คู่มือการอ่านและวิเคราะห์ค่า RSSI เพื่อหาจุดบอด (Blind Spot Analysis).md

Add new file
jeerapol kumkeam authored just now

คู่มือการอ่านและวิเคราะห์ค่า RSSI เพื่อหาจุดบอด (Blind Spot Analysis).md 18.15 KiB

📡 คู่มือการอ่านและวิเคราะห์ค่า RSSI เพื่อหาจุดบอด (Blind Spot Analysis)

จากเอกสาร **RFID Reader Data Communication Protocol** Reader รองรับการรายงานค่า **RSSI, Phase, Frequency และ Antenna ID** ในการอ่านแต่ละครั้งอย่างครบถ้วน ซึ่งเหมาะมากสำหรับการทำ **RF Site Survey / Blind Spot Analysis** ครับ

---

1. โครงสร้างข้อมูล RSSI ที่ Reader ส่งกลับมา

เมื่อส่งคำสั่ง **Read EPC Tag (MID = 0x10)** Reader จะส่ง **Notification (Flag=1, MID=0x00)** กลับมาทุกครั้งที่อ่าน Tag ได้ โดยมีโครงสร้างดังนี้:

1.1 Mandatory Parameters (ต้องมีทุกครั้ง)

Parameter	PID	Type	Size	Description
Tag EPC Code	(M)	U16	Variable	EPC ของ Tag ที่อ่านได้
Tag PC	(M)	U16	2	PC Value (ระบุความยาว EPC)
Antenna ID	(M)	U8	1	🔑 **เสาอากาศที่อ่านได้** (1, 2, 3, 4, ...)

1.2 Optional Parameters (สำคัญสำหรับการวิเคราะห์)

Parameter	PID	Type	Size	Description
RSSI	`0x01`	U8	1	🔑 **ค่าสัญญาณ (dBm)**
Read Result	`0x02`	U8	1	ผลการอ่าน (0=Success)
TID Data	`0x03`	U16	Variable	TID ของ Tag
User Data	`0x04`	U16	Variable	User Memory
Reserve Data	`0x05`	U16	Variable	Reserve Memory
Sub-Antenna No.	`0x06`	U8	1	เสาอากาศย่อย (กรณีต่อ Hub)
UTC Time	`0x07`	U32×2	8	🔑 **เวลาที่อ่านได้** (Second + Microsecond)
Frequency	`0x08`	U32	4	🔑 **ความถี่ที่ใช้ (KHz)**
Phase	`0x09`	U8	1	🔑 **Phase (0-128)** → คำนวณเป็น `(value/128)*2π`

> 💡 **ข้อมูล 4 อย่างที่ใช้วิเคราะห์จุดบอด:** `RSSI` + `Antenna ID` + `Frequency` + `Phase`

---

2. การตั้งค่า Reader เพื่อให้ได้ข้อมูล RSSI ที่แม่นยำ

2.1 ตั้งค่า Tag Uploading Parameter (MID = 0x09)

Parameter	PID	ค่าที่แนะนำ	เหตุผล
Repeated Tag Filtering Time	`0x01`	`0` (0 ms)	**สำคัญมาก!** ตั้งเป็น 0 เพื่อให้ Reader ส่ง RSSI ทุกครั้งที่อ่านเจอ Tag (ไม่ Filter)
RSSI Threshold	`0x02`	`0` หรือ `-80`	ตั้ง Threshold ต่ำๆ เพื่อให้เก็บข้อมูลสัญญาณอ่อนได้ครบ

ตัวอย่างการส่งคำสั่ง (HEX):

```
MID = 0x09, Category = 2
Data:
  PID 0x01, Length 0x0002, Value 0x0000   (Filter Time = 0)
  PID 0x02, Length 0x0001, Value 0x00     (RSSI Threshold = 0)
```

2.2 ตั้งค่า RF Parameters ให้เหมาะสม

Parameter	ค่าแนะนำ	เหตุผล
Power	26-30 dBm	กำลังส่งสูงสุดเพื่อครอบคลุมพื้นที่
Q Value	4-7	ปรับตามจำนวน Tag ในพื้นที่
Session	S0 หรือ S1	S0 สำหรับ Tag เคลื่อนที่, S1 สำหรับ Tag หยุดนิ่ง
Frequency Hopping	Auto = 1	ให้ Reader กระโดดความถี่อัตโนมัติ (ได้ข้อมูลหลายความถี่)

2.3 ใช้โหมด Continuous Read (MID = 0x10)

```
Mode = 1 (Continuous Read)
Antenna Port = 0x0000000F (เปิดทั้ง 4 เสา)
```

Reader จะอ่านและ Push ข้อมูล RSSI กลับมาอย่างต่อเนื่อง

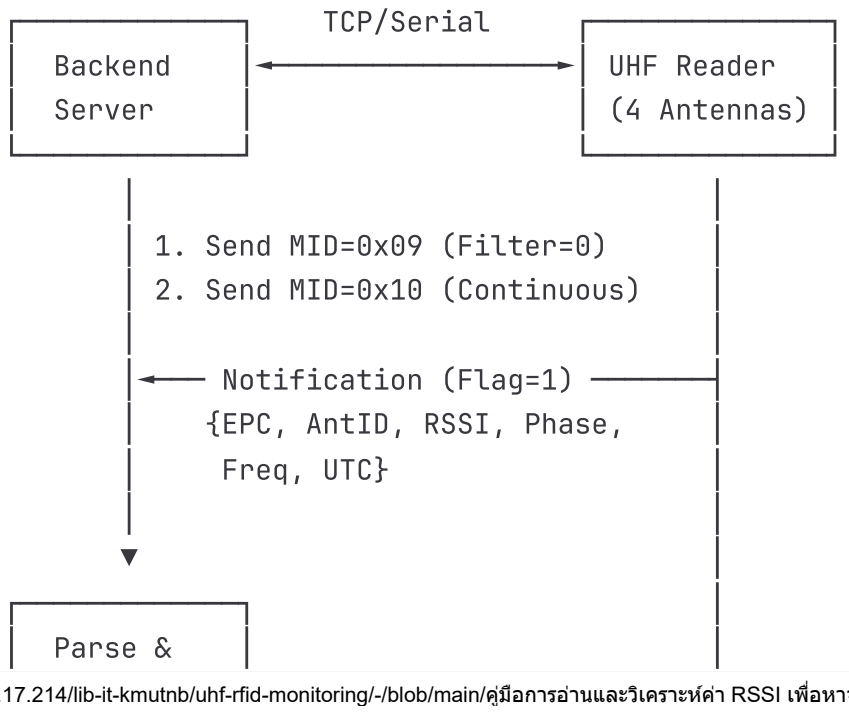
---

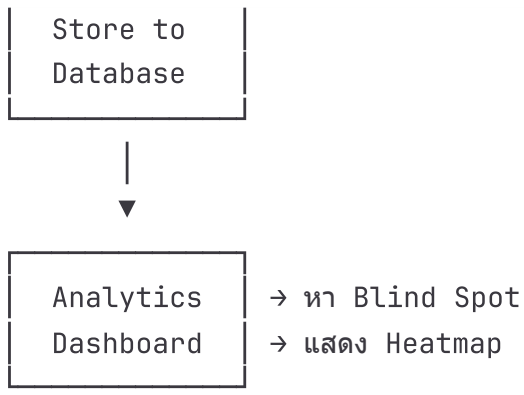
3. การออกแบบระบบเก็บข้อมูล (Data Collection)

3.1 โครงสร้างข้อมูลที่ควรเก็บ (Database Schema)

```sql
CREATE TABLE rssi_samples (
    id              BIGINT PRIMARY KEY AUTO_INCREMENT,
    tag_epc         VARCHAR(64) NOT NULL,      -- EPC ของ Tag
    tag_tid         VARCHAR(64),               -- TID (ถ้ามี)
    antenna_id      TINYINT NOT NULL,          -- เสาอากาศที่ 1-4
    rssi            SMALLINT NOT NULL,         -- ค่าสัญญาณ (dBm)
    phase           DECIMAL(5,2),              -- Phase (เรเดียน)
    frequency_khz   INT,                       -- ความถี่ (KHz)
    utc_seconds     BIGINT,                    -- UTC timestamp
    utc_microsec    INT,
    sub_antenna     TINYINT,                   -- Hub sub-antenna
    read_result     TINYINT,                   -- 0=success
    sample_time     DATETIME DEFAULT NOW(),    -- เวลาที่บันทึก
    INDEX idx_epc (tag_epc),
    INDEX idx_antenna (antenna_id),
    INDEX idx_time (sample_time)
);
```

3.2 Flow การทำงาน





4. ตัวอย่างโค้ด Python (Parser)

```python
import struct
import socket
from datetime import datetime

def parse_tag_notification(data: bytes):
    """Parse EPC Tag Notification (MID=0x00, Flag=1)"""
    offset = 0
    result = {}

    # 1. Parse Mandatory Fields
    # Tag EPC (U16 length + data)
    epc_len = struct.unpack_from('>H', data, offset)[0]
    offset += 2
    result['epc'] = data[offset:offset+epc_len].hex().upper()
    offset += epc_len

    # Tag PC (U16)
    result['pc'] = struct.unpack_from('>H', data, offset)[0]
    offset += 2

    # Antenna ID (U8)
    result['antenna_id'] = struct.unpack_from('B', data, offset)[0]
    offset += 1

    # 2. Parse Optional PID Fields
    while offset < len(data):
        pid = struct.unpack_from('B', data, offset)[0]
        offset += 1

        if pid == 0x01:  # RSSI
            result['rssi'] = struct.unpack_from('B', data, offset)[0]
            offset += 1
        elif pid == 0x02:  # Read Result
            result['read_result'] = struct.unpack_from('B', data, offset)[0]
            offset += 1
        elif pid == 0x03:  # TID
            tid_len = struct.unpack_from('>H', data, offset)[0]
            offset += 2
            result['tid'] = data[offset:offset+tid_len].hex().upper()
            offset += tid_len
        elif pid == 0x07:  # UTC Time (8 bytes)
            sec, usec = struct.unpack_from('>II', data, offset)
            result['utc_sec'] = sec
            result['utc_usec'] = usec
            offset += 8
        elif pid == 0x08:  # Frequency (KHz)
            result['freq_khz'] = struct.unpack_from('>I', data, offset)[0]
            offset += 4
        elif pid == 0x09:  # Phase
            phase_val = struct.unpack_from('B', data, offset)[0]
            result['phase_rad'] = (phase_val / 128.0) * 2 * 3.14159265
            offset += 1
        elif pid == 0x06:  # Sub-Antenna
```

```
            result['sub_antenna'] = struct.unpack_from('B', data, offset)[0]
            offset += 1
        else:
            # Unknown PID - skip (read length first)
            if offset + 2 > len(data): break
            skip_len = struct.unpack_from('>H', data, offset)[0]
            offset += 2 + skip_len

    return result

# ตัวอย่างการใช้งาน
sample_data = bytes.fromhex('000C E0040100123456789ABC 5C40 01 01 28 07 ...')
parsed = parse_tag_notification(sample_data)
print(f"Tag: {parsed['epc']}")
print(f"Antenna: {parsed['antenna_id']}")
print(f"RSSI: -{parsed['rssi']} dBm")
print(f"Phase: {parsed['phase_rad']:.2f} rad")
print(f"Freq: {parsed['freq_khz']} KHz")
```

---

5. การวิเคราะห์หา Blind Spot

5.1 เมตริกที่ใช้วิเคราะห์

เมตริก	สูตร	การตีความ
Average RSSI per Antenna	`AVG(rssi) GROUP BY antenna_id`	เสาไหนมีค่าเฉลี่ยต่ำ = จุดบอด
Read Rate per Antenna	`COUNT(*) / time`	เสาไหนอ่านได้น้อย = อาจมีจุดบอด
RSSI StdDev	`STDDEV(rssi)`	ค่าเบี่ยงเบนสูง = สัญญาณไม่เสถียร
Phase Variance	`VARIANCE(phase)`	Phase เปลี่ยนเร็ว = Tag เคลื่อนที่/สัญญาณสะท้อน
Detection Probability	`unique_tags / total_tags`	ต่ำกว่า 95% = มีจุดบอด

5.2 SQL Query สำหรับวิเคราะห์

```
-- 1. ค่า RSSI เฉลี่ยต่อเสาอากาศ
SELECT
    antenna_id,
    AVG(rssi) as avg_rssi,
    MIN(rssi) as min_rssi,
    MAX(rssi) as max_rssi,
    STDDEV(rssi) as rssi_stddev,
    COUNT(*) as read_count
FROM rssi_samples
WHERE sample_time >= NOW() - INTERVAL 1 HOUR
GROUP BY antenna_id;

-- 2. หา Tag ที่อ่านได้ไม่ครบทุกเสา (อาจอยู่ในจุดบอด)
SELECT
    tag_epc,
    COUNT(DISTINCT antenna_id) as antenna_count,
    AVG(rssi) as avg_rssi
FROM rssi_samples
GROUP BY tag_epc
HAVING antenna_count < 4  -- อ่านได้จากไม่ถึง 4 เสา
ORDER BY avg_rssi ASC;

-- 3. วิเคราะห์ตามช่วงเวลา (หาช่วงเวลาที่มีปัญหา)
SELECT
    DATE_FORMAT(sample_time, '%H:%i:00') as time_slot,
    antenna_id,
    AVG(rssi) as avg_rssi,
    COUNT(*) as reads
FROM rssi_samples
```

```
GROUP BY time_slot, antenna_id
ORDER BY time_slot, antenna_id;

-- 4. วิเคราะห์ตามความถี่ (หาความถี่ที่มีปัญหา)
SELECT
    freq_khz,
    antenna_id,
    AVG(rssi) as avg_rssi,
    COUNT(*) as reads
FROM rssi_samples
GROUP BY freq_khz, antenna_id
ORDER BY freq_khz, antenna_id;
```

5.3 การแสดงผลแบบ Heatmap

```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns

# โหลดข้อมูลจาก DB
df = pd.read_sql("""
    SELECT antenna_id, rssi, freq_khz, phase_rad, sample_time
    FROM rssi_samples
    WHERE sample_time >= NOW() - INTERVAL 1 HOUR
""", conn)

# 1. Heatmap: Antenna vs Time
pivot = df.pivot_table(
    index='antenna_id',
    columns=pd.Grouper(key='sample_time', freq='5T'),
    values='rssi',
    aggfunc='mean'
)
sns.heatmap(pivot, cmap='RdYlGn', annot=True)
plt.title('RSSI Heatmap by Antenna & Time')
plt.show()

# 2. Boxplot: เปรียบเทียบการกระจายของ RSSI ต่อเสา
sns.boxplot(data=df, x='antenna_id', y='rssi')
plt.title('RSSI Distribution per Antenna')
plt.ylabel('RSSI (dBm)')
plt.show()

# 3. Phase Plot (สำหรับวิเคราะห์ Multipath)
plt.scatter(df['sample_time'], df['phase_rad'],
            c=df['rssi'], cmap='viridis', alpha=0.5)
plt.title('Phase over Time (colored by RSSI)')
plt.show()
```

---

6. เกณฑ์การตัดสินจุดบอด (Threshold Guidelines)

ค่า RSSI	ระดับ	การตีความ	การดำเนินการ
-30 ถึง -50 dBm	🟢 Excellent	สัญญาณดีมาก	ปกติ
-50 ถึง -65 dBm	🟡 Good	สัญญาณดี	ปกติ
-65 ถึง -75 dBm	🟠 Fair	สัญญาณปานกลาง	ตรวจสอบระยะทาง
-75 ถึง -85 dBm	🔴 Poor	สัญญาณอ่อน	**จุดบอด!** ปรับมุมเสา
< -85 dBm	⚫ Critical	สัญญาณแย่มาก	ต้องปรับแต่งใหม่

เกณฑ์เพิ่มเติม:

- ถ้า **Read Rate < 95%** ของ Tag ที่คาดไว้ → มีจุดบอด

- ถ้า **RSSI StdDev > 10 dB** → สัญญาณไม่เสถียร (อาจมีสิ่งกีดขวาง)
- ถ้า **Phase เปลี่ยนแบบสุ่ม** → มี Multipath สูง (ต้องดูดซับสัญญาณสะท้อน)

---

7. ข้อแนะนำในการปรับมุมเสาอากาศ

7.1 ขั้นตอนการทำ Site Survey

1. **ติดตั้ง Temp Tag** ที่ตำแหน่งต่างๆ ในพื้นที่ (Grid 1x1 เมตร)
2. **เปิด Continuous Read** เก็บข้อมูล RSSI ต่อเนื่อง 5-10 นาทีต่อจุด
3. **วิเคราะห์ข้อมูล** ด้วย Query ด้านบน
4. **ทำ Heatmap** ของพื้นที่
5. **ปรับมุมเสา** ในบริเวณที่ RSSI ต่ำ
6. **วัดซ้ำ** เพื่อยืนยันผล

7.2 เทคนิคการปรับมุม

ปัญหา	สาเหตุที่เป็นไปได้	วิธีแก้
RSSI ต่ำเฉพาะเสาใดเสาหนึ่ง	มุมเสาไม่ครอบคลุม	ปรับ Tilt/Azimuth
RSSI ต่ำทุกเสาในจุดหนึ่ง	มีสิ่งกีดขวาง (โลหะ/น้ำ)	ย้ายเสา/ใช้ RFID Absorber
RSSI แกว่งมาก	Multipath จากพื้น/เพดาน	ใช้ RFID Foam/Absorber
อ่านได้เฉพาะบางความถี่	ความถี่นั้นถูก RFI	ปรับ Frequency List (MID=0x05)
Phase ไม่คงที่	Tag เคลื่อนที่เร็ว	ลด Q Value, เพิ่ม Power

7.3 การใช้ Phase Data เพิ่มเติม

ค่า **Phase** มีประโยชน์มากในการวิเคราะห์:

- **Phase Slope** → บอกทิศทางเคลื่อนที่ของ Tag
- **Phase Variance** → บอกความเสถียรของสัญญาณ
- **Phase vs Frequency** → ใช้คำนวณตำแหน่ง Tag แบบ RFID Phase-based Localization (ความแม่นยำระดับ cm)

---

8. Checklist สำหรับทีมพัฒนา

- [ ] ตั้งค่า **Filter Time = 0** (MID=0x09) เพื่อให้ได้ RSSI ทุกครั้ง
- [ ] ตั้งค่า **RSSI Threshold = 0** เพื่อเก็บข้อมูลสัญญาณอ่อน
- [ ] เปิด **Continuous Read** (MID=0x10, Mode=1)
- [ ] เปิดทั้ง 4 เสาอากาศ (Antenna Port Bitmask = 0x0000000F)
- [ ] Parse Notification (Flag=1, MID=0x00) ให้ได้ **RSSI, Antenna ID, Phase, Frequency, UTC**
- [ ] เก็บข้อมูลลง Database พร้อม Index
- [ ] สร้าง Dashboard แสดง Heatmap / Boxplot
- [ ] ตั้ง Alert เมื่อ RSSI เฉลี่ยต่ำกว่า -75 dBm
- [ ] ทำ Site Survey ทุก 3-6 เดือน

📎 สรุป

Reader ของคุณรองรับข้อมูลครบถ้วนสำหรับการทำ **Blind Spot Analysis** แล้วครับ สิ่งที่ต้องทำคือ:

1. **ตั้งค่า Reader** ให้ส่ง RSSI ทุกครั้ง (Filter=0)
2. **Parse Notification** ให้ได้ RSSI + Antenna ID + Phase + Freq + Time
3. **เก็บข้อมูลลง DB** พร้อมทำ Index
4. **วิเคราะห์ด้วย SQL/Python** เพื่อหาจุดบอด
5. **ปรับมุมเสา** ตามผลวิเคราะห์

หากต้องการให้ช่วยเขียน **โค้ด Parser เต็มรูปแบบ** หรือ **Dashboard Template** เพิ่มเติม แจ้งได้เลยครับ 🚀