

Part 2

Lesson

12

**Analog Joystick
Module**

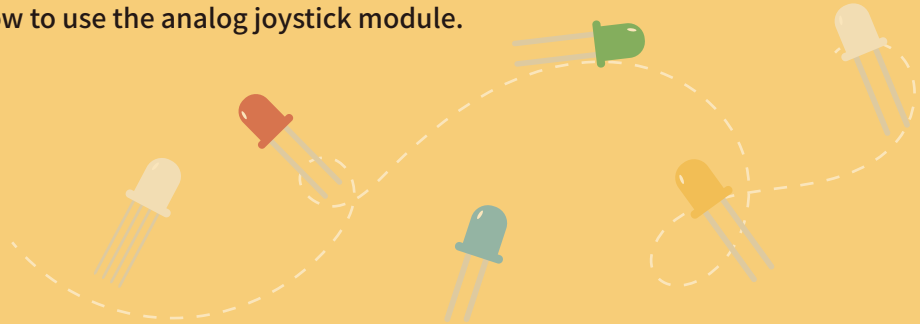
Overview

Analog joysticks are a great way to add some control in your projects.

In this tutorial we will learn how to use the analog joystick module.

Component Required:

- (1) x Elegoo Uno R3
- (1) x Joystick module
- (5) x F-M wires (Female to Male DuPont wires)

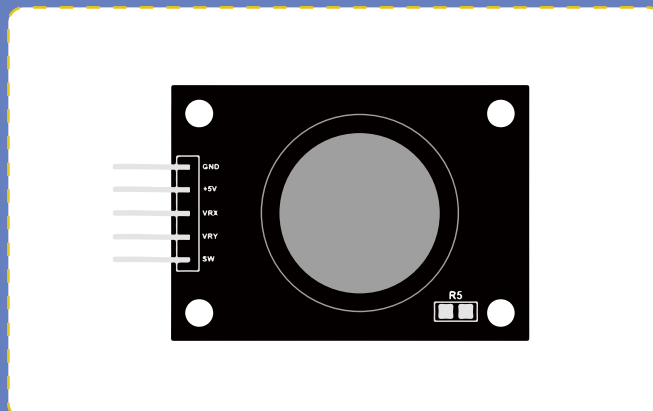


Component Introduction

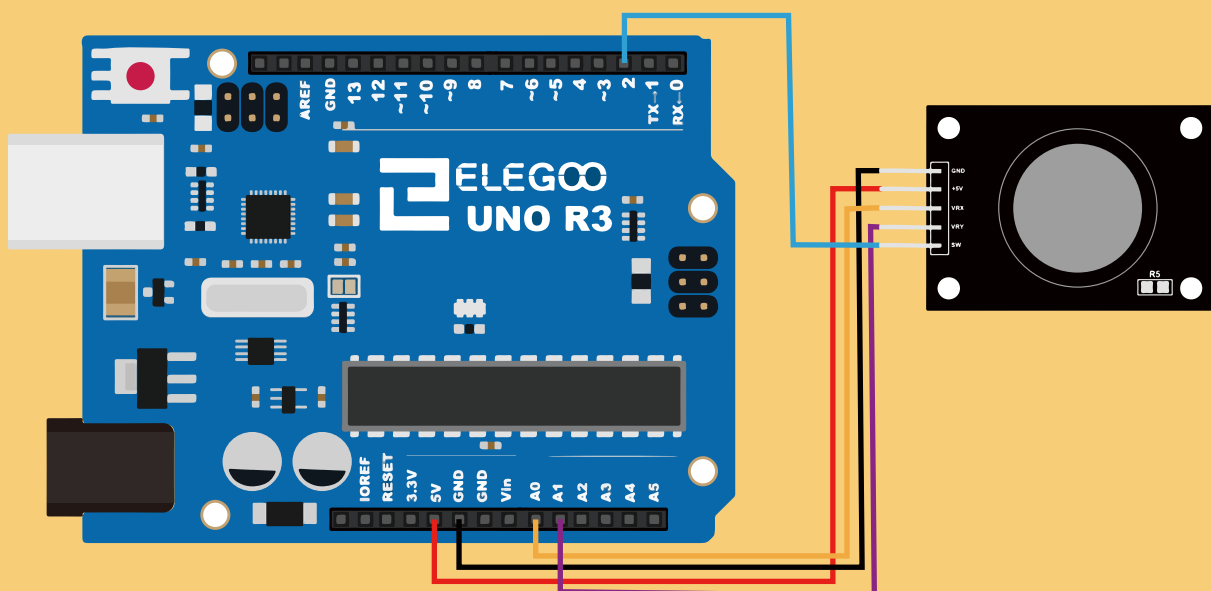
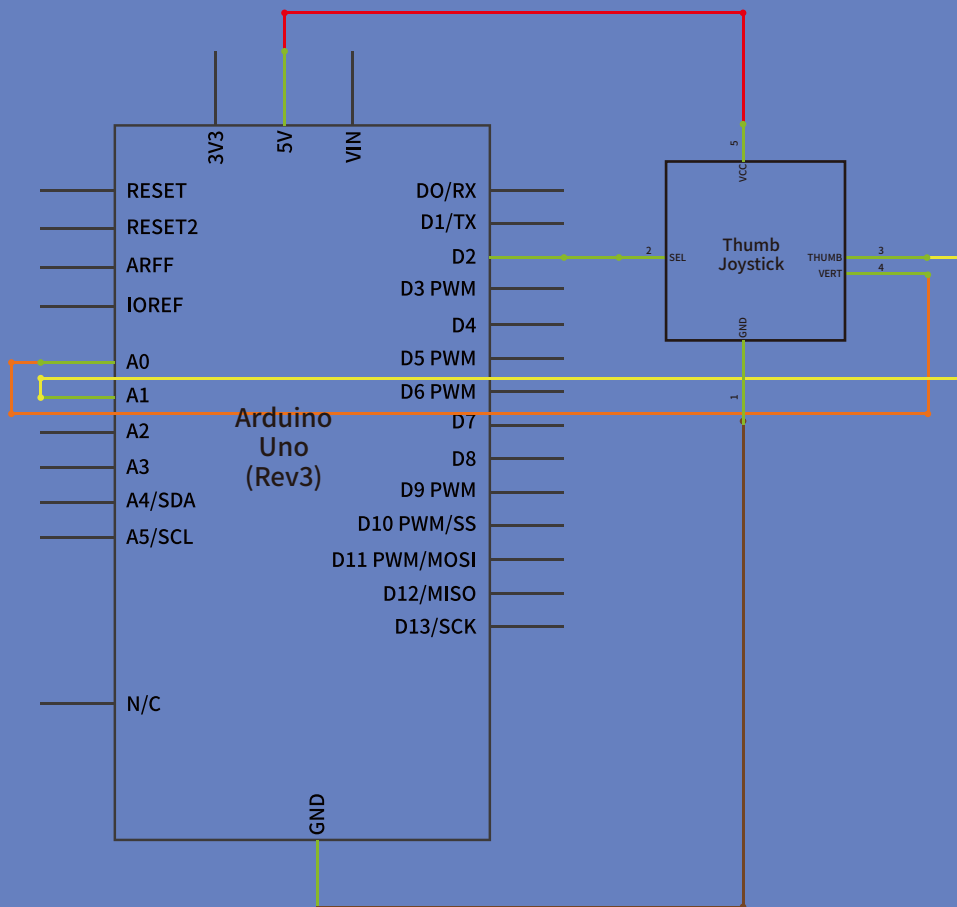
Joystick

The module has 5 pins: VCC, Ground, X, Y, Key. Note that the labels on yours may be slightly different, depending on where you got the module from. The thumb stick is analog and should provide more accurate readings than simple 'directional' joysticks that use some forms of buttons, or mechanical switches. Additionally, you can press the joystick down (rather hard on mine) to activate a 'press to select' push- button.

We have to use analog Arduino pins to read the data from the X/Y pins, and a digital pin to read the button. The Key pin is connected to ground, when the joystick is pressed down, and is floating otherwise. To get stable readings from the Key /Select pin, it needs to be connected to VCC via a pull-up resistor. The built in resistors on the Arduino digital pins can be used. For a tutorial on how to activate the pull-up resistors for Arduino pins, configured as inputs. For example: **pinMode(SW_pin, INPUT);**



Connection Schematic



We need 5 connections to the joystick.

The connections are: Key, Y, X, Voltage and Ground.

“Y and X” are Analog and “Key” is Digital. If you don't need the switch then you can use only 4 pins.

Connection Schematic

Code

After wiring, please open the program in the code folder- **Analog_Joystick** and click UPLOAD to upload the program. See Lesson 5 in part 1 for details about program uploading if there are any errors.

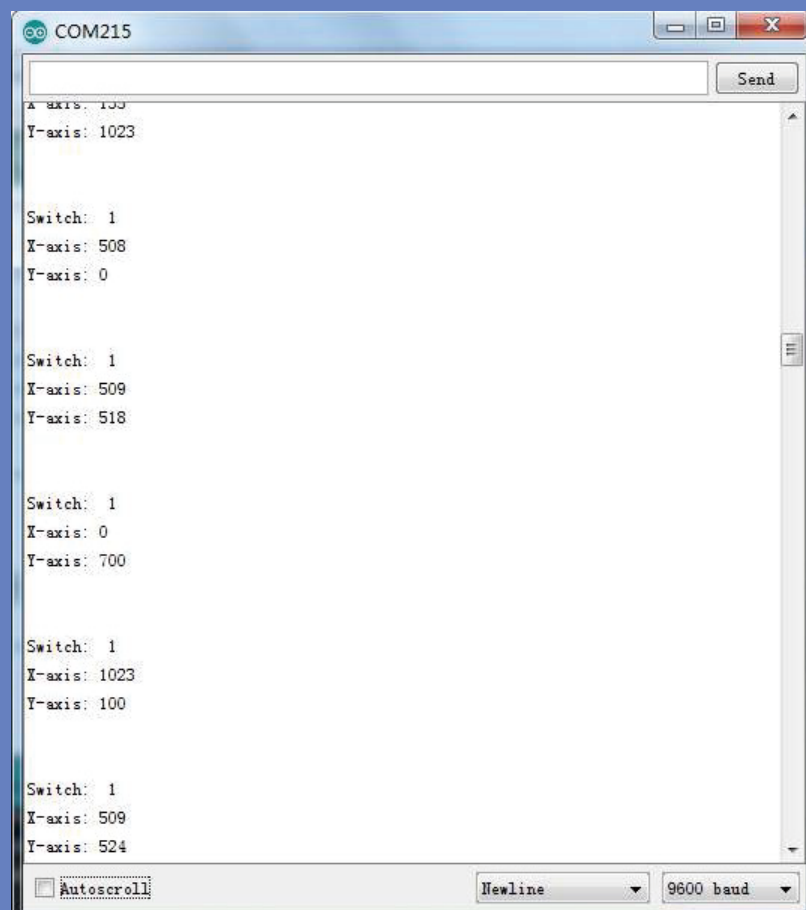
Analog joysticks are basically potentiometers so they return analog values.

When the joystick is in the resting position or middle, it should return a value of about 512.

The range of values goes from 0 to 1024.

Open the monitor then you can see the data as blow:

Click the Serial Monitor button to turn on the serial monitor. The basics about the serial monitor are introduced in details in part 2 Lesson 4 .



```
Serial.print("X-axis: ");
Serial.print(analogRead(X_pin));
```

analogRead()

[Analog I/O]

Description

Reads the value from the specified analog pin. Arduino boards contain a multichannel, 10-bit analog to digital converter. This means that it will map input voltages between 0 and the operating voltage (5V or 3.3V) into integer values between 0 and 1023. On an Arduino UNO, for example, this yields a resolution between readings of: 5 volts / 1024 units or, 0.0049 volts (4.9 mV) per unit. See the table below for the usable pins, operating voltage and maximum resolution for some Arduino boards.

The input range can be changed using `analogReference()`, while the resolution can be changed (only for Zero, Due and MKR boards) using `analogReadResolution()`.

It takes about 100 microseconds (0.0001 s) to read an analog input, so the maximum reading rate is about 10,000 times a second.

*A0 through A5 are labelled on the board, A6 through A11 are respectively available on pins 4, 6, 8, 9, 10, and 12.

**The default `analogRead()` resolution for these boards is 10 bits, for compatibility. You need to use `analogReadResolution()` to change it to 12 bits.

BOARD	BOARD OPERATING VOLTAGE	USABLE PINS	MAX RESOLUTION
Uno	5 Volts	A0 to A5	10 bits
Mini, Nano	5 Volts	A0 to A7	10 bits
Mega, Mega2560, MegaADK	5 Volts	A0 to A14	10 bits
Micro	5 Volts	A0 to A11*	10 bits
Leonardo	5 Volts	A0 to A11*	10 bits
Zero	3.3 Volts	A0 to A5	12 bits**
Due	3.3 Volts	A0 to A11	12 bits**
MKR Family boards	3.3 Volts	A0 to A6	12 bits**

Syntax

```
analogRead(pin)
```

Parameters

pin: the name of the analog input pin to read from (A0 to A5 on most boards, A0 to A6 on MKR boards, A0 to A7 on the Mini and Nano, A0 to A15 on the Mega).

Returns

The analog reading on the pin. Although it is limited to the resolution of the analog to digital converter (0-1023 for 10 bits or 0-4095 for 12 bits). Data type: `int`.