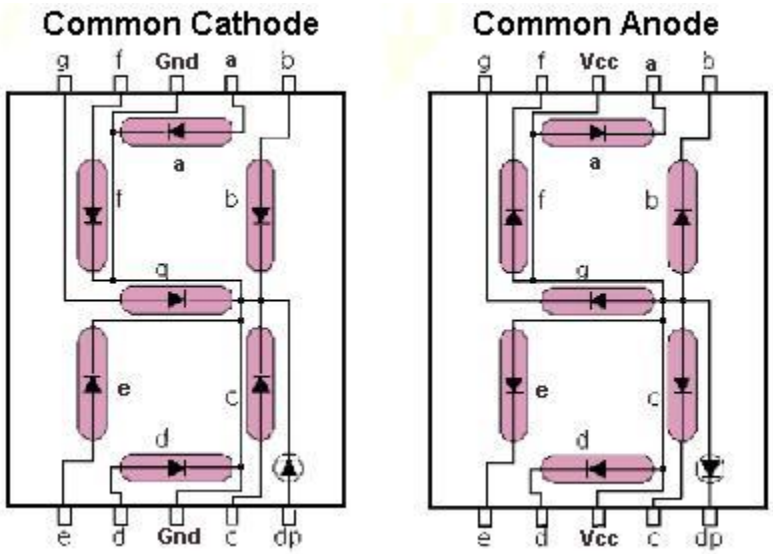
Phần 2: Lập trình LED 7 thanh với STM32F103C8.

Bài trước chúng ta học lập trình giao tiếp để điều khiển Output (LED) theo Pin và theo Port. Ở bài này chúng ta sẽ áp dụng những kiến thức đó để thực hiện lập trình điều khiển hiển thị trên LED 7 thanh theo để hiển thị theo ý mình. Về cơ bản, chúng ta dùng 8 chân đầu ra để điều khiển LED 7 thanh với mục đích hiển thị số theo ý mình.

Sau đây là hình ảnh và chi tiết về 1 con LED 7 thanh và bảng dữ liệu mã hóa với các số từ 0-9 theo kiểu LED sử dụng Anot chung.



Chữ số	A	B	C	D	E	F	G	Dp	Mã
0	0	0	0	0	0	0	1	1	**0xC0**
1	1	0	0	1	1	1	1	1	**0xF9**
2	0	0	1	0	0	1	0	1	**0xA4**
3	0	0	0	0	1	1	0	1	**0xB0**
4	1	0	0	1	1	0	0	1	**0x99**
5	0	1	0	0	1	0	0	1	**0x92**
6	0	1	0	0	0	0	0	1	**0x82**
7	0	0	0	1	1	1	1	1	**0xF8**
8	0	0	0	0	0	0	0	1	**0x80**
9	0	0	0	0	1	0	0	1	**0x90**

Các bạn ghi nhớ bảng trên để có thể áp dụng trong các bài toán liên quan đến LED 7 thanh mà không cần lặp lại việc này.

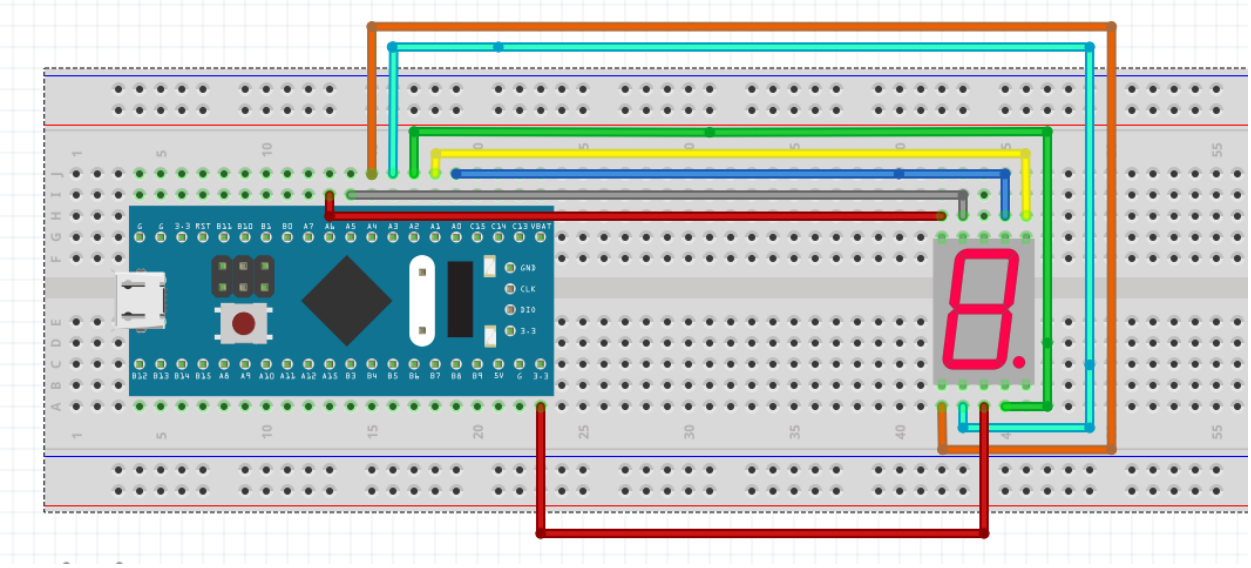
Bài 1: Hãy lập bảng các giá trị từ 0-9 cho LED 7 thanh sử dụng Cathode chung. (Viết ra giấy nộp cho nhóm trưởng)

Chúng ta sẽ lần lượt thực hiện những bài mô phỏng sau:

Bài 2: Điều khiển 1 LED 7 thanh.

Để thực hiện bài tập này các bạn thực hiện vẽ như mạch. Linh kiện cần thêm cho bài này là Led 7 cạnh, Anode chung. Các em search từ "7SEG-COM_ANODE".

Sau đây là chương trình điều khiển hiển thị lần lượt các số từ 0 đến 9 trên LED 7 thanh. Chương trình có nội dung như sau:



// led 7 thanh

```
/*
	A0...6 = a,b,c,d,e,f,g
*/
#include "stm32f10x.h"
#include "stm32f10x_gpio.h"
#include "stm32f10x_rcc.h"

void GPIO_configuration(void);
void Delay(unsigned int time);

void GPIO_configuration(void)
{
	RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);

	GPIO_InitTypeDef GPIO_InitStructure;
```

```
	GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;

	GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0 |GPIO_Pin_1 |GPIO_Pin_2 |GPIO_Pin_3 |GPIO_Pin_4 |GPIO_Pin_5 |GPIO_Pin_6;

	GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;

	GPIO_Init(GPIOA, &GPIO_InitStructure);

}


void Delay(unsigned int time){


	unsigned int i,j;


	for(i=0;i<time;i++){

		for(j=0;j< 0x2AFF; j++);

	}

}


int main(void)
```

```
{
    SystemInit();
    GPIO_configuration();
    uint16_t mang[10]={0x0040, 0x00F9, 0x0024, 0x0030, 0x0019, 0x0012, 0x0002, 0x00F8, 0x0000, 0x0010};
    while(1)
    {
        for(int i=0;i<=9;i++)
        {
            GPIOA->ODR = mang[i];
            Delay(1000);
        }
    }
    /*
    while(1)
    {
        GPIOA->ODR = 0x0040; //so 0
```

```
Delay(1000);

GPIOA->ODR = 0x00F9; //so 1

Delay(1000);

GPIOA->ODR = 0x0024; //so 2

Delay(1000);

GPIOA->ODR = 0x0030; //so 3

Delay(1000);

GPIOA->ODR = 0x0019; //so 4

Delay(1000);

GPIOA->ODR = 0x0012; //so 5

Delay(1000);

GPIOA->ODR = 0x0002; //so 6

Delay(1000);

GPIOA->ODR = 0x00F8; //so 7

Delay(1000);

GPIOA->ODR = 0x0000; //so 8

Delay(1000);
```

```
            GPIOA->ODR = 0x0010; //so 9
            Delay(1000);
        }
        */
}
```

Ở đây chúng ta thực hiện lưu mảng dữ liệu LED 7 thanh ở bảng trên vào một mảng. Khi đó phần tử mang[i] chứa dữ liệu hiển thị của số i trên LED thanh ở trên mạch với điều kiện i nằm từ 0-9.

Ở vòng lặp for, chúng ta thực hiện đẩy dữ liệu ra chân Port A lần lượt theo các mã của các số từ 0 đến 9 ra. Chạy mô phỏng chúng ta sẽ thấy các số này lần lượt hiển thị trên LED 7 thanh.

Câu hỏi 2.1: Muốn hiển thị các số chẵn hoặc lẻ thì sao?

Câu hỏi 2.2: Muốn hiển thị theo thứ tự ngược lại (đếm ngược) thì làm ntn?

Bài 3: Chọn LED 7 thanh Cathode chung, viết chương trình mới và mô phỏng để đạt kết quả như bài 2.
Lặp lại 2 cầu hỏi ở bài 2.

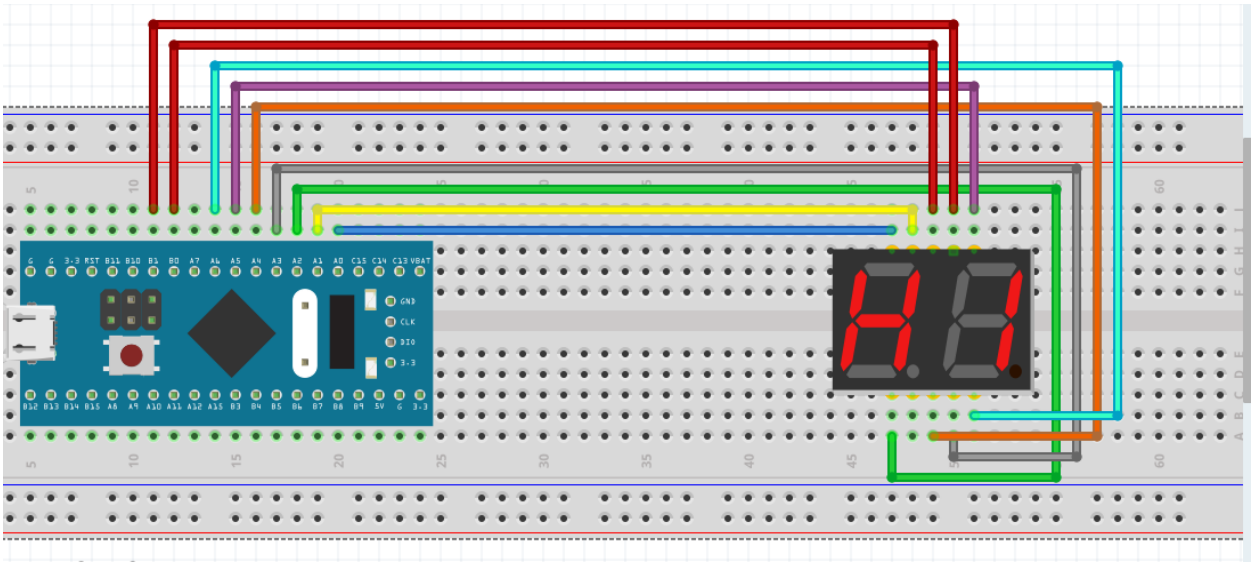
Bài 4: Điều khiển nhiều LED 7 thanh.

Ở bài toán điều khiển LED 7 thanh chúng ta có thể áp dụng theo bài toán bước 1 nối mỗi con LED 7 thanh vào 8 chân data độc lập. Tuy nhiên việc này sẽ gây lãng phí số chân điều khiển LED và limit số LED có thể điều khiển (4 chiếc). Với số LED tăng lên đủ lớn số chân cần cũng tăng lên rất nhiều. Để giải quyết bài này có một kỹ thuật nêu ra là kỹ thuật *"Quét LED"*.

Kỹ thuật Quét LED thực hiện theo nguyên tắc một thời điểm chỉ bật một LED 7 thanh với dữ liệu nó cần hiển thị, các LED còn lại được tắt. Việc quét LED thực hiện luôn phiên sáng các LED với yêu cầu trên. Có một hiện tượng hay xảy ra với những người mới thực hiện lập trình quét LED là hiện tượng "bóng ma" đó là hiện tượng xuất hiện các bóng mờ LED không mong muốn do quá trình điều khiển. Quá trình quét LED chuẩn được thực hiện theo các bước sau:

1. Xuất ra mã hiển thị.
2. Cấp nguồn cho LED muốn hiển thị.
3. Trễ 1 khoảng thời gian để duy trì sáng.
4. Cắt nguồn LED vừa hiển thị.

Các em chú ý thực hiện việc quét LED trong code theo quy trình để tránh gặp những rắc rối không mong muốn. Sau đây là một đoạn chương trình hiển thị các số từ 00-99.



Các bạn thực hiện chương trình code như sau:

// 2 led 7 thanh (anot chung)

```
/*

        A0...6 = a,b,c  ,d,e,f,g

        B0 = L7S1

        B1 = L7S2

*/



#include "stm32f10x.h"

#include "stm32f10x_gpio.h"

#include "stm32f10x_rcc.h"
```

```
void GPIO_configuration(void);

void Delay(unsigned int time);



void GPIO_configuration(void)

{

	RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);

	RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE);



	GPIO_InitTypeDef GPIO_InitStructure;



	GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;

	GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0 |GPIO_Pin_1 |GPIO_Pin_2 |GPIO_Pin_3
|GPIO_Pin_4 |GPIO_Pin_5 |GPIO_Pin_6;

	GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;

	GPIO_Init(GPIOA, &GPIO_InitStructure);



	GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0 | GPIO_Pin_1;

	GPIO_Init(GPIOB, &GPIO_InitStructure);

}
```

```
void Delay(unsigned int time){

	unsigned int i,j;

	for(i=0;i<time;i++){
		for(j=0;j< 0x2AFF; j++);
	}
}

int main(void)
{
	SystemInit();
	GPIO_configuration();
	uint16_t mang[10]={0x0040, 0x00F9, 0x0024, 0x0030, 0x0019, 0x0012, 0x0002,
0x00F8, 0x0000, 0x0010};
	unsigned char i;
	unsigned int j;
```

```
GPIO_ResetBits(GPIOB, GPIO_Pin_0);       //L7S1 = 0;
GPIO_ResetBits(GPIOB, GPIO_Pin_1); //L7S2 = 0;
while(1)
{
        for(i=0;i<100;i++)
        {
                for(j=0;j<200;j++)
                {
                        GPIOA->ODR = mang[i/10];
                        GPIO_ResetBits(GPIOB, GPIO_Pin_0);       //L7S1=0
                        Delay(1);
                        GPIO_SetBits(GPIOB, GPIO_Pin_0); //L7S1 = 1
                        GPIOA->ODR = mang[i%10];
                        GPIO_ResetBits(GPIOB, GPIO_Pin_1); //L7S2 = 0
                        Delay(1);
                        GPIO_SetBits(GPIOB, GPIO_Pin_1); //L7S2 = 1
                }
        }
```

```
        }
}
```

Trên đây chúng ta đã thực hiện theo quy trình 4 bước quét LED:

```
 GPIOA->ODR = mang[i/10];// Xuất dữ liệu
LEDGPIO_SetBits(GPIOB,  GPIO_Pin_0);  //L7S1=1
Delay(1);  // Trễ một khoảng thời gian
GPIO_ResetBits(GPIOB, GPIO_Pin_0); //L7S1 = 0
```

Ở đây ngoài vòng for thực hiện đếm từ 00-99 chúng ta cần thêm một vòng for cho biến j chạy từ 0-200 để đảm bảo mỗi số được hiển thị trong một khoảng thời gian đủ lâu để chúng ta có thể nhìn thấy. Kết quả của chương trình như sau:

Như vậy là chúng ta đã nắm được quy trình hiển thị nhiều LED 7 thanh theo phương pháp quét LED. Từ những kiến thức cơ bản này các em có thể áp dụng để hiển thị cho nhiều LED hơn cũng như hiển thị thông tin theo ý muốn của các em. Sau này khi tìm hiểu về Timer các bạn có thể kết hợp với Tỉmer để thực hiện quét LED tránh sử dụng các hàm delay gây lãng phí thời gian xử lý của vi điều khiển hoặc có thể hiển thị số theo thời gian thực (như đồng hồ điện tử)

Câu hỏi 4.1- Hiển thị các số chẵn tăng dần từ 0-99

Câu hỏi 4.2- Hiển thị các số lẻ giảm dần từ 99-0