Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh

Khoa Cơ khí Chế tạo Máy

Bộ môn Cơ Điện tử

BÁO CÁO DỰ ÁN

1. Thông tin người thực hiện:

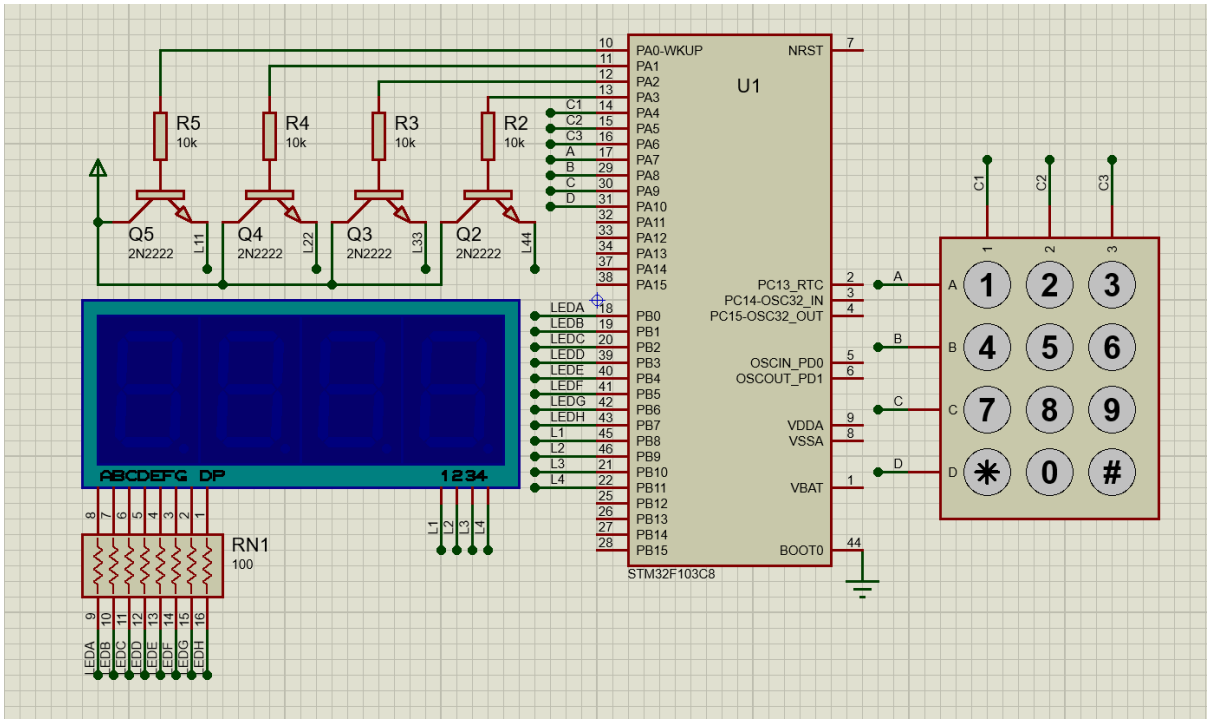
STT	Người thực hiện	MSSV	Ngày
01	Nguyễn Hữu Chí	20146479	9/5/2023

2. Yêu cầu dự án

- Cho sơ đồ mạch điện như bên dưới (bản thiết kế bằng phần mềm Proteus được đính kèm). Yêu cầu viết chương trình đọc KEYPAD để sáng LED 7 đoạn tương ứng với số được bấm. Nếu bấm nhiều số thì hiển thị thêm hàng chục, hàng trăm hàng nghìn như cách nhập liệu Calculator thông thường.

Bấm * để xoá hết và bấm # để xoá 1 số hàng đơn vị.

Lưu ý các chân đọc input của STM32 sử dụng điện trở kéo lên bên trong.



(Đã sửa lại phần cứng)

3. Nội dung:

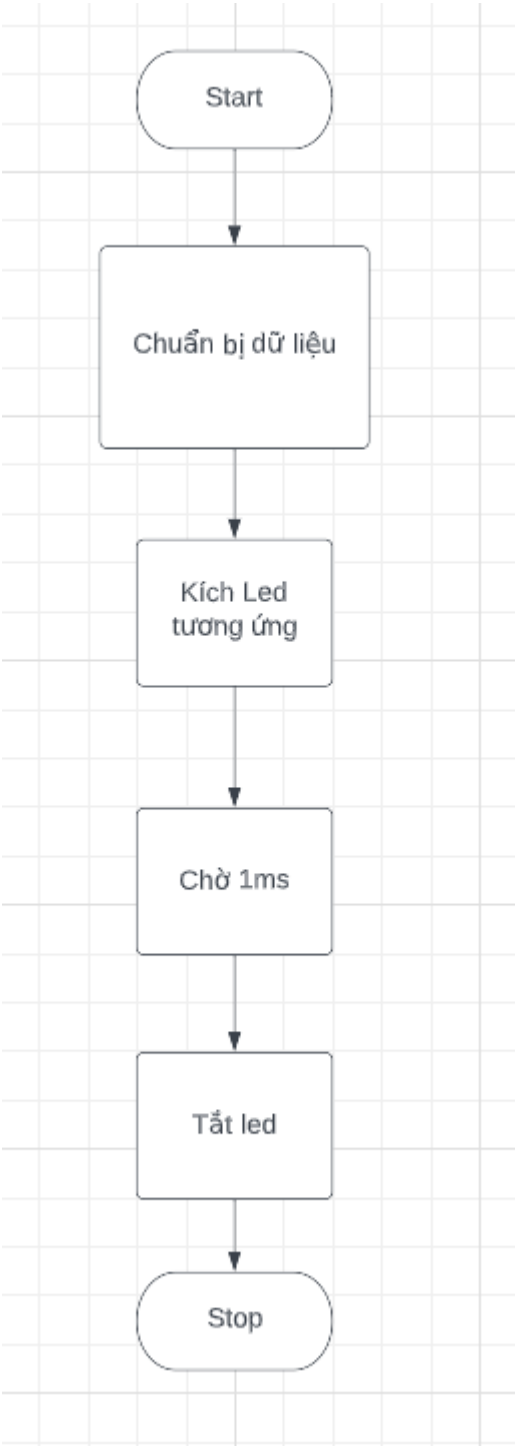
3.1. Phân tích dự án:

- Bài này có 2 vấn đề cần xử lý: 1 là phần hiển thị, 2 là phần kiểm tra nút nào được nhấn

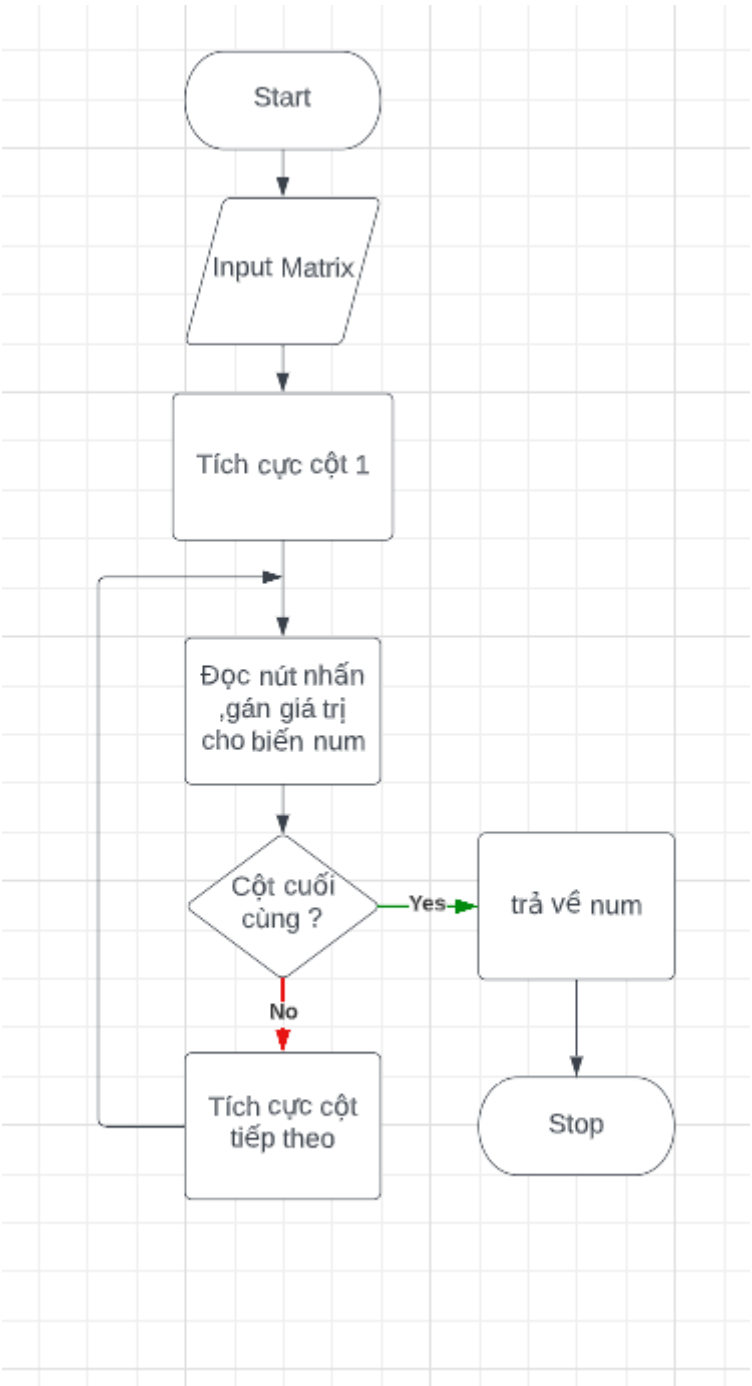
- Đối với phần hiển thị thì ta dùng phương pháp quét Led như các bài trước. Sử dụng logicstate để biết được chế độ hoạt động của Led 7SEG từ đó xây dựng được bộ mã số 0 ->9

- Đối với phần kiểm tra nút nhấn thì tương ứng với mỗi lần nhấn nút sẽ trả về số phù hợp (xử lý khi nhấn nhiều số và nhấn kí tự * # bằng các công thức phù hợp để trả về số mong muốn)

3.2. Lưu đồ lập trình:



Hàm hiển thị



Hàm đọc nút nhấn

3.3. Mã nguồn chương trình:

- Khai báo mảng chứa bộ mã hiển thị, mảng chứa các chân GPIO cần thiết, biến num dùng để chứa số hiển thị lên Led

```
43  /* USER CODE BEGIN PV */
44  const uint8_t LED[] = {0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F,0x00};
45  GPIO_TypeDef* GPIO_PORTS[] = {LEDA_GPIO_Port, LEDB_GPIO_Port, LEDC_GPIO_Port, LEDD_GPIO_Port, LEDE_GPIO_Port,
46  LEDF_GPIO_Port, LEDG_GPIO_Port,LEDH_GPIO_Port};
47  uint16_t GPIO_PINS[] = {LEDA_Pin, LEDB_Pin, LEDC_Pin, LEDD_Pin, LEDE_Pin, LEDF_Pin, LEDG_Pin, LEDH_Pin};
48  uint32_t num = 0;
49  /* USER CODE END PV */
```

- Hàm hiển thị có 2 thông số đầu vào là số hiển thị và vị trí hiển thị. Quá trình thực hiện theo flowchart

- GPIOB->ODR=0xffff; dòng này dùng để tắt hết 4 led L1 ->L4

```
170 void LED_display(uint8_t n,uint8_t pos){
171
172   for(uint8_t i = 0; i<8; i++){
173     HAL_GPIO_WritePin(GPIO_PORTS[i],GPIO_PINS[i],LED[n]>>i&1u);
174   }
175   HAL_GPIO_WritePin(L1_GPIO_Port,L1_Pin,pos%2);
176   HAL_GPIO_WritePin(L2_GPIO_Port,L2_Pin,pos%3);
177   HAL_GPIO_WritePin(L3_GPIO_Port,L3_Pin,pos%5);
178   HAL_GPIO_WritePin(L4_GPIO_Port,L4_Pin,pos%7);
179
180   HAL_Delay(1);
181
182   GPIOB->ODR=0xffff; // tat het
183 }
184 /* USER CODE END 0 */
```

- 3 dòng đầu tương ứng ta đang kích cột 1. Điều kiện if và while kiểm tra khi nhấn nút số 1 trên bàn phím và nhả ra thì biến num sẽ được tính theo công thức dưới. Công thức này vẫn hoạt động tốt khi nhấn thêm nhiều số vào sau

```
61  HAL_GPIO_WritePin(C1_GPIO_Port,C1_Pin,0);
62  HAL_GPIO_WritePin(C2_GPIO_Port,C2_Pin,1);
63  HAL_GPIO_WritePin(C3_GPIO_Port,C3_Pin,1);
64
65  if (!HAL_GPIO_ReadPin(A_GPIO_Port,A_Pin))
66  {
67    while (!HAL_GPIO_ReadPin(A_GPIO_Port,A_Pin)); //while(1) = dung tai day
68    num = num*10+1;
69  }
```

- Sử dụng cấu trúc này cho các nút khác

```
60  uint32_t read_Keypad(void) { // tra ve so 32 bit
61    HAL_GPIO_WritePin(C1_GPIO_Port,C1_Pin,0);
62    HAL_GPIO_WritePin(C2_GPIO_Port,C2_Pin,1);
63    HAL_GPIO_WritePin(C3_GPIO_Port,C3_Pin,1);
64
65    if (!HAL_GPIO_ReadPin(A_GPIO_Port,A_Pin))
66    {
67      while (!HAL_GPIO_ReadPin(A_GPIO_Port,A_Pin)); //while(1) = dung tai day
68      num = num*10+1;
69    }
70    if (!HAL_GPIO_ReadPin(B_GPIO_Port,B_Pin))
71    {
72      while (!HAL_GPIO_ReadPin(B_GPIO_Port,B_Pin));
73      num = num*10+4;
74    }
75    if (!HAL_GPIO_ReadPin(C_GPIO_Port,C_Pin))
76    {
77      while (!HAL_GPIO_ReadPin(C_GPIO_Port,C_Pin));
78      num = num*10+7;
79    }
80    if (!HAL_GPIO_ReadPin(D_GPIO_Port,D_Pin))
81    {
82      while (!HAL_GPIO_ReadPin(D_GPIO_Port,D_Pin));
83      num = 0;
84    }
85
86    HAL_GPIO_WritePin(C1_GPIO_Port,C1_Pin,1);
87    HAL_GPIO_WritePin(C2_GPIO_Port,C2_Pin,0);
88    HAL_GPIO_WritePin(C3_GPIO_Port,C3_Pin,1);
89
90    if (!HAL_GPIO_ReadPin(A_GPIO_Port,A_Pin))
91    {
92      while (!HAL_GPIO_ReadPin(A_GPIO_Port,A_Pin));
93      num = num*10+2;
94    }
95    if (!HAL_GPIO_ReadPin(B_GPIO_Port,B_Pin))
96    {
97      while (!HAL_GPIO_ReadPin(B_GPIO_Port,B_Pin));
98      num = num*10+5;
99    }
100   if (!HAL_GPIO_ReadPin(C_GPIO_Port,C_Pin))
```

- Đối với nút # và * ta có công thức phù hợp với yêu cầu

```
80    if (!HAL_GPIO_ReadPin(D_GPIO_Port,D_Pin))
81    {
82      while (!HAL_GPIO_ReadPin(D_GPIO_Port,D_Pin));
83      num = 0;
84    }
```

```
130   if (!HAL_GPIO_ReadPin(D_GPIO_Port,D_Pin))
131   {
132     while (!HAL_GPIO_ReadPin(D_GPIO_Port,D_Pin));
133     num = num/10;
134   }
```

- Đoạn này dùng để giới hạn số do led chỉ hiển thị được 4 số. Sau đó return num về cho hàm này. Ví dụ bạn nhập 6789 rồi nhập 2 thì chỉ hiển thị 6789

```
135 
136     if (num >9999) return num = num/10;
137     else return num;
138 }
```

- Hàm này trả về số 32 bit. Vì num lớn nhất = 9999

```
60 uint32_t read_Keypad(void) { // tra ve so 32 bit
```

- Trong vòng lặp chính thì ta chỉ cần gọi hàm hiển thị 4 lần tương ứng với 4 số ở 4 vị trí. If else dùng để xóa số 0 vô nghĩa, ví dụ 0098 thì hiển thị 98

```
229   while (1)
230   {
231     /* USER CODE END WHILE */
232 
233 
234     if (read_Keypad() ==0) LED_display(10 ,7); // tat
235     else LED_display(read_Keypad()%10 ,7);
236 
237     if (read_Keypad()/10 ==0) LED_display(10 ,7);
238     else LED_display((read_Keypad()/10)%10 ,5);
239 
240     if (read_Keypad()/100 ==0) LED_display(10 ,7);
241     else LED_display((read_Keypad()/100)%10,3);
242 
243     if (read_Keypad()/1000 ==0) LED_display(10 ,7);
244     else LED_display((read_Keypad()/1000)%10,2);
245 
246     //LED_display( read_Keypad()/1000,2);
247     /* USER CODE BEGIN 3 */
248   }
```

4. Ghi chú khác (nếu có)