

3D-джойстик (Трема-модуль v2.0)



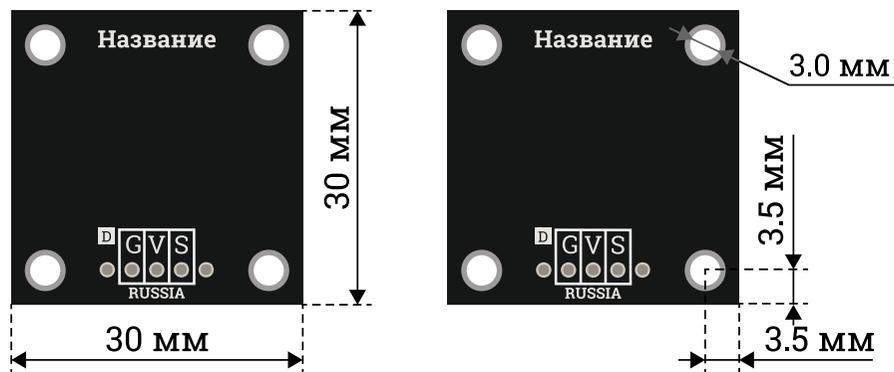
Общие сведения:

[Трема-модуль 3D-джойстик](#) - является модулем ввода данных. С его помощью можно управлять роботами, манипуляторами, станками, различными моделями (машинки, танки, самолёты, вертолёты, квадрокоптеры, лодки и т.д.), а также использовать для создания игровых приставок, выбора пунктов меню на дисплеях и индикаторах, ввода значений, и т.д. Джойстик можно не только перемещать по осям X и Y, но и нажимать на него.

Спецификация:

- Напряжение питания: 5 В / 3,3 В (оба напряжения входят в диапазон допустимых значений).
- Потребляемый ток: < 10 мА
- Габариты: 30x30 мм

Все модули линейки "Трема" выполнены в одном формате



Подключение:

- Выводы «X» и «Y» модуля подключается к любым аналоговым входам [Arduino](#). Значения, считываемые с этих выводов, растут при перемещении джойстика слева на право и снизу вверх.

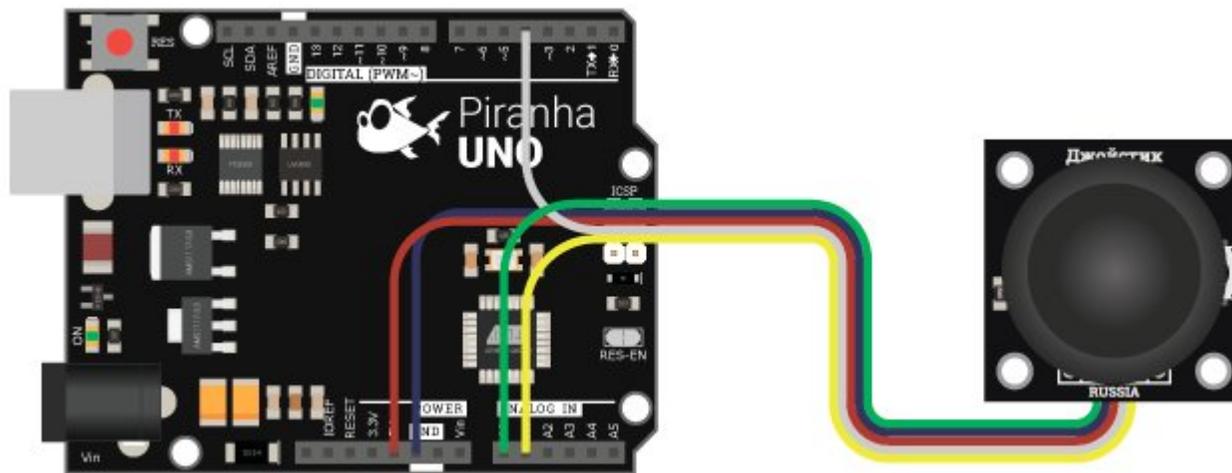
- Вывод «К» является цифровым и подключается к любому выводу [Arduino](#). В обычном состоянии на нём уровень логического «0», а при нажатии на джойстик, он меняется на логическую «1».
- Выводы «V» и «G» являются выводами питания.

[Схема установки модуля](#) при его подключении через [Trema Set Shield](#).

Модуль удобно подключать 3 способами, в зависимости от ситуации:

Способ - 1 : Используя проводной шлейф и Piranha UNO

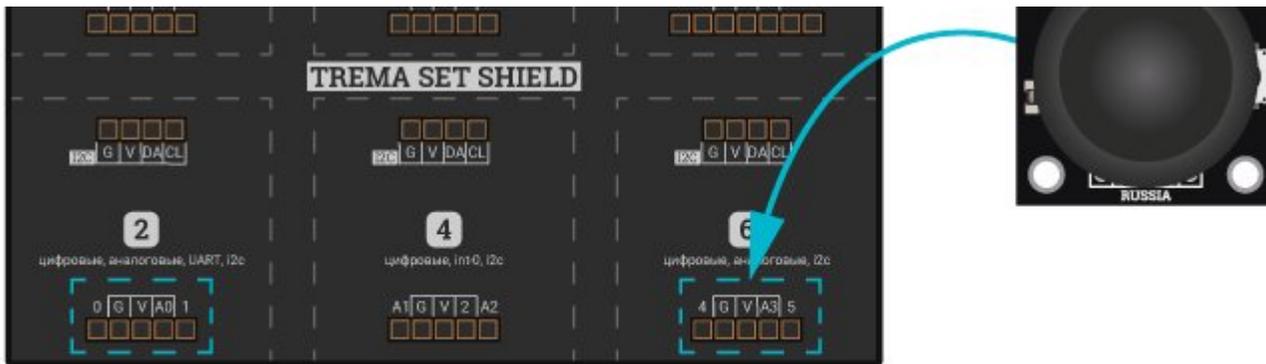
Используя провода «Папа – Мама», подключаем напрямую к контроллеру Piranha UNO



Способ - 2 : Используя Trema Set Shield

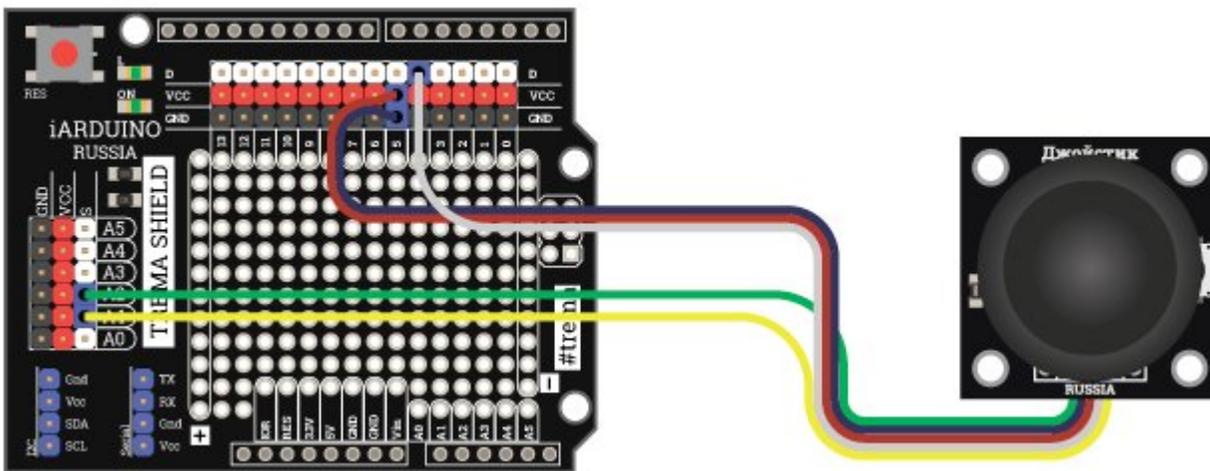
Модуль можно подключить к любому из аналоговых входов Trema Set Shield.





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 или 3,3 В постоянного тока, подаётся на выводы Vcc (V) и GND (G).

Подробнее о модуле:

Данные модуля считываются с двух потенциометров и тактовой кнопки, механически связанных с рычагом джойстика. Кнопка подключена

в разрыв питания Vcc и выхода «К», который прижат к GND через резистор. Следовательно, на выходе «К» может устанавливаться только два состояния: логический «0» (кнопка отпущена) или «1» (кнопка нажата). Выводы координат «X» и «Y» являются аналоговыми выходами модуля, они подключены к потенциометрам так, что напряжение снимаемое между этими выводами и GND растёт при перемещении джойстика слева на право и снизу вверх.

Примеры:

Определение положения джойстика и включение светодиода по нажатию кнопки

```
const int8_t Xaxis = A0;           // Определяем номер вывода, к которому подключен контакт оси X джойстика
const int8_t Yaxis = A1;           // Определяем номер вывода, к которому подключен контакт оси Y джойстика
const int8_t Button = 2;           // Определяем номер вывода, к которому подключен контакт кнопки джойстика
const int8_t LED = 7;              // Определяем номер вывода, к которому подключен светодиод
uint16_t XborderMIN = 505;         // Задаём границу значений, НИЖЕ которой будет считаться, что джойстик отклонён по с
uint16_t XborderMAX = 515;         // Задаём границу значений, ВЫШЕ которой будет считаться, что джойстик отклонён по с
uint16_t YborderMIN = 505;         // Задаём границу значений, НИЖЕ которой будет считаться, что джойстик отклонён по с
uint16_t YborderMAX = 515;         // Задаём границу значений, ВЫШЕ которой будет считаться, что джойстик отклонён по с
uint16_t Xvol = 0, Yvol = 0;       // Задаём переменные, которые будут принимать значения, считанные с осей джойстика

void setup() {
  Serial.begin(9600);              // Инициуруем передачу данных в монитор последовательного порта
  pinMode(LED, OUTPUT);            // Настраиваем вывод LED на работу в режиме выхода
  pinMode(Button, INPUT);          // Настраиваем вывод Button на работу в режиме входа
}

void loop() {
  Xvol = analogRead(Xaxis);        // Считываем значения оси X
  Yvol = analogRead(Yaxis);        // Считываем значения оси Y
  if (Xvol < XborderMIN) {         // Проверяем, полученное значение X меньше нижней границы центрального положения или
    if (Yvol < YborderMIN) {       // проверяем, полученное значение Y меньше нижней границы центрального положения или
      Serial.println("Left-Down"); // значит джойстик находится в положении ВЛЕВО-ВНИЗ
    }
  }
  else if (Yvol > YborderMAX) {    // Если же полученное значение Y больше верхней границы центрального положения, то
    Serial.println("Left-Up");     // значит джойстик находится в положении ВЛЕВО-ВВЕРХ
  }
}
```

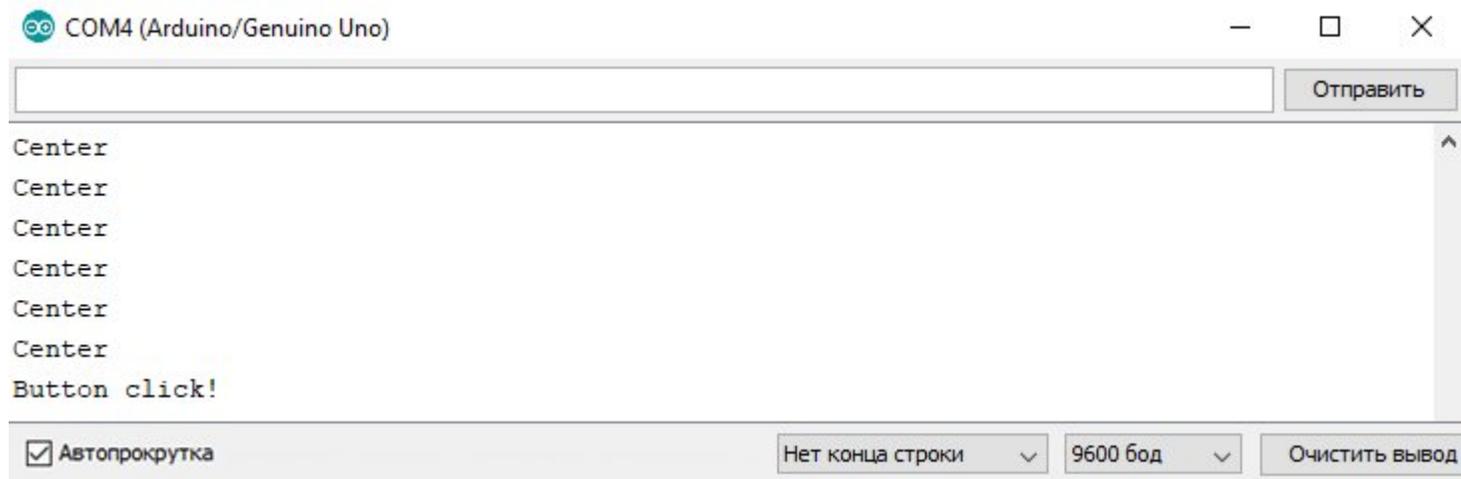
```

}
else {
    Serial.println("Left");           // Если же полученное значение У входит в границы центрального положения по оси У, э
}
}
else if (Xvol > XborderMAX) {        // Проверяем, полученное значение Х больше верхней границы центрального положения или
    if (Yvol < YborderMIN) {         // проверяем, полученное значение У меньше нижней границы центрального положения или
        Serial.println("Right-Down"); // значит джойстик находится в положении ВПРАВО-ВНИЗ
    }
    else if (Yvol > YborderMAX) {     // Если же полученное значение У больше верхней границы центрального положения, то
        Serial.println("Right-Up");  // значит джойстик находится в положении ВПРАВО-ВВЕРХ
    }
    else {
        Serial.println("Right");     // Если же полученное значение У входит в границы центрального положения по оси У, э
    }
}
else {                                // Если полученное значение Х входит в границы центрального положения по оси Х, знач
    if (Yvol < YborderMIN) {         // проверяем, полученное значение У меньше нижней границы центрального положения или
        Serial.println("Down");      // значит джойстик находится в положении ВНИЗ
    }
    else if (Yvol > YborderMAX) {     // Если же полученное значение У больше верхней границы центрального положения, то
        Serial.println("Up");        // значит джойстик находится в положении ВВЕРХ
    }
    else {
        Serial.println("Center");    // Если же полученное значение У входит в границы центрального положения по оси У, э
    }
}
if (digitalRead(Button)) {           // Проверяем, нажата ли кнопка
    delay(1);                         // Если кнопка была нажата, то подавляем дребезг
    digitalWrite(LED, !digitalRead(LED)); // и меняем состояние на выходе светодиода
    Serial.println("Button click!");  // Выводим текст о том, что кнопка была нажата
    while (digitalRead(Button)) {}    // Если кнопка удерживается, то ничего не делаем
}

```

```
delay(10);           // Если кнопка отпускается, то подавляем дребезг
}
}
```

В мониторе последовательного порта вы увидите:



Применение:

- Управление, джойстики, контроллеры;