**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN ĐIỆN**



**ĐỒ ÁN 1**

**TÌM HIỂU VÀ THIẾT KẾ BỘ ĐIỀU KHIỂN ĐÈN LED**

**SỬ DỤNG VI ĐIỀU KHIỂN STM32**

**Giảng viên hướng dẫn:** TS. Nguyễn Mạnh Linh

**Sinh viên thực hiện:**

|  |  |
| --- | --- |
| Đinh Anh Tú | 20181805 |
| Lương Đình Long | 20181606 |
| Đinh Văn Sỹ | 20181737 |
| Trịnh Văn Huy | 20181539 |

**Hà Nội, 07/2021MỤC LỤC**

**Trang**

**LỜI NÓI ĐẦU** 3

**CHƯƠNG 1: CƠ SỞ LÍ THUYẾT**

[1.1. Cấu tạo và nguyên lý chiếu sáng của đèn led 2](file:///C:\Users\Dell%20Precision%205510\Downloads\Báo%20cáo%20ĐA1.docx#_Toc76466261)

[1.2. Tổng quan phương pháp băm xung](file:///C:\Users\Dell%20Precision%205510\Downloads\Báo%20cáo%20ĐA1.docx#_Toc76466262) PWM……………………………………….

[1.3. Nguyên lý bộ băm xung áp (Chopper)](file:///C:\Users\Dell%20Precision%205510\Downloads\Báo%20cáo%20ĐA1.docx#_Toc76466263)

**CHƯƠNG 2: MÔ HÌNH HOÁ LÝ THUYẾT**

2.1. Hàm truyền của hệ thống……………………………………………………….

2.2. Thiết kế bộ điều khiển PI theo phương pháp tối ưu độ lớn

**CHƯƠNG 3: THIẾT KẾ PHẦN CỨNG**

3.1. MOSFET DRIVER

3.2. Khối vi điều khiển

3.3. Khối nguồn

3.4. Các cảm biến

3.4.1 Cảm biến ánh sáng

3.4.2 Cảm biến dòng INA169

**CHƯƠNG 4: LẬP TRÌNH KHỐI VI ĐIỀU KHIỂN**

4.0 Giới thiệu, công cụ lập trình

4.1 Thuật toán chương trình chính

4.2 Thực hiện bộ điều khiển PI số

4.3 Điều chế độ rộng xung PWM

4.3.1 Cấu hình

4.3.2 Lập trình thuật toán

4.4. Đọc dữ liệu từ cảm biến dòng INA169

4.4.1. Cấu hình vi điều khiển qua STM32CubeMX

4.4.2 Đọc dữ liệu từ cảm biến

4.5 Đọc dữ liệu từ quang điện trở

4.5.1 Cấu hình vi điều khiển

4.5.2 Đọc dữ liệu từ quang điện trở

4.6 Mở rộng

**CHƯƠNG 5: KẾT QUẢ ĐẠT ĐƯỢC KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

**LỜI NÓI ĐẦU**

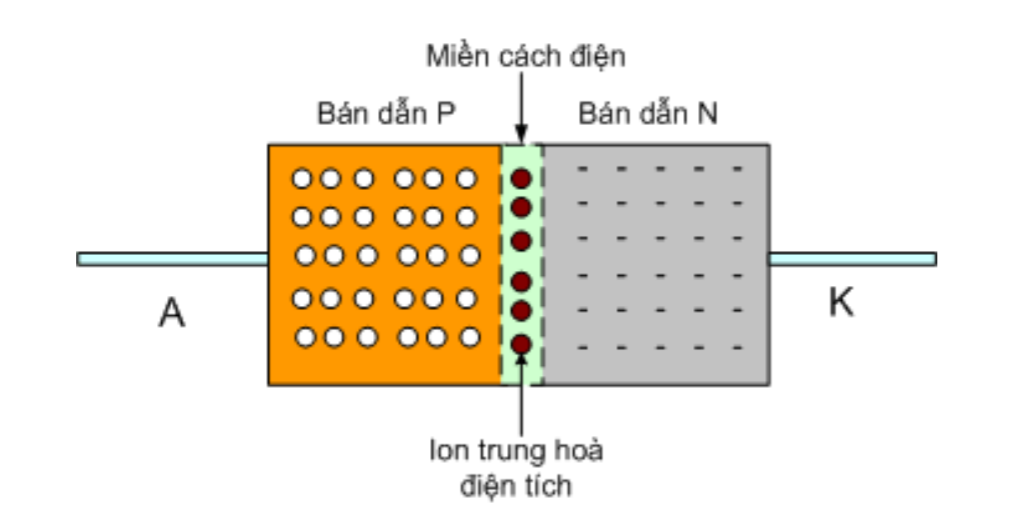
Ngày nay, việc sử dụng đèn led đã trở nên cực kỳ phổ biến và trở thành một phần không thể thiếu của cuộc sống con người. Với các ưu điểm: ánh sáng lớn, độ bền cao và tiêu tốn ít điện năng hơn các loại đèn khác, đèn led được ứng dụng rộng rãi trên các lĩnh vực như chiếu sáng sinh hoạt, nông nghiệp, giao thông hàng hải, y tế và đặc biệt là quảng cáo và trang trí,… Cùng với xu thế tự động hóa hiện nay, việc điều khiển các hệ thống led trong các lĩnh vực một cách tự động là một vấn đề cần thiết.

Nắm bắt được xu thế đó, cùng với sự phân công, hướng dẫn của TS. Nguyễn Mạnh Linh, nhóm đồ án I chúng em đã thực hiện đề tài Thiết kế bộ điều khiển đèn đường (đèn led) sử dụng vi điều khiển STM32. Nhóm chúng em xin cảm ơn GVHD.TS.Nguyễn Mạnh Linh đã giúp đỡ và hỗ trợ để chúng em hoàn thành được đồ án này. Do kiến thức còn hạn chế nên bài báo cáo cũng còn nhiều thiếu sót, chúng em mong các thầy bỏ qua và có thể góp ý để chúng em kịp thời sửa chữa.

Với mục đích nghiên cứu học tập và thử nghiệm với mô hình nhỏ. Nhóm đãtìm hiểu bộ điều khiển với yêu cầu sử dụng bộ nguồn ắc quy 0 - 12V, điều khiển dàn led với công suất thỏa mãn 9W.

**CHƯƠNG 1: CƠ SỞ LÍ THUYẾT**

## Cấu tạo và nguyên lý chiếu sáng của đèn led:



LED là từ viết tắt của **Light Emitting Diode** hay điốt phát quang là một linh kiện điện tử dựa trên chuyển tiếp p-n. LED có cấu trúc cơ bản của một điốt. Cấu trúc cơ bản của LED gồm hai lớp bán dẫn p, n ghép với nhau qua lớp tiếp xúc công nghệ. Hoạt động của LED dựa trên hoạt động của chuyển tiếp p-n. Một khi có dòng điện tác động lên biên giới của 2 bên mặt tiếp giáp thì một số điện tử bị lỗ trống thu hút, sát lại gần nhau hơn.Sau đó, chúng sẽ có xu hướng kết hợp với nhau để tạo thành nguyên tử trung hòa. Sau khi quá trình đó diễn ra sẽ tạo nên hiện tượng giải phóng năng lượng dưới ánh sáng thông qua các lớp bảo vệ. Nó sẽ định hướng bề mặt của đèn mà ánh sáng chiếu ra ngoài theo hướng đã được định sẵn.

Đồ án sử dụng led luxeon với các thông số kỹ thuật:

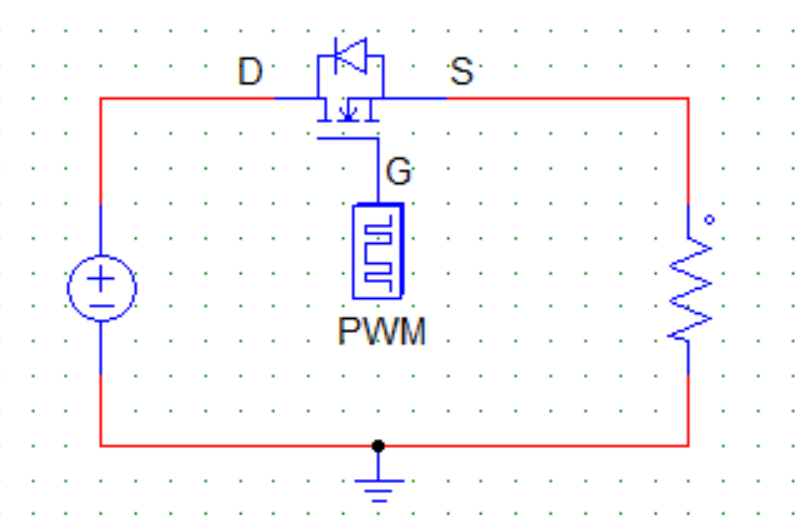


* Công suất: 1W (W)
* Nhiệt độ màu: Tùy chọn
* Độ sáng: 80-90Lm
* Góc nhìn: 180 độ
* Điện áp (VF): 3.15-3.4V
* Dòng đỉnh: (IF): 350mA
* Nhiệt độ hoạt động -20-60 độ
  1. **Tổng quan phương pháp băm xung PWM:**

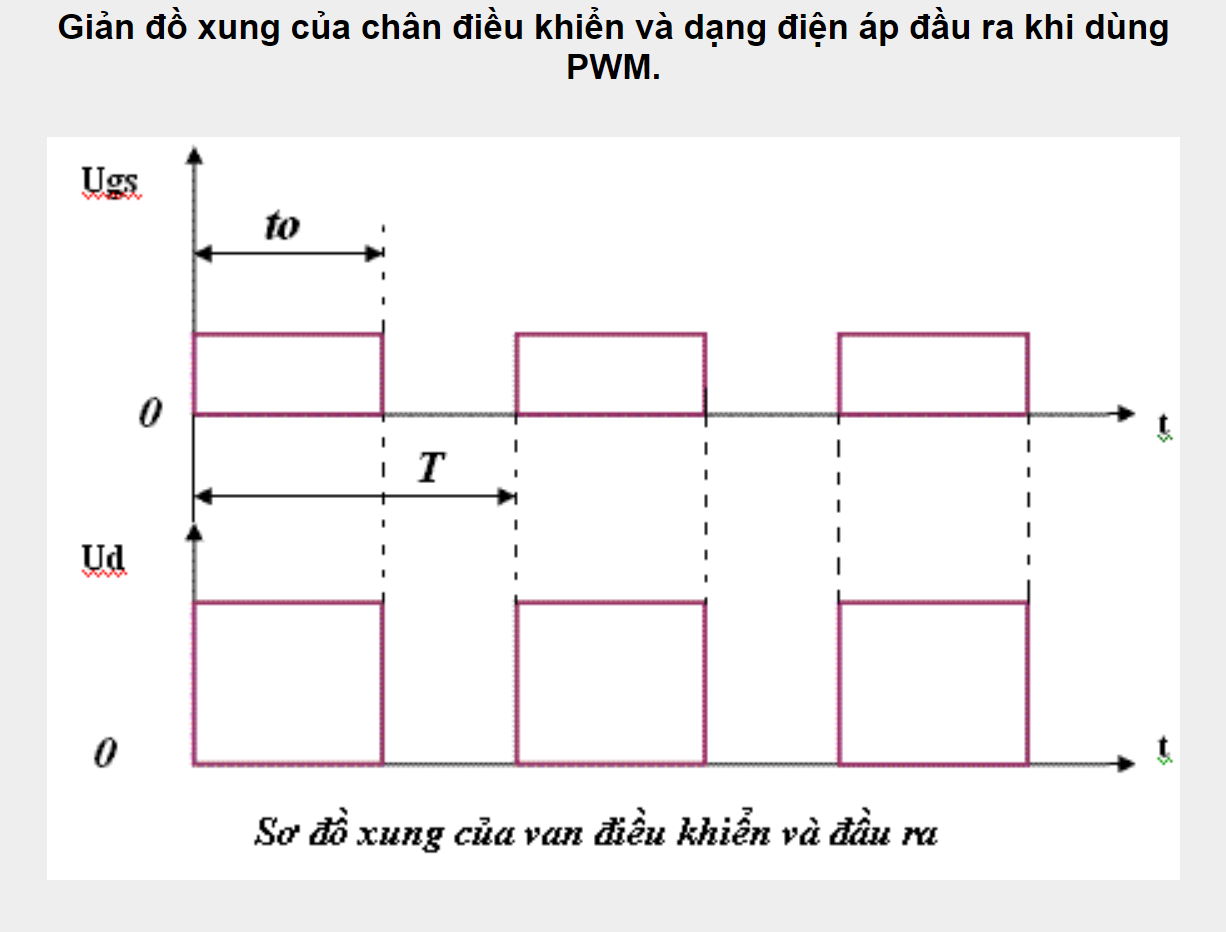
Phương pháp điều chế PWM có tên tiếng anh là Pulse Width Modulation là phương pháp điều chỉnh điện áp ra tải hay nói cách khác là phương pháp điều chế dựa trên sự thay đổi độ rộng của chuỗi xung vuông dẫn đến sự thay đổi điện áp ra. PWM được ứng dụng nhiều trong điều khiển. Điển hình nhất mà chúng ta thường hay gặp là điều khiển động cơ và các bộ xung áp, điều áp...

Điều đặc biệt là PWM chuyên dùng để điều khiển các phần tử điện tử công suất có đường đặc tính là tuyến tính khi có sẵn 1 nguồn 1 chiều cố định. Các PWM khi biến đổi thì có cùng 1 tần số và khác nhau về độ rộng của sườn dương hay sườn âm.

Ứng dụng PWM trong điều khiển:

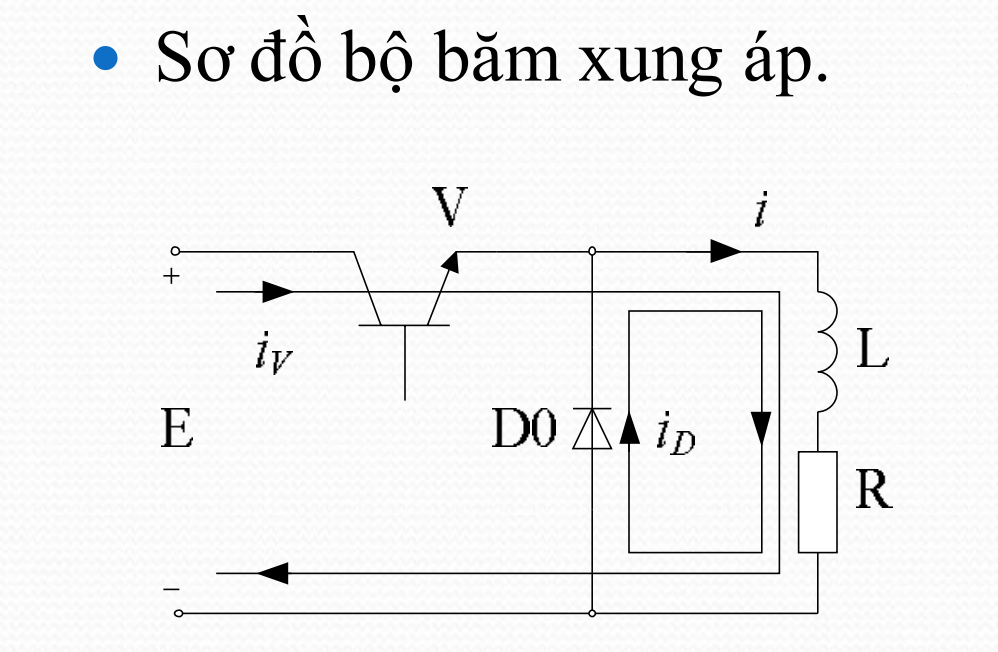


Xét hoạt động đóng cắt của một van bán dẫn, dùng van đóng cắt bằng Mosfet.



Để tạo được ra PWM thì hiện nay có hai cách thông dụng: Bằng phần cứng và bằng phần mềm. Ở đây chúng ta chỉ tập trung tìm hiểu về tạo xung PWM bằng phần mềm dựa vào CPU STM32.Đây là cách tối ưu trong các cách để tạo được xung vuông. Việc tạo bằng phần mềm cho độ chính xác cao về tần số PWM và mạch đơn giản hơn rất nhiều. Xung này được tạo dựa trên xung của CPU.

* 1. **Nguyên lý bộ băm xung áp (Chopper)**

****

Phần tử cơ bản là khoá điện tử V, là một van điều khiển hoàn toàn (GTO, IGBT, MOSFET, BJT), được mắc nối tiếp giữa tải và nguồn.

Điôt D0 có vai trò quan trọng trong sự hoạt động của sơ đồ, gọi là điôt không. Điôt này sẽ dẫn dòng tải khi V khoá.

* Khi V thông: iR+L = E
* Khi V khóa: iR+L = 0

Bộ băm xung áp có thể ứng dụng khiển dòng điện một chiều, ví dụ các bộ điều khiển kích từ cho máy phát đồng bộ, cho máy điện một chiều. Đặc biệt thích hợp cho điều khiển các máy điện một chiều công suất nhỏ.

Ở đây ta cùng xét tải trở cảm ở chế độ dòng liên tục:

Diagram

Description automatically generated

Trong đó: là dòng qua mạch khi V thông.

là dòng qua mạch khi V khóa.

**CHƯƠNG 2: MÔ HÌNH HOÁ LÝ THUYẾT**

**Chart, diagram, schematic

Description automatically generated**

**2.1. Hàm truyền của hệ thống:**

Ta có: dU= Ri+ L

Biến đổi laplace 2 vế ta được:

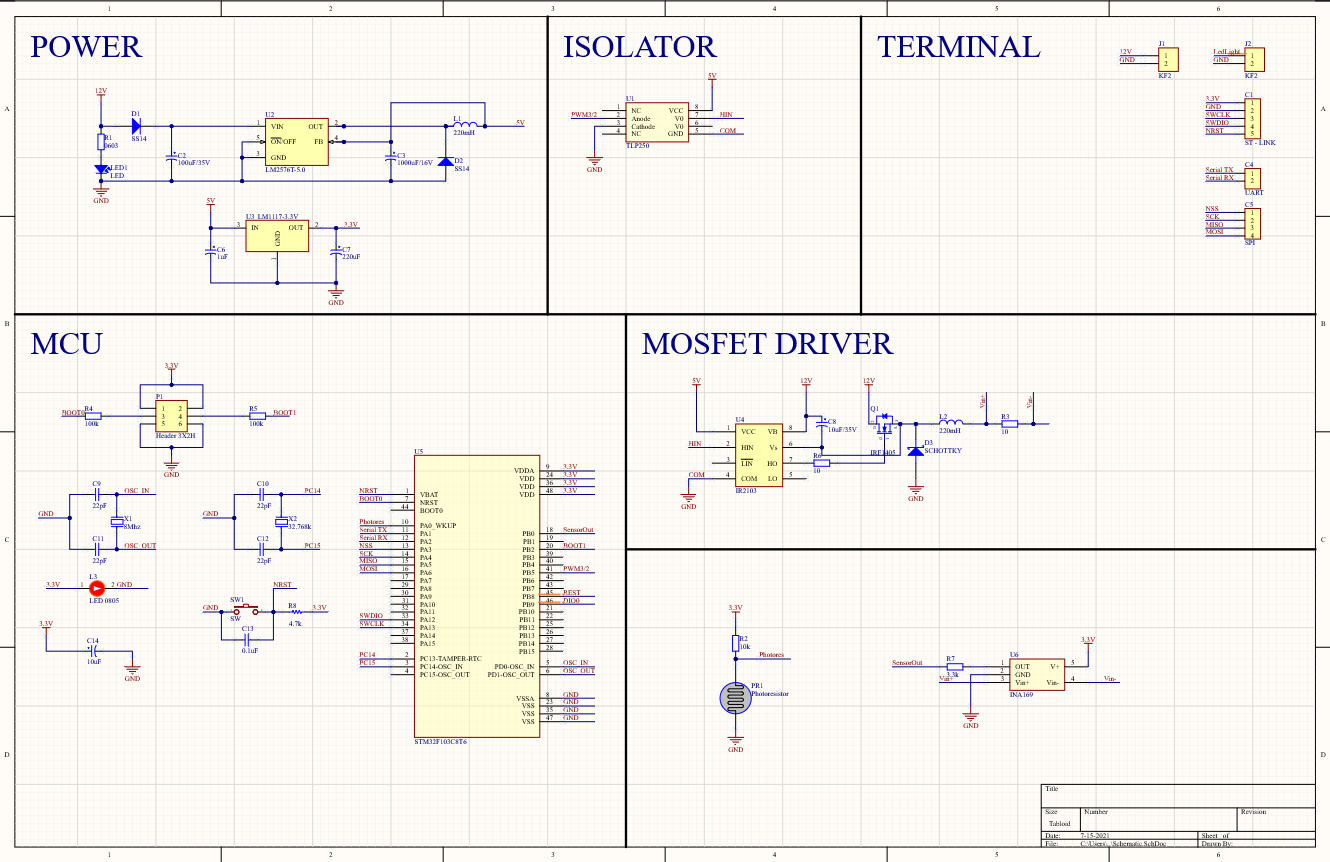
dU(s)= (R+Ls)I(s)

Hàm truyền G(s) của hệ thống sẽ là:

G(s)=I(s)/dU(s) = 1/(R+Ls)

**2.2. Thiết kế bộ điều khiển PI theo phương pháp tối ưu độ lớn**

**CHƯƠNG 3: THIẾT KẾ PHẦN CỨNG**

****

**3.1. MOSFET DRIVER**

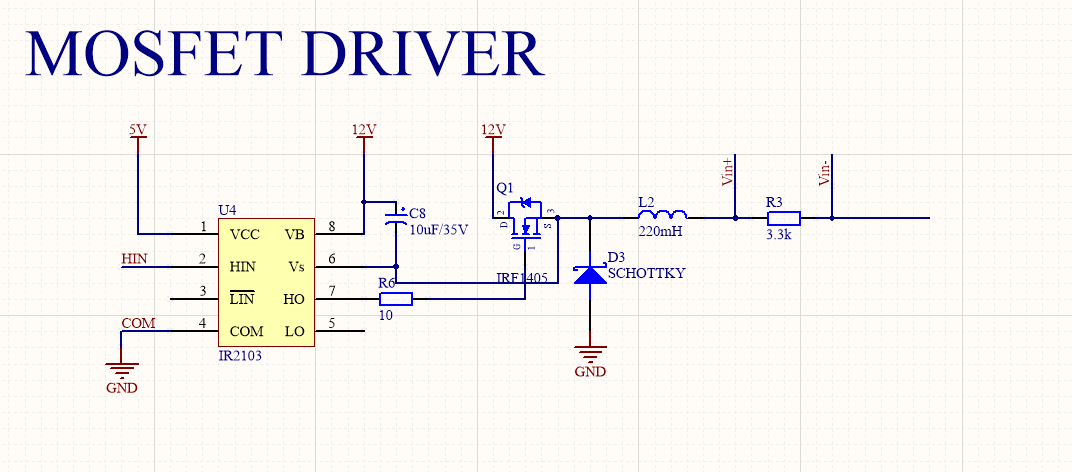
* **Tổng quan về mosfet:**

Mosfet (*Metal-Oxide Semiconductor Field-Effect Transistor*) là một transistor hiệu ứng trường đặc biệt có cấu tạo và hoạt động khác với transistor thông thường. Mosfet có nguyên tắc hoạt động dựa trên hiệu ứng từ trường để tạo ra dòng điện, là linh kiện có trở kháng đầu vào lớn thích hợp cho khuếch đại các nguồn [tín hiệu](https://bkaii.com.vn/tin-tuc/302-khai-niem-va-phan-loai-tin-hieu) yếu. Mosfet có khả năng đóng nhanh với dòng điện và điện áp khá lớn nên nó được sử dụng nhiều trong các bộ dao động tạo ra từ trường. Vì do đóng cắt nhanh làm cho dòng điện biến thiên. Nó thường thấy trong các bộ [nguồn xung](https://bkaii.com.vn/tin-tuc/484-nguon-xung-la-gi-nhung-tim-hieu-co-ban) và cách mạch điều khiển điện áp cao.

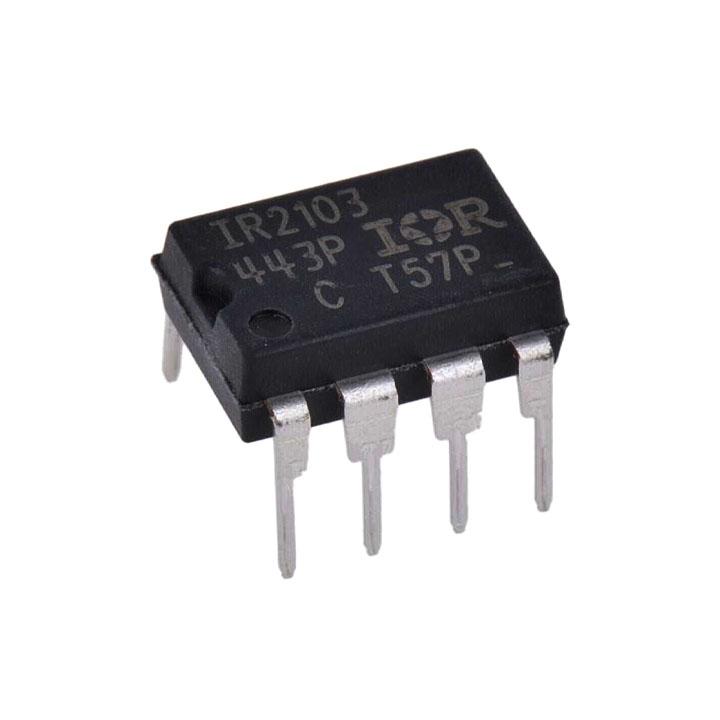
* **Thông số kỹ thuật mosfet IRF1405:**

****

* Phân loại: MOSFET N-CH (kênh N)
* Dòng điện tối đa: 169A
* Điện áp đánh thủng: 55V
* Điện áp VGS: ± 20V
* RDS(on): 5.3mΩ
* Nhiệt độ hoạt động: - 55oC đến 75oC

****

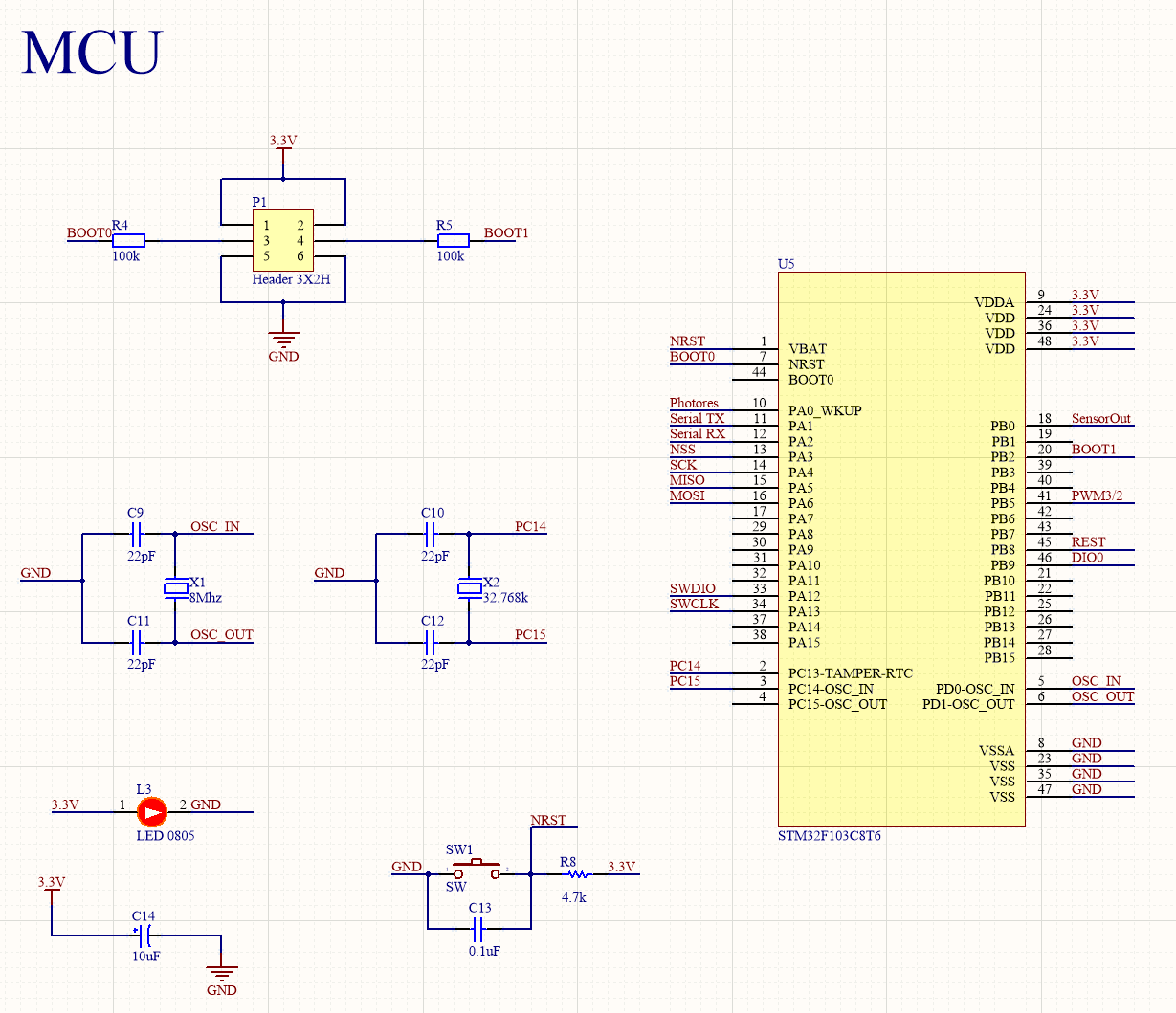
Để vi điều khiển có thể điều khiển được mosfet chính xác cũng như đảm bảo an toàn, tránh xảy ra sự cố trong quá trình điều khiển, chúng tôi sử dụng IC mosfet driver. Trình điều khiển mosfet - mosfet driver (gate driver) - là bộ khuếch đại công suất cung cấp điện áp đầu ra cao và thấp để bật hoặc tắt mosfet. Chúng thường được dùng để điều khiển mosfet công suất trong các ứng dụng biến tần (inverter) và chuyển đổi (converter). Ở đây ta sử dụng IC IR2103 (Half Bridge Drive) kết hợp với mạch boostrap có các thông số kỹ thuật như sau:



* Điện áp max: 600V
* Điện áp điều khiển: 10V - 20V
* Dải nhiệt độ hoạt động: - 40°C - 105°C
* Thời gian ON: t = 680ns
* Thời gian OFF: t = 150ns
* Thời gian trễ: 520ns
* Kiểu chân: DIP 8 / SOP 8
* Dùng để thiết kế mạch cầu H

**3.2. Khối vi điều khiển**

Trong đề tài này khối vi điều khiển được sử dụng với mục đích chính để điều khiển bốn động cơ, thu thập dữ liệu, gửi dữ liệu qua lại giữa các sensor với máy tính , dữ liệu encoder từ động cơ bánh xe với máy tính và ngược lại truyền tín hiệu điều khiển xuống động cơ.





Chúng em đang sử dụng vi điều khiển STM32F103C8T6 là vi điều khiển 32 bit của hãng STMicroelectronics.

Thông số kỹ thuật của vi điều khiển:

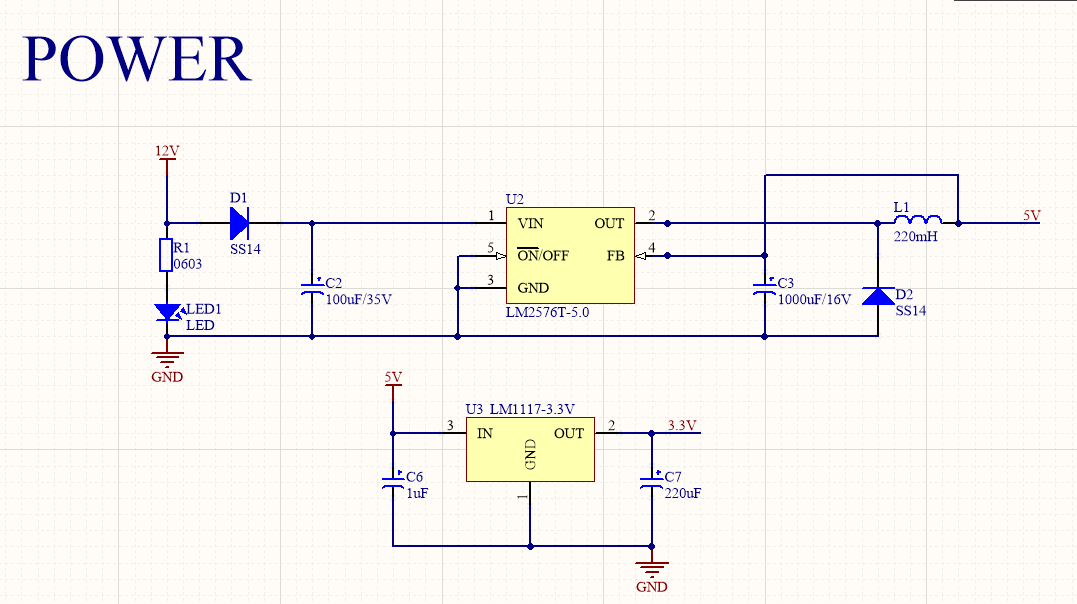
* Điện áp hoạt động: 2.0 -3.6V
* 256 KB flash memory và 48KB SRAM
* Tần số CPU tối đa: 72 MHz.
* 112 GPIO
* Hỗ trợ 3 module ADC 12 bit.
* Có 8 bộ Timer .
* Hỗ trợ các chuẩn giao tiếp : UART , SPI , I2C , CAN.

Trong đồ án chúng tôi sử dụng vi điều khiển STM32F103C8T6 với cấu hình như sau:

* Tần số hoạt động 24MHz
* Sử dụng 2 kênh ADC 12bit có ngắt
* Các GPIO ở chế độ ngắt
* Các GPIO mở rộng với các chuẩn giao tiếp UART, SPI, I2C cho khả năng phát triển sau này

**3.3. Khối nguồn:**

Để cấp nguồn hoạt động cho vi điều khiển, mosfet driver và các thành phần khác, chúng tôi sử dụng hai mạch Buck Converter như hình vẽ:



* Buck Converter 12V - 5V sử dụng IC ổn áp LM2576 cấp nguồn cho mosfet driver IR2103.
* Buck Converter 5V - 3,3V sử dụng IC ổn áp LM1117 để cấp nguồn hoạt động cho vi điều khiển.

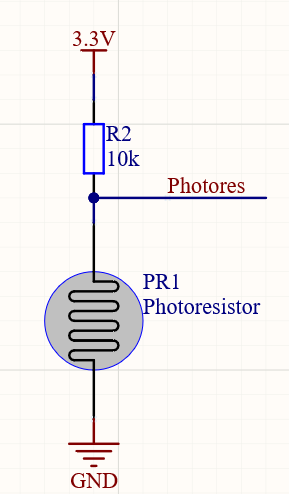
**3.4. Các cảm biến**

**3.4.1 Cảm biến ánh sáng:**

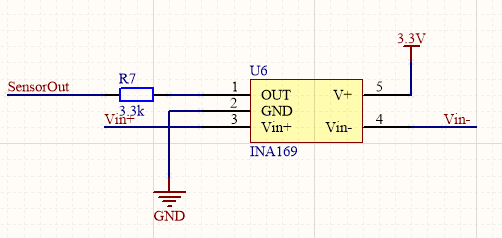
[GL5528 Quang Trở](https://www.thegioiic.com/products/gl5528-quang-tro-5mm) có đường kính 5mm, là một điện trở được làm bằng vật liệu bán dẫn và độ dẫn thay đổi theo sự thay đổi của ánh sáng. [Điện trở quang](https://www.thegioiic.com/product/quang-tro-cds-cells) được sử dụng rộng rãi trong nhiều ngành công nghiệp, chẳng hạn như đồ chơi, đèn, máy ảnh,...

* Thông số kỹ thuật:
* Điện áp max: 250 VDC
* Công suất max: 200mW
* Giá trị đỉnh phổ (Spectrum peak value): 540nm
* Trở kháng ánh sáng (10Lux): 10 ~ 20 (KΩ)
* Trở kháng bóng tối: 2 (MΩ)
* Nhiệt độ môi trường: -30 ~ +70oC
* γ value (1000 | 10): 0.6
* Thời gian phản hồi (ms):

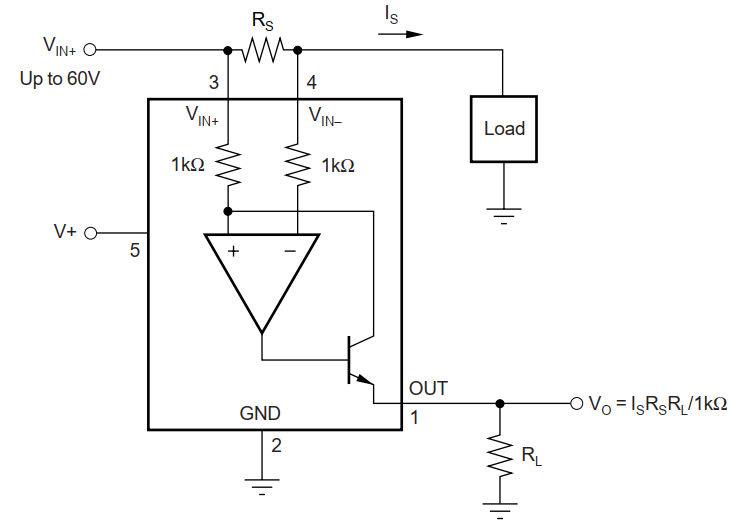
Điện áp của cảm biến ánh sáng được phân áp qua điện trở R = 10kΩ. Khi cường độ ánh sáng thay đổi, quang điện trở có giá trị thay đổi từ 10kΩ đến 2MΩ, như vậy, điện áp đầu ra của cảm biến trước khi đi vào bộ ADC của vi điều khiển sẽ có giá trị Vout = 1,65 - 3,3V, tương thích với điện áp hoạt động của bộ ADC

****

**3.4.2 Cảm biến dòng INA169**

****

* Thông số cảm biến dòng INA169 của Texas Instruments
* Điện áp hoạt động: 2.7 – 75VDC, có thể kết nối với nguồn cần đo hoặc nguồn riêng.
* Điện áp đầu vào đo dòng: ≤ 75VDC
* Dòng tối đa đo được: 5A (có thể thay đổi dựa vào giá trị điện trở sun)
* Ngõ ra dòng, đã tích hợp trở đổi thành áp.
* Có thể chọn điện trở ngõ ra từ 1k – 100k
* Cách tính dòng cần đo: Is = (Vout \* 1kΩ) / (Rs \* Rl)
* Is: dòng cần đo
* Vout: điện áp đo được từ ngõ ra
* 1KΩ: hằng số mặc định
* Rs: điện trở SHUNT
* Rl: điện trở ngõ ra
* Kích thước: 24 x 20mm

****

Dựa trên sơ đồ trên, để đo được dòng điện Is < 600mA, ta sử dụng một điện trở Shunt Rs = 1 (điện trở được lựa chọn giá trị nhỏ để tránh tổn thất, sụt áp), điện trở RL = 330Ω thì đầu ra của cảm biến dòng sẽ là:

tương thích với điện áp đầu vào ADC của vi điều khiển.

**CHƯƠNG 4: LẬP TRÌNH KHỐI VI ĐIỀU KHIỂN**

**4.0 Giới thiệu, công cụ lập trình**

Yêu cầu đặt ra cho vi điều khiển là nhận dữ liệu từ các cảm biến, từ module truyền thông, từ đó điều khiển độ sáng của bóng đèn led. Như vậy, cần lập trình cho vi điều khiển thực hiện các hành động sau:

* Thực hiện bộ điều khiển PI số
* Điều khiển Mosfet thông qua Điều chế độ rộng xung PWM
* Tính toán độ sáng ngoài trời thông qua dữ liệu nhận được từ quang điện trở, từ đó đặt ra giá trị đặt (setpoint cho bộ điều khiển PI số)
* Đọc dữ liệu từ cảm biến dòng INA169

Phần mềm được sử dụng để lập trình cho vi điều khiển STM32F103C8T6 STM32CubeMX của ST và KeilC của ARM. Mạch nạp cho vi điều khiển là ST Link V2.

Logo, company name

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Text, whiteboard

Description automatically generated

Đồ án sử dụng thư viện HAL (Hardware Abtraction Layer). Đây là thư viện do chính ST Electronics tạo ra nhằm mục đích dễ dàng lập trình các ứng dụng trên vi điều khiển của ST, đồng thời thuận tiện chuyển đổi giữa các dòng vi điều khiển khác nhau. Sau khi cấu hình cho vi điều khiển trên phần mềm STM32CubeMX, code được tự động sinh ra với các cấu hình trên dưới dạng các hàm trong thư viện HAL, sử dụng để lập trình cho vi điều khiển trên phần mềm Keil C ARM.

**4.1 Thuật toán chương trình chính**

Để thực hiện được các hành động đã đặt ra cho vi điều khiển STM32F103C8T6, ta cần lập trình chương trình có lưu đồ thuật toán tổng quát như hình.

Diagram

Description automatically generated

Nhiệm vụ chính của chương trình là điều khiển dòng điện đi vào ma trận LED, từ đó điều khiển độ sáng ma trận LED. Thuật toán chính bắt đầu từ việc tính toán tín hiệu vào cho bộ điều khiển PI:

* Giá trị đặt (Setpoint) của bộ điều khiển được tính toán dựa trên cường độ ánh sáng ngoài trời. Ánh sáng ngoài trời càng lớn thì giá trị đặt cho dòng điện đi tới ma trận LED càng cao. Ngoài ra, giá trị đặt của bộ điều khiển còn phụ thuộc vào yêu cầu của người dùng (tắt, bật, sáng yếu)
* Sau khi có giá trị đặt, bộ điều khiển PI sẽ tính toán để tạo ra giá trị duty cycle của xung PWM đi vào mosfet driver.
* Vi điều khiển tạo ra xung PWM với duty cycle đã tính toán ở khâu trước. Xung PWM được tạo ra với tần số xác định và ở GPIO kết nối với đầu vào của mosfer driver
* Cuối cùng là đo dòng điện sinh ra, phản hồi lại để tính toán sai lệch so với giá trị đặt.

**4.2 Thực hiện bộ điều khiển PI số**

Bộ điều khiển PI sau khi được số hoá bằng biến đổi Z được đưa vào chương trình để thực hiện.

**4.3. Điều chế độ rộng xung PWM**

**4.3.1. Cấu hình**

* Cấu hình GPIO PB5 chế độ điều chế độ rộng xung PWM3/2, sử dụng Timer3, Channel 2 (TIM3\_CH2)
* Timer3:

+ Tần số: 24MHz

+ Prescaler: 24

+ Counter Mode: Up

+ Counter Period: 200

**4.3.2. Lập trình thuật toán**

* Sử dụng hàm HAL\_TIM\_PWM\_Start(&htim3,TIM\_CHANNEL\_3); để khởi động Timer3, chế độ PWM
* Hàm\_\_HAL\_TIM\_SetCompare(&htim3,TIM\_CHANNEL\_3, duty\_cycle\*200), trong đó biến duty\_cycle (uint8\_t) là giá trị duty cycle (%) của xung PWM muốn tạo ra

**4.4. Đọc dữ liệu từ cảm biến dòng INA169**

**4.4.1. Cấu hình vi điều khiển qua STM32CubeMX**

* Cấu hình GPIO PB0 ở chế độ ADC1 kênh IN8:

**+** Độ phân giải 12bits, tương đương với 4096 giá trị

**+** Tần số hoạt động 24MHz: Thời gian cho một lần so sánh là

từ đó, thời gian tối đa cho một lần chuyển đổi là 1,7.10-4 s

**+** Chế độ hoạt động: Chế độ chuyển đổi liên tục (Continous Conversion Mode), sau khi thực hiện xong một lần chuyển đổi, bộ ADC chuyển sang một lần chuyển đổi mới

**+** Giá trị thu được căn lề trái (Right alignment): Do giá trị thu được từ ADC là 12bit nên cần sử dụng một thanh ghi 16bit để lưu, ở đây chọn căn lề phải

**+** Sau mỗi lần chuyển đổi, ADC sẽ thực hiện một ngắt

**4.4.2 Đọc dữ liệu từ cảm biến**

Để khởi động ADC có ngắt, sử dụng hàm: HAL\_ADC\_Start\_IT(&hadc1);

Để lưu giá trị đo được từ ADC, ta khởi tạo một biến adc1\_value (Kiểu dữ liệu là uint16\_t). Sử dụng hàm

HAL\_ADC\_ConvCpltCallback(ADC\_HandleTypeDef\* hadc)

là hàm phục vụ ngắt cho ADC.

Sau khi ADC thực hiện xong một lần chuyển đổi, chương trình gọi tới hàm phục vụ ngắt. Hàm phục vụ ngắt sẽ đọc giá trị đo được của ADC vào biến adc1\_value:

adc1\_value = HAL\_ADC\_GetValue(&hadc1);

**4.5 Đọc dữ liệu từ quang điện trở**

**4.5.1 Cấu hình vi điều khiển**

* Cấu hình GPIO PA5 ở chế độ ADC2 kênh IN5
* Do cường độ ánh sáng trong ngày thay đổi chậm nên để tiết kiệm năng lượng, ADC2 sẽ không hoạt động ở chế độ liên tục (Continous Conversion Mode), như vậy ADC2 sẽ dừng hoạt động sau khi chuyển đổi xong một lần
* Các đặc tính còn lại được cấu hình tương tự như bộ ADC1

**4.5.2 Đọc dữ liệu từ quang điện trở**

Cường độ ánh sáng trong ngày thay đổi chậm, vì vậy, chúng tôi lập trình cho ADC2 hoạt động sau mỗi một giờ đồng hồ. Khi đến thời gian cần hoạt động, để khởi động bộ ADC2, gọi hàm:

HAL\_ADC\_Start\_IT(&hadc2);

Giá trị thu được sau khi ADC2 chuyển đổi xong được lưu vào một biến nguyên 16bit (uint16\_t) adc2\_value

**4.6 Mở rộng**

Bộ điều khiển đèn led được thiết kế hoạt động tự động, tuy nhiên, để thuận tiện trong quá trình sử dụng thực tế, các chân PA7, PA8, PA9 của vi điều khiển STM32F103C8T6 được lập trình ngắt với chức năng: Tắt đèn, bật đèn, bật đèn với độ sáng thấp.

Ngoài ra, bộ điều khiển cũng để header các GPIO tương thích với các giao thức như UART, I2C, SPI để có thể mở rộng thêm các chức năng cho bộ điều khiển sau này một cách dễ dàng.

**CHƯƠNG 5: KẾT QUẢ ĐẠT ĐƯỢC**

**KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

**Những việc đã làm được:**

Sau một kì học thực hiện việc tìm hiểu về bộ điều khiển đèn led, lập trình cho khối vi điều khiển. Mặc dù là chỉ với thực hiện được với khối vi điều khiển nhưng đó cũng là kết quả mà nhóm em đã thực hiện được. Khối vi điều khiển đã đáp ứng phần nào so với mục đích đề ra lúc đầu là có thể điều khiển được bốn động cơ, đọc dữ liệu cảm biến và gửi lên PC. Tuy nhiên vẫn còn rất nhiều điều mà nhóm em còn thiếu sót và chưa cải thiện được như độ nhiễu của dữ liệu cảm biến, thuật toán thực thi.

**Hướng phát triển của đồ án trong tương lai:**

Đồ án 1 với mục đích tìm hiểu thiết kế bộ điều khiển đèn led đơn giản. Trong tương lai, có thể phát triển lên Đồ án 2, Đồ án thiết kế với một số tính năng như sau:

* Tích hợp thêm truyền thông qua Lora, Wifi, Zigbee,…
* Cảm biến người đi bộ
* Xử lý ảnh đo mật độ xe
* Điều khiển mạng, điều khiển bầy đàn hệ thống đèn chiếu sáng trong một khu vực nhất định

**TÀI LIỆU THAM KHẢO**

1. Carmine Noviello – “Mastering STM32”
2. Hoàng Minh Sơn – “Cơ sở điều khiển quá trình”