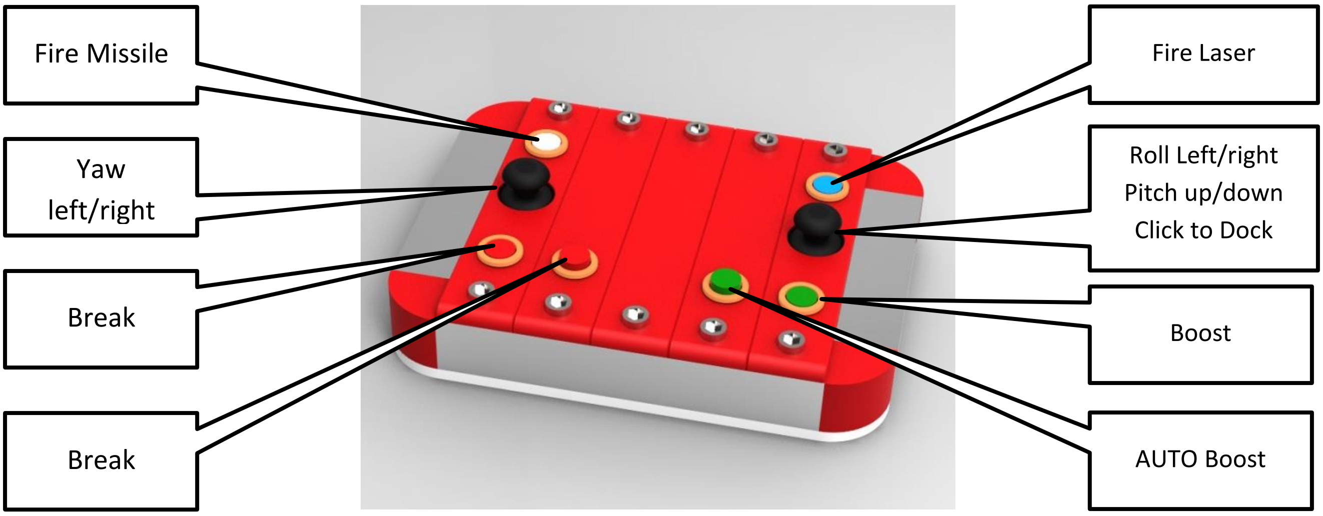
Artemis
Spaceship Bridge Simulator
BE THE CREW.
REPORT TO THE BRIDGE.

Index

Fighter controls:



Fighter joystick

Note: the Joystick is rotated CCW to fit on the 3D printed mount. Therefore the input in the X-axes is reversed

Make the following changes to the controls.ini

```
FIGHTER_ANALOG_ROLL       =          UI_INPUT_JOY0_AXIS0
FIGHTER_ANALOG_PITCH      =          UI_INPUT_JOY0_AXIS1
; un-comment the next line, if your joystick climbs when you want it to dive
MULTIPLY                 =          -1.0
; comment the next line out, if your joystick does NOT twist
FIGHTER_ANALOG_YAW        =          UI_INPUT_JOY0_AXIS2
```

Description	Wire color	Arduino Pro Micro Pin	Extern connection
Joystick 1 Roll/pitch	Purple	Gnd	Gnd
	Blue	Vcc	+5V
Roll left/right	Green	A0	VRX
Pitch up/down	Yellow	A1	VRY
Request Docking (Push button 4)	orange	7	SW

Description	Wire color	Arduino Pro Micro Pin	Extern connection
Joystick 2 YAW	Purple	Gnd	Gnd
	Gray	Vcc	+5V
YAW Left/right	white	A2	VRY

Description	Wire color	Arduino Pro Micro Pin	Extern connection
Beam (Push button 0)		3	2nd pin to gnd
Boost (Push button 1)		4	2nd pin to gnd
Break (Push button 2)		5	2nd pin to gnd
Missile (Push button 3)		6	2nd pin to gnd

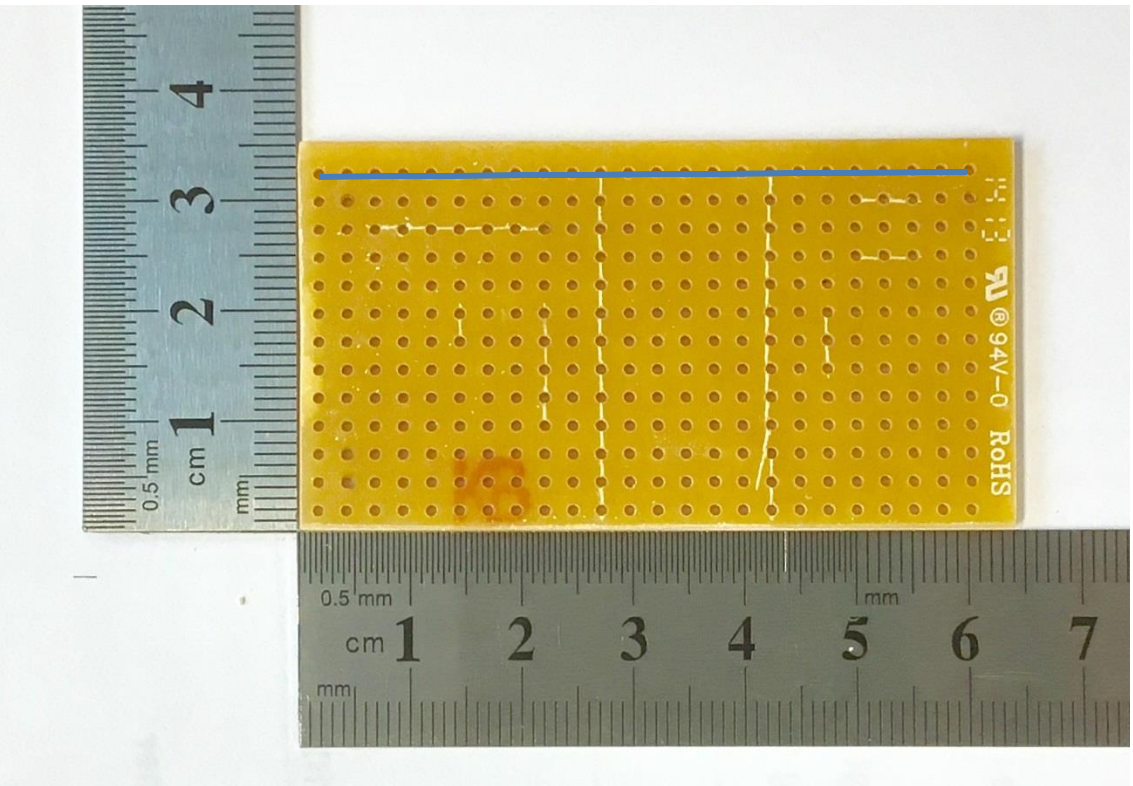
The code:

```
/*i have wrote the code, and everything works like a charm! thanks once again Matthew!
here is my code if anyone is interested in using it, or imporving it(bevause i know its not going to be good
:P)

********************************************
*                  Artemis                 *
*             Fighter joystick              *
*          modified By Rothestar           *
*     to work with my hardware design      *
* see my build blog at http://seosalle.dk  *
********************************************

*/


#include "Joystick.h"

Joystick_ Joystick;

void setup() {

Joystick.begin();

Joystick.setXAxis(0);

Joystick.setYAxis(0);

Joystick.setZAxis(0);

pinMode(3, INPUT_PULLUP);//button 0 Beam

pinMode(4, INPUT_PULLUP);//button 1 Boost

pinMode(5, INPUT_PULLUP);//button 2 Brake

pinMode(6, INPUT_PULLUP);//button 3 Missiles

pinMode(7, INPUT_PULLUP);//button 4 Request Dock

pinMode(8, INPUT_PULLUP);

pinMode(9, INPUT_PULLUP);

pinMode(10, INPUT_PULLUP);
```

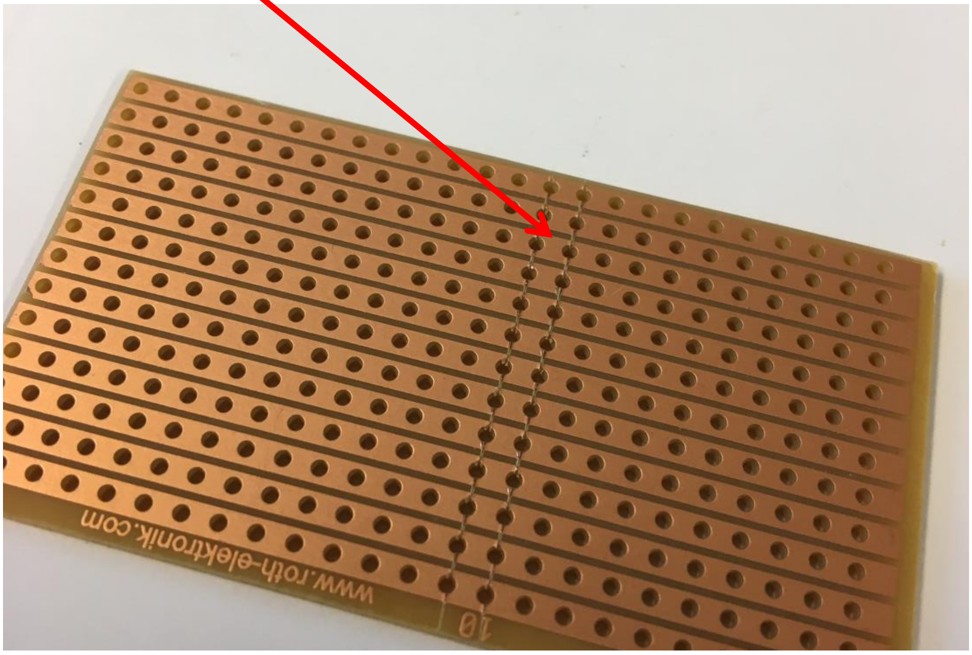
```
int lastButtonState = 0;

}

void loop() {

int button0Val =digitalRead(3);

int button1Val =digitalRead(4);

int button2Val =digitalRead(5);

int button3Val =digitalRead(6);

int button4Val =digitalRead(7);

int button5Val =digitalRead(8);

int button6Val =digitalRead(9);

//int button7Val =digitalRead(10);


int xAxis = analogRead(A1);//switch due to fact that the joystick is turned 90 degrees to fit into the hardware design (original it should be A0).

int yAxis = analogRead(A0);//switch due to fact that the joystick is turned 90 degrees to fit into the hardware design (original it should be A1).

int zAxis = analogRead(A2);

Joystick.setXAxis(xAxis);

Joystick.setYAxis(yAxis);

Joystick.setZAxis(zAxis);

Joystick.setButton(0, !button0Val);

Joystick.setButton(1, !button1Val);

Joystick.setButton(2, !button2Val);
Joystick.setButton(3, !button3Val);

Joystick.setButton(4, !button4Val);

Joystick.setButton(5, !button5Val);

Joystick.setButton(6, !button6Val);

//Joystick.setButton(7, !button7Val);

}
```

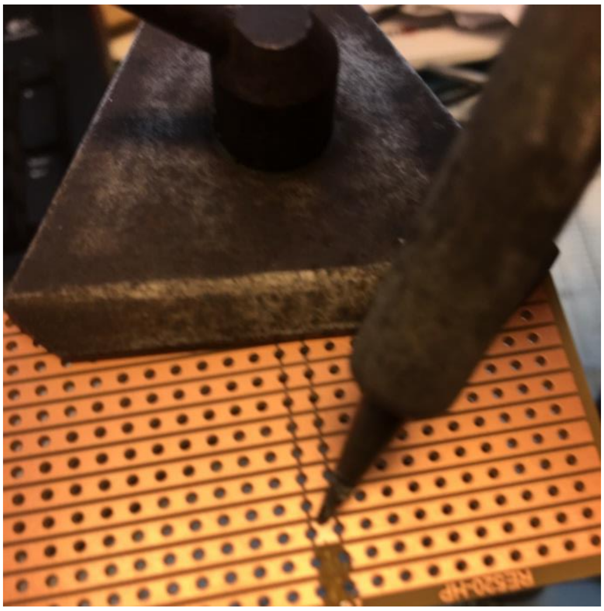
Build instructions:

Stripled PCB rastersize 2,54 x 2,54mm

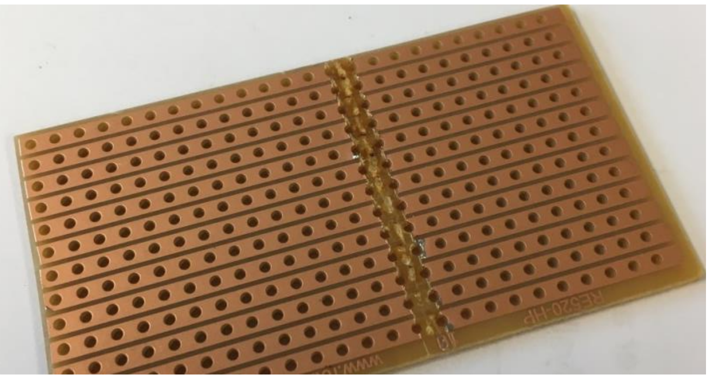


With a sharp knife cut two lines perpendicular to the stripes. This is done to separate the two sides of the Arduino Pro Micro:

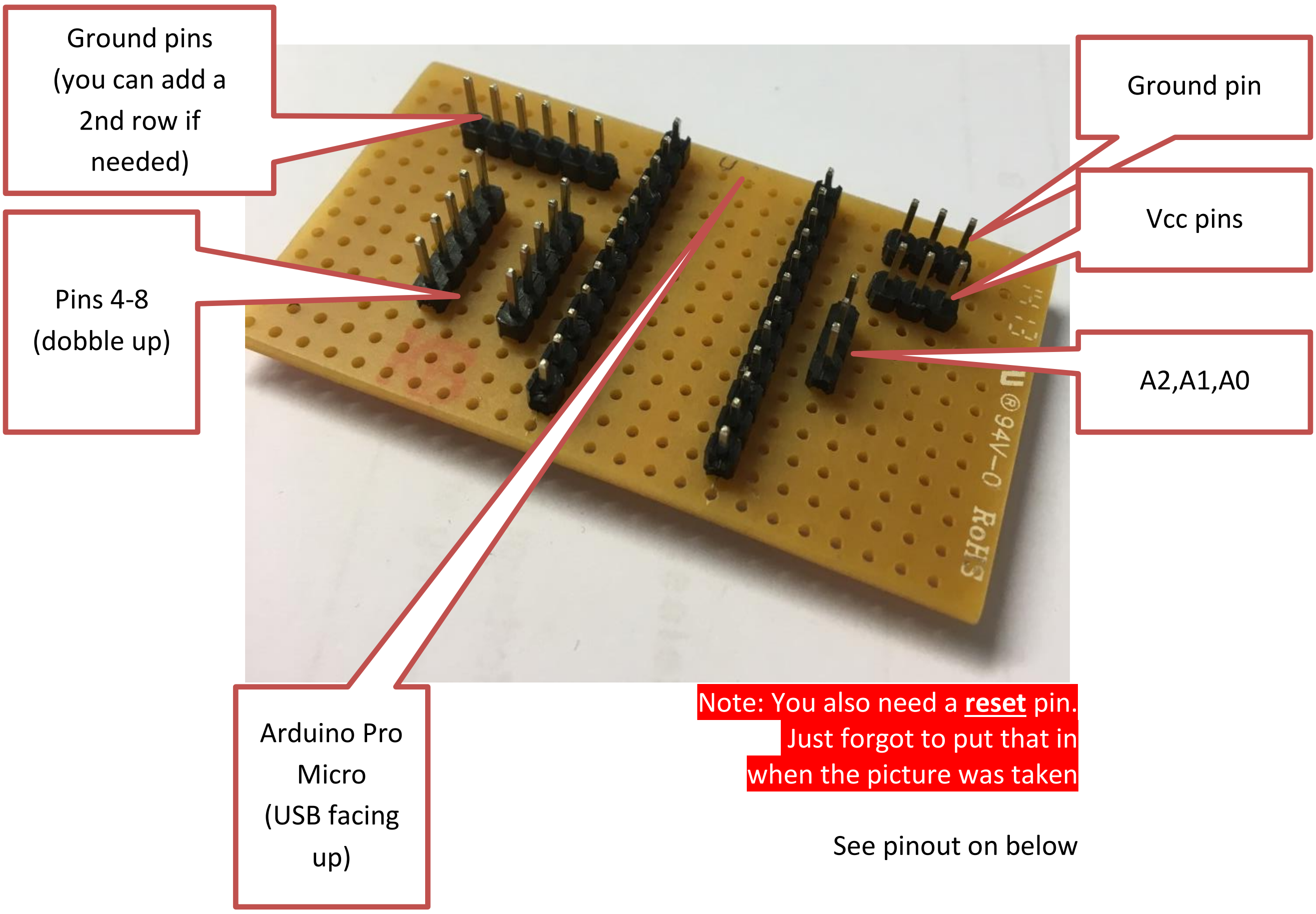




With a hot soldering iron heat up the cobber strip and it will come loose.
That way you get a perfect visible separation of the two sides.



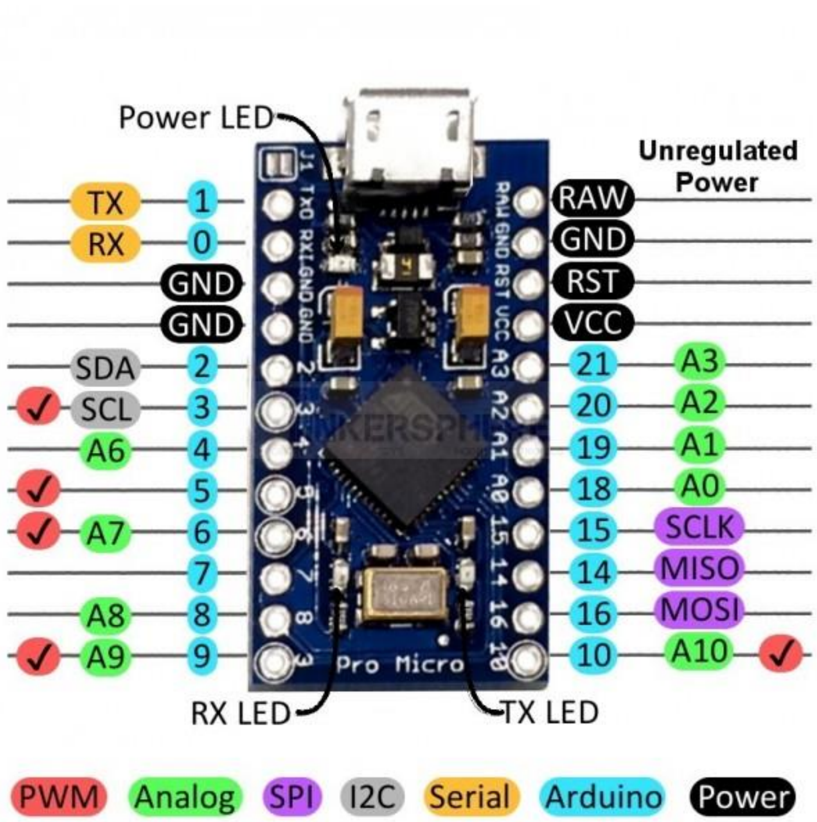
The finished result:


Ground pins (you can add a 2nd row if needed)
Pins 4-8 (dobble up)
Ground pin
Vcc pins
A2,A1,A0
Arduino Pro Micro (USB facing up)

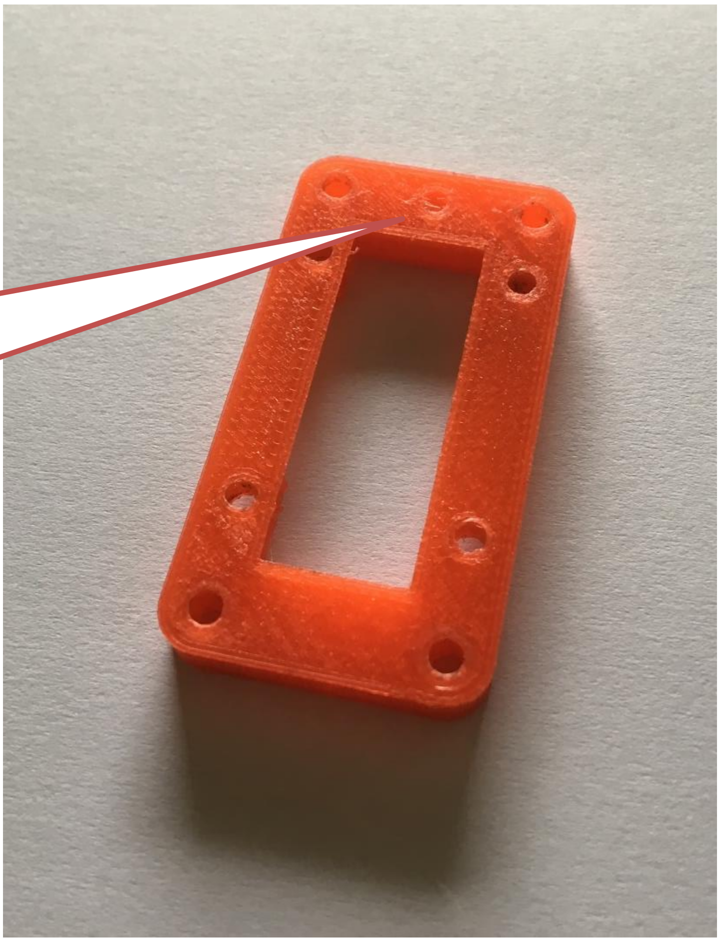
Note: You also need a **reset** pin. Just forgot to put that in when the picture was taken

See pinout on below

Arduino Pro Micro pinout:

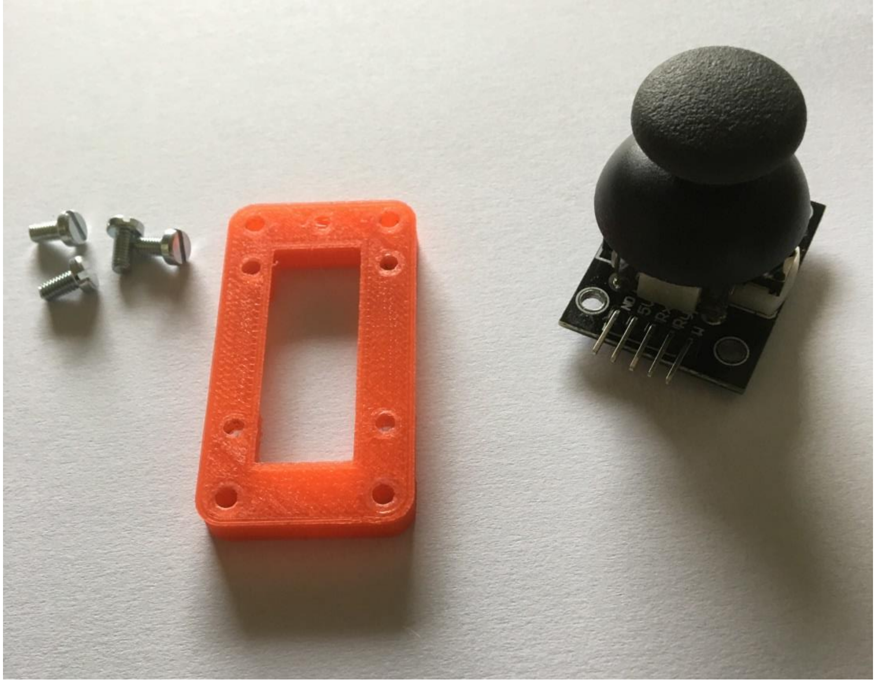


Assembly of the joystick (up/down, roll left/right):



Note the dot

Indicates direction.

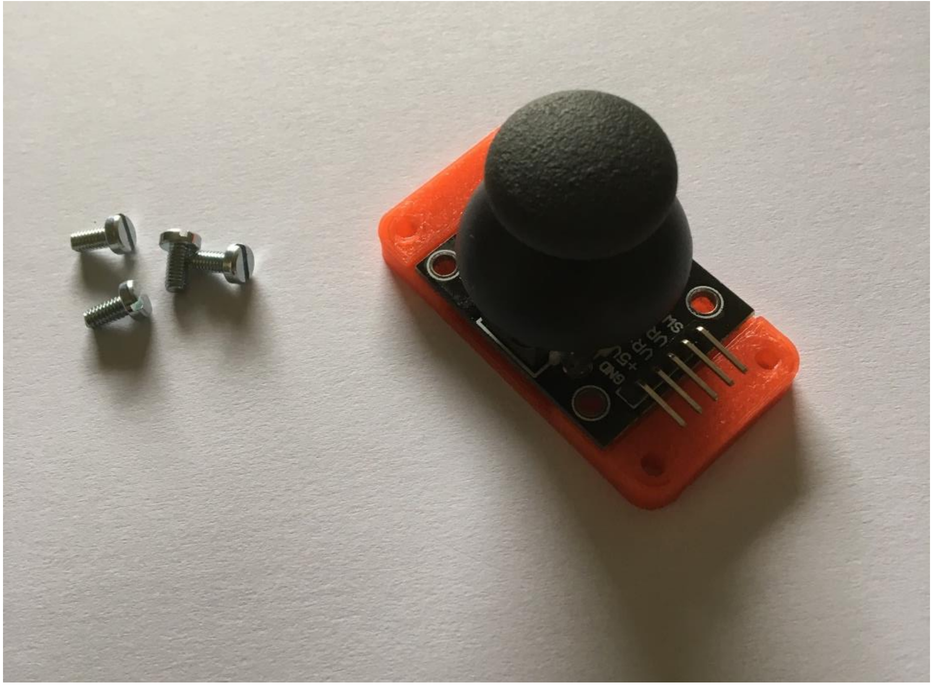


Items:

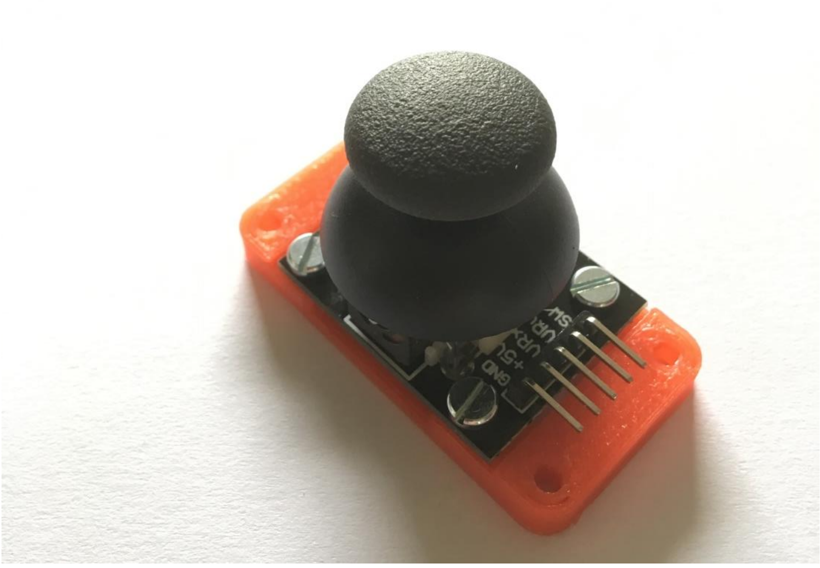
4 pcs. 3x5mm screws

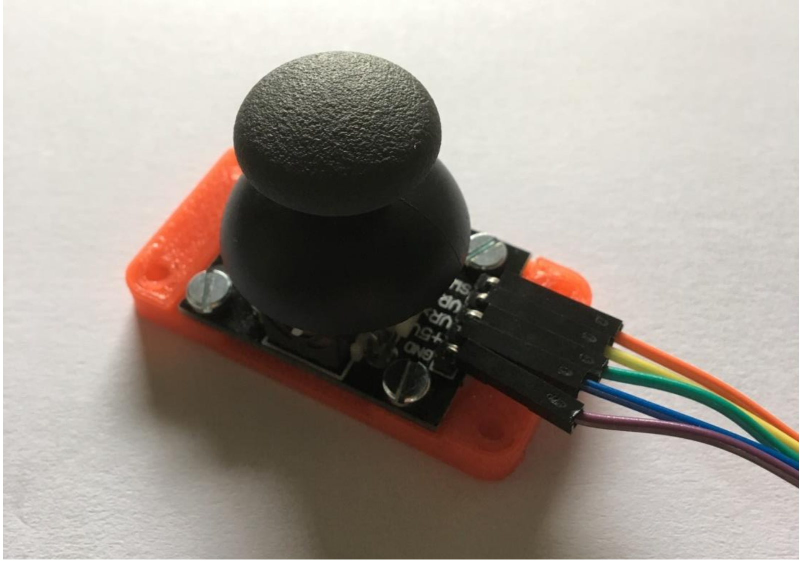
1 pcs. Joystick

1 pcs. joystick bracket FINAL.stl

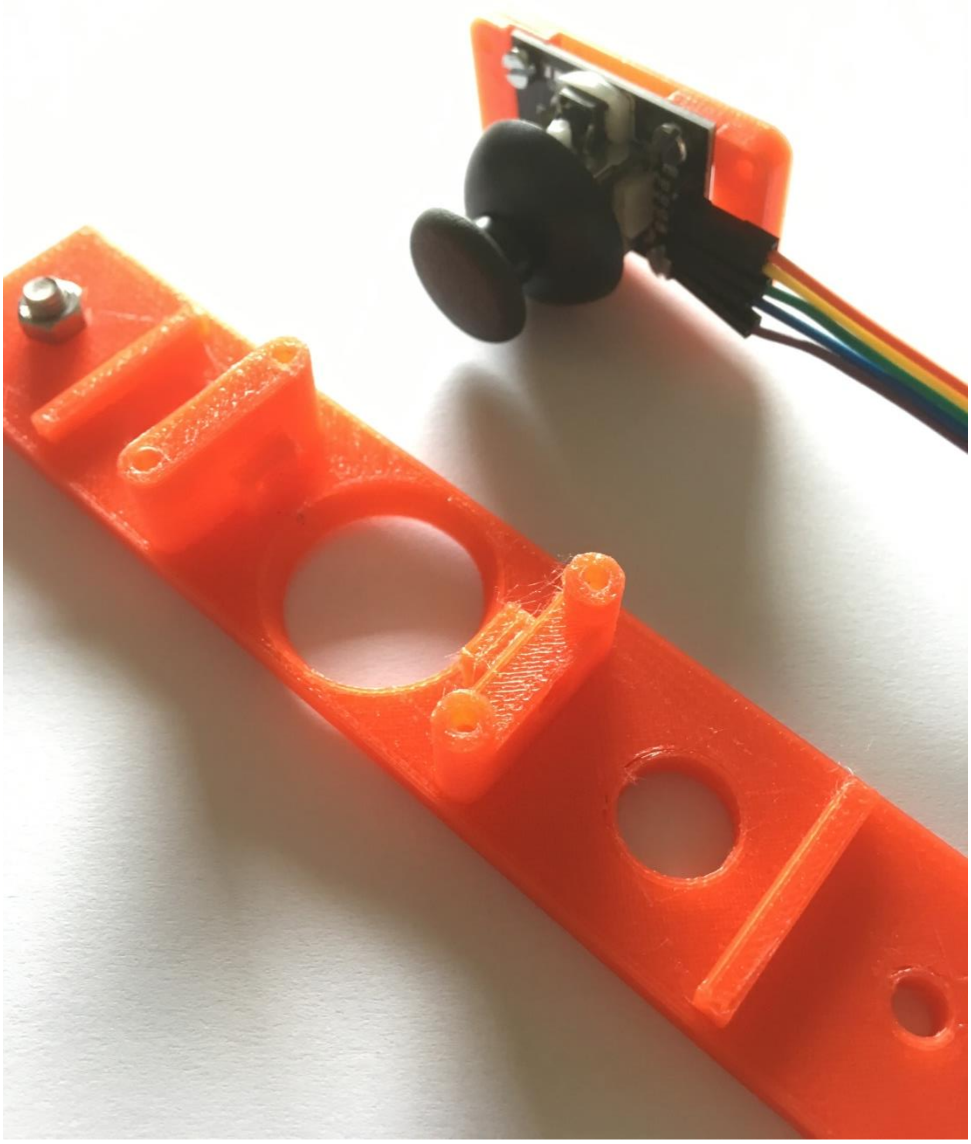


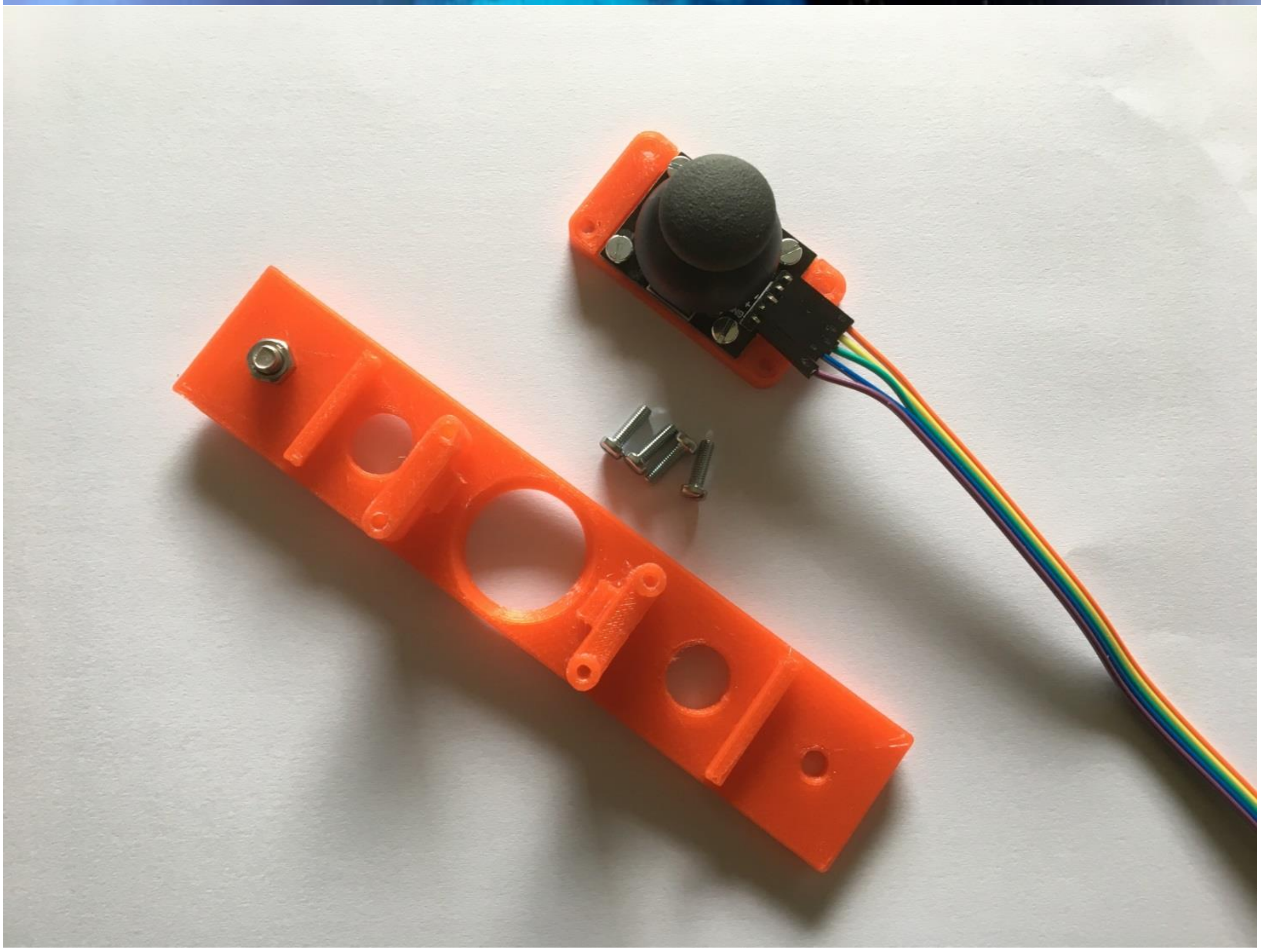
Flip the bracket the dot facing opposite of the header pins.





Connect female – female dupont jumper wires. Note color in the table first page.





Mount the joystick assembly to the strip.

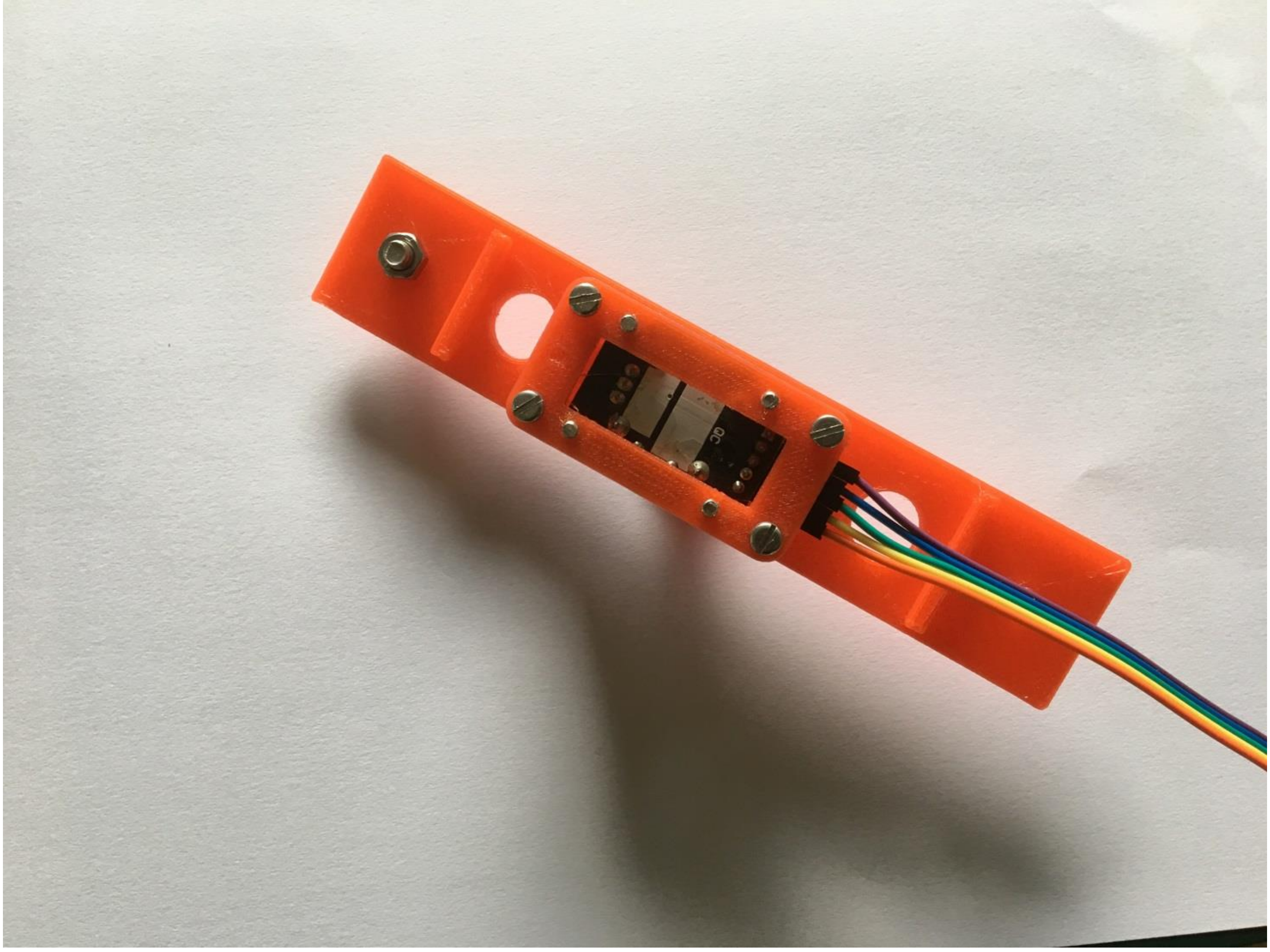
Items:

4 pcs. 3x10mm screws

1 pcs. Joystick assebley

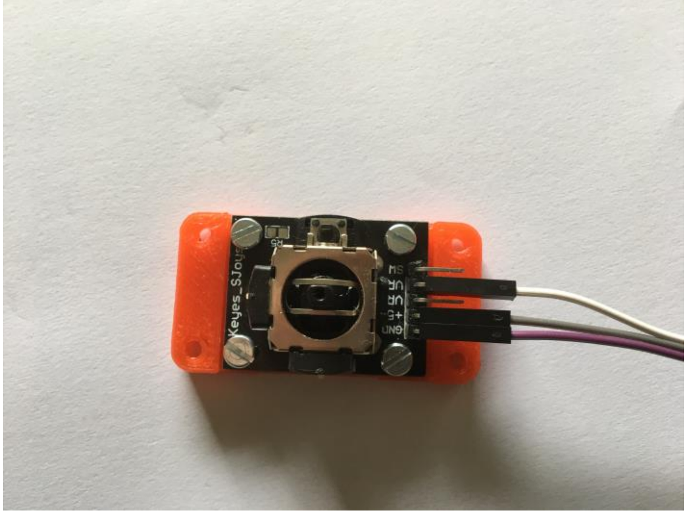
1 pcs. Frame 1 joystick.stl

F

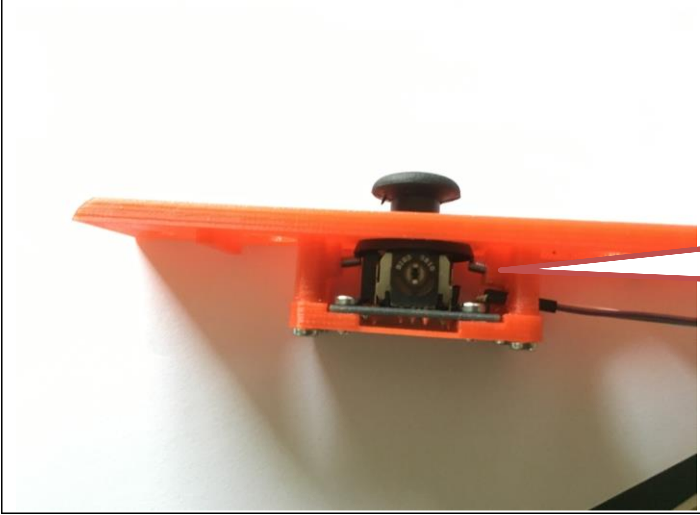


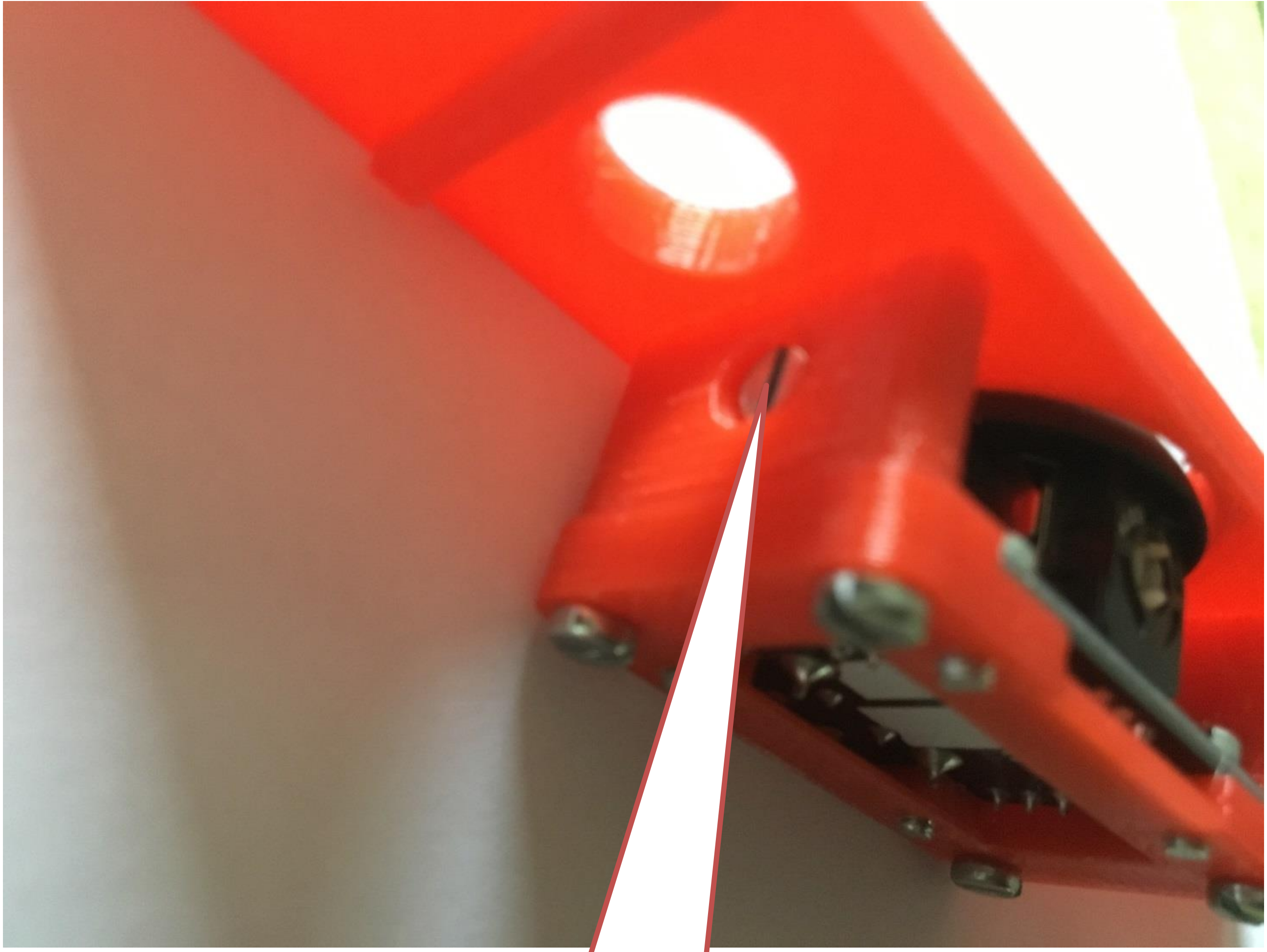
finished assebley.

Assembly of the joystick (YAW left/right):



Same procedure
Note jumper wire color.

	Note 2 pcs. 3x10mm screw to lock the joystick (no up/down



Note 2 pcs. 3x10mm screw to lock the joystick (no up/down

Links:

Rotery encode

https://github.com/buxtronix/arduino/tree/master/libraries/Rotary

more encoder

http://www.buxtronix.net/2011/10/rotary-encoders-done-properly.html

Jumper wires

http://www.ebay.com/itm/40PCS-Dupont-Wire-Jumper-Cables-10cm-Female-To-Female-1P-1P-For-Arduino/311660166123?ssPageName=STRK%3AMEBIDX%3AIT&_trksid=p2057872.m2749.l2649

Joystick

http://www.ebay.com/itm/5PCS-5Pin-JoyStick-Breakout-Module-Shield-PS2-Joystick-Game-Controller-AL/400797796174?ssPageName=STRK%3AMEBIDX%3AIT&_trksid=p2057872.m2749.l2649

Pushbutton:

http://www.ebay.com/itm/5PCS12mm-250V-3A-Red-Push-Button-Switch-PBS-11B-No-Self-Lock-ON-OFF-Lock/311576502410?ssPageName=STRK%3AMEBIDX%3AIT&_trksid=p2057872.m2749.l2649