

# Smart Room Security System Using IoT and Web-Based Monitoring

Mr. Theerut Mueangsa  
B.Eng. Computer Engineering  
Khon Kaen, Thailand

Mr. Akkadech Jaengpromma  
B.Eng. Computer Engineering  
Khon Kaen, Thailand

Mr. Kaisak Kabklon  
B.Eng. Computer Engineering  
Khon Kaen, Thailand

บทความนี้นำเสนองานการพัฒนาระบบรักษาความปลอดภัยห้องอัจฉริยะโดยใช้เทคโนโลยี Internet of Things (IoT) เพื่อเฝ้าระวังเหตุการณ์ผิดปกติภายในห้อง ระบบประกอบด้วยเซนเซอร์ตรวจจับควัน MQ-2 เซนเซอร์วัดอุณหภูมิและความชื้น DHT11 และเซนเซอร์อัลตราโซนิก HC-SR04 สำหรับตรวจสอบสภาพแวดล้อม ปิด-ปิดประตู ข้อมูลจากเซนเซอร์ถูกส่งผ่านเครือข่าย WiFi ไปยัง Web Server ซึ่งพัฒนาด้วยภาษา PHP และจัดเก็บลงฐานข้อมูล MariaDB ระบบสามารถแสดงผลแบบ Real-time ผ่าน Web Dashboard พัฒนาด้วยภาษา PHP ที่แสดงผลลัพธ์ที่ได้จากการวิเคราะห์ ผลการทดลองพบว่าระบบสามารถตรวจจับควันและการเปลี่ยนแปลงสถานะประดิษฐ์ได้อย่างแม่นยำและตอบสนองได้รวดเร็ว หากมีการพยายามเข้ามายังห้องจะได้รับ警報ทันที

IoT, Smart Security, Web Monitoring, MariaDB, PHP

## I. INTRODUCTION (HEADING I)

ในปัจจุบันเทคโนโลยี Internet of Things (IoT) ได้เข้ามามีบทบาทสำคัญในการพัฒนาระบบอัตโนมัติและระบบบันทึกความปลอดภัย ระบบแบบดั้งเดิมมักไม่เพียงการป้องกันการบุกรุก แต่ยังขาดการตรวจสอบสภาพแวดล้อม เช่น ควันไฟหรืออุณหภูมิที่สูงผิดปกติ

โครงการนี้จึงพัฒนาระบบ Smart Room Security ที่สามารถตรวจจับควัน ตรวจวัดอุณหภูมิและความชื้น ควบคุมการเปิด-ปิดประตู และแสดงผลผ่าน Web Application แบบเรียลไทม์ พร้อมจัดเก็บข้อมูลลงฐานข้อมูลเพื่อวิเคราะห์ย้อนหลัง

## II. SYSTEM REQUIREMENT

- INPUT (SENSORS): ต้องมีเซนเซอร์อย่างน้อย 2 ชนิด (กรณีกลุ่ม 3 คนต้องเพิ่มเป็น 3) ที่แตกต่างกัน (เช่น วัดอุณหภูมิ และ ตรวจจับการเคลื่อนไหว) เพื่อกำหนดข้อมูลสภาพแวดล้อม
- Output (Actuators): ต้องมีอุปกรณ์แสดงผลหรือทำงานเชิงกลอย่างน้อย 2 ชนิด (เช่น Relay ควบคุมไฟ, Servo Motor เปิด-ปิด, หรือ Buzzer แจ้งเตือน) (กรณีกลุ่ม 3 คน ต้องเพิ่มเป็น 3 ชนิด)
- Status LED : ต้องมีหลอดไฟ LED แสดงสถานการทำงานของระบบอย่างน้อย 2 ดวง (กรณีกลุ่ม 3 คนต้องเพิ่มเป็น 3 ดวง) เพื่อสื่อสารกับผู้ใช้ได้โดยไม่ต้องดูหน้าจอ เช่น:
  - ไฟกระพริบ: กำลังเริ่มต้น WiFi/MQTT
  - ไฟคำา: เริ่มต่อสำเร็จ / ระบบทำงานปกติ
  - ไฟเปลี่ยนสี/ดับ: เกิดข้อผิดพลาด (Error) หรือค่า Sensor เกินกำหนด
- Physical Model (แบบจำลอง): ต้องสร้างแบบจำลองสถานที่จริง (Diorama/Maket) ด้วยวัสดุ เช่น ก่อกรากด้วยหิน หรือ ปูกระเบื้องด้วยหิน ที่มีความจำลองที่สูง บน Breadboard โดยไม่มีการจำลอง
  - ตัวอย่าง: หากทำ Smart Home ต้องมีโมเดลบ้าน หากทำ Smart Farm ต้องมีแปลงผักจำลอง

## III. SYSTEM ARCHITECTURE

ระบบถูกออกแบบตามสถาปัตยกรรม IoT แบ่งออกเป็น 3 ชั้นหลัก ได้แก่

- Sensor Layer  
ประกอบด้วย MQ-2, DHT11 และ HC-SR04 ทำหน้าที่ตรวจดูข้อมูลสภาพแวดล้อม และสถานะประตู
- Application Layer  
ไมโครคอนโทรลเลอร์ (เช่น ESP8266/ESP32) รับค่าจากเซนเซอร์และส่งข้อมูลผ่าน WiFi ไปยัง Web Server โดยใช้ HTTP Request และมี Rate Limit ควบคุมจำนวนการส่ง Request ป้องกันการรุกราน กำกับดูแลผ่าน Telegram โฉนดแบบ

## 3. Database Layer

Web Server พัฒนาด้วยภาษา PHP รับข้อมูลและบันทึกลงฐานข้อมูล MariaDB พร้อมทั้งดึงข้อมูลมาแสดงผลบน Dashboard

ลำดับการทำงานของระบบ:

- เซนเซอร์
- ไมโครคอนโทรลเลอร์
- Web Server (PHP)
- MariaDB
- Web Dashboard

## IV. HARDWARE COMPONENTS

### 1. MQ-2 Gas Sensor

ใช้ตรวจจับควันและกําชีไฟ เมื่อค่าควันเกินเกณฑ์ที่กำหนด ระบบจะส่งให้ Buzzer ทำงานและแสดงสถานะ Fire Alert

### 2. DHT11 Sensor

ใช้วัดอุณหภูมิและความชื้นภายในห้อง เพื่อตรวจสอบสภาพแวดล้อมและช่วยยืนยันเหตุการณ์ไฟไหม้

### 3. HC-SR04 Ultrasonic Sensor

ใช้วัดระยะห่างของประตูเพื่อรับสถานะเปิด-ปิด

### 4. Servo Motor 180°

ใช้ควบคุมการปิด-ปิดประตู

### 5. IRF520N MOSFET Module

ใช้ขับโหลดที่ต้องการกระแสสูง เช่น มอเตอร์

### 6. LED และ Buzzer

LED ใช้แสดงสถานะ WiFi และสถานะประตู

Buzzer ใช้แจ้งเตือนเมื่อเกิดเหตุคุกคามหรือไฟไหม้

## V. WEB APPLICATION AND DATABASE

ระบบเว็บพัฒนาด้วยภาษา PHP ทำงานร่วมกับ HTML, CSS และ JavaScript เพื่อสร้าง Dashboard และแสดงผลแบบ Real-time

หน้าที่ของ PHP ได้แก่:

- รับข้อมูลจากอุปกรณ์ IoT
- ประมวลผลเมื่อไฟ เก็บ Fire Status
- บันทึกข้อมูลลงฐานข้อมูล MariaDB
- ดึงข้อมูลมาแสดงผล

โครงสร้างฐานข้อมูล (Table: sensor\_data)

- id – Primary Key
- smoke – ค่าควัน (PPM)
- temperature – อุณหภูมิ (°C)
- humidity – ความชื้น (%)
- door\_status – สถานะประตู
- created\_at – เวลาบันทึกข้อมูล

ข้อมูลถูกส่งจากอุปกรณ์ทุก ๆ 5 วินาที และบันทึกลงฐานข้อมูลเพื่อแสดงผลย้อนหลัง ผ่านกราฟอุณหภูมิและความชื้น

## VI. EXPERIMENTAL RESULTS

ผลการทดสอบระบบภายใต้สภาพแวดล้อมจริงพบว่า

- ระบบสามารถตรวจจับควันและสั่ง Buzzer ทำงานภายใน 2-3 วินาที
- การเปิดประตูสามารถตรวจจับได้อย่างแม่นยำจากการเปลี่ยนแปลงระยะของ Ultrasonic Sensor
- ข้อมูลลูกบ้านที่เก็บ MariaDB อย่างต่อเนื่อง
- Dashboard สามารถแสดงผลข้อมูลแบบ Real-time ได้ถูกต้อง

ระบบมีความเสถียรและเหมาะสมสำหรับการใช้งานจริงในระดับบ้านอัจฉริยะ

## VII. CONCLUSION

ระบบ Smart Room Security ที่พัฒนาขึ้นสามารถทำงานได้อย่างมีประสิทธิภาพ โดยบูรณาการเทคโนโลยี IoT เข้ากับ Web Application และฐานข้อมูล ทำให้สามารถเฝ้าระวังเหตุการณ์ได้ปกติแบบต่อเนื่องและจัดเก็บข้อมูลเพื่อการวิเคราะห์ในอนาคตแนวทางพัฒนาต่อไป ได้แก่

- เชื่อมต่อ Cloud Platform
- เพิ่ม Mobile Application
- เพิ่มระบบแจ้งเตือนผ่าน LINE หรือ Email
- วิเคราะห์ข้อมูลด้วย AI

Github : [https://github.com/akkadechja/project\\_room](https://github.com/akkadechja/project_room)

Wokwi : <https://wokwi.com/projects/456824251801729025>

## REFERENCES

- [1] A. Bahga and V. Madisetti, Internet of Things: A Hands-on Approach, 2015.
- [2] Hanwei Electronics, "MQ-2 Gas Sensor Datasheet," 2018.
- [3] MariaDB Foundation, "MariaDB Documentation," 2023.
- [4] PHP Group, "PHP Manual," 2023.

## # 🛡 Smart Room Security System

## TH ภาษาไทย

#### 🚨 รายละเอียดโครงงาน

Smart Room Security System เป็นระบบรักษาความปลอดภัยที่อัจฉริยะที่พัฒนาโดยใช้เทคโนโลยี Internet of Things (IoT) ร่วมกับ Web Application และฐานข้อมูล MariaDB เพื่อเฝ้าระวังเหตุการณ์ผิดปกติแบบ Real-time

ระบบสามารถตรวจจับควัน ตรวจสอบอุณหภูมิและความชื้น ตรวจสอบสถานะประตู และแสดงผลผ่าน Web Dashboard พร้อมบันทึกข้อมูลข้อหนหลัง

---

#### 🔥 ความสามารถของระบบ (Features)

- ตรวจจับควันด้วย MQ-2
- วัดอุณหภูมิและความชื้นด้วย DHT11
- ตรวจสอบสถานะประตูด้วย Ultrasonic Sensor (HC-SR04)
- ควบคุมประตูด้วย Servo Motor 180°
- แจ้งเตือนด้วย Buzzer
- LED แสดงสถานะ WiFi และประตู
- แสดงผลผ่าน Web Dashboard
- บันทึกข้อมูลลงฐานข้อมูล MariaDB
- ปุ่มเปิดปิดประตูอุปกรณ์

---

#### 🏢 โครงสร้างระบบ (System Architecture)

Sensors → ESP32 → Web Server (PHP) → MariaDB → Web Dashboard

---

#### 🛡 อุปกรณ์ที่ใช้ (Hardware)

- MQ-2 Gas Sensor
- DHT11 Temperature & Humidity Sensor
- HC-SR04 Ultrasonic Sensor
- Servo Motor 180°
- IRF520N MOSFET Module
- ESP32
- LED Indicators
- Buzzer
- Emergency Button

#### 💻 เทคโนโลยีที่ใช้ (Software Stack)

- PHP
- MariaDB
- HTML / CSS / JavaScript
- XAMPP (Apache Server)
- Arduino IDE

---

#### 📁 โครงสร้างฐานข้อมูล (Database Structure) \*\*\*

\*\*Table 1: `sensor\_logs`\*\*

Field	Type	Description
id	INT (PK)	รหัส Auto Increment
temperature	FLOAT(5, 2)	ค่าอุณหภูมิ
humidity	FLOAT(5, 2)	ค่าความชื้นสัมพัทธ์
smoke_level	INT	ระดับควัน
created_at	TIMESTAMP	เวลาที่บันทึกข้อมูล

\*\*Table 2: `device\_status`\*\*

Field	Type	Description
id	INT (PK)	รหัสอุปกรณ์
device_name	VARCHAR(50)	ชื่ออุปกรณ์ (door, buzzer, fan)
status	TINYINT(1)	0 = Off/Closed, 1 = On/Open

---

#### 🚨 วิธีติดตั้งระบบ

##### 1 ติดตั้ง Web Server

1. ติดตั้ง XAMPP
2. เปิด Apache และ MariaDB
3. สร้างฐานข้อมูลชื่อ `iot\_security`
4. สร้างตาราง `sensor\_logs` และ `device\_status`
5. นำไฟล์รีบบ์ไปไว้ในโฟเดอร์ `htdocs/`
6. ตั้งค่า WiFi และ URL ของเซิร์ฟเวอร์ในไกด์ ESP
7. อัปโหลดไกด์ (Firmware) ลงบอร์ด

##### 2 นำไฟล์รีบบ์ไปไว้ใน

```

htdocs/

```

เปิดผ่าน Browser:

```  
<http://localhost/iot-miniproject>  
```

- Data logging with MariaDB
- Emergency door release button

---

#### ##### 3 อปปิൾด์ໄດ້ຄົງ ESP

- ຕົກ Arduino IDE
- ໄສ WiFi SSID ແລະ Password
- ກຳນົດ URL ຂອງ Web Server
- Upload ລົມອົບຮັດ

---

#### ##### 🛡️ ນັກການທຳຈານ

1. ESP ອ່ານຄ່າຈາກເຊີ້ນເຊື່ອ
2. ສັງເກືອບຜ່ານ WiFi ໄປສໍ PHP
3. PHP ບັນທຶກຂໍ້ມູຄລົງ MariaDB
4. Dashboard ແສດພົບແບບ Real-time
5. PHP ສັງສານຂອງ Outputs ໄປສໍ ESP

---

#### ### 🖊️ ພັດທະນາ

- ດຽວຈັບຄວນໄດ້ກໍາຍິນ 2–3 ວິນາທີ
- ດຽວຈັບການປິດປະຫຼາດໄດ້ແມ່ນທຳ
- ບັນທຶກຂໍ້ມູຄດ້ວຍເນື່ອງຈຸກ 5 ວິນາທີ

---

#### ## GB English Version

#### ### 🚨 Project Overview

Smart Room Security System is an IoT-based smart room monitoring system integrated with a Web Application and MariaDB database for real-time monitoring and data logging.

The system detects smoke, measures temperature and humidity, monitors door status, and displays data through a web dashboard.

---

#### ### 🔥 Features

- Smoke detection using MQ-2
- Temperature & humidity monitoring using DHT11
- Door status detection using Ultrasonic Sensor (HC-SR04)
- Door control via 180° Servo Motor
- Buzzer alert system
- WiFi and door status LED indicators
- Web-based real-time dashboard

#### #### 🏢 System Architecture

Sensors → ESP8266/ESP32 → Web Server (PHP) → MariaDB → Web Dashboard

---

#### #### 🛡️ Hardware Components

- MQ-2 Gas Sensor
- DHT11 Sensor
- HC-SR04 Ultrasonic Sensor
- 180° Servo Motor
- IRF520N MOSFET Module
- ESP8266 / ESP32
- LEDs
- Buzzer
- Emergency Button

---

#### #### 📜 Software Stack

- PHP
- MariaDB
- HTML / CSS / JavaScript
- Apache (XAMPP)
- Arduino IDE

---

#### #### 🗂️ Database Structure

\*\*Table 1: `sensor\_logs`\*\*

Field	Type	Description
id	INT (PK)	Auto Increment ID
temperature	FLOAT(5, 2)	Temperature value
humidity	FLOAT(5, 2)	Relative humidity value
smoke_level	INT	Smoke level
created_at	TIMESTAMP	Record timestamp

\*\*Table 2: `device\_status`\*\*

Field	Type	Description
id	INT (PK)	Device ID
device_name	VARCHAR(50)	Device name (door, buzzer, fan)
status	TINYINT(1)	0 = Off/Close, 1 = On/Open

---

#### #### 🚀 Installation

1. Install XAMPP
  2. Start Apache & MariaDB
  3. Create database 'iot\_security'
  4. Create table 'sensor\_logs' and 'device\_status'
  5. Upload web files to 'htdocs/'
  6. Configure WiFi and server URL in ESP code
  7. Upload firmware
- 

#### #### 🌐 How It Works

1. ESP reads sensor data
2. Sends data via WiFi to PHP server
3. PHP stores data in MariaDB
4. Dashboard displays real-time information
5. PHP send outputs status to ESP

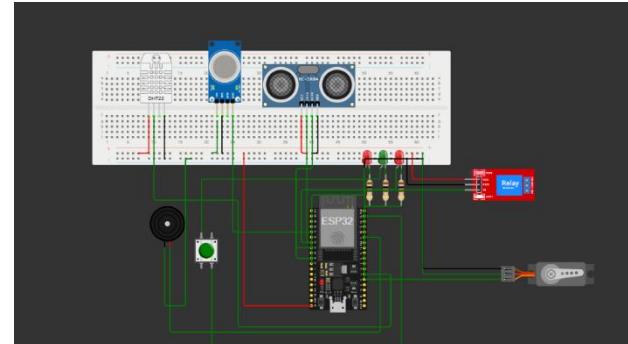
---

#### #### 📈 Testing Results

- Smoke detected within 2–3 seconds
- Accurate door detection
- Continuous data logging every 5 seconds



1	Time	Temp	Hum	Gas	Door	Alarm	L
2	2/24/2026 8:08:3	24.4	47	3628	0	1	
3	2/24/2026 8:10:0	32.5	47	3774	0	1	
4	2/24/2026 8:10:2	32.5	47	3757	0	1	
5	2/24/2026 8:37:5	28.9	47	3628	0	1	
6	2/24/2026 9:17:2	32.5	47	3628	0	1	
7	2/24/2026 9:17:5	32.5	47	1019	1	0	
8	2/24/2026 9:18:4	32.5	47	1019	1	0	
9	2/24/2026 9:19:2	18.1	47	1019	0	0	
10	2/24/2026 9:20:0	18.1	47	1019	0	0	
11	2/24/2026 9:20:5	18.1	47	1019	0	0	
12	2/24/2026 9:21:3	18.1	47	1019	0	0	
13	2/24/2026 9:22:1	18.1	47	1019	0	0	
14	2/24/2026 9:22:5	18.1	47	1019	0	0	
15	2/24/2026 9:23:3	18.1	47	1019	0	0	
16	2/24/2026 9:24:1	18.1	47	1019	0	0	
17	2/24/2026 9:24:5	18.1	47	1019	0	0	
18	2/24/2026 9:25:3	18.1	47	1019	0	0	
19	2/24/2026 9:26:1	18.1	47	1019	0	0	
20	2/24/2026 9:26:5	18.1	47	1019	0	0	



### Environment Monitoring

