



Программируемый контроллер ВЕРТОР JS V1.1

Артикул ВЭЛ10.1680

Технические данные и краткое руководство пользователя

1. Назначение устройства

Программируемый контроллер «Вертор JS» (Рис. 1.1) предназначен для выполнения различных учебных проектов в области электроники, автоматики и робототехники. В рамках стандартных функциональных возможностей контроллер может принимать сигналы от датчиков, обрабатывать их и формировать управляющие сигналы для исполнительных устройств согласно загруженной в него программе. Применение устройства для создания систем управления с мощными исполнительными модулями возможно только при совместном использовании с платами расширения. В качестве таких плат могут использоваться платы расширения для платформы Ардуино, поскольку они совместимы как по присоединительным размерам, так и по сигналам.

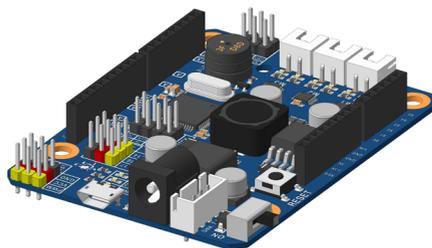


Рис.1.1

Благодаря более производительному 32-битному микроконтроллеру по сравнению 8-битными микроконтроллерами, применяющихся в контроллерах Ардуино, контроллер «Вертор JS» может использоваться для более сложных проектов, требующих более серьезных вычислительных возможностей. А в связи с тем, что программируется контроллер на языке JavaScript, создание программ для проектов в ряде случаев выполняется проще и понятнее, чем на си-подобном языке (используется для программирования Ардуино).

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

Контроллер «Вертор JS» разработан и произведен в рамках образовательного направления «Эвольвектор Умные Контроллеры», входящего в концепцию «Эвольвектор 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.

- ① Выключатель внешнего питания;
- ② Разъем питания от аккумуляторного модуля питания системы Вертор 2.0;
- ③ Разъем питания под штекер размером 5,5x2,1 мм для подачи питания с напряжением 7,5-9 В от батарейного отсека или сетевого блока питания;



- 4 Разъем Micro-USB для загрузки в контроллер управляющих программ с компьютера;
- 5 Светодиодные индикаторы режимов работы контроллера;
- 6 Штыревые контакты для подключения 3-х стандартных хобби-серводвигателей с суммарным токопотреблением 3 А (в среде программирования Espruino IDE имена управляющих контактов MS1, MS2, MS3);
- 7 Дополнительные штыревые контакты для работы с макетной платой;
- 8 Первая группа колодок с гнездовыми контактами для подключения собираемых на макетной плате устройств;
- 9 Вторая группа колодок с гнездовыми контактами для подключения собираемых на макетной плате устройств;
- 10 Встроенный звукоизлучатель (имя управляющего контакта BUZZER в Espruino IDE);
- 11 Штыревые контакты RES, SCK, MISO, MOSI, VCC, GND для подключения устройств работающих по интерфейсу SPI;
- 12 Разъемы M1 – M3 для подключения моторов с токопотреблением до 1 А каждый;
- 13 Отверстия, соответствующие оригинальной плате Arduino Leonardo;
- 14 Отверстия под крепеж M4 для монтажа на детали Эвольвектор, Lego, MakeBlock;
- 15 Штыревые контакты для прошивки контроллера с помощью программатора;
- 16 Кнопка принудительной перезагрузки контроллера.

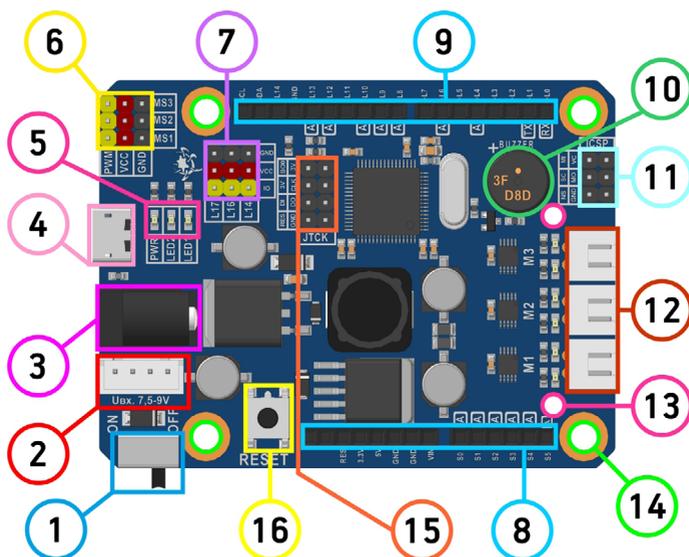


Рис.2.1

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

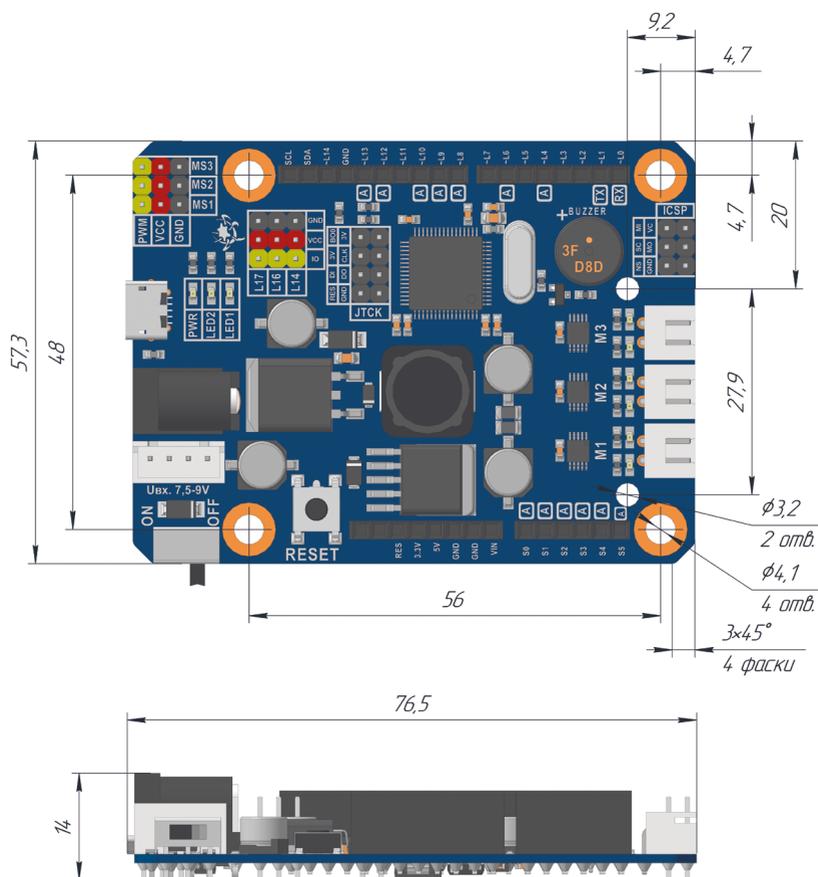


Рис.2.2

Все доступные пользователю выходы в колдовках и дополнительной группе штырей делятся на цифровые и аналоговые. Цифровые выходы пронумерованы с ~L0 до L17, а аналоговые от S0 до S5. При этом все аналоговые выходы могут использоваться в качестве цифровых. Цифровые контакты ~L4, ~L6, ~L8, ~L9, ~L10, ~L12 и ~L13 также могут использоваться в качестве аналоговых выводов. Все цифровые выходы, отмеченные знаком «~», имеют поддержку аппаратной широтно-импульсной модуляции.

Помимо этого, присутствуют контакты питания 5V; питание 3.3V; «земля». Также плата оснащена двумя встроенными светодиодами (имена управляющих контактов LED1 и LED2 в Espruino IDE); выводами SCL, SDA для подключения устройств, обменивающихся данными по интерфейсу I2C; выводом RES для программного сброса контроллера.



3. Принцип работы с контроллером

Контроллер «Вертор JS» построен на основе платформы Espruino. Она включает в себя микроконтроллер, поддерживающий платформу (в нашем случае это STM32F405RGT6), загруженный в него интерпретатор Espruino Firmware, и среду программирования Espruino Web IDE.

Соответственно, в среде программирования на языке JavaScript создаются программы для ваших проектов, после чего через нее же загружаются в контроллер «Вертор JS», и далее уже микроконтроллер с помощью интерпретатора их исполняет.

При этом, общие принципы работы с контроллером «Вертор JS» ничем не отличаются от таковых с другими программируемыми контроллерами. Он может принимать сигналы с датчиков (как аналоговые, так и цифровые), после чего обрабатывать их и формировать уже управляющие сигналы для исполнительных устройств (индикаторов, двигателей и т.д.) в соответствии с алгоритмом вашей программы.

В качестве датчиков могут использоваться любые сенсорные электронные компоненты или электронные модули системы Вертор 2.0, которые выдают аналоговые или цифровые сигналы с рабочим уровнем напряжения 3,3-5 В (подключаются через плату расширения «Разъемы Вертор 2.0»).

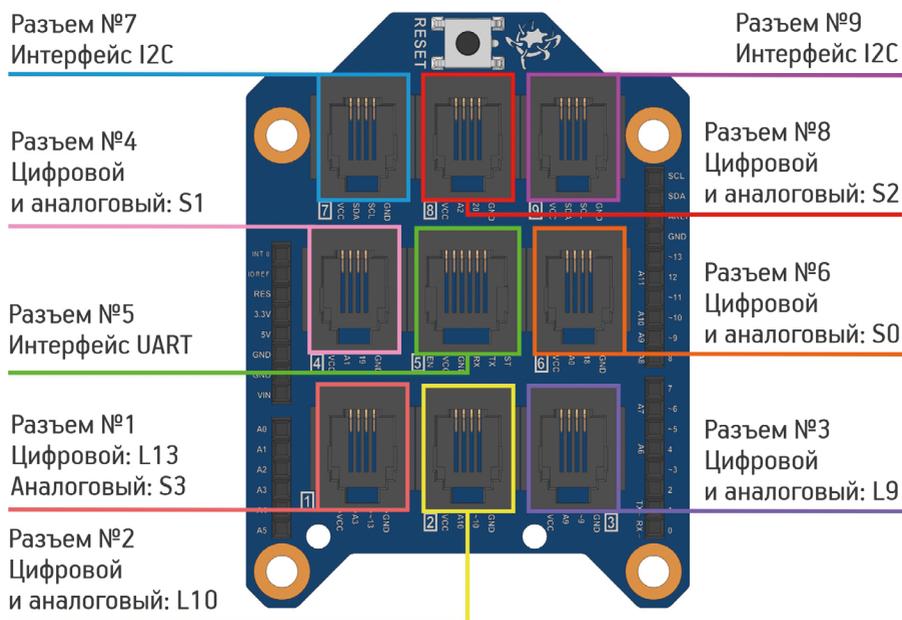


Рис.3.1



Исполнительными устройствами могут выступать индикаторы, экраны или электроприводы. При этом рабочий уровень напряжения выходных сигналов на выводах контроллера для управления исполнительными элементами составляет только 5 В. Допускается суммарное токопотребление всех подключенных электронных модулей не более 1,3 А, а моторов и сервоприводов не более 3 А. Для управления периферией с более высоким потребляемым током необходимо (!!!) использовать вспомогательные модули (драйверы), иначе контроллер может выйти из строя.

Датчики и исполнительные устройства подключаются к контроллеру через контактные колодки с помощью макетной платы и проводов с соблюдением правил схемотехники и допустимых нагрузок на линии управления модулями или приводами. Если необходимо подключить модули системы ВЕРТОР 2.0 с разъемами, то удобно для этого использовать плату расширения «Разъемы Вертор 2.0» (на рисунке 3.1 показано соответствие контактов контроллера разъемам на данной плате).

При подключении моторов и серводвигателей рекомендуется использовать плату расширения «Привод Классик». На рисунке 3.2 показано соответствие контактов контроллера элементам данной платы.

Сервопривод №2 Управляющая линии: L8

Сервопривод №1
Управляющая
линии: S4

Сервопривод №3
Управляющая
линии: L12

Мотор №2 (M2)
Управляющие
линии: L6, L7

Мотор №3 (M3)
Управляющие
линии: S5, L11

Мотор №1 (M1)
Управляющие
линии: L4, L5

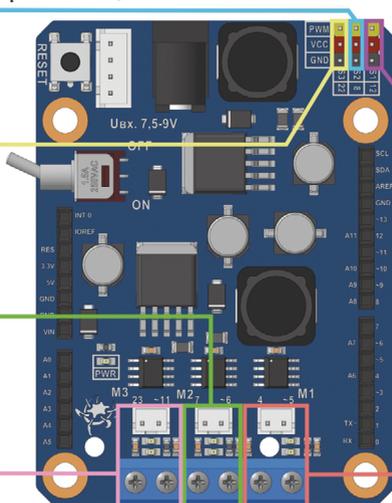


Рис.3.2



4. Выполнение проектов с использованием контроллера «Вертор JS»

Разумеется, каждый проект всегда начинается с определения (проектирования) конструкции будущего устройства и его составных частей. После этого выполняется его сборка с помощью проводов и макетной платы, если устройство состоит из электронных компонентов, и с помощью кабелей и плат расширения, если устройство имеет модульную конструкцию (Рис. 4.1).

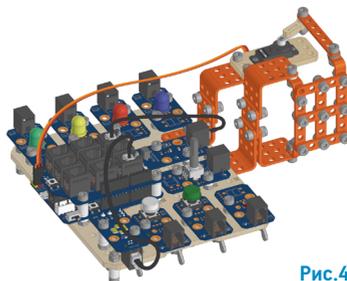


Рис.4.1.

Далее составляется и загружается в контроллер программа, которая задает алгоритм работы собранного устройства. Сделать это можно двумя способами: на языке JavaScript в среде программирования Espruino IDE и с помощью блочно-графической среды программирования «Студия Эвольвектор».

Если принято решение об использовании среды Espruino IDE и она не установлена на ваш компьютер, то ее можно скачать с сайта Академии Эвольвектор из раздела «Скачать» подраздел [«Программное обеспечение и библиотеки для работы с контроллером «Вертор JS»](#). После распаковки скачанного архива необходимо выбрать версию Espruino IDE, соответствующую характеристикам вашей операционной системы Windows (32 или 64 бита), установить и запустить ее. Откроется диалоговое окно, в котором справа будет поле для написания программы на языке JavaScript, а слева общая информация о среде (Рис. 4.2).



Рис.4.2.



Для работы с электронными модулями системы Вертор 2.0 понадобятся библиотеки. Они представлены на той же страничке Академии Эвольвектор. Скачайте их и распакуйте в любую удобную папку, например, в папку «Документы». После чего откройте в Espruino IDE настройки (Setting) и нажмите в верхнем правом углу на значок 

Затем выберете раздел «Project». Нажав на кнопку «Select Directory for Sandbox», укажите в открывшемся окне папку с библиотеками. (Рис. 4.3).

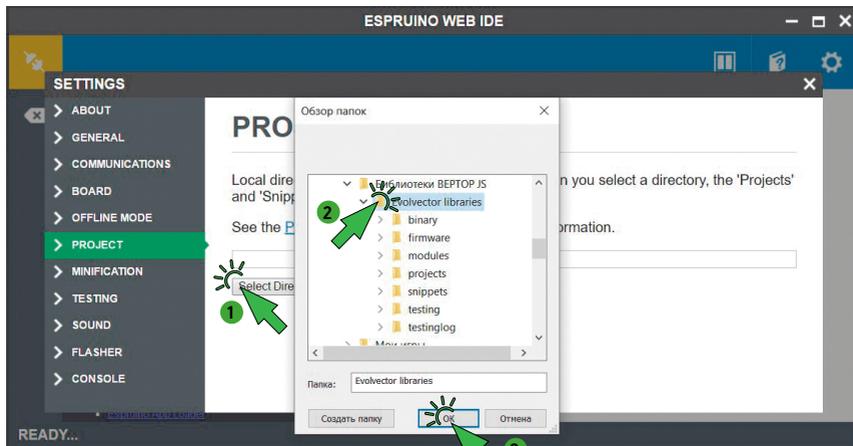


Рис.4.3.

В результате чего в соответствующей строке появится путь к данной папке (Рис. 4.4).

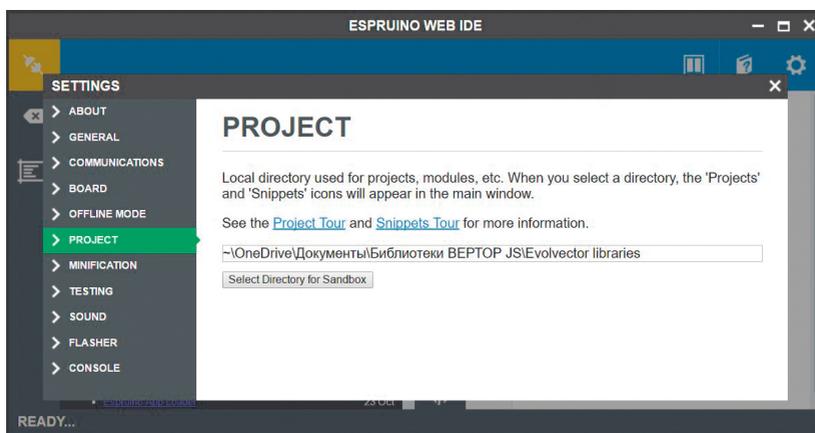


Рис.4.4.



Контроллер Вертор JS подключается к компьютеру с помощью кабеля USB-MicroUSB. После соединения с компьютером подключение необходимо активировать программно. Для этого нажмите на кнопку  находящуюся в верхнем левом углу среды разработки. В появившемся списке выберите Com-порт, к которому подключен контроллер (Рис. 4.5). Под номером этого порта будет указано Espruino Board.

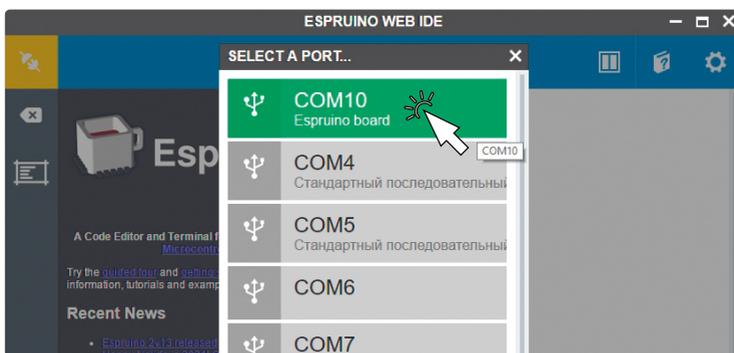


Рис.4.5.

Если все сделано верно, то появится сообщение, что контроллер подключен (Рис. 4.6).

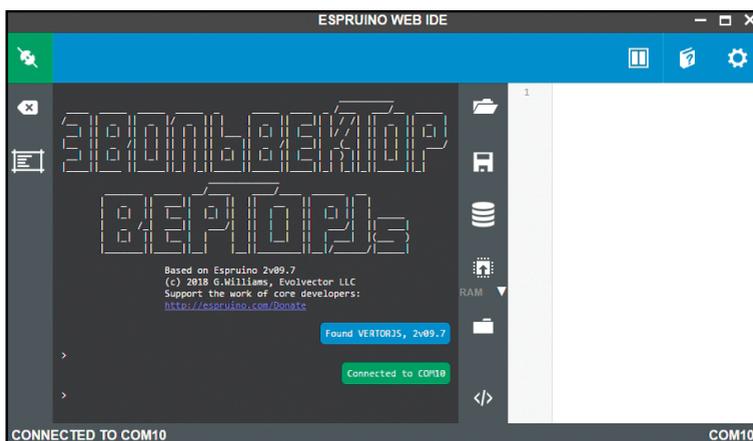


Рис.4.6.

В среде Espruino IDE программы создаются на языке программирования JavaScript в правой части окна среды. Пример такой программы с комментариями показан на рисунке 4.7. Для более удобной работы поле программирования можно расширить, перетаскив вертикальную границу разделения полей влево.



```
1 var LED=L13; // Определение пина, к которому подключен фонарь, и присвоение ему имени LED
2 var BUTTON=L9; // Определение пина, к которому подключена кнопка, и присвоение ему имени BUTTON
3 var STATE = false; // Инициализация логической переменной с именем STATE и присвоение ей значения false
4
5 pinMode(LED, 'output'); // Настройка пина LED на вывод данных
6 pinMode(BUTTON, 'input'); // Настройка пина BUTTON на ввод данных
7
8 clearWatch(); // Удаление ранее созданных условий вызовов функций при определенных изменениях состояний
9 пинов
10 digitalWrite(LED, STATE); // Функция установки на пине LED значения STATE
11
12 function led_toggle() // Определение новой функции и присвоение ей имени led_toggle
13 {
14   STATE=!STATE; // Инвертирование состояния переменной STATE
15   digitalWrite(LED, STATE); // Функция установки на пине LED значения STATE
16 }
17
18 setWatch(led_toggle, BUTTON, {repeat:true, debounce:20, edge:"falling"}); // Отслеживание изменения сигнала на пине
19 // BUTTON с 1 на 0 и выполнение функции led_toggle при выполнении этого условия
// Функция setWatch(function, PIN, {options}) позволяет вызывать на исполнение функцию function при заданном в области
// {options} изменении состояния на пине PIN.
// Область {options} имеет следующую структуру: {repeat:[true/false],debounce:[time], edge:["falling", "rising",
// "rising_falling"]}
```

Рис.4.7.

Загрузка программы в контроллер выполняется путем нажатия на кнопку.



Если нажать на ту часть этой кнопки, которая с треугольником, то можно еще и выбрать параметры выгрузки программы (Рис. 4.8): в оперативную память контроллера (RAM), в энергонезависимую память контроллера с заменой ранее загруженной программы (FLASH), в энергонезависимую память в виде файлов, чтобы потом вызывать их в скетчах (STORAGE).

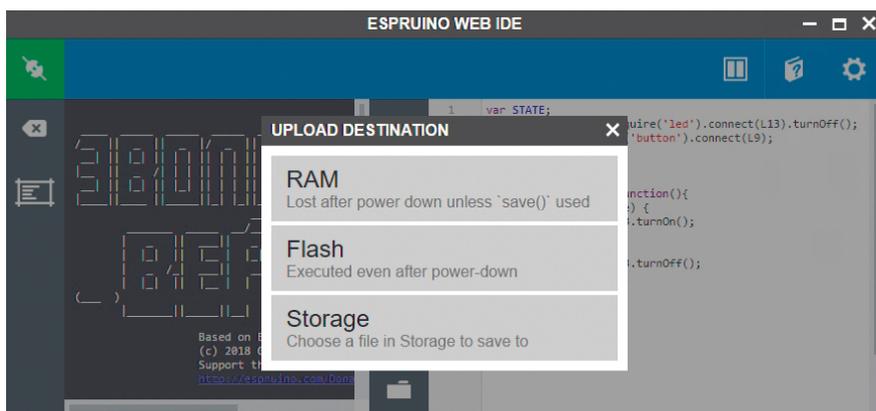


Рис.4.8.

При загрузке программы в оперативную память RAM она будет исполняться контроллером до тех пор, пока не будет выключено питание. Как только контроллер будет обесточен, данные из оперативной памяти будут потеряны.



В случае загрузки программы в постоянную память FLASH, программа будет сохраняться в памяти при отключении и включении питания. При каждой новой загрузке старая программа автоматически удаляется и на ее место записывается новая.

Процесс создания программ можно сильно упростить, если воспользоваться для этого подсистемой Вертор JS в блочно-графической среде программирования «Студия Эвольвектор». В ней программа составляется из наглядных графических элементов с указанием требуемых числовых параметров. Поддерживается прямая загрузка в контроллер из среды с выбором оперативной или энергонезависимой памяти для записи (рис. 4.9)

Загрузка программы в контроллер выполняется путем нажатия на зеленую кнопку с белым треугольником 

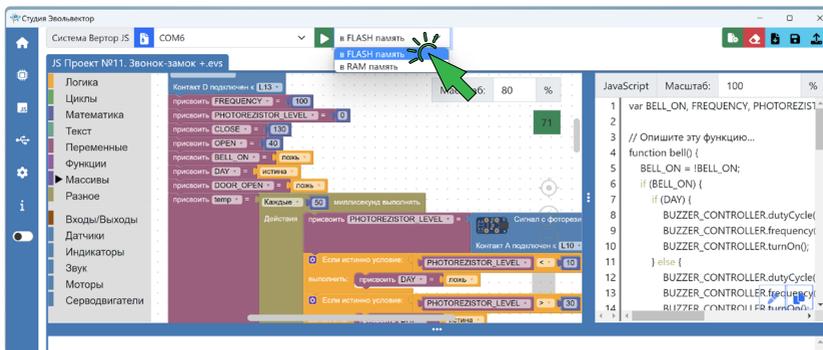


Рис.4.9.

Созданная в Студии программа в виде соединенных между собой блоков в правом поле автоматически преобразуется в программный код JavaScript, который может быть скопирован с помощью кнопки «Скопировать код» и перенесен в Espruino IDE (Рис. 4.10, 4.11).

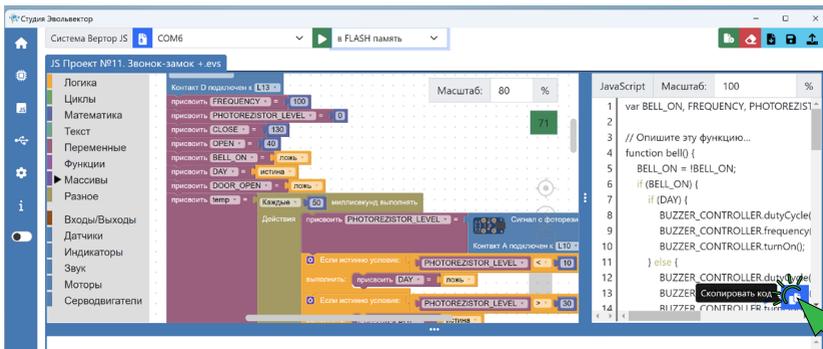


Рис.4.10.

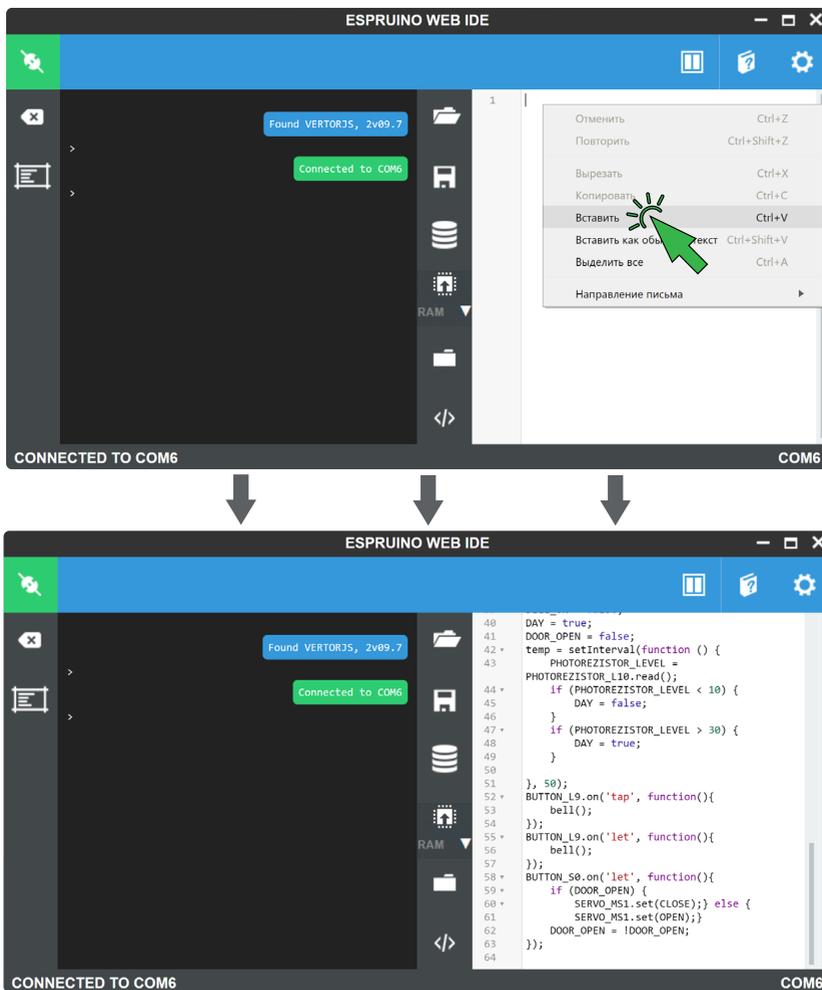


Рис.4.11.

Более подробная информация по работе с контроллером на примере образовательных проектов доступна в учебных материалах, входящих в комплект образовательных конструкторов серии «Умный дом».

**5. Технические характеристики**

Наименование характеристики	Значение
Размеры контроллера	57x76 мм.
Количество цифровых контактов	24
В составе цифровых контактов наличие выводов с поддержкой ШИМ	15
Количество аналоговых выводов	13
Максимальное количество подключаемых серводвигателей (с суммарной нагрузкой не более 3 А)	3
Максимальное количество подключаемых моторов (с суммарной нагрузкой не более 3 А)	3
Размеры разъема питания	5,5x2,1 мм
Тип разъема для подключения к ПК	MicroUSB
Наличие разъема питания от аккумуляторного модуля	Да
Наличие выключателя питания	Да
Наличие звукового пьезоизлучателя	Да
Допустимый диапазон входного напряжения питания	7,5...9 В
Номинальное рабочее напряжение микроконтроллера	5 В
Максимальный суммарный ток потребления модулей и устройств, подключенных к контроллеру	Не более 1,3 А
Тактовая частота микроконтроллера	168 МГц
Оперативная память	192 Кб
Встроенная Флеш-память	1024 Кб (STM32F405RGT6), из которых 330 Кб используются для загрузчика
Программное обеспечение для программирования контроллера	«Студия Эвольвектор», Espruino IDE



6. Условия гарантии

ООО «Эвольвектор» гарантирует работоспособность электронной платы на протяжении всего гарантийного срока эксплуатации, который составляет 6 месяцев с момента приобретения устройства. Также гарантируется совместимость модуля с другими устройствами системы управляющей электроники ВЕРТОР.

Гарантийные обязательства производителя распространяются только на ту продукцию, которая не имеет повреждений и не выведена из строя в результате неверных 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.w