

УТВЕРЖДЕН

XXXX.ЭХХ.001.01.00 13-ЛУ

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ВЗАИМОДЕЙСТВИЯ С
МОДУЛЕМ УПРАВЛЕНИЯ БЕСКОЛЛЕКТОРНЫМИ
ДВИГАТЕЛЯМИ**

XXXX.ЭХХ.001.01.00

Программное обеспечение

Описание программы

XXXX.ЭХХ.001.01.00 13

Листов 12

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

2020

АННОТАЦИЯ

Настоящий документ содержит описание «Программного обеспечения для взаимодействия с модулем управления бесколлекторными двигателями» XXXX.ЭХХ.001.01.00 13, входящего в состав «Программного обеспечения образовательного робототехнического конструктора».

В описании даны сведения о функциональном назначении программного обеспечения, его логической структуры, используемых технических и программных средствах, приведён обобщённый алгоритм работы и информация о входных и выходных данных.

СОДЕРЖАНИЕ

1	ОБЩИЕ СВЕДЕНИЯ.....	4
2	ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ	5
3	ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ	6
4	ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА	8
5	ВЫЗОВ И ЗАГРУЗКА	9
6	ВХОДНЫЕ ДАННЫЕ	10
7	ВЫХОДНЫЕ ДАННЫЕ.....	11

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Обозначение и наименование программы – «Программное обеспечение для взаимодействия с модулем управления бесколлекторными двигателями» XXXX.ЭХХ.001.01.00 13.

1.2 Языки программирования, использованные при написании программы – python.

1.3 Взаимодействие с модулем происходит по интерфейсу связи I2C.

2 ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

2.1 ПО предназначено для управления и взаимодействия с модулем управления бесколлекторными двигателями.

2.2 ПО обеспечивает:

- настройку режима работы модуля;
- управление бесколлекторными двигателями в режиме ПИД-регулятора;
- управление бесколлекторными двигателями в режиме ШИМ;
- настройку коэффициентов ПИД-регулятора.

3 ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

3.1 Алгоритм работы программы

3.1.1 Пользователь имеет возможность импортировать данную программу в свой код

3.1.2 После импортирования программы, пользователь в коде должен создать экземпляр класса управления моторами (BLDCbot) с параметрами в виде экземпляра класса для работы с шиной I2C и адреса модуля на шине (по умолчанию адрес равен 0x27).

3.1.3 Через созданный экземпляр класса пользователь имеет возможность настройки режима работы модуля, настройки коэффициентов ПИД-регулятора, управления бесколлекторными двигателями, для использования данных функций пользователь должен вызывать в коде соответствующие им методы

3.1.4 При вызове определенного метода осуществляется чтение или запись данных на модуль

3.1.5 Список доступных методов:

- `whoIam()` – метод для тестирования связи с модулем, должен вернуть число 42, если связь установлена корректно;

- `setWorkMode(mode)` – метод для установки режима работы модуля (режима управления моторами), где `mode` может принимать следующие значения: `WORK_MODE_PWM` (режим управления двигателями – напрямую через ШИМ разъем на модуле), `WORK_MODE_PID` (режим управления двигателями – через ШИМ разъем на модуле, но с использованием ПИД-регулятора), `WORK_MODE_PWM_I2C` (режим управления двигателями – управление через I2C, пользователь задает параметры ШИМ, который подается на двигатели), `WORK_MODE_PID_I2C`

(режим управления двигателями – управление через I2C, пользователь задает некоторые условные значения скорости двигателей, которые автоматически пересчитываются в ШИМ на самом модуле);

- `setKp(kp)` – метод установки пропорционального коэффициента регулятора, где `kp` – устанавливаемое значение;

- `setKi(ki)` – метод установки интегрального коэффициента регулятора, где `ki` – устанавливаемое значение;

- `setKd(kd)` – метод установки дифференциального коэффициента регулятора, где `kd` – устанавливаемое значение;

- `setPwmA(direction, pwm)`, `setPwmB(direction, pwm)` – методы, используемые для управления двигателями при установленном режиме работы `WORK_MODE_PWM_I2C`, где `direction` может принимать значения `FORWARD` (ШИМ не инвертирован – двигатель будет двигаться в одну сторону) и `BACKWARD` (ШИМ инвертирован – двигатель будет двигаться в обратную сторону), `pwm` – значение заполнения ШИМа, может принимать значения от 0 до 255;

- `setParrotA(speed)`, `setParrotB(speed)` – методы установки скорости двигателей в условных единицах, где `speed` – устанавливаемое значение скорости, может принимать значения от -100 (полная мощность при реверсивном движении) до 100 (полная мощность при прямом движении);

4 ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

4.1 Для использования программы необходим одноплатный компьютер Raspberry pi 3 model B

5 ВЫЗОВ И ЗАГРУЗКА

5.1 Пользователь должен импортировать данную программу в свой код.

5.2 Входной точкой работы программы является создание пользователем экземпляра класса управления модулем.

5.3 После создания экземпляра класса пользователь получает возможность взаимодействия с модулем.

6 ВХОДНЫЕ ДАННЫЕ

6.1 Основными входными данными являются:

- адрес модуля на шине I2C;
- режим работы модуля;
- значения скоростей двигателей при их управлении.

7 ВЫХОДНЫЕ ДАННЫЕ

7.1 Выходными данными являются данные, отправляемые на модуль.

[illegible]