

# BÁO CÁO DỰ ÁN

## 1. Thông tin người thực hiện:

| STT | Người thực hiện | MSSV     | Ngày      |
|-----|-----------------|----------|-----------|
| 01  | Nguyễn Hữu Chí  | 20146479 | 14/5/2023 |

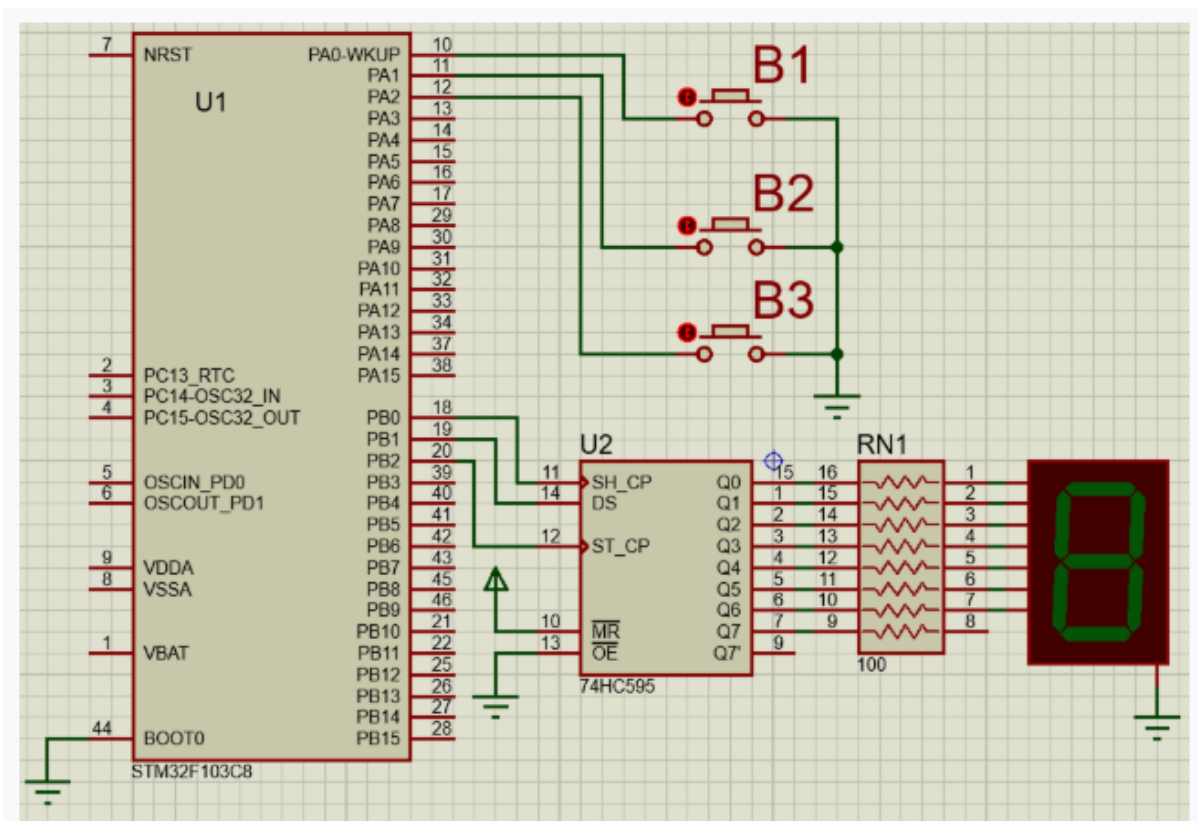
## 2. Yêu cầu dự án

- Cho sơ đồ mạch điện như bên dưới (bản thiết kế bằng phần mềm Proteus được đính kèm). Yêu cầu viết chương trình đọc nút nhấn B1, B2 và B3 để thực hiện: Ban đầu LED 7 đoạn hiển thị số 0.

Yêu Cầu 1: Khi nhấn nút B1 LED 7 đoạn hiển thị tăng lên 1 đơn vị.

Yêu cầu 2: Khi nhấn nút B2 LED 7 đoạn hiển thị giảm 1 đơn vị.

Yêu cầu 3: Khi nhấn nút B3 LED 7 đoạn hiển thị về 0.

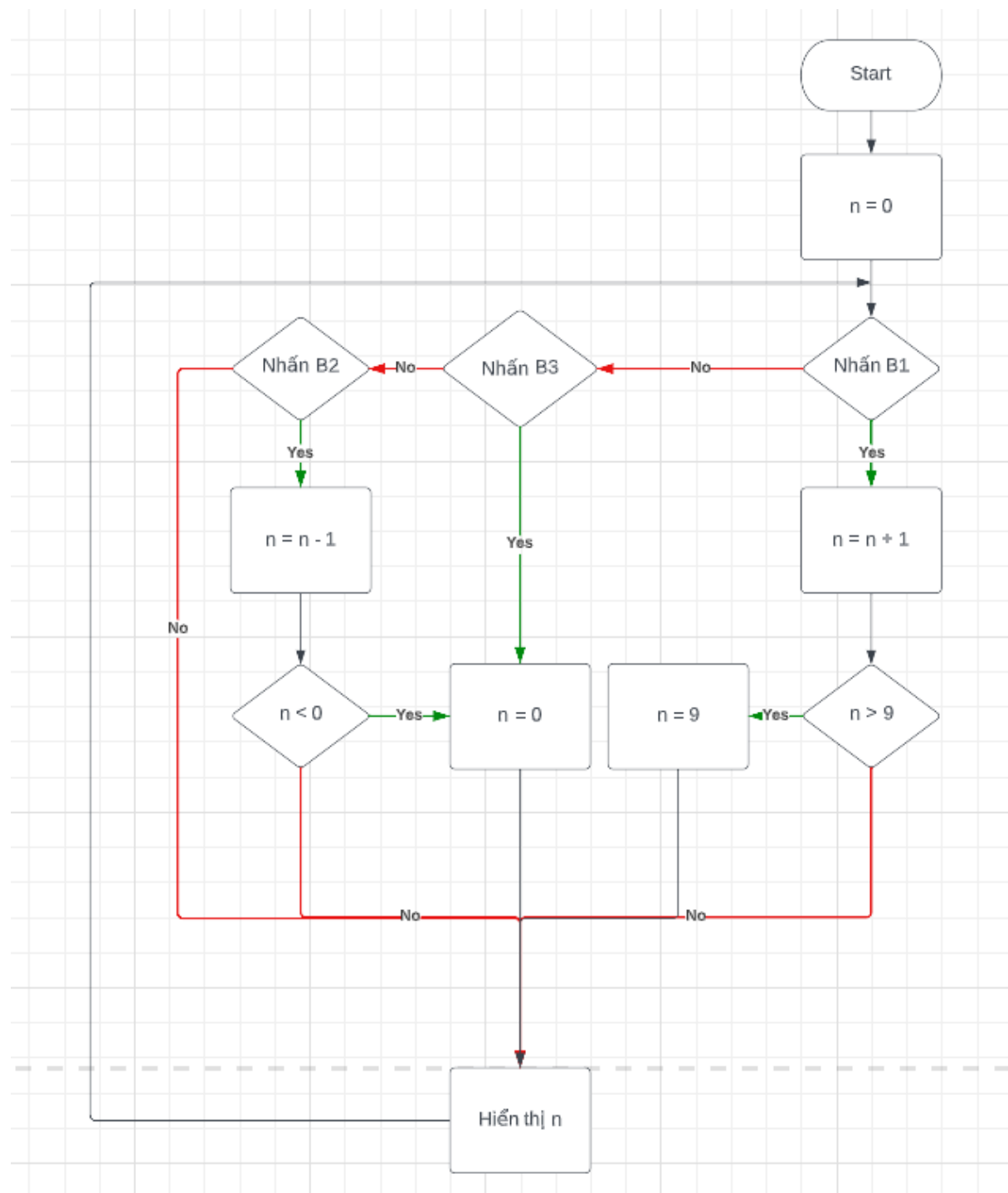


### 3. Nội dung:

#### 3.1. Phân tích dự án:

- Dùng phương pháp Statemachine để giải quyết bài toán. Bao gồm bốn bước:
  - + Scan input
  - + State control
  - + Output control
  - + Output
- Mỗi nút sẽ có 1 biến trạng thái riêng (do mỗi nút có 1 chức năng khác nhau)
- Ta sẽ tạo ra 1 biến để chứa giá trị số hiển thị lên Led. Ở phần Output control tương ứng với 1 nút nhấn sẽ có 1 công thức tính toán lại giá trị của biến hiển thị
- Việc hiển thị số lên Led thì khá đơn giản (dựa theo các bài trước)

### 3.2. Lưu đồ lập trình:



### 3.3. Mã nguồn chương trình:

- Define các biến (clean code)

```
32 /* Private define -----*/
33 /* USER CODE BEGIN PD */
34 #define B1_PRESSED GPIO_PIN_RESET //0
35 #define B2_PRESSED GPIO_PIN_RESET //0
36 #define B3_PRESSED GPIO_PIN_RESET //0
37 /* USER CODE END PD */
```

- Khai báo các biến cần thiết

```
46 /* USER CODE BEGIN PV */
47 uint8_t B1 = 0;
48 uint8_t B2 = 0;
49 uint8_t B3 = 0;
50 uint8_t B1SystemState = 0;
51 uint8_t B2SystemState = 0;
52 uint8_t B3SystemState = 0;
53 int8_t n = 0;
54 /* USER CODE END PV */
```

- Hàm hiển thị số lên Led

```
64 /* USER CODE BEGIN 0 */
65 void LED_put(uint8_t n) {
66     const uint8_t LED[] = {0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7f,0x6F};
67     for (uint8_t i =0; i<8; i++) {
68         HAL_GPIO_WritePin(DAT_GPIO_Port,DAT_Pin,LED[n]&0x80>>i);
69         HAL_GPIO_WritePin(CLK_GPIO_Port,CLK_Pin,GPIO_PIN_SET);
70         HAL_GPIO_WritePin(CLK_GPIO_Port,CLK_Pin,GPIO_PIN_RESET);
71     }
72 }
73 /* USER CODE END 0 */
```

- Kiểm tra nút được nhấn

```
114 // Scan Input
115 B1 = HAL_GPIO_ReadPin(B1_GPIO_Port,B1_Pin); // B1, B2, B3 pull_up
116 B2 = HAL_GPIO_ReadPin(B2_GPIO_Port,B2_Pin);
117 B3 = HAL_GPIO_ReadPin(B3_GPIO_Port,B3_Pin);
118 ...
```

- Thay đổi biến trạng thái tương ứng với các nút khi được nhấn. Lưu ý 3 nút là 3 biến trạng thái khác nhau

```
119 // State control
120 if ((B1SystemState == 0) && (B1 == B1_PRESSED))
121     B1SystemState = 1;
122 else if ((B1SystemState == 2) && (B1 != B1_PRESSED))
123     B1SystemState = 0;
124
125
126 if ((B2SystemState == 0) && (B2 == B2_PRESSED))
127     B2SystemState = 1;
128 else if ((B2SystemState == 2) && (B2 != B2_PRESSED))
129     B2SystemState = 0;
130
131
132 if ((B3SystemState == 0) && (B3 == B3_PRESSED))
133     B3SystemState = 1;
134 else if ((B3SystemState == 2) && (B3 != B3_PRESSED))
135     B3SystemState = 0;
```

- Xử lý giá trị n và hiển thị theo flowchart

```
137 // Output control
138 if (B1SystemState == 1) {
139     n = n + 1;
140     if (n > 9) n = 9;
141     B1SystemState = 2;
142 }
143 if (B2SystemState == 1) {
144     n = n - 1;
145     if (n < 0) n = 0;
146     B2SystemState = 2;
147 }
148 if (B3SystemState == 1) {
149     n = 0;
150     B3SystemState = 2;
151 }
152
153 // Output
154 LED_put(n);
155 HAL_GPIO_WritePin(LAT_GPIO_Port, LAT_Pin, GPIO_PIN_SET);
156 HAL_GPIO_WritePin(LAT_GPIO_Port, LAT_Pin, GPIO_PIN_RESET);
157 /* USER CODE BEGIN 3 */
158
```

#### 4. Ghi chú khác (nếu có)