

## Оглавление

Датчик касания (сенсор датчик).....	3
Контакты и сигнал .....	3
Работа совместно с Arduino UNO.....	3
Датчик движения (HC-SR501).....	5
Контакты и сигнал .....	5
Работа совместно с Arduino UNO .....	6
Датчик температуры DS18B20.....	7
Контакты и сигнал .....	7
Работа совместно с Arduino UNO.....	7
Библиотека OneWire для работы с DS18B20.....	9
Датчик влажности почвы.....	10
Контакты и сигнал .....	10
Работа совместно с Arduino UNO.....	11
TFT LCD дисплей .....	12
Работа совместно с Arduino UNO.....	12
Датчика звука.....	13
Контакты и сигнал .....	13
Работа совместно с Arduino UNO.....	13
Passive Buzzer.....	15
Работа совместно с Arduino UNO.....	15
Датчик протечки и дождя .....	16
Контакты и сигнал .....	16
Работа совместно с Arduino UNO.....	16
Ультразвуковой дальномер HC-SR04.....	18
Контакты и сигнал .....	18
Работа совместно с Arduino UNO.....	18
Датчик температуры DHT22 .....	20
Контакты и сигнал .....	20
Работа совместно с Arduino UNO.....	20
HDC1080 Датчик влажности воздуха и температуры цифровой высокоточный.....	22
Контакты и сигнал .....	22
Работа совместно с Arduino UNO.....	23
Датчик уровня воды .....	24
Контакты и сигнал .....	24
Работа совместно с Arduino UNO.....	24
Датчик огня .....	25
Контакты и сигнал .....	25

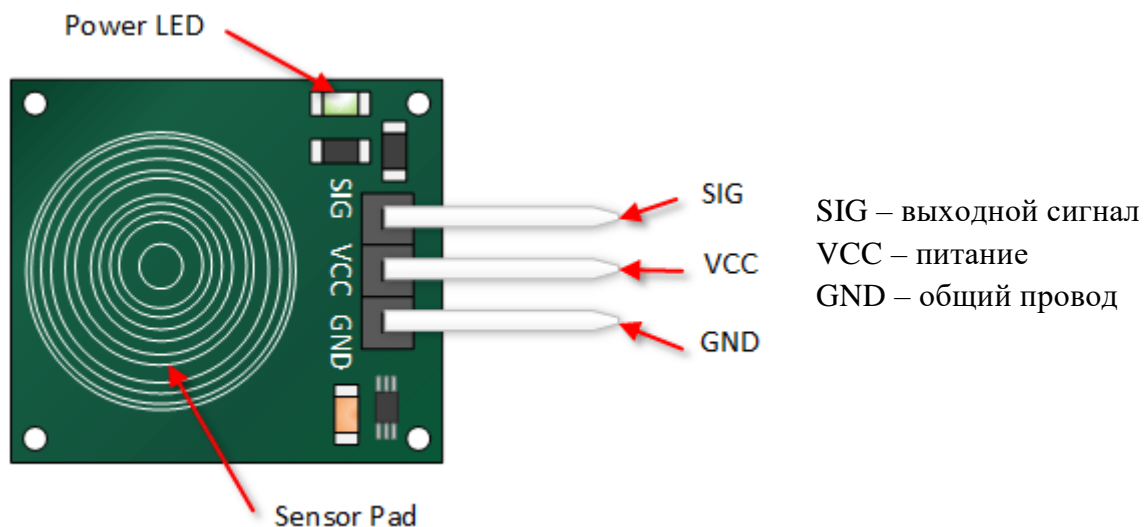
Работа совместно с Arduino UNO .....	25
Датчик широкого спектра газов MQ-2 .....	27
Контакты и сигнал .....	27
Контакты подключения трёхпроводных шлейфов .....	27
Работа совместно с Arduino UNO .....	28
Считыватель карт и брелоков RFID RC52 .....	30
Работа совместно с Arduino UNO .....	30
Модуль часов реального времени DS3231 .....	33
Контакты и сигнал .....	33
Работа совместно с Arduino UNO .....	33
Датчик освещённости.....	35
Контакты и сигнал .....	36
Работа совместно с Arduino UNO .....	37
ESP8266 NodeMcu v3 Lua с WiFi .....	38
Распиновка NodeMcu v3 .....	38
Пример подключения светодиода к NodeMCU .....	39
Датчик пыли GP2Y1010AU0F—SHARP .....	40
Контакты и сигнал .....	40
Работа совместно с Arduino UNO .....	41
Датчики давления bmp280 .....	42
Контакты и сигнал .....	42
Работа совместно с Arduino UNO .....	43
Задания к лабораторной работе.....	44
Вводная информация (ко всех темам).....	44
Тема 1. Охрана дома .....	44
Тема 2. Защита дома от пожара .....	44
Тема 3. Комфортный дом .....	45
Тема 4. Управление предметами по будильнику .....	45
Тема 5. Защита от затопления дома.....	45
Тема 6. Комфортный дом для родных.....	46
Дополнительные ссылки по датчикам.....	46

## *Датчик касания (сенсор датчик)*



Модуль представляет собой сенсорную кнопку, на его выходе формируется цифровой сигнал, напряжение которого соответствует уровням логических единицы и нуля. Относится к емкостным датчикам касания. С таких устройств ввода данных мы сталкиваемся при работе с дисплеем планшета, айфона или тачскрин монитора. Если на мониторе мы нажимаем на иконку стилусом или пальцем, то здесь для этого используется область поверхности платы размером с иконку Windows касание которой производится только пальцем, стилус исключается. Основа модуля микросхема TTP223-BA6. Есть индикатор питания.

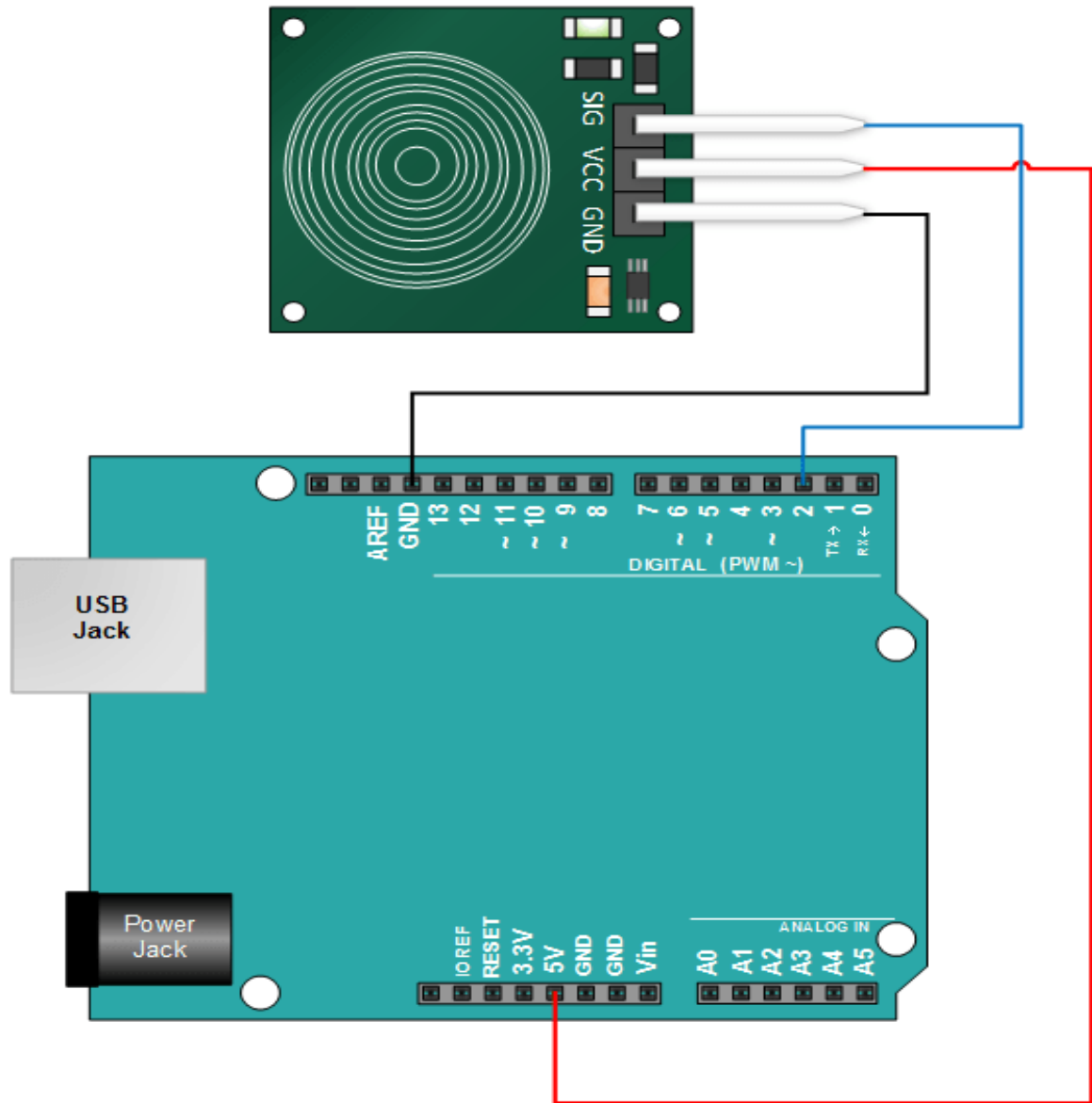
### *Контакты и сигнал*



Нет касания – выходной сигнал имеет низкий логический уровень, касание – на выходе датчика логическая единица.

### *Работа совместно с Arduino UNO*

Соедините датчик касания и Arduino UNO как показано на рисунке.



```
#define ctsPin 2 // Контакт подключения линии сигнала датчика касания
void setup() {
    Serial.begin(9600);
    pinMode(ctsPin, INPUT);
}

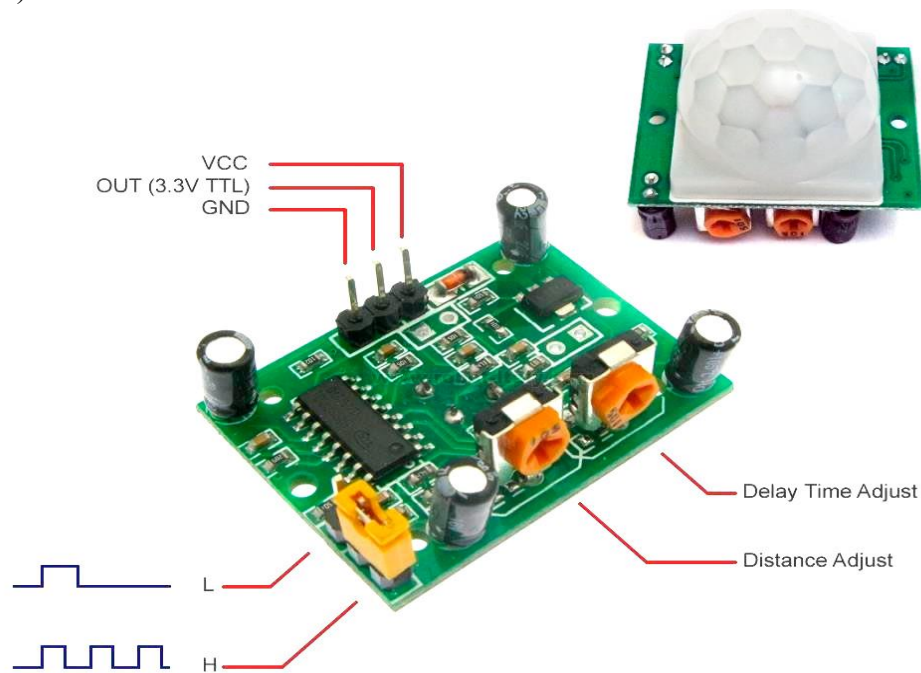
void loop() {
    int ctsValue = digitalRead(ctsPin);
    if (ctsValue == HIGH){
        Serial.println("touched ");
    }
    else{
        Serial.println("not touched");
    }
    delay(500);
}
```

## *Датчик движения (HC-SR501)*

Модуль HC-SR501 - модуль с пироэлектрическим инфракрасным (PIR) датчиком движения, как один из самых популярных и совместимых с Arduino. Чаще всего он используется в устройствах, предназначенных для управления освещением, и для этого может использоваться вкупе с датчиком освещённости.

На модуль установлена линза Френеля, которая фокусирует инфракрасные сигналы на пироэлектрический датчик под названием 500BP. Датчик называется PIR (Passive Infra-Red). Пассивный он потому, что для обнаружения движения не используется какая-либо дополнительная энергия, кроме той, что испускается самими объектами.

500BP состоит из двух чувствительных элементов. Управляющая микросхема модуля регистрирует изменения сигналов от обоих элементов и по характеру их изменения обнаруживает движение объектов, испускающих инфракрасные сигналы (живых организмов).



### **Контакты и сигнал**

Модуль HC-SR501 имеет 3 вывода:

- Питание (**VCC**);
- Земля (**GND**);
- Выход 3v3 (**OUT**).

Сразу после подачи питания несколько секунд модуль будет калиброваться, в это время возможны ложные срабатывания. Примерно через минуту он перейдёт в режим ожидания. При срабатывании датчика на выходе появляется логическая единица, напряжение – 3.3 вольта.

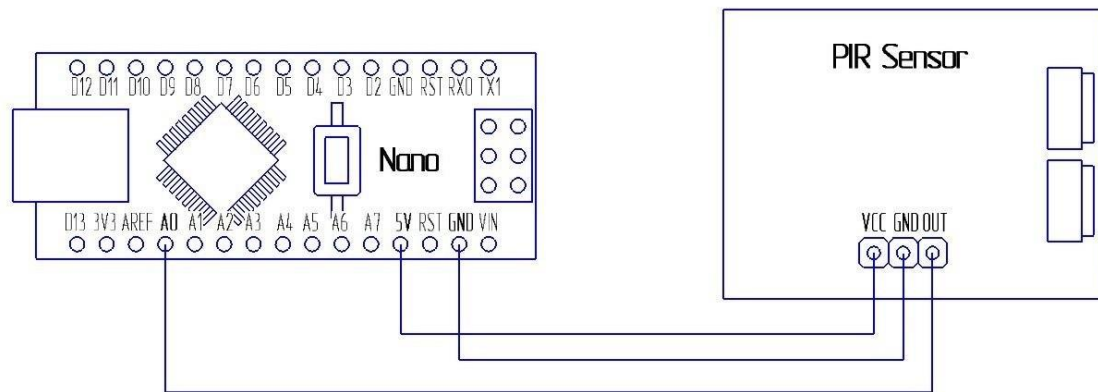
Также на самом модуле можно найти два переменных резистора, регулирующих дистанцию обнаружения движения (**Distance Adjust**) и время, в течение которого на выходе будет логическая единица (**Delay Time Adjust**). Дистанция регулируется в пределах 3 – 7 метров, задержка от 5 до 300 секунд.

И ещё немного о его особенностях. При работе с датчиком следует избегать источников света и тепла, закрывающих поверхность объектива модуля. Ветер также может создавать помехи. На большем расстоянии датчик более чувствителен.

### ***Работа совместно с Arduino UNO***

- **GND** подключаем к одноимённому выводу Arduino;
- **VCC** к **5V**;
- **OUT** подключим к **A0**.

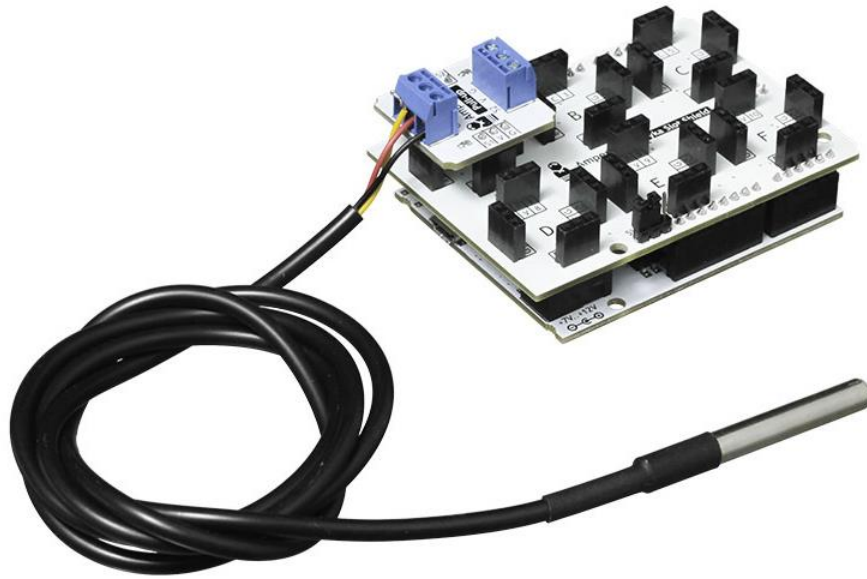
Подключение выхода модуля к аналоговому пину связано с тем, что цифровые пины Arduino работают с пятивольтовой логикой, а модуль рассчитан на 3.3 В.



```
void setup() {  
    Serial.begin(9600);  
}  
void loop() {  
    //Считываем пороговое значение с порта A0  
    if(analogRead(A0) > 500) {  
        //Сигнал с датчика движения  
        Serial.println("Есть движение!");  
    }  
    else {  
        //Нет сигнала  
        Serial.println("Нет движения...");  
    }  
}
```

## ***Датчик температуры DS18B20***

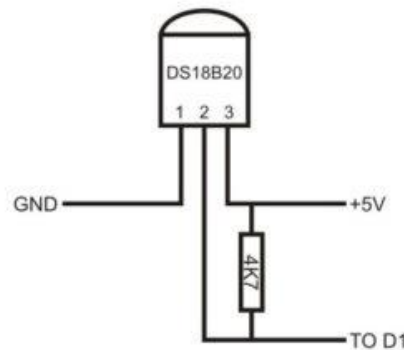
DS18B20 – это цифровой температурный датчик, обладающий множеством полезных функций. Цифровой датчик DS18B20 измерит температуру в воде, на земле и даже в космосе.



Датчик способен считывать показания температуры в диапазоне от  $-55$  до  $+125$  °C и передавать данные на управляющую плату всего через один пин.

### ***Контакты и сигнал***

Микросхема имеет три выхода, из которых для данных используется только один, два остальных – это земля и питание. Число проводов можно сократить до двух, если использовать схему с паразитным питанием и соединить Vdd с землей. К одному проводу с данными можно подключить сразу несколько датчиков DS18B20 и в плате Ардуино будет задействован всего один пин.



### ***Работа совместно с Arduino UNO***

Обмен информацией в 1-Wire происходит благодаря следующим операциям:

- Инициализация – определение последовательности сигналов, с которых начинается измерение и другие операции. Ведущее устройство подает импульс сброса, после этого датчик должен подать импульс присутствия, сообщающий о готовности к выполнению операции.
- Запись данных – происходит передача байта данных в датчик.
- Чтение данных – происходит прием байта из датчика.

Для работы с датчиком нам понадобится программное обеспечение:

- Arduino IDE;

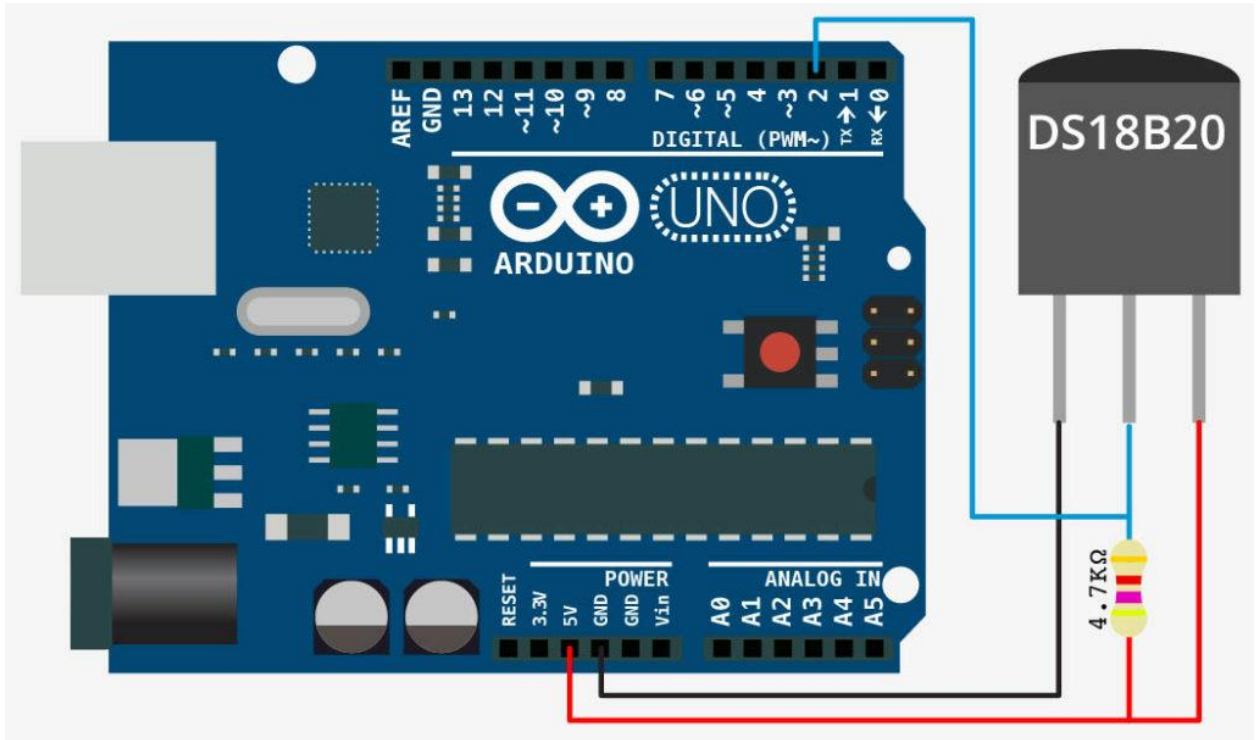


- Библиотека OneWire, если используется несколько датчиков на шине, можно использовать библиотеку DallasTemperature. Она будет работать поверх OneWire.

Из оборудования понадобятся:

- Один или несколько датчиков DS18B20;
- Резистор на 4,7 кОм (в случае подключения одного датчика пойдет резистор номиналом от 4 до 10K).

К плате Ардуино UNO датчик подключается просто: GND с термодатчика присоединяется к GND Ардуино, Vdd подключается к 5V, Data – к любому цифровому пину.



```
#include <OneWire.h>

/* Подключение ds18b20 к ардуино через пин 2 */
OneWire ds(2);
void setup(){
  Serial.begin(9600);
}

void loop(){
  // Определяем температуру от датчика DS18b20
  byte data[2]; // Место для значения температуры
  ds.reset(); // Начинаем взаимодействие со сброса всех предыдущих команд и параметров
  ds.write(0xCC); // Даем датчику DS18b20 команду пропустить поиск по адресу.
  ds.write(0x44); // Даем датчику DS18b20 команду измерить температуру. Само значение температуры мы еще не получаем - датчик его положит во внутреннюю память
  delay(1000); // Микросхема измеряет температуру, а мы ждем.
  ds.reset(); // Теперь готовимся получить значение измеренной температуры
  ds.write(0xCC);
```



```

ds.write(0xBE); // Просим передать нам значение регистров со значением
температуры
// Получаем и считываем ответ
data[0] = ds.read(); // Читаем младший байт значения температуры
data[1] = ds.read(); // А теперь старший
// Формируем итоговое значение: - сперва "склеиваем" значение, - затем
умножаем его на коэффициент, соответствующий разрешающей способности (для
12 бит по умолчанию - это 0,0625)
float temperature = ((data[1] << 8) | data[0]) * 0.0625;
// Выводим полученное значение температуры в монитор порта
Serial.println(temperature);
}

```

### **Библиотека OneWire для работы с DS18B20**

DS18B20 использует для обмена информацией с ардуино протокол 1-Wire, для которого уже написана отличная библиотека. Можно и нужно использовать ее, чтобы не реализовывать все функции вручную.

[Скачать OneWire можно здесь.](#)

Для установки библиотеки скачайте архив, распакуйте в папку library вашего каталога Arduino. Подключается библиотека с помощью команды `#include <OneWire.h>`

Основные команды библиотеки OneWire:

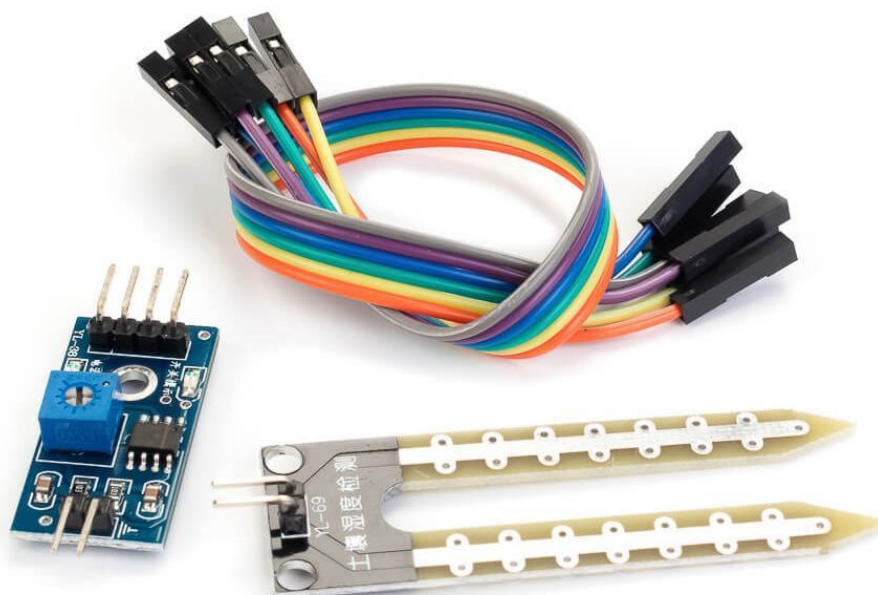
- `search(addressArray)` – ищет температурный датчик, при нахождении в массив `addressArray` записывается его код, в ином случае – `false`.
- `reset_search()` – производится поиск на первом приборе.
- `reset()` – выполнение сброса шины перед тем, как связаться с устройством.
- `select(addressArray)` – выбирается устройство после операции сброса, записывается его ROM код.
- `write(byte)` – производится запись байта информации на устройство.
- `write(byte, 1)` – аналогично `write(byte)`, но в режиме паразитного питания.
- `read()` – чтение байта информации с устройства.
- `crc8(dataArray, length)` – вычисление CRC кода. `dataArray` – выбранный массив, `length` – длина кода.

Важно правильно настроить режим питания в скетче. Для паразитного питания в строке 65 нужно записать `ds.write(0x44, 1);`. Для внешнего питания в строке 65 должно быть записано `ds.write(0x44)`.

Write позволяет передать команду на термодатчик. Основные команды, подаваемые в виде битов:

- `0x44` – измерить температуру, записать полученное значение в SRAM.
- `0x4E` – запись 3 байта в третий, четвертый и пятый байты SRAM.
- `0xBE` – последовательное считывание 9 байт SRAM.
- `0x48` – копирование третьего и четвертого байтов SRAM в EEPROM.
- `0xB8` – копирование информации из EEPROM в третий и четвертый байты SRAM.
- `0xB4` – возвращает тип питания (0 – паразитное, 1 – внешнее).

## *Датчик влажности почвы*



Модуль влажности почвы предназначен для определения влажности земли, в которую он погружен. Он позволяет узнать о недостаточном или избыточном поливе ваших домашних или садовых растений.

### **Контакты и сигнал**

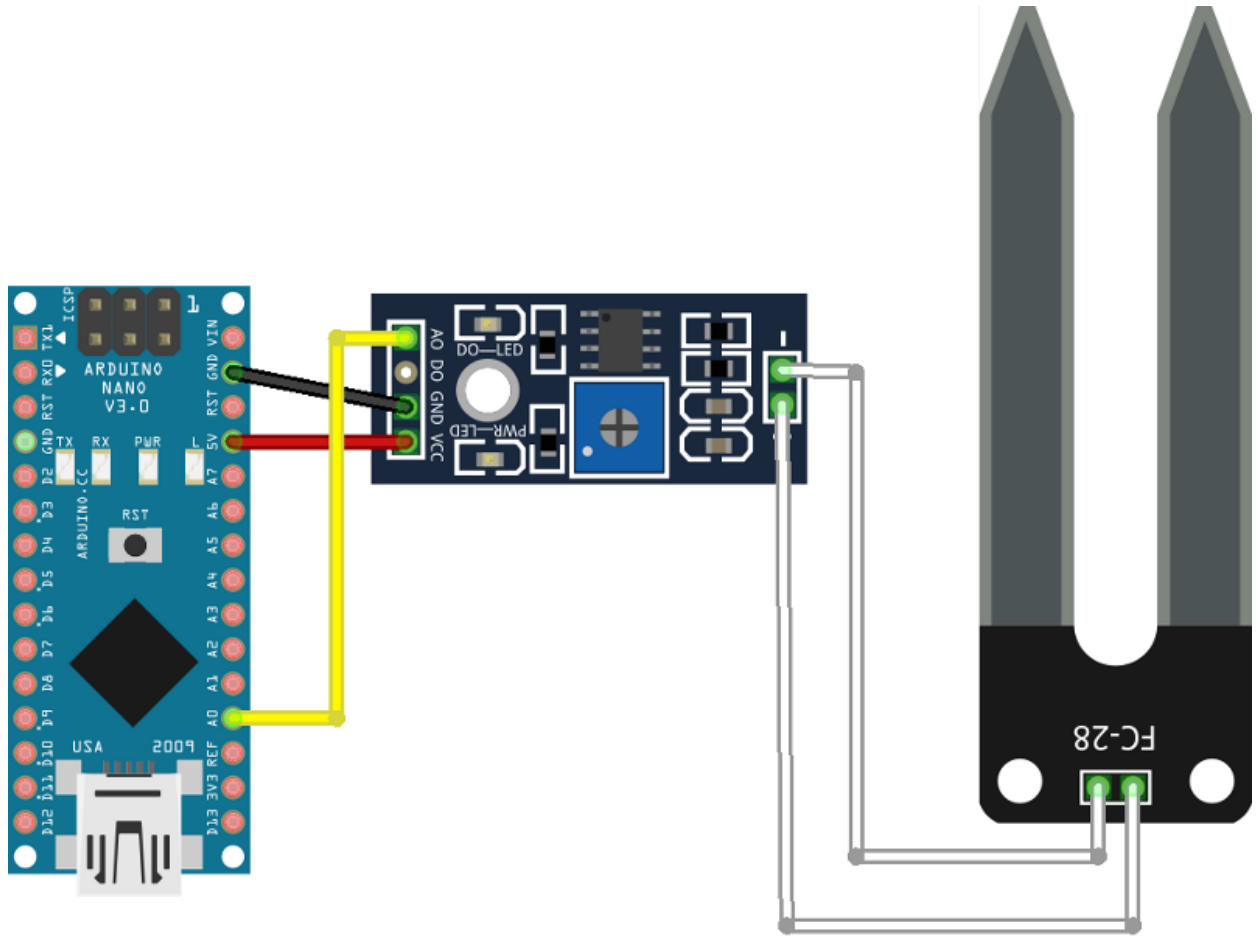
Модуль состоит из двух частей: контактного щупа YL-69 и датчика YL-38, в комплекте идут провода для подключения. Между двумя электродами щупа YL-69 создается небольшое напряжение. Если почва сухая, сопротивление велико и ток будет меньше. Если земля влажная — сопротивление меньше, ток — чуть больше. По итоговому аналоговому сигналу можно судить о степени влажности. Щуп YL-69 соединен с датчиком YL-38 по двум проводам.

Кроме контактов соединения с щупом, датчик YL-38 имеет четыре контакта для подключения к контроллеру.

- Vcc – питание датчика;
- GND – земля;
- A0 - аналоговое значение;
- D0 – цифровое значение уровня влажности.

Датчик YL-38 построен на основе компаратора LM393, который выдает напряжение на выход D0 по принципу: влажная почва – низкий логический уровень, сухая почва – высокий логический уровень. Уровень определяется пороговым значением, которое можно регулировать с помощью потенциометра. На вывод A0 подается аналоговое значение, которое можно передавать в контроллер для дальнейшей обработки, анализа и принятия решений. Датчик YL-38 имеет два светодиода, сигнализирующих о наличии поступающего на датчик питания и уровня цифрового сигнала на выходе D0.

## Работа совместно с Arduino UNO

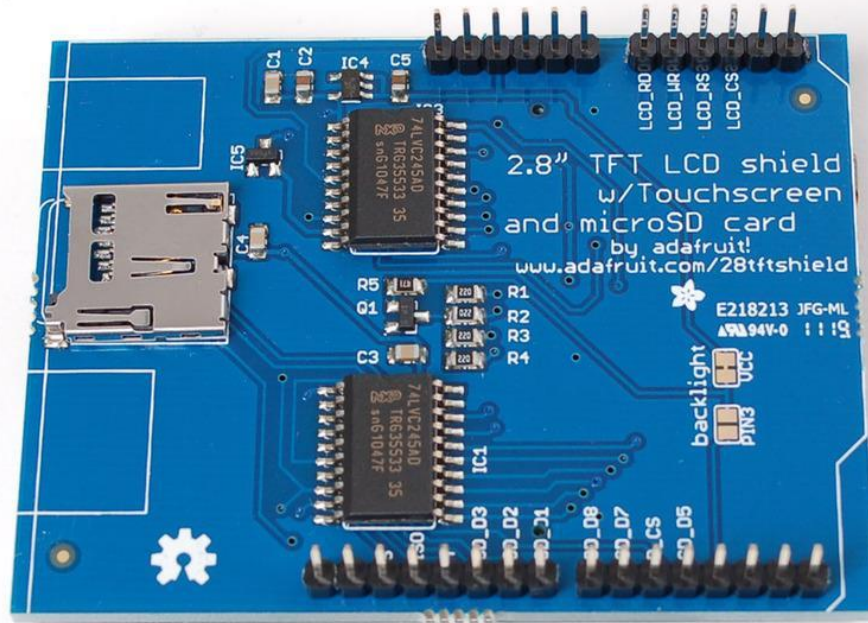


```
int sensorPin = A0;
int sensorValue = 0;

void setup()
{
  Serial.begin(57600);
  pinMode(13, OUTPUT);
}

void loop()
{
  sensorValue = analogRead(sensorPin);
  Serial.println(sensorValue);
  delay(200);
  if(sensorValue > 700) digitalWrite(13, HIGH);
  else digitalWrite(13, LOW);
}
```

## *TFT LCD дисплей*

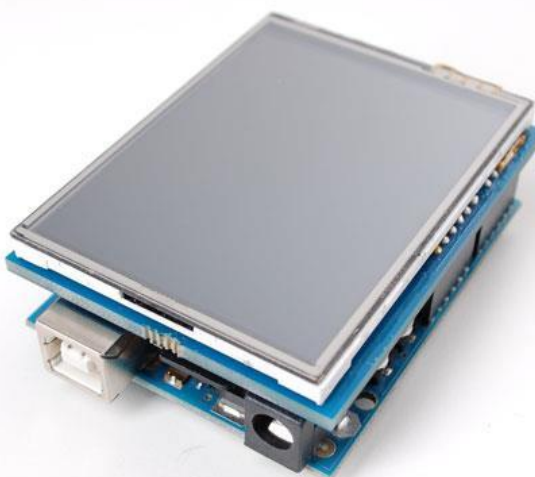


Touchscreen display (TFT) дисплей имеет диагональ 2.8", подсветку, в которой используется 4 белых светодиода, и возможность отображения 262000-оттенков цветов (18 бит)! Дисплей имеет разрешением 240x320 пикселей с индивидуальным управлением.

Достаточно его просто установить сверху на вашу Arduino и подключить необходимую библиотеку в Arduino IDE.

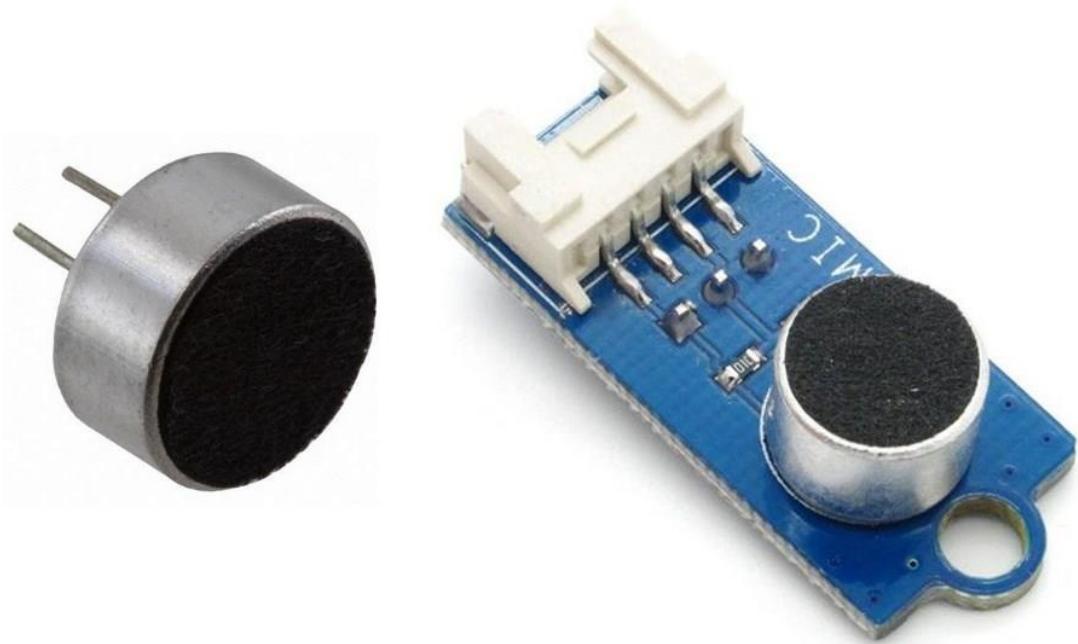
На TFT шилде есть встроенный контроллер с буферизацией RAM. В результате большинство операций передается именно шилду, а не грузят контроллер на Arduino. Для подключения шилда TFT дисплея достаточно несколько пинов: 12 пин отвечает за дисплей, 13 - за работу SD карты, если вы ее используете.

***Работа совместно с Arduino UNO***



## ***Датчика звука***

Электретный микрофон СМА-4544PF-W, который является основой модуля, реагирует на звуковые волны с частотами от 20 Гц до 20 кГц. Микрофон является всенаправленным, т.е. чувствителен к звуку, приходящему со всех направлений, с чувствительностью -44 дБ.

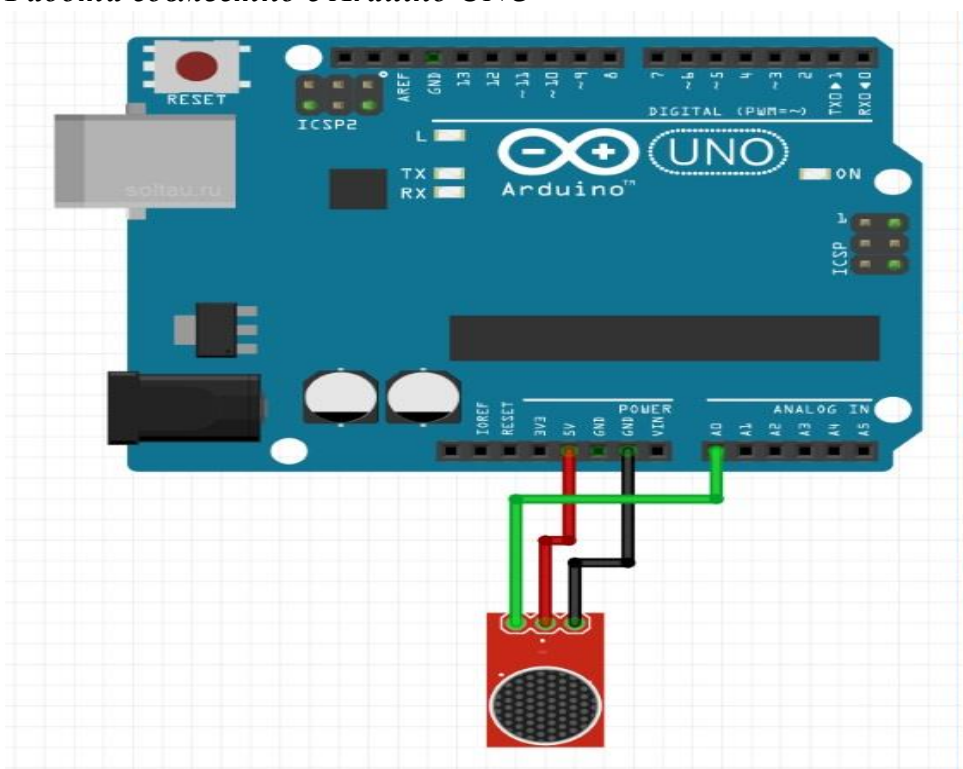


### ***Контакты и сигнал***

Модуль содержит в себе электретный микрофон, которому необходимо питание от 3 до 10 вольт. Полярность при подключении важна. Подключим модуль по простой схеме:

- вывод "V" модуля – к питанию +5 вольт,
- вывод "G" – к GND,
- вывод "S" – к аналоговому порту "A0" Arduino.

### ***Работа совместно с Arduino UNO***



```
const int micPin = A0; // задаём пин, куда подключён микрофон
```

```
void setup() {
```

```
    Serial.begin(9600); // инициализация послед. порта  
}
```

```
void loop() {
```

```
    int mv = analogRead(micPin) * 5.0 / 1024.0 * 1000.0; // значения в милливольтгах  
    Serial.println(mv); // выводим в порт  
}
```



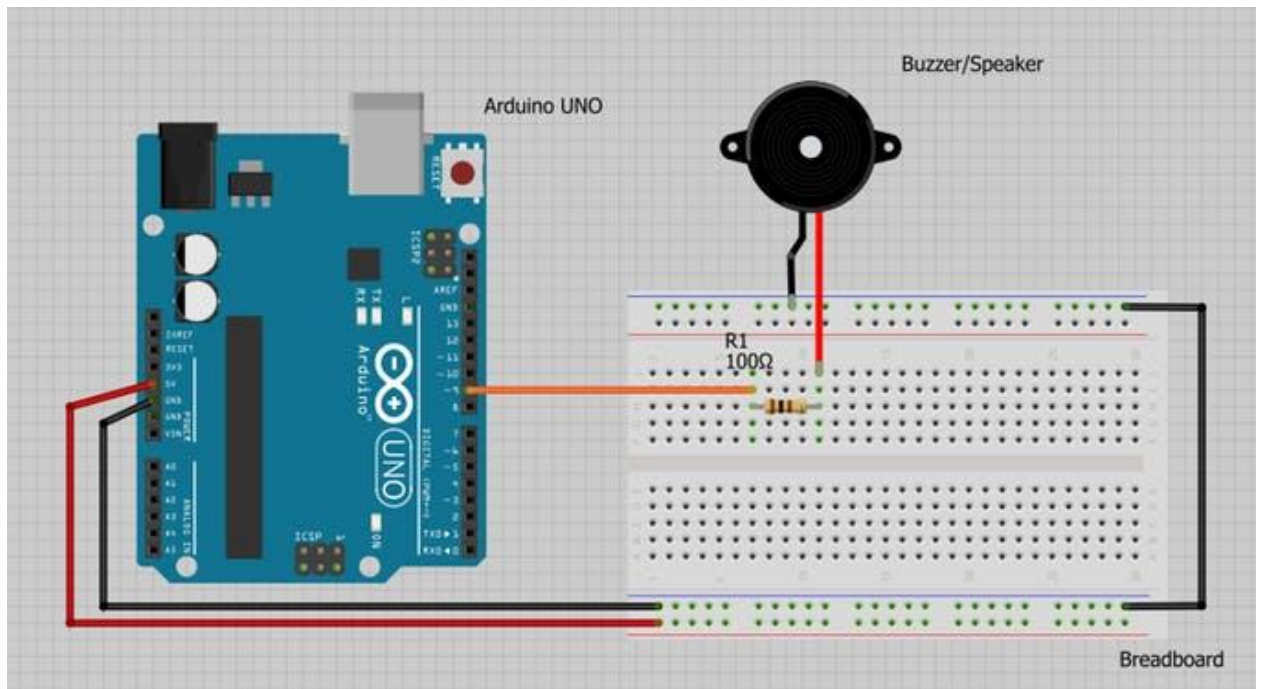
## *Passive Buzzer*



### *Работа совместно с Arduino UNO*

Для работы понадобится:

- Arduino uno
- Buzzer / piezo speaker
- 100 Ohm resistor



```
const int buzzer = 9; //buzzer to arduino pin 9
void setup(){
  pinMode(buzzer, OUTPUT); // Set buzzer - pin 9 as an output
}

void loop(){
  tone(buzzer, 1000); // Send 1KHz sound signal...
  delay(1000);      // ...for 1 sec
  noTone(buzzer);   // Stop sound...
  delay(1000);      // ...for 1sec
}
```



## *Датчик протечки и дождя*

Датчик протечки и дождя в проектах ардуино позволяет определить появление капель влаги и вовремя отреагировать на это, например, включив оповещение. Такие системы активно используются в аграрной отрасли, в автомобилестроении, и в других повседневных сферах нашей жизни.



### **Контакты и сигнал**

Модуль датчика состоит из двух частей:

- «Сенсорная» плата обнаружения капель. Она отслеживает количество попавшей на неё влаги. По сути, сенсор представляет собой простой переменный резистор, замыкаемый водой в разных местах, что вызывает изменение сопротивления.
- Вторая часть датчика – сдвоенный компаратор. Его главная задача – преобразование значения с сенсора в аналоговый сигнал от 0 до 5 вольт.

### **Работа совместно с Arduino UNO**

Если вы хотите проверять интенсивность осадков, то рекомендуется расположить датчик не горизонтально, а под некоторым углом, чтобы накапливаемые капли стекали вниз.

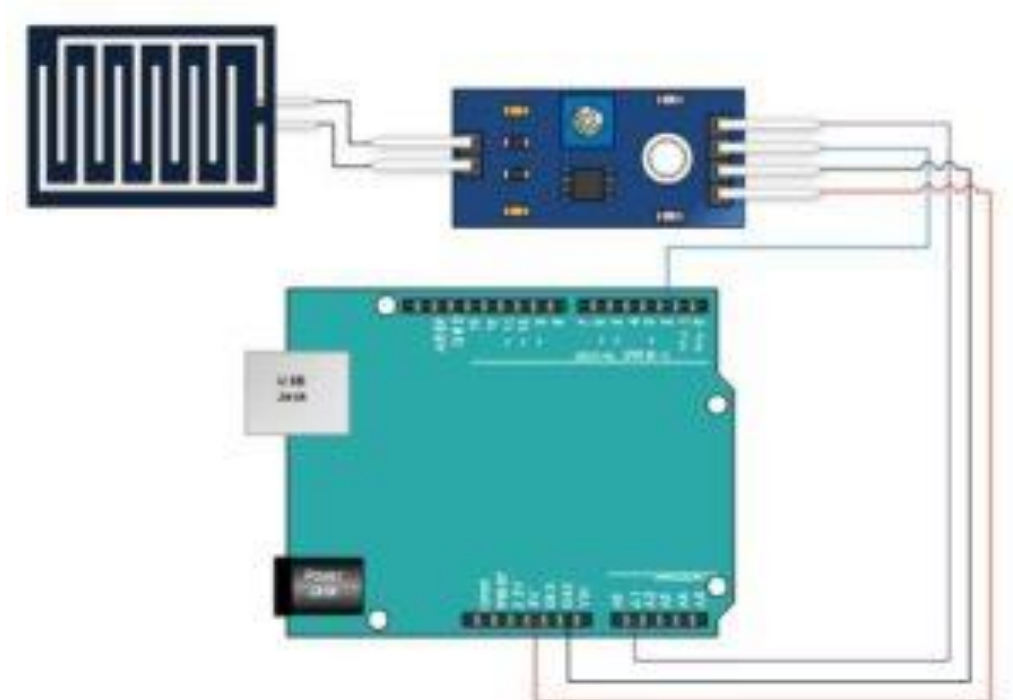


Схема подключения модуля датчика протечки к ардуино:

- VCC (вход питания) – должен совпадать для соединенной схемы ардуино по напряжению и току. То есть в данном случае 5В;
- GND – заземление;
- AO – аналоговый выход;
- DO – цифровой выход.

Аналоговый выход присоединяем к аналоговому пину микроконтроллера, например, A1. Цифровой выход, соответственно подключается к одному из цифровых пинов. Напряжение можно подать с вывода 5В платы ардуино, земля соединяется с землей.

```
#define PIN_ANALOG_RAIN_SENSOR A1 // Аналоговый вход для сигнала датчика
протечки и дождя
```

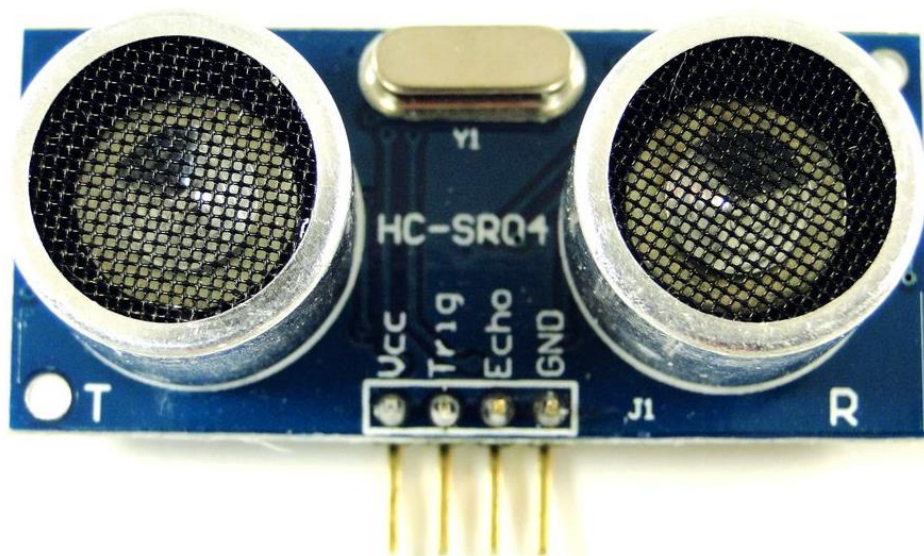
```
#define PIN_DIGITAL_RAIN_SENSOR 5 // Цифровой вход для сигнала датчика протечки
и дождя
```

```
void setup(){
  Serial.begin(9600);
}
```

```
void loop() {
  int sensorValue = analogRead(PIN_ANALOG_RAIN_SENSOR); // Считываем данные с
  аналогового порта
  Serial.print("Analog value: ");
  Serial.println(sensorValue); // Выводим аналоговое значение в монитор порта
  sensorValue = digitalRead(PIN_DIGITAL_RAIN_SENSOR); // Считываем данные с
  цифрового порта
  Serial.print("Digital value: ");
  Serial.println(sensorValue); // Выводим цифровое значение в монитор порта
  delay(1000); // Задержка между измерениями
}
```

## ***Ультразвуковой дальномер HC-SR04***

Дальномер — это устройство для измерения расстояния до некоторого предмета. Дальномер помогает роботу в разных ситуациях. Простой колесный робот может использовать этот прибор для обнаружения препятствий. Летающий дрон использует дальномер для баражирования над землей на заданной высоте. С помощью дальномера можно даже построить карту помещения, применив специальный алгоритм SLAM.



Этот популярный дальномер умеет измерять расстояние от 1-2 см до 4-6 метров. При этом, точность измерения составляет 0.5 — 1 см.

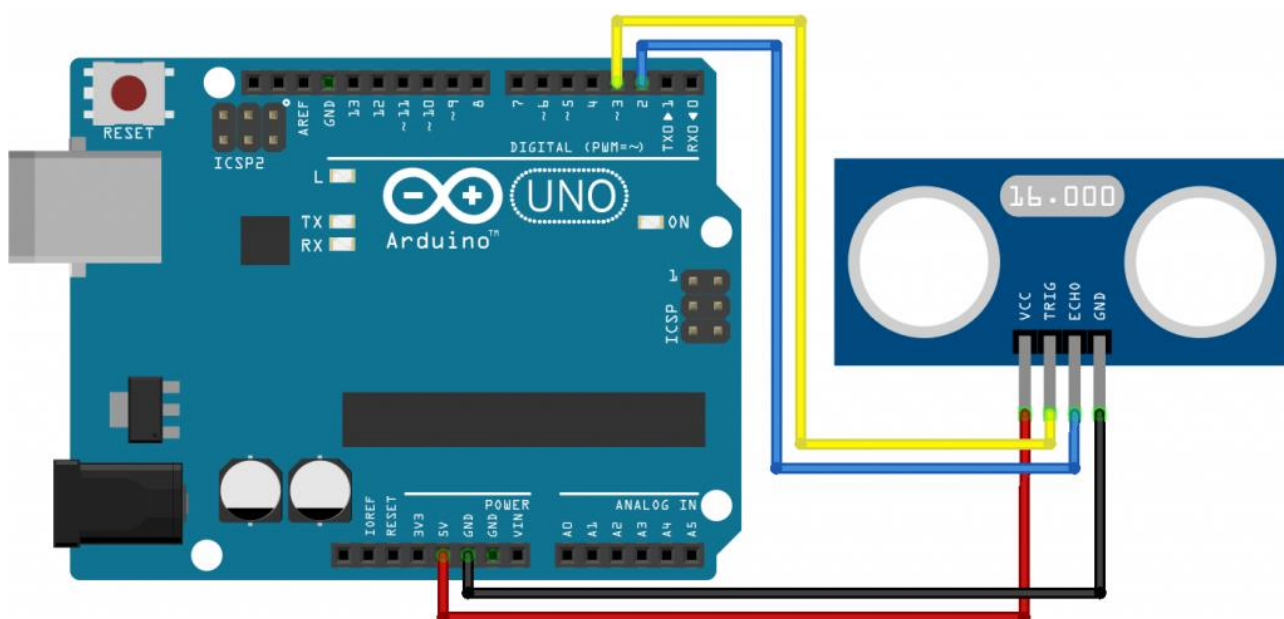
### ***Контакты и сигнал***

Датчик HC-SR04 имеет четыре вывода.

Кроме земли (Gnd) и питания (Vcc) еще есть Trig и Echo.

Оба этих вывода цифровые, так что подключаем их к любым выводам Ардуино Уно.

### ***Работа совместно с Arduino UNO***



fritzing

```

int echoPin = 2;
int trigPin = 3;
void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  int duration, cm;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  cm = duration / 58;
  Serial.print(cm);
  Serial.println(" cm");
  delay(100);
}

```

*Магическое число 58 в скетче.*

Требуется умножить время на скорость звука:

$$s = \textit{duration} * v = \textit{duration} * 340 \text{ м/с}$$

Переводим скорость звука из м/с в см/мкс:

$$s = \textit{duration} * 0.034 \text{ м/мкс}$$

Для удобства преобразуем десятичную дробь в обыкновенную:

$$s = \textit{duration} * 1/29 = \textit{duration} / 29$$

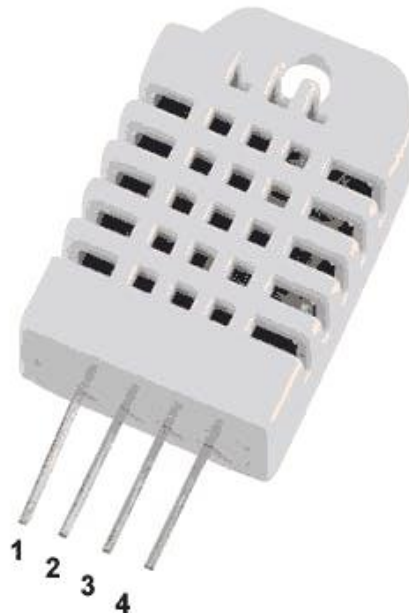
А теперь вспомним, что звук прошел два искомых расстояния: до цели и обратно.

Поделим всё на 2:

$$s = \textit{duration} / 58$$

## Датчик температуры DHT22

DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND



DHT22 Temperature-Humidity Sensor – это калиброванный цифровой модуль измерения температуры и влажности на основе датчика DHT22 (AM2302), который имеет более высокую точность и широкий диапазон измерения, чем DHT11. Он может быть использован для детектирования температуры и влажности окружающей среды, по средствам стандартного однопроводного интерфейса.

Диапазон влажности составляет 0-100%, а диапазон температур составляет -40 - 125 ° C.

### Контакты и сигнал

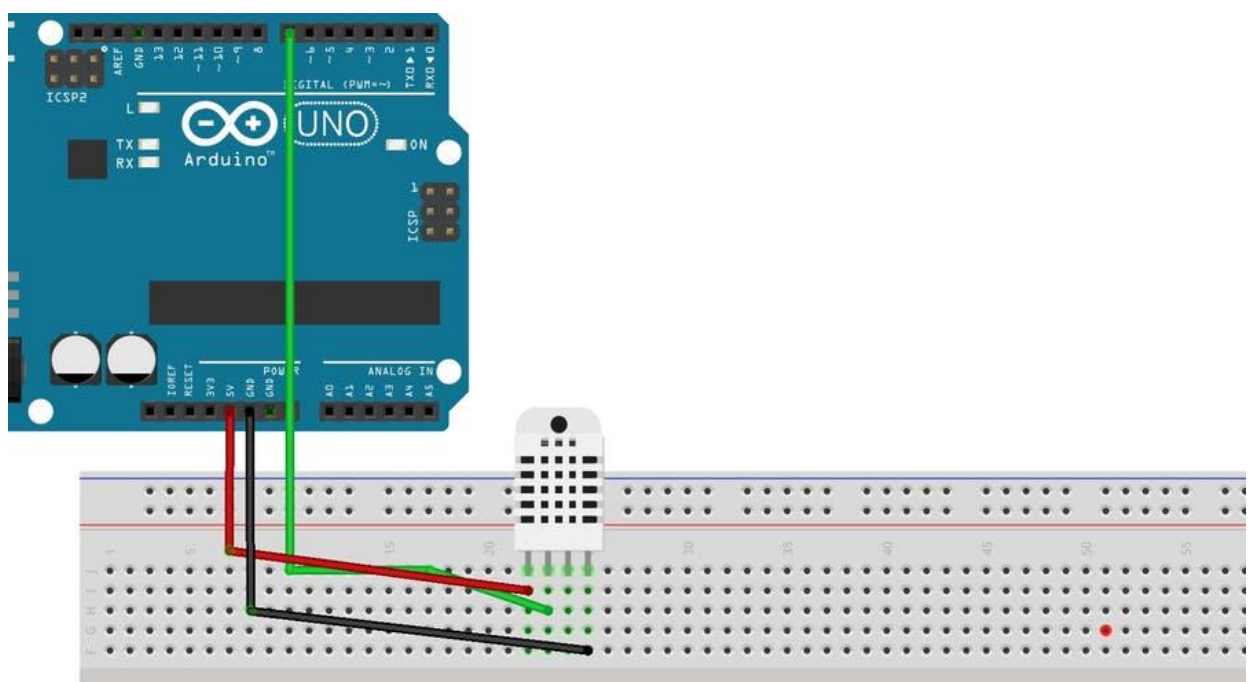
Первый контакт VCC подключите его к 5V.

Второй контакт - это контакт данных. Подключите его к цифровому контакту 7.

Третий пин называется NC его использовать не нужно.

Четвертый подключение к GND.

### Работа совместно с Arduino UNO



```

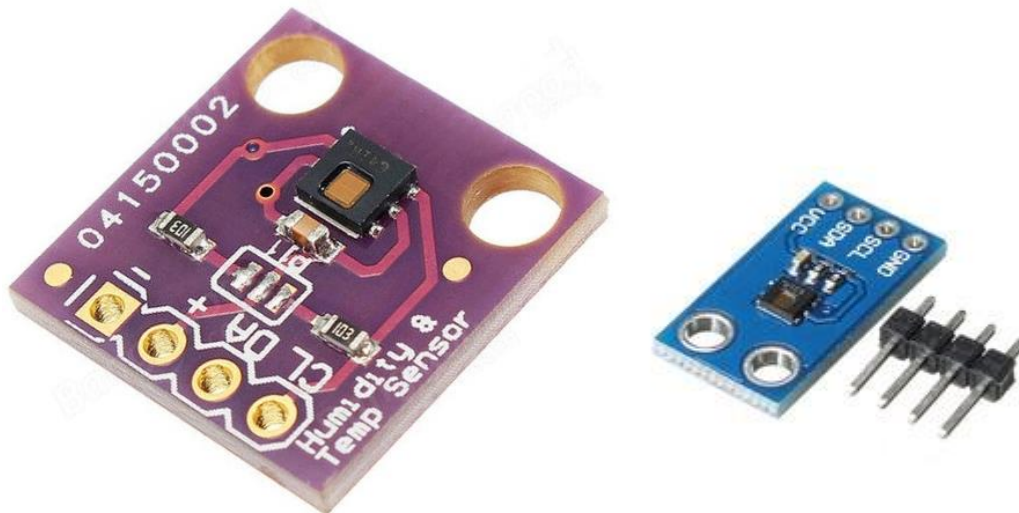
#include <DHT.h>;
#define DHTPIN 7    // what pin we're connected to
#define DHTTYPE DHT22 // DHT 22 (AM2302)
DHT dht(DHTPIN, DHTTYPE); /// Initialize DHT sensor for normal 16mhz Arduino

int chk;
float hum; //Stores humidity value
float temp; //Stores temperature value
void setup()
{
  Serial.begin(9600);
  dht.begin();
}

void loop()
{
  delay(2000);
  //Read data and store it to variables hum and temp
  hum = dht.readHumidity();
  temp= dht.readTemperature();
  //Print temp and humidity values to serial monitor
  Serial.print("Humidity: ");
  Serial.print(hum);
  Serial.print(" %, Temp: ");
  Serial.print(temp);
  Serial.println(" Celsius");
  delay(10000); //Delay 2 sec.
}

```

## ***HDC1080 Датчик влажности воздуха и температуры цифровой высокоточный***



HDC1080 - это цифровой датчик влажности со встроенным датчиком температуры, который обеспечивает превосходную точность измерения при очень низком энергопотреблении. HDC1080 работает в широком диапазоне поставок и является недорогой альтернативой с низким энергопотреблением конкурентным решениям в широком диапазоне общих приложений. Датчики влажности и температуры откалиброваны на заводе.

### **Характеристики:**

Диапазон измерений влажности: 0% - 100%

Типичная погрешность измерений:  $\pm 2\%$

Максимальная погрешность измерений:  $\pm 4\%$

Рекомендуемый производителем диапазон влажности: 0% - 80% (при большей влажности деградация датчика ускоряется и он может выйти из строя раньше)

Диапазон измерений температуры:  $-40^{\circ}\text{C} \dots +125^{\circ}\text{C}$

Погрешность измерений:  $\pm 0,2^{\circ}\text{C}$

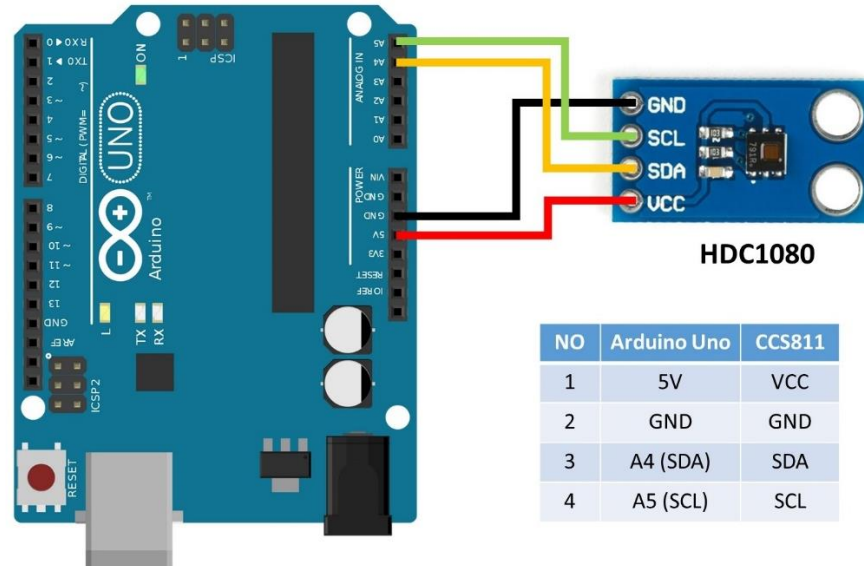
Напряжение питания: 2,7V - 5,5V

### **Контакты и сигнал**

Arduino connection	Module connection
3v3	3v3
GND	GND
SDA – A4	SDA
SCL – A5	SCL



## Работа совместно с Arduino UNO



```
#include <Wire.h>
#include "ClosedCube_HDC1080.h"
ClosedCube_HDC1080 hdc1080;
void setup()
{
    Serial.begin(9600);
    hdc1080.begin(0x40);
    Serial.print("Manufacturer ID=0x");
    Serial.println(hdc1080.readManufacturerId(), HEX); // 0x5449 ID of Texas Instruments
    Serial.print("Device ID=0x");
    Serial.println(hdc1080.readDeviceId(), HEX); // 0x1050 ID of the device
    printSerialNumber();
}
void loop()
{
    Serial.print("T=");
    Serial.print(hdc1080.readTemperature());
    Serial.print("C, RH=");
    Serial.print(hdc1080.readHumidity());
    Serial.println("%");
    delay(3000);
}
void printSerialNumber() {
    Serial.print("Device Serial Number=");
    HDC1080_SerialNumber sernum = hdc1080.readSerialNumber();
    char format[12];
    sprintf(format, "%02X-%04X-%04X", sernum.serialFirst, sernum.serialMid,
    sernum.serialLast);
    Serial.println(format);
}
```

## *Датчик уровня воды*



### **Контакты и сигнал**

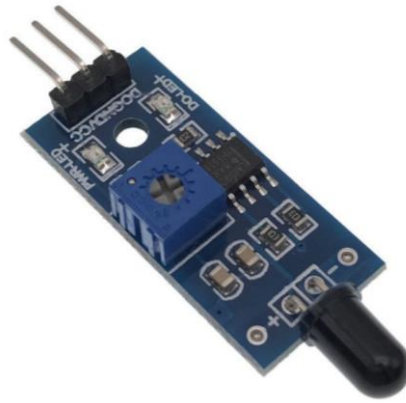
- Один провод датчика подключается к выводу GND (так как второй провод будет подтянут до уровня логической «1»).
- Другой провод можно подключить к любому выводу Arduino.

Так как при разомкнутой цепи датчика, состояние на входе Arduino будет неопределённым, то при конфигурировании вывода как вход, его стоит подтянуть до уровня логической «1», вызвав функцию `pinMode()`, с параметром `INPUT_PULLUP`.

### **Работа совместно с Arduino UNO**

```
const uint8_t pinSensor=2; // Создаём константу, указывая номер вывода к которому
                             // подключён датчик
void setup(){
  Serial.begin(9600);
  pinMode(pinSensor, INPUT_PULLUP); // Конфигурируем вывод как вход, подтягивая его
  // до уровня логической «1» через внутренний подтягивающий резистор Arduino
}
void loop(){
  if(!digitalRead(pinSensor)){ // Если на входе уровень логического «0», то ...
    Serial.println("СЕНСОР ТОНЕТ"); // Выводим сообщение о том что сенсор тонет (датчик
    // погрузился в воду)
  } delay(200); // Приостанавливаем выполнение скетча на 200 мс (0,2 сек)
}
```

## Датчик огня



Инфракрасный датчик огня улавливает излучение в диапазоне 760 — 1100 нм, свойственное пламени свечи, например. На практике, такой датчик реагирует не только на пламя, но и на солнце, и даже на комнатные лампы. Чтобы избежать паразитной засветки, фотодиод необходимо закрывать с боковых сторон непрозрачным материалом. Для лучшей фильтрации посторонних источников света, при детектировании пламени таким датчиком, применяют алгоритм детектирования низкой частоты. Это возможно благодаря тому, что пламя свечи меняет свою интенсивность с частотой 15-20 Гц.

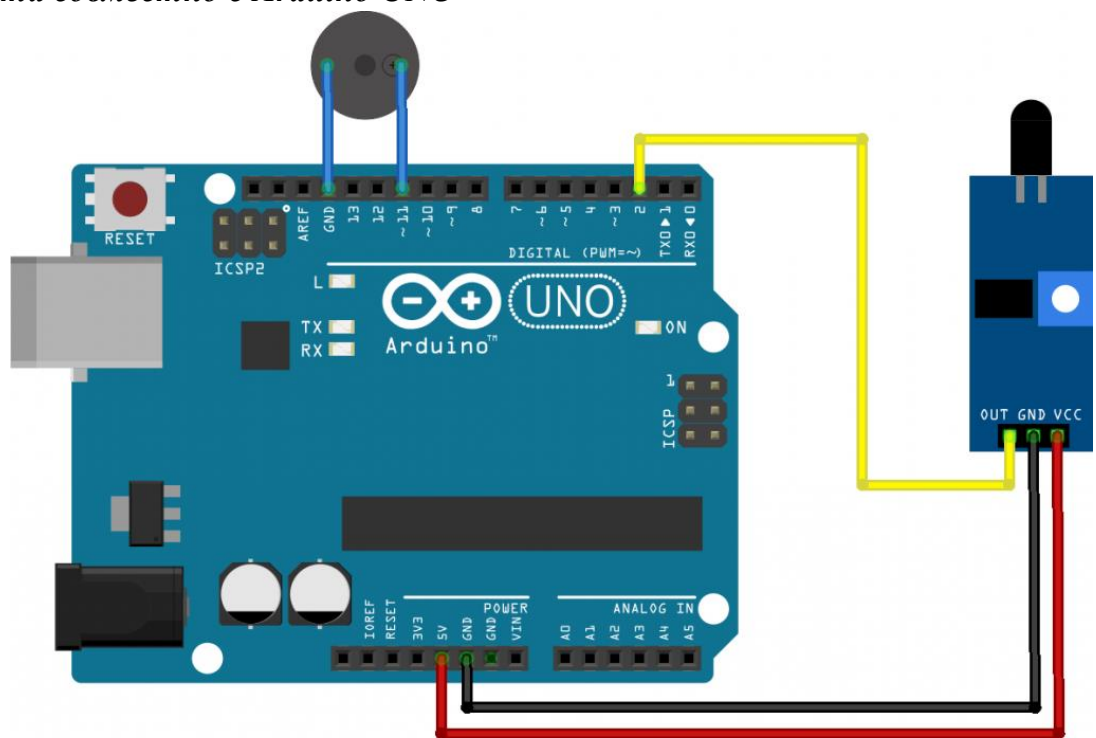
### Контакты и сигнал

У цифрового датчика пламени, который мы подключаем, есть всего три вывода:

- Vcc — питание +5В;
- Gnd — земля;
- Out — сигнал.

Vcc и Gnd датчика подключаем к соответствующим выводам Ардуино Уно, а Out бросаем на любую свободную ногу. В нашем случае, соединяем Out с цифровым входом №2.

### Работа совместно с Arduino UNO



fritzing

Использованный нами датчик пламени, имеет инвертированный выход, а значит, он будет возвращать ложь, если в пределах его видимости есть пламя, и истину — в отсутствии пламени. Напишем простую программу, которая будет включать зуммер, если датчик увидел перед собой огонь.

```
int flamePin = 2;
int buzzPin = 11;

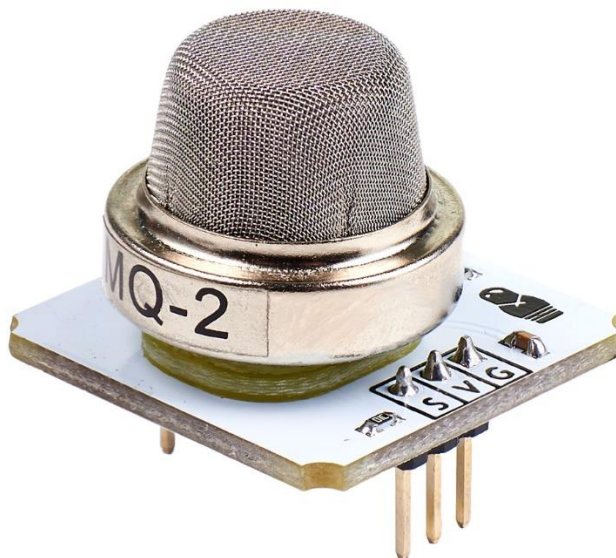
void setup() {
  pinMode( flamePin, INPUT );
  pinMode( buzzPin, OUTPUT );
}

void loop() {
  if( !digitalRead( flamePin ) )
    digitalWrite( buzzPin, HIGH );
  else
    digitalWrite( buzzPin, LOW );
}
```

Записываем программу на Ардуино Уно, достаем зажигалку с крестовой отверткой, и готовимся к последнему этапу — настройке чувствительности датчика. Дело в том, что на плате датчика пламени есть подстроечный потенциометр, с помощью которого мы и будем настраивать порог чувствительности. Для этого, включаем Ардуино Уно в USB, чтобы запитать нашу схему, поджигаем огонь в 10 сантиметрах от датчика, и начинаем крутить потенциометр, пока зуммер не запищит.

## Датчик широкого спектра газов MQ-2

Датчик MQ-2 определит концентрацию углеводородных газов (пропан, метан, н-бутан), дыма (взвешенных частиц, являющихся результатом горения) и водорода в окружающей среде.



Датчик MQ-2 относится к полупроводниковым приборам. Принцип работы датчика основан на изменении сопротивления тонкопленочного слоя диоксида олова  $\text{SnO}_2$  при контакте с молекулами определяемого газа. Чувствительный элемент датчика состоит из керамической трубки с покрытием  $\text{Al}_2\text{O}_3$  и нанесенного на неё чувствительного слоя диоксида олова. Внутри трубки проходит нагревательный элемент, который нагревает чувствительный слой до температуры, при которой он начинает реагировать на определяемый газ. Чувствительность к разным газам достигается варьированием состава примесей в чувствительном слое.

### Контакты и сигнал

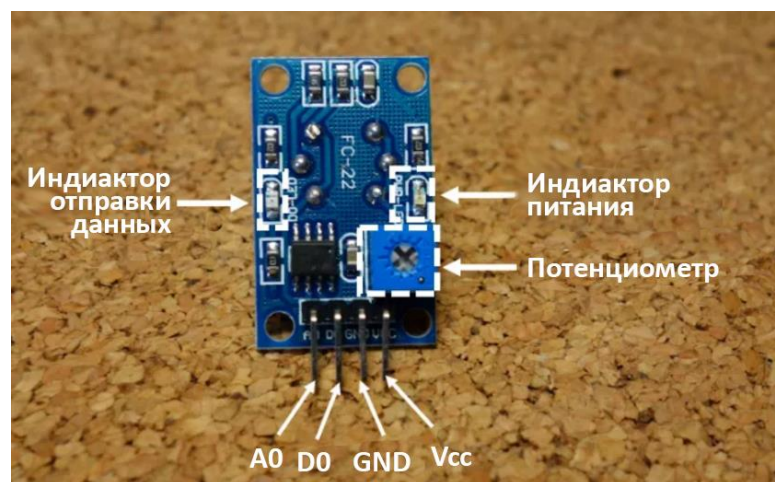
Контакты подключения трёхпроводных шлейфов

#### 1 группа

- Сигнальный (S) — Выходной сигнал сенсора. Подключите к аналоговому входу микроконтроллера.
- Питание (V) — Питание датчика. Соедините с рабочим напряжением микроконтроллера.
- Земля (G) — Соедините с пином `GND` микроконтроллера.

#### 2 группа

- Сигнальный (E) — Управление питанием нагревателя. Подключите к цифровому пину микроконтроллера.
- Питание (H) — Питание нагревателя. Соедините с пином `5V`
- Земля (G) — Соедините с пином `GND` микроконтроллера..



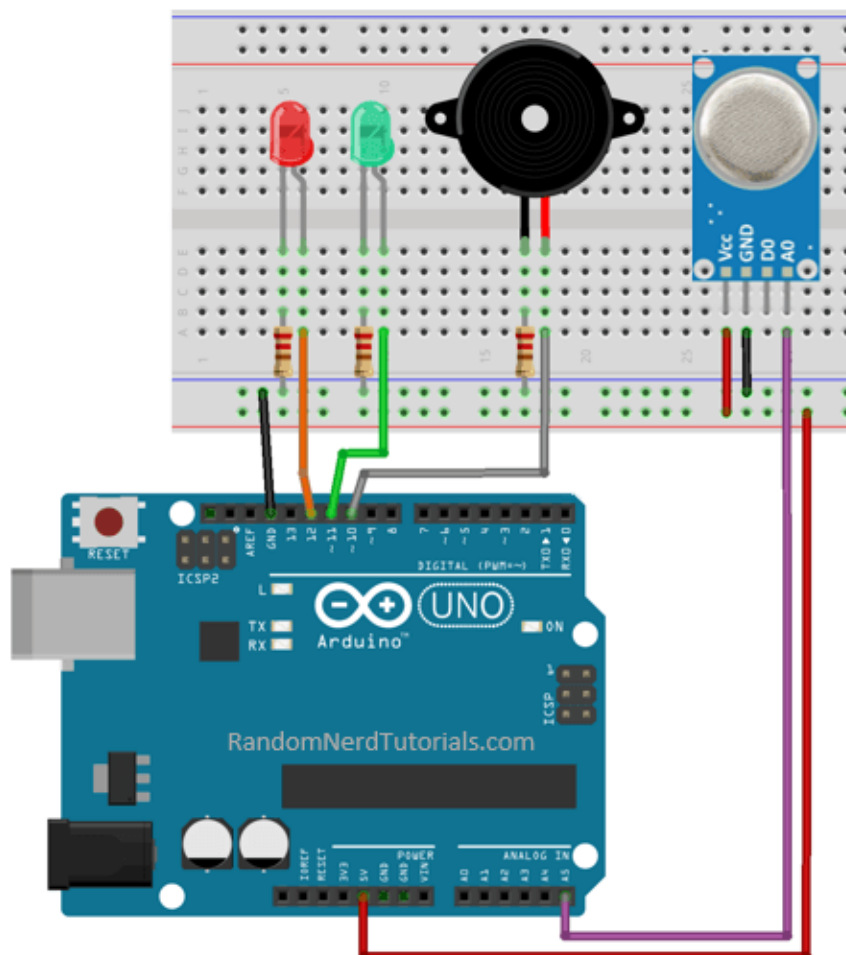
### ***Работа совместно с Arduino UNO***

Для работы понадобятся следующие компоненты:

- Один датчик газа MQ-2
- Одна плата Arduino
- Один красный светодиод
- Один зеленый светодиод
- Один зуммер
- Три резистора на 220 Ом

Датчик MQ-2 оснащен 4 контактами:

- Контакт A0 на MQ-2 нужно подключить к аналоговому контакту на Arduino
- Контакт D0 – к цифровому контакту
- Контакт GND – к контакту GND
- Контакт VCC – к контакту 5V



```

int redLed = 12;
int greenLed = 11;
int buzzer = 10;
int smokeA0 = A5;
// здесь задаем пороговое значение:
int sensorThres = 400;

void setup() {
  pinMode(redLed, OUTPUT);
  pinMode(greenLed, OUTPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(smokeA0, INPUT);
  Serial.begin(9600);
}

void loop() {
  int analogSensor = analogRead(smokeA0);

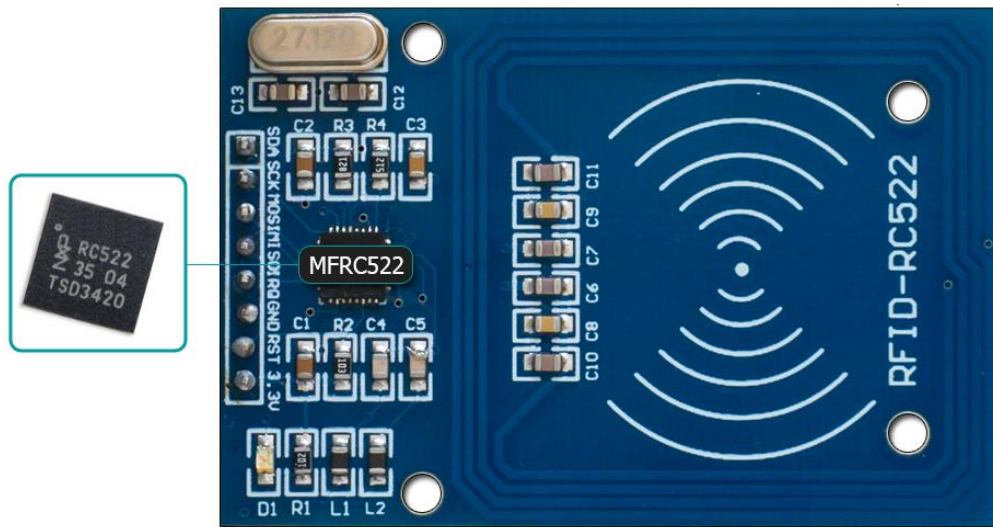
  Serial.print("Pin A0: "); // "Контакт A0: "
  Serial.println(analogSensor);
  // проверяем, не превышено ли пороговое значение:
  if (analogSensor > sensorThres)
  {
    digitalWrite(redLed, HIGH);
    digitalWrite(greenLed, LOW);
    tone(buzzer, 1000, 200);
  }
  else
  {
    digitalWrite(redLed, LOW);
    digitalWrite(greenLed, HIGH);
    noTone(buzzer);
  }
  delay(100);
}

```



## Считыватель карт и брелоков RFID RC52

Модуль RFID-RC522 выполнен на микросхеме MFRC522 фирмы NXP. Эта микросхема обеспечивает двухстороннюю беспроводную (до 6 см) коммуникацию на частоте 13,56 МГц.



С помощью данного модуля можно записывать и считывать данные с различных RFID-меток: брелоков от домофонов, пластиковых карточек-пропусков и билетов на метро и наземный транспорт, а также набирающих популярность NFC-меток.

### Работа совместно с Arduino UNO

Для начала необходимо установить библиотеку [\*RFID Library for MFRC522\*](#).

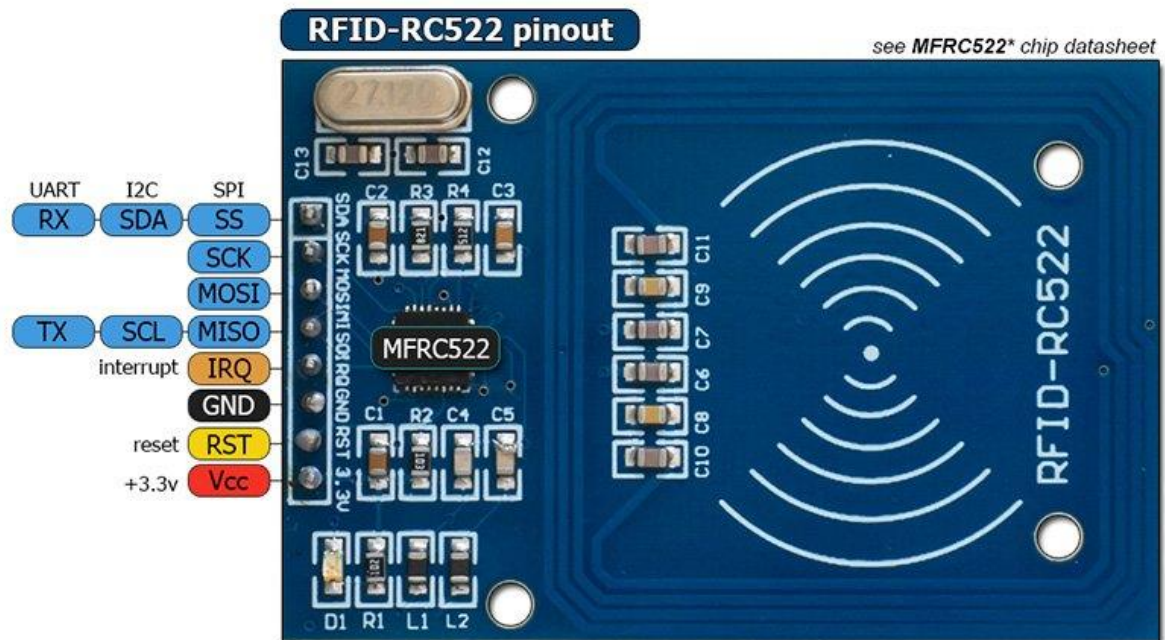
Контакты на модуле RFID-модуль RC522 необходимо подключить к Ардуине. Для подключения удобно использовать провода папа-мама.

Описание контактов у RFID-модуля RC522:

- VCC — Питание. Необходимо 3.3V;
- RST — Reset. Линия сброса. Ни в коем случае не подключать к пину RESET на CraftDuino! Данный пин цепляется на цифровой порт с PWM;
- GND — Ground. Земля
- MISO — Master Input Slave Output — данные от ведомого к ведущему, SPI;
- MOSI — Master Output Slave Input — данные от ведущего к ведомому, SPI;
- SCK — Serial Clock — тактовый сигнал, SPI;
- NSS — Slave Select — выбор ведомого, SPI;
- IRQ — линия прерываний;

MFRC522	<u>Arduino Uno</u>	<u>Arduino Mega</u>	<u>Arduino Nano v3</u>
RST	9	5	D9
SDA(SS)	10	53	D10
MOSI	11 (ICSP-4)	51	D11
MISO	12 (ICSP-1)	50	D12
SCK	13 (ICSP-3)	52	D13
3.3V	3.3V	3.3V	<u>Стабилизатор 3,3В</u>
GND	GND	GND	GND

После того как все будет подключено на модуле будет гореть индикатор, это говорит о том что питание поступает на RFID.



Загружаем скетч.

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

#define RST_PIN          9           //
#define SS_PIN           10          //

MFRC522 mfc522(SS_PIN, RST_PIN);   // Create MFRC522 instance

void setup() {
    Serial.begin(9600);               // Initialize serial communications with the PC
    while (!Serial);                 // Do nothing if no serial port is opened (added for Arduinos
    based on ATMEGA32U4)
    SPI.begin();                     // Init SPI bus
    mfc522.PCD_Init();               // Init MFRC522
    ShowReaderDetails();             // Show details of PCD - MFRC522 Card Reader details
    Serial.println(F("Scan PICC to see UID, type, and data blocks..."));
}

void loop() {
    // Look for new cards
    if ( ! mfc522.PICC_IsNewCardPresent()) {
        return;
    }

    // Select one of the cards
    if ( ! mfc522.PICC_ReadCardSerial()) {
        return;
    }
}
```

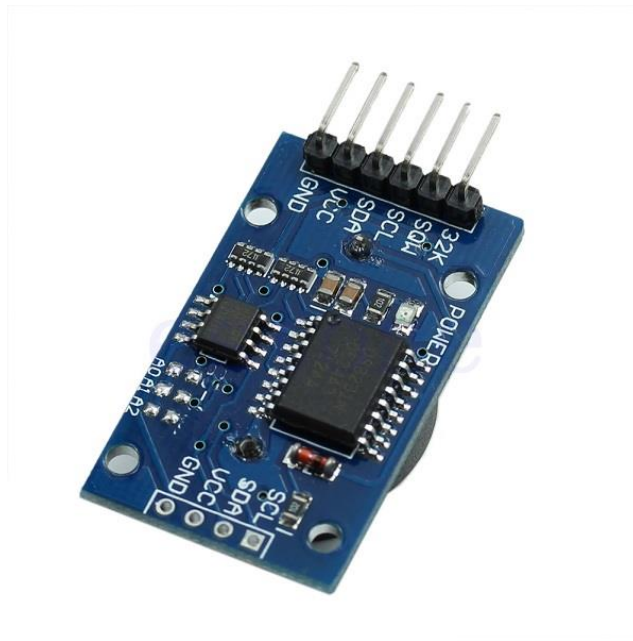
```

    // Dump debug info about the card; PICC_HaltA() is automatically called
    mfrc522.PICC_DumpToSerial(&(mfrc522.uid));
}

void ShowReaderDetails() {
    // Get the MFRC522 software version
    byte v = mfrc522.PCD_ReadRegister(mfrc522.VersionReg);
    Serial.print(F("MFRC522 Software Version: 0x"));
    Serial.print(v, HEX);
    if (v == 0x91)
        Serial.print(F(" = v1.0"));
    else if (v == 0x92)
        Serial.print(F(" = v2.0"));
    else
        Serial.print(F(" (unknown)"));
    Serial.println("");
    // When 0x00 or 0xFF is returned, communication probably failed
    if ((v == 0x00) || (v == 0xFF)) {
        Serial.println(F("WARNING: Communication failure, is the MFRC522 properly
connected?"));
    }
}

```

## Модуль часов реального времени DS3231



Модуль часов реального времени — это электронная схема, предназначенная для учета хронометрических данных (текущее время, дата, день недели и др.), представляет собой систему из автономного источника питания и учитывающего устройства.

Модуль *DS3231* по сути представляет из себя обыкновенные часы. В платах Arduino уже есть встроенный датчик времени *Millis*, однако он работает только при поданном питании на плату. При отключении и дальнейшем включении Arduino отсчет времени *Millis* сбросится до нуля. А *DS3231* имеет на борту батарейку, которая даже при отключенной плате Arduino продолжает «питать» модуль, позволяя ему измерять время.

Модуль можно использовать в качестве часов или будильника, построенных на базе плат Arduino. Или же в качестве оповещения для различных систем.

### Контакты и сигнал

- модуль производит подсчет часов, минут, секунд, дат, месяцев, лет (високосные года учитываются до 2100 года);

- для подключения к различным устройствам, часы подключаются по I2C интерфейсу.

**32K** — Выход, предназначенный для подачи внешнего питания >12В.

**SQW** — Программируемый выход Square-Wave сигнала.

**SCL** — Через этот пин по интерфейсу I2C происходит обмен данными с часами.

**SDA** — Через этот пин передаются данные с часов.

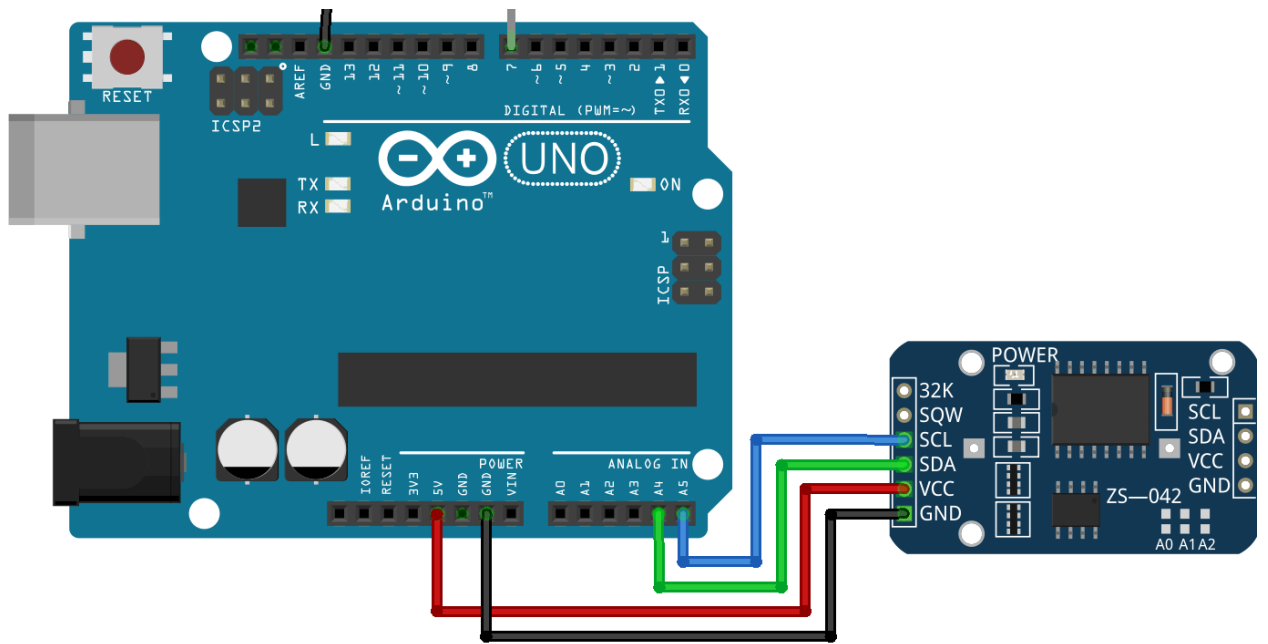
**VCC** — Питание часов реального времени, нужно 5 вольт. Если на этот пин не поступает напряжение, часы переходят в спящий режим.

**GND** — Земля.

### Работа совместно с Arduino UNO

Пины SDA и SCL на разных платах Arduino:

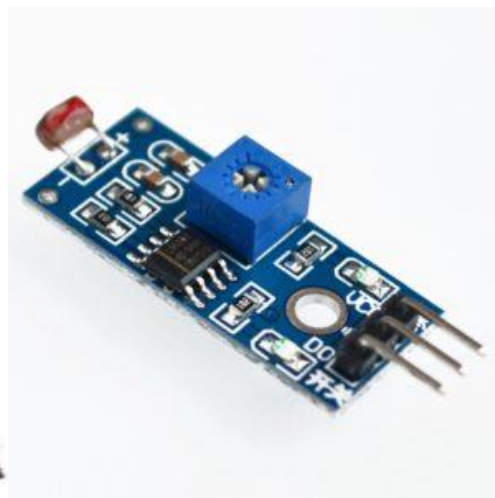
	SDA	SCL
UNO	A4	A5
Mini	A4	A5
Nano	A4	A5
Mega2560	20	21
Leonardo	2	3



fritzing

```
#include <Wire.h>
#include <DS1307RTC.h>
#include <Time.h>
tmElements_t datetime;
void setup() {
    Serial.begin(9600);
}
void loop() {
    // получение данных из ds3231
    if (RTC.read(datetime)) {
        print2(datetime.Hour, ":");
        print2(datetime.Minute, ":");
        print2(datetime.Second, " ");
        print2(datetime.Day, "/");
        print2(datetime.Month, "/");
        print2(tmYearToCalendar(datetime.Year), "");
        Serial.println();
    }
    else {
        Serial.println("error");
        delay(5000);
    }
    delay(1000);
}
void print2(int nn,String str) {
    if (nn >= 0 && nn < 10)
        { Serial.print("0");}
    Serial.print(nn);
    Serial.print(str);
}
```

## *Датчик освещённости*

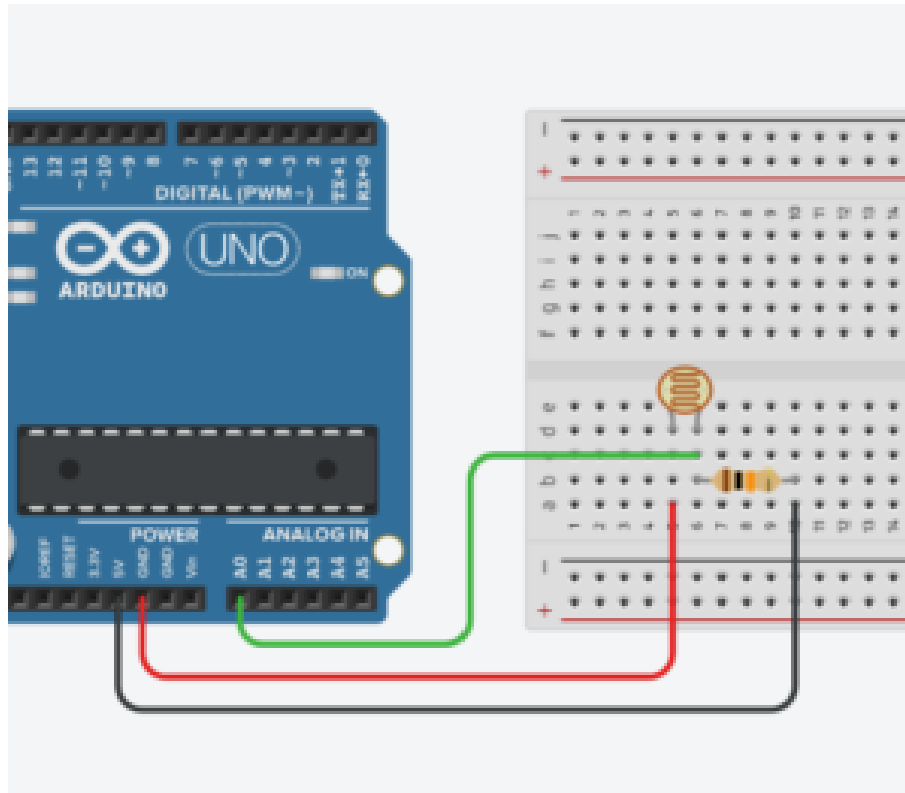


Датчики освещенности (освещения), построенные на базе фоторезисторов, довольно часто используются в реальных ардуино проектах. Фоторезистор ардуино позволяет контролировать уровень освещенности и реагировать на его изменение.

Фоторезистор, как следует из названия, имеет прямое отношение к резисторам, которые часто встречаются практически в любых электронных схемах. Основной характеристикой обычного резистора является величина его сопротивления. От него зависят напряжение и ток, с помощью резистора можно выставить нужные режимы работы других компонентов. Как правило, значение сопротивления у резистора в одних и тех же условиях эксплуатации практически не меняется.

В отличие от обычного резистора, фоторезистор может менять свое сопротивление в зависимости от уровня окружающего освещения. Это означает, что в электронной схеме будут постоянно меняться параметры, в первую очередь нас интересует напряжение, падающее на фоторезисторе. Фиксируя эти изменения напряжения на аналоговых пинах ардуино, мы можем менять логику работы схемы, создавая тем самым адаптирующиеся под внешние условия устройства.

Фоторезисторы достаточно активно применяются в самых разнообразных системах. Самый распространенный вариант применения — фонари уличного освещения. Если на город опускается ночь или стало пасмурно, то огни включаются автоматически. Можно сделать из фоторезистора экономную лампочку для дома, включающуюся не по расписанию, а в зависимости от освещения. На базе датчика освещенности можно сделать даже охранную систему, которая будет срабатывать сразу после того, как закрытый шкаф или сейф открыли и осветили.



Фоторезистор подсоединяется одной ногой к земле, другая подключается к АЦП платы (выше в примере – А0). К этой же ноге подключаем резистор 10 кОм. Естественно, подключать фоторезистор можно не только на аналоговый пин А0, но и на любой другой.

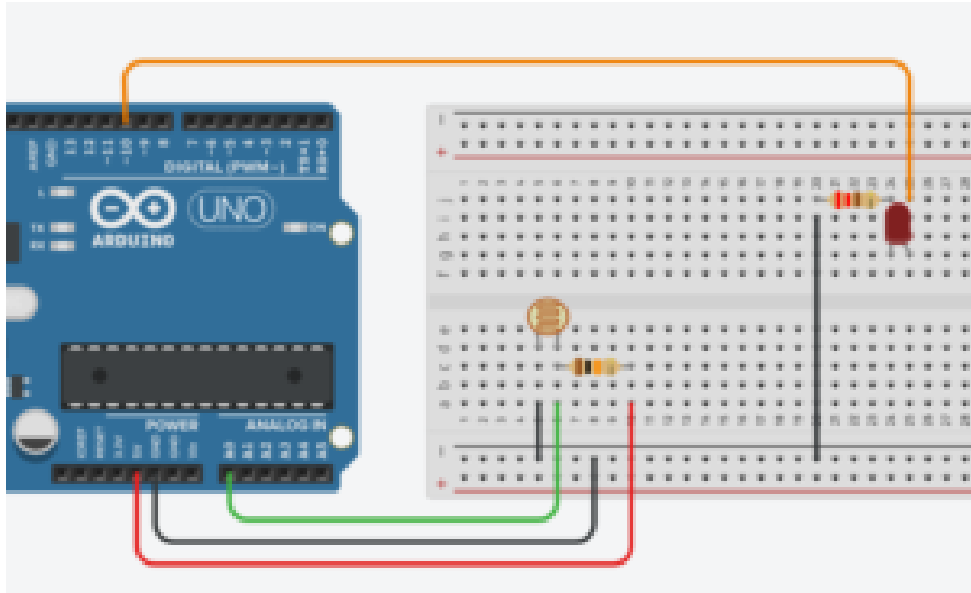
Несколько слов относительно дополнительного резистора на 10 К. У него в нашей схеме две функции: ограничивать ток в цепи и формировать нужное напряжение в схеме с делителем. Ограничение тока нужно в ситуации, когда полностью освещенный фоторезистор резко уменьшает свое сопротивление. А формирование напряжения – для предсказуемых значений на аналоговом порту. На самом деле для нормальной работы с нашими фоторезисторами хватит и сопротивления 1К.

Меня значение резистора мы можем “сдвигать” уровень чувствительности в “темную” и “светлую” сторону. Так, 10 К даст быстрое переключение наступления света. В случае 1К датчик света будет более точно определять высокий уровень освещенности.

Если вы используете готовый модуль датчика света, то подключение будет еще более простым. Соединяем выход модуля VCC с разъемом 5В на плате, GND – с землей. Оставшиеся выводы соединяем с разъемами ардуино.



## Работа совместно с Arduino UNO



Алгоритм работы таков:

- Определяем уровень сигнала с аналогового пина.
- Сравниваем уровень с пороговым значением. Максимально значение будет соответствовать темноте, минимальное – максимальной освещенности. Пороговое значение выберем равное 300.
- Если уровень меньше порогового – темно, нужно включать светодиод.
- Иначе – выключаем светодиод.

```
#define PIN_LED 13
```

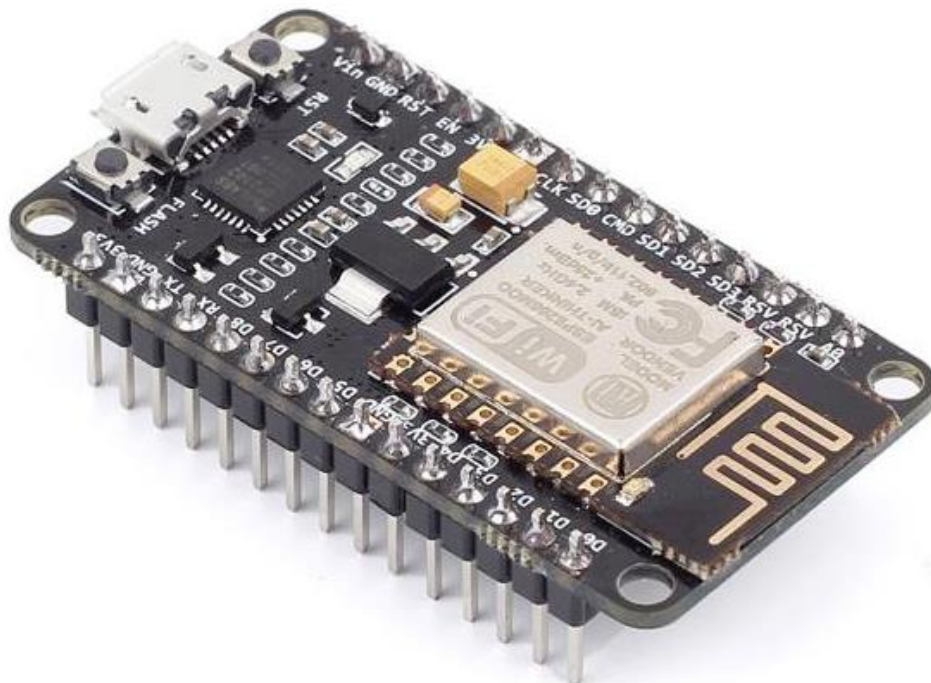
```
#define PIN_PHOTO_SENSOR A0
```

```
void setup() {  
    Serial.begin(9600);  
    pinMode(PIN_LED, OUTPUT);  
}  
  
void loop() {  
    int val = analogRead(PIN_PHOTO_SENSOR);  
    Serial.println(val);  
    if (val < 300) {  
        digitalWrite(PIN_LED, LOW);  
    } else {  
        digitalWrite(PIN_LED, HIGH);  
    }  
}
```

Прикрывая фоторезистор (руками или светонепроницаемым предметом), можем наблюдать включение и выключение светодиода. Изменяя в коде пороговый параметр, можем заставлять включать/выключать лампочку при разном уровне освещения.

## ***ESP8266 NodeMcu v3 Lua с WiFi***

NodeMcu – платформа на основе ESP8266 для создания различных устройств интернета вещей (IoT). Модуль умеет отправлять и получать информацию в локальную сеть либо в интернет при помощи Wi-Fi. Недорогой модуль часто используется для создания систем умного дома или роботов Arduino, управляемых на расстоянии.



Технические характеристики модуля:

- Поддерживает Wi-Fi протокол 802.11 b/g/n;
- Поддерживаемые режимы Wi-Fi – точка доступа, клиент;
- Входное напряжение 3,7В – 20 В;
- Рабочее напряжение 3В-3,6В;
- Максимальный ток 220мА;
- Встроенный стек TCP/IP;
- Диапазон рабочих температур от -40С до 125С;
- 80 МГц, 32-битный процессор;
- Время пробуждения и отправки пакетов 22мс;
- Встроенные TR переключатель и PLL;
- Наличие усилителей мощности, регуляторов, систем управления питанием.

### ***Распиновка NodeMcu v3***

Модуль V3 имеет 11 контактов ввода-вывода общего назначения. Помимо этого некоторые из выводов обладают дополнительными функциями:

- D1-D10 – выходы с широтно-импульсной модуляцией;
- D1, D2 – выходы для интерфейса I<sup>2</sup>C/TWI;
- D5–D8 – выходы для интерфейса SPI;
- D9, D10 – UART;
- A0 – вход с АЦП.



## Датчик пыли GP2Y1010AU0F—SHARP

Датчик пыли GP2Y1010AU0F — предназначен для обнаружения мельчайших частиц пыли в воздухе.

Датчик используется при построении систем очистки воздуха, мониторинга загрязнения воздуха или комплексно с другими системами.

Оптический датчик запылённости воздуха. Принцип действия заключается в том, то светодиод освещает воздух, проходящий через сквозное отверстие. Парящие в воздухе пылинки отражают оптический сигнал на светочувствительный сенсор, а управляющая схема пересчитывает его в количество взвешанных частиц в объёме воздуха. Датчик уверенно регистрирует частицы размером не менее полумикрона.

Приблизительные критерии загрязнённости воздуха микрочастицами пыли:

Концентрация микрочастиц пыли в воздухе	Качество воздуха
<b>0-50</b>	<i>Высокое качество воздуха</i>
<b>51-100</b>	<i>Приемлемое качество воздуха</i>
<b>101-150</b>	<i>Негативное влияние на людей с обострённой восприимчивостью к пыли</i>
<b>151-200</b>	<i>Негативное влияние на все группы людей</i>
<b>201-300</b>	<i>Крайняя степень загрязнённости воздуха пылью</i>

### Характеристики

чувствительность: 0.5В/(100мкг/м3);

измеряемый диапазон: 500мкг/м3;

питание: 2.5-5.5В;

рабочий ток: 20 мА;

рабочая температура: -10~65°C;

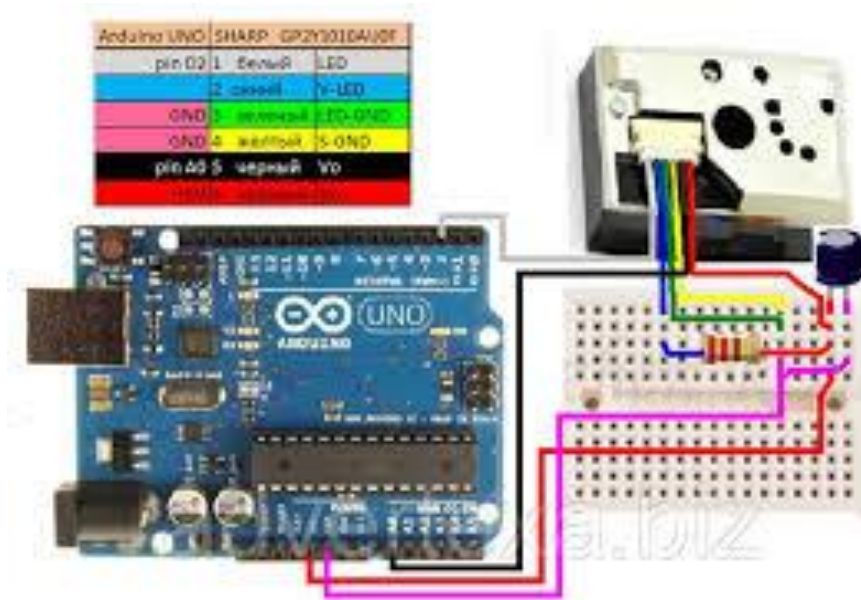
температура хранения: -20~80°C;

габариты: 63.2x41.3x21.1 mm;

### Контакты и сигнал

Вывод датчика GP2Y101AU0F	Цвет провода	Вывод Arduino
V-LED	Белый	5V ( через резистор 150 Ом )
LED-GND	Желтый	GND
LED	Оранжевый	D2 (любой цифровой вывод)
S-GND	Синий	GND
V0	Черный	A0 (любой аналоговый вывод)
VCC	Красный	5V

## Работа совместно с Arduino UNO

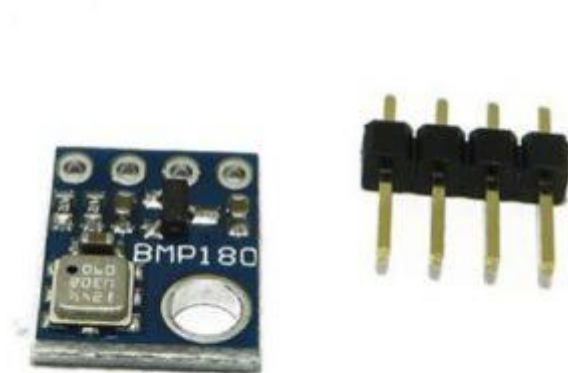


```
int measurePin = A0;
int ledPower = 2;
int samplingTime = 280;
int deltaTime = 40;
int sleepTime = 9680;
float voMeasured;
float calcVoltage;
float dustDensity;

void setup() {
  Serial.begin(9600);
  pinMode(ledPower, OUTPUT);
}

void loop() {
  digitalWrite(ledPower, LOW);
  delayMicroseconds(samplingTime);
  voMeasured = analogRead(measurePin);
  delayMicroseconds(deltaTime);
  digitalWrite(ledPower, HIGH);
  delayMicroseconds(sleepTime);
  calcVoltage = voMeasured * (5.0 / 1024.0);
  dustDensity = 0.17 * calcVoltage - 0.1;
  Serial.print("Raw Signal Value: ");
  Serial.print(voMeasured);
  Serial.print(" --- Voltage: ");
  Serial.print(calcVoltage);
  Serial.print(" --- Dust Density: ");
  Serial.println(dustDensity);
  delay(1000);
}
```

## Датчики давления bmp280



Барометр – устройство, измеряющее атмосферное давление. Электронные барометры используются в робототехнике и различных электронных устройствах. Наиболее распространенными и доступными являются датчики давления от фирмы BOSH: это BMP085, BMP180, BMP280 и другие. Первые два очень похожи между собой, BMP280 – это более новый и усовершенствованный датчик.

Технические характеристики датчика BMP180:

Пределы измеряемого давления 225-825 мм рт. ст.

Напряжение питания 3,3 – 5В;

Ток 0,5мА;

Поддержка интерфейса I2C;

Время срабатывания 4,5мс;

Размеры 15 x 14 мм.

Датчик BMP180 – это дешевый и простой в применении сенсорный датчик, который измеряет атмосферное давление и температуру. Используется обычно для определения высоты и в метеостанциях. Состоит устройство из пьезо-резистивного датчика, термодатчика, АЦП, энергонезависимой памяти, ОЗУ и микроконтроллера.

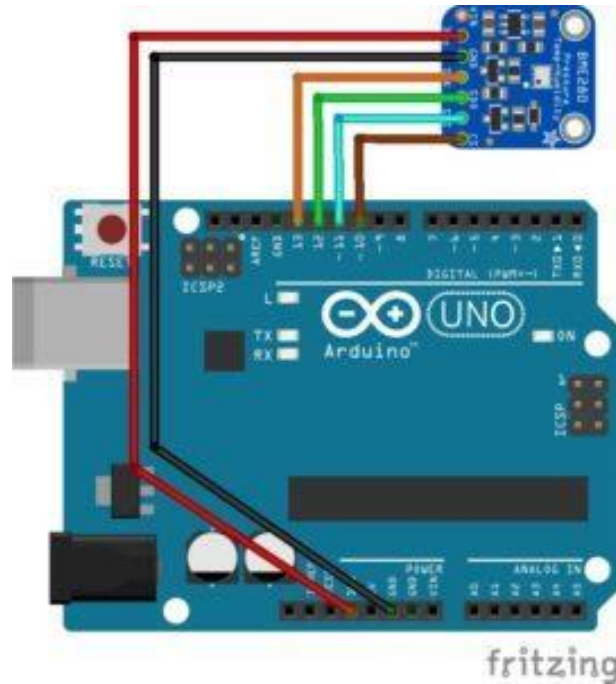
### **Контакты и сигнал**

Для работы с датчиком BMP280 существуют различные библиотеки, упрощающие работу. К ним относятся BMP280\_DEV, Adafruit\_BMP280\_Library. Для датчика BMP280 будет использоваться библиотека от Adafruit.

Arduino Uno/Nano/Pro Mini	BMP280 модуль
GND	GND
5V	Vin
A4	SDA/SDI
A5	SCL/SCK



## Работа совместно с Arduino UNO



```
#include <Adafruit_BMP280.h>
Adafruit_BMP280 bmp280;
void setup() {
  Serial.begin(9600);
  Serial.println(F("BMP280"));
  while (!bmp280.begin(BMP280_ADDRESS - 1)) {
    Serial.println(F("Could not find a valid BMP280 sensor, check wiring!"));
    delay(2000);
  }
}

void loop() {
  float temperature = bmp280.readTemperature();
  float pressure = bmp280.readPressure();
  float altitude = bmp280.readAltitude(1013.25);

  Serial.print(F("Temperature = "));
  Serial.print(temperature);
  Serial.println(" *C");

  Serial.print(F("Pressure = "));
  Serial.print(pressure);
  Serial.println(" Pa");

  Serial.print(F("Altitude = "));
  Serial.print(altitude);
  Serial.println(" m");

  Serial.println();
  delay(2000);
}
```



# Задания к лабораторной работе

## Вводная информация (ко всех темам)

1) Во всех заданиях необходимо разработать программно-аппаратную систему на базе ARDUINO Uno и дополнительной периферии, которая обеспечивает получение информации от определенных датчиков, ее обработку и формирование управляющих воздействий на внешние устройства управления или отображения.

2) Алгоритм функционирования системы, "связывающий" данные, получаемые от датчиков, и управляющие воздействия, которые выдаются на устройства управления/отображения, необходимо разработать и описать самостоятельно.

3) Для взаимодействия платформы ARDUINO с большинством периферийных устройств требуется использование драйверов устройств, которые включены в дополнительные библиотеки (см. сайт дисциплины).

4) Для получения работоспособной аппаратной платформы, правильного подключения некоторых периферийных устройств может потребоваться использование дополнительных радиодеталей, не указанных в задании, но доступных в наборе комплектующих.

## Тема 1. Охрана дома

Нужно защитить дом от несанкционированного доступа. Для этого использовать датчик движения. Но так как датчик движения будет реагировать не только на людей, а еще и на пробегающих мимо животных, то нужно сделать дополнительные проверки. Датчик отпечатков пальцев. И дополнительная проверка для доступа в дом по магнитному ключу или магнитной карточке. Световыми и звуковыми эффектами оповестите хозяев либо о успешном доступе в дом ими самими, либо о незаконном доступе.

Датчики (сенсоры) для съема данных	Устройства управления / отображения
1) Fpm10a отпечатков пальцев 2) HC-SR501 (PIR датчик) 3) RFID МОДУЛЬ RC522 13.56MHZ + КАРТА + БРЕЛОК	1) Светодиоды 2) Пьезоизлучатель 3) Дисплей

## Тема 2. Защита дома от пожара

Нужно защитить дом от несчастных случаев в доме. В доме со время отсутствия хозяев может возникнуть пожар. Поэтому нужно предупредить, что такая ситуация появилась. Но не кому не хочется узнать, что уже его дом горит. Поэтому для начала можно просто сигнализировать о том, что в доме появились газы разных разновидностей. Для этого используйте датчик. А для того чтобы пользователю было известно, какой газ и где появился, используйте монитор для вывода значений и звуковую сигнализацию о уже начинающемся пожаре.

Датчики (сенсоры) для съема данных	Устройства управления / отображения
1) Датчик пламени (огня) 2) Датчик широкого спектра газов MQ-2 3) DHT22 Temperature-Humidity Sensor	1) Светодиоды 2) Пьезоизлучатель 3) Дисплей

### Тема 3. Комфортный дом

Для комфортного проживания в доме, нужно чтобы воздух был чистым, умеренно влажным, теплым, а были растения. Но за этим всегда нужен контроль. Поэтому отслеживайте влажность и температуру воздуха. Освещенность в комнате. Чтобы растения всегда были вовремя политы. Нужно сообщать пользователю очень открыто и явно, если показатели вышли за границы, если нет, просто выводить на дисплей, чтобы пользователь был ознакомлен с ситуацией. Если света в комнате становится мало, то включить свет.

Датчики (сенсоры) для съема данных	Устройства управления / отображения
1) DHT22 Temperature-Humidity Sensor 2) Датчик влажности почвы 3) Фоторезистор	1) TFT touch LCD Экран (сенсорный дисплей) 2) Светодиоды 3) Пьезоизлучатель

### Тема 4. Управление предметами по будильнику

Хочется проснуться утром, а на кухне тебя уже ожидает свежесваренный кофе. Но мы пойдем пока не много меньше. Автоматизируем рутинную работу. Вам нужно в определенное время на улице включать свет и так же выключать. Если вы легли спать и забыли поставить офис на сигнализацию, включите датчик движения в нужное время. Ну и конечно пользователь должен видеть и регулировать время выставления будильников. Сделайте две кнопочки выставления времени для будильника (одна изменяет часы, другая изменяет минуты). А так же если на улице становится темно, включить во время свет в доме или на улице.

Датчики (сенсоры) для съема данных	Устройства управления / отображения
1) Часы реального времени DS3231 2) Тактовые кнопки 3) Фоторезистор	1) Светодиоды 2) TFT touch LCD Экран (сенсорный дисплей) 3) Пьезоизлучатель

### Тема 5. Защита от затопления дома

Трубы имеют свойство ржаветь, а стиральная машинка имеет свойство ломаться. И хорошо, если Вы будете в этот момент дома, а если нет, придется восстанавливать не только прохудившиеся трубы и стиральную машинку, но ещё и возмещать ремонт соседям и себе. Поэтому для контроля таких ситуаций будем использовать датчик влажности, который сообщит нам, о том, что где-то произошла протечка воды. А если Вы любите купаться в полные воды ванной, но при этом легко забываете вовремя выключить воду или Вас в этот момент могут отвлечь, то будем использовать датчик определения уровня воды. И конечно нужно сразу сигнализировать о произошедшей ситуации.

Датчики (сенсоры) для съема данных	Устройства управления / отображения
1) Датчик уровня воды (угловой) 2) МОДУЛЬ ДАТЧИКА ПРОТЕЧКИ FC-37	1) Светодиоды 2) TFT touch LCD Экран (сенсорный дисплей) 3) Пьезоизлучатель

## Тема 6. Комфортный дом для родных

Хорошо когда родные живут близко и есть возможность всегда поддерживать все показатели дома для комфортного и безопасного проживания. А что же делать, если хочется всегда знать состояние дома, но возможности нет!? Тогда на помощь придет система позволяющая отслеживать сложные показатели и сообщать о необходимости дополнительных действий. Система позволит измерять атмосферное давление и температуру в помещении, позволит контролировать влажность и при необходимости оповещать пользователей о важности проветривания комнаты. А также не мало важное измерять уровень пыли в здании. Ведь это так важно, особенно если в помещении маленькие дети.

Датчики (сенсоры) для съема данных	Устройства управления / отображения
1) Датчик BMP180	1) Светодиоды
2) Датчик пыли	2) Пьезоизлучатель
3) Датчик HDC1080	3) Тактовые кнопки

### *Дополнительные ссылки по датчикам*

Сенсор датчик	<a href="https://arduino-kit.ru/product/datchik-kasaniya">https://arduino-kit.ru/product/datchik-kasaniya</a>
HC-SR501 (PIR датчик)	<a href="https://voltiq.ru/arduino-and-sensor-hc-sr501/">https://voltiq.ru/arduino-and-sensor-hc-sr501/</a>
Герметичный датчик температуры DS18B20	<a href="http://wiki.amperka.ru/%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82%D1%8B:ds18b20">http://wiki.amperka.ru/%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82%D1%8B:ds18b20</a>
Датчик влажности почвы	<a href="https://3d-diy.ru/wiki/arduino-datchiki/datchik-vlazhnosti-pochvy-arduino/">https://3d-diy.ru/wiki/arduino-datchiki/datchik-vlazhnosti-pochvy-arduino/</a>
TFT touch LCD Экран (сенсорный дисплей) +microsd для Arduino	<a href="http://arduino-diy.com/arduino-TFT-LCD-displey-2-i-8-dyuyma">http://arduino-diy.com/arduino-TFT-LCD-displey-2-i-8-dyuyma</a>
Электретный микрофон ECM-30A, 4.5 В, 10 мм	<a href="https://soltau.ru/index.php/arduino/item/377-kak-podklyuchit-datchik-zvuka-mikrofon-k-arduino">https://soltau.ru/index.php/arduino/item/377-kak-podklyuchit-datchik-zvuka-mikrofon-k-arduino</a>
Passive buzzer	<a href="https://www.youtube.com/watch?v=YGCbKWZMcTc">https://www.youtube.com/watch?v=YGCbKWZMcTc</a>
Модуль датчика протечки fc-37 для arduino	<a href="https://arduinomaster.ru/datchiki-arduino/datchik-protechki-i-dozhdy-a-v-arduino-opisanie-shemy-sketchi/">https://arduinomaster.ru/datchiki-arduino/datchik-protechki-i-dozhdy-a-v-arduino-opisanie-shemy-sketchi/</a>
HC-SR04	<a href="http://robotclass.ru/tutorials/arduino-sonic-hc-sr04/">http://robotclass.ru/tutorials/arduino-sonic-hc-sr04/</a>
Dht22 temperature-humidity sensor	<a href="https://www.hackster.io/mafzal/temperature-monitoring-with-dht22-arduino-15b013">https://www.hackster.io/mafzal/temperature-monitoring-with-dht22-arduino-15b013</a>
Датчик уровня воды (угловой)	<a href="http://wiki.amperka.ru/projects:barrel">http://wiki.amperka.ru/projects:barrel</a>
Датчик пламени (огня)	<a href="http://robotclass.ru/tutorials/arduino-flame-sensor/">http://robotclass.ru/tutorials/arduino-flame-sensor/</a>
Датчик широкого спектра газов MQ-2 (Тройка-модуль)	<a href="http://wiki.amperka.ru/%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82%D1%8B:mq2">http://wiki.amperka.ru/%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82%D1%8B:mq2</a>
Nodemcu Lua Wi-Fi На Esp8266 для Arduino	<a href="https://arduinomaster.ru/platy-arduino/esp8266-nodemcu-v3-lua/">https://arduinomaster.ru/platy-arduino/esp8266-nodemcu-v3-lua/</a>
Rfid модуль rc522 13.56mhz + карта + брелок	<a href="https://soltau.ru/index.php/arduino/item/399-kak-podklyuchit-rfid-schityvatel-rc522-k-arduino">https://soltau.ru/index.php/arduino/item/399-kak-podklyuchit-rfid-schityvatel-rc522-k-arduino</a>
Часы реального времени DS3231	<a href="http://edurobots.ru/2017/05/arduino-real-time-clock-ds3231/">http://edurobots.ru/2017/05/arduino-real-time-clock-ds3231/</a>
Датчик освещённости (Тройка-модуль)	<a href="http://wiki.amperka.ru/%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82%D1%8B:troyka-light-sensor">http://wiki.amperka.ru/%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82%D1%8B:troyka-light-sensor</a>

Датчик влажности и температуры (HDC1080)	<a href="http://arduinolearning.com/code/arduino-uno-hdc1080-humidity-temperature-sensor.php">http://arduinolearning.com/code/arduino-uno-hdc1080-humidity-temperature-sensor.php</a>
Датчик пыли	<a href="https://arduinoplus.ru/datchik-pyli-arduino/">https://arduinoplus.ru/datchik-pyli-arduino/</a>
Датчик BMP180	<a href="https://micro-pi.ru/подключение-bmp280-к-arduino/">https://micro-pi.ru/ подключение-bmp280-к-arduino/</a>