



โครงการ

เรื่อง นาฬิกาอัจฉริยะ

จัดทำโดย

กีรติ เศรษฐสุข 6733021521

ปรินทร ธรรมรักษ์ 6733140521

พิศิษฐ์ กาญจนภิญพงศ์ 6733178421

วชรวิศร์ ปราษุ์เปี่ยมสุข 6733238421

วิชชุกร เติมวรคุณ 6733240621

โครงการนี้เป็นส่วนหนึ่งของวิชา

ระบบสมองกลฝังตัว รหัสวิชา 2110356

สาขาวิศวกรรมคอมพิวเตอร์และเทคโนโลยีดิจิทัล

ภาควิชาบริการคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2568

บทที่ 1

บทนำและคำอธิบายปัญหา

1.1 ความเป็นมาและความสำคัญของปัญหา

ในการใช้ชีวิตในยุคปัจจุบันที่มีภาระงานและมลภาวะที่มากขึ้น ในปัจจุบันที่มีภาระงานจำนวนมาก อาจจะทำให้สูญเสียเวลาและทรัพยากร สำคัญ การต้องคอยหยิบโทรศัพท์มือถือเพื่อตรวจสอบ อาจจะสร้างความไม่สะดวกในบางสถานการณ์ หรือในบางครั้งต้องการทราบสถานะค่าต่างๆ ภายในห้อง เช่น อุณหภูมิ ค่าฝุ่น แต่ไม่สามารถตรวจสอบได้ง่าย

ดังนั้น การมีอุปกรณ์ที่สามารถดูสถานะต่างๆ การแจ้งเตือนตารางงาน และสามารถตอบโต้ด้วยเสียงได้โดยไม่จำเป็นต้องสัมผัสหน้าจอ จะช่วยสามารถอำนวยความสะดวกได้มากขึ้น

1.2 วัตถุประสงค์ของโครงงาน

เพื่อพัฒนาอุปกรณ์ที่สามารถดูแลความสะอาดในชีวิตประจำวัน ในด้านการอ่านค่าสถานะต่างๆ ภายในห้อง และด้านการแจ้งเตือน กำหนดการต่าง ๆ ตามที่ตั้งไว้ สามารถเข้าถึงผ่านออนไลน์ได้ผ่านเว็บไซต์ที่ทำไว้ เพื่อความสะดวกในการติดตามข้อมูลและเพิ่มข้อมูล และสามารถสั่งด้วยเสียง เสียงได้พร้อมระบบพูดตอบกลับมาด้วยเสียง

1.3 ขอบเขตของโครงงาน

1.3.1 ขอบเขตด้านเนื้อหา

- ใช้บอร์ด ESP32 จำนวน 2 ตัว แบ่งการทำงานเป็น Sensor Node และ Gateway
- รองรับการวัดค่า: อุณหภูมิ, ความชื้น, แสงสว่าง, ฝุ่น PM2.5 และเวลาปัจจุบัน (RTC)
- รองรับการสั่งงานด้วยเสียงผ่านไมโครโฟน และตอบโต้ผ่านลำโพง

1.3.2 ขอบเขตด้านซอฟต์แวร์

- เก็บข้อมูล Sensor ลงบน NETPIE
- จัดการ Task แจ้งเตือนผ่าน Google Sheet และ Google Apps Script
- ประมวลผลภาษาธรรมชาติ (NLP) ผ่าน Gemini API
- แสดงผลผ่าน Web Application (React + Vite)

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- ผู้ใช้สามารถรับสัญญาณแวดล้อมภายในห้องที่มีอุปกรณ์นี้ว่าเป็นอย่างไร ณ ปัจจุบัน
- ผู้ใช้สามารถจัดตารางงานได้สะดวกและสามารถแจ้งเตือนได้ทั้งผ่านการสั่งงานด้วยเสียง และบนเว็บไซต์

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

เพื่อจัดทำโครงงานนี้ ผู้ดัดทำได้ทำการศึกษาเอกสารจาก UIUC Spring 2025 CS598 LLM Agent Workshop Submission หัวข้อ

เรื่อง LLM Agents for Internet of Things (IoT) Applications

เทคโนโลยี Internet of Things (IoT) กำลังเติบโตอย่างรวดเร็วและถูกนำไปใช้ในหลายอุตสาหกรรมเพื่อช่วยให้มีการตรวจสอบแบบเรียลไทม์ ระบบอัตโนมัติ และการวิเคราะห์ข้อมูลเชิงลึกมากขึ้น แต่ระบบ IoT ยังประสบปัญหา เช่น ข้อมูลดิบที่หลากหลาย ขาดความเข้าใจตามบริบท และการวิเคราะห์ที่ยังพึ่งพากฎแบบตายตัวมากเกินไป

ในขณะเดียวกัน Large Language Models (LLMs) แสดงศักยภาพสูงด้านความเข้าใจภาษา การให้เหตุผล และการโต้ตอบกับมนุษย์ การผสาน LLM เข้ากับ IoT จึงกลายเป็นหัวข้อวิจัยที่สำคัญ เพราะช่วยให้ระบบ IoT ที่มีผู้ใช้งานจำนวนมากทำงานได้ฉลาดขึ้น ยืดหยุ่นขึ้น และใช้งานง่ายขึ้น

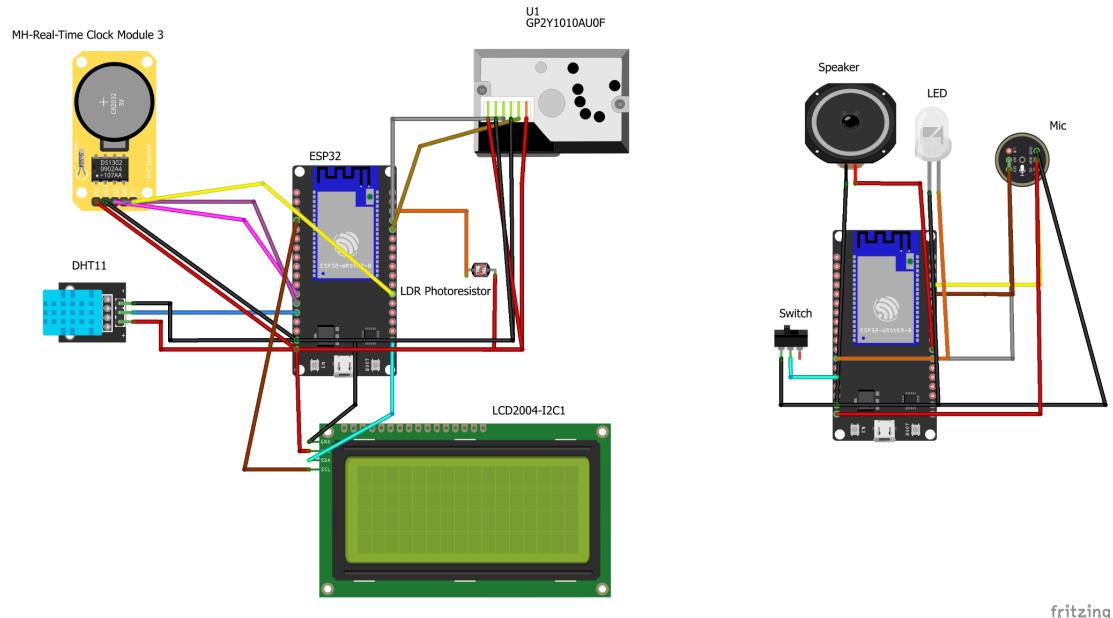
บทความสำรวจนี้มุ่งศึกษาการใช้งาน LLM ใน IoT ตั้งแต่ edge intelligence อินเทอร์เฟซอัจฉริยะ การสั่งงานแบบไดนามิก ไปจนถึงการโต้ตอบเชิงสร้างสรรค์ รวมถึงวิเคราะห์ความท้าทาย เช่น ภาระการประมวลผล ความหน่วง และความเป็นส่วนตัวของข้อมูลเมื่อใช้ LLM ในอุปกรณ์ทรัพยากรจำกัด

จากการทบทวนงานวิจัยล่าสุด บทความชี้ให้เห็นแนวโน้มสำคัญ สถานะปัจจุบันที่ใช้จริง เทคนิคการเพิ่มประสิทธิภาพ ตลอดจนประเด็นด้านเทคนิค จริยธรรม และการปฏิบัติ นอกเหนือไปนี้ยังเสนอทิศทางอนาคตสำหรับการพัฒนาระบบ IoT ที่ช่วยลด มีประสิทธิภาพ และเข้าใจบริบทมากขึ้นด้วยพลังของ LLM agents

บทที่ 3

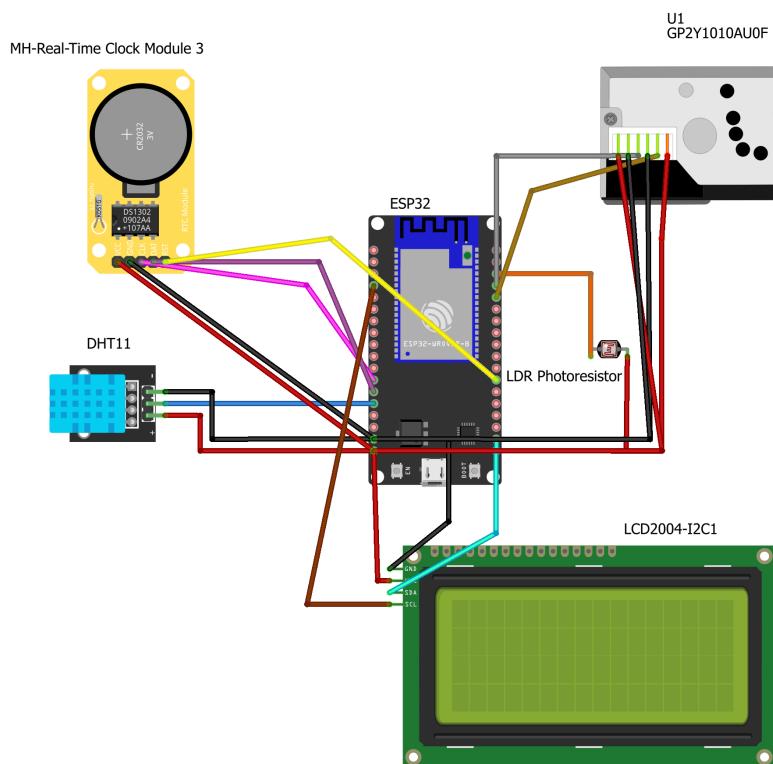
การออกแบบระบบ

3.1 Overview diagram



รูปภาพที่ 1 ภาพการทำงาน overview

3.2 Sensor node



รูปภาพที่ 2 ภาพการทำงาน overview ส่วน sensor node

ในส่วนของ Sensor node จะทำหน้าที่อ่านข้อมูลสภาพแวดล้อม และส่งข้อมูลให้ Gateway โดยจะมีชี้ Sensor ดังนี้

3.2.1 ESP 32: ทำหน้าที่ประมวลผลหลักส่วน Sensor Node

3.2.2 DHT11: ใช้วัดอุณหภูมิและความชื้นสัมพัทธ์

3.2.3 LDR Photoresistor: ใช้วัดค่าความเข้มแสงสว่าง

3.2.4 GP2Y1010AU0F: ใช้วัดค่าฝุ่นละออง PM2.5

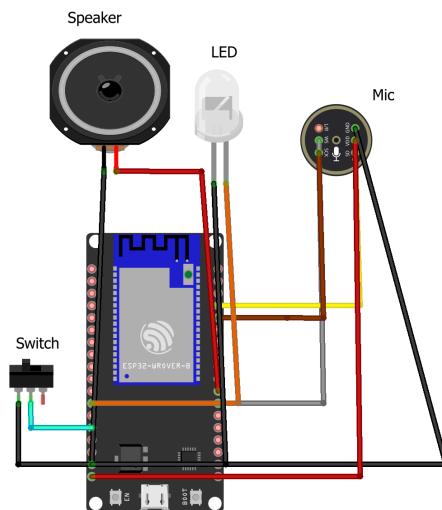
3.2.5 RTC (Real Time Clock): ใช้สำหรับทำให้อุปกรณ์ยังเป็นเวลาตามปัจจุบัน

3.2.6 LCD2004: หน้าจอใช้สำหรับแสดงข้อมูลต่าง ๆ แบบ Matrix

หลักการทำงาน

- ESP32 Sensor จะทำการอ่านค่าจากเซ็นเซอร์ที่อยู่กับบอร์ดนี้และเก็บข้อมูลไว้
- LCD จะแสดงผลเวลา อุณหภูมิ ความชื้น ความเข้มแสง ค่าฝุ่น จาก เซ็นเซอร์
- ESP32 Sensor จะทำการส่งค่าสถานะต่าง ๆ ไปหา ESP32 Gateway ผ่าน esp now

3.3 Gateway Node



รูปภาพที่ 2 ภาพการทำงาน overview ส่วน gateway node

ในส่วนของ Gateway node จะทำหน้าที่เป็นศูนย์กลางการสื่อสาร , เชื่อม WiFi , จัดการระบบรับเสียงและพูดเสียงและระบบ Ai โดยจะมีชี้

Sensor

3.3.1 ESP32 32: ทำหน้าที่ประมวลผลหลักส่วน Gateway

3.3.2 INMP441: เป็นไมโครโฟนแบบ I2S สำหรับรับคำสั่งเสียง

3.3.3 Speaker: เป็นลำโพงสำหรับส่งเสียงโดยต้องแตะแจ้งเตือน (Actuator)

3.3.5 Slider Switch: สวิสต์สำหรับเปิด-ปิดการทำงานของไมโครโฟน

หลักการทำงาน

- ระบบเสียง
 - รับเสียงจาก INMP441
 - ส่งไฟล์เสียงไปยัง Service LLM เพื่อแปลงไฟล์เสียงเป็นข้อความ
 - หลังจากนั้นจะส่งข้อความไปยัง Gemini API
 - Gemini จะส่งคำตอบมาเป็นข้อความมายัง Service LLM
 - Service LLM จะแปลงข้อความเป็นไฟล์เสียง แล้วเล่นผ่าน Speaker
- ระบบส่งค่า Sensor ขึ้นเว็บไซต์
 - รับค่าเซ็นเซอร์ต่างๆ จาก ESP32 Sensor ผ่าน ESP NOW
 - นับค่าที่ได้ขึ้นไปบน netpie เพื่อให้เว็บไซต์สามารถดึงค่ามาจาก netpie ได้
- ระบบ Task และการเตือน
 - รับคำสั่งจาก Service ที่ทำงานไว้วางเวลาแจ้งเตือน
 - ระบบจะทำการแจ้งเตือนผ่านเสียง

3.4 Cloud/ Storage

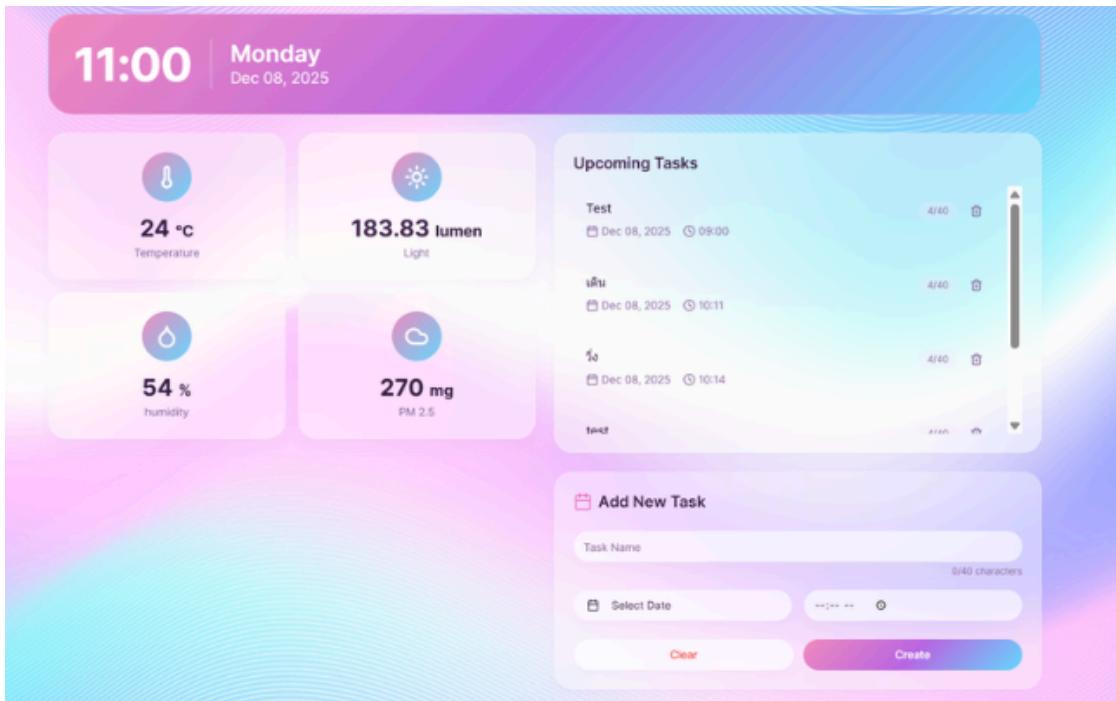
ระบบจะมีการใช้งาน Cloud 2 ส่วนหลัก ๆ คือ

- NETPIE
 - ใช้สำหรับเก็บข้อมูลเซนเซอร์ (อุณหภูมิ, ความชื้น, ผู้นับ, ค่าแสง) เพื่อนำไปแสดงบนเว็บไซต์ที่ทำเอาไว้
- Google Sheets & Apps Script
 - ใช้เป็น Database สำหรับเก็บข้อมูล Task ต่างๆ โดยระบบจะดึงข้อมูลนี้มาเพื่อแจ้งเตือน

3.5 Dashboard

- Web Dashboard: พัฒนาด้วย React และ Vite (Host บน Vercel)
 - แสดงค่า Real-time Sensor จาก NETPIE
 - แสดงรายการ Task ที่ดึงมาจาก Google Sheets

- Notification System
 - ระบบจะรัน Service ตรวจสอบเวลาปัจจุบันเทียบกับเวลาของ Task ใน Google Sheets หากถึงกำหนดเวลา จะส่งข้อมูล text ไปแปลงเป็นเสียงแจ้งเตือนผ่านลำโพงที่ตัวนาฬิกาทันที



รูปภาพที่ 3 หน้าเว็บไซต์ Dashboard

บทที่ 4

ผลการทดลอง

4.1 อภิปรายผลการทดลอง

1. Sensor Node & Display

การทำงานของ Sensor Node สามารถตรวจวัดค่าจาก DHT11, LDR และ GP2Y1010AU0F ได้อย่างต่อเนื่อง หน้าจอ LCD2004 สามารถแสดงผลค่าต่างๆ และเวลาจาก RTC ได้อย่างชัดเจน อย่างไรก็ตามเซ็นเซอร์วัดฝุ่นจำเป็นต้องมีการ Calibrate ค่าเริ่มต้นเพื่อให้ได้ข้อมูลที่ใกล้เคียงความจริงมากที่สุด

2. Gateway & AI Voice Control

การรับคำสั่งเสียงผ่าน INMP441 และส่งไปประมวลผลที่ Gemini API สามารถทำงานได้ตามฟังก์ชันที่ออกแบบไว้ คือสามารถตอบคำถ้าสภาพอากาศ และบันทึก Task ลง Google Sheet ได้ แม้มีความหน่วง (Latency) เล็กน้อย ขึ้นอยู่กับความเร็วอินเทอร์เน็ตในการส่งไฟล์เสียงไป-กลับ

3. Cloud Integration

การแยกเก็บข้อมูลระหว่าง NETPIE (Sensor Data) และ Google Sheets (Task Data) ช่วยให้ระบบทำงานได้รวดเร็ว โดย Google Apps Script ทำหน้าที่เป็น API Middleware ที่ดูในการเชื่อมต่อระหว่าง ESP32 และ Google Sheets

4. Dashboard

เว็บไซต์ที่พัฒนาด้วย React มีความสวยงามและตอบสนองได้ดี (Responsive) สามารถดูค่าจาก nanopress ได้จากทุกที่ที่มีอินเทอร์เน็ต

บทที่ 5

สรุปโครงงานและข้อเสนอแนะเพิ่มเติม

5.1 สรุปโครงงาน

โครงงานนี้มีเป้าหมายที่จะพัฒนาความสามารถในการทำงานได้ตามวัตถุประสงค์ที่ตั้งไว้ ครอบคลุมตามข้อกำหนด (Requirement) ของรายวิชา โดยมีการใช้งานเซ็นเซอร์มากกว่า 5 ชนิด มีการสื่อสารระหว่างบอร์ด 2 ตัว เสียบต่อ Cloud 2 แพลตฟอร์ม และมีการนำ AI มาช่วยในการประมวลผลคำสั่งเสียง ทำให้เกิดเป็นอุปกรณ์ IoT ที่มีความทันสมัยและใช้งานได้จริง

5.2 ข้อเสนอแนะเพิ่มเติม

- การจัดการพลังงาน: ปัจจุบันต้องเสียบปลั๊กตลอดเวลา ในอนาคตควรพัฒนาวงจรจัดการแบตเตอรี่ (Power Management) เพื่อให้เป็นอุปกรณ์พกพาได้จริง
- ความเร็วในการตอบสนอง: อาจพิจารณาใช้ Edge AI หรือ Model ที่เล็กลงในการประมวลผลคำสั่งเสียงเบื้องต้นบนบอร์ด เพื่อลดความหน่วงจากการใช้อินเทอร์เน็ต
- Hardware Design: การออกแบบ PCB และเคส 3D Print จะช่วยให้อุปกรณ์มีความสวยงามและกะทัดรัดมากขึ้น

บทบาทและความรับผิดชอบของสมาชิกในทีม

กีรติ เศรษฐสุข : System Development

- ทำ service llm
- ทำ service task service
- ทำ code ฝั่ง gateway node
- ควบคุมการทำงานของคนในทีม

ปรินทร์ ธรรมรักษ์ : Embedded System Development

- ทำ code รับ sensor
- ต่อ sensor ฝั่ง sensor node

พิศิษฐ์ กาญจนภิญพงศ์ : System Development + Website Development

- ทำระบบ task บน google sheet และ code
- ทำเว็บไซต์ส่วนของดึงข้อมูลจาก google sheet และแก้ไขข้อมูล
- ออกแบบส่วนของกล่องนาฬิกา

วัชรวิศร์ ปราษฐ์เพี่ยมสุข : System Development + Website Development

- เขียน code ให้ gateway node ส่งข้อมูล sensot ไปยัง netpie
- ทำเว็บไซต์และทำในส่วนของการต่อ sensor มาจาก netpie

วิชชุกร เติมารคุณ : Embedded System Development

- ออกรอบบนเว็บไซต์
- ทำ code อ่าน sensor ไมโครโฟน และเล่นเสียงสำเนา
- เขียน code ส่วนของพิ้งเสียงและส่งไปที่ service llm