ИК-передатчик

Общие сведения:

Trema-модуль ИК-передатчик - позволяет управлять проектами на расстоянии совместно с ИК-приёмником. Он исполнен в линейке Trema-модулей, что позволяет включать модуль в проект, без пайки и макетных плат.

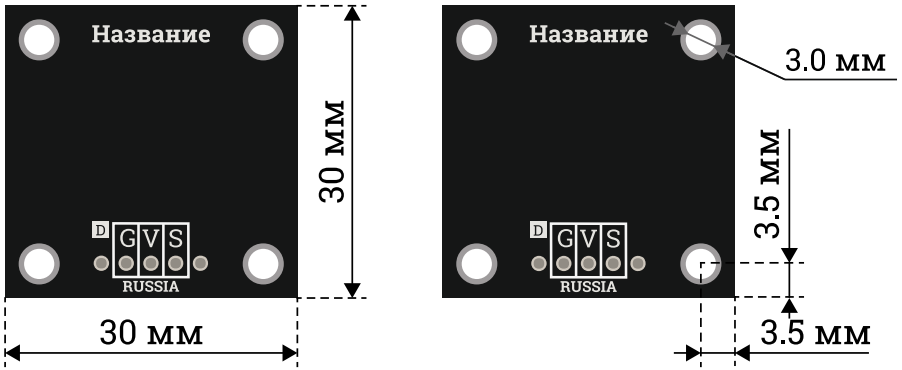
Модуль ИК-передатчика построен на базе ИК-светодиода

U5293IRC.

Спецификация:

- Входное напряжение: 4,0 … 5,5 В (номинально 5 В)
- Потребляемый ток: до 100 мА в импульсном режиме (при Vcc = 5 В)
- Длинна световой волны: 940 нм (пиковое значение)
- Максимальная частота сигнала: до 10 МГц
- Расстояние передачи: до 10 м (при Vcc = 5 В)
- Рабочая температура: -25 … 85 °C
- Угол направленности: 120° (с потерей мощности < 50%)

Все модули линейки "Trema" выполнены в одном формате



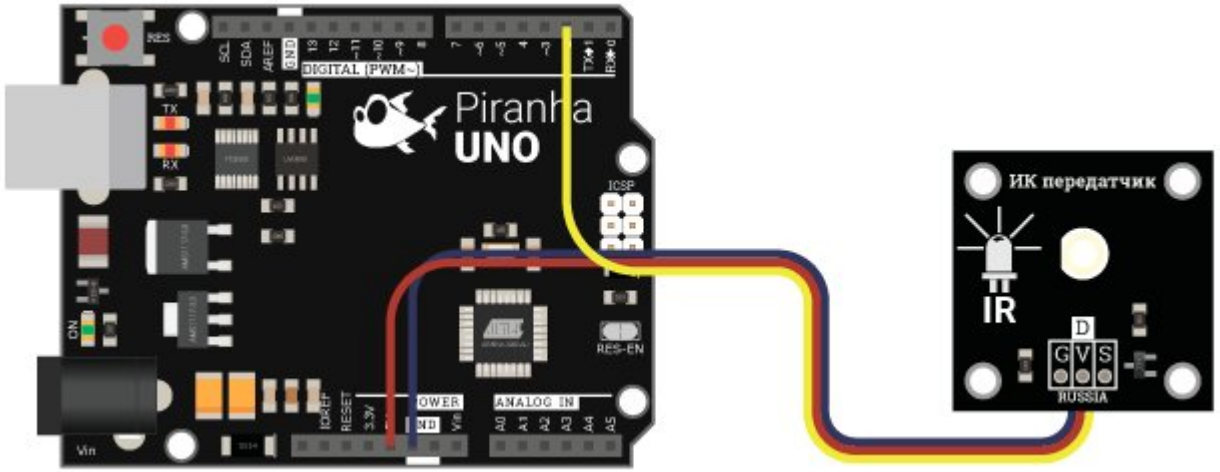
Подключение:

Модуль подключается к любому цифровому выводу Arduino. В комплекте имеется кабель для быстрого и удобного подключения к Trema Shield.

Модуль удобно подключать 3 способами, в зависимости от ситуации:

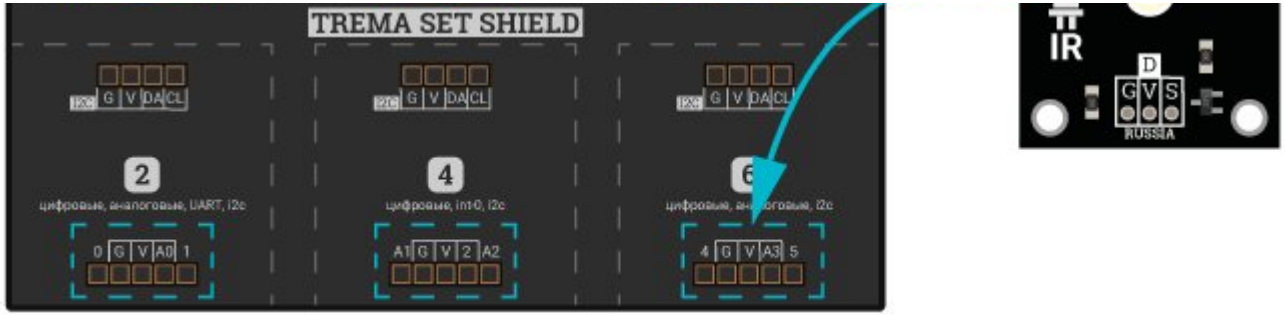
Способ - 1 : Используя проводной шлейф и Piranha UNO

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



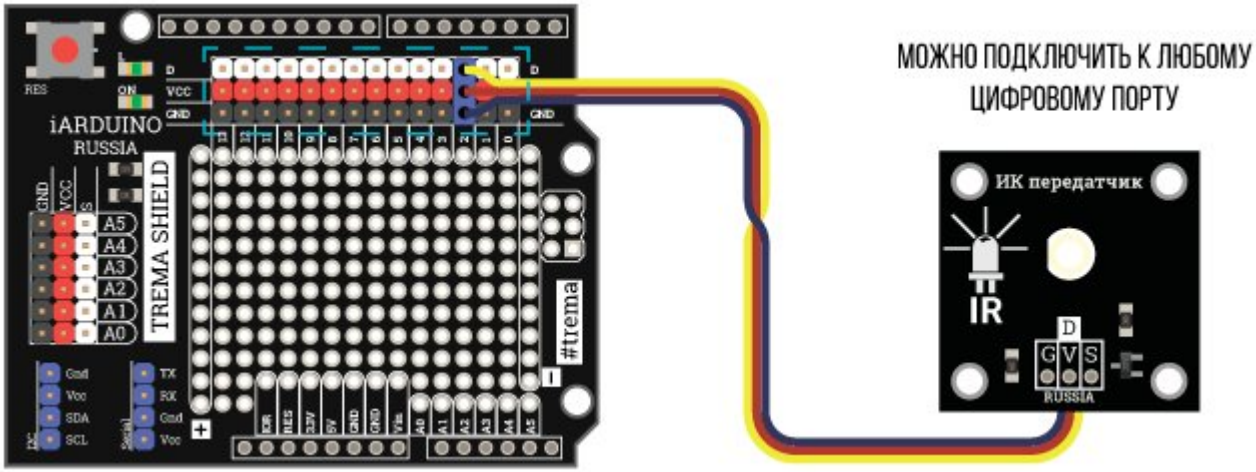
Способ - 2 : Используя Trema Set Shield

Модуль можно подключить к любому из цифровых 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.



Подробнее о модуле:

Для передачи данных ИК-передатчиком, предлагаем воспользоваться библиотекой iarduino_IR, которая позволяет работать с ИК-приёмником и(или) ИК-передатчиком.

Библиотека использует второй аппаратный таймер,
НЕ ВЫВОДИТЕ СИГНАЛЫ ШИМ НА 3 ИЛИ 11 ВЫВОД!

Подробнее про установку библиотеки читайте в нашей инструкции..

Дополнительная информация по работе с модулем:

Пакеты: Практически все пульты отправляют не только информационный пакет (указывающий тип устройства и код нажатой кнопки), но и пакеты повтора, сообщающие устройству об удержании нажатой кнопки. Таким образом принимающее устройство может реагировать на нажатие кнопки однократно или в течении всего времени её удержания.
Например: нажимая и удерживая кнопку с номером телевизионного канала, телевизор переключится на данный канал только один раз. В то время, как нажимая и удерживая кнопку увеличения громкости, телевизор будет её увеличивать в течении всего времени удержания кнопки.

Количество информационных пакетов у большинства пультов равно одному, но некоторые устройства, например кондиционеры, используют 2, 3 и более информационных пакетов.

Состав пакетов: Информационный пакет несёт информацию о коде производителя, типе устройства, коде нажатой кнопки и т.д. Пакеты повтора могут частично или полностью совпадать с информационным пакетом, копировать его биты с инверсией, или не нести никакой информации, представляя последовательность из нескольких одинаковых, для каждого пакета повтора, битов.

Длительность пауз между пакетами: обычно не превышает 200мс.

Протоколы передачи данных: определяют следующие, основные, параметры:

Несущая частота: у большинства пультов равна 38 кГц, именно на эту частоту настроен Trema ИК-приёмник.

Кодирование информации: это принцип передачи битов данных. Выделим три основных вида кодирования, при которых каждый бит передаётся последовательностью из одного импульса и одной паузы:

Сигналы Start, Stop и Toggle: по своему названию располагаются в начале, конце или середине пакета.

Stop: При кодировании длинной паузы, нельзя определить значение последнего бита в пакете, так как после пакета следует большая пауза, и последний бит будет всегда определяться как «1», поэтому в пакет добавляется сигнал Stop представляющий из себя импульс не несущий никакой информации.

Start: При бифазном кодировании требуется подать сигнал Start, так как невозможно начать передачу пакета с паузы.

Toggle: Это бит, который меняет своё значение при каждом новом нажатии на кнопку, используется в протоколах RS5, RS5X, RS6 (Philips), где пакеты повторов полностью повторяют данные информационного пакета. Таким образом принимающее устройство может отличить удержание кнопки от её повторного нажатия.

кодирование длиной импульсов - сначала передаётся импульс, длина которого зависит от значения передаваемого бита, затем следует пауза, длина которой не зависит от значения бита. Например: в протоколе SIRC (Sony), длина импульса для бита «1» = 1200мкс, а для бита «0» = 600мкс, длина пауз всегда равна 600мкс. Таким образом можно отличить «1» от «0» по длине импульса.

кодирование длиной пауз - сначала передаётся импульс, длина которого не зависит от значения передаваемого бита, затем следует пауза, длина которой зависит от значения бита. Например: в протоколе NEC, длина паузы для бита «1» = 1687,5мкс, а для бита «0» = 562,5мкс, длина импульсов всегда равна 562,5мкс. Таким образом можно отличить «1» от «0» по длине паузы.

бифазное кодирование - длина импульса равна длине паузы, а их последовательность определяет тип передаваемого бита. Например: в протоколе RS5 (Philips), для бита «1» импульс следует за паузой, а для бита «0» пауза следует за импульсом. Для протокола NRC (Nokia), наоборот, для бита «1» пауза следует за импульсом, а для бита «0» импульс следует за паузой.

Примеры:

Однократная передача данных:

```
#include <iarduino_IR_TX.h>                     // Подключаем библиотеку для работы с ИК-передатчиком
iarduino_IR_TX VD(10);                          // Объявляем объект VD, с указанием вывода к которому подключён ИК-передатчик
void setup(){
  VD.begin();                                   // Инициируем работу с ИК-передатчиком
  VD.send(0x00FFA25D);                          // Однократно отправляем код 0x00FFA25D, без пакетов повторов
}
void loop(){}                                   // Arduino отправит код 0x00FFA25D, сигнализируя о своём включении
```

Передача данных с пакетами повторов:

```
#include <iarduino_IR_TX.h>                      // Подключаем библиотеку для работы с ИК-передатчиком
iarduino_IR_TX VD(2);                            // Объявляем объект VD, с указанием вывода к которому подключён ИК-передатчик
void setup(){
  pinMode(3,INPUT);                              // Конфигурируем 3 вывод, к которому подключена кнопка, как вход
  pinMode(4,INPUT);                              // Конфигурируем 4 вывод, к которому подключена кнопка, как вход
  pinMode(5,INPUT);                              // Конфигурируем 5 вывод, к которому подключена кнопка, как вход
  VD.begin();                                    // Инициируем работу с ИК-передатчиком
}
void loop(){
  if(digitalRead(3)){VD.send(0x00FFA25D, true);} // Если нажата кнопка, подключённая к 3 выводу, то отправляем код 0x00FFA25D,
  if(digitalRead(4)){VD.send(0x00FF629D, true);} // Если нажата кнопка, подключённая к 4 выводу, то отправляем код 0x00FF629D,
  if(digitalRead(5)){VD.send(0x00FFE21D, true);} // Если нажата кнопка, подключённая к 5 выводу, то отправляем код 0x00FFE21D,
}
```

Передача данных с указанием протокола:

```
#include <iarduino_IR_TX.h>                      // Подключаем библиотеку для работы с ИК-передатчиком
iarduino_IR_TX VD(5);                            // Объявляем объект VD, с указанием вывода к которому подключён ИК-передатчик
void setup() {
  VD.begin();                                    // Инициируем работу с ИК-передатчиком
  pinMode(6,INPUT);                              // Конфигурируем 6 вывод, к которому подключена кнопка, как вход
  pinMode(7,INPUT);                              // Конфигурируем 7 вывод, к которому подключена кнопка, как вход
  pinMode(8,INPUT);                              // Конфигурируем 8 вывод, к которому подключена кнопка, как вход
  VD.protocol("AeQQV~zK]Kp^KJp[@@@@@@@Bp");      // Указываем протокол передачи данных от пульта ELENBERG
}                                                // Получить строку протокола, можно нажав любую кнопку пульта телевизора
                                                 // и вызвав одноименную функцию приёмника, без параметров
void loop(){
  if(digitalRead(4)){VD.send(0x417, true);}      // отправляем сигнал ON/OFF (с  пакетами повторов, пакеты повторяются через з
  if(digitalRead(5)){VD.send(0x425, true);}      // отправляем сигнал VOL-   (с  пакетами повторов, пакеты повторяются через з
  if(digitalRead(6)){VD.send(0x427      );}      // отправляем сигнал VOL+   (без пакетов повторов, громкость будет увеличиват
}
```

Данный пример показывает, как передатчик может полностью имитировать сигналы других ИК-пультов дистанционного управления.

В статье Wiki ИК-приёмник, описано, как получить строку протокола передачи данных ИК-пультов через ИК-приёмник и как получить коды кнопок, передаваемые ИК-пультами.

Полученную строку протокола, нужно передать в качестве параметра функции protocol(), после чего можно отправлять коды кнопок функцией send(). В результате, устройства будут реагировать на ИК-передатчик, как на собственный ИК-пульт.

Описание основных функций библиотеки:

Подключение библиотеки:

```
#include <iarduino_IR_TX.h>                    // Подключаем библиотеку, для работы с ИК-передатчиком.
iarduino_IR_TX VD(№_ВЫВОДА[,ИНВЕРСИЯ]);        // Объявляем объект VD, с указанием номера вывода, к которому подключён ИК-пе
                                               // Вторым параметром, типа bool, можно указать, что данные на передатчик треб
```

Функция begin();

- Назначение: инициализация работы с ИК-передатчиком
- Синтаксис: begin();
- Параметры: Нет.
- Возвращаемые значения: Нет.
- Примечание: Вызывается 1 раз в коде setup.
- Пример:

```
VD.begin(); // Инициируем работу с ИК-передатчиком
```

Функция send();

- Назначение: Передача данных.

- Синтаксис: send(ДАННЫЕ [, УДЕРЖАНИЕ]);
- Параметры:
  - ДАННЫЕ - код, типа uint32_t, который требуется передать;
  - УДЕРЖАНИЕ - необязательный параметр, типа bool - указывающий что необходимо передавать не только код, но и пакеты повторов. Параметр имеет смысл, если функция вызывается пока удерживается кнопка.
- Возвращаемые значения: Нет.
- Примечание: Если функция вызвана без параметра УДЕРЖАНИЕ, или он равен false, то функция, при каждом её вызове, однократно передаст указанный код. Если функция вызвана с параметром УДЕРЖАНИЕ равным true, то функция подавляет дребезг кнопки и отправляет пакеты повторов (с указанным в протоколе интервалом) при её удержании.
- Пример:

```
VD.send(0xCCDDEEFF);       // Отправляем код 0xCCDDEEFF. Если функцию вызывать постоянно, в цикле, то она каждый раз будет отп
VD.send(0xCCDDEEFF, true); // Отправляем код 0xCCDDEEFF. Если функцию вызывать постоянно, в цикле, то она отправит код только
```

Функция protocol();

- Назначение: Установка протокола передачи данных.
- Синтаксис: protocol(СТРОКА);
- Параметры:
  - СТРОКА - состоящая из 25 символов протокола + символ конца строки. Данную строку можно получить вызвав одноимённую функцию, без параметров, для приёмника.
- Возвращаемые значения: bool - строка содержит корректные данные о протоколе или нет.
- Примечание: Функция устанавливает протокол передачи данных, таким образом ИК-передатчик может имитировать сигналы обычных пультов. После вызова данной функции, передачи данных функцией send() будут осуществляться по новому протоколу. Протокол передачи данных по умолчанию, соответствует пульту «Car mp3».
- Пример:

```
VD.protocol("AeQQV~zK]Kp^KJp[@@@@@@@Bp"); // Указываем протокол передачи данных от пульта ELENBERG.
                                          // Теперь передатчик будет отправлять данные, кодируя их, в соответствии с указанным
```

```
                                        // Получить строку протокола, можно нажав любую кнопку пульта телевизора и вызвав од
```

Переменная frequency:

- Значение: Устанавливает несущую частоту передачи данных в кГц;
- Тип данных: uint8_t;
- Примечание: Если переменной не присваивать значение, то передача ведётся на частоте указанной в протоколе. Если указать значение 0, то данные будут передаваться без модуляции.

```
VD.frequency=36;     // Устанавливаем несущую частоту передачи данных в 36 кГц.
VD.send(0xCCDDEEFF); // Отправляем данные с несущей частотой 36 кГц.
VD.send(0xABCDEF);   // Отправляем данные с несущей частотой 36 кГц.
                     // Несущая частота будет изменена, если задать новое значение переменной frequency, или задать новый прот
```

Применение:

- управление роботами, движущимися, летающими и плавающими моделями, бытовой и специализированной техникой.
- включение/выключение освещения, обогрева, вентиляции, полива и т.д.
- открывание/закрывание дверей, жалюзи, мансардных окон, форточек и т.д.