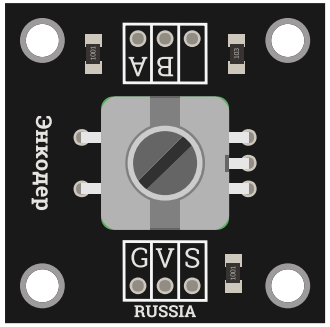
Энкодер (Trema-модуль)



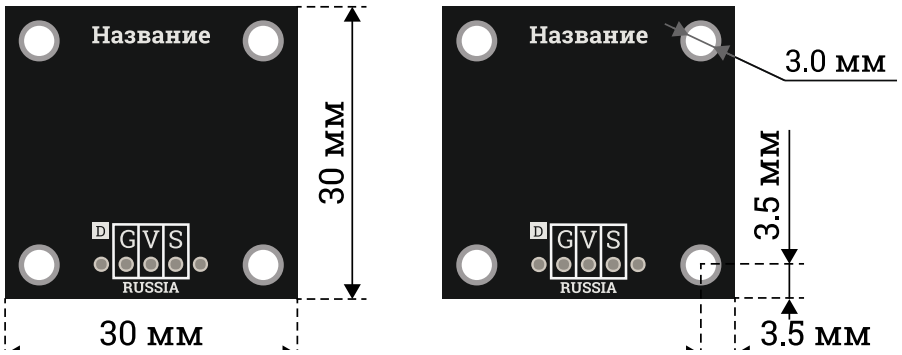
Общие сведения:

Trema-модуль Энкодер - это датчик угла поворота, позволяющий дискретно (прерывисто) определять угол поворота вала и нажатие на него. Основным элементом данного модуля является инкрементальный (пошаговый) энкодер с тактовой (тактильной) кнопкой.

Спецификация:

- Входное напряжение питания модуля: 5 В
- Ток потребляемый модулем: < 3 мА
- Допустимый ток на выходах модуля: < 10 мА
- Сопротивление контактов: < 3 Ом
- Количество циклов вращения: < 30'000
- Количество циклов переключения: < 20'000
- Рабочая температура: -30 ... 70 °C
- Длина вала: 12 мм
- Диаметр вала: Ø6 мм
- Габариты: 30x30x38 (с учётом колодки выводов)

Все модули линейки "Trema" выполнены в одном формате



Подключение:

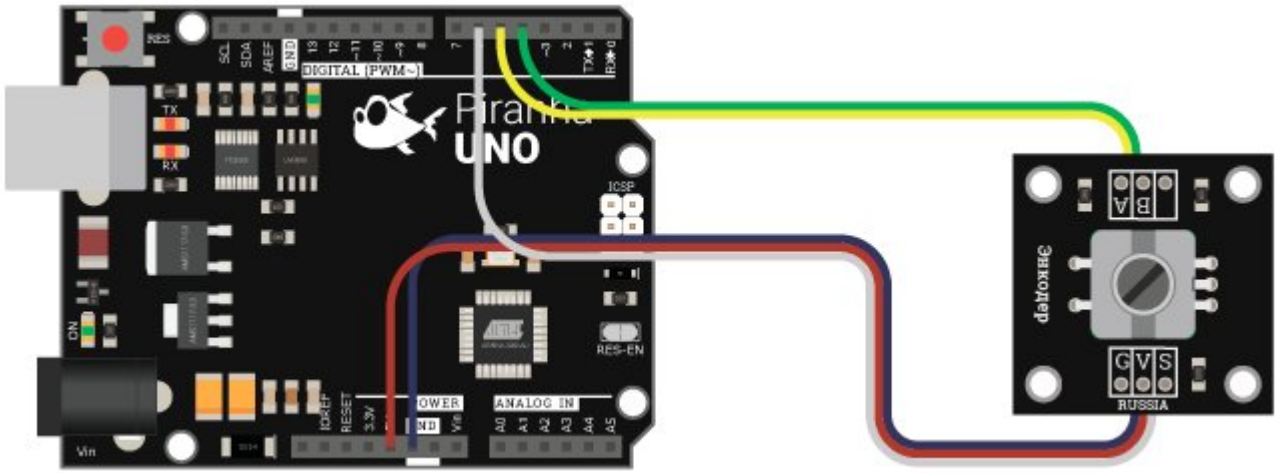
- Выводы A и B модуля являются выходами энкодера
- Вывод S (Signal) модуля является выходом тактовой кнопки
- Выводы V (Vcc) и G (GND) модуля являются входом питания

При использовании библиотеки iarduino_Encoder_tmr, все выходы модуля можно подключать к любым выводам Arduino, а к одной Arduino можно подключить до 8 модулей.

Модуль удобно подключать 2 способами, в зависимости от ситуации:

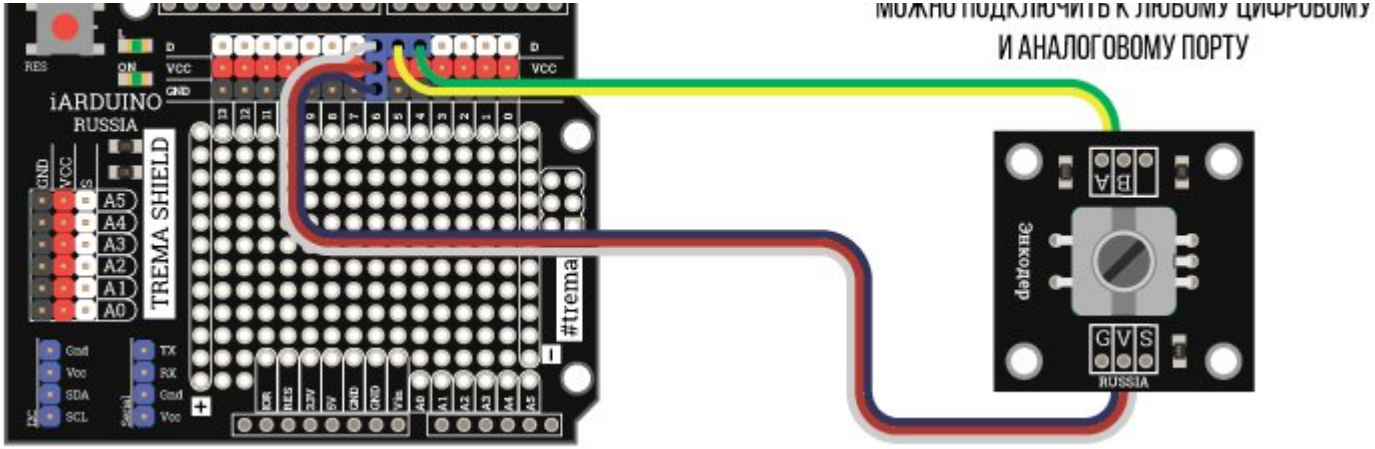
Способ - 1 : Используя проводной шлейф и Piranha UNO

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



Способ - 2 : Используя 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 5 В постоянного тока, подаётся на выводы V (Vcc) и G (GND) модуля.

Подробнее о модуле:

Полный оборот инкрементального энкодера (360°) разбит на дискретные участки (шаги), при прохождении которых энкодер формирует импульсы на своих выходах. Подсчитав количество импульсов, можно определить угол поворота относительно начального положения вала. У инкрементального энкодера имеется два выхода (третий вывод является общим входом, с которым импульсно соединяются выходы) и на обоих выходах формируется одинаковое количество импульсов, но в зависимости от направления поворота, импульсы на одном выводе будут опережать или отставать от импульсов на другом выводе (код Грея). Таким образом можно определить не только угол, но и направление поворота.

Полный оборот вала Trema-энкодера разбит на 20 дискретных участков по 18° на каждый, значит при полном обороте вала энкодера, на выходе каждого вывода модуля сформируется по 20 импульсов.

При неподвижном и не нажатом вале энкодера, на выходах модуля A и B присутствуют уровни логической «1», а на выходе S (Signal) уровень логического «0». При вращении вала энкодера на выходах A и B формируются отрицательные импульсы, а при нажатии на вал, на выходе S (Signal) устанавливается уровень логической «1».

Для работы с модулем предлагаем воспользоваться библиотекой iarduino_Encoder_tmr, которая позволяет работать с несколькими энкодерами используя второй аппаратный таймер. Библиотека постоянно считывает уровни сигналов на выходах A и B модулей, фиксируя наличие поворота и его направление. Получить состояние энкодера можно вызвав функцию библиотеки read, которая вернёт одно из трёх состояний: encLEFT(зафиксирован поворот влево), encRIGHT (зафиксирован поворот вправо), или false (повороты не зафиксированы).

ВАЖНО: библиотека использует **второй** аппаратный таймер,
НЕ ВЫВОДИТЕ СИГНАЛЫ ШИМ НА 3 ИЛИ 11 ВЫВОД!

Подробнее про установку библиотеки читайте в нашей инструкции..

Примеры:

Пример 1:

```
int i=enc.read();                              //  Читаем состояние энкодера в переменную i
if(i){                                         //  Если энкодер зафиксировал поворот (i!=false), то ...
    if( i==encLEFT  ){Serial.println("<");}    //  Если энкодер зафиксировал поворот влево,  выводим символ <
    if( i==encRIGHT ){Serial.println(">");}    //  Если энкодер зафиксировал поворот вправо, выводим символ >
}
```

Пример 2:

```
i = i + enc.read();  //  Если энкодер зафиксирует поворот, то значение переменной i изменится:
                     //  если был поворот влево, то переменная i уменьшится на 1, а если был поворот вправо, то увеличится на
                     //  Единственную строку данного примера можно записать еще короче: i+=enc.read();
```

Вывод состояния энкодера:

```
#include <iarduino_Encoder_tmr.h>              //  Подключаем библиотеку iarduino_Encoder_tmr для работы с энкодерами через апг
iarduino_Encoder_tmr enc(11,12);               //  Объявляем объект enc для работы с энкодером указывая (№ вывода A, № вывода B
```

```
                                        //  Если при объявлении объектов перепутать выводы, то поворот влево будет расце
void setup(){
    Serial.begin(9600);                 //  Инициируем передачу данных в монитор последовательного порта
    enc.begin();                        //  Инициируем работу с энкодером
}
void loop(){
    int i=enc.read();                   //  Читаем состояние энкодера в переменную i
    if(i){                              //  Если энкодер зафиксировал поворот, то ...
        if(i==encLEFT ){Serial.println("<");} //  Если энкодер зафиксировал поворот влево,  выводим символ <
        if(i==encRIGHT){Serial.println(">");} //  Если энкодер зафиксировал поворот вправо, выводим символ >
    }
}
```

При каждом повороте энкодера, в мониторе будет отображаться его направление.

Вывод счетчика энкодера:

```
#include <iarduino_Encoder_tmr.h>       //  Подключаем библиотеку iarduino_Encoder_tmr для работы с энкодерами через апп
iarduino_Encoder_tmr enc(11,12);        //  Объявляем объект enc для работы с энкодером указывая (№ вывода A, № вывода B
int    n = 0;                           //  Определяем переменную n для подсчёта дискретных поворотов энкодера
void setup(){
    Serial.begin(9600);                 //  Инициируем передачу данных в монитор последовательного порта
    enc.begin();                        //  Инициируем работу с энкодером
}
void loop(){
    int i=enc.read();                   //  Читаем состояние энкодера в переменную i
    if(i){                              //  Если энкодер зафиксировал поворот, то ...
        n=n+i;   /* n+=i*/              //  Меняем значение счётчика n, т.к. в переменной i находится -1 (при повороте в
        Serial.println(n);              //  Выводим значение счётчика n
    }
}
```

При повороте энкодера в мониторе будет отображаться значение счётчика, при повороте влево счётчик будет уменьшаться, а при повороте вправо - увеличиваться.

Описание основных функций библиотеки:

Подключение библиотеки:

```
#include <iarduino_Encoder_tmr.h>                  //  Подключаем библиотеку iarduino_Encoder_tmr для работы с энкодерами чере
iarduino_Encoder_tmr enc(№_вывода_A, №_вывода_B);  //  Объявляем объект enc для работы с энкодером (можно использовать любые в
```

Можно объявить несколько объектов, тогда каждый объект будет работать со своим энкодером.
Имена объектов должны отличаться. Библиотека позволяет подключить до 8 энкодеров.

Функция begin();

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

```
void setup(){
    enc.begin();  //  Инициируем работу с энкодером
}
```

Функция read();

- Назначение: Чтение состояния энкодера.
- Синтаксис: read();

- Параметры: Нет.
- Возвращаемые значения: int8_t
  - encLEFT - зафиксирован поворот влево.
  - encRIGHT - зафиксирован поворот вправо.
  - false - повороты не зафиксированы.
- Примечание:
  - Если повернуть энкодер и вызвать функцию read(), она вернёт зафиксированное состояние, а на последующие вызовы функция будет возвращать false до тех пор, пока опять не будет зафиксирован поворот.
  - Если повернуть энкодер, а функцию read() вызвать через пол года, она вернёт зафиксированное состояние.
  - Если повернуть энкодер в разные стороны, а потом вызвать функцию read(), она вернёт последнее зафиксированное состояние.
  - Возвращаемые значения можно использовать для увеличения или уменьшения Ваших переменных:
    encLEFT = -1, encRIGHT = 1, false = 0.

Применение:

- Роботостроение
- Бытовая техника (стиральные машины, микроволновые печи и т.д.)
- Аудиосистемы

Ссылки:

- Библиотека iarduino_Encoder_tmr, позволяющая работать с энкодерами (до 8 шт.)
- Wiki - Установка библиотек в Arduino IDE