

# Цифровой термометр DS18B20 (Трема-модуль v2.0)



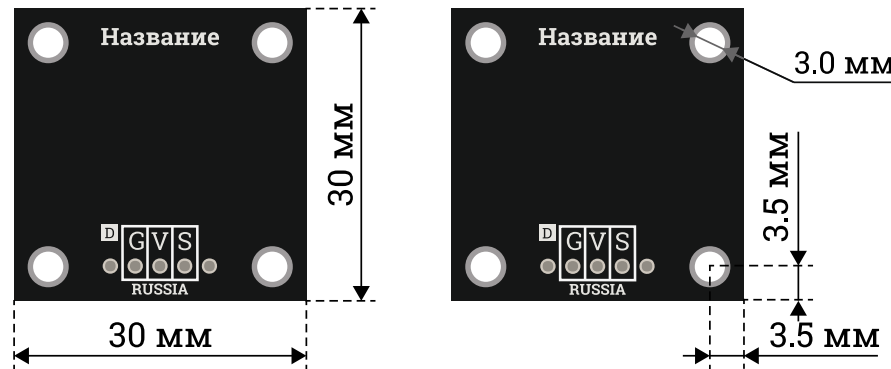
## Общие сведения:

[Трема-модуль цифровой термометр DS18B20](#) - позволит измерять температуру окружающей среды. Исполнен в линейке Трема-модулей, благодаря этому вам не придется ничего паять и собирать на макетной плате.

## Спецификация:

- Диапазон измеряемой температуры:  $-55 \dots 125$  °C
- Напряжение питания: от 3 В до 5,5 В
- Точность:  $\pm 0,5$ °C (в пределах  $-10 \dots 85$  °C)
- Потребляемый ток при опросе: 1 мА
- Габариты: 31×31 мм

Все модули линейки "Трема" выполнены в одном формате



## Подключение:

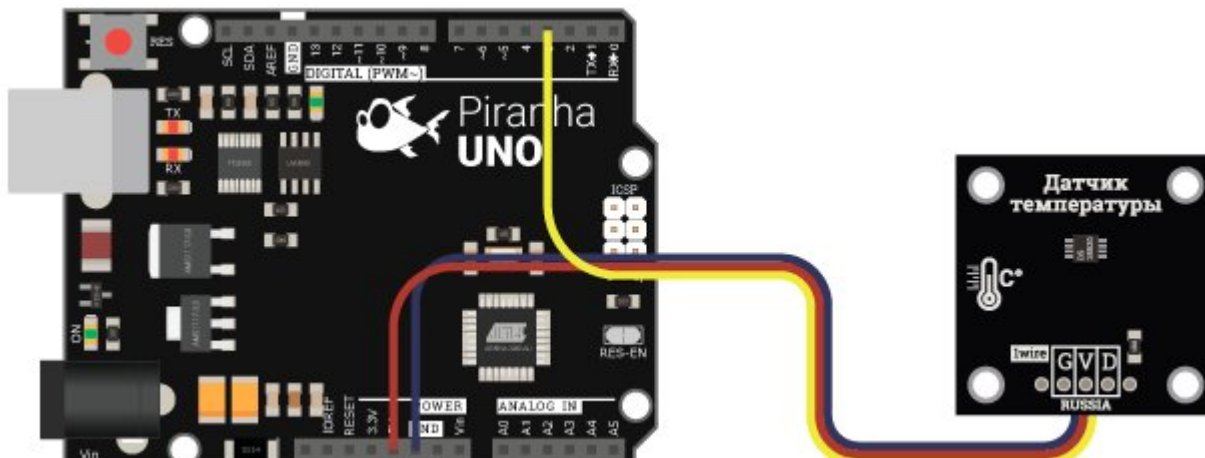
Датчик подключается к любому из цифровых выводов.

В комплекте имеется кабель для быстрого и удобного подключения к [Trema Shield](#).

Модуль удобно подключать 3 способами, в зависимости от ситуации:

### Способ - 1 : Используя проводной шлейф и Piranha UNO

Используя провода «Папа – Мама», подключаем напрямую к контроллеру Piranha UNO







## Подробнее о модуле:

Модуль выполнен на основе микросхемы [DS18B20](#). Он работает в диапазоне температур от  $-55\text{ }^{\circ}\text{C}$  до  $125\text{ }^{\circ}\text{C}$ .

Цифровой сигнал более устойчив от наводок в отличие от [аналогового датчика температуры](#).

Для работы с модулем предлагаем воспользоваться библиотекой [OneWire](#).

Подробнее про установку библиотеки читайте в нашей [инструкции](#).

## Примеры:

### Вывод адреса, модели, температуры в монитор последовательного порта.

```
#include <OneWire.h> // подключаем библиотеку для работы с интерфейсом OneWire
OneWire ds(2); // Создаём объект с именем ds, и указанием пина, к которому подключен датчик

void setup(void) {
  Serial.begin(9600); // иницируем передачу данных в монитор последовательного порта на скорости 9600
}
void loop(void) {
  byte i; // задаём переменную для счётчика
  byte present = 0; // Флаг сброса шины
  byte type_s; // Флаг модели датчика
  byte data[12]; // массив под значения температуры, считанные с датчика
  byte addr[8]; // массив под значение адреса датчика
  float celsius, fahrenheit; // переменные под значения температуры в градусах Цельсия и Фаренгейта
  if ( !ds.search(addr) ) { // если функция принимает значение false, значит
    Serial.println("No more addresses."); // больше устройств не найдено. Выводит в монитор порта сообщение об этом.
    Serial.println(); // добавляем символ новой строки
  }
```

```

ds.reset_search();           // функция начинает новый поиск устройства.
delay(250);                  // задержка 250 мсек
return;
}
Serial.print("ROM =");      // выводим текст в монитор порта. "ROM" - это адрес устройства
for ( i = 0; i < 8; i++) {  // запускаем цикл
    Serial.write(' ');      // между значениями добавляем пробел
    Serial.print(addr[i], HEX); // в монитор порта побайтово выводим адрес устройства в 16-ричной кодировке
}
if (OneWire::crc8(addr, 7) != addr[7]) { // выполняем проверка контрольной суммы CRC
    Serial.println("CRC is not valid!"); // если контрольная сумма не верна - отправляем в монитор порта сообщение об это
    return;
}
Serial.println();           // добавляем символ новой строки
switch (addr[0]) { // Модель датчика температуры
    case 0x10:                Serial.println("  Chip = DS18S20");           Флаг типа модуля           В данн
                             type_s = 1;           break;
    case 0x28:                Serial.println("  Chip = DS18B20");           type_s = 0;           break;
    case 0x22:                Serial.println("  Chip = DS1822");           type_s = 0;           break;
    default:                  Serial.println("Device is not a DS18x20 family device.");           return;
}
ds.reset();                 // производим сброс шины перед началом работы с модулем
ds.select(addr);           // указываем устройство, с которым далее будет работать код
ds.write(0x44, 0);         // отправка команды выполнить конвертацию
delay(1000);               // задержка в 1 сек (для выполнения конвертации)
present = ds.reset();      // флаг сброса шины перед приёмом данных с модуля
ds.select(addr);           // указываем устройство, с которым далее будет работать код
ds.write(0xBE);            // считываем значение из scratchpad-памяти модуля
Serial.print("  Data = "); // в монитор порта выводим текст "Значение"
Serial.print(present, HEX); // выводим в монитор порта значение с модуля, в 16-ричной кодировке
Serial.print(" ");        // добавляем символ новой строки
for ( i = 0; i < 9; i++) { // запускаем цикл считывания 9 байт информации с модуля
    data[i] = ds.read();    // побайтовое чтение значений с модуля
    Serial.print(data[i], HEX); // выводим в монитор порта значение байта, считанного с модуля
}

```

```

    Serial.print(" "); // между значениями добавляем пробел
}
Serial.print(" CRC="); // выводим в монитор порта текст CRC (контрольная сумма)
Serial.print(OneWire::crc8(data, 8), HEX); // выводим в монитор порта значение CRC
Serial.println(); // добавляем символ новой строки
int16_t raw = (data[1] << 8) | data[0]; // конвертация значения температуры в 16-битный тип данных
if (type_s) { // если подключена модель DS18S20
    raw = raw << 3; // тогда сдвигаем значение на 3 влево для получения 12-битного разрешения точно
    if (data[7] == 0x10) { // если указанный бит совпадает с условием, то
        raw = (raw & 0xFFF0) + 12 - data[6]; // пересчитываем сырое значение
    }
} else { // для всех остальных моделей
    byte cfg = (data[4] & 0x60); // для задания разрешения вводится переменная (по умолчанию установлено 12-битно
    if (cfg == 0x00) raw = raw & ~7; // если значение переменной и указанного бита совпадают, тогда устанавливается 9
    else if (cfg == 0x20) raw = raw & ~3; // если значение переменной и указанного бита совпадают, тогда устанавливается 1
    else if (cfg == 0x40) raw = raw & ~1; // если значение переменной и указанного бита совпадают, тогда устанавливается 1
}
celsius = (float)raw / 16.0; // переводим значения в градусы Цельсия
fahrenheit = celsius * 1.8 + 32.0; // переводим значения в градусы Фаренгейта
Serial.println(" Temperature = "); // выводим в монитор порта текста Температура
Serial.print(celsius); // выводим значения в градусах Цельсия
Serial.println(" Celsius, "); // выводим в монитор порта текста "градусы Цельсия"
Serial.print(fahrenheit); // выводим значения в градусах Фаренгейта
Serial.println(" Fahrenheit"); // выводим в монитор порта текста "градусы Цельсия"
}

```

В мониторе порта вы можете наблюдать:



```
Chip = DS18B20
Data = 1 C8 1 0 0 1F FF 8 10 57 CRC=57
Temperature =
28.50 Celsius,
83.30 Fahrenheit
No more addresses.
```

Автопрокрутка    Нет конца строки    9600 бод    Очистить вывод

- **ROM** - адрес устройства на шине OneWire;
- **Chip** - название модуля, подключенного к [Piranha UNO](#);
- **Data** - сырое значение с модуля
- **Temperature Celsius** - температура в градусах Цельсия;
- **Temperature Fahrenheit** - температура в градусах Фаренгейта;

## Применение:

- Метеостанции
- Проекты климат-контроля

## Ссылки:

- Библиотека [OneWire](#);
- [Wiki - Установка библиотек в Arduino IDE](#);
- [Урок 7. Термометр на базе DS18B20 и OLED дисплея 128X64](#);
- [Урок 8.Погодная станция на базе барометра BMP180 термометра DS18B20](#);