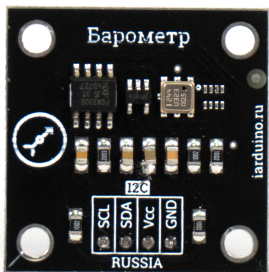


Барометр BMP180 и BMP280 (датчик атмосферного давления, высотомер) (Trema-модуль v2.0)



Общие сведения:

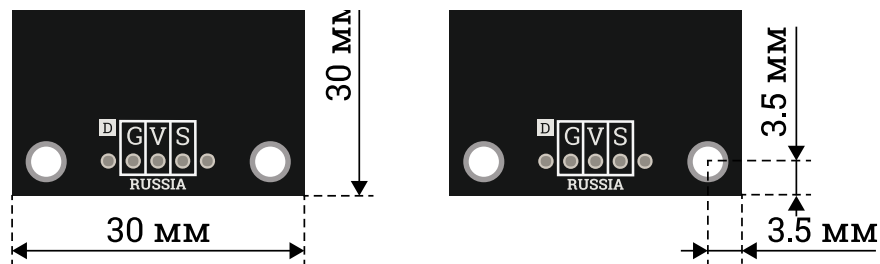
[Trema-модуль датчик давления BMP280](#) - это цифровой модуль на базе чипа BMP280, подключаемый по шине I2C (адрес 0x77), позволяющий получить текущие значения атмосферного давления и температуры окружающей среды. Еще одним применением данного модуля является определение высоты, которая зависит от давления и рассчитывается по международной барометрической формуле. Точность Trema барометра позволяет фиксировать изменение высоты от 20 см.

Спецификация VMP280:

- Напряжение питания модуля: 3,3 или 5 В постоянного тока (поддерживаются оба уровня).
- Потребляемый ток: до 2 мА во время измерений (зависит от режима точности).
- Потребляемый ток: до 0,2 мА в режиме ожидания.
- Измеряемое давление: от 30'000 до 110'000 Па (разрешение 0,16 Па)
- Измеряемая температура: от 0 до +65 °С (разрешение 0,01°С)
- Рабочая частота шины I2C: до 3,4 МГц.
- Адрес модуля на шине I2C: 0x77.
- Уровень логической «1» на шине I2C: от $0,7 \cdot V_{cc}$ до V_{cc} (где V_{cc} это напряжение питания модуля)
- Подготовка к первому запуску после подачи питания: не менее 2 мс.
- Рабочая температура: -40 ... +85 °С
- Габариты: 30x30 мм.

Все модули линейки "Трема" выполнены в одном формате





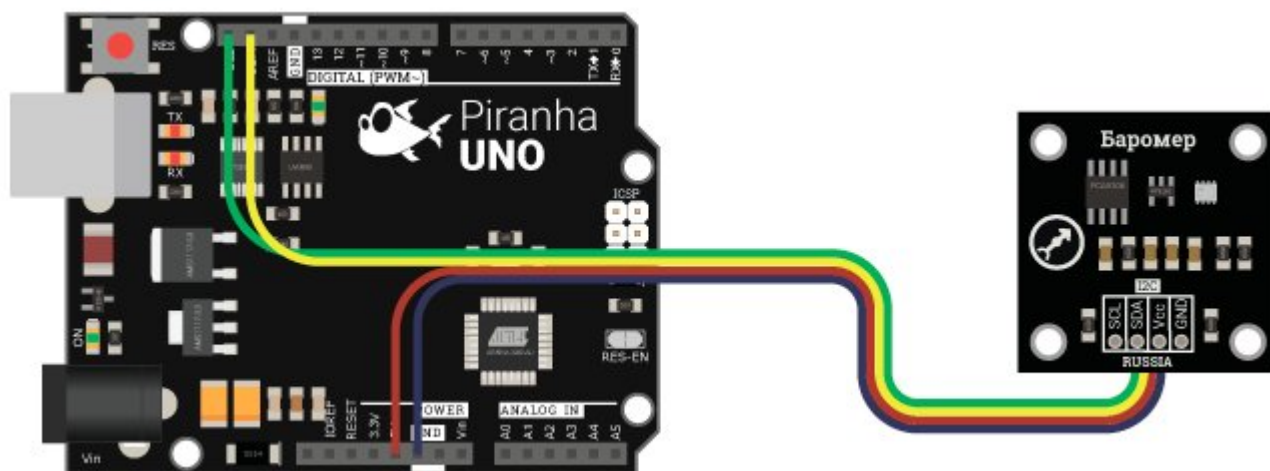
Подключение:

Модуль подключается к [аппаратной](#) или [программной](#) шине I2C [Arduino](#). Логические уровни шины I2C не должны превышать напряжение питания.

Модуль удобно подключать 3 способами, в зависимости от ситуации:

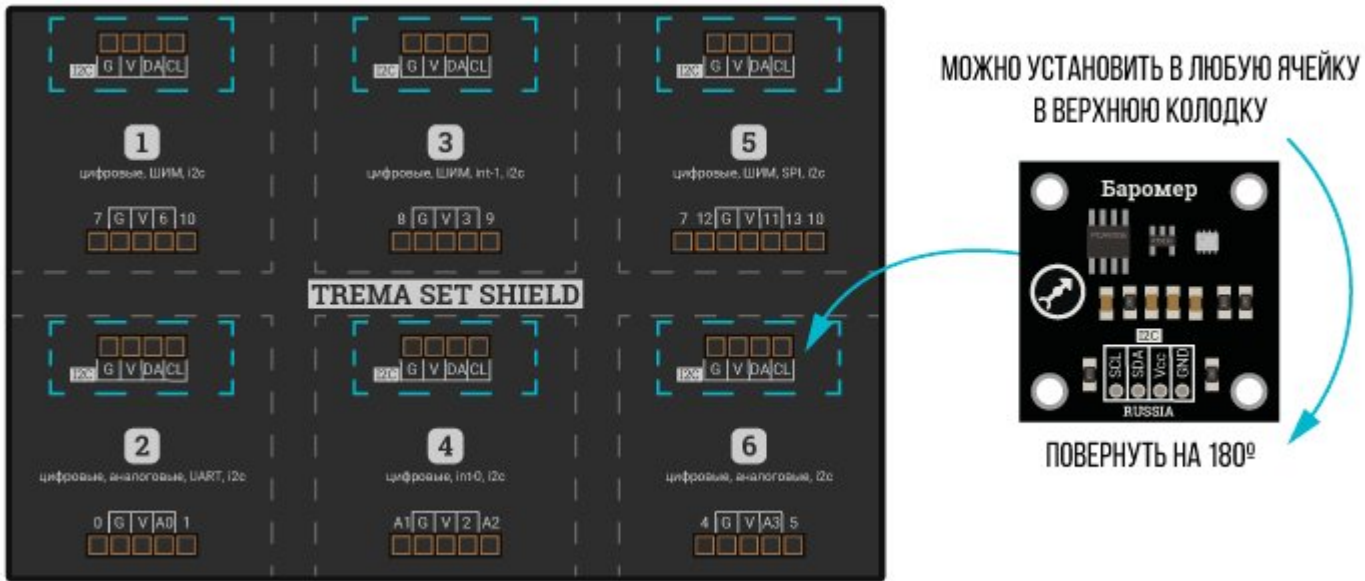
Способ - 1 : Используя проводной шлейф и Piranha UNO

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



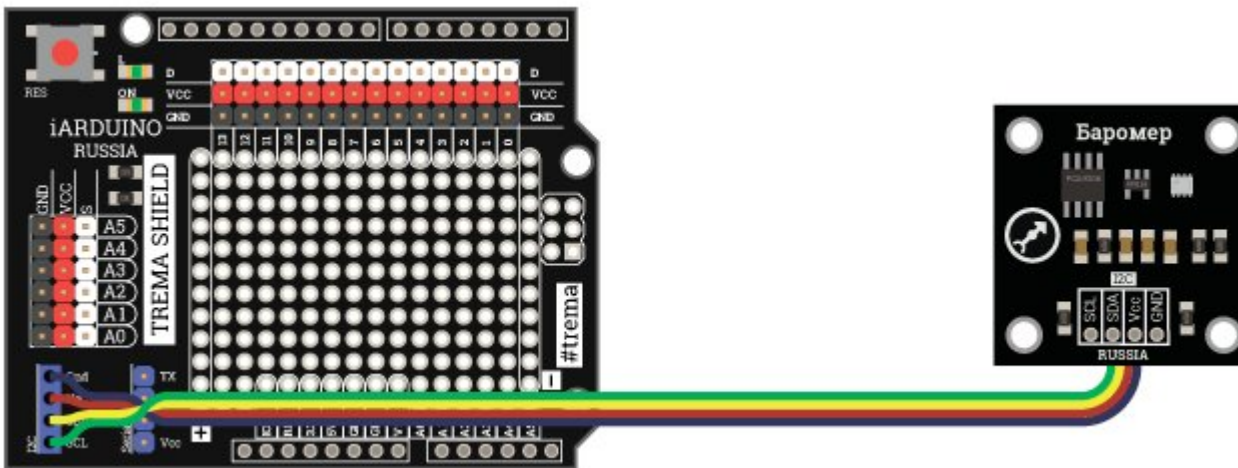
Способ - 2 : Используя Trema Set Shield

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.



При подключении [датчика давления BMP280](#) к другим платам, например, [WEMOS D1 mini](#) или [WEMOS D1 mini Pro](#) на базе

микроконтроллера ESP8266, и т.д. То перед подключением библиотеки [iarduino Pressure BMP](#), нужно подключить библиотеку Wire, как это описано в разделе [Wiki - расширенные возможности библиотек iarduino для шины I2C](#).

Питание:

Входное напряжение питания от 3,3 до 5,5 В постоянного тока, подаётся на выводы VIN и GND модуля.

Подробнее о модуле:

Трета-модуль датчик давления построен на базе чипа [BMP280](#) (Bosch Module Pressure), оснащён преобразователем уровней на базе чипа PCA9306 и линейным стабилизатором на базе чипа RT9193. Чип BMP280 оснащён пьезорезистивным датчиком, термодатчиком, АЦП, памятью EEPROM и RAM, а так же микроконтроллером с поддержкой циклического вычисления измерений (при получении запроса, модуль сразу возвращает ответ, не тратя время на вычисления, как это делал чип BMP180). Чип PCA9306 позволяет передавать данные по шине I2C с уровнями от 3,3 до 5,5 В. Чип RT9193, позволяет подключать Трета-модуль к источнику питания от 3,3 до 5,5 В постоянного тока. Основной чип модуля, BMP280, имеет меньшие габариты и лучшие характеристики по сравнению со своими предшественниками BMP185 и BMP085.

Для работы с модулями на базе чипов BMP180 и [BMP280](#), предлагаем воспользоваться разработанной нами библиотекой [iarduino Pressure BMP](#), позволяющей получать температуру, давление и высоту.

У нашей библиотеки есть ряд преимуществ: она позволяет заменять датчики BMP180 / BMP280 без изменения скетча и схемы включения; она позволяет выбирать режим точности показаний (значение передискретизации) и единицы измерения выводимого давления (Па или мм.рт.ст.); для расчёта высоты, ей не нужно указывать давление над уровнем моря, а достаточно указать любую начальную высоту в качестве аргумента функции begin([высота]). Если вы указали 0 метров (значение по умолчанию), то подняв модуль на 10 метров - получите результат +10, а опустив на 10 метров - получите результат -10. Если в качестве аргумента функции begin([высота]) указать высоту над уровнем моря, то все остальные значения высоты будут соответствовать действительной высоте над уровнем моря. Уменьшая точность измерений Вы ускорите процесс получения результата.

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

Примеры:

Вывод значений давления в мм.рт.ст. и в Па:

```

#include <iarduino_Pressure_BMP.h> // Подключаем библиотеку для работы с датчиками BMP180 или BMP280
iarduino_Pressure_BMP sensor; // Создаём объект sensor для работы с датчиком адрес которого на шине I2C установлен по у
void setup(){
    Serial.begin(9600);
    delay(1000);
    sensor.begin(); // Иницируем работу с датчиком (начальная высота по умолчанию = 0 метров)
}
void loop (){
    // Читаем показания (давление возвращается в мм.рт.ст.)
    if(sensor.read(1)) {Serial.println((String)"СЕНСОР BMP" + sensor.type + ": \t P = " + sensor.pressure + "\tM
    else {Serial.println("НЕТ ОТВЕТА ОТ СЕНСОРА");}
    // Читаем показания (давление возвращается в Па)
    if(sensor.read(2)) {Serial.println((String)"СЕНСОР BMP" + sensor.type + ": \t P = " + sensor.pressure + "\tP
    else {Serial.println("НЕТ ОТВЕТА ОТ СЕНСОРА");}
    delay(3000); // Ждём 3 секунды (не обязательно)
}

```

Описание основных функций библиотеки:

Данная библиотека может использовать как аппаратную, так и программную реализацию шины I2C.

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

Подключение библиотеки:

```

// Для работы с датчиком BMP180 или BMP280 у которого установлен адрес I2C по умолчанию (
#include <iarduino_Pressure_BMP.h> // Подключаем библиотеку для работы с датчиками BMP180 / BMP280
iarduino_Pressure_BMP sensor; // Создаём объект sensor
// Для работы с датчиком BMP180 или BMP280 у которого установлен не стандартный адрес ши-
#include <iarduino_Pressure_BMP.h> // Подключаем библиотеку для работы с датчиками BMP180 / BMP280
iarduino_Pressure_BMP sensor(0x76); // Создаём объект sensor указывая адрес датчика на шине I2C

```

Функция `begin()`;

- Назначение: Инициализация работы с датчиком.
- Синтаксис: `begin([ВЫСОТА]);`
- Параметры:
 - `ВЫСОТА` - начальная высота в метрах (например высота над уровнем моря).
- Возвращаемые значения: `bool` - результат инициализации (`true` или `false`).
- Примечание:
 - Если вызвать функцию без параметра, то начальная высота будет равна 0 м.
 - Функцию можно вызвать однократно (в коде `setup`), а можно и многократно (для корректировки или изменения начальной высоты).
 - После вызова функции в переменные `type` и `version` будет сохранён тип и версия датчика.
- Пример:

```
sensor.begin();           // Иницируем работу датчика с начальной высотой модуля 0 метров.  
                           // Если после этого поднять модуль на 5 метров и вызвать функцию read(), то в переменную  
                           // Если после этого опустить модуль на 25 метров и вызвать функцию read(), то в переменную
```

Функция `read()`;

- Назначение: Чтение показаний датчика в переменные **`temperature`**, **`pressure`** и **`altitude`**.
- Синтаксис: `read([ЦИФРА]);`
- Параметры:
 - `ЦИФРА` - 1 или 2 (по умолчанию 1). Если 1 то давление рассчитывается в мм.рт.ст., если 2 то давление рассчитывается в Па
- Возвращаемые значения: `bool` - результат успешности чтения (`true` или `false`).
- Примечание:
 - Если вызвать функцию без параметра, то атмосферное давление будет рассчитано в мм.рт.ст.
 - После вызова данной функции в переменные **`temperature`**, **`pressure`** и **`altitude`** будут сохранены новые значения.
- Пример:

```
sensor.read(); // Читаем показания датчика и сохраняем их в переменные temperature, pressure и altitude.
```

Функция `measurement()`;

- Назначение: Выбор точности измерений.
- Синтаксис: `measurement(ЦИФРА);`
- Параметры:
 - ЦИФРА - положительное целочисленное значение от 0 (минимальная) до 3 (для `bmp180`) или 4 (для `bmp280`) включительно.
- Возвращаемые значения: `bool` - результат применения (`true` или `false`).
- Примечание:
 - Увеличение точности показаний достигается снижением уровня шумов благодаря увеличению передискретизации при считывании показаний.
 - Данная функция является необязательной, т.к. по умолчанию установлена максимальная точность.
 - Если значение параметра будет выше допустимого, то установится максимальная точность.
 - Уменьшение точности приведёт к уменьшению энергопотребления, а при работе с датчиком `bmp180` еще и к ускорению чтения данных.
- Пример:

```
sensor.measurement(0); // Устанавливаем самый низкий уровень точности (передискретизации) при считывании данных.
```

Переменная `temperature`

- Значение: Содержит температуру окружающей среды в °C.
- Тип данных: `float`.

Переменная `pressure`

- Значение: Содержит атмосферное давление в мм.рт.ст. или Па.
- Тип данных: `float`.

Переменная altitude

- Значение: Содержит высоту относительно начальной в м.
- Тип данных: float.

Переменная type

- Значение: Содержит тип датчика (число 180 или 280)
- Тип данных: int.

Переменная version

- Значение: Содержит версию датчика
- Тип данных: int.

Применение:

- Получение данных для метеостанций
- Получение высоты для летательных аппаратов
- Получение глубины для подводных аппаратов (требуется поместить модуль в додонепроницаемый пакет и пересчитывать глубину исходя из показаний давления, а не по данным высоты из библиотеки)

Ссылки:

- [Библиотека iarduino_Pressure_BMP.](#)
- [Расширенные возможности библиотек iarduino для шины I2C.](#)
- [Wiki - Установка библиотек в Arduino IDE.](#)