

ระบบเครือข่ายและโปรโตคอล



IP Address

ที่อยู่เชิงตรรกะ เปลี่ยนแปลงได้
เมื่อย้ายเครือข่ายหรือเชื่อมต่อ
อินเทอร์เน็ตใหม่ ใช้ส่งข้อมูล
ข้ามเครือข่ายและอินเทอร์เน็ต

เครือข่ายไอพี

IPv4 • IPv6 • WAN • LAN



MAC Address

ที่อยู่ทางกายภาพ กำหนดมาด้วย
อุปกรณ์จากโรงงาน ใช้ระบุ
อุปกรณ์และสื่อสารภายใน
เครือข่ายท้องถิ่น



Subnet Mask

กำหนดขอบเขตของเครือข่าย



DNS Server

แปลงชื่อโดเมนเป็นหมายเลขไอพี



โดยครูนาวิทย์ สี่หาราช

วิทยาลัยเทคนิคกาญจนาภิเษกอุบลราชธานี

ระบบปฏิบัติการหลัก



Linux



Windows Server



Unix

เทคโนโลยีจำลองระบบ



VMware vSphere



Microsoft Hyper-V



KVM

การเลือกใช้ขึ้นอยู่กับภาระงาน ความปลอดภัย งบประมาณ และการบริหารจัดการ



ประเภทของเซิร์ฟเวอร์และหน้าที่

1



Web Server

ให้บริการเว็บไซต์

2



Database Server

จัดเก็บและจัดการฐานข้อมูล

3



Mail Server

รับและส่งอีเมล

4



DNS Server

แปลงชื่อโดเมนเป็น
หมายเลขไอพี

5



Virtualization Server

สร้างเครื่องเสมือน

6



DHCP Server

แจกหมายเลขไอพีอัตโนมัติ

7



File Server

จัดเก็บและแบ่งปันไฟล์

8



Application Server

ประมวลผลแอปพลิเคชัน

9



Backup Server

สำรองและกู้คืนข้อมูล

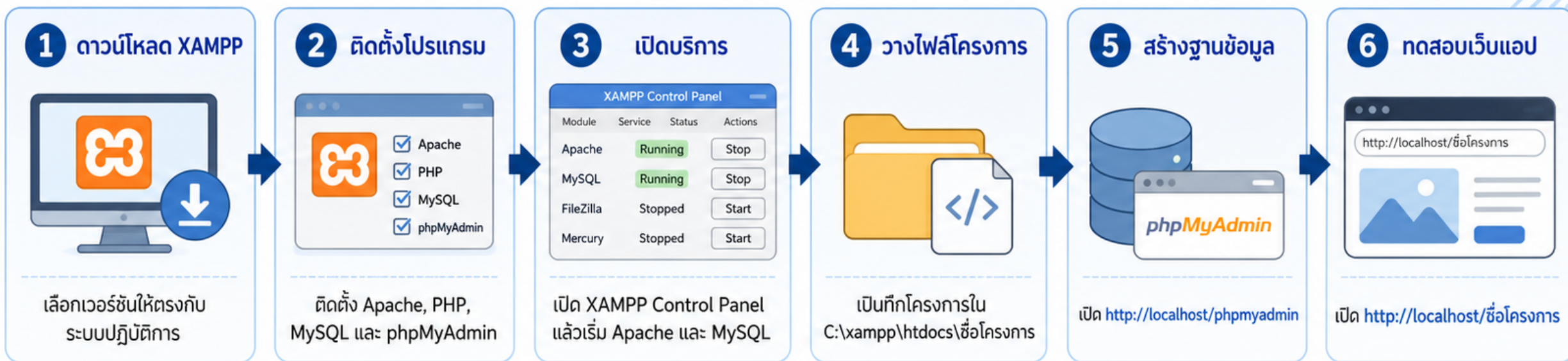


โดยครูณวิทย์ สี่หาราช

วิทยาลัยเทคนิคกาญจนาภิเษกอุดรธานี

การสร้างเว็บแอปด้วย XAMPP บน localhost

3 / 11



ลำดับการทำงาน



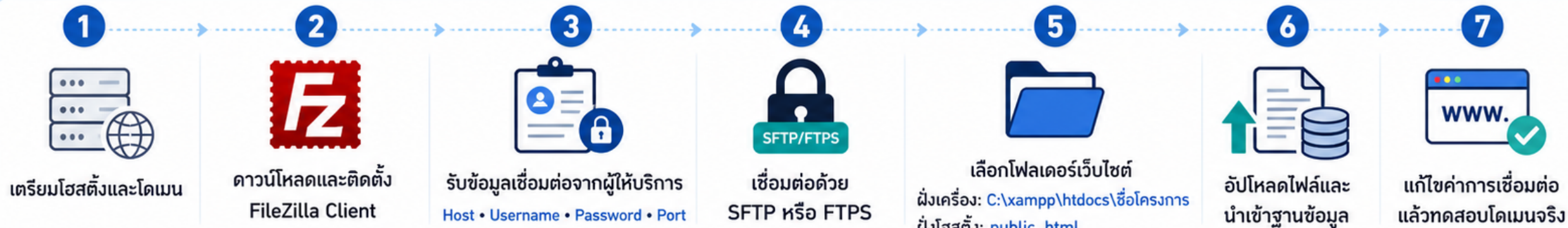
ตรวจสอบพอร์ต **80** และ **3306** เมื่อบริการไม่เริ่มทำงาน



โดยครูนาวิทย์ สีหาราช
วิทยาลัยเทคนิคกาญจนาภิเษกอุดรธานี

อัปโหลดเว็บแอปจาก localhost สู่อโฮสตั้งด้วย FileZilla Client

4 / 11



การตั้งค่าการเชื่อมต่อใน FileZilla

โปรโตคอล (Protocol):

SFTP - SSH File Transfer Protocol

โฮสต์ (Host):

your-domain.com

พอร์ต (Port): 22

รูปแบบการเข้าสู่ระบบ (Logon Type):

ปกติ (Normal)

ชื่อผู้ใช้ (User):

your-username

รหัสผ่าน (Password):

••••••••

เชื่อมต่อด่วน (Quickconnect)

หลักการทำงาน

Local site → Secure transfer → Remote site

ฝั่งเครื่อง (Local site)



C:\xampp\htdocs\ชื่อโครงการ

ฝั่งโฮสตั้ง (Remote site)



public_html

SFTP / FTPS

เข้ารหัสข้อมูลตลอดการรับส่ง

ปลอดภัย มั่นใจ ไร้กังวล



ตรวจสอบสิทธิ์ไฟล์ HTTPS และการป้องกันข้อมูลลับ



ตั้งค่าสิทธิ์ไฟล์ให้เหมาะสม
ไฟล์: 644 | โฟลเดอร์: 755



เปิดใช้งาน HTTPS (SSL/TLS)
ใช้ใบรับรอง SSL ที่ถูกต้อง



ป้องกันข้อมูลลับ
ไม่เก็บรหัสผ่านในโค้ด
ใช้ไฟล์ .env หรือ config แยก



สำรองข้อมูลเป็นประจำ
ไฟล์และฐานข้อมูล



อัปเดตระบบและปลั๊กอินเสมอ
ลดความเสี่ยงจากช่องโหว่



โดยครูณาวิทย์ สี่หาราช



วิทยาลัยเทคนิคกาญจนาภิเษกอุดรธานี

การใช้งาน ESP32 เบื้องต้น

1 เตรียมอุปกรณ์

ESP32 • สาย USB •
คอมพิวเตอร์ • LED



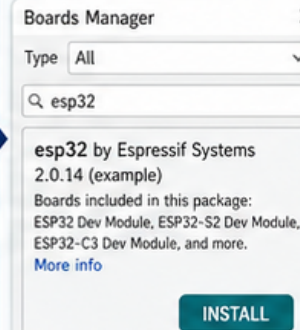
2 ติดตั้ง Arduino IDE



Arduino IDE

DOWNLOAD

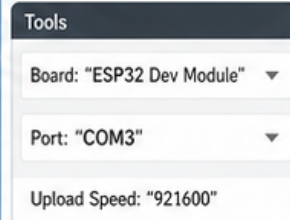
3 เพิ่ม ESP32 Board Manager



4 เลือกบอร์ดและพอร์ต

Tools → Board → ESP32

Tools → Port → COM



5 เขียนโปรแกรมทดสอบ



6 ตรวจสอบและอัปโหลด

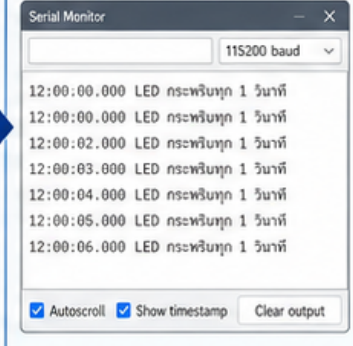
Verify → Upload



Sketch uses 286632 bytes (21%) of program storage space.
Global variables use 10952 bytes (3%) of dynamic memory.
Done uploading.

7 เปิด Serial Monitor

ตั้งค่า 115200 baud

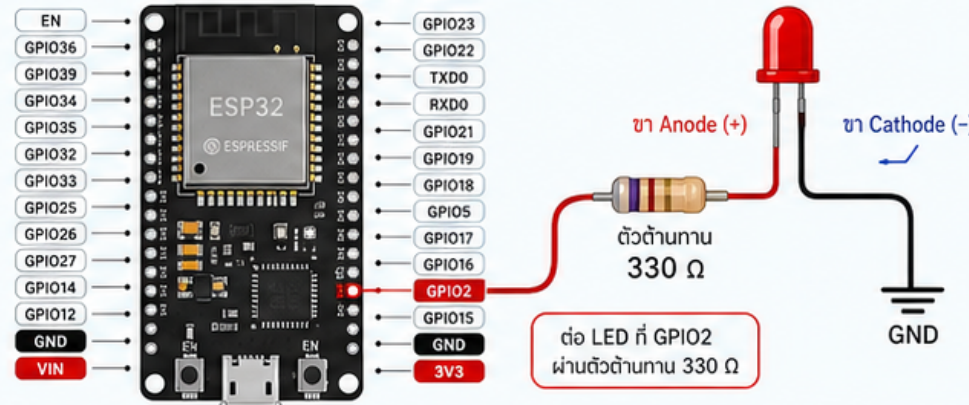


ตัวอย่างโค้ดทดสอบ LED

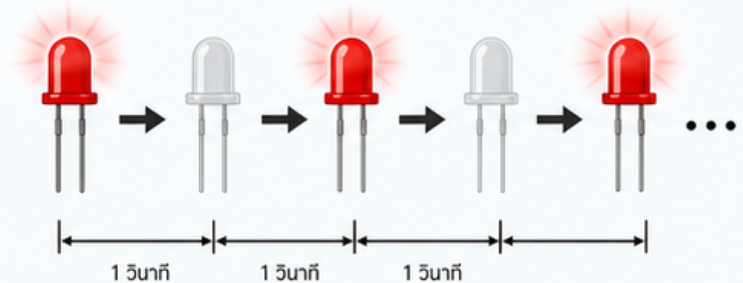
```

1 void setup() {
2   pinMode(2, OUTPUT);
3 }
4
5 void loop() {
6   digitalWrite(2, HIGH);
7   delay(1000);
8   digitalWrite(2, LOW);
9   delay(1000);
10 }
  
```

การต่อวงจรทดสอบ LED



ผลลัพธ์: LED กระพริบทุก 1 วินาที



1ติดตั้ง Arduino IDE และแพ็คเกจ ESP32

2เลือกบอร์ด ESP32 และพอร์ต COM

3เรียกใช้ไลบรารี BluetoothSerial

4ตั้งชื่ออุปกรณ์

5อัปโหลดโปรแกรม

6จับคู่โทรศัพท์กับ ESP32

7ส่งข้อความและตรวจสอบผล



โค้ดหลัก

```
#include "BluetoothSerial.h"

BluetoothSerial SerialBT;

SerialBT.begin("ESP32_BT");

if (SerialBT.available())

    Serial.write(SerialBT.read());
```



ทดสอบส่งข้อมูลทั้งสองทิศทาง

ESP32 เชื่อมต่อ Wi-Fi และเว็บแอป

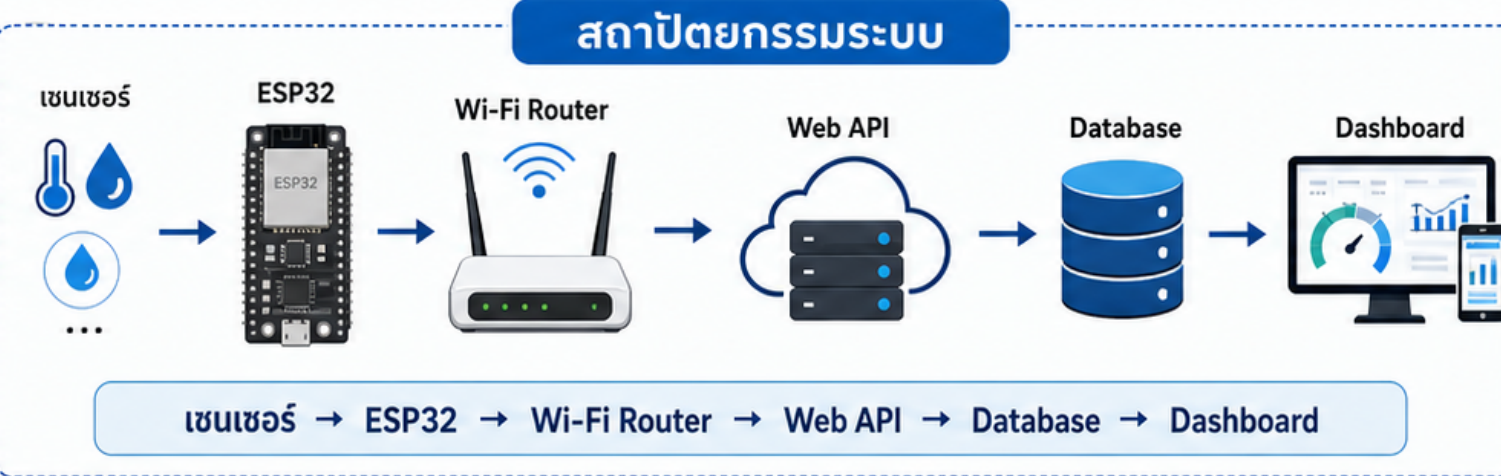


7 / 11

1 ออกแบบข้อมูล
และคำสั่งควบคุม

2 สร้างฐานข้อมูล
และ Web API

3 กำหนด SSID
และรหัสผ่าน



7 แสดงผล
บนเว็บแอป

8 ทดสอบและ
ตรวจสอบ
ข้อผิดพลาด

4 เชื่อมต่อ Wi-Fi

```
WiFi.begin(ssid, password);
```

5 อ่านค่าเซนเซอร์

6 ส่งข้อมูลด้วย
HTTP หรือ MQTT

ทดสอบ: สถานะ Wi-Fi • IP Address • API • ฐานข้อมูล • หน้าจอ

- ✓ ตรวจสอบการเชื่อมต่อ Wi-Fi และความเสถียรของสัญญาณ
- ✓ ตรวจสอบ IP Address ที่ได้รับจากเราเตอร์
- ✓ ตรวจสอบการเรียกใช้งาน Web API (Response ถูกต้อง)
- ✓ ตรวจสอบการบันทึกข้อมูลในฐานข้อมูล
- ✓ ตรวจสอบการแสดงผลบน Dashboard ให้ถูกต้อง

ตัวอย่างโค้ด (Arduino IDE)

```
1 #include <WiFi.h>
2 #include <HTTPClient.h>
3
4 const char* ssid = "YOUR_SSID";
5 const char* password = "YOUR_PASSWORD";
6 const char* serverUrl = "https://your-domain.com/api/data";
7
8 void setup() {
```

```
    Serial.begin(115200);
    WiFi.begin(ssid, password);
    Serial.print("Connecting Wi-Fi..");
    while (WiFi.status() != WL_CONNECTED) {
        delay(500); Serial.print(".");
    }
    Serial.println("\nWi-Fi connected");
    Serial.print("IP Address: "); Serial.println(WiFi.localIP());
}
```

```
void loop() {
    float temp = 25.6; // ตัวอย่างค่าจากเซนเซอร์
    HTTPClient http;
    http.begin(serverUrl);
    http.addHeader("Content-Type", "application/json");
    String payload = "{\"device\":\"esp32-01\",\"temp\":" + String(temp,1) + "}";
    int code = http.POST(payload);
    Serial.printf("HTTP POST code: %d\n", code);
    http.end();
    delay(5000); // ส่งทุก 5 วินาที
}
```



รักษาความปลอดภัยด้วย HTTPS การยืนยันตัวตน และการไม่ฝังรหัสผ่านในโค้ด



โดยครูนาวิทย์ สีหาราช
วิทยาลัยเทคนิคกาญจนาภิเษกอุดรธานี

1



กำหนดความต้องการและคำสั่งควบคุม

2



ให้ Codex สร้างโครงงาน Android

3



ออกแบบหน้าจอและสิทธิ์อินเทอร์เน็ต

4



เขียนโปรแกรม ESP32 Web Server

5



เชื่อม ESP32 และโทรศัพท์เข้าวง Wi-Fi เดียวกัน

6



เรียก REST API จากแอป

7



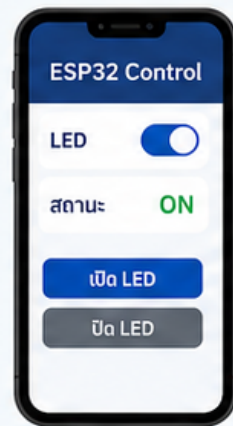
แสดงสถานะและจัดการข้อผิดพลาด

8



Build ติดตั้ง และทดสอบกับอุปกรณ์จริง

สถาปัตยกรรม



Android App



Wi-Fi Router



ESP32 Web Server

Android App ↔ Wi-Fi Router ↔ ESP32 Web Server

ตัวอย่างคำสั่ง

GET http://ESP32_IP/led/on



```
{
  "status": "ok",
  "led": "on"
}
```



ESP32 ตอบกลับ JSON และเปลี่ยนสถานะ LED

ทดสอบ:



IP Address



การตอบสนอง



เวลาแฝง



การเชื่อมต่อซ้ำ



โดยครูณาวิทย์ สีหาราช

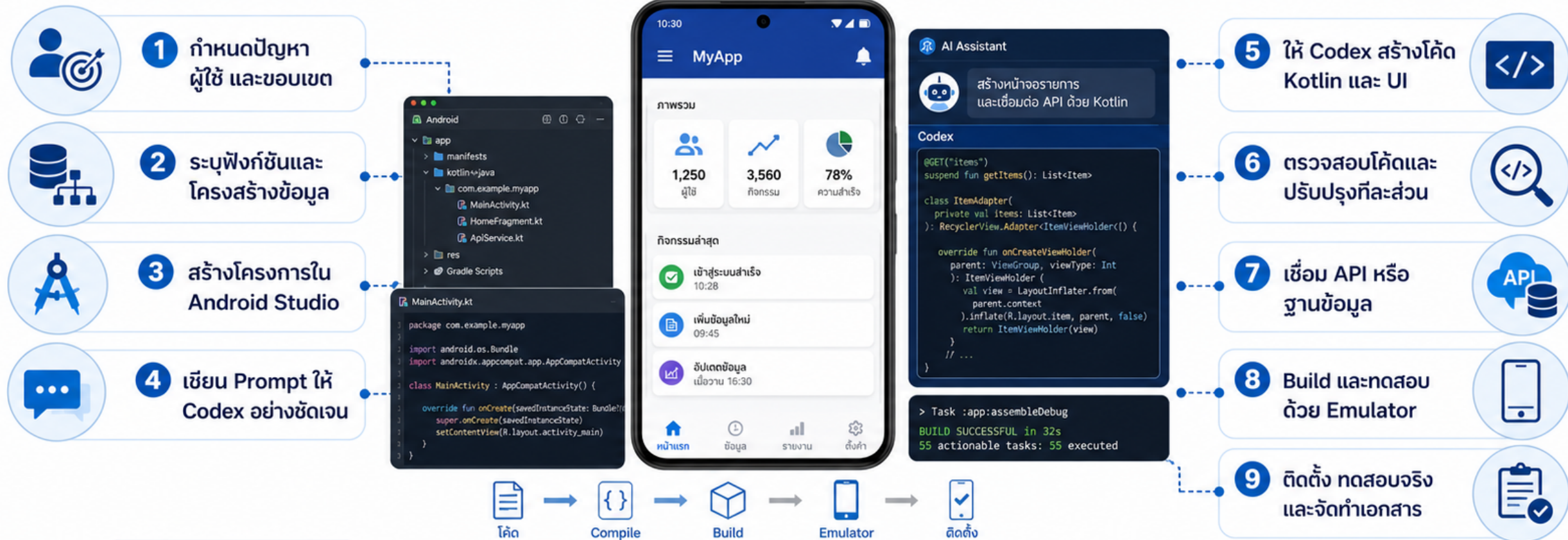


วิทยาลัยเทคนิคกาญจนาภิเษกอุดรธานี



การสร้างแอป Android ด้วย ChatGPT Codex

9 / 11



สูตร Prompt



บริบท + เป้าหมาย + เทคโนโลยี + ข้อจำกัด + ผลลัพธ์ที่ต้องการ

วงจรพัฒนา

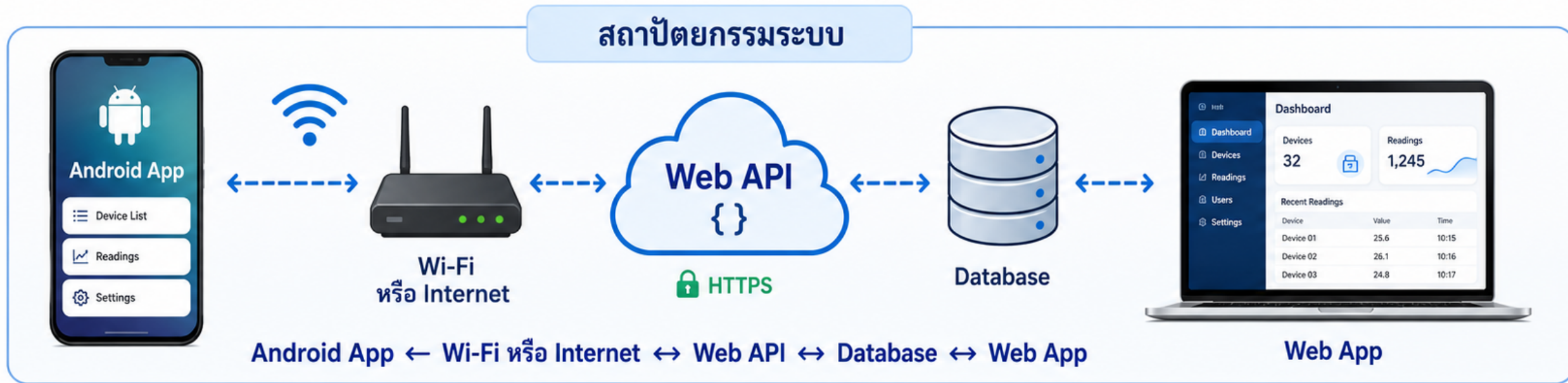


Plan → Generate → Review → Test → Improve

ตรวจสอบความปลอดภัย ประสิทธิภาพ การเข้าถึง และประสบการณ์ผู้ใช้



สร้างแอป Android เชื่อมต่อ Wi-Fi และเว็บแอปด้วย ChatGPT Codex 10 / 11



1



วิเคราะห์ผู้ใช้และกรณีใช้งาน

4



สร้างเว็บแอปสำหรับจัดการข้อมูล

7



จัดการ Login และ Token

2



ออกแบบฐานข้อมูลและ REST API

5



ให้ Codex สร้างแอป Android

8



ทดสอบข้อมูลทั้งสองทิศทาง

3



ให้ Codex สร้าง Backend

6



เชื่อม API ด้วย HTTPS

9



Build ติดตั้ง และติดตามข้อผิดพลาด

ตัวอย่าง API

GET /api/devices
POST /api/readings
PUT /api/device/{id}



ความปลอดภัย:

HTTPS • Authentication
• Validation • Access Control



โดยครูนาวิทย์ สี่หาราช



วิทยาลัยเทคนิคกาญจนาภิเษกอุดรธานี

1 กำหนดปัญหาและข้อกำหนดระบบ

2 ออกแบบอุปกรณ์และวงจร ESP32

3 ออกแบบฐานข้อมูลและ API

4 ให้ Codex ช่วยสร้าง Firmware

5 พัฒนา Backend และ Web App

6 พัฒนาแอป Android

7 เชื่อมระบบด้วย HTTP หรือ MQTT

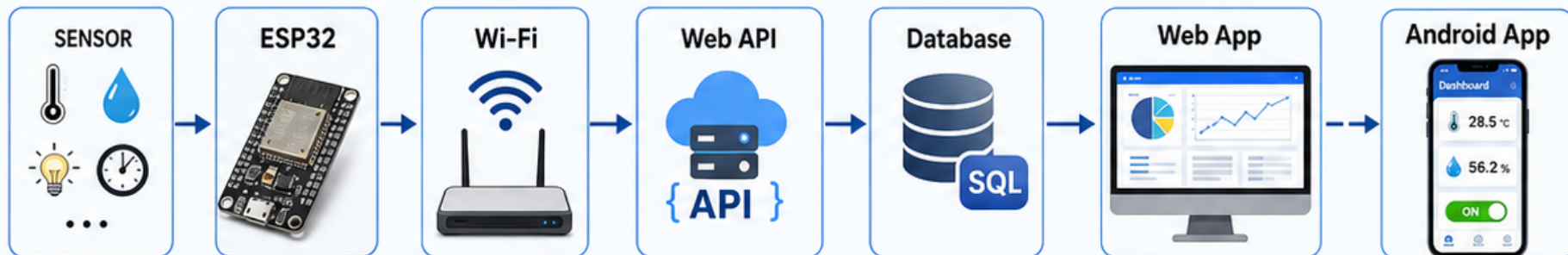
8 เพิ่มการยืนยันตัวตนและ HTTPS

9 ทดสอบแบบหน่วย แบบบูรณาการ และอุปกรณ์จริง

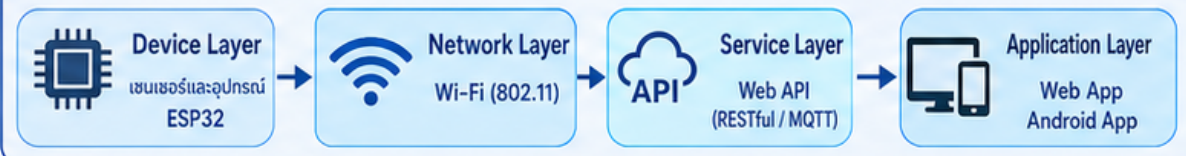
10 ติดตั้งใช้งาน ฝึกระวัง และปรับปรุง

สถาปัตยกรรมระบบครบวงจร

Sensor → ESP32 → Wi-Fi → Web API → Database → Web App และ Android App



ชั้นการสื่อสาร



โปรโตคอลที่ใช้



ความปลอดภัย (Security)



ผลลัพธ์: ตรวจสอบ แสดงผล ควบคุม แจ้งเตือน และบันทึกข้อมูล



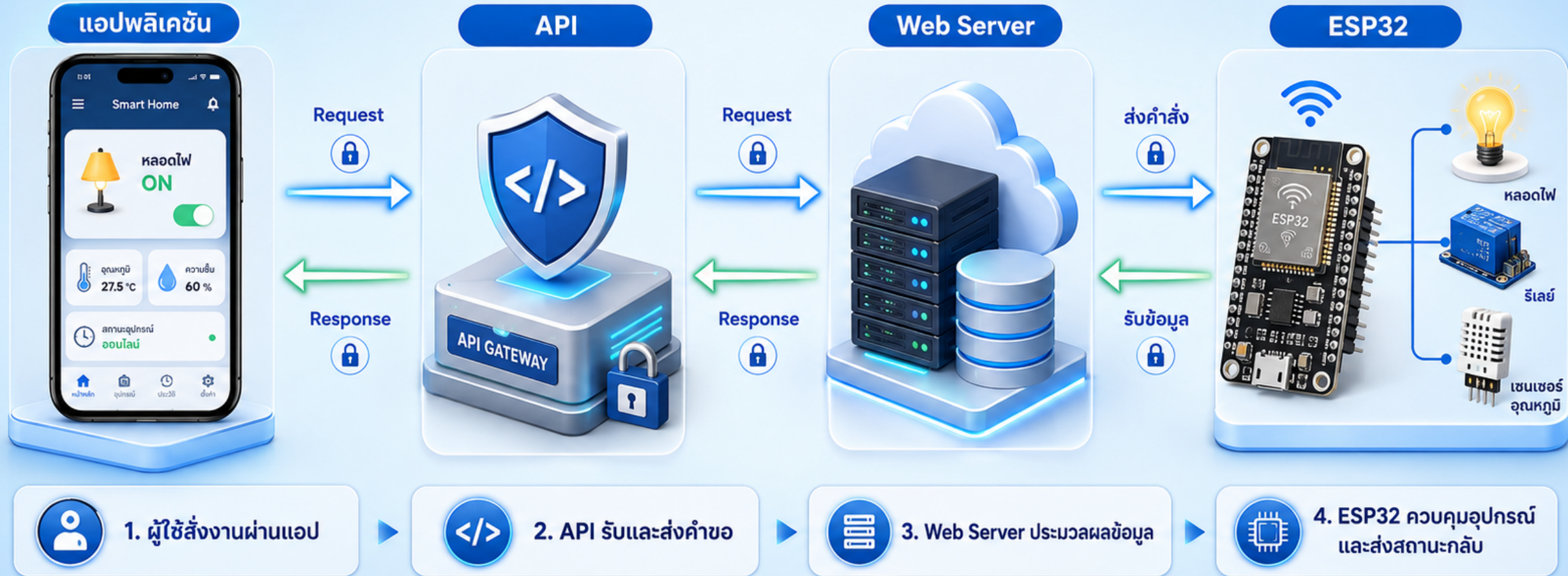
ทดสอบ: • ความถูกต้อง • • เวลาแฝง • • ความเสถียร • • ความปลอดภัย • • การกู้คืน

โดยครูณวิทย์ สิทราช

วิทยาลัยเทคนิคกาญจนาภิเษกอุดรธานี

API คืออะไร?

ตัวกลางที่ช่วยให้แอปพลิเคชัน อุปกรณ์ และเซิร์ฟเวอร์ สื่อสารแลกเปลี่ยนข้อมูลกัน



API คือสะพานเชื่อมระบบ **IoT** ให้ทำงานร่วมกันอย่างเป็นระบบและปลอดภัย

API คืออะไร?

ตัวกลางที่ช่วยให้โปรแกรมต่าง ๆ สื่อสารและแลกเปลี่ยนข้อมูลกัน



เปรียบเทียบพนักงานรับออเดอร์



GET อ่านข้อมูล



POST เพิ่มข้อมูล



PUT แก้ไขข้อมูล



DELETE ลบข้อมูล



ตัวอย่าง: แอปพยากรณ์อากาศเรียก API เพื่อขอข้อมูลอุณหภูมิ

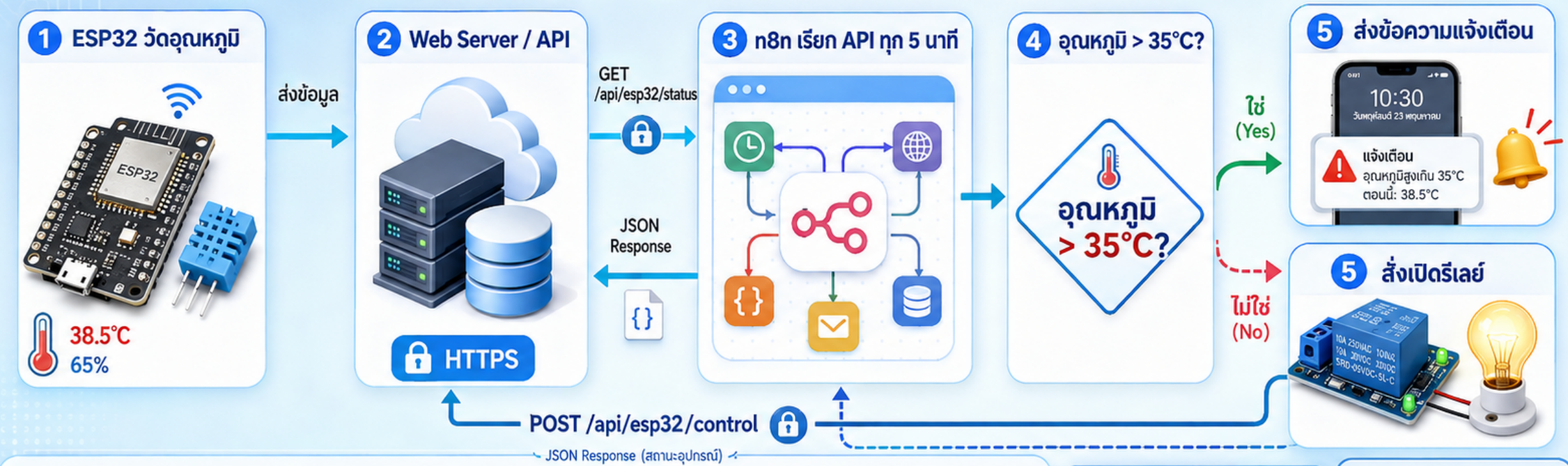


สรุป: API คือ
สะพานเชื่อมระหว่างระบบ



ตัวอย่างการใช้ API ใน n8n

ตรวจสอบอุณหภูมิจาก ESP32 และสั่งงานอัตโนมัติ



1. ESP32 ส่งข้อมูลไปยัง Web Server

- ESP32 วัดอุณหภูมิและความชื้น
- ส่งข้อมูลเป็น JSON ไปยัง Web Server
- เก็บข้อมูลในฐานข้อมูล

2. n8n ใช้ HTTP Request เรียก API

- n8n ตั้งเวลาเรียก API ทุก 5 นาที
- ใช้ HTTP Request (GET)
- รับข้อมูลแบบ JSON

3. IF Node ตรวจสอบเงื่อนไข

- ตรวจสอบว่าอุณหภูมิ > 35°C หรือไม่
- ถ้าใช่ → ดำเนินการแจ้งเตือนและสั่งงาน
- ถ้าไม่ใช่ → จบการทำงาน

4. แจ้งเตือนหรือควบคุมอุปกรณ์

- ส่งข้อความแจ้งเตือนผ่านมือถือ
- ส่งคำสั่งไปยัง ESP32 ให้เปิดรีเลย์
- ควบคุมอุปกรณ์อัตโนมัติ

ตัวอย่างข้อมูลตอบกลับ (JSON Response)

```
{
  temperature: 38.5,
  humidity: 65,
  relay: OFF
}
```

ความปลอดภัย

ควรใช้ HTTPS และ API Key เพื่อความปลอดภัย

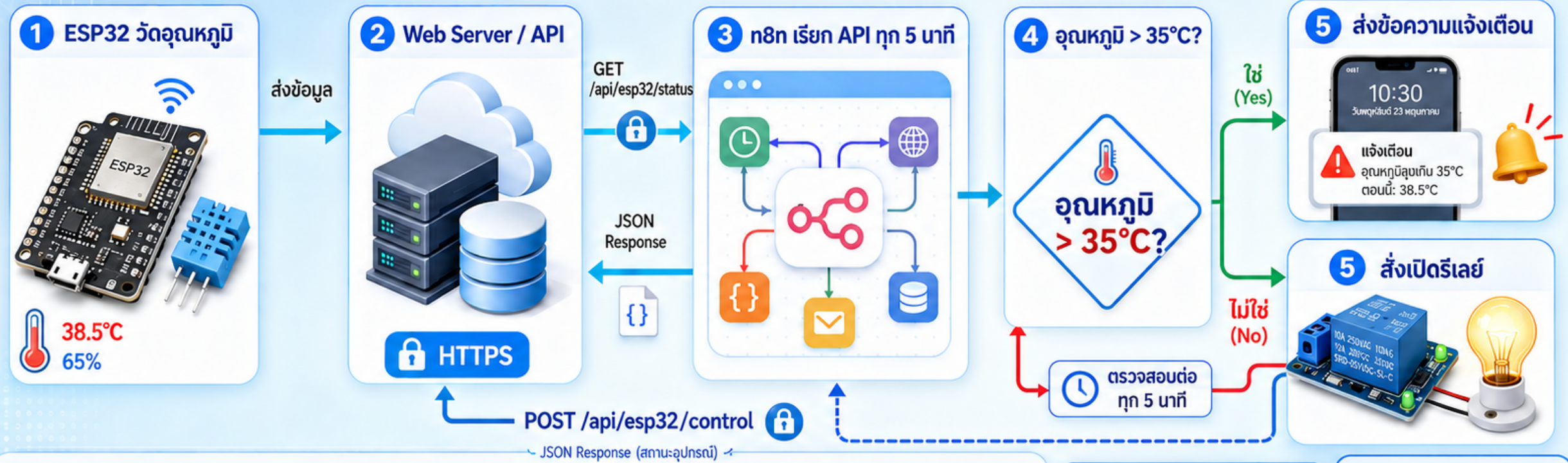


n8n + API ช่วยเชื่อมต่อข้อมูล ตัดสินใจ และสั่งงาน IoT โดยอัตโนมัติ



ตัวอย่างการใช้ API ใน n8n

ตรวจสอบอุณหภูมิจาก ESP32 และสั่งงานอัตโนมัติ



n8n + API ช่วยเชื่อมต่อข้อมูล ตัดสินใจ และสั่งงาน IoT โดยอัตโนมัติ





n8n เชื่อมต่อแพลตฟอร์มต่าง ๆ

ศูนย์กลางระบบอัตโนมัติที่รับข้อมูล ประมวลผล และส่งงานข้ามแพลตฟอร์ม



LINE

รับข้อความ • ตอบกลับ • แจ้งเตือน



Facebook

รับข้อมูลเพจ • เผยแพร่เนื้อหา



Instagram

จัดการข้อมูลบัญชีธุรกิจ



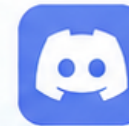
Telegram

รับคำสั่ง • ส่งข้อความและไฟล์



Discord

แจ้งเตือน • ส่งข้อความไปยังช่อง



WhatsApp

รับและส่งข้อความธุรกิจ



Email

รับอีเมล • ส่งอีเมลอัตโนมัติ

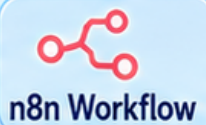


ESP32 / IoT

รับข้อมูลเซนเซอร์ • ควบคุมอุปกรณ์



Trigger



n8n Workflow



ตรวจสอบเงื่อนไข



Action

ตัวอย่างการใช้งาน

1

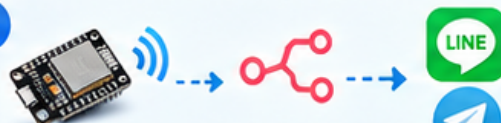


ลูกค้าส่งข้อความ →
n8n ตอบกลับอัตโนมัติ



Google Sheets → n8n → โพสต์หลายช่องทาง

3



ESP32 แจ้งเตือน → n8n → LINE และ Telegram



เชื่อมต่อผ่าน
API • Webhook • OAuth2
และจัดเก็บ Token
ใน Credentials



n8n ช่วยลดงานซ้ำ เชื่อมโยงข้อมูล และบริหารหลายแพลตฟอร์มจาก Workflow เดียว



n8n เชื่อมต่อแพลตฟอร์มต่าง ๆ

ศูนย์กลางระบบอัตโนมัติที่เชื่อมต่อโซเชี่ยลมีเดีย AI ฐานข้อมูล และอุปกรณ์ IoT

โซเชี่ยลมีเดียและระบบ



LINE

รับข้อความ • ตอบกลับ • แจ้งเตือน



Facebook

รับข้อมูลเพจ • เผยแพร่เนื้อหา



Instagram

จัดการข้อมูลบัญชีธุรกิจ



Telegram

รับคำสั่ง • ส่งข้อความและไฟล์

Discord



แจ้งเตือน • ส่งข้อความไปยังช่อง

WhatsApp



รับและส่งข้อความธุรกิจ

Email



รับอีเมล • ส่งอีเมลอัตโนมัติ

ESP32 / IoT



รับข้อมูลเซนเซอร์ • ควบคุมอุปกรณ์

n8n

เชื่อมต่อ AI



ChatGPT / OpenAI



Gemini



Claude

ความสามารถ AI



วิเคราะห์ข้อมูล



สรุปเนื้อหา



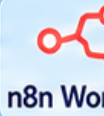
สร้างข้อความ



ตอบคำถามอัตโนมัติ



Trigger



n8n Workflow



ตรวจสอบเงื่อนไข



AI ประมวลผล



Action

API

ตัวอย่างการใช้งาน

1



ลูกค้าส่งข้อความ → n8n → AI วิเคราะห์ → ตอบกลับอัตโนมัติ

2



Google Sheets → n8n → สร้างเนื้อหาด้วย AI → โพสต์หลายช่องทาง

3



ESP32 แจ้งเตือน → n8n → AI สรุปข้อมูล → LINE และ Telegram



เชื่อมต่อผ่าน
API • Webhook • OAuth2
และจัดเก็บ Token
ใน Credentials



n8n + AI ช่วยวิเคราะห์ข้อมูล สร้างเนื้อหา และทำงานข้ามแพลตฟอร์มโดยอัตโนมัติ

