



## Smart Window Control System Using ESP32

### รายวิชา

การออกแบบระบบอินเทอร์เน็ตของสรรพสิ่ง

[31-407-105-404]

### จัดทำโดย

นายธีรภัทร์ ถมคำ 66332110062-4

นายภาณุภัทร กันหาวรรณะ 66332110285-8

### อาจารย์ผู้สอน

อาจารย์ประภาส ฝ่องสนาม

### สาขาวิชา

วิศวกรรมคอมพิวเตอร์ (ECP3N)

มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน วิทยาเขตขอนแก่น

## บทคัดย่อ

โครงการนี้นำเสนอระบบควบคุมหน้าต่างอัจฉริยะ (Smart Window Control System) โดยใช้ ESP32 และเทคโนโลยี Internet of Things (IoT) สำหรับการตรวจวัดสภาพแวดล้อมและควบคุมการเปิด-ปิดหน้าต่างแบบอัตโนมัติ ระบบทำการวัดค่าอุณหภูมิด้วยเซนเซอร์ DHT22 และวัดค่าความเข้มแสงด้วยเซนเซอร์ LDR ซึ่งแปลงค่าเป็นหน่วยลักซ์ (lux) ข้อมูลที่ได้จะถูกส่งแบบเรียลไทม์ผ่านโปรโตคอล WebSocket ไปยังเซิร์ฟเวอร์ จากนั้นจัดเก็บลงใน Firebase Realtime Database และแสดงผลผ่านเว็บแอปพลิเคชันในรูปแบบ Gauge สำหรับอุณหภูมิ และกราฟ (Graph) สำหรับค่าความเข้มแสง

ระบบสามารถควบคุมการเปิด-ปิดหน้าต่างได้ทั้งแบบอัตโนมัติตามเงื่อนไขที่กำหนด และแบบสั่งการผ่านหน้า Dashboard นอกจากนี้ยังมีระบบแจ้งเตือนผ่าน Telegram เมื่ออุณหภูมิหรือความเข้มแสงเกินค่าที่กำหนด หรือเมื่อสถานะหน้าต่างมีการเปลี่ยนแปลง ระบบที่พัฒนาขึ้นช่วยเพิ่มประสิทธิภาพการระบายอากาศภายในอาคาร ลดการสะสมความร้อน

Github: <https://github.com/Teeraphat22/Smart-window-control.git>

Wokwi Simulation: <https://wokwi.com/projects/455875799657775105>

## บทนำ

ในปัจจุบัน เทคโนโลยี Internet of Things (IoT) ได้เข้ามามีบทบาทสำคัญในการพัฒนาระบบอัตโนมัติภายในอาคารและบ้านอัจฉริยะ (Smart Home) ระบบควบคุมหน้าต่างอัจฉริยะเป็นหนึ่งในแนวทางที่ช่วยเพิ่มประสิทธิภาพการระบายอากาศ ลดความร้อนสะสมภายในอาคาร และช่วยประหยัดพลังงานจากการใช้เครื่องปรับอากาศ

โครงการนี้พัฒนาระบบ Smart Window Control System โดยใช้ ESP32 เป็นไมโครคอนโทรลเลอร์หลัก เชื่อมต่อเซนเซอร์วัดอุณหภูมิ (DHT22) และเซนเซอร์วัดแสง (LDR) เพื่อตรวจสอบสภาพแวดล้อมแบบ Real-time ข้อมูลถูกส่งผ่าน WebSocket ไปยัง Server และจัดเก็บใน Firebase Realtime Database พร้อมแสดงผลผ่าน Web Dashboard และมีการแจ้งเตือนไป Telegram

## ทบทวนวรรณกรรม

การพัฒนาระบบตรวจสอบและควบคุมอัตโนมัติด้วยเทคโนโลยี Internet of Things (IoT) ได้รับความสนใจอย่างต่อเนื่องในงานวิจัยด้าน Smart Home และ Smart Monitoring โดยงานวิจัยที่เกี่ยวข้องให้พื้นฐานสำคัญทั้งในด้านการสื่อสารข้อมูลแบบเรียลไทม์ โครงข่ายเซนเซอร์ไร้สาย และการประมวลผลบนคลาวด์ ซึ่งเป็นแนวคิดหลักที่ถูกนำมาประยุกต์ใช้ในโครงการ Smart Window Control System

Soewito และคณะ [1] ได้ศึกษาเทคโนโลยี WebSocket เพื่อรองรับการทำงานของระบบ Smart Home แบบเรียลไทม์

โดยผลการวิจัยแสดงให้เห็นว่า WebSocket สามารถลดความหน่วง (latency) เมื่อเทียบกับการใช้ HTTP polling แบบ

เดิม และรองรับการสื่อสารสองทาง (bidirectional communication) ได้อย่างมีประสิทธิภาพ แนวคิดนี้มีความสำคัญต่อ

ระบบที่ต้องการการอัปเดตข้อมูลทันที ซึ่งสอดคล้องกับการออกแบบระบบแสดงผลข้อมูลแบบ Real-time Dashboard ในโครงการนี้

Kodali และคณะ [2] ได้นำเสนอระบบ Smart Security และ Home Automation บนพื้นฐาน IoT โดยใช้การเชื่อมต่อ

แบบไร้สายและ Web-based Monitoring ระบบดังกล่าวผสานการทำงานของเซนเซอร์ การเชื่อมต่อคลาวด์ และส่วน

ติดต่อผู้ใช้ผ่านเว็บแอปพลิเคชัน แนวคิดการจัดเก็บข้อมูลบนคลาวด์และการควบคุมอุปกรณ์ผ่านเว็บถูกนำมาประยุกต์ใช้

ในโครงการนี้ผ่านการใช้ Firebase สำหรับจัดเก็บข้อมูลแบบเรียลไทม์ และ Web Dashboard สำหรับการควบคุม

หน้าตาอัจฉริยะ

Buratti และคณะ [3] ได้อธิบายภาพรวมของเทคโนโลยี Wireless Sensor Networks (WSNs) โดยกล่าวถึงโครงสร้าง

เครือข่าย การจัดการพลังงาน และรูปแบบการสื่อสารของเซนเซอร์ไร้สาย งานวิจัยนี้ให้พื้นฐานทางทฤษฎีเกี่ยวกับการ

ออกแบบระบบตรวจวัดแบบกระจาย (distributed sensing system) ซึ่งสอดคล้องกับการใช้ ESP32 ในโครงการนี้ในการ

เชื่อมต่อ Wi-Fi และส่งข้อมูลเซนเซอร์แบบไร้สายไปยังฐานข้อมูลบนคลาวด์

Tanir และคณะ [4] ได้ศึกษาแนวทางการส่งข้อมูลแบบเรียลไทม์ระหว่าง IoT Gateway และ Cloud Platform โดยเน้น

การสื่อสารสองทางที่มีความเสถียรและหน่วงต่ำ งานวิจัยนี้เน้นความสำคัญของระบบสื่อสารที่เชื่อถือได้สำหรับงาน

Monitoring System ซึ่งถูกนำมาประยุกต์ใช้ในโครงการนี้ผ่านกลไกการซิงโครไนซ์ข้อมูลของ Firebase Realtime

Database เพื่อให้ข้อมูลระหว่าง ESP32 และ Web Application มีความสอดคล้องกันแบบทันที

### จากการทบทวนวรรณกรรม

พบว่า องค์ประกอบสำคัญของระบบ IoT สมัยใหม่ประกอบด้วย

- การสื่อสารข้อมูลแบบเรียลไทม์
- โครงข่ายเซนเซอร์ไร้สาย
- การจัดเก็บและประมวลผลข้อมูลบนคลาวด์

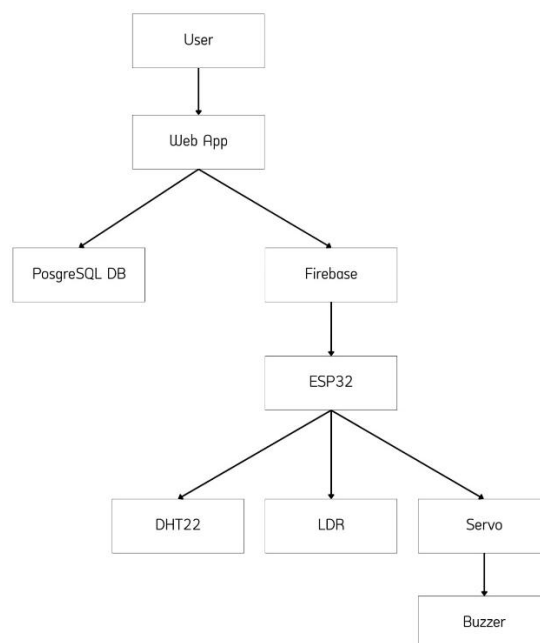
- ระบบควบคุมและแสดงผลผ่าน Web-based Dashboard

โครงการ Smart Window Control System ได้นำแนวคิดดังกล่าวมาบูรณาการร่วมกัน โดยใช้ ESP32 เป็นอุปกรณ์ Edge Device สำหรับเก็บข้อมูลเซนเซอร์ ใช้ Firebase สำหรับจัดเก็บข้อมูลแบบเรียลไทม์ และใช้ PostgreSQL สำหรับจัดการข้อมูลผู้ใช้งานและระบบยืนยันตัวตน (Authentication System) ส่งผลให้ระบบมีความครบถ้วนทั้งในด้านการสื่อสาร ความปลอดภัย และความสามารถในการขยายระบบในอนาคต

### วัตถุประสงค์

1. พัฒนาระบบควบคุมหน้าต่างอัจฉริยะโดยใช้ ESP32
2. ตรวจสอบอุณหภูมิและความเข้มแสงแบบ Real-time
3. แสดงผลผ่าน Web Dashboard (Gauge และ Graph)
4. ใช้ WebSocket ในการสื่อสารข้อมูลแบบ Real-time
5. จัดเก็บข้อมูลใน Firebase Realtime Database
6. แจ้งเตือนผ่าน Telegram เมื่ออุณหภูมิสูงเกินค่าที่กำหนด, ความเข้มแสงสูงเกินค่าที่กำหนดและสถานะหน้าต่างเปลี่ยน

### กระบวนการพัฒนาระบบ



## คำอธิบาย Block Diagram

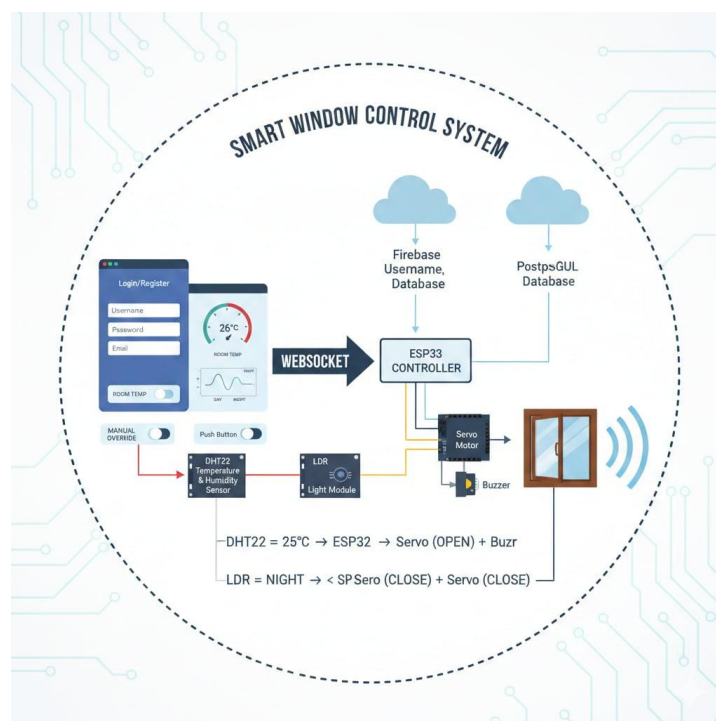
1. ESP32 รับค่าจากเซนเซอร์
2. ส่งข้อมูลไปยัง Firebase แบบเรียลไทม์
3. ผู้ใช้เข้าสู่ระบบผ่านหน้า Login
4. Node.js ตรวจสอบสิทธิ์กับ PostgreSQL
5. เมื่อยืนยันตัวตนสำเร็จ Dashboard แสดงข้อมูลจาก Firebase
6. ผู้ใช้สามารถสั่งควบคุมอุปกรณ์ผ่านระบบ

## การทำงานของระบบ

ในงานวิจัยนี้ ได้พัฒนาระบบ Smart Window Control System โดยใช้ไมโครคอนโทรลเลอร์ ESP32 เป็นหน่วยประมวลผลหลัก ทำหน้าที่เชื่อมต่อกับเซนเซอร์ตรวจวัดสภาพแวดล้อม ระบบฐานข้อมูลแบบ Cloud และเว็บแอปพลิเคชันสำหรับผู้ใช้งาน ระบบสามารถตรวจวัดข้อมูลแบบเรียลไทม์และควบคุมการเปิด-ปิดหน้าต่างตามเงื่อนไขที่กำหนด

เซนเซอร์ DHT22 ถูกเชื่อมต่อกับ ESP32 เพื่อวัดค่าอุณหภูมิ (Temperature: T) และความชื้นสัมพัทธ์ (Humidity: H) ขณะที่เซนเซอร์ LDR ใช้สำหรับตรวจวัดค่าความเข้มแสง (Light Intensity: L) ข้อมูลที่ได้จะถูกอ่านเป็นช่วงเวลา (Sampling Interval) และประมวลผลภายในตัวไมโครคอนโทรลเลอร์

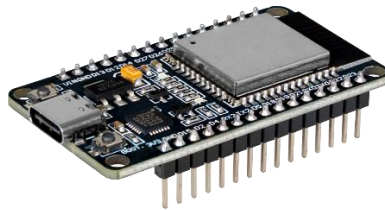
ค่าที่วัดได้จะถูกส่งผ่านเครือข่าย Wi-Fi ไปยัง Firebase Realtime Database ซึ่งทำหน้าที่เป็นฐานข้อมูลบน Cloud เพื่อจัดเก็บและแสดงผลข้อมูลแบบเรียลไทม์ผ่าน Web Dashboard



## Hardware & Tools

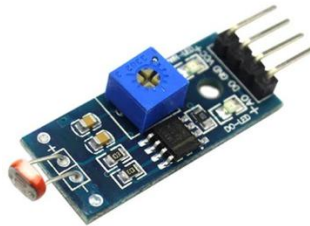
### อุปกรณ์ฮาร์ดแวร์ (Hardware Components)

#### 1) บอร์ดไมโครคอนโทรลเลอร์ ESP32



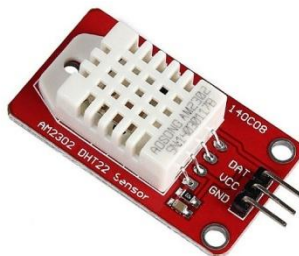
ใช้เป็นหัวใจหลักในการเชื่อมต่อและควบคุมอุปกรณ์ทุกอย่างด้วยกัน และส่งข้อมูลต่างผ่าน WiFi ไปยังระบบฐานข้อมูล และแสดงผลผ่านเว็บไซต์

#### 2) เซนเซอร์วัดความสว่าง (LDR)



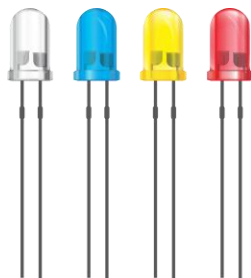
ใช้ในระบบตรวจจับความสว่าง เพื่อนำค่าไปประมวลผลในการ เปิด-ปิด หน้าต่างว่าในช่วงเวลานั้น เป็นกลางคือหรือกลางวัน

#### 3) โมดูลวัดอุณหภูมิและความชื้น (DHT22)



เซ็นเซอร์วัดอุณหภูมิและความชื้นใช้เพื่อตรวจสอบอุณหภูมิภายในห้อง และนำค่าไปประเมินว่าอุณหภูมิในห้อง นั้นร้อนเกินไป หรืออุณหภูมিপกติ

#### 4) ไดโอดเปล่งแสง (LED)



ใช้ในการบอกสถานะของตัวบอร์ด ESP32 ว่าเชื่อมต่อเสร็จสิ้นหรือยัง และใช้ในการบอกสถานะการทำงานของ Servo ว่าทำงานหรือไม่?

#### 5) เซอร์โวมอเตอร์ (Servo motor)



ใช้ Servo motor เพื่อควบคุมการ เปิด-ปิด ของหน้าต่าง โดยรับคำสั่งมาจากตัว ESP32 และการควบคุมผ่านเว็บไซต์

#### 6) บัซเซอร์ (Buzzer)



ใช้ในการแจ้งเตือนของระบบต่างๆ เช่น การ เปิด-ปิด ของหน้าต่าง ความร้อนในห้องมากเกินไป

## 7) ตัวต้านทาน (Resistor)

ใช้ในการจำกัดกระแสให้ตัว LEDs

### รายการอุปกรณ์ Software ต่างๆ

1. Arduino IDE
2. Wokwi
3. Firebase
4. Web Dashboard
5. Telegram
6. Visual Studio Code
7. PosegerSQL

### แนวทางในการต่อยอด

แม้ว่า Smart Window Control System ที่พัฒนาขึ้นจะสามารถตรวจวัดสภาพแวดล้อม ควบคุมหน้าต่างแบบอัตโนมัติ และแสดงผลแบบเรียลไทม์ได้อย่างมีประสิทธิภาพ แต่ยังมีแนวทางในการพัฒนาเพิ่มเติมเพื่อเพิ่มประสิทธิภาพ ความปลอดภัย และความสามารถในการขยายระบบ ดังนี้

#### 1. การเพิ่มเซนเซอร์ตรวจวัดสภาพอากาศ

ในอนาคตสามารถเพิ่มเซนเซอร์อื่น ๆ เช่น

- เซนเซอร์ตรวจจับฝน (Rain Sensor)
- เซนเซอร์คุณภาพอากาศ (Air Quality Sensor – PM2.5)
- เซนเซอร์ตรวจจับแก๊สหรือควัน

เพื่อให้ระบบสามารถตัดสินใจได้แม่นยำมากขึ้น และเพิ่มความปลอดภัยต่อผู้ใช้งาน

#### 2. การพัฒนาอัลกอริทึมอัจฉริยะ (Intelligent Decision System)

สามารถพัฒนาไปสู่การใช้ Machine Learning หรือ Artificial Intelligence เพื่อวิเคราะห์รูปแบบข้อมูลย้อนหลัง เช่น

- วิเคราะห์แนวโน้มอุณหภูมิ
- คำนวณช่วงเวลาที่เหมาะสมเปิด-ปิดหน้าต่าง
- ปรับค่า Threshold อัตโนมัติตามฤดูกาล



ซึ่งจะช่วยให้ระบบมีความชาญฉลาดและประหยัดพลังงานมากยิ่งขึ้น

### 3. การพัฒนา Mobile Application

ปัจจุบันระบบทำงานผ่าน Web Dashboard ในอนาคตสามารถพัฒนาเป็น Mobile Application บนระบบปฏิบัติการ Android และ iOS เพื่อเพิ่มความสะดวกในการใช้งาน และรองรับ Push Notification แบบเรียลไทม์

### 4. การเพิ่มระบบแจ้งเตือน (Notification System)

สามารถเพิ่มระบบแจ้งเตือนผ่าน

- LINE Notify
- Telegram Bot
- Firebase Cloud Messaging (FCM)

เพื่อแจ้งเตือนผู้ใช้งานเมื่อเกิดเหตุการณ์ผิดปกติ เช่น อุณหภูมิสูงเกินค่ากำหนด หรือระบบขัดข้อง

### 5. การเสริมความปลอดภัยของระบบ (Enhanced Security)

แนวทางพัฒนาเพิ่มเติมด้านความปลอดภัย ได้แก่

- การใช้ HTTPS และ SSL Certificate
- การใช้ JWT Authentication แบบหมดอายุอัตโนมัติ
- การทำ Role-based Access Control (RBAC)
- การป้องกันการโจมตีแบบ Brute Force และ SQL Injection

### 6. การขยายระบบสู่ Smart Home Integration

ระบบสามารถพัฒนาให้เชื่อมต่อกับแพลตฟอร์ม Smart Home อื่น ๆ เช่น

- Google Home
- Home Assistant
- ระบบ IoT Gateway

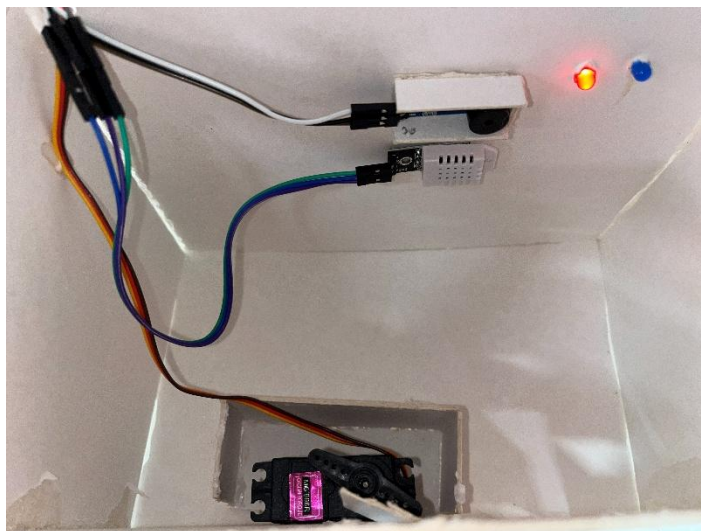
เพื่อให้สามารถควบคุมผ่านคำสั่งเสียง (Voice Control) และทำงานร่วมกับอุปกรณ์อัจฉริยะอื่นภายในบ้าน

## การทดลองลงมือทำจริง

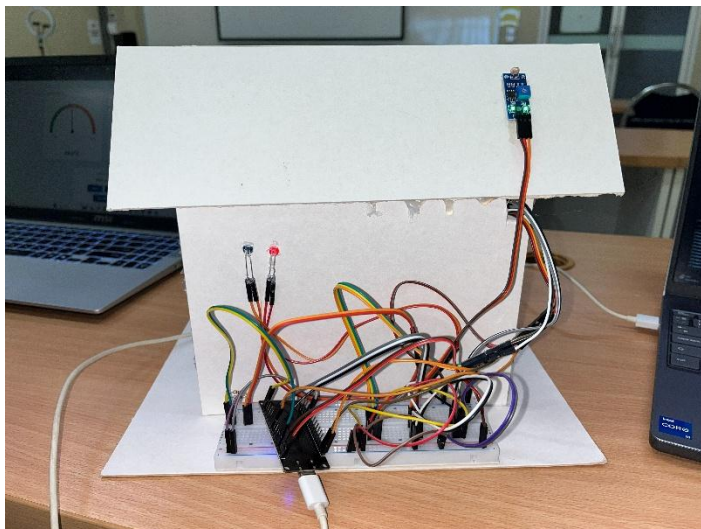
ภาพรวมของชิ้นงาน



: ภายใน



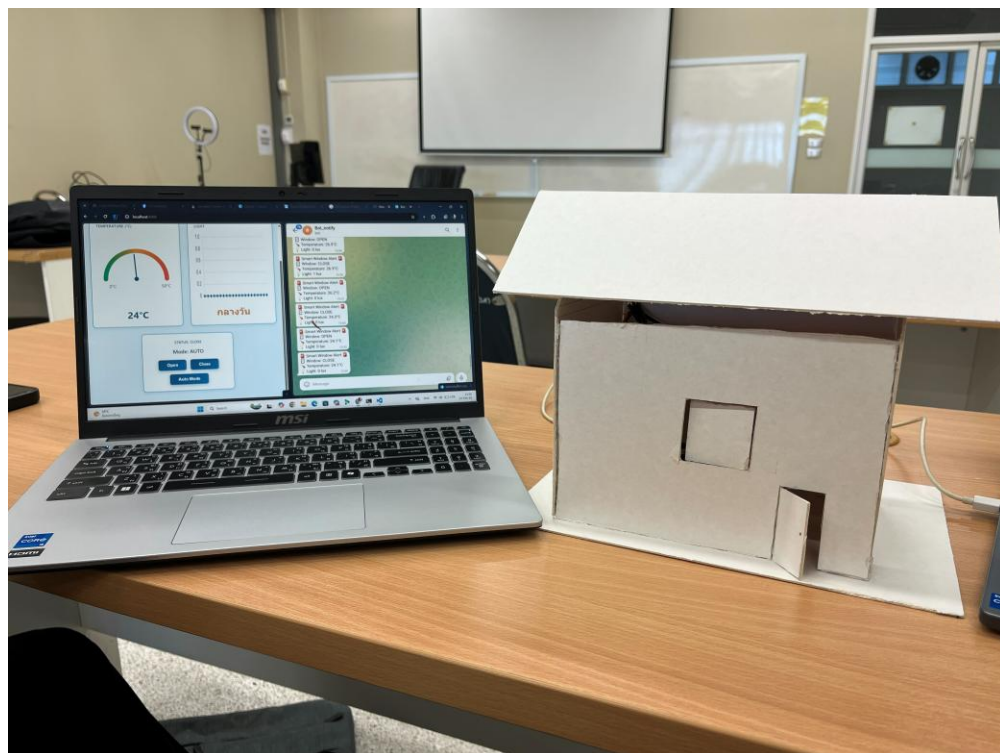
: ด้านหลัง



: MODE Auto / เมื่อแสงสว่างกระทบที่ตัวเซนเซอร์ LDR หน้าต่างจะเปิดออก

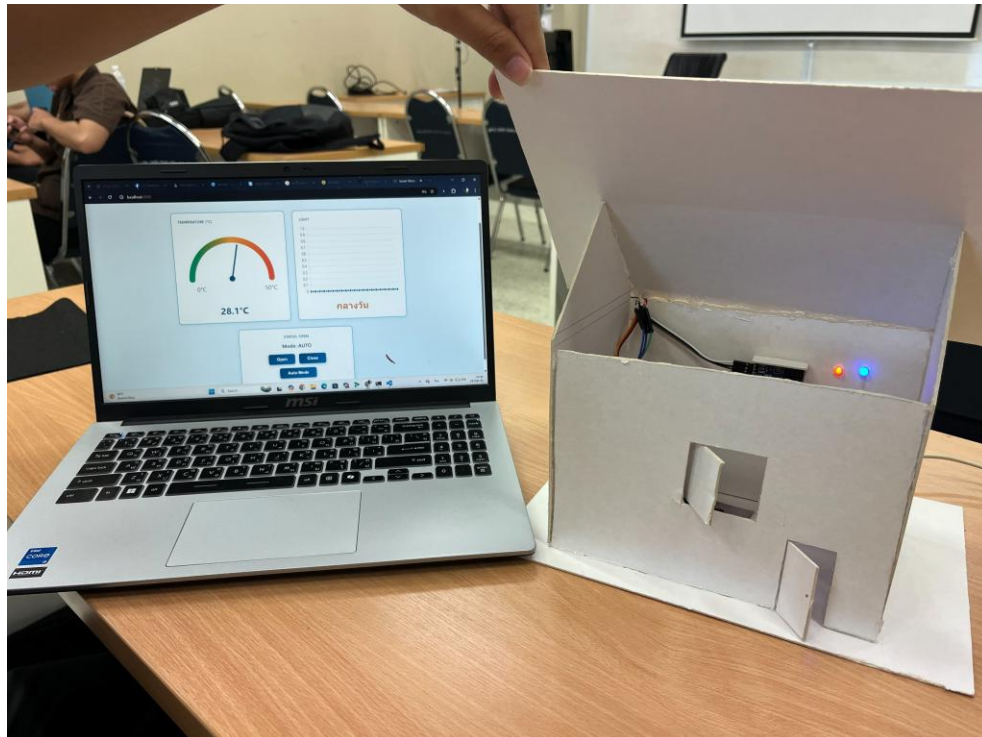


: MODE Auto / เมื่อไม่มีแสงสว่างกระทบที่ตัวเซนเซอร์ LDR หน้าต่างจะปิด

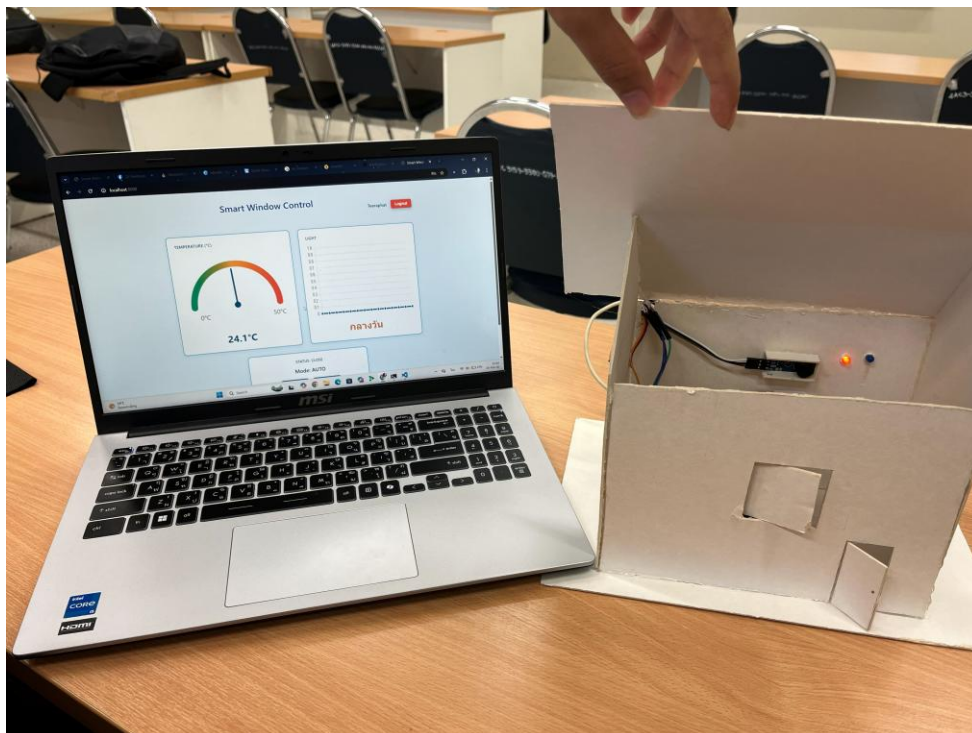




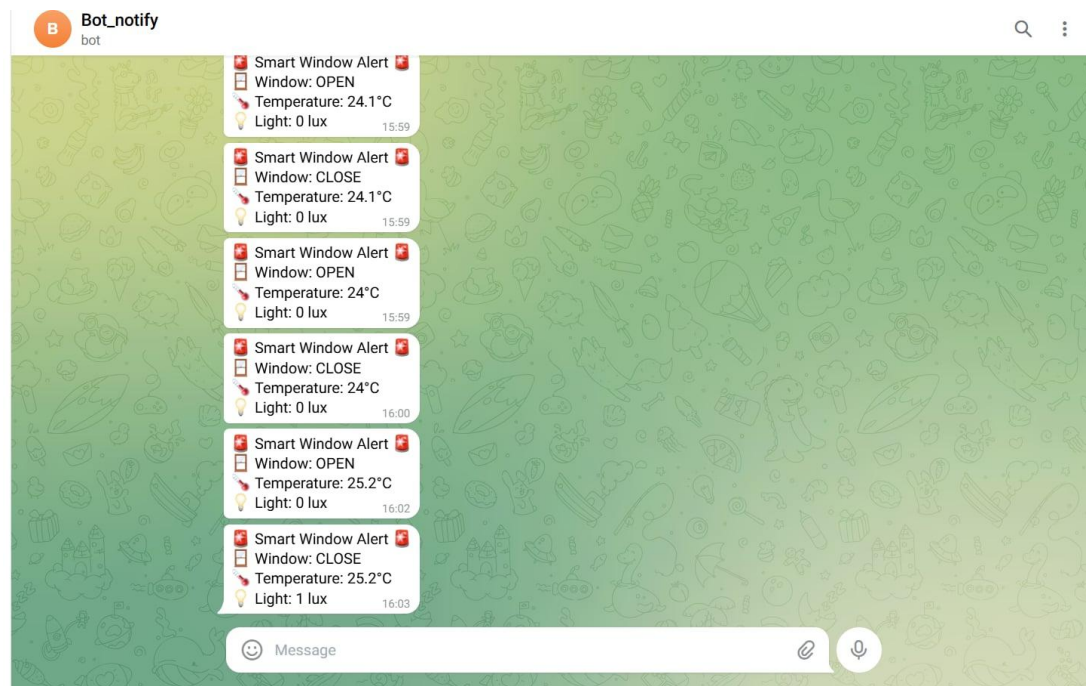
: เมื่ออุณหภูมิในห้องสูงเกิน 25 องศา หน้าต่างจะเปิดออก



: เมื่ออุณหภูมิในห้องต่ำกว่า 25 องศา หน้าต่างจะปิด



: แจ้งเตือนสถานะต่างๆ ผ่าน Telegram เช่น การ เปิด-ปิด ของหน้าต่าง , อุณหภูมิภายในห้อง , กลางวันหรือกลางคืน



## บทสรุปการดำเนินงาน

จากการทดลองจะเห็นได้ว่าเมื่อ อุณหภูมิห้องมากกว่าหรือเท่ากับ 25 องศาให้หน้าต่างทำการเปิดโดยอัตโนมัติและปิดเมื่ออุณหภูมิในห้องน้อยกว่า 25 องศา และ วัดค่าแสงว่าเป็นกลางวันหรือกลางคืน ถ้าเป็นกลางคืนให้ปิดหน้าต่างอัตโนมัติโดยใช้เซ็นเซอร์ LDR Modul ที่เป็นแบบ Digital ที่รับค่าเป็น 0 กับ 1 ในกราฟจึงแสดงสถานะแค่ 0 กับ 4095 เท่านั้น และยังสามารถ ควบคุมการเปิด-ปิดหน้าต่างผ่านหน้าเว็บไซต์ได้อีกด้วยและตรงตามขอบเขตที่ตั้งไว้

## ผลลัพธ์การทดลอง

ผลจากการทดลองจะเห็นได้ว่าเวลามีหน้าต่างจะการปิดอัตโนมัติ

ผลจากการทดลองจะเห็นได้ว่าเวลาสว่างและอุณหภูมิมากกว่า 25 องศาหน้าต่างจะเปิดออก

ผลจากการทดลองจะเห็นได้ว่าเวลาสว่างและอุณหภูมิน้อยกว่า 25 องศาหน้าต่างจะปิด

ผลจากการทดลองจะเห็นได้ว่ากดปิด หน้าต่างก็จะปิดไม่สนว่าอุณหภูมิเท่าไรและสว่างหรือไม่

ผลจากการทดลองจะเห็นได้ว่ากดเปิด หน้าต่างก็จะเปิดไม่สนว่าอุณหภูมิเท่าไรและสว่างหรือไม่

## บรรณานุกรม

- [1] B. Soewito, A. R. Nugroho, and S. Widodo, “Websocket to Support Real Time Smart Home Applications,” in Proc. \*Procedia Computer Science\*, vol. 157, pp. 560–566, 2019, doi:10.1016/j.procs.2019.09.081.
- [2] R. K. Kodali, V. Jain, S. Bose, and L. Boppana, “IoT based smart security and home automation system,” in \*2016 International Conference on Computing, Communication and Automation (ICCCA)\*, Greater Noida, India, 2016, pp. 1286–1289, doi:10.1109/CCAA.2016.7813916.
- [3] C. Buratti, A. Conti, D. Dardari, and R. Verdone, “An Overview on Wireless Sensor Networks Technology and Evolution,” \*Sensors (Basel)\*, vol. 9, no. 9, pp. 6869–6896, 2009, doi:10.3390/s90906869.
- [4] D. Taniar, J. Barthelemy, and L. Cheng, “Research on Real-time Data Transmission between IoT Gateway and Cloud Platform based on Two-way Communication Technology,” \*International Journal of Smartcare Home\*, vol. 1, no. 1, pp. 61–74, 2021.