

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
THÀNH PHỐ HỒ CHÍ MINH



ĐỒ ÁN TỐT NGHIỆP
NGÀNH CNKT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA

**THIẾT KẾ VÀ CHẾ TẠO ROBOT 4 BẬC TỰ DO TÍCH HỢP
CÔNG NGHỆ XỬ LÝ LÝ ẢNH ĐỂ PHÂN LOẠI LINH KIỆN LỐI**

GVHD: TS. NGUYỄN VĂN THÁI
SVTH: VÕ HOÀNG DUY
NGUYỄN HOÀNG KHA



Tp.Hồ Chí Minh, tháng 8/2022

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO
BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG



ĐỒ ÁN TỐT NGHIỆP
THIẾT KẾ VÀ CHẾ TẠO ROBOT 4 BẬC
TỰ DO TÍCH HỢP CÔNG NGHỆ XỬ LÝ
ẢNH ĐỂ PHÂN LOẠI LINH KIỆN LỒI

GVHD: TS. Nguyễn Văn Thái

SVTH: 1. Võ Hoàng Duy MSSV: 18151058

2. Nguyễn Hoàng Kha MSSV: 18151083

TP. HCM, Tháng 8, Năm 2022

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ và tên sinh viên: Võ Hoàng Duy

MSSV: 18151058

Họ và tên sinh viên: Nguyễn Hoàng Kha

MSSV: 18151083

Ngành: Công nghệ kỹ thuật Điều khiển và Tự động hóa

Lớp: 18151CL1A

Giảng viên hướng dẫn: TS. Nguyễn Văn Thái

Ngày nhận đề tài: 5/3/2022

Ngày nộp đề tài: 6/8/2022

1. Tên đề tài: THIẾT KẾ VÀ CHẾ TẠO ROBOT 4 BẬC TỰ DO TÍCH HỢP CÔNG NGHỆ XỬ LÝ ẢNH ĐỂ PHÂN LOẠI LINH KIỆN LỒI.

Nội dung thực hiện đề tài:

- Thiết kế phần cứng robot trên phần mềm SolidWorks.
- Thiết kế mạch PCB điều khiển robot chân trên phần mềm Eagle.
- Tính toán phương trình động học thuận, động học nghịch cho robot 4 bậc tự do.
- Thiết kế giao diện điều khiển robot trên phần mềm Qt Designer.
- Lập trình động học cho robot 4 bậc tự do.
- Tạo ứng dụng điện thoại điều khiển robot.

2. Sản phẩm:

- Robot 4 bậc tự do.
- Mạch điều khiển cho robot.
- Ứng dụng điều khiển qua điện thoại.

Tp. Hồ Chí Minh, ngày.... tháng.... năm 2022

Tp. Hồ Chí Minh, ngày.... tháng.... năm 2022

TRƯỜNG NGÀNH
(Ký và ghi rõ họ tên)

GIÁO VIÊN HƯỚNG DẪN
(Ký và ghi rõ họ tên)

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên sinh viên: Võ Hoàng Duy

MSSV: 18151058

Họ và tên sinh viên: Nguyễn Hoàng Kha

MSSV: 18151083

Ngành: Công nghệ kỹ thuật Điều khiển và Tự động hóa

Lớp: 18151CL1A

Giáo viên hướng dẫn: TS. Nguyễn Văn Thái

Tên đề tài: THIẾT KẾ VÀ CHẾ TẠO ROBOT 4 BẬC TỰ ĐỘNG TÍCH HỢP CÔNG NGHỆ
XỬ LÝ ẢNH ĐỂ PHÂN LOẠI LINH KIỆN LỖI.

1. Về nội dung đề tài và khối lượng thực hiện: Hoàn thành các nội dung đặt ra của đề tài. Cụ thể là:

- Thiết kế được phần cứng robot.
- Thiết kế được bo mạch điều khiển robot.
- Điều khiển được robot gấp tụ điện thông qua giao diện điều khiển tự thiết kế.
- Thiết kế được ứng dụng điện thoại giám sát hoạt động của robot.

2. Hạn chế:

- Mô hình robot cần cải tiến về thiết kế để nâng cao độ chính xác.
- Ứng dụng xử lý ảnh chỉ có khả năng phân loại được tụ điện.

3: Đề nghị cho bảo vệ hay không: Được bảo vệ.

4. Đánh giá loại:

5. Điểm: **Bằng chữ:**

Tp. Hồ Chí Minh, ngày.... tháng.... năm 2022

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

Họ và tên sinh viên: Võ Hoàng Duy

MSSV: 18151058

Họ và tên sinh viên: Nguyễn Hoàng Kha

MSSV: 18151083

Ngành: Công nghệ kỹ thuật Điều khiển và Tự động hóa

Lớp: 18151CL1A

Giáo viên phản biện:

Tên đề tài: THIẾT KẾ VÀ CHẾ TẠO ROBOT 4 BẬC TỰ DO TÍCH HỢP CÔNG NGHỆ XỬ LÝ ẢNH ĐỂ PHÂN LOẠI LINH KIỆN LỒI.

1. Về nội dung đề tài và khối lượng thực hiện:

.....
.....

2. Ưu điểm:

.....
.....

3. Khuyết điểm:

.....
.....

4: Đề nghị cho bảo vệ hay không?

.....
.....

5. Câu hỏi phản biện:

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

6. Điểm: **Bằng chữ:**

Tp. Hồ Chí Minh, ngày.... tháng.... năm 2022

GIÁO VIÊN PHẢN BIỆN

(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Đối với một sinh viên sắp ra trường, bước vào cuộc sống thì đồ án là một bước nhảy quan trọng. Đồ án tốt nghiệp với đề tài **“Thiết kế và chế tạo robot 4 bậc tự do tích hợp công nghệ xử lý ảnh để phân loại linh kiện lỗi”** là kết quả của quá trình cố gắng không ngừng nghỉ của nhóm em và nhận được sự hướng dẫn tận tình của Thầy - TS. Nguyễn Văn Thái cùng các anh chị và bạn bè trong 3DVisionLab. Qua đây, nhóm em xin gửi lời cảm ơn chân thành tới những thành viên đã giúp đỡ nhóm em hoàn thành được đồ án này.

Nhóm em xin tỏ lòng kính trọng và biết ơn sâu sắc đến Thầy - TS. Nguyễn Văn Thái là người trực tiếp hướng dẫn đồ án. Thầy đã tận tình giúp đỡ, hướng dẫn đồ án tốt nghiệp, giải đáp thắc mắc, tận tình giúp đỡ, động viên, khích lệ chúng em, tạo mọi điều kiện tốt nhất từ bố trí phòng thí nghiệm đến các thiết bị vật tư trong suốt quá trình nghiên cứu, thực hiện đề tài.

Nhóm em xin gửi lời cảm ơn đến các quý thầy cô trong Khoa Đào Tạo Chất Lượng Cao nói chung và các thầy cô ngành Điều Khiển và Tự Động Hóa đã nhiệt tình giảng dạy cho chúng em kiến thức về các môn đại cương cũng như các môn chuyên ngành, giúp em có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ chúng em trong suốt bốn năm học tập tại đây. Đặc biệt, nhóm em xin cảm ơn quý Thầy, Cô trong Hội đồng đã dành thời gian nhận xét, góp ý để đồ án của chúng em được hoàn thiện hơn.

Xin cảm ơn gia đình và bạn bè luôn động viên, ủng hộ và giúp đỡ chúng em những lúc chúng em gặp khó khăn trong việc làm đồ án tốt nghiệp, để chúng em hoàn thành được đồ án như ngày hôm nay.

Mặc dù đã rất cố gắng thực hiện đề tài một cách hoàn chỉnh nhất. Nhưng do chưa được làm quen nhiều với công tác nghiên cứu khoa học, tiếp xúc thực tế với các thiết bị, cũng như những hạn chế về kiến thức và kinh nghiệm nên không thể tránh khỏi những thiếu sót. Nhóm em rất mong nhận được sự góp ý của quý Thầy – Cô và các bạn để khóa luận được hoàn chỉnh hơn.

Nhóm em xin chân thành cảm ơn!

LỜI CAM ĐOAN

Nhóm em xin cam đoan đê tài “*Thiết kế và chế tạo robot 4 bậc tự do tích hợp công nghệ xử lý ảnh để phân loại linh kiện lỗi*” là nhóm em tự thực hiện, dựa vào tham khảo một số tài liệu trước đó, dưới sự hỗ trợ giúp đỡ của Thầy - TS Nguyễn Văn Thái và không sao chép từ tài liệu hay công trình nào khác.

Tp. Hồ Chí Minh, ngày.... tháng.... năm 2022

Sinh viên thực hiện

(Ký và ghi rõ họ tên)

Võ Hoàng Duy

Nguyễn Hoàng Kha

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN	1
1.1. Đặt vấn đề	1
1.2. Mục tiêu đề tài.....	1
1.3. Nội dung nghiên cứu:.....	1
1.3. Giới hạn đề tài	1
1.4. Nội dung đề tài	2
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	3
2.1. Cơ sở lý thuyết về robot.....	3
2.1.1. Sơ lược về sự hình thành và phát triển robot.....	3
2.1.1. Robot công nghiệp	3
2.1.3. Động học robot	4
2.1.4. Bộ thông số Denavit – Hartenberg (DH)	5
2.2. Thiết kế ứng dụng điều khiển và giám sát robot.....	6
2.2.1. Android Studio.....	6
2.2.2. Cơ sở dữ liệu Firebase	8
2.2.3. Firebase Authentication	9
2.2.4 Firebase Realtime Database.....	10
2.3. Lý thuyết thiết kế giao diện GUI điều khiển ROBOT	12
2.3.1. Qt Designer	12
2.4. Lý thuyết xử lý ảnh	13
2.4.1. Giới thiệu về hệ thống xử lý ảnh	13
2.4.2. Điểm ảnh.....	14
2.4.3. Độ phân giải của ảnh	14
2.4.4. Tổng quan về tiền xử lý ảnh	15
2.4.5. Phân đoạn ảnh.....	15
2.4.4. Phân loại ảnh.....	16
2.4.5 Giới thiệu về YOLO	17
2.4.6 Kiến trúc mạng YOLO	17
2.5. Lý thuyết thiết kế mạch cầu H điều khiển động cơ	18
2.6. Bộ điều khiển PID	20
CHƯƠNG 3: THIẾT KẾ VÀ TÍNH TOÁN ROBOT	22
3.1. Thiết kế cơ khí robot 4 bậc tự do trên phần mềm Solidworks	22

3.1.1. Giới thiệu phần mềm Solidworks	22
3.1.2. Vật liệu thiết kế và ý tưởng mô hình	22
3.2. Thiết kế mạch điện tử.....	28
3.2.1. Giới thiệu Arduino.....	28
3.2.2. Giới thiệu ESP32-WROOM-32E	30
3.2.3. Các linh kiện khác	31
3.2.3 Thiết kế mạch điều khiển robot:	37
3.2.4 Mạch ESP32 truyền dữ liệu:	40
3.3. Bài toán động học thuận robot	44
3.4. Bài toán động học nghịch robot	46
3.4. Mô phỏng kiểm chứng chứng trên matlab	50
CHƯƠNG 4: THIẾT KẾ ỨNG DỤNG VÀ GIAO DIỆN ĐIỀU KHIỂN, GIÁM SÁT ROBOT	54
4.1. Thiết kế ứng dụng điều khiển và giám sát Robot	54
4.1.1. Hướng dẫn sử dụng cơ sở dữ liệu Firebase	54
4.1.2. Tổng quan giao diện ứng dụng	62
4.2. Thiết kế giao diện điều khiển và giám sát Robot trên phần mềm Qt Designer	69
4.2.1 Home page	69
4.2.2 Setting page.....	69
4.2.3 Monitor page.....	70
4.2.3 Error page	71
4.2.4 Image process page.....	71
4.2.5 Information page.....	72
CHƯƠNG 5: XỬ LÝ ẢNH PHÁT HIỆN LỖI LINHKIỆN ĐIỆN TỬ SỬ DỤNG YOLOV5	73
5.1. Hướng dẫn training dữ liệu bằng YOLOv5	73
5.1.1. Chuẩn bị dữ liệu.....	73
5.1.2. Huấn luyện mô hình.....	79
5.2. Kết quả huấn luyện mô hình YOLOv5	82
5.3. Phát hiện tọa độ	83
5.3. Lưu đồ điều khiển robot 4 bậc tự do	85
CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	87
6.1. Kết luận	87
6.1.1. Những công việc đã thực hiện	87

6.1.2. Những mặt hạn chế	87
6.2. Hướng phát triển	88
TÀI LIỆU THAM KHẢO	89

DANH MỤC BẢNG BIỂU

Bảng 2. 1. Bảng thông số robot 4 bậc tự do	44
Bảng 2. 2. Bảng thông số D-H của robot	45
Bảng 3. 1. Danh sách linh kiện sử dụng làm robot.....	27
Bảng 3. 2. Bảng thông số bo Arduino Mega.	29
Bảng 3. 3. Danh sách linh kiện sử dụng làm mạch điều khiển động cơ robot	40
Bảng 3. 4. Thông số đặt các khớp trường hợp 1	50
Bảng 3. 5. Kết quả tính động học thuận trường hợp 2	51
Bảng 3. 6. Vị trí đặt px , py , pz của trường hợp 1.....	52
Bảng 3. 7. Vị trí đặt px , py , pz của trường hợp 2.....	53
Bảng 4. 1. Mô tả hoạt động chức năng đăng nhập	63
Bảng 4. 2. Mô tả hoạt động chức năng tạo tài khoản.	64

DANH MỤC HÌNH VẼ

Hình 2. 1. Mô hình robot 4 bậc tự do	4
Hình 2. 2. Góc xoắn và chiều dài của một khâu.....	5
Hình 2. 3. Các thông số của khâu: θ , d, a và α	6
Hình 2. 4. Cấu trúc Android Studio.....	7
Hình 2. 5. Các chức năng của cơ sở Firebase.....	8
Hình 2. 6. Giao diện Firebase	9
Hình 2. 7. Giá trị ID, thiết lập và cấu hình của SDK	9
Hình 2. 8. Hình ảnh các tính năng Firebase Authentication.....	10
Hình 2. 9. Hình ảnh Firebase Realtime Database.....	11
Hình 2. 10. Phần mềm PyQt	12
Hình 2. 11 Giao diện thiết kế Qt Designer	13
Hình 2. 12. Một điểm ảnh hay pixel ảnh	14
Hình 2. 13. Độ phân giải của ảnh	15
Hình 2. 14. Ảnh màu- Ảnh xám- Ảnh nhị phân	16
Hình 2. 15. Kiến trúc mạng YOLO	17
Hình 2. 16. Sơ đồ chân của IC IR2184.....	18
Hình 2. 17. Biểu đồ tín hiệu của IC IR2184.....	19
Hình 2. 18. Sơ đồ kết nối điển hình IC IR2184.....	20
Hình 2. 19. Bộ điều khiển PID	21
Hình 3. 1. Phôi nhôm trước khi CNC.....	22
Hình 3. 2. Thiết kế cụm đế của robot trên SolidWorks.....	23
Hình 3. 3. Link 1 của robot.....	23
Hình 3. 4. Thiết kế cụm bạc đạn và truyền động của robot.....	24
Hình 3. 5. Thiết kế chốt lắp động cơ trên link 1.....	24
Hình 3. 6. Thiết kế link 2 của robot.....	25
Hình 3. 7. Thiết kế link 3 của robot.....	25
Hình 3. 8. Thiết kế 3 khớp 1,2,3 của robot trên SolidWorks.	25
Hình 3. 9. Thiết kế khớp 4 của robot trên SolidWorks.	26
Hình 3. 10. Thiết kế robot 4 bậc tự do hoàn chỉnh trên SolidWorks.	26
Hình 3. 11. Mô hình robot 4 bậc tự do thực tế	27
Hình 3. 12. Giao diện làm việc của phần mềm lập trình Arduino IDE 1.8.15	28
Hình 3. 13. Bo Arduino Mega	29
Hình 3. 14. ESP32-WROOM-32E.	30
Hình 3. 15. Sơ đồ chân ESP32-WROOM-32E.	31
Hình 3. 16. Sơ đồ chân IC IR2184.	31
Hình 3. 17. Sơ đồ chân IC IRFR3708.	32
Hình 3. 18. Sơ đồ chân IC IRFR3708.	32
Hình 3. 19. Sơ đồ chân IC LM7805.	33
Hình 3. 20. Sơ đồ chân IC AMS1117-3.3V.	33

Hình 3. 21. Động cơ DC Servo JGB37-545 DC	34
Hình 3. 22. Động cơ DC Servo JGB37-520	34
Hình 3. 23. Động cơ DC Servo JGY370	35
Hình 3. 24. Cảm biến quang chữ U	36
Hình 3. 25. Kích thước Servo MG90S.	36
Hình 3. 26. IC FT232	36
Hình 3. 27. Sơ đồ khói mạch điện điều khiển robot.....	37
Hình 3. 28. Sơ đồ nguyên lý của shield điều khiển động cơ DC	37
Hình 3. 29. Sơ đồ nguyên lý 1 cụm cầu H điều khiển robot.	38
Hình 3. 30. Sơ đồ mạch hạ áp DC-DC LM7805-5V	38
Hình 3. 31. Layout PCB mặt Top.....	39
Hình 3. 32. Layout PCB mặt Bottom	39
Hình 3. 33. Lắp và hàn linh kiện lên mạch và thử nghiệm, kiểm tra mạch.....	39
Hình 3. 34. Sơ đồ nguyên lý mạch ESP32	40
Hình 3. 35. Khối MCU sử dụng chip ESP32-WROOM-32E	41
Hình 3. 36. Khối mạch kích relay.....	41
Hình 3. 37. Khối mạch nạp code ESP32 và nguồn	42
Hình 3. 38. Layout PCB mặt Top.....	42
Hình 3. 39. Layout PCB mặt Bottom	42
Hình 3. 40. Lắp và hàn linh kiện lên mạch và thử nghiệm, kiểm tra mạch.....	43
Hình 3. 41. Đặt hệ trục tọa độ tối ưu cho robot.....	44
Hình 3. 42. Phép chiếu trên mặt phẳng Oxy	47
Hình 3. 43. Hình chiếu của robot trường hợp $\theta1_1 = atan2(py, px)$	48
Hình 3. 44. Hình chiếu của robot trường hợp $\theta1_1 = atan2(Py, Px) - \pi$	49
Hình 3. 45. Kết quả tính động học thuận trường hợp 1	51
Hình 3. 46. Hình vẽ robot động học thuận trường hợp 1	51
Hình 3. 47. Kết quả tính động học thuận trường hợp 2.....	51
Hình 3. 48. Kết quả hình vẽ robot trường hợp 2	52
Hình 3. 49. Các bộ nghiệm robot của trường hợp 1	52
Hình 3. 50. Dùng động học thuận để kiểm chứng các bộ nghiệm trường hợp 1	53
Hình 3. 51. Dùng động học thuận để kiểm chứng các bộ nghiệm trường hợp 2	53
Hình 4. 1. Thêm firebase cho project	54
Hình 4. 2. Chọn Email and Password Authentication trong mục Authentication.....	54
Hình 4. 3. Tiến hành liên kết mới firebase	55
Hình 4. 4. Bật Firebase Authentication	55
Hình 4. 5. Chuyển nút nhấn trạng thái sang Enable để tiến hành bật.....	56
Hình 4. 6. Truy cập vào trang chủ Firebase	56
Hình 4. 7. Add project để bắt đầu dự án mới	57
Hình 4. 8. Đặt tên cho project.....	57
Hình 4. 9. Nhấn continue để chuyển sang trang mới	58

Hình 4. 10. Chọn Default Account for Firebase và Create project	58
Hình 4. 11. Nhấn Continue để hoàn thành	59
Hình 4. 12. Giao diện fire base	59
Hình 4. 13. Create Realtime Database.....	60
Hình 4. 14. Đổi chế độ mặc định và bấm Next.....	60
Hình 4. 15. Giao diện của Database chưa có dữ liệu	61
Hình 4. 16. Tạo các dữ liệu chứa các vị trí và các góc theta của robot.....	61
Hình 4. 17. Hình ảnh màn hình giới thiệu	62
Hình 4. 18. Màn hình đăng nhập	62
Hình 4. 19. Màn hình tạo tài khoản	63
Hình 4. 20. Hình ảnh màn hình điều hướng	64
Hình 4. 21. Hình ảnh màn hình Home.....	65
Hình 4. 22. Hình ảnh màn hình Monitor	66
Hình 4. 23. Màn hình control	66
Hình 4. 24. Màn hình history.....	67
Hình 4. 25. Màn hình Register.	67
Hình 4. 26. Màn hình Change Password.	68
Hình 4. 27. Hình ảnh màn hình Delete Account.	68
Hình 4. 28. Hình ảnh Home page.	69
Hình 4. 29. Hình ảnh setting page.	70
Hình 4. 30. Hình ảnh Monitor Page.	70
Hình 4. 31. Hình ảnh Error Page.	71
Hình 4. 32. Hình ảnh Image process page.....	71
Hình 4. 33. Hình ảnh Information page.....	72
Hình 5. 1. Tạo project mới	73
Hình 5. 2. Đặt tên cho project mới.	73
Hình 5. 3. Nhấn Upload Image để thêm ảnh muốn label.	74
Hình 5. 4. Tiến hành upload ảnh lên web.....	74
Hình 5. 5. Nhấn Start annotating để tiến hành label cho project.....	75
Hình 5. 6. Chọn và gắn label cho các object muốn phân loại	75
Hình 5. 7. Đặt tên cho dữ liệu đã được label.....	76
Hình 5. 8. Export data.....	76
Hình 5. 9. Đặt tên và tạo export data.....	77
Hình 5. 10. Tiến hành tải data về máy.....	77
Hình 5. 11. Dữ liệu của ảnh sau khi label	78
Hình 5. 12. File .txt dữ liệu sau khi chuyển đổi	78
Hình 5. 13. Tạo folder chứa data	79
Hình 5. 14. Giao diện Google Colab	79
Hình 5. 15. Repo YOLOv5 được lưu trong thư mục content.....	80
Hình 5. 16. Bấm chọn biểu tượng drive để kết nối	80
Hình 5. 17. Ô chứa mã thực thi kết nối drive	80

Hình 5. 18. Thư mục drive được tạo trong thư mục content.....	80
Hình 5. 19. Thông tin trong file cấu hình capacitor.yaml	81
Hình 5. 20. Thông số cấu hình để huấn luyện mô hình.....	81
Hình 5. 21. Ô chứa lệnh thực thi quá trình huấn luyện.	81
Hình 5. 22. Quá trình huấn luyện mô hình	82
Hình 5. 23. Đồ thị biểu diễn các giá trị hàm mát mẻ trong quá trình huấn luyện.....	82
Hình 5. 24. Kết quả nhận diện	83
Hình 5. 25. Các bit nhị phân	84
Hình 5. 26. Xác định vị trí vật	84
Hình 5. 27. Kết quả sau khi chuyển đổi ảnh sang vị trí thực	85
Hình 5. 28. Lưu đồ điều khiển robot	85

TÓM TẮT

Dựa vào các kiến thức đã học ở trường về điều khiển tự động và những sự tìm hiểu về ngành công nghiệp robot. Chúng em nhận thấy nghiên cứu robot là một trong những lĩnh vực được quan tâm nhất ngày nay. Robot ngày càng được sử dụng nhiều trong sản xuất do tính ưu việt của nó mang so với sử dụng nhân công. Từ xu hướng thực tế đó, nhóm chúng em đã quyết định chọn đề tài đồ án tốt nghiệp liên quan đến robot và ứng dụng của robot trong sản xuất. Đề tài **“Thiết kế và chế tạo robot 4 bậc tự do tích hợp công nghệ xử lý ảnh để phân loại linh kiện lỗi”** được thực hiện từ tháng 2/2022 đến tháng 8/2022 tại 3DVisionLab. Mục tiêu của nhóm là thiết kế và chế tạo một robot 4 bậc tự do nhỏ gọn và có thể hoạt động linh hoạt được điều khiển thông qua giao diện điều khiển trên máy tính. Robot được điều khiển bằng board Arduino Mega kết hợp với board ESP 32. Lực chuyển động cho các khớp được cung cấp bởi 4 động cơ DC có hộp số tích hợp sẵn encoder để điều khiển chính xác vị trí góc quay. Robot được lập trình để tính toán động học sau đó đưa cơ cấu tay kẹp đến vị trí mong muốn để thực hiện các công việc được lập trình.

Sơ lược quá trình thực hiện đề tài bao gồm:

Đầu tiên, nhóm thiết kế mô hình robot 4 bậc tự do trên phần mềm SolidWorks. Mô hình robot của nhóm được tính toán thiết kế để sử dụng những vật liệu có bán sẵn trên thị trường khu vực thành phố Hồ Chí Minh để rút ngắn thời gian thi công mô hình robot đồng thời tạo sự chủ động, thuận lợi trong việc sửa chữa robot khi bị hư hỏng.

Sau khi hoàn tất việc thi công mô hình robot 4 bậc tự do, nhóm đã sử dụng phần mềm Autodesk Eagle để thiết kế vă mạch shield cầu H dùng để điều khiển 4 động cơ và 1 servo của robot. Cùng với đó là thiết kế thêm 1 mạch ESP32 để truyền nhận dữ liệu điều khiển robot hoạt động.

Tiếp theo, nhóm tính toán động học thuận và động học nghịch cho robot, áp dụng và xử lý để phân loại linh kiện điện tử bị lỗi. Đồng thời, nhóm viết ứng dụng giám sát hoạt động robot thông qua kết nối wifi trên điện thoại.

CHƯƠNG 1: TỔNG QUAN

1.1. Đặt vấn đề

Trong cuộc cách mạng công nghiệp 4.0, với sự phát triển vượt bậc của khoa học kỹ thuật về máy móc nói chung và về robot nói riêng. Trong từng giai đoạn phát triển, robot không chỉ hỗ trợ những nhu cầu của con người, mà còn thực hiện những hành động, cử chỉ thông minh như một con người thực sự. Việc sở hữu tốc độ nhanh và trí thông minh nhân tạo, robot có thể thực hiện những công việc nguy hiểm đòi hỏi tốc độ hoặc những công việc đòi hỏi độ chính xác cao. Ngành công nghiệp tự động hóa với robot đang là xu hướng phát triển của các nhà máy hiện nay, bên cạnh đó lĩnh vực công nghiệp thì robot cũng có vai trò vô cùng to lớn trong các lĩnh vực khác như y tế, giáo dục đào tạo và an ninh quốc phòng.

Trong những năm gần đây, xử lý ảnh đang được ứng dụng rộng rãi và phổ biến trên toàn thế giới. Nhờ các hệ thống xử lý ảnh con người đã giảm phần lớn khối lượng công việc cũng như tiết kiệm thời gian và tăng độ chính xác trong công việc. Vì thế nên nhóm đã nghiên cứu và chọn đề tài: **“Thiết kế và chế tạo robot 4 bậc tự do tích hợp công nghệ xử lý ảnh để phân loại linh kiện lỗi”**. Với đề án này, chúng em hướng tới việc xây dựng nên một cánh tay robot 4 bậc tự do và sau khi đã xây dựng được mô hình robot nhóm chúng em sẽ tiến hành điều khiển robot hoạt động tại những vị trí tùy chỉnh.

1.2. Mục tiêu đề tài

Mục tiêu nghiên cứu của đề tài là chế tạo thành công một robot 4 bậc tự do có khả năng nhận diện và phân loại lỗi của linh kiện điện tử.

1.3. Nội dung nghiên cứu:

Robot 4 bậc tự do là 1 hệ thống cần có sự phối hợp giữa 3 phần: cơ khí, mạch điện và điều khiển, do đó đề tài sẽ gồm những nội dung nghiên cứu sau:

- Thiết kế và chế tạo robot 4 bậc tự do.
- Thiết kế và chế tạo mạch cầu H điều khiển 4 động cơ và mạch ESP32 trên phần mềm Eagle.
- Nghiên cứu phương pháp tính động học thuận, động học nghịch của robot 4 bậc tự do.
- Thiết kế và phát triển ứng dụng xử lý ảnh để phân loại linh kiện lỗi.
- Thiết kế app điện thoại để giám sát hoạt động của robot.

1.3. Giới hạn đề tài

Đề tài có những giới hạn như sau:

- Robot 4 bậc tự do với góc xoay các bậc 2,3,4 bị giới hạn do thiết kế cơ khí nên nên sẽ có những vị trí mà robot không thể đưa tay gấp đến. Mặc dù đã khắc phục toàn toàn độ rơ tại khi kết nối các link lại với nhau nhưng robot vẫn còn bị rơ do hộp số động cơ. Điều này đã làm ảnh hưởng đến độ chính xác của robot khi hoạt động do đó robot chỉ có thể gấp những linh kiện có kích thước to như tụ điện.

- Mô hình nhận dạng xử lý ảnh chỉ nhận dạng được những dấu hiệu lỗi sai rõ ràng, chưa thể nhận ra những lỗi sai nhỏ.

1.4. Nội dung đề tài

Đề “*Thiết kế và chế tạo robot 4 bậc tự do tích hợp công nghệ xử lý ảnh để phân loại linh kiện lỗi*” sẽ được trình bày theo từng phần từ lý thuyết đến ứng dụng. Mỗi vấn đề sẽ được trình bày qua từng chương sau:

Chương 1: Tổng quan

Trình bày tổng quan sơ bộ về các yêu cầu của cuốn báo cáo như đặt vấn đề, mục tiêu, giới hạn và nội dung đề tài.

Chương 2: Cơ sở lý thuyết

Trình bày phương pháp tính động học thuận, động học nghịch, điều khiển robot và lý thuyết xử lý ảnh, giới thiệu cơ bản về mạng YOLO.

Chương 3: Thiết kế và tính toán robot

Trình bày về thiết kế cơ khí robot 4 bậc tự do trên SolidWorks và thiết kế mạch điện cầu H và mạch ESP32 trên phần mềm Autodesk Eagle. Tính toán động học thuận và nghịch cho robot.

Chương 4: Thiết kế ứng dụng và giao diện điều khiển, giám sát robot

Trình bày thiết kế giao diện điều khiển robot trên phần mềm Qt Designer.

Chương 5: Xử lý ảnh phát hiện lỗi linh kiện

Trình bày quá trình tìm hiểu và ứng dụng YOLOv5 để phân loại tụ điện bị lỗi.

Chương 6: Kết luận và hướng phát triển

Trình bày kết quả mà nhóm đã đạt được trong quá trình nghiên cứu, những hạn chế và đưa ra hướng phát triển cho đề tài.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Cơ sở lý thuyết về robot

2.1.1. Sơ lược về sự hình thành và phát triển robot

Lịch sử của tự động hóa công nghiệp được đặc trưng bởi các giai đoạn thay đổi nhanh chóng bằng các phương pháp phổ biến, những giai đoạn thay đổi như vậy gần như gắn chặt với kinh tế thế giới. Robot công nghiệp dường như trở thành thiết bị hiện đại nhất vào những năm 60 của thế kỷ trước, cùng với hệ thống thiết kế có sự hỗ trợ của máy tính (CAD) và hệ thống sản xuất có sự hỗ trợ của máy tính (CAM) đã góp phần phát triển xu hướng mới nhất trong quá trình sản xuất và tự động hóa [1].

Quá trình phát triển robot đạt một số thành tựu nổi bật như sau:

- Robot Humanoid (giống người) được triển lâm vào năm 1939 và 1940 tại hội chợ thế giới.
- Sự phát triển của công nghiệp điện tử dẫn đến sự ra đời của robot tự động điện tử đầu tiên, được tạo bởi William Grey Walter ở Bristol, Anh năm 1948.
- Năm 1956, Robot thương mại đầu tiên do công ty Unimation được thành lập bởi George Devol và Joseph Engelberger, dựa trên các phát minh của Devol.
- Năm 1961, robot công nghiệp được đưa vào ứng dụng đầu tiên ở một nhà máy ô tô của General Motors tại Trenton, New Jersey, Hoa Kỳ.
- Năm 1974, Robot công nghiệp được điều khiển bởi vi máy tính đầu tiên, IRB 6 do công ty ASEA sản xuất, được giao cho một công ty cơ khí nhỏ ở miền nam Thụy Điển. Thiết kế của robot này đã được cấp bằng sáng chế đã được năm 1972.

Ngày nay, ngành công nghệ kỹ thuật robot có thể được mô tả như một ngành khoa học xử lý chuyển động thông minh của các robot có cơ chế khác nhau và ứng dụng ngày càng rộng rãi trong công nghiệp, y tế, giáo dục đào tạo, giải trí và đặc biệt trong an ninh quốc phòng thì việc phát triển robot và các ngành dịch vụ kéo theo sẽ đem đến một lợi ích vô cùng to lớn.

2.1.1. Robot công nghiệp

Robot công nghiệp có thể được hiểu là những thiết bị tự động linh hoạt, bắt chước được các chức năng lao động công nghiệp của con người.

Hiện nay, các quốc gia như Mỹ, Nhật Bản và một số nước châu Âu là các nước đi đầu trong lĩnh vực công nghiệp Robot. Robot được sử dụng tại nhiều quốc gia, trong hầu như mọi lĩnh vực và ngày càng được mở rộng. Ở nước ta, ứng dụng của robot công nghiệp rất đa dạng, tùy vào những ngành, công việc khác nhau mà ta có thể áp dụng những robot

công nghiệp riêng biệt. Dưới đây là một số ngành trong hệ thống sản xuất mà áp dụng robot công nghiệp [1]:

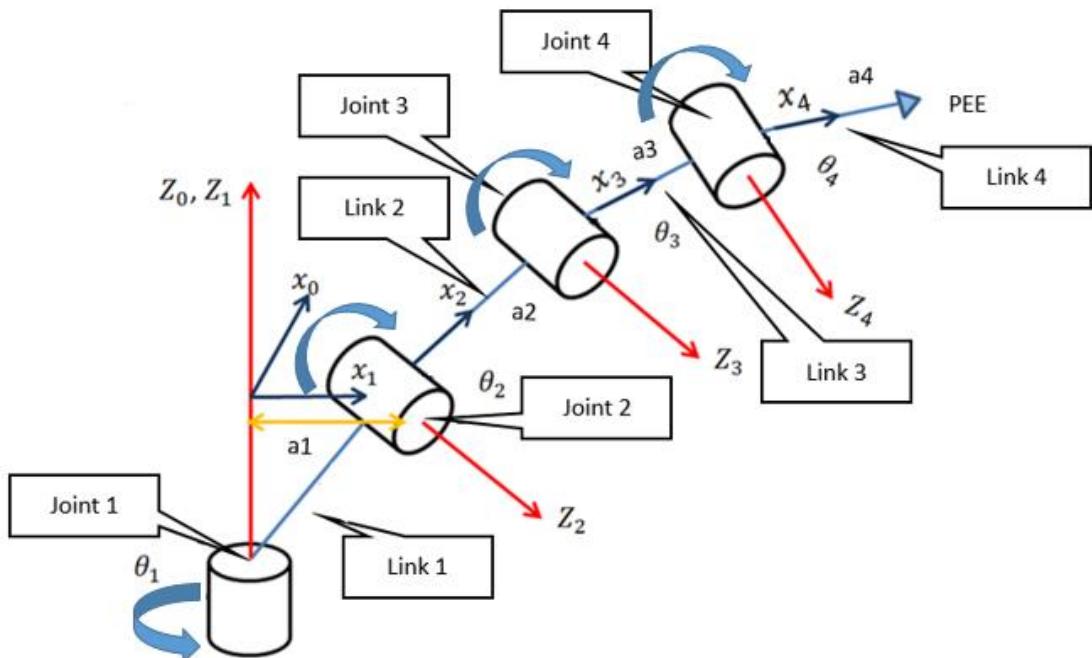
- Robot công nghiệp trong công nghiệp đúc – rèn: Công việc rất đa dạng với điều kiện làm việc khắc nghiệt, sản phẩm luôn thay đổi và đặc biệt là chất lượng của sản phẩm phụ thuộc quá nhiều vào quá trình thao tác.

- Robot công nghiệp trong công nghệ gia công và lắp ráp: Độ chính xác định vị và thời gian là yếu tố quan trọng nhất khi thiết kế các robot lắp ráp, các dây chuyền đạt mức độ tự động hóa cao, tự động hoàn toàn không có sự tham gia của con người.

- Robot công nghiệp trong nhà máy sản xuất: Trong các nhà máy sản xuất lớn, những robot này là những hệ thống được tự động hóa hoàn toàn, chúng có khả năng thực hiện chính xác các công việc của mình mà hầu như không cần sự giúp đỡ của con người. Chỉ với vài người giám sát công việc, các máy móc này có thể hoạt động suốt ngày đêm.

2.1.3. Động học robot

Động học của robot rất cần thiết để mô tả vị trí, hướng cũng như chuyển động của tất cả khớp nối, cùng với mô hình động lực học là yếu tố quan trọng để phân tích và tổng hợp các hành động của robot. Nghiên cứu về động học của robot là bước quan trọng để điều khiển sự chuyển động của robot. Bất kỳ một robot nào cũng có thể coi là một tập hợp các khâu (links) gắn liền với các khớp (joints) [2].



Hình 2. 1. Mô hình robot 4 bậc tự do

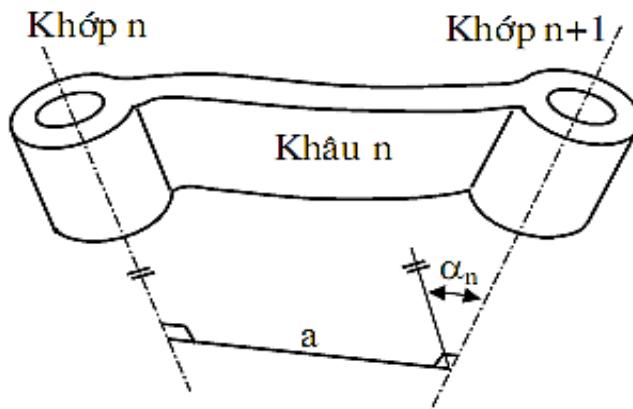
Ta đặt trên mỗi khâu của robot một hệ tọa độ. Sử dụng các phép biến đổi thuần nhất có thể mô tả vị trí tương đối và hướng giữa các hệ tọa độ này bằng các bài toán động học. Động học là một phần của cơ học cổ điển nghiên cứu các đặc trưng của chuyển động mà không tính đến nguyên nhân gây ra chúng như lực và mô men. Khoa học động học nghiên cứu về quỹ đạo, vận tốc, gia tốc. Do đó, trong lĩnh vực chế tạo robot chủ yếu quan tâm đến quỹ đạo và vận tốc, vì cả hai đều được đo bởi các cảm biến. Trong các khớp của robot, các quỹ đạo được đo dưới dạng góc trong khớp quay hoặc khoảng cách trong một khớp tịnh tiến. Tính toán động học và điều khiển robot liên quan đến hướng và vị trí của khâu chấp hành cuối cùng bởi sự ràng buộc của các khớp. Động học thuận và động học nghịch chính là hai nội dung trong bài toán động học. Bài toán động học thuận tức là từ các góc quay của các khớp robot ta sẽ tính ra được vị trí điểm đầu cuối. Với bài toán động học nghịch thì ngược lại, từ vị trí điểm đầu cuối của robot ta tìm được góc quay của các khớp cho phù hợp.

2.1.4. Bộ thông số Denavit – Hartenberg (DH)

Nhóm em lựa chọn và sử dụng thuật toán Denavit – Hartenberg để tìm động học của robot 4 bậc tự do.

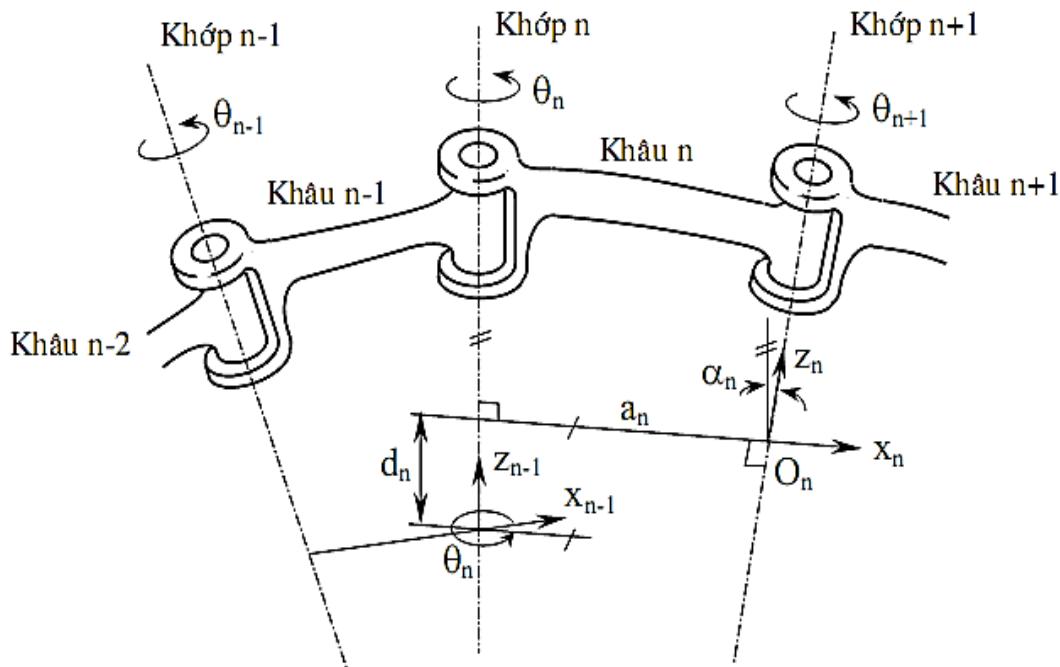
Một robot gồm nhiều khâu cấu thành từ các khâu nối tiếp nhau thông qua các khớp động. Gốc chuẩn (Base) của một robot là khâu số 0 và không tính vào số các khâu. Khâu 1 nối với khâu chuẩn bởi khớp 1 và không có khớp ở đầu mút của khâu cuối cùng. Bộ thông số D-H gồm có 4 thông số [2]:

- Độ dài pháp tuyến chung: a_n – là chiều dài đường vuông góc chung của 2 trục z.
- Góc xoắn hay góc giữa các trục trong mặt phẳng vuông góc với a_n : α_n



Hình 2. 2. Góc xoắn và chiều dài của một khâu

- Khoảng cách giữa các pháp tuyến đo dọc theo trục khớp n: d_n .
- Góc giữa các pháp tuyến đo trong mặt phẳng vuông góc với trục: θ_n .



Hình 2. 3. Các thông số của khâu: θ , d , a và α .

Hệ tọa độ của các khâu được gắn lên dựa theo nguyên tắc sau [2]:

- Tại khâu thứ n, gốc của hệ tọa độ được gắn tại giao điểm của khớp thứ n+1 với pháp tuyến a_n . Nếu hai trục cắt nhau, điểm cắt của hai trục sẽ là nơi đặt gốc tọa độ. Trường hợp hai trục song song nhau, gốc tọa độ sẽ được đặt tại điểm thích hợp ở trên trục khớp của khâu kế tiếp.

- Trục z_n của hệ tọa độ sẽ được gắn lên khâu thứ n đặt dọc theo trục khớp thứ n+1.
- Trục x_n thường được đặt hướng từ trục khớp n đến trục khớp n+1 và dọc theo pháp tuyến chung (đường vuông góc chung).

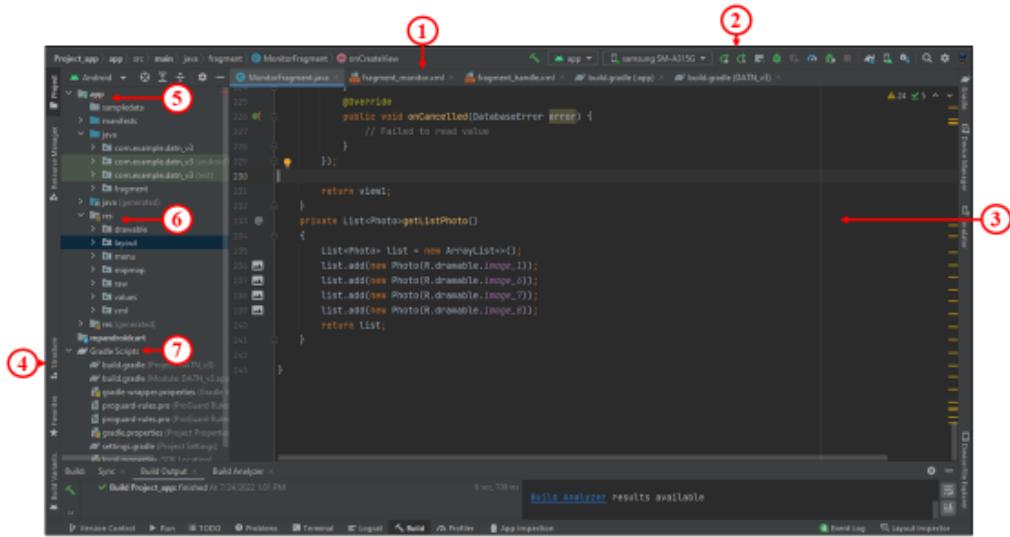
2.2. Thiết kế ứng dụng điều khiển và giám sát robot

2.2.1. Android Studio

Android Studio là môi trường phát triển tích hợp (IDE) chính thức để phát triển ứng dụng Android, dựa trên IntelliJ IDEA. Dựa trên các trình soạn thảo mã và công cụ phát triển mạnh mẽ của IntelliJ, Android Studio còn cung cấp thêm nhiều tính năng giúp bạn nâng cao năng suất khi xây dựng ứng dụng Android, chẳng hạn như [3]:

- Một hệ thống xây dựng linh hoạt dựa trên Gradle.
- Một trình mô phỏng nhanh và nhiều tính năng.
- Một môi trường hợp nhất nơi bạn có thể phát triển cho mọi thiết bị Android.
- Áp dụng thay đổi để đẩy mã và tài nguyên cần thay đổi vào ứng dụng đang chạy mà không cần khởi động lại ứng dụng.

- Các ví dụ mẫu và tích hợp GitHub để giúp bạn xây dựng các tính năng ứng dụng phổ biến.
- Đa dạng khung và công cụ thử nghiệm.



Hình 2. 4. Cấu trúc Android Studio

Cấu trúc Android Studio gồm [3]:

- Navigation bar (thanh điều hướng): giúp bạn điều hướng trong dự án và mở các tệp để chỉnh sửa. Công cụ này mang một cấu trúc nhỏ gọn hơn cấu trúc bạn thấy trong cửa sổ project.
- Toolbar (thanh công cụ): cho phép bạn thực hiện nhiều thao tác, bao gồm cả việc chạy ứng dụng của bạn và chạy các công cụ Android.
- Editor window (cửa sổ trình chỉnh sửa): là nơi bạn tạo và sửa đổi mã. Tuỳ thuộc vào loại tệp hiện tại, trình chỉnh sửa có thể thay đổi. Ví dụ: khi bạn xem tệp bố cục, trình chỉnh sửa sẽ xuất hiện dưới dạng layout editor (trình chỉnh sửa bố cục).
- Tool window bar (thanh cửa sổ công cụ): nằm xung quanh bên ngoài cửa sổ IDE và chứa các nút cho phép bạn mở rộng hoặc thu gọn từng cửa sổ công cụ riêng lẻ.
- Thư mục app: đây là thư mục chứa toàn bộ project. Gồm có thư mục manifests chứa file AndroidManifest.xml, thư mục java chứa các file mã nguồn Java và thư mục res chứa các file layout, xml, giao diện người dùng, ảnh, video,...
- Thư mục res: đây là thư mục chứa tài nguyên của project, thư mục này chứa các file giao diện, hình ảnh mà ứng dụng cần đến khi chạy bao gồm:
 - + Thư mục drawable: đây là thư mục chứa các file hình ảnh.
 - + Thư mục drawables: đây là thư mục chứa các file hình ảnh.

- + Thư mục layout: đây là thư mục chứa các file layout sử dụng để tạo giao diện người dùng của ứng dụng.
- + Thư mục menu: chứa các file giao diện của menu.
- + Thư mục mipmap: đây là thư mục chứa các biểu tượng của ứng dụng được sử dụng để hiển thị trên màn hình.
- + Thư mục values: đây là nơi chứa các file XML khác nhau như strings.xml, colors.xml (strings.xml: file quản lý các chuỗi, colors.xml: file quản lý các mã màu).

2.2.2. Cơ sở dữ liệu Firebase

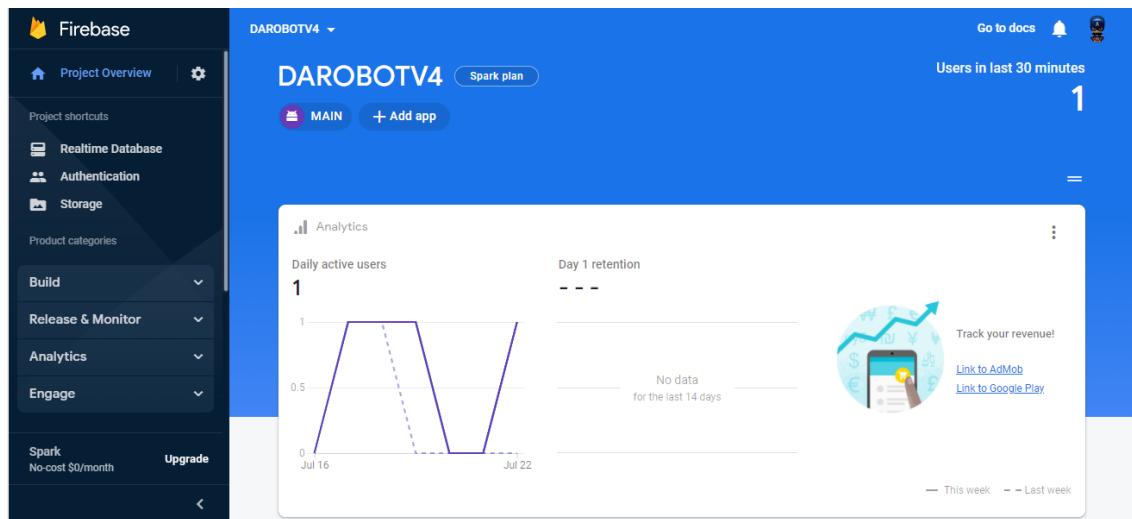
Firebase là một nền tảng được sở hữu bởi Google giúp chúng ta phát triển các ứng dụng như ứng dụng di động và web. Firebase cung cấp rất nhiều công cụ và dịch vụ tiện ích để phát triển và tạo nên một ứng dụng chất lượng. Điều đó rút ngắn thời gian phát triển và giúp ứng dụng sớm được hoàn thành. Firebase cung cấp cho người sử dụng các dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây với hệ thống máy chủ cực kỳ mạnh mẽ và ổn định của Google [4].



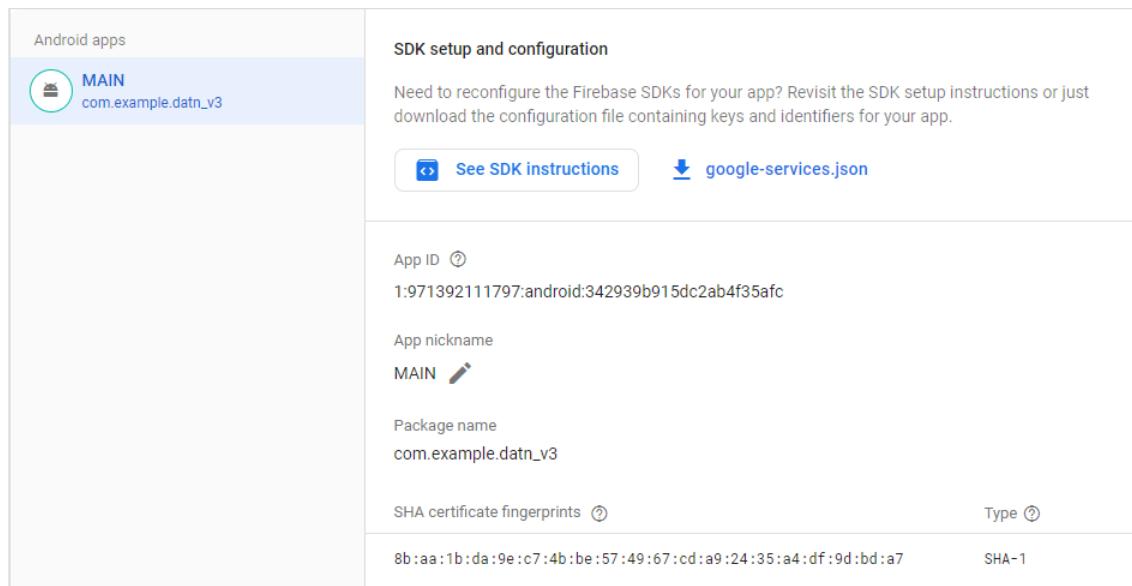
Hình 2. 5. Các chức năng của cơ sở Firebase

Chức năng chính của Firebase là giúp người dùng lập trình các ứng dụng, phần mềm trên các nền tảng web hoặc di động bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu. Người có thể tạo ra những ứng dụng real-time như ứng dụng chat và cùng nhiều tính năng như xác thực người dùng, ... Người dùng có thể dùng Firebase giống như phần backend của app.

Các dịch vụ của Firebase hoàn toàn được sử dụng miễn phí, tuy nhiên người sử dụng cần phải trả thêm tiền nếu muốn nâng cấp lên. Điều này nên cân nhắc nếu muốn xây dựng một ứng dụng lớn sử dụng phần backend là Firebase, vì giá thành khi nâng cấp còn khá đắt đỏ so với việc xây dựng backend truyền thống.



Hình 2. 6. Giao diện Firebase

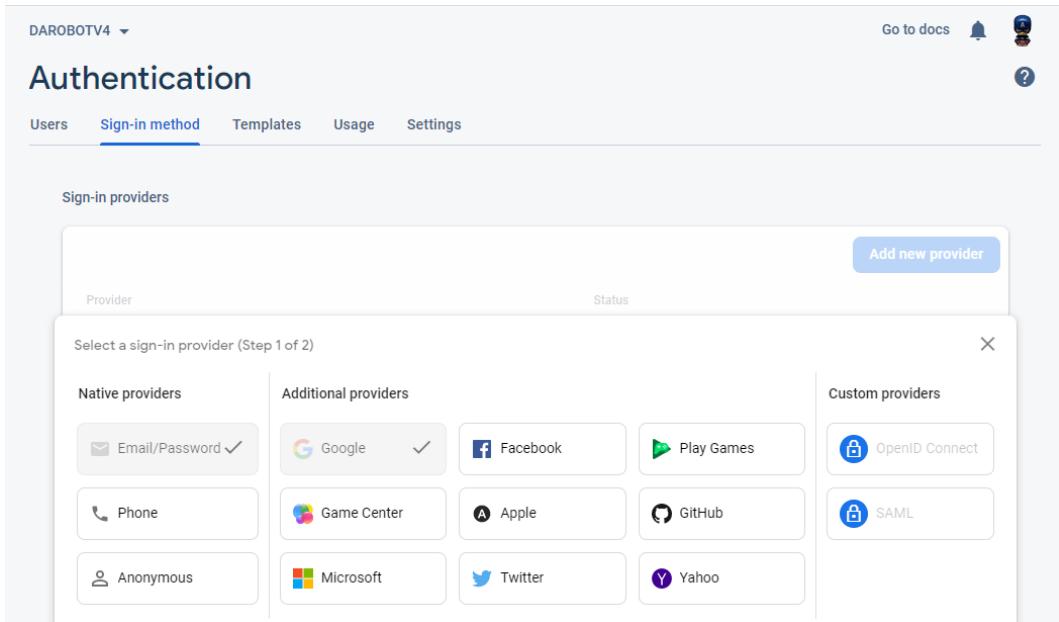


Hình 2. 7. Giá trị ID, thiết lập và cấu hình của SDK

2.2.3. Firebase Authentication

Hầu hết các ứng dụng cần biết danh tính của người dùng. Việc biết danh tính của người dùng cho phép ứng dụng lưu trữ dữ liệu người dùng một cách an toàn trên đám mây và cung cấp trải nghiệm được cá nhân hóa giống nhau trên tất cả các thiết bị của người dùng. Firebase Authentication cung cấp các dịch vụ phụ trợ, SDK dễ sử dụng và thư viện giao diện người dùng được tạo sẵn để xác thực người dùng với ứng dụng của người sử dụng. Nó hỗ trợ xác thực bằng mật khẩu, số điện thoại, các nhà cung cấp danh tính liên hợp phổ biến như Google, Facebook và Twitter. Để đăng nhập một người dùng vào ứng dụng của người sử dụng, trước tiên người sử dụng nhận được thông tin xác thực từ người dùng đó. Các thông tin xác thực này có thể là địa chỉ email và mật khẩu của người dùng hoặc mã thông báo OAuth từ nhà cung cấp danh tính được liên kết. Sau đó, người sử dụng sẽ chuyển

các thông tin xác thực này cho Firebase Authentication SDK. Các dịch vụ backend của Firebase sau đó sẽ xác minh các thông tin đăng nhập đó và trả lại phản hồi cho khách hàng. Sau khi đăng nhập thành công, người sử dụng có thể truy cập thông tin hồ sơ cơ bản của người dùng và có thể kiểm soát quyền truy cập của người dùng vào dữ liệu được lưu trữ trong các sản phẩm Firebase khác. Người sử dụng cũng có thể sử dụng mã xác thực được cung cấp để xác minh danh tính của người dùng trong các dịch vụ phụ trợ [4].



Hình 2. 8. Hình ảnh các tính năng Firebase Authentication

Các tính năng của Firebase SDK Authentication:

- Xác thực dựa trên email và mật khẩu.
- Xác thực người dùng bằng cách tích hợp với các nhà cung cấp danh tính được liên kết. Firebase SDK Authentication cung cấp các phương pháp cho phép người dùng đăng nhập bằng tài khoản Google, Facebook, Twitter và GitHub của họ.
- Xác thực người dùng bằng cách gửi tin nhắn SMS đến điện thoại của họ.
- Kết nối hệ thống đăng nhập hiện có của ứng dụng người sử dụng với Firebase SDK Authentication và có quyền truy cập vào Firebase Realtime Database và các dịch vụ Firebase khác.
- Sử dụng các tính năng yêu cầu xác thực mà không yêu cầu người dùng đăng nhập trước bằng cách tạo tài khoản ẩn danh tạm thời. Nếu sau đó người dùng chọn đăng ký, người sử dụng có thể nâng cấp tài khoản ẩn danh lên tài khoản thông thường.

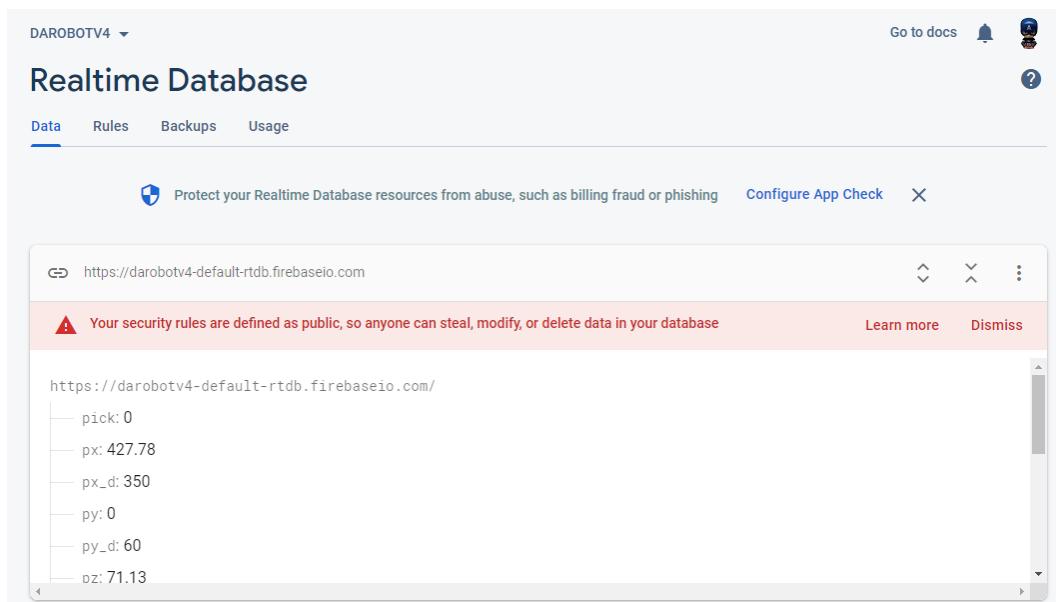
2.2.4 Firebase Realtime Database

Firebase Realtime database là cơ sở dữ liệu được lưu trữ trên đám mây. Dữ liệu được lưu trữ dưới dạng JSON và được đồng bộ hóa trong thời gian thực cho mọi máy khách

được kết nối. Khi bạn tạo các ứng dụng đa nền tảng với các ứng dụng Apple, Android và JavaScript SDK của Firebase, tất cả khách hàng của người sử dụng đều chia sẻ một cơ sở dữ liệu thời gian thực và tự động nhận các bản cập nhật với dữ liệu mới nhất [4].

Các tính năng:

- Realtime: Cơ sở dữ liệu thời gian thực của Firebase sử dụng đồng bộ hóa dữ liệu, mỗi khi dữ liệu thay đổi, mọi thiết bị được kết nối sẽ nhận được bản cập nhật đó trong vòng mili giây.
- Offline: Các ứng dụng Firebase vẫn phản hồi ngay cả khi ngoại tuyến vì Firebase Realtime Database SDK sẽ duy trì dữ liệu của người sử dụng. Khi kết nối được thiết lập lại, thiết bị khách sẽ nhận được bất kỳ thay đổi nào mà nó đã bỏ qua, đồng bộ hóa nó với trạng thái máy chủ hiện tại.
- Có thể truy cập từ thiết bị khách: Firebase Realtime database có thể được truy cập trực tiếp từ thiết bị di động hoặc trình duyệt web.
- Quy mô trên nhiều cơ sở dữ liệu: Người sử dụng có thể hỗ trợ cung cấp dữ liệu của ứng dụng trên quy mô lớn bằng cách tách dữ liệu của bạn trên nhiều phiên bản cơ sở dữ liệu trong cùng một dự án Firebase. Hợp lý hóa việc xác thực với Firebase trong dự án của bạn và xác thực người dùng trên các phiên bản cơ sở dữ liệu của bạn. Kiểm soát quyền truy cập vào dữ liệu trong từng cơ sở dữ liệu bằng quy tắc cơ sở dữ liệu thời gian thực của Firebase tùy chỉnh cho từng phiên bản cơ sở dữ liệu.



Hình 2. 9. Hình ảnh Firebase Realtime Database

Cách hoạt động: Firebase Realtime Database cho phép bạn xây dựng các ứng dụng dễ dàng, đa dạng, an toàn với cơ sở dữ liệu trực tiếp từ client. Dữ liệu được duy trì cục bộ và ngay cả khi offline, các sự kiện thời gian thực vẫn sẽ được tiếp tục khi thiết bị lấy lại

kết nối, Realtime Database sẽ đồng bộ hóa các thay đổi dữ liệu cục bộ với các dữ liệu từ máy chủ và tự động hợp nhất mọi xung đột. Firebase Realtime database còn cung cấp một quy tắc bảo mật để xác định cách cấu trúc dữ liệu và quyền đọc ghi dữ liệu thường kết hợp với Authentication Firebase. Realtime Database là một cơ sở dữ liệu NoQuery và do đó có các tối ưu hóa chức năng so với cơ sở dữ liệu quan hệ. Realtime Database được thiết kế cho việc thực hiện nhanh chóng. Điều này cho phép bạn xây dựng trải nghiệm thời gian thực tuyệt vời có thể phục vụ hàng triệu người dùng mà không cần suy nghĩ đến khả năng đáp ứng.

2.3. Lý thuyết thiết kế giao diện GUI điều khiển ROBOT

2.3.1. *Qt Designer*

Qt là một application framework đa nền tảng viết trên ngôn ngữ C++, được dùng để phát triển các ứng dụng trên máy tính, hệ thống nhúng và điện thoại. Hỗ trợ trên các nền tảng bao gồm: Linux, OS X, Windows, VxWorks, QNX, Android, Ios, BlackBerry, Sailfish OS và một số nền tảng khác. PyQt là giao diện kết hợp giữa ngôn ngữ lập trình Python và thư viện Qt. Thư viện Qt là một thư viện bao gồm các thành phần giao diện điều khiển (Widgets, graphical control elements).



Hình 2. 10. Phần mềm PyQt

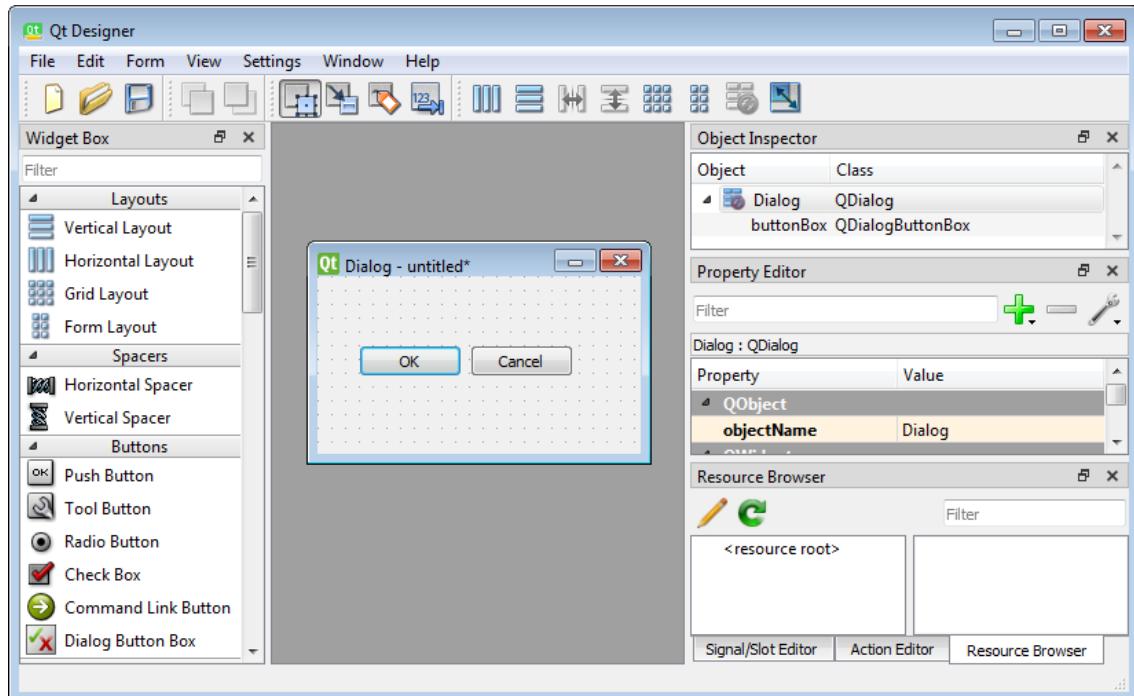
PyQt API sử dụng nhiều các class và function hỗ trợ cho việc thiết kế ra các giao diện giao tiếp với người dùng, phần mềm hỗ trợ với ngôn ngữ lập trình Python 2.x và 3.x [5].

Về chức năng thiết kế, phần mềm PyQt5 hỗ trợ các class sau:

- QtCore: là module bao gồm phần lõi không thuộc chức năng GUI, dùng để làm việc với thời gian, thư mục, các loại dữ liệu, URLs, mime type, hoặc threads.
- QtGui: bao gồm các class dùng cho việc lập trình giao diện (Windowing system integration), event handling, 2D graphics, basic imaging, fonts và text.
- QtWidgets: bao gồm các class cho widget.

- QtMultimedia: thư viện cho việc sử dụng âm thanh, hình ảnh, camera, ...
- QtBluetooth: bao gồm các class giúp tìm kiếm và kết nối với các thiết bị có giao tiếp với phần mềm.
- QtNetwork: bao gồm các class dùng cho việc lập trình mạng, hỗ trợ lập trình TCP/IP và UDP client, server hỗ trợ việc lập trình mạng.
- Bên cạnh đó còn một số như là QtPositioning, QtWebSockets, QtWebKit, ...

Trong đồ án này, sử dụng QT Designer để thiết kế giao diện điều khiển và giám sát.



Hình 2. 11 Giao diện thiết kế Qt Designer

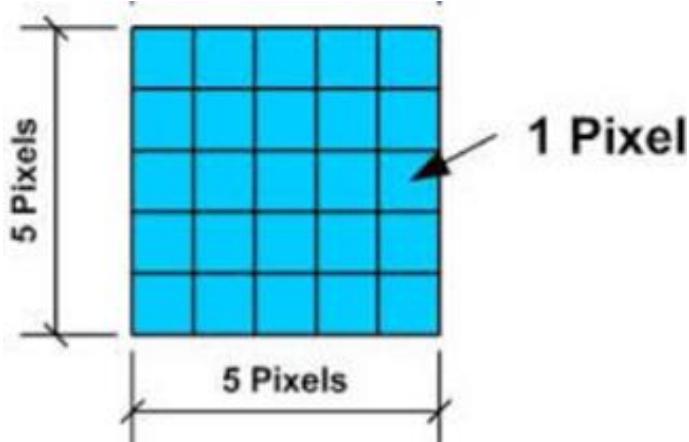
2.4. Lý thuyết xử lý lý ảnh

2.4.1. Giới thiệu về hệ thống xử lý lý ảnh

Trong xã hội loài người, ngôn ngữ là một phương tiện trao đổi thông tin phổ biến trong quá trình giao tiếp. Bên cạnh ngôn ngữ, hình ảnh cũng là một cách trao đổi thông tin mang tính chính xác cao và đặc biệt không bị cảm giác chủ quan của đối tượng giao tiếp chi phối. Thông tin trên hình ảnh rất phong phú, đa dạng và có thể xử lý bằng máy tính. Hiện nay, lĩnh vực xử lý ảnh là một lĩnh vực mang tính khoa học và công nghệ. Nó là một ngành khoa học mới mẻ so với nhiều ngành khoa học khác nhưng tốc độ phát triển của nó rất nhanh, kích thích các trung tâm nghiên cứu, ứng dụng, đặc biệt là máy tính chuyên dụng riêng cho nó. Xử lý ảnh gồm 4 lĩnh vực chính: xử lý nâng cao chất lượng ảnh, nhận dạng ảnh, nén ảnh và truy vấn ảnh. Sự phát triển của xử lý ảnh đem lại rất nhiều lợi ích cho cuộc sống của con người. Phạm vi đề tài này tập trung vào mảng nhận diện ảnh, mục tiêu phục vụ cho lĩnh vực tự động hóa.

2.4.2. Điểm ảnh

Ảnh số là tập hợp hữu hạn các điểm ảnh với mức xám phù hợp dùng để mô tả ảnh gần với ảnh thật. Là đơn vị cơ bản nhất để tạo nên một bức ảnh kỹ thuật số. Địa chỉ của điểm ảnh được xem như là một tọa độ (x, y) nào đó. Một bức ảnh kỹ thuật số, có thể được tạo ra bằng cách chụp hoặc bằng một phương pháp đồ họa nào khác, được tạo nên từ hàng ngàn hoặc hàng triệu pixel riêng lẻ. Bức ảnh càng chứa nhiều pixel thì càng chi tiết. Một triệu pixel thì tương đương với 1 megapixel. Xử lý ảnh số là một dạng xử lý tín hiệu mà trong đó đầu vào là một ảnh và đầu ra của quá trình xử lý có thể là một ảnh khác hoặc là một tập hợp chứa các tính chất hoặc tham số liên quan đến hình ảnh được cung cấp từ đầu vào [6].

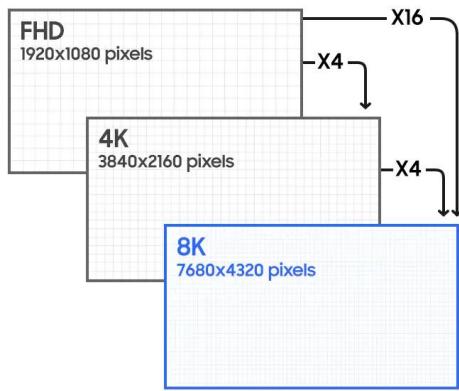


Hình 2. 12. Một điểm ảnh hay pixel ảnh

Điểm ảnh (Pixel) là một phần tử của ảnh số tại tọa độ (x, y) với độ xám hoặc màu nhất định. Kích thước và khoảng cách giữa các điểm ảnh đó được chọn thích hợp sao cho mắt người cảm nhận sự liên tục về không gian và mức xám (hoặc màu) của ảnh số gần như ảnh thật. Mỗi phần tử trong ma trận được gọi là một phần tử ảnh.

2.4.3. Độ phân giải của ảnh

Độ phân giải (resolution) của ảnh là mật độ điểm ảnh được xác định trên một ảnh số được hiển thị. Theo định nghĩa, khoảng cách giữa các điểm ảnh phải được chọn sao cho mắt người vẫn thấy được sự liên tục của ảnh. Việc lựa chọn khoảng cách thích hợp tạo nên một mật độ phân bổ, đó chính là độ phân giải và được phân bổ theo trục x và y trong không gian hai chiều. Ảnh có độ phân giải càng cao thì các biến đổi trong ảnh sẽ càng mịn và càng rõ nét.

*Hình 2. 13. Độ phân giải của ảnh*

2.4.4. Tổng quan về tiền xử lý ảnh

Đầu vào của các hệ thống xử lý ảnh thường là các tệp ảnh được thu nhận từ các thiết bị như: máy ảnh, máy quay, điện thoại,... hay các thiết bị thu nhận hình ảnh khác. Các ảnh này thường có chất lượng thấp (bị lấn các nhiễu, nhòe ảnh, các chi tiết của ảnh không rõ nét). Nguyên nhân là do: thiết bị thu nhận không đảm bảo, điều kiện thu nhận không tốt (độ sáng thay đổi, thiết bị ghi hình bị rung lắc hoặc do quá trình gửi đi các hình ảnh bị mất thông tin). Ở bước tiền xử lý ảnh, hình ảnh sẽ được cải thiện về độ tương phản, khử nhiễu, khử bóng, khử độ lệch với mục đích làm cho ảnh trở nên tốt hơn và thường được thực hiện bởi những bộ lọc. Có rất nhiều phương pháp để xử lý ảnh ở giai đoạn này bao gồm các công đoạn khôi phục và tăng cường ảnh [7]:

- Khôi phục ảnh nhằm mục đích loại bỏ hay giảm thiểu các ảnh hưởng của môi trường tác động lên ảnh. Bao gồm các bước: lọc ảnh, khử nhiễu, xoay ảnh nhằm giảm bớt các biến dạng của ảnh và đưa ảnh về trạng thái gần như ban đầu.

- Tăng cường ảnh không phải làm tăng lượng thông tin trong ảnh mà là làm nổi bật các đặc trưng của ảnh giúp cho công việc phía sau được hiệu quả hơn. Công đoạn này bao gồm các việc như: lọc độ tương phản, làm trơn ảnh, nhị phân ảnh.

2.4.5. Phân đoạn ảnh

Phân đoạn ảnh là một thao tác trong quá trình xử lý ảnh. Quá trình này thực hiện việc phân vùng ảnh thành các vùng rời rạc và đồng nhất với nhau hay nói cách khác là xác định các biên của các vùng ảnh đó. Các vùng ảnh đồng nhất này thông thường sẽ tương ứng với toàn bộ hay từng phần của các đối tượng thật sự bên trong ảnh. Vì thế, trong hầu hết các ứng dụng của lĩnh vực xử lý ảnh thì khâu phân đoạn ảnh luôn đóng một vai trò cơ bản và thường là bước tiền xử lý đầu tiên trong toàn bộ quá trình trước khi thực hiện các thao tác khác ở mức cao hơn như nhận dạng đối tượng, biểu diễn đối tượng, nén ảnh dựa trên đối tượng hay truy vấn ảnh dựa vào nội dung. Vào những thời gian đầu, các phương pháp phân vùng ảnh được đưa ra chủ yếu làm việc trên các ảnh mức xám do hạn chế về phương tiện

thu thập và lưu trữ. Ngày nay, cùng với sự phát triển về các phương diện thu thập và lưu trữ nên các ảnh màu đã hầu như thay thế toàn bộ các ảnh mức xám trong việc biểu diễn và lưu trữ thông tin do các ưu thế vượt trội hơn hẳn so với ảnh mức xám. Do đó, các kỹ thuật thực hiện việc phân vùng ảnh trên các loại ảnh màu liên tục được phát triển để đáp ứng các nhu cầu mới. Các kỹ thuật này thường được phát triển dựa trên nền tảng các thuật giải phân vùng ảnh mức xám đã có sẵn [7].

Phân đoạn ảnh là chia ảnh thành các vùng không trùng lặp. Mỗi vùng gồm một nhóm pixel liên thông và đồng nhất theo một tiêu chí nào đó. Tiêu chí này phụ thuộc vào mục tiêu của quá trình phân đoạn. Ví dụ như đồng nhất về màu sắc, mức xám, kết cấu, độ sâu của layer... Sau khi phân đoạn mỗi pixel chỉ thuộc về một vùng duy nhất. Để đánh giá chất lượng của quá trình phân đoạn ảnh là rất khó. Vì vậy, trước khi phân đoạn ảnh cần xác định rõ mục tiêu của quá trình phân đoạn là gì.

2.4.4. *Phân loại ảnh*

Ảnh gồm các loại [1]:

- **Ảnh màu:** Là ảnh kết hợp của 3 màu cơ bản lại với nhau để tạo ra một thế giới màu sinh động. Người ta thường dùng 3 byte để mô tả mức màu, tức là có khoảng 16.7 triệu mức màu.
- **Ảnh xám:** Giá trị mỗi điểm ảnh nằm trong dải giá trị từ 0 đến 255, nghĩa là cần 8 bits tương ứng với 1 byte để biểu diễn mỗi điểm ảnh.
- **Ảnh nhị phân:** Là ảnh có 2 mức trắng và đen, chỉ có 2 giá trị 0 và 1 và chỉ sử dụng 1 bit dữ liệu trên 1 điểm ảnh.



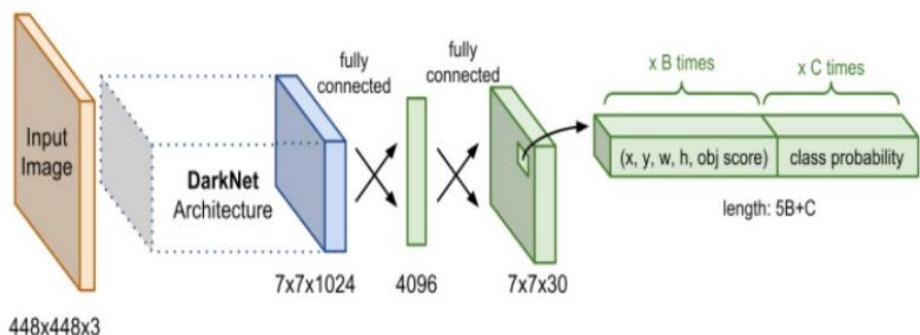
Hình 2. 14. Ảnh màu- Ảnh xám- Ảnh nhị phân

2.4.5 Giới thiệu về YOLO

YOLO (You only look once) là một mô hình mạng CNN cho việc phát hiện, nhận dạng, phân loại đối tượng. YOLO được tạo ra từ việc kết hợp giữa các convolutional layers và connected layers. Trong đó các convolutional layers sẽ trích xuất ra các feature của ảnh, còn full-connected layers sẽ dự đoán ra xác suất đó và tọa độ của đối tượng. YOLO có thể không phải là thuật toán tốt nhất nhưng nó là thuật toán nhanh nhất trong các lớp mô hình object detection. Nó có thể đạt được tốc độ gần như real time mà độ chính xác không quá giảm so với các model thuộc top đầu. YOLO là thuật toán object detection nên mục tiêu của mô hình không chỉ là dự báo nhãn cho vật thể như các bài toán classification mà nó còn xác định location của vật thể. Do đó YOLO có thể phát hiện được nhiều vật thể có nhãn khác nhau trong một bức ảnh thay vì chỉ phân loại duy nhất một nhãn cho một bức ảnh. Một trong những ưu điểm mà YOLO đem lại đó là chỉ sử dụng thông tin toàn bộ bức ảnh một lần và dự đoán toàn bộ object box chứa các đối tượng, mô hình được xây dựng theo kiểu end-to-end nên được huấn luyện hoàn toàn bằng gradient descent. Tính đến thời điểm hiện tại YOLO đã có tổng cộng 5 phiên bản (v1, v2, v3, v4, v5). Trong đó bản v5 là bản mới nhất, khắc phục được các nhược điểm của các phiên bản trước như: lỗi về việc xác định vị trí của vật thể, các ràng buộc về không gian trên những bounding box, mỗi grid cell chỉ có thể predict rất ít bounding box [8].

2.4.6 Kiến trúc mạng YOLO

Kiến trúc YOLO bao gồm: Base network là các mạng convolution làm nhiệm vụ trích xuất đặc trưng. Phần phía sau là những Extra Layers được áp dụng để phát hiện vật thể trên feature map của base network. Base network của YOLO sử dụng chủ yếu là các convolutional layer và các fully connected layer. Các kiến trúc YOLO cũng khá đa dạng và có thể tùy biến thành các version cho nhiều input shape khác nhau.



Hình 2. 15. Kiến trúc mạng YOLO

Thành phần Darknet Architecture được gọi là base network có tác dụng trích suất đặc trưng. Output của base network là một feature map có kích thước $7 \times 7 \times 1024$ sẽ được sử dụng làm input cho các Extra layers có tác dụng dự đoán nhãn và tọa độ bounding box

của vật thể. Ở phiên bản thứ 3 của YOLO tức là YOLOv3 tác giả áp dụng một mạng feature extractor là darknet-53. Mạng này gồm 53 convolutional layers kết nối liên tiếp, mỗi layer được theo sau bởi một batch normalization và một activation Leaky Relu. Để giảm kích thước của output sau mỗi convolution layer, tác giả down sample bằng các filter với kích thước là 2. Mẹo này có tác dụng giảm thiểu số lượng tham số cho mô hình. Các bức ảnh khi được đưa vào mô hình sẽ được scale để về chung một kích thước phù hợp với input shape của mô hình và sau đó được gom lại thành batch đưa vào huấn luyện. Hiện tại YOLO đang hỗ trợ 2 đầu vào chính là 416x416 và 608x608. Mỗi một đầu vào sẽ có một thiết kế các layers riêng phù hợp với shape của input. Sau khi đi qua các layer convolutional thì shape giảm dần theo cấp số nhân là 2. Cuối cùng ta thu được một feature map có kích thước tương đối nhỏ để dự báo vật thể trên từng ô của feature map. Kích thước của feature map sẽ phụ thuộc vào đầu vào. Đối với input 416x416 thì feature map có các kích thước là 13x13, 26x26 và 52x52. Và khi input là 608x608 sẽ tạo ra feature map 19x19, 38x38, 72x72.

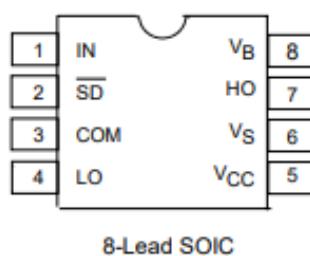
Để tìm được bounding box cho vật thể, YOLO sẽ cần các anchor box làm cơ sở ước lượng. Những anchor box này sẽ được xác định trước và sẽ bao quanh vật thể một cách tương đối chính xác. Sau này thuật toán regression bounding box sẽ tinh chỉnh lại anchor box để tạo ra bounding box dự đoán cho vật thể.

Sau khi đã định nghĩa được những thông tin mà mô hình cần phải dự đoán, và kiến trúc của mô hình CNN. Bây giờ là lúc mà chúng ta sẽ định nghĩa hàm lỗi. YOLO sử dụng hàm độ lỗi bình phương giữ dự đoán và nhãn để tính độ lỗi cho mô hình. Cụ thể, độ lỗi tổng của chúng ta sẽ là tổng của 3 độ lỗi sau:

- Độ lỗi của việc dữ đoán loại nhãn của Object-Classification loss.
- Độ lỗi của dự đoán tạo độ cũng như chiều dài, rộng của boundary box -Localization loss.
- Độ lỗi của ô vuông có chứa object nào hay không - Confidence loss.

2.5. Lý thuyết thiết kế mạch cầu H điều khiển động cơ

Mạch cầu H sử dụng 4 mosfet được điều khiển thông qua IC lái IR2184 [9].



Hình 2. 16. Sơ đồ chân của IC IR2184

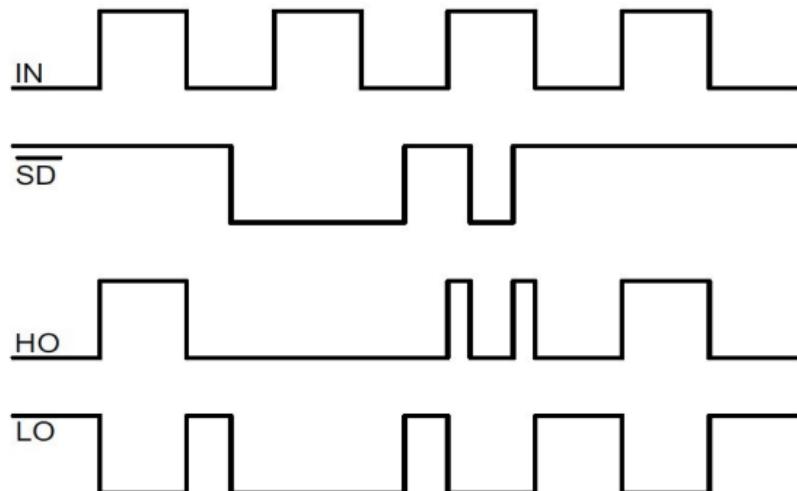
Chức năng các chân của IC IR2184:

- IN: chân nhận tín hiệu PWM để điều khiển ngõ ra.
- SD: chân ín hiệu cho phép hoạt động của IC.
- COM: chân chung.
- HO: chân điều khiển kích dẫn mosfet thứ nhất.
- LO: chân điều khiển kích dẫn mosfet thứ hai.
- VB: chân kết nối với tụ điện bootstrap.
- VS: chân kết nối với động cơ.
- VCC: nguồn cung cấp cho IC.

Nguyên lý hoạt động của IC IR2184:

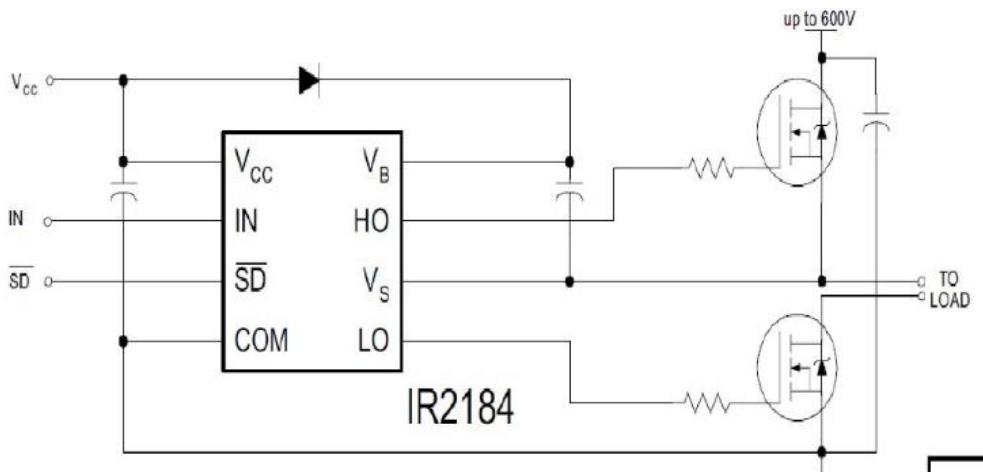
Nếu tín hiệu SD ở mức logic 1 thì hai tín hiệu ra LO và HO phụ thuộc hoàn toàn vào trạng thái của xung băm, nếu xung ở trạng thái 1 thì HO ở mức 1 còn LO ở mức 0.

Ngược lại, Nếu xung ở mức 0 thì HO ở mức 0 và LO ở mức 1. Nói cách khác khi SD ở mức 1 thì HO lặp lại trạng thái xung băm còn LO thì ngược lại trạng thái của xung băm. Ngược lại, nếu tín hiệu SD ở mức logic 0 thì tín hiệu LO và HO luôn ở mức logic không mà không phụ thuộc và trạng thái của xung băm. Tín hiệu này thường dùng để khóa cả 2 mosfet khi cần thiết. Để đảm bảo an toàn cho các mosfet, tránh hiện tượng trùng dẫn hay còn gọi là hiện tượng ngắn mạch dẫn tới hư hỏng các mosfet thì IC này được thiết kế một cách đặc biệt với thời gian trễ trước khi xuất tín hiệu ra. Cụ thể được thể hiện thông qua biểu đồ trạng thái sau:



Hình 2. 17. Biểu đồ tín hiệu của IC IR2184

Nhìn vào biểu đồ này, ta thấy khi tín hiệu SD ở mức logic 1 và tín hiệu băm xung thay đổi trạng thái thì các tín hiệu ra không thay đổi ngay lập tức mà có trễ thời gian để đảm bảo an toàn. Cụ thể xét trên sơ đồ kết nối điển hình ta có [9]:



Hình 2. 18. Sơ đồ kết nối điện hình IC IR2184

Khi tín hiệu IN đang ở mức 0 thì tín hiệu LO đang ở mức 1 tức là mosfet ở phía bên dưới đang dẫn, còn tín hiệu HO đang ở mức 0 tức là mosfet đang ở phía trên bên khóa. Nếu mosfet phía dưới chưa khóa mà mosfet phía trên đã dẫn thì sẽ khiến cho nguồn được đóng kín trực tiếp từ VCC qua 3 mosfet tới GND, tức là bị ngắn mạch nguồn qua 2 mosfet, điều này làm cho 2 mosfet này bị hỏng ngay lập tức. Để tránh hiện tượng này, khi tín hiệu IN chuyển từ mức 0 lên mức 1, thay vì tín hiệu HO chuyển lên mức 1 và tín hiệu LO chuyển xuống mức 0 ngay lập tức thì tín hiệu LO được chuyển xuống mức 0 trước khi tín hiệu HO chuyển lên mức 1. Khi mà LO chuyển hoàn toàn xuống mức 0 (tức là mosfet phía dưới bị khóa hoàn toàn) thì tín hiệu trên chân HO mới bắt đầu được chuyển lên mức 1 (tức là mosfet phía trên mới được dẫn). Tương tự như vậy, khi tín hiệu IN chuyển từ mức 1 xuống mức 0 thì tín hiệu HO sẽ chuyển từ 1 xuống mức 0 trước, sau khi tín hiệu HO đã chuyển hoàn toàn mức 0 thì tín hiệu LO mới bắt đầu chuyển từ mức 0 lên mức 1 [9].

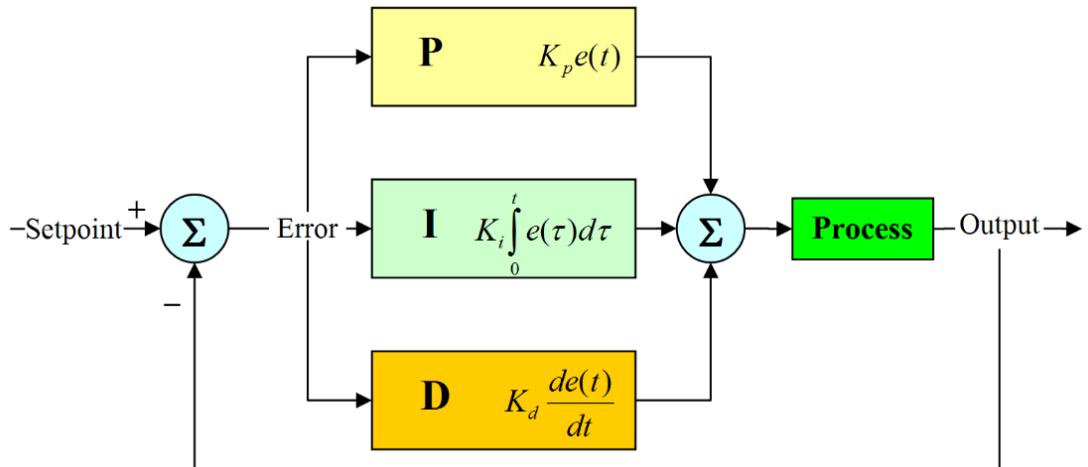
2.6. Bộ điều khiển PID

Một bộ điều khiển vi tích phân tỉ lệ (PID- Proportional Integral Derivative) là một cơ chế phản hồi vòng điều khiển được sử dụng rộng rãi trong các hệ thống điều khiển công nghiệp. Bộ điều khiển PID sẽ tính toán giá trị "sai số" là hiệu số giữa giá trị đo thông số biến đổi và giá trị đặt mong muốn. Bộ điều khiển sẽ thực hiện giảm tối đa sai số bằng cách điều chỉnh giá trị điều khiển đầu vào.

Giải thuật tính toán bộ điều khiển PID bao gồm 3 thông số riêng biệt, do đó đôi khi nó còn được gọi là điều khiển ba khâu: các giá trị tỉ lệ, tích phân và đạo hàm, viết tắt là P, I, và D. Giá trị tỉ lệ xác định tác động của sai số hiện tại, giá trị tích phân xác định tác động của tổng các sai số quá khứ, và giá trị vi phân xác định tác động của tốc độ biến đổi sai số. Tổng hợp của ba tác động này dùng để điều chỉnh quá trình thông qua một phần tử điều khiển như vị trí của van điều khiển hay bộ nguồn của phần tử gia nhiệt. 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.

Bằng cách điều chỉnh 3 hằng số trong giải thuật của bộ điều khiển PID, bộ điều khiển có thể dùng trong những thiết kế có yêu cầu đặc biệt. Đáp ứng của bộ điều khiển có thể được mô tả dưới dạng độ nhạy sai số của bộ điều khiển, giá trị mà bộ điều khiển vọt lố điểm đặt và giá trị dao động của hệ thống [10].



Hình 2. 19. Bộ điều khiển PID

Sơ đồ điều khiển PID được đặt tên theo ba khâu hiệu chỉnh của nó, tổng của ba khâu này tạo thành bởi các biến điều khiển (MV). Ta có [10]:

$$MV(t) = P_{out} + I_{out} + D_{out} \quad (2.1)$$

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

Khâu tích phân: tỉ lệ thuận với cả biến độ sai số lẫn quãng thời gian xảy ra sai số. Tổng sai số tức thời theo thời gian (tích phân sai số) cho ta tích lũy bù đã được hiệu chỉnh trước đó. Tích lũy sai số sau đó được nhân với độ lợi tích phân và cộng với tín hiệu đầu ra của bộ điều khiển. Biến độ phân phối của khâu tích phân trên tất cả tác động điều chỉnh được xác định bởi độ lợi tích phân (K_i).

Khâu vi phân: Tốc độ thay đổi của sai số quá trình được tính toán bằng cách xác định độ dốc của sai số theo thời gian (tức là đạo hàm bậc một theo thời gian) và nhân tốc độ này với độ lợi tỉ lệ K_d . Biến độ của phân phối khâu vi phân (đôi khi được gọi là tốc độ) trên tất cả các hành vi điều khiển được giới hạn bởi độ lợi vi phân (K_d).

CHƯƠNG 3: THIẾT KẾ VÀ TÍNH TOÁN ROBOT

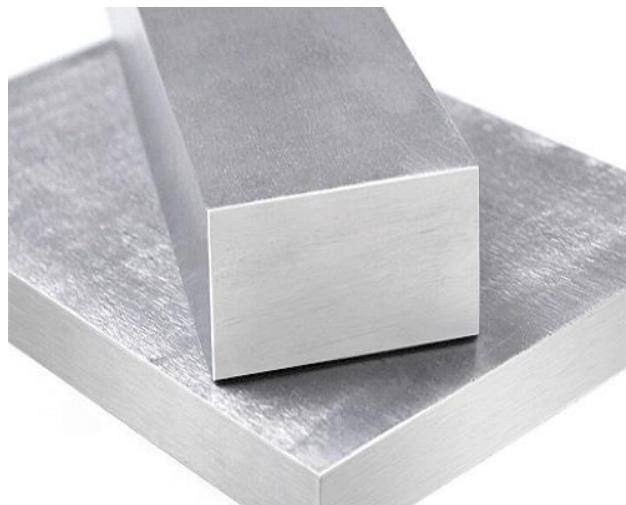
3.1. Thiết kế cơ khí robot 4 bậc tự do trên phần mềm Solidworks

3.1.1. Giới thiệu phần mềm Solidworks

Việc thiết kế cơ khí cho một robot 4 chân đòi hỏi một phần mềm thiết kế có sự chính xác cao cũng như khả năng xây dựng chi tiết 3D nhanh chóng, dễ dàng và tiện lợi cho người sử dụng. Hơn thế, các chi tiết 3D sau khi được thiết kế xong cũng có thể lắp ráp lại với nhau tạo thành một bộ phận hoàn chỉnh. Dựa theo những nhu cầu đó, nhóm em lựa chọn và sử dụng phần mềm Solidworks.

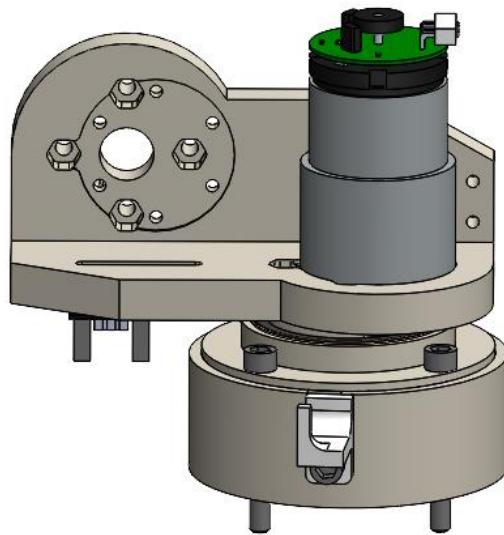
3.1.2. Vật liệu thiết kế và ý tưởng mô hình

Hiện nay có rất nhiều loại vật liệu có thể dùng để thiết kế mô hình robot như nhựa in 3D, mica, gỗ ván ép, nhựa đúc nhưng nhóm lựa chọn gia công CNC nhôm nhám nâng cao độ chính xác khi lắp ráp robot. Robot khi được làm bằng nhôm sẽ có độ cứng, độ bền cao, đảm bảo được tính thẩm mỹ của robot trong khi khối lượng robot không quá lớn. Mô hình robot của nhóm được tính toán thiết kế để sử dụng những vật liệu có bán sẵn trên thị trường khu vực thành phố Hồ Chí Minh để rút ngắn thời gian thi công mô hình robot đồng thời tạo sự chủ động, thuận lợi trong việc sửa chữa robot khi bị hư hỏng. Mô hình robot được thiết kế có nhiều nét riêng biệt nhưng vẫn đảm bảo tính linh hoạt khi vận hành.



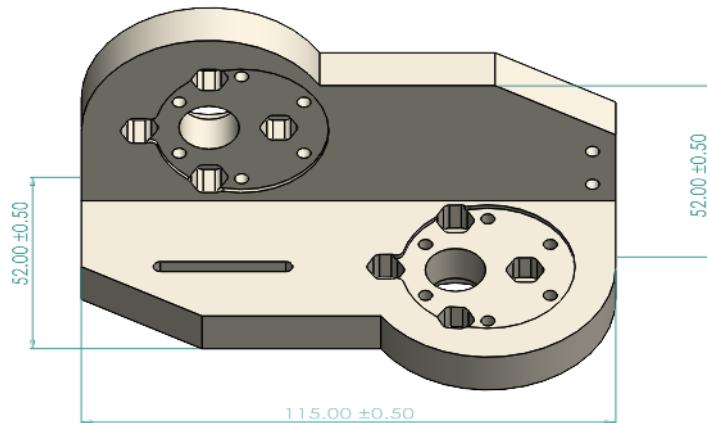
Hình 3. 1. Phôi nhôm trước khi CNC

Phần đế của robot 4 bậc tự do sẽ đỡ toàn bộ khối lượng cánh tay robot và chịu thêm các lực moment sinh ra khi robot chuyển được nên sẽ được thiết kế để có độ chịu lực cao, khả năng kết nối tốt với joint phía trên để không có độ rơ khi xoay từ đó tạo nên sự mượt mà trong chuyển động. Theo đó, nhóm lên ý tưởng và bắt đầu thiết kế phần đế của robot như sau.

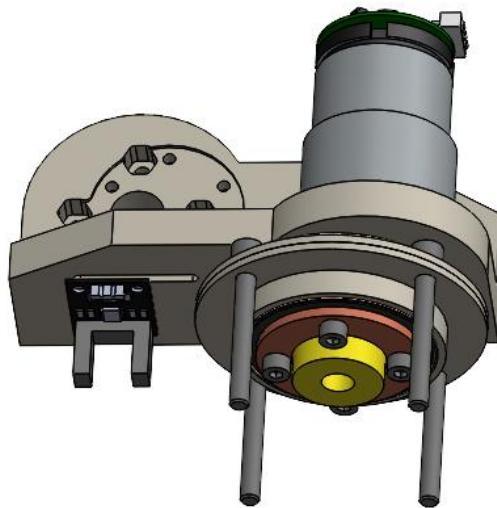


Hình 3. 2. Thiết kế cụm đέ của robot trên SolidWorks.

Phần đέ của robot sẽ được thiết kế CNC dạng hình tròn bao quanh toàn bộ khớp của robot. Khớp 1 được kề có có hai bạc đạn 6807 để đỡ toàn bộ khối lượng cánh tay trên nên giảm đáng kể lực ma sát, đó từ đó tạo nên sự mượt mà khi xoay. Hai bạc đạn này sẽ được siết chặt vào nhau bằng 4 ốc M4 để tạo thành một khối vững chắc nên sẽ không tạo ra độ rơ khi khớp xoay. Động cơ dùng cho robot là loại động cơ tích hợp sẵn hộp số và encoder. Trục của hộp số động cơ được thiết kế đồng tâm đặt với trục xoay của các joint để có thể truyền động trực tiếp thông qua khớp nối mềm mà không cần thông qua bất kỳ bộ truyền nào khác. Thiết kế này mang lại sự nhỏ gọn, linh hoạt hơn cho robot khi vận hành. Bên cạnh đó phần khung robot được thiết kế các vị trí gắn cảm biến chữ U để sethome cho robot khi bắt đầu vận hành nhưng không làm ảnh hưởng nhiều đến không gian làm việc của robot.

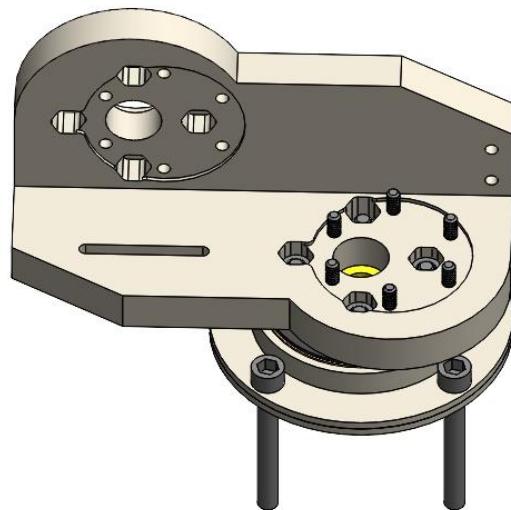


Hình 3. 3. Link 1 của robot



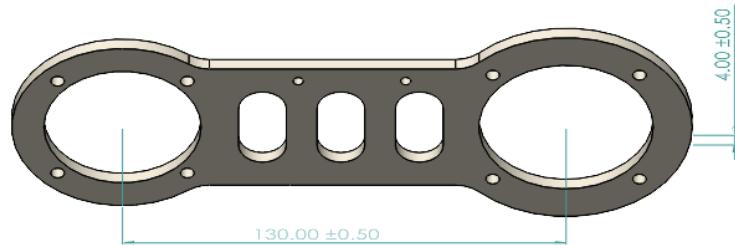
Hình 3. 4. Thiết kế cụm bạc đạn và truyền động của robot.

Moment sẽ được truyền từ trục hộp số động cơ sang trục ty trên để thông qua khớp nối mềm D20L30.



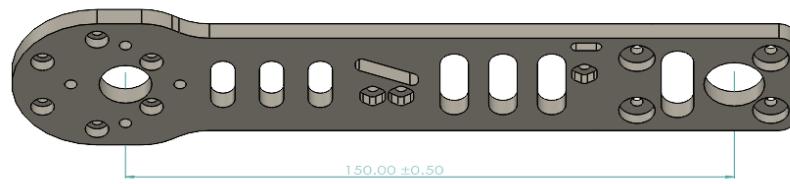
Hình 3. 5. Thiết kế chốt lắp động cơ trên link 1.

Nhóm thiết kế 4 lỗ chứa tán M4 sẽ được thiết kế âm vào trong link 1 trong khi đó động cơ của khớp 1 sẽ được gắn chặt và link 1 thông qua 6 ốc lục giác M3 được đặt trong link 1. Sau khi đã thi công được một khớp của robot nhóm sẽ tiến hành lắp ráp hoàn chỉnh để kiểm tra khả năng chịu tải và chịu lực của khớp ở nhiều mức độ khác nhau. Sau khi khớp xoay đạt được những yêu cầu đã đặt ra trước đó, nhóm tiến hành thiết kế khớp 2 và khớp 3 theo khớp 1. Do khớp 3 sẽ chịu ít lực hơn khớp 2 nên khớp 3 sẽ được thiết kế nhỏ gọn để tối ưu về khối lượng nhưng vẫn đảm bảo khả năng chịu lực tốt.



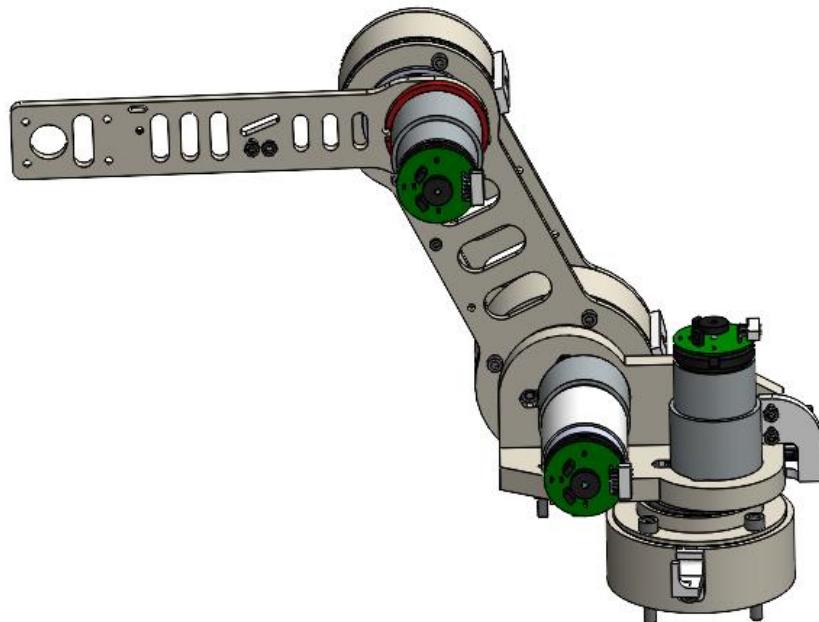
Hình 3. 6. Thiết kế link 2 của robot

Link 2 của robot được thiết kế gồm 2 tấm nhôm dày 4.5mm lắp song song với nhau và trên các tấm nhôm đó có cắt các lỗ ứng suất để giảm khối lượng và lắp cảm biến chữ U nhưng không ảnh hưởng nhiều đến khả năng chịu lực của robot.



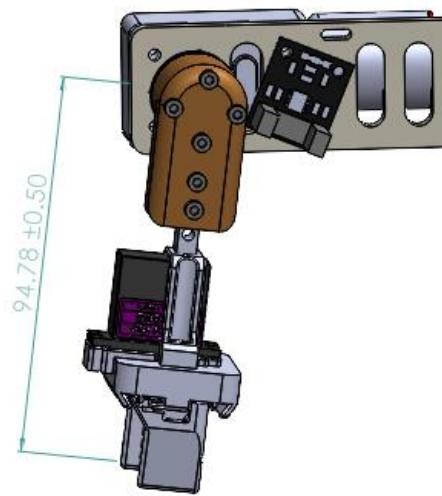
Hình 3. 7. Thiết kế link 3 của robot

Link 3 của robot được thiết kế gồm 1 tấm nhôm dày 5mm và khoan các lỗ để lắp ốc âm, động cơ link 4, cảm biến.



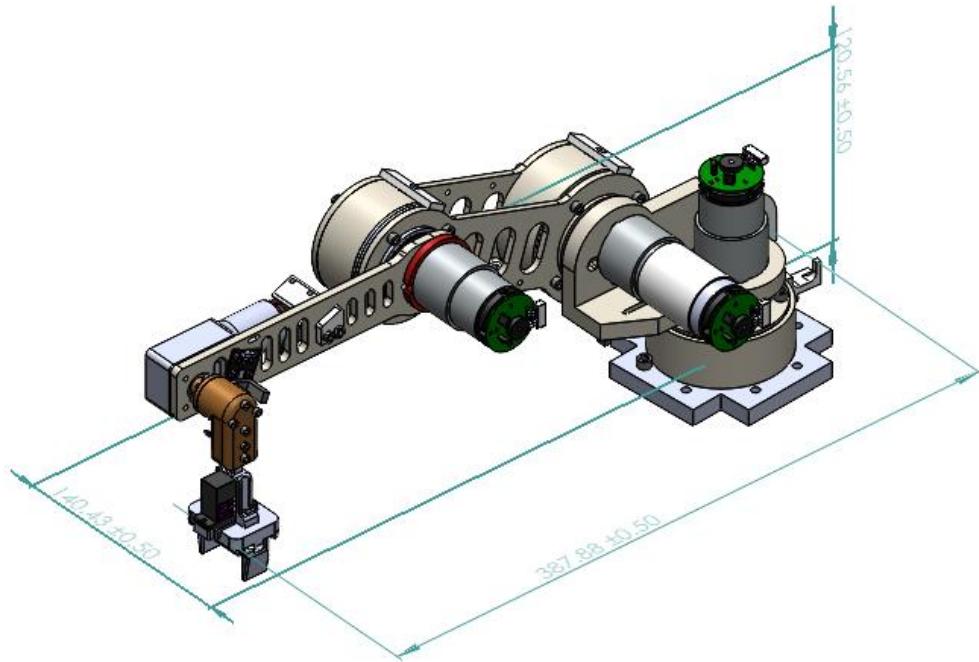
Hình 3. 8. Thiết kế 3 khớp 1,2,3 của robot trên SolidWorks.

Link 4 của robot được thiết kế gắn trực tiếp lên trực hộp số của động cơ khớp 4 nhằm mục đích giảm khối lượng và tăng tính linh hoạt khi vận hành cho khớp 4.

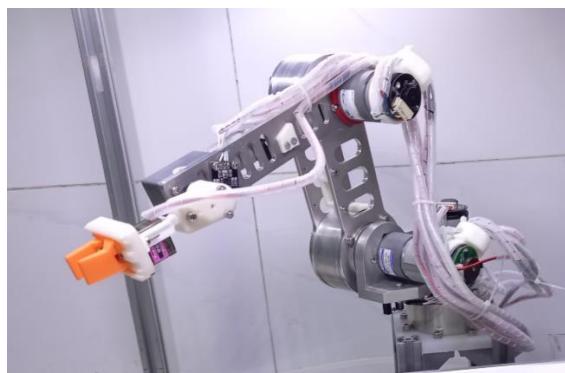


Hình 3. 9. Thiết kế khớp 4 của robot trên SolidWorks.

Sau khi đã vẽ hoàn chỉnh 4 khớp của robot, nhóm tiến hành lắp ráp các bộ phận của robot lại với nhau bằng tính năng Assembly của Solidworks.



Hình 3. 10. Thiết kế robot 4 bậc tự do hoàn chỉnh trên SolidWorks.



Hình 3. 11. Mô hình robot 4 bậc tự do thực tế.

STT	Tên linh kiện	Số lượng
1	Nhôm định hình	3 (mét)
2	Động cơ JGB37-545	1
3	Động cơ JGB37-520	2
4	Động cơ JGY	1
5	Óc lục giác M3	32
6	Tán M3	20
7	Óc lục giác M4	20
8	Tán M4	20
9	Óc lục giác M5	12
10	Tán M5	12
11	CNC nhôm	22 (chi tiết)
12	Bạc đạn 6808	6
13	Bạc đạn 6807	2
14	Khớp nối mềm D20L30	3
15	Cảm biến chữ U	4
16	Servo	1
17	Mặt bích	1

Bảng 3. 1. Danh sách linh kiện sử dụng làm robot

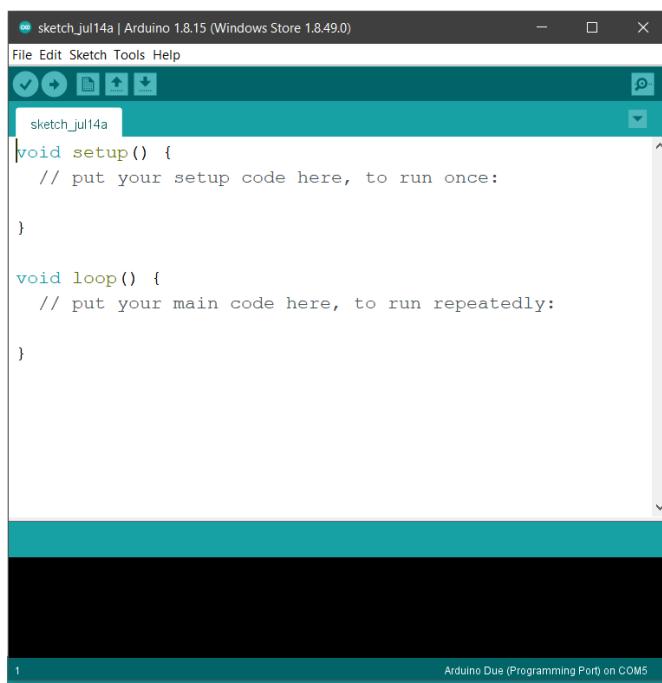
3.2. Thiết kế mạch điện tử

3.2.1. Giới thiệu Arduino

Arduino là một bo mạch vi xử lý được dùng để lập trình tương tác với các thiết bị phần cứng như cảm biến, động cơ, đèn hoặc các thiết bị khác. Đặc điểm nổi bật của Arduino là môi trường phát triển ứng dụng cực kỳ dễ sử dụng, với một ngôn ngữ lập trình có thể học một cách nhanh chóng ngay cả với người ít am hiểu về điện tử và lập trình. Và điều làm nên hiện tượng Arduino chính là mức giá rất thấp và tính chất nguồn mở từ phần cứng tới phần mềm [1].

Các Arduino tích hợp môi trường phát triển (IDE) là một nền tảng ứng dụng (ví dụ của Windows, MacOS, Linux) được viết bằng ngôn ngữ lập trình Java. Nó được sử dụng để viết và tải các chương trình lên các bảng tương thích Arduino, nhưng cũng với sự trợ giúp của các lõi bên thứ 3, các bảng phát triển nhà cung cấp khác. Arduino IDE hỗ trợ các ngôn ngữ C và C++ bằng cách sử dụng các quy tắc đặc biệt về cấu trúc mã [11].

Arduino IDE cung cấp một thư viện phần mềm từ dự án Wires, cung cấp nhiều thủ tục đầu vào và đầu ra phổ biến. Mã do người dùng viết chỉ yêu cầu hai hàm cơ bản, để bắt đầu phác thảo và vòng lặp chương trình chính, được biên dịch và liên kết với một chương trình chính thành một chương trình điều hành theo chu kỳ thực thi với chuỗi công cụ GNU, cũng được bao gồm trong bản phân phối IDE.



Hình 3. 12. Giao diện làm việc của phần mềm lập trình Arduino IDE 1.8.15



Hình 3. 13. Bo Arduino Mega

Arduino Mega là board sử dụng nhân ARM 32bit cortex-M3 và được nạp bootloader tương thích code như các board arduino khác. Arduino Due có 54 chân Digital I/O (với 12 chân có thể sử dụng chức năng PWM), 12 ngõ vào tín hiệu Analog, 4 UARTs (cổng truyền nối tiếp), board tích hợp 1 cổng USB OTG, 2 bộ chuyển đổi DAC (digital to Analog), 2 TWI, một jack cấp nguồn, 1 jack SPI, một jack chuẩn Jtag, một nút reset, và một nút xóa chương trình. [11] Thông số kỹ thuật chi tiết của Arduino Due được mô tả ở bảng bên dưới.

Thông số	Giá trị
Điện áp hoạt động	3.3 VDC
Điện áp vào	7~12 V
Điện áp giới hạn	6~20 V
Số chân Digital	54 (12 chân PWM)
Số chân vào Analog	16
Số chân ra Analog	2
Dòng điện DC trên chân 3.3V	800 mA
Dòng điện DC trên chân 5V	800 mA
Bộ nhớ Flash	512 KB
SRAM	96 KB
Tần số xung clock	84 MHz

Bảng 3. 2. Bảng thông số bo Arduino Mega.

Bên cạnh các dòng vi điều khiển Atmega thì ESP32 là sản phẩm rất được người sử dụng ưa chuộng, được ứng dụng trong các đề tài liên quan đến IoT và robot hiện nay. ESP32 là phiên bản nâng cấp của ESP8266. Nó được trang bị vi xử lý dual-core 32-bit và tích hợp sẵn WiFi và Bluetooth. Ngoài ra, bo mạch cũng có nhiều RAM, nhiều GPIO, ADC và cổng kết nối nên ESP32 là lựa chọn phù hợp. Trong đề tài này nhóm chúng em đã sử dụng ESP32 cho mục đích truyền nhận dữ liệu để điều khiển robot.

3.2.2. Giới thiệu **ESP32-WROOM-32E**

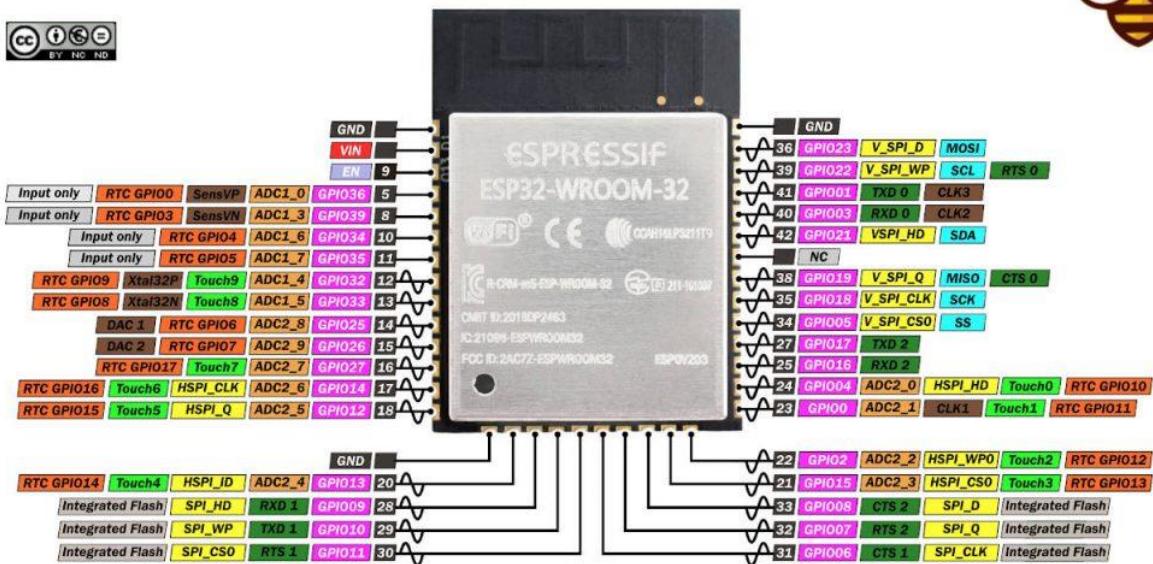


Hình 3. 14. ESP32-WROOM-32E.

- Kích thước: 18 mm x 20 mm x 3 mm
- CPU: Xtensa Dual-Core 32-bit LX6 với tần số hoạt động lên đến 240 MHz
- Kết nối WiFi:
 - +Wi-Fi: 802.11 b/g/n/e/i
 - +Bluetooth: BR/EDR phiên bản v4.2 và BLE
- Ngoại vi:
 - +Bộ chuyển đổi ADC 12 bit, 16 kênh.
 - +Bộ chuyển đổi 8-bits DAC: 2 kênh.
 - +10 chân để giao tiếp với cảm biến chạm (touch sensor).
 - +IR (TX/RX).
 - +Ngõ ra PWM cho điều khiển Motor.
 - +LED PWM: 16 kênh.
- Nhiệt độ hoạt động ổn định: -40⁰C đến 85⁰C.
- Điện áp hoạt động: 3-3.6V.

ESP32-wroom-32 PINOUT

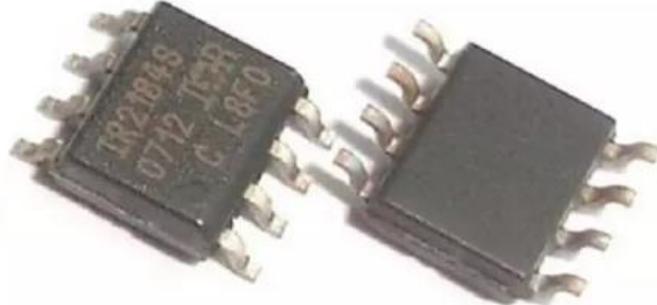
www.mischianti.org (cc) BY-NC-ND



Hình 3. 15. Sơ đồ chân ESP32-WROOM-32E.

3.2.3. Các linh kiện khác

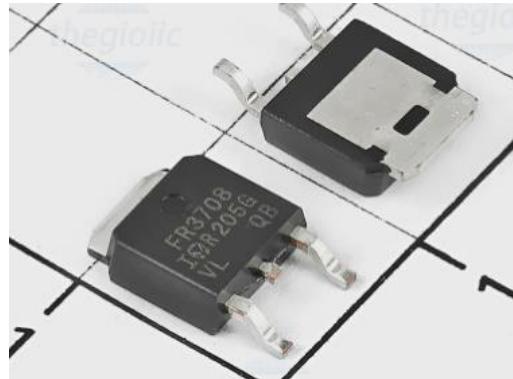
* **IC IR2184:**



Hình 3. 16. Sơ đồ chân IC IR2184.

- Điện áp max: 600V
- Điện áp điều khiển: 10-20V
- Dải nhiệt hoạt động: -40 đến 105 độ C.
- Thời gian ON: $t = 680$ ns
- Thời gian OFF: $t = 270$ ns
- Thời gian trễ: $t = 500$ ns

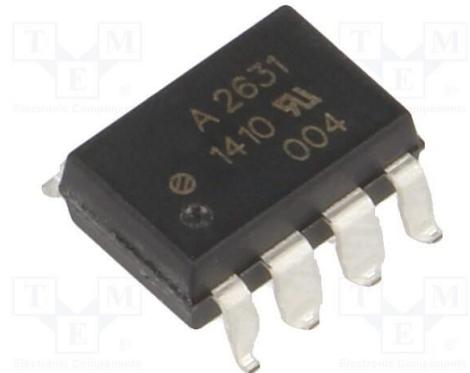
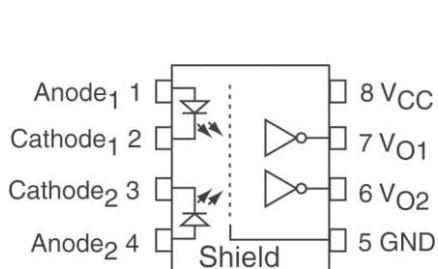
*** IC IRFR3708:**



Hình 3. 17. Sơ đồ chân IC IRFR3708.

- Loại kênh: N
- Công nghệ: MOSFET (Metal Oxide)
- Điện áp xả vào nguồn (V_{dss}): 30V
- Dòng điện xả liên tục (I_d): 61A (T_c)
- Công suất max: 87W (T_c)
- Nhiệt độ hoạt động: -55°C ~ 150°C

*** IC HCPL 2631:**



Hình 3. 18. Sơ đồ chân IC IRFR3708.

- Số kênh: 2
- Điện áp cách ly: 2500Vrms
- Loại ngõ vào: DC
- Loại ngõ ra: Open Collector

- Dòng ngõ ra/ Kênh: 50mA
- Tốc độ dữ liệu: 10Mbps
- Loại ngõ vào: DC
- Loại ngõ ra: Open Collector
- Thời gian lên/xuống (Typ): 50ns, 12ns
- Nguồn cấp: 4.5V ~ 5.5V

***LM7805:**



Hình 3. 19. Sơ đồ chân IC LM7805.

- Điện áp đầu vào: 35V
- Điện áp đầu ra: 5 V
- Dòng ra max: 1.5A
- Kiểu chân: TO-263-3
- Nhiệt độ hoạt động: 40°C ~ 125°C

***AMS1117-3.3V**



Hình 3. 20. Sơ đồ chân IC AMS1117-3.3V.

- Điện áp đầu vào: 5V.
- Điện áp đầu ra: 3.3V.

- Dòng ra max: 800mA.
 - Kiểu chân: SOT223.
 - Nhiệt độ hoạt động: 40°C ~ 125°C.
- * **Động cơ DC Servo JGB37-545 DC**



Hình 3. 21. Động cơ DC Servo JGB37-545 DC

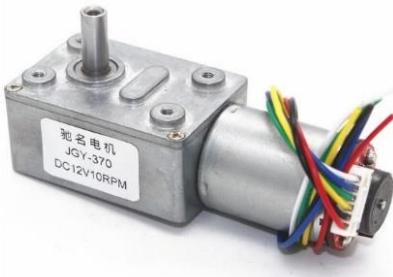
- Tỉ số truyền 168:1
- Dòng không tải: 200mA
- Dòng chịu đựng tối đa khi có tải: 5A
- Tốc độ không tải: 37RPM
- Tốc độ chịu đựng tối đa khi có tải: 30RPM
- Lực kéo Moment định mức: 21KG.CM
- Lực leo Moment tối đa: 84KG.CM
- Số xung Encoder mỗi kênh trên 1 vòng quay trực chính: 16 x 168 = 2688 xung

* **Động cơ DC Servo JGB37-520**



Hình 3. 22. Động cơ DC Servo JGB37-520

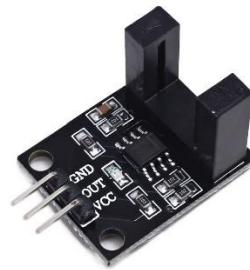
- Tỉ số truyền 270:1
 - Dòng không tải: 120mA
 - Dòng chịu đựng tối đa khi có tải: 1A
 - Tốc độ không tải: 37RPM
 - Tốc độ chịu đựng tối đa khi có tải: 28RPM
 - Lực kéo Moment định mức: 30KG.CM
 - Lực leo Moment tối đa: 35KG.CM
 - Số xung Encoder mỗi kênh trên 1 vòng quay trực chính: $11 \times 270 = 2970$ xung.
- * **Động cơ DC Servo JGY370**



Hình 3. 23. Động cơ DC Servo JGY370

- Tỉ số truyền 200:1.
- Dòng không tải: 60mA.
- Dòng chịu đựng tối đa khi có tải: 1.3A.
- Tốc độ không tải: 30RPM.
- Tốc độ chịu đựng tối đa khi có tải: 22RPM.
- Lực kéo Moment định mức: 7.4KG.CM.
- Lực leo Moment tối đa: 25KG.CM.
- Số xung Encoder mỗi kênh trên 1 vòng quay trực chính: $11 \times 200 = 2200$ xung.

* **Cảm Biến Quang Chữ U**



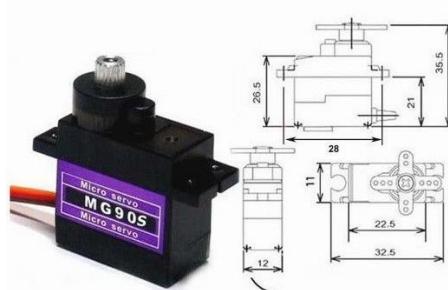
Hình 3. 24. Cảm biến quang chữ U

- Kích thước: 23 x 20 x 20mm.

- IC: LM393.

- Điện áp làm việc: 5VDC.

* **MG90S**



Hình 3. 25. Kích thước Servo MG90S.

- Điện áp hoạt động: 4.8 ~ 6VDC.

- Bánh răng: Kim loại.

- Độ dài dây nối: 175mm.

- Trọng lượng: 13.4g.

- Kích thước: 22.8 x 12.2 x 28.5mm

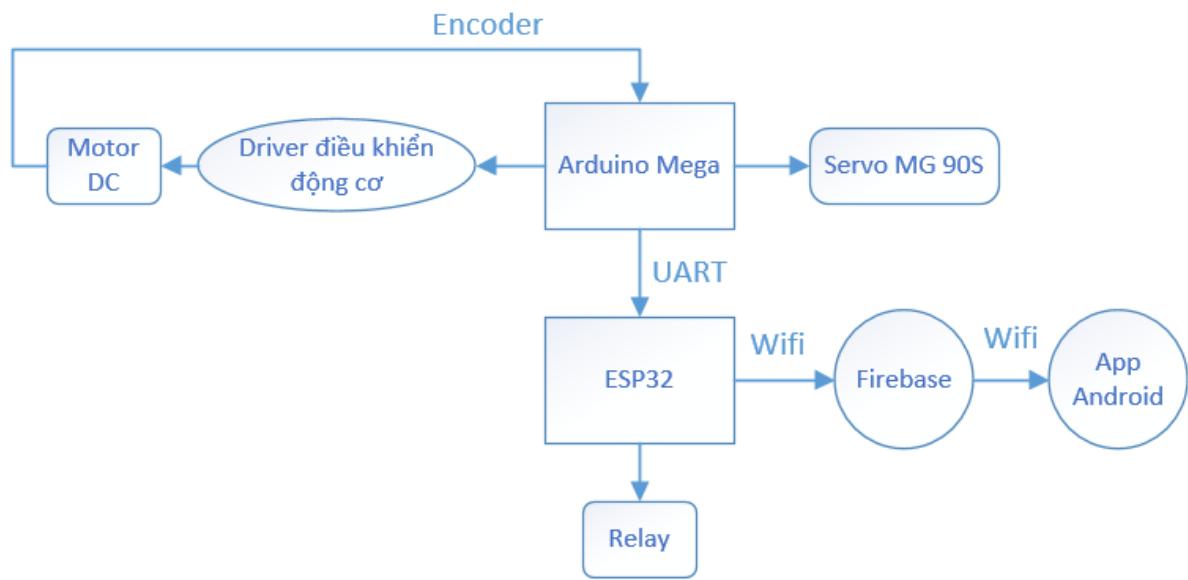
* **IC FT232:**



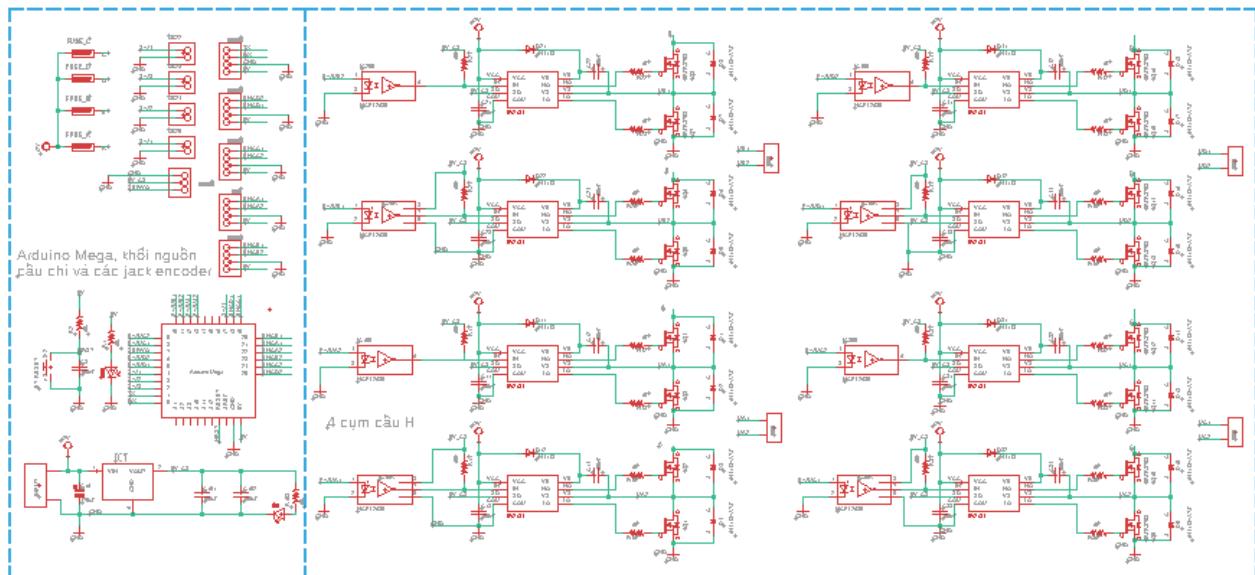
Hình 3. 26. IC FT232

- Giao thức: USB.
- Chức năng: USB to UART.
- Giao tiếp: UART.
- Điện áp cung cấp: 3,3V ~ 5,25V.
- Dòng điện: 15mA.
- Nhiệt độ hoạt động: -40°C ~ 85°C.

3.2.3 Thiết kế mạch điều khiển robot:



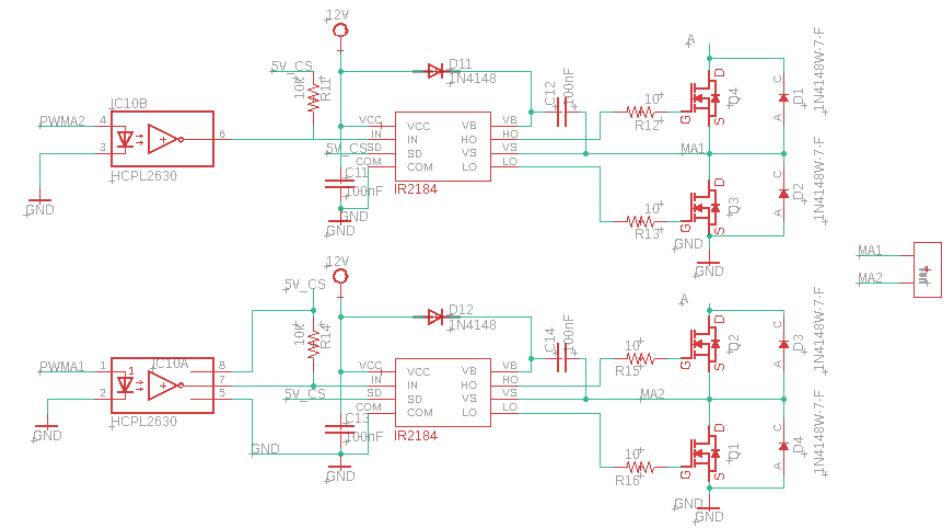
Hình 3. 27. Sơ đồ khái niệm điện điều khiển robot



Hình 3. 28. Sơ đồ nguyên lý của shield điều khiển động cơ DC

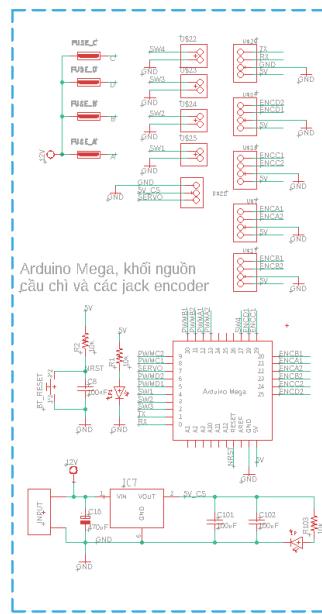
Nguyên lý làm việc:

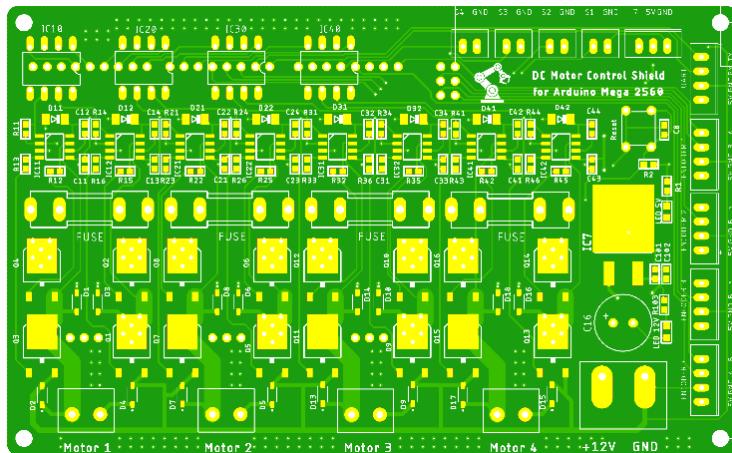
- Khi cấp nguồn, điện áp 12V sẽ đi vào IC 7805 để hạ áp từ 12V xuống 5V cấp nguồn cho các IC của cầu H bao gồm: HCPL2631, IR2184. Các chân PWM của Arduino Mega sẽ được kết nối với các ngõ vào IC cách ly HCPL 2631. Tín hiệu PWM sau khi cách ly sẽ đi vào IC lái IR2184 dùng để điều khiển đóng ngắt các từng cụm 4 mosfet IR3708 từ đó điều khiển được chiều và tốc độ quay của động cơ DC. Các chân ngắt của Arduino Mega được kết nối các jack ra chân để đọc tín hiệu encoder phản hồi về từ động cơ DC. Trong khi điều khiển robot Arduino Mega sẽ gửi các thông số như: các góc theta robot, vị trí hiện tại, sai số bằng chuẩn giao tiếp UART đến board ESP32 để cập nhật lên Firebase.



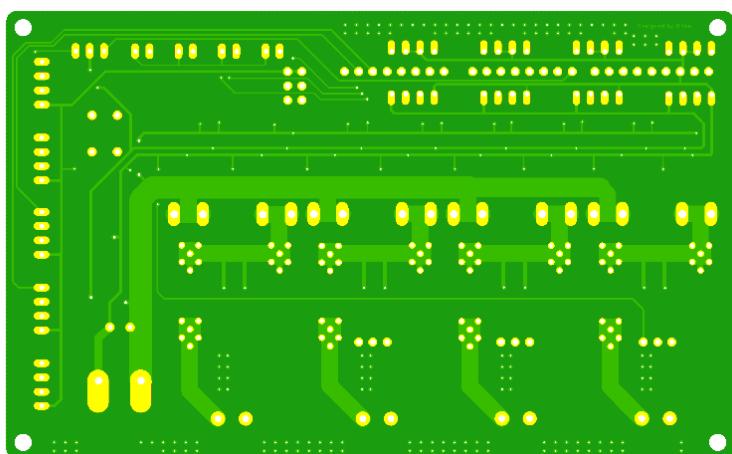
Hình 3. 29. Sơ đồ nguyên lý 1 cụm cầu H điều khiển robot.

- Khối Arduino Mega, nguồn, cầu chì, jack cảm encoder:

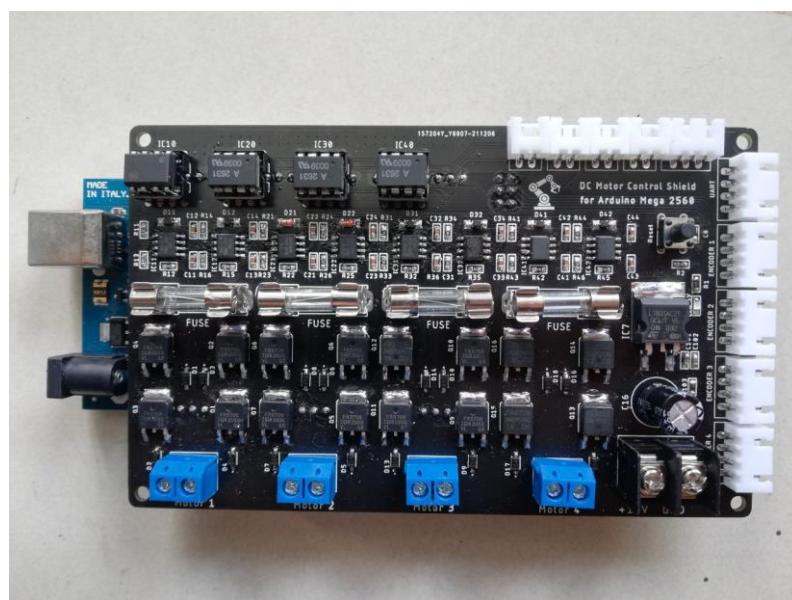




Hình 3. 31. Layout PCB mặt Top



Hình 3. 32. Layout PCB mặt Bottom

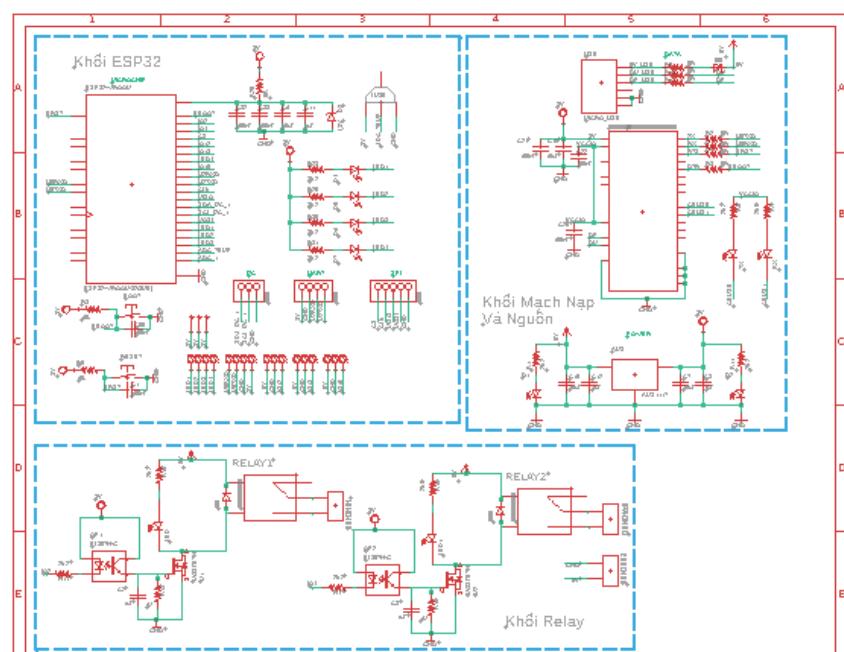


Hình 3. 33. Lắp và hàn linh kiện lên mạch và thử nghiệm, kiểm tra mạch.

STT	Tên linh kiện	Số lượng
1	Bo Arduino Mega	1
2	HCPL	4
3	IR2184	8
4	Cầu chì và đế cầu chì	4
5	Domino	5
6	7805	1
7	XH 2.54 2 chân	4
8	XH 2.54 4 chân	5
9	IR3708	16
10	Diode	24
11	Hàng rào đục	3

Bảng 3. 3. Danh sách linh kiện sử dụng làm mạch điều khiển động cơ robot

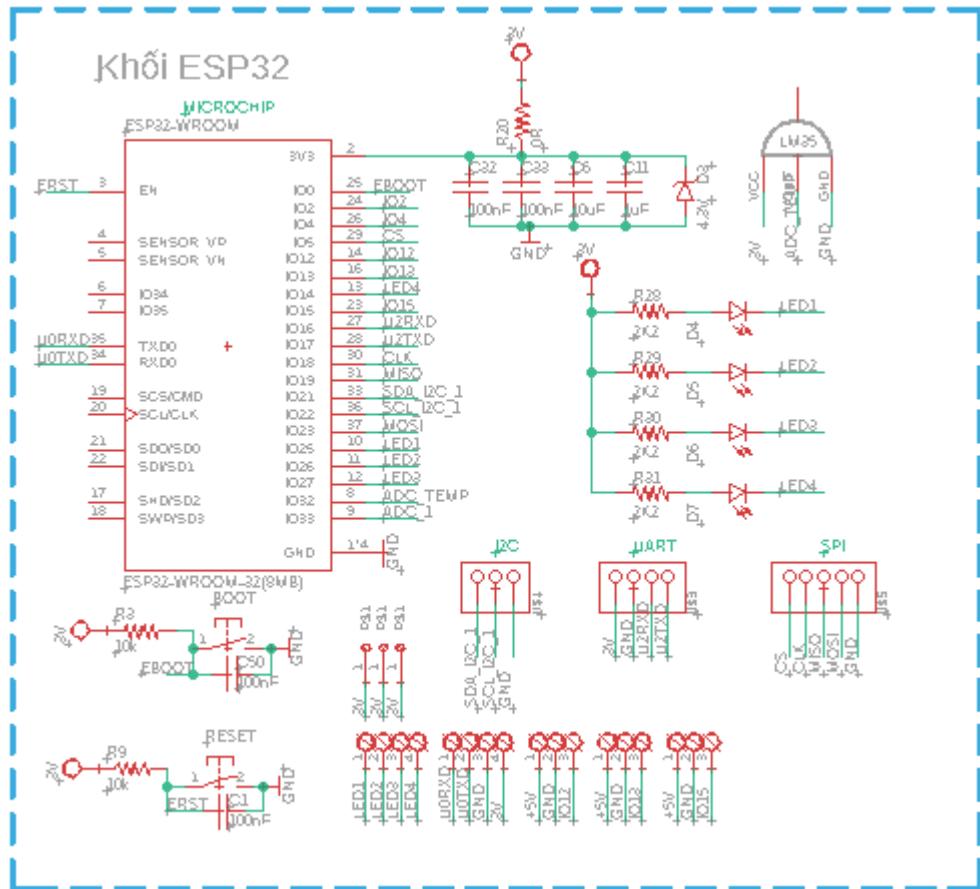
3.2.4 Mạch ESP32 truyền dữ liệu:



Hình 3. 34. Sơ đồ nguyên lý mạch ESP32

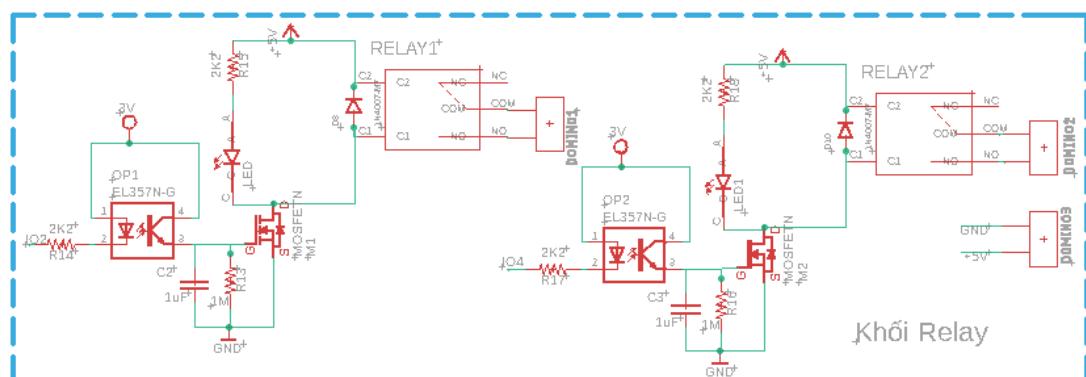
Nguyên lý làm việc: Khi cấp nguồn cho mạch điện áp 5V sẽ đi vào IC AMS1117-3.3V để hạ áp từ 5V xuống 3.3V cấp nguồn cho ESP32. ESP32 được lập trình với các ngõ ra đèn led, relay, tín hiệu UART. ESP32 sẽ nhận dữ liệu các góc theta và tạo độ (px, py, pz) từ Arduino Mega bằng chuẩn giao tiếp UART sau đó cập nhật lên Firebase. Ngoài ra ESP32 còn có chức năng điều khiển tay gắp trên robot.

- Khối MCU sử dụng chip ESP32-WROOM-32E:



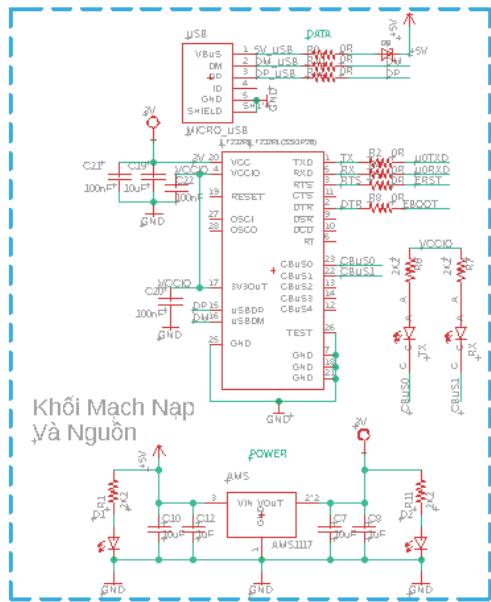
Hình 3. 35. Khối MCU sử dụng chip ESP32-WROOM-32E

- Khối mạch kích relay:

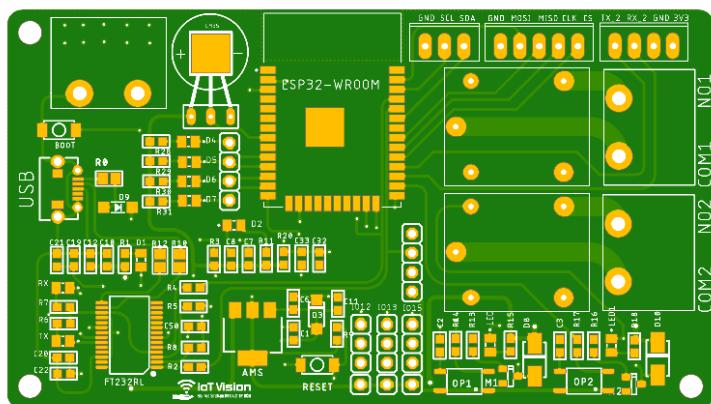


Hình 3. 36. Khối mạch kích relay

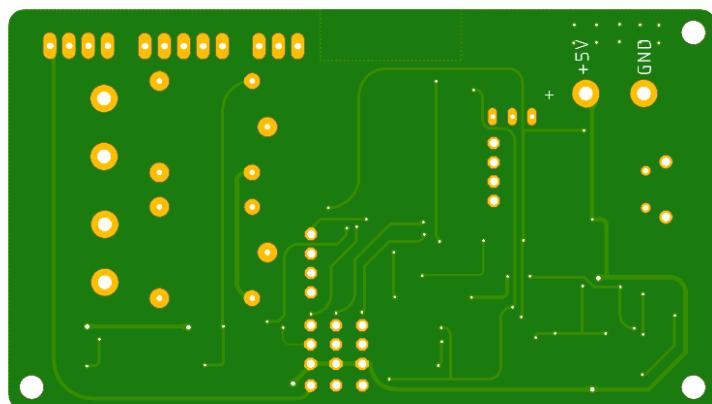
- Khởi mạch nạp code ESP32 và nguồn:



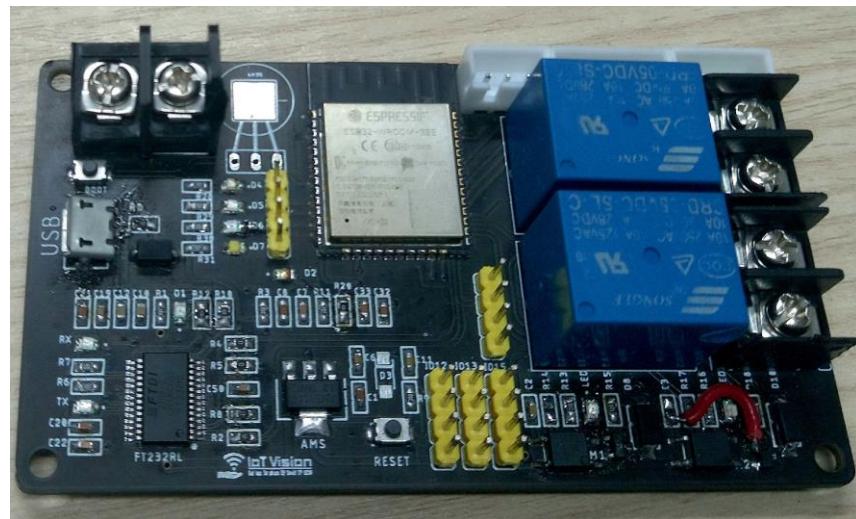
Hình 3. 37. Khởi mạch nạp code ESP32 và nguồn



Hình 3. 38. Layout PCB mặt Top



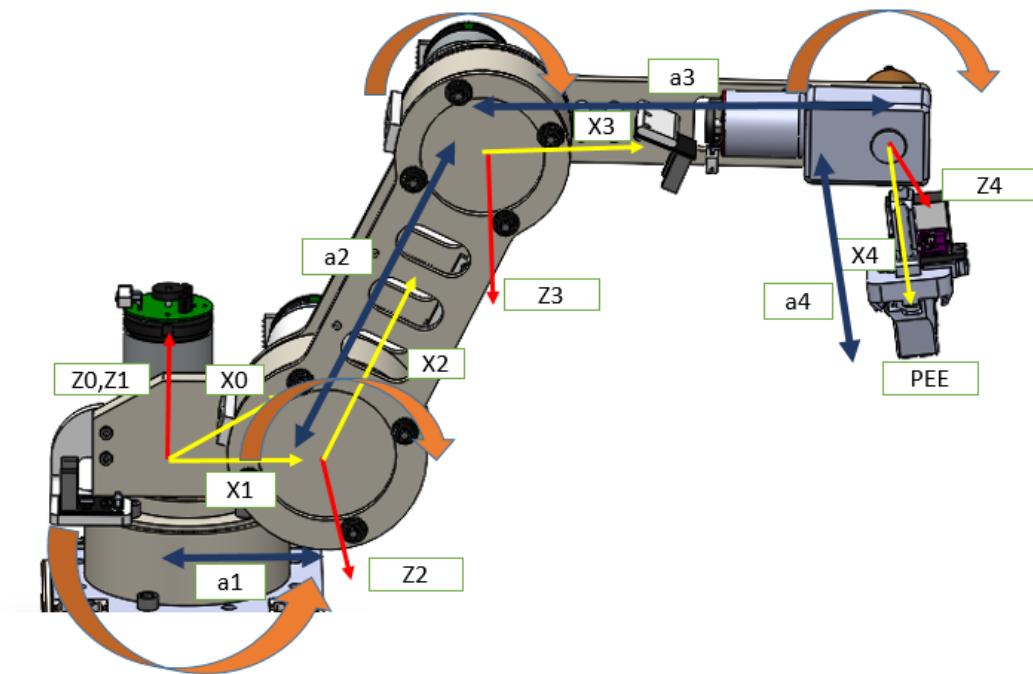
Hình 3. 39. Layout PCB mặt Bottom



Hình 3. 40. Lắp và hàn linh kiện lên mạch và thử nghiệm, kiểm tra mạch

STT	Tên linh kiện	Số lượng
1	ESP32	1
2	Domino	3
3	FT232	1
4	LM1117	1
5	Cổng Micro USB	1
6	Relay	2
7	Nút nhấn	2
8	XH 2.54 4 chân	3
9	Hàng rào đục	5

3.3. Bài toán động học thuận robot



Hình 3. 41. Đặt hệ trục tọa độ tối ưu cho robot

Trong đó:

- a_1 : độ dài từ trục hộp số động cơ 1 đến trục hộp số động cơ 2.
- a_2 : độ dài từ trục hộp số động cơ 2 đến trục hộp số động cơ 3.
- a_3 : độ dài từ trục hộp số động cơ 3 đến trục hộp số động cơ 4.
- a_4 : độ dài từ trục hộp số động cơ 4 đến đầu của tay kẹp.

Thành phần	Kích thước (mm)
a_1	53
a_2	130
a_3	150
a_4	70

Bảng 2. 1. Bảng thông số robot 4 bậc tự do.

Từ các hệ trục tọa độ đã đặt nhóm sẽ tiến hành lập bảng thông số D-H cho robot 4 bậc tự do.

Khâu	\mathbf{a}	α	\mathbf{d}	θ
1	0	0	0	$*\theta_1$
2	a_1	90°	0	$*\theta_2$
3	a_2	0	0	$*\theta_3$
4	a_3	0	0	$*\theta_4$
5	a_4	0	0	0

Bảng 2.2. Bảng thông số D-H của robot

Ta có:

a_i là khoảng cách \hat{Z}_i tới \hat{Z}_{i+1} được đo dọc theo trục \hat{X}_{i+1}

α_i là góc từ \hat{Z}_i tới \hat{Z}_{i+1} được đo dọc theo trục \hat{X}_{i+1}

d_i là khoảng cách \hat{X}_{i-1} tới \hat{X}_i dọc theo trục \hat{Z}_{i-1}

θ_i là góc từ \hat{X}_{i-1} tới \hat{X}_i được đo theo trục \hat{Z}_{i-1}

Sau khi có được bảng thông số D-H, nhóm sử dụng ma trận A_n theo giáo trình “Introduction to Robotics” của John J. Craig [2].

$$A_n = \begin{bmatrix} \cos \theta_n & -\sin \theta_n & 0 & a_{n-1} \\ \sin \theta_n \cdot \cos \alpha_{n-1} & \cos \theta_n \cdot \cos \alpha_{n-1} & -\sin \alpha_{n-1} & -\sin \alpha_{n-1} \cdot d_n \\ \sin \theta_n \cdot \sin \alpha_{n-1} & \cos \theta_n \cdot \sin \alpha_{n-1} & \cos \alpha_{n-1} & \cos \alpha_{n-1} \cdot d_n \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ta kí hiệu ma trận chuyển đổi từ khâu $n-1$ sang n như sau:

$${}^{n-1}T = A_n$$

Quy ước:

$$s_1 = \sin \theta_1; c_1 = \cos \theta_1; s_{12} = \sin(\theta_1 + \theta_2); c_{12} = \cos(\theta_1 + \theta_2)$$

$$s_2 = \sin \theta_2; c_2 = \cos \theta_2; s_{23} = \sin(\theta_2 + \theta_3); c_{23} = \cos(\theta_2 + \theta_3)$$

$$s_{234} = \sin(\theta_2 + \theta_3 + \theta_4); c_{234} = \cos(\theta_2 + \theta_3 + \theta_4)$$

- Ma trận chuyển đổi từ hệ 0 sang hệ 1:

$${}^0_1T = \begin{bmatrix} c_1 & 0 & -s_1 & a_1 * c_1 \\ s_1 & 0 & c_1 & a_1 * s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Ma trận chuyển đổi từ hệ 1 sang hệ 2:

$${}^1_2T = \begin{bmatrix} c_2 & -s_2 & 0 & a_2 * c_2 \\ s_2 & c_2 & 0 & a_2 * s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Ma trận chuyển đổi từ hệ 2 sang hệ 3:

$${}^2_3T = \begin{bmatrix} c_3 & -s_3 & 0 & a_3 * c_3 \\ s_3 & c_3 & 0 & a_3 * s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Ma trận chuyển đổi từ hệ 3 sang hệ 4:

$${}^3_4T = \begin{bmatrix} c_4 & -s_4 & 0 & a_4 * c_4 \\ s_4 & c_4 & 0 & a_4 * s_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Ma trận chuyển đổi từ hệ 0 sang hệ 4 [11]:

$$\begin{aligned} {}^0_4T &= {}^0_1T * {}^1_2T * {}^2_3T * {}^3_4T \\ &= \begin{bmatrix} c_{234} * c_1 & -s_{234} * c_1 & -s_1 & c_1 * (a_1 + a_3 * c_{23} + a_2 * c_2 + a_4 * c_{234}) \\ c_{234} * s_1 & -s_{234} * s_1 & c_1 & s_1 * (a_1 + a_3 * c_{23} + a_2 * c_2 + a_4 * c_{234}) \\ -s_{234} & -c_{234} & 0 & a_3 * s_{23} + a_2 * s_2 + a_4 * s_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

Với hệ phương trình động học thuận, vị trí điểm cuối của chân robot sẽ có hệ phương trình như sau :

$$P_x = c_1 * (a_1 + a_3 * c_{23} + a_2 * c_2 + a_4 * c_{234})$$

$$P_y = s_1 * (a_1 + a_3 * c_{23} + a_2 * c_2 + a_4 * c_{234})$$

$$P_z = a_3 * s_{23} + a_2 * s_2 + a_4 * s_{234}$$

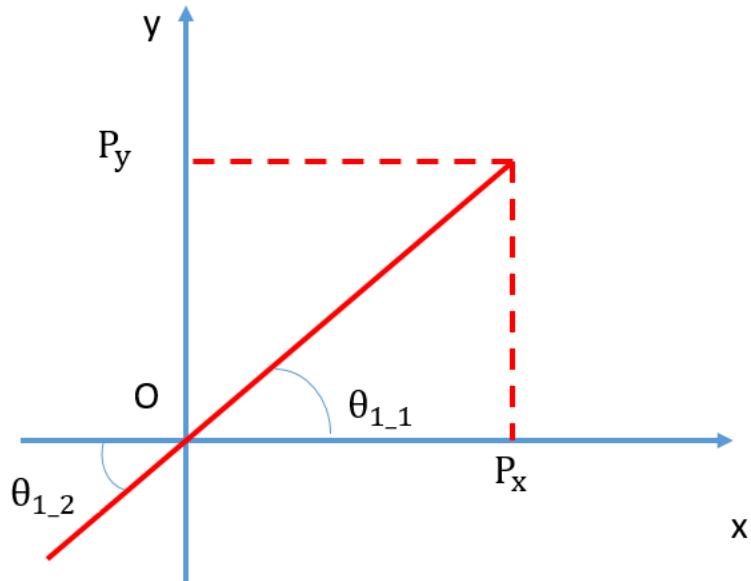
3.4. Bài toán động học nghịch robot

Bài toán động học nghịch nhằm xác định giá trị các biến khớp khi biết được vị trí và hướng của chân robot.

Để có thể đơn giản bài toán động học nghịch nhóm em lựa chọn và áp dụng phương pháp hình học kết hợp phương pháp đại số.

* *Tính θ_1 :*

Thực hiện phép chiếu lên mặt phẳng Oxy, nhóm em thu được góc θ_1 bằng định lý Pytago trong tam giác [12].



Hình 3. 42. Phép chiếu trên mặt phẳng Oxy

Từ hình chiếu này nhóm em sẽ tính được hai giá trị của θ_1 là:

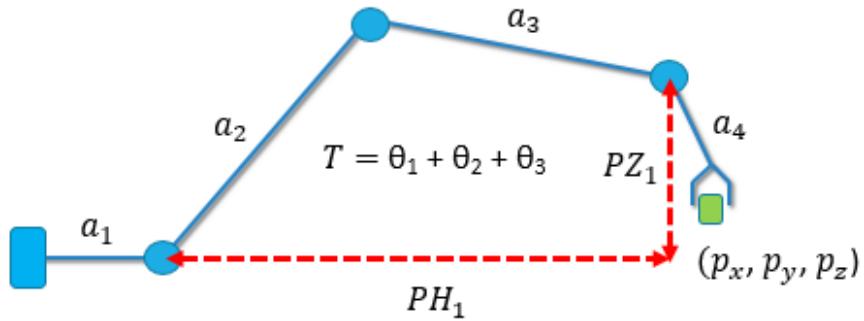
$$\theta_{1_1} = \text{atan2}(p_y, p_x)$$

$$\theta_{1_2} = \text{atan2}(p_y, p_x) - \pi$$

* *Tính θ_2, θ_3 và θ_4 :*

Sau khi đã tính được góc theta 1, nhóm em sẽ chiếu robot lên trên mặt phẳng song song với các link a_1, a_2, a_3, a_4 . Robot được thiết kế tay gấp ở cuối link 4 nhằm thực hiện cho mục đích gấp linh kiện. Để thuận tiện cho việc này, nhóm em sẽ đặt một hằng số T gọi là góc tạo bởi tay kẹp và linh kiện cần gấp. Hằng số sẽ được điều chỉnh tùy theo mỗi loại linh kiện cần gấp để tăng khả năng gấp được linh kiện và hằng số này chính là tổng của 3 góc: $\theta_1, \theta_2, \theta_3$.

Đối với trường hợp $\theta_{1_1} = \text{atan2}(p_y, p_x)$ thì nhóm em sẽ có hình chiếu:



Hình 3. 43. Hình chiếu của robot trường hợp $\theta_{1_1} = \text{atan2}(p_y, p_x)$

PH_1 chính là độ dài hình chiếu của link 2 và link 3 lên trên mặt phẳng Oxy.

$$PH_1 = \sqrt{p_x^2 + p_y^2} - a_1 - a_4 * \cos(T).$$

PZ_1 chính là độ cao của của khớp 4 so với khớp 2.

$$PZ_1 = p_z - h - a_4 * \sin(T)$$

Sau khi đã tính được PH_1 và PZ_1 lúc này nhóm em sẽ giải bài toán robot 2 bậc tự do trong cùng một mặt phẳng để tìm góc θ_2 và góc θ_3 . Đầu tiên ta sẽ kiểm tra xem điểm PEE mong muốn có nằm trong không gian hoạt động của robot hay không :

Nếu $\sqrt{PH_1^2 + PZ_1^2} > a_2 + a_3$ (độ dài link 2 và link 3) thì điểm đặt (p_x, p_y, p_z) đã nằm ngoài không gian làm việc của robot.

Nếu $\sqrt{PH_1^2 + PZ_1^2} < a_2 + a_3$ (độ dài link 2 và link 3) thì điểm đặt (p_x, p_y, p_z) nằm trong không gian làm việc của robot. Lúc này nhóm em sẽ sử dụng công thức Basic_01 để giải động học robot và tìm ra 2 bộ nghiệm của góc θ_2 và góc θ_3 . Khi đã có được góc θ_2 và góc θ_3 kết hợp với hằng số góc T biết trước nhóm em sẽ tính được góc θ_4 .

Giải bộ nghiệm 1:

$$A = 2 * PH_1 * a_2$$

$$B = 2 * PZ_1 * a_2$$

$$D = PH_1^2 + PZ_1^2 + a_2^2 - a_3^2$$

$$\theta_{2_1} = \text{anp} + \text{atan2}\left(\sqrt{1 - \frac{D^2}{A^2 + B^2}}, \frac{D}{\sqrt{A^2 + B^2}}\right)$$

$$\theta_{3_1} = \text{atan2}(PZ_1 - a_2 * \sin(\theta_{2_1}), PH_1 - a_2 * \cos(\theta_{2_1}))$$

$$\theta_{4_1} = T - \theta_{2_1} - \theta_{3_1}$$

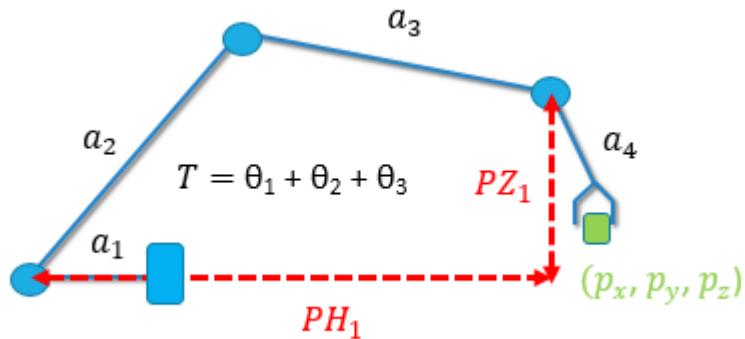
Giải bộ nghiệm 2:

$$\theta_{2_2} = \text{atan2}\left(-\sqrt{1 - \frac{D^2}{A^2+B^2}}, \frac{D}{\sqrt{A^2+B^2}}\right)$$

$$\theta_{3_2} = \text{atan2}(PZ_1 - a_2 * \sin(\theta_{2_2}), PH_1 - a_2 * \cos(\theta_{2_2}))$$

$$\theta_{4_2} = T - \theta_{2_2} - \theta_{3_2}$$

Đối với trường hợp $\theta_{1_1} = \text{atan2}(P_y, P_x) - \pi$ thì ta sẽ có hình chiếu:



Hình 3. 44. Hình chiếu của robot trường hợp $\theta_{1_1} = \text{atan2}(P_y, P_x) - \pi$
 PH_1 chính là độ dài hình chiếu của link 2 và link 3 lên mặt phẳng Oxy.

$$PH_1 = \sqrt{P_x^2 + P_y^2} + a_1 + a_4 * \cos(T)$$

PZ_1 chính là độ cao của khớp 4 so với khớp 2.

$$PZ_1 = p_z - h - a_4 * \sin(T)$$

Sau khi đã tính được PH_1 và PZ_1 lúc này ta sẽ giải bài toán robot 2 bậc tự do trong cùng một mặt phẳng để tìm góc θ_2 và góc θ_3 . Đầu tiên nhóm em sẽ kiểm tra xem điểm PEE mong muốn có nằm trong không gian hoạt động của robot hay không :

Nếu $\sqrt{PH_1^2 + PZ_1^2} > a_2 + a_3$ (độ dài link 2 và link 3) thì điểm đặt (p_x, p_y, p_z) đã nằm ngoài không gian làm việc của robot.

Nếu $\sqrt{PH_1^2 + PZ_1^2} < a_2 + a_3$ (độ dài link 2 và link 3) thì điểm đặt (p_x, p_y, p_z) nằm trong không gian làm việc của robot. Lúc này nhóm em sẽ sử dụng Basic_01 để giải động học robot.

Giải bộ nghiệm 3:

$$A = 2 * PH_1 * a_2$$

$$B = 2 * PZ_1 * a_2$$

$$D = PH_1^2 + PZ_1^2 + a_2^2 - a_3^2$$

$$\theta_{2_3} = \text{anp} + \text{atan2}\left(\sqrt{1 - \frac{D^2}{A^2+B^2}}, \frac{D}{\sqrt{A^2+B^2}}\right)$$

$$\theta_{3_3} = \text{atan2}(PZ_1 - a_2 * \sin(\theta_{2_3}), PH_1 - a_2 * \cos(\theta_{2_3}))$$

$$\theta_{4_3} = T - \theta_{2_3} - \theta_{3_3}$$

Giải bộ nghiệm 4:

$$\theta_{2_4} = \text{anp} + \text{atan2}\left(-\sqrt{1 - \frac{D^2}{A^2+B^2}}, \frac{D}{\sqrt{A^2+B^2}}\right)$$

$$\theta_{3_4} = \text{atan2}(PZ_1 - a_2 * \sin(\theta_{2_4}), PH_1 - a_2 * \cos(\theta_{2_4}))$$

$$\theta_{4_4} = T - \theta_{2_4} - \theta_{3_4}$$

Tổng hợp tất cả lại tối đa chúng nhóm em sẽ có được 4 bộ nghiệm để điều khiển robot:

Bộ nghiệm 1: $\theta_{1_1}, \theta_{2_1}, \theta_{3_1}, \theta_{4_1}$

Bộ nghiệm 2: $\theta_{1_1}, \theta_{2_2}, \theta_{3_2}, \theta_{4_2}$

Bộ nghiệm 3: $\theta_{1_2}, \theta_{2_3}, \theta_{3_3}, \theta_{4_3}$

Bộ nghiệm 4: $\theta_{1_2}, \theta_{2_4}, \theta_{3_4}, \theta_{4_4}$

3.4. Mô phỏng kiểm chứng trên matlab

Sử dụng phần mềm Matlab để hỗ trợ việc tính toán và mô phỏng.

Mô phỏng kiểm chứng động học thuận:

- Trường hợp 1:

Khớp thứ i	Thông số	Đơn vị
1	0	Độ
2	90	Độ
3	0	Độ
4	0	Độ

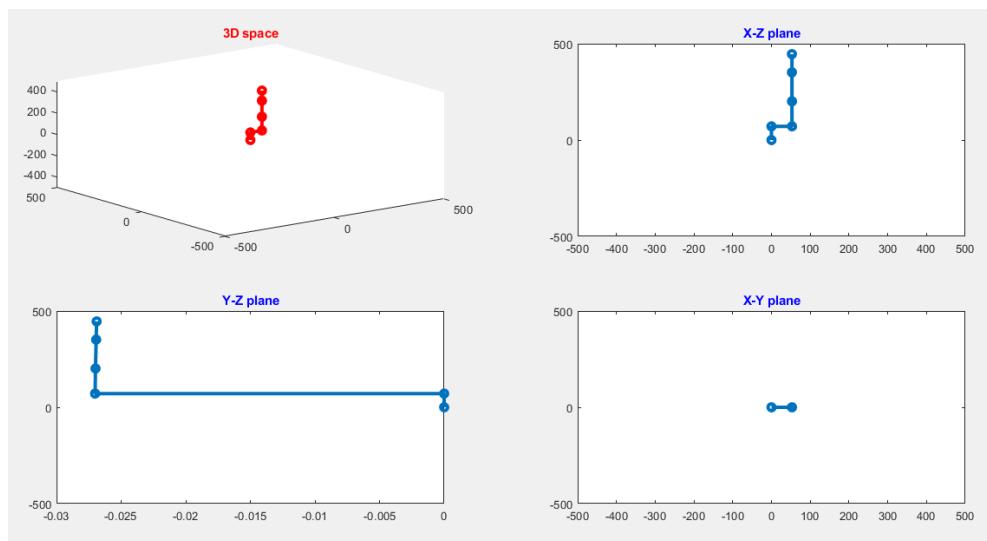
Bảng 3. 4. Thông số đặt các khớp trường hợp 1

Kết quả trên Matlab:

```
>> PEE = F_4dof_TT(0,pi/2,0,0,d1,12,13,14,h)

PEE =
53.0000      0  421.1300
```

Hình 3. 45. Kết quả tính động học thuận trường hợp 1



Hình 3. 46. Hình vẽ robot động học thuận trường hợp 1

- Trường hợp 2:

Khớp thứ i	Thông số	Đơn vị
1	90	Độ
2	0	Độ
3	90	Độ
4	0	Độ

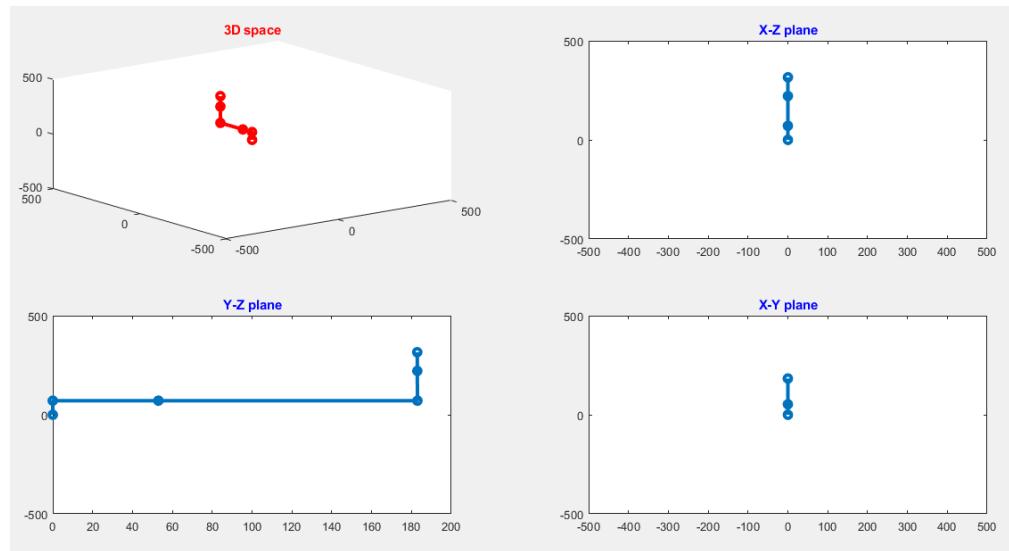
Bảng 3. 5. Kết quả tính động học thuận trường hợp 2

Kết quả trên Matlab:

```
>> PEE = F_4dof_TT(pi/2,0,pi/2,0,d1,12,13,14,h)

PEE =
0.0000  183.0000  291.1300
```

Hình 3. 47. Kết quả tính động học thuận trường hợp 2



Hình 3. 48. Kết quả hình vẽ robot trường hợp 2

Mô phỏng kiểm chứng động học nghịch:

- Trường hợp 1:

Nhập vị trí đặt:

$P_x = 180$	$P_y = 60$	$P_z = 140$
-------------	------------	-------------

Bảng 3. 6. Vị trí đặt p_x, p_y, p_z của trường hợp 1

BoNghiem_1 =

0.3218 1.6707 -1.6072 -1.6343

BoNghiem_2 =

0.3218 -0.0844 1.6072 -3.0936

BoNghiem_3 =

-2.8198 2.5685 0.0998 -4.2391

BoNghiem_4 =

-2.8198 2.6754 -0.0998 -4.1464

Hình 3. 49. Các bộ nghiệm robot của trường hợp 1

```

P_EE_BoNghiem1 =
180.0000  60.0000  140.0000

P_EE_BoNghiem2 =
180.0000  60.0000  140.0000

P_EE_BoNghiem3 =
180.0000  60.0000  140.0000

P_EE_BoNghiem4 =
180.0000  60.0000  140.0000

```

Hình 3. 50. Dùng động học thuận để kiểm chứng các bộ nghiệm trường hợp 1

- Trường hợp 2:

Nhập vị trí đặt:

$P_x = -271$	$P_y = -50$	$P_z = 50$
--------------	-------------	------------

Bảng 3. 7. Vị trí đặt p_x, p_y, p_z của trường hợp 2

```

BoNghiem_1 =
0.1825  0.8892  -1.2438  -1.2162

BoNghiem_2 =
0.1825  -0.4569  1.2438  -2.3577

P_EE_BoNghiem1 =
271.0000  50.0000  50.0000

P_EE_BoNghiem2 =
271.0000  50.0000  50.0000

```

Hình 3. 51. Dùng động học thuận để kiểm chứng các bộ nghiệm trường hợp 2

CHƯƠNG 4: THIẾT KẾ ỨNG DỤNG VÀ GIAO DIỆN ĐIỀU KHIỂN, GIÁM SÁT ROBOT

4.1. Thiết kế ứng dụng điều khiển và giám sát Robot

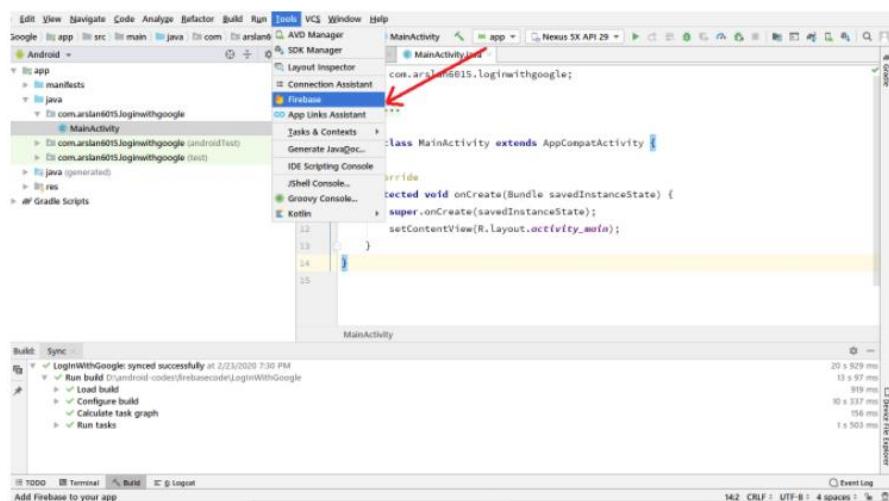
Mô tả ứng dụng: App cho phép người dùng điều khiển và giám sát hoạt động của robot.

4.1.1. Hướng dẫn sử dụng cơ sở dữ liệu Firebase

Tạo xác thực người dùng trên Firebase:

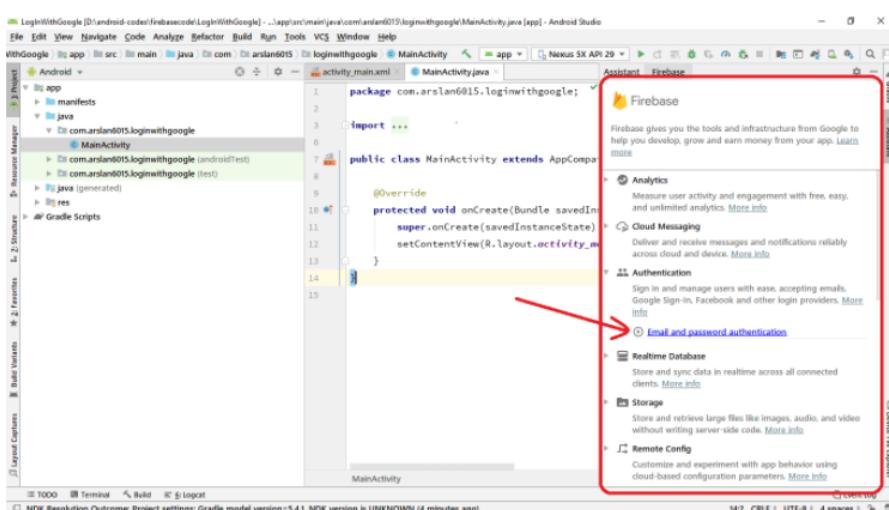
Cách bước tiến hành tạo xác thực người dùng trên Firebase:

- Tiến hành thêm firebase cho Project (**Tools -> Firebase**)



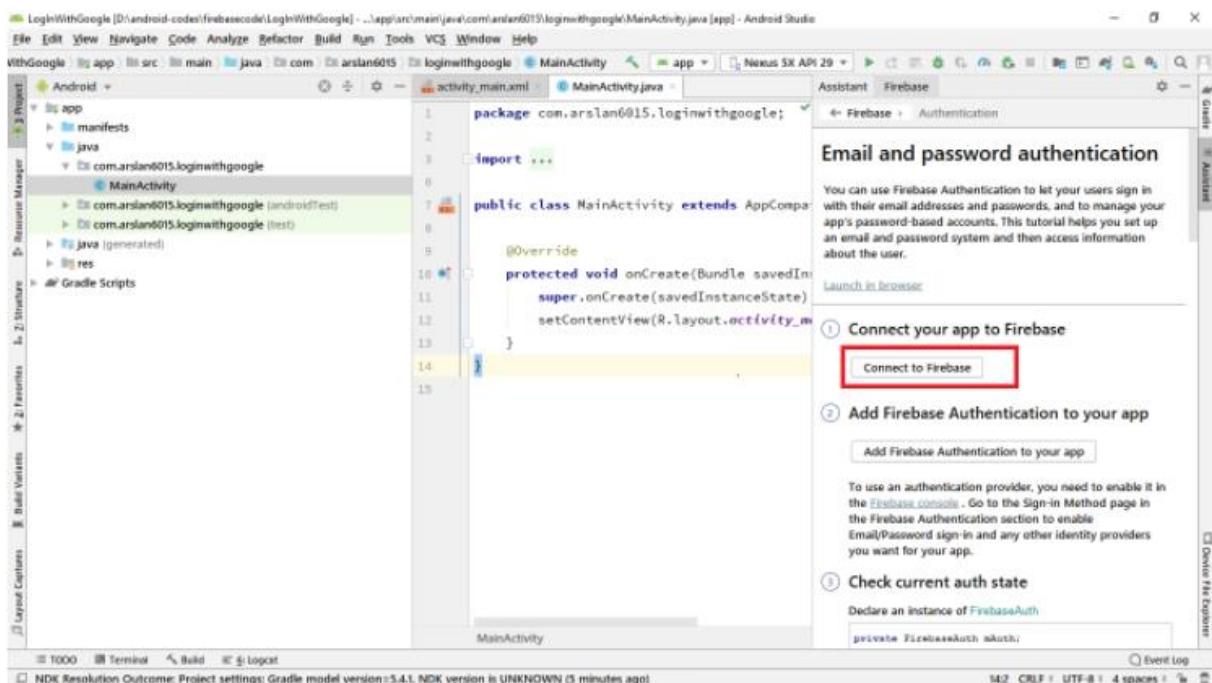
Hình 4. 1. Thêm firebase cho project

- Sau đó chọn **Email and Password Authentication** trong mục Authentication.



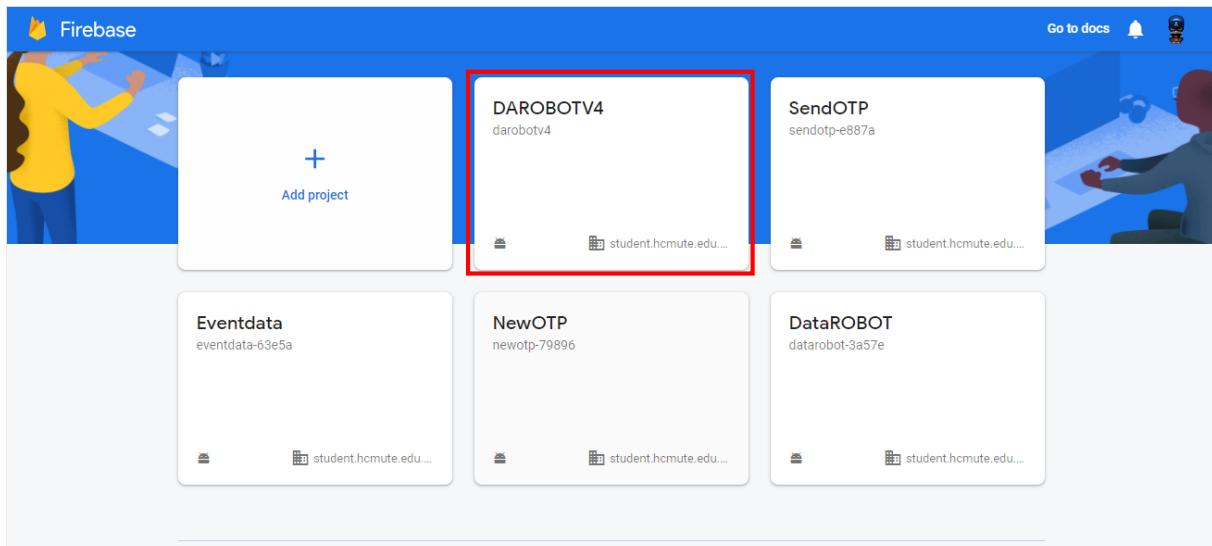
Hình 4. 2. Chọn Email and Password Authentication trong mục Authentication

- Tiến hành liên kết mới firebase.



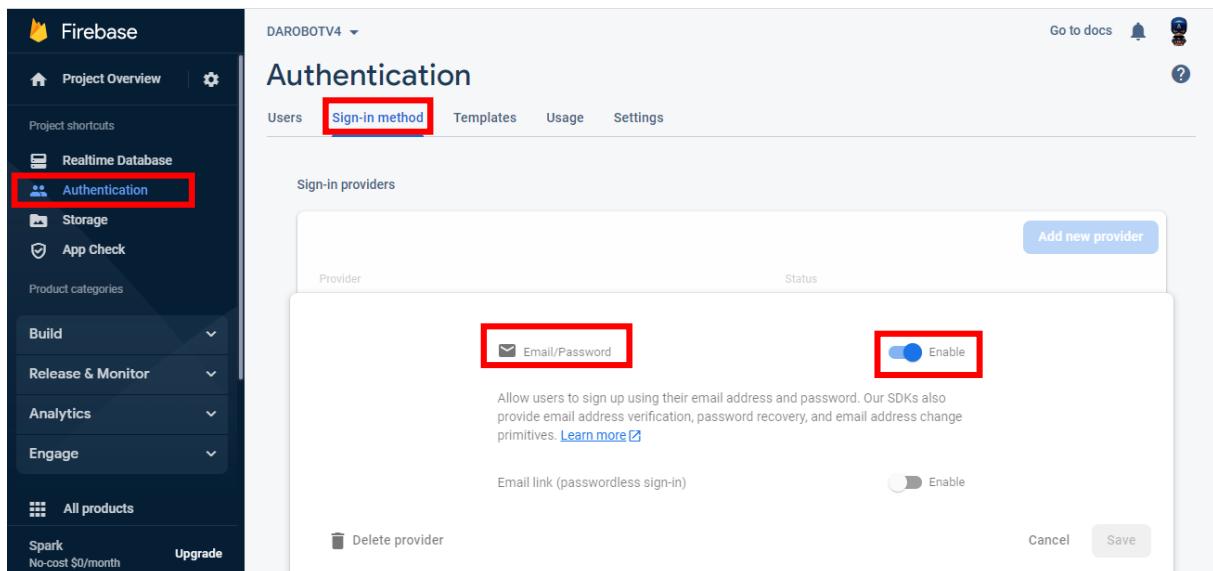
Hình 4. 3. Tiến hành liên kết mới firebase

- Tiến hành tạo vào đăng nhập firebase với tài khoản google.
- Sau khi liên kết xong, đăng nhập vào tài khoản firebase tạo project mới rồi nhấn vào project để bật Firebase Authentication.



Hình 4. 4. Bật Firebase Authentication

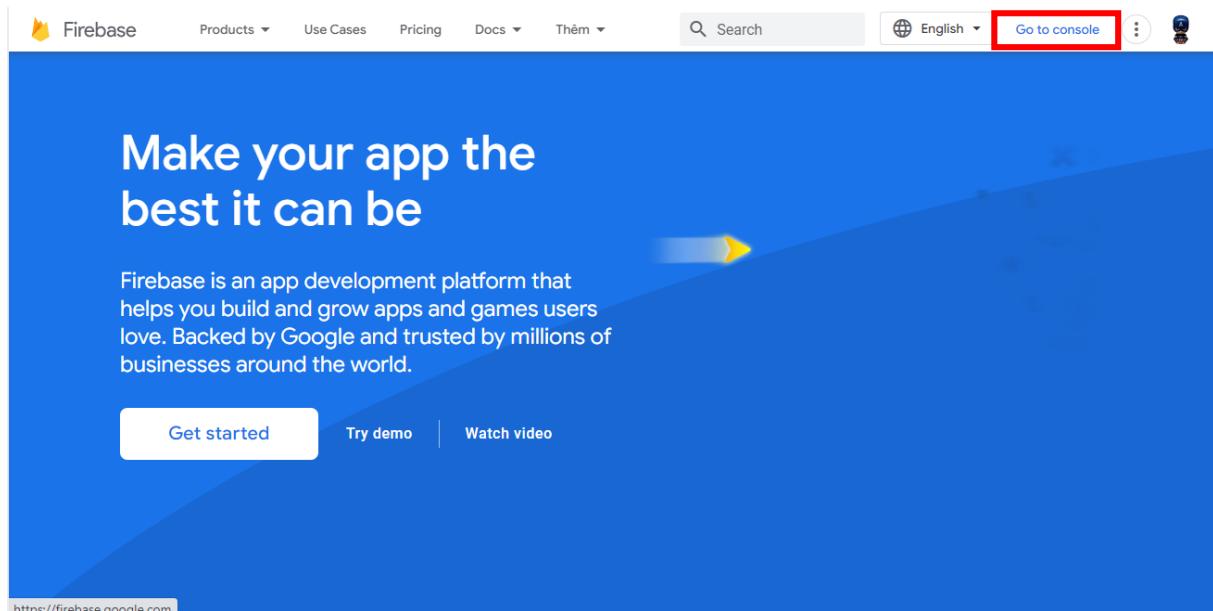
- Chọn mục Authentication -> Sign-In method -> Chọn mục Email/Password -> chuyển nút nhấn trạng thái sang Enable để tiến hành bật.



Hình 4. 5. Chuyển nút nhấn trạng thái sang Enable để tiến hành bật

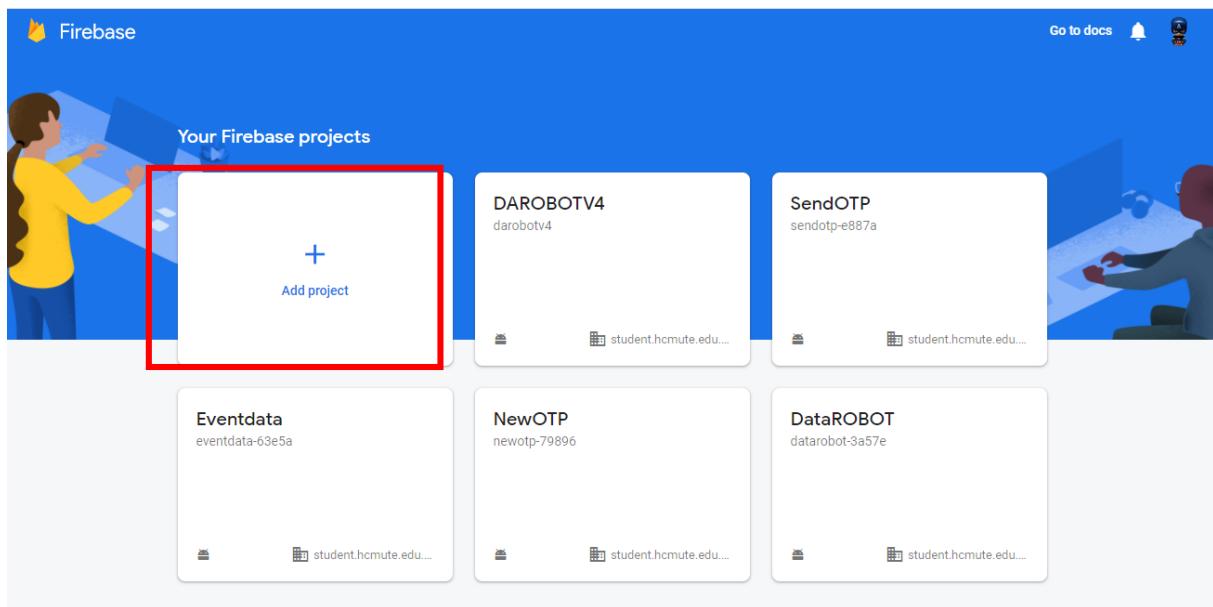
Sau khi lưu thì đã có thể lập trình để có thể tạo tài khoản bằng Email/ Password cho project. Các bước tiến hành tạo cơ sở dữ liệu Realtime Database trên Firebase:

- Tạo project mới trên Firebase. Để bắt đầu sử dụng Firebase chúng ta truy cập vào trang chủ Firebase đăng nhập bằng gmail của mình.



Hình 4. 6. Truy cập vào trang chủ Firebase

- Sau đó ta nhấn **Go to console** và chọn **Add project** để bắt đầu dự án mới.



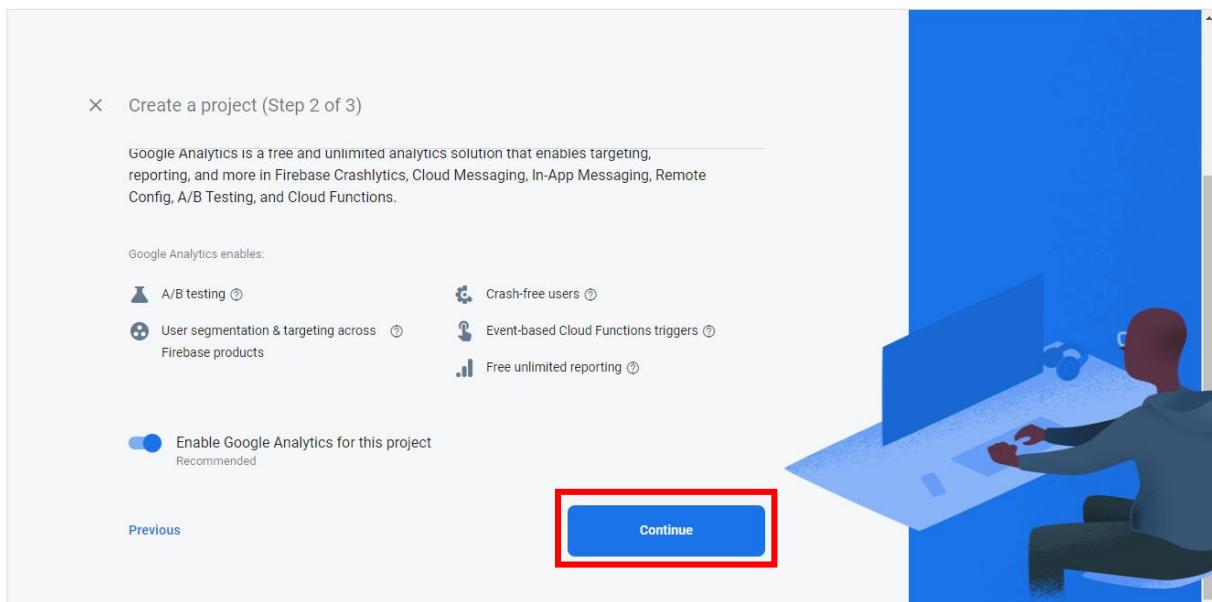
Hình 4. 7. Add project để bắt đầu dự án mới

- Đặt tên cho project.



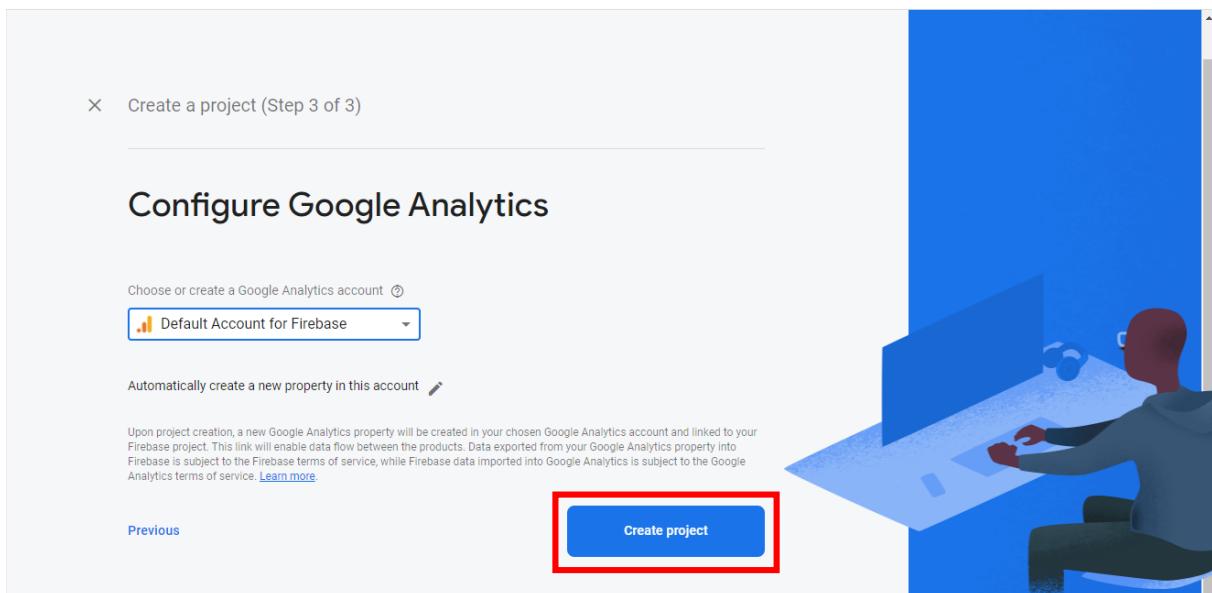
Hình 4. 8. Đặt tên cho project

- Nhấn Continue để chuyển sang trang mới.



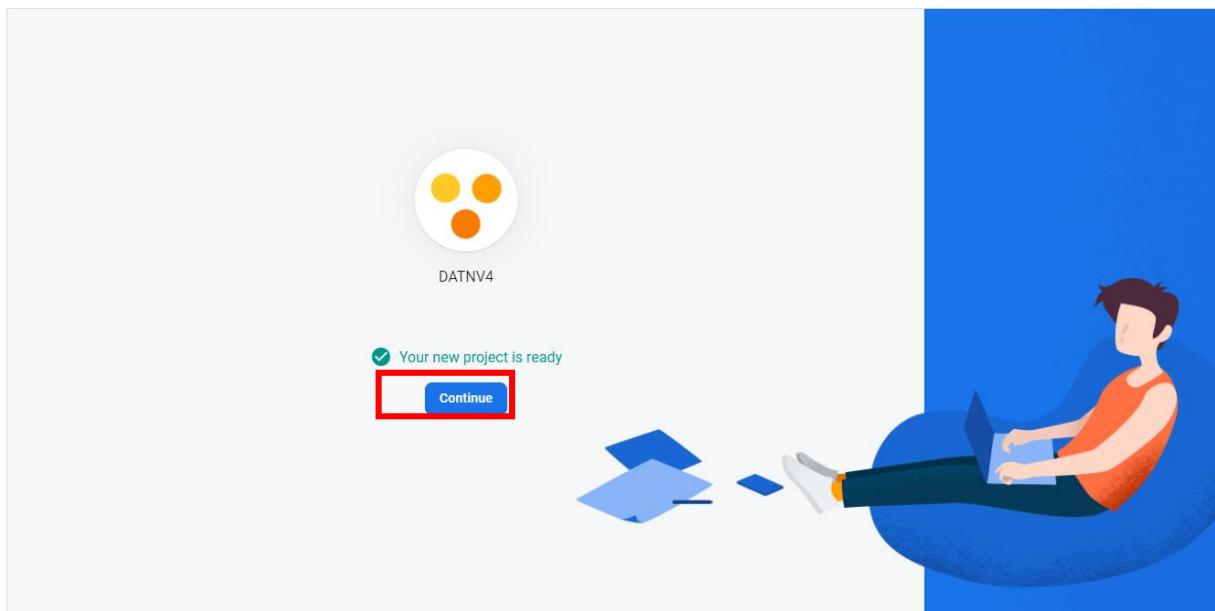
Hình 4. 9. Nhấn continue để chuyển sang trang mới

- Tiếp theo chọn Default Account for Firebase và Create project.



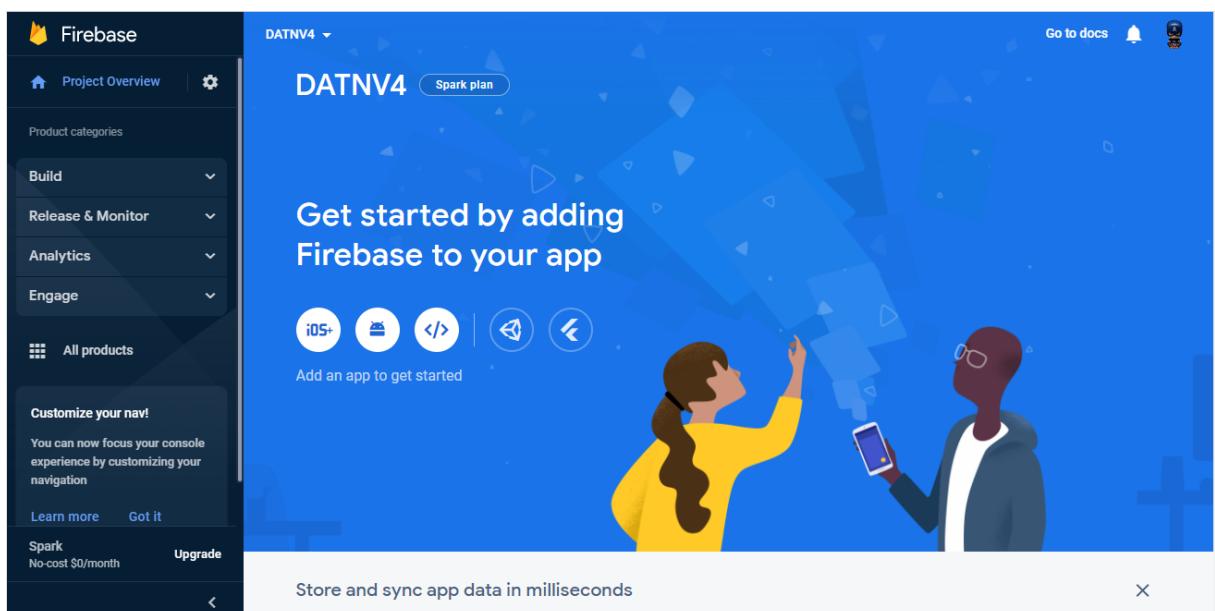
Hình 4. 10. Chọn Default Account for Firebase và Create project

- Sau đó ta đợi Firebase khởi tạo project của chúng ta rồi nhấn Continue để hoàn thành.



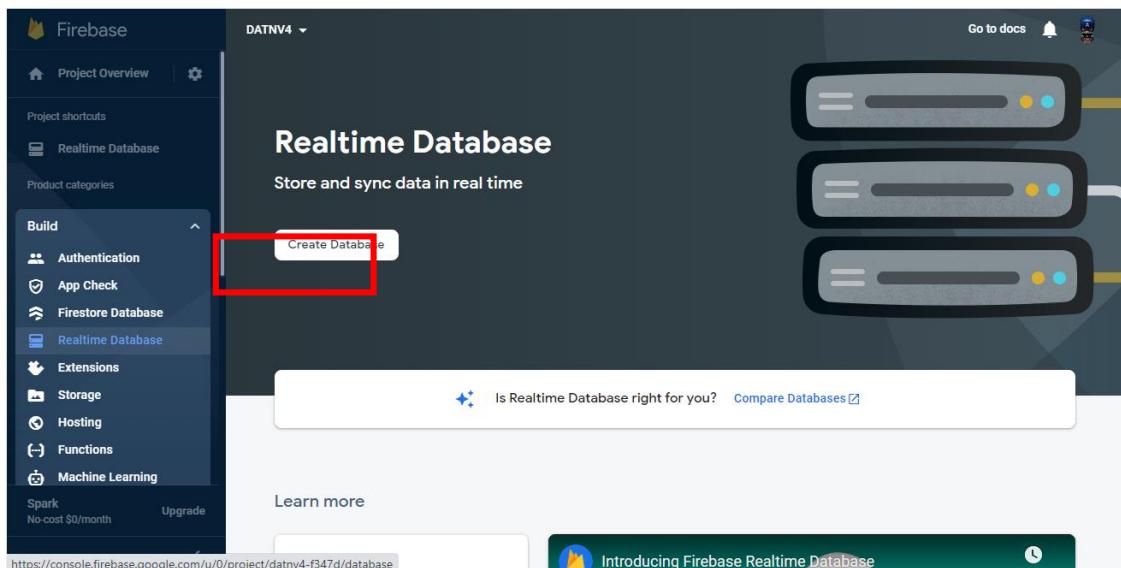
Hình 4. 11. Nhấn Continue để hoàn thành

- Cuối cùng Firebase sẽ dẫn ta đến giao diện như sau:



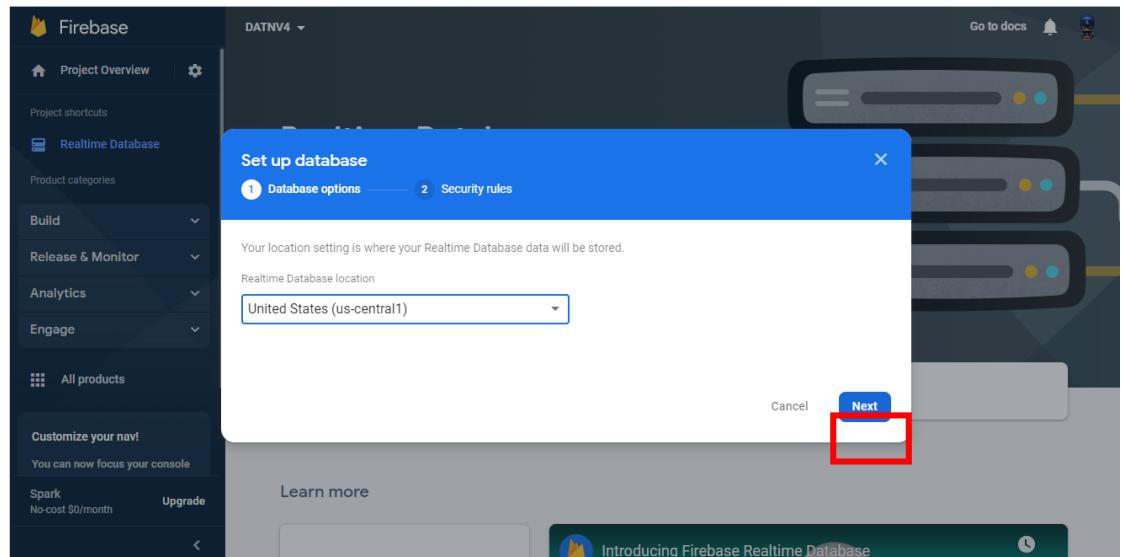
Hình 4. 12. Giao diện fire base

Để khởi tạo và cài đặt Firebase Realtime Database ta sẽ sử dụng Services Realtime Database của Firebase với mục đích làm cơ sở dữ liệu để các client có thể tương tác qua lại. Đầu tiên ta Create Realtime Database.



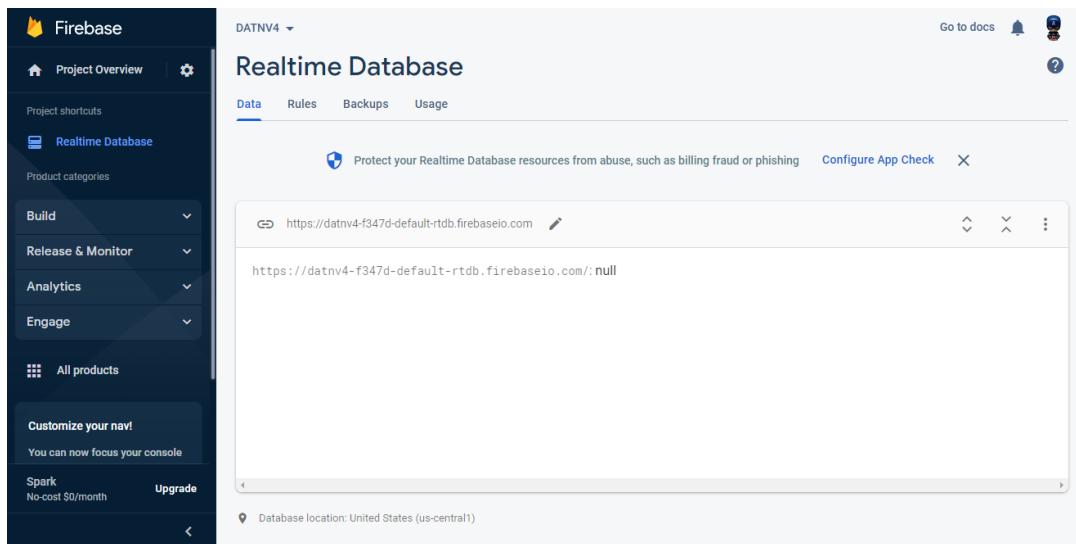
Hình 4. 13. Create Realtime Database

- Để ché độ mặc định và bấm Next.



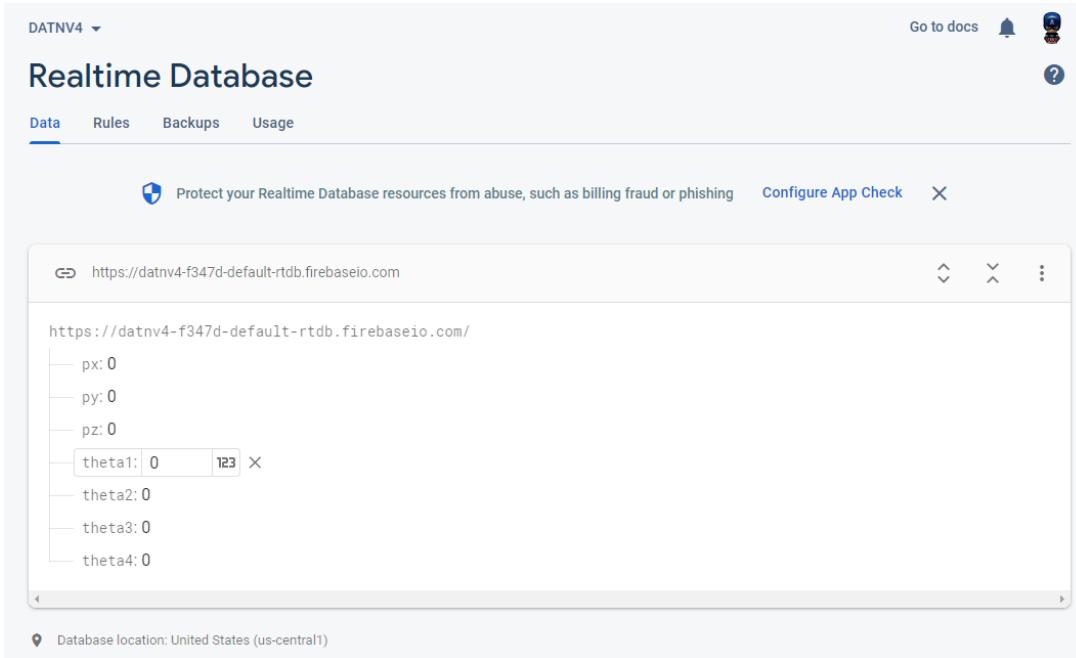
Hình 4. 14. Để ché độ mặc định và bấm Next

Bước tiếp theo chọn 2 option cho project: nếu đang phát triển dự án với mục đích để test các chức năng thì chọn test mode, nhưng database sẽ ở trạng thái open trong 30 ngày đầu tiên. Ngược lại khi cần một database để phục vụ cho mobile app hoặc các ứng dụng web chính thức thì chọn locked mode và database sẽ ở chế độ private và dưới các rules đã được thiết lập. Giao diện của Database của chúng ta hiện tại chưa có một trường dữ liệu và dữ liệu nào.



Hình 4. 15. Giao diện của Database chưa có dữ liệu

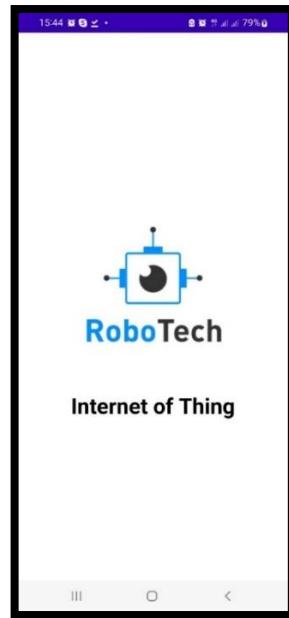
Tiếp theo ta sẽ tạo ra trường dữ liệu và gán dữ liệu theo cách thủ công bằng cách nhấn vào dấu cộng (add child). Dữ liệu có thể là kiểu int, string, float... tùy theo dữ liệu ban đầu bạn nhập vào. Ở project này mình sẽ tạo các dữ liệu chứa các vị trí và các góc theta của Robot.



Hình 4. 16. Tạo các dữ liệu chứa các vị trí và các góc theta của robot.

4.1.2. Tổng quan giao diện ứng dụng.

a. Màn hình giới thiệu

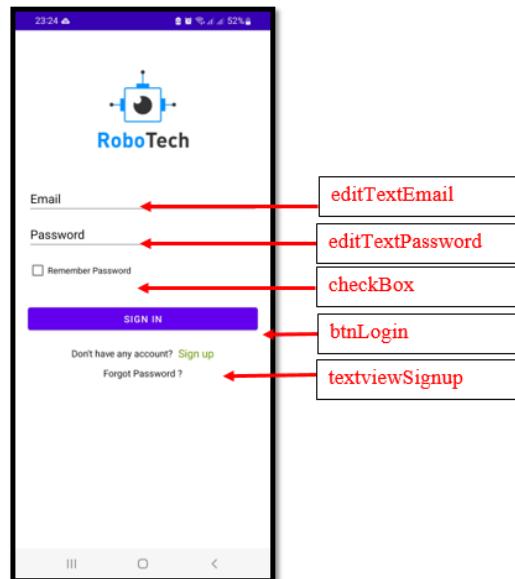


Hình 4. 17. Hình ảnh màn hình giới thiệu

Khi bật vào ứng dụng màn hình giới thiệu sẽ tiến hành chạy trong 2 giây. Sau đó sẽ chuyển vào màn hình đăng nhập để vào ứng dụng.

b. Màn hình đăng nhập

Ứng dụng sử dụng cơ sở dữ liệu Firebase Authentication để thiết kế giao diện login. Sử dụng Firebase SDK Authentication xác thực dựa trên email và mật khẩu.



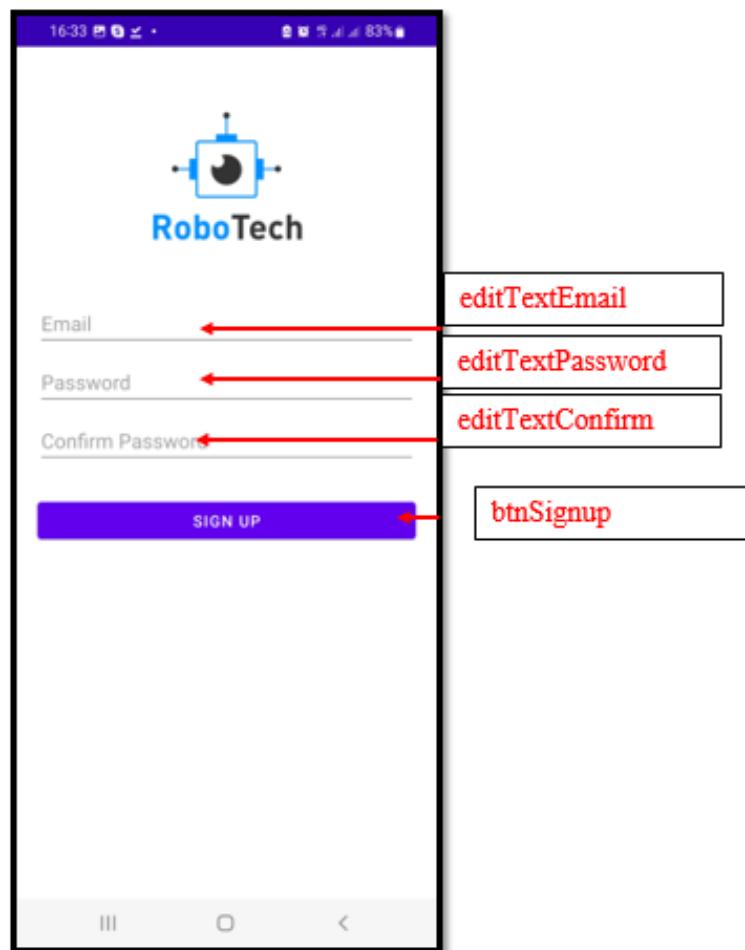
Hình 4. 18. Màn hình đăng nhập.

* Mô tả hoạt động:

STT	ID	Sự kiện	Mô tả hoạt động
1	[editTextEmail]	Nhập	Nhập email để đăng nhập
2	[editTextPassword]	Nhập	Nhập mật khẩu để đăng nhập
3	[checkBox]	Nhấn	Nhấn để tiến hành nhớ mật khẩu đã nhập
4	[btnLogin]	Nhấn	Kiểm tra tài khoản và mật khẩu, nếu đúng sẽ chuyển sang trang chủ
5	[textviewSignup]	Nhấn	Nhấn để tiến hành tạo tài khoản mới

Bảng 4. 1. Mô tả hoạt động chức năng đăng nhập

c. Màn hình tạo tài khoản



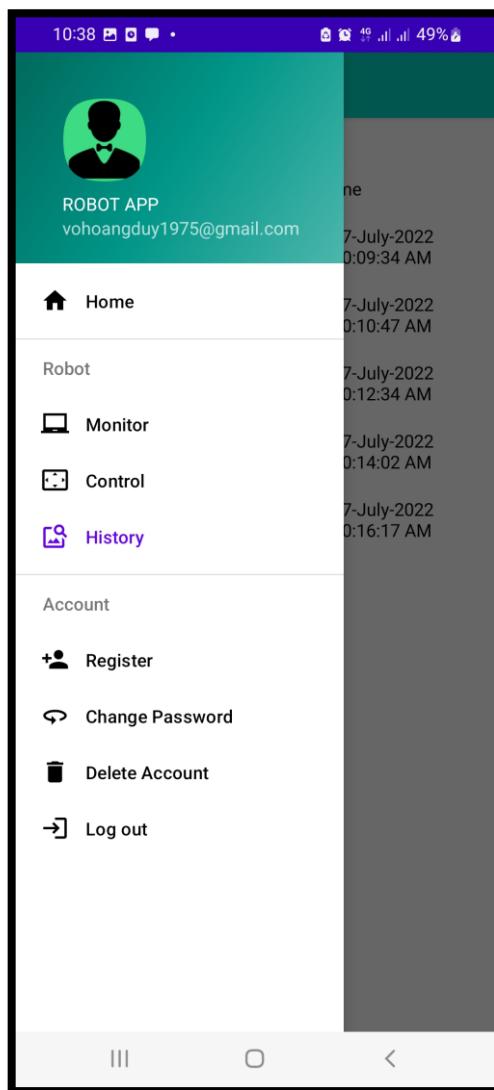
Hình 4. 19. Màn hình tạo tài khoản

* Mô tả hoạt động:

STT	ID	Sự kiện	Mô tả hoạt động
1	[editTextEmail]	Nhập	Nhập email để đăng nhập
2	[editTextPassword]	Nhập	Nhập mật khẩu để đăng nhập
3	[editTextConfirm]	Nhập	Nhập lại mật khẩu để tiến hành xác thực
4	[btnSignup]	Nhấn	Tiến hành tạo tài khoản mới

Bảng 4. 2. Mô tả hoạt động chức năng tạo tài khoản.

d. Các màn hình khác trong ứng dụng



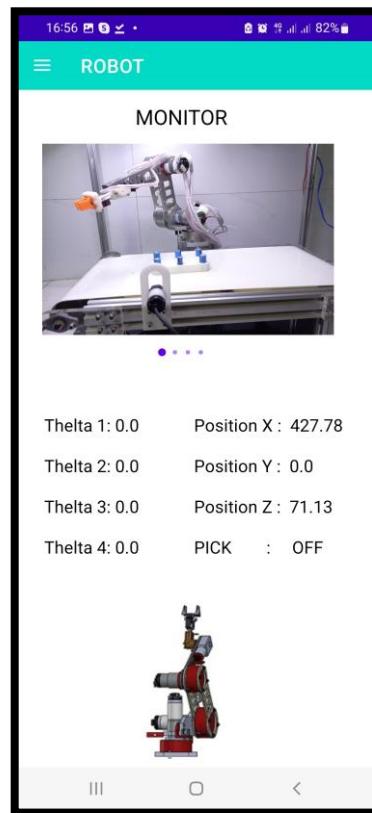
Hình 4. 20. Hình ảnh màn hình điều hướng

Màn hình điều hướng gồm có: Home, Monitor, Control, History, Register, Change Password, Delete Account và Logout.

- **Home:** màn hình trang chủ, hiển thị tên giáo viên hướng dẫn thầy Nguyễn Văn Thái và sinh viên thực hiện.
- **Monitor:** Hiển thị các thông số của Robot đang hoạt động được cập nhật trên Fire Realtime Database.
- **Control:** Được thiết kế để điều khiển robot gồm có điều khiển các góc của robot hoặc giá trị vị trí của robot.
- **History:** Hiển thị có sự kiện cảnh báo của Robot trong quá trình hoạt động.
- **Register, Change Password và Delete Account** các chức năng lần lượt là đăng ký tài khoản mới, thay đổi mật khẩu và xóa tài khoản.
- **Logout:** Chức năng đăng xuất khỏi ứng dụng.



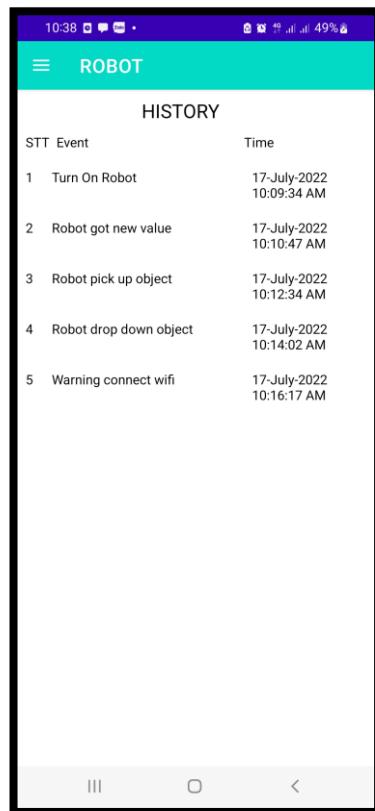
Hình 4. 21. Hình ảnh màn hình Home



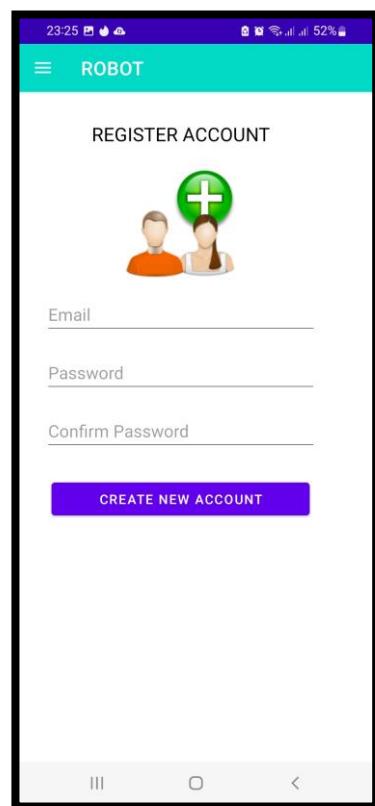
Hình 4. 22. Hình ảnh màn hình Monitor



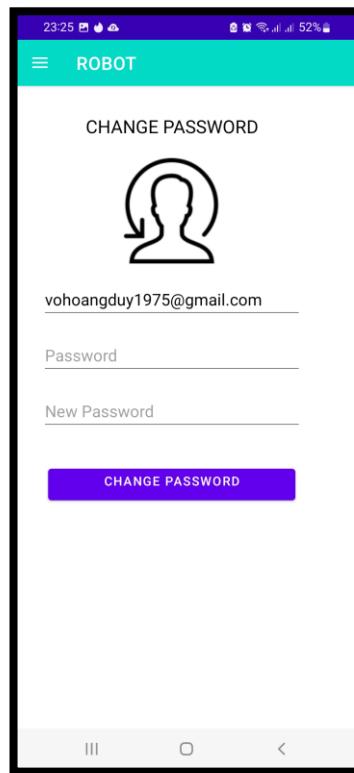
Hình 4. 23. Màn hình control



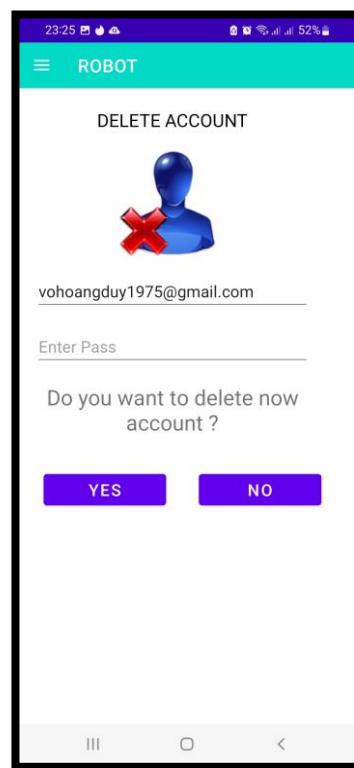
Hình 4. 24. Màn hình history



Hình 4. 25. Màn hình Register.



Hình 4. 26. Màn hình Change Password.



Hình 4. 27. Hình ảnh màn hình Delete Account.

4.2. Thiết kế giao diện điều khiển và giám sát Robot trên phần mềm Qt Designer

Robot sẽ được điều khiển và giám sát ngay trên màn hình máy tính. Giao diện được thiết kế trên ngôn ngữ Python. Giao diện điều khiển robot gồm có 6 trang chính:

- Home page.
- Setting page.
- Monitor page.
- Error page.
- Image process page.
- Information page.

4.2.1 Home page

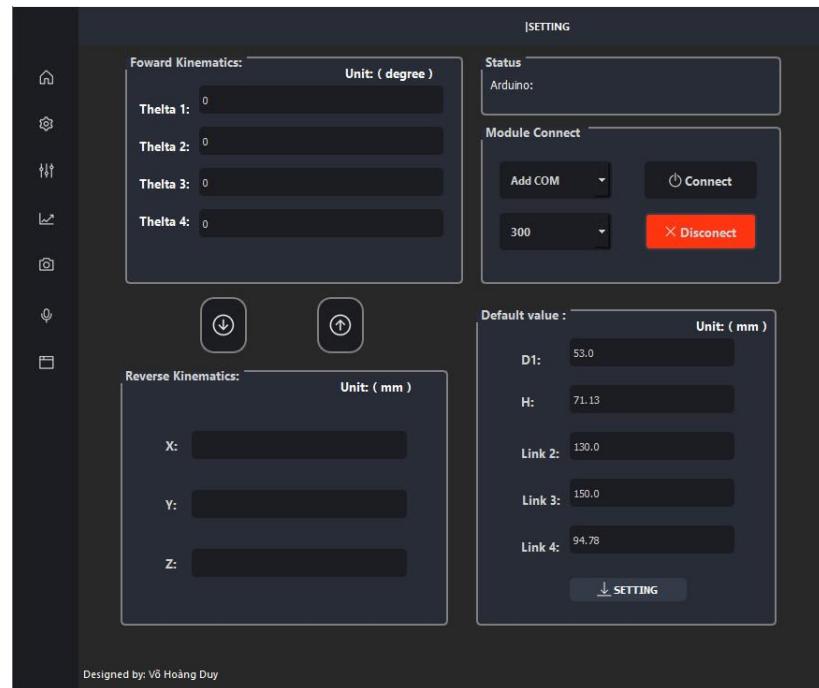
Home page được thiết kế bao gồm: tên đề tài, hình ảnh robot thực tế, tên giáo viên hướng dẫn và tên sinh viên thực hiện.



Hình 4. 28. Hình ảnh Home page.

4.2.2 Setting page

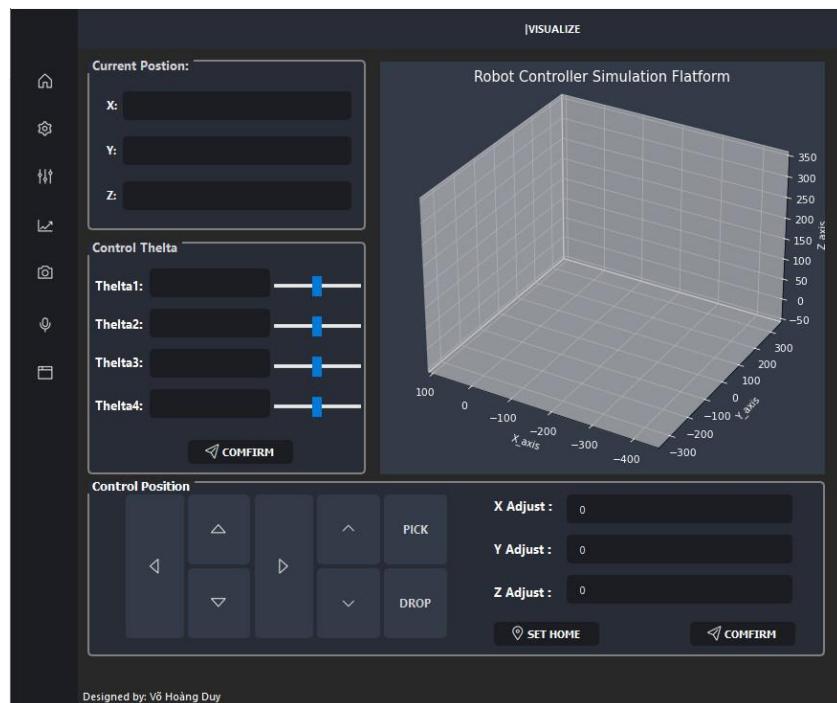
Setting page được thiết kế bao gồm: các nút nhấn và trạng thái kết nối với vi điều khiển, các tính toán khảo sát vị trí của Robot trước khi điều khiển và giá trị độ dài của các link được tính bằng đơn vị mm.



Hình 4. 29. Hình ảnh setting page.

4.2.3 Monitor page

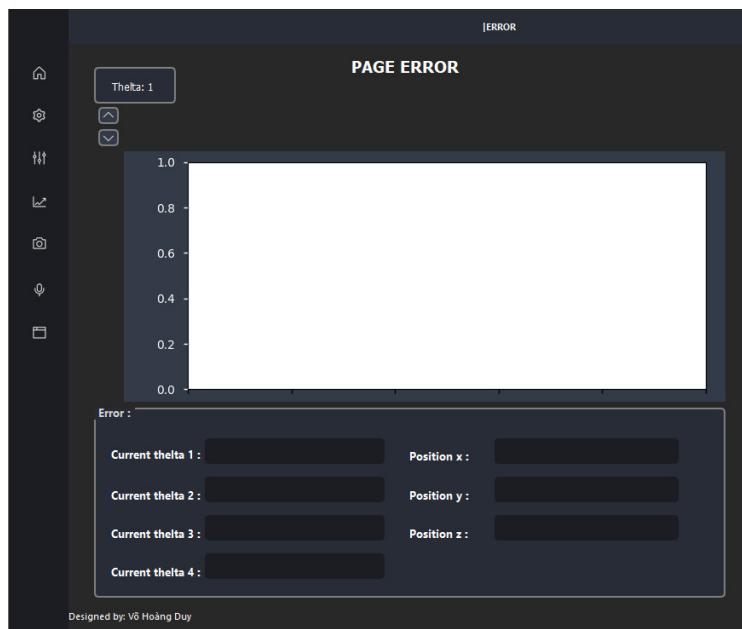
Monitor page hiển thị các giá trị vị trí hiện tại của robot, ta có thể điều khiển robot bằng cách truyền cho robot một vị trí cụ thể bằng cách nhập tọa độ X, Y, Z hoặc nhập giá trị 4 góc theta của robot. Ngoài ra, giao diện còn chứa mô tả 3D vùng làm việc của robot giúp người điều khiển hình dung khái quát về hình dáng robot.



Hình 4. 30. Hình ảnh Monitor Page.

4.2.3 Error page

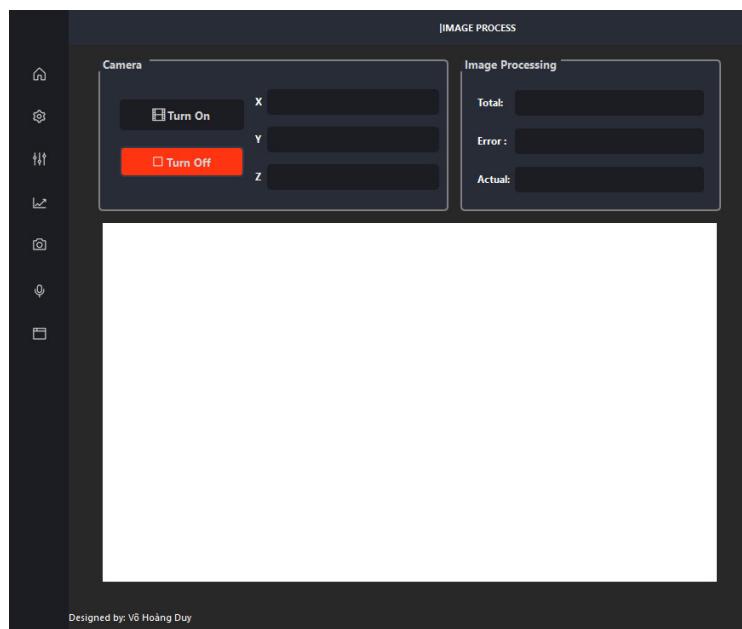
Giao diện hiển thị giá trị góc của các khớp và tọa độ X, Y, Z hiện tại của robot, vẽ biển đồ 2D hình ảnh thay đổi góc theo thời gian của Robot.



Hình 4. 31. Hình ảnh Error Page.

4.2.4 Image process page

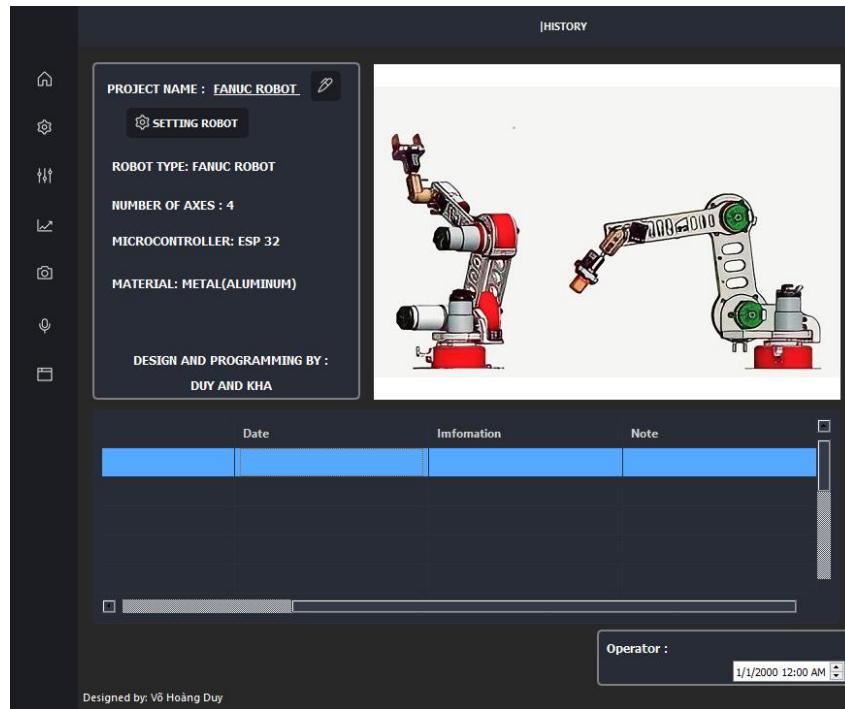
Giao diện dùng để tiến hành xử lý ảnh nhận diện linh kiện lỗi cụ thể là tụ điện phân cực giao diện sẽ hiển thị số linh kiện lỗi, tổng số linh kiện phát hiện được và tổng số linh kiện lỗi hiện tại. Hiển thị hình ảnh sau khi xử lý ảnh được.



Hình 4. 32. Hình ảnh Image process page.

4.2.5 *Information page*

Giao diện hiển thị thông tin của robot hình ảnh thiết kế 3D của robot vẽ trên ứng dụng Solidwork và hiển thị các cảnh báo của Robot khi robot gặp các lỗi.



Hình 4. 33. Hình ảnh Information page.

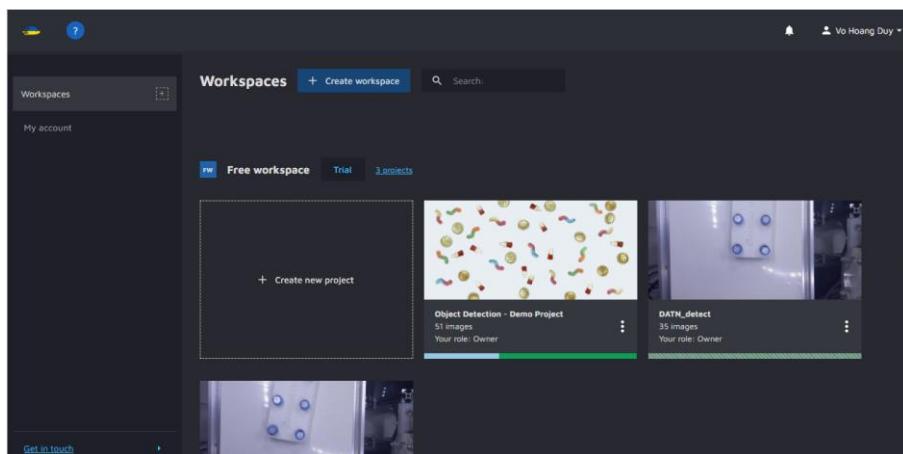
CHƯƠNG 5: XỬ LÝ ẢNH PHÁT HIỆN LỖI LINH KIỆN ĐIỆN TỬ SỬ DỤNG YOLOV5

5.1. Hướng dẫn training dữ liệu bằng YOLOv5

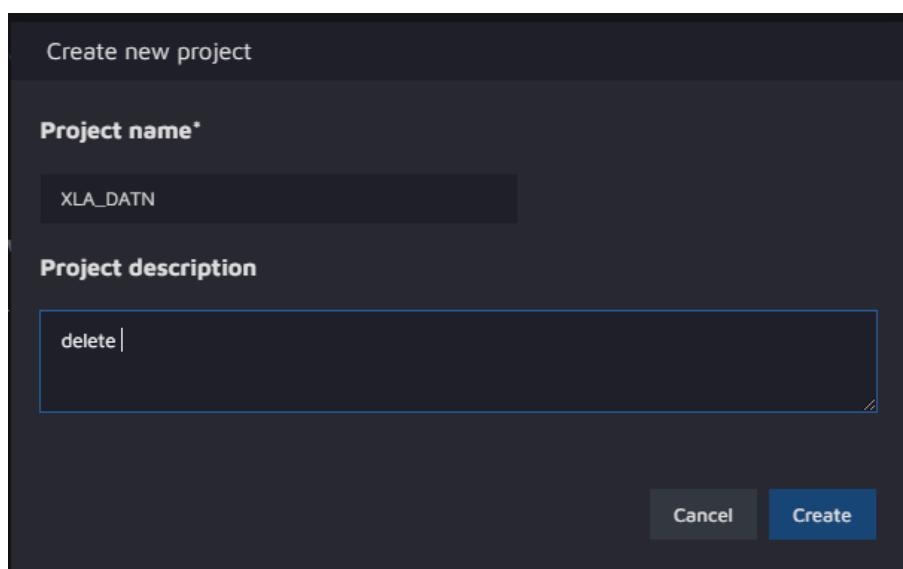
Ở đồ án của nhóm sẽ tiến hành train dữ liệu là các linh kiện lỗi để phát hiện và loại bỏ trước quá trình lắp ráp cụ thể là sẽ nhận diện những tụ điện lỗi, sai cực và có không đúng sẽ được nhận diện và loại bỏ.

5.1.1. Chuẩn bị dữ liệu.

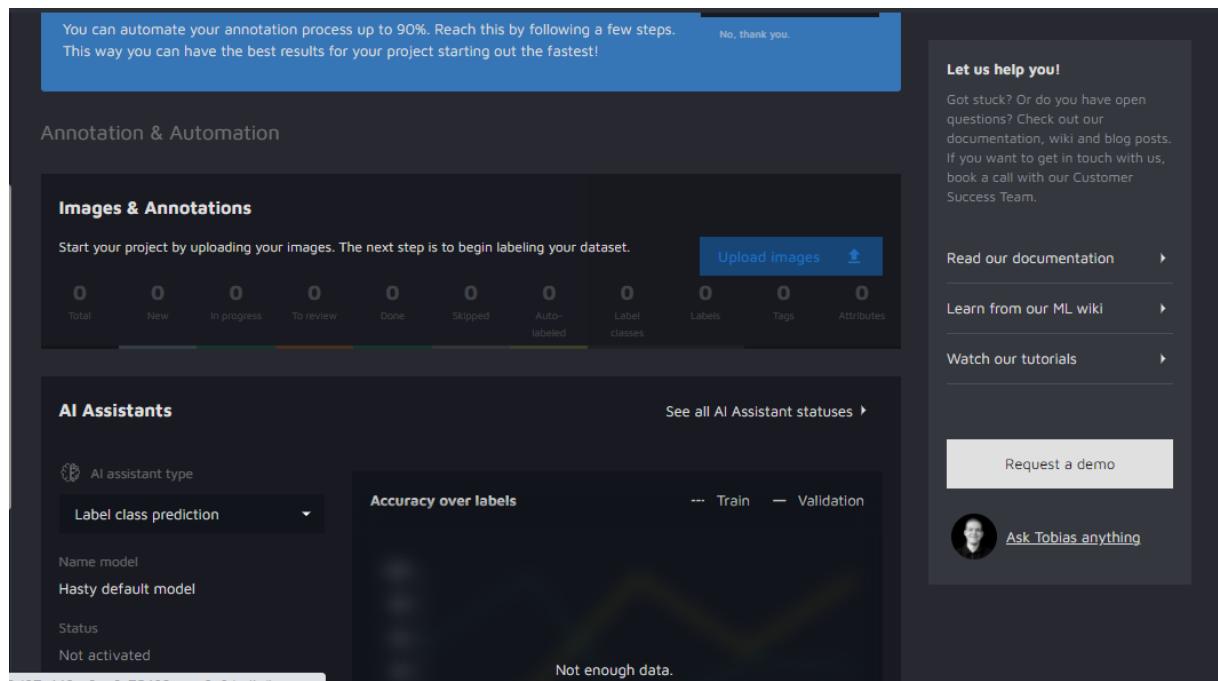
Đầu tiên chúng ta phải chuẩn bị một bộ dữ liệu hình ảnh vật thể cần huấn luyện. Sau đó dùng công cụ dán nhãn Hasty để gán nhãn cho các vật thể trong bộ dữ liệu. Truy cập vào Hasty.ai sau đó tạo tài khoản để thực hiện việc gán nhãn cho dữ liệu. Nhấn Create new project để tạo project mới.



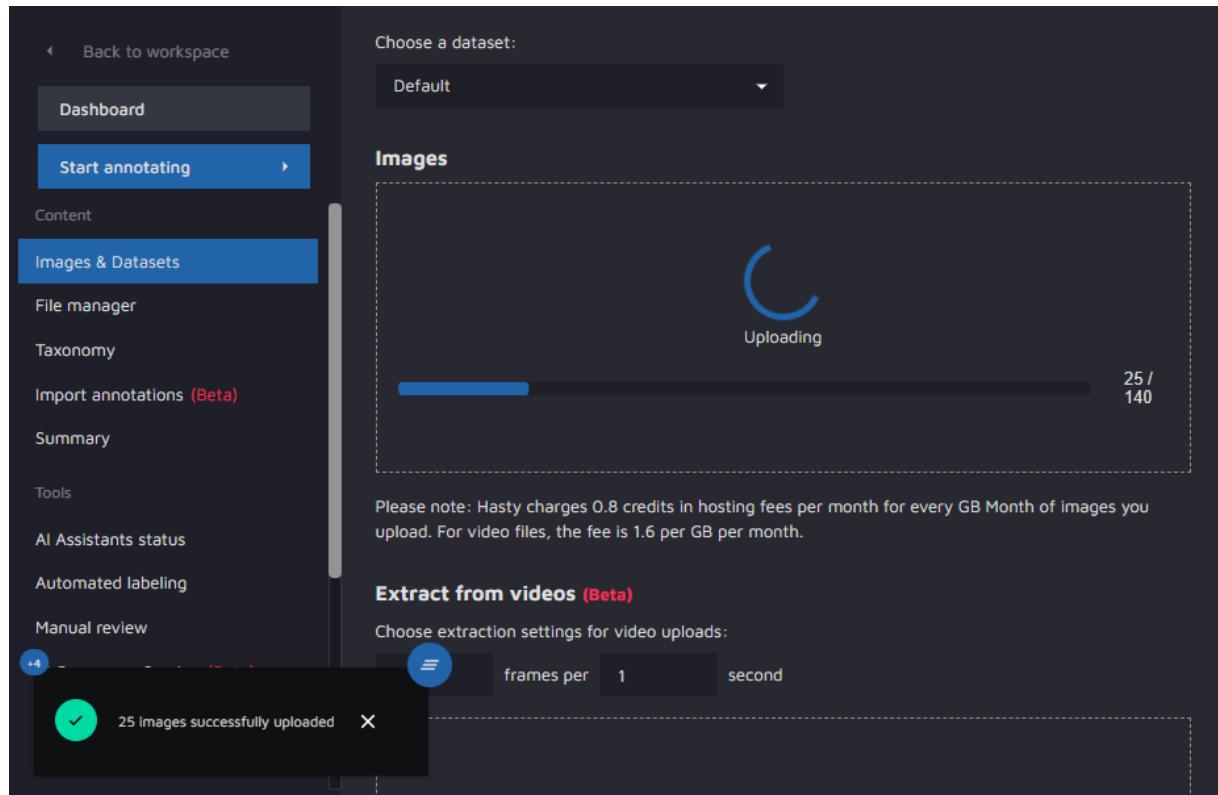
Hình 5. 1. Tạo project mới



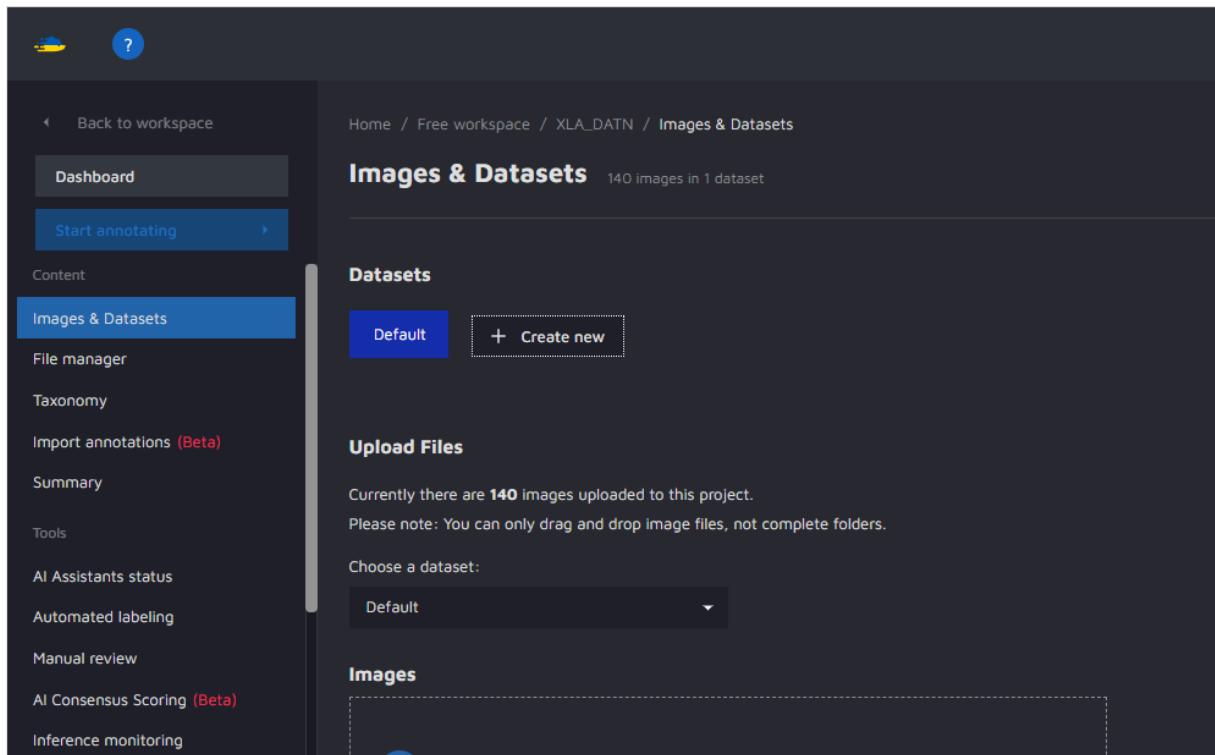
Hình 5. 2. Đặt tên cho project mới.



Hình 5. 3. Nhấn Upload Image để thêm ảnh muốn label.

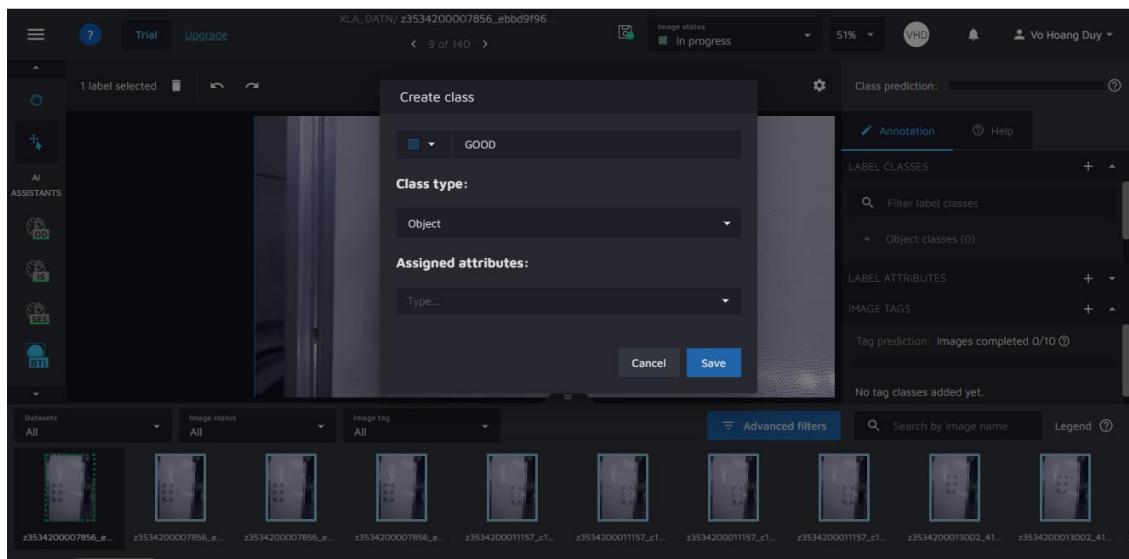


Hình 5. 4. Tiến hành upload ảnh lên web.



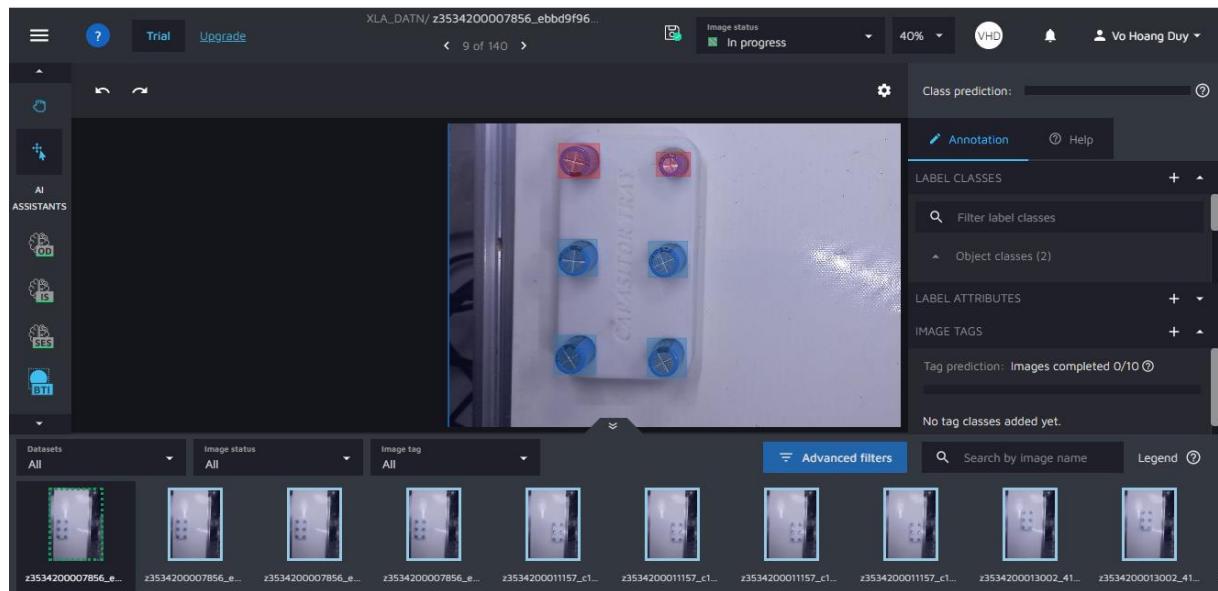
Hình 5. 5. Nhấn Start annotating để tiến hành label cho project.

Tiến hành tạo class cho project. Ở dự án này chúng tôi tạo ra 2 class là GOOD và NG. GOOD là những tụ điện quay đúng cực và đúng giá trị. NG là những tụ điện quay không đúng cực và không đúng giá trị của tụ điện.



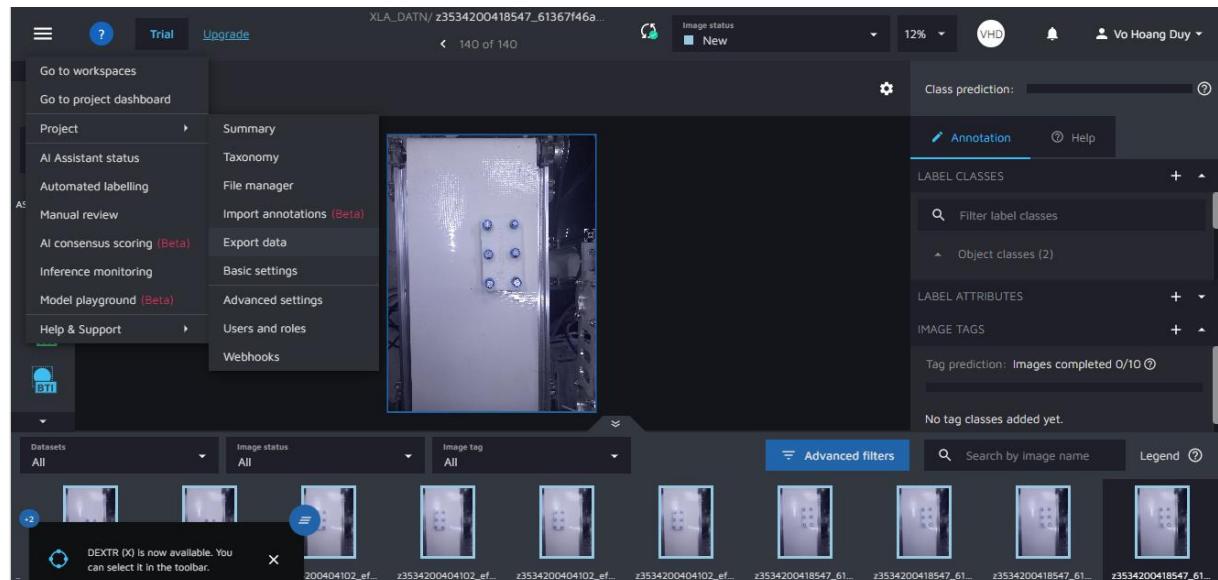
Hình 5. 6. Chọn và gắn label cho các object muốn phân loại

Tiến hành chọn và gắn label cho các object muốn phân loại. Dùng phím R trên bàn phím để tiến hành chọn object muốn label.



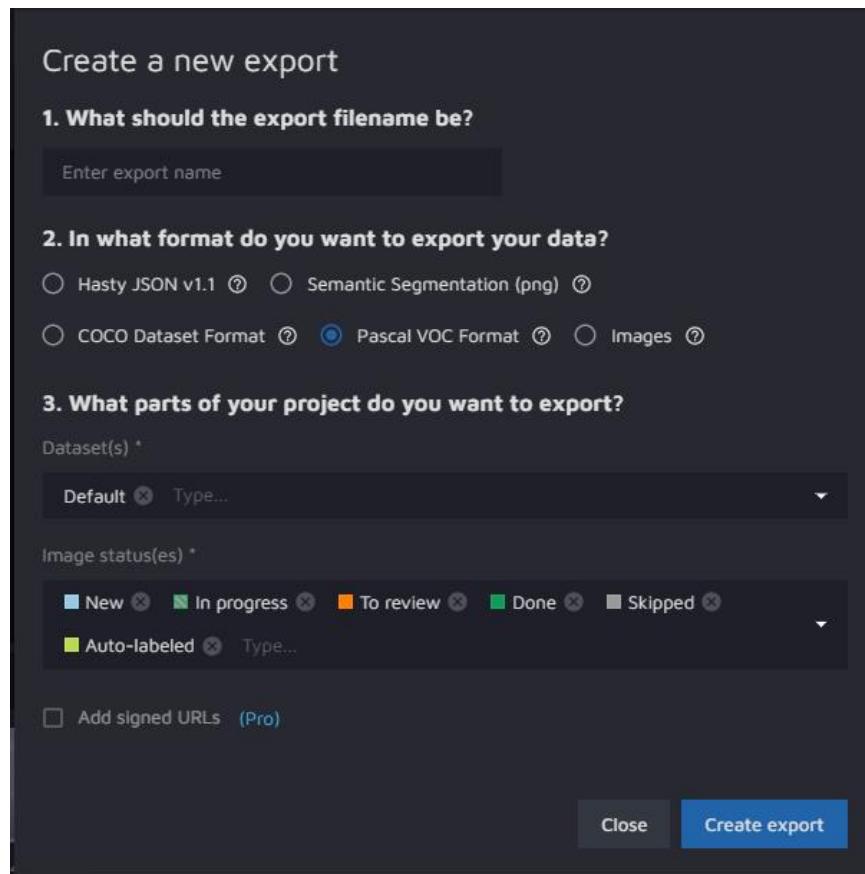
Hình 5. 7. Đặt tên cho dữ liệu đã được label

Sau khi label xong tiến hành chọn Project -> chọn Export data.



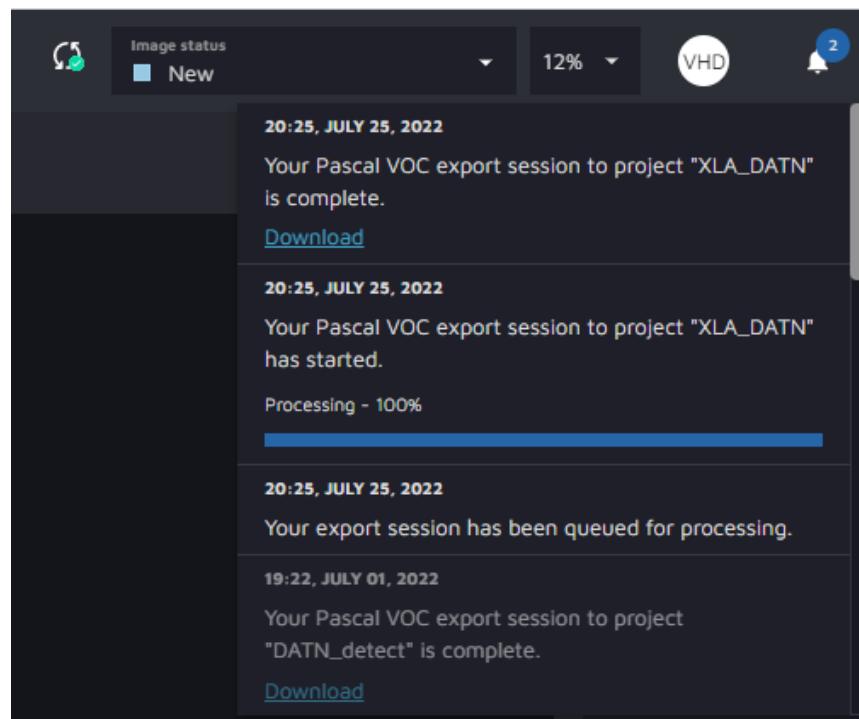
Hình 5. 8. Export data

Tiến hành đặt tên cho dữ liệu đã được label, chọn kiểu dữ liệu sau khi label. Ở project này nhóm mình xuất dữ liệu dạng Pascal VOC Format. Nhấn Create export để tạo data.



Hình 5. 9. Đặt tên và tạo export data

Nhấn vào biểu tượng thông báo để tiến hành tải về máy.



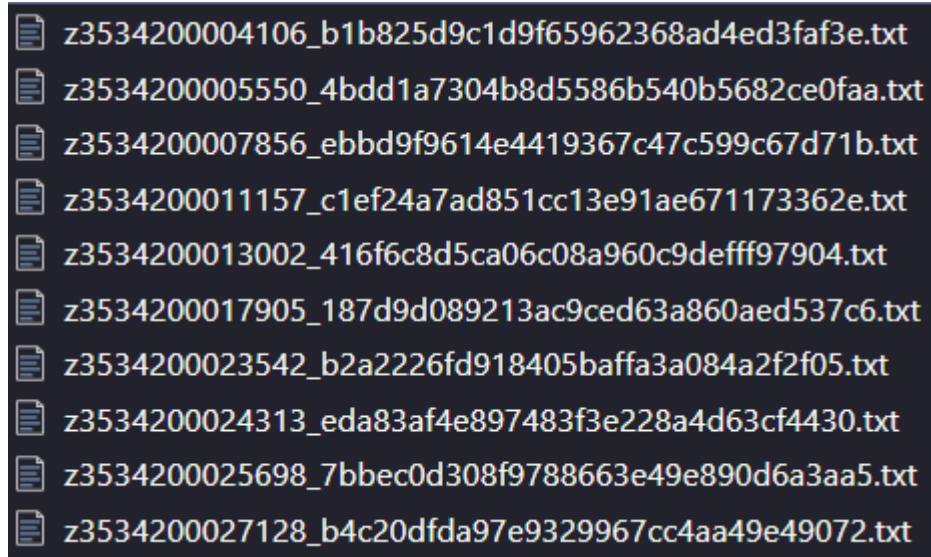
Hình 5. 10. Tiến hành tải data về máy

Dữ liệu sau khi label sẽ chứa các giá trị vị trí trong ảnh kích thước và các giá trị cần thiết để training dữ liệu.

```
<annotation>
  <folder>Default</folder>
  <filename>z3534200004106_b1b825d9c1d9f65962368ad4ed3faf3e.jpg</filename>
  <path>Default/z3534200004106_b1b825d9c1d9f65962368ad4ed3faf3e.jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>1920</width>
    <height>2560</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>NG</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>695</xmin>
      <ymin>685</ymin>
      <xmax>791</xmax>
      <ymax>793</ymax>
    </bndbox>
  </object>
</annotation>
```

Hình 5. 11. Dữ liệu của ảnh sau khi label

Định dạng dữ liệu sau khi xuất ra là Pascal VOC, để có thể đưa dữ liệu vào mô hình YOLOv5 chúng ta cần chuyển đổi định dạng dữ liệu sang định dạng của YOLO. Nội dung những file gán nhãn định dạng .xml sau khi chuyển đổi sẽ được lưu dưới định dạng file .txt.



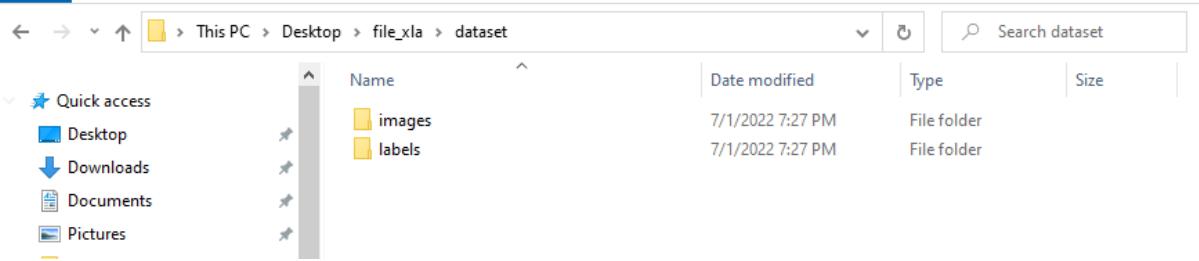
z3534200004106_b1b825d9c1d9f65962368ad4ed3faf3e.txt
z3534200005550_4bdd1a7304b8d5586b540b5682ce0faa.txt
z3534200007856_ebbd9f9614e4419367c47c599c67d71b.txt
z3534200011157_c1ef24a7ad851cc13e91ae671173362e.txt
z3534200013002_416f6c8d5ca06c08a960c9deff97904.txt
z3534200017905_187d9d089213ac9ced63a860aed537c6.txt
z3534200023542_b2a2226fd918405baffa3a084a2f2f05.txt
z3534200024313_eda83af4e897483f3e228a4d63cf4430.txt
z3534200025698_7bbec0d308f9788663e49e890d6a3aa5.txt
z3534200027128_b4c20dfda97e9329967cc4aa49e49072.txt

Hình 5. 12. File .txt dữ liệu sau khi chuyển đổi

Tạo folder Dataset chứa 2 folder:

- Image chứa các ảnh chọn để label.

- Label chứa các file dữ liệu sau khi label.



Hình 5. 13. Tạo folder chứa data

5.1.2. Huấn luyện mô hình

Mô hình được huấn luyện trên môi trường Google Colab. Truy cập vào đường dẫn: https://colab.research.google.com/github/ultralytics/yolov5/blob/master/tutorial.ipynb#scrollTo=1NcFxRcFdJ_O để mở notebook của YOLOv5 trên Google Colab [14].

```

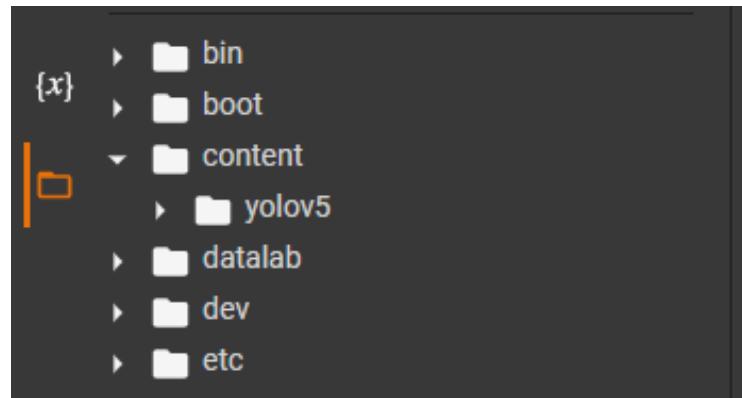
# Train YOLOv5 on COCO128 for 3 epochs
!python train.py --img 640 --batch 16 --epochs 3 --data coco128.yaml --weights yolov5s.pt --cache
[...]
train: weights=yolov5s.pt, cfgs, data=coco128.yaml, hyp-data/hyp.scratch-low.yaml, epochs=3, batch_size=16, imgsz=640, rect=False, resume=False, nosave=False, noval=False, noautoanchor=False, mplots=False, evolve=None, bucket=None, up_to_date with https://github.com/ultralytics/yolov5
YOLOv5 v6.1-157-g609f707 Python-3.7.13 torch-1.11.0+cu113 CUDA:0 (Tesla V100-SXM2-16GB, 16160MiB)
hyperparameters: img=640, lr=0.01, momentum=0.997, weight_decay=0.0005, warmup_epochs=3.0, warmup_momentum=0.8, warmup_bias_lr=0.1, box=0.05, cls=0.5, cls_pw=1.0, obj=1.0, obj_pw=1.0, iou_t=0.2, anchor_t=4.0, fl_gamma=0.0, hsv_h=0.0, hsv_s=0.0, hsv_v=0.0, weights & Biases: run 'pip install wandb' to automatically track and visualize YOLOv5 & runs (RECOMMENDED)
Weights & Biases: run 'pip install wandb' to automatically track and visualize YOLOv5 & runs (RECOMMENDED)
TensorBoard: Start with 'tensorboard --logdir runs/train', then at http://localhost:6006

from n import params module arguments
0 [-1 1 3520 models.common.Conv [32, 32, 6, 2, 2]
1 [-1 1 1024 models.common.Conv [64, 64, 3, 2]
2 [-1 1 18816 models.common.C3 [64, 64, 1]
3 [-1 1 73984 models.common.Conv [64, 128, 3, 2]
4 [-1 2 115712 models.common.C3 [128, 128, 2]
5 [-1 1 29424 models.common.Conv [128, 256, 3, 2]
6 [-1 1 62848 models.common.C3 [128, 256, 3, 2]
7 [-1 1 118672 models.common.Conv [256, 512, 3, 2]
8 [-1 1 118772 models.common.C3 [512, 512, 1]
9 [-1 1 656896 models.common.SPPf [512, 512, 5]
10 [-1 1 131584 models.common.Conv [512, 256, 1, 1]
11 [-1 1 131584 models.common.modules.upsampling.Upsample [512, 256, 1, 2, 'nearest']
12 [-1, 6] 1 0 models.common.Concat [1]
13 [-1 1 361984 models.common.C3 [512, 256, 1, False]
14 [-1 1 31024 models.common.Conv [256, 128, 1, 1]
15 [-1 1 0 torch.nn.modules.upsampling.Upsample [None, 2, 'nearest']
16 [-1, 1 0 models.common.Concat [1]
17 [-1 1 98080 models.common.C3 [128, 128, 1, False]
18 [-1 1 147712 models.common.Conv [128, 128, 3, 2]
19 [-1, 14] 1 0 models.common.Concat [1]
20 [-1 1 256448 models.common.C3 [256, 256, 1, False]
21 [-1 1 598336 models.common.Conv [256, 256, 3, 2]
22 [-1, 18] 1 0 models.common.Concat [1]
23 [-1 1 1183720 models.common.C3 [512, 512, 1, False]
24 [17, 20, 23] 1 220245 models.yolo.Detect [80, [[10, 13, 16, 38, 33, 23], [38, 61, 62, 45, 59, 119], [116, 90, 156, 198, 373, 326]], [128, 256, 512]]]
Model summary: 270 layers, 7235389 parameters, 7235389 gradients
<> Transferred 349/349 items from yolov5.pt
AMP: checks passed ✓
Scaled weight_decay = 0.0005
optimizer: SGD with parameter groups 57 weight (no decay), 60 weight, 60 bias
albumentations: version 1.0.3 required by YOLOv5, but version 0.1.12 is currently installed
train: Scanning '/content/datasets/coco128/labels/train2017.cache' images and labels... 128 found, 0 missing, 2 empty, 0 corrupt: 100% 128/128 [00:00<?, ?it/s]

```

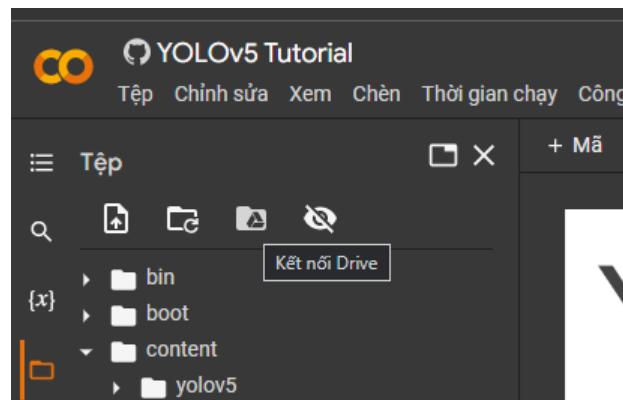
Hình 5. 14. Giao diện Google Colab

Đầu tiên chạy ô dưới đây để tải YOLOv5 về môi trường Colab đồng thời cài đặt những thư viện cần thiết.



Hình 5. 15. Repo YOLOv5 được lưu trong thư mục content

Tiếp theo mount drive ta đã tải dữ liệu lên bằng cách bấm chọn Kết nối Drive



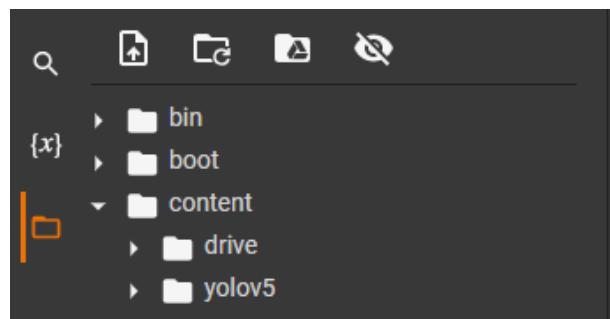
Hình 5. 16. Bấm chọn biểu tượng drive để kết nối

Sau khi chọn một ô chứa mã thực thi kết nối drive sẽ được tạo, ta chạy ô đó để kết nối drive

```
▶ from google.colab import drive
  drive.mount('/content/drive')
```

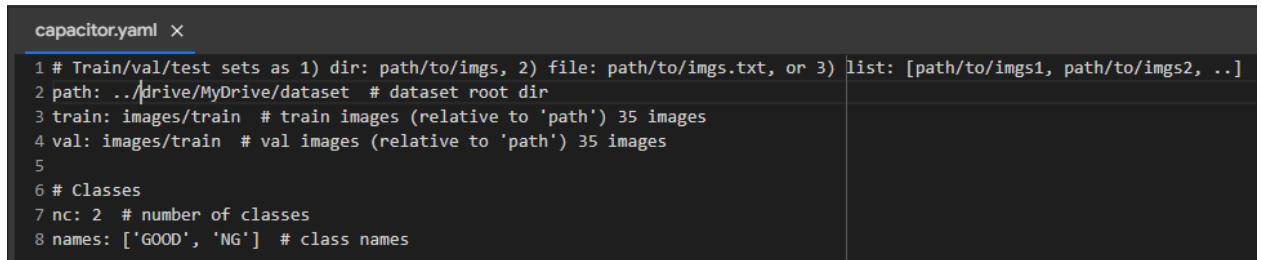
Hình 5. 17. Ô chứa mã thực thi kết nối drive

Thư mục drive sẽ được tạo trong thư mục content sau khi kết nối drive thành công.



Hình 5. 18. Thư mục drive được tạo trong thư mục content

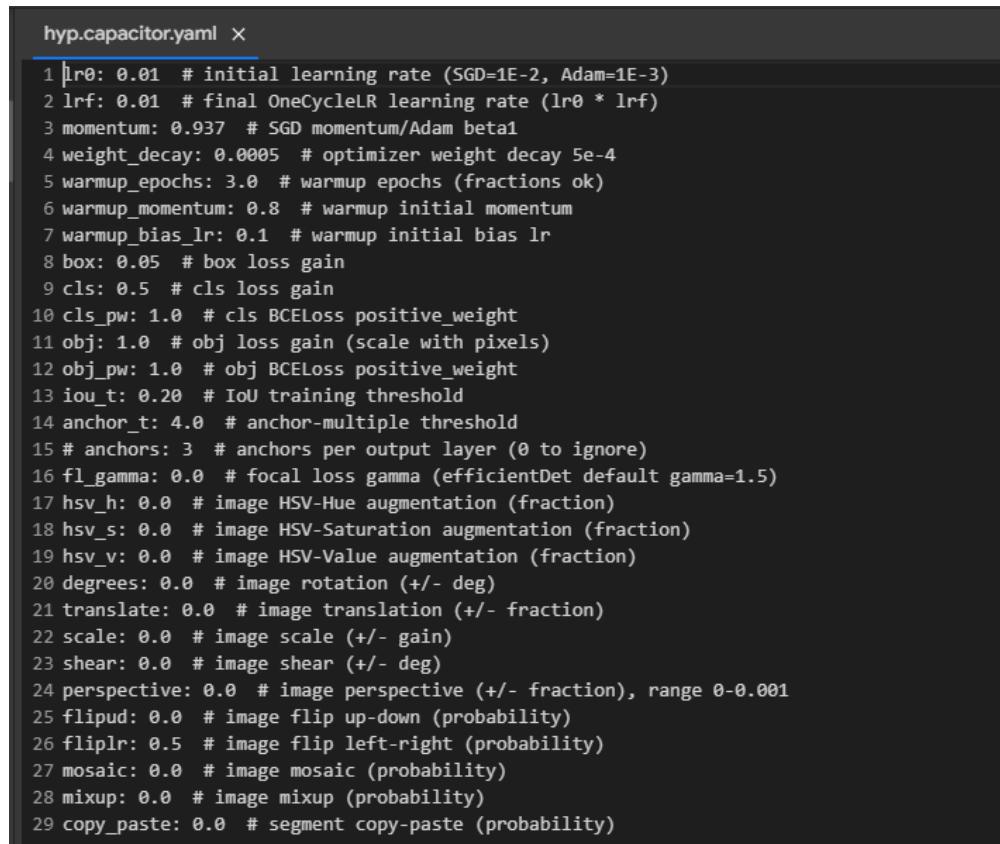
Vào thư mục yolov5/data, ta tạo một file cấu hình tên **capacitor.yaml** chứa các thông tin sau: đường dẫn đến thư mục chứa dữ liệu, số lượng class, tên class.



```
capacitor.yaml ×
1 # Train/val/test sets as 1) dir: path/to/imgs, 2) file: path/to/imgs.txt, or 3) list: [path/to/imgs1, path/to/imgs2, ...]
2 path: ../drive/MyDrive/dataset # dataset root dir
3 train: images/train # train images (relative to 'path') 35 images
4 val: images/train # val images (relative to 'path') 35 images
5
6 # Classes
7 nc: 2 # number of classes
8 names: ['GOOD', 'NG'] # class names
```

Hình 5. 19. Thông tin trong file cấu hình capacitor.yaml

Trong thư mục yolov5/data/hyps tạo một file tên hyp.capacitor.yaml chứa các thông số cấu hình để huấn luyện mô hình.



```
hyp.capacitor.yaml ×
1 lr0: 0.01 # initial learning rate (SGD=1E-2, Adam=1E-3)
2 lrf: 0.01 # final OneCycleLR learning rate (lr0 * lrf)
3 momentum: 0.937 # SGD momentum/Adam beta1
4 weight_decay: 0.0005 # optimizer weight decay 5e-4
5 warmup_epochs: 3.0 # warmup epochs (fractions ok)
6 warmup_momentum: 0.8 # warmup initial momentum
7 warmup_bias_lr: 0.1 # warmup initial bias lr
8 box: 0.05 # box loss gain
9 cls: 0.5 # cls loss gain
10 cls_pw: 1.0 # cls BCELoss positive_weight
11 obj: 1.0 # obj loss gain (scale with pixels)
12 obj_pw: 1.0 # obj BCELoss positive_weight
13 iou_t: 0.20 # IoU training threshold
14 anchor_t: 4.0 # anchor-multiple threshold
15 # anchors: 3 # anchors per output layer (0 to ignore)
16 fl_gamma: 0.0 # focal loss gamma (efficientDet default gamma=1.5)
17 hsv_h: 0.0 # image HSV-Hue augmentation (fraction)
18 hsv_s: 0.0 # image HSV-Saturation augmentation (fraction)
19 hsv_v: 0.0 # image HSV-Value augmentation (fraction)
20 degrees: 0.0 # image rotation (+/- deg)
21 translate: 0.0 # image translation (+/- fraction)
22 scale: 0.0 # image scale (+/- gain)
23 shear: 0.0 # image shear (+/- deg)
24 perspective: 0.0 # image perspective (+/- fraction), range 0-0.001
25 flipud: 0.0 # image flip up-down (probability)
26 fliplr: 0.5 # image flip left-right (probability)
27 mosaic: 0.0 # image mosaic (probability)
28 mixup: 0.0 # image mixup (probability)
29 copy_paste: 0.0 # segment copy-paste (probability)
```

Hình 5. 20. Thông số cấu hình để huấn luyện mô hình

Sau khi đã chuẩn bị các file cấu hình ta tiến hành huấn luyện mô hình, ta sẽ chỉnh sửa đoạn lệnh như sau rồi chạy ô để bắt đầu quá trình huấn luyện.

```
[ ] # Train YOLOv5s on COCO128 for 3 epochs
!python train.py --img 1280 --batch 16 --epochs 50 --data capacitor.yaml --hyp hyp.capacitor.yaml --weights yolov5n.pt
```

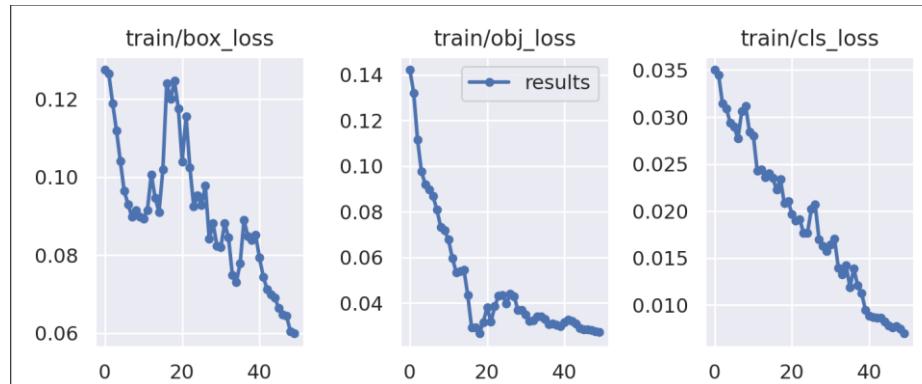
Hình 5. 21. Ô chứa lệnh thực thi quá trình huấn luyện.

Sao chép vào Drive								
	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95:	100% 2/2 [00:02<00:00, 1.45s/it]
...	all	35	210	0.0445	0.601	0.0456	0.0134	
Epoch 20/49	gpu_mem	box	obj	cls	labels	img_size		
	7.69G	0.1039	0.03803	0.01966	18	1280: 100% 3/3 [00:01<00:00, 1.62it/s]		
	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95:	100% 2/2 [00:04<00:00, 2.11s/it]
	all	35	210	0.0637	0.228	0.0675	0.015	
Epoch 21/49	gpu_mem	box	obj	cls	labels	img_size		
	7.69G	0.1156	0.03175	0.01898	18	1280: 100% 3/3 [00:01<00:00, 2.28it/s]		
	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95:	100% 2/2 [00:02<00:00, 1.42s/it]
	all	35	210	0.135	0.454	0.121	0.0323	
Epoch 22/49	gpu_mem	box	obj	cls	labels	img_size		
	7.69G	0.1024	0.0387	0.01909	18	1280: 100% 3/3 [00:01<00:00, 2.18it/s]		
	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95:	100% 2/2 [00:02<00:00, 1.41s/it]
	all	35	210	0.177	0.6	0.148	0.0298	
Epoch 23/49	gpu_mem	box	obj	cls	labels	img_size		
	7.69G	0.09241	0.04325	0.01767	18	1280: 100% 3/3 [00:01<00:00, 2.09it/s]		
	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95:	100% 2/2 [00:02<00:00, 1.34s/it]
	all	35	210	0.688	0.381	0.265	0.0638	
Epoch 24/49	gpu_mem	box	obj	cls	labels	img_size		
	7.69G	0.09523	0.04346	0.01771	18	1280: 100% 3/3 [00:01<00:00, 2.19it/s]		
	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95:	100% 2/2 [00:02<00:00, 1.40s/it]
	all	35	210	0.107	0.523	0.0893	0.016	
Epoch 25/49	gpu_mem	box	obj	cls	labels	img_size		
	7.69G	0.09284	0.03978	0.0202	18	1280: 100% 3/3 [00:01<00:00, 2.18it/s]		
	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95:	100% 2/2 [00:02<00:00, 1.24s/it]
	all	35	210	0.109	0.151	0.0644	0.00802	
Epoch 26/49	gpu_mem	box	obj	cls	labels	img_size		
	7.69G	0.09777	0.04402	0.02068	18	1280: 100% 3/3 [00:01<00:00, 2.04it/s]		
	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95:	100% 2/2 [00:02<00:00, 1.44s/it]
	all	35	210	0.447	0.812	0.787	0.331	
Epoch 27/49	gpu_mem	box	obj	cls	labels	img_size		
	7.69G	0.08409	0.04288	0.01696	18	1280: 100% 3/3 [00:01<00:00, 2.18it/s]		
	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95:	0% 0/2 [00:00<?, ?it/s]

Hình 5. 22. Quá trình huấn luyện mô hình

5.2. Kết quả huấn luyện mô hình YOLOv5

Mô hình sẽ lưu hai file trọng số là best.pt và last.pt tương ứng với trọng số của epoch tốt nhất và epoch cuối cùng, ta sẽ sử dụng file best.pt.



Hình 5. 23. Đồ thị biểu diễn các giá trị hàm mất mát trong quá trình huấn luyện

Ta có thể thấy giá trị của hàm mất mát giảm trong suốt quá trình huấn luyện và gần tiệm cận về không thể hiện mô hình học tốt các đối tượng trong tập dữ liệu.

Tiến hành dữ dụng file dữ liệu best.pt sau train để tiến hành nhận diện linh kiện. Sau khi nhận diện thu được kết quả chính xác so với dữ liệu được train như hình dưới.



Hình 5. 24. Kết quả nhận diện

Sau khi sử dụng xử lý ảnh để nhận diện tụ điện bị lỗi đã thu được kết quả đúng với dữ liệu đã training với tỉ lệ chính xác từ 85% trở lên. Với kết quả đạt được ta có thể dùng model YOLOv5 để tiến hành training các ứng dụng thực tế của công đoạn IQC giúp phân loại và loại bỏ các sản phẩm không đạt ra khỏi băng chuyền sản xuất, nâng cao được hiệu quả làm việc và tiết kiệm được khá nhiều chi phí nhân công.

5.3. Phát hiện tọa độ

Để gửi tọa độ vật thể đến bộ điều khiển trung tâm trước tiên chúng ta cần xác định độ tâm của vật thể sử dụng hàm: cv.moments(cnt).

Trong xử lý ảnh, moment của ảnh được dùng để nêu bật đặc trưng hình dạng của một ảnh. Những moment này ghi lại những thuộc tính của hình ảnh, bao gồm diện tích của đối tượng (the area), trọng tâm (centroid), hướng (orientation) và những thuộc tính liên quan khác. Ở trên ta vẽ ra viền của đa giác bằng hàm cv2.drawContours, tiếp đến ta xác định tọa độ tâm (C_x , C_y) cho đối tượng theo công thức sau [15]:

$$\begin{aligned} C_x &= \frac{M_{10}}{M_{00}} \\ C_y &= \frac{M_{01}}{M_{00}} \end{aligned} \tag{5.1}$$

Với C_x là tọa độ x, C_y là tọa độ y.

Để giải thích cho công thức trên ta sẽ sử dụng hình ảnh bên dưới có chứa các giá trị pixel nhị phân. Ta chọn các giá trị nhị phân như hình dưới để đơn giản hóa.

	1	2	3	4
1	0	1	1	0
2	0	1	1	0
3	0	1	1	0
4	0	0	0	0

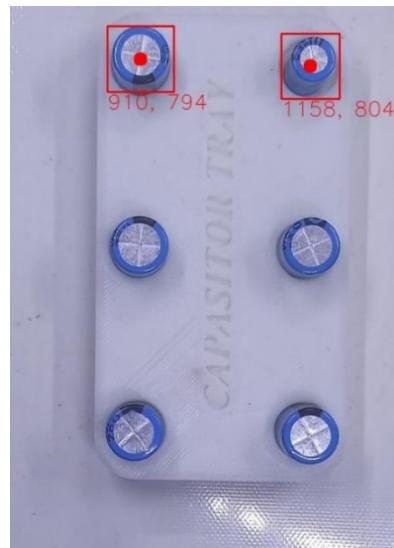
Hình 5. 25. Các bit nhị phân

Các giá trị $M_{00}, M_{01}, M_{10}, M_{00}$ được xác định bởi công thức:

$$M_{ij} = \sum x \sum y x^i y^j I(x, y) \quad (5.2)$$

M_{ij} là viết tắt của moments được tính cho thứ tự (i, j) . Vì vậy $I(x, y)$ là cường độ cho mỗi pixel của hình ảnh, x và y tham chiếu đến hàng và cột của hình ảnh.

Với việc xác định tọa độ tâm là đã định được vị trí của vật được mô tả như hình.

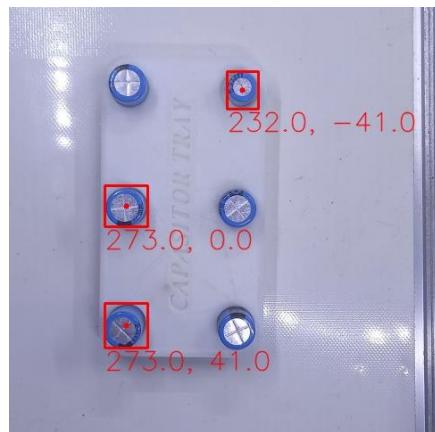


Hình 5. 26. Xác định vị trí vật

Các bước tiến hành chuyên đổi ảnh sang vị trí thực robot:

- Kẻ 1 đường thẳng hoặc đánh dấu 2 vị trí biết trước khoảng cách thực tế.
- Tiến hành get giá trị pixel của 2 vị trí tính số pixel giữa hai điểm rồi chuyển sang giá trị mm theo đơn vị của robot.

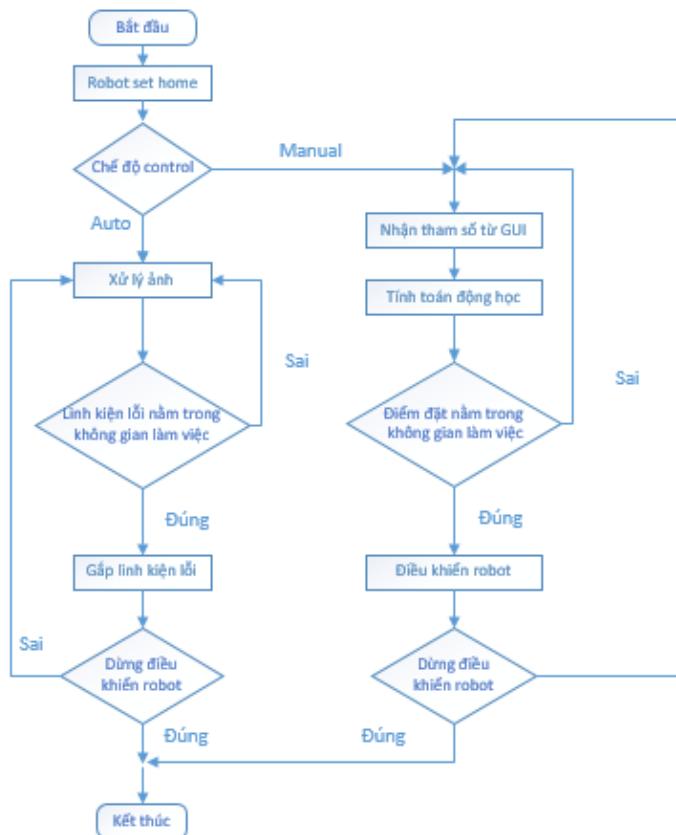
- Sau đó tính giá trị thực tế theo tỉ lệ pixel đã tính.



Hình 5. 27. Kết quả sau khi chuyển đổi ảnh sang vị trí thực

Kết quả hiển thị box chứa tụ điện bị lỗi và tọa độ tâm xy của nó từ đó sẽ gửi tọa độ cho robot gấp bỏ tụ điện bị lỗi ra khỏi bãng chuyên. Tọa độ thu được sau khi xử lý ảnh có sai số < 0.5 cm so với thực tế giúp cho robot có thể đưa tay kẹp đến vị trí tụ điện và gấp tụ điện khỏi tray chứa linh kiện. Tỉ lệ robot gấp thành công tụ điện đạt trên 90%.

5.3. Lưu đồ điều khiển robot 4 bậc tự do.



Hình 5. 28. Lưu đồ điều khiển robot

Khi mạch được cấp nguồn, MCU thực hiện khởi tạo ngoại vi phần cứng cần thiết. Sau khi robot sethome hoàn tất, MCU chờ nhận tín hiệu điều khiển từ giao diện GUI điều khiển robot. Tùy thuộc vào mã lệnh nhận được, MCU sẽ thực hiện các hành động sau:

- Nếu nhận được lệnh chạy theo chế độ auto thì khi có tín hiệu từ cảm biến phát hiện có khay chứa tụ điện tại vị trí camera, máy tính sẽ bắt đầu xử lý ảnh để tìm ra vị trí của tụ điện bị lỗi nếu có. Từ vị trí có được robot sẽ tính toán động học để tìm ra bộ nghiệm các góc theta của từng khớp. Nếu tìm ra được nghiệm máy tính sẽ truyền tín hiệu các góc xuống board Arduino Mega để để khiển robot xoay đèn đến gắp tụ điện bị hỏng.

- Nếu nhận được lệnh chạy theo chế độ manual thì robot sẽ đợi tín hiệu điều khiển từ GUI trên máy tính. Bằng giao diện GUI chúng ta có thể điều khiển robot bằng cách nhập góc theta cho từng góc hoặc nhập vị trí mà robot cần đến. Từ vị trí chúng ta nhập, máy tính sẽ tính toán động học và truyền tín hiệu các góc theta xuống board Arduino Mega để để khiển các góc xoay của robot.

CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1. Kết luận

6.1.1. *Những công việc đã thực hiện*

Sau thời gian nghiên cứu và tìm hiểu, nhóm đã thực hiện và đạt được những kết quả sau:

- Thiết kế và thi công thành công phần cơ khí cho robot 4 bậc tự do. Tất cả các chi tiết chính của robot đều được gia công CNC chính xác bằng nhôm nguyên khối.
- Thiết kế và thi công thành công mạch điều khiển động cơ cho robot và mạch truyền nhận dữ liệu cho robot. Bo mạch điều khiển đã được thử nghiệm điều khiển robot ổn định trong suốt thời gian robot hoạt động.
- Thiết kế thành công giao diện GUI trên màn hình destop máy tính PC để điều khiển robot hoạt động theo 2 chế độ cài đặt là Auto và Manual.
- Thiết kế thành công ứng dụng xử lý ảnh để phân loại tụ điện bị lỗi và thuật toán xác định tọa độ tụ điện để robot tiến hành gấp tụ điện bị lỗi loại bỏ ra bên ngoài.
- Thiết kế thành công app điện thoại Android để giám sát hoạt động của robot từ xa thông qua mạng Internet.

6.1.2. *Những mặt hạn chế*

Đề tài có những hạn chế sau:

- Phần cứng robot chưa được thiết kế tối ưu (hộp số động cơ dùng cho robot có độ rơ lớn làm ảnh hưởng đến độ chính xác của robot) do tài chính không cho phép.
- Robot hoạt động chưa được mượt mà, độ ổn định trong chuyển động của robot chưa cao do độ rơ của hộp số động cơ được lựa chọn để thi công robot, vấn đề này có thể được khắc phục dễ dàng bởi loại hộp số không độ rơ với chi phí rất cao.
- Mô-men được truyền từ trực hộp số đến khớp (joint) của robot thông qua khớp nối mềm nên khả năng chịu tải của robot kém, chỉ có thể gấp các vật nhỏ.
- Ứng dụng trên điện thoại chỉ có chức năng giám sát hoạt động của robot.
- Ứng dụng xử lý ảnh để nhận dạng tụ điện bị lỗi có độ chính xác chưa cao do lượng data dùng để train tương đối ít.
- Tốc độ đáp ứng khi xoay của các khớp robot chưa cao do cấu hình vi điều khiển Atmega2560 chưa đủ mạnh để điều khiển PID đồng thời 4 động cơ.

6.2. Hướng phát triển

Với những thành công của đề tài **“Thiết kế và chế tạo robot 4 bậc tự do tích hợp công nghệ xử lý ảnh để phân loại linh kiện lỗi”** mà nhóm đã đạt được, nhóm có những hướng phát triển sau:

- Cải thiện thiết kế và thi công mô hình robot sử dụng động cơ 57HS21A+J5710 cung cấp mô-men lớn hơn trong khi đó độ rơ chì ở mức 15-25 arcmin (1 arcmin=1/60 độ) nhằm tăng độ chính xác khi hoạt động của robot, nhờ đó robot có thể thực hiện được nhiều động tác phức tạp hơn, hoặc có thể sử dụng động cơ có hộp số không độ rơ với chi phí rất cao.
- Thay thế các cảm biến chữ U hiện tại bằng loại nhỏ hơn để tăng tính linh hoạt cho robot.
- Nghiên cứu các thuật toán quy hoạch quỹ đạo để áp dụng vào điều khiển robot giúp cho robot vận hành êm hơn.
- Thay thế khớp nối mềm bằng khớp nối các đằng để tăng khả năng chịu tải cho robot.
- Phát triển ứng dụng xử lý ảnh để nhận dạng nhiều loại linh kiện khác nhau và tăng độ chính xác khi nhận diện.

TÀI LIỆU THAM KHẢO

[1]. Phùng Hưng Bình, Phạm Việt Hoàng (2021), “Robot 6 bậc tự do điều khiển bằng giọng nói”, Luận án tốt nghiệp, Trường Đại học Sư phạm kỹ thuật thành phố Hồ Chí Minh.

[2]. John J.Craig (2005), “Introduction to Robotics”, Third. United States of America: Pearson Prentice Hall.

[3]. Sơn Dương, Hướng dẫn toàn tập về cách sử dụng Android Studio, <<https://vntalking.com/cach-su-dung-android-studio.html>>.

[4]. Mắt Bão, Firebase là gì? Giải pháp lập trình không cần Backend từ Google, <<https://wiki.matbao.net/firebase-la-gi-giai-phap-lap-trinh-khong-can-backend-tu-google/>>.

[5]. Hưng Bình, GUI Creation Step-by-Step, <<https://economic-anchovy-981.notion.site/GUI-Creation-Step-by-Step-bdce072336084d52b6a7105cd08420e7>>.

[6]. Nguyễn Đình Phúc (2018), “Phân Đoạn Ảnh Dựa Trên Thuật Toán Nở Vùng”, Luận án tốt nghiệp, Trường Đại học Dân Lập Hải Phòng.

[7]. Lê Nam (2018), “Một Số Phương Pháp Phân Đoạn Ảnh Theo Nguồn”, Luận án tốt nghiệp, Trường Đại học Điện Lực.

[8]. Trương Thanh Sang, Phan Hữu Phúc (2021), “Nhận Diện Động Vật Bằng Yolov5”, Báo cáo tiểu luận, Trường Đại học Sư phạm kỹ thuật thành phố Hồ Chí Minh.

[9]. Trần Văn Vinh, Nguyễn Thành Nhân (2014), “Mạch điều khiển động cơ DC dùng IR2184”.

[10]. Wikipedia, “Bộ điều khiển PID”<https://vi.wikipedia.org/wiki/B%CE%BB%99_%C4%91i%CE%BB%81u_khi%CE%BB%83n_PID>.

[11]. Quốc Bảo, “Bạn có biết Arduino là gì không” <<http://arduino.vn/bai-viet/40-ban-co-biet-arduino-la-gi-khong-tim-hieu-them>>

[12]. TS. Nguyễn Văn Thái (2016), “Đóng học thuận tay máy”, https://www.youtube.com/watch?v=Rvod_NM4Vso.

[13]. TS. Nguyễn Văn Thái (2016), “Đóng học nghịch tay máy”, <https://www.youtube.com/watch?v=0vnku9z3sNY&list=PLn3q6sc1zsV4np-f2wUTYS377lyU8oJ-o&index=17>.

[14]. *Blog chia sẻ về trí tuệ nhân tạo, “Google Colab – Hướng dẫn sử dụng cơ bản”* <<https://trituenhantao.io/lap-trinh/lam-quen-voi-google-colab/>>.

[15]. *Phạm Đình Khanh, “Bài 21 - Tiết xử lý ảnh OpenCV”*<<https://phamdinhkhanh.github.io/2020/01/06/ImagePreprocessing.html>>.

