УТВЕРЖДЕН
ХХХХ.ЭХХ.001.03.00 13-ЛУ

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ВЗАИМОДЕЙСТВИЯ С ПЛАТОЙ РАСШИРЕНИЯ EDUBOT

ХХХХ.ЭХХ.001.03.00

Программное обеспечение

Описание программы

ХХХХ.ЭХХ.001.03.00 13

Листов 13

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

2020

АННОТАЦИЯ

Настоящий документ содержит описание «Программного обеспечения для взаимодействия с платой расширения Edubot» ХХХХ.ЭХХ.001.03.00 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

1.1 Обозначение и наименование программы – «Программное обеспечение для взаимодействия с платой расширения Edubot» ХХХХ.ЭХХ.001.03.00 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 платой расширения Edubot.

2.2 ПО обеспечивает:

– настройку режима управления бесколлекторными двигателями, подключенными к плате;

– управление бесколлекторными двигателями в режиме ПИД-регулятора;

– управление бесколлекторными двигателями в режиме ШИМ;

– настройку коэффициентов ПИД-регулятора;

– управление сервоприводами;

– управление звукоизлучателем;

– управление посылкой онлайн-меток.

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 взаимодействия с платой расширения (EduBot) с параметрами в виде экземпляра класса для работы с шиной I2C и адреса платы расширения на шине (по умолчанию адрес равен 0x27).

3.1.3 Через созданный экземпляр пользователь имеет возможность настройки режима работы управления бесколлекторными двигателями, настройки коэффициентов ПИД-регулятора, управления бесколлекторными двигателями, управления сервоприводами, управления звукоизлучателем, управления посылкой онлайн-меток, для использования данных функций пользователь должен вызывать в коде соответствующие методы или изменить соответствующие поля.

3.1.4 При вызове определенного метода осуществляется чтение или запись данных на плату расширения

3.1.5 После создания экземпляра класса пользователю необходимо разрешить посылку онлайн-меток путем изменения поля экземпляра класса online в значение True, а также запустить поток отправки онлайн-меток вызовом метода start(). В случае, если онлайн-метки не отправляются на плату расширения, то через три секунды она разрывает соединение и останавливает свою работу до момента, пока онлайн-метки снова не начнут отправляться.

3.1.6 Список доступных методов:

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

– setMotorMode(mode) – метод для установки режима управления моторами, где mode может принимать следующие значения: MOTOR_MODE_PWM (режим управления двигателями – управление через I2C, пользователь задает параметры ШИМ, который подается на двигатели), MOTOR_MODE_PID (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;

– setPwm0(direction, pwm), setPwm1(direction, pwm) – методы, используемые для управления двигателями при установленном режиме работы MOTOR_MODE_PWM, 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;

– setParrot0(speed), setParrot1(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);

– beep() – метод, при вызове которого звукоизлучатель издает короткий гудок.

– setServo0(pos), setServo1(pos) – методы установки позиции сервоприводов в условных единицах, где pos – устанавливаемая позиция, может принимать значения от 0 до 255.

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 платой расширения.

5.3 После создания экземпляра класса пользователь получает возможность взаимодействия с платой расширения.

5.4 Пользователь должен включить посылку онлайн-меток, иначе спустя три секунды после начала работы, плата расширения перейдет в спящий режим.

6 ВХОДНЫЕ ДАННЫЕ

6.1 Основными входными данными являются:

– адрес платы расширения на шине I2C;

– режим управления бесколлекторными двигателями;

– значения скоростей двигателей при их управлении;

– значения позиций сервоприводов;

– режим работы звукоизлучателя.

7 ВЫХОДНЫЕ ДАННЫЕ

7.1 Выходными данными являются данные, отправляемые на плату расширения.

Лист регистрации изменений

зм.	Номера листов (страниц)				Всего листов (страниц) в документе	Номер документа	Входящий номер сопроводительного документа и дата	Подпись	Дата
	измененных	замененных	новых	аннулированных					