

TRƯỜNG ĐẠI HỌC KỸ THUẬT- CÔNG NGHỆ CẦN THƠ
KHOA KỸ THUẬT CƠ KHÍ



ĐỒ ÁN THỰC TẬP THỰC TẾ
ĐỀ TÀI
LẬP TRÌNH VÀ ĐIỀU KHIỂN ĐỘNG CƠ BƯỚC
SỬ DỤNG ARDUINO

GIẢNG VIÊN HƯỚNG DẪN	SINH VIÊN THỰC HIỆN	MSSV
<i>TS. Đỗ Vinh Quang</i>	<i>Trương Minh Hiếu</i>	<i>2000326</i>
	<i>Nguyễn Minh Quân</i>	<i>2000456</i>

Ngành: Công nghệ kỹ thuật điều khiển và tự động hóa

Cần thơ, Tháng 12 năm 2023

TRƯỜNG ĐẠI HỌC KỸ THUẬT - CÔNG NGHỆ CẦN THƠ
KHOA KỸ THUẬT CƠ KHÍ



ĐỒ ÁN THỰC TẬP THỰC TẾ

**ĐỀ TÀI : LẬP TRÌNH VÀ ĐIỀU KHIỂN ĐỘNG CƠ BƯỚC
SỬ DỤNG ARDUINO**

GIẢNG VIÊN HƯỚNG DẪN	SINH VIÊN THỰC HIỆN	MSSV
<i>TS. Đỗ Vinh Quang</i>	<i>Trương Minh Hiếu</i>	<i>2000326</i>
	<i>Nguyễn Minh Quân</i>	<i>2000456</i>

Ngành: Công nghệ kỹ thuật điều khiển và tự động hóa

Cần thơ, Tháng 12 năm 2023

LỜI CAM ĐOAN

Tôi xin cam đoan rằng đây là công trình nghiên cứu của tôi, có sự hỗ trợ từ giảng viên hướng dẫn là TS. Đỗ Vinh Quang Các nội dung nghiên cứu và kết quả trong đề tài này là trung thực và chưa từng được ai công bố trong bất cứ công trình nghiên cứu nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi trong phần tài liệu tham khảo.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm trước Hội đồng cũng như kết quả đồ án của mình

Cần Thơ, ngày...tháng...năm 2023

Giảng viên hướng dẫn

Tác giả thực hiện

LỜI CẢM ƠN

Báo cáo đồ án đề tài “Lập trình và điều khiển động cơ bước sử dụng arduino ” là kết quả của quá trình cố gắng không ngừng của bản thân và được sự giúp đỡ, động viên khích lệ của các thầy, bạn bè đồng nghiệp và người thân. Qua trang viết này tác giả xin gửi lời cảm ơn tới những người đã giúp đỡ tôi trong thời gian học tập - nghiên cứu khoa học vừa qua.

Tôi xin tỏ lòng kính trọng và biết ơn sâu sắc đối với TS. Đỗ Vinh Quang đã trực tiếp tận tình hướng dẫn cũng như cung cấp tài liệu thông tin khoa học cần thiết cho luận văn này.

Xin chân thành cảm ơn Lãnh đạo trường Đại học Kỹ thuật – Công nghệ Cần Thơ, khoa Kỹ thuật cơ khí đã tạo điều kiện cho tôi hoàn thành tốt công việc nghiên cứu khoa học của mình.

Do sự eo hẹp về thời gian, do sự hiểu biết còn hạn chế của bản thân, mặc dù em đã rất cố gắng nhưng đồ án này còn nhiều thiếu sót. Tôi rất mong nhận được nhiều ý kiến đóng góp, bổ sung từ phía thầy, cô, bạn bè và những người quan tâm đến đề tài này.

Đồ án là cơ hội cho tôi một lần nữa kiểm tra và đánh giá lại kiến thức mà mình đã học được trên ghế nhà trường. Trong quá trình thực tập và làm đồ án, tôi đã được TS. Đỗ Vinh Quang hướng dẫn chỉ bảo nhiệt tình để em hoàn thành tốt đồ án thực tập thực tế của mình.

Cần Thơ, ngày 21 tháng 12 năm 2023

Tôi xin chân thành cảm ơn!

NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Mục lục

LỜI CAM ĐOAN	2
LỜI CẢM ƠN.....	3
NHẬN XÉT CỦA GIẢNG VIÊN	4
I. Lí do chọn đề tài.....	7
II Tính cấp thiết của đề tài	7
III.Đối tượng nghiên cứu, phạm vi nghiên cứu, phương tiện nghiên cứu, phương pháp nghiên cứu, kết quả dự kiến	8
1 Đối tượng nghiên cứu	8
2 Phạm vi nghiên cứu	8
3 Phương tiện nghiên cứu	8
4 Phương pháp nghiên cứu	8
5 Kết quả dự kiến.....	8
IV. NỘI DUNG	9
Chuẩn bị các linh kiện	9
Ưu điểm của Động cơ Bước:	11
Nhược Điểm của Động cơ Bước:	12
Arduino Uno	12
Phương pháp thực hiện điều khiển động cơ bước	14
Xác định dây của động cơ bước.....	14
Điều khiển động cơ bước	15
Các định nghĩa quan trọng:	17
Timer/Counter 1	17
Chế độ Fast PWM của Timer/Counter 1	19
Điều khiển số xung PWM để điều khiển động cơ trên Timer/Counter 1	21
Timer/Counter 2	21
Chế độ Fast PWM của Timer/Counter 2	22
Điều khiển số xung PWM để điều khiển động cơ trên Timer/Counter 2	24
Cách tính số xung để điều khiển động cơ bước.....	24
+Lưu đồ giải thuật.....	26

+ Sơ đồ nguyên lí.....	27
IV. Kết luận	28

I. Lí do chọn đề tài

Lý do chọn đề tài này xuất phát từ nhận thức về sự cần thiết phải nâng cao hiệu suất và linh hoạt trong việc lập trình và điều khiển động cơ bước. Các nghiên cứu trước đây đã tập trung vào khía cạnh cơ bản của động cơ bước, nhưng chúng tôi tin rằng sự hiểu biết sâu sắc về lập trình và điều khiển sẽ mở ra những cánh cửa mới cho ứng dụng rộng rãi.

Ngoài ra, với sự gia tăng của Internet of Things (IoT) và sự phổ biến của các hệ thống tự động, việc hiểu rõ cách lập trình và điều khiển động cơ bước có thể tích hợp vào các hệ thống tổng thể trở nên ngày càng quan trọng. Đề tài này không chỉ mang tính ứng dụng mà còn đáp ứng nhu cầu ngày càng tăng về sự tự động và hiệu suất cao trong các hệ thống tự động hóa.

Qua đó, nghiên cứu này nhằm mục tiêu cung cấp một cái nhìn tổng quan về cách lập trình và điều khiển động cơ bước để đáp ứng nhu cầu ngày càng đa dạng và phức tạp của công nghiệp hiện đại.

II Tính cấp thiết của đề tài

Đề tài này không chỉ là một bước tiến quan trọng trong nghiên cứu về động cơ bước mà còn đặt ra một loạt các thách thức và cơ hội mới. Tính cấp thiết của nghiên cứu này có thể được nhìn nhận qua một số góc độ: Nâng Cao Hiệu Suất Lập trình và điều khiển động cơ bước một cách hiệu quả có thể mang lại cải thiện đáng kể về hiệu suất, giảm độ trễ và tăng khả năng đáp ứng của hệ thống. Tích Hợp trong Công Nghiệp 4.0 Trong bối cảnh Công Nghiệp 4.0, nghiên cứu này đáp ứng nhu cầu ngày càng cao về sự tự động và tích hợp thông tin, giúp các hệ thống trở nên linh hoạt và thông minh hơn. Giảm Chi Phí Sản Xuất Hiểu rõ cách lập trình và điều khiển động cơ bước cũng mở ra cơ hội giảm chi phí sản xuất thông qua tối ưu hóa quy trình và sử dụng hiệu quả năng lượng. Áp Dụng Rộng Rãi Khả năng áp dụng của nghiên cứu này không chỉ giới hạn trong ngành công nghiệp, mà còn mở ra các ứng dụng trong y tế, robot học, và nhiều lĩnh vực khác. Với những thách thức và cơ hội trên, việc lập trình và điều

khiển động cơ bước trở thành một ưu tiên cấp thiết, đặc biệt trong bối cảnh môi trường công nghiệp ngày nay đang đòi hỏi sự đổi mới và hiệu suất tối đa.

III. Đối tượng nghiên cứu, phạm vi nghiên cứu, phương tiện nghiên cứu, phương pháp nghiên cứu, kết quả dự kiến

1 Đối tượng nghiên cứu

- DRV8825 + đế ra chân DRV
- Động cơ bước
- Arduino Uno

2 Phạm vi nghiên cứu

- Nghiên cứu hệ thống cấu trúc của vi điều khiển Arduino Uno và tập lệnh của nó trên phần mềm Arduino IDE, cấu tạo và nguyên lý hoạt động của Driver8825 và động cơ bước
- Không gian: Nghiên cứu tại Trường Đại học Kỹ Thuật – Công Nghệ Cần Thơ.
- Thời gian: Bắt đầu nghiên cứu từ ngày 19/9/2022 đến ngày 31/12/2022.

3 Phương tiện nghiên cứu

- Sách tài liệu và giáo trình về vi điều khiển
- Nghiên cứu thực tiễn, quan sát, mô phỏng

4 Phương pháp nghiên cứu

- Dựa trên cơ sở lý thuyết để nghiên cứu hệ thống
- Mô hình hóa sản phẩm giúp phát hiện những vấn đề mà phương pháp lý thuyết còn hạn chế

5 Kết quả dự kiến

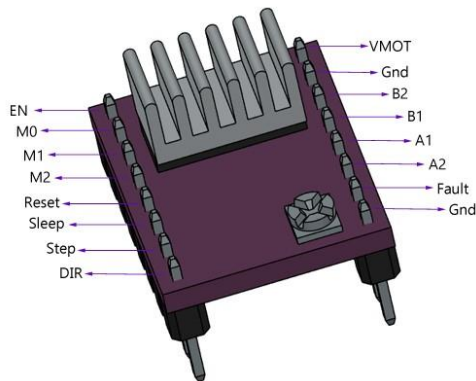
Sản phẩm là thành quả từ những kiến thức lập trình, sự thông hiểu về vi điều khiển.

IV. NỘI DUNG

Chuẩn bị các linh kiện

Tên linh kiện	Số lượng
Động cơ bước	2
DRV8825 + đế ra chân	2
Arduino Uno	1
Nút nhấn	3
LCD+I2C	1

DRV8825



Hình 4.1 DRV8825

Điện áp cung cấp: 8.2~45VDC

Dòng trung bình (RMS) : 1.5A , dòng đỉnh (Peak) lên đến 2.5A.

6 độ phân giải bước khác nhau: full, half step, 1/4 step, 1/8 step, 1/16 step, 1/32 step.

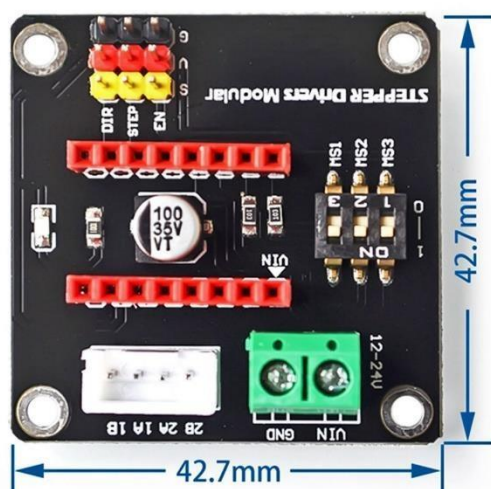
Điện áp điều khiển: 3.3V và 5V.

Tự động shutdown khi quá nhiệt, quá dòng

Bảo vệ ngắn mạch và bảo vệ quá tải.

Mạch 4 lớp, 2 lớp phủ đồng giúp cải thiện khả năng tản nhiệt .

-Đế ra chân DRV8825



Tích hợp cho driver: A4988,
DRV8825

Điện áp logic: 5v

Điện áp đầu vào động cơ: 12v – 30v

-Động cơ bước

Động cơ bước, là một loại động cơ điện một chiều có khả năng chuyển động theo các bước rời rạc. Nó bao gồm nhiều cuộn dây được tổ chức thành các nhóm, được gọi là pha. Bằng cách cung cấp năng lượng cho từng pha theo thứ tự, động cơ sẽ quay từng bước một. Stepper Motor thực chất là một động cơ đồng bộ được sử dụng để chuyển đổi tín hiệu điều khiển thành các chuyển động góc quay hoặc chuyển động của Roto, có khả năng giữ Roto ở các vị trí cố định.

Với khả năng điều khiển chính xác từ máy tính, Stepper Motor thường được lựa chọn cho các ứng dụng yêu cầu chuyển động chính xác.

Cấu tạo của nó bao gồm hai thành phần chính là Stato và Roto. Stato được làm từ sắt với nhiều rãnh nhỏ để đặt cuộn dây, trong khi Roto (nam châm vĩnh cửu) được xếp chồng lên nhau một cách cẩn thận, có thể được sắp xếp thành các cặp đối xứng và được sử dụng để tạo ra chuyển động bước.

Nguyên lý hoạt động của động cơ stepper motor dựa trên việc quay theo từng bước rời rạc, không giống như cơ chế quay liên tục của động cơ thông thường. Điều này mang lại độ chính xác cao trong việc điều khiển chuyển động. Động cơ bước hoạt động thông qua các bộ chuyển mạch điện tử, và nguyên tắc hoạt động được mô tả như sau:

Tín Hiệu Điều Khiển:

Các mạch điều khiển chuyển đổi tín hiệu lệnh từ máy tính hoặc bộ điều khiển vào các cuộn dây của động cơ theo thứ tự xác định.

Chuyển Mạch Điện Tử:

Các chuyển mạch điện tử mở hoặc đóng từng cuộn dây theo thứ tự và tần số nhất định. Điều này tạo ra các bước rời rạc trong quá trình quay của động cơ.

Góc Quay và Tốc Độ:

Tổng số bước quay của rotor tương ứng với số lần chuyển mạch được thực hiện.

Chiều quay và tốc độ của rotor phụ thuộc vào thứ tự và tần số chuyển mạch.

Cụ thể, có ba dạng điều khiển chính:

+Điều khiển dạng Full-Step (Wave Drive - 1 phase on):

Mỗi lần chỉ có một cuộn dây được cung cấp năng lượng.

Ưu điểm là đơn giản, nhưng công suất thấp, gây ra lực xoắn yếu.

+Điều khiển dạng Full-Step (Two Phase Drive - 2 phase on):

Hai cuộn dây được cung cấp năng lượng đồng thời, tăng công suất so với điều khiển dạng 1 phase.

+Điều khiển dạng Half-Step (1 or 2 phase on):

Kết hợp giữa hai dạng trên, tạo ra các bước nhỏ hơn giữa các bước lớn, cải thiện độ chính xác và lực xoắn so với dạng điều khiển full-step.

Ưu điểm của Động cơ Bước:

Mô Men Xoắn Lớn ở Dải Vận Tốc Trung Bình và Thấp: Động cơ bước cung cấp mô men xoắn lớn ở các dải vận tốc trung bình và thấp, giúp nó phù hợp cho nhiều ứng dụng.

Điều Chỉnh Chính Xác Góc Quay: Có khả năng điều chỉnh chính xác góc quay, giúp đáp ứng tốt trong các ứng dụng yêu cầu độ chính xác cao về vị trí.

Tuổi Thọ Lâu Dài và Bền Bỉ: Động cơ bước thường có tuổi thọ lâu dài và khả năng hoạt động bền bỉ, giảm cần thiết về bảo trì.

Dễ Lắp Đặt và Thay Thế: Dễ dàng lắp đặt và thay thế, giúp tăng tính linh hoạt trong quá trình sử dụng.

Giá Cả Thấp: Động cơ bước có giá thành thấp, làm cho chúng trở thành lựa chọn phổ biến cho nhiều ứng dụng.

Nhược Điểm của Động cơ Bước:

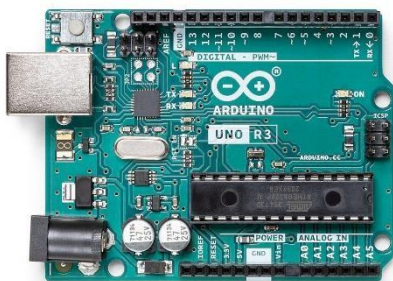
Hiện Tượng Trượt Bước: Khi lực từ nam châm vĩnh cửu yếu, động cơ có thể trượt bước, dẫn đến vị trí không chính xác hoặc khi nguồn điện cung cấp không đủ. *Ồn và Nhiệt:* Động cơ bước có thể tạo ra tiếng ồn và nóng lên khi hoạt động, tuy nhiên, các công nghệ mới giúp giảm độ ồn và nhiệt độ.

Nhiều và Rung Động: Gây ra nhiều nhiễu và rung động hơn so với động cơ servo, điều này có thể ảnh hưởng đến chất lượng của các ứng dụng yêu cầu sự ổn định.

Không Thích Hợp cho Tốc Độ Cao: Không phù hợp cho các ứng dụng đòi hỏi tốc độ cao, đặc biệt khi so sánh với động cơ servo.

Mặc dù có nhược điểm nhất định, động cơ bước vẫn được ưa chuộng trong nhiều ứng dụng nhờ vào ưu điểm về chi phí, độ chính xác, và độ linh hoạt trong điều khiển.

Arduino Uno



Hình 4.4 Arduino Uno

Vi điều khiển	ATmega328
Điện áp hoạt động	5V(cấp qua cổng usb)
Điện áp khuyến nghị	6-9V
Số chân digital I/O	14 chân(6 chân PWM)
Số chân analog	6 chân

LCD+I2C



- Thông số kỹ thuật của LCD 1602
- + Điện áp MAX: 7V
- + Điện áp MIN: - 0,3V
- + Hoạt động ổn định: 2.7-5.5V
- + Điện áp ra mức cao: > 2.4
- + Điện áp ra mức thấp: <0.4V
- + Dòng điện cấp nguồn: 350uA - 600uA
- + Nhiệt độ hoạt động: - 30 - 75 độ C -

Chức năng của từng chân LCD 1602:

- + Chân số 1 - VSS: chân nối đất cho LCD được nối với GND của mạch điều khiển
- + Chân số 2 - VDD: chân cấp nguồn cho LCD, được nối với VCC=5V của mạch điều khiển
- + Chân số 3 - VE: điều chỉnh độ tương phản của LCD

- + Chân số 4 – RS: chân chọn thanh ghi, được nối với logic "0" hoặc logic "1":
 - Logic “0”: Bus DB0 - DB7 sẽ nối với thanh ghi lệnh IR của LCD (ở chế độ “ghi” - write) hoặc nối với bộ đếm địa chỉ của LCD (ở chế độ “đọc” - read)
 - Logic “1”: Bus DB0 - DB7 sẽ nối với thanh ghi dữ liệu DR bên trong LCD
- + Chân số 5 - R/W: chân chọn chế độ đọc/ghi (Read/Write), được nối với logic “0” để ghi hoặc nối với logic “1” đọc
- + Chân số 6 - E: chân cho phép (Enable). Sau khi các tín hiệu được đặt lên bus DB0-DB7, các lệnh chỉ được chấp nhận khi có 1 xung cho phép của chân này như sau:
 - Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào thanh ghi bên trong khi phát hiện một xung (high-to-low transition) của tín hiệu chân E
 - Ở chế độ đọc: Dữ liệu sẽ được LCD xuất ra DB0-DB7 khi phát hiện cạnh lên (low-to-high transition) ở chân E và được LCD giữ ở bus đến khi nào chân E xuống mức thấp
 - Chân số 7 đến 14 - D0 đến D7: 8 đường của bus dữ liệu dùng để trao đổi thông tin với MPU. Có 2 chế độ sử dụng 8 đường bus này là: Chế độ 8 bit (dữ liệu được truyền trên cả 8 đường, với bit MSB là bit DB7) và Chế độ 4 bit (dữ liệu được truyền trên 4 đường từ DB4 tới DB7, bit MSB là DB7)
 - Chân số 15 - A: nguồn dương cho đèn nền
 - Chân số 16 - K: nguồn âm cho đèn nền

Phương pháp thực hiện điều khiển động cơ bước

Xác định dây của động cơ bước

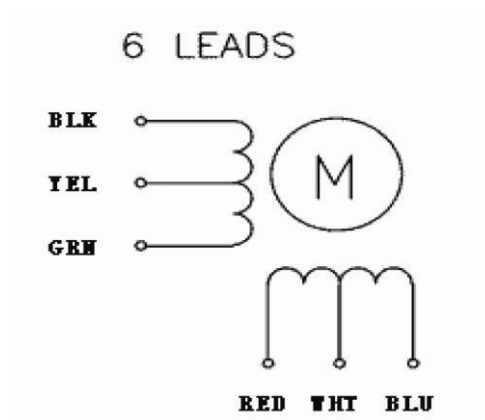
Động cơ nhóm sử dụng là loại động cơ bước đơn cực có 6 đầu dây gồm có cặp dây A1/B1 và dây chung, cặp dây A2/B2 và dây chung khi điều khiển động cơ ta bỏ dây chung và cắm các chân A1/B1, A2/B2 vào driver để điều khiển

Để xác định các dây A1/B1, A2/B2 ta tiến hành đo điện trở các dây ta được bảng như sau:

A1/Dây chung	B1/Dây chung	A1/B1	A2/Dây chung	B2/Dây chung	A2/B2
3.5 ohm	3.5 ohm	6 ohm	1.6 ohm	1.6 ohm	3 ohm

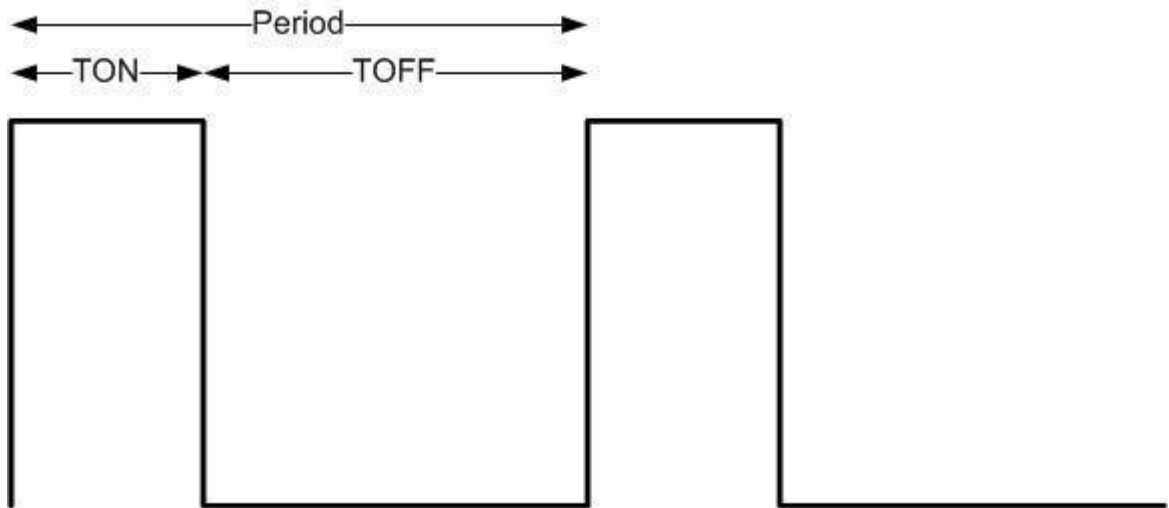
Động cơ 1

A1/Dây chung	B1/Dây chung	A1/B1	A2/Dây chung	B2/Dây chung	A2/B2
1.5 ohm	1.5 ohm	3.3 ohm	1.5 ohm	1.5 ohm	3.3 ohm



Điều khiển động cơ bước

Để điều khiển động cơ bước ta cần phát xung vuông với độ rộng là 50%



$$\text{Period} = \text{TON} + \text{TOFF}$$

$$\text{Frequency} = 1 / \text{Period}$$

$$\text{Duty Cycle} = \frac{\text{TON}}{\text{TON} + \text{TOFF}} * 100$$

Qua thực nghiệm nhóm quyết định sử dụng xung vuông có tần số là 1khz và độ rộng xung là 50% để điều khiển động cơ vì ở mức này động cơ hoạt động tốt nhất.

Từ tần số ta có công thức tính chu kỳ như sau:

$$T = \frac{1}{f}$$

Trong đó: T

là chu kỳ f

là tần số

Thay 1khz vào công thức trên ta có:

$$T = \frac{1}{1000} = 0.001 \text{ sec} = 1000 \mu\text{sec}$$

Với độ rộng xung là 50% ta tính được thời gian mức cao là 500μs và mức thấp là 500μs

2 động cơ bước được điều khiển bất đồng bộ bằng cách sử dụng chế độ Fast PWM của 2 timer 1 và 2 của vi điều khiển Atmega328P trong Arduino R3

Timer/counter 1 là timer counter 16 bit đa năng trong arduino gồm 5 chế độ hoạt động:

Timer/counter 2 là timer counter 8 bit gồm có 4 chế độ hoạt động

Các chế độ hoạt động của timer/counter 1 như sau:

- Chế độ Normal
- Chế độ CTC
- Chế độ Fast PWM
- Chế độ Phase Correct PWM
- Chế độ Phase and Frequency Correct PWM

Các chế độ hoạt động của timer/counter 2 như sau:

- Chế độ Normal
- Chế độ CTC
- Chế độ Fast PWM
- Chế độ Chế độ Phase Correct PWM

Các định nghĩa quan trọng:

BOTTOM: là giá trị thấp nhất mà 1 Timer/Counter đạt được, BOTTOM luôn bằng 0.

MAX: là giá trị lớn nhất mà 1 Timer/Counter đạt được, ở thanh ghi 8 bit giá trị MAX = $2^8 - 1 = 255$, ở thanh ghi 16 bit giá trị MAX = $2^{16} - 1 = 65535$. Giá trị MAX là cố định với từng Timer/Counter.

TOP: là giá trị đỉnh mà tại có Timer/Counter thay đổi trạng thái, giá trị TOP không nhất thiết phải bằng MAX mà có thể thay đổi bằng các thanh ghi

Interrupt: (còn gọi là Ngắt) là 1 chương trình có độ ưu tiên cao nhất, được thực hiện ngay lập tức khi có tín hiệu Interrupt

Timer/Counter 1

Gồm có 2 kênh OC1A và OC1B tương ứng với 2 chân số 9 và số 10 trên Arduino.

Gồm có các thanh ghi sau:

Thanh ghi TCNT1 là thanh ghi 16bit lưu giá trị đếm của timer/counter cho phép đọc ghi trực tiếp

Bit	7	6	5	4	3	2	1	0	
(0x85)	TCNT1[15:8]								TCNT1H
(0x84)	TCNT1[7:0]								TCNT1L
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Thanh ghi TCCR1A (Timer/Counter1 Control RegisterA) và TCCR1B(Timer/Counter1 Control Register B)là thanh ghi điều khiển hoạt động của timer/counter 1

Bit	7	6	5	4	3	2	1	0	
(0x80)	COM1A1	COM1A0	COM1B1	COM1B0	—	—	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
(0x81)	ICNC1	ICES1	—	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Thanh ghi TIMSK1 (Timer/Counter1 Interrupt Mask Register) là thanh ghi lưu trữ các Interrupt Mask của Timer/Counter 1

Bit	7	6	5	4	3	2	1	0	
(0x6F)	—	—	ICIE1	—	—	OCIE1B	OCIE1A	TOIE1	TIMSK1
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

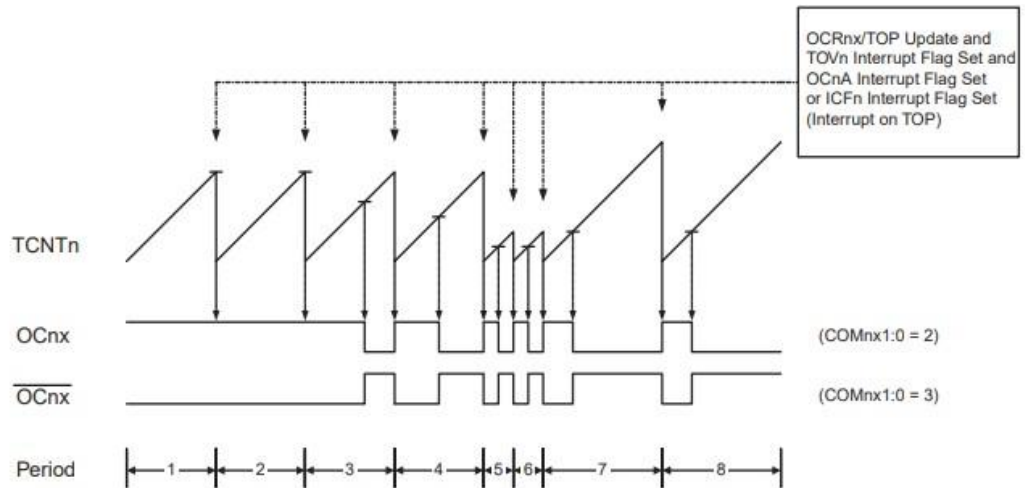
Thanh ghi OCR1A (Output Compare Register 1 A) là thanh ghi lưu giá trị so sánh đầu ra

Bit	7	6	5	4	3	2	1	0	
(0x89)	OCR1A[15:8]								OCR1AH
(0x88)	OCR1A[7:0]								OCR1AL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Thanh ghi ICR1 (Input Capture Register 1) dùng để lưu giá trị TOP khi dùng chế độ Fast PWM của timer/counter 1

Bit	7	6	5	4	3	2	1	0	
(0x87)	ICR1[15:8]								ICR1H
(0x86)	ICR1[7:0]								ICR1L
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Chế độ Fast PWM của Timer/Counter 1



Chế độ Fast PWM hoạt động như sau:

Chúng ta sẽ đặt giá trị so sánh OCR1A và giá trị TOP ICR1 vào thanh ghi điều khiển, khi giá trị đếm TCNT1 của timer đếm đến giá trị của OCR1A thì đầu ra OC1A sẽ lên mức cao, khi TCNT1 đếm đến giá trị TOP thì sẽ bị tràn timer và reset TCNT1 về 0, đầu ra sẽ xuống mức thấp và lặp lại

Theo như datasheet của Atmega tần số xung pwm được tính như sau:

$$f_{OCnxPWM} = \frac{f_{clk\ I/O}}{N \cdot (1 + TOP)}$$

Foc là xung pwm điều khiển động cơ

Fclk là dao động thạch anh của Arduino

N là bộ chia tần của Timer có các giá trị 1, 8, 64, 256, 1024

TOP là giá trị tối đa của thanh ghi TCNT1

Đề tài sử dụng xung pwm có tần số là 1khz và độ rộng xung là 50% để điều khiển động cơ thay vào công thức trên ta tính được giá trị TOP là 15999 với bộ chia tần bằng 1

$$TOP = \frac{f_{clk}}{f_{OCn \times PWM \cdot N}} - 1 = \frac{16000000}{1000} - 1 = 15999$$

Để cấu hình Timer/Counter1 sử dụng chế độ Fast PWM trong arduino ta làm như sau:

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 14 or 15: Toggle OC1A on Compare Match, OC1B disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B disconnected.
1	0	Clear OC1A/OC1B on Compare Match, set OC1A/OC1B at BOTTOM (non-inverting mode)
1	1	Set OC1A/OC1B on Compare Match, clear OC1A/OC1B at BOTTOM (inverting mode)

Bật 2 bit COM1A1 và COM1B1 trong thanh ghi TCCR1A để sử dụng chế độ noninverting mode

Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation	TOP	Update of OCR1x at	TOV1 Flag Set on
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICR1	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCR1A	BOTTOM	TOP

Bật bit WGM11 trong thanh ghi TCCR1A và 2 bit WGM13, WGM12 trong thanh ghi TCCR1B để sử dụng chế độ Fast PWM với giá trị TOP tự định nghĩa bằng cách ghi vào thanh ghi ICR1

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$\text{clk}_{\text{IO}}/1$ (No prescaling)
0	1	0	$\text{clk}_{\text{IO}}/8$ (From prescaler)
0	1	1	$\text{clk}_{\text{IO}}/64$ (From prescaler)
1	0	0	$\text{clk}_{\text{IO}}/256$ (From prescaler)
1	0	1	$\text{clk}_{\text{IO}}/1024$ (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

Bật bit CS10 trong thanh ghi TCCR1A để sử dụng chế độ No prescaling đồng thời bắt đầu khởi chạy timer. Để điều khiển timer bật tắt theo ý muốn ta chỉ cần tắt và bật các bit CSx

Để xung PWM có độ rộng xung là 50% thì ta cần định nghĩa giá trị cho thanh ghi OCR1A

bằng cách lấy giá trị ICR1 chia 2 ta được 8000

Điều khiển số xung PWM để điều khiển động cơ trên Timer/Counter 1

Để điều khiển số xung PWM điều khiển động cơ ta cần bật bit OCIE1A trong thanh ghi TIMSK1 để kích hoạt ngắt khi giá trị TCNT1 đếm đến giá trị OCR1A. Khi sự kiện ngắt xảy ra ta sử dụng 1 biến đếm để lưu số lần giá trị TCNT1 đếm đến TOP khi biến đếm đạt đến số lượng yêu cầu thì tiến hành dừng timer bằng cách tắt các bit CS10 trong thanh ghi TCCR1A

Timer/Counter 2

Gồm có 2 kênh OC2A và OC2B tương ứng với các chân 11 và 3 trên Arduino. Gồm có các thanh ghi sau:

Thanh ghi TCNT2 là thanh ghi 8 bit lưu giá trị đếm của timer/counter cho phép đọc ghi trực tiếp

Thanh ghi TCCR2A và TCCR2B là 2 thanh ghi điều khiển của Timer/Counter 2

Bit (0xB0)	7	6	5	4	3	2	1	0	
	COM2A1	COM2A0	COM2B1	COM2B0	–	–	WGM21	WGM20	TCCR2A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

	7	6	5	4	3	2	1	0	
	FOC2A	FOC2B	–	–	WGM22	CS22	CS21	CS20	TCCR2B
	W	W	R	R	R/W	R/W	R/W	R/W	
	0	0	0	0	0	0	0	0	

Thanh ghi TIMSK2 (Timer/Counter2 Interrupt Mask Register) là thanh ghi lưu trữ các Interrupt Mask của Timer/Counter 2

	7	6	5	4	3	2	1	0	
	–	–	–	–	–	OCIE2B	OCIE2A	TOIE2	TIMSK2
	R	R	R	R	R	R/W	R/W	R/W	
	0	0	0	0	0	0	0	0	

Thanh ghi OCR2A (Output Compare Register A) là thanh ghi 8 bit lưu trữ giá trị so sánh

Bit (0xB3)	7	6	5	4	3	2	1	0	
	OCR2A[7:0]								OCR2A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Chế độ Fast PWM của Timer/Counter 2

Cách hoạt động Tương tự như chế độ Fast PWM của Timer/Counter 1. Ta cấu hình Timer/Counter 2 như sau:

COM2A1	COM2A0	Description
0	0	Normal port operation, OC2A disconnected.
0	1	WGM22 = 0: Normal Port Operation, OC0A Disconnected. WGM22 = 1: Toggle OC2A on Compare Match.
1	0	Clear OC2A on Compare Match, set OC2A at BOTTOM, (non-inverting mode).
1	1	Set OC2A on Compare Match, clear OC2A at BOTTOM, (inverting mode).

Bật bit COM2A1 để chọn chế độ non-inverting mode

Mode	WGM22	WGM21	WGM20	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on ⁽¹⁾⁽²⁾
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

Bật 2 bit WGM21 và WGM20 để chọn chế độ Fast PWM với giá trị top là 0xFF vì là timer 8 bit nên thanh ghi TCNT2 chỉ đếm được đến 255

Công thức tính tần số PWM của Timer/Counter 2 được tính như sau:

$$f_{OCnXPWM} = \frac{f_{clk I/O}}{N \cdot 256}$$

Với:

f_{oc} là tần số PWM

f_{cl} là tần số thạch anh của Arduino

N là bộ chia tần có các giá trị sau 1, 8, 32, 64, 128, 256 và 1024

Qua quá trình tính toán nhóm chọn được giá trị $N = 64$ cho ra tần số PWM là 976.5625 Hz gần đúng với 1kHz

CS22	CS21	CS20	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{T2S}/(No \text{ prescaling})$
0	1	0	$clk_{T2S}/8$ (From prescaler)
0	1	1	$clk_{T2S}/32$ (From prescaler)
1	0	0	$clk_{T2S}/64$ (From prescaler)

Bật bit CS22 trong thanh ghi TCCR2B để chia tần số thạch anh cho 64

Để độ rộng xung là 50% ta cần chọn giá trị cho OCR2A bằng cách lấy giá trị TOP chia 2 ta được giá trị xấp xỉ 128

Điều khiển số xung PWM để điều khiển động cơ trên Timer/Counter 2

Cách điều khiển tương tự như cách điều khiển của Timer/Counter 1

Cách tính số xung để điều khiển động cơ bước

Mô hình sử dụng pulley có đường kính là 9.8 mm để kéo con trượt, để tính được hành trình con trượt khi motor quay được 1 vòng ta cần biết chu vi của pulley, công thức chu vi đường tròn được tính như sau:

$$C = D * \pi \text{ Với:}$$

C là chu vi đường tròn

D là đường kính đường tròn

Áp dụng công thức trên ta tính được nếu motor quay được 1 vòng thì con trượt di chuyển được xấp xỉ 30.77 mm

Để tính được góc của motor để con trượt đi được khoảng cách y mm ta áp dụng tam suất

$$\begin{aligned} 360^\circ &= 30 \text{ mm} \\ x^\circ &= y \text{ mm} \\ \Rightarrow x^\circ &= \frac{y \cdot 360}{30} \end{aligned}$$

Với: y là khoảng cách được nhập vào bằng nút nhấn

Khi đã có góc cần quay để motor đi được khoảng cách y mm thì ta tính số xung cần thiết để cấp cho động cơ như sau:

$$\text{xung} = \text{góc cần quay} / N$$

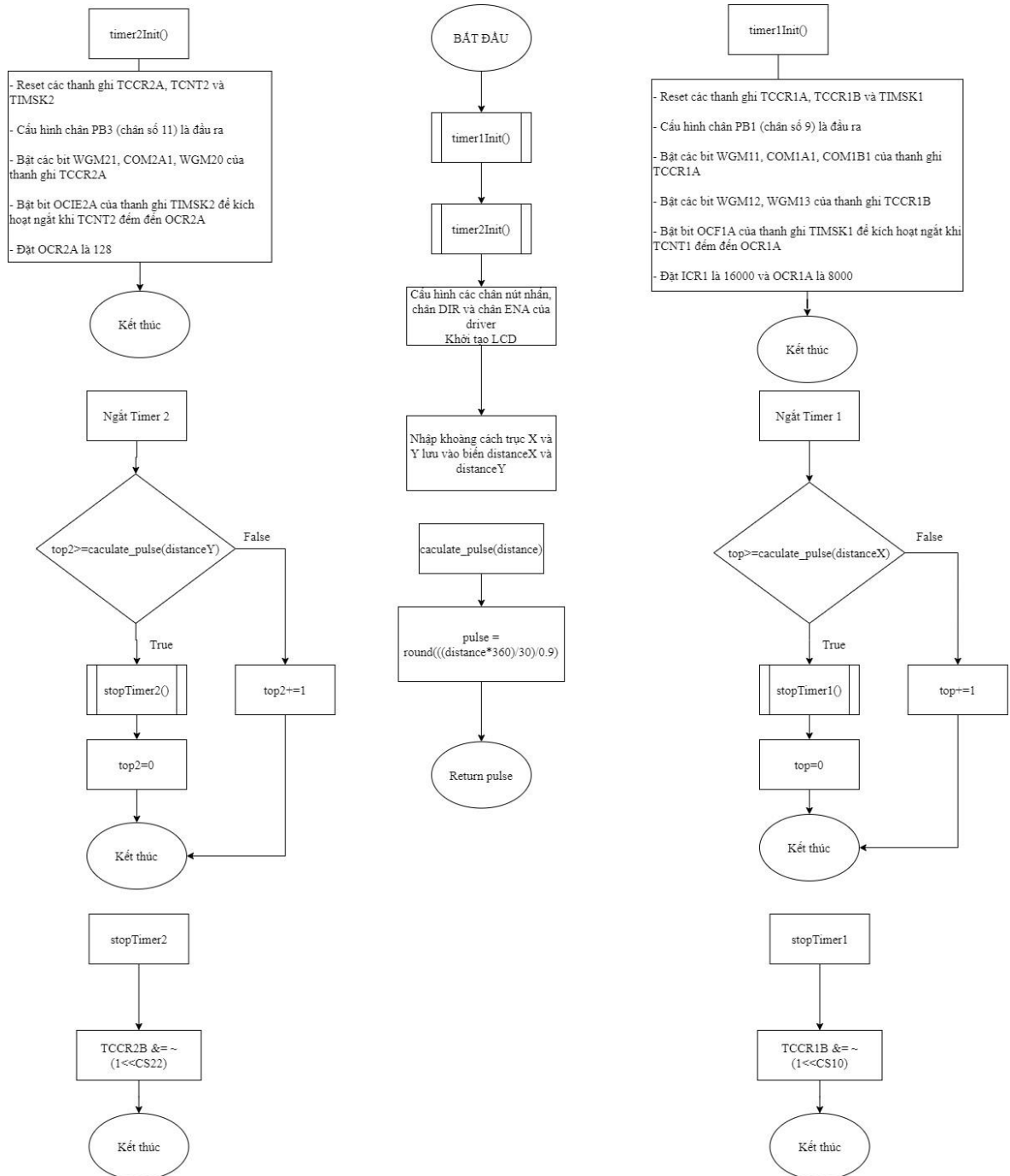
N là cấu hình vi bước của driver được cho bởi datasheet của driver ở bảng phía dưới

MODE0	MODE1	MODE2	Microstep Resolution
Low	Low	Low	Full step
High	Low	Low	Half step
Low	High	Low	1/4 step
High	High	Low	1/8 step
Low	Low	High	1/16 step
High	Low	High	1/32 step
Low	High	High	1/32 step
High	High	High	1/32 step

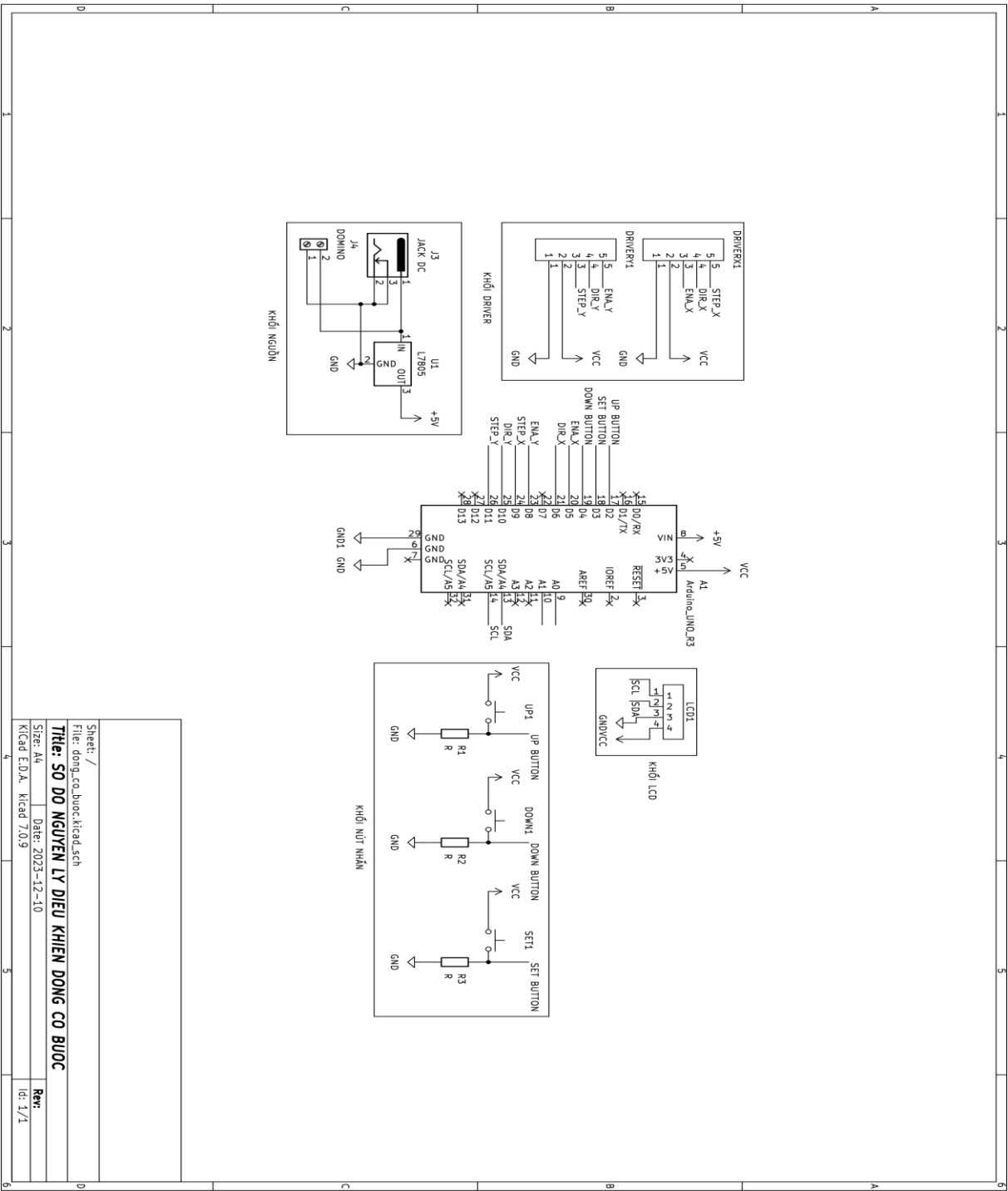
Nếu cấu hình là Full step thì 1 xung motor sẽ quay được 1.87 độ, half step sẽ quay được 1.87/2 độ và tương tự với các chế độ khác

Khi đã có số xung cần thiết ta đưa vào chương trình xử lý ngắt của 2 timer, nếu giá trị đếm cạnh lên của 2 timer lớn hơn hoặc bằng số xung cần thiết thì cho dừng timer không phát xung nữa

+Lưu đồ giải thuật



+ Sơ đồ nguyên lí



IV. Kết luận

Trong quá trình nghiên cứu và trình bày thông tin trong bài báo cáo này, chúng em đã đặt ra những quan điểm quan trọng về Lập trình và điều khiển động cơ bước . Các kết quả thu được không chỉ đưa ra những chi tiết mới và sâu sắc về đối tượng nghiên cứu, mà còn mở ra những triển vọng lớn về ứng dụng hoặc hướng phát triển tiếp theo. Chúng em hy vọng rằng bài báo cáo này không chỉ là một đóng góp cho lĩnh vực nghiên cứu của chúng em mà còn là nguồn cảm hứng cho các nhà nghiên cứu khác để tiếp tục khám phá và phát triển. Sự tiếp tục của công việc nghiên cứu này có thể đóng góp vào sự hiểu biết chung và mang lại những lợi ích thực tế cho xã hội.

Cuối cùng, chúng em xin bày tỏ lòng biết ơn sâu sắc đến tất cả những người đã đóng góp và hỗ trợ chúng em trong quá trình nghiên cứu này. Sự hỗ trợ này đã giúp chúng em vượt qua những thử thách và đạt được những thành công mà không thể nào đạt được một mình.

