

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH



BÀI TẬP LỚN ĐIỆN TỬ ỨNG DỤNG
Đề tài: Thiết kế mạch đo dòng điện AC-DC

LỚP L01 --- NHÓM 12 --- HK231

NGÀY NỘP 02/01/2024

Giảng viên hướng dẫn: ThS. Nguyễn Trung Hiếu

Sinh viên thực hiện	Mã số sinh viên	Điểm số
Châu Ngọc Tình	2110588	
Hồ Lê Quốc Thắng	2114826	
Phan Thị Ngọc Yến	2015142	

Thành phố Hồ Chí Minh – 2024

MỤC LỤC

MỞ ĐẦU	2
I. CƠ SỞ LÝ THUYẾT.....	3
1. Điện trở Shunt	3
2. Mạch khuếch đại vi sai.....	4
II. PHẦN CỨNG.....	6
1. Thiết kế mạch đo lường	6
➤ Tính toán ảnh hưởng I_{iov} và I_{ib} :	7
2. Mạch thực tế	7
3. Kết quả đo	9
III. PHẦN MỀM	13
1. Lựa chọn số bit ADC và Vi điều khiển.....	13
2. Phương pháp đọc giá trị ADC.....	13

MỞ ĐẦU

Đo dòng điện, dù là dòng điện xoay chiều (AC) hay một chiều (DC), đóng vai trò quan trọng trong nhiều ứng dụng điện, điện tử. Thực tế, có nhiều phương pháp thiết kế mạch đo dòng khác nhau như phương pháp dựa trên hiệu ứng Hall, điện trở shunt, hiệu ứng tụ điện,... Ở đề tài này, phương pháp sử dụng điện trở shunt được áp dụng với các ưu điểm về độ chính xác, phạm vi dòng đo được,... Bên cạnh đó, đề tài cũng tập trung vào việc phát triển một mạch đo dòng đơn giản nhưng hiệu quả, sử dụng kiến thức cơ bản về Định luật Ohm và kiến thức về OPAMP để đo lường độ chính xác dòng điện AC và DC. Mục tiêu chính là xây dựng một thiết bị đo lường an toàn, linh hoạt, dễ sử dụng trong các ứng dụng thực tế, đồng thời giới thiệu các yếu tố thiết thực và thách thức kỹ thuật mà một mạch đo dòng đơn giản có thể phải đối mặt. Một khía cạnh quan trọng là thiết kế giao diện người dùng sao cho người sử dụng có thể dễ dàng tương tác với mạch đo.

I. CƠ SỞ LÝ THUYẾT

Mạch đo dòng điện bằng điện trở Shunt chủ yếu được sử dụng để đo lường và giám sát dòng điện trong các hệ thống điện công nghiệp và hệ thống điều khiển quy mô lớn. Đây là một thành phần quan trọng giúp kiểm soát, bảo vệ, và theo dõi hiệu suất của các thiết bị điện và hệ thống điện tử.

1. Điện trở Shunt

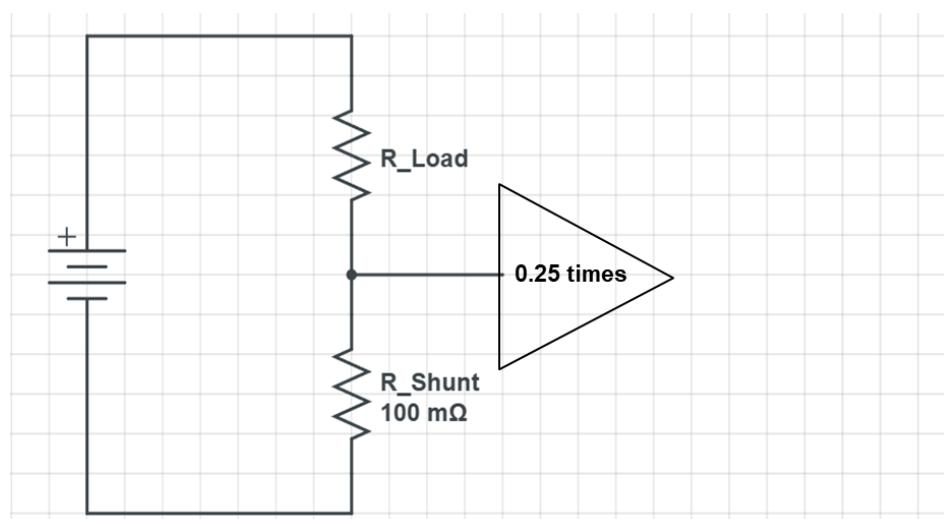
- **Định nghĩa:** Điện trở Shunt là một loại điện trở được kết nối song song với mạch, được thiết kế để tạo ra một điện áp rơi qua và dựa vào đó để đo lường dòng điện trong mạch. Thường được sử dụng để đo lường dòng điện lớn và giữ cho mức điện áp qua nó là ổn định.
- **Nguyên Tắc Hoạt Động:** Khi dòng điện đi qua điện trở Shunt, nó tạo ra một điện áp rơi qua theo công thức định luật Ohm

$$U = I * R$$

Với: U là điện áp rơi trên điện trở.

I là dòng điện chạy qua điện trở.

R là giá trị của điện trở shunt.



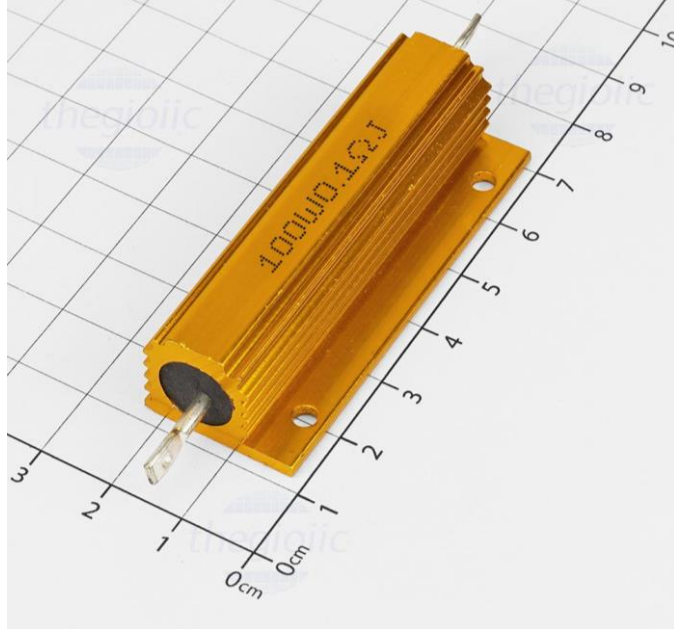
Hình 1. Sơ đồ nguyên lý mạch đo dòng điện

Ta có: $\Delta V = I_{Load} * R_{shunt}$ với R_{shunt} có giá trị 0.1Ω. Khi I_{Load} từ 0 - 20A, dẫn đến ΔV sẽ nằm trong đoạn từ 0 - 2V.

Ta có dòng điện chạy qua R_{shunt} từ 0 đến 20A, dẫn đến công suất tối đa chạy qua $R_{shunt} = 0.1\Omega$ được tính bằng công thức:

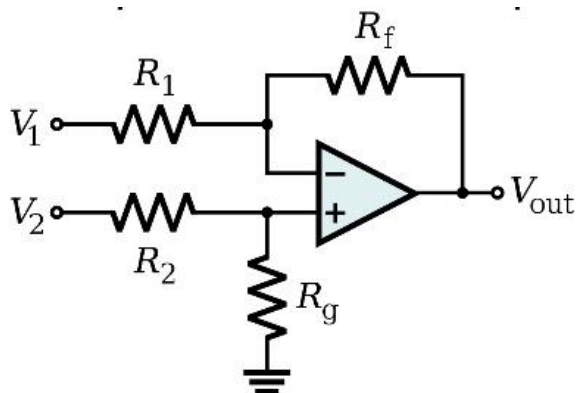
$$P = I^2 * R_{shunt}$$

Khi $I = 20A \rightarrow P = 40W \rightarrow$ Ta cần chọn điện trở có công suất lớn hơn 40W. Ta chọn điện trở công suất 0.1Ω 100W.



2. Mạch khuếch đại vi sai

Ta sử dụng một mạch khuếch đại có hệ số khuếch đại bé hơn 1 để ΔV nhỏ hơn giúp nằm trong khoảng đo của vi điều khiển.



Chứng minh:

Ta có:

$$V_+ = V_- = \frac{R_g}{R_2 + R_g} V_2$$

Lại có:

$$\frac{V_{out} - V_-}{R_f} = \frac{V_- - V_1}{R_1}$$
$$\Rightarrow V_{out} = \frac{(R_f + R_l)R_g}{(R_g + R_2)R_l} V_2 - \frac{R_f}{R_l} V_l$$

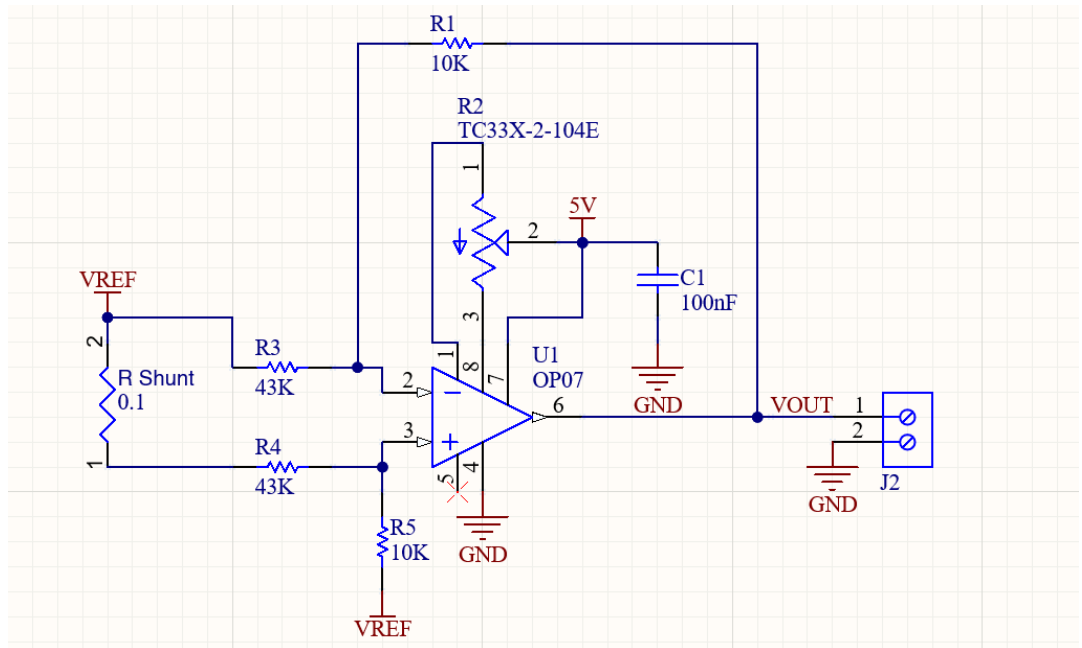
Nếu $R_l = R_2$ và $R_f = R_g$:

$$V_{out} = \frac{R_f}{R_l} (V_2 - V_l)$$

II. PHẦN CỨNG

1. Thiết kế mạch đo lường

➤ Thiết kế mạch khuếch đại vi sai:



Chọn mạch khuếch đại vi sai với $R1 = R5 = 10k$, $R3 = R4 = 43k$

$$\Rightarrow V_{out} = \frac{R1}{R3} * \Delta V_{shunt} = \frac{10}{43} * I_{Load} * R_{shunt}$$
$$\Rightarrow V_{out} = 0.023 * I_{Load}$$

Để tránh bị xén ở ngõ ra, ta chọn điện áp cấp cho OPAMP $V_{ref} =$

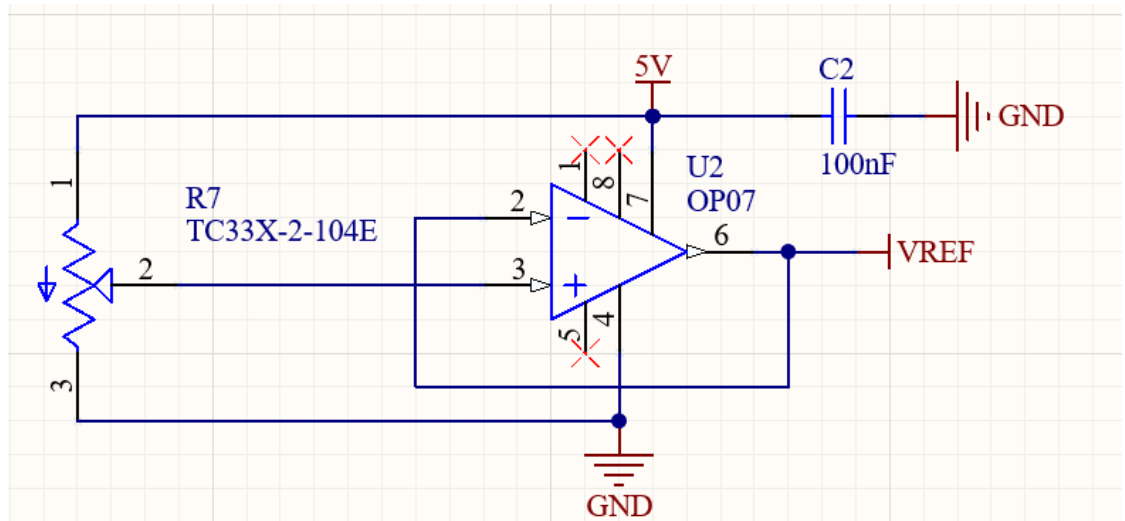
$V_{cc}/2 = 2.5V$ (tương tự cấp nguồn đôi $\pm 2.5V$). Như vậy, điện áp ngõ ra sẽ dao động quanh điện áp 2.5V

Như vậy:

- Khi dòng điện ngõ vào thay đổi 100mA -> điện áp ngõ ra thay đổi 2.3mV (lựa chọn ADC có độ phân giải phù hợp).
- Dòng điện là dòng điện tối đa (dòng AC 20Arms) điện áp ngõ ra thay đổi: 0.65V (từ 1.85V -> 3.15V)

➤ Mạch tạo điện áp V_{ref} (mạch tạo nguồn đôi):

Để tạo điện áp $V_{ref} = 2.5V$, ta sử dụng mạch OPAMP có biến trở để có thể điều chỉnh V_{ref} .



➤ **Tính toán ảnh hưởng I_{io} và I_{ib} :**

Tra datasheet OP07, ta có: $I_{OS} = 6nA$; $I_{ib} = 7nA$

$$\Rightarrow I_- = 10nA$$

Độ sai lệch điện áp do dòng rò: $\Delta V = I_- R_- = 0.1mV \ll 2.3mV$

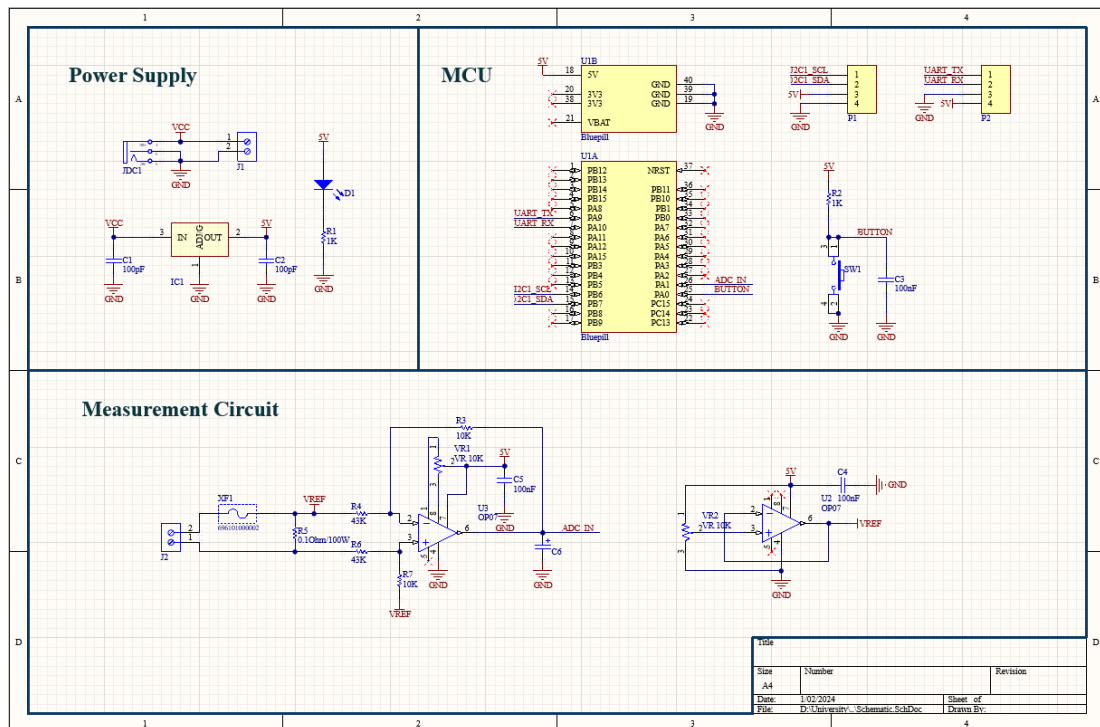
(với 2.3mV là điện áp thay đổi khi dòng thay đổi 100mA)

\Rightarrow Dòng rò ảnh hưởng không đáng kể lên điện áp ngõ ra.

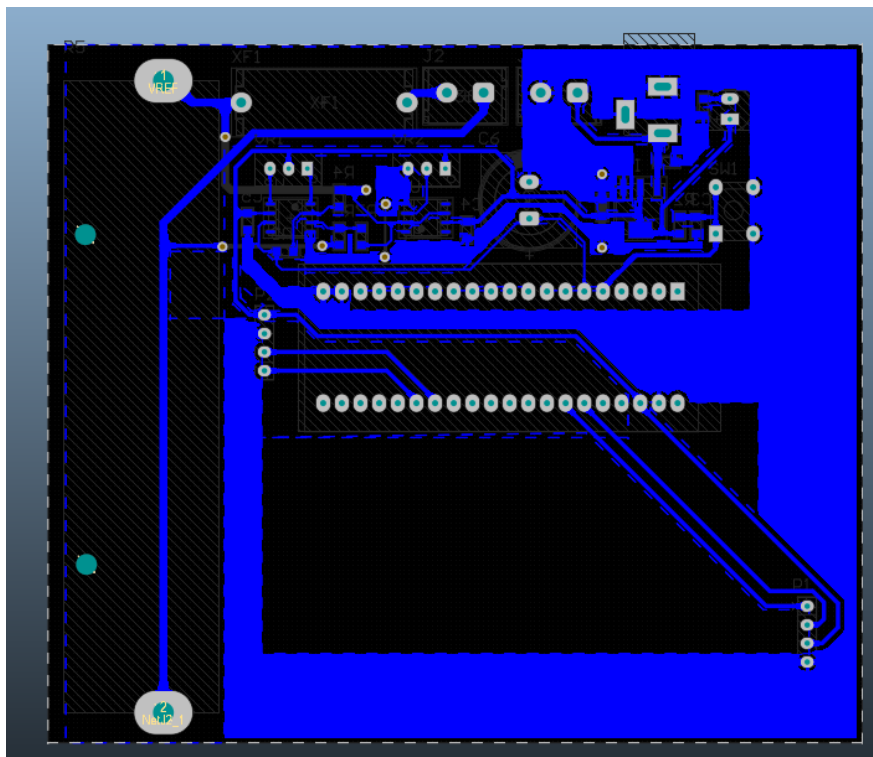
2. Mạch thực tế

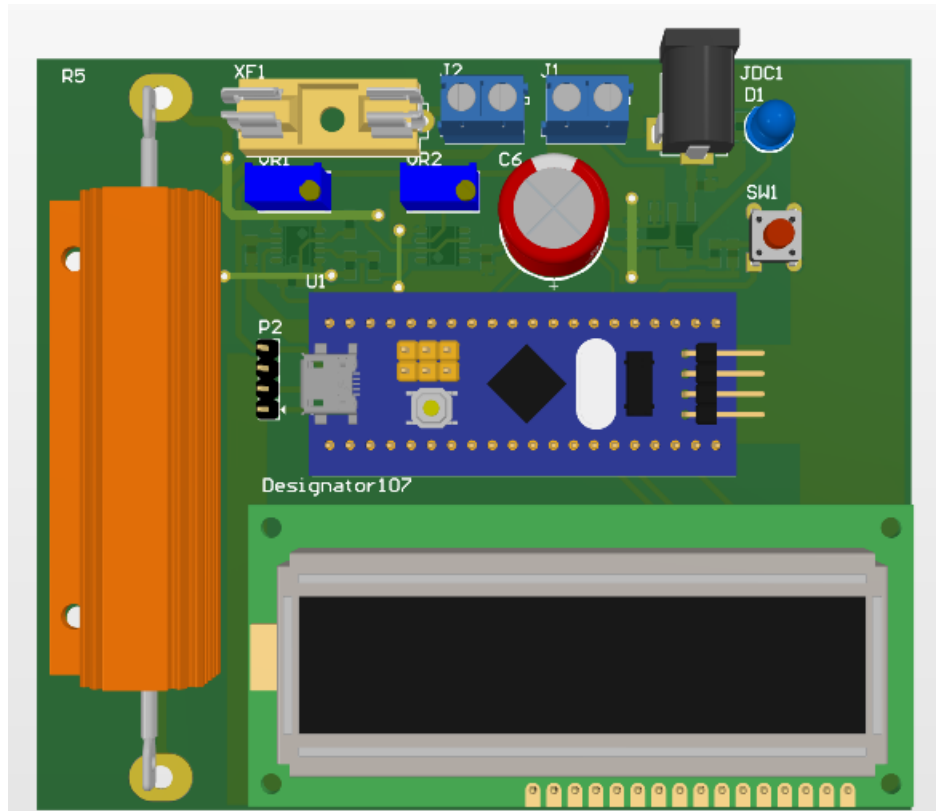
➤ **Sơ đồ mạch chi tiết:**

- Khối nguồn: nguồn cấp với điện áp trong khoảng 7 - 20V; sử dụng IC LM1117 để điện áp ổn định ở 5V.
- Khối đo lường: sử dụng mạch giải thích trên mục 2 (cơ sở lý thuyết) chuyển đổi dòng cần đo thành điện áp để đọc ADC; bảo vệ mạch đo với cầu chì 20A.
- Khối điều khiển, hiển thị: sử dụng vi điều khiển STM32F103C8T6 đọc ADC điện áp ngõ ra từ khối đo lường và hiển thị lên LCD1602 qua I2C driver.



➤ Layout:





3. Kết quả đo

➤ Đo dòng DC:

Kết quả đo dòng điện DC từ 0 đến 5,24A

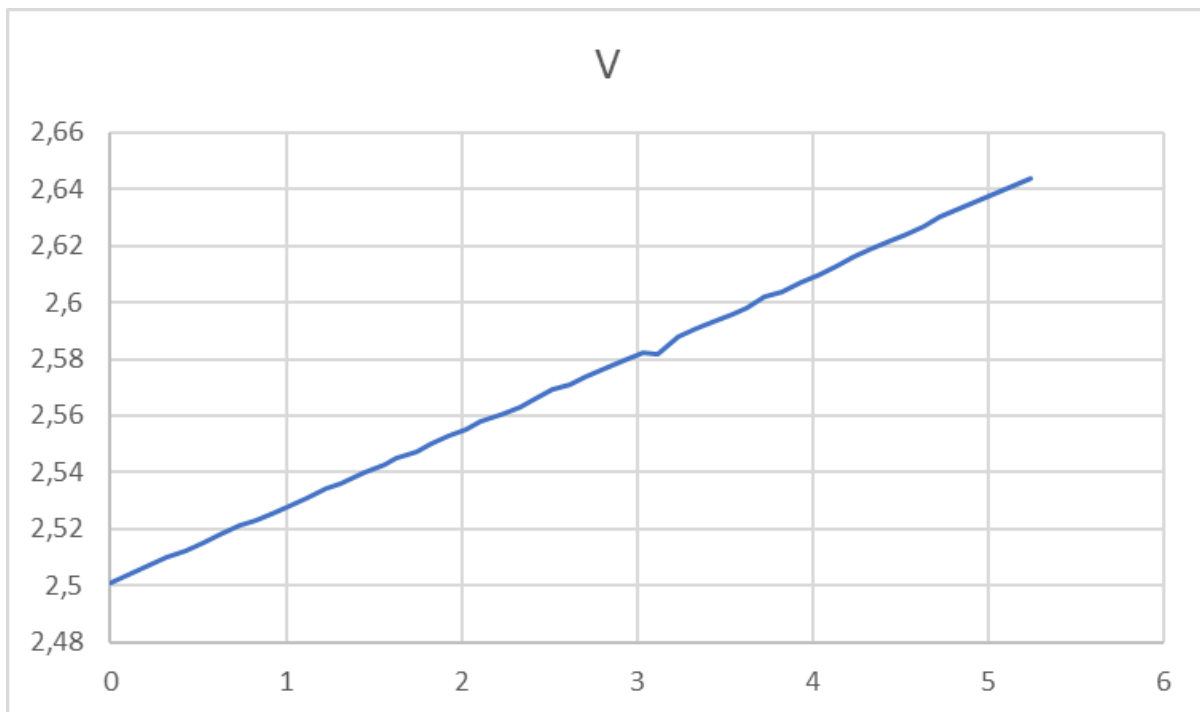
I (A)	0	0,32	0,43	0,53	0,62	0,74	0,83	0,94	1,02	1,12	1,23	1,31
V (V)	2,501	2,51	2,512	2,515	2,518	2,521	2,523	2,526	2,528	2,531	2,534	2,536

I (A)	I	1,44	1,56	1,63	1,74	1,83	1,93	2,03	2,11	2,24	2,33	2,42	2,51
V (V)	V	2,54	2,543	2,545	2,547	2,55	2,553	2,555	2,558	2,561	2,563	2,566	2,569

I (A)	I	2,62	2,71	2,83	2,94	3,03	3,12	3,23	3,33	3,42	3,54	3,62	3,71
V (V)	V	2,571	2,574	2,577	2,58	2,582	2,581	2,588	2,591	2,593	2,596	2,598	2,601

I (A)	I	3,82	3,93	4,04	4,13	4,23	4,33	4,44	4,53	4,63	4,72	5
V (V)	V	2,604	2,607	2,61	2,613	2,616	2,619	2,622	2,624	2,627	2,63	2,

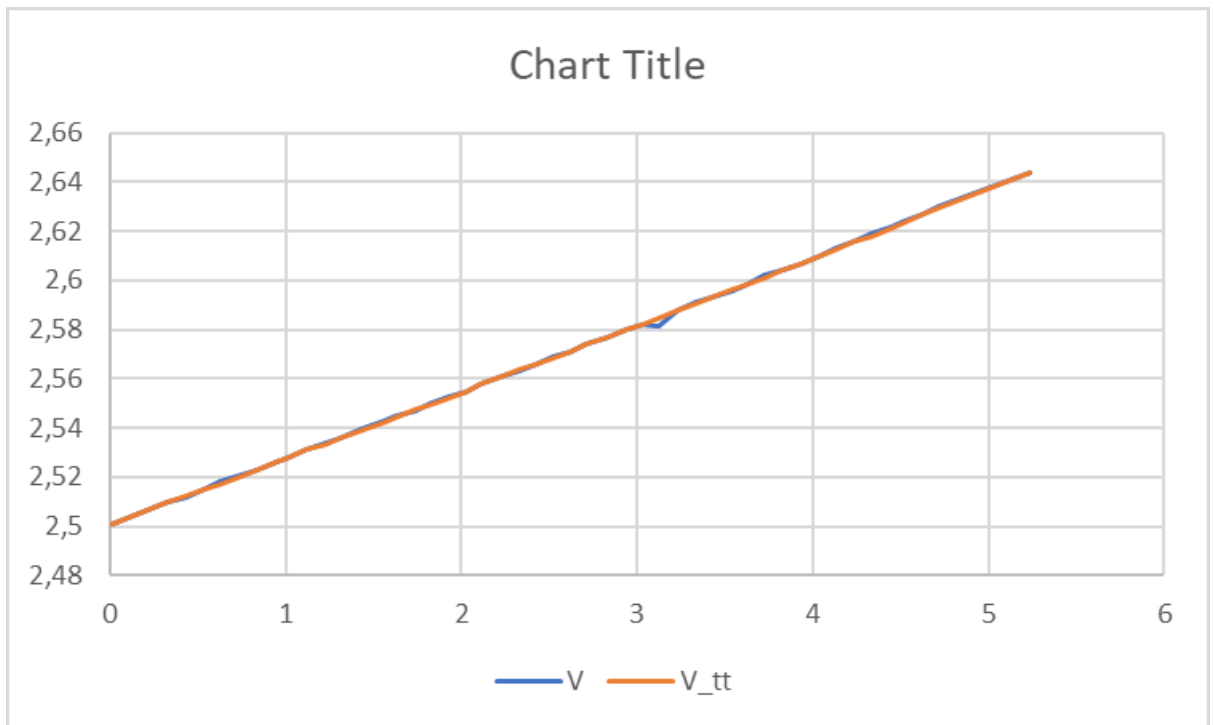
Từ kết quả, ta vẽ được đồ thị giữa trục V-I:



Nhận thấy đồ thị giống hàm tuyến tính, ta suy ra được phương trình tuyến tính giữa I-V:

$$\rightarrow I = a * V + b \rightarrow I = 0.0297 * V - 89.49681818 \text{ (A)}$$

Sau khi tính toán, ta được đồ thị như dưới:

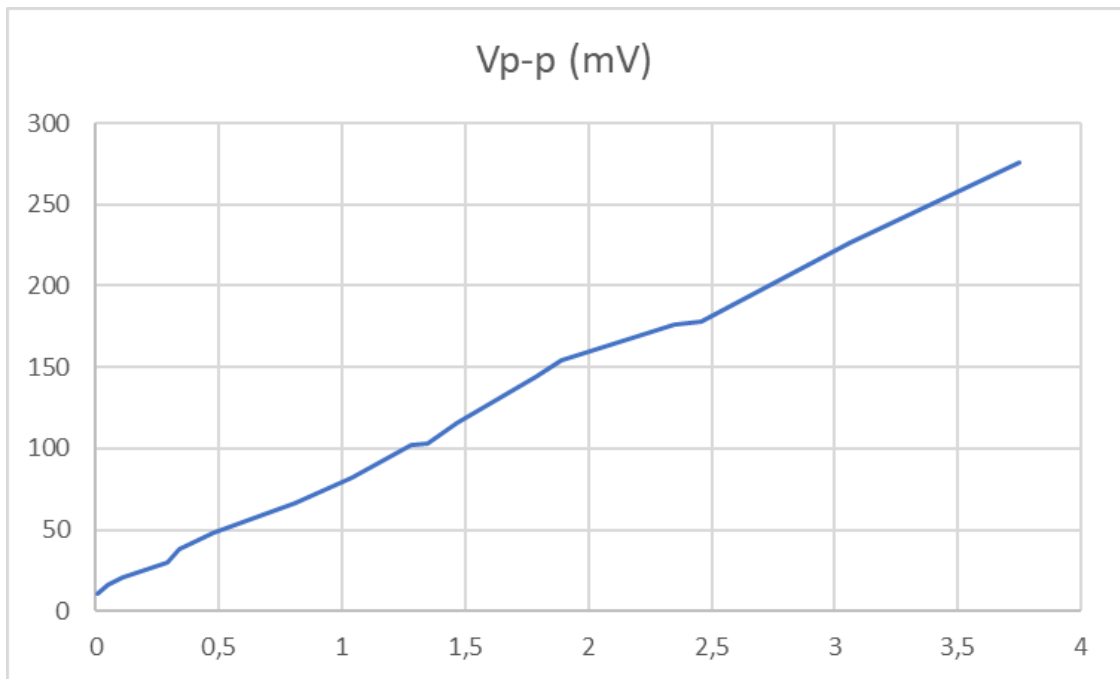


➤ **Đo dòng AC:**

Kết quả đo dòng điện AC

Irms (A)	0,01	0,05	0,11	0,18	0,29	0,34	0,48	0,81	1,04	1,28
Vp-p (mV)	10,6	16,4	20,4	24,8	30,4	38,4	48	66,4	82,4	102,4

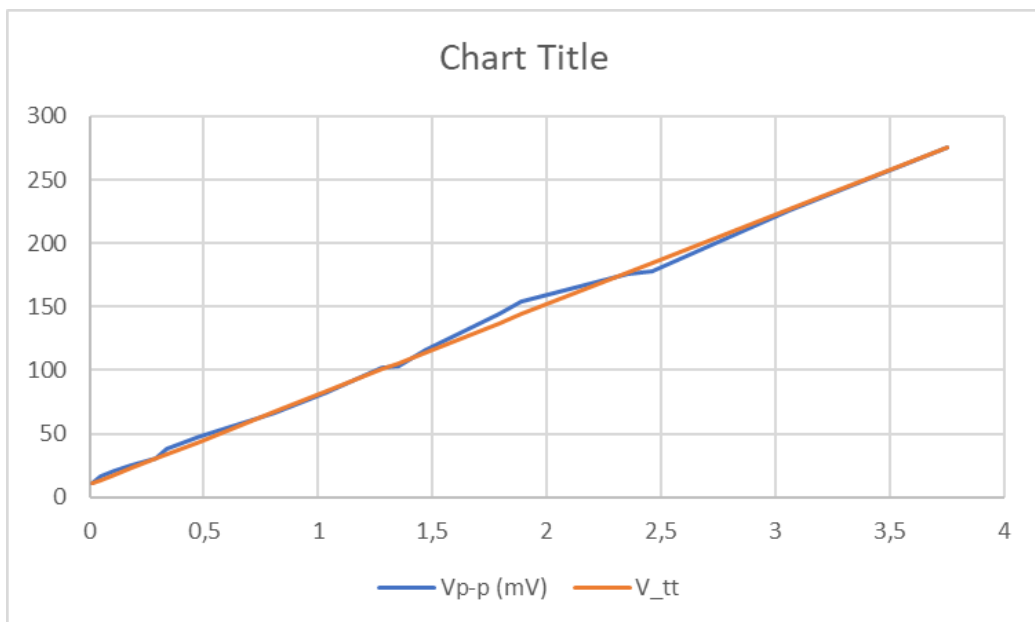
Irms (A)	1,79	1,89	2,35	2,46	3,06	3,75	1,35	1,47
Vp-p (mV)	144	154	176	178	226	276	103,2	116



Nhận thấy đồ thị giống hàm tuyến tính, ta suy ra được phương trình tuyến tính giữa I-V:

$$\rightarrow V = a * I + b \rightarrow V = 70.9625 * I + 9.8904 (V)$$

Sau khi tính toán, ta được đồ thị như dưới:



III. PHẦN MỀM

1. Lựa chọn số bit ADC và Vi điều khiển


Xét tầm đo dòng điện của mạch từ 0 - 20A, khoảng điện áp từ mạch đo đưa vào bộ ADC là 2.5V - 3V. Suy ra: $V_{ADC} = 0,025 \cdot I_{in}$. Độ phân giải của mạch đo là 100mA, vậy V_{ADC} có giá trị tối thiểu:

$$V_{ADC_{min}} = 0,025 \cdot 100 \cdot 10^{-3} = 2,5mV$$

Vậy ta chọn bộ ADC có độ phân giải nhỏ hơn $V_{ADC_{min}}$:

$$\frac{V_{ref}}{2^n} < V_{ADC_{min}}, \text{ chọn } V_{ref} = 3,3V \Rightarrow \frac{3,3}{2^n} < 2,5 \cdot 10^{-3} \Rightarrow n > 11 \text{ bit}$$

Chọn bộ ADC 12 bit, chọn vi điều khiển STM32F103C8T6.

	<p>Bộ vi xử lý: Cortex M3.</p> <p>Core size: 32 bit.</p> <p>Tốc độ tối đa: 72MHz.</p> <p>FLASH: 64KB.</p> <p>Điện áp cấp: 2V ~ 3.6V.</p> <p>ADC: 10 channel, 12 bit.</p>
------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------

2. Phương pháp đọc giá trị ADC

Đối với chế độ đo dòng điện DC: đọc giá trị trung bình của giá trị ADC và dựa vào hàm tuyến tính đã thực nghiệm để tính giá trị dòng điện.

Đối với chế độ đo dòng điện AC: trong một chu kỳ (20ms) ta đọc giá trị ADC lớn nhất đo được, sau đó dựa vào hàm tuyến tính đã thực nghiệm để tính giá trị dòng điện.

Cấu hình chi tiết bộ ADC:

Vi điều khiển STM32 có chức năng lấy mẫu ADC theo trigger của timer, chọn bộ ADC1 - Channel 1 (chân PA1) và chọn chế độ Timer 3 Trigger Out event.

$$T_{Count-up} = \frac{Prescaler + 1}{f_{Timer_3}} = \frac{27 + 1}{28MHz} = 1\mu s$$

Mỗi $30\mu s$ tạo một event: $Counter Period + 1 = \frac{30\mu s}{1\mu s} = 30$

▼ Counter Settings	
Prescaler (PSC - 16 bits value)	27
Counter Mode	Up
Counter Period (AutoReload Regi...	29
Internal Clock Division (CKD)	No Division
auto-reload preload	Disable
▼ Trigger Output (TRGO) Parameters	
Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)
Trigger Event Selection	Update Event

Ở chế độ đo DC, mỗi lần chuyển đổi ta sẽ cộng dồn giá trị ADC, nếu lấy mẫu đủ 1000 lần (sau 30ms kể từ lần đầu lấy mẫu sau reset) ta sẽ tính giá trị trung bình của 1000 mẫu giá trị ADC đó. Đoạn code thực hiện chức năng này trong hàm ngắt Update Event của Timer 3:

```
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef *hadc)
{
    adc_count = adc_count + 1;
    if (adc_count >= 1001)
    {
        adc_count = 0;
        adc_avrvalue = adc_sum/1000;
        adc_sum = 0;
    }
    adc_sum = adc_sum + adc_value;
}
```

Ở chế độ đo AC, sau mỗi chu kỳ sóng (20ms hay sau 666 lần trigger), ta sẽ cập nhật giá trị cao nhất trong toàn bộ chu kỳ sóng.

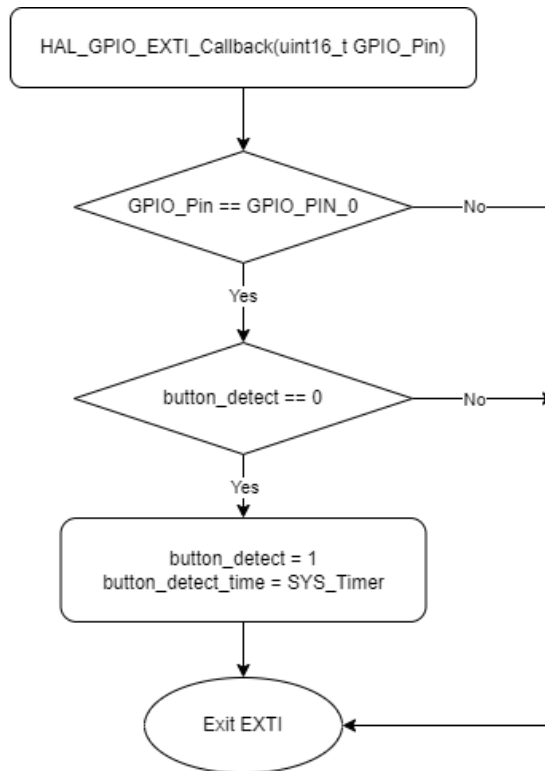
```
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef *hadc)
{
    if (adc_count >= 666)
    {
        adc_count = 0;
        adc_avrvalue = adc_max;
        adc_max = 0;
    }

    if ((adc_value > 3722) || (adc_value < 2481)) return;
    if ((adc_value > adc_prev) && (adc_value > 3101))
    {
        if (adc_value - adc_prev >= 10)
        {
            adc_prev = adc_max;
            return;
        }
        adc_max = adc_value;
    }
}
```


3. Lưu đồ giải thuật

Hàm ngắt ngoài

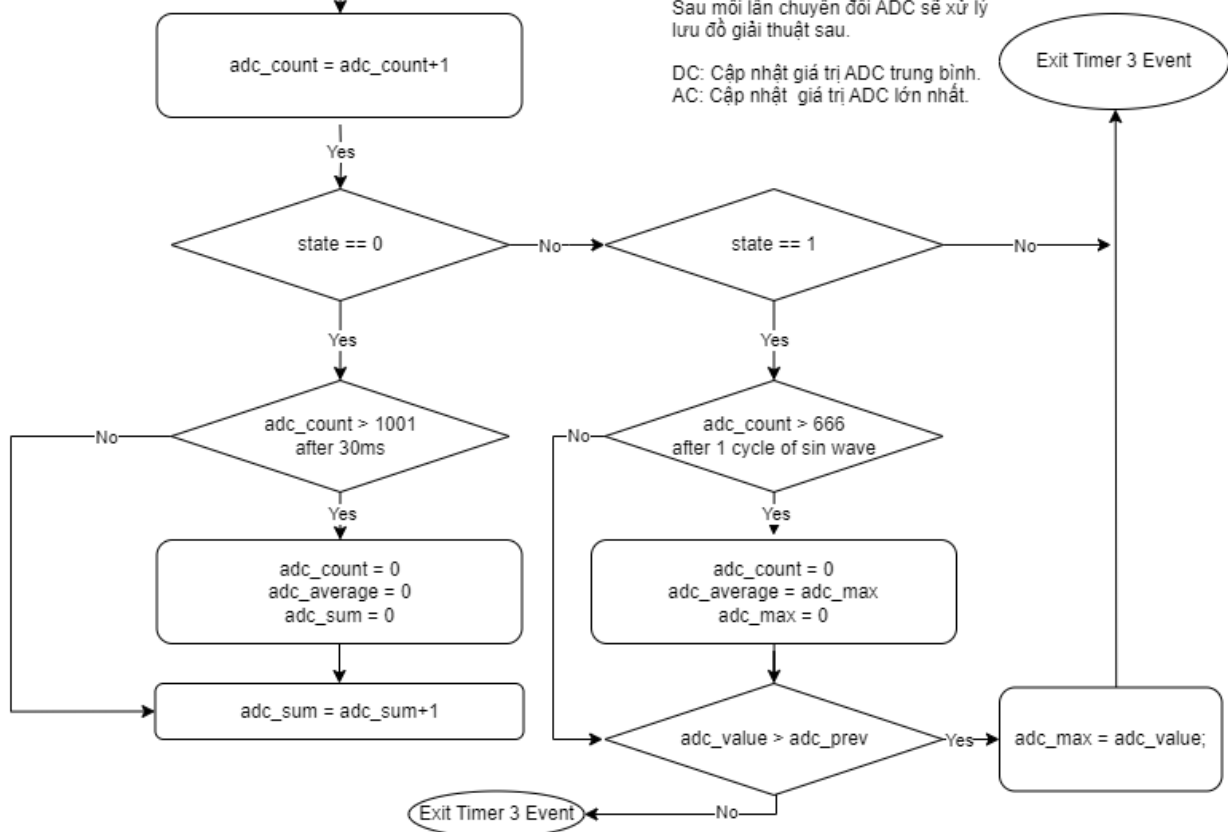
Thay đổi trạng thái khi phát hiện có nhấn nút.



Hàm xử lý event timer 3

Sau mỗi lần chuyển đổi ADC sẽ xử lý lưu đồ giải thuật sau.

DC: Cập nhật giá trị ADC trung bình.
AC: Cập nhật giá trị ADC lớn nhất.



volatile uint8_t button_detect
volatile uint32_t button_detect_time

Biến được sử dụng trong hàm ngắt
ngoài dùng để chống rung nút nhấn

volatile uint8_t state
Biến trạng thái, đổi trạng thái
mỗi khi có ngắt ngoài

Hàm main()

Khởi tạo GPIO, Ngắt ngoài, ADC_DMA, I2C
cho giao tiếp với LCD, Timer 3, khởi tạo màn
hình LCD1602.

Calibraion cho ADC, khởi động bộ ADC và
bắt đầu đếm Timer 3.

