32	FullCurrent_B	Коэффициент сдвига для аналоговых измерений полного тока.
INT8U	Reserved [88]	Зарезервировано (88 байт)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Команда записи калибровочных коэффициентов. Эта функция записывает структуру калибровочных коэффициентов в память контроллера.

Команда GCAL

```
result_t get_calibration_settings (device_t id, calibration_settings_t* calibration_settings)
```

Код команды (CMD): "gcal" Или 0x6C616367.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (118 байт)

INT32U	CMD	Команда (возврат)
FLT32	CSS1_A	Коэффициент масштабирования для аналоговых измерений тока в обмотке A.
FLT32	CSS1_B	Коэффициент сдвига для аналоговых измерений тока в обмотке A.
FLT32	CSS2_A	Коэффициент масштабирования для аналоговых измерений тока в обмотке B.
FLT32	CSS2_B	Коэффициент сдвига для аналоговых измерений тока в обмотке B.
FLT32	FullCurrent_A	Коэффициент масштабирования для аналоговых измерений полного тока.
FLT32	FullCurrent_B	Коэффициент сдвига для аналоговых измерений полного тока.
INT8U	Reserved [88]	Зарезервировано (88 байт)
INT16U	CRC	Контрольная сумма

Описание:
Команда чтения калибровочных коэффициентов. Эта функция заполняет структуру калибровочных коэффициентов.

Команда SNMF

```
result_t set_controller_name (device_t id, const controller_name_t* controller_name)
```

Код команды (CMD): "snmf" Или 0x666D6E73.

Запрос: (30 байт)

INT32U	CMD	Команда
CHAR	ControllerName [16]	Пользовательское имя контроллера. Может быть установлено пользователем для его удобства. Максимальная длина строки: 16 символов.
INT8U	CtrlFlags	Настройки контроллера.
		0x01 - EEPROM_PRECEDENCE (Если флаг установлен, то настройки в EEPROM подвижки имеют приоритет над текущими настройками и заменяют их при обнаружении EEPROM.)
INT8U	Reserved [7]	Зарезервировано (7 байт)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись пользовательского имени контроллера и настроек в FRAM.

Команда GNMF

```
result_t get_controller_name (device_t id, controller_name_t* controller_name)
```

Код команды (CMD): "gnmf" Или 0x666D6E67.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (30 байт)

INT32U	CMD	Команда (возврат)
CHAR	ControllerName [16]	Пользовательское имя контроллера. Может быть установлено пользователем для его удобства. Максимальная длина строки: 16 символов.
INT8U	CtrlFlags	Настройки контроллера.
		0x01 - EEPROM_PRECEDENCE (Если флаг установлен, то настройки в EEPROM подвижки имеют приоритет над текущими настройками и заменяют их при обнаружении EEPROM.)
INT8U	Reserved [7]	Зарезервировано (7 байт)
INT16U	CRC	Контрольная сумма

Описание:
Чтение пользовательского имени контроллера и настроек из FRAM.

Команда SNVM

```
result_t set_nonvolatile_memory (device_t id, const nonvolatile_memory_t* nonvolatile_memory)
```

Код команды (CMD): "snvm" Или 0x6D766E73.

Запрос: (36 байт)

INT32U	CMD	Команда
INT32U	UserData [7]	Пользовательские данные. Могут быть установлены пользователем для его удобства. Каждый элемент массива хранит только 32 бита пользовательских данных. Это важно на системах где тип int содержит больше чем 4 байта. Например это все системы amd64
INT8U	Reserved [2]	Зарезервировано (2 байта)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись пользовательских данных во FRAM.

Команда GNVM

```
result_t get_nonvolatile_memory (device_t id, nonvolatile_memory_t* nonvolatile_memory)
```

Код команды (CMD): "gnvm" Или 0x6D766E67.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (36 байт)

INT32U	CMD	Команда (возврат)
INT32U	UserData [7]	Пользовательские данные. Могут быть установлены пользователем для его удобства. Каждый элемент массива хранит только 32 бита пользовательских данных. Это важно на системах где тип int содержит больше чем 4 байта. Например это все системы amd64
INT8U	Reserved [2]	Зарезервировано (2 байта)
INT16U	CRC	Контрольная сумма

Описание:
Чтение пользовательских данных из FRAM.

Группа команд управления движением

Команда STOP

```
result_t command_stop (device_t id)
```

Код команды (CMD): "stop" Или 0x706F7473.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Немедленная остановка двигателя, переход в состояние STOP, ключи в режиме BREAK (обмотки накоротко замкнуты), режим "удержания" дезактивируется для DC двигателей, удержание тока в обмотках для шаговых двигателей (с учётом Power management настроек).

Команда ASIA

```
result_t command_add_sync_in_action (device_t id, const command_add_sync_in_action_t* the_command_add_sync_
in_action)
```

Код команды (CMD): "asia" Или 0x61697361.

Запрос: (22 байта)

INT32U	CMD	Команда
INT32S	Position	Желаемая позиция или смещение (целая часть)
INT16S	uPosition	Дробная часть позиции или смещения в микрошагах. Используется только с шаговым двигателем. Диапазон: -255..255.
INT32U	Time	Время, за которое требуется достичь требуемой позиции, в микросекундах.
INT8U	Reserved [6]	Зарезервировано (6 байт)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Это команда добавляет один элемент в буфер FIFO команд, выполняемых при получении входного импульса синхронизации. Каждый импульс синхронизации либо выполнится то действие, которое описано в SSNI, если буфер пуст, либо самое старое из загруженных в буфер действий временно подменяет скорость и координату в SSNI. В последнем случае это действие стирается из буфера. Количество оставшихся пустыми элементов буфера можно узнать в структуре GETS.

Команда PWOF

```
result_t command_power_off (device_t id)
```

Код команды (CMD): "pwof" Или 0x666F7770.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Немедленное отключение питания двигателя вне зависимости от его состояния. Команда предначена для ручного управления питанием двигателя. Не следует использовать эту команду для отключения двигателя во время движения, так как питание может снова включиться для завершения движения. Для автоматического управления питанием двигателя и его отключении после остановки следует использовать систему управления электропитанием.

Команда MOVE

```
result_t command_move (device_t id, int Position, int uPosition)
```

Код команды (CMD): "move" Или 0x65766F6D.

Запрос: (18 байт)

INT32U	CMD	Команда
INT32S	Position	Желаемая позиция (целая часть).
INT16S	uPosition	Дробная часть позиции в микрошагах. Используется только с шаговым двигателем. Диапазон: -255..255.
INT8U	Reserved [6]	Зарезервировано (6 байт)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
При получении команды "move" двигатель начинает перемещаться (если не используется режим "ТТЛСинхроВхода"), с заранее установленными параметрами (скорость, ускорение, удержание), к точке указанной в полях Position, uPosition. Для шагового мотора uPosition задает значение микрошага, для DC мотора это поле не используется.

Команда MOVR

```
result_t command_movr (device_t id, int DeltaPosition, int uDeltaPosition)
```

Код команды (CMD): "movr" Или 0x72766F6D.

Запрос: (18 байт)

INT32U	CMD	Команда
INT32S	DeltaPosition	Смещение (дельта) позиции
INT16S	uDeltaPosition	Дробная часть смещения в микрошагах, используется только с шаговым двигателем. Диапазон: -255..255.
INT8U	Reserved [6]	Зарезервировано (6 байт)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
При получении команды "movr" двигатель начинает смещаться (если не используется режим "ТТЛСинхроВхода"), с заранее установленными параметрами (скорость, ускорение, удержание), влево или вправо (зависит от знака DeltaPosition) на количество импульсов указанное в полях DeltaPosition, uDeltaPosition. Для шагового мотора uDeltaPosition задает значение микрошага, для DC мотора это поле не используется.

Команда HOME

```
result_t command_home (device_t id)
```

Код команды (CMD): "home" Или 0x656D6F68.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Поля скоростей знаковые. Положительное направление это вправо. Нулевое значение флага направления инвертирует направление, заданное скоростью. Ограничение, накладываемые концевиками, действуют так же, за исключением того, что касание концевика не приводит к остановке. Ограничения максимальной скорости, ускорения и замедления действуют. 1) Двигает мотор согласно скоростям FastHome, uFastHome и флагу HOME_DIR_FAST до достижения концевика, если флаг HOME_STOP_ENDS установлен, до достижения сигнала с входа синхронизации, если установлен флаг HOME_STOP_SYNC (важно как можно точнее поймать момент срабатывания концевика) или до поступления сигнала с датчика оборотов, если установлен флаг HOME_STOP_REV_SN 2) далее двигает согласно скоростям SlowHome, uSlowHome и флагу HOME_DIR_SLOW до достижения сигнала с входа синхронизации, если установлен флаг HOME_MV_SEC. Если флаг HOME_MV_SEC сброшен, пропускаем этот пункт. 3) далее двигает мотор согласно скоростям FastHome, uFastHome и флагу HOME_DIR_SLOW на расстояние HomeDelta, uHomeDelta. Описание флагов и переменных см. описание команд GHOM/SHOM

Команда LEFT

```
result_t command_left (device_t id)
```

Код команды (CMD): "left" Или 0x7466656C.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
При получении команды "left" двигатель начинает смещаться, с заранее установленными параметрами (скорость, ускорение), влево.

Команда RIGT

```
result_t command_right (device_t id)
```

Код команды (CMD): "rigt" Или 0x74676972.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
При получении команды "rigt" двигатель начинает смещаться, с заранее установленными параметрами (скорость, ускорение), вправо.

Команда LOFT

```
result_t command_loft (device_t id)
```

Код команды (CMD): "loft" Или 0x74666F6C.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
При получении команды "loft" двигатель смещается из текущей точки на расстояние GENG::Antiplay, затем двигается в ту же точку.

Команда SSTP

```
result_t command_sstp (device_t id)
```

Код команды (CMD): "sstp" Или 0x70747373.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Плавная остановка. Двигатель останавливается с ускорением замедления.

Группа команд установки текущей позиции

Команда GPOS

```
result_t get_position (device_t id, get_position_t* the_get_position)
```

Код команды (CMD): "gpos" Или 0x736F7067.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (26 байт)

INT32U	CMD	Команда (возврат)
INT32S	Position	Позиция в основных шагах двигателя
INT16S	uPosition	Позиция в микрошагах(используется только с шаговыми двигателями).
INT64S	EncPosition	Позиция энкодера.
INT8U	Reserved [6]	Зарезервировано (6 байт)
INT16U	CRC	Контрольная сумма

Описание:
Считывает значение положения в шагах и микрошагах для шагового двигателя и в шагах энкодера всех двигателей.

Команда SPOS

```
result_t set_position (device_t id, const set_position_t* the_set_position)
```

Код команды (CMD): "spos" Или 0x736F7073.

Запрос: (26 байт)

INT32U	CMD	Команда
INT32S	Position	Позиция в основных шагах двигателя
INT16S	uPosition	Позиция в микрошагах(используется только с шаговыми двигателями).
INT64S	EncPosition	Позиция энкодера.
INT8U	PosFlags	Флаги
		0x01 - SETPOS_IGNORE_POSITION (Если установлен, то позиция в шагах и микрошагах не обновляется.)
		0x02 - SETPOS_IGNORE_ENCODER (Если установлен, то счётчик энкодера не обновляется.)
INT8U	Reserved [5]	Зарезервировано (5 байт)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Устанавливает произвольное значение положения в шагах и микрошагах для шагового двигателя и в шагах энкодера всех двигателей. То есть меняется основной показатель положения.

Команда ZERO

```
result_t command_zero (device_t id)
```

Код команды (CMD): "zero" Или 0x6F72657A.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Устанавливает текущую позицию и позицию в которую осуществляется движение по командам move и movr равными нулю для всех случаев, кроме движения к позиции назначения. В последнем случае установить нулём текущую позицию, а позицию назначения пересчитать так, что в абсолютном положении точка назначения не меняется. То есть если мы находились в точке 400 и двигались к 500, то команда Zero делает текущую позицию 0, а позицию назначения - 100. Не изменяет режим движения т.е. если движение осуществлялось, то оно продолжается; если мотор находился в режиме "удержания", то тип удержания сохраняется.

Группа команд сохранения и загрузки настроек

Команда SAVE

```
result_t command_save_settings (device_t id)
```

Код команды (CMD): "save" Или 0x65766173.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
При получении команды контроллер выполняет операцию сохранения текущих настроек во встроенную энергонезависимую память контроллера.

Команда READ

```
result_t command_read_settings (device_t id)
```

Код команды (CMD): "read" Или 0x64616572.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Чтение всех настроек контроллера из flash памяти в оперативную, заменяя текущие настройки.

Команда SARS

```
result_t command_save_robust_settings (device_t id)
```

Код команды (CMD): "sars" Или 0x73726173.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
При получении команды контроллер выполняет операцию сохранения важных настроек (калибровочные коэффициенты и т.п.) во встроенную энергонезависимую память контроллера.

Команда RERS

```
result_t command_read_robust_settings (device_t id)
```

Код команды (CMD): "rers" Или 0x73726572.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Чтение важных настроек (калибровочные коэффициенты и т.п.) контроллера из flash памяти в оперативную, заменяя текущие настройки.

Команда EESV

```
result_t command_eesave_settings (device_t id)
```

Код команды (CMD): "eesv" Или 0x76736565.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись настроек контроллера в EEPROM память позиционера, которые непосредственно связаны с позиционером и не меняются без его механической переделки. Функция должна использоваться только производителем.

Команда EERD

```
result_t command_eeread_settings (device_t id)
```

Код команды (CMD): "eerd" Или 0x64726565.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Чтение настроек контроллера из EEPROM памяти позиционера, которые непосредственно связаны с позиционером и не меняются без его механической переделки. Эта операция также автоматически выполняется при подключении позиционера с EEPROM памятью. Функция должна использоваться только производителем.

Группа команд получения статуса контроллера

Команда GETS

```
result_t get_status (device_t id, status_t* status)
```

Код команды (CMD): "gets" Или 0x73746567.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (54 байта)

INT32U	CMD	Команда (возврат)
INT8U	MoveSts	Состояние движения.
		0x01 - MOVE_STATE_MOVING (Если флаг установлен, то контроллер пытается вращать двигателем. Не используйте этот флаг для ожидания завершения команды движения. Вместо него используйте MVCMD_RUNNING из поля MvCmdSts.)
		0x02 - MOVE_STATE_TARGET_SPEED (Флаг устанавливается при достижении заданной скорости.)
		0x04 - MOVE_STATE_ANTIPLAY (Выполняется компенсация люфта, если флаг установлен.)
INT8U	MvCmdSts	Состояние команды движения (касается command_move, command_movr, command_left, command_right, command_stop, command_home, command_loft).
		0x3F - MVCMD_NAME_BITS (Битовая маска активной команды.)
		0x00 - MVCMD_UKNWN (Неизвестная команда.)
		0x01 - MVCMD_MOVE (Команда move.)
		0x02 - MVCMD_MOVR (Команда movr.)
		0x03 - MVCMD_LEFT (Команда left.)
		0x04 - MVCMD_RIGHT (Команда rigt.)
		0x05 - MVCMD_STOP (Команда stop.)
		0x06 - MVCMD_HOME (Команда home.)
		0x07 - MVCMD_LOFT (Команда loft.)
		0x08 - MVCMD_SSTP (Команда плавной остановки(SSTP).)

		0x40 - MVCMD_ERROR (Состояние завершения движения (1 - команда движения выполнена с ошибкой, 0 - команда движения выполнена корректно). Имеет смысл если MVCMD_RUNNING указывает на завершение движения.)
		0x80 - MVCMD_RUNNING (Состояние команды движения (0 - команда движения выполнена, 1 - команда движения сейчас выполняется).)
INT8U	PWRSts	Состояние питания шагового двигателя (используется только с шаговым двигателем).
		0x00 - PWR_STATE_UNKNOWN (Неизвестное состояние, которое не должно никогда реализовываться.)
		0x01 - PWR_STATE_OFF (Обмотки мотора разомкнуты и не управляются драйвером.)
		0x03 - PWR_STATE_NORM (Обмотки запитаны номинальным током.)
		0x04 - PWR_STATE_REDUCT (Обмотки намеренно запитаны уменьшенным током от рабочего для снижения потребляемой мощности.)
		0x05 - PWR_STATE_MAX (Обмотки запитаны максимально доступным током, который может выдать схема при данном напряжении питания.)
INT8U	EncSts	Состояние энкодера.
		0x00 - ENC_STATE_ABSENT (Энкодер не подключен.)
		0x01 - ENC_STATE_UNKNOWN (Состояние энкодера неизвестно.)
		0x02 - ENC_STATE_MALFUNC (Энкодер подключен и неисправен.)
		0x03 - ENC_STATE_REVERS (Энкодер подключен и исправен, но считает в другую сторону.)
		0x04 - ENC_STATE_OK (Энкодер подключен и работает адекватно.)
INT8U	WindSts	Состояние обмоток.
		0x00 - WIND_A_STATE_ABSENT (Обмотка A не подключена.)
		0x01 - WIND_A_STATE_UNKNOWN (Состояние обмотки A неизвестно.)
		0x02 - WIND_A_STATE_MALFUNC (Короткое замыкание на обмотке A.)
		0x03 - WIND_A_STATE_OK (Обмотка A работает адекватно.)
		0x00 - WIND_B_STATE_ABSENT (Обмотка B не подключена.)
		0x10 - WIND_B_STATE_UNKNOWN (Состояние обмотки B неизвестно.)
		0x20 - WIND_B_STATE_MALFUNC (Короткое замыкание на обмотке B.)
		0x30 - WIND_B_STATE_OK (Обмотка B работает адекватно.)
INT32S	CurPosition	Первичное поле, в котором хранится текущая позиция, как бы ни была устроена обратная связь. В случае работы с DC-мотором в этом поле находится текущая позиция по данным с энкодера, в случае работы с ШД-мотором в режиме, когда первичными являются импульсы, подаваемые на мотор, в этом поле содержится целое значение шагов текущей позиции.
INT16S	uCurPosition	Дробная часть текущей позиции в микрошагах (-255..255). Используется только с шаговым двигателем.
INT64S	EncPosition	Текущая позиция по данным с энкодера в импульсах энкодера, используется только если энкодер установлен, активизирован и не является основным датчиком положения, например при использовании энкодера совместно с шаговым двигателем для контроля проскальзования.
INT32S	CurSpeed	Текущая скорость.
INT16S	uCurSpeed	Дробная часть текущей скорости в микрошагах (-255..255). Используется только с шаговым двигателем.
INT16S	Ipwr	Ток потребления силовой части.
INT16S	Upwr	Напряжение на силовой части, десятки мВ.
INT16S	Iusb	Ток потребления по USB.
INT16S	Uusb	Напряжение на USB, десятки мВ.
INT16S	CurT	Температура процессора в десятых долях градусов цельсия.
INT32U	Flags	Флаги состояний.
		0x00003F - STATE_CONTR (Флаги состояния контроллера.)
		0x000001 - STATE_ERRC (Недопустимая команда.)
		0x000002 - STATE_ERRD (Нарушение целостности данных.)
		0x000004 - STATE_ERRV (Недопустимое значение данных.)
		0x000010 - STATE_EEPROM_CONNECTED (Подключена память EEPROM с настройками.)
		0x000020 - STATE_IS_HOMED (Калибровка выполнена)

		0x73FFC0 - STATE_SECUR (Флаги опасности.)
		0x000040 - STATE_ALARM (Контроллер находится в состоянии ALARM, показывая, что случилась какая-то опасная ситуация. В состоянии ALARM все команды игнорируются пока не будет послана команда STOP и состояние ALARM деактивируется.)
		0x000080 - STATE_CTP_ERROR (Контроль позиции нарушен(используется только с шаговым двигателем).)
		0x000100 - STATE_POWER_OVERHEAT (Перегрелась силовая часть платы.)
		0x000200 - STATE_CONTROLLER_OVERHEAT (Перегрелась микросхема контроллера.)
		0x000400 - STATE_OVERLOAD_POWER_VOLTAGE (Превышено напряжение на силовой части.)
		0x000800 - STATE_OVERLOAD_POWER_CURRENT (Превышен максимальный ток потребления силовой части.)
		0x001000 - STATE_OVERLOAD_USB_VOLTAGE (Превышено напряжение на USB.)
		0x002000 - STATE_LOW_USB_VOLTAGE (Слишком низкое напряжение на USB.)
		0x004000 - STATE_OVERLOAD_USB_CURRENT (Превышен максимальный ток потребления USB.)
		0x008000 - STATE_BORDERS_SWAP_MISSET (Достижение неверной границы.)
		0x010000 - STATE_LOW_POWER_VOLTAGE (Напряжение на силовой части ниже чем напряжение Low Voltage Protection)
		0x020000 - STATE_H_BRIDGE_FAULT (Получен сигнал от драйвера о неисправности)
		0x0C0000 - STATE_CURRENT_MOTOR_BITS (Биты, показывающие текущий рабочий мотор на платах с несколькими выходами для двигателей.)
		0x000000 - STATE_CURRENT_MOTOR0 (Мотор 0.)
		0x040000 - STATE_CURRENT_MOTOR1 (Мотор 1.)
		0x080000 - STATE_CURRENT_MOTOR2 (Мотор 2.)
		0x0C0000 - STATE_CURRENT_MOTOR3 (Мотор 3.)
		0x100000 - STATE_WINDING_RES_MISMATCH (Сопротивления обмоток отличаются друг от друга слишком сильно)
		0x200000 - STATE_ENCODER_FAULT (Получен сигнал от энкодера о неисправности)
		0x400000 - STATE_MOTOR_CURRENT_LIMIT (Превышен предел по току)
INT32U	GPIOFlags	Флаги состояний GPIO входов.
		0xFFFF - STATE_DIG_SIGNAL (Флаги цифровых сигналов.)
		0x0001 - STATE_RIGHT_EDGE (Достижение правой границы.)
		0x0002 - STATE_LEFT_EDGE (Достижение левой границы.)
		0x0004 - STATE_BUTTON_RIGHT (Состояние кнопки "вправо" (1, если нажата).)
		0x0008 - STATE_BUTTON_LEFT (Состояние кнопки "влево" (1, если нажата).)
		0x0010 - STATE_GPIO_PINOUT (Если флаг установлен, ввод/вывод общего назначения работает как выход; если флаг сброшен, ввод/вывод работает как вход.)
		0x0020 - STATE_GPIO_LEVEL (Состояние ввода/вывода общего назначения.)
		0x0040 - STATE_HALL_A (Состояние вывода датчика холла(a)(флаг "1", если датчик активен).)
		0x0080 - STATE_HALL_B (Состояние вывода датчика холла(b)(флаг "1", если датчик активен).)
		0x0100 - STATE_HALL_C (Состояние вывода датчика холла(c)(флаг "1", если датчик активен).)
		0x0200 - STATE_BRAKE (Состояние вывода управления тормозом(флаг "1" - если на тормоз подаётся питание, "0" - если тормоз не запитан).)
		0x0400 - STATE_REV_SENSOR (Состояние вывода датчика оборотов(флаг "1", если датчик активен).)
		0x0800 - STATE_SYNC_INPUT (Состояние входа синхронизации(1, если вход синхронизации активен).)
		0x1000 - STATE_SYNC_OUTPUT (Состояние выхода синхронизации(1, если выход синхронизации активен).)
		0x2000 - STATE_ENC_A (Состояние ножки A энкодера(флаг "1", если энкодер активен).)
		0x4000 - STATE_ENC_B (Состояние ножки B энкодера(флаг "1", если энкодер активен).)
INT8U	CmdBufFreeSpace	Это поле показывает количество свободных ячеек буфера цепочки синхронизации.
INT8U	Reserved [4]	Зарезервировано (4 байта)
INT16U	CRC	Контрольная сумма

Описание:
Возвращает информацию о текущем состоянии устройства.

Команда STMS

```
result_t command_start_measurements (device_t id)
```

Код команды (CMD): "stms" Или 0x736D7473.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Начать измерения и буферизацию скорости, ошибки следования.

Команда GETM

```
result_t get_measurements (device_t id, measurements_t* measurements)
```

Код команды (CMD): "getm" Или 0x6D746567.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (216 байт)

INT32U	CMD	Команда (возврат)
INT32S	Speed [25]	Текущая скорость.
INT32S	Error [25]	Текущая скорость.
INT32U	Length	Длина фактических данных в буфере.
INT8U	Reserved [6]	Зарезервировано (6 байт)
INT16U	CRC	Контрольная сумма

Описание:
Команда чтения буфера данных для построения графиков скорости и ошибки следования. Заполнение буфера начинается по команде "start_measurements". Буффер вмещает 25 точек, точки снимаются с периодом 1 мс. Для создания устойчивой системы следует считывать данные каждые 20 мс, если буффер полностью заполнен, то рекомендуется повторять считывания каждые 5 мс до момента пока буффер вновь не станет заполнен 20-ю точками.

Команда GETC

```
result_t get_chart_data (device_t id, chart_data_t* chart_data)
```

Код команды (CMD): "getc" Или 0x63746567.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (38 байт)

INT32U	CMD	Команда (возврат)
INT16S	WindingVoltageA	В случае ШД, напряжение на обмотке A; в случае бесщеточного, напряжение на первой обмотке; в случае DC на единственной.
INT16S	WindingVoltageB	В случае ШД, напряжение на обмотке B; в случае бесщеточного, напряжение на второй обмотке; в случае DC не используется.
INT16S	WindingVoltageC	В случае бесщеточного, напряжение на третьей обмотке; в случае ШД и DC не используется.
INT16S	WindingCurrentA	В случае ШД, ток в обмотке A; в случае бесщеточного, ток в первой обмотке; в случае DC в единственной.
INT16S	WindingCurrentB	В случае ШД, ток в обмотке B; в случае бесщеточного, ток в второй обмотке; в случае DC не используется.
INT16S	WindingCurrentC	В случае бесщеточного, ток в третьей обмотке; в случае ШД и DC не используется.
INT16U	Pot	Значение на аналоговом входе. Диапазон: 0..10000
INT16U	Joy	Положение джойстика в десятитысячных долях. Диапазон: 0..10000
INT16S	DutyCycle	Коэффициент заполнения ШИМ.
INT8U	Reserved [14]	Зарезервировано (14 байт)
INT16U	CRC	Контрольная сумма

Описание:
Команда чтения состояния обмоток и других не часто используемых данных. Предназначена в первую очередь для получения данных для построения графиков в паре с командой GETS.

Команда GETI

```
result_t get_device_information (device_t id, device_information_t* device_information)
```

Код команды (CMD): "geti" Или 0x69746567.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (36 байт)

INT32U	CMD	Команда (возврат)
CHAR	Manufacturer [4]	Производитель
CHAR	ManufacturerId [2]	Идентификатор производителя
CHAR	ProductDescription [8]	Описание продукта
INT8U	Major	Основной номер версии железа.
INT8U	Minor	Второстепенный номер версии железа.
INT16U	Release	Номер правок этой версии железа.
INT8U	Reserved [12]	Зарезервировано (12 байт)
INT16U	CRC	Контрольная сумма

Описание:
Возвращает информацию об устройстве. Доступна как из прошивки, так и из бутлоадера.

Команда GSER

```
result_t get_serial_number (device_t id, unsigned int* SerialNumber)
```

Код команды (CMD): "gser" Или 0x72657367.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (10 байт)

INT32U	CMD	Команда (возврат)
INT32U	SerialNumber	Серийный номер платы.
INT16U	CRC	Контрольная сумма

Описание:
Чтение серийного номера контроллера.

Группа команд для работы с прошивкой контроллера

Команда GFWV

```
result_t get_firmware_version (device_t id, unsigned int* Major, unsigned int* Minor, unsigned int* Release
)
```

Код команды (CMD): "gfwv" Или 0x76776667.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (10 байт)

INT32U	CMD	Команда (возврат)
INT8U	Major	Мажорный номер версии прошивки
INT8U	Minor	Минорный номер версии прошивки
INT16U	Release	Номер релиза версии прошивки
INT16U	CRC	Контрольная сумма

Описание:
Чтение номера версии прошивки контроллера.

Команда UPDF

```
result_t service_command_updf (device_t id)
```

Код команды (CMD): "updf" Или 0x66647075.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Команда переводит контроллер в режим обновления прошивки. Получив такую команду, прошивка платы устанавливает флаг (для загрузчика), отправляет эхо-ответ и перезагружает контроллер.

Группа сервисных команд

Команда SSER

```
result_t set_serial_number (device_t id, const serial_number_t* serial_number)
```

Код команды (CMD): "sser" Или 0x72657373.

Запрос: (50 байт)

INT32U	CMD	Команда
INT32U	SN	Новый серийный номер платы.
INT8U	Key [32]	Ключ защиты для установки серийного номера (256 бит).
INT8U	Major	Основной номер версии железа.
INT8U	Minor	Второстепенный номер версии железа.
INT16U	Release	Номер правок этой версии железа.
INT8U	Reserved [4]	Зарезервировано (4 байта)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись серийного номера и версии железа во flash память контроллера. Вместе с новым серийным номером и версией железа передаётся "Ключ", только при совпадении которого происходит изменение и сохранение. Функция используется только производителем.

Команда RDAN

```
result_t get_analog_data (device_t id, analog_data_t* analog_data)
```

Код команды (CMD): "rdan" Или 0x6E616472.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (76 байт)

INT32U	CMD	Команда (возврат)
INT16U	A1Voltage_ADC	"Выходное напряжение на 1 выводе обмотки A" необработанные данные с АЦП.
INT16U	A2Voltage_ADC	"Выходное напряжение на 2 выводе обмотки A" необработанные данные с АЦП.
INT16U	B1Voltage_ADC	"Выходное напряжение на 1 выводе обмотки B" необработанные данные с АЦП.
INT16U	B2Voltage_ADC	"Выходное напряжение на 2 выводе обмотки B" необработанные данные с АЦП.
INT16U	SupVoltage_ADC	"Напряжение питания ключей H-моста" необработанные данные с АЦП.
INT16U	ACurrent_ADC	"Ток через обмотку A" необработанные данные с АЦП.
INT16U	BCurrent_ADC	"Ток через обмотку B" необработанные данные с АЦП.
INT16U	FullCurrent_ADC	"Полный ток" необработанные данные с АЦП.
INT16U	Temp_ADC	Напряжение с датчика температуры, необработанные данные с АЦП.
INT16U	Joy_ADC	Джойстик, необработанные данные с АЦП.
INT16U	Pot_ADC	Напряжение на аналоговом входе, необработанные данные с АЦП
INT16U	L5_ADC	Напряжение питания USB после current sense резистора, необработанные данные с АЦП.
INT16U	H5_ADC	Напряжение питания USB, необработанные данные с АЦП
INT16S	A1Voltage	"Выходное напряжение на 1 выводе обмотки A" откалиброванные данные.
INT16S	A2Voltage	"Выходное напряжение на 2 выводе обмотки A" откалиброванные данные.
INT16S	B1Voltage	"Выходное напряжение на 1 выводе обмотки B" откалиброванные данные.
INT16S	B2Voltage	"Выходное напряжение на 2 выводе обмотки B" откалиброванные данные.
INT16S	SupVoltage	"Напряжение питания ключей H-моста" откалиброванные данные.
INT16S	ACurrent	"Ток через обмотку A" откалиброванные данные.
INT16S	BCurrent	"Ток через обмотку B" откалиброванные данные.
INT16S	FullCurrent	"Полный ток" откалиброванные данные.
INT16S	Temp	Температура, откалиброванные данные.
INT16S	Joy	Джойстик во внутренних единицах. Диапазон: 0..10000
INT16S	Pot	Аналоговый вход во внутренних единицах. Диапазон: 0..10000
INT16S	L5	Напряжение питания USB после current sense резистора
INT16S	H5	Напряжение питания USB
INT16U	deprecated	
INT32S	R	Сопротивление обмоток двигателя(для шагового двигателя), в мОм
INT32S	L	Псевдоиндуктивность обмоток двигателя(для шагового двигателя), в мкГн
INT8U	Reserved [8]	Зарезервировано (8 байт)
INT16U	CRC	Контрольная сумма

Описание:
Чтение аналоговых данных, содержащих данные с АЦП и нормированные значения величин. Эта функция используется для тестирования и калибровки устройства.

Команда DBGR

```
result_t get_debug_read (device_t id, debug_read_t* debug_read)
```

Код команды (CMD): "dbgr" Или 0x72676264.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (142 байта)

INT32U	CMD	Команда (возврат)
INT8U	DebugData [128]	Отладочные данные.
INT8U	Reserved [8]	Зарезервировано (8 байт)
INT16U	CRC	Контрольная сумма

Описание:
Чтение данных из прошивки для отладки и поиска неисправностей. Получаемые данные зависят от версии прошивки, истории и контекста использования.

Команда DBGW

```
result_t set_debug_write (device_t id, const debug_write_t* debug_write)
```

Код команды (CMD): "dbgw" Или 0x77676264.

Запрос: (142 байта)

INT32U	CMD	Команда
INT8U	DebugData [128]	Отладочные данные.
INT8U	Reserved [8]	Зарезервировано (8 байт)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись данных в прошивку для отладки и поиска неисправностей.

Группа команд работы с EEPROM подвижки

Команда SNME

```
result_t set_stage_name (device_t id, const stage_name_t* stage_name)
```

Код команды (CMD): "snme" Или 0x656D6E73.

Запрос: (30 байт)

INT32U	CMD	Команда
CHAR	PositionerName [16]	Пользовательское имя подвижки. Может быть установлено пользователем для его удобства. Максимальная длина строки: 16 символов.
INT8U	Reserved [8]	Зарезервировано (8 байт)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись пользовательского имени подвижки в EEPROM.

Команда GNME

```
result_t get_stage_name (device_t id, stage_name_t* stage_name)
```

Код команды (CMD): "gnme" Или 0x656D6E67.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (30 байт)

INT32U	CMD	Команда (возврат)
CHAR	PositionerName [16]	Пользовательское имя подвижки. Может быть установлено пользователем для его удобства. Максимальная длина строки: 16 символов.
INT8U	Reserved [8]	Зарезервировано (8 байт)
INT16U	CRC	Контрольная сумма

Описание:
Чтение пользовательского имени подвижки из EEPROM.

Команда SSTI

```
result_t set_stage_information (device_t id, const stage_information_t* stage_information)
```

Код команды (CMD): "ssti" Или 0x69747373.

Запрос: (70 байт)

INT32U	CMD	Команда
CHAR	Manufacturer [16]	Производитель. Максимальная длина строки: 16 символов.
CHAR	PartNumber [24]	Серия и номер модели. Максимальная длина строки: 24 символа.
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись информации о позиционере в EEPROM. Функция должна использоваться только производителем.

Команда GSTI

```
result_t get_stage_information (device_t id, stage_information_t* stage_information)
```

Код команды (CMD): "gsti" Или 0x69747367.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (70 байт)

INT32U	CMD	Команда (возврат)
CHAR	Manufacturer [16]	Производитель. Максимальная длина строки: 16 символов.
CHAR	PartNumber [24]	Серия и номер модели. Максимальная длина строки: 24 символа.
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Описание:
Чтение информации о позиционере из EEPROM.

Команда SSTS

```
result_t set_stage_settings (device_t id, const stage_settings_t* stage_settings)
```

Код команды (CMD): "ssts" Или 0x73747373.

Запрос: (70 байт)

INT32U	CMD	Команда
FLT32	LeadScrewPitch	Шаг ходового винта в мм. Тип данных: float.
CHAR	Units [8]	Единицы измерения расстояния, используемые в полях MaxSpeed и TravelRange (шаги, градусы, мм, ...), Максимальная длина строки: 8 символов.
FLT32	MaxSpeed	Максимальная скорость (Units/c). Тип данных: float.
FLT32	TravelRange	Диапазон перемещения (Units). Тип данных: float.
FLT32	SupplyVoltageMin	Минимальное напряжение питания (В). Тип данных: float.
FLT32	SupplyVoltageMax	Максимальное напряжение питания (В). Тип данных: float.
FLT32	MaxCurrentConsumption	Максимальный ток потребления (А). Тип данных: float.
FLT32	HorizontalLoadCapacity	Горизонтальная грузоподъемность (кг). Тип данных: float.
FLT32	VerticalLoadCapacity	Вертикальная грузоподъемность (кг). Тип данных: float.
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись настроек позиционера в EEPROM. Функция должна использоваться только производителем.

Команда GSTS

```
result_t get_stage_settings (device_t id, stage_settings_t* stage_settings)
```

Код команды (CMD): "gsts" Или 0x73747367.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (70 байт)

INT32U	CMD	Команда (возврат)
FLT32	LeadScrewPitch	Шаг ходового винта в мм. Тип данных: float.
CHAR	Units [8]	Единицы измерения расстояния, используемые в полях MaxSpeed и TravelRange (шаги, градусы, мм, ...), Максимальная длина строки: 8 символов.
FLT32	MaxSpeed	Максимальная скорость (Units/c). Тип данных: float.
FLT32	TravelRange	Диапазон перемещения (Units). Тип данных: float.
FLT32	SupplyVoltageMin	Минимальное напряжение питания (В). Тип данных: float.
FLT32	SupplyVoltageMax	Максимальное напряжение питания (В). Тип данных: float.
FLT32	MaxCurrentConsumption	Максимальный ток потребления (А). Тип данных: float.
FLT32	HorizontalLoadCapacity	Горизонтальная грузоподъемность (кг). Тип данных: float.
FLT32	VerticalLoadCapacity	Вертикальная грузоподъемность (кг). Тип данных: float.
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Описание:
Чтение настроек позиционера из EEPROM.

Команда SMTI

```
result_t set_motor_information (device_t id, const motor_information_t* motor_information)
```

Код команды (CMD): "smti" Или 0x69746D73.

Запрос: (70 байт)

INT32U	CMD	Команда
CHAR	Manufacturer [16]	Производитель. Максимальная длина строки: 16 символов.
CHAR	PartNumber [24]	Серия и номер модели. Максимальная длина строки: 24 символа.
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись информации о двигателе в EEPROM. Функция должна использоваться только производителем.

Команда GMTI

```
result_t get_motor_information (device_t id, motor_information_t* motor_information)
```

Код команды (CMD): "gmti" Или 0x69746D67.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (70 байт)

INT32U	CMD	Команда (возврат)
CHAR	Manufacturer [16]	Производитель. Максимальная длина строки: 16 символов.
CHAR	PartNumber [24]	Серия и номер модели. Максимальная длина строки: 24 символа.
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Описание:
Чтение информации о двигателе из EEPROM.

Команда SMTS

```
result_t set_motor_settings (device_t id, const motor_settings_t* motor_settings)
```

Код команды (CMD): "smts" Или 0x73746D73.

Запрос: (112 байт)

INT32U	CMD	Команда
INT8U	MotorType	Тип двигателя
		0x00 - MOTOR_TYPE_UNKNOWN (Неизвестный двигатель)
		0x01 - MOTOR_TYPE_STEP (Шаговый двигатель)
		0x02 - MOTOR_TYPE_DC (DC двигатель)
		0x03 - MOTOR_TYPE_BLDC (BLDC двигатель)
INT8U	ReservedField	Зарезервировано
INT16U	Poles	Кол-во пар полюсов у DC или BLDC двигателя или кол-во шагов на оборот для шагового двигателя.
INT16U	Phases	Кол-во фаз у BLDC двигателя.
FLT32	NominalVoltage	Номинальное напряжение на обмотке (В). Тип данных: float.
FLT32	NominalCurrent	Максимальный постоянный ток в обмотке для DC и BLDC двигателей, номинальный ток в обмотке для шаговых двигателей (А). Тип данных: float.
FLT32	NominalSpeed	Номинальная скорость (об/мин). Применяется для DC и BLDC двигателей. Тип данных: float.
FLT32	NominalTorque	Номинальный крутящий момент (мН м). Применяется для DC и BLDC двигателей. Тип данных: float.
FLT32	NominalPower	Номинальная мощность(Вт). Применяется для DC и BLDC двигателей. Тип данных: float.
FLT32	WindingResistance	Сопротивление обмотки DC двигателя, каждой из двух обмоток шагового двигателя или каждой из трёх обмоток BLDC двигателя (Ом). Тип данных: float.
FLT32	WindingInductance	Индуктивность обмотки DC двигателя, каждой из двух обмоток шагового двигателя или каждой из трёх обмоток BLDC двигателя (мГн). Тип данных: float.
FLT32	RotorInertia	Инерция ротора (г см2). Тип данных: float.
FLT32	StallTorque	Крутящий момент удержания позиции для шагового двигателя или крутящий момент при неподвижном роторе для других типов двигателей (мН м). Тип данных: float.
FLT32	DetentTorque	Момент удержания позиции с незапитанными обмотками (мН м). Тип данных: float.
FLT32	TorqueConstant	Константа крутящего момента, определяющая коэффициент пропорциональности максимального момента силы ротора от протекающего в обмотке тока (мН м/А). Используется в основном для DC двигателей. Тип данных: float.
FLT32	SpeedConstant	Константа скорости, определяющая значение или амплитуду напряжения наведённой индукции при вращении ротора DC или BLDC двигателя (об/мин / В) или шагового двигателя (шаг/с / В). Тип данных: float.
FLT32	SpeedTorqueGradient	Градиент крутящего момента (об/мин / мН м). Тип данных: float.
FLT32	MechanicalTimeConstant	Механическая постоянная времени (мс). Тип данных: float.
FLT32	MaxSpeed	Максимальная разрешённая скорость для шаговых двигателей (шаг/с) или для DC и BLDC двигателей (об/мин). Тип данных: float.
FLT32	MaxCurrent	Максимальный ток в обмотке (А). Тип данных: float.
FLT32	MaxCurrentTime	Безопасная длительность максимального тока в обмотке (мс). Тип данных: float.
FLT32	NoLoadCurrent	Ток потребления в холостом режиме (А). Применяется для DC и BLDC двигателей. Тип данных: float.
FLT32	NoLoadSpeed	Скорость в холостом режиме (об/мин). Применяется для DC и BLDC двигателей. Тип данных: float.
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись настроек двигателя в EEPROM. Функция должна использоваться только производителем.

Команда GMTS

```
result_t get_motor_settings (device_t id, motor_settings_t* motor_settings)
```

Код команды (CMD): "gmts" Или 0x73746D67.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (112 байт)

INT32U	CMD	Команда (возврат)
INT8U	MotorType	Тип двигателя
		0x00 - MOTOR_TYPE_UNKNOWN (Неизвестный двигатель)
		0x01 - MOTOR_TYPE_STEP (Шаговый двигатель)
		0x02 - MOTOR_TYPE_DC (DC двигатель)
		0x03 - MOTOR_TYPE_BLDC (BLDC двигатель)
INT8U	ReservedField	Зарезервировано
INT16U	Poles	Кол-во пар полюсов у DC или BLDC двигателя или кол-во шагов на оборот для шагового двигателя.
INT16U	Phases	Кол-во фаз у BLDC двигателя.
FLT32	NominalVoltage	Номинальное напряжение на обмотке (В). Тип данных: float.
FLT32	NominalCurrent	Максимальный постоянный ток в обмотке для DC и BLDC двигателей, номинальный ток в обмотке для шаговых двигателей (А). Тип данных: float.
FLT32	NominalSpeed	Номинальная скорость (об/мин). Применяется для DC и BLDC двигателей. Тип данных: float.
FLT32	NominalTorque	Номинальный крутящий момент (мН м). Применяется для DC и BLDC двигателей. Тип данных: float.
FLT32	NominalPower	Номинальная мощность(Вт). Применяется для DC и BLDC двигателей. Тип данных: float.
FLT32	WindingResistance	Сопротивление обмотки DC двигателя, каждой из двух обмоток шагового двигателя или каждой из трёх обмоток BLDC двигателя (Ом). Тип данных: float.
FLT32	WindingInductance	Индуктивность обмотки DC двигателя, каждой из двух обмоток шагового двигателя или каждой из трёх обмоток BLDC двигателя (мГн). Тип данных: float.
FLT32	RotorInertia	Инерция ротора (г см2). Тип данных: float.
FLT32	StallTorque	Крутящий момент удержания позиции для шагового двигателя или крутящий момент при неподвижном роторе для других типов двигателей (мН м). Тип данных: float.
FLT32	DetentTorque	Момент удержания позиции с незапитанными обмотками (мН м). Тип данных: float.
FLT32	TorqueConstant	Константа крутящего момента, определяющая коэффициент пропорциональности максимального момента силы ротора от протекающего в обмотке тока (мН м/А). Используется в основном для DC двигателей. Тип данных: float.
FLT32	SpeedConstant	Константа скорости, определяющая значение или амплитуду напряжения наведённой индукции при вращении ротора DC или BLDC двигателя (об/мин / В) или шагового двигателя (шаг/с / В). Тип данных: float.
FLT32	SpeedTorqueGradient	Градиент крутящего момента (об/мин / мН м). Тип данных: float.
FLT32	MechanicalTimeConstant	Механическая постоянная времени (мс). Тип данных: float.
FLT32	MaxSpeed	Максимальная разрешённая скорость для шаговых двигателей (шаг/с) или для DC и BLDC двигателей (об/мин). Тип данных: float.
FLT32	MaxCurrent	Максимальный ток в обмотке (А). Тип данных: float.
FLT32	MaxCurrentTime	Безопасная длительность максимального тока в обмотке (мс). Тип данных: float.
FLT32	NoLoadCurrent	Ток потребления в холостом режиме (А). Применяется для DC и BLDC двигателей. Тип данных: float.
FLT32	NoLoadSpeed	Скорость в холостом режиме (об/мин). Применяется для DC и BLDC двигателей. Тип данных: float.
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Описание:
Чтение настроек двигателя из EEPROM.

Команда SENI

```
result_t set_encoder_information (device_t id, const encoder_information_t* encoder_information)
```

Код команды (CMD): "seni" Или 0x696E6573.

Запрос: (70 байт)

INT32U	CMD	Команда
CHAR	Manufacturer [16]	Производитель. Максимальная длина строки: 16 символов.
CHAR	PartNumber [24]	Серия и номер модели. Максимальная длина строки: 24 символа.
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись информации об энкодере в EEPROM. Функция должна использоваться только производителем.

Команда GENI

```
result_t get_encoder_information (device_t id, encoder_information_t* encoder_information)
```

Код команды (CMD): "geni" Или 0x696E6567.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (70 байт)

INT32U	CMD	Команда (возврат)
CHAR	Manufacturer [16]	Производитель. Максимальная длина строки: 16 символов.
CHAR	PartNumber [24]	Серия и номер модели. Максимальная длина строки: 24 символа.
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Описание:
Чтение информации об энкодере из EEPROM.

Команда SENS

```
result_t set_encoder_settings (device_t id, const encoder_settings_t* encoder_settings)
```

Код команды (CMD): "sens" Или 0x736E6573.

Запрос: (54 байта)

INT32U	CMD	Команда
FLT32	MaxOperatingFrequency	Максимальная частота (кГц). Тип данных: float.
FLT32	SupplyVoltageMin	Минимальное напряжение питания (В). Тип данных: float.
FLT32	SupplyVoltageMax	Максимальное напряжение питания (В). Тип данных: float.
FLT32	MaxCurrentConsumption	Максимальное потребление тока (мА). Тип данных: float.
INT32U	PPR	Количество отсчётов на оборот
INT32U	EncoderSettings	Флаги настроек энкодера
		0x001 - ENCSET_DIFFERENTIAL_OUTPUT (Если флаг установлен, то энкодер имеет дифференциальный выход, иначе - несимметричный выход)
		0x004 - ENCSET_PUSHPULL_OUTPUT (Если флаг установлен, то энкодер имеет двухтактный выход, иначе - выход с открытым коллектором)
		0x010 - ENCSET_INDEXCHANNEL_PRESENT (Если флаг установлен, то энкодер имеет дополнительный индексный канал, иначе - он отсутствует)
		0x040 - ENCSET_REVOLUTIONSENSOR_PRESENT (Если флаг установлен, то энкодер имеет датчик оборотов, иначе - он отсутствует)
		0x100 - ENCSET_REVOLUTIONSENSOR_ACTIVE_HIGH (Если флаг установлен, то активное состояние датчика оборотов соответствует логической 1, иначе - логическому 0)
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись настроек энкодера в EEPROM. Функция должна использоваться только производителем.

Команда GENS

```
result_t get_encoder_settings (device_t id, encoder_settings_t* encoder_settings)
```

Код команды (CMD): "gens" Или 0x736E6567.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (54 байта)

INT32U	CMD	Команда (возврат)
FLT32	MaxOperatingFrequency	Максимальная частота (кГц). Тип данных: float.
FLT32	SupplyVoltageMin	Минимальное напряжение питания (В). Тип данных: float.
FLT32	SupplyVoltageMax	Максимальное напряжение питания (В). Тип данных: float.
FLT32	MaxCurrentConsumption	Максимальное потребление тока (мА). Тип данных: float.
INT32U	PPR	Количество отсчётов на оборот
INT32U	EncoderSettings	Флаги настроек энкодера
		0x001 - ENCSET_DIFFERENTIAL_OUTPUT (Если флаг установлен, то энкодер имеет дифференциальный выход, иначе - несимметричный выход)
		0x004 - ENCSET_PUSHPULL_OUTPUT (Если флаг установлен, то энкодер имеет двухтактный выход, иначе - выход с открытым коллектором)
		0x010 - ENCSET_INDEXCHANNEL_PRESENT (Если флаг установлен, то энкодер имеет дополнительный индексный канал, иначе - он отсутствует)
		0x040 - ENCSET_REVOLUTIONSENSOR_PRESENT (Если флаг установлен, то энкодер имеет датчик оборотов, иначе - он отсутствует)
		0x100 - ENCSET_REVOLUTIONSENSOR_ACTIVE_HIGH (Если флаг установлен, то активное состояние датчика оборотов соответствует логической 1, иначе - логическому 0)
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Описание:
Чтение настроек энкодера из EEPROM.

Команда SHSI

```
result_t set_hallsensor_information (device_t id, const hallsensor_information_t* hallsensor_information)
```

Код команды (CMD): "shsi" Или 0x69736873.

Запрос: (70 байт)

INT32U	CMD	Команда
CHAR	Manufacturer [16]	Производитель. Максимальная длина строки: 16 символов.
CHAR	PartNumber [24]	Серия и номер модели. Максимальная длина строки: 24 символа.
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись информации о датчиках Холла в EEPROM. Функция должна использоваться только производителем.

Команда GHSI

```
result_t get_hallsensor_information (device_t id, hallsensor_information_t* hallsensor_information)
```

Код команды (CMD): "ghsi" Или 0x69736867.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (70 байт)

INT32U	CMD	Команда (возврат)
CHAR	Manufacturer [16]	Производитель. Максимальная длина строки: 16 символов.
CHAR	PartNumber [24]	Серия и номер модели. Максимальная длина строки: 24 символа.
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Описание:
Чтение информации о датчиках Холла из EEPROM.

Команда SHSS

```
result_t set_hallsensor_settings (device_t id, const hallsensor_settings_t* hallsensor_settings)
```

Код команды (CMD): "shss" Или 0x73736873.

Запрос: (50 байт)

INT32U	CMD	Команда
FLT32	MaxOperatingFrequency	Максимальная частота (кГц). Тип данных: float.
FLT32	SupplyVoltageMin	Минимальное напряжение питания (В). Тип данных: float.
FLT32	SupplyVoltageMax	Максимальное напряжение питания (В). Тип данных: float.
FLT32	MaxCurrentConsumption	Максимальное потребление тока (мА). Тип данных: float.
INT32U	PPR	Количество отсчётов на оборот
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись настроек датчиков Холла в EEPROM. Функция должна использоваться только производителем.

Команда GHSS

```
result_t get_hallsensor_settings (device_t id, hallsensor_settings_t* hallsensor_settings)
```

Код команды (CMD): "ghss" Или 0x73736867.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (50 байт)

INT32U	CMD	Команда (возврат)
FLT32	MaxOperatingFrequency	Максимальная частота (кГц). Тип данных: float.
FLT32	SupplyVoltageMin	Минимальное напряжение питания (В). Тип данных: float.
FLT32	SupplyVoltageMax	Максимальное напряжение питания (В). Тип данных: float.
FLT32	MaxCurrentConsumption	Максимальное потребление тока (мА). Тип данных: float.
INT32U	PPR	Количество отсчётов на оборот
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Описание:
Чтение настроек датчиков Холла из EEPROM.

Команда SGRI

```
result_t set_gear_information (device_t id, const gear_information_t* gear_information)
```

Код команды (CMD): "sgri" Или 0x69726773.

Запрос: (70 байт)

INT32U	CMD	Команда
CHAR	Manufacturer [16]	Производитель. Максимальная длина строки: 16 символов.
CHAR	PartNumber [24]	Серия и номер модели. Максимальная длина строки: 24 символа.
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись информации о редукторе в EEPROM. Функция должна использоваться только производителем.

Команда GGRI

```
result_t get_gear_information (device_t id, gear_information_t* gear_information)
```

Код команды (CMD): "ggri" Или 0x69726767.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (70 байт)

INT32U	CMD	Команда (возврат)
CHAR	Manufacturer [16]	Производитель. Максимальная длина строки: 16 символов.
CHAR	PartNumber [24]	Серия и номер модели. Максимальная длина строки: 24 символа.
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Описание:
Чтение информации о редукторе из EEPROM.

Команда SGRS

```
result_t set_gear_settings (device_t id, const gear_settings_t* gear_settings)
```

Код команды (CMD): "sgrs" Или 0x73726773.

Запрос: (58 байт)

INT32U	CMD	Команда
FLT32	ReductionIn	Входной коэффициент редуктора. (Выход = (ReductionOut/ReductionIn) * вход) Тип данных: float.
FLT32	ReductionOut	Выходной коэффициент редуктора. (Выход = (ReductionOut/ReductionIn) * вход) Тип данных: float.
FLT32	RatedInputTorque	Максимальный крутящий момент (Н м). Тип данных: float.
FLT32	RatedInputSpeed	Максимальная скорость на входном валу редуктора (об/мин). Тип данных: float.
FLT32	MaxOutputBacklash	Выходной люфт редуктора (градус). Тип данных: float
FLT32	InputInertia	Эквивалентная входная инерция редуктора(г см2). Тип данных: float.
FLT32	Efficiency	КПД редуктора (%). Тип данных: float.
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись настроек редуктора в EEPROM. Функция должна использоваться только производителем.

Команда GGRS

```
result_t get_gear_settings (device_t id, gear_settings_t* gear_settings)
```

Код команды (CMD): "ggrs" Или 0x73726767.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (58 байт)

INT32U	CMD	Команда (возврат)
FLT32	ReductionIn	Входной коэффициент редуктора. (Выход = (ReductionOut/ReductionIn) * вход) Тип данных: float.
FLT32	ReductionOut	Выходной коэффициент редуктора. (Выход = (ReductionOut/ReductionIn) * вход) Тип данных: float.
FLT32	RatedInputTorque	Максимальный крутящий момент (Н м). Тип данных: float.
FLT32	RatedInputSpeed	Максимальная скорость на входном валу редуктора (об/мин). Тип данных: float.
FLT32	MaxOutputBacklash	Выходной люфт редуктора (градус). Тип данных: float
FLT32	InputInertia	Эквивалентная входная инерция редуктора(г см2). Тип данных: float.
FLT32	Efficiency	КПД редуктора (%). Тип данных: float.
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Описание:
Чтение настроек редуктора из EEPROM.

Команда SACC

```
result_t set_accessories_settings (device_t id, const accessories_settings_t* accessories_settings)
```

Код команды (CMD): "sacc" Или 0x63636173.

Запрос: (114 байт)

INT32U	CMD	Команда
CHAR	MagneticBrakeInfo [24]	Производитель и номер магнитного тормоза, Максимальная длина строки: 24 символов.
FLT32	MBRatedVoltage	Номинальное напряжение для управления магнитным тормозом (В). Тип данных: float.
FLT32	MBRatedCurrent	Номинальный ток для управления магнитным тормозом (А). Тип данных: float.
FLT32	MBTorque	Удерживающий момент (мН м). Тип данных: float.
INT32U	MBSettings	Флаги настроек магнитного тормоза.
		0x01 - MB_AVAILABLE (Если флаг установлен, то магнитный тормоз доступен)
		0x02 - MB_POWERED_HOLD (Если флаг установлен, то магнитный тормоз находится в режиме удержания (активен) при подаче питания)
CHAR	TemperatureSensorInfo [24]	Производитель и номер температурного датчика, Максимальная длина строки: 24 символов.
FLT32	TSMin	Минимальная измеряемая температура (град Цельсия). Тип данных: float.
FLT32	TSMax	Максимальная измеряемая температура (град Цельсия) Тип данных: float.
FLT32	TSGrad	Температурный градиент (В/град Цельсия). Тип данных: float.
INT32U	TSSettings	Флаги настроек температурного датчика.
		0x07 - TS_TYPE_BITS (Биты, отвечающие за тип температурного датчика.)
		0x00 - TS_TYPE_UNKNOWN (Неизвестный сенсор)
		0x01 - TS_TYPE_THERMOCOUPLE (Термопара)
		0x02 - TS_TYPE_SEMICONDUCTOR (Полупроводниковый температурный датчик)
		0x08 - TS_AVAILABLE (Если флаг установлен, то датчик температуры доступен)
INT32U	LimitSwitchesSettings	Флаги настроек концевиков.
		0x01 - LS_ON_SW1_AVAILABLE (Если флаг установлен, то концевик, подключенный к ножке SW1, доступен)
		0x02 - LS_ON_SW2_AVAILABLE (Если флаг установлен, то концевик, подключенный к ножке SW2, доступен)
		0x04 - LS_SW1_ACTIVE_LOW (Если флаг установлен, то концевик, подключенный к ножке SW1, считается сработавшим по низкому уровню на контакте)
		0x08 - LS_SW2_ACTIVE_LOW (Если флаг установлен, то концевик, подключенный к ножке SW2, считается сработавшим по низкому уровню на контакте)
		0x10 - LS_SHORTED (Если флаг установлен, то концевики закорочены.)
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Запись информации о дополнительных аксессуарах в EEPROM. Функция должна использоваться только производителем.

Команда GACC

```
result_t get_accessories_settings (device_t id, accessories_settings_t* accessories_settings)
```

Код команды (CMD): "gacc" Или 0x63636167.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (114 байт)

INT32U	CMD	Команда (возврат)
CHAR	MagneticBrakeInfo [24]	Производитель и номер магнитного тормоза, Максимальная длина строки: 24 символов.
FLT32	MBRatedVoltage	Номинальное напряжение для управления магнитным тормозом (В). Тип данных: float.
FLT32	MBRatedCurrent	Номинальный ток для управления магнитным тормозом (А). Тип данных: float.
FLT32	MBTorque	Удерживающий момент (мН м). Тип данных: float.
INT32U	MBSettings	Флаги настроек магнитного тормоза.
		0x01 - MB_AVAILABLE (Если флаг установлен, то магнитный тормоз доступен)
		0x02 - MB_POWERED_HOLD (Если флаг установлен, то магнитный тормоз находится в режиме удержания (активен) при подаче питания)
CHAR	TemperatureSensorInfo [24]	Производитель и номер температурного датчика, Максимальная длина строки: 24 символов.
FLT32	TSMin	Минимальная измеряемая температура (град Цельсия). Тип данных: float.
FLT32	TSMax	Максимальная измеряемая температура (град Цельсия) Тип данных: float.
FLT32	TSGrad	Температурный градиент (В/град Цельсия). Тип данных: float.
INT32U	TSSettings	Флаги настроек температурного датчика.
		0x07 - TS_TYPE_BITS (Биты, отвечающие за тип температурного датчика.)
		0x00 - TS_TYPE_UNKNOWN (Неизвестный сенсор)
		0x01 - TS_TYPE_THERMOCOUPLE (Термопара)
		0x02 - TS_TYPE_SEMICONDUCTOR (Полупроводниковый температурный датчик)
		0x08 - TS_AVAILABLE (Если флаг установлен, то датчик температуры доступен)
INT32U	LimitSwitchesSettings	Флаги настроек концевиков.
		0x01 - LS_ON_SW1_AVAILABLE (Если флаг установлен, то концевик, подключенный к ножке SW1, доступен)
		0x02 - LS_ON_SW2_AVAILABLE (Если флаг установлен, то концевик, подключенный к ножке SW2, доступен)
		0x04 - LS_SW1_ACTIVE_LOW (Если флаг установлен, то концевик, подключенный к ножке SW1, считается сработавшим по низкому уровню на контакте)
		0x08 - LS_SW2_ACTIVE_LOW (Если флаг установлен, то концевик, подключенный к ножке SW2, считается сработавшим по низкому уровню на контакте)
		0x10 - LS_SHORTED (Если флаг установлен, то концевики закорочены.)
INT8U	Reserved [24]	Зарезервировано (24 байта)
INT16U	CRC	Контрольная сумма

Описание:
Чтение информации о дополнительных аксессуарах из EEPROM.

Команды загрузчика

Команда GBLV

```
result_t get_bootloader_version (device_t id, unsigned int* Major, unsigned int* Minor, unsigned int* Relea
se)
```

Код команды (CMD): "gblv" Или 0x766C6267.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (10 байт)

INT32U	CMD	Команда (возврат)
INT8U	Major	Мажорный номер версии загрузчика
INT8U	Minor	Минорный номер версии загрузчика
INT16U	Release	Номер релиза версии загрузчика
INT16U	CRC	Контрольная сумма

Описание:
Чтение номера версии прошивки контроллера.

Команда IRND

```
result_t get_init_random (device_t id, init_random_t* init_random)
```

Код команды (CMD): "irnd" Или 0x646E7269.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (24 байта)

INT32U	CMD	Команда (возврат)
INT8U	key [16]	Случайный ключ.
INT8U	Reserved [2]	Зарезервировано (2 байта)
INT16U	CRC	Контрольная сумма

Описание:
Чтение случайного числа из контроллера.

Команда GUID

```
result_t get_globally_unique_identifier (device_t id, globally_unique_identifier_t* globally_unique_identif
ier)
```

Код команды (CMD): "guid" Или 0x64697567.

Запрос: (4 байта)

INT32U	CMD	Команда

Ответ: (40 байт)

INT32U	CMD	Команда (возврат)
INT32U	UniqueID0	Уникальный ID 0.
INT32U	UniqueID1	Уникальный ID 1.
INT32U	UniqueID2	Уникальный ID 2.
INT32U	UniqueID3	Уникальный ID 3.
INT8U	Reserved [18]	Зарезервировано (18 байт)
INT16U	CRC	Контрольная сумма

Описание:
Считывает уникальный идентификатор каждого чипа, это значение не является случайным. Уникальный идентификатор может быть использован в качестве инициализационного вектора для операций шифрования бутлоадера или в качестве серийного номера для USB и других применений.

Команда CHMT

```
result_t command_change_motor (device_t id, const command_change_motor_t* the_command_change_motor)
```

Код команды (CMD): "chmt" Или 0x746D6863.

Запрос: (22 байта)

INT32U	CMD	Команда
INT8U	Motor	Номер мотора, на который следует переключить реле [0..1]
INT8U	Reserved [15]	Зарезервировано (15 байт)
INT16U	CRC	Контрольная сумма

Ответ: (4 байта)

INT32U	CMD	Команда (возврат)

Описание:
Сменить двигатель - команда для переключения выходного реле.

Процедура записи прошивки

Обмен данными с загрузчиком ведется по такому же протоколу, что и с прошивкой. Предварительно, контроллер должен быть переведён в режим Загрузчика, командой UPDF.

Для записи прошивки необходимо наличие ключа (записывается командой WKEY), по которому будет декодироваться принимаемая шифрованная прошивка.
Процедура записи прошивки состоит из следующих шагов:

1. Команда CONN
начало ISP сессии

2. Команда WDAT
Передает закодированные данные для записи во флеш память.

...

n-1. Команда WDAT

n. Команда DISC
конец ISP сессии.

В случае успешного перепрограммирования происходит переход из бутлоадера в прошивку.

Отрицательные ответы контроллера

Отрицательный ответ ERRC

Ответ: (4 байта)
Код: "errc" или 0x63727265

INT32U	"errc"	Команда недоступна

Описание:
Ответ на команду, в случае если команда неизвестна, либо не может быть выполнена и/или обработана в данный момент (в данном состоянии). Устанавливает соответствующий бит в поле "flags" структуры состояния.

Отрицательный ответ ERRD

Ответ: (4 байта)
Код: "errd" или 0x64727265

INT32U	"errd"	Неверные данные

Отрицательный ответ ERRV

Ответ: (4 байта)
Код: "errv" или 0x76727265

INT32U	"errv"	Неверное значение

Описание:
Ответ на команду, в случае если команда корректна, контрольная сумма правильная, но передаваемые значения (хотя бы одно из них) выходят за допустимый диапазон и не могут быть приняты. При этом неверное значение заменяется одним из верных методами округления, ограничения или сбрасывания в некое стандартное состояние. Устанавливает соответствующий бит в поле "flags" структуры состояния.

6.3. Совместимость с ПО для 8SMC1-USBhF

Новые контроллеры можно использовать с программами, написанными для контроллеров 8SMC1-USBhF. Для этого есть две возможности:
1. Рекомендуемая. Использование версии ПО MicroSMC, которая поддерживает как 8SMC1-USBhF так и новые контроллеры. В этом случае ПО MicroSMC открывает все найденные новые контроллеры в режиме эксклюзивного доступа для обеспечения разделения доступа и взаимодействие с ними возможно только с помощью функций API контроллеров 8SMC1-USBhF через библиотеку USMCDLL, использующую MicroSMC.
Скачать версию MicroSMC с поддержкой новых контроллеров можно на странице Программное обеспечение.
2. Управление только новыми контроллерами без ПО MicroSMC с помощью переходной библиотеки USMCDLL/libximc. Используйте эту возможность, если вам требуется одновременная работа ПО MicroSMC (для управления контроллерами 8SMC1-USBhF) и работа с контроллерами новыми с помощью библиотеки libximc.
Скачать переходную библиотеку USMCDLL/libximc можно по этой ссылке.

Ниже в таблице описано соответствие функций API контроллеров 8SMC1-USBhF и новых контроллеров: в первом столбце указана функция библиотеки USMCDLL, во втором - соответствующий параграф в описании контроллера 8SMC1-USBhF, в третьем - особенности этой функции в переходной библиотеке при использовании с новыми контроллерами.

Функция	**8SMC1-USBhF**	**Новый контроллер**
USMC_Init	7.5.3	Использует функцию libximc enumerate_devices, которая опрашивает все COM-порты в системе, а также функции get_enumerate_device_information, get_enumerate_device_serial, get_device_count, get_device_name, open_device.
USMC_GetState	7.5.4	Использует функции libximc get_status и get_engine_settings. Флаги AReset, EMReset, RotTrErr возвращаемой структуры USMC_State всегда принимают значение false.
USMC_SaveParametersToFlash	7.5.5	Использует функцию libximc command_save_settings.
USMC_GetMode	7.5.6	Использует функции libximc get_edges_settings, get_power_settings, get_control_settings, get_ctp_settings, get_sync_out_settings. В возвращаемой структуре USMC_Mode флаги EMReset, ResetRT, SyncOUTR, EncoderEn, EncoderInv, ResBEnc, ResEnc всегда принимают значение false, флаг SyncINOp всегда равен true.
USMC_SetMode	7.5.7	Использует те же функции, что и USMC_GetMode, а также их "set_" аналоги, игнорирует флаги RotTeEn, RotTrOp, ResetRT, SyncOUTR, SyncINOp, EncoderEn.
USMC_GetParameters	7.5.8	Использует функции libximc get_secure_settings, get_engine_settings, get_move_settings, get_feedback_settings, get_power_settings, get_control_settings, get_ctp_settings, get_home_settings, get_sync_out_settings. Возвращает нулевые значения параметров BTimeoutR, BTimeoutD, MinP, MaxLoft, StartPos.
USMC_SetParameters	7.5.9	Использует те же функции, что и USMC_GetParameters, а также их "set_" аналоги, игнорирует параметры BTimeoutR, BTimeoutD, MinP, MaxLoft, StartPos.
USMC_GetStartParameters	7.5.10	Использует функции libximc get_move_settings, get_engine_settings. Флаги WSyncIN, SyncOUTR, ForceLoft возвращаемой структуры USMC_StartParameters всегда принимают значение false.
USMC_Start	7.5.11	Использует функции libximc get_move_settings, get_engine_settings, а также их "set_" аналоги и функцию command_move.
USMC_Stop	7.5.12	Использует функцию libximc command_stop.
USMC_SetCurrentPosition	7.5.13	Использует функцию libximc set_position.
USMC_GetEncoderState	7.5.14	Использует функцию libximc get_status.
USMC_GetLastErr	7.5.15	Не модифицирует передаваемый ей параметр. Ошибку выполнения функций USMC_ можно отследить по ненулевому коду возврата.
USMC_Close	7.5.16	Использует функцию libximc close_device.

Переходная библиотека USMCDLL не требует для своей работы запущенного в фоне приложения MicroSMC.exe и использует библиотеку libximc.dll для взаимодействия с новыми контроллерами.
Все функции USMC_, включающие в себя несколько вызовов функций libximc, прерывают своё выполнение, если одна из составляющих их функций libximc возвращает ошибку. В этом случае возможна неполная запись настроек в контроллер. Ненулевой код возврата функции USMC_ равен коду возврата выполненной с ошибкой функции libximc.

Пример приложения: usmcdll_libximc_test.zip

6.4. Таймауты libximc

При работе с программой XiLab или написании собственных приложений с использованием libximc действуют таймауты для детектирования ошибок или более стабильной работы контроллера. Ниже приведён список таймаутов, их длительность и условия применения. Таймауты оптимизированы для работы через соединение USB на современном компьютере. При создании собственной цепи передачи управляющего сигнала необходимо учитывать задержки линии связи, чтобы таймауты не срабатывали.

Когда происходит	Название	Время в миллисекундах
Таймаут при перечислении устройств. Если не удаётся определить тип устройства.	ENUMERATE_TIMEOUT_TIME	100
Попытка открыть порт.	DEFAULT_TIMEOUT_TIME	5000
Ожидание данных от устройства.	DEFAULT_TIMEOUT_TIME	5000
От открытия устройства до начала работы с ним.	RESET_TIME/2	50
Ожидание появления устройства при запуске процедуры перезаливки и его перезагрузке.	RESET_TIME*1.2 + DEFAULT_TIMEOUT_TIME	5120
Ожидание после записи сектора флэш памяти при перезаливке.	FLASH_SECTIONWRITE_TIME	100
Таймаут попыток установить связь с контроллером после его перезагрузки для перезаливки.	XISM_PORT_DETECT_TIME	60000

6.5. Скрипты XILab

6.5. Скрипты XILab
Краткое описание языка
Типы данных
Инструкции
Объявление переменных
Ключевые и зарезервированные слова
Функции
Подсветка синтаксиса
Дополнительные функции, предоставляемые XILab
Запись в лог XILab
Задержка выполнения скрипта
Создание объекта типа "ось"
Создание объекта типа "файл"
Создание структуры калибровки
Получение следующего серийного номера
Ожидание остановки движения
Функции библиотеки libximc
Примеры
Скрипт циклического движения
Скрипт сканирования и записи в файл
Скрипт движения по списку позиций из csv файла
Скрипт, который строит список осей с серийными номерами и позицией каждой оси
Скрипт-пример работы с битовыми масками

Скриптовый язык Xilab реализован с помощью QtScript, он в свою очередь основан на ECMAScript. ECMAScript — это встраиваемый расширяемый не имеющий средств ввода/вывода язык программирования, используемый в качестве основы для построения других скриптовых языков. Стандартизирован международной организацией ECMA в спецификации ECMA-262.
Используется третья редакция стандарта.

Краткое описание языка

Типы данных

В ECMAScript поддерживаются девять типов данных. Три из них (Reference, List, и Completion) используются только как промежуточные результаты расчета значений выражений. Оставшиеся шесть типов это:

- неопределённый (англ. Undefined)
- нулевой (англ. Null),
- логический (англ. Boolean),
- строковый (англ. String),
- числовой (англ. Number),
- объектный (англ. Object).

Инструкции

Наиболее распространенные конструкции языка ECMAScript представлены в таблице ниже:

Название	Применение	Краткие сведения
Блок	{[<список инструкций>]}	Несколько инструкций можно объединить в один блок фигурными скобками.
Объявление переменной	var <список объявления переменных>	Переменная объявляется с помощью ключевого слова "var".
Пустая инструкция	;	Точка с запятой является пустой инструкцией. Заканчивать строки точкой с запятой не обязательно.
Условие	if (<условие>) <инструкция> [else <инструкция>]	Условное выполнение можно производить с помощью ключевых слов "if ... else". Если условие верно, то выполняется инструкция блока if, в противном случае выполняется инструкция блока else, если он присутствует.

Цикл	do <тело цикла> while (<условие>)
	while (<условие>) <тело цикла>
	for ([<выражение 1>]; [<условие>]; [<выражение 2>]) <тело цикла>
		Цикл можно реализовать несколькими различными способами. Форма "do ... while ..." выполняет тело цикла как минимум один раз и пока условие верно. Форма "while ... do ..." выполняет тело цикла пока условие верно. Форма "for ..." выполняет выражение до начала (выражение 1), а затем выполняет тело цикла каждый раз после выполнения итеративного выражения (выражение 2) и проверки условия на истинность.

Возврат	return [<выражение>]	Прекращает выполнение функции и возвращает выражение как результат.
Генерация исключения	throw <выражение>	Генерирует исключение, которое может быть обработано конструкцией try (см.ниже).
Блок try	try <блок> catch (<идентификатор>) <блок> try <блок> finally <блок> try <блок> catch (<идентификатор>) <блок> finally <блок>	Используется совместно с исключениями. Конструкция "try ... catch ... finally" пытается выполнить блок try. Если в этом блоке происходит исключение идентификатор, то выполняется содержимое блока catch. После всего безусловно выполняется блок finally. Один из блоков "catch" и "finally" может отсутствовать.

Объявление переменных

Переменные определяются с помощью ключевого слова var. При объявлении переменная помещается в область видимости, соответствующую функции, в которой она объявляется. Если переменная объявляется вне функций, она помещается в глобальную область видимости. Создание переменной происходит при получении управления функцией с её объявлением. Или программой, если переменная глобальна. При создании переменной в ECMAScript она приобретает значение undefined. Если переменная объявлена с инициализацией, инициализация происходит не в момент создания переменной, а при выполнении строки с инструкцией var.

Ключевые и зарезервированные слова

Следующие слова являются ключевыми в языке и не могут быть использованы как идентификаторы:

```
break     else        new     var
case      finally     return  void
catch     for         switch  while
continue  function    this    with
default   if          throw
delete    in          try
do        instanceof  typeof
```

Следующие слова используются как ключевые в предлагаемых расширениях и зарезервированы для возможного использования в будущем:

```
abstract  enum        int         short
boolean   export      interface   static
byte      extends     long        super
char      final       native      synchronized
class     float       package     throws
const     goto        private     transient
debugger  implements  protected   volatile
double    import      public
```

Функции

Функции в ECMAScript являются объектами. Функции, как и любые другие объекты, могут храниться в переменных, объектах и массивах, могут передаваться как аргументы в другие функции и могут возвращаться функциями. Функции, как и любые другие объекты, могут иметь свойства. Существенной специфической чертой функций является то, что они могут быть вызваны.

В тексте программы именованную функцию в ECMAScript обычно определяют следующим способом:

```
function sum(arg1, arg2) { // функция принимает два параметра
    return arg1 + arg2;    // и возвращает их сумму
}
```

Подсветка синтаксиса

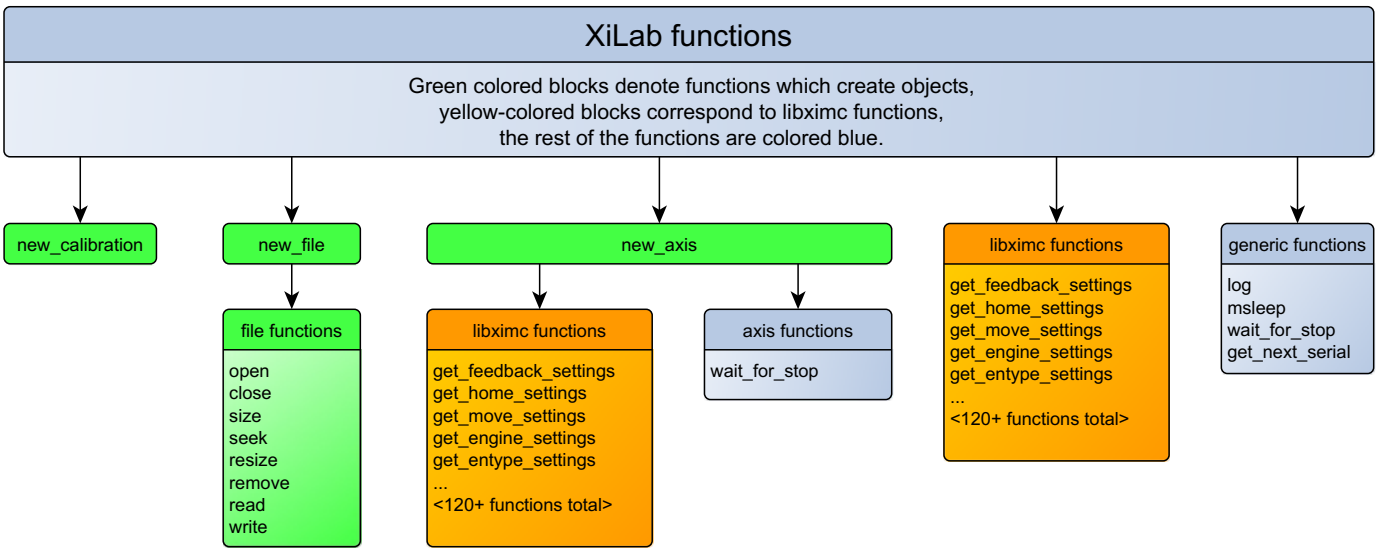
Шрифт текста в окне скрипта имеет подсветку синтаксиса. Цвета:

Тип выражения	*цвет*	*пример отображения*
произвольные функции	фиолетовый	my_function();
функции XILab	синий	get_status();
положительные числа	зеленый	a = 100;
отрицательные числа	красный	b = -200;
комментарии	серый	// a comment
все остальное	черный	var s = "a string";

Во время выполнения скрипта фон строки с последней выполненной командой меняется на тёмно-серый с частотой обновления 1 раз в 20 мс.

Дополнительные функции, предоставляемые XILab

На данной картинке изображены функции, которые XILab предоставляет для использования в скриптах в дополнение к стандартным функциям языка. Зеленым цветом отмечены функции, создающие объект, желтым - функции, вызывающие библиотеку libximc, синим - все остальные функции.



- log(string text [, int loglevel]) - запись в лог XILab
- msleep(int ms) - задержка выполнения скрипта
- new_axis(int serial_number) - создание объекта типа "ось"
- new_file(string filename) - создание объекта типа "файл"
- new_calibration(int A, int Microstep) - создание структуры калибровки для передачи калиброванным функциям
- get_next_serial(int serial) - получение следующего серийного номера

- command_wait_for_stop(int refresh_period) - ожидание остановки движения
- а также все функции библиотеки libximc (см. Руководство по программированию)

Кроме этого, в скриптах определены и доступны для использования все константы протокола обмена. Пример использования.

Запись в лог XILab

Производится вызовом функции log(string text [, int loglevel]).
Дописывает в лог XiLab строку text. Если передаётся второй параметр loglevel, то сообщение получает соответствующий уровень логгирования и отображается соответствующим цветом.

Loglevel	Тип
1	Error
2	Warning
3	Info

Пример:

```
var x = 5;
log("x = " + x);
```

Пример использования в коде скрипта.

Замечание: не рекомендуется вызывать функции интерфейса Xilab (запись в лог) чаще одного раза в 20 мс.

Задержка выполнения скрипта

Производится вызовом функции msleep(int ms).
Скрипт делает паузу в данном месте выполнения длиной ms миллисекунд.
Пример:

```
msleep(200);
```

Пример использования в коде скрипта.

Создание объекта типа "ось"

Многоосевой интерфейс XILab также предоставляет возможность управлять контроллерами посредством скриптов. Отличия состоят в том, что необходимо явно указывать, какому контроллеру посылается команда. Для этого вводится новый тип объекта "ось", который имеет методы, совпадающие по именам с функциями библиотеки. Идентификация происходит по серийному номеру контроллера.
Пример:

```
var x = new_axis(123);
x.command_move(50);
```

В этом примере в первой строке скрипта происходит создание оси с именем переменной x, которая соответствует контроллеру с серийным номером "123". Если такой контроллер не подключен к компьютеру, то скрипт выдаст ошибку выполнения и завершится. Во второй строке оси x подается команда переместиться в координату 50 [шагов].

Пример использования в коде скрипта.

Создание объекта типа "файл"

Скрипты Xilab имеют возможность чтения из файла и записи в файл. Для этого необходимо создать объект типа файл и работать с ним. Имя файла указывается при создании в конструкторе. Объект имеет следующие функции:

Тип_возврата Имя_функции	Описание
bool open()	Открывает файл. Файл открывается на чтение-запись, если это возможно; если нет, то только на чтение.
void close()	Закрывает файл.
Number size()	Возвращает размер файла в байтах.
bool seek(Number pos)	Устанавливает текущую позицию в файле в *pos* байт[1].
bool resize(Number size)	Изменяет размер файла до *size* байт. Если *size* меньше текущего размера, то файл обрезается, если больше, то дополняется нулями.
bool remove()	Удаляет файл.
String read(Number maxsize)	Читает строку из файла, но не более *maxsize* байт. Данные читаются в кодировке utf-8.
Number write(String s, Number maxsize)	Записывает строку в файл, но не более *maxsize* байт. Данные записываются в кодировке utf-8, символ конца строки пользователь должен записать самостоятельно. Возвращает количество записанных байт, либо -1, если произошла ошибка.

Все функции работы с файлами, возвращающие значение типа *bool*, возвращают true в случае успеха, в противном случае false. Используйте символ "/" как разделитель путей для работы скриптов на всех платформах (Windows/Linux/Mac).

[1] Использование функции *seek*: если позиция больше размера файла, то вызов функции не приведет к немедленному увеличению размера файла. Если в этой позиции произойдет запись, то размер файла будет увеличен. Содержимое файла между предыдущим концом файла и новыми записанными данными в этом случае НЕ ОПРЕДЕЛЕНО и может принимать различные значения на разных платформах и типах файловых систем.

Пример:

```
var winf = new_file("C:/file.txt"); // An example of file name and path on Windows
var linf = new_file("/home/user/Desktop/file.txt"); // An example of file name and path on Linux
var macf = new_file("/Users/macuser/file.txt"); // An example of file name and path on Mac

var f = winf; // Pick a file name
if (f.open()) { // Try to open the file
  f.write( "some text" ); // If successful, then write desired data to the file
  f.close(); // Close the file
} else { // If file open failed for some reason
  log( "Failed opening file" ); // Log an error
}
```

Пример использования в коде скрипта.

Создание структуры калибровки

Функция new_calibration(double A, int Microstep) принимает в качестве параметров коэффициент A пересчета из шагов в пользовательские единицы и деление микрошага Microstep (либо полученное ранее вызовом функции get_engine_settings в поле MicrostepMode, либо задаваемое одной из констант MICROSTEP_MODE_) и возвращает структуру типа calibration_t, которую необходимо передать в калиброванные get_/set_ функции для получения/задания величин в пользовательских единицах. Следующие две записи функционально эквивалентны:

```
// create calibration: type 1
var calb = new_calibration(c1, c2);
```

```
// create calibration: type 2
var calb = new Object();
calb.A = c1;
calb.MicrostepMode = c2;
```

Пример использования в коде скрипта.

Получение следующего серийного номера

Функция get_next_serial(int serial) принимает в качестве параметра число и возвращает наименьший серийный номер из списка серийных номеров открытых устройств, который больше переданного ей параметра. Если такого серийного номера нет, то возвращается 0.
Эта функция удобна для автоматического создания объекта типа "ось" без задания фиксированного серийного номера.
Пример:

```
var first_serial = get_next_serial(0);
var x = new_axis(first_serial);
var y = new_axis(get_next_serial(first_serial));
```

В этом примере в первой строке происходит получение первого серийного номера, во второй - создание объекта оси по этому серийному номеру, в третьей - получение следующего серийного номера и создание оси для него.

Пример использования в коде скрипта.

Ожидание остановки движения

Функция command_wait_for_stop(int refresh_period) останавливает выполнение скрипта до тех пор пока контроллер не прекратит движение, то есть, пока флаг MVCMD_RUNNING в члене MvCmdSts структуры, возвращаемой функцией get_status(), не будет снят. Функция скриптов command_wait_for_stop напрямую использует функцию command_wait_for_stop библиотеки libximc и принимает в качестве параметра число, означающее периодичность (в миллисекундах) считывания состояния контроллера.
Эта функция также присутствует как метод объекта типа "ось".

Пример использования в коде скрипта.

Функции библиотеки libximc

Функции библиотеки libximc начинающиеся на "get_" считывают из контроллера настройки и возвращают соотвествующую команде структуру данных. Функции библиотеки libximc начинающиеся на "set_" принимают как параметр структуру данных и записывают эти настройки в контроллер. Заполнить структуру данных для set-функций можно двумя способами:

1. вызвать соответствующую get-функцию и модифицировать необходимые поля

```
// set settings: type 1
var m = get_move_settings();
m.Speed = 100;
set_move_settings(m);
```

2. создать объект (Object) и заполнить все его свойства (properties) с именами членов структуры с учетом регистра.

```
// set settings: type 2
var m = new Object;
m.Speed = 100;
m.uSpeed = 0;
m.Accel = 300;
m.Decel = 500;
m.AntiplaySpeed = 10;
m.uAntiplaySpeed = 0;
set_move_settings(m);
```

Необходимо помнить, что при использовании первого способа в контроллер посылается дополнительная команда (вызывается не только set-функция, но и get- перед этим), а при использовании второго способа необходимо инициализировать все свойства объекта, соответствующие именам членов структуры. Если в объекте какое-то свойство будет пропущено, то оно будет считаться равным нулю. Если в объекте будет объявлено какое-либо дополнительное свойство, не входящее в структуру данных, то оно будет проигнорировано. Если тип данных какого-либо поля не будет соответствовать типу данных структуры, то будет произведено приведение типов по правилам EcmaScript. Структуры данных для всех команд описаны в главе Описание протокола обмена.

Пример использования в коде скрипта.

Примеры

В этом разделе приведены примеры типичных действий, которые можно выполнять с помощью скриптов XILab.

Скрипт циклического движения

```
var first_border = -10; // first border coordinate in mm
var second_border = 10; // second border coordinate in mm
var mm_per_step = 0.005; // steps to distance translation coefficient
var delay = 100; // delay in milliseconds
var calb = new_calibration(mm_per_step, get_engine_settings().MicrostepMode); // create calibration structu
re
command_stop(); // send STOP command (does immediate stop)
command_zero(); // send ZERO command (sets current position and encoder value to zero)
while (1) { // infinite loop
  command_move_calb(first_border, calb); // move towards one border
  command_wait_for_stop(delay); // wait until controller stops moving
  command_move_calb(second_border, calb); // move towards another border
  command_wait_for_stop(delay); // wait until controller stops moving
}
```

Скрипт сканирования и записи в файл

```
var start = 0; // Starting coordinate in steps
var step = 10; // Shift amount in steps
var end = 100; // Ending coordinate in steps

var speed = 300; // maximum movement speed in steps / second
var accel = 100; // acceleration value in steps / second^2
var decel = 100; // deceleration value in steps / second^2
var delay = 100;

var m = get_move_settings(); // read movement settings from the controller
m.Speed = speed; // set movement speed
m.Accel = accel; // set acceleration
m.Decel = decel; // set deceleration
set_move_settings(m); // write movement settings into the controller

var f = new_file("C:/a.csv"); // Choose a file name and path
f.open(); // Open a file
f.seek( 0 ); // Seek to the beginning of the file

command_move(start); // Move to the starting position
command_wait_for_stop(delay); // Wait until controller stops moving

while (get_status().CurPosition < end) {
  f.write( get_status().CurPosition + "," + get_chart_data().Pot + "," + Date.now() + "\n" ); // Get curren
t position, potentiometer value and date and write them to file
  command_movr(step); // Move to the next position
  command_wait_for_stop(delay); // Wait until controller stops moving
}
f.close(); // Close the file
```

Скрипт движения по списку позиций из csv файла

```
var axis = new_axis(get_next_serial(0)); // Use first available controller
var x; // A helper variable, represents coordinate
var ms; // A helper variable, represents wait time in milliseconds
var f = new_file("./move_and_sleep.csv"); // Choose a file name and path; this script uses a file from exam
ples in the installation directory
f.open(); // Open a file
while ( str = f.read(4096) ) { // Read file contents string by string, assuming each string is less than 4
KiB long
  var ar = str.split(","); // Split the string into substrings with comma as a separator; the result is an
array of strings
  x = ar[0]; // Variable assignment
  ms = ar[1]; // Variable assignment
  log( "Moving to coordinate " + x ); // Log the event
  axis.command_move(x); // Move to the position
  axis.command_wait_for_stop(100); // Wait until the movement is complete
  log( "Waiting for " + ms + " ms" ); // Log the event
  msleep(ms); // Wait for the specified amount of time
}
log ( "The end." );
f.close(); // Close the file
```

move_and_sleep.csv - пример файла для использования с приведенным выше скриптом.

Скрипт, который строит список осей с серийными номерами и позицией каждой оси

```
var i = 0; // Declare loop iteration variable
var serial = 0; // Declare serial number variable
var axes = Array(); // Declare axes array
while (true) { // The loop
  serial = get_next_serial(serial); // Get next serial
  if (serial == 0) // If there are no more controllers then...
      break; // ...break out of the loop
  var a = new Object(); // Create an object
  a.serial = serial; // Assign serial number to its "serial" property
  a.handle = new_axis(serial); // Assign new axis object to its "handle" property
  axes[i] = a; // Add it to the array
  i++; // Increment counter
}
for (var k=0; k < axes.length; k++) { // Iterate through array elements
  log ( "Axis with S/N " + axes[k].serial + " is in position " + axes[k].handle.get_status().CurPosition );
 // For each element print saved axis serial and call a get_status() function
}
```

Скрипт-пример работы с битовыми масками

```
/*
  Bitmask example script
*/
var a = new_axis(get_next_serial(0)); // take first found axis
var gets = a.get_status(); // read status once and reuse it

var gpio = gets.GPIOFlags;
var left = STATE_LEFT_EDGE;
var right = STATE_RIGHT_EDGE;
var mask = left | right;
var result = gpio & mask;
log( to_binary(left) + " = left limit switch flag" );
log( to_binary(right) + " = right limit switch flag" );
log( to_binary(mask) + " = OR operation on flags gives the mask" );
log( to_binary(gpio) + " = gpio state" );
log( to_binary(result) + " = AND operation on state and mask gives result" );
if ( result ) {
  log("At least one limit switch is on");
} else {
  log("Both limit switches are off");
}

// Binary representation function
function to_binary(i)
{
  bits = 32;
  x = i >>> 0; // coerce to unsigned in case we need to print negative ints
  str = x.toString(2); // the binary representation string
  return (repeat("0", bits) + str).slice (-bits); // pad with zeroes and return
}

// String repeat function
function repeat(str, times)
{
  var result="";
  var pattern=str;
  while (times > 0) {
    if (times&1) {
      result+=pattern;
    }
    times>>=1;
    pattern+=pattern;
  }
  return result;
}
```

7. Файлы

1. Конфигурационные файлы
2. Программное обеспечение

7.1. Конфигурационные файлы

1. Линейные трансляторы
2. Ротаторы
3. Вертикальные моторизованные трансляторы
4. Моторизованные винты
5. Моторизованные гониометры
6. Моторизованные держатели зеркал
7. Моторизованные аттенуаторы
8. Моторизованные диафрагмы

7.1.1. Линейные трансляторы


8MT160 - моторизированная линия задержки


8MT295 - моторизованный линейный транслятор большого диапазона


8MT195 - моторизованный линейный транслятор большого диапазона


8MT167 - моторизованный линейный транслятор


8MT173 - компактный моторизованный линейный транслятор


8MT173DC - компактный моторизованный линейный транслятор


8MT50 - моторизованный линейный транслятор


8MT30 - узкий моторизованный линейный транслятор


8MT175 - моторизованный линейный транслятор


8MT177 - моторизованный линейный транслятор


8MT184 - моторизованный линейный транслятор


8MT193 - моторизованный линейный транслятор



8MT200 - моторизованный линейный транслятор



8MTF - двухосный моторизованный линейный транслятор



8MTF2 - двухосный моторизованный линейный транслятор



8MTFV - моторизованный линейный транслятор



8MT60 - моторизованный линейный транслятор

8MT160 - моторизированная линия задержки

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MT160-300	300	mm	8MT160-300.cfg	8MT160	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT160-300-MEn1	300	mm	8MT160-300-MEn1.cfg	8MT160	4247R	HEDM5500R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8MT295 - моторизованный линейный транслятор большого диапазона

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MT295X-240-2.5-DC	240	mm	8MT295X-240-2.5-DC.cfg	8MT295-2.5	Maxon-370355	HEDL5540	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT295X-340-2.5	340	mm	8MT295X-340-2.5.cfg	8MT295-2.5	5918-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT295X-540-4	540	mm	8MT295X-540-4.cfg	8MT295-4	5918-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT295X-740-5	740	mm	8MT295X-740-5.cfg	8MT295-5-740	5918-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT295X-840-10	840	mm	8MT295X-840-10.cfg	8MT295-10	5918-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT295X-1040-5	1040	mm	8MT295X-1040-5.cfg	8MT295-5-1040	5918-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT295Z-340-2.5	340	mm	8MT295Z-340-2.5.cfg	8MT295-2.5	5918-R	NoEnc	BRAKE-AB41	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT295Z-540-4	540	mm	8MT295Z-540-4.cfg	8MT295-4	5918-R	NoEnc	BRAKE-AB41	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT295Z-740-5	740	mm	8MT295Z-740-5.cfg	8MT295-5-740	5918-R	NoEnc	BRAKE-AB41	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT295Z-840-10	840	mm	8MT295Z-840-10.cfg	8MT295-10	5918-R	NoEnc	BRAKE-AB41	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8MT195 - моторизованный линейный транслятор большого диапазона

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MT195X-340-2.5	340	mm	8MT195X-340-2.5.cfg	8MT195-2.5	5918-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT195X-540-4	540	mm	8MT195X-540-4.cfg	8MT195-4	5918-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT195X-540-10	540	mm	8MT195X-540-10.cfg	8MT195-10	5918-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT195X-740-5	740	mm	8MT195X-740-5.cfg	8MT195-5-740	5918-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT195X-840-10	840	mm	8MT195X-840-10.cfg	8MT195-10	5918-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT195X-1040-10	1040	mm	8MT195X-1040-10.cfg	8MT195-10	5918-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT195Z-240-2.5-DC	240	mm	8MT195Z-240-2.5-DC.cfg	8MT195-2.5	Maxon-370355	HEDL9140	BRAKE-AB41	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT195Z-240-2.5	240	mm	8MT195Z-240-2.5.cfg	8MT195-2.5	5918-R	NoEnc	BRAKE-AB41	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT195Z-340-2.5	340	mm	8MT195Z-340-2.5.cfg	8MT195-2.5	5918-R	NoEnc	BRAKE-AB41	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT195Z-540-4	540	mm	8MT195Z-540-4.cfg	8MT195-4	5918-R	NoEnc	BRAKE-AB41	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT195Z-740-5	740	mm	8MT195Z-740-5.cfg	8MT195-5-740	5918-R	NoEnc	BRAKE-AB41	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT195Z-840-10	840	mm	8MT195Z-840-10.cfg	8MT195-10	5918-R	NoEnc	BRAKE-AB41	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8MT167 - моторизованный линейный транслятор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MT167-100	100	mm	8MT167-100.cfg	8MT167	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT167-100-28	100	mm	8MT167-100-28.cfg	8MT167	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT167-100C28	100	mm	8MT167-100C28.cfg	8MT167	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT167S-100	100	mm	8MT167S-100.cfg	8MT167	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT167S-100-28	100	mm	8MT167S-100-28.cfg	8MT167	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT167S-100C28	100	mm	8MT167S-100C28.cfg	8MT167	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT167-25BS1	25	mm	8MT167-25BS1.cfg	8MT167BS	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT167-25BS1-MEn1	25	mm	8MT167-25BS1-MEn1.cfg	8MT167BS	28R	HEDM5500R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT167-25LS	25	mm	8MT167-25LS.cfg	8MT167LS	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT167-25LS-MEn1	25	mm	8MT167-25LS-MEn1.cfg	8MT167LS	28R	HEDM5500R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT167M-25BS1	25	mm	8MT167M-25BS1.cfg	8MT167BS	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT167M-25LS	25	mm	8MT167M-25LS.cfg	8MT167LS	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT167S-25BS1	25	mm	8MT167S-25BS1.cfg	8MT167BS	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT167S-25LS	25	mm	8MT167S-25LS.cfg	8MT167LS	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT167S-25LS-MEn1	25	mm	8MT167S-25LS-MEn1.cfg	8MT167LS	28R	HEDM5500R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT167SV-100-VSS42	100	mm	8MT167SV-100-VSS42.cfg	8MT167	VSS42-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT167V-100-VSS42	100	mm	8MT167V-100-VSS42.cfg	8MT167	VSS42-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT167MV-25LS-VSS42	25	mm	8MT167MV-25LS-VSS42.cfg	8MT167	VSS42-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT167SV-25LS-VSS42	25	mm	8MT167SV-25LS-VSS42.cfg	8MT167	VSS42-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT167V-25LS-VSS42	25	mm	8MT167V-25LS-VSS42.cfg	8MT167	VSS42-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8MT173 - компактный моторизованный линейный транслятор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MT173-10	10	mm	8MT173-10.cfg	8MT173	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT173-10-MEn1	10	mm	8MT173-10-MEn1.cfg	8MT173	28R	HEDM5500R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT173-20	20	mm	8MT173-20.cfg	8MT173	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT173-20-E3	20	mm	8MT173-20-E3.cfg	8MT173	28R	RevS	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT173D-20-E3	20	mm	8MT173D-20-E3.cfg	8MT173	28R	RevS	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT173D2-20-E3	20	mm	8MT173D2-20-E3.cfg	8MT173	28R	RevS	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT173-20-50-E3	20	mm	8MT173-20-50-E3.cfg	8MT173	28R	RevS	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT173-20-1-28S-E3	20	mm	8MT173-20-1-28S-E3.cfg	8MT173	28S	RevS	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT173-20-E4	20	mm	8MT173-20-E4.cfg	8MT173	28R	RevS	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT173-20-MEn1	20	mm	8MT173-20-MEn1.cfg	8MT173	28R	HEDM5500R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT173-25	25	mm	8MT173-25.cfg	8MT173	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT173-25-MEn1	25	mm	8MT173-25-MEn1.cfg	8MT173	28R	HEDM5500R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT173-30	30	mm	8MT173-30.cfg	8MT173	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT173-30-MEn1	30	mm	8MT173-30-MEn1.cfg	8MT173	28R	HEDM5500R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT173V-10-VSS42	10	mm	8MT173V-10-VSS42.cfg	8MT173V	VSS42-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT173V-20-VSS42	20	mm	8MT173V-20-VSS42.cfg	8MT173V	VSS42-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT173V-25-VSS42	25	mm	8MT173V-25-VSS42.cfg	8MT173V	VSS42-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT173V-30-VSS42	30	mm	8MT173V-30-VSS42.cfg	8MT173V	VSS42-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8MT173DC - компактный моторизованный линейный транслятор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MT173-10DCE2	10	mm	8MT173-10DCE2.cfg	8MT173-DCE2	Maxon-118512-R	EncMRtypeS-R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefDC
8MT173-20DCE2	20	mm	8MT173-20DCE2.cfg	8MT173-DCE2	Maxon-118512-R	EncMRtypeS-R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefDC
8MT173-25DCE2	25	mm	8MT173-25DCE2.cfg	8MT173-DCE2	Maxon-118512-R	EncMRtypeS-R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefDC
8MT173-30DCE2	30	mm	8MT173-30DCE2.cfg	8MT173-DCE2	Maxon-118512-R	EncMRtypeS-R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefDC

8MT50 - моторизованный линейный транслятор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MT50-100BS1	100	mm	8MT50-100BS1.cfg	8MT50	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT50-100BS1-MEn1	100	mm	8MT50-100BS1-MEn1.cfg	8MT50	4247R	HEDM5500R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT50-100XY	100	mm	8MT50-100XY.cfg	8MT50	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT50-100XYZ	100	mm	8MT50-100XYZ.cfg	8MT50	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT50-150BS1	150	mm	8MT50-150BS1.cfg	8MT50	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT50-150BS1-MEn1	150	mm	8MT50-150BS1-MEn1.cfg	8MT50	4247R	HEDM5500R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT50-150XY	150	mm	8MT50-150XY.cfg	8MT50	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT50-150XYZ	150	mm	8MT50-150XYZ.cfg	8MT50	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT50-200BS1	200	mm	8MT50-200BS1.cfg	8MT50	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT50-200BS1-MEn1	200	mm	8MT50-200BS1-MEn1.cfg	8MT50	4247R	HEDM5500R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT50-200XY	200	mm	8MT50-200XY.cfg	8MT50	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT50-200XYZ	200	mm	8MT50-200XYZ.cfg	8MT50	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT50Z-100BS1	100	mm	8MT50Z-100BS1.cfg	8MT50	ST4118-K2V17690	NoEnc	BRAKE-BKE-04-05	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT50Z-150BS1	150	mm	8MT50Z-150BS1.cfg	8MT50	ST4118-K2V17690	NoEnc	BRAKE-BKE-04-05	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT50Z-200BS1	200	mm	8MT50Z-200BS1.cfg	8MT50	ST4118-K2V17690	NoEnc	BRAKE-BKE-04-05	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8MT30 - узкий моторизованный линейный транслятор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MT30-50	50	mm	8MT30-50.cfg	8MT30	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT30-50-MEn1	50	mm	8MT30-50-MEn1.cfg	8MT30	28R	HEDM5500R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT30-50DCE	50	mm	8MT30-50DCE.cfg	8MT30DC	Maxon-339152-R	EncMRtypeML-R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefDC
8MT30V-50-VSS42	50	mm	8MT30V-50-VSS42.cfg	8MT30V	VSS42-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT30V-50DCE	50	mm	8MT30V-50DCE.cfg	8MT30VDC	FAULHABER-1524T006SR	IE2-16	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefDC

8MT175 - моторизованный линейный транслятор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MT175-100	100	mm	8MT175-100.cfg	8MT175	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT175-100-E3	100	mm	8MT175-100-E3.cfg	8MT175	4247R	RevS	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT175-100-MEn1	100	mm	8MT175-100-MEn1.cfg	8MT175	4247R	HEDM5500R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT175-150	150	mm	8MT175-150.cfg	8MT175	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT175-150-MEn1	150	mm	8MT175-150-MEn1.cfg	8MT175	4247R	HEDM5500R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT175-200	200	mm	8MT175-200.cfg	8MT175	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT175-200-MEn1	200	mm	8MT175-200-MEn1.cfg	8MT175	4247R	HEDM5500R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT175-50	50	mm	8MT175-50.cfg	8MT175	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT175-50-MEn1	50	mm	8MT175-50-MEn1.cfg	8MT175	4247R	HEDM5500R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT175V-100-VSS42	100	mm	8MT175V-100-VSS42.cfg	8MT175V	VSS42-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT175V-150-VSS42	150	mm	8MT175V-150-VSS42.cfg	8MT175V	VSS42-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT175V-200-VSS42	200	mm	8MT175V-200-VSS42.cfg	8MT175V	VSS42-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT175V-50-VSS42	50	mm	8MT175V-50-VSS42.cfg	8MT175V	VSS42-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8MT177 - моторизованный линейный транслятор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MT177-100	100	mm	8MT177-100.cfg	8MT177	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT177-100-E4	100	mm	8MT177-100-E4.cfg	8MT177	4247R	RevS	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT177-100XY	100	mm	8MT177-100XY.cfg	8MT177	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT177-100XYZ	100	mm	8MT177-100XYZ.cfg	8MT177	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT177-100-28	100	mm	8MT177-100-28.cfg	8MT177	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT177-100-28XY	100	mm	8MT177-100-28XY.cfg	8MT177	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT177-100-28XYZ	100	mm	8MT177-100-28XYZ.cfg	8MT177	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8MT184 - моторизованный линейный транслятор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MT184-13	13	mm	8MT184-13.cfg	8MT184	20-Rev	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT184-13DC	13	mm	8MT184-13DC.cfg	8MT184DC	FAULHABER-1524T006SR	IE2-256	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefDC
8MT184-13XY	13	mm	8MT184-13XY.cfg	8MT184	20-Rev	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT184-13XYZ	13	mm	8MT184-13XYZ.cfg	8MT184	20-Rev	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT184V-13DC	13	mm	8MT184V-13DC.cfg	8MT184DCV	FAULHABER-1524T006SR	IE2-16	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefDC

8MT193 - моторизованный линейный транслятор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MT193-100	100	mm	8MT193-100.cfg	8MT193	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT193-100-E4-DS	100	mm	8MT193-100-E4-DS.cfg	8MT193	4247R	RevS	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8MT200 - моторизованный линейный транслятор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MT200-100	100	mm	8MT200-100.cfg	8MT200	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8MTF - двухосный моторизованный линейный транслятор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MTF-102LS05	102	mm	8MTF-102LS05.cfg	8MTF-102	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MTF-75LS05	75	mm	8MTF-75LS05.cfg	8MTF-75	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MTF-75LS1	75	mm	8MTF-75LS1.cfg	8MTF-75-2	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8MTF2 - двухосный моторизованный линейный транслятор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MTF2	2	mm	8MTF2.cfg	8MTF2	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8MTFV - моторизованный линейный транслятор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MTFV-75_40LS05-42.3	40	mm	8MTFV-75_40LS05-42.3.cfg	8MTFV	D423	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8MT60 - моторизованный линейный транслятор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MT60-200	200	mm	8MT60-200.cfg	8MT60	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MT60V-200	200	mm	8MT60V-200.cfg	8MT60V	VSS42-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

7.1.2. Ротаторы


8MR151 - моторизованный ротатор


8MR170 - моторизованный ротатор


8MR174 - моторизованный ротатор


8MR190 - моторизованный ротатор


8MR191 - моторизованный ротатор


8MRB250 - большой моторизованный ротатор


8MPR16-1 - моторизованный вращатель поляризатора


8MRU - универсальный моторизованный ротатор


8MRH240 - большой моторизованный ротатор

8MR151 - моторизованный ротатор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MR151-1	360	deg	8MR151-1.cfg	8MR151	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR151-1-MEn1	360	deg	8MR151-1-MEn1.cfg	8MR151	28	HEDM5500	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR151-30	360	deg	8MR151-30.cfg	8MR151	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR151-30-E4	360	deg	8MR151-30-E4.cfg	8MR151	28	RevS	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR151-30-MEn1	360	deg	8MR151-30-MEn1.cfg	8MR151	28	HEDM5500	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR151E-1	360	deg	8MR151E-1.cfg	8MR151	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR151E-30	360	deg	8MR151E-30.cfg	8MR151	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR151-30-E3	360	deg	8MR151-30-E3.cfg	8MR151	28	RevS	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot

8MR170 - моторизованный ротатор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MR170-190	6	deg	8MR170-190.cfg	8MR170	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot

8MR174 - моторизованный ротатор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MR174-11-20	360	deg	8MR174-11-20.cfg	8MR174	20	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR174-11-28	360	deg	8MR174-11-28.cfg	8MR174	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR174-11-28-E4	360	deg	8MR174-11-28-E4.cfg	8MR174	28	RevS	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR174-11-28S	360	deg	8MR174-11-28S.cfg	8MR174	28S	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR174-11-28-E3	360	deg	8MR174-11-28-E3.cfg	8MR174	28	RevS	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR174-11-28S-E3	360	deg	8MR174-11-28S-E3.cfg	8MR174	28S	RevS	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR174-11-28-MEn1	360	deg	8MR174-11-28-MEn1.cfg	8MR174	28	HEDM5500R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR174E-11-20	360	deg	8MR174E-11-20.cfg	8MR174	20	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR174E-11-28	360	deg	8MR174E-11-28.cfg	8MR174	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR174E-11-28S	360	deg	8MR174E-11-28S.cfg	8MR174	28S	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR174E-11-28-MEn1	360	deg	8MR174E-11-28-MEn1.cfg	8MR174	28	HEDM5500R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR174EV-11-VSS42	360	deg	8MR174EV-11-VSS42.cfg	8MR174V	VSS42	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR174V-11-VSS42	360	deg	8MR174V-11-VSS42.cfg	8MR174V	VSS42	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot

8MR190 - моторизованный ротатор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MR190-2-28	360	deg	8MR190-2-28.cfg	8MR190	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR190-2-28-E4	360	deg	8MR190-2-28-E4.cfg	8MR190	28	RevS	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR190-2-WH-28-E3	360	deg	8MR190-2-WH-28-E3.cfg	8MR190	28	RevS	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR190-2-28-MEn1	360	deg	8MR190-2-28-MEn1.cfg	8MR190	28	HEDM5500	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR190-2-4233	360	deg	8MR190-2-4233.cfg	8MR190	4233	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR190-2-4247	360	deg	8MR190-2-4247.cfg	8MR190	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR190-2-ZSS43	360	deg	8MR190-2-ZSS43.cfg	8MR190	ZSS43	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR190V-2-VSS42	360	deg	8MR190V-2-VSS42.cfg	8MR190V	VSS42-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR190E-2-28	360	deg	8MR190E-2-28.cfg	8MR190	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR190E-2-28-E3	360	deg	8MR190E-2-28-E3.cfg	8MR190	28	RevS	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR190-2-28-E3	360	deg	8MR190-2-28-E3.cfg	8MR190	28	RevS	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR190E-2-4233	360	deg	8MR190E-2-4233.cfg	8MR190	4233	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR190E-2-4247	360	deg	8MR190E-2-4247.cfg	8MR190	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR190E-2-ZSS43	360	deg	8MR190E-2-ZSS43.cfg	8MR190	ZSS43	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR190EV-2-VSS42	360	deg	8MR190EV-2-VSS42.cfg	8MR190V	VSS42	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR190-90-59	360	deg	8MR190-90-59.cfg	8MR190-90-59	5918-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR190-90-59-MEn1	360	deg	8MR190-90-59-MEn1.cfg	8MR190-90-59	5918-R	HEDM5500R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR190-90-4247	360	deg	8MR190-90-4247.cfg	8MR190-90-4247	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR190-90-4247-MEn1	360	deg	8MR190-90-4247-MEn1.cfg	8MR190-90-4247	4247R	HEDM5500R	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR190V-90-VSS43	360	deg	8MR190V-90-VSS43.cfg	8MR190-90-VSS43	VSS43	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR190-2-DCE	360	deg	8MR190-2-DCE.cfg	8MR190-2-DCE	Maxon-339152	EncMRtypeML	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot

8MR191 - моторизованный ротатор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MR191-1-28	360	deg	8MR191-1-28.cfg	8MR191	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191-1-4233	360	deg	8MR191-1-4233.cfg	8MR191	4233	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191-1-4247	360	deg	8MR191-1-4247.cfg	8MR191	4247	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191-1-ZSS43	360	deg	8MR191-1-ZSS43.cfg	8MR191	ZSS43	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191-1-4209-E4	360	deg	8MR191-1-4209-E4.cfg	8MR191	4209	RevS	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191-30-28	360	deg	8MR191-30-28.cfg	8MR191	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191-30-4233	360	deg	8MR191-30-4233.cfg	8MR191	4233	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191-30-4247	360	deg	8MR191-30-4247.cfg	8MR191	4247	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191-30-4209-E4	360	deg	8MR191-30-4209-E4.cfg	8MR191	4209	RevS	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191-30-ZSS43	360	deg	8MR191-30-ZSS43.cfg	8MR191	ZSS43	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191-28	360	deg	8MR191-28.cfg	8MR191	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191-4233	360	deg	8MR191-4233.cfg	8MR191	4233	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191-4247	360	deg	8MR191-4247.cfg	8MR191	4247	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191-ZSS43	360	deg	8MR191-ZSS43.cfg	8MR191	ZSS43	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191E-1-28	360	deg	8MR191E-1-28.cfg	8MR191	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191E-1-4233	360	deg	8MR191E-1-4233.cfg	8MR191	4233	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191E-1-4247	360	deg	8MR191E-1-4247.cfg	8MR191	4247	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191E-1-ZSS43	360	deg	8MR191E-1-ZSS43.cfg	8MR191	ZSS43	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191E-30-28	360	deg	8MR191E-30-28.cfg	8MR191	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191E-30-4233	360	deg	8MR191E-30-4233.cfg	8MR191	4233	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191E-30-4247	360	deg	8MR191E-30-4247.cfg	8MR191	4247	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191E-30-ZSS43	360	deg	8MR191E-30-ZSS43.cfg	8MR191	ZSS43	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191E-28	360	deg	8MR191E-28.cfg	8MR191	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191E-4233	360	deg	8MR191E-4233.cfg	8MR191	4233	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191E-4247	360	deg	8MR191E-4247.cfg	8MR191	4247	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191E-ZSS43	360	deg	8MR191E-ZSS43.cfg	8MR191	ZSS43	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot

8MR191V-1-VSS42	360	deg	8MR191V-1-VSS42.cfg	8MR191V	VSS42	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191V-30-VSS42	360	deg	8MR191V-30-VSS42.cfg	8MR191V	VSS42	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191V-VSS42	360	deg	8MR191V-VSS42.cfg	8MR191V	VSS42	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191EV-1-VSS42	360	deg	8MR191EV-1-VSS42.cfg	8MR191V	VSS42	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191EV-30-VSS42	360	deg	8MR191EV-30-VSS42.cfg	8MR191V	VSS42	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MR191EV-VSS42	360	deg	8MR191EV-VSS42.cfg	8MR191V	VSS42	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot

8MRB250 - большой моторизованный ротатор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MRB240-152-59	360	deg	8MRB240-152-59.cfg	8MRB240	5918	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MRB240-152-59D	360	deg	8MRB240-152-59D.cfg	8MRB240	5918	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MRB450-350-60-MEn2	360	deg	8MRB450-350-60-MEn2.cfg	8MRB450	6018	WEDL5541-B	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot

8MRU - универсальный моторизованный ротатор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MRU-1	360	deg	8MRU-1.cfg	8MRU	28S	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MRU-1TP	360	deg	8MRU-1TP.cfg	8MRU	28S	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot
8MRU-1WA	360	deg	8MRU-1WA.cfg	8MRU	28S	NoEnc	NoBrake	NoAcc	Buttons	WA	XismusbDef	XIDefRot

8MPR16-1 - Моторизованный вращатель поляризатора

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MPR16-1-F	360	deg	8MPR16-1-F.cfg	8MPR16	FAULHABER-AM1020-V-3-16	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot

8MRH240 - большой моторизованный ротатор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MRH240-60	360	deg	8MRH240-60.cfg	8MRH240	5918	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefRot

7.1.3. Вертикальные моторизованные трансляторы


8MVT100 - вертикальный моторизованный транслятор


8MVT120 - вертикальный моторизованный транслятор


8MVT188 - вертикальный моторизованный транслятор


8MVT40 - вертикальный моторизованный транслятор


8MVT70 - вертикальный моторизованный транслятор

8MVT100 - вертикальный моторизованный транслятор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MVT100-25-1	26	mm	8MVT100-25-1.cfg	8MVT100	4118L1804R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8MVT120 - вертикальный моторизованный транслятор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MVT120-5-4247	5	mm	8MVT120-5-4247.cfg	8MVT120-5	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MVT120-12-4247	12.7	mm	8MVT120-12-4247.cfg	8MVT120-12	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MVT120-25-4247	25.4	mm	8MVT120-25-4247.cfg	8MVT120-25	4247R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8MVT188 - вертикальный моторизованный транслятор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MVT188-20	8744052	steps	8MVT188-20.cfg	8MVT188	5918	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8MVT40 - вертикальный моторизованный транслятор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MVT40-13-1	13	mm	8MVT40-13-1.cfg	8MVT40	28SR	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8MVT70 - вертикальный моторизованный транслятор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MVT70-13-1	13	mm	8MVT70-13-1.cfg	8MVT70	4118S1404R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

7.1.4. Моторизованные винты


8MS00 - моторизованный винт


8CMA06 - моторизованный винт


8CMA20 - моторизованный винт


8CMA28 - моторизованный винт


8CMA16DC- моторизованный винт

8MS00 - моторизованный винт

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MS00-10	10	mm	8MS00-10.cfg	8MS00-10	4233-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MS00-10-28	10	mm	8MS00-10-28.cfg	8MS00-10	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MS00-25	25	mm	8MS00-25.cfg	8MS00-25	4233-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MS00-25-28	25	mm	8MS00-25-28.cfg	8MS00-25	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MS00V-10-VSS43	10	mm	8MS00V-10-VSS43.cfg	8MS00V-10	VSS43-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MS00V-25-VSS43	25	mm	8MS00V-25-VSS43.cfg	8MS00V-25	VSS43-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8CMA06 - моторизованный винт

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8CMA06-13_10	13	mm	8CMA06-13_10.cfg	8CMA06-13	20-Rev	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8CMA06-13_15	13	mm	8CMA06-13_15.cfg	8CMA06-13	20-Rev	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8CMA06-25_15	25	mm	8CMA06-25_15.cfg	8CMA06-25	20-Rev	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8CMA20 - моторизованный винт

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8CMA20-8_15	8	mm	8CMA20-8_15.cfg	8CMA20	20-Rev	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8CMA28 - моторизованный винт

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8CMA28-10	10	mm	8CMA28-10.cfg	8CMA28	28R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8CMA16DC - моторизованный винт

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8CMA16DC-13_15	13	mm	8CMA16DC-13_15.cfg	8CMA16DC	FAULHABER-1524T006SR	IE2-256	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefDC
8CMA16DC-25_15	25	mm	8CMA16DC-25_15.cfg	8CMA16DC	FAULHABER-1524T006SR	IE2-256	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefDC
8CMA16DCV-13_15	13	mm	8CMA16DCV-13_15.cfg	8CMA16DCV	FAULHABER-1524T006SR	IE2-16	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefDC
8CMA16DCV-25_15	25	mm	8CMA16DCV-25_15.cfg	8CMA16DCV	FAULHABER-1524T006SR	IE2-16	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDefDC

7.1.5. Моторизованные гониометры



8MG00 - моторизованный гониометр



8MG99 - моторизованный гониометр

8MG00 - моторизованный гониометр

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MG00-50	0.5	deg	8MG00-50.cfg	8MG00-50	4233-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MG00-80	0.5	deg	8MG00-80.cfg	8MG00-80	4233-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MG00V-50	0.5	deg	8MG00V-50.cfg	8MG00-50	VSS42-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MG00V-80	0.5	deg	8MG00V-80.cfg	8MG00-80	VSS42-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8MG99 - моторизованный гониометр

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MG99-50	0.5	deg	8MG99-50.cfg	8MG99-50	4233-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MG99-80	0.5	deg	8MG99-80.cfg	8MG99-80	4233-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MG99V-50	0.5	deg	8MG99V-50.cfg	8MG99-50	VSS42-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MG99V-80	0.5	deg	8MG99V-80.cfg	8MG99-80	VSS42-R	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

7.1.6. Моторизованные держатели зеркал



8MTOM2 - моторизованный оптический держатель



8MUP21 - моторизованный оптический держатель



8MBM24 - моторизованный держатель зеркала



8MMA60 - моторизованный держатель зеркала



8MBM57 - моторизованный держатель зеркала с большой апертурой



8MKVDOM - вертикальный моторизованный оптический держатель

8MTOM2 - моторизованный оптический держатель

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MTOM2-1	4	mm	8MTOM2-1.cfg	8MTOM2	20	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8MUP21 - моторизованный оптический держатель

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MUP21-2	4	mm	8MUP21-2.cfg	8MUP21	20-Rev	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8MBM24 - моторизованный держатель зеркала

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MBM24-1-2	4	mm	8MBM24-1-2.cfg	8MBM24	20-Rev	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MBM24-2-2	4	mm	8MBM24-2-2.cfg	8MBM24	20-Rev	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MBM24-3-2	4	mm	8MBM24-3-2.cfg	8MBM24	20-Rev	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8MMA60 - моторизованный держатель зеркала

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MMA60-1	5	deg	8MMA60-1.cfg	8MMA60	4233	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MMA60-2	5	deg	8MMA60-2.cfg	8MMA60	4233	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MMA60-40	5	deg	8MMA60-40.cfg	8MMA60	4233	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8MBM57 - моторизованный держатель зеркала с большой апертурой

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MBM57-2	5	deg	8MBM57-2.cfg	8MBM57	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MBM57-3	5	deg	8MBM57-3.cfg	8MBM57	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MBM57-4	5	deg	8MBM57-4.cfg	8MBM57	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MBM57-6	5	deg	8MBM57-6.cfg	8MBM57	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

8MKVDOM - вертикальный моторизованный оптический держатель

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MKVDOM-1	6	mm	8MKVDOM-1.cfg	8MKVDOM	20	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MKVDOM-2	6	mm	8MKVDOM-2.cfg	8MKVDOM	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

Моторизованные аттенуаторы



10MCWA168 - моторизованный аттенуатор



10MWA168 - моторизованный аттенуатор

10MCWA168 - моторизованный аттенуатор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
10MCWA168-1	360	deg	10MCWA168-1.cfg	10MCWA168	4247	RevS	NoBrake	NoAcc	Buttons	WA168	XismusbDef	XIDef
10MCWA168-20	360	deg	10MCWA168-20.cfg	10MCWA168	4247	RevS	NoBrake	NoAcc	Buttons	WA168	XismusbDef	XIDef

10MWA168 - моторизованный аттенуатор

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
10MWA168-1	360	deg	10MWA168-1.cfg	10MWA168	4247ROT	NoEnc	NoBrake	NoAcc	Buttons	WA168	XismusbDef	XIDef
10MWA168-20	360	deg	10MWA168-20.cfg	10MWA168	4247ROT	NoEnc	NoBrake	NoAcc	Buttons	WA168	XismusbDef	XIDef

Моторизованные диафрагмы



8MID98 - моторизованная диафрагма

8MID98 - моторизованная диафрагма

Дата обновления: 15 March 2018

Модель	Ход	Ед. изм.	Профиль	Позиционер	Мотор	Энкодер	Тормоз	Аксессуары	Периферия	Аттенюатор	Контроллер	XILab
8MID98-4-90	93	mm	8MID98-4-90.cfg	8MID98-4-90	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MID98-4-H	94	mm	8MID98-4-H.cfg	8MID98	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef
8MID98-4-N	94	mm	8MID98-4-N.cfg	8MID98	28	NoEnc	NoBrake	NoAcc	Buttons	NoAtten	XismusbDef	XIDef

7.2. Программное обеспечение

Полный комплект программного обеспечения для Windows, обновлено 02.03.2018

Включает в себя программу XILab только для Windows, комплект разработчика, прошивку последних версий и пользовательскую документацию.

Загрузите комплект программного обеспечения. Убедитесь, что все контроллеры отключены от компьютера. Запустите программу установки XILab. Более полная информация по установке помещена в разделе Краткое руководство и начало работы .

XILab, версия 1.14.8, обновлено 15.02.2018

Программу XILab вы можете скачать отсюда:

Windows XP, Vista, 7, 8, 10	xilab-1.14.8-win32_win64.exe
Generic Linux 32 bit	xilab-1.14.8-ia32.AppImage
Generic Linux 64 bit	xilab-1.14.8-x86_64.AppImage
MacOS X	xilab-1.14.8-osx64.tar.gz

Все версии
Таблица совместимости
Список изменений

Комплект разработчика, версия 2.9.12, обновлено 15.02.2018

Загрузите и разархивируйте файл: libximc-2.9.12-all.tar.gz

С описанием комплекта разработчика можно ознакомиться в разделе Руководство по программированию или в папке /docs-ru/index.html указанного выше архива. Библиотека находится в /ximc-2.9.12

PDF-версия руководства по программированию может быть скачана отсюда

Все версии
Таблица совместимости
Список изменений

Микропрограмма контроллера

Скачать для 8SMC5: ximc-4.1.8-hw2.3.x.cod, обновлено 12.01.2018
Скачать для 8SMC4: ximc-3.9.18-hw2.2.x.cod, обновлено 06.06.2017

Процедура обновления микропрограммы:

1. Скачайте файл с нужной версией микропрограммы.
2. Запустите XIlab
3. Рекомендуется на всякий случай сохранить конфигурационный файл XiLab перед обновлением микропрограммы.
4. Откройте Settings... -> Device configuration -> About device, нажмите кнопку "Update" и выберите новый файл прошивки.
5. Не беспокойтесь, если программа XILab не отвечает. Подождите пока прошивка будет успешно обновлена. Обычно это занимает около 10-15 секунд.
6. Если настройки изменились, то загрузите сохранённый конфигурационый файл и сохраните его в энергонезависимую память контроллера с помощью кнопки Save to Flash...

Все версии
Таблица совместимости
Список изменений

Конфигурационные файлы

Данные файлы помещены в разделе Конфигурационные файлы.

Примеры для LabView

Примеры использования контроллера в среде LabView можно скачать по ссылке: labview-2.9.12-Labview_12.0.7z

Драйверы

Драйверы контроллеру не требуются, но в среде Windows требуется inf-файл. Он устанавливается автоматически при установке программного обеспечения XILab. Этот файл вы можете найти в папке C:\Program Files\XILab\Driver после установки программы XILab. Также указанный файл можно скачать по ссылке: XIMC_driver.inf

Драйверы или специальные файлы для Linux или Mac не требуются.

Программное обеспечение для контроллеров 8SMC1 с поддержкой контроллеров 8SMC4

Программное обеспечение можно скачать по ссылке: microsmc-2.3.0-win32_win64.zip
Подробная информация находится здесь.

Прошивка для USB-Ethernet переходника, версия 1.1.0, обновлено 11.07.2016

Программное обеспечение можно скачать по ссылке: cubian-1.1.0-armhf.7z
Подробная информация находится здесь

Все версии
Список изменений

REVEALER, версия 0.1.0, обновлено 16.04.2017

Программное обеспечение можно скачать по ссылке

Windows XP, Vista, 7, 8, 10	revealer-0.1.0-win32.zip
Linux 32-bit	revealer-0.1.0-lin32.tar.gz
Linux 64-bit	revealer-0.1.0-lin64.tar.gz
MacOS	revealer-0.1.0-osx64.zip
Java	revealer-0.1.0-j.jar

Подробная информация находится здесь

Все версии
Список изменений

Все версии XILab

Версия	Windows XP, Vista, 7, 8, 10	MacOS X	Generic Linux 32-bit	Generic Linux 64-bit	Debian/Ubuntu 32-bit	Debian/Ubuntu 64-bit	RedHat/OpenSUSE 64-bit	RedHat/OpenSUSE 32-bit
1.14.8	Скачать	Скачать	Скачать	Скачать	-	-	-	-
1.14.7	Скачать	Скачать	Скачать	Скачать	-	-	-	-
1.14.5	Скачать	Скачать	Скачать	Скачать	-	-	-	-
1.13.14	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	Скачать
1.13.13	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	Скачать
1.12.18	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	Скачать
1.12.15	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	Скачать
1.12.14	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	Скачать
1.12.13	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	Скачать
1.12.10	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	Скачать
1.12.8	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	Скачать
1.11.12	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	Скачать
1.11.10	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	-
1.10.14	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	-
1.10.11	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	-
1.10.10	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	-
1.10.6	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	-
1.9.18	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	-
1.9.17	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	-
1.9.16	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	-
1.9.15	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	-
1.9.14	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	-
1.9.13	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	-
1.9.12	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	-
1.9.11	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	-
1.8.30	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	-
1.8.29	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	-
1.8.28	Скачать	Скачать	-	-	Скачать	Скачать	Скачать	-

Список изменений XILab

▸ Посмотреть...

Все версии библиотеки libximc, MicroSMC и XIMC_Labview

Версия	Libximc	Libximc sources	XIMC_Labview	MicroSMC
2.9.12	Скачать	Скачать	Скачать для Labview 12.0	-
2.9.11	Скачать	-	Скачать для Labview 12.0	-
2.9.8	Скачать	-	-	-
2.8.10	Скачать	-	-	-
2.8.9	Скачать	-	-	-
2.8.8	Скачать	-	-	-
2.8.7	Скачать	-	-	-
2.8.5	Скачать	-	Скачать для Labview 11.0 Скачать для Labview 12.0 Скачать для Labview 8.2	-
2.8.4	Скачать	-	Скачать для Labview 12.0	-
2.8.0	Скачать	-	Скачать для Labview 12.0 Скачать для Labview 8.0	-
2.3.2	Скачать	-	-	-
2.3.1	Скачать	-	-	-
2.3.0	Скачать	-	Скачать для Labview 12.0 Скачать для Labview 8.6	Скачать
2.2.2	Скачать	-	Скачать для Labview 12.0	Скачать
2.2.1	Скачать	-	Скачать для Labview 12.0	Скачать
2.2.0	Скачать	-	-	-
2.0.5	Скачать	-	-	-

Официально поддерживается Labview 12.0, остальные версии являются пересохранением с совместимостью для более ранних версий (не тестировалось)

Инструкция по сборке libximc-src находится в архиве libximc папке ximc\doc-ru

Список изменений libximc

▸ Посмотреть...

Список изменений XIMC_Labview

▸ Посмотреть...

Список изменений MicroSMC

▸ Посмотреть...

Все версии прошивки

Версия (для 8SMC4)	Файл	Версия (для 8SMC5)	Файл
3.9.18	Скачать	4.1.8	Скачать
3.9.16	Скачать	4.1.6	Скачать
3.9.14	Скачать	4.1.5	Скачать
3.9.12	Скачать	4.1.4	Скачать
3.9.11	Скачать	4.0.9	Скачать
3.9.10	Скачать	4.0.7	Скачать
3.9.9	Скачать	4.0.4	Скачать
3.9.8	Скачать		
3.9.7	Скачать		
3.9.5	Скачать		
3.8.19	Скачать		
3.8.17	Скачать		
3.8.16	Скачать		
3.8.14	Скачать		
3.8.11	Скачать		
3.8.10	Скачать		
3.8.9	Скачать		
3.8.8	Скачать		
3.8.7	Скачать		
3.8.4	Скачать		
3.8.3	Скачать		

Список изменений прошивки

▸ Посмотреть...

Таблица совместимости

Версия протокола	XILab	libximc	прошивка
16.3	1.8.28-1.8.30	2.0.5	3.8.3-3.8.4
16.6	1.9.11-1.9.18	2.2.1-2.2.2	3.8.7-3.8.11
16.6	1.10.6-1.10.14	2.3.0-2.3.2	3.8.7-3.8.11
16.8	1.11.10-1.11.12	2.5.0	3.9.5-3.9.8
16.8	1.12.7-1.12.8	2.8.0	3.9.5-3.9.8
16.10	1.12.10	2.8.4-2.8.5	3.9.9
16.10	1.12.13	2.8.7	3.9.10-3.9.11
16.11	1.12.14	2.8.8-2.8.9	3.9.12
16.12	1.12.15	2.8.10	3.9.14
16.12	1.12.18	2.8.10-2.8.12	3.9.14-3.9.16
17.2	1.13.13-1.13.14	2.9.8-2.9.12	3.9.16-3.9.18, 4.0.4-4.0.9
18.1	1.14.5-1.14.8	2.10.3	4.1.4-4.1.8

Все версии прошивок для USB-Ethernet переходника

Версия прошивки	прошивка
1.1.0	Скачать
1.0.0	Скачать

Подробная информация находится здесь

Список изменений USB-Ethernet переходника

▸ Посмотреть...

Все версии программ для USB-Ethernet переходника REVEALER

Версия	Linux x32	Linux x64	Mac OS	Windows	Java
0.1.0	Скачать	Скачать	Скачать	Скачать	Скачать

Список изменений REVEALER

▸ Посмотреть...

8. Сопутствующие устройства

1. Ethernet переходник

8.1. Управление контроллером через Ethernet

1. Обзор
2. Администрирование

8.1.1. Обзор Ethernet переходника

Общая информация и внешний вид

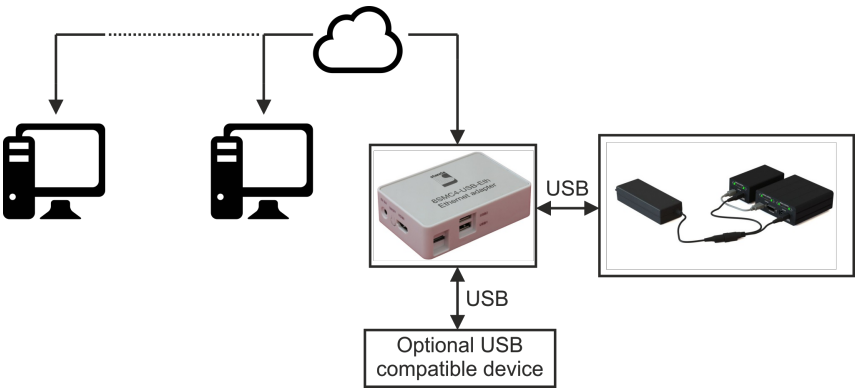


Внешний вид адаптера 8SMC4-USB-Eth1

8SMC4-USB-Eth1[1] - это универсальное устройство, основанное на одноплатном компьютере Cubieboard2 под управлением ОС Linux, основной задачей которого является предоставление доступа к контроллерам моторов через различные Ethernet-ориентированные интерфейсы. Помимо этого, **8SMC4-USB-Eth1** представляет собой агрегатор сервисов, связанных с наиболее широко распространенными областями применения контроллеров. Так, например, в комплектации по умолчанию устройство оснащено сервисом сетевой трансляции с подключенных web-камер, системой автоматического обнаружения в LAN, а также несколькими интерфейсами для удаленной работы с контроллерами:

1. Интерфейс системы управления TANGO.
2. Интерфейс XIMC (совместимый с XiLab и libximc).

Помимо этого переходник также имеет встроенный веб-интерфейс администрироания, который позволяет с лёгкостью управлять устройством и следить за его состоянием.



Упрощённая блок схема системы с управлением через Ethernet переходник

Внешний вид системы с разных ракурсов (жирным отмечены разъёмы, используемые в текущей версии устройства):



Вид адаптера спереди. Слева направо: ***разъём питания, кнопка включения питания****, разъём HDMI.*



Вид адаптера справа. Слева направо: разъём для микро-SD карты, ***два разъёма USB****.*



Вид адаптера сзади. Слева направо: ***Ethernet разъём****, mini-USB типа B, кнопка перехода в режим FEL, аналоговые вход/выход для микрофона и наушников.*



Вид адаптера слева. Отверстие с ИК приёмником.

Основные требования

Конифгурация сети

- Необходимо наличие DHCP сервера, поддерживающего автоматическую раздачу ip адресов.
- Поддержка протокола IPv4 сервером и компьютером.
- Порт 48150 не должен быть заблокирован. Причиной блокирования зачастую может быть наличие антивирусов, программ, осуществляющих контроль и фильтрацию сетевых пакетов (фаерволы, брандмауэры).

Другое

- Наличие розетки 220В.
- Ethernet кабель, USB кабель.

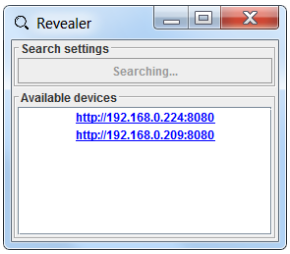
[1] Артикул был изменен в 2017 г., ранее устройство имело артикул 8SMC4-USB-Eth.

8.1.2. Администрирование

> *Примечание.* Если вы не можете войти в интерфейс администрирования, то велика вероятность того, что у вас старая версия устройства без поддержки интерфейса администрирования и TANGO. Вы можете самостоятельно обновить прошивку устройства при помощи инструкции.

Автоматическое обнаружение устройств

Для того, чтобы облегчить задачу поиска подключенных к вашей локальной сети устройств **Standa 8SMC4-USB-Eth1**, мы предоставляем небольшую утилиту под названием «Revealer». Вы можете скачать её со страницы программного обеспечения.



Пользовательский интерфейс утилиты «Revealer»

Графический пользовательский интефрейс утилиты «Revealer» очень прост в использовании. Нажатие на кнопку Search (или Restart) на панели Search settings запустит сканирование локальной сети, которые занимает приблизительно 3 секунды. В результате сканирования на панель Available devices будет выведен интерактивный список всех обнаруженных в вашей локальной сети устройств **Standa 8SMC4-USB-Eth1**. Нажатие на какую-либо ссылку в этом списке приведёт к открытию нового окна (или вкладки) вашего стандратного системного браузера со страницей веб-интерфейса Администрирования соответствующего устройства.

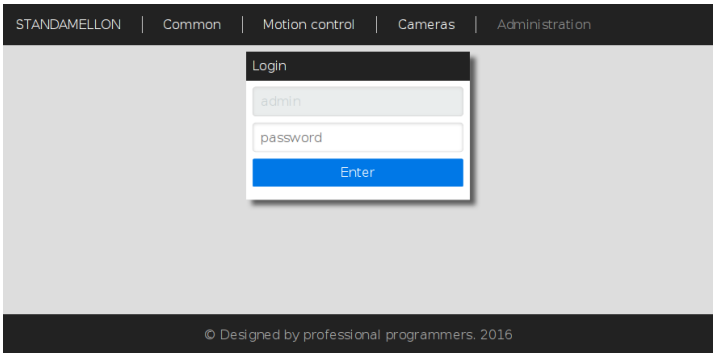
Следует отметить, что утилита для своей работы требует Java Runtime Environment (также известную как JRE) версии 6 или выше. Велика вероятность того, что на вашем компьютере уже установлена подходящая версия JRE т.к. этот пакет требуется для работы множества широкораспространенных программных продуктов. В таком случае для запуска «Revealer» вам всего лишь необходимо сделать двойной клик на файле "revealer-j_0.1.0.jar", что приведёт к открытию вышеописанного пользовательского интерфейса. Иначе на вашем компьютере отсутствует подходящая версия JRE и вам следует:

1. Загрузить специально подготовленный пакет для вашей операционной системы, который содержат в себе всё необходимое для запуска «Revealer» без JRE. В этом случае вам следует запускать исполняемый файл «Revealer» из такого пакета.
2. Установить JRE самостоятельно. Хорошим кандидатом является версия JRE от компании Oracle, установить которую можно следую официальным инструкциям.

> **Предупреждение.** Для того, чтобы обнаружить все **Standa 8SMC4-USB-Eth1** в вашей локальной сети, "Revealer" использует механизм широковещательных UDP-запросов. Это означает, что его использование может быть нежелательным/невозможным в сетях, где в том или ином виде запрещены широковещательные UDP-запросы.

Обзор

Standa 8SMC4-USB-Eth1 имеет предустановленный веб-интерфейс администрирования, который позволяет в интерактивном режиме управлять функционалом устройства, а также получать актуальную информацию о его состоянии.



Страница входа в интерфейс Администрирования

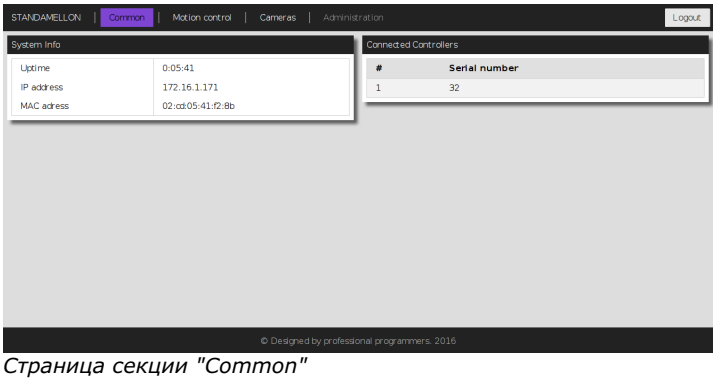
Для доступа к веб-интерфейсу устройства необходимо открыть URL http://[address], где *address* является IP-адресом устройства в вашей локальной сети (узнать который можно в т.ч. используя программу "Revealer"). В случае если вы входите в веб-интерфейс первый раз (а также если вы не включали опцию запоминание паролей в браузере или выключили cookies), то вам будет необходимо пройти процедуру аутентификации, используюя "admin" в качестве логина и пароля, после чего нажать "Enter".

> *Примечание.* Следует обратить внимание, что для наиболее комфортной работы с веб-интерфейсом javascript должен быть включен в вашем браузере.

> *Примечание.* Работа веб-интерфейса протестирована на следующих браузерах: MS IE 9, MS IE 10, MS IE 11, MS Spartan, Firefox, Chrome.

Интерфейс администрирования разделён на три функциональные секции.

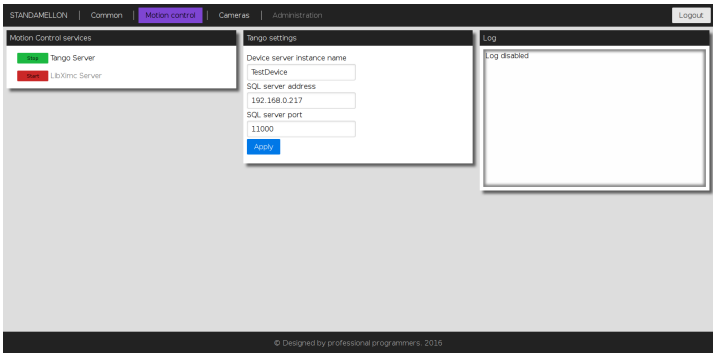
Секция "Common"



Страница секции "Common"

Здесь содержится общая информация о системе и список серийных номеров подключенных к устройству контроллеров.

Секция "Motion control"

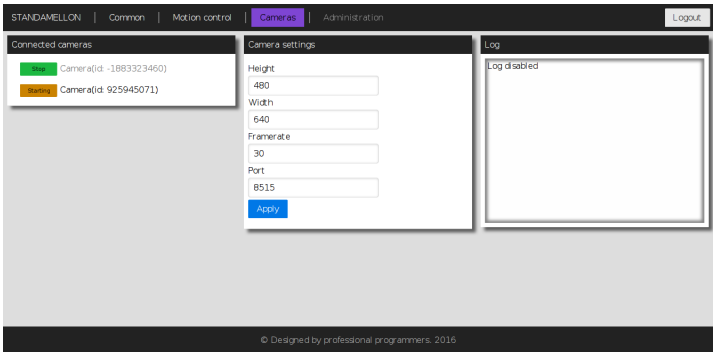


Страница секции "Motion control"

Эта секция целиком посвящяна поддерживаемым устройством сервисов управления движением. Панель "Motion Control services" содержит список всех доступных в данный момент сервисов управления движением. Нажатие на название какого-либо сервиса откроет соответсутвующую панель конфигурации (если сервис имеет какие-либо изменяемы настройки). Нажатие на кнопку "Apply" в панели настроек приведёт к сохранению настроек и полному перезапуску сервиса.

> *Примечание.* В любой момент времени может быть активен только один из сервисов группы управления движением контроллеров. Запуск какого-либо сервиса приведёт к автоматической остановке предыдущего активного (при наличии такового).

Секция "Cameras"



Страница секции "Cameras"

Включает в себя список подключенных к устройству сетевых камер, настройки серверов трансляции, индикаторы и переключатели состояния для каждой из них.

Управление сервисами

Интерфейс Администрирования предназначен для того, чтобы конечный пользователь мог свободно конфигурировать и изменять состояние сервисов, досутпных на устройстве. Для управления всеми сервисами - вне зависимости от их типа, функционала и предназначения - существует единый набор метафор. С каждым доступным для управляения сервисом всегда ассоциирована соответсвующая кнопка-индикатор, которая служит одновременно для отображения статуса сервиса и управления им:

- Красный цвет индикатора и надпись «Stopped» означают, что сервис остановлен; нажатие на кнопку начнёт запуск процесса.
- Оранжевый цвет индикатора и надпись «Starting» означают, что сервис в процессе запуска; нажатие на кнопку-индикатор остановит запуск процесса.
- Зелёный цвет конпки-индикатора означает, что процесс запущен; нажатие на кнопку начнёт остановку процесса.
- Оранжевый цвет кнопки-индикатора означает, что процесс останавливается; в этом состоянии кнопка недоступна для нажатия т.к. остановка процесса не может быть прервана.

> *Примечание.* Многие из сервисов имеют изменяемые настройки и в случае изменения какой-либо из них соответствующий процесс будет немедленно перезапущен (что, к примеру, может привести к потере связи с контроллером и потребовать переподключения в случае использования сервиса "LibXimc server"). Состояние сервиса можно отслеживать по цвету и надписи на кнопке-индикаторе.

Обновление прошивки

> **Предупреждение.** В ходе обновления прошивки содержимое вашей MicroSD карты и flash-памяти устройства 8SMC4-USB-Eth1 будет целиком стёрто. Убедитесь, что вы сделали резервные копии всех ваших данных.

1. Загрузите необходимую версию прошивки со страницы программного обеспечения.
2. Вам понадобится MicroSD на 4 Гбайт (или больше).
3. Загрузите и разархивируйте образ автоинсталлятора (в случае если вы используете Windows и у вас нет подходящего архиватора, вы можете воспользоваться 7-Zip (официальный сайт).
4. Запишите образ автоинсталлятора на MicroSD карту:
    - Для Unix-подобных ОС:
        1. Вы можете использовать стандартную системную утилиту **dd** (например

           ```
           dd if=autoinstall_image_2015-12-17.bin of=/dev/sdX bs=4M; sync
           ```

           , но не забудьте изменить пути к файлам на ваши собственные).
    - ОС семейства Windows:
        1. Загрузите Win32 Disk Imager (официальный сайт) и запустите его.
        2. Укажите путь к файлу автоинсталлятора в поле "Image File".
        3. Выберите диск, который соответствует вашей MicroSD карте в поле «Device».
        4. Нажмите кнопку «Write» и дождитесь окончания записи.
5. Вставьте MicroSD карту с записанным на неё автоинсталлятором в 8SMC4-USB-Eth1 и включите его.
6. Установка занимает приблизительно 10 минут. Вы можете узнать статус установки по LED-индикатору, который находится на плате. Его можно увидеть через отверстия для USB-разъемов в корпусе устройства:
    1. После включения устройства LED горит или периодически загорается одинарными вспышками — автоинсталлятор запускается.
    2. LED периодически загорается двойными зелёными вспышками — идёт установка.
    3. LED горит зелёным — установка завершена.
7. Выключите устройство (вытащите кабель питания) и извлеките MicroSD карту.

8. Включите устройство и подождите 2-7 минут пока завершится инициализация после перепрошивки.

Предупреждение. Если после перепрошивки revealer не способен обнаружить устройство даже после указанного выше периода инициализации(и вы не можете подключиться к веб-интерфейсу администрирования и/или использовать другие сервисы), то попробуйте перезагрузить устройство. Если это не помогло, то повторите шаги 5-8.

После вышеописанных шагов ваш 8SMC4-USB-Eth1 перепрошит и может свободно использоваться в стандартном режиме.

Возможные проблемы и их решения

Предупреждение. Если после перепрошивки в работе устройства наблюдаются сбои, в процессе перепрошивки диоды не горели, как описано выше, и ваше устройство выпущено **после октября 2017 г.**, обратитесь в тех. поддержку.

Неинициализированный загрузчик Cubieboard2

В некоторых редких случаях из-за технических проблем вышеприведённая инструкция может не дать ожидаемого результат. Одной из причин может быть то, что загрузчик одноплатного компьютера Cubieboard2, на базе которого работает 8SMC4-USB-Eth1, не был соответствующим образом инициализирован. Можно провести инициализацию самостоятельно:

1. Загрузите PhoenixSuit (официальный сайт) и распакуйте его;
2. Загрузите образ «Lubuntu server» с поддержкой SoC A20 rev.B. для PhoenixSuit(официальный сайт , «Firmwares» -> «Cubieboard2» -> «Lubuntu Server») и распакуйте его.
3. Запустите PhoenixSuite. Если программе потребуется обновление, то дайте согласие и дождитесь его окончания.
4. Перейдите на страницу «Firmware» и укажите путь к распакованному образу «Lubuntu server».
5. Выключите устройство. Нажмите и удерживайте кнопку FEL. Подсоедините устройство к вашему PC через USB-OTG. Отпустите кнопку FEL.
6. На ОС семейства Windows вам может потребоваться установить драйвер для устройства вручную.
    1. Откройте Диспетчер Устройств.
    2. Найдите «Неизвестное устройство» и нажмите на него правой кнопкой мыши.
    3. В появившемся контекстном меню нажмите «Обновить драйвер...».
    4. Укажите «PhoenixSuit1.0.7\PhoenixSuit\Drivers\AW_Driver» в качестве папки для поиска драйверов.
7. После этого в PhoenixSuite появится модальное окно, в котором необходимо нажать «YES (Full install)» и дождаться окончания процесса.
8. Выполнить шаги инструкции по обновлению прошивки.