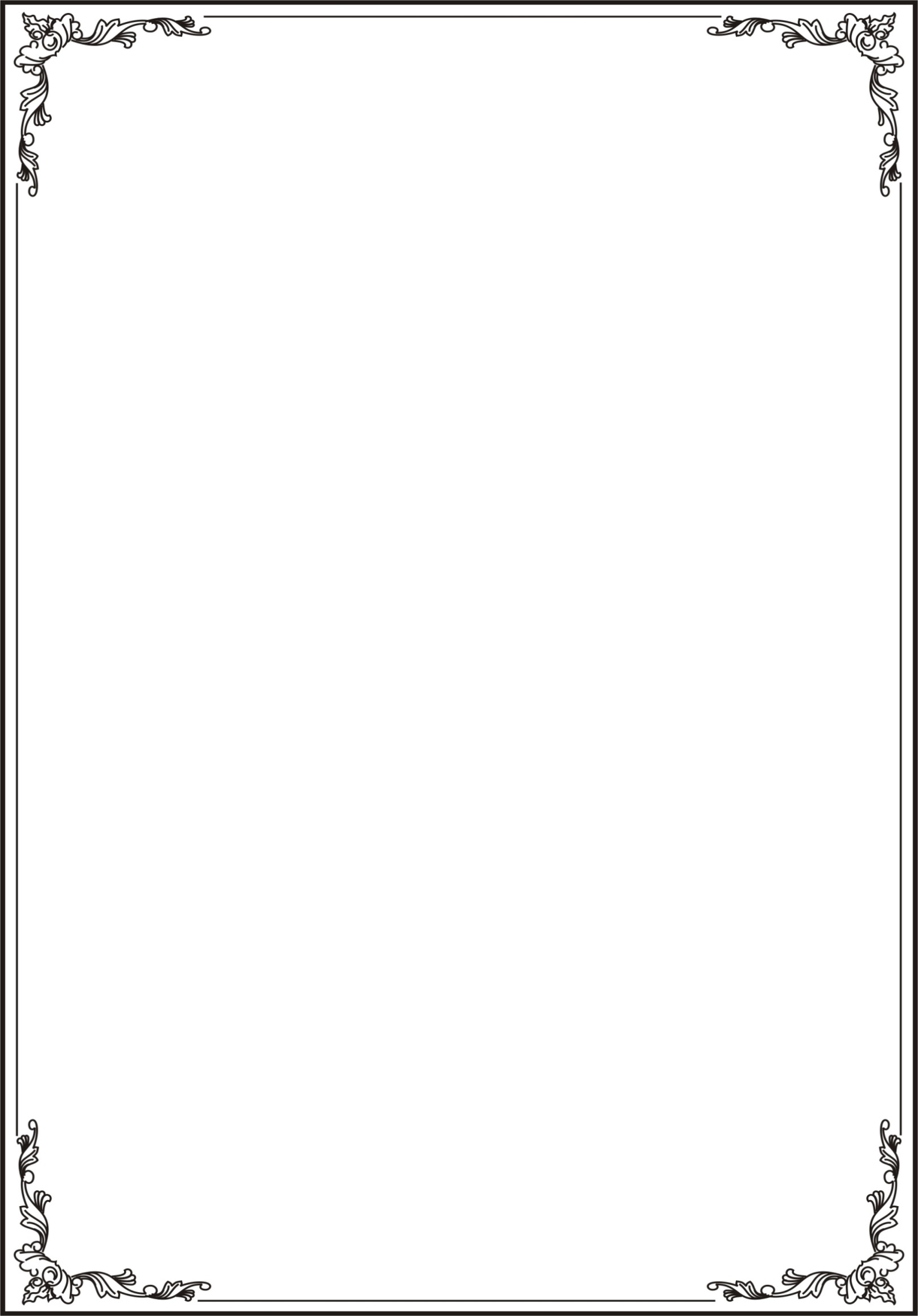
****

**LUẬN VĂN TỐT NGHIỆP**

Đề tài: **ROBOT HÚT BỤI TỰ ĐỘNG**

**(**ROBOTIC VACUUM CLEANER**)**

GVHD: KS. Bùi Thanh Huyền

Nhóm :

**Lê Thị Mỹ Duyên 1510525**

**Nguyễn Tuấn 1513831**

TP.HCM ,ngày …...tháng ..…năm…..

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐH BÁCH KHOA CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Thành phố Hồ Chí Minh Độc Lập – Tự Do – Hạnh Phúc

🙣 🟑 🙡 🙣 🟑 🙡

Số: \_/BKĐT Khoa: Điện – Điện tử Bộ Môn: Tự Động Hóa

**NHIỆM VỤ LUẬN VĂN TỐT NGHIỆP**

|  |  |  |
| --- | --- | --- |
| Họ và tên: | **NGUYỄN TUẤN** | MSSV: **1513831** |
| Họ và tên: | **LÊ THỊ MỸ DUYÊN** | MSSV: **1510525** |
| Ngành: | **TỰ ĐỘNG HÓA** |  |

1. **Đầu đề luận văn: “Robot hút bụi tự động”**
2. **Nhiệm vụ** ( Yêu cầu về nội dung và số liệu ban đầu):

......................................................................................................................................

......................................................................................................................................

......................................................................................................................................

......................................................................................................................................

1. **Ngày giao nhiệm vụ luận văn:**
2. **Ngày hoàn thành nhiệm vụ:**
3. **Họ và tên người hướng dẫn: Phần hướng dẫn**

................................................

Nội dung và yêu cầu LVTN đã được thông qua Bộ Môn.

Ngày . ........ tháng ........... năm 2019

CHỦ NHIỆM BỘ MÔN NGƯỜI HƯỚNG DẪN CHÍNH

(*Ký và ghi rõ họ tên*) (*Ký và ghi rõ họ tên*)

PHẦN DÀNH CHO KHOA, BỘ MÔN:

Người duyệt (chấm sơ bộ): Đơn vị:

Ngày bảo vệ: Điểm tổng kết:

Nơi lưu trữ luận văn:

TRƯỜNG ĐH BÁCH KHOA CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

KHOA ĐIỆN – ĐIỆN TỬ Độc Lập – Tự Do – Hạnh Phúc

BỘ MÔN: ĐIỀU KHIỂN TỰ ĐỘNG --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**

##### Đề tài: “Robot hút bụi tự động”

1. Họ tên người phản biện:
2. Tổng quát về bản thuyết minh:

Số trang ........ Số chương ........

Số bảng số liệu ........ Số hình vẽ ........

Số tài liệu tham khảo ........ Phần mềm tính toán ........

1. Những ưu điểm chính của LVTN:
2. Những thiếu sót chính của LVTN:
3. Về thái độ làm việc của sinh viên:
4. Đề nghị: Được bảo vệ 🞏, Bổ sung thêm để bảo vệ 🞏, Không được bảo vệ 🞏 .
5. Đánh giá chung (*bằng chữ: Giỏi, Khá, TB*): Điểm …………**/10**

TRƯỜNG ĐH BÁCH KHOA CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

KHOA ĐIỆN – ĐIỆN TỬ Độc Lập – Tự Do – Hạnh Phúc

BỘ MÔN: ĐIỀU KHIỂN TỰ ĐỘNG --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**

##### Đề tài: “Robot hút bụi tự động”

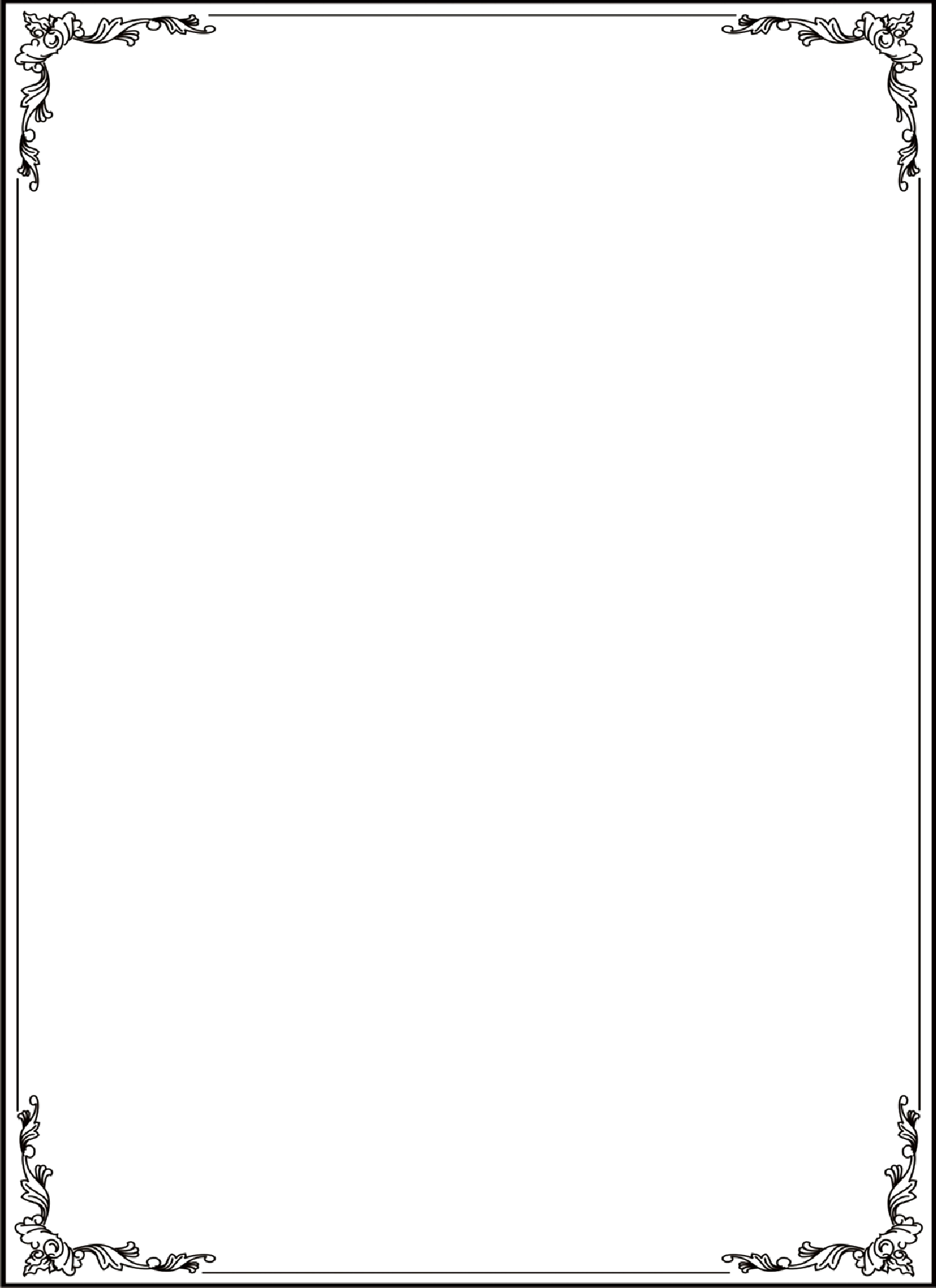
1. Họ tên người phản biện:
2. Tổng quát về bản thuyết minh:

Số trang ........ Số chương ........

Số bảng số liệu ........ Số hình vẽ ........

Số tài liệu tham khảo ........ Phần mềm tính toán ........

1. Những ưu điểm chính của LVTN:
2. Những thiếu sót chính của LVTN:
3. Về thái độ làm việc của sinh viên:
4. Đề nghị: Được bảo vệ 🞏, Bổ sung thêm để bảo vệ 🞏 , Không được bảo vệ 🞏 .
5. Đánh giá chung (*bằng chữ: Giỏi, Khá, TB*): Điểm …………**/10**



# 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 tôi trong thời gian học tập tại Trường Đại Học Bách Khoa và trong quá trình hoàn 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*

NGUYỄN TUẤN & LÊ THỊ MỸ DUYÊN

# LỜI NÓI ĐẦU

*Theo xã hội hiện đại ngày nay, khoa học kĩ thuật phát triển rất mạnh mẽ và và đi cùng với những thành tựu vượt bậc đó là việc tạo ra những sản phẩm ứng dụng vào thực tế trong nhiều lĩnh vực: y tế , giáo dục, nông nghiệp, sản phẩm dân dụng… . Nhìn thấy được xu hướng này và mong muốn  áp dụng kiến thức mà mình đã học được trong hơn bốn năm ở giảng đường đại học, nhóm chúng em đã quyết định làm một sản phẩm mang tình ứng dụng vào đời sống thực tiễn. Sản phẩm đó chính là một robot hút bụi tự động.*

*Trong khuôn khổ của luận văn, nhóm sẽ tập trung xây dựng nghiên cứu xây dựng giải thuật dùng vi điều khiển STM32F4 và các module để điều khiển robot hút bụi tự động với một số tính năng đặc biệt, hạn chế tối đa các thao tác của con người, tìm hiểu phương pháp sử dụng các cảm biến để robot di chuyển linh hoạt trong các không gian tránh chướng ngại vật trên quãng đường di chuyển, đạt được hiệu quả cao vừa tối ưu hóa và tiết kiệm.. Một điểm mới được nhấn mạnh là nhóm sẽ xây dựng phần mềm giao diện người dùng để có thế điều khiển robot thông qua đường truyền không dây, tiện lợi hơn trong việc điều khiển cũng như theo dõi hoạt động của robot.*

***Nhóm sinh viên thực hiện***

***NGUYỄN TUẤN***

***LÊ THỊ MỸ DUYÊN***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH  **KHOA ĐIỆN – ĐIỆN TỬ**  **BỘ MÔN: ĐIỀU KHIỂN TỰ ĐỘNG** | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  Độc lập - Tự do - Hạnh phúc | |  | *TP. HCM, ngày….tháng…..năm……..* | |  |

**ĐỀ CƯƠNG CHI TIẾT**

|  |  |
| --- | --- |
| **TÊN ĐỀ TÀI: MÁY HÚT BỤI TỰ ĐỘNG** | |
| **Cán bộ hướng dẫn: KS. Bùi Thanh Huyền** | |
| **Thời gian thực hiện:** | |
| **Sinh viên thực hiện: NGUYỄN TUẤN – 1513831**  **LÊ THỊ MỸ DUYÊN – 1510525** | |
| Nội dung đề tài:   1. Nghiên cứu xây dựng giải thuật dùng vi điều khiển STM32F4 và các module để điều khiển robot hút bụi tự động với một số tính năng đặc biệt, hạn chế tối đa các thao tác của con người. 2. 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. 3. Xây dựng phần mềm giao diện người dùng để có thế điều khiển robot thông qua đường truyền không dây.   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** | |  |  |  | | |
| **Xác nhận của Cán bộ hướng dẫn** | TP. HCM, ngày….tháng …..năm…..  **Sinh viên** |

**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.

**ĐỀ MỤC**

[LỜI CẢM ƠN 1](#_Toc6235829)

[LỜI NÓI ĐẦU 2](#_Toc6235830)

[MỤC LỤC 6](#_Toc6235831)

[DANH MỤC HÌNH VẼ 8](#_Toc6235832)

[DANH MỤC BẢNG 11](#_Toc6235833)

[DANH MỤC VIẾT TẮT 12](#_Toc6235834)

# MỤC LỤC

[CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN ĐỀ TÀI 1](#_Toc6474112)

[1.1. Đặt vấn đề 2](#_Toc6474113)

[1.2. Giới thiệu một số robot hút bụi ngày nay 3](#_Toc6474114)

[1.3. Phạm vi đề tài và phương pháp thực hiện 7](#_Toc6474115)

[1.4. Sơ lược nội dung luận văn 9](#_Toc6474116)

[CHƯƠNG 2: TÌM HIỂU MỘT SỐ PHƯƠNG PHÁP VẬN HÀNH DI CHUYỂN 10](#_Toc6474117)

[2.1. Phương pháp di chuyển ZigZag 11](#_Toc6474118)

[2.2. Phương pháp di chuyển xoắn ốc 11](#_Toc6474119)

[2.3. Phương pháp di chuyển spot 11](#_Toc6474120)

[CHƯƠNG 3: GIỚI THIỆU MỘT SỐ MODULE SỬ DỤNG TRONG ROBOT 12](#_Toc6474121)

[3.1. Vi điều khiển trung tâm STM32F407 13](#_Toc6474122)

[3.2. Phát hiện vật cản, vùng chênh lệch độ cao tránh rơi rớt 18](#_Toc6474123)

[**3.2.1** **Module cảm biến siêu âm SRF-04** 18](#_Toc6474124)

[**3.2.2** **Module cảm biến hồng ngoại** 21](#_Toc6474125)

[3.3. Động cơ và module điều khiển tốc độ 22](#_Toc6474126)

[**3.3.1** **Động cơ encoder** 22](#_Toc6474127)

[**3.3.2** **Mạch lái động cơ L298** 23](#_Toc6474128)

[3.4. Thời gian thực – RTC 25](#_Toc6474129)

[3.5. Module kết nối không dây truyền dữ liệu UART 27](#_Toc6474130)

[CHƯƠNG 4: THIẾT KẾ MÔ HÌNH PHẦN CƠ KHÍ 29](#_Toc6474131)

[4.1 Giới thiệu phần mềm Soliwork 30](#_Toc6474132)

[4.2 Giới thiệu mô hình Robot 31](#_Toc6474133)

[4.3 Thiết kế phần thân Robot 31](#_Toc6474134)

[4.4 Thiết kế phần đế Robot 35](#_Toc6474135)

[4.5 Thiết kế hộp bụi 36](#_Toc6474136)

[4.6 Kết quả thiết kế 37](#_Toc6474137)

[CHƯƠNG 5: LẬP TRÌNH ROBOT 38](#_Toc6474138)

[5.1 Phương pháp điều chế PWM 39](#_Toc6474139)

[**5.1.1** **Giới thiệu thuật toán** 39](#_Toc6474140)

[**5.1.2** **Giới thiệu phương pháp** 40](#_Toc6474141)

[**5.1.3** **Ứng dụng vào robot** 40](#_Toc6474142)

[**5.1.4** **Sơ đồ thuật toán** 40](#_Toc6474143)

[5.2 Thuật toán PID 40](#_Toc6474144)

[**5.2.1** **Giới thiệu thuật toán** 40](#_Toc6474145)

[**5.2.2** **Giải thuật PID** 40](#_Toc6474146)

[5.3 Thuật toán tránh vật cản, vùng chênh lệch độ cao 44](#_Toc6474147)

[5.4 Phương pháp truyền UART qua kết nối không dây 44](#_Toc6474148)

[5.5 Đọc giá trị ADC xác định giá trị pin 44](#_Toc6474149)

[5.6 Lập trình đọc giá trị thời gian thực 44](#_Toc6474150)

[5.7 Thuật toán di chuyển của robot 44](#_Toc6474151)

[5.8 Sơ đồ ghép nối các module 45](#_Toc6474152)

[CHƯƠNG 6: THIẾT KẾ GIAO DIỆN NGƯỜI BẰNG NGÔN NGỮ C# 46](#_Toc6474153)

[6.1 Đặt vấn đề 47](#_Toc6474154)

[6.2 Giới thiệu giao diện người dùng (UI) 47](#_Toc6474155)

[6.3 Giới thiệu chương trình Visual Studio 49](#_Toc6474156)

[6.4 Các tính năng chính của giao diện 52](#_Toc6474157)

[**6.4.1** **Kết nối thiết bị** 53](#_Toc6474158)

[**6.4.2** **Cài đặt cho robot** 56](#_Toc6474160)

[**6.4.3** **Theo dõi trạng thái của robot** 60](#_Toc6474161)

[**6.4.4** **Ghi chép quá trình làm việc** 63](#_Toc6474162)

[CHƯƠNG 7: THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ 67](#_Toc6474163)

[7.1 Thông số kĩ thuật Robot hút bụi 68](#_Toc6474164)

[7.2 Thực nghiệm khả năng di chuyển và tránh vật cản 68](#_Toc6474165)

[7.3 Nhận xét kết quả thực nghiệm 68](#_Toc6474166)

[CHƯƠNG 8: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 69](#_Toc6474167)

[8.1 Kết luận đề tài 70](#_Toc6474168)

[8.2 Hướng phát triển 70](#_Toc6474169)

[TÀI LIỆU THAM KHẢO 71](#_Toc6474170)

# 

# DANH MỤC HÌNH VẼ

[*Hình 1.2. 1: Một số sản phẩm Botvac* 3](#_Toc6077937)

[*Hình 1.2. 2: Robot hút bụi Neato* 4](#_Toc6077938)

[*Hình 1.2. 3: Robot hút bụi Deetbot* 6](#_Toc6077939)

[*Hình 3.1 1: Sơ đồ biễu diễn xung PWM* 15](#_Toc6078192)

[*Hình 3.1 2* 18](#_Toc6078193)

[*Hình 3.1 3* 18](#_Toc6078194)

[*Hình 3.2. 1: Cảm biến siêu âm* 19](#_Toc6078197)

[*Hình 3.2. 2* 20](#_Toc6078198)

[*Hình 3.2. 3* 21](#_Toc6078199)

[*Hình 3.2. 4* 21](#_Toc6078200)

[*Hình 3.3. 1:* 24](#_Toc6078141)

[*Hình 3.3. 2* 25](#_Toc6078142)

[*Hình 3.3. 3* 26](#_Toc6078143)

[*Hình 3.3. 4* 27](file:///C:\Users\admin\Desktop\report\Report-thesis\ReportThesis.docx#_Toc6078144)

[*Hình 4.1. 1: Phần mềm Soliwork* 30](#_Toc6079438)

[*Hình 4.3. 1: Phần thân của Robot* 32](#_Toc6079446)

[*Hình 4.3. 2 Phần thân ở góc nhìn khác* 33](#_Toc6079447)

[*Hình 4.3. 3: Cách bố trí cảm biến trên thân* 34](#_Toc6079448)

[*Hình 4.4. 1: Phần đế của Robot* 35](#_Toc6079455)

[*Hình 4.4. 2: Phần đế ở góc nhìn khác* 36](#_Toc6079456)

[*Hình 4.5. 1: Phần thiết kế chứa bụi của Robot* 37](#_Toc6079460)

[*Hình 5.2. 1 Sơ đồ liên hệ* 40](#_Toc6079466)

[*Hình 5.2. 2: Sơ đồ bộ điều khiển PID* 41](#_Toc6079467)

[*Hình 5.2. 3: Đáp ứng của khâu tỷ lệ* 42](#_Toc6079468)

[*Hình 5.2. 4: Đáp ứng của khâu tích phân* 42](#_Toc6079469)

[*Hình 5.2. 5: Đáp ứng khâu vi phân* 43](#_Toc6079470)

[*Hình 6.2. 1 Chơng trình Visual Studio 2017* 47](file:///C:\Users\admin\Desktop\report\Report-thesis\ReportThesis.docx#_Toc6079475)

[*Hình 6.2. 2: Giao diện của chương trình* 48](#_Toc6079476)

# DANH MỤC BẢNG

[*Bảng 3.1 1: Các chế độ timer* 14](#_Toc6079522)

[*Bảng 5.2. 1: Bảng điều chỉnh PID* 44](#_Toc6079529)

# DANH MỤC VIẾT TẮT

**A**

**ADC** Analog Digital Converter

**AGV** Autonomous Guided Vehicles

**API** Application Programming Interface **AUV** Autonomous Underwater Vehicles **AUX** AUXiliary

**C**

**CCP** Capture/Compare/PWM

**CL** Code Laboratories

**CMOS** Complementary Metal – Oxide – Semiconductor

**CNC** Computerized Numerical Control

**CPU** Central Processing Unit

**E**

**ECCP** Enhanced Capture/Compare/PWM

**EEPROM** Electrically Erasable Programmable Read – Only Memory

**EUSART** Enhanced Universal Synchronous Asynchronous Receiver Transmitter

**F**

**FLANN** Fast Library for Approximate Nearest Neighbors

**G**

**GPIO** General Purpose Input Ouput

**H**

**HDD** Hard Disk Drive

**I**

**ICD** In – Circuit Debugger

**ICSP** In – Circuit Serial Programming

**IR** Infrared

**J**

**JNA** Java Native Access

**JNI** Java Native Interface

**M**

**MFC** Microsoft Foundation Class Library **MSSP** Master Synchronous Serial Port **MUX** Multiplexer

**N**

**NI** Natural Interaction

**NUI** Natural User Interface

**Q**

**QVGA** Quarter Video Graphics Array

**R**

**RAM** Random Access Memory **RANSAC** RANdom SAmple Consensus **RGB** Red, Green, Blue

**S**

**SDK** Software Development Kit

**SSP** Synchronous Serial Port

**SXGA** Super eXtended Graphics Array

**T**

**TOF** Time Of Flight

**TTL** Transistor – Transistor Logic

**U**

**UAV** Unmanned Arial Vehicles

**USART** Universal Synchronous Asynchronous Receiver Transmitter

**USB** Universal Serial Bus

**V**

##### VGA VTK

Video Graphics Array

Visualization Too

## CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN ĐỀ TÀI

* 1. **Đặt vấn đề**
  2. **Giới thiệu một số Robot hút bụi hiện nay**
  3. **Phạm vi đề tài và phương pháp thực hiện**
  4. **Sơ lược nội dung luận văn**

Nội dung chính

### 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, 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 khôn 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, giải trí,… 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, nghiên cứu.

Nhiệm vụ của đề tài là khảo sát các tính năng của robot hút bụi trên thị trường, tham khảo các tài lệu 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.

### Giới thiệu một số robot hút bụi ngày nay

[Robot hút bụi thông minh](https://meta.vn/robot-hut-bui-thong-minh-c1230) 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 1.2. 1: Một số sản phẩm Botvac*

* ***Giới thiệu***

Neato – thương hiệu đến từ Mỹ - là một trong những nhà sản xuất có tên tuổi nhất trên thị trường robot hút bụi thông minh hiện nay. Robot hút bụi của Neato có nhiều sản phẩm thuộc dòng Botvac như Botvac Connected, Botvac D5 Connected, Botvac D3 Connected và Botvac D85,... Trong đó sản phẩm cao cấp nhất là phải kể đến là dòng Botvac Connected với rất nhiều tính năng thông minh, cải tiến kỹ thuật mang lại sự ưu Việt khi sử dụng. Nó đã dành được rất nhiều các giải thưởng danh giá khi được giới thiệu. Điển hình như giải Robot hút bụi thông minh tốt nhất năm 2016 do CNET bình chọn.

**

*Hình 1.2. 2: Robot hút bụi Neato*

* ***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. Thiết kế hình chữ D cũng mang lại sự vượt trội về chổi quét chính và hộp chứa bụi. Chổi quét chính và hộp chứa bụi của Botvac Connected nói riêng và các sản phẩm Botvac nói chung có kích thước lớn gần gấp đôi so với các sản phẩm thông thường. Nhờ chổi quét lớn hơn này, sản phẩm có thể làm sạch nhanh hơn và diện tích làm sạch cũng lớn hơn.

* ***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. Bản đồ này sẽ bao gồm chu vi hoạt động, các vật cản có trong căn phòng, các vị trí cửa ra vào, cầu thang… 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 ổ sạc sau khi hoàn thành công việc của mình chứ không chạy lòng vòng cho đến khi hết pin như một số loại khác. Sở dĩ có được điều này là bởi hệ thống điều hướng của tháp laser giúp thiết lập lộ trình cho robot hoạt động.

* ***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. Phía bên trong, hộp chứa bụi được trang bị bộ lọc hiệu suất cao, có thể lọc được các hạt bụi với kích cỡ 0,3 micron giúp người dùng tránh được các tác nhân gây hại cho sức khỏe và dị ứng.

* ***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. Và ngược lại bạn cũng sẽ nhận được thông tin về căn phòng, khu vực đã được làm sạch,… từ robot được gửi về điện thoại.

* **Dòng sản phẩm Deebot của Ecovacs**



*Hình 1.2. 3: Robot hút bụi Deetbot*

* ***Giới thiệu***

Deebot Ozmo 930, Deebot Ozmo 600 hay Deebot R98 là 3 sản phẩm nổi bật của Ecovacs hiện đang được phân phối tại Việt Nam. Trên 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, căn phòng của bạn sẽ luôn sạch bong sáng bóng.

* ***Thiết kế***

Giống như phần lớn các robot hút bụi hiện nay tại Việt Nam, thì robot của Ecovacs cũng mang dạng hình tròn. Chứng tỏ 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. Hai cây chổi quét ven có mục đích giúp robot có thể thu được nhiều bụi hơn ở các khu vực góc tường.

Trên Deebot có một cảm biến giúp máy tự động nhận diện khu vực nào là sàn, khu vực nào trải thảm để thay đổi chế độ làm việc giúp mang lại hiệu quả làm sạch cao nhất.

* ***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.

### 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 ZICZAC. 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 chiếc máy không ngã cầu thang, các bậc lên xuống trong quá trình làm việc.
* Xây dựng phần mềm giao diện người dùng để có thế 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 robot hiện có mặt trên thị trường. Do vậy, tính khả thi của đề tài này sẽ rất là cao, và nhóm chúng em mong muốn sản phẩm sẽ dần hoàn thiện hơn và sớm đi 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, công tắc hành trình và encoder . Tất cả các cảm biến này sẽ gửi tín hiệu về vi vi xử lý chính STM32f407 để xử lý và đưa ra  các thuật tóan, giải thuật điều khiển nhằm tránh vật cản, những nơi chênh lệch độ cao và định hướng đường đi. Tích hợp thêm các thành phần khác như module thời gian thực (DS 3231), mạch chia áp, module wifi (ESP 8266),… để máy hoạt động ở chế độ thời gian thực, truyền nhận không dây và kiểm soát dung lượng pin. Bên cạnh đó, xây dựng phần mềm giao diện người dùng điều khiển Robot qua đường truyền không dây với 2 chế độ bằng tay và tự động, giúp người dùng thuận tiện hơn trong việc điều khiển và theo dõi sản phẩm.

Cảm biến từ trường dùng để định hướng cho robot,  xác định hướng đi chính xác và thông minh nhất. Encoder dùng để phát hiện robot dừng do bị trượt bánh hoặc bị mắc do các vật cản.

### Sơ lược nội dung luận văn

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 Soliwork 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 của UI đi kèm với sản phẩm Robot hụt 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

* 1. **Phương pháp di chuyển ZigZag**
  2. **Phương pháp di chuyển xoắn ốc**
  3. **Phương pháp di chuyển spot**

Nội dung chính

### Phương pháp di chuyển ZigZag

### Phương pháp di chuyển xoắn ốc

### Phương pháp di chuyển spot

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

**3.1 Vi điều khiển trung tâm**

**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 Thời gian thực – RTC**

**3.5 Module kết nối không dây truyền dữ liệu UART**

Nội dung chính

### 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 Cortex™-M4F CPU với FPU, tần số lên đến 168 MH
* *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
* Cấp nguồn 3.3V riêng
* **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 -> Px15). Mỗi chân IO có thể được lập trình một cách riêng rẽ , 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 16bits). Timer của STM32F407 là timer 16 bits 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:

UEV = TIM\_CLK/((PSC + 1)\*(ARR + 1)\*(RCR + 1))

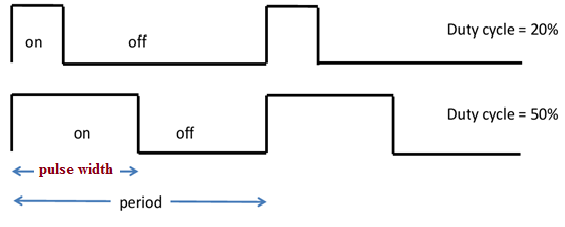
STM32F407 có tổng cộng 14 timer. Trong đó mỗi timer sẽ có chức năng và độ phân giải khác nhau. Có thể tóm tắt theo bảng sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **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 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 1: Sơ đồ biễu diễn xung PWM*

Duty cycle = pulse width x 100/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 (Cicular).
* 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) chuyển 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 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.

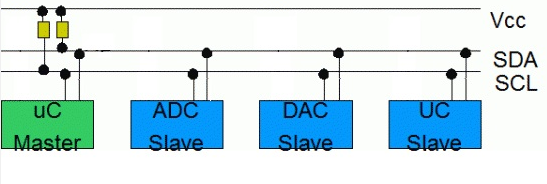
Để sử dụng USART phải cấu hình các chi tiết sau:

* Tốc độ baud: tốc độ cài đặt phải nằm tròn khoảng từ 642bits/s đến 2625 Mbits/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 ngắt 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 Intergrated Circuit)**

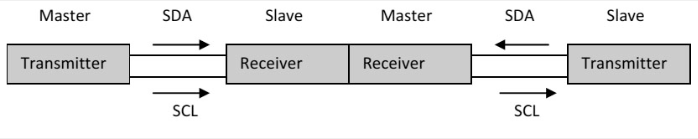
I2C là một chuẩn giao tiếp nối tiếp 2 dây (Serial Data - SDA và Serial Clock - SCL) được sử dụng làm bus giao tiếp ngoại vi cho rất nhiều loại IC khác nhau như các loại Vi điều khiển 8051, PIC, AVR, ARM... Trong đó, SDA là đường truyền dữ liệu 2 hướng, còn SCL là đường truyền xung đồng hồ để đồng bộ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

**

*Hình 3.1 2*

Hình trên cho thấy sẽ có nhiều thiếu 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 3.1 3*

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 tớ đến chủ.

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 100kbps.
* 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.

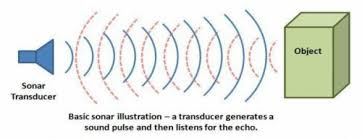
### Phát hiện vật cản, vùng chênh lệch độ cao tránh rơi rớt

#### **Module cảm biến siêu âm SRF-04**

**

*Hình 3.2. 1: 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 (chất lỏng như nước, nước thải, chất lỏng dạng kết dính…, 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 3.2. 2*

**Thông số kỹ thuật:**

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

**Giao tiếp:**

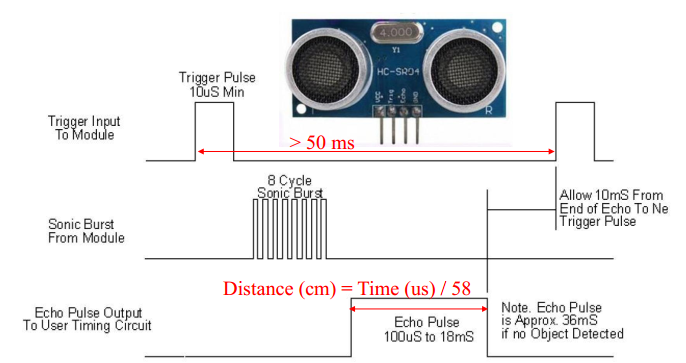
* VCC: điện áp cấp từ 3.3V đến 5V
* GND: GND ngoài
* Trig: chân Trigger để nhận tín hiệu điều khiển
* Echo: chân Echo để phản hồi về xung có độ rộng tương ứng với khoảng cách vật

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. Chiều rộng của xung sẽ bằng với thời gian sóng siêu âm được phát từ cảm biển và quay trở lại. Quá trình được lặp lại với chu kì phải lớn hơn 50ms để đảm bảo độ chính xác.

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

Distance = (cm)

với T là độ rộng xung Echo (us)

**

*Hình 3.2. 3*

#### **Module cảm biến hồng ngoại**

**

*Hình 3.2. 4*

Cảm biến 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 đầu cho 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....

**Thông số kỹ thuật:**

* Bộ so sánh sử dụng LM393, làm việc ổn định
* Điện áp làm việc: 3.3V - 5V DC.
* Khi bật nguồn, đèn báo nguồn sáng.
* Lỗ vít 3 mm, dễ dàng cố định, lắp đặt.
* Kích thước: 3.2cm \* 1.4cm

**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 ngoài
* OUT: đầu ra kỹ thuật số (0 và 1) với mức 0 khi phát hiện có vật cản.

### Động cơ và module điều khiển tốc độ

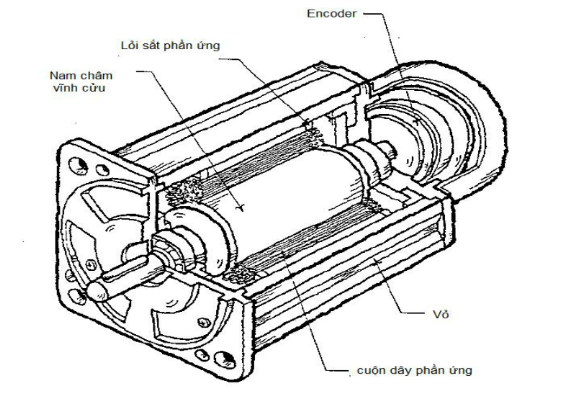
#### **Động cơ encoder**

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 nhiếu 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 bầt 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 3.3. 1:*

#### **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 7805 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.

**Thông số kỹ thuật:**

* IC chính: L298N (tích hợp 2 mạch cầu H)
* Điện áp điều khiển: 5V – 12V
* Công suất tối đa mỗi cầu H: 25W
* Dòng tối đa mỗi cầu H: 2A
* Điện áp của tín hiệu điều khiển: +5V - +7V
* Dòng điện của tín hiệu điều khiển: 0 – 36mA
* Nhiệt độ bảo quản: -250C -> 1300C

**

*Hình 3.3. 2*

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 sẽ A 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, giả sử ta cấp một xung PWM có duty cycle = 50% vào chân INA và giữ chân INB ở mức 0 thì động cơ A sẽ quay với tốc độ 50%. Tương tự với các chân còn lại.

### 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 3.3. 3*

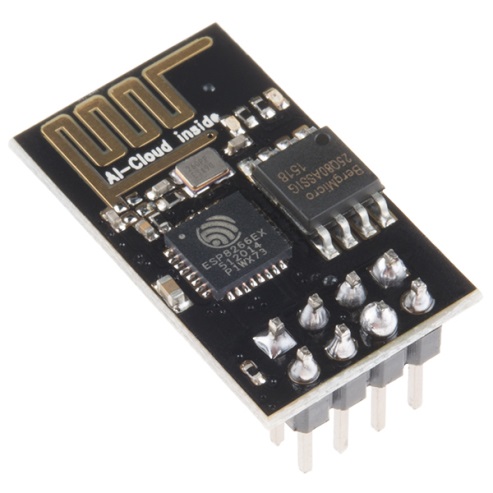
Thời gian trong IC được giữ ở dạng: giờ, phút, giây, ngày, thứ, tháng, năm. Các tháng có ít hơn 31 ngày sẽ tự động được điều chỉnh, các năm Nhuận cũng được chỉnh đúng số ngày. Thời gian có thể hoạt động ở chế độ 24h hoặc 12h AM/PM. IC còn có chức năng báo động, có thể cài đặt 2 thời gian báo và lịch, 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. Có tín hiệu Reset xuất ra cho mạch ngoài, MCU khi nguồn điện phục hồi trạng thái. Ngoài ra trong IC còn có sẵn cảm biến nhiệt độ, có độ chính xác là ± 3°C.

**Thông số kĩ thuật:**

* Size: dài 38mm, rộng 22mm, cao 14mm
* Khối lượng: 8g
* Điện thế hoạt động: 3.3 - 5.5V
* Độ chính xác: trong tầm nhiệt 400C, với độ chênh lệch 2 phần triệu thì có sai số khoảng 1 phút.
* Thông tin thời gian: giờ, phút, giây, ngày, thứ, tháng, năm, đến 2100.
* Cảm biến nhiệt trên IC có độ chính xác ± 3 ℃
* I2C bus có tốc độ tối đa 400Khz
* Kèm thêm pin sạc được CR2032
* Kèm thêm memory IC AT24C32 (32k bits)

### Module kết nối không dây



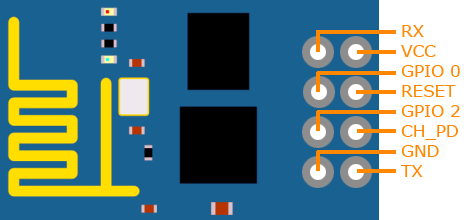
Mạch thu phát Wifi ESP8266 Uart ESP-01 sử dụng IC Wifi SoC ESP8266 của hãng Espressif, được sử dụng để kết nối với vi điều khiển thực hiện chức năng truyền nhận dữ liệu qua Wifi,. Mạch có thiết kế nhỏ gọn, sử dụng giao tiếp UART với bộ thư viện và code mẫu rất nhiều từ cộng đồng. ESP-01 được sử dụng trong các ứng dụng IoT và điều khiển thiết bị qua Wifi,...với thiết kế dễ sử dụng và đặc biệt là có thể sử dụng trực tiếp trình biên dịch của Arduino để lập trình và nạp code, điều này khiến việc sử dụng và lập trình các ứng dụng trên ESP8266 trở nên rất đơn giản.

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 có khả năng tự nhận Driver trên tất cả các hệ điều hành Window là Linux, đây là phiên bản nâng cấp từ các phiên bản sử dụng IC nạp giá rẻ CH340).

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) — dùng để nhận 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
* UTXD (TX) — dùng để truyền tín hiệu trong giao tiếp UART với vi điều khiển



**Thông số kỹ thuật:**

* Điện áp sử dụng: 3.3VDC
* Điện áp giao tiếp: 3.3VDC
* Dòng tiêu thụ: Max 320mA (do đó nên sử dụng [module cấp nguồn riêng](http://hshop.vn/products/mach-giamop-3-3v-ams1117) cho mạch).
* Hỗ trợ chuẩn 802.11 b/g/n.
* Wi-Fi 2.4 GHz, hỗ trợ các chuẩn bảo mật như: OPEN, WEP, WPA\_PSK, WPA2\_PSK, WPA\_WPA2\_PSK.
* 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.
* Kích thước: 24.8 x 14.3mm

## CHƯƠNG 4: THIẾT KẾ MÔ HÌNH PHẦN CƠ KHÍ

**4.1 Giới thiệu phần mềm Soliwork**

**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**

**4.6 Kết quả thiết kế**

Nội dung chính

### Giới thiệu phần mềm Soliwork

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 tác giả 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. 1: Phần mềm Soliwork*

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 lắp 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.

### 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 gió của robot. Sau đó, lớp bụi này được hút bởi cụm quạt hút ở phía sau.

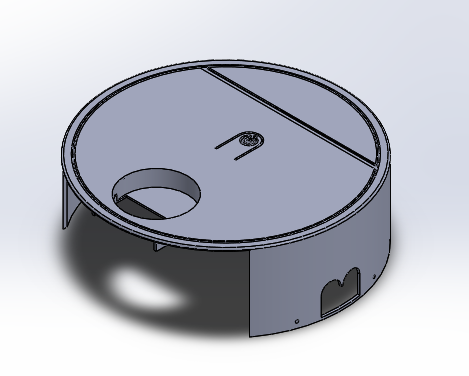
Bố trí vị trí cảm biến trên robot: robot sử dụng các cảm biến hồng ngoại để tránh chống rơi và cảm biến siêu âm để phát hiện vật cản. Từ mục đích trên, các cảm biến được thiết kế với các vị trí như sau:

* Có 3 cảm biến hồng ngoại chống rơi được đặt hai bên và phía trước, ngay mép của thân robot hút bụi hướng xuống phía dưới để xác định mặt sàn như là một 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, tác giả đề xuất cấu tạo của robot hút bụi như sau:

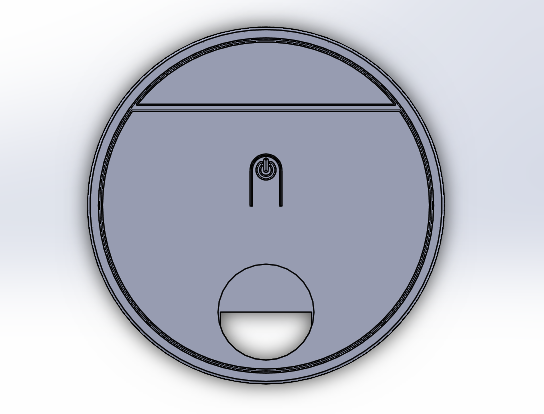
### Thiết kế phần thân Robot

**

*Hình 4.3. 1: Phần thân của Robot*

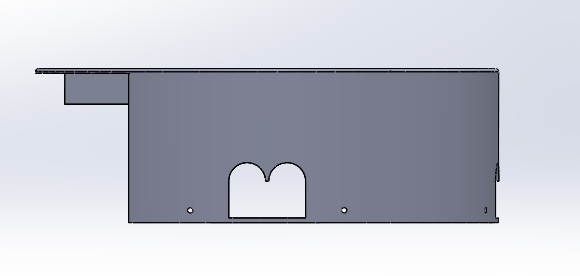
Phần trên robot mang dạng tròn, đường kính 30 cm, cao 10cm, thành dày 3mm và rỗng ruột, có chức năng như là nắp của robot. Kích thước của robot được tính toán phù hợp sao cho vừa có sức chứa đầy đủ các thành phần cấu tạo vừa không quá cồng kềnh và đặc biệt phải có độ bền, tránh hư hại trong quá trình 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 bề 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. 2 Phần thân ở góc nhìn khác*

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.



Lỗ bắt vít

Khung siêu âm

Không gian để hộp chứa bụi

*Hình 4.3. 3: 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 dạng 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 đạc 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. 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.

### 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.

Động cơ

Cảm biến

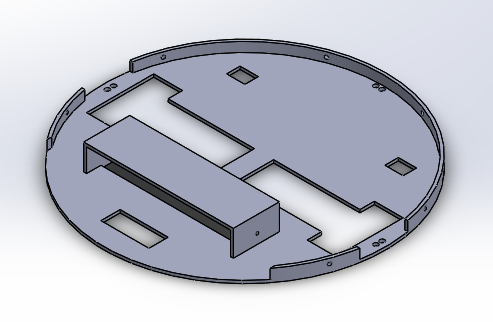
hồng ngoại

Bánh đa hướng

Chổi quét

Bánh xe

Miệng hút



*Hình 4.4. 1: Phần đế của Robot*

Có thể thấy trên phần đế các 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. Đặc biêt, bộ phần miệng hút được thiết kế mở rộng 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.

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 chừa 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.

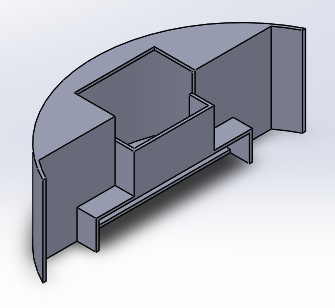
**

*Hình 4.4. 2: Phần đế ở góc nhìn khác*

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 mĩ cho mặt ngoài của robot.

### 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 mĩ.

**

Lỗ đặt quạt

Lỗ hút bụi

Phần rìa ngoài

*Hình 4.5. 1: 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 khối giúp cho bụi được giữ kĩ 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.

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).

### Kết quả thiết kế

## CHƯƠNG 5: LẬP TRÌNH ROBOT

**5.1 Phương pháp điều chế PWM**

**5.2 Thuật toán PID**

**5.3 Thuật toán tránh vật cản, vùng chênh lệch độ cao**

**5.5 Phương pháp truyền UART qua kết nối không dây**

**5.5 Đọc giá trị ADC xác định giá trị pin**

**5.6 Lập trình đọc giá trị thời gian thực**

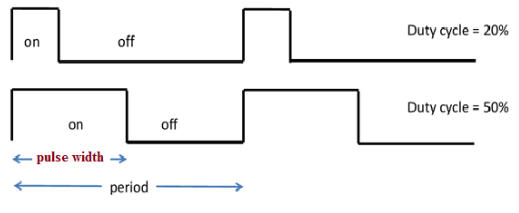
**5.7 Thuật toán di chuyển của robot**

**5.8 Sơ đồ ghép nối các module**

Nội dung chính

### Phương pháp điều chế PWM

#### **Giới thiệu thuật toán**

Điều chế PWM (Pulse Width Modulation) hay còn gọi là 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. 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).

Như trên hình đã rất rõ : Duty cycle = pulse width x 100/period

– Duty cycle là tỷ lệ phần trăm mức cao.

– Period là chu kỳ xung.

– Pulse width là giá trị mức cao so với period.

Dựa trên nguyên lý bên trên mà trong STM32 hay các loại vi điều khiển khác điều hỗ trợ bộ tạo độ rộng xung tự động mà bạn 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. PWM thuộc khối timer

#### **Giới thiệu phương pháp**

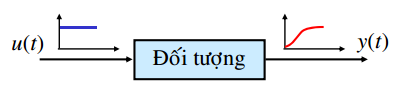
#### **Ứng dụng vào robot**

#### **Sơ đồ thuật toán**

### Thuật toán PID

#### **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 KP, KI và KD sao cho đáp ứng của hệ thống về gần nhất với giá trị đặt.

**

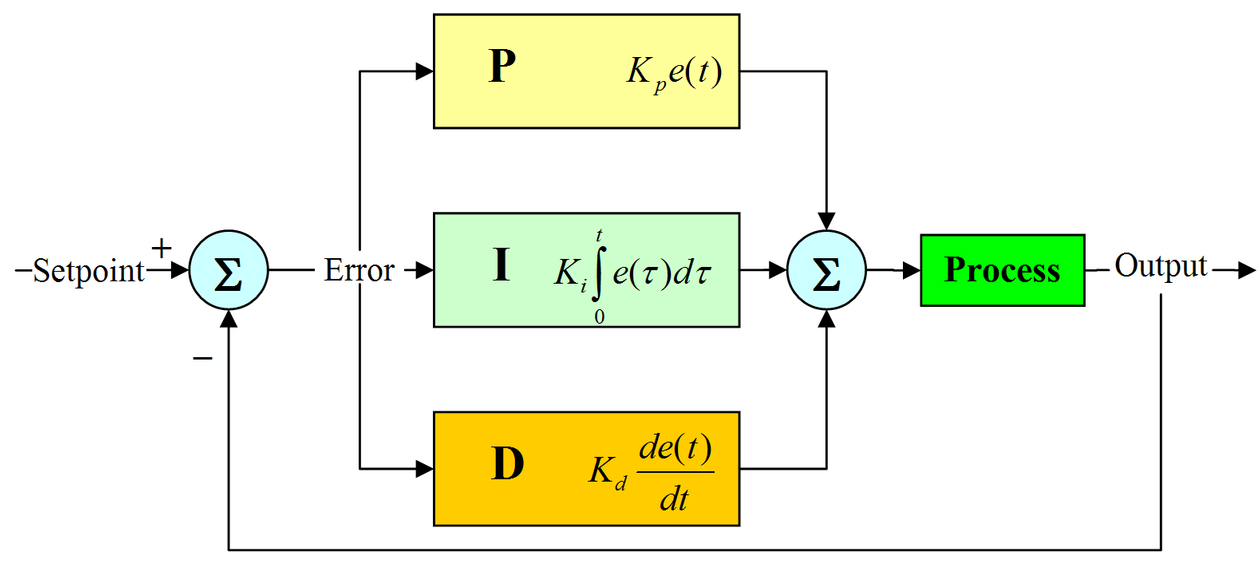
*Hình 5.2. 1 Sơ đồ liên hệ*

#### **Giải thuật PID**

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 phận* 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 5.2. 2: Sơ đồ bộ điều khiển PID*

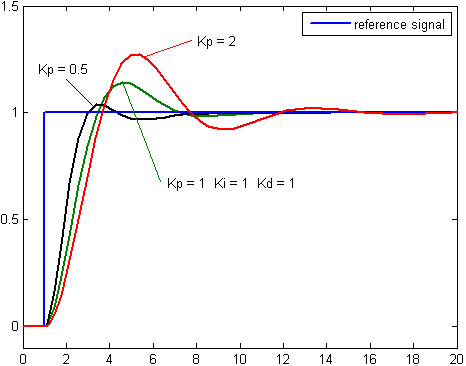
Hàm truyền bộ điều khiển PID



Hay 

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, đặc biệt.

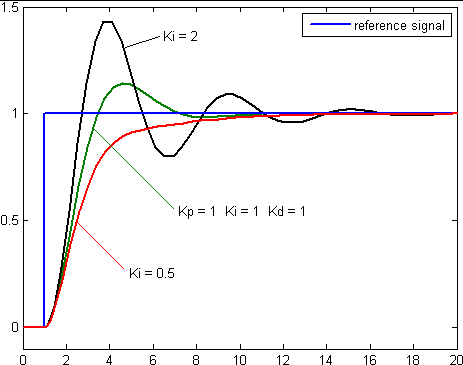
* **Khâu tỷ lệ**

**

*Hình 5.2. 3: 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 KP, được gọi là độ lợi tỷ lệ.

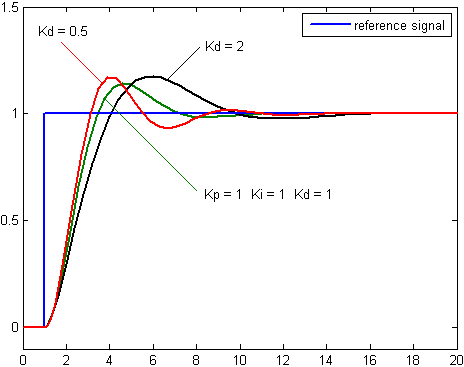
* **Khâu tích phân**

**

*Hình 5.2. 4: Đáp ứng của khâu tích phân*

Phân phối của khâu tích phân tỷ lệ với cả biên độ sai số lẫn quảng 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.

* **Khâu vi phân**

**

*Hình 5.2. 5: Đáp ứng khâu vi phân*

Tốc độ thay đổi của sai số quá trình được tính toán bằng cách xác định đọ 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 KD. 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 bọ điều khiển. Do đó, điệu khiển vi phân được sử dụng để làm giảm biên độ vọt lố 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 tình trở nên không ổn định nếu nhiễu và độ lợi vi phân đủ lớn.

***Tóm lại:***

* Độ lợi tỷ lệ (KP): 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 (KI): giá trị càng lớn kéo theo sai số ổn định bị khử càng nhanh. Đổi lại là vọt lố càng lớn.
* Độ lợi vi phân (KD): giá trị càng lớn càng giảm độ vọt lố, 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 lố | Thời gian quá độ | Sai số xác lập |
| Kp | Tăng | Tăng | Thay đổi nhỏ | Tăng |
| Ki | Giảm | Tăng | Tăng | Triệt tiêu |
| Kd | Thay đổi nhỏ | Giảm | Giảm | Thay đổi nhỏ |

*Bảng 5.2. 1: Bảng điều chỉnh PID*

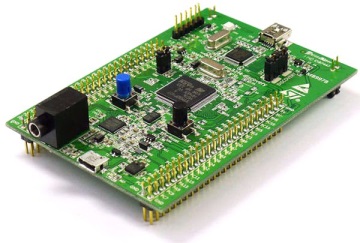
* Mỗi hệ số sẽ tác động đến chất lượng hệ thống theo một chiều hướng nhất định. Và các hệ số này có mối liên hệ tác động lẫn nhau. Do đó ta cần phải điểu chỉnh các hệ số một cách thích hợp với từng hệ thống để đạt được chất lượng đáp ứng tốt nhất.

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

### Phương pháp truyền UART qua kết nối không dây

Để đáp ứng cho ứng dụng điều khiển từ xa, nhóm quyết định sử dụng thêm module Wifi ESP8266. 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:



UART

UART

WIFI

MCU

Server

Client

UI

Theo như sơ đồ trên thì 1 module ESP sẽ được kết nối với vi điều khiển 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:

|  |  |  |
| --- | --- | --- |
| ***Thông số*** | ***Kit STM32F4*** | ***Giao diện người dùng*** |
| 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 |

**Giải thuật Client**:

**Đ**

**S**

**S**

**Đ**

**Đ**

**S**

**Đ**

**S**

Read chuỗi line gửi từ Server

Truyền chuỗi line vừa nhận cho MCU

Begin

Có kết nối

Khởi tạo kết nối

In “Connect..”

Delay 500ms

Kết nối đúng client

In “Connection fail”

Delay 1s

Read chuỗi data\_in gửi từ MCU

Delay 100ms

Truyền chuỗi data\_in cho Server

Có thông tin từ MCU

Có thông tin từ Server

**Giải thuật Server:**

i = i+1

Read chuỗi data\_in gửi từ PC

Truyền chuỗi data\_in cho Client

Số client (i) < = CLIENT MAX

i = i+1

**Đ**

**Đ**

**S**

Read chuỗi line gửi từ client

Truyền chuỗi line vừa nhận cho MCU

Begin

Có client mới kết nối

Khởi tạo client

In “New client: i

Số client (i) < = CLIENT MAX

Có thông tin từ PC

Client available

Số client (i) < = CLIENT MAX

i = i+1

Khởi tạo kết nối

**Đ**

**Đ**

**S**

**S**

**Đ**

**Đ**

**S**

**S**

### Đọc giá trị ADC xác định giá trị pin

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 .

Trong phần lập trình, module ADC được sử dụng là bộ ADC 1, channel 1 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 0 từ 4096 đến 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 theo 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.

Giá trị Pin 12V (Analog)

Giá trị Pin 3V (Analog)

Giá trị Pin

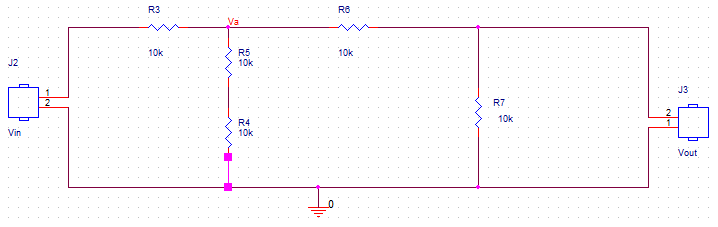
(Digital)

Mạch chia áp

Module

ADC

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 không may có sự cố xảy ra (vì điện áp 3V sẽ an toàn cho vi điều khiển hơn điện áp 12V)



Trong đó: 





* 

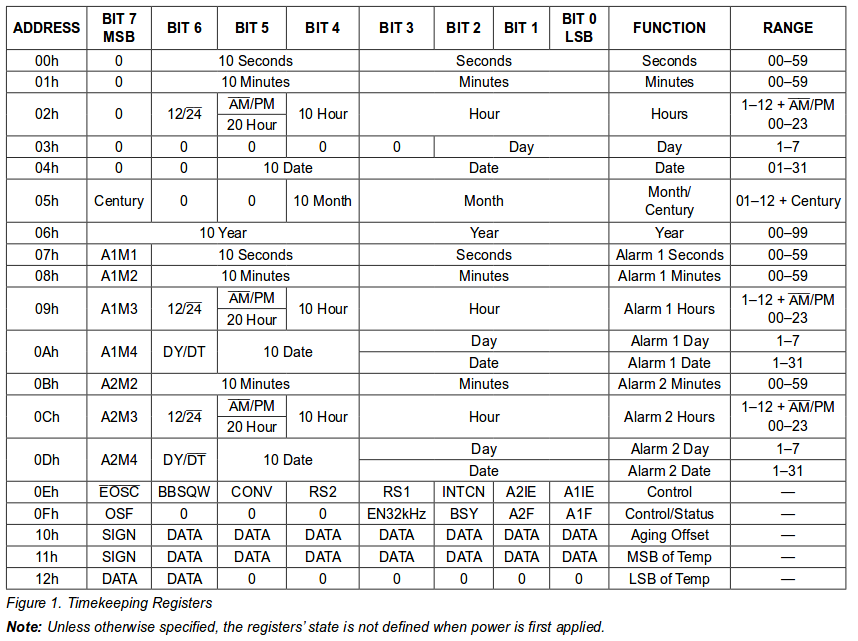
### Lập trình đọc giá trị thời gian thực

Vì yêu cầu của một robot dân dụng nói chung mà 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.

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



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 (ngày, tháng, năm, thứ, giờ, phút, giây) ở dạng số BCD. Dó 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.

Trước khi nhận thông tin cần gửi 1 byte có giá trị tương ứng là địa chỉ thanh ghi dữ liệu đầu tiên trong chuỗi thanh ghi lưu trữ dữ liệu cần nhận. Trong trường hợp này là thanh ghi có địa chỉ 0x00.

Xác định giá trị thời gian hiện tại

Chuyển đổi về dạng số BCD

Lưu các thông tin thời gian (BCD) vào Buffer

Gửi thông tin đi bằng I2C đến địa chỉ Slave (0x68)

Lưu giá trị 0x00 (Hex) vào Buffer[0]

Gửi thông tin trong Buffer đi bằng I2C đến địa chỉ slave (0x68)

Nhận thông tin thời gian(BCD) từ Slave gửi về Master lưu vào Buffer

Chuyển đổi thông tin từ BCD về dạng số thập phân và lưu vào các biến tương ứng

Module RTC được cài đặt giá trị thời gian mặc định là 0h:0m:0s day0, date0, month 0, year 0. 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 nhằm duy trì cho IC hoạt động trong khoảng thời gian không được sử dụng.

### Thuật toán di chuyển của robot

### Sơ đồ ghép nối các module

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

**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**

**6.4.1 Kết nối thiết bị**

**6.4.2 Cài đặt cho robot**

**6.4.3 Theo dõi trạng thái của robot**

**6.4.4 Ghi chép quá trình làm việc**

Nội dung chính

### 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 có thể 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 đinh 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.

### 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 quả cao trong quá trình quản lí ..v.v.

**

**

*Hình 6.3. 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 gia 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.

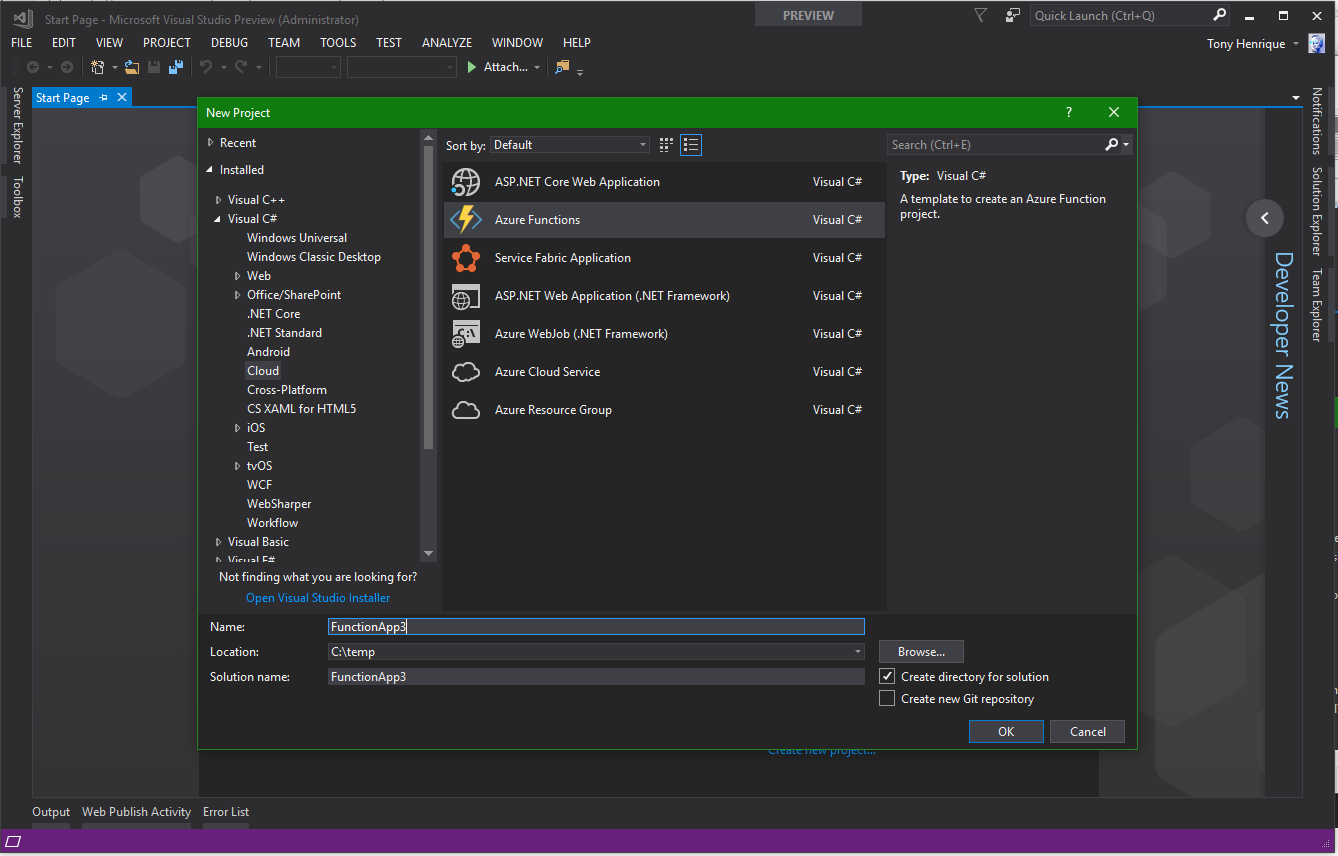
### Giới thiệu chương trình Visual Studio



*Hình 6.2. 1 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 các 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 6.2. 2: 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 các sử dụng [IntelliSense](https://msdn.microsoft.com/en-us/library/hcw1s69b.aspx) 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 tức 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

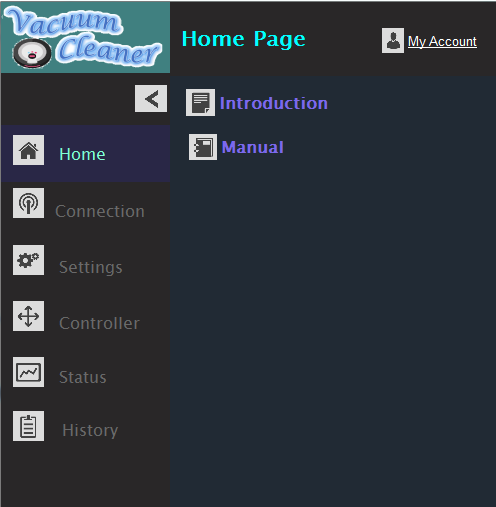
Tính năng này cũng giống như Windows Forms Designer có công dụng hỗ trợ kéo và thả ẩn dụ. Sử dụng tương tác giữa người và máy tính nhắm mục tiêu vào [Windows Presentation Foundation](https://vi.wikipedia.org/wiki/Windows_Presentation_Foundation).

* 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.

### Các tính năng chính của giao diện



*Hình 6.4. 1 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, buồng chứa bụi 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.

#### **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 chứng 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 dung 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;

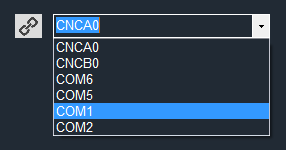
Đây là thư viện có sẵn trong Visual Studio được dùng cho các chương trình cân thiết lập sử dụng đến cổng COM.

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

Khai báo lệnhSerialPort 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.

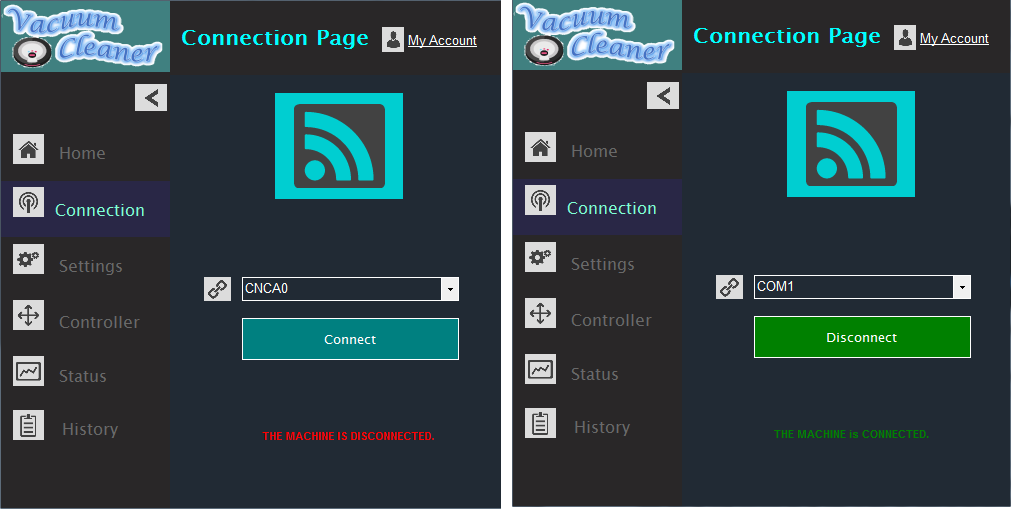
* *Kết nối và hủy kết nối cổng COM*:

Sau khi đã chọn được cổng COM kết nối với module truyên 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 hổ trợ sẵn.

UART.PortName = cbxCom.Text; //UART là một lớp Serial Port (cổng COM) đã được khai báo ở trên.

UART.Open();//mở cổngCOM **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, sãn sàng cho việc truyền nhận dữ liệu.

**

*Hình 6.4. 2: 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 lí 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)

{}

}

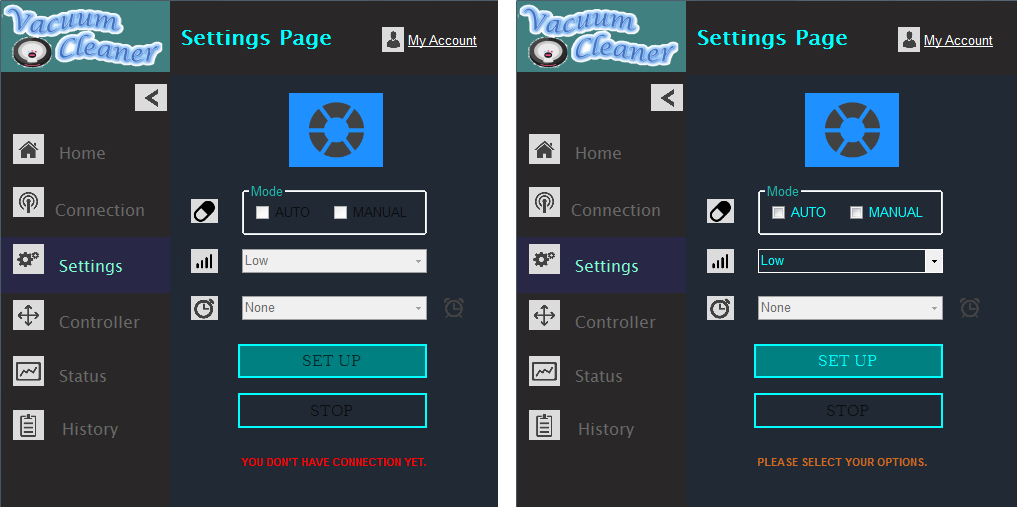
Đối với việc gửi dữ liệu qua cổng COM, thư viện trong Visual Studio cũng có hổ trợ với cách thực hiện đơn giản hơn bằng câu lệnh:

UART.Write(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.

#### **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 nhưng 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:

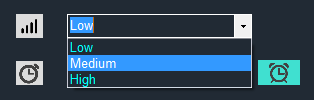


*Hình 6.4. 3: Giao diện cài đặt cho Robot*

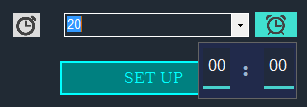
Tương tự trên giao diện **Connection**, khi thiết bị chưa giao tiếp với PC, ta sẽ nhận được nhãn dán với dòng chữ đỏ báo chưa được kết nối và kèm theo là các vùng để cài đặt sẽ bị vô hiệu hóa không thể nhập cũng như tác động trực tiếp.Người dùng chỉ có thể bắt đầu cài đặt các thông số và chế độ cho chiếc 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 cải 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.Write(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

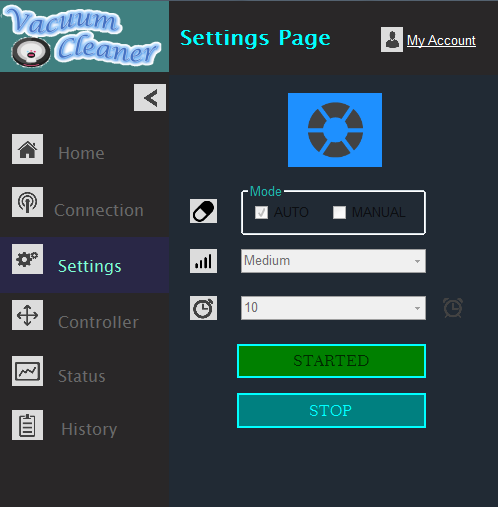
Time = string.Format("{0:00}", cbxTime.Text);

* *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.*

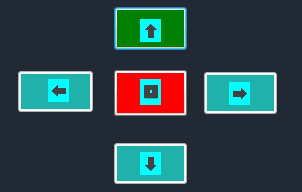
Riêng với *Time*, *startHour & StartMin* sẽ được mặc định với tham số “00” ban đầu hoặc khi người dùng không cài đặt gì ở thông số này. Việc lập trình như vậy để đảm bảo rằng độ dài của chuỗi sẽ là 9 kí tự không đổi trong quá trình truyền nhận, giúp việc phân tích và quản lí chuối trong vi điều khiển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

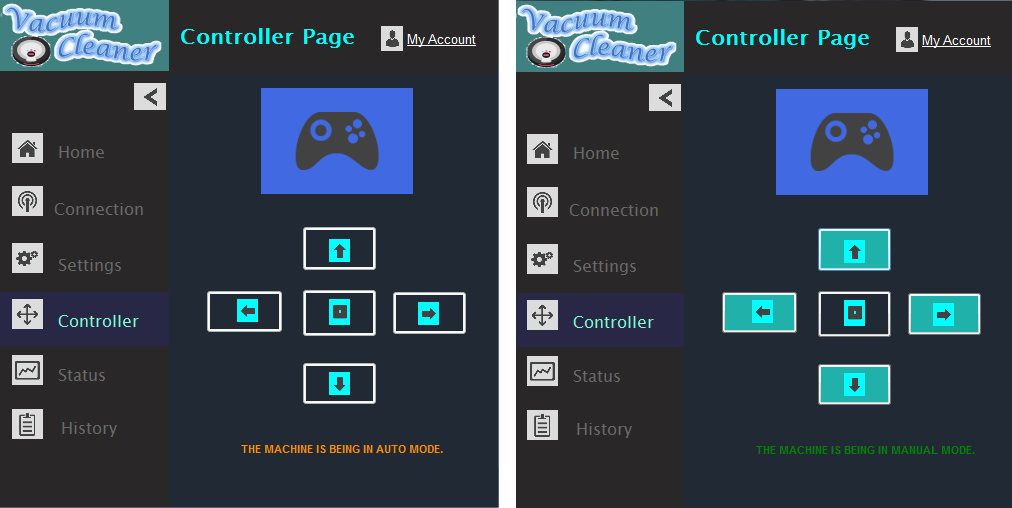
Khi PC gửi thành công chuỗi dữ liệu qua Serial Port, , nút **Set Up** sẽ chuyển qua màu xanh lá với chữ **Started**. giao diện cài đặt của ứng dụng sẽ bị vô hiệu hóa, người dùng không còn thực hiện được những tác vụ thiết lập thông số cho đến khi nhấn **Stop** và trở về với trạng thái ban đầu chuẩn bị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



*Hình 6.4. 4: 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** trênUIvới cách bố trí gồm 5 nút nhấn, trong đó 4 nút là *trái, phải, tiến, lùi* và một nút *Stop* để dừng Robot.

Khi một nút nào đó trong bộ điều khiển được nhấn, tương ứng với hướng điều khiển mong muốn của người sử dụng, nó sẽ chuyển sang màu xanh lá, đồng thời nút Stop cũng chuyển sang màu đỏ và tất cả 4 nút sẽ bị vô hiệu hóa. Người dùng chỉ có thể điều chỉnh hướng khác cho đến khi nhấn nút *Stop*.

****

*Hình 6.4. 5: 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”, “d” với thứ tự các nút *trái, lùi, phải, tiến, dừng.* 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 gừi xuống Robot hút bụi khi lập trình cho phần cài đặt của ứng dụng.

#### **Theo dõi trạng thái của robot**

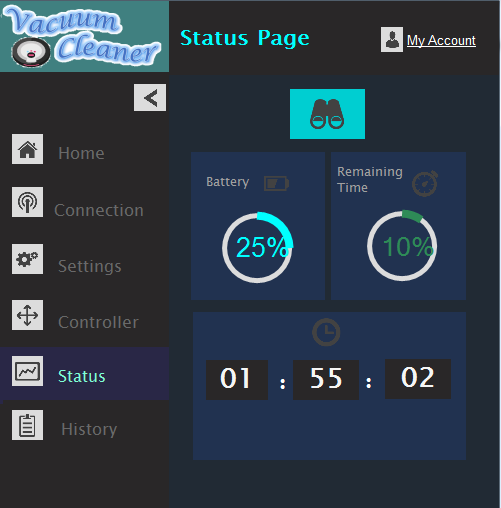
Ngày nay ngoài việc có thể cài đặt chế độ làm việc cho máy, người dùng còn có nhu cầu cao hơn, đó chính là theo dõi và giám sát trạng thái hoạt động của Robot. Đối với những loại máy thông thường, chúng ta không thể xác định được giá trị pin hiện tại của sản phẩm cũng như không gian của khoang chứa bụi đã đầy hay chư. Việc không thể biết trước máy sẽ dừng hoạt động thời điểm nào hoặc khi nào khoang đầy cũng có thể dẫn đến một số điều không thuận tiện khi sử dụng như pin không còn vào những lúc cần thiết hay giảm hiệu quả khi khoan chứa đã đầy,…Chính vì vậy, nhóm quyết định sẽ thiết kế một giao diện hiển thị trạng thái của Robot trên ứng dụng để ngưòi dùng có thể theo dõi giá trị pin, thời gian thực của chiêc 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 lí 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 đồi 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 và hiển thị lên giao diện để có thể quan sát một cách trực quan và thuận tiện.

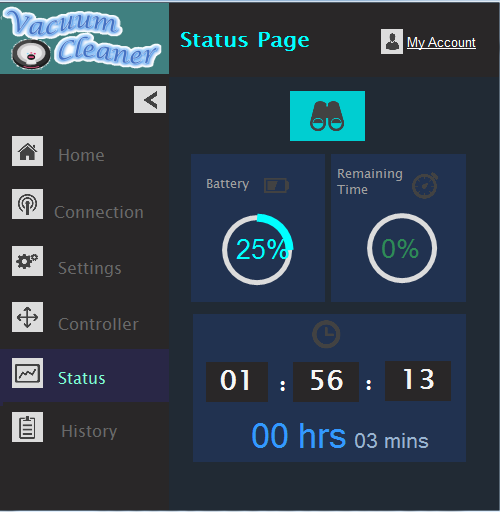
Sau đây là một số kết quả hiển thị của giao diện **Status** ứng với một số trường hợp cài đặt của người dùng:

* 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 6.4. 6: 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 6.4. 7: Giao diện trạng thái trong TH2*

*Lưu ý:* phần remaining time sẽ hiển thị giá trị 0 khi Robot hoạt ở chế độ Manual họăc 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.

#### **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

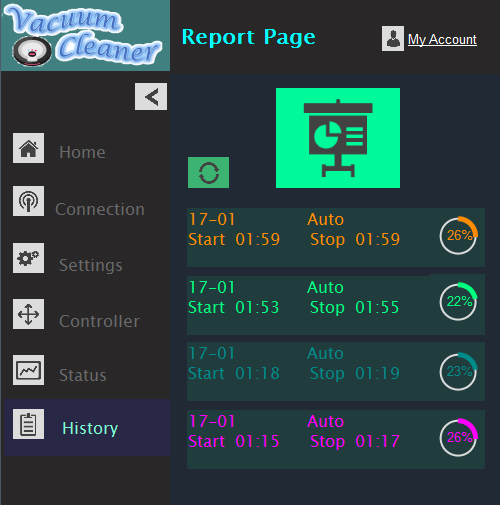
Lưu ý: Sau mỗi lần thực hiện lưu dữ liệu, chúng ta phải dùng dòng lệnh write.Close(); đóng các đối tượng writer hiện tại và giải phóng mọi nguồn lực chiếm dụng đang liên kết với Writer. Có như vậy, ta mới có thể sử dụng lớp Reader để đọc dữ liệu từ tập tin văn bản ở phần sau. Tương tự với khi sử dụng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

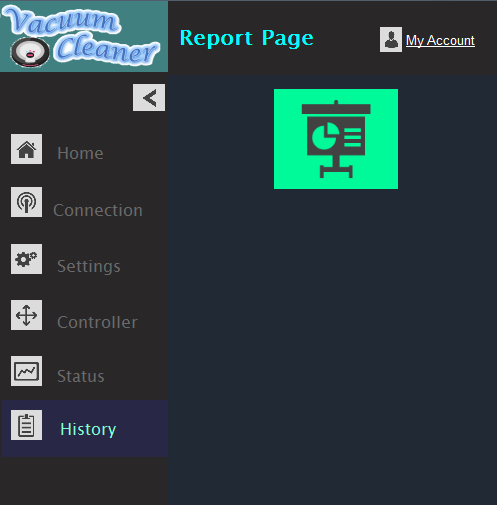
Phần giao diện sau khi lập trình sẽ có kết quả hiển thị như sau:



**Nút Reset**

*Hình 6.4. 8: 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 6.4. 9 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 khiểm 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

**7.1 Thông số kĩ thuật Robot hút bụi**

**7.2 Thực nghiệm khả năng di chuyển và tránh vật cản**

**7.3 Nhận xét kết quả thực nghiệm**

Nội dung chính

### Thông số kĩ thuật Robot hút bụi

### Thực nghiệm khả năng di chuyển và tránh vật cản

### Nhận xét kết quả thực nghiệm

(đánh giá tính hiệu quả, giá thành, điện áp)

## 

## CHƯƠNG 8: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

**8.1 Kết luận đề tài**

**8.2 Hướng phát triển**

Nội dung chính

### Kết luận đề tài

### Hướng phát triển

## TÀI LIỆU THAM KHẢO