



Tecnológico de Monterrey

Raspberry y I2C

Diseño de sistemas en chip

Guillermo Ramos Domínguez A01736484

Arick Morelos del Campo A01735692

Fernando Estrada Silva A01736094

IRS

2/04/23

INDICE

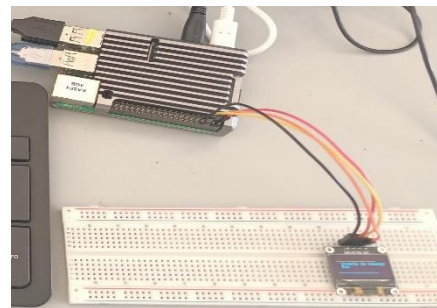
índice	1
Introducción	2
Desarrollo	2
Codigo	2-6
Terminal	6-7
Conclusiones	8

Introducción

Durante la clase de sistemas en chips nos tocó realizar una actividad la cual consistía en una programa similar a una actividad anterior. Con la diferencia de que en esta ocasión los mensajes se tienen que proyectar en nuestra pantalla oled. La cual conectaremos a nuestra tarjeta raspberry por medio de una protoboard y unos cuantos cables jumper.

Desarrollo

Lo primero hacer la instalación del programa Arduino IDE en una de las laptops de nuestro equipo. Luego dentro del software programamos los mensaje que queremos que se escriban en la terminal por medio de la tarjeta Arduino. Lo siguiente fue hacer la conexión de la tarjeta con nuestro monitor para poder visualizar su contenido. A la tarjeta raspberry, conectamos una fuente de alimentación de un cable tipo usb C. Además, después de haber instalado nuestro programa dentro de la tarjeta hay que conectar nuestro tarjeta Arduino UNO a uno de los puertos UART de la tarjeta Raspberry. Además de esto con ayuda de una protoboard conectamos a la raspberry nuestra pantalla oled. Finalmente desarrollamos el programa en Python que se encargara de llamar al programa dentro de la Arduino y desplegar la información en la terminal.



Código en Python:

```
from board import SCL, SDA

import busio

from PIL import Image, ImageDraw, ImageFont

import adafruit_ssd1306
```

```

import time as t

import serial

i2c = busio.I2C(SCL, SDA)

disp = adafruit_ssd1306.SSD1306_I2C(128, 64, i2c)

disp.fill(0)

disp.show()


def names():

    image = Image.new('1', (128, 64))

    width = disp.width

    height = disp.height

    image = Image.new('1', (width, height))

    draw = ImageDraw.Draw(image)

    font = ImageFont.load_default()

    draw.rectangle((100,100,width,height), outline=100, fill=255)

    draw.text((15, 16), 'Fernando Estrada', font=font, fill=255)

    draw.text((15, 26), 'Arick Morelos', font=font, fill=255)

    draw.text((15, 36), 'Guillermo Ramos', font=font, fill=255)

    # Muestra Texto

    disp.image(image)

    disp.show()

    t.sleep(2)

def entrada():

```

```
image1 = Image.new('1', (128, 64))

width = disp.width

height = disp.height

image1 = Image.new('1', (width, height))

#draw.rectangle((0,0,width,height), outline=0, fill=0)

draw = ImageDraw.Draw(image1)

font = ImageFont.load_default()

#raw.text((30, 16), 'Oskar Villa', font=font, fill=255)

draw.line((5,15, 100, 15), fill=255)

draw.line((5,50, 100, 50), fill=255)


draw.text((15, 16), 'Inserte su mensaje', font=font, fill=255)

message = input("Inserte su mensaje : ")


if (len(message) >= 50):

    draw.text((15, 16), 'Mensaje mayor a 50 caracteres', font=font, fill=255)

    t.sleep(3)

else:

    draw.text((15, 26), message, font=font, fill=255)


# Muestra Texto

disp.image(image1)
```

```
disp.show()
```

```
t.sleep(3)
```

```
def uart():
```

```
    image2 = Image.new('1', (128, 64))
```

```
    width = disp.width
```

```
    height = disp.height
```

```
    #image = Image.new('1', (width, height))
```

```
    draw2 = ImageDraw.Draw(image2)
```

```
    font = ImageFont.load_default()
```

```
    #draw2.rectangle((100,100,width,height), outline=100, fill=255)
```

```
    s= serial.Serial("/dev/ttyACM0",9600)
```

```
    print(s.name)
```

```
    for i in range (7):
```

```
        x = s.readline()
```

```
        print (x)
```

```
    s.close()
```

```
    # Muestra Texto
```

```
    disp.image(image2)
```

```
    disp.show()
```

```
    t.sleep(2)
```

```
names()
```

entrada()

uart()**Código Arduino**

void setup() {

Serial.begin(9600);

}

void loop() {

Serial.println("mexicanos"); //imprime el mensaje

Serial.println("al grito"); //imprime el mensaje

Serial.println("de guerra"); //imprime el mensaje

Serial.println(" El acero aprestad"); //imprime el mensaje

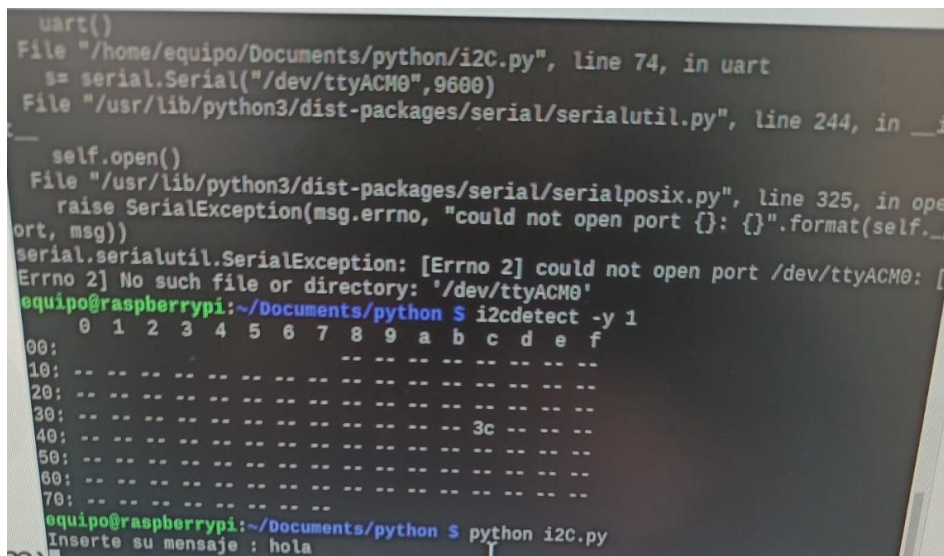
Serial.println("y el bridon"); //imprime el mensaje

Serial.println("y retiemble en su centro la tierra"); //imprime el mensaje

Serial.println("al sonoro rugir del cannon"); //imprime el mensaje

delay (4000); //crea un delay de 4000 milisegundos

Funcionamiento desde la terminal:



```
uart()
File "/home/equipo/Documents/python/i2C.py", line 74, in uart
s= serial.Serial("/dev/ttyACM0",9600)
File "/usr/lib/python3/dist-packages/serial/serialutil.py", line 244, in __i
self.open()
File "/usr/lib/python3/dist-packages/serial/serialposix.py", line 325, in ope
raise SerialException(msg.errno, "could not open port {}: {}".format(self._p
ort, msg))
serial.serialutil.SerialException: [Errno 2] could not open port /dev/ttyACM0: [
Errno 2] No such file or directory: '/dev/ttyACM0'
equipo@raspberrypi:~/Documents/python $ i2cdetect -y 1
00:  0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  3c  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
equipo@raspberrypi:~/Documents/python $ python i2C.py
Inserte su mensaje : hola
```

En la vista desde la terminal podemos ver como al iniciar el programa inmediatamente nos pide como usuario que escribamos el mensaje que queremos que se proyecte en la pantalla oled. Para posteriormente imprimir los mensajes que designamos en el programa de la actividad anterior.



Antes de proyectar el mensaje establecido por el usuario. Primero proyecta dentro de la pantalla el nombre de todos los integrantes del equipo. Tal y como lo establecimos dentro del programa en Python con el fin de verificar el correcto funcionamiento de la pantalla oled.



Una vez que el programa realiza el paso anterior dentro de la terminal nos arroja el texto de que quiere que nosotros escribamos un mensaje para que se proyecte en la pantalla. Después de que el usuario escribe su mensaje el programa automáticamente lo manda a la pantalla oled, para que así esta lo proyecte. Esto dura unos cuantos segundos ya que después el mensaje desaparece mostrando así el fin de nuestro programa.

Conclusiones

Esta si bien era una actividad con una complejidad mayor a la anterior. Era relativamente sencilla de hacer. Debido a que la problemática más grande que se nos presento fue el entender como hacer funcionar la pantalla oled estableciendo la conexión entre esta y la tarjeta raspberry. Sin embargo fue algo interesante ya que a pesar de su diminuto tamaño la calidad de la imagen de la pantalla era mayor a la que pensábamos en un inicio que esta iba a alcanzar.