UWB Indoor Positioning System

หลักการทำงานทั้งหมด · โปรเจกต์ **uwbtt2** (SK TechChange) · 21 มิ.ย. 2026

ภาพรวมระบบ

ระบบหาตำแหน่งภายในอาคาร (indoor positioning) แบบ real-time โดยใช้คลื่นวิทยุ **UWB (Ultra-Wideband)** ผ่านชิป **Qorvo DW3000** บนบอร์ด **ESP32-S3** แบ่งการทำงานเป็น 3 ชั้น:

```
ฮาร์ดแวร์ (Tag + Anchors)  →  คลาวด์ (Firebase)  →  เว็บ (เบราว์เซอร์)
```

1. ชั้นฮาร์ดแวร์ (Edge)

Tag — ตัวที่เคลื่อนที่ (initiator)

- ESP32-S3 + DW3000 ที่พกติดตัวหรือติดวัตถุที่ต้องการตามหา
- วัดระยะไปยัง **Anchor 1** → **2** → **3** ทีละตัวในแต่ละรอบ ด้วยเทคนิค **SS-TWR (Single-Sided Two-Way Ranging)**
- หลักการ SS-TWR: Tag ส่งสัญญาณ → Anchor ตอบกลับ → Tag วัด **เวลาไป-กลับ** แล้วคำนวณระยะจากความเร็วแสง
- ได้ระยะ `d1, d2, d3` (เมตร) แล้วส่งขึ้น Firebase ที่ `/live` ผ่าน WiFi (~3 ครั้ง/วินาที)

Anchors — 3 ตัวที่ติดตั้งอยู่กับที่ (responder)

- ใช้ **เฟิร์มแวร์ตัวเดียวกัน** flash ลงทั้ง 3 บอร์ด เปลี่ยนแค่ `ANCHOR_ID` = 1, 2, 3
- หน้าที่หลัก: ตอบสัญญาณ ranging ให้ Tag ตลอดเวลา
- หน้าที่พิเศษ: **calibration ที่สั่งจากเว็บ** — anchor วัดระยะระหว่างกันเอง (`d12, d13, d23`) เพื่อให้ระบบรู้ผังตำแหน่งของ anchor โดยไม่ต้องวัดเอง

2. หลักการ Calibration (จุดเด่นของระบบ)

แทนที่จะวัดระยะ anchor ด้วยมือแล้วพิมพ์เข้าไป ระบบใช้ **web-triggered calibration**:

- กดปุ่ม **"Calibrate anchors"** ในเว็บ → เว็บเขียน "token" ใหม่ลง `/control/calibrate`
- anchor ทุกตัว **poll token นั้นทุก 3 วินาที** (`CALIB_POLL_MS`)
- เมื่อเจอ token ใหม่ → anchor วัดระยะระหว่างกัน **1 ครั้ง** แล้ว `PATCH` ขึ้น `/calib`
- กติกาว่าใครเริ่ม: anchor ที่ ID น้อยกว่าเป็นฝ่ายเริ่มวัด → A1 วัด `d12, d13` · A2 วัด `d23` · A3 ตอบอย่างเดียว
- เพราะ anchor ได้ token พร้อมกัน จึง **หน่วงเวลาเริ่มต่างกัน** ตัวละ `(ID-1) × 1.2 วินาที` เพื่อให้ตัวที่เริ่มวัดทำงานตอนคู่ของมันยังฟังอยู่

> **สำคัญ:** ตอนเปิดเครื่อง anchor จะรับ token เดิมที่มีอยู่โดยไม่ calibrate → ไม่ calibrate เองตอนบูต ค่าที่วัดได้เก็บถาวรใน Firebase ต้องกดใหม่เฉพาะตอนย้าย anchor เท่านั้น

3. ชั้นคลาวด์ (Firebase)

Firebase Realtime Database ทำหน้าที่เป็น **"กระดานกลาง" (message bus)** — ทุกฝ่ายสื่อสารผ่านการอ่าน/เขียน database นี้ ไม่มีการคุยตรงกันแบบ peer-to-peer

Path	ใครเขียน	ข้อมูล
`/live`	Tag	`{d1, d2, d3, seq, ts}` ระยะ tag→anchor
`/calib`	Anchors	`{d12, d13, d23, ts}` ระยะระหว่าง anchor
`/control/calibrate`	เว็บ control	token (ตัวเลข) สั่ง calibrate
`/display`	เว็บ control	config สำหรับ viewer (zones, geometry, options)

นอกจากนี้ **Firebase Hosting** ยัง host เว็บ (static CDN) ที่ `https://uwb-positioning-a2892.web.app`

> **กลไกกันสัญญาณหลุด:** ถ้า ranging หลุดชั่วคราว ระบบจะ **ค้างค่าดีล่าสุดไว้** (สูงสุด ~9 วินาที, `RANGE_HOLD_MAX`) เพื่อให้ตำแหน่งนิ่ง ถ้าหลุดนานจริงถึงรายงานเป็น `-1` ("ไม่มีข้อมูล")

4. ชั้นเว็บ (Frontend) — การคำนวณตำแหน่งจริงอยู่ที่นี่

เว็บเป็น HTML/JS ล้วน **ไม่มี build step** แบ่งเป็น 2 หน้า:

Control site (`index.html` / `app.js`)

- หน้าตั้งค่าทั้งหมด: ใส่ Firebase host, สั่ง calibrate, ตั้ง geometry, ดู telemetry
- มี **zone editor** วาดพื้นที่ trigger บนแผนที่ แล้ว publish ค่าทั้งหมดขึ้น `/display`

Viewer site (`viewer.html` / `viewer.js`)

- หน้า **อ่านอย่างเดียว** เหมาะกับจอติดผนัง/แท็บเล็ต (แสดงชื่อ **"SK TechChange"**)
- อ่าน `/live`, `/calib`, `/display` มาแสดงแผนที่สดและ pop-up แจ้งเตือน

หัวใจการคำนวณ — Trilateration (ทำในเบราว์เซอร์)

Database เก็บแค่ "ระยะทาง" ไม่เคยเก็บ "พิกัด" — เบราว์เซอร์เป็นคนคำนวณพิกัดเอง:

- จาก `/calib` หาผัง anchor: วาง `A1=(0,0)`, `A2=(d12,0)` แล้วแก้สมการหา `A3` จาก `d13` / `d23`
- จาก `/live` (d1, d2, d3) ใช้ trilateration หาตำแหน่ง **X / Y / Z** ของ tag
- แสดงผล: ค่า X/Y/Z, fit residual (ยิ่งน้อยยิ่งแม่น), Top view (X-Y), Side view (X-Z) พร้อม smoothing (EMA)

Zone triggers — พื้นที่แจ้งเตือน

- วาดกล่องพื้นที่บนแผนที่ ตั้งชื่อ/ข้อความ/สี/ไอคอน/เสียงบีป/รูป/audio guide/ลิงก์
- เมื่อ tag เดินเข้าพื้นที่ → เด้ง pop-up + บีป + บันทึก log
- ชุดเริ่มต้นใน `zones.config.js` คือทัวร์ **SK TechChange** 6 จุด พร้อมรูป+MP3

> **เรื่องความสูง Z มีความกำกวม:** Anchor 3 ตัวกำหนด **ระนาบแบน** → Z ที่คำนวณได้ไม่รู้ว่าอยู่เหนือหรือใต้ระนาบ ต้องใช้ปุ่ม **"Tag is below anchor plane"** เลือกด้าน ถ้าอยากได้ 3 มิติแม่นจริงต้องเพิ่ม **anchor ตัวที่ 4 ที่ติดนอกระนาบ**

สรุป Flow ทั้งหมด

```
┌──────── EDGE (ESP32-S3 + DW3000) ────────┐
│  TAG ──SS-TWR──► ANCHOR 1 / 2 / 3        │
│   │ PUT /live {d1,d2,d3}    │ PATCH /calib │
└───┼─────────────────────────┼──────────────┘
    │ WiFi / HTTPS            │
    ▼                         ▼
┌──────── CLOUD (Firebase) ─────────────────┐
│  RTDB: /live /calib /control /display     │
│  Hosting: uwb-positioning-a2892.web.app   │
└───┼───────────────────────────────────────┘
    │ HTTPS polling (~3 Hz)
    ▼
┌──────── FRONTEND (browser) ───────────────┐
│  Control: index.html → /control, /display │
│  Viewer:  viewer.html ← /live /calib      │
│  Trilateration ทำในเบราว์เซอร์ → X/Y/Z     │
└───────────────────────────────────────────┘

1. Tag วัดระยะ 3 anchor → เขียน /live
2. (กด calibrate) Anchor วัดกันเอง → เขียน /calib
3. Control site เขียน config → /display
4. เบราว์เซอร์อ่านทั้งหมด → คำนวณ geometry + trilateration → แสดงแผนที่ + แจ้งเตือน zone
```

ทุกอย่างเชื่อมกันผ่าน Firebase เท่านั้น ทำให้แยกฮาร์ดแวร์ คลาวด์ และเว็บออกจากกันอย่างชัดเจน