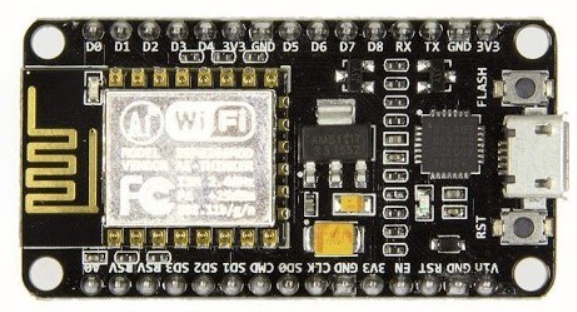
# **CHƯƠNG 1: TỔNG QUÁT**

1. **Giới thiệu chung về board Esp8266Node MCU:**

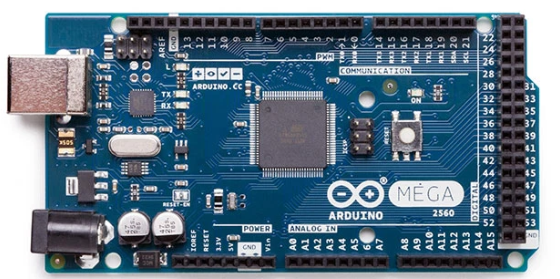
Esp8266 là một dòng chip tích hợp Wifi-2.4Ghz có thể lập trình được.



Hình 1: Module Esp8266 Node MCU

* Thông số kỹ thuật:
  + Wifi: 2.4 GHz hỗ trợ chuẩn 802.11 b/g/n
  + Điện áp hoạt động: 3.3V
  + Điện áp vào: 5V thông qua cổng USB
  + Số chân I/O: 11 (tất cả các chân I/O đều có Interrupt/PWM/I2C/One-wire)
  + Số chân Analog Input: 1 (điện áp vào tối đa 3.3V)
  + Bộ nhớ Flash: 4MS
  + Hỗ trợ bảo mật: WPA/WPA2
  + Tích hợp giao thức TCP/IP
  + Lập trình trên các ngôn ngữ: C/C++, Micropython, …

1. **Giới thiệu chung về Arduino Mega:**



Hình 2: Board Arduino Mega 2560

Arduino Mega 2560 là một vi điều khiển bằng cách sử dụng Atmega2560. Arduino mega khác với các vi xử lý khác vì không sử dụng FTDI chip điều khiển chuyển tín hiệu từ USB để xử lý. Thay vào đó, nó sử dụng Atmega16U2 lập trình như là một công cụ chuyển đổi tín hiệu USB. Cơ bản arduino mega2560 vẫn giống arduino uno r3 chỉ khác số lượng chân và nhiều tính năng mạnh mẽ hơn.

* Thông số kỹ thuật:
  + Điện áp hoạt động: 5V (để đảm bảo mạch hoạt động tốt thì nên cấp nguồn từ 6~9VDC)
  + Điện áp vào: 5V thông qua cổng USB
  + Số chân I/O: 54 (trong đó có 15 chân cung cấp PWM)
  + Số chân Analog Input: 1 (điện áp vào tối đa 3.3V)
  + Bộ nhớ Flash: 256KB

1. **Giới thiệu chung về mạch điều khiển động cơ DC BTS7960 43A:**



Hình 3: Mạch điều khiển động cơ DC

Mạch cầu H-BTS7960 43A có đầy đủ tính năng current sense (kết hợp với điện trở đo dòng), tạo dead time, chống qúa nhiệt, quá áp, quá dòng, sụt áp và ngắn mạch.

* Thông số kỹ thuật:
  + Điện áp hoạt động: 6~27V
  + Tín hiệu logic điều khiển: 3.3V ~ 5V
  + Tự động shutdown khi điện áp thấp: để tránh điều khiển động cơ ở mức điện áp thấp thiết bị sẽ tự shutdown. Nếu điện áp < 5.5V, driver sẽ tự ngắt điện và sẽ mở lại sau khi điện áp > 5.5V.
  + Bảo vệ quá nhiệt: BTS7960 bảo vệ chống quá nhiệt bằng cảm biến nhiệt tích hợp bên trong. Đầu ra sẽ bị ngắt khi có hiện tượng quá nhiệt.
* Sơ đồ chân:
  + VCC: nguồn tạo các mức logic điều khiển (5V – 3V3)
  + R\_EN = 0: Disable nửa cầu H phải, R\_EN = 1: Enable nửa cầu H bên phải.
  + L\_EN = 0: Disable nửa cầu H trái, L\_EN = 1: Enable nửa cầu H bên trái.
  + RPWM = 1 và LPWM = 0: mô tơ quay thuận.
  + RPWM = 0 và LPWM = 1: mô tơ quay nghịch.
  + RPWM = 0 và LPWM = 0 hoặc RPWM = 1 và LPWM = 1: Dừng.

1. **Giới thiệu chung về động cơ DC servo giảm tốc hành tinh Planetary GP36:**



Hình 4: Động cơ DC Servo Giảm tốc

Động cơ được tích hợp encoder quang với độ phân giải đĩa encoder lên đến 500CPR, 2 kênh A-B, hộp số giảm tốc hành tinh, lõi động cơ chất lượng tốt cho moment lực kéo khỏe thích hợp cho các ứng dụng robot hoặc các cơ cấu cơ khí cần độ chính xác cao: Robocon, Robot tự hành, …

* Thông số kỹ thuật:
  + Điện áp sử dụng: 12VDC
  + Dòng tiêu thụ: 3A
  + Điện áp cấp cho encoder: 5VDC
  + Hộp số kim loại giảm tốc hành tinh Planetary
  + Tỉ số giảm tốc: 1:14 hoặc 1:27

1. **Mạch giảm áp Buck DC-DC LM2596 3A:**

****

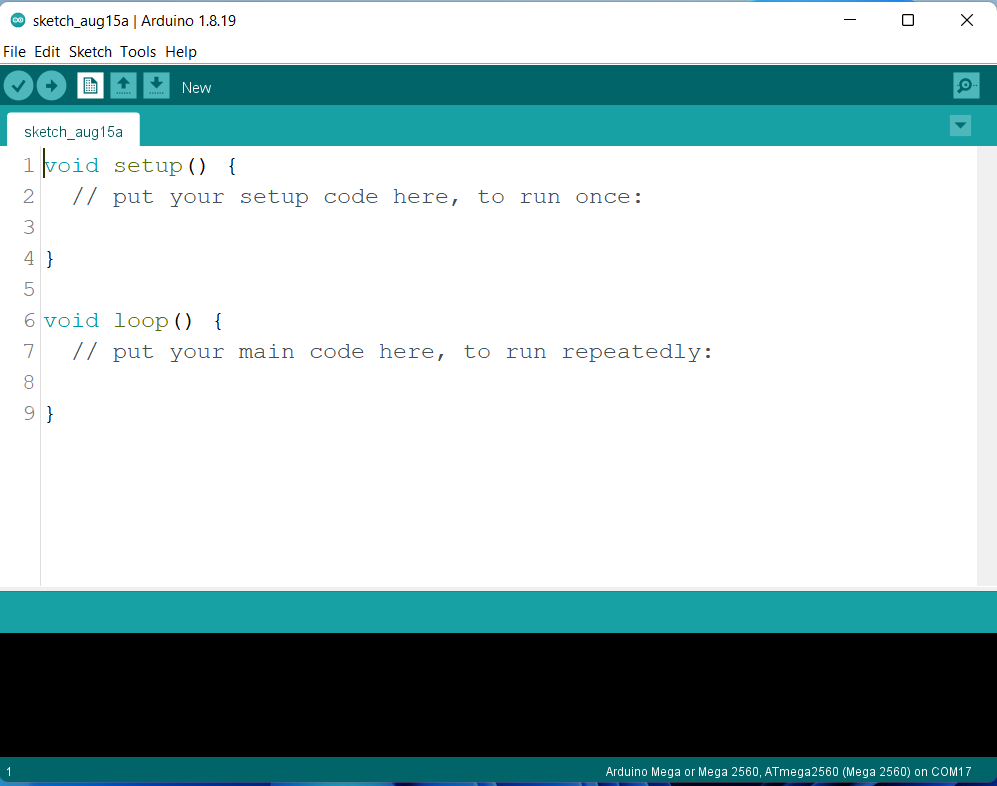
Hình 5: Mạch giảm áp DC-DC

Mạch có khả năng giảm áp từ 30V xuống 1.5V mà vẫn đạt hiệu suất 92%.

* Thông số kỹ thuật:
  + Điện áp đầu vào: từ 3V đến 30V
  + Điện áp đầu ra: điều chỉnh được từ 1.5V đến 30V
  + Dòng đáp ứng tối đa là 3A
  + Công suất: 15W

1. **Giới thiệu về Aruino IDE và ngôn ngữ lập trình cho Arduino Mega2560 và Esp 8266:**

Môi trường lập trình đơn giản và dễ sử dụng, ngôn ngữ lập trình Wiring dễ hiểu và dựa trên nền tảng C/C++ rất quen thuộc với người làm kỹ thuật. Và quan trọng là số lượng thưu viện code được viết sẵn và chia sẻ bởi cộng đồng nguồn mở là cực kỳ lớn.

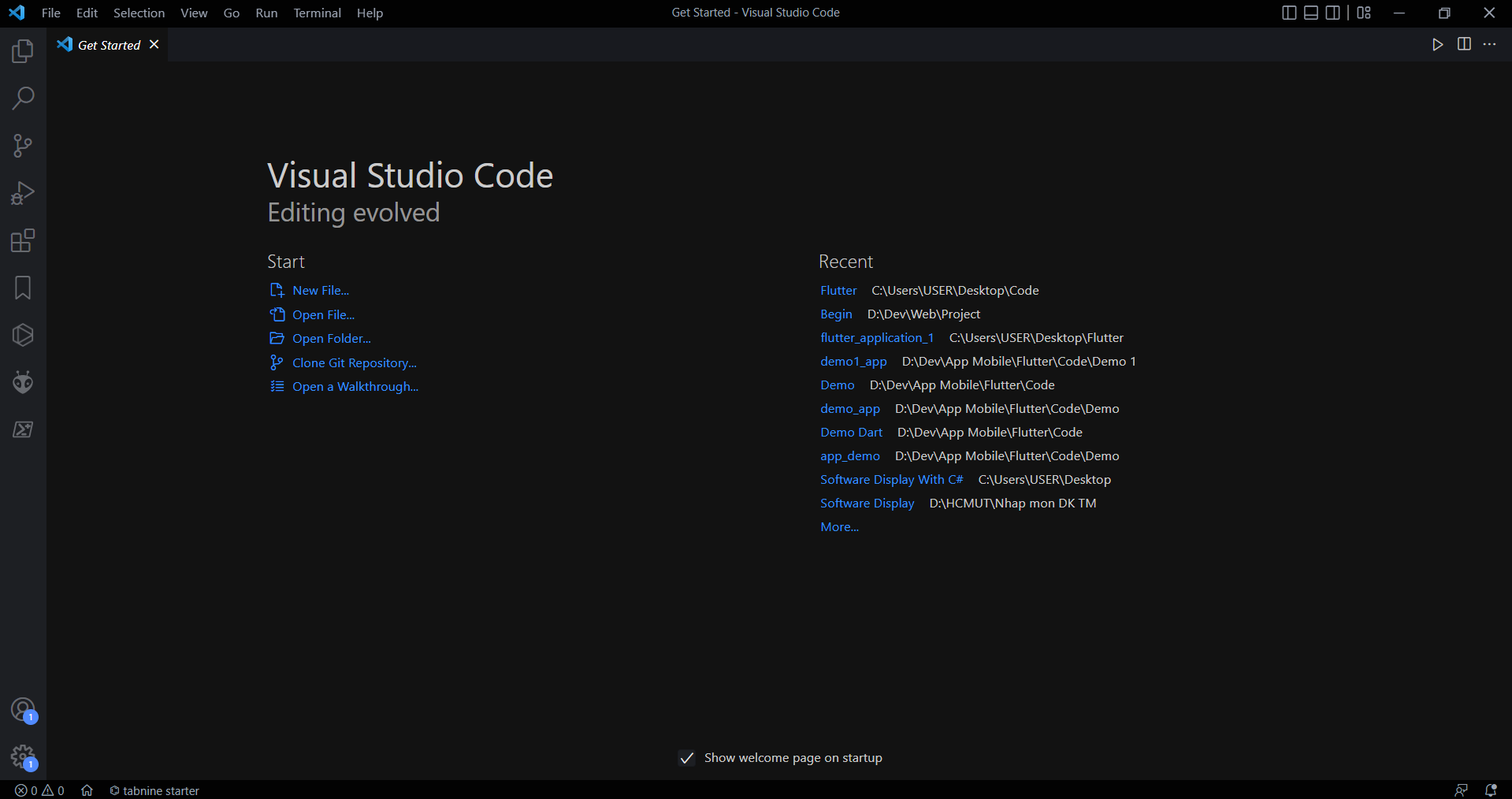


Hình 6: Giao diện phần mềm Arduino IDE

Môi trường lập trình đơn giản và dễ sử dụng, ngôn ngữ lập trình Wiring dễ hiểu và dựa trên nền tảng C/C++ rất quen thuộc với người làm kỹ thuật. Và quan trọng là số lượng thư viện code được viết sẵn và chia sẻ bởi cộng đồng nguồn mở là cực kỳ lớn. Arduino IDE là phần mềm dùng để lập trình cho Aruino Mega2560 và ESP8266. Môi trường lập trình có thể chạy trên cả 3 nền tảng phổ biến hiện nay: Windows, MacOS và Linux. Do có tính chất nguồn mở nên môi trường lập trình này hoàn toàn miễn phí và có thể mở rộng thêm bởi người dùng.

1. **Giới thiệu về Visual Studio Code và ngôn ngữ lập trình Dart:**

Visual Studio Code là một trình biên tập lập trình code miễn phí dành cho Windows, MacOS và Linux. Đây là trình biên dịch được phát triển bởi Microsoft. Nó được xem là một sự kết hợp hoàn hảo giữa IDE và Code Editor. Visual Studio Code hỗ trợ chức năng debug, đi kèm với Git, có syntax highlighting, tự hoàn thành mã thông minh, … Nhờ tính năng tùy chỉnh, Visual Studio Code cũng cho phép người dùng thay đổi theme, phím tắt, và các tùy chọn khác.



Hình 7: Giao diện Visual Studio Code

Dart là ngôn ngữ lập trình cho Flutter – bộ công cụ giao diện người dùng của Google để xay dựng các ứng dụng Mobile, Web và Desktop app đẹp, được biên dịch nguyên bản từ 1 cơ sở mã code duy nhất. Dart là ngôn ngữ hỗ trợ lập trình hướng đối tượng, cú pháp kiểu C, mã code Dart có thể biên dịch thành JavaSript để chạy trên trình duyệt. Từ ngôn ngữ lập trình Dar, Google giới thiệu SDK Flutter để sử dụng ngôn ngữ Dart phát triển ứng dụng di động chạy đa nền tảng (1 lần viết code build ra ứng dụng ISO và Android).

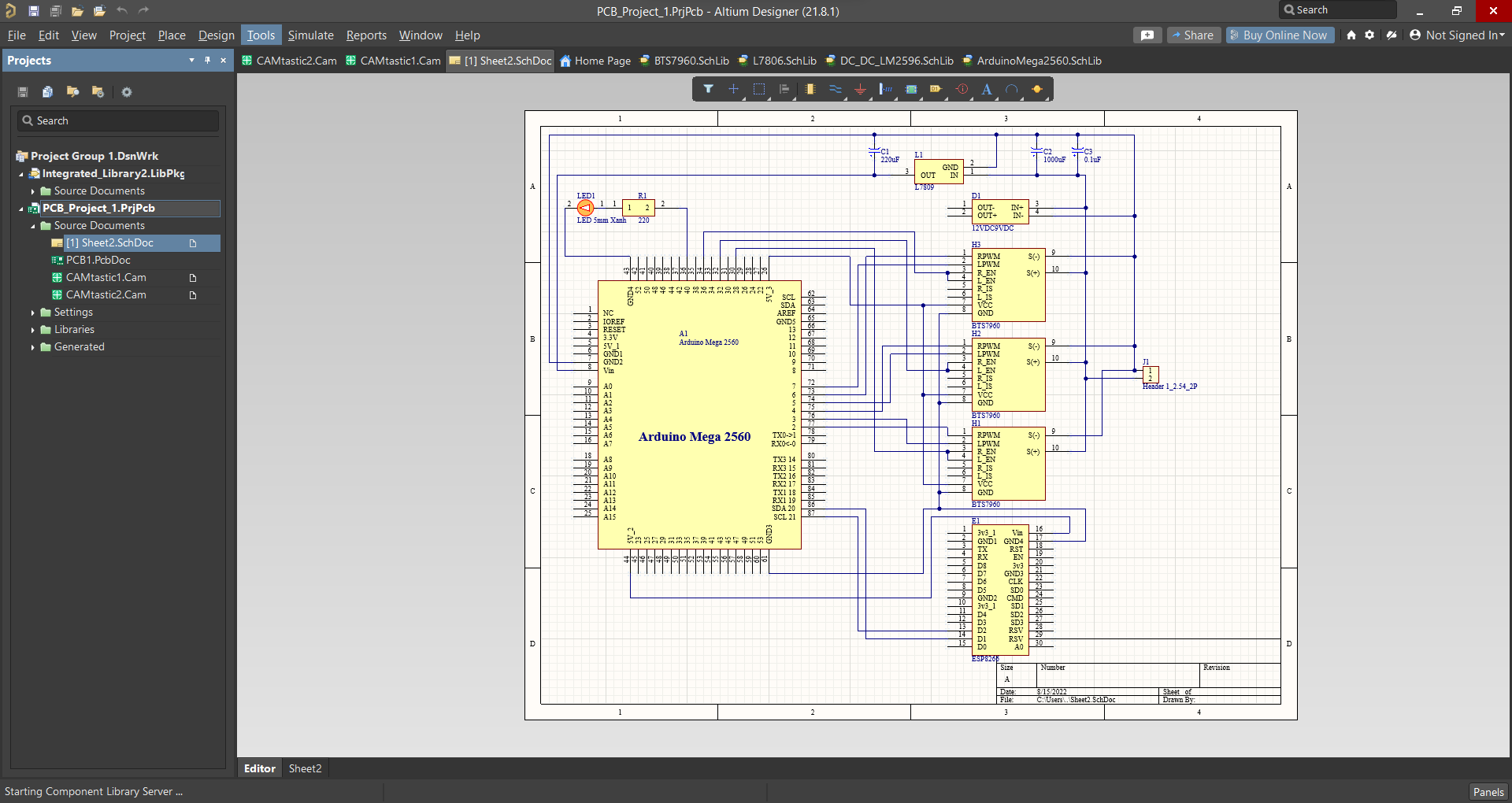


Hình 8: Flutter

1. **Giới thiệu về phần mềm thiết kế mạch Altium:**

Altium Designer là một trong những công cụ vẽ mạch điện tử mạnh nhất hiện nay. Được phát triển bởi hãng Altium Limited. Đây là phần mềm chuyển ngành được sử dụng trong thiết kế mạch điện tử. Nó là phần mềm mạch với nhiều tính năng thú vị, tuy nhiên phần mềm này còn được ít người biết đến so với các phần mềm khác.

Một số đặc trưng của phần mềm Altium Designer là: giao diện thiết kế, quản lý và chỉnh sửa thân thiện, dễ dàng biên dịch, quản lý file, quản lý phiên bản cho các tài liệu thiết kế. Hỗ trợ manh mẽ cho việc thiết kế tự động, đi dây theo thuật toán tối ưu, phân tích lắp ráp linh kiện, …

****

Hình 9: Giao diện Altium

# **CHƯƠNG 2: ĐẶC TẢ HỆ THỐNG**

1. **Nguyên lý hoạt động:**

Module Esp8266 sẽ như 1 trạm phát wifi (bao gồm id và password để truy cập vào Esp) nhận các tín hiệu điều khiển thông qua 1 app được thiết kế trên điện thoại.

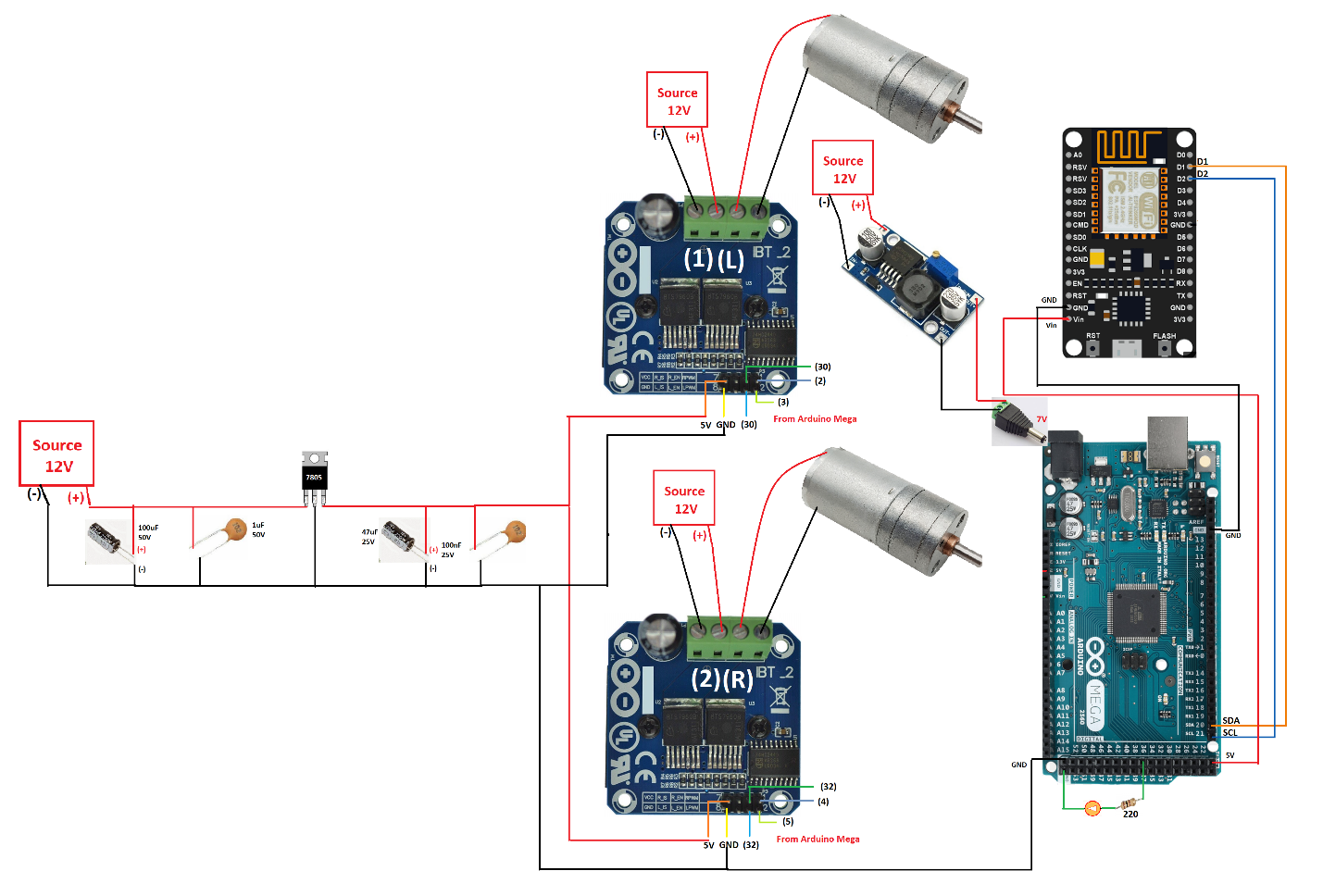
Module Esp8266 sẽ truyền dự liệu vừa nhận được từ app điện thoại đó và truyền cho Arduino Mega (thông qua giao thức I2C) để đưa các tín hiệu điều khiển mạch cầu H và động cơ.

Đồng thời khi nhận thành công tín hiện, đèn sẽ lần lượt On/Off để thông báo.

1. **Danh sách linh kiện cần thiết cho dự án:**

|  |  |
| --- | --- |
| **Tên linh kiện** | **Số lượng** |
| Node MCU Esp8266 | 1 |
| Arduino Mega | 1 |
| Mạch điều khiển động cơ DC BTS7960 | 2 |
| Động cơ DC servo giảm tốc hành tinh Planetary GP36 | 2 |
| Nguồn 12VDC | 1 |
| LED | 1 |
| Trở 220 ohm | 1 |
| IC L7805 | 1 |
| Tụ 1000uF-50V, 1uF-50V,  47uF-25V, 100nF-25V | 1 |
| Mạch DC – DC | 1 |

1. **Sơ đồ mạch:**

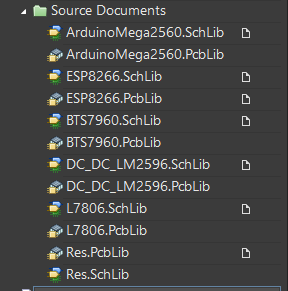


Hình 10: Sơ đồ mạch

**CHƯƠNG 3: THIẾT KẾ PHẦN CỨNG**

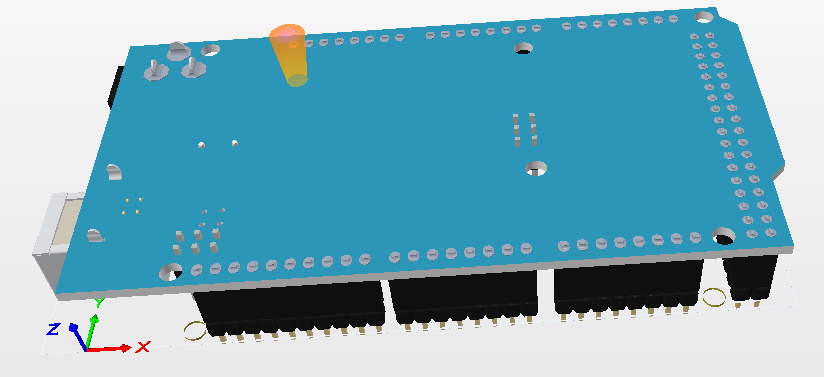
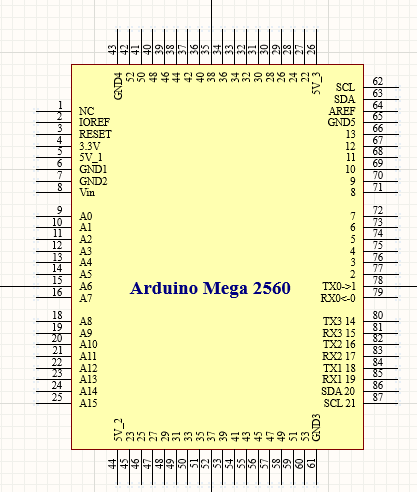
1. **Xây dựng sơ đồ nguyên lý trên phần mềm Altium:**

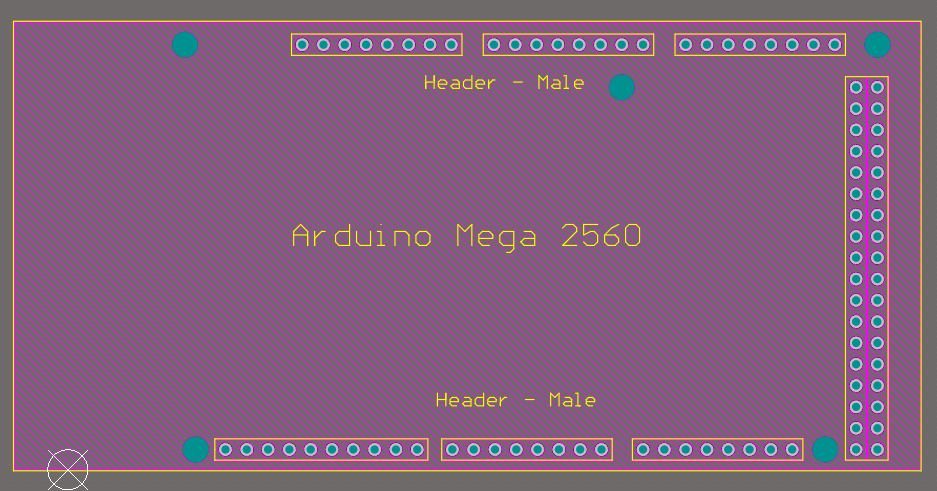
* Xây dựng thư viện của các linh kiện trong mạch:



Hình 11: List thư viện

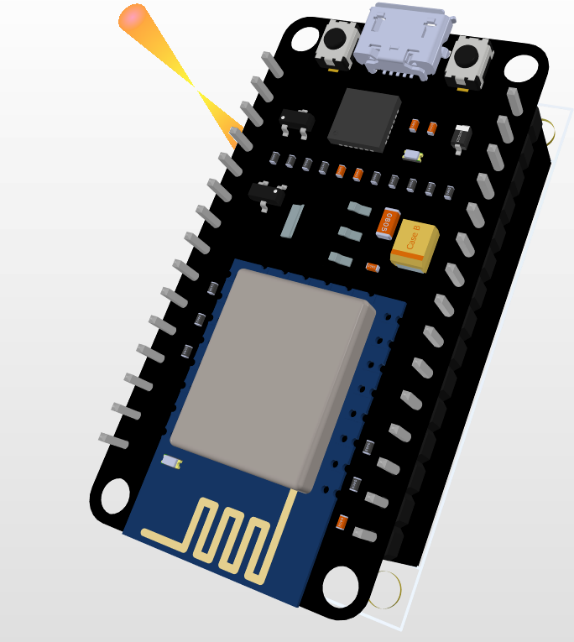
* + Arduino Mega2560:





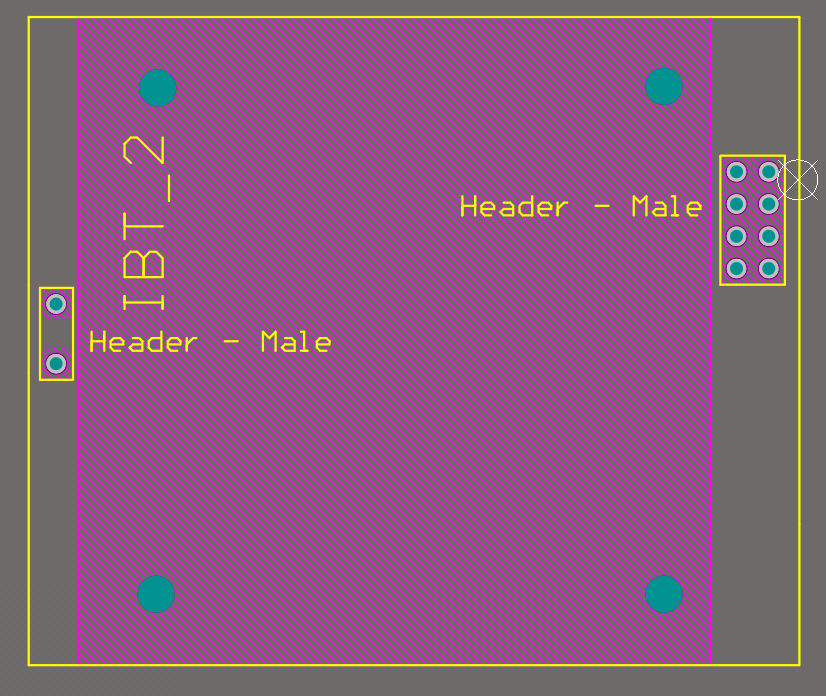
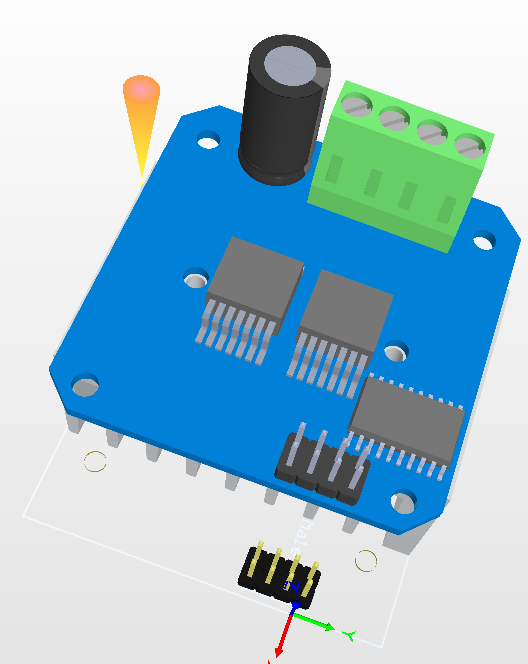
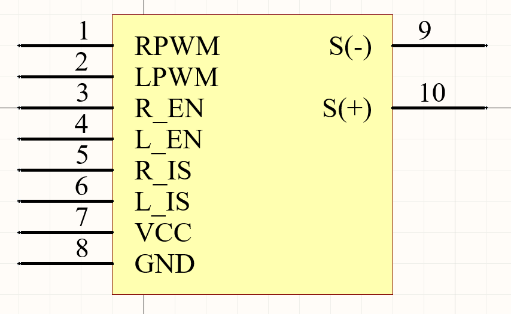
Hình 12: Thư viện Arduino Mega2560

* + Esp8266:



Hình 13: Thư viện ESP8266

* + BTS7960:

****

Hình 14: Thư viện BTS7960

* + Và một số linh kiện khác …
* Sơ đồ nguyên lý trên phần mềm Altium:

****

Hình 15: Sơ đồ nguyên lý trên phần mềm Altium

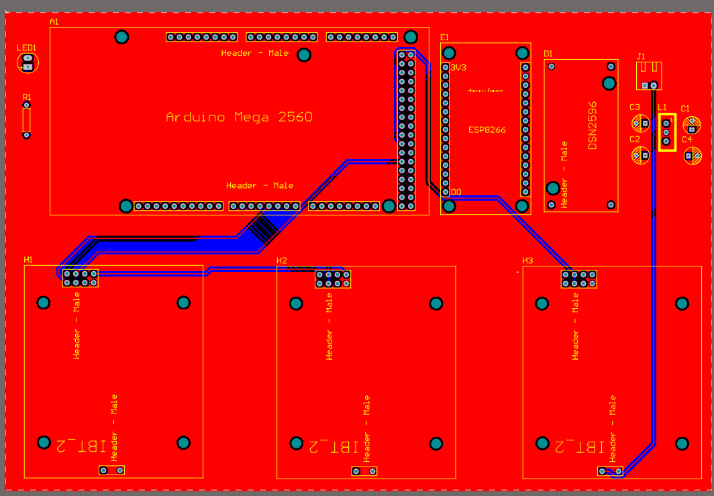
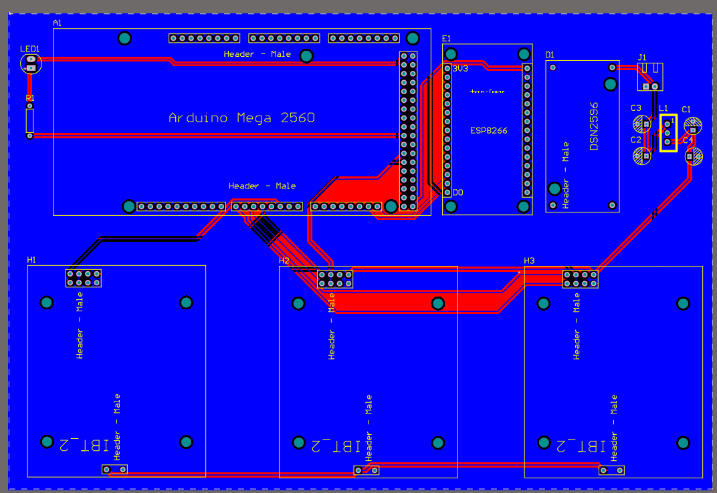
* + Nguồn tổng của toàn bộ mạch là nguồn 12V DC được chuyển từ 220V AC.
    - Với Arduino, sử dụng mach giảm áp LM2596 có thể điều chỉnh được mức điện áp ngõ ra từ 12V DC xuống 7V DC.
    - Với Esp, sử dụng chân nguồn 5V và GND từ Arduino cấp vào chân Vin và GND của Esp.
    - Với mạch cầu H, đi vào chân VCC là điện áp 5V nên ở đây sử dụng mạch giảm áp với IC L7805 và các tụ lọc. Mặc dù Arduino Mega có tích hợp chân 5V cấp cho 2 mạch cầu nhưng khi sử dụng chân 5V đó thì mạch giảm áp trên arduino rất nóng nên ở đây mạch cầu H sử dụng nguồn ngoài riêng.
  + Arduino Mega2560 kết nối với Esp8266 thông qua phương thức I2C với master là Arduino và slave là Esp.

1. **Xây dựng sơ đồ nguyên lý trên phần mềm Altium:**

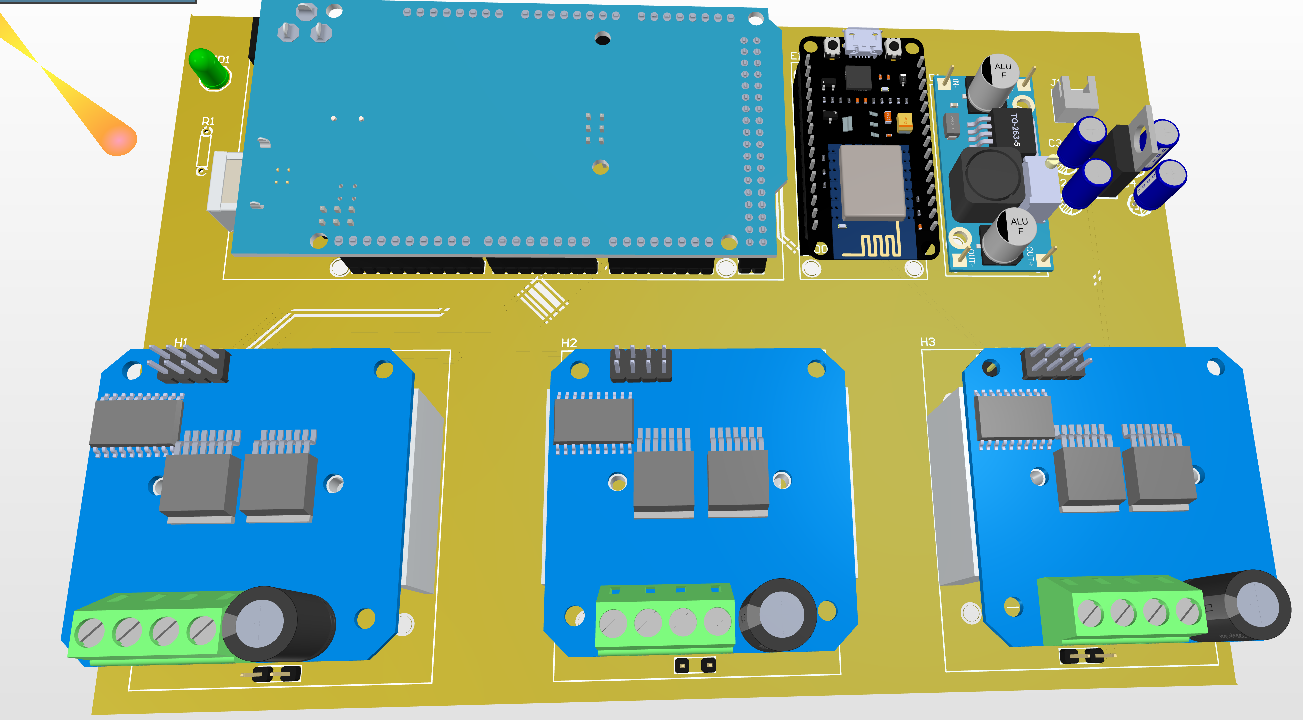
* Sơ đồ nối dây 2 mặt của mạch in PCB:

****

Hình 16: Sơ đồ nối dây PCB



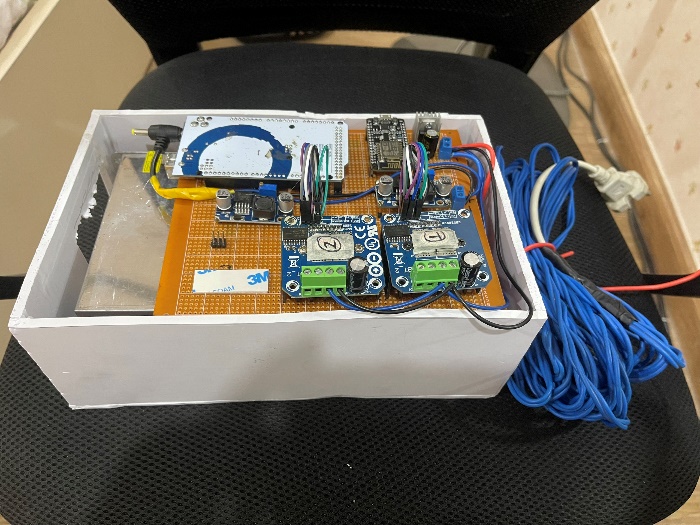
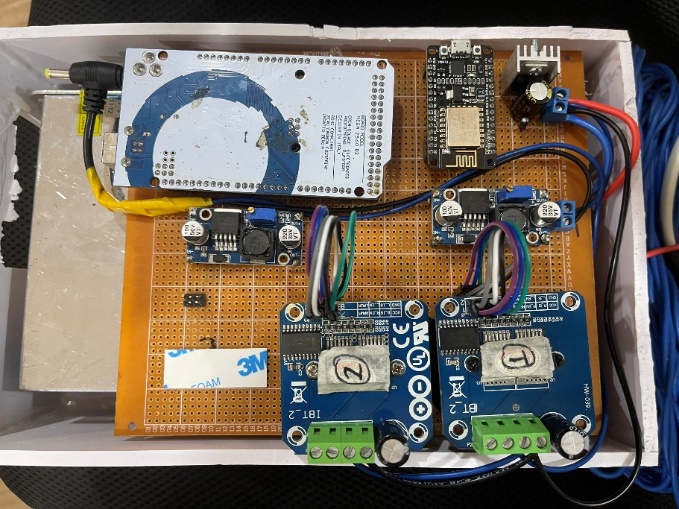
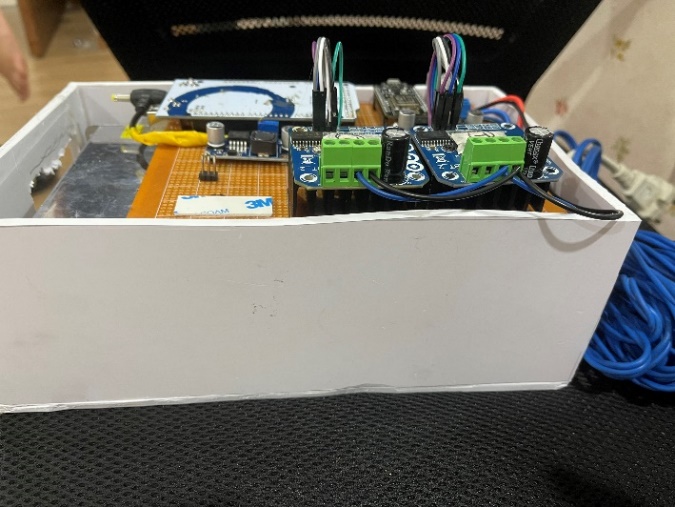
Hình 17: Mạch PCB phủ đồng 2 lớp



Hình 18: Hình 3D mạch

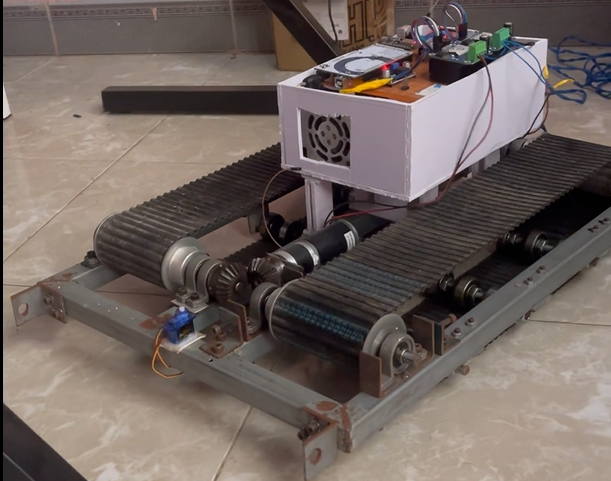
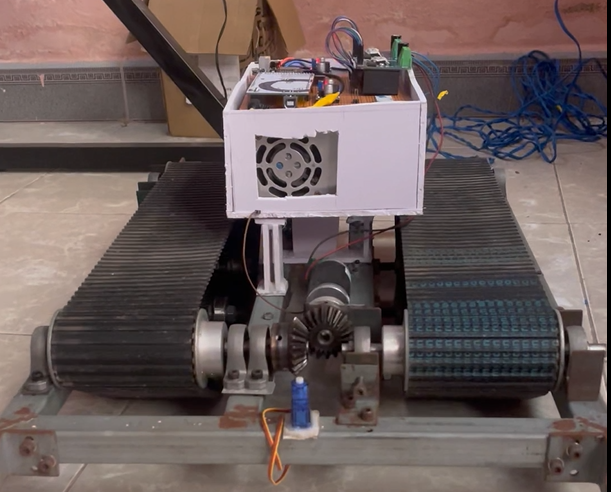
1. **Hoàn thiện sản phẩm:**

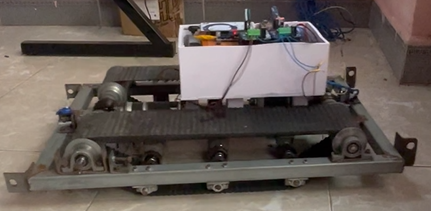
Do thời gian hoàn thiện gấp rút nên mạch hiện tại là mạch đi dây ở lớp dưới chứ chưa là mạch in như thiết kế ban đầu.



Hình 19: Mạch và nguổn của hệ thống

Hộp được làm bằng chất liệu formex nhằm đựng phần nguồn 12V DC nằm bên dưới và bên trên là board mạch.

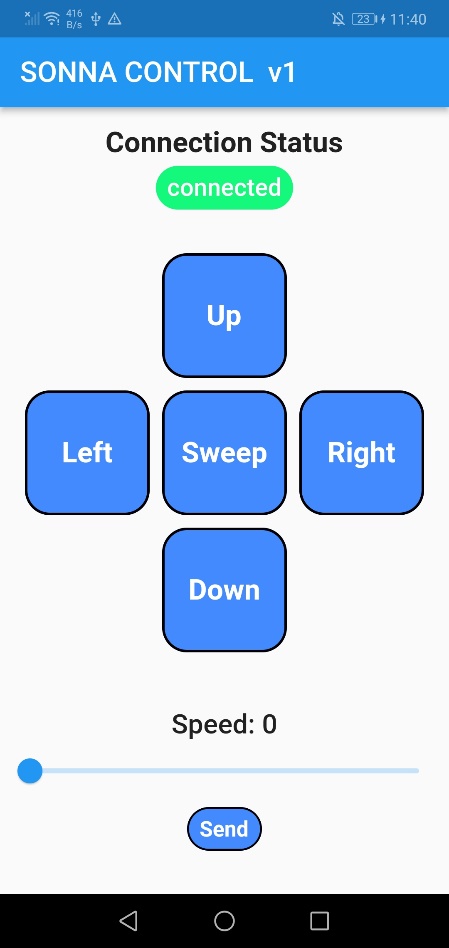
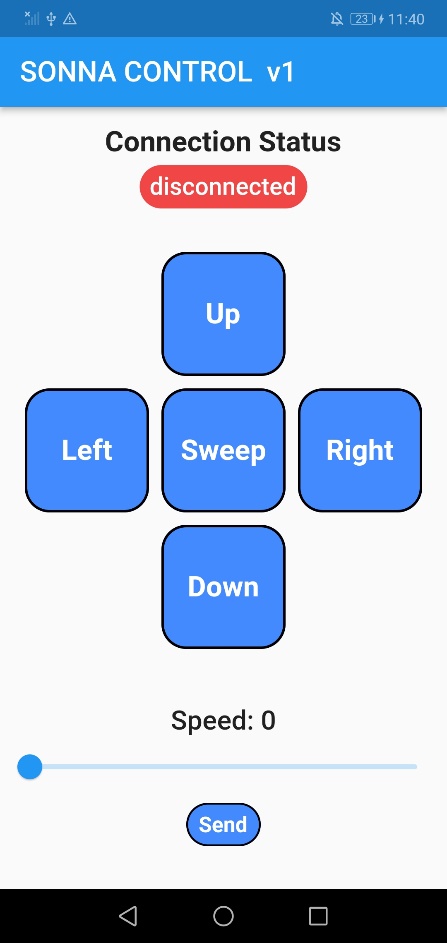




Hình 20: Mô hình thực tế

# **CHƯƠNG 4: THIẾT KẾ, LẬP TRÌNH PHẦN MỀM**

1. **Lập trình app điện thoại thông qua Flutter:**

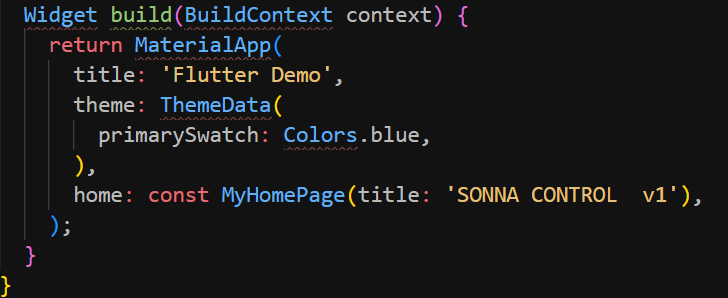
****

Hình 21: Giao diện app điện thoại

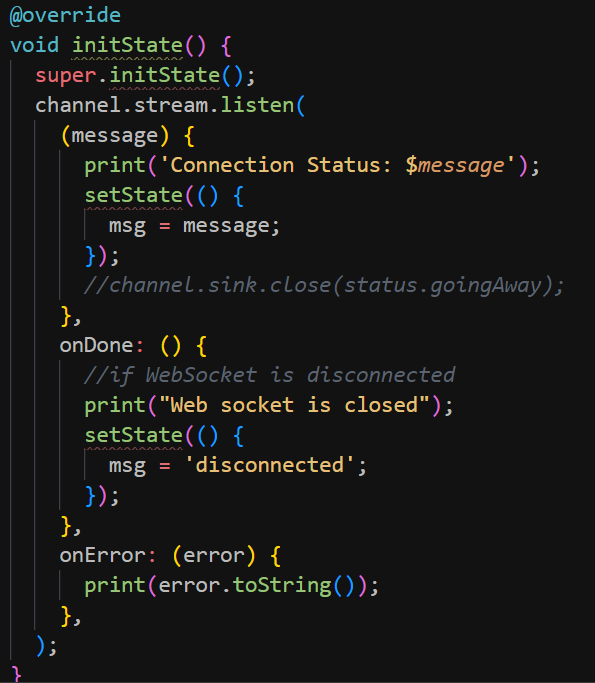
* Chứng năng:
  + Connection Status: hiển thị trạng thái kết nối của app với module điều khiển động cơ (để kết nối wifi lại nhấn nút ***disconnected***)

|  |  |  |  |
| --- | --- | --- | --- |
| **Nút nhấn** | **Nhấn xuống** | **Nhả ra** | **Chức năng** |
| "Up" | Gửi kí tự 'f' cho Esp | Gửi kí tự ‘s’ cho Esp | Robot đi thẳng |
| "Down" | Gửi kí tự 'b' cho Esp | Robot đi lùi |
| "Left" | Gửi kí tự 'l' cho Esp | Robot rẽ trái |
| "Right" | Gửi kí tự 'r' cho Esp | Robot rẽ phải |

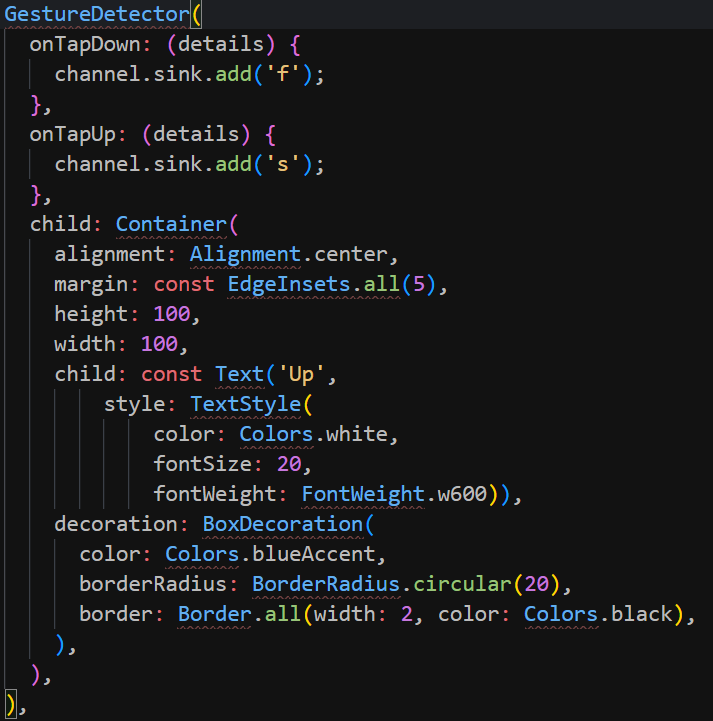
* + Nút nhấn “Sweep”: bật/tắt động cơ quét (gửi dữ liệu ‘o’/’m’ cho esp)
  + Speed: lựa chọn tốc độ cho robot
  + Nút nhấn “Send”: gửi dữ liệu cho esp dưới dạng: “p” + number
* Giải thích code:



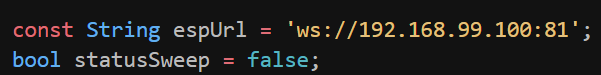
* + Khởi tạo theme cho ứng dụng bao gồm: màu sắc, tên.

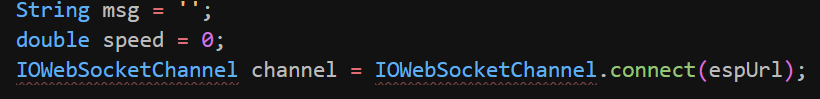


* + Hiển thị trạng thái kết nối được hay không với ESP8266 thông qua wifi

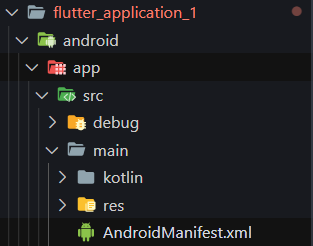


* + Cấu trúc nút nhấn bao gồm: cấu tạo, màu sắc, khi nhấn xuống thì gửi kí tự ‘f’ còn khi nhả ra thì gửi kí tự ‘s’.
  + Nút nhấn ở đây vẫn còn điểm hạn chế đó là khi người dùng giữ nhưng lại di chuyển sang vị trí khác trên màn hình thì khi nhả ra sẽ không gửi được kí tự ‘s’.



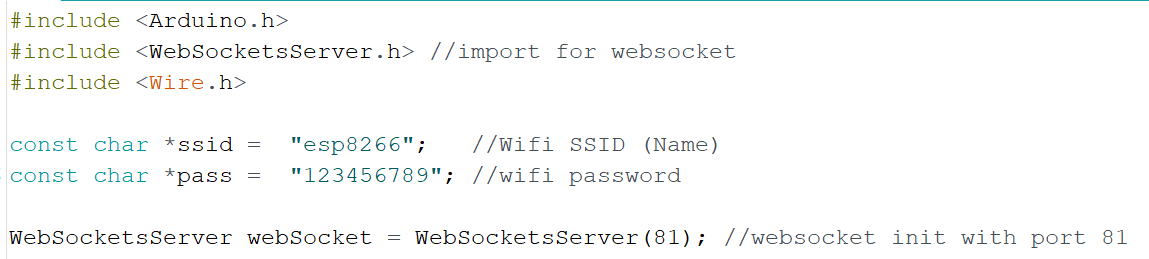


* + Cài đặt địa chỉ để ứng dụng có thể kết nối với esp8266 thông qua wifi

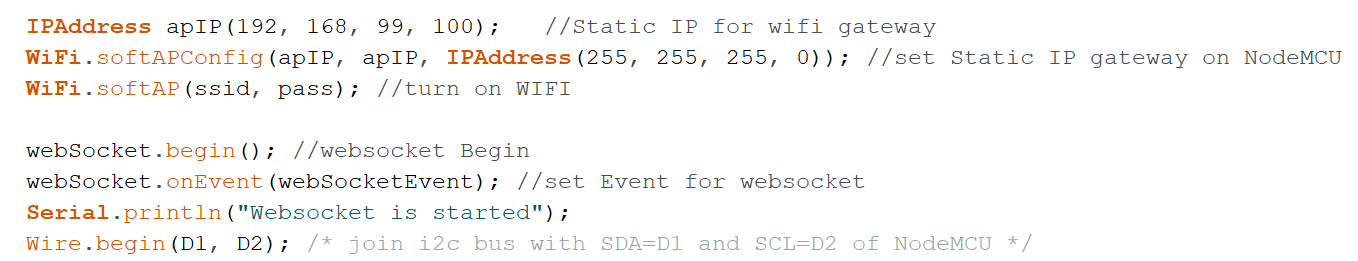
 Sau khi code xong để có thể release được phải thêm 2 dòng code vào file AndroidManifest.xml để có thể release được ứng dụng.



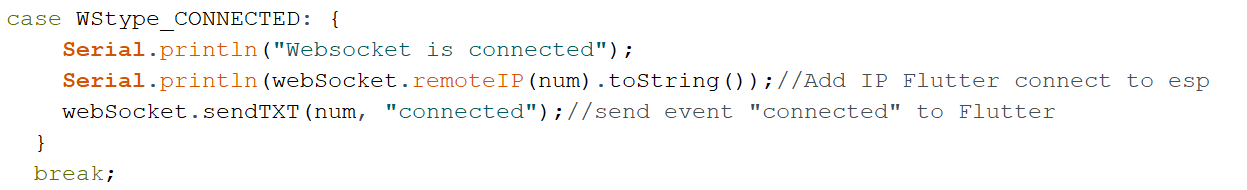
1. **Lập trình Esp8266:**



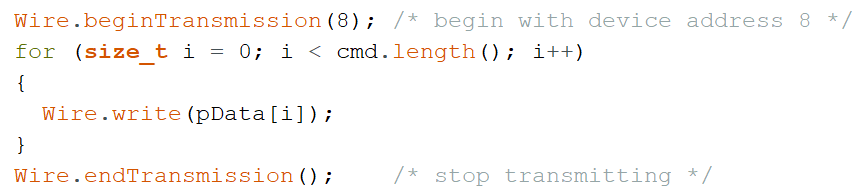
* Cài đặt các thư viện cho giao thức I2C, websocket và tên, mật khẩu để esp phát wifi.



* Cặt đặt nhận dự kiện từ app và kết nối chân I2C của Esp (D1, D2).

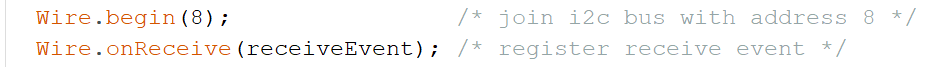


* Khi kết nối thành công, esp gửi data “connected” lên app điện thoại.

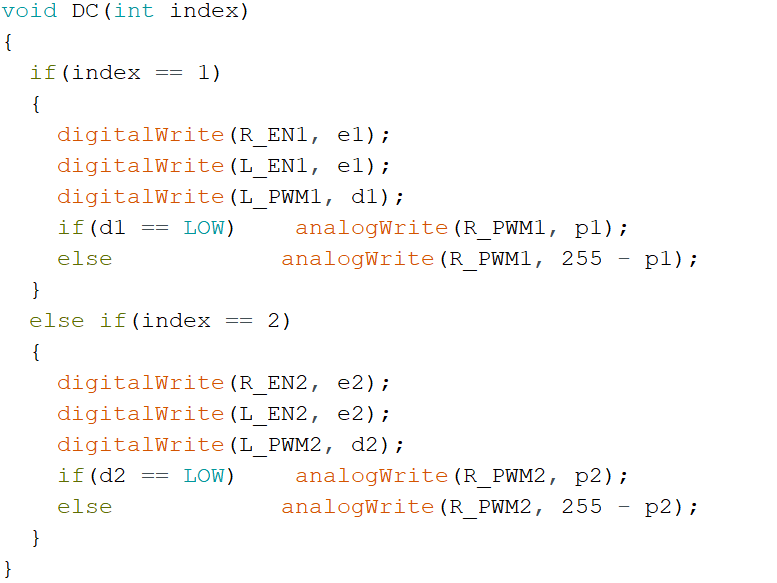


* Gửi data qua I2C với Esp là Master và Arduino là Slave với địa chỉ là (8).

1. **Lập trình Arduino Mega:**



* Cài đặt Arduino là Slave với địa chỉ là (8) và ở chế độ nhận data từ Master.



* Hàm điều khiển động cơ DC với index (có giá trị là 1 và 2) nhằm điều khiển động cơ thứ 1 hoặc thứ 2. Các biến: e1, e2 – (có giá trị: ENABLE, DISABLE – nhằm cho phép quay hay không quay); d1, d2 – (có giá trị: LOW, HIGH – điều khiển động cơ theo chiều thuận hay nghịch với chiều kim đồng hồ); p1,p2 – giá trị tốc độ của động cơ.