

Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh

Khoa Cơ khí Chế tạo Máy

Bộ môn Cơ Điện tử

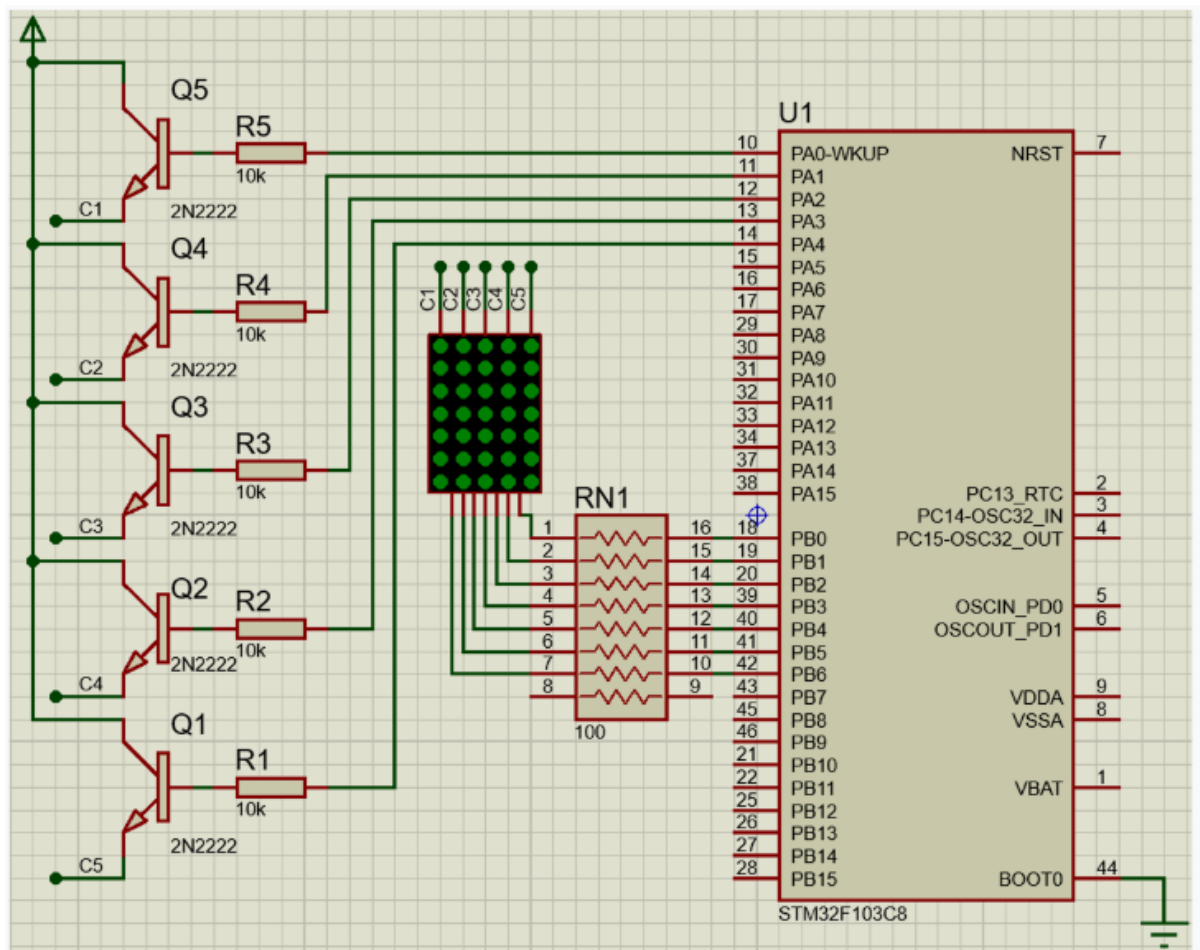
BÁO CÁO DỰ ÁN

1. Thông tin người thực hiện:

STT	Người thực hiện	MSSV	Ngày
01	Nguyễn Hữu Chí	20146479	8/5/2023

2. Yêu cầu dự án

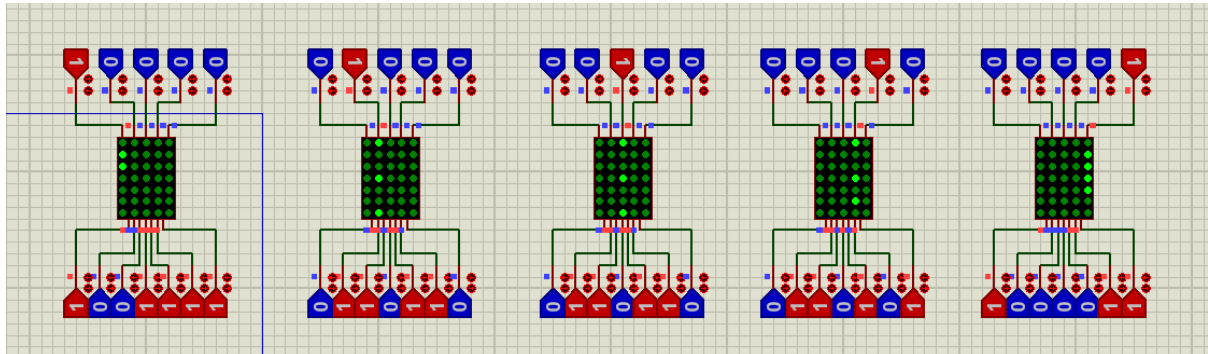
Cho sơ đồ mạch điện như bên dưới (bản thiết kế bằng phần mềm Proteus được đính kèm).
Viết chương trình điều khiển đèn LED ma trận sáng số từ 0 đến 9 và lặp lại. Mỗi lần tăng trị
hoãn 0.2s. Font số tự chọn.



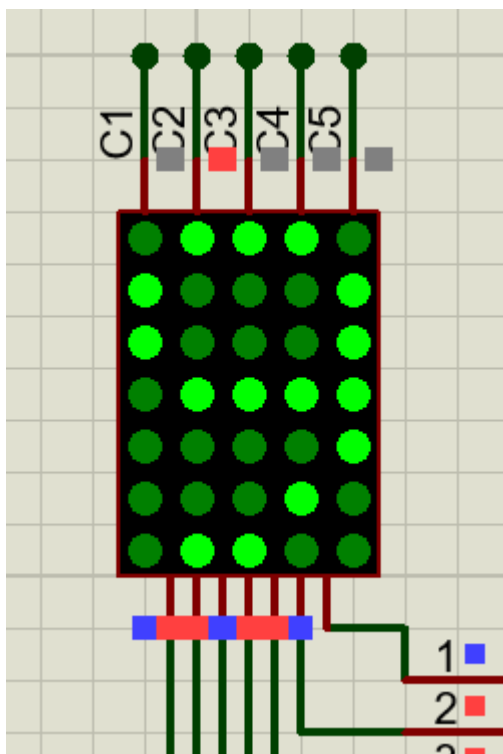
3. Nội dung:

3.1. Phân tích dự án:

- Để thực hiện được yêu cầu ta sử dụng phương pháp quét led
- Dưới đây là cách nó hoạt động



- Tại 1 thời điểm ta sẽ chuẩn bị 1 bộ mã và kích 1 cột sáng. Thời gian delay giữa 2 lần kích phải nhanh hơn tốc độ quét của mắt thì ta sẽ nhận được kết quả tương ứng với 5 lần kích ở trên là số 9



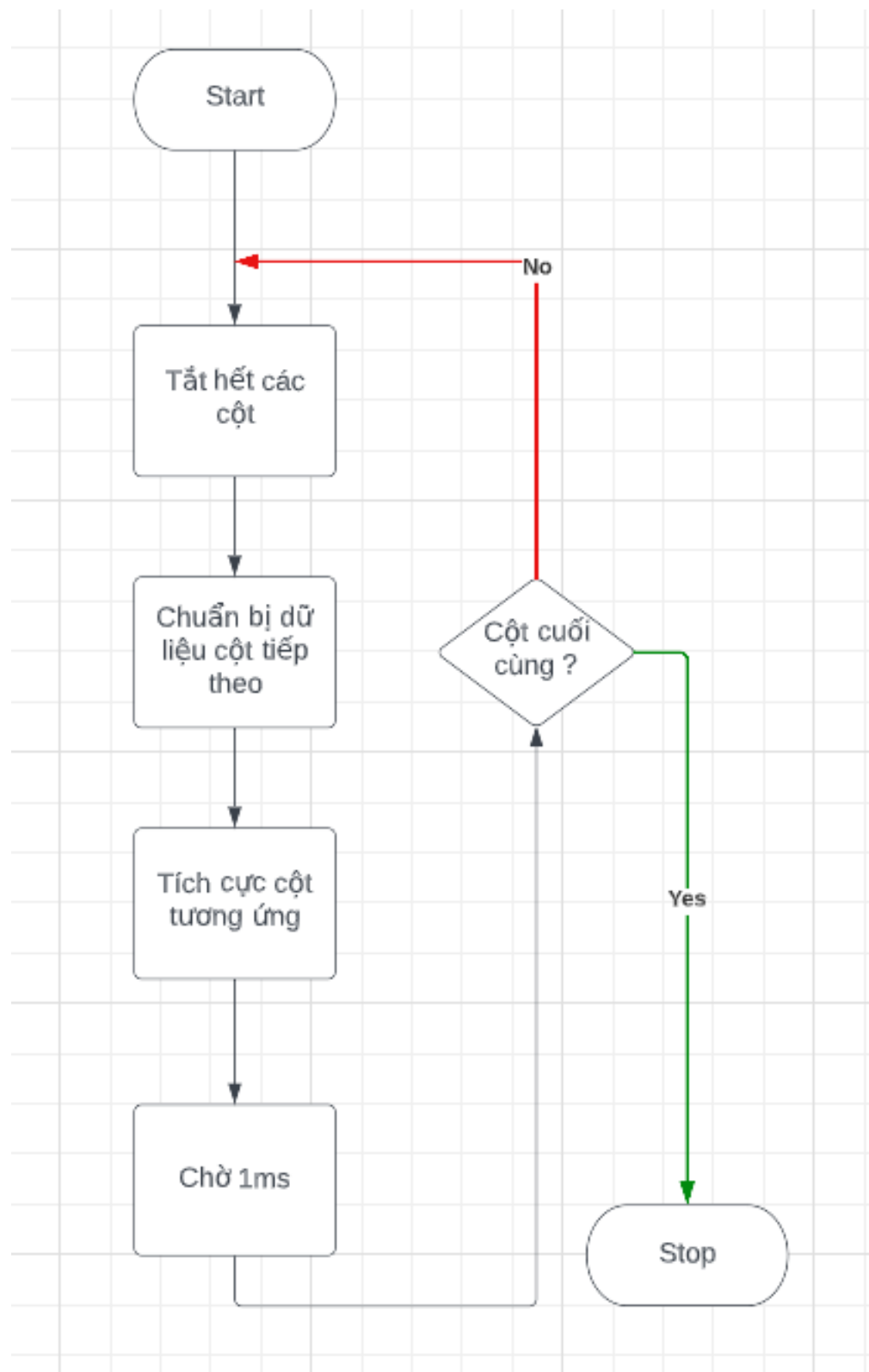
Giải thích thêm về phần delay: 1 giây mắt quét được tối đa 24 ảnh tương ứng với 1 ảnh là 42 ms

. Đồng nghĩa với việc mắt người chỉ nhìn được tối đa 1 ảnh trong thời gian < 42ms

Trong bài toán này chúng ta kích 5 lần tương ứng 5 bức ảnh. Nếu tổng thời gian delay của 5 bức ảnh này phải < 42 giây thì ta sẽ không phân biệt được 5 bức ảnh, lợi dụng vào điều này chúng ta chỉ nhìn được 1 bức ảnh duy nhất (ảnh số 9 bên trái). Vì vậy thời gian delay **tối đa** cho phép là $42/5 = 8.4$ ms.

- Dựa vào 5 hình tương ứng trên ta có thể tự thiết kế bộ mã cho việc hiển thị

3.2. Lưu đồ lập trình:



3.3. Mã nguồn chương trình:

- Chuẩn bị bộ mã tương ứng với 10 số hiển thị lên Led

```
43  /* USER CODE BEGIN PV */
44  const uint8_t NUMDATA[10][5] = {0x41,0x3a,0x36,0x2e,0x41, //0      const nam trong ROM
45                                     0x7f,0x5e,0x00,0x7e,0x7f, //1
46                                     0x5e,0x3c,0x3a,0x36,0x4e, //2
47                                     0x3d,0x3e,0x2e,0x16,0x39, //3
48                                     0x73,0x6b,0x5b,0x00,0x7b, //4
49                                     0x0d,0x2e,0x2e,0x2e,0x31, //5
50                                     0x61,0x56,0x36,0x36,0x79, //6
51                                     0x3f,0x38,0x37,0x2f,0x1f, //7
52                                     0x49,0x36,0x36,0x36,0x49, //8
53                                     0x4f,0x36,0x36,0x35,0x43}; //9
54  /* USER CODE END PV */
```

- Vòng lặp chính

```
108  /* USER CODE BEGIN WHILE */
109  while (1)
110  {
111      /* USER CODE END WHILE */
112      for (uint8_t m = 0;m<10;m++)
113      {
114          for (uint8_t k = 0;k<40;k++) // tao do tre = delay
115          {
116              for (uint8_t n = 0;n<5;n++)
117              {
118                  HAL_GPIO_WritePin(C1_GPIO_Port,C1_Pin,GPIO_PIN_RESET);
119                  HAL_GPIO_WritePin(C2_GPIO_Port,C2_Pin,GPIO_PIN_RESET);
120                  HAL_GPIO_WritePin(C3_GPIO_Port,C3_Pin,GPIO_PIN_RESET);
121                  HAL_GPIO_WritePin(C4_GPIO_Port,C4_Pin,GPIO_PIN_RESET);
122                  HAL_GPIO_WritePin(C5_GPIO_Port,C5_Pin,GPIO_PIN_RESET);
123
124                  // Chuan bi du lieu
125                  HAL_GPIO_WritePin(LED1_GPIO_Port,LED1_Pin,NUMDATA[m][n]&1);
126                  HAL_GPIO_WritePin(LED2_GPIO_Port,LED2_Pin,NUMDATA[m][n]&1<<1);
127                  HAL_GPIO_WritePin(LED3_GPIO_Port,LED3_Pin,NUMDATA[m][n]&1<<2);
128                  HAL_GPIO_WritePin(LED4_GPIO_Port,LED4_Pin,NUMDATA[m][n]&1<<3);
129                  HAL_GPIO_WritePin(LED5_GPIO_Port,LED5_Pin,NUMDATA[m][n]&1<<4);
130                  HAL_GPIO_WritePin(LED6_GPIO_Port,LED6_Pin,NUMDATA[m][n]&1<<5);
131                  HAL_GPIO_WritePin(LED7_GPIO_Port,LED7_Pin,NUMDATA[m][n]&1<<6);
132
133                  uint8_t col = 1<<n;
134                  HAL_GPIO_WritePin(C1_GPIO_Port,C1_Pin,col&1); // sang cot 1 khi n = 0
135                  HAL_GPIO_WritePin(C2_GPIO_Port,C2_Pin,col&1<<1); // sang cot 2 khi n = 1
136                  HAL_GPIO_WritePin(C3_GPIO_Port,C3_Pin,col&1<<2);
137                  HAL_GPIO_WritePin(C4_GPIO_Port,C4_Pin,col&1<<3);
138                  HAL_GPIO_WritePin(C5_GPIO_Port,C5_Pin,col&1<<4);
139                  HAL_Delay(1); // < 8.4ms
140              }
141          }
142      }
143      HAL_Delay(1);
144
145      /* USER CODE BEGIN 3 */
146  }
```

- Phần này để tắt hết các cột của Led

```
118  HAL_GPIO_WritePin(C1_GPIO_Port,C1_Pin,GPIO_PIN_RESET);
119  HAL_GPIO_WritePin(C2_GPIO_Port,C2_Pin,GPIO_PIN_RESET);
120  HAL_GPIO_WritePin(C3_GPIO_Port,C3_Pin,GPIO_PIN_RESET);
121  HAL_GPIO_WritePin(C4_GPIO_Port,C4_Pin,GPIO_PIN_RESET);
122  HAL_GPIO_WritePin(C5_GPIO_Port,C5_Pin,GPIO_PIN_RESET);
123
```

- Phần này dùng để chuẩn bị dữ liệu cho cột (1 cột có 6 hàng)

```

125     HAL_GPIO_WritePin(LED1_GPIO_Port,LED1_Pin,NUMDATA[m][n]&1);
126     HAL_GPIO_WritePin(LED2_GPIO_Port,LED2_Pin,NUMDATA[m][n]&1<<1);
127     HAL_GPIO_WritePin(LED3_GPIO_Port,LED3_Pin,NUMDATA[m][n]&1<<2);
128     HAL_GPIO_WritePin(LED4_GPIO_Port,LED4_Pin,NUMDATA[m][n]&1<<3);
129     HAL_GPIO_WritePin(LED5_GPIO_Port,LED5_Pin,NUMDATA[m][n]&1<<4);
130     HAL_GPIO_WritePin(LED6_GPIO_Port,LED6_Pin,NUMDATA[m][n]&1<<5);
131     HAL_GPIO_WritePin(LED7_GPIO_Port,LED7_Pin,NUMDATA[m][n]&1<<6);

```

- Phần này dùng để kích cột tương ứng sau khi chuẩn bị mã.

```

133     uint8_t col = 1<<n;
134     HAL_GPIO_WritePin(C1_GPIO_Port,C1_Pin,col&1); // sang cot 1 khi n = 0
135     HAL_GPIO_WritePin(C2_GPIO_Port,C2_Pin,col&1<<1); // sang cot 2 khi n = 1
136     HAL_GPIO_WritePin(C3_GPIO_Port,C3_Pin,col&1<<2);
137     HAL_GPIO_WritePin(C4_GPIO_Port,C4_Pin,col&1<<3);
138     HAL_GPIO_WritePin(C5_GPIO_Port,C5_Pin,col&1<<4);
139     HAL_Delay(1); // < 8.4ms

```

Dòng HAL_Delay(1) khá quan trọng. Nếu không có dòng này thì Led sẽ không hiển thị gì do máy tính xử lý quá nhanh mắt người không thể nhìn được

- Vòng for nhỏ nhất dùng để chuyển đổi giữa các cột trong quá trình hiển thị 1 số
- Vòng for giữa dùng để lặp lại vòng for thứ nhất (k lần) nhằm tạo độ trễ giữa mỗi lần chuyển số (độ trễ này không chính xác, chỉ tương đối). Ở ví dụ này thì độ trễ bằng $40 \times 5 \times 1\text{ms} = 200\text{ms}$
- Vòng for lớn nhất dùng để thay đổi số hiển thị 0 -> 9

```

112     for (uint8_t m = 0;m<10;m++)
113     {
114         for (uint8_t k = 0;k<40;k++) // tao do tre = delay
115         {
116             for (uint8_t n = 0;n<5;n++)
117             {

```

4. Ghi chú khác (nếu có)