

Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Puebla



Diseño de sistemas en chip
TE2003B.501

Dr. Emmanuel Torres

Raspberry y I2C

RoboInges | Integrantes:

Alan Iván Flores Juárez	A01736001
Givka Morales López	A01423534
Ulises Hernandez Hernández	A01735823

Abril 21, 2023

Raspberry y I2C

Introducción

Continuando con la actividad de comunicación mediante UART, en esta actividad se trabajara con el protocolo de comunicación I2C con el objetivo de desplegar texto en una pantalla OLED de 128x64, para esto se requiere trabajar de manera síncrona con ambos protocolos de comunicación ya que se requiere de la comunicación del Arduino con la RaspBerry y la comunicación de la RaspBerry con la pantalla OLED.

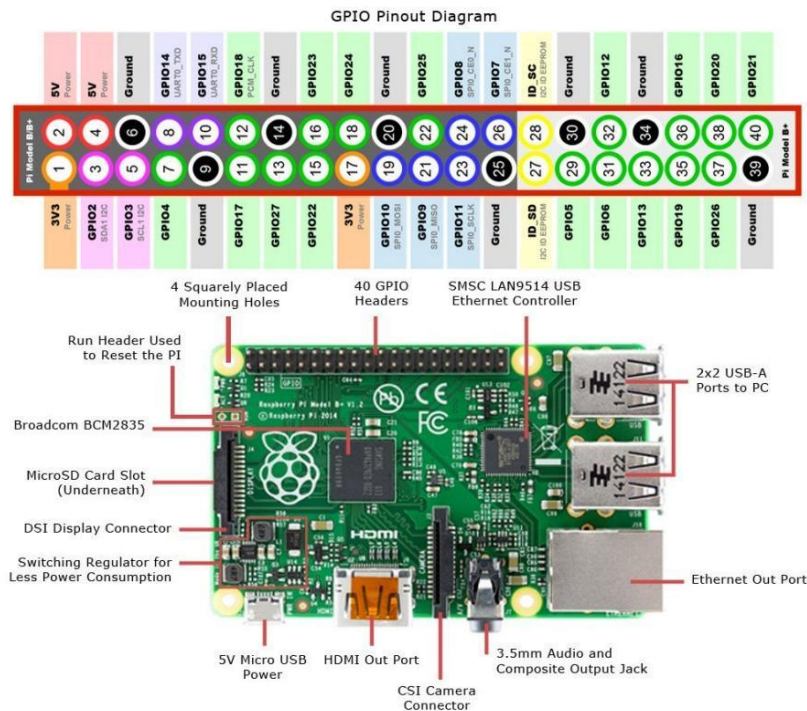
El objetivo de esta actividad es desplegar en la pantalla el nombre de los integrantes del equipo, así como tras un pequeño delay, recibir mensajes de la Arduino mediante la comunicación UART y también ser desplegados en la pantalla.

Desarrollo

Como se mencionó anteriormente, para realizar esta práctica, fue necesario realizar prácticas para aprender a utilizar una RaspBerry Pi, un Arduino Mega y se utilizaron los siguientes periféricos para las conexiones correspondientes.



El monitor, teclado y mouse, los ocupamos para poder ver el sistema de la Raspberry Pi, en la cual, después de entrar, podemos cargar y correr archivos por medio de la memoria micro SD. En el caso para esta actividad utilizamos el protocolo de comunicación I2C, la tarjeta arduino mega y una pantalla oled, para enviar mensajes por medio de la pantalla oled.



Primero conectamos la raspberry a la pantalla oled. En las siguientes imágenes se muestran los propósitos de cada pin.

Como se puede visualizar en las imágenes, se muestran los Pinout de la pantalla y de la raspberry. De acuerdo a eso conectamos la pantalla a voltaje de 3.3 volts en el pin 1, el GND lo conectamos al pin 6, el SCL (Serial Clock) al pin GPIO3 y el SDA (Serial Data) al pin GPIO2, estos últimos 2 tienen su pin reservado con ese mismo propósito y se señala en el pinout de la raspberry. Una vez conectado la pantalla oled, hicimos pruebas para mandar datos a la pantalla desde un código escrito en python y ya teniendo eso, conectamos la arduino mega con un código previamente cargado hecho arduino, para que mande datos por el puerto UART a la raspberry y nosotros podamos mandar los datos recibidos a la pantalla oled, para hacer esto posible creamos varias imágenes en el código en python para que no se sobrescriba el texto.

- Código Arduino:

```
enviar_datos_serial
void setup() {
  Serial.begin(9600);

}

void loop() {
  Serial.println("Holaaaaa");
  Serial.println("mundooo");
  Serial.println("Esta es la materia de:");
  Serial.println("Diseno");
  Serial.println("en");
  Serial.println("Chip");
  Serial.println("Hasta luego companero");
  delay(3000);
}
```

Este código se utiliza para establecer la comunicación serial entre el Arduino y la RaspBerry Pi, primero establece la velocidad de transmisión de datos o baudios a los cuales se comunican ambos dispositivos en la función setup() como se requiere en el protocolo UART para finalmente mandar los 7 mensajes arbitrarios en la función loop utilizando Serial.println("") y un delay de 3 segundos una vez enviados los 7 mensajes.

- Código Python:

```
1 import serial
2 from board import SCL, SDA
3 import busio
4 from PIL import Image, ImageDraw, ImageFont
5 import adafruit_ssd1306
6 import time
7 i2c = busio.I2C(SCL, SDA)
8 disp = adafruit_ssd1306.SSD1306_I2C(128, 64, i2c)
9 #crea pantalla en negro
10 disp.fill(0)
11 disp.show()
12 width = disp.width
13 height = disp.height
14 #Crea nueva imagen
15 image = Image.new('1', (width, height))
16 draw = ImageDraw.Draw(image)
17 font = ImageFont.load_default()
18 draw.text((10, 0), '*****', font=font, fill=255)
19 draw.text((10, 10), 'Ulises Hernandez', font=font, fill=255)
20 draw.text((10, 20), 'Alan Flores', font=font, fill=255)
21 draw.text((10, 30), 'Givka Morales', font=font, fill=255)
22 draw.text((10, 40), '*****', font=font, fill=255)
23 # Muestra Texto
24 disp.image(image)
25 disp.show()
26 time.sleep(3)
```

```

27 #Crea pantalla en negro
28 disp.fill(0)
29 disp.show()
30 # Crea nueva imagen
31 image1 = Image.new('1', (width, height))
32 draw1 = ImageDraw.Draw(image1)
33 font = ImageFont.load_default()
34 #Lee el puerto Serial
35 s = serial.Serial('/dev/ttyUSB0',9600)
36 x=0
37 for i in range(7):
38     lectura = s.readline()
39     draw1.text((10, x), lectura, font=font, fill=255)
40     x=x+7
41 s.close()
42 # Muestra Texto
43 disp.image(image1)
44 disp.show()

```

Este código en python establece el protocolo de comunicación I2C y asigna los pines de SCL y SDA y también lee datos del puerto serial que envía la tarjeta arduino mega, después básicamente lo que hace es crear imágenes en negro, sobre las cuales dibuja el texto ingresado en blanco y en las mismas funciones se definen los parametros de las coordenadas, para poner el texto más a la izquierda, derecha, arriba o abajo y para borrar texto, lo que hace es mejor crear otra imagen y despues de crear una nueva pantalla en negro, manda a llamar la nueva imagen creada con los nuevos dibujos.

Resultados

Con el código que se realizó y las conexiones debidamente hechas obtuvimos el resultado esperado. Al correr el código se muestra primero una pantalla donde se puede visualizar los nombres de los integrantes del equipo y al cabo de 3 segundos, la pantalla se pone en negro y cambia a otra imagen en la cual se muestran las 7 lectura del puerto serial con su debido espacio entre linea y linea de texto.

- Video de funcionamiento:



[Link](#)

Conclusión

Esta actividad nos ayudó a implementar diversos protocolos de comunicación entre diversos dispositivos y periféricos, como lo son el Arduino, la RaspBerry y la pantalla OLED para poder desplegar en la pantalla los mensajes requeridos, en nuestro caso los nombres de los integrantes del equipo, así como los 7 mensajes transmitidos por el Arduino a la RaspBerry, utilizando el mismo proceso descrito en la actividad anterior aunado a la comunicación por I2C con el display. Para esto se utilizaron los puertos designados en la RaspBerry designados para este tipo de comunicaciones, así como requirió de la soldadura de los pines en el display para su funcionamiento. Al final fue una actividad muy completa que involucra aplicar los conocimientos adquiridos en este bloque y los anteriores de la carrera en cuanto a escritura de código y protocolos de comunicación entre dispositivos.