



Программируемый контроллер ВЕРТОР СТАНДАРТ V2.7

Артикул ВЭЛ10.5586

Технические данные и краткое руководство пользователя

1. Назначение устройства

Программируемый контроллер «Вертор Стандарт» (Рис. 1.1) представляет собой решение для изучения принципов создания систем управления различного назначения на основе модульной электроники. Благодаря наличию разъемов к контроллеру могут подключаться с помощью кабелей датчики и исполнительные устройства исключительно в виде электронных модулей. Это позволяет сделать процесс кон-

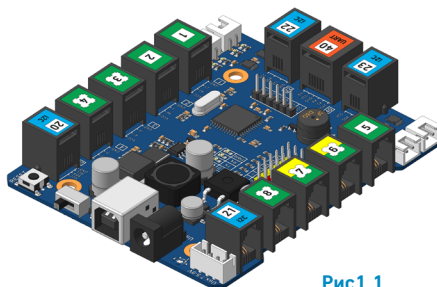


Рис.1.1

струирования электронных систем управления в рамках ведения проектной деятельности быстрым и безошибочным. Поэтому устройство ориентировано прежде всего на создание учебных моделей роботов и средств автоматизации, поскольку помогает сосредоточиться в большей степени на программировании и сборке механических конструкций в процессе выполнения образовательных проектов.

Контроллер «Вертор Стандарт» разработан и произведен в рамках образовательного направления «Эвольтектор Умные Контроллеры», входящего в концепцию «Эвольтектор Умный Мир».

2. Конструкция контроллера и назначение выводов (контактов)

Контроллер выполнен в виде печатной платы, которая оснащена набором функциональных элементов, показанных на Рис. 2.1.

На рисунке 2.2 представлен внешний вид контроллера с указанием габаритных размеров, а также расположение и размеры крепежных отверстий.

Контроллер «Вертор Стандарт» оснащен отверстиями под крепеж М4, с помощью которого можно крепить контроллер к деталям конструктора «Эвольтектор», а также к деталям конструкторов Lego и Makeblock. Межосевые расстояния этих отверстий кратны шагу 8 мм, также как у деталей указанных конструкторов.

На плате присутствует множество разъемов для подключения электронных модулей, приводов и питания от универсального аккумуляторного источника питания. Так, на контроллере представлены:

- 4 разъема для подключения электромоторов в желтом пластиковом корпусе с напряжением



Рис.2.1

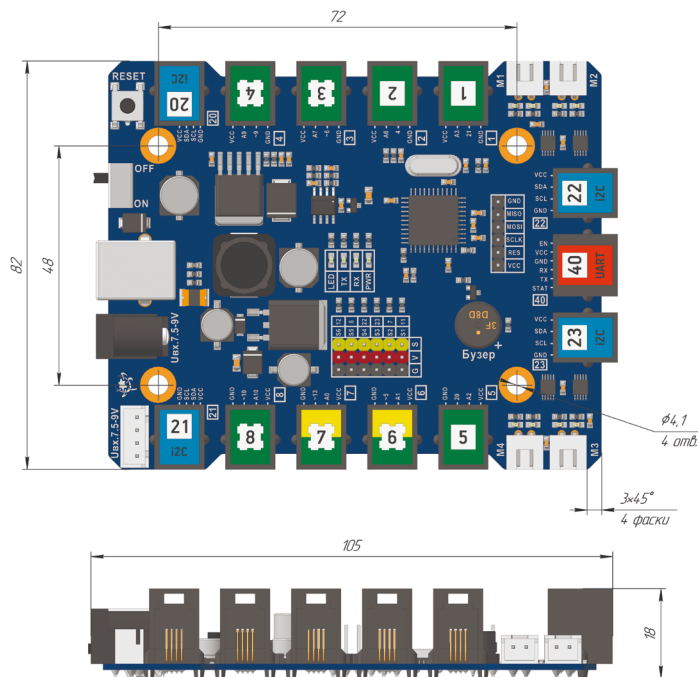


Рис.2.2



питания 6 В (обозначены на плате M1, M2, M3, M4);

- 6 разъемов для подключения стандартных серводвигателей с напряжением питания 6 вольт (допускается общий ток всех сервоприводов и двигателей не более 5 А);
- 4 разъема с интерфейсом I2C для подключения модулей с данным интерфейсом (имеют номера 20, 21, 22, 23);
- 1 разъем с интерфейсом UART (номер 40), который может использоваться для коммуникации с одноплатным миникомпьютером Raspberry Pi через специальную интерфейсную плату или с Bluetooth-модулем «Вертор»;
- 8 разъемов общего назначения для подключения различных электронных модулей системы «Вертор» (пронумерованы цифрами от 1 до 8).

Дополнительных пояснений требуют особенности разъемов общего назначения. К данным разъемам могут подключаться как цифровые, так и аналоговые модули системы электроники «Вертор». Поэтому в каждом из разъемов присутствует по 4 контакта - два отвечают за питание, а остальные предназначены для передачи сигналов. При этом 1 сигнальный контакт является цифровым, а другой аналоговым. Таким образом, если подключается аналоговый модуль, то передача сигналов идет по аналоговому контакту в разьеме. При подключении цифрового модуля сигналы передаются по цифровому контакту.

Микроконтроллер Ардуино Леонардо устроен так, что большая часть его сигнальных линий могут работать в цифровом и аналоговом режимах. В связи с этим в ряде разъемов (портов) контроллера «Вертор Стандарт» к цифровому и аналоговому контакту разъема подключена одна и та же линия микроконтроллера. У таких разъемов наклейка целиком заполнена зеленым цветом (порты под номерами 1, 2, 3, 4, 5, 8). К таким разъемам подключаются одноканальные модули, у которых сигнал передается только через один контакт.

Разъемы под номерами 6 и 7 имеют отдельные цифровую и аналоговую линии. Они позволяют подключать модули, сигналы от которых передаются по двум каналам, то есть через два независимых контакта (к такому классу модулей относится ультразвуковой дальномер). Поэтому наклейки на данных портах выполнены двухцветными, а именно желто-зелеными.

Также следует обратить внимание на рамочки, в которых расположены номера портов на разъемах общего назначения. Если рамка имеет ступенчатые границы, то это означает, что цифровой контакт в этом разьеме имеет аппаратную поддержку генерации сигнала с широтно-импульсной модуляцией (сокращенно ШИМ). Если рамка простая квадратная, то поддержка ШИМ отсутствует. Поэтому модули, которые требуют управления с помощью ШИМ-сигналов (например, светодиоды), следует подключать только к разъемам со ступенчатой рамкой.

3. Принцип работы с контроллером

Контроллер «Вертор Стандарт» является одним из ключевых элементов системы модульной электроники «Вертор 2.0». Благодаря относительно мощной системе питания (допускается суммарный ток потребления до 5 А), большому количеству разъемов и удобному креплению к



механическим конструкциям он прекрасно подходит для создания робототехнических моделей начального уровня, а также для конструирования моделей автоматических производственных линий, моделей умных домов, умных городов и других образовательных проектов.

Для программирования контроллера при выполнении проектов наилучшим образом подходит графическая среда программирования «Студия Эвольвектор» (доступ к онлайн версии среды по ссылке www.studio.evolvektor.ru). В ней модули задаются в виде отдельных графических блоков с изображением модулей и удобно указываются номера портов контроллера, к которым модули подключаются. И уже после программа переводится в программный код для среды Arduino IDE и с ее помощью загружается в контроллер.

Также создание программ для контроллера возможно и непосредственно в Arduino IDE. В этом случае в программе для модулей указываются уже номера контактов, через которые они подключены к контроллеру. Обозначения данных контактов нанесены на плате контроллера рядом с разъемами.

Общий порядок работы с контроллером.

А. Сборка устройства или модели по вашему проекту (на рисунке 3.1 в качестве примера такого устройства показана модель робота-снегоуборщика).

Б. Подключение контроллера к компьютеру с помощью кабеля USB (на контроллере используется разъем USB тип B). Если на ПК установлена операционная система Windows, то необходимо убедиться, что контроллер автоматически распознан системой и для него установились драйвера. В случае, если этого не произошло (данная ситуация может возникнуть в Windows 7 и более ранних версиях) — установите драйвер вручную, скачав его по ссылке academy.evolvektor.ru/vertordrivers. В операционных системах Linux установка драйвера не требуется.

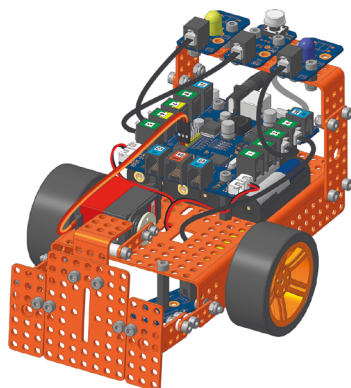


Рис.3.1

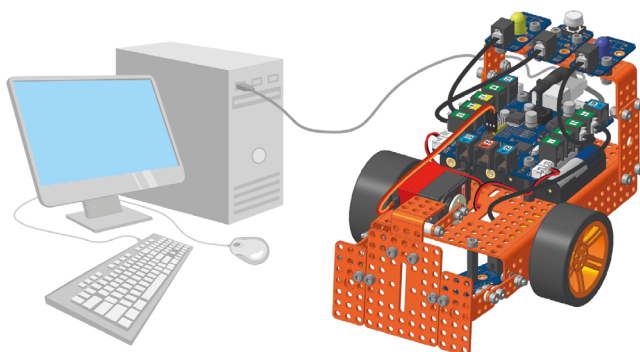


Рис.3.2



В. Создание и загрузка в контроллер программы, которая отражает алгоритм обработки входных сигналов и порядок генерирования выходных для работы проекта. Указанная программа может быть создана в графической среде программирования «Студия Эвольвектор», после чего она автоматически переводится в программный код и загружается в контроллер через программное обеспечение Arduino IDE. Либо же программный код может быть сразу написан в Arduino IDE.

Г. Отключение контроллера от ПК, подключение питания и проверка модели в работе. Если что-то не работает или работает некорректно, то произведите доработку конструкции или программы.

Более подробно о подключении к контроллеру «Верто Стандарт» различных электронных модулей и особенностях использования платы в проектах можно узнать из образовательных курсов к конструкторам «Эвольвектор» из серии «Умный мир», где используется данный контроллер.

```

1 //*****Среда разработки*****
2 #include <MotorVS.h> // Подключение библиотеки <MotorVS.h> для работы с моторами
3 #include <Servo.h> // Подключение библиотеки <Servo.h> для управления сервоприводами
4
5 #define KOVSH_UP 0 // Назначение угла поворота вала сервопривода, при котором ковш поднимат
6 #define KOVSH_DOWN 10 // Назначение угла поворота вала сервопривода, при котором ковш опуститс
7
8 #define LEFT_LED 6 // Назначение пина, к которому подключен левый светодиод
9 #define RIGHT_LED 5 // Назначение пина, к которому подключен правый светодиод
10 #define BUTTON_START 20 // Назначение пина, к которому подключена тактовая кнопка запуска
11 #define LINE_SENSOR 9 // Назначение пина, к которому подключен датчик линии
12
13 bool robot_state=false; // Б.1 - Объявление переменной robot_state и присвоение ей значения false
14
15 int angle_was=KOVSH_UP;
16
17 int speed=100; // Б.1 - Объявление переменной speed и присвоение ей значения 100
18 int counter=0; // Б.1 - Объявление переменной counter и присвоение ей значения 0
19 int timer=0; // Б.1 - Объявление переменной timer и присвоение ей значения 0
20
21 void button_read(); // Объявление функции button_read()
22 void zapusk(); // Объявление функции zapusk()
23
24 MotorVS I2C motor_right = MotorVS_I2C(4); // Инициализация мотора и присвоение ему имени motor_right
25 MotorVS_I2C motor_left = MotorVS_I2C(1,false); // Инициализация мотора и присвоение ему имени motor_left
26
27 Servo kovsh; // Инициализация сервопривода и присвоение ему имени kovsh
28
29 void setup() {
30   motor_right.motorStop(); // Б.2 - Команда на остановку вращения мотора motor_right
31   motor_left.motorStop(); // Б.2 - Команда на остановку вращения мотора motor_left
32
33   kovsh.attach(11); // Указание номера штыря, к которому подключен сервопривод kovsh
34   kovsh.write(KOVSH_UP); // Б.3 - Подача ковшу
35
36   pinMode(LEFT_LED,OUTPUT); // Настройка пина LEFT_LED на вывод
37   pinMode(RIGHT_LED,OUTPUT); // Настройка пина RIGHT_LED на вывод
38   pinMode(BUTTON_START,INPUT); // Настройка пина BUTTON_START на ввод
39   pinMode(LINE_SENSOR,INPUT); // Настройка пина LINE_SENSOR на ввод
40 }
41
42 void loop() {
43   button_read(); // Б.4 - Вызов на исполнение функции button_read
44   if(robot_state=true) // Б.5 - Если значение переменной robot_state равно true, то начать выполнение инструкций в скобках
45   {
46     zapusk(); // Б.6 - Вызов на исполнение функции zapusk
47     while(robot_state=true) // Б.7 - Выполнение инструкций в скобках пока значение переменной robot_state равно true
48     {
49       if(digitalRead(LINE_SENSOR)==false) // Б.8, Б.9 - Если под датчиком линии черная линия, то
50       {
51         motor_right.motorStop(); // Б.10 - Команда на остановку вращения мотора motor_right
52         motor_left.motorStop(); // Б.10 - Команда на остановку вращения мотора motor_left
53         kovsh.write(KOVSH_UP); // Б.11 - Команда на плавный поворот вала сервопривода kovsh в положение KOVSH_UP градусов
54         delay(100); // Б.12 - Ожидание 0,4 секунды
55       }
56     }
57 }

```

Рис.3.3



4. Технические характеристики

Наименование характеристики	Значение
Размеры контроллера, мм	82x105
Количество разъемов общего назначения	8 шт.
Количество разъемов с поддержкой ШИМ	5 шт.
Количество разъемов с аналоговым контактом	8 шт.
Количество разъемов с интерфейсом i2c	4 шт.
Количество разъемов с интерфейсом UART	1 шт.
Количество подключаемых серводвигателей	6 шт.
Количество подключаемых моторов	4 шт.
Тип USB разъема	Тип B
Размеры разъема питания	5,5x2,1 мм
Наличие разъема питания от аккумуляторного модуля	Да
Наличие выключателя питания	Да
Наличие звукового пьезоизлучателя (контакт MISO)	Да
Допустимый диапазон входного напряжения питания	7,5...9 В
Номинальное рабочее напряжение микроконтроллера	5 В
Максимальный суммарный ток потребления устройств, подключенных к контроллеру	до 5 А
Тактовая частота микроконтроллера	16 МГц
Оперативная память	2 КБ
Встроенная Флеш-память	32 КБ
Программное обеспечение для программирования контроллера	«Студия Эвольвектор», Arduino IDE

5. Условия гарантии

ООО «Эвольвектор» гарантирует работоспособность контроллера на протяжении 12 месяцев с момента приобретения устройства, при условии, что плата в течение указанного срока не получила механических повреждений и не выведена из строя в результате ошибочных действий пользователя.

По вопросам гарантийного обслуживания, а также по всем техническим и информационным вопросам можно обращаться на электронную почту info@evolvector.ru, help@evolvector.ru, а также по телефону +7 (499) 391-01-05. Адрес для корреспонденции: 143300, Московская область, г. Наро-Фоминск, ул. Московская, д.15.