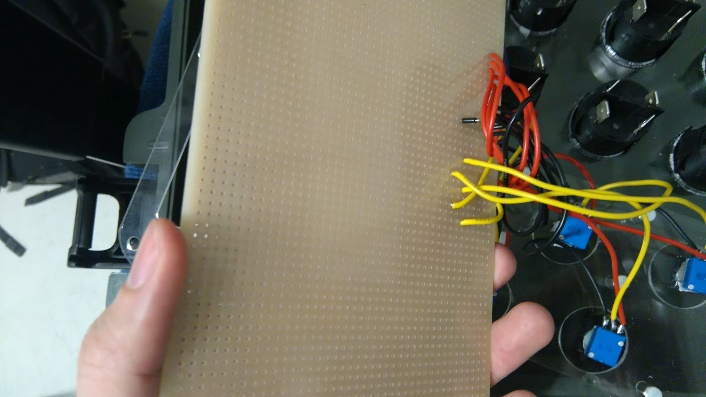
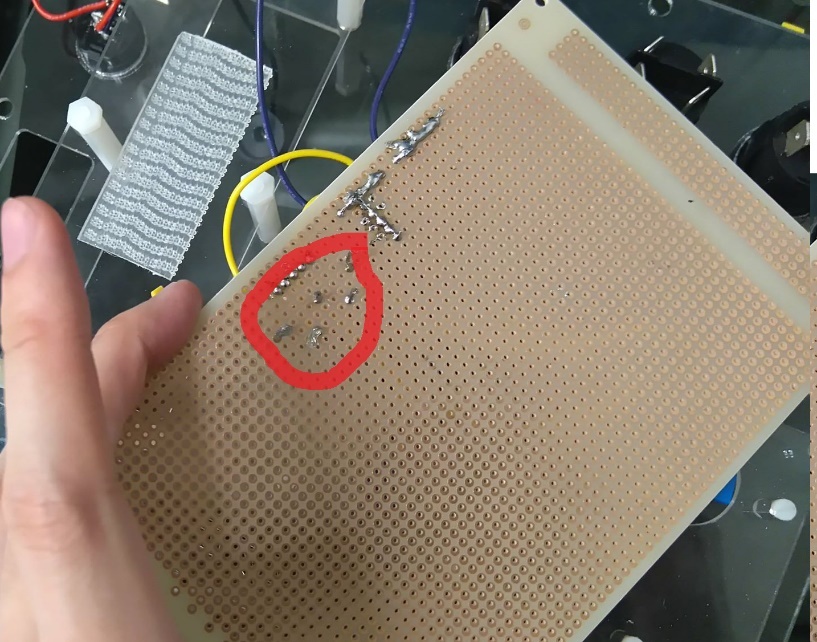
Gimbal Control

# How it works

# 1. Potentiometer varies voltage

* Potentiometers are in the ground control station.
* The potentiometers share a common ground with each other and the power supply (and the Arduino as well.
* The power supply goes into the barrel jack to supply 5V. This voltage enters each potentiometer.
* Turning the potentiometer will let through varying voltages between 0 and 5V. This output is the yellow wires (one for each potentiometer).
* Connect the yellow wires (from the back in the Figure on bottom left) to analog ports in the Arduino and connect the ground wire (in black, can be seen in Figure on bottom right near the red wires) to the ground port in the Arduino.

# 2. Arduino reads voltage from analog pin

/\*

Arduino reads voltage value

\*/

int bottomLeft = 3; // bottom left potentiometer wiper (middle terminal) connected to analog pin 3

int bottomRight = 4;

int topLeft = 5;

int topCenter = 6;

int topRight = 7;

// outside leads to ground and +5V

int bLval = 0; // variable to store the value read

int bRval = 0;

int tLval = 0;

int tCval = 0;

int tRval = 0;

int bLvalPrev = -1; // variable to store the last value read (to detect any changes)

int bRvalPrev = -1;

int tLvalPrev = -1;

int tCvalPrev = -1;

int tRvalPrev = -1;

float voltage = -1;

void setup()

{

Serial.begin(9600); // setup serial

}

void loop()

{

//Read all input pins

bLval = analogRead(bottomLeft); // read the input pin

bRval = analogRead(bottomRight);

tLval = analogRead(topLeft);

tCval = analogRead(topCenter);

tRval = analogRead(topRight);

//Check if any change was made to the values: write to serial Voltage + 10\*identifier

//Identifiers:

//1: change bLval

//2: change bRval

//3: change tLval

//4: change tCval

//5: change tRval

voltage=bLval/204.0 + 10;

Serial.println(voltage);

voltage=bRval/204.0 + 20;

Serial.println(voltage);

voltage=tLval/204.0 + 30;

Serial.println(voltagef);

voltage=tCval/204.0;

Serial.println(voltage + 40);

voltage=tRval/204.0;

Serial.println(voltage + 50);

}

bottomLeft, etc. indicate the ports that the potentiometers are plugged into (vary if needed). The Arduino must be plugged into the Ground Control Station via USB cable (check that the correct com port is connected). An offset >10 is applied to the signal from each potentiometer, which allows the unique signals to be sent on the same transmission link.

# 3. Python Script (on MissionPlanner)

ser=serial.Serial('COM7', 9600) #May need to change this COM

min\_pwm=553 #pwm value of left-most rotation

max\_pwm=2250

#pwm value of right-most rotation

max\_voltage=5.0

slope=(max\_pwm-min\_pwm)/max\_voltage

counter = 0

while True:

#1. Read serial line

a=ser.readline()

#2. Decode

voltage=float(a.decode('utf-8'))

print(voltage)

#3. See if there was any change

#3A. Figure out which servo needs to be changed using identifier

'''Identifier:

1: bLval (mapped to channel 9)

2: bRval (mapped to channel 10)

3: TLval (mapped to channel 11)

4: TCval (mapped to channel 12)

5: TRval (mapped to channel 13)

'''

identifier = voltage//10

print ("Identifier: ", identifier)

#3B. Write to the relevant channel based on identifier

#Take remainder of dividing voltage by 10 (releases identifier)

try:

pwm=min\_pwm+(voltage%10)\*slope

MAV.doCommand(MAVLink.MAV\_CMD.DO\_SET\_SERVO, (8 + identifier), pwm, 0, 0, 0, 0, 0);

except ValueError:

bbb=1

#Housekeeping

print(counter)

print (voltage)

counter +=1

To load this file into Mission Planner, click on the ‘Scripts’ tab, and load this file to run (see below). If anything changes (ie lost connection, you get an error, etc) the script must be restarted.

