PROTOTIPO INTEGRADO DE MONITOREO Y REGISTRO DE DATOS METEOROLOGICOS EN EDIFICACIONES DEL SIGLO XVI

José Alfredo Rodríguez Pérez, Jesús Contreras González, Víctor Manuel Vázquez Báez

Facultad de Ingeniería – BUAP

Ingeniería electrónica - ITSTB

Resumen

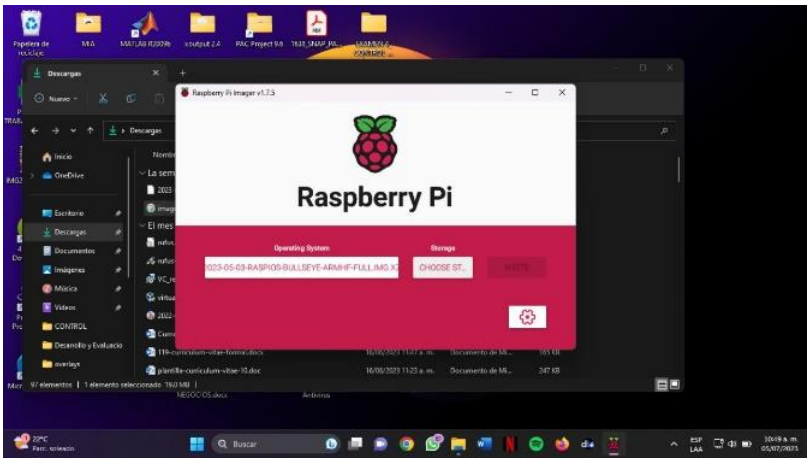
El proyecto realizado es experimental y en este se abarcan áreas de programación, electrónica, infraestructura y edificaciones y de habilidades de procesos de manufactura para el desarrollo de estaciones de monitoreo y registros de datos meteorológicos en edificaciones de alto riesgo ante algún posible fenómeno natural que llegue a suceder en un futuro, y estas puedan ser de ayuda para prevenir algún desastre mayor.

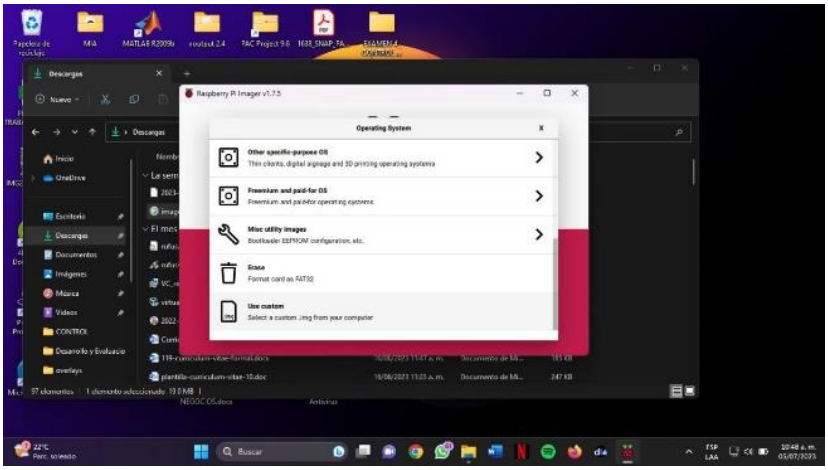
Introducción

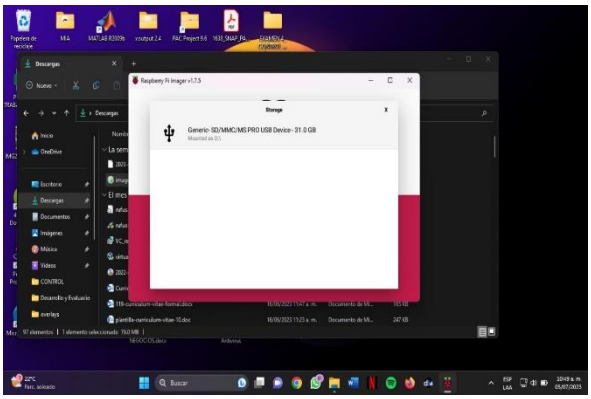
Hoy en día, el centro del país se encuentra susceptible a algún movimiento sísmico y así también, en las diferentes edificaciones del centro se encuentran muy vulnerables a padecer las consecuencias de no dar mantenimiento a ciertos edificios, como puede ser la humedad, la temperatura o inclusive, algunas edificaciones con daños de los sismos del 2017 que no han sido atendidos y estos pueden ser los causantes de próximos accidentes, es por esto que se busca la creación de estos prototipos para crear una base de datos de los diversas vibraciones que suceden en la zona y estas afectan a las edificaciones de la zona es por eso, que se elabora el siguiente manual para que los próximos años se le pueda dar seguimiento al proyecto.

Metodología

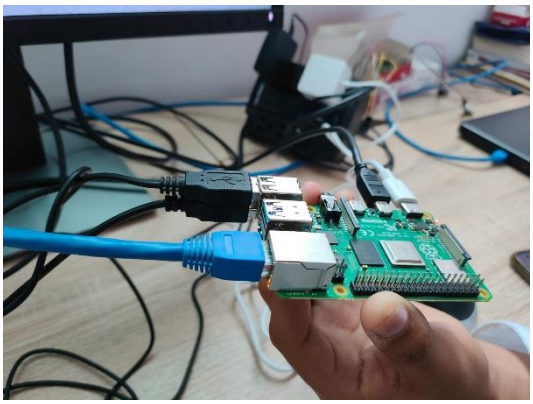
1. Como primer paso, se procede a soldar los sensores que se usaron en este proyecto; es decir, el acelerómetro MPU6050, el DHT22, el FC-28, el LM35. Con esto ya podemos proseguir con la configuración de la Raspberry Pi 4.
2. En segundo paso, se procede a configurar la Raspberry Pi 4, y el uso de una memoria SD , esto para instalar el sistema operativo, ya que para usar y programar la tarjeta Raspberry se necesita el SO de Raspbian OS, y para eso se descarga la imagen ISO de la pagina oficial de Raspberry (https://www.raspberrypi.com/software/) ahí se descargar la imagen ISO y se procede a instalar el SO en la memoria SD desde otro ordenador, tal como se muestra en las imágenes.

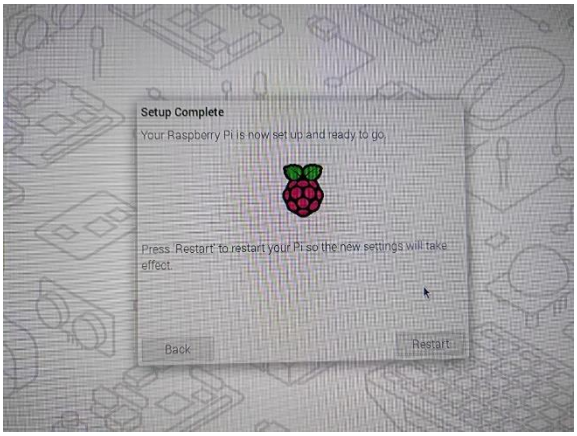


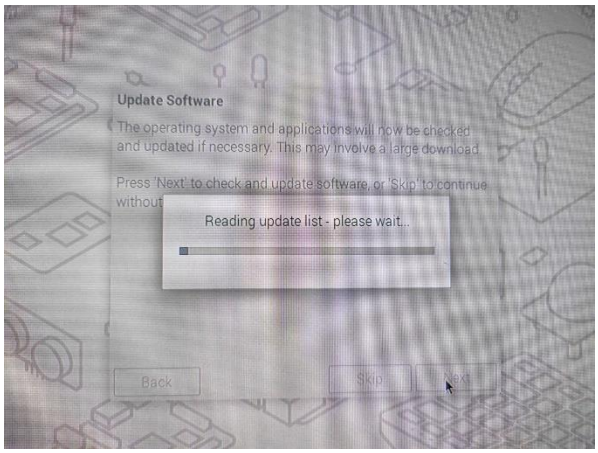


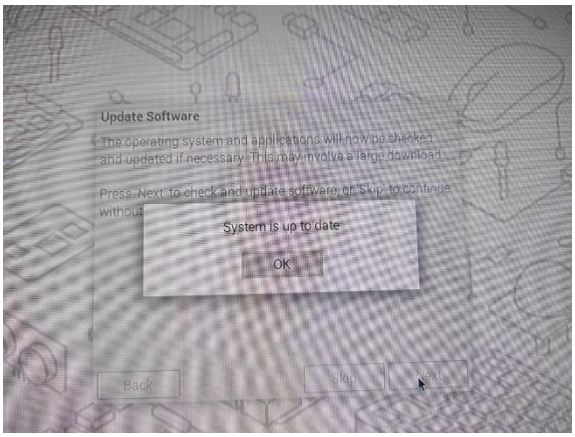


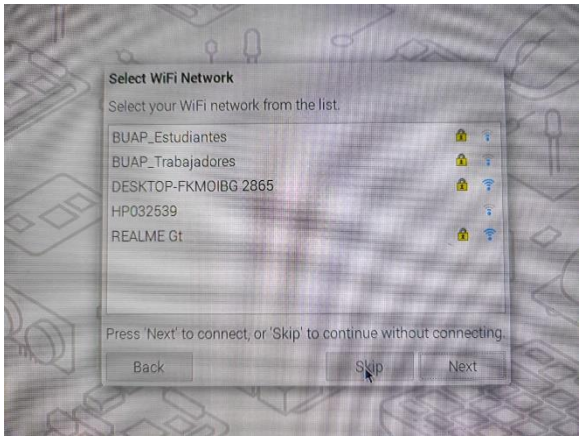
3. Una vez que el sistema operativo se instaló en la memoria SD, se procede a retirar del ordenador y se coloca en la Raspberry, y a su vez, se le conectan los periféricos de salida y entrada de datos, como lo son, el mouse, el teclado, el monitor, y el cable ethernet. Una vez que esta todo conectado, se conecta a la corriente la Raspberry y el monitor y se termina de configurar el sistema operativo para la placa.

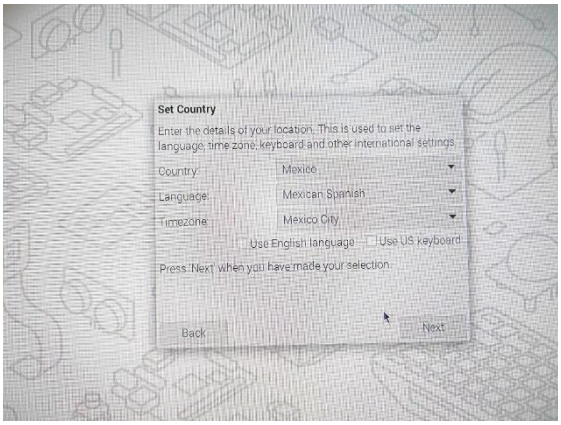


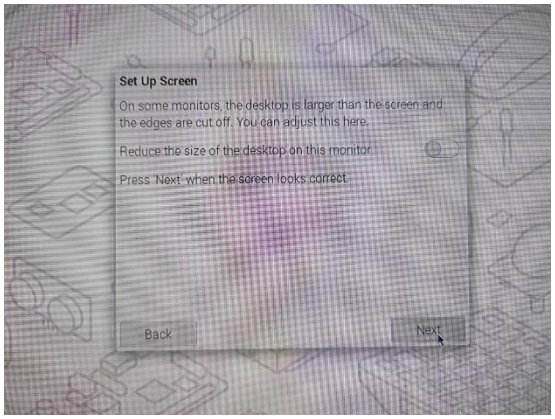

Setup Complete
Your Raspberry Pi is now set up and ready to go.
Press 'Restart' to restart your Pi so the new settings will take effect.
Back
Restart


Update Software
The operating system and applications will now be checked and updated if necessary. This may involve a large download.
Press 'Next' to check and update software, or 'Skip' to continue without
Reading update list - please wait...
Back
Skip
Next


Update Software
The operating system and applications will now be checked and updated if necessary. This may involve a large download.
Press 'Next' to check and update software, or 'Skip' to continue without
System is up to date
OK
Back
Skip
Next

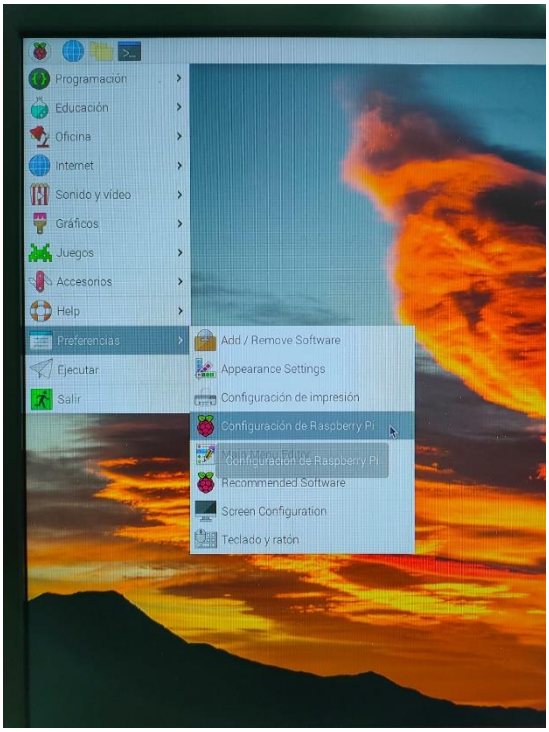

Select WiFi Network
Select your WiFi network from the list.
BUAP_Estudiantes
BUAP_Trabajadores
DESKTOP-FKMOIBG 2865
HP032539
REALME Gt
Press 'Next' to connect, or 'Skip' to continue without connecting.
Back
Skip
Next

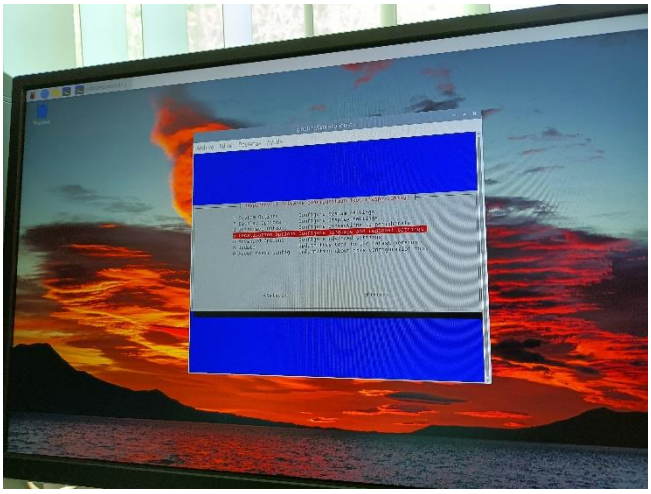

Set Country
Enter the details of your location. This is used to set the language, time zone, keyboard and other international settings.
Country:
Mexico
Language:
Mexican Spanish
Timezone:
Mexico City
Use English language
Use US keyboard
Press 'Next' when you have made your selection.
Back
Next

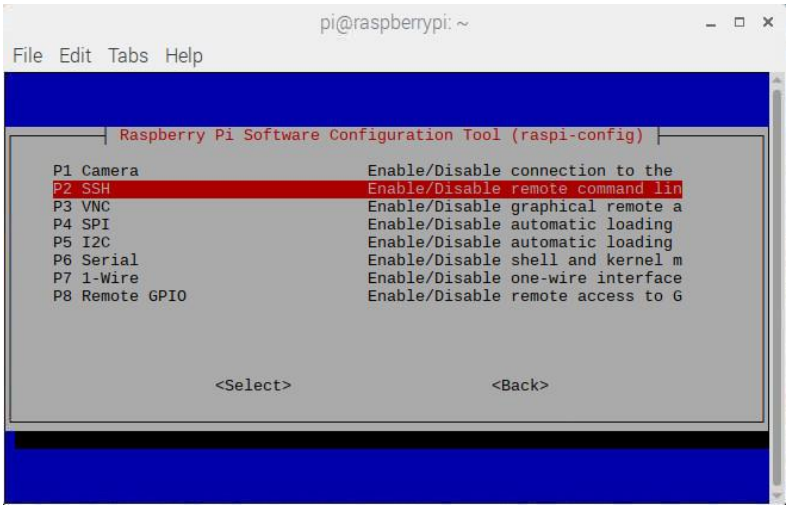

Set Up Screen
On some monitors, the desktop is larger than the screen and the edges are cut off. You can adjust this here.
Reduce the size of the desktop on this monitor
Press 'Next' when the screen looks correct.
Back
Next

En las imágenes anteriores, son capturas de los permisos y configuraciones que se le debe de hacer a la Raspberry para poder usarla con normalidad.

4. Una vez que la placa Raspberry se configuro correctamente, nos lanza a la pantalla inicia y procederemos a activar todos los puertos de comunicación para que podamos interactuar con los pines de la Pi 4 y así podamos usar los sensores como entradas y salidas de nuestro proyecto.

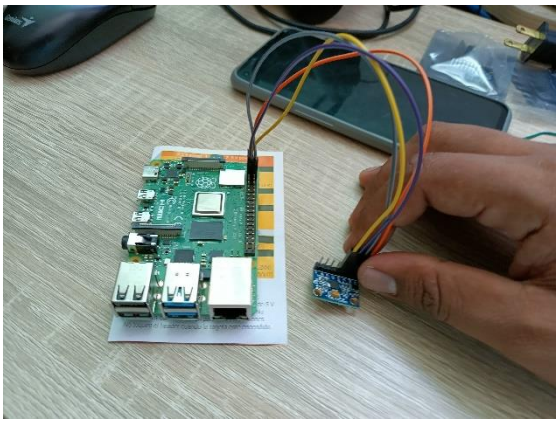


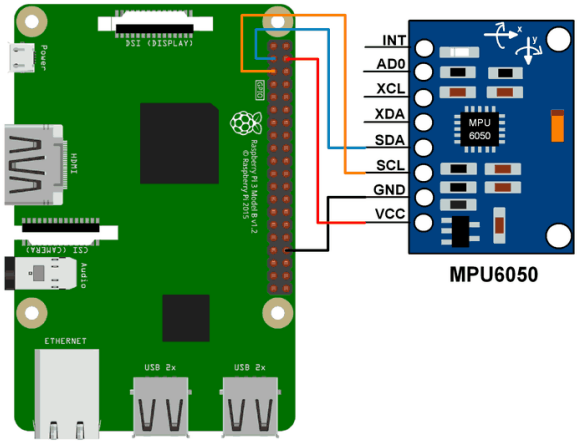


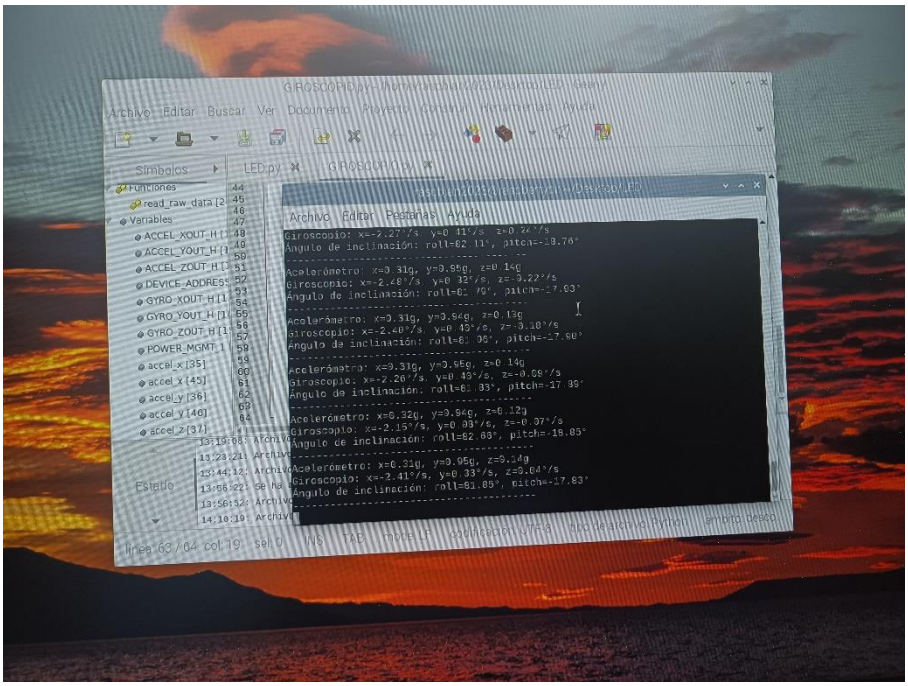


Aquí se configuran todos los puertos de comunicación, la SSH, I2C, Serial, y Remote GPIO, esto para poder usar los pines de entrada y salida de señales que tiene la Raspberry.

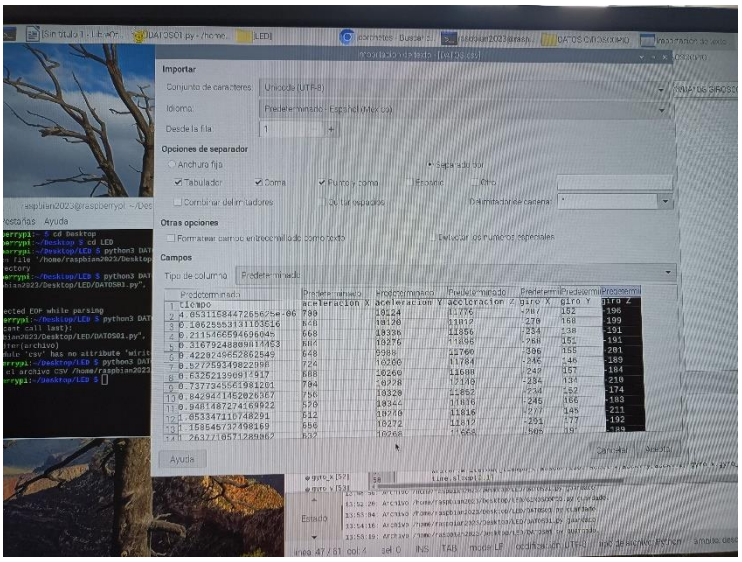
5. Por consiguiente, se prosigue a prosigue a probar los sensores y a hacer la programación, se inicia con el sensor MPU6050 (acelerómetro), el cual se muestra como harán las conexiones y hasta el final se anexa el código para sus futuras pruebas, justamente para hacer las pruebas, en la programación se usa el modo GPIO.BCM, que es el numero del pin de la Raspberry, para evitar tantas confusiones.

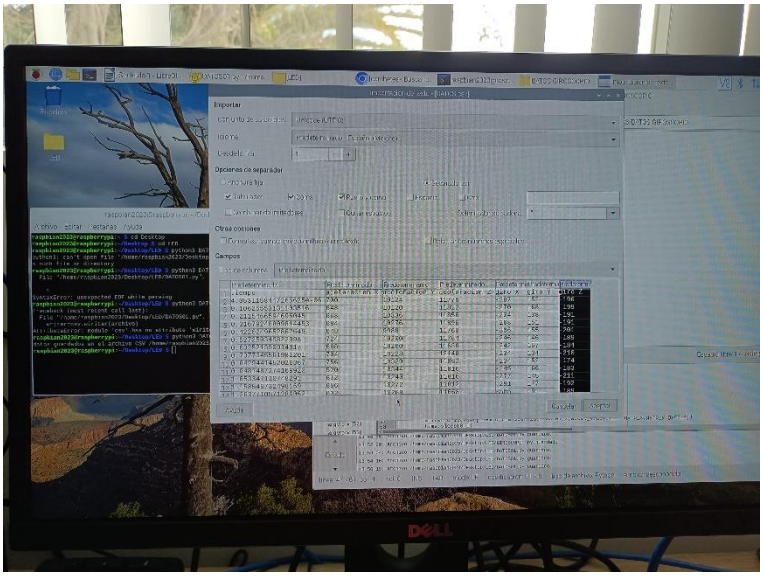




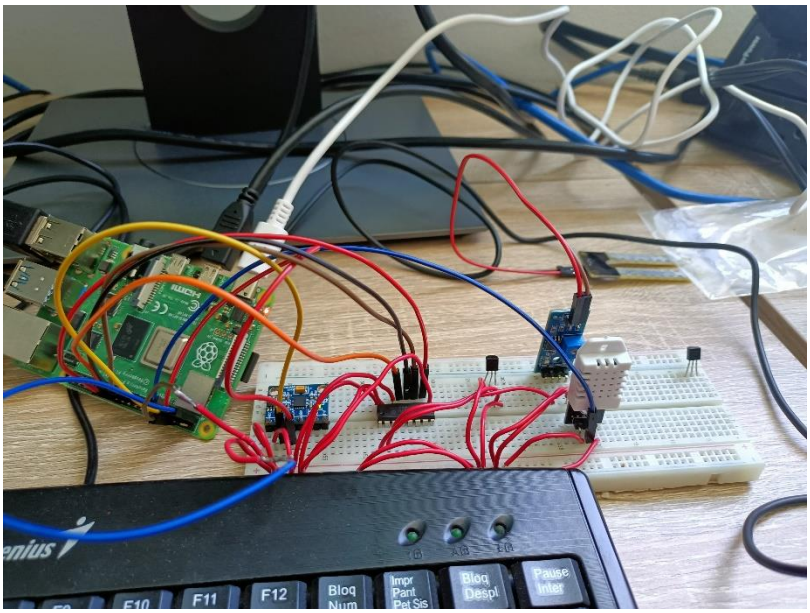


6. Como se requería que los datos se almacenaran en un archivo CSV, se prosiguió a cambiar la programación para exportar los datos a un archivo CSV y guardarlos en una carpeta especial para enviar los datos.

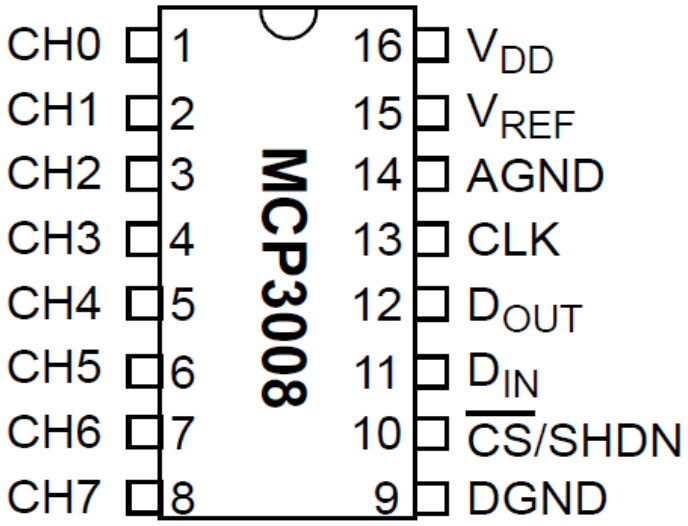




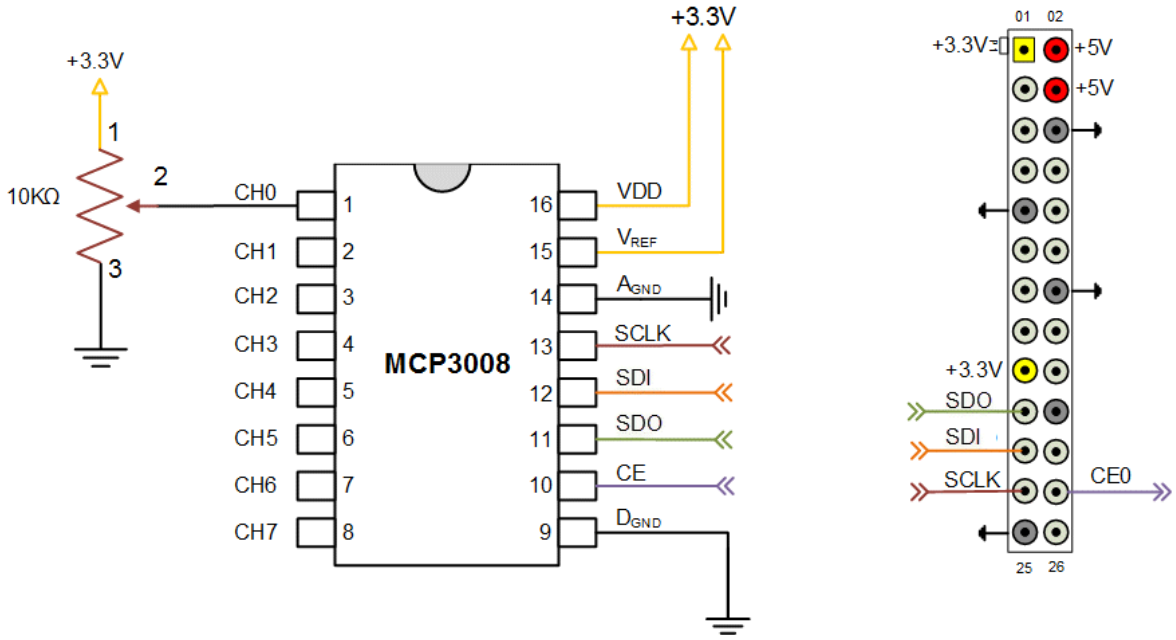
7. A continuación, se probaron todos los sensores que respectaban a la temperatura y humedad; el LM35, FC-28 y DHT22, estos detectan los valores requeridos para dentro de la edificación y se pueda ver y analizar en que condición se encuentra la estructura. En la imagen de abajo se observan todos los sensores ubicados en una protoboard.



Para poder usar el sensor LM35 y FC-28, se necesitó un convertidor de señales analógicas a digitales, esto porque los sensores anteriores son analógicos y la Raspberry solo puede leer datos digitales, en este caso se uso un circuito integrado denominado MCP3008, el cual puede leer hasta 8 señales analógicas y convertirlas a digital, y tiene una resolución de 10 bits.

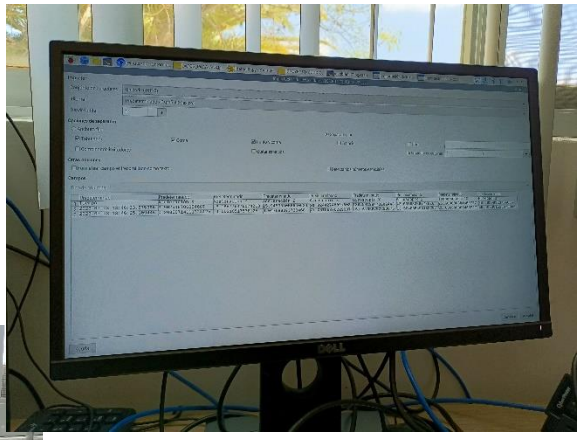


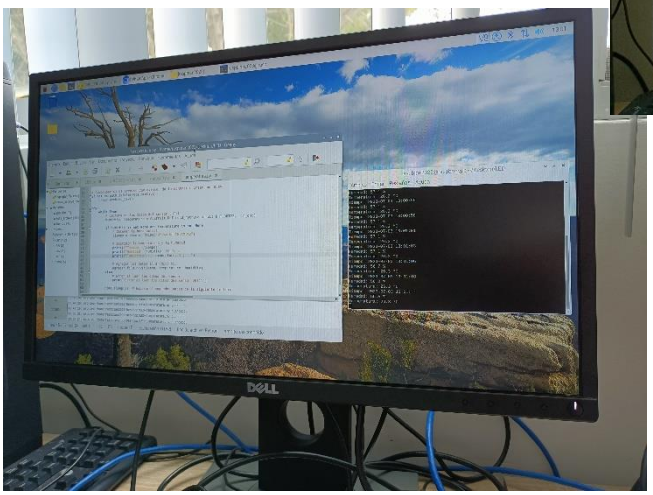
En esta imagen se observa el datasheet del integrado. Y a continuación se muestra como se va a conectar cada sensor.



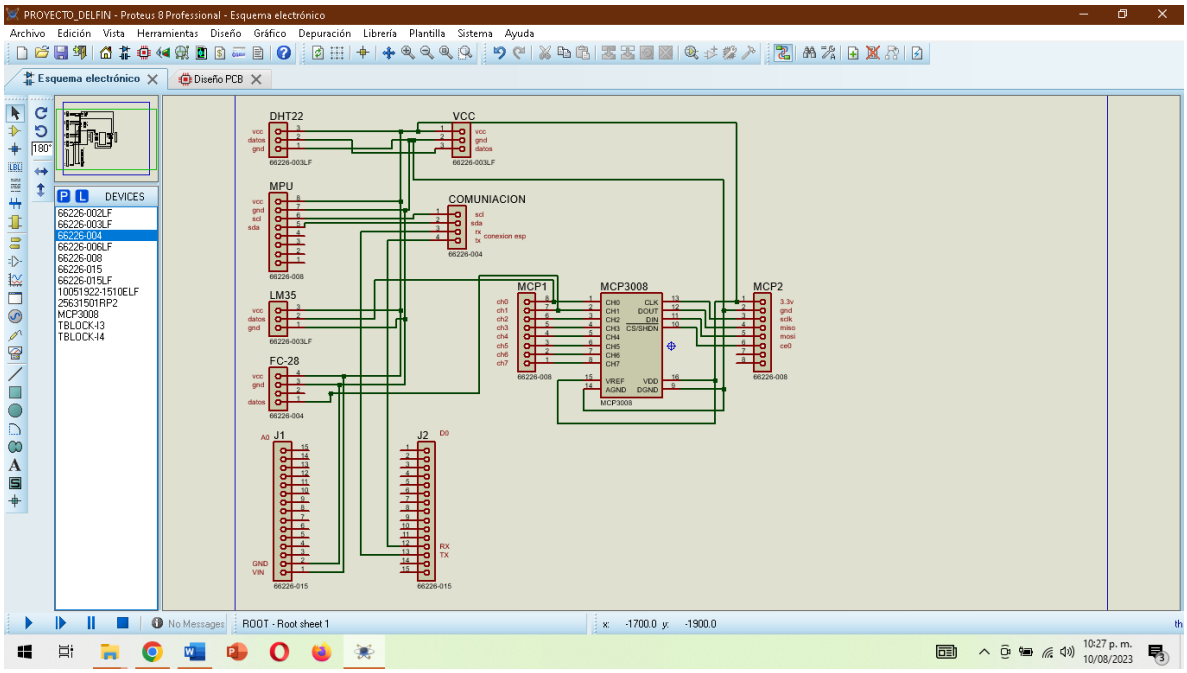
En los CH0 y CH1 ahí se conectan los sensores analógicos que se requieren, inclusive, pueden agregar más sensores analógicos para su análisis, cabe recalcar que, PARA LEER LAS SALIDAS DIGITALES, se especifican en la PROGRAMACIÓN y ahí mismo se decodifica las señales para su análisis.

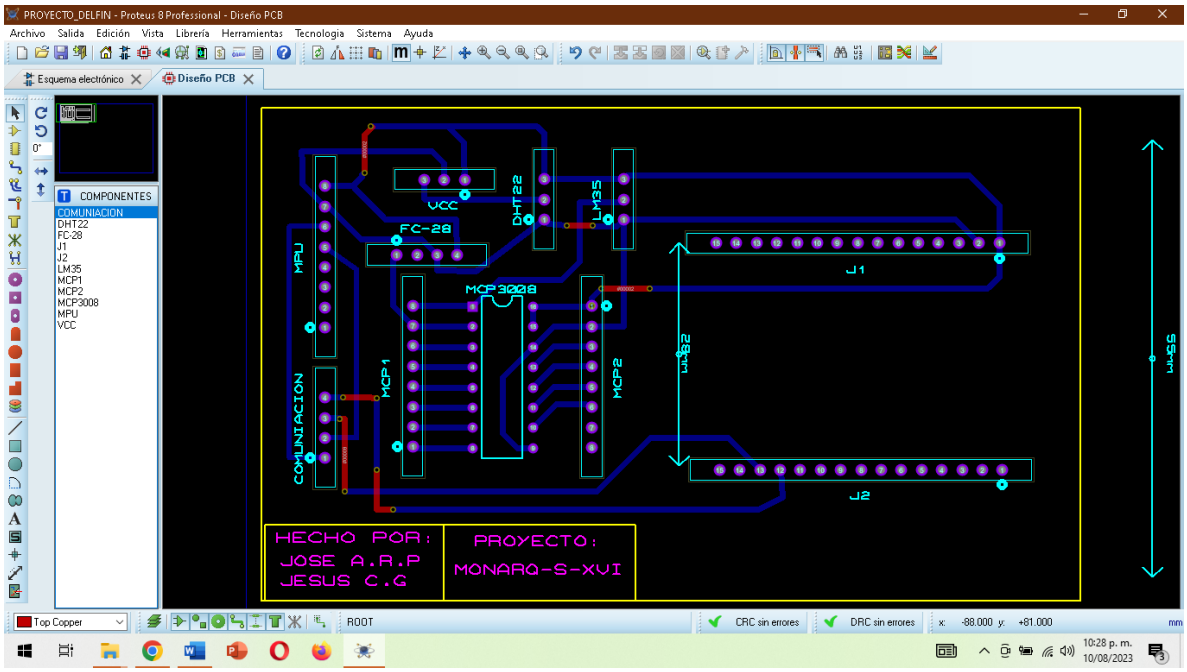
A continuación, se muestra los resultados, y como se menciono anteriormente, todos los códigos se anexarán hasta el final de este documento.





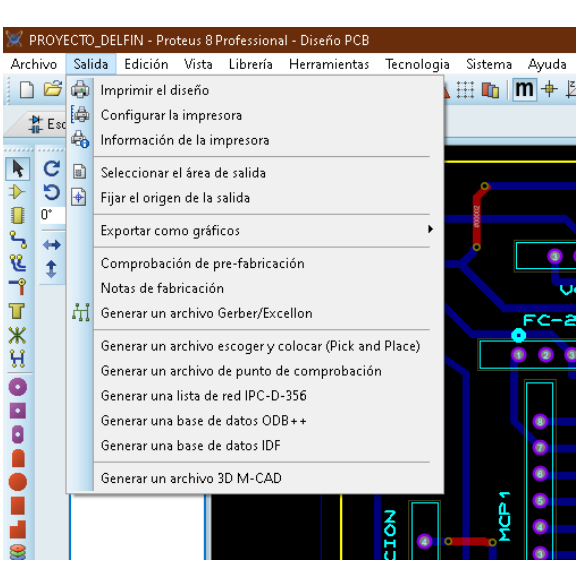
8. Al final, ya para juntar todo en un solo punto, se decidió crear una placa PCB para integrar todos los sensores y la Raspberry, entonces, se busco hacer la placa en un software llamado, ISIS de Proteus y se adjunta la evidencia como fue elaborada.

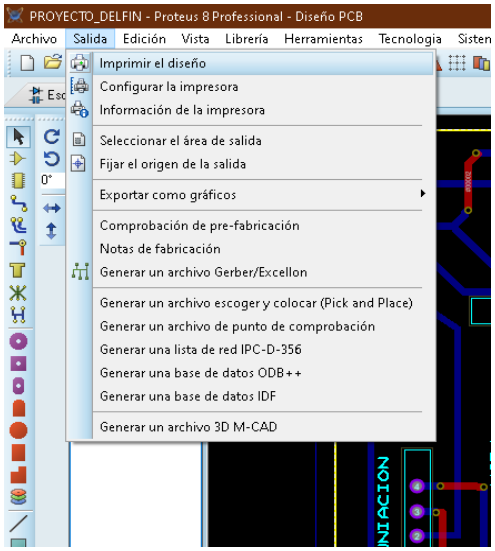




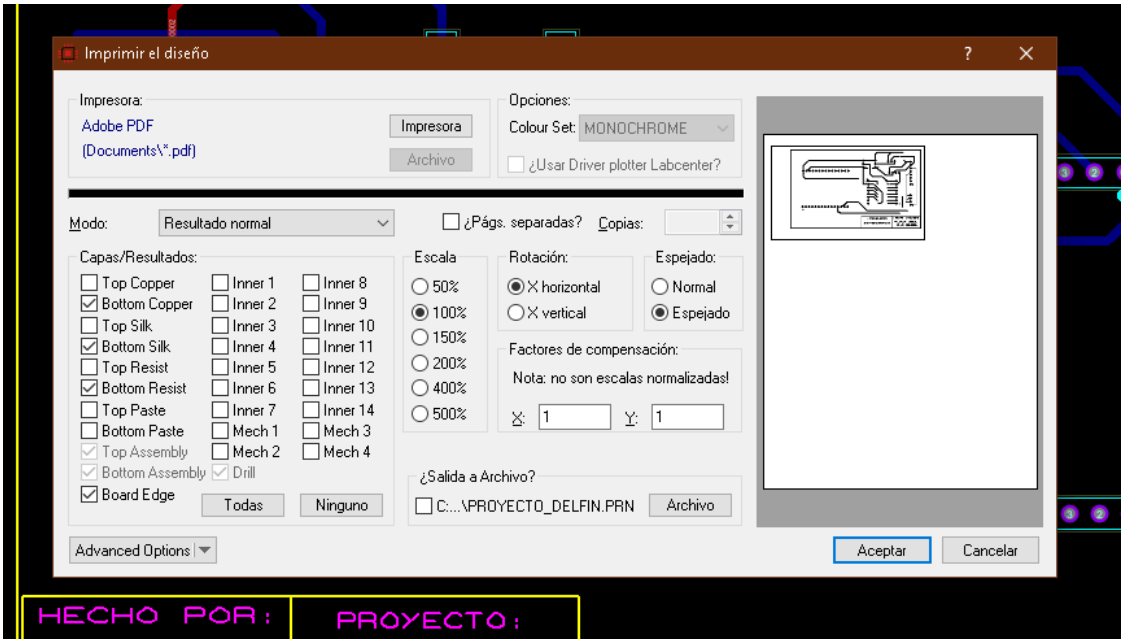
9. Como siguiente paso, se procede a descargar la salida del archivo en formato PDF, para imprimirlo en una impresora a tinta Tonner en una hoja tipo COUCHE, para poder aplicar la técnica del “planchado” y poder traspasar el

circuito a la placa fenólica. Y para eso, uno se dirige a la herramienta de “salida” y se selecciona la opción “Imprimir el diseño”



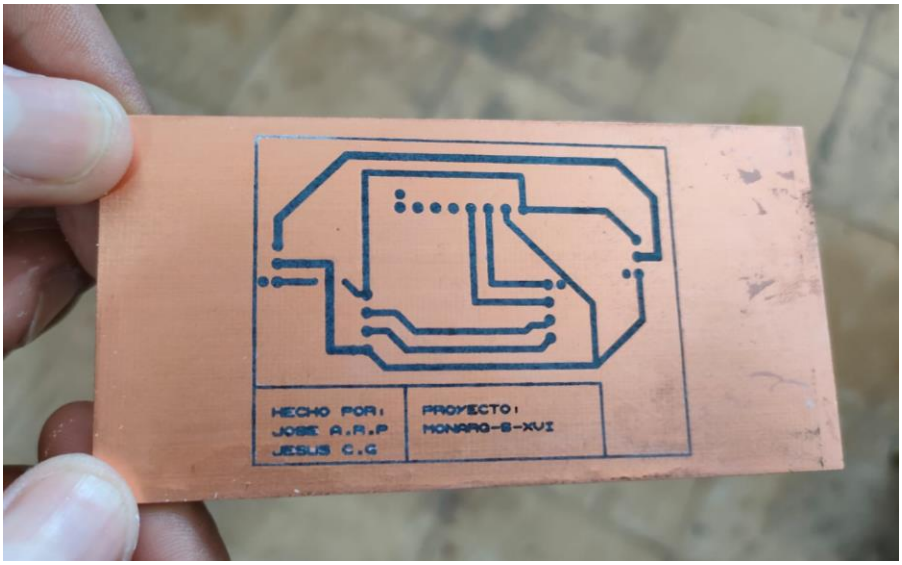


10. Cuando seleccionamos la opción de imprimir, nos arroja un cuadro de selección, y desmarcamos todas las opciones y solo marcamos las siguientes opciones y, sobre todo, marcar la opción “Espejado” porque al momento de imprimir y planchar, se va a invertir el circuito y saldrá como debería de salir.



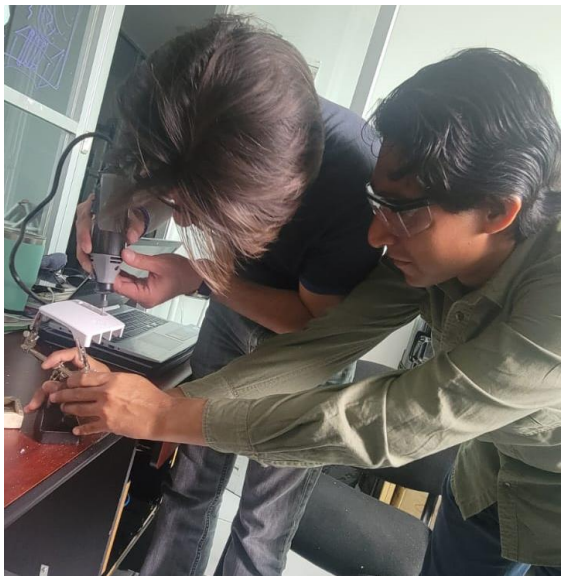
11. Se imprime y se procedemos a fijar el circuito a la placa y plancharla, con una plancha, un trapo y agregarles pequeñas gotas de agua, es como si fuera un tatuaje de los chicles, se aplica con agua y presión, justamente es lo

mismo, solo que con la plancha se le aplica calor por unos 20 a 25 minutos para comprobar que el circuito se paso correctamente a la placa.



12. Y, por último, se procede a pasar por el cloruro férrico y la placa se deja ahí y se mueve hasta que el exceso de cobre se le quita y solo queda las líneas negras con la pista de cobre lista para usarse.

13. Como ultimo paso, con un taladro mini y con apoyo de una broca de medida de 1/32” o de 1/16”, se procede a hacerle las perforaciones para los pines y que se puedan soldar a la placa.



14. Al final, todos los códigos se unieron para que todos los sensores funcionaran al mismo tiempo y que se activen en ciertos intervalos, a continuación, se dejan los códigos y las librerías a usar.
15. A continuación, se enlistan todas las librerías que hay que instalar en la Raspberry para que se pueda usar los sensores, todos se ejecutan desde la “LXTerminal” del sistema operativo, tal y como se dejan a continuación hay que escribirlo:
    - pip install adafruit-circuitpython-dht
    - pip install adafruit-circuitpython-mcp3xxx
    - pip install RPi.GPIO
    - sudo pip3 install gpiozero
    - sudo apt-get install python-spidev
    - sudo pip3 install Adafruit_DHT
    - sudo apt-get update
    - sudo apt-get upgrade
    - sudo apt-get install python-spidev
    - pip install mpu6050-raspberrypi
    - pip install pandas
    - pip install --upgrade pip
    - pip install smbus2
    - pip install numpy
    - pip install pandas
    - pip install smbus
    - pip install csv
    - sudo apt-get install python3
    - sudo apt-get install python3-rpi.gpio

16. Y por ultimo se anexa el código final, en el cual se integran todos los sensores que se están usando en este proyecto.

CODIGO FINAL

```
import os
import time
import spidev
import RPi.GPIO as GPIO
from gpiozero import MCP3008
import csv
import Adafruit_DHT
import smbus
import math

# Configuración del MCP3008
spi = spidev.SpiDev()
spi.open(0, 0)  # Bus SPI 0, dispositivo 0
spi.max_speed_hz = 1000000  # Velocidad de transferencia del SPI

# Definir el canal del MCP3008 conectado al sensor FC28
FC28_CHANNEL = 0

# Definir el pin GPIO conectado al pin D0 del sensor FC28
FC28_DO_PIN = 17

# Configuración del pin GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(FC28_DO_PIN, GPIO.IN)
```

```
# Crear el objeto MCP3008 para leer la temperatura desde el sensor LM35
lm35_reading = MCP3008(channel=0)

# Carpeta donde se guardarán los archivos CSV
folder_path = "/home/raspberry03/Desktop/PROYECTOS/END_DATAS"

# Comprobar si la carpeta existe. Si no existe, crearla.
if not os.path.exists(folder_path):
    os.makedirs(folder_path)

# Función para leer la humedad del suelo
def read_fc28_humidity():
    # Leer el valor del MCP3008 en el canal FC28_CHANNEL
    adc_value = spi.xfer2([1, (8 + FC28_CHANNEL) << 4, 0])
    # Convertir el valor leído en humedad (0-1023)
    humidity_value = ((adc_value[1] & 3) << 8) + adc_value[2]
    return humidity_value

# Dirección del dispositivo MPU-6050
DEVICE_ADDRESS = 0x68

# Registros del MPU-6050
POWER_MGMT_1 = 0x6B
ACCEL_XOUT_H = 0x3B
ACCEL_YOUT_H = 0x3D
ACCEL_ZOUT_H = 0x3F
GYRO_XOUT_H = 0x43
GYRO_YOUT_H = 0x45
```

```
GYRO_ZOUT_H = 0x47

# Inicializar el bus I2C
bus = smbus.SMBus(1)

# Configurar el MPU-6050 para operación
bus.write_byte_data(DEVICE_ADDRESS, POWER_MGMT_1, 0)

# Función para leer un valor de 16 bits desde el MPU-6050
def read_raw_data(reg):
    high = bus.read_byte_data(DEVICE_ADDRESS, reg)
    low = bus.read_byte_data(DEVICE_ADDRESS, reg+1)
    value = (high << 8) | low

    if value > 32767:
        value = value - 65536

    return value

try:
    for i in range(144):  # Realizar la recopilación de datos durante 24 horas (144 ciclos, 1 vez cada 10 minutos)
        # Obtener el tiempo de inicio para usar como marca de tiempo única para el archivo CSV
        start_time = time.strftime('%Y%m%d_%H%M%S')
        file_path = os.path.join(folder_path, f'datos_sensores_{start_time}.csv')
        with open(file_path, 'a', newline='') as csvfile:
            csvwriter = csv.writer(csvfile)
```

```
        # Escribir los encabezados en el archivo CSV si es un nuevo archivo
        if csvfile.tell() == 0:
            csvwriter.writerow(['Tiempo', 'Temperatura del edificio (C)', 'Humedad del edifico (%)', 'Temperatura del ambiente (C)', 'Humedad del ambiente (%)', 'Acelerómetro X', 'Acelerómetro Y', 'Acelerómetro Z', 'Giroscopio X', 'Giroscopio Y', 'Giroscopio Z'])

    # Leer los sensores 1 vez con un intervalo de 10 minutos
    # El bucle for j no es necesario en este caso
    # Se tomará una muestra cada 10 minutos

    # Leer la temperatura desde el sensor LM35
    temp_lm35_c = round((lm35_reading.value * 3.3) * 100, 2)

    # Leer la humedad del suelo desde el sensor FC28
    humidity = read_fc28_humidity()
    humidity_percentage = (1023 - humidity) * 100 / 1023

    # Leer la temperatura y humedad del DHT22
    dht22_pin = 4  # Conectar el DHT22 al pin GPIO 4
    dht22_humidity, dht22_temperature_c = Adafruit_DHT.read_retry(Adafruit_DHT.DHT22, dht22_pin)
    if dht22_humidity is not None and dht22_temperature_c is not None:
        dht22_temperature_c = round(dht22_temperature_c, 2)
        dht22_humidity_percentage = round(dht22_humidity, 2)
    else:
        dht22_temperature_c = -999
        dht22_humidity_percentage = -999

    # Leer los datos del acelerómetro y giroscopio
```

```
        accel_x = read_raw_data(ACCEL_XOUT_H) / 16384.0
        accel_y = read_raw_data(ACCEL_YOUT_H) / 16384.0
        accel_z = read_raw_data(ACCEL_ZOUT_H) / 16384.0
        gyro_x = read_raw_data(GYRO_XOUT_H) / 131.0
        gyro_y = read_raw_data(GYRO_YOUT_H) / 131.0
        gyro_z = read_raw_data(GYRO_ZOUT_H) / 131.0

        # Obtener el tiempo actual
        current_time = time.strftime('%Y-%m-%d %H:%M:%S')

        # Imprimir los valores en la consola
        print(f'Tiempo: {current_time}  Temp LM35: {temp_lm35_c}ºC  Humedad del suelo: {humidity_percentage}%')
        print(f'Temp DHT22: {dht22_temperature_c}ºC  Humedad DHT22: {dht22_humidity_percentage}%')
        print(f'Acelerómetro: X={accel_x}g, Y={accel_y}g, Z={accel_z}g')
        print(f'Giroscopio: X={gyro_x}°/s, Y={gyro_y}°/s, Z={gyro_z}°/s')
        print("----------------------------------------")

        # Abrir el archivo CSV en modo de apendizaje (append) y escribir los valores
        with open(file_path, 'a', newline='') as csvfile:
            csvwriter = csv.writer(csvfile)
            # Escribir los valores en el archivo CSV
            csvwriter.writerow([current_time, temp_lm35_c, humidity_percentage, dht22_temperature_c, dht22_humidity_percentage, accel_x, accel_y, accel_z, gyro_x, gyro_y, gyro_z])

        time.sleep(600)  # Esperar 10 minutos antes de comenzar el siguiente ciclo
```

```
except KeyboardInterrupt:
    spi.close()
    GPIO.cleanup()
    print("\nPrograma terminado.")
```

17. Para ponerlo en acción, hay que abrir un archivo nuevo en Scratch Geany y copias y pegas el código anterior, posteriormente se presionan las teclas “ctrl + S” esto para guardar el archivo, guardas en el archivo en donde tu lo vallas a guardar y lo guardas con la terminación “.py” es importante que se le agregue esta terminación.
18. En la terminal de Raspbian, abres la terminal y escribes los comandos para buscar y ejecutar el programa. Escribir “Cd Desktop”, después dar “enter”, posterior a esto escribir “Cd -nombre de la carpeta donde se guardó el archivo-” y enter y como ultimo paso, una vez que ya hayamos llegado a la carpeta donde se guardo el código, lo ejecutamos con la sentencia: “python3 -nombre del código.py” y ejecutamos con enter.
19. Con esto anterior, ya se estará ejecutando el código y el programa todo con normalidad.