คู่มือการใช้งาน

LoRa SOS Relay Demo — หลักสูตร AIR (Search & Rescue)

โปรเจกต์นี้จำลองสถานการณ์ภัยพิบัติที่เสาสัญญาณโทรศัพท์ล่ม โดยใช้บอร์ด LILYGO T-Beam ส่งสัญญาณ SOS พร้อมพิกัด GPS ผ่านคลื่น LoRa (923MHz) ไปยังศูนย์บัญชาการกลางโดยตรง ไม่พึ่งพาเครือข่ายมือถือใดๆ
เอกสารนี้สรุปทุกอย่างที่ต้องติดตั้งและขั้นตอนใช้งานไฟล์ทั้งหมดในโฟลเดอร์นี้

1. ไฟล์ในโฟลเดอร์นี้คืออะไรบ้าง

ไฟล์	คำอธิบาย
LoRa_SOS_Beacon/LoRa_SOS_Beacon.ino	โค้ดฝั่งตัวส่ง SOS — อ่านปุ่มกด + พิกัด GPS + แสดงผลจอ OLED แล้วส่งผ่าน LoRa
LoRa_SOS_Receiver/LoRa_SOS_Receiver.ino	โค้ดฝั่งตัวรับสัญญาณ — รับ SOS จากทุก Beacon แล้วส่งออกทาง Serial (ไม่มีจอ OLED)
sos_live_map.py	สคริปต์ Python รันบนโน้ตบุ๊กที่ต่อกับ Receiver เปิดแผนที่สดอัตโนมัติเมื่อมี SOS เข้ามา
map.html	หน้าเว็บแผนที่ที่ sos_live_map.py เปิดให้ดู (สลับดาวเทียม/ถนนได้ มีปุ่มดาวน์โหลด GeoJSON)
sos_log.geojson	ไฟล์ผลลัพธ์พิกัด SOS ที่บันทึกไว้ (สร้าง/อัปเดตอัตโนมัติตอนรัน sos_live_map.py)
THSarabunNew*.ttf	ไฟล์ฟอนต์ที่ใช้สร้างเอกสาร PDF นี้

2. โปรแกรมและไลบรารีที่ต้องติดตั้ง

2.1 Arduino IDE (สำหรับอัปโหลดโค้ดใส่บอร์ด T-Beam)

ดาวน์โหลด Arduino IDE เวอร์ชั่น 2.x จาก arduino.cc/en/software

Board support (Tools > Board > Boards Manager):

- ค้นหา "esp32" แล้วติดตั้งตัวที่ชื่อ esp32 by Espressif Systems

Library (Tools > Manage Libraries):

- RadioLib โดย Jan Gromes — ควบคุมโมดูล LoRa
- XPowersLib โดย lewisxhe — ควบคุมชิปจ่ายไฟ AXP2101 บนบอร์ด
- TinyGPSPlus โดย Mikal Hart — ถอดรหัสข้อมูล GPS (ใช้เฉพาะฝั่ง Beacon)
- ESP8266 and ESP32 OLED driver for SSD1306 displays โดย ThingPulse — จอ OLED (ใช้เฉพาะฝั่ง Beacon)

ตั้งค่าบอร์ดก่อนอัปโหลด (เมนู Tools):

- Board: ESP32 Dev Module
- Upload Speed: 921600 (ถ้าอัปโหลดไม่ผ่านให้ลดเป็น 115200)

2.2 Python (สำหรับแผนที่สด — ใช้บนคอมที่ต่อกับ Receiver เท่านั้น)

ดาวน์โหลด Python 3.x จาก python.org/downloads (เช็คว่ามีอยู่แล้วหรือยัง เปิด Command Prompt พิมพ์ python --version) จากนั้นติดตั้งไลบรารีด้วยคำสั่ง:

```
pip install pyserial
```

3. การต่อสายฮาร์ดแวร์ (เฉพาะ Beacon)

ตัว Receiver ไม่ต้องต่อสายอะไรเพิ่มเลย ใช้แค่บอร์ด T-Beam เปล่าเสียบ USB เข้าโน้ตบุ๊กเท่านั้น ส่วน Beacon ต้องต่อเพิ่ม 2 ส่วนคือโมดูล GPS และปุ่มกด SOS ตามนี้:

โมดูล GPS (GY-NEO-M8N)

- VCC ของ GPS --> ขา 3.3V บน T-Beam
- GND ของ GPS --> ขา GND บน T-Beam
- TX ของ GPS --> ขา GPIO 15 บน T-Beam
- RX ของ GPS --> ขา GPIO 13 บน T-Beam

ปุ่มกด SOS

- ขาหนึ่งของปุ่ม --> ขา GPIO 14 บน T-Beam
- อีกขาของปุ่ม --> ขา GND บน T-Beam (ใช้จุด GND ร่วมกับ GPS ได้ ไม่ต้องแยก)

หมายเหตุ: ปุ่มกดแบบ 4 ขา ต้องใช้ขาคู่ที่อยู่คนละฝั่งกัน (ไม่ใช่ขาคู่ที่ติดกันฝั่งเดียว
เพราะคู่นั้นต่อกันตลอดเวลาอยู่แล้วโดยไม่ต้องกด) แนะนำเช็ค Continuity ด้วยมัลติมิเตอร์ก่อนต่อจริง
เพื่อยืนยันว่าใช้ขาคู่ที่ถูกต้อง (ร้องเฉพาะตอนกดเท่านั้น)

4. ขั้นตอนอัปโหลดโค้ด

Beacon

- เปิดไฟล์ LoRa_SOS_Beacon.ino ใน Arduino IDE
- ถ้ามี Beacon หลายตัว ให้เปลี่ยนค่า BEACON_ID ในโค้ด (บรรทัดบนๆ ของไฟล์) ให้ไม่ซ้ำกันในแต่ละตัว เช่น "B01", "B02", ...
- เลือก Tools > Port ให้ตรงกับบอร์ดที่เสียบอยู่
- กด Upload รอจนขึ้น "Hash of data verified"

Receiver

- เปิดไฟล์ LoRa_SOS_Receiver.ino ใน Arduino IDE
- เลือก Tools > Port ให้ตรงกับบอร์ด Receiver
- กด Upload รอจนขึ้น "Hash of data verified"

5. การใช้งานแผนที่สด (หลังอัปโหลดโค้ดเสร็จแล้ว)

- เสียบ Receiver ไว้กับโน้ตบุ๊กผ่าน USB ค้างไว้ตลอด (ปิด Serial Monitor ใน Arduino IDE ก่อน ไม่งั้นพอร์ตจะชนกัน)
- เปิด Command Prompt แล้วพิมพ์คำสั่งนี้เพื่อไปที่โฟลเดอร์นี้:

```
cd "C:\Users\Asus\Desktop\Claude\BOI\SoT"
```

- เช็คเลข COM Port ของ Receiver จาก Arduino IDE > Tools > Port แล้วรันคำสั่ง (เปลี่ยน COM8 เป็นเลขจริง):

```
python sos_live_map.py COM8
```

- เบราว์เซอร์จะเปิดหน้าแผนที่ขึ้นมาอัตโนมัติ รอ Beacon กดปุ่ม SOS จุดจะขึ้นบนแผนที่ทันที
- กดปุ่ม "ดาวน์โหลด GeoJSON" มุมซ้ายบนของแผนที่ เพื่อ export ไฟล์ผลลัพธ์เมื่อไหร่ก็ได้
- ต้องมีอินเทอร์เน็ตเปิดไว้ระหว่างใช้งาน (โหลดภาพแผนที่พื้นหลังจาก OpenStreetMap/Esri)