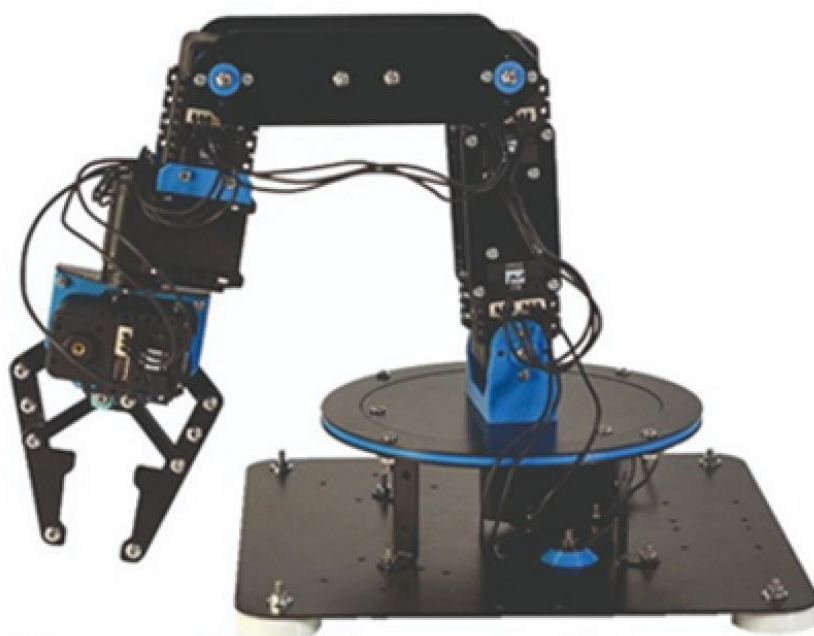




Сделано в России

ROBOTS 1



УЧЕБНОЕ ПОСОБИЕ

Общеобразовательный конструктор для практического изучения принципов создания электронных устройств на основе электронных компонентов и программируемых контроллеров

Hobots 1

Актуальная версия настоящего документа расположена на сайте производителя www.hobots1.ru

Добрый день, дорогие друзья!

Перед вами поурочные разработки к комплекту конструктора для практического изучения принципов создания электронных устройств на основе электронных компонентов и программируемых контроллеров для обучения программированию и работе с данными – HOBOTS 1, который имеет многогранный спектр возможностей. Настоящий учебно-методический комплекс предназначен для проведения учебного процесса в средних, специальных технических и высших учебных заведениях. Данные методические материалы помогут осуществить первоначальную сборку, освоить основы программирования и электроники, а также последовательно изучить аппаратную часть конструктора. Однако, для успешного выполнения работы необходимы инициативная техническая работа, анализ, творчество и навыки программирования.

Данный набор имеет расширенные возможности, а конструктивные особенности модели образовательного конструктора HOBOTS 1 позволяют постоянно его модернизировать и дорабатывать. Составные части, такие как силовые элементы и управляющая электроника могут быть заменены в самых разных комбинациях, а элементы изготовленные из пластика методом 3D печати открыты для изменения и модернизации. Наборы серии HOBOTS 1 представляют собой развитие линейки конструкторов для создания робототехнических устройств и могут быть дополнительно оснащены разнообразными датчиками и модулями.

Набор обеспечивает возможность разработки модели мобильного робота, управляемого посредством программного обеспечения для персонального компьютера и мобильных устройств на базе ОС Android, IOS, обеспечивающего возможность взаимодействия с манипуляционным роботом посредством программного обеспечения, включающим в себя набор органов управления.

Набор обеспечивает возможность изучения основ электроники и схемотехники, разработки и прототипирования моделей роботов, разработки программных и аппаратных комплексов инженерных систем, решений в сфере "Интернет вещей", а также решений в области робототехники, искусственного интеллекта и машинного обучения.

Изменяя отдельные компоненты данного конструктора в различных комбинациях Вы сможете на практике смоделировать все основные процессы в современной робототехнике.

Добро пожаловать в мир робототехники!

Желаем Вам успехов в работе с конструктором «HOBOTS»!

Оглавление

1	Требования по технике безопасности	3
2	Спецификация.....	5
3	Комплект поставки	7
4	Урок №1. Знакомство с контроллерами семейства Arduino	26
5	Урок №2. Установка и настройка Arduino IDE	35
6	Урок №3. Установка и настройка графической системы программирования ArduBlock	40
7	Урок №4. Основная информация по структуре и командам языка программирования C++	45
8	Урок № 5 Сборка манипулятора	52
9	Урок № 6 Подключение манипулятора	57
10	Урок №7. Программное обеспечение ROBOTS	60
11	Урок № 8. Техническое зрение	63

1 Требования по технике безопасности

В этой теме описаны меры предосторожности, на которые следует обратить внимание при использовании конструктора Nobots 1.

Пожалуйста, внимательно прочитайте этот документ перед первым использованием роботизированного манипулятора. Данное изделие необходимо использовать в условиях, соответствующих проектным спецификациям, запрещается вносить изменения в конструктивную и аппаратную часть, это может привести к поломке изделия, травмам, поражению электрическим током, пожару и т.д.

При использовании конструктора следует соблюдать следующие правила безопасности:

- используйте роботизированный манипулятор в указанном диапазоне условий. В противном случае превышение технических характеристик и условий нагрузки приведет к сокращению срока службы изделия и даже к повреждению.

- перед установкой, эксплуатацией и обслуживанием роботизированного манипулятора персонал, ответственный за установку, эксплуатацию и обслуживание, должен соблюдать меры безопасности и правила эксплуатации и обслуживания;

- для очистки роботизированного манипулятора не рекомендуется использовать бытовую химию агрессивного воздействия, рекомендуется использовать влажную мягкую губку или тканевый материал для избежания повреждения покрытия;

- утилизацию проводить в соответствии с действующим законодательством, защищайте окружающую среду;

- в комплекте конструктора Nobots 1 есть мелкие детали, пожалуйста, держите их подальше от детей, чтобы избежать несчастных случаев;

- не позволяйте детям играть с конструктором Nobots 1 в одиночку. Все процессы должны контролироваться во время работы. После завершения процессов, пожалуйста, немедленно выключите оборудование;

- не опускайте руки в рабочую зону роботизированного манипулятора во время работы, чтобы избежать травм;

- будьте осторожны во время переноски или установки роботизированного манипулятора. Пожалуйста, следуйте инструкциям для правильной транспортировки конструктора и его правильного расположения перед выполнением работ;

- при выключении роботизированный манипулятор может медленно перемещаться в определенное положение. НЕ кладите руки в рабочее пространство манипулятора во время работы, чтобы избежать травм;

- если положение Nobots 1, отображаемое на экране программы «Nobots», не соответствует фактическому, пожалуйста, нажмите кнопку сброс на странице «Nobots», чтобы выполнить калибровку;

- во время калибровки, Nobots 1 автоматически займет положение точки наведения;

- запрещается подключать или отключать внешнее оборудование, без полного выключения Nobots 1, иначе это может привести к повреждению устройства;

- запрещается выключать Nobots 1 во время записи прошивки, чтобы избежать повреждения изделия.

- учебный манипулятор должен располагаться на горизонтальной устойчивой поверхности;

- область в зоне работы манипулятора должна быть освобождена от каких-либо посторонних предметов, не предназначенных для работы с учебным манипулятором.

Несмотря на проверенную конструкцию и постоянную модернизацию, конструкторы представляют собой наборы для инженерного и технического творчества, что предполагает работу непосредственно с источниками повышенной опасности: паяльной станцией, припоями, техническими жидкостями, клеями, электрическим током, вращающимися деталями, сервоприводами и рычагами, аккумуляторами и в совокупности – с мобильными роботами. Отдельно стоит отметить обязательную необходимость применения средств защиты при работе с композитными материалами, углепластиковыми, "карбоном" и прочими элементами повышенной опасности. Источники повышенной опасности требуют обязательного проведения инструктажа по соблюдению техники безопасности для учащихся и преподавателя ответственным лицом до начала работы с конструкторами, безусловное соблюдение ее на всем протяжении работы и особенно в режиме автономной работы мобильных роботов с применением машинного зрения.

Долговременная плодотворная эксплуатация конструктора зависит от соблюдения техники безопасности, требования инструкции и выполнения требований преподавателя.

Желаем вам успехов в работе!

2 Спецификация

Спецификация конструктора Robots 1. Таблица 1.

№ п/п	Наименование характеристики	Значение характеристики
1	Комплект конструктивных элементов из металла и пластика для сборки моделей манипуляционных роботов с угловой кинематикой, плоскопараллельной кинематикой, Delta-кинематикой	Наличие
2	Интеллектуальный сервомодуль с интегрированной системой управления	7 шт
2.1	Возможность последовательного подключения друг с другом и управления сервомодулями по последовательному полудуплексному асинхронному интерфейсу	Соответствие
3	Робототехнический контроллер, представляющий собой модульное устройство, включающее в себя одноплатный микрокомпьютер для выполнения сложных вычислительных операций, периферийный контроллер для управления внешними устройствами и плату расширения для подключения внешних устройств	1 шт
3.1	Модули робототехнического контроллера должны обладать одновременной конструктивной, аппаратной и программной совместимостью друг с другом	Соответствие
3.2	Интерфейсы SPI и I2C и 1-wire TTL и UART и PWM	Наличие
3.3	Цифровые порты для подключения внешних устройств	16 шт
3.4	Аналоговые порты для подключения внешних устройств	8 шт
3.5	WiFi и/или Bluetooth для коммуникации с внешними устройствами	Наличие
3.6	Возможность программирования с помощью средств языков C/C++, Python и свободно распространяемой среды Arduino IDE, а также управления моделями робототехнических систем с помощью среды ROS	Соответствие
4	Программируемый контроллер представляющий собой вычислительный модуль	1 шт
5	Плата расширения программируемого контроллера	Наличие
5.1	Возможность подключения универсального вычислительного модуля к сети посредством интерфейса Ethernet	Соответствие
5.2	Плата расширения должна обладать портами ввода-вывода для подключения цифровых и аналоговых устройств	40 шт
5.3	Возможностью подключения внешней карты памяти	Наличие
5.4	Интерфейс SPI	Наличие
6	Модуль технического зрения, представляющий собой устройство на базе вычислительного микроконтроллера и интегрированной камеры, обеспечивающее распознавание простейших изображений на модуле за счет собственных вычислительных возможностей	1 шт
6.1	Модуль технического зрения должен обеспечивать возможность коммуникации с аналогичными модулями посредством шины на базе последовательного интерфейса с целью дальнейшей передачи результатов измерений группы модулей на управляющее вычислительное устройство, подключенное к данной шине	Соответствие
6.2	Модуль технического зрения должен обеспечивать возможность осуществлять настройку модуля технического зрения - настройку экспозиции, баланса белого, цветоразностных составляющих, площади обнаруживаемой области изображения, округлости обнаруживаемой области изображения, положение обнаруживаемых областей относительно друг друга.	Соответствие
6.3	Модуль технического зрения должен обеспечивать возможность настройки на одновременное обнаружение не менее 10 различных одиночных объектов в секторе обзора, а также не менее 5 составных объектов, состоящих из не менее 3 различных графических примитивов.	Соответствие
6.4	Модуль технического зрения должен обладать встроенными интерфейсами: USB и UART, и 1-wire TTL, и I2C, и SPI для коммуникации со внешними подключаемыми устройствами.	Соответствие

7	Цифровые информационно-сенсорные модули, представляющие собой устройства на базе программируемого контроллера и измерительного элемента	Соответствие
7.1	Интерфейсы для подключения к внешним устройствам: цифровые и аналоговые порты, и 1-wire TTL, и разъем типа RJ	Наличие
7.2	Цифровой модуль должен обеспечивать возможность коммуникации с аналогичными модулями посредством шины на базе последовательного интерфейса с целью дальнейшей передачи результатов измерений группы модулей на управляющее вычислительное устройство, подключенное к данной шине.	Соответствие
7.3	Цифровой модуль тактовой кнопки	3 шт
7.4	Цифровой модуль светодиода	3 шт
7.5	Цифровой модуль концевого прерывателя	3 шт
7.6	Цифровой модуль датчика цвета	1 шт
7.7	Цифровой модуль RGB светодиода	1 шт
8	Элементы для сборки вакуумного захвата	Соответствие
8.1	Вакуумная присоска	1 шт
8.2	Электромагнитный клапан	1 шт
8.3	Вакуумный насос	1 шт
8.4	Соединительный шланг	Наличие
9	Набор библиотек трехмерных элементов для прототипирования моделей манипуляционных роботов	Наличие
9.1	Учебное пособие которое должно содержать материалы по разработке трехмерных моделей мобильных роботов, манипуляционных роботов с различными типами кинематики (угловая кинематика, плоско-параллельная кинематика, дельта-кинематика, SCARA, рычажная кинематика, платформа Стюарта и т.п.), инструкции по проектированию роботов, инструкции и методики осуществления инженерных расчетов при проектировании (расчеты нагрузки и моментов, расчет мощности приводов, расчет параметров кинематики и т.п.), инструкции по разработке систем управления и программного обеспечения для управления роботами, инструкции и методики по разработке систем управления с элементами искусственного интеллекта и машинного обучения	Наличие
9.2	Возможность трехмерной визуализации модели манипуляционного робота (с угловой, плоскопараллельной и дельта-кинематикой) в процессе работы, обеспечивать построение пространственной траектории движения исполнительного механизма манипуляционного робота, возможность задания последовательности точек для прохождения через них исполнительного механизма манипуляционного робота. Возможность построения графиков заданных и текущих обобщенных координат манипуляционного робота, графиков значений скоростей и ускорения, графиков расчетных значений нагрузки. Возможность задавать последовательность передвижений манипулятора посредством набора команд в блочно-графическом интерфейсе.	Наличие

3 Комплект поставки

Комплект поставки конструктора Nobots 1 включает в себя:

- конструктивные элементы из металла и пластика;
- интеллектуальный сервомодуль с интегрированной системой управления;
- робототехнический контроллер;
- программируемый контроллер;
- модуль технического зрения;
- цифровые информационно-сенсорные модули;
- элементы для сборки вакуумного захвата
- комплект крепежа;
- набор проводов;
- блок питания;
- кабель USB;
- набор библиотек трехмерных элементов;

Конструктивные элементы из металла и пластика

Общеобразовательный конструктор для практического изучения принципов создания электронных устройств на основе электронных компонентов и программируемых контроллеров Hobots 1 содержит конструктивные элементы из пластика, выполненных с помощью 3D печати (Рисунок 3-1) и металла изготовленных путем лазерной ЧПУ резки (Рисунок 3-2).



Рисунок 3-1 Конструктивные элементы из пластика



Рисунок 3-2 Конструктивные элементы из металла

Интеллектуальный сервомодуль с интегрированной системой управления

Общеобразовательный конструктор комплектуется интеллектуальными сервоприводами dynamixel ax-12a (Рисунок 3-3).

Сервомодуль представляет собой единый электромеханический модуль, включающий в себя привод на базе двигателя постоянного тока, понижающий редуктор, встроенную систему управления. Сервомодуль обладает интегрированной системой управления, обеспечивающей обратную связь, контроль параметров - положение вала, скорость вращения, нагрузка привода, а также обеспечивающей возможность последовательного подключения друг с другом и управления сервомодулями по последовательному полудуплексному асинхронному интерфейсу. Имеется режим постоянного вращения выходного вала.

Технические характеристики привода:

Нижняя граница диапазона допустимого напряжения питания - 9 В.
Верхняя граница диапазона допустимого напряжения питания - 12 В.
Передаточное отношение редуктора - 254. Максимальный момент - 1,5 Н*м.
Нижняя граница диапазона номинальной скорости вращения в режиме постоянного вращения - 0 об/мин. Верхняя граница диапазона номинальной скорости вращения в режиме постоянного вращения - 59 об/мин.
Максимальная величина угла поворота в режиме позиционного управления - 300 угловых градусов. Разрешающая способность - 0,29 угловых градусов.
Размеры сервомодуля (ДхШхВ) - 32х50х40 мм (Рисунок 3-4).



Рисунок 3-3 Сервомодуль DYNAMIXEL AX-12A

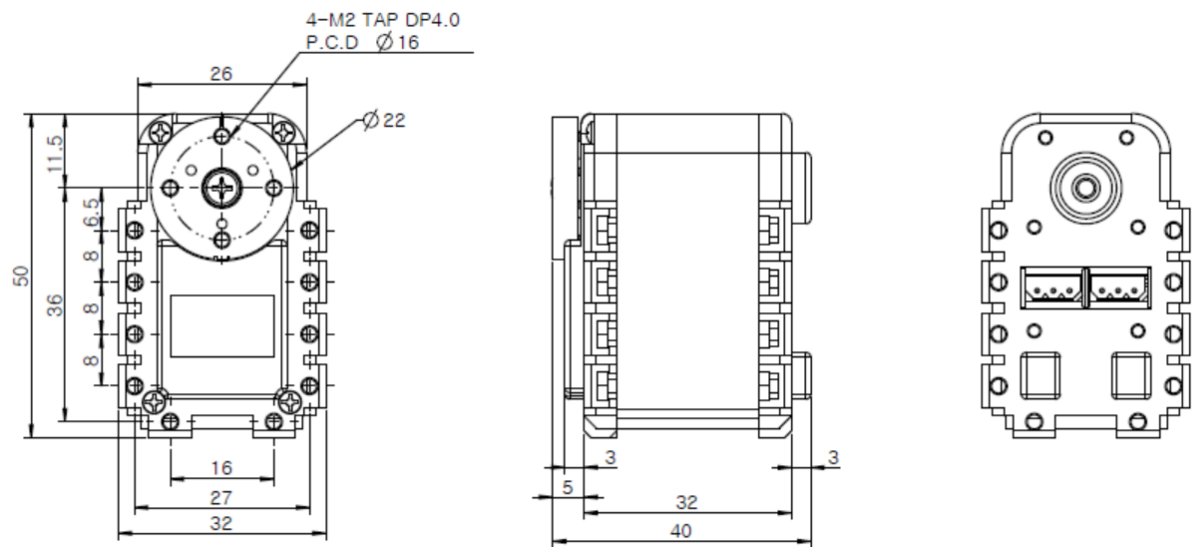


Рисунок 3-4 Размеры сервомодуля DYNAMIXEL AX-12A

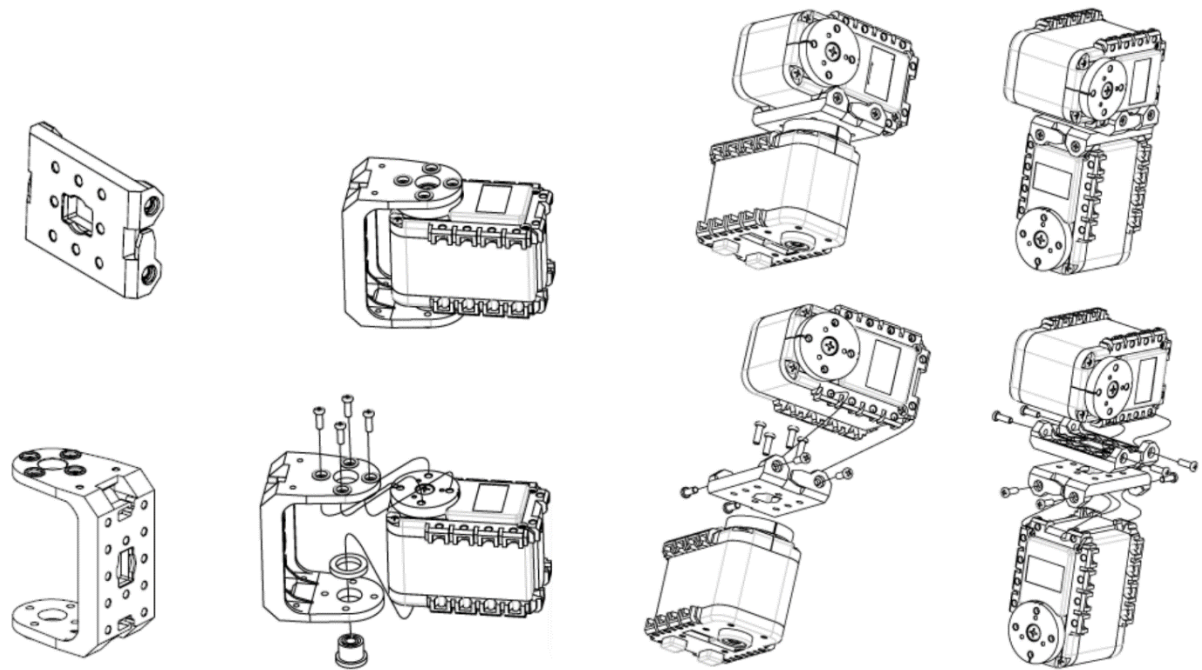


Рисунок 3-5 Варианты сборки сервоприводов

Робототехнический контроллер

Робототехнический контроллер, представляет собой модульное устройство, включающее в себя одноплатный микрокомпьютер для выполнения сложных вычислительных операций, периферийный контроллер для управления внешними устройствами и плату расширения для подключения внешних устройств. Модули обладают конструктивной, аппаратной и программной совместимостью. Обеспечивает возможность программирования с помощью средств языков C/C++, Python и свободно распространяемой среды Arduino IDE, а также управления моделями робототехнических систем с помощью среды ROS.

Периферийный контроллер выполнен на базе управляющей платы ATMEGA 2560 (Рисунок 3-6) предназначенной для сбора, преобразования, обработки, хранения информации и выработки команд управления, имеющий конечное количество входов и выходов, подключенных к ним датчиков, ключей, исполнительных механизмов и предназначенный для работы в режимах реального времени. Программируемый контроллер, это на сегодняшний день относительно небольшой компьютер в промышленном исполнении, управляемый микропроцессором, адаптированный под нужды решения задач автоматизации в режиме реального времени, с максимально коротким откликом.

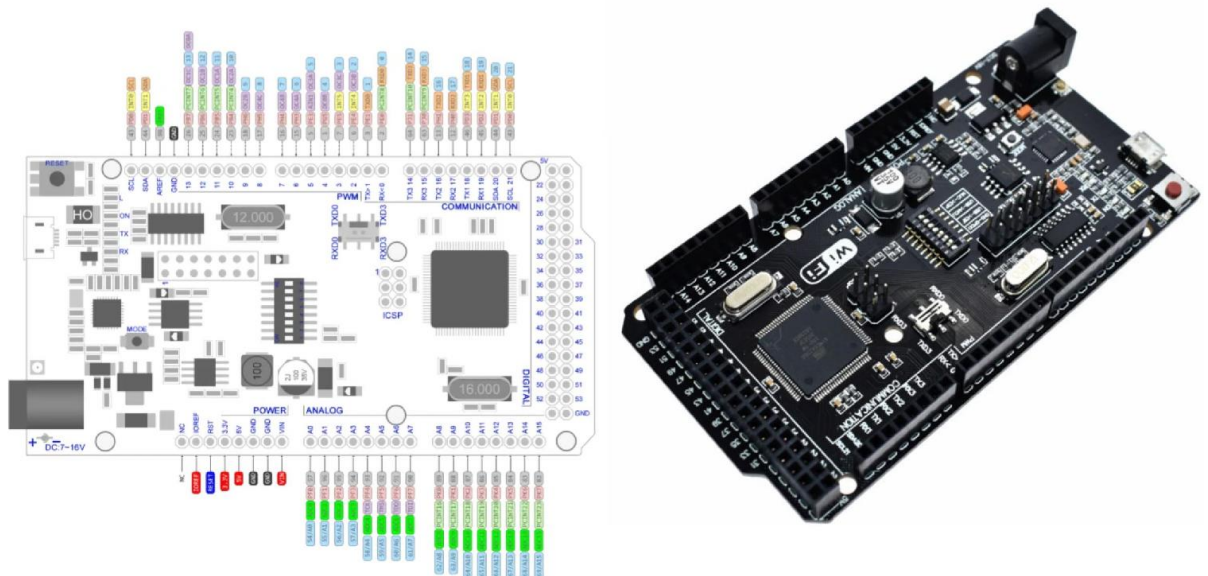


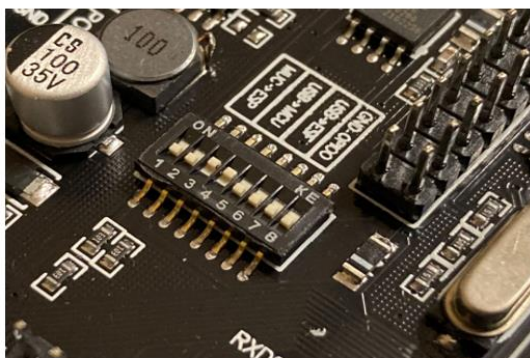
Рисунок 3-6 Управляющая плата программируемого контроллера

Для связи с окружающей средой программируемый контроллер оборудован входными периферийными устройствами (входами), на которые подаются сигналы от управляемого процесса, дискретные сигналы в виде состояния включения/выключения (например, определение положения

конечным датчиком) или непрерывные аналоговые сигналы с датчиков (например, значения температуры, давления, уровня).

На «противоположной» стороне программируемый контроллер имеет выходные периферийные устройства (выходы), к которым подключены элементы управляющие автоматизируемым процессом, опять же в виде дискретного сигнала управления вкл/выкл (например, электромагнитное реле, контактор двигателя, катушка клапана, сигнальная лампа) или в виде непрерывного аналогового выходного управляющего сигнала (например, для управления скоростью двигателя, положением регулирующего клапана и т. д.).

Между входами и выходами «располагается» управляющая логика — ЦП, которая на основе состояния входов управляет выходами таким образом, чтобы добиться минимального отклонения от желаемого или заданного состояния всего устройства. Интерфейсы: UART и I2C, SPI, TTL, WiFi.



Соединение	1	2	3	4	5	6	7
1. USB <-> ESP8266 (загрузка скетча)	OFF	OFF	OFF	OFF	ON	ON	ON
2. USB <-> ESP8266 (соединение)	OFF	OFF	OFF	OFF	ON	ON	OFF
3. USB <-> ATmega2560	OFF	OFF	ON	ON	OFF	OFF	OFF
4. USB <-> ATmega2560 <-> ESP8266	ON	ON	ON	ON	OFF	OFF	OFF
5. ATmega2560 <-> ESP8266	ON	ON	OFF	OFF	OFF	OFF	OFF
6. Все независимы	OFF	OFF	OFF	OFF	OFF	OFF	OFF

Коммутационная **плата расширения Mega Sensor Shield V2.0** предназначена для расширения функциональности контроллеров на платформе Arduino Mega (Рисунок 3-7).

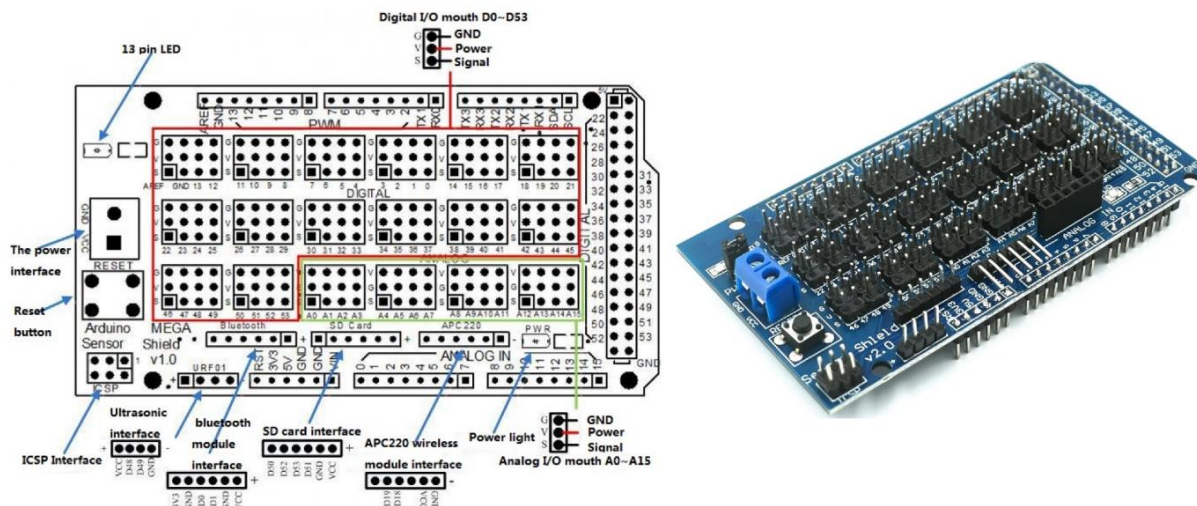


Рисунок 3-7 Mega Sensor Shield V2.0

Плата позволяет облегчить подключение к контроллеру Arduino MEGA большого количества простых периферийных устройств. Это удобно в проектах с использованием множества датчиков, светодиодов, кнопок, двигателей, реле.

- На плате предусмотрена кнопка сброса или перезагрузки RESET.
- Есть индикатор включения питания подписанный PWR.
- Контактные площадки Mega Sensor Shield V2.0 совместимы с Arduino Mega 2560 и Mega 1280 (Рисунок 3-8).
- SD card interface (интерфейс для подключения карт памяти SD) – имеет 6 выводов, первый вывод обозначен на плате знаком «+». Обозначение выводов: VCC, GND, D51, D53, D52, D50.
- Ultrasonic interface (интерфейс для подключения ультразвуковых датчиков) – имеет 4 вывода, первый вывод обозначен на плате знаком «+». Обозначение выводов: VCC, D18, D19, GND.
- Встроенный микрофон.
- Колодка питания Arduino, обозначение выводов: RESET, 3V3, 5V, GND, VIN.
- Digital IO ports (интерфейс аналоговых входов-выходов) имеет 16 выводов. Обозначение выводов: 0 – 53. Каждый вход-выход имеет 3 контакта GND, VCC, Signal.
- Analog IO ports (интерфейс аналоговых входов-выходов) имеет 16 выводов. Обозначение выводов: A0 – A15. Каждый вход-выход имеет 3 контакта GND, VCC, Signal.
- ICSP (внутрисхемное программирование) – для подключения программаторов. Имеет 6 выводов, первый контакт обозначен на схеме интерфейсов платы.
- Входы и выходы интерфейса RS232 (последовательная шина COM).

- ACP220 wireless module interface (интерфейс для подключения внешнего Wi-Fi модуля ACP220) имеет 6 входов. Обозначение входов: GND, VCC, не задействован, D18, D19, не задействован.

- Входы-выходы Arduino контроллера, обозначенные: 0 – 13, GND, AREF. Вывод AREF (опорное напряжение для аналоговых входов).



Рисунок 3-8 Плата расширения установленная на базовую плату

Микрокомпьютер

Микрокомпьютер Orange Pi Zero обладает достаточно высокой вычислительной мощностью по сравнению с обычным микроконтроллером, при наличии привычных всем стандартных интерфейсов SPI, USART, I2C и т.д. Можно с легкостью, к примеру, запустить веб-сервер не вдаваясь в тонкости работы с PHY и MAC-уровнем Ethernet, легко поработать с USB-устройством, при написании программы не нужно постоянно помнить об ограниченном количестве памяти и быстродействии процессора, как это часто бывает при работе с обычными микроконтроллерами.

Одноплатный компьютер Orange Pi Zero построен на базе четырех ядерного процессора Allwinner H3 quad core Cortex A7 1.2 ГГц с GPU Mali-400MP2 GPU 600 МГц, и имеющего 512 МБ DDR3 SDRAM.

На Orange Pi Zero присутствует 1 порт USB 2.0 и стомегабитный Ethernet-интерфейс с поддержкой технологии PoE (Power over Ethernet), позволяющей подавать на устройство питание прямо по Ethernet-кабелю. Чаще всего эта технология применяется в видеонаблюдении и для ее работы нужен поддерживающий PoE сетевой коммутатор.

Самый крупный чип — SoC Allwinner H2+, рядом с ним расположен модуль оперативной памяти емкостью 256 Мб или 512 Мб в зависимости от версии Orange Pi Zero. А маленькая квадратная микросхема — чип Allwinner XR819, дешевый и компактный модуль Wi-Fi. Обычно Wi-Fi модули делают совмещенными с модулями Bluetooth, но XR819 Bluetooth не поддерживает. Нужно иметь это в виду и, если планируется использовать подключаемую по Bluetooth периферию, заранее озаботиться покупкой USB-адаптера.

GPIO-интерфейс представлен двумя гребенками: 13-пиновой для подключения платы расширения и 26-пиновой для всего остального.

Также на плате располагается отладочный UART-разъем и коннектор для съемной антенны Wi-Fi. Антенна уже идет в комплекте.

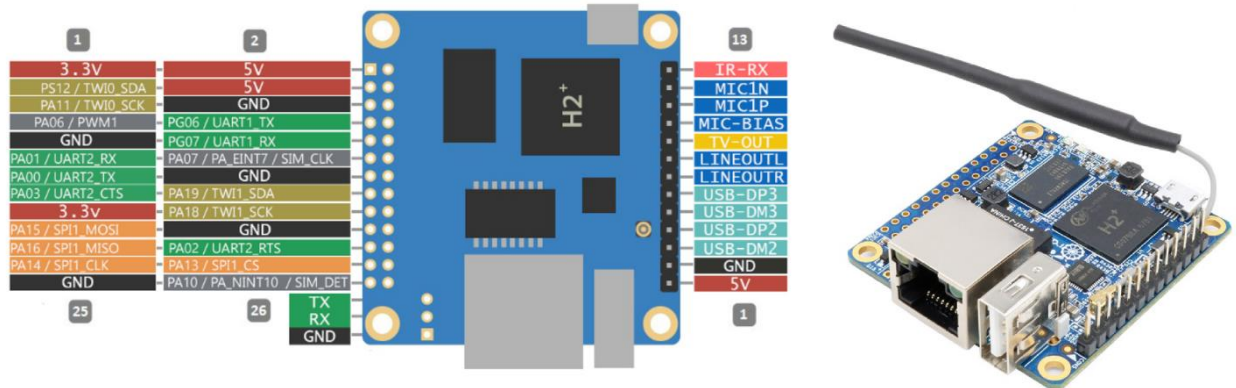


Рисунок 3-9 Микрокомпьютер робототехнического контроллера

Программируемый контроллер

Программируемый контроллер представляет из себя модульное устройство, состоящее из управляющей платы ATMEGA 2560 (Рисунок 3-6), коммутационной платы расширения Mega Sensor Shield V2.0 (Рисунок 3-7) и Arduino Ethernet W510 (Рисунок 3-10).

Плата Arduino Ethernet обладает стандартным 8P8C (также называемым RJ45) ethernet-портом для подключения к сети с помощью патч-корда витой пары, набором контактов для подключения к Arduino, интерфейсом для microSD карты памяти.

Плата построена на основе микросхемы wiznet w5100, которая способна поддерживать обмен данными с постоянной скоростью в 100 Мбит/сек. Для устройств на базе w5100 написаны готовые библиотеки.

Характеристики:

- рабочее напряжение – 5 Вольт, подходит питание с платы Arduino.
- внутренний буфер 16 Кб.
- скорость соединения достигает значения в 10/100 Мбит/сек.
- связь с платой ардуино осуществляется посредством порта SPI.
- поддерживает TCP и UDP.

Назначение светодиодов Ethernet Shield:

- PWR показывает наличие питания на плате.
- LINK светится при наличии сети и мигает при передаче/приеме данных.
- FULLD обозначает сетевое полнодуплексное соединение.
- 100M обозначает сетевое соединение со скоростью 10мбит/сек.
- RX мигает при приеме экраном данных.
- TX мигает при отправке данных экраном.
- COLL мигает при обнаружении сетевых конфликтов.

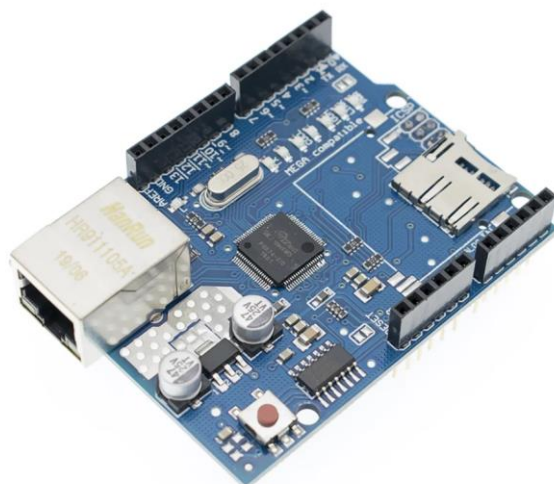


Рисунок 3-10 Arduino Ethernet

Модуль технического зрения

Модуль технического зрения MES Visio (Рисунок 3-11) представляет собой вычислительное устройство со встроенным микроконтроллером, интегрированной телекамерой и оптической системой и предназначен для распознавания изображений, простых и составных объектов посредством собственных вычислительных мощностей.

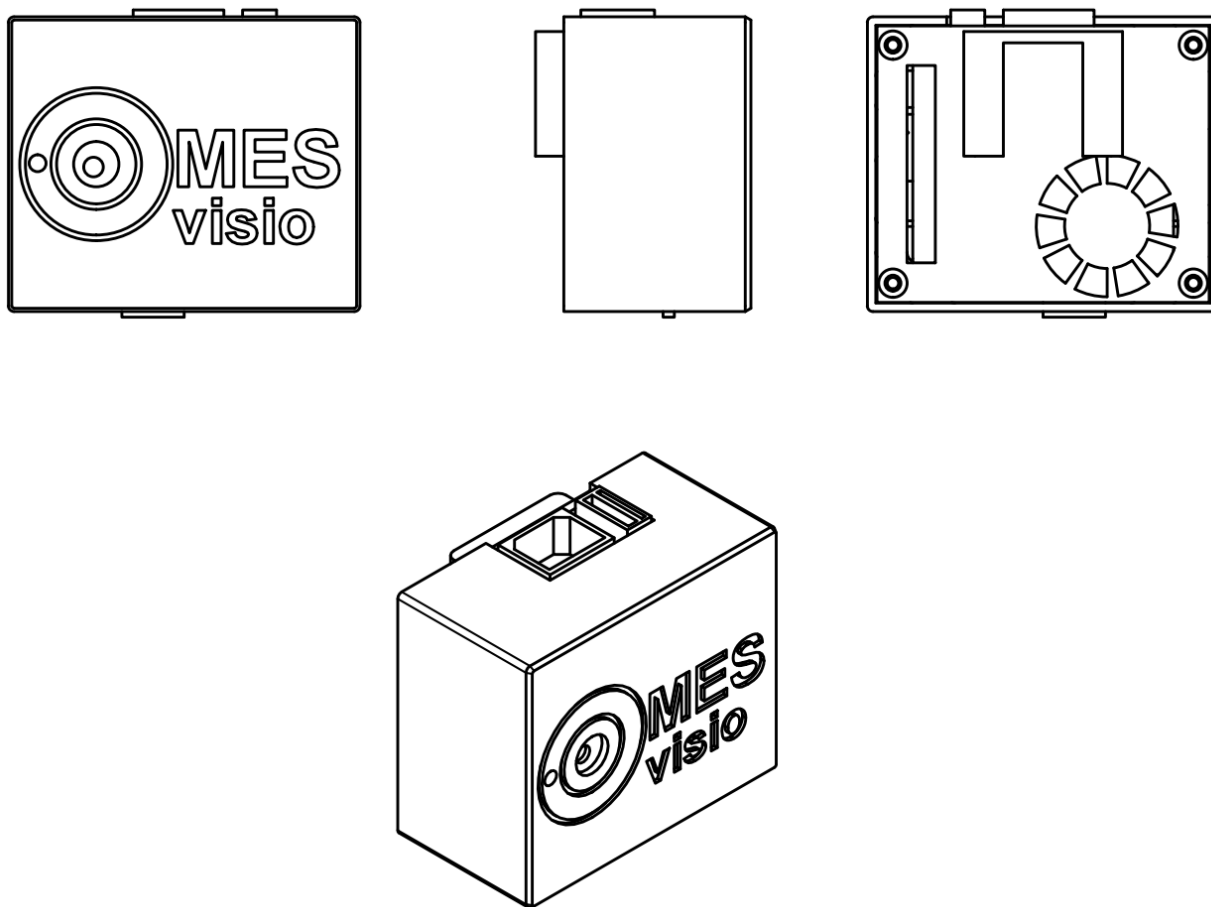


Рисунок 3-11 Модуль технического зрения MES Visio

Техническое зрение применяется в различных производственных отраслях для выполнения многочисленных задач, связанных с визуальным контролем поверхностей, форм, управлением машинами и механизмами подачи, учетом (идентификация, сортировка, маркировка и пр.).

Модуль технического зрения:

- совместим с программируемым контроллером из комплекта поставки;
- измерения и вычисления выполняются посредством собственных вычислительных возможностей встроенного микропроцессора;
- обеспечивает возможность коммуникации с аналогичными модулями посредством шины на базе последовательного интерфейса с целью дальнейшей передачи результатов измерений группы модулей на управляющее вычислительное устройство, подключенное к данной шине;
- осуществляет настройку экспозиции, баланса белого, цветоразностных составляющих, площади обнаруживаемой области изображения, округлости обнаруживаемой области изображения, положение обнаруживаемых областей относительно друг друга;
- обладает интерфейсом USB для настройки модуля;
- может одновременно распознавать до 10 объектов, находящихся в области поля отображения зрения камеры в web-интерфейсе;
- обладает интерфейсами: USB и UART, и 1-wire TTL, и I2C, и SPI для коммуникации со внешними подключаемыми устройствами.

Цифровые информационно-сенсорные модули

Цифровые информационно-сенсорные модули, представляют собой устройства на базе программируемого контроллера и измерительного элемента. Интерфейсы для подключения к внешним устройствам: цифровые и аналоговые порты, и 1-wire TTL, и разъем типа RJ.

Цифровой модуль обеспечивает возможность коммуникации с аналогичными модулями посредством шины на базе последовательного интерфейса с целью дальнейшей передачи результатов измерений группы модулей на управляющее вычислительное устройство, подключенное к данной шине.

Модуль датчика цвета (Рисунок 3-12): датчик излучает три цвета (красный, синий, зеленый), рассчитывает хроматичность и насыщенность отраженного луча и сравнивает полученные результаты с ранее заданными значениями цветовых координат. Если результаты сравнительного анализа находятся в пределах допустимых отклонений, генерируется выходной электрический сигнал.

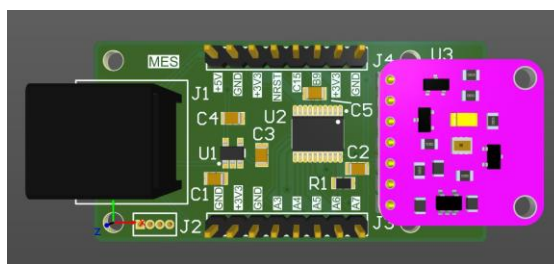


Рисунок 3-12 Модуль датчика цвета

Модуль светодиода (Рисунок 3-13): в зависимости от полученного сигнала с программируемого контроллера коммутируется питание светодиода, алгоритм может быть применен, например, для создания систем умного освещения или сигнализации заданных в управляющем устройстве событий.



Рисунок 3-13 Модуль светодиода

Модуль RGB светодиода (Рисунок 3-14): модуль получает от контроллера значения каждого компонента цвета от 0 до 255, переход к следующему компоненту осуществляется после получения значения «-1», таким образом получается непрерывный цикл переключения между красным, зеленым и синим цветом заданной тональности.

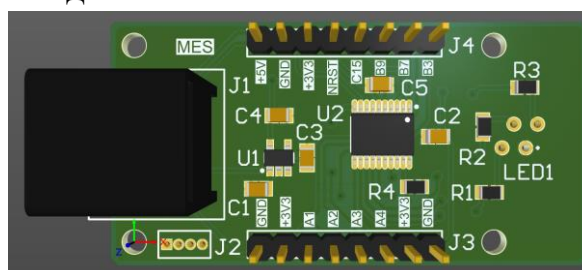


Рисунок 3-14 Модуль RGB светодиода

Модуль тактовой кнопки и концевого выключателя (Рисунок 3-15) имеют схожую архитектуру и принцип действия, при воздействии на исполнительный элемент модуля высокий уровень сигнала подается на управляющий контроллер. При отсутствии воздействия уровень сигнала равен нулю.

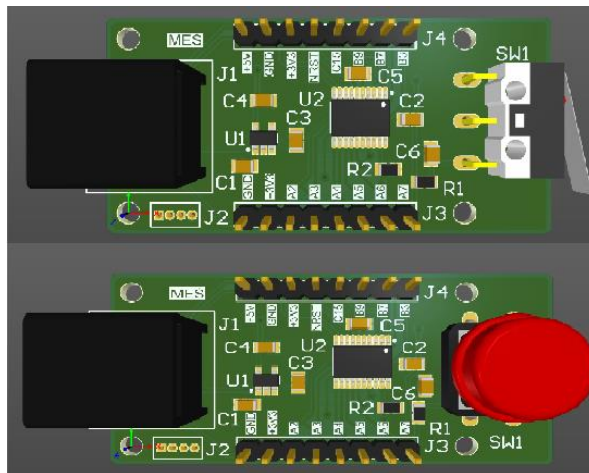


Рисунок 3-15 Модули концевого выключателя и тактовой кнопки

Перед началом работы с цифровыми модулями необходимо осуществить их прошивку (см. учебное пособие).

Цифровые модули обеспечивают возможность коммуникации друг с другом посредством шины на базе последовательного интерфейса с целью дальнейшей передачи результатов измерений группы модулей на управляющее вычислительное устройство, подключенное к данной шине.

Элементы для сборки вакуумного захвата

Вакуумный захват (Рисунок 3-16) позволит Вам реализовать алгоритм сортировки и комплектования средних и крупных изделий весом до 0,1 кг.

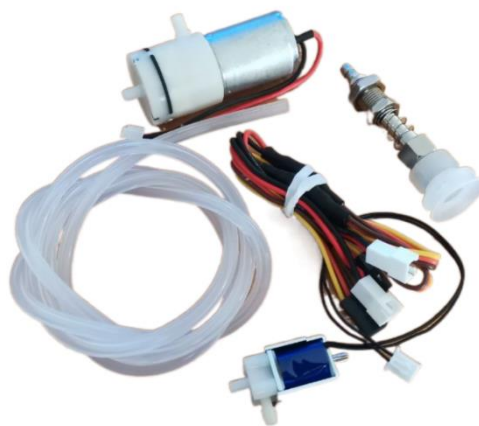


Рисунок 3-16 Комплект вакуумного захвата

Комплект состоит из вакуумной присоски, винилового шланга, пневматической помпы, двухпозиционного клапана и управляющих плат.

Технические характеристики :

Вакуумный насос

- Номинальное напряжение: DC4.5-6.6 В
- Ток холостого хода: 0.15А
- Диапазон давления: 400-650 мм рт.ст.
- Максимальный вакуум: -350 мм рт.ст.
- Вес: 65 г

Двухпозиционный электронный клапан

- Номинальное напряжение: 6 В постоянного тока
- Мощность: 220 мА
- Подходящее напряжение: DC4.8-6.6 В
- Мощность: 2 Вт
- Вес: 16 г

Вакуумная присоска

- Размер установки: диаметр установочного отверстия 10 мм
- Общая длина 60 мм
- Диаметр присоски 20 мм

Комплект крепежа

Винт с полукруглой головкой с крестообразным шлицем M2x10.
Винт с полукруглой головкой с крестообразным шлицем M2x6.
Винт с полукруглой головкой с крестообразным шлицем M2x8.
Винт с полукруглой головкой с крестообразным шлицем M3x12.
Винт с полукруглой головкой с крестообразным шлицем M3x16.
Винт с полукруглой головкой с крестообразным шлицем M3x25.
Винт с полукруглой головкой с крестообразным шлицем M3x6.
Винт с полукруглой головкой с крестообразным шлицем M3x8.
Гайка M3.
Гайка нейлоновая M3.
Гайка M2.
Гайка самоконтрящаяся M3.
Стойка нейлоновая M3x20.
Шайба 3,2.
Шайба увеличенная 3,2.

Набор проводов

Набор робототехнического конструктора комплектуется соединительными проводами Dupont, предназначенными для быстрой сборки и проектирования электронных схем.

В набор входят провода с разъемами типа: папа – папа, папа – мама, мама – мама.

Тип провода – многожильный, материал провода – медь, материал изоляции – ПВХ, цвет – разноцветные (Рисунок 3-17).



Рисунок 3-17 Набор проводов

Блок питания 220В

Питание роботизированного манипулятора осуществляется сетевым блоком питания, преобразующим 220 В переменного тока в 12 В постоянного тока (Рисунок 3-18)



Рисунок 3-18 Блок питания 220 В

USB кабель

Коммуникация роботизированного манипулятора с ПК осуществляется с помощью кабеля micro USB (Рисунок 3-19).



Рисунок 3-19 Кабель micro USB

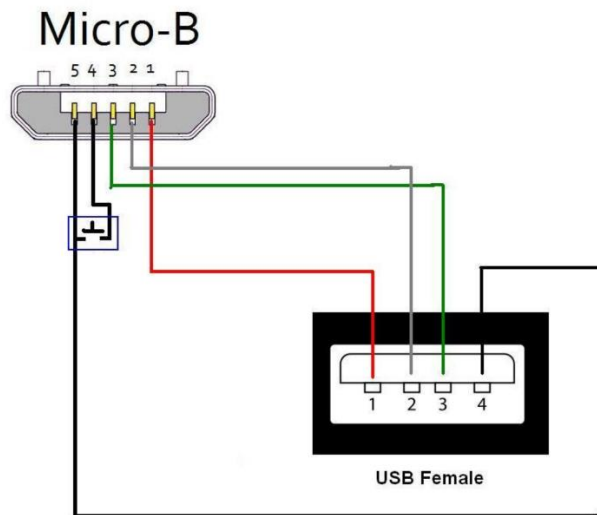


Рисунок 3-20 Распиновка разъемов micro-USB и USB

Набор библиотек трехмерных элементов

На сайте производителя www.hobots1.ru представлен набор библиотек трехмерных элементов для прототипирования моделей манипуляционных роботов (Рисунок 3-21).



Рисунок 3-21 Набор 3D элементов

4 Урок №1. Знакомство с контроллерами семейства Arduino

Программируемая аппаратная платформа «Arduino» - это один из самых популярных в современной технике комплексов для создания программируемых электронных устройств, открывает широкие возможности по автоматизации и роботизации бытовых и промышленных устройств, роботов, CNC-станков и т.д. При этом не обязательно быть программистом и обладать специальными знаниями, достаточно иметь желание и творческую идею. С помощью «Arduino» можно создавать как элементарные, так и более сложные системы управления.

История «Arduino»

В 2005 году итальянец Массимо Банци создал первое устройство, которое отдаленно напоминало современный контроллер, и назвал его в честь короля Ардуина. Устройство развивалось и совершенствовалось командой энтузиастов и «Arduino» стало брендом аппаратно-программных средств, с помощью которых можно автоматизировать несложные системы. В основе аппаратной платформы «Arduino» лежит программируемый контроллер фирмы ATMEL с классической архитектурой RISC. Его достоинства – универсальность, модульность, цена.

Другими словами, «Arduino» - это миникомпьютер, который можно запрограммировать для выполнения определенных действий. Имеет «на борту» самостоятельный процессор, который обладает памятью и снабжен множеством стандартных портов вводов и выводов. К нему легко подключаются самые различные устройства и механизмы, сенсоры, датчики, моторы и прочее.



Рисунок 4-1 Внешний вид Arduino UNO

Принцип модульности

Платы «Arduino» не просто совместимы, а могут соединяться друг с другом посредством штыревых разъемов. Фактически они «насаживаются» друг на друга как элементы «бутерброда». Такая доступная модульность – одно из основных преимуществ плат «Arduino».



Рисунок 4-2 Изображение принципа «бутерброда»

Программирование контроллеров «Arduino» осуществляется в интегрированной среде разработки – Arduino IDE, размещенной на сайте <https://www.arduino.cc/>. Она распространяется по бесплатной лицензии. Программа загружается с компьютера, к которому плата присоединяется через USB-порт.

Преимущества «Arduino»

- Невысокая цена и возможность самостоятельной сборки
- Широкие возможности, модульность и универсальность
- Доступность для непрофессионального пользователя
- Большое количество информации в сети: обучающие ролики, обзоры, пошаговые инструкции
- сообщества, где можно получить ответы на интересующие вопросы
- Наличие дополнительного оснащения: кнопки, датчики, индикаторы, а также другие устройства и аксессуары, обеспечивающие взаимодействие контроллера с окружающим миром
- Простой процесс разработки системы и ее отладки

Рассмотрим более подробно семейство платформ «Arduino», возможности и особенности работы с ними на примере нескольких моделей программируемых контроллеров.

Программируемый контроллер Arduino UNO R3 (ATMEGA16U2 + ATMEGA328P) – это самый массовый представитель контроллеров семейства «Arduino», созданных на базе микроконтроллера ATmega328p.



Рисунок 4-3 Внешний вид платы Arduino UNO R3

Плата имеет 14 цифровых входов/выходов, 6 из которых могут использоваться как выходы с ШИМ, а также имеется 6 аналоговых входов. В данной плате в качестве встроенного программатора применяется чип Atmega16U2, который позволяет превратить плату в любое USB-устройство. По сравнению с предыдущими версиями Arduino UNO R3, она характеризуется более удобной маркировкой входов и выходов.

Питание Arduino UNO R3 подается по USB или от внешнего источника питания, в качестве которого может использоваться аккумуляторная батарея или сетевой AC/DC-адаптер. Рекомендуется к применению источник питания с напряжением в диапазоне 7-12 В, 12 Вт. Объем флеш-памяти составляет 32 КБ.

Контроллер Arduino UNO R3 может быть подключен к компьютеру, другой плате Arduino или к другому микроконтроллеру, не обязательно Arduino-подобному. Для этого он оснащен всем необходимым.

Обратите внимание, что запрещается превышать допустимые величины силы тока на выводах контроллера! Для одного любого вывода ток не должен составлять более 40 мА. Ток для одной группы выводов не может быть более 100 мА. Ток для всего микроконтроллера не должен превышать 200 мА.

Программируемый контроллер Arduino MEGA2560 R3 (CH340G + ATMEGA2560-16AU) разработан на базе микроконтроллера ATMEGA2560.



Рисунок 4-4 Внешний вид платы Arduino MEGA

Данный микроконтроллер имеет собственный загрузчик, который облегчает установку новых программ без использования дополнительных программаторов.

В отличие от Arduino Uno R3, платформа имеет большое количество входов/выходов цифровых и аналоговых. Ее габариты, соответственно, больше. Данная плата совместима с расширениями и модулями для платформ Duemilanove и Uno. Флеш-память составляет 256 КБ. В отношении этой платы действуют те же принципы работы, как и в отношении платы Arduino UNO R3.

Сердцем платформы Arduino Mega является 8-битный микроконтроллер семейства AVR — ATmega2560 с тактовой частотой 16 МГц. Контроллер предоставляет 256 КБ Flash-памяти для хранения прошивки, 8 КБ оперативной памяти SRAM и 4 КБ энергонезависимой памяти EEPROM для хранения данных.

Микроконтроллер ATmega16U2 обеспечивает связь микроконтроллера ATmega2560 с USB-портом компьютера. При подключении к ПК Arduino Mega 2560 определяется как виртуальный COM-порт.

Светодиодная индикация.

RX и TX Мигают при обмене данными между Arduino Mega 2560 и ПК. L Пользовательский светодиод подключённый к 13 пину микроконтроллера. При высоком уровне светодиод включается, при низком – выключается. ON Наличие питания на Arduino Mega.

Разъём USB Type-B для питания и прошивки платформы Arduino Mega 2560 с помощью компьютера.

Разъём внешнего питания

Разъём для подключения внешнего питания от 7 В до 12 В.

Кнопка сброса

Аналог кнопки RESET обычного компьютера. Служит для сброса микроконтроллера.

Регулятор напряжения 5 В

Линейный понижающий регулятор напряжения с выходом 5 вольт обеспечивает питание микроконтроллеров ATmega2560, ATmega16U2 и другой логики платформы. Максимальный выходной ток составляет 800 мА.

Регулятор напряжения 3,3 В

Линейный понижающий регулятор напряжения с выходом 3,3 вольта. Линия выведена только на пин 3.3V. Максимальный выходной ток составляет 150 мА.

Разъём ICSP

ICSP-разъём предназначен для внутрисхемного программирования микроконтроллера ATmega2560. Также с применением библиотеки SPI данные выводы могут осуществлять связь с платами расширения по интерфейсу SPI. Линии SPI выведены на 6-контактный разъём, а также продублированы на цифровых пинах 50(MISO), 51(MOSI), 52(SCK) и 53(SS).

Разъём ICSP1

ICSP-разъём для внутрисхемного программирования микроконтроллера ATmega16U2.

VIN: Входной пин для подключения внешнего источника питания с напряжением в диапазоне от 7 до 12 вольт. Через контакт можно потреблять напряжение, когда устройство запитано через внешний разъём питания.

5V: Выходной пин от регулятора напряжения на плате с выходом 5 вольт и максимальным током 800 мА. Питать устройство через вывод 5V не рекомендуется.

3.3V: Выходной пин от регулятора напряжения с выходом 3,3 вольта и максимальным током 150 мА. Питать устройство через вывод 3V3 не рекомендуется.

GND: Выводы земли.

IOREF: Контакт предоставляет платам расширения информацию о рабочем напряжении микроконтроллера. В зависимости от напряжения, плата расширения может переключиться на соответствующий источник питания либо задействовать преобразователи уровней.

AREF: Пин для подключения внешнего опорного напряжения АЦП относительно которого происходят аналоговые измерения при использовании функции `analogReference()` с параметром «EXTERNAL».

Порты ввода/вывода

Цифровые входы/выходы: пины 0–53

Логический уровень единицы — 5 В, нуля — 0 В. Максимальный ток выхода — 40 мА. К контактам подключены подтягивающие резисторы, которые по умолчанию выключены, но могут быть включены программно.

ШИМ: пины 2–13 и 44–46

Позволяет выводить аналоговые значения в виде ШИМ-сигнала. Разрядность ШИМ не меняется и установлена в 8 бит.

АЦП: пины A0–A16

Позволяет представить аналоговое напряжение в цифровом виде. Разрядность АЦП не меняется и установлена в 10 бит. Диапазон входного напряжения от 0 до 5 В.

TWI/I2C: пины 20(SDA) и 21(SCL)

Для общения с периферией по интерфейсу I2C используйте библиотеку `Wire`.

SPI: пины 50(MISO), 51(MOSI), 52(SCK) и 53(SS).

Для общения с периферией по интерфейсу SPI используйте библиотеку `SPI`.

UART: пины 0(RX) и 1(TX), 19(RX1) и 18(TX1), 17(RX2) и 16(TX2), 15(RX3) и 14(TX3)

Используется для коммуникации платы Arduino с компьютером или другими устройствами по последовательному интерфейсу. Выводы 0(RX) и 1(TX) соединены с соответствующими выводами микроконтроллера

ATmega16U2, выполняющего роль USB-UART преобразователя. Для работы с последовательным интерфейсом — используйте методы библиотеки Serial.

Программируемый контроллер Arduino Nano V3.0 (CH340G + ATMEGA328P-AU) – компактная плата на базе микроконтроллера ATMEGA328P-AU.

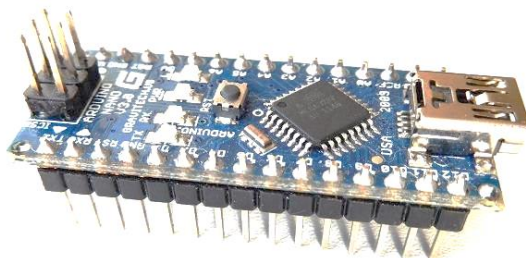


Рисунок 4-5 Программируемый контроллер Arduino Nano V3.0

Она имеет функциональность, схожую с платформами Arduino UNO, но отличается меньшими габаритами. Обладает 14 цифровыми входами/выходами, 6 из которых могут использоваться как выходы ШИМ, 8 аналоговыми выходами. Флеш-память составляет 32 КБ. Это бюджетная модель на чипе CH340G.

На плате расположен стабилизатор 5 В. Также имеется загрузчик, который позволяет платформе «перепрошивать» саму себя. Как и в рассмотренных выше платах Arduino, есть возможность не использовать загрузчик, а запрограммировать микроконтроллер через ICSP-выводы. С помощью кнопки RESET производится перезагрузка в случае непредвиденного сбоя. Плата подключается к компьютеру через miniUSB-кабель.

Arduino Pro mini – программируемый контроллер (ATMEGA328P) разработан на микроконтроллере ATMEGA328P, имеет 14 цифровых входов/выходов и 6 аналоговых.

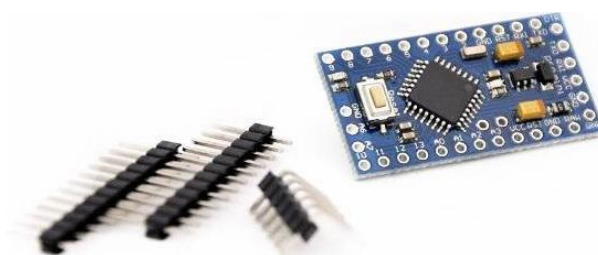


Рисунок 4-6 Программируемый контроллер Arduino Pro mini

Arduino Pro mini питается через кабель FTDI-программатора или от внешних источников питания через имеющиеся выходы (Vcc/RAW). Прежде всего, данная плата отличается от всех предыдущих своей миниатюрностью. Именно поэтому ее использование особенно популярно при работе с небольшими устройствами. Соответственно, из-за компактных размеров данной платы к ней не подключить «по принципу бутерброда» расширения более

крупных плат, однако их можно подключить с помощью дополнительных коннекторов. Для хранения кода программ предусмотрена флеш-память 16 КБ.

В целом, технические характеристики программируемого контролера Arduino Pro mini очень схожи с характеристиками других плат Arduino. Основное отличие – это отсутствие USB-интерфейса. Благодаря этому достигается миниатюрность в размерах, но возникает необходимость в использовании дополнительного модуля, типа мини-USB-коннектора, чтобы подключить плату к компьютеру для загрузки программы.

Программируемый контроллер Arduino DUE 2012 R3 (ARM 32 + AT91SAM3X8EA) – плата на базе микроконтроллера AT91SAM3X8EA.

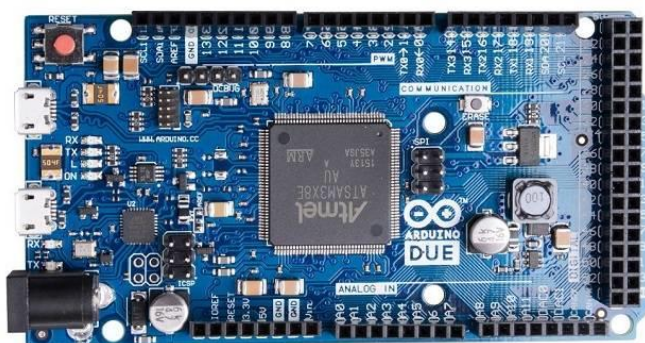


Рисунок 4-7 Программируемый контроллер Arduino DUE 2012 R3

Она имеет 54 цифровых входов/выходов и 12 аналоговых. Ее основное отличие от других плат Arduino – это работа при напряжении не выше 3,3 В.

В плате есть все необходимое для того, чтобы сразу начать работу. Достаточно подключить ее к компьютеру через микро-USB-кабель или подать питание, подключив плату к батарейке. Можно также использовать AC/DC-преобразователь.

Arduino DUE отличается тем, что на ней установлено 32-битное ARM-ядро, обеспечивающее более высокую производительность платформы по сравнению с другими контроллерами Arduino. Кроме того, Arduino DUE имеет огромную память для записи программ – 512 КБ. Есть возможность стереть информацию из памяти с помощью специальной кнопки.

Программируемый контроллер Arduino Leonardo R3 (ATMEGA32U4) выполнен на базе микроконтроллера ATMEGA32U4.

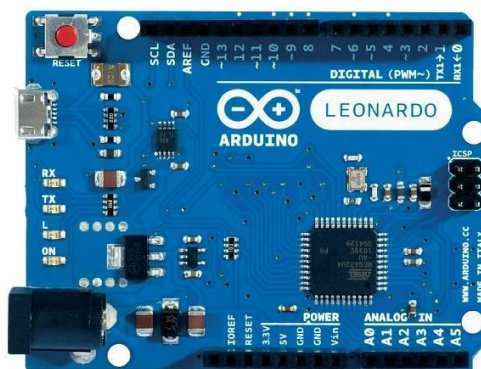


Рисунок 4-8 Программируемый контроллер Arduino Leonardo R3

Имеет 20 цифровых входов/выходов, 7 из которых могут использоваться как выходы ШИМ и 12 как аналоговые выходы. Данная плата является обновленной версией платы Arduino Uno. Отличие платы Arduino Leonardo заключается в том, что она имеет встроенную поддержку для USB-подключения.

Необходимую документацию на любой программируемый контроллер Arduino на английском языке можно найти на официальном сайте компании <https://www.arduino.cc/>.

Выбор платы Arduino

Плата Arduino MEGA входит во многие робототехнические наборы, которые позволяют приступить к изучению основ электроники, конструирования и программирования. Это стандартная плата для начала работы с «Arduino». Выбирая плату для проекта, прежде всего, следует обратить внимание на ее технические характеристики, а именно:

- на базе какого микроконтроллера создана плата. Именно от этого будет зависеть скорость ее работы
- входное и выходное напряжение портов платы
- количество входов и выходов. Эта характеристика определяет то количество устройств, которое можно будет подключить к плате и по какому интерфейсу
- объем флеш-памяти, от которой зависит объем рабочей программы
- размер платы
- напряжение питания и наличие стабилизатора напряжения

Выводы

Итак, теперь мы знаем, что такое Arduino, ее технические параметры и различия между различными платами. Выбор был сделан в сторону самой распространенной и достаточно производительной платы MEGA со встроенным WiFi – модулем на борту, чтобы не усложнять конструкцию дополнительными устройствами, а также простота программирования модуля как единой платы, облегчает сборку устройства. Теперь нам необходимо установить среду

разработки Arduino IDE, которая позволяет программировать все контроллеры семейства в режиме текстового редактора кода на языке C++, а также в режиме графического редактора CodeBlocks.

5 Урок №2. Установка и настройка Arduino IDE

На данном уроке мы установим необходимую для программирования контроллеров среду разработки – Arduino IDE. Это официальный проект разработчиков, распространяемый по бесплатной лицензии.

Установка ПО

На компьютер с операционной системой Windows устанавливаем программное обеспечение - интегрированную среду разработки Arduino IDE, которую можно скачать, пройдя по ссылке <https://www.arduino.cc/en/software>

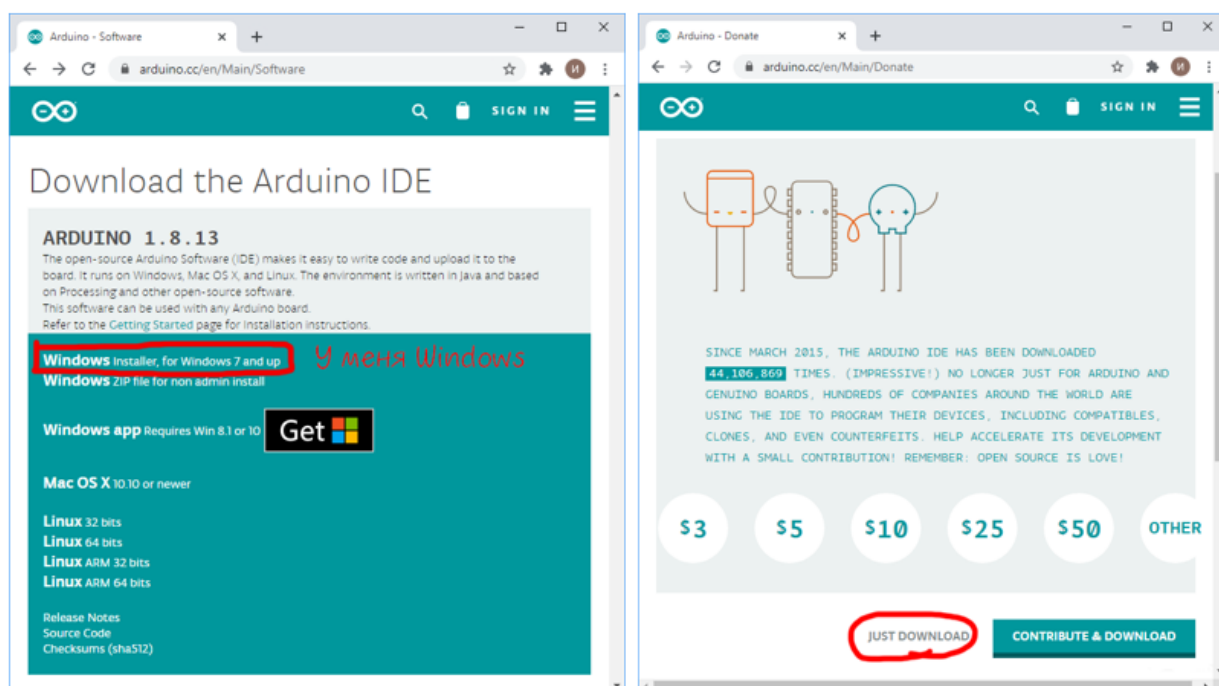


Рисунок 5-1 Сайт разработчиков. Страница загрузки

Нажмите на кнопку «JUST DOWNLOAD» для бесплатной загрузки программы. После того, как файл загрузился, запустите его, и соглашаясь со всеми пунктами программы-установщика, а также кнопками «доверять программному обеспечению», завершите установку, дождавшись надписи «complete».

Запустите установленную программу. Если Arduino IDE не запускается, вероятнее всего, на компьютере некорректно установлена JRE — Java Runtime Environment. Для решения проблемы переустановите программу.

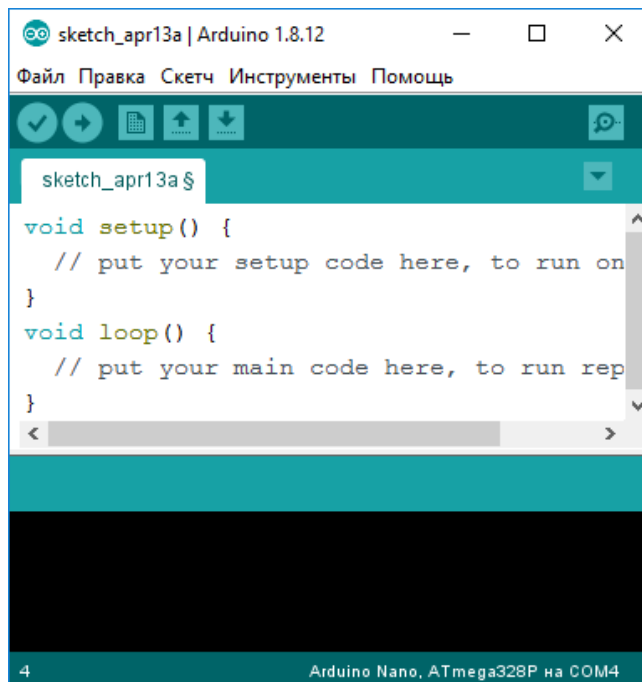


Рисунок 5-2 Вид Arduino IDE при первом запуске

Подключение платы Arduino к компьютеру

Соедините Arduino с компьютером по USB-кабелю. На плате загорится светодиод «ON» и начнёт мигать светодиод «L». Это значит, что на плату поступает питание и микроконтроллер начал выполнять запрограммированную на заводе программу «Blink».

Для настройки Arduino IDE под конкретную модель узнайте, какой номер COM-порта присвоил компьютер вашей плате. Зайдите в «Диспетчер устройств» Windows и раскройте вкладку «Порты (COM и LPT)».

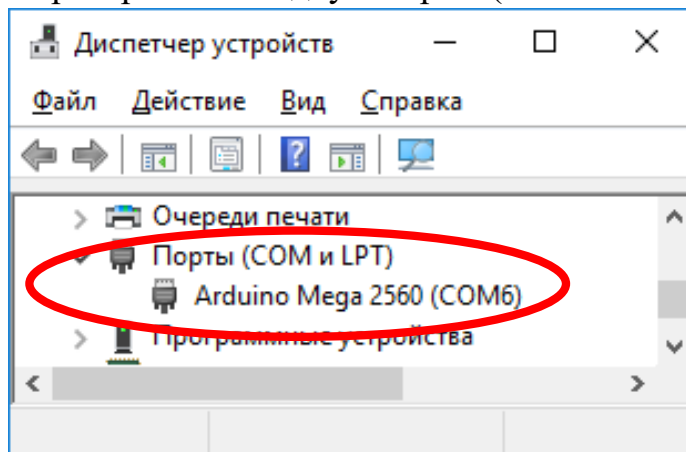


Рисунок 5-3 Общий вид диспетчера устройств

Операционная система назначила номер 2. Если вы подключите к компьютеру другую плату, операционная система назначит ей другой номер. Если у вас несколько платформ, очень важно не запутаться в номерах COM-портов.

Если после подключения Arduino к компьютеру, в диспетчере устройств не появляются новые устройства, это может быть следствием следующих причин:

- Неисправный USB-кабель или порт
- Блокировка со стороны операционной системы
- Неисправная плата

Настройка Arduino IDE

Для работы среды Arduino IDE с конкретной платформой необходимо выбрать название модели и номер присвоенного плате COM-порта.

Для выбора модели Arduino зайдите в меню: Инструменты Плата и укажите соответствующую плату.

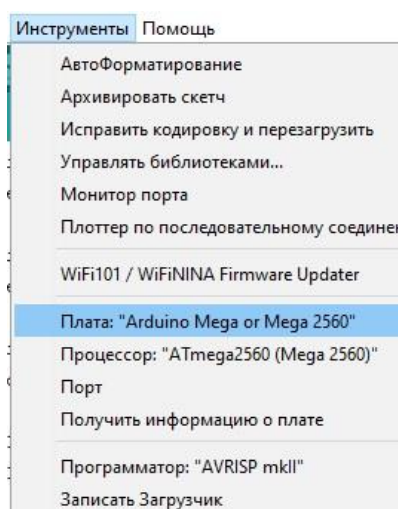


Рисунок 5-4 Меню выбора платы

Для выбора номера COM-порта перейдите в меню: Инструменты Порт и укажите нужный порт.

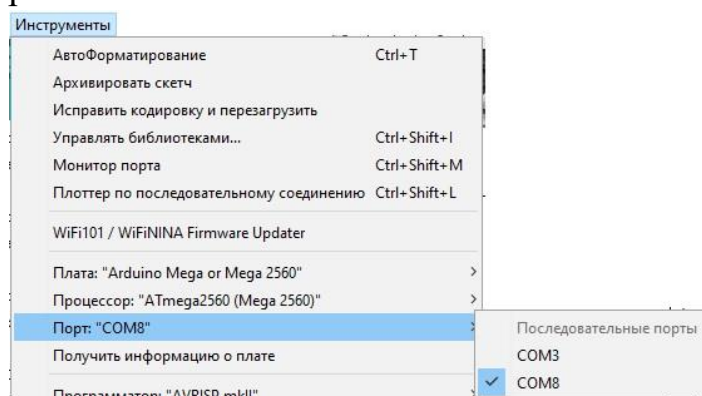


Рисунок 5-5 Меню выбора порта

После данных действий, среда Arduino IDE настроена для прошивки платы.

Если список последовательных портов пуст, значит платформа некорректно подключена к компьютеру или не установлен драйвер (<http://hobots.ru>). Вернитесь к настройке соединения между платой и ПК.

Если Arduino IDE «тормозит» при навигации по меню, отключите в диспетчере устройств все внешние устройства типа «Bluetooth Serial». Например, виртуальное устройство для соединения с мобильным телефоном по Bluetooth может вызвать такое поведение.

Среда настроена, плата подключена. Пора прошивать платформу. Arduino IDE содержит большой список готовых примеров, в которых можно быстро подсмотреть решение какой-либо задачи. Откройте распространенный пример — «Blink»: Файл Примеры 01.Basics → Blink.

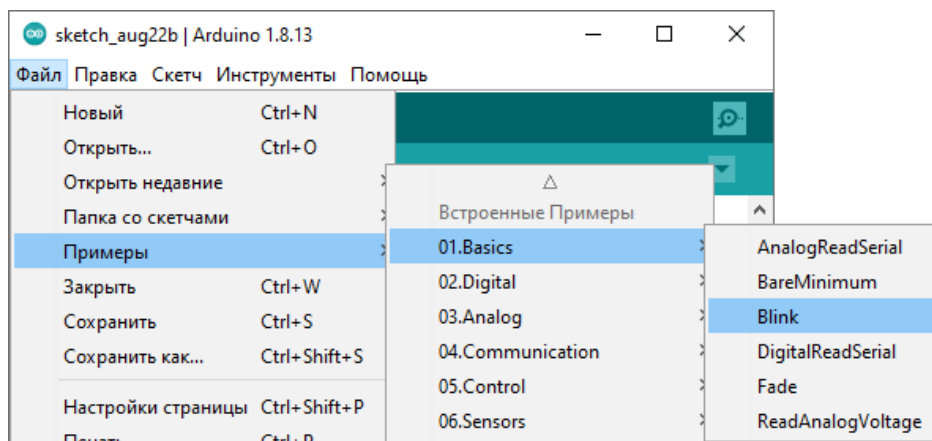


Рисунок 5-6 Меню выбора примера

Откроется окно с демонстрационным примером.



Рисунок 5-7 Окно с примером

Немного модифицируйте код, чтобы увидеть разницу с заводским миганием светодиода. Функция `setup()` останется без изменений. Замените строчки в функции `loop()`:

```
delay(1000);
```


на:

```
delay(100);
```

После модификации код будет иметь вид:

```
void loop() {  
  // turn the LED on (HIGH is the voltage level)  
  digitalWrite(LED_BUILTIN, HIGH);  
  // wait for a second  
  delay(100);  
  // turn the LED off by making the voltage LOW  
  digitalWrite(LED_BUILTIN, LOW);  
  // wait for a second  
  delay(100);  
}
```

Нажмите на иконку «Компиляция» для проверки кода на ошибки, затем на иконку «Загрузка» для программирования контроллера.

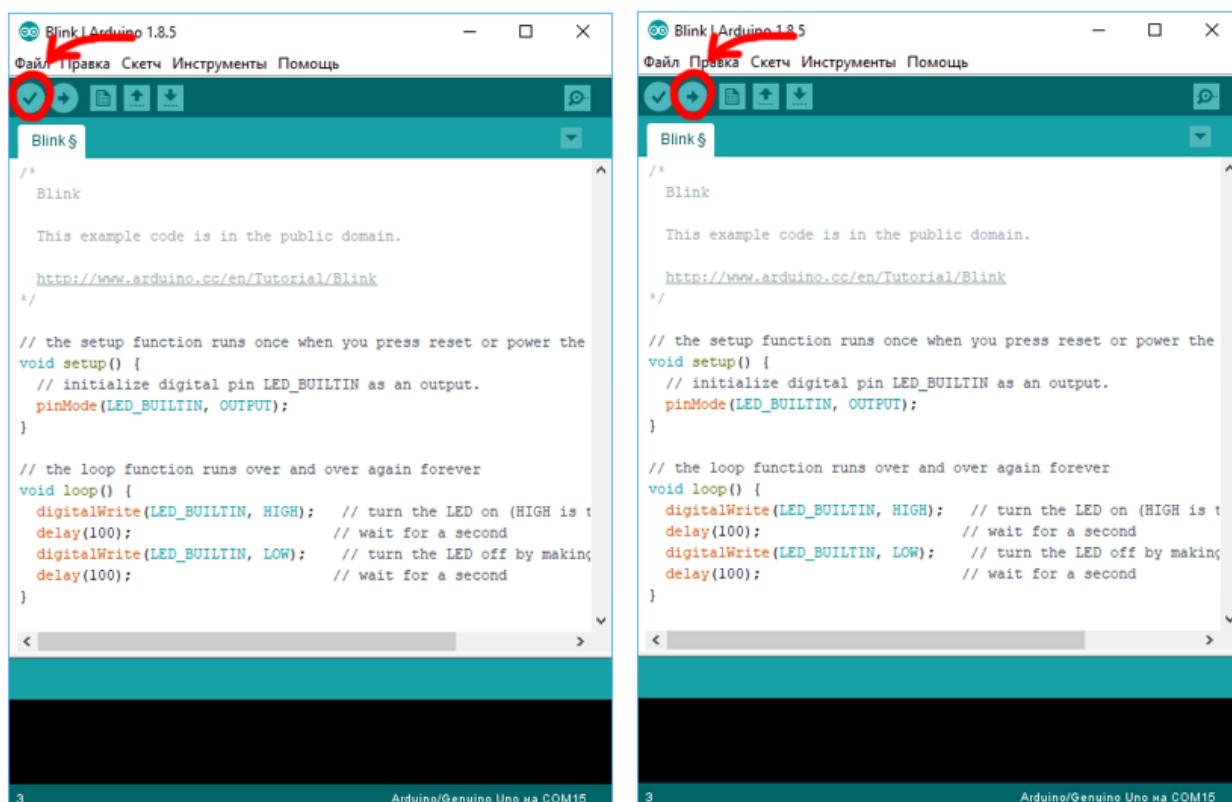


Рисунок 5-8 Кнопка проверки скетча Кнопка загрузки в плату

После прошивки платформы светодиод «L» начнёт загораться и гаснуть каждые 100 миллисекунд — в 10 раз быстрее исходной версии. Это значит, что ваш тестовый код успешно загрузился и заработал. Теперь смело переходите к последующим экспериментам!

6 Урок №3. Установка и настройка графической системы программирования ArduBlock



Рисунок 6-1 Внешний вид блоков

Что такое ArduBlock?

ArduBlock – это графический язык программирования для Arduino, предназначенный для непрограммистов и простой в использовании.



Рисунок 6-2 Внешний вид интеграции Arduino IDE и ArduBlock.

1. Скачайте архив `ardublock-all.jar` с сайта ArduBlock: <http://ardublock.ru/ru/>
2. Откройте “Arduino IDE/Menu /Arduino/ Preferences”, там вы найдете строку
3. «Sketch location» и скопируйте архив “ardublock-all.jar” в папку “tool”.
4. Если имя пользователя ПК “abu”, то путь будет иметь вид: `C:\Users\abu\Documents\Arduino`
5. Перезапустите Arduino IDE и у Вас должен появиться пункт “ArduBlock” в меню “Tool”.

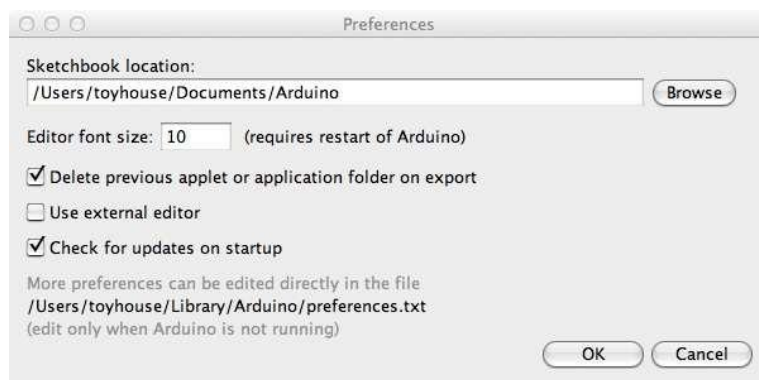


Рисунок 6-3 Вид окна Arduino IDE – Preferences

Внимание! Обращайте внимание на написание названий папок прописными и строчными буквами.

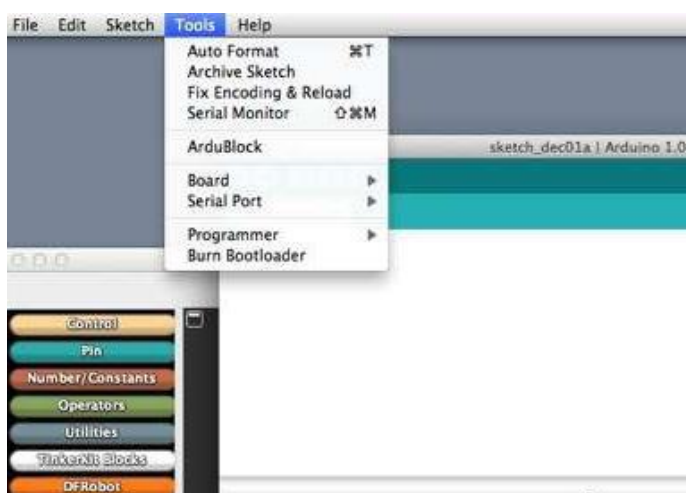


Рисунок 6-4 Вид окна меню «Инструменты» Arduino IDE

Возможен вариант скачивания визуальной системы программирования в качестве отдельного приложения, в этом случае необходимо разархивировать скаченный архив и в папке app запустить файл ArduBlock.exe. При необходимости, правой кнопкой мыши нажав на иконку ArduBlock.exe выбрать пункт меню Отправить → Рабочий стол (Создать ярлык), на рабочем столе появится ярлык с программой.

Использование

Блоки ArduBlock разделены на несколько категорий.

Control - категория управляющих блоков.

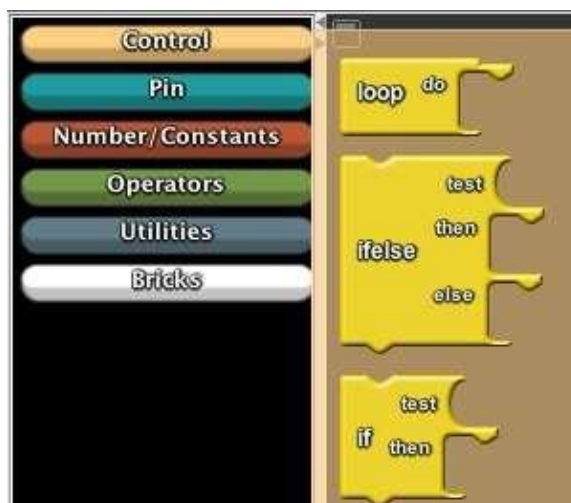


Рисунок 6-5 Блоки категории “Control”

Цифры, константы и Переменные

Блоки категорий “Numbers/Constants” это переменные:



Рисунок 6-6 Блоки категории “Numbers/Constants”

Эта категория включает в себя логические и математические операторы.



Рисунок 6-7 Блоки категории логических и математических операторов

Эти блоки являются функциями, которые обычно используются в скетчах.



Рисунок 6-8 Блоки функций

Каждый блок данной категории изображает тип реального устройства, который вы можете напрямую подключить к вашему скетчу.



Рисунок 6-9 Изображение типа реальных устройств

Эти блоки действуют как виртуальные контакты на плате Arduino.



Рисунок 6-10 Виртуальная установка типа пинов

Программирование

1. Компиляция должна завершиться успешно. Если порт, указанный в среде Arduino или сама плата не обнаружены, то появляется окно с сообщением об ошибке.



Рисунок 6-11 окно с сообщением об ошибке

2. Графические блоки с разъемами одинаковой формы можно соединить друг с другом.

Соединение устанавливается просто, для этого нужно перетащить один блок к тому, с которым Вы хотите его соединить.

Как только будет нажата кнопка “upload”, ArduBlock автоматически сгенерирует код Arduino который потом будет загружен на плату Arduino (при этом в окне скетчей среды разработки Arduino появится текст программы, полученной в ходе компиляции).

7 Урок №4. Основная информация по структуре и командам языка программирования C++

Данный посвящен основам программирования на языке C++, на уровне, необходимом для разработки проектов на базе аппаратной платформы Arduino. Начнем с рассмотрения примера минимально возможной программы на C++ для Arduino, которая «ничего не делает»:

```
void setup()
{
}

void loop()
{
}
```

Разберёмся, что здесь написано и почему это обязательно: почему нельзя обойтись просто пустым файлом.

Из чего состоит программа

Для начала стоит понять, что программу нельзя читать и писать как книгу: от корки до корки, сверху вниз, строку за строкой. Любая программа состоит из отдельных блоков. Начало блока кода в C/C++ обозначается левой фигурной скобкой {, его конец — правой фигурной скобкой }.

Блоки бывают разных видов и какой из них, когда будет исполняться зависит от внешних условий. На примере минимальной программы можем видеть 2 блока. В этом примере блоки называются определением – функцией. Функция — это отдельный блок кода с заданным именем, к которому можно обратиться из любого места программы.

В данном случае – 2 функции с именами setup() и loop(). Их присутствие обязательно в любой программе на C++ для Arduino. Они могут ничего и не содержать, как в нашем случае, но должны быть написаны. Иначе на стадии компиляции вы получите ошибку.

Мигающий светодиод

Давайте теперь дополним нашу программу так, чтобы происходил визуальное доказательство работы. На Arduino, к 13-му пину подключён светодиод. Им можно управлять.

```
void setup()
{
  pinMode(13, OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(100);
  digitalWrite(13, LOW);
  delay(900);
}
```

Скомпилируйте, загрузите программу в плату. Вы увидите, что каждую секунду светодиод на плате помигивает.

В наши ранее пустые функции мы добавили несколько выражений. Они были размещены между фигурными скобками функций `setup()` и `loop()`. В `setup()` появилось одно выражение, а в `loop` сразу четыре.

Каждое выражение — это приказ процессору что-то сделать. Выражения в рамках одного блока исполняются одно за другим, строго по порядку без пауз и переключений. То есть, если мы говорим об одном конкретном блоке кода, его можно читать сверху вниз, чтобы понять, что выполняется.

В каком порядке исполняются сами блоки функций `setup()` и `loop()`. Как только Arduino включается, перепрошивается или нажимается кнопка RESET, сразу вызывается функция `setup()`. То есть исполняются выражения в ней.

Как только работа кода в `setup()` завершается, сразу же вызывается функция `loop()`. Как только работа `loop()` завершается, сразу же вызывается функция `loop()` ещё раз и так — бесконечно.

Если пронумеровать выражения по порядку, как они исполняются, получится:

```
void setup()
{
  pinMode(13, OUTPUT); ❶
}

void loop()
{
  digitalWrite(13, HIGH); ❷   ❸   ❹
  delay(100);              ❺   ❻   ❼
  digitalWrite(13, LOW); ❽   ❾   ❿
  delay(900);
}
```

Ещё раз напомним, что не стоит пытаться воспринимать всю программу, читая сверху вниз. Сверху вниз читается только содержимое блоков. Мы вообще можем поменять порядок объявлений `setup` и `loop`.

```
void loop()
{
  digitalWrite(13, HIGH); ❷   ❸   ❹
  delay(100);              ❺   ❻   ❼
  digitalWrite(13, LOW); ❽   ❾   ❿
  delay(900);
}
void setup()
{
  pinMode(13, OUTPUT); ❶
}
```

Результат от этого не изменится ни на йоту: после компиляции вы получите абсолютно эквивалентный бинарный файл.

Выражения

Теперь попробуем понять почему написанная программа приводит в итоге к миганию светодиода.

Пины Arduino могут работать и как выходы и как входы. Когда мы хотим чем-то управлять, то есть выдавать сигнал, нам нужно перевести управляющий пин в состояние работы на выход. В нашем примере мы управляем светодиодом на 13-м пине, поэтому 13-й пин перед использованием нужно сделать выходом.

Это делается выражением в функции `setup()`:

```
pinMode(13, OUTPUT);
```

Выражения могут быть арифметическими, декларациями, определениями, условными и т.д. В данном случае мы в выражении осуществляем вызов функций `setup()` и `loop()`, но теперь мы вызываем функции, которые уже написаны где-то.

Конкретно в нашем `setup()` мы вызываем функцию с именем `pinMode()`, устанавливающая в круглых скобках заданный по номеру пин в заданный режим: вход или выход. В нашем случае 13-й пин работает как выход. `OUTPUT` означает выход, `INPUT` — вход.

Уточняющие значения, такие как 13 и `OUTPUT` называются аргументами функции. Совершенно не обязательно, что у всех функций должно быть по 2 аргумента. Сколько у функции аргументов зависит от того, что написано в самой функции. Могут быть функции с одним аргументом, тремя, двадцатью; функции могут быть без аргументов вовсе. Тогда для их вызова круглые скобка открывается и тут же закрывается:

```
noInterrupts();
```

На самом деле, вы могли заметить, наши функции `setup()` и `loop()` также не принимают никакие аргументы.

Вернёмся к нашему коду. Итак, поскольку планируется вечно мигать светодиодом, управляющий пин должен один раз быть сделан выходом и затем мы не хотим вспоминать об этом. Для этого идеологически и предназначена функция `setup`: настроить плату как нужно, чтобы затем с ней работать.

Перейдём к функции `loop()`:

```
void loop()
{
    digitalWrite(13, HIGH);
    delay(100);
    digitalWrite(13, LOW);
    delay(900);
}
```

Она, как говорилось, вызывается сразу после `setup()`. И вызывается бесконечно, после того, как сама заканчивается. Функция `loop()` называется основным циклом программы и предназначена для выполнения полезной работы. В нашем случае полезная работа — мигание светодиодом.

Первое выражение — это вызов встроенной функции `digitalWrite()`. Она предназначена для подачи на заданный пин логического нуля (LOW, 0 вольт) или логической единицы (HIGH, 5 вольт) В функцию `digitalWrite()` передаётся 2 аргумента: номер пина и логическое значение. В итоге, первым делом мы зажигаем светодиод на 13-м пине, подавая на него 5 вольт.

Как только это сделано, процессор моментально приступает к следующему выражению – вызов функции `delay()` — это, опять же, встроенная функция, которая заставляет процессор «уснуть» на определённое время. Она принимает всего один аргумент: время в миллисекундах, которое следует спать. В нашем случае это 100 мс.

Пока мы спим всё остаётся как есть, т.е. светодиод продолжает гореть. Как только 100 мс истекают, процессор просыпается и тут же переходит к следующему выражению. В нашем примере это снова вызов знакомой нам встроенной функции `digitalWrite()`. Но на этот раз вторым аргументом мы передаём значение LOW. То есть устанавливаем на 13-м пине логический ноль, то есть подаём 0 вольт, то есть гасим светодиод.

После того, как светодиод погашен, приступаем к следующему выражению. И снова это вызов функции `delay()`. На этот раз мы «засыпаем» на 900 мс. После этого функция `loop()` завершается, но вызывается её ещё раз и всё происходит снова: светодиод поджигается, горит, гаснет, ждёт и т.д.

Если перевести написанное на русский, получится следующий алгоритм:

- Поджигаем светодиод
- Спим 100 миллисекунд
- Гасим светодиод
- Спим 900 миллисекунд
- Переходим к пункту 1

Таким образом мы получили Arduino с маячком, мигающим каждые 100 + 900 мс = 1000 мс = 1 сек.

Изменим программу

Пользуясь только полученными знаниями сделаем несколько вариаций программы, чтобы лучше понять принцип работы.

Можно подключить внешний светодиод или другое устройство, которым нужно «мигать» на другой пин. Например, на 5-й. В этом случае должна измениться программа: всюду, где обращались к 13-му пину, нужно заменить номер на 5-й:

```
void setup()
{
  pinMode(5, OUTPUT);
}
```

```

void loop()
{
  digitalWrite(5, HIGH);
  delay(100);
  digitalWrite(5, LOW);
  delay(900);
}

```

Компилируем, загружаем, проверяем.

Сделаем так, чтобы светодиод мигал 2 раза в секунду. Уменьшить время «сна» так, чтобы в сумме получилось 500 мс:

```

void loop()
{
  digitalWrite(5, HIGH);
  delay(50);
  digitalWrite(5, LOW);
  delay(450);
}

```

Сделаем так, чтобы в устройстве было 2 светодиода, которые мигали бы каждую секунду поочередно? Нужно будет общаться с двумя пинами и работать в loop() то с одним, то с другим:

```

void setup()
{
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
}

void loop()
{
  digitalWrite(5, HIGH);
  delay(100);
  digitalWrite(5, LOW);
  delay(900);
  digitalWrite(6, HIGH);
  delay(100);
  digitalWrite(6, LOW);
  delay(900);
}

```

Сделаем так, чтобы в устройстве были 2 светодиода, которые переключались бы на манер железнодорожного светофора: горел бы то один то другой. Нужно просто не выключать горящий светодиод тут же, а дожидаться момента переключения:

```

void setup()
{
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
}

void loop()
{
  digitalWrite(5, HIGH);
  digitalWrite(6, LOW);
  delay(1000);
  digitalWrite(5, LOW);
  digitalWrite(6, HIGH);
  delay(1000);
}

```

```
}
```

Проверьте другие идеи самостоятельно.

Правильное оформление кода

В языке C++ пробелы, переносы строк, символы табуляции не имеют большого значения для компилятора. Там, где стоит пробел, может быть перенос строки и наоборот. На самом деле 10 пробелов подряд, 2 переноса строки и ещё 5 пробелов — это всё эквивалент одного пробела.

Пустое пространство — это инструмент программиста, с помощью которого можно или сделать программу понятной и наглядной, или изуродовать до неузнаваемости. Например, вспомним программу для мигания светодио́дом:

```
void setup()
{
    pinMode(5, OUTPUT);
}

void loop()
{
    digitalWrite(5, HIGH);
    delay(100);
    digitalWrite(5, LOW);
    delay(900);
}
```

Мы можем изменить её так:

```
void setup(
)
{
    pinMode(5, OUTPUT);
}

void loop
() {
digitalWrite(5,HIGH);
delay(100
)
;
    digitalWrite(5,LOW);
    delay(900); }
```

Всё, что мы сделали — немного «поработали» с пустым пространством.

Точки с запятыми

Зачем в конце каждого выражения ставится точка с запятой? Таковы правила языка C++. Подобные правила называются синтаксисом языка. По символу « ; » компилятор понимает где заканчивается выражение.

Как уже говорилось, переносы строк для него — пустой звук, поэтому ориентируется он на этот знак препинания. Это позволяет записывать сразу несколько выражений в одной строке:

```
void loop()
{
    digitalWrite(5, HIGH); delay(100); digitalWrite(5, LOW); delay(900);
}
```

Программа корректна и эквивалентна тому, что мы уже видели. Однако писать так — это дурной тон. Код гораздо сложнее читается. Поэтому если у вас нет 100% веских причин писать в одной строке несколько выражений, не делайте этого.

Комментарии

Одно из правил качественного программирования: «пишите код так, чтобы он был настолько понятным, что не нуждался бы в пояснениях». Это возможно, но не всегда. Для того, чтобы пояснить какие-то не очевидные моменты в коде его читателям: вашим коллегам или вам самому через месяц, существуют так называемые комментарии.

Это конструкции в программном коде, которые полностью игнорируются компилятором и имеют значение только для читателя. Комментарии могут быть многострочными или однострочными:

```
/*  
    Функция setup вызывается самой первой,  
    при подаче питания на Arduino  
  
    А это многострочный комментарий  
*/  
void setup()  
{  
    // устанавливаем 13-й пин в режим вывода  
    pinMode(13, OUTPUT);  
}  
  
void loop()  
{  
    digitalWrite(13, HIGH);  
    delay(100); // спим 100 мс  
    digitalWrite(13, LOW);  
    delay(900);  
}
```

Как видите, между символами `/*` и `*/` можно писать сколько угодно строк комментариев. А после последовательности `//` комментарием считается всё, что следует до конца строки.

Выводы

Итак, надеемся основные принципы составления написания программ стали понятными. Полученные знания позволят управлять подачей питания на пины Arduino по определённым временным схемам. Это не так уж много, но всё же достаточно для первых экспериментов.

8 Урок № 5 Сборка манипулятора

Общеобразовательный конструктор для практического изучения принципов создания электронных устройств на основе электронных компонентов и программируемых контроллеров позволяет осуществить сборку манипуляционных роботов с угловой кинематикой, плоскопараллельной кинематикой, дельта-кинематикой.

Манипуляторы с угловым типом кинематической схемы

Манипуляторы с угловым типом кинематической схемы (рис. 4-1) могут применяться для свободного перемещения рабочего инструмента в пространстве в рамках сферической рабочей зоны относительно системы координат основания. В большинстве случаев подобные манипуляторы применяются для выполнения задач по перемещению и точному позиционированию объектов, обработки пространственных контуров и поверхностей. Данный тип кинематики является наиболее распространенным в промышленности в области машиностроения и механической обработки, сварочных и покрасочных работ.

NOBOTS 1 в сборке угловой кинематики представляет собой доступную учебную модель промышленного манипуляционного робота, обеспечивающую возможность на практике изучить большинство аспектов разработки систем управления манипуляционных роботов с угловым типом

кинематики, и базовых принципов применения робототехники при автоматизации технологических процессов.

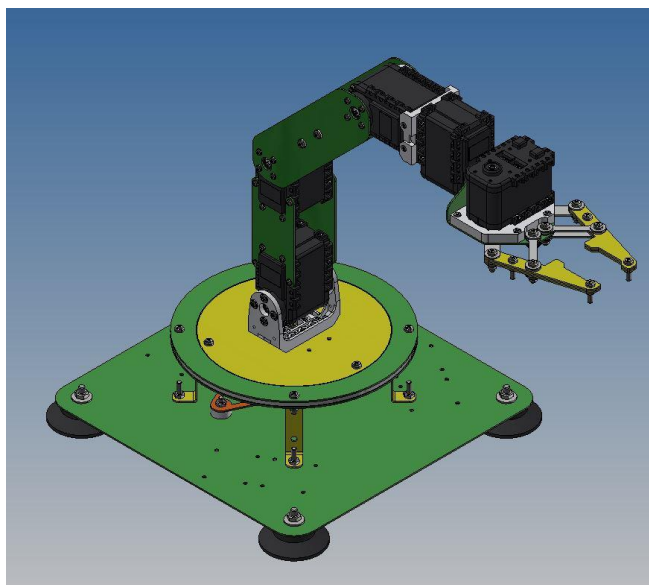


Рисунок 8-1 Манипуляционный робот с угловой кинематикой

Манипуляторы с плоско-параллельным типом кинематической схемы

Манипуляторы с плоско-параллельным типом кинематической схемы (рис. 4-2) могут применяться для перемещения рабочего инструмента параллельно горизонтальной плоскости в цилиндрической системе координат относительно основания. Важным преимуществом данной кинематики является повышенная грузоподъемность на захвате за счет облегчения конструкции манипулятора ближе к захвату за счет расположения приводов в основании. В большинстве случаев подобные манипуляторы применяются для выполнения задач по перемещению грузов большой массы в машиностроении или логистике.

NOBOTS 1 в сборке плоско-параллельной кинематики представляет собой доступную учебную модель промышленного манипуляционного робота, обеспечивающую возможность на практике изучить большинство аспектов разработки систем управления манипуляционных роботов с плоско-параллельной кинематикой, и базовых принципов применения робототехники при автоматизации технологических процессов.

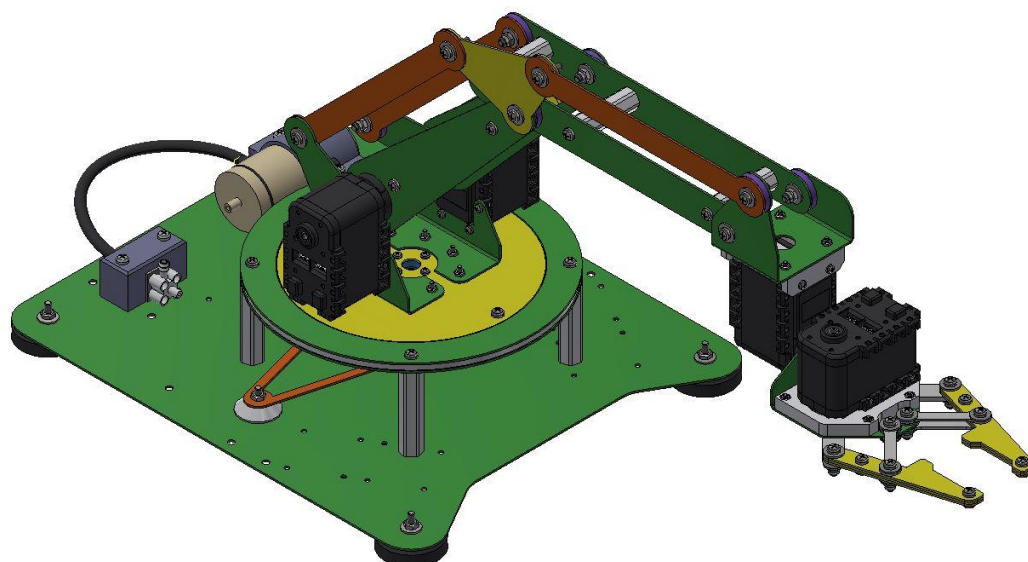


Рисунок 8-2 Манипуляционный робот с плоско-параллельной кинематикой

Манипуляторы с дельта-кинематической схемой

Манипуляторы с дельта-кинематической схемой (рис. 4-3) могут применяться для перемещения рабочего инструмента с высокой скоростью. Высокая скорость манипулирования лёгкими объектами, достигается за счёт того, что тяжёлые приводы расположены на неподвижном основании, а все двигающиеся звенья механизма выполнены из лёгких, зачастую, композитных материалов. Рабочее звено дельта-робота соединено с неподвижным основанием тремя независимыми кинематическими цепями, а сам механизм воспринимает нагрузку, как пространственная ферма

NOBOTS 1 в сборке дельта-кинематики представляет собой доступную учебную модель промышленного манипуляционного робота, обеспечивающую возможность на практике изучить большинство аспектов разработки систем управления манипуляционными роботами с дельта-кинематикой, и базовых принципов применения робототехники при автоматизации технологических процессов.

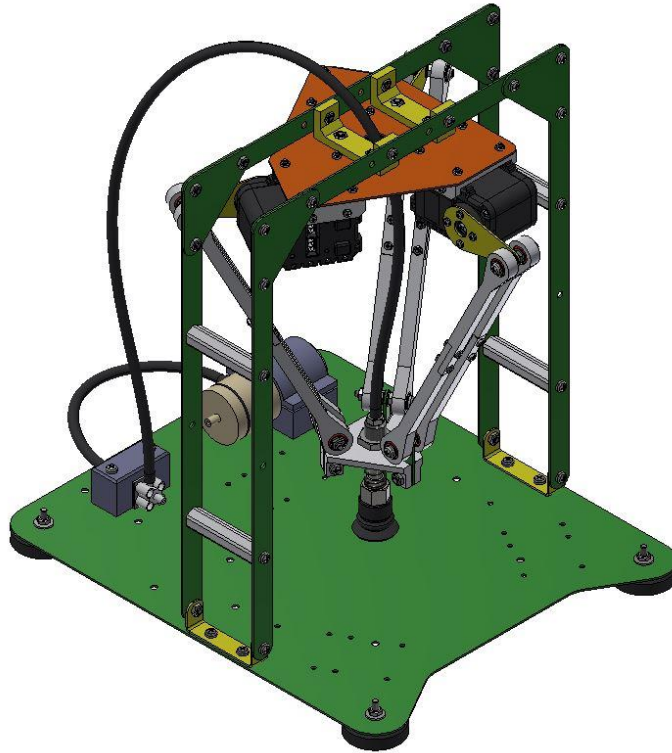


Рисунок 8-3 Манипулятор с дельта-кинематикой

Платформа Стюарта

Платформа Стюарт представляет собой параллельный манипулятор с 6-независимыми опорами площадки в которых шарниры используются для соединения всех 12 точек (Рисунок 8-4).

Предметы, установленные на верхней плите, могут перемещаться со всеми шестью степенями свободы, доступными для свободно подвешенного тела: три линейных – поперечная, продольная и вертикальная, а также три вращения – тангаж, крен и рыскание.

Эта кинематика применяется в авиасимуляторах, станках, роботах и т.д.



Рисунок 8-4 Платформа Стюарта

SCARA – рычажная кинематика

Scara - сокращение от англ. Selective Compliance Assembly Robot Arm (сборочный робот-манипулятор с избирательной приспособляемостью) Рисунок 8-5. Все сочленения этих устройств, представляющих собой модифицированный вариант манипуляторов с цилиндрической системой координат, располагаются в горизонтальной плоскости, благодаря чему механизм способен "разворачиваться" подобно складной ширме.

Его зона действия имеет цилиндрическую форму. Роботы типа SCARA отличаются высокими быстродействием и точностью.

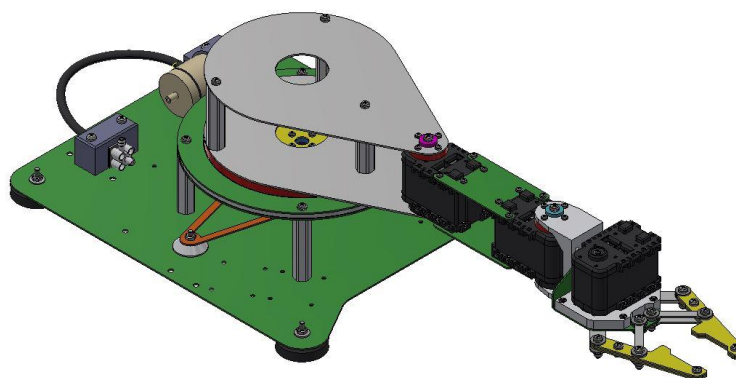


Рисунок 8-5 робот SCARA

Подробная инструкция по сборке манипуляционных роботов представлена на сайте производителя www.hobots1.ru

9 Урок № 6 Подключение манипулятора

Для подключения электронных плат базового манипуляционного конструктора соедините их согласно схеме, изображенной на рисунке 8.

В комплектации набора возможно изменение одноплатного компьютера Raspberry Pi на Orange Pi, в таком случае стоит обратиться к актуальной информации на сайте www.hobots1.ru

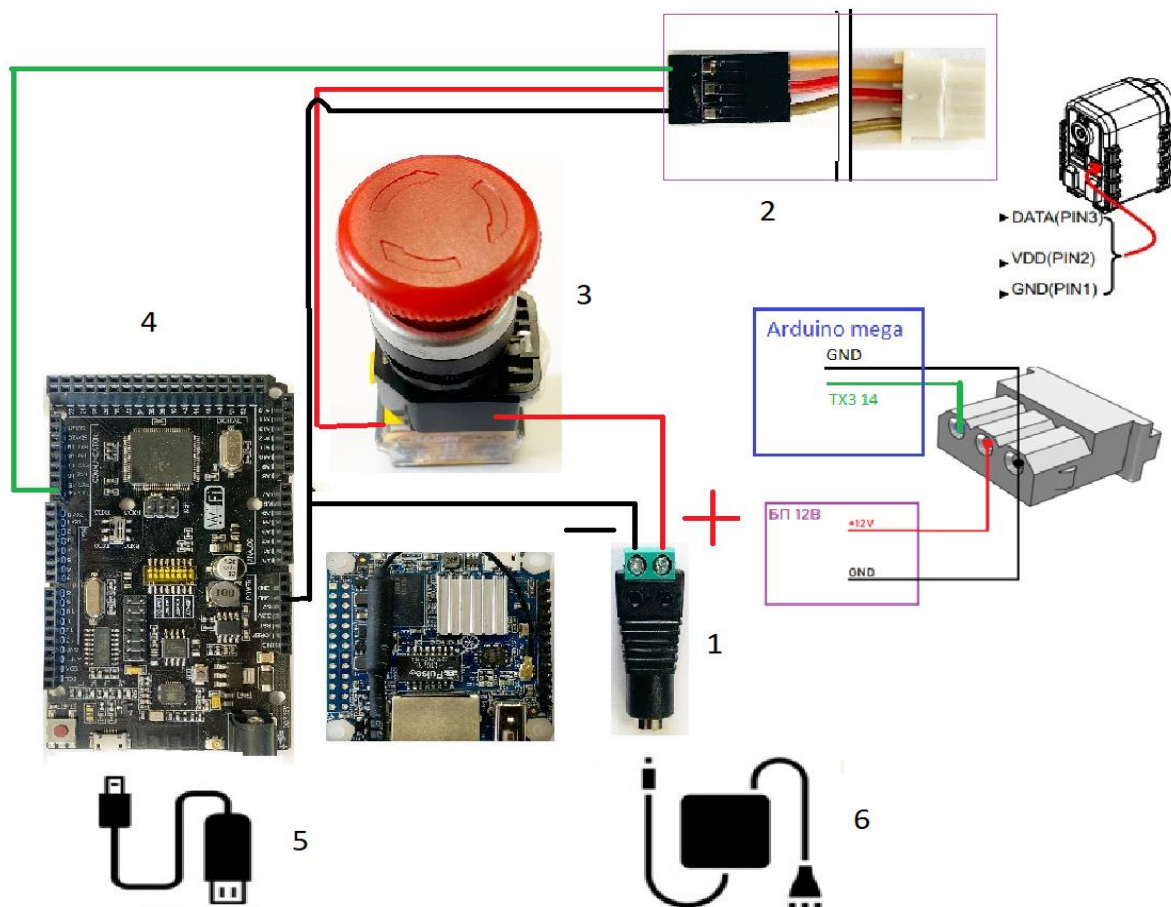


Рисунок 9-1 Принципиальная схема аппаратной части Базового конструктора манипуляционных устройств «HOBOTS»

1. Разъем питания; 2. Провод «dupont-dynamiXel»; 3. Выключатель питания; 4. Плата Arduino mega; 5. USB кабель; 6. Блок питания.

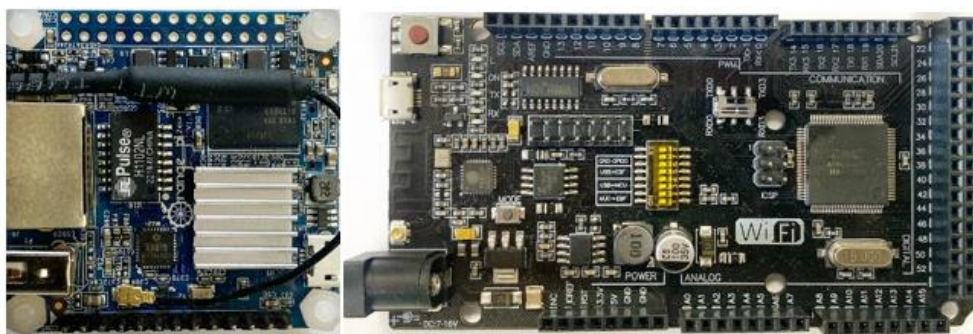
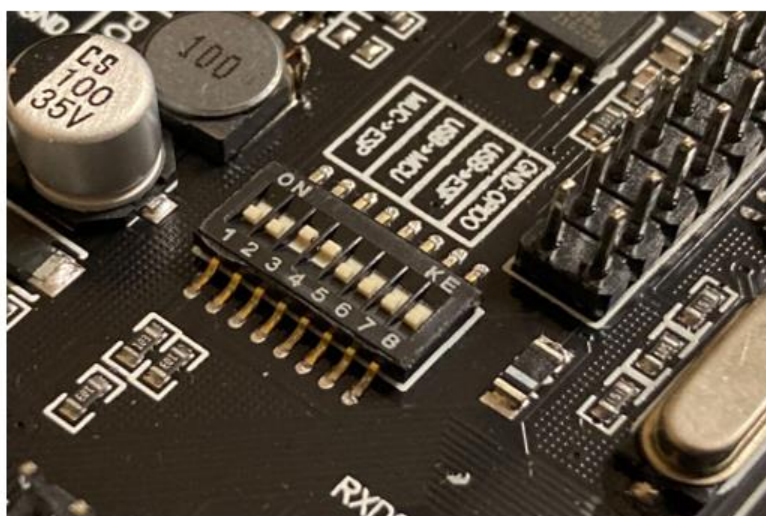


Рисунок 9-2 Базовые электронные платы конструктора манипуляционных устройств «HOBOTS»
Одноплатный компьютер Orange Pi и Arduino MEGA



Соединение	1	2	3	4	5	6	7
1. USB <-> ESP8266 (загрузка скетча)	OFF	OFF	OFF	OFF	ON	ON	ON
2. USB <-> ESP8266 (соединение)	OFF	OFF	OFF	OFF	ON	ON	OFF
3. USB <-> ATmega2560	OFF	OFF	ON	ON	OFF	OFF	OFF
4. USB <-> ATmega2560 <-> ESP8266	ON	ON	ON	ON	OFF	OFF	OFF
5. ATmega2560 <-> ESP8266	ON	ON	OFF	OFF	OFF	OFF	OFF
6. Все независимы	OFF	OFF	OFF	OFF	OFF	OFF	OFF

Рисунок 9-3 Установка переключателей на плате Arduino Mega

Сборка аппаратной части конструктора производится при отключенном питании.

С разъема питания 5.5 мм типа «мама» (№ 1 на рисунке №8) контакт «+» подключается к центральному контакту разъема «dupont» провода «dupont-dynamixel» (№ 2 на рисунке №8) через контакты 11 и 12 экстренного выключателя питания (№ 3 на рисунке №8). В случае возникновения нештатной ситуации путем нажатия на выключатель происходит отключение питания от сервоприводов конструктора. Управляющий провод с контакта «TX3 14» платы Arduino mega подключается к крайнему контакту разъема «dupont» провода «dupont-dynamixel» отмеченного стрелкой на корпусе разъема. Контакт «-» с разъема питания 5.5 мм типа «мама» подключается к

оставшемуся свободным контакту разъема «dupont» провода «dupont-dynamixel» и контакту «GND» платы Arduino mega. Назначение контактов разъема, подключаемого к сервоприводу представлены на рисунке 8.

10 Урок №7. Программное обеспечение NOBOTS

Для управления манипулятором надо установить программу NOBOTS, которую можно скачать с официального сайта.

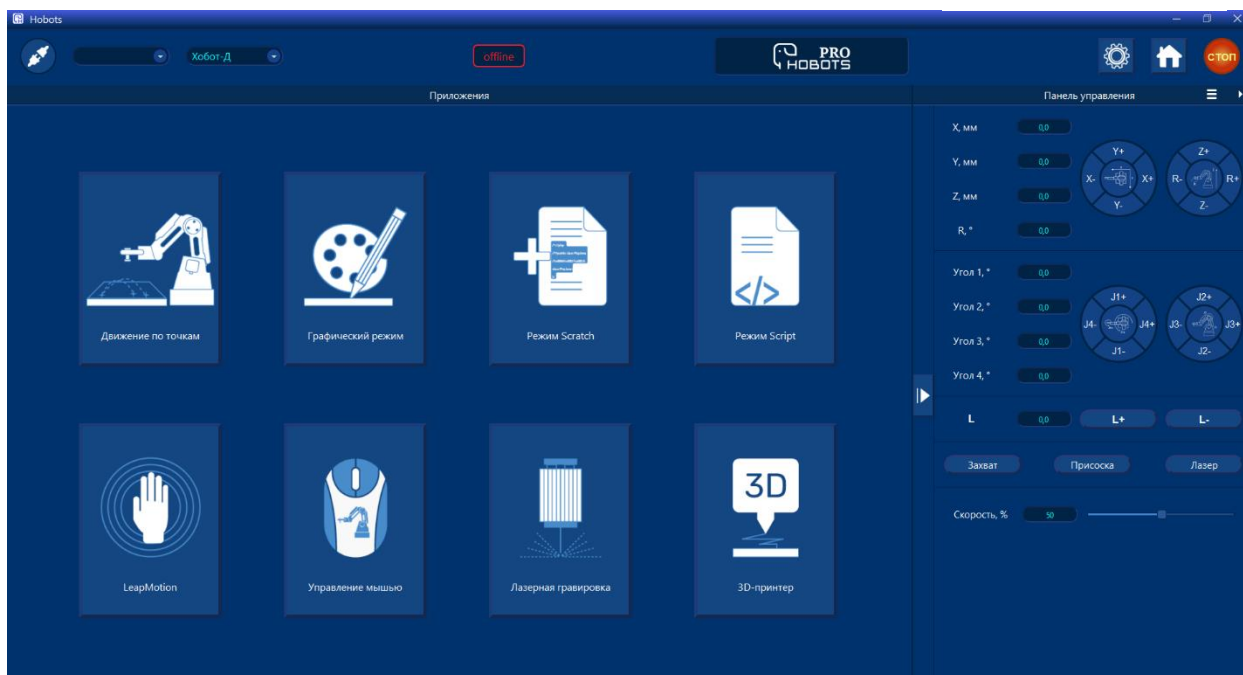
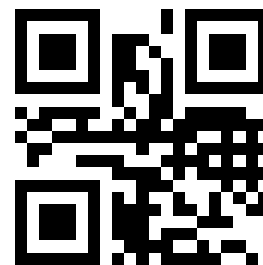


Рисунок 10-1 Основной экран

Конструктор Hobots 1 гибок в управлении — им можно управлять как в режиме реального времени, так и применяя программирование с помощью средств языков C/C++, Python и свободно распространяемой среды Arduino IDE. В приложении представлены модели с плоскопараллельной, угловой и дельта-кинематикой (рис. 9-2, 9-3, 9-4)

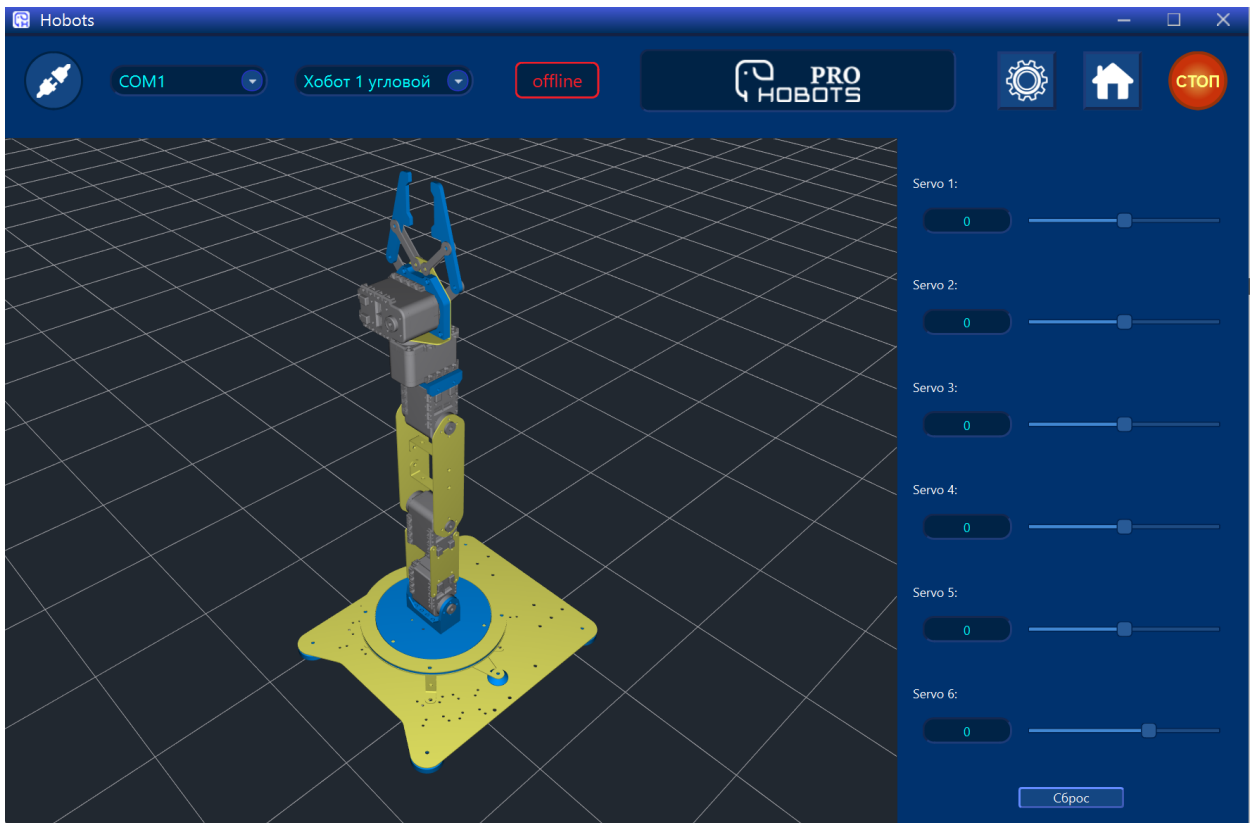


Рисунок 10-2 Программное обеспечение HOBOTS 1 с угловой кинематикой

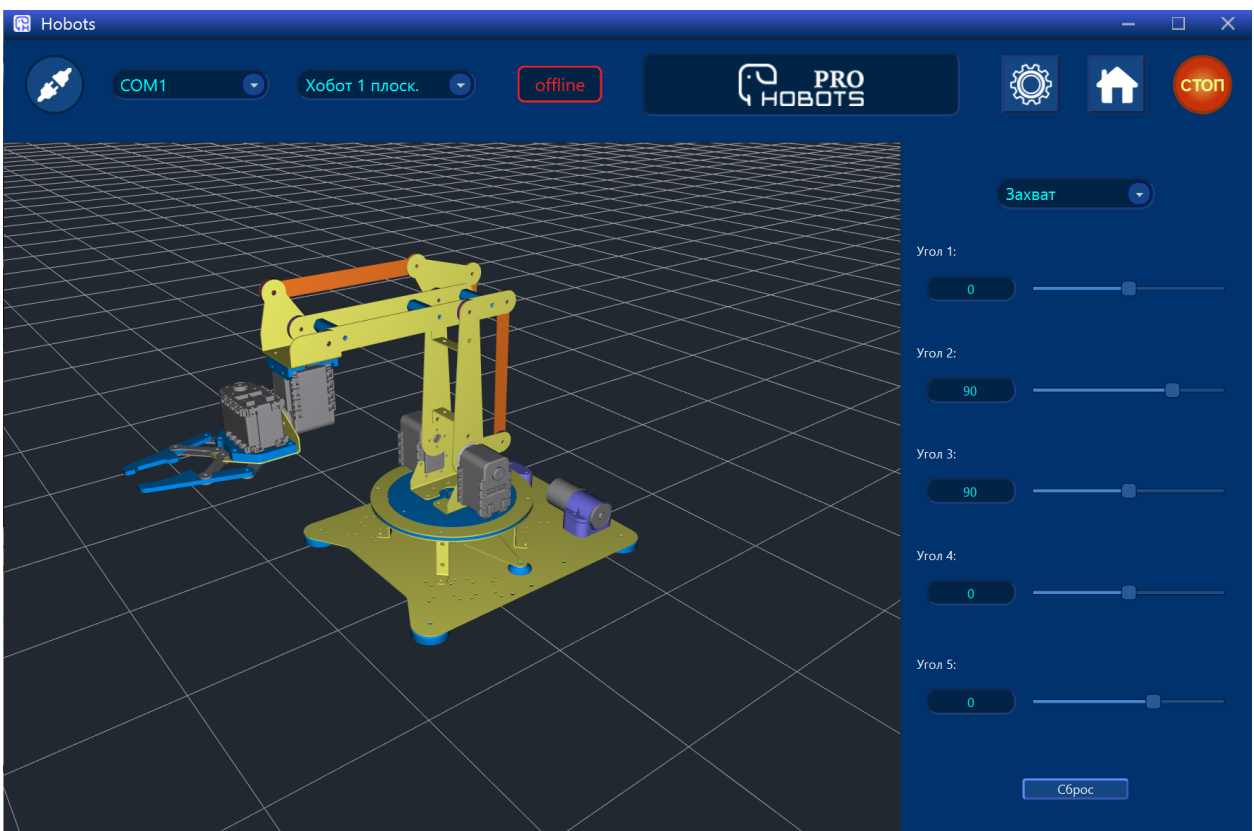


Рисунок 10-3 Программное обеспечение HOBOTS 1 с плоско-параллельной кинематикой

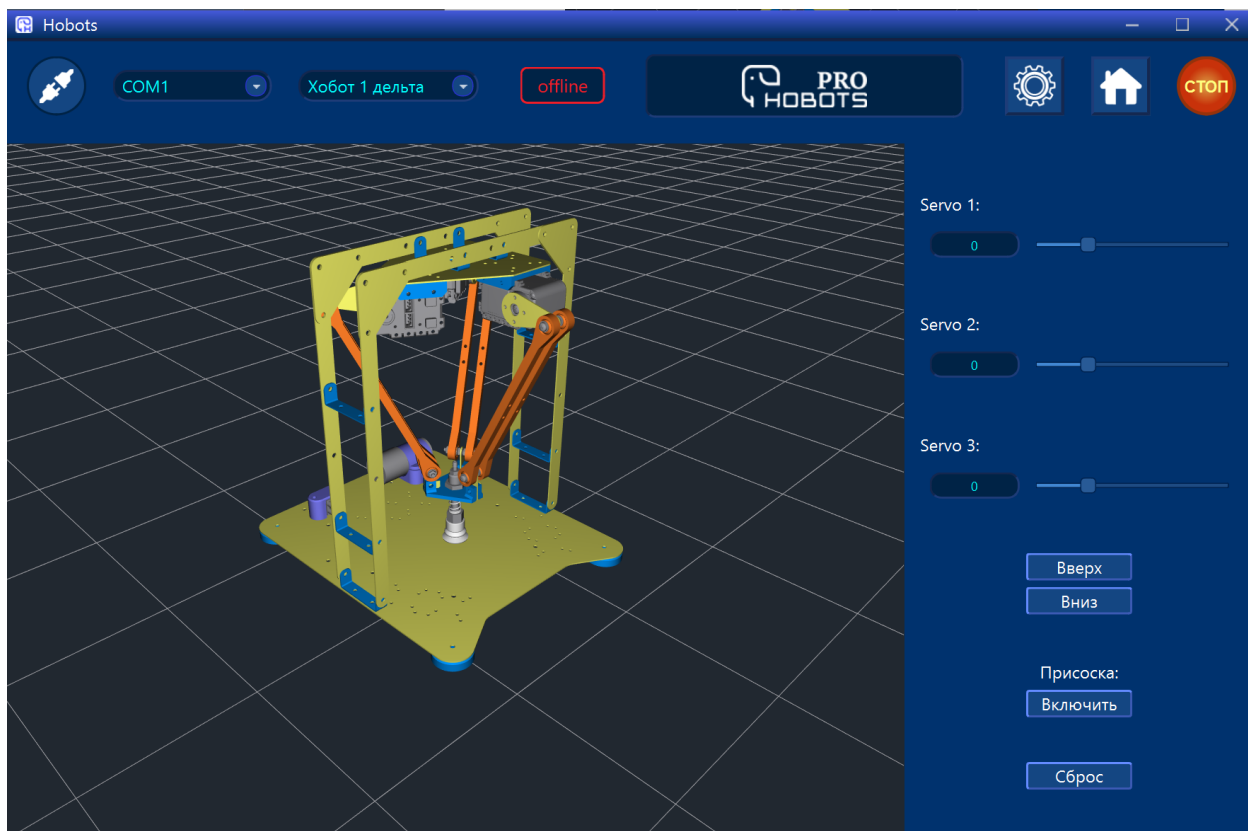


Рисунок 10-4 Программное обеспечение ROBOTS 1 с дельта-кинематикой

11 Урок № 8. Техническое зрение

Описание работы модуля компьютерного зрения MES Visio

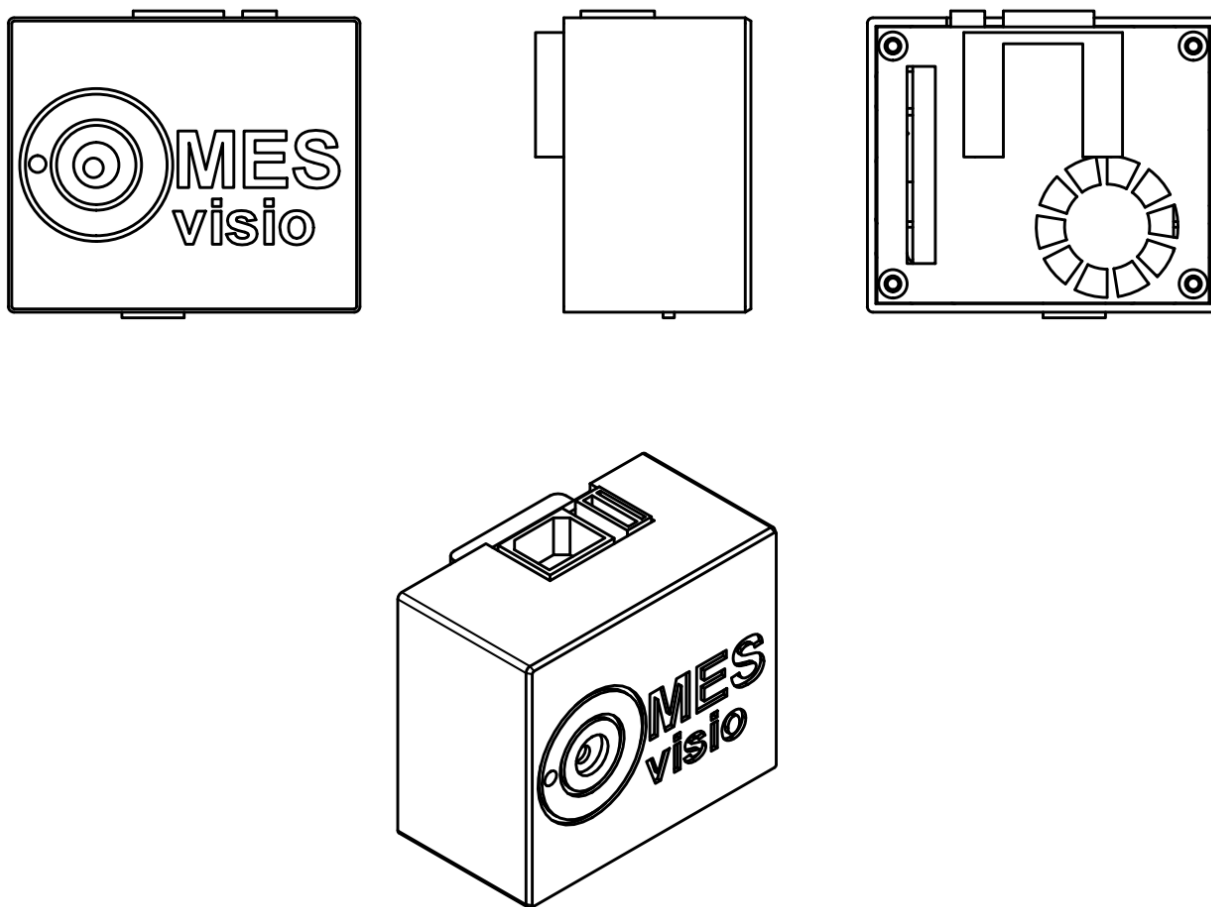


Рисунок 10-5 Модуль технического зрения MES Visio

Для подключения по Wi-Fi требуется компьютер или телефон с поддержкой Wi-Fi, порядок запуска:

1. Подключить к разъему microUSB модуля технического зрения кабель питания.
2. Подключить кабель питания к внешнему источнику питания 5В.
3. Через 30-60 секунд после включения питания подключиться с персонального компьютера к Wi-Fi сети hobotscv, пароль для подключения hobots123.
4. В браузере на подключенном к указанной Wi-Fi сети перейти по адресу <http://192.168.0.4:5000/>. Рекомендованный браузер – Google Chrome.
5. В браузере должен стать доступен интерфейс программы Технического зрения.

Для подключения по USB требуется компьютер с свободным портом USB 2.0, инструкция актуальна для подключения к компьютеру под управлением WINDOWS 10, порядок запуска:

1. Соединить USB-ETHERNET адаптер с модулем технического зрения с помощью патч-корда, при правильном подключении патч-корда должен быть слышен щелчок.
2. Воткнуть USB-ETHERNET адаптер в USB порт компьютера.
3. Подключить к разъему microUSB модуля технического зрения кабель питания.
4. Подключить кабель питания к внешнему источнику питания 5В.
5. Через 30-60 секунд после включения питания перейти в Панель управления Windows.
6. В вкладке Сеть и интернет перейти в пункт Просмотр состояния сети и задач.
7. В вкладке Просмотр активных сетей найти сеть Ethernet, которая относится к подключенному модулю.
8. В подпункте Подключения найденной сети нажать на активную ссылку.
9. В открывшемся окне перейти в Свойства.
10. Нажать на пункт IP версии 4, нажать Свойства.
11. Поставить галочку Использовать следующий IP адрес.
12. Выполнить настройку как на Рисунок 10-6.

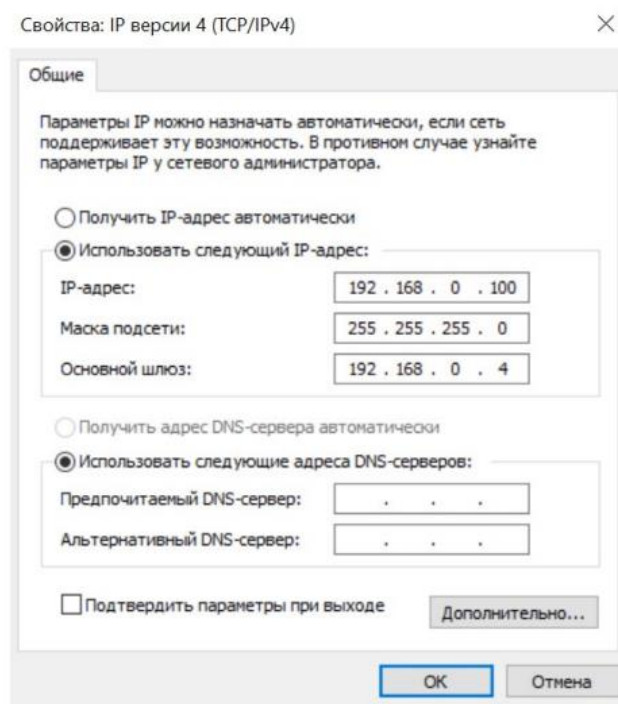


Рисунок 10-6 Параметры IP

13. После настройки нажать Ок и Закрывать в следующем окне.
14. Далее требуется отключить модуль технического зрения от питания, подождать 15 секунд и подключить питание обратно.

15. В браузере после полной настройки перейти по адресу <http://192.168.0.4:5000/>. Рекомендованный браузер – Google Chrome.
16. В браузере должен стать доступен интерфейс программы Технического зрения.

Основные настройки технического зрения.

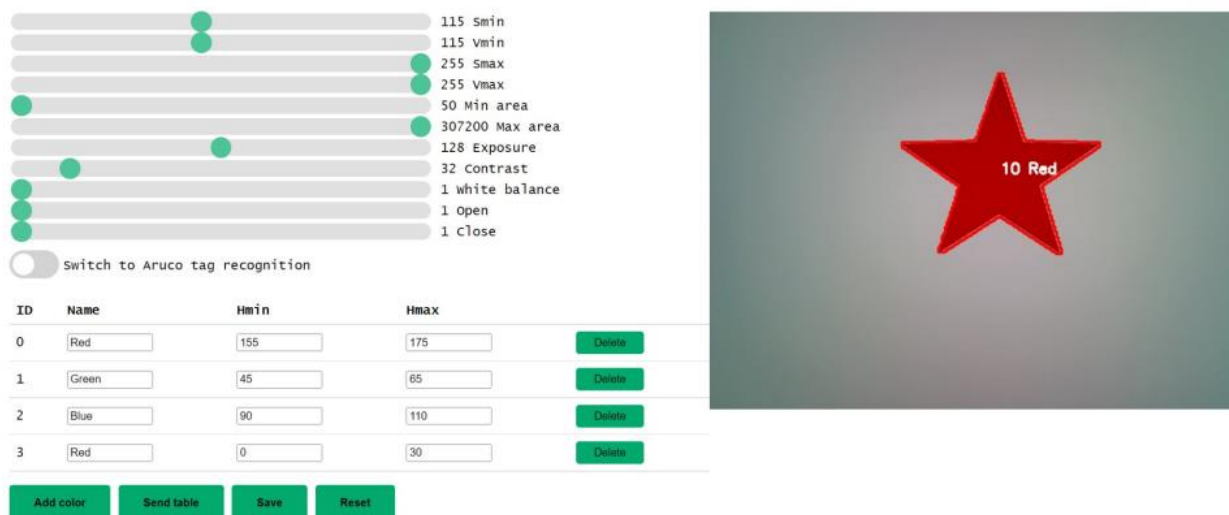


Рисунок 10-7 Основной экран веб-интерфейса устройства

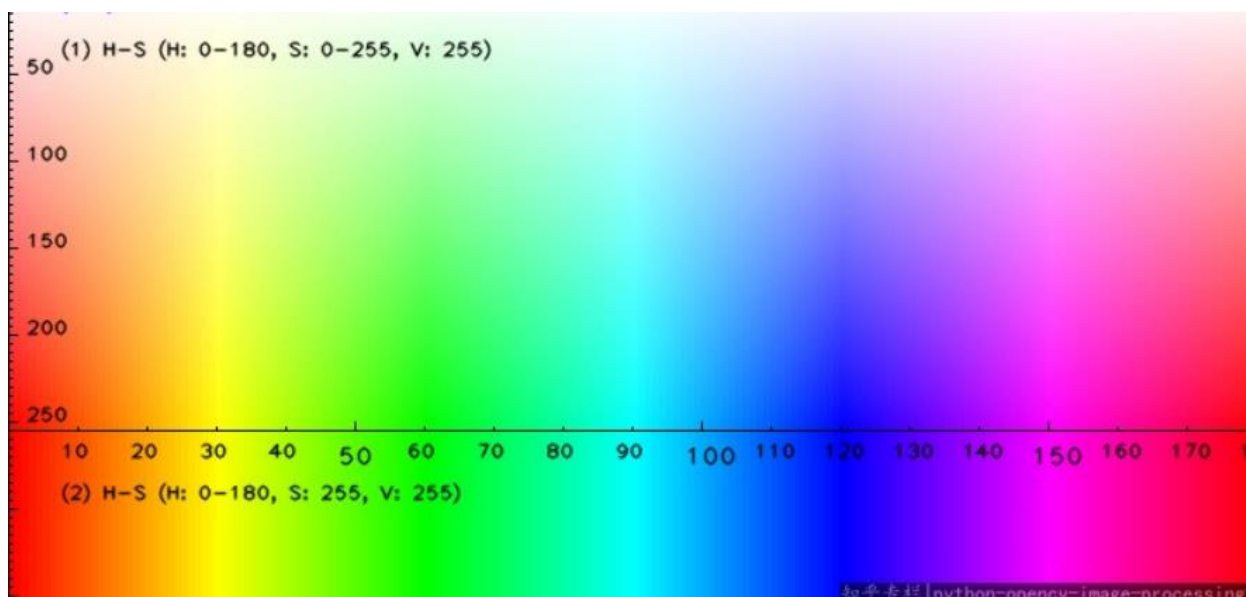


Рисунок 10-8 Визуализация палитры HSV, по X – H составляющая, по Y – S составляющая

HSV цветовая модель

HSV (англ. Hue, Saturation, Value — тон, насыщенность, значение) цветовая модель, в которой координатами цвета являются:

- Hue — цветовой тон, (например, красный, зелёный или сине-голубой). Варьируется в пределах 0—180.
- Saturation — насыщенность. Варьируется в пределах 0—255. Чем больше этот параметр, тем «чище» цвет, поэтому этот параметр иногда

называют чистотой цвета. Чем ближе этот параметр к нулю, тем ближе цвет к нейтральному серому.

- Value (значение цвета) или Brightness — яркость. Также задаётся в пределах 0—255.

Параметры технического зрения:

Smin, Vmin – задает цветоразностные составляющие, минимальный порог

распознаваемого цвета, а именно показатели S и V в палитре HSV.

Smax, Vmax – задает цветоразностные составляющие, максимальный порог

распознаваемого цвета, а именно показатели S и V в палитре HSV.

Min area – задает минимальный порог площади распознаваемых фигур: окружностей, квадратов и треугольников, в пикселях.

Max area – задает максимальный порог площади распознаваемых фигур: окружностей, квадратов и треугольников, в пикселях.

Contrast – задает контрастность камеры.

White balance – задает баланс белого камеры, 1 – автоматический баланс белого.

Open – позволяет выполнять настройку округлости обнаруживаемой области изображения, положение обнаруживаемых областей относительно друг друга, кривизны и положения распознаваемых линий. Также она частично реализована на основе алгоритмов машинного обучения, значит позволяет выполнять настройку нейронной сети (Рисунок 10-9).

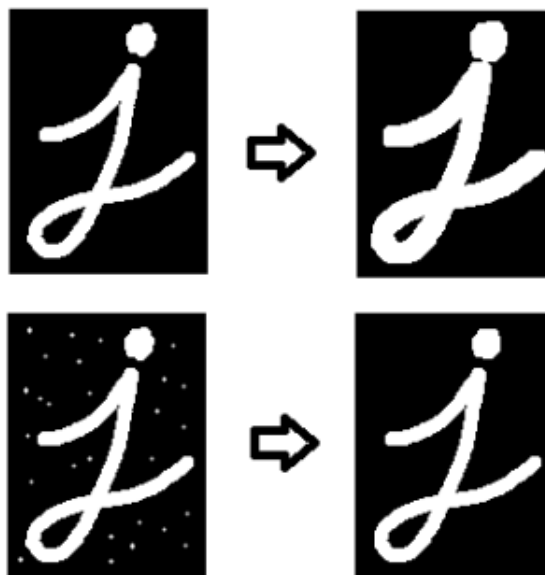


Рисунок 10-9 Пример настройки параметра open

Close – повторяет функционал предыдущего пункта, но в обратную сторону (Рисунок 10-10).

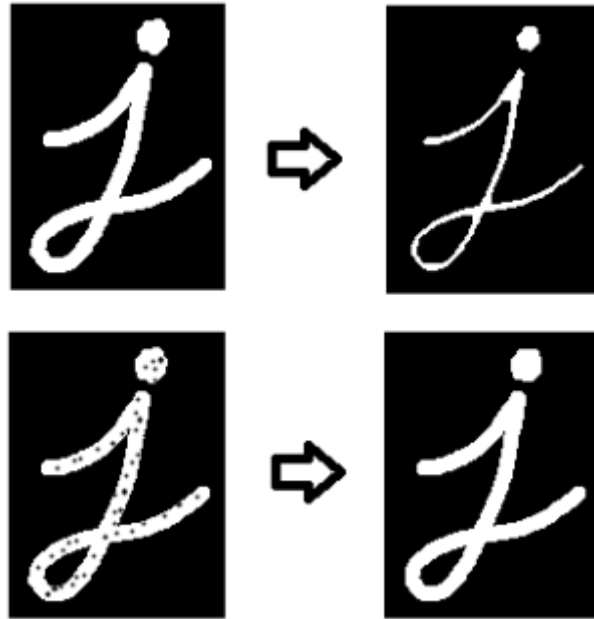


Рисунок 10-10 Пример настройки параметра close

Switch to Aruco tag recognition – переключает модуль в режим распознавания Арукометок, с выводом их закодированного значения.

Таблица для настройки распознаваемых цветов.

- ID – порядковый номер цвета
- Name – названия цвета
- Hmin, Hmax – от и до каких значений H в палитре HSV модуль компьютерного зрения будет называть цвет фигуры обозначенным именем.

Программа позволяет добавлять несколько диапазонов тона, которому пользователь может назначить любое наименование. По умолчанию программа предлагает три диапазона: Green (H = 45..65), Blue (H = 90..110), Red (H = 155..175).

Сохранение и сброс параметров

- При изменении параметров с ползунками они применяются автоматически, и сохраняются в оперативную память, что бы их сохранить в постоянную, требуется нажать кнопку Save.

- Для того что бы сбросить параметры до последних сохраненных в постоянной памяти, требуется нажать кнопку Reset.

- При работе с таблицей, для добавления еще одного цвета, требуется нажать кнопку Add color.

- Для сохранения таблицы в оперативную память, и применить настройки, требуется нажать кнопку Send table.

- Таблица сохраняется в постоянную память, как и остальные параметры по нажатию кнопки Save.

Доступ в консоль

- Консоль расположена в веб интерфейсе по адресу:
`http://192.168.0.4/web_ssh`
- Логин root, пароль hobots .

Работа с консолью

- Консоль является стандартным терминалом linux
- Для передачи и исполнения команды требуется нажать Enter
- Для работы с файлами используются команды:
- `cd` – перейти в определенную директорию
- `rm` – для удаления файлов
- `rmdir` – для удаления папок
- `sudo apt-get update` — обновление системы
- `killall {название процесса}` — для принудительной остановки определенного процесса
- `armbianmonitor -m` — для просмотра основных системных параметров
- `sudo armbian-config` – для настройки основных системных параметров, в том числе и wifi.

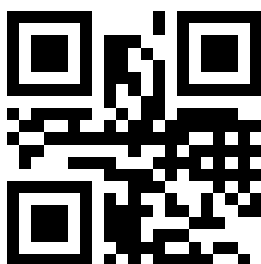
Nobots 1 отлично подойдет для обучения. Учебный процесс следует начинать со сборки. Следующий шаг — основы программирования.

Столь наглядное и интерактивное обучение намного эффективнее работы с “сухим” кодом, особенно в самом начале.

Учитывая интересы более продвинутых потенциальных пользователей, предусмотрена возможность подключения разнообразной периферии. Это могут быть всевозможные датчики, сервоприводы и шаговые двигатели, светодиодные приборы и множество других электронных устройств, включая микрокомпьютеры и микроконтроллерные платформы. Учитывая это, можно рекомендовать Nobots 1 не только школам, но и техникумам, и вузам, как исключительно удачное наглядное пособие.

Эта инженерная платформа нового поколения позволит Вам уверенно продвигаться в изучении промышленной робототехники и методов автоматизации.

Желаем Вам успехов в работе с NOBOTS 1!



Коллектив компании
MES Group
2023 г.