



Thumb Joystick Retail

Skill Level: Beginner

by MikeGrusin | March 22, 2011 |

Thank you for purchasing our Thumb Joystick! Whether you're blasting aliens or driving a robot, you'll find it a very useful addition to your projects.

Tools and parts you'll need

...to attach the breakout board to the joystick:

- Soldering iron and solder
- A 5-pin 0.1" header (handy if you'll be inserting the joystick into a breadboard), or wires to connect the joystick to your project.

Things you'll want to know

- How to solder to a PCB (it's easy, see our tutorials)
- How to program a microcontroller to read analog and digital inputs (don't worry, we'll have some example Arduino code below)

How does it work?

Imagine a volume control or other knob. These controls are generally made from potentiometers, which are variable resistors. By connecting the potentiometer as a voltage divider, you can get a varying voltage out of the device, which can be fed to an analog to digital converter on a microprocessor. This allows the software to read the knob position.

This joystick contains two potentiometers, connected with a gymbal mechanism that separates the "horizontal" and "vertical" movements. (If the joystick is flat on a table, these movements are more "front / back" and "left / right", but you get the idea.) The potentiometers are the two blue boxes on the sides of the joystick. If you move the joystick while watching the center shaft of

each potentiometer, you'll see that each of the potentiometers pick up movement in only one direction. Clever, isn't it!

This joystick also contains a switch which activates when you push down on the cap. The switch is the small black box on the rear of the joystick. If you push down on the cap, you can see a lever pushing down on the head of the switch. The lever works no matter what position the joystick is in. Cool!



Assembly

The retail package includes the joystick and a "breakout board" (the red printed-circuit board, or PCB). You'll solder the breakout board to the joystick. Once this is done, it will be much easier to connect the joystick to your project (you won't have to solder wires to all those little pins). To attach the breakout board to the joystick, follow these steps:

1. Test-fit the breakout board to the joystick. If any pins on the joystick are bent, gently straighten them. The joystick goes into the side of the board with the white outline (*not* the side with the SparkFun logo). Insert the joystick into the board, matching the white outline, and ensure that all the pins go into their holes (if they don't, you may not have the joystick oriented correctly).

2. If everything fits OK, carefully solder the pins from the opposite side of the board (the side *with* the SparkFun logo). Never soldered before? It's easy, check out our tutorials. If this is your first time soldering, practice a bit before working on this board. Before and during soldering, ensure that the joystick stays tight against the breakout board. There are 14 solder connections to make, including the four posts at the corners of the joystick.
3. If you'd like to connect a 5-pin header to the board, or solder bare wires to the output holes, you may do that now. Once you're done soldering, you're ready to use your joystick!



Use it!

The breakout board provides five labeled connections. You'll connect these lines to your project or microcontroller:

- VCC: connect this to your positive supply (usually 5V or 3.3V)
- VERT: this is the vertical output voltage (will be half of VCC when the joystick is centered)
- HORIZ: this is the horizontal output voltage (will be half of VCC when the joystick is centered)
- SEL: this is the output from the pushbutton, normally open, will connect to GND when the button is pushed (see example code below)
- GND: connect this to your ground line (GND)

Note that the pushbutton (SEL) is connected to GND when pressed, and open (disconnected) when unpressed. Use a pullup resistor on your digital input so that when it's unpressed, your input will read 1 (HIGH), and when pressed, the input will read 0 (LOW). Many microcontrollers have internal pullup resistors you can use for this purpose. See the example code below for how to do this on an Arduino.

Here's an example Arduino sketch which sets up the microcontroller to read the inputs, and then continuously prints the values to the serial monitor. If you're using a different microcontroller, the steps should be similar; you'll just need to determine how to read analog and digital inputs on your device.

```
// Thumb Joystick example
// Mike Grusin, SparkFun Electronics 3/11
// This code is free, baby. Use it however you like.

// This sketch shows the basic operation of the Thumb Joystick (COM-
09032) and breakout board (BOB-09110).
// The joystick outputs two analog voltages (VERT and HORIZ), and one digital
signal (SEL) for the pushbutton.

// Connections to joystick (change if you use different pins):

const int VERT = 0; // analog
const int HORIZ = 1; // analog
const int SEL = 2; // digital

// Also connect the joystick VCC to Arduino 5V, and joystick GND to Arduino G
ND.

// This sketch outputs serial data at 9600 baud (open Serial Monitor to view)
.

void setup()
{
  // make the SEL line an input
  pinMode(SEL, INPUT);
  // turn on the pull-up resistor for the SEL line (see
http://arduino.cc/en/Tutorial/DigitalPins)
  digitalWrite(SEL, HIGH);

  // set up serial port for output
  Serial.begin(9600);
}

void loop()
{
  int vertical, horizontal, select;

  // read all values from the joystick

  vertical = analogRead(VERT); // will be 0-1023
  horizontal = analogRead(HORIZ); // will be 0-1023
```

```
    select = digitalRead(SEL); // will be HIGH (1) if not pressed, and LOW (0)
    if pressed

    // print out the values

    Serial.print("vertical: ");
    Serial.print(vertical,DEC);
    Serial.print(" horizontal: ");
    Serial.print(horizontal,DEC);
    Serial.print(" select: ");
    if(select == HIGH)
        Serial.println("not pressed");
    else
        Serial.println("PRESSED!");
}
```

Tips

- Because of resistor manufacturing tolerances, the joystick may not read exactly half of VCC when at rest. You may want to make a note of the "zero" reading, and hardcode that value into your program if necessary.
- This joystick is a low-power device. The analog and switch outputs should be connected only to high-impedance inputs, such as I/O ports on a microcontroller or op-amp inputs. Don't try to hook the joystick directly to LEDs, motors, or other high-current devices (more than 1mA), or you may damage it.

Have fun!

Enjoy your new joystick! If you have questions or problems (or want to show us what you're building), please don't hesitate to contact us at techsupport@sparkfun.com.