

# FAN LEVELLING

BY : YONE L5

# YONE



**Attivich  
Saveevanlop**



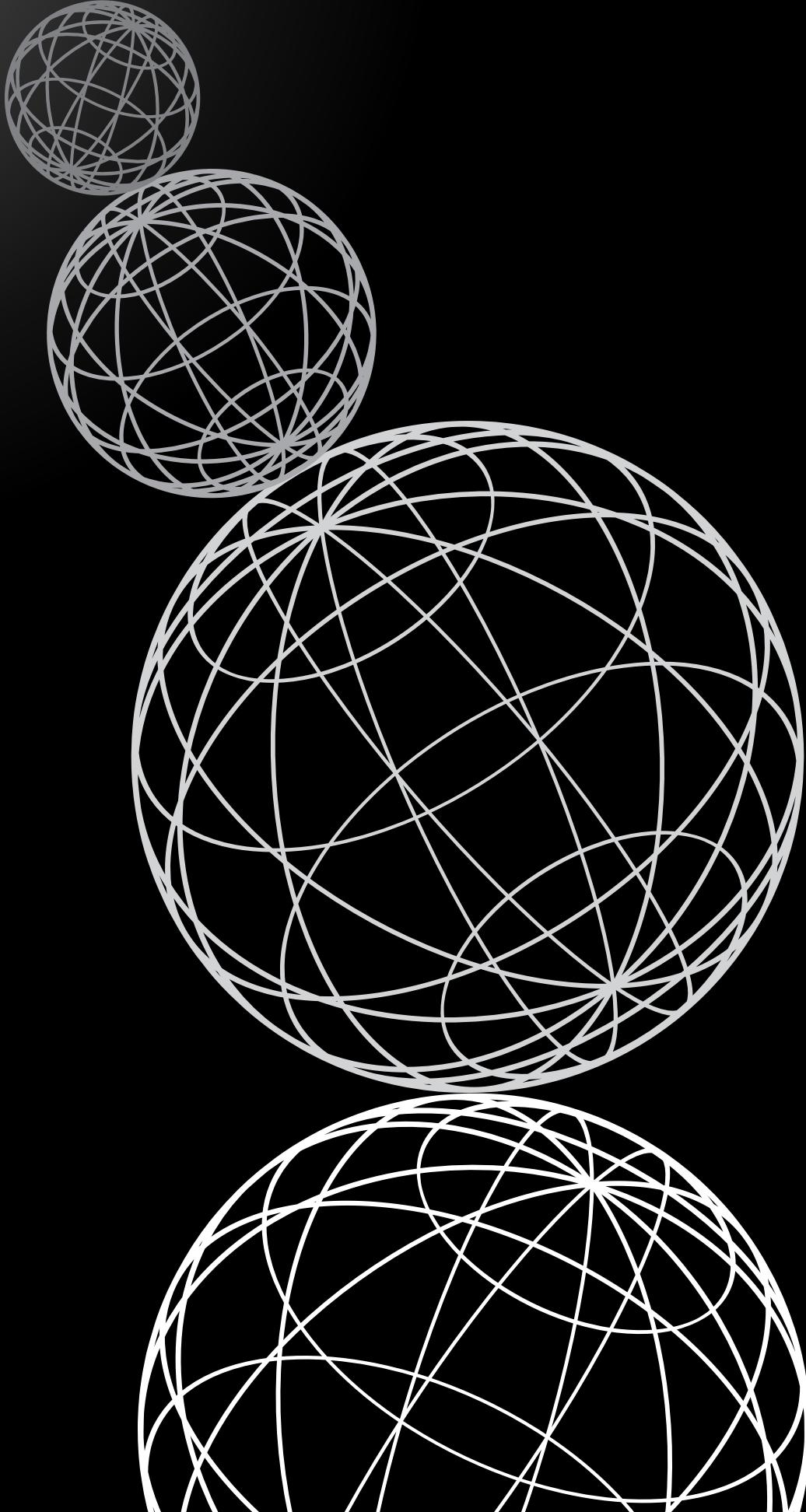
**Ratchapong  
Sertthikul**



**Chaiwat  
Phumjaroen**

# OVERVIEW

พัฒนาที่ปรับความเร็วตามอุณหภูมิโดยใช้เซนเซอร์ DHT11 ใน  
การวัดอุณหภูมิ สามารถปรับเป็น manual ได้ และสามารถดู  
ความเร็วของใบพัดได้ผ่านทาง webpage โดยใช้ Sensor  
TCRT5000 ในการวัดรอบพัดลม



# DETAIL/REQUIREMENT

- ESP32 DevKit
- DHT11 sensor (temperature)
- TCRT5000 IR sensor (RPM measurement)
- L293D motor driver IC
- DC Fan
- Power supply:
  - ESP32: 5V
  - Fan: 5V–12V

# SPECIFICATION

Temperature update rate: every 2s (DHT11 limitation)

Fan PWM resolution: 8-bit (0–255)

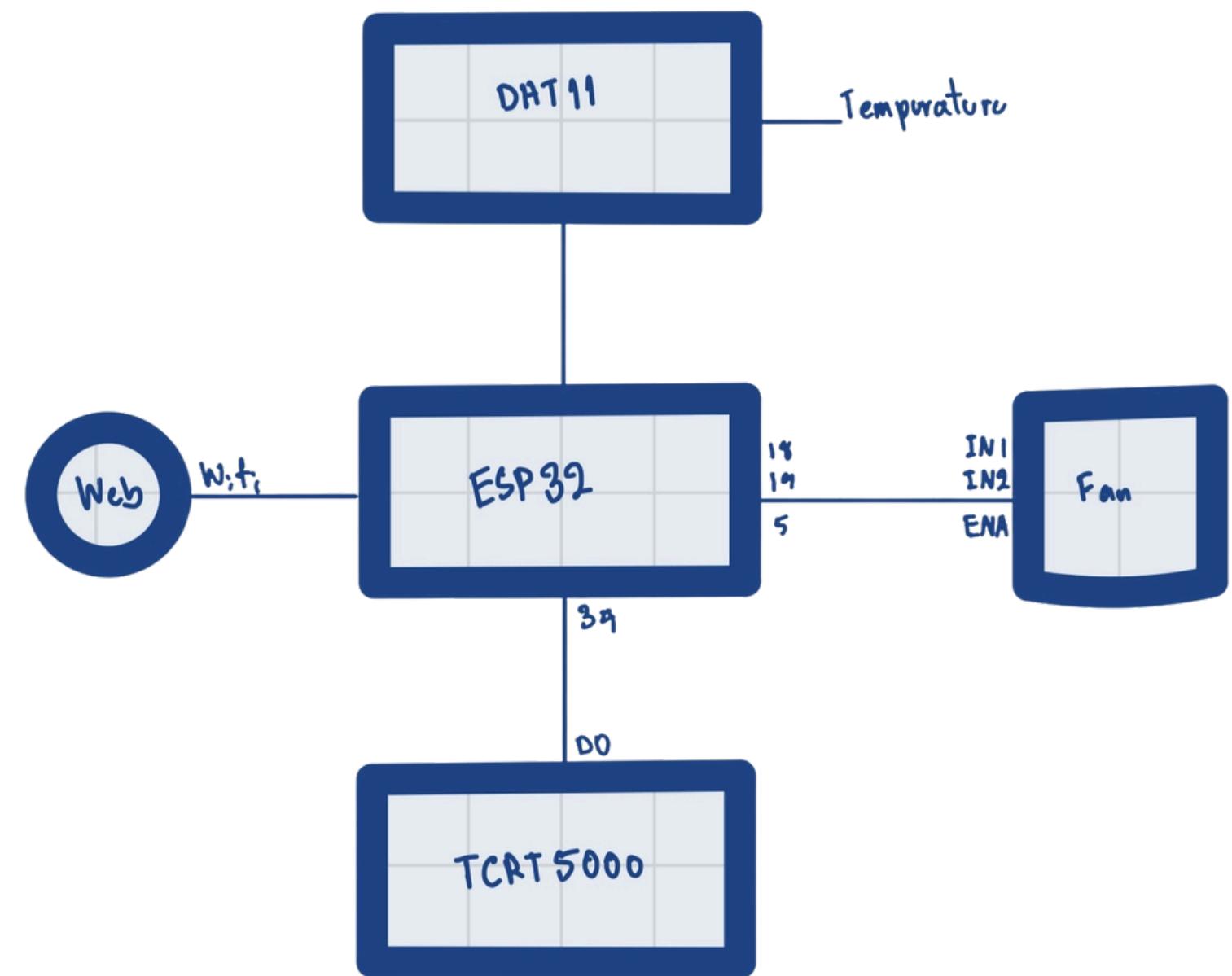
PWM frequency: 1 kHz (LEDC)

RPM calculation window: 1s

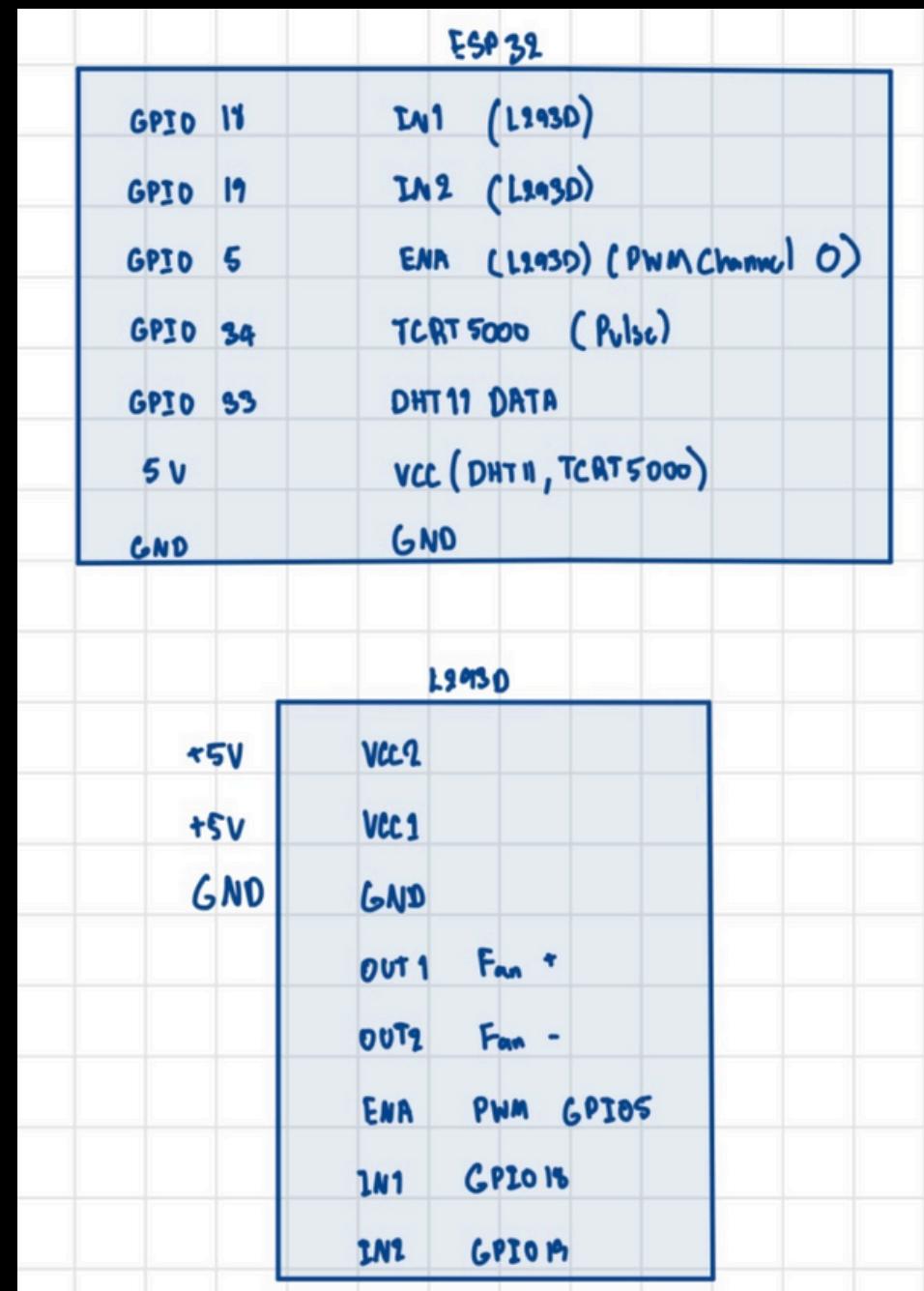
Web refresh interval: 1s

Startup time: <5s to connect to WiFi

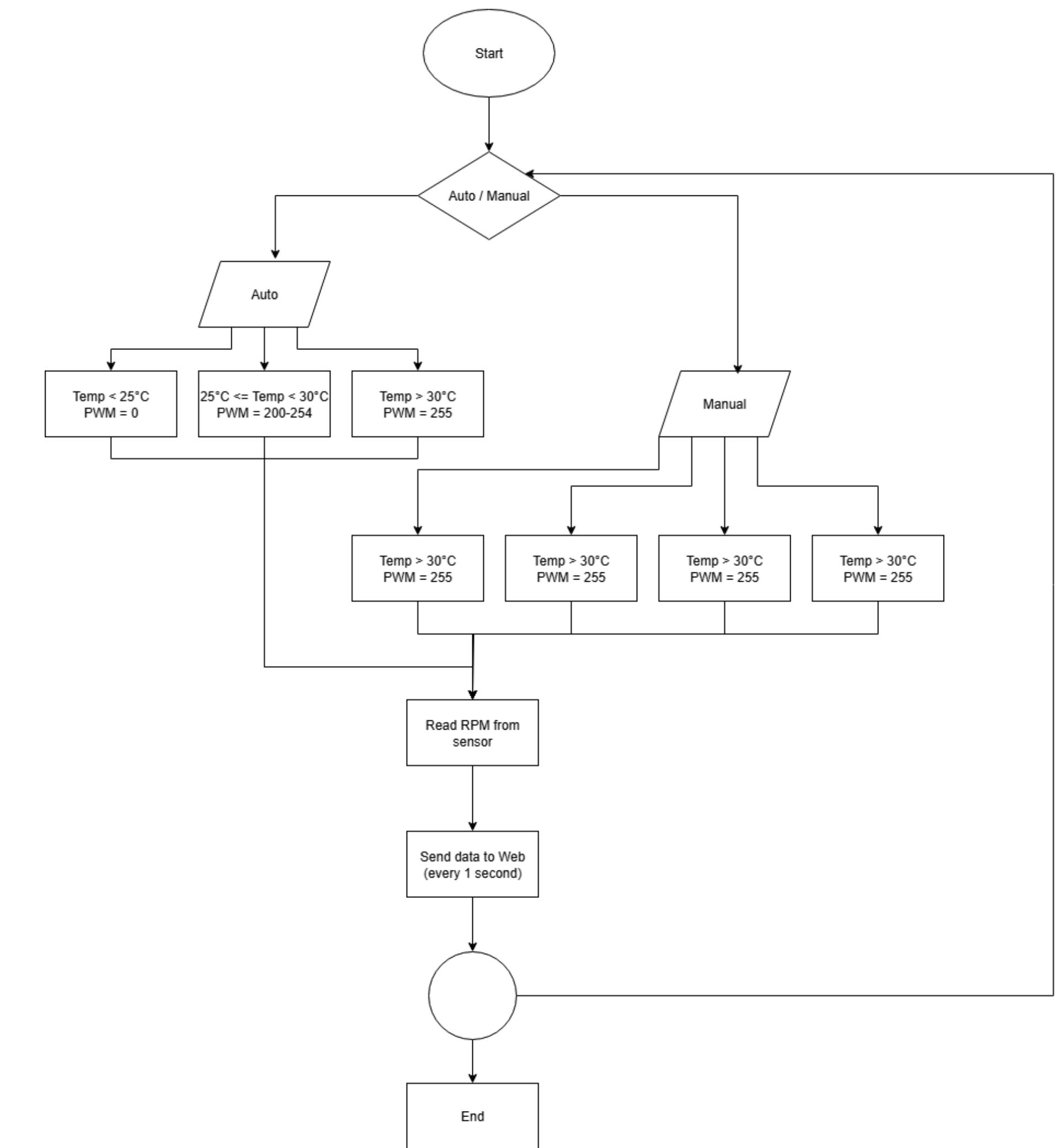
# ARCHITECTURAL DESIGN



# DETAILED DESIGN



# FLOWCHART & CODE



## กำหนดการพัดลมให้หมุนไปทางเดียว

```
Serial.begin(115200);
pinMode(IN1, OUTPUT);
pinMode(IN2, OUTPUT);
digitalWrite(IN1, HIGH); // Forward only
digitalWrite(IN2, LOW);
Pin Setup
```

## PWM ใช้ LEDC ของ ESP32

```
ledcSetup(0, 1000, 8);
ledcAttachPin(ENA, 0);
ledcWrite(0, pwmValue);
```

## ตั้งค่า tachometer & attach interrupt

```
pinMode(TACH_PIN, INPUT_PULLUP);
attachInterrupt(digitalPinToInterrupt(TACH_PIN), tachISR, FALLING);
```

## เริ่ม DHT11 และเชื่อมต่อ Wi-Fi

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

## ปรับอุณหภูมิ

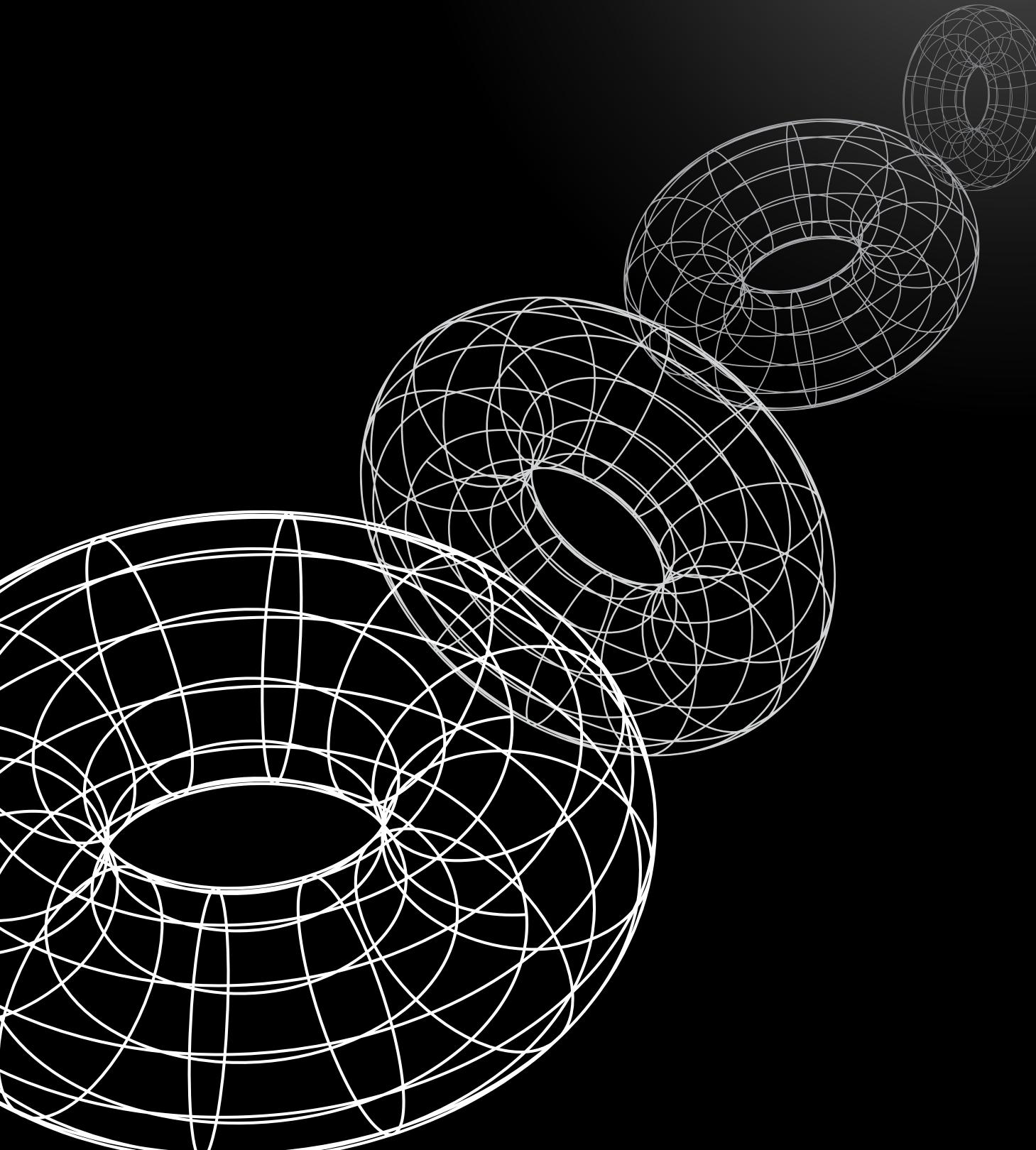
```
float tempC = dht.readTemperature();
if (isAutoMode) {
    if (tempC < 25) pwmValue = 0;
    else if (tempC >= 25 && tempC < 30) pwmValue = map(tempC * 10, 250, 300, 200, 255);
    else pwmValue = 255;
} else {
    switch(manualSpeed) {
        case 0: pwmValue = 0; break;
        case 1: pwmValue = 180; break;
        case 2: pwmValue = 200; break;
        case 3: pwmValue = 255; break;
    }
}
ledcWrite(0, pwmValue);
```

## บัน Pulse RPM

```
void IRAM_ATTR tachISR() {
    pulseCount++;
}
```

## คำนวณ RPM

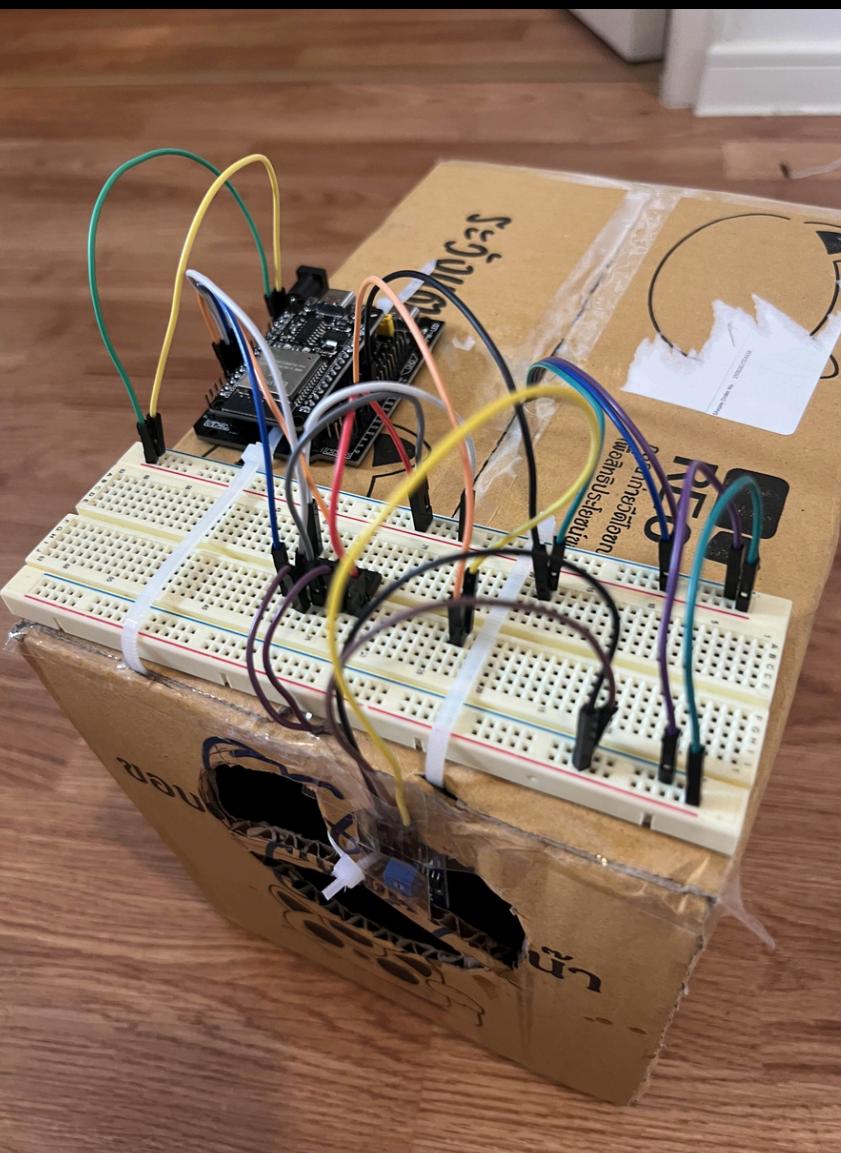
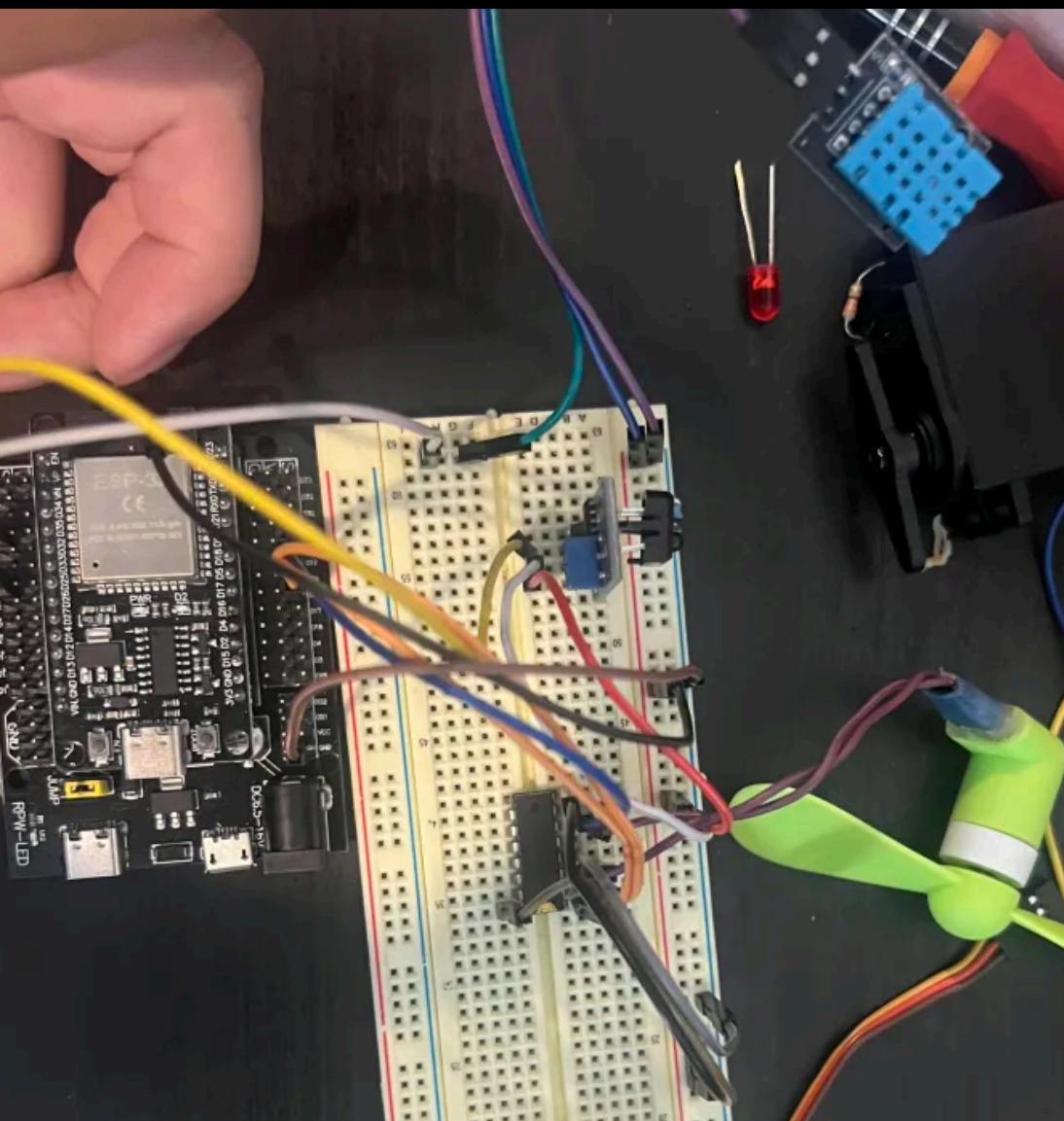
```
noInterrupts();
unsigned long count = pulseCount;
pulseCount = 0;
interrupts();
rpmValue = (count * 60) / 2; // 2 pulses per revolution
```



# TESTING AND RESULT

- **โหมด Auto**
  - <25 °C พัดลมหยุด (PWM = 0)
  - 25-30 °C พัดลมเริ่มทำงานแบบໄລ่ระดับ (PWM ประมาณ 200-255)
  - ≥30 °C → พัดลมทำงานเต็มสปีด (PWM = 255)
- **โหมด Manual**
  - Speed 0 พัดลมหยุด (0 RPM)
  - Speed 1 ความเร็วต่ำ (PWM ~180, RPM ~4500)
  - Speed 2 ความเร็วกลาง (PWM ~200, RPM ~5000)
  - Speed 3 ความเร็วสูงสุด (PWM 255, RPM ~5500)
- หน้าเว็บแสดงผลแบบ RealTime ทุก 1 วินาที
  - อุณหภูมิ, ค่า PWM, ค่า RPM, และสถานะโหมด

# PICTURE



## Fan Levelling

Temperature: 28.6°C

Fan PWM: 239

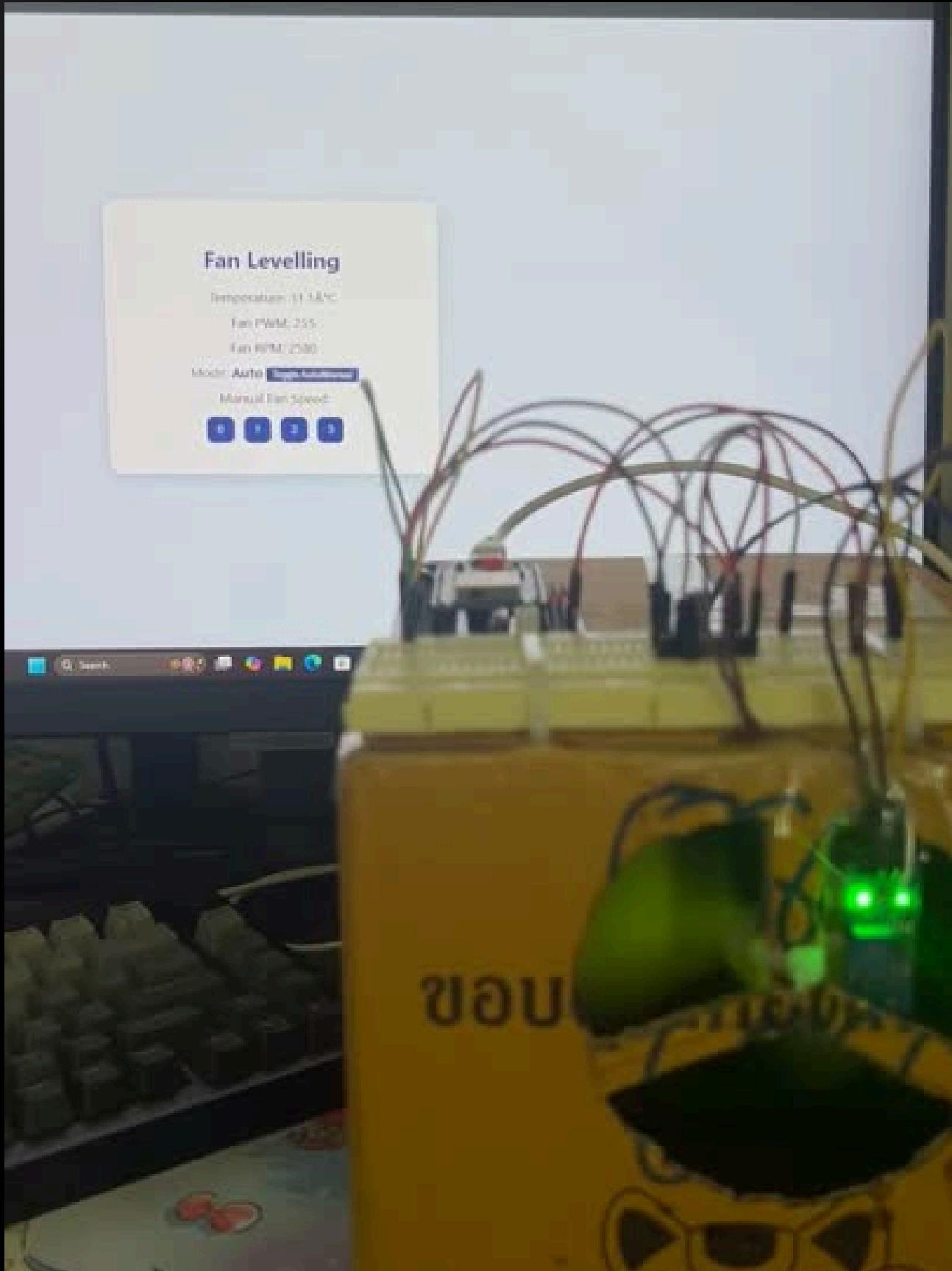
Fan RPM: 5730

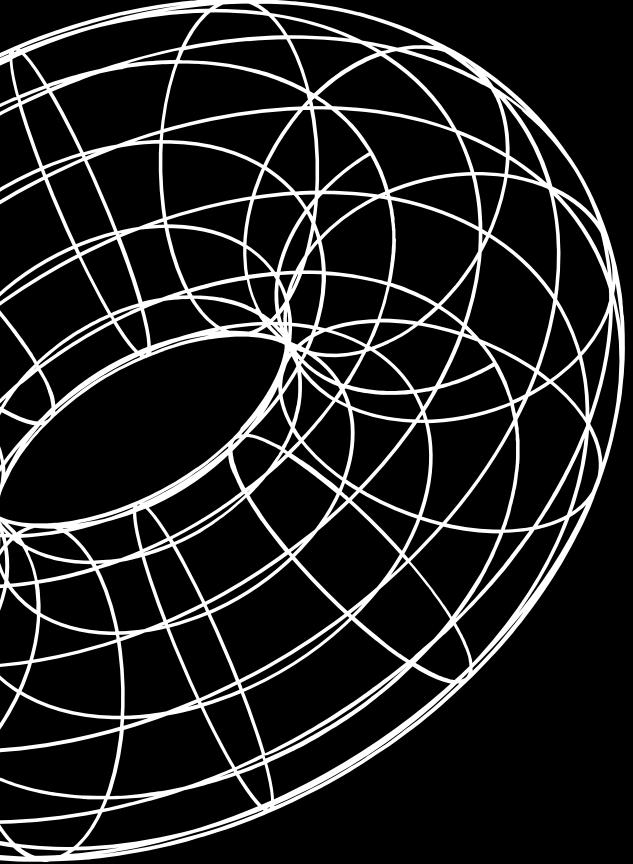
Mode: **Auto** [Toggle Auto/Manual](#)

Manual Fan Speed:

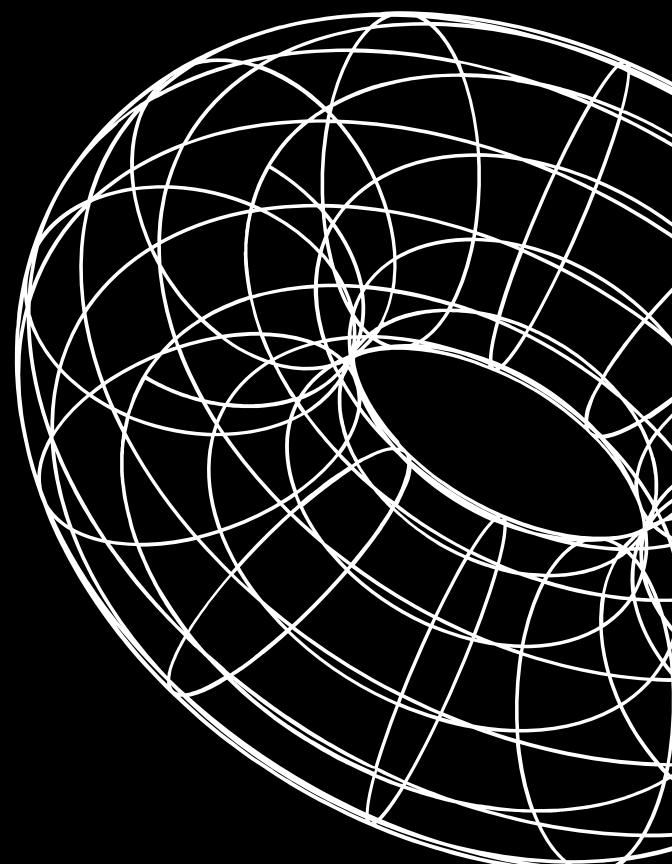
0 1 2 3

# VDO





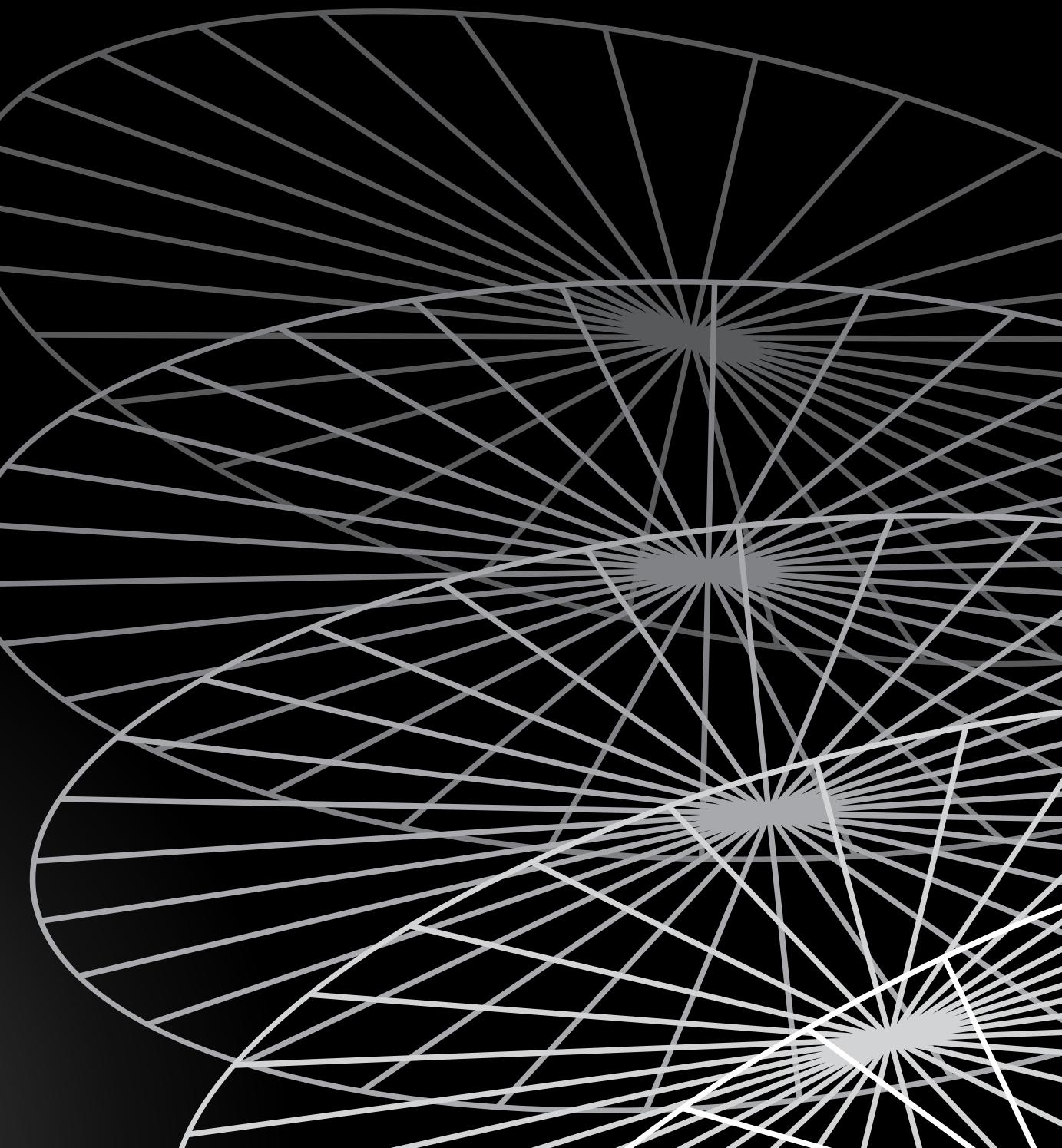
# PROBLEM AND SOLUTION



ใช้ rpm ในการปรับระดับใบพัดแต่ใบพัดหมุนไม่เสถียรจึงใช้ duty ในการปรับความเร่งพัดลม และใช้ tachometer มาแสดงผลเพื่อเช็คว่าพัดลมหมุนเร็วแค่ไหนแทน

# GANTT CHART

gantt		
Fan Levelling		
09/21/2025		
ศึกษาข้อมูล ESP32, DHT11, Fan Driver	done	Tong
ออกแบบ Flowchart & Block Diagram	done	Dale, Yim
เขียนโค้ดควบคุมพัดลม (PWM + Auto/Manual)	done	Yim
เพิ่ม Tachometer Feedback (RPM calculation)	done	Tong
สร้าง Web UI (ESP32 WebServer + HTML/JS)	done	Dale
ทดสอบโหมด Auto (Temp → PWM mapping)	done	All
ทดสอบโหมด Manual (Preset Speed)	done	All
ทดสอบ Tachometer (RPM accuracy)	done	All
เก็บข้อมูลการทดสอบ (ตาราง + กราฟ)	done	All
เขียนรายงาน + สุรุปผลการทำงาน	done	Yim
เตรียมการนำเสนอ (Presentation)	done	Dale



# THANK YOU

YONE L2