

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH
KHOA ĐIỆN – ĐIỆN TỬ
BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG

NGUYỄN TUẤN - 1513831

LÊ THỊ MỸ DUYÊN - 1510525

LUẬN VĂN TỐT NGHIỆP

ROBOT HÚT BỤI TỰ ĐỘNG

ROBOTIC VACUUM CLEANER

KỸ SƯ NGÀNH KỸ THUẬT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA

GIẢNG VIÊN HƯỚNG DẪN

KS.BÙI THANH HUYỀN

TP. HỒ CHÍ MINH, 2019

TRƯỜNG ĐH BÁCH KHOA
KHOA ĐIỆN - ĐIỆN TỬ
BỘ MÔN: ĐIỀU KHIỂN TỰ ĐỘNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc Lập – Tự Do – Hạnh Phúc
--o0o--

TP.HCM, ngày ... tháng ... năm 2019

PHIẾU CHẤM BẢO VỆ LVTN
(Dành cho cán bộ hướng dẫn)

Họ và tên:

NGUYỄN TUẤN

MSSV: **1513831**

Họ và tên:

LÊ THỊ MỸ DUYÊN

MSSV: **1510525**

Ngành:

TỰ ĐỘNG HÓA

1. Đề tài: "**Robot hút bụi tự động**"

2. Họ tên người phản biện:

3. Tổng quát về bản thuyết trình:

Số trang	Số chương
Số bảng số liệu	Số hình vẽ
Số tài liệu tham khảo	Phần mềm tính toán

4. Những ưu điểm chính của LVTN:

5. Những thiếu sót chính của LVTN:

6. Về thái độ làm việc của sinh viên:

7. Đề nghị: Được bảo vệ ,
Không được bảo vệ .
Bổ sung thêm để bảo vệ ,

8. Đánh giá chung (*bằng chữ: Giỏi, Khá, TB*): Điểm/**10**

Người nhận xét
(Ký tên và ghi rõ họ tên)

TRƯỜNG ĐH BÁCH KHOA
KHOA ĐIỆN - ĐIỆN TỬ
BỘ MÔN: ĐIỀU KHIỂN TỰ ĐỘNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc Lập - Tự Do - Hạnh Phúc
--o0o--

TP.HCM, ngày ... tháng ... năm 2019

PHIẾU CHẤM BẢO VỆ LVTN
(Dành cho cán bộ phản biện)

Họ và tên:

NGUYỄN TUẤN

MSSV: **1513831**

Họ và tên:

LÊ THỊ MỸ DUYÊN

MSSV: **1510525**

Ngành:

TỰ ĐỘNG HÓA

1. Đề tài: "**Robot hút bụi tự động**"

2. Họ tên người phản biện:

3. Tổng quát về bản thuyết trình:

Số trang

Số chương

Số bảng số liệu

Số hình vẽ

Số tài liệu tham khảo

Phần mềm tính toán

4. Những ưu điểm chính của LVTN:

5. Những thiếu sót chính của LVTN:

6. Về thái độ làm việc của sinh viên:

7. Đề nghị: Được bảo vệ ,
Không được bảo vệ .

Bổ sung thêm để bảo vệ ,

8. Đánh giá chung (*bằng chữ: Giỏi, Khá, TB*): Điểm/**10**

Người nhận xét
(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Lời đầu tiên, chúng em xin gửi đến Cô KS. Bùi Thanh Huyền lời cảm ơn chân thành và sâu sắc nhất. Nhờ có sự hướng dẫn và giúp đỡ của Cô trong suốt thời gian qua, chúng em đã có thể thực hiện và hoàn thành Đồ Án Môn Học, Thực Tập Tốt Nghiệp và Luận Văn Tốt Nghiệp. Với những lời nhận xét, góp ý và hướng dẫn tận tình, Cô đã giúp chúng em có một định hướng đúng đắn trong suốt quá trình thực hiện Đề tài, cũng như nhìn ra được những ưu, khuyết điểm của sản phẩm và từng bước hoàn thiện hơn.

Đồng thời, chúng em xin trân trọng cảm ơn các Thầy Cô của Trường Đại Học Bách Khoa nói chung và của khoa Điện - Điện Tử nói riêng đã dạy dỗ và truyền đạt cho chúng em những bài học vô cùng bổ ích suốt quãng thời gian ngồi trên ghế giảng đường Đại học. Những lời giảng của Thầy Cô trên bục giảng đã trang bị cho chúng em nhiều kiến thức cần thiết và giúp chúng em có thể tích lũy thêm kinh nghiệm cho tương lai sau này.

Bên cạnh đó, chúng tôi xin cảm ơn sự hỗ trợ, giúp đỡ của bạn bè và sự ủng hộ của gia đình- những người thân luôn là chỗ dựa tinh thần vững chắc cho chúng em trong thời gian học tập tại Trường Đại Học Bách Khoa và trong quá trình hoàn thành Luận Văn Tốt Nghiệp này.

Trong thời gian thực hiện Luận Văn Tốt Nghiệp, mặc dù có nhiều sự cố gắng nhưng chắc chắn đề tài sẽ không tránh khỏi sự thiếu sót. Kính mong Thầy Cô chỉ bảo và góp ý kiến để đề tài được hoàn thiện hơn.

Cuối cùng, kính chúc quý Thầy Cô và gia đình luôn dồi dào sức khỏe, niềm vui và hạnh phúc trong công việc và cuộc sống. Xin chân thành cảm ơn!

Hồ Chí Minh, ngày tháng năm 2019
Sinh viên thực hiện

NGUYỄN TUẤN & LÊ THỊ MỸ DUYÊN

TRƯỜNG ĐH BÁCH KHOA
KHOA ĐIỆN - ĐIỆN TỬ
BỘ MÔN: ĐIỀU KHIỂN TỰ ĐỘNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc Lập - Tự Do - Hạnh Phúc
--o0o--

TP.HCM, ngày... tháng... năm 2019

ĐỀ CƯƠNG CHI TIẾT

TÊN ĐỀ TÀI: ROBOT HÚT BỤI TỰ ĐỘNG

Cán bộ hướng dẫn: KS. Bùi Thanh Huyền

Thời gian thực hiện: từ ngày 15/1/2019 đến ngày 7/6/2019

Sinh viên thực hiện: NGUYỄN TUẤN – 1513831

LÊ THỊ MỸ DUYÊN – 1510525

Nội dung đề tài:

Xây dựng mô hình robot hút bụi có hai chế độ hoạt động là bằng tay và tự động, có khả năng định hướng di chuyển tránh vật cản và rơi rớt. Tích hợp thêm các chức năng về thời gian thực, giám sát pin và kết nối wifi.

Thiết kế mô hình cơ khí cho khung robot tương tự với các sản phẩm trong thực tế, mang đến cái nhìn trực quan về sản phẩm.

Xây dựng giao diện người dùng để điều khiển robot từ máy tính và nhận các thông tin được truyền về qua kết nối wifi.

Kết quả mong đợi:

- ❖ Xây dựng thành công mô hình máy hút bụi.
- ❖ Điều khiển máy hút bụi tự động di chuyển với các thông số được cài đặt online từ giao diện người dùng.
- ❖ Điều khiển máy hút bụi vận hành bằng tay theo sự quan sát của người dùng.

Kế hoạch thực hiện:

TT	Công việc	Thời gian thực hiện
1	_Tìm hiểu cách sử dụng module RTC DS3231, truyền UART không dây dùng ESP8266	15/1-29/1

2	_Lập trình sơ bộ giao diện người dùng	30/1-13/2
3	_Lập trình PID điều khiển động cơ, sử dụng ADC đọc giá trị điện áp của nguồn pin _Lập trình ứng dụng các module DS3231, ESP8266 vào robot	14/2-6/3
4	_Tìm hiểu cách sử dụng và thiết kế mô hình robot sử dụng phần mềm 3D Soliworks _Lập trình một số tính năng trên giao diện người dùng	7/3-23/3
5	_Viết báo cáo các phần làm được chuẩn bị cho báo cáo 50% _Tổng hợp và chỉnh sửa các phần của code	24/3-10/4
6	_Tìm hiểu các phương pháp điều hướng robot _Thiết kế mạch phân áp đọc giá trị pin	11/4-29/4
7	_Lập trình giải thuật di chuyển cho robot _Lắp đặt, chỉnh sửa mô hình robot _Báo cáo các phần đã làm được	1/5-14/5
8	_Thực nghiệm và đánh giá kết quả _Tinh chỉnh thông số, chỉnh sửa code _Viết báo cáo	15/5-31/5
9	_Chỉnh sửa nội dung luận văn _Chuẩn bị cho báo cáo luận văn sắp tới	1/6-7/6

Xác nhận của Cán bộ hướng dẫn

TP. HCM, ngày....thángnăm.....

Sinh viên thực hiện

DANH SÁCH HỘI ĐỒNG BẢO VỆ LUẬN VĂN

Hội đồng chấm luận văn tốt nghiệp, thành lập theo Quyết định số
ngày của Hiệu trưởng Trường Đại học Bách khoa TP.HCM.

1. – Chủ tịch.
2. – Thư ký.
3. – Ủy viên.
4. – Ủy viên.

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN ĐỀ TÀI	2
1.1. Đặt vấn đề	3
1.2. Giới thiệu một số robot hút bụi ngày nay.....	4
1.3. Phạm vi đề tài và phương pháp thực hiện.....	6
CHƯƠNG 2: TÌM HIỂU MỘT SỐ PHƯƠNG PHÁP VẬN HÀNH DI CHUYÊN	9
2.1. Phương pháp di chuyển tự động.....	10
2.2. Phương pháp di chuyển ZigZag	12
2.3. Phương pháp di chuyển xoắn ốc	13
CHƯƠNG 3: GIỚI THIỆU MỘT SỐ MODULE SỬ DỤNG TRONG ROBOT	16
3.1. Vi điều khiển trung tâm STM32F407.....	17
3.2. Phát hiện vật cản, vùng chênh lệch độ cao tránh rơi rớt	22
3.2.1 Module cảm biến siêu âm SRF-04	22
3.2.2 Module cảm biến hồng ngoại.....	24
3.3. Động cơ và module điều khiển tốc độ.....	25
3.3.1 Động cơ encoder	25
3.3.2 Mạch lái động cơ L298.....	26
3.4. Module thời gian thực – RTC	28
3.5. Module kết nối không dây truyền dữ liệu UART.....	29
CHƯƠNG 4: THIẾT KẾ MÔ HÌNH PHẦN CƠ KHÍ.....	31
4.1 Giới thiệu phần mềm Soliworks.....	32
4.2 Giới thiệu mô hình robot.....	33
4.3 Thiết kế phần thân robot.....	34
4.4 Thiết kế phần đế robot.....	37
4.5 Thiết kế hộp bụi.....	40
CHƯƠNG 5: LẬP TRÌNH ROBOT	42
5.1 Sơ đồ khối và sơ đồ nguyên lý.....	43
5.1.1 Sơ đồ khối.....	43
5.1.2 Sơ đồ nguyên lý.....	45
5.2 Phương pháp điều chế PWM.....	46

5.2.1	<i>Giới thiệu thuật toán</i>	46
5.2.2	<i>Ứng dụng vào robot</i>	47
5.3	<i>Thuật toán PID</i>	48
5.3.1	<i>Giới thiệu thuật toán</i>	48
5.3.2	<i>Ứng dụng PID vào ổn định hướng đi</i>	52
5.4	<i>Thuật toán tránh vật cản, vùng chênh lệch độ cao</i>	56
5.4.1	<i>Giới thiệu thuật toán</i>	56
5.4.2	<i>Ứng dụng vào robot</i>	56
5.5	<i>Phương pháp truyền UART qua kết nối không dây</i>	59
5.5.1	<i>Giới thiệu phương pháp truyền UART không dây</i>	59
5.5.2	<i>Ứng dụng vào điều khiển và theo dõi robot</i>	59
5.6	<i>Xác định giá trị pin</i>	63
5.6.1	<i>Giới thiệu ADC</i>	63
5.6.2	<i>Sử dụng ADC vào đọc giá trị nguồn cấp</i>	63
5.7	<i>Lập trình đọc giá trị thời gian thực</i>	64
5.7.1	<i>Giới thiệu hệ thống thời gian thực</i>	64
5.7.2	<i>Sử dụng module thời gian thực cho robot</i>	65
5.8	<i>Thuật toán di chuyển của robot</i>	67
5.8.1	<i>Giới thiệu thuật toán di chuyển</i>	67
5.8.2	<i>Ứng dụng thuật toán</i>	67
5.9	<i>Giải thuật hệ thống</i>	71

CHƯƠNG 6: THIẾT KẾ GIAO DIỆN NGƯỜI DÙNG BẰNG NGÔN NGỮ C#74

6.1	<i>Đặt vấn đề</i>	75
6.2	<i>Giới thiệu giao diện người dùng (UI)</i>	75
6.3	<i>Giới thiệu chương trình Visual Studio</i>	77
6.4	<i>Các tính năng chính của giao diện</i>	80
6.4.1	<i>Kết nối thiết bị</i>	81
6.4.2	<i>Cài đặt cho robot</i>	84
6.4.3	<i>Theo dõi trạng thái của robot</i>	88
6.4.4	<i>Ghi chép quá trình làm việc</i>	91

CHƯƠNG 7: THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ95

7.1	Kết quả đạt được.....	96
7.1.1	<i>Phần cơ khí và mạch điều khiển.....</i>	96
7.1.2	<i>Phần lập trình vi điều khiển.....</i>	99
7.1.3	<i>Phần lập trình ứng dụng.....</i>	100
7.2	Thực nghiệm quá trình hoạt động của robot.....	100
7.2.1	<i>Quá trình truyền nhận không dây.....</i>	100
7.2.2	<i>Khả năng di chuyển và tránh vật cản</i>	102
7.2.3	<i>Khả năng hút bụi của robot</i>	103
7.3	Nhận xét và đánh giá kết quả thực nghiệm	104
7.3.1	<i>Tính hiệu quả.....</i>	104
7.3.2	<i>Nguồn pin và giá thành</i>	105
7.4	Một số điểm cần khắc phục	107
CHƯƠNG 8: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN		108
8.1	<i>Kết luận đề tài.....</i>	109
8.2	<i>Hướng phát triển.....</i>	109
TÀI LIỆU THAM KHẢO		110
<i>Phụ lục: Cách nạp code module wifi ESP dùng USB UART</i>		112

DANH MỤC HÌNH ẢNH

Chương 1:

Hình 1. 1 Một số sản phẩm Botvac	4
Hình 1. 2. Robot hút bụi Deebot	5

Chương 2:

Hình 2. 1 Phương pháp di chuyển tự động	10
Hình 2. 2 Thuật toán di chuyển tổng hợp	11
Hình 2. 3 Thuật toán di chuyển zigzag	12
Hình 2. 4 Thuật toán di chuyển xoắn ốc	13

Chương 3:

Hình 3. 1. Sơ đồ biểu diễn xung PWM	19
Hình 3. 2. Sơ đồ hoạt động I2C	21
Hình 3. 3. Sơ đồ truyền nhận I2C	22
Hình 3. 4. Cảm biến siêu âm	22
Hình 3. 5. Nguyên lý đo khoảng cách bằng truyền âm	23
Hình 3. 6. Cách sử dụng cảm biến siêu âm	24
Hình 3. 7. Cảm biến hồng ngoại	24
Hình 3. 8. Cấu tạo động cơ	26
Hình 3. 9. Mạch lái LH298	27
Hình 3. 10. Module thời gian thực DS3231	28
Hình 3. 11. Module wifi ESP8266	29
Hình 3. 12. Sơ đồ dây của ESP8266	30

Chương 4:

Hình 4.1. Phần mềm Solidworks	32
Hình 4.2. Phần thân robot	34
Hình 4.3. Phần thân ở góc nhìn từ trên xuống	35

<i>Hình 4.4. Thực tế phần thân ở góc nhìn từ bên trái</i>	35
<i>Hình 4.5. Cách bố trí cảm biến trên thân</i>	36
<i>Hình 4.6. Thực tế cách bố trí cảm biến trên thân</i>	36
<i>Hình 4.7. Thực tế phần thân ở góc nhìn từ trên xuống</i>	37
<i>Hình 4.8. Phần đế của robot</i>	38
<i>Hình 4.9. Thực tế mặt dưới của robot</i>	39
<i>Hình 4.10. Phần đế ở góc nhìn từ phía sau</i>	40
<i>Hình 4.11. Phần thiết kế chứa bụi của robot</i>	40
<i>Hình 4.12. Khoang chứa bụi</i>	41
<i>Hình 4.13. Thực tế hộp bụi</i>	41

Chương 5:

<i>Hình 5.1. Sơ đồ khối hệ thống</i>	44
<i>Hình 5.2. Sơ đồ nguyên lý</i>	45
<i>Hình 5.3. Sơ đồ nối dây động cơ.....</i>	48
<i>Hình 5.4. Sơ đồ liên hệ</i>	49
<i>Hình 5.5. Sơ đồ bộ điều khiển PID</i>	49
<i>Hình 5.6. Đáp ứng của khâu tỷ lệ</i>	50
<i>Hình 5.7. Đáp ứng của khâu tỷ lệ</i>	51
<i>Hình 5.8. Đáp ứng khâu vi phân</i>	51
<i>Hình 5.9. Sơ đồ phần cứng sử dụng PID</i>	53
<i>Hình 5.10. Hình ảnh mô phỏng cho thuật toán Master-Slave</i>	54
<i>Hình 5.11. Sơ đồ giải thuật đếm xung của động cơ phái</i>	55
<i>Hình 5.12. Giải đố xung encoder</i>	56
<i>Hình 5.13. Sơ đồ thuật toán cảm biến siêu âm</i>	57
<i>Hình 5.14. Sơ đồ thuật toán cảm biến hồng ngoại</i>	58
<i>Hình 5.15. Sơ đồ thuật toán encoder</i>	58
<i>Hình 5.16. Sơ đồ kết nối không dây</i>	60

<i>Hình 5.17. Sơ đồ giải thuật client</i>	61
<i>Hình 5.18. Sơ đồ giải thuật Server.....</i>	62
<i>Hình 5.19. Sơ đồ nguyên lý mạch chia áp.....</i>	64
<i>Hình 5.20. Datasheet của DS3231</i>	65
<i>Hình 5.21. Lưu đồ giải thuật lập trình sử dụng DS3231</i>	66
<i>Hình 5.22. Di chuyển zigzag.....</i>	68
<i>Hình 5.23. Lưu đồ giải thuật tìm tường nhà</i>	69
<i>Hình 5.24. Lưu đồ giải thuật zigzag</i>	70
<i>Hình 5.25. Lưu đồ giải thuật chương trình chính của vi điều khiển</i>	72
<i>Hình 5.26. Lưu đồ giải thuật phân tích chuỗi dữ liệu</i>	73

Chương 6:

<i>Hình 6. 1. Ví dụ điển hình về UI.....</i>	76
<i>Hình 6. 2. Chương trình Visual Studio 2017.....</i>	77
<i>Hình 6. 3. Giao diện của chương trình</i>	78
<i>Hình 6. 4. Giao diện chính của ứng dụng</i>	80
<i>Hình 6. 5. Giao diện kết nối của ứng dụng</i>	83
<i>Hình 6. 6. Giao diện cài đặt cho robot</i>	84
<i>Hình 6. 7. Giao diện cài đặt sau khi nhấn SET UP</i>	87
<i>Hình 6. 8. Giao diện bộ điều khiển ở 2 chế độ Auto & Manual</i>	88
<i>Hình 6. 9. Giao diện trạng thái trong TH1</i>	90
<i>Hình 6. 10. Giao diện trạng thái trong TH2</i>	90
<i>Hình 6. 11. Giao diện hiển thị quá trình làm việc</i>	93
<i>Hình 6. 12. Giao diện khi nhấn Reset</i>	93

Chương 7

<i>Hình 7.1 Nắp và đế robot thực tế</i>	96
<i>Hình 7.2 Thân phía bên trái robot thực tế</i>	96
<i>Hình 7.3 Thân phía trước robot thực tế</i>	97

<i>Hình 7.4 Lắp linh kiện phía trong robot</i>	98
<i>Hình 7.5 Tổng quát phía trong thân robot</i>	98
<i>Hình 7.6 Giao diện kết nối thành công và hiển thị thông số robot</i>	101
<i>Hình 7.7 Số lượng hạt xốp trước và sau khi được vệ sinh</i>	103
<i>Hình 7.8 Số lượng mẩu giấy vụn trước và sau khi được vệ sinh</i>	104

DANH MỤC BẢNG

Chương 2:

Bảng 2. 1. So sánh thuật toán di chuyển zigzag và xoắn ốc 14

Chương 3:

Bảng 3. 1. Các chế độ timer..... 18

Chương 5:

Bảng 5. 1. Bảng điều chỉnh PID 52

Bảng 5. 2. Bảng thiết lập thông số Serial port 60

Chương 7:

Bảng 7. 1. Kết quả robot tránh vật cản với 10 lần chạy thử cho mỗi trường hợp 102

Bảng 7. 2. Diện tích sàn được làm sạch ở từng khu vực trong nhà 105

Bảng 7. 3. Bảng thống kê giá thành linh kiện 107

TÓM TẮT LUẬN VĂN

Luận văn trình bày những vấn đề sau:

1. Tìm hiểu phương pháp sử dụng các cảm biến để robot di chuyển linh hoạt trong các không gian, môi trường làm việc có vật cản...đạt được hiệu quả cao vừa tối ưu hóa và tiết kiệm.
2. Nghiên cứu xây dựng giải thuật dùng vi điều khiển STM32F4 và các module tích hợp thêm để điều khiển robot hút bụi tự động.
3. Phát triển một số tính năng đặc biệt, nâng cao tính tiện dụng và hạn chế tối đa các thao tác của con người.
4. Thiết kế khung robot với đầy đủ các bộ phận và đảm bảo tính linh hoạt khi di chuyển.
5. Xây dựng phần mềm giao diện người dùng để có thể điều khiển robot thông qua đường truyền không dây.

CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN ĐỀ TÀI

Nội dung chính

1.1 Đặt vấn đề

1.2 Giới thiệu một số robot hút bụi hiện nay

1.3 Phạm vi đề tài và phương pháp thực hiện

1.1. Đặt vấn đề

Trong cuộc sống hiện đại ngày nay, khái niệm robot đã xuất hiện vô cùng rộng rãi và phổ biến, không còn xa lạ với chúng ta. Nó có thể là món đồ chơi trẻ em, một công cụ phụ trợ trong công việc hoặc thậm chí là một hướng dẫn viên du lịch. Robot là sự tổ hợp khả năng hoạt động linh hoạt của các cơ cấu điều khiển tự động và từ xa, với mức độ “tri thức” ngày càng phong phú của hệ thống điều khiển theo chương trình số cũng như kỹ thuật chế tạo các bộ cảm biến, công nghệ lập trình và các phát triển của trí tuệ nhân tạo, ...

Không những được ứng dụng nhiều trong công nghiệp, robot cũng được xem là trợ thủ đắc lực, phục vụ trong gia đình, xã hội,... dần thay thế các hoạt động tay chân, tự động hóa mọi thứ, giúp cho cuộc sống trở nên tiện nghi, dễ dàng hơn trong khuôn nhịp hối hả, bận rộn như hiện nay. Và một ví dụ điển hình đó chính là robot hút bụi tự động - giúp chúng ta thực hiện công việc làm sạch sàn nhà hiệu quả, không mất sức lực và tiết kiệm thời gian.

Bên cạnh đó, chúng ta cũng thấy được nhiều cải tiến, nâng cấp trong lĩnh vực vi điều khiển. Các module vi điều khiển đã xuất hiện trên thị trường ngày càng nhiều, cực kì đa dạng với tốc độ xử lí cao, cung cấp những tính năng hỗ trợ ngày càng mạnh mẽ. Có thể kể đến PIC, STM, AVR,... Đó là điều kiện thuận lợi để ngành công nghiệp hệ thống nhúng phát triển hơn. Và trong xu thế cạnh tranh, các hãng đang cố gắng phát triển vi mạch, tăng độ tích hợp để có thể hạ giá thành đến mức thấp nhất. Người sử dụng nói chung và sinh viên nói riêng có thể tìm được một bộ vi điều khiển phù hợp, giá rẻ phục vụ cho mục đích chế tạo cũng như học tập và nghiên cứu.

Nhiệm vụ của đề tài là khảo sát các tính năng của robot hút bụi trên thị trường, tham khảo các tài liệu sẵn có, sử dụng vi điều khiển STM và các cảm biến liên quan,..để từ đó phát triển nên một robot có các tính năng tương tự với giá thành thấp hơn, góp phần hỗ trợ cho con người tiết kiệm được thời gian công sức.

1.2. Giới thiệu một số robot hút bụi ngày nay

Robot hút bụi thông minh xuất hiện tại Việt Nam và trở nên phổ biến trong khoảng vài năm trở lại đây. Trong các thương hiệu robot hút bụi thông minh thì Deeboot của Ecovacs và Botvac của Neato là 2 dòng robot hút bụi đang “làm mưa làm gió” trên thị trường trong và ngoài nước, đây cũng là sự lựa chọn số 1 của các gia đình tại Việt Nam.

❖ Dòng sản phẩm Botvac của Neato



Hình 1.1. Một số sản phẩm Botvac

➤ Thiết kế

Botvac là sản phẩm robot hút bụi đầu tiên trên thị trường sở hữu một thiết kế hình chữ D. Nó có thể tiến sâu vào các góc vuông trong căn nhà mà các robot hình tròn không thể làm được. Tuy nhiên, nó vẫn không thể xóa bỏ hoàn toàn cấu trúc dạng tròn của các loại robot hút bụi truyền thống.

➤ Định hướng đường đi

Botvac được trang bị công nghệ dẫn đường bằng Laser. Phía trên bề mặt robot có một tháp Laser được thiết kế nhô hẳn lên trên. Khi hoạt động, tháp Laser này sẽ quét toàn bộ căn phòng để tạo ra một bản đồ hoạt động. Từ đó sản phẩm sẽ tiến hành dọn dẹp căn phòng theo bản đồ làm việc mà tháp laser thu được.

➤ **Sạc**

Một ưu điểm lớn nữa là robot sẽ tự động quay trở lại ô sạc sau khi hoàn thành công việc của mình chứ không chạy lòng vòng cho đến khi hết pin như một số loại khác.

➤ **Hút bụi**

Các sản phẩm Botvac được trang bị công nghệ làm sạch SpinFlow Power Clean. Công nghệ này được tối ưu thiết kế đường dẫn bụi để tạo ra dòng khí xoáy, kết hợp động cơ hút công suất lớn có lực hút mạnh mẽ, hút mọi bụi bẩn trên sàn nhà hay các khu vực thảm.

➤ **Điều khiển**

Dòng Botvac connected còn được trang bị khả năng kết nối với Smartphone giúp người dùng có thể điều khiển robot hoạt động từ xa, đặt lịch làm việc cho robot... thông qua ứng dụng đi kèm của nhà sản xuất.

❖ **Dòng sản phẩm Deebot của Ecovacs**



Hình 1.2. Robot hút bụi Deebot

➤ **Giới thiệu**

Ở các dòng sản phẩm này, người dùng còn được cung cấp thêm tính năng lau nhà tự động. Như vậy là sau khi hút sạch bụi của căn phòng thì robot sẽ tiến hành lau nhà luôn.

➤ **Thiết kế**

Giống như phần lớn các robot hút bụi hiện nay tại Việt Nam, thì robot của Ecovacs cũng mang dạng hình tròn. Chúng tôi đây vẫn là hình dạng thông dụng nhất trong thiết kế robot hút bụi với những đặc tính ưu việt trong thiết kế cũng như vận hành.

➤ **Định hướng đường đi**

Hệ thống dẫn đường của Deebot cũng sử dụng một tháp Laser điều hướng thông minh với công nghệ SmartNavi. Nhờ đó robot có thể làm sạch căn phòng hoàn hảo, tự động dừng khi hoàn thành công việc.

➤ **Hút bụi**

Nói về khả năng làm sạch, Deebot được trang bị tới 2 chổi quét ven và 1 chổi chính hình chữ V, giúp cho robot có thể quét tất cả mọi thứ trên đường nó đi qua và khu vực góc tường.

➤ **Điều khiển**

Deebot cũng được trang bị khả năng kết nối và điều khiển thông qua Smartphone mang lại sự tiện lợi cho người dùng.

1.3. Phạm vi đề tài và phương pháp thực hiện

❖ **Phạm vi đề tài:**

Xây dựng mô hình robot có những tính năng cơ bản sau:

- Robot có khả năng hút các hạt bụi li ti, những bụi bẩn có trọng lượng

tương đối nhẹ để làm sạch nền nhà với các giải thuật ZigZag phù hợp với từng loại không gian.

- Robot tránh chướng ngại vật trên quãng đường di chuyển, ngăn việc va đập gây hư hỏng. Ngoài ra nó cũng được trang bị cảm biến để phát hiện ‘vực sâu’ giúp cho robot không rơi xuống cầu thang hay các bậc thang lên xuống trong quá trình làm việc.
- Xây dựng phần mềm giao diện người dùng để có thể điều khiển robot thông qua đường truyền không dây, tiện lợi hơn trong việc điều khiển cũng như theo dõi hoạt động của robot.

Với các tính năng như trên, ta có thể thấy sản phẩm hoàn toàn có thể đảm bảo được yêu cầu tối thiểu để làm sạch một ngôi nhà không khác những chiếc máy hút bụi hiện có mặt trên thị trường. Do vậy, tính khả thi của đề tài này là tương đối cao, và nhóm chúng em mong muốn sản phẩm sẽ dần hoàn thiện hơn và có thể được ứng dụng vào thực tế.

❖ Phương pháp thực hiện:

Thiết kế mô hình một robot di động tự động, trong đó bao gồm phần gia công cơ khí, mạch công suất, mạch vi điều khiển. Robot hoạt động dựa trên các cảm biến chính là cảm biến hồng ngoại, cảm biến siêu âm và encoder. Tất cả các cảm biến này sẽ gửi tín hiệu về vi xử lý chính STM32f407 để xử lý và đưa ra các thuật toán, giải thuật điều khiển nhằm tránh vật cản, tránh những nơi chênh lệch độ cao và định hướng đường đi. Tích hợp thêm các chức năng khác như module thời gian thực (DS3231), mạch chia áp, module wifi (ESP8266),... để máy hoạt động ở chế độ thời gian thực, truyền nhận không dây và kiểm soát dung lượng pin. Encoder dùng để phát hiện robot dừng do bị trượt bánh hoặc bị mắc kẹt do các vật cản. Bên cạnh đó, xây dựng phần mềm giao diện người dùng điều khiển robot qua đường truyền không dây với hai chế độ bằng tay và tự động, giúp người dùng thuận tiện hơn trong việc điều khiển và theo dõi sản phẩm.

Nội dung luận văn bao gồm 8 chương:

⇒ **Chương 1: Giới thiệu tổng quan đề tài**

Giới thiệu sơ lược về nội dung đề tài và những sản phẩm trên thị trường.

⇒ **Chương 2: Tìm hiểu một số phương pháp vận hành di chuyển**

Nội dung chính giới thiệu một số cách thức di chuyển của robot trong quá trình vận hành để thực thi công việc làm sạch trên những khoảng không gian khác nhau.

⇒ **Chương 3: Giới thiệu một số module sử dụng trong robot**

Giới thiệu sơ lược các module được sử dụng trong quá trình chế tạo robot cũng như công dụng chính của chúng trong quá trình làm việc.

⇒ **Chương 4: Thiết kế phần cơ khí**

Giới thiệu phần mềm Solidworks và quá trình sử dụng để thiết kế 3D, chế tạo phần cơ khí cho robot.

⇒ **Chương 5: Lập trình robot**

Tập trung vào các giải thuật, phương pháp dùng để lập trình cho robot cũng như cách kết nối chúng để tạo nên giải thuật hoàn chỉnh giúp cho sản phẩm vận hành.

⇒ **Chương 6: Thiết kế giao diện người dùng bằng C#**

Giới thiệu phần mềm Visual Studio và giao diện người dùng trên thị trường hiện nay và những tính năng chính đi kèm với sản phẩm robot hút bụi trong đề tài này.

⇒ **Chương 7: Thực nghiệm và đánh giá kết quả**

Tiến hành sử dụng sản phẩm trên các môi trường thực tế, từ đó thống kê và đánh giá mức độ hiệu quả cũng như mức độ hoàn thành công việc của robot, nêu ra những ưu điểm, khuyết điểm cần cải thiện.

⇒ **Chương 8: Kết luận và hướng phát triển**

Đưa ra điểm mạnh, điểm còn thiếu của sản phẩm. Từ đó, đề ra phương hướng phát triển trong tương lai.

CHƯƠNG 2: TÌM HIỂU MỘT SỐ PHƯƠNG PHÁP VẬN HÀNH DI CHUYỂN

Nội dung chính

- 2.1 Phương pháp di chuyển tự động**
- 2.2 Phương pháp di chuyển ZigZag**
- 2.3 Phương pháp di chuyển xoắn ốc**

Tùy theo linh kiện và công nghệ được sử dụng mà robot sẽ được lập trình hướng di chuyển khác nhau. Một vài dòng robot hiện nay được tích hợp công nghệ Smart Move , Smart Navi,... giúp cho robot có thể tạo bản đồ địa hình sau đó lập kế hoạch làm sạch phù hợp với từng không gian riêng, mang lại hiệu quả cao nhất. Tuy nhiên, đây là những công nghệ đòi hỏi phải có những linh kiện chuyên dụng với chi phí khá đắt đỏ và phương pháp lập trình chặt chẽ và chuyên nghiệp. Bên cạnh đó, cũng có những phương pháp mang thiên hướng thủ công hơn, sử dụng những linh kiện thông dụng với chi phí phù hợp hơn để thuận tiện cho việc sử dụng và thay thế nếu có hỏng, kết hợp với phương pháp di chuyển truyền thống của các robot tự hành nói chung và robot hút bụi nói riêng.

Có thể kể đến một số phương pháp như sau:

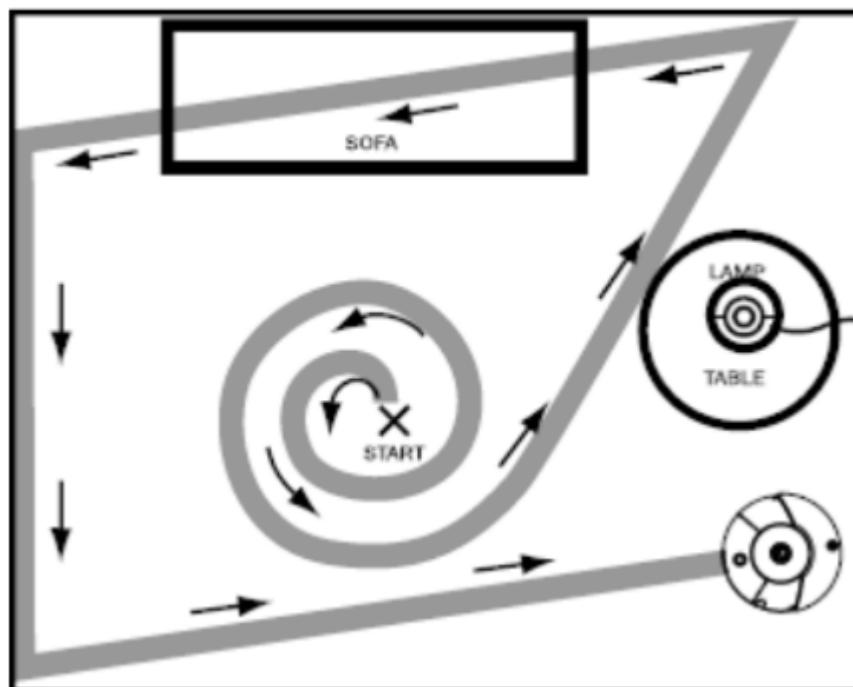
2.1. Phương pháp di chuyển tự động

Với những dòng robot cao cấp, nó sẽ có thể quét lazer và lập bản đồ căn phòng ngay từ lần đầu tiên hoạt động. Thiết bị sẽ vẽ ra căn phòng, định vị không gian và lập trình hướng di chuyển theo công nghệ có sẵn.



Hình 2.1. Phương pháp di chuyển tự động

Sau khi đã có được không gian tổng thể của căn phòng sau lần hoạt động đầu tiên, robot sẽ lập kế hoạch quét để có thể di chuyển hết căn phòng đó theo các cách di chuyển đã được lập trình. Quá trình hoạt động này diễn ra hoàn toàn tự động và robot có thể thay đổi nhiều kiểu di chuyển khác nhau trong một lần làm việc cho phù hợp với không gian đã có. Một số robot còn được tích hợp thêm thiết bị “tường ảo” nhằm giới hạn không gian làm việc ở những nơi không cần thiết hoặc robot không thể hoạt động được (như phòng tắm, phòng trẻ em,...).

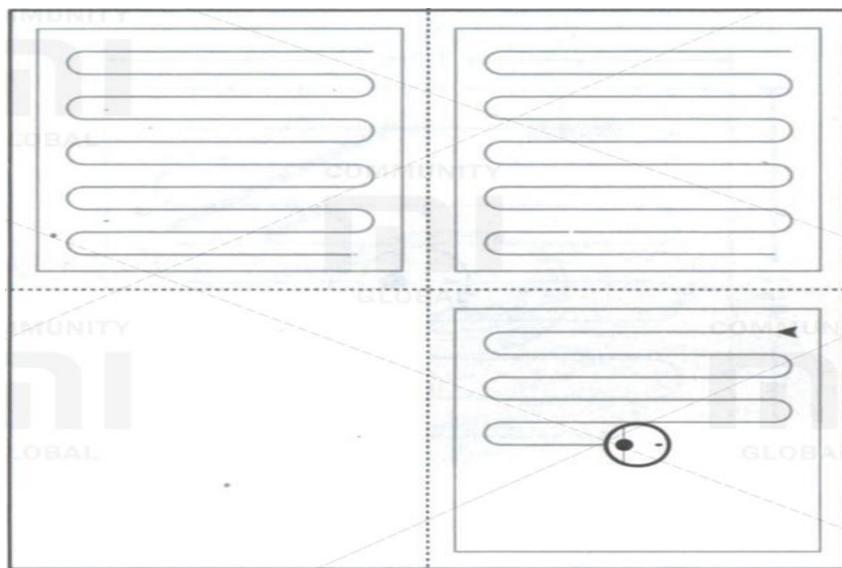


Hình 2.2. Thuật toán di chuyển tổng hợp

Đây có thể là phương pháp di chuyển thông minh và toàn diện nhất. Tuy nhiên có thể đây không phải là phương pháp khả quan nhất đối với sản phẩm của nhóm thực hiện. Nhìn chung phương pháp có những thuận lợi và khó khăn có thể liệt kê như sau:

- Thuận lợi:
 - ✓ Khả năng nhận diện không gian
 - ✓ Phương pháp quét đa dạng
 - ✓ Cách hoạt động linh hoạt và thông minh
- Khó khăn:
 - ✓ Chi phí thiết bị đắt đỏ
 - ✓ Phương pháp lập trình mới và chưa được phổ biến rộng rãi
 - ✓ Cần được điều khiển bởi vi điều khiển có cấu hình mạnh

2.2. Phương pháp di chuyển ZigZag



Hình 2.3. Thuật toán di chuyển zigzag

Phương pháp zig zag sẽ điều khiển cho robot di chuyển theo dạng những đường thẳng song song với nhau. Theo đó, nó sẽ đi thẳng tại một vị trí bắt kè (hoặc được người sử dụng đặt tại một vị trí cụ thể) cho đến khi gặp được một vách đứng (vách

tường, tủ đứng, ...) thì sẽ quay trái hoặc phải lần đầu rồi di chuyển 1 đoạn nhỏ tùy vào độ rộng của thân robot rồi tiếp tục quay một lần nữa (mỗi lần quay 90^0) để chuyển về hướng ngược lại. Và sẽ tiếp tục di chuyển theo cách trên cho đến khi dừng. Sơ đồ đường đi tạo được nhìn chung gần giống như một đường zig zag.

Với phương pháp di chuyển này sẽ có thuận lợi và khó khăn:

- Thuận lợi:
 - ✓ Cách di chuyển đơn giản (robot chỉ chuyển động thẳng hoặc quay một góc 90^0)
 - ✓ Khả năng quét toàn bộ không gian với độ chính xác cao
- Khó khăn:
 - ✓ Đòi hỏi giải thuật lập trình để robot có thể chuyển động thẳng và quay chính xác.
 - ✓ Đối với không gian có nhiều đồ vật có thể làm cho robot di chuyển bị trùng lặp hoặc bỏ sót.

2.3. Phương pháp di chuyển xoắn ốc



Hình 2.4. Thuật toán di chuyển xoắn ốc

Với phương pháp di chuyển kiểu xoắn ốc, robot sẽ khởi động tại một vị trí và bắt đầu di chuyển xung quanh và đổi xứng tại điểm ban đầu đó. Kiểu di chuyển xoắn ốc sẽ phát huy lợi thế trong trường hợp robot chỉ cần hoạt động trong một diện tích cố định mà người dùng mong muốn. Tùy phương pháp lập trình mà kiểu dáng của hình

xoắn ốc có thể là các đường thẳng vuông góc hoặc các đường tròn với bán kính tăng dần. Với dạng đường tròn thì robot phải được lập trình để luôn di chuyển theo đường cong với sự tăng dần bán kính. Với dạng các đường vuông góc thì có cách di chuyển gần giống với phương pháp zig zag.

Tính chất	Zig Zag	Xoắn ốc
Di chuyển thẳng	Cho đến khi gặp vách đứng	Khoảng ngắn và thay đổi tăng dần
Góc quay	90^0	90^0
Số lần quay	2 lần cách nhau khoảng nhỏ	1 lần sau mỗi đoạn đi thẳng

Bảng 2.1. So sánh thuật toán di chuyển zigzag và xoắn ốc dạng vuông góc

Phương pháp di chuyển kiểu xoắn ốc cũng có những thuận lợi và khó khăn:

- Thuận lợi:
 - ✓ Robot có thể hoạt động trong vùng mong muốn
 - ✓ Robot chỉ đi theo một hướng mà không lặp lại vùng đã đi qua
- Khó khăn:
 - ✓ Đòi hỏi việc lập trình khó hơn để định hướng đường đi chính xác
 - ✓ Sai số tích lũy nhiều hơn sau mỗi vòng di chuyển
 - ✓ Chỉ thích hợp hoạt động trong vùng không gian trống, không có vật cản

Sau khi tham khảo các phương pháp trên và dựa vào điều kiện và khả năng hiện tại, nhóm đã quyết định sử dụng phương pháp di chuyển zig zag bởi các lý do sau:

- ✓ Đây tuy không phải là phương pháp thông minh nhất nhưng lại là phương pháp được sử dụng rộng rãi nhất, hầu như tất cả các robot hút bụi đều được tích hợp cách di chuyển này.

Chương 2: Tìm hiểu một số phương pháp vận hành di chuyển

- ✓ Với phương pháp này sẽ tối ưu được diện tích sàn nhà được quét qua, cụ thể nếu ở lần quét đi robot có thể bỏ sót ở một bên của vật cản nhưng ở lần quét ngược lại thì phần sót này sẽ được quét. Vì vậy đảm bảo phần diện tích được quét là lớn nhất.
- ✓ Các linh kiện sử dụng có giá thành rẻ, thông dụng sẽ dễ thay thế khi có hư hỏng.

CHƯƠNG 3: GIỚI THIỆU MỘT SỐ MODULE SỬ DỤNG TRONG ROBOT

Nội dung chính

- 3.1 Vi điều khiển trung tâm STM32F407**
- 3.2 Phát hiện vật cản, vùng chênh lệch độ cao tránh rơi rớt**
- 3.3 Động cơ và module điều khiển tốc độ**
- 3.4 Module thời gian thực – RTC**
- 3.5 Module kết nối không dây truyền dữ liệu UART**

3.1. Vi điều khiển trung tâm STM32F407

STM32 được thiết kế dựa trên dòng Cortex-M3, dòng Cortex-M3 được thiết kế đặc biệt để nâng cao hiệu suất hệ thống, kết hợp với tiêu thụ năng lượng thấp. CortexM3 được thiết kế trên nền kiến trúc mới, do đó chi phí sản xuất đủ thấp để cạnh tranh với các dòng vi điều khiển 8 và 16-bit truyền thống.

KIT STM32F4 có những đặc trưng như sau:

- *Bộ nhớ:* 1MB Flash, 192KB SRAM
- ARM 32-bit CortexTM-M4F CPU với FPU, tần số lên đến 168 MHz
- *Giao thức truyền thông:* Full Speed USB 2.0 Port, Ethernet LAN 10/100Mb, 2 kênh mạng CAN chuẩn, 4 SCI (UART), 3 SPI, 3 I2C Giao tiếp camera 8-14 bit song song với tốc độ lên đến 54 Mbytes /s
- *Analog:* 3 kênh ADC 12 bit, 2 kênh DAC 12 bit
- 16 kênh DMA
- *Timer:* Lên đến 17 timers (16 và 32 bit) với tốc độ lên đến 168MHz mỗi timer có 4 IC/OC/PWM hoặc đếm xung encoder.
- *GPIO:* Có đến 100 chân ngoại vi được tích hợp chức năng ngắt ngoài
- *Nguồn:* cấp nguồn 5V riêng, có ngõ ra nguồn 5V và 3.3V

❖ GPIO (General Purpose Input/Output)

Vi điều khiển STM32F407 có 5 port ngoại vi (PA, PB, PC, PD, PE), với 16 chân mỗi port ($Px0 \rightarrow Px15$). Mỗi chân IO có thể được lập trình một cách riêng lẻ, độc lập dưới dạng input hoặc output. Đối với output có thể thiết lập các chế độ như: push-pull, open-drain hoặc có điện trở pull-up/pull-down. Đối với input có thể chọn dạng floating, pull-up/pull-down hoặc analog. Đặc biệt tất cả các chân ngoại vi đều được tích hợp chức năng ngắt ngoài và có thể giao tiếp trực tiếp với các thiết bị ngoại vi có mức logic 0-5V.

❖ Timer

Timer là bộ định thời có thể sử dụng để tạo ra thời gian cơ bản dựa trên các thông số: **TIM_CLOCK** (xung clock cấp cho timer), **PSC** (Prescaler - bộ chia từ 1 đến 65535), **ARR** (auto-reload register - giá trị đếm của timer 16 hoặc 32 bit), **RCR** (repetition counter register - giá trị đếm lặp lại 16 bits). Timer của STM32F407 là timer 16 bit có thể tạo ra các sự kiện trong khoảng thời gian từ nano giây tới vài phút gọi là **UEV** (update event).

Giá trị UEV được tính theo công thức sau:

$$\text{UEV} = \text{TIM_CLK}/((\text{PSC} + 1)*(\text{ARR} + 1)*(\text{RCR} + 1))$$

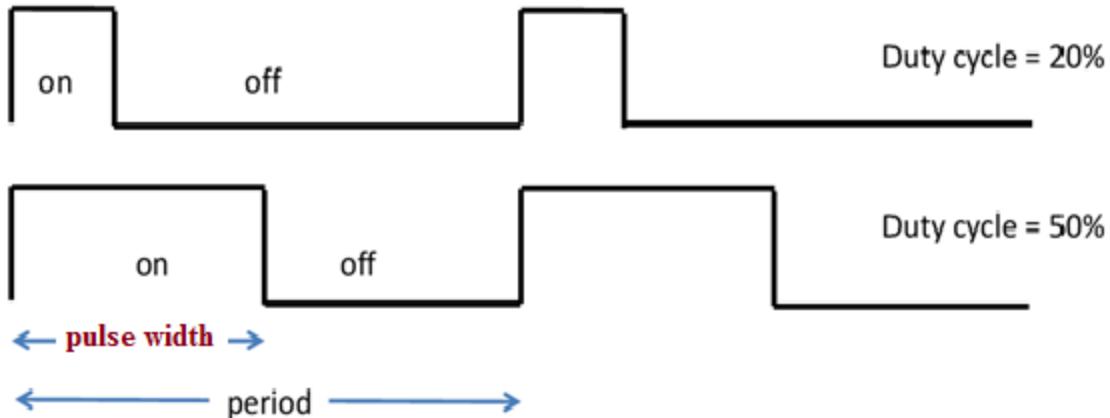
STM32F407 có tổng cộng 14 timer. Trong đó mỗi timer sẽ có chức năng và độ phân giải khác nhau. Có thể tóm tắt theo bảng sau:

TIMER	Loại	Độ phân giải	Độ chia
TIM1, TIM8	Cao cấp	16bit	16bit
TIM2, TIM5	Dùng chung	32bit	16bit
TIM3, TIM4 TIM9 -> TIM14	Dùng chung	16bit	16bit
TIM6, TIM7	Cơ bản	16bit	16bit

Bảng 3.1. Các chế độ timer

❖ PWM (Pulse Width Modulation)

PWM (bộ điều chế xung, hay bộ “băm xung”) là bộ xử lý và điều khiển tạo ra dạng xung vuông chu kỳ thay đổi theo cấu hình. Đây là phương pháp điều chỉnh điện áp ra tải dựa vào trung bình tín hiệu điều chế. Khi độ rộng xung tăng, trung bình điện áp ra tăng và ngược lại. Các module PWM thường sử dụng tần số điều chế không đổi, và điều chỉnh dựa trên sự thay đổi của chu kỳ nhiệm vụ (duty cycle). Trong STM32F407, PWM thuộc khối timer.



Hình 3.1. Sơ đồ biểu diễn xung PWM

$$\text{Duty cycle} = \text{pulse width} \times 100/\text{period}$$

Để cấu hình sử dụng PWM ta có thể chia làm 2 phần như sau :

- Cấu hình timer cho PWM: sẽ quyết định độ rộng của 1 chu kỳ xung PWM là bao nhiêu (period)
- Cấu hình PWM : sẽ quyết định phần trăm của xung mức cao là bao nhiêu phần trăm (pulse width)

❖ DMA (Direct Memory Access)

DMA là một kỹ thuật chuyển dữ liệu tốc độ cao giữa bộ nhớ và ngoại vi hoặc giữa các vùng nhớ mà không yêu cầu đến sự thực thi của lõi vi xử lý. Ở STM32F407 có 2 bộ DMA (DMA1 và DMA2) với 8 kênh mỗi bộ (Stream 0 -> Stream 7) hỗ trợ các chức năng như ADC, SPI, USART, I2C, timer,...

Trong đó có các đặc trưng chính:

- Mỗi kênh đều có thể được cấu hình riêng biệt.
- Với mỗi chức năng sẽ có các kênh hỗ trợ riêng.
- Có 4 mức ưu tiên có thể lập trình: rất cao (very high), cao (high), trung bình (medium), thấp (low).

- Kích thước data được sử dụng là: Byte (8bit), Half Word (16bit), Word (32bit).
- Có 2 chế độ lưu trữ dữ liệu: Bình thường (Normal), Xoay tròn (Circular).
- Có 3 loại truyền dữ liệu là ngoại vi tới vùng nhớ, vùng nhớ tới vùng nhớ và vùng nhớ tới ngoại vi.
- Số lượng data có thể lên đến 65535.

Khi cấu hình DMA kèm theo cấu hình các chức năng chính phải cho phép ngắt DMA. Khi lập trình có thể cho phép hoặc không cho phép DMA hoạt động.

❖ ADC (Analog-to-Digital Converter)

Trong STM32F407 có hỗ trợ 3 bộ ADC, mỗi bộ 15 kênh (ADC1 -> ADC15) chuyên đổi tín hiệu điện áp thành tín hiệu số với độ phân giải lên tới 12bit. Có thể sử dụng cùng lúc nhiều kênh của các bộ ADC khác nhau và có thể lập trình thời gian lấy mẫu riêng cho từng kênh. Ngoài ra có thể thiết lập các chế độ cho các bộ ADC như quét lần lượt từng kênh (Scan Conversion Mode), quét liên tục (Continuous Conversion Mode). Đặc biệt có thể kết hợp DMA để lưu trữ giá trị giúp cho việc chuyển đổi và xử lý sau chuyển đổi diễn ra nhanh hơn và không làm gián đoạn chương trình thực thi những thao tác khác.

❖ USART (Universal Synchronous Asynchronous Receiver Transmitter)

USART là một phương thức truyền nhận nối tiếp đồng bộ và không đồng bộ. Ưu điểm của truyền thông nối tiếp là vi điều khiển có khả năng truyền-nhận nhiều dữ liệu, tiết kiệm đường IO, nhưng nhược điểm là không được nhanh như truyền song song. STM32F407 có 4 bộ USART (1,2,3,6) và 2 bộ UART (4,5) nằm trên các port từ PA đến PD.

Để sử dụng USART phải cấu hình các chi tiết sau:

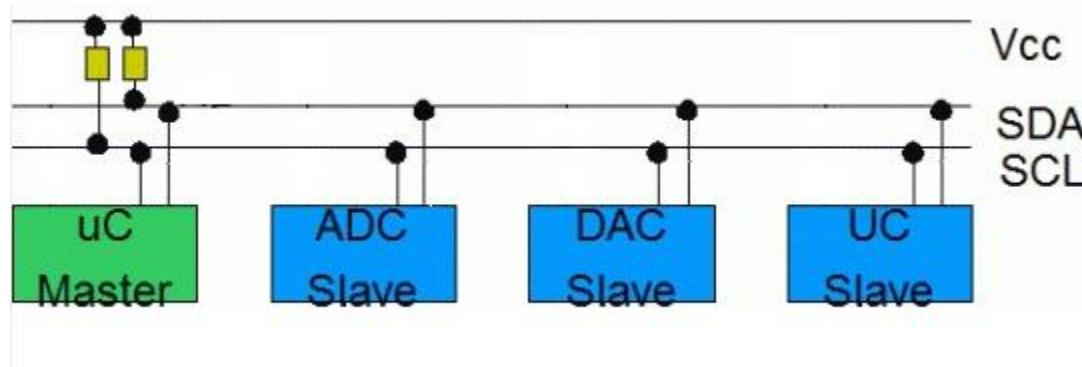
- Tốc độ baud: nằm trong khoảng từ 642 bit/s đến 2625 Mbit/s.

- Độ dài chuỗi truyền: 8 bit hoặc 9bit (đã bao gồm Parity).
- Parity: có thể chọn có hoặc không.
- Stop Bits: Chọn số stop bit (1 hoặc 2).

Ngoài ra, còn có thể cấu hình ngắn truyền nhận nối tiếp và kết hợp sử dụng DMA lưu trữ dữ liệu giúp cho quá trình truyền nhận diễn ra nhanh và thuận tiện hơn.

❖ I2C (Inter Integrated Circuit)

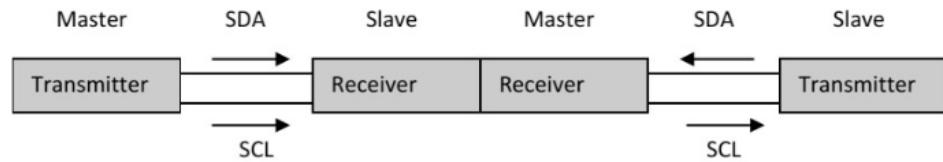
I2C là một chuẩn giao tiếp nối tiếp 2 dây (Serial Data - SDA và Serial Clock - SCL) được sử dụng làm bus giao tiếp ngoại vi cho rất nhiều loại IC khác nhau như các loại Vi điều khiển 8051, PIC, AVR, ARM... Trong đó, SDA là đường truyền dữ liệu hai hướng, còn SCL là đường truyền xung đồng hồ để đồng bộ và chỉ theo một hướng.



Hình 3.2. Sơ đồ hoạt động I2C

Hình trên cho thấy sẽ có nhiều thiết bị cùng kết nối trên 1 bus I2C, do đó phải có sự phân địa chỉ và một quan hệ chủ / tớ cho từng thiết bị tránh sự nhầm lẫn trong suốt quá trình giao tiếp. Mỗi thiết bị có thể hoạt động như là thiết bị nhận hoặc truyền dữ liệu hay có thể vừa truyền vừa nhận tùy thuộc vào việc thiết bị đó là chủ (master) hay tớ (slave).

Trong giao tiếp I2C chỉ có thiết bị chủ (master) mới có quyền điều khiển, nó sẽ tạo xung đồng hồ và quản lý địa chỉ cho toàn hệ thống bus I2C đó. Thiết bị chủ giữ vai trò chủ động, còn thiết bị tớ giữ vai trò bị động trong việc giao tiếp.



Hình 3.3 Sơ đồ truyền nhận I2C

Nhìn hình trên ta thấy xung đồng hồ chỉ có một hướng từ chủ đến tớ, còn luồng dữ liệu có thể đi theo hai hướng, từ chủ đến tớ hay ngược lại.

STM32F407 có 2 bộ I2C với 2 chế độ hoạt động: Standard mode và Fast mode.

- Standard mode (chế độ chuẩn)
 - Đây là chế độ chuẩn ban đầu được phát hành vào đầu những năm 80.
 - Có tốc độ dữ liệu tối đa 100 kbps.
 - Sử dụng 7-bit địa chỉ và 112 địa chỉ tớ
- Fast mode (chế độ nhanh)
 - Tốc độ dữ liệu tối đa được tăng lên đến 400 kbps.
 - Có thể lựa chọn duty cycle Tlow/Thigh.

3.2. Phát hiện vật cản, vùng chênh lệch độ cao tránh rơi rớt

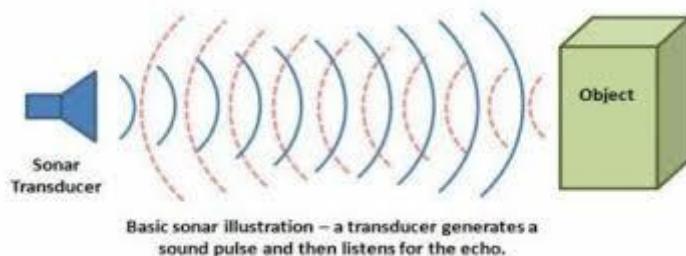
3.2.1 Module cảm biến siêu âm SRF-04



Hình 3.4. Cảm biến siêu âm

Cảm biến siêu âm hoạt động dựa trên nguyên lý phát sóng âm ra ngoài, khi sóng âm tiếp cận tới các vật thể (nước, nước thải, chất lỏng dạng kết dính..., các loại chất rắn như hạt nhựa, cát, đá, xi măng, bột, cám gạo...) sẽ phản xạ sóng âm về cảm

biến. Sau đó cảm biến sẽ xử lý và đưa về thành tín hiệu dòng 4-20mA tiếp tục truyền đi tới các thiết bị kết nối để phát tín hiệu kết quả đo được cho người dùng.



Hình 3.5. Nguyên lý đo khoảng cách bằng truyền âm

Thông số kỹ thuật:

- Điện thế hoạt động: DC 5V
- Dòng tiêu thụ: 2 mA
- Góc quét: < 15 độ
- Khoảng cách nhận: 2~100cm
- Độ chính xác: lên đến 0.3cm

Giao tiếp:

- VCC: điện áp cấp từ 3.3V đến 5V
- GND: GND ngoài
- Trig: chân Trigger để nhận tín hiệu điều khiển
- Echo: chân Echo để phản hồi về xung có độ rộng tương ứng với khoảng cách vật

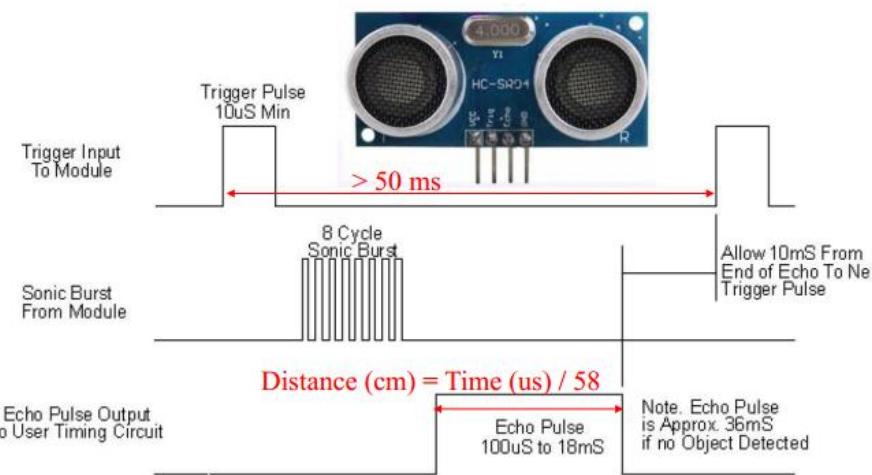
Để đo khoảng cách, ta sẽ phát 1 xung rất ngắn (5 microSeconds - us) từ chân **Trig**. Sau đó, cảm biến sẽ phát ra lượt sóng siêu âm và tạo ra 1 xung HIGH ở chân **Echo** cho đến khi nhận lại được sóng phản xạ thì hạ xuống mức LOW. Độ rộng của xung sẽ bằng với thời gian sóng siêu âm được phát từ cảm biến và quay trở

lại. Quá trình được lặp lại với chu kỳ phải lớn hơn 50ms để đảm bảo độ chính xác.

Tốc độ của âm thanh trong không khí là 340 m/s (hàng số vật lý). Khoảng cách từ cảm biến đến vật được tính theo công thức:

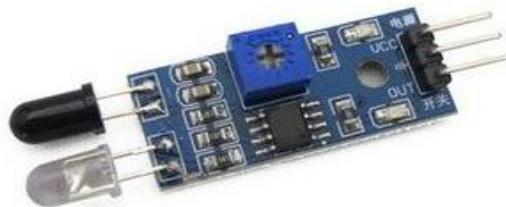
$$\text{Distance} = \frac{T}{2} * 0.034 \text{ (cm)}$$

với T là độ rộng xung Echo (us)



Hình 3.6. Cách tính khoảng cách dùng cảm biến siêu âm

3.2.2 Module cảm biến hồng ngoại



Hình 3.7. Cảm biến hồng ngoại

Cảm biến hồng ngoại có khả năng nhận biết vật cản trực diện, có một cặp truyền và nhận tia hồng ngoại và một biến trở điều chỉnh khoảng cách.

Tia hồng ngoại phát ra một tần số nhất định, khi phát hiện hướng truyền có vật

cản (mặt phản xạ), phản xạ vào đèn thu hồng ngoại, sau khi so sánh, đèn màu xanh sẽ sáng lên, đồng thời phát tín hiệu số đầu ra (một tín hiệu bậc thấp).

Khoảng cách làm việc hiệu quả 2 ~ 5cm, điện áp làm việc là 3.3 V đến 5V. Độ nhạy sáng của cảm biến được điều chỉnh bằng chiết áp, cảm biến dễ lắp ráp, dễ sử dụng,...

Có thể được sử dụng rộng rãi trong robot tránh chướng ngại vật, xe tránh chướng ngại vật và dò đường,...

Giao tiếp:

- VCC: điện áp chuyển đổi từ 3.3V đến 5V (có thể được kết nối trực tiếp đến vi điều khiển 5V và 3.3V)
- GND: GND
- OUT: đầu ra kỹ thuật số (0 và 1) với mức 0 khi phát hiện có vật cản.

3.3. Động cơ và module điều khiển tốc độ

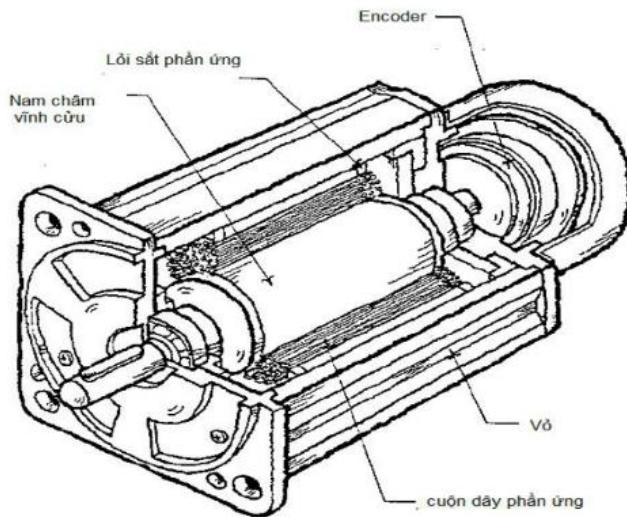
3.3.1 Động cơ encoder

Động cơ servo là “cơ bắp” của hệ thống điều khiển chuyển động. Chúng cung cấp lực cần thiết để di chuyển các thiết bị theo yêu cầu của ứng dụng. Động cơ servo có nhiều kiểu dáng và kích thước, được sử dụng trong nhiều máy khác nhau, từ máy tiện điều khiển bằng máy tính cho đến các mô hình máy bay và xe hơi. Ứng dụng mới nhất của động cơ servo là trong các robot, cùng loại với các động cơ dùng trong mô hình máy bay và xe hơi.

Động cơ servo được thiết kế cho những hệ thống hồi tiếp vòng kín. Tín hiệu ra của động cơ được nối với một mạch điều khiển. Khi động cơ quay, vận tốc và vị trí sẽ được hồi tiếp về mạch điều khiển này. Nếu có bất kỳ lý do nào ngăn cản chuyển động quay của động cơ, cơ cấu hồi tiếp sẽ nhận thấy tín hiệu ra chưa đạt được vị trí mong muốn. Mạch điều khiển tiếp tục chỉnh sai lệch cho động cơ đạt được điểm chính xác.

Cấu tạo cơ bản của một động cơ servo gồm có:

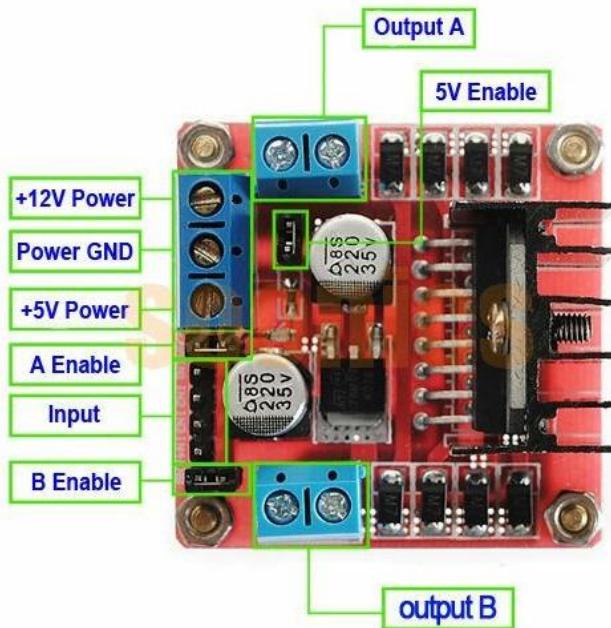
1. Nam châm vĩnh cửu
2. Lõi sắt phần cứng
3. Bộ Encoder
4. Vỏ
5. Cuộn dây phần cứng



Hình 3.8. Cấu tạo động cơ

3.3.2 Mạch lái động cơ L298

Mạch điều khiển động cơ DC L298 là module được tích hợp sẵn IC L298, các diode bảo vệ vi xử lý giúp chống lại các dòng điện cảm ứng từ việc bật/tắt động cơ và một IC LM7805 để cấp nguồn 5VDC cho các thiết bị khác. Module có khả năng điều khiển 2 động cơ DC hoặc 1 động cơ bước, dòng tối đa là 2A mỗi động cơ.



Hình 3.9. Mạch lái L298

Trên module có:

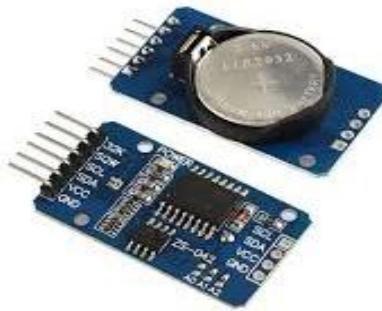
- Chân nguồn vào 12V để cấp điện áp cho động cơ hoạt động.
- Chân nguồn ra 5V để cấp áp cho các thiết bị khác.
- Chân A Enable và B Enable đã được đưa lên tích cực mức 1.
- 1 chân 5V Enable để cho phép có ngõ ra 5V.
- Chân input để cấp ngõ vào điều rộng xung (PWM) để điều khiển động cơ.
- Ngõ ra output A và output B để xuất điện áp ra điều khiển động cơ.
- Input: Nhóm này gồm 4 chân IN1, IN2, IN3, IN4. Đây là 4 chân chính dùng để nhận tín hiệu điều khiển động cơ từ vi điều khiển.
 - IN1 = 1 , IN2 = 0: Động cơ A sẽ quay theo chiều thuận.
 - IN1 = 0 , IN2 = 1: Động cơ A sẽ quay theo chiều nghịch.
 - IN1 = 0 , IN2 = 0: Dừng động cơ A.
 - IN3 = 1 , IN4 = 0: Động cơ B quay theo chiều thuận.

- IN3 = 0 , IN4 = 1: Động cơ B quay theo chiều nghịch.
- IN3 = 0 , IN4 = 0: Dừng động cơ B

Nếu 4 chân input được thiết lập như trên thì động cơ sẽ quay với tốc độ tối đa. Tương tự như các chân A Enable và B Enable ta cũng có thể điều khiển tốc độ động cơ bằng cách cấp xung PWM vào các chân này.

3.4. Module thời gian thực – RTC

DS3231 là IC thời gian thực giá rẻ, rất chính xác với thạch anh tích hợp sẵn có khả năng điều chỉnh nhiệt. IC có nguồn pin đầu vào riêng, tách biệt khỏi nguồn chính đảm bảo cho việc giữ thời gian chính xác trong thời gian không sử dụng. Thạch anh tích hợp sẵn giúp tăng độ chính xác trong thời gian dài hoạt động và giảm số lượng linh kiện cần thiết khi làm board.

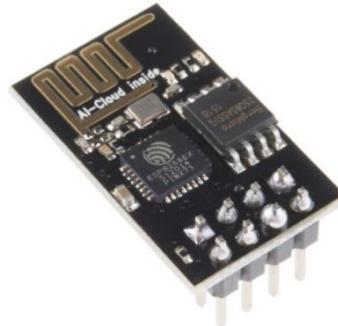


Hình 3. 10. Module thời gian thực DS3231

Thời gian trong IC được giữ ở dạng: giờ, phút, giây, thứ, ngày, tháng, năm. Các tháng có ít hơn 31 ngày sẽ tự động được điều chỉnh, các năm nhuận cũng được chỉnh đúng số ngày. Thời gian có thể hoạt động ở chế độ 24h hoặc 12h AM/PM. IC còn có chức năng báo động, có thể cài đặt 2 thời gian báo và lịch, có tín hiệu ra là xung vuông. Giao tiếp với IC được thực hiện thông qua I2C bus.

Trong chip có mạch điện áp chuẩn dùng để theo dõi trạng thái của nguồn VCC, phát hiện lỗi nguồn, tự động chuyển nguồn khi có vấn đề. Ngoài ra trong IC còn có sẵn cảm biến nhiệt độ, có độ chính xác là $\pm 3^{\circ}\text{C}$.

3.5. Module kết nối không dây truyền dữ liệu UART



Hình 3.11. Module wifi ESP8266

Mạch thu phát Wifi ESP8266 Uart ESP-01 sử dụng IC Wifi SoC ESP8266 của hãng Espressif, được sử dụng để kết nối với vi điều khiển thực hiện chức năng truyền nhận dữ liệu qua Wifi. Mạch có thiết kế nhỏ gọn, sử dụng giao tiếp UART với bộ thu viễn và code mẫu rất nhiều từ cộng đồng. ESP-01 được sử dụng trong các ứng dụng IoT và điều khiển thiết bị qua Wifi.

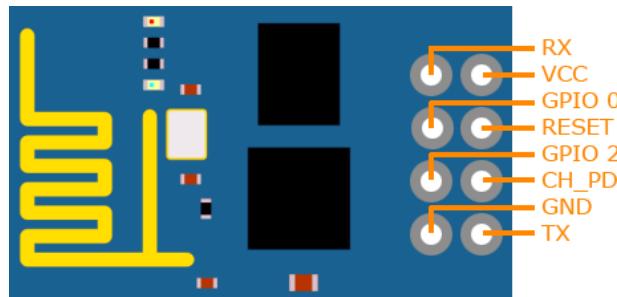
Mạch thu phát Wifi ESP8266 Uart ESP-01 không được tích hợp sẵn chip nạp. Tuy nhiên, ta có thể sử dụng một mạch giao tiếp USB-UART để giao tiếp UART và nạp code cho chip (ví dụ như mạch CP2102).

ESP8266 có chức năng phát wifi và truyền nhận gói tin được gửi từ vi điều khiển tới PC và ngược lại. Tuy nhiên để thực hiện được chức năng trên thì ESP phải được lập trình và nạp code phù hợp. Do đó, ESP phải hoạt động tốt ở cả hai chế độ là Run mode (load code từ bộ nhớ flash khi reset hoặc cấp nguồn) và Flash mode (nạp code mới). Ứng với mỗi mode sẽ có cách nối các chân với các mức logic khác nhau.

Sơ đồ nối chân:

- URXD(RX) — nhận tín hiệu trong giao tiếp UART với vi điều khiển
- UTXD (TX) — truyền tín hiệu trong giao tiếp UART với vi điều khiển
- VCC — đầu vào 3.3V

- GPIO 0 — kéo xuống thấp cho chế độ upload bootloader
- RST — chân reset cứng của module, kéo xuống mass để reset
- GPIO 2 — thường được dùng như một cổng TX trong giao tiếp UART để debug lỗi
- CH_PD — kích hoạt chip, sử dụng cho Flash Boot và updating lại module, nối với mức cao
- GND — nối với mass



Hình 3.12. Sơ đồ dây của ESP8266

Thông số kỹ thuật:

- Điện áp sử dụng: 3.3VDC
- Dòng tiêu thụ: Max 320mA
- Hỗ trợ cả 2 giao tiếp TCP và UDP.
- Chuẩn giao tiếp UART với Firmware hỗ trợ bộ tập lệnh AT Command, tốc độ Baudrate mặc định 9600 hoặc 115200.
- Có 3 chế độ hoạt động: Client, Access Point, Both Client and Access Point.

CHƯƠNG 4: THIẾT KẾ MÔ HÌNH PHẦN CƠ KHÍ

Nội dung chính

- 4.1 Giới thiệu phần mềm Soliworks**
- 4.2 Giới thiệu mô hình robot**
- 4.3 Thiết kế phần thân robot**
- 4.4 Thiết kế phần đế robot**
- 4.5 Thiết kế hộp bụi**

4.1 Giới thiệu phần mềm Solidworks

Vì đề tài robot hút bụi tự động mang tính ứng dụng thực tế nên vấn đề cần có mô hình cụ thể là cần thiết vì như vậy mới đáp ứng được các yêu cầu đặt ra. Để có thể khái quát cấu trúc của một robot hút bụi, nhóm đã tham khảo các mô hình máy thực tế đang được bán trên thị trường hiện nay và sử dụng phần mềm SolidWorks để vẽ lên mô hình 3D một cách trực quan nhất.



Hình 4.1. Phần mềm SolidWorks

SOLIDWORKS là phần mềm thiết kế 3D chạy trên hệ điều hành Windows. Đây là một công cụ hỗ trợ mạnh mẽ cho việc thiết kế, mô hình hóa vật thể, phục vụ không chỉ riêng cho lĩnh vực cơ khí chế tạo mà còn được ứng dụng cho các ngành nghề khác như: kiến trúc, xây dựng, mỹ thuật,...

Trong SolidWorks có tính năng nổi bật nhất là có thể thiết kế các biên dạng 2D và từ đó có thể dựng lên các khối 3D theo yêu cầu. Các chi tiết 3D sau đó có thể lắp ráp lại với nhau để tạo thành một bộ phận hoặc một máy hoàn chỉnh.

Phần mềm SolidWorks cho phép tạo ra các hình chiếu vuông góc hoặc các bản láp với tỉ lệ và vị trí do người thiết kế quy định mà không ảnh hưởng đến kích thước.

Đặc biệt, SolidWorks còn hỗ trợ để xuất các file thiết kế ra dạng đuôi *.STL phù hợp với hầu hết các máy in 3D hiện nay trên thị trường. Do đó, nhóm đã quyết định thiết kế robot bằng phần mềm SolidWorks sau đó đặt in 3D mô hình.

4.2 Giới thiệu mô hình robot

Yêu cầu của robot hút bụi là nhỏ gọn, di chuyển linh hoạt. Do đó, cơ cấu di chuyển của robot hút bụi sử dụng mô hình di động sử dụng hai bánh dẫn chủ động và một bánh dẫn hướng. Vì nhiệm vụ chính của robot là hút bụi nên cụm hút bụi được ưu tiên khi thiết kế. Cụm hút bụi được đặt ở phần sau của robot, trong đó hộp chứa bụi có thể được tháo rời để làm sạch một cách dễ dàng. Vùng hút bụi được mở rộng tối đa với kích thước miệng hút mở rộng theo phương ngang (phương vuông góc với phương di chuyển của robot). Để tăng hiệu suất hút bụi, cụm chổi quét được thiết kế phía trước cụm quạt hút. Cụm chổi quét hoạt động liên tục để khoáy đảo lớp bụi bám trên bề mặt sàn đồng thời gom bụi trên đường đi và bụi ở những góc tường tới gần vị trí cửa hút của robot. Sau đó, lớp bụi này được hút bởi cụm quạt hút ở phía sau.

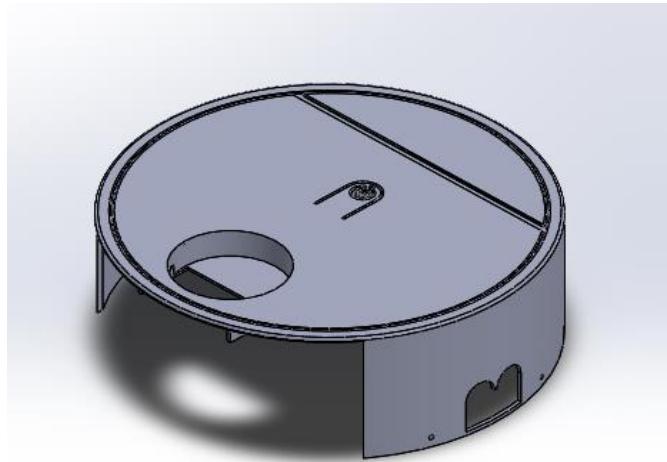
Bố trí vị trí cảm biến trên robot: robot sử dụng các cảm biến hồng ngoại để tránh chướng ngại vật và cảm biến siêu âm để phát hiện vật cản. Từ mục đích trên, các cảm biến được thiết kế với các vị trí như sau:

- Có 3 cảm biến hồng ngoại được đặt phía trước: 1 để tránh chướng ngại vật, đặt ngay mép của thân robot hút bụi hướng xuống phía dưới để xác định mặt sàn như là một vật cản, 2 cảm biến còn lại được đặt hai bên cảm biến siêu âm hướng ra phía trước để hỗ trợ tìm vật cản.
- Cảm biến phát hiện vật cản trên đường đi được đặt phía trước robot và 2 bên thân. Nhằm xác định vật cản phía trước (khi chạy thẳng) và hai bên hông (khi xoay).

Ngoài ra, trên thân robot còn thiết kế vị trí lắp đặt nguồn nuôi cho robot và mạch điện, quạt hút bụi, động cơ bánh xe và các chi tiết khác.

Từ các ý tưởng thiết kế trên, nhóm đề xuất cấu tạo của robot hút bụi như sau:

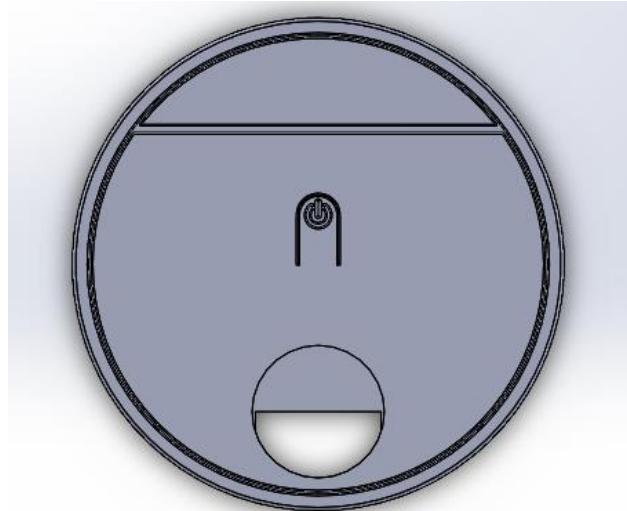
4.3 Thiết kế phần thân robot



Hình 4.2. Phần thân robot

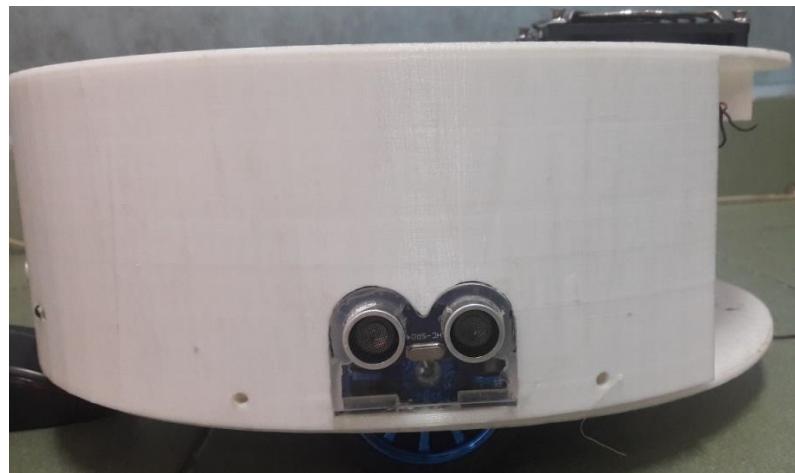
Phần trên robot mang dạng tròn, đường kính 30 cm, cao 10cm, thành dày 3mm và rỗng ruột, có chức năng như là nắp của robot. Kích thước của robot được tính toán phù hợp sao cho vừa có sức chứa đầy đủ các thành phần cấu tạo vừa không quá cồng kềnh và đặc biệt phải có độ bền, tránh hư hại trong quá trình lắp ráp cũng như vận hành.

Thiết kế của robot rất quan trọng do nó ảnh hưởng trực tiếp đến cách di chuyển và hút bụi của chính nó. Robot hút bụi thông minh hiện nay hầu hết được thiết kế theo dạng hình tròn. Dạng tròn nhằm tối ưu hóa cho việc di chuyển linh hoạt dễ dàng, lăn xả vào những nơi ngóc ngách nhất trong căn nhà, tại các vách tường hoặc các bệ mặt đứng. Hơn nữa, thiết kế dạng tròn giúp tránh va đập mạnh, gây xây xước và tổn hại cho cả vật, người và robot. Ngoài ra nó cũng giúp cho các cảm biến xác định tương đối chính xác khoảng cách từ robot đến vật cản phía trước. Về mặt hình học thì hình tròn còn mang ý nghĩa của sự hoàn hảo và trọn vẹn, mang đến cảm giác tự nhiên, thuận mắt người nhìn.

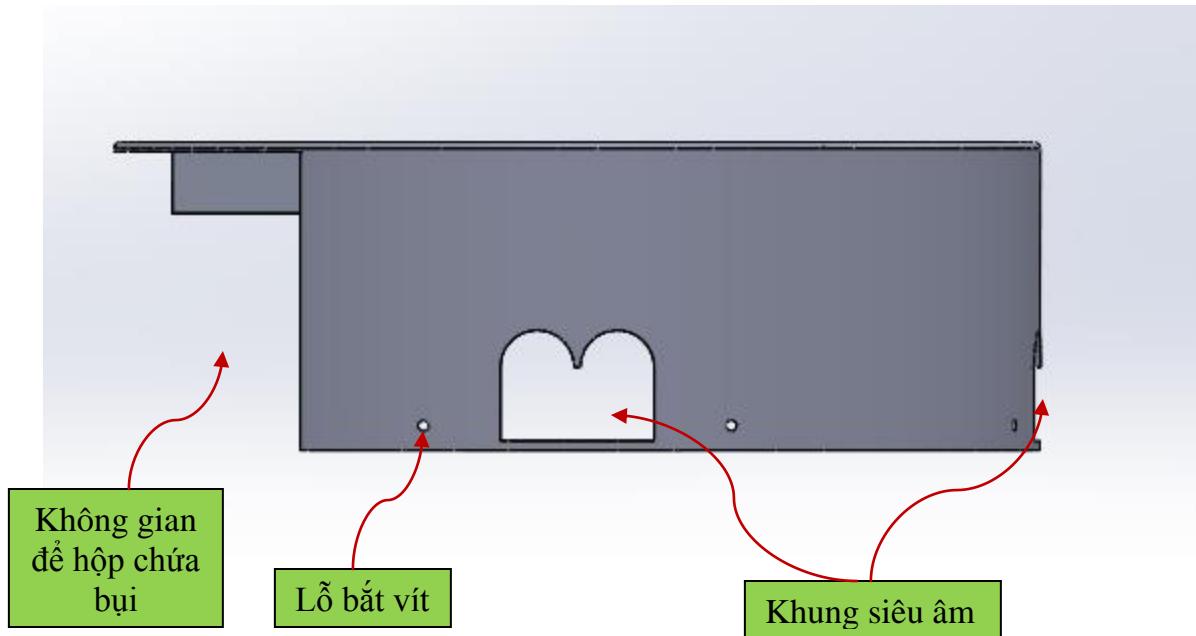


Hình 4.3. Phần thân ở góc nhìn từ trên xuống

Phía trên của nắp có thiết kế nút nhấn với hình ảnh kí hiệu nguồn trực quan tinh xảo giúp cho việc bật/tắt robot thuận tiện hơn khi vận hành. Một lỗ tròn ở phía đuôi robot là để gắn quạt hút bụi, lợi dụng phần không gian phía trên để tiết kiệm diện tích trong thân. Lỗ quạt này sẽ được liên kết với hộp bụi phía trong tạo nên một khoảng không gian kín, đủ để tạo áp suất hút tại miệng hút bụi. Ngoài ra, trên nắp còn có các đường rãnh mang mục đích trang trí, tăng tính thẩm mĩ, giảm độ thô khi nhìn vào sản phẩm. Tại mỗi cạnh trên phần nắp đều được thiết kế bo góc tròn nhằm tăng sự mềm mại cũng như tránh để cạnh vuông góc sắc nhọn gây nguy hại cho vật và người sử dụng.

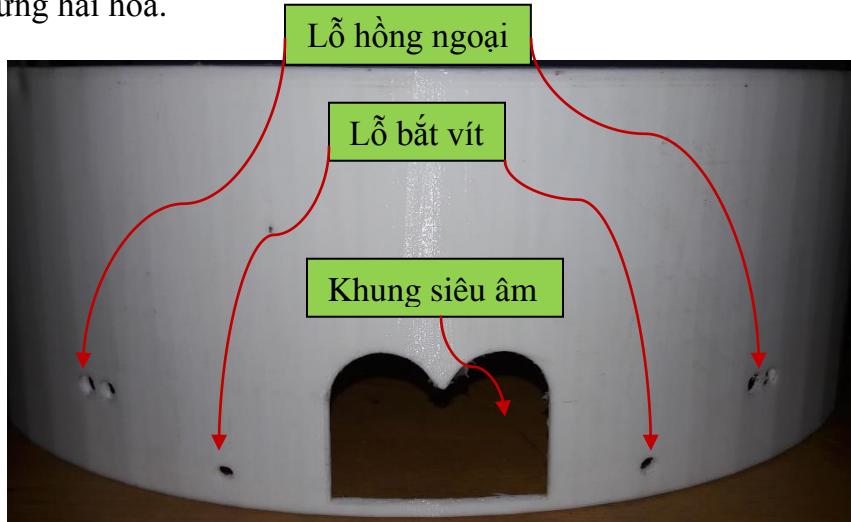


Hình 4.4. Thực tế phần thân ở góc nhìn từ bên trái



Hình 4.5. Cách bố trí cảm biến trên thân

Phần thân có độ dày 3mm với độ cứng tương đối cao, khó gãy vỡ, đủ để định hình và cố định cho phần nắp của robot. Trên thân tại các vị trí phía trước và hai bên bánh xe được khoét các lỗ để gắn giá đỡ cảm biến siêu âm với dạng hai cung tròn liên kết với hình chữ nhật bên dưới. Hình dạng của khung siêu âm này được đo đặc cẩn thận từ kích thước của khung trong thực tế để tạo nên sự đồng nhất. Hai bên mỗi khung siêu âm sẽ được khoét hai lỗ tròn để bắt vít cố định phần nắp với phần đế của robot tạo nên sự đối xứng hài hòa.



Hình 4.6. Thực tế cách bố trí cảm biến trên thân

Ở phía đuôi được thiết kế trống hoàn toàn vì đây sẽ là nơi để hộp chứa bụi. Việc thiết kế hộp chứa bụi nằm ở một phía của robot nhằm tách biệt khu vực hút với các khu vực còn lại, tránh để bụi làm ảnh hưởng đến các thiết bị điện tử bên trong, không những vậy nó còn giúp tiết kiệm diện tích không gian, dễ dàng thiết kế, lắp đặt và tháo rời.

Do hạn chế trong khâu thiết kế và in 3D nên mặt nền của phần nắp có dạng bằng phẳng, trong khi một số máy hút bụi hiện nay với thiết kế tinh xảo hơn thì phần nền này có dạng hơi lồi lên và có cấu tạo lắp ghép phức tạp hơn, để tạo sự đẹp mắt và tăng thêm không gian phía trong của máy. Tuy nhiên, nhóm nhận thấy việc này thực sự không cần thiết và tăng chi phí in ấn lên nhiều lần nên vẫn giữ thiết kế bằng phẳng với điểm nhấn là những đường viền trên thân.

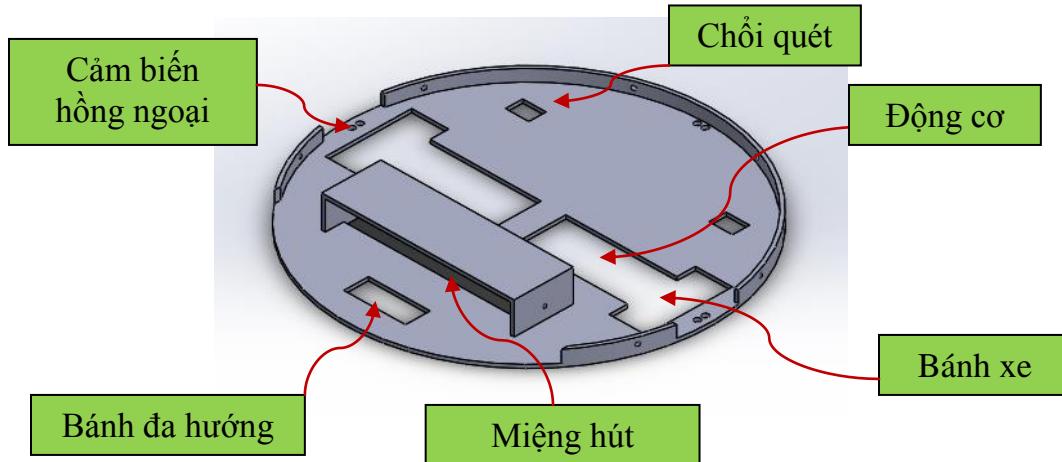


Hình 4. 7. Thực tế phần thân ở góc nhìn từ trên xuống

4.4 Thiết kế phần đế robot

Phần đế được thiết kế với hình dáng và kích thước tương ứng với phần thân của robot, trong đó đã tính tới các trường hợp giãn nở khi in ấn. Các chi tiết được đo đạc cẩn thận sao cho khi lắp ráp các bộ phận phải khít với nhau, tránh cong vênh. Đồng

thời các chi tiết này phải hỗ trợ định hình, kết nối để chịu được trọng lượng lớn mà không bị gãy do robot được lắp ghép thêm các bộ phận có khối lượng tương đối nặng.



Hình 4.8. Phần đế của robot

Có thể thấy trên phần đế các chi tiết được thiết kế để gắn các bộ phận của robot như động cơ, bánh xe, chổi quét, cảm biến hồng ngoại,...với kích thước đã được đo sao cho phù hợp với vật thể thực tế. Riêng phần đế lắp động cơ đã không được thiết kế phần khung trong bản vẽ này mà chỉ được tạo một lỗ trống với độ dài và rộng phù hợp. Đặc biệt, bộ phận miệng hút được thiết kế mở với chiều ngang tối đa để có được diện tích một lần quét là lớn nhất có thể, giúp tiết kiệm thời gian và năng lượng để hoạt động.

Tuy nhiên, sau khi vẫn hành thực tế nhóm đã nhận thấy do miệng hút được thiết kế quá rộng, dẫn đến lực hút bị giảm đi đáng kể và không thể hút bụi một cách hiệu quả. Nên đã có sự điều chỉnh phần cứng sau khi in bằng bìa carton nhằm thu nhỏ độ rộng của miệng hút và thu hẹp khoảng cách từ miệng hút tới bề mặt sàn. Đồng thời, khung chứa động cơ cũng được gắn thêm vào sau khi in. Việc không thêm phần khung động cơ vào thiết kế bản vẽ ban đầu bởi hai lý do. Một là để thuận tiện cho việc in ấn (nhóm đã tham khảo với nhiều đơn vị in ấn và được khuyên nên bỏ đi phần nhô ra của khung động cơ để cho bản in được dễ dàng, đẹp mắt và tiết kiệm chi phí hơn). Hai là để chủ động trong việc điều chỉnh độ cao của bánh xe và linh hoạt hơn trong việc cài đặt.

động cơ vào khung, vì khung bằng bìa carton sẽ dễ uốn để ôm sát động cơ hơn khung bằng nhựa.



Hình 4.9. Thực tế mặt dưới của robot

Đế của robot có độ dày 3mm đảm bảo cho sức chịu trọng lượng của các chi tiết lắp ghép. Xung quanh đế có phần thành nhô cao lên 1cm. Khi lắp ráp thì mặt ngoài của phần thành này sẽ áp sát với mặt trong của phần nắp. Đồng thời trên thành này cũng sẽ khoét những khoảng trống để lắp khung cảm biến siêu âm, hộp chứa bụi và lỗ bắt vít với vị trí tương ứng với nắp. Thành được thiết kế dày 3mm đủ chắc chắn để chịu được sức nặng của toàn bộ phần đế khi kết nối với nắp mà không bị nứt gãy.

Khung đế sau khi được in sẽ được khoan thêm những lỗ để bắt ốc vít cố định mạch điều khiển, mạch công suất và công tắc nguồn nhằm đảm bảo cho việc sắp xếp linh kiện bên trong hoàn chỉnh và hợp lý nhất. Tránh tình trạng có sự điều chỉnh vị trí trong quá trình lắp ráp khác với thiết kế sẽ dẫn đến việc phải khoan lại, làm xuất hiện những lỗ hỏng dư thừa, làm xấu bè mặt robot.

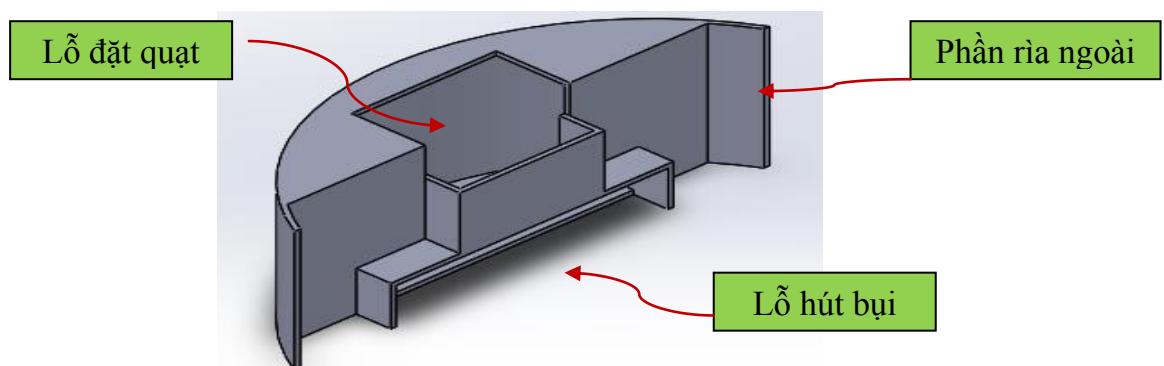


Hình 4.10. Phần đế ở góc nhìn từ phía sau

Cũng giống như phần nắp robot, đế robot cũng được thiết kế với phần mặt dưới bằng phẳng, sau đó sẽ được gia công thêm các chi tiết lồi ra sau khi in (khung chứa động cơ, thanh cố định bánh đa hướng) để tiết kiệm chi phí cũng như đảm bảo tính thẩm mỹ cho mặt ngoài của robot.

4.5 Thiết kế hộp bụi

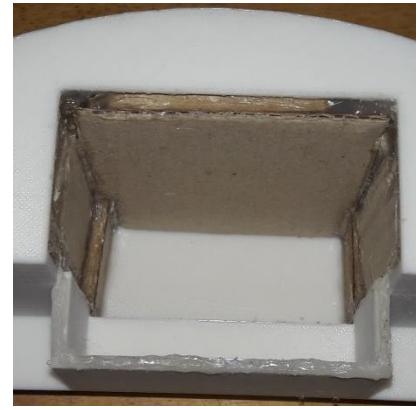
Hộp bụi được coi là phần quan trọng nhất của khung cơ khí vì nó quyết định cấu trúc bên trong robot cũng như khả năng hút bụi khi làm việc. Do đó, việc thiết kế đòi hỏi sự tính toán và tỉ mỉ tròn từng chi tiết. Nó phải vừa vặn với hai phần còn lại nhưng vẫn phải đảm bảo sự nhỏ gọn, tiện lợi và thẩm mỹ.



Hình 4.11. Phần thiết kế chứa bụi của robot

Hộp bụi được thiết kế để lắp vào phần còn trống của thân robot, với cấu tạo liền khói giúp cho bụi được giữ kín bên trong cho tới khi được làm sạch. Lỗ hút bụi và lỗ đặt quạt cũng được đo đạc sao cho vừa vặn với phần thân và đế robot, tạo không gian kín có áp suất lớn để hút được các loại bụi khác nhau.

Như đã trình bày ở phần trên, việc không gian hút bụi quá lớn sẽ làm giảm hiệu năng hút bụi của robot. Do đó nhóm cũng đã sử dụng thêm bìa carton để thu hẹp không gian bên trong hộp chứa bụi. Tuy nhiên vẫn đảm bảo phần không gian đủ lớn để thực hiện đầy đủ chức năng của mình. Còn phần khung bên ngoài vẫn giữ nguyên hình dạng và kích thước như thiết kế ban đầu.



Hình 4.12. Khoang chứa bụi

Ở hai cạnh bên của hộp được thiết kế thêm phần rìa dài thêm, phần rìa này sẽ ôm sát thân robot ở phía ngoài khi lắp ráp nhằm che đi mối nối giữa thân và hộp. Đồng thời nó còn giúp cho hộp có diện tích lớn hơn do phần cung tròn có đường kính ngoài lớn hơn đường kính thân của robot (như vậy mới có thể ôm sát thân từ phía ngoài).



Hình 4. 13. Thực tế hộp bụi

CHƯƠNG 5: LẬP TRÌNH ROBOT

Nội dung chính

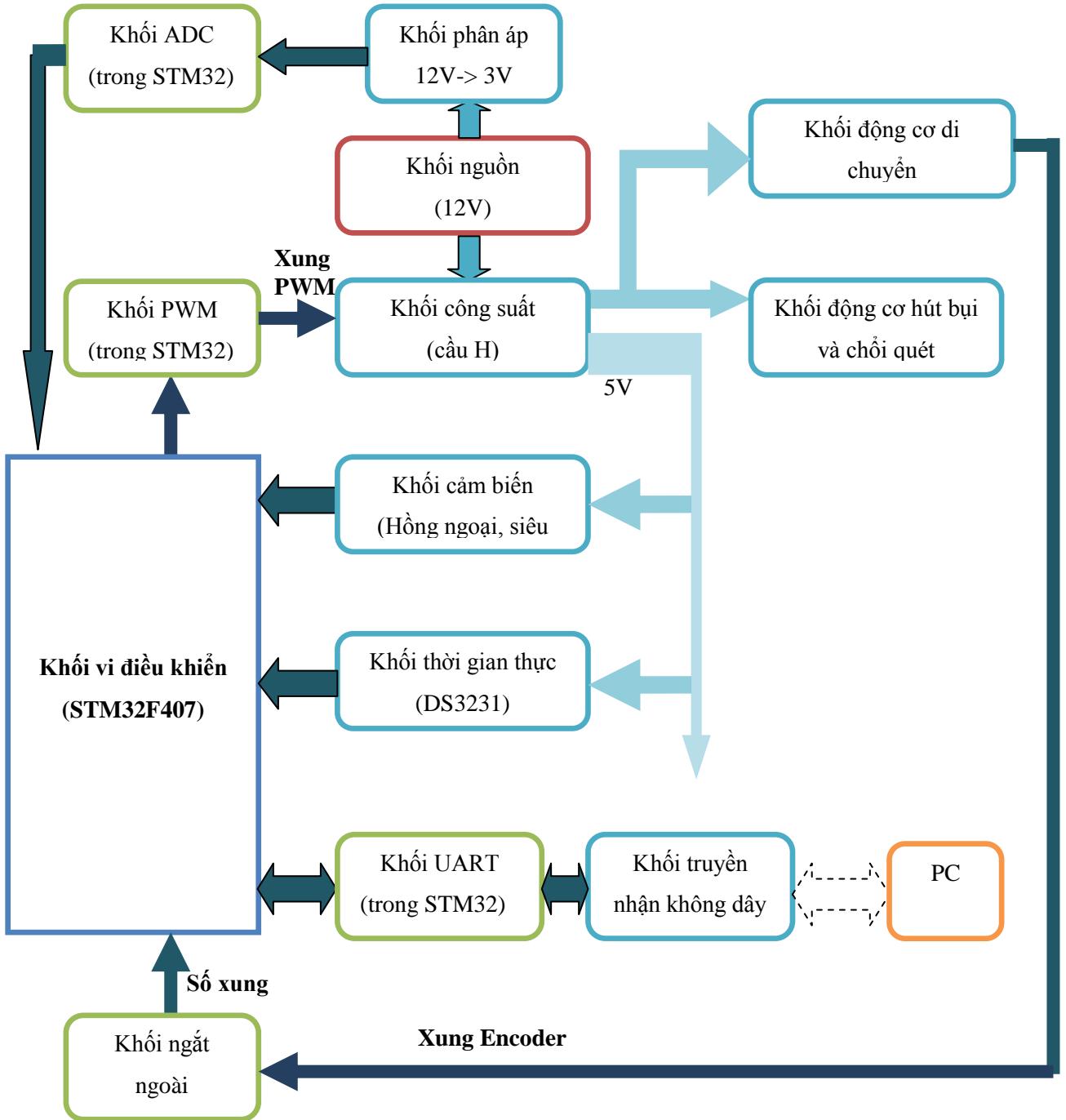
- 5.1 Sơ đồ khối và sơ đồ nguyên lý**
- 5.2 Phương pháp điều chế PWM**
- 5.3 Thuật toán PID**
- 5.4 Thuật toán tránh vật cản, vùng chênh lệch độ cao**
- 5.5 Phương pháp truyền UART qua kết nối không dây**
- 5.6 Xác định giá trị pin**
- 5.7 Lập trình đọc giá trị thời gian thực**
- 5.8 Thuật toán di chuyển của robot**
- 5.9 Giải thuật hệ thống**

5.1 Sơ đồ khối và sơ đồ nguyên lý

5.1.1 Sơ đồ khối

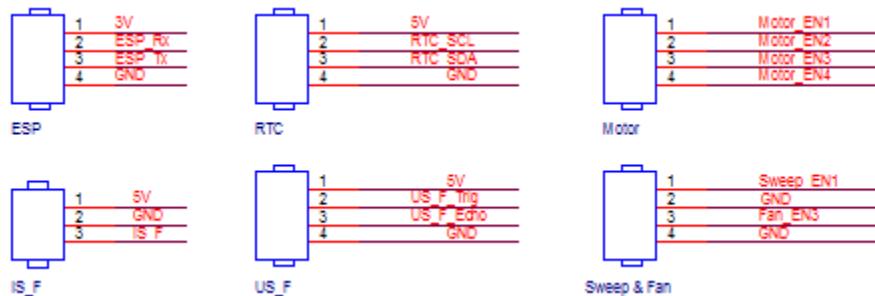
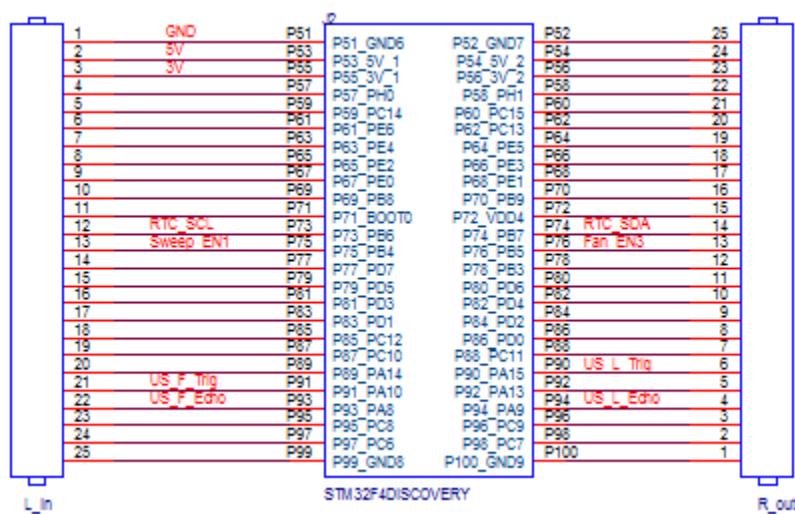
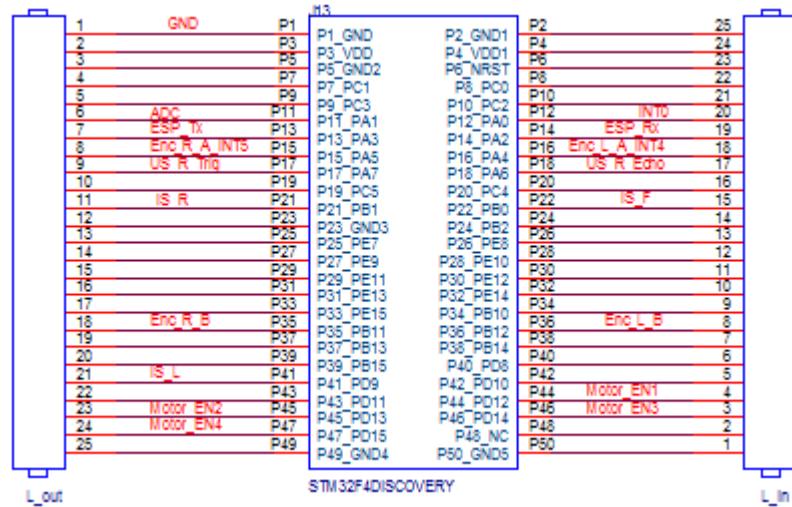
Sau khi tìm hiểu cách hoạt động cũng như nguyên lí của các phương pháp, nhóm tiến hành lắp đặt, kết nối cảm biến, nguồn, vi điều khiển,... để tạo nên hệ thống hoàn chỉnh. Sơ đồ thiết kế gồm các khối như sau:

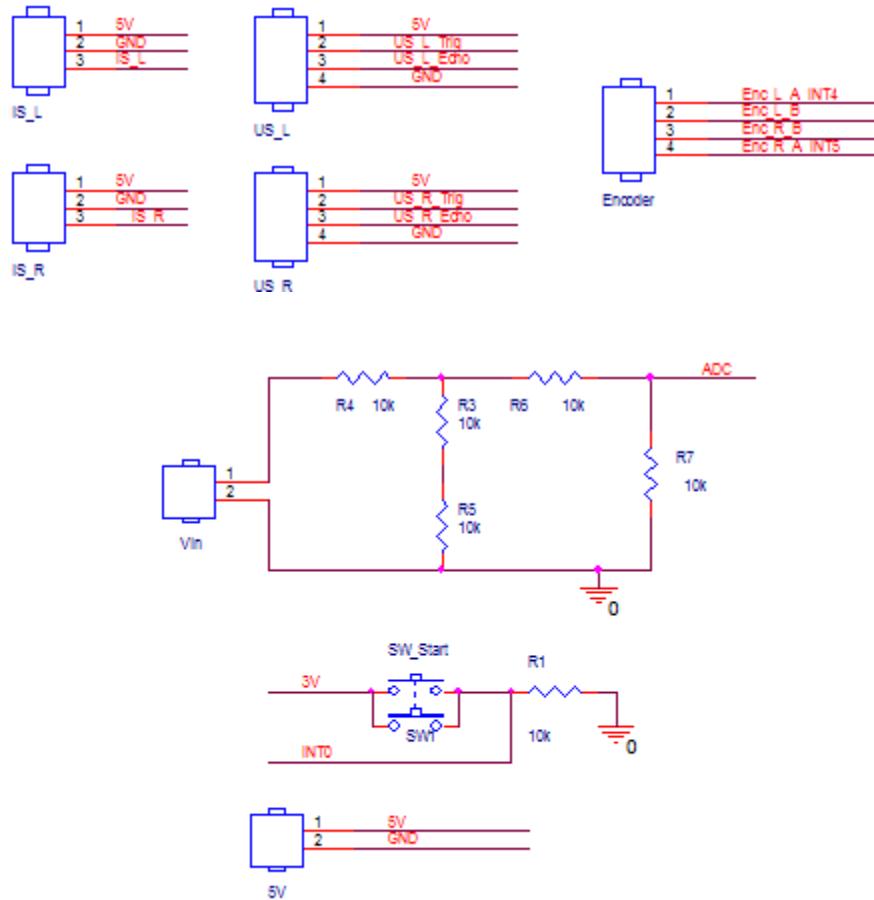
- *Khối nguồn*: Sử dụng pin Lipo 12V cấp nguồn cho vi điều khiển, IC, các cảm biến, động cơ cho xe chạy.
- *Khối cảm biến*: Bao gồm cảm biến hồng ngoại, cảm biến siêu âm phát hiện vật cản đưa tín hiệu về vi điều khiển.
- *Khối động cơ, công suất*: giúp robot hoạt động di chuyển, đồng thời điều khiển tốc độ hút và chổi quét và động cơ của robot.
- *Khối thời gian thực*: Module DS3231 gửi tín hiệu về vi điều khiển hiển thị thời gian thực cho robot, giúp ích cài đặt chế độ làm việc.
- *Khối truyền nhận không dây*: Module ESP8266 giúp truyền nhận dữ liệu PC và robot.
- *Khối phân áp*: mạch chuyển áp của pin từ 12V sang 3V phù hợp với đầu vào kênh ADC.
- *Khối xử lý*: Vi điều khiển nhận giá trị cảm biến, xử lý thông tin đưa ra tín hiệu điều khiển. Bên trong Kit có sử dụng một số khối lập trình khác như *Khối PWM*: Điều khiển động cơ, *Khối ngắt ngoài*: đếm xung Encoder, *Khối ADC*: cập nhật giá trị pin của robot.



Hình 5.1. Sơ đồ khối hệ thống

5.1.2 Sơ đồ nguyên lý





Hình 5.2. Sơ đồ nguyên lý

❖ Trong sơ đồ trên thể hiện cách kết nối các module với vi điều khiển thông qua các header và bus dây. Cụ thể các ngõ ra của kit vi điều khiển sẽ được hàn riêng với từng chân header, sau đó nhóm tiến hành liên kết các module chức năng với các chân header đó thông qua bus dây. Việc kết nối bằng bus dây nhằm tạo sự linh hoạt trong quá trình lập trình trong trường hợp có sự thay đổi các chân sử dụng cũng như vị trí đặt các module trong mô hình robot.

5.2 Phương pháp điều chế PWM

5.2.1 Giới thiệu thuật toán

Phản định nghĩa của phương pháp điều chế PWM (hay còn gọi là băm xung) đã được giới thiệu sơ lược ở phần trước. Dựa trên nguyên lý bên trên mà trong STM32F4

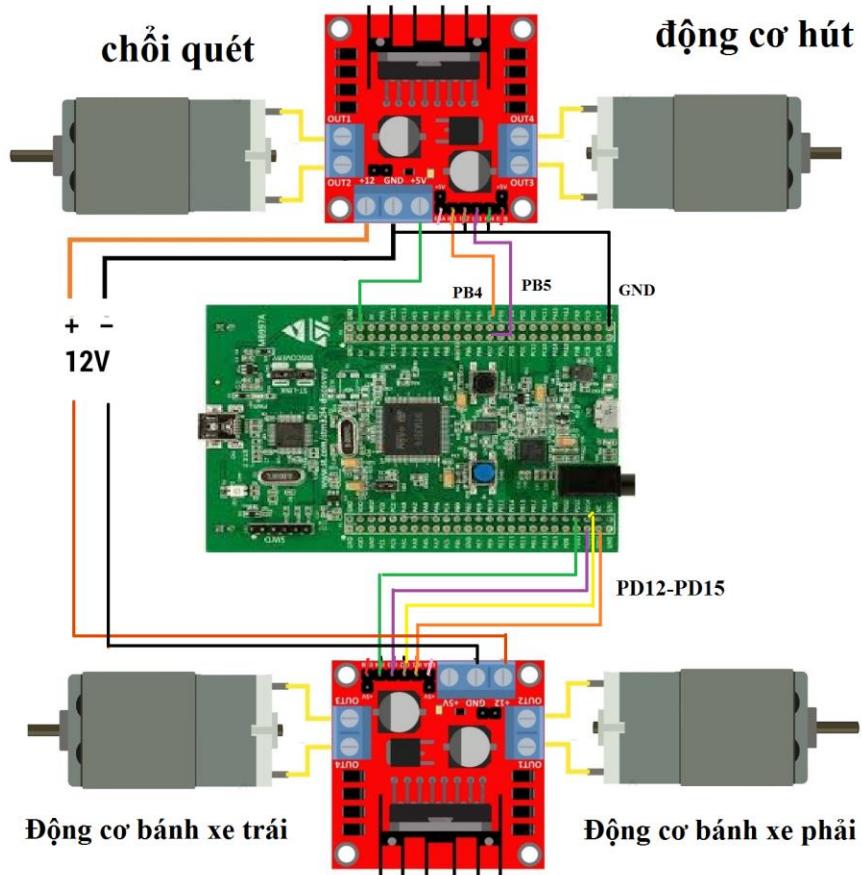
hay các loại vi điều khiển khác điều hổ trợ bộ tạo độ rộng xung tự động mà ta không phải tốn thời gian bật tắt chân IO, có thể thiết lập chu kỳ, tần số, độ rộng xung và một số chức năng đặc biệt.

Trong đề tài này, nhóm cũng đã lập trình sử dụng timer của vi điều khiển để tạo xung PWM đồng thời kết hợp một số module mạch lái (L298) để cung cấp điện áp cũng như điều chỉnh tốc độ, hướng của động cơ di chuyển và một số motor khác được dùng trong robot.

5.2.2 *Ứng dụng vào robot*

Robot di chuyển nhờ hai động cơ encoder gắn với bánh xe. Mỗi động cơ được điều khiển tốc độ và chiều quay nhờ 2 chân PWM xuất ra từ vi điều khiển thông qua mạch cầu H (L298). Trong luận văn, nhóm chọn chân PD12 và PD13 là chân PWM cho động cơ bên phải và PD14, PD15 là chân PWM cho động cơ bên trái. Ngoài ra, nhóm còn sử dụng thêm 2 chân PWM là PB4 và PB5 để điều khiển tốc độ cho chổi quét được lắp dưới để robot và động cơ hút bụi. Vì đây là những động cơ quay một chiều xác định trước nên chỉ cần một chân PWM, kèm theo chân GND đi vào mạch lái để điều chỉnh tốc độ và hướng quay .

Các module mạch lái L298 sẽ được cấp nguồn 12V từ pin Lipo, không chỉ đóng vai trò điều khiển động cơ mà còn ổn áp cấp nguồn 5V cho vi điều khiển hoạt động. Đặc biệt trong quá trình lắp ráp, chân GND của module không chỉ nối với nguồn pin mà còn phải được nối với GND của vi điều khiển. Đây là điểm cần lưu ý vì nếu không lắp nối một cách chính xác, chúng ta không thể sử dụng PWM để thay đổi áp ra từ module.



Hình 5.3. Sơ đồ nối dây động cơ

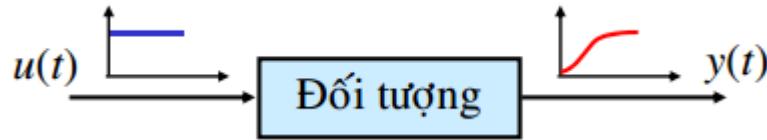
➤ **Nhận xét:** Việc áp dụng PWM trong điều khiển động cơ giúp ta có thể dễ dàng và chủ động hơn trong điều chỉnh được hướng cũng như tốc độ động cơ thông qua các chân vi điều khiển. Nhưng để robot di chuyển một cách chính xác, bên cạnh phương pháp PWM, ta còn cần phải kết hợp với thuật toán PID sẽ được giới thiệu ở phần sau.

5.3 Thuật toán PID

5.3.1 Giới thiệu thuật toán

Bộ điều khiển PID (hay bộ điều khiển vi tích phân tỉ lệ) là một cơ chế phản hồi vòng điều khiển tổng quát, là bộ điều khiển được sử dụng nhiều nhất trong các bộ điều khiển phản hồi. Bộ điều khiển PID sẽ tính toán giá trị sai số là hiệu giữa giá trị đặt và

giá trị phản hồi về. Nó sẽ thực hiện giảm tối đa sai số bằng cách điều chỉnh giá trị điều khiển đầu vào, các giá trị này là các thông số K_p , K_i và K_d sao cho đáp ứng của hệ thống về gần nhất với giá trị đặt.

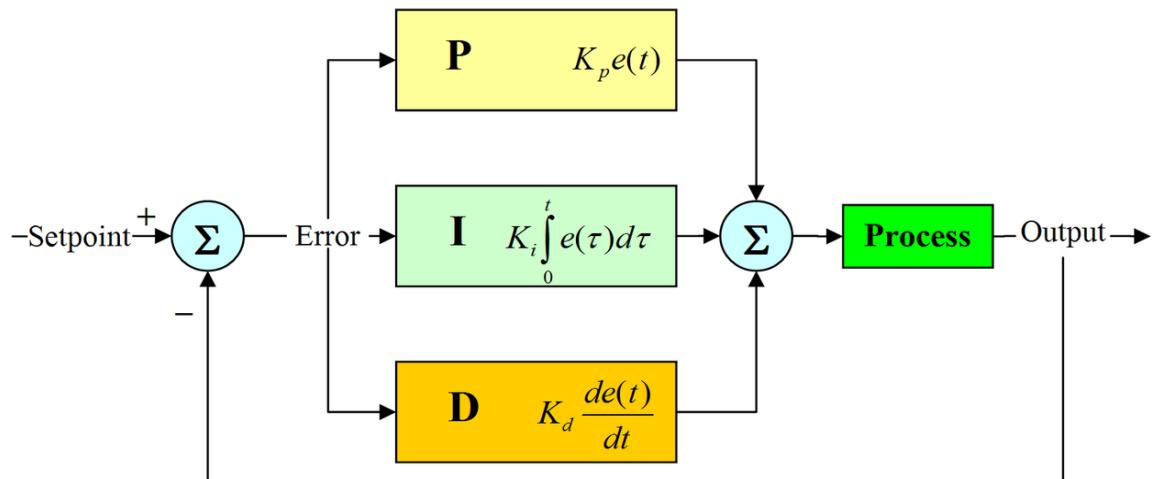


Hình 5.4. Sơ đồ liên hệ

Giải thuật tính toán bộ điều khiển PID bao gồm 3 thông số riêng biệt đại diện cho 3 khâu: tỷ lệ (P), tích phân(I) và vi phân (D). Trong đó:

- Giá trị *tỷ lệ* xác định tác động của sai số hiện tại
- Giá trị *tích phân* xác định tác động của tổng các sai số quá khứ
- Giá trị *vi phân* xác định tác động của tốc độ biến đổi sai số

Tổng của ba tác động này dùng để điều chỉnh quá trình thông qua một phần tử điều khiển. Nhờ vậy, những giá trị này có thể làm sáng tỏ về quan hệ thời gian: P phụ thuộc vào sai số hiện tại, I phụ thuộc vào tích lũy các sai số quá khứ và D dự đoán các sai số tương lai dựa vào tốc độ thay đổi hiện tại.



Hình 5.5. Sơ đồ bộ điều khiển PID

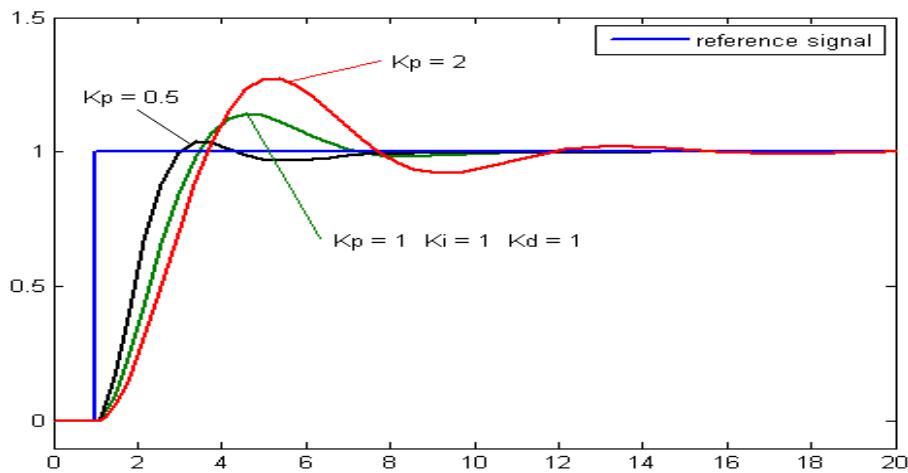
Hàm truyền bộ điều khiển PID

$$G_{PID}(s) = K_p + \frac{K_I}{s} + K_D s$$

$$\text{Hay } G_{PID}(s) = K_p \left(1 + \frac{1}{T_I s} + T_D s \right)$$

Bằng cách điều chỉnh 3 hằng số trong giải thuật của bộ điều khiển PID, ta có thể ứng dụng bộ điều khiển này vào những dự án, công việc có yêu cầu cụ thể.

❖ Khâu tỷ lệ

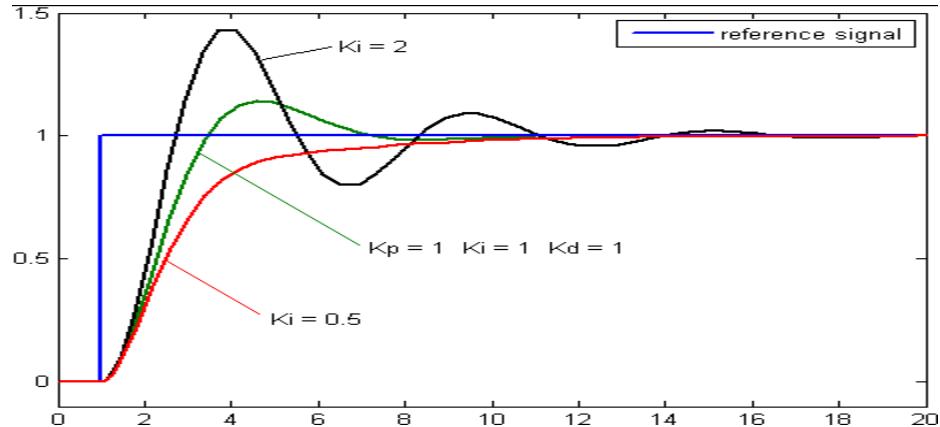


Hình 5.6. Đáp ứng của khâu tỷ lệ

Khâu tỷ lệ (hay còn gọi là độ lợi) làm thay đổi giá trị đầu ra, tỉ lệ với giá trị sai số hiện tại. Đáp ứng tỉ lệ có thể được điều chỉnh bằng cách nhân sai số đó với một hằng số K_p , được gọi là độ lợi tỷ lệ.

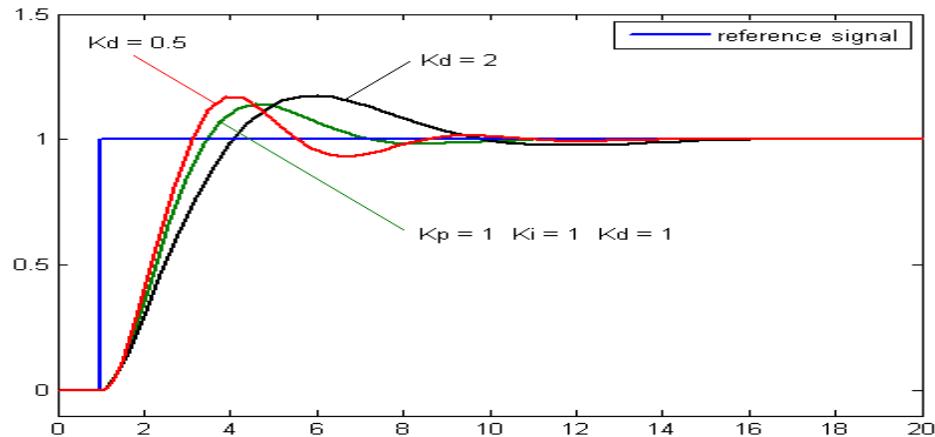
❖ Khâu tích phân

Phân phối của khâu tích phân tỷ lệ với cả biên độ sai số lẫn quãng thời gian xảy ra sai số. Khâu tích phân nếu cộng thêm khâu tỷ lệ sẽ tăng tốc chuyển động của quá trình tới điểm đặt và khử số dư sai số ổn định với một tỷ lệ chỉ phụ thuộc vào bộ điều khiển. Tuy nhiên, do khâu tích phân là đáp ứng của sai số tích lũy trong quá khứ, nó có thể khiến giá trị hiện tại vọt lố qua giá trị đặt.



Hình 5.7. Đáp ứng của khâu tỷ lệ

❖ Khâu vi phân



Hình 5.8. Đáp ứng khâu vi phân

Tốc độ thay đổi của sai số quá trình được tính toán bằng cách xác định độ dốc của sai số theo thời gian (tức là đạo hàm bậc 1 theo thời gian) và nhân tốc độ này với độ lợi tỷ lệ K_D . Khâu vi phân làm chậm tốc độ thay đổi của đầu ra bộ điều khiển và đặc tính này là đáng chú ý nhất để đạt tới giá trị đặt của bộ điều khiển. Do đó, điều khiển vi phân được sử dụng để làm giảm biên độ vọt lồ được tạo ra bởi thành phần tích phân và tăng cường tính ổn định của bộ điều khiển. Tuy nhiên, khâu này sẽ nhạy hơn đối với nhiễu trong sai số và có thể khiến quá trình trở nên không ổn định nếu nhiễu và độ lợi vi phân đủ lớn.

Tóm lại:

- Độ lợi tỷ lệ (K_p): giá trị càng lớn thì đáp ứng càng nhanh do đó sai số càng lớn và khi giá trị đạt tới mức quá lớn sẽ dẫn đến hệ thống mất ổn định và dao động.
- Độ lợi tích phân (K_i): giá trị càng lớn kéo theo sai số ổn định bị khử càng nhanh. Đổi lại là vọt lô càng lớn.
- Độ lợi vi phân (K_d): giá trị càng lớn càng giảm độ vọt lô, nhưng lại làm chậm đáp ứng quá độ và có thể dẫn đến mất ổn định.

Tác động của mỗi thông số điều khiển được trình bày trong bảng sau đây

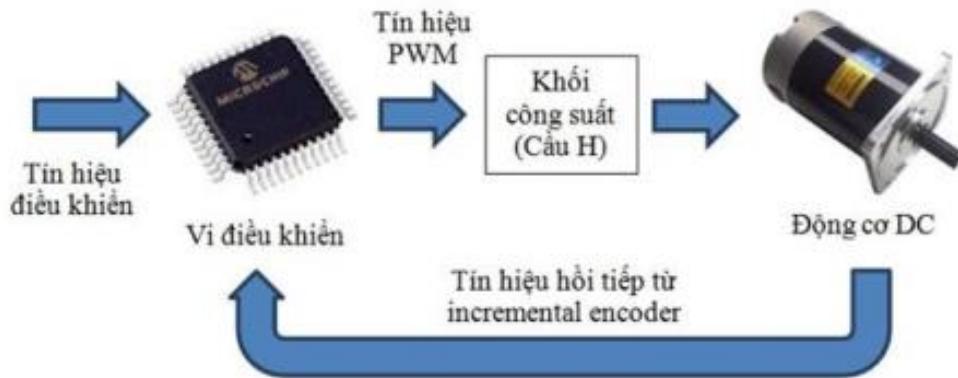
	Thời gian lên	Độ vọt lô	Thời gian quá độ	Sai số xác lập
K_p	Tăng	Tăng	Thay đổi nhỏ	Tăng
K_i	Giảm	Tăng	Tăng	Triệt tiêu
K_d	Thay đổi nhỏ	Giảm	Giảm	Thay đổi nhỏ

Bảng 5.1. Bảng điều chỉnh PID

⇒ Mỗi hệ số sẽ tác động đến chất lượng hệ thống theo một chiều hướng nhất định. Và các hệ số này có mối liên hệ tác động lẫn nhau. Do đó ta cần phải điều chỉnh các hệ số một cách thích hợp với từng hệ thống để đạt được chất lượng đáp ứng tốt nhất.

5.3.2 **Ứng dụng PID vào ổn định hướng đi**

Trong đề tài này, nhóm sử dụng PID để điều chỉnh tốc độ cho động cơ, giúp cho robot có thể di chuyển ổn định theo giá trị đặt cho trước. Bên cạnh đó, các bài toán được đặt ra là có thể giúp robot đi theo đường thẳng không chêch hướng và quay với góc 90^0 một cách chính xác. Đây cũng là những vấn đề quan trọng cần được giải quyết trong quá trình vận hành của robot, giúp sản phẩm hoạt động tốt hơn, đạt được hiệu quả cao và tối ưu trong giải thuật di chuyển. Tiếp theo trong phần báo cáo, nhóm sẽ trình bày cụ thể hướng xử lí cho từng vấn đề nêu trên.



Hình 5.9. Sơ đồ phần cứng sử dụng PID

Giải thuật chung vẫn sẽ là ứng dụng phương pháp PID vòng kín, trong đó vai trò trung tâm là vi điều khiển. Kit STM32F4 sẽ được lập trình nhận tín hiệu điều khiển (được thiết lập sẵn) và tín hiệu hồi tiếp từ động cơ thông qua encoder (cụ thể là xung encoder) để tính toán giá trị PWM cần thiết xuất ra khối công suất điều khiển động cơ đạt tốc độ hay vị trí mong muốn. Sự khác nhau giữa các cách giải quyết vấn đề chính là tín hiệu điều khiển và dữ liệu hồi tiếp, vi điều khiển sẽ biến đổi số xung của động cơ trả về thành những dữ liệu phù hợp trước khi đưa vào hệ thống PID vòng kín.

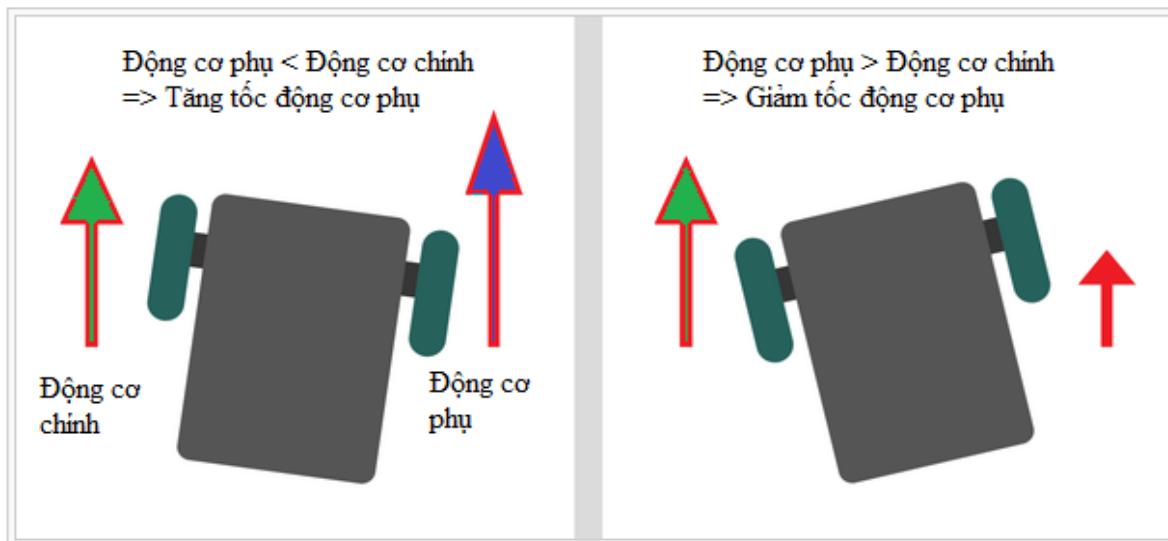
❖ Robot đi với tốc độ cài đặt trước

Với mục đích điều khiển cho robot đi đúng tốc độ mong muốn, tín hiệu đầu vào sẽ là vòng/phút. Để đếm được xung encoder, nhóm lập trình sử dụng các ngắt ngoài của vi điều khiển, lần lượt nối 2 chân ra của encoder của 2 động cơ vào chân PA4 (bánh xe trái) và PA5 (bánh xe phải). Khi có xung hồi tiếp ở chân ngắt, chương trình sẽ nhảy tới ngắt ngoài và thực hiện đếm xung lên. Xung trả về sẽ được đếm trong khoảng thời gian lấy mẫu là 20ms (dùng ngắt timer) và được tính toán ra cùng đơn vị vòng/phút như giá trị đặt. Dữ liệu sau khi được chuyển đổi đưa vào giải thuật PID, vi điều khiển sẽ tính sai số giữa giá trị đặt và giá trị hiện tại cho ra được giá trị PWM thích hợp để điều khiển động cơ, đưa sai số tiến về giá trị nhỏ nhất. Đây là phương pháp khá phổ biến trong điều khiển tốc độ động cơ hiện nay vì dễ lập trình ứng dụng và tính hiệu quả cao.

❖ Robot đi thẳng

Đi thẳng không chêch hướng là một trong những bài toán khó khăn khi lập trình hướng di chuyển cho robot. Nguyên nhân chính là do bánh xe robot không đi cùng tốc độ chính xác như nhau. Ngay cả khi chúng ta gửi cùng một giá trị công suất (cùng giá trị PWM) thì nó vẫn không đảm bảo rằng mỗi bánh xe sẽ di chuyển cùng tốc độ. Đó là do các biến số nhỏ như lực cản, ma sát và sự khác biệt trong sản xuất. Một động cơ ở công suất 50% sẽ không thể đi cùng tốc độ với một động cơ khác ở công suất 50%. Dù có sử dụng PID tốc độ cho cả 2 bánh xe thì chúng sẽ đi lệch nhau trong thời gian tăng tốc (hay thời gian quá độ) hoặc độ lệch sẽ được tích lũy dần trong thời gian đáp ứng khi tốc độ 2 xe di chuyển quanh giá trị đặt trước. Đây là phương pháp ban đầu mà nhóm đã ứng dụng nhưng không giải quyết được vấn đề đặt ra ở đây.

Có một số phương pháp để giải quyết vấn đề này và qua quá trình tìm hiểu nhóm chúng em chọn điều khiển theo mô hình “Master - Slave”. Giải thuật chính của phương pháp này là chọn một động cơ là ‘động cơ chính’ di chuyển với tốc độ đặt (dùng PID tốc độ) và ‘động cơ phụ’ thay đổi để đảm bảo đi cùng tốc độ động cơ chủ, cụ thể là vẫn dùng PID tốc độ nhưng tín hiệu điều khiển (giá trị đặt) của động cơ phụ là tốc độ hiện tại thu được của ‘động cơ chính’.

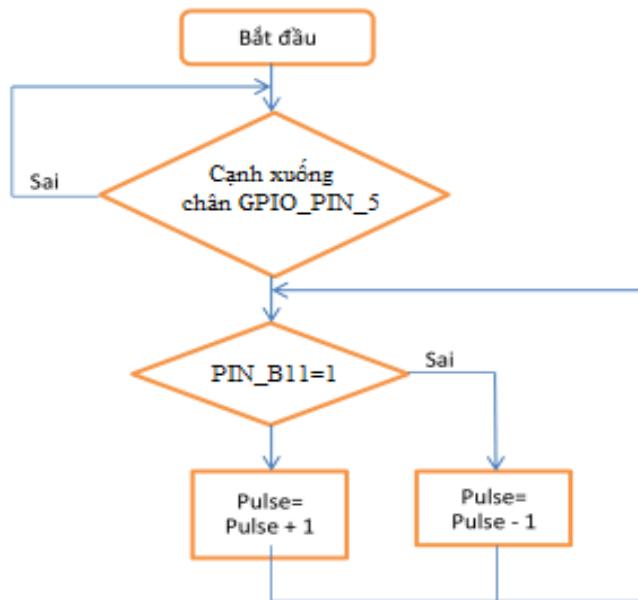


Hình 5.10. Hình ảnh mô phỏng cho thuật toán Master-Slave

Trong đề tài này chúng em sẽ sử dụng động cơ bên trái làm động cơ chính và động cơ bên phải làm động cơ phụ. Đây là phương pháp đơn giản, dễ sử dụng vì khi biết tốc độ của từng bánh xe, chúng ta có thể so sánh chúng, tính toán xuất xung PWM nhiều hơn hoặc ít hơn cho một bánh xe để bắt kịp / giảm tốc độ bằng với cấp độ của động cơ khác. Điều này sẽ làm cho robot lái thẳng hơn rất nhiều.

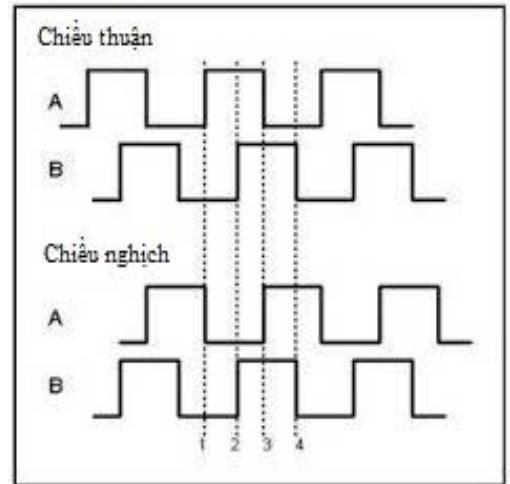
❖ Robot quay trái/phải 90^0

Tín hiệu đặt và hồi tiếp trong trường hợp này chỉ là số xung. Nhóm sẽ tiến hành thực nghiệm và lấy mẫu số xung của bánh xe trả về khi cho xe quay 90^0 rồi thiết lập giá trị đặt cố định. Trong quá trình quay, một bánh sẽ được giữ đứng yên, không di chuyển, bánh còn lại sẽ di chuyển lùi về phía sau để thực thi nhiệm vụ chuyển hướng robot. Việc sử dụng PID giúp vi điều khiển điều chỉnh chiều quay động cơ giúp bánh xe quay và trả về đúng vị trí mong muốn. Chính vì vậy trong giải thuật này, chúng ta cần phải biết được chiều quay của động cơ và thực hiện đếm xung theo sơ đồ giải thuật sử dụng ngắn ngoài của vi điều khiển như sau:



Hình 5.11. Sơ đồ giải thuật đếm xung của động cơ phải

Theo lí thuyết, 2 kênh A, B của encoder sẽ lệch pha nhau 90^0 . Cho nên khi kết hợp 2 kênh này ta có thể xác định được chiều quay của động cơ một cách dễ dàng. Ví dụ, đối với bánh xe phải, khi có ngắt ngoài ở chân PA5, chương trình sẽ ‘đọc’ giá trị chân PB11 đang ở mức cao hay thấp, từ đó sẽ thực thi tác vụ cộng hay trừ biến đếm Pulse giúp việc xác định vị trí của động cơ thêm chính xác. Tương tự với động cơ trái, nhóm sử dụng chân PB12 để lấy giá trị của kênh encoder còn lại.



Hình 5. 12. Giải đồ xung encoder

➤ **Nhận xét:** Thuật toán PID giúp robot có thể vận hành theo giá trị mong muốn, đạt hiệu quả cao hơn rất nhiều trong quá trình hoạt động. Trong thử nghiệm thực tế, hướng đi robot tương đối ổn định, xe có thể bám sát hướng cần đi. Tuy nhiên vẫn còn hiện tượng rung lắc nhẹ và ‘ngắt quãng’ khi di chuyển nên còn cần phải xử lý và tinh chỉnh các thông số của bộ điều khiển PID sao cho phù hợp.

5.4 Thuật toán tránh vật cản, vùng chênh lệch độ cao

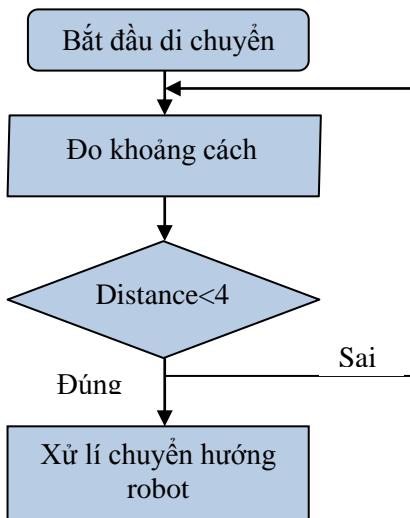
5.4.1 Giới thiệu thuật toán

Trong quá trình vận hành, việc gặp phải vật cản hay những vùng chênh lệch độ cao như bậc thang trong nhà là điều không thể tránh khỏi. Và với những trường hợp nêu trên, nếu chúng ta không tìm cách xử lý và thay đổi hướng đi thích hợp cho robot thì việc va chạm thường xuyên hay rơi rớt trong quá trình hoạt động sẽ dễ gây hư hỏng cho sản phẩm. Chính vì vậy, để robot có thể phản ứng linh hoạt trong môi trường có vật cản thì nhóm đã sử dụng một số loại cảm biến như cảm biến siêu âm SRF04, cảm biến hồng ngoại, encoder giúp robot có thể phát hiện các mối nguy hại và từ đó lập trình giải thuật xử lý chúng một cách hiệu quả.

5.4.2 Ứng dụng vào robot

❖ Cảm biến siêu âm SRF04

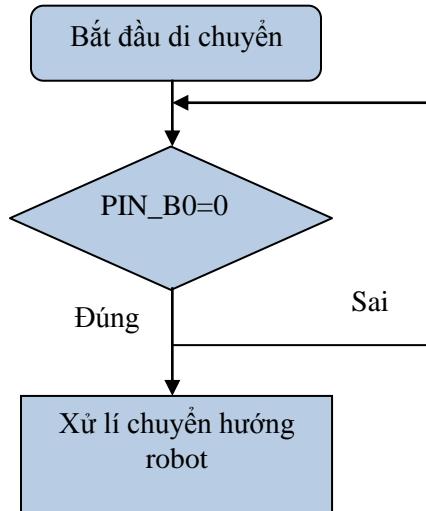
Với nguyên lí đã được giới thiệu kĩ ở phần trước, trong đề tài này, nhóm sử dụng 3 cảm biến này để đo khoảng cách. Cảm biến thứ nhất đặt ở giữa và phía trước thân robot để phát hiện vật cản khi di chuyển (sử dụng chân PA8-PA10 lần lượt là 2 chân Echo và Trigger của cảm biến). Hai cảm biến còn lại đặt hai bên giúp robot xác định vật cản khi quay (sử dụng chân PA9-PA15 & PA6-PA7 cho 2 cảm biến siêu âm còn lại). Khi phát hiện vật cản (khoảng cách trả về nhỏ hơn 4cm) robot sẽ dừng lại và đổi hướng để tránh va chạm.



Hình 5.13. Sơ đồ thuật toán cảm biến siêu âm

❖ *Cảm biến hồng ngoại*

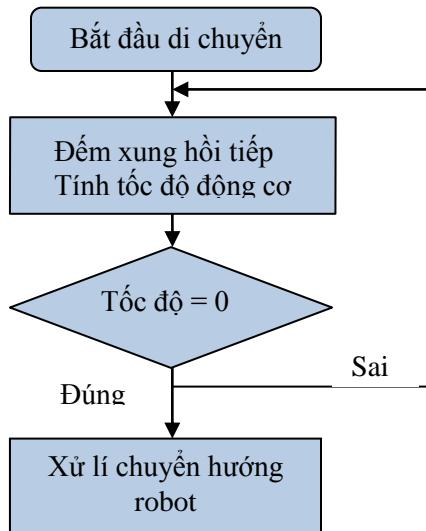
Bên cạnh siêu âm, nhóm còn sử dụng thêm ba cảm biến hồng ngoại. Hai trong số đó được dùng hỗ trợ phát hiện vật cản phía trước cùng với cảm biến siêu âm (chân tín hiệu của 2 cảm biến này lần lượt nối với chân PB1 và PD9). Cảm biến còn lại được đặt ở phía trước và hướng xuống dưới với chân tín hiệu được nối với PB0 giúp robot tránh bị rơi cầu thang hay những nơi chênh lệch độ cao. Trái ngược với cảm biến siêu âm, hồng ngoại này luôn kiểm tra xem phía dưới trước robot có phải mặt đất hay không với khoảng cách cố định thiết lập trước trên module. Khi gặp khoảng chênh lệch độ cao, hồng ngoại sẽ báo không còn thấy ‘vật cản’, lúc đó ta biết có hiểm nguy cần xử lí.



Hình 5.14. Sơ đồ thuật toán cảm biến hồng ngoại

❖ Encoder

Encoder ngoài mục đích xác định tốc độ và vị trí của động cơ, còn giúp robot phát hiện những trường hợp xấu khác làm cho robot ngưng chuyển động như trượt bánh, mắc kẹt. Giải thuật ở đây sẽ là kiểm tra xung encoder đưa về vi điều khiển bằng ngắn, nếu không thấy có tín hiệu của xung trả về khi xe đang trong chế độ di chuyển (động cơ đang được cấp xung PWM) tức là bánh xe đã bị mắc kẹt. Khi đó ta cũng sẽ lập trình để điều khiển robot đổi hướng để thoát khỏi tình huống xấu này.



Hình 5.15. Sơ đồ thuật toán encoder

➤ **Nhận xét:** Sau khi lập trình kết hợp ứng dụng các thuật toán trên, robot di chuyển tránh được các vật cản như mong muốn, ít xảy ra va chạm hơn. Tuy nhiên, trong quá trình hoạt động thực tế, luôn xảy ra nhiều không mong muốn, khiến cho các cảm biến không thể phát hiện được vật cản hay chênh lệch độ cao. Do đó, cần có những biện pháp chống nhiễu khi phát triển sản phẩm, giúp robot hoạt động một cách ổn định nhất.

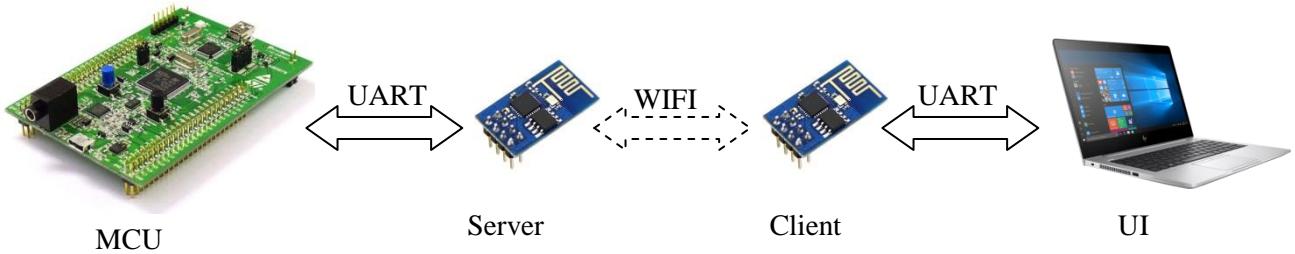
5.5 Phương pháp truyền UART qua kết nối không dây

5.5.1 Giới thiệu phương pháp truyền UART không dây

Việc truyền nhận chuỗi dữ liệu, các thông tin cần thiết như cài đặt, hiển thị giữa sản phẩm điện tử và người dùng đã dần trở thành một xu thế phổ biến trong nền công nghiệp phát triển điện tử ngày nay. Có một số cách để truyền dữ liệu không dây, tùy thuộc vào khoảng cách vị trí, nguồn năng lượng, tài chính như bluetooth, wifi, lora,.. và để đáp ứng cho ứng dụng điều khiển từ xa, nhóm quyết định sử dụng thêm module Wifi ESP8266.

5.5.2 Ứng dụng vào điều khiển và theo dõi robot

Có nhiều phương pháp kết nối ESP để truyền nhận thông tin tùy vào số lượng thiết bị và khả năng lập trình. Ở đây, nhóm đã sử dụng 2 module ESP8266 kết nối wifi với nhau và mỗi module sẽ nhận và truyền tải thông tin với thiết bị mà nó kết nối thông qua giao thức truyền thông UART. Vì việc tìm hiểu và lập trình còn hạn chế nên phương pháp mà nhóm quyết định sử dụng chưa thực sự tối ưu về cả mặt lập trình và kinh tế tuy nhiên nó vẫn có thể giải quyết được yêu cầu truyền nhận không dây đã đề ra. Về tổng quan có thể khái quát mô hình kết nối ESP8266 với vi điều khiển mà nhóm đã thực hiện như sau:



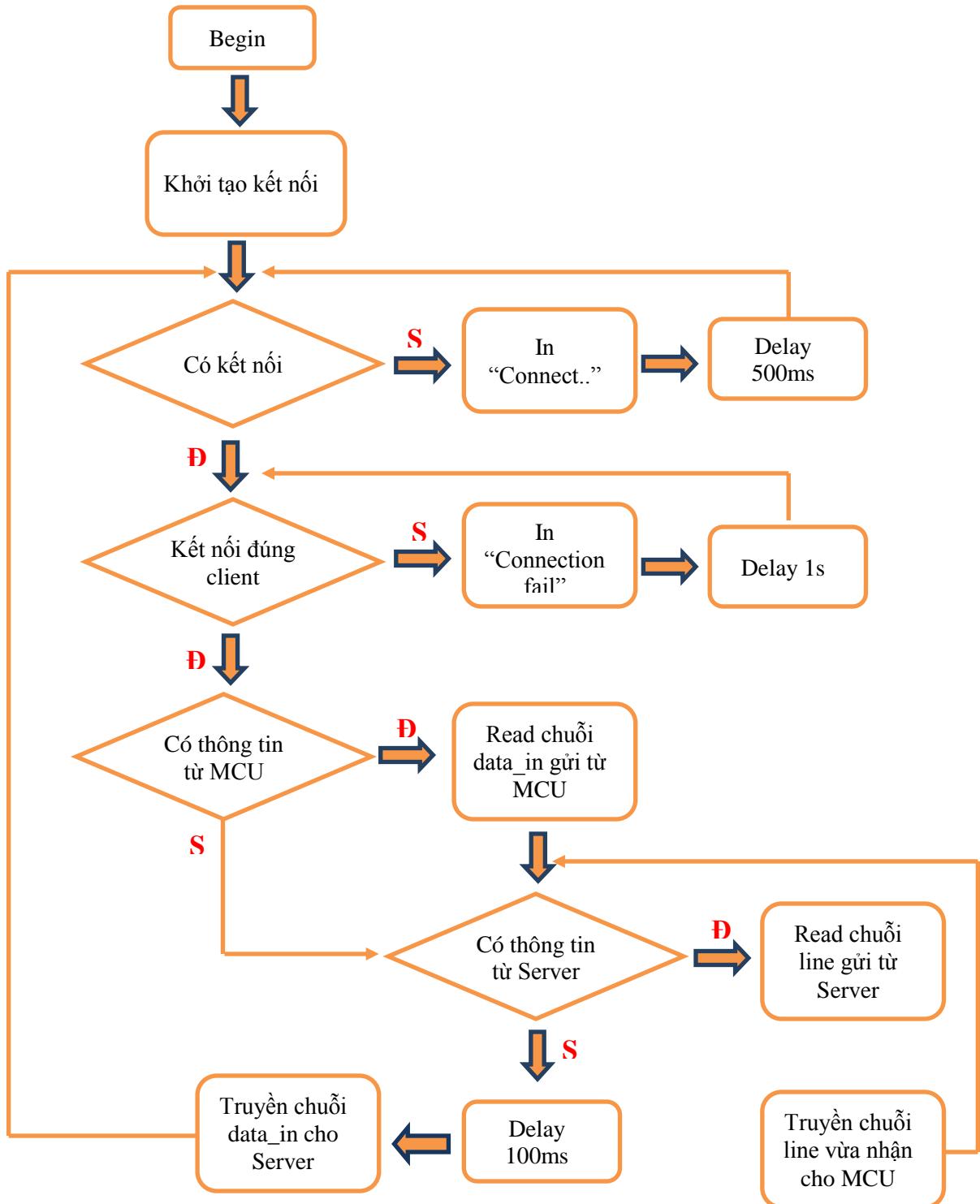
Hình 5.16. Sơ đồ kết nối không dây

Theo như sơ đồ trên thì 1 module ESP sẽ được kết nối với vi điều khiển (chân PA2 và PA3) và module còn lại sẽ được kết nối với máy tính thông qua chuẩn truyền UART. Vì vậy, cần phải cấu hình cho cả vi điều khiển và giao diện người dùng trên máy tính có thể truyền nhận UART với các thông số cơ bản như sau:

<i>Thông số</i>	<i>Kit STM32F4</i>	<i>Giao diện người dùng</i>
Tốc độ Baud	9600	115200
Chiều dài chuỗi	8 bit	8 bit
Parity	Không	Không
Stop bits	1	1
Chiều thông tin	Truyền – Nhận	Truyền – Nhận

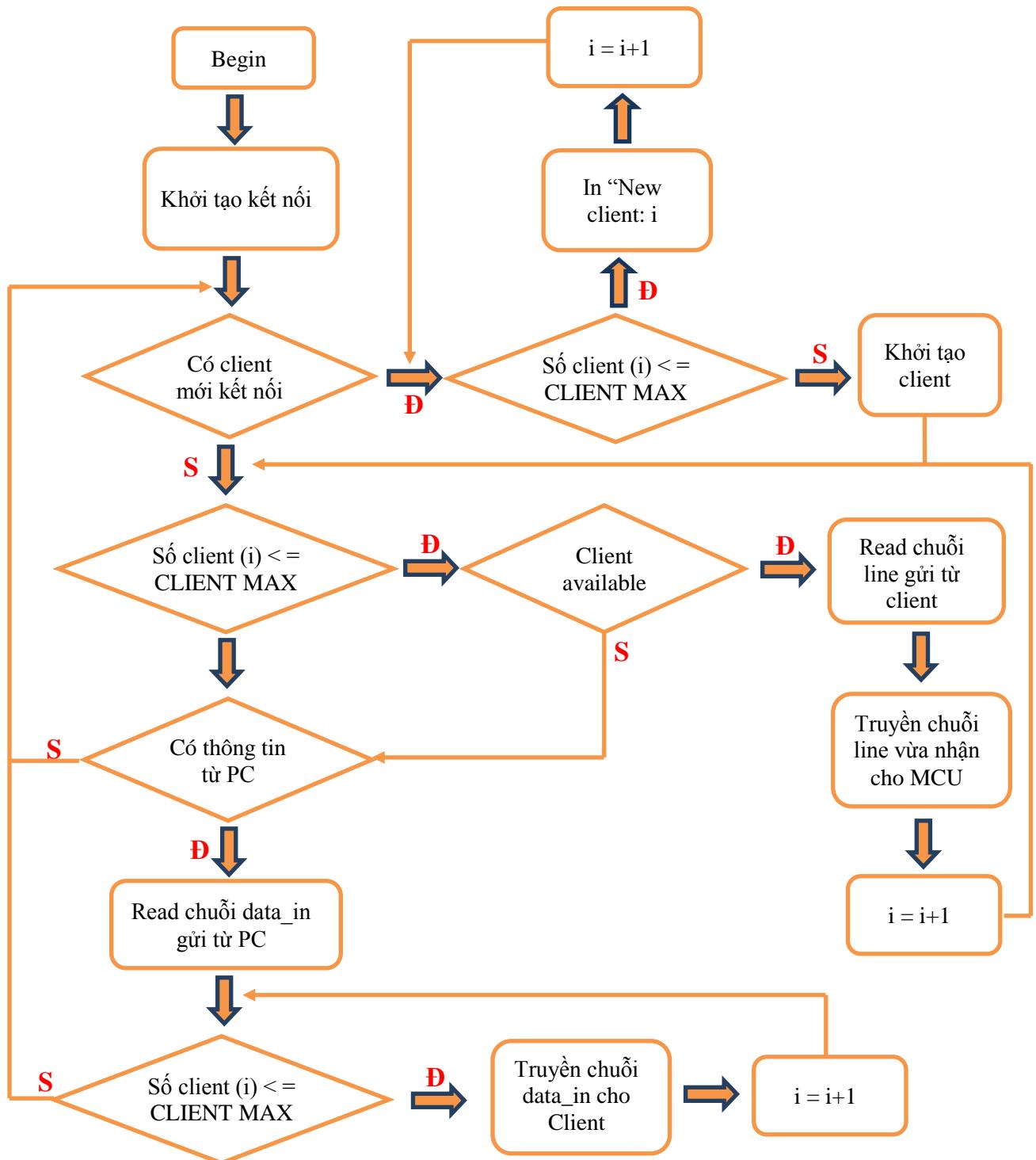
Bảng 5.2. Bảng thiết lập thông số Serial port

Giải thuật Client:



Hình 5.17. Sơ đồ giải thuật client

Giải thuật Server:



Hình 5.18. Sơ đồ giải thuật Server

➤ **Nhận xét:** Việc truyền nhận dữ liệu tương đối hiệu quả, người dùng có thể kết nối và cài đặt thông số cho robot cũng như theo dõi được số liệu quan trọng được gởi lên từ vi điều khiển. Tuy nhiên, mặc dù không đáng kể hay ảnh hưởng nhiều đến việc vận hành sản phẩm nhưng thực nghiệm cho thấy vẫn còn xảy ra độ trễ không mong muốn trong quá trình giao tiếp, đặc biệt là khi thực thi cài đặt thiết lập chế độ cho sản phẩm.

5.6 Xác định giá trị pin

5.6.1 Giới thiệu ADC

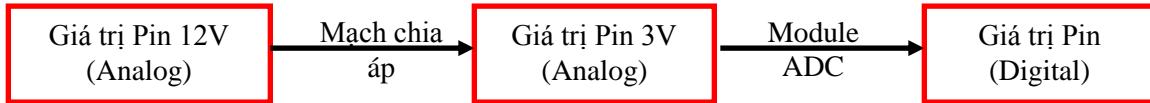
Trong các ứng dụng đo lường và điều khiển bằng vi điều khiển, bộ chuyển đổi tương tự số (hay còn gọi là ADC) là một thành phần rất quan trọng để có thể chuyển đổi các dữ liệu dạng analog trong thế giới của chúng ta (nhiệt độ, điện áp,...) sang tín hiệu số mà vi điều khiển có thể xử lý được. Trong đề tài này, nhóm sử dụng module ADC được dùng để chuyển điện áp pin từ giá trị Vôn (analog) về thành dạng số (digital) để thuận tiện cho việc lập trình xác định, hiển thị và quản lý.

5.6.2 Sử dụng ADC vào đọc giá trị nguồn cấp

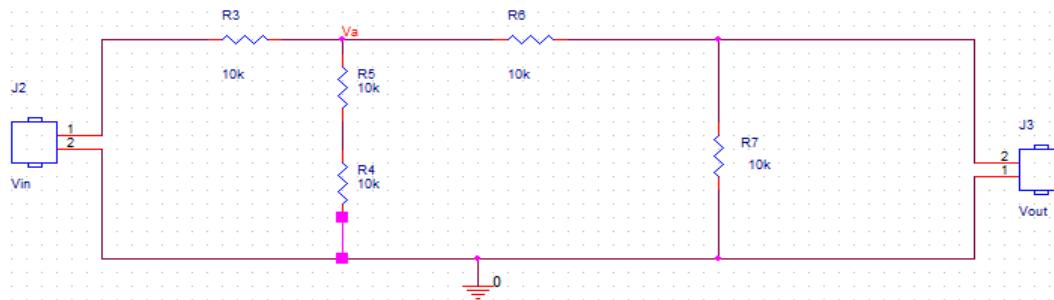
Trong phần lập trình, module ADC được sử dụng là bộ ADC 1, channel 1 (chân PA1) với cấu hình như sau:

- Độ phân giải: 12 bits
- Chế độ quét từng kênh: Disable
- Chế độ quét liên tục: Disable
- Request DMA: không

Với độ phân giải 12bit thì bộ ADC sẽ xuất ra giá trị số từ 0 đến 4096 tương ứng với giá trị điện áp từ 0V đến 3V. Như vậy, độ chia sẽ đủ nhỏ để bộ ADC có thể nhận thấy được sự thay đổi điện áp của nguồn với độ lệch xuống tới 0.0007V, đảm bảo cho việc cập nhật giá trị pin nhanh và chính xác nhất.



Trong thiết kế phần cứng, nhóm đã thiết kế một mạch chia áp đơn giản nhằm chuyển điện áp pin từ 12V về thành 3V để phù hợp với giá trị đầu vào của bộ ADC của vi điều khiển. Đồng thời nhằm bảo vệ vi điều khiển khỏi hư hỏng nếu may có sự cố xảy ra (vì điện áp 3V sẽ an toàn cho vi điều khiển hơn điện áp 12V)



Hình 5.19. Sơ đồ nguyên lý mạch chia áp

Trong đó: $R_a = (R_6 + R_7) / (R_4 + R_5) = (10k + 10k) / (10k + 10k) = 10k$

$$V_a = \frac{R_a}{R_3 + R_a} V_{in} = \frac{10k}{10k + 10k} V_{in} = \frac{1}{2} V_{in}$$

$$V_{out} = \frac{R_7}{R_7 + R_6} V_a = \frac{10k}{10k + 10k} V_a = \frac{1}{2} V_a$$

$$\Rightarrow V_{out} = \frac{1}{4} V_{in}$$

➤ **Nhận xét:** Phương pháp sử dụng ADC để đọc được giá trị pin sản phẩm hoạt động khá tốt trong đè tài, giúp việc xác định giá trị nguồn cung cấp trở nên dễ dàng, thuận tiện cho việc quản lí thời gian sử dụng và sạc pin.

5.7 Lập trình đọc giá trị thời gian thực

5.7.1 Giới thiệu hệ thống thời gian thực

Vì yêu cầu của một robot dân dụng nói chung và một robot hút bụi nói riêng là phải hoạt động được trong điều kiện thời gian thực. Do đó, việc tích hợp cho robot một module thời gian thực là một yêu cầu cần thiết và bắt buộc.

Hầu hết các module thời gian thực hiện nay được bán trên thị trường đều dùng giao thức truyền thông I2C. Vì vậy, để có thể giao tiếp với module thời gian thực (cụ thể nhóm sử dụng là mạch DS3231) thì cần cấu hình cho vi điều khiển để sử dụng được module I2C được hỗ trợ sẵn.

5.7.2 Sử dụng module thời gian thực cho robot

Trong kit STM32F4 có tích hợp sẵn 3 bộ I2C (gồm I2C1, I2C2 và I2C3) với chức năng như nhau. Trong phần lập trình, nhóm đã sử dụng bộ I2C1 với các cấu hình chính như sau: Mode: Mode chuẩn, Tốc độ clock: 100000 Hz, Số bit địa chỉ: 7 bit, Ngắt I2C: Enable, DMA: sử dụng cho I2C_Rx

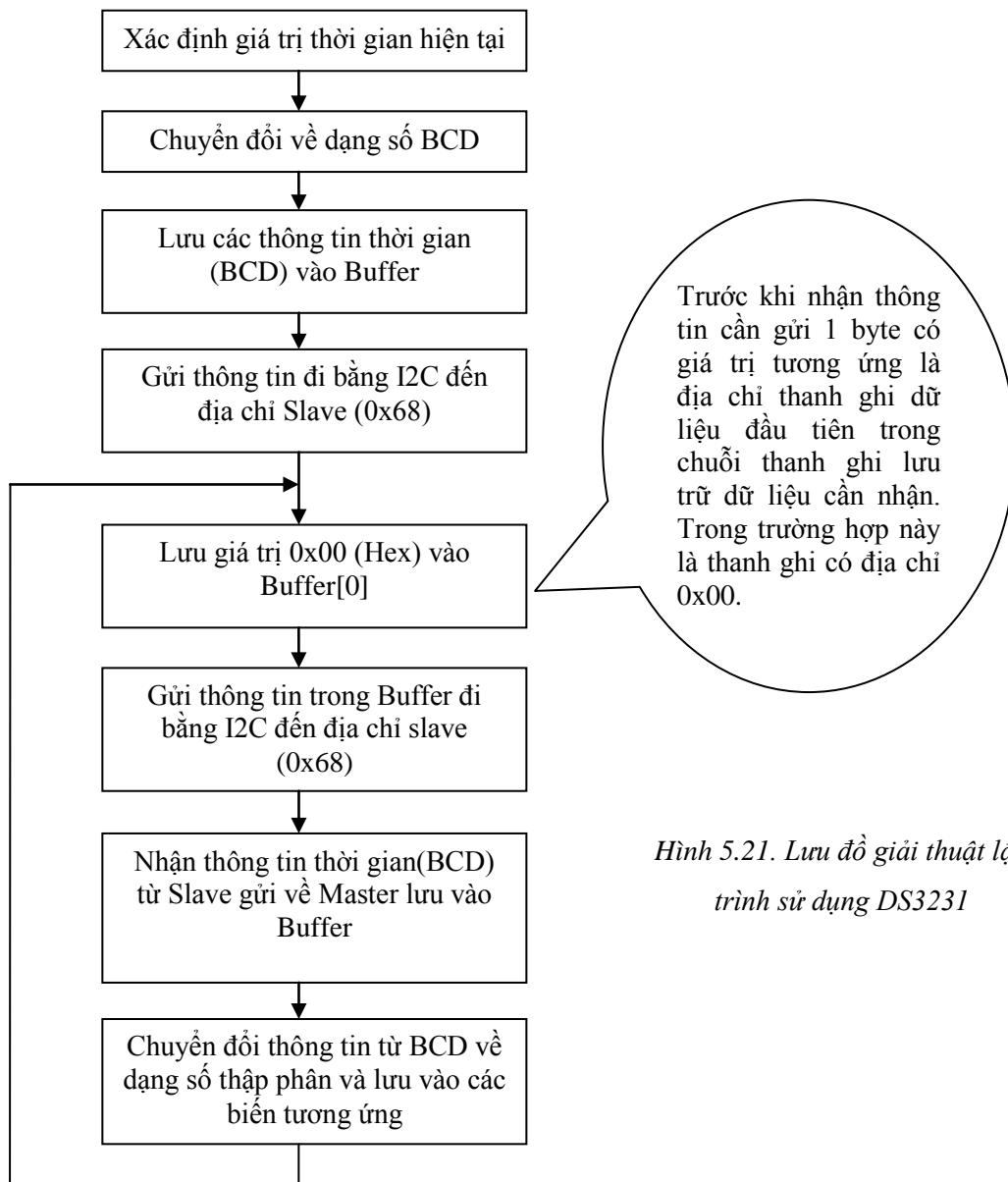
ADDRESS	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0 LSB	FUNCTION	RANGE		
00h	0	10 Seconds			Seconds				Seconds	00–59		
01h	0	10 Minutes			Minutes				Minutes	00–59		
02h	0	12/24	AM/PM	10 Hour	Hour				Hours	1–12 + AM/PM 00–23		
03h	0	0	0	0	0	Day			Day	1–7		
04h	0	0	10 Date		Date				Date	01–31		
05h	Century	0	0	10 Month	Month				Month/ Century	01–12 + Century		
06h	10 Year				Year				Year	00–99		
07h	A1M1	10 Seconds			Seconds				Alarm 1 Seconds	00–59		
08h	A1M2	10 Minutes			Minutes				Alarm 1 Minutes	00–59		
09h	A1M3	12/24	AM/PM	10 Hour	Hour				Alarm 1 Hours	1–12 + AM/PM 00–23		
0Ah	A1M4	DY/DT	10 Date		Day				Alarm 1 Day	1–7		
0Bh	A2M2	10 Minutes			Date				Alarm 1 Date	1–31		
0Ch	A2M3	12/24	AM/PM	10 Hour	Minutes				Alarm 2 Minutes	00–59		
0Dh	A2M4	DY/DT	10 Date		Hour				Alarm 2 Hours	1–12 + AM/PM 00–23		
0Eh	EOSC	BBSQW	CONV	RS2	RS1	INTCN	A2IE	A1IE	Control	—		
0Fh	OSF	0	0	0	EN32kHz	BSY	A2F	A1F	Control/Status	—		
10h	SIGN	DATA	DATA	DATA	DATA	DATA	DATA	DATA	Aging Offset	—		
11h	SIGN	DATA	DATA	DATA	DATA	DATA	DATA	DATA	MSB of Temp	—		
12h	DATA	DATA	0	0	0	0	0	0	LSB of Temp	—		

Figure 1. Timekeeping Registers

Note: Unless otherwise specified, the registers' state is not defined when power is first applied.

Hình 5.20. Datasheet của DS3231

Dữ liệu của DS3231 được lưu trong tổng cộng là 19 thanh ghi, có địa chỉ từ 00h-12h. Đối với ứng dụng thời gian thực thì chỉ cần sử dụng các thông tin về thời gian được lưu trong các thanh ghi từ 00h đến 06h (lần lượt ứng với các giá trị: giây, phút, giờ, thứ, ngày, tháng, năm). Và đây là các thanh ghi Read/Write cho phép cả đọc và ghi thông tin lên thanh ghi. Các thanh ghi còn lại có các chức năng khác như báo thức, thanh ghi trạng thái, điều khiển, nhiệt độ,... những chức năng này trong luận văn này sẽ không được dùng tới, tuy nhiên có thể được dùng để mở rộng ứng dụng của robot trong tương lai.



Lưu ý: Module RTC DS3231 định dạng các thông tin thời gian ở dạng số BCD. Do đó thông tin sau khi nhận được và trước khi gửi đi phải được chuyển đổi về dạng về số BCD rồi mới thực hiện bước tiếp theo.

Module RTC được cài đặt giá trị thời gian mặc định là 0h:0m:0s day0, date0, month0, year0. Vì vậy, ngay lần đầu tiên được sử dụng thì người dùng cần phải cài đặt giá trị thời gian bắt đầu đếm cho nó. Sau lần cài đặt này thì không cần phải thực hiện lại bước này (nếu như không cần điều chỉnh thời gian thực và giả sử độ sai lệch thời gian là không đáng kể) vì trên module RTC đã được tích hợp sẵn 1 viên pin nhảm duy trì cho IC hoạt động trong khoảng thời gian không được sử dụng.

➤ **Nhận xét:** Module RTC DS3231 hoạt động khá tốt trong đề tài này, không chỉ được sử dụng với mục đích hiển thị các thông số thời gian thực mà còn là tiền đề để nhóm có thể phát triển thêm tính năng cài đặt thời gian vận hành hay hẹn giờ làm việc cho sản phẩm.

5.8 Thuật toán di chuyển của robot

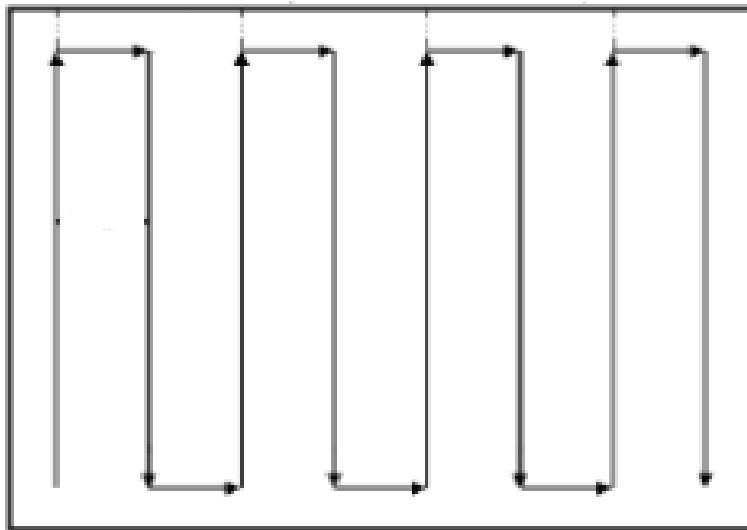
5.8.1 Giới thiệu thuật toán di chuyển

Như đã giới thiệu ở phần trước, hiện nay có rất nhiều thuật toán di chuyển cho robot như xoắn ốc, ngẫu nhiên, zigzag,... mỗi loại đều thích hợp cho môi trường làm việc, không gian đặc trưng riêng và cũng tùy thuộc vào nhu cầu của người dùng. Trong đề tài này, qua việc đánh giá ưu nhược điểm của các phương pháp di chuyển cũng như xem xét khả năng và điều kiện của nhóm, chúng em đã quyết định chọn phương pháp di chuyển zigzag. Mặc dù đây không phải là phương pháp hiệu quả và tốt nhất nhưng là phương pháp được sử dụng rộng rãi với ưu điểm là tối ưu được diện tích sàn nhà được vệ sinh. Tuy nhiên cần phải có được thuật toán PID kết hợp dùng PWM tốt để robot có thể di chuyển ổn định, chính xác nhất.

5.8.2 Ứng dụng thuật toán

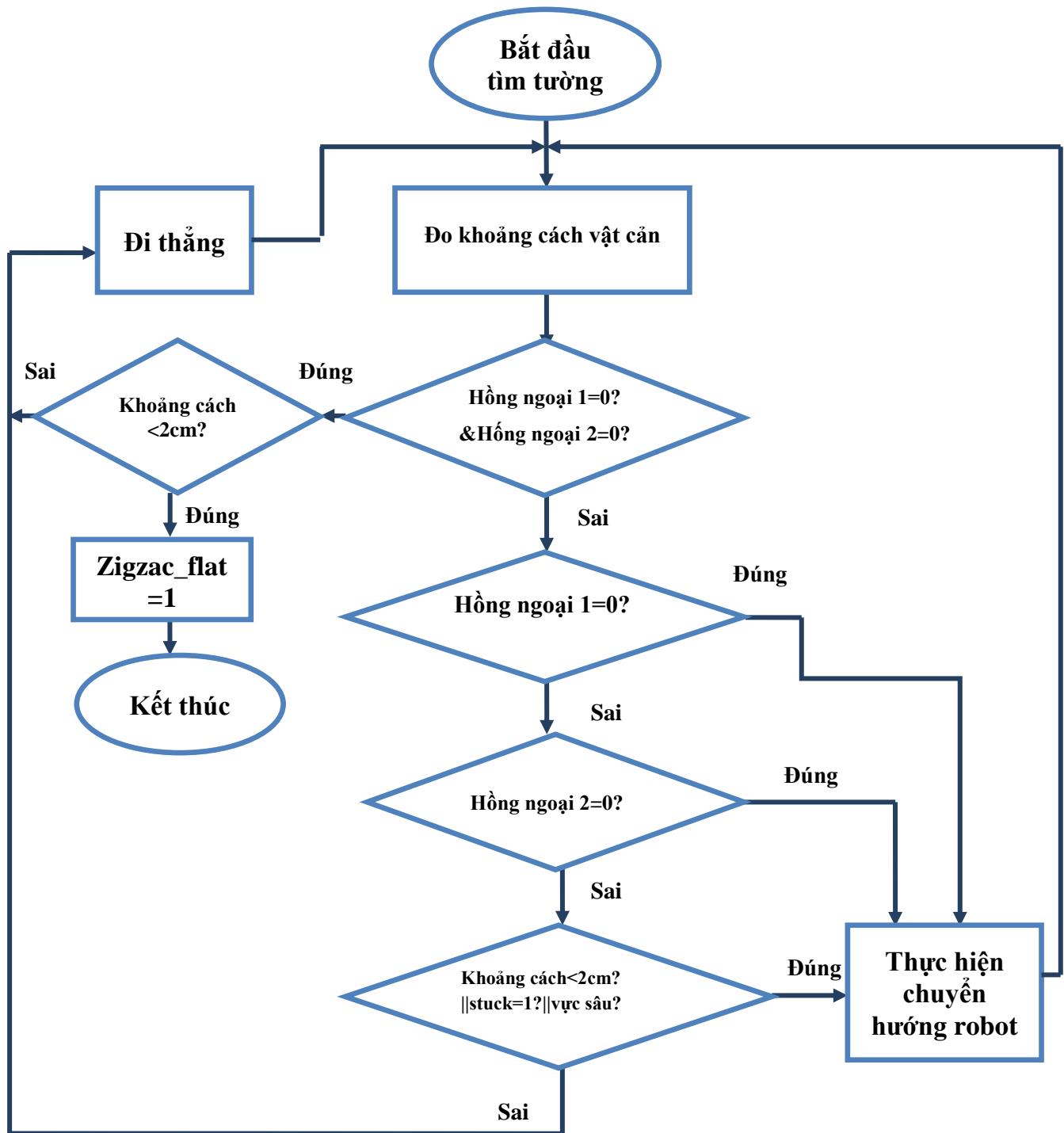
Thuật toán di chuyển Zigzag được áp dụng khi robot đang ở chế độ tự động

được cài đặt trước qua ứng dụng người dùng hay nút nhấn trên nắp sản phẩm. Nhưng để giải thuật Zigzag đạt hiệu quả tốt nhất thì robot phải di chuyển song song với các bức tường của căn phòng cũng như là có thể đi trên đường thẳng và quay trái/phải 90^0 một cách chính xác (các ván đè đã được trình bày, giải quyết ở phần trước). Có như vậy, diện tích được vệ sinh mới đạt tối ưu và lớn nhất.



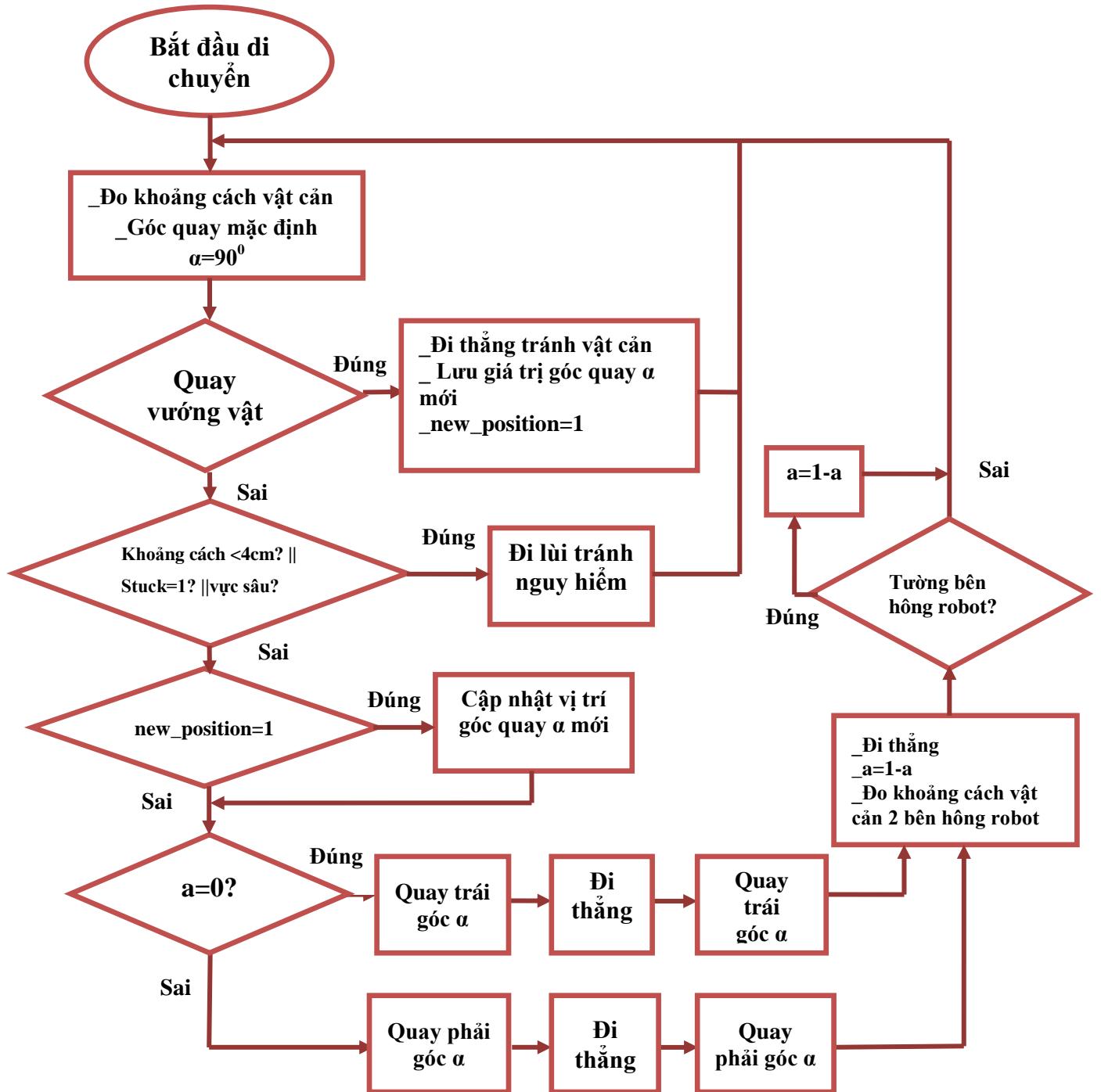
Hình 5.22. Di chuyển zigzag

Do đó, trước khi tiến hành di chuyển Zigzag, robot sẽ được lập trình để tìm thấy một trong những bức tường của căn phòng. Để thực hiện bước dò tìm, robot sẽ di chuyển tự do thẳng về phía trước và vẫn thực hiện né vật cản khi một trong ba cảm biến (gồm một siêu âm và hai hồng ngoại) phát hiện thấy chướng ngại vật. Cờ zigzag sẽ được bật lên 1 khi cả 3 cảm biến đều phát hiện vật cản, tức là lúc này mặc định đã phát hiện được bức tường. Quá trình này sẽ được thể hiện cụ thể trong lưu đồ giải thuật dưới đây.



Hình 5.23. Lưu đồ giải thuật tìm tường nhà

Sau khi đã dò tìm được tường nhà, robot thực hiện di chuyển zigzag theo lưu đồ giải thuật sau:



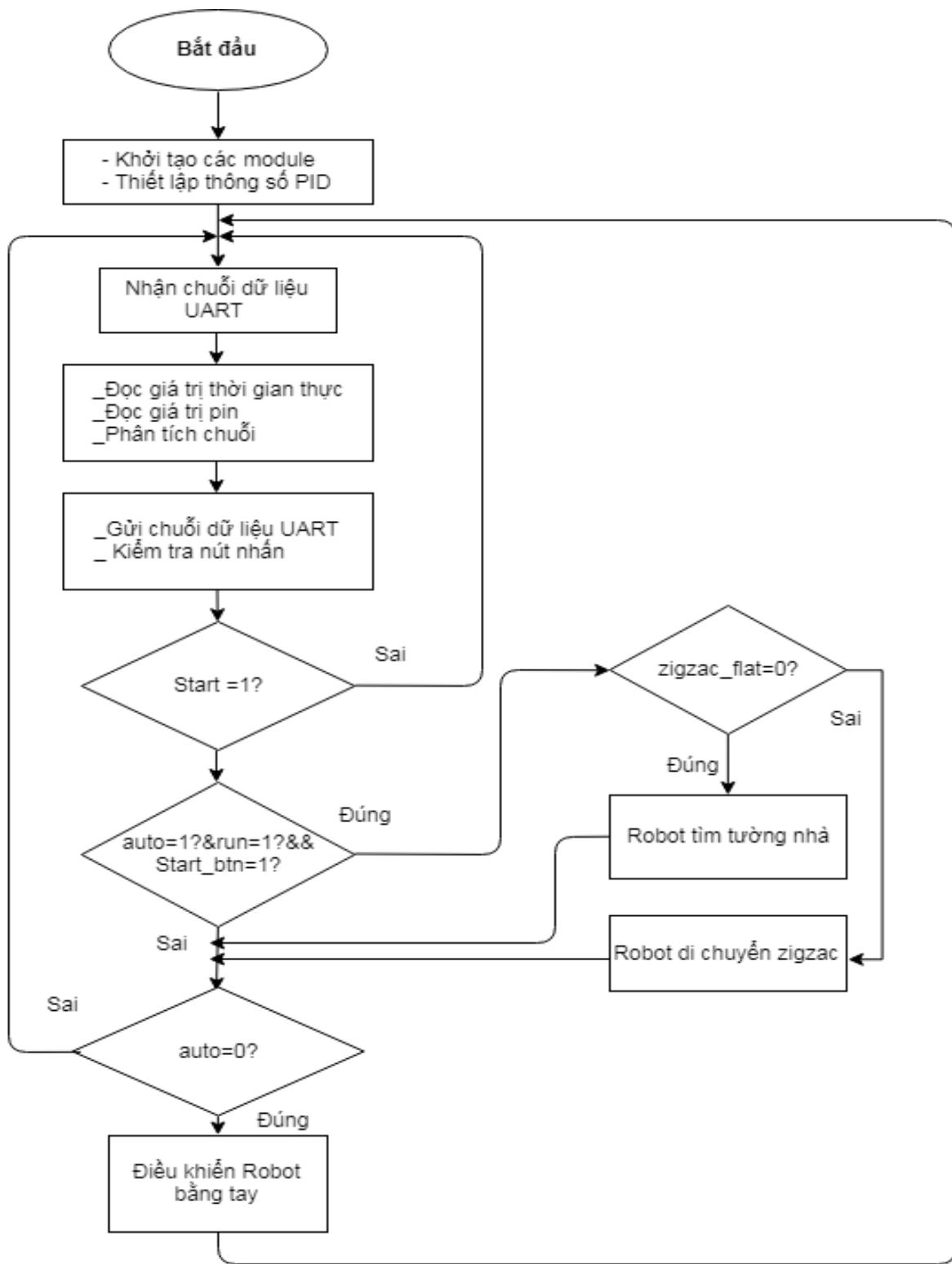
Hình 5.24. Lưu đồ giải thuật zigzag

* *Biến Stuck: bằng 1 khi Encoder phát hiện robot bị mắc kẹt trong lúc di chuyển.*

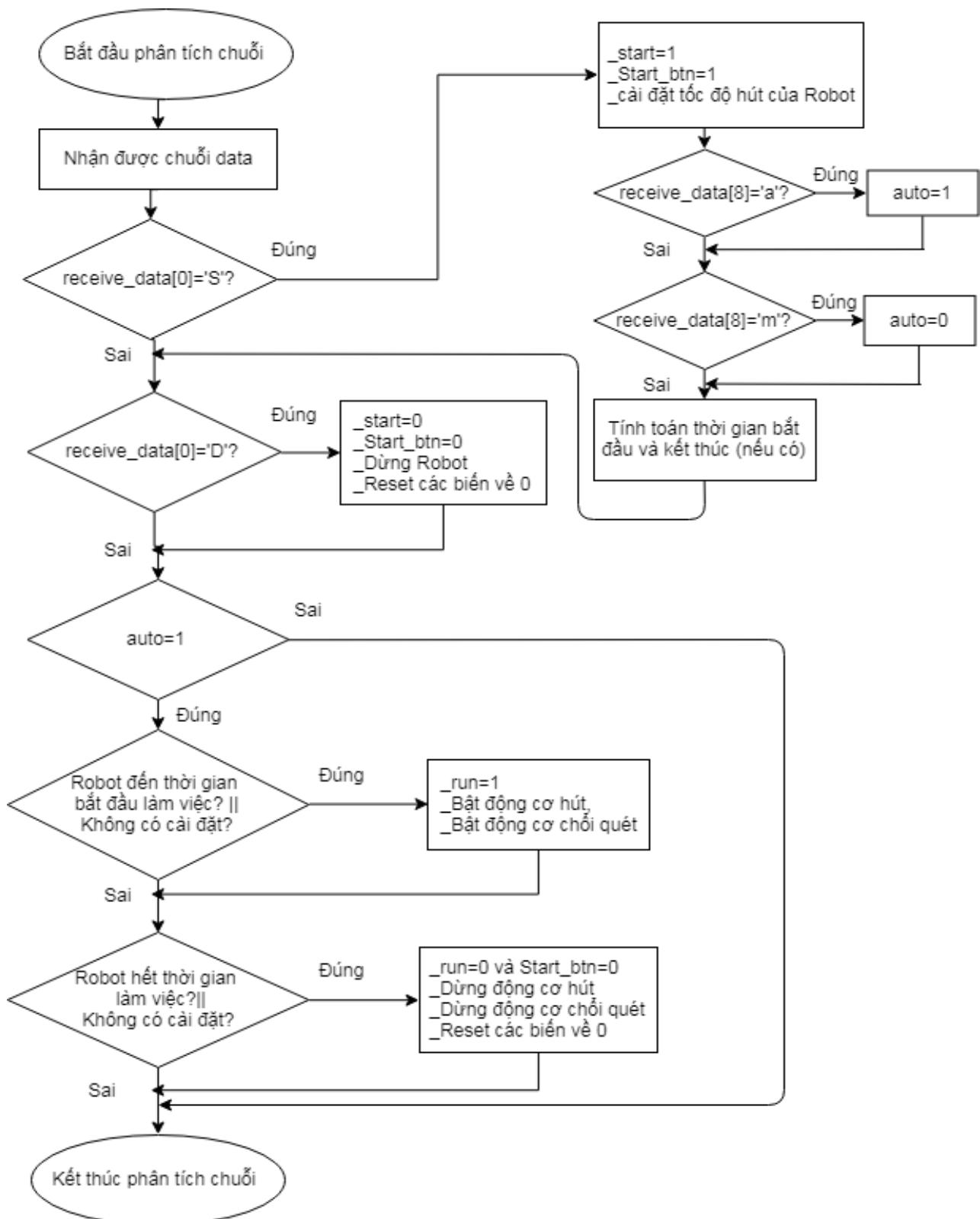
➤ **Nhận xét:** Mặc dù chưa thực sự hoàn thiện nhưng thuật toán di chuyển tương đối hiệu quả và ổn định trong đề tài này, diện tích khu vực được vệ sinh rộng hơn so với di chuyển bất kì, đồng thời còn hạn chế được sự trùng lặp.

5.9 Giải thuật hệ thống

Sau khi đã lắp ráp các module và lập trình sử dụng các giải thuật nêu trên thành các hàm con riêng lẻ, nhóm tiến hành tổng hợp tất cả để cho ra đời giải thuật của toàn bộ hệ thống một cách hoàn chỉnh, dưới đây là lưu đồ giải thuật mô tả hoạt động của chương trình chính bên trong vi xử lí STM32F407 và một số hàm con liên quan.



Hình 5.25. Lưu đồ giải thuật chương trình chính của vi điều khiển



Hình 5.26. Lưu đồ giải thuật phân tích chuỗi dữ liệu

CHƯƠNG 6: THIẾT KẾ GIAO DIỆN NGƯỜI DÙNG BẰNG NGÔN NGỮ C#

Nội dung chính

- 6.1 Đặt vấn đề**
- 6.2 Giới thiệu giao diện người dùng**
- 6.3 Giới thiệu chương trình Visual Studio**
- 6.4 Các tính năng chính của giao diện**

6.1 Đặt vấn đề

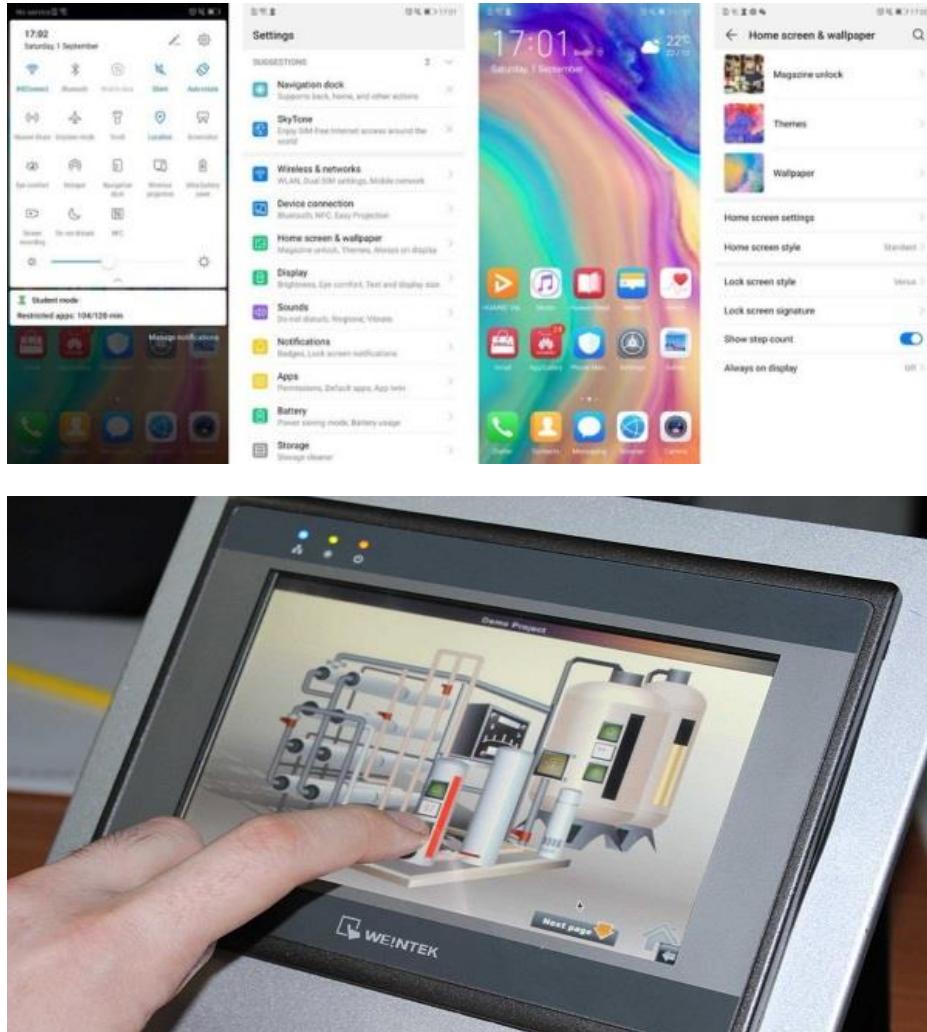
Đối với những dòng sản phẩm máy hút bụi quen thuộc trước đây, đa phần người dùng thường sử dụng tính năng điều khiển bằng remote để dễ dàng cho việc điều chỉnh và thiết lập chế độ cho robot. Nhưng hiện nay, với cuộc chạy đua về phát triển khoa học ứng dụng, đã có nhiều hãng nổi tiếng cho ra đời dòng sản phẩm có thể điều khiển từ xa bằng các ứng dụng trên điện thoại qua kết nối wifi. Việc tiếp thu công nghệ hiện đại chế tạo dòng máy hút bụi lập trình qua App, ứng dụng trên điện thoại một cách thông minh là bước cải tiến vượt bậc, giúp người có thể linh động hơn trong việc điều khiển cũng như theo dõi sản phẩm trong quá trình làm việc một cách thuận tiện, phù hợp cho thời đại smartphone ngày nay.

Trong phạm vi đề tài này, vì phần hiểu biết của các thành viên trong nhóm về lập trình ứng dụng trên Android còn ít và thời gian hạn chế nên chúng em chưa thể lập trình một ứng dụng trên điện thoại điều khiển sản phẩm bằng kết nối wifi giống với thực tế. Nhưng với những kiến thức sẵn có về chương trình Visual Studio và ngôn ngữ C#, nhóm chúng em quyết định sẽ thiết kế một Window Form Application để mô phỏng gần giống với phần mềm giao diện trên điện thoại, có thể điều khiển robot qua đường truyền không dây bằng cách sử dụng module wifi không dây ESP8266.

6.2 Giới thiệu giao diện người dùng (UI)

Giao diện người dùng hay còn gọi là UI, là giao diện của người dùng và phần mềm phía sau nó, có thể coi là những gì chúng ta nhìn thấy trên màn hình và tương tác với các thiết bị công nghệ thông qua những câu lệnh được mã hóa. Ví dụ như trong một chiếc smartphone, màn hình người dùng của các ứng dụng như Google Maps hay là Facebook, thông qua đó người dùng có thể đưa thông tin và nhận phản hồi, đó chính là giao diện người dùng. Hoặc cũng có thể là những giao diện HMI quen thuộc trong các nhà máy với hệ thống tự động hóa, hiển thị thông tin, trạng thái vận hành của một quá trình công nghiệp, đồng thời chấp nhận và thực hiện các lệnh điều khiển và cài đặt giúp các vận hành viên có thể dễ dàng theo dõi, thiết lập thông số yêu cầu,...đạt hiệu

quá cao trong quá trình quản lý ..v.v.



Hình 6.1. Ví dụ điển hình về UI

UI ban đầu được thiết kế để làm việc với chuột và bàn phím, nhưng với sự ra đời của màn hình cảm ứng, mọi thứ từ điện thoại thông minh đến máy ATM và thậm chí cả tủ lạnh thông minh giờ đây đều có một số hình thức UI. UI đã phát triển đáng kể qua nhiều năm và tiếp tục với sự tăng nhận dạng giọng nói và công nghệ xử lý ngôn ngữ tự nhiên, điều khiển nhiều thiết bị thông qua trợ lý thoại.

Lợi ích chính của UI là trình bày thông tin cho người dùng các thiết bị công nghệ theo một cách dễ hiểu và tương tác thông qua thực hiện hành động bằng cách nhấp, nhấn hoặc điều hướng đến chỉ báo và biểu tượng trực quan giúp mọi người ở mọi

cấp độ kiến thức, từ người mới bắt đầu đến người dùng thành thạo, nhà phát triển nâng cao hoặc các cá nhân hiểu biết về công nghệ khác đều có thể học và sử dụng chúng trong khoảng thời gian ngắn.

Ngày nay, UI được sử dụng cho phần lớn các hệ điều hành máy tính, hệ điều hành di động và các phần mềm tồn tại khác và đặc biệt, trong thời đại Smartphone cùng với kỉ nguyên 4.0, nơi mà thiết bị di động có thể thực hiện hầu hết mọi thứ từ giải trí đến công việc, giao tiếp thì việc xây dựng và ứng dụng giao diện người dùng với 3 tiêu chí đẹp, đơn giản, dễ sử dụng vào các sản phẩm công nghệ đã trở thành xu hướng mới, mang lại nhiều bước cải tiến đáng kể trong sự nghiệp phát triển khoa học kĩ thuật.

Chính vì vậy, để có thể bắt kịp xu thế hiện nay, ứng dụng được giao diện người dùng vào quá trình sử dụng cũng như theo dõi trạng thái của robot, nhóm chúng em sẽ thiết kế một giao diện đơn giản để có thể cài đặt và thiết lập những thông số cho chiếc máy hút bụi thông qua kết nối không dây.

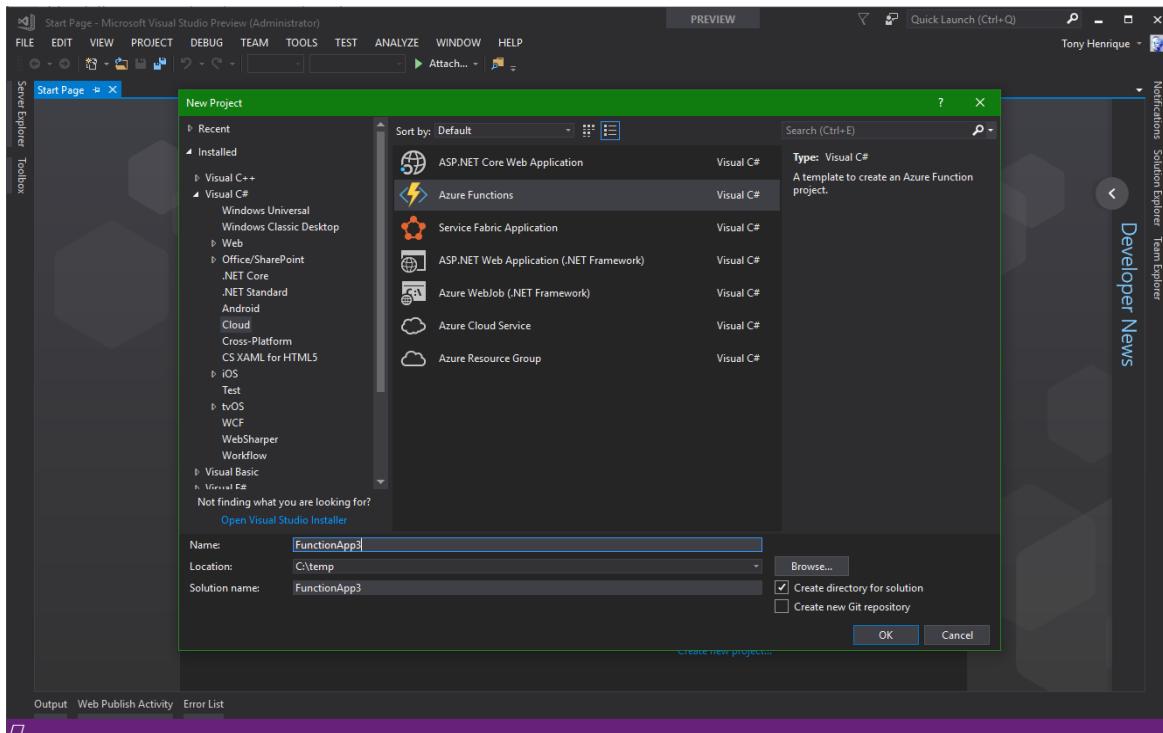
6.3 Giới thiệu chương trình Visual Studio



Hình 6.2. Chương trình Visual Studio 2017

Visual studio là một trong những công cụ hỗ trợ lập trình rất nổi tiếng nhất hiện nay của Microsoft và chưa có một phần mềm nào có thể thay thế được nó. Visual Studio được viết bằng 2 ngôn ngữ chính đó là C# và VB+. Đây là 2 ngôn ngữ lập trình giúp người dùng có thể lập trình được hệ thống một cách dễ dàng và nhanh chóng nhất thông qua Visual Studio. Ngoài ra, Visual Studio còn hỗ trợ lập trình trên nhiều ngôn ngữ khác như: C/C++, F#, HTML, CSS, JavaScript.

Từ khi ra đời đến nay, Visual Studio đã có rất nhiều các phiên bản sử dụng khác nhau. Điều đó giúp cho người dùng có thể lựa chọn được phiên bản tương thích với dòng máy của mình cũng như cấu hình sử dụng phù hợp nhất. Bên cạnh đó, Visual Studio còn cho phép người dùng có thể tự chọn lựa giao diện chính cho máy của mình tùy thuộc vào nhu cầu sử dụng.



Hình 6.3. Giao diện của chương trình

Một số tính năng của phần mềm Visual Studio:

- Biên tập mã:

Giống như bất kỳ một IDE khác, Visual Studio gồm có một trình soạn thảo mã hỗ trợ tô sáng cú pháp và hoàn thiện mã bằng cách sử dụng IntelliSense không chỉ cho các hàm, biến và các phương pháp mà còn sử dụng cho các cấu trúc ngôn ngữ như: truy vấn hoặc vòng điều khiển. Bên cạnh đó, các trình biên tập mã Visual Studio cũng hỗ trợ cài đặt dấu trang trong mã để có thể điều hướng một cách nhanh chóng và dễ dàng. Hỗ trợ các điều hướng như: Thu hẹp các khối mã lệnh, tìm kiếm gia tăng,...

Visual Studio còn có tính năng biên dịch nền tảng là khi mã đang được viết thì phần mềm này sẽ biên dịch nó trong nền để nhằm cung cấp thông tin phản hồi về cú pháp cũng như biên dịch lỗi và được đánh dấu bằng các gạch gợn sóng màu đỏ

➤ Trình gỡ lỗi:

Visual Studio có một trình gỡ lỗi có tính năng vừa lập trình gỡ lỗi cấp máy và gỡ lỗi cấp mã nguồn. Tính năng này hoạt động với cả hai mã quản lý giống như ngôn ngữ máy và có thể sử dụng để gỡ lỗi các ứng dụng được viết bằng các ngôn ngữ được hỗ trợ bởi Visual Studio.

➤ Thiết kế:

- Windows Forms Designer

Được sử dụng với mục đích xây dựng GUI sử dụng Windows Forms, được bố trí dùng để xây dựng các nút điều khiển bên trong hoặc cũng có thể khóa chúng vào bên cạnh mẫu. Điều khiển trình bày dữ liệu có thể được liên kết với các nguồn dữ liệu như: cơ sở dữ liệu hoặc truy vấn.

- WPF Designer

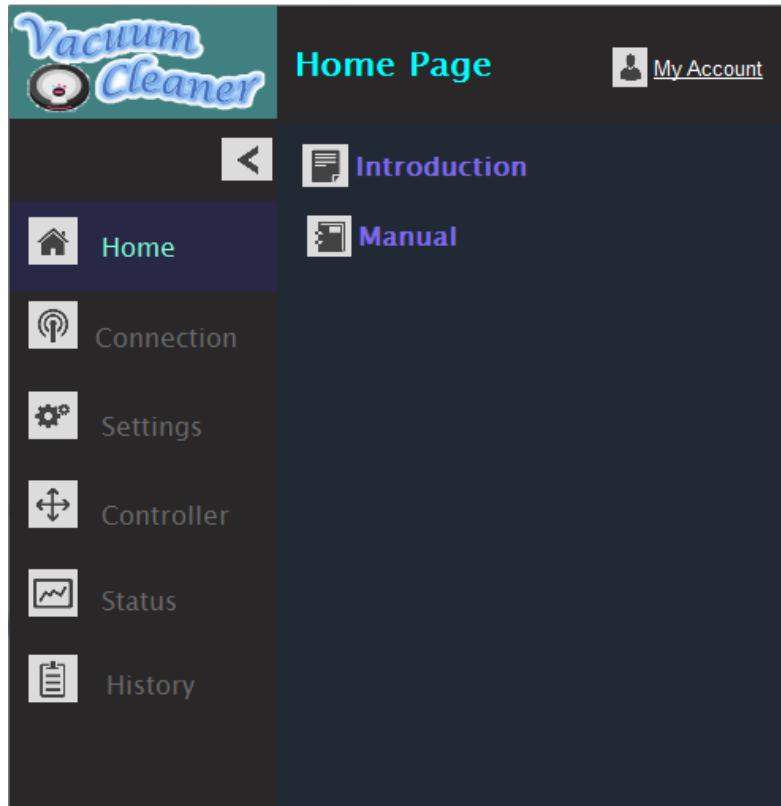
Tính năng này cũng giống như Windows Forms Designer có công dụng hỗ trợ kéo và thả ẩn dụ. Sử dụng tương tác giữa người và máy tính nhằm mục tiêu vào Windows Presentation Foundation.

- Web designer/development

Visual Studio cũng có một trình soạn thảo và thiết kế website cho phép các trang web được thiết kế theo tính năng kéo và thả đối tượng

⇒ Với những tính năng đặc biệt, dễ sử dụng và quen thuộc trong thiết kế GUI, nhóm đã quyết định dùng chương trình Visual Studio để mô phỏng một ứng dụng trên android nhằm giúp mọi người có thể hình dung cách vận hành cũng như lập trình cho robot trong quá trình sử dụng qua kết nối không dây.

6.4 Các tính năng chính của giao diện



Hình 6.4. Giao diện chính của ứng dụng

Trong phạm vi đề tài, nhóm chúng em sẽ tập trung thiết kế giao diện người dùng với những tính năng cơ bản như sau: Có thể kết nối sản phẩm qua đường truyền không dây, theo dõi trạng thái pin cũng như hiển thị thời gian thực của robot hút bụi. Bên cạnh đó, cách vận hành cho robot còn được thiết lập với 2 chế độ tự động và bằng tay tùy vào mong muốn của người sử dụng giúp chúng ta chủ động hơn trong việc điều khiển. Ngoài ra nhóm còn bổ sung thêm tính năng ghi chép quá trình làm việc của sản phẩm và cài đặt thời gian hoạt động hay thời gian bắt đầu công việc cho chiếc máy. Tất cả những tính năng trên sẽ hạn chế tối đa các thao tác của con người, giúp cho quá trình sử dụng trở nên linh hoạt và tiện lợi hơn. Phần tiếp theo của báo cáo sẽ lần lượt giới thiệu chi tiết các tính năng có trong ứng dụng và mỗi tính năng sẽ ứng với một nhãn trên UI.

6.4.1 Kết nối thiết bị

Đầu tiên, để có thể thực hiện các tác vụ cài đặt và điều khiển, ta phải kết nối được tín hiệu với robot. Như đã giới thiệu các ở các phần trên, nhóm chúng em sẽ sử dụng phương thức truyền UART không dây giữa robot và PC bằng cách sử dụng 2 ESP8266. Một ESP đóng vai trò là máy chủ, sau khi nhận được chuỗi dữ liệu từ chiếc máy hút bụi sẽ truyền tất cả sang PC thông qua cổng COM. Chính vì vậy, nhiệm vụ chính của phần này sẽ là dùng ngôn ngữ C# tạo kết nối giữa ứng dụng trên máy tính với ESP giao tiếp bên ngoài thông qua cổng COM đã được chọn, gồm các bước chính sau:

- ❖ *Khai báo thư viện chức năng:*

```
using System.IO.Ports;
```

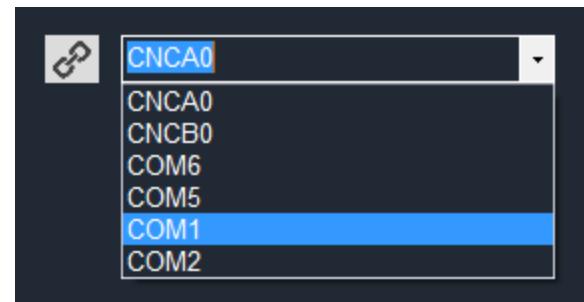
Đây là thư viện có sẵn trong Visual Studio được dùng cho các chương trình cần thiết lập sử dụng đến cổng COM.

- ❖ *Lấy tất cả COM có trong máy tính và cài đặt thông số:*

Khai báo lệnh `SerialPort` `UART = new SerialPort();` trong lớp `public partial class Form1 : Form`. Khi ta cần lấy danh sách các cổng COM hiện có trên máy tính, thông thường ta sử dụng phương thức:

```
string[] ports = SerialPort.GetPortNames();
cbxCom.Items.AddRange(ports);
```

Kết quả thì hệ thống sẽ trả về danh sách các cổng dạng COM1, COM2,... hiện tại đang có trên hệ thống vào mảng `ports[]`. Giả sử danh sách cập nhật cổng COM được đưa vào ComboBox.



Để xác định được thiết bị nào ứng với cổng COM nào thì ta phải xem lại trong Device Manager -> Port (COM & LPT) để kết nối được chính xác.

Sau khi đã có được danh sách, ta thiết lập thông số cho phương thức truyền

```
UART.ReadTimeout = 2000;  
UART.BaudRate = 115200; //cài đặt tốc độ Baud  
UART.Parity = Parity.None; //cài đặt cờ Parity  
UART.StopBits = StopBits.One; //cài đặt Stopbit
```

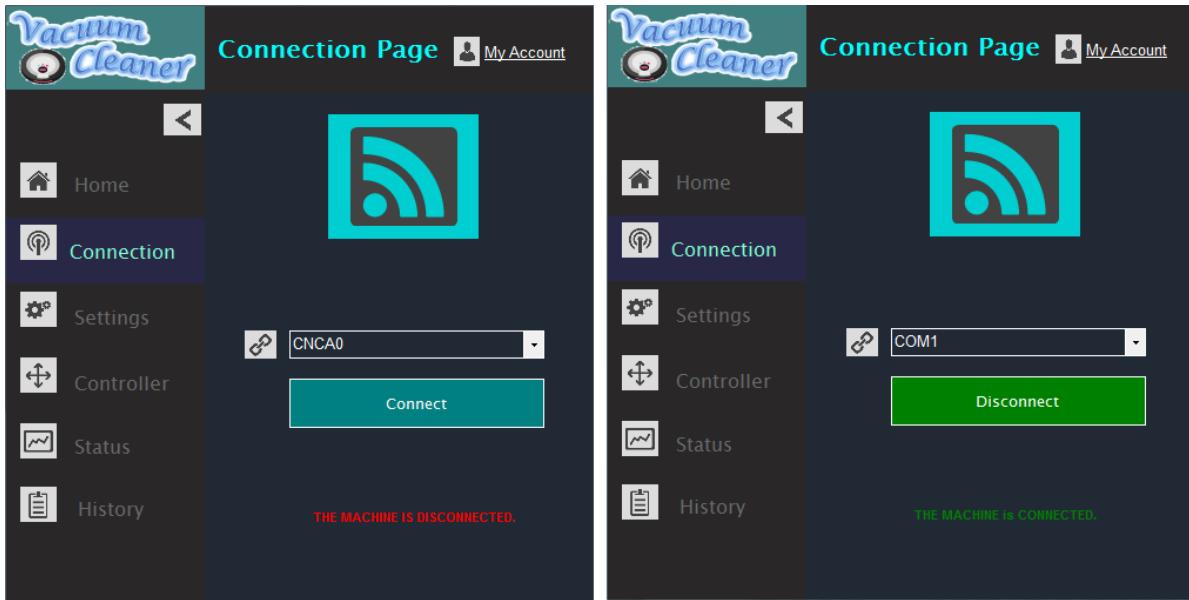
Lưu ý: Thông số được cài đặt trong C# phải tương thích với những giá trị đã được thiết lập trong ESP8266 khi lập trình, có như vậy, chuỗi dữ liệu mới được truyền nhận đúng từ PC sang robot và ngược lại.

❖ Kết nối và hủy kết nối cổng COM:

Sau khi đã chọn được cổng COM kết nối với module truyền nhận ESP từ ComboBox, ta chỉ cần kết nối hoặc ngắt kết nối với COM bằng những câu lệnh đơn giản mà thư viện đã hỗ trợ sẵn.

```
UART.PortName = cbxCom.Text; //UART là một lớp Serial Port  
(cổng COM) đã được khai báo ở trên.  
UART.Open(); //mở cổng COM  
Hoặc UART.Close(); //đóng cổng COM
```

Khi UI chưa được kết nối với cổng COM, trên giao diện kết nối (nằm trong nhãn **Connection** trên UI) sẽ hiện thị nhãn dán với dòng chữ đỏ báo PC chưa được kết nối, và sẽ là dòng chữ xanh khi đã kết nối thành công, sẵn sàng cho việc truyền nhận dữ liệu.



Hình 6.5. Giao diện kết nối của ứng dụng

❖ Nhận và gửi chuỗi dữ liệu:

Để lấy được chuỗi kí tự ta lần lượt thêm câu lệnh và hàm con nhận dữ liệu.

```
this.UART.DataReceived += new  
System.IO.Ports.SerialDataReceivedEventHandler(  
this.UART_DataReceived);
```

Câu lệnh trên giúp ta khai báo một hàm con mà ở đó ta có thể nhận chuỗi dữ liệu theo cách mong muốn, đồng thời cũng là nơi có thể xử lý chuỗi để lấy được những thông tin cần thiết hiển thị trên UI.

```
private void UART_DataReceived(object sender,  
SerialDataReceivedEventArgs e)  
{ try  
    { Tam = UART.ReadTo("\r"); //nhận dữ liệu từ  
    cổng COM cho đến khi gặp kí tự xuống hàng.  
    catch (Exception)  
    {}  
}
```

Đối với việc gửi dữ liệu qua cổng COM, thư viện trong Visual Studio cũng có

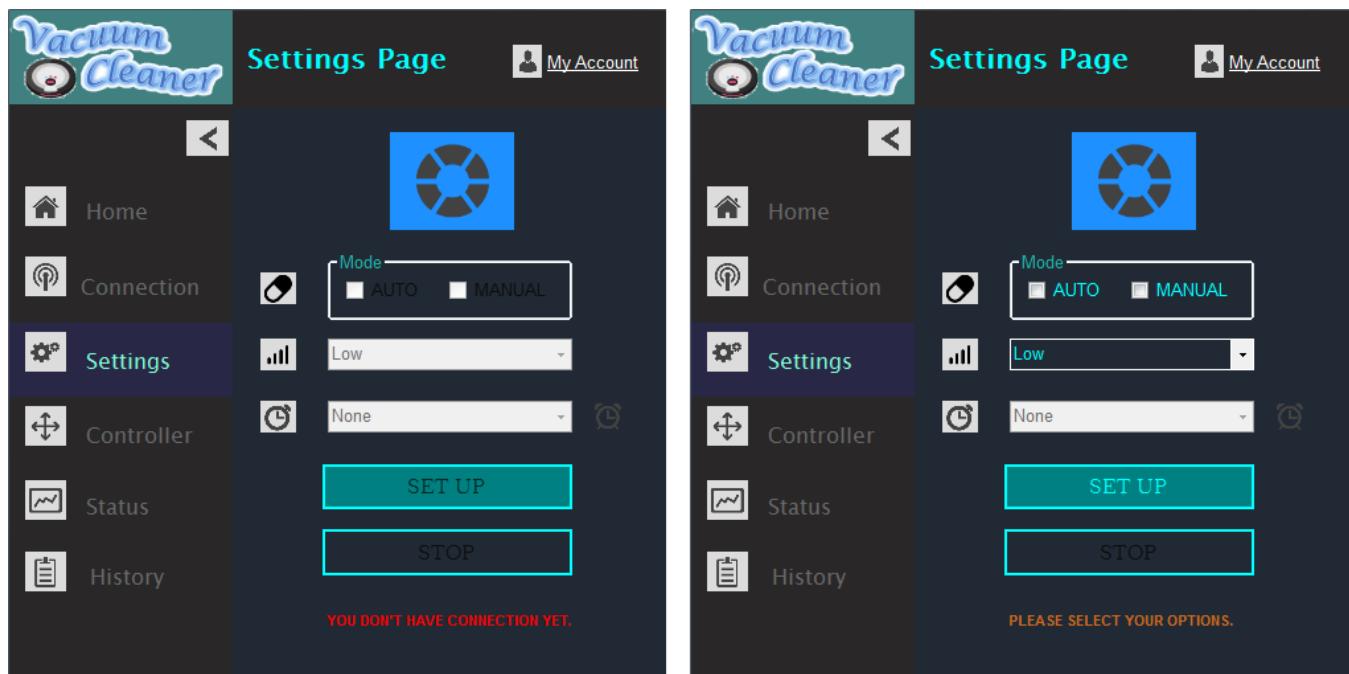
hỗ trợ với cách thực hiện đơn giản hơn bằng câu lệnh:

```
UART.WriteLine(data); // với data là chuỗi kí tự cần truyền
```

⇒ Tất cả những bước trên chính là những thiết lập, lập trình cơ bản trong C# khi sử dụng Serial Port (cổng COM) để giao tiếp ngoại vi với các module bên ngoài.

6.4.2 Cài đặt cho robot

Khi đã giao tiếp thành công với vi điều khiển, có thể truyền nhận chuỗi dữ liệu một cách chính xác, ta bước vào phần cài đặt cho robot hút bụi. Ở mục này, phần báo cáo sẽ tập trung trình bày nội dung của chuỗi truyền cũng như những tác vụ cơ bản để người dùng có thể sử dụng giao diện thiết lập thông số cho sản phẩm. Phần giao diện cài đặt chính sẽ nằm trong nhãn **Settings** trên UI với bố cục như sau:



Hình 6.6. Giao diện cài đặt cho robot

Tương tự trên giao diện **Connection**, khi thiết bị chưa giao tiếp với PC, ta sẽ nhận được nhãn dán với dòng chữ đỏ báo chưa được kết nối và kèm theo là các vùng cài đặt sẽ bị vô hiệu hóa không thể nhập cũng như tác động trực tiếp.Người dùng chỉ

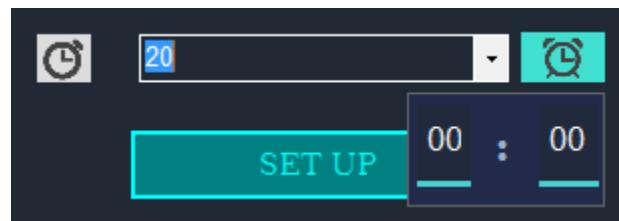
có thể bắt đầu cài đặt các thông số và chế độ cho chiếc máy khi có dòng chữ cam với trạng thái chờ, cho phép thi hành tác vụ nhập và chọn trên giao diện, tức là thời điểm PC đã kết nối thành công với robot.

Trên giao diện **Settings** của ứng dụng, người dùng sẽ có thể cài đặt cách vận hành cho robot với những thông số như sau:

- *Chế độ (mode)*: robot sẽ được hoạt động theo 2 chế độ tự động (Auto) và bằng tay (Manual) ứng với mỗi CheckBox trên UI tùy vào mong muốn và mục đích của người sử dụng cũng như không gian làm việc của robot.
- *Tốc độ của máy hút*: Có 3 mức từ thấp đến cao tương ứng **Low**, **Medium**, **High**. Thông số này sẽ tùy thuộc vào loại bụi bẩn, rác mà người dùng muốn dọn dẹp. Ví dụ đối với loại bụi có trọng lượng nhỏ nhẹ, khoảng không gian tương đối ít bụi thì người dùng chỉ cần chọn tốc độ hút **Low** hay **Medium** để có thể tiết kiệm năng lượng cho sản phẩm,..



- *Thời gian (time)*: Chỉ được sử dụng trong chế độ tự động với mục đích thiết lập thời gian làm việc cho robot. Ngoài ra, trong phần cài đặt này, nhóm có thực hiện bước cài tiến nhỏ bằng cách cho phép người dùng có thể thiết lập cả thời gian bắt đầu làm việc cho robot, thuận tiện cho những lúc bận rộn không để *Settings* trực tiếp cho sản phẩm.



Sau khi đã hoàn thành bước chọn thông số cho robot, người dùng sẽ nhấn nút **Set Up** trên UI để gửi chuỗi dữ liệu xuống robot qua kết nối không dây.

```
data = "S" + set_point + Time + startHour +
startMin + Mode;
UART.WriteLine(data); //data là chuỗi dữ liệu cần truyền
```

Với mỗi thông số được thiết lập, nhóm sẽ lập trình gán một chuỗi kí tự vào từng biến trong chuỗi **data**. Cụ thể như sau:

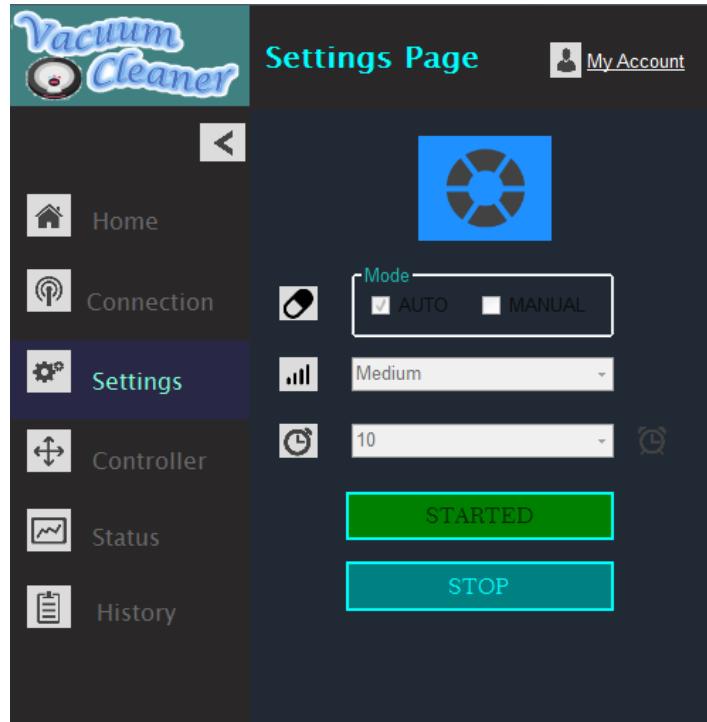
- “S”: kí tự “S” đầu tiên của chuỗi để kí hiệu rằng đây là chuỗi dữ liệu *Settings* giúp vi điều khiển có thể nhận biết và phân biệt với những dữ liệu khác được.
- *Set_point* : sẽ ứng với tốc độ máy hút với các kí tự “L”, ”M”, ”H” tương ứng với 3 mức độ Low, Medium, High.
- *Time* : thời gian hoạt động của robot, được format về 2 kí tự bằng hàm

```
Time = string.Format("{0:00}", cbxTime.Text);
```
- *startHour & StartMin*: Thời gian hoạt động của robot, cũng được format về 2 kí tự.
- *Mode*: sẽ ứng với 2 kí tự “a”, ”m” với 2 chế độ tự động và bằng tay.

Ví dụ, khi nhấn nút **Set Up** và chế độ Auto được chọn thì biến *Mode* sẽ được gán kí tự ‘a’, hoặc sẽ là kí tự ‘m’ khi người dùng chọn chế độ Manual. Tương tự cho biến tốc độ máy hút *Set_point*.

Riêng với *Time*, *startHour & StartMin* sẽ được mặc định với tham số “00” ban đầu hoặc khi người dùng không cài đặt gì ở thông số này. Việc lập trình như vậy để đảm bảo rằng độ dài của chuỗi sẽ là 9 kí tự không đổi trong quá trình truyền nhận, giúp việc phân tích và quản lý chuỗi trong vi điều khiển trở nên đồng bộ, dễ dàng hơn.

Khi PC gửi thành công chuỗi dữ liệu qua Serial Port , nút Set Up sẽ chuyển qua màu xanh lá với chữ **Started**. Giao diện cài đặt của ứng dụng sẽ bị vô hiệu hóa, người dùng không còn thực hiện được những tác vụ thiết lập thông số cho đến khi nhấn **Stop** và trở về với trạng thái ban đầu chuẩn bị cho lần *Settings* tiếp theo.



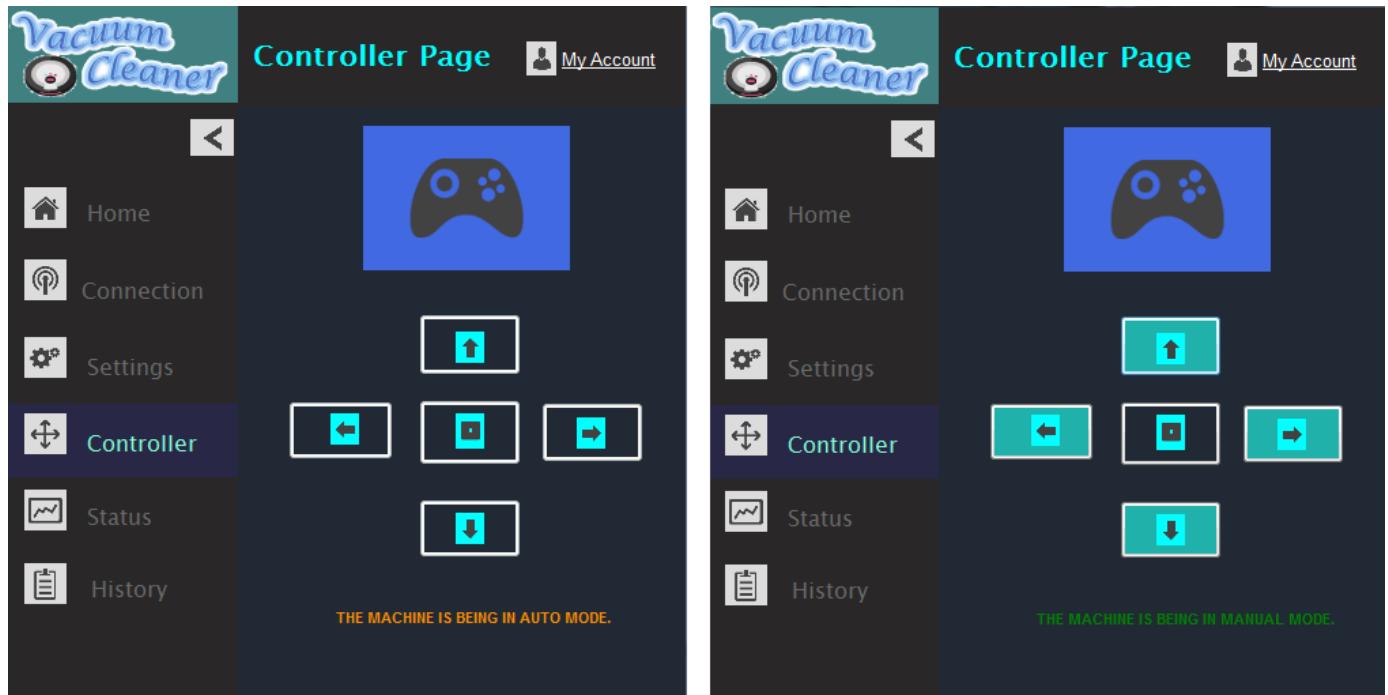
Hình 6.7. Giao diện cài đặt sau khi nhấn SET UP

Ngoài ra ở chế độ điều khiển bằng tay, người dùng có thể điều chỉnh hướng đi cho robot di chuyển đến khu vực mong muốn và dọn dẹp một cách nhanh chóng mà không cần phải chờ robot hoạt động theo chu trình. Giao diện điều khiển bằng tay sẽ được đặt trong nhãn **Controller** trên UI với cách bố trí gồm 4 nút nhấn là trái, phải, tiến, lùi.

Khi một nút nào đó trong bộ điều khiển được nhấn giữ, tương ứng với hướng điều khiển mong muốn của người sử dụng, nó sẽ chuyển sang màu xanh lá tất cả 4 nút sẽ bị vô hiệu hóa. Người dùng chỉ có thể điều chỉnh



hướng khác cho đến khi nhả nút nhấn.



Hình 6.8. Giao diện bộ điều khiển ở 2 chế độ Auto & Manual

Tương ứng với mỗi lần một nút trong bộ điều khiển được nhấn, chương trình sẽ gửi một chuỗi 9 bit với kí tự đầu lần lượt là “t”, “l”, “p”, “h” với thứ tự các nút trái, lùi, phải, tiến, 8 kí tự còn lại sẽ là chữ số “0”. Như đã đề cập ở phần trên, việc truyền 9 kí tự giúp cho việc đồng bộ tất cả các chuỗi truyền, khiến việc phân tích trở nên dễ dàng hơn khi lập trình vi điều khiển STM32f4xx.

⇒ Đó là tất cả những phương thức truyền và nội dung của chuỗi dữ liệu mà nhóm muốn gửi xuống robot hút bụi khi lập trình cho phần cài đặt của ứng dụng.

6.4.3 Theo dõi trạng thái của robot

Ngày nay ngoài việc có thể cài đặt chế độ làm việc cho máy, người dùng còn có nhu cầu cao hơn, đó chính là theo dõi và giám sát trạng thái hoạt động của robot. Đối với những loại máy thông thường, chúng ta không thể xác định được giá trị pin hiện tại của sản phẩm. Việc không thể biết trước máy sẽ dừng hoạt động thời điểm nào có thể dẫn đến một số điều không thuận tiện khi sử dụng như pin không còn vào những lúc

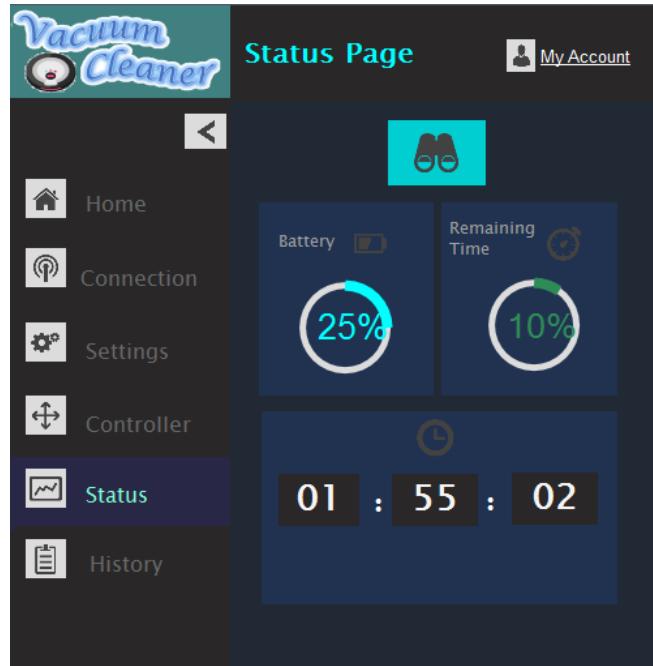
cần thiết. Chính vì vậy, nhóm quyết định sẽ thiết kế một giao diện hiển thị trạng thái của robot trên ứng dụng để người dùng có thể theo dõi giá trị pin, thời gian thực của chiếc máy và thời gian hoạt động còn lại của máy khi ở chế độ tự động.

Những thông số trên sẽ được vi điều khiển nhận từ các Module ngoại vi, sau đó xử lý rồi tạo thành một chuỗi dữ liệu truyền lên PC. Nhiệm vụ chính của quá trình xây dựng nên tính năng này của ứng dụng là tìm cách thực hiện phân tích chuỗi kí tự nhận được, biến đổi và hiển thị lên giao diện trạng thái trong nhãn Status trên UI .

Chuỗi dữ liệu mà PC nhận được sẽ bao gồm 18 kí tự gồm các giá trị thời gian ngày, tháng, năm, phút, giây, giá trị nguồn còn lại theo phần trăm, cờ báo khi robot hoạt động xong theo những tham số đã cài đặt trước. Chuỗi nhận được sẽ được lưu vào biến tạm, sau đó dùng hàm Tam.Substring(1, 3); phân tích thành các chuỗi kí tự nhỏ hơn rồi chuyển sang kiểu dữ liệu số đúng với với các giá trị mong muốn được gửi lên từ vi điều khiển. Việc còn lại là tính toán những tham số, hiển thị lên giao diện để có thể quan sát một cách trực quan và thuận tiện.

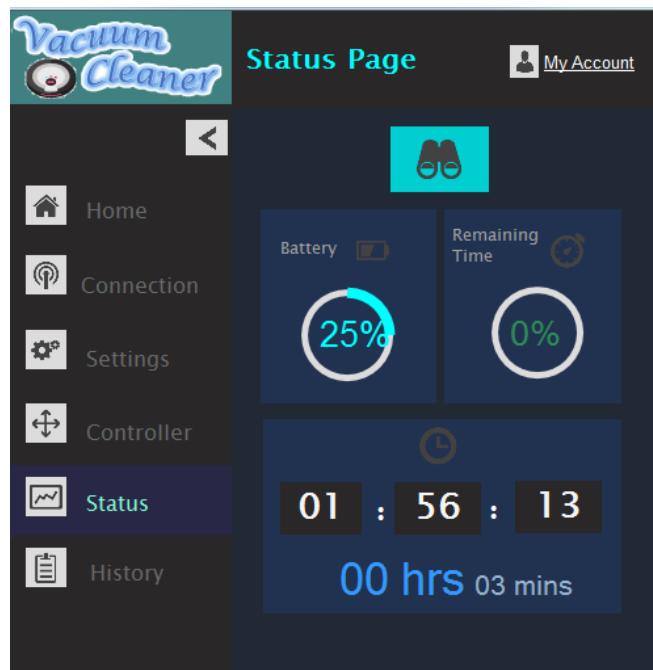
Sau đây là một số kết quả hiển thị của giao diện **Status** ứng với một số trường hợp cài đặt của người dùng:

- ❖ Trường hợp 1: Người sử dụng muốn robot hoạt động trong khoảng thời gian ấn định chọn trước, ngoài giá trị phần trăm pin, thời gian thực, giao diện sẽ hiển thị phần trăm thời gian làm việc còn lại của sản phẩm trong mục **Remaining time**.



Hình 6.9. Giao diện trạng thái trong TH1

- ❖ Trường hợp 2: Người sử dụng muốn cài đặt thời gian bắt đầu làm việc cho robot. Giao diện hiển thị thời gian còn lại để robot bắt đầu làm việc trong phần **Clock** (Có biểu tượng đồng hồ xem giờ)



Hình 6.10. Giao diện trạng thái trong TH2

Lưu ý: phần remaining time sẽ hiển thị giá trị 0 khi robot hoạt ở chế độ Manual hoặc người sử dụng không cài đặt thời gian làm việc cho robot.

⇒ Nhìn chung, nhóm đã xây dựng được phần giao diện có thể hiển thị được trạng thái của pin và một số giá trị liên quan đến thời gian cài đặt.

6.4.4 Ghi chép quá trình làm việc

Tính năng cuối cùng của ứng dụng là một tính năng đặc biệt ít xuất hiện trong phần lớn các sản phẩm máy hút bụi trên thị trường hiện nay. Đó chính là ghi chép, hiển thị lại một số quá trình làm việc gần nhất của robot, giúp người dùng có thể nắm bắt và biết được những lần sử dụng trước đây của mình. Đây là một tính năng không mấy quan trọng trong quá trình làm việc của sản phẩm, nhưng lại có ý nghĩa rất lớn trong việc đánh giá khả năng hoạt động hiệu quả của robot giúp người dùng có thể theo dõi được lượng pin tiêu hao cũng như thời gian bắt đầu và kết thúc có đúng với những thông số đã cài đặt hay không. Từ đó sẽ nhận biết được chất lượng của sản phẩm còn tốt hay xấu đi, có cần thiết bảo hành sửa chữa.,v.v.

Trong phần xây dựng nên tính năng này, nhóm sẽ tập trung tìm cách lưu lại các giá trị cần thiết như ngày, tháng, thời gian bắt đầu và kết thúc quá trình của chiếc máy, phần trăm giá trị pin, chế độ hoạt động đã được cài đặt và vận hành trước đó. Vì chỉ để mô phỏng, nên nhóm quyết định sẽ chỉ thực hiện ghi chép 4 lần sử dụng gần nhất trước đây.

Có một số cách để có thể lưu trữ những thông số trên như liên kết với cơ sở dữ liệu (hay còn gọi là database) trong chương trình Visual Studio hay tạo ra tập tin trong thư mục để ghi chép lại chúng...v.v. Nhóm đã sử dụng phương thức tạo ra tập tin để lưu trữ bằng một số câu lệnh đơn giản:

```
StreamWriter write1 = new StreamWriter("database", true);
// Khai báo lớp StreamWriter và tạo tệp văn bản để lưu thông số
StreamReader read1 = new StreamReader("database");
// Khai báo lớp StreamReader để đọc dữ liệu từ tệp văn bản
```

Lớp **StreamWriter** và **StreamReader** được thiết kế để thao tác dễ dàng trong việc xuất nhập các tập tin loại văn bản, cả 2 sẽ làm việc với kí tự Unicode. Để sử dụng được lớp **Stream** trước tiên phải khai báo thư viện:

```
using System.IO.Ports;
```

Đây là thư viện sẵn có trong C# cho phép người lập trình có thể đọc và ghi dữ liệu dạng text vào hoặc từ một tệp văn bản (có dạng .txt). Khi robot kết thúc quá trình làm việc hoặc được dừng hoạt động dưới sự điều khiển của người dùng, ứng dụng sẽ lập tức nhảy đến câu lệnh đã nêu trên vào tạo ra một tập tin .txt trong thư mục **bin** của thư mục project rồi tiếp tục ghi dữ liệu bằng dòng code đơn giản với những thông số cần thiết.

```
write1.WriteLine(day + " - " + month + " Start " + startHour +  
": " + startMin + " Stop " + hour + ":" + minute +  
battery + Mode); // lưu chuỗi text rồi xuống dòng
```

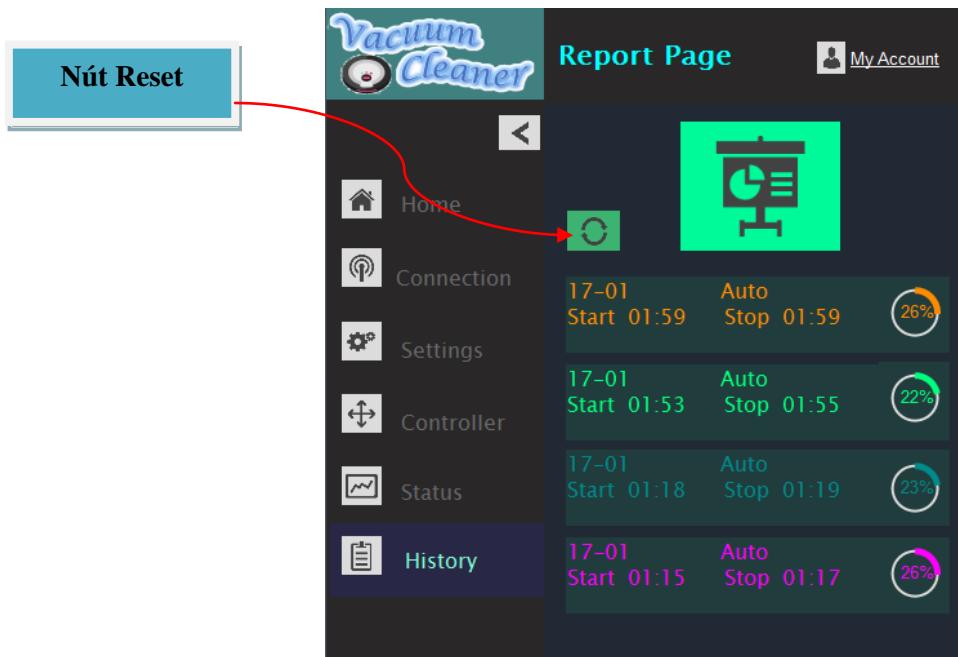
Lưu ý: Sau mỗi lần thực hiện lưu dữ liệu, chúng ta phải dùng dòng lệnh **write.Close();** đóng các đối tượng writer hiện tại và giải phóng mọi nguồn lực chiếm dụng đang liên kết với Writer. Có như vậy, ta mới có thể sử dụng lớp Reader để đọc dữ liệu từ tập tin văn bản ở phần sau. Tương tự với khi sử dụng **StreamReader**.

Vì giao diện sẽ lưu chuỗi kí tự tiếp tục vào tệp văn bản theo từng dòng nên để có thể trích xuất được 4 chuỗi dữ liệu tương thích với 4 lần hoạt động gần nhất của robot, ta cần phân tích, dùng một số giải thuật lập trình cùng với câu lệnh:

```
read.ReadLine(); // đọc chuỗi text đến khi gặp kí tự xuống dòng
```

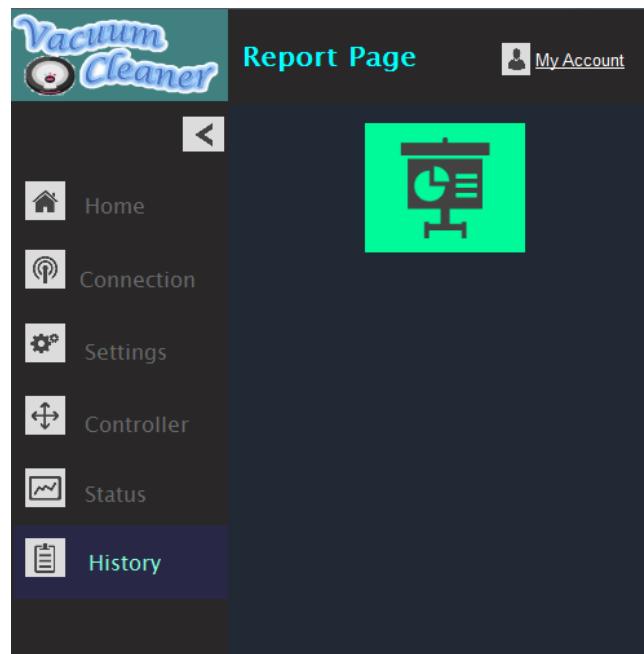
Những thông số về quá trình làm việc sẽ được cập nhật trên giao diện **History** sau mỗi lần người dùng nhấn **Set Up** khi đã cài đặt lại chế độ làm việc cho chiếc máy.

Phần giao diện sau khi lập trình sẽ có kết quả hiển thị như sau:



Hình 6.11. Giao diện hiển thị quá trình làm việc

Ngoài ra, nhóm còn khởi tạo một nút **Reset** nằm bên góc trái của giao diện nhằm giúp người sử dụng có thể xóa hết những số liệu hiển thị trên UI và trở về trạng thái ban đầu, sẵn sàng cho những lần ghi chép tiếp theo.



Hình 6. 12. Giao diện khi nhấn Reset

➤ Kết luận: đó là toàn bộ những tính năng hiện có trên ứng dụng mà nhóm đã triển khai xây dựng. Mặc dù còn có những khiếm khuyết, vẫn chưa bắt kịp được những cải tiến của một số loại sản phẩm trên thị trường hiện nay nhưng nhìn chung mô phỏng được phần nào quá trình cài đặt và thiết lập thông số, giúp chúng ta hình dung được cách vận hành, điều khiển cũng như có cái nhìn khái quát hơn trong việc sử dụng robot hút bụi qua đường truyền không dây.

CHƯƠNG 7: THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

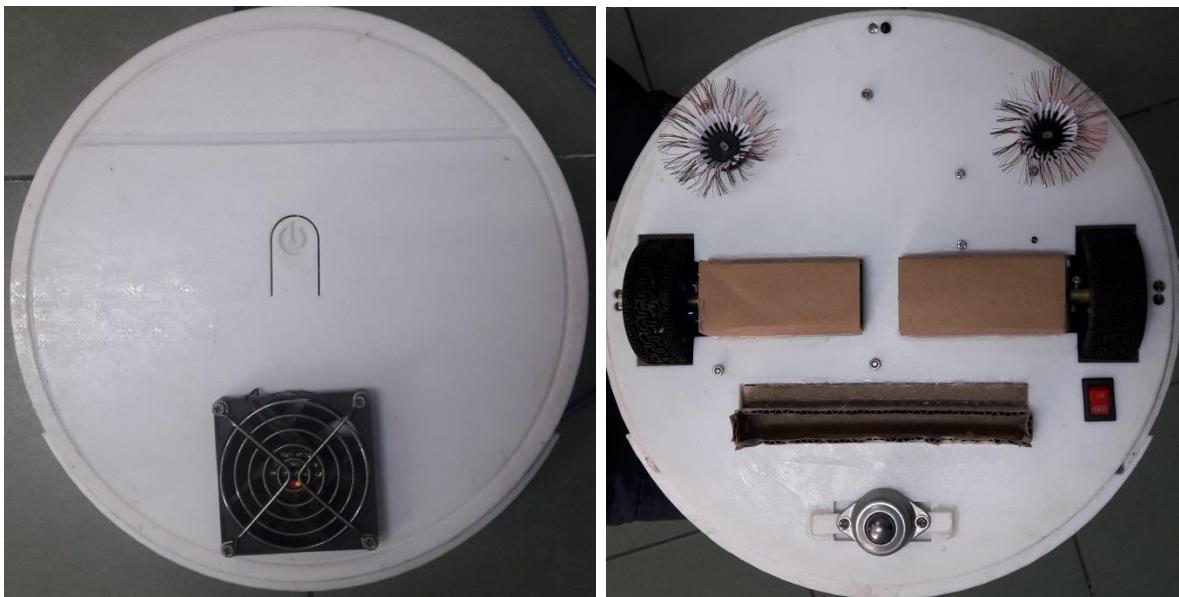
Nội dung chính

- 7.1 Kết quả đạt được**
- 7.2 Thực nghiệm quá trình hoạt động của robot**
- 7.3 Nhận xét và đánh giá kết quả thực nghiệm**
- 7.4 Một số điểm cần khắc phục**

7.1 Kết quả đạt được

7.1.1 Phần cơ khí và mạch điều khiển

Sau quá trình thiết kế, in ấn và sửa đổi sau in cho phù hợp với điều khiển vận hành thực tế kết hợp với việc lắp đặt các thiết bị, linh kiện vào bên trong thân thì sản phẩm có cái nhìn tổng quan như sau:



Hình 7.1. Nắp và đế robot thực tế



Hình 7.2. Thân phía bên trái robot thực tế



Hình 7.3. Thân phía trước robot thực tế

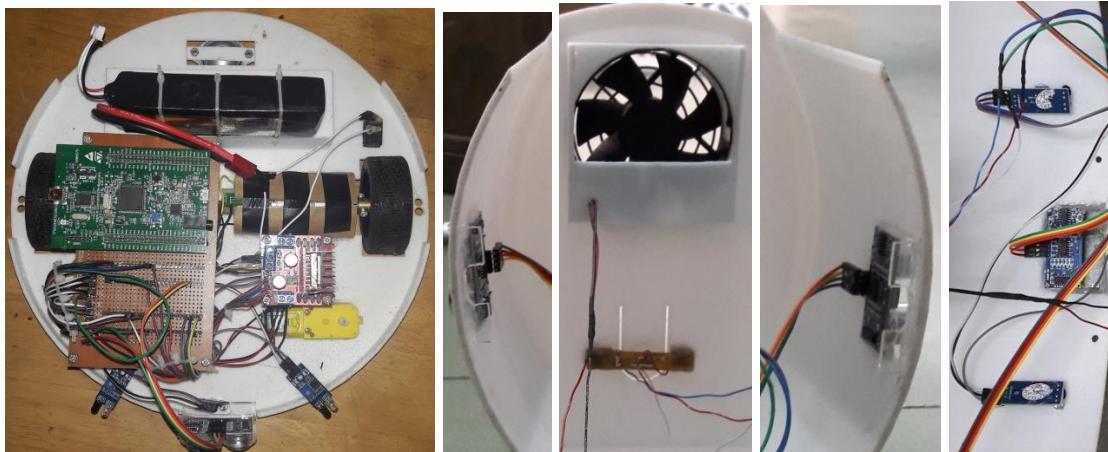
Sản phẩm có đường kính là 30cm, cao 10cm với các cạnh được bo tròn thuận tiện cho việc di chuyển. Bên cạnh đó, robot gần như có đủ các bộ phận cần thiết cho quá trình vận hành, giống với đa số các sản phẩm máy hút bụi hiện có trên thị trường gồm 2 bánh xe điều hướng, 1 bánh xe đa hướng ở giữa, khoang chứa bụi, cụm chổi quét bụi,... và các cảm biến được bố trí thích hợp trên khung robot phục vụ cho việc phát hiện vật cản, vùng chêch lệch độ cao khi di chuyển.

Các bộ phận lần lượt được lắp vừa vặn vào đế của robot. Trong đó các board kit sẽ được gắn cố định bằng cách bắt vít thông qua các lỗ trống đã được khoan từ trước xuống mặt dưới của robot. Các bộ phận khác cũng được dán cố định chắc chắn bằng keo. Có thể lợi dụng khoảng không gian trống bên dưới kit điều khiển để đặt module thời gian thực. Các bus dây cũng được cột cẩn thận để tránh tình trạng rơi rớt hoặc đứt trong quá trình vận hành.

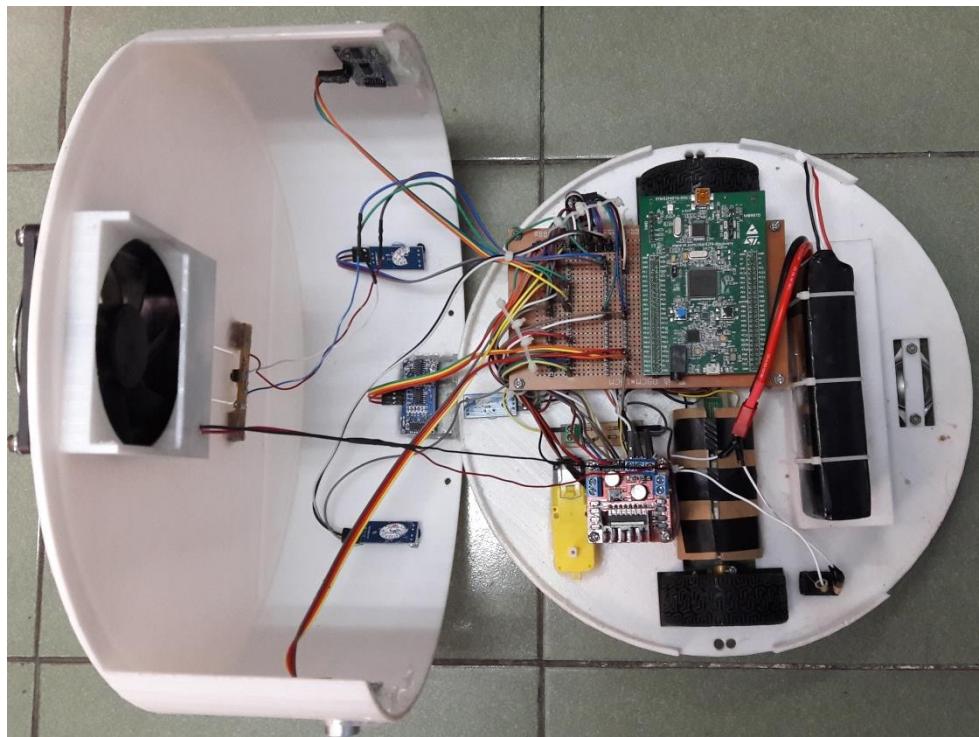
Còn một số cảm biến do phải được gắn với khung của robot nên tạm thời không thể cố định vào đế mà chỉ có thể nối dây trực tiếp từ cảm biến đến vi điều khiển khi lắp đầy đủ các bộ phận lại với nhau. Việc lắp dây này cũng cần sự cẩn thận, tránh tình trạng dây bị vuông vào bánh xe khi hoạt động.

Pin nguồn của robot được đặc biệt để ở phía đuôi (phía trên miệng hút bụi, cạnh hộp bụi) một phần sẽ tiết kiệm diện tích trong thân, mặt khác nó còn có tác dụng đối

trọng cho robot cân bằng (pin có khối lượng khá nặng). Vì phần lớn linh kiện được đặt ở phía trước của robot nên khối lượng sẽ tập trung về phía trước, mà ở phía trước lại không có bánh xe để chịu lực nên robot sẽ bị nhổng, không thể chạy được.



Hình 7.4. Lắp linh kiện phía trong robot



Hình 7.5. Tổng quát phía trong thân robot

➤ **Nhận xét:** thiết kế sản phẩm còn đơn giản, không quá cầu kì về mặt hình thức bên ngoài, chất lượng nhựa in cứng, đẹp và thân thiện môi trường. Còn nhiều điều

chỉnh, thiết kế thêm sau khi in tuy nhiên phần lắp ráp vẫn đáp ứng được các yêu cầu đề ra khi thiết kế. Các nối dây không tối ưu vì có quá nhiều dây chồng lên nhau. Vẫn còn một vấn đề về công suất hút bụi chưa được giải quyết triệt để vì nó liên quan đến nhiều khía cạnh khác của phần cứng cần được phát triển thêm như loại quạt, mạch công suất, nguồn nuôi,...

Ưu điểm:

- Giống với mô hình thực tế, có thể theo dõi một cách trực quan.
- Dễ dàng thiết kế bằng phần mềm Soliworks.
- Sản phẩm chắc chắn, bền, nguyên liệu thân thiện với môi trường.

Khuyết điểm:

- Kích thước còn to nên di chuyển không linh hoạt trong góc nhỏ hẹp.
- Giá thành tương đối lớn.
- Cần nhiều sự chỉnh sửa sau thiết kế và in cho phù hợp.

7.1.2 Phần lập trình vi điều khiển

Đối với sản phẩm, vi điều khiển được xem như là vị trí trung tâm, ‘đầu não’ xử lý toàn bộ thông tin nhận được từ các module khác gửi đến, tính toán và cho ra được tín hiệu thích hợp, điều khiển robot. Trong thời gian thực hiện đề tài, nhóm sử dụng chương trình Keil C version 5 để kết nối và lập trình cho Kit STM32F407. Đây là chương trình quen thuộc, dễ sử dụng, được dùng rộng rãi trong lập trình các dòng vi điều khiển. Ngoài ra, nhóm còn sử dụng thêm phần mềm STM studio để có thể theo dõi các thông số, quan sát giá trị của biến được khai báo bên trong Kit STM32F407 khi nó hoạt động, thuận tiện cho việc phát hiện lỗi và chỉnh sửa.

Ở đây, bằng ngôn ngữ C, nhóm đã chuyển hóa tất cả các giải thuật nêu trên thành các dòng lệnh, mỗi giải thuật, phương pháp được sử dụng sẽ ứng với một hàm con trong file ‘main.c’. Công việc này giúp cho việc kết hợp cũng như kiểm tra, tái sử dụng từng module trở nên dễ dàng.

⇒ Vì điều khiển sau khi được lập trình đã có thể đọc giá trị của cảm biến và giao tiếp với các module thành phần cũng như áp dụng được các giải thuật để tính toán đưa ra tín hiệu điều khiển robot một cách hợp lý, chính xác.

7.1.3 Phần lập trình ứng dụng

Về phần kết nối điều khiển không dây, nhóm đã xây dựng thành công một giao diện người dùng có thể kết nối trực tiếp với sản phẩm qua sóng Wifi và tích hợp các tính năng chính như sau:

- Cài đặt robot với 2 chế độ: Tự động và bằng tay
- Thiết lập thời gian hoàn thành công việc hay thời gian bắt đầu làm việc (chỉ ở chế độ tự động)
- Theo dõi trạng thái pin của sản phẩm
- Hiển thị thời gian thực và thời gian vận hành còn lại của robot (khi đã được thiết lập thời gian làm việc)
- Ghi chép quá trình làm việc

Ưu điểm:

- Mô phỏng được giao diện người dùng để điều khiển robot
- Dễ lập trình, áp dụng được những kiến thức sẵn có.

Khuyết điểm:

- Cần được phát triển thành ứng dụng android phù hợp với thực tế.

Chính vì vậy, trong tương lai phát triển sản phẩm, nhóm sẽ tìm hiểu và học hỏi thêm về kiến thức lập trình trên android để có thể thiết kế tạo nên một ứng dụng có đủ các tính năng trên và sử dụng được trên smartphone.

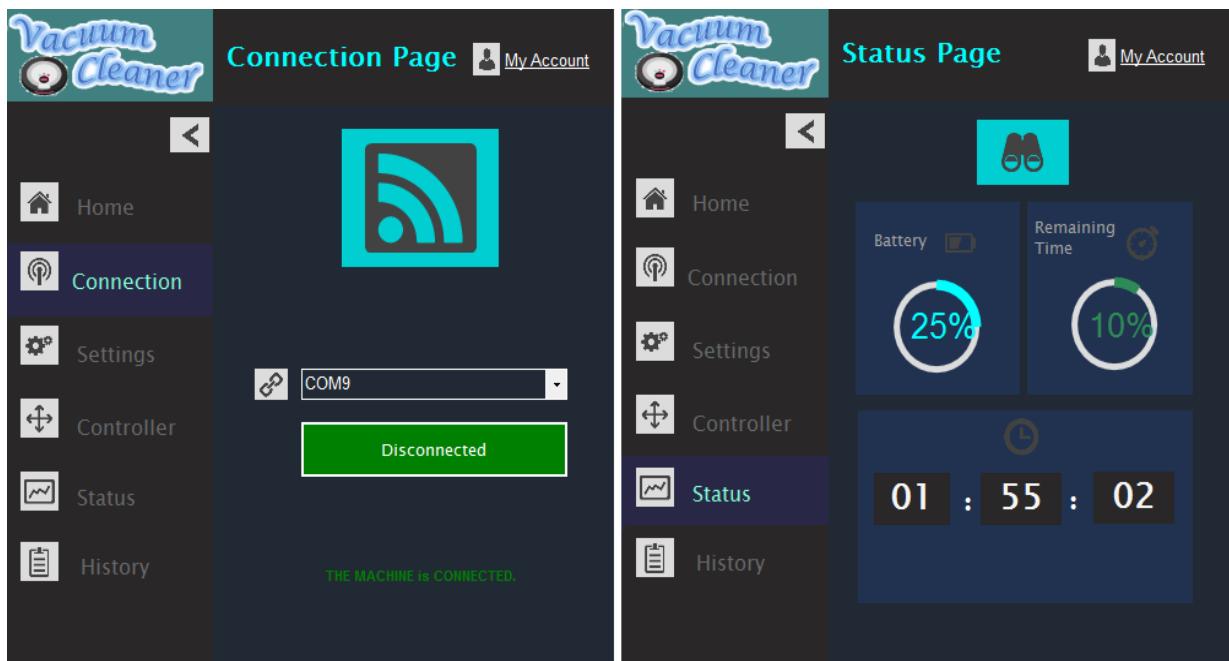
7.2 Thực nghiệm quá trình hoạt động của robot

7.2.1 Quá trình truyền nhận không dây

Trong phần thực nghiệm này, nhóm tiến hành mở giao diện người dùng và kết nối với sản phẩm bằng việc lựa chọn cổng COM ứng với module ESP đã cắm vào. Sau

đó, lần lượt cài đặt các chế độ bằng tay và tự động, thiết lập thời gian, ... Đồng thời theo dõi các thông số hiển thị trên giao diện.

Kết quả cho thấy, robot vận hành đúng theo các chế độ đã được cài đặt, có thể bắt đầu làm việc và hoạt động trong khoảng thời gian thiết lập trước một cách chính xác. Bên cạnh đó, giá trị pin và thời gian thực cũng được giao diện xử lý, hiển thị đúng với thông số hiện tại của robot và thực tế. Khoảng cách xa nhất mà giao diện có thể kết nối được với sản phẩm là **khoảng 20m**.



Hình 7.6. Giao diện kết nối thành công và hiển thị thông số robot

Tuy nhiên, quá trình truyền nhận dữ liệu vẫn còn xảy ra độ trễ không mong muốn (lớn nhất khoảng 1s), chủ yếu là khi thực thi truyền chuỗi dữ liệu cài đặt xuống robot. Do đó, quá trình thiết lập, điều khiển robot sẽ mất ổn định khi người dùng thực hiện thao tác nhấn nhanh các nút trên giao diện quá nhanh khiến cho vi điều khiển không thể nhận và xử lý kịp thời các chuỗi dữ liệu truyền tới.

Ưu điểm:

- Đáp ứng được yêu cầu đặt ra là điều khiển không dây đặt ra
- Dễ lập trình sử dụng

Khuyết điểm:

- Chưa phải là cách tối ưu nhất, tốt nhất
- Vẫn còn độ trễ trong quá trình truyền nhận

7.2.2 *Khả năng di chuyển và tránh vật cản*

Trong phần thử nghiệm này, robot được cài đặt chế độ tự động và chạy thử lần lượt trong các môi trường không có hoặc có ít hay có nhiều vật cản. Hướng di chuyển robot tương đối ổn định khi đi thẳng. Sản phẩm có thể di chuyển theo giải thuật zigzag nhưng vẫn bị sai lệch khi quay 90^0 , dẫn tới sai số tích lũy ngày càng tăng làm cho hướng đi của robot chưa thực sự chính xác và như mong muốn. Ngoài ra, sản phẩm chỉ có thể di chuyển tốt trên bề mặt phẳng, gạch men. Đối với các mặt phẳng gồ ghề, robot dễ bị mắc kẹt hay tăng tốc bất ngờ khó kiểm soát.

Robot có thể tránh được đa số các vật cản lớn thường gặp trong gia đình nhờ cảm biến siêu âm. Đối với trường hợp mắc kẹt hay không thể chuyển động khi gặp vật cản, Encoder có thể phát hiện và chuyển hướng để giải quyết. Dưới đây là bảng thống kê một số trường hợp của robot khi gặp các loại vật cản.

Các vật cản thường gặp	Số lần ngưng hoạt động
Tường	1
Khe hẹp	7
Chân bàn, ghế	2
Bậc thang	4

Bảng 7. 1. Kết quả robot tránh vật cản với 10 lần chạy thử cho mỗi trường hợp

Ưu điểm:

- Tránh vật cản phía trước tương đối tốt và có thể xử lý chuyển hướng đối với một số trường hợp bị mắc kẹt
- Robot có thể bám sát hướng cần đi
- Giải thuật Zigzag được ứng dụng hiệu quả, hạn chế sự trùng lặp khi di

chuyển

Khuyết điểm:

- Giải thuật chưa chua thực sự hoàn thiện, vẫn còn sai lệch khi quay
- Vẫn còn xảy ra va chạm đối với các vật cản bên hông robot hoặc không nằm trong góc quét của cảm biến siêu âm.
- Chỉ di chuyển được ở các mặt phẳng phẳng, không gồ ghề

7.2.3 Khả năng hút bụi của robot

Để đánh giá khả năng hút bụi của robot, nhóm thực hiện giả lập môi trường có bụi bẩn vừa và nhỏ (bụi li ti hay những mảnh giấy vụn, hạt xốp...) rồi lần lượt thay đổi tốc độ hút của động cơ gắn trên khung sản phẩm để chạy thử từng trường hợp. Kết quả cho thấy, robot chỉ làm việc hiệu quả với các loại bụi bẩn có trọng lượng nhẹ và nhỏ, còn đối với các loại rác bẩn khác như mảnh giấy hay vụn bánh mì... , sản phẩm vẫn chưa thể hút hết được. Dưới đây là một số hình ảnh trước và sau khi robot thực hiện tác vụ hút đối với một số loại bụi bẩn.



Hình 7.7. Số lượng hạt xốp trước và sau khi được vệ sinh



Hình 7.8. Số lượng mảnh giấy vụn trước và sau khi được vệ sinh

Với công dụng chỉ hút được các loại bụi nhỏ, nhẹ, sản phẩm chỉ phù hợp cho những không gian cần làm sạch bẩn bụi, ít rác có kích thước lớn. Chính vì vậy, trong tương lai phát triển sản phẩm, nhóm sẽ thiết kế và tìm ra phương pháp làm tăng công suất hút bụi của robot, giúp hiệu quả làm sạch cao hơn.

7.3 Nhận xét và đánh giá kết quả thực nghiệm

7.3.1 Tính hiệu quả

Sản phẩm hoạt động tương đối hiệu quả trong môi trường không có hoặc ít vật cản, bề mặt di chuyển phẳng, sáng. Tùy vào từng khu vực có nhiều hay ít vật cản mà diện tích được làm sạch khác nhau. Nhóm tiến hành chạy thử nghiệm sản phẩm ở các khu vực trong nhà và ghi chép lại kết quả vào bảng dưới đây như sau:

Khu vực vệ sinh	Diện tích được làm sạch
Phòng khách	70%

Phòng ngủ	70%
Phòng Bếp	60%
Hành lang	80%

Bảng 7. 2. Diện tích sàn được làm sạch ở từng khu vực trong nhà

Các vị trí góc khuất hay góc cạnh vẫn chưa thể làm sạch. Đồng thời, với những hạn chế về kích thước, đối các không gian dưới gầm giường, bàn,...robot vẫn chưa thể di chuyển linh hoạt hoặc hướng tới để thực hiện công việc vệ sinh. Đây là những điểm thiết kế cần phải được cải thiện trong thời gian nghiên cứu và phát triển sản phẩm sau này.

7.3.2 Nguồn pin và giá thành

➤ *Nguồn pin:*

Nguồn được sử dụng trong đề tài này là pin Lipo **12V, 5200mAh** và cấp cho toàn bộ hệ thống bao gồm các module thành phần và các động cơ điều hướng, hút bụi trong sản phẩm. Vì điều khiển sử dụng nguồn nuôi từ IC mạch lái LH298. Vì module có tích hợp mạch ổn áp 5V, nên hoạt động ổn định, không gây hiện tượng quá dòng dẫn đến việc hư hỏng Kit nhưng dễ phát nóng khi sử dụng trong thời gian dài. Quá trình thử nghiệm thực tế cho thấy robot có thể hoạt động liên tục trong **3-4 giờ**. Quá trình sạc pin cần có thiết bị chuyên dùng, mất khoảng 2 giờ khi dùng loại tốt và 8 giờ đối với loại bình thường.

Ưu điểm:

- Nguồn hoạt động ổn định, thời gian sử dụng tương đối lâu

Khuyết điểm:

- Giá thành khá cao
- Thời gian sạc còn lâu

➤ *Giá thành*

Một trong những tiêu chí được quan tâm trong việc đánh giá, chọn mua robot

hút bụi thông minh hiện nay đó chính là giá thành. Về thương mại, đa số những chiếc máy hút bụi thông minh hiện nay đều có giá tương đối cao. Theo khảo sát, đối với các loại máy robot hút bụi có những tính năng mức trung bình, mức giá dao động từ khoảng 5-8 triệu đồng, còn đối với dòng cao cấp, có thương hiệu uy tín thì giá thành rơi vào khoảng 10-20 triệu tùy loại. Chính vì vậy, việc chế tạo thiết kế một chiếc máy hút bụi tự động với giá thành thấp nhưng có đầy đủ các tính năng thông minh sẽ mang lại tính cạnh tranh rất cao. Dưới đây là bảng liệt kê giá thành của các module và bộ phận mà nhóm đã sử dụng để chế tạo sản phẩm trong quá trình thực hiện đề tài. Từ đó có thể đánh giá một cách khách quan giá trị thương mại của sản phẩm.

Linh kiện	Số lượng	Đơn giá (VND)
Thiết kế mô hình	1	1 350 000
Pin Lipo+sạc	1	1 100 000
STM32F407	1	650 000
Động cơ Encoder	2	150 000
Động cơ chổi quét	2	25 000
Động cơ hút bụi	1	40 000
Mạch lái L298	2	45 000
Module ESP 8266	2	68 000
Module RTC DS3231	1	40 000
Cảm biến siêu âm SRF04	3	30 000
Cảm biến hồng ngoại	3	20 000
Bánh xe điều hướng	2	50 000
Bánh xe đa hướng	1	35 000

Dây, ốc, vít,..	N/A	50 000
Tổng cộng		4 091 000

Bảng 7.3. Bảng thống kê giá thành linh kiện

7.4 Một số điểm cần khắc phục

Qua quá trình thực nghiệm và xem xét kết quả, nhìn chung sản phẩm đã đạt được những yêu cầu đề ra, thực hiện được công việc làm sạch giống với các sản phẩm thực tế khác trên thị trường. Tuy nhiên, robot vẫn còn một số mặt hạn chế cần phải khắc phục như sau:

- Phần cơ khí chưa được hoàn hảo, kích thước còn hơi lớn và cồng kềnh không thể di chuyển làm sạch ở các gầm ghế, tủ, góc hẹp...
- Công suất máy hút còn thấp, chỉ hút được các loại bụi bẩn nhỏ, nhẹ.
- Ở những khu vực màu tối, cảm biến hồng ngoại không phát huy được hiệu quả và dễ bị nhiễu với ánh sáng mặt trời.
- Chỉ có thể phát hiện tốt vật cản phía trước robot, còn bên hông hay phía sau khi di chuyển lùi hoặc quay vẫn chưa xác định được vật cản dẫn tới va đập. Vấn đề có thể được giải quyết bằng cách trang bị thêm các cảm biến xung quanh robot để tăng khả năng thích ứng, phát hiện nguy hiểm.
- Giao diện ứng dụng người chưa phù hợp với thực tế và vẫn còn độ trễ trong quá trình truyền nhận dữ liệu.

CHƯƠNG 8: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Nội dung chính

8.1 Kết luận đề tài

8.2 Hướng phát triển

8.1 Kết luận đề tài

Trong đề tài này, nhóm chúng em đã thiết kế và chế tạo được sản phẩm robot hút bụi tự động với một số tính năng cơ bản, có thể vận hành ở những môi trường khác nhau và đặc biệt hoạt động hiệu quả hơn trong môi trường có ít vật cản, cần làm sạch bụi bẩn vừa và nhỏ. Thông qua việc thử nghiệm thực tế, nhóm cũng đã tinh chỉnh một vài thông số cũng như chỉnh sửa phần thiết kế để sản phẩm hoạt động tốt hơn. Tuy vậy, robot vẫn còn những nhược điểm cần phải khắc phục và phát triển để ngày một hoàn thiện hơn.

Mặc dù chiếc máy không thể bắt kịp những tính năng mới mẻ, theo kịp xu thế của các dòng sản phẩm khác trên thị trường nhưng nhìn chung vẫn đạt được những kết quả như yêu cầu đặt ra, giúp chúng em có cái nhìn sâu hơn và có những trải nghiệm thú vị về ngành lập trình nhúng, chế tạo robot thông minh.

8.2 Hướng phát triển

Hướng phát triển của đề tài trong tương lai là có thể phát triển sản phẩm rộng hơn với một số tính năng khác như ứng dụng AI để vẽ lại bản đồ, đồng thời ghi nhớ những khu vực đường đi để quá trình làm việc, tránh vật cản trở nên hiệu quả hơn, rút ngắn thời gian hoạt động, tiết kiệm năng lượng. Ngoài ra, nhóm sẽ tìm hiểu thêm về kiến thức lập trình Android để xây dựng giao diện điều khiển thật sự, có thể hiển thị và sử dụng trên smartphone với những chức năng cài đặt như đã mô phỏng trong đề tài, phù hợp với thực tế hiện nay.

TÀI LIỆU THAM KHẢO

- [1] arduino.vn, “Lập trình ESP8266 bằng Arduino IDE” [Online]
<http://arduino.vn/bai-viet/1172-lap-trinh-esp8266-bang-arduino-ide>
- [2] Xuân Thiếp, "Học lập trình ARM STM32F4 Discovery," [Online]
<https://www.youtube.com/watch?v=deQxE Pv0nkY&list=PLZgzYRCEZ0Ufr47bGogY0hIonJday4eWr>
- [3] tapit.vn, “Giao tiếp I2C trên STM32F103 với module RTC DS3231” [Online]
<https://tapit.vn/giao-tiep-i2c-tren-stm32f103-voi-module-rtc-ds3131/>
- [4] Trung tâm Đào tạo Solidworks Thành Đô, “Học Solidworks cơ bản” [Online]
https://www.youtube.com/watch?v=HUEGfQkM3sY&list=PLFx-mAYr-Hw1d3UOJwVFtIOf_inrBvxOJ
- [5] DS3231 Datasheet, [Online]
<https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>
- [6] . STM32F4 Discovery User’s Manual, January, 2012
- [7] STM32F4 Reference Manual, September, 2014
- [8] Kazi Mahmud Hasan, Abdullah -Al-Nahid, Khondker Jahid Reza, “Path Planning Algorithm Development for Autonomous Vacuum Cleaner robots” [Paper]
- [9] robotc.net, “Using encoders to drive strength” [Online]
http://www.robotc.net/wikiarchive/Tutorials/Arduino_Projects/Mobile_robots/VEX/Using_encoders_to_drive_straight?fbclid=IwAR18_s2a9J7s_JVkfivxAbE3uJfpFr1gNGXOQ8aA1_ff8fme5nGX62t_bKo

Tài liệu tham khảo

- [10] robot-vacuum-cleaners.com, “Most common robotic cleaning patterns” [Online]
<https://robot-vacuum-cleaners.com/most-common-robotic-cleaning-patterns/?fbclid=IwAR1LbwX2UUE5bWNRpm7KP1BCDYayQIufeQMwUW3E0gAHoepTSe-aIQwZh-o>

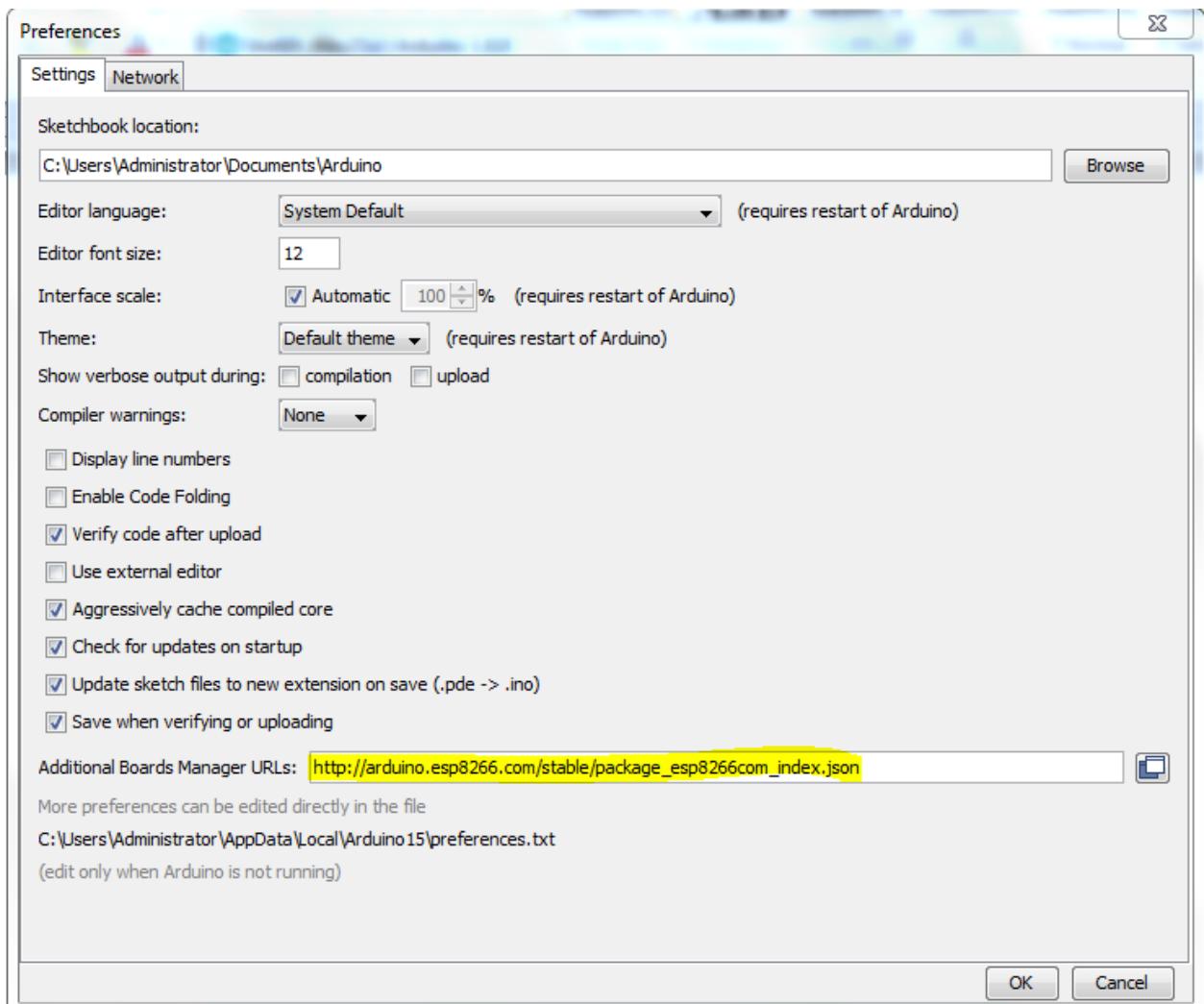
Phụ lục: Cách nạp code module wifi ESP dùng USB UART

Để nạp được code cho ESP cần có 2 công cụ quan trọng là: mạch nạp USB-UART (CP2102) và phần mềm Arduino IDE.

Bước 1: cài đặt driver ESP trong Arduino IDE

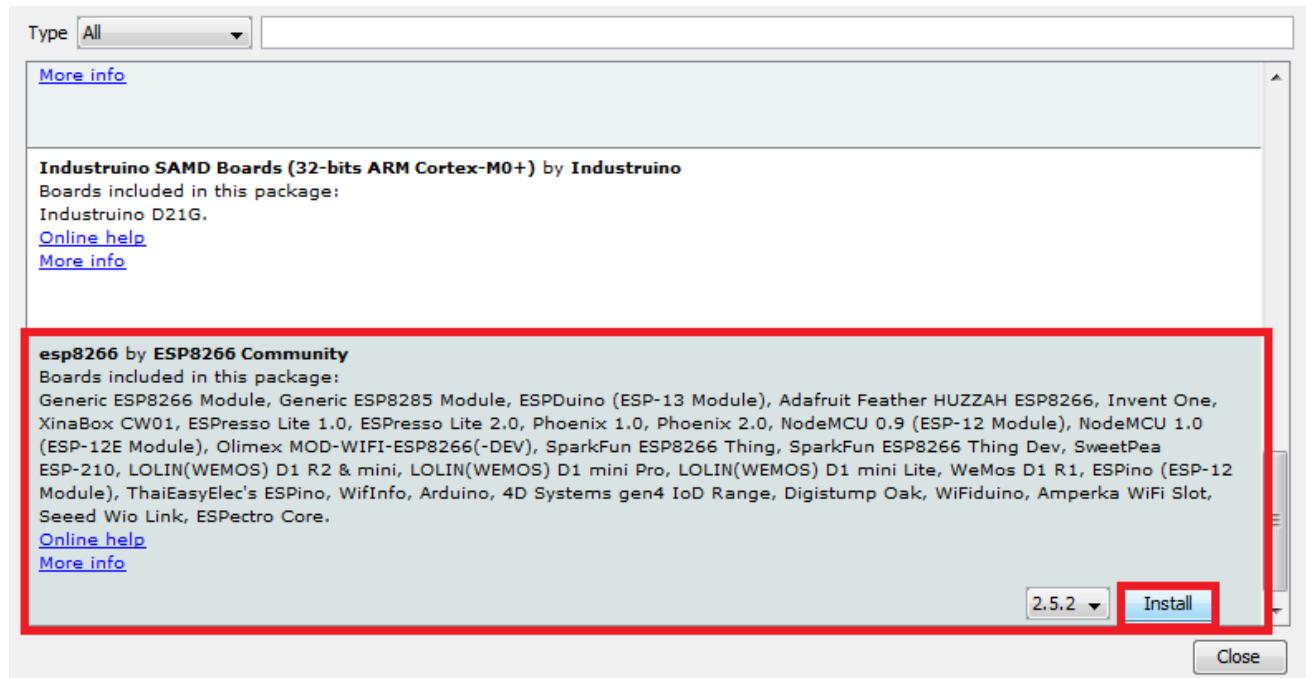
_Vào **File -> Preferences** copy dòng bên dưới vào mục *Additional Boards Manager URLs* và nhấn OK

“http://arduino.esp8266.com/stable/package_esp8266com_index.json”



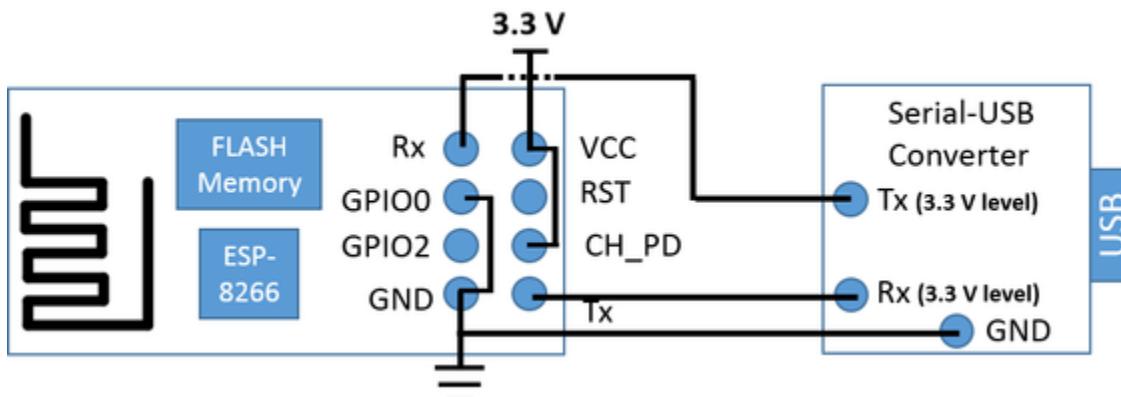
_Tiếp theo, vào **Tools -> Board: “tên board” -> Board Manager...** sẽ xuất hiện hộp thoại *Boards Manager*. Tìm mục “**esp8266 by ESP8266 Community**” nhấn

Install và đợi quá trình tải xuống và cài đặt hoàn tất.



Bước 2: kết nối ESP và USB-UART

_Sơ đồ cắm ESP với USB-UART

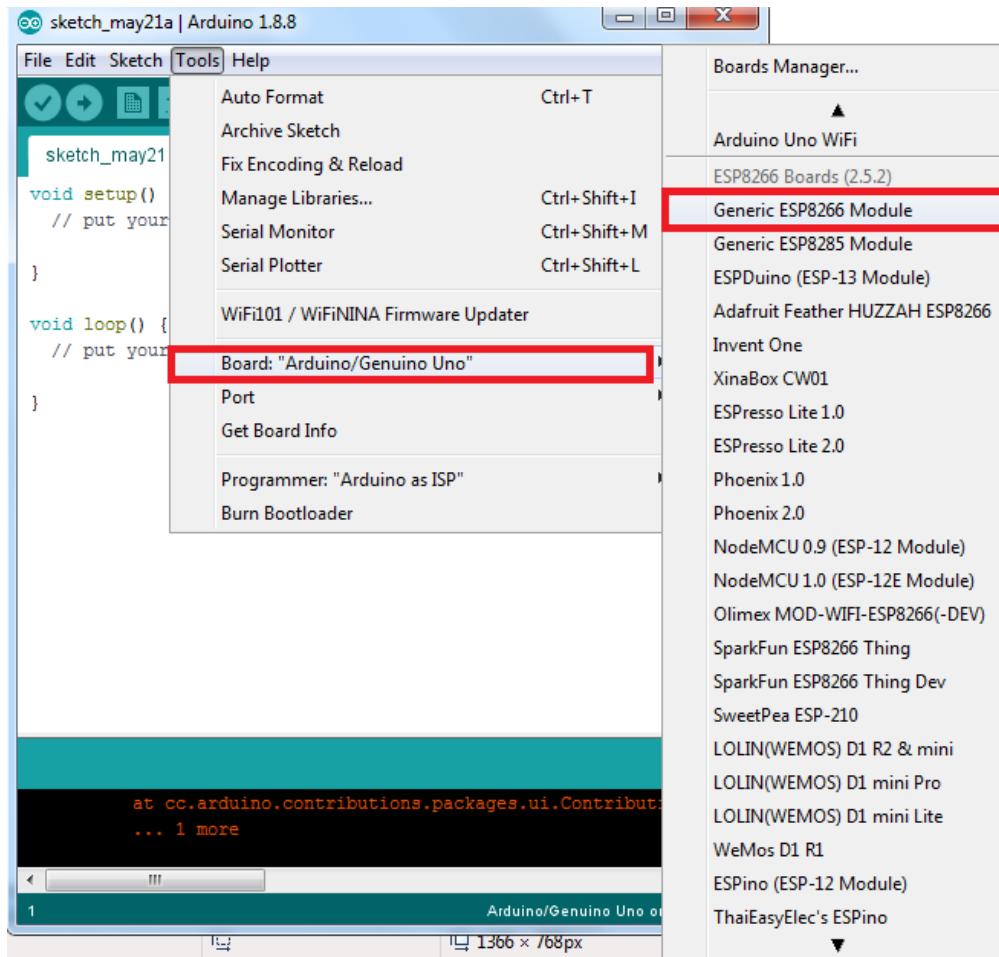


Lưu ý: _ Phải đưa ESP vào chế độ Flashmode (chế độ nạp) bằng cách nối GPIO0 về GND.

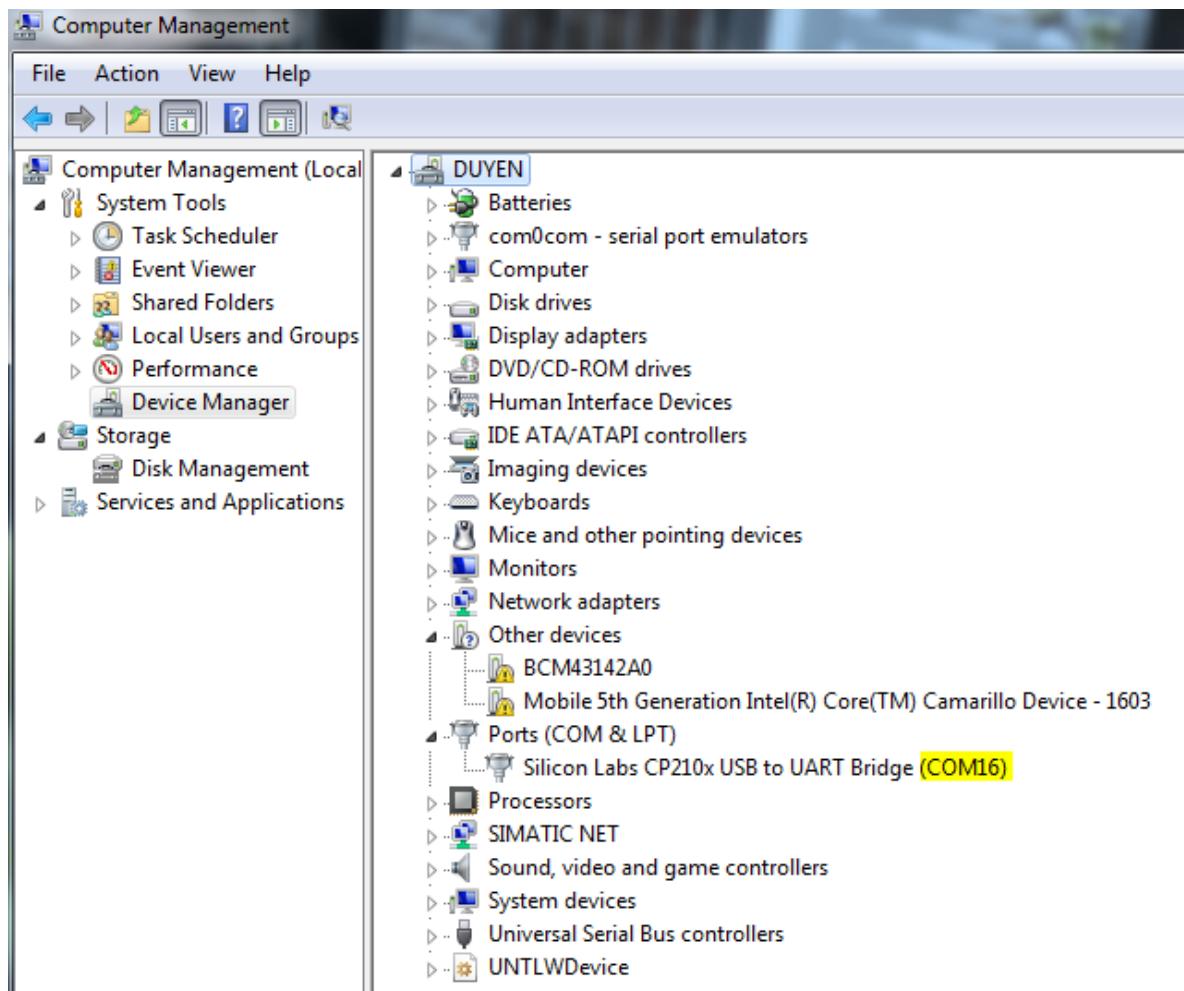
_ Nguồn cấp cho ESP là 3.3V, có thể lấy trực tiếp từ USB-UART (nếu có) hoặc cấp nguồn rời. Không được cấp nguồn 5V, sẽ làm hư ESP.

Bước 3: Nạp code

_ Tại giao diện chính của Arduino, vào **Tools -> Board: “tên board”** -> **Chọn “Generic ESP8266 Module”** để chọn board nạp. Với Board ESP8266 V1 thì chọn như hình, còn với ESP8266 nodeMCU thì chọn dòng : **“NodeMCU 1.0 (ESP-12E Module”**



- _ Vào **File -> Open** để mở đoạn code cần nạp
- _ Cắm USB_UART vào cổng USB và vào Device Manager để xác định tên port của cổng USB vừa cắm vào. (Ở đây là port 16)



- _ Vào **Tools -> Port -> Chọn port** vừa xác định
- _ Tiến hành nạp code. Khi phần thông báo hiển thị như hình tức là quá trình nạp đã thành công.

```
Writing at 0x00000000... (7 %)
Writing at 0x00004000... (15 %)
Writing at 0x00008000... (23 %)
Writing at 0x0000c000... (30 %)
Writing at 0x00010000... (38 %)
Writing at 0x00014000... (46 %)
Writing at 0x00018000... (53 %)
Writing at 0x0001c000... (61 %)
```

Compiling sketch... Uploading... Done uploading.