import serial  
import RPi.GPIO as GPIO  
import threading  
  
sPortToUse = "/dev/serial0"  
serialPort = serial.Serial(sPortToUse, 230400, timeout = 2)  
  
  
def fSerialCallBack(iChannel):  
  
 sReceive = b""  
 try:  
 if serialPort.open:  
 #disable uno from collecting  
 #and sending data while Pi  
 #is busy processing  
 GPIO.output(iPiReady, GPIO.HIGH)  
 sReceive = serialPort.readline()  
 fProcessData(sReceive.decode('utf-8'))   
 else:  
 print("Port", sPortToUse, " !open")  
 except:  
 pass  
 finally:  
 #RASPBERRY PI READY  
 if serialPort.open:  
 try:  
 serialPort.flushInput()  
 except serial.serialutil.SerialException:  
 pass  
 return;  
  
def fProcessData(sTandH):  
 sTandH=sTandH.split()  
   
 if (sTandH[iZero] == "cm"):  
 Do some more stuff  
   
 elif (sTandH[iZero] == "VOLTS") :  
 fVoltsIn(int(sTandH[iOne]))  
   
 except Exception as e:  
 print("fProcessData(sTandH) " + sTandH + " " + str(e))  
 finally:  
 #enable uno to collect and send data  
 GPIO.output(iPiReady, GPIO.LOW)  
 return;  
  
def fSetup():  
 GPIO.setmode(GPIO.BCM)  
 ##triggred by Arduino to indicate serial data has been sent  
 # event to read serial port  
 GPIO.add\_event\_detect(iUnoDataReady, GPIO.RISING, callback=fSerialCallBack)  
  
 GPIO.setup(iUnoDataReady, GPIO.IN, pull\_up\_down = GPIO.PUD\_DOWN)  
 serialPort.flushOutput()  
 serialPort.flushInput()  
  
 return;  
def fSendSerial(sToSend):  
 if (not (GPIO.input(iPiReady))):  
 serialPort.flushOutput()  
 serialPort.write(sToSend.encode('utf-8'))  
  
 return;  
  
def fMeasureVoltageArduino():  
##threaded function called by the 1PPS interrupt  
 fSendSerial("doVolts")  
 return;  
   
def fDoTheThing():  
  
 global bIsGPIO\_CleanUp  
 global bResetTimePassed  
 global bOLED\_ContinueCount  
   
 try:  
 fSetup()  
 #wait for stop signal  
 GPIO.wait\_for\_edge(iHalt, GPIO.FALLING)  
 except Exception as e:  
 print("fDoTheThing() " + str(e))  
   
 finally:  
 fDisplayOLED\_Message("S T O P")  
 GPIO.cleanup()  
 bIsGPIO\_CleanUp = True  
 serialPort.close()  
 return;