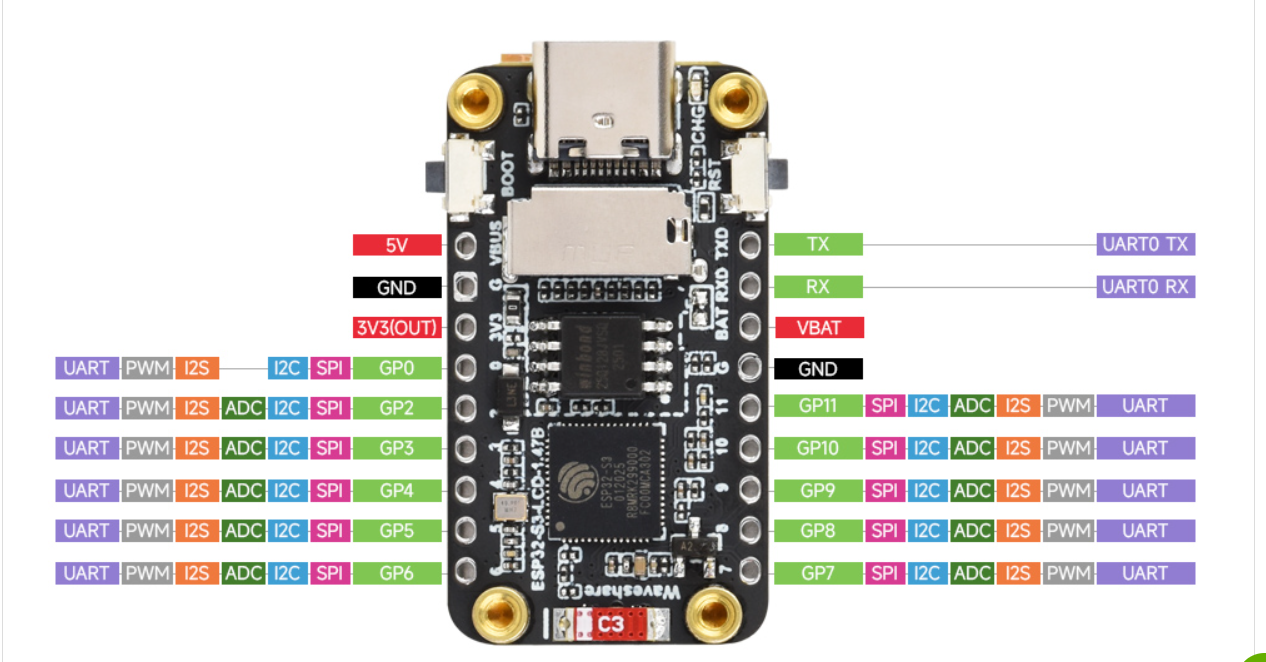
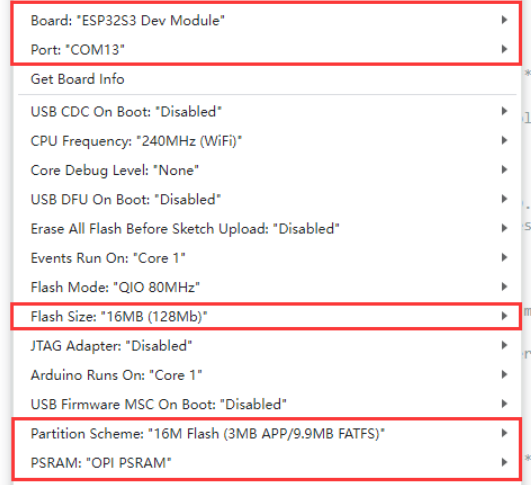
# Tư liệu về esp32s

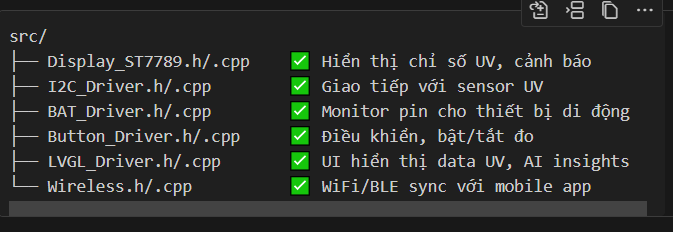
link thông tin:[esp32s3](https://www.waveshare.com/product/esp32-s3-lcd-1.47b.htm)

sơ đồ mạch pdf:<ESP32-S3-LCD-1.47_schematic_diagram.pdf>

## sơ đồ chân:



set up arduino IDE:  


code hareware tái sử dụng:  


# HƯỚNG DẪN CHÂN PIN & PHẦN CỨNG ESP32-S3 WAVESHARE

## Phân tích từ dự án mẫu PlatformIO

═════════════════════════════════════════

📋 MỤC LỤC

1. TỔNG QUAN DỰ ÁN & CẤU HÌNH CHUNG

2. SƠ ĐỒ CHÂN PIN CHI TIẾT THEO MODULE

3. CẤU HÌNH PHẦN CỨNG & THÔNG SỐ KỸ THUẬT

4. SO SÁNH THƯ VIỆN CỦA NHÀ SẢN XUẤT VÀ ARDUINO CHUẨN

5. MẪU CẤU HÌNH PLATFORMIO.INI

6. CÁC PHƯƠNG PHÁP TỐI ƯU VÀ LỜI KHUYÊN

7. HƯỚNG DẪN TÁI SỬ DỤNG CHO DỰ ÁN MỚI

═════════════════════════════════════════

## 1. TỔNG QUAN DỰ ÁN & CẤU HÌNH CHUNG

🎯 Mục tiêu:

Phân tích và tái sử dụng cấu hình phần cứng từ dự án ESP32-S3 Waveshare

📁 Framework: Arduino trên PlatformIO

🔧 Board: esp32-s3-devkitc-1

📊 Platform: espressif32

### CÁC MODULE ĐÃ PHÂN TÍCH:

• Màn hình Display ST7789 (LCD 1.47 inch, giao tiếp SPI)

• Giao tiếp I2C (Cảm biến con quay QMI8658)

• Giám sát Pin (ADC - Analog to Digital Converter)

• Nút nhấn người dùng (Buttons)

• Điều khiển đèn LED RGB (WS2812/Neopixel)

• Lưu trữ thẻ nhớ SD Card (SPI)

• Truyền thông không dây (WiFi/Bluetooth)

• Thư viện đồ họa LVGL

═════════════════════════════════════════

## 2. SƠ ĐỒ CHÂN PIN CHI TIẾT THEO MODULE

### 🖥️ MÀN HÌNH DISPLAY ST7789 (Giao tiếp SPI)

|  |  |  |
| --- | --- | --- |
| Chức năng | GPIO | Mô tả |
| SPI\_MOSI | 45 | Gửi dữ liệu từ ESP32 đến màn hình |
| SPI\_SCK | 40 | Xung đồng hồ SPI |
| SPI\_CS | 42 | Chọn thiết bị (LOW = hoạt động) |
| DC | 41 | Chuyển giữa dữ liệu và lệnh |
| RST | 39 | Reset phần cứng màn hình |
| BL | 46 | Đèn nền (PWM - điều chỉnh độ sáng 0–255) |

💡 Ghi chú về Màn hình:

• Độ phân giải: 172x320 pixels (điểm ảnh)

• IC điều khiển: ST7789V

• Điện áp hoạt động: 3.3V

• Chế độ SPI: Mode 0 (CPOL=0, CPHA=0)

• Tần số SPI tối đa: 40MHz

• Xoay màn hình: Hỗ trợ 0°, 90°, 180°, 270°

### 📱 GIAO TIẾP I2C (Inter-Integrated Circuit)

|  |  |  |
| --- | --- | --- |
| Chức năng | GPIO | Mô tả |
| SDA | 6 | Truyền dữ liệu I2C |
| SCL | 7 | Xung đồng hồ I2C |

💡 Ghi chú về I2C:

• Tần số hoạt động: 400kHz (Chế độ nhanh)

• Điện trở kéo lên: 4.7kΩ (đã có sẵn trên board)

• Điện áp logic: 3.3V

• Hỗ trợ nhiều thiết bị trên cùng bus

### 🔋 GIÁM SÁT PIN (ADC - Analog to Digital Converter)

|  |  |  |
| --- | --- | --- |
| Chức năng | GPIO | Mô tả |
| BAT\_ADC | 1 | Đo điện áp pin qua mạch chia điện áp |

💡 Ghi chú về Pin:

• Độ phân giải ADC: 12-bit (0-4095)

• Dải điện áp: 0-3.3V (ánh xạ tới điện áp pin)

• Tỉ lệ chia điện áp: 2:1 (cần nhân x2 để có điện áp thực)

• Hiệu chỉnh: 350 (trừ đi để có giá trị chính xác)

• Công thức: Điện áp pin = ((giá trị ADC - 350) \* 2 \* 3.3) / 4095

### 🔘 NÚT NHẤN NGƯỜI DÙNG (User Input)

|  |  |  |
| --- | --- | --- |
| Chức năng | GPIO | Mô tả |
| USER\_BTN | 0 | LOW = nhấn, HIGH = thả |

💡 Ghi chú về Nút nhấn:

• Kích hoạt điện trở kéo lên bên trong

• Thời gian chống rung: 50ms (khuyến nghị)

• Logic mức thấp hoạt động

• Chức năng nút boot

### 🌈 ĐIỀU KHIỂN ĐÈN LED RGB (WS2812/Neopixel)

|  |  |  |
| --- | --- | --- |
| Chức năng | GPIO | Mô tả |
| RGB\_LED | 38 | Gửi tín hiệu điều khiển LED RGB WS2812 |

💡 Ghi chú về LED RGB:

• Giao thức: WS2812 (800kHz)

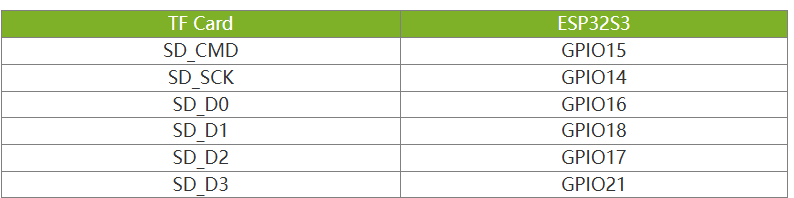
• Màu sắc: 24-bit RGB (8-bit cho mỗi kênh màu)

• Số lượng LED: 1 (có thể mở rộng thêm)

• Nguồn điện: 5V (với level shifter) hoặc 3.3V

• Dòng điện: ~20mA cho mỗi kênh màu

### 💾 LƯU TRỮ THẺ NHỚ SD CARD (SPI)



💡 Ghi chú về Thẻ SD:

• Chế độ SPI: Mode 0

• Tần số khuyến nghị: 25MHz, tối đa: 40MHz

• Hệ thống tập tin: FAT16/FAT32

• Dung lượng: Hỗ trợ đến 32GB (SDHC)

• Điện áp hoạt động: 3.3V

### 🌀 CẢM BIẾN CON QUAY QMI8658 (I2C)

|  |  |  |
| --- | --- | --- |
| Thông tin | Giá trị | Mô tả |
| Địa chỉ I2C | 0x6B | Địa chỉ 7-bit cố định |
| Chân SDA | 6 | Dùng chung với bus I2C chính |
| Chân SCL | 7 | Dùng chung với bus I2C chính |

💡 Ghi chú về Cảm biến Con quay:

• 6 trục: 3 trục gia tốc kế + 3 trục con quay hồi chuyển

• Dải gia tốc: ±2g, ±4g, ±8g, ±16g

• Dải con quay: ±256°/s, ±512°/s, ±1024°/s, ±2048°/s

• Tần số lấy mẫu: 1Hz đến 8kHz

• Chế độ nguồn: Bình thường, Tiết kiệm điện, Tạm ngừng

### 📡 TRUYỀN THÔNG KHÔNG DÂY (Wireless Communication)

|  |  |  |
| --- | --- | --- |
| Tính năng | Trạng thái | Mô tả |
| WiFi | Tích hợp sẵn | Chuẩn 802.11 b/g/n, 2.4GHz |
| Bluetooth | Tích hợp sẵn | Hỗ trợ BLE 5.0 và Bluetooth cổ điển |
| Anten | PCB/Ngoài | Tùy thuộc vào thiết kế phần cứng |

═════════════════════════════════════════

## 3. CẤU HÌNH PHẦN CỨNG & THÔNG SỐ KỸ THUẬT

### 🔧 CẤU HÌNH GIAO TIẾP SPI

|  |  |  |  |
| --- | --- | --- | --- |
| Thông số | Màn hình ST7789 | Thẻ SD | Mô tả |
| Bus SPI | HSPI | VSPI | Sử dụng 2 bus SPI riêng biệt |
| Tần số | 40MHz | 25MHz | Tần số xung đồng hồ |
| Chế độ | 0 | 0 | CPOL = 0, CPHA = 0 |
| Thứ tự bit | MSB trước | MSB trước | Gửi bit có nghĩa cao trước |

### 🔧 CẤU HÌNH GIAO TIẾP I2C

|  |  |  |
| --- | --- | --- |
| Thông số | Giá trị | Mô tả |
| Tần số | 400kHz | Chế độ I2C nhanh |
| Chân SDA | GPIO 6 | Đường truyền dữ liệu |
| Chân SCL | GPIO 7 | Đường xung đồng hồ |
| Điện trở kéo lên | 4.7kΩ | Có sẵn trên board |
| Thời gian chờ | 1000ms | Timeout chờ giao tiếp (khuyến nghị) |

═════════════════════════════════════════

## 4. SO SÁNH THƯ VIỆN CỦA NHÀ SẢN XUẤT VÀ ARDUINO CHUẨN

### 📚 THƯ VIỆN CHO MÀN HÌNH

|  |  |  |  |
| --- | --- | --- | --- |
| Khía cạnh | Waveshare | Arduino (TFT\_eSPI) | Adafruit GFX |
| Cài đặt | Cắm và chạy | Cấu hình thủ công | Cấu hình thủ công |
| Hiệu suất | Tối ưu | Nhanh | Trung bình |
| Dung lượng bộ nhớ | Vừa phải | Thấp | Cao |
| Tính năng | Cơ bản | Phong phú | Toàn diện |
| Tùy chỉnh | Hạn chế | Cao | Cao |
| Tài liệu | Tốt | Xuất sắc | Xuất sắc |

### 🎯 KHUYẾN NGHỊ SỬ DỤNG:

• Dự án thử nghiệm/học tập: Sử dụng thư viện Arduino chuẩn

• Dự án thương mại: Kết hợp cả hai, ưu tiên hiệu suất

• Tùy chỉnh cao: Thư viện Arduino chuẩn

• Chạy nhanh không cần tùy chỉnh: Thư viện Waveshare

═════════════════════════════════════════

## 6. CÁC PHƯƠNG PHÁP TỐI ƯU VÀ LỜI KHUYÊN

### ⚡ TỐI ƯU HÓA HIỆU SUẤT

|  |  |
| --- | --- |
| Thành phần | Mẹo tối ưu |
| SPI | Dùng DMA, gộp buffer, set đúng tần số phù hợp |
| I2C | Gộp lệnh, sử dụng ACK/NACK hiệu quả, tránh delay không cần thiết |
| RAM | Dùng PSRAM cho ảnh/đệm lớn, tránh malloc liên tục |

### 🛡️ ĐỘ TIN CẬY & CHỐNG LỖI

|  |  |
| --- | --- |
| Kỹ thuật | Gợi ý thực hiện |
| Xử lý lỗi | Kiểm tra giá trị trả về, set timeout |
| Giám sát phần cứng | Theo dõi GPIO, điện áp, nhiệt độ |
| Phần mềm dự phòng | Thêm watchdog, cơ chế reset mềm |

═════════════════════════════════════════

## 7. HƯỚNG DẪN TÁI SỬ DỤNG CHO DỰ ÁN MỚI

### 🚀 DANH SÁCH KIỂM TRA NHANH

|  |  |
| --- | --- |
| Việc cần làm | Trạng thái |
| Sao chép platformio.ini | ☐ |
| Kiểm tra chân pin | ☐ |
| Cài thư viện cần thiết | ☐ |
| Kiểm tra GPIO cơ bản | ☐ |
| Kiểm tra I2C/SPI hoạt động đúng | ☐ |
| Thêm xử lý lỗi & watchdog | ☐ |
| Xem lại quản lý nguồn điện & dòng | ☐ |
| Tối ưu hóa bộ nhớ & tốc độ | ☐ |

## 📋 HƯỚNG DẪN TRIỂN KHAI TỪNG BƯỚC

|  |  |
| --- | --- |
| Bước | Mô tả công việc |
| 1 | Tạo dự án PlatformIO mới, chỉnh sửa platformio.ini |
| 2 | Kiểm tra thao tác GPIO, nguồn điện, không xung đột chân |
| 3 | Thêm thiết bị ngoại vi từng phần (LED → Cảm biến → Màn hình → SD) |
| 4 | Tích hợp vào main app, kiểm thử toàn bộ |

═════════════════════════════════════════

# 📝 GHI CHÚ & TÀI LIỆU THAM KHẢO

🔗 Tài nguyên hữu ích:

• Sổ tay kỹ thuật ESP32-S3

• Tài liệu PlatformIO

• Tài liệu Arduino ESP32 Core

• Tài liệu chính thức Waveshare

⚠️ Cảnh báo quan trọng:

• Luôn xác minh mức điện áp trước khi kết nối

• Kiểm tra yêu cầu dòng điện cho tất cả thiết bị ngoại vi

• Xác thực chân pin với board cụ thể của bạn

• Kiểm thử kỹ lưỡng trước khi triển khai sản xuất

## 📅 Thông tin phiên bản:

• Phiên bản tài liệu: 1.0

• Ngày: 28 tháng 6, 2025

• Dựa trên: Dự án mẫu Waveshare ESP32-S3

• Framework: Arduino/PlatformIO

## 💡 Mẹo sử dụng hiệu quả:

• Đọc phần "Tổng quan" trước khi bắt đầu

• Tham khảo bảng pin mapping khi đấu nối

• Sử dụng template platformio.ini cho dự án mới

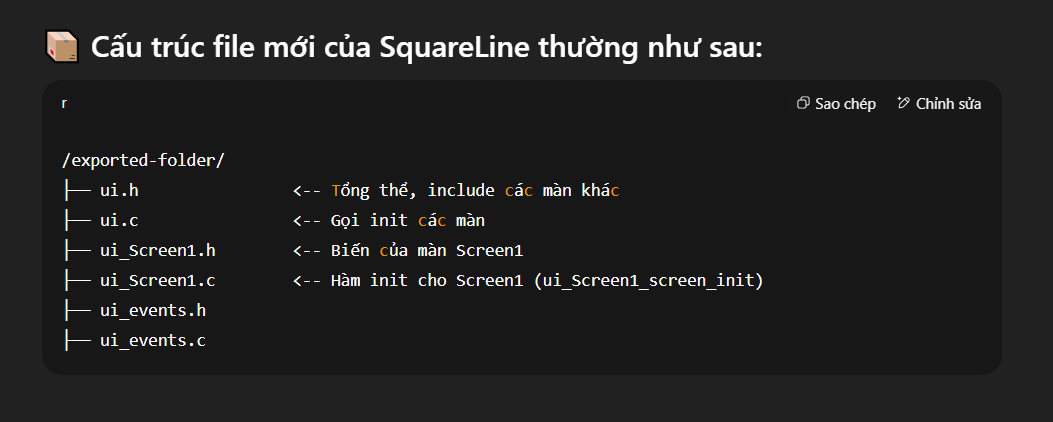
• Theo dõi checklist khi triển khai

═════════════════════════════════════════

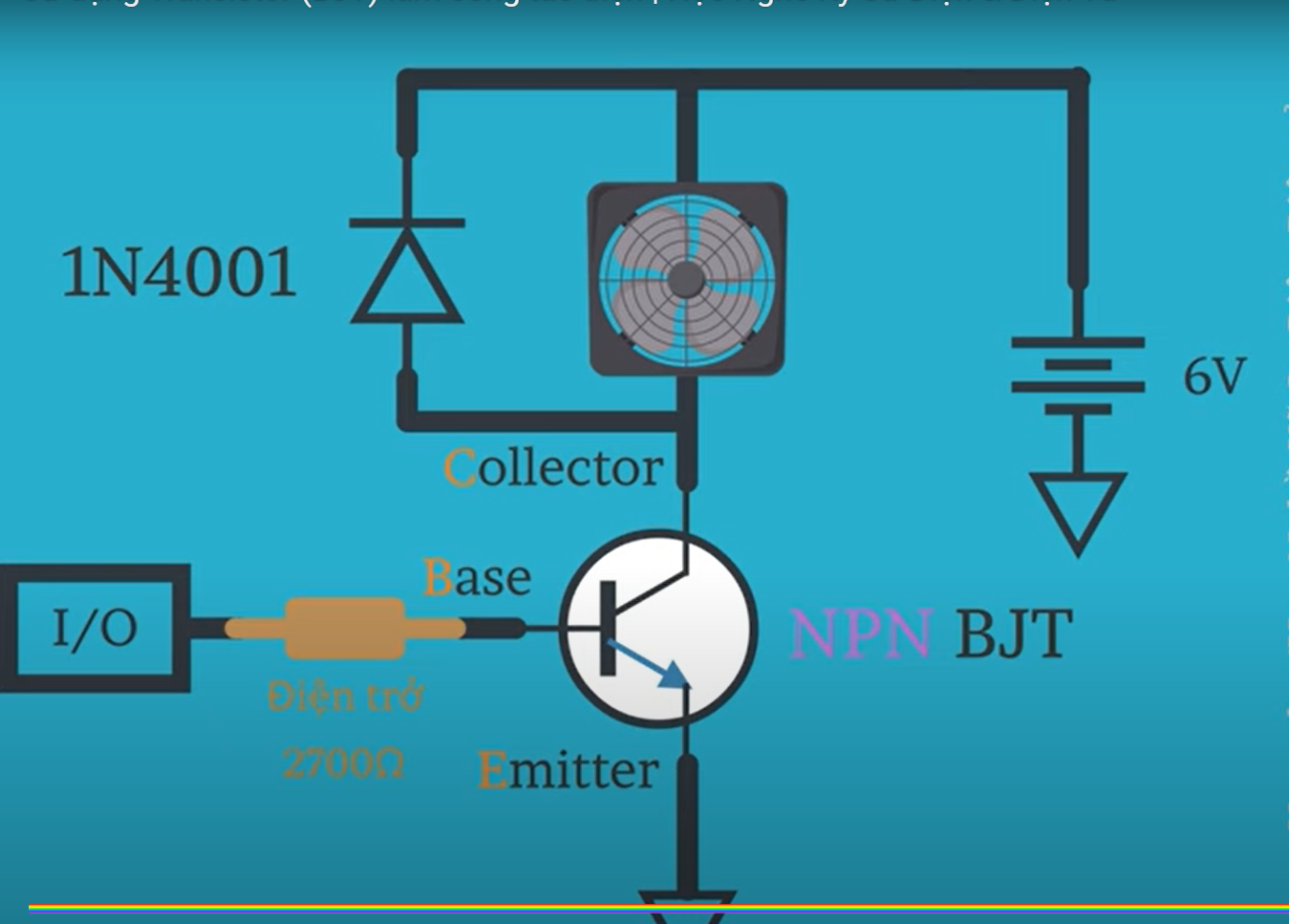
KẾT THÚC TÀI LIỆU

═════════════════════════════════════════

## Cấu trúc cải 1 file UI



## Sơ đồ nối chân cho motor rung



Fix lỗi pwm trên esp32  
https://youtu.be/FDRiwAaMyNE?si=RrAChvgZ9FT9h8yC

## Form dev

oke chúng ta sẽ bỏ qua lỗi vì chúng ta sẽ lập trình trên đây sau đó tôi sẽ copy mã qua .ino bên arduino để chạy

## tự học thời gian thực

🎯 TÓM LẠI - IMPLEMENTATION STRATEGY:

🔧 Step 1: Basic Online Learning (Arduino IDE)

1. Copy code ESP32-S3 adaptive ở trên
2. Upload lên ESP32-S3
3. Device sẽ tự học local patterns

📊 Step 2: Data Collection

* ESP32-S3 tự động lưu predictions vs actual
* Learning data stored trong Flash memory
* Display learning statistics trên LCD

🌐 Step 3: Cloud Integration (Optional)

* ESP32-S3 sync data lên server mỗi tuần
* Server aggregate learning từ nhiều devices
* Download improved corrections về ESP32-S3

📈 Step 4: Continuous Improvement

* Model base không đổi (trained trên Kaggle)
* Chỉ adaptation layer thay đổi
* Performance tăng dần theo thời gian

🚀 KẾT QUẢ MONG ĐỢI:

* Week 1: MAE = 0.5 (baseline)
* Month 1: MAE = 0.3 (learned local patterns)
* Month 6: MAE = 0.2 (optimized for location)
* Year 1: MAE = 0.15 (fully adapted)