import os #importing os library so as to communicate with the system

import time #importing time library to make Rpi wait because its too impatient

os.system ("sudo pigpiod") #Launching GPIO library

time.sleep(1) # As i said it is too impatient and so if this delay is removed you will get an error

#import pigpio #importing GPIO library

import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM) #BCM

MIN\_POT=0

MAX\_POT=1023

# freely chosen SPI pins

SPICLK = 16 # BCM 16 BOARD 36

SPIMISO = 19 # BCM 19 BOARD 35

SPIMOSI = 20 # BCM 20 BOARD 38

SPICS = 25 # BCM 25 BOARD 22

GPIO.setup(SPIMOSI, GPIO.OUT)

GPIO.setup(SPICLK, GPIO.OUT)

GPIO.setup(SPICS, GPIO.OUT)

GPIO.setup(SPIMISO, GPIO.IN )

ESC=4 #Connect the ESC in this GPIO pin

GPIO.setup(ESC, GPIO.OUT)

potentiometer\_adc = 0

trim\_pot=0

max\_value = 80 #change this if your ESC's max value is different or leave it be

min\_value = 0 #change this if your ESC's min value is different or leave it be

p = GPIO.PWM(ESC, 50)

p.start(0)

trim=0

def readadc(adcnum, clockpin, mosipin, misopin, cspin): #adcout veriyor

if ((adcnum > 7) or (adcnum < 0)):

return -1

GPIO.output(cspin, True)

GPIO.output(clockpin, False) # start clock low

GPIO.output(cspin, False) # bring CS low

commandout = adcnum

commandout |= 0x18 # start bit + single-ended bit

commandout <<= 3 # we only need to send 5 bits here

for i in range(5):

if (commandout & 0x80):

GPIO.output(mosipin, True)

else:

GPIO.output(mosipin, False)

commandout <<= 1

GPIO.output(clockpin, True)

GPIO.output(clockpin, False)

adcout = 0

# read in one empty bit, one null bit and 10 ADC bits

for i in range(12):

GPIO.output(clockpin, True)

GPIO.output(clockpin, False)

adcout <<= 1

if (GPIO.input(misopin)):

adcout |= 0x1

GPIO.output(cspin, True)

adcout >>= 1 # first bit is 'null' so drop it

return adcout

def read\_potentiometer():

trim\_pot = readadc(potentiometer\_adc, SPICLK, SPIMOSI, SPIMISO, SPICS)

return round(trim\_pot / 1024.0, 2)

def map(x, in\_min, in\_max, out\_min, out\_max):

float(x)

return float((x - in\_min) \* (out\_max - out\_min) / (in\_max - in\_min) + out\_min)

try:

while True:

trim\_pot = readadc(potentiometer\_adc, SPICLK, SPIMOSI, SPIMISO, SPICS)

trim=map(trim\_pot,MIN\_POT,MAX\_POT,min\_value,max\_value)

print("Trim: ",trim)

if trim >= 80:

trim=80

elif trim<=30:

trim=30

p.ChangeDutyCycle(trim) # turn towards 90 degree

except KeyboardInterrupt:

p.stop()

GPIO.cleanup()