Medición de la Intensidad de Señal WiFi con Raspberry Pi Pico W

Universidad Militar Nueva Granada
Comunicaciones Digitales
jose.rugeles@unimilitar.edu.co

Marzo 2025

1 Introducción

La intensidad de la señal recibida (RSSI, por sus siglas en inglés: *Received Signal Strength Indicator*) es un parámetro clave en las comunicaciones inalámbricas, ya que indica la potencia de la señal WiFi recibida por un dispositivo. Este laboratorio tiene como objetivo diseñar e implementar un programa en MicroPython para medir el RSSI en función de la distancia utilizando un Raspberry Pi Pico W.Los estudiantes explorarán cómo varía la señal con la distancia en un entorno real, recolectarán datos y generarán curvas de RSSI vs. distancia con barras de error basadas en la desviación estándar. Este ejercicio combina programación, manejo de hardware y análisis estadístico, ofreciendo una experiencia práctica en el estudio de señales inalámbricas.

2 Requisitos

2.1 Hardware

Para realizar este laboratorio, necesitarán los siguientes componentes:

- **Raspberry Pi Pico W**: Microcontrolador con capacidad WiFi, con firmware MicroPython instalado.
- **Pantalla OLED SSD1306**: 128x32 píxeles, conectada vía I2C (GPIO 14: SDA, GPIO 15: SCL).
- **Pulsador**: Conectado entre GPIO 18 y GND, con pull-up interno para pulso negativo.
- **LEDs Indicadores**:
  - LED "Avanzar" (GPIO 16): Indica cuándo moderse para tomar la nueva medida.

  - LED "Midiendo" (GPIO 17): Indica que se está midiendo la señal RSSI.
  - LED "WiFi" (GPIO 19): Parpadea si la conexión WiFi se pierde.
  - Resistencias de 330Ω en serie con cada LED a GND.
- **Fuente de Alimentación**: Batería para el Pico W.
- **Red WiFi**: Punto de acceso estable (ej. hotspot de un teléfono) con SSID y contraseña conocidos.
- **Cinta métrica**: Para marcar distancias de 1 metro.

2.2 Software

- **Entorno**: Thonny IDE para cargar el programa y acceder al sistema de archivos del Pico W.
- **Librerías**: MicroPython estándar (incluye `network`, `time`, `machine`, `math`, `os`) y `ssd1306.py` (descargar y cargar al Pico W).

3 Procedimiento

3.1 Diseño del Programa

Los estudiantes deben diseñar un programa en MicroPython con las siguientes funcionalidades:

1. **Configuración**: Definir pines GPIO y inicializar la pantalla OLED y el WiFi.
2. **Conexión WiFi**: Implementar una función para conectar y reconectar a la red WiFi, mostrando el estado en la pantalla.
3. **Medición**: Tomar 200 muestras de RSSI durante 20 segundos (0.1s por muestra) en cada punto, calculando promedio y desviación estándar.
4. **Interacción**: Usar un pulsador para dar inicio a la toma de mediciones. Debe usar Leds para guiar al usuario.
5. **Almacenamiento**: Guardar distancia, RSSI promedio y desviación estándar en `rssi_measurements.txt`.
6. **Visualización**: Mostrar RSSI en tiempo real y barras de señal en la pantalla OLED.

3.2 Prueba de Campo

Se deben realizar dos pruebas en lugares diferentes, de manera que se puedan comparar luego los resultados. Se aconseja ejecutar el programa y estimar incialmente el alcance máximo, separandose del punto de acceso y luego iniciar las mediciones cada metro. El punto de acceso no debe estar en el piso. Es necesario usar algún tipo de tripode o soporte para dejarlo fijo.

Se debe realizar la prueba en un área abierta siguiendo estos pasos:

1. Colocar el punto de acceso WiFi en el punto 0m.
2. Ubicarse a 0m y ejecutar el programa.
3. Presionar el pulsador para medir el RSSI durante 20 segundos.
4. Moverse a 1 metro y repetir hasta que se pierda la conexión.
5. Monitorear el LED "WiFi"; si parpadea, esperar la reconexión antes de continuar.
6. Descargar `rssi_measurements.txt` al finalizar.

4 Análisis de resultados

Partiendo de las mediciones obtenidas :

- Se debe realizar el análisis de los resultados a partir de importar los datos del archivo `rssi_measurements.txt` (formato: distancia, RSSI, desviación estándar) en Matlab o Python.
- Se deben generar las gráficas de RSSI vs. distancia con barras de error usando la desviación estándar.
- Se debe comparar en la misma grafica el modelo teorico de propagación en espacio libre. (Ver anexo 1).

5 Entregables

- Código fuente en MicroPython con comentarios. El código debe publicarse en github.
- Archivos `rssi_measurements.txt` con los datos recolectados.
- Gráficas de RSSI vs. distancia con barras de error.
- Informe breve orientado al análisis de resultados.

Anexo: Comparación con el Modelo de Propagación en Espacio Libre

El modelo de propagación en espacio libre (FSPL) predice la pérdida de potencia de una señal en un entorno ideal sin obstáculos. Las mediciones de RSSI se pueden comparar con el modelo teórico, ajustándolo con una constante K de manera que se puedan analizar las diferencias entre las condiciones ideales y las reales, utilizando la frecuencia específica del canal WiFi seleccionado.

Fundamento Teórico

La pérdida de propagación en espacio libre, con la distancia d en kilómetros y la frecuencia f en MHz, se calcula como:

$$FSPL(dB) = 20\log_{10}(d) + 20\log_{10}(f) + 32.44 \tag{1}$$

donde:

- $FSPL$: Pérdida en decibelios (dB).
- d: Distancia en kilómetros (km).
- f: Frecuencia en megahertzios (MHz), específica del canal WiFi.
- 32.44: Constante derivada de $20\log_{10}\left(\frac{4\pi}{c}\right)$.

El RSSI teórico, ajustado con una constante K (en dB), se define como:

$$RSSI_{teórico}(dBm) = P_{TX} - FSPL + K \tag{2}$$

donde:

- P_{TX}: Potencia transmitida por el punto de acceso (en dBm), típicamente entre 15 y 20 dBm.
- K: Constante de ajuste (en dB) para compensar diferencias entre el modelo y los datos experimentales.

Frecuencias por Canal WiFi (2.4 GHz)

Seleccionen la frecuencia correcta según el canal WiFi usado, conforme a la Tabla 1.

Instrucciones para la Comparación

Para comparar los datos experimentales de RSSI con el modelo FSPL ajustado:

1. **Identificar el canal WiFi**: Determinen el canal usado por su punto de acceso (ej. en la configuración del hotspot).
2. **Seleccionar la frecuencia**: Usar la Tabla 1 para obtener f en MHz (ej. canal 6: $f = 2437\,\text{MHz}$).

Canal	Frecuencia (MHz)
1	2412
2	2417
3	2422
4	2427
5	2432
6	2437
7	2442
8	2447
9	2452
10	2457
11	2462
12	2467
13	2472
14	2484

Table 1: Frecuencias de los canales WiFi en la banda de 2.4 GHz.

3. **Calcular el FSPL teórico**:
   - Convertir la distancias: $d_{\text{km}} = d_{\text{m}}/1000$.
   - Calcular: $FSPL(dB) = 20\log_{10}(d_{\text{km}}) + 20\log_{10}(f) + 32.44$.
   - Ejemplo: Para canal 6 ($f = 2437\,\text{MHz}$), $FSPL \approx 20\log_{10}(d_{\text{km}}) + 100.17$.
4. **Estimar P_{TX} y K**:
   - Suponer$P_{TX} = 20\,\text{dBm}$ (valor típico).
   - Calcular el RSSI teórico inicial sin K ($K = 0$).
   - Comparar con el RSSI experimental en $d = 1\,\text{m}$ (0.001 km) y ajustar K para minimizar la diferencia: $K = RSSI_{\text{exp}}(1\,\text{m}) - (P_{TX} - FSPL(1\,\text{m}))$.
5. **Calcular RSSI teórico ajustado**: $RSSI_{teórico} = P_{TX} - FSPL + K$ para cada distancia.
6. **Graficar**: Añadan la curva teórica ajustada a su gráfica experimental, indicando el canal y K en la leyenda o título.
7. **Analizar**: Respondan en su informe:
   - ¿Cuál fue el valor de K y qué podría explicarlo (pérdidas adicionales, entorno)?
   - ¿Cómo se compara la curva ajustada con la experimental?
   - ¿Qué factores afectan las diferencias restantes?

Anexo: Programa de Ejemplo para Medir RSSI

Este anexo presenta un programa en MicroPython diseñado para medir la intensidad de la señal WiFi (RSSI) utilizando un Raspberry Pi Pico W. El código configura una conexión WiFi, toma 10 muestras de RSSI cada 0.2 segundos, calcula el promedio y lo muestra tanto en la consola como en una pantalla OLED SSD1306.

```
# Programa para medir la intensidad de se al WiFi (RSSI)
# jose.rugeles@unimilitar.edu.co

import network
import time
from machine import Pin, I2C
from ssd1306 import SSD1306_I2C

WIDTH = 128
HEIGHT = 32
i2c = I2C(1, scl=Pin(15), sda=Pin(14), freq=200000)
oled = SSD1306_I2C(WIDTH, HEIGHT, i2c)

wifi = network.WLAN(network.STA_IF)
wifi.active(True)
ssid = "********"
password = "********"

wifi.connect(ssid, password)
while not wifi.isconnected():
    time.sleep(1)

print("Conexi n establecida!")
print("Direcci n IP:", wifi.ifconfig()[0])

while True:
    rssi_values = []

    for _ in range(10):
        rssi = wifi.status('rssi')
        rssi_values.append(rssi)
        time.sleep(0.2)

        rssi_average = sum(rssi_values) / len(rssi_values)
        print("Promedio de RSSI:", rssi_average, "dBm")

    oled.fill(0)
    oled.invert(True)
    oled.text("RSSI:", 2, 6)
    oled.text(str(round(rssi_average, 2)) + " dBm", 10, 20)
    oled.show()

    time.sleep(2)
```

Notas sobre el Código

- Las credenciales WiFi (`ssid` y `password`) han sido reemplazadas por asteriscos. Los estudiantes deben sustituirlas por las de su propia red.
- El programa toma muestras cada 0.2 segundos durante 2 segundos, calcula el promedio y lo actualiza cada 4 segundos (2s de medición + 2s de espera).
- Requiere la biblioteca `ssd1306.py` cargada en el Raspberry Pi Pico W para la pantalla OLED.

6 Referencias

`https://docs.micropython.org/en/latest/library/network.WLAN.html`