



Practical Assignment № 4

Digital and Microcontroller Devices



variant number : 6

Student Name : Xu Miao, Zhou Haojie

HDU Number : 19322103, 19322233

ITMO number : 293687, 293806

Practical Task 4

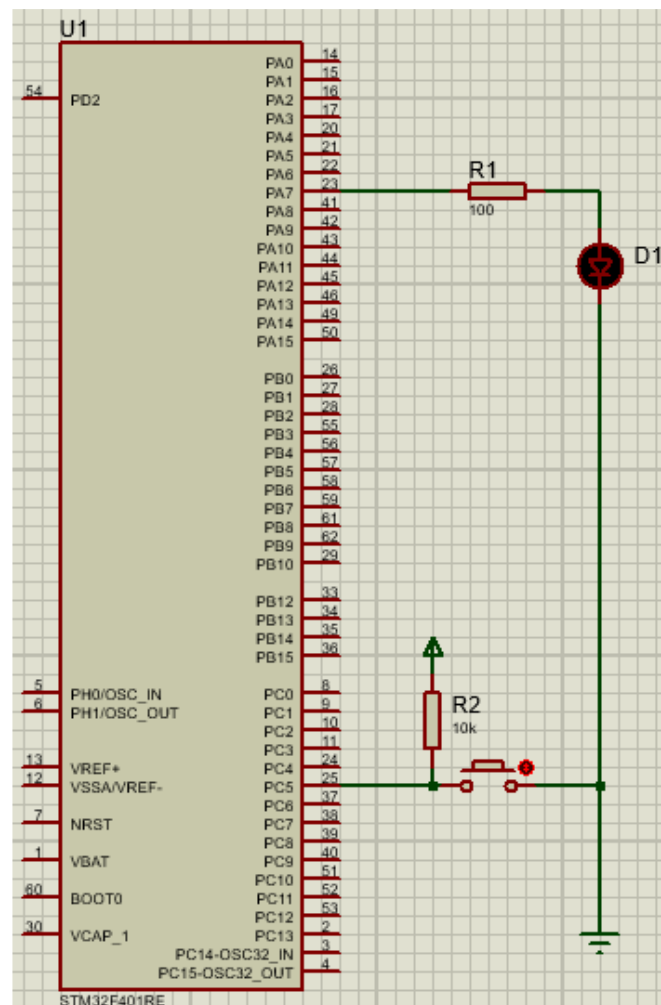
1) Purpose of work:

Acquaintance with the development environment.

2) Follow all the steps in Part I:

Part I. Create a project by using a code generator.

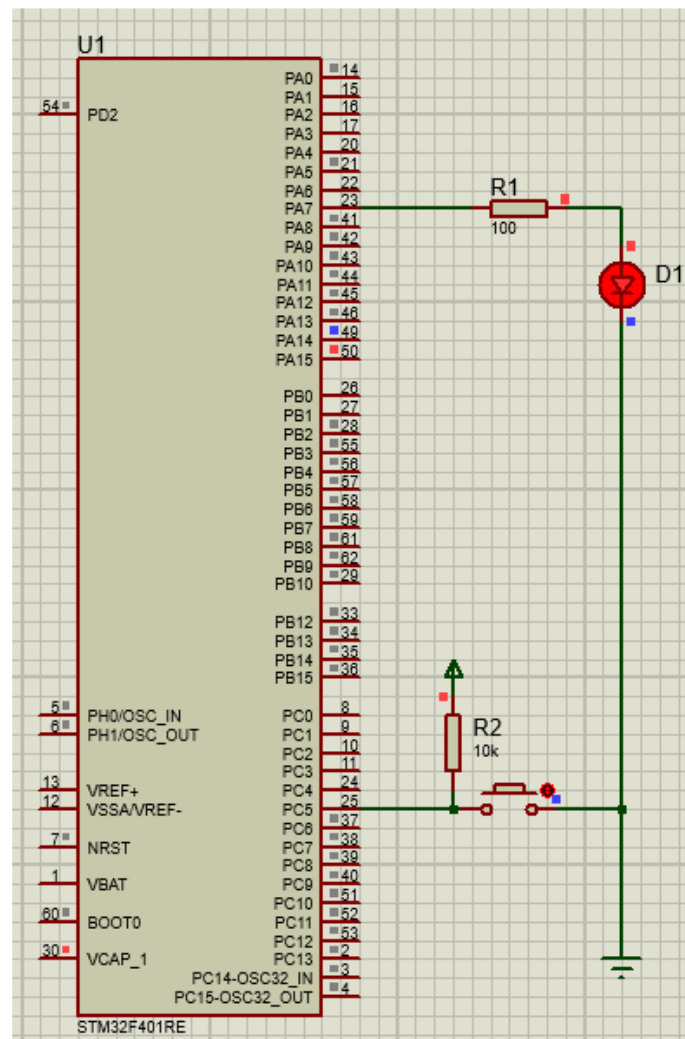
- Proteus scheme



- Program code

```
93 while (1)
94 {
95     HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_7);
96     HAL_Delay(1000);
97     /* USER CODE END WHILE */
```

- Simulation Result

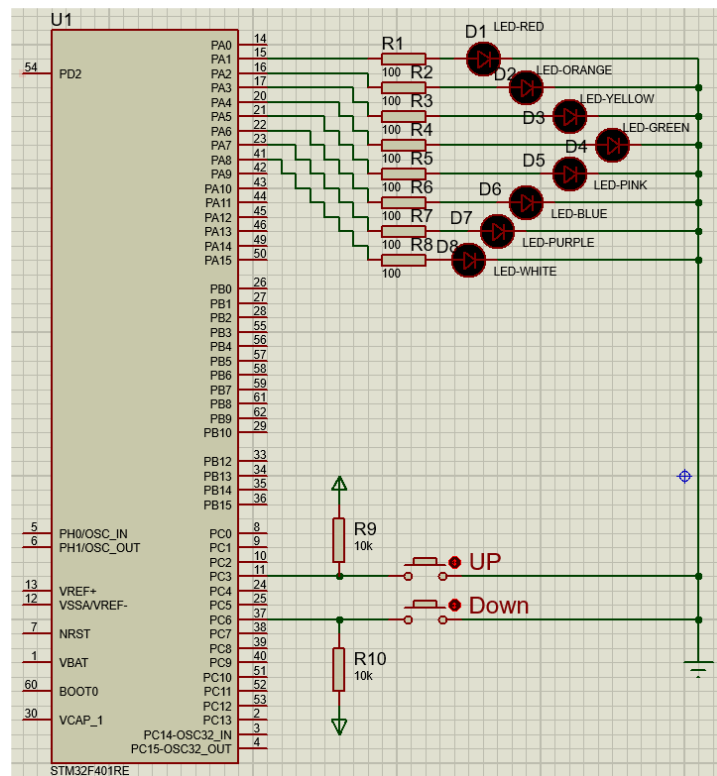


LED lights flash intermittently

3) Modify the scheme, the scheme must include the following elements:

- a) microcontroller;
- b) buttons, name one of them is "up", another is "down";
- c) 8 LEDs, which form an 8-bit number in binary representation.

● Proteus scheme



4) The program must perform the following functions:

- a) In the initial state, all LEDs are off.
- b) When you press «up" button, the value of the 8-bit number increases, the corresponding LEDs light up.
- c) When you press «down" button, the value of the 8-bit number decreases, the corresponding LEDs light up.

● Program code

➤ Variable initialization

```
87  /* USER CODE BEGIN 2 */
88  uint32_t button_up = 0, button_down = 0, LED_value, i, n;
89  LED_value = 0;
90  /* USER CODE END 2 */
```

➤ Operation LED_value change

```

93  /* USER CODE BEGIN WHILE */
94
95  while (1)
96  {
97      button_up = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_3);
98      button_down = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_6);
99      if(!button_up)
100      {
101          if(LED_value < 255)
102          {
103              LED_value = LED_value + 1;
104          }
105          else
106          {
107              LED_value = 255;
108          }
109      }
110      if(!button_down)
111      {
112          if(LED_value > 0)
113          {
114              LED_value = LED_value - 1;
115          }
116          else
117          {
118              LED_value = 0;
119          }
120      }

```

➤ Light up the LEDs according to the LED_value value

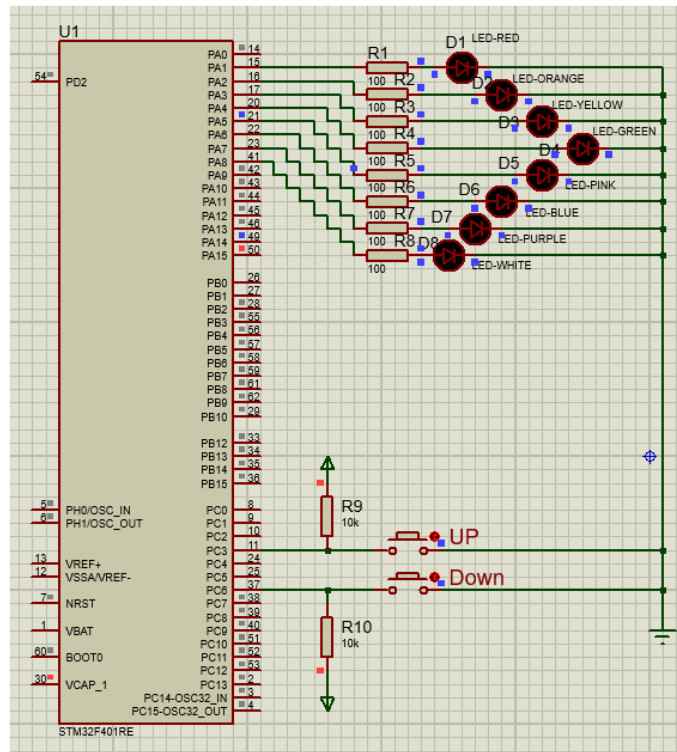
```

121  uint32_t LED[8] = {0};
122  if(LED_value == 0){}
123  else
124  {
125      n = LED_value;
126      i = 0;
127      while(n>0)
128      {
129          LED[i] = n%2;
130          i = i+1;
131          n = n/2;
132      }
133  }
134  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, LED[0]);
135  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, LED[1]);
136  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, LED[2]);
137  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, LED[3]);
138  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, LED[4]);
139  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, LED[5]);
140  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, LED[6]);
141  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, LED[7]);
142  HAL_Delay(100);
143  /* USER CODE END WHILE */

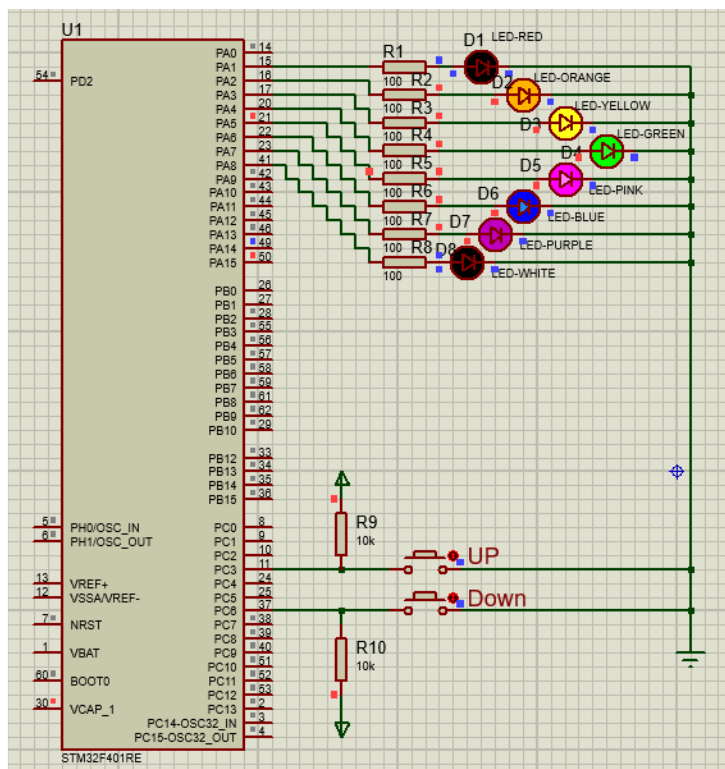
```

● Simulation Result

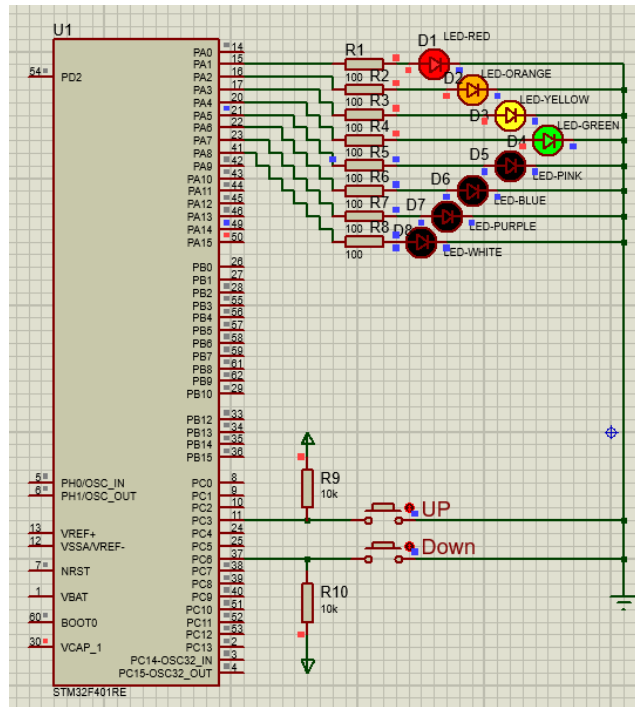
1. In the initial state, all LEDs are off.



- When you press «up" button, the value of the 8-bit number increases, the corresponding LEDs light up.

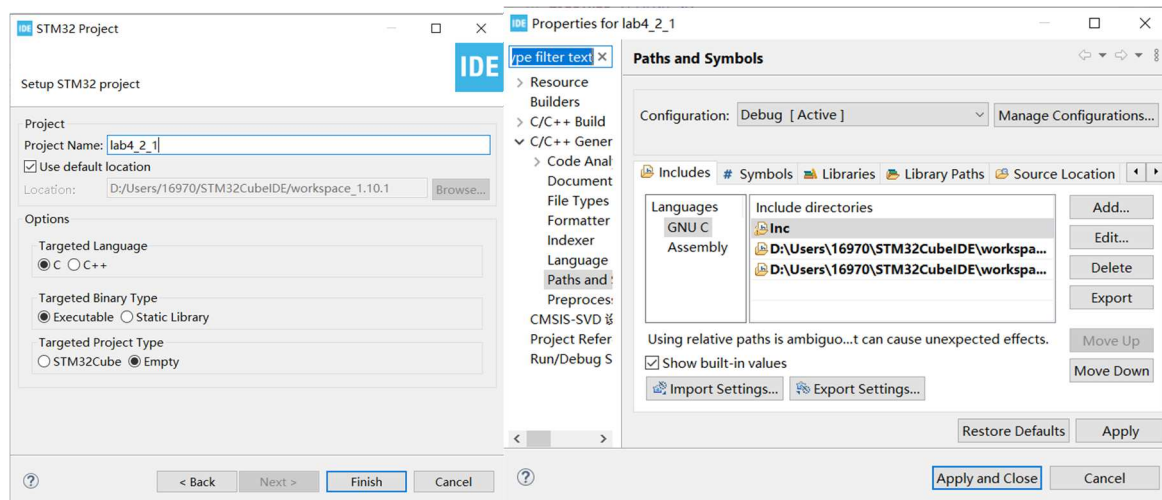


- When you press «down" button, the value of the 8-bit number decreases, the corresponding LEDs light up.



5) Follow all the steps in Part II.

- add libraries from project



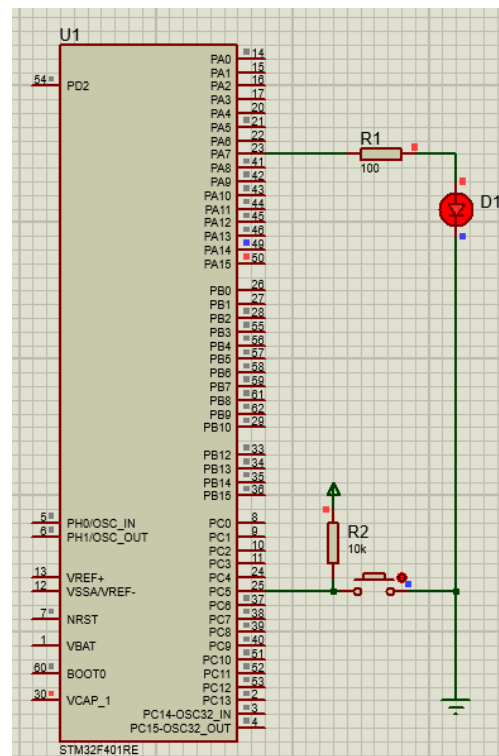
- Writing the program:

```

19 #include <stdint.h>
20 #include "stm32f401xe.h"
21
22 #if !defined(__SOFT_FP__) && defined(__ARM_FP)
23 #warning "FPU is not initialized, but the project is compiling for an FPU. Please initialize the FPU before use."
24 #endif
25
26 int main(void)
27 {
28     RCC->AHB1ENR |= RCC_AHB1ENR_GPIOAEN;
29     GPIOA->MODER |= 1 << 7*2;
30     /* Loop forever */
31     for(;;)
32     {
33         GPIOA->ODR ^= GPIO_ODR_OD7;
34         for(uint32_t i=0; i<1600000; i++);
35     }
36 }
37

```

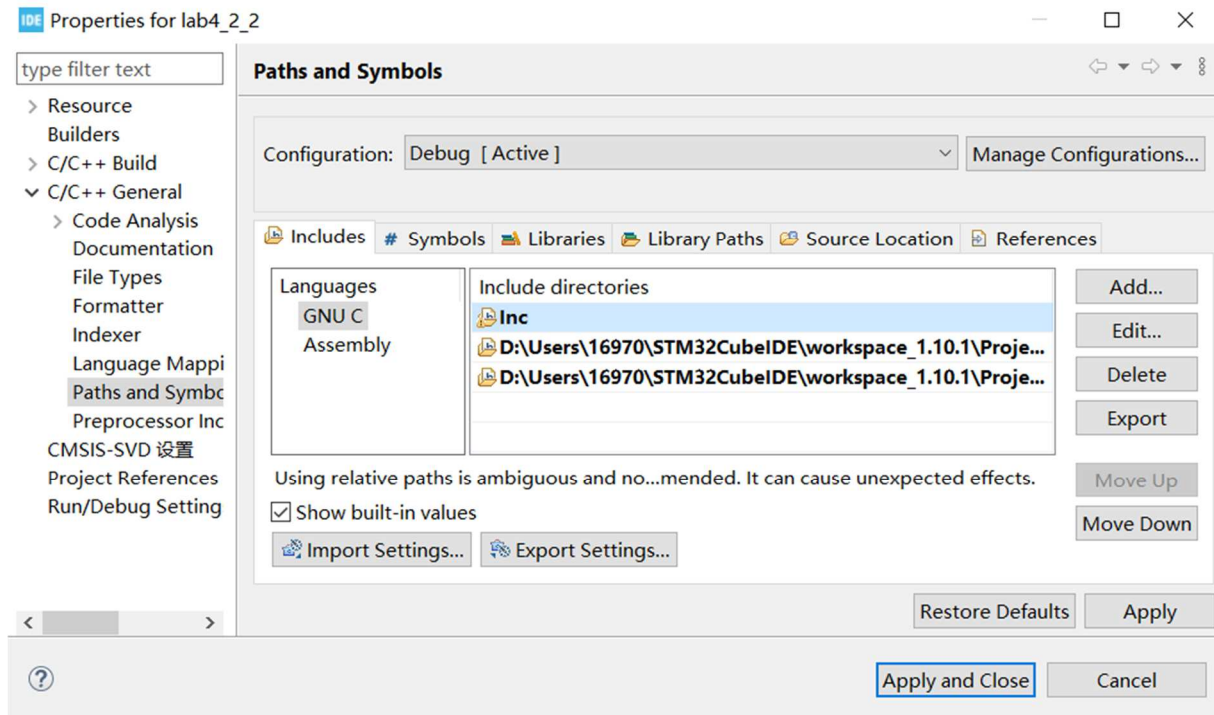
- Simulation Result



LED lights flash intermittently

6) complete Tasks 2 and 3 for Part II

- add libraries from project



- Writing the program:

➤ GPIOA,GPIOC enable and set mode

```
19 #include <stdint.h>
20 #include "stm32f401xe.h"
21
22 #if !defined(__SOFT_FP__) && defined(__ARM_FP)
23     #warning "FPU is not initialized, but the project is compiling for an FPU. Please ..."
24 #endif
25
26 int main(void)
27 {
28     RCC->AHB1ENR |= RCC_AHB1ENR_GPIOAEN;
29     RCC->AHB1ENR |= RCC_AHB1ENR_GPIOCEN;
30     GPIOA->MODER |= 1 << 1*2;
31     GPIOA->MODER |= 1 << 2*2;
32     GPIOA->MODER |= 1 << 3*2;
33     GPIOA->MODER |= 1 << 4*2;
34     GPIOA->MODER |= 1 << 5*2;
35     GPIOA->MODER |= 1 << 6*2;
36     GPIOA->MODER |= 1 << 7*2;
37     GPIOA->MODER |= 1 << 8*2;
38     /* Loop forever */
39     uint32_t button_up = 0, button_down = 0, LED_value, i, n;
40     LED_value = 0;
```

➤ Operation LED_value change

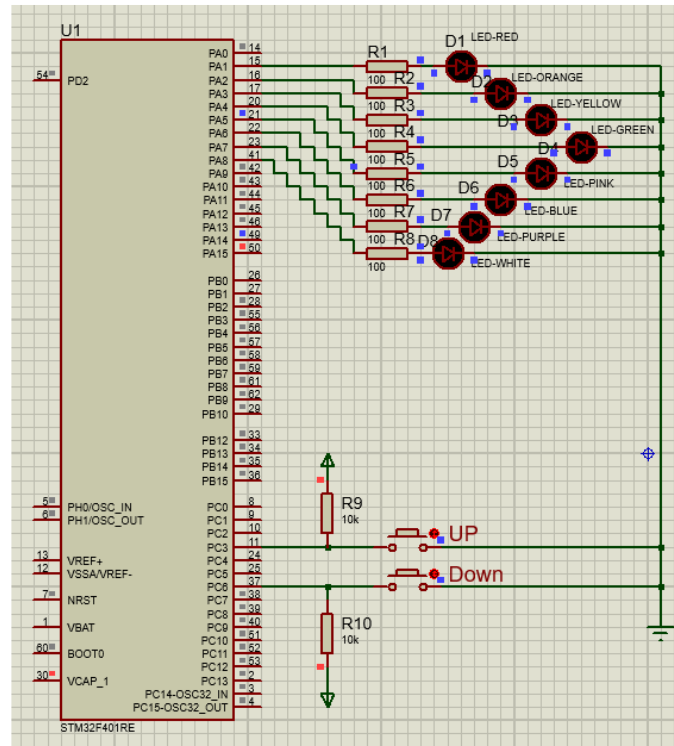
```
41     for(;;)
42     {
43         button_up = GPIOC->IDR & 1 << 3;
44         button_down = GPIOC->IDR & 1 << 6;
45         if(!button_up)
46         {
47             if(LED_value < 255)
48             {
49                 LED_value = LED_value + 1;
50             }
51             else
52             {
53                 LED_value = 255;
54             }
55         }
56         if(!button_down)
57         {
58             if(LED_value > 0)
59             {
60                 LED_value = LED_value - 1;
61             }
62             else
63             {
64                 LED_value = 0;
65             }
66         }
67     }
```

➤ Light up the LEDs according to the LED_value value

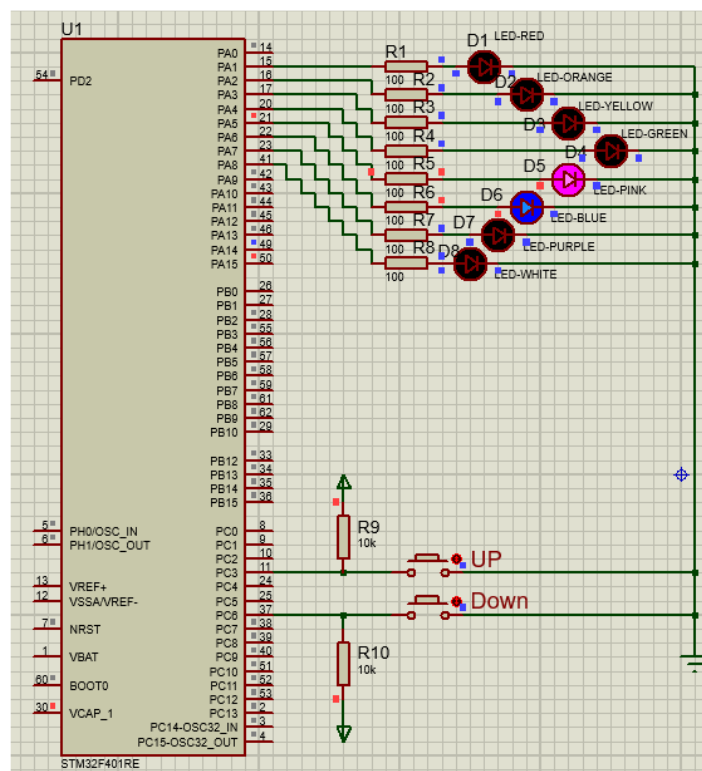
```
67     uint32_t LED[8] = {0};
68     if(LED_value == 0){}
69     else
70     {
71         n = LED_value;
72         i = 0;
73         while(n>0)
74         {
75             LED[i] = n%2;
76             i = i+1;
77             n = n/2;
78         }
79     }
80     for (i=0;i<=7;i++)
81     {
82         if (LED[i]==1)
83         {
84             GPIOA->ODR |= (1 << (i+1));
85         }
86         else
87         {
88             GPIOA->ODR &= ~(1 << (i+1));
89         }
90     }
91     for(uint32_t i=0;i<1600000;i++);
92 }
93 }
```

● Simulation Result

1. In the initial state, all LEDs are off.

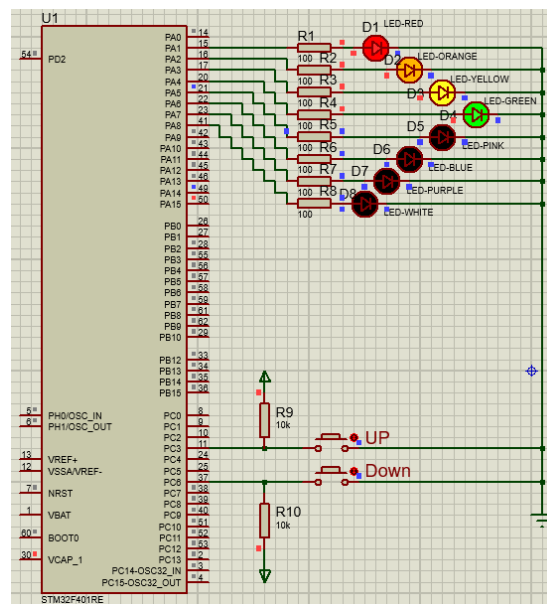


2. When you press «up" button, the value of the 8-bit number increases, the corresponding LEDs light up.



3. When you press «down" button, the value of the 8-bit number decreases, the

corresponding LEDs light up.



7) Conclusions:

conclusion:

- In this experiment we used two methods to accomplish the LED light flashing and key LED flow lights representing binary numbers:
 - Use the HAL library
 - Work with registers
- After the experiments, we were initially familiar with the development environment of STM32cubeIDE and proteus