



INFORME DE LABORATORIO 4

Autores: Angie Paola Jaramillo Ortega, Juan Manuel Rivera Florez

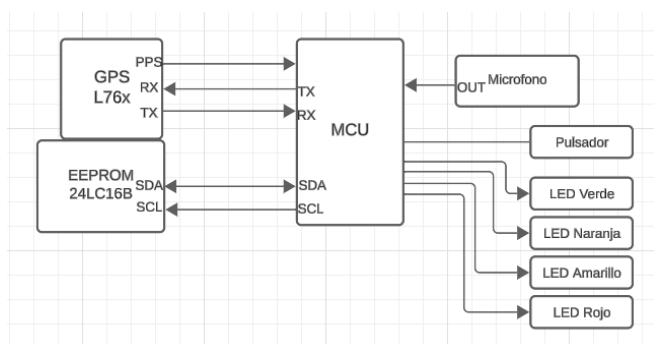
Laboratorio de Electrónica Digital 3
Departamento de Ingeniería Electrónica y de Telecomunicaciones
Universidad de Antioquia

0.1. Introducción

En este laboratorio se desarrolló un sistema de medición de ruido ambiental con geolocalización, empleando una Raspberry Pi Pico. El sistema utiliza un receptor GPS para obtener coordenadas geográficas mediante UART, un micrófono conectado al ADC interno para estimar el nivel de ruido en dB, y una memoria EEPROM externa para almacenar los datos vía I2C.

El funcionamiento se organiza mediante una máquina de estados finita (FSM) que controla la captura, almacenamiento y visualización de los datos. El objetivo principal fue integrar múltiples periféricos y protocolos de comunicación para construir un sistema embebido funcional y robusto.

0.2. Implementación



0.2.1. Configuración física del sistema

El sistema fue desarrollado sobre una Raspberry Pi Pico, seleccionada por su eficiencia energética, capacidad de procesamiento y versatilidad. Se integraron los siguientes componentes:

- Micrófono eléctrico con amplificador MAX4466: encargado de captar el sonido ambiental. La salida analógica fue conectada al GPIO 26 ADC del microcontrolador, permitiendo convertir las señales de audio en valores digitales.
- Módulo GPS L76X: utilizado para obtener las coordenadas geográficas del lugar donde se realiza la medición. El módulo fue conectado mediante UART y configurado para proporcionar latitud y longitud.
- Memoria EEPROM 24LC16: permitió almacenar de forma segura las mediciones, incluso ante cortes de energía o reinicios del sistema. La comunicación se realizó a través del bus I2C.
- Pulsador: utilizado por el usuario para iniciar una medición.
- LEDs: se emplearon cuatro LEDs (verde, amarillo, naranja y rojo) para informar el estado del sistema.

- Alimentación: el sistema se alimentó con una batería de 9V conectada a un regulador de voltaje a 3.3V.

0.2.2. Arquitectura general y lógica de funcionamiento

El sistema fue diseñado de forma modular basado en una máquina de estados, en la que cada módulo se encarga de una tarea específica (GPS, EEPROM, ADC).

Al encender el dispositivo:

1. Se espera la señal de PPS del GPS para sincronizar el sistema.
2. Se entra en modo de espera (estado IDLE) con el LED verde encendido.
3. Al presionar el botón, se inicia la captura de audio y GPS(estado capturing).
4. Si todo es exitoso, se almacena en EEPROM y se muestra LED naranja(estado storing).
5. Si ocurre un error, se enciende LED rojo por 3s (estado error).
6. Si en lugar del botón el usuario envía el comando DUMP por el serial se mostraran las mediciones (estado dump).

Se define una enumeración de estados y un puntero a función que ejecuta el estado actual. Cada estado se implementa como una función independiente en FSM.c. Los estados principales son:

INIT_STATE

Inicializa periféricos, limpia buffers, configura interrupciones y espera la primera señal de sincronización (basado en la señal de pulsos PPS) del GPS para asegurar que está enganchado. Una vez enganchado el GPS se pasa al estado IDLE

STATE_IDLE

Espera acción del usuario o comando DUMP. Está en espera activa (`_wfi()`) para consumir poca energía.

- Si se presiona el botón (interrupción GPIO), se pasa a STATE_CAPTURING.
- Si se recibe el comando "DUMP", se pasa a STATE_DUMP.
- el LED verde se encuentra encendido para indicar estado inactivo.

STATE_CAPTURING

Realiza tres tareas:

1. Adquisición de coordenadas GPS:

- Se recibe la trama \$GNRMC por UART.
- Se verifica que el GPS esté activo (`status == 'A'`) y se extraen latitud y longitud.
- Si se pierde el GPS o se presiona el botón, se cancela la captura y transiciona a STATE_ERROR.

Durante la práctica, se utilizó un módulo GPS que se comunica con la Raspberry Pi Pico mediante la interfaz UART, enviando continuamente mensajes en formato texto, conocidos como sentencias NMEA. La obtención de las coordenadas se realiza a través del procesamiento de estas sentencias.

El módulo GPS transmite múltiples tipos de sentencias NMEA por segundo, pero en esta práctica se usó únicamente la sentencia \$GNRMC (Minimum Specific GNSS Data), ya que contiene los datos relevantes para la práctica en un solo mensaje:

\$GNRMC, Timestamp, Status, Lat, N/S,
Long, E/W, SOG, COG, Date,
MagVar, MagVarDir, mode,
NavStatus*checksum, CR, LF,
Timestamp, Status, Lat,
N/S, Long, E/W, SOG, COG, Date,
MagVar, MagVarDir, mode,
NavStatus*checksum CR LF

Donde los datos usados fueron:

- \$GNRMC: Identificador de la sentencia.
- Status: Estado del fix (A = válido, V = inválido).
- lat + N/S: Latitud en grados y minutos + hemisferio.
- lon + E/W: Longitud en grados y minutos + hemisferio.

Esta estructura fue consultada en [1] sobre la estructura de sentencias NMEA.

2. Captura de audio:

Para capturar el nivel de ruido ambiental, se utilizó un micrófono analógico conectado a una de las entradas ADC de la Raspberry. Este sensor emite una señal de voltaje proporcional a la intensidad sonora, con un acoplamiento DC centrado en $V_{cc}/2$, lo cual implica que en ausencia de sonido la señal se mantiene en un nivel constante cercano a 1.65V (para $V_{cc} = 3.3V$).

La lectura del nivel de sonido se realiza a través de una rutina periódica basada en interrupciones. Un temporizador se configura para dispararse cada 1ms, y en cada interrupción se realiza una lectura del ADC. Esta muestra se almacena en un búfer de 10000 posiciones, de manera que el tiempo total de adquisición es de 10s.

3. Cálculo del nivel de ruido: El ADC tiene una resolución de 12 bits, por lo que las

lecturas x'_i están en el rango de 0 a 4095. Para centrar la señal, se resta 2048:

$$x'_i = x_i - 2048 \quad (0-1)$$

Luego, se calcula el valor cuadrático medio (RMS) a partir de las muestras centradas. Primero, se obtiene la media cuadrática de los x'_i y luego se convierte el resultado a voltios:

$$V_{rms} = \frac{3,3}{4095} \cdot \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - 2048)^2} \quad (0-2)$$

donde N es el número total de muestras tomadas, x'_i es el valor de la muestra del ADC y 3.3V es el voltaje de referencia del ADC (VCC).

Por último El valor V_{rms} se compara con un voltaje de referencia V_{ref} , obtenido durante una etapa de calibración en un ambiente de ruido, y luego se calcula el nivel de presión sonora:

$$dB = 20 \cdot \log_{10} \left(\frac{V_{rms}}{V_{ref}} \right) \quad (0-3)$$

V_{ref} fue calibrado anteriormente en un ambiente ruidoso para obtener valores en dB que coincidan con los valores de nivel de ruido reales.

STATE STORING

En este estado se copian los valores de latitud y longitud a un buffer de 16 bytes. Cada coordenada se encuentra representada como un número double, lo que ocupa 8 bytes. Los primeros 8 bytes del buffer corresponden a la latitud y los siguientes 8 bytes corresponden a la longitud. Luego es este buffer que contiene ambos datos el que se escribe en la memoria.

La EEPROM utilizada en esta práctica, modelo 24LC16B, es una memoria no volátil de tipo I2C que ofrece un total de 16 kb de capacidad,

equivalentes a 2048 bytes. Esta memoria tiene un arreglo interno que condiciona la forma en la que se deben organizar y escribir los datos.

La memoria está dividida en 8 bloques de 256 bytes cada uno, accesibles mediante diferentes direcciones base del bus I2C:

- Bloque 0 → Dirección: 0x50
- Bloque 1 → Dirección: 0x51
- ...
- Bloque 7 → Dirección: 0x57

Cada bloque funciona como una memoria independiente, con su propio espacio de direcciones de 0 a 255. Es decir, al acceder a la memoria no se puede escribir en más de 256 bytes por bloque, y para cambiar de bloque se debe usar una dirección I2C diferente.

Otra característica relevante es el tamaño de página de escritura, que en este caso es de 16 bytes. Esto significa que se pueden escribir hasta 16 bytes en una sola operación, siempre que no se crucen los límites de página puesto que si se intenta escribir más allá de un límite de página, los datos pueden sobrescribirse o corromperse. Por ejemplo, si se escriben 17 bytes desde la posición 250, los primeros 6 bytes se escribirán y los 11 restantes se sobrescribirán desde la posición 0 de la misma página.

Para esta práctica se deben almacenar tres datos por medición (latitud, longitud, nivel de ruido) que en total ocupan 17 bytes. Debido a que el tamaño de página es de 16 bytes, almacenar los tres valores juntos en una sola operación no sería eficiente, ya que constantemente se violaría el límite de página.

Por estas razones, se optó por un diseño más modular y seguro, que aprovecha la arquitectura en bloques: El bloque 0 (0x50) se usa exclusivamente para almacenar las coordenadas GPS (latitud y longitud), ocupando exactamente 16

bytes por medición. Esto se ajusta perfectamente al tamaño de página de la EEPROM y el bloque 1 (0x51) se utiliza para almacenar el nivel de ruido de cada medición, que ocupa solo 1 byte.

STATE_ERROR

Se activa cuando ocurre una condición de error durante el proceso de adquisición de datos. Las causas que disparan este estado son:

- Pérdida de señal del GPS.
- Cancelación manual del proceso por medio del botón durante la captura.
- Fallos al escribir o leer en la memoria EEPROM.

Cuando se entra a este estado, se ejecutan rutinas específicas para notificar mediante LEDs el tipo de error detectado. Para ello, se utiliza el LED rojo conectado a un pin GPIO. Esto se logra mediante una variable auxiliar motivo_error y capture_cancelled que indican el origen del error.

Una vez finalizado el tiempo, la FSM regresa al estado init_state, donde se revalida la presencia de señal GPS antes de continuar con nuevas tomas de muestras.

ESTATE_DUMP

El estado state_dump permite consultar los datos almacenados en la EEPROM, es decir, las mediciones previamente capturadas de coordenadas GPS y niveles de ruido ambiental. Esta funcionalidad se activa cuando el sistema recibe el comando DUMP a través de la conexión USB.

En el estado state_dump, el sistema recorre secuencialmente ambos bloques y reconstruye los datos originales combinando una entrada del bloque de coordenadas con su correspondiente entrada del bloque de niveles de ruido. Después de la lectura y el ensamble se imprimen los datos en consola con el siguiente formato:

Coordenadas: 6.322202, -75.560342, Nivel de ruido: 55 dB

Tras completar el volcado de datos, la FSM retorna automáticamente al estado INIT_STATE, verificando la señal gps que permita nuevas mediciones.

0.3. Resultados

Durante la ejecución de la práctica, se realizaron mediciones en cinco ubicaciones correspondientes a las distintas porterías del campus. Para cada punto se capturaron las coordenadas geográficas (latitud y longitud) usando el módulo GPS, junto con el nivel de ruido ambiental en decibelios obtenido a partir del micrófono.

La siguiente tabla resume los resultados obtenidos:

Portería	Latitud	Longitud	dB
Avenida del Ferrocarril	6.267443	-75.567058	73
Peatonal	6.265182	-75.567838	69
Barranq...	6.265233	-75.569208	71
Avenida del Río	6.268370	-75.570612	76
Metro	6.269038	-75.566675	73

Para validar la correcta localización de las mediciones, se usó un script en Python que traza las coordenadas sobre un mapa interactivo. El siguiente código se ejecutó en Google Colab:

```
import folium
from IPython.display import display
from folium import Map, Marker
from folium.plugins import MarkerCluster

m = Map(location=[6.267, -75.568], zoom_start=17)

# Lista de ubicaciones
locations = [
    (6.267443, -75.567058, "Portería 1 - 73 dB"),
    (6.265182, -75.567838, "Portería 2 - 69 dB"),
    (6.265233, -75.569208, "Portería 3 - 71 dB"),
    (6.268370, -75.570612, "Portería 4 - 76 dB"),
    (6.269038, -75.566675, "Portería 5 - 73 dB")
]

marker_cluster = MarkerCluster().add_to(m)

for lat, lon, label in locations:
    Marker([lat, lon], popup=label).add_to(marker_cluster)

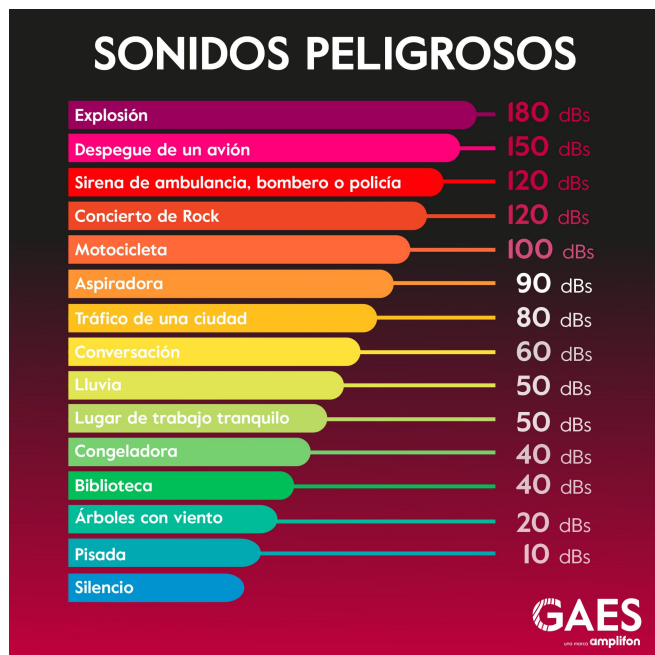
display(m)
```

La figura 0-1 muestra la visualización obtenida al ejecutar el código anterior:



Figura 0-1: Visualización de las ubicaciones de medición.

Estos resultados se compararon con una referencia externa que establece una escala de niveles sonoros típicos. De acuerdo con esta fuente, niveles de 70–80 dB son característicos del tráfico urbano o de la operación de electrodomésticos como aspiradoras.



En este contexto, los niveles obtenidos se consideran coherentes. Todas las mediciones se encuentran en un rango de 69 a 76 dB, lo que indica un ambiente urbano con ruido moderado, posiblemente influenciado por los vehículos y el flujo de personas que transitan por estas zonas. No se evidencian valores atípicos ni fuera del rango esperado para zonas cercanas a calles.

Además, las diferencias entre mediciones son pequeñas, lo que respalda la consistencia del sistema de medición y el proceso de calibración aplicado al micrófono. Por lo tanto, se concluye que los resultados obtenidos son técnicamente razonables y representativos del entorno en el que fueron tomados.

0.4. Conclusiones

Durante el desarrollo de esta práctica se lograron aplicar de manera efectiva los protocolos de comunicación y conversión analógica-digital para adquirir y almacenar información de ambiente.

- Se comprendió y aplicó el protocolo I2C

para la comunicación con una memoria EEPROM externa, lo cual permitió almacenar de forma persistente los datos recolectados. La correcta administración de bloques de memoria y el conocimiento del tamaño de página fueron esenciales para evitar errores de escritura y pérdida de información.

- Se implementó el protocolo UART para la recepción de datos desde un módulo GPS externo. Se trabajó con sentencias del estándar NMEA, en particular la sentencia \$GNRMC, lo que requirió comprender tanto el formato de datos como las condiciones necesarias para determinar si había una "fix" válida del GPS. Además, se utilizó la señal PPS como indicador de conexión integrándola mediante interrupciones.
- El uso del ADC de la Raspberry Pi Pico permitió capturar el nivel de señal del micrófono conectado al sistema. Para obtener una medida significativa del nivel de presión sonora, se diseñó un sistema de muestreo periódico y cálculo RMS, demostrando que es posible traducir señales analógicas a información útil mediante el procesamiento digital.

Bibliografía

- [1] TavoTech. Gps nmea sentence structure, 2021. URL <https://tavotech.com/gps-nmea-sentence-structure/>. Último acceso: 18 de julio de 2025.