

УРОК 8

Программирование ARDUINO

Никитин
Владимир
Николаевич

Что мы уже знаем

- Что такое переменная (переменные типа int, float, boolean)
- Функция ? `void rgb (?, ?, ?) { }`
- !
- &&
- `map ()`
- % - что это?

```
#include <SoftwareSerial.h>
#include <Time.h>

int data = 0;           // Переменная для хранения принятых данных

int gLedPin = 4;
int gRxPin = 2;
int gTxPin = 3;

SoftwareSerial BTSerial(gRxPin, gTxPin); // функция подключения блютуз сериал

void setup() {
  pinMode(4, OUTPUT);
  pinMode(7, OUTPUT);
  //# 38400 - для метода №1, 9600 - для метода №2
  BTSerial.begin(38400); // открываем блютуз порт
  Serial.begin(9600);   // открываем монитор порта
  delay(500);
}
```

```
void loop() {  
  
  if (BTSerial.available()) { // available() функция получает количество байт(символов) доступных для чтения из последовательного интерфейса связи.  
    //Это те байты которые уже поступили и записаны в буфер последовательного порта.  
    data = BTSerial.read(); // запишем в переменную из блютуз порта  
    Serial.write(data);  
    if(data == '1') { // Если data равна 1  
      digitalWrite(4, HIGH);  
    } // Включаем светодиод  
    else if(data == '0') { // Если data равна 0  
      digitalWrite(4, LOW); // Выключаем светодиод  
    }  
    if(data == '7') { // Если data равна 1  
      digitalWrite(7, HIGH);  
    } // Включаем светодиод  
    else if(data == '4') { // Если data равна 0  
      digitalWrite(7, LOW); // Выключаем светодиод  
    }  
  }  
}
```

```
void loop() {  
  if (BTSerial.available()) {  
    int duration, cm; // назначаем переменную "cm" и "duration" для показаний датчика  
    digitalWrite(trigPin, LOW); // изначально датчик не посылает сигнал  
    delayMicroseconds(2); // ставим задержку в 2 микросекунд  
  
    digitalWrite(trigPin, HIGH); // посылает сигнал  
    delayMicroseconds(10); // ставим задержку в 10 микросекунд  
    digitalWrite(trigPin, LOW); // выключаем сигнал  
  
    duration = pulseIn(echoPin, HIGH); // включаем прием сигнала  
  
    cm = duration / 58; // вычисляем расстояние в сантиметрах  
    Serial.println(cm); // выводим расстояние в сантиметрах  
    BTSerial.print(cm); // пишем в software-порт  
    BTSerial.println(" cm");  
    delay(1000); // ставим паузу в 1 секунду  
  }  
}
```

- В первом скетче мы рассмотрим как управлять сервоприводом на Arduino с помощью команды `myservo.write(0)`. Также мы будем использовать стандартную библиотеку `Servo.h`. Подключите сервомашинку к плате Ардуино, согласно схеме на фото выше и загрузите готовый скетч. В процедуре `void loop()` мы будем просто задавать для сервопривода необходимый угол поворота и время ожидания до следующего поворота.

Подключение сервопривода к Ардуино



черный - GND
красный - 5V
желтый - аналоговый выход

Схема подключения сервопривода к Ардуино UNO

```
#include <Servo.h> // подключаем библиотеку для работы с сервоприводом

Servo servo1; // объявляем переменную servo типа "servo1"

void setup() {
  servo1.attach(11); // привязываем сервопривод к аналоговому выходу 11
}

void loop() {
  servo1.write(0); // ставим угол поворота под 0
  delay(2000); // ждем 2 секунды

  servo1.write(90); // ставим угол поворота под 90
  delay(2000); // ждем 2 секунды

  servo1.write(180); // ставим угол поворота под 180
  delay(2000); // ждем 2 секунды
}
```

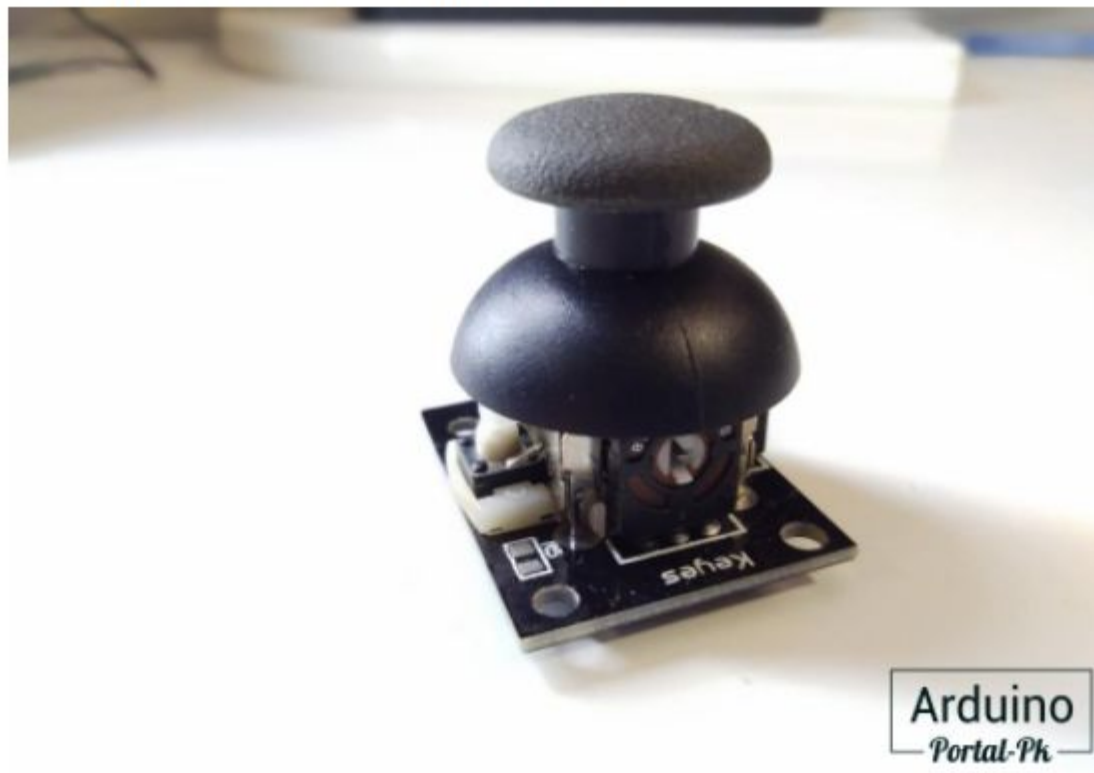

Пояснения к коду:

- 1 переменная `Servo` необходима, чтобы не запутаться при подключении нескольких сервоприводов к Ардуино. Мы назначаем каждому приводу свое имя;
- 2 команда `servo1.attach(10)` привязывает привод к цифровому выходу 10.
- 3 в программе мы вращаем привод на 0-90-180 градусов и возвращаем в начальное положение, поскольку [процедура void loop](#) повторяется циклично.

Подключаем джойстик

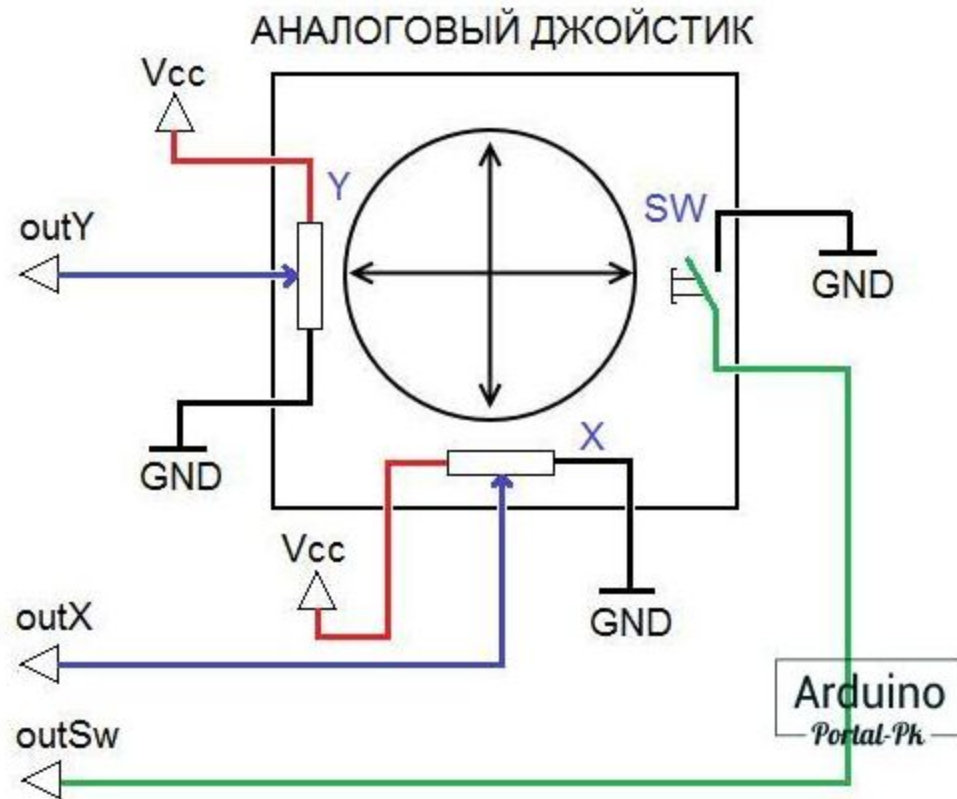
Представленная на рисунке ниже схема подключения может вас немного смутить. Но поверьте, все не так сложно как кажется на первый взгляд. Опять таки для подключения используем монтажную плату.

Устройство аналогово джойстика и принцип работы.



Arduino
Portal-Pk

Джойстик представляет из себя модуль, на который установлены 2 потенциометра и одна тактовая кнопка.



Управляет всем стик. При изменении положения стика по оси X - вращается потенциометр, выход которого *outX*, а при перемещении по оси Y - меняется значение потенциометра с выводом *outY*. По сути, это 2 потенциометра, с которых снимаем показания. Стик устроен таким образом, что он возвращается самостоятельно в центральное положение. Что обеспечивает центрование потенциометров. В данном состоянии джойстика на выходах *outX*, *outY* будем 511.

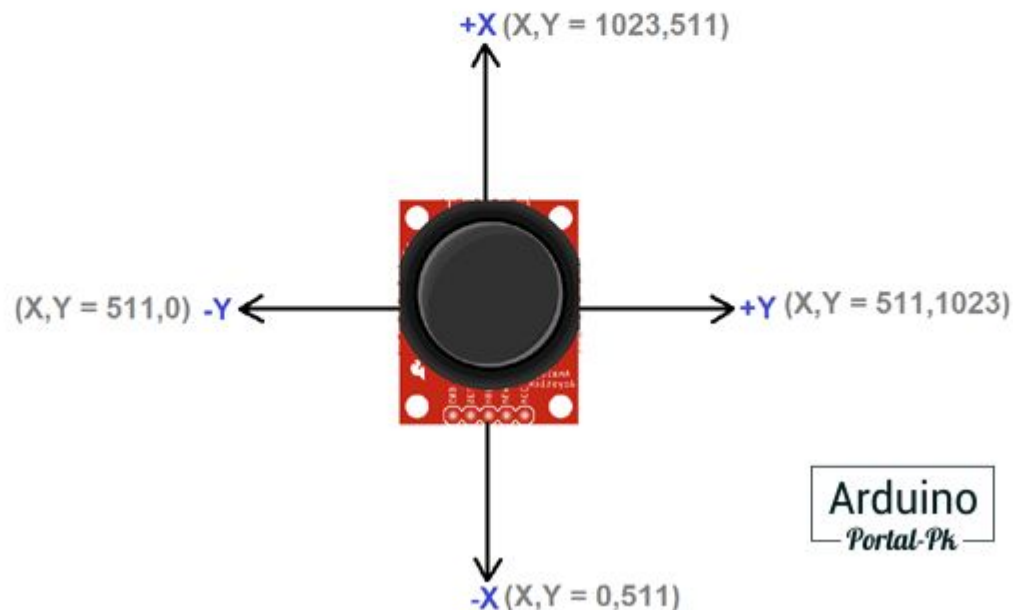
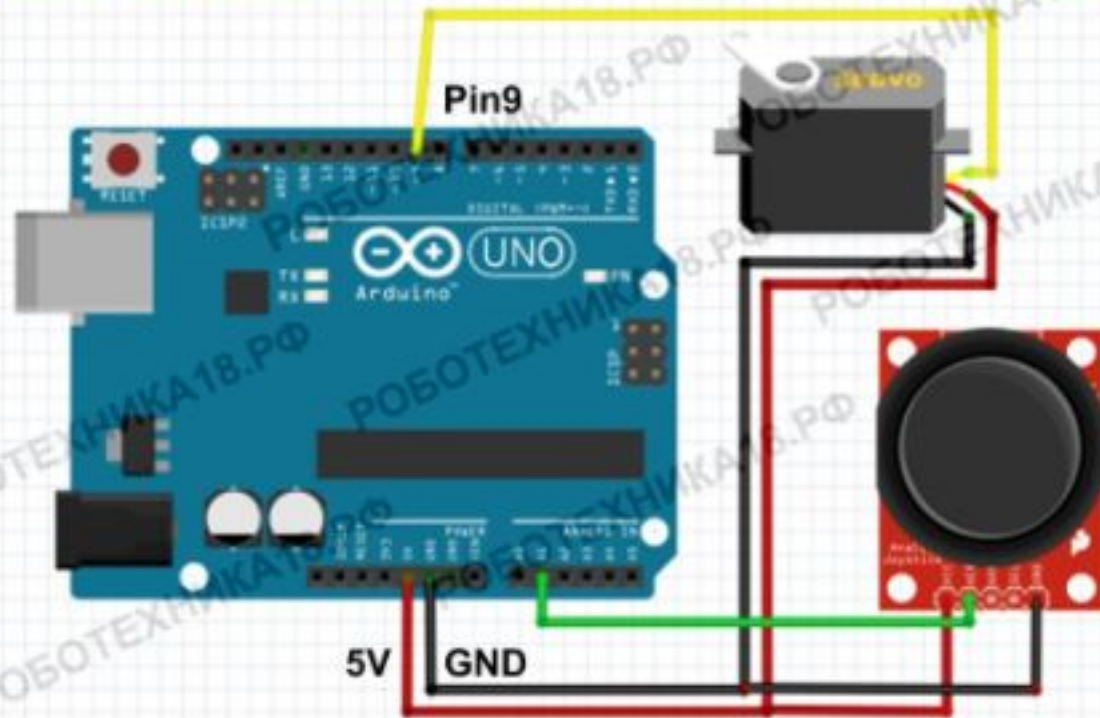


Схема подключения серво и джойстика к Ардуино



```
#define pinX A1          // ось X джойстика

#include <Servo.h> // подключаем библиотеку для серво
Servo servo1;      // объявляем объект servo1

void setup() {
  pinMode(pinX, INPUT); // указываем пин для джойстика

  servo1.attach(9);      // указываем пин для микро серво
}

void loop() {
  int X = analogRead(pinX); // считываем значение оси X

  X = map(X, 0, 1023, 0, 180); // переводим значение X в новый диапазон

  servo1.write(X);          // поворачиваем серво от джойстика
}
```

Схема подключения джойстика к Arduino UNO

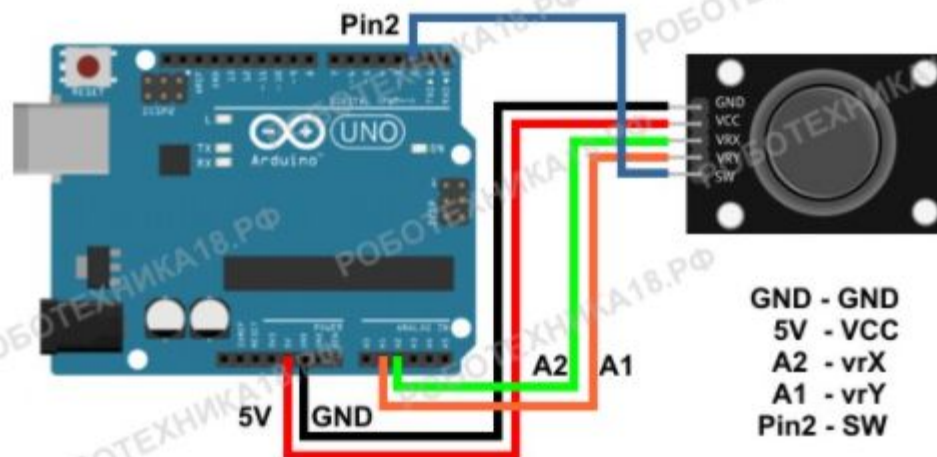


Схема подключения аналогового джойстика к Ардуино Уно


```

#define pinX    A2 // ось X джойстика
#define pinY    A1 // ось Y джойстика
#define swPin   2 // кнопка джойстика
#define ledPin  13 // светодиод на Pin 13

void setup() {
    Serial.begin(9600);

    pinMode(ledPin, OUTPUT);
    pinMode(pinX, INPUT);
    pinMode(pinY, INPUT);

    pinMode(swPin, INPUT);
    digitalWrite(swPin, HIGH);
}

void loop() {
    boolean ledState = digitalRead(swPin); // считываем состояние кнопки
    digitalWrite(ledPin, ledState);      // вкл./выкл. светодиод

    int X = analogRead(pinX);            // считываем значение оси X
    int Y = analogRead(pinY);            // считываем значение оси Y

    Serial.print(X);                     // выводим в Serial Monitor
    Serial.print("\t");                   // табуляция
    Serial.println(Y);
}

```

Схема подключения двух серво и джойстика к Ардуино

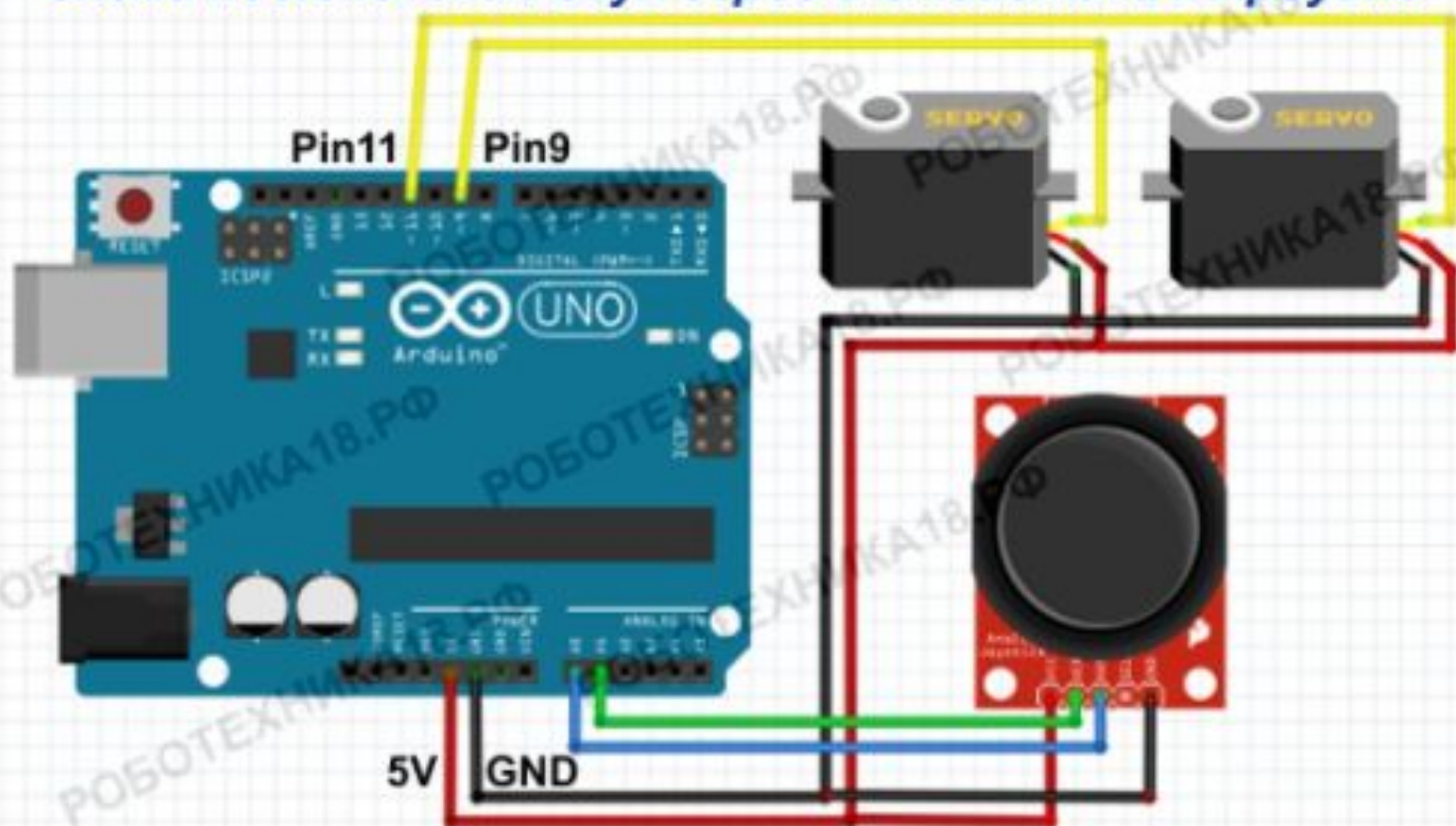


Схема подключения двух сервоприводов к Ардуино с джойстиком

```
#define pinX A1      // ось X джойстика
#define pinY A0      // ось Y джойстика
#define swPin  2     // кнопка джойстика
#define ledPin 13    // светодиод на Pin 13

#include <Servo.h> // подключаем библиотеку для серво
Servo servo1;     // объявляем объект servo1
Servo servo2;     // объявляем объект servo2

void setup() {
```

```
void setup() {  
  Serial.begin(9600);  
  
  pinMode(pinX, INPUT); // указываем пин для джойстика  
  pinMode(pinY, INPUT); // указываем пин для джойстика  
  
  servo1.attach(9);           // указываем пин для первого серво  
  servo2.attach(8);          // указываем пин для второго серво  
  pinMode(swPin, INPUT);  
  digitalWrite(swPin, HIGH);  
}
```

```
void loop() {
  boolean ledState = digitalRead(swPin); // считываем состояние кнопки
  digitalWrite(ledPin, ledState);      // вкл./выкл. светодиод

  int X = analogRead(pinX);           // считываем значение оси X
  int Y = analogRead(pinY);           // считываем значение оси Y

  X = map(X, 0, 1023, 0, 180); // переводим значение X в новый диапазон
  Y = map(Y, 0, 1023, 0, 180); // переводим значение Y в новый диапазон

  Serial.print(X);                   // выводим в Serial Monitor
  Serial.print("\t");                 // табуляция
  Serial.println(Y);

  servo1.write(X);                   // поворачиваем первый серво
  servo2.write(Y);                   // поворачиваем второй серво
  delay(1000);
}
```

Home work

- Write a sketch, than data is send device with Blurtooth