

Работа с датчиками

- **Датчик** — конструктивно обособленное устройство предназначенное для преобразования входной измеряемой величины (обычно **неэлектрической**) в выходную (обычно **электрическую**) удобную для дальнейшего использования.
- # **датчик температуры** (преобразует температуру в электрическое напряжение, пропорциональное температуре); **датчик освещенности** (преобразует световой поток в величину напряжения или ЭДС и т.д.)

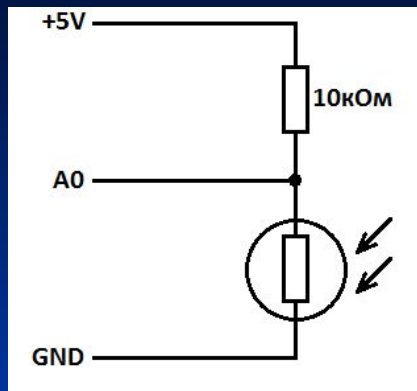
Устройство датчиков

- В основе любого датчика лежит **первичный измерительный преобразователь**, который непосредственно преобразуют входную величину в **электрический сигнал**.
- Преобразователи могут быть 1) **генераторные** — преобразуют входную величину в **ЭДС**, т.е. являются **источниками энергии**: термопара, фотодиод, тахогенератор или 2) **параметрические** — преобразуют входную величину в изменение электрического параметра: терморезистор, емкостный преобразователь.
- Для работы датчиков на основе **параметрических преобразователей** обязательно требуется **источник питания и измерительная цепь**.

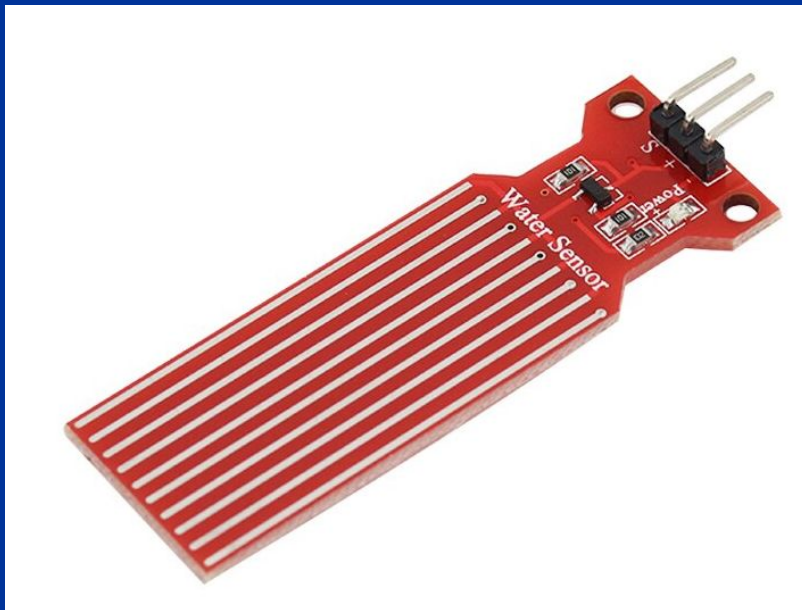
Разновидности датчиков

- Аналоговые датчики – формируют на выходе непрерывное напряжение пропорциональное измеряемой величине.
- Могут быть изготовлены самостоятельно на базе генераторных или параметрических преобразователей. Существуют также промышленно изготовленные датчики для *Arduino*.
- При работе таких датчиков с *Arduino* необходимо использовать аналоговые входы с учетом согласования уровней напряжения.
- Самостоятельно изготовленные датчики требуют понимания принципов работы преобразователей и схемотехники измерительной цепи.

■ Примеры аналоговых датчиков:



Датчик освещенности на фоторезисторе (параметрический преобразователь) самостоятельного изготовления.



Датчик уровня жидкости для Arduino промышленного изготовления.

- **Цифровые датчики** — датчики, имеющие в своем составе помимо **первичных преобразователей** еще и **АЦП**, передающие данные в цифровом коде **определенного формата**.
- Как правило это датчики в **интегральном исполнении** (в виде микросхем) или в виде готовых **печатных плат**.
- Обычно не требуют специальных знаний при использовании кроме **схемы подключения**.
- Взаимодействие с **Arduino** происходит через **цифровые порты**. Для удобства программирования используются специальные подключаемые **библиотеки**.

■ Примеры цифровых датчиков для Arduino:

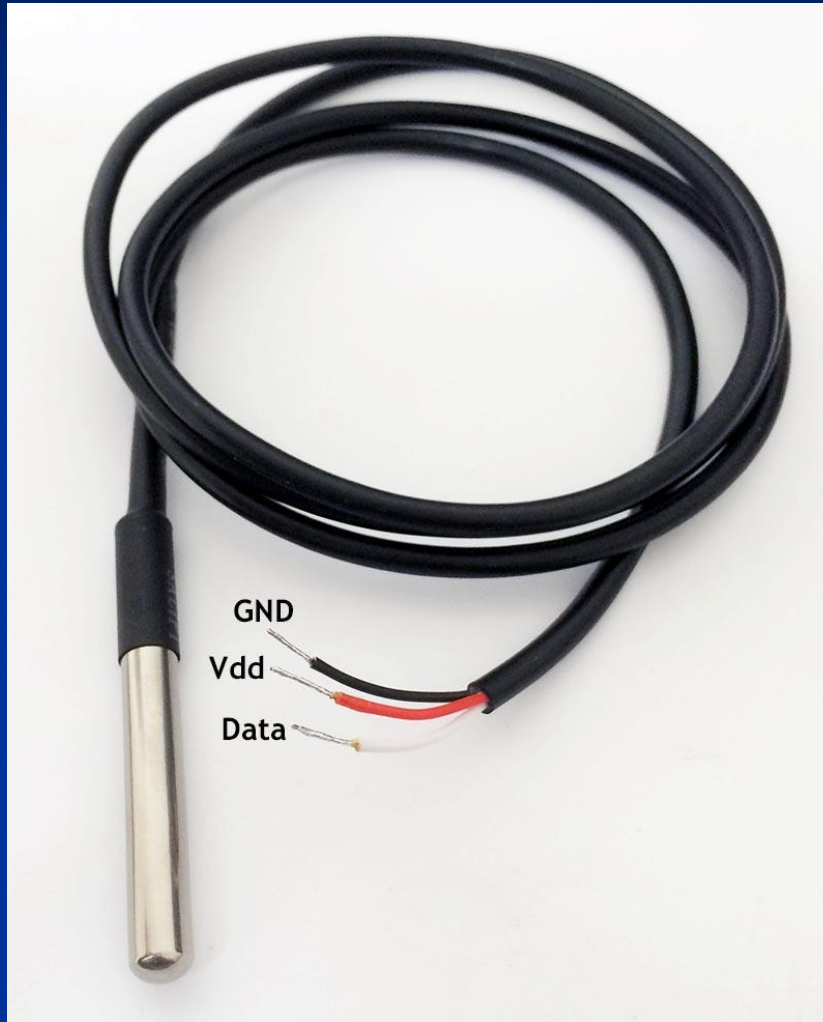


Цифровые датчики температуры
DS18B20



Ультразвуковой датчик
расстояния HC-SR04

Работа с датчиком температуры DS18B20



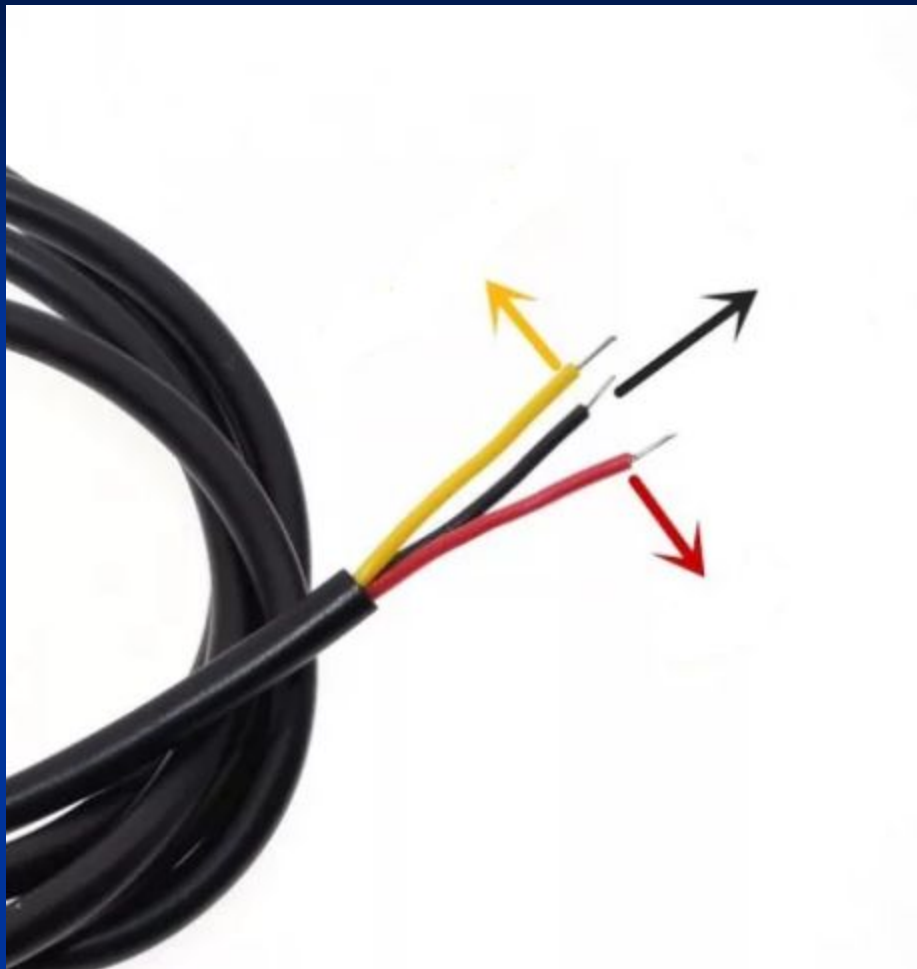
Диапазон
температур:

-55 до + 125 C°

Диапазон
питающих
напряжений:

3,0 В до 5,5 В

■ Назначение выводов:

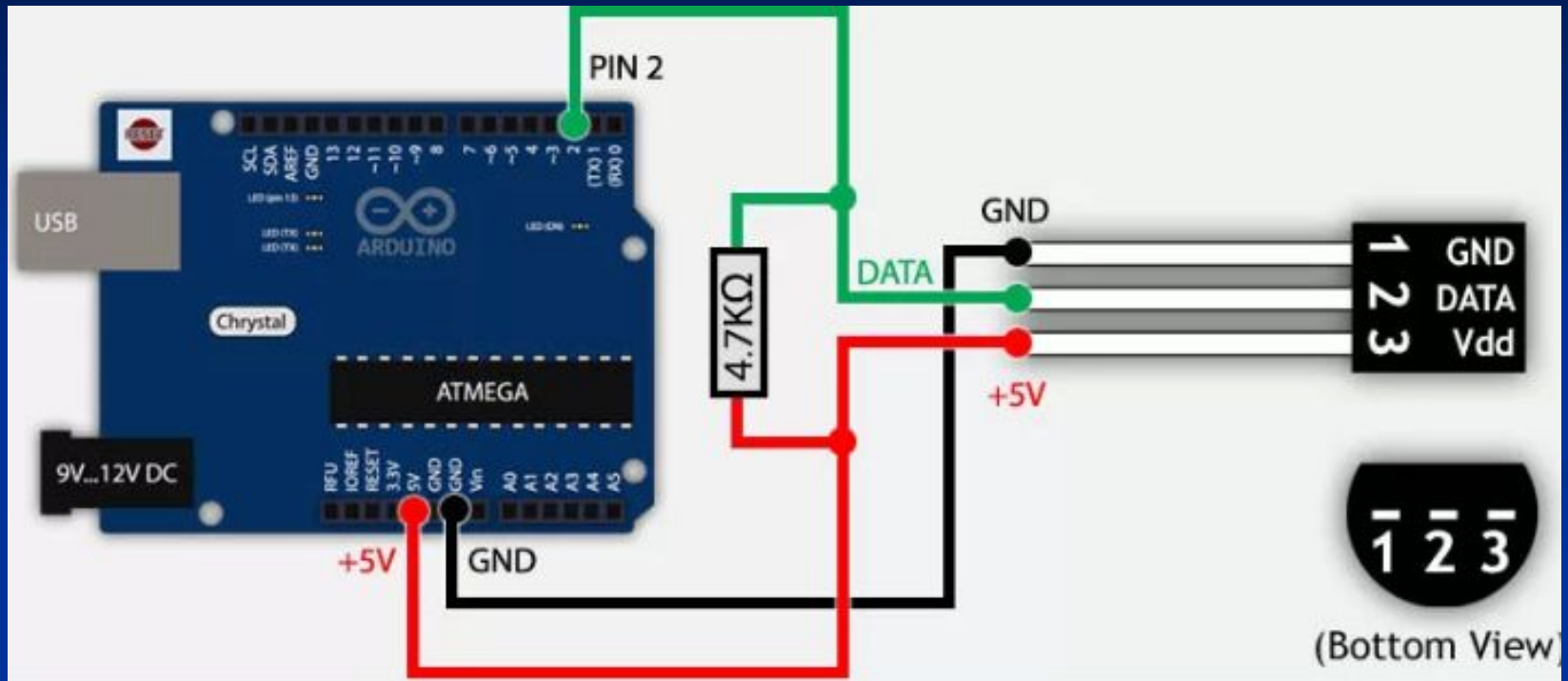


Черный - GND (общий)

Красный + 5V (питание)

Другие цвета - сигнал

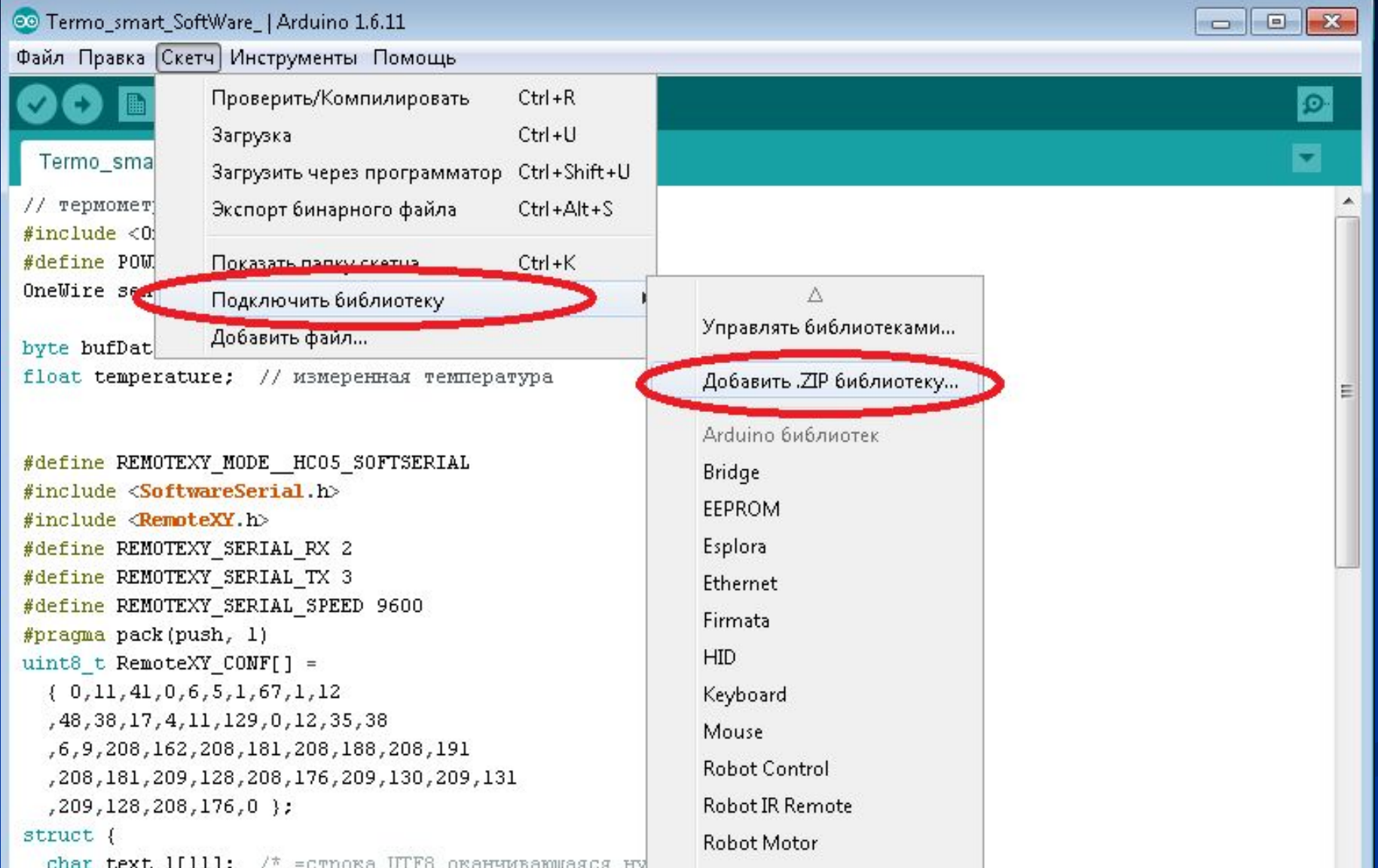
■ Схема подключения:



* Неправильное подключение датчика может вывести его из строя!

■ Считывание данных:

* Для удобства работы с датчиком понадобятся дополнительные библиотеки **OneWire** и **DallasTemperature**. Их следует предварительно скачать и добавить в среду разработки: меню **Скетч** – **Подключить библиотеку** – **добавить ZIP библиотеку**.



Пример простого вывода температуры с датчика в терминал

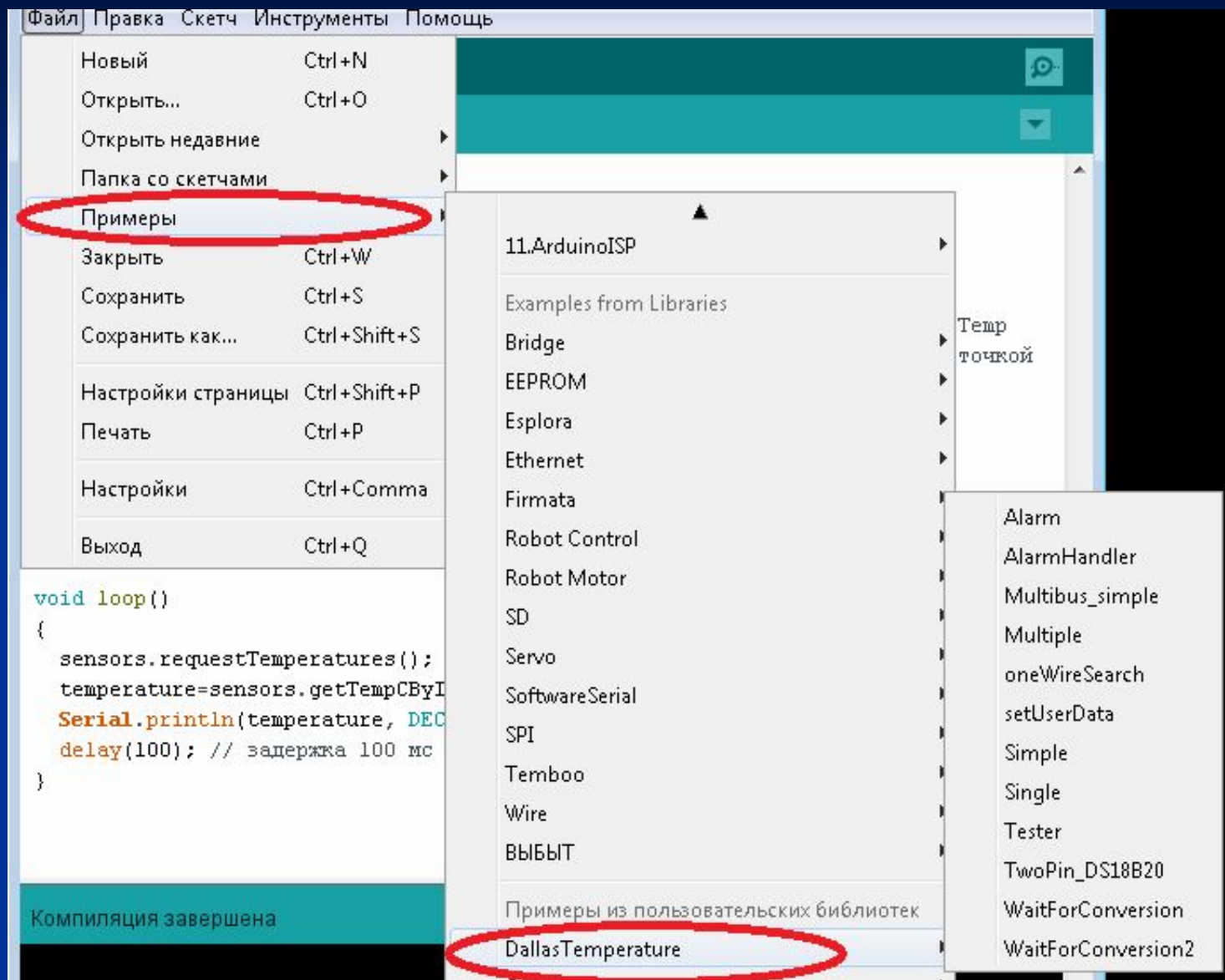
```
// термометр, датчик DS18B20
#include <OneWire.h> // подключение библиотеки протокола One Wire
#include <DallasTemperature.h> // подключение библиотеки dallasTemperature

OneWire sensDs (10); // определение входа к которому подключен датчик
DallasTemperature sensors(&sensDs); // передача ссылки в библиотеку DallasTemp
float temperature; // переменная измеренной температуры, тип с плавающей точкой

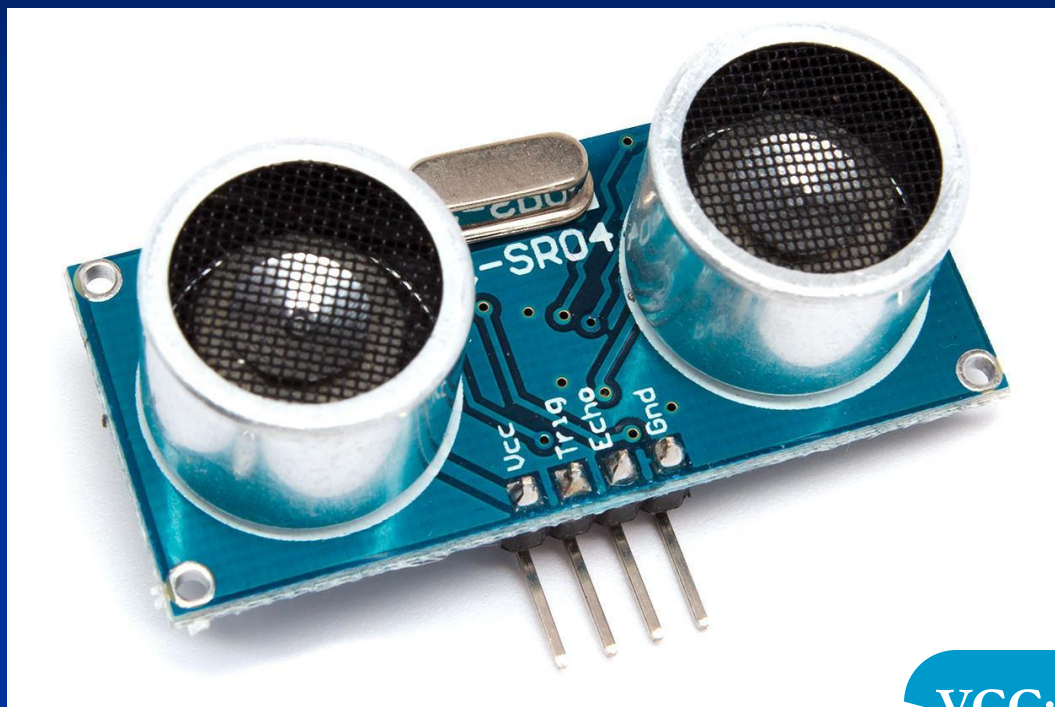
void setup()
{
  sensors.begin(); // запуск библиотеки
  Serial.begin(9600); // инициализация порта для вывода информации
}

void loop()
{
  sensors.requestTemperatures(); // команда запроса температуры
  temperature=sensors.getTempCByIndex(0); // сохранение температуры в переменную
  Serial.println(temperature, DEC); // вывод температуры в терминал
  delay(100); // задержка 100 мс
}
```

* Для более подробного изучения возможностей библиотек, можно обратиться к примерам, поставляемым вместе с библиотекой.



Работа с ультразвуковым датчиком расстояния **HC-SR04**



Напряжение питания: **+5В**;
Эффективный рабочий
угол: **< 15°**;
Расстояние измерений: от **2**
см до **400 см**;
Разрешающая
способность: **0.3 см**;

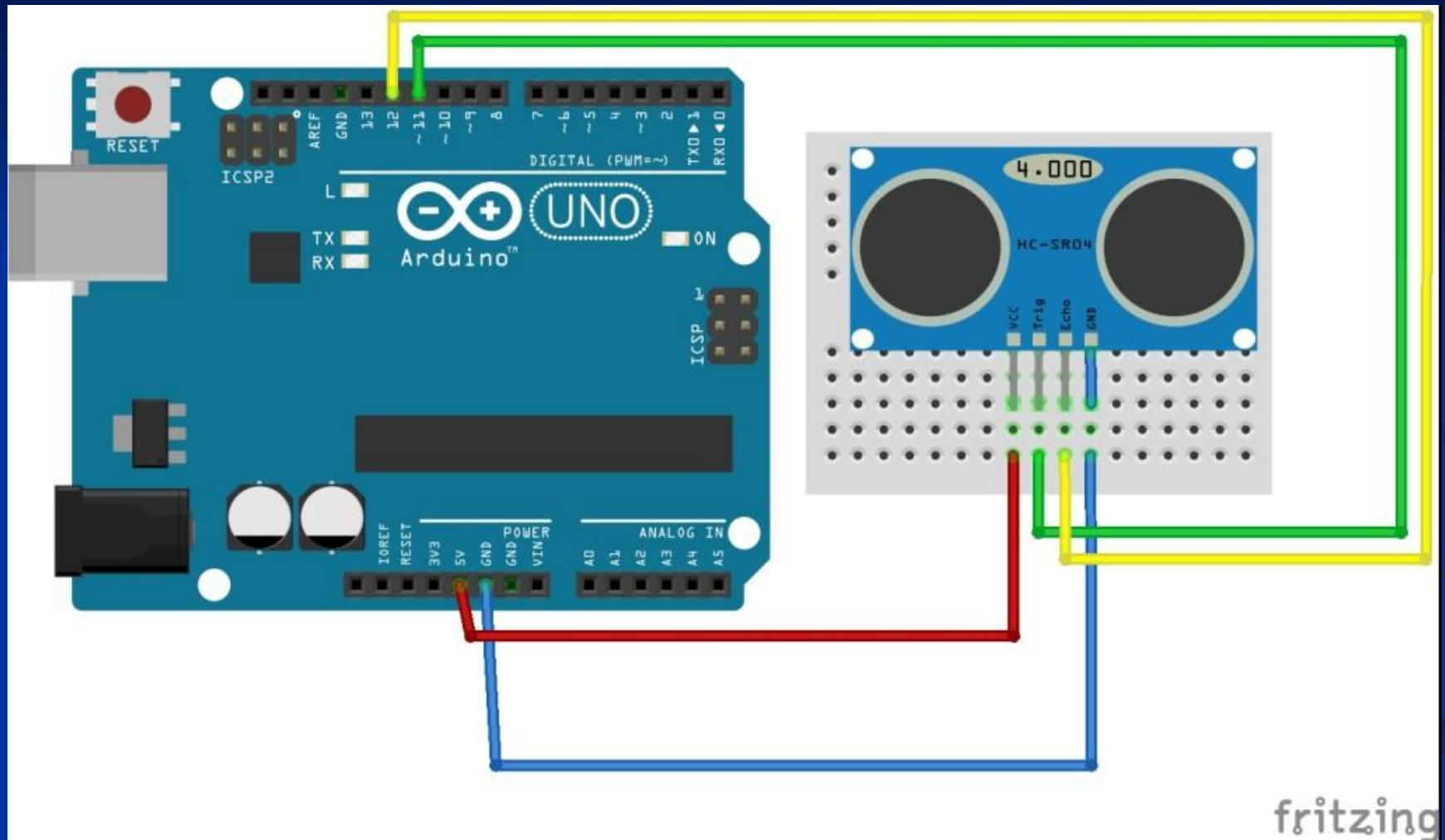
VCC: **+5 вольт (постоянный ток)**

Trig : **Триггер (INPUT)**

Echo: **Эхо (OUTPUT)**

GND: **Земля**

■ Схема подключения датчика



■ Вывод расстояния с датчика

```
#include <NewPing.h>

#define TRIGGER_PIN 12 // Arduino pin tied to trigger pin on the ultrasonic sensor.
#define ECHO_PIN 11 // Arduino pin tied to echo pin on the ultrasonic sensor.
#define MAX_DISTANCE 200 // Maximum distance we want to ping for (in centimeters). Maximum sensor

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup of pins and maximum distance.

void setup() {
  Serial.begin(9600); // Open serial monitor at 115200 baud to see ping results.
}

void loop() {
  delay(50); // Wait 50ms between pings (about 20 pings/sec). 29ms should be t
  Serial.print("Ping: ");
  Serial.print(sonar.ping_cm()); // Send ping, get distance in cm and print result (0 = outside se
  Serial.println("cm");
}
```


Задание 1

- 1) Подключить и запрограммировать датчик температуры на вывод текущей температуры в терминал.
- 2) Подключить и запрограммировать датчик расстояния на вывод данных в терминал.

Задание 2

- Создайте автоматическое устройство, которое бы **включало** светодиод **при превышении** определенной температуры или расстояния.