

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Локтионова Оксана Геннадьевна

Должность: проректор по учебной работе

Дата подписания: 26.07.2022 10:13:58

Уникальный программный ключ:

0b817ca911e6668abb13a5d426d39e5f1c11eabb7624311447851ba56d089

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
ВОЛГОГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

На правах рукописи

**Драгунов Станислав Евгеньевич
Матохина Анна Владимировна**

Разработка алгоритма автономного функционирования робота.

*Методические указания для лабораторных и практических работ
по дисциплине «Системы искусственного интеллекта»*

Волгоград 2021

Задание на лабораторную работу: в соответствии с моделью робота, разработать алгоритм функционирования робота манипулятора с шестью степенями свободы или четырехногого робота в зависимости от задачи.

Задание на практическую работу: провести тестирование функционирования робота, обосновать результаты тестирования.

- Разработка алгоритма движения робота.
- Разработка системы дистанционного управления роботом посредством Bluetooth модуля
- Разработка системы балансировки робота

Дополнительное задание

- Установка внешних датчиков
- Разработка системы автономного функционирования

Описание модели робота.

В качестве учебной модели используется модель робота “собаки”. Робот состоит из 4 конечностей, и туловища. на каждую ногу приходится по 3 сервопривода тип SG-90 вращающимся на 180 градусов. В качестве управляющей платы используется контроллер Arduino Uno R3, для подключения 12 сервоприводов используется плата расширения servo shield, приёмо - передача осуществляется посредством bluetooth модуля, ориентация в пространстве осуществляется с помощью гироскопа



Рисунок 1- робот в сборе

Подготовка

В начале сборки робота, необходимо проверить корректности печати деталей, их совместимость с сервоприводами,



Рисунок 2. Сборка ног

Перед установкой сервоприводов с помощью Arduino/ESP32 или servo tester необходимо установить все сервоприводы в среднее положение (90 градусов). Далее сервоприводы соединяют части ноги, что изображено на Рисунке 2.

При сборке, колен и ног направлены назад относительно робота. После установки сервопривода необходимо протянуть сервокабель через одно из отверстий части. Далее повторить с другими ногами.

Тело и ноги вместе

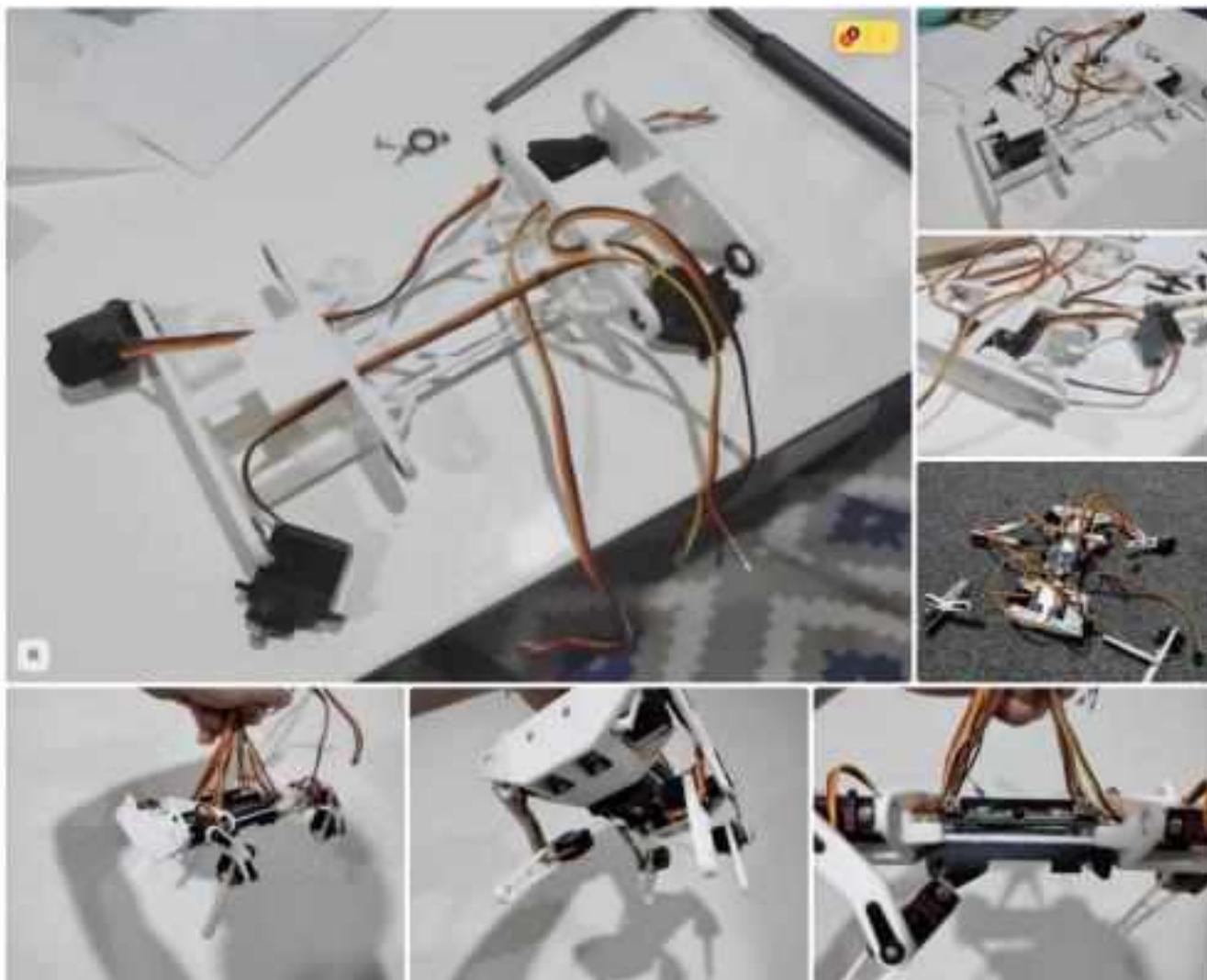


Рисунок 3. сборка тела и корпуса

На следующем этапе сборки производится соединение деталей корпуса, как показано на Рисунке 4. Тело состоит из колодок под ноги и центральной панели.

Электронные компоненты

Arduino Uno

Arduino Uno построена на базе микроконтроллера ATmega328. Платформа имеет 14 цифровых вводов/ выходов (6 из которых могут использоваться как выходы ШИМ), 6 аналоговых входов, кварцевый генератор 16 МГц, разъем USB, силовой разъем, разъем ICSP и кнопку перезагрузки. Для работы необходимо подключить платформу к компьютеру посредством кабеля USB или подать питание при помощи адаптера AC/DC, или аккумуляторной батареей.

Микроконтроллер	ATmega328
Рабочее напряжение	5В

Напряжение питания (рекомендуемое)	7-12В
Напряжение питания (предельное)	6-20В
Цифровые входы/выходы	14 (из них 6 могут использоваться в качестве ШИМ-выходов)
Аналоговые входы	6
Максимальный ток одного вывода	40 мА
Максимальный выходной ток вывода 3.3V	50 мА
Flash-память	32 КБ (ATmega328) из которых 0.5 КБ используются загрузчиком
SRAM	2 КБ (ATmega328)
EEPROM	1 КБ (ATmega328)
Тактовая частота	16 МГц

Функциональная схема Arduino UnoR3

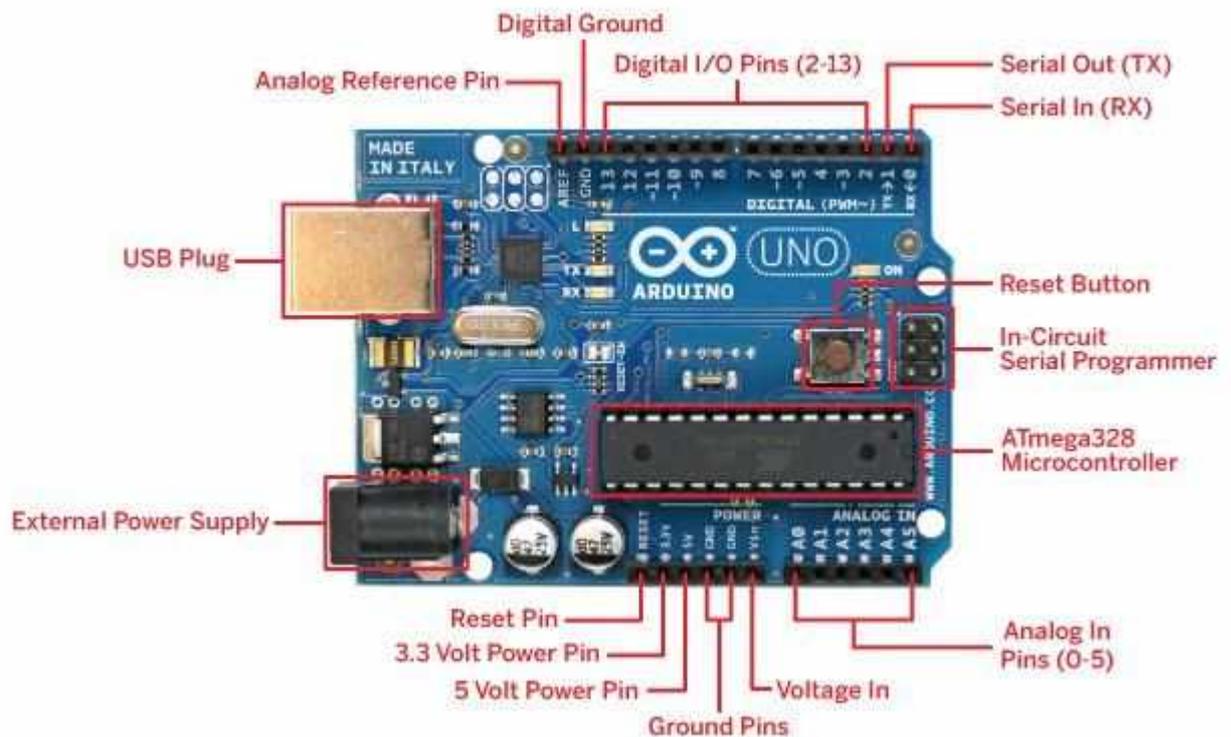


Рисунок 3. Функциональная схема Arduino Uno R3

Пины Ардуино используются для подключения внешних устройств и могут работать как в режиме входа (INPUT), так и в режиме выхода (OUTPUT). К каждому входу может быть подключен встроенный резистор 20-50 кОм с помощью выполнения команды `pinMode ()` в

режиме INPUT_PULLUP. Допустимый ток на каждом из выходов – 20 мА, не более 40 мА в пике.

Для удобства работы некоторые пины совмещают в себе несколько функций:

- Пины 0 и 1 – контакты UART (RX и TX соответственно) .
- Пины с 10 по 13 – контакты SPI (SS, MOSI, MISO и SCK соответственно)
- Пины A4 и A5 – контакты I2C (SDA и SCL соответственно).

Пины с номерами от 0 до 13 являются цифровыми. Это означает, что вы можете считывать и подавать на них только два вида сигналов: HIGH и LOW. С помощью ШИМ также можно использовать цифровые порты для управления мощностью подключенных устройств.

Подключение внешних устройств к плате осуществляется путем присоединения к контактам, расположенным на плате контроллера: одному из цифровых или аналоговых пинов или пинам питания.

Программирование для платы Uno

Для написания программ (скетчей) для контроллер Ардуино необходимо установить среду программирования. Самым простым вариантом будет установка бесплатной Arduino IDE, скачать ее можно с официального сайта.

После установки IDE вам нужно выбрать нужную плату. Для этого у Arduino IDE в меню “Инструменты” и подпункте “Плата” следует выбрать плату (Arduino/Genuino Uno). После выбора платы автоматически изменятся параметры сборки проекта и итоговый скетч будет скомпилирован в формат, который поддерживает плата. Подключив контроллер к компьютеру через USB, вы сможете в одно загрузить на него вашу программу, используя команду “Загрузить”.

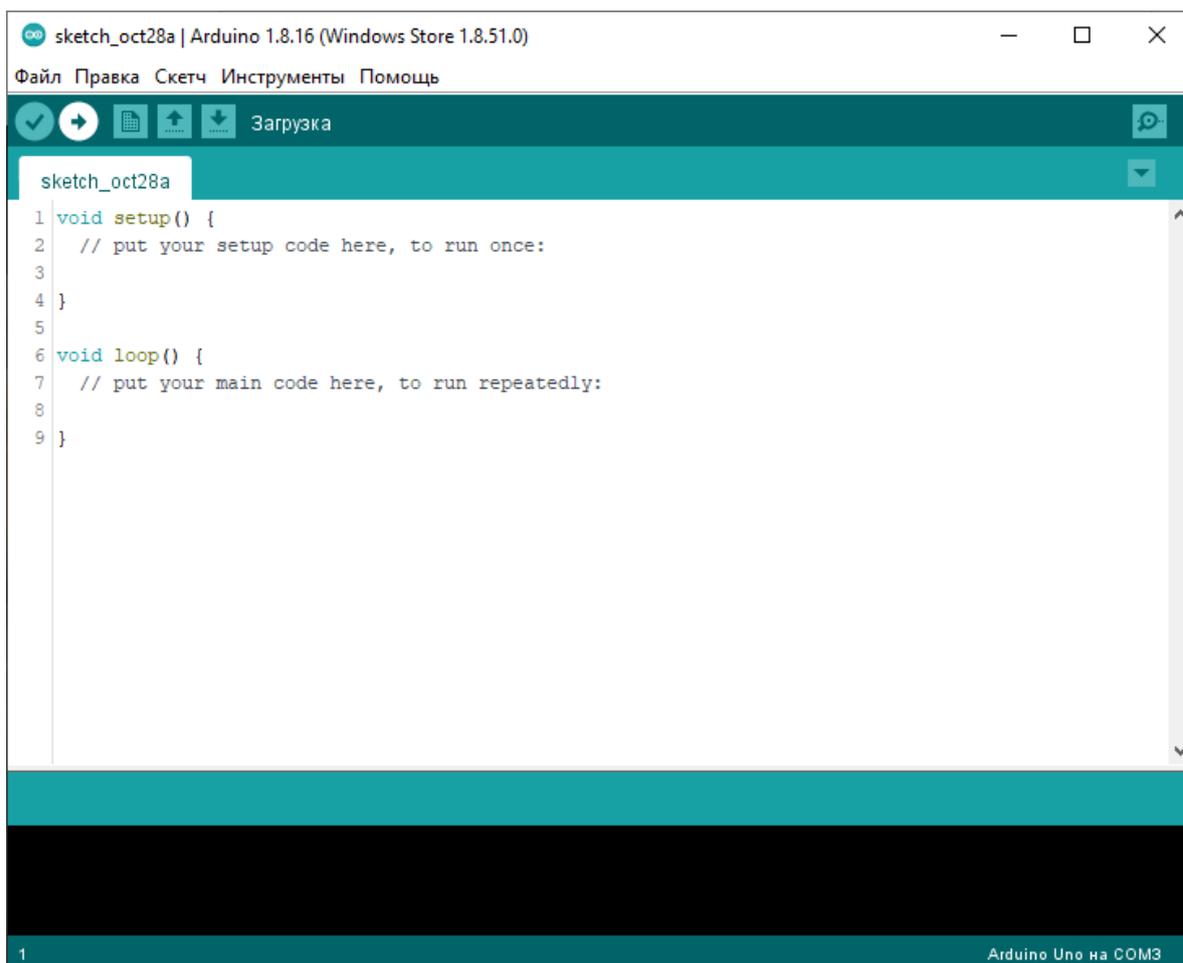


Рисунок 4- Интерфейс Arduino IDE

Сам скетч представляет собой бесконечный цикл, в котором регулярно опрашиваются пины с присоединенными датчиками и устройствами. При программировании микроконтроллера есть возможность подключить готовые библиотеки, как встроенные в IDE, так и доступные на многочисленных сайтах и форумах. (Скетч - Подключить библиотеку)

Написанная и скомпилированная программа загружается через USB-соединение (UART-Serial). Со стороны контроллера за этот процесс отвечает bootloader.

Сервоприводы

Сервопривод – вид привода, который может точно управлять параметрами движения, что позволяет повернуть вал на определенный угол или поддерживать непрерывное вращение с точным периодом.

Схема работы сервопривода основана на использовании обратной связи (контура с замкнутой схемой, в котором сигнал на входе и выходе не согласован). В качестве сервопривода может выступать любой тип механического привода, в составе которого есть датчик и блок управления, который автоматически поддерживает все установленные параметры на датчике. Конструкция сервопривода состоит из двигателя, датчика позиционирования и управляющей системы. Основной задачей таких устройств является реализация в области сервомеханизмов. Также сервоприводы нередко используются в таких сферах как обработка материалов, производство транспортного оборудования, обработка древесины, изготовление металлических листов, производство

стройматериалов и другие.

Принцип работы сервопривода основан на обратной связи с одним или несколькими системными сигналами. Выходной показатель подается на вход, где сравнивается его значение с задающим действием и выполняются необходимые действия – например, выключается двигатель. Самым простым вариантов реализации является переменный резистор, который управляется валом – при изменении параметров резистора меняются параметры питающего двигателя тока.

В более сложных сервоприводах механизм управления сложнее и использует встроенные микросхемы-контроллеры. В зависимости от типа используемого механизма обратной связи выделяют **аналоговые** и **цифровые** сервоприводы. Первые используют что-то, похожее на потенциометр, вторые – контроллеры.



Рисунок 4 - Структура сервопривода

Решающее значение в управлении сервоприводами выполняет управляющий сигнал, который представляет собой импульсы постоянной частоты и переменной ширины. Длина импульса определяет положение сервопривода. Эту длину можно задать в программе вручную методом подбора через угол или использовать команды библиотеки. Для каждой марки устройства длина может быть различной.

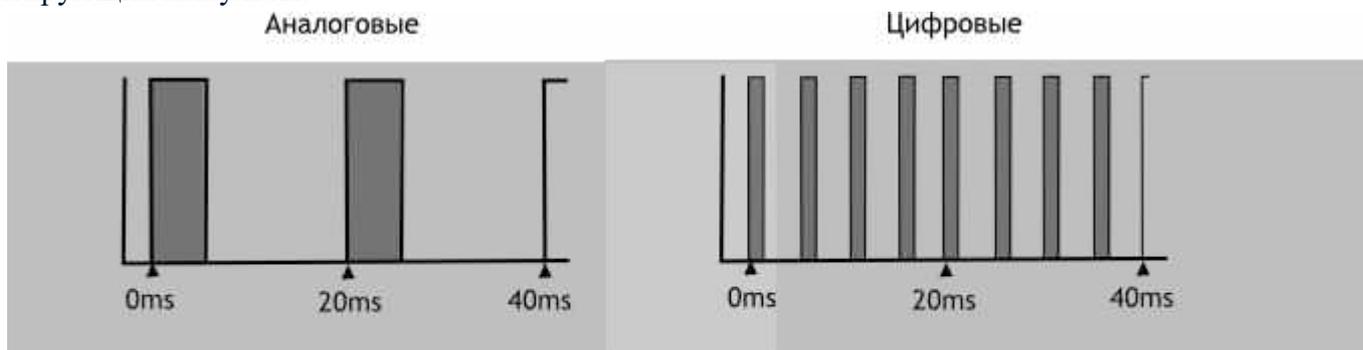
Когда сигнал попадает в управляющую схему, генератор подает свой импульс, длительность которого определяется с помощью потенциометра. В другой части схемы происходит сравнение длительности поданного сигнала и сигнала с генератора. Если эти сигналы разные по длительности, включается электромотор, направление вращения которого определяется тем, какой из импульсов короче. При равенстве длины импульсов мотор останавливается.

Стандартная частота, с которой подаются импульсы, равна 50 Гц, то есть 1 импульс в 20 миллисекунд. При таких значениях длительность составляет 1520 микросекунд, и сервопривод занимает среднее положение. Изменение длины импульса приводит к повороту сервопривода – при увеличении длительности поворот осуществляется по часовой стрелке, при уменьшении – против часовой стрелки. Имеются границы длительности – в Ардуино в библиотеке Servo для 0° установлено значение импульса в 544 мкс (нижняя граница), для 180° – 2400 мкс (верхняя граница).

Важно учитывать, что на конкретном устройстве настройки могут несколько отличаться от общепринятых значений. У некоторых устройств среднее положение и ширина импульса может быть равной 760 мкс. Все принятые значения также могут незначительно отличаться из-за погрешности, которая может быть допущена при производстве устройства.

Способ управления приводом часто по ошибке называют PWM/ШИМ, но это не совсем корректно. Управление напрямую зависит именно от длины импульса, частота их появления не так важна. Корректная работа будет обеспечена как при 40 Гц, так и при 60 Гц, вклад внесет только сильное уменьшение или увеличение частоты. При резком спаде сервопривод начнет работать рывками, при завышении частоты выше 100 Гц устройство может перегреться. Поэтому правильнее называть PDM.

По внутреннему интерфейсу можно выделить аналоговые и цифровые сервоприводы. Внешних отличий нет – все различия только во внутренней электронике. Аналоговый сервопривод внутри содержит специальную микросхему, цифровой – микропроцессор, принимающий и анализирующий импульсы.



При получении сигнала аналоговый сервопривод принимает решение, менять или нет положение, и по необходимости подает на мотор сигнал с частотой 50 Гц. За время реакции (20 мс) могут произойти внешние воздействия, которые изменяют положение сервопривода, и устройство не успеет среагировать. Цифровой сервопривод использует процессор, который подает и обрабатывает сигналы с большей частотой – от 200 Гц, поэтому он может быстрее отреагировать на внешние воздействия, быстрее развивать нужную скорость и крутящий момент. Следовательно, цифровой сервопривод будет лучше удерживать заданное положение. При этом для работы цифрового сервопривода требуется больше электроэнергии, что повышает их стоимость. Большой вклад в цену делает и сложность их производства. Высокая стоимость – единственный недостаток цифровых сервоприводов, в техническом плане они намного лучше аналоговых устройств.

Подключение серводвигателя к ардуино

Сервопривод обладает тремя контактами, которые окрашены в разные цвета. Коричневый провод ведет к земле, красный – к питанию +5В, провод оранжевого или желтого цвета – сигнальный. К Ардуино устройство подключается через макетную указанным на рисунке образом. Оранжевый провод (сигнальный) подключается к цифровому пину, черный и красный – к земле и питанию соответственно. Для управления серводвигателем не требуется подключение именно к шим-пинам – принцип управления серво мы уже описывали ранее.

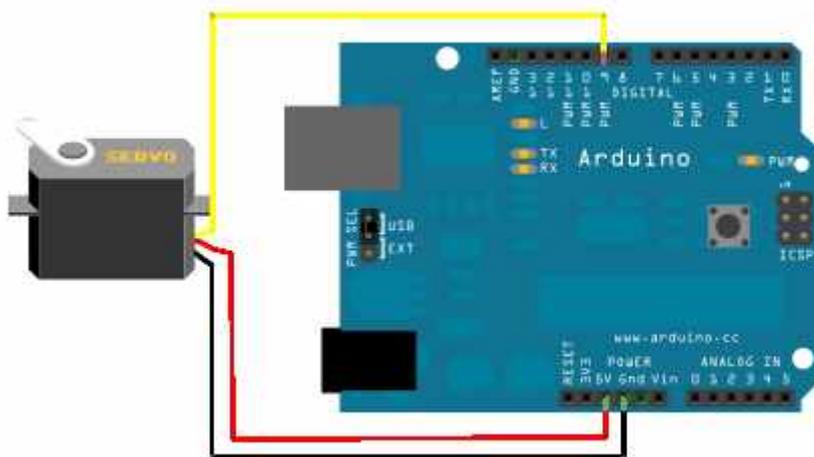


Рисунок -6 Подключение сервопривода к плате ардуино

Не рекомендуется подключать мощные серво напрямую к плате , в виду создания слишком высокого тока, приводящего к перегоранию платы. Симптомы перегрузки и неправильного питания сервопривода заключаются в “дергании” серво, неприятному звуку и перезагрузке платы. Для питания лучше использовать внешние источники, обязательно объединяя земли двух контуров.

Скетч для управления сервоприводом в Arduino

Пример простого скетча для работы с сервоприводом

Пример проекта, в котором мы сразу сначала устанавливаем серводвигатель на нулевой угол, а затем поворачиваем на 90 градусов.

```
#include <Servo.h>
Servo myservo;
void setup(){
myservo.attach(9);
myservo.write(0);
}
void loop(){
myservo.write(0);
delay(1000);
myservo.write(90);
delay(1000);
}
```

Bluetooth

Bluetooth - стандарт передачи данных технология беспроводной передачи данных. Bluetooth обеспечивает обмен информацией между такими устройствами как персональные компьютеры (настольные, карманные, ноутбуки), мобильные телефоны, принтеры, цифровые фотоаппараты, мышки, клавиатуры, джойстики, наушники, гарнитуры на надёжной, недорогой, повсеместно доступной радиочастоте для ближней связи.

Bluetooth позволяет этим устройствам общаться, на расстоянии от 1 до 100 метров друг от друга (дальность сильно зависит от преград и помех), даже в разных помещениях.

Технология Bluetooth предполагает два вида связи: синхронную - SCO (Synchronous Connection Oriented) и асинхронную - ACL (Asynchronous Connectionless). Первый вид, SCO, рассчитан на установление симметричного соединения "точка - точка" и служит преимущественно для передачи речевых сообщений. Скорость передачи информации SCO равна 64 Кбит/с. Второй, ACL, предназначен для пакетной передачи данных. Он поддерживает симметричные и асимметричные соединения типа "точка - много точек". Скорость передачи пакетной информации при ACL составляет порядка 721 Кбит/с. Пакеты данных имеют фиксированный формат. В начале блока находится 72-бит код доступа. Он может применяться, в частности, для синхронизации устройств. За ним следует 54-бит заголовок пакета, содержащий контрольную сумму пакета и информацию о его параметрах (например, о повторной передаче блока данных). Замыкает пакет область, непосредственно содержащая пересылаемую информацию. Размер этой области варьируется от 0 до 2745 бит.

Таблица 1- Распиновка arduino bluetooth модуля

Arduino	Bluetooth
Pin 1(TX)	RXD
Pin 0(RX)	TXD
GND	GND
5V	VCC

Рисунок 3. Подключение Bluetooth модуля к Arduino.

ВАЖНО при подключении модуля к arduino пины подключаются ***TX -> RXD ,RX -> TXD***.

ВАЖНО Во время загрузки скетча необходимо что бы Bluetooth модуль был отключен от микроконтроллера arduino. В противном случае скетч не запишется, потому что связь с Bluetooth модулем происходит по одному и тому же порту RX и TX, что и USB.

Скетч для управления светодиодом

```
int val;  
int LED = 13;  
void setup()  
{  
  Serial.begin(9600);  
  pinMode(LED, OUTPUT);
```

```

digitalWrite(LED, HIGH);
}
void loop()
{
  if (Serial.available())
  {
    val = Serial.read();
    // При символе "1" включаем светодиод
    if (val == '1')
    {
      digitalWrite(LED, HIGH);
    }
    // При символе "0" выключаем светодиод
    if ( val == '0')
    {
      digitalWrite(LED, LOW);
    }

    digitalWrite(LED, LOW);
  }
}
}

```

Во время загрузки скетча необходимо чтобы Bluetooth модуль был отключен от микроконтроллера arduino. В противном случае скетч не запишется, связь с Bluetooth модулем происходит по одному и тому же порту RX и TX, что и USB.

1. Включаем Bluetooth на телефоне и ищем новые устройства
2. Находим в списке расстройств "**HC-06**" и подключаемся к нему.
3. Телефон спросит пин-код. необходимо ввести "**1234**" или "**0000**"

После того как мы установили терминал, запускаем его выбираем наш bluetooth модуль HC-06 и подключаемся к нему.

Пришло время попробовать проект в деле. Пишем в терминале цифру "0" и отправляем. Светодиод L который находится на плате arduino рядом с pin 13, должен погаснуть. Теперь отправим через терминал цифру "1" и светодиод L должен зажечься.

Подключать акселерометр к микроконтроллеру Arduino.

В основе компонента лежит микросхема MPU-6050. В комплект входят 2 предмета – гироскоп и акселерометр. Данные устройства перед конструированием обрабатываются и затем переносятся прямым образом в микроконтроллер через интерфейс.

Акселерометр

Акселерометр — это прибор, позволяющий измерять ускорение тела под действием внешних сил. Подробно об устройстве этого датчика мы уже рассказывали на одном из уроков: Акселерометр: что это такое и как им определять наклон тела

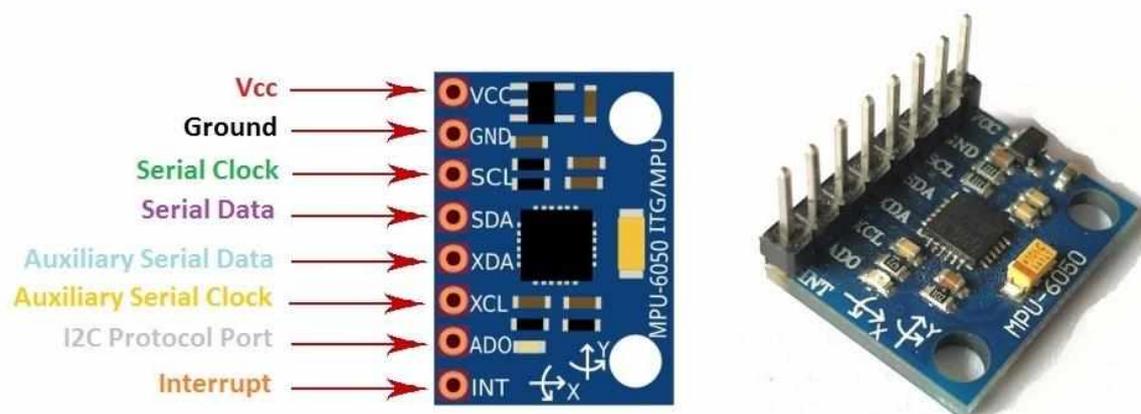


Рисунок 7 - устройство акселерометра Arduino

НА ПЛАТЕ ИМЕЕТСЯ 8 КОНТАКТОВ:

VCC — положительный контакт питания;

GND — земля;

SDA — линия данных I2C;

SCL — линия синхроимпульсов I2C;

INT — настраиваемое прерывание;

AD0 — I2C адрес; по-умолчанию AD0 подтянут к земле, поэтому адрес устройства — 0x68; если соединить AD0 к контактом питания, то адрес изменится на 0x69;

XCL, XDA — дополнительный I2C интерфейс для подключения внешнего магнитометра.

Программа для вычисления угла наклона акселерометра MPU6050

Для работы программы потребуются библиотеки: MPU6050 и I2Cdev.

Литература

1. МакКомб, Г. Робот на Arduino / Г. МакКомб ; перевод с английского Н. Чередниченко. — Москва : ДМК Пресс, 2018. — 52 с. — ISBN 978-5-97060-656-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/107893> (дата обращения: 28.10.2021). — Режим доступа: для авториз. пользователей.
2. Петин, В. В. Практическая энциклопедия Arduino : энциклопедия / В. В. Петин, А. А. Биняковский. — 2-ое изд., испр. и доп. — Москва : ДМК Пресс, 2020. — 166 с. — ISBN 978-5-97060-798-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/131675> (дата обращения: 28.10.2021). — Режим доступа: для авториз. пользователей.
3. Bluetooth модуль HC-06 подключение к Arduino. Управление устройствами с телефона. сайт <https://lesson.iarduino.ru/> [Электронный ресурс]. 2021 режим доступа к журн.: <https://lesson.iarduino.ru/page/bluetooth-modul-hc-06-podklyuchenie-k-arduino-upravlenie-ustroystvami-s-telefona/> (дата обращения: 28.10.2021).
4. Олег Е.А, Ардуино: акселерометр MPU6050 сайт <https://robotclass.ru/> [Электронный ресурс]. 2021 режим доступа к журн.: <https://robotclass.ru/tutorials/arduino-accelerometer-mpu6050/> (дата обращения: 28.10.2021).
5. Бейктал, Д. Конструируем роботов на Arduino. . Первые шаги : руководство / Д. Бейктал ; перевод с английского О. А. Трефиловой. — 3-е изд. — Москва : Лаборатория знаний, 2020. — 323 с. — ISBN 978-5-00101-900-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/151517> (дата обращения: 28.10.2021). — Режим доступа: для авториз. пользователей.
6. Бейктал, Д. Конструируем роботов. Дроны. Руководство для начинающих : руководство / Д. Бейктал ; перевод с английского Ф. Г. Хохлова. — Москва : Лаборатория знаний, 2018. — 226 с. — ISBN 978-5-00101-569-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/103895> (дата обращения: 28.10.2021). — Режим доступа: для авториз. пользователей.
7. Филиппов, С. А. Уроки робототехники. Конструкция. Движение. Управление : учебное пособие / С. А. Филиппов. — 2-е изд. — Москва : Лаборатория знаний, 2018. — 193 с. — ISBN 978-5-00101-595-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/103897> (дата обращения: 28.10.2021). — Режим доступа: для авториз. пользователей.
8. Лесков, А. Г. Кинематика и динамика исполнительных механизмов манипуляционных роботов : учебное пособие / А. Г. Лесков, К. В. Бажинова, Е. В. Селиверстова. — Москва : МГТУ им. Н.Э. Баумана, 2017. — 104 с. — ISBN 978-5-7038-4752-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/103405> (дата обращения: 28.10.2021). — Режим доступа: для авториз. пользователей.