

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH
KHOA ĐIỆN – ĐIỆN TỬ
BỘ MÔN CNKT ĐIỆN – ĐIỆN TỬ

ĐỒ ÁN TỐT NGHIỆP



HCMUTE

Đề tài: THIẾT KẾ VÀ ỨNG DỤNG CẢM BIẾN XÚC GIÁC VÀO BÀN TAY ROBOT

NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN – ĐIỆN TỬ

SVTH: Nguyễn Ngọc Khiêm 18142314
Nguyễn Hoàng Xuân 17142350
Nguyễn Trần Văn Thịnh 18142390

GVHD: ThS. Phan Văn Hoàn

Tp. Hồ Chí Minh, tháng 6 năm 2022

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH
KHOA ĐIỆN – ĐIỆN TỬ
BỘ MÔN CNKT ĐIỆN – ĐIỆN TỬ

ĐỒ ÁN TỐT NGHIỆP



HCMUTE

Đề tài: THIẾT KẾ VÀ ỨNG DỤNG CẢM BIẾN XÚC GIÁC VÀO BÀN TAY ROBOT

NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN – ĐIỆN TỬ

SVTH: Nguyễn Ngọc Khiêm 18142314
Nguyễn Hoàng Xuân 17142350
Nguyễn Trần Văn Thịnh 18142390

GVHD: ThS. Phan Văn Hoàn

Tp. Hồ Chí Minh, tháng 6 năm 2022

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin sinh viên

Họ và tên: Nguyễn Ngọc Khiêm MSSV: 18142314

Email: 18142314@student.hcmute.edu.vn

Họ và tên: Nguyễn Hoàng Xuân MSSV: 17142350

Email: 17142350@student.hcmute.edu.vn

Họ và tên: Nguyễn Trần Văn Thịnh MSSV: 18142390

Email: 18142390@student.hcmute.edu.vn

2. Thông tin đề tài

Tên của đề tài: THIẾT KẾ VÀ ỨNG DỤNG CẢM BIẾN XÚC GIÁC VÀO BÀN TAY ROBOT

Mục đích của đề tài: Nghiên cứu, phân tích và ứng dụng cảm biến xúc giác vào bàn tay robot để tăng tính linh hoạt cho robot.

Đồ án tốt nghiệp được thực hiện tại: Bộ môn Điện – Điện Tử, Khoa Điện – Điện Tử, Trường Đại Học Sư Phạm Kỹ Thuật Thành Phố Hồ Chí Minh.

Thời gian thực hiện: Từ ngày 1/3/2022 đến ngày 8/7/2022

3. Các nhiệm vụ cụ thể của đề tài

- Phân tích cảm biến xúc giác
- Thiết kế bàn tay robot
- Thiết kế mạch cho hệ thống
- Lập trình phần mềm cho hệ thống
- Cung cấp giao diện cho người dùng

4. Lời cam đoan của sinh viên

Chúng tôi – Nguyễn Ngọc Khiêm, Nguyễn Hoàng Xuân và Nguyễn Trần Vân
Thịnh cam đoan Đồ án tốt nghiệp là công trình nghiên cứu của bản thân chúng tôi dưới
sự hướng dẫn của thạc sỹ Phan Văn Hoàn.

Các kết quả công bố trong Đồ án tốt nghiệp là trung thực và không sao chép từ
bất kỳ công trình nào khác.

Tp.HCM, ngày tháng 6 năm 2022

SV thực hiện đồ án

Nguyễn Ngọc Khiêm

Nguyễn Hoàng Xuân

Nguyễn Trần Văn Thịnh

Giáo viên hướng dẫn xác nhận về mức độ hoàn thành và cho phép được bảo vệ:

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

Tp.HCM, ngày tháng năm 2014

Xác nhận của Bộ Môn

Giáo viên hướng dẫn

(Ký ghi rõ họ tên và học hàm học vị)

LỜI CẢM ƠN

Lời nói đầu tiên chúng tôi xin gửi lời cảm ơn chân thành đến thầy **Phan Văn Hoàn**. Thầy đã dành thời gian quý báu trực tiếp hướng dẫn và tận tình giúp đỡ, tạo điều kiện để chúng tôi hoàn thành đề tài này. Bên cạnh đó, chúng tôi cũng xin chân thành cảm ơn quý **thầy cô trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh** nói chung và quý **thầy cô khoa Điện – Điện tử** nói riêng, cùng với tri thức và tâm huyết của mình đã truyền đạt những kiến thức quý báu, tạo dựng nền móng đầu tiên cho chúng em có cơ sở cũng như những kiến thức cần thiết trong suốt thời gian học tập tại trường, nhờ đó em có thể hoàn thành tốt đồ án tốt nghiệp.

Đồng thời, nhóm em cũng xin cảm ơn các thầy cô trong khoa Điện - Điện Tử đã tạo điều kiện, cung cấp những kiến thức cơ bản, cần thiết cho nhóm em trong suốt quá trình thực hiện đề tài.

Cảm ơn gia đình, cha mẹ đã là nguồn động viên to lớn về vật chất và tinh thần trong suốt thời gian học tập và thực hành. Để nhóm em có được ngày hôm nay, được theo đuổi ước mơ và sự nghiệp.

Nhóm em xin cảm ơn sự chỉ bảo của các anh chị đi trước. Các anh chị đã hướng dẫn và giới thiệu tài liệu tham khảo thêm trong việc thực hiện nghiên cứu đề tài.

Cuối cùng xin kính chúc quý thầy, cô, gia đình và bạn bè nhiều sức khỏe, thành công trong công việc.

Xin chân thành cảm ơn!

TÓM TẮT

TÓM TẮT

Sự mở rộng lớn của lĩnh vực robot trong những thập kỷ qua đã tạo ra mối quan tâm ngày càng lớn đến việc nghiên cứu và phát triển các giải pháp cảm nhận giác quan để tích hợp vào các bộ phận của robot. Xúc giác là một yếu tố thiết yếu để robot có được thao tác khéo léo và linh hoạt giống với con người. Nó cung cấp thông tin về lực tương tác và đặc tính bề mặt tại các điểm tiếp xúc giữa các cánh tay robot và các đối tượng. Những tiến bộ gần đây trong lĩnh vực cảm nhận xúc giác của robot đã dẫn đến sự phát triển của nhiều kỹ thuật tính toán khai thác giác quan quan trọng này.

. Để đáp ứng nhu cầu trên chúng tôi quyết định thực hiện đề tài “**THIẾT KẾ VÀ ỨNG DỤNG CẢM BIẾN XÚC GIÁC VÀO BÀN TAY ROBOT**”. Trong thực tế, có rất nhiều phương pháp và vật liệu có thể được sử dụng để tạo ra cảm biến xúc giác với độ linh hoạt và tính phù hợp. Đề tài này cung cấp đánh giá và phân tích sơ bộ một số phương pháp chế tạo cảm biến xúc giác với các vật liệu có cơ chế khác nhau và đưa ra giải pháp cụ thể được cho là phù hợp với đề tài.

Việc mô tả về các giải pháp cảm biến xúc giác từ quan điểm của kiến trúc, thiết kế và hiệu suất cũng mang lại cái nhìn sơ bộ về công nghệ này và mang tới triển vọng về các ứng dụng đầy hứa hẹn hơn nữa trong lĩnh vực robot trong tương lai gần.

MỤC LỤC

MỤC LỤC

LỜI CẢM ƠN	i
TÓM TẮT	ii
MỤC LỤC	iii
MỤC LỤC HÌNH	vii
MỤC LỤC BẢNG	xv
CHƯƠNG 1. TỔNG QUAN	1
1.1. ĐẶT VĂN ĐỀ	1
1.2. MỤC TIÊU	2
1.3. NỘI DUNG NGHIÊN CỨU	3
1.4. GIỚI HẠN	4
1.5. BỐ CỤC	4
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	6
2.1. CẢM BIẾN XÚC GIÁC	6
2.1.1. Khái quát về xúc giác	6
2.1.2. Công nghệ cảm biến xúc giác	6
2.2. VI ĐIỀU KHIỂN STM32F103C8T6	11
2.2.1. Reset And Clock Control.....	11
2.2.2. GPIO và AFIO.....	13
2.2.3. Ngắt.....	14
2.2.4. Bộ chuyển analog sang digital (ADC)	14

MỤC LỤC

2.2.5. Bộ định thời (Timer)	17
2.2.6. Giao thức UART	19
2.2.7. Giao thức USB CDC	21
2.3. Hệ điều hành FreeRTOS (sử dụng CMSIS_OS API V2)	24
2.3.1. Quản lý bộ nhớ trong FreeRTOS	24
2.3.2. Cấp phát bộ nhớ trong FreeRTOS	26
2.3.3. Bộ lập lịch (Scheduler)	29
2.3.4. Ngắt trong hệ điều hành FreeRTOS	30
2.3.5. Tác vụ (task)	31
2.3.6. Giao tiếp giữa các tác vụ	33
2.3.7. Bộ định thời tạo bởi phần mềm (Software timer)	34
2.4. MÔ HÌNH HỌC SÂU SỬ DỤNG YOLOV5.	35
2.4.1. CNN (mạng nơ-ron tích chập)	35
2.4.2. Bài toán object detection	38
2.4.3. YOLOv5	39
2.4.4. Default box	41
2.4.5. Non-max suppression	43
CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG.....	44
3.1. SƠ ĐỒ KHỐI.....	44
3.1.1. Sơ đồ khái niệm	44
3.2. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG	46

MỤC LỤC

3.2.1. Khối cảm biến	46
3.2.2. Khối động cơ	56
3.2.3. Khối điều khiển trung tâm	62
3.2.4. Khối giao diện	72
3.2.5. Khối nguồn	76
CHƯƠNG 4. THI CÔNG HỆ THỐNG	83
4.1. GIỚI THIỆU	83
4.2. THI CÔNG HỆ THỐNG	83
4.2.1. Thi công board mạch	83
4.2.2. Thi công mô hình	95
4.2.3. Chuẩn bị dữ liệu huấn luyện mô hình học sâu	113
4.3. LẬP TRÌNH HỆ THỐNG	115
4.3.1. Lưu đồ giải thuật	115
4.3.2. Phần mềm lập trình cho vi điều khiển	128
4.3.3. Phần mềm lập trình cho giao diện người dùng	137
4.3.4. Phần mềm lập trình để nhận dạng vật thể	144
CHƯƠNG 5. KẾT QUẢ, NHẬN XÉT VÀ ĐÁNH GIÁ	148
5.1. KẾT QUẢ	148
5.1.1. Kết quả huấn luyện mô hình học sâu	148
5.1.2. Đánh giá mô hình học sâu trên tập dữ liệu	149
5.1.3. Kết quả nghiên cứu	151

MỤC LỤC

5.1.4. Kết quả thi công	151
5.2. NHẬN XÉT ĐÁNH GIÁ	164
CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	165
6.1. KẾT LUẬN	165
6.2. HƯỚNG PHÁT TRIỂN.....	165
TÀI LIỆU THAM KHẢO	167
PHỤ LỤC	168

MỤC LỤC HÌNH

MỤC LỤC HÌNH

Hình 2.1. Mảng cảm biến xúc giác Piezoresistive: (a) minh họa về sự thay đổi điện trở trong cao su dẫn điện, (b) hình ảnh quy mô nano của cao su dẫn điện, (c) cấu trúc của mảng xúc giác piezoresistive, (d) cảm biến xúc giác vải piezoresistive, (e) sơ đồ lớp điện cực của cảm biến xúc giác hình 3D, (f) hình ảnh xúc giác của mảng cảm biến áp suất phản ứng áp	7
Hình 2.2. Công nghệ cảm biến xúc giác điện dung: (a) điện dung của tụ điện bǎn song song phụ thuộc vào khoảng cách giữa các bǎn d và diện tích của các bǎn A (q là diện tích tích trữ); (b) hai bǎn dẫn điện được ngăn cách bởi một chất điện môi đàn hồi - khi có lực tác dụng, khoảng cách giữa các bǎn cực giảm, làm thay đổi điện dung; (c) lưỡi các cảm biến điện dung hình tam giác cho lòng bàn tay của robot hình người iCub.....	9
Hình 2.3. Cảm giác xúc giác áp điện: (a) hiệu ứng áp điện - một lực tác dụng gây ra sự sắp xếp lại các phần tử Si dương và O₂ âm dẫn đến sự gia tăng điện thế; (b) mảng cảm biến xúc giác dựa trên hiệu ứng áp điện với các điện cực ở lớp dưới cùng, vật liệu áp điện ở giữa và cao su ở trên cùng, (c) mô hình giản đồ của tactel cảm biến áp điện.	10
Hình 2.4. Hình ảnh của vi điều khiển STM32F103C8T6	11
Hình 2.5. Mạch reset.	12
Hình 2.6. Sơ đồ thời gian	15
Hình 2.7. Dữ liệu được cǎn phǎi	16
Hình 2.8. Chế độ capture/compare kênh 1.	18
Hình 2.9. Dạng sóng Edge-aligned PWM.	19
Hình 2.10. Lập trình độ dài 1 khung dữ liệu UART	21

MỤC LỤC HÌNH

Hình 2.11. Một frame của USB control transfer	23
Hình 2.12. Quản lý dung lượng các task trong bộ nhớ heap	25
Hình 2.13. Phân vùng bộ nhớ heap cho hệ điều hành	27
Hình 2.14. Mô tả sự liên kết khi các khối bộ nhớ được giải phóng trong heap_4.	28
Hình 2.15. Mô tả các ngắt trong hệ điều hành	30
Hình 2.16. Phân loại các ngắt trong chương trình sử dụng FreeRTOS	31
Hình 2.17. Tương tác giữa các trạng thái của tác vụ với nhau trong FreeRTOS.	32
Hình 2.18. Mô tả hoạt động của Queue.....	34
Hình 2.19. Hoạt động của chế độ định kỳ của bộ định thời	35
Hình 2.20. Hoạt động của chế độ 1 xung của bộ định thời.....	35
Hình 2.21. Hình nhân tích chập	36
Hình 2.22. Ví dụ pooling layer	37
Hình 2.23. Đồ thị hàm ReLU	38
Hình 2.24. Ảnh minh họa đầu ra của bài toán object detection	39
Hình 2.25. Hiệu suất của các phiên bản YOLOv5 trên tập dữ liệu COCO.....	40
Hình 2.26. Cấu trúc của mô hình yolov5.....	41
Hình 2.27. Chia ảnh thành nhiều vùng anchor box	42
Hình 2.28. Ảnh minh họa cách tính IoU	43
Hình 3.1. Sơ đồ khái niệm	44
Hình 3.2. (a): Thiết kế lắp ráp (b): Mạch điều khiển cảm biến	47
Hình 3.3. Thiết kế lắp ráp cảm biến xúc giác của chúng tôi	48

MỤC LỤC HÌNH

Hình 3.4. Tấm Velostat	50
Hình 3.5. Sơ đồ nguyên lý khói cảm biến.....	54
Hình 3.6. Tín hiệu điều khiển servo.....	57
Hình 3.7. Động cơ servo MG946R.....	59
Hình 3.8. Sơ đồ nguyên lý khói động cơ.....	61
Hình 3.9. Sơ đồ nguyên lý khói xử lý trung tâm	67
Hình 3.10. Sơ đồ nguyên lý khói điều khiển	70
Hình 3.11. Tính toán cặp dây D+ và D- cho USB trong phần mềm Altium	75
Hình 3.12. Sơ đồ nguyên lý phần kết nối khói xử lý trung tâm với khói giao diện	76
Hình 3.13. Khối nguồn trong nguyên lý mạch điều khiển.....	82
Hình 3.14. Nguồn tổ ong 5V 10A	82
Hình 4.1. Hình ảnh 2D mặt trước và sau của PCB cho cảm biến xúc giác ở các ngón tay trên Altium	84
Hình 4.2. Hình ảnh 3D mô phỏng mặt trước và sau của PCB cho cảm biến xúc giác ở các ngón tay trên Altium	84
Hình 4.3. Hình ảnh 2D mặt trước của PCB cho cảm biến xúc giác ở lòng tay trên Altium	85
Hình 4.4. Hình ảnh 2D mặt sau của PCB cho cảm biến xúc giác ở lòng tay trên Altium	85
Hình 4.5. Hình ảnh 3D mô phỏng mặt trước của PCB cho cảm biến xúc giác ở lòng tay trên Altium	86
Hình 4.6. Hình ảnh 3D mô phỏng mặt sau của PCB cho cảm biến xúc giác ở lòng tay trên Altium	86

MỤC LỤC HÌNH

Hình 4.7. Hình ảnh 2D mặt trước của mạch điều khiển chính trên Altium	87
Hình 4.8. Hình ảnh 2D mặt sau của mạch điều khiển chính trên Altium.....	88
Hình 4.9. Hình ảnh 3D mô phỏng mặt trước của mạch điều khiển chính trên Altium	89
Hình 4.10. Hình ảnh 3D mô phỏng mặt sau của mạch điều khiển chính trên Altium	90
Hình 4.11. Hình ảnh 2D mặt sau của mạch điều khiển phụ trên Altium	92
Hình 4.12. Hình ảnh 3D mô phỏng mặt trước của mạch điều khiển phụ trên Altium	93
Hình 4.13. Hình ảnh 3D mô phỏng mặt sau của mạch điều khiển phụ trên Altium	94
Hình 4.14. Phân chia các tầng của hệ thống cơ khí.....	96
Hình 4.15. Tổng quan cánh tay	97
Hình 4.16. Khớp trên của ngón tay robot	98
Hình 4.17. Khớp dưới của ngón tay robot	98
Hình 4.18. Thành phần chính cấu tạo nên khớp	99
Hình 4.19. Khớp nối của 2 khớp	100
Hình 4.20. Khớp dưới cố định chung với khớp nối.....	100
Hình 4.21. Nối 2 khớp thông qua khớp nối.....	100
Hình 4.22. Vị trí dự định của lòng bàn tay	101
Hình 4.23. Lòng bàn tay khi được nâng lên.....	102
Hình 4.24. Lòng bàn tay	102

MỤC LỤC HÌNH

Hình 4.25. Khớp đỡ cả ngón tay của robot.....	103
Hình 4.26. Nền của tầng bàn tay	103
Hình 4.27. Nền của tầng động cơ với các khớp đỡ ngón tay robot.....	104
Hình 4.28. Đĩa RC Servo có sẵn trên thị trường	105
Hình 4.29. Đĩa RC Servo được mô phỏng trên phần mềm Solidworks	105
Hình 4.30. Đĩa động cơ phụ trách kéo dây điều khiển ngón tay robot	106
Hình 4.31. Lắp ráp động cơ với phần đĩa để kéo ngón tay robot	106
Hình 4.32. Mặt trên của tầng động cơ.....	107
Hình 4.33. Mặt dưới của tầng động cơ.....	108
Hình 4.34. Bố trí động cơ ở tầng động cơ của bàn tay robot	109
Hình 4.35. Bố trí tầng mạch điện của bàn tay robot.....	110
Hình 4.36. Nền và cố định cho robot nằm ngang	111
Hình 4.37. Gá cố định với phần nền để đặt robot nằm ngang	111
Hình 4.38. Robot hoàn thiện sau khi lắp nền.....	112
Hình 4.39. Đánh nhãn dữ liệu	113
Hình 4.40. Tăng cường gói dữ liệu ảnh bằng roboflow	114
Hình 4.41. Dữ liệu ảnh sau khi tăng cường.....	114
Hình 4.42. Biểu đồ phân bố dữ liệu sử dụng cho quá trình huấn luyện	115
Hình 4.43. Tổng quan lưu đồ giải thuật của chương trình vi điều khiển	116
Hình 4.44. Lưu đồ tổng quát hoạt động và ưu tiên của các ngắt và các tác vụ ...	117
Hình 4.45. Lưu đồ giải thuật Process Output Task	118

MỤC LỤC HÌNH

Hình 4.46. Lưu đồ giải thuật Scan ADC Task.....	119
Hình 4.47. Lưu đồ giải thuật Transmit Task.....	121
Hình 4.48. Lưu đồ giải thuật Matrix Handler Task	122
Hình 4.49. Lưu đồ giải thuật Peak Control Task.....	123
Hình 4.50. Lưu đồ giải thuật Button Handler Task.....	124
Hình 4.51. Lưu đồ giải thuật UART RxHandler Task	125
Hình 4.52. Lưu đồ giải thuật tác vụ điều khiển servo theo lệnh nhận được.....	126
Hình 4.53. Lưu đồ giải thuật chương trình nhận dạng vật thể.....	127
Hình 4.54. Trang web tải phần mềm lập trình vi điều khiển.....	130
Hình 4.55. Trang đầu tiên trong cửa sổ cài đặt STM32CubeIDE	131
Hình 4.56. Trang thứ hai trong cửa sổ cài đặt STM32CubeIDE.....	131
Hình 4.57. Trang thứ ba trong cửa sổ cài đặt STM32CubeIDE.....	132
Hình 4.58. Cửa sổ cài đặt STM32CubeIDE bắt đầu quá trình cài đặt	133
Hình 4.59. Icon phần mềm STM32CubeIDE	133
Hình 4.60. Cửa sổ lựa chọn workspace cho phần mềm STM32CubeIDE	134
Hình 4.61. Tạo project trong STM32CubeIDE.....	134
Hình 4.62. Cửa sổ lựa chọn loại vi điều khiển cho dự án của STM32CubeIDE .	135
Hình 4.63. Đặt tên và đường dẫn cho dự án trong STM32CubeIDE.....	135
Hình 4.64. Cửa sổ làm việc sau khi tạo project của STM32CubeIDE	136
Hình 4.65. Cửa sổ làm việc sau khi tạo code của STM32CubeIDE.....	136
Hình 4.66. Một số chức năng chính trong STM32CubeIDE	137

MỤC LỤC HÌNH

Hình 4.67. Cửa sổ tạo dự án mới của Visual Studio 2022	138
Hình 4.68. Cửa sổ chọn template trong Visual Studio 2022.....	138
Hình 4.69. Cửa sổ cấu hình dự án trong Visual Studio 2022	139
Hình 4.70. Giao diện với cửa sổ giao diện trong Visual Studio 2022	140
Hình 4.71. Kéo thả nút nhấn trong Visual Studio 2022.....	140
Hình 4.72. Đổi văn bản nút nhấn trong cửa sổ Properties Visual Studio 2022 ...	141
Hình 4.73. Đổi tên nút nhấn trong cửa sổ Properties Visual Studio 2022	141
Hình 4.74. Nút chạy ứng dụng trong Visual Studio 2022.....	143
Hình 4.75. Ứng dụng nút nhấn đơn giản trong Visual Studio 2022	144
Hình 4.76. Logo Visual Studio Code và Python	145
Hình 4.77. Tạo workspace trên Visual Studio Code	146
Hình 4.78. Chạy Python file trong terminal	147
Hình 5.1. Đồ thị thể hiện loss, precision, recall trong quá trình huấn luyện.....	148
Hình 5.2. Confusion matrix	150
Hình 5.3. Giao diện chính của phần mềm khi mở phần mềm	152
Hình 5.4. Giao diện điều khiển động cơ servo	152
Hình 5.5. Giao diện hiển thị đồ họa ma trận cảm biến dưới dạng cánh tay.....	153
Hình 5.6. Giao diện hiển thị đồ họa ma trận cảm biến (dùng để debug)	153
Hình 5.7.	154
Hình 5.8. Cửa sổ nhận dạng vật thể của giao diện.....	154
Hình 5.9. Nhận dạng vật thể hủ snack khoai tây và hiển thị trên giao diện.....	155

MỤC LỤC HÌNH

Hình 5.10. Nhận dạng vật thể chai nước muối súc miệng và hiển thị trên giao diện	156
Hình 5.11. Giao diện linegraph hiển thị giá trị tối đa của các đốt trên, đốt dưới và lòng bàn tay	157
Hình 5.12. Robot thao tác với chai nước suối	158
Hình 5.13. Kết quả khi cho bàn tay bóp toàn lực vào chai nước suối	159
Hình 5.14. Kết quả linechart khi cho bàn tay bóp toàn lực vào chai nước suối..	159
Hình 5.15. Kết quả khi cho bàn tay bóp với lực cấp độ 7 vào chai nước suối.....	160
Hình 5.16. Kết quả linechart khi cho bàn tay bóp với lực cấp độ 7 vào chai nước suối.....	160
Hình 5.17. Cánh tay robot thao tác với hủ snack khoai tây	161
Hình 5.18. Kết quả khi cho bàn tay bóp toàn lực vào hủ snack khoai tây	162
Hình 5.19. Kết quả linechart khi cho bàn tay bóp toàn lực vào hủ snack khoai tây	162
Hình 5.20. Kết quả khi cho bàn tay bóp với lực cấp độ 6 vào hủ snack khoai tây	163
Hình 5.21. Kết quả linechart khi cho bàn tay bóp với lực cấp độ 6 vào hủ snack khoai tây	163

MỤC LỤC BẢNG

MỤC LỤC BẢNG

Bảng 2.1. Điện áp tham chiếu.....	16
Bảng 3.1. Dòng và áp các linh kiện trong hệ thống.....	77
Bảng 4.1. Danh sách linh kiện cho mạch trung tâm	91
Bảng 4.2. Danh sách linh kiện cho mạch điều khiển.....	95
Bảng 5.1. Bảng thống kê đánh giá trên tập thử.....	150

CHƯƠNG 1. TỔNG QUAN

CHƯƠNG 1. TỔNG QUAN

1.1. ĐẶT VẤN ĐỀ

Khả năng cầm nắm và điều khiển đồ vật một cách khéo léo của các robot hiện đại vẫn còn rất xa so với kỹ năng thao tác của con người. Mặc dù các bàn tay nhân tạo phức tạp, thậm chí là nhân hình, đã được chế tạo, công nghệ chế tác vẫn còn sơ khai, không chỉ do những hạn chế hiện tại của cả hệ thống truyền động và cảm biến được tích hợp vào các thiết bị phức tạp như vậy, mà còn do sự thiếu thuật toán điều khiển phù hợp sẽ cho phép robot thực hiện các tác vụ thao tác. Một trong những tính năng chính mà bất kỳ thiết bị cầm nắm nào cũng nên có là khả năng nắm bắt kỳ loại vật thể nào một cách chắc chắn nhất có thể, tránh bị trượt ngay cả khi có các nhiễu động bên ngoài tác động lên vật thể đó. Các tính năng như vậy đòi hỏi khả năng điều chỉnh lực nắm để cho phép robot thao tác với cả vật cứng và vật dễ vỡ cần phải nắm với lực tối thiểu cần thiết để giữ chúng mà không gây gãy. Ở bàn tay nhiều ngón, việc kiểm soát lực nắm bắt buộc phải đo đúng lực và mômen tiếp xúc cũng như vị trí tiếp xúc ở mỗi ngón tay để có thể ước lượng và kiểm soát đúng các lực bên ngoài và bên trong. Các loại phép đo này có thể được thực hiện với việc sử dụng kết hợp cảm biến lực và xúc giác hoặc bằng cách sử dụng cảm biến xúc giác tích hợp. Gần đây, loại cảm biến này đã được tích hợp vào các tay kẹp song song để kiểm soát cả trượt tuyến tính và quay của các vật thể cứng có hình dạng song song, chứng tỏ rằng nó cho phép nắm bắt an toàn các vật thể trong các điều kiện không chắc chắn, cụ thể là trọng lượng chưa biết, tâm của khối lượng và ma sát, cũng như chịu tác dụng của các lực bên ngoài. Bàn tay robot mềm không bị suy giảm gần đây đã được sử dụng để nắm các vật cứng và dễ vỡ một cách an toàn.

Nắm bắt là một vấn đề, nhưng làm thế nào về thao tác trong tay? Ngay từ cái nhìn đầu tiên, người ta có thể nghĩ rằng nó chỉ có thể được thực hiện bởi những bàn tay nhiều ngón. Tuy nhiên, các bàn tay robot nhân hình hiện có vẫn có độ tin cậy hạn chế và độ

CHƯƠNG 1. TỔNG QUAN

phức tạp cao, trong khi các tay robot mềm chưa phân tích có một số bậc tự do được kiểm soát hạn chế, do đó chúng chủ yếu được sử dụng để nắm bắt tuân thủ. Đó là lý do tại sao các ứng dụng thao tác chủ yếu sử dụng bộ kẹp đơn giản. Máy kẹp hàm song song cho đến nay là phổ biến nhất, do độ tin cậy, chi phí thấp, dễ điều khiển và tích hợp vào các robot công nghiệp tiêu chuẩn. Những loại kẹp như vậy có độ khéo léo hạn chế, tuy nhiên, các nhà nghiên cứu gần đây đã chứng minh rằng chúng có thể được sử dụng để thực hiện các thao tác khéo léo khai thác cái gọi là khái niệm khéo léo bên ngoài. Trong thực tế, sự khéo léo không chỉ được cung cấp bởi bậc tự do của thiết bị cầm nắm mà còn bởi các hỗ trợ bên ngoài như lực hấp dẫn hoặc các ràng buộc về môi trường mà chúng ta có thể tìm thấy các ví dụ về thao tác bằng tay với các thiết bị đơn giản. Nhiệm vụ tương tự đã được thực hiện chỉ bằng cách sử dụng phản hồi xúc giác. Trong đó, phép đo cả thành phần lực bình thường và lực tiếp tuyến cùng với mômen xoắn được chứng minh là có hiệu quả mà không cần thêm bất kỳ cảm biến bên ngoài nào. Tuy nhiên, những thí nghiệm này chỉ được thực hiện với các vật thể cứng có dạng trụ.

Để khắc phục nhược điểm nêu trên của các robot đang có trên thị trường, chúng tôi quyết định thực hiện đề tài “**THIẾT KẾ VÀ ỨNG DỤNG CẢM BIẾN XÚC GIÁC VÀO BÀN TAY ROBOT**”, nhằm nghiên cứu, xây dựng một hệ thống hỗ trợ xúc giác có sự phản hồi để nâng cao độ tin cậy của hệ thống, giúp cho robot có khả năng khéo léo một cách tự động trong quá trình điều khiển.

1.2. MỤC TIÊU

Trong đề tài này, tính toán và thiết kế bàn tay robot 3 ngón tích hợp với cảm biến xúc giác để có thể cầm nắm vật thể với lực vừa phải được trình bày. Thiết kế cảm biến mới bắt đầu từ yêu cầu chính là thao tác các vật thể có hình trụ để tránh trượt. Mục tiêu này dẫn đến sự cần thiết của một bề mặt tiếp xúc mềm để có thể giữ các mômen xoắn đáng kể bởi cảm biến. Để cho phép áp dụng các thuật toán tránh trượt khi miếng cảm

CHƯƠNG 1. TỔNG QUAN

biến mềm tương tác với một vật thể có hình dạng chung, ước tính hình dạng tiếp xúc là cần thiết để xác định các thông số cho quá trình phân tích xử lý để có dữ liệu hỗ trợ cho quá trình cầm nắm khéo léo của robot một cách tự động. Do đó cần thiết kế một bản đồ xúc giác với độ phân giải không gian phù hợp dựa trên yêu cầu về độ chính xác trong ước lượng ma trận xúc giác.

Ngoài ra, robot sẽ có khả năng nhận dạng được vật thể mà nó thao tác cũng như con người có thị giác để có thể ước lượng trước về độ lớn, hình dạng, khối lượng hay kể cả nhiệt độ của vật thể phía trước họ.Thêm vào đó một giao diện người dùng cũng sẽ được thiết kế để người dùng có thể điều khiển và nhận thông tin về dữ liệu xúc giác của robot

1.3. NỘI DUNG NGHIÊN CỨU

❖ Các nội dung mà chúng tôi cần thực hiện bao gồm:

- NỘI DUNG 1: Tìm hiểu nguyên lý, đánh giá, phân tích các vật liệu có thể chế tạo cảm biến xúc giác.
- NỘI DUNG 2: Đưa ra giải pháp thiết kế cảm biến và sơ đồ mạch điện liên quan.
- NỘI DUNG 3: Thiết kế và thi công mạch kiểm tra cảm biến xúc giác và giao tiếp với máy tính bằng UART và hiển thị dữ liệu lên giao diện người dùng .
- NỘI DUNG 4: Thiết kế và thi công cánh tay robot.
- NỘI DUNG 5: Thiết kế mạch điều khiển động cơ và mạch chính với chức năng chủ yếu để thu thập và trao đổi dữ liệu.
- NỘI DUNG 6: Viết chương trình cho vi điều khiển.
- NỘI DUNG 7: Viết giao diện người dùng cho dự án.
- NỘI DUNG 8: Lắp ráp các khối và tiến hành điều khiển thử nghiệm.

CHƯƠNG 1. TỔNG QUAN

- NỘI DUNG 9: Chính sửa các lỗi xuất hiện.
- NỘI DUNG 10: Viết báo cáo luận văn.
- NỘI DUNG 11: Bảo vệ luận văn.

1.4. GIỚI HẠN

❖ Các giới hạn của đề tài:

- Việc tự thi công cảm biến có thể mang lại sự khác nhau giữa các điểm lực
- Dữ liệu của cảm biến phức tạp, khó xử lý
- Hạn chế về mặt cơ khí, cũng như khả năng thiết kế cánh tay robot

1.5. BỐ CỤC

❖ Bố cục của các chương:

- Chương 1: Tổng quan.

Chương này trình bày đát vấn đề dẫn nhập lý do chọn đề tài, mục tiêu, nội dung nghiên cứu, các giới hạn thông số và bố cục đồ án.

- Chương 2: Cơ sở lý thuyết.

Giới thiệu công nghệ cảm biến xúc giác, hệ điều hành thời gian thực (RTOS), vi điều khiển STM32F103C8T6 với các ngoại vi sử dụng.

- Chương 3: Tính toán thiết kế.

Chương này trình bày sơ đồ khái hệ thống, cách tính toán các thông số kỹ thuật của các khói và lý do chọn linh kiện hay vật liệu phù hợp, thiết kế các khói với yêu cầu đặt ra ban đầu.

- Chương 4: Thi công hệ thống.

CHƯƠNG 1. TỔNG QUAN

Trình bày các mạch đã thiết kế và trình tự lắp ráp thi công, giao diện người dùng để tương tác với tay robot, hướng dẫn cách cài đặt, sử dụng các phần mềm lập trình và trình bày các thao tác hướng dẫn sử dụng hệ thống.

- Chương 5: Kết quả, nhận xét và đánh giá

Chương này trình bày các kết quả đạt được và chưa đạt.

- Chương 6: Kết luận và hướng phát triển.

Đưa ra đánh giá về những kết quả đạt được so với mục tiêu ban đầu, đề ra các phương án khắc phục và hướng phát triển của đề tài.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. CẢM BIẾN XÚC GIÁC

2.1.1. Khái quát về xúc giác

Xúc giác là những cảm giác có được khi đụng chạm, tiếp xúc bằng da (qua tay, chân,...). Những nhận thức này được coi là một trong năm giác quan của động vật có thể nhận biết hay ước lượng cấu tạo bề mặt, trọng lượng, độ lớn, nhiệt độ của vật chất hay đối tượng qua tiếp xúc bằng da, qua các động tác như rờ, nắn, nâng, cọ xát, ôm... Những nhận thức này được chuyển thảng lên não đánh giá và xác định nhiệt độ, sự nguy hiểm (tạo cảm giác đau đớn, nóng lạnh,... tạo bỗng và bị thương)...

2.1.2. Công nghệ cảm biến xúc giác

Thông tin về các thuộc tính tương tác có thể được thu thập từ các cảm biến nội vi, chẳng hạn như cảm biến góc khớp với cảm biến mô-men xoắn của cơ cấu truyền động và cảm biến xúc giác qua da. Mặc dù, cảm biến nội vi có thể cung cấp thông tin gần đúng về lực tương tác như được hiển thị ở những nơi khác, cảm biến xúc giác bên ngoài cung cấp thông tin đa phương thức và chính xác hơn nhiều về các thuộc tính tương tác. Do đó, cảm biến xúc giác có thể được định nghĩa là một công cụ có thể đánh giá một thuộc tính nhất định của một đối tượng thông qua tiếp xúc vật lý giữa bàn tay và đối tượng. Khi một cảm biến xúc giác được biểu thị bằng một mảng, mỗi phần tử cảm nhận của cảm biến được đề cập khác nhau trong tài liệu về người máy, ví dụ: tế bào cảm biến hay đơn vị phân loại.

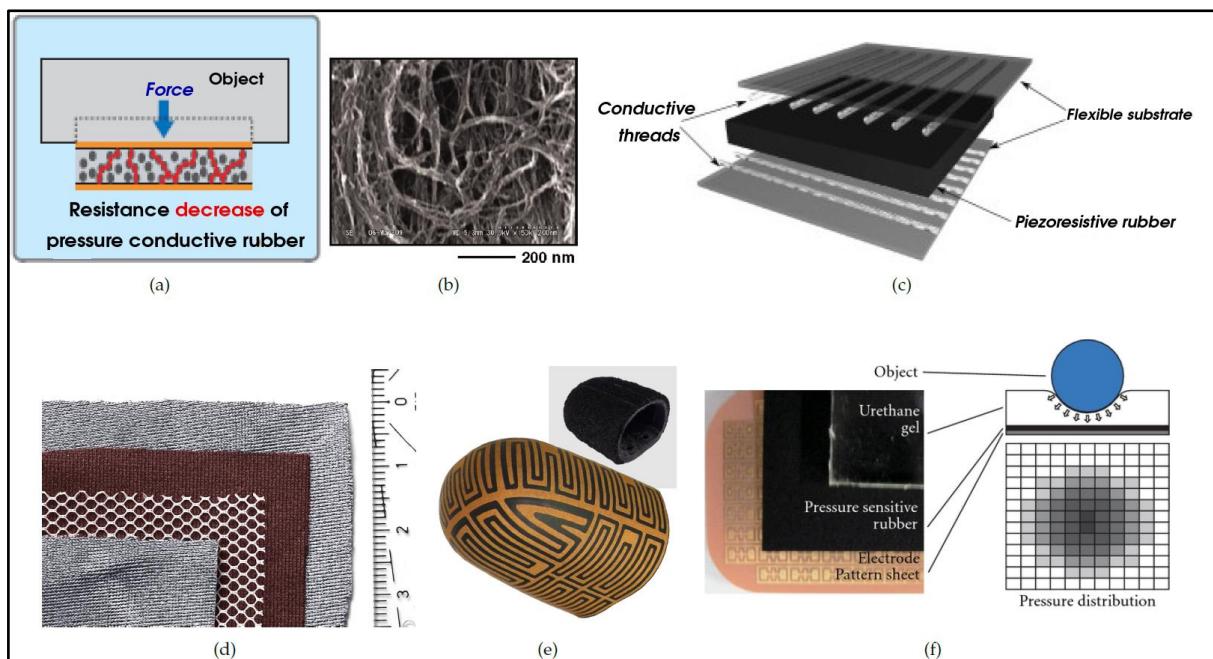
Cảm biến xúc giác đáp ứng các yêu cầu liên quan đến tác vụ về khả năng đáp ứng, khả năng thăm dò, khả năng thao tác của thao tác bằng tay.

Tùy thuộc vào nhiệm vụ, cảm biến có các thông số kỹ thuật thiết kế khác nhau. Các tiêu chí thiết kế cơ bản cho cảm biến xúc giác đã được báo cáo trước đây trong cho

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

robot hình người, trong kỹ thuật y sinh, trong tay giả và trong sản xuất và triển khai hệ thống xúc giác lớn. Trong các ứng dụng thao tác tự động, cảm biến xúc giác đáp ứng các yêu cầu về đặc tính và nhận dạng đối tượng (ví dụ: chúng ước tính sự tuân thủ, các đặc tính nhiệt và kết cấu) và cho thao tác (ví dụ: chúng kiểm soát lực tác dụng lên đối tượng).

Thay đổi điện dung, điện trở, sự phân bố quang học, điện tích có thể được sử dụng trong các hệ thống cảm biến. Và các loại cảm biến xúc giác khác nhau tùy thuộc vào quá trình truyền.



Hình 2.1. Mảng cảm biến xúc giác Piezoresistive: (a) minh họa về sự thay đổi điện trở trong cao su dẫn điện, (b) hình ảnh quy mô nano của cao su dẫn điện, (c) cấu trúc của mảng xúc giác piezoresistive, (d) cảm biến xúc giác vải piezoresistive, (e) sơ đồ lớp điện cực của cảm biến xúc giác hình 3D, (f) hình ảnh xúc giác của mảng cảm biến áp suất phản ứng áp

Sau đây, chúng tôi mô tả các loại cảm biến xúc giác cơ bản và phương pháp truyền tải của 3 loại được sử dụng phổ biến.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

❖ Cảm biến piezoresistive:

Hiệu ứng piezoresistive là một quá trình vật lý trong đó điện trở thay đổi khi vật liệu bị biến dạng cơ học (Hình 2.1. (a)). Vật liệu sở hữu hiệu ứng này được gọi là piezoresistors.

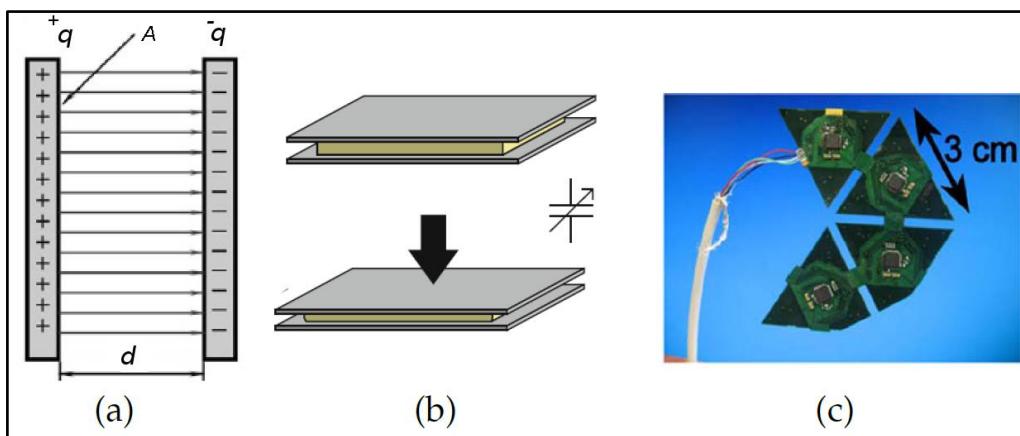
Có một số công nghệ để cảm nhận xúc giác nhân tạo dựa trên vật liệu phản ứng piezoresistive: Điện trở cảm ứng lực (FSR), cao su dẫn điện nhạy cảm với áp suất, bọt áp suất và vải phản ứng áp. Cách đơn giản nhất để kết hợp cảm biến xúc giác thông qua các thành phần rời rạc là sử dụng FSRs và chúng được sử dụng rộng rãi trong các thiết bị định vị như cần điều khiển. Cao su Piezoresistive là một vật liệu tổng hợp được tạo ra bằng cách trộn chất đàn hồi không dẫn điện với các hạt cacbon dẫn điện phân bố đồng nhất. Hình 2.1. (b) cho thấy cấu trúc của cao su dẫn điện ở cấp độ nano. Cảm biến dựa trên cao su dẫn điện với cấu trúc nhiều lớp như trong (Hình 2.1. (c), (d)) có thể bị tách lớp trên cùng. Điều này có thể tránh được bằng cách sử dụng một lớp cao su dẫn điện với một dây d้าย được khâu lại theo các hướng trực giao như trong. Một phương pháp khác để thiết kế mảng cảm biến xúc giác bằng cách sử dụng cao su dẫn điện kết hợp một mẫu không linh hoạt của các điện cực trên một lớp và cao su điện áp trên lớp thứ hai (Hình 2.1. (e)).

Điều đáng nói là các cảm biến xúc giác được phát triển hiện nay dựa trên cao su nhạy cảm với áp suất và các bóng bán dẫn hữu cơ, chẳng hạn như các bóng bán dẫn được sử dụng trong da sinh học, đặc biệt mỏng và rất linh hoạt. Mặt cao su dẫn điện được sử dụng trong cảm biến điện trở áp có đặc tính kháng lực phi tuyến. Do sử dụng vật liệu đàn hồi, các cảm biến có độ trễ nghiêm trọng. Độ nhạy trong cảm biến điện áp có thể giảm do mòn và rách, vì điện trở của cao su dẫn điện không chỉ phụ thuộc vào độ biến dạng mà còn phụ thuộc vào độ dày. Hơn nữa, các vật liệu được sử dụng trong cảm biến điện áp có thể thay đổi đặc tính của chúng do sự thay đổi của nhiệt độ và độ ẩm.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Piezoresistors cũng có độ lặp lại thấp hơn: sau nhiều lần biến dạng, vật liệu đàn hồi có thể không bao giờ lấy lại được hình dạng ban đầu. Một số mảng cảm biến áp suất cũng rất mỏng manh đối với lực cắt, ví dụ: Cảm biến xúc giác Weiss. Mặc dù có những nhược điểm này, một số tay robot kết hợp các mảng cảm biến xúc giác piezoresistive, vì các cảm biến này được sản xuất tương đối đơn giản, có thể linh hoạt và tồn tại nhiều giải pháp thương mại. So với các cảm biến điện dung sẽ được thảo luận trong phần tiếp theo, các cảm biến áp điện tốt hơn (không hoàn toàn) đối với các nhiễu điện từ.

❖ Cảm biến điện dung



Hình 2.2. Công nghệ cảm biến xúc giác điện dung: (a) điện dung của tụ điện bản song song phụ thuộc vào khoảng cách giữa các bản d và diện tích của các bản A (q là diện tích tích trữ); (b) hai bản dẫn điện được ngăn cách bởi một chất điện môi đàn hồi - khi có lực tác dụng, khoảng cách giữa các bản cực giảm, làm thay đổi điện dung; (c) lưới các cảm biến điện dung hình tam giác cho lòng bàn tay của robot hình người iCub.

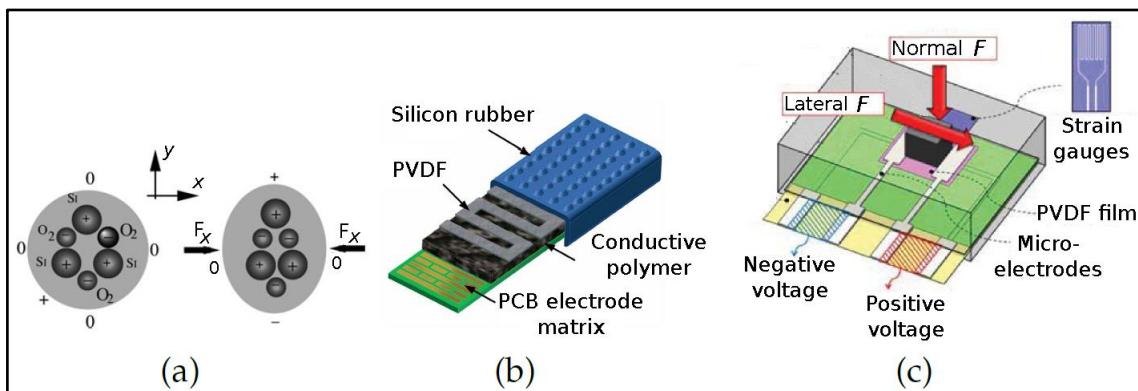
Cảm biến điện dung bao gồm hai tấm dẫn điện (Hình 2.2. (a)) được ngăn cách bởi vật liệu điện môi có thể nén được (Hình 2.2. (b)). Khi khe hở giữa các tấm thay đổi dưới tác dụng của lực thì điện dung cũng thay đổi. Bên cạnh các lực bình thường, lực cắt có thể được tính toán bằng cảm biến với việc sử dụng nhiều tụ điện nhúng. Các mảng cảm nhận áp suất có thể được xây dựng bằng các điện cực hàng và cột chồng lên nhau được

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

cách ly với nhau bằng chất điện môi đàn hồi. Có thể đạt được độ nhạy với các lực nhỏ bằng cách sử dụng các vật liệu đàn hồi có thể nén được nhiều hơn hoặc các cảm biến mỏng. Khi một bọt dẻo giữa hai tấm mỏng hơn một lượng điện tích nhỏ hơn trong cảm biến có thể được đo, điều đó có nghĩa là độ nhạy cao hơn.

Nhược điểm chính của cảm biến điện dung là nhạy cảm với nhiễu điện từ, nhạy cảm với nhiệt độ, phản ứng không tuyến tính và độ trễ. Ưu điểm của chúng bao gồm đáp ứng tần số cao hơn tương đối so với cảm biến điện áp. Kể từ khi công nghệ điện dung được sử dụng trong mọi ứng dụng cuộc sống hàng ngày, chẳng hạn như màn hình cảm ứng, loại cảm giác xúc giác này đã được nghiên cứu và sử dụng tốt trong robot và đặc biệt là trong tay robot.

❖ Cảm biến áp điện



Hình 2.3. Cảm giác xúc giác áp điện: (a) hiệu ứng áp điện - một lực tác dụng gây ra sự sắp xếp lại các phân tử Si dương và O₂ âm dẫn đến sự gia tăng điện thế; (b) mảng cảm biến xúc giác dựa trên hiệu ứng áp điện với các điện cực ở lớp dưới cùng, vật liệu áp điện ở giữa và cao su ở trên cùng, (c) mô hình giản đồ của tactel cảm biến áp điện.

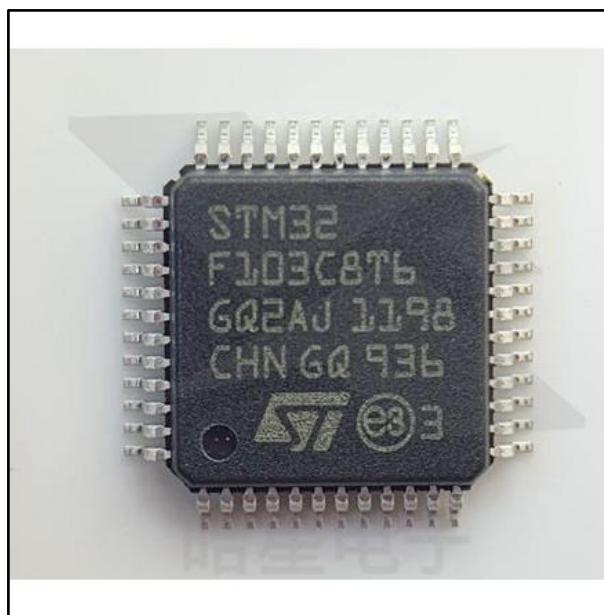
Hiệu ứng áp điện (Hình 2.3. (a)) được mô tả là sự tạo ra điện tích trong vật liệu tinh thể do sự biến dạng gây ra bởi lực hay áp suất tác dụng. Hiệu ứng áp điện được tạo ra trong tinh thể thạch anh, cũng như trong đồ gỗ và polyme do con người tạo ra, chẳng

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

hạn như polyvinylidene fluoride (PVDF). Một cảm biến xúc giác áp điện có thể được tạo ra với các dải phim PVDF được nhúng vào vật liệu cao su. Vật liệu áp điện, bị hạn chế cho phép đo động và được sử dụng trong cảm biến dựa trên siêu âm, thích hợp cho cảm biến xúc giác động. Trong số các vật liệu áp điện khác, polyme PVDF có các tính năng như tính linh hoạt và ổn định hóa học, nên được ưu tiên sử dụng trong các cảm biến cảm ứng.

Nhược điểm của chúng bao gồm sự dễ vỡ của các mối nối điện, độ nhạy nhiệt độ và chúng chỉ thích hợp cho các phép đo động lực học.

2.2. VI ĐIỀU KHIỂN STM32F103C8T6



Hình 2.4. Hình ảnh của vi điều khiển STM32F103C8T6

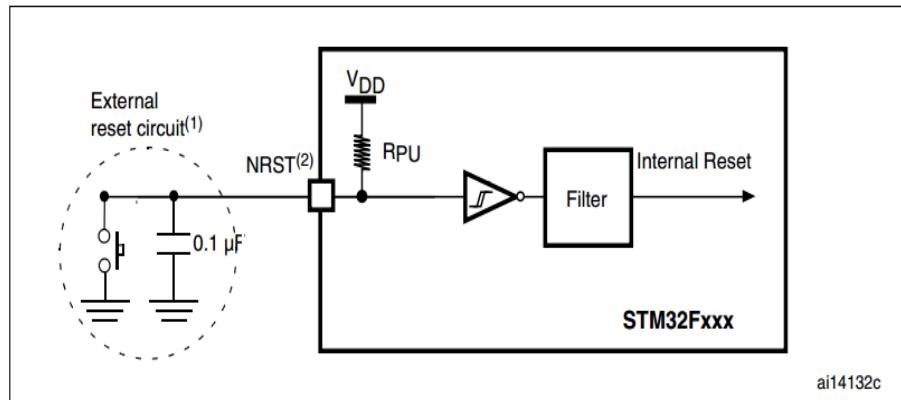
2.2.1. Reset And Clock Control

❖ Bộ reset của vi điều khiển.

STM32C8T6 có nhiều nguồn Reset khác nhau ngoài đường Reset bên ngoài. STM32C8T6 có thể bị buộc khởi động lại từ: các bộ Watchdogs nội, một Reset mềm

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

through NVIC, bộ Reset mở/tắt nguồn nội và mạch phát hiện điện áp nguồn thấp. Nếu một tín hiệu Reset xuất hiện, một bộ cờ trong thanh ghi kiểm soát và trạng thái RCC có thể được đọc để xác định nguyên nhân gây ra Reset. Trạng thái của những cờ này vẫn còn tồn tại cho đến khi hệ thống được cấp nguồn trở lại hoặc cho đến khi người dùng thiết lập bit Remove Reset.



Hình 2.5. Mạch reset.

❖ Xung clock của vi điều khiển STM32C8T6

Có 3 loại xung clock có thể được sử dụng để làm xung clock hệ thống (SYSCLK - xung clock cho khối xử lý).

- HSI (High Speed Internal-16MHZ) nguồn xung clock tốc độ cao bên trong Arm.
- HSE (High Speed External) nguồn xung clock tốc độ cao bên ngoài Arm.
- PLL (Phase Locked Loop) nguồn xung clock lấy từ bộ nhân tần số(hay sử dụng).
- Ngoài ra STM32C8T6 còn có các nguồn clock phụ khác như:
- LSI (Low speed internal) 32Khz dùng để cấp cho independent watchdog hay RTC dùng để định thời.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

- LSE (Low Speed External) nguồn xung clock tốc độ chậm thường được nối với thạch anh 32.768 kHz từ bên ngoài, xung clock này có thể được dùng để cấp cho RTC.

2.2.2. GPIO và AFIO.

❖ GPIO của vi điều khiển.

STM32C8T6 có 48 chân. Mỗi chân điều khiển có thể cấu hình như là GPIO hoặc có chức năng thay thế khác, hoặc mỗi chân có thể cùng lúc là nguồn ngắt ngoại.

Các cổng GPIO được đánh số A->C và mức tiêu thụ ở 5V. Nhiều chân có thể được cấu hình như là input/output tương tác với các thiết bị ngoại vi riêng của người dùng như USART hoặc I2C... Mỗi port I/O có hai thanh ghi cấu hình 32 bit (GPIOx_CRL, GPIOx_CRH), có hai thanh ghi dữ liệu 32 bit (GPIOx_IDR, GPIOx_ODR), thanh ghi set/reset 32-bit GPIOx_BSRR và thanh ghi reset 16-bit GPIOx_BRR. Tùy vào cách sử dụng mà GPIO được cấu hình là ngõ vào, ngõ ra, ngõ vào tương tự, hoặc vừa ngõ vào vừa ngõ ra.

❖ AFIO của vi điều khiển

Chức năng thay thế cho phép người dùng sử dụng các cổng GPIO với các ngoại vi khác. Để thuận tiện cho thiết kế phần cứng một thiết bị ngoại vi có thể được ánh xạ tới một hay nhiều chân của vi xử lý.

Sử dụng các tính năng thay thế của STM32C8T6 được điều khiển bởi các thanh ghi “Remap & Debug I/O”. Mỗi thiết bị ngoại vi (USART, CAN, Timers, I2C và SPI) có 1 hoặc 2 trường bit điều khiển ánh xạ đến các chân của vi điều khiển. Một khi các chân được cấu hình sử dụng chức năng thay thế, các thanh ghi điều khiển GPIO sẽ được sử dụng để điều khiển chức năng thay thế thay vì tác vụ I/O. Các thanh ghi Remap có điều khiển bộ JTAG. Khi hệ thống khởi động, cổng JTAG được kích hoạt tuy nhiên chức

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

năng theo dõi dữ liệu (data trace) vẫn chưa khởi động. JTAG có thể chuyển qua chế độ Debug, xuất dữ liệu theo dõi ra ngoài hoặc đơn giản chỉ sử dụng cổng GPIO.

2.2.3. Ngắt

NVIC (Nested Vector Interrupt Controller) là một khối nằm trong lõi Cortex của STM32 có nhiệm vụ quản lý các nguồn ngắt, do có rất nhiều nguồn có khả năng yêu cầu CPU ngắt nên chúng cần có một “người quản lý” chung để sắp xếp, điều khiển mọi hoạt động ngắt được diễn ra đúng theo ý muốn tránh trường hợp chồng chéo, xung đột giữa các yêu cầu

Để sử dụng NVIC cần phải qua ba bước cấu hình. Đầu tiên cấu hình bảng vector cho các nguồn ngắt mà ta muốn sử dụng. Tiếp theo cấu hình bảng thanh ghi NVIC để cho phép và thiết lập các mức ưu tiên của các ngắt trong NVIC và cuối cùng cần phải cấu hình các thiết bị ngoại vi và cho phép ngắt tương ứng. Bảng vector ngắt của Cortex bắt đầu ở vùng dưới của bảng địa chỉ. Tuy nhiên bảng vector bắt đầu từ địa chỉ 0x00000004 thay vì là 0x00000000 như ARM7 và ARM9, bốn byte đầu tiên được sử dụng để lưu trữ địa chỉ bắt đầu của con trỏ ngăn xếp.

Mỗi vector ngắt có độ rộng 4 byte và giữ địa chỉ bắt đầu của chương trình phục vụ ngắt tương ứng, 15 vector ngắt đầu tiên là các ngắt đặc biệt chỉ xảy ra trong lõi Cortex. Các ngắt ngoại vi người dùng bắt đầu từ vector 16, được định nghĩa bởi nhà sản xuất và được liên kết đến thiết bị ngoại vi. Trong phần mềm, bảng vector thường được giữ trong chương trình khởi động bằng cách định vị các địa chỉ trình phục vụ ngắt tại địa chỉ nền của bộ nhớ.

2.2.4. Bộ chuyển analog sang digital (ADC)

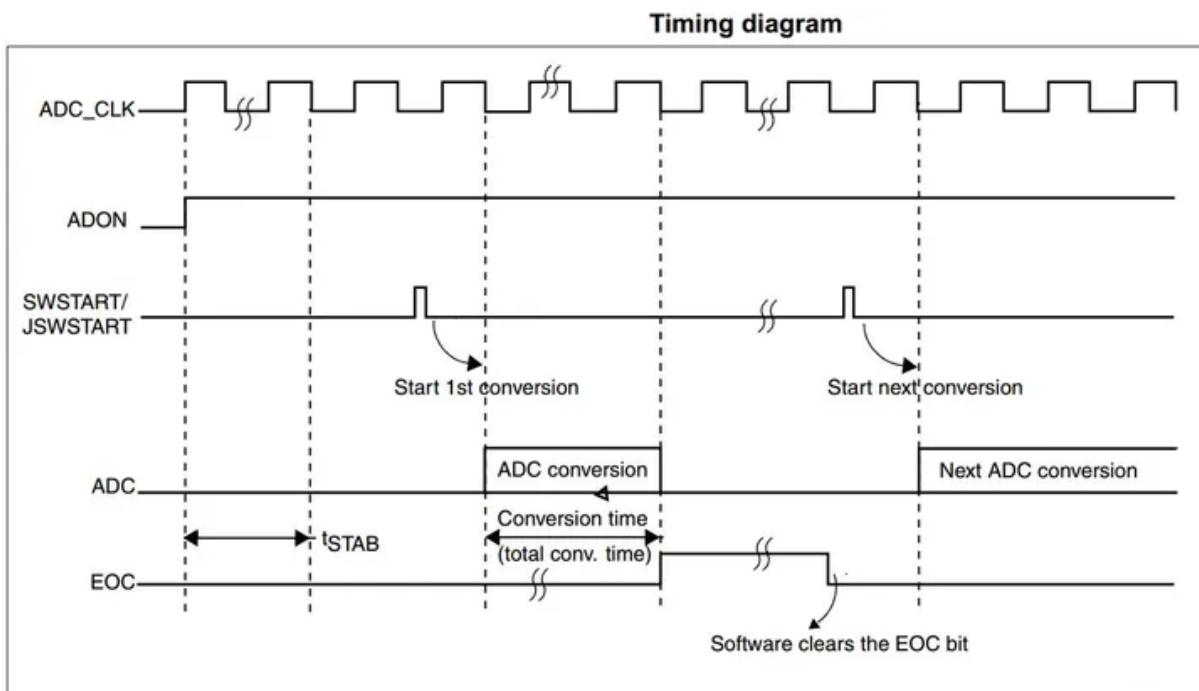
STM32F103C8T6 có ADC 12 bit là bộ chuyển đổi analog-digital gần. Nó có tới 18 kênh ghép kênh cho phép nó đo tín hiệu từ mười sáu nguồn bên ngoài và hai nguồn

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

bên trong. Chuyển đổi A/D của các kênh khác nhau có thể được thực hiện ở chế độ đơn, liên tục, quét hoặc không liên tục.

❖ Sơ đồ thời gian & thời gian chuyển đổi ADC

ADC cần một thời gian ổn định của tSTAB trước khi nó bắt đầu chuyển đổi chính xác. Sau khi bắt đầu chuyển đổi ADC và sau 14 chu kỳ đồng hồ, cờ EOC được thiết lập và thanh ghi dữ liệu (ADC_DR) ADC 16 bit chứa kết quả của chuyển đổi.

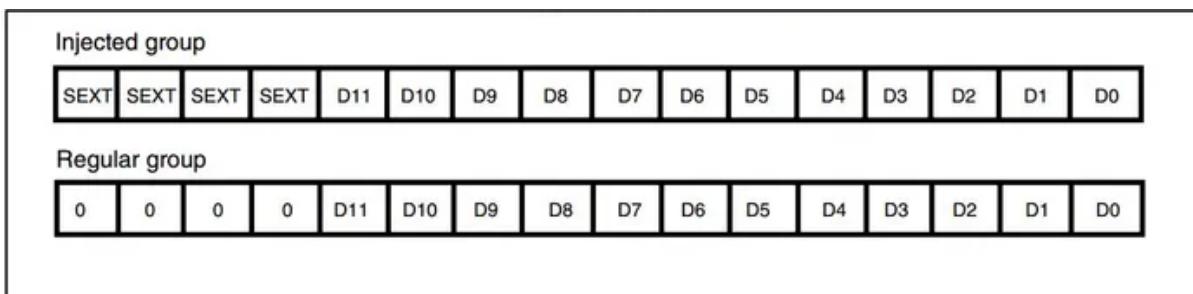


Hình 2.6. Sơ đồ thời gian

❖ Căn chỉnh dữ liệu kết quả ADC

Bit ALIGN trong thanh ghi ADC_CR2 chọn sự liên kết của dữ liệu được lưu trữ sau khi chuyển đổi. Dữ liệu có thể được căn trái hoặc căn phải như trong sơ đồ bên dưới.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT



Hình 2.7. Dữ liệu được căn phẩ

❖ Độ phân giải, tham chiếu, công thức

- Độ phân giải

STM32 ADC có độ phân giải 12 bit dẫn đến tổng thời gian chuyển đổi SamplingTime + 12,5 chu kỳ đồng hồ. Tuy nhiên, tỷ lệ lấy mẫu cao hơn có thể đạt được bằng cách hy sinh độ phân giải cao. Do đó, độ phân giải có thể được giảm xuống 10-Bit, 8-Bit hoặc 6-Bit, và do đó thời gian chuyển đổi ngắn hơn nhiều và tốc độ lấy mẫu tăng lên. Điều này có thể được lập trình cấu hình và triển khai trong phần mềm.

- Điện áp tham chiếu

Các chân điện áp tham chiếu ADC được xác định trong biểu dữ liệu và được giả định là được kết nối với mức điện áp trong một phạm vi nhất định.

Bảng 2.1. Điện áp tham chiếu

Name	Signal type	Remarks
V_{REF+}	Input, analog reference positive	The higher/positive reference voltage for the ADC, $2.4 \text{ V} \leq V_{REF+} \leq V_{DDA}$
$V_{DDA}^{(1)}$	Input, analog supply	Analog power supply equal to V_{DD} and $2.4 \text{ V} \leq V_{DDA} \leq 3.6 \text{ V}$
V_{REF-}	Input, analog reference negative	The lower/negative reference voltage for the ADC, $V_{REF-} = V_{SSA}$
$V_{SSA}^{(1)}$	Input, analog supply ground	Ground for analog power supply equal to V_{SS}

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Nếu bạn đang sử dụng bảng phát triển, bạn có thể cần phải kiểm tra sơ đồ của nó vì nó có thể không kết nối ADC Vref hoặc kết nối nó với 2.5v chặng hạn, vì vậy ADC sẽ bão hòa và cung cấp cho bạn 4096 trước khi điện áp analog đầu vào đạt 3.3v và bạn đang thắc mắc tại sao! có thể là do điện áp tham chiếu được đặt ở giá trị nhỏ hơn 3,3v, vì vậy, đó là điều cần xem xét.

❖ Công thức

Thời gian chuyên đổi ADC

- $T_{conv} = \text{Thời gian lấy mẫu} + 12,5 \text{ chu kỳ}$

Tỷ lệ lấy mẫu ADC

- $\text{SamplingRate} = 1 / T_{conv}$

Điện áp kết quả ADC (Giá trị đầu vào tương tự)

- $V_{in} = \text{ADC_Res} \times (\text{Điện áp tham chiếu} / 4096)$

Trong đó: Điện áp Tham chiếu = (VREF+) - (VREF-)

2.2.5. Bộ định thời (Timer)

STM32 có bốn khối định thời. Timer1 là khối nâng cao dành cho điều khiển động cơ, 3 khối còn lại đảm nhiệm chức năng đa nhiệm. Tất cả chúng đều có chung kiến trúc, khối nâng cao sẽ có thêm các đặc tính phần cứng riêng.

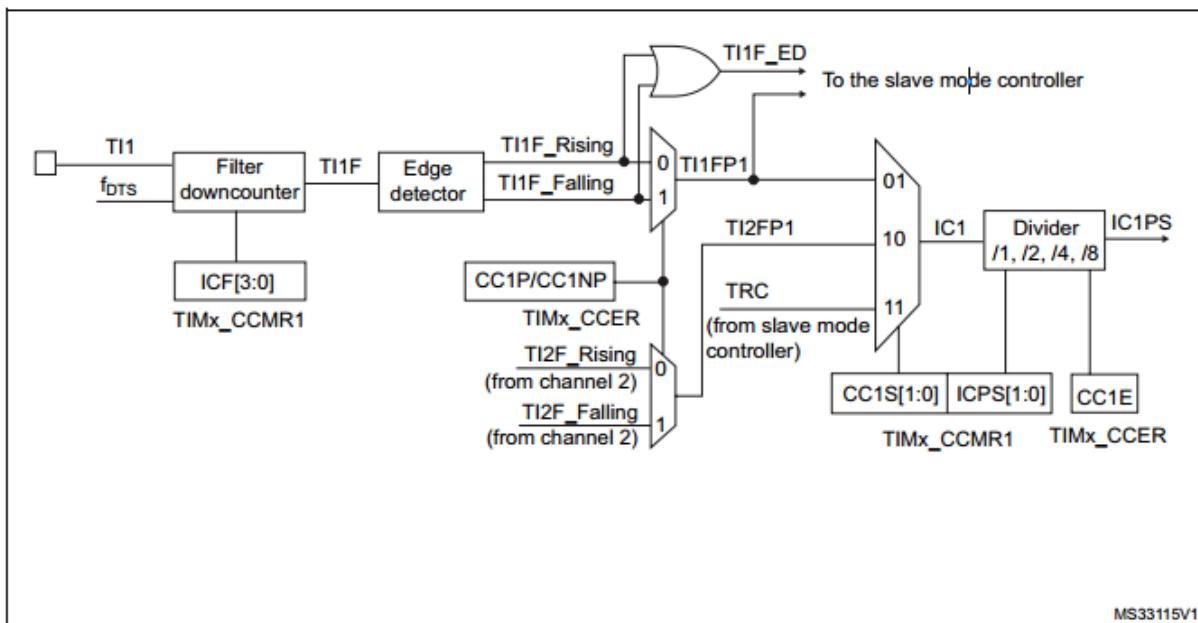
Tất cả các bộ TIMER đều gồm bộ đếm 16-bit với thanh ghi chia tần số dao động 16-bit (prescaler) và thanh ghi tự nạp (auto-reload). Bộ đếm của khối định thời có thể được cấu hình để đếm lên, đếm xuống hay trung tính (lên xuống xen kẽ nhau). Xung nhịp cho đồng hồ có thể được lựa chọn dựa trên 8 nguồn khác nhau: từ đồng hồ chuyên biệt được lấy từ đồng hồ hệ thống, từ xung nhịp chân ra lấy từ khối định thời khác, hoặc

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

từ nguồn xung nhịp ngoại. Khối định thời sử dụng cổng chọn để lấy xung nhịp đầu vào thích hợp, người dùng có thể sử dụng chân ETR để điều khiển cổng chọn này.

❖ Kênh capture/compare

Mỗi kênh Capture/ Compare được điều khiển bởi duy nhất một thanh ghi. Chức năng của thanh ghi này có thể thay đổi tùy thuộc cấu hình. Ở chế độ Capture, thanh ghi này có nhóm các bit đảm nhiệm thiết bị lọc dữ liệu đầu vào và chế độ đánh giá các ngõ PWM. Một khối Capture cơ bản gồm có bốn kênh vào để cấu hình bộ phát hiện xung (Edge Detector). Khi một xung cạnh lên hay xung cạnh xuống được phát hiện, bộ đếm hiện thời sẽ được cập nhập vào các thanh ghi 16 bit Capture/ Compare. Khi sự kiện Capture xảy ra bộ đếm có thể được khởi động lại hay tạm ngưng. Một ngắt DMA có thể được sử dụng trong trường hợp này. Ở chế độ Compare, STM32 cung cấp hàm chuẩn so sánh và bộ tạo xung PWM.

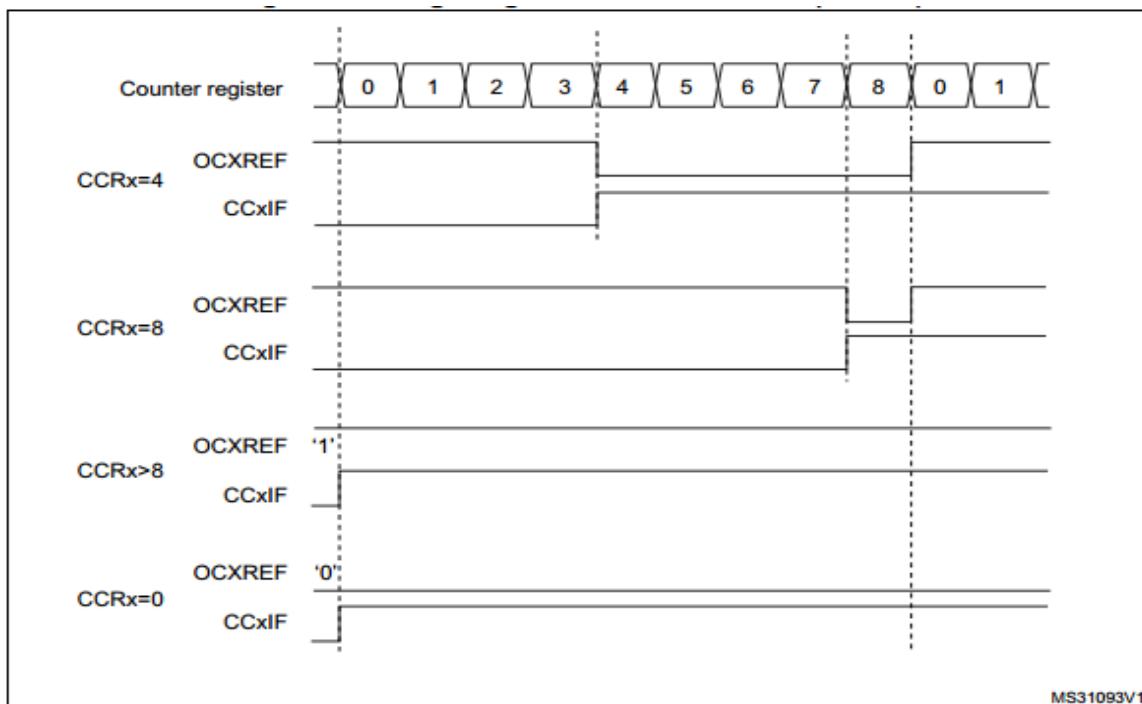


Hình 2.8. Chế độ capture/compare kênh 1.

❖ Chế độ PWM.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Mỗi khối Timer đều có khả năng tạo xung nhịp PWM. Ở chế độ xung PWM, giá trị Peroid được lưu vào thanh ghi Auto Reload còn giá trị Duty được lưu trong thanh ghi Capture/Compare. Có 2 kiểu tạo PWM, một là canh lề (edge-aligned) và canh lề giữa (center-aligned). Với Edge-aligned cạnh xuống của tín hiệu trùng với thời điểm thanh ghi reload cập nhật lại giá trị. Với center-aligned thời điểm thanh ghi reload cập nhật lại là khoảng giữa của chu kỳ Duty.



Hình 2.9. Dạng sóng Edge-aligned PWM.

2.2.6. Giao thức UART

Bộ thu/phát không đồng bộ đa năng hay viết tắt là UART đại diện cho mạch phần cứng (mô-đun) được sử dụng cho giao tiếp nối tiếp. UART được xem như một mạch tích hợp độc lập (IC) hoặc như một mô-đun bên trong bộ vi điều khiển. Trong hướng dẫn này, chúng tôi thực sự quan tâm đến mô-đun UART nội bộ trong Bộ vi điều khiển STM32.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Thực tế có hai dạng phần cứng UART như sau:

UART – Bộ thu/Bộ phát không đồng bộ đa năng

USART - Bộ thu/Bộ phát đồng bộ/không đồng bộ đa năng

Loại phát đồng bộ tạo ra clock dữ liệu và gửi nó đến máy thu hoạt động tương ứng theo cách đồng bộ. Mặt khác, kiểu máy phát không đồng bộ tạo ra clock dữ liệu bên trong. Không có tín hiệu clock nối tiếp đến, vì vậy để đạt được giao tiếp thích hợp giữa hai đầu, cả hai đầu phải sử dụng cùng tốc độ truyền.

❖ **UART trong STM32**

Máy phát thu không đồng bộ đa năng (USART) cung cấp phương tiện trao đổi dữ liệu song công linh hoạt với thiết bị bên ngoài yêu cầu định dạng dữ liệu nối tiếp không đồng bộ NRZ (Non-Return-To-Zero) theo tiêu chuẩn công nghiệp. USART cung cấp một loạt các tốc độ truyền bằng cách sử dụng bộ tạo tốc độ truyền phân đoạn.

Có thể dễ dàng nhận ra trong sơ đồ khái niệm cho mô-đun phần cứng USART này, có hai thanh ghi dịch chuyển riêng biệt và dữ liệu vào / ra được đệm kép cho hoạt động truyền và nhận dữ liệu song công. Cả hai thanh ghi dịch chuyển mà dữ liệu chuyển vào hoặc chuyển ra trong quá trình nhận/truyền đang được điều chỉnh tốc độ của mạch BRG (bộ tạo tốc độ truyền) ở cuối biểu đồ.

Có một thanh ghi địa chỉ cho chế độ giao tiếp đa xử lý. Có một đơn vị điều khiển luồng dữ liệu phần cứng để hỗ trợ tính năng này. Và cũng có một mạch giải mã IrDA, và bộ điều khiển ngắt để tạo ra các tín hiệu ngắt khác nhau trên các sự kiện phần cứng USART khác nhau.

❖ **Gói dữ liệu USART**

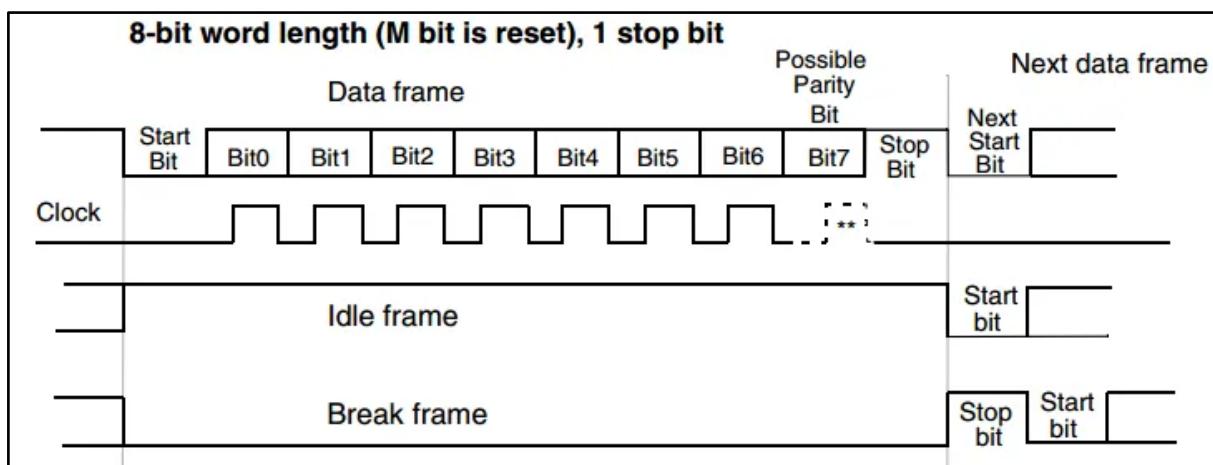
Độ dài từ có thể được chọn là 8 hoặc 9 bit bằng cách lập trình bit M trong thanh ghi USART_CR1. Quá trình truyền và nhận được điều khiển bởi một bộ tạo tốc độ truyền

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

chung, đồng hồ cho mỗi bộ được tạo ra khi bit cho phép được đặt tương ứng cho bộ phát và bộ thu. Chân TX ở trạng thái thấp trong thời gian bắt đầu bit. Nó ở trạng thái cao trong thời gian dừng bit.

Một ký tự Chờ được hiểu là toàn bộ khung gồm “1” theo sau là bit bắt đầu của khung tiếp theo chứa dữ liệu (Số “1” sẽ bao gồm số bit dừng).

Ký tự Break được diễn giải khi nhận được “0” trong một khoảng thời gian khung. Ở cuối khung Break, bộ truyền sẽ chèn 1 hoặc 2 bit dừng (bit logic “1”) để xác nhận bit bắt đầu.



Hình 2.10. Lập trình độ dài 1 khung dữ liệu UART

2.2.7. Giao thức USB CDC

Giao thức USB nói chung và STM32 USB CDC nói riêng là một giao thức khá khó lập trình. Nhưng ST đã hỗ trợ lập trình viên rất nhiều bằng cách tạo ra các gói source Midware USB hỗ trợ rất nhiều cho lập trình.

a. Định nghĩa giao thức USB

USB (Universal Serial Bus) là một chuẩn kết nối tuần tự đa dụng trong máy tính. Dùng để kết nối các ngoại vi với máy tính theo quy chuẩn Plug and Play (cắm và chạy),

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

với tính năng cảm nóng thiết bị (không phải khởi động lại hệ thống) giúp việc kết nối trở nên đơn giản hơn rất nhiều.

Giao thức USB được phát hành vào năm 1996, hiện đang được duy trì bởi tổ chức USB-IF. Theo wikipedia

Đã có bốn thế hệ USB, gồm: USB 1.x, USB 2.0, USB 3.x và USB4

b. Một số kiến thức cơ bản về giao thức usb

❖ Kiến trúc hệ thống USB

Một hệ thống USB được mô tả bởi 3 định nghĩa:

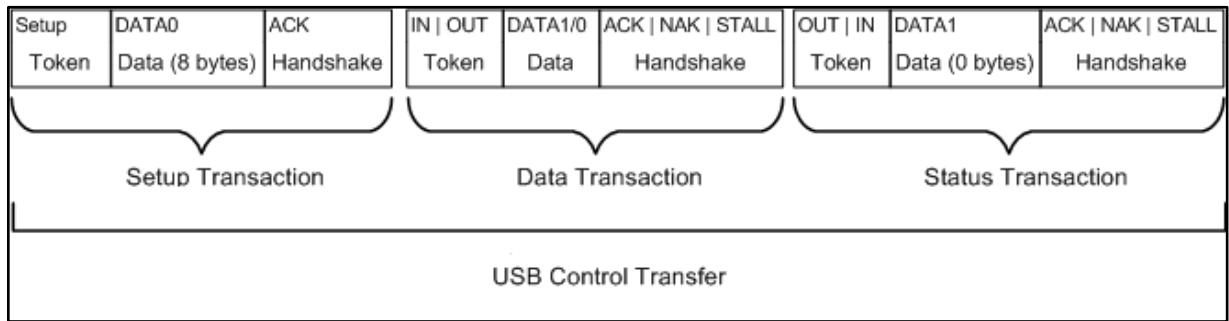
- USB Host: Chỉ có 1 Host duy nhất trong hệ thống (thông thường sẽ là máy tính cá nhân hoặc máy tính nhúng). Chúng điều khiển mọi luồng dữ liệu và cách thức giao tiếp trong hệ thống.
- USB Device: Trong một hệ thống có thể có nhiều device chúng có thể đóng nhiều vai trò như: Thiết bị đầu cuối tương tác với người dùng (HID), lớp giao tiếp (CDC), lưu trữ dữ liệu (MSP), Mở rộng cổng kết nối (HUB)
- USB Connect: Được hiểu là kiểu kết nối để các thiết bị USB có thể giao tiếp được với Host.

Kiến trúc bus: Về mặt vật lý, kiến trúc Bus USB là một tầng sao, với Host là trung tâm. Mỗi tia là 1 kết nối giữa Host với Hub, Host với Device hoặc Device với Hub. Với 7bit địa chỉ, một Host có thể quản lý 127 thiết bị trong mạng lưới của nó.

❖ Cấu trúc điều khiển truyền nhận

Để tất cả các thiết bị trong cùng hệ thống hiểu nhau chúng phải tuân thủ theo quy tắc điều khiển truyền nhận (Control Transfer).

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT



Hình 2.11. Một frame của USB control transfer

Một gói Control Transfer được chia thành 3 Transaction:

- Setup Transaction: Thiết lập
- Data Transaction: Dữ liệu
- Status Transaction: Trạng thái

Mỗi Transaction lại chia thành các gói tin:

- Setup hoặc IN/OU: Gói thông báo, cho biết rằng các data phía sau dùng để làm gì, được xác định rõ bởi các Mã Token tương ứng với chức năng đó
- DATA: Gói dữ liệu xác định dữ liệu truyền của transaction
- ACK: Gói bắt tay, thông báo dữ liệu truyền thành công hay thất bại

c. Giao thức USB CDC

❖ Định nghĩa USB CDC

USB communications device class hay được hiểu là Lớp thiết bị giao tiếp thông qua USB. Nghe có vẻ hơi tối nghĩa nhưng bạn có thể hiểu rằng nó được sử dụng để các thiết bị giao tiếp với nhau chứ không tương tác với con người như HID class.

Các thiết bị thuộc lớp này cũng được triển khai trong các hệ thống nhúng như điện thoại di động để điện thoại có thể được sử dụng làm modem, fax hoặc cổng mạng. Các giao diện dữ liệu thường được sử dụng để thực hiện truyền dữ liệu hàng loạt.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

❖ Giao thức USB CDC trên STM32

Trong STM32F103C8T6 chỉ hỗ trợ giao thức USB kiểu Device, thế nên chúng tôi sẽ sử dụng như một thiết bị để truyền nhận dữ liệu giữa nó và máy tính. Cách thức giao tiếp đã được chuẩn hóa trong thư viện USB.

2.3. Hệ điều hành FreeRTOS (sử dụng CMSIS_OS API V2)

2.3.1. Quản lý bộ nhớ trong FreeRTOS

Nhân RTOS cần RAM mỗi khi task (tác vụ), queue (hàng đợi), mutex, software timer (bộ định thời tạo bởi phần mềm), semaphore hoặc event group (nhóm sự kiện) được tạo và khi cấp phát động các biến. FreeRTOS heap khác với heap hệ thống được xác định ở cấp trình biên dịch. RAM có thể được tự động phân bổ động từ RTOS trong các chức năng tạo đối tượng API của RTOS hoặc nó có thể được cung cấp bởi người viết ứng dụng.

Nếu các đối tượng RTOS được tạo động thì các hàm malloc () và free () của thư viện C chuẩn đối khi có thể được sử dụng cho mục đích này, nhưng

- chúng không phải lúc nào cũng có sẵn trên các hệ thống nhúng,
- chúng chiếm không gian mã có giá trị,
- chúng không an toàn
- chúng không mang tính xác định (lượng thời gian thực hiện hàm sẽ khác với lệnh gọi)

Một hệ thống nhúng/thời gian thực có thể có các yêu cầu về RAM và thời gian rất khác nhau - vì vậy, một thuật toán phân bổ RAM duy nhất sẽ chỉ thích hợp cho một tập hợp con các ứng dụng.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Để giải quyết vấn đề này, FreeRTOS giữ API cấp phát bộ nhớ trong lớp di động của nó. Lớp di động nằm bên ngoài các tệp nguồn triển khai chức năng RTOS cốt lõi, cho phép cung cấp triển khai ứng dụng cụ thể phù hợp với hệ thống thời gian thực đang được phát triển.

a. Quản lý task trong bộ nhớ heap

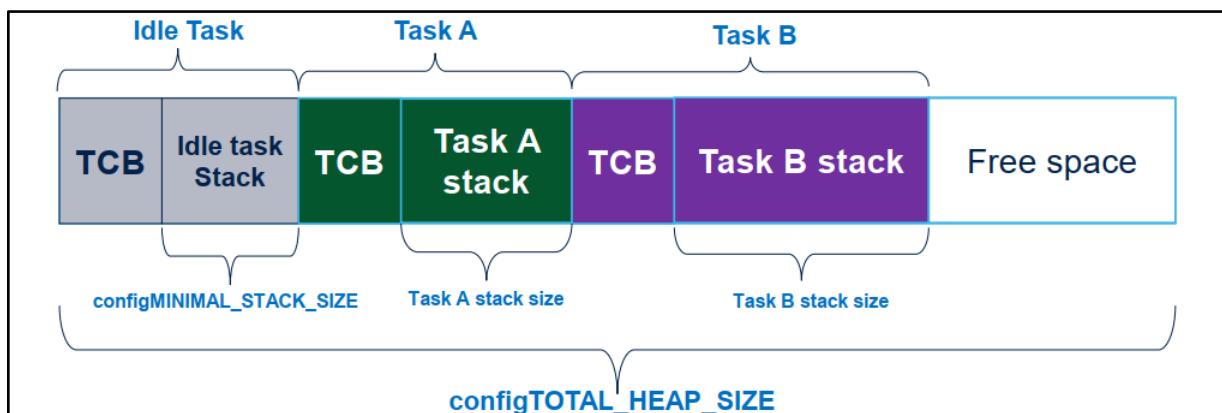
Mỗi tác vụ (task) đã tạo (bao gồm cả task nhàn rỗi) yêu cầu khối điều khiển tác vụ (TCB) và một ngăn xếp được phân bổ trong heap.

Kích thước TCB tính bằng byte phụ thuộc vào các tùy chọn được bật trong FreeRTOSConfig.h.

Kích thước ngăn xếp tác vụ được chuyển làm đối số khi tạo tại tác vụ. Kích thước ngăn xếp tác vụ được xác định bằng từ 32 bit không tính bằng byte.

FreeRTOS yêu cầu phân bổ trong heap cho mỗi tác vụ:

$$\text{số byte} = \text{TCB_size} + (4 \times \text{kích thước ngăn xếp tác vụ})$$



Hình 2.12. Quản lý dung lượng các task trong bộ nhớ heap

b. Quản lý semaphore, queue, mutex trong bộ nhớ heap

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Mỗi semaphore được khai báo bởi ứng dụng người dùng yêu cầu 88 byte được cấp phát trong heap.

Mỗi mutex được khai báo bởi ứng dụng người dùng yêu cầu 88 byte được phân bổ trong heap (+8 byte trong TCB của mỗi tác vụ).

FreeRTOS yêu cầu phân bổ trong heap cho mỗi hàng đợi tin nhắn (khối điều khiển và vùng lưu trữ hàng đợi):

- số byte = 76 + queue_storage_area.
- queue_storage_area (tính bằng byte) = (kích thước phần tử * nb_elements) + 16

c. Quản lý software timer trong bộ nhớ heap

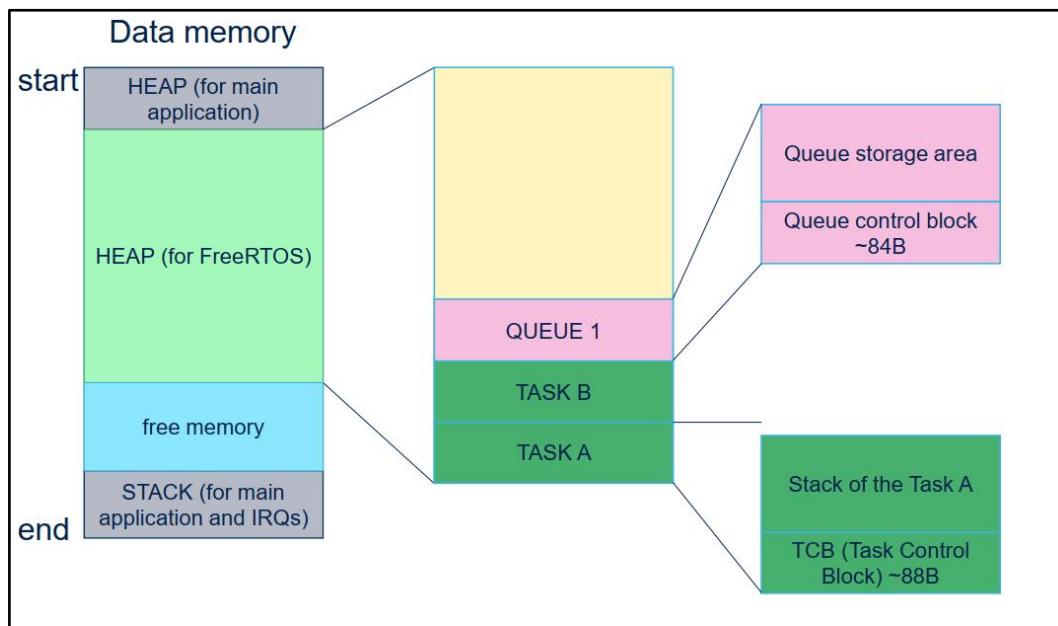
Khi software timer được bật (đã bật configUSE_TIMERS), bộ lập lịch tự động tạo tác vụ bộ định thời (daemon) khi khởi động. Tác vụ bộ định thời được sử dụng để kiểm soát và giám sát (nội bộ) tất cả bộ hẹn giờ mà người dùng sẽ tạo. Các tham số tác vụ bộ định thời được đặt thông qua các định nghĩa sau:

Bộ lập lịch cũng tự động tạo ra một hàng đợi tin nhắn được sử dụng để gửi các lệnh đến tác vụ dịch vụ bộ định thời (khởi động bộ định thời, dừng bộ định thời...)

2.3.2. Cấp phát bộ nhớ trong FreeRTOS

FreeRTOS bao gồm năm loại triển khai cấp phát bộ nhớ mẫu. Mỗi triển khai đã cung cấp được chứa trong một tệp nguồn riêng biệt. Chính xác một trong các tệp nguồn này nên được đưa vào dự án tại một thời điểm.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT



Hình 2.13. Phân vùng bộ nhớ heap cho hệ điều hành

Sau đây là 5 loại triển khai cấp phát bộ nhớ:

- heap_1 - đơn giản nhất, không cho phép giải phóng bộ nhớ.
- heap_2 - cho phép giải phóng bộ nhớ, nhưng không kết hợp các khối trống liền kề.
- heap_3 - chỉ cần kết thúc chuẩn malloc() và free() để đảm bảo an toàn cho luồng.
- heap_4 - liên kết các khối tự do liền kề để tránh phân mảnh. Bao gồm tùy chọn vị trí địa chỉ tuyệt đối.
- heap_5 - theo heap_4, với khả năng mở rộng heap trên nhiều vùng bộ nhớ không liền kề.

Với độ phổ biến cùng với những ưu điểm của mình, heap_4 là loại khai triển được sử dụng phổ biến nhất được chọn trong các ứng dụng hệ thống nhúng sử dụng FreeRTOS nói chung và đề tài này nói riêng.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

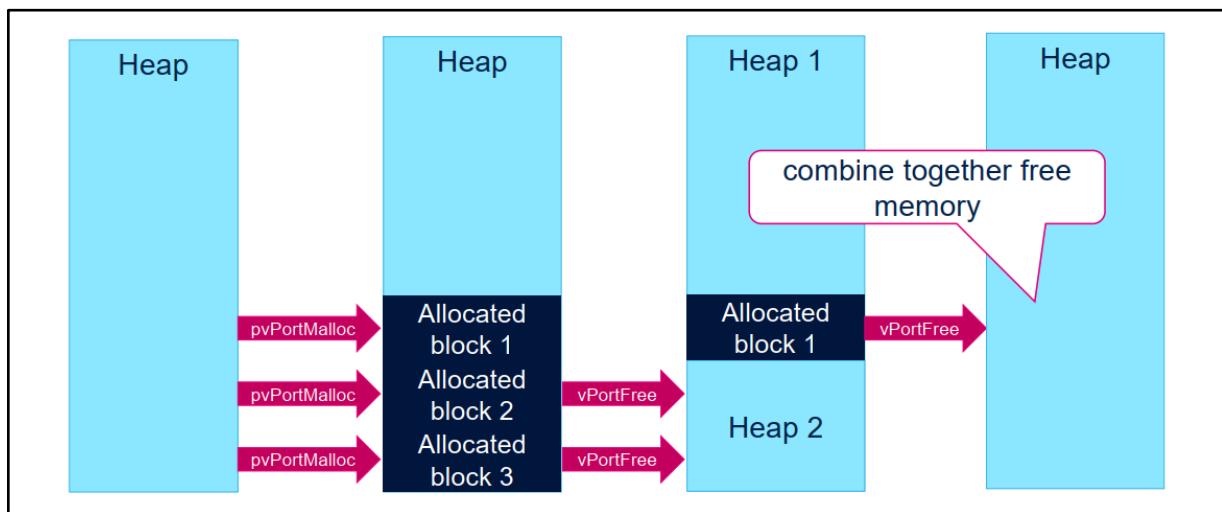
❖ heap_4:

Có thể được sử dụng ngay cả khi ứng dụng liên tục xóa các tác vụ, hàng đợi, semaphores, mutexes, v.v.

Khả năng triển khai heap_2 ít hơn nhiều so với việc triển khai heap_2 dẫn đến không gian heap bị phân mảnh thành nhiều khối nhỏ - ngay cả khi bộ nhớ được cấp phát và giải phóng có kích thước ngẫu nhiên.

Không phải là xác định - nhưng hiệu quả hơn nhiều so với hầu hết các triển khai malloc thư viện C tiêu chuẩn.

heap_4.c đặc biệt hữu ích cho các ứng dụng muốn sử dụng lược đồ phân bổ bộ nhớ lỏp di động trực tiếp trong mã ứng dụng.



Hình 2.14. Mô tả sự liên kết khi các khối bộ nhớ được giải phóng trong heap_4

Như hình trên, chúng ta có thể thấy với 1 heap liền mạch khi chúng chúng ta cấp phát bộ nhớ cho 3 khối và giải phóng bộ nhớ 2 khối bên dưới. Lúc này, heap sẽ bị chia ra thành 2 vùng bộ nhớ không liền mạch (khối 2 và khối 3 liền kề thì khi giải phóng sẽ mang lại 1 heap với bộ nhớ trống liền mạch gồm dung lượng của 2 khối đã giải phóng).

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Cuối cùng khi 1 khối còn lại được giải phóng thì bộ nhớ heap trở về thành 1 khối bộ nhớ liền mạch như ban đầu.

Mảng bộ nhớ được heap_4 sử dụng được khai báo trong tệp heap_4.c và địa chỉ bắt đầu của nó được trình liên kết cấu hình tự động.

2.3.3. Bộ lập lịch (Scheduler)

Bộ lập lịch là một thuật toán xác định tác vụ nào sẽ thực thi.

- Điểm chung giữa các bộ lập lịch là chúng phân biệt giữa các tác vụ sẵn sàng được thực hiện (ở trạng thái READY) và những tác vụ đang bị chặn vì bất kỳ lý do gì (bị trì hoãn, đang chờ đợi điều gì đó)
- Sự khác biệt chính giữa các bộ lập lịch là cách chúng phân phối thời gian CPU giữa các tác vụ ở trạng thái READY.

Trong FreeRTOS, thuật toán lập lịch vòng lặp được thực hiện:

- Round-robin có thể được sử dụng với đa nhiệm phủ đầu hoặc hợp tác.
- Nó hoạt động tốt nếu thời gian phản hồi không phải là vấn đề hoặc tất cả các nhiệm vụ có cùng mức độ ưu tiên.
- Quyền sở hữu CPU thay đổi định kỳ sau thời gian thực thi được xác định trước được gọi là timeslice.

a. Chế độ ưu tiên (Preemptive)

Đa nhiệm ưu tiên (mặc định trong FreeRTOS)

- Các tác vụ có cùng mức ưu tiên chia sẻ thời gian CPU
- Bối cảnh được chuyển đổi khi
 - Phần thời gian đã trôi qua
 - Tác vụ với mức độ ưu tiên cao hơn đã đến

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

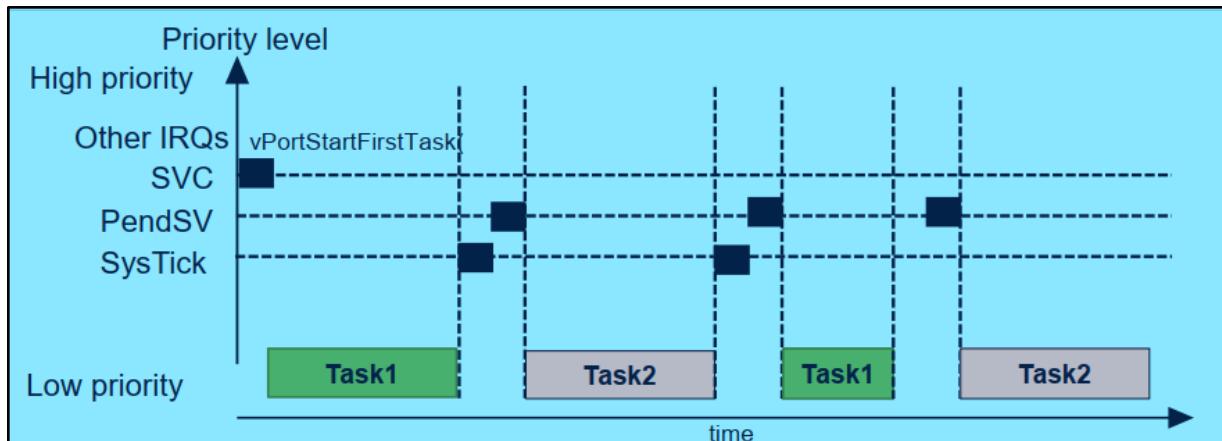
- Tác vụ chuyển sang trạng thái BLOCKED
- Tác vụ chuyển sang trạng thái READY

b. Hợp tác với quyền ưu tiên bởi ngắt đa nhiệm

- IRQ được sử dụng để kích hoạt chuyển đổi ngữ cảnh bằng cách gọi trực tiếp hàm yield hoặc sử dụng portEND_SWITCHING_ISR hoặc portYIELD_FROM_ISR
- Hệ thống dự phòng không có thời gian

2.3.4. Ngắt trong hệ điều hành FreeRTOS

❖ Vector ngắt



Hình 2.15. Mô tả các ngắt trong hệ điều hành

❖ Ngắt PendSV

- Được sử dụng để chuyển đổi nhiệm vụ trước khi tỷ lệ đánh dấu
- Mức độ ưu tiên ngắt NVIC thấp nhất
- Không được kích hoạt bởi bất kỳ thiết bị ngoại vi nào

❖ Ngắt SVC

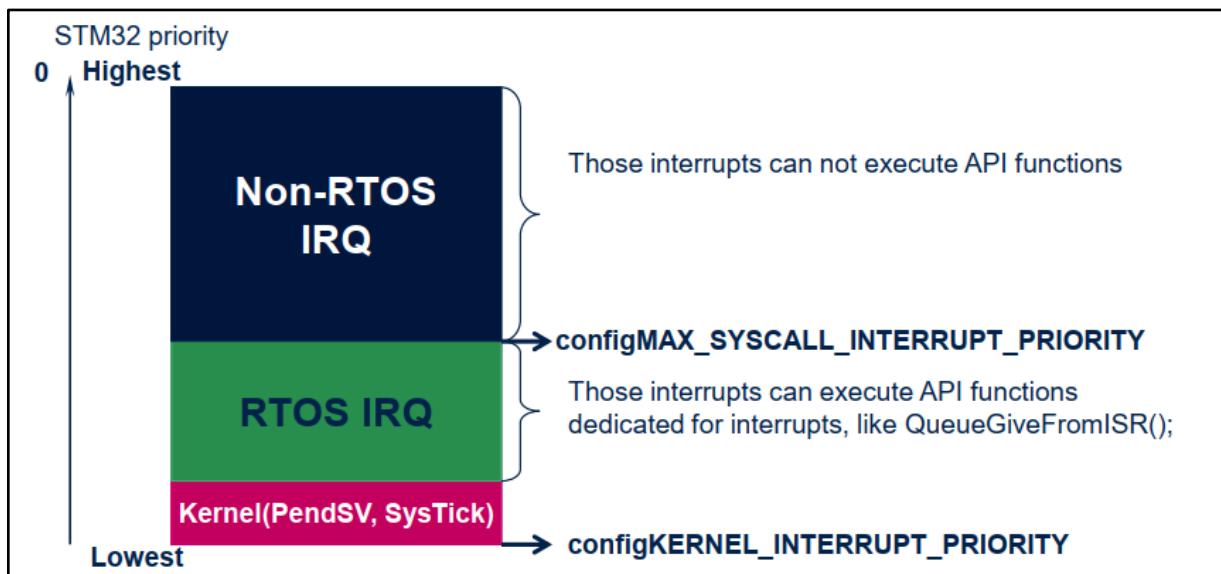
- Gián đoạn tăng lên bởi lệnh SVC

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

- Lệnh gọi SVC 0 chỉ được sử dụng một lần, để khởi động bộ lập lịch (trong vPortStartFirstTask() được sử dụng để khởi động nhân)

❖ Bộ định thời SysTick

- Mức độ ưu tiên ngắn NVIC thấp nhất
- Được sử dụng để chuyển đổi tác vụ trên cơ sở thời gian thường xuyên
- Đặt PendSV nếu cần chuyển đổi ngữ cảnh



Hình 2.16. Phân loại các ngắt trong chương trình sử dụng FreeRTOS

2.3.5. Tác vụ (task)

a. Định nghĩa Task

- Nó là một hàm C
- Nó phải được chạy trong vòng lặp vô hạn
- Nó có thể được sử dụng để tạo ra bất kỳ số lượng tác vụ nào
- Nó có một phần riêng của stack (mỗi trường hợp) và mức độ ưu tiên
- Có 4 trạng thái cấu trúc của tác vụ

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Một tác vụ bao gồm ba phần:

- Mã chương trình (ROM)
- Một ngăn xếp nằm trong vùng RAM có thể truy cập bởi con trỏ ngăn xếp
- TCB - khôi điều khiển tác vụ

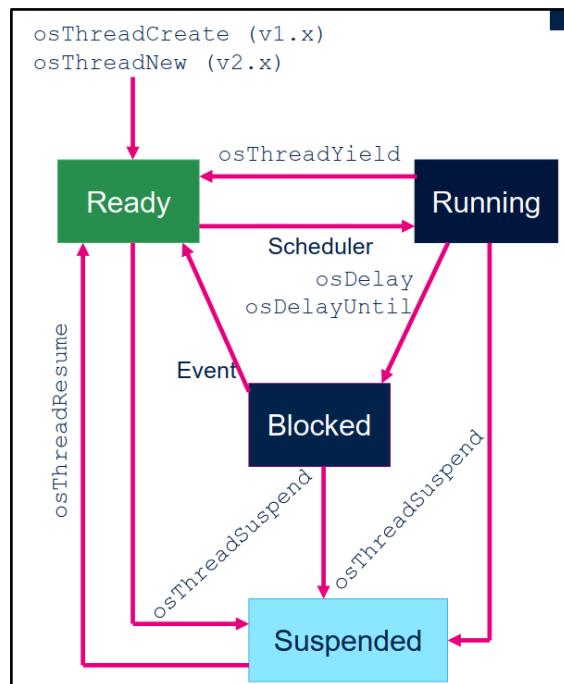
b. Trạng thái của tác vụ

READY: Tác vụ đã sẵn sàng để thực thi nhưng hiện không thực thi vì một tác vụ khác có mức độ ưu tiên bằng hoặc cao hơn đang chạy

RUNNING: Tác vụ đang thực sự chạy (chỉ 1 tác vụ có thể ở trạng thái này tại 1 thời điểm)

BLOCKED: Tác vụ đang chờ đợi 1 sự kiện tạm thời hay 1 sự kiện bên ngoài

SUSPENDED: Tác vụ không khả dụng, nhưng vẫn được lưu trong bộ nhớ



Hình 2.17. Tương tác giữa các trạng thái của tác vụ với nhau trong FreeRTOS

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

c. Ưu tiên của các tác vụ

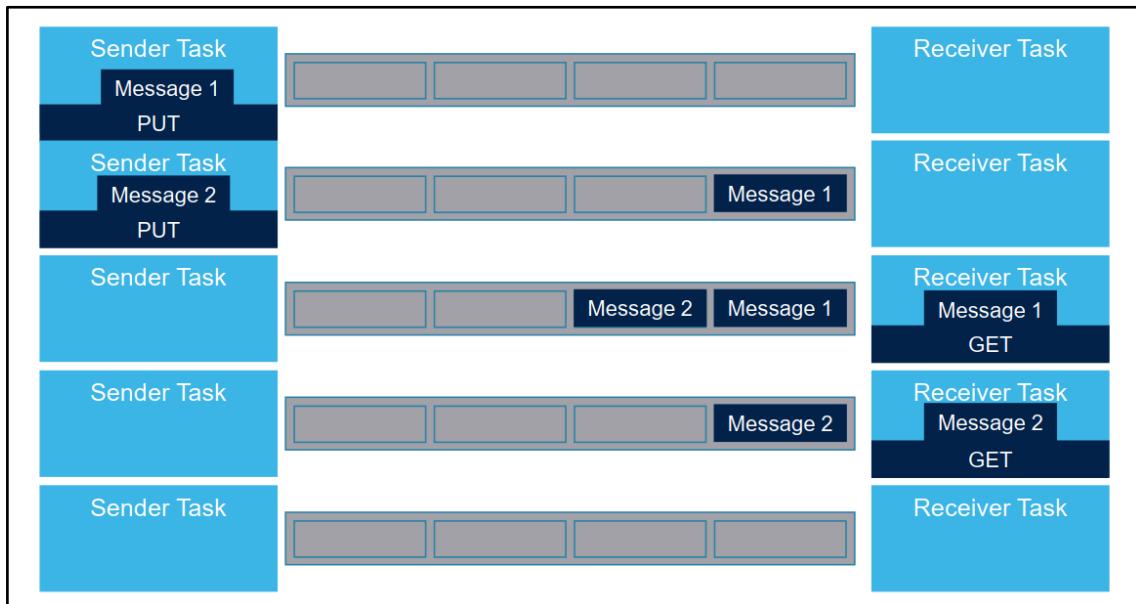
Mỗi tác vụ đã chỉ định một mức độ ưu tiên. Thứ tự thực hiện các tác vụ phụ thuộc vào mức độ ưu tiên này. Bộ lập lịch kích hoạt tác vụ có mức ưu tiên cao nhất trong tất cả các tác vụ ở trạng thái READY. Tác vụ có mức độ ưu tiên cao hơn có thể đặt trước tác vụ đang chạy. Các ưu tiên tác vụ có thể được thay đổi trong quá trình làm việc của ứng dụng. Số càng thấp mức độ ưu tiên càng thấp

2.3.6. Giao tiếp giữa các tác vụ

a. Queue

- Hàng đợi là đường để truyền dữ liệu giữa các tác vụ trong RTOS.
- Theo mặc định, hàng đợi hoạt động theo FIFO (First In - First Out).
- Tất cả dữ liệu gửi theo hàng đợi phải cùng loại, được khai báo trong giai đoạn tạo hàng đợi.
- Nó có thể là biến hoặc cấu trúc đơn giản.
- Các tác vụ có thể chặn việc gửi hoặc nhận dữ liệu trong hàng đợi với thời gian chờ hoặc vô hạn.
- Nếu nhiều tác vụ bị chặn khi chờ nhận hay gửi dữ liệu đến một hàng đợi thì chỉ tác vụ có mức ưu tiên cao nhất mới được bỏ chặn khi có dữ liệu hoặc không gian.
- Nếu cả hai tác vụ có mức độ ưu tiên như nhau, tác vụ đã chờ lâu nhất sẽ được bỏ chặn.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT



Hình 2.18. Mô tả hoạt động của Queue

b. Event flags/Event bits

Các bit sự kiện được sử dụng để cho biết một sự kiện đã xảy ra hay chưa. Các bit sự kiện thường được gọi là cờ sự kiện.

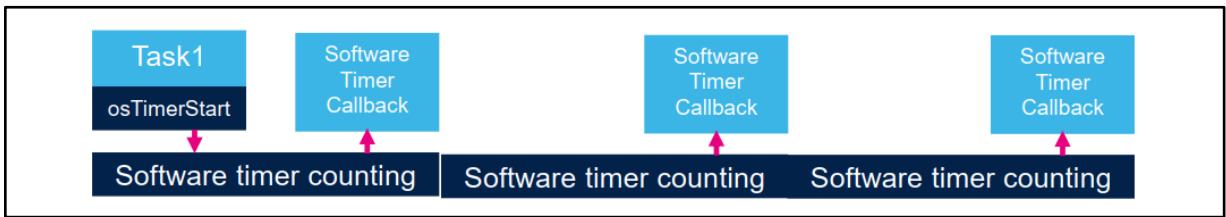
2.3.7. Bộ định thời tạo bởi phần mềm (Software timer)

Bộ định thời phần mềm là một trong những thành phần tiêu chuẩn của mọi RTOS. Nó cho phép thực hiện một cuộc gọi lại tại một thời điểm nhất định. Các hàm gọi lại bộ định thời thực thi trong ngũ cảnh của nhiệm vụ dịch vụ bộ định thời. Do đó, điều cần thiết là các chức năng gọi lại của bộ hẹn giờ không bao giờ có găng chẵn. Nó không chính xác, nhằm xử lý các hành động định kỳ và tạo độ trễ

Hai loại bộ định thời phần mềm có sẵn:

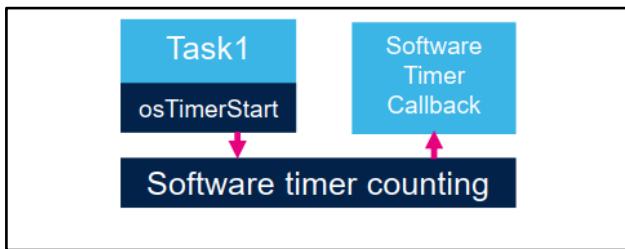
- Periodic (thực hiện lệnh gọi lại của nó theo định kỳ với chức năng tải tự động)

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT



Hình 2.19. Hoạt động của chế độ định kỳ của bộ định thời

- One Pulse (thực hiện lệnh gọi lại của nó chỉ một lần với tùy chọn kích hoạt lại thủ công)



Hình 2.20. Hoạt động của chế độ 1 xung của bộ định thời

2.4. MÔ HÌNH HỌC SÂU SỬ DỤNG YOLOV5.

2.4.1. CNN (mạng nơ-ron tích chập)

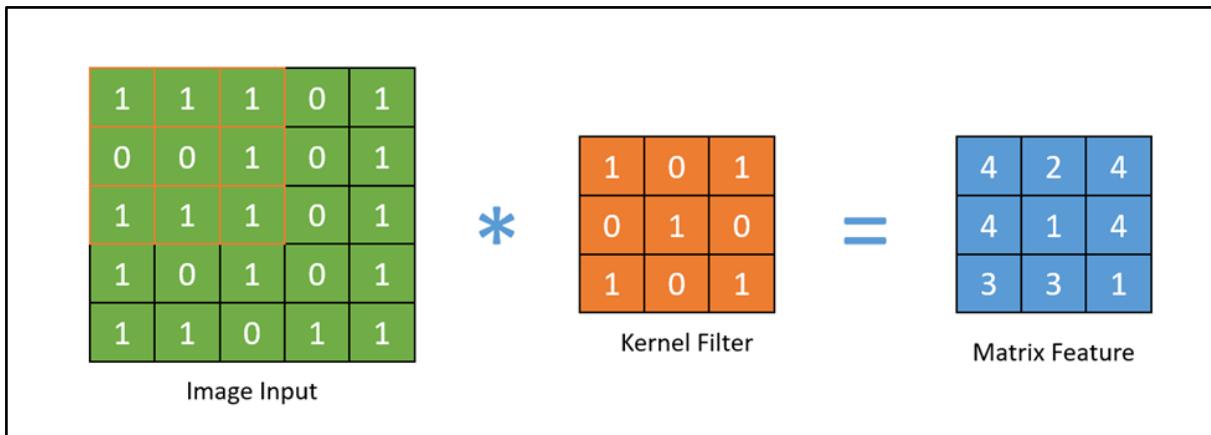
CNN là tên viết tắt của Convolution Neural Network (mạng nơ-ron tích chập) là một trong những mô hình Deep Learning phổ biến hiện nay. CNN cho phép chúng ta ứng dụng nhiều vào các bài toán đòi hỏi độ phức tạp cao và yêu cầu độ chính xác lớn. CNN được sử dụng đa dạng trong các bài toán classification, object detection, segmentaion,...

Các lớp cơ bản trong mạng CNN bao gồm: lớp tích chập, lớp pooling, hàm kích hoạt.

❖ Lớp tích chập

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Từ ảnh đầu vào lớp đầu tiên của CNN là lớp tích chập, lớp này có nhiệm vụ trích xuất các đặc trưng của ảnh đầu vào. CNN sử dụng các kernel ma trận, chứa các tham số được huấn luyện trong quá trình học, kernel này nhân tích chập với ảnh đầu vào để tạo ra một ma trận đặc trưng có kích thước bé hơn.



Hình 2.21. Hình nhân tích chập

Để dễ hình dung việc tính tích chập dựa vào hình XX lần lượt nhân tích chập và trượt Kernel Filter từ trái qua phải sẽ tính được ma trận đặc trưng.

Các thông số liên quan:

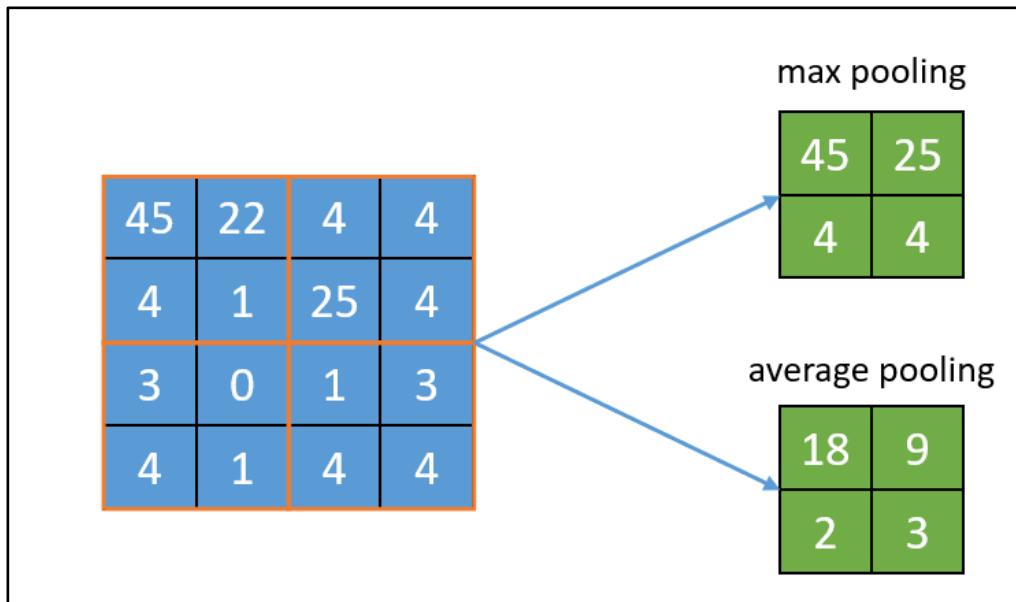
- Stride: khi dịch chuyển kernel filter theo stride pixel từ trái sang phải. Ở ví dụ trên stride là 1.
- Padding: đối với những yêu cầu bài toán cần trích xuất đặc trưng ở những viền cạnh của ảnh đầu vào thì sẽ chọn giá trị padding khác 0.

❖ Lớp pooling

Khi đầu ra của feature map quá lớn lớp pooling sẽ được xếp vào sau đó trước khi đi qua những lớp tích chập tiếp theo, nhằm giảm các tham số trong việc tính toán ở các lớp sau mà vẫn giữ lại các đặc trưng của các lớp trước. Có hai loại pooling được sử dụng

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

rộng rãi với kích thước 2x2 hoặc 4x4 nếu ảnh đầu vào lớn là: max pooling và average pooling.



Hình 2.22. Ví dụ pooling layer

Như tên gọi:

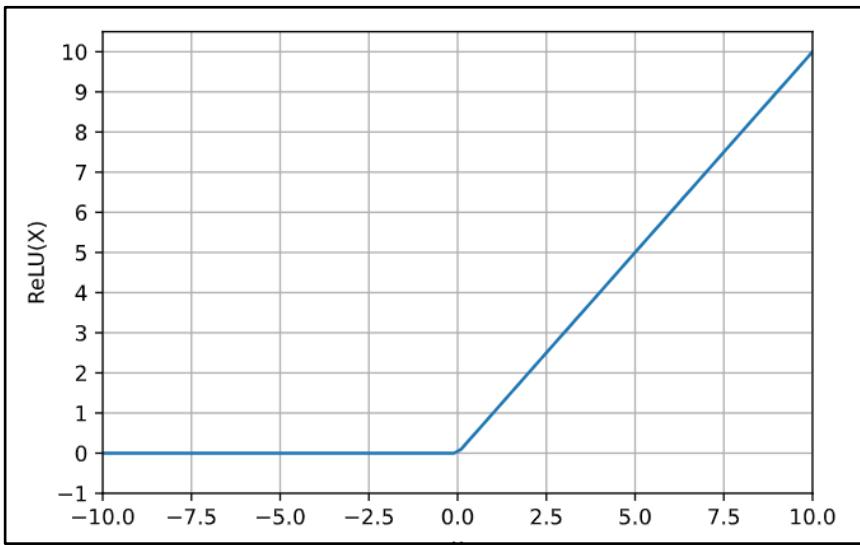
- Max pooling: lấy giá trị lớn nhất tại cửa sổ pooling đang trượt đến.
- Average pooling: lấy giá trị trung bình tại cửa sổ pooling đang trượt đến.

❖ Hàm kích hoạt

Trong mạng CNN hàm kích hoạt được dùng nhiều là hàm phi tuyến ReLU (Rectified Linear Unit) với đầu ra được tính là:

$$f(x) = \max(0, x)$$

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT



Hình 2.23. Đồ thị hàm ReLU

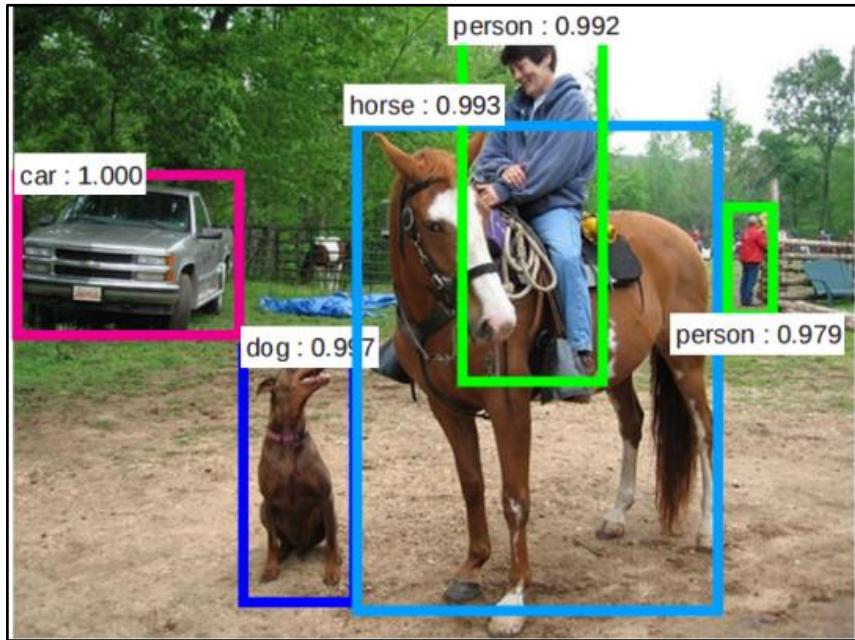
Khi sử dụng hàm kích hoạt ReLU thì các đặc trưng âm được loại bỏ và chỉ giữ lại phần dương. Ngoài ReLU còn có một số hàm phi tuyến khác là tanh, sigmoid cũng có thể thay thế, tuy nhiên ReLU được dùng nhiều hơn do hiệu suất tốt hơn.

Nhược điểm của hàm ReLU là khi các đặc trưng nhỏ hơn 0, qua ReLU activation sẽ thành 0, hiện tượng đáy gọi là “Dying ReLU“. Nếu các node bị chuyển thành 0 thì sẽ không có ý nghĩa với bước linear activation ở lớp tiếp theo và các hệ số tương ứng từ node đáy cũng không được cập nhật với gradient descent, do đó hàm Leaky ReLU ra đời khắc phục nhược điểm này.

2.4.2. Bài toán object detection

Object detection là bài toán tìm kiếm và phân loại số lượng đối tượng trên một hình ảnh, có thể có một hoặc nhiều đối tượng trên cùng một ảnh. Với đầu vào là ảnh màu, đầu ra là tọa độ của các đối tượng được phát hiện trên ảnh và tỉ lệ phần trăm chẵn chẵn của mỗi đối tượng.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT



Hình 2.24. Ảnh minh họa đầu ra của bài toán object detection

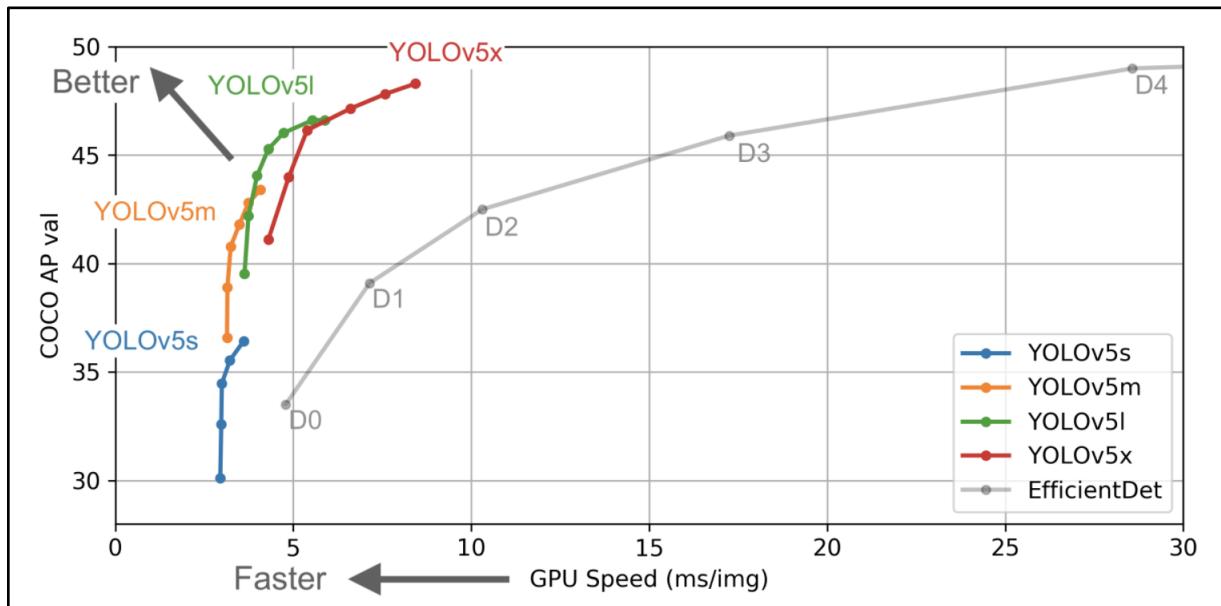
2.4.3. YOLOv5

YOLOv5 được công ty Ultralytics cho ra mắt vào năm 2020. Đây là mô hình đầu tiên của YOLO được triển khai trên Pytorch framework thay vì Darknet framework so với các phiên bản trước đó, dó đó nó nhẹ và dễ sử dụng hơn so. YOLOv5 dựa trên YOLO1-YOLO4 được cải tiến liên tục và đạt được hiệu suất hàng đầu trên hai bộ dữ liệu phát hiện đối tượng chính thức: Pascal VOC và CoCo. Tuy nhiên cũng còn nhiều tranh cãi xung quanh YOLOv5 do chưa có một bài báo nào được xuất bản nói chi tiết về nó.

YOLOv5 dựa trên smart Convolutional Neural Network (CNN) để phát hiện đối tượng trong thời gian thực. Thuật toán này chia hình ảnh thành các vùng và tính toán các bouding box và xác suất cho từng vùng. Thuật toán chỉ cần một lần truyền chuyển tiếp qua mạng nơ-ron để đưa ra dự đoán, vì vậy nó còn được gọi là "only looks once". Sau

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

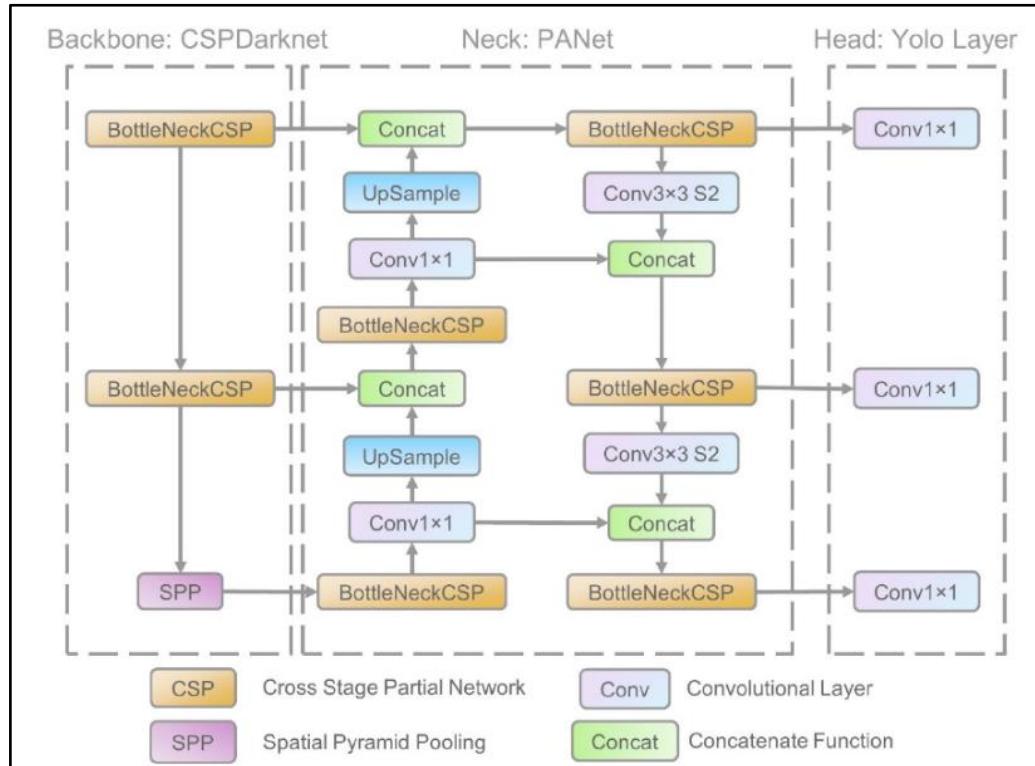
đó, nó xuất ra các đối tượng đã biết cùng với các hộp giới hạn sau đó sử dụng non-max suppression để loại bỏ những hộp trùng lặp từ đó đưa ra kết quả cuối cùng.



Hình 2.25. Hiệu suất của các phiên bản YOLOv5 trên tập dữ liệu COCO

YOLOv5 là một mô hình có hiệu suất cao AP lớn và thời gian đáp ứng nhanh, điều này làm YOLOv5 nổi trội nhiều hơn các mô hình khác.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT



Hình 2.26. Cấu trúc của mô hình yolov5

YOLOv5 đã kết hợp partial net-work CPSNet vào Darknet, tạo CSPDarknet làm backbone cho nó. CSPNet giải quyết các vấn đề về thông tin gradient lặp đi lặp lại trong backbone với quy mô lớn và tích hợp các thay đổi gradient vào feature map, do đó làm giảm các tham số và FLOPS (floating-point operations per second) của mô hình, điều này không chỉ đảm bảo tốc độ và độ chính xác mà còn giảm kích thước mô hình.

YOLOv5 áp dụng mạng Path Aggregation Network (PANet) làm neck để thúc đẩy luồng thông tin. PANet áp dụng cấu trúc Feature Pyramid Network (FPN) tính năng mới giúp cải thiện việc lan truyền các đặc trưng ở mức thấp. PANet cải thiện việc sử dụng các accurate localization các lớp thấp hơn, điều này có thể nâng cao độ chính xác vị trí của đối tượng.

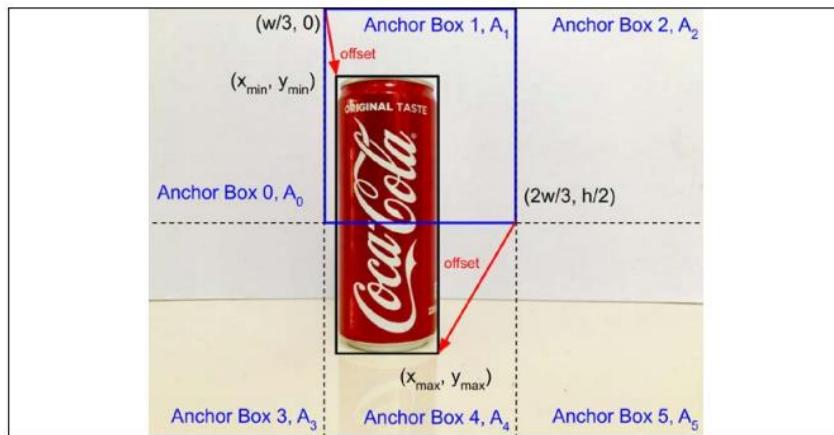
2.4.4. Default box

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Để phát hiện đối tượng trong một bức ảnh, mô hình cần phải dự đoán được vùng giới hạn đối tượng trong ảnh đó (bounding box) và đối tượng bên trong ảnh đó là gì. Có nhiều cách khác nhau xác định tọa độ $(xmin, ymin)$ và $(xmax, ymax)$ của bounding box. (*Định nghĩa $(xmin, ymin)$ và $(xmax, ymax)$ tương ứng với tọa độ pixel góc trên bên trái và pixel góc dưới bên phải của bounding box*).

Một bức ảnh ban đầu với chiều rộng là w chiều cao là h . Offset chính là giá trị sai lệch giữa bounding box phù hợp với vật thể cần phát hiện so với bounding box ta tìm ra. Do đó offset là $(xmin, ymin)$ và $(xmax-w, ymax-h)$. Nếu vật thể nằm vừa gọn trong khung ảnh thì bounding box cần tìm chính là khung ảnh. Offset lúc này là $(0,0)$ và $(0,0)$.

Yêu cầu đặt ra là làm sao tạo ra các bounding box phù hợp với vật thể cần phát hiện. Để làm điều đó, người ta chia ảnh thành nhiều vùng và ta gọi các vùng này là anchor box với mong muốn trong những anchor box được chia ra sẽ có một anchor box thích hợp làm bounding box của vật thể.



Hình 2.27. Chia ảnh thành nhiều vùng anchor box

Để đáp ứng yêu cầu tìm được anchor box thích hợp, sẽ chia ảnh thành nhiều vùng như hình trên. Để giảm offset, tiếp tục chia nhỏ ảnh thành nhiều anchor box hơn. Do đó, ý tưởng chính là trong quá trình tạo ra các vùng khác nhau nhằm đa dạng kích thước.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

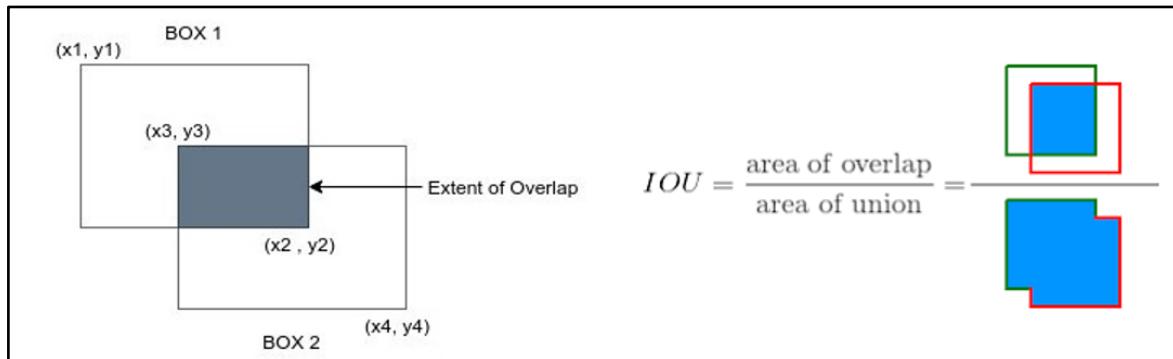
Một kích thước anchor box tối ưu gần nhất so với bounding box của đối tượng sẽ xuất hiện.

2.4.5. Non-max suppression

Bài toán đặt ra tiếp là trong nhiều default box ta tạo ra sẽ có nhiều default box chứa cùng một đối tượng vậy ta sẽ chọn default box nào để làm bounding box cho đối tượng. Chính vì vấn đề này non-maximum suppression sinh ra để loại bỏ đi các default box dư thừa của cùng một đối tượng trong ảnh. Thuật toán non-maximum suppression dựa trên một chỉ số gọi là intersection over union (IoU).

Giả sử ta có 2 boxes với thông tin:

- Box 1 có tọa độ $(x_1, y_1), (x_2, y_2)$
- Box 2 có tọa độ $(x_1, y_1), (x_2, y_2)$



Hình 2.28. Ảnh minh họa cách tính IoU

Khi đó IoU được tính theo công thức:

$$IoU = \frac{|Box 1 \cap Box 2|}{|Box 1 \cup Box 2|} = \frac{|Box 1 \cap Box 2|}{|Box 1| + |Box 2| - |Box 1 \cap Box 2|}$$

Từ đó dựa trên giá trị ngưỡng của IoU mà loại bỏ những default box có IoU lớn hơn ngưỡng.

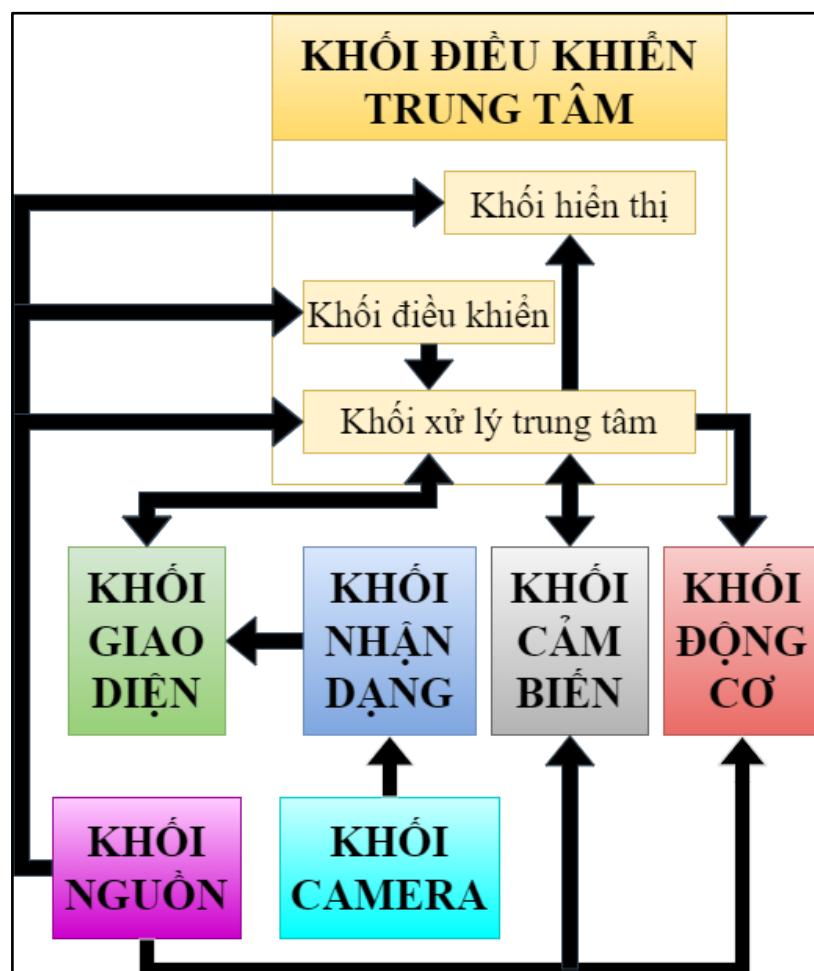
CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

Phần này chúng tôi tiến hành thiết kế sơ đồ khối toàn mạch, mỗi khối sẽ đưa ra mục tiêu thiết kế. Dựa vào mục tiêu này chúng tôi nêu ra những phương án lựa chọn, giải pháp tối ưu để tiến hành thiết kế. Từ đó dựa vào cơ sở lý thuyết đã trình bày ở phần trên để tính toán, căn chỉnh và hoàn thiện thiết kế.

3.1. SƠ ĐỒ KHỐI

3.1.1. Sơ đồ khối hệ thống



Hình 3.1. Sơ đồ khối hệ thống

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

❖ Nhiệm vụ các khối trong khói điều khiển trung tâm

- Khối xử lý trung tâm: đọc và xử lý dữ liệu từ khói tín hiệu cảm biến, nhận tín hiệu điều khiển từ khói điều khiển, xuất tín hiệu cho khói hiển thị, điều khiển ma trận khói cảm biến, giao tiếp với khói giao diện để truyền nhận dữ liệu.
- Khối điều khiển: được sử dụng để giao tiếp giữa thao tác của người dùng với hệ thống. Hỗ trợ người dùng thao tác gửi tín hiệu đến khói xử lý trung tâm.
- Khối hiển thị: hiển thị trạng thái khi có tín hiệu hoặc nhận tín hiệu phản hồi từ khói xử lý trung tâm.

❖ Chức năng các khói khác

- Khối nguồn: có chức năng cấp nguồn cho khói hiển thị, khói điều khiển, khói xử lý trung tâm, khói cảm biến và khói động cơ.
- Khối giao diện: chức năng là truyền và nhận dữ liệu nối tiếp từ khói nhận dạng và khói xử lý trung tâm. Là trung gian giữa bàn tay robot và người dùng với khả năng điều khiển và hiển thị các thông tin quan trọng liên quan.
- Khối nhận dạng: chức năng nhận dữ liệu camera dưới dạng ảnh và nhận dạng vật thể mà bàn tay thao tác với và truyền dữ liệu cho khói giao diện.
- Khối Camera: thu thập dữ liệu hình ảnh đưa vào cho khói nhận dạng để nhận dạng vật thể.
- Khối động cơ: là nơi nhận tín hiệu từ khói xử lý trung tâm và điều khiển các động cơ. Các động cơ sẽ điều khiển các khớp tay của robot.
- Khối cảm biến: nhận tín hiệu điều khiển của khói điều khiển trung tâm và xuất tín hiệu để có thể thu thập dữ liệu ở ma trận lực khi thao tác với vật thể và đưa vào cho khói xử lý trung tâm.

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

3.2. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

Tiếp theo chúng tôi nêu chức năng, phương án chọn lựa, thông số kĩ thuật, tính toán, thiết kế nguyên lý và giải thích cho các khối trong hệ thống. Bên cạnh đó, nếu các khối là phần mềm chúng tôi sẽ tập trung vào phần giao thức kết nối của khối đó với các khối khác.

3.2.1. Khối cảm biến

❖ Chức năng

Nhận tín hiệu điều khiển hàng của ma trận và xuất tín hiệu cảm biến của các cột trên hàng đó giúp khối xử lý trung tâm thu thập dữ liệu ở ma trận lực khi thao tác với vật thể.

❖ Phương án và lựa chọn

- Lựa chọn loại cảm biến và phương án thực hiện

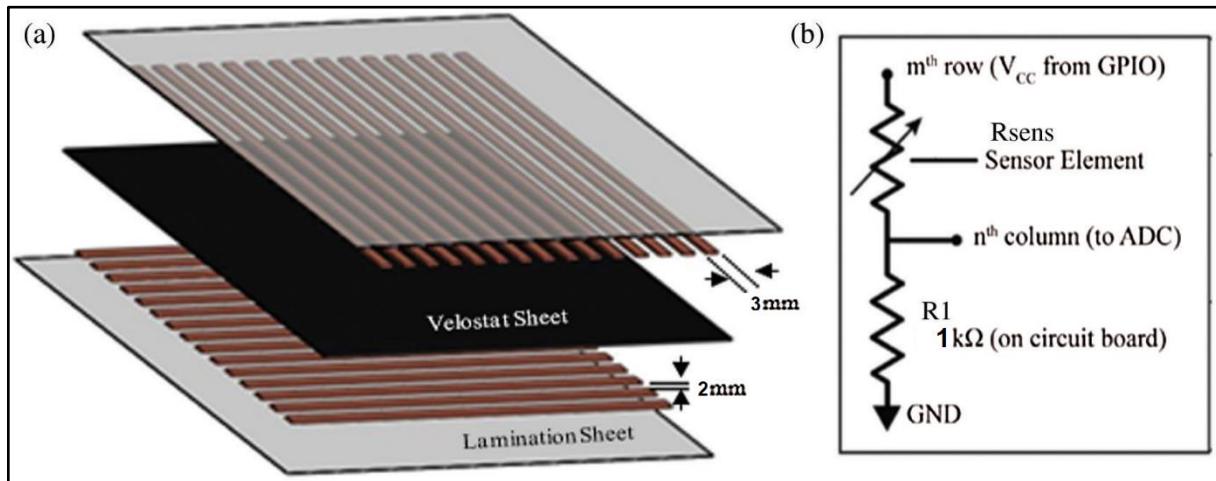
Hiện nay có rất nhiều công nghệ chế tạo cảm biến xúc giác. Như đã đề cập chi tiết ở phần trước, các công nghệ nổi bật để chế tạo cảm biến xúc giác có thể kể đến các loại như cảm biến áp điện trở (piezoresistive sensors), cảm biến điện dung (capacitive sensors), cảm biến áp điện (piezoelectric sensors), cảm biến hiệu ứng xung xuyên hầm lượng tử (quantum tunnel effect sensors), cảm biến quang (optical sensors), cảm biến dựa trên phép đo khí áp (sensors based on barometric measurements), cảm biến xúc giác đa phương thức (multi-modal tactile sensors), cảm biến xúc giác âm thanh từ cấu trúc (structure-borne sound tactile sensors).

Chúng tôi chọn vật liệu Velostat là phương án cảm biến piezoresistive với những ưu điểm nó mang lại và độ phổ biến của nó trong việc được sử dụng để chế tạo cảm biến xúc giác.Thêm vào đó loại cảm biến này còn đáp ứng đủ những yêu cầu đặt ra về khả năng phản hồi, độ bền theo thời gian, độ linh hoạt cao,...

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

Velostat là một vật liệu đóng gói được làm bằng lá polyme (polyolefin) ngâm tẩm với muội than để làm cho nó dẫn điện. Nó được sử dụng để bảo vệ các vật dụng hoặc thiết bị dễ bị hư hỏng do phóng tĩnh điện.

Các chất độn dẫn điện tập trung cao được phân bố đồng đều tạo nên một ma trận polyme liên kết chặt chẽ. Cấu trúc ma trận polyme này đảm bảo tính liên tục trong mạng dẫn và tạo thành một vật liệu đệm như một chất bán dẫn điện. Điện trở số lượng lớn là một trong những đặc tính điện quan trọng của tấm polyme dẫn điện. Điện trở số lượng lớn được thay đổi bằng cách áp dụng lực uốn hoặc áp suất, do phản ứng điện trở này mà nó được sử dụng để chế tạo cảm biến áp suất rẻ tiền, do đó sẽ thuận tiện cho việc làm cho cảm biến linh hoạt hơn. Tấm cảm biến áp suất được thiết kế với kích thước phù hợp với từng bộ phận của bàn tay. Độ dẫn điện giữa các cảm biến áp suất được đảm bảo bằng lá đồng vì nó có độ dẫn điện rất cao khi so sánh với lá nhôm. Lá nhôm có sẵn và rẻ tiền, trong khi lá đồng chiếm ưu thế về cơ sở hàn và ít hơn về điện trở suất. Do đó, lá đồng được chọn cho công việc được đề xuất. Các lá đồng được sắp xếp như vậy với khoảng cách 2mm.



Hình 3.2. (a): Thiết kế lắp ráp (b): Mạch điều khiển cảm biến

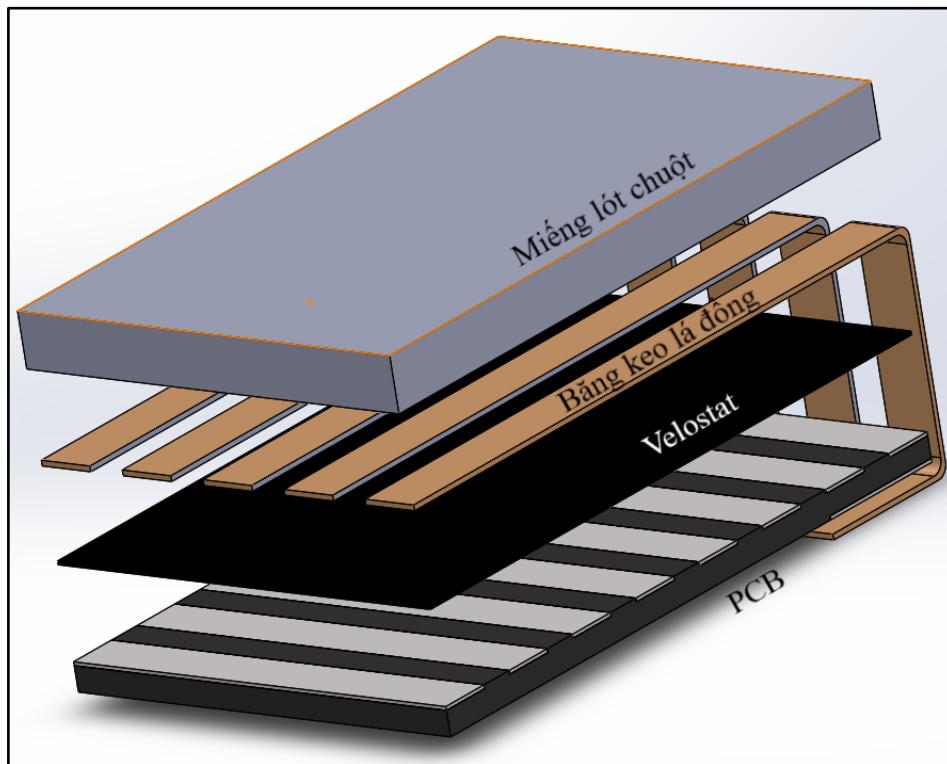
CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

Với yêu cầu cho mô hình đã được để ra ban đầu, chúng tôi quyết định phần cảm biến ở các ngón tay sẽ là những miếng hình chữ nhật với ma trận 8×5 .

Các vật liệu chính được chúng tôi sử dụng để thiết kế cảm biến:

- Lamination Sheet: sử dụng miếng lót chuột
- Velostat Sheet: là tấm cảm biến

Như vậy với sơ đồ thiết kế lắp ráp cảm biến như hình trên, một tấm cảm biến sẽ được kết nối hàng và cột như hình mạch điều khiển cảm biến. Phương pháp lấy dữ liệu cảm biến dựa vào số hàng và cột của cảm biến. Khi một hàng được bật lên thì tất cả dữ liệu các cột của hàng đó sẽ được vi điều khiển scan về. Cứ như vậy, các hàng được luân phiên bật cho đến hết thì lại quay về hàng đầu tiên.



Hình 3.3. Thiết kế lắp ráp cảm biến xúc giác của chúng tôi

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

Chúng tôi sử dụng chất liệu của miếng lót chuột làm vật liệu cho lớp ngoài cùng để tăng độ bám, tản lực và cải thiện độ êm cho bề mặt ngón tay. Ngoài ra, để tiện cho việc thi công về mặt thẩm mĩ và thời gian, chúng tôi thiết kế những PCB đơn giản thay thế lớp ngoài cùng và băng keo lá đồng tích hợp ở một mặt và ra các tiếp điểm cho các đường dẫn của hàng và cột của ma trận. Các đường băng keo lá đồng ở mặt sau còn lại sẽ được kéo dài đưa ra sau để hàn và các pad đồng ở mặt sau của PCB và sẽ ra chân hàng và cột của cảm biến ở mặt sau tấm PCB để có thể hàn dây dẫn vào.

- **Lựa chọn linh kiện**

Để mở rộng ngõ ra thành 44 để đưa vào 44 hàng của ma trận cảm biến có rất nhiều IC hỗ trợ mở rộng ngõ ra có thể kể đến như 74HC595, sử dụng I2C,... thường được sử dụng để mở rộng ngõ ra cho vi điều khiển. Với nhu cầu thiết kế chỉ cần bật 1 hàng của ma trận tại 1 thời điểm, chúng tôi chọn IC 74HC238 và dùng kĩ thuật ghép tầng ghép 6 IC lại với nhau để mở rộng đủ 44 ngõ ra. Đồng thời, chúng tôi có sử dụng 2 mạch NOT sử dụng điện trở và transistor để đáp ứng nhu cầu ngõ vào của IC này khi ghép tầng. Ngoài ra, chúng tôi còn bổ sung các diode vào để tránh dòng điện dội trong hệ thống cảm biến xúc giác đi qua các điểm có điện trở thấp và dội về. Diode được chúng tôi sử dụng là loại diode chỉnh lưu 1N5819 vì loại diode này phù hợp, nhỏ gọn và giá thành rẻ.

Để có tín hiệu từ khói cảm biến và xử lý, phân tích thì có rất nhiều phương án như hàng rào, domino, jack nối,... có thể được sử dụng để kết nối dây dẫn. Trong đó, chúng tôi lựa chọn phương án đơn giản, thô sơ và không mất chi phí mà mang lại hiệu quả là 1 loạt các lỗ trên mạch để có thể hàng dây dẫn trực tiếp vào.

- ❖ **Thông số kĩ thuật**

Tấm Velostat:

- **Độ lớn: 11" x 11" (280mm x 280mm)**

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

- Độ dày: 8 mil/0.2mm
- Trọng lượng: 18.66g
- Giới hạn nhiệt độ: -45°C to 65°C (-50°F to 150°F)
- Có thể hàn kín nhiệt: Có
- Điện trở suất khói: <500 ohm-cm
- Điện trở suất bề mặt: <31,000 ohms / sq.cm



Hình 3.4. Tấm Velostat

IC 74HC238

- Series: 74HC
- Loại IC: Decoder/Demultiplexer
- Dòng điện - Đầu ra Cao, Thấp: 5.2mA, 5.2mA
- Nguồn cung cấp điện áp: Đơn
- Điện áp cung cấp: 2V ~ 6V
- Kiểu lắp: Dán bề mặt
- Đóng gói: 16-SOIC

Transistor 2SC1815

- Loại transistor: NPN
- Dòng điện $I_{c\max}$: 0.15A

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

- Điện áp $V_{ce\ max}$: 50 V
- Kiểu đóng gói: SOT-23
- Kiểu chân: Dán bì mặt
- Số chân: 3
- Công suất max: 0.2 W

Diode chỉnh lưu 1N5819

- Điện áp ngược DC max (V_F): 40V
- Dòng chỉnh lưu trung bình (I_o): 1A
- Điện áp Forward max (V_F): 510mV @ 1A
- Tốc độ phục hồi: =< 500ns, > 200mA (I_o)
- Dòng rò ngược: 500 μ A @ 40V
- Điện dung: 30pF @ 10V, 1MHz
- Kiểu chân: Dán bì mặt
- Kiểu đóng gói: SOD-123F

❖ Tính toán

Như vậy, với thiết kế đưa ra và yêu cầu ban đầu ta có tổng hợp như sau về số cảm biến, số điểm xúc giác, số lượng đường dây cần kết nối với hàng và cột.

- Số lượng cảm biến xúc giác: 6 miếng nhỏ ở các ngón tay (ma trận 5×8) và 1 miếng lớn ở lòng bàn tay (ma trận 20×10)
- Số lượng điểm xúc giác của cả bàn tay:

$$6 \times (8 \times 5) + (20 \times 10) = 440 \text{ điểm xúc giác}$$

- Với 1 miếng cảm biến xúc giác $m \times n$ thì số lượng đường dây cần kết nối được tính bằng tổng số hàng và cột của nó (tức là $m+n$). Trong đề tài này, với vi điều khiển đã lựa chọn thì n là hằng số ($n = 10$) với giá trị là số kênh ADC

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

của vi điều khiển STM32F103C8T6. Vì vậy mà ta có thể quy 6 cảm biến 8×5 về 3 cảm biến 8×10 . Để cụ thể hơn ta có thể chia ra tính cả số lượng hàng và cột của bàn tay robot.

- Số lượng đường dây kết nối với hàng:

$$3 \times 8 + 20 = 44 \text{ hàng}$$

- Số lượng đường dây kết nối với cột:

$$3 \times 10 + 10 = 40 \text{ cột}$$

- Số lượng đường dây kết nối với hàng và cột của tay robot:

$$3 \times (8 + 10) + (20 + 10) = 84 \text{ đường dây}$$

Tính toán số IC 74HC238 cần sử dụng để mở rộng ngõ ra điều khiển ma trận cảm biến xúc giác

- 1 IC 74HC238 cung cấp tối đa 8 ngõ ra
- Số IC 74HC238 cần thiết:

$$44 : 8 = 5.5$$

- Như vậy số IC 74HC238 cần thiết để mở rộng ngõ ra điều khiển các hàng trong ma trận cảm biến là 6 IC.
- Vì: $2^5 - 1 = 31 < 44 < 63 = 2^6 - 1 \Rightarrow$ Số lượng đầu vào tối thiểu để điều khiển 6 IC 74HC238 là 6.

Tính toán điện trở chia áp R1 ở Hình 3.2.

- Nguồn 3.3V được chọn để cấp cho các IC 74HC238 với mục đích ngõ ra mang logic ‘1’ với điện áp là 3.3V và logic ‘0’ với điện áp là 0V với mục đích phù hợp cho tín hiệu trả về trong khoảng điện áp tham chiếu của bộ ADC của vi điều khiển.

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

- 1 ô cảm biến xúc giác trong ma trận có diện tích là:

$$3\text{mm} \times 3\text{mm} = 9\text{mm}^2.$$

- Với thông số từ datasheet, điện trở bề mặt nhỏ hơn $31000\Omega/\text{cm}^2$, điện áp mong muốn thu được từ 1V đến 3.3V (mạch điện nêu ở hình mạch điều khiển cảm biến). Chúng tôi tính toán và tìm ra giá trị điện trở như sau.
 - Với giới hạn dưới của 1 ô Velostat theo những gì chúng tôi kiểm tra có thể tụt xuống vài chục Ohm thì đạt được 3.3V theo sơ đồ mạch là điều hiển nhiên.
 - Tính toán điện trở theo nguồn trên, điện trở bề mặt của 1 ô Velostat 9mm^2 trên ma trận cảm biến nhỏ hơn $0.09 \times 31000 = 2790$ Ohm. Do đó, điện trở chia áp:

$$\frac{3.3 \times R}{2790 + R} = 1 \Rightarrow R = 1213.04 \text{ Ohm.}$$

- Trong thực tế, chúng tôi chọn điện trở là 1KOhm gần với giá trị đã tính toán, dễ tìm và mua.

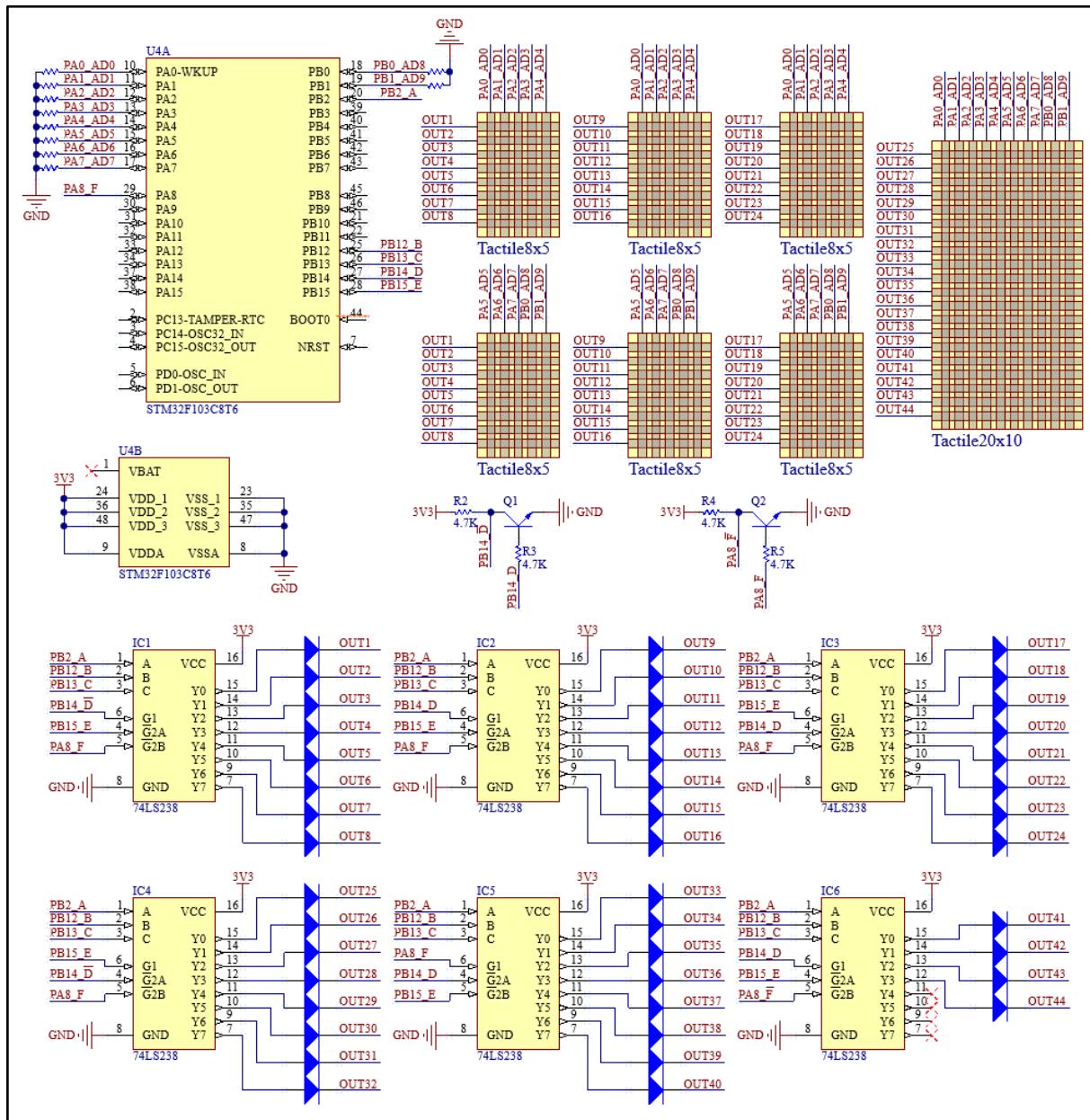
Thời gian chuyển đổi để đọc được một tín hiệu ADC với thời gian lấy mẫu là 239.5 chu kỳ

- SYSCLK = 72MHz tương ứng với ADCCLK = 12MHz
- $T_{conv} = 239.5 + 12.5 = 252$ chu kỳ = 21 us.
- Với ma trận 440 điểm thời gian chuyển đổi ADC tối thiểu là:

$$21 \times 440 = 9240 \text{ us} = 9.24 \text{ ms.}$$

❖ Sơ đồ nguyên lý

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG



Hình 3.5. Sơ đồ nguyên lý khói cảm biến

❖ Giải thích sơ đồ nguyên lý khói cảm biến

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

- Dựa vào sơ đồ trên, 6 ngõ vào được cung cấp bởi vi điều khiển (với 6 bit là liên tục, A mang trọng số thấp nhất và F mang trọng số lớn nhất) mang ý nghĩa như sau (với x là bit mang giá trị ngẫu nhiên):
 - Ngõ vào: 0b000xxx. IC1 là IC được ra lệnh bật cột duy nhất, tất cả ngõ ra còn lại là 0.
 - Ngõ vào: 0b001xxx. IC2 là IC được ra lệnh bật cột duy nhất, tất cả ngõ ra còn lại là 0.
 - Ngõ vào: 0b010xxx. IC3 là IC được ra lệnh bật cột duy nhất, tất cả ngõ ra còn lại là 0.
 - Ngõ vào: 0b011xxx. IC4 là IC được ra lệnh bật cột duy nhất, tất cả ngõ ra còn lại là 0.
 - Ngõ vào: 0b100xxx. IC5 là IC được ra lệnh bật cột duy nhất, tất cả ngõ ra còn lại là 0.
 - Ngõ vào: 0b101xxx. IC6 là IC được ra lệnh bật cột duy nhất, tất cả ngõ ra còn lại là 0.
- Mạch sử dụng transistor Q1 kèm điện trở R2, R3 và mạch sử dụng transistor Q2 kèm điện trở R4, R5 là 2 mạch NOT dùng để đảo logic 2 chân tín hiệu PB14 và PA8 của vi điều khiển để đáp ứng ngõ vào yêu cầu của mạch ghép tầng 6 IC 74HC238.
- Tactile8x5 là cảm biến nhỏ đặt ở các ngón tay và Tactile20x10 là cảm biến lớn đặt ở lòng bàn tay. Sơ đồ nguyên lý của các cảm biến sau khi thi công sẽ có số chân bằng tổng hàng và cột như trên. Các đường màu xám trong khói cảm biến có thể hiểu là vị trí băng keo lá đồng trong tấm cảm biến.
- Như đã đề cập, chúng tôi quy cảm biến về 3 tấm 8×10 và 1 tấm 20×10 để có 10 cột ứng với 10 kênh ADC của vi điều khiển. Với một ma trận tổng quát

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

có độ phân giải cần phải đọc là 44×10 thì các đường dây tên OUTx (x là một con số) thể hiện các mà khói xử lý trung tâm có thể điều khiển ngõ ra để chọn hàng nào trong ma trận lực được bật để đọc về lần lượt 10 kênh ADC của hàng đó.

- Chúng tôi disable chế độ continuous trong ADC để tránh việc lệch dữ liệu và dễ dàng hơn trong quá trình sắp xếp dữ liệu theo ý định từ trước. Thêm vào đó, thời gian lấy mẫu ADC được chọn ở mức tối đa là 239.5 chu kỳ để mang lại độ chính xác cao cho kết quả đọc được.
- Các diode chỉnh lưu 1N5819 được thêm vào ở ngõ ra các IC 74HC238 để tránh dòng điện dội trong hệ thống cảm biến xúc giác đi qua các điểm có điện trở thấp và dội về.

3.2.2. Khối động cơ

❖ Chức năng khôi

Khối động cơ có chức năng nhận tín hiệu điều khiển từ khói xử lý trung tâm và điều khiển các động cơ. Mỗi động cơ được điều khiển có chức năng điều khiển 1 khớp của tay robot.

❖ Phương án và lựa chọn

- Lựa chọn động cơ

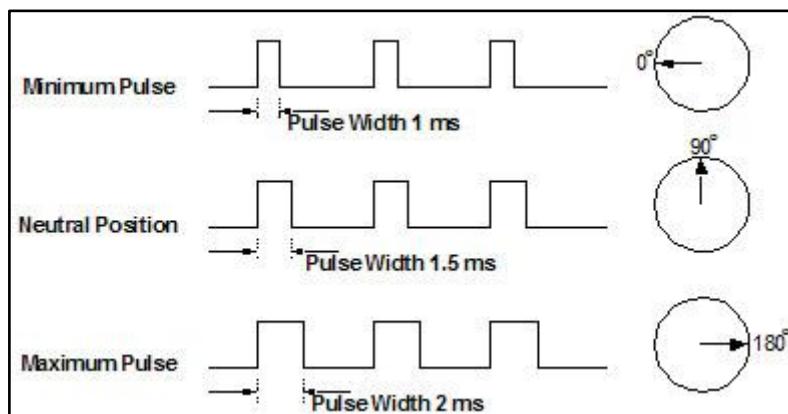
Động cơ xoay chiều quá lớn, động cơ DC quá nhanh và mô-men xoắn thấp. Mặt khác, động cơ bước và động cơ servo lại có những phẩm chất cần thiết để chế tạo một robot - hoạt động tốt với các dự án cơ khí, định vị chính xác và kích thước nhỏ gọn. Hai động cơ này giống nhau ở chỗ đều đảm bảo điều chỉnh linh hoạt tốc độ, vị trí và lực ra. Tuy nhiên, chúng tôi lựa chọn động cơ servo để điều khiển robot vì nó mang lại độ chính xác cao hơn về góc quay của động cơ và mô-men xoắn ổn định.

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

Với việc tự do trong việc lựa chọn vật thể mà cánh tay thao tác với, việc tính toán lực và chọn động cơ đã được chúng tôi bỏ qua trong quá trình chọn động cơ. Thay vào đó vấn đề cơ khí được chú trọng hơn. Do đó chúng tôi chọn loại servo khá truyền thống là MG946R với bánh răng kim loại, lực kéo tốt và có hỗ trợ phụ kiện đĩa servo để tiện lợi hơn trong quá trình thiết kế phần cơ khí cho robot

- Phương án điều khiển

Để điều khiển động cơ servo, vi điều khiển cần xuất tín hiệu PWM để có góc quay động cơ servo tương ứng. Tương tác giữa tín hiệu PWM và góc quay của động cơ servo được thể hiện trong hình sau



Hình 3.6. Tín hiệu điều khiển servo

Để truyền tín hiệu PWM cho khối điều khiển có rất nhiều phương án như hàng rào, domino, jack nối,... có thể được sử dụng để kết nối dây dẫn. Trong đó, chúng tôi lựa chọn phương án đơn giản, rẻ tiền nhất mà mang lại hiệu quả là 1 hàng rào đực và sử dụng dây bẹ để kết nối là phù hợp.

- Lựa chọn linh kiện

Trên thực tế có rất nhiều phương án để có thể điều khiển động cơ servo từ tín hiệu PWM của vi điều khiển, có thể kể đến như dùng các mạch transistor và điện trở khuếch

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

đại dòng, thiết kế các mạch cầu H dùng trở và transistor, các mô-đun điều khiển động cơ có sẵn trên thị trường, các IC mạch cầu điều khiển động cơ,... Trong đó, để phù hợp với yêu cầu đề tài đồng thời đáp ứng dòng điện ngõ ra cung cấp cho mỗi servo phải chịu được hơn 1.2A, chúng tôi lựa chọn phương án thiết kế mạch sử dụng IC L298N là IC mạch cầu kép có dòng điện ngõ ra mỗi kênh lên đến 2A.Thêm vào đó, chúng tôi còn trang bị cho IC một đế tản nhiệt để giảm nhiệt độ trong quá trình hoạt động với công suất cao. Ngoài ra, ở ngõ ra của mỗi kênh nhà sản xuất IC đề xuất thêm các diode nên chúng tôi chọn loại diode chỉnh lưu 1N5401 với dòng chỉnh lưu 3A là dư để hoạt động với dòng điện của động cơ.

Để truyền tín hiệu điều khiển cho khôi động cơ có rất nhiều phương án như hàng rào, domino, jack nối,... có thể được sử dụng để kết nối dây dẫn. Trong đó, chúng tôi lựa chọn phương án đơn giản, rẻ tiền nhất mà mang lại hiệu quả là các hàng rào đục để kết nối vào các đầu dây của động cơ servo.

❖ Thông số kỹ thuật

Động cơ Servo MG946R

- Điện áp: 4.8 → 7.2 V dc
- Dòng điện tối đa: 1.2A
- Kích thước: 40 x 19 x 43mm
- Nặng: 74g
- Tốc độ hoạt động: 0.2 giây / 60° (4.8V); 0.17 giây / 60° (6V)
- Mô-men xoắn: 9 kg.cm tại 4.8V; 12 kg.cm tại 6V
- Nhiệt độ hoạt động: 0°C ~ 55°C
- Chiều dài dây kết nối: 300mm

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG



Hình 3.7. Động cơ servo MG946R

IC L298N

- Loại động cơ bước: Lưỡng cực
- Loại động cơ AC, DC: Động cơ DC có chổi than
- Cấu hình ngõ ra Full H-Bridge, Dual
- Giao tiếp Parallel
- Dòng điện ngõ ra: 2A
- Điện áp cấp: 4.5 V ~ 7 V
- Điện áp tải: 4.8 V ~ 46 V
- Nhiệt độ hoạt động: -25°C ~ 130°C
- Kiểu chân: Xuyên lỗ

Diode chỉnh lưu 1N5401

- Loại Diode: Standard
- Điện áp ngược DC max (V_r): 100V

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

- Dòng chỉnh lưu trung bình (Io): 3A
- Điện áp Forward max (Vf): 1.2V @ 3A
- Tốc độ phục hồi: >500ns, > 200mA (Io)
- Dòng rò ngược: 5µA @ 100V
- Điện dung: 30pF @ 4V, 1MHz
- Kiểu chân: Xuyên lõi

❖ Tính toán

Dòng điện tối đa của động cơ servo MG946R lên đến 1.2A. Do đó, chúng tôi lựa chọn IC điều khiển động cơ là L298N vì trong datasheet IC này đề cập dòng điện ngõ ra ở mỗi kênh tín hiệu có thể lên đến 2A.

Dựa theo dòng điện trên đường tín hiệu ngõ ra của IC L298N có thể lên đến 1.2A. Do đó, chúng tôi có thể chọn diode chỉnh lưu với dòng định mức khoảng 1A. Nhưng trong thực tế, chúng tôi chọn loại diode chỉnh lưu 1N5401 với dòng điện định mức lên đến 3A vì yếu tố có sẵn và dư về mặt công suất cho linh kiện.

Do phần mạch này công suất khá cao nên phần đường dây của PCB cũng cần phải tính toán để đáp ứng đủ công suất theo công thức (3.1)

$$\text{Độ rộng dây} = \left(\frac{I}{k \times \text{Temp_Rise}^b} \right)^{\frac{1}{c}} \times \frac{1}{\text{Thickness} \times 1.378} \quad (3.1)$$

- Đường dây ngõ ra mỗi kênh của IC L298N, với I = 1.2 A, Temp_Rise = 10, Thickness = 1 oz, k = 0.048, b = 0.44, c = 0.725

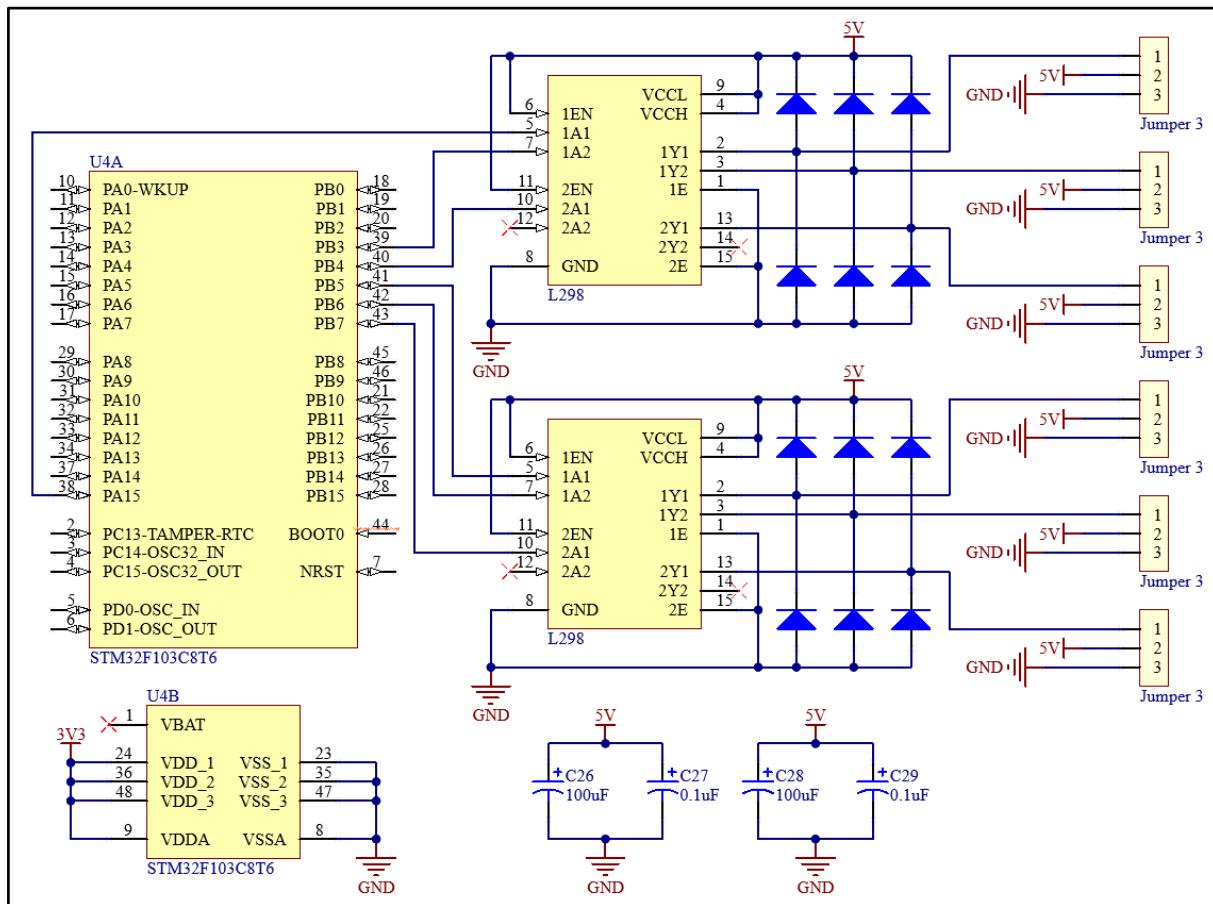
$$\text{Độ rộng dây} = \left(\frac{1.2}{0.048 \times 10^{0.44}} \right)^{\frac{1}{0.725}} \times \frac{1}{1 \times 1.378} = 15.21 \text{ mil}$$

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

- Trong thực tế, phần này của mạch được làm thủ công nên chúng tôi đặt ra độ rộng tối thiểu là 20mil để tiện lợi trong việc thi công mạch. Đồng thời, nó cũng thỏa mãn điều kiện tính toán đường dây ngõ ra mỗi kênh của IC L298N.
- Đường dây cung cấp nguồn cho mỗi IC L298N, với $I = 1.2 \times 3 = 3.6$ A, Temp_Rise = 10, Thickness = 1 oz, k = 0.048, b = 0.44, c = 0.725

$$\text{Độ rộng dây} = \left(\frac{3.6}{0.048 \times 10^{0.44}} \right)^{\frac{1}{0.725}} \times \frac{1}{1 \times 1.378} = 69.21 \text{ mil}$$

❖ Sơ đồ nguyên lý



Hình 3.8. Sơ đồ nguyên lý khôi động cơ

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

❖ Giải thích sơ đồ nguyên lý

- Các tụ C26, C27, C28, C29 là các tụ phân cực được đặt gần IC L298N để lọc nhiễu ở tần số cao và tần số thấp nguồn cung cấp cho IC. Đồng thời, việc thêm các tụ này gần IC L298N hoạt động với điện áp ổn định hơn.
- Để xuất tín hiệu này, chúng tôi lựa chọn sử dụng kênh 1 và 2 của 3 bộ Timer TIMER2, TIMER3, TIMER4 là chân PA15 (TIM2_CH1), PB3 (TIM2_CH2), PB4 (TIM3_CH1), PB5 (TIM3_CH2), PB6 (TIM4_CH1), PB7 (TIM4_CH2) là 6 chân kế nhau của STM32F103C8T6 để tiện đi dây trong quá trình thiết kế mạch.
- Tín hiệu PWM được xuất từ vi điều khiển sẽ được IC L298N nhận như tín hiệu ngõ vào của IC L298N để xuất tín hiệu điều khiển dựa trên điện áp được cấp ở chân 4 của IC L298N.
- Dựa theo datasheet của L298N, các diode ở ngõ ra được nhà sản xuất đề xuất khi hoạt động với các tải như động cơ. Yêu cầu đưa ra là điện áp rơi nhỏ nhất có thể, dựa vào các yêu cầu chúng tôi chọn diode 1N5401 ($V_r = 100V$, $I_o = 3A$, $V_F = 1.2V$).
- Các Jumper 3 là phần hàng rào đặc sẽ kết nối tín hiệu ngõ ra của IC điều khiển động cơ để kết nối trực tiếp với dây của động cơ servo.
- Ngoài ra, vì công suất khá lớn các IC L298N còn được chúng tôi trang bị phần tản nhiệt để hoạt động hiệu quả hơn.

3.2.3. Khối điều khiển trung tâm

a. Khối xử lý trung tâm

❖ Chức năng

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

Khối xử lý trung tâm xuất tín hiệu điều khiển ma trận cảm biến, đọc và xử lý dữ liệu từ khói cảm biến, nhận tín hiệu điều khiển từ khói điều khiển, xuất tín hiệu cho khói hiển thị, xuất tín hiệu điều khiển khói động cơ, giao tiếp với khói giao diện để truyền nhận dữ liệu.

❖ Phương án và lựa chọn

Với số cảm biến xúc giác đã thiết kế và cách lấy dữ liệu của cảm biến được trình bày như hình trên, ta có thể đưa ra một số yêu cầu cho mạch trung tâm như sau:

- 10 kênh ADC
- 44 ngõ ra cho kết nối hàng của các miếng cảm biến. Với số ngõ ra nhiều như vậy thì phương án tốt nhất là sử dụng IC mở rộng port 74HC238. Lúc này từ yêu cầu 44 ngõ ra ta có thể quy về thành 6 ngõ ra.
- Tối thiểu 1 bộ UART và 1 bộ USB dùng để truyền nhận dữ liệu với giao diện người dùng để chúng tôi có thể thử và kiểm tra giao tiếp dữ liệu với cả 2 giao thức.
- Tối thiểu 1 bộ I2C (dự phòng để hiển thị LCD hoặc gắn cảm biến).
- Tối thiểu 3 bộ Timer. 2 bộ Timer để điều khiển 6 động cơ servo và 1 bộ dùng cho RTOS.
- Yêu cầu tối thiểu của vi điều khiển sử dụng FreeRTOS
 - Tốc độ xử lý lớn hơn 25MHz
 - Dung lượng RAM từ 64Kbyte trở lên

Như vậy, STM32F103C8T6 là vi điều khiển được chọn vì sự phù hợp với những yêu cầu đặt ra cho hệ thống.

Thêm vào đó, vi điều khiển đã lựa chọn có 2 phương án lựa chọn debugger là giao thức JTAG và SWD (serial wired debug). Trong thực tế có rất nhiều mạch nạp được sản

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

xuất dựa trên 2 chuẩn trên đã kể, chúng tôi sử dụng mạch nạp ST-Link V2 kết nối với mạch thông qua 1 hàng rào cong 4 chân để làm debugger vì sự nhỏ gọn và giá thành rẻ của nó. ST-Link V2 hỗ trợ đầy đủ các STM32 giao thức SWD nạp và gỡ lỗi, giao diện đơn giản 4 dây (bao gồm cả dây cấp nguồn), tốc độ nhanh, hoạt động ổn định. Trong đó, phần mềm STM32CubeIDE mà chúng tôi sử dụng cũng tương tác rất tốt với mạch nạp này. Ngoài ra, ST-Link V2 cũng hỗ trợ nâng cấp firmware tự động, theo sản phẩm của ST.

❖ Thông số kỹ thuật

Vì điều khiển STM32F103C8T6

- Lõi CPU ARM 32-bit Cortex-M3
 - Tần số tối đa 72 MHz, hiệu suất 1.25 DMIPS/MHz (Dhrystone 2.1) khi truy cập bộ nhớ trạng thái chờ 0
 - Nhân chu kỳ đơn và phân chia phần cứng
- Memory
 - 64 hoặc 128 Kbyte bộ nhớ Flash
 - 20 Kbyte SRAM
- Clock, reset và quản lý nguồn cung cấp
 - Nguồn cung cấp ứng dụng 2.0 đến 3.6 V và I/O
 - POR, PDR, và bộ dò điện áp có thể lập trình (PVD)
 - Bộ dao động tinh thể 4 đến 16 MHz
 - RC nội bộ được cắt tỉa tại nhà máy 8 MHz
 - RC bên trong 40 kHz
 - PLL cho xung nhịp CPU
 - Bộ dao động 32 kHz cho RTC với hiệu chuẩn

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

- Tiết kiệm năng lượng
 - Chế độ Ngủ, Dừng và Chế độ chờ
 - Cung cấp VBAT cho RTC và thanh ghi dự phòng
- Bộ chuyển đổi Analog/Digital 2 x 12 bit, 1 µs (lên đến 16 kênh)
 - Phạm vi chuyển đổi: 0 đến 3,6 V
 - Lấy mẫu kép và khả năng giữ
 - Cảm biến nhiệt độ
- DMA
 - Bộ điều khiển DMA 7 kênh
 - Các thiết bị ngoại vi được hỗ trợ: bộ định thời, ADC, SPI, I2C và USART
- Lên đến 37 cổng I/O nhanh
 - 37 I/O, tất cả đều có thể lập bản đồ trên 16 vectơ ngắn bên ngoài
 - Chế độ kiểm tra sửa lỗi: Giao diện gỡ lỗi dây nối tiếp SWD & JTAG
- Timer
 - 4 x Bộ định thời 16 bit, mỗi bộ có tới 4 IC/OC/PWM hoặc bộ đếm xung và đầu vào bộ mã hóa vuông góc (tăng dần)
 - Bộ hẹn giờ PWM điều khiển động cơ, 16-bit với tạo thời gian chót và dừng khẩn cấp
 - Bộ định giờ cơ quan giám sát (Độc lập và Cửa sổ)
 - SysTick timer 24-bit downcounter
- Giao thức giao tiếp
 - Lên đến 2 x giao diện I2C (SMBus/PMBus)

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

- Lên đến 3 USART (giao diện ISO 7816, LIN, khả năng IrDA, điều khiển modem)
- Lên đến 2 SPI (18 Mbit/s)
- Giao diện CAN (2.0B Hoạt động)
- Giao diện tốc độ đầy đủ USB 2.0
- Đơn vị tính CRC, ID duy nhất 96 bit

ST-Link V2

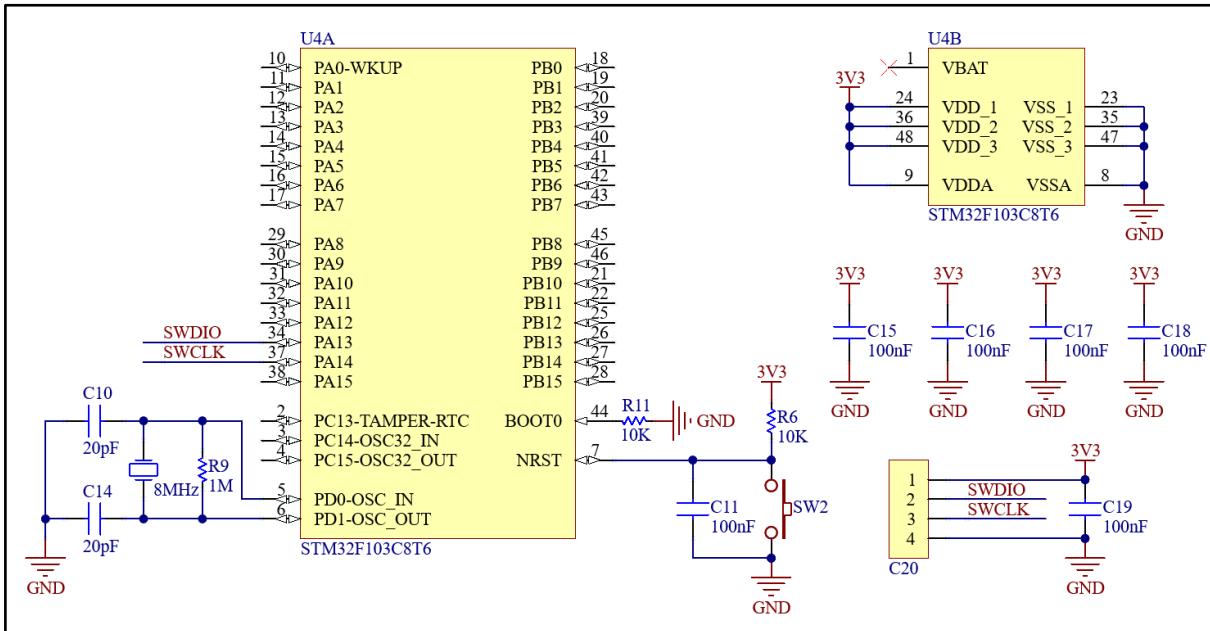
- Mạch nạp STM8, STM32 ST-Link V2 Mini.
- Chuẩn nạp: JTAG, SWD, SWV.

❖ Tính toán

Khối xử lý trung tâm có công suất rất nhỏ nên chủ yếu các linh kiện điện trở và tụ trong khối được chọn là loại SMD 0603 vì độ nhỏ gọn và phù hợp cho việc thi công mạch. Vì có công suất nhỏ nên việc lựa chọn linh kiện liên quan hay tính toán về đường dây để linh kiện có thể đáp ứng là không cần thiết nên chúng tôi sẽ bỏ qua tính toán cho khối này.

❖ Sơ đồ nguyên lý

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG



Hình 3.9. Sơ đồ nguyên lý khôi xử lý trung tâm

❖ Giải thích sơ đồ nguyên lý

- Vì lập trình hệ điều hành FreeRTOS tốn thời gian chuyển đổi giữa nhiều tác vụ, chúng tôi mong muốn vi điều khiển có thể hoạt động với clock hệ thống đạt mức tối đa là 72 MHz nên một thạch anh ngoại là điều cần thiết. Mạch dao động thạch anh ngoại được tham khảo từ sơ đồ nguyên lý của board Blue Pill sử dụng vi điều khiển tương tự được cung cấp từ chính hãng. Để tăng độ ổn định tần số, người ta dùng thêm 2 tụ nhỏ C10, C14 (20pF), tụ bù nhiệt ổn tần. Các linh kiện tụ, điện trở và thạch anh đều được sử dụng ở dạng dán. Tụ và điện trở sử dụng loại linh kiện mã 0603 để phù hợp với việc tự thi công và mạch dao động thạch anh có công suất nhỏ. Khi layout phần này, thạch anh được đặt gần vi điều khiển, đường dây xuất phát từ vi điều khiển nối qua 2 chân thạch anh rồi đến tụ và xuống ground, tổng chiều dài dây dẫn từ 2 chân vi điều khiển đến 2 chân thạch anh bằng nhau, chỉ đi dây trên 1 lớp,

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

bố trí thạch anh xa những linh kiện hoặc những bộ phận có khả năng gây nhiễu.

- Với mong muốn đề ra là chỉ nạp chương trình vào bộ nhớ Flash của vi điều khiển nên đó là lý do chúng tôi để mặc định chân BOOT0 được kết nối với mức logic ‘0’ thông qua một điện trở 10KOhm để giảm cường độ dòng điện. Với dòng điện qua điện trở 10KOhm như vậy thì sử dụng loại điện trở 0603 với công suất định mức vào khoảng 0.1W (tùy vào nhà cung cấp) thì công suất loại điện trở này được sử dụng dư và kích thước của nó là hợp lý để có thể tự thi công mạch.
- Ngoài ra 4 tụ C15, C16, C17, C18 là 4 tụ đặt gần 4 cặp chân nguồn nhất có thể mang giá trị 100nF để loại bỏ nhiễu từ nguồn cung cấp đồng thời ổn định nguồn điện cung cấp cho vi điều khiển.
- Tụ C19 với giá trị 100nF được đặt sát hàng rào để lọc nguồn cung cấp. Chân SWDIO của ST-Link V2 kết nối với chân PA13 và chân SWCLK của ST-Link V2 kết nối với chân PA14.

b. Khối điều khiển

❖ Chức năng

Khối điều khiển được sử dụng để giao tiếp giữa thao tác của người dùng với hệ thống. Hỗ trợ người dùng thao tác gửi tín hiệu đến khói xử lý trung tâm.

❖ Phương án và lựa chọn

Một số thiết bị dùng để giao tiếp giữa thao tác của người sử dụng với mạch điện như nút nhấn, phím nhấn cảm ứng, switch, encoder vô cáp... Với mục tiêu thiết kế đơn giản, tiện lợi, chi phí thấp nên chúng tôi chọn nút nhấn. Việc sử dụng nút nhấn giúp người dùng thao tác gửi tín hiệu một cách đơn giản, dễ sử dụng.

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

❖ Thông số kỹ thuật

Nút nhấn

- Cấu hình tiếp điểm: Single Pole Single Throw (SPST)
- Kiểu chân: Dán bề mặt
- Hoạt động: Nhấn nhả
- Màu nút nhấn: Đỏ
- Kích thước: 3 x 6 x 2.5mm
- Dòng định mức tiếp điểm: 50 mA @ 12 V dc
- Vật liệu tiếp điểm: Niken
- Tuổi thọ: 100000 lần

❖ Tính toán

Nhóm chọn thiết kế nút nhấn theo dạng tích cực mức thấp như sơ đồ nguyên lý trong hình sau.

Việc tính toán trở kéo lên phải ta dựa vào datasheet của vi điều khiển, để đảm bảo cho vi điều khiển hoạt động an toàn và tiết kiệm năng lượng. Ta mong muốn dòng tiêu thụ khi nhấn phím sẽ rất nhỏ để an toàn cho vi điều khiển.

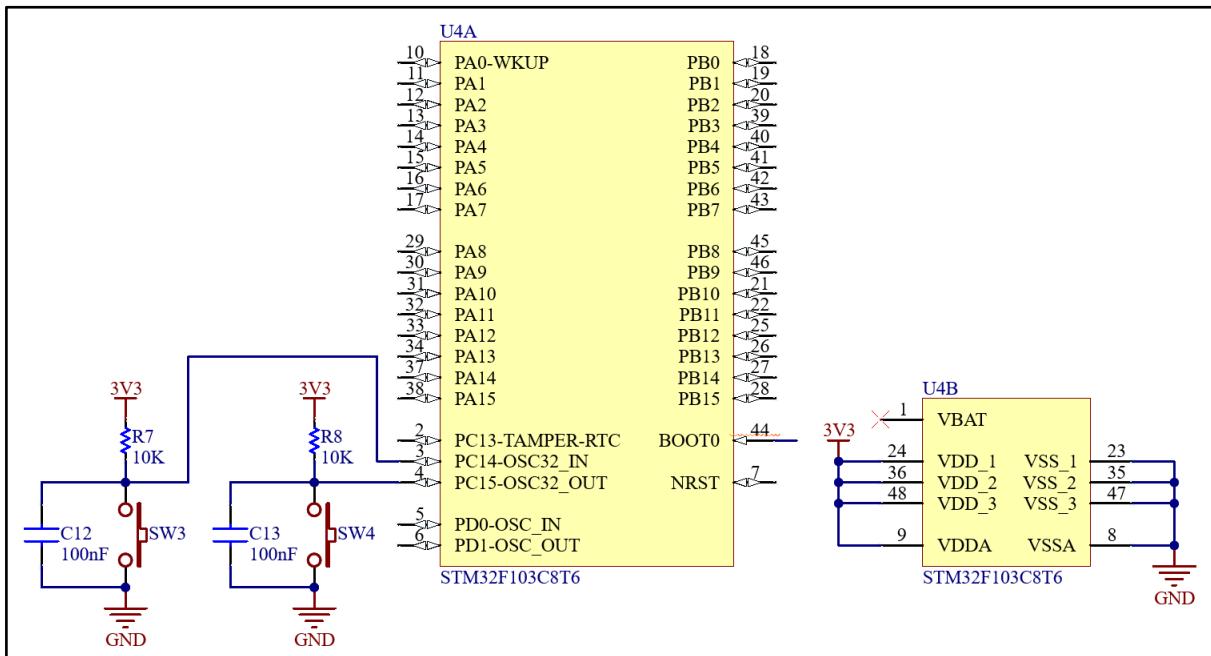
Trong thực tế, có thể chọn điện trở từ 4.7 Kohm đến 10 KOhm. Chọn điện trở giá trị 10 KOhm, áp dụng định luật Ohm ta tính được giá trị dòng điện như sau.

$$I = \frac{V - 0}{R_{pullup}} = \frac{3.3 - 0}{10000} = 0.00033 \text{ (A)}$$

Vậy ta chọn giá trị $R = 10K\Omega$.

❖ Sơ đồ nguyên lý

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG



Hình 3.10. Sơ đồ nguyên lý khói điều khiển

❖ Giải thích sơ đồ nguyên lý

- Quan sát hình sơ đồ nguyên lý khói điều khiển, nút nhấn được kết nối với chân PC14 và PC15 của MCU và được kéo lên nguồn 3.3V thông qua trở R7 và R8, chân còn lại của nút nhấn sẽ nối GND, tụ 100nF được mắc song song để làm phẳng tín hiệu hơn và hỗ trợ chống dội nút nhấn. Khi không nhấn nút thì tín hiệu có mức logic “1” do có điện trở kéo lên nguồn 3.3V, ngược lại tín hiệu có mức logic “0” vì khi nút nhấn được nhấn.

c. Khối hiển thị

❖ Chức năng

Khối hiển thị hiển thị trạng thái khi có tín hiệu hoặc nhận tín hiệu phản hồi từ khói xử lý trung tâm.

❖ Phương án và lựa chọn

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

Để hiển thị, báo trạng thái cho người sử dụng khi hệ thống truyền, nhận dữ liệu thành công. LED, màn hình LCD, màn hình OLED hoặc Buzzer ... thường được sử dụng để hiển thị trong các hệ thống, thiết bị trong thực tế, ... Đối với nhu cầu thiết kế nêu trên chúng tôi chọn LED 0603 để hiển thị trạng thái vừa phù hợp về giá thành và tối ưu kích thước của mạch.

❖ Thông số kỹ thuật

LED 0603

- Đóng gói: 1608M (0603)
- Kiểu chân: Dán bề mặt
- Điện áp thông dòng (V_F): 2V
- Cường độ sáng: 180 mcd
- Số chân: 2
- Kích thước: 1.6 x 0.8 x 0.8mm

❖ Tính toán

LED 0603 xanh được nhóm sử dụng cho khối hiển thị với $V_F = 2V$, $I_F = 20mA$, với điện áp cung cấp $V = 3.3V$ giá trị điện trở hạn dòng cho LED được tính:

$$R = \frac{V - V_F}{I_F} = \frac{3.3 - 2}{0.02} = 65 \text{ Ohm}$$

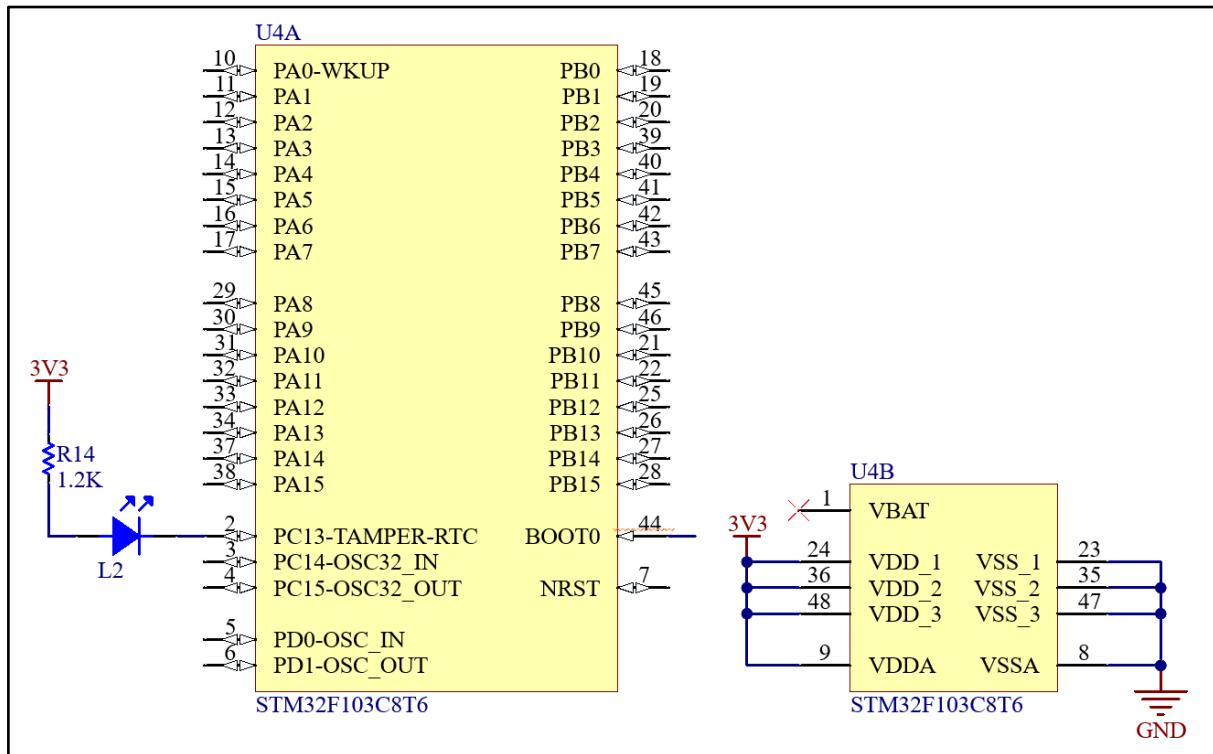
$$P = V_F \times I_F = 0.04 \text{ W}$$

Từ công suất trên, chúng tôi chọn LED và điện trở loại 0603 để phù hợp với yêu cầu của mạch

Vì nhu cầu dùng LED để báo tín hiệu, nên không cần LED quá sáng, vì vậy để nâng cao tuổi thọ cho LED cũng như tốt cho mắt trong quá trình làm việc nên trong thực tế chúng tôi chọn $R = 1.2 \text{ KOhm}$.

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

❖ Sơ đồ nguyên lý



Hình 3.11. Sơ đồ nguyên lý khối hiển thị

❖ Giải thích sơ đồ nguyên lý

- LED hiển thị được mắc như sơ đồ nguyên lý hình trên, chân Cathode của LED được nối với chân PC13 của STM32F103C8T6, còn chân Anode qua trở R14 kéo lên 3.3V. Trở RR14 có tác dụng hạn dòng cho LED L2.

3.2.4. Khối giao diện

❖ Chức năng

Khối giao diện là truyền và nhận dữ liệu nối tiếp từ khối nhận dạng và khối xử lý trung tâm. Là trung gian giữa bàn tay robot và người dùng với khả năng điều khiển và hiển thị các thông tin quan trọng liên quan.

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

❖ Phương án và lựa chọn

Để giao tiếp với khói giao diện người ta thường sử dụng 2 loại dữ liệu nối tiếp có dây là UART và USB để có sự ổn định về đường truyền và tín hiệu nhất định, nhiều công cụ hỗ trợ debug tín hiệu và nhiều API hỗ trợ. Trong đó, UART được biết đến là truyền nhận theo tín hiệu single-ended signal có thể hiểu đơn giản là tín hiệu trên đường truyền được quyết định bởi tín hiệu trên 1 đường dây. Còn USB được biết đến là truyền nhận theo tín hiệu differential signal có thể hiểu đơn giản là tín hiệu trên đường truyền được quyết định bởi chênh lệch điện áp của 1 cặp dây. Theo đó, USB nổi lên như là một phương án mạnh mẽ hơn trong việc giao tiếp dữ liệu nối tiếp về độ ổn định, chống nhiễu mạnh mẽ. Nhưng với mục đích kiểm thử cả 2 giao thức, chúng tôi thiết kế mạch có cả UART và USB để giao tiếp dữ liệu nối tiếp.

Với giao thức UART, chúng tôi chọn 2 bộ UART1 (chân PA9 truyền và chân PA10 nhận dữ liệu) và UART3 (chân PB10 truyền và chân PB11 nhận dữ liệu) cho khói giao tiếp dữ liệu. Ngoài ra, chúng tôi nhận thấy sử dụng hàng rào kết hợp mô-đun UART to USB PL2303 để giao tiếp chuẩn UART sẽ tiết kiệm chi phí và đơn giản hơn trong việc thiết kế mạch.

Đối với giao thức USB, sơ đồ mạch kết nối USB (tham khảo từ mạch Blue Pill của chính hãng với mục đích tương tự) mang tính chất đơn giản hơn và không có sẵn mô-đun trên thị trường. Vì vậy, chúng tôi sử dụng cổng microUSB kết hợp điện trở để thiết kế cho giao thức USB (chân PA12 là D+ và chân PA11 là D-).

Thêm vào đó, tốc độ giao tiếp dữ liệu nối tiếp mà chúng tôi thường xuyên sử dụng là 115200 và 250000 baud/s. Lý do là vì:

- 115200 là mức tốc độ tối đa mà các công cụ terminal hỗ trợ debug dữ liệu nối tiếp hỗ trợ (ví dụ như hercules, realterm,...)

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

- 250000 là mức tốc độ tối đa mà vẫn bảo đảm an toàn cho các mô-đun UART to USB

❖ Thông số kĩ thuật

Cổng microUSB

- Loại đầu: Cái
- Loại USB: Micro B
- Kiểu chân: Dán bè mặt
- Cách gắn: Hàn
- Phiên bản USB: 2.0
- Hướng thân: Thẳng hàng
- Dòng định mức: 1.8A
- Vật liệu tiếp điểm: Đồng

USB to UART TTL PL2303

- Điện áp 5 VDC cấp trực tiếp từ cổng USB
- Ngõ ra mức TTL với 2 chân Tx, Rx
- Với 3 LED hiển thị trạng thái: Led nguồn, Led Tx, và Led Rx
- Kích thước: 15 x 31 mm

❖ Tính toán

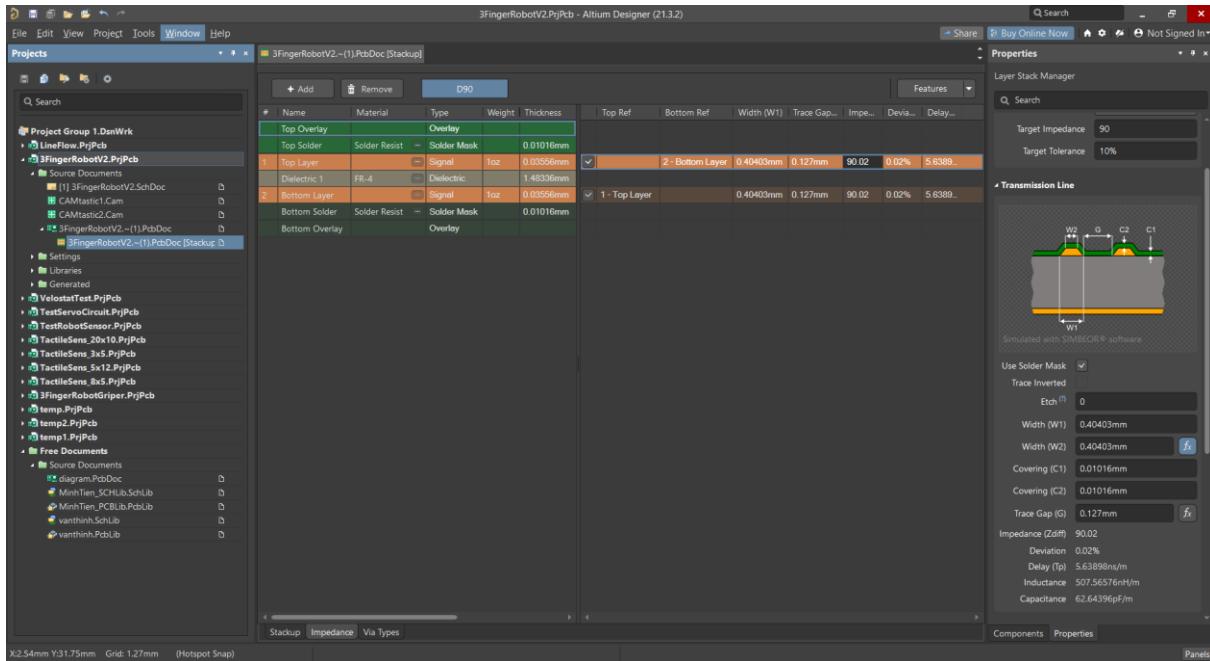
Gắn tụ với giá trị 100nF được đặt gần mô-đun của giao thức UART để lọc nhiễu và ổn định nguồn cung cấp.

Đối với USB, các điện trở đặt ở chân D+ và D- được chúng tôi chọn là 20 Ohm và khi layout PCB cần đặt 2 điện trở này gần với vi điều khiển STM32. Đồng thời cặp dây D+ và D- là một cặp dây differential signal vì vậy mà 2 dây này được tính toán kĩ

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

lưỡng, đặt sát nhau ở khoảng cách nhất định để có độ nhiễu tương đương nhau và có độ dài dây bằng nhau.

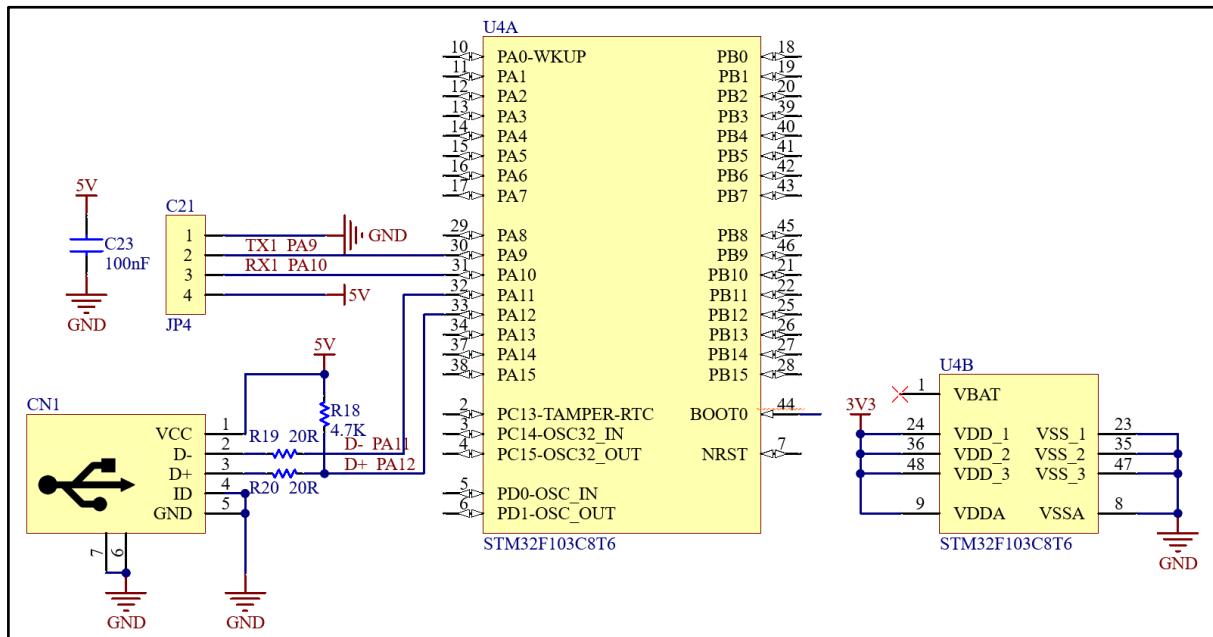
Sau đây là phần hỗ trợ tính toán cho cặp dây D+ và D- bởi phần mềm Altium với trở kháng vi sai của USB là 90 Ohm.



Hình 3.12. Tính toán cặp dây D+ và D- cho USB trong phần mềm Altium

❖ Sơ đồ nguyên lý

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG



Hình 3.13. Sơ đồ nguyên lý phần kết nối khối xử lý trung tâm với khối giao diện

❖ Giải thích sơ đồ nguyên lý

- Quan sát hình trên, nhóm sử dụng UART với tốc độ baud 115200 để giao tiếp với khối giao diện thông qua UART to USB PL2303. Chân TX của module UART to USB PL2303 nối với chân PA10_RX của bộ USART1 của STM32F103C8T6. Chân RX của module UART to USB PL2303 nối với chân PA9_TX của bộ USART1 của STM32F103C8T6.
- Chân D- và D+ đi qua 2 điện trở 20 Ohm và vào 2 chân của vi điều khiển theo thứ tự lần lượt là PA11 và PA12.

3.2.5. Khối nguồn

❖ Chức năng

Khối nguồn: có chức năng cấp nguồn cho khối hiển thị, khối điều khiển, khối xử lý trung tâm, khối cảm biến và khối động cơ.

CHƯƠNG 4. THI CÔNG HỆ THỐNG

❖ Phương án và lựa chọn

Để thiết kế khôi nguồn phù hợp cung cấp cho hệ thống, chúng tôi có bảng sau liệt kê mức điện áp hoạt động và dòng tiêu thụ tối đa của các linh kiện chính thuộc khôi trung tâm.

Bảng 3.1. Dòng và áp các linh kiện trong hệ thống

STT	Linh kiện	Số lượng	Điện áp hoạt động (V)	Dòng điện tối đa (A)	Tổng dòng điện tối đa (mA)
1	Động cơ servo MG946R	6	5	1.2	7.2
2	L298N	2	5	0.1	0.2
3	UART to USB PL2303	1	5	0.024	0.024
4	Dự phòng			0.5	0.5
Tổng dòng tiêu thụ 5V					7.924
1	STM32F103C8T6	1	3.3	0.05	0.05
2	74HC238	6	3.3	0.026	0.156
3	ST-Link v2	1	3.3	0.025	0.025
4	Dự phòng			0.3	0.3
Tổng dòng tiêu thụ 3.3V					0.531

Các linh kiện khác như tụ điện, điện trở ... không được liệt kê ở bảng trên do dòng tiêu thụ nhỏ và có thể bỏ qua.

Qua bảng trên ta thấy mạch của khôi điều khiển gồm 2 mức điện áp hoạt động là 5V với dòng tiêu thụ có thể lên đến khoảng 8A và 3.3V với dòng điện tiêu thụ chỉ khoảng 0.5A. Vì thế để có mức điện áp nguồn 5V chúng tôi chọn cấp nguồn từ bộ nguồn tổ ong 5V – 10A có sẵn trên thị trường, loại nguồn này có thể điều chỉnh điện áp lên mức hơn 6V. Tiếp theo dùng nguồn 6V qua mạch ổn áp công suất sử dụng IC ổn áp 7805, 1 điện trở công suất và 1 transistor công suất để thu được điện áp ổn định ở mức 5V và dòng điện chịu được lên đến 15A (phụ thuộc vào việc lựa chọn transistor công suất). Thêm

CHƯƠNG 4. THI CÔNG HỆ THỐNG

vào đó, chúng tôi còn sử dụng một LED để báo trạng thái nguồn mức 5V. Để lấy điện áp 3.3V, ta dùng nguồn điện áp cung cấp cho ngõ vào của IC ổn áp AMS1117 – 3.3V, nhóm lựa chọn IC ổn áp tuyển tính này có kiểu chân SOT-223 nhỏ gọn và có khả năng cấp dòng 800mA, là nguồn cung cấp cho mạch. Và một LED báo trạng thái nguồn 3.3V.

❖ Thông số kỹ thuật

IC ổn áp 7805

- Loại IC: Ổn áp
- Điện áp ngõ ra: 5V
- Dòng điện ngõ ra: 1.5A
- Điện áp ngõ vào max: 35V
- Điện áp ngõ vào min: 7V
- Nhiệt độ hoạt động: 0°C ~ 125°C
- Kiểu chân: Xuyên lỗ

IC AMS1117

- Điện áp đầu vào tối đa: 30V
- Điện áp đầu vào tối đa (khuyên dùng): 18V
- Điện áp đầu ra: ổn định ở mức 3.3/5V
- Dòng điện ngõ ra: 1A
- Nhiệt độ hoạt động: -40°C ~ 125°C
- Kiểu chân: Dán bề mặt
- Kiểu đóng gói: SOT-223

Transistor BUW42AP

- Loại transistor: PNP
- Dòng điện I_c max: 15A

CHƯƠNG 4. THI CÔNG HỆ THỐNG

- Điện áp $V_{ce\ max}$: 400 V
- Kiểu chân: Xuyên lõi
- Số chân: 3
- Công suất max: 80 W

LED đỗ 5mm

- Kiểu chân: Xuyên lõi
- Điện áp cấp: 2V
- Số chân: 2
- Góc nhìn: 60°
- Kiểu lens: tròn

LED 0603 đỗ

- Đóng gói: 1608M (0603)
- Kiểu chân: Dán bè mặt
- Điện áp: 2 V
- Cường độ sáng: 180 mcd
- Số chân: 2
- Kích thước: 1.6 x 0.8 x 0.8mm

❖ Tính toán

Khi mạch điều khiển cần dòng điện lớn như đã tính toán ở trên, dòng điện từ nguồn sẽ qua các tụ đèn điện trở công suất 1 Ohm 10W được đặt giữa cực B và cực E sẽ được điều chỉnh và khuếch đại ở cực C của transistor để đáp ứng dòng ngõ ra đủ lớn, dòng ngõ ra này sẽ có điện áp được quy định bằng ngõ ra ổn áp ở mức 5V của IC 7805 ở nhánh dưới. Dòng điện tối đa mà khối nguồn có thể sinh ra được phụ thuộc vào lựa chọn transistor. Với yêu cầu trên, chúng tôi chọn transistor BUW42AP với dòng điện

CHƯƠNG 4. THI CÔNG HỆ THỐNG

cực C có thể chịu đến 15A để transistor có thể thoải mái trong việc chịu dòng cao mà không dễ hư hỏng cũng như khả năng bị nóng cũng ít hơn. Ngoài ra, transistor và IC ổn áp còn được trang bị tản nhiệt để giảm nhiệt trong quá trình hoạt động với công suất lớn. Quan sát sơ đồ nguyên lý Hình 3.2. ta có tụ C5 và C6 có chức năng lọc nhiễu tần số thấp và nhiễu tần số cao trên chân V_{IN} , nhằm ổn định điện áp ngõ vào cho IC 7805, chúng tôi dùng nguồn tổ ong cấp cho ngõ vào V_{IN} . Tụ C7 và C8, cũng có chức năng lọc nhiễu tần số thấp và tần số cao cho ngõ ra có điện áp ổn định. Ở ngõ ra nguồn 5V nhóm thiết kế một LED L1 để báo trạng thái nguồn, khi được cấp nguồn và được ổn áp 5V qua IC, LED sẽ sáng, ngược lại LED tắt.

Tiếp theo việc tính toán, chọn lựa giá trị linh kiện để IC AMS1117-3.3V hoạt động nhóm dựa vào datasheet của nhà sản xuất AMS. Quan sát sơ đồ nguyên lý Hình 3.2. ta có tụ C1 và C2 có chức năng lọc nhiễu tần số thấp và nhiễu tần số cao trên chân V_{IN} , nhằm ổn định điện áp ngõ vào cho IC AMS1117 – 3.3V, chúng tôi dùng nguồn tổ ong cấp cho ngõ vào V_{IN} . Tụ C3 và C4, cũng có chức năng lọc nhiễu tần số thấp và tần số cao cho ngõ ra có điện áp ổn định. Ở ngõ ra nguồn 3.3V nhóm thiết kế một LED L3 để báo trạng thái nguồn, khi được cấp nguồn và được ổn áp 3.3V qua IC, LED sẽ sáng, ngược lại LED tắt.

Do phần công suất ngõ ra ổn áp 5V cao nên phần đường dây của PCB cũng cần phải tính toán để đáp ứng đủ công suất theo công thức (3.1)

- Với $I = 8 \text{ A}$, $\text{Temp_Rise} = 10$, $\text{Thickness} = 1 \text{ oz}$, $k = 0.048$, $b = 0.44$, $c = 0.725$, áp dụng công thức (3.1)

$$\text{Độ rộng dây} = \left(\frac{8}{0.048 \times 10^{0.44}} \right)^{\frac{1}{0.725}} \times \frac{1}{1 \times 1.378} = 208.2 \text{ mil}$$

- Trong thực tế, phần này của mạch được làm thủ công nên chúng tôi đặt ra độ rộng vào khoảng 70 mil để tiện lợi trong việc thi công mạch. Đồng thời, chúng

CHƯƠNG 4. THI CÔNG HỆ THỐNG

tối ưu hóa phần mạch bằng cách tra thiếc lên các đường dây để có thể chịu được công suất như trên.

LED đỏ 5mm được nhóm sử dụng để báo trạng thái nguồn 5V, với $V_F = 2V$, $I_F = 20mA$, với điện áp cung cấp $V = 5V$ giá trị điện trở hạn dòng cho LED được tính:

$$R = \frac{V - V_F}{I_F} = \frac{5 - 2}{0.02} = 150 \text{ Ohm}$$

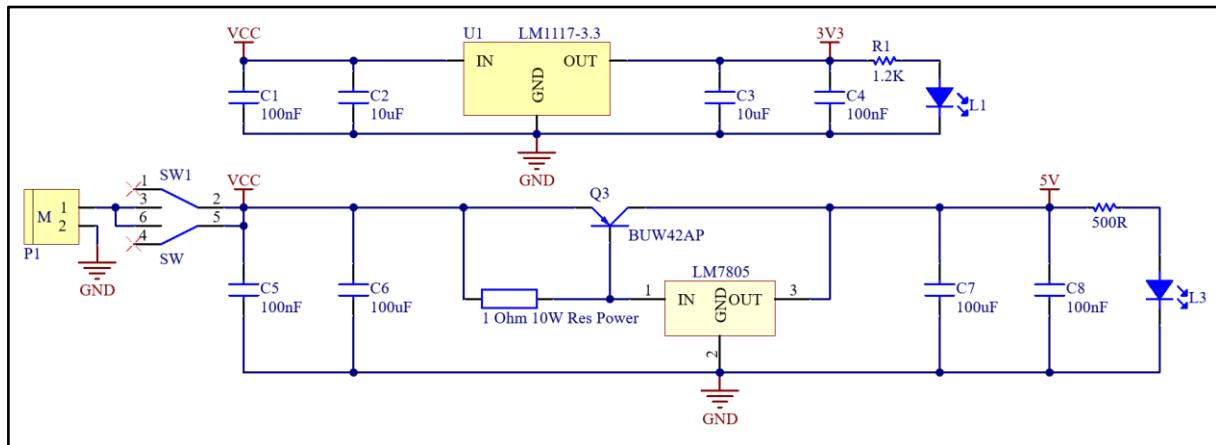
Vì nhu cầu dùng LED để báo nguồn, nên không cần LED quá sáng, vì vậy để nâng cao tuổi thọ cho LED cũng như tốt cho mắt trong quá trình làm việc nên trong thực tế chúng tôi chọn $R = 500 \text{ Ohm}$.

LED 0603 đỏ được nhóm sử dụng để báo trạng thái nguồn 3.3V, với $V_F = 2V$, $I_F = 20mA$, với điện áp cung cấp $V = 3.3V$ giá trị điện trở hạn dòng cho LED được tính:

$$R = \frac{V - V_F}{I_F} = \frac{3.3 - 2}{0.02} = 65 \text{ Ohm}$$

Vì nhu cầu dùng LED để báo nguồn, nên không cần LED quá sáng, vì vậy để nâng cao tuổi thọ cho LED cũng như tốt cho mắt trong quá trình làm việc nên trong thực tế chúng tôi chọn $R = 1.2 \text{ KOhm}$.

❖ Sơ đồ nguyên lý



CHƯƠNG 4. THI CÔNG HỆ THỐNG

Hình 3.14. Khối nguồn trong nguyên lý mạch điều khiển

❖ **Giải thích sơ đồ nguyên lý**

- Nguồn cung cấp từ nguồn tổ ong 5V – 10A đi qua lọc nhiễu sẽ được cấp cho mạch ổn áp dùng 7805 – 5V, phần công suất lớn chủ yếu ở nhánh trên cùng của sơ đồ và chịu bởi transistor công suất Q3. Đôi với phần điện áp 3.3V, nguồn cung cấp cũng dùng nguồn tổ ong 5V – 10A qua lọc nhiễu đến mạch ổn áp AMS1117 – 3.3V, với nguyên lý mạch ổn áp tuyển tính nối tiếp như sơ đồ nguyên lý Hình 3.2., sẽ cho điện áp ngõ ra ổn áp 3.3V.



Hình 3.15. Nguồn tổ ong 5V 10A

CHƯƠNG 4. THI CÔNG HỆ THỐNG

CHƯƠNG 4. THI CÔNG HỆ THỐNG

4.1. GIỚI THIỆU

Sau khi tính toán và thiết kế chúng tôi đã thi công phần cơ khí cánh tay, mạch điều khiển và mạch trung tâm, phần mềm giao diện người dùng và phần mềm nhận dạng vật thể với các yêu cầu đã được đặt ra và nêu lên ở phần tính toán thiết kế.

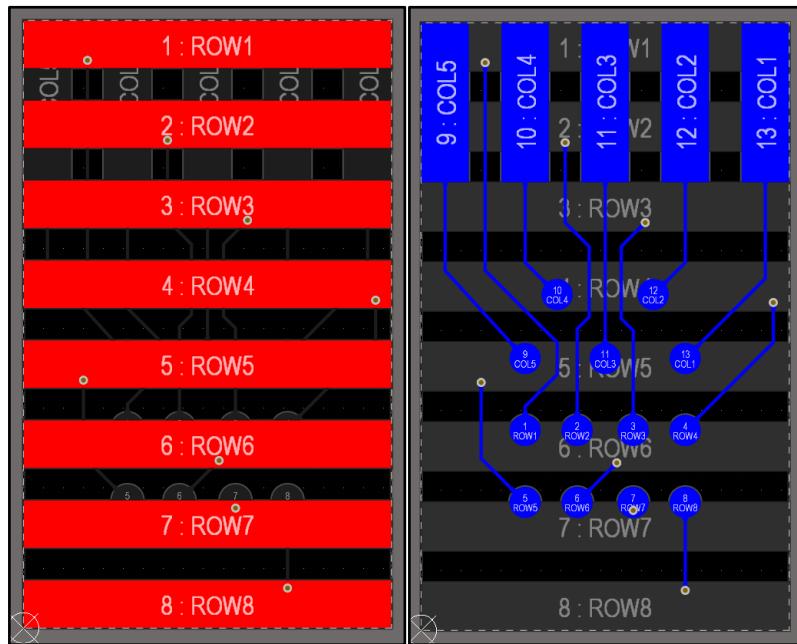
4.2. THI CÔNG HỆ THỐNG

4.2.1. Thi công board mạch

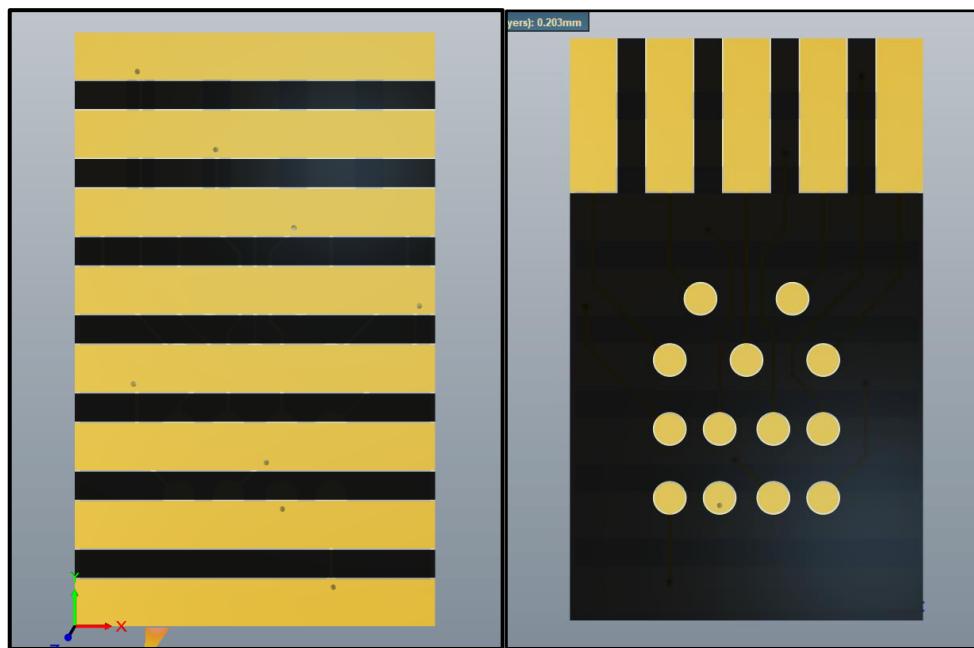
Nhằm thích hợp với yêu cầu đặt ra đồng thời linh hoạt trong việc thay đổi và thay thế, chúng tôi chia mạch điều khiển trung tâm thành 2 mạch với đặc tính và công dụng của chúng: mạch điều khiển chính và mạch điều khiển phụ. Mạch điều khiển chính có chức năng xuất tín hiệu điều khiển ma trận cảm biến từ đó đọc tín hiệu về phân tích và xử lý.Thêm vào đó, mạch điều khiển chính còn có thể nhận tín hiệu từ khói nút nhấn, khói hiển thị, xuất tín hiệu điều khiển ma trận khói cảm biến, giao tiếp dữ liệu với khói giao diện. Bên cạnh đó, mạch điều khiển phụ là phần công suất lớn hơn có chức năng nhận tín hiệu điều khiển từ khói xử lý trung tâm và xuất tín hiệu điều khiển khói động cơ.

a. Mạch PCB cho cảm biến ở các ngón tay

CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.1. Hình ảnh 2D mặt trước và sau của PCB cho cảm biến xúc giác ở các ngón tay trên Altium



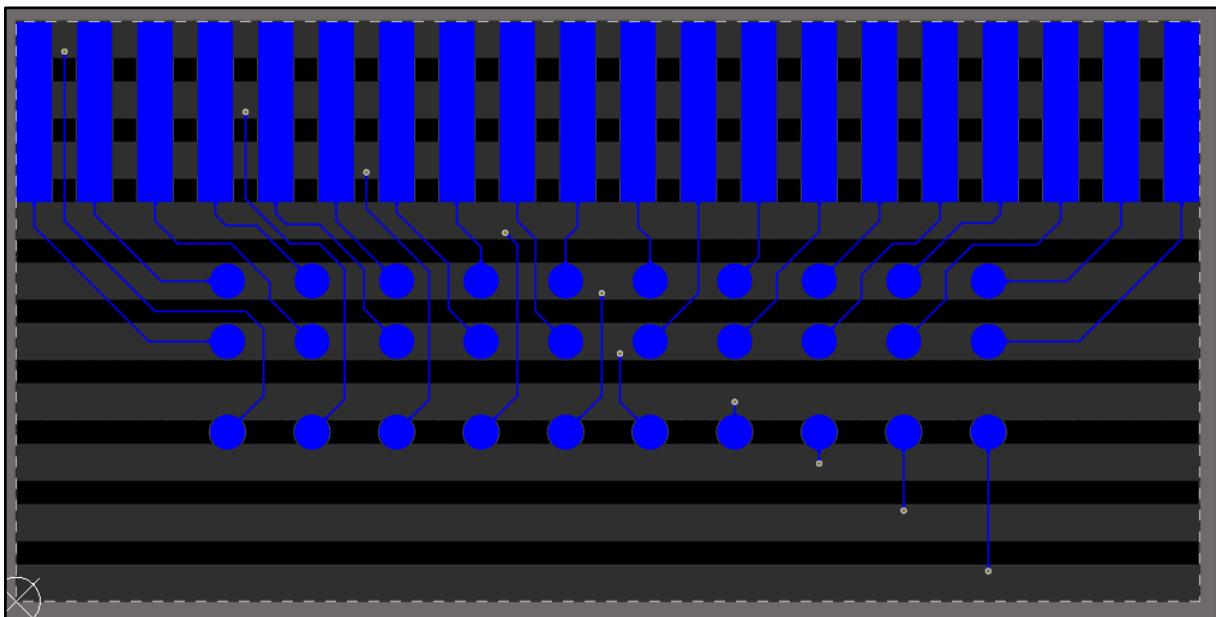
Hình 4.2. Hình ảnh 3D mô phỏng mặt trước và sau của PCB cho cảm biến xúc giác ở các ngón tay trên Altium

CHƯƠNG 4. THI CÔNG HỆ THỐNG

b. Mạch PCB cho cảm biến ở lòng bàn tay

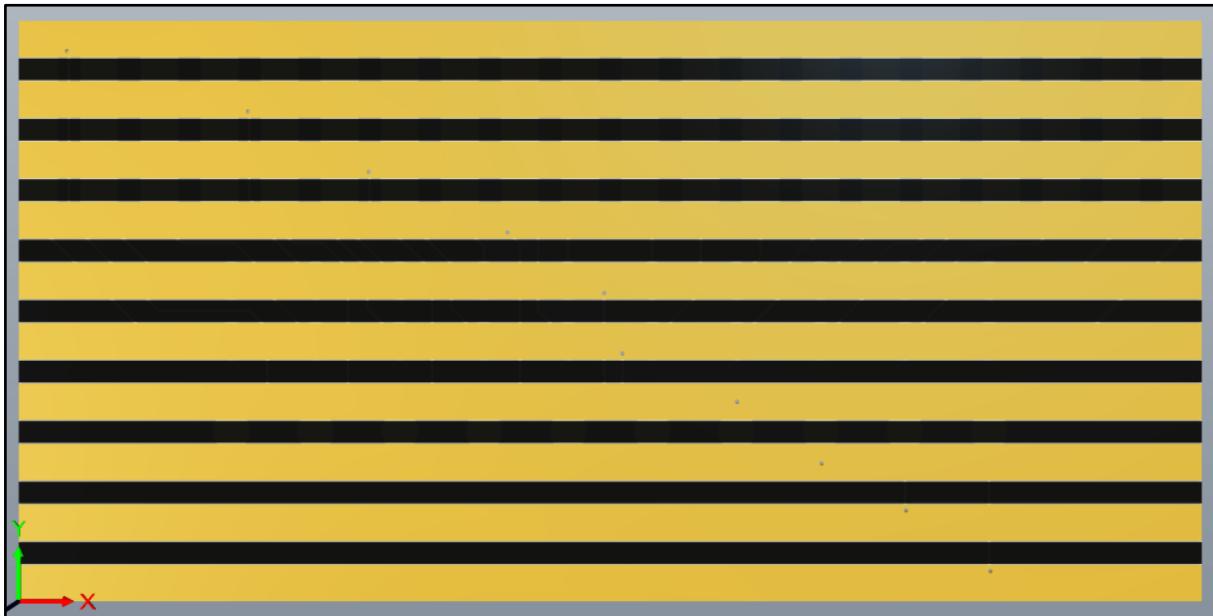


Hình 4.3. Hình ảnh 2D mặt trước của PCB cho cảm biến xúc giác ở lòng tay trên Altium

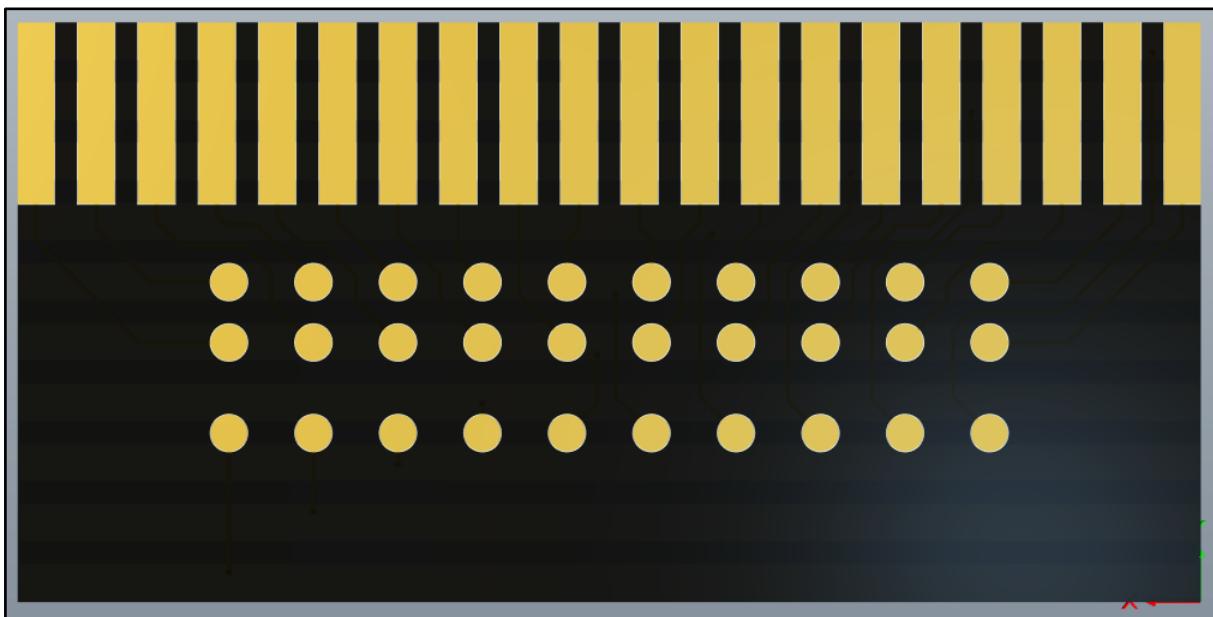


Hình 4.4. Hình ảnh 2D mặt sau của PCB cho cảm biến xúc giác ở lòng tay trên Altium

CHƯƠNG 4. THI CÔNG HỆ THỐNG



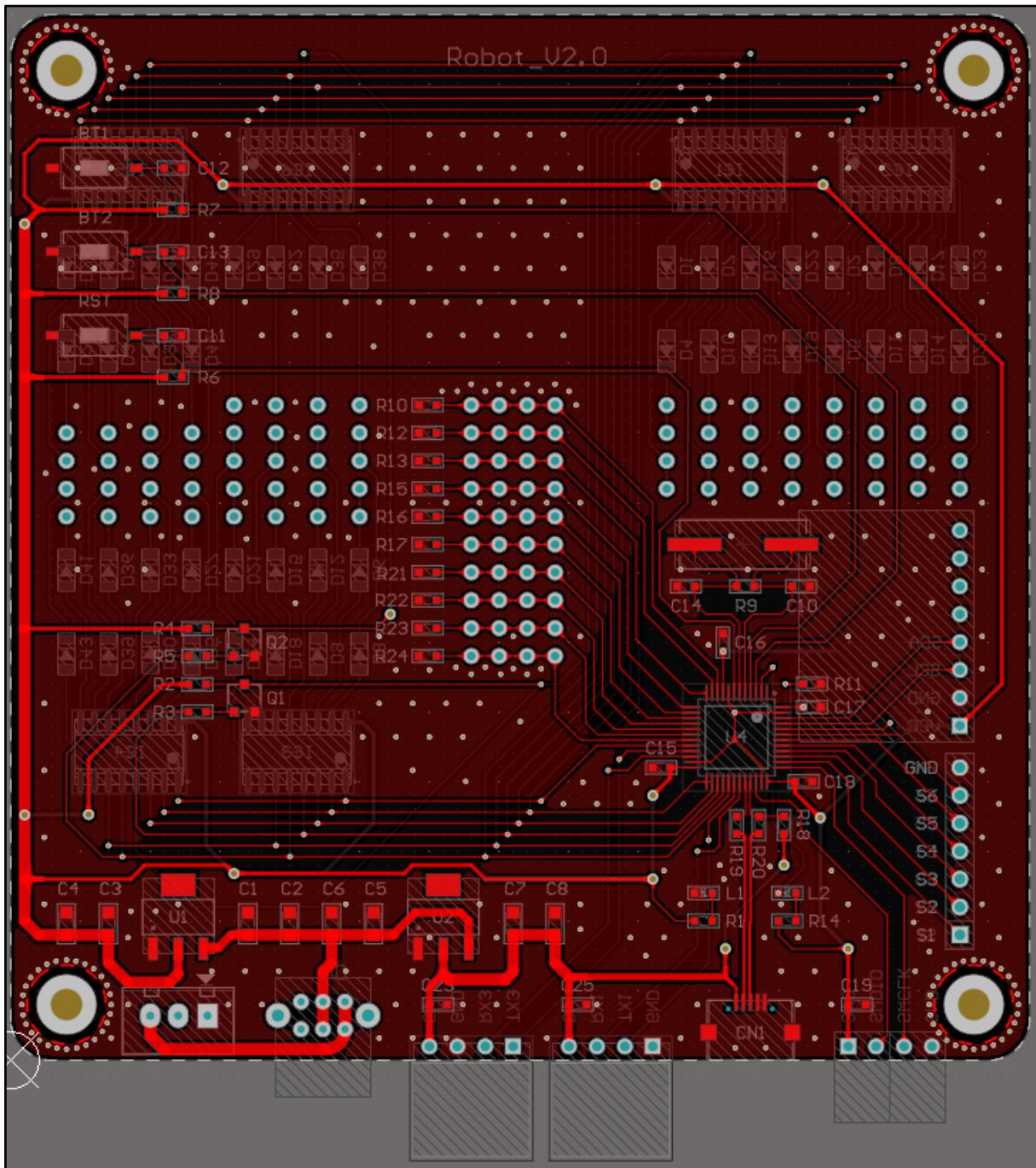
Hình 4.5. Hình ảnh 3D mô phỏng mặt trước của PCB cho cảm biến xúc giác ở lòng tay trên Altium



Hình 4.6. Hình ảnh 3D mô phỏng mặt sau của PCB cho cảm biến xúc giác ở lòng tay trên Altium

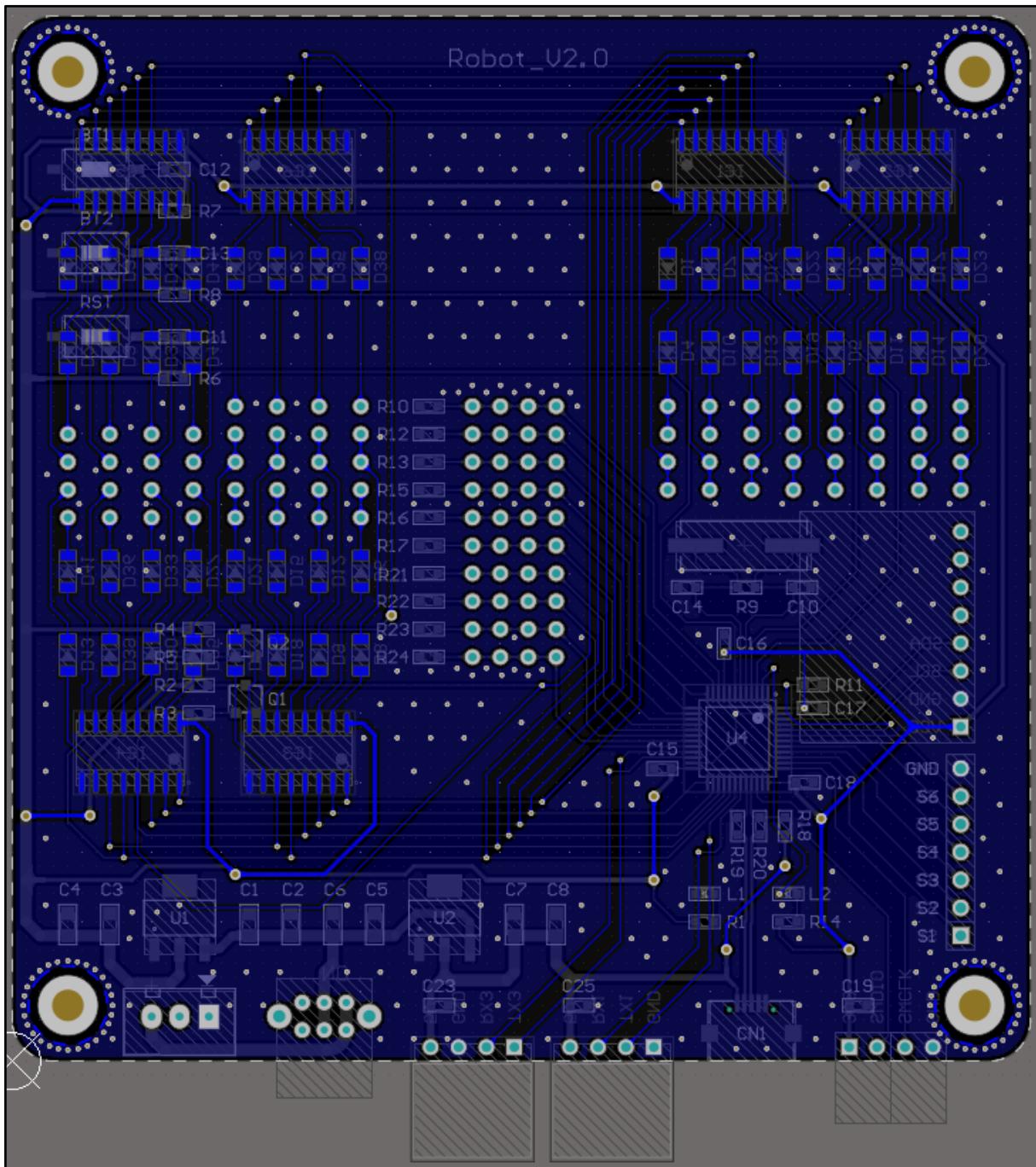
CHƯƠNG 4. THI CÔNG HỆ THỐNG

c. Mạch điều khiển trung tâm chính



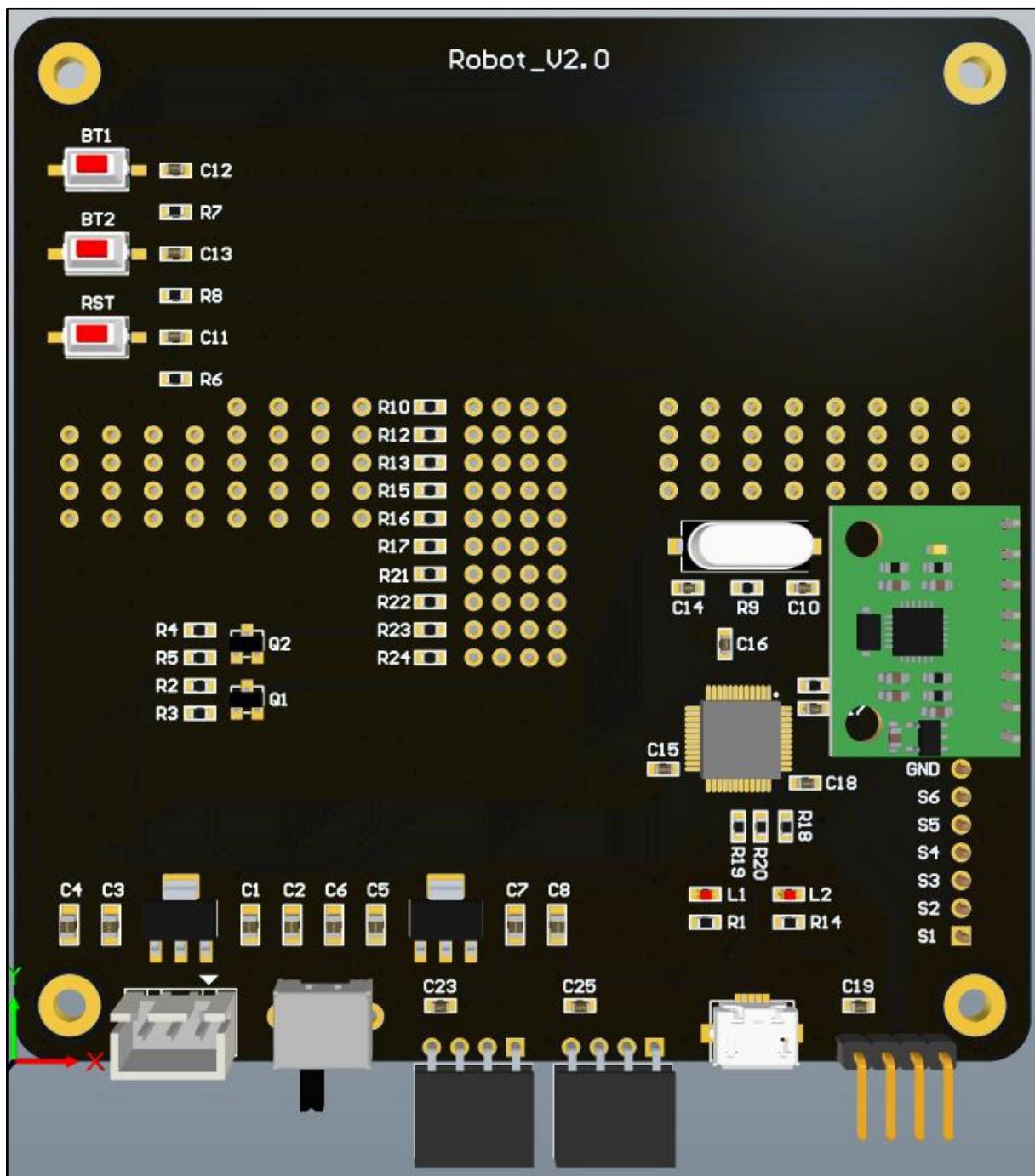
Hình 4.7. Hình ảnh 2D mặt trước của mạch điều khiển chính trên Altium

CHƯƠNG 4. THI CÔNG HỆ THỐNG



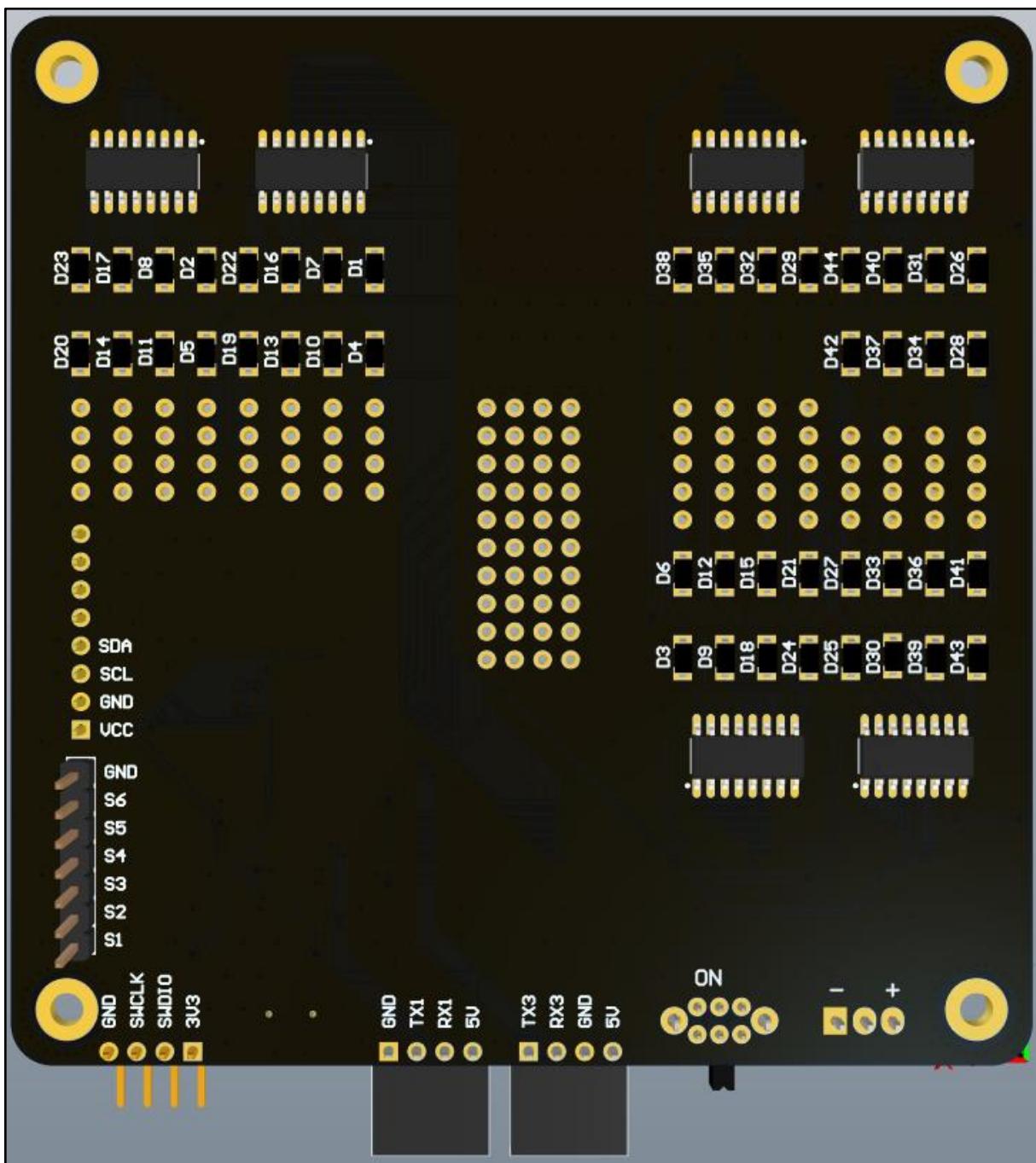
Hình 4.8. Hình ảnh 2D mặt sau của mạch điều khiển chính trên Altium

CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.9. Hình ảnh 3D mô phỏng mặt trước của mạch điều khiển chính trên Altium

CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.10. Hình ảnh 3D mô phỏng mặt sau của mạch điều khiển chính trên Altium

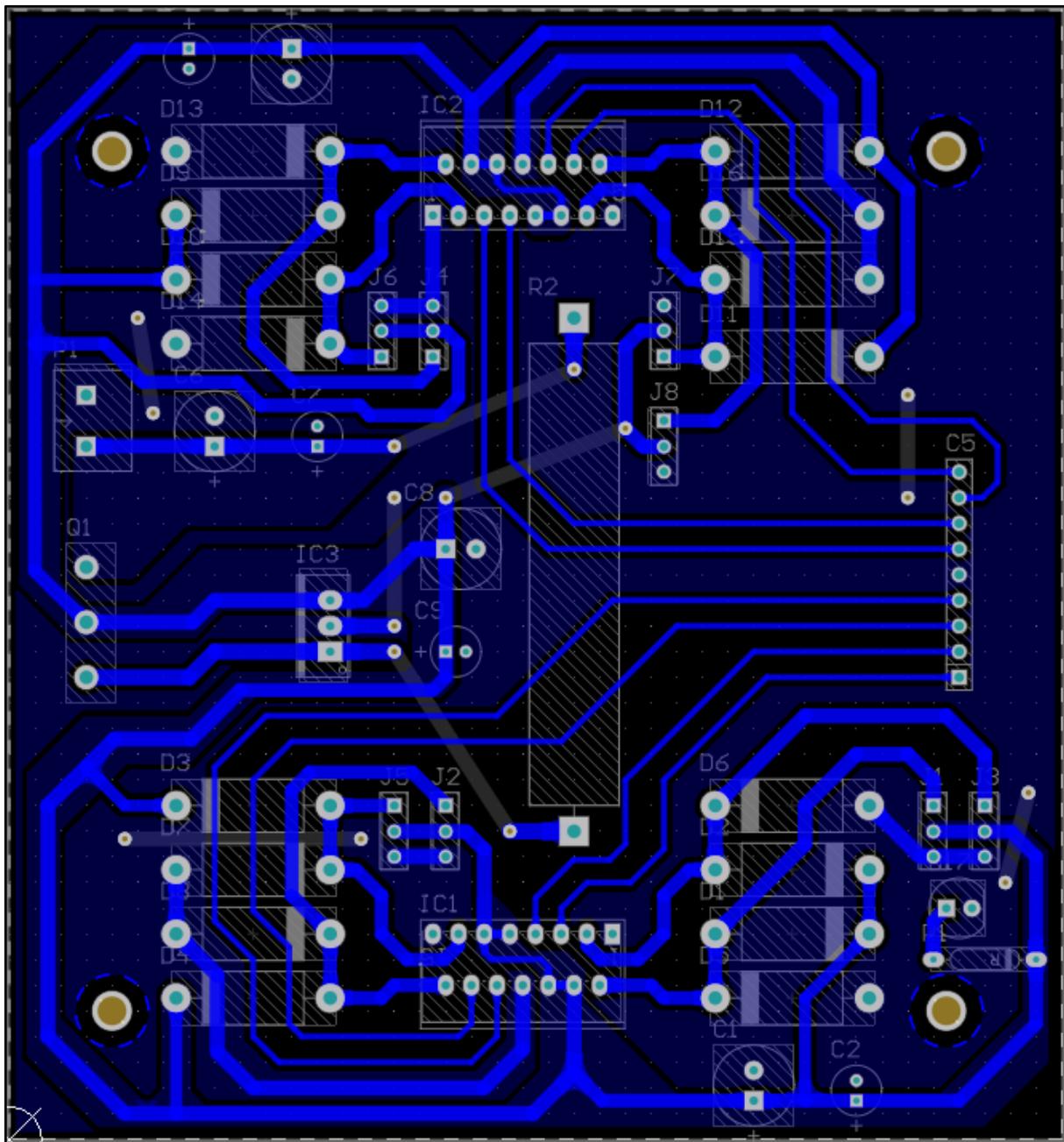
CHƯƠNG 4. THI CÔNG HỆ THỐNG

Bảng 4.1. Danh sách linh kiện cho mạch trung tâm

STT	ID	Tên sản phẩm	Đơn vị tính	Số lượng mua	Đơn giá (VND)	Thành tiền (VND)
1	713	STM32F103C8T6	Cái	1	229.000	229.000
2	3100	Điện Trở 10 KOhm 0603 1%	Gói(50con)	1	2.300	2.300
3	540	Thạch Anh 8Mhz 2 Chân SMD HC49-S	Cái	1	1.900	1.900
4	4893	Tụ Gỗm 0603 20pF 50V	Gói(10con)	1	2.070	2.070
5	3120	Điện Trở 1 MOhm 0603 1%	Gói(50con)	1	2.300	2.300
6	5178	Tụ Gỗm 0603 100nF (0.1uF) 25V	Gói(10con)	1	2.180	2.180
7	262	Tụ Gỗm 0805 100nF (0.1uF) 50V	Gói(10con)	1	2.300	2.300
8	266	Tụ Gỗm 0805 10uF 6.3V	Con	2	1.300	2.600
9	1934	74HC238D IC Decoder/Demultiplexer, 16-SOIC	Cái	6	5.800	34.800
10	11423	IN5819 SMD Diode Chính Lưu 1A 40V	Cái	44	470	20.680
11	2762	Điện Trở 4.7 KOhm 0603 1%	Gói(50con)	1	2.300	2.300
12	846	2SC1815-HF Transistor NPN 50V 0.15A 3 Chân SO-23	Cái	2	420	840
13	3090	Điện Trở 1 KOhm 0603 1%	Gói(50con)	1	2.300	2.300
14	3091	Điện Trở 1.2 KOhm 0603 1%	Gói (50pes)	1	2.300	2.300
15	31616	LED Đỏ 0603 Dán SMD Trong Suốt Harvatek	Cái	1	350	350
16	31620	LED Xanh Dương 0603 Dán SMD Trong Suốt Harvatek	Cái	1	350	350
17	383	AMS1117-5.0V IC Ôn Áp 5V 1A	Cái	1	3.500	3.500
18	1025	AMS1117-3.3V IC Ôn Áp 3.3V 1A	Cái	1	3.500	3.500
19	15137	SK-22D07 Công Tắc Trượt 6 Chân Xuyên Lỗ ON-ON 2P2T 1A	Cái	1	920	920
20	2592	Kim Bấm Dây XH2.54 Đầu Cái 2.54mm Mạ Thiếc Dây 26AWG	Gói(10cái)	1	1.200	1.200
21	5287	XH-3Y Đầu Cái XH2.54 2.54mm 3 Chân 1 Hàng	Cái	1	140	140
22	2115	Đầu XH 2.54mm 3 Chân Thẳng Xuyên Lỗ	Cái	1	300	300
23	13332	Hàng Rào Đực Đơn 2.54mm 40 Chân Cong 1 Hàng Cao 6mm Xuyên Lỗ	Cái	1	1.200	1.200
24	985	Nút Nhấn 3x6mm Cao 2.5mm 2 Chân SMD Đầu Đỏ	Gói(2cái)	3	1.030	3.090
25	3074	Điện Trở 20 Ohm 0603 1%	Gói(50con)	1	2.300	2.300
26	2896	Cổng USB Micro-B 2.0 Đầu Cái 5 Chân SMD V3	Cái	1	800	800
Tổng tiền						325.520

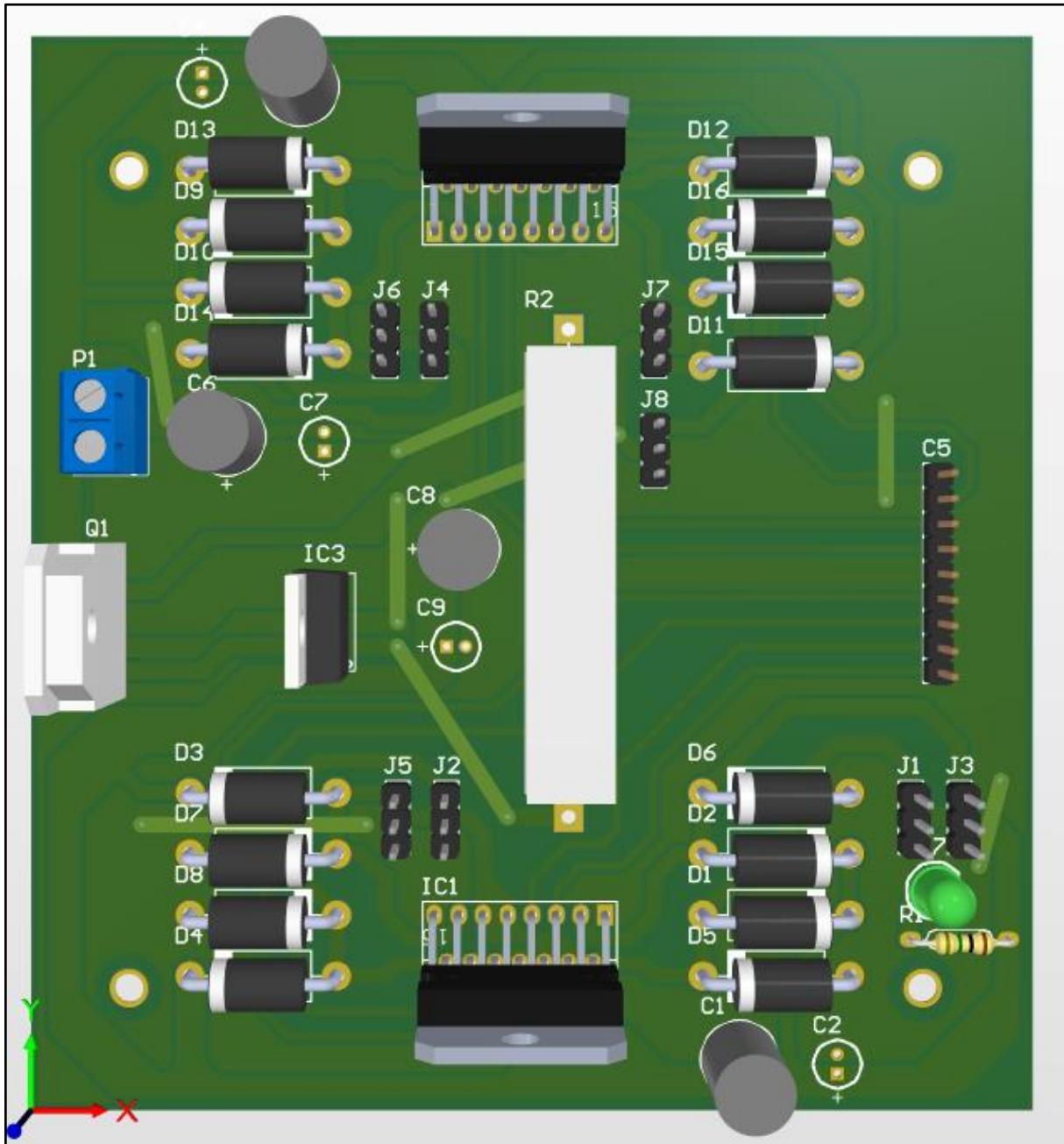
d. Mạch điều khiển

CHƯƠNG 4. THI CÔNG HỆ THỐNG



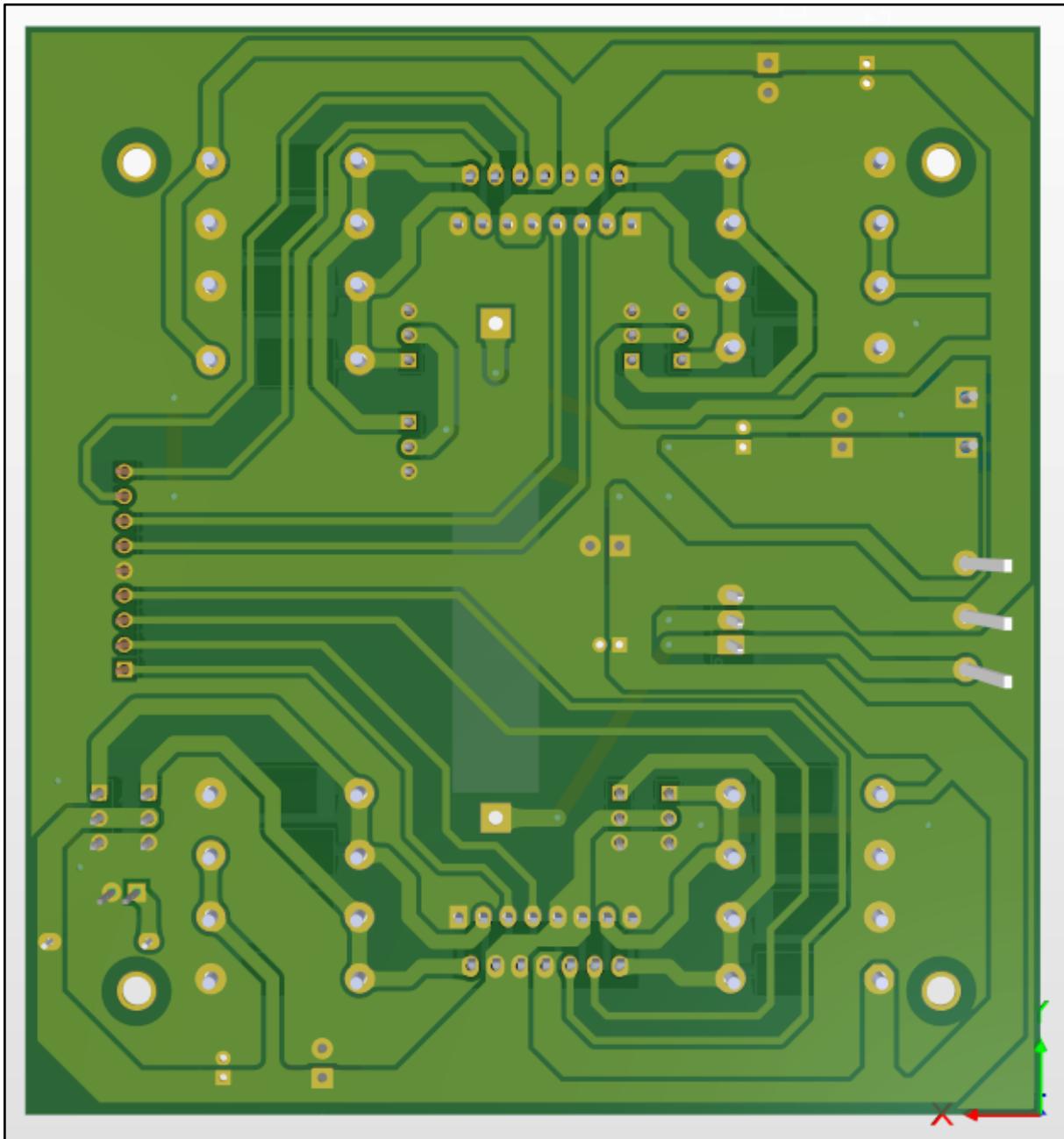
Hình 4.11. Hình ảnh 2D mặt sau của mạch điều khiển phụ trên Altium

CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.12. Hình ảnh 3D mô phỏng mặt trước của mạch điều khiển phụ trên Altium

CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.13. Hình ảnh 3D mô phỏng mặt sau của mạch điều khiển phụ trên Altium

CHƯƠNG 4. THI CÔNG HỆ THỐNG

Bảng 4.2. Danh sách linh kiện cho mạch điều khiển

STT	ID	Tên sản phẩm	Đơn vị tính	Số lượng mua	Đơn giá (VND)	Thành tiền (VND)
1	887	1N5401 Diode Chính Lưu 3A 100V	Cái	16	1.200	19.200
2	1902	L298N IC Điều Khiển Động Cơ Có Chổi Than 46V 2A 15-Multiwatt	Cái	2	44.000	88.000
3	760	Tụ Hoá 100uF 50V 8x12.5mm Xuyên Lõi	Cái	4	750	3.000
4	4425	Tụ Hoá 0.1uF 50V 5x11mm Xuyên Lõi	Cái	4	300	1.200
5	5881	BUW42AP Transistor PNP 400V 15A 3 Chân TO-3P	Cái	1	25.000	25.000
6	20346	Điện Trở Sứ 1 Ohm 10W 5%	Cái	1	2.500	2.500
7	694	L7805CV IC Ôn Áp 5V 1.5A	Cái	1	4.600	4.600
8	19460	Điện Trở 510 Ohm 1/4W 5% 4 Vòng Màu	Gói(50con)	1	2.000	2.000
9	1491	LED Đỏ 5mm Chân Dài	Con	1	280	280
10	22930	Hàng Rào Đực Đơn 2.54mm 40 Chân 1 Hàng Cao 11.2mm Xuyên Lõi Loại Tốt	Cái	1	1.400	1.400
11	13332	Hàng Rào Đực Đơn 2.54mm 40 Chân Cong 1 Hàng Cao 6mm Xuyên Lõi	Cái	1	1.200	1.200
12	3393	Nhôm Tân Nhiệt 25x24x16 Màu Đen	Cái	3	2.900	8.700
13	6189	Nhôm Tân Nhiệt 20x15x10 Hàn PCB Màu Đen	Cái	1	2.100	2.100
Tổng tiền						159.180

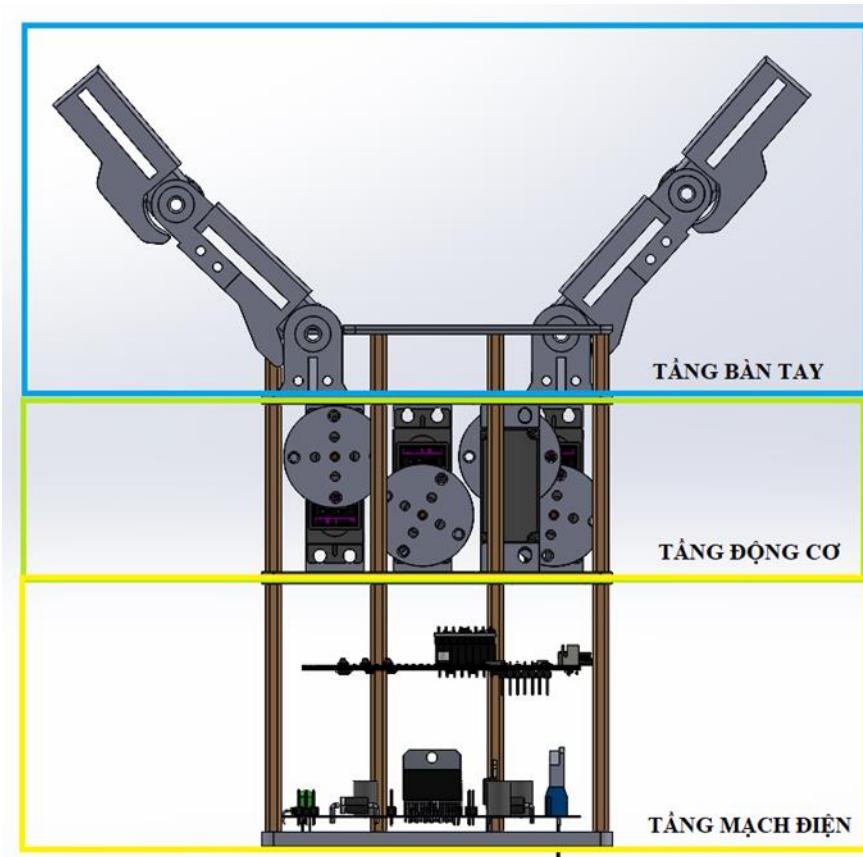
4.2.2. Thi công mô hình

Bàn tay robot được thiết kế dựa trên các yêu cầu trên với 3 tầng như sau

- Tầng bàn tay bao gồm: khớp ngón tay, lòng bàn tay
- Tầng động cơ: phần mở rộng của động cơ, phần cố định động cơ
- Tầng mạch điện: phần cố định mạch điện

Ngoài 3 tầng chính ra thì có một phần nhám cố định cả động cơ

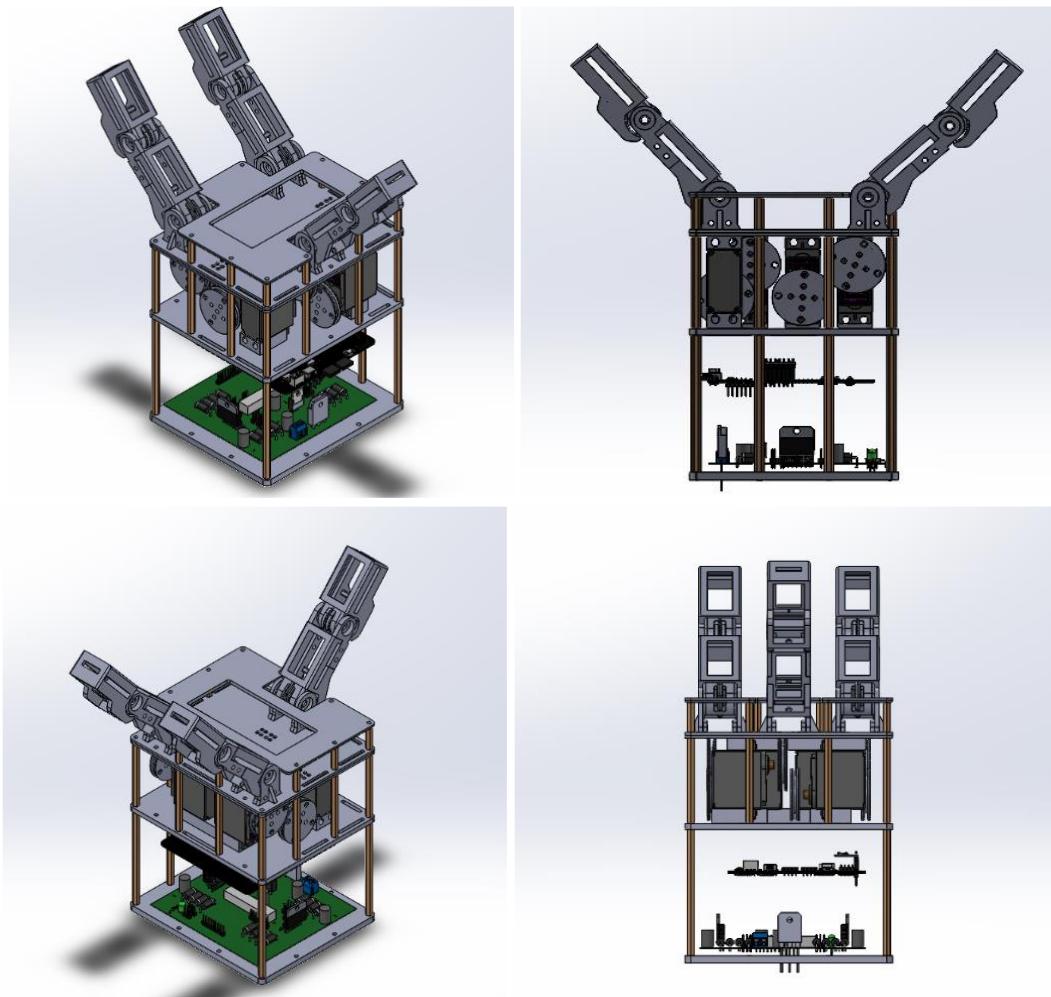
CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.14. Phân chia các tầng của hệ thống cơ khí

Hệ thống cánh tay robot qua các góc nhìn khác nhau trên phần mềm Solidworks

CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.15. Tổng quan cánh tay

Như vậy, cánh tay robot được chia thành những tầng chính là tầng bàn tay, tầng động cơ và tầng mạch điện. Ta sẽ trình bày cấu tạo và thành phần của robot chi tiết ở những phần sau.

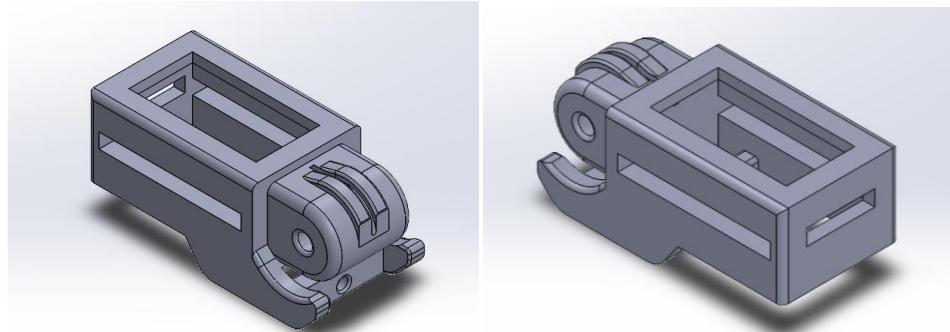
a. Tầng bàn tay

❖ Khớp cánh tay

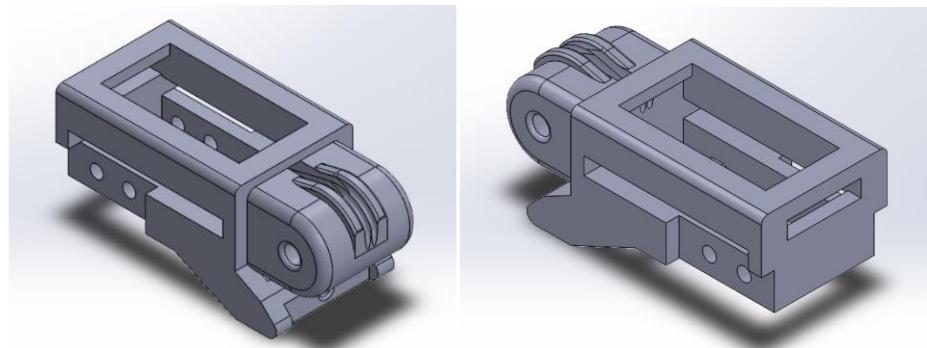
Tầng bàn tay gồm 2 thành phần chính là các khớp tay và lòng bàn tay. Ở đè tài này, những cảm biến hoàn chỉnh sẽ được đặt trên những khớp tay và lòng bàn tay.

CHƯƠNG 4. THI CÔNG HỆ THỐNG

Khớp cánh tay gồm 6 khớp có 2 loại khớp trên và khớp dưới. Một khớp trên và một khớp dưới tạo thành một ngón tay và tổng cộng có 3 ngón tương ứng với 6 khớp.



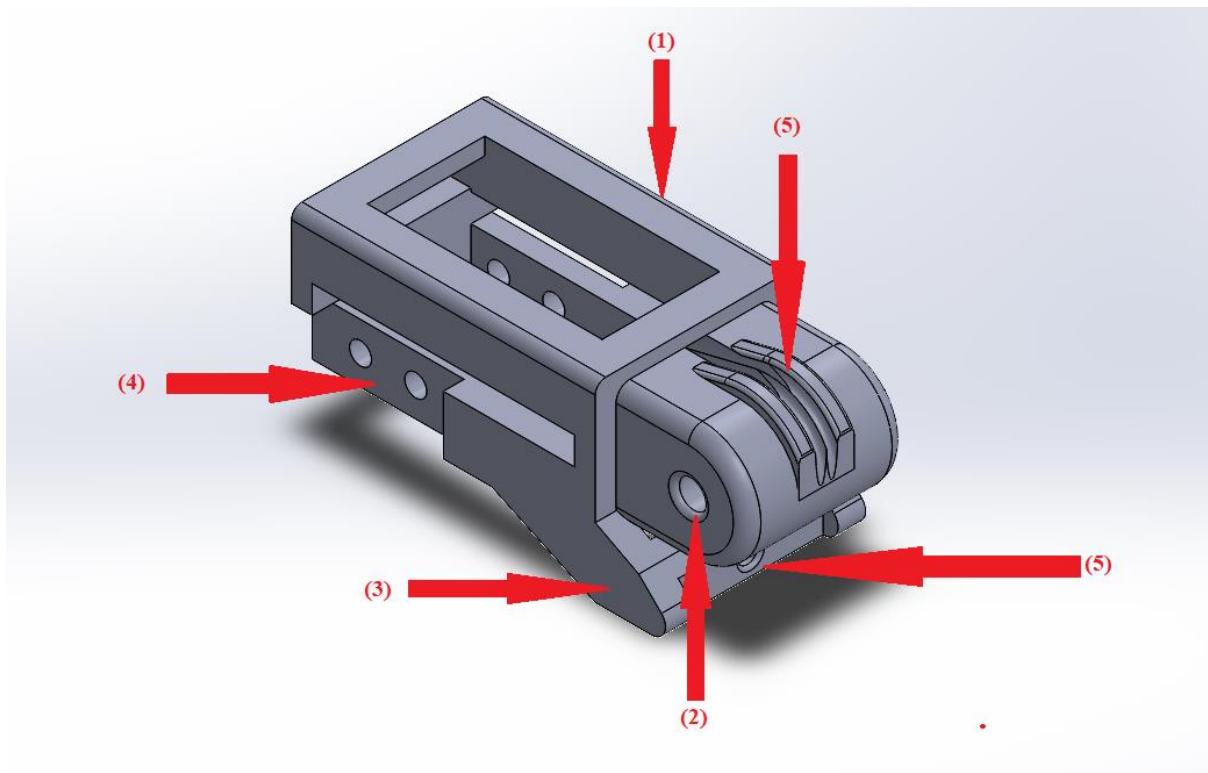
Hình 4.16. Khớp trên của ngón tay robot



Hình 4.17. Khớp dưới của ngón tay robot

Cấu tạo của khớp trên và khớp dưới khá giống nhau, bao gồm:

CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.18. Thành phần chính cấu tạo nên khớp

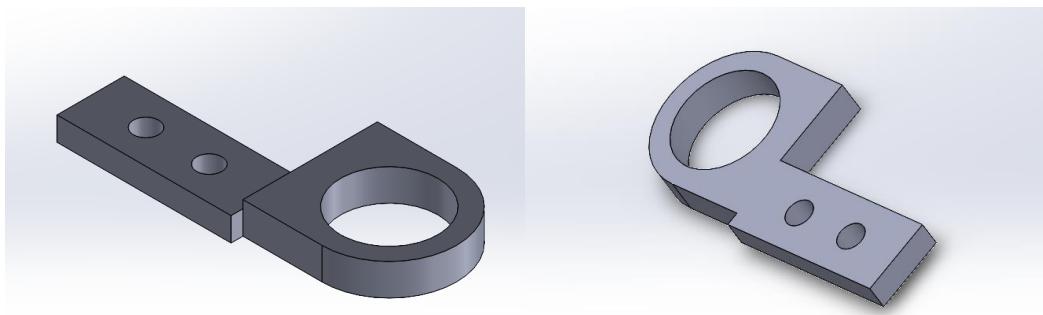
(1) Phần đặt cảm biến: yêu cầu của 2 khớp cần phải đặt được miếng cảm biến mà mình thiết kế. Cần có một ô trống có kích thước đủ lớn đủ lớn ở mặt trên dây điện đi từ tần mạch điện đi đến để kết nối với cảm biến.

(2) Phần xoay của khớp cánh tay: Ở phần đầu của khớp ta đục một lỗ tròn 4mm, khi đó thông qua trụ sắt và bạc đạn nhắm khớp xoay được linh hoạt

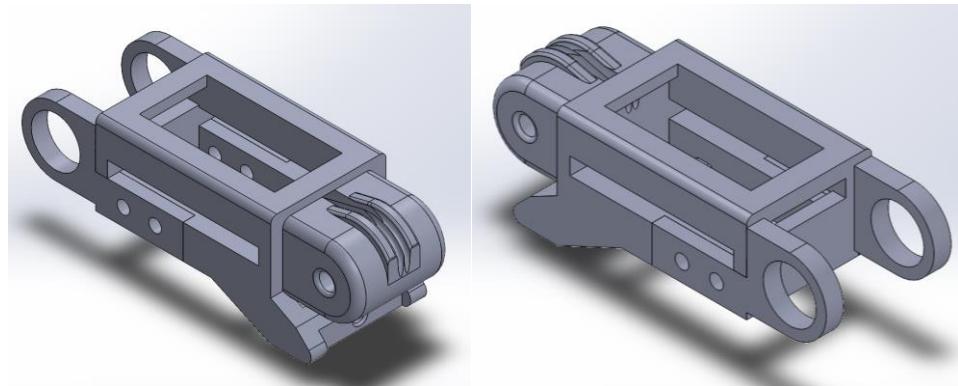
(3) Phần chắn của khớp: khớp khi xoay ở một khoảng góc có góc tối đa và góc tối thiểu vì vậy cần có phần chắn để tránh khớp có thể xoay quá nhiều không phù hợp với những khớp tay của robot và robot này cũng không cần thiết phải như thế.

(4) Phần nối giữa khớp trên và khớp dưới (chỉ có ở khớp dưới): Khớp trên và khớp dưới được nối thông qua thanh nối như sau:

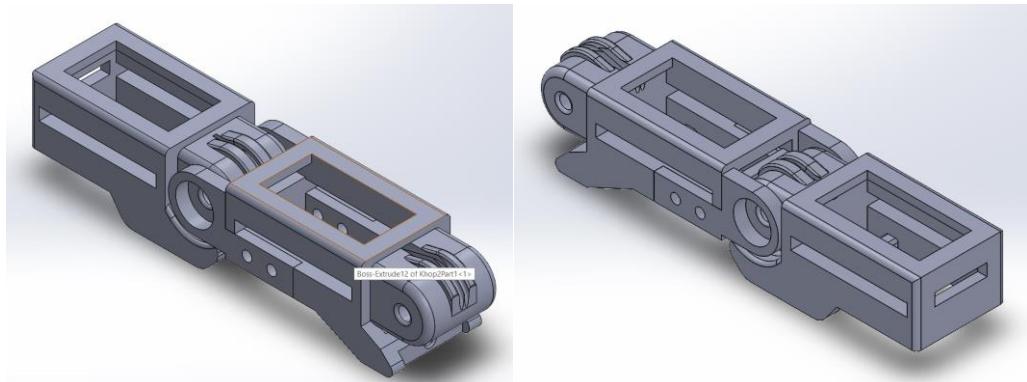
CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.19. Khớp nối của 2 khớp



Hình 4.20. Khớp dưới cố định chung với khớp nối



Hình 4.21. Nối 2 khớp thông qua khớp nối

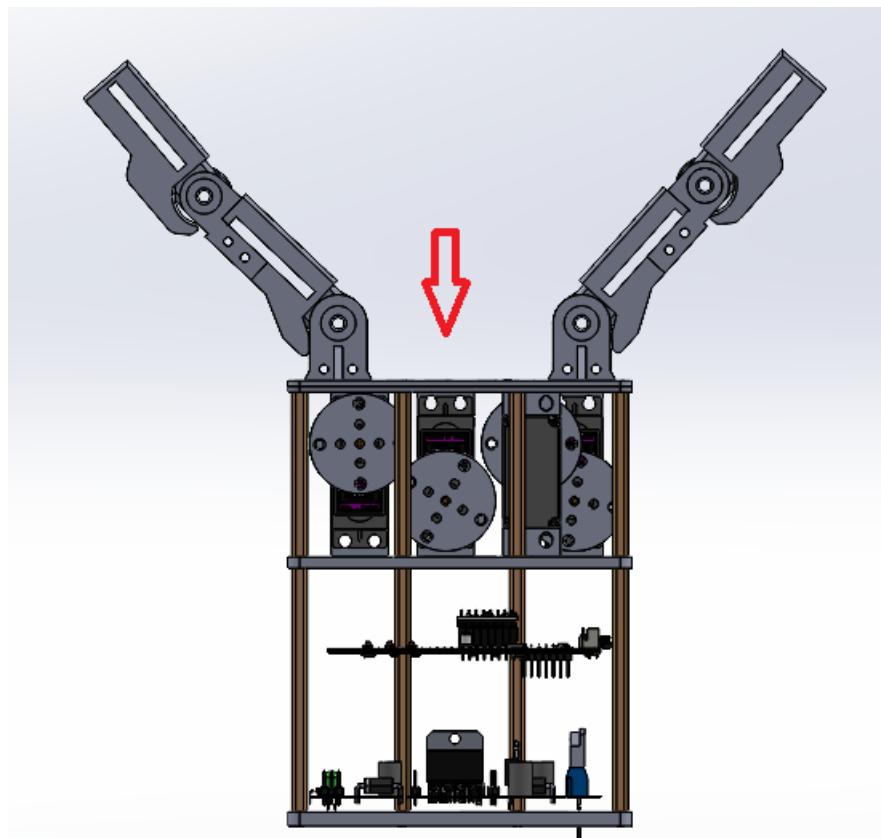
(5) Phần đường dây kéo: hệ thống động cơ sử dụng cho mô hình là động cơ kéo dây các khớp nên cần 2 đường đi dây cho các khớp.

CHƯƠNG 4. THI CÔNG HỆ THỐNG

❖ Lòng bàn tay:

Cấu tạo của lòng bàn tay đơn giản hơn so với khớp cánh tay rất nhiều chỉ cần kích thước chỉ cần kích thước cần đặt được cảm biến và có một ô trống đủ lớn ở mặt trên dây điện đi từ tần mạch điện đi đến để kết nối với cảm biến.

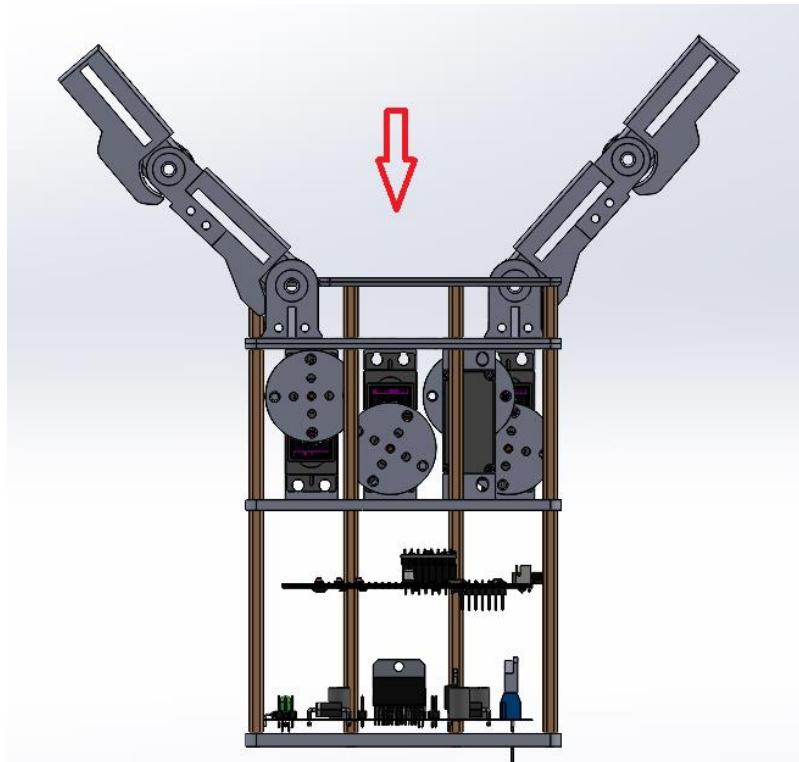
Khi ban đầu thiết kế lòng bàn tay được định là nền của tầng một:



Hình 4.22. Vị trí dự định của lòng bàn tay

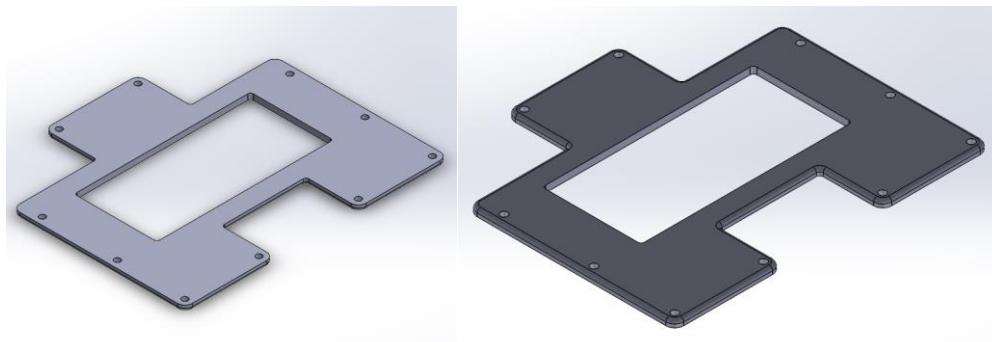
Nhưng khi mà đặt vật thể lên với nền như vậy ta sẽ bị vướng 2 bên bởi những trụ đỡ 2 khớp, vì vậy ta chọn nâng cao lòng bàn tay lên cao để giảm bị vướng bởi những trụ đỡ thấp nhất có thể. Khi đó:

CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.23. Lòng bàn tay khi được nâng lên

Lòng bàn tay được thiết kế như sau:



Hình 4.24. Lòng bàn tay

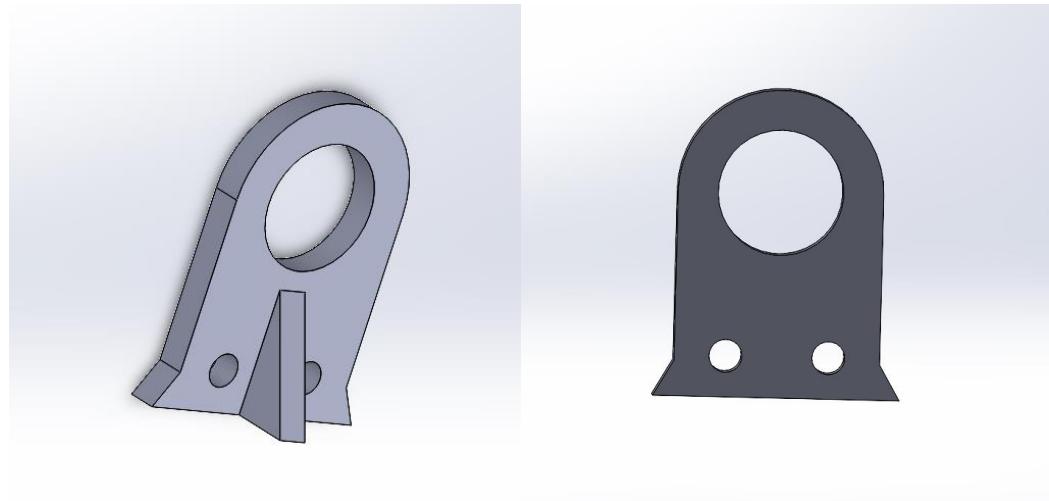
Lòng bàn tay được cố định và chịu lực thông qua các thanh đồng mà các thanh đồng được bố trí theo biên của lòng bàn tay nên khả năng chịu lực ở bên trong lòng bàn

CHƯƠNG 4. THI CÔNG HỆ THỐNG

tay là rất lớn, nên chọn lòng bàn tay bằng nhôm để tăng độ chịu lực so với khi sử dụng nhựa.

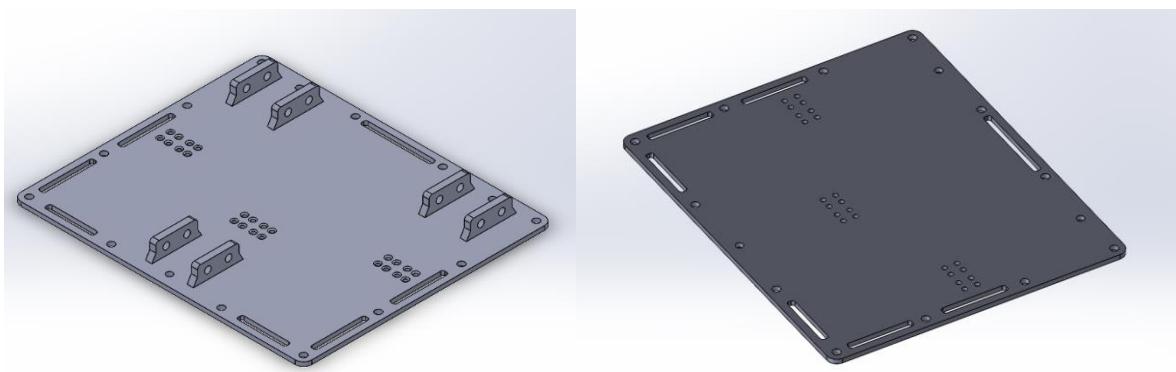
❖ Trụ đỡ các khớp cánh tay

Các khớp đỡ các khớp cánh tay cần đạt được yêu cầu là đỡ được các khớp cánh tay và chịu lực và giảm áp lực lên nền của lòng bàn tay. Ngoài ra cần phải cố định không bị lệch nhằm cánh tay hoạt động tốt và ổn định



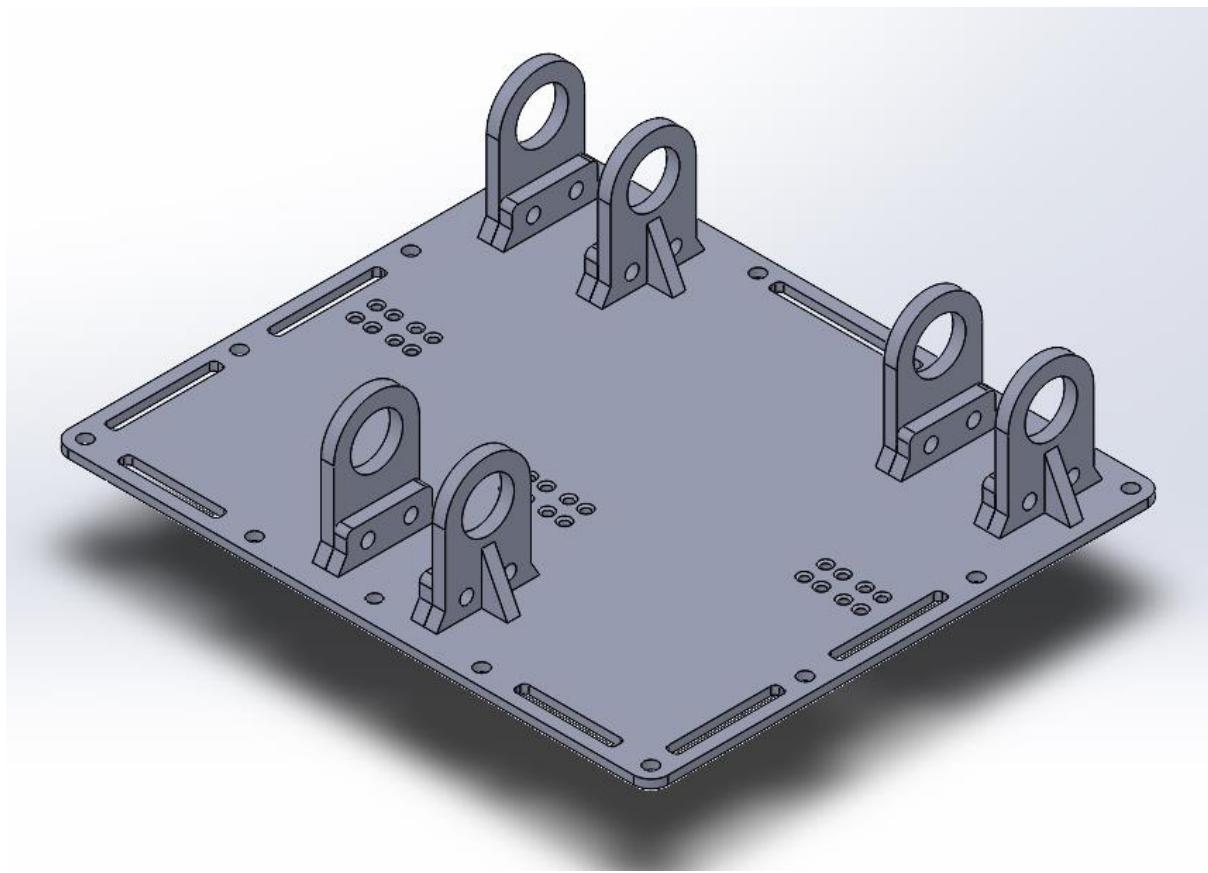
Hình 4.25. Khớp đỡ cả ngón tay của robot

Thiết kế nền cơ sở cho tầng bàn tay



Hình 4.26. Nền của tầng bàn tay

CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.27. Nền của tầng động cơ với các khớp đỡ ngón tay robot

b. Tầng động cơ

Tầng động cơ gồm 2 mặt phẳng chính được thiết kế nhắm từ 2 mặt phẳng gồm mặt trên và mặt dưới ép các động cơ và dùng ốc vít nhằm cố định động cơ (động cơ cần cố định vững chắc khi mà động cơ hoạt động lực khá lớn). Ngoài ra động cơ được bố trí theo kiểu đang xen và ngược chiều.

❖ Địa chỉ động cơ để đặt dây kéo

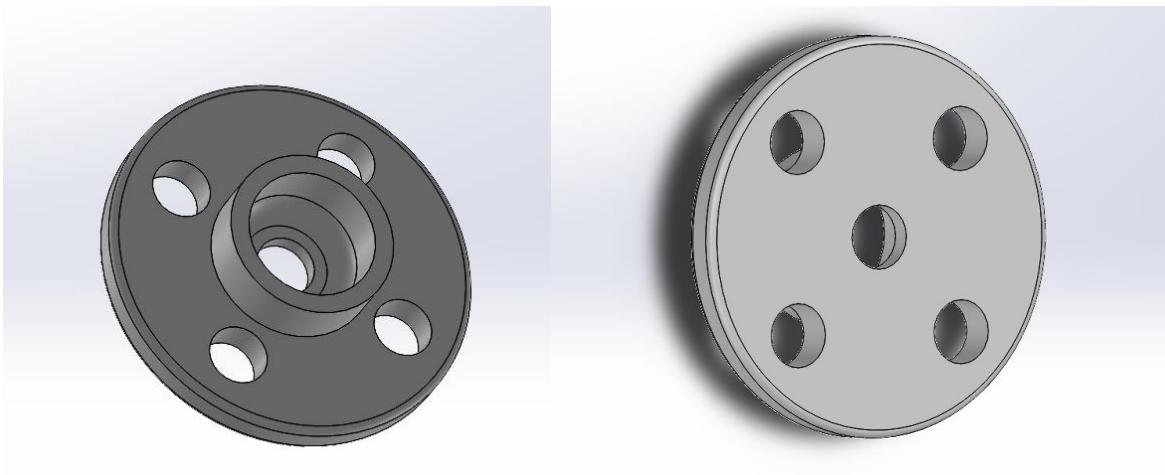
Khi mua động cơ servo, ta sẽ không có được vật dụng kèm theo phù hợp với đề tài nên khi đó ta cần thiết kế riêng cho hệ thống.

CHƯƠNG 4. THI CÔNG HỆ THỐNG

Ta sử dụng một cái đĩa trung gian là đĩa RC Servo, nhằm mục đích làm trung gian cho động cơ và đĩa cần thiết kế.



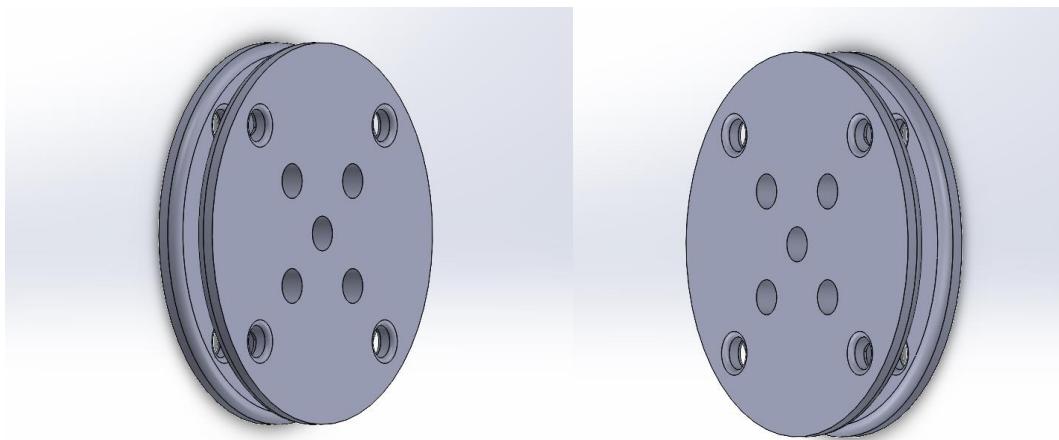
Hình 4.28. Đĩa RC Servo có sẵn trên thị trường



Hình 4.29. Đĩa RC Servo được mô phỏng trên phần mềm Solidworks

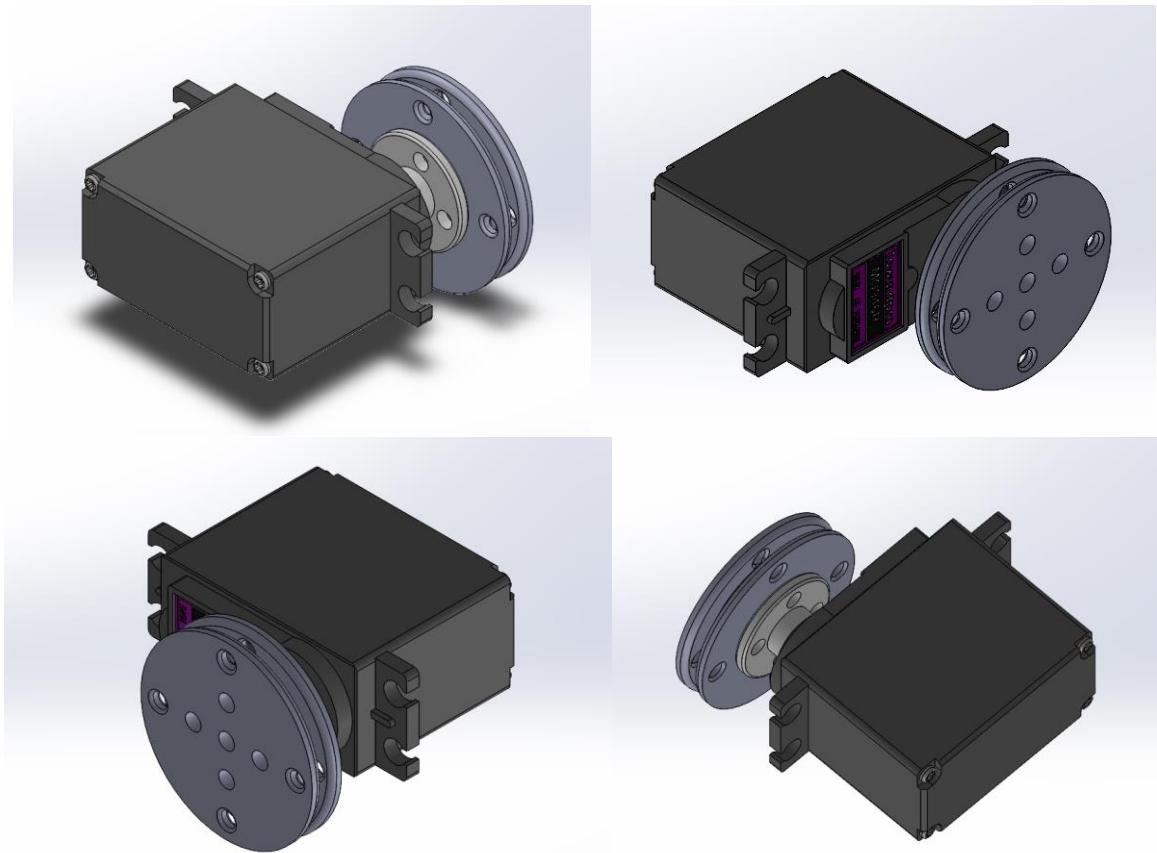
Khi đó ta thiết kế đĩa đính kèm để phụ trách cuộn dây điều khiển các khớp của ngón tay robot như sau

CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.30. Đĩa động cơ phụ trách kéo dây điều khiển ngón tay robot

Lắp ráp vào mô hình

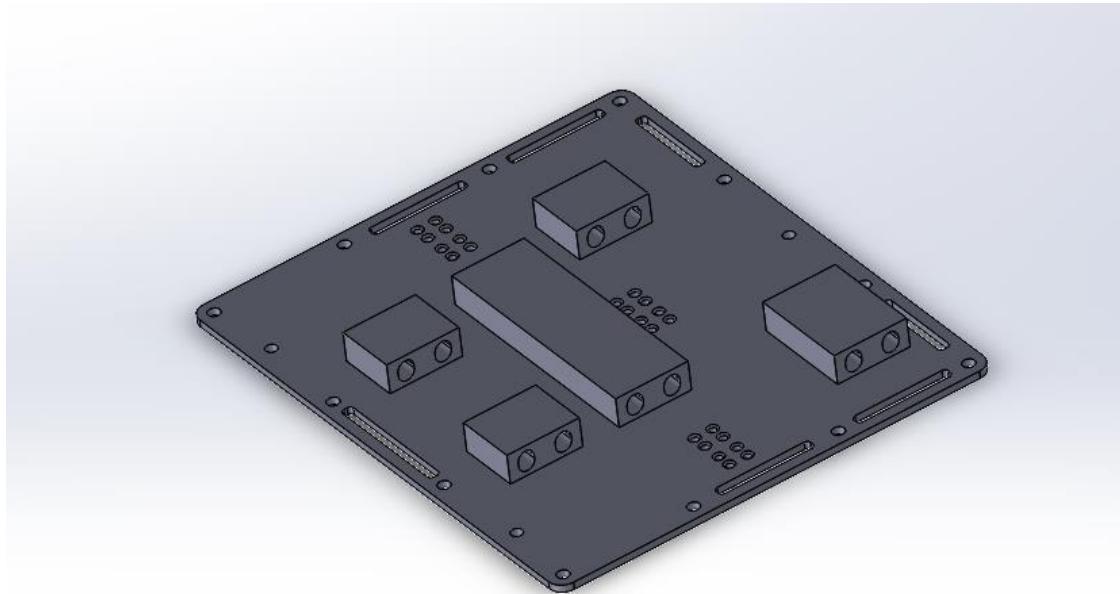


Hình 4.31. Lắp ráp động cơ với phần đĩa để kéo ngón tay robot

CHƯƠNG 4. THI CÔNG HỆ THỐNG

❖ **Mặt trên của tầng động cơ**

Mặt trên của tần động cơ được bố trí để đặt vừa cho phần động cơ và có những lỗ để những cái vít có thể đi qua và đồng thời vít này đi qua lỗ động cơ nhằm cố định động cơ

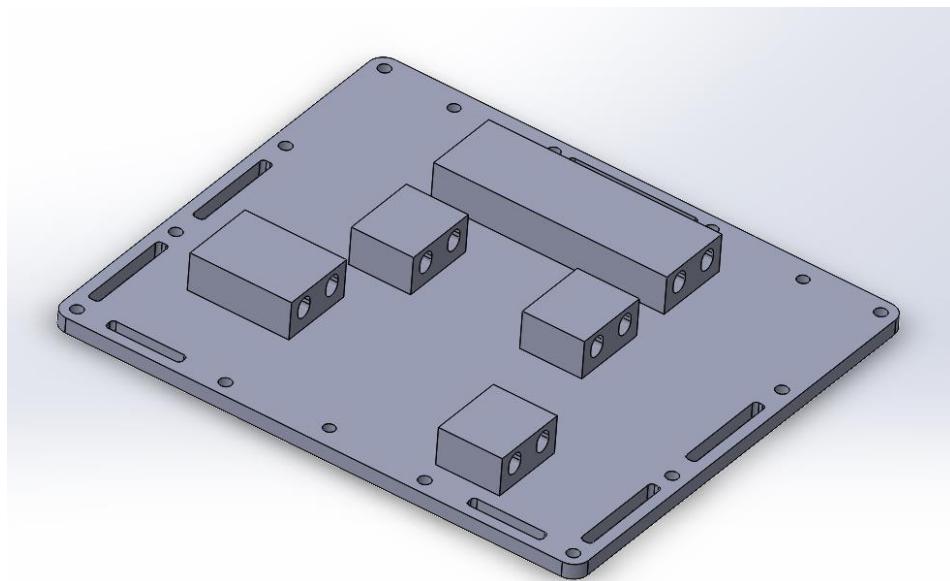


Hình 4.32. Mặt trên của tầng động cơ

❖ **Mặt dưới của tầng động cơ**

Mặt trên của tần động cơ được bố trí để đặt vừa cho phần động cơ và có những lỗ để những cái vít có thể đi qua và đồng thời vít này đi qua lỗ động cơ nhằm cố định động cơ

CHƯƠNG 4. THI CÔNG HỆ THỐNG

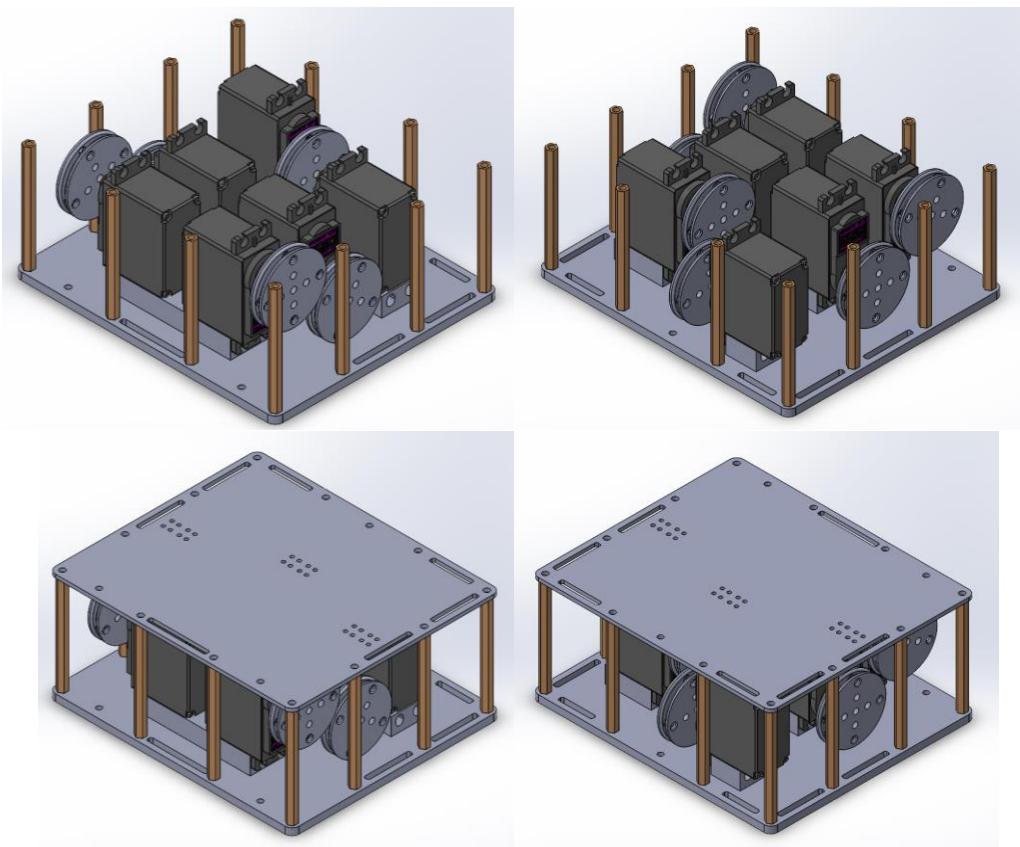


Hình 4.33. Mặt dưới của tầng động cơ

❖ Lắp ghép và phân bố động cơ

Kích thước của Hệ thống phụ thuộc rất lớn vào kích thước tầng động cơ vì vậy động cơ của hệ thống sẽ được bố trí đan xen và chiều khác nhau nhằm mục đích tối ưu độ lớn kích thước của tầng động cơ, và động cơ được bố trí như sau

CHƯƠNG 4. THI CÔNG HỆ THỐNG

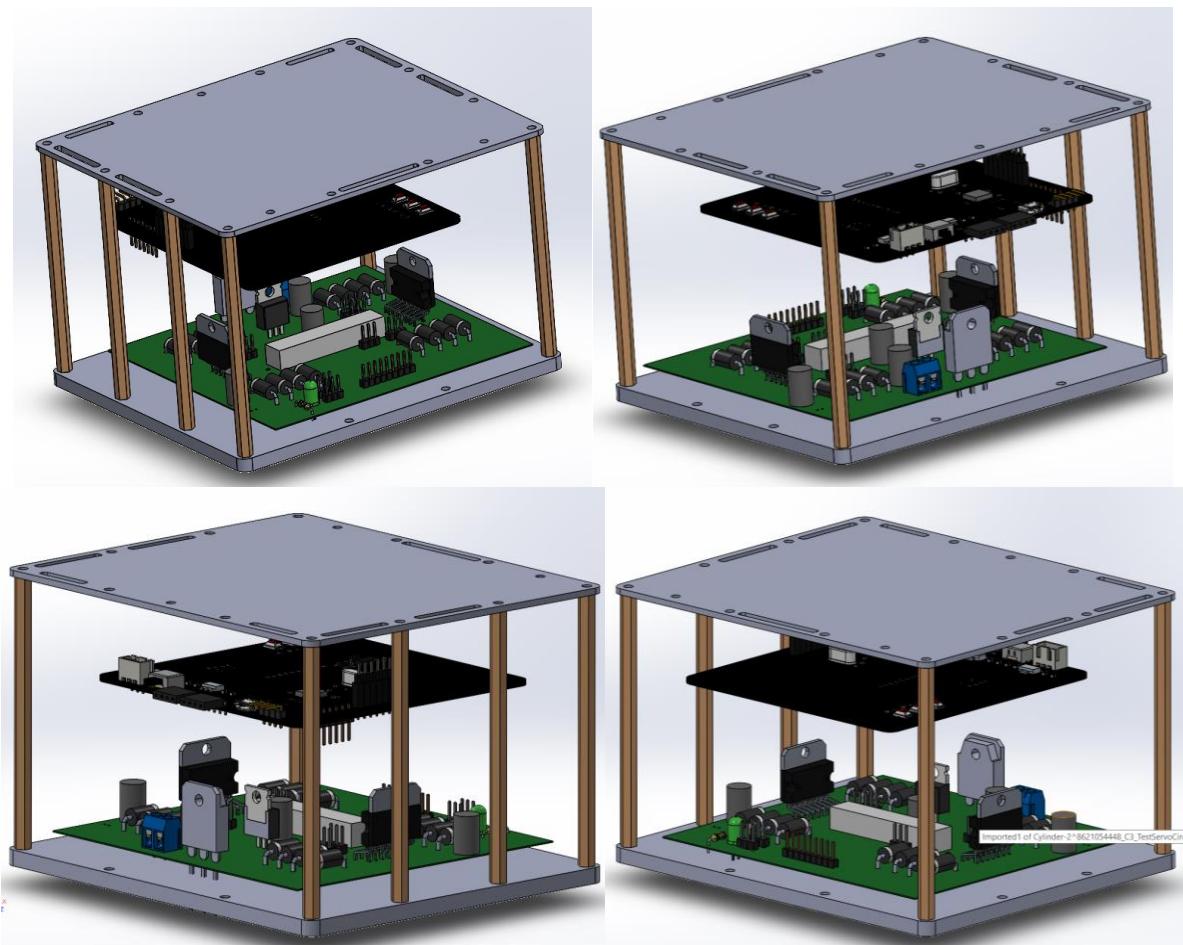


Hình 4.34. Bố trí động cơ ở tầng động cơ của bàn tay robot

c. Tầng mạch điện

Tầng mạch điện là tầng khá đặc thù khi chỉ cần có phần cố định các mạch điện là được, ngoài ra tầng mạch điện có không gian khá lớn bởi vì dây điện ở phần mạch điện rất nhiều và nhiều dây điện đi 2 tầng khác.

CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.35. Bố trí tầng mạch điện của bàn tay robot

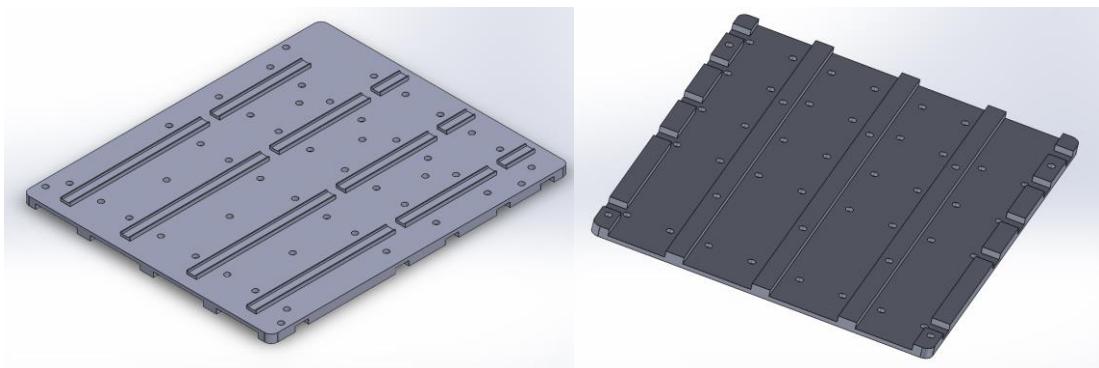
Các mạch điện được được cố định thông qua các trụ đồng và vít

d. Các phần khác của hệ thống:

❖ **Phần nền và cố định cho hệ thống**

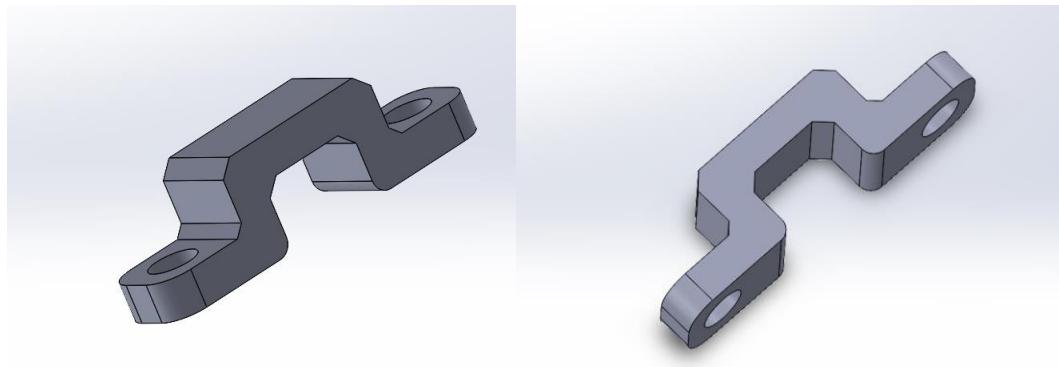
Vì robot để ngang chứ không phải để dọc nên cần phần cố định cho robot và ta dựa vào những cái thanh đồng ta dùng để cố định cho hệ thống. Ta sử dụng những phần nối lên có chấn 2 bên để tránh thanh đồng của robot có khả năng bị lệch (vì khối lượng của robot rất lớn mà không có chấn 2 bên rất dễ bị lệch)

CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.36. Nền và cố định cho robot nằm ngang

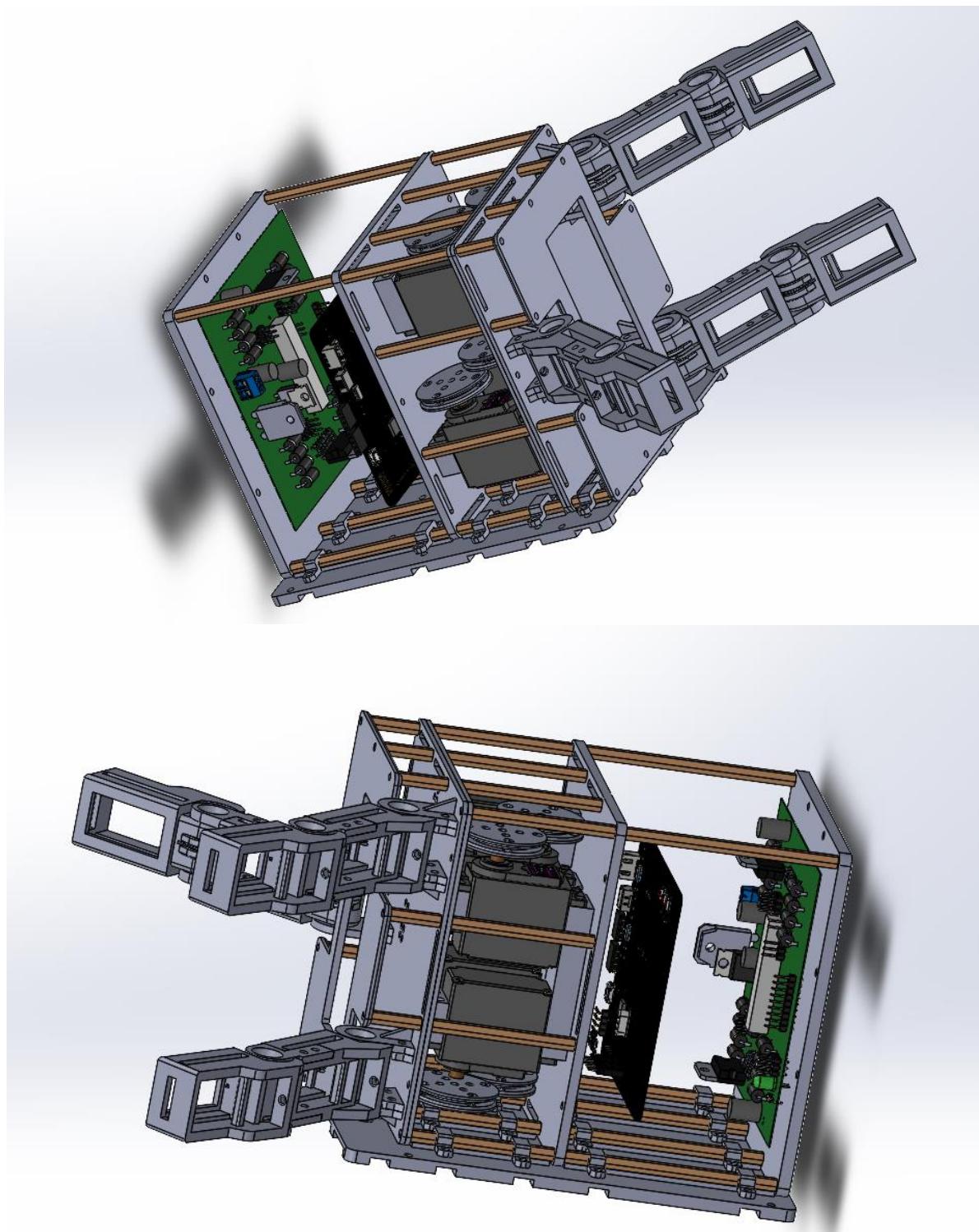
Khi đặt robot vào vị trí tương ứng, ta dùng thêm cái gá để áp đặt từ 2 mặt tác dụng lên thanh đồng, nhằm chắc chắn hơn



Hình 4.37. Gá cố định với phần nền để đặt robot nằm ngang

Khi lắp ghép ta được như sau:

CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.38. Robot hoàn thiện sau khi lắp nền

CHƯƠNG 4. THI CÔNG HỆ THỐNG

4.2.3. Chuẩn bị dữ liệu huấn luyện mô hình học sâu

Chuẩn bị dữ liệu bằng cách chụp ảnh từ camera và tiến hành đánh nhãn dữ liệu, truy cập <https://app.roboflow.com/> tiến hành upload dữ liệu và tiến hành đánh nhãn.



Hình 4.39. Đánh nhãn dữ liệu

Sau khi đánh nhãn dữ liệu ta tiến hành tăng cường dữ liệu bằng cách sử dụng các công cụ hỗ trợ trên roboflow. Tăng cường dữ liệu ảnh bằng cách:

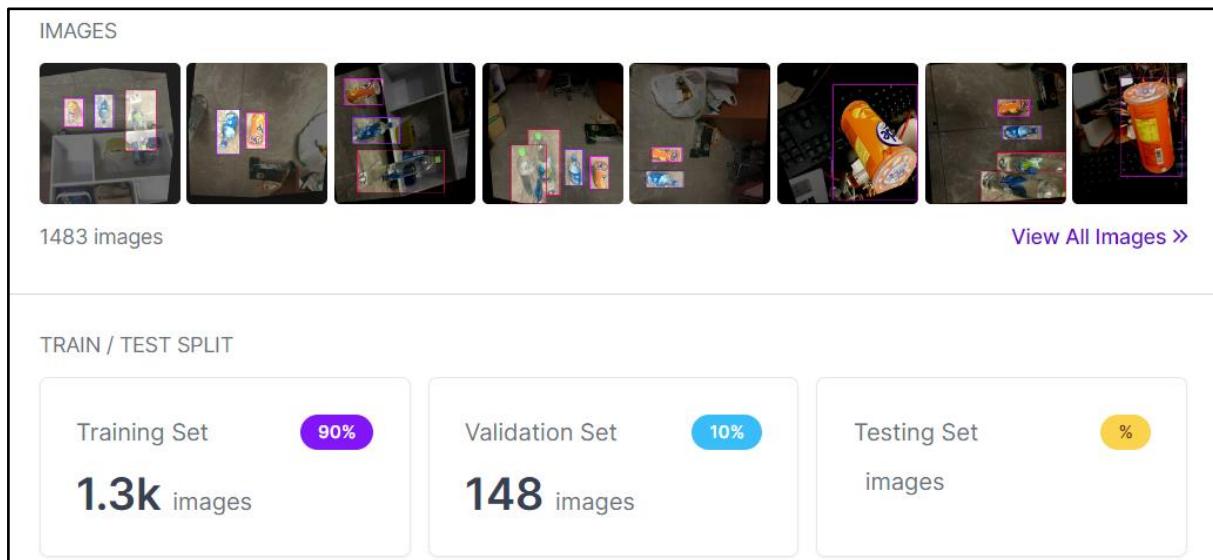
- Lật ảnh.
- Xoay ảnh gốc trong khoảng -20° và 20° .
- Phóng to, thu nhỏ ảnh.
- Tăng giảm độ sáng ảnh trong mức -40% đến 40% .

CHƯƠNG 4. THI CÔNG HỆ THỐNG

AUGMENTATIONS	Outputs per training example: 3
	Flip: Horizontal, Vertical
	90° Rotate: Clockwise, Counter-Clockwise, Upside Down
	Rotation: Between -20° and +20°
	Shear: ±15° Horizontal, ±15° Vertical
	Saturation: Between -40% and +40%
	Brightness: Between -25% and +25%
	Exposure: Between -16% and +16%

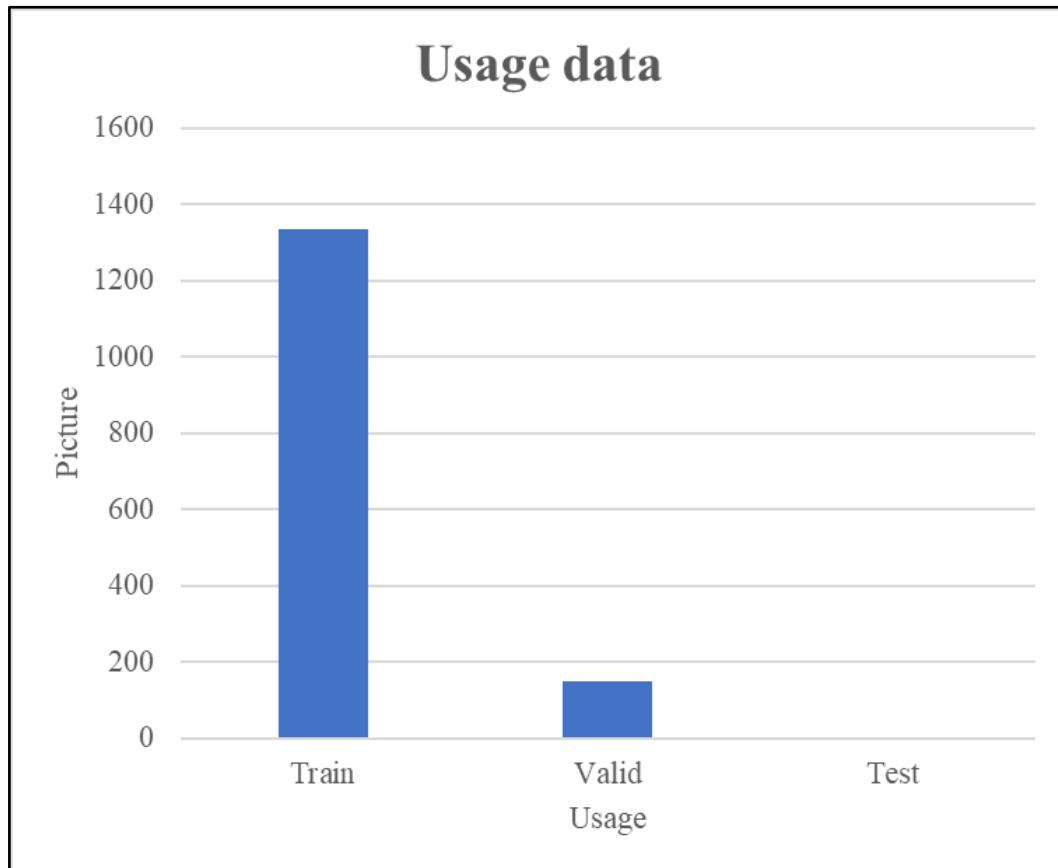
Hình 4.40. Tăng cường gói dữ liệu ảnh bằng roboflow

Từ 593 ảnh đầu vào đã chụp sau khi tăng cường ảnh thu được 1483 ảnh



Hình 4.41. Dữ liệu ảnh sau khi tăng cường

CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.42. Biểu đồ phân bố dữ liệu sử dụng cho quá trình huấn luyện

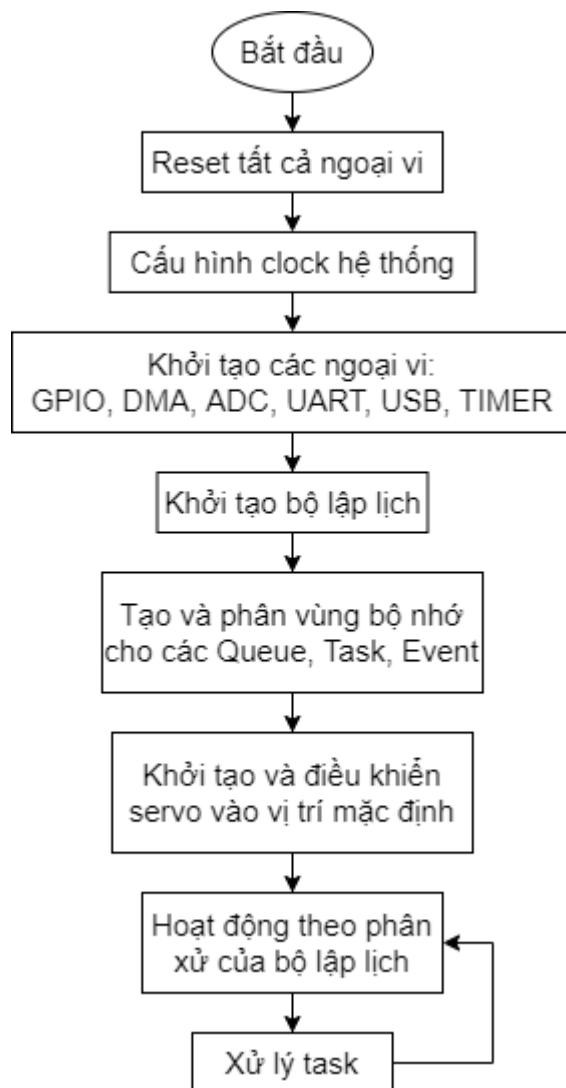
Ta chia tập dữ liệu ra thành 1335 ảnh cho quá trình huấn luyện, 148 ảnh cho quá trình quan sát, 0 ảnh cho quá trình kiểm thử vì chúng tôi cho rằng dữ liệu này không quá phức tạp nên không cần kiểm thử.

4.3. LẬP TRÌNH HỆ THỐNG

4.3.1. Lưu đồ giải thuật

a. Lưu đồ chương trình vi điều khiển STM32F103C8T6

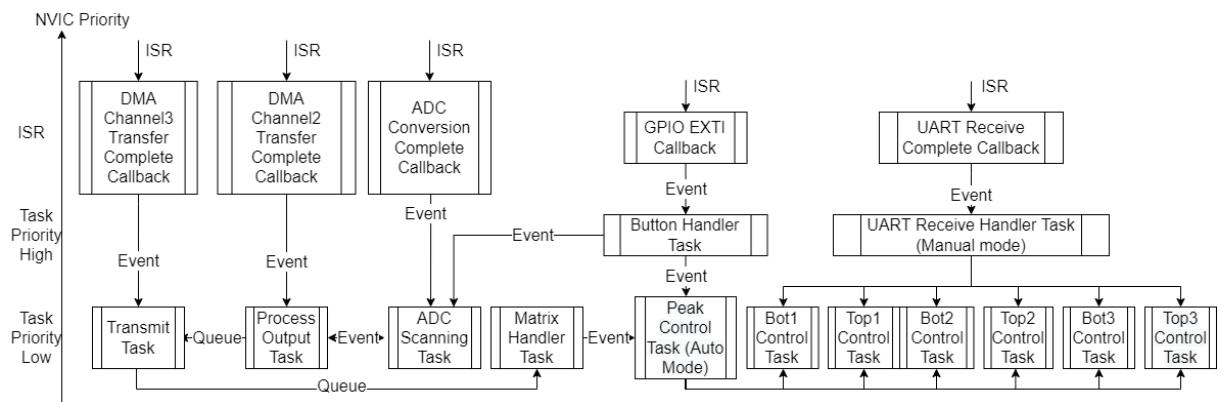
CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.43. Tổng quan lưu đồ giải thuật của chương trình vi điều khiển

Trong chương trình của vi điều khiển, sau khi cấu hình các clock và ngoại vi, các phần cần thiết cho 1 chương trình sử dụng FreeRTOS sẽ được thiết lập như cơ sở lý thuyết đã đề cập như queue, task, event,... Khi mọi thứ được thiết lập xong, động cơ sẽ được thiết lập và sau đó là công việc của bộ lập lịch. Nó sẽ phân xử để các tác được hoạt động theo lưu đồ tổng quan của hình trên.

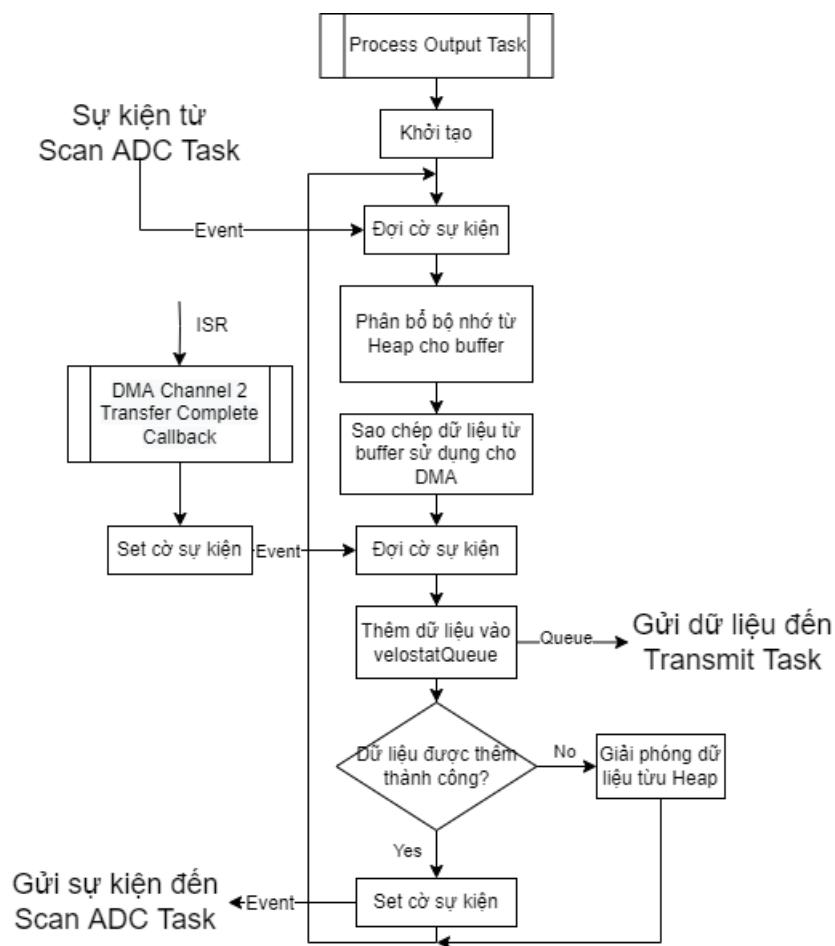
CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.44. Lưu đồ tổng quát hoạt động và ưu tiên của các ngắt và các tác vụ

Hình trên thể hiện các ưu tiên của các tác vụ và các ngắt trong chương trình vi điều khiển thông qua ưu tiên của NVIC. Trong hệ thống, các ngắt có mức ưu tiên là cao nhất, sau đó là những task có độ ưu tiên cao rồi đến những task có độ ưu tiên thấp. Dựa vào sơ đồ trên, bộ lập lịch sẽ phân xử để chương trình hoạt động. Các task được đồng bộ với nhau bằng Event group hoặc Queue.

CHƯƠNG 4. THI CÔNG HỆ THỐNG



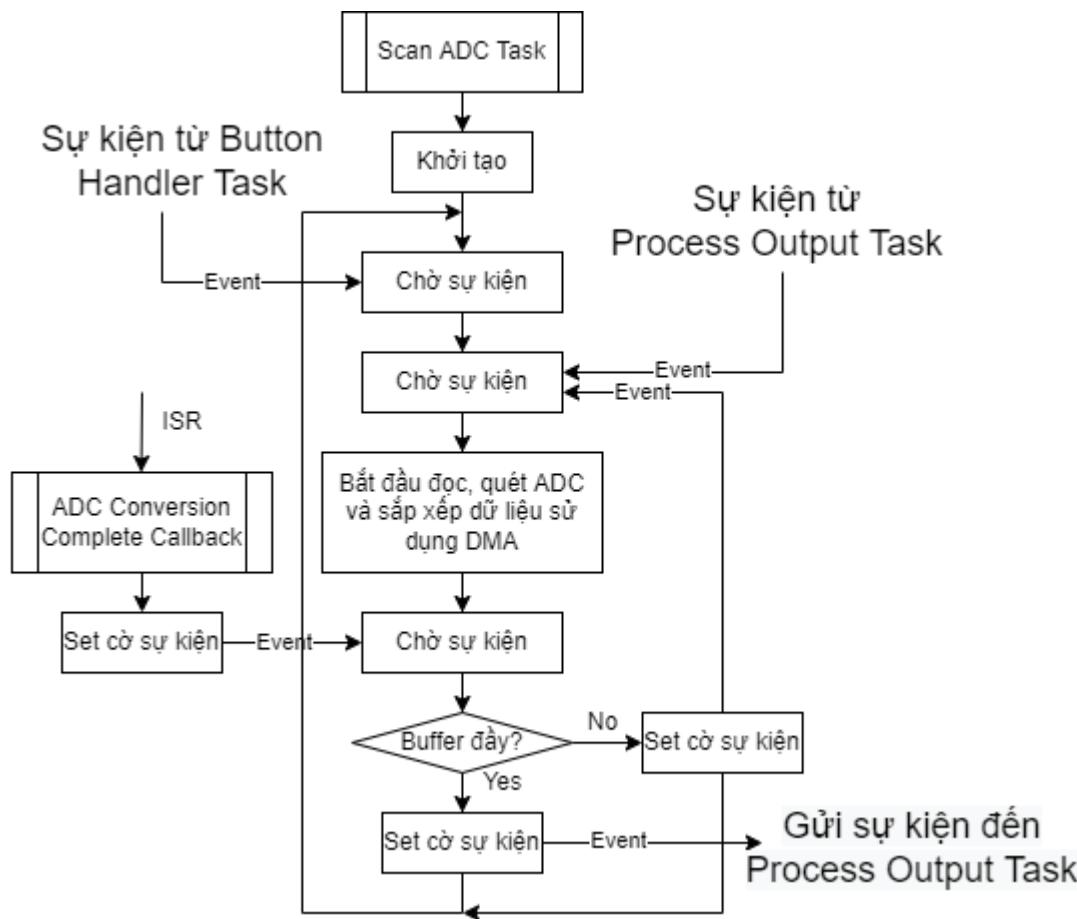
Hình 4.45. Lưu đồ giải thuật Process Output Task

Quy trình làm việc của Process Output Task như sau:

- Process Output Task được đồng bộ với Scan ADC Task bằng Event group, task sẽ ở trạng thái ngủ cho tới khi nhận cờ sự kiện từ Scan ADC Task để tiết kiệm tài nguyên CPU.
- Khi nhận được cờ sự kiện từ Scan ADC Task, task này sẽ cấp phát một vùng nhớ trong heap cho buffer.

CHƯƠNG 4. THI CÔNG HỆ THỐNG

- Sau đó dùng DMA để sao chép dữ liệu vào buffer để xử lý trên buffer, tránh đụng vào dữ liệu cũ để gây lỗi xung đột truy cập bộ nhớ. Task sẽ rời vào chế độ ngủ cho tới khi DMA vận chuyển xong dữ liệu đến buffer.
- Sau khi DMA vận chuyển xong, Task sẽ enqueue vào velostatQueue dữ liệu từ buffer. Sau đó task sẽ kiểm tra việc enqueue vào velostatQueue có thành công hay chưa. Nếu enqueue thành công thì task sẽ gửi cờ sự kiện tới Scan ADC Task để bắt đầu quét loạt dữ liệu mới. Nếu không thành công thì task sẽ giải phóng toàn bộ dữ liệu của buffer để tránh tràn bộ nhớ heap.



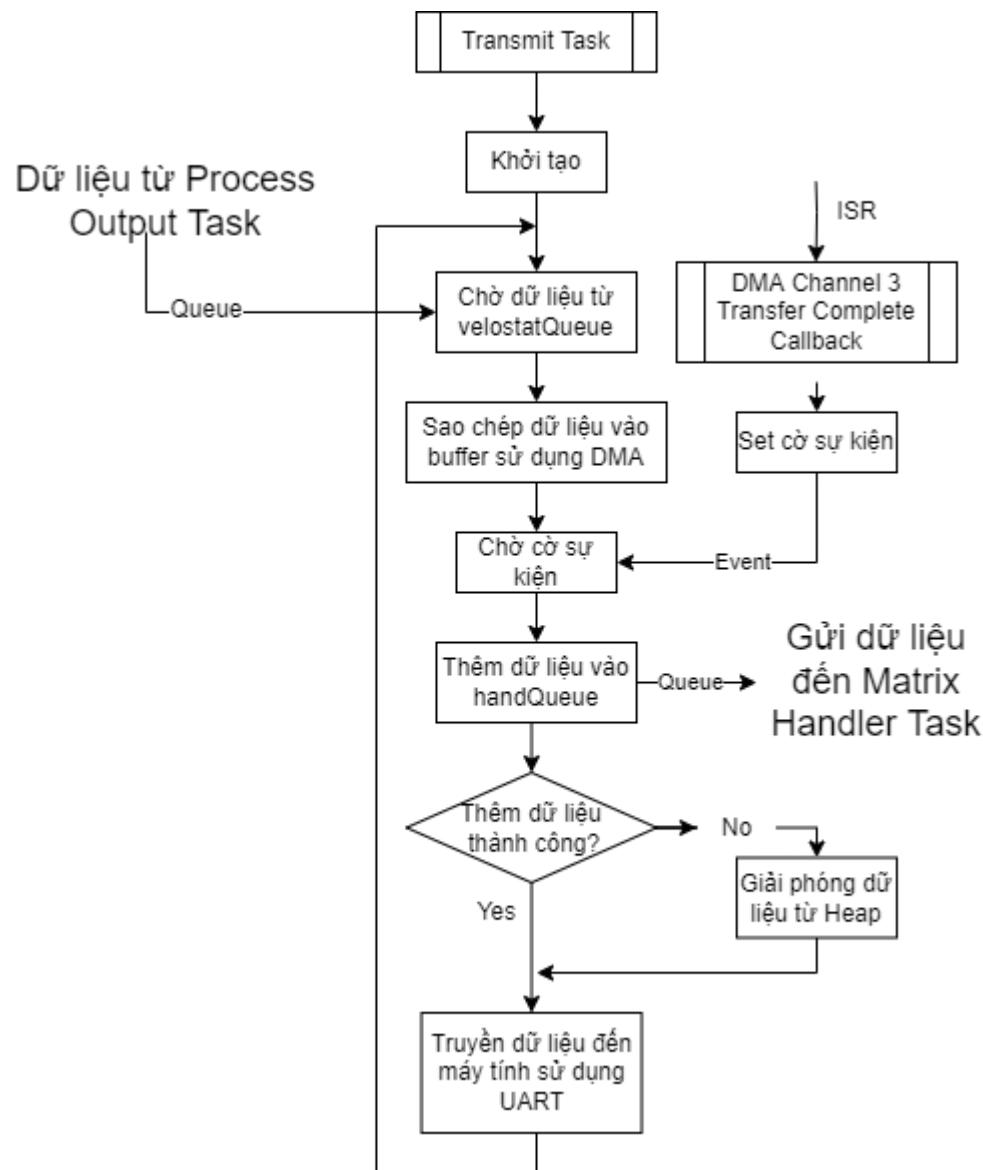
Hình 4.46. Lưu đồ giải thuật Scan ADC Task

Quy trình làm việc của Scan ADC Task như sau:

CHƯƠNG 4. THI CÔNG HỆ THỐNG

- Scan ADC Task được đồng bộ với Process Output Task và Button Handler Task, task sẽ rơi vào trạng thái ngủ cho tới khi được set tắt cả các cờ sự kiện. để tiết kiệm tài nguyên CPU.
- Sau khi nhận cờ sự kiện từ Button Handler Task thì task sẽ không tự động xóa đi cờ sự kiện này, task sẽ tiếp tục rơi vào trạng thái ngủ cho tới khi nhận cờ sự kiện tiếp theo. Có 2 cờ sự kiện có thể đánh thức task khỏi trạng thái ngủ là cờ sự kiện từ chính task khi báo hiệu buffer chưa đầy và cờ sự kiện từ Process Output Task báo hiệu đã enqueue thành công dữ liệu vào velostatQueue.
- Sau khi nhận cờ sự kiện thì task sẽ bắt đầu quá trình quét và đọc dữ liệu cảm biến, sau đó sắp xếp dữ liệu vào buffer dùng DMA.
- Task sẽ rơi vào trạng thái ngủ cho tới khi hoàn thành việc sắp xếp dữ liệu vào buffer.
- Sau khi nhận sự kiện từ ISR báo việc hoàn thành vận chuyển DMA xong thì task sẽ kiểm tra buffer đã đầy hay chưa. Nếu chưa thì task sẽ tự set cờ sự kiện cho mình để tiếp tục quá trình quét và đọc dữ liệu cảm biến mới. Nếu buffer đã đầy thì task sẽ set cờ dữ liệu cho Process Output Task để gửi dữ liệu vào velostatQueue.

CHƯƠNG 4. THI CÔNG HỆ THỐNG



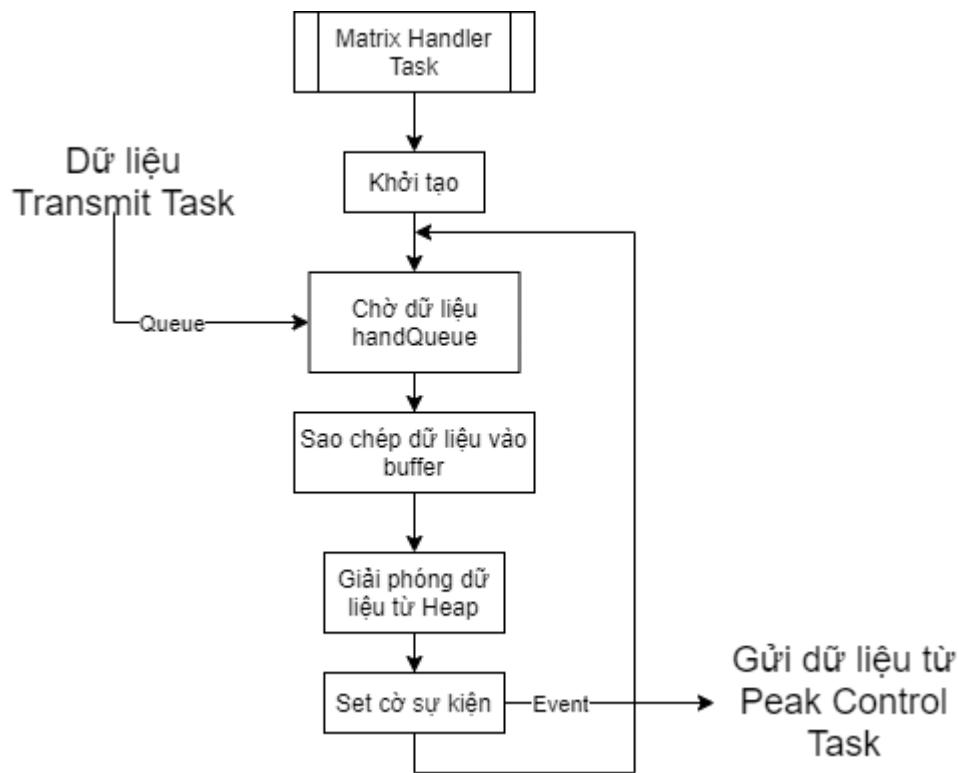
Hình 4.47. Lưu đồ giải thuật Transmit Task

Quy trình làm việc của Transmit Task như sau:

- Task sẽ rơi vào trạng thái ngủ cho tới khi có dữ liệu từ velostatQueue để tiết kiệm tài nguyên CPU.
- Khi có dữ liệu thì task sẽ dùng DMA để copy dữ liệu từ velostatQueue vào buffer để xử lý trên buffer, tránh lỗi truy cập vùng nhớ.

CHƯƠNG 4. THI CÔNG HỆ THỐNG

- Task sẽ rơi vào trạng thái khi cho tới khi DMA hoàn thành việc vận chuyển dữ liệu.
- Khi nhận cờ sự kiện, task bắt đầu enqueue vào handQueue (được đồng bộ với Matrix Handler Task). Sau đó kiểm tra việc enqueue đã thành công hay chưa. Nếu enqueue không thành công thì task sẽ giải phóng vùng nhớ của buffer để tránh tràn vùng nhớ heap.
- Sau đó task bắt đầu dùng UART để truyền dữ liệu sang máy tính.



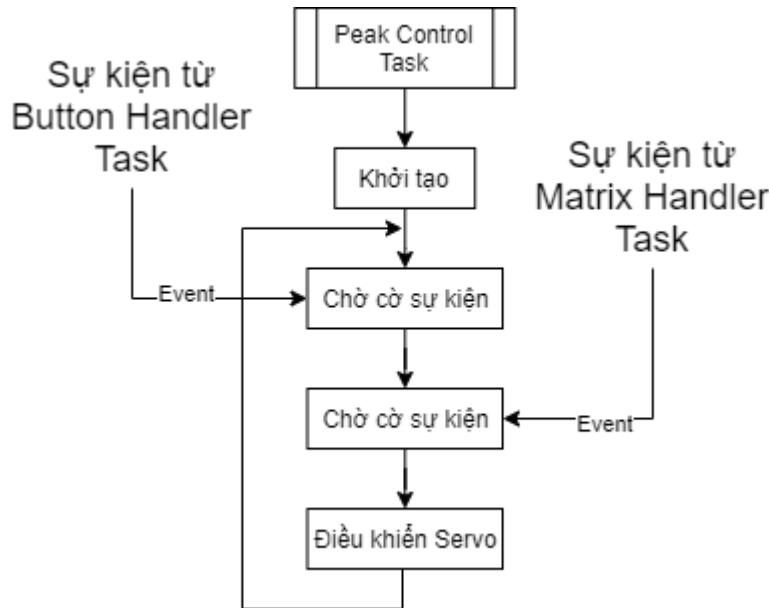
Hình 4.48. Lưu đồ giải thuật Matrix Handler Task

Quy trình làm việc của Matrix Handler Task như sau:

- Matrix Handler Task được đồng bộ với handQueue, task sẽ rơi vào trạng thái ngủ cho tới khi handQueue có dữ liệu.

CHƯƠNG 4. THI CÔNG HỆ THỐNG

- Khi có dữ liệu từ handQueue, task bắt đầu sao chép dữ liệu từ handQueue sang buffer để xử lý trên buffer, tránh lỗi truy cập vùng nhớ. Sau khi sử dụng dữ liệu xong thì task sẽ giải phóng vùng nhớ của buffer trong heap để có thể tái sử dụng lại vùng nhớ, tránh tràn vùng nhớ heap.
- Sau đó task sẽ set cờ sự kiện được đồng bộ với Peak Control Task.

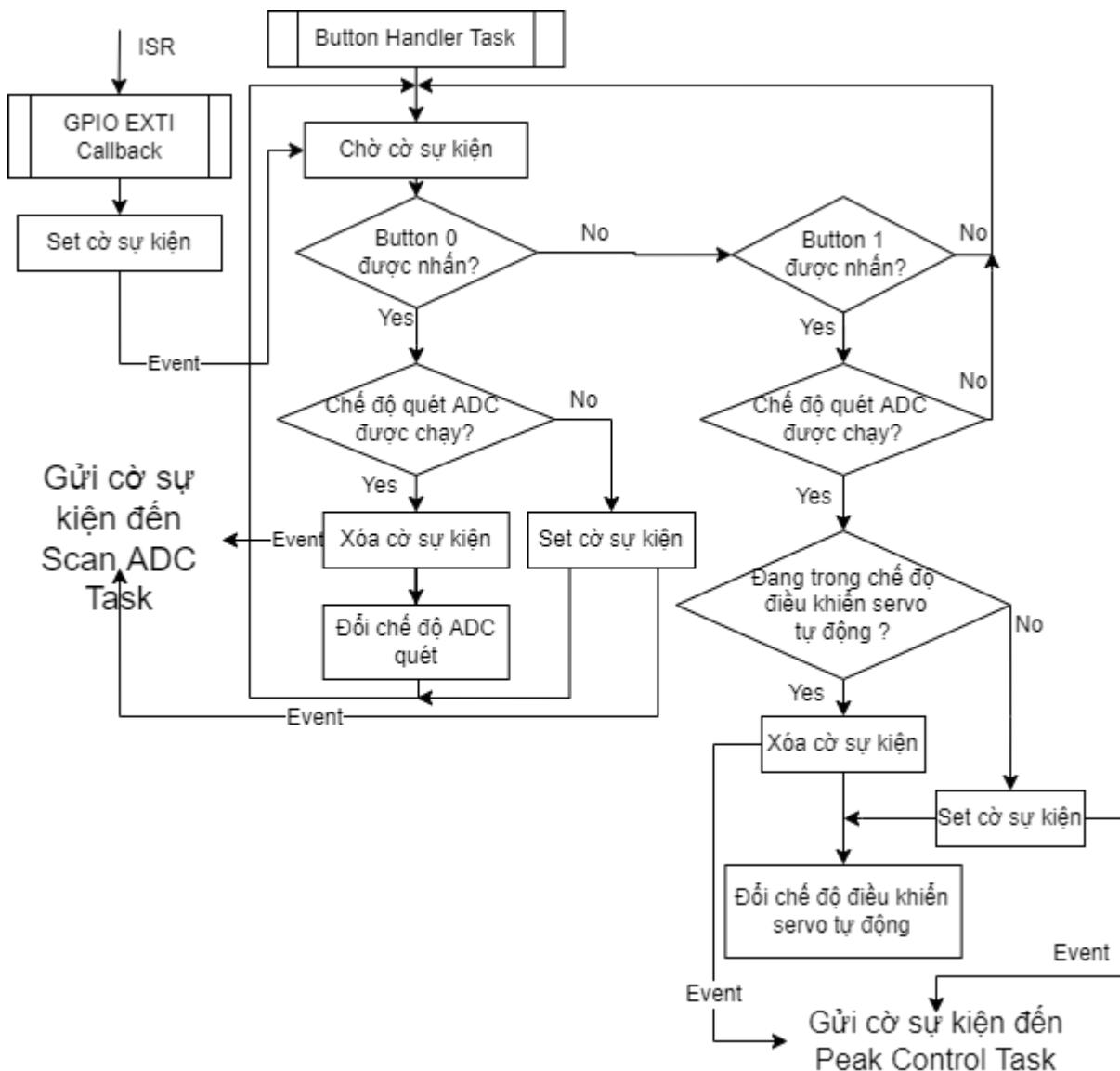


Hình 4.49. Lưu đồ giải thuật Peak Control Task

Quy trình làm việc của Peak Control Task như sau:

- Task được đồng bộ với Button Handler Task và Matrix Handler Task, task sẽ rơi vào trạng thái ngủ cho tới khi được set tắt cả cờ sự kiện.
- Khi được đánh thức thì task sẽ tiến hành dùng dữ liệu từ cảm biến để điều khiển servo.

CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.50. Lưu đồ giải thuật Button Handler Task

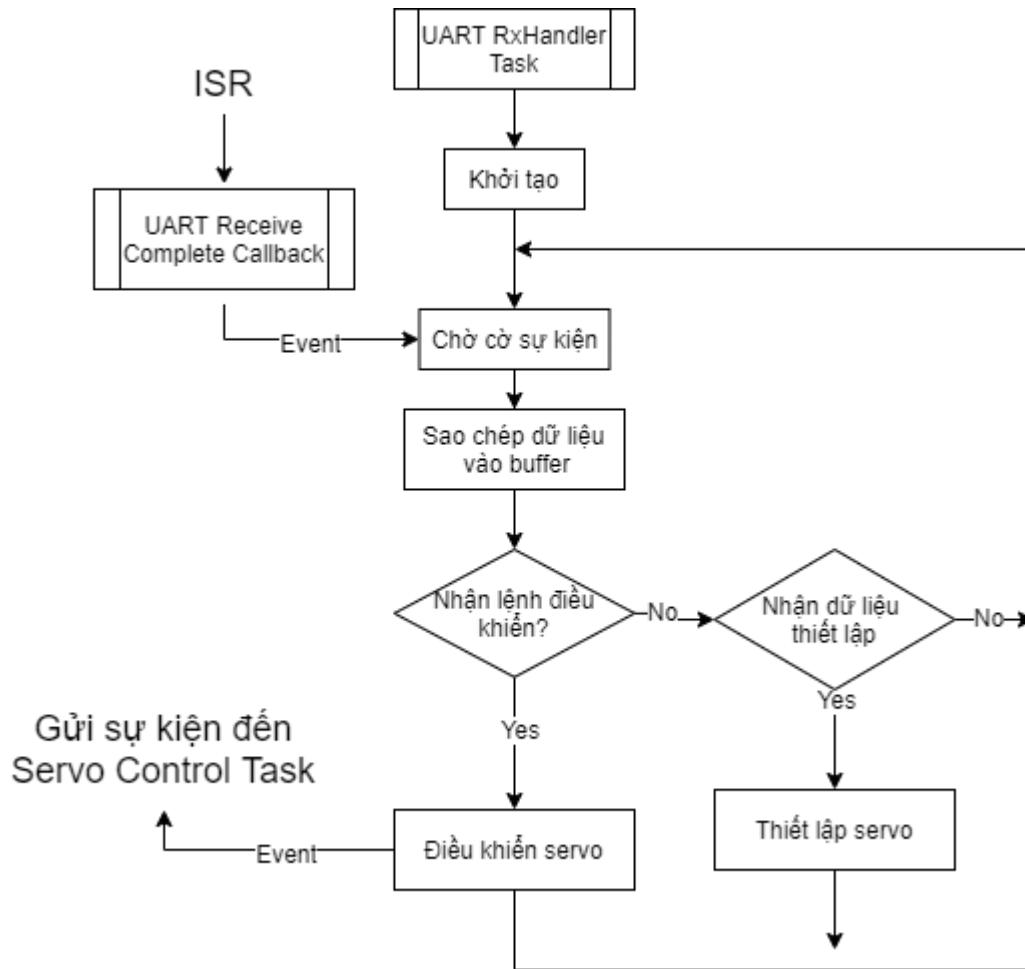
Quy trình làm việc của Button Handler Task như sau:

- Task được đồng bộ với ISR qua Event group, task sẽ rơi vào trạng thái ngủ cho tới khi cờ sự kiện được set.
- Nếu nút nhấn 0 được nhấn thì task sẽ kiểm tra chế độ quét ADC. Nếu chế độ quét ADC đang hoạt động thì task sẽ cho dừng hoạt động bằng cách xóa

CHƯƠNG 4. THI CÔNG HỆ THỐNG

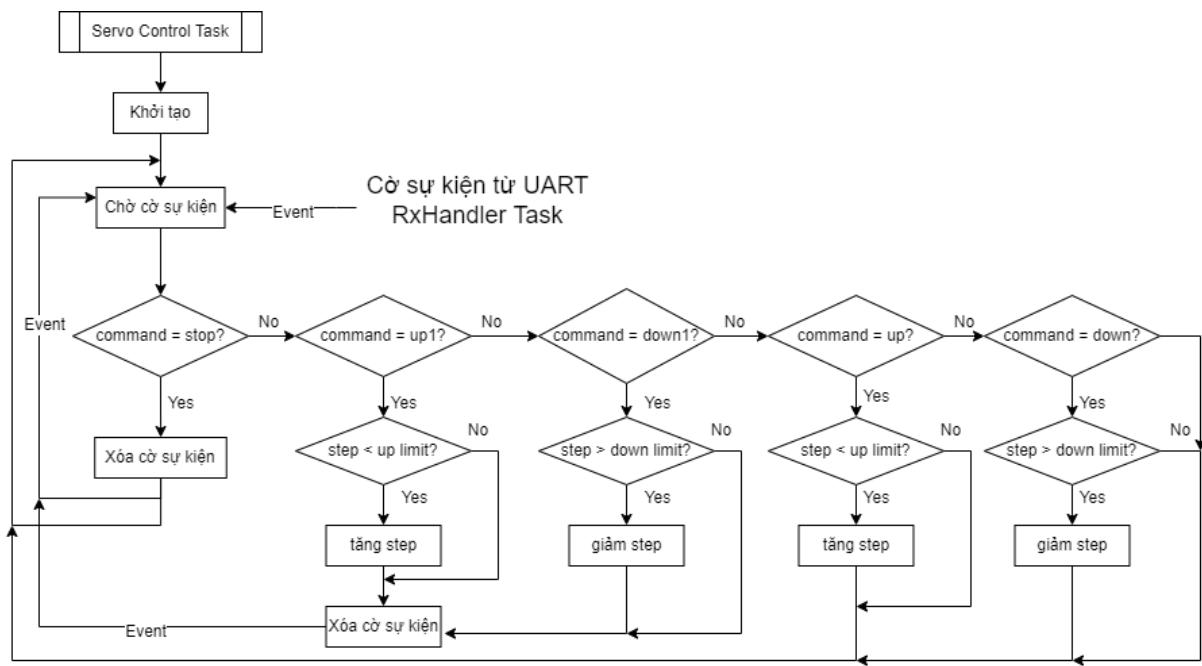
đi cờ sự kiện được đồng bộ với Scan ADC Task. Nếu chế độ quét ADC chưa hoạt động thì task sẽ cho hoạt động bằng cách set cờ sự kiện được đồng bộ với Scan ADC Task.

- Nếu nút nhấn 1 được nhấn thì Task sẽ kiểm tra chế độ quét ADC có đang hoạt động hay không. Sau đó task sẽ set hoặc clear cờ sự kiện được đồng bộ với Peak Control Task nếu chế độ quét ADC hoạt động hoặc không hoạt động (vì Peak Control Task sẽ điều khiển servo dựa trên giá trị của cảm biến).



Hình 4.51. Lưu đồ giải thuật UART RxHandler Task

CHƯƠNG 4. THI CÔNG HỆ THỐNG



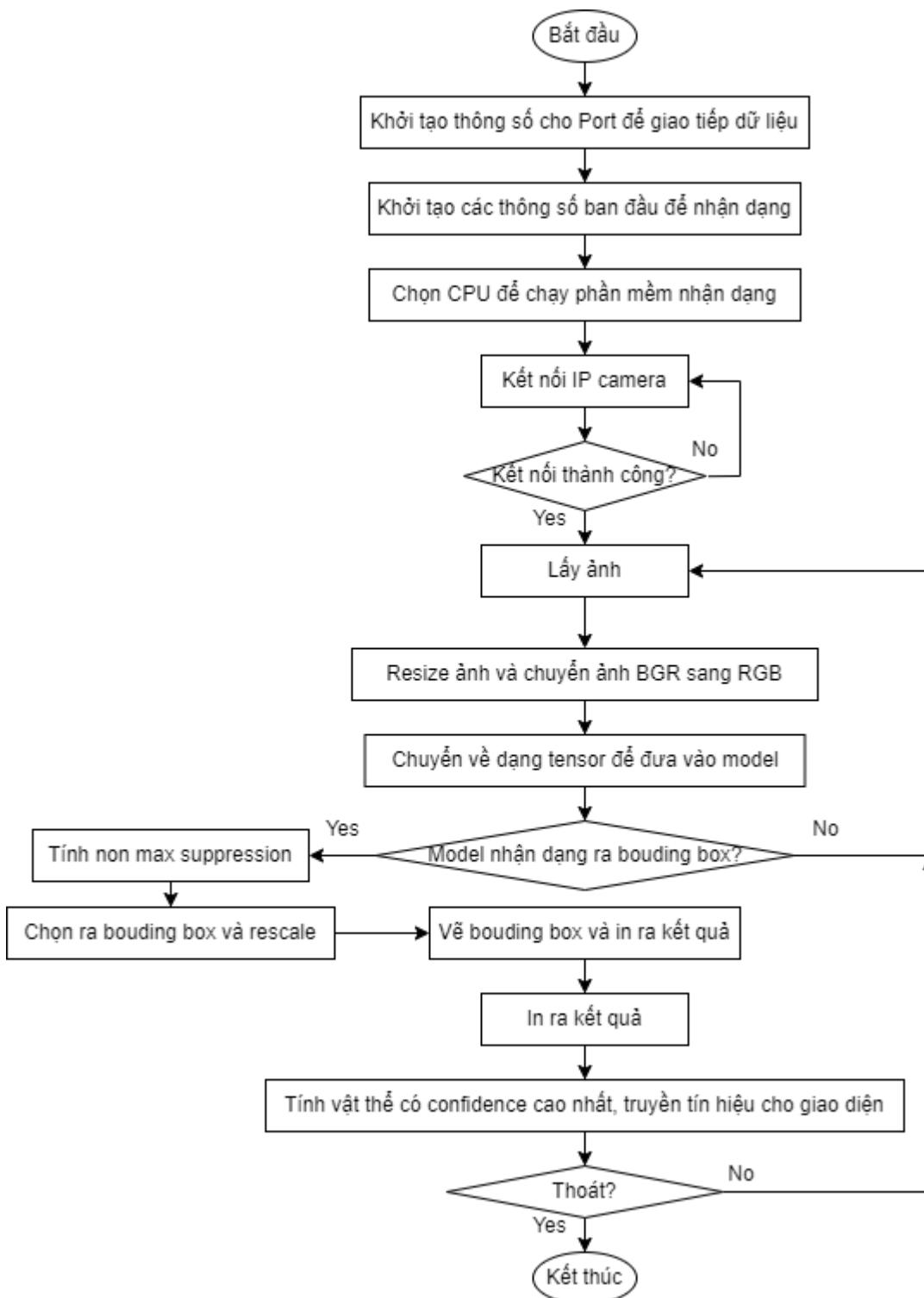
Hình 4.52. Lưu đồ giải thuật tác vụ điều khiển servo theo lệnh nhận được

b. Lưu đồ giải thuật phần mềm giao diện người dùng

Vì lý do phức tạp của phần mềm, rất khó để mô tả hết bằng lưu đồ nên chúng tôi xin phép mời bạn tham khảo code phần mềm trong phần phụ lục.

c. Lưu đồ giải thuật phần mềm nhận dạng

CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.53. Lưu đồ giải thuật chương trình nhận dạng vật thể

CHƯƠNG 4. THI CÔNG HỆ THỐNG

Phần mềm nhận dạng vật thể sẽ hoạt động khi chạy python script với file chương trình là detect_stream.py. Khi bắt đầu và kết nối thành công tất cả, phần mềm sẽ bắt đầu nhận diện tự động và truyền tín hiệu cho giao diện mỗi khi có cập nhật về vật thể xuất hiện.

4.3.2. Phần mềm lập trình cho vi điều khiển

a. Giới thiệu phần mềm lập trình STM32CubeIDE

❖ Giới thiệu

STM32CubeIDE là một công cụ phát triển đa hệ điều hành tất cả trong một, là một phần của hệ sinh thái phần mềm STM32Cube. STM32CubeIDE là một nền tảng phát triển C/C++ tiên tiến với các tính năng cấu hình ngoại vi, tạo mã, biên dịch mã và gỡ lỗi cho vi điều khiển và vi xử lý STM32. Nó dựa trên khung Eclipse / CDT™ và chuỗi công cụ GCC để phát triển và GDB để gỡ lỗi. Nó cho phép tích hợp hàng trăm plugin hiện có để hoàn thiện các tính năng của Eclipse IDE.

STM32CubeIDE tích hợp cấu hình STM32 và các chức năng tạo dự án từ STM32CubeMX để cung cấp trải nghiệm công cụ tất cả trong một và tiết kiệm thời gian cài đặt và phát triển. Sau khi lựa chọn MCU hoặc MPU STM32 trống, hoặc bộ vi điều khiển hoặc bộ vi xử lý được cấu hình sẵn từ việc chọn bảng hoặc chọn một ví dụ, dự án sẽ được tạo và mã khởi tạo được tạo. Bất kỳ lúc nào trong quá trình phát triển, người dùng có thể quay lại quá trình khởi tạo và cấu hình thiết bị ngoại vi hoặc phần mềm trung gian và tạo lại mã khởi tạo mà không ảnh hưởng đến mã người dùng.

STM32CubeIDE bao gồm các bộ phân tích xây dựng và ngăn xếp cung cấp cho người dùng thông tin hữu ích về tình trạng dự án và các yêu cầu về bộ nhớ.

CHƯƠNG 4. THI CÔNG HỆ THỐNG

STM32CubeIDE cũng bao gồm các tính năng gỡ lỗi tiêu chuẩn và nâng cao bao gồm chế độ xem thanh ghi lỗi CPU, bộ nhớ và thanh ghi ngoại vi, cũng như đồng hồ biến trực tiếp, giao diện Serial Wire Viewer hoặc bộ phân tích lỗi.

❖ Các tính năng

- Tích hợp các dịch vụ từ STM32CubeMX: vi điều khiển STM32, bộ vi xử lý, nền tảng phát triển và lựa chọn dự án mẫu
- Dựa trên Eclipse® / CDT™, với sự hỗ trợ cho các tiện ích bổ sung Eclipse®, GNU C/C++ cho chuỗi công cụ Arm® và trình gỡ lỗi GDB
- Dòng STM32MP1: Hỗ trợ cho các dự án OpenSTLinux: LinuxSupport cho Linux
- Các tính năng gỡ lỗi nâng cao bao gồm: lỗi CPU, thanh ghi ngoại vi và chế độ xem bộ nhớ
- Hỗ trợ cho các đầu dò gỡ lỗi ST-LINK (STMicroelectronics) và J-Link (SEGGER)
- Nhập dự án từ Atollic® TrueSTUDIO® và AC6 System Workbench cho STM32 (SW4STM32)
- Hỗ trợ đa hệ điều hành: Windows®, Linux® và macOS®, chỉ phiên bản 64 bit

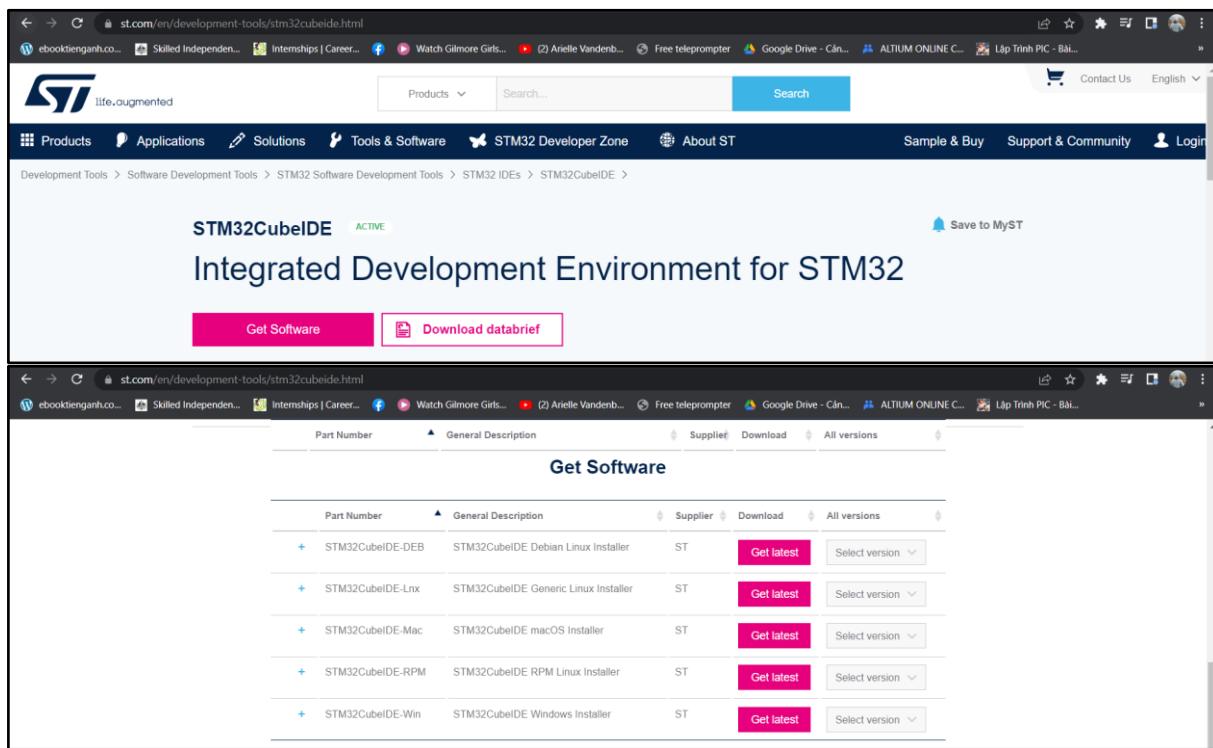
❖ Tải và cài đặt

- Vào link sau để tải:

<https://www.st.com/en/development-tools/stm32cubeide.html>

- Click chọn Get Software, sau đó chọn bản cài đặt trên hệ điều hành mà máy tính bạn hoạt động và phiên bản mà bạn mong muốn

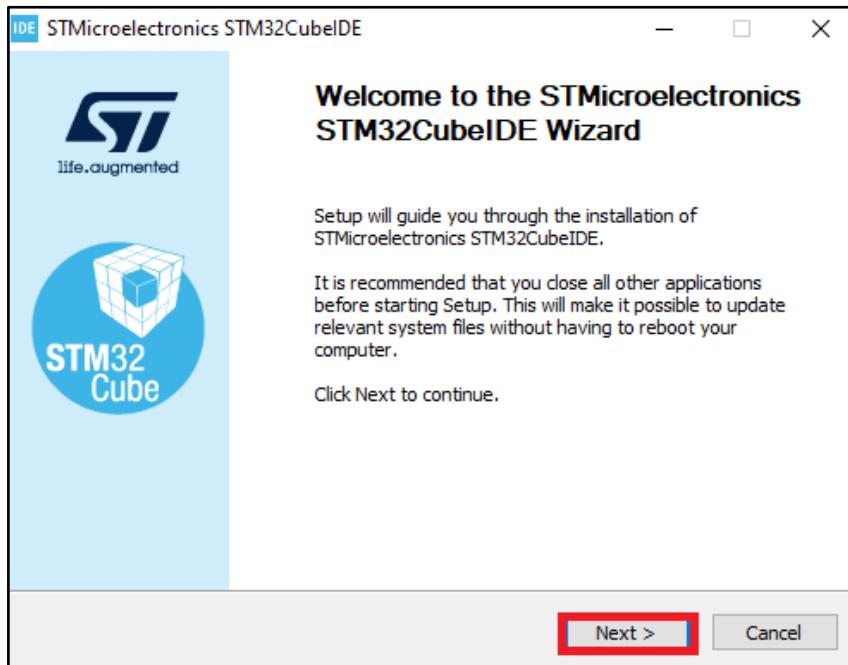
CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.54. Trang web tải phần mềm lập trình vi điều khiển

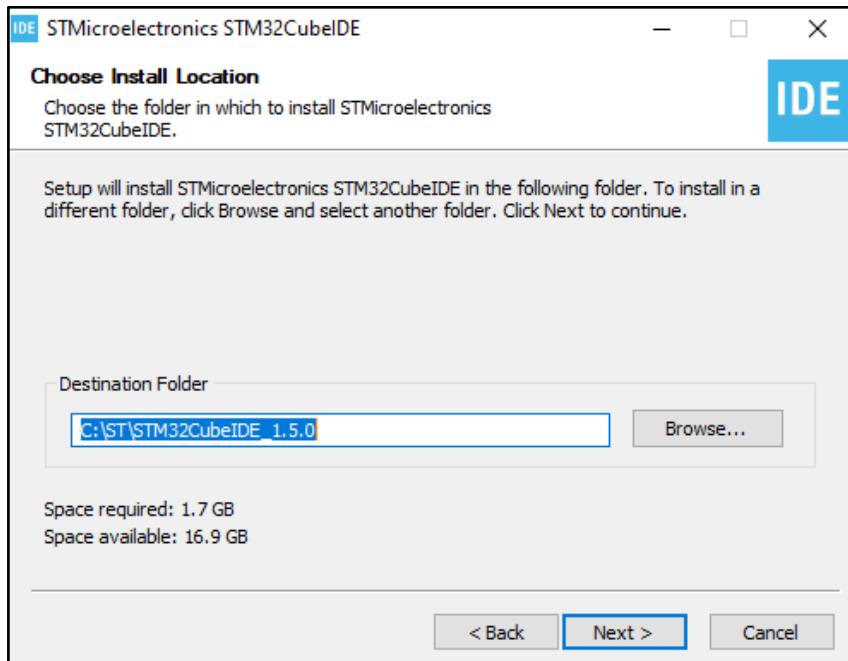
- Nếu như chưa tạo tài khoản trên trang chủ st.com, sau khi chọn Get Software, các bạn sẽ phải để lại e-mail để gửi link download. Sau khi xác nhận trong e-mail thì phần mềm có thể download.
- Vào thư mục vừa tải, giải nén và click vào app cài đặt phần mềm.

CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.55. Trang đầu tiên trong cửa sổ cài đặt STM32CubeIDE

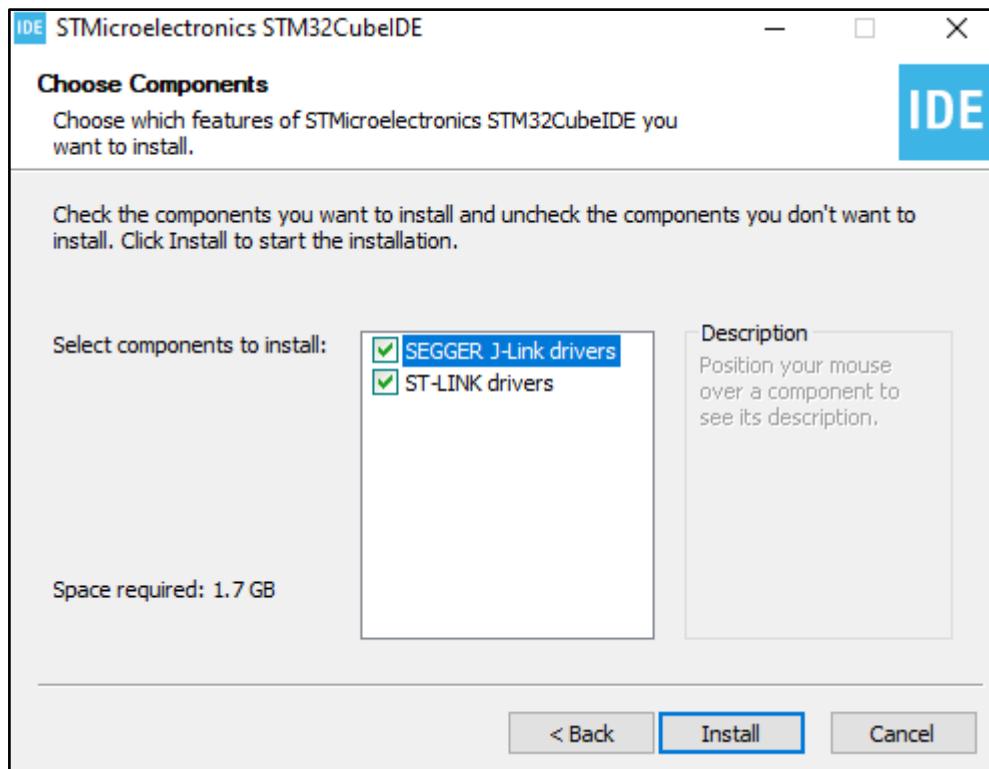
- Cứ I agree rồi next.



Hình 4.56. Trang thứ hai trong cửa sổ cài đặt STM32CubeIDE

CHƯƠNG 4. THI CÔNG HỆ THỐNG

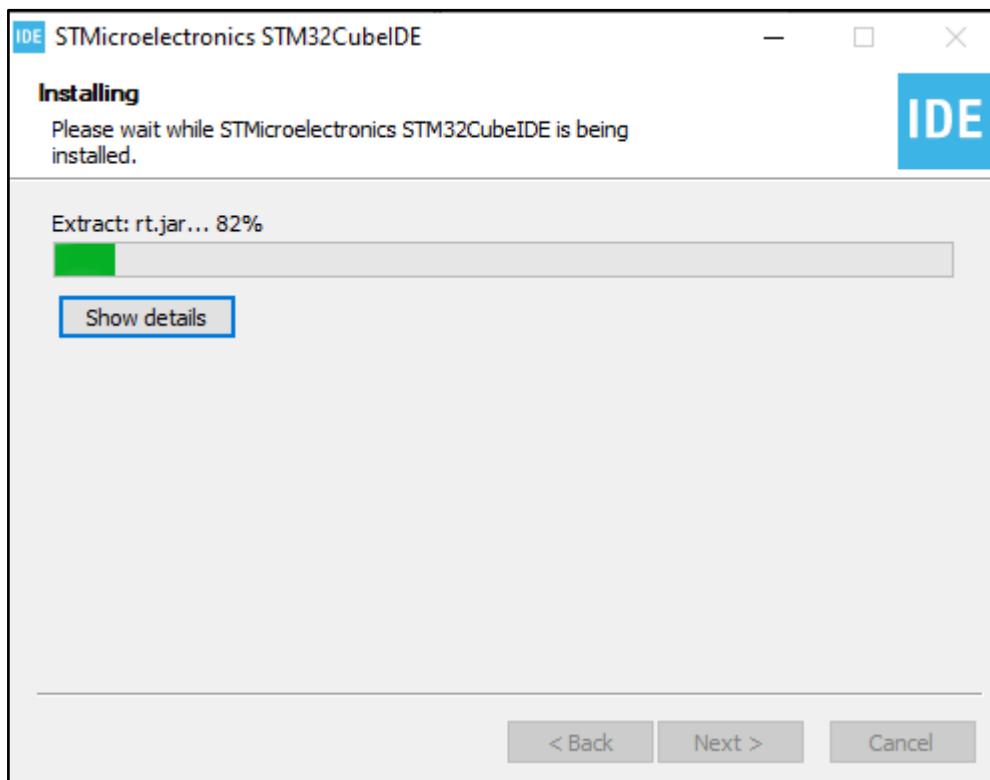
- Chọn đường dẫn lưu phần mềm. Bạn có thể thay đổi địa chỉ lưu hoặc để mặc định.
- Chọn install để tiến hành cài đặt



Hình 4.57. Trang thứ ba trong cửa sổ cài đặt STM32CubeIDE

- Chờ phần mềm chạy xong rồi bấm Finish

CHƯƠNG 4. THI CÔNG HỆ THỐNG



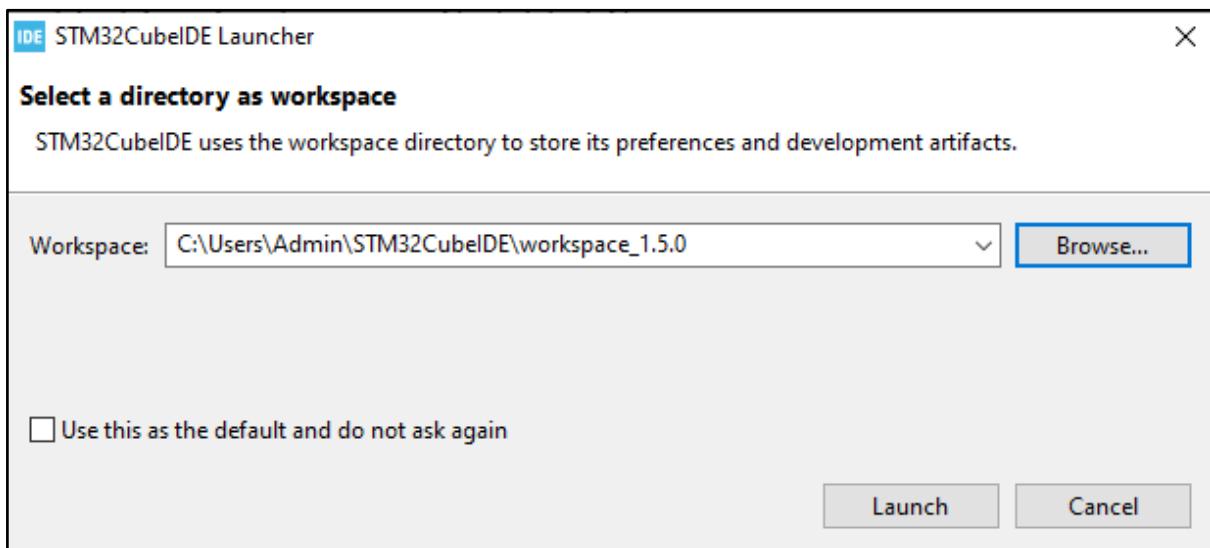
Hình 4.58. Cửa sổ cài đặt STM32CubeIDE bắt đầu quá trình cài đặt



Hình 4.59. Icon phần mềm STM32CubeIDE

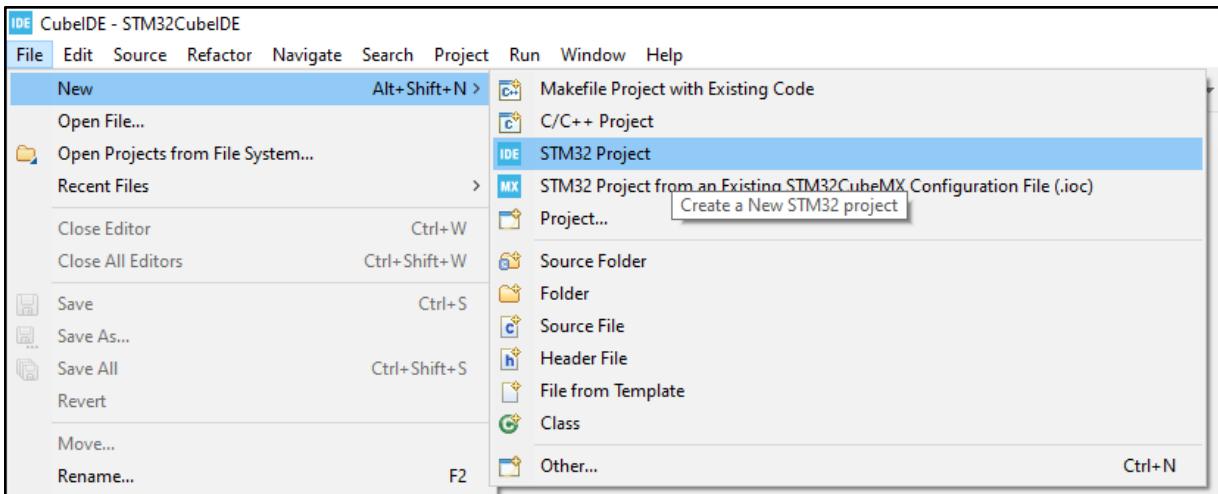
- Tại lần truy cập đầu tiên, sẽ hiện ra cửa sổ để cài đặt workspace cho các project.
- Hiểu tóm tắt là vị trí lưu các file làm việc của project các bạn tạo. Bạn có thể thay đổi hoặc để mặc định như dưới. Chúng tôi sẽ đổi đường dẫn để dễ kiểm soát.

CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.60. Cửa sổ lựa chọn workspace cho phần mềm STM32CubeIDE

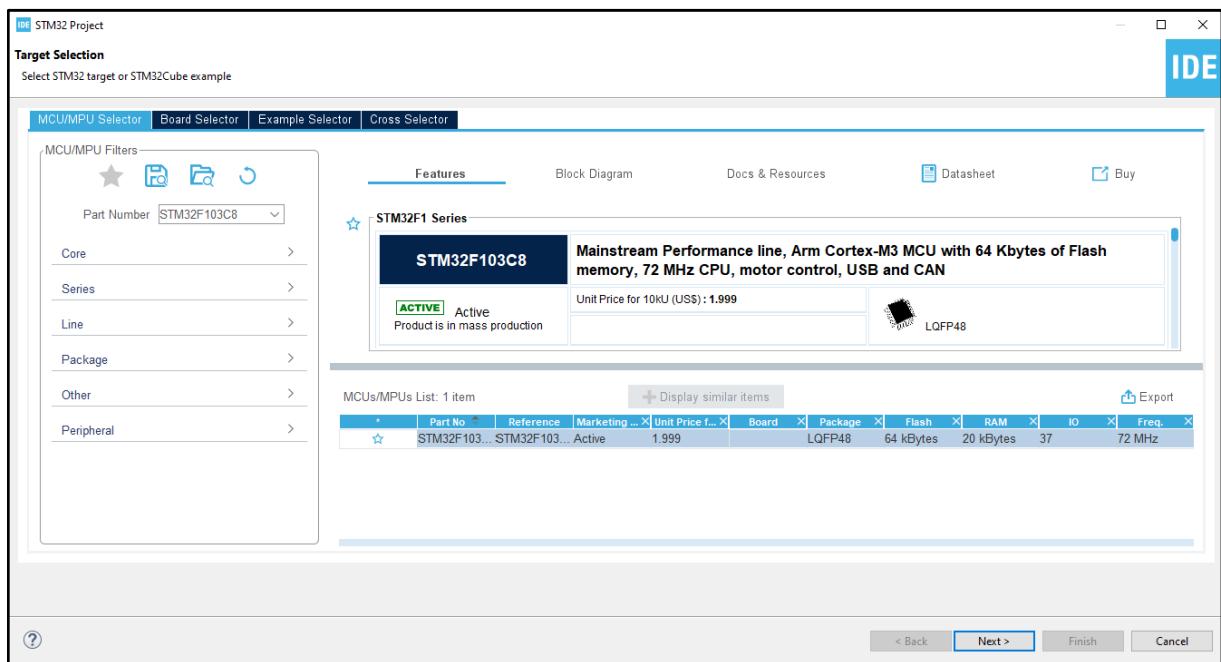
- Để tạo 1 project mới: File → New → STM32 Project



Hình 4.61. Tạo project trong STM32CubeIDE

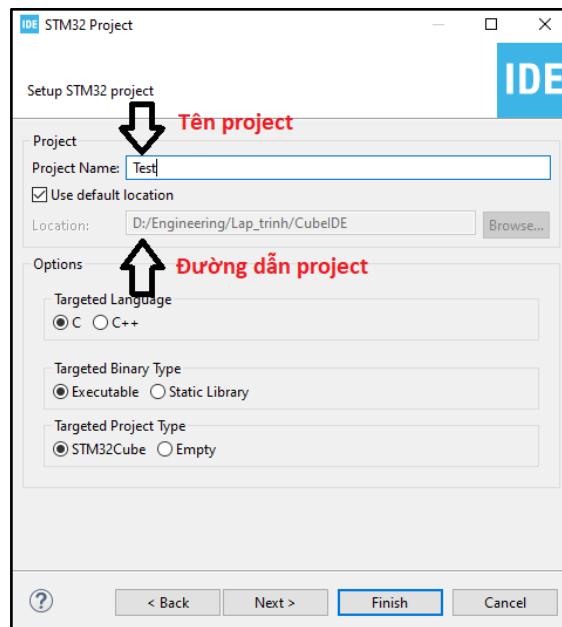
- Tiếp theo là cửa sổ chọn chip, ở đây chúng tôi chọn STM32F103C8

CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.62. Cửa sổ lựa chọn loại vi điều khiển cho dự án của STM32CubeIDE

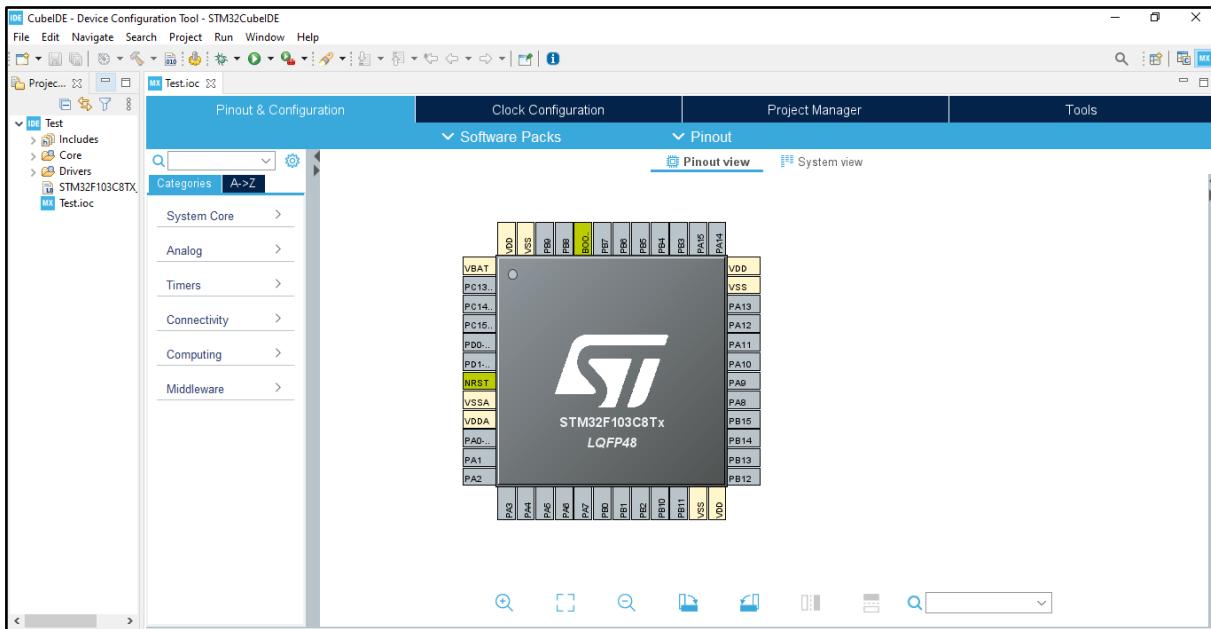
- Đặt tên và đường dẫn đến project



Hình 4.63. Đặt tên và đường dẫn cho dự án trong STM32CubeIDE

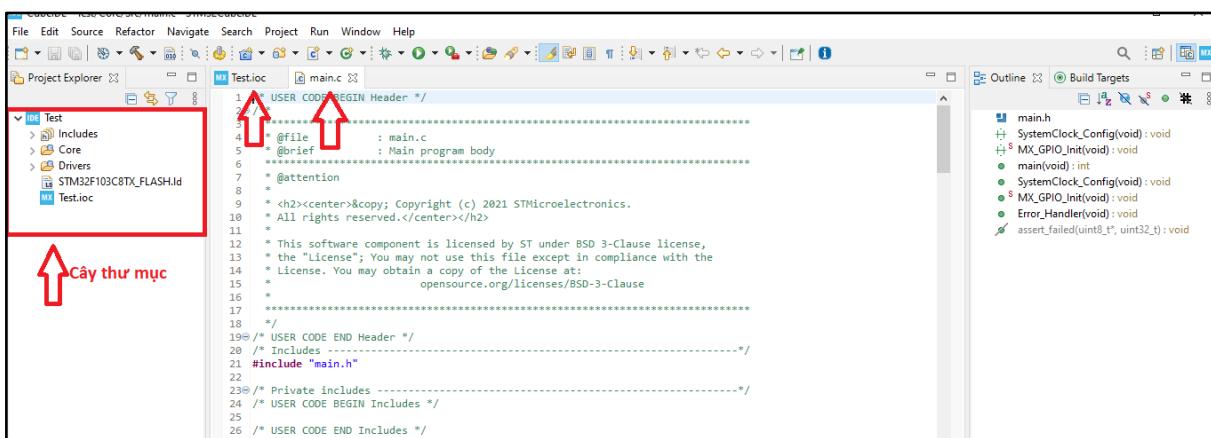
CHƯƠNG 4. THI CÔNG HỆ THỐNG

- Cửa sổ làm việc hiện ra



Hình 4.64. Cửa sổ làm việc sau khi tạo project của STM32CubeIDE

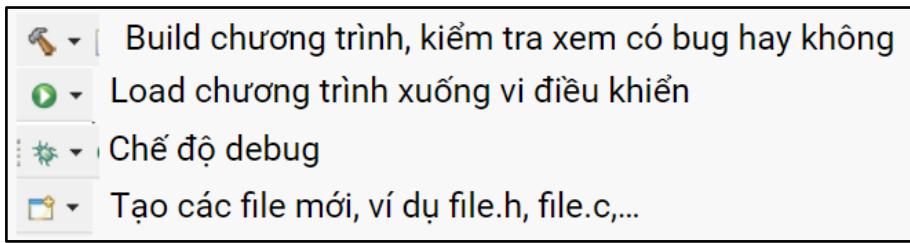
- Sau khi hoàn thành cấu hình cho chip, chọn Save hay Ctrl+S, để lưu đồng thời phần mềm sẽ tự động generate code sang file main.c
 - File .ioc là file để cấu hình chip
 - File main.c là chương trình code chính



Hình 4.65. Cửa sổ làm việc sau khi tạo code của STM32CubeIDE

CHƯƠNG 4. THI CÔNG HỆ THỐNG

- Các chức năng chính trong phần mềm



Hình 4.66. Một số chức năng chính trong STM32CubeIDE

4.3.3. Phần mềm lập trình cho giao diện người dùng

Trong phần giới thiệu ngắn này về môi trường phát triển tích hợp Visual Studio (IDE), bạn sẽ tạo một ứng dụng C# đơn giản có giao diện người dùng (UI) dựa trên Windows.

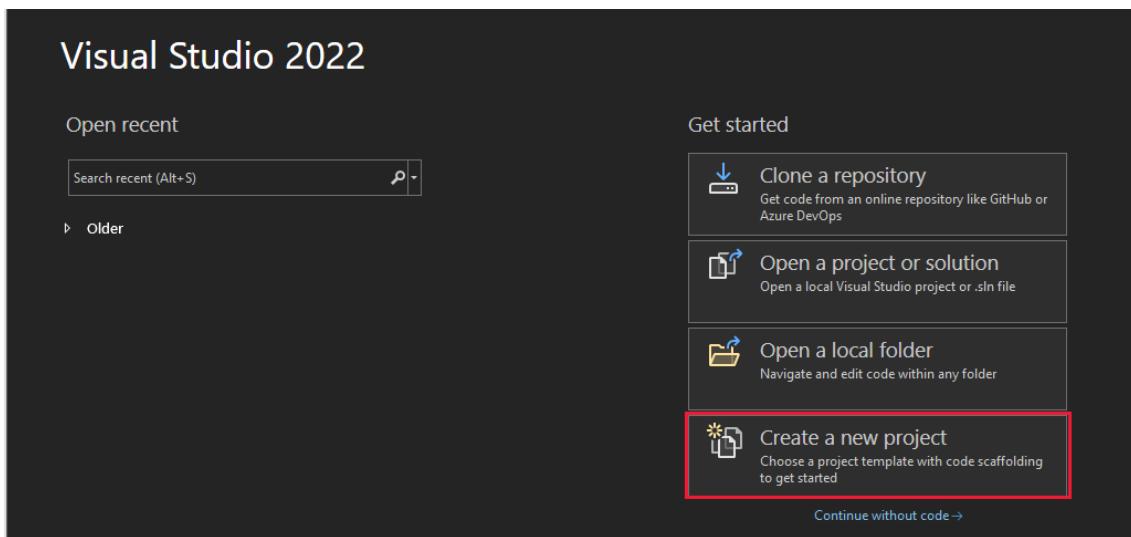
Truy cập trang <https://visualstudio.microsoft.com/fr/downloads/> tải xuống Visual Studio 2022 để cài đặt miễn phí.

❖ Tạo một dự án

Đầu tiên, bạn sẽ tạo một dự án ứng dụng C#. Loại dự án đi kèm với tất cả các tệp mẫu bạn cần, trước khi bạn thêm bất cứ thứ gì.

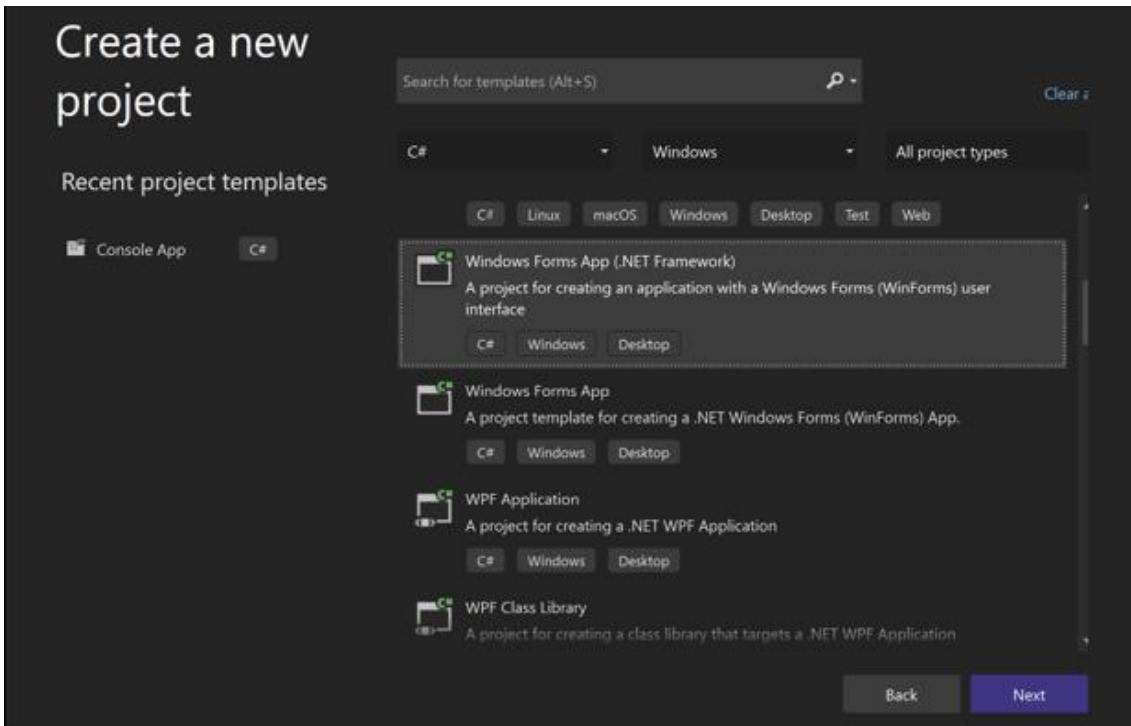
- Mở Visual Studio.
- Trên cửa sổ bắt đầu, chọn Tạo dự án mới.

CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.67. Cửa sổ tạo dự án mới của Visual Studio 2022

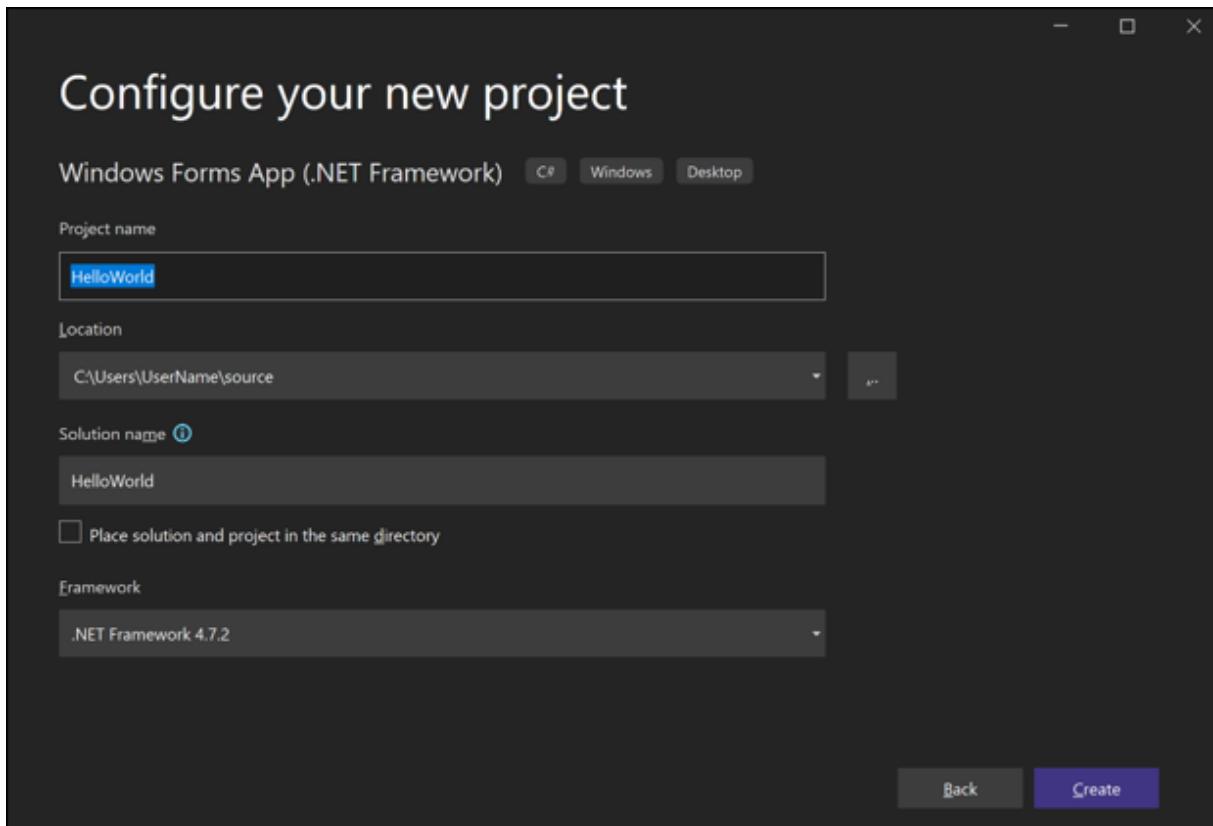
- Trên cửa sổ Tạo dự án mới, chọn mẫu Windows Forms App (.NET Framework) cho C #.



Hình 4.68. Cửa sổ chọn template trong Visual Studio 2022

CHƯƠNG 4. THI CÔNG HỆ THỐNG

- Trong cửa sổ cấu hình dự án mới của bạn, nhập hoặc nhập HelloWorld vào hộp Project Name. Sau đó, chọn Create.



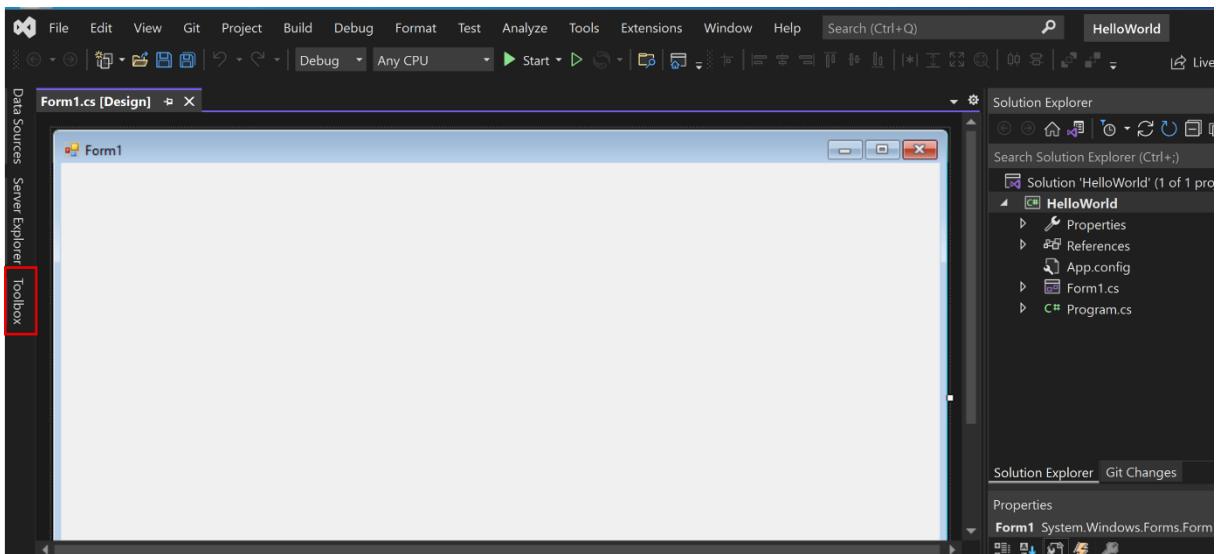
Hình 4.69. Cửa sổ cấu hình dự án trong Visual Studio 2022

❖ Tạo ứng dụng

Sau khi bạn chọn mẫu dự án C# và đặt tên cho tệp của mình, Visual Studio sẽ mở một biểu mẫu cho bạn. Biểu mẫu là một giao diện người dùng Windows. Chúng tôi sẽ tạo ứng dụng "Hello World" bằng cách thêm các điều khiển vào biểu mẫu và sau đó chúng tôi sẽ chạy ứng dụng.

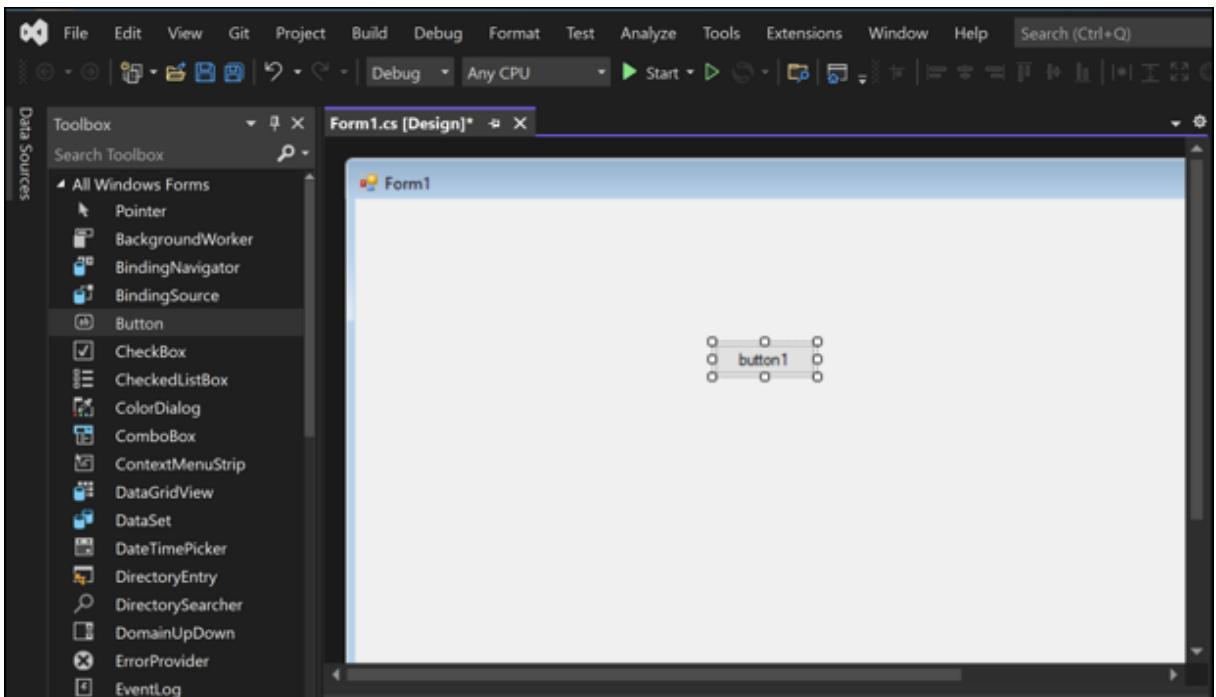
- Thêm một nút vào biểu mẫu
 - Chọn Toolbox để mở cửa sổ hiện ra.

CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.70. Giao diện với cửa sổ giao diện trong Visual Studio 2022

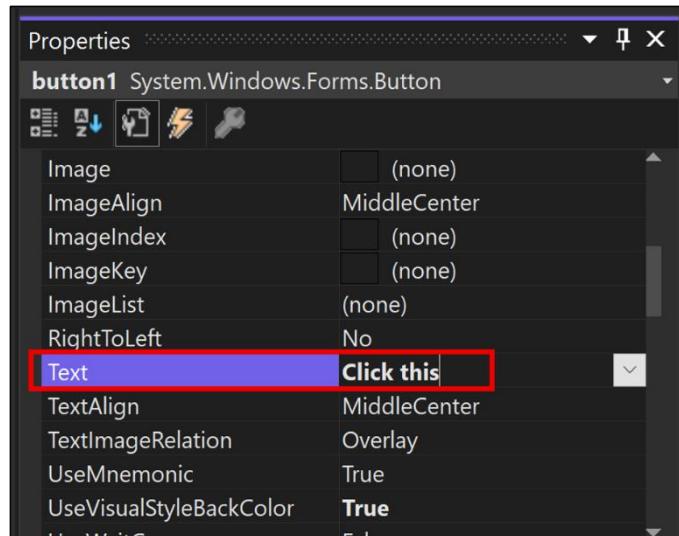
- Chọn điều khiển Button và sau đó kéo nó vào biểu mẫu.



Hình 4.71. Kéo thả nút nhấn trong Visual Studio 2022

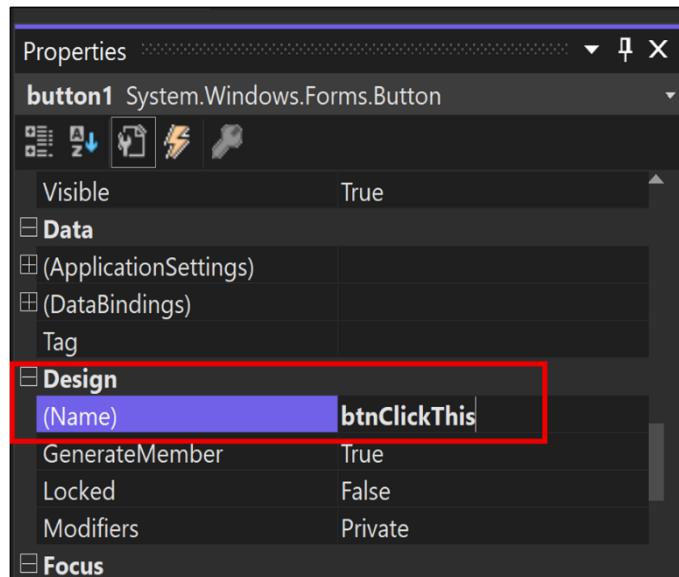
CHƯƠNG 4. THI CÔNG HỆ THỐNG

- Trong cửa sổ Properties, vào ô Text, đổi tên từ button1 thành Click this và nhấn Enter.



Hình 4.72. Đổi văn bản nút nhấn trong cửa sổ Properties Visual Studio 2022

- Trong phần Design của cửa sổ Properties, đổi tên từ button1 thành btnClickThis và nhấn Enter

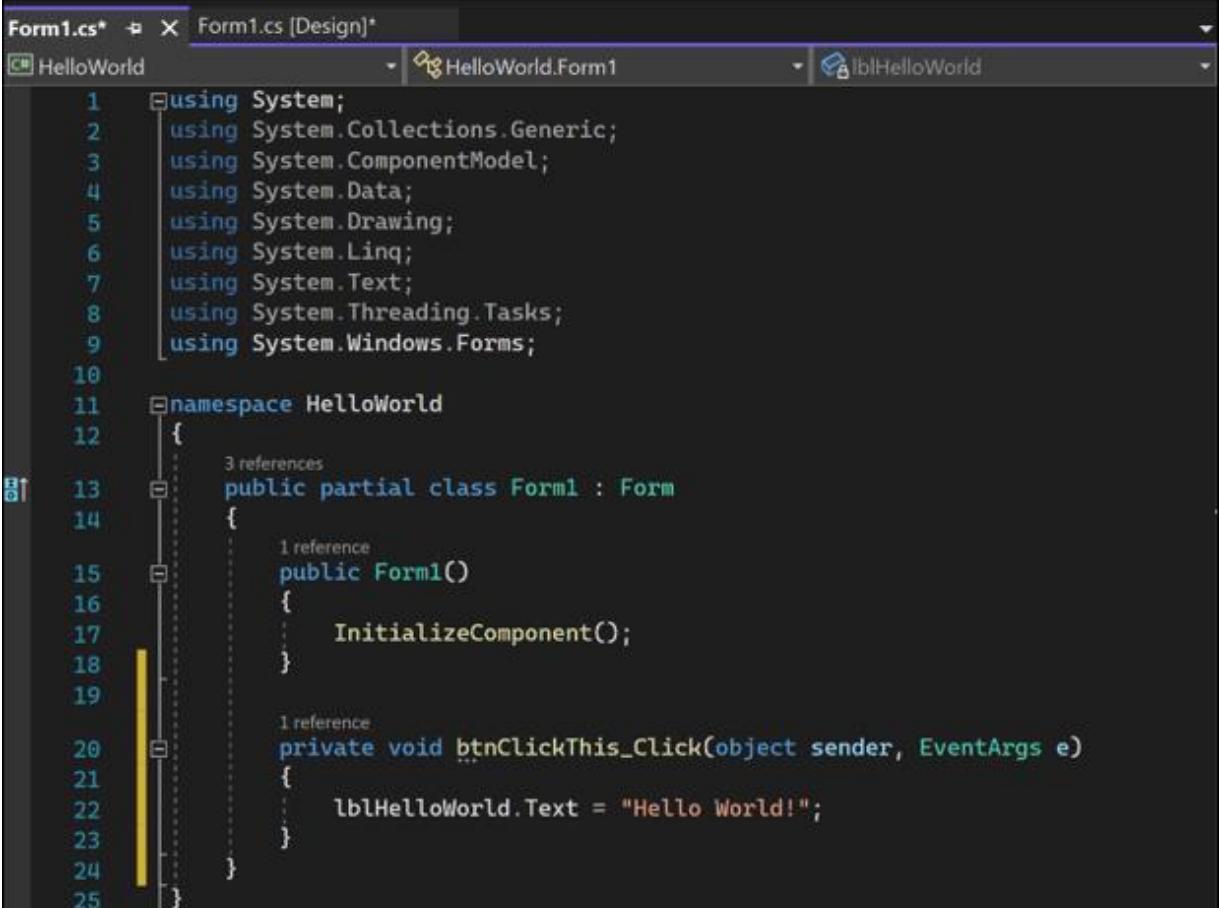


Hình 4.73. Đổi tên nút nhấn trong cửa sổ Properties Visual Studio 2022

CHƯƠNG 4. THI CÔNG HỆ THỐNG

- Thêm nhãn vào cửa sổ
 - Chọn điều khiển Label từ cửa sổ Toolbox, kéo thả biểu mẫu vào cửa sổ đặt dưới nút Click this
 - Trong phần Design trong cửa sổ Properties, đổi tên từ label1 thành lblHelloWorld và nhấn Enter
- Thêm code vào
 - Trong cửa sổ Form1.cs [Design], bấm đúp vào nút Click this để mở cửa sổ Form1.cs.
 - Trong cửa sổ Form1.cs, sau dòng void private, gõ hoặc nhập lblHelloWorld.Text = "Hello World!"; như thể hiện trong ảnh chụp màn hình sau

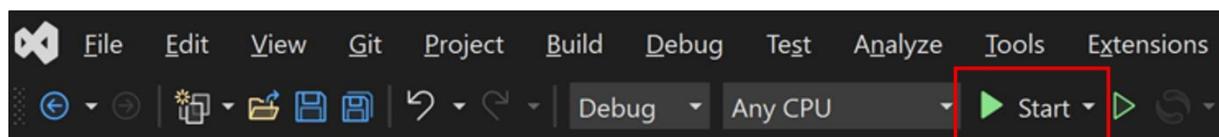
CHƯƠNG 4. THI CÔNG HỆ THỐNG



```
Form1.cs*  X  Form1.cs [Design]*  
C# HelloWorld  ↗ HelloWorld.Form1  ↗ lblHelloWorld  
1  using System;  
2  using System.Collections.Generic;  
3  using System.ComponentModel;  
4  using System.Data;  
5  using System.Drawing;  
6  using System.Linq;  
7  using System.Text;  
8  using System.Threading.Tasks;  
9  using System.Windows.Forms;  
10  
11  namespace HelloWorld  
12  {  
13      public partial class Form1 : Form  
14      {  
15          public Form1()  
16          {  
17              InitializeComponent();  
18          }  
19  
20          private void btnClickThis_Click(object sender, EventArgs e)  
21          {  
22              lblHelloWorld.Text = "Hello World!";  
23          }  
24      }  
25  }
```

Đoạn code ứng dụng nút nhấn đơn giản trong Visual Studio 2022

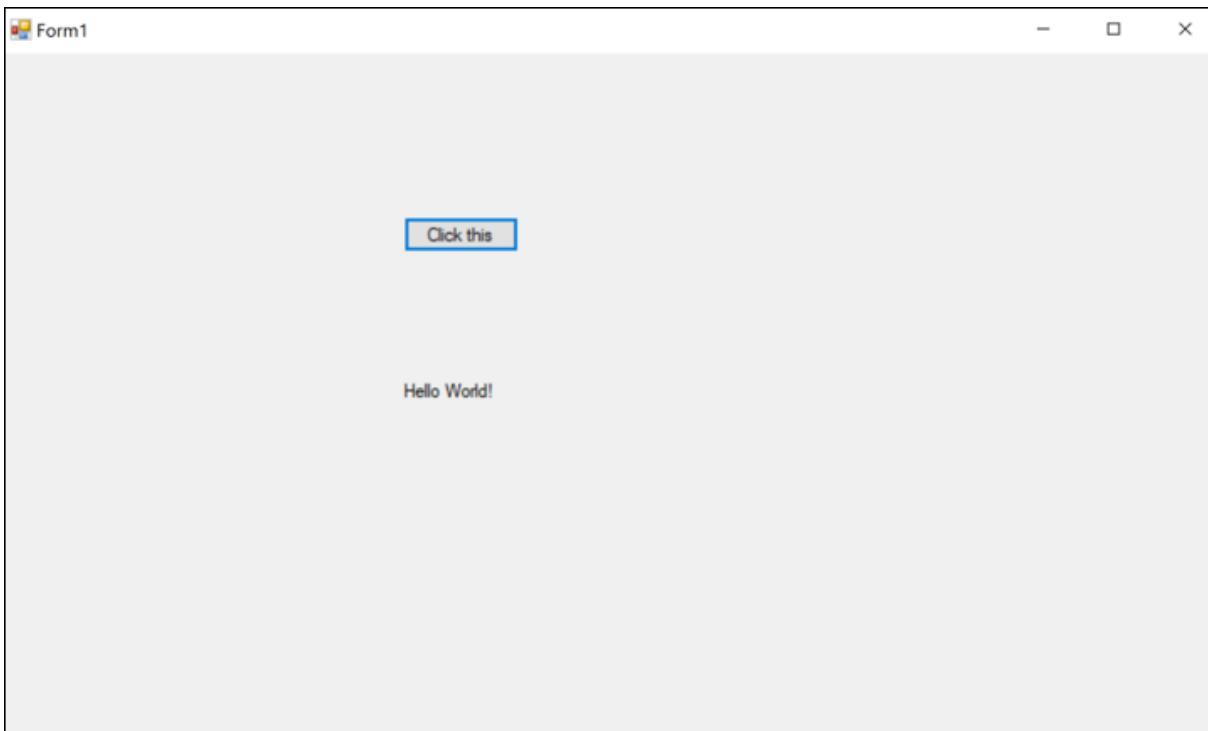
- Chạy ứng dụng
 - Chọn nút Start để chạy ứng dụng



Hình 4.74. Nút chạy ứng dụng trong Visual Studio 2022

- Chọn nút Click this trong hộp thoại Form1. Lưu ý rằng văn bản label1 thay đổi thành Hello World !.

CHƯƠNG 4. THI CÔNG HỆ THỐNG



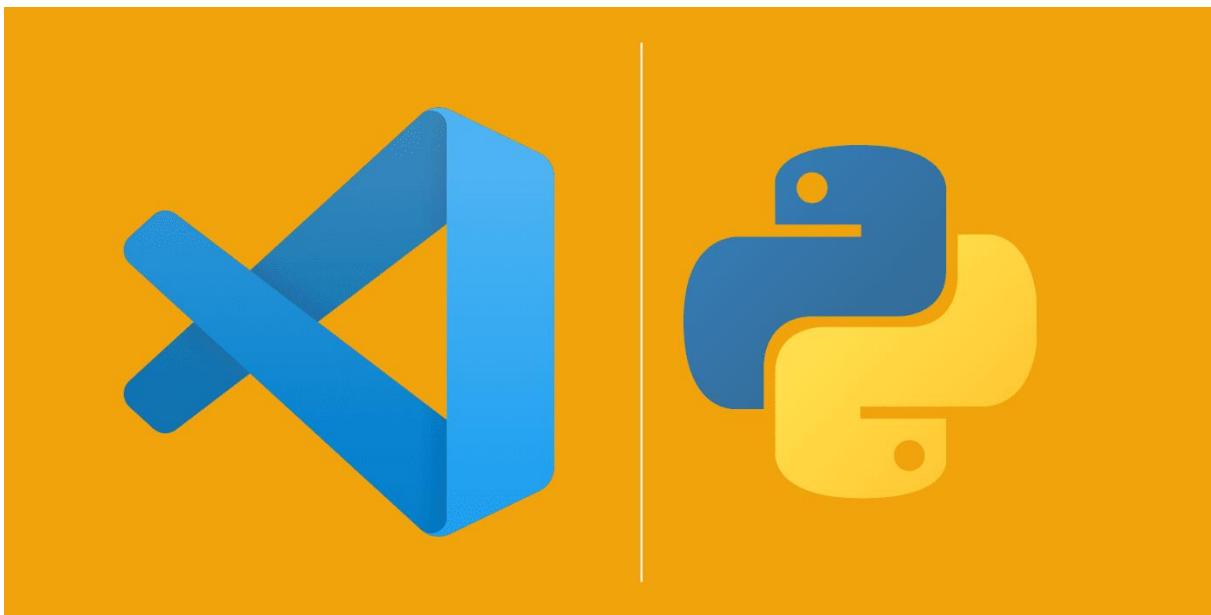
Hình 4.75. Ứng dụng nút nhấn đơn giản trong Visual Studio 2022

- Đóng Form1 để ngưng ứng dụng

4.3.4. Phần mềm lập trình để nhận dạng vật thể

Visual Studio Code là trình biên tập code hoàn toàn miễn phí, dành cho các hệ điều hành macOS, Windows và Linux được phát triển bởi Microsoft. Visual Code Studio hỗ trợ các chức năng như debug, tự hoàn thành mã thông minh, cài tiền mã nguồn, cho phép người dùng thay đổi theme, phím tắt...

CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.76. Logo Visual Studio Code và Python

Download Visual Studio Code từ link chính thức của Microsoft và chọn file cài đặt tương ứng với hệ điều hành mình đang sử dụng.

❖ Extension hỗ trợ lập trình Python trên Visual Studio Code

Để lập trình Python trên Visual Studio Code, cần cài đặt một số extension sau đây.

- Python
 - Extension được Microsoft phát triển, có hỗ trợ gợi nhớ code (suggestion), tự động hoàn thành code, chạy ứng dụng Python và debug.
 - Lệnh cài đặt nhanh: ext install ms-python.python
- Remote Development
 - Extension hỗ trợ code, debug, chạy chương trình Python từ xa
 - Lệnh cài đặt nhanh: ext install ms-vscode-remote.vscode-remote-extensionpack

CHƯƠNG 4. THI CÔNG HỆ THỐNG

❖ Cách cài đặt Extension

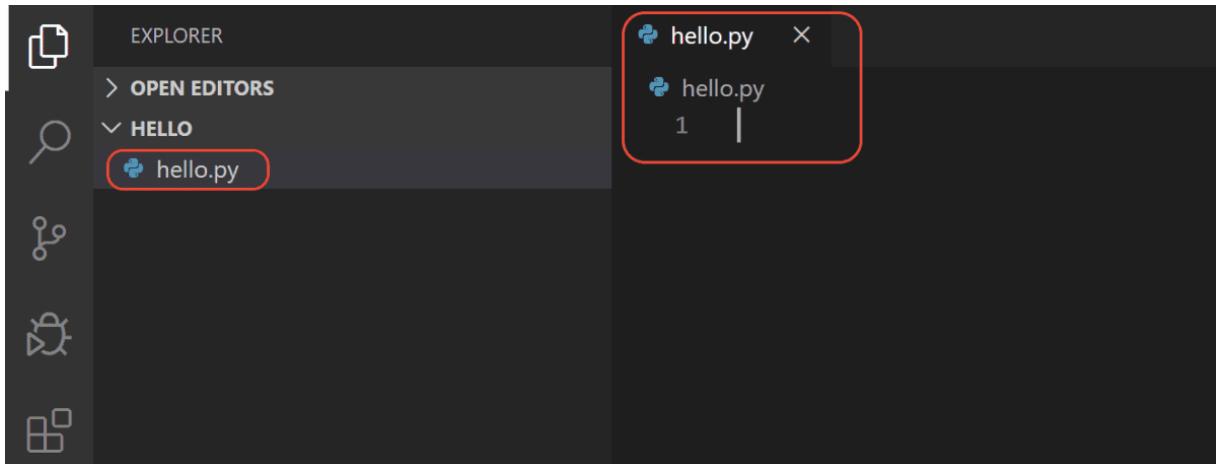
Cài đặt bằng lệnh: bấm tổ hợp phím Ctrl + P, nhập lệnh cài đặt tương ứng và gõ Enter

Cài đặt thông thường: bấm tổ hợp phím Ctrl + Shift + X hoặc click vào biểu tượng Extension, chọn Install để cài đặt.

Hướng dẫn các bước lập trình Python trên Visual Studio cơ bản

❖ Tạo Workspace

- Bước 1: Chọn File > Open Folder > Chọn Project Python > Tạo file hello-world.py
- Bước 2: Chọn File > Save Workspace As. Project đã được lưu, có thể mở lại bằng cách chọn Open Workspace và mở file Project

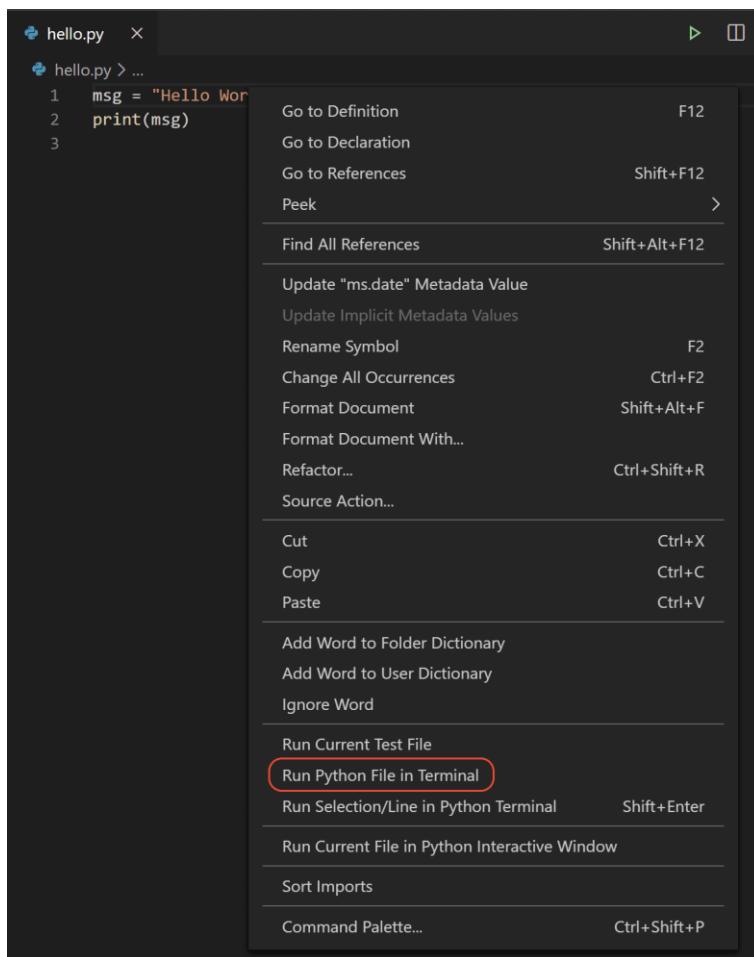


Hình 4.77. Tạo workspace trên Visual Studio Code

❖ Chạy Python Script

- Cách 1: Chạy bằng Terminal của hệ điều hành
 - Click chuột phải vào file vừa tạo, chọn Run Python File in Terminal

CHƯƠNG 4. THI CÔNG HỆ THỐNG



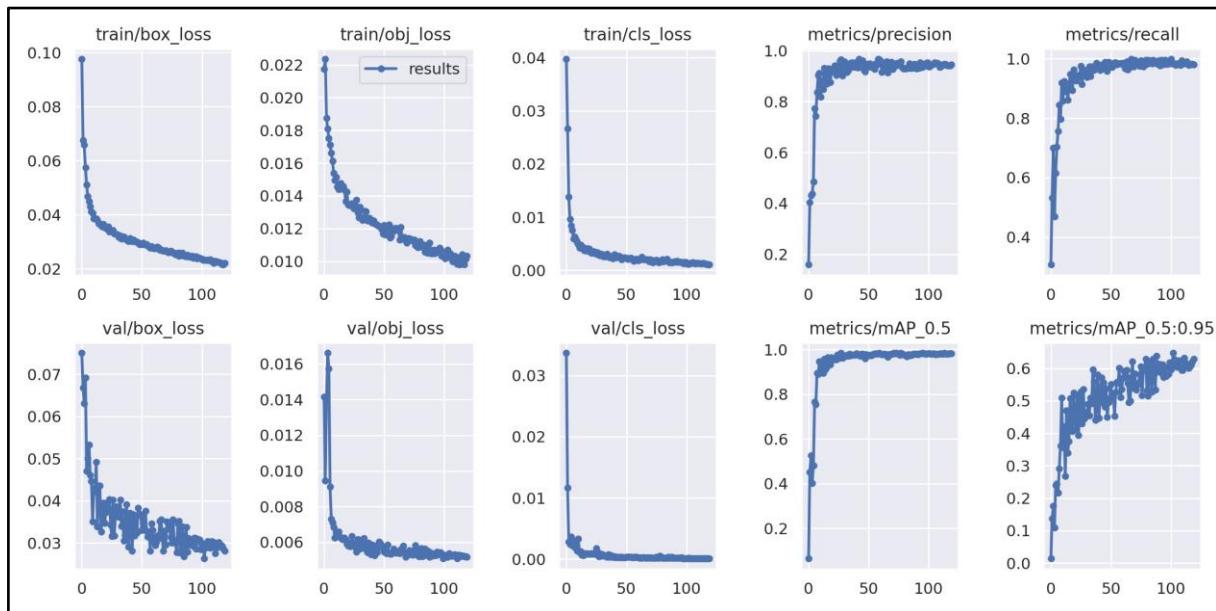
Hình 4.78. Chạy Python file trong terminal

- Cách 2: Chạy bằng Python Interactive Window
 - Cài đặt Jupyter Server bằng command sau
 - Click chuột phải vào script cần chạy và chọn Run Current File in Python Interactive Window

CHƯƠNG 5. KẾT QUẢ, NHẬN XÉT VÀ ĐÁNH GIÁ

5.1. KẾT QUẢ

5.1.1. Kết quả huấn luyện mô hình học sâu



Hình 5.1. Đồ thị thể hiện loss, precision, recall trong quá trình huấn luyện

Có ba loại loss (hàm mất mát) khác nhau được thể hiện trong hình trên: box loss, object loss và class loss.

- “Box loss” là hàm mất mát biểu thị mức độ thuật toán có thể xác định vị trí tâm của một đối tượng và hộp giới hạn được dự đoán bao phủ một đối tượng tốt như thế nào.
- “Object loss” là hàm mất mát của thước đo xác suất một đối tượng tồn tại trong một khu vực quan tâm được đề xuất. Nếu độ tin cậy cao, điều này có nghĩa là cửa sổ hình ảnh có khả năng chứa một đối tượng.
- “Class loss” là hàm mất mát phân loại cho ta một ý tưởng về việc thuật toán có thể dự đoán đúng lớp của đối tượng trả về tốt như thế nào.

CHƯƠNG 5. KẾT QUẢ, NHẬN XÉT VÀ ĐÁNH GIÁ

- “Precision” đại diện cho độ tin cậy của mô hình, “recall” đại diện cho khả năng tìm thấy các phát hiện chính xác (độ nhạy). Nói cách khác “precision” là số lần dự đoán đúng trên tổng số lần dự đoán, “recall” là số lần dự đoán đúng trên tổng số đúng thực sự. “Precision” và “recall” được tính như sau

$$precision = \frac{TP}{TP + FP} \quad recall = \frac{TP}{TP + FN}$$

TP (true positive) là số lần mô hình dự đoán đúng là đúng.

FP (false positive) là số lần mô hình dự đoán đúng nhưng là sai.

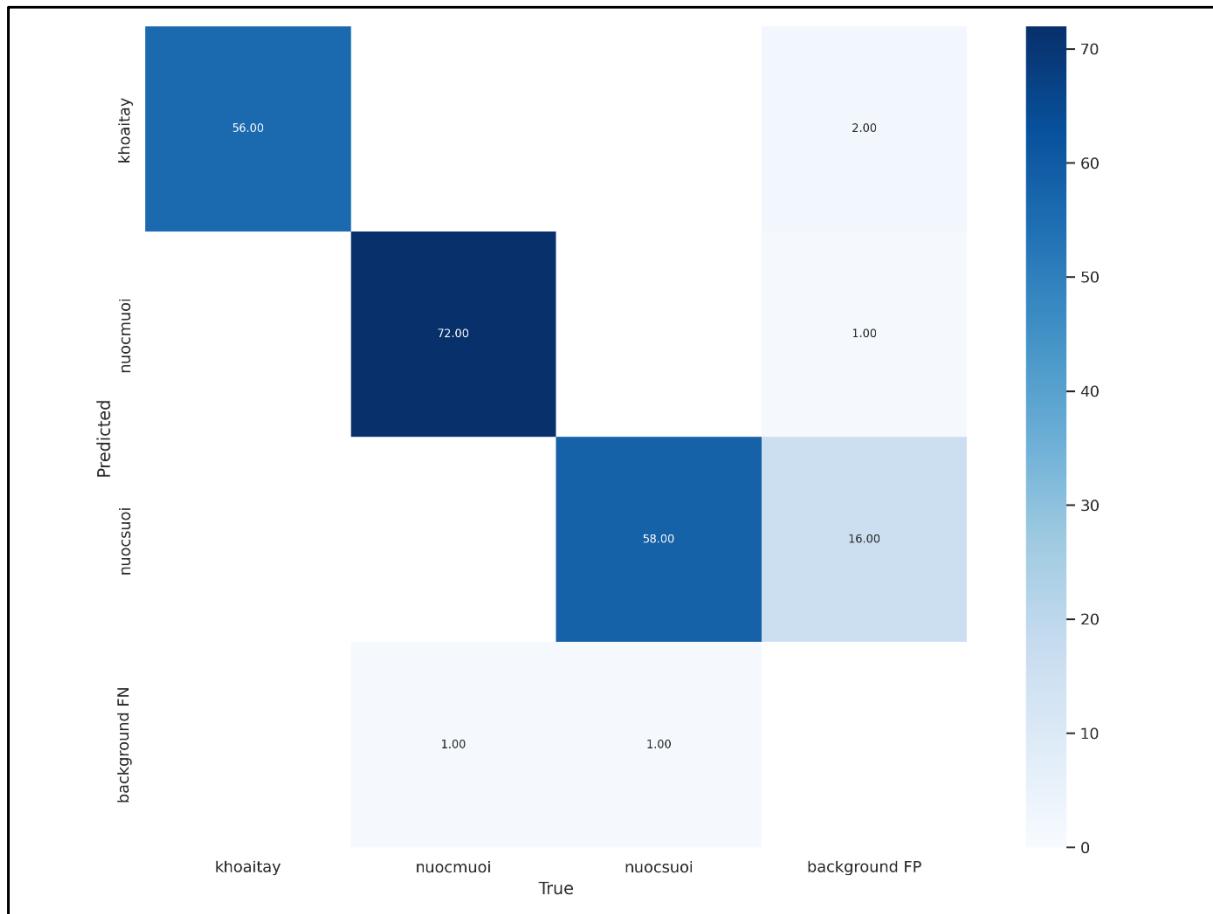
FN (false negative) là số lần mô hình dự đoán sai nhưng là đúng.

Sau 120 epochs cả ba hàm mất mát box loss, obj loss, cls loss đều hội tụ về gần 0, chỉ số “precision” và “recall” cao, qua đó cho thấy mô hình đã học tốt trên tập huấn luyện.

5.1.2. Đánh giá mô hình học sâu trên tập dữ liệu

Để biết mô hình học sâu hoạt động tốt hay không chỉ dựa vào kết quả đánh giá trên tập dữ liệu huấn luyện thì chư thể đưa ra kết luận cuối cùng. Cần đánh giá mô hình trên tập dữ liệu mà mô hình chưa được học, bằng cách đó mới cho ta thấy mô hình có hoạt động thực sự tốt không, có bị overfitting (hiện tượng mô hình chỉ hoạt động tốt trên tập dữ liệu huấn luyện) không.

CHƯƠNG 5. KẾT QUẢ, NHẬN XÉT VÀ ĐÁNH GIÁ



Hình 5.2. Confusion matrix

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95
All	148	188	0.95	0.98	0.982	0.648
khoaitay	148	56	1	1	0.995	0.674
nuocmuoi	148	73	1	0.99	0.995	0.644
nuocsuoi	148	59	0.86	0.97	0.956	0.624

Bảng 5.1. Bảng thống kê đánh giá trên tập thử

Dựa vào hình trên ta thấy rằng hàng background FN có 1 trường hợp xảy ra điều này có nghĩa là trên tập dữ liệu thử mô hình có nhận sai các class (nuocmuoi, nuocsuoi) thành background. Không có trường hợp nào nhận nhầm giữa khoaitay, nuocmuoi và nuocsuoi. Ở cột background FP có khá nhiều trường hợp bị nhầm lẫn với background.

CHƯƠNG 5. KẾT QUẢ, NHẬN XÉT VÀ ĐÁNH GIÁ

Qua đó cho ta thấy rằng mô hình hoạt động chỉ ở mức khá tốt ở công việc phân loại giữa các class với nhau.

mAP@.5 là độ chính xác mà vùng đề xuất mô hình đưa ra thực sự có một đối tượng trong đó với ngưỡng IOU là 0.5 nghĩa là chỉ lấy những vùng đề xuất mà IOU giữa nó và ground truth (vùng thực) lớn hơn 0.5.

mAP@.5:.95 tương tự như mAP@.5 tuy nhiên sẽ được tính trung bình các ngưỡng IOU khác nhau từ 0.5 đến 0.95.

Dựa vào bảng trên các chỉ số Precision, Recall, mAP@.5, mAP@.5:.95 ở mức cao điều đó thể hiện rằng mô hình hoạt động tốt trên tập thử, đưa ra những vùng đề xuất tốt và dự đoán chính xác khá cao.

5.1.3. Kết quả nghiên cứu

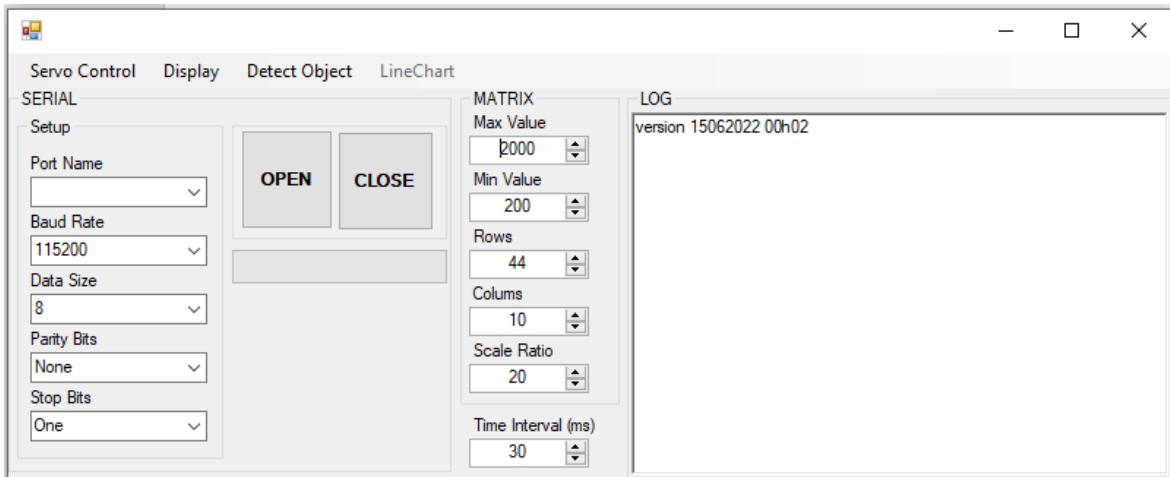
❖ Trong vòng 12 tuần làm đồ án nhóm nghiên cứu đã đạt được các kết quả như sau

- Nghiên cứu về vi điều khiển STM32 và dòng tương tự là STM8.
- Giao tiếp được với máy tính qua UART
- Thiết kế và hiểu và lập trình giao diện winform bằng C#
- Tìm hiểu được cách thiết kế và sử dụng các linh kiện trong mạch công suất và mạch điều khiển.
- Tìm hiểu và biết các sử dụng phần mềm như phần mềm thiết kế mạch chuyên nghiệp Altium Designer, phần mềm lập trình vi điều khiển CubeIDE và phần mềm Visual Studio để lập trình giao diện winform.
- Thi công thành công mạch điều khiển trung tâm và mạch điều khiển động cơ và mạch quét ma trận cảm biến.

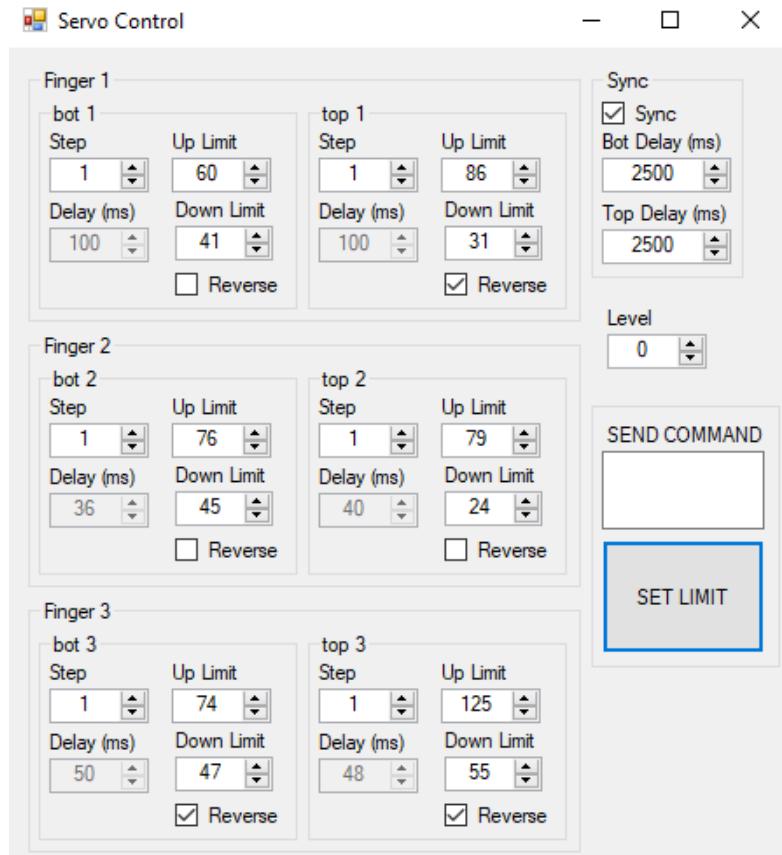
5.1.4. Kết quả thi công

CHƯƠNG 5. KẾT QUẢ, NHẬN XÉT VÀ ĐÁNH GIÁ

a. Giao diện người dùng



Hình 5.3. Giao diện chính của phần mềm khi mở phần mềm



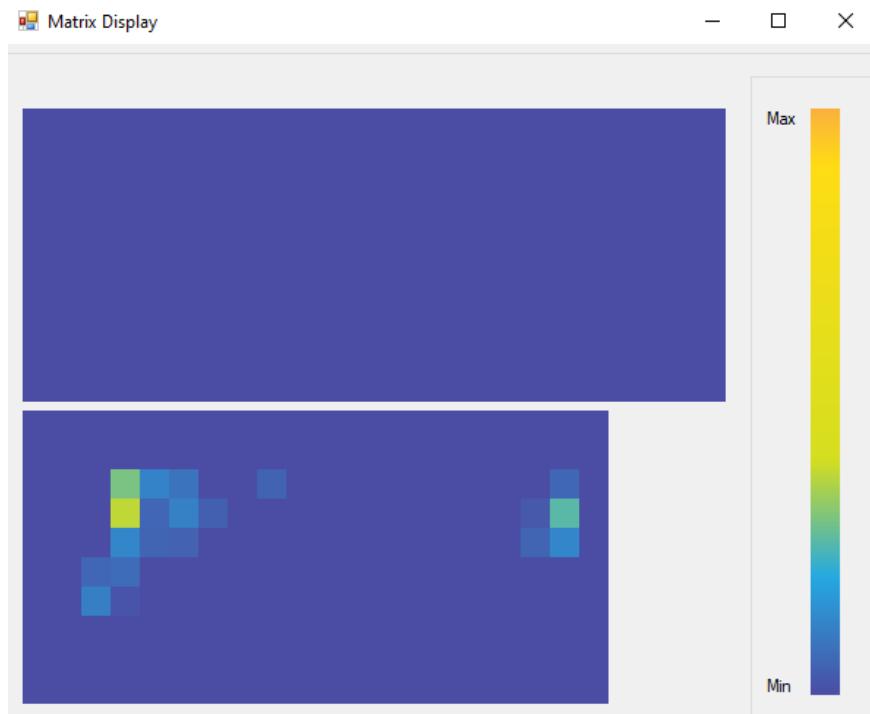
Hình 5.4. Giao diện điều khiển động cơ servo

CHƯƠNG 5. KẾT QUẢ, NHẬN XÉT VÀ ĐÁNH GIÁ

Người dùng có thể tùy chỉnh các thông số cho các động cơ servo và gửi lệnh điều khiển servo tới vi điều khiển STM32



Hình 5.5. Giao diện hiển thị đồ họa ma trận cảm biến dưới dạng cánh tay

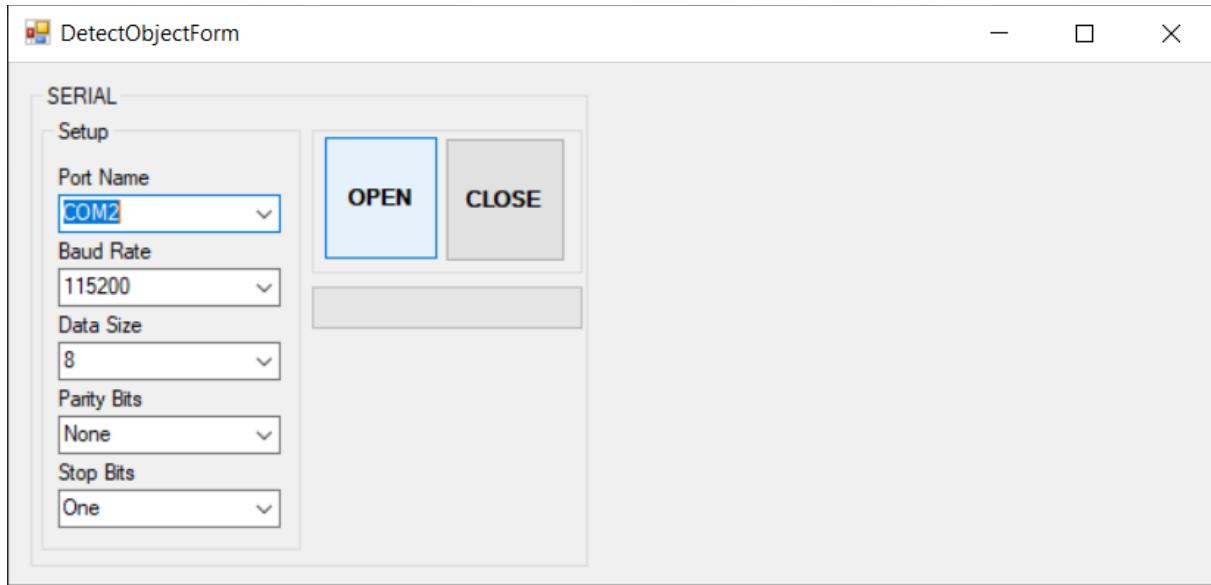


Hình 5.6. Giao diện hiển thị đồ họa ma trận cảm biến (dùng để debug)

CHƯƠNG 5. KẾT QUẢ, NHẬN XÉT VÀ ĐÁNH GIÁ

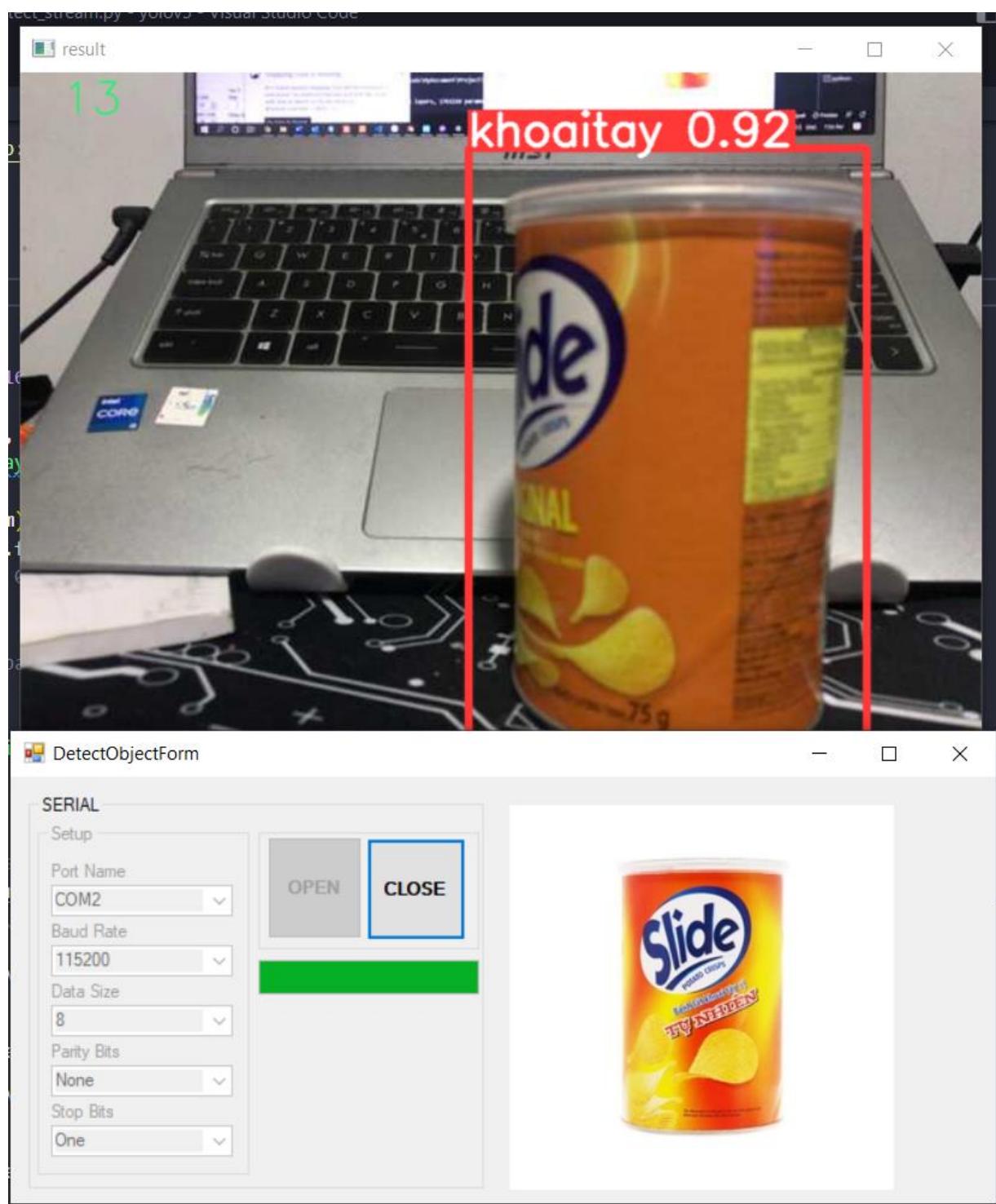
Hình 5.7.

Đối với chức năng nhận dạng vật thể, khi cửa sổ giao diện hiện ra, vào Detect Object. Cửa sổ nhận diện hiện ra như sau



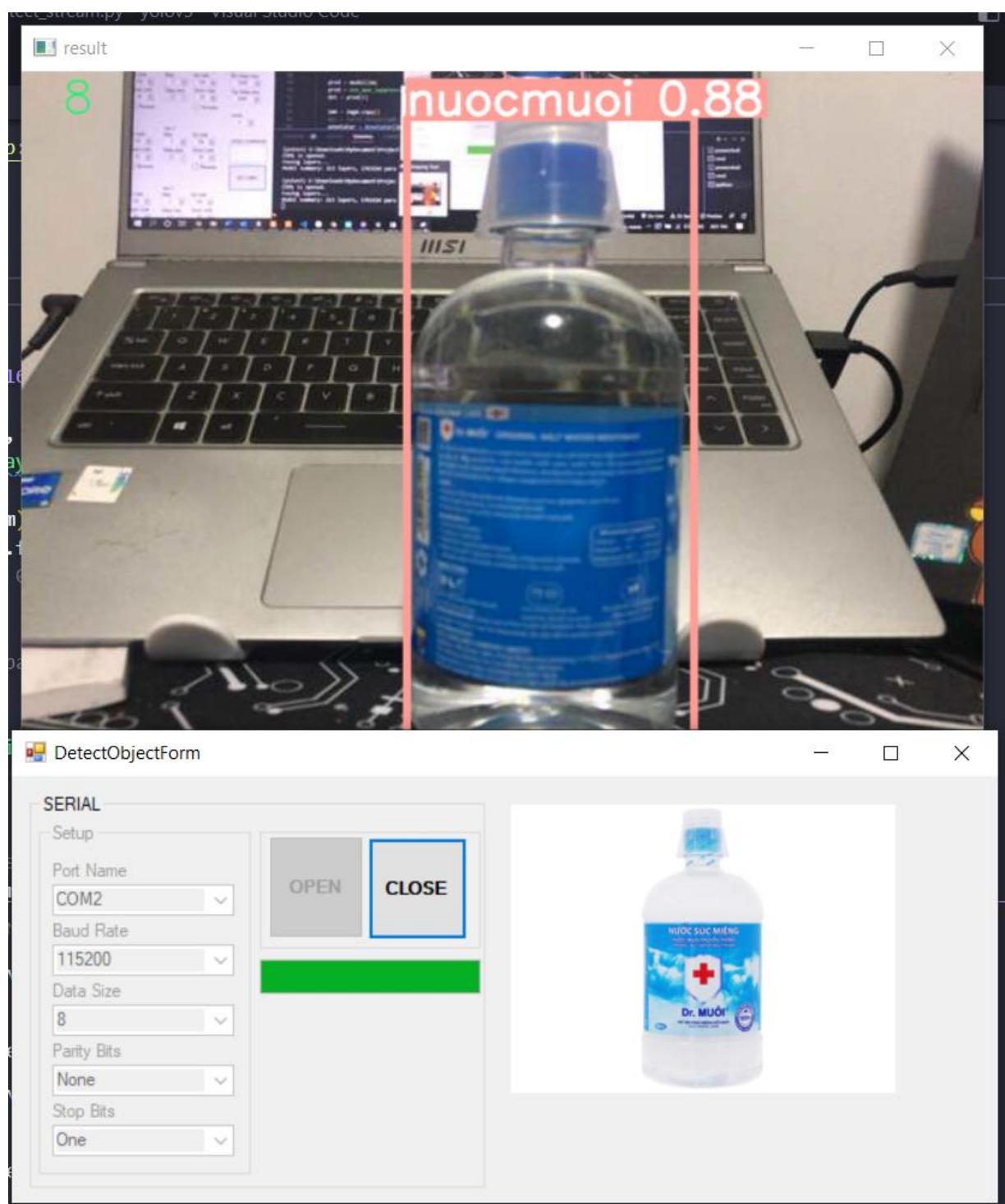
Hình 5.8. Cửa sổ nhận dạng vật thể của giao diện

CHƯƠNG 5. KẾT QUẢ, NHẬN XÉT VÀ ĐÁNH GIÁ



Hình 5.9. Nhận dạng vật thể hủ snack khoai tây và hiển thị trên giao diện

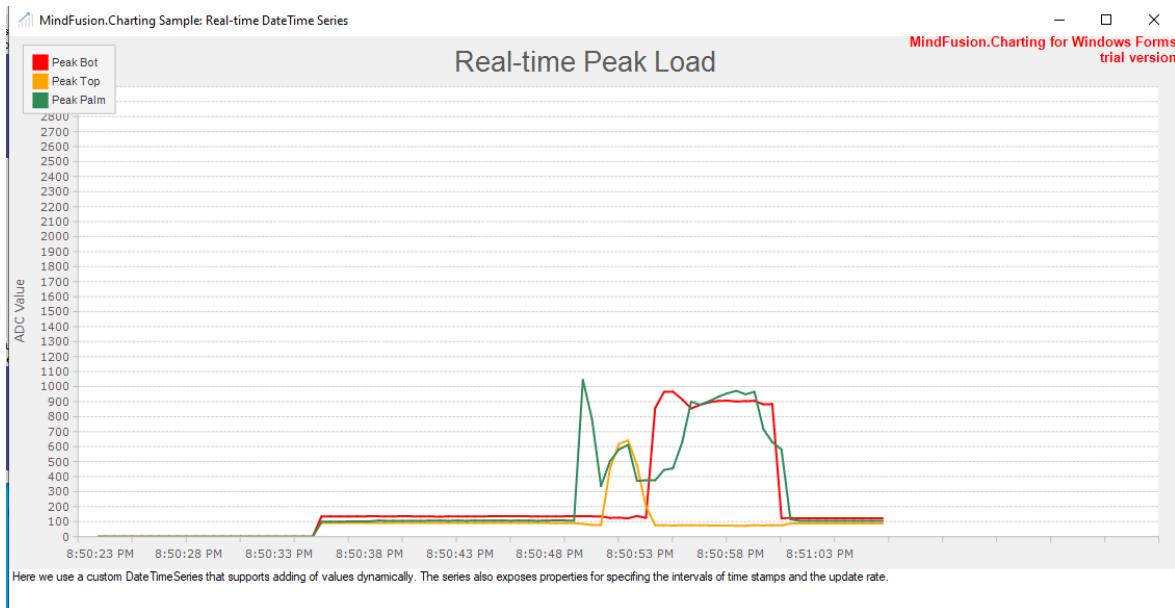
CHƯƠNG 5. KẾT QUẢ, NHẬN XÉT VÀ ĐÁNH GIÁ



Hình 5.10. Nhận dạng vật thể chai nước muối súc miệng và hiển thị trên giao diện

CHƯƠNG 5. KẾT QUẢ, NHẬN XÉT VÀ ĐÁNH GIÁ

Sau khi Port được kết nối, dữ liệu từ phần mềm nhận dạng sẽ truyền cho giao diện mỗi khi có sự thay đổi về vật thể. Từ đó giao diện sẽ hiển thị và gửi thông tin cho vi điều khiển để điều khiển robot.



Hình 5.11. Giao diện linegraph hiển thị giá trị tối đa của các đốt trên, đốt dưới và lòng bàn tay

❖ **Kết quả điều khiển servo, đo và hiển thị giá trị cảm biến**

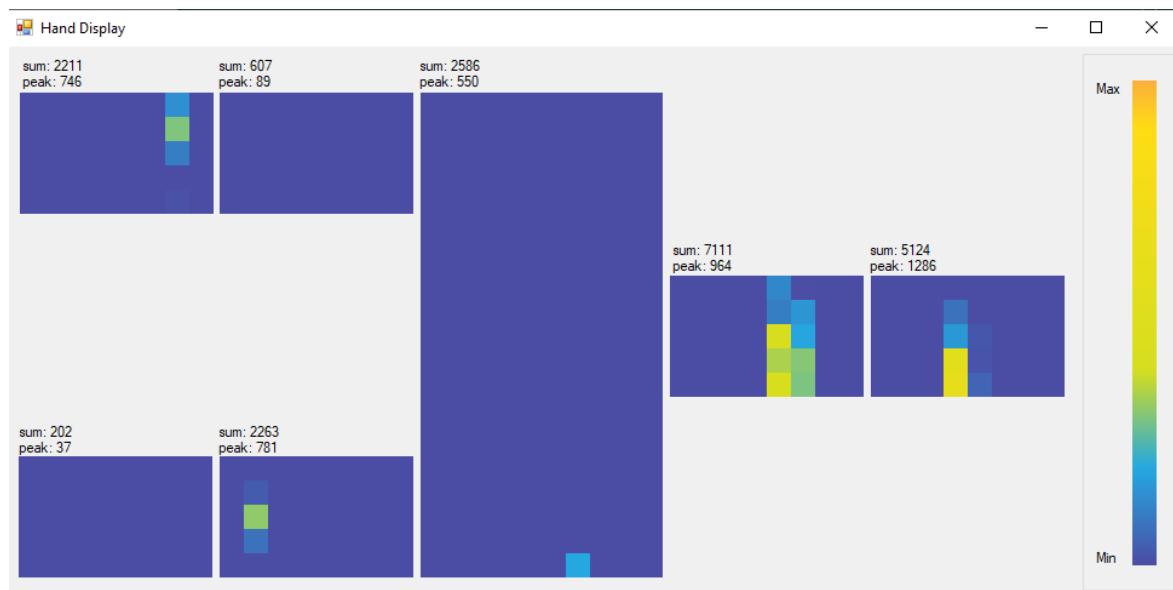
- Thực hiện với chai nước suối

CHƯƠNG 5. KẾT QUẢ, NHẬN XÉT VÀ ĐÁNH GIÁ

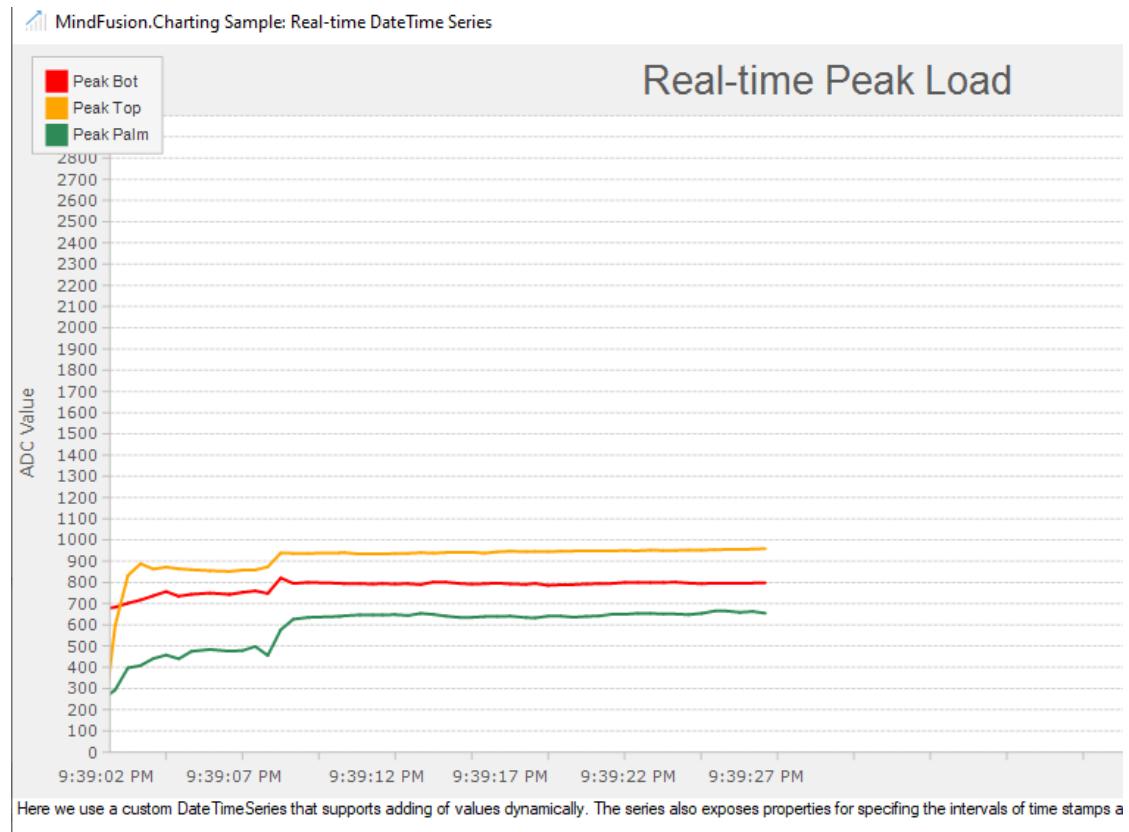


Hình 5.12. Robot thao tác với chai nước suối

CHƯƠNG 5. KẾT QUẢ, NHẬN XÉT VÀ ĐÁNH GIÁ

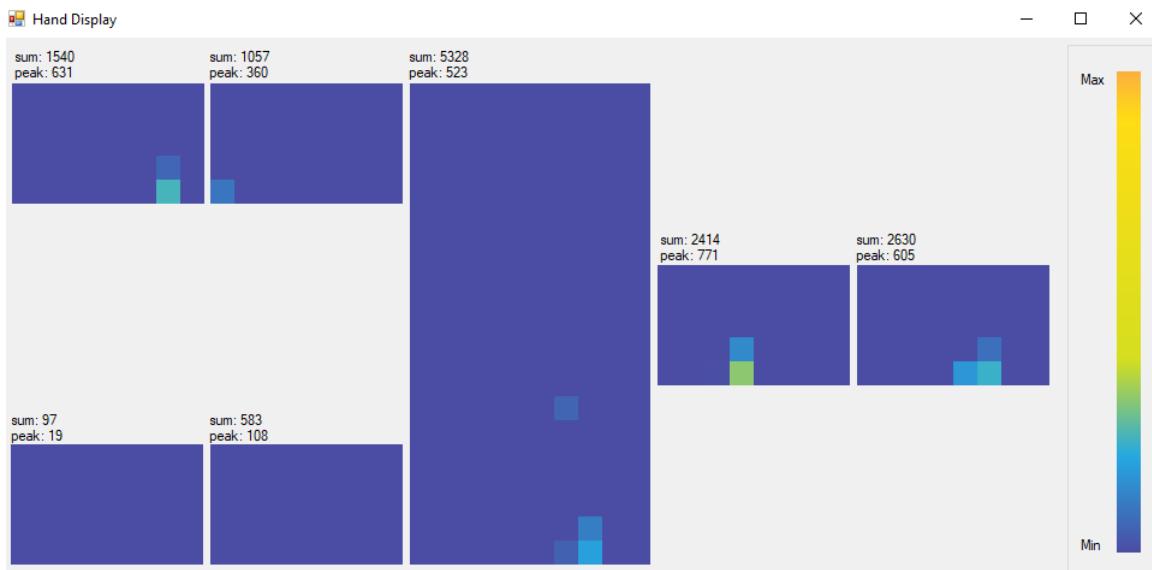


Hình 5.13. Kết quả khi cho bàn tay bóp toàn lực vào chai nước suối

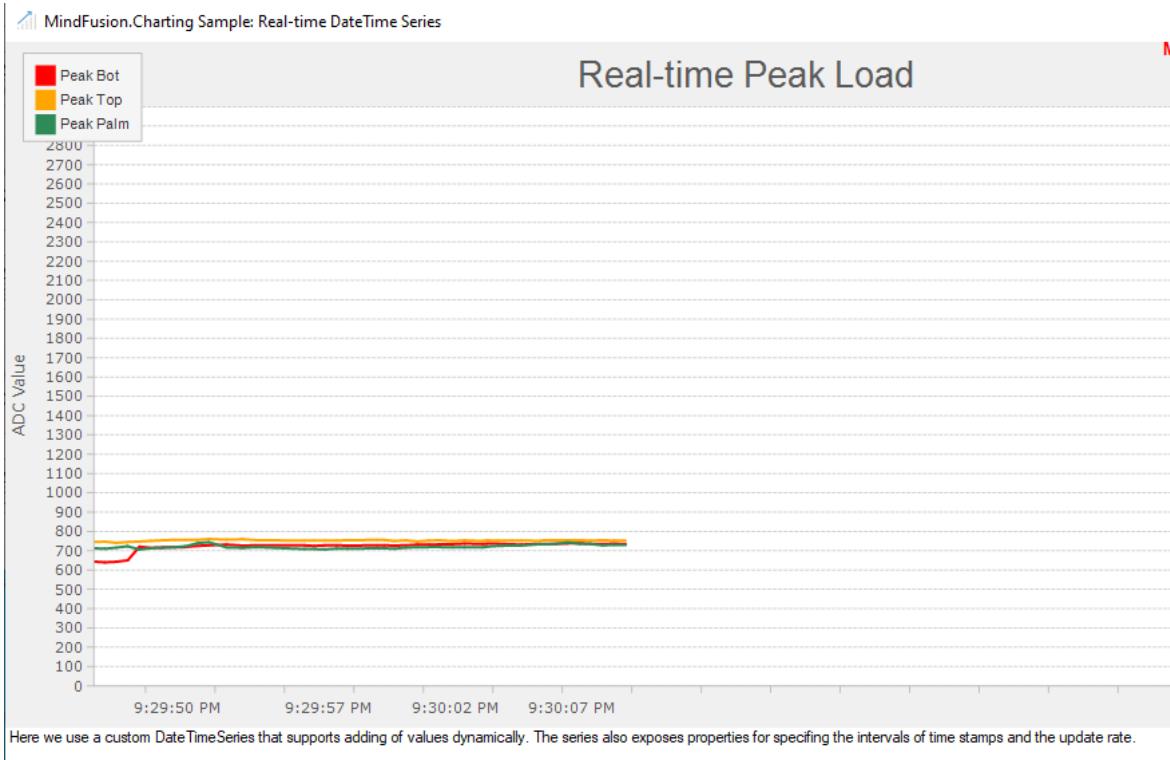


Hình 5.14. Kết quả linechart khi cho bàn tay bóp toàn lực vào chai nước suối

CHƯƠNG 5. KẾT QUẢ, NHẬN XÉT VÀ ĐÁNH GIÁ



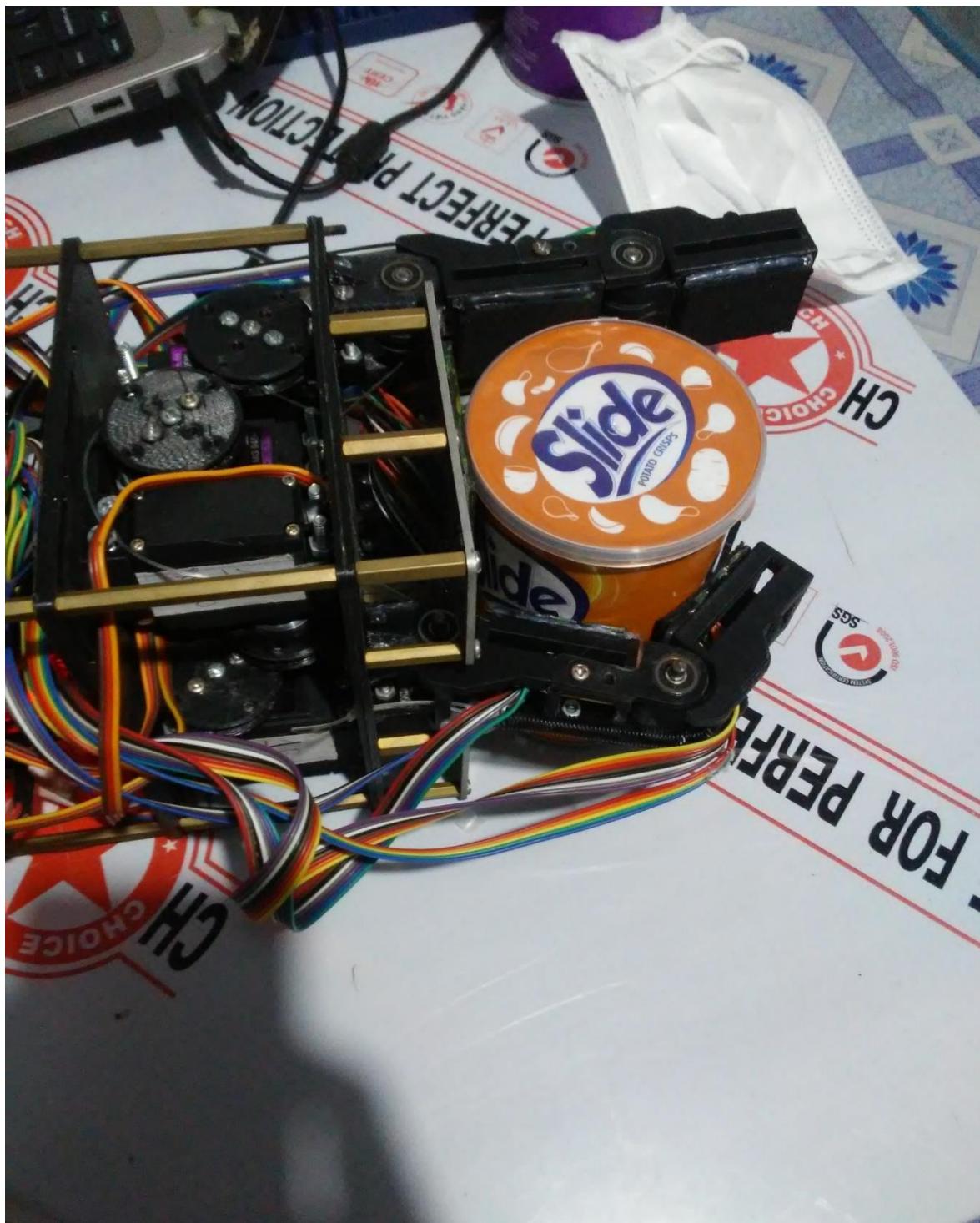
Hình 5.15. Kết quả khi cho bàn tay bóp với lực cấp độ 7 vào chai nước suối



Hình 5.16. Kết quả linechart khi cho bàn tay bóp với lực cấp độ 7 vào chai nước suối

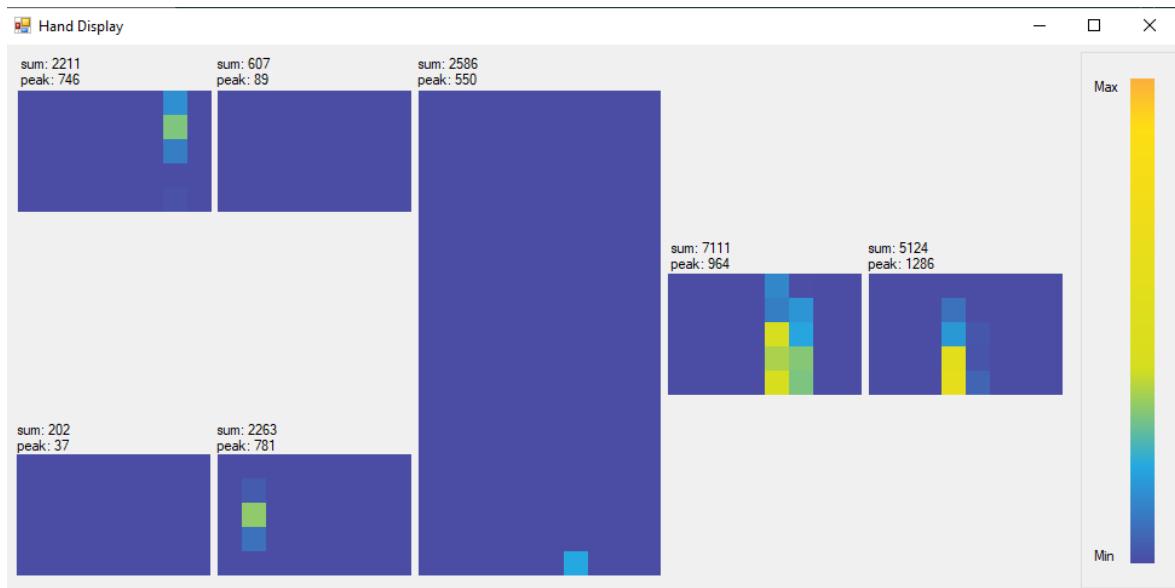
- Thực hiện với hủ snack khoai tây

CHƯƠNG 5. KẾT QUẢ, NHẬN XÉT VÀ ĐÁNH GIÁ

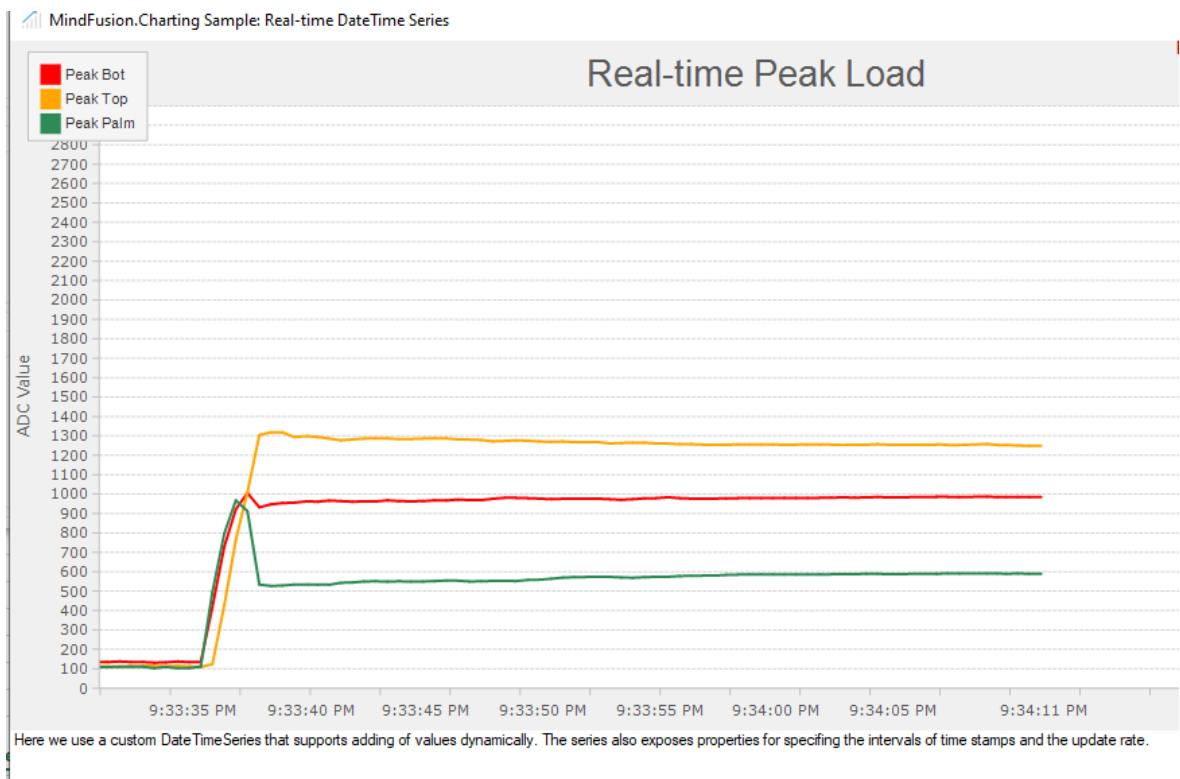


Hình 5.17. Cánh tay robot thao tác với hủ snack khoai tây

CHƯƠNG 5. KẾT QUẢ, NHẬN XÉT VÀ ĐÁNH GIÁ

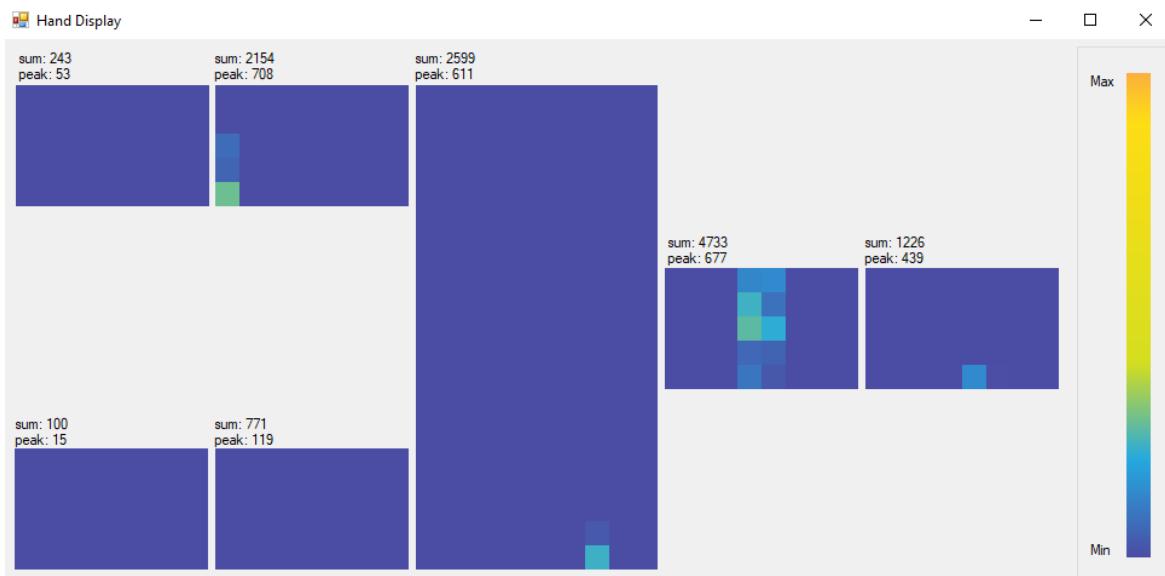


Hình 5.18. Kết quả khi cho bàn tay bóp toàn lực vào hủ snack khoai tây

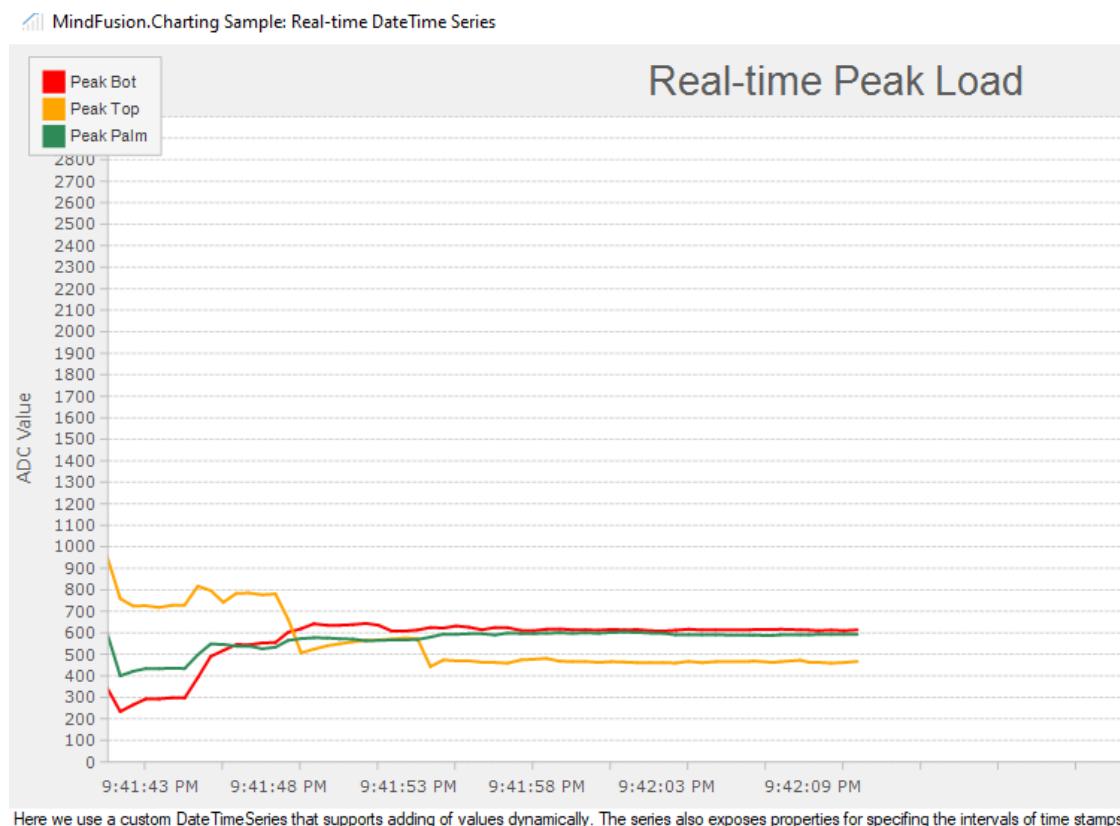


Hình 5.19. Kết quả linechart khi cho bàn tay bóp toàn lực vào hủ snack khoai tây

CHƯƠNG 5. KẾT QUẢ, NHẬN XÉT VÀ ĐÁNH GIÁ



Hình 5.20. Kết quả khi cho bàn tay bóp với lực cấp độ 6 vào hủ snack khoai tây



Hình 5.21. Kết quả linechart khi cho bàn tay bóp với lực cấp độ 6 vào hủ snack khoai tây

CHƯƠNG 5. KẾT QUẢ, NHẬN XÉT VÀ ĐÁNH GIÁ

5.2. NHẬN XÉT ĐÁNH GIÁ

❖ Các nội dung đạt được so với yêu cầu đặt ra

- Phần mềm và giao diện trên máy tính để nhận diện và điều khiển tay robot
- Hiển thị dữ liệu ma trận cảm biến lực và có thể tùy chỉnh giới hạn trên dưới để ma trận hiện thị theo ý người dùng
- Cung cấp phần điều khiển và cài đặt chi tiết dễ dàng cho từng động cơ servo để có thể tinh chỉnh mang lại hoạt động của robot tối ưu hơn
- Cung cấp đồ thị phân tích lực trên các miếng cảm biến
- Hiển thị vật thể được thao tác lên giao diện
- Lập trình các tác vụ hoạt động gần như song song để mang lại tính đồng bộ và độ trễ thấp nhất có thể trong hệ thống.
- Tùy chọn cho robot hoạt động ở 2 chế độ tự động và chỉnh tay

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1. KẾT LUẬN

Chương kết quả đã phản ánh sinh viên chỉ thực hiện được robot hoạt động với các vật thể với khối lượng định sẵn. So mục tiêu đặt ra ở chương 1 thì nhiệm vụ đã được hoàn thành. Các nội dung mà nhóm đã thực hiện được đó là thiết kế và thi công được một robot có thể điều khiển qua giao diện người dùng và ứng dụng rộng rãi trong nền công nghiệp robot để gắp vật với lực vừa phải và không làm biến dạng vật thể. Nhìn chung đê tài chỉ mới hoàn thành ở mức khá. Tuy nhiên, chúng tôi vẫn chưa thể tạo ra một hệ thống ổn định do phần tính toán thiết kế còn nhiều sai sót, phần điều khiển nhiều động cơ, mỗi động cơ phụ trách riêng từng khớp phù hợp với dữ liệu cảm biến vẫn còn quá phức tạp và chưa kịp tối ưu.Thêm vào đó, phần cơ khí của robot với nhiều khuyết điểm vẫn còn là vấn đề lớn ảnh hưởng bao quát đến đê tài.

Trong quá trình làm đồ án, chúng tôi đã rút ra được nhiều kinh nghiệm để tạo ra một sản phẩm hoàn thiện như: làm việc nhóm, tìm kiếm tài liệu, đầu tư thời gian, vật liệu chế tạo cảm biến, vật liệu sử dụng trong cơ khí, hiểu biết về các linh kiện, thiết kế board mạch, tạo ra giao diện cho người dùng, ứng dụng trí tuệ nhân tạo...

6.2. HƯỚNG PHÁT TRIỂN

❖ Một số hướng phát triển để hoàn thiện đê tài như là:

- Cải thiện thiết kế cơ khí để các khớp di chuyển mượt hơn, đầu tư động cơ servo với độ bền và khỏe hơn, giảm kích thước bàn tay robot, nâng cao chất lượng cũng như độ bền các mối nối và thảm mĩ.
- Kết nối không dây để tiện cho việc gỡ lỗi cũng như cập nhật phần mềm.
- Thêm cảm nhận về nhiệt độ cho xúc giác

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

- Thiết kế các mạch dẻo (Flex PCB) để tăng tính linh động cho cảm biến xúc giác.
- Thêm cảm biến gia tốc để nhận biết về không gian và từ đó có thể cảm nhận hình dạng của vật thể khi robot được cầm nắm.
- Có thể ứng dụng vào môi trường y tế hỗ trợ hoặc khuếch đại xúc giác.
- Có thể ứng dụng và các sản phẩm thường ngày như ghế, nệm, lót giày để hỗ trợ thu thập data người dùng trong thời đại mà các vật dụng trong văn phòng đều được chú trọng và dần được công thái học hóa để phù hợp với mọi người.

TÀI LIỆU THAM KHẢO

TÀI LIỆU THAM KHẢO

- [1] ARM Việt Nam, Cấu trúc cơ bản của STM32_ARM Cortex M3, 18/3/2010.
- [2] Phan Văn Hoàn, Giáo trình vi xử lý nâng cao, Trường ĐH Sư Phạm Kỹ Thuật Tp.HCM, 2015.
- [3] STM32F103xx Datasheet, www.st.com/resource/en/datasheet/stm32f103rg.pdf
- [4] Velostat Sheet, <https://www.farnell.com/datasheets/1815591.pdf>
- [5] The FreeRTOS Reference Manual,
https://www.freertos.org/Documentation/FreeRTOS_Reference_Manual_V9.0.0.pdf
- [6] Zhanat Kappassov, Juan Antonio Corrales Ramon, Véronique Perdereau, Tactile sensing in dexterous robot hands – Review, Tháng 7/2015.
- [7] Marco Costanzo, Giuseppe De Maria, Ciro Natale, Salvatore Pirozzi, Design of a Force/Tactile Sensor for Robotic Grippers, 25/07/2019.
- [8] Roberto Calandra, Andrew Owens, Dinesh Jayaraman, Justin Lin, Wenzhen Yuan, Learning to Grasp and Regrasp using Vision and Touch, 2018
- [9] Marco Costanzo, Giuseppe De Maria , Ciro Natale, Salvatore Pirozzi, Design and Calibration of a Force/Tactile Sensor for Dexterous Manipulation, 2019
- [10] Stefano Stassi, Valentina Cauda, Giancarlo Canavese, Candido Fabrizio Pirri, Flexible Tactile Sensing Based on Piezoresistive Composites, 2014
- [11] Ayoosh Kathuria, How to Train YOLO v5 on a Custom Dataset, <https://blog.paperspace.com/train-yolov5-custom-data/>
- [13] Kukil, Object Detection using YOLOv5 and OpenCV DNN in C++ and Python, <https://learnopencv.com/object-detection-using-yolov5-and-opencv-dnn-in-c-and-python/>

PHỤ LỤC

PHỤ LỤC

Tất cả code, mô hình, bản vẽ và tài liệu liên qua của robot được đính kèm trong link sau

<https://drive.google.com/drive/folders/10Jv1oGPRboKWPAjANsQZ4bPpYHkrqxoJ?usp=sharing>