

Энкодер с эффектом памяти (Трета-модуль v2.0)



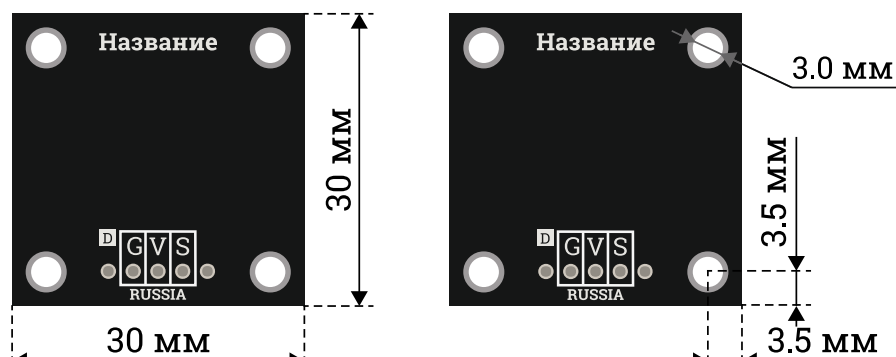
Общие сведения:

[Трета-модуль Энкодер с памятью](#) - это датчик угла поворота, позволяющий дискретно (прерывисто) определять угол поворота вала, его направление и нажатие на вал. Но в отличие от обычных энкодеров, данный модуль сам определяет факт и направление каждого шага поворота вала.

Спецификация:

- Входное напряжение питания модуля: 5 В
- Ток потребляемый модулем: параметр проходит тестирование
- Допустимый ток на выходах модуля: параметр проходит тестирование
- Количество циклов вращения: < 30'000
- Количество циклов переключения: < 20'000
- Рабочая температура: -30 ... 70 °С
- Длина вала: 12 мм
- Диаметр вала: Ø6 мм
- Габариты: 30x30 мм

Все модули линейки "Тема" выполнены в одном формате



Подключение:

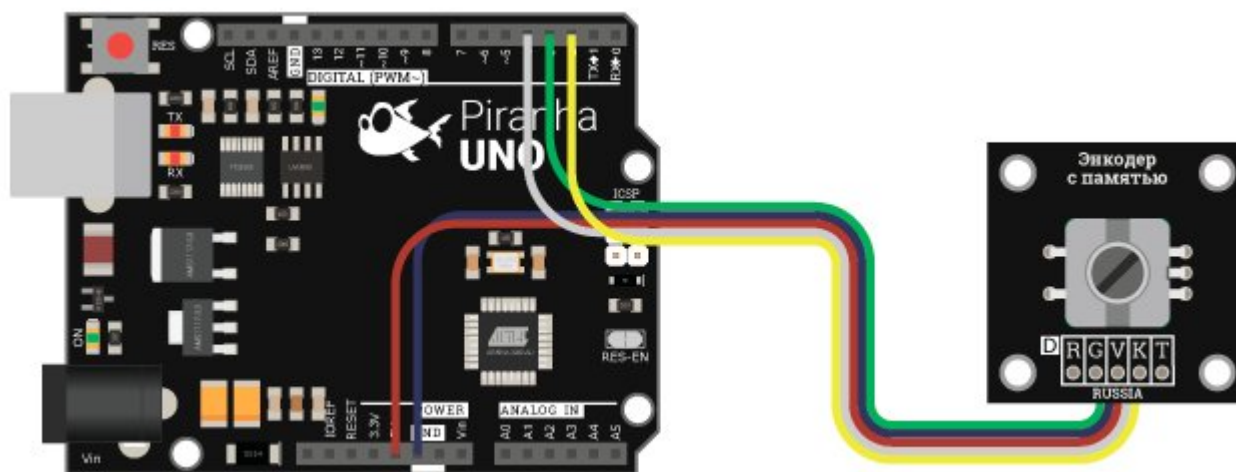
- Вывод «K» (Key) - выход кнопки модуля.
- Вывод «R» (Route) - выход направления поворота вала модуля.
- Вывод «T» (Turn) - двунаправленный вывод наличия поворота вала модуля.
- Выводы «V» (Vcc) и «G» (GND) - входы питания модуля.

Выводы «R», «T» и «K» модуля можно подключать к любым выводам [Arduino](#).

Модуль удобно подключать 3 способами, в зависимости от ситуации:

Способ - 1 : Используя проводной шлейф и Piranha UNO

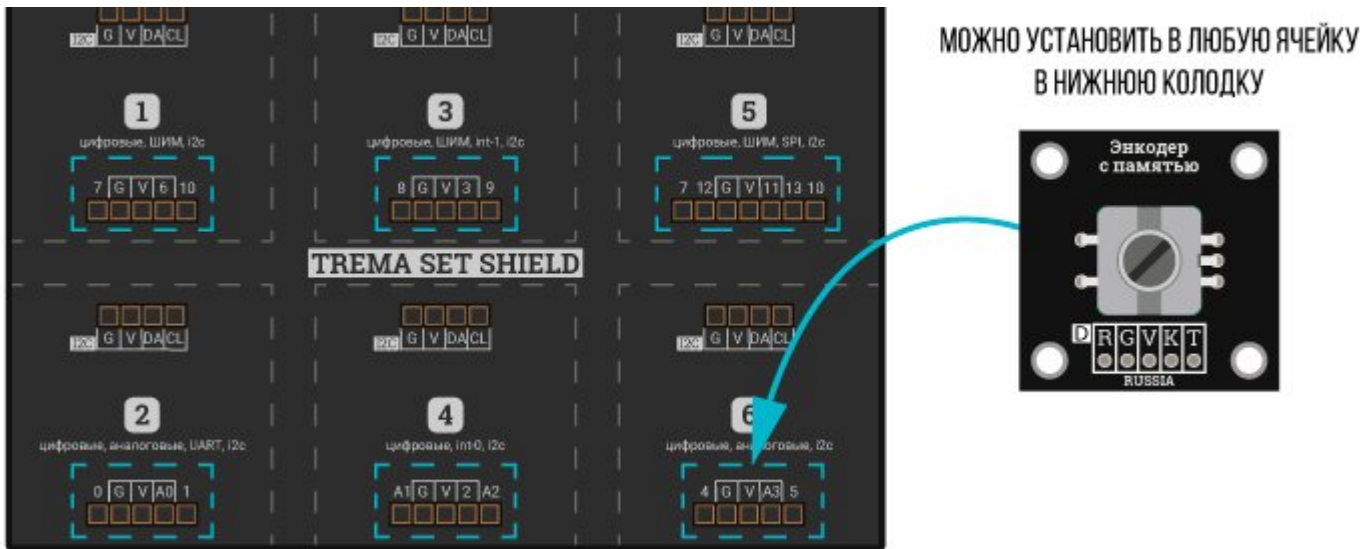
Используя провода «Папа – Мама», подключаем напрямую к контроллеру Piranha UNO.



Способ - 2 : Используя Trema Set Shield

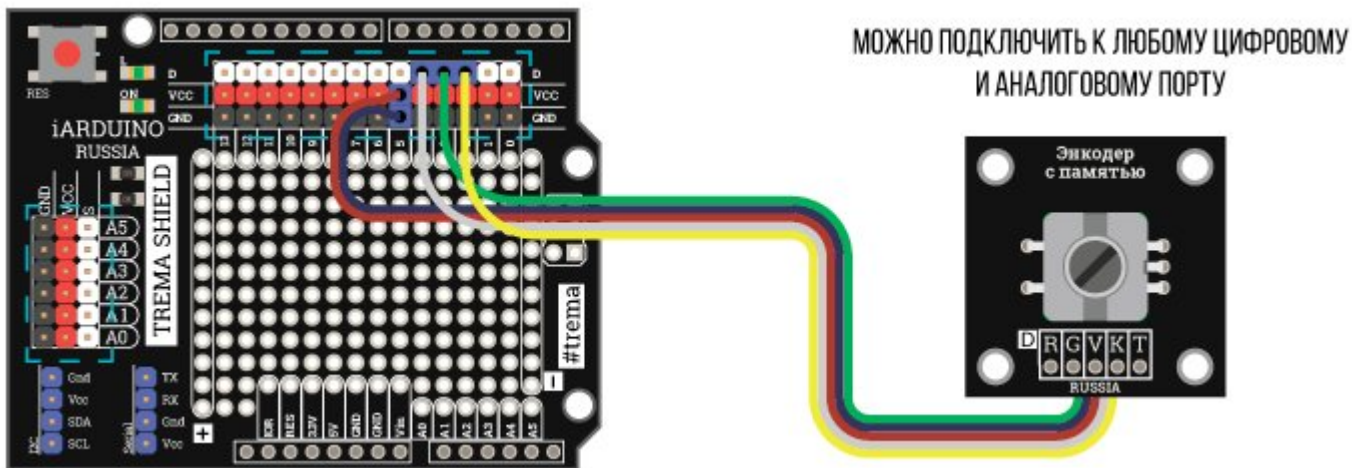
Модуль можно подключить к любому из цифровых или аналоговых входов Trema Set Shield.





Способ - 3 : Используя проводной шлейф и Shield

Используя 5-и 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:

Основным элементом данного модуля является инкрементальный (пошаговый) энкодер, с описанием работы которого можно ознакомиться в разделе [Wiki - энкодер](#). Полный оборот вала энкодера разбит на 20 участков (шагов) по 18° на каждый. Данные с энкодера поступают на схему модуля, которая определяет наличие каждого шага поворота вала и его направление, устанавливая (и сохраняя) определённые логические уровни на выводах «Т» (Turn - поворот) и «R» (Route - направление) модуля.

- **Вывод «R» (Route)** - является выходом модуля указывающим направление поворота вала энкодера. На нём устанавливается уровень логического «0» если вал повернули на шаг по часовой стрелке, или уровень логической «1» если вал повернули на шаг против часовой стрелки. Установленный логический уровень сохраняется на выводе до следующего поворота вала энкодера.
- **Вывод «Т» (Turn)** - является двунаправленным выводом модуля указывающим наличие поворота вала энкодера, вне зависимости от его направления. На нём устанавливается уровень логической «1» при каждом повороте вала на 1 шаг. Логическая «1» сохраняется на выводе до тех пор, пока на этот же вывод не подать уровень логического «0» (тогда модуль "понимает" что данные считаны и установит очередную «1» на следующем шаге поворота вала).
- **Вывод «К» (Key)** - является выходом тактовой кнопки. На нём устанавливается уровень логической «1» при нажатии на вал энкодера, установленный уровень сбрасывается в «0» при отпуске вала.

Примеры:

Вывод в монитор последовательного порта числа, которое меняется при вращении энкодера.

Для чтения данных с модуля используются обычные функции `digitalRead` без использования библиотек и таймеров.

```
const uint8_t pinT = 10;           // Определяем константу указывая ей номер вывода Arduino к которому подключён вывод T
const uint8_t pinR = 7;           // Определяем константу указывая ей номер вывода Arduino к которому подключён вывод R
const uint8_t pinK = 6;           // Определяем константу указывая ей номер вывода Arduino к которому подключён вывод K
int i = 0;                         // Определяем переменную значение которой будет меняться при повороте вала энкодера с
                                   //
void setup(){                       //
    pinMode(pinT, INPUT);           // Устанавливаем режим работы вывода pinT Arduino как вход
    pinMode(pinR, INPUT);           // Устанавливаем режим работы вывода pinR Arduino как вход
```

```

pinMode(pinK, INPUT);           // Устанавливаем режим работы вывода pinK Arduino как вход
Serial.begin(9600);             // Инициуем передачу данных в монитор последовательного порта на скорости 9600 бит/с
}                                 //
//
void loop(){                     //
  if(digitalRead(pinT)){        // Если вывод pinT установлен в 1, значит произошёл поворот вала энкодера с памятью на
    if(digitalRead(pinR)){i--;} // Если вывод pinR установлен в 1, то уменьшаем значение переменной i
    else {i++;}                 // Если вывод pinR сброшен в 0, то увеличиваем значение переменной i
    delay(2);                   // Подавляем дребезг контактов
    pinMode(pinT, OUTPUT);      // Устанавливаем режим работы вывода pinT Arduino как выход
    digitalWrite(pinT, 0);      // Сбрасываем состояние вывода pinT в логический 0
    pinMode(pinT, INPUT);       // Устанавливаем режим работы вывода pinT Arduino как вход
    Serial.println(i);          // Выводим значение переменной i в монитор последовательного порта
  }                               //
  if(digitalRead(pinK)){        // Если вывод pinK установлен в 1, значит произошло нажатие на вал энкодера
    Serial.println("Key");      // Выводим текст "Key" в монитор последовательного порта
  }                               //
}                                 //

```

Применение:

- Элементы управления, меню;