

Práctica 4 – Mapeando el nivel de Ruido

1st Oscar Andres Gutierrez Rivadeneira
11993515015
oscar.gutierrezr@udea.edu.co

2nd Imar Nayeli Jimenez Arango
1007424872
imar.jimenez@udea.edu.co

I. RESUMEN

En esta práctica, se desarrolló un sistema integrado que combina la funcionalidad de un módulo GPS L76X y un sensor de sonido MAX4466 utilizando una placa Raspberry Pi Pico. El objetivo principal fue adquirir datos de posicionamiento dentro de la universidad y niveles de ruido en decibelios durante 10 segundos, almacenarlos en la memoria flash del microcontrolador y proporcionar una interfaz para la lectura de estos datos.

II. INTRODUCCIÓN

La práctica realizada en el laboratorio de Electrónica Digital 3 tiene una gran relevancia en la vida cotidiana debido a la amplia aplicación de los sistemas de posicionamiento global (GPS) y la medición de niveles de ruido en diversos campos. Desde la navegación y el seguimiento de vehículos hasta el monitoreo ambiental y la gestión de ciudades inteligentes, estas tecnologías juegan un papel crucial. La habilidad para implementar y diseñar sistemas que integren estos componentes es fundamental para el desarrollo de soluciones tecnológicas avanzadas.

III. IMPLEMENTACIÓN Y DISEÑO

mhj

A. Componentes de Hardware

- **Raspberry Pi Pico:** Microcontrolador utilizado como unidad principal de adquisición, control y almacenamiento de datos.
- **Módulo GPS L76X:** Permite obtener coordenadas geográficas precisas (latitud, longitud, altitud) y sincronización horaria a través del protocolo NMEA.
- **Sensor de sonido MAX4466:** Micrófono amplificado conectado a un pin ADC para capturar los niveles de ruido en el entorno.
- **Memoria EEPROM 24LC16B:** Memoria externa no volátil usada para almacenar los niveles de ruido y ubicación donde se tomó la muestra.
- **Botón (SW):** Permite al usuario iniciar la adquisición de datos de forma manual, controlando el flujo del estado del sistema.
- **LEDs RGB (Rojo, Naranja, Verde):** Señalizan los estados del sistema: éxito en adquisición (verde), error (rojo) y captura en curso (naranja).
- **Protoboard y cables:** Utilizados para realizar las conexiones entre los distintos módulos del sistema.

B. Asignación de Pines

- **GPS (UART0):**
 - TX del GPS → GPIO8 (RX UART0)
 - RX del GPS → GPIO9 (TX UART0)
 - PPS del GPS → GPIO7
- **Sensor de sonido (Micrófono):**
 - Salida analógica → GPIO26 (ADC0)
- **EEPROM 24LC16B (I2C0):**
 - SDA → GPIO4
 - SCL → GPIO5
- **Botón (SW):**
 - Entrada digital → GPIO6
- **LEDs de estado:**
 - LED Rojo → GPIO19
 - LED Naranja → GPIO20
 - LED Verde → GPIO21

IV. DESCRIPCIÓN DEL FUNCIONAMIENTO DEL SISTEMA

A. Proceso de Datos GPS

El sistema utiliza un módulo GPS L76X, el cual transmite datos mediante el protocolo NMEA a través de una interfaz UART. La Raspberry Pi Pico recibe estas tramas por el UART, y procesa únicamente la trama \$GNRMC, que contiene información sobre la hora, latitud, longitud, estado del fix, y velocidad.

La función `parse_gps_data()` identifica la trama \$GNRMC dentro del buffer recibido, y verifica la validez de los datos mediante el cálculo del checksum. Solo si el estado del receptor es 'A' (activo), se extraen los campos de interés, como la latitud y longitud en formato decimal. Estos datos se almacenan en una estructura tipo `GPSTData`, la cual se utiliza luego para el almacenamiento en la EEPROM.

El procesamiento de datos GPS se realiza una única vez por ciclo de adquisición, justo antes de iniciar la toma de muestras de audio, asegurando que cada bloque de datos tenga una posición geográfica actualizada asociada.

B. Medición de Nivel de Ruido

El sistema utiliza un micrófono conectado al pin 26 del ADC de la Raspberry Pi Pico para realizar la adquisición de señales acústicas del entorno. Se configuró una rutina que toma un total de 20.000 muestras del canal ADC, con un retardo de `sleep_us(500)` entre lecturas, lo que equivale a

una frecuencia de muestreo de aproximadamente 2 kHz. Esto permite capturar 10 segundos continuos de señal de audio.

Las muestras se almacenan en un arreglo en memoria RAM para su posterior procesamiento. Primero se calcula el valor promedio (DC offset) de las muestras, y luego se obtiene la raíz cuadrática media (RMS) sobre las componentes AC. Finalmente, se calcula el nivel de señal en dBFS (decibelios relativos a la escala completa del ADC) mediante la expresión:

$$\text{dBFS} = 20 \cdot \log_{10} \left(\frac{\text{RMS}}{\text{RMS}_{\text{máx}}} \right)$$

donde el valor $\text{RMS}_{\text{máx}}$ corresponde al máximo nivel que puede alcanzar el ADC sin saturación (en este caso, aproximadamente 1.165 V para una señal de 3.3 Vpp sinusoidal). Este valor en dBFS representa una medida relativa de la intensidad del ruido ambiental, útil para comparar distintos entornos sin requerir calibración en dB SPL.

C. Almacenamiento en Memoria Flash

Para garantizar la persistencia de los datos recolectados, el sistema utiliza una memoria flash externa EEPROM conectada mediante el bus I2C. Esta memoria almacena los resultados de cada medición, que incluyen tanto el nivel de ruido en dBFS como las coordenadas GPS asociadas a la ubicación donde se realizó la captura.

Cada registro de medición está compuesto por:

- Nivel de ruido ambiental (`int8_t`): 1 byte
- Latitud (`float`): 4 bytes
- Longitud (`float`): 4 bytes
- Conteo de muestras GPS válidas (`uint8_t`): 1 byte

En total, cada registro ocupa 10 bytes. Esto permite aprovechar eficientemente el espacio disponible en la EEPROM, que en este caso tiene una capacidad de 32 KB. Por lo tanto, se pueden almacenar hasta 3200 registros distintos antes de sobrescribir o requerir descarga.

La escritura se realiza en bloques secuenciales utilizando una variable de índice que actúa como puntero de escritura. Este índice se incrementa automáticamente después de cada medición, manteniendo una estructura de tipo cola FIFO. Para prevenir errores de escritura o corrupción, se incluye una verificación previa del estado de la EEPROM.

Los datos almacenados pueden ser recuperados a través del puerto UART para su análisis posterior en una estación de trabajo externa.

D. Lógica del Programa y Máquina de Estados

El sistema se encuentra estructurado mediante una máquina de estados finitos (FSM) que define el comportamiento global del dispositivo portátil de medición de ruido ambiental. Esta organización facilita una secuencia clara de operaciones, garantiza la sincronización entre tareas críticas y permite mantener una implementación eficiente en términos de recursos de CPU.

Los estados principales del sistema son:

- **ESPERA:** Estado inicial y de bajo consumo. El sistema permanece inactivo hasta que el usuario presiona el botón de inicio conectado al GPIO6.
- **ADQUISICIÓN GPS:** Se activa el módulo GPS y se capturan datos durante un periodo fijo de tiempo (e.g., 10 segundos). Se recopilan y filtran las sentencias GPGLL para obtener coordenadas válidas, mientras se ignoran mensajes incompletos o corruptos.
- **ADQUISICIÓN DE AUDIO:** Simultáneamente o inmediatamente después del GPS, se inicia la captura de señales acústicas a través del ADC. Se recolectan 20000 muestras a una frecuencia de 2 kHz, lo cual cubre un intervalo de 10 segundos. Posteriormente, se calcula el nivel de presión sonora usando el método de dBFS.
- **ALMACENAMIENTO:** Se escribe un registro compacto en memoria EEPROM, el cual incluye la latitud, longitud, conteo de sentencias GPS válidas y el nivel de ruido en dBFS.
- **INDICADOR:** Se hace parpadear un LED indicando que el registro ha sido exitosamente almacenado. Luego, el sistema retorna al estado de ESPERA.

Este diseño garantiza que cada adquisición sea autónoma, replicable y sincronizada, permitiendo que el usuario recopile múltiples registros secuenciales en diferentes ubicaciones sin intervención adicional ni necesidad de monitoreo continuo.

V. RESULTADOS

Se realizaron pruebas del sistema en tres porterías de la Universidad de Antioquia: Ferrocarril, Metro y Barranquilla.

En cada portería se activó la captura de datos, los cuales fueron posteriormente extraídos utilizando el comando "dump" desde la memoria flash del dispositivo.

Los datos obtenidos corresponden a las últimas mediciones realizadas en cada punto. Para validar el correcto funcionamiento del módulo GPS, se ingresaron las coordenadas registradas en Google Maps, observándose coincidencia con las ubicaciones reales de las porterías, como se muestra en las siguientes figuras.

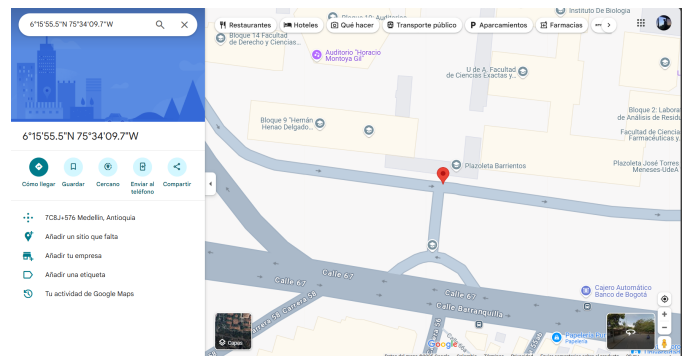


Fig. 1. Ubicación registrada en la portería Barranquilla.

VI. CONCLUSIONES

- La implementación del protocolo UART para la comunicación con el módulo GPS resultó eficiente y confiable.

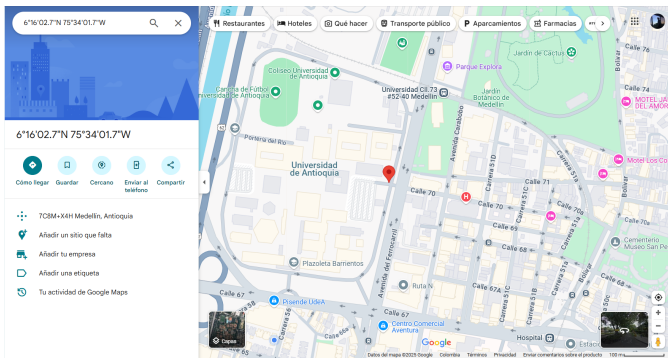


Fig. 2. Ubicación registrada en la portería Ferrocarril.

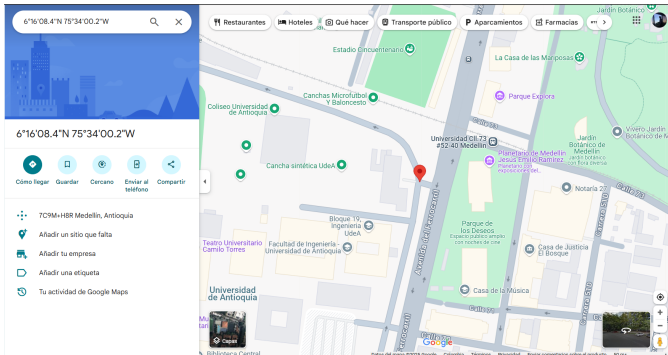


Fig. 3. Ubicación registrada en la portería Metro.

Este protocolo permitió una transferencia de datos rápida y estable, lo cual es fundamental en aplicaciones de tiempo real como el seguimiento de posición. Se comprobó que una correcta configuración del baud rate y demás parámetros es esencial para garantizar la integridad de los datos recibidos.

- La medición del nivel de ruido utilizando el ADC interno de la Raspberry Pi Pico evidenció la importancia de contar con convertidores de alta resolución en aplicaciones de adquisición de datos. En este proyecto, el ADC de 12 bits ofreció una precisión suficiente para digitalizar las señales analógicas provenientes del micrófono de manera efectiva.
- La estructura del programa, basada en un bucle principal combinado con el uso de interrupciones y temporizadores, permitió un flujo de ejecución ordenado y eficiente. El uso de interrupciones para eventos como la pulsación del botón y la temporización periódica habilitó una respuesta rápida a estímulos externos sin necesidad de sondeo continuo, optimizando el rendimiento general del sistema.