from serial import\*

from Tkinter import\*

from webbrowser import\*

from time import\*

import random

ser = Serial('/dev/ttyACM0', 9600) #Para cuando no estas en la pi

#ser = Serial('/dev/tty.usbserial', 9600) #Para cuando no estas en la pi

tinicial= int(time())

def fd():

ser.write("Fd.")

def left():

ser.write("Lf.")

def staph():

ser.write("St.")

def right():

ser.write("Rt.")

def bd():

ser.write("Bd.")

def prender1():

global pto

pto = 1

def apagar1():

global pto

pto = 0

def prender2():

global tem

tem = 1

def apagar2():

global tem

tem = 0

vent = Tk()

global pto

pto = 0

global tem

tem = 0

marco1 = Frame(vent)

marco1.pack()

fd = Button(marco1, width=20,text="Adelante",command=fd)

fd.pack()

marco2 = Frame(vent)

marco2.pack()

left = Button(marco2, width=20,text="Izquierda",command=left)

left.pack(side=LEFT)

stop = Button(marco2, width=20,text="Detenerse",command=staph)

stop.pack(side=LEFT)

right = Button(marco2, width=20,text="Derecha",command=right)

right.pack(side=LEFT)

marco3 = Frame(vent)

marco3.pack()

Bd = Button(marco3, width=20,text="Atras",command=bd)

Bd.pack()

marco4 = Frame(vent)

marco4.pack(side=LEFT)

si1 = Button(marco4, text="Prender",command=prender1)

si.pack(side=LEFT)

no1 = Button(marco4, text="Apagar",command=apagar1)

no.pack(side=LEFT)

acelerometro = Label(marco4, text="Acelerometro =" )

acelerometro.pack(side=LEFT)

acce = Label(marco4, text="")

acce.pack(side=LEFT)

marco5 = Frame(vent)

marco5.pack(side=LEFT)

hora = Label(marco5, text="Tiempo =" )

hora.pack(side=LEFT)

rolex = Label(marco5, text="")

rolex.pack(side=LEFT)

marco6 = Frame(vent)

marco6.pack()

si2 = Button(marco6, text="Prender",command=prender2)

si.pack(side=LEFT)

no2 = Button(marco6, text="Apagar",command=apagar2)

no.pack(side=LEFT)

temperatura = Label(marco6, text="Temperatura=" )

temperatura.pack(side=LEFT)

temper = Label(marco6,text="")

temper.pack(side=LEFT)

tiles\_letter = ['']

def add\_letter():

tiempo = str( int(time()) - tinicial -1)

rand = random.choice(tiles\_letter) + tiempo

rolex.config(text=rand)

vent.after(500, add\_letter)

def lectura():

if pto == 1:

ser.write("AC.")

a = '['

if ser.read() == '[':

for x in range(0,15):

a = a + ser.read()

acce.config(text=a)

vent.after(1000, lectura)

if pto == 0:

acce.config(text="Desactivado")

vent.after(1000, lectura)

def temp():

if tem == 1:

ser.write("TP.")

a = ""

if ser.read() == '[':

for x in range(0,10):

if ser.read() == ']':

break

a = a + ser.read()

a = a + '°'

temper.config(text=a)

vent.after(1000,temp)

if pto == 0:

temper.config(text="Desactivado")

vent.after(1000, temp)

vent.after(0, add\_letter)

vent.after(1000, lectura)

vent.after(1000,temp)

vent.mainloop()

Arduino ------------------------------

/\*

Control de motores de CC con CI L298

\*/

// Pines de entrada para CI L298

#define MOTOR\_CTL1 8 // I1 Input 1

#define MOTOR\_CTL2 10 // I2 Input 1

#define MOTOR\_PWM 9 // EA Enable A

#define MOTOR\_CTR1 5 // I1 Input 2

#define MOTOR\_PWM1 6 //

#define MOTOR\_CTR2 7 // I2 Input 2

#define MOTOR\_DIR\_FORWARD 0 // Adelante

#define MOTOR\_DIR\_BACKWARD 1 // Atras

#define MOTOR\_DIR\_RIGHT 0 // Adelante

#define MOTOR\_DIR\_LEFT 1 // Atras

int v=255;

String inputString = "";

boolean stringComplete = false;

String c;

int x, y, z;

void setup(){

// Configuracion de pines para control del motor

// Control de sentido de giro

pinMode(MOTOR\_CTL1,OUTPUT);

pinMode(MOTOR\_CTL2,OUTPUT);

// Control de velocidad

pinMode(MOTOR\_PWM,OUTPUT);

// Control de sentido de giro del motor 2

pinMode(MOTOR\_CTR1,OUTPUT);

pinMode(MOTOR\_CTR2,OUTPUT);

pinMode(MOTOR\_PWM1,OUTPUT);

Serial.begin(9600);

inputString.reserve(200);

Serial.println('xx,yy,zz');

}

// Control de velocidad mediante PWM

// 0 < motor\_speed < 255

void setSpeed(byte motor\_speed)

{

analogWrite(MOTOR\_PWM, motor\_speed);

analogWrite(MOTOR\_PWM1, motor\_speed);

}

void turnAngle(boolean direction)

{

switch (direction)

{

case MOTOR\_DIR\_RIGHT:

{

digitalWrite(MOTOR\_CTL1,HIGH);

digitalWrite(MOTOR\_CTR1,HIGH);

digitalWrite(MOTOR\_CTL2,LOW);

digitalWrite(MOTOR\_CTR2,LOW);

}

break;

case MOTOR\_DIR\_LEFT:

{

digitalWrite(MOTOR\_CTL1,LOW);

digitalWrite(MOTOR\_CTR1,LOW);

digitalWrite(MOTOR\_CTL2,HIGH);

digitalWrite(MOTOR\_CTR2,HIGH);

}

break;

}

}

// Cambiar el sentido de giro

void motorMove(boolean direction)

{

switch (direction)

{

case MOTOR\_DIR\_FORWARD:

{

digitalWrite(MOTOR\_CTL1,LOW);

digitalWrite(MOTOR\_CTR1,HIGH);

digitalWrite(MOTOR\_CTL2,HIGH);

digitalWrite(MOTOR\_CTR2,LOW);

}

break;

case MOTOR\_DIR\_BACKWARD:

{

digitalWrite(MOTOR\_CTL1,HIGH);

digitalWrite(MOTOR\_CTR1,LOW);

digitalWrite(MOTOR\_CTL2,LOW);

digitalWrite(MOTOR\_CTR2,HIGH);

}

break;

}

}

// Frenar el motor

void motorStop()

{

setSpeed(255); // Habilitar

digitalWrite(MOTOR\_CTL1,HIGH); // Frenar

digitalWrite(MOTOR\_CTL2,HIGH);

digitalWrite(MOTOR\_CTR1,HIGH);

digitalWrite(MOTOR\_CTR2,HIGH);

}

// Motor libre

void motorFree()

{

setSpeed(255); // Habilitar

digitalWrite(MOTOR\_CTL1,LOW); // Liberar

digitalWrite(MOTOR\_CTL2,LOW);

digitalWrite(MOTOR\_CTR1,LOW);

digitalWrite(MOTOR\_CTR2,LOW);

}

void loop(){

setSpeed(v);

if (stringComplete){

c = inputString.substring(0,2);

if (c=="Fd"){

motorFree();

delay(100);

motorMove(MOTOR\_DIR\_FORWARD);

Serial.println("Fd")

}

else if (c=="Bd"){

motorFree();

delay(100);

motorMove(MOTOR\_DIR\_BACKWARD);;

Serial.println("Bd");

}

else if(c=="Lf"){

motorFree();

delay(100);

turnAngle(MOTOR\_DIR\_LEFT);

Serial.println("Lf");

}

else if(c=="Rt"){

motorFree();

delay(100);

turnAngle(MOTOR\_DIR\_RIGHT);

Serial.println("Rt");

}

else if(c=="St"){

motorStop();

Serial.println("St");

}

else if(c == "AC"){

x = analogRead(0); // read analog input pin 0

y = analogRead(1); // read analog input pin 1

z = analogRead(2); // read analog input pin 1

String a;

a = '[' + String(x) + 'X' + ',' + String(y) + 'Y' + ',' + String(z) + 'Z' +']' ;

Serial.println(a);

}

else if(c == "TP"){

tempC = analogRead(pinLM35);

tempC = (5.0 \* tempC \* 100.0)/1024.0;

temp = '[' + String(tempC) + ']' ;

Serial.println(temp);

}

inputString = "";

c = "";

stringComplete = false;

}

}

void serialEvent() {

while (Serial.available()) {

char inChar = (char)Serial.read();

inputString += inChar;

if (inChar == '.'){

stringComplete = true;

}

}

}