

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA**



BÀI TẬP LỚN MÔN THIẾT KẾ HỆ THỐNG NHÚNG

ĐỀ TÀI:

PET FEEDER SYSTEM

LỚP L02 --- NHÓM 6 --- HK251

TÊN NHÓM: DÂY GIỮA BÀN ĐẦU

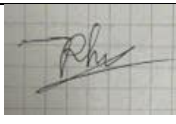
NGÀY NỘP: 8/1/2025

Giảng viên hướng dẫn: Th.S TRẦN HOÀNG QUÂN

Sinh viên thực hiện	Mã số sinh viên	Tỉ lệ đóng góp
Lý Minh Thiên	2313214	100%
Nguyễn Hoàng Khánh	2311516	100%
Nguyễn Hoàng Anh Kiệt	2311774	100%
Huỳnh Lương Nhựt Phát	2312571	100%
Trương Phước Bình	2310315	100%

Thành phố Hồ Chí Minh – 2026

BÁO CÁO KẾT QUẢ LÀM VIỆC NHÓM

STT	MSSV	Họ và tên	Nhiệm vụ được phân công	% Điểm BTL	Ký tên
1	2313214	Lý Minh Thiên	Thiết kế phần cứng, thiết kế phần mềm	100%	
2	2311516	Nguyễn Hoàng Khánh	Thiết kế phần cứng, thiết kế phần mềm	100%	
3	2311774	Nguyễn Hoàng Anh Kiệt	Thiết kế phần cứng, thiết kế phần mềm	100%	
4	2312571	Huỳnh Lương Nhựt Phát	Thiết kế phần mềm	100%	
5	2310315	Trương Phước Bình	Thiết kế phần mềm	100%	

Họ và tên nhóm trưởng: Lý Minh Thiên

Số ĐT: 0767229573

Email: thien.lyminh@hcmut.edu.vn

Nhận xét của GV:

.....

.....

.....

.....

GIẢNG VIÊN

(Ký và ghi rõ họ, tên)

Trần Hoàng Quân

NHÓM TRƯỞNG

(Ký và ghi rõ họ, tên)



MỤC LỤC

LỜI CẢM ƠN.....	4
I. GIỚI THIỆU ĐỀ TÀI	5
1.1 Tổng quan đề tài.....	5
1.2 Nhiệm vụ đề tài	5
1.3 Kế hoạch dự án.....	5
1.3.1 Phân công.....	5
1.3.2 Kế hoạch dự án	7
II. ĐẶC TẢ HỆ THỐNG	8
2.1 Đặc tả sản phẩm	8
2.2 Đặc tả kỹ thuật	9
2.3 Hardware Specification	10
2.4 Software Specification	11
2.5 Test Specification.....	11
III. CÁC VẤN ĐỀ THIẾT KẾ.	12
3.1 Constraint Issues (Ràng buộc thiết kế)	12
3.1.1 Ràng buộc về chi phí	13
3.1.2 Ràng buộc về độ an toàn.....	13
3.1.3 Ràng buộc độ tin cậy và chính xác	13
3.1.4 Ràng buộc về bộ nguồn	14
3.1.5 Ràng buộc về kích thước	14
3.2 Functional Issues (Vấn đề chức năng)	14
3.2.1 Ảnh hưởng đến sức khỏe vật nuôi	14
3.2.2. An toàn cơ học	15
3.2.3. Vệ sinh và Môi trường.....	15
3.3 Realtime issue	15
3.4 Concurrent Issues (Vấn đề xử lý đồng thời)	16
3.5 Reactive Issues (Vấn đề phản ứng).....	16
IV. Thiết kế phần cứng	17
4.2 Thông số các linh kiện	18
4.2.1 MCU ATmega16	18
4.2.2 Keypad 4x4	19

4.2.3 RTC DS1307.....	20
4.2.4 LCD 16x2	21
4.2.5 Servo sg90.....	22
4.2.6 Loadcell	23
4.3 Thiết kế sơ đồ khối.....	25
4.4 Thiết kế sơ đồ nguyên lí.....	25
4.4.1 Khối vi xử lý	25
4.4.2 Khối LCD 16x2	26
4.4.3 Khối keypad 4x4.....	27
4.4.4 Khối buzzer và motor	27
4.4.5 Khối HX711 và loadcell	28
4.4.6 Khối RTC DS1307.....	28
4.4.7 Sơ đồ nguyên lí toàn mạch	29
4.5 Thiết kế mạch mô phỏng.....	29
4.6 Thiết kế PCB	30
V. Thiết kế phần mềm	31
5.1 Thiết kế các thư viện	31
5.2 Thiết kế hàm main.....	33
5.2.1 Lưu đồ giải thuật.....	33
5.2.2 Code hàm main	33
VI. Kết quả	37
6.1 Kết quả mô phỏng trên proteus	37
6.2 Kết quả thử nghiệm trên breadboard.....	38
6.3 Kết quả trên mạch in.....	39
6.4 Kết quả với mô hình.....	39
6.5 Kết quả thực tế	40
VII. Kết luận	40
7.1 Đánh giá đề tài	40
7.2 Hướng phát triển	41
VIII Tài liệu tham khảo	41

Hình 1: Biểu đồ Gant kế hoạch đề tài.....	8
Hình 2: ATmega16A	19
Hình 3: Keypad 4x4.....	20
Hình 4: RTC DS1307	21
Hình 5: LCD 1602	22
Hình 6: Motor servo sg90	23
Hình 7: Loadcell 5kg	24
Hình 8: HX711	25
Hình 9: Sơ đồ khối phần cứng hệ thống	25
Hình 10: Thiết kế khối MCU.....	26
Hình 11: Thiết kế khối LCD 1602	27
Hình 12: Thiết kế keypad 4x4	27
Hình 13: Thiết kế buzzer và motor	27
Hình 14: Thiết kế loadcell và HX711	28
Hình 15: Thiết kế RTC DS1307	28
Hình 16: Sơ đồ nguyên lí toàn hệ thống.....	29
Hình 17: Thiết kế mô phỏng Proteus	30
Hình 18: Thiết kế PCB	30
Hình 19: 3D của PCB	31
Hình 20: Các thư viện cần thiết kế	31
Hình 21: FLOWchart của thư viện RTC, keypad, servo buzzer	32
Hình 22: Flowchart của thư viện HX711, LCD, I2C.....	32
Hình 23: FLOWchart hàm main.....	33
Hình 24: Kết quả mô phỏng proteus.....	37
Hình 25: Kết quả thử nghiệm trên Breadboard	38
Hình 26: Kết quả thực nghiệm PCB.....	39
Hình 27: Kết quả với mô hình thực nghiệm	39

LỜI CẢM ƠN

Lời đầu tiên, tập thể nhóm chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến Thầy Trần Hoàng Quân. Chúng em vô cùng biết ơn Thầy vì đã luôn tận tình truyền đạt những kiến thức quý báu, từ những nguyên lý nền tảng đến những kỹ thuật chuyên sâu trong suốt quá trình học tập. Sự hướng dẫn tận tâm và những bài giảng của Thầy không chỉ là kim chỉ nam giúp chúng em vượt qua những rào cản về mặt kỹ thuật mà còn là nguồn cảm hứng giúp chúng em hoàn thành dự án này một cách trọn vẹn nhất.

Bên cạnh đó, sự thành công của bài tập lớn này còn đến từ sự nỗ lực, đoàn kết và trách nhiệm của từng thành viên trong nhóm. Chúng em đã cùng nhau thảo luận, chia sẻ những khó khăn và không ngừng học hỏi lẫn nhau để đảm bảo đề tài được hoàn thiện đúng tiến độ và đạt được các yêu cầu kỹ thuật đã đề ra.

Cuối cùng, dù đã đầu tư nhiều tâm huyết để hoàn thành báo cáo một cách chu đáo nhất, song do kinh nghiệm thực tế còn hạn chế, bài tập lớn chắc chắn không tránh khỏi những thiếu sót. Nhóm rất mong nhận được sự phê bình và đóng góp ý kiến từ Thầy để đề tài được hoàn thiện hơn.

Một lần nữa, chúng em xin kính chúc Thầy sức khỏe, hạnh phúc và gặt hái được nhiều thành công trong sự nghiệp giáo dục.

Chúng em xin trân trọng cảm ơn Thầy!

I. GIỚI THIỆU ĐỀ TÀI

1.1 Tổng quan đề tài

Trong bối cảnh cuộc sống hiện đại, nhu cầu chăm sóc thú cưng ngày càng tăng cao, đặc biệt đối với những người có lịch trình bận rộn hoặc thường xuyên vắng nhà. Việc cho thú cưng ăn đúng giờ, đúng lượng đóng vai trò quan trọng trong việc duy trì sức khỏe và thói quen sinh hoạt ổn định cho vật nuôi. Tuy nhiên, phương pháp cho ăn thủ công truyền thống còn tồn tại nhiều hạn chế như phụ thuộc vào thời gian của chủ nuôi, thiếu tính chính xác và khó kiểm soát.

Xuất phát từ thực tế đó, hệ thống cho thú cưng ăn tự động (Pet Feeder System) được đề xuất như một giải pháp ứng dụng công nghệ nhúng nhằm tự động hóa quá trình cho ăn. Hệ thống sử dụng vi điều khiển làm trung tâm điều khiển, kết hợp với các linh kiện phần cứng như động cơ, cảm biến, bộ định thời gian và giao diện người dùng để thực hiện việc cấp thức ăn theo lịch trình đã thiết lập. Ngoài ra, hệ thống có thể được mở rộng với các chức năng thông minh như điều chỉnh khẩu phần, hiển thị trạng thái hoạt động hoặc điều khiển từ xa.

1.2 Nhiệm vụ đề tài

Đề tài Pet Feeder System mang ý nghĩa thực tiễn cao giúp sinh viên vận dụng kiến thức đã học về thiết kế phần cứng, lập trình vi điều khiển, xử lý tín hiệu từ cảm biến và xây dựng hệ thống nhúng hoàn chỉnh. Thông qua việc thực hiện đề tài, sinh viên có cơ hội tiếp cận quy trình thiết kế một sản phẩm nhúng thực tế, từ khâu phân tích yêu cầu, thiết kế hệ thống đến triển khai và đánh giá kết quả.

Với Pet Feeder System, nhóm thiết kế với hai chế độ chính bao gồm cho ăn hẹn giờ và cho ăn tức thì. Với cho ăn hẹn giờ sẽ cho phép cho ăn 2 buổi tùy vào chủ nuôi và với cho ăn tức thì chỉ cần nhấn nút đồ ăn sẽ tự động rơi. Cùng với đó hệ thống sẽ đính kèm thêm một cảm biến khối lượng để theo dõi lượng thức ăn và báo khi lượng thức ăn hết.

1.3 Kế hoạch dự án

1.3.1 Phân công

HỢP ĐỒNG NHÓM		
Tên nhóm: Dây giữa bàn đầu (6)		Ngày lập nhóm: 04/09/2025
Thành viên	Vai trò	Ghi chú
Lý Minh Thiên	Nhóm trưởng, thiết kế phần cứng, thiết kế phần mềm, tổng hợp báo cáo	100%
Nguyễn Hoàng Khánh	Thiết kế phần cứng, thiết kế phần mềm	100%
Nguyễn Hoàng Anh Kiệt	Thiết kế phần cứng, thiết kế phần mềm, tổng hợp báo cáo	100%
Huỳnh Lương Nhựt Phát	Thiết kế phần mềm, tổng hợp báo cáo	100%
Trương Phước Bình	Thiết kế phần mềm, tổng hợp báo cáo	100%
Phân công nhiệm vụ		Thành viên thực hiện
Nghiên cứu đề tài		Tất cả thành viên
Chọn linh kiện và thiết kế sơ đồ mạch nguyên lí		Lý Minh Thiên
Thiết kế mạch trong phần mềm mô phỏng		Nguyễn Hoàng Khánh Nguyễn Hoàng Anh Kiệt
Thiết kế lưu đồ giải thuật		Huỳnh Lương Nhựt Phát Trương Phước Bình
Thực hiện code và mô phỏng		Tất cả thành viên
Thử nghiệm mạch bằng breadboard		Lý Minh Thiên
Thực hiện mạch in		Lý Minh Thiên Nguyễn Hoàng Khánh
Họp nhóm:		4 giờ chiều thứ 5 hàng tuần
Nội quy nhóm: <ol style="list-style-type: none"> 1. Tất cả các thành viên phải tham gia họp nhóm 2. Ghi nhận tất cả ý kiến từ các thành viên trong nhóm 3. Hoàn thành các nhiệm vụ được giao đúng hạn 4. Thành viên không hoàn thành đúng nội quy nhóm sẽ trừ phần trăm điểm 		

1.3.2 Kế hoạch dự án

Để đảm bảo hoàn thành đề tài "Hệ thống cho thú cưng ăn tự động" (Pet Feeder System) đúng thời hạn và đáp ứng đầy đủ các yêu cầu kỹ thuật, nhóm thực hiện đã xây dựng một lộ trình làm việc chi tiết kéo dài từ ngày 01/09/2025 đến ngày 02/12/2025. Kế hoạch được chia thành bốn giai đoạn chính, đảm bảo sự phối hợp nhịp nhàng giữa thiết kế lý thuyết và thi công thực tế.

Giai đoạn 1: Khởi tạo và xác định yêu cầu (01/09 - 22/09) Trong ba tuần đầu tiên, nhóm tập trung vào việc định hình đề tài và xây dựng nền tảng lý thuyết. Sau khi đề xuất ý tưởng và đánh giá tính khả thi trong tuần đầu tháng 9, nhóm đã tiến hành biên soạn đồng bộ tài liệu đặc tả kỹ thuật bao gồm: Đặc tả phần mềm, Đặc tả phần cứng, Đặc tả kiểm thử và Đặc tả sản phẩm. Việc hoàn tất các tài liệu này vào ngày 22/09 đóng vai trò quan trọng trong việc thống nhất các thông số kỹ thuật trước khi bước vào thiết kế chi tiết.

Giai đoạn 2: Thiết kế và lập trình cơ sở (23/09 - 20/10) Giai đoạn này được triển khai song song giữa phần cứng và phần mềm. Về mặt thiết kế, nhóm đã hoàn thành sơ đồ nguyên lý, thiết kế mạch in (PCB) và phác thảo mô hình vỏ thiết bị vào giữa tháng 10. Đồng thời, các module phần mềm cơ sở điều khiển ngoại vi như Keypad, LCD, RTC, Loadcell và Servo cũng được lập trình và kiểm tra. Từ ngày 07/10, logic điều khiển chính của hệ thống bắt đầu được xây dựng dựa trên các module đã hoàn thiện.

Giai đoạn 3: Thi công và lắp ráp (07/10 - 10/11) Sau khi chốt thiết kế, công tác mua sắm linh kiện được thực hiện từ ngày 07/10. Quá trình gia công các chi tiết cơ khí diễn ra trong hai tuần cuối tháng 10, tạo tiền đề cho việc lắp ráp hoàn chỉnh hệ thống điện - cơ vào đầu tháng 11. Đây là giai đoạn chuyển hóa các bản vẽ thiết kế thành sản phẩm vật lý, đòi hỏi sự chính xác cao trong việc kết nối các thành phần.

Giai đoạn 4: Tích hợp và Kiểm thử (11/11 - 02/12) Giai đoạn nước rút bắt đầu từ giữa tháng 11, tập trung vào việc tích hợp toàn bộ phần cứng và nạp chương trình điều khiển hoàn chỉnh. Hệ thống được chạy thử nghiệm thực tế để đánh giá độ ổn định và phát hiện lỗi. Các vấn đề phát sinh trong quá trình kiểm thử (Testing issues) đã được nhóm tập trung xử lý và hoàn tất vào ngày 02/12/2025, đảm bảo sản phẩm hoạt động trơn tru trước khi báo cáo kết quả cuối cùng.

Project start day: 9/1/2025 Project end day: 12/11/2026							2025																		
							Sep		Oct		Nov		Dec												
Project Name: PET FEEDER							1-Sep	8-Sep	15-Sep	22-Sep	29-Sep	6-Oct	13-Oct	20-Oct	27-Oct	3-Nov	10-Nov	17-Nov	24-Nov	1-Dec	8-Dec	15-Dec	22-Dec	29-Dec	
No.	Task	Assigned to	Start	End	Plan	%Done	Status																		
1	Propose 2 project topics	H.L.N.Phát	1-Sep-25	5-Sep-25	5	100%	Completed																		
2	Draft Software Requirements Specification		15-Sep-25	22-Sep-25	8	100%	Completed																		
3	Implement keypad driver		23-Sep-25	6-Oct-25	14	100%	Completed																		
4	Sketch device housing		7-Oct-25	13-Oct-25	7	100%	Completed																		
5	Propose 2 project topics	T.P.Bình	1-Sep-25	5-Sep-25	5	100%	Completed																		
6	Draft Test Specification		15-Sep-25	22-Sep-25	8	100%	Completed																		
7	Implement speaker and servo control		23-Sep-25	6-Oct-25	14	100%	Completed																		
8	Propose 2 project topics		1-Sep-25	5-Sep-25	5	100%	Completed																		
9	Draft Hardware Specification	N.H.A.Kiệt	15-Sep-25	22-Sep-25	8	100%	Completed																		
10	Implement LCD and RTC drivers		23-Sep-25	6-Oct-25	14	100%	Completed																		
11	Fabricate mechanical parts		14-Oct-25	27-Oct-25	14	100%	Completed																		
12	Evaluate feasibility of topics		6-Sep-25	12-Sep-25	7	100%	Completed																		
13	Draft Engineering Specification	N.H.Khánh	15-Sep-25	22-Sep-25	8	100%	Completed																		
14	Implement Loadcell driver		23-Sep-25	6-Oct-25	14	100%	Completed																		
15	Source components		7-Oct-25	20-Oct-25	14	100%	Completed																		
16	Fabricate electro-mechanical system		28-Oct-25	10-Nov-25	14	100%	Completed																		
17	System integration and testing	L.M.Thiên	11-Nov-25	2-Dec-25	22	100%	Completed																		
18	Finalize project topic		13-Sep-25	15-Sep-25	3	100%	Completed																		
19	Draft Product Specification		15-Sep-25	22-Sep-25	8	100%	Completed																		
20	Design schematic and PCB layout		23-Sep-25	6-Oct-25	14	100%	Completed																		
21	Implement main system logic	L.M.Thiên	7-Oct-25	20-Oct-25	14	100%	Completed																		
22	Verify design via simulation		21-Oct-25	27-Oct-25	7	100%	Completed																		
23	Resolve testing issues		12-Nov-25	2-Dec-25	21	100%	Completed																		

Hình 1: Biểu đồ Gant kế hoạch đề tài

II. ĐẶC TẢ HỆ THỐNG

2.1 Đặc tả sản phẩm

Hệ thống cho thú cưng ăn tự động được thiết kế nhằm cung cấp thức ăn đúng thời gian và đúng khẩu phần đã cài đặt. Các chức năng chính bao gồm:

Cho ăn tự động theo lịch được cài đặt trước.

Cho ăn thủ công thông qua nút nhấn.

Theo dõi và đo lường khối lượng thức ăn bằng cảm biến load cell.

Cảnh báo khi thức ăn trong ngăn chứa sắp hết hoặc xảy ra lỗi hệ thống.

Lưu trữ lịch cho ăn và khẩu phần vào bộ nhớ ROM.

Ngõ vào (Input):

Cảm biến load cell kết hợp module ADC HX711 để đo khối lượng thức ăn.

Keypad để cài đặt giờ cho ăn và điều khiển các chế độ cho ăn.

Nút nhấn reset khi hệ thống có vấn đề.

Ngõ ra (Output):

Servo motor điều khiển cửa xả thức ăn.

Màn hình LCD 16×2 hiển thị thông tin.

Buzzer phát cảnh báo.

LED báo trạng thái hoạt động của hệ thống.

Giao diện người dùng (User Interface):

Màn hình LCD 16×2 hiển thị thời gian thực, lịch cho ăn, chế độ hoạt động và lượng thức ăn.

Keypad cho phép người dùng cài đặt thời gian cho ăn.

Nút nhấn điều khiển các chế độ và cài đặt giờ cho ăn.

Giao tiếp ngoài (External Interface):

Giao tiếp I²C giữa vi điều khiển ATmega16 và module RTC DS1307 để lấy thời gian thực.

Giao tiếp với module HX711 để đọc dữ liệu từ load cell.

Ràng buộc hệ thống (Constraints):

Hệ thống phải hoạt động ổn định, tránh kẹt cơ khí khi cấp thức ăn.

Độ chính xác đo khối lượng thức ăn sai số không quá $\pm 5g$.

Servo motor quay chậm để đảm bảo an toàn cho thú cưng.

Tiêu thụ điện năng thấp, sử dụng nguồn adapter 5V DC.

Giá thành thấp, phù hợp cho mục đích dân dụng.

2.2 Đặc tả kỹ thuật

Phần cứng sử dụng

Vi điều khiển: ATmega16.

Cảm biến: Load cell + module HX711.

Hiển thị: LCD 16×2.

Chấp hành: Servo motor điều khiển cửa xả thức ăn.

Giao diện điều khiển: Keypad 4×4 hoặc nút nhấn.

Thời gian thực: RTC DS1307 (I²C).

Cảnh báo: Buzzer và LED.

Nguồn: Adapter 5V DC.

Phần mềm điều khiển hệ thống phải đáp ứng các yêu cầu sau:

Lập trình vi điều khiển để động cơ cấp thức ăn theo thời gian được thiết lập.

Thu thập và xử lý dữ liệu từ cảm biến đảm bảo lượng thức ăn còn lại chính xác.

Đảm bảo khả năng xử lý thời gian thực (real-time processing).

Cho phép người dùng cài đặt và thay đổi thời gian cho ăn.

Hiển thị đầy đủ các thông tin cần thiết.

Yêu cầu chức năng hệ thống Pet Feeder System phải đáp ứng các chức năng sau:

Tự động cấp thức ăn cho thú cưng đúng thời gian theo thiết lập.

Hoạt động ổn định, chính xác và an toàn trong quá trình vận hành.

Có khả năng cập nhật thông số và cấu hình hệ thống trong khi đang hoạt động.

Có khả năng truyền dữ liệu hoặc trạng thái hoạt động của hệ thống tới thiết bị khác thông qua SCK và I²C.

2.3 Hardware Specification

Dựa trên đặc tả kỹ thuật đã đề ra, hệ thống Pet Feeder System cần đảm bảo khả năng tự động cấp thức ăn cho thú cưng theo thời gian và khẩu phần đã cài đặt, hoạt động ổn định, chính xác và an toàn. Phần cứng của hệ thống phải đáp ứng các yêu cầu về xử lý thời gian thực, điều khiển cơ cấu chấp hành và giao tiếp với người dùng.

Hệ thống sử dụng các linh kiện phần cứng sau:

Vi điều khiển: ATmega16A với xung nhịp 8Mhz, có thể dùng thạch anh ngoại hoặc nội

Động cơ: Servo motor dùng để điều khiển đóng/mở cửa xả thức ăn.

Cảm biến: load cell kết hợp module HX711 để đo khối lượng thức ăn.

Module thời gian thực (RTC): DS1307, cung cấp thời gian chính xác cho hệ thống.

Màn hình hiển thị: LCD 16x2 hiển thị thời gian và khối lượng.

Keypad: Dùng để cài đặt thời gian cho ăn và chế độ ăn.

Nguồn cấp: Adapter DC 5V đảm bảo cấp nguồn ổn định cho toàn hệ thống.

Vi điều khiển đóng vai trò trung tâm điều khiển của hệ thống. Tín hiệu thời gian được lấy từ module RTC để xác định thời điểm cho ăn. Khi đến thời gian đã cài đặt, vi điều khiển phát tín hiệu điều khiển động cơ để mở xả thức ăn.

Trong quá trình cấp thức ăn, cảm biến khối lượng (Load Cell) gửi tín hiệu analog về module HX711, sau đó được chuyển đổi sang dữ liệu số và truyền tới vi điều khiển.

Dựa trên giá trị đo được, vi điều khiển điều chỉnh hoạt động của động cơ nhằm đảm bảo lượng thức ăn còn lại bao nhiêu đúng với khẩu phần đã xả ra.

Thông tin về thời gian, trạng thái hệ thống và lượng thức ăn được hiển thị trực tiếp trên màn hình LCD. Người dùng có thể thay đổi các thông số thông qua hệ thống nút nhấn.

Vi điều khiển giao tiếp với:

Module RTC qua giao thức I2C.

Module HX711 qua giao tiếp số.

LCD thông qua giao diện song song 4bit.

Động cơ được điều khiển bằng tín hiệu PWM từ vi điều khiển.

Nút nhấn được kết nối vào các chân GPIO được xử lý bằng phần mềm.

2.4 Software Specification

Phần mềm của hệ thống Pet Feeder System phải đáp ứng các yêu cầu được nêu trong đặc tả kỹ thuật, bao gồm khả năng tự động cấp thức ăn theo thời gian và chế độ đã thiết lập, xử lý dữ liệu cảm biến chính xác, đảm bảo tính thời gian thực và cung cấp giao diện tương tác với người dùng.

Phần mềm được lập trình và nạp trực tiếp vào vi điều khiển, giao tiếp với các module ngoại vi thông qua các giao thức:

I2C: giao tiếp với RTC và LCD (nếu sử dụng I2C)

GPIO/PWM: điều khiển động cơ và đọc nút nhấn

UART/I2C: truyền dữ liệu trạng thái hệ thống tới các module.

Phần mềm được tổ chức theo mô hình vòng lặp chính (main loop) kết hợp với các hàm xử lý chức năng riêng biệt. Vi điều khiển liên tục đọc thời gian từ module RTC và so sánh với thời gian cho ăn đã cài đặt. Khi đến thời điểm cấp thức ăn, chương trình kích hoạt động cơ và đồng thời còi báo hoạt động để báo thức ăn đang được xả ra và giám sát dữ liệu từ cảm biến khối lượng để cập nhật lên lcd.

Thông tin về trạng thái hoạt động, thời gian hiện tại và khẩu phần ăn được cập nhật liên tục và hiển thị trên màn hình LCD. Người dùng có thể thay đổi các thông số hệ thống thông qua các nút nhấn, phần mềm sẽ ghi nhận và lưu trữ các giá trị mới.

2.5 Test Specification

Để kiểm tra và đánh giá hoạt động của hệ thống Pet Feeder System, các thiết bị và điều kiện sau được sử dụng:

Thiết bị đo: đồng hồ đo điện áp (Voltage meter), cân điện tử (đổi chiều khối lượng thức ăn).

Thiết bị lập trình & nạp chương trình: máy tính cá nhân, cáp USB nạp chương trình cho vi điều khiển.

Môi trường kiểm thử: tại nhà, đảm bảo an toàn điện.

Mô hình thử nghiệm (Prototype):

Hệ thống được lắp ráp dưới dạng prototype trên breadboard hoặc mạch thử nghiệm.

Các linh kiện chính (vi điều khiển, động cơ, cảm biến, RTC, LCD, nút nhấn) được kết nối đầy đủ theo sơ đồ nguyên lý.

Mô hình mô phỏng cơ cấu cấp thức ăn thực tế để kiểm tra hoạt động của hệ thống.

Quy trình kiểm thử hệ thống được thực hiện theo các bước sau:

Kiểm tra nguồn và phần cứng

Đảm bảo các linh kiện hoạt động đúng thông số kỹ thuật.

Hiệu chỉnh (calibrate) cảm biến khối lượng (Load Cell).

So sánh giá trị đo được với cân điện tử để đánh giá độ chính xác.

Kiểm tra hoạt động của màn hình LCD (hiển thị thời gian, trạng thái).

Kiểm tra các nút nhấn cài đặt thời gian và khẩu phần ăn.

Kiểm tra tín hiệu điều khiển động cơ.

Đánh giá khả năng đóng/mở và dừng động cơ đúng thời điểm.

Thiết lập thời gian cho ăn và quan sát hệ thống hoạt động tự động.

Kiểm tra tính ổn định

Ghi nhận các lỗi hoặc sai lệch (nếu có).

III. CÁC VẤN ĐỀ THIẾT KẾ.

3.1 Constraint Issues (Ràng buộc thiết kế)

Trong quá trình thiết kế hệ thống nhúng cho thiết bị "Pet Feeder", nhóm thực hiện đã phân tích và giải quyết các bài toán ràng buộc (Constraint issues) dựa trên các nguyên lý thiết kế cơ bản.

3.1.1 Ràng buộc về chi phí

Theo lý thuyết thiết kế hệ thống nhúng, chi phí sản phẩm thường là yếu tố ưu tiên hàng đầu, đôi khi quan trọng hơn tốc độ xử lý (cost may matter more than speed).

Thách thức: Hệ thống hướng đến mục đích dân dụng, yêu cầu giá thành thấp dưới 1 triệu để phù hợp với thị trường.

Giải pháp thiết kế:

Lựa chọn vi điều khiển ATmega16: Đây là dòng vi điều khiển 8-bit có chi phí thấp, đủ tài nguyên (Flash, RAM, I/O) để đáp ứng các tác vụ điều khiển cơ bản mà không gây lãng phí như các dòng 32-bit hiệu năng cao.

Sử dụng LCD 16x2 và Keypad 4x4: Thay vì sử dụng màn hình cảm ứng đắt tiền, việc sử dụng giao diện nút nhấn vật lý và màn hình text giúp giảm chi phí phần cứng nhưng vẫn đảm bảo đầy đủ chức năng giao tiếp người dùng (UI).

3.1.2 Ràng buộc về độ an toàn

Hệ thống yêu cầu hoạt động ổn định trong thời gian dài mà không cần sự can thiệp của con người, đồng thời phải đảm bảo an toàn tuyệt đối cho vật nuôi.

An toàn cơ khí (Safety): việc sử dụng động cơ Servo để đóng/mở cửa xả tiềm ẩn nguy cơ gây thương tích nếu thú cưng tiếp xúc khi máy đang hoạt động.

Giải pháp: Thiết kế phần mềm điều khiển xung PWM để Servo quay chậm, tránh kẹt cơ khí và đảm bảo an toàn theo đúng đặc tả hệ thống.

3.1.3 Ràng buộc độ tin cậy và chính xác

Hệ thống phải đảm bảo hoạt động đúng chức năng trong thời gian dài mà không có sự giám sát của con người. Thách thức nằm ở hai yếu tố: đo lường khối lượng và duy trì thời gian thực.

Thách thức đầu tiên: Cảm biến Loadcell và module HX711 rất nhạy cảm với nhiễu nguồn và rung động. Việc động cơ servo hoạt động có thể gây rung lắc làm sai lệch kết quả đo, không sai số quá 10g.

Giải pháp: Thực hiện thuật toán lấy mẫu trung bình (averaging) trong phần mềm và thiết kế độ trễ (delay) hợp lý để đo khối lượng khi hệ thống ổn định.

Thách thức thứ hai: Hệ thống sử dụng module RTC DS1307 để quản lý lịch trình. Đây là dòng chip giá rẻ, không tích hợp bộ dao động bù nhiệt (TCXO). Hoạt động của nó phụ thuộc vào thạch anh ngoại 32.768kHz, vốn rất nhạy cảm với thay đổi nhiệt độ môi trường. Phân tích sai số: Với thạch anh tiêu chuẩn, sai số tích lũy có thể đạt 1 đến 2 phút mỗi tháng. Trong điều kiện nhiệt độ thay đổi khắc nghiệt, sai số này có thể lớn hơn.

Giải pháp: Đánh đổi việc sai lệch vài phút không ảnh hưởng nghiêm trọng đến sinh hoạt của thú cưng, nên việc sử dụng DS1307 là chấp nhận được để giảm chi phí so với các dòng chip cao cấp (như DS3231). Thêm vào đó ta có thể thiết kế tính năng cho phép người dùng cài đặt lại giờ dễ dàng qua Keypad sau một thời gian sử dụng.

3.1.4 Ràng buộc về bộ nguồn

Mặc dù hệ thống sử dụng nguồn Adapter 5V DC (không chạy pin), vấn đề ổn định nguồn là một thách thức lớn trong thiết kế mạch (Design issue).

Thách thức: Động cơ Servo là tải cảm, khi khởi động hoặc đảo chiều sẽ tạo ra dòng khởi động lớn (inrush current) và nhiễu xung ngược (back EMF). Điều này có thể gây sụt áp đường 5V, dẫn đến việc vi điều khiển ATmega16 bị Reset đột ngột hoặc treo hệ thống.

Giải pháp: Thiết kế mạch nguồn cần tách biệt hoặc lọc nhiễu tốt giữa khối động lực (Servo) và khối điều khiển (MCU, Cảm biến), đảm bảo hệ thống hoạt động liên tục (Long life cycle).

3.1.5 Ràng buộc về kích thước

Kích thước: Hệ thống cần tích hợp gọn gàng giữa các thành phần: thùng chứa thức ăn, máng ăn, bo mạch và màn hình điều khiển.

Môi trường: Do hoạt động trong môi trường có thức ăn khô, hệ thống phải đối mặt với bụi bẩn hoặc mảnh vụn thức ăn có thể gây kẹt cơ cấu chấp hành. Đây là một vấn đề thiết kế phần cứng cần được tính toán kỹ lưỡng để đảm bảo tuổi thọ sản phẩm.

3.2 Functional Issues (Vấn đề chức năng)

Khác với các yêu cầu chức năng thông thường (như hiển thị LCD hay bấm nút), các vấn đề về chức năng (Functional Issues) trong hệ thống Pet Feeder tập trung phân tích các rủi ro có thể ảnh hưởng đến sức khỏe và sự an toàn của vật nuôi nếu hệ thống vận hành sai lệch..

3.2.1 Ảnh hưởng đến sức khỏe vật nuôi

Đây là vấn đề quan trọng nhất vì hệ thống thay thế con người trong việc chăm sóc sinh vật sống.

Vấn đề: Nếu hệ thống gặp lỗi (treo vi điều khiển, mất nguồn, hoặc kẹt thức ăn) và không cấp thức ăn trong thời gian dài, vật nuôi sẽ bị đói, ảnh hưởng trực tiếp đến sức khỏe. Ngược lại, nếu hệ thống xả quá nhiều thức ăn do lỗi cảm biến Loadcell, vật nuôi có thể gặp vấn đề về tiêu hóa hoặc béo phì.

Giải pháp: Hệ thống cần có cơ chế Watchdog Timer để tự khởi động lại khi bị treo và cơ chế cảnh báo khi thức ăn không xả ra được để người dùng can thiệp kịp thời.

3.2.2. An toàn cơ học

Vấn đề: Cơ cấu chấp hành sử dụng động cơ Servo để đóng/mở cửa xả. Nếu không được kiểm soát tốt, cửa xả có thể kẹp vào chân hoặc mõm của thú cưng khi chúng đang cố lấy thức ăn, gây thương tích vật lý.

Giải pháp: Áp dụng thuật toán điều khiển Servo quay chậm và giới hạn lực (hoặc thiết kế cơ khí có khe hở an toàn) để đảm bảo không gây tổn thương cho vật nuôi ngay cả khi có sự cố.

3.2.3. Vệ sinh và Môi trường

Vấn đề: Thức ăn lưu trữ trong thùng chứa nếu không kín có thể bị ẩm mốc hoặc kiến gián xâm nhập. Nếu hệ thống cấp thức ăn hỏng này, sẽ gây ngộ độc cho thú cưng.

Giải pháp: Thiết kế cơ khí phải đảm bảo độ kín cho thùng chứa và vật liệu tiếp xúc với thức ăn phải là nhựa an toàn thực phẩm.

3.3 Realtime issue

Dựa trên phân tích đặc thù hoạt động của hệ thống Pet Feeder và lý thuyết về thiết kế hệ thống nhúng, nhóm thực hiện xác định các vấn đề về thời gian thực như sau:

Hệ thống Pet Feeder được phân loại là Hệ thống thời gian thực mềm (Soft Real-time System).

Lý do: việc cấp thức ăn cần tuân thủ lịch trình, nhưng nếu hệ thống xả thức ăn chậm trễ một vài giây (hoặc thậm chí 1 phút) so với giờ cài đặt thì vẫn không gây ra lỗi hệ thống nghiêm trọng hay hậu quả thảm khốc (như hệ thống túi khí ô tô - Hard real-time). Tuy nhiên, hệ thống không thể là "Non real-time" vì nó vẫn yêu cầu phản hồi kịp thời với thao tác của người dùng và tín hiệu cảm biến để đảm bảo trải nghiệm sử dụng.

Thay vì xét trên từng linh kiện riêng lẻ, hệ thống Pet Feeder chịu một ràng buộc thời gian chung nhất định để đảm bảo tính ổn định và chính xác.

Ràng buộc: Thời gian phản hồi toàn hệ thống phải $< 200\text{ms}$. Định nghĩa: Đây là khoảng thời gian tối đa cho phép tính từ lúc xuất hiện bất kỳ sự kiện đầu vào nào (Input Event: tín hiệu cảm biến, thao tác người dùng, hoặc ngắt thời gian) cho đến khi hệ thống hoàn tất xử lý và đưa ra tác động đầu ra tương ứng.

Lý do: Ngưỡng 200ms là giới hạn tiêu chuẩn cho các hệ thống tương tác người máy và điều khiển cơ bản (Soft Real-time). Việc duy trì chu kỳ xử lý dưới ngưỡng này đảm bảo hệ thống không bao giờ rơi vào trạng thái "treo" hoặc mất kiểm soát quá trình cấp thức ăn, đồng thời đảm bảo sai số vận hành nằm trong phạm vi cho phép của thiết kế mà không cần các phần cứng xử lý tốc độ cao đắt tiền.

Giải pháp: Thiết kế kiến trúc phần mềm theo cơ chế không chặn (Non-blocking). Chương trình điều khiển tuyệt đối không sử dụng các lệnh dừng chờ (như delay thời

gian dài) trong vòng lặp chính, đảm bảo vi điều khiển luôn ở trạng thái sẵn sàng tiếp nhận và xử lý sự kiện mới trong chu kỳ cho phép.

3.4 Concurrent Issues (Vấn đề xử lý đồng thời)

Trong quá trình vận hành, hệ thống Pet Feeder không hoạt động tuần tự đơn thuần mà phải xử lý nhiều tác vụ song song để đáp ứng kịp thời các thay đổi từ môi trường và người dùng.

Hệ thống phải thực hiện đồng thời các chức năng sau trong cùng một khoảng thời gian:

Đọc cảm biến: Liên tục lấy mẫu từ Loadcell (qua HX711) để giám sát lượng thức ăn và đọc thời gian thực từ RTC (qua I2C).

Xử lý giao diện: Quét nút nhấn để nhận lệnh từ người dùng và cập nhật thông tin lên màn hình LCD.

Điều khiển chấp hành: Xuất xung PWM để điều khiển Servo đóng/mở cửa xả.

Vấn đề: Vi điều khiển ATmega16 là loại đơn nhân, tại một thời điểm chỉ thực hiện được một lệnh. Nếu chương trình sa đà vào việc đợi Servo quay, hệ thống sẽ bị "mù" trước các sự kiện khác như nút nhấn dừng khẩn cấp hoặc tín hiệu thay đổi trọng lượng đột ngột từ Loadcell.

3.5 Reactive Issues (Vấn đề phản ứng)

Hệ thống Pet Feeder được thiết kế là một hệ thống phản ứng (Reactive System), duy trì tương tác liên tục với môi trường vận hành thay vì chỉ hoạt động theo yêu cầu tức thời.

Về phương thức tương tác, hệ thống thuộc nhóm "Always ON" (Luôn bật), yêu cầu vi điều khiển phải hoạt động 24/7 để duy trì bộ đếm thời gian và giám sát trạng thái cảm biến. Đặc thù này dẫn đến vấn đề thiết kế quan trọng là việc ngưng hoạt động đột ngột được coi là hành vi lỗi nghiêm trọng, có thể gây ảnh hưởng trực tiếp đến việc chăm sóc vật nuôi. Do đó, hệ thống bắt buộc phải tích hợp cơ chế giám sát Watchdog Timer để tự động khôi phục trạng thái nếu xảy ra sự cố treo phần mềm.

Bên cạnh đó, hệ thống phải xử lý đồng thời hai loại sự kiện đầu vào. Đối với các sự kiện có tính chu kỳ (Periodic events) như cập nhật thời gian thực từ module RTC và lấy mẫu tín hiệu Loadcell, hệ thống cần một cơ chế lập lịch để đảm bảo các tác vụ này được thực thi đúng thời điểm mà không bị trôi thời gian. Ngược lại, đối với các sự kiện phi chu kỳ (Non-periodic events) xuất hiện ngẫu nhiên như thao tác nhấn nút cài đặt của người dùng, vấn đề thiết kế nằm ở việc ước lượng và loại bỏ các trường hợp bỏ lỡ sự kiện (miss event cases). Điều này đòi hỏi thuật toán điều khiển phải có tần suất phản

hồi đủ nhanh hoặc sử dụng cơ chế ngắt (Interrupt) để đảm bảo mọi tương tác của người dùng đều được hệ thống ghi nhận tức thì.

IV. Thiết kế phần cứng

4.1 Chọn linh kiện

Từ đặc tả phần cứng phần 2.3 ta lựa chọn các linh kiện như sau:

STT	Tên linh kiện	Số lượng	Đơn giá	Thành tiền
1	Atmega 16	1	85000	85000
2	Loadcell	1	34000	34000
3	Module HX711	1	12000	12000
4	RTC DS1307	1	52000	52000
5	Keypad 4x4	1	9000	9000
6	Màn hình LCD 16x2	1	32000	32000
7	LED	10	250	2500
8	Buzzer	1	3000	3000
9	Servo motor	1	27000	27000
10	Thạch anh 8Mhz	1	1500	1500
11	Thạch anh 32 768Hz	1	1500	1500
12	Biến trở	1	6500	6500
13	Trở 10K	50	60	3000
14	Trở 1K	50	60	3000
15	Nút nhấn	10	400	4000
16	Adapter 5V	1	45000	45000
18	Đế IC	1	10000	10000
19	Tụ 22pF	10	100	1000
20	Đế loadcell	1	100000	100000
21	BJT BC 547	1	500	500
22	Hàng rào đục	5	700	700
23	Hàng rào cái	5	700	700
24	Mica	4	38750	155000
25	Băng keo	1	20000	20000
TỔNG		608.900		

Nhận xét: nhóm mua những IC chính hãng nên giá sẽ cao hơn so với mặt bằng chung nếu ta mua giá sẽ rẻ hơn. Ngoài ra để thiết kế userface thì nhóm đã chuẩn bị băng keo và mica để làm thành mô hình cho đề tài này vậy giá sẽ có phần cao hơn.

4.2 Thông số các linh kiện

4.2.1 MCU ATmega16

Có giao tiếp I2C, số chân I/O PORT cần thiết: 7 chân LCD (trong đó 3 chân điều khiển và 4 chân dữ liệu) + 2 chân I2C + 8 chân Keypad + 2 chân ngõ ra điều khiển hoạt động. Tổng cộng cần thiết 19 chân I/O. Tốc độ xử lý đáp ứng nhanh với sự thay đổi của LCD lệnh dài nhất là 200 (us) nhưng vẫn đáp ứng được tốc độ đếm thời gian chính xác bằng giây, cần vi xử lý có tốc độ trên 1MHz, có chế độ tạo độ rộng xung của Timer để điều khiển Servo.

Nếu được có thể quen thuộc với nhóm thiết kế, qua đó nhóm cân nhắc dòng AVR cụ thể là Atmega16 và Atmega32. Nhưng do chi phí Atmega32 có phần đắt hơn Atmega16 và dòng Atmega16 đã đủ đáp ứng những yêu cầu thiết kế nên nhóm quyết định chọn vi điều khiển Atmega16.

Thông số linh kiện:

MCU: Atmega16 (8-bit AVR)

Tần số: 16 (MHz)

Bộ nhớ Flash: 16 (KB)

SRAM: 1 (KB)

Số chân: 44

Số chân I/O: 32

Thạch anh: 1 đến 16 (MHz)

Cấu trúc: RISC

ADC 10 bit: 8 (kênh)

Giao diện JTAG: gỡ lỗi, hỗ trợ thông lượng 16 MIPS ở 16 MHz
giữa 4.5 và 5.5 (V)

Điện áp ngõ ra mức cao: từ $V_{cc} - 0.6$ đến V_{CC}

Điện áp ngõ ra mức thấp: 0.5 đến 0.6 (V)

Dòng ngõ ra tối đa: 40 (mA)



Hình 2: ATmega16A

4.2.2 Keypad 4x4

	Bàn phím mềm 4x4 (có in ký tự)	Bàn phím mềm 4x4 (nút nhấn cơ bản)
Độ linh hoạt	Kém	Cao
Giao tiếp	I/O	I/O
Giá tiền	15 000	9 000

Để lựa chọn keypad cho phần nút nhấn cài đặt hệ thống, nhóm loại bỏ loại keypad có in sẵn ký tự dù nhìn bắt mắt và thân thiện hơn với người dùng nhưng lại kém linh hoạt khi ký tự được in chưa thật sự khớp với ý đồ thiết kế. Do đó việc lựa chọn bàn phím chưa được đánh dấu chức năng là lựa chọn để nhóm tự do cài đặt chức năng.

Thông số linh kiện:

Nhiệt độ hoạt động: 0 đến 70 (độ C)

Đầu ra: 8 chân (I/O)

Kích thước: 77 x 69 (mm)



Hình 3: Keypad 4x4

4.2.3 RTC DS1307

	RTC DS3231	RTC DS1307
Giao tiếp	I2C	I2C
Chân	Dán	Cắm
Giá tiền	57000	52000

Do số lượng giao tiếp và ngoại vi của hệ thống khác nhiều nên việc tiết kiệm được số chân giao tiếp là cần thiết nên nhóm lựa chọn giao tiếp I2C của RTC DS1307 để đơn giản hóa sơ đồ mạch và có chân cắm để đơn giản hóa việc gia công mạch in. Giá tiền chênh lệch không nhiều nên nhóm có thể cân nhắc phát triển sâu hơn với nguồn pin cung cấp vào chân Vbat của DS1307 trong trường hợp ngắt nguồn điện.

Thông số linh kiện:

Giao tiếp: I2C

Điện áp hoạt động: 5 (V)

Dòng điện hoạt động: Dưới 500 (nA)

Bộ nhớ SVRAM: 56 (byte)

Hoạt động có thể cấp nguồn bằng VCC hoặc pin.

Chân số 1 – 2 (X1 và X2) gắn với bộ dao động thạch anh. Chân số 3 (Vbat) nguồn pin, chân số 4 – 5 (SDA và SCL) là 2 chân giao tiếp I2C. Chân số (SQW) cho phép xuất xung vuông và chân số 8 (VCC) nối với nguồn điện cấp vào.



Hình 4: RTC DS1307

4.2.4 LCD 16x2

Với phân khúc LCD phân bố đa dạng nhưng về chi phí thấp vẫn đảm bảo hiển thị thông tin thời gian thì nhóm lựa chọn LCD 1602. LCD ký tự 16x2 chữ đen nền vàng xanh có IC điều khiển HD44780 Super Twisted Nematic (STN) LCD. Giao tiếp đơn giản với 4 bit hoặc 8 bit. Giao tiếp dễ dàng với các loại vi điều khiển và board Arduino, ATMELE AVR hoặc PIC. Thật sự rất khó có thể đưa ra lựa chọn LCD khác vì các phân khúc còn lại của LCD giá thành rất cao nhưng dư thừa về chức năng hệ thống cần thiết.

Thông số linh kiện:

Loại: LCD 1602

Điện áp chịu đựng: -0.3 đến 7 (V)

Hoạt động ổn định: 2.7 đến 5 (V)

Điện áp mức cao: > 2.4 (V)

Điện áp mức thấp: < 0.4 (V)

Dòng cấp nguồn: 350 đến 600 (uA)

Nhiệt độ hoạt động: 30 đến 75 (độ C)

Chân số 1 (VSS) nối đất (GND), chân số 2 (VDD) nối lên nguồn, chân số 3 (VEE) dùng biến trở để điều chỉnh sự tương phản của LCD. Chân số 4 (RS) cho phép truy xuất dữ liệu (data RS = 1) hoặc lệnh (command khi RS = 0). Chân 7 đến 14 (D0 – D7) là ngõ vào dữ liệu để giao tiếp với MPU (giao tiếp I/O). Chân 15 – 16 là nguồn dương và nguồn âm cho đèn nền.



Hình 5: LCD 1602

4.2.5 Servo sg90

	Terasic FXX-3037	SER0047	MG996R	SG90
Kích thước (chiều dài mm)	23	22.9	42.9	29.5
Mô – men xoắn (kg)	9 - 11	10	11	1.2 - 14
Giá tiền	317 762	175 405	90 000	27000

Theo sự đánh giá về kích thước hệ thống, trọng lượng servo, mô – men xoắn không cần quá lớn. Cộng thêm sự hài hòa về giá tiền chi trả và vị trí đặt hàng thì nhóm loại bỏ 2 loại servo đầu tiên vì khá bất tiện trong việc đặt hàng và chi phí cao. Khối lượng quay không đòi hỏi cao nên nhóm quyết định sử dụng servo SG90.

Thông số linh kiện:

Loại: SG90 – 360

Kích thước: 23 x 12.5 x 29.5 (mm)

Trọng lượng: 9 (g)

Tốc độ không tải: 0.12s/60 độ (Trong điều kiện 4.8V)

Mô – men xoắn chặn: 1.2 – 1.4 (kg) (Trong điều kiện 4.8V)

Nhiệt độ hoạt động tốt: -30 độ C đến +60 độ C

Cài đặt vùng chết: 7 (us)

Điện áp làm việc: 4.8 – 6 (V)

Sơ đồ dây: GND (màu nâu), VCC (màu đỏ từ 4.8 – 6V), đầu vào xung (màu cam).



Hình 6: Motor servo sg90

4.2.6 Loadcell

Trong bối cảnh đa dạng các giải pháp hiển thị hiện nay, nhóm thực hiện quyết định lựa chọn module LCD 1602 (Liquid Crystal Display 16x2) làm giao diện giao tiếp người dùng nhờ sự cân bằng tối ưu giữa chi phí và công năng sử dụng. Về mặt kỹ thuật, đây là loại màn hình sử dụng công nghệ STN (Super Twisted Nematic) hiển thị ký tự đen trên nền vàng xanh, được tích hợp sẵn IC điều khiển HD44780 chuẩn công nghiệp cho phép giao tiếp linh hoạt và ổn định với vi điều khiển ATmega16 cũng như các họ vi xử lý khác (AVR, PIC, Arduino) thông qua chế độ 4-bit hoặc 8-bit. Quan trọng hơn, việc lựa chọn LCD 1602 xuất phát từ phân tích hiệu quả kinh tế, khi thiết bị này có giá thành thấp nhất nhưng vẫn đáp ứng trọn vẹn nhu cầu hiển thị các thông tin văn bản cơ bản (thời gian, khối lượng) của hệ thống, trong khi các phân khúc màn hình đồ họa cao cấp khác được đánh giá là dư thừa chức năng và làm tăng chi phí sản phẩm một cách không cần thiết đối với một thiết bị dân dụng.

Thông số

Rated Output (mV/V): 1.0 ± 0.15

Độ lệch tuyến tính (%): 0.05

Creep (5min) %: 0.1

Ảnh hưởng nhiệt độ tới độ nhạy %RO/ độ C: 0.003

Ảnh hưởng nhiệt độ tới điểm không %RO/ độ C: 0.02

Độ cân bằng điểm không %RO: ± 0.1

Trở kháng đầu vào (Ω) : 1066 ± 20

Trở kháng ngõ ra (Ω) : 1000 ± 20

Điện áp hoạt động: 5V

Nhiệt độ hoạt động: $-20 \sim 65$ độ C

Safe Overload %RO:120

Ultimate overload %RO: 150

Chất liệu cảm biến: Nhôm

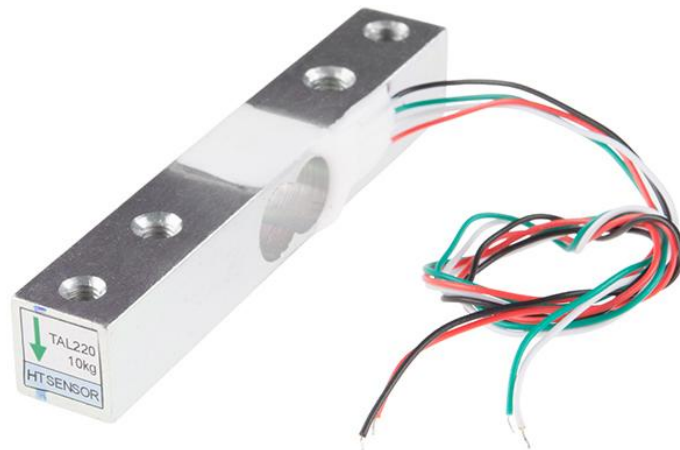
Độ dài dây: 180mm

Dây đỏ: Ngõ vào (+)

Dây đen: Ngõ vào (-)

Dây xanh lá: Ngõ ra (+)

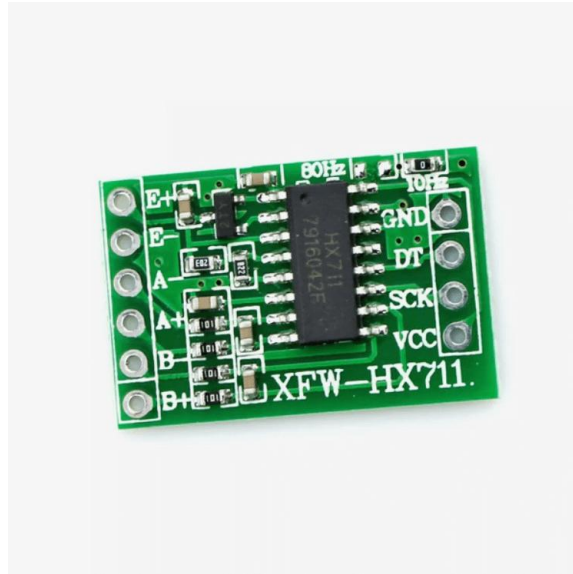
Dây trắng: Ngõ ra (-)



Hình 7: Loadcell 5kg

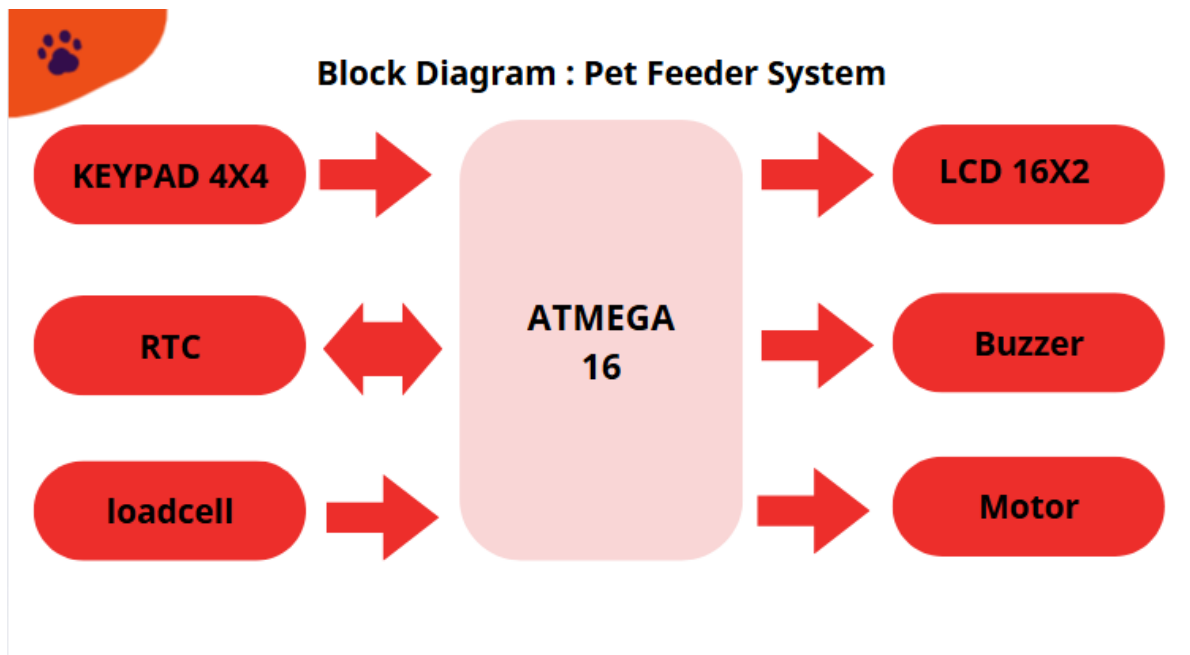
4.4.7 HX711

Để giải quyết bài toán đo lường khối lượng với độ chính xác cao, hệ thống sử dụng module chuyển đổi tương tự - số (ADC) chuyên dụng HX711. Đây là thành phần trung gian đóng vai trò quan trọng trong việc kết nối cảm biến Loadcell với vi điều khiển trung tâm ATmega16.



Hình 8: HX711

4.3 Thiết kế sơ đồ khối



Hình 9: Sơ đồ khối phần cứng hệ thống

4.4 Thiết kế sơ đồ nguyên lí

4.4.1 Khối vi xử lý

Vi điều khiển được cấp nguồn VCC và GND từ VCC và GND chung của hệ thống.

PORTA từ PA0 tới PA3 của vi điều khiển nối với ngõ vào Data D4 – D7 của LCD 1602.

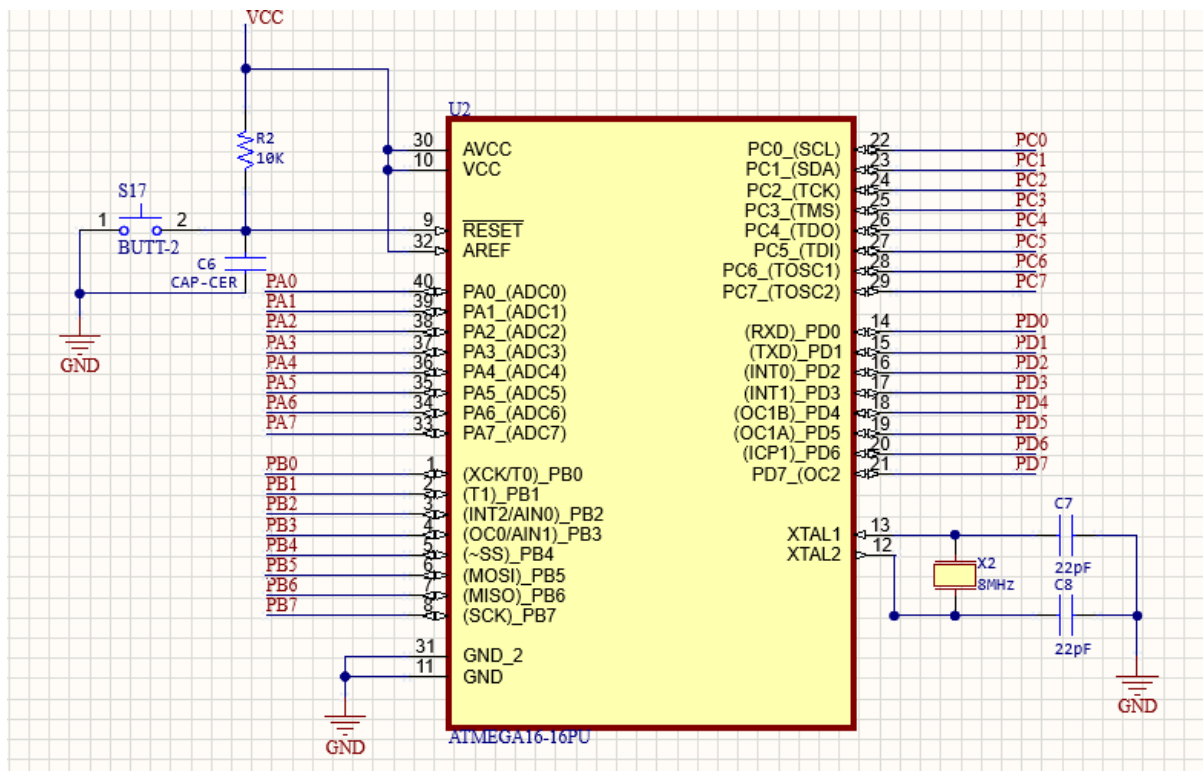
Chân PB3 – PB5 là ngõ điều khiển RS RW và E của LCD.

PC0 và PC1 lần lượt là chân SDA và SCL kết nối với RTC DS1307 trong giao thức I2C.

Chân PB0 nối với Servo SG90 để cung cấp xung điều khiển. Chân PB1 nối ra buzzer. Chân PB6 và 7 nối vào ngõ ra HX711

PORTD của Atmega16 nối với các chân của Keypad 4x4.

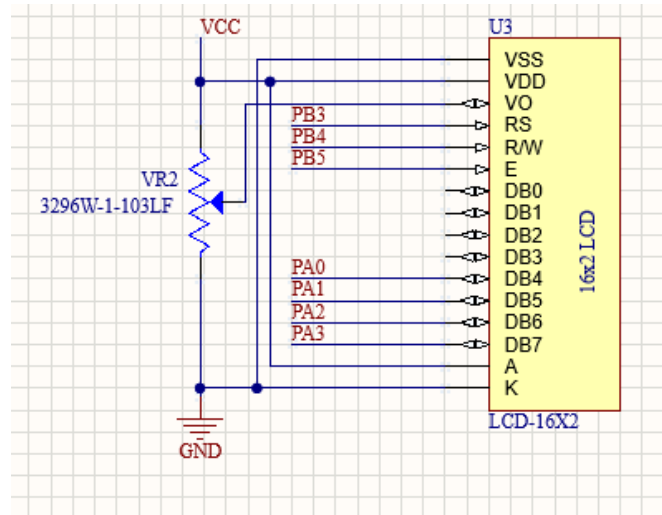
Chân Xtal1 và Xtal2 liên kết với ngõ ra thạch anh 8MHz.



Hình 10: Thiết kế khối MCU

4.4.2 Khối LCD 16x2

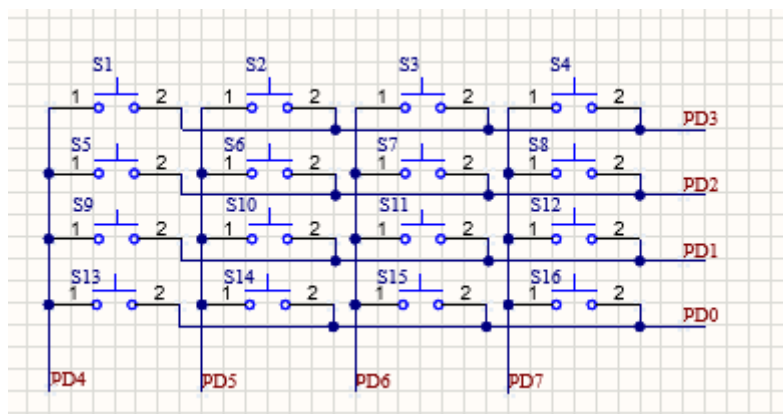
Chân VSS nối vào VCC và VDD nối vào GND, VO nối vào biến trở để hiển thị màn hình, các chân RS RW E và data nối vào MCU như mô tả phần trên



Hình 11: Thiết kế khối LCD 1602

4.4.3 Khối keypad 4x4

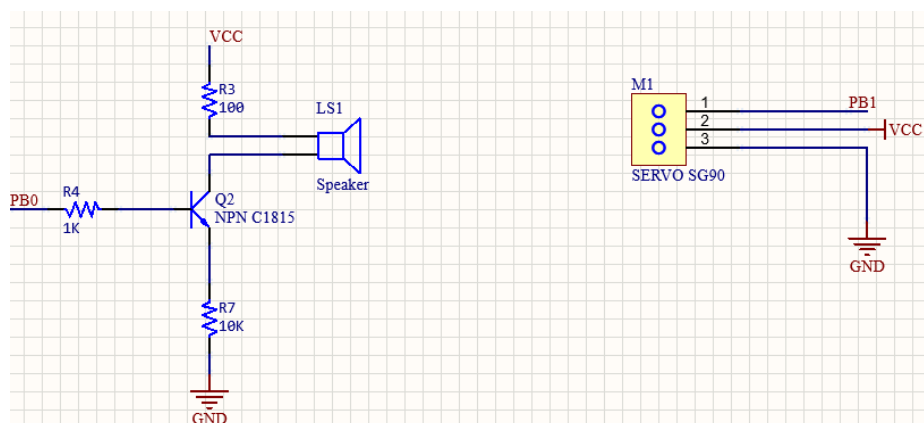
Kết nối vào PORT D của MCU



Hình 12: Thiết kế keypad 4x4

4.4.4 Khối buzzer và motor

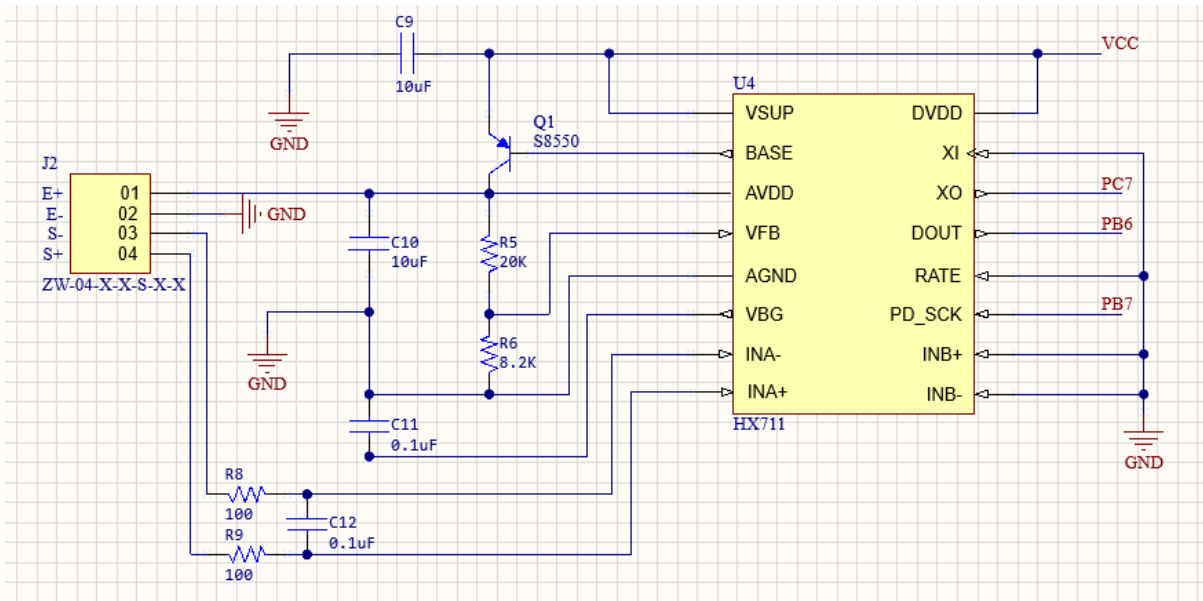
Buzzer được điều khiển bởi BJT thông qua PB0 và motor điều khiển qua PB1



Hình 13: Thiết kế buzzer và motor

4.4.5 Khối HX711 và loadcell

HX711 kết nối với MCU thông qua giao tiếp SCK nên nối 2 chân với nhau



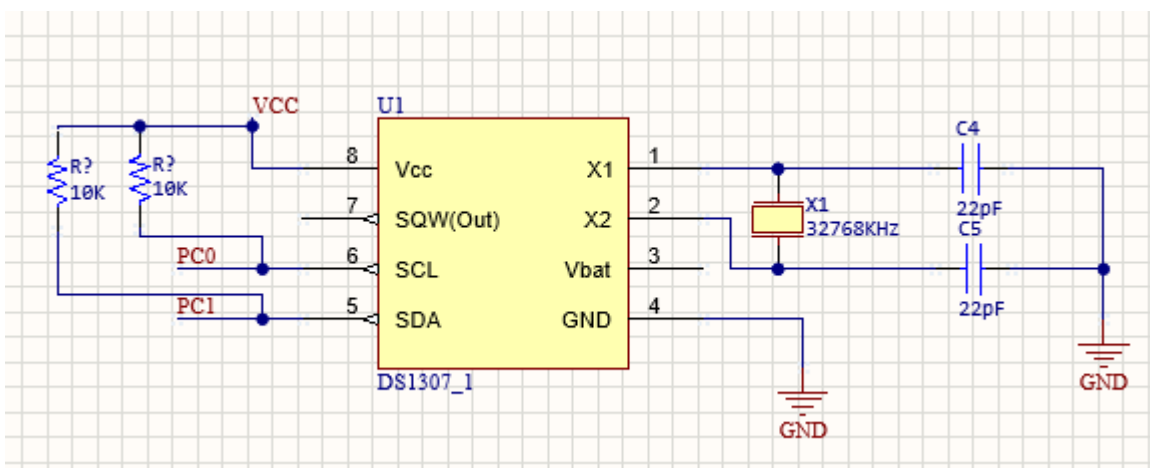
Hình 14: Thiết kế loadcell và HX711

4.4.6 Khối RTC DS1307

RTC DS1307: Chân Xtal1 và Xtal2 gắn vào ngõ ra dao động thạch anh 32 768 Hz.

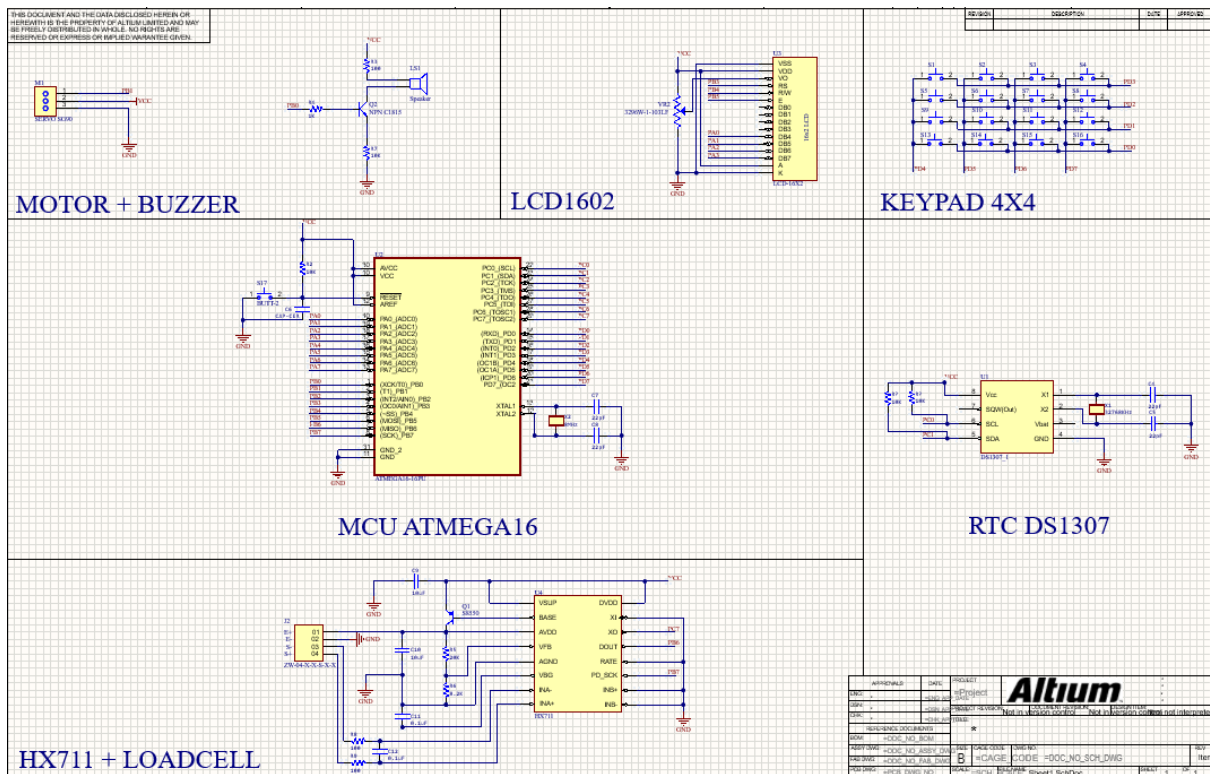
VCC và GND dùng chung với toàn bộ hệ thống. Chân SDA và SCL liên kết với PC0 và PC1

của vi điều khiển.



Hình 15: Thiết kế RTC DS1307

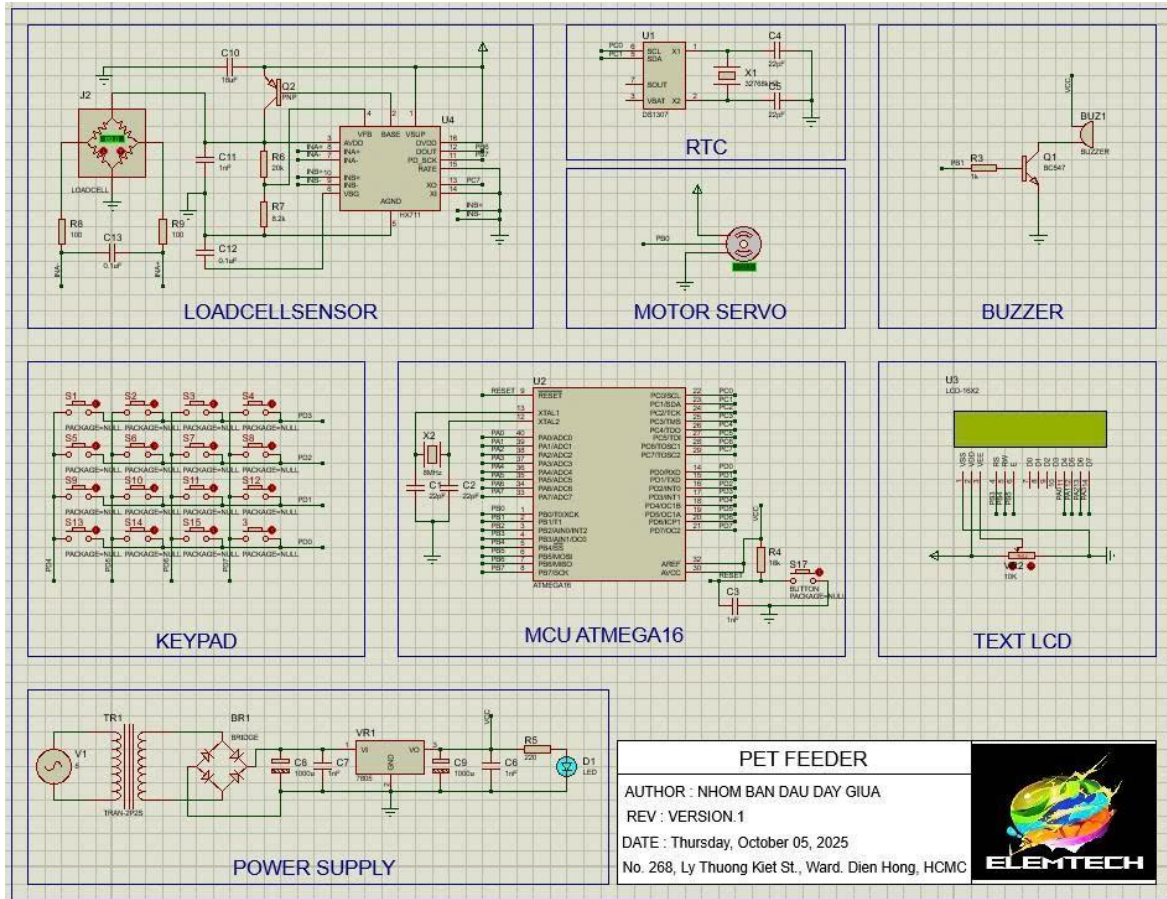
4.4.7 Sơ đồ nguyên lý toàn mạch



Hình 16: Sơ đồ nguyên lý toàn hệ thống

4.5 Thiết kế mạch mô phỏng

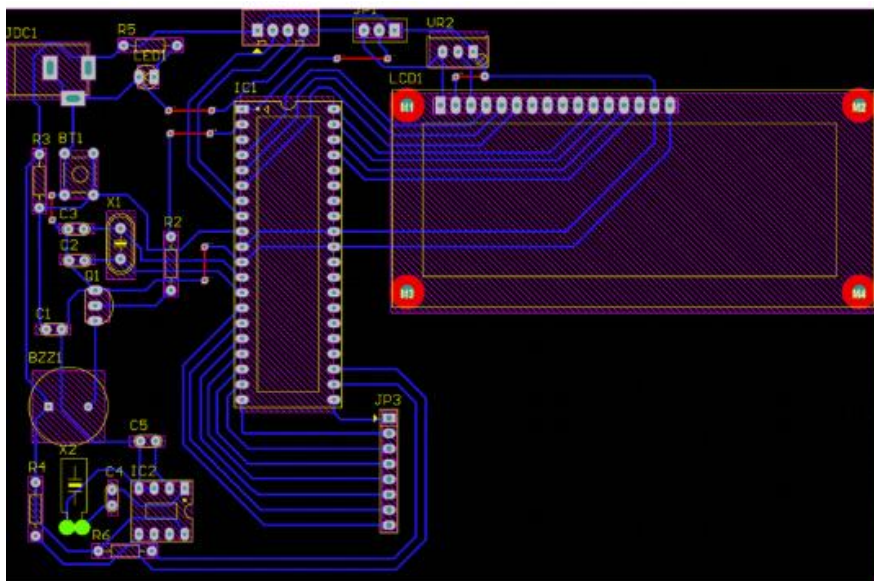
Nhóm chọn mô phỏng bằng proteus và có mạch mô phỏng như sau



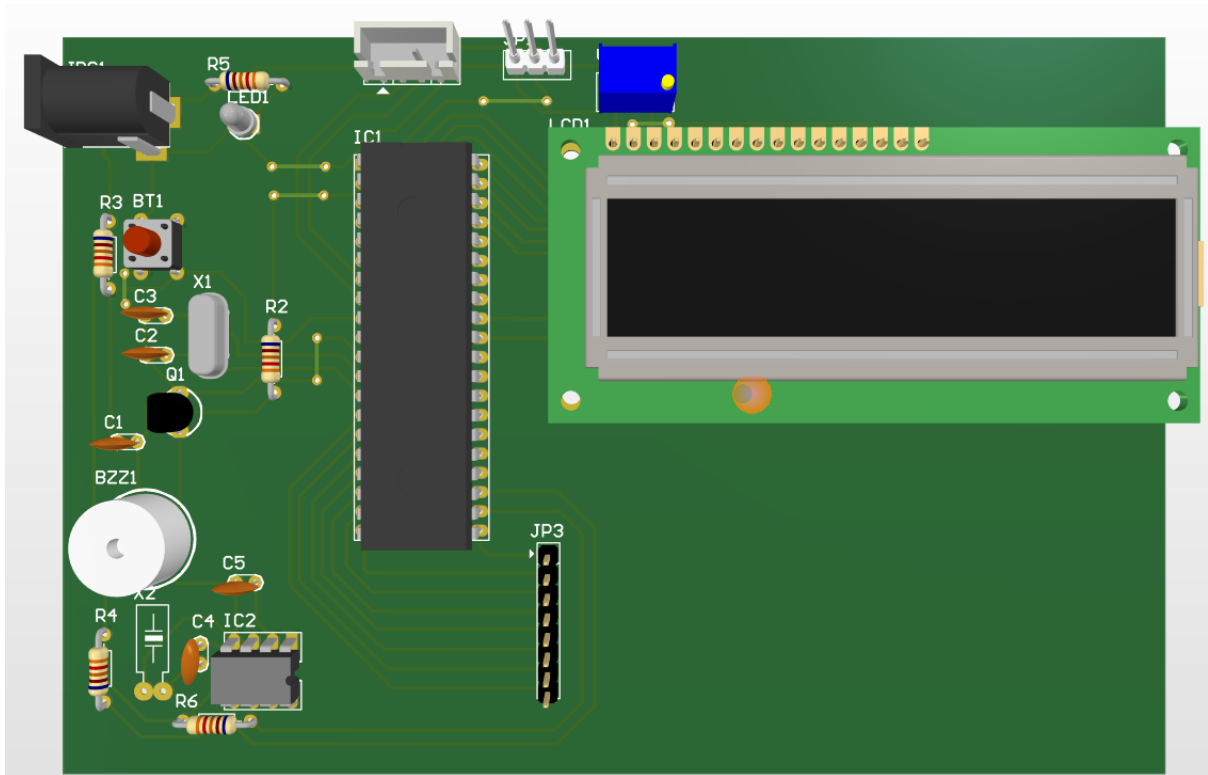
Hình 17: Thiết kế mô phỏng Proteus

4.6 Thiết kế PCB

Nhóm chọn thiết kế mạch layout 1 lớp để dễ dàng tự gia công và hàn mạch, có một số nơi đi layout lớp trên, nhóm sẽ đi dây ngoài.



Hình 18: Thiết kế PCB



Hình 19: 3D của PCB

V. Thiết kế phần mềm

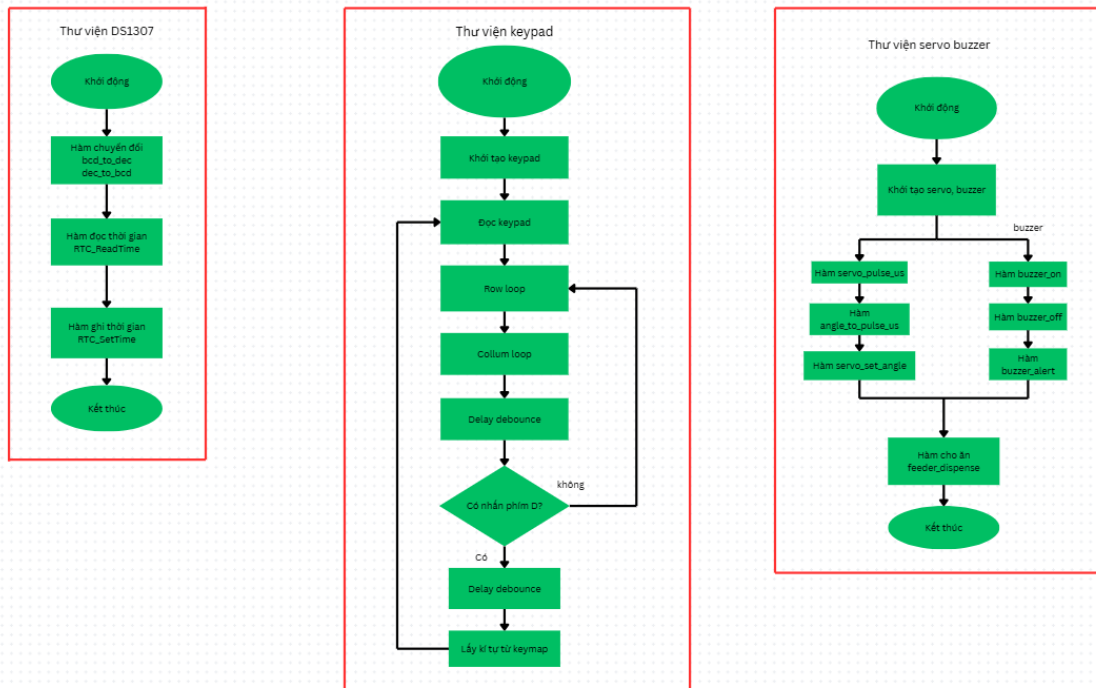
5.1 Thiết kế các thư viện

Nhóm sử dụng các thư viện sau để phù hợp với các khối phần chính ở phần trên:

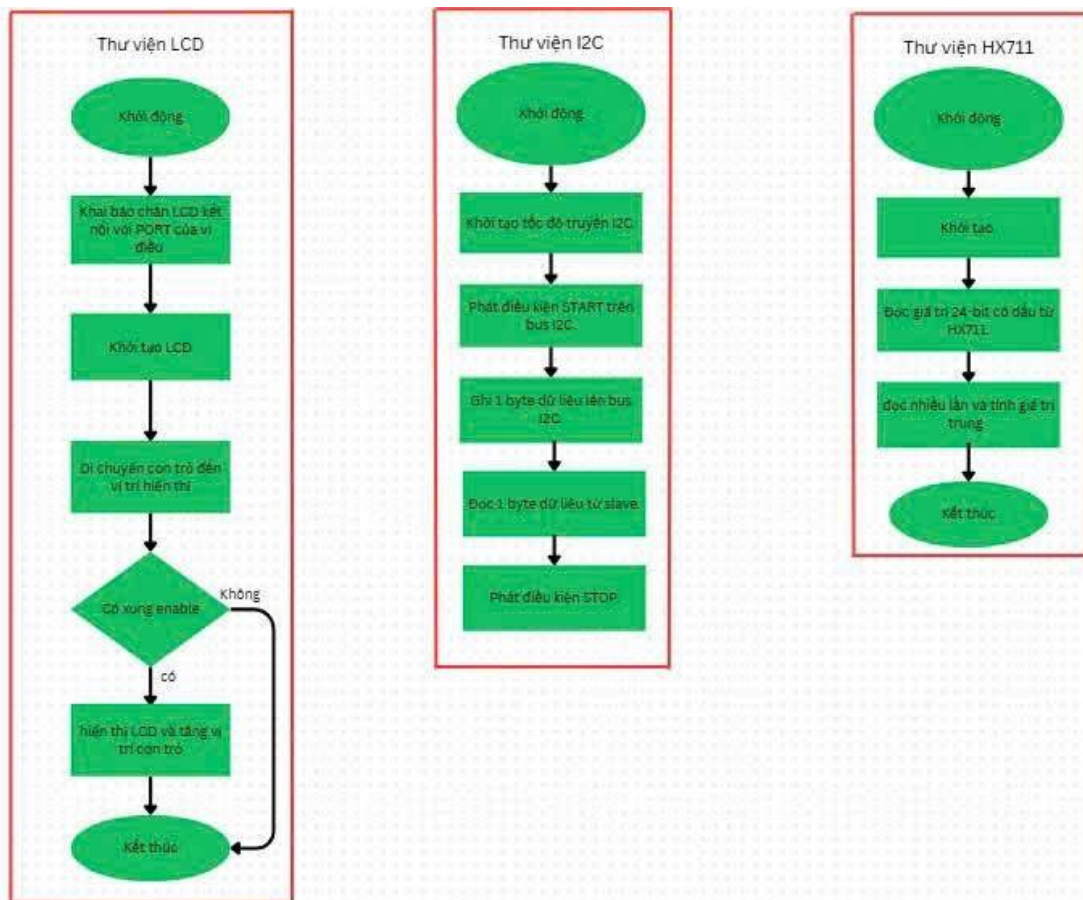
```
// Include các thư viện
#include "lcd.h"
#include "i2c.h"
#include "ds1307.h"
#include "hx711.h"
#include "keypad.h"
#include "servobuzzer.h"
```

Hình 20: Các thư viện cần thiết kế

Và đây là flowchart của các thư viện:



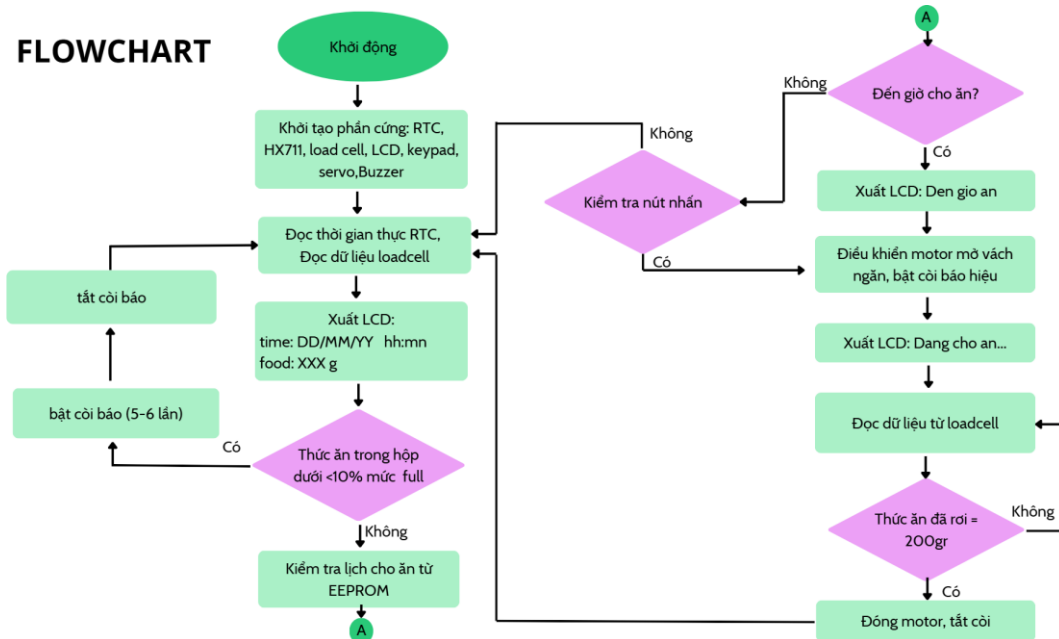
Hình 21: Flowchart của thư viện RTC, keypad, servo buzzer



Hình 22: Flowchart của thư viện HX711, LCD, I2C

5.2 Thiết kế hàm main

5.2.1 Lưu đồ giải thuật



Hình 23: FLOWchart hàm main

5.2.2 Code hàm main

```
#define F_CPU 8000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <stdlib.h>

#include "lcd.h"
#include "i2c.h"
#include "ds1307.h"
#include "hx711.h"
#include "keypad.h"
#include "servobuzzer.h"

int32_t TARE_OFFSET = 0;

uint8_t alarm1_h = 7, alarm1_m = 0, alarm1_s = 0;
uint8_t alarm2_h = 17, alarm2_m = 0, alarm2_s = 0;

void display_set_time_screen(uint8_t alarm_num, uint8_t h, uint8_t m, uint8_t s,
uint8_t cursor_pos) {
```

```

uint8_t lcd_map[] = {0, 1, 3, 4, 6, 7};

lcd_gotoxy(0, 0);
if (alarm_num == 1) lcd_string("SET TIME 1:  ");
else                lcd_string("SET TIME 2:  ");

lcd_gotoxy(0, 1);
lcd_print2(h); lcd_char(':');
lcd_print2(m); lcd_char(':');
lcd_print2(s);
lcd_string(" (D:OK)");
lcd_gotoxy(lcd_map[cursor_pos], 1);
}

void set_alarm_mode(uint8_t alarm_num) {
    uint8_t h, m, s;
    uint8_t pos = 0; // 0->5
    char key;

    if (alarm_num == 1) { h = alarm1_h; m = alarm1_m; s = alarm1_s; }
    else                { h = alarm2_h; m = alarm2_m; s = alarm2_s; }

    lcd_clear();
    lcd_cursor_on(); // B?t con tr?

    display_set_time_screen(alarm_num, h, m, s, pos);

    while(1) {
        key = read_keypad();
        if (key) {
            if (key == 'L') {
                if (pos > 0) pos--;
            }
            else if (key == 'R') {
                if (pos < 5) pos++;
            }

            else if (key == 'D') {
                // Lưu l?i vào bi?n toàn c?c
                if (alarm_num == 1) { alarm1_h = h; alarm1_m = m; alarm1_s = s; }
                else                { alarm2_h = h; alarm2_m = m; alarm2_s = s; }
                break; // Thoát
            }
        }

        // --- NH?P S? (0-9) ---
    }
}

```

```

else if (key >= '0' && key <= '9') {
    uint8_t val = key - '0';
    switch(pos) {
        case 0: h = (h % 10) + val * 10; break; // Gi? ch?c
        case 1: h = (h / 10) * 10 + val; break; // Gi? đon v?
        case 2: m = (m % 10) + val * 10; break; // Phút ch?c
        case 3: m = (m / 10) * 10 + val; break; // Phút đon v?
        case 4: s = (s % 10) + val * 10; break; // Giây ch?c
        case 5: s = (s / 10) * 10 + val; break; // Giây đon v?
    }

    if (h > 23) h = 23;
    if (m > 59) m = 59;
    if (s > 59) s = 59;

    if (pos < 5) pos++;
}

display_set_time_screen(alarm_num, h, m, s, pos);
delay_ms(200); // Ch?ng d?i phím
}
}

lcd_cursor_off(); // T?t con tr?
lcd_clear();
lcd_string("DA LUU CAI DAT!");
delay_ms(1000);
lcd_clear();
}

int main(void)
{
    uint8_t hour, min, sec, old_sec = 0xFF;
    float weight_gram = 0.0;
    int32_t gia_tri_raw, gia_tri_net;
    char key;

    lcd_init();
    I2C_Init();
    hx711_init();
    keypad_init();
    system_init();

    lcd_clear();
    lcd_string("Dang khoi dong..");

```

```

TARE_OFFSET = hx711_get_average(10);
delay_ms(500);
lcd_clear();

RTC_SetTime(14, 34, 00);

while (1)
{
    RTC_ReadTime(&hour, &min, &sec);

    gia_tri_raw = hx711_get_average(5);
    gia_tri_net = gia_tri_raw - TARE_OFFSET;
    weight_gram = (float)gia_tri_net / SCALE_FACTOR;
    if (weight_gram < 0.0f) weight_gram = 0.0f;

    key = read_keypad();

    if (key == 'A') {
        set_alarm_mode(1);
    }
    else if (key == 'B') {
        set_alarm_mode(2);
    }
    else if (key == 'C') {
        lcd_clear();
        lcd_string("Dang cho an...");
        feeder_dispense(); // C?i k?u -> Motor quay
        lcd_clear();
    }

    if (hour == alarm1_h && min == alarm1_m && sec == alarm1_s) {
        lcd_clear();
        lcd_string("DEN GIO AN 1!");
        feeder_dispense(); // Th?c hi?n cho ăn
        lcd_clear();
    }
    // Alarm 2
    else if (hour == alarm2_h && min == alarm2_m && sec == alarm2_s) {
        lcd_clear();
        lcd_string("DEN GIO AN 2!");
        feeder_dispense(); // Th?c hi?n cho ăn
        lcd_clear();
    }
}

```

```

if (weight_gram <= 20.0f) {
    buzzer_alert(1, 100, 0); // Kêu bíp ng?n
}

if (sec != old_sec) {
    old_sec = sec;

    lcd_gotoxy(0, 0);
    lcd_print2(hour); lcd_char(':');
    lcd_print2(min); lcd_char(':');
    lcd_print2(sec);

    // Hàng 2: Kh?i lư?ng "Mass: 123.4g"
    lcd_gotoxy(0, 1);
    lcd_string("Mass: ");
    lcd_float(weight_gram, 1);
    lcd_string("g ");

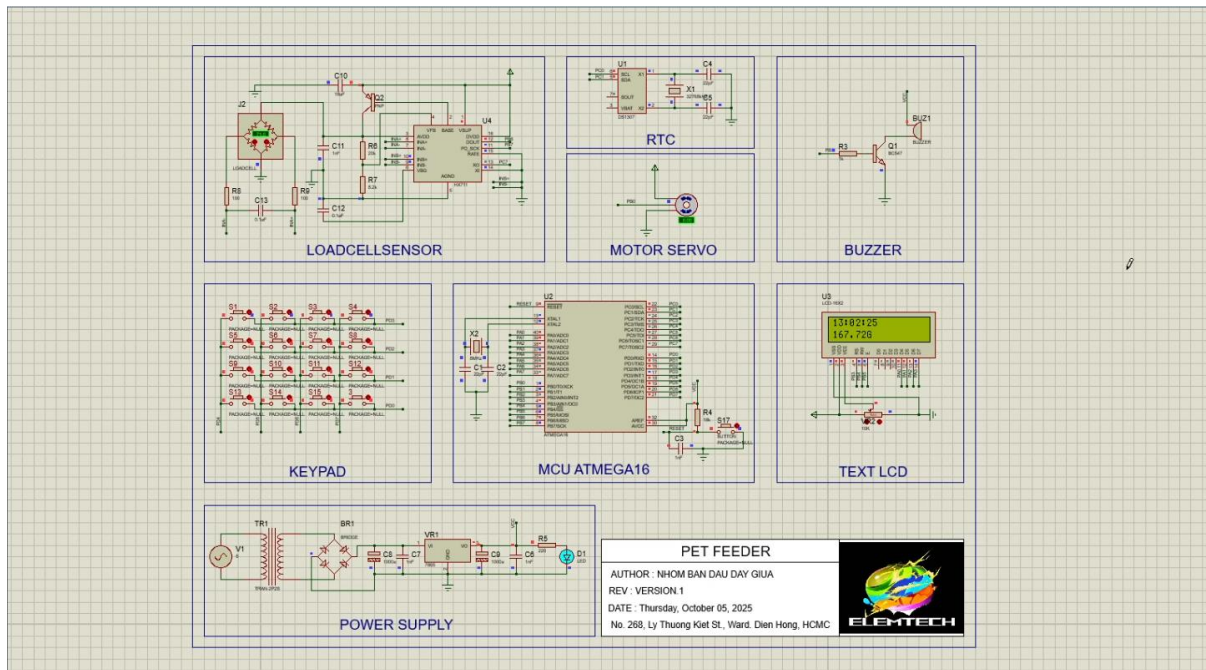
}

    _delay_ms(100);
}
}

```

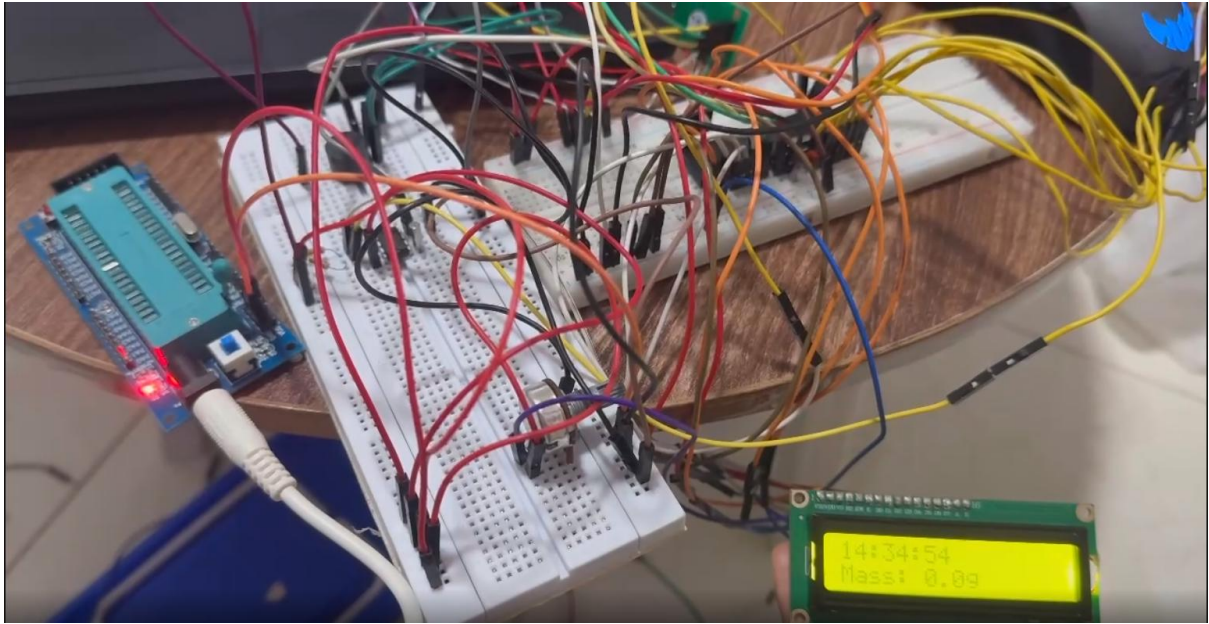
VI. Kết quả

6.1 Kết quả mô phỏng trên proteus



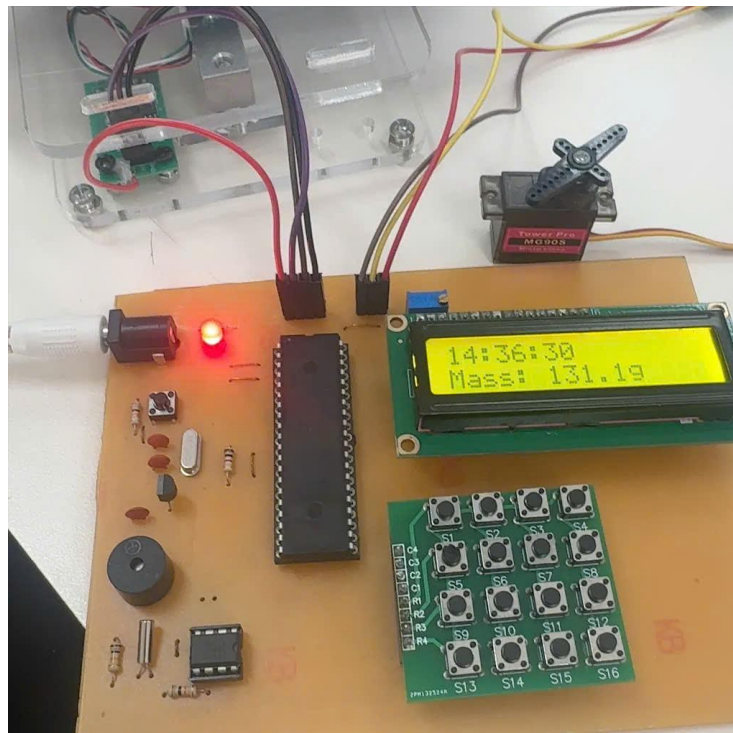
Hình 24: Kết quả mô phỏng proteus

6.2 Kết quả thử nghiệm trên breadboard



Hình 25: Kết quả thử nghiệm trên Breadboard

6.3 Kết quả trên mạch in



Hình 26: Kết quả thực nghiệm PCB

6.4 Kết quả với mô hình



Hình 27: Kết quả với mô hình thực nghiệm

6.5 Kết quả thực tế

Nhóm đã đạt được top 3 nhóm xuất sắc nhất L02 và thuyết trình chạy sản phẩm trước lớp nên video chạy kết quả nhóm xin được thực hiện đơn giản như sau trong link drive:

https://drive.google.com/drive/folders/1TIR9X1_KXfh36y-6WsQdqPRZPQzmsWnW

VII. Kết luận

7.1 Đánh giá đề tài

Sau quá trình thiết kế và thi công, nhóm thực hiện đã tiến hành vận hành thử nghiệm và đưa ra những đánh giá khách quan về ưu điểm cũng như các mặt hạn chế của hệ thống "Pet Feeder System" như sau:

Ưu điểm của hệ thống là hệ thống đã đạt được độ hoàn thiện cao về mặt chức năng và tính ứng dụng thực tế.

Về mặt vận hành: Nhóm đã giải quyết triệt để bài toán nguồn nuôi, cho phép thiết bị hoạt động ổn định bằng nguồn điện sinh hoạt dân dụng (thông qua Adapter) mà không cần các kết nối phức tạp. Các chức năng điều khiển như cài đặt thời gian thực (Set time), định lượng khối lượng (Set weight) và chế độ cho ăn thủ công tức thì đều hoạt động trơn tru, đáp ứng đầy đủ nhu cầu chăm sóc thú cưng cơ bản.

Về mặt quản lý: Điểm mạnh của hệ thống là khả năng hiển thị trực quan thông tin về thời gian thực và khối lượng thức ăn còn lại trong khoang chứa lên màn hình LCD. Tính năng này giúp người dùng dễ dàng kiểm soát lịch trình ăn uống và chủ động nạp thêm thức ăn khi sắp hết.

Về thiết kế cơ khí: Kích thước tổng thể, thể tích thùng chứa và kết cấu cửa xả đã được nhóm tính toán và tối ưu hóa dựa trên các số liệu khảo sát thực tế về thói quen ăn uống của các loài thú cưng phổ biến tại nhà, đảm bảo sự nhỏ gọn nhưng vẫn đủ công suất phục vụ.

Nhược điểm và hạn chế bên cạnh các kết quả đạt được, sản phẩm vẫn tồn tại một số hạn chế cần khắc phục trong các phiên bản sau:

Thiết kế vỏ hộp và độ bền: Vỏ thiết bị hiện tại được gia công còn khá sơ sài, chưa đảm bảo tính thẩm mỹ công nghiệp và độ kín khít cần thiết. Điều này dẫn đến rủi ro mạch in (PCB) bên trong dễ bị tác động bởi độ ẩm môi trường hoặc bụi bẩn, có thể gây chập chập hoặc giảm tuổi thọ linh kiện theo thời gian.

Thiếu cơ chế phản hồi (Feedback): Hệ thống hoạt động theo nguyên lý vòng hở (Open-loop) đối với bát ăn. Cụ thể, máy chưa có cảm biến để phát hiện thức ăn cũ trong bát còn hay hết. Nếu thú cưng không ăn hết khẩu phần cũ, máy vẫn tiếp tục xả thêm vào giờ ăn tiếp theo, gây ra tình trạng tràn thức ăn ra ngoài, gây lãng phí và mất vệ sinh.

Hạn chế về kết nối: Hệ thống chưa hỗ trợ các giao thức kết nối không dây (IoT). Người dùng bắt buộc phải thao tác trực tiếp trên các nút nhấn tại máy để cài đặt, gây bất tiện nếu muốn thay đổi lịch trình từ xa hoặc khi vắng nhà dài ngày.

7.2 Hướng phát triển

Để hoàn thiện sản phẩm và nâng cao tính thương mại, nhóm đề xuất các hướng cải tiến sau:

Tích hợp IoT: Nâng cấp vi điều khiển (hoặc bổ sung module ESP8266/ESP32) để kết nối WiFi, cho phép người dùng điều khiển, giám sát lượng thức ăn và nhận cảnh báo qua ứng dụng trên điện thoại thông minh (Mobile App).

Bổ sung cảm biến tại bát ăn: Lắp đặt thêm cảm biến hồng ngoại (IR Sensor) hoặc cảm biến tiệm cận tại vị trí bát ăn. Hệ thống sẽ kiểm tra nếu bát còn đầy thì sẽ tạm hoãn lần cho ăn tiếp theo để tránh tình trạng tràn thức ăn.

Nguồn dự phòng (Backup Power): (Ý này rất quan trọng với máy cho ăn) Bổ sung slot pin dự phòng hoặc pin sạc. Trong trường hợp nhà mất điện, hệ thống sẽ tự động chuyển sang dùng pin để đảm bảo thú cưng không bị bỏ đói, giữ đúng vai trò của một thiết bị tự động tin cậy.

Cải thiện vỏ hộp: Thiết kế và in 3D vỏ hộp hoặc sử dụng mica cắt CNC để tăng tính thẩm mỹ và bảo vệ mạch điện tốt hơn.

VIII Tài liệu tham khảo

1. Hoàng Trang, Lưu Phú, Giáo trình Vi xử lý. Nhà xuất bản Đại học Quốc gia TP. Hồ Chí Minh
2. Embedded C programming and ATMEL AVR 2nd Edition