



ELECTRONICS ENGINEERING  
ELEC335 - MICROPROCESSORS LABORATORY

LAB #3

Muhammet Cemal Eryiğit	1801022024
Şahabettin Alpcan Soydaş	1801022014
Mert Tuncay Firil	1901022285

- 1) Writing a program to blink an external LED at roughly 1 second intervals.

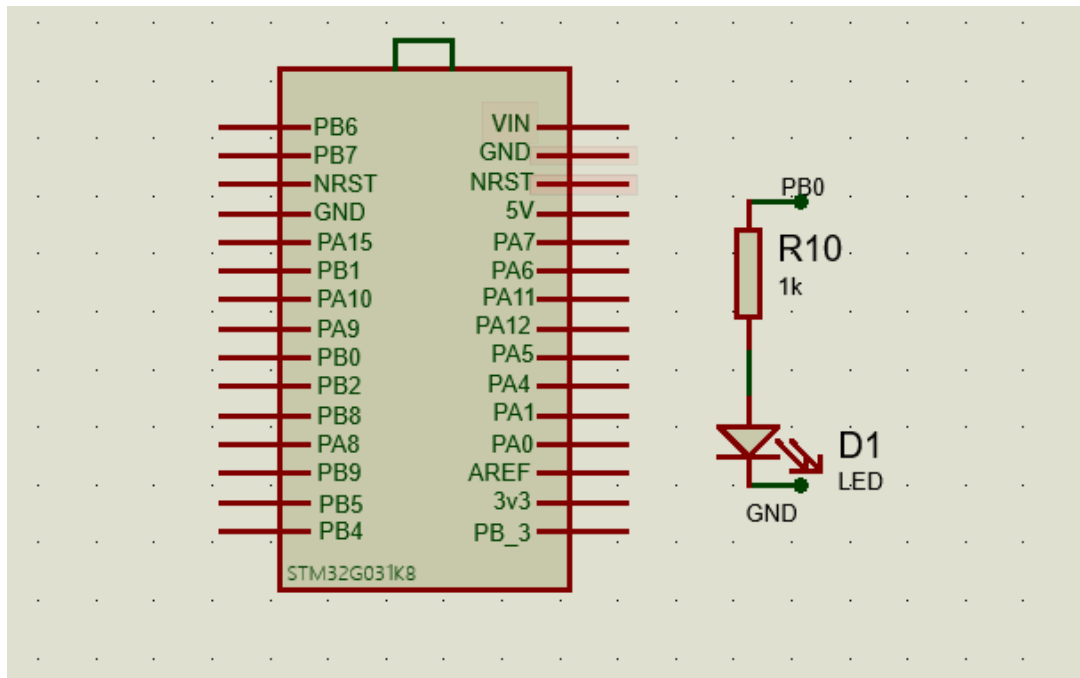


Figure 1.1: Circuit schematic for problem 1

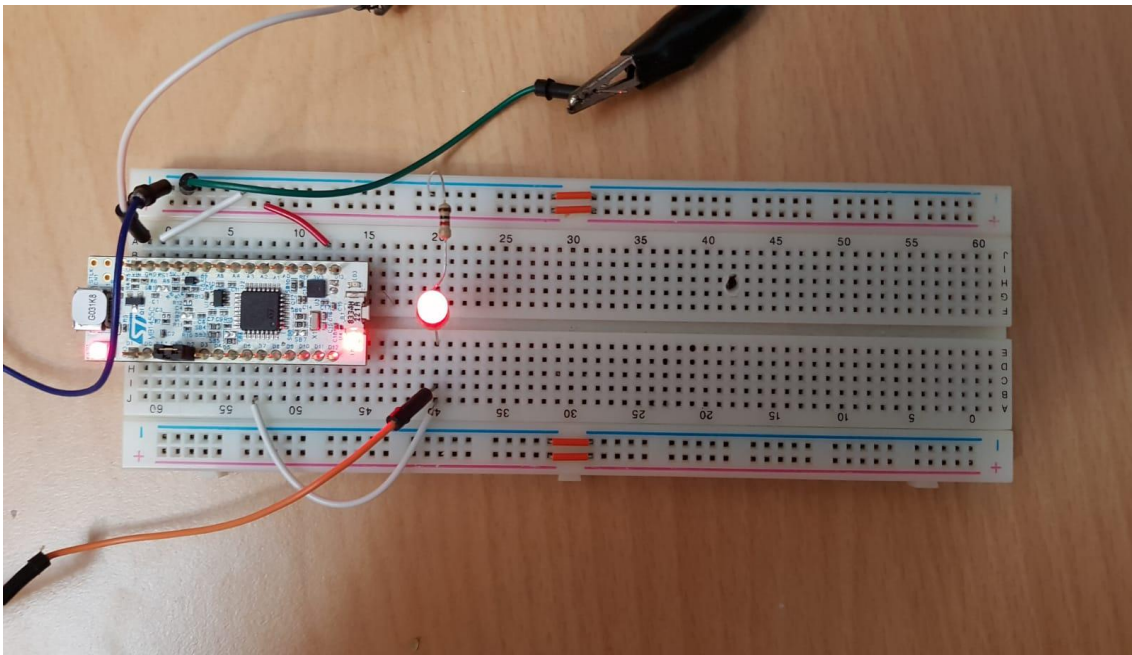


Figure 1.2: Circuit for problem 1

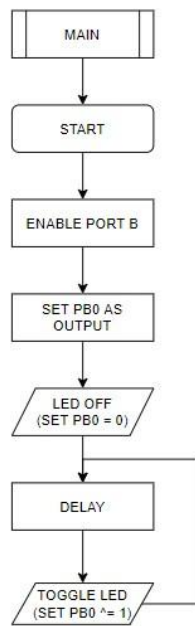


Figure 1.3 Problem 1 flow chart

1.1) Is there any difference between the code size when you implemented this in Assembly? What do you think accounts for that?

```

23:22:40 **** Incremental Build of configuration Debug for project asm ****
make -j8 all
arm-none-eabi-size  asm.elf
   text    data    bss     dec     hex filename
   212      0      0      212     d4 asm.elf
Finished building: default.size.stdout

23:22:40 Build Finished. 0 errors. 0 warnings. (took 189ms)
  
```

Figure 1.4 Assembly code size -o0 optimization

```

22:37:24 **** Incremental Build of configuration Debug for project blinky ***
make -j8 all
arm-none-eabi-size  blinky.elf
   text    data    bss     dec     hex filename
   552      8   1568   2128    850 blinky.elf
Finished building: default.size.stdout

22:37:24 Build Finished. 0 errors. 0 warnings. (took 175ms)
  
```

Figure 1.5 C code size in -o0 optimization

→ Implementing in Assembly requires less code size than implementing in C.

1.2) Is the delay number different then the assembly implementation? Explain.

→ Yes, the delay value was higher in assembly. Because there are fewer instructions in assembly code.

1.3) Change the optimization to -O2, and try again, is there any change? If so, explain what happened. Is there any difference between the code sizes?

```
23:35:48 **** Incremental Build of configuration Debug for project blinky ****
make -j8 all
arm-none-eabi-size  blinky.elf
   text    data    bss     dec     hex filename
   504      8    1568    2080     820 blinky.elf
Finished building: default.size.stdout

23:35:49 Build Finished. 0 errors, 0 warnings. (took 277ms)
```

Figure 1.6 C code size in -o2 optimization

→ Since there is less code in the -o2 optimization than the -o0 optimization, the delay value is selected more.

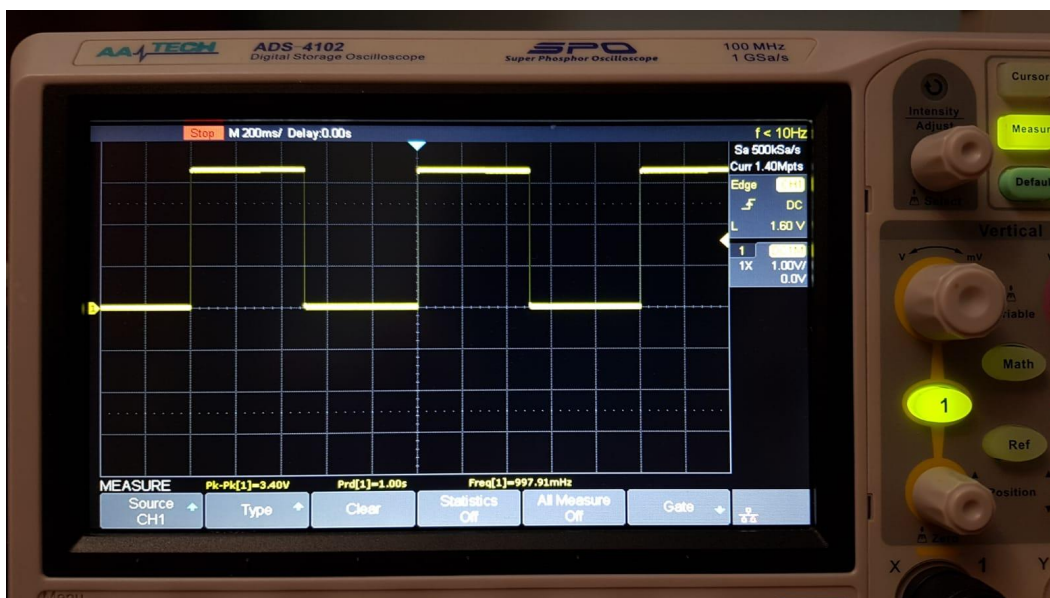


Figure 1.7 Problem 1 LED toggle oscilloscope display

- 2) Using a state machine blink the external LED at different intervals. Assign each speed to a mode, and attach a button to cycle through the modes. (Each button press will cycle through these modes.) You should do polling for the button press.

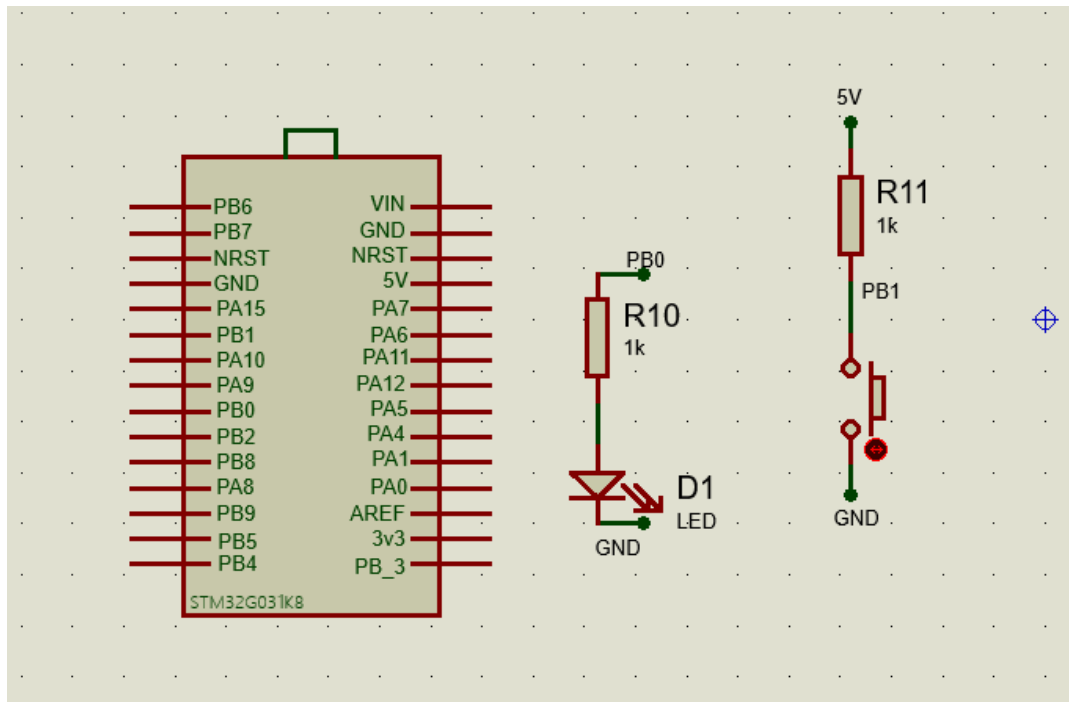


Figure 2.1: Schematic for problem 2

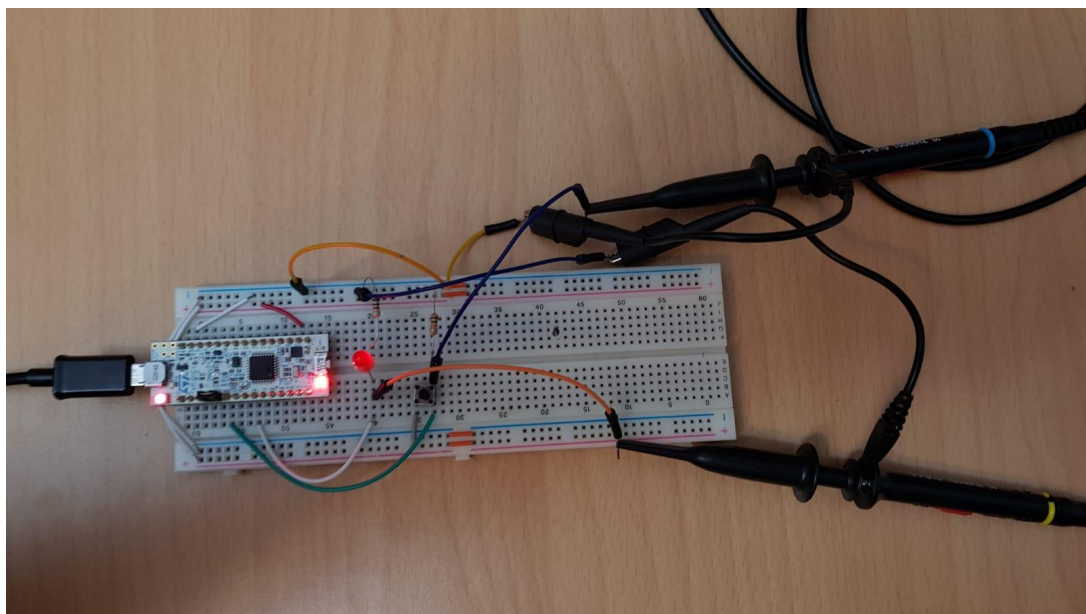
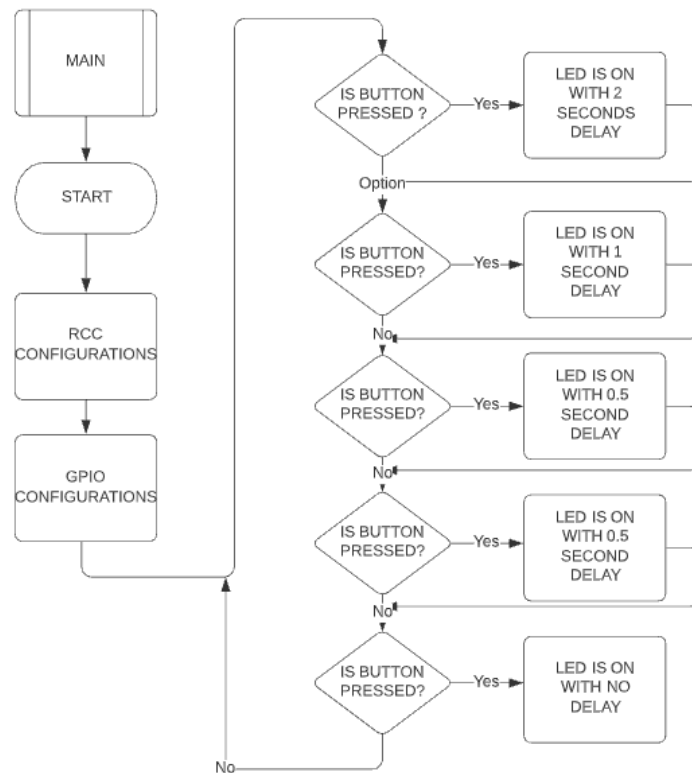


Figure 2.2: Circuit for problem 2



**Figure 2.3: Flowchart of problem 2**

**2.1)** What is the difference in code size when the optimization is enabled / disabled? How about the actual blinking speed of the LED? Is there any change? If so, what would be the difference?

```

19:07:54 **** Incremental Build of configuration Debug for project blinky ****
make -j8 all
arm-none-eabi-size  blinky.elf
   text  data  bss   dec   hex filename
    820     8  1568  2396   95c blinky.elf
Finished building: default.size.stdout

19:07:54 Build Finished. 0 errors, 0 warnings. (took 235ms)

```

**Figure 2.4 C code size in -o0 optimization**



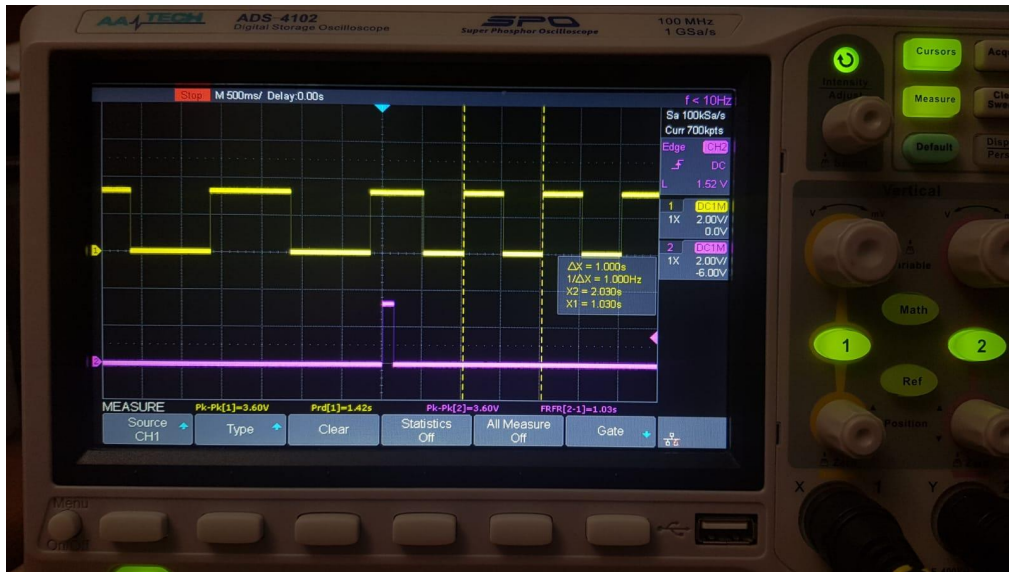


Figure 2.5 Problem 2 oscilloscope display in o0 optimization when button pressed

```
19:33:19 **** Incremental Build of configuration Debug for project blinky ****
make -j8 all
arm-none-eabi-size  blinky.elf
  text  data  bss   dec   hex filename
  720    8   1568  2296   8f8 blinky.elf
Finished building: default.size.stdout

19:33:19 Build Finished. 0 errors, 0 warnings. (took 182ms)
```

Figure 2.6 C code size in -o2 optimization

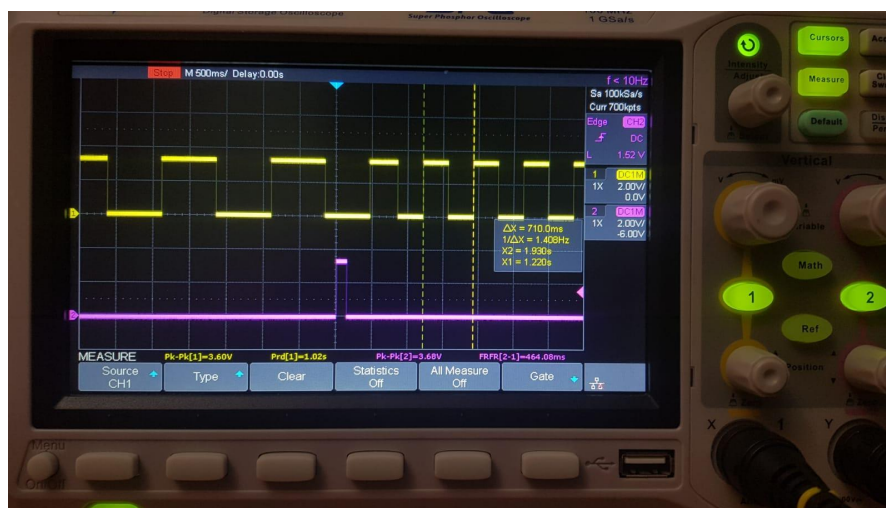


Figure 2.7 Problem 2 oscilloscope display in o2 optimization when button pressed

→ Yes, after the optimisation, the period of the LED's on-off is decreased. The reason of this situation o0 optimisation has more code lines.

- 3) Implement the same state machine in Problem 2, but this time use external interrupts to detect button press, and use the handler to change the state.

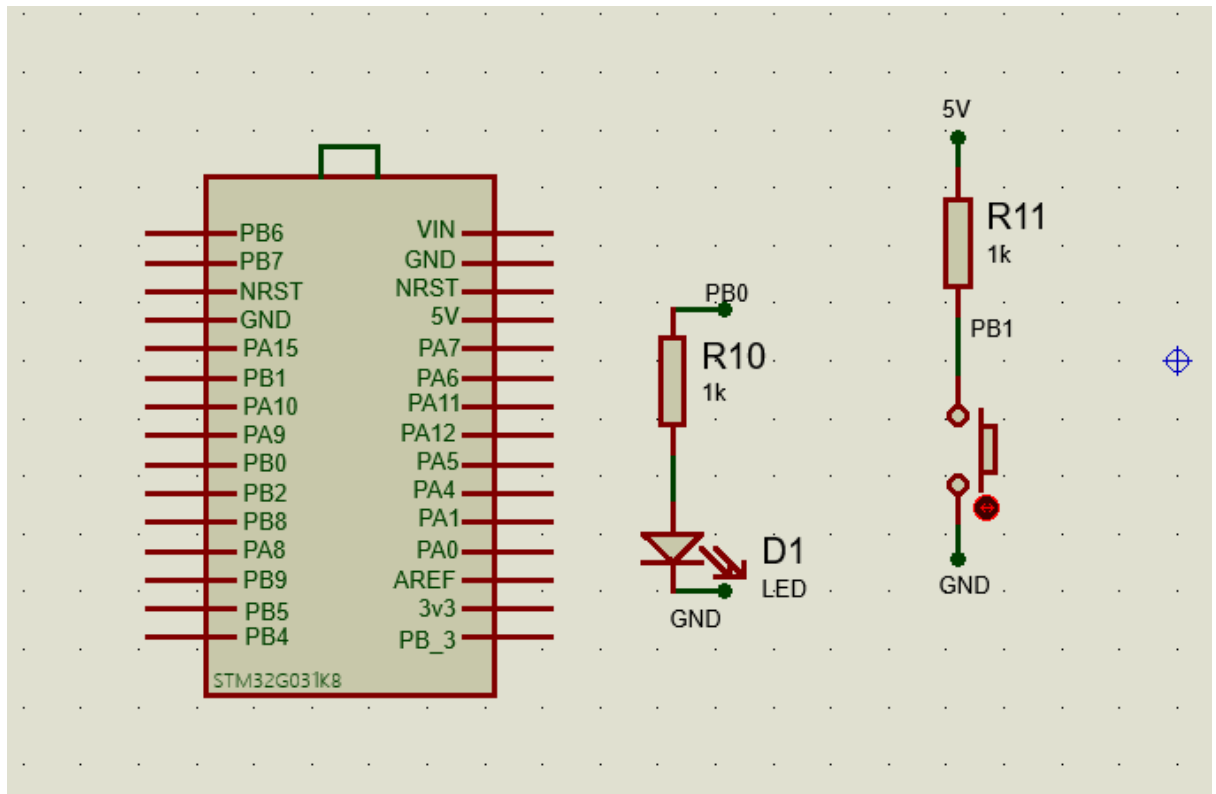


Figure 3.1: Schematic for problem 3

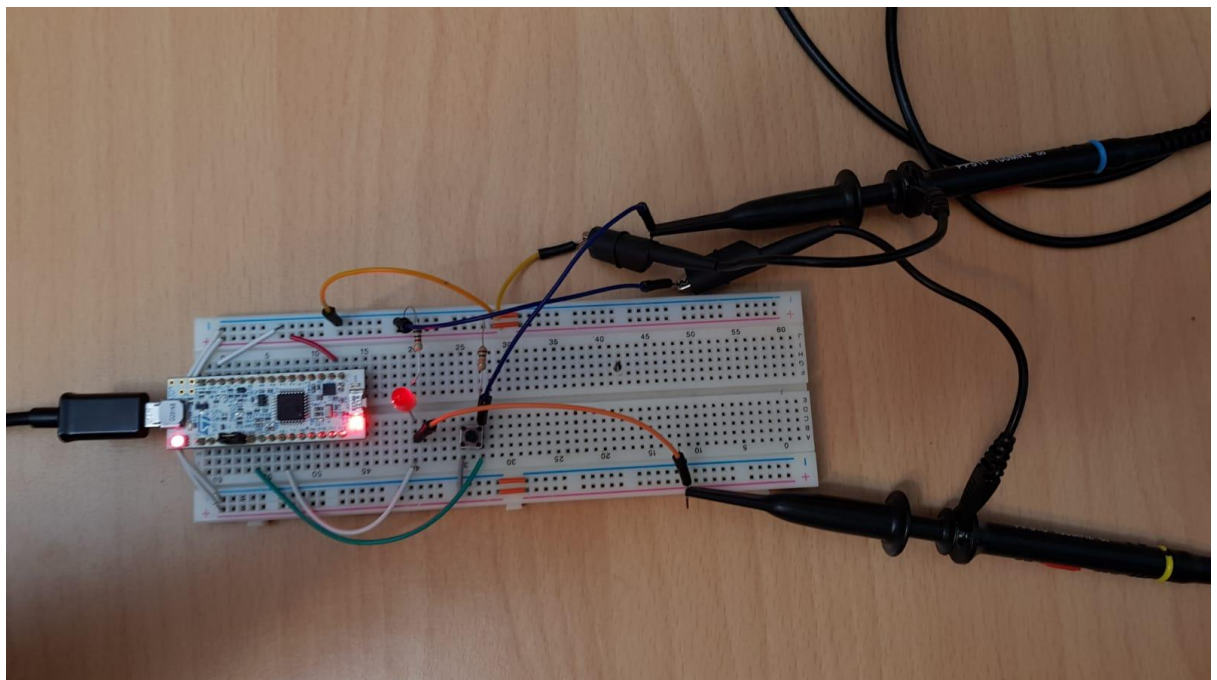
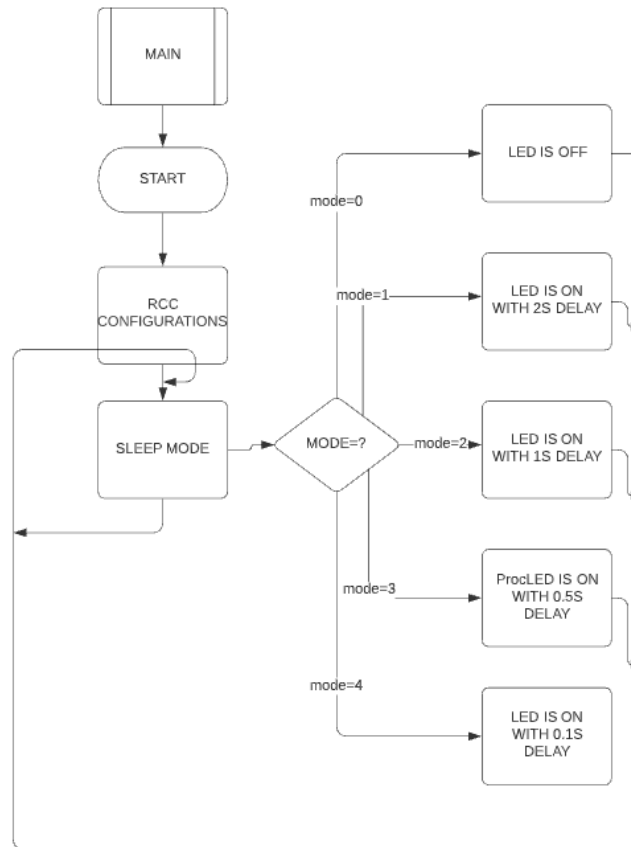


Figure 3.2: Circuit for problem 3





**Figure 3.3: Flowchart of problem 3**

**3.1)** What is the difference between Problem 2 and Problem 3 in terms of scalability, clarity and responsiveness? Compare the oscilloscope outputs for both of them and explain.

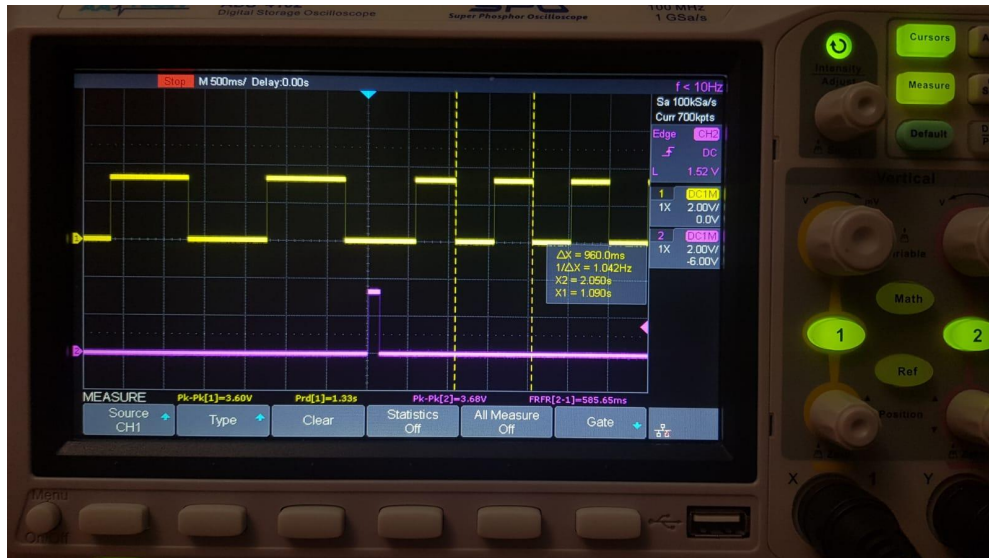
```

23:04:20 **** Incremental Build of configuration Debug for project blinky ****
make -j8 all
arm-none-eabi-size  blinky.elf
   text    data    bss     dec     hex filename
   1172      8    1576    2756    ac4 blinky.elf
Finished building: default.size.stdout

23:04:20 Build Finished. 0 errors, 0 warnings. (took 235ms)

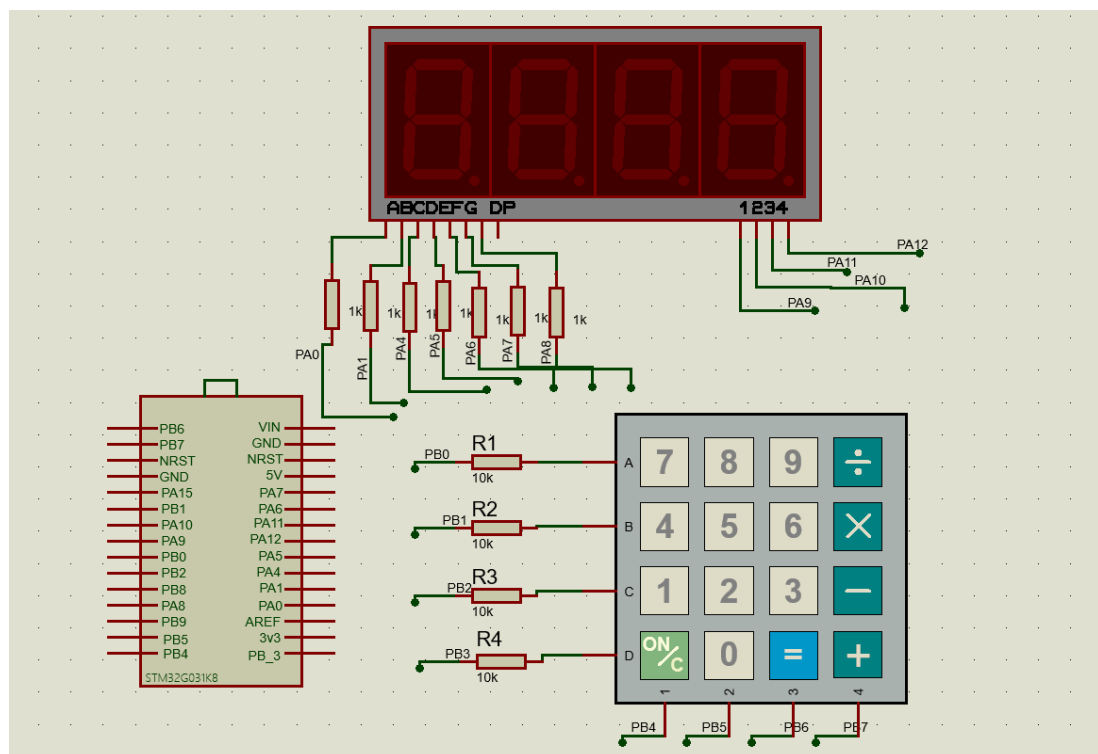
```

**Figure 3.4 C code size in -o0 optimization**

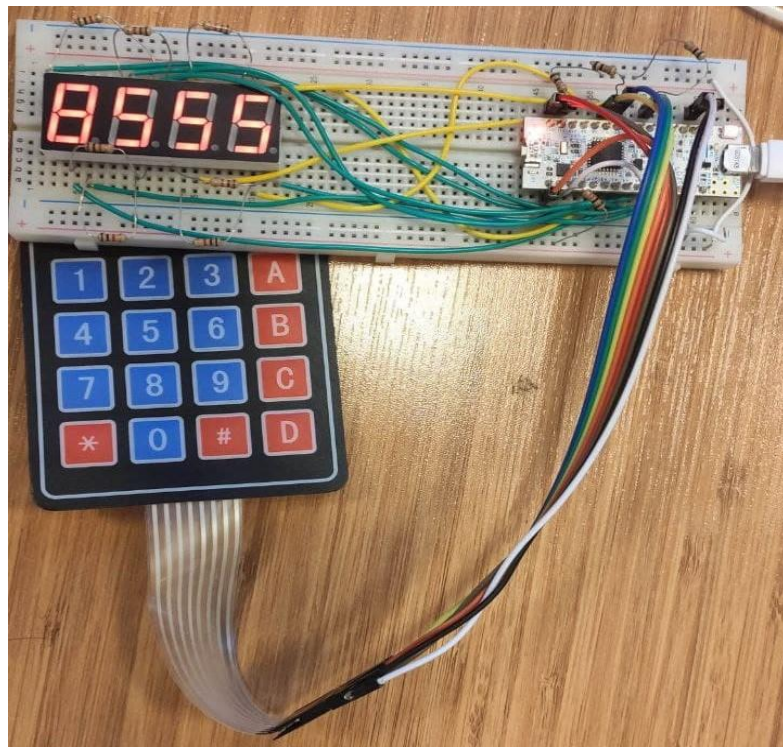


→ In problem 2, the button delay time is longer than problem 3 because the button input is read continuously. LED's on-off time is changed because in problem3 interrupt is used. To solve this problem, variables that are used for LED's on-off time settings are increased.

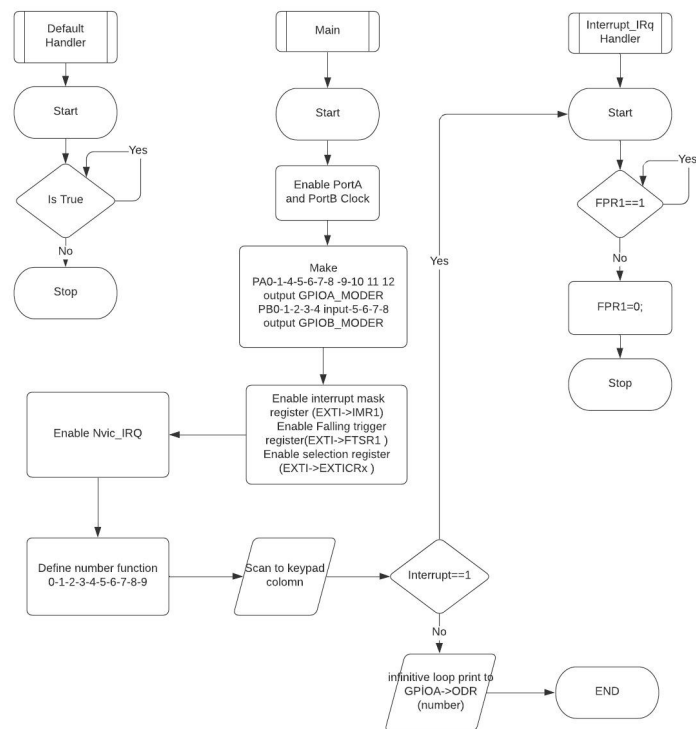
- 4) **Connect the keypad to the microcontroller, and using external interrupts detect button presses. Use an SSD to display the pressed button. Your main loop should only be used to display the SSDs.**



**Figure 4.1: Schematic for problem 4**



**Figure 4.2: Circuit for problem 4**



**Figure 4.3: Flowchart for problem 4**

**4.1)** Try to figure out the processing delay of the interrupt looking at the scope output.

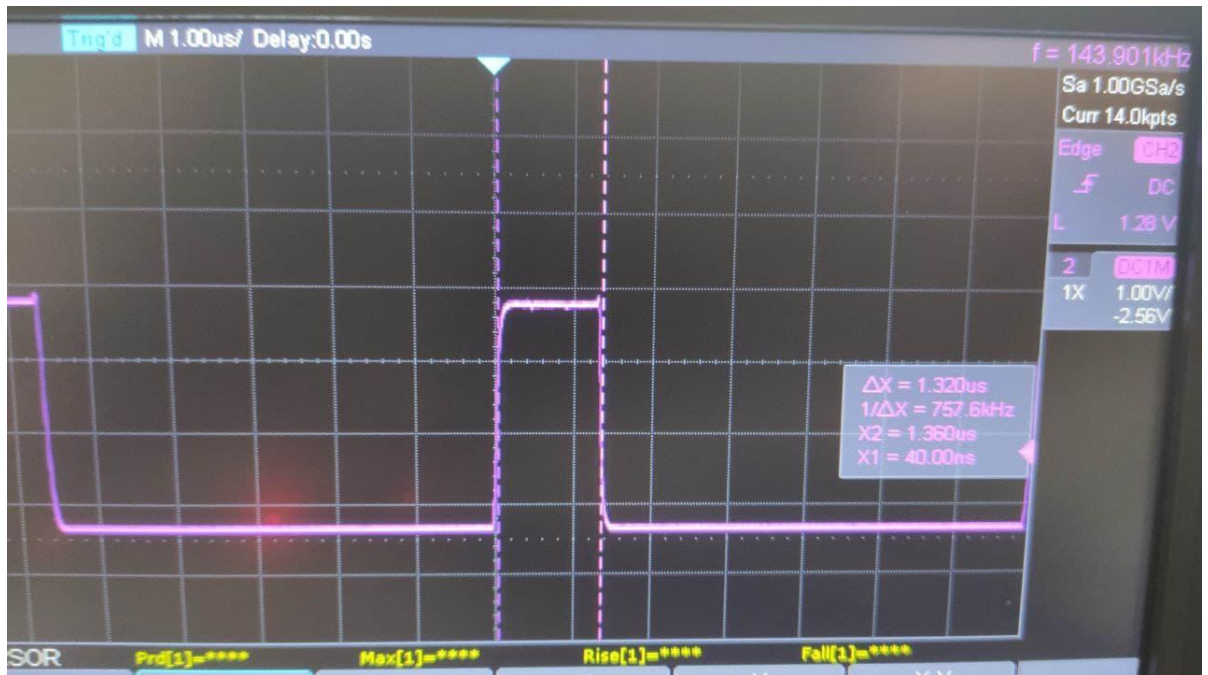


Figure 4.4: Observing interrupt delay on oscilloscope

4.2) Is there a brightness difference between the numbers Seven Segments? How did you solve it? Show the scope output of a single segment when you light up the same segment on all Seven Segments. What happens if you decrease the delay / increase the delay?

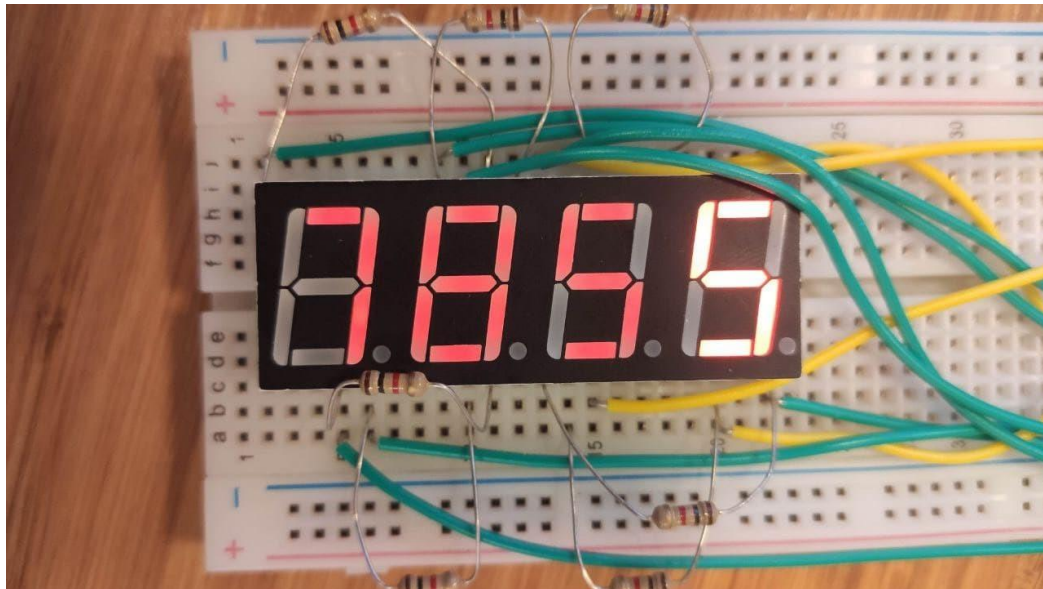


Figure 4.5: Difference brightness for each digit



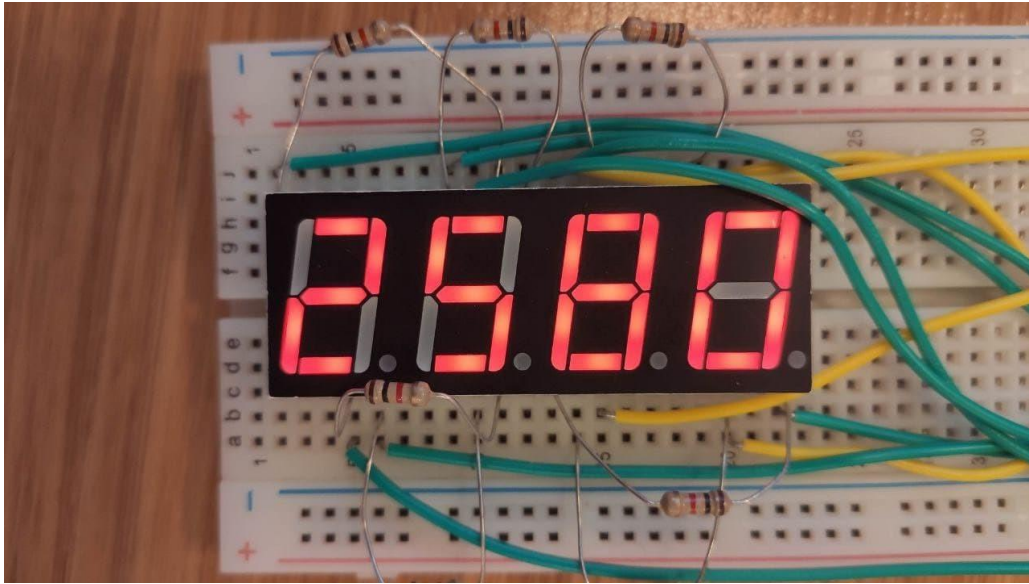


Figure 4.6: Same brightness for all digits

4) When the keys are pressed on the keypad, the microprocessor goes to interrupt. then it goes to number functions. In the number function, it saves the number in the variable and assigns it to the ODR register. When the button is pressed again, the same operations are performed from the beginning. Old number slides to another digit.

4.1- When the microprocessor enters the interrupt, the flag is triggered and goes into the interrupt as in the figure. This time takes 1.3 ms.

4.2 We print the numbers held in the variables on the screen by looping them at the same time. This loop causes the last number printed on the screen to more bright. This brightness issue is resolved by putting a short delay after the screen press for each number.

## APPENDIX

### 1) Code for problem 1

```
/*
 * main.c
 * Author: Mert Tuncay Firil
 */
#include "stm32g0xx.h"

#define LEDDELAY 804020 // Required value for delay 1 second in optimization o0
// #define LEDDELAY 2006138 // Required value for delay 1 second in optimization o2
void delay(volatile uint32_t);
```

```

int main(void) {

    /* Enable GPIOB clock */
    RCC->IOPENR = 0x2;

    /* Setup PB0 as output */
    GPIOB->MODER = 0x00000001;

    /* Turn off LED */
    GPIOB->ODR = 0x0;

    while(1) {
        delay(LEDDELAY);
        /* Toggle LED */
        GPIOB->ODR ^= 0x0001;
    }

    return 0;
}

void delay(volatile uint32_t s) {
    for(; s>0; s--);
}

```

## 2) Code for problem 2

```

/*
 * main.c
 * Author: Mert Tuncay Firil
 */
#include "stm32g0xx.h"

#define BUTTON_DELAY 200000
enum LED_LED_DELAY {delay1 = 290909,
                    delay2 = 145985,
                    delay3 = 72992,
                    delay4 = 15384
                    }; //optimization o0

/*enum LED_LED_DELAY {delay1 = 382775,
                    delay2 = 205902,
                    delay3 = 103096,
                    delay4 = 20789
                    }; //optimization o2*/

int main(void) {

    volatile uint32_t button_pressed;
    volatile uint8_t mode = 0; // for mode select
    volatile uint32_t counter = 0; // for button delay
    volatile uint32_t delay = 0; // for LED delay

```



```

RCC->IOPENR = 0x2; // Enable GPIOB clock
GPIOB->MODER = 0x00000001; // Setup PB0 as output, PB1 as input
GPIOB->ODR = 0x0; // Set PB0 low

while(1) {

    button_pressed = GPIOB->IDR & 0x0002;

    if ((counter >= BUTTON_DELAY) && (button_pressed == 0x0002)){ //Mode select
        if (mode == 5){
            mode = 0;
        }
        else{
            mode++;
        }
        counter = 0;
        delay = 0;
    }
    else{
        switch(mode){
            case 0: // No toggling, LED is off
                delay = 0;
                GPIOB->ODR = 0x0;
                break;
            case 1: //LED is toggling at roughly 2 second intervals
                if(delay >= delay1){
                    delay = 0;
                    GPIOB->ODR ^= 0x0001;
                }

                break;
            case 2: //LED is toggling at roughly 1 second intervals
                if(delay >= delay2){
                    delay = 0;
                    GPIOB->ODR ^= 0x0001;
                }
                break;
            case 3: //LED is toggling at roughly 0.5 second intervals
                if(delay >= delay3){
                    delay = 0;
                    GPIOB->ODR ^= 0x0001;
                }
                break;
            case 4: //LED is toggling at roughly 0.1 second intervals
                if(delay >= delay4){
                    delay = 0;
                    GPIOB->ODR ^= 0x0001;
                }
                break;
            case 5: //No toggling, LED is on
                delay = 0;
                GPIOB->ODR = 0x0001;
                break;
        }
    }
}

```

```

    }

    delay++;
    if(counter <= BUTTON_DELAY)
    {
        counter++;
    }
}

return 0;
}

```

### 3) Code for problem 3

```

/*
 * main.c
 * Author: Mert Tuncay Firil
 */

#include "stm32g0xx.h"

#define BUTTON_DELAY 200000
enum LED_LED_DELAY {delay1 = 342245,
                    delay2 = 170942,
                    delay3 = 85470,
                    delay4 = 17888,
                    }; //optimization o0

volatile uint8_t mode = 0; // for mode select
volatile uint32_t counter = 0; // for button delay
volatile uint32_t delay = 0; // for LED delay

void EXTI0_1_IRQHandler(void){
    if ((counter >= BUTTON_DELAY)){ //Mode select
        if (mode == 5){
            mode = 0;
        }
        else{
            mode++;
        }
        counter = 0;
        delay = 0;
    }
    EXTI->RPR1 = (1U << 1);
}

int main(void) {

    RCC->IOPENR = 0x2; // Enable GPIOB clock
    GPIOB->MODER = 0x00000001; // Setup PB0 as output, PB1 as input

```

```

GPIOB->ODR = 0x0; // Set PB0 low

EXTI->EXTICR[0] = (1U << 8*1);
EXTI->RTSR1 = (1U << 1);
EXTI->IMR1 = (1U << 1);
NVIC_SetPriority(EXTI0_1_IRQn, 0);
NVIC_EnableIRQ(EXTI0_1_IRQn);

while(1) {
    switch(mode){
        case 0: // No toggling, LED is off
            delay = 0;
            GPIOB->ODR = 0x0;
            break;
        case 1: //LED is toggling at roughly 2 second intervals
            if(delay >= delay1){
                delay = 0;
                GPIOB->ODR ^= 0x0001;
            }

            break;
        case 2: //LED is toggling at roughly 1 second intervals
            if(delay >= delay2){
                delay = 0;
                GPIOB->ODR ^= 0x0001;
            }
            break;
        case 3: //LED is toggling at roughly 0.5 second intervals
            if(delay >= delay3){
                delay = 0;
                GPIOB->ODR ^= 0x0001;
            }
            break;
        case 4: //LED is toggling at roughly 0.1 second intervals
            if(delay >= delay4){
                delay = 0;
                GPIOB->ODR ^= 0x0001;
            }
            break;
        case 5: //No toggling, LED is on
            delay = 0;
            GPIOB->ODR = 0x0001;
            break;
    }
    delay++;
    if(counter <= BUTTON_DELAY){
        counter++;
    }
}

return 0;
}

```

#### 4) Code for problem 4

```
/*
 * main.c
 * Author: Muhammed Cemal Eryigit
 */

#include "stm32g0xx.h"
#define delayms 1600

volatile uint32_t counter = 0; // represent which digit the number is in
volatile uint32_t port1 = 10; // keep number in variable port1
volatile uint32_t port2 = 10; // keep number in variable port2
volatile uint32_t port3 = 10; // keep number in variable port3
volatile uint32_t port4 = 10; // keep number in variable port4

void delay(volatile uint32_t);
void gpio_config(void);
void clock_config(void);
void EXTI_config(void);
void EXTI0_IRQHandler(void);
void check(void);
void number0(void); // number function
void number1(void); // number function
void number2(void); // number function
void number3(void); // number function
void number4(void); // number function
void number5(void); // number function
void number6(void); // number function
void number7(void); // number function
void number8(void); // number function
void number9(void); // number function

int main(void)
{
    clock_config();
    gpio_config();
    EXTI_config();
    while(1)
    {
        check();
    }
    return 0;
}

void clock_config(void)
{
    /* Enable GPIOA and GPIOB clock */
    RCC->IOPENR |= (1U << 0);
    RCC->IOPENR |= (1U << 1);
}

void EXTI_config(void)
{
    EXTI->IMR1 |= (15U << 0); // Enable interrupt mask register
    EXTI->FTSR1 = (15U << 0); // Enable Falling trigger register
    EXTI->EXTICR[0] |= (1U << 8*0); // Enable interrupt
    EXTI->EXTICR[0] |= (1U << 8*1); // Enable interrupt
}
```

```

EXTI->EXTICR[0] |= (1U << 8*2); //Enable interrupt
EXTI->EXTICR[0] |= (1U << 8*3); //Enable interrupt

NVIC_EnableIRQ(EXTI0_1_IRQn); //Enable NVIC EXTI0_1_IRQn
NVIC_EnableIRQ(EXTI2_3_IRQn); //Enable NVIC EXTI2_3_IRQn
}
void gpio_config(void)
{
    /* Setup PA0 as output */
    GPIOA->MODER &= ~(3U << 2*0);
    GPIOA->MODER |= (1U << 2*0);
    /* Setup PA1 as output */
    GPIOA->MODER &= ~(3U << 2*1);
    GPIOA->MODER |= (1U << 2*1);
    /* Setup PA4 as output */
    GPIOA->MODER &= ~(3U << 2*4);
    GPIOA->MODER |= (1U << 2*4);
    /* Setup PA5 as output */
    GPIOA->MODER &= ~(3U << 2*5);
    GPIOA->MODER |= (1U << 2*5);
    /* Setup PA6 as output */
    GPIOA->MODER &= ~(3U << 2*6);
    GPIOA->MODER |= (1U << 2*6);
    /* Setup PA7 as output */
    GPIOA->MODER &= ~(3U << 2*7);
    GPIOA->MODER |= (1U << 2*7);
    /* Setup PA8 as output */
    GPIOA->MODER &= ~(3U << 2*8);
    GPIOA->MODER |= (1U << 2*8);
    /* Setup PA9 as output */
    GPIOA->MODER &= ~(3U << 2*9);
    GPIOA->MODER |= (1U << 2*9);
    /* Setup PA10 as output */
    GPIOA->MODER &= ~(3U << 2*10);
    GPIOA->MODER |= (1U << 2*10);
    /* Setup PA11 as output */
    GPIOA->MODER &= ~(3U << 2*11);
    GPIOA->MODER |= (1U << 2*11);
    /* Setup PA12 as output */
    GPIOA->MODER &= ~(3U << 2*12);
    GPIOA->MODER |= (1U << 2*12);

    /* Setup PB0 as input */
    GPIOB->MODER &= ~(3U << 2*0);
    /* Setup PB1 as input */
    GPIOB->MODER &= ~(3U << 2*1);
    /* Setup PB2 as input */
    GPIOB->MODER &= ~(3U << 2*2);
    /* Setup PB3 as input */
    GPIOB->MODER &= ~(3U << 2*3);
    /* Setup PB4 as output */
    GPIOB->MODER &= ~(3U << 2*4);
    GPIOB->MODER |= (1U << 2*4);
    /* Setup PB5 as output */
    GPIOB->MODER &= ~(3U << 2*5);
    GPIOB->MODER |= (1U << 2*5);
    /* Setup PB6 as output */
    GPIOB->MODER &= ~(3U << 2*6);

```

```

GPIOB->MODER |= (1U << 2*6);
/* Setup PB7 as output */
GPIOB->MODER &= ~(3U << 2*7);
GPIOB->MODER |= (1U << 2*7);
}

void EXTI0_1_IRQHandler(void)
{
    if(EXTI->FPR1 & 0x1) //keypad ABCD column
    {
        EXTI->FPR1 = 0xFF; //reset Falling trigger flag
    }
    if(EXTI->FPR1 & 0x2) //keypad 963square column
    {
        //scanning to c2 column (963square)
        GPIOB->ODR |= 0x70;
        if(GPIOB->IDR == (0x7D))
        {
            number3();
            GPIOB->ODR=0x0;
        }
        else
        {
            GPIOB->ODR=0x0;
        }
        GPIOB->ODR |= 0xB0;
        if(GPIOB->IDR == (0xBD))
        {
            number6();
            GPIOB->ODR =0x0;
        }
        else
        {
            GPIOB->ODR =0x0;
        }
        GPIOB->ODR |= 0xD0;
        if(GPIOB->IDR == (0xDD))
        {
            number9();
            GPIOB->ODR=0x0;
        }
        else
        {
            GPIOB->ODR=0x0;
        }
    }
    EXTI->FPR1 = 0xFF;//reset falling trigger flag
}

void EXTI2_3_IRQHandler(void){
    if(EXTI->FPR1 & 0x4) //keypad 2580 column
    {
        //scanning to c3 column (2580)
        GPIOB->ODR |= 0x70;
    }
}

```



```

        if(GPIOB->IDR ==(0x7B))
        {
            number2();
            GPIOB->ODR=0x0;
        }
        else
        {
            GPIOB->ODR=0x0;
        }
GPIOB->ODR |=0xB0;
        if(GPIOB->IDR ==(0xBB))
        {
            number5();
            GPIOB->ODR =0x0;
        }
        else
        {
            GPIOB->ODR =0x0;
        }
GPIOB->ODR |=0xD0;
        if(GPIOB->IDR ==(0xDB))
        {
            number8();
            GPIOB->ODR=0x0;
        }
        else
        {
            GPIOB->ODR=0x0;
        }
GPIOB->ODR |=0xE0;
        if(GPIOB->IDR ==(0xEB))
        {
            number0();
            GPIOB->ODR=0x0;
        }
        else
        {
            GPIOB->ODR =0x0;
        }
    }
    if(EXTI->FPR1 & 0x8) //keypad 147* column
    {
        //scanning to c4 column (147*)
        GPIOB->ODR |=0x70;
        if(GPIOB->IDR ==(0x77))
        {
            number1();
            GPIOB->ODR=0x0;
        }
        else
        {
            GPIOB->ODR=0x0;
        }
GPIOB->ODR |=0xB0;
        if(GPIOB->IDR ==(0xB7))
        {
            number4();

```

```

        GPIOB->ODR =0x0;
    }
    else
    {
        GPIOB->ODR =0x0;
    }
    GPIOB->ODR |=0xD0;
    if(GPIOB->IDR ==(0xD7))
    {
        number7();
        GPIOB->ODR=0x0;
    }
    else
    {
        GPIOB->ODR=0x0;
    }
}
EXTI->FPR1 = 0xFF;//reset falling trigger flag
}
void check() //output to ssd
{
    for(;;)
    {
        if(port1!=10 && port2==10 && port3==10 && port4==10)
        {
            GPIOA->ODR=port1;
            delay(delayms);
        }
        else if(port1!=10 && port2!=10 && port3==10 && port4==10)
        {
            GPIOA->ODR=port1;
            delay(delayms);
            GPIOA->ODR=port2;
            delay(delayms);
        }
        else if(port1!=10 && port2!=10 && port3!=10 && port4==10)
        {
            GPIOA->ODR=port1;
            delay(delayms);
            GPIOA->ODR=port2;
            delay(delayms);
            GPIOA->ODR=port3;
            delay(delayms);
        }
        else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
        {
            GPIOA->ODR=port1;
            delay(delayms);
            GPIOA->ODR=port2;
            delay(delayms);
            GPIOA->ODR=port3;
            delay(delayms);
            GPIOA->ODR=port4;
            delay(delayms);
        }
    }
}
void number0(void)

```

```

{
    if(counter==0)
    {
        counter++;
        GPIOA->ODR &=0x00;
        if((port1==10) && (port2==10) && (port3==10) && (port4==10))
        {
            EXTI->FPR1 = 0xFF;

            GPIOA->ODR &=0x00;
            GPIOA->ODR |=0x1C00;
            GPIOA->ODR |= 0xF3;
            port1=GPIOA->ODR;
            GPIOA->ODR &=0x00;
        }
        else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
        {
            EXTI->FPR1 = 0xFF;

            port1 &=0x1FF;
            port2 &=0x1FF;
            port3 &=0x1FF;
            port4 &=0x1FF;
            port4=port3;
            port3=port2;
            port2=port1;
            port2|=0x1A00;
            port3|=0x1600;
            port4|=0xE00;
            GPIOA->ODR &=0x00;
            GPIOA->ODR |=0x1C00;
            GPIOA->ODR |= 0xF3;
            port1=GPIOA->ODR;
            GPIOA->ODR &=0x00;
        }
    }
    else if(counter==1)
    {
        counter++;
        if(port1!=10 && port2==10 && port3==10 && port4==10)
        {
            EXTI->FPR1 = 0xFF;

            port1 &=0x1FF;
            port2=port1;
            port2|=0x1A00;
            GPIOA->ODR &=0x00;
            GPIOA->ODR |=0x1C00;
            GPIOA->ODR |= 0xF3;
            port1=GPIOA->ODR;
            GPIOA->ODR &=0x00;
        }
        else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
        {
            EXTI->FPR1 = 0xFF;

            port1 &=0x1FF;
            port2 &=0x1FF;

```

```

        port3 &=0x1FF;
        port4 &=0x1FF;
        port4=port3;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0xF3;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
}
else if(counter==2)
{
    counter++;
    if(port1!=10 && port2!=10 && port3==10 && port4==10)
    {
        EXTI->FPR1 = 0xFF;

        port1 &=0x1FF;
        port2 &=0x1FF;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0xF3;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
}
else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
{
    EXTI->FPR1 = 0xFF;

    port1 &=0x1FF;
    port2 &=0x1FF;
    port3 &=0x1FF;
    port4 &=0x1FF;
    port4=port3;
    port3=port2;
    port2=port1;
    port2|=0x1A00;
    port3|=0x1600;
    port4|=0xE00;
    GPIOA->ODR &=0x00;
    GPIOA->ODR |=0x1C00;
    GPIOA->ODR |= 0xF3;
    port1=GPIOA->ODR;
    GPIOA->ODR &=0x00;
}
}
else if(counter==3)
{
    counter=0;

```

```

        if(port1!=10 && port2!=10 && port3!=10 && port4==10)
        {
            EXTI->FPR1 = 0xFF;

            port1 &=0x1FF;
            port2 &=0x1FF;
            port3 &=0x1FF;
            port4=port3;
            port3=port2;
            port2=port1;
            port2|=0x1A00;
            port3|=0x1600;
            port4|=0xE00;
            GPIOA->ODR &=0x00;
            GPIOA->ODR |=0x1C00;
            GPIOA->ODR |= 0xF3;
            port1=GPIOA->ODR;
            GPIOA->ODR &=0x00;
        }
    else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
    {
        EXTI->FPR1 = 0xFF;
        port1 &=0x1FF;
        port2 &=0x1FF;
        port3 &=0x1FF;
        port4 &=0x1FF;
        port4=port3;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0xF3;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
}
}
void number1(void)
{
    if(counter==0)
    {
        counter++;
        GPIOA->ODR &=0x00;
        if((port1==10) && (port2==10) && (port3==10) && (port4==10))
        {
            EXTI->FPR1 = 0xFF;

            GPIOA->ODR &=0x00;
            GPIOA->ODR |=0x1C00;
            GPIOA->ODR |= 0x12;
            port1=GPIOA->ODR;
            GPIOA->ODR &=0x00;
        }
    else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
    {
        EXTI->FPR1 = 0xFF;

```

```

        port1 &=0x1FF;
        port2 &=0x1FF;
        port3 &=0x1FF;
        port4 &=0x1FF;
        port4=port3;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x12;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
}
else if(counter==1)
{
    counter++;
    if(port1!=10 && port2==10 && port3==10 && port4==10)
    {
        EXTI->FPR1 = 0xFF;

        port1 &=0x1FF;
        port2=port1;
        port2|=0x1A00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x12;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
    else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
    {
        EXTI->FPR1 = 0xFF;

        port1 &=0x1FF;
        port2 &=0x1FF;
        port3 &=0x1FF;
        port4 &=0x1FF;
        port4=port3;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x12;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
}
else if(counter==2)
{
    counter++;
    if(port1!=10 && port2!=10 && port3==10 && port4==10)

```



```

        {
            EXTI->FPR1 = 0xFF;

            port1 &= 0x1FF;
            port2 &= 0x1FF;
            port3 = port2;
            port2 = port1;
            port2 |= 0x1A00;
            port3 |= 0x1600;
            GPIOA->ODR &= 0x00;
            GPIOA->ODR |= 0x1C00;
            GPIOA->ODR |= 0x12;
            port1 = GPIOA->ODR;
            GPIOA->ODR &= 0x00;
        }
    }

    else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
    {
        EXTI->FPR1 = 0xFF;
        port1 &= 0x1FF;
        port2 &= 0x1FF;
        port3 &= 0x1FF;
        port4 &= 0x1FF;
        port4 = port3;
        port3 = port2;
        port2 = port1;
        port2 |= 0x1A00;
        port3 |= 0x1600;
        port4 |= 0xE00;
        GPIOA->ODR &= 0x00;
        GPIOA->ODR |= 0x1C00;
        GPIOA->ODR |= 0x12;
        port1 = GPIOA->ODR;
        GPIOA->ODR &= 0x00;
    }

    else if(counter==3)
    {
        counter=0;
        if(port1!=10 && port2!=10 && port3!=10 && port4==10)
        {
            EXTI->FPR1 = 0xFF;

            port1 &= 0x1FF;
            port2 &= 0x1FF;
            port3 &= 0x1FF;
            port4 = port3;
            port3 = port2;
            port2 = port1;
            port2 |= 0x1A00;
            port3 |= 0x1600;
            port4 |= 0xE00;
            GPIOA->ODR &= 0x00;
            GPIOA->ODR |= 0x1C00;
            GPIOA->ODR |= 0x12;
            port1 = GPIOA->ODR;
            GPIOA->ODR &= 0x00;
        }
        else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)

```

```

        {
            EXTI->FPR1 = 0xFF;
            port1 &= 0x1FF;
            port2 &= 0x1FF;
            port3 &= 0x1FF;
            port4 &= 0x1FF;
            port4=port3;
            port3=port2;
            port2=port1;
            port2|=0x1A00;
            port3|=0x1600;
            port4|=0xE00;
            GPIOA->ODR &= 0x00;
            GPIOA->ODR |= 0x1C00;
            GPIOA->ODR |= 0x12;
            port1=GPIOA->ODR;
            GPIOA->ODR &= 0x00;
        }
    }

}

void number2(void)
{
    if(counter==0)
    {
        counter++;
        GPIOA->ODR &= 0x00;
        if((port1==10) && (port2==10) && (port3==10) && (port4==10))
        {
            EXTI->FPR1 = 0xFF;

            GPIOA->ODR &= 0x00;
            GPIOA->ODR |= 0x1C00;
            GPIOA->ODR |= 0x163;
            port1=GPIOA->ODR;
            GPIOA->ODR &= 0x00;
        }
        else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
        {
            EXTI->FPR1 = 0xFF;

            port1 &= 0x1FF;
            port2 &= 0x1FF;
            port3 &= 0x1FF;
            port4 &= 0x1FF;
            port4=port3;
            port3=port2;
            port2=port1;
            port2|=0x1A00;
            port3|=0x1600;
            port4|=0xE00;
            GPIOA->ODR &= 0x00;
            GPIOA->ODR |= 0x1C00;
            GPIOA->ODR |= 0x163;
            port1=GPIOA->ODR;
            GPIOA->ODR &= 0x00;
        }
    }
}

else if(counter==1)

```

```

{
    counter++;
    if(port1!=10 && port2==10 && port3==10 && port4==10)
    {
        EXTI->FPR1 = 0xFF;

        port1 &=0x1FF;
        port2=port1;
        port2|=0x1A00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x163;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
    else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
    {
        EXTI->FPR1 = 0xFF;
        port1 &=0x1FF;
        port2 &=0x1FF;
        port3 &=0x1FF;
        port4 &=0x1FF;
        port4=port3;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x163;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
}
else if(counter==2)
{
    counter++;
    if(port1!=10 && port2!=10 && port3==10 && port4==10)
    {
        EXTI->FPR1 = 0xFF;

        port1 &=0x1FF;
        port2 &=0x1FF;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x163;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
}
else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
{
    EXTI->FPR1 = 0xFF;
    port1 &=0x1FF;

```

```

        port2 &=0x1FF;
        port3 &=0x1FF;
        port4 &=0x1FF;
        port4=port3;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x163;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
else if(counter==3)
{
    counter=0;
    if(port1!=10 && port2!=10 && port3!=10 && port4==10)
    {
        EXTI->FPR1 = 0xFF;

        port1 &=0x1FF;
        port2 &=0x1FF;
        port3 &=0x1FF;
        port4=port3;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x163;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
{
    EXTI->FPR1 = 0xFF;
    port1 &=0x1FF;
    port2 &=0x1FF;
    port3 &=0x1FF;
    port4 &=0x1FF;
    port4=port3;
    port3=port2;
    port2=port1;
    port2|=0x1A00;
    port3|=0x1600;
    port4|=0xE00;
    GPIOA->ODR &=0x00;
    GPIOA->ODR |=0x1C00;
    GPIOA->ODR |= 0x163;
    port1=GPIOA->ODR;
    GPIOA->ODR &=0x00;
}
}
}

```

```

void number3(void)
{
    if(counter==0)
    {
        counter++;
        GPIOA->ODR &=0x00;
        if((port1==10) && (port2==10) && (port3==10) && (port4==10))
        {
            EXTI->FPR1 = 0xFF;

            GPIOA->ODR &=0x00;
            GPIOA->ODR |=0x1C00;
            GPIOA->ODR |= 0x133;
            port1=GPIOA->ODR;
            GPIOA->ODR &=0x00;
        }
        else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
        {
            EXTI->FPR1 = 0xFF;

            port1 &=0x1FF;
            port2 &=0x1FF;
            port3 &=0x1FF;
            port4 &=0x1FF;
            port4=port3;
            port3=port2;
            port2=port1;
            port2|=0x1A00;
            port3|=0x1600;
            port4|=0xE00;
            GPIOA->ODR &=0x00;
            GPIOA->ODR |=0x1C00;
            GPIOA->ODR |= 0x133;
            port1=GPIOA->ODR;
            GPIOA->ODR &=0x00;
        }
    }

    else if(counter==1)
    {
        counter++;
        if(port1!=10 && port2==10 && port3==10 && port4==10)
        {
            EXTI->FPR1 = 0xFF;

            port1 &=0x1FF;
            port2=port1;
            port2|=0x1A00;
            GPIOA->ODR &=0x00;
            GPIOA->ODR |=0x1C00;
            GPIOA->ODR |= 0x133;
            port1=GPIOA->ODR;
            GPIOA->ODR &=0x00;
        }
        else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
        {
            EXTI->FPR1 = 0xFF;
            port1 &=0x1FF;

```

```

        port2 &=0x1FF;
        port3 &=0x1FF;
        port4 &=0x1FF;
        port4=port3;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x133;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
}
else if(counter==2)
{
    counter++;
    if(port1!=10 && port2!=10 && port3==10 && port4==10)
    {
        EXTI->FPR1 = 0xFF;

        port1 &=0x1FF;
        port2 &=0x1FF;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x133;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
}
else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
{
    EXTI->FPR1 = 0xFF;
    port1 &=0x1FF;
    port2 &=0x1FF;
    port3 &=0x1FF;
    port4 &=0x1FF;
    port4=port3;
    port3=port2;
    port2=port1;
    port2|=0x1A00;
    port3|=0x1600;
    port4|=0xE00;
    GPIOA->ODR &=0x00;
    GPIOA->ODR |=0x1C00;
    GPIOA->ODR |= 0x133;
    port1=GPIOA->ODR;
    GPIOA->ODR &=0x00;
}
}

else if(counter==3)
{

```



```

counter=0;
if(port1!=10 && port2!=10 && port3!=10 && port4==10)
{
    EXTI->FPR1 = 0xFF;

    port1 &=0x1FF;
    port2 &=0x1FF;
    port3 &=0x1FF;
    port4=port3;
    port3=port2;
    port2=port1;
    port2|=0x1A00;
    port3|=0x1600;
    port4|=0xE00;
    GPIOA->ODR &=0x00;
    GPIOA->ODR |=0x1C00;
    GPIOA->ODR |= 0x133;
    port1=GPIOA->ODR;
    GPIOA->ODR &=0x00;
}
else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
{
    EXTI->FPR1 = 0xFF;
    port1 &=0x1FF;
    port2 &=0x1FF;
    port3 &=0x1FF;
    port4 &=0x1FF;
    port4=port3;
    port3=port2;
    port2=port1;
    port2|=0x1A00;
    port3|=0x1600;
    port4|=0xE00;
    GPIOA->ODR &=0x00;
    GPIOA->ODR |=0x1C00;
    GPIOA->ODR |= 0x133;
    port1=GPIOA->ODR;
    GPIOA->ODR &=0x00;
}
}

}
void number4(void)
{
    if(counter==0)
    {
        counter++;
        GPIOA->ODR &=0x00;
        if((port1==10) && (port2==10) && (port3==10) && (port4==10))
        {
            EXTI->FPR1 = 0xFF;

            GPIOA->ODR &=0x00;
            GPIOA->ODR |=0x1C00;
            GPIOA->ODR |= 0x192;
            port1=GPIOA->ODR;
            GPIOA->ODR &=0x00;
        }
        else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)

```

```

        {
            EXTI->FPR1 = 0xFF;

            port1 &= 0x1FF;
            port2 &= 0x1FF;
            port3 &= 0x1FF;
            port4 &= 0x1FF;
            port4=port3;
            port3=port2;
            port2=port1;
            port2|=0x1A00;
            port3|=0x1600;
            port4|=0xE00;
            GPIOA->ODR &= 0x00;
            GPIOA->ODR |= 0x1C00;
            GPIOA->ODR |= 0x192;
            port1=GPIOA->ODR;
            GPIOA->ODR &= 0x00;
        }

    }
    else if(counter==1)
    {
        counter++;
        if(port1!=10 && port2==10 && port3==10 && port4==10)
        {
            EXTI->FPR1 = 0xFF;

            port1 &= 0x1FF;
            port2=port1;
            port2|=0x1A00;
            GPIOA->ODR &= 0x00;
            GPIOA->ODR |= 0x1C00;
            GPIOA->ODR |= 0x192;
            port1=GPIOA->ODR;
            GPIOA->ODR &= 0x00;
        }
        else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
        {
            EXTI->FPR1 = 0xFF;
            port1 &= 0x1FF;
            port2 &= 0x1FF;
            port3 &= 0x1FF;
            port4 &= 0x1FF;
            port4=port3;
            port3=port2;
            port2=port1;
            port2|=0x1A00;
            port3|=0x1600;
            port4|=0xE00;
            GPIOA->ODR &= 0x00;
            GPIOA->ODR |= 0x1C00;
            GPIOA->ODR |= 0x192;
            port1=GPIOA->ODR;
            GPIOA->ODR &= 0x00;
        }
    }
    else if(counter==2)

```

```

{
    counter++;
    if(port1!=10 && port2!=10 && port3==10 && port4==10)
    {
        EXTI->FPR1 = 0xFF;

        port1 &=0x1FF;
        port2 &=0x1FF;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x192;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
}
else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
{
    EXTI->FPR1 = 0xFF;
    port1 &=0x1FF;
    port2 &=0x1FF;
    port3 &=0x1FF;
    port4 &=0x1FF;
    port4=port3;
    port3=port2;
    port2=port1;
    port2|=0x1A00;
    port3|=0x1600;
    port4|=0xE00;
    GPIOA->ODR &=0x00;
    GPIOA->ODR |=0x1C00;
    GPIOA->ODR |= 0x192;
    port1=GPIOA->ODR;
    GPIOA->ODR &=0x00;
}
}

else if(counter==3)
{
    counter=0;
    if(port1!=10 && port2!=10 && port3!=10 && port4==10)
    {
        EXTI->FPR1 = 0xFF;

        port1 &=0x1FF;
        port2 &=0x1FF;
        port3 &=0x1FF;
        port4=port3;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x192;
    }
}

```

```

        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
    else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
    {
        EXTI->FPR1 = 0xFF;

        port1 &=0x1FF;
        port2 &=0x1FF;
        port3 &=0x1FF;
        port4 &=0x1FF;
        port4=port3;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x192;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
}

void number5(void)
{
    if(counter==0)
    {
        counter++;
        GPIOA->ODR &=0x00;
        if((port1==10) && (port2==10) && (port3==10) && (port4==10))
        {
            EXTI->FPR1 = 0xFF;

            GPIOA->ODR &=0x00;
            GPIOA->ODR |=0x1C00;
            GPIOA->ODR |= 0x1B1;
            port1=GPIOA->ODR;
            GPIOA->ODR &=0x00;
        }
        else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
        {
            EXTI->FPR1 = 0xFF;

            port1 &=0x1FF;
            port2 &=0x1FF;
            port3 &=0x1FF;
            port4 &=0x1FF;
            port4=port3;
            port3=port2;
            port2=port1;
            port2|=0x1A00;
            port3|=0x1600;
            port4|=0xE00;
            GPIOA->ODR &=0x00;
            GPIOA->ODR |=0x1C00;
            GPIOA->ODR |= 0x1B1;
            port1=GPIOA->ODR;

```

```

        GPIOA->ODR &=0x00;
    }

}
else if(counter==1)
{
    counter++;
    if(port1!=10 && port2==10 && port3==10 && port4==10)
    {
        EXTI->FPR1 = 0xFF;

        port1 &=0x1FF;
        port2=port1;
        port2|=0x1A00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x1B1;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
    else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
    {
        EXTI->FPR1 = 0xFF;
        port1 &=0x1FF;
        port2 &=0x1FF;
        port3 &=0x1FF;
        port4 &=0x1FF;
        port4=port3;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x1B1;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
}
}
else if(counter==2)
{
    counter++;
    if(port1!=10 && port2!=10 && port3==10 && port4==10)
    {
        EXTI->FPR1 = 0xFF;

        port1 &=0x1FF;
        port2 &=0x1FF;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x1B1;
        port1=GPIOA->ODR;
    }
}
}

```

```

        GPIOA->ODR &=0x00;
    }
}
else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
{
    EXTI->FPR1 = 0xFF;
    port1 &=0x1FF;
    port2 &=0x1FF;
    port3 &=0x1FF;
    port4 &=0x1FF;
    port4=port3;
    port3=port2;
    port2=port1;
    port2|=0x1A00;
    port3|=0x1600;
    port4|=0xE00;
    GPIOA->ODR &=0x00;
    GPIOA->ODR |=0x1C00;
    GPIOA->ODR |= 0x1B1;
    port1=GPIOA->ODR;
    GPIOA->ODR &=0x00;
}

else if(counter==3)
{
    counter=0;
    if(port1!=10 && port2!=10 && port3!=10 && port4==10)
    {
        EXTI->FPR1 = 0xFF;

        port1 &=0x1FF;
        port2 &=0x1FF;
        port3 &=0x1FF;
        port4=port3;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x1B1;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
}
else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
{
    EXTI->FPR1 = 0xFF;
    port1 &=0x1FF;
    port2 &=0x1FF;
    port3 &=0x1FF;
    port4 &=0x1FF;
    port4=port3;
    port3=port2;
    port2=port1;
    port2|=0x1A00;
    port3|=0x1600;
    port4|=0xE00;
    GPIOA->ODR &=0x00;
}

```

```

        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x1B1;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
}

void number6(void)
{
    if(counter==0)
    {
        counter++;
        GPIOA->ODR &=0x00;
        if((port1==10) && (port2==10) && (port3==10) && (port4==10))
        {
            EXTI->FPR1 = 0xFF;

            GPIOA->ODR &=0x00;
            GPIOA->ODR |=0x1C00;
            GPIOA->ODR |= 0x1F1;
            port1=GPIOA->ODR;
            GPIOA->ODR &=0x00;
        }
        else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
        {
            EXTI->FPR1 = 0xFF;

            port1 &=0x1FF;
            port2 &=0x1FF;
            port3 &=0x1FF;
            port4 &=0x1FF;
            port4=port3;
            port3=port2;
            port2=port1;
            port2|=0x1A00;
            port3|=0x1600;
            port4|=0xE00;
            GPIOA->ODR &=0x00;
            GPIOA->ODR |=0x1C00;
            GPIOA->ODR |= 0x1F1;
            port1=GPIOA->ODR;
            GPIOA->ODR &=0x00;
        }
    }
    else if(counter==1)
    {
        counter++;
        if(port1!=10 && port2==10 && port3==10 && port4==10)
        {
            EXTI->FPR1 = 0xFF;

            port1 &=0x1FF;
            port2=port1;
            port2|=0x1A00;
            GPIOA->ODR &=0x00;

```

```

        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x1F1;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
    else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
    {
        EXTI->FPR1 = 0xFF;
        port1 &=0x1FF;
        port2 &=0x1FF;
        port3 &=0x1FF;
        port4 &=0x1FF;
        port4=port3;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x1F1;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
}
else if(counter==2)
{
    counter++;
    if(port1!=10 && port2!=10 && port3==10 && port4==10)
    {
        EXTI->FPR1 = 0xFF;

        port1 &=0x1FF;
        port2 &=0x1FF;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x1F1;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
}
else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
{
    EXTI->FPR1 = 0xFF;
    port1 &=0x1FF;
    port2 &=0x1FF;
    port3 &=0x1FF;
    port4 &=0x1FF;
    port4=port3;
    port3=port2;
    port2=port1;
    port2|=0x1A00;
    port3|=0x1600;
    port4|=0xE00;

```



```

        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x1F1;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }

else if(counter==3)
{
    counter=0;
    if(port1!=10 && port2!=10 && port3!=10 && port4==10)
    {
        EXTI->FPR1 = 0xFF;

        port1 &=0x1FF;
        port2 &=0x1FF;
        port3 &=0x1FF;
        port4=port3;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x1F1;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
    else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
    {
        EXTI->FPR1 = 0xFF;
        port1 &=0x1FF;
        port2 &=0x1FF;
        port3 &=0x1FF;
        port4 &=0x1FF;
        port4=port3;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x1F1;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
}

}
void number7(void)
{
    if(counter==0)
    {
        counter++;
        GPIOA->ODR &=0x00;
        if((port1==10) && (port2==10) && (port3==10) && (port4==10))
        {

```

```

EXTI->FPR1 = 0xFF;

GPIOA->ODR &=0x00;
GPIOA->ODR |=0x1C00;
GPIOA->ODR |= 0x13;
port1=GPIOA->ODR;
GPIOA->ODR &=0x00;
}
else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
{
    EXTI->FPR1 = 0xFF;

    port1 &=0x1FF;
    port2 &=0x1FF;
    port3 &=0x1FF;
    port4 &=0x1FF;
    port4=port3;
    port3=port2;
    port2=port1;
    port2|=0x1A00;
    port3|=0x1600;
    port4|=0xE00;
    GPIOA->ODR &=0x00;
    GPIOA->ODR |=0x1C00;
    GPIOA->ODR |= 0x13;
    port1=GPIOA->ODR;
    GPIOA->ODR &=0x00;
}

}
else if(counter==1)
{
    counter++;
    if(port1!=10 && port2==10 && port3==10 && port4==10)
    {
        EXTI->FPR1 = 0xFF;

        port1 &=0x1FF;
        port2=port1;
        port2|=0x1A00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x13;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
    else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
    {
        EXTI->FPR1 = 0xFF;
        port1 &=0x1FF;
        port2 &=0x1FF;
        port3 &=0x1FF;
        port4 &=0x1FF;
        port4=port3;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
    }
}

```

```

        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x13;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
}
else if(counter==2)
{
    counter++;
    if(port1!=10 && port2!=10 && port3==10 && port4==10)
    {
        EXTI->FPR1 = 0xFF;

        port1 &=0x1FF;
        port2 &=0x1FF;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x13;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
}
else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
{
    EXTI->FPR1 = 0xFF;
    port1 &=0x1FF;
    port2 &=0x1FF;
    port3 &=0x1FF;
    port4 &=0x1FF;
    port4=port3;
    port3=port2;
    port2=port1;
    port2|=0x1A00;
    port3|=0x1600;
    port4|=0xE00;
    GPIOA->ODR &=0x00;
    GPIOA->ODR |=0x1C00;
    GPIOA->ODR |= 0x13;
    port1=GPIOA->ODR;
    GPIOA->ODR &=0x00;
}
}

else if(counter==3)
{
    counter=0;
    if(port1!=10 && port2!=10 && port3!=10 && port4==10)
    {
        EXTI->FPR1 = 0xFF;

        port1 &=0x1FF;
        port2 &=0x1FF;
        port3 &=0x1FF;

```

```

        port4=port3;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x13;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
{
    EXTI->FPR1 = 0xFF;
    port1 &=0x1FF;
    port2 &=0x1FF;
    port3 &=0x1FF;
    port4 &=0x1FF;
    port4=port3;
    port3=port2;
    port2=port1;
    port2|=0x1A00;
    port3|=0x1600;
    port4|=0xE00;
    GPIOA->ODR &=0x00;
    GPIOA->ODR |=0x1C00;
    GPIOA->ODR |= 0x13;
    port1=GPIOA->ODR;
    GPIOA->ODR &=0x00;
}
}
}
void number8(void)
{
    if(counter==0)
    {
        counter++;
        GPIOA->ODR &=0x00;
        if((port1==10) && (port2==10) && (port3==10) && (port4==10))
        {
            EXTI->FPR1 = 0xFF;

            GPIOA->ODR &=0x00;
            GPIOA->ODR |=0x1C00;
            GPIOA->ODR |= 0x1F3;
            port1=GPIOA->ODR;
            GPIOA->ODR &=0x00;
        }
        else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
        {
            EXTI->FPR1 = 0xFF;

            port1 &=0x1FF;
            port2 &=0x1FF;
            port3 &=0x1FF;
            port4 &=0x1FF;
            port4=port3;
            port3=port2;

```

```

        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x1F3;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }

}
else if(counter==1)
{
    counter++;
    if(port1!=10 && port2==10 && port3==10 && port4==10)
    {
        EXTI->FPR1 = 0xFF;

        port1 &=0x1FF;
        port2=port1;
        port2|=0x1A00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x1F3;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
    else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
    {
        EXTI->FPR1 = 0xFF;
        port1 &=0x1FF;
        port2 &=0x1FF;
        port3 &=0x1FF;
        port4 &=0x1FF;
        port4=port3;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x1F3;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
}
else if(counter==2)
{
    counter++;
    if(port1!=10 && port2!=10 && port3==10 && port4==10)
    {
        EXTI->FPR1 = 0xFF;

        port1 &=0x1FF;
        port2 &=0x1FF;

```

```

        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x1F3;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
}
else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
{
    EXTI->FPR1 = 0xFF;
    port1 &=0x1FF;
    port2 &=0x1FF;
    port3 &=0x1FF;
    port4 &=0x1FF;
    port4=port3;
    port3=port2;
    port2=port1;
    port2|=0x1A00;
    port3|=0x1600;
    port4|=0xE00;
    GPIOA->ODR &=0x00;
    GPIOA->ODR |=0x1C00;
    GPIOA->ODR |= 0x1F3;
    port1=GPIOA->ODR;
    GPIOA->ODR &=0x00;
}

else if(counter==3)
{
    counter=0;
    if(port1!=10 && port2!=10 && port3!=10 && port4==10)
    {
        EXTI->FPR1 = 0xFF;

        port1 &=0x1FF;
        port2 &=0x1FF;
        port3 &=0x1FF;
        port4=port3;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x1F3;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
    else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
    {
        EXTI->FPR1 = 0xFF;
        port1 &=0x1FF;
        port2 &=0x1FF;
        port3 &=0x1FF;

```

```

        port4 &=0x1FF;
        port4=port3;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x1F3;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }

}

void number9(void)
{
    if(counter==0)
    {
        counter++;
        GPIOA->ODR &=0x00;
        if((port1==10) && (port2==10) && (port3==10) && (port4==10))
        {
            EXTI->FPR1 = 0xFF;

            GPIOA->ODR &=0x00;
            GPIOA->ODR |=0x1C00;
            GPIOA->ODR |= 0x1B3;
            port1=GPIOA->ODR;
            GPIOA->ODR &=0x00;
        }
        else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
        {
            EXTI->FPR1 = 0xFF;

            port1 &=0x1FF;
            port2 &=0x1FF;
            port3 &=0x1FF;
            port4 &=0x1FF;
            port4=port3;
            port3=port2;
            port2=port1;
            port2|=0x1A00;
            port3|=0x1600;
            port4|=0xE00;
            GPIOA->ODR &=0x00;
            GPIOA->ODR |=0x1C00;
            GPIOA->ODR |= 0x1B3;
            port1=GPIOA->ODR;
            GPIOA->ODR &=0x00;
        }
    }

}

else if(counter==1)
{
    counter++;
    if(port1!=10 && port2==10 && port3==10 && port4==10)

```

```

        {
            EXTI->FPR1 = 0xFF;

            port1 &= 0x1FF;
            port2 = port1;
            port2 |= 0x1A00;
            GPIOA->ODR &= 0x00;
            GPIOA->ODR |= 0x1C00;
            GPIOA->ODR |= 0x1B3;
            port1 = GPIOA->ODR;
            GPIOA->ODR &= 0x00;
        }
    else if(port1 != 10 && port2 != 10 && port3 != 10 && port4 != 10)
    {
        EXTI->FPR1 = 0xFF;
        port1 &= 0x1FF;
        port2 &= 0x1FF;
        port3 &= 0x1FF;
        port4 &= 0x1FF;
        port4 = port3;
        port3 = port2;
        port2 = port1;
        port2 |= 0x1A00;
        port3 |= 0x1600;
        port4 |= 0xE00;
        GPIOA->ODR &= 0x00;
        GPIOA->ODR |= 0x1C00;
        GPIOA->ODR |= 0x1B3;
        port1 = GPIOA->ODR;
        GPIOA->ODR &= 0x00;
    }
}
else if(counter == 2)
{
    counter++;
    if(port1 != 10 && port2 != 10 && port3 == 10 && port4 == 10)
    {
        EXTI->FPR1 = 0xFF;

        port1 &= 0x1FF;
        port2 &= 0x1FF;
        port3 = port2;
        port2 = port1;
        port2 |= 0x1A00;
        port3 |= 0x1600;
        GPIOA->ODR &= 0x00;
        GPIOA->ODR |= 0x1C00;
        GPIOA->ODR |= 0x1B3;
        port1 = GPIOA->ODR;
        GPIOA->ODR &= 0x00;
    }
}
else if(port1 != 10 && port2 != 10 && port3 != 10 && port4 != 10)
{
    EXTI->FPR1 = 0xFF;
    port1 &= 0x1FF;
    port2 &= 0x1FF;
    port3 &= 0x1FF;

```



```

        port4 &=0x1FF;
        port4=port3;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x1B3;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }

else if(counter==3)
{
    counter=0;
    if(port1!=10 && port2!=10 && port3!=10 && port4==10)
    {
        EXTI->FPR1 = 0xFF;

        port1 &=0x1FF;
        port2 &=0x1FF;
        port3 &=0x1FF;
        port4=port3;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x1B3;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
    else if(port1!=10 && port2!=10 && port3!=10 && port4!=10)
    {
        EXTI->FPR1 = 0xFF;
        port1 &=0x1FF;
        port2 &=0x1FF;
        port3 &=0x1FF;
        port4 &=0x1FF;
        port4=port3;
        port3=port2;
        port2=port1;
        port2|=0x1A00;
        port3|=0x1600;
        port4|=0xE00;
        GPIOA->ODR &=0x00;
        GPIOA->ODR |=0x1C00;
        GPIOA->ODR |= 0x1B3;
        port1=GPIOA->ODR;
        GPIOA->ODR &=0x00;
    }
}
}

```

```
void delay(volatile uint32_t s)
{
    for(; s>0; s--);
}
```