**ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH**



TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA ĐIỆN – ĐIỆN TỬ

**BỘ MÔN ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA**

---------------o0o---------------



***ĐỀ TÀI***: *Thiết kế và thi công bộ điều khiển   
nhiệt độ PID lò nhiệt dùng vi xử lý,   
cài đặt và hiển thị nhiệt độ trên smart phone.*

**Giảng viên hướng dẫn : Nguyễn Đức Thành**

**Sinh viên thực hiện : Nguyễn Minh Nhật**

**Lớp : DD17TD03 (Nhóm A04)**

**Mã số sinh viên : 1712472**

**TP. HỒ CHÍ MINH, THÁNG 7 NĂM 2020**

# LỜI CẢM ƠN

Để luận văn này đạt kết quả tốt đẹp, em đã nhận được nhiều sự hỗ trợ, giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy cô, anh chị, bạn bè. Với tình cảm sâu sắc, chân thành, em xin được bày tỏ lòng biết ơn sâu sắc đến tất cả các cá nhân đã tạo điều kiện giúp đỡ trong quá trình học tập và nghiên cứu đề tài.

Trước hết em xin gửi tới các thầy cô Khoa Điện – điện tử nói chung và bộ môn Điều khiển và tự động hóa nói riêng của trường Đại học Bách Khoa Thành phố Hồ Chí Minh lời chào trân trọng, lời chúc sức khỏe và lời cảm ơn sâu sắc. Với sự quan tâm, dạy dỗ, chỉ bảo tận tình chu đáo của thầy cô, đến nay em đã có thể hoàn thành đồ án môn học với đề tài:

"*Thiết kế và thi công bộ điều khiển nhiệt độ PID lò nhiệt dùng vi xử lý, cài đặt và hiển thị nhiệt độ trên smart phone.*".

Đặc biệt em xin gửi lời cảm ơn chân thành nhất tới giảng viên – TS. Nguyễn Đức Thành quan tâm giúp đỡ, hướng dẫn em hoàn thành tốt đồ án môn học này trong thời gian qua.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một sinh viên, luận văn này không thể tránh được những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để em có điều kiện bổ sung, nâng cao kiến thức của mình, phục vụ tốt hơn cho công việc thực tế sau này.

Em xin chân thành cảm ơn

*Tp. Hồ Chí Minh, ngày 20 tháng 7 năm 2020 .*

**Sinh viên**

Nguyễn Minh Nhật

# TÓM TẮT ĐỒ ÁN

Đồ án này trình bày về bộ điều khiển PID – một sự kết hợp của 3 bộ điều khiển: tỉ lệ, tích phân và vi phân, có khả năng điều chỉnh sai số thấp nhất có thể, tăng tốc độ đáp ứng, giảm độ vọt lố, hạn chế sự dao động.

Trên hết là áp dụng được bộ điều khiển PID vào ứng dụng thực tế, cụ thể ở đây là điều khiển hệ thống lò nhiệt dựa trên bộ điều khiển PID thông qua vi xử lý. Trong đề tài lần này, ngoài việc điều khiển PID hệ thống lò nhiệt bằng phần cứng, còn mở rộng thêm yêu cầu điều khiển thông qua smartphone. Cụ thể ở đây là sẽ gởi giá trị nhiệt độ lên smartphone và gởi giá trị đặt từ smartphone về cho hệ thống xử lý.

Trong đồ án lần này, em chọn vi điều khiển ATmega328 họ 8 bit được tích hợp trong board mạch Arduino dùng để lập trình bộ điều khiển PID cho lò nhiệt. Bên cạnh đó, em còn dùng kit node MCU ESP8266 với IC chính là ESP8266 Wifi SoC, nhiệm vụ chính của kit này là truyền nhận dữ liệu thông qua internet, ngoài ra, Esp8266 còn thực hiện nhiệm vụ giao tiếp UART với Arduino để có thể truyền nhận các giá trị nhiệt độ. Các linh kiện khác phục vụ cho đồ án như cảm biến nhiệt độ Thermocouple loại K, màn hình LCD 16x2,…

Quá trình chính của hệ thống sẽ là điều khiển giá trị nhiệt độ đặt ban đầu, có thể điều khiển nút nhấn từ phần cứng đã thiết kế truyền vào Arduino, hoặc là ESP8266 sẽ đọc giá trị cài đặt từ smartphone và truyền dữ liệu về Arduino; Dựa vào giá trị ban đầu đó, Arduino sẽ tính toán các giá trị của bộ điều khiển PID theo phần mềm đã thiết kế, điều khiển xung PWM của ngõ ra thành phần gia nhiệt. Nhiệt độ hiện tại mà cảm biến nhiệt độ đo được sẽ được hiển thị lên LCD và truyền tới ESP8266 để gởi lên Webserver và hiển thị lên Smartphone. Sau thời gian nhất định, hệ thống sẽ đạt được nhiệt độ mà người điều khiển mong muốn.

# MỤC LỤC

[LỜI CẢM ƠN 2](#_Toc45302006)

[TÓM TẮT ĐỒ ÁN 3](#_Toc45302007)

[MỤC LỤC 4](#_Toc45302008)

[DANH SÁCH HÌNH MINH HỌA 6](#_Toc45302009)

[DANH SÁCH BẢNG SỐ LIỆU 8](#_Toc45302010)

[PHẦN 1: GIỚI THIỆU 9](#_Toc45302011)

[PHẦN 2: LÝ THUYẾT VÀ KIẾN THỨC LIÊN QUAN 11](#_Toc45302012)

[I. Các hệ thống điều khiển - Hệ thống điều khiển lò nhiệt 11](#_Toc45302013)

[II. Nhiệt độ - Các loại cảm biến nhiệt độ 13](#_Toc45302014)

[*1.* *Nhiệt độ và các thang đo nhiệt độ* 13](#_Toc45302015)

[*2.* *Sơ lược về một số loại cảm biến nhiệt độ* 14](#_Toc45302016)

[III. Lý thuyết về Thermocouple 16](#_Toc45302017)

[IV. Một số phương pháp điều khiển lò nhiệt 19](#_Toc45302018)

[*1.* *Phương pháp điều khiển ON – OFF:* 19](#_Toc45302019)

[*2.* *Phương pháp điều khiển PID* 20](#_Toc45302020)

[V. Mạch điều khiển Arduino uno 23](#_Toc45302021)

[VI. Kit NodeMCU ESP8266 26](#_Toc45302022)

[VII. Một số modules và linh kiện liên quan 30](#_Toc45302023)

[*1.* *Module I2C* 30](#_Toc45302024)

[*2. Module MAX6675* 31](#_Toc45302025)

[PHẦN 3: THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG 32](#_Toc45302026)

[I. Yêu cầu thiết kế: 32](#_Toc45302027)

[II. Sơ đồ khối tổng quát 32](#_Toc45302028)

[III. Sơ đồ khối chi tiết 34](#_Toc45302029)

[IV. Sơ đồ mạch chi tiết 35](#_Toc45302030)

[PHẦN 4: THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM: 41](#_Toc45302031)

[I. Yêu cầu cho phần mềm: 41](#_Toc45302032)

[II. Lưu đồ giải thuật 41](#_Toc45302033)

[III. Chương trình cho Arduino và NodeMCU ESP8266, lập trình giao diện Blynk. 43](#_Toc45302034)

[PHẦN 5: KẾT QUẢ THỰC HIỆN 49](#_Toc45302035)

[PHẦN 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 57](#_Toc45302036)

[I. Kết luận 57](#_Toc45302037)

[II. Hướng phát triển. 57](#_Toc45302038)

[PHẦN 7: TÀI LIỆU THAM KHẢO 58](#_Toc45302039)

# DANH SÁCH HÌNH MINH HỌA

[*Hình 1. 1. Điều khiển nhiệt độ trong nhành luyện kim* 8](#_Toc45300001)

[*Hình 2. 1. Hệ thống điều khiển hồi tiếp* 10](#_Toc45300017)

[*Hình 2. 2. Hệ thống điều khiển tuần tự* 11](#_Toc45300018)

[*Hình 2. 3. Hệ thống điều khiển nhiệt độ tổng quát* 11](#_Toc45300019)

[*Hình 2. 4. Mô hình RTD 3 dây* 14](#_Toc45300020)

[*Hình 2. 5. IC cảm biến LM35* 15](#_Toc45300021)

[*Hình 2. 6. Đặc điểm của hoạt động ON-OFF* 18](#_Toc45300022)

[*Hình 2. 7. Đặc điểm của Hunting* 19](#_Toc45300023)

[*Hình 2. 8. Mô hình điều khiển PID tổng quát* 19](#_Toc45300024)

[*Hình 2. 9. Vi điều khiển Arduino Uno* 22](#_Toc45300025)

[*Hình 2. 10. Sơ đồ chân chip ESP8266* 26](#_Toc45300026)

[*Hình 2. 11. Board mở rộng NodeMCU ESP8266* 26](#_Toc45300027)

[*Hình 2. 12. ESP8266 chế độ Acess Point* 27](#_Toc45300028)

[*Hình 2. 13. ESP8266 chế độ Station* 28](#_Toc45300029)

[*Hình 2. 14. Module I2C* 29](#_Toc45300030)

[*Hình 2. 15. MAX6675* 30](#_Toc45300031)

[*Hình 3. 1. Sơ đồ khối tổng quát* 33](#_Toc45300045)

[*Hình 3. 2. Sơ đồ khối chi tiết* 34](#_Toc45300046)

[*Hình 3. 3. Sơ đồ mạch chi tiết* 35](#_Toc45300047)

[*Hình 3. 4. Khối nút nhấn* 36](#_Toc45300048)

[*Hình 3. 5. I2C & LCD* 36](#_Toc45300049)

[*Hình 3. 6. Thermocouple K & MAX6675* 37](#_Toc45300050)

[*Hình 3. 7. Mạch điều khiển* 38](#_Toc45300051)

[*Hình 3. 8. Mạch động lực* 38](#_Toc45300052)

[*Hình 3. 9. NodeMCU ESP8266* 39](#_Toc45300053)

[*Hình 3. 10. Các chân được nối của Arduino* 40](#_Toc45300054)

[*Hình 5. 1. Sơ đô kết nối LCD và Thermocouple* 49](#_Toc45300055)

[*Hình 5. 2. Điểm 0 của sóng sine* 50](#_Toc45300056)

[*Hình 5. 3. Điện áp ra cầu diode* 50](#_Toc45300057)

[*Hình 5. 4. Tín hiệu ngõ ra Photocoupler* 51](#_Toc45300058)

[*Hình 5. 5. Xung kích và tín hiệu điều khiển* 52](#_Toc45300059)

[*Hình 5. 6. Mô hình lò nhiệt* 53](#_Toc45300060)

[*Hình 5. 7. Giao diện điều khiển Blynk* 54](#_Toc45300061)

[*Hình 5. 8. Kết quả chạy phần cứng* 55](#_Toc45300062)

[*Hình 5. 9. Kết quả trên Blynk* 56](#_Toc45300063)

# DANH SÁCH BẢNG SỐ LIỆU

[*Bảng 1: Điện áp đầu ra Thermocouple K* 18](#_Toc45300066)

[*Bảng 2: Hiệu chỉnh tham số PID theo phương pháp Nichols - Ziegler* 23](#_Toc45300067)

[*Bảng 3: Thông số cơ bản của Arduino Uno* 26](#_Toc45300068)

[*Bảng 4: Thông số NodeMCU ESP8266* 28](#_Toc45300069)

# PHẦN 1: GIỚI THIỆU

1. ***Tổng quan***

Như chúng ta đã biết, nhiệt độ là một trong những thành phần vật lý rất quan trọng. Việc thay đổi nhiệt độ của một vật chất ảnh hưởng rất nhiều đến cấu tạo, tính chất và các đại lượng vật lý khác của vật chất. Ví dụ như sự thay đổi nhiệt độ của một chất khí sẽ làm thay đổi thể tích, áp suất của chất khí trong bình. Vì vậy, trong nghiên cứu khoa học, trong công nghiệp và trong đời sống sinh hoạt, thu thập các thông số và điều khiển nhiệt độ là điều rất cần thiết.

Trong các lò nhiệt, máy điều hòa, lò vi sóng ,… điều khiển nhiệt độ là tính chất quyết định cho sản phẩm ấy. Trong ngành luyện kim, cần phải đạt đến một nhiệt độ nào đó để kim loại nóng chảy, và cũng cần đạt một nhiệt độ nào đó để ủ kim loại nhằm đạt được tốt các đặc tính cơ học như độ bền, độ dẻo, độ chống gỉ sét ,… Trong ngành thực phẩm, cần duy trì một nhiệt độ nào đó để nướng bánh, để nấu, để bảo quản ,… Việc thay đổi thất thường nhiệt độ, không chỉ gây hư hại đến chính thiệt bị đang hoạt động, còn ảnh hưởng đến quá trình sản xuất, ngay cả trên chính sản phẩm ấy.



*Hình 1. 1. Điều khiển nhiệt độ trong nhành luyện kim*

Có nhiều phương pháp điều khiển lò nhiệt độ. Mỗi phương pháp đều mang đến một kết quả khác nhau thông qua những phương pháp điều khiển khác nhau đó. Trong nội dung đồ án lần này, em đã tìm hiểu về bộ điều khiển PID và áp dụng bộ điều khiển đó cho một hệ thống lò nhiệt tự thiết kế. Ngoài ra, để nâng cao sự đa dụng của sản phẩm, thì điều khiển thông qua smartphone là một vấn đề cần nghiên cứu và áp dụng vào đồ án.

1. ***Nhiệm vụ đề tài***

Để hoàn thành nội dung đồ án môn học lần này, em đã ôn lại những kiến thức đã được dạy trong thời gian qua, đồng thời học hỏi và tìm hiểu các nội dung liên quan khác đến đề tài đồ án. Cụ thể:

***Nội dung 1:*** Tìm hiểu về các loại hệ thống điều khiển và hệ thống điều khiển lò nhiệt.

***Nội dung 2:*** Giới thiệu về các loại cảm biến đo nhiệt độ. Tìm hiểu lý thuyết về cảm biến nhiệt độ được sử dụng trong đồ án lần này là Thermocouple.

***Nội dung 3:*** Tìm hiểu một số phương pháp điều khiển lò nhiệt. Ôn lại lý thuyết về điều khiển PID

***Nội dung 4:*** Tìm hiểu về vi điều khiển Arduino và kit wifi node MCU ESP8266. Sau đó thiết kế phần cứng và phần mềm sử dụng bộ điều khiển PID cho hệ thống điều khiển nhiệt độ.

***Nội dung 5:*** Tìm hiểu về ứng dụng Blynk, thiết kế giao diện để điều khiển nhiệt độ qua Smartphone.

# PHẦN 2: LÝ THUYẾT VÀ KIẾN THỨC LIÊN QUAN

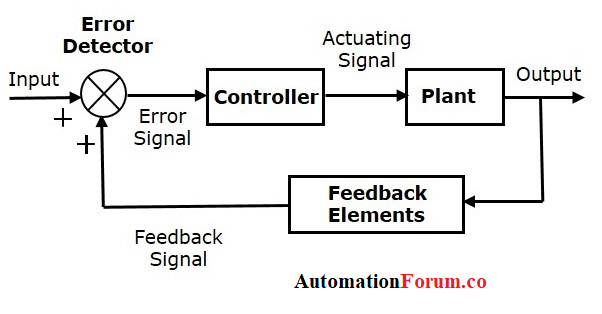
## Các hệ thống điều khiển - Hệ thống điều khiển lò nhiệt

Nhiệt độ là một đại lượng vật lý mà xuất hiện ở khắp mọi nơi trong cả sản xuất lẫn đời sống sinh hoạt hằng ngày. Quá trình đo và kiểm soát nhiệt độ trong sản xuất công nghiệp đóng vai trò rất lớn trong hệ thống điều khiển tự động, góp phần quyết định chất lượng sản phẩm. Khi thu thập dữ liệu cho quá trình điều khiển và giám sát trong nhà máy thì nhiệt độ là một thông số không thể bỏ qua.

Tùy theo yêu cầu và tính chất của quá trình điều khiển mà ta sử dụng phương pháp điều khiển thích hợp. Tính chính xác và ổn định nhiệt độ cũng đặt ra vấn đề cần giải quyết.

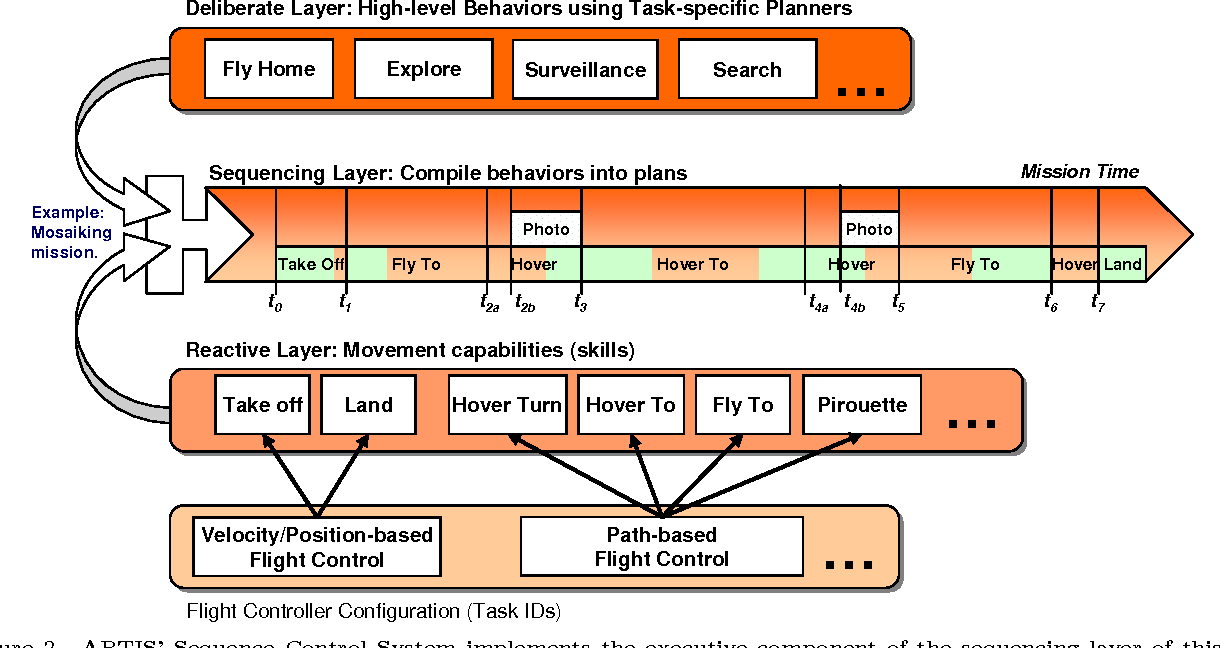
Hệ thống điều khiển nhiệt độ thường được chia làm 2 loại:

* ***Hệ thống điều khiển hồi tiếp (feedback control system):*** thường xác định và giám sát kết quả điều khiển, so sánh với tín hiệu đặt và tự động điều chỉnh lại cho đúng.



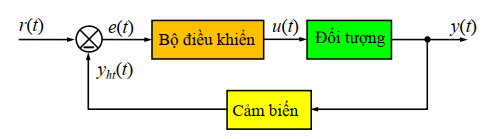
*Hình 2. 1. Hệ thống điều khiển hồi tiếp*

* ***Hệ thống điều khiển tuần tự (sequence control system):*** thực hiện từng bước điều khiển tùy theo hoạt động điều khiển trước khi xác định tuần tự.



*Hình 2. 2. Hệ thống điều khiển tuần tự*

Một hệ thống muốn đạt được độ chính xác cần thiết phải thực hiện hồi tiếp, tín hiệu phản hồi về so sánh với tín hiệu vào và sai lệch sẽ được đưa tới bộ điều chỉnh đầu ra. Hệ thống điều khiển này có nhiều ưu điểm được sử dụng nhiều trên thực tế trong các hệ thống điều khiển tự động. Hệ thống điều khiển nhiệt độ sẽ gồm có 3 thành phần cơ bản: bộ điều khiển, đối thượng (trong lò nhiệt sẽ là bộ phận gia nhiệt) và cảm biến nhiệt độ. Dạng tổng quát của hệ thống điều khiển được mô tả trên nguyên tắc như hình sau:



*Hình 2. 3. Hệ thống điều khiển nhiệt độ tổng quát*

*Trong đó:*

+ r(t) : tín hiệu đặt

+ u(t): tín hiệu vào của đối tượng điều khiển (trong đề tài lần này là xung PWM)

+ y(t): tín hiệu ra của đối tượng điều khiển (nhệt độ)

+ yht(t): tín hiệu hồi tiếp

+ e(t): sai số.

## Nhiệt độ - Các loại cảm biến nhiệt độ

### *Nhiệt độ và các thang đo nhiệt độ*

Nhiệt độ là một trong những thành phần chủ yếu trong hệ thống thu nhập dữ liệu. Do vậy, nếu chọn lựa thiết bị đo lường nhiệt độ chính xác ta có thể tiết kiệm chi phí năng lượng, tăng độ an toàn và giảm thời gian kiểm tra ,… thiết bị đo lường nhiệt độ thường dùng là cảm biến nhiệt độ. Căp nhiệt điện, điện trở nhiệt, thermistors và infrared thermometers là những loại cảm biến nhiệt độ thông thường. Việc chọn lựa thiết bị để hoạt động chính xác tùy thuộc vào nhiệt độ tối đa, tối thiểu cần đo, độ chính xác và những điều kiện về môi trường.

Galileo được cho là người đầu tiên phát minh ra thiết bị đo nhiệt độ vào khoảng năm 1592. Đầu những năm 1700, Gabriel Fahrenheit, nhà chế tạo thiết bị đo người Hà Lan, đã tạo ra một thiết bị đo chính xác và cho phép lặp lại nhiều lần. Khoảng năm 1742, Anders Celsius hoàn thiện thang đo 0 – 100 oC. Đầu những năm 1800, Lord Kelvin phát triển một tầm đo phổ quát dựa trên hệ số giãn nở của khí lý tưởng. Kelvin thiết lập khái niệm về độ 0 tuyệt đối và tầm đo này được chọn là tiêu chuẩn cho đo nhiệt hiện tại. Từ thang nhiệt độ nhiệt động học tuyệt đối (Thang Kelvin), người ta đã xác định thang mới là thang Celsius và thang Fahrenheil.

### *Sơ lược về một số loại cảm biến nhiệt độ*

* ***Thermocouple:***

Thermocouple hay còn gọi là cặp nhiệt điện, bao gồm 2 dây dẫn khác nhau. Thermocouple sẽ tạo ra một điện áp phụ thuộc vào nhiệt độ dựa theo hiệu ứng nhiệt điện.

Cặp nhiệt điện được sử dụng rộng rãi trong khoa học và công nghiệp, thường được sử dụng trong lò nung, động cơ diesel và các quy trình công nghiệp khác.

Ưu điểm:

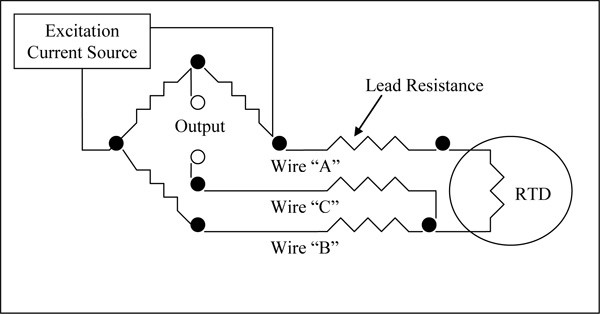
* Là thành phần tích cực, tự cung cấp công suất.
* Đơn giản, rẻ tiền.
* Tầm đo nhiệt rộng.

Khuyết điểm:

* Phi tuyến.
* Điện áp cung cấp thấp.
* Đòi hỏi điện áp tham chiếu.
* Kém ổn định và kém nhạy.
* ***RTD (Resistance temperature detector):***

RTD – nhiệt điện trở là thiết bị phổ biến để đo nhiệt độ trong các ngành công nghiệp, phổ biến nhất là Pt100 chiếm đến 90% nhu cầu sử dụng.

RTD có cấu tạo gồm 2, 3 hoặc 4 dây kim loại cùng loại (3 dây là loại được sử dụng phổ biến). RTD chuyển đổi tín hiệu nhiệt độ đo được thành tín hiệu điện trở.



*Hình 2. 4. Mô hình RTD 3 dây*

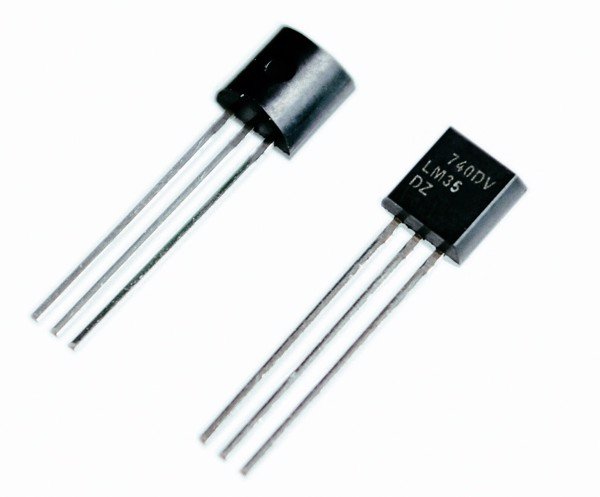
Ưu điểm:

* Ổn định và chính xác.
* Tuyến tính hơn thermocouple.

Khuyết điểm:

* Đắt tiền.
* Cần phải cung cấp nguồn dòng.
* Điện trở tuyệt đối thấp.
* Tự gia tăng nhiệt.
* Tự gia tăng nhiệt.
* ***Cảm biến IC:***

Phổ biến nhất là IC cảm biến LM35, được sử dụng rất nhiều trong các đồ án, luận văn, dự án nhỏ,…



*Hình 2. 5. IC cảm biến LM35*

Ưu điểm:

* Tuyến tính nhất và ngõ ra có giá trị cao nhất.
* Rẻ tiển

Khuyết điểm:

* Nhiệt độ đo dưới 200oC
* Cần cung cấp nguồn cho cảm biến.

## III. Lý thuyết về Thermocouple

1. ***Giới thiệu:***

Cặp nhiệt độ (thermocouple) là một cảm biến nhiệt độ sử dụng 1 mối nối được hình thành bởi các kim loại khác nhau. Chúng thường dùng để đo nhiệt độ rất cao, cao hơn nhiều so với RTD (nhiệt điện trở). Nó chỉ đơn giản là một cặp dây có thành phần khác nhau, được nối ở một đầu. Cặp nhiệt điện tạo ra một điện áp nhỏ, còn được gọi là điện thế cặp nhiệt điện, tỉ lệ thuận với nhiệt độ.

Điện thế cặp nhiệt điện được tạo ra một phần bởi hiệu ứng tiếp giáp, nhưng chủ yếu là do ảnh hưởng của sai lệch nhiệt độ trên mỗi dây, được gọi là hiệu ứng Peltier and seeback. Trừ khi có quy định khác, tất cả các kiểu và bảng căọ nhiệt điện tiêu chuẩn đều hiển thị các điện áp được tham chiếu đến một điểm nối lạnh ở 0 độ C.

Trái ngược với hầu hết các phương pháp đo nhiệt độ khác, cặp nhiệt điện tự cấp nguồn và không yêu cầu kích thích bên ngoài. Thiết bị này được ứng dụng đo nhiệt độ cho lò nung, động cơ diesel. Cặp nhiệt điện còn được sử dụng trong nhà, văn phòng và doanh nghiệp, làm cảm biến nhiệt độ trong các bộ điều nhiệt, trong thiết bị báo cháy…

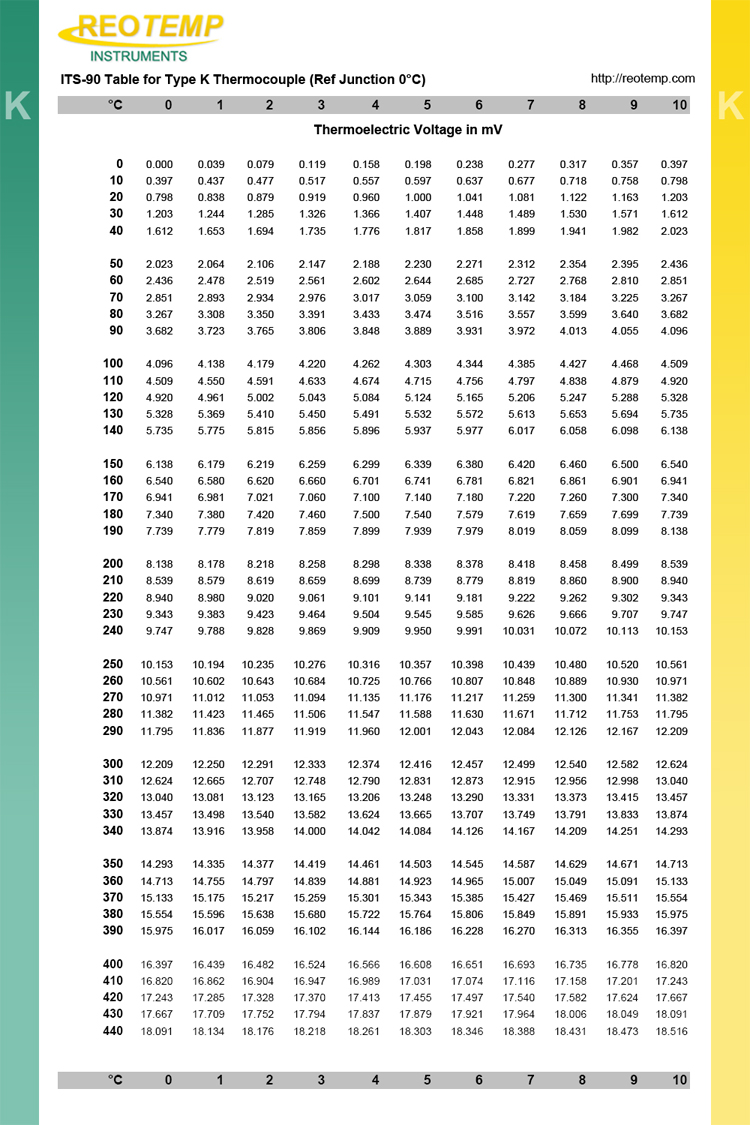
1. ***Thermocouple loại K:***

Được cấu tạo bởi hai dây kim loại: Chromel (hợp kim gồm 90% Niken và 10% Crom) làm dây dương và dây âm là Alumel (hợp kim gồm 95% Niken, 2% mangan, 2% Nhôm, 1% Silic). Đây là loại Thermocouple phổ biến nhất với độ nhạy khoảng 41 μV/ºC.

Loại Thermocouple này tương đối tuyến tính, ít tốn kém, phạm vi hoạt động của nó khá rộng từ -270oC đến +1372oC.

Thermocouple loại K thường được sử dụng trong môi trường nhiệt độ khắc nghiệt. Thường đi chung với bộ chuyển đổi tín hiệu nhiệt độ, để chuyển đổi tín hiệu nhiệt mà cảm biến đo được thành dòng 4-20mA rồi xuất ra thông số nhiệt độ tại thời điểm đo thông qua bộ hiển thị hoặc PLC.

Vì độ phổ biến, giá thành và tính năng của cảm biến nhiệt độ này, em quyết định sử dụng Thermocouple loại K cho đồ án môn học lần này.



*Bảng 1: Điện áp đầu ra Thermocouple K*

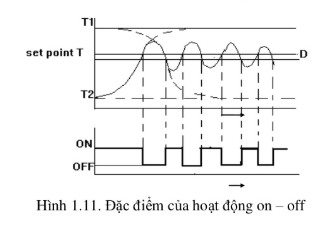
## IV. Một số phương pháp điều khiển lò nhiệt

### *Phương pháp điều khiển 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.

1. *Mô tả hoạt động on-off:*

Với cấu 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 khiển 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 khiển hai vị trí vì hai biến đặt cũng liên quan tới điểm đặt.



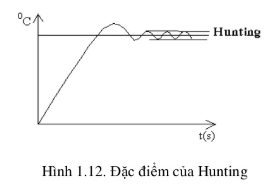
*Hình 2. 6. Đặc điểm của hoạt động ON-OFF*

1. *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 thống điều khiển có thể bị ảnh hưởng nhiễu. Vì lý do này mà tù trễ giữa on off thường xảy ra ở ngõ ra như hình 1.11. Từ trễ này được gọi là hiệu chỉnh độ nhạy. Điều chỉnh độ nhạy cao đòi hỏi cần phải tránh tần số hoạt động on-off.

1. *Hunting*

Khi điểm đặt được điều khiển bằng hoạt động on – off, biến đặt thay đổi tuần hoàn như trong hình 1.12. Sự thay đổi tuần hoàn này được gọi là hunting. Kết quả tốt nhất của hoạt động on-off đạt được nếu như biên độ hunting là nhỏ nhất.



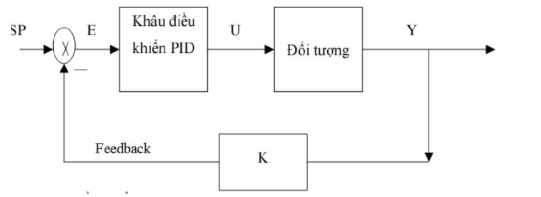
*Hình 2. 7. Đặc điểm của Hunting*

1. *Hệ thống thích hợp cho điều khiển on – off*

Điều khiển on-off tốt nhất cho hệ thống điều khiển 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 tắt bằng hình thức on-off.

### *Phương pháp điều khiển PID*

Một bộ điều khiển PID nói chung đều có mô hình tổng quan dưới dạng:



*Hình 2. 8. Mô hình điều khiển PID tổng quát*

Quá trình điều khiển theo mô hình trên là một quá trình khép kín. Giá trị setpoint là giá trị đặt trước mà hệ thống phải làm việc xung quanh giá trị đó tùy thuộc vào yêu cầu chất lượng của hệ thống. Việc đảm bảo tính ổn định cũng như chất lượng của hệ thống thực chất là đưa hệ thống luôn bám sát SP với độ sai lệch nhỏ nhất và thời gian quá độ nhanh nhất.

Bộ điều khiển PID gồm 3 phần: Tỷ lệ P, tích phân I, Vi phân D. Mỗi thành phần có tác động khác nhau tới quá trình điều khiển của hệ thống. Cụ thể:

1. *Khâu tỷ lệ P:*

Tín hiệu điều khiển u(t) tỉ lệ với tín hiệu sai lệch e(t)

Phương trình sai phân mô tả động học:

Pout = Kp.e(t)

Trong đó: Pout: tín hiệu ra của bộ điều khiển

e(t): tín hiệu vào

Kp: hệ số khuếch đại của bộ điều khiển

1. *Khâu tích phân I:*

Khâu I cộng thêm 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: Iout : Giá trị ngõ ra khâu I

Ki : hệ số tích phân

E: sai số

1. *Khâu vi phân D:*

Khâu D cộng thêm tốc độ thay đổi 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 khiển. 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 D

KD : hệ số vi phân

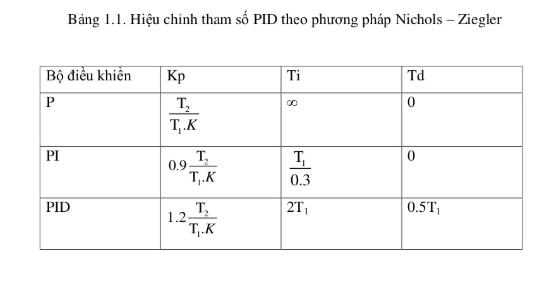
E : sai số

1. *Bộ điều khiển PID cho lò nhiệt*

Tổng hợp ba khâu P, I, D ta được bộ điều khiển PID. Bộ điều khiển PID được mô tả bằng hàm truyền đạt sau:

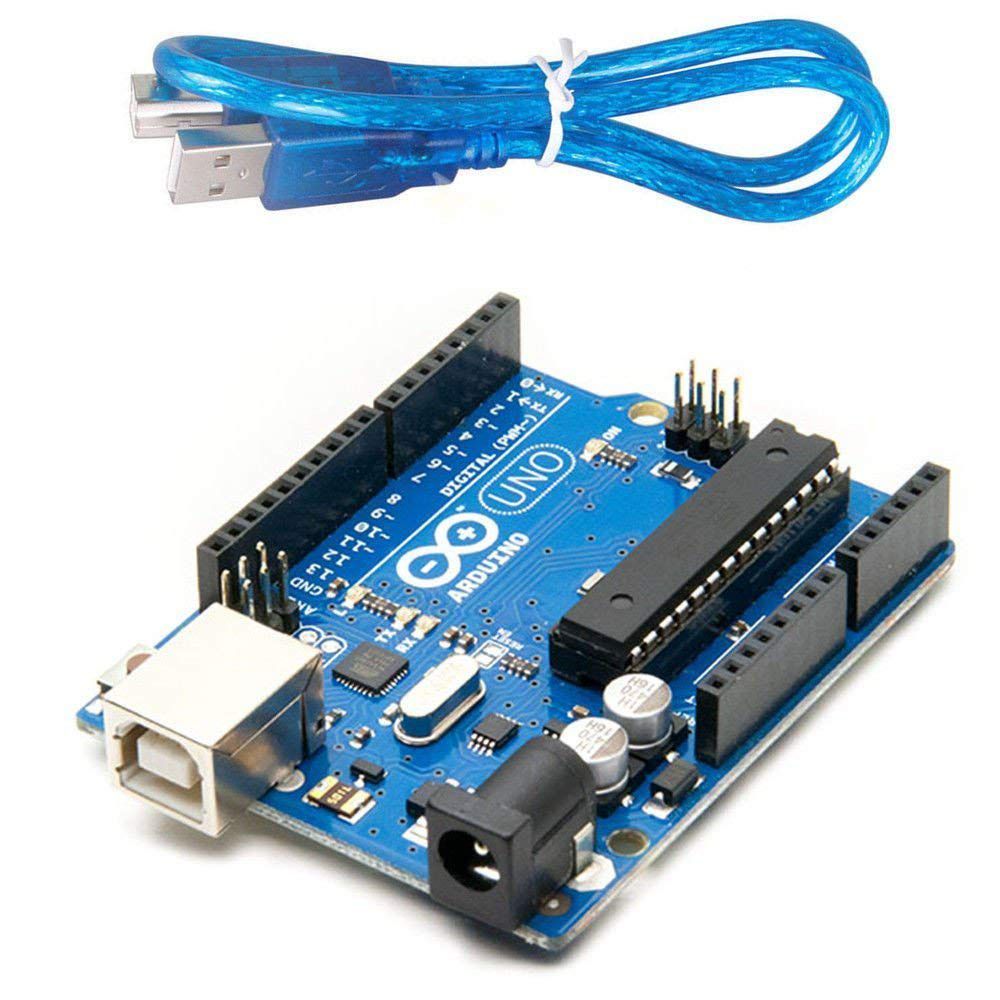
Lò nhiệt có hàm truyền đạt:

Theo Nichols – Ziegler các hằng số Kp, thời hằng tích phân Ti, hằng số vi phân Td của hàm hiệu chỉnh thích hợp ứng với hàm truyền các thông số K, T1, T2 được xác định qua bảng sau:



*Bảng 2: Hiệu chỉnh tham số PID theo phương pháp Nichols - Ziegler*

## V. Mạch điều khiển Arduino uno



*Hình 2. 9. Vi điều khiển Arduino Uno*

Nhắc tới dòng mạch Arduino dùng để lập trình, cái đầu tiên mà ta thường nói tới chính là dòng Arduino UNO. Hiện dòng mạch này đã phát triển tới thế hệ thứ 3 (R3).

1. ***Vi điều khiển***

Arduino UNO có thể sử dụng 3 vi điều khiển họ 8 bit AVR là Atmega8, Atmega168, Atmega328. Bộ não này có thể xử lý nhiều tác vụ đơn giản khác nhau như điều khiển, thu thập - xử lý tín hiệu, hiện thị LCD,… Thiết kế tiêu chuẩn của Arduino UNO là sử dụng vi điều khiển Atmega328.

1. ***Năng lượng***

Arduino UNO có thể được cấp nguồn 5V thông qua cổng USB hoặc cấp nguồn ngoài với điện áp khuyên dùng là 7-12V DC và giới hạn là 6-20V.

Các chân năng lượng:

* **GND (Ground)**: cực âm của nguồn điện cấp cho Arduino UNO. Khi dùng các thiết bị sử dụng những nguồn điện riêng biệt thì những chân này phải được nối với nhau.
* **5V**: cấp điện áp 5V đầu ra. Dòng tối đa cho phép ở chân này là 500mA.
* **3.3V**: cấp điện áp 3.3V đầu ra. Dòng tối đa cho phép ở chân này là 50mA.
* **Vin (Voltage Input)**: để cấp nguồn ngoài cho Arduino UNO, bạn nối cực dương của nguồn với chân này và cực âm của nguồn với chân GND.
* **IOREF**: điện áp hoạt động của vi điều khiển trên Arduino UNO có thể được đo ở chân này. Và dĩ nhiên nó luôn là 5V. Mặc dù vậy, không được lấy nguồn 5V từ chân này để sử dụng bởi chức năng của nó không phải là cấp nguồn.
* **RESET**: việc nhấn nút Reset trên board để reset vi điều khiển tương đương với việc chân RESET được nối với GND qua 1 điện trở 10KΩ.

1. ***Bộ nhớ:***

Vi điều khiển Atmega328 tiêu chuẩn cung cấp cho người dùng:

* **32KB bộ nhớ Flash**: những đoạn lệnh bạn lập trình sẽ được lưu trữ trong bộ nhớ Flash của vi điều khiển. Thường thì sẽ có khoảng vài KB trong số này sẽ được dùng cho bootloader
* **2KB cho SRAM** (**S**tatic **R**andom **A**ccess **M**emory): giá trị các biến bạn khai báo khi lập trình sẽ lưu ở đây. Bạn khai báo càng nhiều biến thì càng cần nhiều bộ nhớ RAM. Khi mất điện, dữ liệu trên SRAM sẽ bị mất.
* **1KB cho EEPROM**(**E**lectrically **E**raseble **P**rogrammable **R**ead **O**nly **M**emory): đây giống như một chiếc ổ cứng mini – nơi có thể đọc và ghi dữ liệu của mình vào đây mà không phải lo bị mất khi cúp điện giống như dữ liệu trên SRAM.

1. ***Các cổng vào/ra:***

Arduino UNO có 14 chân digital dùng để đọc hoặc xuất tín hiệu. Chúng chỉ có 2 mức điện áp là 0V và 5V với dòng vào/ra tối đa trên mỗi chân là 40mA. Ở mỗi chân đều có các điện trở pull-up từ được cài đặt ngay trong vi điều khiển ATmega328 (mặc định thì các điện trở này không được kết nối).

Một số chân digital có các chức năng đặc biệt như sau:

* **2 chân Serial**: 0 (RX) và 1 (TX): dùng để gửi (transmit – TX) và nhận (receive – RX) dữ liệu TTL Serial. Arduino Uno có thể giao tiếp với thiết bị khác thông qua 2 chân này. Kết nối bluetooth thường thấy nói nôm na chính là kết nối Serial không dây. Nếu không cần giao tiếp Serial, không nên sử dụng 2 chân này nếu không cần thiết
* **Chân PWM (~): 3, 5, 6, 9, 10, và 11**: cho phép xuất ra xung PWM với độ phân giải 8 bit (giá trị từ 0 → 28-1 tương ứng với 0V → 5V) bằng hàm analogWrite(). Nói một cách đơn giản, bạn có thể điều chỉnh được điện áp ra ở chân này từ mức 0V đến 5V thay vì chỉ cố định ở mức 0V và 5V như những chân khác.
* **Chân giao tiếp SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).  Ngoài các chức năng thông thường, 4 chân này còn dùng để truyền phát dữ liệu bằng giao thức SPI với các thiết bị khác.
* **LED 13**: trên Arduino UNO có 1 đèn led màu cam (kí hiệu chữ L). Khi bấm nút Reset, bạn sẽ thấy đèn này nhấp nháy để báo hiệu. Nó được nối với chân số 13. Khi chân này được người dùng sử dụng, LED sẽ sáng.

Arduino UNO có 6 chân analog (A0 → A5) cung cấp độ phân giải tín hiệu 10bit (0 → 210-1) để đọc giá trị điện áp trong khoảng 0V → 5V. Với chân **AREF** trên board, bạn có thể để đưa vào điện áp tham chiếu khi sử dụng các chân analog. Tức là nếu bạn cấp điện áp 2.5V vào chân này thì bạn có thể dùng các chân analog để đo điện áp trong khoảng từ 0V  → 2.5V với độ phân giải vẫn là 10bit.

Đặc biệt, Arduino UNO có 2 chân A4 (SDA) và A5 (SCL) hỗ trợ giao tiếp I2C/TWI với các thiết bị khác.

1. ***Lập trình cho Arduino:***

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++. Gọi chung là “*ngôn ngữ Arduino*”, và đội ngũ phát triển Arduino cũng gọi như vậy. Ngôn ngữ Arduino bắt nguồn từ C/C++ phổ biến hiện nay do đó rất dễ học, dễ hiểu.

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).

1. ***Một số thông số cơ bản***

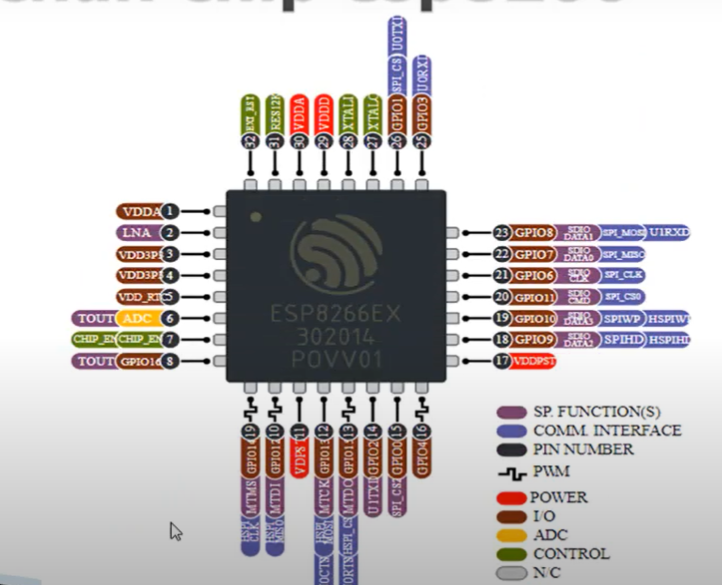
|  |  |
| --- | --- |
| Vi điều khiển | Atmega328P |
| Điện áp hoạt động | 5 V |
| Điện áp đầu vào (khuyến cáo) | 7-12 V |
| Điện áp đầu vào (giới hạn) | 6-20 V |
| Chân I/O Digital | 14 (6 ngõ ra PWM) |
| Chân ngõ vào Analog | 6 (độ phân giải 10 bits) |
| Dòng tối đa trên chân I/O | 30 mA |
| Dòng ra tối đa (5V) | 500 mA |
| Dòng ra tối đa (3.3V) | 50mA |
| Flash Memory | 32 KB (0.5KB được sử dụng cho bootloader) |
| SRAM | 2 KB |
| EEPROM | 1 KB |
| Tốc độ xung clock | 16 MHz |
| LED\_BUILTIN | 13 |
| Chiều dài | 68.6 mm |
| Chiều rộng | 53.4 mm |
| Trọng lượng | 25g |

*Bảng 3: Thông số cơ bản của Arduino Uno*

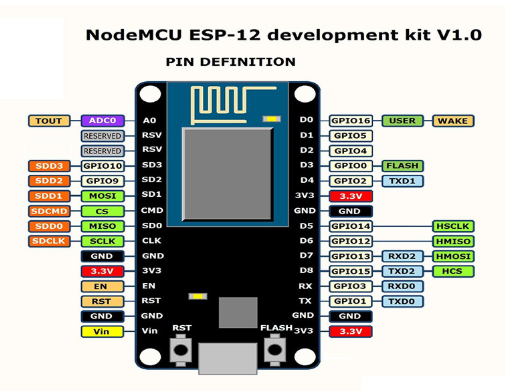
## VI. Kit NodeMCU ESP8266

1. ***Khái niệm***

ESP8266 là một mạch vi điều khiển có thể giúp chúng ta điều khiển các thiết bị điện tử. Điều đặc biệt của nó, đó là sự kết hợp của module Wifi 2.4GHz tích hợp sẵn bên trong con vi điều khiển chính. Hiện nay, ESP8266 rất được giới nghiên cứu tự động hóa Việt Nam ưa chuộng vì giá thành cực kỳ rẻ.



*Hình 2. 10. Sơ đồ chân chip ESP8266*



*Hình 2. 11. Board mở rộng NodeMCU ESP8266*

ESP8266 có nhiều phiên bản và được đóng gói theo nhiều cách khác nhau, tuy nhiên nó lại khá giống nhau về chức năng và khả năng lập trình. Trên thị trường phổ biến nhất hiện nay là ESP8266v1, ESP8266v7 và ESP8266v12.

1. ***Các thống số cơ bản của NodeMCU ESP8266***

|  |  |
| --- | --- |
| Wifi | 2.4 GHz hỗ trợ chuẩn 802.11 b/g/n |
| Điện áp hoạt động | 3.3 V |
| Điện áp vào | 5V |
| Số chân I/O | 11 |
| Số chân Analog input | 1 |
| Bộ nhớ Flash | 4MB |
| Giao tiếp | Cáp Micro USB |
| Hỗ trợ bảo mật | WPA/WPA2 |
| Tích hợp giao thức | TCP/IP |
| Ngôn ngữ lập trình | C/C++, Micropython,… |

*Bảng 4: Thông số NodeMCU ESP8266*

1. ***Chế độ hoạt động esp8266***
2. ***Access Point Mode***

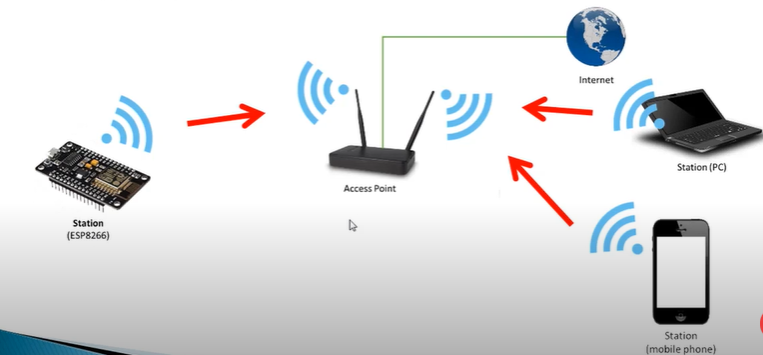


*Hình 2. 12. ESP8266 chế độ Acess Point*

Mạch ESP8266 sẽ biến thành một điểm phát wifi, để cho các thiết bị của chúng ta có thể truy cập vào. Đồng thời, nó sẽ xây dựng một webserver để thiết bị có thể truy cập vào để điều khiển thiết bị mà vi xử lý này điều khiển.

1. ***Station Mode***

Chip vi xử lý sẽ truy cập vào điểm phát wifi, sau đó sẽ xây dựng một webserver trên địa chỉ IP mà ESP8266 kết nối vào wifi. Ta sử dụng thiết bị kết nối vào điểm phát wifi đó và truy cập vào IP chứa webserver mà ESP8266 tạo ra để điều khiển thiết bị.



*Hình 2. 13. ESP8266 chế độ Station*

1. ***Station và Access Point Mode***

Chế độ này sẽ bao gồm cả hai chế độ ở trên. Chip ESP8266 sẽ vừa truy cập vào điểm phát wifi, cập nhật trạng thái mà chip này điều khiển gởi lên internet, để mọi thiết bị đều có thể điều khiển; vừa là điểm phát wifi cho các thiết bị truy cập.

## VII. Một số modules và linh kiện liên quan

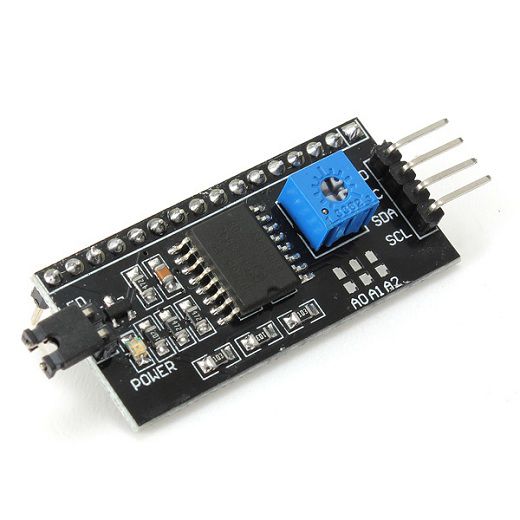
### *Module I2C*

Đầu năm 1980, Philips đã phát triển một chuẩn giao tiếp nối tiếp 2 dây đó chính là I2C (Inter-Intergrated Circuit). Đây là đường bus giao tiếp giữa các IC với nhau, được rất nhiều nhà sản xuất IC trên thế giới sử dụng.

I2C sử dụng 2 đường truyền tín hiệu:

* Một đường xung nhịp đồng hồ (SCL) chỉ do Master phát đi
* Một đường dữ liệu SDA theo 2 hướng

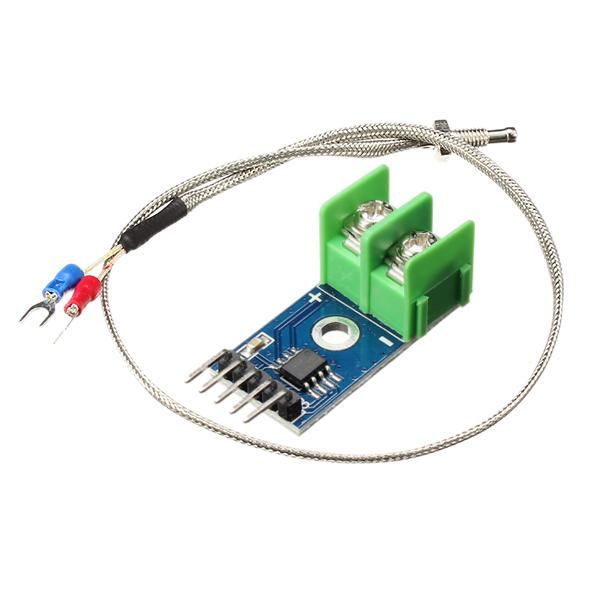
Mỗi thiết bị có 1 địa chỉ được cài sẵn hoặc 1 địa chỉ thiết bị duy nhất để thiết bị chủ (Master) có thể giao tiếp. 2 chân SDA VÀ SCL là 2 chân của giao tiếp I2C, trong đó chân SCL là chân Clock, có tác dụng đồng bộ hóa việc truyền dữ liệu giữa các thiết bị, và việc tạo ra xung clock đó là do thiết bị chủ (Master). Chân còn lại là chân SDA là chân truyền dữ liệu (DATA). 2 chân này luôn hoạt động ở chế độ mở, vì vậy để sử dụng được cần phải có trở kéo. tức là nối +5v => trở => I2C bởi các thiết bị trên bus i2c hoạt động ở mức thấp.



*Hình 2. 14. Module I2C*

### *2. Module MAX6675*

Đây là module thường được sử dụng chung với Thermocouple loại K, điện áp hoạt động từ 3 – 5.5V. Độ phân giải ADC 12 bits tương ứng khoảng 0.25oC trên 1 đơn vị ADC. Module nàu đóng vai trò như một mạch khuếch đại, có tính chính xác cao và ổn định trong quá trình hoạt động.



*Hình 2. 15. MAX6675*

# PHẦN 3: THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG

## I. Yêu cầu thiết kế:

Thi công một lò nhiệt tượng trưng, vì điều kiện hạn chế nên em chỉ thiết kế lò nhiệt công suất nhỏ, sử dụng bóng đèn làm điện trở đốt nóng, dùng nguồn điện 220V có thể gia nhiệt lên khoảng 120oC.

Thiết kế sơ đồ mạch dùng để điều khiển lò nhiệt. Cảm biến nhận giá trị nhiệt độ từ cảm biến, đưa tín hiệu về vi xử lý, hiển thị lên LCD. Có các nút nhấn để điều khiển giá trị setpoint lên xuống. Lập trình bộ điều khiển PID cho vi xử lý, xuất tín hiệu điều khiển bộ phận gia nhiệt, điều khiển nhiệt độ bám theo nhiệt độ đặt. Sai số giữa nhiệt độ lò nhiệt và nhiệt độ đặt không quá 2oC.

Sau đó lập trình phần mềm cho kit NodeMCU ESP8266, giao tiếp với Arduino, truyền dữ liệu từ Aruino đến ESP8266, truyền dữ liệu lên điện thoại thông qua app Blynk. Bên cạnh đó, ESP8266 sẽ nhận giá trị từ điện thoại về, truyền đến Arduino để xử lý.

## II. Sơ đồ khối tổng quát

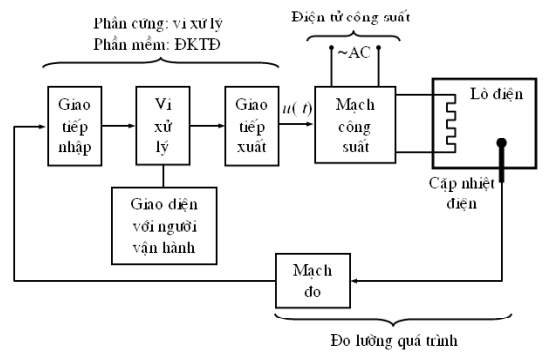
Nhìn chung, một hệ thống điều khiển lò nhiệt PID không quá phức tạp. Để thiết kế điều khiển PID cho một lò nhiệt cần tổng hợp kiến thức của nhiều môn cơ sở và chuyên ngành trong quá trình học.

Để có thể thiết kế một hệ thống điều khiển lò nhiệt từ vi điều khiển, quá trình nối dây các chân, lập trình cơ bản cho vi điều khiển,… đều là kiến thức có được từ môn Vi xử lý.

Về việc lập trình phần mềm quá trình điều khiển PID, em cần kiến thức từ môn Cơ sở Điều khiển Tự động. Kiến thức từ môn này sẽ giúp thiết kế một hệ thống điều khiển được nhiệt độ bám theo nhiệt độ đặt một cách chính xác nhất.

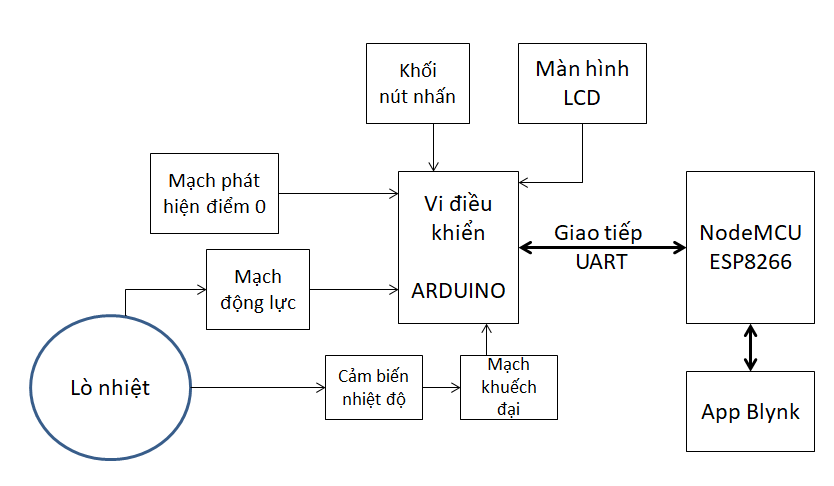
Bên cạnh đó cần có một mạch công suất – thuộc kiến thức của môn Điện tử công suất để điều khiển thiết bị gia nhiệt với điện áp AC 220V.

Phần không thể thiếu đó chính là đọc nhiệt độ của lò nhiệt. Thông qua giá trị điện áp của 2 đầu cảm biến, đi qua một mạch đo hay mạch khuếch đại sẽ cho ta giá trị nhiệt độ hiện thời của lò nhiệt. Đây là nền tảng của môn Đo lường công nghiệp.



*Hình 3. 1. Sơ đồ khối tổng quát*

## III. Sơ đồ khối chi tiết



*Hình 3. 2. Sơ đồ khối chi tiết*

\_ Hai khối có nhiệm vụ nhập và xuất dữ liệu là 2 khối nút nhấn và màn hình LCD. Khối nút nhấn có nhiệm vụ nhập giá trị đặt nhiệt độ ban đầu cho vi điều khiển xử lý. Màn hình LCD có chức năng hiển thị giá trị đặt và giá trị nhiệt độ hiện tại cho người điều khiển giám sát. Vi điều khiển sẽ xử lý sao cho nhiệt độ hiện tại luôn bám sát theo nhiệt độ đặt của người điều khiển chọn.

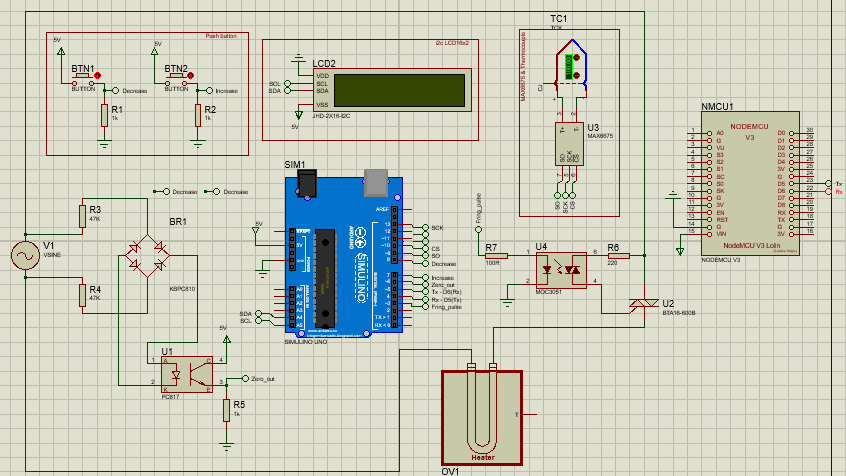
\_ Cảm biến nhiệt độ sẽ nhận nhiệt từ lò nhiệt xuất ra giá trị điện áp. Vì điện áp của cảm biến sẽ nhỏ nên cần đi qua khối khuếch đại để khuếch đại giá trị điện áp đầu ra của cảm biến, giúp cho vi điều khiển xử lý dữ liệu một cách chính xác hơn.

\_ Mạch điều khiển (hay mạch phát hiện điểm 0): có nhiệm vụ chính là nhận biết điểm 0 của điện áp AC 220V, phát tín hiệu điểm 0 về vi điều khiển.

\_ Mạch động lực sẽ dựa vào tín hiệu điểm 0 của mạch điều khiển mà sẽ kích triac để cung cấp nguồn cho lò nhiệt hoạt động.

\_ Cuối cung là phần giao tiếp giữa Arduino và ESP8266 để truyền nhận dữ liệu hoạt động. Cụ thể, Arduino sẽ truyền dữ liệu của nhiệt độ hiện tại qua NodeMCU. Điện thoại và NodeMCU ESP8266 sẽ truy cập vào cùng một điểm phát wifi. Khi MCU nhận được dữ liệu của Arduino, nó sẽ truyền dữ liệu lên Webserver để điện thoại có thể nhận dữ liệu về thông qua app blynk. Đồng thời, giá trị nhiệt độ đặt từ điện thoại cũng sẽ truyền lên Webserver và gởi về kit nodeMCU, dữ liệu đó sẽ truyền về lại Arduino xử lý.

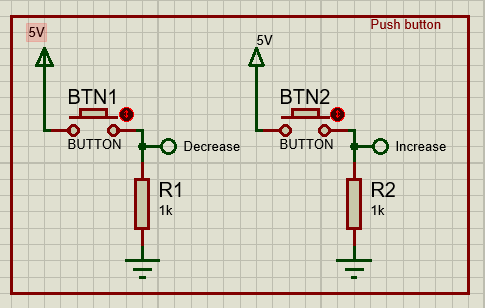
## IV. Sơ đồ mạch chi tiết



*Hình 3. 3. Sơ đồ mạch chi tiết*

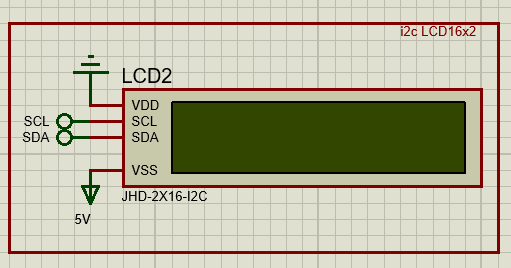
Sơ đồ mạch chi tiết gồm khối nút nhấn, màn hình LCD, cảm biến thermocouple và MAX6675, gia nhiệt heater, mạch phát hiện điểm 0 và mạch động lực. Để giao tiếp với smartphone, có kit wifi nodeMCU ESP8266.

Các chân nguồn 5V và GND được lấy từ vi điều khiển Arduino, nguồn điện AC là dòng điện trong sinh hoạt 220V. Sau đây là chi tiết các khối trong sơ đồ mạch.



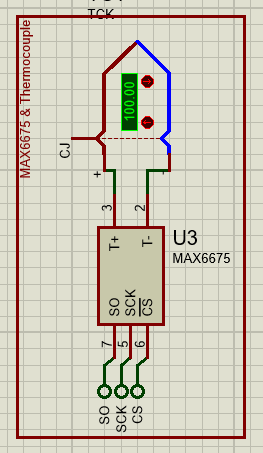
*Hình 3. 4. Khối nút nhấn*

\_ Khối nút nhấn bao gồm 2 nút nhấn. Một nút nhấn để điều khiển tăng nhiệt độ đặt, một nút dùng để điều khiển giảm nhiệt độ đặt, bước tăng giảm sẽ phụ thuộc vào lập trình cho vi điều khiển Arduino. Hai chân Increase và Decrease nối vào 2 pins của Arduino. Khi nút nhấn được nhấn, Chân Increase/Decrease sẽ lên mức 1, giá trị đó truyền về Arduino, xử lý tăng hoặc giảm giá trị đặt.



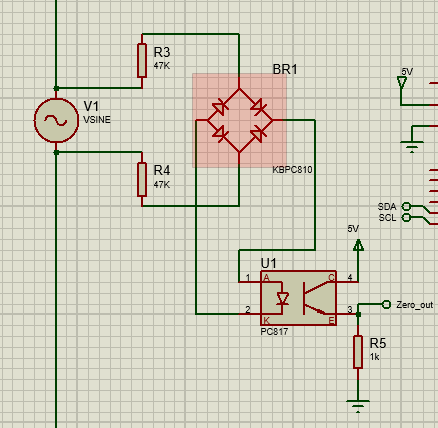
*Hình 3. 5. I2C & LCD*

\_ Một LCD thông thường để sử dụng thì phải sử dụng rất nhiều chân, như thế tốn rất nhiều tài nguyên. Vì vậy, để đơn giản và thuận tiện thì việc sử dụng I2C là lựa chọn tối ưu. Về nguyên lý hoạt động của I2C đã được trình bày ở Phần 2 mục VII. Nhiệm vụ của LCD đơn giản là nhận dữ liệu từ Arduino thông qua chân SDA, hiển thị lên LCD giá trị đặt setpoint và giá trị nhiệt độ hiện tại current temperature.



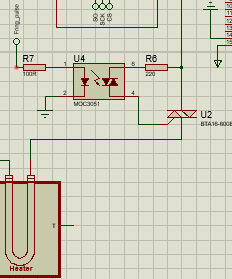
*Hình 3. 6. Thermocouple K & MAX6675*

\_ Thông thường, khi sử dụng một cảm biến nhiệt độ thì ta cần thiết kế một mạch khuếch đại, vì tín hiệu ở đầu ra cảm biến thường rất nhỏ. Nhưng với cảm biến Thermocouple loại K, có module đi kèm chính là MAX6675 đóng vai trò như một mạch khuếch đại. Với module này sẽ giúp rút ngắn thời gian thiết kế mạch, tiện lợi hơn trong quá trình làm việc.



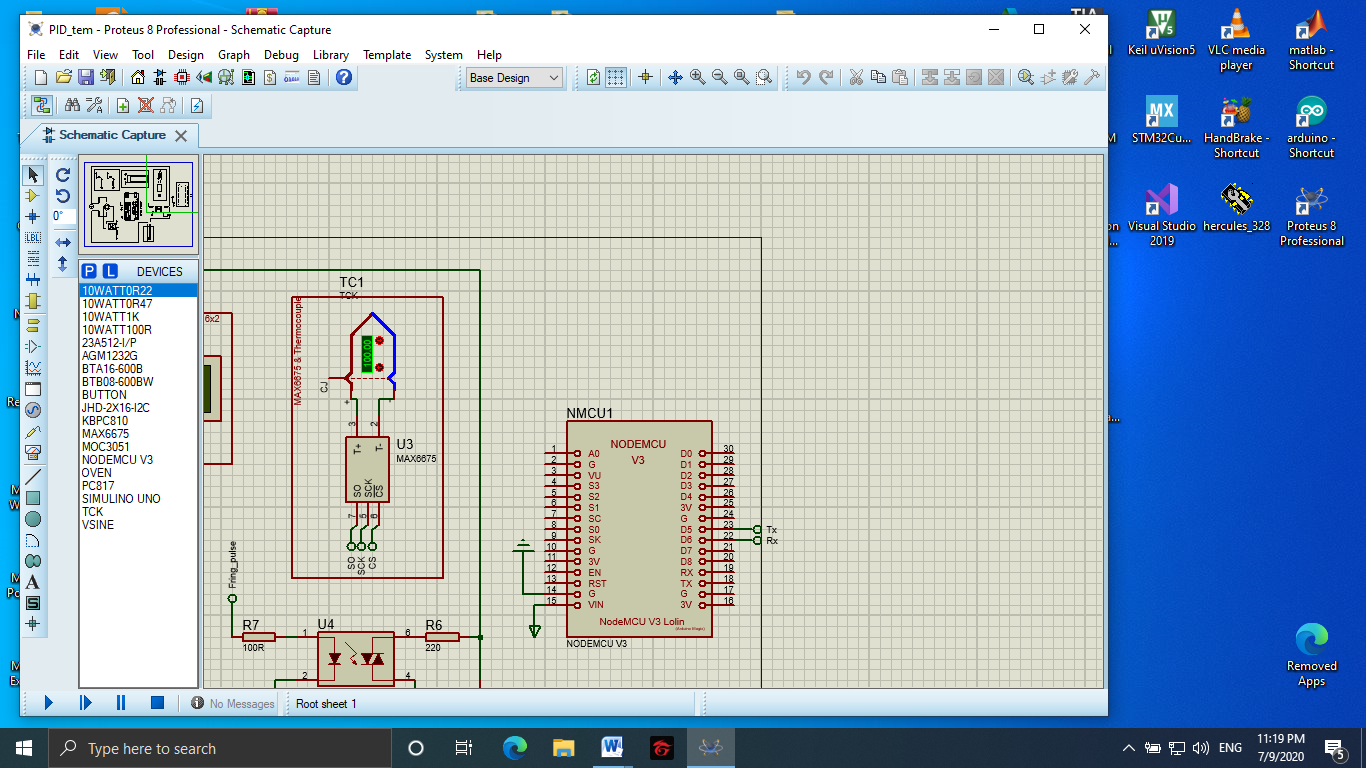
*Hình 3. 7. Mạch điều khiển*

\_ Mạch điều khiển này đóng vai trò là một mạch phát hiện điểm 0. Tại ngõ ra của photo coupler, ta sẽ được mức tín hiệu điện áp cao khi có điểm không. Dựa vào mạch phát hiện điểm 0 này, sẽ làm cơ sở cho mạch động lực

.

*Hình 3. 8. Mạch động lực*

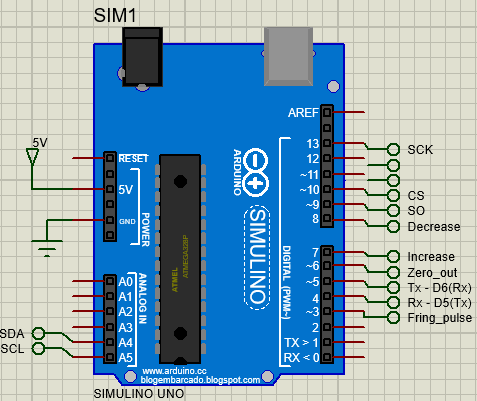
Khi tín hiệu phát hiện điểm 0 được truyền về vi điều khiển, vi điều khiển sẽ tính toán và xuất xung ra chân Fring\_pulse đóng vai trò làm xung kích cho triac, cấp điện áp cho heater. Ttheo vòng lặp đó, ta sẽ có được một xung PWM điều khiển lò nhiệt theo bộ điều khiển PID



*Hình 3. 9. NodeMCU ESP8266*

\_ Sử dụng hai chân D5 và D6 giao tiếp UART với Arduino kết hợp với sử dụng ngắt. Thông qua đó, tín hiệu sẽ được liên tục truyền và nhận theo chu kỳ mà không ảnh hưởng đến chương trình chính.

\_ Các chân được kí hiệu sẽ được nối vào các pins Arduino, nguồn 5V và Gnd cũng sẽ được lấy từ Arduino. Điện áp cấp cho Arduino sẽ được cấp thông qua cáp USB. Sơ đồ nối vào Arduino được thể hiện thông qua Arduino mô phỏng dưới đây.



*Hình 3. 10. Các chân được nối của Arduino*

# PHẦN 4: THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM:

## I. Yêu cầu cho phần mềm:

\_ Khai báo đầy dủ các biến cần sử dụng cho 2 chương trình cho Arduino và cho ESP8266. Tạo được biến real\_temperature để đọc giá trị nhiệt độ tù cảm biến

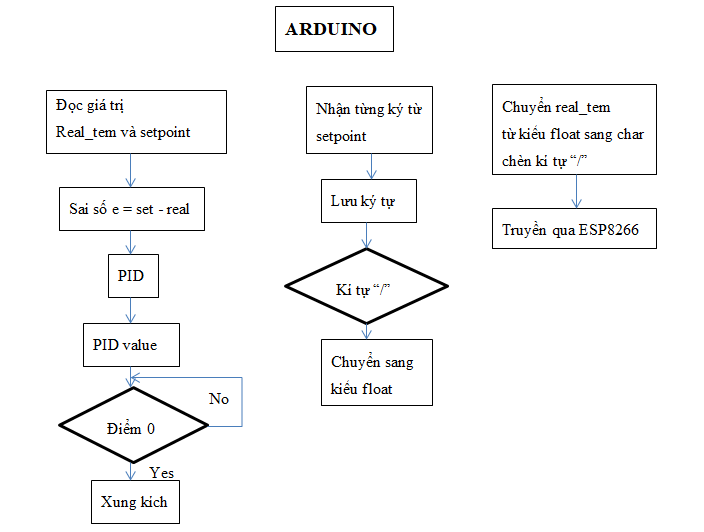
\_ Viết chương trình sử dụng thuật toán PID, sử dụng các hệ số P, I, D để điều khiển thành phần gia nhiệt của lò nhiệt.

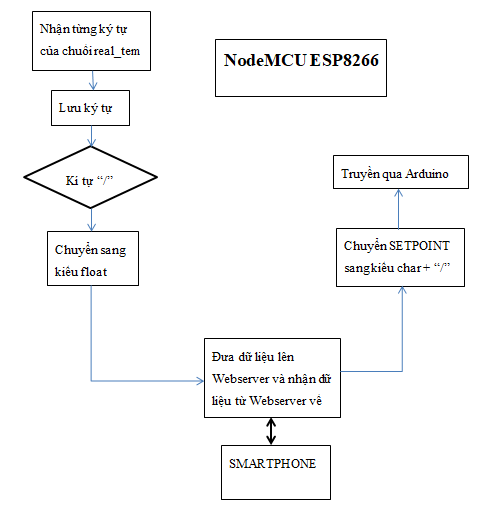
\_ Lập trình để điều khiển giá trị setpoint khi có tác động từ nút nhấn. Đồng thời, viết code để nhận tín hiệu điểm 0 khi mạch phát hiện được điểm 0. Từ tín hiệu đó, sẽ xuất ra được một tín hiệu ở chân Fring\_pulse làm xung kích, để tạo tín hiệu điều khiển cho thành phần gia nhiệt.

\_ Viết được chương trình giao tiếp giữa Arduino với NodeMCU, Arduino gởi được chuỗi real\_temperature cho MCU nhận và MCU gởi chuỗi setpoint cho Arduino.

\_ Viết chương trình giao tiếp giữa smartphone và NodeMCU thông qua module Wifi. Lập trình giao diện trên phần mềm Blynk để điều khiển hệ thống.

## II. Lưu đồ giải thuật





## III. Chương trình cho Arduino và NodeMCU ESP8266, lập trình giao diện Blynk.

Code Arduino Uno:

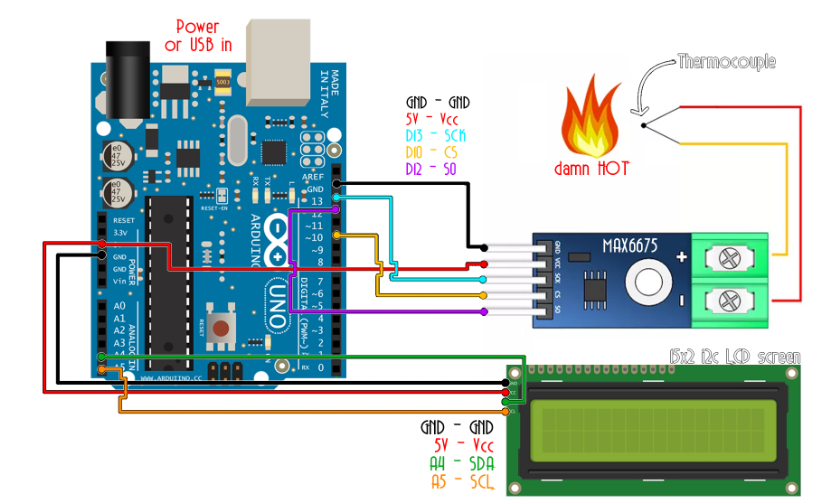
|  |
| --- |
| #include <SoftwareSerial.h>  #include "max6675.h"  #include <Wire.h>  #include <LiquidCrystal\_I2C.h>  LiquidCrystal\_I2C lcd(0x27,20,4);  SoftwareSerial SoftSerial(4,5); // RX,TX  //Inputs and outputs  int firing\_pin = 3; //xung kich  int increase\_pin = 8; //tang  int decrease\_pin = 7; //giam  int zero\_cross = 6; //diem 0  int thermoDO = 9;  int thermoCS = 10;  int thermoCLK = 13;  //Start a MAX6675 communication with the selected pins  MAX6675 thermocouple(thermoCLK, thermoCS, thermoDO);  //Variables  //int last\_CH1\_state = 0;  bool zero\_cross\_detected = false;  int firing\_delay = 7600;  int maximum\_firing\_delay = 7600;  unsigned long previousMillis = 0;  unsigned long currentMillis = 0;  int temp\_read\_Delay = 500;  float real\_temperature = 0;  int setpoint = 50;  bool pressed\_1 = false;  bool pressed\_2 = false;  //PID variables  float PID\_error = 0;  float previous\_error = 0;  int PID\_value = 0;  //PID constants  int kp =150; int ki=0.85; int kd = 0.001;  int PID\_p = 0; int PID\_i = 0; int PID\_d = 0;  String combineValue = ""; // Bien luu chuoi nhan tu nodemcu  String combineTemp = ""; // Bien luu chuoi gui nhiet do den nodemcu  int temp\_send\_Delay = 2000; // Gui nhiet do den nodemcu moi 2s  unsigned long timeUpdateStamp = 0; // Bien ho tro cho viec dung timer  void setup() {  //Define the pins    pinMode (firing\_pin,OUTPUT);  pinMode (zero\_cross,INPUT);  pinMode (increase\_pin,INPUT);  pinMode (decrease\_pin,INPUT);  lcd.init();  lcd.backlight();  Serial.begin(9600);  SoftSerial.begin(9600);  }  void loop() {    currentMillis = millis();  if(currentMillis - previousMillis >= temp\_read\_Delay){  //previousMillis += temp\_read\_Delay;  previousMillis = currentMillis;    real\_temperature = thermocouple.readCelsius();  previous\_error = PID\_error;  PID\_error = setpoint - real\_temperature;    PID\_p = kp \* PID\_error;  PID\_i = PID\_i + (ki \* PID\_error)\*temp\_read\_Delay/1000;  PID\_d = kd\*((PID\_error - previous\_error)/(temp\_read\_Delay/1000));  if(PID\_i > 7400)  {PID\_i = 7400;}    PID\_value = PID\_p + PID\_i + PID\_d;    if(PID\_value < 0)  { PID\_value = 0; }  if(PID\_value > 7400)  { PID\_value = 7400; }    //Printe the values on the LCD  lcd.clear();  lcd.setCursor(0,0);  lcd.print("Set: ");  lcd.setCursor(5,0);  lcd.print(setpoint);  lcd.setCursor(0,1);  lcd.print("Real temp: ");  lcd.setCursor(11,1);  lcd.print(real\_temperature);  }  if (zero\_cross\_detected)  {  delayMicroseconds(maximum\_firing\_delay - PID\_value); //This delay controls the power  digitalWrite(firing\_pin,HIGH);  delayMicroseconds(100);  digitalWrite(firing\_pin,LOW);  zero\_cross\_detected = false;  }  if(digitalRead(zero\_cross) == 1){  zero\_cross\_detected = true;  }  if(digitalRead(decrease\_pin) == 1){  while(digitalRead(decrease\_pin) == 1);  if (!pressed\_1)  {  setpoint = setpoint + 2;  delay(20);  pressed\_1 = true;  }  }  else if (pressed\_1)  {  pressed\_1 = false;  }  if(digitalRead(increase\_pin) == 1){  while(digitalRead(increase\_pin) == 1);  if (!pressed\_2)  {  setpoint = setpoint - 2;  delay(20);  pressed\_2 = true;  }  }  else if (pressed\_2)  {  pressed\_2 = false;  }  if(SoftSerial.available())  {  char tmp = SoftSerial.read();  if(tmp != '/')  combineValue += tmp;  else  {  Serial.println(combineValue);  float convertToFloat = combineValue.toFloat();  setpoint = convertToFloat;  combineValue = "";  }  }  if(millis() - timeUpdateStamp >= temp\_send\_Delay){  timeUpdateStamp = millis();  combineTemp += (String)real\_temperature + "/";  SoftSerial.print(combineTemp);  combineTemp = "";  }  } |

Code NodeMCU ESP8266

|  |
| --- |
| #include <SoftwareSerial.h>  #include <ESP8266WiFi.h>  #include <BlynkSimpleEsp8266.h>  SoftwareSerial SoftSerial(12,14); // RX,TX  //BlynkTimer timer;  char auth[] = "afzWIITHDFXKyMFmB39H-26AWK5mhXho";  char ssid[] = "HUAWEI nova 3i";  char pass[] = "1hai3bon";  float valueTempControl = 0;  boolean checkRecData = false;  String combineValue = "";  // Dong bo blynk va phan cung  BLYNK\_CONNECTED() {  Blynk.syncAll();  }  // Nhan gia tri tu bien V1 duoc gui xuong tu Blynk  BLYNK\_WRITE(V1)  {  valueTempControl = param.asFloat(); // lay bien ra va doi sang dang float  checkRecData = true;  }    void setup()  {  Serial.begin(9600);  SoftSerial.begin(9600);  Blynk.begin(auth, ssid, pass);  }    void loop()  {  Blynk.run();  //timer.run();  if(checkRecData == true)  {  checkRecData = false;  String tmp = "";  tmp = (String)valueTempControl + '/';  SoftSerial.print(tmp);  Serial.println(tmp);  }  //Gui den blynk  if(SoftSerial.available())  {  char tmp = SoftSerial.read();  if(tmp != '/')  combineValue += tmp;  else  {  Serial.println(combineValue);  float convertToFloat = combineValue.toFloat();  Blynk.virtualWrite(V0, convertToFloat);  combineValue = "";  }  }  } |

# PHẦN 5: KẾT QUẢ THỰC HIỆN

Đầu tiên, viết chương trình thử nghiệm Thermocouple K và MAX6675 để đảm bảo nhiệt độ thu được từ thermocouple hiển thị đúng lên LCD.



*Hình 5. 1. Sơ đô kết nối LCD và Thermocouple*

Khi đã tiến hành đọc được nhệt độ từ cảm biến, ta mới đi sâu vào thiết kế phần cứng. Các linh kiện cần để thiết kế mô hình điều khiển lò nhiệt.

Cầu diode KBPC1010

Photocoupler PC814

Opto MOC3020

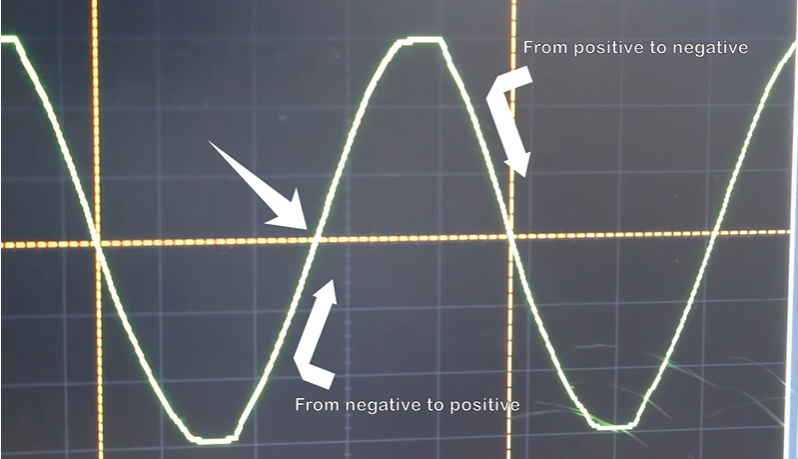
Triac BTA16

2 nút nhấn

Thermocouple K & MAX6675

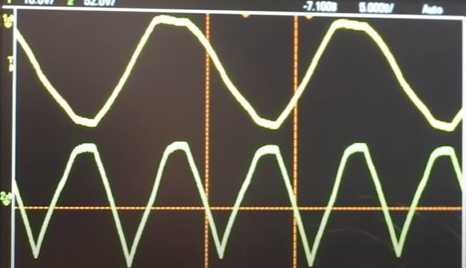
LCD & I2C

Điện áp AC 220V có dạng sóng sine, điểm 0 xuất hiện khi điện áp đổi từ âm sang dương hay từ dương sang âm.



*Hình 5. 2. Điểm 0 của sóng sine*

Đầu ra của cầu diode có dạng:



*Hình 5. 3. Điện áp ra cầu diode*

Tín hiêu ngõ ra của photocoupler, khi điện áp lớn hơn 5V thì điện áp sẽ ở mức 5V là mức cao.



*Hình 5. 4. Tín hiệu ngõ ra Photocoupler*

Dựa vào mạch phát hiện điểm 0, thiết kế mạch động lực để điều khiển thành phần gia nhiệt. Dưới đây là Tín hiệu điều khiển thành phần gia nhiệt.

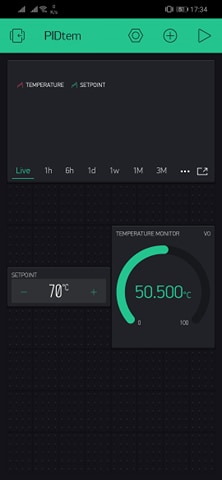


*Hình 5. 5. Xung kích và tín hiệu điều khiển*

Do thời gian có hạn, nên em không thể thực hiện phần cứng một cách chỉnh chu, dễ nhìn. Dưới đây là mô hình lò nhiệt đã được thiết kế



*Hình 5. 6. Mô hình lò nhiệt*

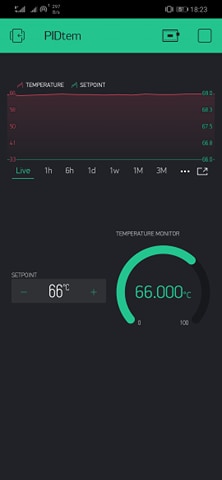


*Hình 5. 7. Giao diện điều khiển Blynk*

Kết quả thực hiện:



*Hình 5. 8. Kết quả chạy phần cứng*



*Hình 5. 9. Kết quả trên Blynk*

# PHẦN 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## I. Kết luận

Từ đồ án mà em đã tìm hiểu và thực hiện, giúp em hiểu rõ hơn về bộ điều khiển PID, cách thức tìm được các hệ số Kp, Ki, Kd. Đồng thời giúp hiểu hơn về cách thức làm việc của vi điều khiển, cũng như cách giao tiếp giữa 2 vi xử lý, biết thêm một số linh kiện và công dụng của nó.

Hơn hết là kiến thức về IoT, điều khiển một thiết bị, hệ thống thông qua wifi và điện thoại. Cách thức hoạt động của module Wifi ESP8266.

Ưu điểm của kết quả nghiên cứu lần này là mô hình không quá phức tạp, điều khiển dễ dàng. Có thể điều khiển thông qua Smartphone

Khuyết điểm là chưa có thể thi công mạch in để giúp mô hình gọn, dễ nhìn hơn. Thiết bị gia nhiệt chưa có thể gia nhiệt lên nhiệt độ quá cao.

## II. Hướng phát triển.

Theo kiến thức đã học, bộ PID có thể phát triển lên thành bộ điều khiển Fuzzy PID để có thể điều khiển một cách chính xác ở mọi nhiệt độ. Điều khiển không được chính xác ở nhiều nhiệt độ khác nhau chính là nhược điểm của bộ điều khiển PID.

Đề tài này có thể ứng dụng để thiết kế để điều khiển lò nhiệt, máy điều hòa, nò nung,…

# PHẦN 7: TÀI LIỆU THAM KHẢO

1. TS. Huỳnh Thái Hoàng, Slide bài giảng môn Cơ sở Tự động , trường Đại học Bách Khoa TPHCM.

2. Alan S.Morris, Reza Langari, “Measurement and Instrumentation, Second Edition: Theory àn Application”.

3. IoT Maker VietNam, “Arduino cho người mới bắt đầu”

4. Nguyễn Văn Nhờ, “Điện tử công suất 1”, NXB ĐHQG TP.HCM

5. <https://www.thermocoupleinfo.com/type-k-thermocouple.htm>

6. [http://arduino.vn/](http://arduino.vn/bai-viet/42-arduino-uno-r3-la-gi)

7. <https://www.alldatasheet.com/>