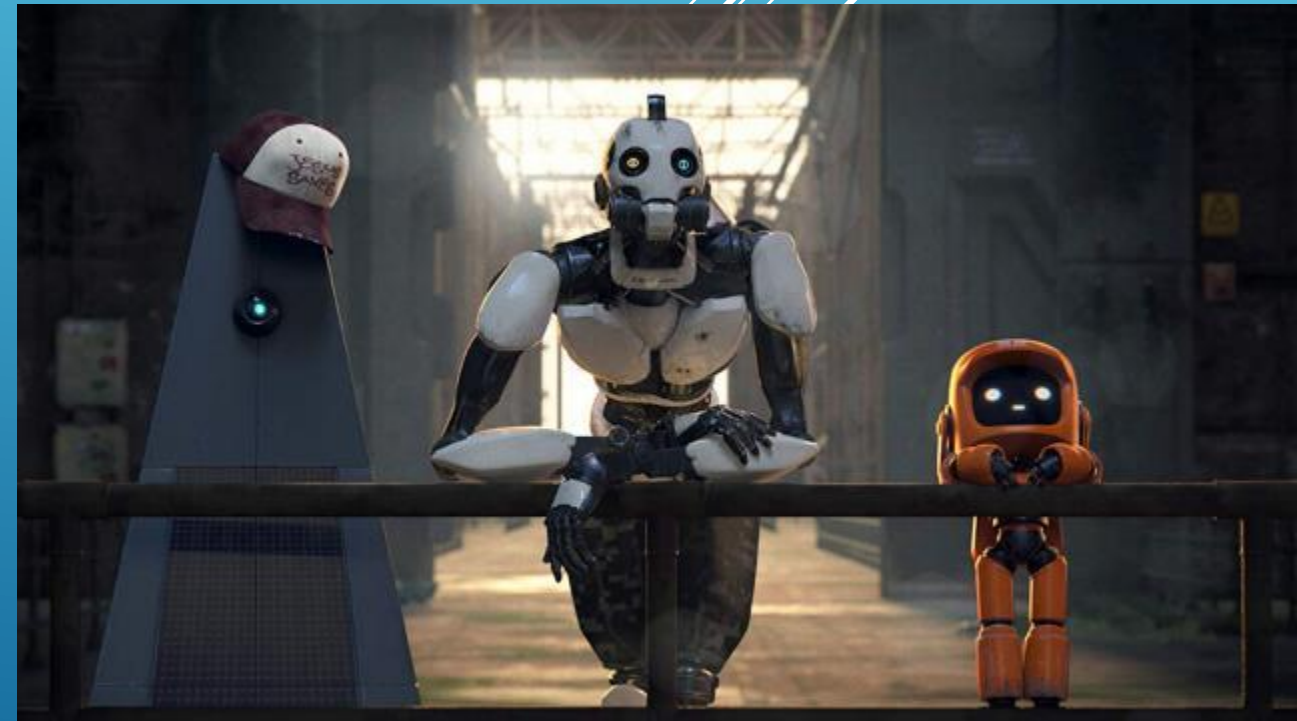


# ОБРАЗОВАТЕЛЬНАЯ РОБОТОТЕХНИКА



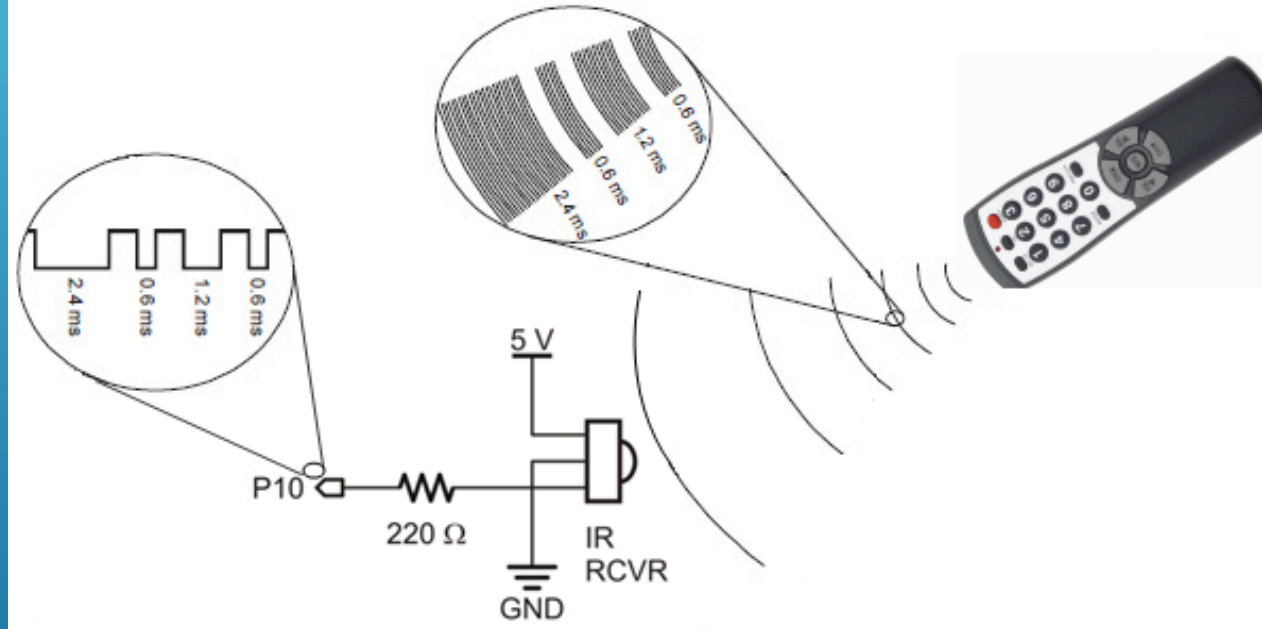
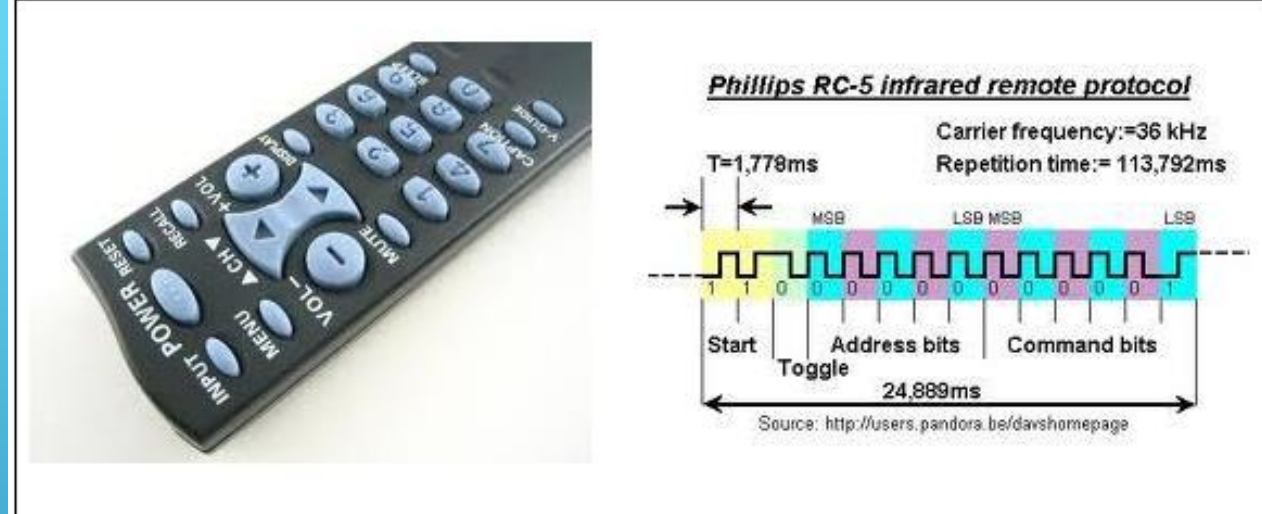
Лекция 4. Беспроводные  
интерфейсы к Arduino:

IRDA, RF, Bluetooth, WiFi, GSM



# ИНТЕРФЕЙС IRDA

- ▶ Управление с помощью инфракрасных импульсов является типичным для многих современных устройств, подходит оно и для управления роботами (в случае их использования в помещении). Широкое применение ИК излучателей стало возможным благодаря их низкой стоимости, простоте и удобству в использовании.
- ▶ ИК-приемник на Ардуино способен принимать и обрабатывать ИК сигнал в виде импульсов заданной длительности и частоты. Так, TSOP 4838 настроен на частоту 38 кГц.
- ▶ Приёмник содержит фотодиод, усилители, полосовой фильтр, демодулятор и преобразователь уровней. Корпус приемника содержит оптический фильтр для защиты от внешних электромагнитных полей и ИК-линзу для фокусировки принимаемого излучения на фотодиоде. Для подключения приемника к Arduino используют три ножки: GND, 5V и любой цифровой вход. Распиновка разная у разных типов датчиков.



# ИК-ИНТЕРФЕЙС

- ▶ Для работы Arduino с инфракрасными датчиками используется библиотека IRremote Arduino Library. Разные версии библиотеки очень сильно отличаются по синтаксису и возможностям.
- ▶ Сравнивая коды, приходящие от пульта через приёмник, с заданными в ветвлениях значениями, можно реализовать логику удалённого управления роботом. FFFFFFFF (или иной специфический код) означает, что кнопка удерживается. В примерах библиотеки показано, как можно подстроиться к конкретному ИК-пульту.

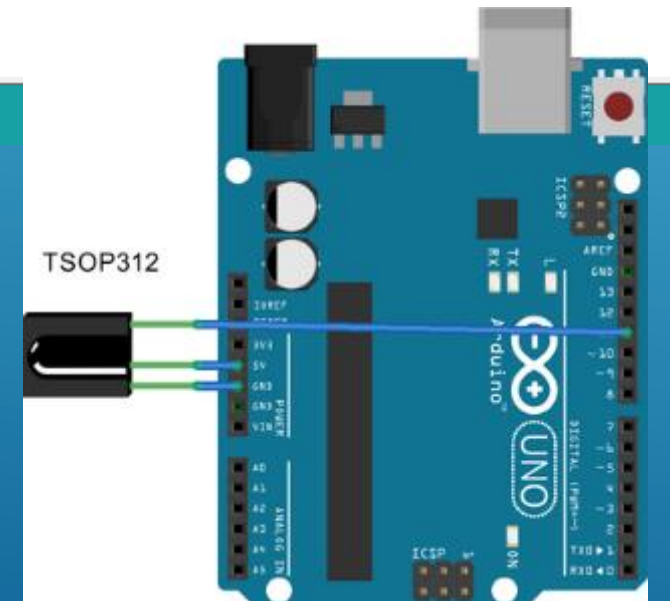
```
if (IrReceiver.decodedIRData.command == 0x10) {  
    // do something  
}  
else if (IrReceiver.decodedIRData.command == 0x11) {  
    // do something else  
}
```

sketch\_nov23a

```
#include <IRremote.h>  
  
const byte IR_PIN = 11;  
  
void setup()  
{  
    Serial.begin(9600);  
    IrReceiver.begin(IR_PIN, ENABLE_LED_FEEDBACK);  
}  
  
void loop()  
{  
    if (IrReceiver.decode())  
    {  
        Serial.println(IrReceiver.decodedIRData.command, HEX);  
        IrReceiver.resume();  
    }  
}
```

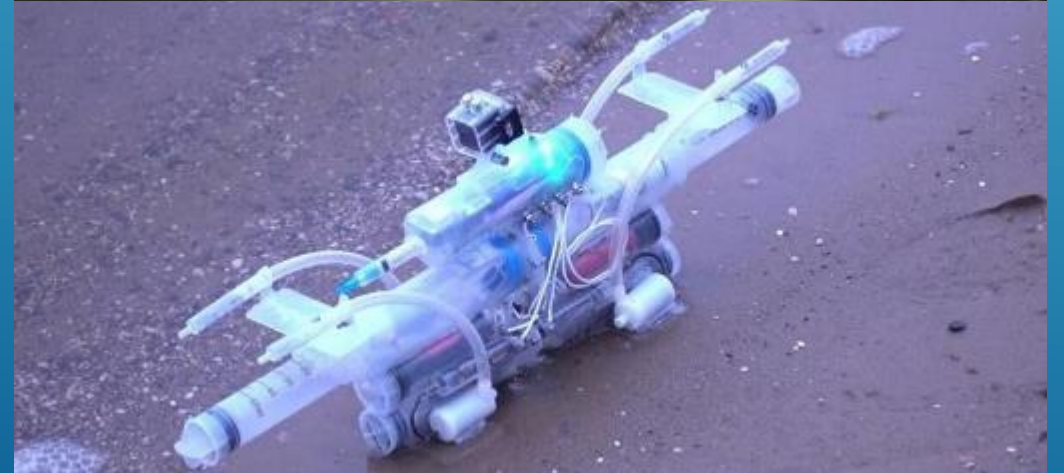
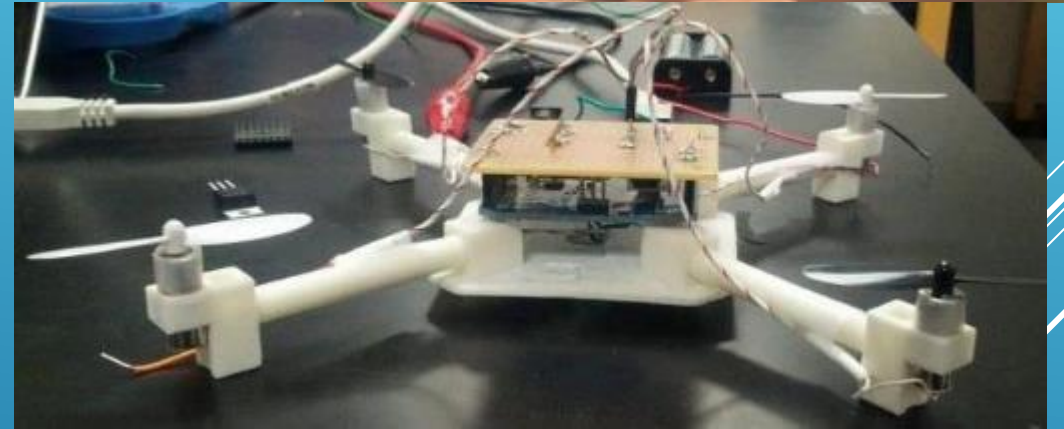
Компиляция завершена

Программа позволяет  
посмотреть в консоли  
коды любого  
совместимого по  
частоте ИК-пульта



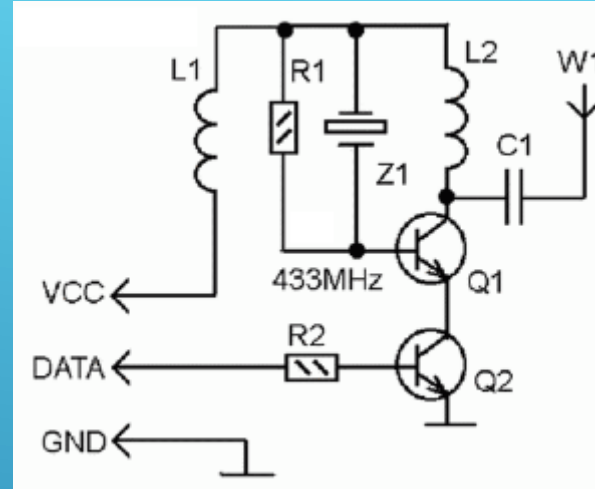
# РАДИОУПРАВЛЕНИЕ ДЛЯ ARDUINO

- ▶ Радиомодули весьма часто используются в робототехнике для управления роботизированными устройствами. Отличие различных модулей в основном состоит в используемых частотах приёма-передачи (количестве частотных каналов), в мощности сигнала (дальности управления – возможны варианты до нескольких километров).
- ▶ Обычно радиомодули работают в частотном диапазоне 2.4 ГГц (WiFi, BlueTooth, радиостанции, и др. устройства) либо 433.920 МГц (авто-сигнализации, шлагбаумы, игрушки). Это может приводить к глушению некоторых каналов в выбранном диапазоне.
- ▶ Для управления движущимися моделями требуется выполнение одновременно нескольких функций. Поэтому передатчики радиоуправления делают многоканальными. Так, для авто- и судомоделей требуются два канала: управление направлением движения и оборотами двигателя. Для полноценного управления самолетом нужно не менее четырех, а коптером — пяти каналов.



# РАДИОМОДУЛЬ FS-1000A

- ▶ Напряжение питания передатчика: 3...12 В
- ▶ Напряжение питания приемника: 5 В
- ▶ Несущая частота: 433 МГц
- ▶ Потребляемый ток передатчиков: 8 мА
- ▶ Потребляемый ток приемником: 4,5 мА
- ▶ Чувствительность приемника: -106...-110 дБм
- ▶ Выходная мощность передатчика: 32 мВт
- ▶ Макс пропускная способность передатчика: 8 кб/сек
- ▶ Макс пропускная способность приемника: 5 кб/сек
- ▶ Диапазон рабочих температур: -20...+80 °С
- ▶ Низкая скорость, однонаправленный характер передачи и примитивный способ передачи информации
- ▶ Для увеличения дальности приёма (до 50 м) нужна внешняя антенна



Если на вход DATA подается логическая 1 - открывается транзистор Q2 и генератор формирует несущую волну на частоте 433 МГц. Когда на вход DATA устанавливается логический 0, генератор останавливается.

Для корректного устранения шума приёмника при написании скетча лучше пользоваться библиотекой **RCSwitch**.

Передатчик



Приёмник



# РАДИОМОДУЛЬ NRF24L01

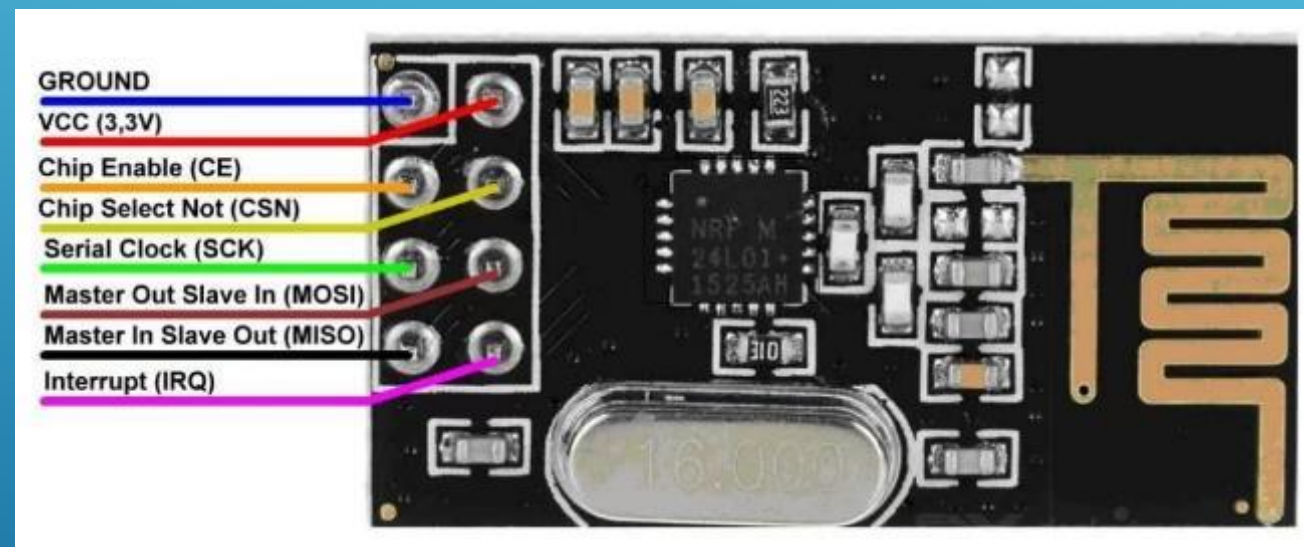
Характеристики модуля:

- ▶ Напряжение питания: 1,9В – 3,6В;
- ▶ Интерфейс обмена данными: SPI;
- ▶ Частота приёма и передачи: 2,400...2.525 ГГц;
- ▶ Количество каналов: 128 с шагом 1МГц;
- ▶ Тип модуляции: GFSK (ЧМ);
- ▶ Скорость передачи данных: 250kbps, 1Mbps и 2Mbps;
- ▶ Чувствительность приёмника: -82 dBm;
- ▶ Расстояние приёма/передачи данных: 100м – прямая видимость; 30м – помещение.

Большинство проектов начального уровня предусматривают использование двух модулей NRF24L01, один из которых работает в режиме передатчика, а другой как приёмник на одинаковой частоте. Однако на одной частоте могут работать до 6 передатчиков и 1 приёмник.

Достоинства: малое энергопотребление, значительный радиус действия, высокая скорость передачи, низкая цена.

Недостатки: быстрое затухание в плотных средах.



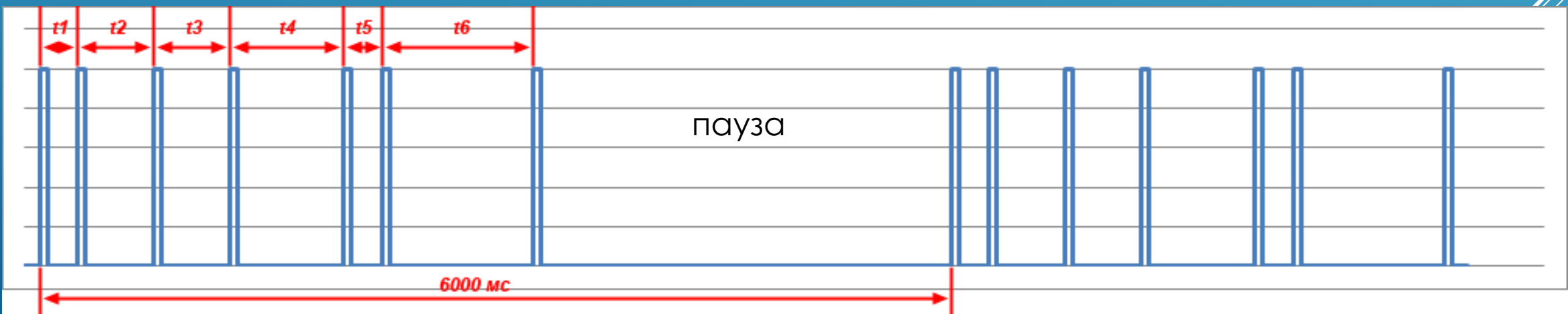
Радиомодуль NRF24L01	GND	VCC	CE	CSN	MOSI	MISO	SCK
Arduino Uno	GND	+3,3 В	9	10	11	12	13

# PPM-КОДИРОВАНИЕ

- ▶ PPM (Pulse-position modulation) - распространенный метод кодирования сигналов, передаваемых дистанционно. Большое распространение данный метод получил в системах радио-управления моделей самолетов и лодок. PPM используется и в обычных пультах дистанционного управления с инфракрасным передатчиком, а также в некоторых других системах связи, где отсутствуют требования серьезной помехоустойчивости.
- ▶ Продолжительность PPM-фрейма для передачи ШИМ-сигналов (на сервоприводы) составляет 20 мс. Один такой фрейм может включать в себя до восьми каналов управления. Это означает, что информация о положениях ручек управления на передатчике попадает на модель 50 раз в секунду, что определяет быстродействие аппаратуры управления.

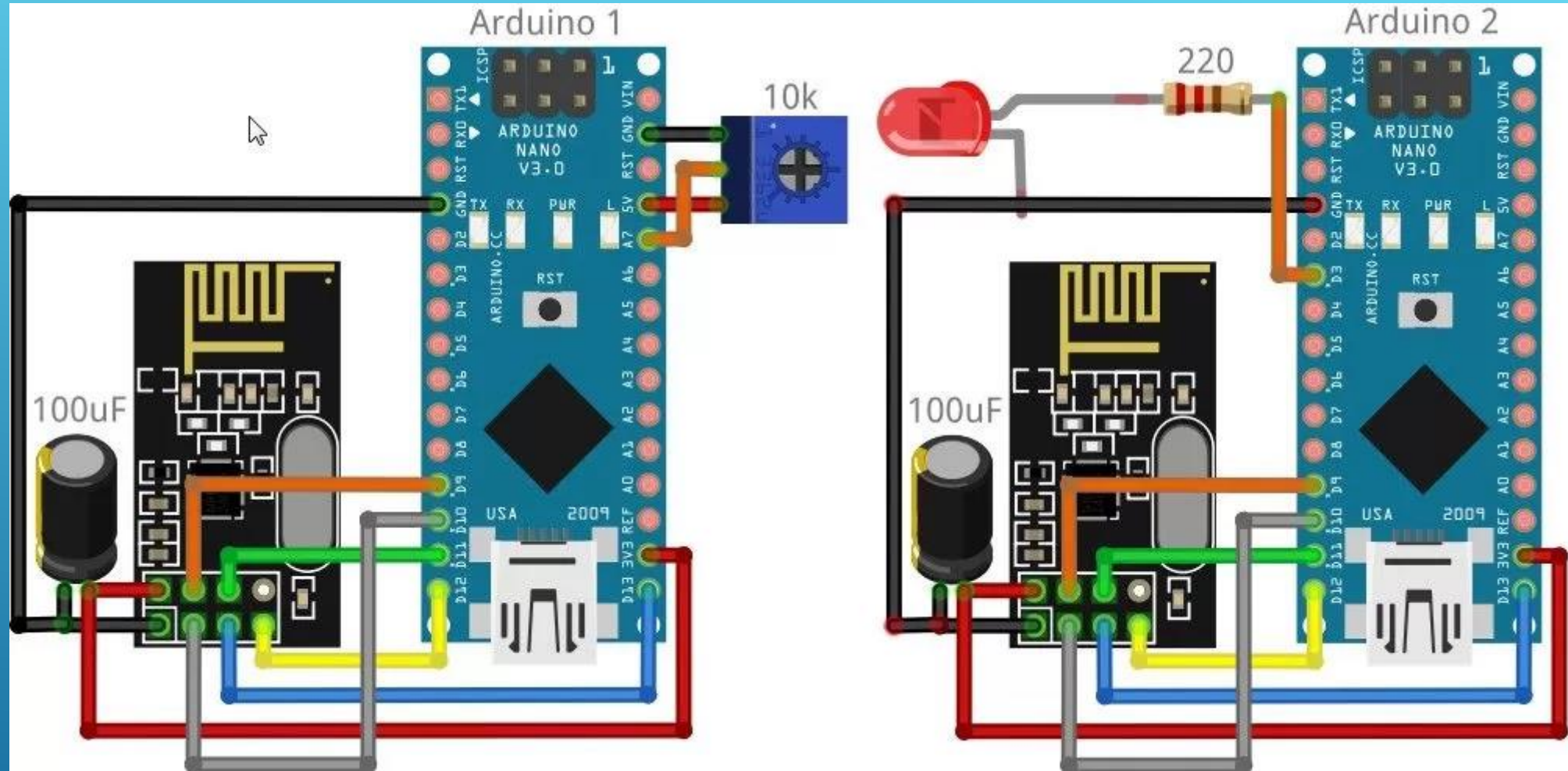
Передано 6 целых чисел ("каналы") в диапазоне от 0 до 1000: 250, 500, 500, 750, 250, 1000.

Каждое число соответствует промежутку между импульсами плюс ширина самого импульса.



условный фрейм

# СОЕДИНЕНИЕ ПЛАТ ARDUINO ПО РАДИОКАНАЛУ



К одной плате Arduino подключен потенциометр, к другой – светодиод. Задача: путём вращения ручки потенциометра регулировать яркость светодиода по радиоканалу с использованием модуля NRF24L01.



# СКЕТЧ ДЛЯ ПЕРВОЙ ПЛАТЫ ARDUINO

```
#include <SPI.h> // Подключаем библиотеку для работы с SPI-интерфейсом
#include <nRF24L01.h> // Подключаем файл конфигурации из библиотеки RF24
#include <RF24.h> // Подключаем библиотеку для работа для работы с модулем NRF24L01
#define PIN_POT A7 // Номер пина Arduino, к которому подключен потенциометр
#define PIN_CE 10 // Номер пина Arduino, к которому подключен вывод CE радиомодуля
#define PIN_CSN 9 // Номер пина Arduino, к которому подключен вывод CSN радиомодуля
RF24 radio(PIN_CE, PIN_CSN); // Создаём объект radio с указанием выводов CE и CSN

int potValue[1]; // Создаём массив для передачи значений потенциометра

void setup() {
  radio.begin(); // Инициализация модуля NRF24L01
  radio.setChannel(5); // Обмен данными будет вестись на пятом канале (2,405 ГГц)
  radio.setDataRate (RF24_1MBPS); // Скорость обмена данными 1 Мбит/сек
  radio.setPALevel(RF24_PA_HIGH); // Выбираем высокую мощность передатчика (-6dBm)
  radio.openWritingPipe(0x7878787878LL); // Открываем трубу с личным ID
}

void loop() {
  potValue[0] = analogRead(PIN_POT); // считываем показания потенциометра
  radio.write(potValue, 1); // отправляем считанные показания по радиоканалу
}
```

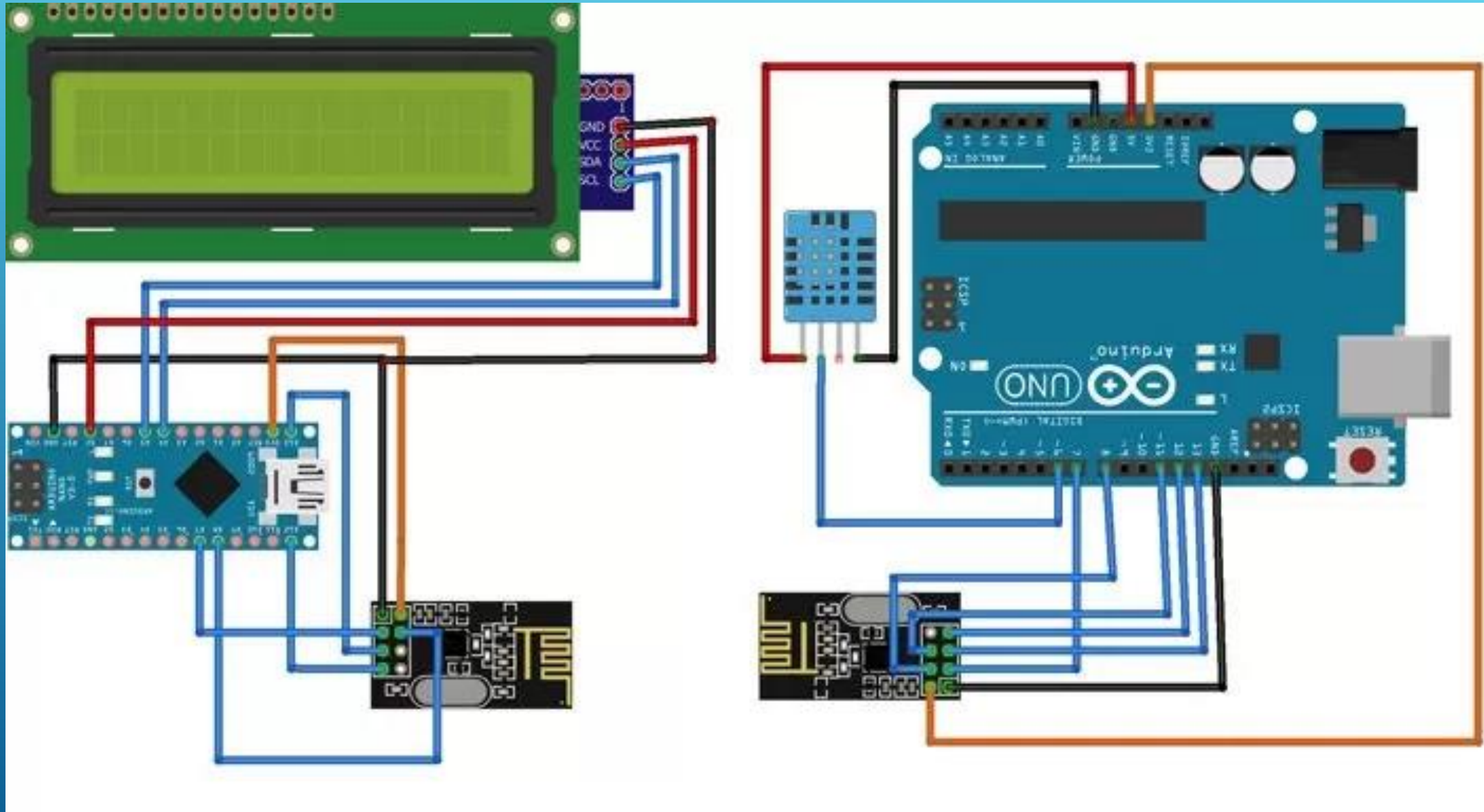
# СКЕТЧ ДЛЯ ВТОРОЙ ПЛАТЫ ARDUINO

```
#include <SPI.h> // Подключаем библиотеку для работы с SPI-интерфейсом
#include <nRF24L01.h> // Подключаем файл конфигурации из библиотеки RF24
#include <RF24.h> // Подключаем библиотеку для работы с модулем NRF24L01
#define PIN_LED 3 // Номер пина Arduino, к которому подключен светодиод
#define PIN_CE 10 // Номер пина Arduino, к которому подключен вывод CE радиомодуля
#define PIN_CSN 9 // Номер пина Arduino, к которому подключен вывод CSN радиомодуля
RF24 radio(PIN_CE, PIN_CSN); // Создаём объект radio с указанием выводов CE и CSN
int potValue[1]; // Создаём массив для приёма значений потенциометра

void setup() {
  pinMode(PIN_LED, OUTPUT); // Настраиваем на выход пин светодиода
  radio.begin(); // инициализация модуля NRF24L01
  radio.setChannel(5); // Обмен данными будет вестись на пятом канале (2,405 ГГц)
  radio.setDataRate (RF24_1MBPS); // Скорость обмена данными 1 Мбит/сек
  radio.setPALevel(RF24_PA_HIGH); // Выбираем высокую мощность передатчика (-6dBm)
  radio.openReadingPipe (1, 0x7878787878LL); // Открываем трубу с известным ID
  radio.startListening(); // начинаем прослушивать трубу
}

void loop() {
  if(radio.available()){ // Если в буфер приёмника поступили данные
    radio.read(&potValue, sizeof(potValue)); // читаем показания потенциометра
    analogWrite(PIN_LED, map(potValue[0],0,1023,0,255)); // Регулируем яркость диода
  }
}
```

# УДАЛЁННЫЙ КОНТРОЛЬ ТЕМПЕРАТУРЫ И ВЛАЖНОСТИ



# МОДУЛЬ BLUETOOTH ДЛЯ ARDUINO

- ▶ Беспроводной интерфейс Bluetooth является одним из самых популярных интерфейсов, используемых для связи создаваемого устройства на базе Arduino с мобильным приложением. Связь, как правило, осуществляется с помощью Bluetooth-модулей HC-05 или HC-06 (на основе чипа BC417) — недорогих и широко распространенных.
- ▶ Bluetooth-модуль общается с платой Arduino по последовательному порту (UART) и работает в двух режимах:
  - ▶ отправка/получение данных, поступающих на него по последовательному порту;
  - ▶ программирование модуля путём отправки AT-команд.
- ▶ Некоторое неудобство использования Arduino Uno заключается в том, что у него один UART-интерфейс – поэтому его можно использовать либо для работы с компьютером через USB (загрузка скетчей), либо для работы с Bluetooth-модулем: подключив последний, мы автоматически отключаем USB-контроллер. Для Arduino Mega можно использовать любой из трёх незадействованных портов UART.

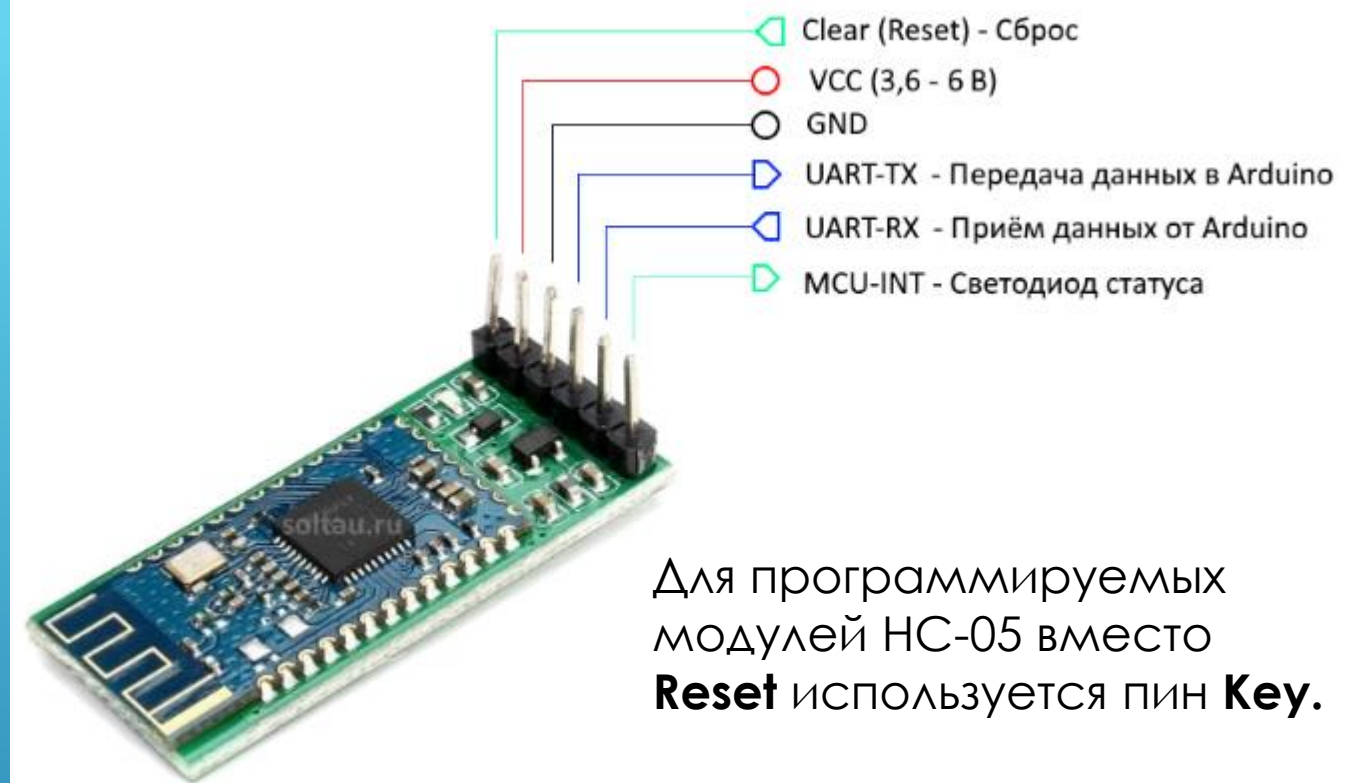


Технология Bluetooth обеспечивает хорошую устойчивость к широкополосным помехам, что позволяет множеству устройств, находящихся в одном месте, одновременно общаться между собой, не мешая друг другу.

# МОДУЛЬ BLUETOOTH HC-06

Основные характеристики модуля:

- ▶ Питание 3,3В – 6 В;
- ▶ Максимальное входное напряжение 5 В;
- ▶ Максимальный ток 45 мА;
- ▶ Скорость передачи данных 1200–1382400 бод;
- ▶ Рабочие частоты 2,40 ГГц – 2,48ГГц;
- ▶ Поддержка спецификации bluetooth версии 2.1;
- ▶ Малое потребление энергии;
- ▶ Высокий уровень защиты данных;
- ▶ Дальность связи 30 м;
- ▶ Для подключения к смартфону по умолчанию используются следующие данные – пароль 1234 или 0000, скорость передачи данных 9600, имя модуля HC-06.



Для программируемых модулей HC-05 вместо **Reset** используется пин **Key**.

Модуль HC-06 используется только в режиме *slave*, то есть он не может самостоятельно подключаться к другим устройствам Bluetooth. Все настройки для подключения (пароль, скорость передачи данных) можно изменить при помощи AT-команд. Модуль HC 05 может работать и в качестве ведущего (*master*), и в качестве ведомого (*slave*).

# САМЫЙ ПРОСТОЙ СКЕТЧ ДЛЯ BLUETOOTH

Во время записи скетча Arduino Uno должно быть отключено от модуля Bluetooth. Во время работы Arduino Uno должно быть отключено от компьютера и подключено к внешнему источнику питания.



На смартфоне нужно установить приложение, подобное Serial Bluetooth Terminal или Arduino Bluetooth Control

```
20:34:11.030 Connecting to HC05 ...
20:34:12.130 Connected
20:34:19.558 s 3 aaa
20:34:25.078 s 4 bbb
20:34:32.068 g 3
20:34:32.559 aaa
20:34:43.104 g 4
20:34:43.666 bbb
20:34:45.167 Disconnected
```

```
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);
    }
  }
}
```

# ПРОГРАММИРОВАНИЕ МОДУЛЯ BLUETOOTH

Настройка модуля осуществляется в режиме программирования отправкой AT-команд по последовательному порту. AT-команда - это текстовая строка, начинающаяся с букв «AT» (от английского attention). Модуль выполняет поступившую команду и отправляет обратно ответ (результат выполнения команды), который также является строкой. На контакт KEY модуля (или на вывод 34 внутренней платы) подаётся 3.3В. AT-команды можно отправлять из монитора последовательного порта Arduino IDE, а для подключения модуля придётся использовать программный UART. Скорость UART в режиме программирования 38400 бод.

команда	ответ	описание
<b>AT</b> \r\n	<b>OK</b> \r\n	<b>Команда Тест:</b> Используется для проверки связи с модулем.
<b>AT+RESET</b> \r\n	<b>OK</b> \r\n	<b>Команда программной перезагрузки модуля:</b> Модуль ведёт себя так, как после кратковременного отключения питания.
<b>AT+NAME?</b> \r\n <b>AT+NAME=ИМЯ</b> \r\n	+NAME: <b>ИМЯ</b> \r\n <b>OK</b> \r\n	<b>Запрос / установка имени модуля:</b> Имя модуля представлено строкой до 32 байт.
<b>AT+PSWD?</b> \r\n <b>AT+PSWD=КОД</b> \r\n	+PSWD: <b>КОД</b> \r\n <b>OK</b> \r\n	<b>Запрос / Установка PIN-кода:</b> Код доступа представлен строкой до 16 байт. Код модуля в роли ведомого устройства является паролем доступа к текущему модулю. Код модуля в роли ведущего устройства является паролем доступа к внешним Bluetooth устройствам.

# МОДУЛЬ WI-FI ДЛЯ ARDUINO

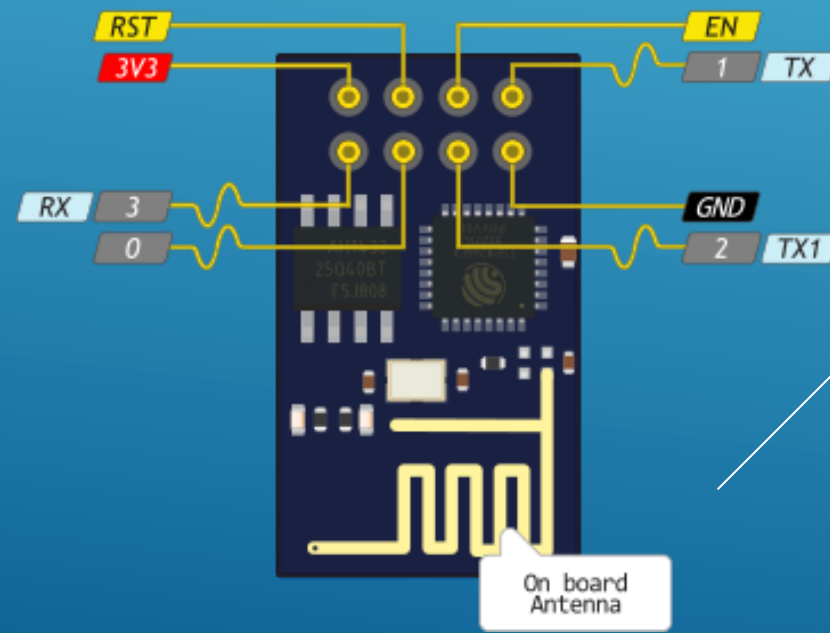
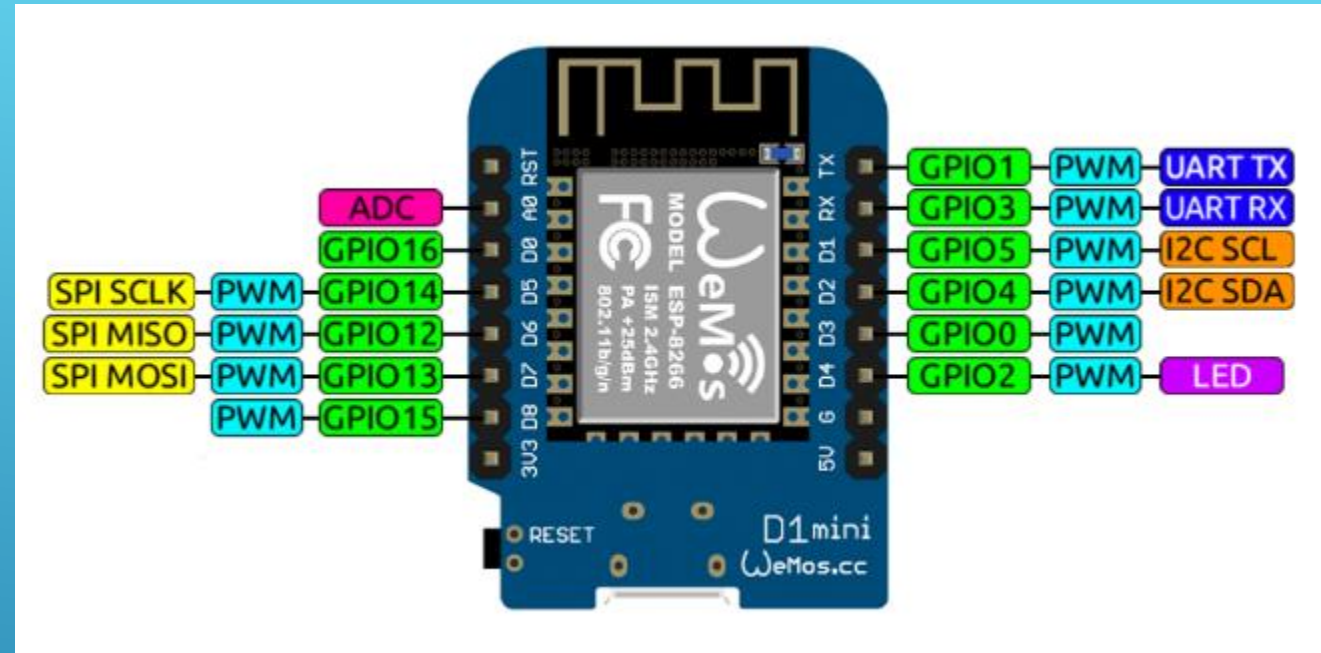
- ▶ Модуль Wi-Fi (Wireless Fidelity, технология беспроводной локальной сети) – это отдельное устройство, обычно на базе микроконтроллера **ESP 8266** (модули обычно обозначают как ESP-01, ESP-02 и т.д.), которое может отдельно программироваться и самостоятельно работать с различными периферийными устройствами. При этом современные версии модулей и прошивок к ним позволяют работать с актуальными стандартами беспроводных сетей **802.11 b/g/n** 2.4 ГГц с протоколами защиты WEP, WPA и WPA2.
- ▶ Питание модуля – **3.3 В** с током до **250-300 мА**. Такой ток плата Arduino выдать не может, поэтому требуется отдельное питание для Wi-Fi модуля. Модуль содержит собственный банк ППЗУ объёмом от 2 до 16 Мбайт.
- ▶ Несмотря на большое число контактов модуля и возможность использования различных вариантов подключения (I<sup>2</sup>C, SPI, контакты для передачи аналогового сигнала), обычно для соединения с Arduino используется последовательный интерфейс **UART**. Для Arduino Uno использование аппаратного интерфейса UART, как обычно, требует отключения платы от компьютера при работе с модулем. При программировании модуля его можно соединить непосредственно с компьютером через переходник USB-UART.





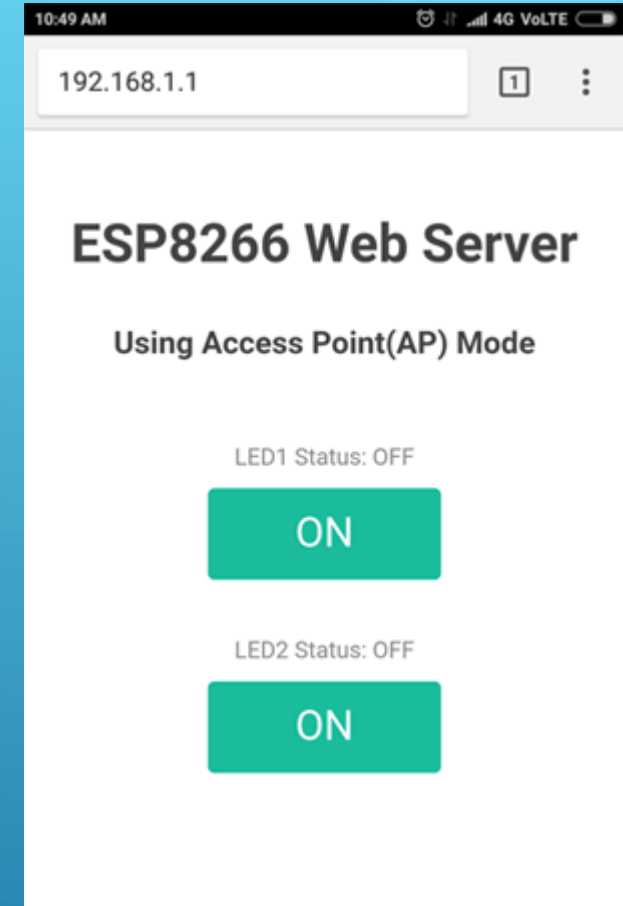
# ПРОГРАММИРОВАНИЕ МОДУЛЯ WI-FI

- ▶ Работа с модулем, в том числе и программирование модуля, осуществляется с использованием UART через AT-команды, в числе которых:
  - ▶ **AT** – проверка состояния модуля
  - ▶ **AT+GMR** – данные о версии модуля и ПО
  - ▶ **AT+RST** - рестарт модуля
  - ▶ **AT+CWJAP="name","pass"** – подключение к WiFi-точке
  - ▶ **AT+CWLAP** - список доступных WiFi-сетей
  - ▶ **AT+CIFSR** - получить локальный IP-адрес
- ▶ Возможен вариант низкоуровневой подготовки сетевых пакетов с помощью соответствующего набора AT-команд. Для работы с UDP-протоколом разумно использовать библиотеку **WiFi.h**.
- ▶ Многие из модулей имеют собственный USB-интерфейс, а программа IDE Arduino имеет возможность подключения напрямую к таким модулям для интерактивного управления параметрами и программирования модуля.



# ARDUINO КАК WEB-СЕРВЕР - ИДЕИ

- ▶ Запрос всегда формирует браузер удалённого клиента, отвечает – сервер. Правильным решением для Wi-Fi модуля является реализация web-сервера непосредственно внутри модуля ESP. В таком случае плата Arduino будет служить вспомогательным контроллером для мониторинга датчиков и передачи показаний удалённым клиентам через Wi-Fi модуль.
- ▶ Обмен данными происходит по протоколу HTTP через формирование TCP/IP-пакетов.
- ▶ В основном цикле loop () проверяются переменные состояния, чтобы отслеживать, что было выполнено и что еще должно произойти для успешной связи с клиентом. Модуль постоянно будет ожидать клиентских подключений к серверу. В ответ на запрос клиента ответ содержит HTTP- заголовок ответа и тип форматирования веб-страницы, выдаваемой по запросу. Заголовок "404" указывает браузеру, что сервер не может найти запрашиваемую страницу. Ответ "200 Response" указывает, что запрос был получен и что HTML-данные будут передаваться в браузер.



# GSM-МОДУЛЬ ДЛЯ ARDUINO

- ▶ **GSM** (Global System for Mobile Communications) и **GPRS** (General Packet Radio Service) - модуль для Arduino - позволяет подключаться к удаленным автономным устройствам через сети сотовой связи. С помощью этого модуля можно отправлять команды на устройства и принимать информацию на телефон удалённого клиента с помощью SMS-команд или через интернет-подключение, открытое по протоколу GPRS.
- ▶ Модуль **A6** разработан фирмой AI-THINKER в 2016 году. Устройство используется для обмена смс-сообщениями и обмена данными по GPRS. Плата отличается низким потреблением энергии и малыми размерами. Устройство полностью совместимо с российскими мобильными операторами. Сим-карта стандарта "микро". Модуль содержит внешнюю антенну и разъём для подключения гарнитуры (а также отдельные контакты внешних микрофонов и динамика для осуществления полноценных голосовых звонков). Для питания модуля необходим отдельный источник **+5 В** с пиковым током до **2 А** (рабочий – 0.9 А).
- ▶ Для связи с Arduino используется последовательный UART интерфейс. Скорость по умолчанию 115200 бит/с.

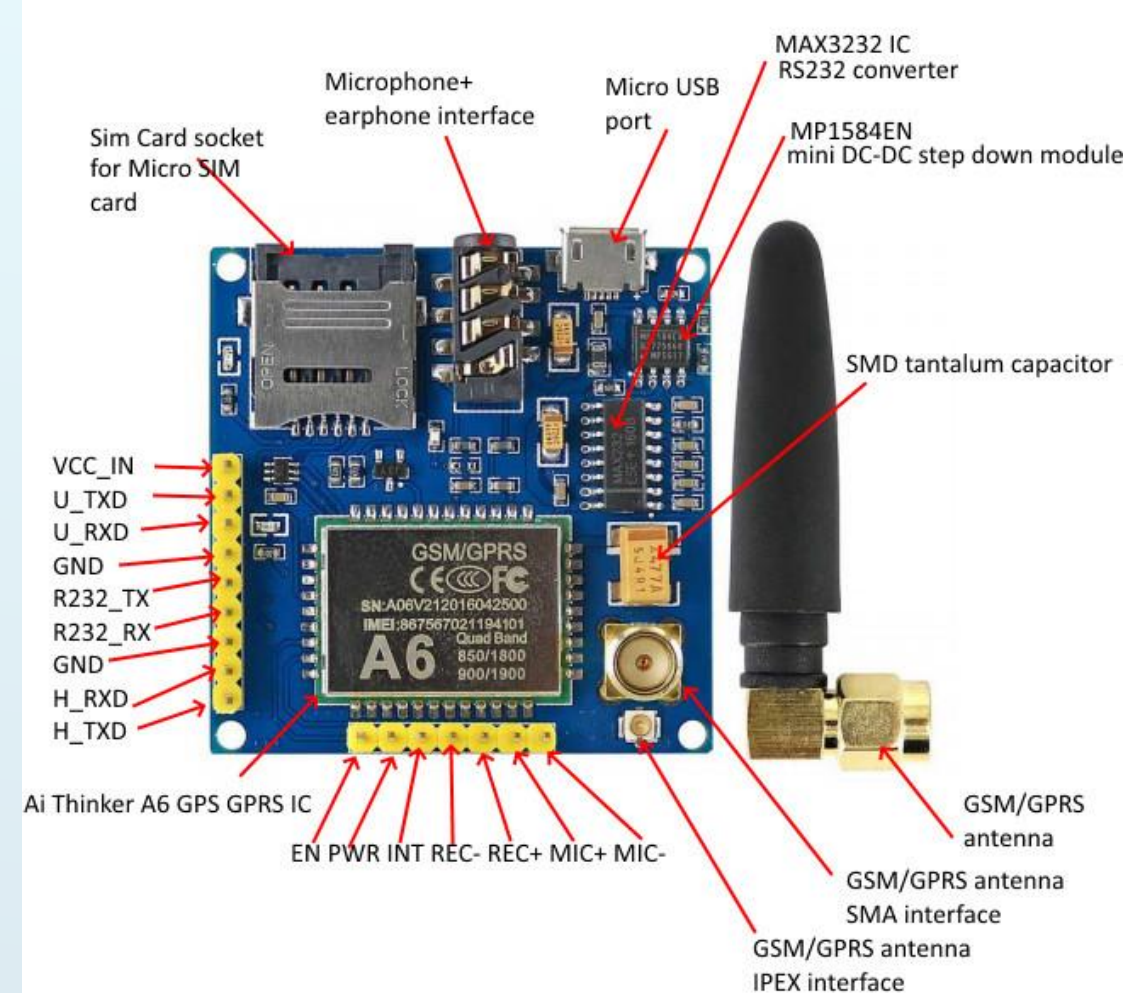


# ПРОГРАММИРОВАНИЕ GSM-МОДУЛЯ

▶ Взаимодействие с модулем осуществляется по последовательному порту посредством AT-команд. Как и в предыдущих устройствах, модуль можно непосредственно запрограммировать через UART компьютера, либо через переходник USB-UART. Для работы подойдёт любая программа-монитор последовательного порта, в том числе IDE Arduino.

▶ Если используется Arduino Mega и задействовано несколько последовательных портов, то можно организовать "туннель" между компьютером и A6 следующим способом:

```
▶ void A6_tunnel() {  
▶ do {  
▶     if (A6_serial.available())  
▶         serial.write( A6_serial.read());  
▶     if (serial.available())  
▶         A6_serial.write( serial.read());  
▶ }  
▶ while (true);  
▶ }
```





# ПРИМЕНЕНИЕ GSM-МОДУЛЯ

- ▶ Типичный вариант применения - запуск исполнительных устройств по SMS (Short Message Service) либо по телефонному звонку. К исполнительным устройствам робототехники гражданского назначения можно отнести шлагбаумы, гаражные либо въездные ворота, систему отопления умного дома и др.
- ▶ Обратный вариант использования – функции охраны, в т.ч. сообщение от датчиков умного дома, связанных с платой Arduino, в виде SMS-сообщений хозяевам дома о возникшей ситуации. Удобством такой системы является моментальное информирование о проблемах в зоне охраны (конечно, при условии, что устройство и клиент находятся в зоне покрытия сотовой связи).



# RFID-СИСТЕМЫ ДЛЯ ARDUINO

Радиочастотная идентификация (RFID) — это технология бесконтактной идентификации объектов при помощи радиочастотного канала связи.

Идентификация объектов производится по уникальному идентификатору, который имеет каждая электронная метка.



- ▶ Считыватель излучает электромагнитные волны определенной частоты. Метки отправляют в ответ информацию – идентификационный номер, данные памяти и пр. Типичные диапазоны частот: LF (125 - 134 кГц), HF (13.56 МГц), UHF (860 - 960 МГц).
- ▶ В наборе Arduino Start RFID-модуль выполнен на микросхеме MFRC522 фирмы NXP, которая обеспечивает работу с метками HF (на частоте **13,56 МГц**, криптографическая защита, протокол Mifare). В комплекте с модулем RFID-RC522 идут две метки: одна - в виде карты, другая - в виде брелока. На метки можно (во время считывания идентификатора) записывать различную информацию (например, фамилию и имя владельца).
- ▶ Напряжение питания модуля – **3.3 В**, максимальная дальность считывания – 60 мм, интерфейс подключения – **SPI** (в зависимости от варианта исполнения возможны интерфейсы UART и I<sup>2</sup>C).
- ▶ Для работы с модулем удобно использовать библиотеку **MFRC522**, в которой реализованы все низкоуровневые операции обмена информацией с картой.

# ПРОГРАММИРОВАНИЕ RFID

Функции библиотеки MFRC522:

- ▶ **PCD\_DumpVersionToSerial()** - выводит версию прошивки ридера в монитор порта
- ▶ **PICC\_IsNewCardPresent()** - проверяет наличие поднесённой метки
- ▶ **PICC\_ReadCardSerial()** - считывает данные с метки
- ▶ **PICC\_DumpToSerial()** - выводит данные метки в монитор порта
- ▶ **PICC\_GetType()** – выводит тип карты
- ▶ **PICC\_GetTypeName()** – выводит имя карты



Сигнал сброса RST – это сигнал, поступающий от цифрового выхода контроллера. При поступлении сигнала низкого уровня происходит перезагрузка считывателя. Также ридер установкой на RST низкого уровня сообщает, что находится в режиме сна, для вывода модуля из режима сна необходимо подать на данный вывод сигнал высокого уровня.



# ПРОГРАММИРОВАНИЕ RFID

```
#include <SPI.h>
#include <MFRC522.h>

// контакты
#define SS_PIN 10
#define RST_PIN 9

MFRC522 mfrc522(SS_PIN, RST_PIN);

// Разрешённый идентификатор Card UID
byte uidCard[4] = {0x77, 0xE7, 0xBC, 0x3A};

void setup() {
  Serial.begin(9600);
  SPI.begin();

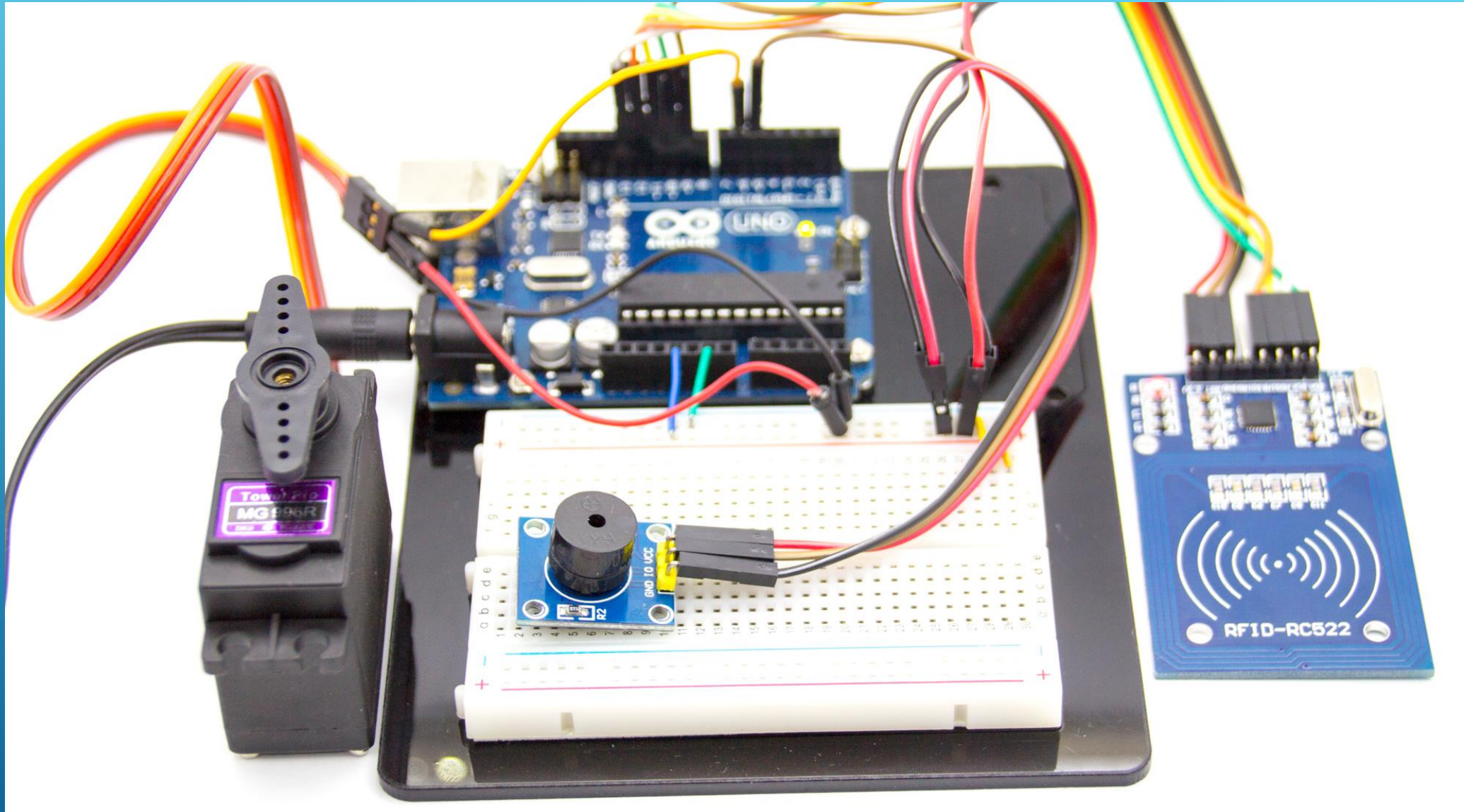
  mfrc522.PCD_Init();
}
```

```
void loop() {
  // ожидание
  if ( !mfrc522.PICC_IsNewCardPresent() )
    return;

  // чтение
  if ( !mfrc522.PICC_ReadCardSerial() )
    return;

  for (byte i = 0; i < 4; i++) {
    if (uidCard[i] != mfrc522.uid.uidByte[i])
    {
      Serial.println("У вас нет доступа");
      return;
    }else{
      Serial.println("Проходите!");
      // и открываем замок...
    }
  }
}
```

# ЭЛЕКТРОННЫЙ КОДОВЫЙ ЗАМОК



# СПАСИБО ЗА ВНИМАНИЕ!

