Зуммер (Trema-модуль)

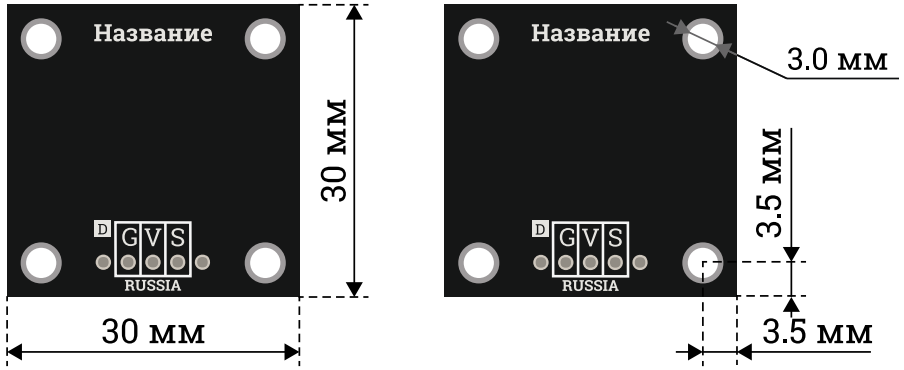
Общие сведения:

Trema-модуль зуммер пассивный *и* *Trema-модуль зуммер активный* - позволяют излучать звук различными способами, в зависимости от выбранной модели зуммера.

Спецификация:

- Напряжение питания: 5 В
- Потребляемый ток: до 30 мА
- Интенсивность звука: >= 85 дБ
- Резонансная частота: 2048 Гц
- Сопротивление обмотки: 40 Ом
- Рабочая температура: -20 ... 70 °C
- Габариты: 30x30x9 (без учёта выводов)

Все модули линейки "Trema" выполнены в одном формате



Подключение:

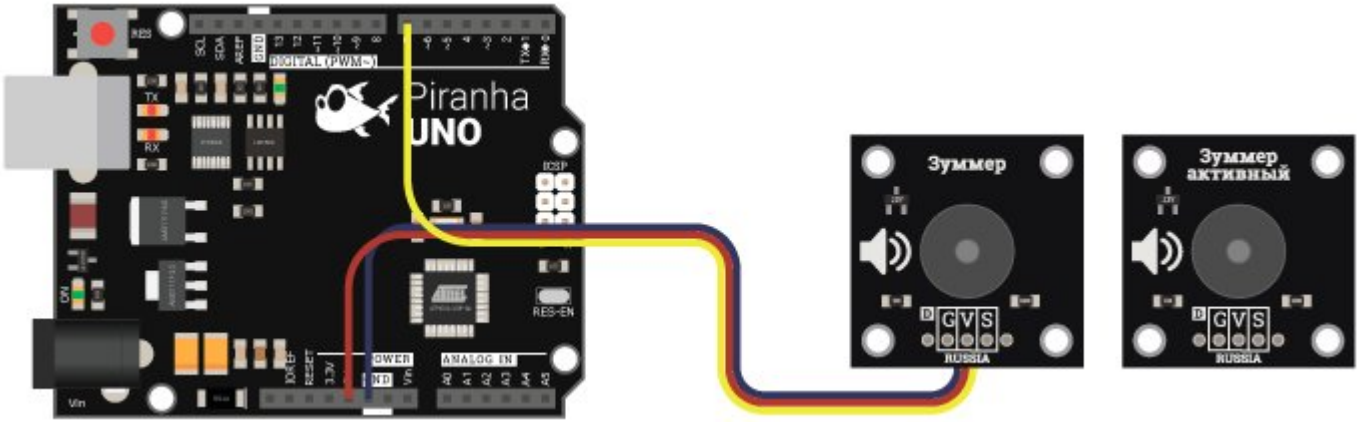
Trema-модуль зуммер пассивный и *Trema-модуль зуммер активный* входят в линейку Trema-модулей, что позволяет подключить их к Arduino через Trema Shield по 3-проводному шлейфу (который идёт в комплекте с зуммером) без пайки, без дополнительных проводов и переходников. Их можно подключать к любому выводу Arduino, как цифровому, так и аналоговому.

Модули имеют три вывода: Signal (S) - вход и два вывода питания Vcc (V) и GND (G).

Модули 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.

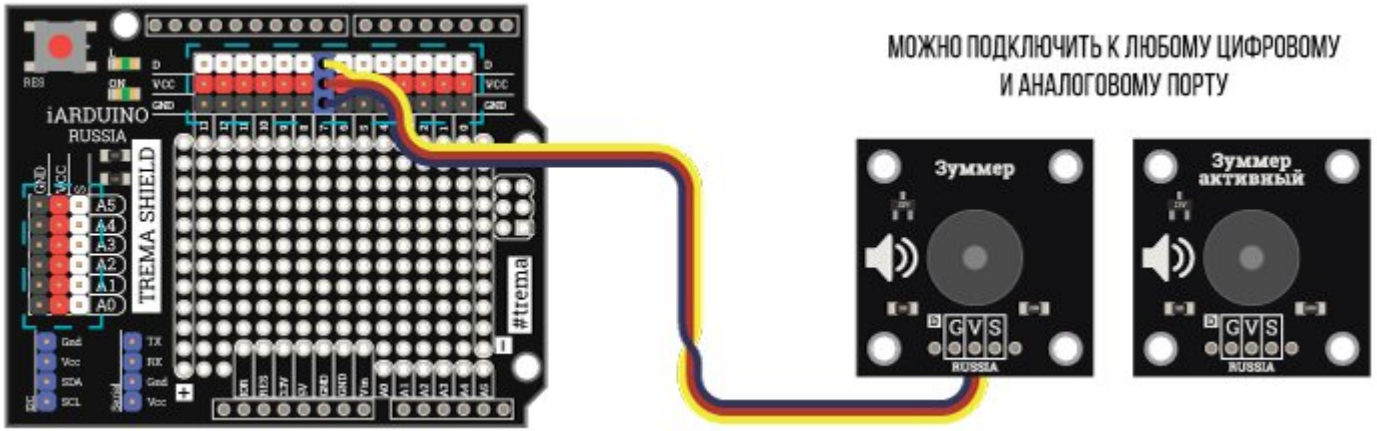


Способ - 2 : Используя Trema Set Shield

Модули можно подключить к любому из цифровых или аналоговых входов Trema Set Shield.

Способ - 3 : Используя проводной шлейф и Shield

Используя 3-х проводной шлейф, к Trema Shield, Trema-Power Shield, Motor Shield, Trema Shield NANO и тд.



Питание:

Входное напряжение питания 5 В постоянного тока, подаётся на выводы Vcc (V) и GND (G).

Подробнее о модулях:

Trema-модуль зуммер пассивный основан на электромагнитном излучателе, который состоит из кольцевого магнита, сердечника с электромагнитной катушкой и гибкой металлической мембраны. Электромагнитная катушка преобразует электрические колебания в магнитные, а мембрана преобразует магнитные колебания в механические. Полученные механические колебания распространяются по воздуху в виде звуковых волн. Пластиковый корпус, с отверстием, усиливает акустический эффект.

Trema-модуль зуммер пассивный не имеет встроенного генератора, а преобразует электрический сигнал со входа (S) в механические колебания воздуха. Таким образом частота излучаемого звука соответствует частоте сигнала подаваемого на вход модуля. Чем выше частота, тем «тоньше» звук. Чем ближе частота к резонансной, тем звук сильнее.

Самый простой способ получения звука заключается в применении функции **tone()**. Данная функция генерирует меандр (сигнал прямоугольной формы с равной длительностью импульсов и пауз), с заданной частотой и длительностью.

Trema-модуль зуммер активный состоит из 5В генератора прямоугольных импульсов (меандра) с частотой 2,3 кГц, и электромагнитного излучателя в одном корпусе. Сигнал с генератора подается на электромагнитный излучатель и преобразуется в звуковые волны той же частоты.

Trema-модуль зуммер активный уже имеет встроенный генератор и для генерации звука ему не требуется использование функции beep() и ей аналогичных (как для простого Trema-зуммера). Достаточно установить состояние логической «1» на выводе «S» и Вы услышите сигнал с частотой 2,3 кГц и уровнем звукового давления не ниже 85дБ/10см.

Примеры для зуммера пассивного :

Вывод двух коротких звуковых сигнала функцией tone(), сигнализирующих о включении Arduino:

```
const uint8_t pinBF = 2;                    // Определяем номер вывода к которому подключён зуммер
                                            //
void setup(){                               //
    tone(pinBF, 2048, 100);                 // Выводим звуковой сигнал с частотой 2048 Гц и длительностью 0,1 сек
    delay(200);                             // Не выводим звук в течении 0,1 сек (см. ниже)
    tone(pinBF, 2048, 100);                 // Выводим звуковой сигнал длительностью 0,1 сек с частотой 2048 Гц
}                                           //
void loop(){}                               // Функция loop в данном примере не используется
```

Обратите внимание на то, что в примере между двумя вызовами функции tone() устанавливается задержка на **0,2** секунды, а в комментарии написано «не выводим звук в течении **0,1** сек». Дело в том, что функция tone() выводит сигнал используя прерывания

аппаратного таймера и не приостанавливает выполнение скетча на время вывода сигнала. Значит, сразу после начала вывода первого звукового сигнала, стартует функция delay(), приостанавливая выполнение скетча на 0,2 сек. но звук продолжает выводиться. Значит первые 0,1 сек - выводится сигнал, следующие 0,1 сек - тишина и последние 0,1 - опять выводится сигнал.

Может возникнуть ситуация, когда использование функции tone() невозможно, например, если аппаратный таймер используется для других целей. Тогда сигнал придётся генерировать самим, используя функцию digitalWrite():

Тот же пример, но без использования функции tone():

```
void myTone(uint8_t, uint32_t, uint32_t);         // Определяем собственную функцию для генерации звука
const uint8_t pinBF = 2;                          // Определяем номер вывода к которому подключён зуммер
                                                  //
void setup(){                                     //
    myTone(pinBF, 2048, 100);                     // Выводим звуковой сигнал длительностью 0,1 сек с частотой 2048 Гц
    delay(100);                                   // Ждём 0,1 сек
    myTone(pinBF, 2048, 100);                     // Выводим звуковой сигнал длительностью 0,1 сек с частотой 2048 Гц
}                                                 //
void loop(){}                                     // Функция loop в данном примере не используется
                                                  //
void myTone(uint8_t i, uint32_t j, uint32_t k){   // Определяем функцию myTone
  j=500000/j;                                     // Меняем значение переменной j на время одного полупериода в мкс
  k+=millis();                                    // Меняем значение переменной k на время завершения вывода сигнала
  pinMode(i, OUTPUT);                             // Конфигурируем вывод для вывода сигнала как выход
  while(k>millis()){                              // Выводим сигнал, пока не истечёт указанное время
    digitalWrite(i, HIGH); delayMicroseconds(j); // Устанавливаем на выходе i уровень логической  «1» на время j
    digitalWrite(i, LOW ); delayMicroseconds(j); // Устанавливаем на выходе i уровень логического «0» на время j
  }
}
```

Обратите внимание на то, что теперь задержка между первым и вторым вызовом функции myTone() соответствует паузе между сигналами в 0,1 сек, так как функция myTone() приостанавливает выполнение скетча на время вывода звукового сигнала.

Сама функция **myTone()** не сложна в понимании:

- Сначала значение переменной **j** получившей частоту в Гц преобразуем в длительность одного полупериода в мкс.
  Период **T = 1 / F** (сек), значит полупериод **L = 0,5 T** (сек) = **0,5 / F** (сек) = **500'000 / F** (мкс).
- Далее к переменной **k** получившей длительность импульса, добавляется время с начала старта скетча **millis()**
- Конфигурируем вывод номер которого получила переменная **i** как выход
- Чередуем на выводе **i** логические уровни, устанавливая их на длительность **J**
  пока время прошедшее с момента старта скетча **millis()** не сравняется со значением переменной **k**.

Примеры для зуммера активного:

Вывод короткого звукового сигнала.

```
uint8_t pinBuzzer = 2;                          // Определяем № вывода к которому подключён зуммер со встроенным генератором
                                                // Можно использовать любой вывод Arduino
void setup(){
  pinMode(pinBuzzer,  OUTPUT  );                // Переводим вывод pinBuzzer в режим выхода
  digitalWrite(pinBuzzer, LOW );                // Устанавливаем уровень логического «0» на выводе pinBuzzer
}
void loop(){
  digitalWrite(pinBuzzer, HIGH); delay(500);    // Включаем звук на 0,5 секунд
  digitalWrite(pinBuzzer, LOW ); delay(1000);   // Выключаем звук на 1 секунду
}
```

Как видно из скетча, управлять Trema-зуммером со встроенным генератором так же легко, как и обычным светодиодом.

Применение:

- Информирование о событиях
- Вывод мелодий
- Создание микровибраций