# **LỜI NÓI ĐẦU**

Hiện nay, việc ấp nở gia cầm của người nông dân chăn nuôi nhỏ lẻ vẫn dựa vào gia cầm mẹ để ấp trứng, việc này gây ảnh hưởng đến năng suất con giống sinh ra.

Máy ấp trứng tự động là một giải pháp tốt cho người nông dân tăng năng suất con giống. Nhưng với lò ấp trứng lớn thì chi phí quá đắt so với thu nhập của người nông dân và kích thước của lò ấp công nghiệp quá lớn gây tốn diện tích.

Ngày này, điều khiển tự động đã trở thành một nhu cầu không thể thiếu được trong hệ thống công nghiệp. Một trong các yếu tố được điều khiểnnhiều trong hệ thống công nghiệp đó là nhiệt độ. Nhiệt độ được đo, điềukhiển theo nhu cầu sử dụng ví dụ như trong các hệ thống nhiệt của nồi hơi,các lò ấp, các lò sấy,… Các hệ thống đo và điều khiển nhiệt hiện nay xuất

hiện nhiều trên thị trường với nhiều phương pháp đo và điều khiển khác nhau. Mỗi phương pháp đều có ưu và nhược điểm riêng.

Trong quá trình học môn *Cảm biến trong điều khiển tự động,* chúng em đã nhận được đề tài “*Xây dựng lò ấp trứng điều khiển nhiệt độ bằng thuật toán PID”*. Để hoàn thành đề tài này, ngoài sự nỗ lực của bản thân chúng em ,chúng em xin chân thành gửi lời cảm ơn sâu sắc tới các thầy cô Khoa Công Nghệ Tự Động Hóa trường Đại học Công nghệ thông tin và Truyền thông Thái Nguyên, đặc biệt là thầy **Th.S Đặng Văn Ngọc** đã tận tình hướng dẫn, giúp đỡ và tạo điều kiện tốt nhất cho chúng em kể từ khi nhận đề tài tới khi hoàn thành đề tài này.

Em xin chân thành cảm ơn!

# **MỤC LỤC**

[**LỜI NÓI ĐẦU** 1](#_Toc58417774)

[**MỤC LỤC** 2](#_Toc58417775)

[**DANH MỤC HÌNH ẢNH** 3](#_Toc58417776)

[**CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI** 5](#_Toc58417777)

[**1.1** **Giới thiệu chung về máy ấp trứng** 5](#_Toc58417778)

[**1.2** **Mục đích nghiên cứu** 6](#_Toc58417779)

[**CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG** 8](#_Toc58417780)

[**2.1** **Đặt vấn đề** 8](#_Toc58417783)

[**2.2** **Giải quyết vấn đề** 8](#_Toc58417784)

[**2.3** **Giới thiệu một số phương pháp và lựa chọn phương pháp điều chỉnh nhiệt độ lò ấp trứng** 8](#_Toc58417785)

[*2.3.1* *Mô tả toán học của lò nhiệt trong phòng thí nghiệm* 8](#_Toc58417786)

[*2.3.2* *Phương pháp điều khiến on — off* 9](#_Toc58417787)

[*2.3.3* *Phương pháp điều khiến PID* 11](#_Toc58417788)

[**2.4** **Xây dựng cơ sở lý thuyết cho bộ điều khiển PID số** 13](#_Toc58417789)

[*2.4.1* *Cấu trúc chung của một hệ điều khiển tự động* 13](#_Toc58417790)

[**2.5** **Mô tả toán học hệ rời rạc** 15](#_Toc58417791)

[*2.5.1* *Đặc điểm lấy mẫu* 16](#_Toc58417792)

[*2.5.2* *Khâu giữ dữ liệu (ZOH)* 17](#_Toc58417793)

[*2.5.3* *Phép biến đổi Z* 18](#_Toc58417794)

[**2.6** **Mô tả hệ thống rời rạc bằng hàm truyền** 20](#_Toc58417795)

[*2.6.1* *Định nghĩa* 20](#_Toc58417796)

[*2.6.2* *Cách tìm hàm truyền* 20](#_Toc58417797)

[*2.6.3* *Hàm truyền hệ rời rạc* 20](#_Toc58417798)

[**2.7** **Phân tích thiết kế hệ thống điều khiển rời rạc** 21](#_Toc58417799)

[*2.7.1* *Điều kiện ổn định của hệ rời rạc* 21](#_Toc58417800)

[**2.8** **Chất lượng hệ thống rời rạc** 22](#_Toc58417801)

[*2.8.1* *Đáp ứng quá độ* 22](#_Toc58417802)

[*2.8.2* *Độ vọt lố* 22](#_Toc58417803)

[*2.8.3* *Sai số xác lập* 23](#_Toc58417804)

[**2.9** **Hàm truyền của các khâu hiệu chỉnh rời rạc** 24](#_Toc58417805)

[*2.9.1* *Khâu tỉ lệ* 24](#_Toc58417806)

[*2.9.2* *Khâu vi phân* 24](#_Toc58417807)

[*2.9.3* *Khâu tích phân* 25](#_Toc58417808)

[*2.9.4* *Bộ điều khiển PI, PD, PID rời rạc* 25](#_Toc58417809)

[*2.9.5* *Tính toán xây dựng mô hình toán học cho bộ điều khiển PID* 25](#_Toc58417810)

[**2.10** **Thiết lập mô hình toán học lò ấp trứng** 26](#_Toc58417811)

[*2.10.1* *Mô hình toán học của hệ thống* 26](#_Toc58417812)

[*2.10.2* *Hàm truyền lò ấp trứng* 26](#_Toc58417813)

[*2.10.3* *Hàm chuyển đổi PID rời rạc* 28](#_Toc58417814)

[**CHƯƠNG 3: THIẾT KẾ THI CÔNG MẠCH ĐIỀU KHIỂN VÀ MẠCH CÔNG SUẤT** 33](#_Toc58417815)

[**3.1** **Xây dựng phần cứng** 33](#_Toc58417817)

[*3.1.1* *Sơ đồ chức năng* 33](#_Toc58417818)

[*3.1.2* *Sơ đồ nguyên lý* 34](#_Toc58417819)

[*3.1.3* *Mạch phát hiện điểm không* 34](#_Toc58417820)

[*3.1.4* *Mạch động lực* 35](#_Toc58417821)

[*3.1.5* *Sơ đồ nguyên lý toàn mạch* 36](#_Toc58417822)

[**3.2** **Giới thiệu một số công cụ sử dụng** 36](#_Toc58417823)

[*3.2.1* *Cảm biến nhiệt độ LM35* 36](#_Toc58417824)

[*3.2.2* *Atmega328p* 38](#_Toc58417825)

[*3.2.3* *Arduino IDE* 40](#_Toc58417826)

[*3.2.4* *Màn hình LCD 16x2* 41](#_Toc58417827)

[*3.2.4.Module I2C* 42](#_Toc58417828)

[*3.2.5* *Opto MOC3021-DIP 6* 43](#_Toc58417829)

[*3.2.6* *Triac BTA12* 44](#_Toc58417830)

[*3.2.7* *Đồng hồ thời gian thực DS1307* 45](#_Toc58417831)

[**3.3** **Xây dựng phần mềm** 46](#_Toc58417832)

[*3.3.1* *Lưu đồ thuật toán điều khiển* 46](#_Toc58417833)

[*3.3.2* *Lưu đồ thuật toán PID* 48](#_Toc58417834)

[**3.4** **Kết quả mô hình thực tế** 49](#_Toc58417835)

[*3.4.1* *Sản phẩm thực tế* 49](#_Toc58417836)

[*3.4.2* *Hướng dẫn sử dụng sản phẩm* 50](#_Toc58417837)

# **DANH MỤC HÌNH ẢNH**

[Hình 1. 1 Máy ấp trứng công nghiệp 7](#_Toc58417838)

[Hình 2. 1 Đặc điểm của hoạt động on – off 11](#_Toc58417848)

[Hình 2. 2 Đặc điểm của Hunting 11](#_Toc58417849)

[Hình 2. 3 Hệ thống điều khiển vòng kín với bộ điều khiến PID 12](#_Toc58417850)

[Hình 2. 4 Hệ thống điều khiển vòng kín với bộ điều khiến PID 15](#_Toc58417851)

[Hình 2. 5 Sơ đồ khối hệ thống điều khiển số 17](#_Toc58417852)

[Hình 2. 6 Khâu giữ bậc 0 (ZOH) 18](#_Toc58417853)

[Hình 2. 7 Sơ đồ mô tả hệ thống 27](#_Toc58417854)

[Hình 2. 8 Thí nhiệm xác định hàm truyền lò ấp trứng 28](#_Toc58417855)

[*Hình 2. 9 Đặc tính của lò ấp trứng* 28](#_Toc58417856)

[Hình 3. 1 Sơ đồ khối hệ thống 35](#_Toc58417868)

[Hình 3. 2 Sơ đồ nguyên lý khối nguồn 36](#_Toc58417869)

[Hình 3. 3 Mạch phát hiện điểm 0 36](#_Toc58417870)

[Hình 3. 4 Mạch động lực 37](#_Toc58417871)

[Hình 3. 5 Sơ đồ nguyên lý toàn mạch 38](#_Toc58417872)

[Hình 3. 6 Atmega328p 41](#_Toc58417873)

[Hình 3. 7 Sơ đồ chân Atmega328p 42](#_Toc58417874)

[Hình 3. 8 Giao diện phần mềm Arduino 43](#_Toc58417875)

[Hình 3. 9 Màn hình LCD 16x2 43](#_Toc58417876)

[Hình 3. 10 Sơ đồ chân Opto MOC3201-DIP 6 46](#_Toc58417877)

[Hình 3. 11 Sơ đồ chân Triacs BTA12 47](#_Toc58417878)

[Hình 3. 12 Module DS1307 48](#_Toc58417879)

[Hình 3. 13 Lưu đồ thuật toán điều khiển 49](#_Toc58417880)

[Hình 3. 14 Lưu đồ thuật toán PID 50](#_Toc58417881)

[Hình 3. 15 Sản phẩm lò ấp trứng 51](#_Toc58417882)

[Hình 3. 16 Mạch điều khiển lò ấp 51](#_Toc58417883)

[Hình 3. 17 Bộ phận sinh nhiệt cho lò ấp trứng 52](#_Toc58417884)

[Hình 3. 18 Các nút nhấn trên lò ấp 52](#_Toc58417885)

# **CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI**

## **Giới thiệu chung về máy ấp trứng**

Máy ấp trứng là thiết bị ấp trứng gia cầm chuyên nghiệp và hiện đại, áp dụng kỹ thuật điện tử dùng nhiệt độ để ấp trứng thay thế cho gà mẹ ấp trứng truyền thống. Máy ấp trứng cho kết quả tỷ lệ thành công cao vì nó là hoàn toàn tự động. Các thao tác bạn thực hiện với máy ấp trứng đơn giản, dễ dàng thao tác, tất cả những gì bạn cần làm là cho trứng vào trong máy và chọn các thông số là máy có thể tiến hành áp trứng.

Dựa trên các yếu tố của việc ấp trứng tự nhiên do gia cầm thực hiện. Máy ấp trứng đưa ra những thông số kỹ thuật tương tự. Với ưu thế hơn hẳn về sản lượng ấp trứng mỗi mẻ trứng.

Các loại trứng gia cầm đều có yêu cầu khác nhau về nhiệt độ , độ ẩm, thời gian ấp nở . Về cơ bản máy ấp trứng hoạt động gồm 4 khâu : nhiệt độ, độ ẩm, đảo trứng, thông gió các thông số kỹ thuật này đều được điều khiển bằng các mạch bán dẫn và vi điện tử. Để có được con giống khoẻ mạnh, tỷ lệ ấp nở cao máy ấp trứng phải đáp ứng được 4 yếu tố trên .

Trong đó khâu nhiệt độ đóng vai trò quan trọng nhất , quả trứng ấp không đủ nhiệt thì phôi sẽ không phát triển . Để giữ nhiệt , các vỏ máy được thiết kế dày và có chức năng cách ly tốt , góp phần lưu nhiệt khi mất điện . Trong máy có các hệ thống dây điện trở , có chức năng sinh nhiệt , mỗi dãy có công suất tùy thuộc vào thể tích của lồng ấp . Để đóng ngắt mạch điện và dây điện trở sinh nhiệt , có thể sử dụng rơle điện tử không tiếp điểm , dùng tri - ắc Công suất lớn , bộ đóng ngắt hoạt động với độ tin cậy cao . Khâu đảo trứng là khâu thứ hai trong quá trình ấp . Thông thường trứng được đảo vài giờ một lần , một lần kéo dài khoảng 10 phút . Việc đảo trứng thực hiện chậm vị tránh hiện tượng va đập làm hỏng trứng . Dàn đảo sẽ đảo với một góc không quá 60 độ hoặc thấp hơn tùy vào thiết kế của gia đẻ trứng . Một quả trứng bình thường chứa 6,5 % đến 6,6 % lượng nước . Trong quá trình tiếp nhiệt độ để phát triển thành con giống , lượng nước sẽ bị bay hơi dần . Mày ắp trừng phải có hệ thống cung cấp độ ẩm tự động và điều chỉnh được tuỳ ý . Thống thường máy ấp Có giàn phun nước tự động để giữ cho độ ẩm không thay đổi tùy thuộc vào từng giai đoạn của trứng . Thông gió là phần không thể thiếu trong quá trình ấp . Các quạt thông gió phải gắn với cửa chớp mở tự động mỗi khi quạt hoạt động . Việc gắn với cửa chớp là để đảm bảo việc cách ly với môi trường bên ngoài, đảm bảo việc giữ nhiệt. Việc thông gió có thể kết hợp với việc giảm nhiệt cho lò.



Hình 1. 1 Máy ấp trứng công nghiệp

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Loại trứng* | *Thời gian ấp (ngày)* | *Nhiệt độ (\*C)* | *Độ ẩm (%)* | *Ngày ngừng đảo trứng* | *Độ ẩm 3 ngày cuối (%)* |
| *Gà* | 21 | 37-38 | 55-65 | Ngày 18(2 giờ/1 lần đảo) | 60-70 |
| *Vịt* | 28 | 37-38 | 55-65 | Ngày 25(2 giờ/1 lần đảo) | 60-70 |
| *Chim cút* | 17 | 36-37 | 55-65 | Ngày 15(2 giờ/1 lần đảo) | 60-70 |

*Bảng 1. 1 Các giá trị nhiệt độ ,độ ẩm , thời gian trong quá trình ấp trứng*

## **Mục đích nghiên cứu**

Khâu kiểm soát nhiệt độ trong quá trình ấp trứng là quan trọng nhất trong 4 khâu của một hệ thống ấp trứng quyết sản lượng và chất lượng của con giống . Xây dựng và thiết kế mạch điều khiển nhiệt độ một cách tự động để có thể nâng cao hiệu quả ấp trứng , đảm bảo chất lượng con giống giảm bớt sức lao con người . Kiểm soát tốt giá trị nhiệt độ đảm bảo độ an toàn cho người điều khiển tăng độ bền bỉ của thiết bị trong suốt quá trình sản xuất.

# **CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG**



## **Đặt vấn đề**

Như chúng ta đã biết điều khiển tự động đã trở thành nhu cầu không thể thiếu trong cuộc sống hiện đại . Một trong yếu tố được điều khiển tự động nhiều là nhiệt độ.Để làm được điều đó chúng ta cần phải sử dụng các thiết bị đo và điều khiển tự động ví dụ như cảm biến , rơle , ADC ... Một trong những ứng dụng quan trọng và phổ biến nhất của điều khiển tự động nhiệt độ là ứng dụng để điều khiển nhiệt độ trong lồ ấp trứng công nghiệp . Với ưu điểm của lò ấp trứng công nghiệp là tỉ lệ ấp thành công lớn , cho hiệu quả kinh tế cao . Vì vậy hệ thống lò ấp trứng công nghiệp ứng dụng rộng rãi trong phát triển kinh tế địa phương .

## **Giải quyết vấn đề**

Để giải quyết vấn đề trên ta sử dụng phương pháp hiện đại dùng cảm biến và vi xử lý, bóng đèn cung cấp nhiệt. Điều khiển và ổn định nhiệt độ bằng vi xử lý .

*Ưu điểm*: Đo và điều khiển giá trị nhiệt độ ,độ ẩm thời gian một cách chính xác tỷ lệ trứng nở cao đem lại hiệu quả về kinh tế, công suất ấp trứng mỗi mẻ lớn.

*Nhược điểm* : Do hệ thống điều khiển tự động dựa trên nền tảng vi điều khiển nên đòi hỏi kiến thức về điều khiển tự động , vi sử lý, cảm biến , lập trình.

## **Giới thiệu một số phương pháp và lựa chọn phương pháp điều chỉnh nhiệt độ lò ấp trứng**

### *Mô tả toán học của lò nhiệt trong phòng thí nghiệm*

Lò nhiệt có đầu vào là điện áp cung cấp cho dây đốt (hay là công suất cung cấp) và ngõ ra là nhiệt độ của sản phẩm cần nung hay nhiệt độ vùng sử dụng. Thực tế lượng nhiệt này ngoài việc đốt nóng để tăng nhiệt độ bên trong mà nó còn cung cấp nhệt lượng bên ngoài nên thực thế phương trình cân bằng năng lượng này rất khó thiết lập. Một cách gần đúng ta có thể xem môi trường nung là đồng chất, đẳng nhiệt ta có hàm truyền của lò là.

P

t0C

*Trong đó:*

P: công suất cung cấp dưới dạng điện năng.

K: hệ số tỷ lệ cho biết quan hệ giữa ngõ vào và ngõ ra ở trạng thái xác lập.

T: hằng số thời gian thể hiện quán tính của hệ thống.

Theo thực nghiệm Ziegler-Nichols để xuất ra hệ thống là khâu biểu diễn ở bậc cao do tính chất có trễ của lò nhiệt, xấp xỉ về khâu quán tính bậc nhất có trễ ta được:

*Trong đó:*

L: Thời gian trễ của lò nhiệt.

Khai triển Taylor gần đúng e-Ls ta được hàm truyền sẽ là hệ thống tuyến tính bậc 2:

### *Phương pháp điều khiến on — off*

Điều khiển on- off là lặp lại trạng thái on- off của hệ thống điều khiển theo điểm đặt. Ví dụ trong hình, relay ngõ ra là on khi nhiệt độ trong lò dưới điểm đặt, và off khi nhiệt độ đến điểm đặt.

*a, Mô tả hoạt động on-off*

Với cầu hình của hệ thống điều khiển được trình bày như ở trên, relay ngõ ra on, cấp điện tới sợi nung khi giá trị nhiệt độ hiện tại trong lò dưới điểm đặt. Relay ngõ ra off khi nhiệt độ lên cao hơn điểm đặt. Nhờ phương pháp điều khiến nhiệt độ mà nhiệt độ được đặt ở giá trị nào đó bằng cách bật on và off nguồn cho sợi nung được gọi là điều khiển on-off. Hoạt động này cũng được gọi là điều khiến hai vị trí vì hai biến đặt cũng liên quan tới điểm đặt.

T1

T2

set point T

D

ON

OF

→

→

Hình 2. 1 Đặc điểm của hoạt động on – off

*b, Điều chỉnh độ nhạy*

Nếu relay ngõ ra được bật on hoặc off ở một điểm đặt chattering của ngõ ra có thể xảy ra làm hệ thông điều khiển có thể bị ảnh hưởng nhiễu. Vì lý đo này mà từ trễ giữa on và off thường xảy ra ở ngõ ra như hình 1.1. Từ trễ này được gọi là hiệu chỉnh độ nhạy. Điều chỉnh độ nhạy cao đi hỏi cần phải tránh tần số hoạt động on-off.

*c, Hunting*

Khi điểm đặt được điều khiển bằng hoạt động on-off, biến đặt thay đổi tuần hoàn như trong hình 1.2. Sự thay đổi tuần hoàn này được gọi là hunting.

0C

t(s)

Hunting

Hình 2. Đặc điểm của Hunting

*Kết quả tốt nhất của hoạt động on-off đạt được nếu như biên độ hunting là nhỏ nhất.*

*d, Hệ thống thích hợp cho điều khiển on-off*

Điều khiến on-off tốt nhất cho hệ thống điều khiến khi nhiệt độ tăng lên chậm và sai phân G giữa cân bằng nhiệt khi ngõ ra là on và khi ngõ ra là off nhỏ. Ví dụ, G nhỏ duy trì đáp ứng nhiệt nhanh và hunting được bật tắt bằng hình thức on-off được trình bảy như trong hình. Trong hình này dùng đèn ở ngõ ra.

Nhiệt độ trên tới giá trị tới hạn thấp của ngõ ra đèn được điều khiến bằng hai sợi nung với tổng công suất là 600W. Trong lân cận điểm đặt. nhiệt độ điều khiển mỗi sợi nung là 300W.

### *Phương pháp điều khiến PID*

PID là bộ điều khiến tỷ lệ - tích phân - vi phân (Propotional – Integral - Derivative). Bộ điều khiển PID được sử dụng rộng rãi để điều khiển đổi tượng SISO theo nguyên tắc sai lệch:

U(t)

PID

Đối tượng

y(t)

e(t)

-

+

Hình 2. Hệ thống điều khiển vòng kín với bộ điều khiến PID

Nếu e(t) càng lớn thì thông qua thành phần tỷ lệ làm cho x(t) càng lớn (vai trò của khâu P).

Nếu e(t) chưa bằng 0 thì thông qua thành phần tích phân. PID vẫn tạo tín hiệu điều chỉnh (vai trò của khâu I).

Nếu e(t) thay đổi lớn thì thông qua thành phân ví phân, phản ứng tích hợp x (t) càng nhanh (vai trò của khâu D).

*a, Khâu tỷ lệ D*

Khâu P tạo ra tín hiệu điều khiển tỉ lệ với giá trị của sai lệch. Việc này được thực hiện bằng cách nhân sai lệch e với hằng số K gọi là hằng số tỉ lệ.

Khâu P được tính dựa trên công thức:

Trong đó:

Pout: giá trị ngõ ra

Kp: hằng số tỷ lệ

e: sai lệch, t: thời gian tưc thời (hiện tại)

*Sơ đồ khối của khâu P:*

e(t)

Kp

u(t)

*b, Khâu tích phân I*

Khâu I cộng thêm tổng các sai số trước đó vào giá trị điều khiển. Việc tính tổng các sai số được thực hiện liên tục cho đến khi giá trị đạt được bằng với giá trị đặt, và kết quả là khi hệ cân bằng thì sai số bằng 0.

Khâu I được tính theo công thức:

*Trong đó:*

I: giá trị ngõ ra khâu I

K: hệ số tích phân

e: sai số

τ: một biến phân tích trung gian

t: thời gian tưc thời (hiện tại)

*Sơ đồ khối khâu I:*

e(t)

Ki

u(t)

Khâu I thường đi kèm với khâu P. hợp thành bộ điều khiển PI. Nếu chỉ sử dụng khâu I thì đáp ứng của hệ thống sẽ chậm và thường bị dao động.

*c, Khâu vi phân D*

Khâu D cộng thêm tốc độ thay đối sai số vào giá trị điều khiển ở ngõ ra. Nếu sai số thay đổi nhanh thì sẽ tạo ra thành phần cộng thêm vào giá trị điều khiến. Điều này cải thiện đáp ứng của hệ thống, giúp trạng thái của hệ thống thay đổi nhanh chóng và đạt được giá trị mong muốn.

*Khâu D được tính theo công thức:*

*Trong đó:*

Dout: ngõ ra khâu

KD: hệ số vi phân

e: sai số

t: thời gian tưc thời (hiện tại)

e(t)

KD

u(t)

*d, Bộ điều khiển PID cho lò nhiệt*

Tổng hợp 3 khâu cho P, I, D cho bộ điều khiển PID. Bộ điều khiển PID được mô tả như hàm truyền sau:

Ta có lò nhiệt có hàm truyền đạt:

c(t)

Đối tượng

r(t)

## **Xây dựng cơ sở lý thuyết cho bộ điều khiển PID số**

### *Cấu trúc chung của một hệ điều khiển tự động*

Hệ thống tự động là một hệ thống được xây dựng từ ba bộ phận chủ yếu:

* Thiết bị điều khiển.
* Đối tượng điều khiển.
* Thiết bị đo lường.

Đó là một hệ thống có phản hồi hay còn gọi là hệ thống có liên hệ ngược. Sơ đồ hệ thống điều khiển tự động được giới thiệu trên hình như sau:

x(t)

y(t)

e(t)

-

Đo lường

PID

Đối tượng

u(t)

Fn(t)

z(t)

+

Hình 2. 4 Hệ thống điều khiển vòng kín với bộ điều khiến PID

*Trong đó:*

x(t): Tín hiệu đầu vào của hệ thống. Còn gọi là giá trị đặt.

y(t): Tín hiệu ra của hệ thống. Còn gọi là lượng được điều chỉnh.

u(t): Tín hiệu điều khiển tác động lên đối tượng.

e(t): Sai lệch điều khiển.

z(t): tín hiệu phản hồi.

Fn(t): tín hiệu nhiễu tác động lên đối tượng.

Thiết bị điều khiển là một bộ phẩn vô cùng quan trọng nó tạo ra tín hiệu điều khiển u(t), tác động vào đối tượng điều khiển. Với mong muốn điều khiển được lượng ra của đối tượng y(t) thoả mãn yêu cầu, đáp ứng được chỉ tiêu kinh tế và kỹ thuật từ hình vẽ 2.1 ta thấy đầu vào của bộ điều khiển là tín hiệu sai lệch điều khiển e(t) = x(t) – z(t) đầu ra của bộ điều khiển là u(t) = f[e(t)], ở bộ điều khiển đơn giản thì có một đầu vào và một đầu ra, trong trường hợp phức tạp bộ điều khiển có nhiều đầu vào, ra.

Bộ điều khiển có nhiệm vụ duy trì sự ổn định cho hệ thống nghĩa là nó điều chỉnh quá trình hệ thống quanh một giá trị xác lập.

## **Mô tả toán học hệ rời rạc**

Chương này đề cập đến một loại hệ thống điều khiển có hồi tiếp, trong đó tín hiệu tại một hay nhiều điểm là một chuỗi xung, không phải là hàm liên tục theo thời gian. Tùy thuộc vào phương pháp lượng tử hóa tín hiệu mà ta có các loại hệ thống xử lý tín hiệu khác nhau. Phương pháp lượng tử hóa theo thời gian cho tín hiệu có biên liên tục thời gian rời rạc. Hệ thống xử lý loại tín hiệu này được gọi là hệ thống rời rạc. Nếu phép lượng tử hóa được tiến hành theo thời gian và cả theo biên độ thì kết quả nhận được là tín hiệu số. Hệ thống xử lý tín hiệu số gọi là hệ thống rời rạc.

Trong hệ thống rời rạc và hệ thống số, thống số điều khiển - biên độ của tín hiệu chỉ xuất hiện tại các thời điểm rời rạc cách đều nhau đúng bằng một chu kỳ lấy mẫu tín hiệu. Vì có thời gian trễ tất yếu do lấy mẫu, việc ổn định hệ thống trở nên phức tạp hơn so với hệ liên tục, do đó đòi hỏi những kỹ thuật phân tích và thiết kế đặc biệt. Sự phát triển mạnh mẽ của kỹ thuật số, kỹ thuật vi xứ lý và kỹ thuật máy tính làm cho ngày càng có nhiều hệ thống điểu khiển số được sử dụng để điều khiển các đối tượng. Hệ thống điều khiển số có nhiều ưu điểm so với hệ thống điều khiển liên tục như uyển chuyển, linh hoạt, dễ dàng đổi thuật toán điều khiển, dễ dàng áp dụng các thuật toán điều khiển phức tạp bằng cách lập trình. Máy tính số còn có thể điểu khiến nhiễu đối tượng cùng một lúc. Ngoài ra, giá máy tính ngày càng hạ trong khi đó tốc độ xử lý, độ tin cậy ngày càng tăng lên cũng góp phần làm cho việc sử dụng các hệ thống điểu khiển số trở nên phổ biến. Hiện nay các hệ thống điều khiển số được sử dụng rất rộng rãi, từ các bộ điều khiển đơn giản như điều khiển nhiệt độ, điều khiển động cơ DC, AC,... đến các hệ thống điều khiển phức tạp như điểu khiển robot, máy bay, tàu vũ trụ, các hệ thống điều khiển quá trình công nghệ hóa học và các hệ thống tự động cho những ứng dụng khác nhau.

Máy tính số

D/A

Đối tượng

Cảm biến

A/D

Cht(kT)

r(kT)

u(kT)

uR(t)

c(t)

Hình 2. 5 Sơ đồ khối hệ thống điều khiển số

### *Đặc điểm lấy mẫu*

Lấy mẫu dữ liệu là biến đổi tín hiệu liên tục theo thời gian thành tín hiệu rời rạc theo thời gian.

x(t)

x\*(t)

T

x(t)

0

t

s(t)

0

t

T 2T 3T 4T 5T 6T 7T 8T ...

x\*(t)

0

t

T 2T 3T 4T 5T 6T 7T 8T ...

1

Xét bộ lấy mẫu có đầu vào là tín hiệu liên tục x(t) và đầu ra thì tín hiệu rời rạc x\*(t) H(2.4). Quá trình lấy mẫu có thể mô tả bởi biểu thức toán học sau:

Trong đó s(t) là chuỗi xung dirac:

### *Khâu giữ dữ liệu (ZOH)*

Khâu giữ dữ liệu là khâu chuyển tín hiệu rời rạc theo thời gian thành tín hiệu liên tục theo thời gian.

Khâu giữ dữ liệu có nhiều dạng khác nhau, dơn giản nhất và được sử dụng nhiều nhất trong các hệ thống điều khiển rời rạc là khâu giữ bậc 0 (Zero-Order Hold-ZOH) (H4)

x\*(t)

0

t

T 2T 3T 4T 5T 6T 7T 8T ...

xR(t)

0

t

T 2T 3T 4T 5T 6T 7T 8T ...

1

0

r(t)

t

1

0

r(t)

t

T

Hình 2. 6 Khâu giữ bậc 0 (ZOH)

Ta tìm hàm truyền của khâu ZOH. Để ý rằng nếu tín hiệu vào của khâu ZOH là xung dirac thì tín hiệu ra là xung vuông có độ rộng bằng T (H 2.5). Ta có:

R(s) = 1 (vì r(t) là hàm dirac).

Theo định nghĩa:

Do đó:

Biểu thức (1) chính là hàm truyền của khâu giữ bậc 0. Trong các hệ thống điều khiển thực tế, nếu có thể bỏ qua được sai số lượng tử hóa thì các khâu chuyển đổi D/A chính là khâu giữ bậc 0 (ZOH).

Nhận xét:Bằng cách sử dụng phép biến đổi Laplace ta có thể mô tả quá trình lấy mẫu và giữ dữ liệu bằng các biểu thức toán học (H.2.5) và (1). Tuy nhiên các biểu thức toán học này lại chứa hàm ex nên nếu ta sử dụng để mô tả hệ rời rạc thì khi phân tích, thiết kế hệ thống sẽ gặp nhiều khó khăn. Ta cần mô tả toán học khác giúp khảo sát hệ thống rời rạc dễ dàng hơn, nhờ phép biến đổi Z trình bày dưới đây.

### *Phép biến đổi Z*

*a, Biến đổi Laplace*

Khái niệm và tính chất của phép biến đổi Laplace: khi sử dụng các phép biến đổi tín hiệu hệ thống từ miền thời gian sang miền khác để thuận tiện trong việc xử lý tín hiệu. Như trong hệ thống liên tục người ta hay sử dụng phép biến đổi Laplace để biến đổi từ miền thời gian sang miền tần số phức. Các phương trình vi tích phân sẽ chuyển đổi thành các phương trình đại số thông thường. Trong các hệ thống rời rạc người ta hay sử dụng phép biến đổi Z để chuyển tín hiệu từ miền thời gian sang miền tần số phức. Trong thực tế người ta còn sử dụng các phép biến đổi khác để xử lý tín hiệu như giải tương quan, mã hoá có hiệu quả, chống nhiễu,….

Thực hiện các phép biến đổi có công cụ toán học như máy tính số, công cụ phổ biến và hiệu quả là phần mềm Matlab hay thực hiện biến đổi bằng tay.

*Định nghĩa phép biến đổi Z*

Cho x(k) là chỗi tín hiệu rời rạc, biến đổi Z của x(k) là:

Trong đó:

z = (s là biến đổi Laplace).

X(z): biến đổi Z của chuỗi x(k).

Ký hiệu: x(k)X(z)

Nếu x(k) = 0, k < 0 thì biểu thức định nghĩa trở thành:

Miền hội tụ (Region Of Convergence - ROC).

ROC là tập hơp tất cả các giá trị z sao cho X(z) hữu hạn.

*b, Ý nghĩa phép biến đổi Z*

Giả sử x(t) là tín hiệu liên tục trong miền thời gian, lấy mẫu x(t) với chu kỳ lấy mẫu T ta được chuỗi rời rạc x(k) = x(kT).

Biểu thức lấy mẫu tín hiệu x(t).

Biểu thức biến đổi Z chuỗi x(k) = x(kT).

Do z = nên vế phải của hai biểu thức lấy mẫu và biến đổi Z là như nhau, do đó bản chất của việc biến đổi Z một tín hiệu chính là rời rạc hóa tín hiệu đó.

## **Mô tả hệ thống rời rạc bằng hàm truyền**

### *Định nghĩa*

Hàm truyền của hệ thống là tỉ số giữa biến đổi Laplace của tín hiệu ra và biến đổi Laplace của tín hiệu vào khi điều kiện đầu bằng 0.

Chú ý: Mặc dù hàm truyền được định nghĩa là tỉ số giữa biến đổi Laplace của tín hiệu ra và biến đổi Laplace của tín hiệu vào nhưng hàm truyền không phụ thuộc vào tín hiệu ra và tín hiệu vào mà chỉ phụ thuộc vào cấu trúc và thông số của hệ thống. Do đó có thể dùng hàm truyền để mô tả hệ thống.

### *Cách tìm hàm truyền*

*Bước 1*: Thành lập phương trình vi phân mô tả quan hệ vào – ra của phần tử bằng cách:

* Áp dụng các định luật Kirchoff, quan hệ dòng – áp trên điện trở, tụ điện, cuộn cảm,… đối với các phần tử điện.
* Áp dụng các định luật Newton, quan hệ giữa lực ma sát và vận tốc, quan hệ giữa lực và biến dạng của lò xo,… đối với các phần tử cơ khí.
* Áp dụng các định luật truyền nhiệt, định luật bảo toàn năng lượng,… đối với các phần tử nhiệt.

*Bước 2*: Biến đổi Laplace hai vế phương trình vi phân vừa thành lập ở bước 1, ta được hàm truyền cần tìm.

*Chú ý:* Đối với các mạch điện có thể tìm hàm truyền theo phương pháp tổng trở phức

### *Hàm truyền hệ rời rạc*

Hệ rời rạc

r(k)

c(k)

Quan hệ giữa tín hiệu vào và tín hiệu ra của hệ thống rời rạc được mô tả bằng phương trình sai phân:

Trong đó n, n gọi là bậc của hệ thống rời rạc Biến đổi z hai vế phương trình trên ta được:

⬄

Hai cách biểu diễn trên hoàn toàn tương đương nhau, trong thực tế hàm truyền dạng thứ hai được sử dụng nhiều hơn.

## **Phân tích thiết kế hệ thống điều khiển rời rạc**

### *Điều kiện ổn định của hệ rời rạc*

Hệ thống được gọi là ổn định nếu tín hiệu vào bị chặn thì tín hiệu ra bị chặn (ổn định BIBO - Bounded Input Bounded Output).

Ta đã biết hệ thống điều khiển liên tục ổn định nếu tất cả các nghiệm của phương trình đặc tính đều nằm bên trái mặt phẳng phức. Do quan hệ giữa biến z và biến s là z = nên s nằm bên trái mặt phẳng phức tương đương với z nằm bên trong vòng tròn đơn vị. Do đó hệ thống điều khiển rời rạc ổn định nếu tất cả các nghiệm của phương trình đặc trưng đều nằm bên trong vòng tròn đơn vị.

Miền ổn định

Res

Ims

Miền ổn định của hệ liên tục

Rez

Imz

Miền ổn định của hệ rời rạc

Hệ thống rời rạc ổn định |z|<1

Hệ thống rời rạc cho bởi sơ đồ khối:

G(s)

H(s)

*Phương trình đặc trưng là:*

1 + GH(z) = 0

Hệ thống rời rạc cho hệ phương trình biến trạng thái:

*Phương trình đặc tính là:*

## **Chất lượng hệ thống rời rạc**

### *Đáp ứng quá độ*

Một hệ thống điều khiển tự động được gọi là ổn định khi tín hiệu ra của hệ thống tắt dần theo thời gian. Để mô tả quá trình động hay quá trình tắt dẫn của hệ thống ổn định đó theo thời gian người tả sử dụng đặc tính quá độ. Như vậy đặc tính quá độ mô tả quá trình quá độ của hệ thống.

Có thể xác định được đáp ứng của hệ thống rời rạc bằng một trong hai cách sau đây:

* *Cách 1*: tính C(z), sau đó dùng phép biến đổi Z ngược để tìm c(k).
* *Cách 2*: tính nghiệm x(k) của phương trình trạng thái của hệ rời rạc c(k).

Cặp cực quyết định: hệ bậc cao có thể xấp xỉ gần đúng về hệ bậc hai với hai cực với hai cực là cặp cực quyết định.

Đối với hệ liên tục, cặp cực quyết định là cặp cực nằm gần trục ảo nhất. Do z=eTs, nên đối với hệ rời rạc cặp cực quyết định là cặp cực nằm gần vòng tròn đơn vị nhất.

### *Độ vọt lố*

Đối với hệ rời rạc, cách thường sử dụng để tính độ vọt lố là dùng biểu thức định nghĩa:

Trong đó:

Cmax: là giá trị cực đại của c(k).

Cxl: là giá trị cực đại của c(k).

Cách thứ hai cũng được sử dụng khi biết cặp cực quyết định z\* = r của hệ rời rạc là dựa vào quan hệ z = eTs để suy ra nghiệm s\*, từ đó tính được .

Sau đó áp dụng các công thức đã trình bày để tính POT,txl....

### *Sai số xác lập*

Theo định lý giá trị cuối:

Các công thức tính sai số xác lập:

G(s)

ZOH

H(s)

r(t)

e(k)

c(t)

T

Sai số xác lập của hệ điều khiển rời rạc có sơ dồ như trên là:

Đặt Kp =

=>

Nếu tín hiệu vào là hàm dốc đơn vị:

=>

Đặt

=>

## **Hàm truyền của các khâu hiệu chỉnh rời rạc**

### *Khâu tỉ lệ*

### *Khâu vi phân*

Vi phân

e(t)

u(t)

Khâu vi phân liên tục:

Khâu vi phân rời rạc: Được tính bằng các công thức sai phân, có ba cách tính

*Sai phân tới:*

*Sai phân lùi:*

*Sai phân giữa:*

Công thức sai phân tới và sai phân giữa cần tín hiệu e(k+1) là tín hiệu sai số trong tương lai, ma trong các bài toán điều khiển thời gian thực ta không thể có được tín hiệu trong tương lai (trừ khi sử dụng bộ dự báo) nên thực tế chỉ có công thức sai phân lùi dược sử dụng phổ biến nhất, do đó.

### *Khâu tích phân*

Tích phân

e(t)

u(t)

Khâu tích phân liên tục:

Khâu tích phân rời rạc:

### *Bộ điều khiển PI, PD, PID rời rạc*

Từ các hàm truyền rời rạc cơ bản vừa phân tích ở trên, ta rút ra được hàm truyền của bộ điều khiển PI, PD, PID số như sau:

### *Tính toán xây dựng mô hình toán học cho bộ điều khiển PID*

Quá trình gia nhiệt là quá trình phổ biến được ứng dụng rộng rãi trong công nghiệp như thực phẩm, sản xuất và nông nghiệp. Sưởi ấm thích hợp trong quá trình sản xuất sản phẩm để đảm bảo chất lượng và kéo dài tuổi thọ. Điều khiển PID được biết đến như một điều khiển đơn giản và mạnh mẽ được ứng dụng điều khiển quá trình gia nhiệt này. Theo thống kê có hơn 97% bộ điều khiển trong công nghiệp là cùng loại với bộ điều khiển PID. Điều chỉnh đúng các thông số điều khiển sẽ đảm bảo tối ưu hiệu suất trên phạm vi hoạt động đã được chỉ định. Một số phương pháp nổi bật là phương pháp Ziegler-Nichols thứ nhất và thứ hai, PID liên tục, và PID số (PID rời rạc)…Một số kỹ thuật điều khiển PID nổi tiếng được tìm thấy trong đồ án này, trong bài viết này em sẽ đề cập đến phương pháp điều khiển PID số (PID rời rạc) để điều khiển nhiệt độ lò nhiệt. Đầu tiên, em sẽ thiết kế hàm truyền rời rạc cho hệ thống. Sau đó, em mô phỏng hàm truyền thiết kế được bằng phần mềm Matlab/Simulink, để tìm ra quy luật thay đổi của hệ thống. Từ đó, áp dụng vào thực nghiệm tìm ra được bộ thông số điều khiển tốt nhất cho hệ thống.

## **Thiết lập mô hình toán học lò ấp trứng**

### *Mô hình toán học của hệ thống*

Up(t)

Rd(t)

GPID(z) =

ZOH

Hệ thống

e(t)=yd(t)-y(t)

u(k)

+

-

Hình 2. Sơ đồ mô tả hệ thống

### *Hàm truyền lò ấp trứng*

Lò ấp trứng có đầu vào là điện áp (hay công suất) cung cấp cho dây điện trở và ngõ ra là nhiệt độ bên trong lò. Để lập hàm truyền lò nhiệt ta phải khảo sát phương trình vi phân mô tả các quan hệ nhiệt độ và năng lượng. Đây là một bài toán phức tạp nếu muốn mô tả chính xác hàm truyền phi tuyến của hệ thống. Một cách gần đúng, ta có thể xem môi trường nung là đồng chất, đẳng nhiệt. Từ phương trình cân bằng năng lượng : điện năng cung cấp sẽ được dùng để bù vào năng lượng nhiệt truyền ra bên ngoài và tích nhiệt vào môi trường nung.

*a, Mô hình lò điện trở:*

Lò ấp trứng

Công suất P= 100%

Nhiệt độ 0t

c(t) 0C

K

T1 T2 t(sec)

c(t) 0C

K

T1 T2 t(sec)

a)

b)

Hình 2. 8 Thí nhiệm xác định hàm truyền lò ấp trứng

*Hình 2. 9 Đặc tính của lò ấp trứng*

*a) chính xác b) gần đúng*

*b, Hàm truyền*

Hàm truyền được xác định gần đúng theo:

Tín hiệu vào của hệ thống lò nhiệt ở dạng hàm nấc đơn vị (công suất trên điện trở 100%)

Tín hiệu ra của hệ thống gần đúng Hình 4.3.b chính là hàm: c(t) = ƒ(t – T1)

Trong đó:

Từ đây tra bảng biến đổi Laplace ta được:

Áp dụng tính chất trễ của biến đổi Laplace ta được:

Suy ra hàm truyền của lò nhiệt:

*Trong đó :*

K: là hệ số tỉ lệ cho biết quan hệ vào ra ở chế độ xác lập.

T1: Thời gian trễ

T2: Thời gian đáp ứng

Mô hình hàm truyền này cho thấy quá trình quá độ với đầu vào hàm nấc có dạng hàm mũ. Thực tế cho thấy mô hình trên chỉ là gần đúng, hệ thống có bậc cao hơn nhưng quá trình quá độ đầu vào hàm nấc vẫn là không vọt lố, có dạng như hình sau khi cho nhiệt độ đầu bằng 0.

### *Hàm chuyển đổi PID rời rạc*

Hàm truyền khâu hiệu chỉnh PID:

Suy ra:

Tín hiệu điều khiển PID cho hệ rời rạc:

Trong đó:

* u(k): là tín hiệu điều khiển thời điểm hiện tại e(k) là sai số ở thời điểm hiện tại.
* e(k-1): là sai số trước đó 1 mẫu.
* e(k-2): là sai số trước đó 2 mẫu.
* T: là thời gian lấy mẫu

*Xác định thông số PID của lò ấp trứng*

Việc tìm và xác định thông số PID một cách chính xác chúng ta có tới 4 phương pháp. Các phương pháp này có thể áp dụng theo phương thức thủ công hoặc tính toán dựa vào phần mềm hoặc công thức.

*Phương pháp kiểm soát thông số PID bằng tay:*

Là phương thức đưa các giá trị Ki = Kd đồng thời quy chúng về giá trị = 0. Sau đó; ta hiệu chỉnh giá trị Kp tăng dần đến khi nào hệ thống dao động tuần hoàn ổn định thì ngưng

Kiểm soát lại giá trị Kp vừa khớp với yêu cầu bài toán. Trong qua trình hiệu chỉnh, nếu xảy ra hiện tượng thay đổi do dao động đột biến thì ta hiệu chuẩn thêm giá trị Kd

*Phương pháp Ziegler – Nichols:*

Được xem là một trong những phương pháp chọn ra thông số điều khiển PID hữu hiệu nhất

Với phương pháp này ta giả sử Ki = Kd trong điều kiện đưa về giá trị 0 giống phương pháp trên. Tiếp tục tăng Kp ( Lúc này Kp cũng được xem là Kc ).

Sau đó, ta đo đếm chu kỳ dao động Pc là xong.

*Sử dụng phần mềm chọn thống số PID:*

Xác định thông số trong điều khiển pid bằng phần mềm được xem là phương án tối ưu vì nó có độ chính xác rất cao

Ở đây ta có thể tìm pid dựa trên thuật toán điều khiển hiệu chỉnh từ phần mềm trong một mô hình thực nghiệm PID ảo.

*Kết luận:* Do nhận thấy phương pháp kiểm soát thông số PID bằng tay khá đơn giản và đạt hiệu quả cao nên nhóm chùng em quyết định sử dụng phương pháp kiểm soát thông số PID bằng tay để lựa chọn thông số PID cho bài toán của mình.

*Các thông số PID được lựa chọn trong bài:*

* Kp = 9
* Kd = 0.001
* Ki = 0.01

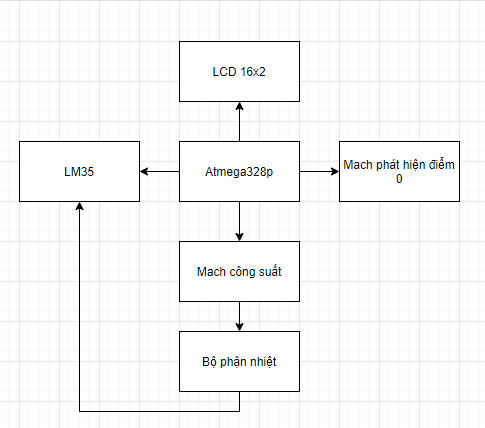
# **CHƯƠNG 3: THIẾT KẾ THI CÔNG MẠCH ĐIỀU KHIỂN VÀ MẠCH CÔNG SUẤT**



## **Xây dựng phần cứng**

### *Sơ đồ chức năng*

*a, Sơ đồ khối*



Hình 3. 1 Sơ đồ khối hệ thống

*b, Chức năng các khối*

*LCD16x2:* Hiển thị dữ liệu

*LM35:* Đo nhiệt dộ hiện tại của môi trường

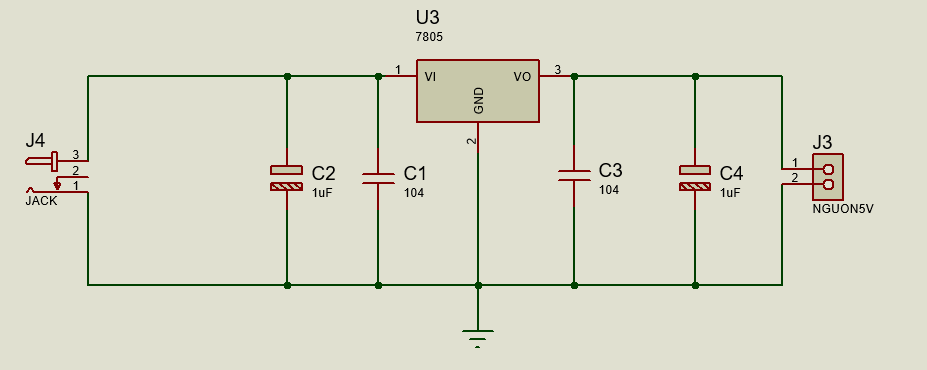
*Mạch phát hiện điểm 0:* Ngoài ra ta còn phải quan tâm đến điểm đồng bộ, tức là điểm giao nhau giữa chu kỳ âm và chu kỳ dương. Ta chỉ điều khiển được góc kích α khi tìm được điểm đồng bộ.

*Mạch công suất:* Vì ta sử dụng vi điều khiển để điều khiển triac nên có khối cách ly, để cách ly điện áp bên khối công suất 200VAC với điện áp của vi điều khiển 5VDC.

*Bộ phận nhiệt:* Gồm 1 bóng đèn sợi đôt 60W.

### *Sơ đồ nguyên lý*

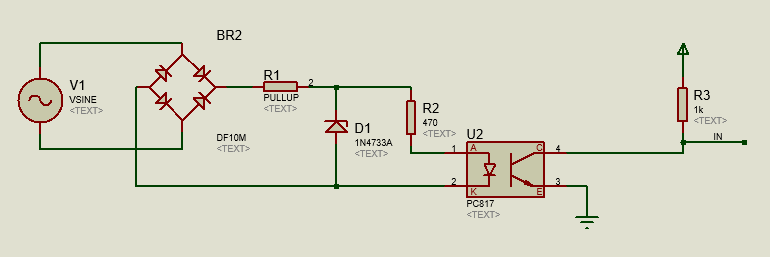
*a, Khối nguồn*



Hình 3. 2 Sơ đồ nguyên lý khối nguồn

*Chức năng:* Cung cấp nguồn cho toàn bộ hệ thống hoạt động

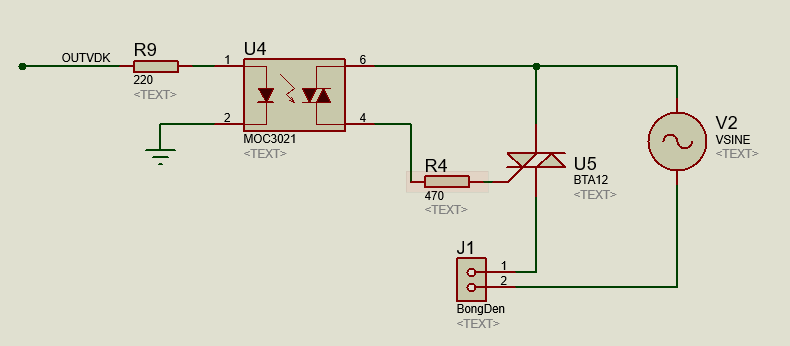
### *Mạch phát hiện điểm không*



Hình 3. 3 Mạch phát hiện điểm 0

Sau khi nhận tín hiệu từ cảm biến vi điều khiển sẽ đọc tín hiệu từ chân Vout của mạch cảm biến và xử lí cho ra tín hiệu phù hợp để kích chân G của triac làm cho đèn sáng. Sự thay đổi của góc kích quyết đinh công suất mở của triac. Khi góc kích thì điện áp đầu ra của triac cũng sẽ thay đổi. Góc kích = 0 công suất đạt 100% triac mở hoàn toàn, góc lúc công suất đạt 50% triac mở một nửa, góc lúc công suất đạt 0% triac ngưng dẫn. Em điều khiển được góc kích khi tìm được điểm 0, tức là điểm giao nhau giữa chu kỳ âm và chu kỳ dương của điện áp xoay chiều.

### *Mạch động lực*

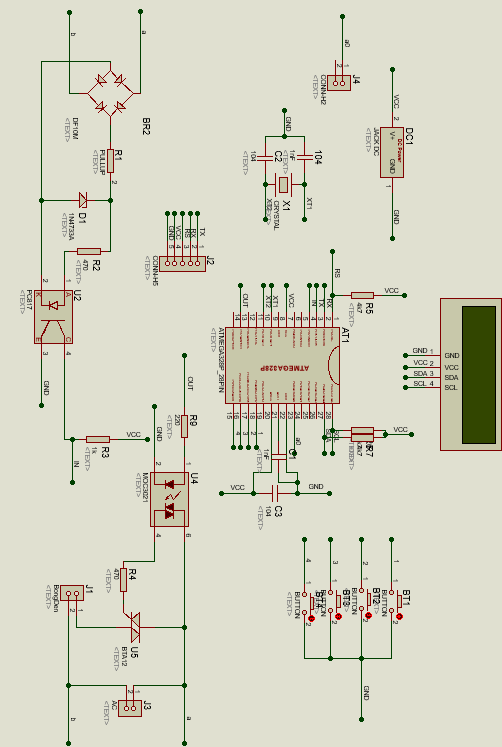


Hình 3. 4 Mạch động lực

Vì ta sử dụng vi điều khiển để điều khiển triac nên có khối cách ly, để cách ly điện áp bên khối công suất 200VAC với điện áp của vi điều khiển 5VDC.

Sử dụng MOC3021 để làm khối cách ly và cầu chì F1 để bảo vệ van công suất.

### *Sơ đồ nguyên lý toàn mạch*



Hình 3. 5 Sơ đồ nguyên lý toàn mạch

## **Giới thiệu một số công cụ sử dụng**

### *Cảm biến nhiệt độ LM35*

*a, Giới thiệu*

Đặc điểm:

* Dải nhiệt độ biến đổi: 0 đến 100 độ.
* Nhiệt độ ra thẳng thang đo Celcius nghĩa là ở 25 độ C điện áp ra là 0.25V.
* Tương ứng 10mV/độ C.
* Đảm bảo độ chính xác 0.5 độ C tại nhiệt độ 25 độ C.
* Làm việc với nguồn nôi là 4V đến 30V.
* Trở kháng ra thấp 0.1 ohm với tải 1mA.
* Khả năng tự làm nóng thấp 0.08 độ C trong không khí.
* Cách mắc: đơn giản là nối chân +Vs với nguồn và chân GND với đất chân OUTPUT nối với mạch khuếch đại.

*b, Dải nhiệt độ và sự thay đối trở kháng theo nhiệt độ của LM35*

Các bộ biến đổi (Transducer) chuyển đổi các đại lượng vật lý ví dụ như nhiệt độ, cường độ ánh sáng, lưu tốc và tốc độ thành các tín hiệu điện phụ thuộc vào bộ biến đổi mà đầu ra có thể là tín hiệu dạng điện áp, dòng, trở kháng hay dung kháng. Ví dụ, nhiệt độ được biến đổi thành về các tín hiệu sử dụng một bộ biến đổi gọi là Thermistor (bộ cảm biến nhiệt), một bộ cảm biến nhiệt đáp ứng sự thay đổi nhiệt độ bằng cách thay đối trở kháng nhưng đáp ứng của nó không tuyến tính.

|  |  |
| --- | --- |
| Nhiệt độ (0C) | Trở kháng của cảm biến (kΩ) |
| 0 | 29.490 |
| 25 | 10.000 |
| 50 | 3.893 |
| 75 | 1.700 |
| 100 | 0.817 |

*Bảng 3. 1 Trở kháng của bộ cảm biến nhiệt theo nhiệt độ*

|  |  |  |  |
| --- | --- | --- | --- |
| Mã sản phẩm | Dải nhiệt độ | Độ chính xác | Đầu ra |
| LM35A | -55 0C to +150 0C | + 1.0 0C | 10mV/F |
| LM35 | -55 0C to +150 0C | + 1.5 0C | 10mV/F |
| LM35CA | -40 0C to +110 0C | + 1.0 0C | 10mV/F |
| LM35C | -40 0C to +110 0C | + 1.5 0C | 10mV/F |
| LM35D | 0 0C to +100 0C | + 2.0 0C | 10mV/F |

Bảng 3. 2 Hướng dẫn chọn loạt các cảm biến nhiệt độ LM35

Loạt các bộ cảm biến LM35 là bộ cảm biến nhiệt mạch tích hợp chính xác cao mà điện áp đầu ra của nó tỷ lệ tuyến tính với nhiệt độ theo thang độ Celsius. Chúng cũng không yêu cầu cân chỉnh ngoài vì vốn chúng đã được cân chỉnh. Chúng đưa ra điện áp 10mV cho mỗi sự thay đổi 1°C. Bảng 2.2 hưởng dẫn ta chọn các cảm biến họ LM35.

*c, Nguyên lý đo và chuyên đổi tương tự/số của ADC*

Khi nhiệt độ môi trường thay đổi làm cho trở kháng của cảm biến LM35 thay đổi dẫn đến điện áp đầu vào Vin của ADC thay đổi. Điện áp Vin vào ADC sẽ được so sánh với Ud của ADC. Ud có thể thay đổi từ 0V đến 2(Vref/2).

Ban đầu Ud =0, nếu Vin > Ud khi đó Ud sẽ được cộng thêm một giá trị là .

Ud = Ud +U , trong đó :

Đồng thời giá trị bộ đếm tăng thêm 1. Quá trình so sảnh cứ như vậy đến khi nào Ud = Vin thì dừng. Khi đồ giá trị của bộ đếm chính là giá trị thập phân. Khả năng tự làm nóng của LM35DZ trong không khí là 0,1°C.

Cảm biến LM35 có hệ số nhiệt là 10mV/°C, do đó sai số về nhiệt LM35DZ sẽ là 0.5°C. Khi đó điện áp đầu vào Vin sẽ được làm tròn lên hoặc xuống . Khi Vin đi vào so sánh với điện áp Ud của ADC, ta nhận thấy rằng áp của ADC là 10mV nên sai số mà nó gây ra là 0,5mV tương ứng với giá trị nhiệt độ là 0,5°C.

Vậy sai số tông cộng của hệ thống là 1. 1°C.

Do vậy nhiệt độ thực tế của ta sẽ là : treal = tđo 1,1°C.

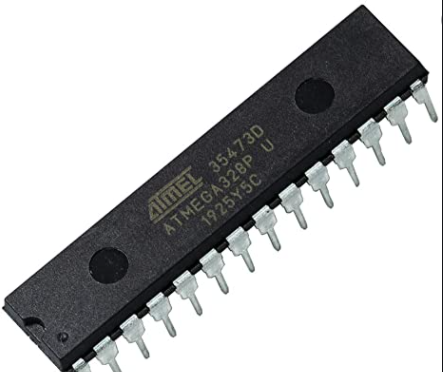
Để tăng độ chính xác của phép đo lên ta có thể dùng một cảm biến khác có độ chính xác cao hơn, có thể giảm điện áp đưa vào chân Vref/2 để giảm bước điện áp vi phân bậc thang của ADC, Tuy nhiên, Khi bước điện áp của ADC và cảm biến không đồng nhất thì sẽ gây khó khăn cho quá trình xử lý dữ liệu đưa ra khâu hiển thị hoặc có thể gây ra sai số. Tùy thuộc vào phép hiện thị mà người ta có thể đặt giá trị điện áp cho chân Vref/2 sao cho hợp lý.

### *Atmega328p*

*a, Giới thiệu*

Atmega328 là một chíp vi điều khiển được sản xuất bời hãng [Atmel](http://www.atmel.com/) thuộc họ MegaAVR có sức mạnh hơn hẳn [Atmega8](https://linhkienbandan.com/shop/atmega8-16pu-vi-dieu-khien-8-bit/). Atmega 328 là một bộ vi điều khiển 8 bít dựa trên kiến trúc RISC bộ nhớ chương trình 32KB ISP flash có thể ghi xóa hàng nghìn lần, 1KB EEPROM, một bộ nhớ RAM vô cùng lớn trong thế giới vi xử lý 8 bít (2KB SRAM).

Với 23 chân có thể sử dụng cho các kết nối vào hoặc ra i/O, 32 thanh ghi, 3 bộ timer/counter có thể lập trình, có các gắt nội và ngoại (2 lệnh trên một vector ngắt), giao thức truyền thông nối tiếp USART, SPI, I2C. Ngoài ra có thể sử dụng bộ biến đổi số tương tự 10 bít (ADC/DAC) mở rộng tới 8 kênh, khả năng lập trình được watchdog timer, hoạt động với 5 chế độ nguồn, có thể sử dụng tới 6 kênh điều chế độ rộng xung (PWM), hỗ trợ bootloader.

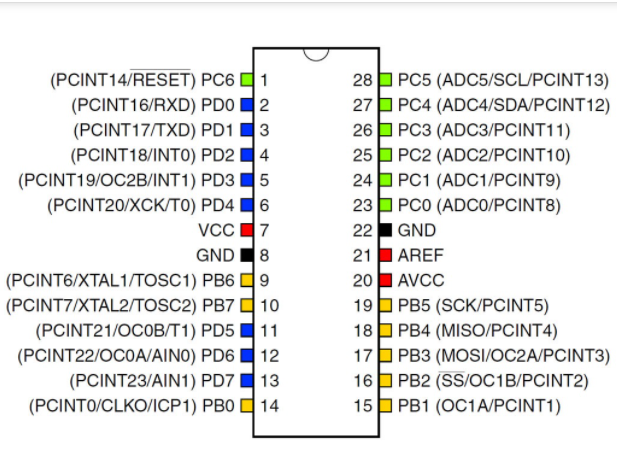


Hình 3. 6 Atmega328p

*b, Thông số kỹ thuật*

* Điện áp sử dụng: 5 VDC
* Điện áp giao tiếp: 5 VDC
* Dòng tiêu thụ: Max 0,2mA (nên sử dụng module cấp nguồn riêng cho mạch).
* Kiến trúc: AVR 8bit
* Xung nhịp lớn nhất: 20Mhz
* Bộ nhớ chương trình (FLASH): 32KB
* Bộ nhớ EEPROM: 1KB
* Bộ nhớ RAM: 2KB
* Số timer: 3 timer gồm 2 timer 8-bit và 1 timer 16-bit
* Số kênh xung PWM: 6 kênh (1timer 2 kênh)

*c, Chức năng các chân*

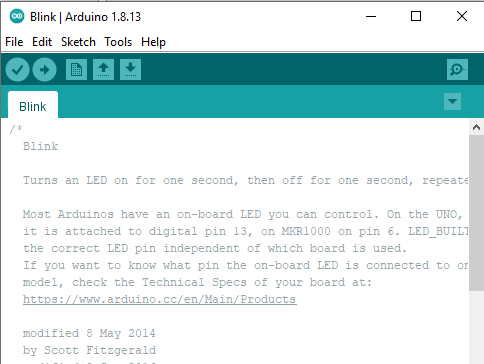


Hình 3. 7 Sơ đồ chân Atmega328p

### *Arduino IDE*

Các thiết bị dựa trên nền tảng Arduino được lập trình bằng ngôn riêng. Ngôn ngữ này dựa trên ngôn ngữ Wiring được viết cho phần cứng nói chung. Và Wiring lại là một biến thể của C/C++. Một số người gọi nó là Wiring, một số khác thì gọi là C hay C/C++ và Ngôn ngữ Arduino bắt nguồn từ C/C++ phổ biến hiện nay do đó rất dễ học, dễ hiểu. Nếu học tốt chương trình Tin học 11 thì việc lập trình Arduino sẽ rất dễ lập trình.

Để lập trình cũng như gửi lệnh và nhận tín hiệu từ mạch Arduino, nhóm phát triển dự án này đã cũng cấp đến cho người dùng một môi trường lập trình Arduino được gọi là Arduino IDE (**I**ntergrated **D**evelopment **E**nvironment) như hình dưới đây.

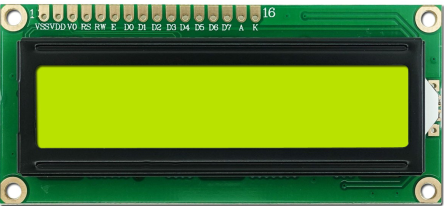


Hình 3. 8 Giao diện phần mềm Arduino

### *Màn hình LCD 16x2*

*a, Giới thiệu*

Ngày nay, thiết bị hiển thị LCD (Liquid Crystal Display) được sử dụng trong rất nhiều các ứng dụng của VĐK. LCD có rất nhiều ưu điểm so với các dạng hiển thị khác: Nó có khả năng hiển thị kí tự đa dạng, trực quan (chữ, số và ký tự đồ họa), dễ dàng đưa vào mạch ứng dụng theo nhiều giao thức giao tiếp khác nhau, tốn rất ít tài nguyên hệ thống và giá thành rẻ …



Hình 3. 9 Màn hình LCD 16x2

*b, Sơ đồ chân và chức năng*

|  |  |  |
| --- | --- | --- |
| Chân | Ký hiệu | Mô tả |
| 1 | Vss | Chân nối đất cho LCD, khi thiết kế mạch ta nối chân này với GND của mạch điều khiển |
| 2 | VDD | Chân cấp nguồn cho LCD, khi thiết kế mạch ta nối chân này với VCC=5V của mạch điều khiển |
| 3 | VEE | Điều chỉnh độ tương phản của LCD. |
| 4 | RS | Chân chọn thanh ghi (Register select). Nối chân RS với logic “0” (GND) hoặc logic “1” (VCC) để chọn thanh ghi.  + 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. |
| 5 | R/W | Chân chọn chế độ đọc/ghi (Read/Write). Nối chân R/W với logic “0” để LCD hoạt động ở chế độ ghi, hoặc nối với logic “1” để LCD ở chế độ đọc. |
| 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 E.  + Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào(chấp nhận) thanh ghi bên trong nó 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. |
| 7 - 14 | DB0 - DB7 | Tám đường của bus dữ liệu dùng để trao đổi thông tin với MPU. Có 2 chế độ sử dụng 8 đường bus này :  + Chế độ 8 bit : Dữ liệu được truyền trên cả 8 đường, với bit MSB là bit DB7.  + Chế độ 4 bit : Dữ liệu được truyền trên 4 đường từ DB4 tới DB7, bit MSB là DB7 |
| 15 | - | Nguồn dương cho đèn nền |
| 16 | - | GND cho đèn nền |

Bảng 3. 3 Chức năng các chân LCD 16x2

### *3.2.4.Module I2C*

*a, Giới thiệu*

LCD có quá nhiều nhiều chân gây khó khăn trong quá trình đấu nối và chiếm dụng nhiều chân trên vi điều khiển. **Module I2C LCD** ra đời và giải quyết vấn để này cho bạn.

*b, Thông số kỹ thuật*

* Điện áp hoạt động: 2.5-6V DC.
* Hỗ trợ màn hình: LCD1602,1604,2004 (driver HD44780).
* Giao tiếp: I2C.
* Địa chỉ mặc định: 0X27 (có thể điều chỉnh bằng ngắn mạch chân A0/A1/A2).
* Tích hợp Jump chốt để cung cấp đèn cho LCD hoặc ngắt.
* Tích hợp biến trở xoay điều chỉnh độ tương phản cho LCD.

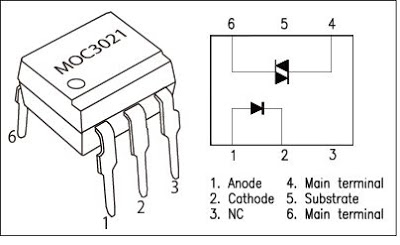
### *Opto MOC3021-DIP 6*

*a, Thông số kỹ thuật*

|  |  |
| --- | --- |
| Điện áp cách ly | 5000Vrms |
| Số kênh | 1 |
| Điện áp tắt trạng thái | 400V |
| Loại đầu ra | AC, Triac, Tiêu chuẩn |
| Dòng điện tối đa | 15mA |
| Dòng giữ | 100A |
| Dòng chuyển tiếp DC | 50mA |
| Đầu ra/kênh | - |
| Kiểu lắp | Qua lỗ |
| Gói/vỏ | 6-DIP (0.300", 7.62mm) |

*Bảng 3. 4 Thông số kĩ thuật của Opto MOC3021-DIP6*

*b. Sơ đồ chân*



Hình 3. Sơ đồ chân Opto MOC3201-DIP 6

*c. Tính năng ứng dụng*

MOC3021 - DIP 6 là một loại Opto - Triac là linh kiện được sử dụng để cách ly giữa các khối chênh lệch nhau về điện hay công suất.

Cách ly điều khiển giữa hai tầng mạch điện khác nhau.

Ngăn các xung điện áp cao hay các phần tử công suất lớn có thể làm hư hỏng các ngõ điều khiển công suất nhỏ trên board mạch.

Với giá thành rẻ, hữu ích nên rất được ưa dùng trong mạch điện tử.

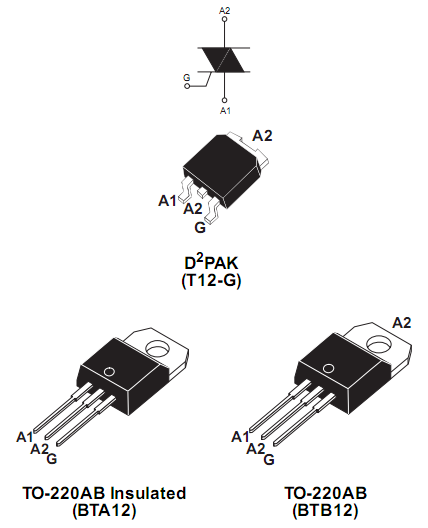
### *Triac BTA12*

*a, Thông số kỹ thuật*

|  |  |
| --- | --- |
| Điện áp cực đại | 600A |
| Dòng điện thuận cực đại | 12A |
| Điện áp điều khiển mở van | 1.3V |
| Dòng điện mở | 50mA |
| Nhiệt độ làm việc | -400C đến 1250C |

Bảng 3. 5 Thông số kĩ thuật của Triac BTA12

*b, Sơ đồ chân*



Hình 3. Sơ đồ chân Triacs BTA12

*c, Ứng dụng*

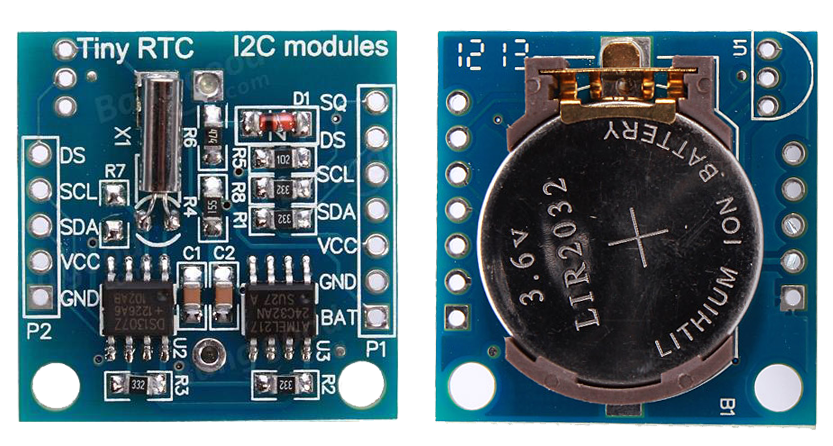
**Triac BTA12**  thích hợp sử dụng trong các mạch chuyển đổi AC. Chúng có thể được sử dụng như một bật/tắt chức năng trong các ứng dụng, cảm ứng khởi động động cơ mạch, bộ điều khiển tốc độ động cơ,..v.v.

### *Đồng hồ thời gian thực DS1307*

Đồng hồ thời gian thực hay còn gọi với cái tên là **Read Time Clock sử dụng Module DS1307**. Được sử dụng rất phổ biến trong các ứng dụng của Arduino.

Để có thể làm việc với module này chúng ta cần sử dụng một chuẩn giao tiếp cũng khá quen thuộc là I2C Bus.

Để các bạn có thể hiểu rõ hơn chúng ta đi ngay vào bài viết nhé.



Hình 3. 12 Module DS1307

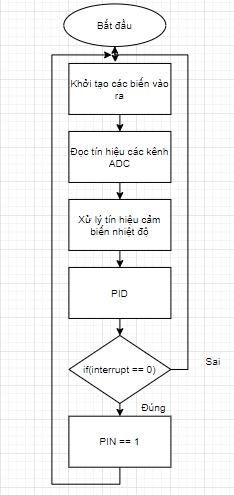
Một số tính năng quan trọng của **DS1307** là:

* Khả năng tạo sóng vuông có thể lập trình.
* Dòng điện thấp, dưới 500mA trong chế độ sao lưu pin.
* Khả năng thiết lập ngày đến năm 2100.
* Sử dụng chuẩn giao tiếp I2C.
* Module DS1307 sử dụng pin CR2023 3 volt. Bộ nhớ EEPROM 24c32 nhúng trên mô-đun này có thể tiết kiệm 32kb dữ liệu.
* Ngoài ra, các bạn có thể đo nhiệt độ môi trường bằng cách sử dụng cảm biến DS18B20 đã được tích hợp sẵn trên board mạch.

## **Xây dựng phần mềm**

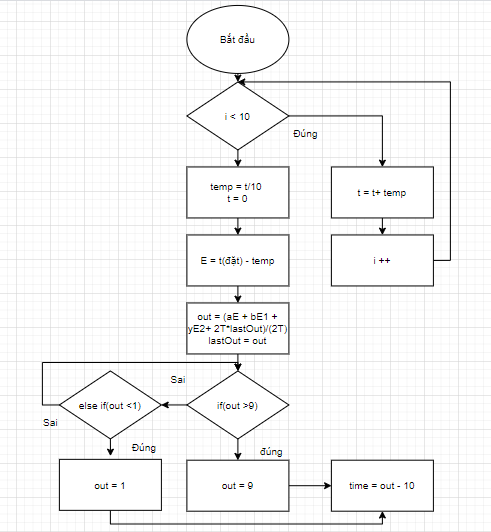
### *Lưu đồ thuật toán điều khiển*

%3CmxGraphModel%3E%3Croot%3E%3CmxCell%20id%3D%220%22%2F%3E%3CmxCell%20id%3D%221%22%20parent%3D%220%22%2F%3E%3CmxCell%20id%3D%222%22%20style%3D%22edgeStyle%3DorthogonalEdgeStyle%3Brounded%3D0%3BorthogonalLoop%3D1%3BjettySize%3Dauto%3Bhtml%3D1%3BexitX%3D0.5%3BexitY%3D1%3BexitDx%3D0%3BexitDy%3D0%3BentryX%3D0.455%3BentryY%3D-0.028%3BentryDx%3D0%3BentryDy%3D0%3BentryPerimeter%3D0%3B%22%20edge%3D%221%22%20source%3D%223%22%20target%3D%225%22%20parent%3D%221%22%3E%3CmxGeometry%20relative%3D%221%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%223%22%20value%3D%22B%E1%BA%AFt%20%C4%91%E1%BA%A7u%22%20style%3D%22ellipse%3BwhiteSpace%3Dwrap%3Bhtml%3D1%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22354%22%20y%3D%2270%22%20width%3D%22120%22%20height%3D%2250%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%224%22%20style%3D%22edgeStyle%3DorthogonalEdgeStyle%3Brounded%3D0%3BorthogonalLoop%3D1%3BjettySize%3Dauto%3Bhtml%3D1%3BexitX%3D0.5%3BexitY%3D1%3BexitDx%3D0%3BexitDy%3D0%3BentryX%3D0.5%3BentryY%3D0%3BentryDx%3D0%3BentryDy%3D0%3B%22%20edge%3D%221%22%20source%3D%225%22%20target%3D%227%22%20parent%3D%221%22%3E%3CmxGeometry%20relative%3D%221%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%225%22%20value%3D%22Kh%E1%BB%9Fi%20t%E1%BA%A1o%20c%C3%A1c%20bi%E1%BA%BFn%20v%C3%A0o%20ra%22%20style%3D%22rounded%3D0%3BwhiteSpace%3Dwrap%3Bhtml%3D1%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22360%22%20y%3D%22140%22%20width%3D%22120%22%20height%3D%2260%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%226%22%20style%3D%22edgeStyle%3DorthogonalEdgeStyle%3Brounded%3D0%3BorthogonalLoop%3D1%3BjettySize%3Dauto%3Bhtml%3D1%3BexitX%3D0.5%3BexitY%3D1%3BexitDx%3D0%3BexitDy%3D0%3BentryX%3D0.5%3BentryY%3D0%3BentryDx%3D0%3BentryDy%3D0%3B%22%20edge%3D%221%22%20source%3D%227%22%20target%3D%229%22%20parent%3D%221%22%3E%3CmxGeometry%20relative%3D%221%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%*3CmxCell*%20id%3D%227%22%20value%3D%22%C4%90%E1%BB%8Dc%20t%C3%ADn%20hi%E1%BB%87u%20c%C3%A1c%20k%C3%AAnh%20ADC%22%20style%3D%22rounded%3D0%3BwhiteSpace%3Dwrap%3Bhtml%3D1%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22360%22%20y%3D%22220%22%20width%3D%22120%22%20height%3D%2260%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%228%22%20style%3D%22edgeStyle%3DorthogonalEdgeStyle%3Brounded%3D0%3BorthogonalLoop%3D1%3BjettySize%3Dauto%3Bhtml%3D1%3BexitX%3D0.5%3BexitY%3D1%3BexitDx%3D0%3BexitDy%3D0%3BentryX%3D0.5%3BentryY%3D0%3BentryDx%3D0%3BentryDy%3D0%3B%22%20edge%3D%221%22%20source%3D%229%22%20target%3D%2211%22%20parent%3D%221%22%3E%3CmxGeometry%20relative%3D%221%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%229%22%20value%3D%22X%E1%BB%AD%20l%C3%BD%20t%C3%ADn%20hi%E1%BB%87u%20c%E1%BA%A3m%20bi%E1%BA%BFn%20nhi%E1%BB%87t%20%C4%91%E1%BB%99%22%20style%3D%22rounded%3D0%3BwhiteSpace%3Dwrap%3Bhtml%3D1%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22360%22%20y%3D%22300%22%20width%3D%22120%22%20height%3D%2260%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%2210%22%20style%3D%22edgeStyle%3DorthogonalEdgeStyle%3Brounded%3D0%3BorthogonalLoop%3D1%3BjettySize%3Dauto%3Bhtml%3D1%3BexitX%3D0.5%3BexitY%3D1%3BexitDx%3D0%3BexitDy%3D0%3BentryX%3D0.5%3BentryY%3D0%3BentryDx%3D0%3BentryDy%3D0%3B%22%20edge%3D%221%22%20source%3D%2211%22%20target%3D%2214%22%20parent%3D%221%22%3E%3CmxGeometry%20relative%3D%221%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%2211%22%20value%3D%22PID%22%20style%3D%22rounded%3D0%3BwhiteSpace%3Dwrap%3Bhtml%3D1%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22360%22%20y%3D%22380%22%20width%3D%22120%22%20height%3D%2260%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%2212%22%20style%3D%22edgeStyle%3DorthogonalEdgeStyle%3Brounded%3D0%3BorthogonalLoop%3D1%3BjettySize%3Dauto%3Bhtml%3D1%3BexitX%3D0.5%3BexitY%3D1%3BexitDx%3D0%3BexitDy%3D0%3BentryX%3D0.5%3BentryY%3D0%3BentryDx%3D0%3BentryDy%3D0%3B%22%20edge%3D%221%22%20source%3D%2214%22%20target%3D%2216%22%20parent%3D%221%22%3E%3CmxGeometry%20relative%3D%221%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%2213%22%20style%3D%22edgeStyle%3DorthogonalEdgeStyle%3Brounded%3D0%3BorthogonalLoop%3D1%3BjettySize%3Dauto%3Bhtml%3D1%3BexitX%3D1%3BexitY%3D0.5%3BexitDx%3D0%3BexitDy%3D0%3BentryX%3D0.5%3BentryY%3D1%3BentryDx%3D0%3BentryDy%3D0%3B%22%20edge%3D%221%22%20source%3D%2214%22%20target%3D%223%22%20parent%3D%221%22%3E%3CmxGeometry%20relative%3D%221%22%20as%3D%22geometry%22%3E%3CArray%20as%3D%22points%22%3E%3CmxPoint%20x%3D%22500%22%20y%3D%22500%22%2F%3E%3CmxPoint%20x%3D%22500%22%20y%3D%22130%22%2F%3E%3CmxPoint%20x%3D%22414%22%20y%3D%22130%22%2F%3E%3C%2FArray%3E%3C%2FmxGeometry%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%2214%22%20value%3D%22if(interrupt%20%3D%3D%200)%22%20style%3D%22rhombus%3BwhiteSpace%3Dwrap%3Bhtml%3D1%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22360%22%20y%3D%22460%22%20width%3D%22120%22%20height%3D%2280%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%2215%22%20style%3D%22edgeStyle%3DorthogonalEdgeStyle%3Brounded%3D0%3BorthogonalLoop%3D1%3BjettySize%3Dauto%3Bhtml%3D1%3BexitX%3D0.5%3BexitY%3D1%3BexitDx%3D0%3BexitDy%3D0%3B%22%20edge%3D%221%22%20source%3D%2216%22%20parent%3D%221%22%3E%3CmxGeometry%20relative%3D%221%22%20as%3D%22geometry%22%3E%3CmxPoint%20x%3D%22409.9999999999998%22%20y%3D%22129.9999999999999%22%20as%3D%22targetPoint%22%2F%3E%3CArray%20as%3D%22points%22%3E%3CmxPoint%20x%3D%22420%22%20y%3D%22635%22%2F%3E%3CmxPoint%20x%3D%22340%22%20y%3D%22635%22%2F%3E%3CmxPoint%20x%3D%22340%22%20y%3D%22130%22%2F%3E%3C%2FArray%3E%3C%2FmxGeometry%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%2216%22%20value%3D%22PIN%20%3D%3D%201%22%20style%3D%22rounded%3D0%3BwhiteSpace%3Dwrap%3Bhtml%3D1%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22360%22%20y%3D%22555%22%20width%3D%22120%22%20height%3D%2260%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%2217%22%20style%3D%22edgeStyle%3DorthogonalEdgeStyle%3Brounded%3D0%3BorthogonalLoop%3D1%3BjettySize%3Dauto%3Bhtml%3D1%3BexitX%3D0.5%3BexitY%3D1%3BexitDx%3D0%3BexitDy%3D0%3B%22%20edge%3D%221%22%20source%3D%2214%22%20target%3D%2214%22%20parent%3D%221%22%3E%3CmxGeometry%20relative%3D%221%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%2218%22%20value%3D%22Sai%22%20style%3D%22text%3Bhtml%3D1%3BstrokeColor%3Dnone%3BfillColor%3Dnone%3Balign%3Dcenter%3BverticalAlign%3Dmiddle%3BwhiteSpace%3Dwrap%3Brounded%3D0%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22510%22%20y%3D%22480%22%20width%3D%2240%22%20height%3D%2220%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%2219%22%20value%3D%22%C4%90%C3%BAng%22%20style%3D%22text%3Bhtml%3D1%3BstrokeColor%3Dnone%3BfillColor%3Dnone%3Balign%3Dcenter%3BverticalAlign%3Dmiddle%3BwhiteSpace%3Dwrap%3Brounded%3D0%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22440%22%20y%3D%22535%22%20width%3D%2240%22%20height%3D%2220%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3C%2Froot%3E%3C%2FmxGraphModel%3E



Hình 3. 13 Lưu đồ thuật toán điều khiển

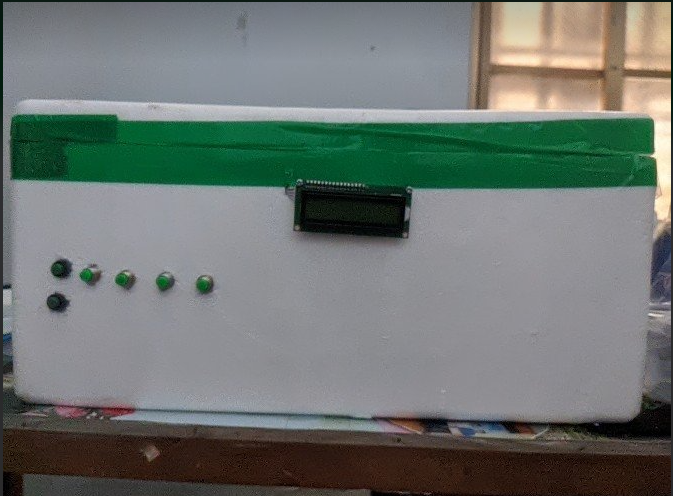
### *Lưu đồ thuật toán PID*



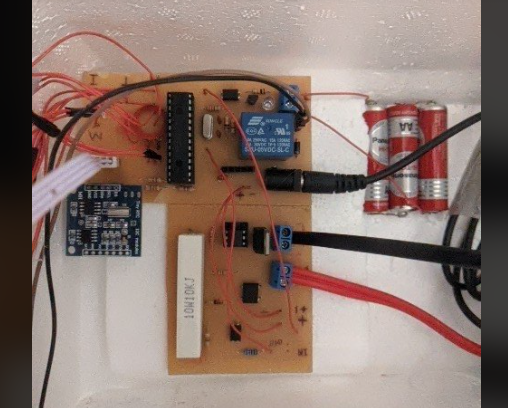
Hình 3. 14 Lưu đồ thuật toán PID

## **Kết quả mô hình thực tế**

### *Sản phẩm thực tế*



Hình 3. 15 Sản phẩm lò ấp trứng



Hình 3. 16 Mạch điều khiển lò ấp



Hình 3. 17 Bộ phận sinh nhiệt cho lò ấp trứng

Mô hình lò nhiệt trong bài báo cáo gồm thùng xốp. Bên trong lò, một bóng đèn thay cho thanh gia nhiệt và cảm biến nhiệt độ LM35 đặt bên trong để đọc nhiệt độ của lò nhiệt.

### *Hướng dẫn sử dụng sản phẩm*

Sản phẩm gồm 6 nút bấm có các chức năng:



Hình 3. 18 Các nút nhấn trên lò ấp

* Hai nút ngoài cùng bên trái tăng giảm nhiệt độ
* Nút thứ ba cài đặt chế độ mở để lò hoạt động
* Nút thứ tư tắt chế độ hoạt động của lò
* Nút thứ năm để next giữa các lựa chọn
* Nút thứ sáu tăng lên chế độ

*Bước 1:*  Khi người dùng muốn chọn loại trứng ấp và nhiệt độ cần ấp thì ấn nút tăng giảm nhiệt độ cho phù hợp.

*Bước 2:* Khi chọn xong chế độ thì chọn thời gian để lò bắt đầu hoạt động.

*Bước 3:* Chọn thời gian kết thúc hoạt động của lò ấp.

*Bước 4:* Sau khi chọn các chế độ và thời gian kết thúc để lò bắt đầu hoạt động.

*Bước 5:* Chờ đợi kết quả đạt được.

**PHỤ LỤC**

#include <Wire.h>

#include <TimerOne.h>

#include <LiquidCrystal\_I2C.h>

#include <RTClib.h>

#include<EEPROM.h>

LiquidCrystal\_I2C lcd(0x27,16,2);

#define LM35\_PIN A0

#define TRIAC\_PIN 7

#define Kp 3

#define Kd 0.001

#define Ki 0.01

int minTempC = 0 ;

int maxTempC = 60;

int nhietdodat = 40;

float nhietdo;

float t;

float E,E1,E2,alpha,gamma,beta;

float Output=0;

float LastOutput=0;

float thoigian=0;

int flag=0;

float T=1.5; //thoi gian lay mau 1.5s

float timerloop;

float tam=0;

byte degree[8] =

{

0B01110,

0B01010,

0B01110,

0B00000,

0B00000,

0B00000,

0B00000,

0B00000

};

RTC\_DS1307 RTC;

#define relay 6

int temp,inc,hours1,minut,add=11;

int HOUR,MINUT,SECOND;

int tang =13;

int giam =12;

int set\_mo =11;

int set\_tat=10;

int INC =9 ;

int next =8 ;

unsigned long time\_now = 0;

void Temperature()

{

float read\_ADC;

read\_ADC=(analogRead(LM35\_PIN)\*5.0/1024.0);

nhietdo=read\_ADC\*100.0; //1000/10

return nhietdo;

}

/\*HAM DIEU KHIEN TRIAC\*/

void TriacControl()

{

delayMicroseconds(thoigian\*1000);

digitalWrite(TRIAC\_PIN,HIGH);

delay(1);

digitalWrite(TRIAC\_PIN,LOW);

}

void PID()

{

for (int i=0; i<10;i++)

{Temperature();

tam+=nhietdo;}

nhietdo=tam/10.0;

tam=0;

E=nhietdodat-nhietdo;

alpha = 2\*T\*Kp + Ki\*T\*T + 2\*Kd;

beta = T\*T\*Ki - 4\*Kd -2\*T\*Kp;

gamma = 2\*Kd;

Output = (alpha\*E + beta\*E1 + gamma\*E2 + 2\*T\*LastOutput)/(2\*T);

LastOutput=Output;

E2=E1;

E1=E;

if (Output>9)

Output=9;

else if (Output<=1)

Output=1;

thoigian = 10-Output; /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

}

void setup() {

Serial.begin(9600);

lcd.init();

lcd.backlight();

lcd.createChar(1, degree);

Wire.begin();

pinMode(relay,OUTPUT);

pinMode(INC, INPUT\_PULLUP);

pinMode(next, INPUT\_PULLUP);

pinMode(set\_mo, INPUT\_PULLUP);

pinMode(set\_tat, INPUT\_PULLUP);

pinMode(tang, INPUT\_PULLUP);

pinMode(giam, INPUT\_PULLUP);

pinMode(TRIAC\_PIN,OUTPUT);

pinMode(LM35\_PIN ,INPUT);

digitalWrite(relay, HIGH);

E=0;E1=0;E2=0;

lcd.setCursor(3,0);

lcd.print("LO AP TRUNG ");

lcd.setCursor(5,1);

lcd.print("TU DONG ");

delay(2000);

if (! RTC.begin()) {

while (1);

}

if (! RTC.isrunning()) {

RTC.adjust(DateTime(F(\_\_DATE\_\_), F(\_\_TIME\_\_)));

}

}

void loop() {

//nut\_tang

if (digitalRead(tang)==LOW){ // Trường hợp nút nhấn được nhấn

nhietdodat= nhietdodat+1 ; // tăng giá trị cài lên 1 đơn vị

if(nhietdodat < minTempC) nhietdodat = minTempC;

}

//nut\_giam

if (digitalRead(giam)==LOW){

nhietdodat= nhietdodat-1 ; // giảm giá trị cài lên 1 đơn vị

if(nhietdodat > maxTempC) nhietdodat = maxTempC;

}

//nut\_mo

int temp=0,val=1,temp4;

DateTime now = RTC.now();

if(digitalRead(set\_mo) == LOW) //set Alarm time

{

lcd.setCursor(0,0);

lcd.print(" Set ON ");

delay(2000);

defualt();

time();

delay(1000);

lcd.clear();

lcd.setCursor(0,0);

lcd.print(" Alarm time ");

lcd.setCursor(0,1);

lcd.print(" has been set ");

delay(2000);

}

//nut\_tat

if(digitalRead(set\_tat) == LOW) //set Alarm time

{

lcd.setCursor(0,0);

lcd.print(" Set OFF ");

delay(2000);

defualt();

time();

delay(1000);

lcd.clear();

lcd.setCursor(0,0);

lcd.print(" Alarm time ");

lcd.setCursor(0,1);

lcd.print(" has been set ");

delay(2000);

}

for (int i=0; i<10;i++)

{Temperature();

tam+=nhietdo;}

nhietdo=tam/10.0;

tam=0;

lcd.clear();

lcd.setCursor(0,0);

lcd.print("NDT:");

lcd.setCursor(4,0);

lcd.print(nhietdo);

lcd.setCursor(0,1);

lcd.print("SET:");

lcd.print(nhietdodat);

lcd.setCursor(8, 1);

lcd.print(HOUR=now.hour(),DEC);

lcd.print(":");

lcd.print(MINUT=now.minute(),DEC);

lcd.print(":");

lcd.print(SECOND=now.second(),DEC);

lcd.setCursor(10, 0);

lcd.print(now.day(),DEC);

lcd.print("/");

lcd.print(now.month(),DEC);

match();

match1();

delay(500);

}

void defualt()

{

lcd.setCursor(0,1);

lcd.print(HOUR);

lcd.print(":");

lcd.print(MINUT);

lcd.print(":");

lcd.print(SECOND);

}

void time()

{

int temp=1,minuts=0,hours=0,seconds=0;

while(temp==1)

{

if(digitalRead(INC)==LOW)

{

HOUR++;

if(HOUR==24)

{

HOUR=0;

}

while(digitalRead(INC)==LOW);

}

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Set Alarm Time ");

//lcd.print(x);

lcd.setCursor(0,1);

lcd.print(HOUR);

lcd.print(":");

lcd.print(MINUT);

lcd.print(":");

lcd.print(SECOND);

delay(500);

if(digitalRead(next)==LOW)

{

hours1=HOUR;

EEPROM.write(add++,hours1);

temp=2;

while(digitalRead(next)==LOW);

}

}

while(temp==2)

{

if(digitalRead(INC)==LOW)

{

MINUT++;

if(MINUT==60)

{MINUT=0;}

while(digitalRead(INC)==LOW);

}

// lcd.clear();

lcd.setCursor(0,1);

lcd.print(HOUR);

lcd.print(":");

lcd.print(MINUT);

lcd.print(":");

lcd.print(SECOND);

delay(500);

if(digitalRead(next)==LOW)

{

minut=MINUT;

EEPROM.write(add++, minut);

temp=0;

while(digitalRead(next)==LOW);

}

}

delay(1000);

}

void match()

{

int tem[23];

for(int i=11;i<23;i++)

{

tem[i]=EEPROM.read(i);

}

if(HOUR == tem[11] && MINUT == tem[12])

{

digitalWrite(relay, LOW) ;

attachInterrupt(0, TriacControl, RISING);

Timer1.initialize(1500000); //don vi us

Timer1.attachInterrupt(PID);

}

}

void match1()

{

int tem1[23];

for(int j=11;j<23;j++)

{

tem1[j]=EEPROM.read(j);

}

if(HOUR == tem1[13] && MINUT == tem1[14])

{

digitalWrite(relay, HIGH);

digitalWrite(TRIAC\_PIN, LOW);

lcd.clear();

lcd.setCursor(0,5);

lcd.print("FINISH");

delay(1000);

}

}