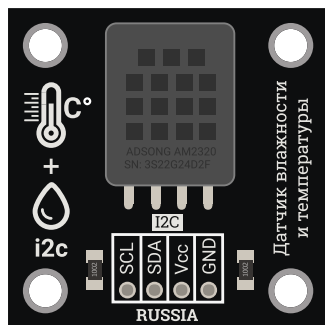


Цифровой датчик температуры и влажности I2C (Трета-модуль)



-
-
-

В этой статье, рассматривается модуль подключаемый к шине I2C, который позволяет, с большой точностью, определить температуру и влажность окружающей среды.

Общие сведения:

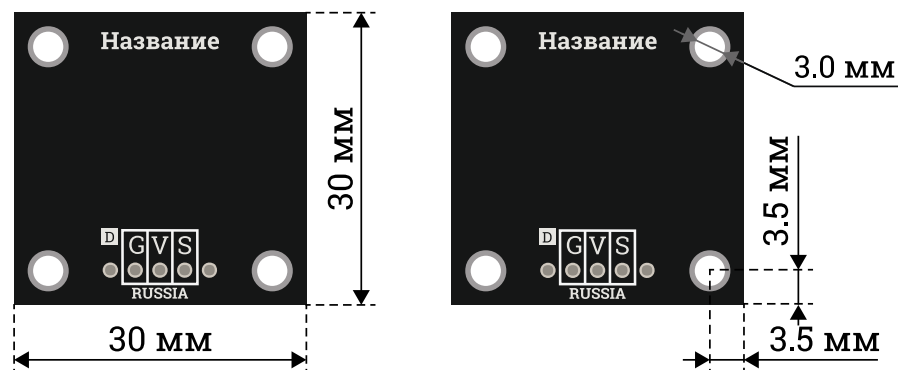
Модуль построен на базе чипа [AM2320](#). По своим техническим характеристикам, модуль превосходит датчики [DHT11](#) и [DHT22](#). Модуль на базе чипа [AM2320](#) позволяет проверять достоверность данных не проверкой контрольной суммой, как в датчиках [DHT11](#) и [DHT22](#), а проверкой циклически избыточного кода, что в разы уменьшает вероятность получения ошибочных данных.

Спецификация:

- Интерфейс: I2C.
- Питание: 3,1 ... 5,5 (номинально 5В).
- Потребляемый ток:
 - В режиме ожидания: 8 ... 10 мкА.
 - В режиме измерений: 350 мкА, пиковый до 950 мкА.
- Измерение температуры:
 - Допустимый диапазон: -40°C .. 80°C.
 - Максимальная погрешность: $\pm 0,5^\circ\text{C}$.
 - Разрешение шкалы: 0,1°C.
- Измерение относительной влажности:

- Допустимый диапазон: 0...99,9%
- Максимальная погрешность: $\pm 3\%$ при $t=25^{\circ}\text{C}$
- Разрешение шкалы: 0.1%
- Уход показаний: 0,5%/год
- Минимальное время между считываниями показаний: 2 сек
- Габариты: 30x30x6,5 мм
- Вес: 4 гр.

Все модули линейки "Трема" выполнены в одном формате



Подключение:

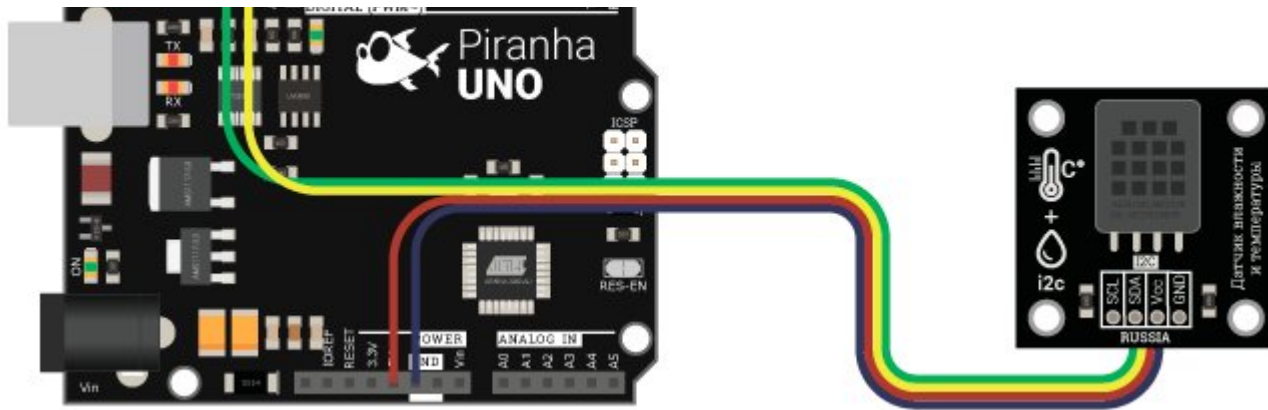
Модуль подключается к [аппаратной](#) или [программной](#) шине I2C [Arduino](#) и имеет адрес 0x5C. Для удобства подключения, предлагаем воспользоваться [TremaShield](#).

Модуль удобно подключать 4 способами, в зависимости от ситуации:

Способ - 1 : Используя проводной шлейф и Piranha UNO

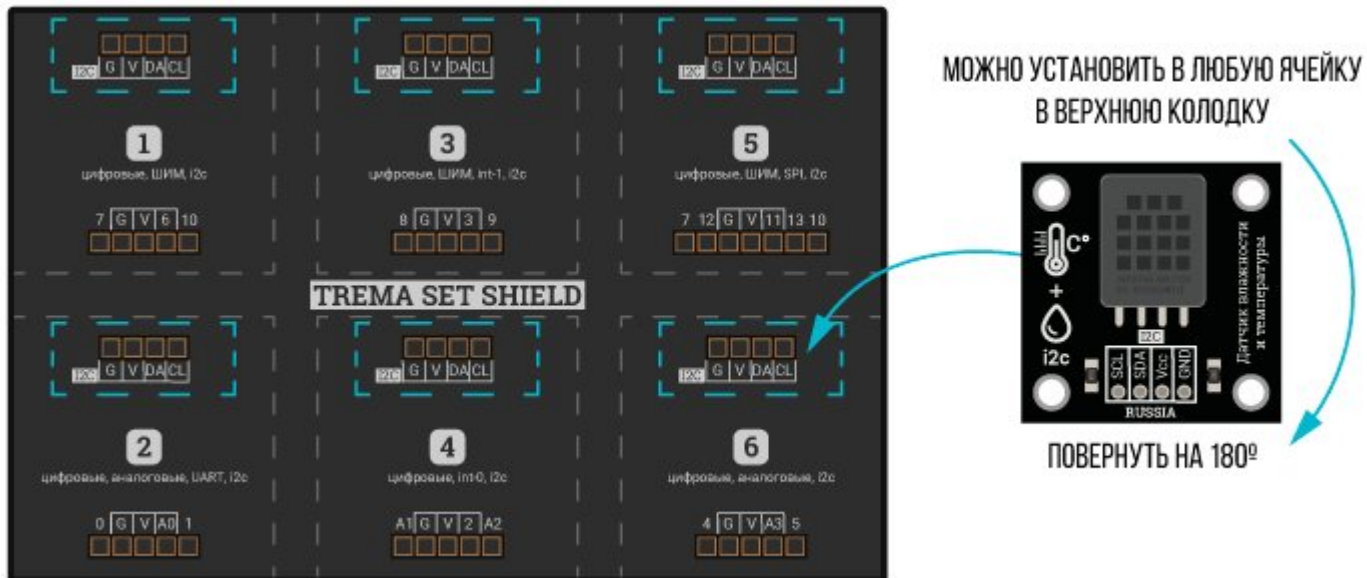
Используя провода «[Папа – Мама](#)», подключаем напрямую к контроллеру Piranha UNO.





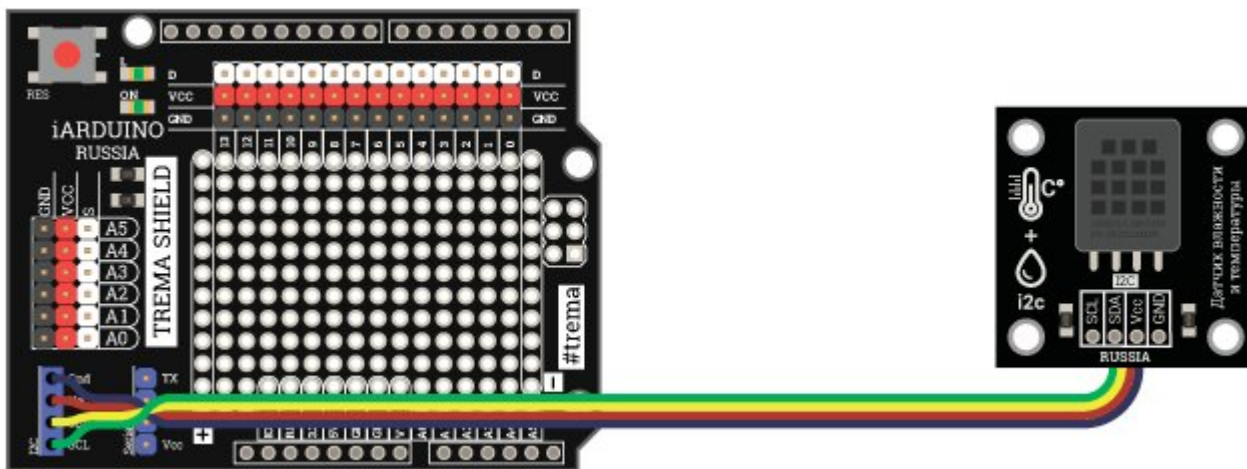
Способ - 2 : Используя Trema Set Shield

Модуль можно подключить к любому из I2C входов Trema Set Shield.



Способ - 3 : Используя проводной шлейф и Shield

Используя 4-х проводной шлейф, к Trema Shield, Trema-Power Shield, Motor Shield, Trema Shield NANO и тд.



Способ - 4 : Альтернативный вариант подключения

Модуль можно подключить к любому выводу [Arduino](#), как при подключении датчиков [DHT11](#) и [DHT22](#):

- Вывод SCL модуля нужно соединить с выводом GND до подачи питания.
- Вывод SDA модуля нужно подключить к любому выводу [Arduino](#) и подтянуть его к Vcc (+5V) через резистор 4,7 кОм ... 10 кОм.
- При таком варианте подключения, для работы с модулем потребуется библиотека [iarduino_DHT](#), а не [iarduino_AM2320](#).
- Преимуществом альтернативной схемы подключения является то, что к одной плате [Arduino](#) можно подключить множество модулей.
- Недостатком альтернативной схемы является то, что между опросами модуля должна выдерживаться пауза в 3-4 секунды.

При подключении [датчика температуры и влажности AM2320](#) по шине I2C к другим платам, например, [WEMOS D1 mini](#) или [WEMOS D1 mini Pro](#) на базе микроконтроллера ESP8266, и т.д. То перед подключением библиотеки [iarduino_AM2320](#), нужно подключить библиотеку Wire, как это описано в разделе [Wiki - расширенные возможности библиотек iarduino для шины I2C](#).

Питание:

Модуль поддерживает питание 3,3 В или 5 В постоянного тока.

Подробнее о модуле:

Для работы с модулем, предлагаем воспользоваться библиотекой [iarduino_AM2320](#), синтаксис которой идентичен библиотеке [iarduino_DHT](#)

для работы с датчиками [DHT11](#) и [DHT22](#).

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

Примеры:

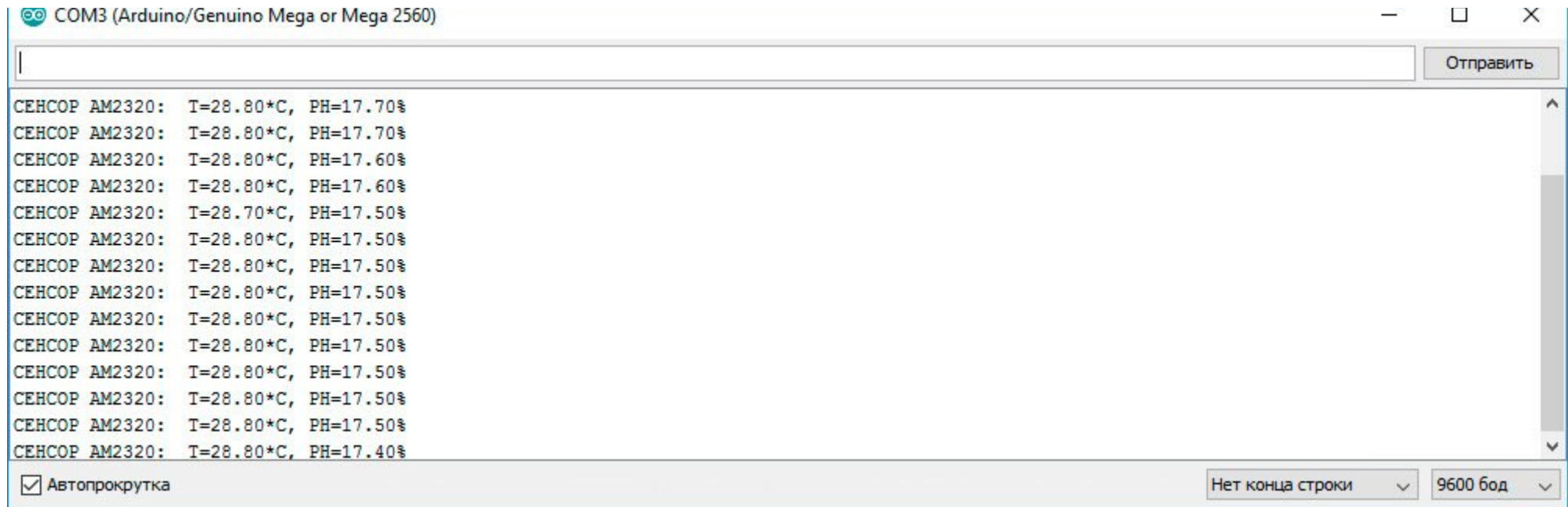
Получение данных от модуля в монитор последовательного порта, с выводом всех возможных ошибок:

```
#include <iarduino_AM2320.h> // Подключаем библиотеку iarduino_AM2320.
iarduino_AM2320 sensor; // Объявляем объект sensor для работы с датчик
//
//
void setup(){ // Иницируем работу с датчиком AM2320.
  sensor.begin(); // Иницируем передачу данных в монитор послед
  Serial.begin(9600); //
//
//
} // Читаем показания датчика.

void loop(){
  switch(sensor.read()){
    case AM2320_OK: Serial.println((String) "СЕНСОР AM2320: T=" + sensor.tem + "*C, PH=" + sensor.hum + "%"); break;
    case AM2320_ERROR_LEN: Serial.println("ОТИПАВКА НЕВОЗМ."); break; // объем передаваемых данных превышает буфер I
    case AM2320_ERROR_ADDR: Serial.println("НЕТ СЕНСОРА"); break; // получен NACK при передаче адреса датчика
    case AM2320_ERROR_DATA: Serial.println("ОТИПАВКА НЕВОЗМ."); break; // получен NACK при передаче данных датчику
    case AM2320_ERROR_SEND: Serial.println("ОТИПАВКА НЕВОЗМ."); break; // ошибка при передаче данных
    case AM2320_ERROR_READ: Serial.println("НЕТ ОТВЕТА ОТ СЕНСОРА"); break; // получен пустой ответ датчика
    case AM2320_ERROR_ANS: Serial.println("ОТВЕТ НЕКОРРЕКТЕН"); break; // ответ датчика не соответствует запросу
    case AM2320_ERROR_LINE: Serial.println("НЕПАВЕНСТВО CRC"); break; // помехи в линии связи (не совпадает CRC)
  }
  delay(2000); // ждём 2 секунды
}
```

Результат работы скетча:





Описание основных функций библиотеки:

Данная библиотека может использоваться как аппаратная, так и программная реализация шины I2C.

О том как выбрать тип шины I2C рассказано в статье [Wiki - расширенные возможности библиотек iarduino для шины I2C](#).

Подключение библиотек:

```
#include <iarduino_AM2320.h> // Подключаем библиотеку iarduino_AM2320, для работы с модулем  
iarduino_AM2320 sensor;    // Объявляем объект sensor, для работы с функциями библиотеки iarduino_AM2320
```

В библиотеке реализованы 2 функции:

Функция `begin()`;

- Назначение: Инициализация работы с датчиком AM2320.
- Синтаксис: `begin()`;
- Параметры: Нет.

- Возвращаемые значения: Нет.
- Примечание: Вызывается 1 раз в коде setup.
- Пример:

```
void setup(){  
  sensor.begin();           // Инициуруем работу с датчиком AM2320  
}
```

Функция read();

- Назначение: Чтение показаний датчика в переменные: tem и hum.
- Синтаксис: read();
- Параметры: Нет.
- Возвращаемые значения:
 - AM2320_OK - успешное чтение данных
 - AM2320_ERROR_LEN - объем передаваемых данных превышает буфер I2C
 - AM2320_ERROR_ADDR - получен NACK при передаче адреса датчика
 - AM2320_ERROR_DATA - получен NACK при передаче данных датчику
 - AM2320_ERROR_SEND - ошибка при передаче данных
 - AM2320_ERROR_READ - получен пустой ответ датчика
 - AM2320_ERROR_ANS - ответ датчика не соответствует запросу
 - AM2320_ERROR_LINE - помехи в линии связи (не совпадает CRC)
- Примечание: Значения переменных tem и hum обновляются только при успешном чтении данных датчика.
- Пример:

```
int A = sensor.read();      // Читаем показания датчика (данные с датчика будут записаны в переменные tem и hum, а в перемен
```

```
sensor.read();             // Читаем показания датчика (данные с датчика будут записаны в переменные tem и hum). В данном п
```



```
// Если данные от датчика прочитаны с ошибкой, то в переменных tem и hum останутся предыдущие зн
```

Переменные:

- `tem` : тип float, содержит значение температуры окружающей среды в °C
- `hum` : тип float, содержит значение атмосферного давления в %
- Значения переменных `tem` и `hum` обновляются только при успешном чтении данных датчика функцией `read()`.

```
sensor.read();           // Читаем показания датчика
float A = sensor.tem;    // Сохраняем прочитанное значение температуры в переменную A
float B = sensor.hum;    // Сохраняем прочитанное значение влажности в переменную B
```

Применение:

- Метеостанции
- Проекты климат-контроля

Ссылки:

- [Библиотека `iarduino_AM2320`.](#)
- [Расширенные возможности библиотек `iarduino` для шины I2C.](#)
- [Урок 21.2 Метеостанция на базе датчика AM2320 I2C и четырехразрядного LED индикатора.](#)
- [Wiki - Установка библиотек в Arduino IDE.](#)