

Датчик температуры

DHT11 — это популярный цифровой датчик, который позволяет измерять сразу два параметра: температуру и влажность воздуха. Датчик прост в использовании и идеально подходит для проектов, связанных с мониторингом микроклимата.



Схема подключения DHT11 к Arduino:

Arduino	DHT11
5V	VCC (+)
GND	GND (-)
Pin 2 (цифровой)	DATA (сигнал)



Какие библиотеки

DHT Sensor Library от Adafruit Industries — поддерживает DHT11



Как установить

Вот шаги установки в Arduino IDE:

1. Откройте Arduino IDE.
2. В меню: **Sketch (Скетч)** → **Include Library (Включить библиотеку)** → **Manage Libraries...** (Управление библиотеками...).
3. В окне Library Manager в поле поиска введите, например: **“DHT sensor library”**.
 - Найдите библиотеку «DHT Sensor Library» от Adafruit. Установите её.
4. После установки библиотек перезапустите Arduino IDE, чтобы всё корректно отобразилось.

```
#include "DHT.h"

#define DHTPIN 2           // Пин, к которому подключён датчик
#define DHTTYPE DHT11     // Тип датчика: DHT11

DHT dht(DHTPIN, DHTTYPE);

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

void loop() {
```

```

// Считываем значения с датчика
float h = dht.readHumidity();
float t = dht.readTemperature();

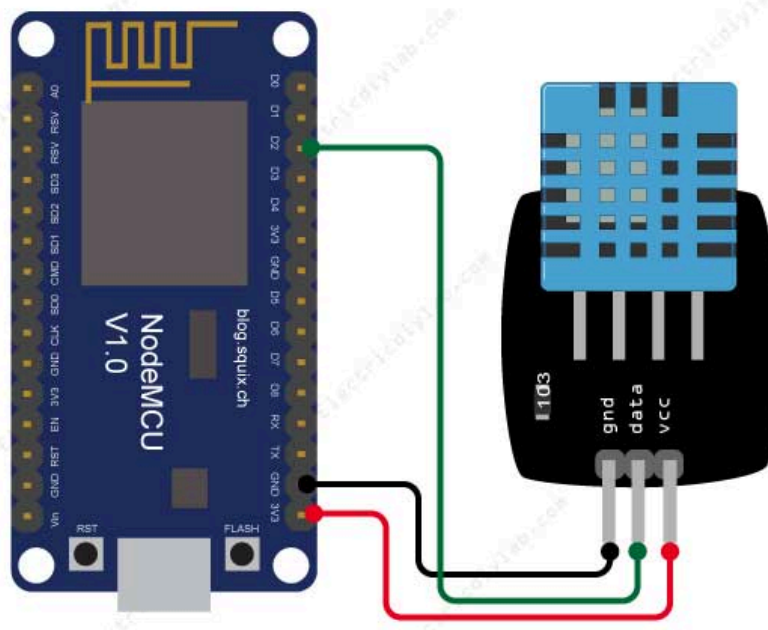
// Проверка, успешно ли считаны данные
if (isnan(h) || isnan(t)) {
    Serial.println("Ошибка чтения с датчика!");
    return;
}

// Выводим данные на монитор порта
Serial.print("Температура: ");
Serial.print(t);
Serial.print(" °C | ");

Serial.print("Влажность: ");
Serial.print(h);
Serial.println(" %");

delay(2000); // пауза 2 секунды между измерениями
}

```



DHT	ESP8266
VCC	- 5V
GND	- GND
DATA	- D2

ESP Easy

После того как физически подключили, переходим к настройке в прошивке ESP Easy:

1. Веб-интерфейс ESP Easy: зайдите на IP вашего модуля (после подключения к WiFi/точке доступа)
2. В меню **Devices** (Устройства) — добавьте новое устройство.
3. В списке устройств найдите «DHT Humidity Sensor» (или аналог, зависит от версии).
4. Укажите:
 - Тип: DHT11 (или DHT22, если используете)
 - Пин: GPIO к которому подключена линия DATA
5. Сохраните и перезапустите модуль при необходимости.
6. После запуска — в разделе **Tools** → **Logs** (или аналог) проверьте, что датчик читается корректно: появятся значения температуры и влажности.

Датчик расстояния

HC-SR04 — ультразвуковой датчик, который широко используется для измерения расстояния до объектов с помощью ультразвука. Он отправляет короткий ультразвуковой импульс, отражающийся от препятствия и возвращающийся обратно. По времени возврата сигнала рассчитывается расстояние до объекта.

 Схема подключения HC-SR04 к Arduino:

Подключите датчик следующим образом

Arduino	HC-SR04
5V	VCC
GND	GND
Pin 9	Trig
Pin 8	Echo

дзHC-SR04 — ультразвуковой датчик, который широко используется для измерения расстояния до объектов с помощью ультразвука. Он отправляет короткий ультразвуковой импульс, отражающийся от препятствия и возвращающийся обратно. По времени возврата сигнала рассчитывается расстояние до объекта.д

 Схема подключения HC-SR04 к Arduino:

Подключите датчик следующим образом

Arduino	HC-SR04
5V	VCC
GND	GND
Pin 9	Trig
Pin 8	Echo

```
nMode(echoPin, INPUT);
}

void loop() {
    long duration, cm;

    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
```

```
digitalWrite(trigPin, HIGH);  
delayMicroseconds(10);  
digitalWrite(trigPin, LOW)  
  
duration = pulseIn(echoPin, HIGH, 30000);  
  
cm = duration / 58;  
  
Serial.print("Расстояние: ");
```

```
const int trigPin = 9;  
const int echoPin = 8;  
  
void setup() {  
  Serial.begin(9600);  
  pinMode(trigPin, OUTPUT);  
  pinMode(echoPin, INPUT);  
}
```

```
void loop() {  
  long duration, cm;  
  
  digitalWrite(trigPin, LOW);  
  delayMicroseconds(2);  
  digitalWrite(trigPin, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(trigPin, LOW);  
  
  duration = pulseIn(echoPin, HIGH, 30000);  
  
  cm = duration / 58;
```

```
Serial.print("Расстояние: ");
Serial.print(cm);
Serial.println(" cm");

delay(500);
}

const int trigPin = 9;
const int echoPin = 8;

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

void loop() {
  long duration, cm;

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH, 30000);

  cm = duration / 58;

  Serial.print("Расстояние: ");
  Serial.print(cm);
  Serial.println(" cm");

  delay(500);
}
```



```
const int trigPin = 9;
const int echoPin = 8;

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

void loop() {
  long duration, cm;

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH, 30000);

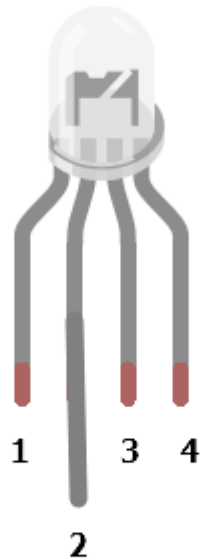
  cm = duration / 58;

  Serial.print("Расстояние: ");
  Serial.print(cm);
  Serial.println(" cm");

  delay(500);
}
```

RGB

COMMON ANODE RGB LED



- 1 - Red
- 2 - Common Anode (+)
- 3 - Green
- 4 - Blue

плавная радуга 🌈

🎯 Что делает скетч:

- Постепенно меняет цвет от красного → зелёного → синего → обратно
- Создаёт эффект "бегающего" цвета по кругу

// Пины для RGB-светодиода

```
const int redPin = 9;
```

```
const int greenPin = 10;
```

```
const int bluePin = 11;
```

```
void setup() {
```

```
  pinMode(redPin, OUTPUT);
```

```
  pinMode(greenPin, OUTPUT);
```

```
  pinMode(bluePin, OUTPUT);
```

```
}
```

```
void loop() {
```

```
  // Плавно: красный -> зелёный
```

```
  for (int i = 0; i < 256; i++) {
```

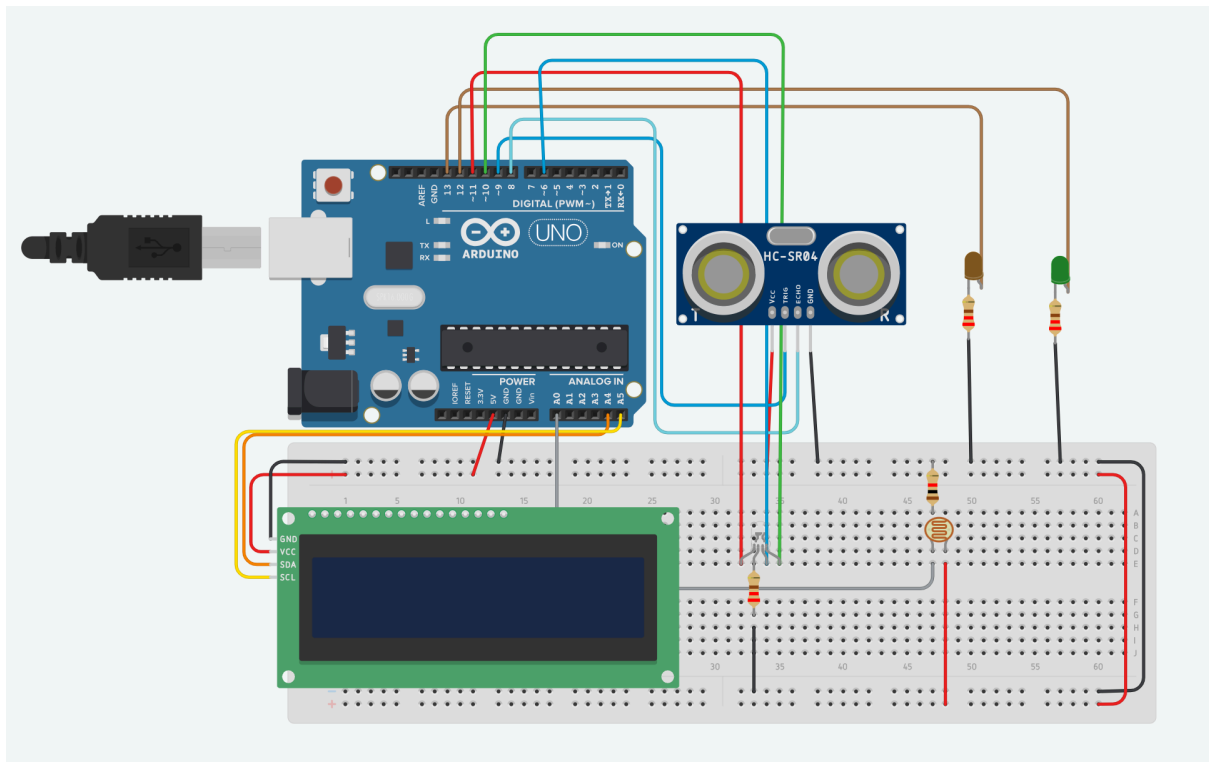
```
    setColor(255 - i, i, 0);  
    delay(10);  
}
```

```
// Плавно: зелёный -> синий  
for (int i = 0; i < 256; i++) {  
    setColor(0, 255 - i, i);  
    delay(10);  
}
```

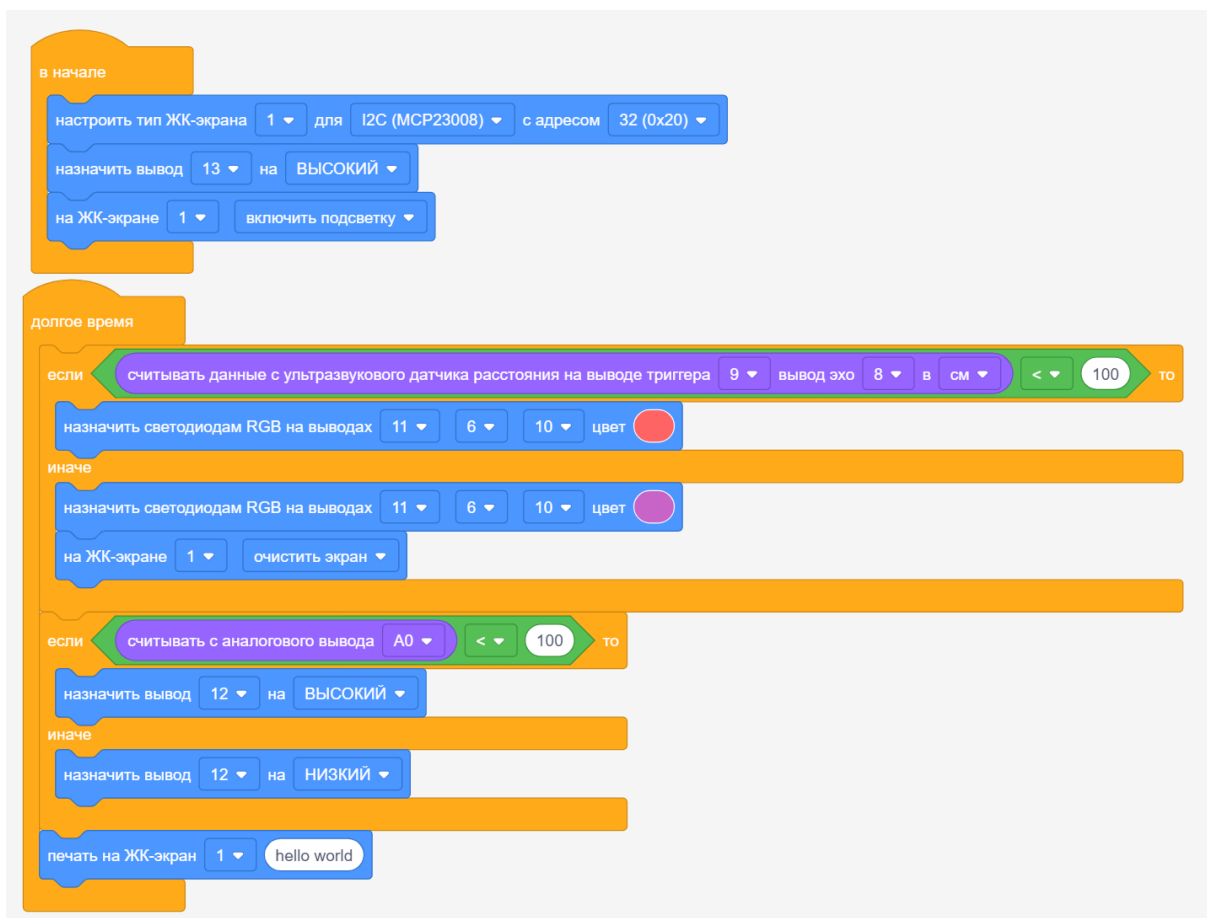
```
// Плавно: синий -> красный  
for (int i = 0; i < 256; i++) {  
    setColor(i, 0, 255 - i);  
    delay(10);  
}  
}
```

```
// Установка цвета  
void setColor(int red, int green, int blue) {  
    analogWrite(redPin, red);  
    analogWrite(greenPin, green);  
    analogWrite(bluePin, blue);  
}
```

Схема ардуино целиком



https://www.tinkercad.com/things/1gE0sWUzJ5M-neat-blad-blorr?sharecode=kfzitURdTBC24_Dm2sqZqKPfIEQFKbmtc7X2EpqI-4Y



Протестируем дома!!

Программа 1: Светодиоды горят, дисплей "Welcome", расстояние в монитор порта

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Настройка дисплея (адрес может быть 0x27 или 0x3F)
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Пины светодиодов
const int led1 = 13;
const int led2 = 12;
const int led3 = 11;

// Пины ультразвукового датчика
const int trigPin = 9;
const int echoPin = 8;

void setup() {
  // Инициализация светодиодов
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);

  // Включаем светодиоды
  digitalWrite(led1, HIGH);
  digitalWrite(led2, HIGH);
  digitalWrite(led3, HIGH);

  // Инициализация датчика расстояния
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  // Инициализация порта для вывода
  Serial.begin(9600);

  // Инициализация дисплея
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Welcome");
}

void loop() {
```



```

long duration;
float distance;

// Отправка ультразвукового импульса
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

// Измерение времени отклика
duration = pulseIn(echoPin, HIGH);

// Вычисление расстояния (в сантиметрах)
distance = duration * 0.034 / 2;

// Вывод расстояния в монитор порта
Serial.print("Distance: ");
Serial.print(distance);
Serial.println(" cm");

delay(500); // Пауза между измерениями
}

```

Программа 2: Светодиоды мигают при приближении + Danger

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Настройка дисплея
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Пины светодиодов
const int led1 = 13;
const int led2 = 12;
const int led3 = 11;

// Пины ультразвукового датчика
const int trigPin = 9;
const int echoPin = 8;

// Пороговое расстояние для срабатывания (в сантиметрах)
const float dangerDistance = 20.0;

```

```

void setup() {
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  Serial.begin(9600);

  lcd.init();
  lcd.backlight();
}

void loop() {
  long duration;
  float distance;

  // Отправка ультразвукового импульса
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);

  distance = duration * 0.034 / 2;

  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (distance < dangerDistance) {
    // Мигаем светодиодами
    digitalWrite(led1, HIGH);
    digitalWrite(led2, HIGH);
    digitalWrite(led3, HIGH);
    delay(300);
    digitalWrite(led1, LOW);
    digitalWrite(led2, LOW);
    digitalWrite(led3, LOW);
    delay(300);

    // Надпись Danger
    lcd.clear();
    lcd.setCursor(0, 0);
  }
}

```

```
    lcd.print("Danger!");  
} else {  
    // Светодиоды постоянно горят  
    digitalWrite(led1, HIGH);  
    digitalWrite(led2, HIGH);  
    digitalWrite(led3, HIGH);  
  
    // Надпись Welcome  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Welcome");  
    delay(500);  
}  
}
```