**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**🙤🙦**

**LUẬN VĂN NGÀNH**

**TRUYỀN THÔNG VÀ MẠNG MÁY TÍNH**



**ĐỀ TÀI**

**XE CÁNH TAY ROBOT**

**ĐIỀU KHIỂN TỪ XA**

**Phân hệ 2: Lập trình điều khiển cánh tay robot**

**GVHD:PGS. TS. Đỗ Thanh Nghị**

***Sinh viên thực hiện*:**

**Nguyễn Văn Vinh**

**B1611116**

**Khóa 42**

**Cần Thơ, 07/2020**

# NHẬN XÉT GIẢNG VIÊN HƯỚNG DẪN

Cần thơ, ngày… tháng… năm 2020

Giáo viên hướng dẫn

**PGS.TS. Đỗ Thanh Nghị**

# NHẬN XÉT GIẢNG VIÊN PHẢN BIỆN

Cần thơ, ngày… tháng… năm 2020

Giáo viên phản biện

# LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cám ơn đến Quý Thầy Cô và Bạn bè đã giúp đỡ chúng em trong suốt thời gian thực hiện Luận văn tốt nghiệp này.

Xin chân thành cảm ơn Quý Thầy Cô Trường Đại học Cần Thơ nói chung và Thầy Cô Khoa Công nghệ thông tin & Truyền thông nói riêng đã tận tình dạy bảo, truyền đạt những kiến thức quý báo, những kinh nghiệm và những kỹ năng chuyên môn để chúng em vận dụng thực hiện vào đề tài lần này.

Đặc biệt, em xin gửi lời cám ơn sâu sắc đến Thầy Đỗ Thanh Nghị đã tận tình giúp đỡ, định hướng cách tư duy và cách làm việc khoa học. Nhờ vào sự quan tâm giúp đỡ, hướng dẫn rất tận tình, tâm huyết của Thầy trong quá trình hướng dẫn và tìm hiểu em đã có thêm cho mình nhiều khiến thức bổ ích, tinh thần học tập hiệu quả, là hành trang để em có thể vững bước sau này.

Do chưa có nhiều kinh nghiệm trong lĩnh vực sáng chế cũng như hạn chế về kiến thức, trong bài luận văn chắc chắn sẽ không tránh khỏi những thiếu sót. Rất mong nhậng được sự nhận xét, ý kiến đóng góp từ phía Thầy Cô để chúng em học thêm được nhiều kinh nghiệm.

Lời cuối cùng, em xin kính chúc Quý Thầy Cô nhiều sức khỏe, hạnh phúc và thành công trên con đường sự nghiệp của mình.

Một lần nữa em xin chân thành cám ơn.

Cần thơ, ngày… tháng… năm 2020

Sinh viên thực hiện

Nguyễn Văn Vinh

# MỤC LỤC

[NHẬN XÉT GIẢNG VIÊN HƯỚNG DẪN 2](#_Toc61781118)

[NHẬN XÉT GIẢNG VIÊN PHẢN BIỆN 3](#_Toc61781119)

[LỜI CẢM ƠN 4](#_Toc61781120)

[MỤC LỤC 5](#_Toc61781121)

[MỤC LỤC HÌNH ẢNH 7](#_Toc61781122)

[MỤC LỤC BẢNG 8](#_Toc61781123)

[TÓM TẮT ĐỀ TÀI 9](#_Toc61781124)

[ABSTRACT 9](#_Toc61781125)

[PHẦN GIỚI THIỆU 1](#_Toc61781126)

[1. Đặt vấn đề 1](#_Toc61781127)

[2. Lịch sử giải quyết vấn đề 1](#_Toc61781128)

[3. Mục tiêu đề tài 1](#_Toc61781129)

[4. Đối tượng và phạm vi nghiên cứu 1](#_Toc61781130)

[5. Nội dung nghiên cứu 2](#_Toc61781131)

[6. Bố cục luận văn 2](#_Toc61781132)

[PHẦN NỘI DUNG 3](#_Toc61781133)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 3](#_Toc61781134)

[1.1 Arduino IDE (Integrated Development Environment) 3](#_Toc61781135)

[1.2 Hệ điều hành Android 6](#_Toc61781136)

[1.3 Inventor 8](#_Toc61781137)

[1.4 Các gói cài đặt và thư viện 10](#_Toc61781138)

[1.5 Phần cứng thiết bị 11](#_Toc61781139)

[1.6.1 Arduino Mega 2560 11](#_Toc61781140)

[1.6.2 Module Esp32 Camera 13](#_Toc61781141)

[1.6.3 Module Bluetooth HC-05 17](#_Toc61781142)

[1.6.4 Động cơ bước Nema 17 size 42 20](#_Toc61781143)

[1.6.5 Động cơ Servo SG90 21](#_Toc61781144)

[1.6.6 Mạch điều khiển động cơ bước DRV8825 22](#_Toc61781145)

[1.6.7 Động cơ RC Servo MG996 24](#_Toc61781146)

[CHƯƠNG 2: MÔ TẢ BÀI TOÁN 26](#_Toc61781147)

[1.1 Mô tả tổng quan 26](#_Toc61781148)

[1.1.1 Bối cảnh sản phẩm 26](#_Toc61781149)

[1.1.2 Các tính năng của mô hình 26](#_Toc61781150)

[1.1.3 Thiết kế sơ đồ khối 26](#_Toc61781151)

[1.1.4 Phần cứng yêu cầu các thiết bị 27](#_Toc61781152)

[1.1.5 Phần mềm yêu cầu 27](#_Toc61781153)

[1.2 Yêu cầu giao tiếp 27](#_Toc61781154)

[CHƯƠNG 3: CÀI ĐẶT GIẢI PHÁP 28](#_Toc61781155)

[1.1 Cài đặt 28](#_Toc61781156)

[1.2 Lưu đồ giải thuật 28](#_Toc61781157)

[1.1.1 Thuật toán phần mềm điều khiển 28](#_Toc61781158)

[1.3 Lập trình cánh tay và app MIT inventor 30](#_Toc61781159)

[1.1.1 Code app: Blocks 30](#_Toc61781160)

[1.1.2 Code điều khiển cánh tay trên app MIT: 32](#_Toc61781161)

[1.1.3 Điều khiển các khớp cánh tay 34](#_Toc61781162)

[CHƯƠNG 4: ĐÁNH GIÁ VÀ KIỂM THỬ 36](#_Toc61781163)

[1.1 Mục tiêu kiểm thử 36](#_Toc61781164)

[1.2 Kịch bản kiểm thử 36](#_Toc61781165)

[1.3 Kết quả kiểm thử 36](#_Toc61781166)

[PHẦN KẾT LUẬN 37](#_Toc61781167)

[1. Kết quả đạt được 37](#_Toc61781168)

[2. Khó khăn 37](#_Toc61781169)

[3. Hướng phát triển 37](#_Toc61781170)

[TÀI LIỆU THAM KHẢO 53](#_Toc61781171)

# MỤC LỤC HÌNH ẢNH

[Hình 1.1. Tải Arduno IDE 3](#_Toc60938285)

[Hình 1. 2. Giải nén file Download 4](#_Toc60938286)

[Hình 1. 3. Arduino IDE 5](#_Toc60938287)

[Hình 1. 4. Thông báo trong Arduino IDE 6](#_Toc60938288)

[Hình 1. 5. MIT App Inventor 8](#_Toc60938289)

[Hình 1. 6. . Giao diện quản lý project 9](#_Toc60938290)

[Hình 1. 7. Giao diện thiết kế (Design) 9](#_Toc60938291)

[Hình 1. 8. Giao diện lập trình (Blocks) 10](#_Toc60938292)

[Hình 1. 9. Arduino Mega 2560 11](#_Toc60938293)

[Hình 1. 10. Mạch điều khiển máy in 3D RAMPS 1.4 12](#_Toc60938294)

[Hình 1. 11. Arduino Shield Mega Proto 12](#_Toc60938295)

[Hình 1. 12. Esp32 Camera 13](#_Toc60938296)

[Hình 1. 13. Kết nối Camera với mạch nạp 17](#_Toc60938297)

[Hình 1. 14. Module HC-05 17](#_Toc60938298)

[Hình 1. 15. Các chân của HC-05 18](#_Toc60938299)

[Hình 1. 16. Kết nối HC-05 với Arduino 19](#_Toc60938300)

[Hình 1. 17. Động cơ bước Size 42 20](#_Toc60938301)

[Hình 1. 18. Sơ đồ kỹ thuật động cơ bước 21](#_Toc60938302)

[Hình 1. 19. Động cơ Servo 22](#_Toc60938303)

[Hình 1. 20. Mạch điều khiển động cơ bước 23](#_Toc60938304)

[Hình 1. 21. Động cơ RC 24](#_Toc60938305)

[Hình 1. 22. Code Bluetooth và lập bước 31](#_Toc60938306)

[Hình 1. 23. Servo 1 và 2 32](#_Toc60938307)

[Hình 1. 24. Servo 3 và 4 32](#_Toc60938308)

[Hình 1. 25. Servo 5 và 6 33](#_Toc60938309)

# MỤC LỤC BẢNG

[Bảng 1. 1. Chức năng icon trên Arduino IDE 5](#_Toc60938466)

[Bảng 1. 2. Thông số kỹ thuật Esp 32 Camera 14](#_Toc60938467)

[Bảng 1. 3. Bảng sử dụng chân Esp32 Camera 16](#_Toc60938468)

# TÓM TẮT ĐỀ TÀI

Công nghiệp 4.0 còn được coi là cuộc cách mạng lần thứ 4 diễn ra trên thế giới, với tốc phát triển nhanh chóng, làm thay đổi toàn bộ các ngành, lĩnh vực và đời sống của chúng ta. Rất nhiều khái niệm đi kèm theo công nghiệp 4.0 như trí tuệ nhân tạo AI, Internet Of Things (IoT), Big Data, Robot, công nghệ 3D,… Đặc biệt, trong ngành công nghiệp ngày này, robot được ứng dụng nhiều và rộng rãi nhằm mục đích mang đến sự tiện ích, tiết kiệm thời gian, công sức cho con người. Vì vậy, với hy vọng xây dựng một hệ thống thông minh phục vụ cho nhu cầu của con người, chúng tôi chọn đề tài **“Xe cánh tay robot điều khiển từ xa”** để xây dựng đề tài luận văn của mình.

Trong nghiên cứu này, việc điều khiển từ xa bằng điện thoại thông minh sử dụng hệ điều hành android điều khiển cánh tay. Mục đích của việc nghiên cứu này là thiết kế mô hình xe sử dụng board mạch Arduino Mega 2560 kết hợp với động cơ. Để điều khiển cánh tay và xe sử dụng App MIT Inventor để lập trình các nút hiển thị trên màn hình điện thoại đồng thời gửi lệnh đến Arduino Mega xử lý để thực thi hành động gắp vật. Xe cánh tay robot điều khiển từ xa sẽ là giải pháp hoàn hảo ngành công nghiệp tự động hóa thay thế sức lao động con người.

# ABSTRACT

Industry 4.0 is 4th revolution talking place in the world. With repid growth changes industries, field, life. Include AI, Internet of Things, Big Data, Robot, 3D printing,... Special, In this industry, Robots are applied crushing and spacious.It brings many utilities.The desire to build smart systems to serve people, We were inclined to save the project “Remote-controlled robotic wing vehicles” .

In this study, smartphone control with android operating system. This tilt purpose is to build model vehicles using Arduino Mega 2560 circuits combined steep motor.To controls need to be programmed on the app MIT inventor. “Remote-controlled robotic wing vehicles” are the perfect solution for the industrial automation.

# PHẦN GIỚI THIỆU

## Đặt vấn đề

Với sự tiến bộ của khoa học kĩ thuật. Đồng nghĩa với việc, người lao động cần tiếp xúc trực tiếp với môi trường làm việc mà họ có thể bị hạn chế tiếp xúc chẳng hạn phải tiếp xúc với nơi dễ cháy nổ hoặc các lò phản ứng hạt nhân, không những vậy mà họ còn phải làm việc liên tục.

Không những thế, người lao động cần phải tăng ca nhiều giờ để đạt năng xuất cho doanh nghiệp, công ty. Tuy nhiên, nếu ta thay thế cánh tay robot thì có thể giải quyết vấn đề năng xuất cũng như chi phí thay cho nguồn lao động sẽ giảm đi.

## Lịch sử giải quyết vấn đề

Hiện nay, Robot là đề tài nghiên cứu của nhiều người và nhiều nước trên thế giới tuy nhiên chỉ chủ yếu tập chung ứng dụng vào công việc cố định đạt được hiệu quả tốt nhất.

Bên cạnh đó, một số sinh viên tại các trường đại học trên cả nước việt nam. Trường đại học bách khoa Hà Nội với đề tài “Cánh tay robot dành cho người khuyết tật” của nhóm 3 sinh viên gồm Đỗ Xuân Vương, Hoàng Thế Nam và Ngô Quan Tài.Trường đại học dân lập Hải Phòng với đề tài “Robot 5 bậc tự do” của sinh viên Nguyễn Thái Phong.

Để đáp ứng dược nhu cầu hạn chế, mọi người đã nghiên cứu phát triển Robot kết hợp nhiều tính năng để có thể thay thế con người vào những nơi nguy hiểm ảnh hưởng đến tính mạng cũng như có thể mang vác các vật cần thiết vào ra những nơi nguy hiểm.

## Mục tiêu đề tài

Mục tiêu của đề tài là thiết kế Robot Cánh tay kết nối với điện thoại di động thông minh chạy trên nền Android để điều khiển từ xa các hoạt động nhằm phục vụ mục đích của người dùng một cách nhanh chóng và tiện lợi hơn.

**Đề tài đặt ra mục tiêu:**

* Điện thoại nhận lệnh từ người dùng thông qua giao diện đồ họa trên màn hình và gửi tín hiệu đển Arduino Mega để xử lý.
* Động cơ thực thi hành động từ tín hiệu yêu cầu của Arduino Mega.
* Camera ghi nhận hình ảnh, đồng thời hiển thị hình ảnh trực tiếp trên điện thoại của người dùng để điều khiển xe và cánh tay đến vị trí cần thiết.

## Đối tượng và phạm vi nghiên cứu

Đề tài mô hình “Xe cánh tay robot điều khiển từ xa ” với phân hệ “điều khiển xe di chuyển và cánh tay robot hoạt động rắp vật”. Đã tiến hành thực hiện và nghiên cứu nội dung sau:

* Tìm hiểu cơ sở lí thuyết liên quan đề tài.
* Tìm hiểu mô hình xe di chuyển.
* Tìm hiểu cách thức hoạt động của cánh tay.
* Tìm hiểu cách lập trình dành cho người không chuyên trên app MIT inventor.
* Tìm hiểu cách thức hoạt động của động cơ bước
* Tìm hiểu cách thức giao tiếp giữa app MIT invenrtor và Arduino MEGA 2560.

## Nội dung nghiên cứu

|  |  |
| --- | --- |
| **Nội dung công việc** | **Người thực hiện** |
| Tìm hiểu và lập trình điều khiển xe và streaming video | Nguyễn Ngọc Khánh Lam |
| Tìm hiều và lập trình điều khiển cánh tay robot | Nguyễn Văn Vinh |
| Xây dựng mô hình, lắp đặt phần cứng | Nguyễn Văn Vinh  Nguyễn Ngọc Khánh Lam |
| Thiết lập kết nối và gửi dữ liệu giữa điện thoại và hệ thống | Nguyễn Văn Vinh  Nguyễn Ngọc Khánh Lam |

## Bố cục luận văn

Bài báo cáo được trình bày như sau:

* ***Phần Giới thiệu:*** Trình bày sơ lược mục tiêu, đối tượng và phạm vi nghiên cứu của đề tài
* ***Phần Nội dung:***

Chương 1: Trình bày cơ sở lý thuyết mô hình xe cánh tay robot gồm: giới thiệu hiểu biết cơ bản về phần mềm và phần cứng như là: Arduino Mega 2560, Module Esp32 Camera, Module HC-05, động cơ bước Nema 17, động cơ Servo SG90, mạch điều khiển động cơ bước Drv8825, Động cơ RC Servo MG996.

Chương 2: Trình bày tổng quan về bài toán giúp người đọc hiểu rõ hơn chức năng của các module trong mô hình, phân tích.

Chương 3: Trình bày thiết kế, chức năng của mô hình, cách thức cài đặt xe, điều khiển xe và streaming video

Chương 4: Mô tả quy trình kiểm thử và đánh giá kết quả của mô hình.

* ***Phần Kết luận:*** Trình bày kết quả đạt được, thuận lợi, khó khăn và hướng phát triển của các module trong mô hình

# PHẦN NỘI DUNG

## CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

### Arduino IDE (Integrated Development Environment)

Để lập trình được cho các board Arduino, các bạn cần phải có một công cụ gọi là Intergrated Development Environment (IDE). Công cụ này được đội ngũ kĩ sư của Arduino phát triển và có thể chạy trên Windows , MAC OS X và Linux

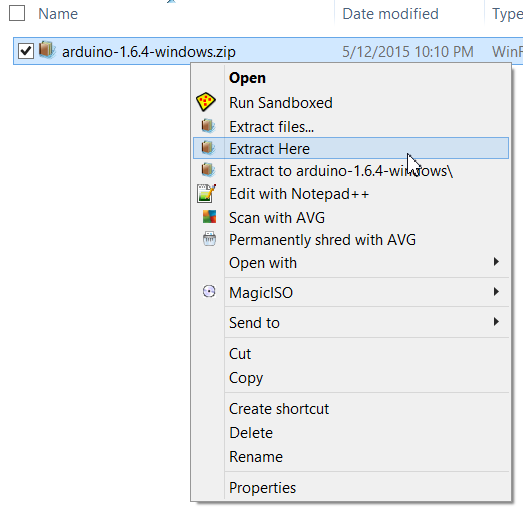
1. **Cài đặt:**

* ***Bước 1:*** Truy cập địa chỉ [http://arduino.cc/en/Main/Software/...](http://arduino.cc/en/Main/Software/) . Đây là nơi lưu trữ cũng như cập nhật các bản IDE của Arduino. Bấm vào mục [Windows ZIP](http://arduino.cc/download_handler.php?f=/arduino-1.6.4-windows.zip) [**file for non admin install**](http://arduino.cc/download_handler.php?f=/arduino-1.6.4-windows.zip) như hình minh họa.



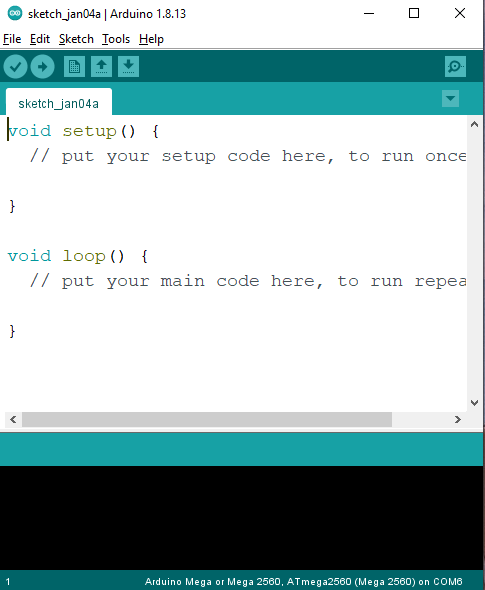
Hình 1.1. ****Tải Arduno IDE****

* ***Bước 2:*** Sau khi download xong, các bạn bấm chuột phải vào file vừa download arduino-1.6.4-windows.zipvà chọn “Extract here” để giải nén.



Hình 1. 2. ****Giải nén file Download****

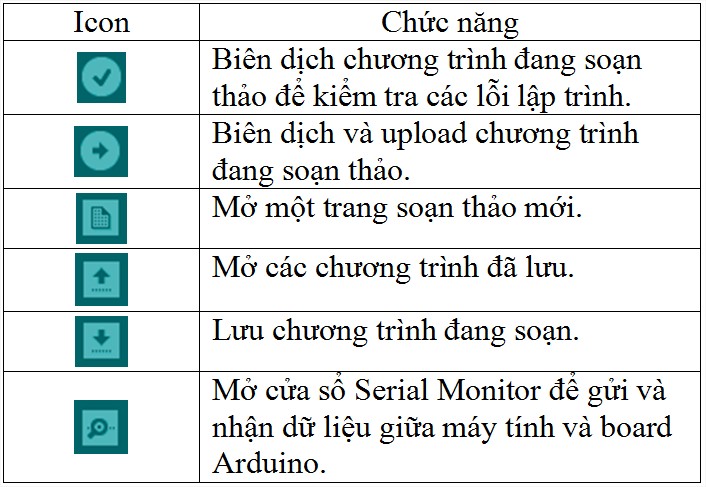
* ***Bước 3:*** Copy thư mục arduino-1.6.4 vừa giải nén đến nơi lưu trữ.
* ***Bước 4:*** Chạy file [](http://k3.arduino.vn/img/2015/05/13/0/1342_81220-1431518023-0-2015-05-13-18h53-25.png) trong thư mục arduino-1.6.4\ để khởi động Arduino IDE



Hình 1. 3. Arduino IDE

1. **Vùng lệnh**

Bao gồm các nút lệnh menu (File, Edit, Sketch, Tools, Help). Phía dưới là các icon cho phép sử dụng nhanh các chức năng thường dùng của IDE được miêu tả như sau:



Bảng 1. 1. Chức năng icon trên Arduino IDE

1. **Vùng thông báo**

Những thông báo từ IDE sẽ được hiển thị tại đây. Để ý rằng góc dưới cùng bên phải hiển thị loại board Arduino và cổng COM được sử dụng. Luôn chú ý tới mục này bởi nếu chọn sai loại board hoặc cổng COM, bạn sẽ không thể upload được code của mình.



Hình 1. 4. ****Thông báo trong Arduino IDE****

### Hệ điều hành Android

* [Android](https://www.dienmayxanh.com/dien-thoai?g=android) là một hệ điều hành có mã nguồn mở dựa trên nền tảng Linux, được phát triển bởi Google (GOOGL), sử dụng chủ yếu trong các thiết bị màn hình cảm ứng, điện thoại di động và máy tính bảng. Nó cho phép người dùng thao tác với các thiết bị một cách trực quan thông qua chuyển động của ngón tay như: chạm, vuốt. Google cũng sử dụng phần mềm Android trên tivi, ô tô và đồng hồ đeo tay
* Sử dụng ARM cho nền tảng phần cứng; các phiên bản sau của hệ điều hành Android hỗ trợ kiến trúc x86 và x86-64. Yêu cầu phần cứng tối thiểu của Android phụ thuộc vào kích thước màn hình của thiết bị, loại và mật độ CPU. Ban đầu, Google yêu cầu bộ xử lý 200MHz, 32MB bộ nhớ và 32 MB RAM.

1. **Ưu nhược điểm:**

* Ưu điểm:
  + Là hệ điều hành có mã nguồn mở nên khả năng tuỳ biến cao, có thể tùy ý chỉnh sửa mà không có sự can thiệp hay cấm cản từ Google.
  + Đa dạng sản phẩm, rất nhiều hãng điện thoại, thiết bị công nghệ đã ưu ái chọn Android cho thiết bị của họ, giá cả thì hợp lý từ bình dân đến cao cấp.
  + Kho ứng dụng Google Play Store đồ sộ.
  + Thân thiện và dễ sử dụng.
  + Khả năng đa nhiệm, chạy cùng lúc nhiều ứng dụng cao.
* Nhược điểm:
  + Dễ nhiễm phần mềm độc hại và virus. Do tính chất mã nguồn mở, nhiều phần mềm không được kiểm soát có chất lượng không tốt hoặc lỗi bảo mật vẫn được sử dụng.
  + Kho ứng dụng quá nhiều dẫn đến khó kiểm soát chất lượng, thiếu các ứng dụng thật sự tốt.
  + Sự phân mảnh lớn. Trong khi một số thiết bị Android xuất sắc đã trình làng như [Galaxy S5](https://www.dienmayxanh.com/dien-thoai/samsung-galaxy-s5-g900), [Galaxy Note 4](https://www.dienmayxanh.com/dien-thoai/samsung-galaxy-note-4), [Xperia Z3](https://www.dienmayxanh.com/dien-thoai/sony-xperia-z3)…, vẫn còn rất nhiều sản phẩm giá rẻ bình thường khác.
  + Cập nhật không tự động với tất cả thiết bị. Khi một phiên bản hệ điều hành mới ra mắt, không phải tất cả sản phẩm đều được cập nhật, thậm chí nếu muốn trải nghiệm bạn thường xuyên phải mua mới thiết bị.

1. **Các phiên bản của Android**

* Android 1.0: Phát hành vào 23/9/2008. Sở hữu bộ ứng dụng Google, bao gồm Gmail, Maps, Lịch và YouTube.
* Android 1.5 (Cupcake): Phát hành 27/4/2009. Ra mắt bàn phím ảo trên màn hình và framewrk cho app các widget bên thứ ba.
* Android 1.6 (Donut): Phát hành 15/9/2009. Giới thiệu HĐH có thể chạy trên các kích thước và độ phân giải màn hình khác nhau; tăng cường hỗ trợ cho mạng CDMA.
* Android 2.0 (Eclair): Phát hành 26/10/2009. Ra mắt tính năng điều hướng bằng giọng nói theo từng chặng, cập nhật thông tin giao thông thời gian thực, kéo mở để thu phóng.
* Android 2.2 (Froyo): Phát hành 20/5/2010. Thêm dock ở cuối màn hình chính và tác vụ thoại, cho phép người dùng chạm vào biểu tượng và nói lệnh. Hỗ trợ Flash cho trình duyệt web.
* Android 2.3 (Gingerbread): Phát hành 6/12/2010. Ra mắt giao diện người dùng đen và xanh lục.
* Android 3.0 đến 3.2 (Honeycomb): Phát hành 22/2/2011. Bản phát hành này dành riêng cho máy tính bảng và có thêm thiết kế không gian ba chiều màu xanh lam.
* Android 4.0 (Ice Cream Sandwich): Phát hành 18/10/2011. Giới thiệu giao diện người dùng thống nhất cho cả máy tính bảng và điện thoại thông minh; với thay đổi đáng chú ý nhất là chủ yếu sử dụng thao tác vuốt để điều hướng.
* Android 4.1 đến 4.3 (Jelly Bean): Phát hành lần lượt vào 9/7/2012, 13/11/2012 và 24/7/2013. Giới thiệu Google Now, một dịch vụ lập kế hoạch trong ngày. Ngoài ra còn có thông báo dạng tương tác và cải tiến hệ thống tìm kiếm bằng giọng nói.
* Android 4.4 (KitKat): Phát hành chính thức 31/10/2013. Ra mắt giao diện người dùng với các màu sáng hơn, cùng với thanh trạng thái trong suốt và các biểu tượng màu trắng.
* Android 5.0 (Lollipop): Phát hành chính thức 12/11/2014. Kết hợp thiết kế giao diện dạng card-based với các yếu tố như thông báo và danh sách Ứng dụng gần đây. Giới thiệu tính năng điều khiển bằng giọng nói thông qua lệnh "OK, Google".
* Android 6.0 (Marshmallow): Được phát hành chính thức hồi tháng 10 năm 2015. Bản phát hành này đánh dấu việc Google thông qua lịch phát hành hàng năm. Ra mắt tính năng quản lý quyền truy cập  ứng dụng chi tiết hơn, hỗ trợ cho đầu đọc vân tay và USB-C.
* Android 7.0 và 7.1 (Nougat): Phát hành lần lượt vào 22/8/2016 và 4/10/2016. Ra mắt chế độ chia đôi màn hình gốc và tính năng nhóm các thông báo theo ứng dụng.
* Android 8.0 và 8.1 (Oreo): Phát hành lần lượt vào 21/8/2017 và 5/12/2017. Các phiên bản này đã ra mắt chế độ picture-inpicture (PIP) nguyên bản - tính năng đáng chú ý nhất cho phép thực hiện đa tác vụ - và tính năng nhắc lại thông báo. Oreo là phiên bản đầu tiên kết hợp Project Treble, một nỗ lực của các OEM nhằm cung cấp các bản cập nhật phần mềm với tiêu chuẩn cao hơn.
* Android 9.0 (Pie): Phát hành 6/ 8/2018. Phiên bản này đã thay thế các nút Back, Home và Overview thành nút Home đa chức năng và nút Back nhỏ hơn. Ra mắt các tính năng quản lý hiệu suất, bao gồm gợi ý trả lời tự động cho tin nhắn và quản lý độ sáng.
* Android 10 (Android Q): Phát hành vào 3/9/2019. Bỏ nút Back để chuyển sang điều hướng hoàn toàn bằng thao tác vuốt. Có thêm Dark theme và Focus mode cho phép người dùng bớt bị phân tâm từ những ứng dụng khác.

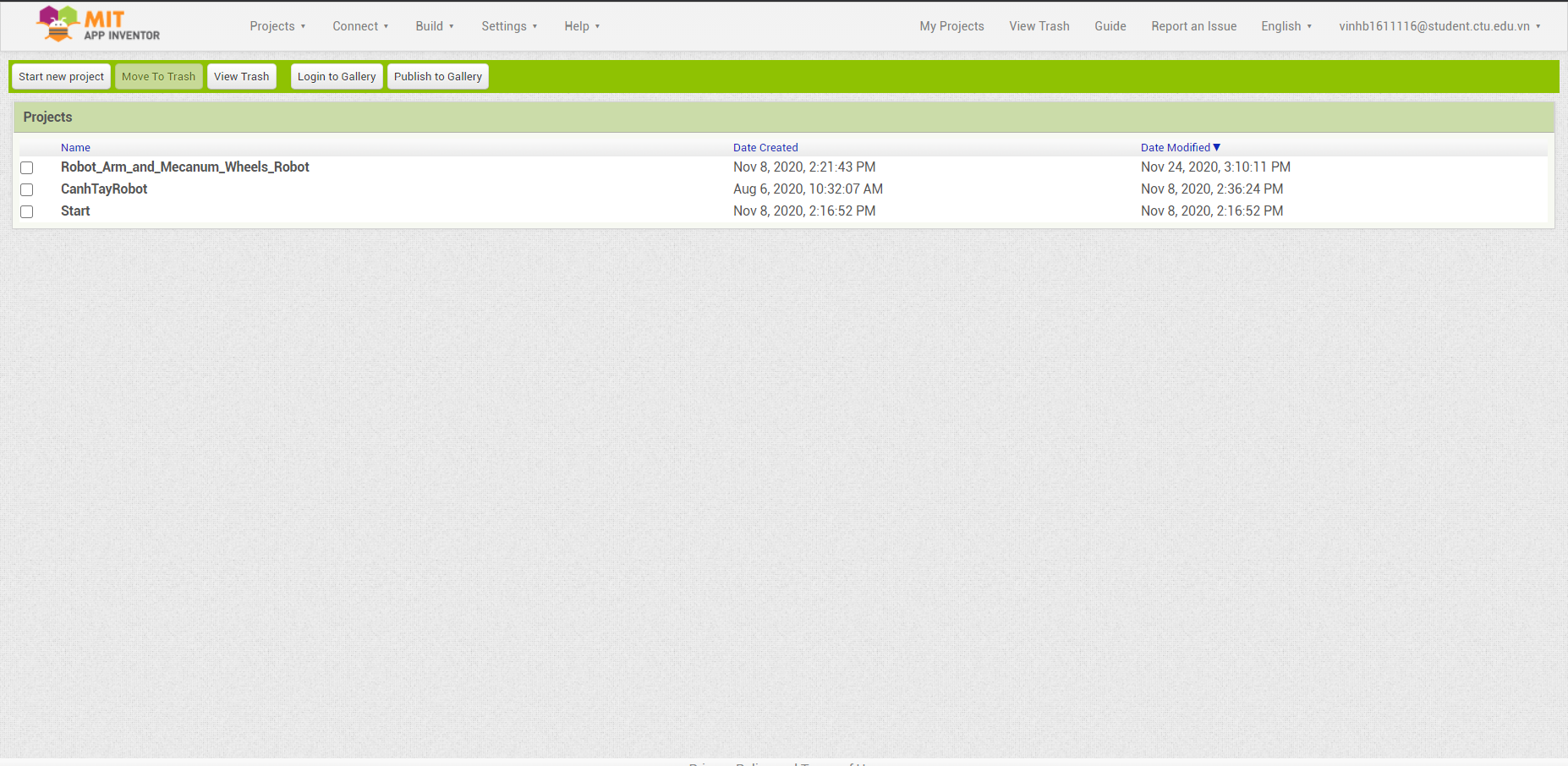
### Inventor

* MIT App Inventor dành cho Android là một ứng dụng web nguồn mở ban đầu được cung cấp bởi Google và hiện tại được duy trì bởi Viện Công nghệ Massachusetts (MIT). Nền tảng cho phép nhà lập trình tạo ra các ứng dụng phần mềm cho hệ điều hành Android (OS). Bằng cách sử dụng giao diện đồ họa, nền tảng cho phép người dùng kéo và thả các khối mã (blocks) để tạo ra các ứng dụng có thể chạy trên thiết bị Android. Đến thời điểm hiện tại 07/2017, phiên bản iOS của nền tảng này đã bắt đầu được đưa vào thử nghiệm bởi Thunkable, là một trong các nhà cung cấp ứng dụng web cho ngôn ngữ này.

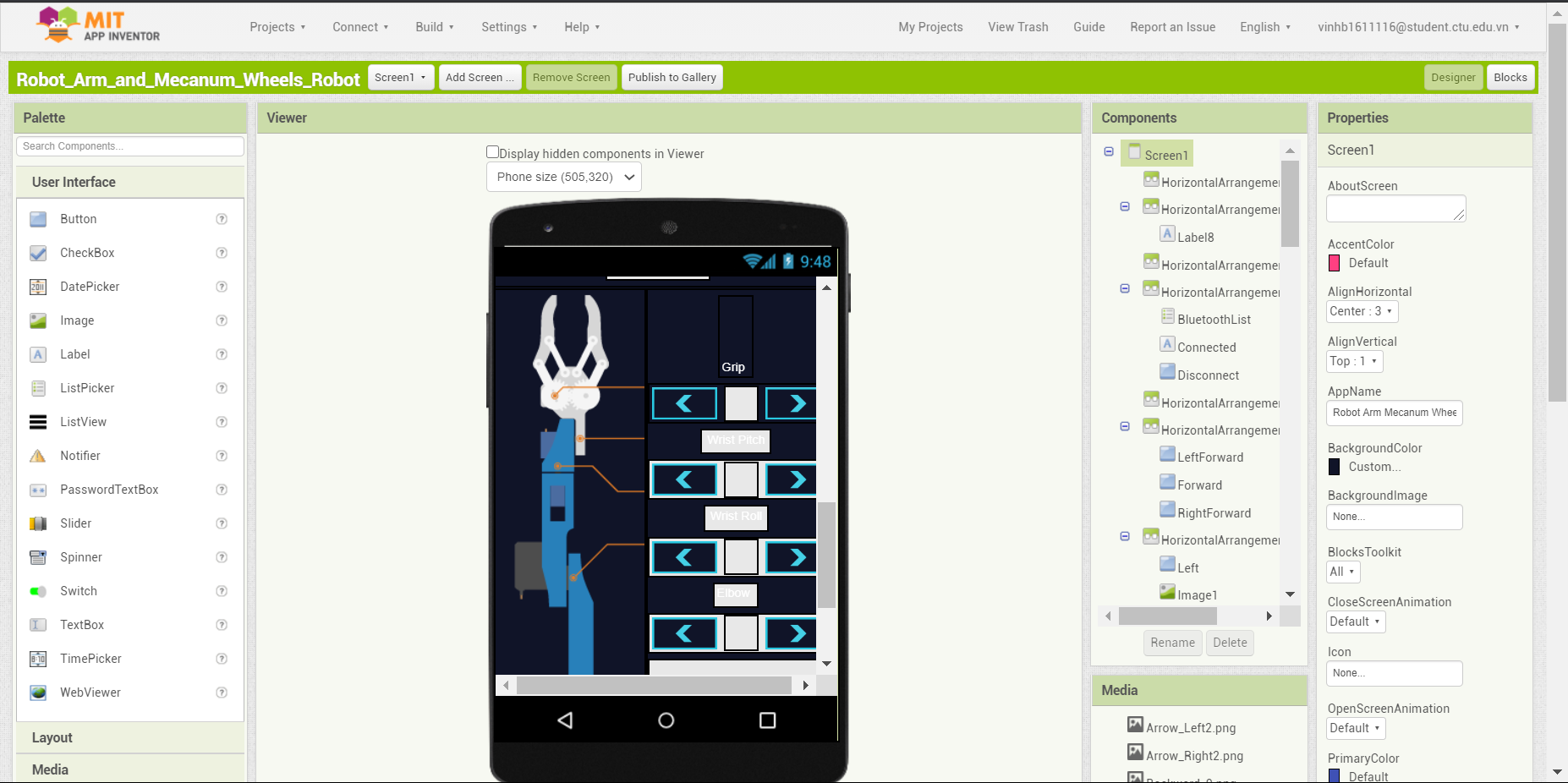


Hình 1. 5. ****MIT App Inventor****

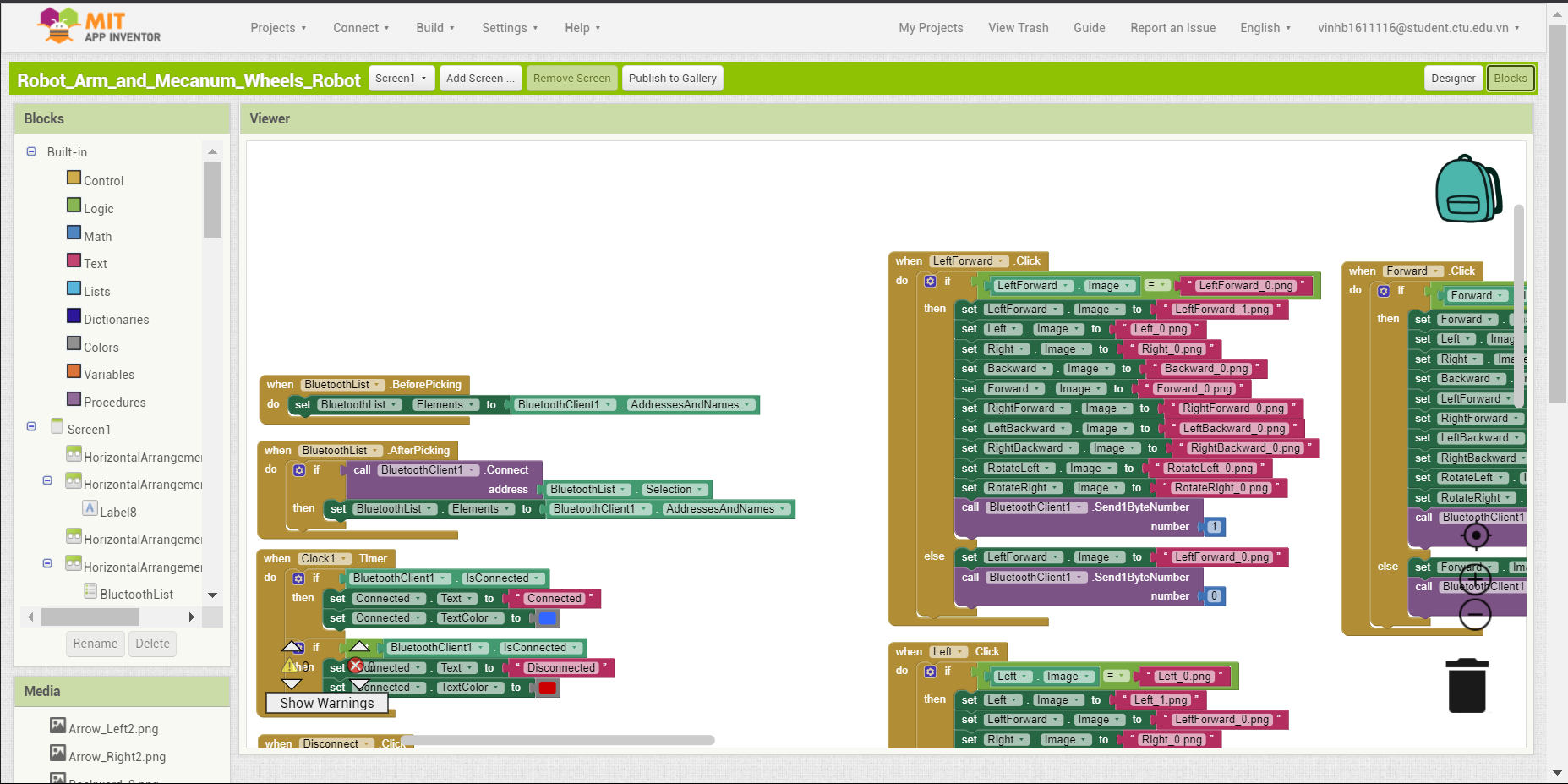
* Mục tiêu cốt lõi của MIT App Inventor là giúp đỡ những người chưa có kiến thức về ngôn ngữ lập trình từ trước có thể tạo ra những ứng dụng có ích trên hệ điều hành Android. Phiên bản mới nhất là MIT App Inventor 2.



Hình 1. 6. ****Giao diện quản lý project****



Hình 1. 7. ****Giao diện thiết kế (Design)****



Hình 1. 8. ****Giao diện lập trình (Blocks)****

1. **Kết nối điện thoại với App Inventor**

Trong quá trình lập trình, có những lúc chúng ta cần phải thử trên điện thoại thật. Do đó, App Inventor cung cấp ba chế độ xem thử cho người dùng. Người dùng có thể tạo một máy ảo để chạy thử chương trình. Hay kết nối với điện thoại bằng cáp USB hoặc thông qua mạng wifi.

Để kết nối với điện thoại, người dùng cần cài đặt ứng dụng MIT AI2 Companion trên Play store.

1. **Kết nối điện thoại với App Inventor qua AI Companion**

Nếu chọn cách kết nối qua mạng. Trên thanh menu của App Inventor, chọn Connect rồi chọn AI Companion. Trên điện thoại, vào ứng dụng MIT AI2 Companion, rồi nhập hoặc quét mã để kết nối.

1. **Kết nối điện thoại với App Inventor qua USB**

Nếu muốn kết nối qua cáp USB, thì cần cài đặt [App Inventor Setup](http://appinv.us/aisetup_windows) cho máy tính trước.

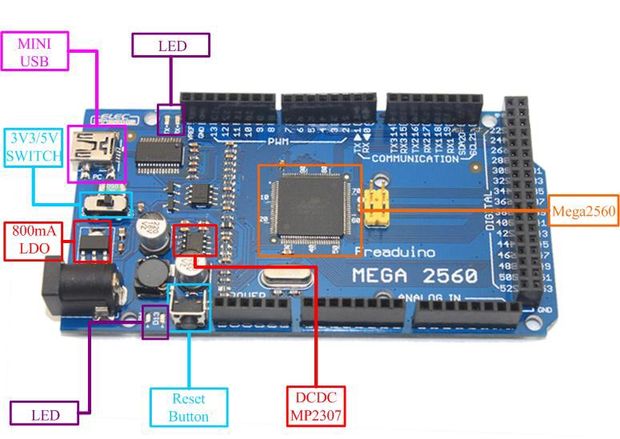
Trên điện thoại, người dùng phải bật chế độ nhà phát triển và USB debugging. Để bật chế độ nhà phát triển, trong phần thông tin của điện thoại. Tìm đến mục Build number và nhấn nhiều lần vào nó cho đến khi có thông báo “Bạn đã là nhà phát triển”. Sau đó, tìm đến tùy chỉnh cho nhà phát triển (developer options) trong cài đặt, rồi check vào USB debugging.

### Các gói cài đặt và thư viện

Thư viện: Servo, Esp32, SoftwareSerial, AccelStepper, Wifi, WebServer, WifiClient, Stepper.

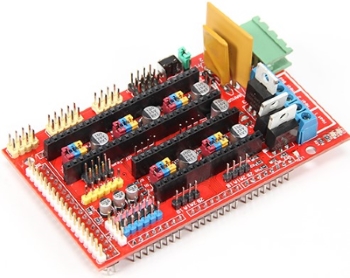
### Phần cứng thiết bị

#### Arduino Mega 2560

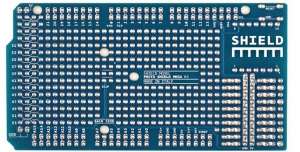


Hình 1. 9. ****Arduino Mega 2560****

* Là một vi điều khiển bằng cách sử dụng ATmega2560 lập trình như là một công cụ chuyển đổi tín hiệu từ USB đang được sử dụng rộng rãi và ứng dụng nhiều hơn.
* Chip ATmega2560.
* Bộ nhớ flash memory 256 KB.
* SRAM 8KB
* EEPROM 4 KB.
* **Cơ bản** **giống Arduino Uno R3**, chỉ khác số lượng chân và nhiều tính năng mạnh mẽ hơn, nên có thể lập trình cho con vi điều khiển này bằng chương trình lập trình cho Arduino Uno R3.
* Arduino Mega có sơ đồ linh kiện bao gồm:
  + 5 Chân GND
  + 3 chân 5V
  + 1 chân 3.3v
  + 1 nút reset
  + 16 chân analog
  + 4 chân UART
  + 54 Chân digital trong đó có 15 chân chúng ta có thể sử dụng như PWM
  + 6 Chân lập trình ISP
* Hiện nay có một số shiled hỗ trợ cho Arduino Mega rất nhiều như:
* Mạch điều khiển máy in 3D RAMPS 1.4
* Arduino Shield Mega Proto: một board mạch mở rộng cho Arduino Mega 2560, Arduino Mega 1280.



Hình 1. 10. ****Mạch điều khiển máy in 3D RAMPS 1.4****



Hình 1. 11. ****Arduino Shield Mega Proto****

* Một số dự án có thể làm bằng Arduino Mega như: ma trận LED 8x8x8 dùng để hiển thị hình ảnh, máy gắp thú bông, các robot điều khiển từ xa, máy in 3D, cánh tay robot, …
* Một số lưu ý khi sử dụng Arduino Mega:
  + Khi bắt đầu sử dụng Arduino Mega 2560, bạn nên chú ý lựa chọn lại board. Bằng cách vào Tool → Board → Arduino Mega 2560. Nhằm tránh trước đó bạn đã sử dụng loại Arduino khác cổng vẫn còn nhận là board cũ nên khi build bạn sẽ gặp lỗi.
  + Khi sử dụng chân RX, TX cuả Arduino, các bạn nên nhớ rút dây cắm tại 2 chân này ra rồi hãy bắt đầu upload. Sau đó hãy cắm lại bình thường và sử dụng để tránh gặp phải lỗi.
  + Không được phép cắm trực tiếp chân GND vào chân nguồn 5V,  có thể dẫn tới hỏng mạch.

#### Module Esp32 Camera

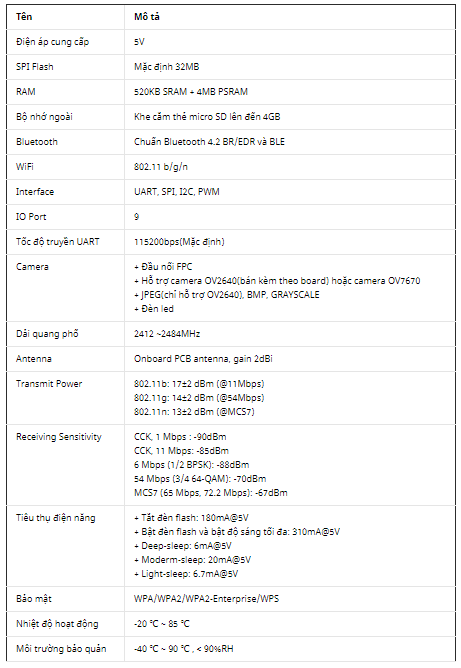
* ESP32-CAM có một module camera cỡ nhỏ có thể hoạt động như một hệ thống độc lập với kích thước chỉ 27×40.5×4.5mm và dòng tiêu thụ ở chế độ deep sleep chỉ có 6 mA.
* ESP32-CAM có kiểu chân DIP-16 hai hàng, có thể dễ dàng lắp đặt trên bread board hay là tích hợp lên bo mạch sản phẩm một cách chắc chắn và  có tính tuỳ biến cao trong mọi ứng dụng IoT.
* Các ứng dụng của bo mạch trong IoTs như: Camera IP, nhận dạng khuôn mặt, nhận dạng màu sắc, nhận dạng hình dạng sản phẩm, nhận diện đồ vật,…



Hình 1. 12. ****Esp32 Camera****

* Thông số kỹ thuật:

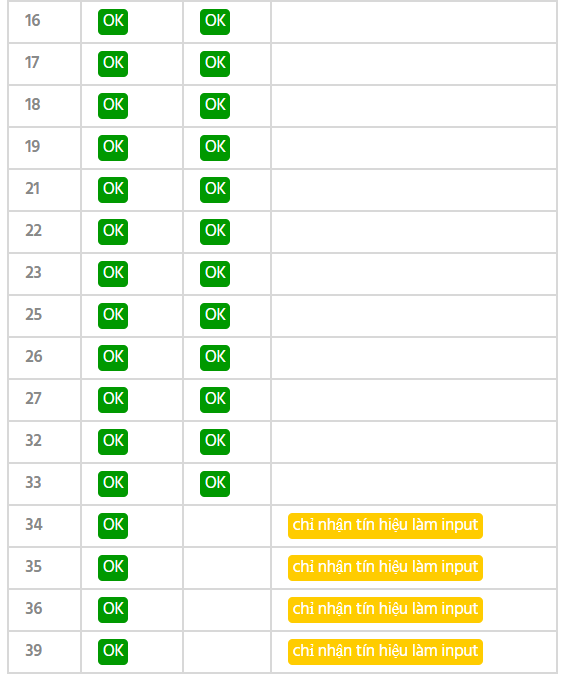
Table 1



Bảng 1. 2. Thông số kỹ thuật Esp 32 Camera

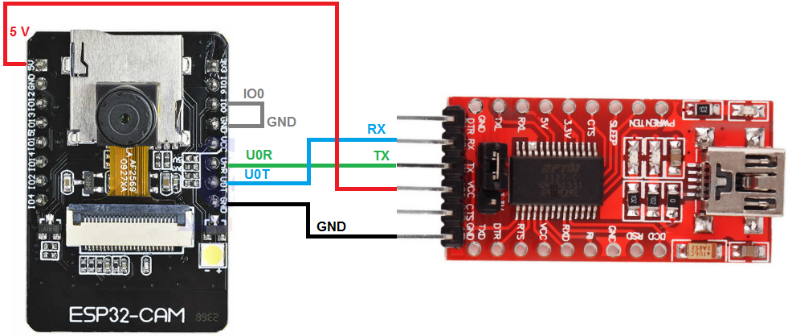
* Bảng sử dụng các chân của module Esp32





Bảng 1. 3. Bảng sử dụng chân Esp32 Camera

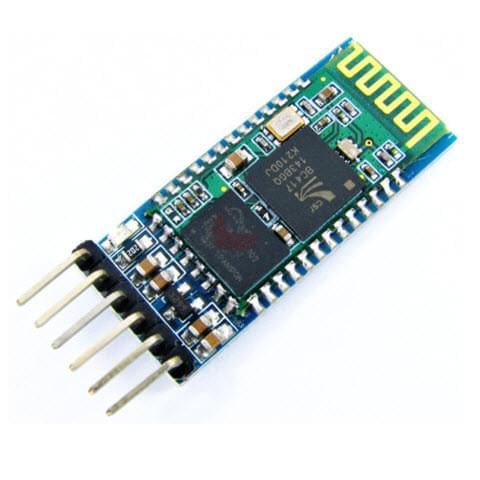
* Một số lưu ý:
  + Đảm bảo công suất đầu vào của module tối thiểu là 5V/2A, nếu không hình ảnh sẽ có hiện tượng bị gợn sóng nước.
  + Pin GPIO32 của ESP32 điều khiển nguồn điện của camera. Khi camera hoạt động, kéo pin GPIO32 xuống mức thấp.
  + Vì IO0 được kết nối với XCLK của camera, do đó để trống IO0 nếu sử dụng.
* Kết nối để nạp code cho Camera:



Hình 1. 13. ****Kết nối Camera với mạch nạp****

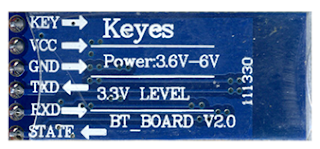
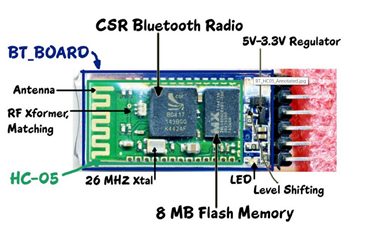
#### Module Bluetooth HC-05

* Là cầu nối giữa vi điều khiển, arduino với các thiết bị ngoại vi như: Điện thoại thông minh, máy tính, Laptop, USB bluetooth thông qua Bluetooth (giao tiếp serial gửi và nhận tín hiệu 2 chiều).
* Module gồm có 6 chân theo thứ tự: EN, VCC, GND, TX, RX, STATE.
* Đây là loại module SLAVE: có nghĩa là không thể chủ động kết nối từ vi điều khiển đến các thiết bị ngoại vi mà cách kết nối là: bạn phải sử dụng thiết bị ngoại vi (điện thoại thông minh, máy tính laptop) để dò tín hiệu kết nối Buletooth mà HC-05 phát ra. Sau khi pair thành công bạn có thể gửi tín hiệu từ vi điều khiển đến các thiết bị ngoại vi này, và ngược lại.
* Module này gồm 6 chân KEY,GND, VCC, TX, RX , STATE  Khi kết nối bạn chỉ cần nối chân TX với chân 0(RX) và chân RX nối với chân 1 (chân TX) trên Arduino uno r3 hoặc tương đương sau đó bạn có thể lập trình gửi và nhận dữ liệu như 1 cổng Serial thông thường.



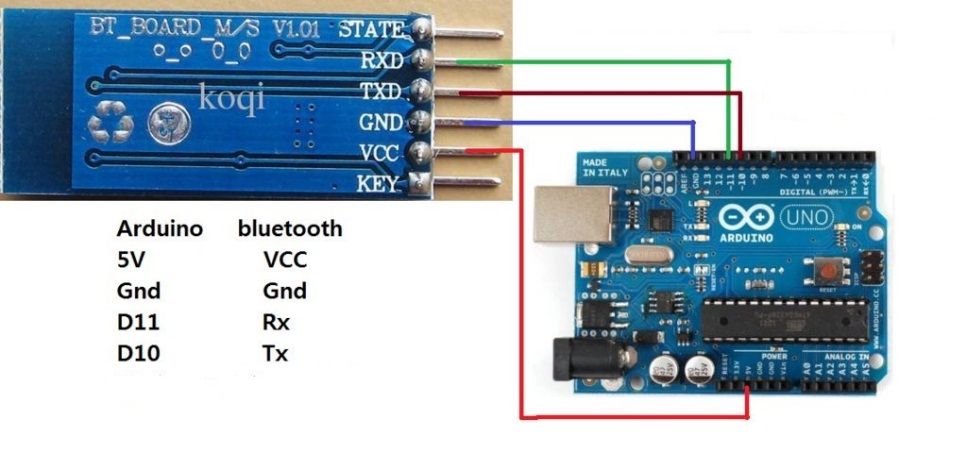
Hình 1. 14. ****Module HC-05****

* Thông số kỹ thuật:
  + Điện áp hoạt động: 3.3 - 5V
  + Dòng tải: khi ghép đôi 30 A, khi truyền tải 8mA
  + Baudrate UART tùy chỉnh: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
  + Dải tần sóng: 2.4GHz
  + Bluetooth protocol:  Bluetooth Specification v2.0+EDRo
  + Kích thước: 26.9 x 13 x 2.2mm
* Sơ đồ chân HC-05 gồm có:
  + KEY: chọn chế độ hoạt động AT Mode hoặc Data Mode.
  + VCC: có thể cấp nguồn từ 3.6V đến 6V bên trong module đã có một ic nguồn chuyển về điện áp 3.3V và cấp cho IC BC417.
  + GND nối với chân nguồn GND
  + TXD,RND: đây là hai chân UART để giao tiếp module hoạt động ở mức logic 3.3V
  + STATE các bạn chỉ cần thả nổi và không cần quan tâm đến chân này.



Hình 1. 15. ****Các chân của HC-05****

* HC-05 có hai chế độ hoạt động là Command Mode và Data Mode:
  + Ở chế độ Command Mode ta có thể giao tiếp với module thông qua cổng serial trên module bằng tập lệnh AT quen thuộc. Ở chế độ Data Mode module có thể truyền nhận dữ liệu tới module bluetooth khác. Chân KEY dùng để chuyển đổi qua lại giữa hai chế độ này.
    - Nếu đưa chân này lên mức logic cao trước khi cấp nguồn module sẽ đưa vào chế độ Command Mode với baudrate mặc định 38400. Chế độ này khá hữu ích khi không biết baudrate trong module được thiết lập ở tốc độ bao nhiêu. Khi chuyển sang chế độ này đèn led trên module sẽ nháy chậm (khoảng 2s) và ngược lại khi chân KEY nối với mức logic thấp trước khi cấp nguồn module sẽ hoạt động chế độ Data Mode.
    - Nếu module đang hoạt động ở chế Data Mode để có thể đưa module vào hoạt động ở chế độ Command Mode bạn đưa chân KEY lên mức cao. Lúc này module sẽ vào chế độ Command Mode nhưng với tốc độ Baud Rate được bạn thiết lập lần cuối cùng. Vì thế bạn phải biết baudrate hiện tại của thiết bị để có thể tương tác được với nó. Chú ý nếu module của bạn chưa thiết lập lại lần nào thì mặc định của nó như sau:
      * Baudrate 9600, data 8 bits, stop bits 1, parity : none, handshake: none
      * Passkey: 1234
      * Device Name: HC-05
  + Ở chế độ Data Mode HC-05 có thể hoạt động như một master hoặc slave tùy vào việc bạn cấu hình (riêng HC-06 bạn chỉ có thể cấu hình ở chế độ SLAVE)
    - Ở chế độ SLAVE: bạn cần thiết lập kết nối từ smartphone, laptop, usb bluetooth để dò tìm module sau đó pair với mã PIN là 1234. Sau khi pair thành công, bạn đã có 1 cổng serial từ xa hoạt động ở baud rate 9600.
    - Ở chế độ MASTER: module sẽ tự động dò tìm thiết bị bluetooth khác (1 module bluetooth HC-06, usb bluetooth, bluetooth của laptop...) và tiến hành pair chủ động mà không cần thiết lập gì từ máy tính hoặc smartphone.
* Kết nối HC-05 với Arduino



Hình 1. 16. Kết nối HC-05 với Arduino

#### Động cơ bước Nema 17 size 42

* Động cơ bước (Stepper Motor), thực chất là một động cơ đồng bộ dùng để biến đổi các tín hiệu điều khiển dưới dạng các xung điện rời rạc kế tiếp nhau thành các chuyển động góc quay.
* Là loại động cơ bước thường được dùng khi chế tạo máy in 3D, CNC mini và thường được lắp cùng với puli GT2.

1. **Cấu tạo:**

* Về cấu tạo động cơ bước gồm có các bộ phận là stato, roto là nam châm vĩnh cửu hoặc trong trường hợp của động cơ biến từ trở là những khối răng làm bằng vật liệu nhẹ có từ tính. Động cơ bước được điều khiển bởi bộ điều khiển bên ngoài. Động cơ bước và bộ điều khiển được thiết kế sao cho động cơ có thể giữ nguyên bất kỳ vị trí cố định nào cũng như quay đến một vị trí bất kỳ nào.
* Động cơ bước có thể sử dụng trong hệ thống điều khiển vòng hở đơn giản, hoặc vòng kín, tuy nhiên khi sử dụng động cơ bước trong hệ điều khiển vòng hở khi quá tải, tất cá các giá trị của động cơ đều bị mất và hệ thống cần nhận diện lại.

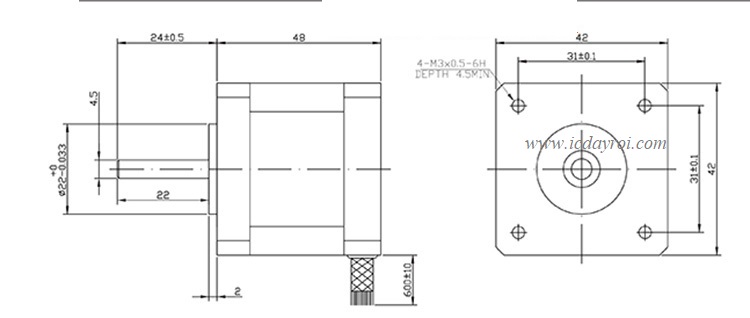
1. **Đặc điểm:**

* Động cơ bước hoạt động dưới tác dụng của các xung rời rạc và kế tiếp nhau. Khi có dòng điện hay điện áp đặt vào cuộn dây phần ứng của động cơ bước làm cho roto của động cơ quay một góc nhất định gọi là bước của động cơ.
* Góc bước là góc quay của trục động cơ tương ứng với một xung điều khiển. Góc bước được xác định dựa vào cấu trúc của động cơ bước và phương pháp điều khiển động cơ bước.
* Tính năng mở máy của động cơ được đặc trưng bởi tần số xung cực đại có thể mở máy mà không làm cho roto mất đồng bộ.
* Chiều quay động cơ bước không phụ thuộc vào chiều dòng điện mà phụ thuộc vào thứ tự cấp xung cho các cuộn dây.



Hình 1. 17. ****Động cơ bước Size 42****

* Đặc điểm nổi bật:
  + Đường kính trục 5mm. Đầu trục vát phẳng, giúp puli/khớp nối không đỡ bị lỏng khi hoạt động
  + Cường độ định mức 1.5A, mô men giữ 0.55 Nm, góc quay mỗi bước 1.8 °
  + Dây nối dài 1m, đầu dây chuẩn XH2.54. Tương thích với đầu ra động cơ bước trên mạch RAMPS 1.5 hoặc CNC shield V3
  + Công suất phù hợp cho máy in 3D và CNC mini
  + Ít tỏa nhiệt, chuyển động êm
  + Khối lượng: 400g



Hình 1. 18. ****Sơ đồ kỹ thuật động cơ bước****

* Thông số kỹ thuật chính:
  + Kích thước mặt bích: 42×42 mm.
  + Chiều dài thân: 40 mm.
  + Dòng chịu tải: 1.6A.
  + Moment xoắn: 0.45Nm
  + Góc bước: 1.8°/step

#### Động cơ Servo SG90

* Servo là một dạng động cơ điện đặc biệt. Không giống như động cơ thông thường cứ cắm điện vào là quay liên tục, servo chỉ quay khi được điều khiển (bằng xung PPM) với góc quay nằm trong khoảng bất kì từ 0º -180º. Mỗi loại servo có kích thước, khối lượng và cấu tạo khác nhau. Có loại thì nặng chỉ 9g (chủ yếu dùng trên máy bay mô mình), có loại thì sở hữu một momen lực rất mạnh (vài chục Newton/m), hoặc có loại thì khỏe và nhông sắc chắc chắn,...
* Động cơ servo được thiết kế 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. Các động cơ servo điều khiển bằng liên lạc vô tuyến được gọi là động cơ servo RC (radio-controlled). Trong thực tế, bản thân động cơ servo không phải được điều khiển bằng vô tuyến, nó chỉ nối với máy thu vô tuyến trên máy bay hay xe hơi. Động cơ servo nhận tín hiệu từ máy thu này.
* Động cơ RC Servo 9G là động phổ biến dùng trong các mô hình điều khiển nhỏ và đơn giản như cánh tay robot. Động cơ có tốc độ phản ứng nhanh, được tích hợp sẵn Driver điều khiển động cơ, dễ dàng điều khiển góc quay bằng phương pháp điều độ rộng xung PWM.

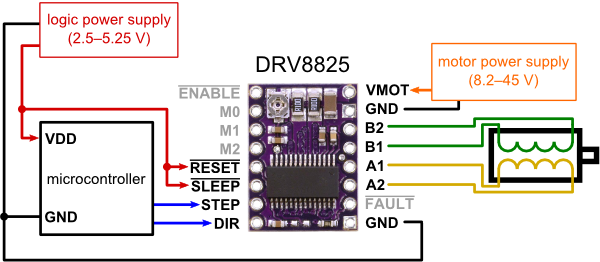


Hình 1. 19. ****Động cơ Servo****

* Thông số kỹ thuật:
  + Tên: Servo MSG90
  + Điện áp hoạt động: 4.8V ~ 6V DC
  + Tốc độ quay: 0.12 giây/60° (4.8V) , 0.1 giây/60° (6V)
  + Mômen xoắn: 1.8kg/cm (4.8V) , 2.5kg/cm (6V)
  + Góc quay: 180°
  + Bánh răng: nhựa
  + Kích thước: 22.5 \* 11.8 \* 30 mm
  + Chiều dài dây điện: 175mm
  + Trọng lượng: 9g
  + Nhiệt độ hoạt động: 0°C ~ 55°C
    - Dây cam: Xung
    - Dây đỏ: Vcc (4.8V ~ 6V)
    - Dây đen: GND / 0V

#### Mạch điều khiển động cơ bước DRV8825

* DRV8825 là mạch điều khiển động cơ bước lưỡng cực microstepping. Module có chân ra và giao tiếp hoàn toàn giống driver A4988, vì vậy có thể được dùng nâng cao hiệu xuấtthay thế cho board A4988 trong nhiều loại ứng dụng khác nhau. DRV8825 có chức năng điều chỉnh dòng điện, có mạch bảo về quá tải dòng điện và quá tải nhiệt độ. Độ phân giải 6 microstep tương đương 1/32 bước. Hoạt động trong khoản 8.2V đến 45V và có thể chuyển tải xấp xỉ 1.5A trên mỗi phase mà không cần tản nhiệt. Cường độ lên đến 2.2A trên mỗi cuộn nếu được tản nhiệt tốt.
* Mạch điều khiển động cơ bước DRV8825 với đầy đủ các tính năng của môt driver chuyên nghiệp: điều chỉnh dòng giới hạn, vi bước ( 1/32 bước) ,bảo vệ quá dòng, quá nhiệt,v.v…



Hình 1. 20. Mạch điều khiển động cơ bước

* Tính năng:
  + PWM khiển vi stepper motor điều khiển
    - Built-In khiển vi Indexer
    - Lên đến 1/32 vi bước
  + Nhiều chế độ Decay
    - Mixed Decay
    - Chậm Decay
    - Nhanh Decay
  + 8.2-V đến 45 V-điều hành
  + Cung cấp điện áp Phạm vi
  + 2.5-A hiện tại Maximum Drive tại 24 V TA = 25 ° C
  + Đơn giản BƯỚC / Interface DIR
  + Low Chế độ Ngủ hiện tại
  + Built-In 3,3-V Reference Output
  + Gói nhỏ và Footprint
* Đặc điểm
  + Bảo vệ pin.
    - Bảo vệ quá dòng (OCP)
    - Nhiệt Shutdown (TSD)
    - VM undervoltage Lockout (UVLO)
    - Điều kiện lỗi Chỉ Pin (nFAULT)
* Thông số kỹ thuật:
  + Dòng cực đại cấp cho động cơ: **2.5A**.
  + Điều khiển bước động cơ: 1; 1/2; 1/4; 1/8; 1/16; **1/32**.
  + Điện trở trong IC thấp, hạn chế tỏa nhiệt.
  + Kích thước 1.5mm X 2mm (and the same 4988)
  + Thích hợp điều khiển động cơ bước từ 8.2V ~ 45V 2.5A;
  + Giao thức điều khiển số bước và chiều quay đơn giản
  + Điều chỉnh dòng cực đại cấp cho động cơ dễ dàng
  + Bảo vệ ngắn mạch, quá dòng, quá nhiệt…
* Module DRV8825 thích hợp cho các ứng dụng
  + Điều khiển động cơ bước thông thường
  + Máy in 3D, CNC mini (tương thích với board RAMPS 1.4 hoặc CNC Shield V3)
  + Dòng cực đại cấp cho động cơ bước có thể lên tới 2.5 A
  + Điều khiển bước của động cơ bước có thể xuống tới 1/32 bước.
  + Mạch in 4 lớp.
  + Điện trở trong của IC thấp, giảm phát nhiệt, hoạt động ổn định.

#### Động cơ RC Servo MG996

Động cơ servo là một thiết bị tự động có sử dụng lỗi cảm biến phản hồi âm để điều chỉnh hành động của một cơ cấu. Trong đó RC Servo là một loại động cơ của Servo được ứng dụng rất nhiều trong công nghiệp.

Động cơ RC Servo là loại động cơ có tốc độ thấp, mô - men xoắn cao và có nhiều kích thước to nhỏ khác nhau tùy vào thiết bị. Khác hẳn với động cơ DC và Stepper, động cơ RC Servo không xoay ở góc 360 độ như bình thường mà loại động cơ này bị giới hạn trong phạm vi ở góc 180, 270 và 90 độ

. 

Hình 1. 21. ****Động cơ RC****

1. **Cấu tạo:**

Cấu tạo của một động cơ RC servo bao gồm 10 phần cơ bản:

* Động cơ chính
* Board điều khiển tín hiệu hồi tiếp
* Dây nguồn (red)
* Dây tín hiệu vào (Yellow or White)
* Dây mass (black)
* Volt kế
* Trục/ bánh răng output
* Horn/Wheel/Arm gắn kèm
* Vỏ servo
* Chip điều khiển chính

1. **Thông số kỹ thuật:**

* Chủng loại: Analog RC Servo.
* Điện áp hoạt động: 4.8~6.6VDC
* Lực kéo:
* 3.5 kg-cm (180.5 ozin) at 4.8V-1.5A
* 5.5 kg-cm (208.3 ozin) at 6V-1.5A
* Tốc độ quay:
* 0.17sec / 60 degrees (4.8V no load)
* 0.13sec / 60 degrees (6.0V no load)
* Kích thước: 40mm x 20mm x 43mm
* Trọng lượng: 55g

1. **Ứng dụng:**

* Các servo RC được sử dụng chủ yếu để cung cấp các phát động cho các hệ thống cơ khí khác nhau như: Hệ thống lái của xe hơi, các bề mặt điều khiển trên máy bay hay các bánh lái của một chiếc thuyền.
* RC servo thường được sử dụng trong các mô hình điều khiển bằng radio mà sử dụng một động cơ chuyển động tự do và một cảm biến vị trí chiết áp đơn giản với một bộ điều khiển nhúng.

**Tổng kết chương 1**

Trong chương 1 đã trình bày sơ lược về các khái niệm, phần mềm và phần cứng, cácc thông số kỹ thuật và nguyên lý hoạt động của các thiết bị, các board mạch Mega. Trình bày cách làm việc và các thư viện cần cho việc lập trình mô hình. Tiếp theo trong chương 2 sẽ trình bày nội dung mô tả bài toán của đề tài dựa trên cơ sở lý thuyết của chương 1.

## CHƯƠNG 2: MÔ TẢ BÀI TOÁN

### Mô tả tổng quan

#### Bối cảnh sản phẩm

Hiện tại Xe cánh tay Robot được xây dựng với dạng mô hình thử nghiệm gắp được những vật dụng nhỏ, nhẹ và di chuyển trên địa hình phẳng, nơi có điều kiện ánh sáng đầy đủ. Với mong muốn đây sẽ là nền tảng để có thể phát triển mô hình thay thế con người tại những nơi cần vận chuyện nhiều hàng hóa, vật dụng như: nhà máy, nhà kho, xưởng sản xuất, kho vận chuyển,…

#### Các tính năng của mô hình

* Điều khiển từ xa cánh tay robot.
* Điều khiển gắp vật từ xa.

#### Thiết kế sơ đồ khối

**Pin**

(Cấp nguồn)

**Điện thoại**

(Nhận lệnh)

**Mega**

(Xử lý trung tâm)

**Động cơ**

(Thực thi)

**Ắc quy**

(Cấp nguồn)

**Hình 2.1. Sơ đồ bài toán**

**Chức năng của các thành phần trong khối:**

* Điện thoại: Nhận lệnh từ người dùng bằng giao diện đồ họa trên màn hình từ đó gửi tín hiệu về khối xử lý trung tâm
* Mega (Khối xử lý trung tâm): Nhận lệnh từ điện thoại gửi đến và xử lý lệnh nhận được, đồng thời gửi tín hiệu thực thi các động cơ.
* Động cơ: Nhận tín hiệu yêu cầu thực thi của khối xử lý trung tâm và thực thiện nó.
* Ắc quy: Cung cấp nguồn cho động cơ.
* Pin: Cung cấp nguồn cho khối xử lý trung tâm, Camera.
* Camera: Thu hình ảnh trong khi hoạt động, gửi hình ảnh cho điện thoại.

#### Phần cứng yêu cầu các thiết bị

* Arduino Mega 2560
* Ba Servo SG90
* Ba Động cơ RC Servo MG996
* Bốn motor động cơ bước
* Bốn bánh xe Omni
* Bốn mạch điều khiển động cơ bước DRV8825
* Esp32 Camera
* Pin
* Ắc quy

#### Phần mềm yêu cầu

* Arduino yêu cầu phiên bản 1.6.0 trở lên.
* Các gói thư viện: Servo, Esp32, SoftwareSerial, AccelStepper, Wifi, WebServer, WifiClient, Stepper

### Yêu cầu giao tiếp

* Giao tiếp phần cứng:
  + Mega kết nối với nguồn bằng cổng USB.
  + Động cơ và Camera kết nối trực tiếp với Mega thông qua các chân Analog, Digital
* Giao tiếp phần mềm:
  + Sử dụng hỗ trợ phần mềm Arduino IDE 1.8.13 để lập trìng và upload chương trình vào board Mega.
  + Điện thoại truy cập từ xa trong môi trường mạng Lan hoặc mạng Internet.

**Tổng kết chương 2**

Trong chương 2 đã trình bày sơ lược về những mô tả chi tiết cho bài toán, đưa ra các phân tích về yêu cầu bài toán. Tiếp theo, trong chương 3 sẽ trình bày nội dung cài đặt giải pháp của đề tài trên từng phân hệ dựa trên mô tả bài toán từ chương 2.

## CHƯƠNG 3: CÀI ĐẶT GIẢI PHÁP

### Cài đặt

Để xây dựng mô hình cần cài đặt

* Trình soạn thảo văn bản Arduino IDE
* Phần mềm lập trình cho di động MIT App Inventor
* Sử dụng ngôn ngữ C lập trình cho Mega và ngôn ngữ ….. để lập trình cho động cơ
* Thư viện Servo, Esp32, SoftwareSerial, AccelStepper, Wifi, WebServer, WifiClient, Stepper

### Lưu đồ giải thuật

#### Thuật toán phần mềm điều khiển

Khởi tạo giao diện

Bluetooth on

ooth on

Mở phần mềm điều khiển

Sai

Tìm kiếm xe

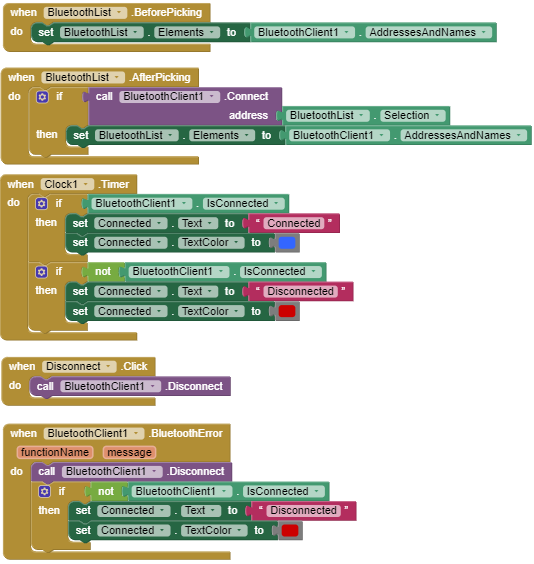
Đúng

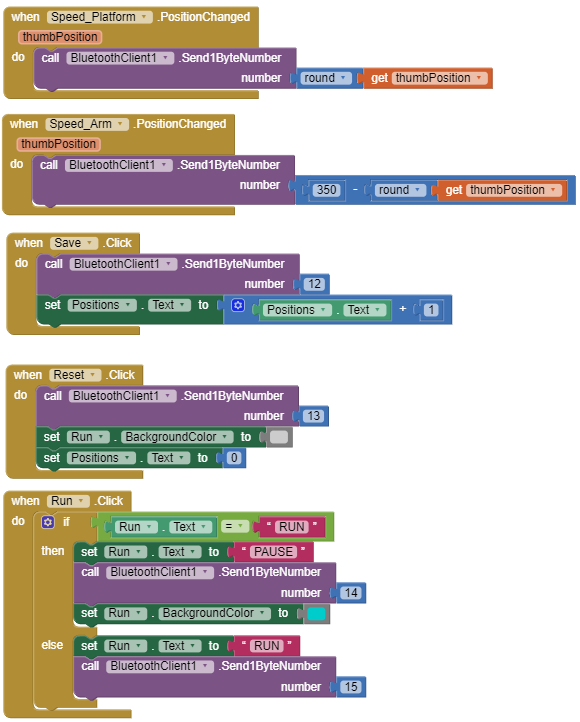
Đợi even từ người dùng

Hành động cánh tay

### Lập trình cánh tay và app MIT inventor

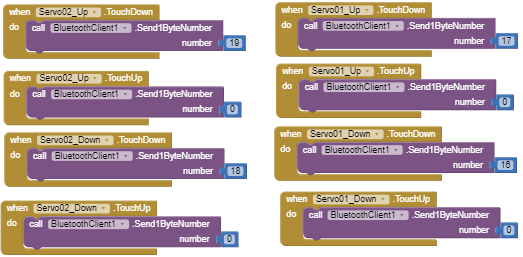
#### Code app: Blocks



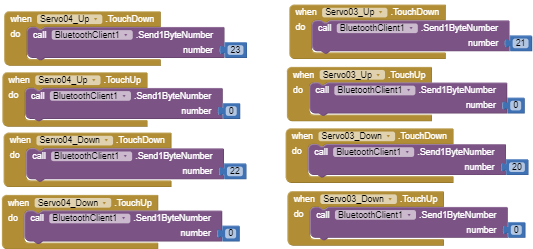


Hình 1. 22. ****Code Bluetooth và lập bước****

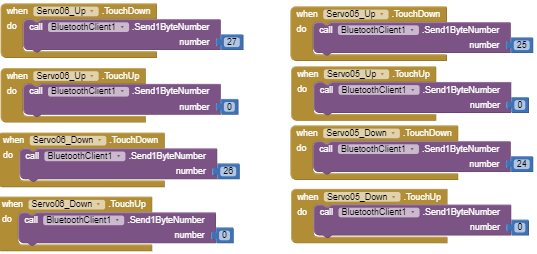
#### Code điều khiển cánh tay trên app MIT:



Hình 1. 23. ****Servo 1 và 2****



Hình 1. 24. ****Servo 3 và 4****



Hình 1. 25. ****Servo 5 và 6****

#### **Các khớp cánh tay:**

Các khớp cánh tay tương ứng với 6 bậc tùy theo chúng ta muốn cánh tay ở tư thế như thế nào một cách linh hoạt nhất.

#### **Bậc 1: khớp lưng cánh tay tương ứng với nút “Waist” trên giao diện.**

#### **Bậc 2: Khớp vai cánh tay tương ứng với nút “Shoulder” trên giao diện.**

#### **Bậc 3: Khớp khủy tay tương ứng với nút “Elbow” trên giao diện.**

Bậc 4: Khớp xoay khủy tay **tương ứng với nút “Wrist roll” trên giao diện.**

Bậc 5: Khớp lên xuống cổ tay **tương ứng với nút “Wrist Pitch” trên giao diện.**

Bậc 6: Khớp ngón tay **tương ứng với nút “Grip” trên giao diện.**

**Tổng kết chương 3**

Trong chương 3 đã trình bày nội dung cài đặt giải pháp của đề tài, phân tích cụ thể cách lập trình cho mô hình. Tiếp theo, trong chương 4 sẽ đánh giá ưu nhược điểm và kiểm thử hoạt động của mô hình.

## CHƯƠNG 4: ĐÁNH GIÁ VÀ KIỂM THỬ

### Mục tiêu kiểm thử

Khi sử dụng mô hình vào ứng dụng thực tế như: nhà máy, xí nghiệp có phạm vi lớn, cần vận chuyển những vật dụng nặng có được hay không. Đồng thời kiểm tra các ứng dụng của động cơ cũng như kết nối các thiết bị qua bluetooth có như mong đợi hay không

### Kịch bản kiểm thử

* Bước 1: Bật nút nguồn để khởi động Bluetooth và nguồn cho cánh tay.
* Bước 2: Mở ứng dụng trên điện thoại và kết nối với cánh tay thông qua Bluetooth.
* Bước 3: Di chuyển cánh tay bằng các nút trên điện thoại .
* Bước 4: Gắp thử vật nhẹ.

### Kết quả kiểm thử

* Kết nối Bluetooth dễ dàng.
* Các servo di chuyển mượt mà, tốc độ có thể tùy chỉnh.
* Các servo có độ ồn nhỏ.
* Cánh tay có thể gắp các vật nhẹ nhỏ hơn 50g với bề mặt tiếp xúc phẳng.
* Khả năng chịu lực toàn cánh tay là nhỏ hơn 200g
* Phạm vi nối kết với bán kính nhỏ hơn 5m.

**Tổng kết chương 4**

Trong chương 4 đã phân tích mục tiêu và đề ra kịch bản để kiểm thử, đánh giá được mô hình để vận dụng và phát triển thêm khi đưa vào thực tiễn.

# PHẦN KẾT LUẬN

## Kết quả đạt được

* Nắm rõ về Bluetooth
* Nắm rõ về hoạt động của động cơ servo
* Tìm hiểu về hệ điều hành Android
* Tìm hiểu về Camera
* Thực hiện viết ứng dụng trên Android
* Thực hiện kết nối Camera với ứng dụng trên Android
* Thực hiện kết nối và trao đổi dữ liệu giữa điện thoại thông minh, servo và Arduino Mega qua module Bluetooth

## Khó khăn

* Mạch dễ xảy ra cháy nổ.
* Servo hoạt động dễ nóng.
* Hạn chế nguồn điện không thể hoạt động lâu.

## Hướng phát triển

* Tích hợp thêm chức năng cho xe như: đèn, cảm biến (gas, lửa, ánh sáng, khoảng cách,…)
* Ứng dụng công nghệ Bluetooth vào các hệ thống khác.
* Có thể tính hợp thêm Google Assistant để điều khiển cánh tay bằng giọng nói.

**CODE NẠP CHO ARDUINO**

#include <SoftwareSerial.h>

#include <AccelStepper.h>

#include <Servo.h>

Servo servo01;

Servo servo02;

Servo servo03;

Servo servo04;

Servo servo05;

Servo servo06;

SoftwareSerial Bluetooth(A8, 38); // Arduino(RX, TX) - HC-05 Bluetooth (TX, RX)

#define led 14

int lbw[50], lfw[50], rbw[50], rfw[50];

int servo1Pos, servo2Pos, servo3Pos, servo4Pos, servo5Pos, servo6Pos;

int servo1PPos, servo2PPos, servo3PPos, servo4PPos, servo5PPos, servo6PPos; position

int servo01SP[50], servo02SP[50], servo03SP[50], servo04SP[50], servo05SP[50], servo06SP[50]; // for storing positions/steps

int speedDelay = 20;

int index = 0;

int dataIn;

int m = 0;

void setup() {

pinMode(led, OUTPUT);

servo01.attach(5);

servo02.attach(6);

servo03.attach(7);

servo04.attach(8);

servo05.attach(9);

servo06.attach(10);

Bluetooth.begin(38400);

Bluetooth.setTimeout(5);

delay(20);

Serial.begin(38400);

// Move robot arm to initial position

servo1PPos = 90;

servo01.write(servo1PPos);

servo2PPos = 100;

servo02.write(servo2PPos);

servo3PPos = 120;

servo03.write(servo3PPos);

servo4PPos = 95;

servo04.write(servo4PPos);

servo5PPos = 60;

servo05.write(servo5PPos);

servo6PPos = 110;

servo06.write(servo6PPos);

}

void loop() {

// Check for incoming data

if (Bluetooth.available() > 0) {

dataIn = Bluetooth.read(); // Read the data

if (dataIn == 0) {

m = 0;

}

if (dataIn == 1) {

m = 1;

}

if (dataIn == 2) {

m = 2;

}

if (dataIn == 3) {

m = 3;

}

if (dataIn == 4) {

m = 4;

}

if (dataIn == 5) {

m = 5;

}

if (dataIn == 6) {

m = 6;

}

if (dataIn == 7) {

m = 7;

}

if (dataIn == 8) {

m = 8;

}

if (dataIn == 9) {

m = 9;

}

if (dataIn == 10) {

m = 10;

}

if (dataIn == 11) {

m = 11;

}

if (dataIn == 12) {

m = 12;

}

if (dataIn == 14) {

m = 14;

}

if (dataIn == 16) {

m = 16;

}

if (dataIn == 17) {

m = 17;

}

if (dataIn == 18) {

m = 18;

}

if (dataIn == 19) {

m = 19;

}

if (dataIn == 20) {

m = 20;

}

if (dataIn == 21) {

m = 21;

}

if (dataIn == 22) {

m = 22;

}

if (dataIn == 23) {

m = 23;

}

if (dataIn == 24) {

m = 24;

}

if (dataIn == 25) {

m = 25;

}

if (dataIn == 26) {

m = 26;

}

if (dataIn == 27) {

m = 27;

}

// Move the Mecanum wheels platform

// Mecanum wheels speed

// Move robot arm

// Move servo 1 in positive direction

while (m == 16) {

if (Bluetooth.available() > 0) {

m = Bluetooth.read();

}

servo01.write(servo1PPos);

servo1PPos++;

delay(speedDelay);

}

// Move servo 1 in negative direction

while (m == 17) {

if (Bluetooth.available() > 0) {

m = Bluetooth.read();

}

servo01.write(servo1PPos);

servo1PPos--;

delay(speedDelay);

}

// Move servo 2

while (m == 19) {

if (Bluetooth.available() > 0) {

m = Bluetooth.read();

}

servo02.write(servo2PPos);

servo2PPos++;

delay(speedDelay);

}

while (m == 18) {

if (Bluetooth.available() > 0) {

m = Bluetooth.read();

}

servo02.write(servo2PPos);

servo2PPos--;

delay(speedDelay);

}

// Move servo 3

while (m == 20) {

if (Bluetooth.available() > 0) {

m = Bluetooth.read();

}

servo03.write(servo3PPos);

servo3PPos++;

delay(speedDelay);

}

while (m == 21) {

if (Bluetooth.available() > 0) {

m = Bluetooth.read();

}

servo03.write(servo3PPos);

servo3PPos--;

delay(speedDelay);

}

// Move servo 4

while (m == 23) {

if (Bluetooth.available() > 0) {

m = Bluetooth.read();

}

servo04.write(servo4PPos);

servo4PPos++;

delay(speedDelay);

}

while (m == 22) {

if (Bluetooth.available() > 0) {

m = Bluetooth.read();

}

servo04.write(servo4PPos);

servo4PPos--;

delay(speedDelay);

}

// Move servo 5

while (m == 25) {

if (Bluetooth.available() > 0) {

m = Bluetooth.read();

}

servo05.write(servo5PPos);

servo5PPos++;

delay(speedDelay);

}

while (m == 24) {

if (Bluetooth.available() > 0) {

m = Bluetooth.read();

}

servo05.write(servo5PPos);

servo5PPos--;

delay(speedDelay);

}

// Move servo 6

while (m == 26) {

if (Bluetooth.available() > 0) {

m = Bluetooth.read();

}

servo06.write(servo6PPos);

servo6PPos++;

delay(speedDelay);

}

while (m == 27) {

if (Bluetooth.available() > 0) {

m = Bluetooth.read();

}

servo06.write(servo6PPos);

servo6PPos--;

delay(speedDelay);

}

// If arm speed slider is changed

if (dataIn > 101 & dataIn < 250) {

speedDelay = dataIn / 10; // Change servo speed (delay time)

}

// If button "SAVE" is pressed

if (m == 12) {

servo01SP[index] = servo1PPos; // save position into the array

servo02SP[index] = servo2PPos;

servo03SP[index] = servo3PPos;

servo04SP[index] = servo4PPos;

servo05SP[index] = servo5PPos;

servo06SP[index] = servo6PPos;

index++; // Increase the array index

m = 0;

}

// If button "RUN" is pressed

if (m == 14) {

runSteps();

// If button "RESET" is pressed

if (dataIn != 14) {

stopMoving();

memset(lbw, 0, sizeof(lbw)); // Clear the array data to 0

memset(lfw, 0, sizeof(lfw));

memset(rbw, 0, sizeof(rbw));

memset(rfw, 0, sizeof(rfw));

memset(servo01SP, 0, sizeof(servo01SP)); // Clear the array data to 0

memset(servo02SP, 0, sizeof(servo02SP));

memset(servo03SP, 0, sizeof(servo03SP));

memset(servo04SP, 0, sizeof(servo04SP));

memset(servo05SP, 0, sizeof(servo05SP));

memset(servo06SP, 0, sizeof(servo06SP));

index = 0; // Index to 0

}

}

}

}

// Automatic mode custom function - run the saved steps

void runSteps() {

while (dataIn != 13) { // Run the steps over and over again until "RESET" button is pressed

for (int i = 0; i <= index - 2; i++) { // Run through all steps(index)

if (Bluetooth.available() > 0) { // Check for incomding data

dataIn = Bluetooth.read();

if ( dataIn == 15) { // If button "PAUSE" is pressed

while (dataIn != 14) { // Wait until "RUN" is pressed again

if (Bluetooth.available() > 0) {

dataIn = Bluetooth.read();

if ( dataIn == 13) {

break;

}

}

}

}

// If speed slider is changed

if (dataIn > 100 & dataIn < 150) {

speedDelay = dataIn / 10; // Change servo speed (delay time)

}

// Mecanum wheels speed

if (dataIn > 30 & dataIn < 100) {

wheelSpeed = dataIn \* 10;

dataIn = 14;

}

}

// Servo 1

if (servo01SP[i] == servo01SP[i + 1]) {

}

if (servo01SP[i] > servo01SP[i + 1]) {

for ( int j = servo01SP[i]; j >= servo01SP[i + 1]; j--) {

servo01.write(j);

delay(speedDelay);

}

}

if (servo01SP[i] < servo01SP[i + 1]) {

for ( int j = servo01SP[i]; j <= servo01SP[i + 1]; j++) {

servo01.write(j);

delay(speedDelay);

}

}

// Servo 2

if (servo02SP[i] == servo02SP[i + 1]) {

}

if (servo02SP[i] > servo02SP[i + 1]) {

for ( int j = servo02SP[i]; j >= servo02SP[i + 1]; j--) {

servo02.write(j);

delay(speedDelay);

}

}

if (servo02SP[i] < servo02SP[i + 1]) {

for ( int j = servo02SP[i]; j <= servo02SP[i + 1]; j++) {

servo02.write(j);

delay(speedDelay);

}

}

// Servo 3

if (servo03SP[i] == servo03SP[i + 1]) {

}

if (servo03SP[i] > servo03SP[i + 1]) {

for ( int j = servo03SP[i]; j >= servo03SP[i + 1]; j--) {

servo03.write(j);

delay(speedDelay);

}

}

if (servo03SP[i] < servo03SP[i + 1]) {

for ( int j = servo03SP[i]; j <= servo03SP[i + 1]; j++) {

servo03.write(j);

delay(speedDelay);

}

}

// Servo 4

if (servo04SP[i] == servo04SP[i + 1]) {

}

if (servo04SP[i] > servo04SP[i + 1]) {

for ( int j = servo04SP[i]; j >= servo04SP[i + 1]; j--) {

servo04.write(j);

delay(speedDelay);

}

}

if (servo04SP[i] < servo04SP[i + 1]) {

for ( int j = servo04SP[i]; j <= servo04SP[i + 1]; j++) {

servo04.write(j);

delay(speedDelay);

}

}

// Servo 5

if (servo05SP[i] == servo05SP[i + 1]) {

}

if (servo05SP[i] > servo05SP[i + 1]) {

for ( int j = servo05SP[i]; j >= servo05SP[i + 1]; j--) {

servo05.write(j);

delay(speedDelay);

}

}

if (servo05SP[i] < servo05SP[i + 1]) {

for ( int j = servo05SP[i]; j <= servo05SP[i + 1]; j++) {

servo05.write(j);

delay(speedDelay);

}

}

// Servo 6

if (servo06SP[i] == servo06SP[i + 1]) {

}

if (servo06SP[i] > servo06SP[i + 1]) {

for ( int j = servo06SP[i]; j >= servo06SP[i + 1]; j--) {

servo06.write(j);

delay(speedDelay);

}

}

if (servo06SP[i] < servo06SP[i + 1]) {

for ( int j = servo06SP[i]; j <= servo06SP[i + 1]; j++) {

servo06.write(j);

delay(speedDelay);

}

}

}

}

}

# TÀI LIỆU THAM KHẢO

<http://arduino.vn/bai-viet/542-gioi-thieu-arduino-mega2560>

<https://www.stdio.vn/dien-tu-ung-dung/gioi-thieu-ve-arduino-mega-2560-j1NjQF>

<https://ohtech.vn/all-courses/lap-trinh-esp32-cam-voi-arduino-ide/lessons/gioi-thieu-esp32-cam-bluetooth-camera-ov2640/>

<https://vidieukhien.xyz/2020/02/08/so-do-chan-esp32-su-dung-cac-chuc-nang-dung-cach-nhat/>

<https://www.dientudat.com/module-bluetooth-hc-05>

<http://damibot.com/san-pham/module-cam-bien/module-bluetooth-hc05-2874.html>

<https://linhkiencnc.vn/?product=dong-co-buoc-nema-17-1-6a-42x40m>

<https://vietmachine.com.vn/cac-loai-dong-co-buoc-step-phan-loai-ung-dung-va-dieu-khien.html>

<http://arduino.vn/bai-viet/181-gioi-thieu-servo-sg90-va-cach-dieu-khien-bang-bien-tro>

<https://iotmaker.vn/dong-co-rc-servo-9g-sg90.html>

<https://dientutuonglai.com/dong-co-servo-sg90-goc-quay-180.html>

<https://www.esshop.vn/mach-dieu-khien-dong-co-buoc-drv8825-kem-tan-nhiet>

<https://iknowvations.in/vi/Arduino/drv8825-stepper-motor-driver-with-arduino-tutorial/>

<http://www.3dmaker.vn/shop/drv8825-pololu-stepper-motor-driver/>

<http://arduino.vn/bai-viet/68-cai-dat-driver-va-arduino-ide>

<https://www.dienmayxanh.com/kinh-nghiem-hay/tim-hieu-he-dieu-hanh-android-la-gi-595278>

<https://bizflycloud.vn/tin-tuc/tim-hieu-he-dieu-hanh-android-la-gi-va-nhung-dieu-thu-vi-nen-biet-ve-android-20200819174936539.htm>

<https://vngiotlab.github.io/vbluno/vi/mydoc_mobile_tut3_vi.html#22-c%C3%A1c-b%C6%B0%E1%BB%9Bc-c%E1%BB%A5-th%E1%BB%83>

<https://uniduc.com/vi/blog/tai-lieu-ve-dong-co-rc-servo-nguyen-ly-va-cau-tao>