ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

THÍ NGHIỆM

VI XỬ LÍ - VI ĐIỀU KHIỂN (CO 3010)

BÁO CÁO LAB 1

Giảng viên: PHAN VĂN SỸ

Họ và tên	**Mã số sinh viên**
Nguyễn Văn Sang	2212912

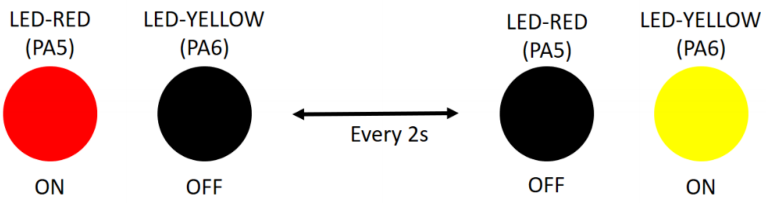
TP.Hồ Chí Minh, 24/09/2025

Mục lục

1 Bài 1

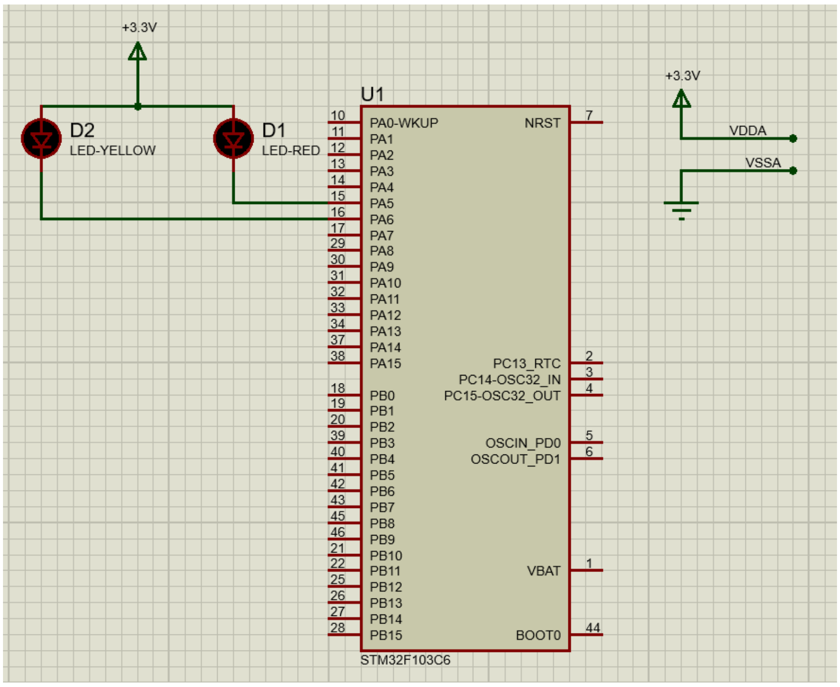
From the simulation on Proteus, one more LED is connected to pin PA6 of the STM32 (negative pin of the LED is connected to PA6). The component suggested in this exercise is LED-YELLOW, which can be found from the device list.

In this exercise, the status of two LEDs are switched every 2 seconds, as demonstrated in the figure bellow.



Hình 1: State transitions for 2 LEDs

Report 1:



Hình 2: Schematic of exercise 1

Report 2:

```
while (1) {
    // Led_Do turn on, Led_Vang turn off
    HAL_GPIO_WritePin(Led_Do_GPIO_Port, Led_Do_Pin, RESET);
    HAL_GPIO_WritePin(Led_Vang_GPIO_Port, Led_Vang_Pin, SET);
      HAL_Delay(2000);

    // Led_Vang turn on, Led_Do turn off
    HAL_GPIO_WritePin(Led_Do_GPIO_Port, Led_Do_Pin, SET);
    HAL_GPIO_WritePin(Led_Vang_GPIO_Port, Led_Vang_Pin, RESET);
      HAL_Delay(2000);
}
```

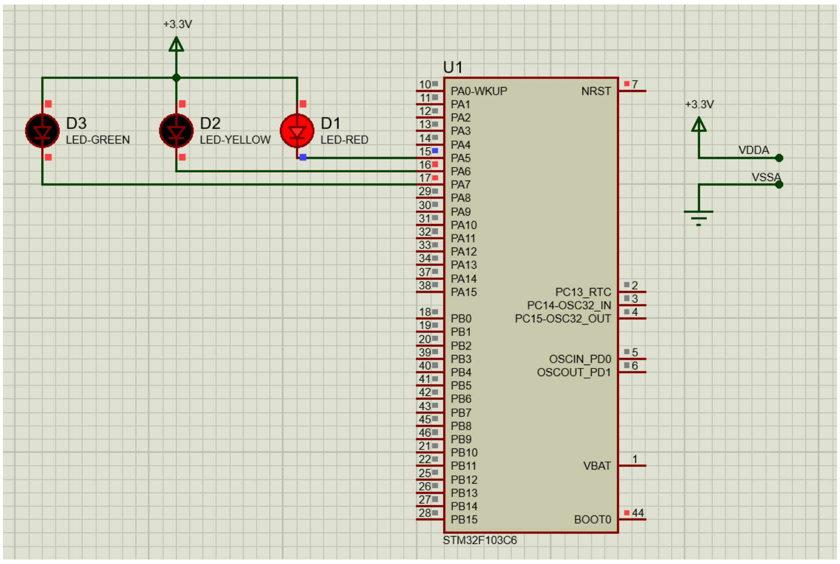
Link Github bài 1: https://github.com/SangNguyen-232/VXLVDK-LAB1/tree/main/B1

2 Bài 2

Extend the first exercise to simulate the behavior of a traffic light. A third LED, named LED-GREEN is added to the system, which is connected to PA7. A cycle in this traffic light is 5 seconds for the RED, 2 seconds for the YELLOW and 3 seconds for the GREEN. The LED-GREEN is also controlled by its negative pin.

Similarly, the report in this exercise includes the schematic of your circuit and a your source code in the while loop.

Report 1:



Hình 3: Schematic of exercise 2

Report 2:

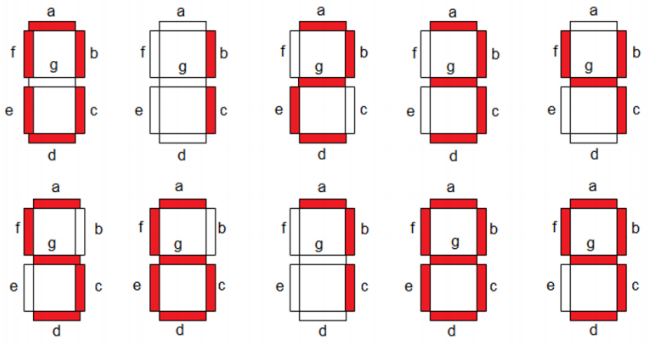
```
while (1) {
    // Led_Xanh turn on, Led_Vang and Led_Do turn off
    HAL_GPIO_WritePin(Led_Do_GPIO_Port, Led_Do_Pin, SET);
    HAL_GPIO_WritePin(Led_Vang_GPIO_Port, Led_Vang_Pin, SET);
    HAL_GPIO_WritePin(Led_Xanh_GPIO_Port, Led_Xanh_Pin, RESET);
      HAL_Delay(3000);

    // Led_Vang turn on, Led_Xanh and Led_Do turn off
    HAL_GPIO_WritePin(Led_Do_GPIO_Port, Led_Do_Pin, SET);
    HAL_GPIO_WritePin(Led_Vang_GPIO_Port, Led_Vang_Pin, RESET);
    HAL_GPIO_WritePin(Led_Xanh_GPIO_Port, Led_Xanh_Pin, SET);
      HAL_Delay(2000);

    // Led_Do turn on, Led_Xanh and Led_Vang turn off
    HAL_GPIO_WritePin(Led_Do_GPIO_Port, Led_Do_Pin, RESET);
    HAL_GPIO_WritePin(Led_Vang_GPIO_Port, Led_Vang_Pin, SET);
    HAL_GPIO_WritePin(Led_Xanh_GPIO_Port, Led_Xanh_Pin, SET);
      HAL_Delay(5000);
}
```

Link Github bài 2: https://github.com/SangNguyen-232/VXLVDK-LAB1/tree/main/B2

3 Bài 4

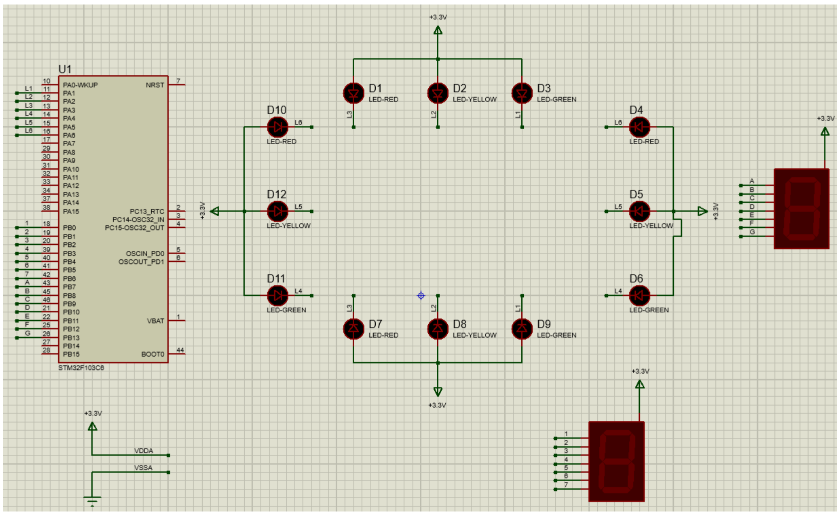
Add only one 7 led segment to the schematic in Exercise 3. This component can be found in Proteus by the keyword 7SEG-COM-ANODE. For this device, the common pin should be connected to the power supply and other pins are supposed to connected to PB0 to PB6. Therefore, to turn-on a segment in this 7SEG, the STM32 pin should be in logic 0 (0V).

Implement a function named display7SEG(int num). The input for this function is from 0 to 9 and the outputs are listed as following:



Hình 4: Display a number on 7 segment LED

Report 1:



Hình 5: Schematic of exercise 4 and 5

Report 2:

```c
int main(void) {
    // Turn off traffic LEDs and clear both 7-seg
    setNS(0,0,0);
    setEW(0,0,0);
    clear7SEG_NS();
    clear7SEG_EW();

    while (1) {
        // NS RED, EW GREEN and YELLOW
        setNS(0,0,1);   // NS red on
        setEW(1,0,0);   // EW green on
        for (int s = 0; s < EW_GREEN_TIME; s++) {
          display7SEG_NS(NS_RED_TIME - s - 1); // NS red counts 8->4
          display7SEG_EW(EW_GREEN_TIME - s - 1); // EW green counts 5->1
          HAL_Delay(1000);
        }
        setEW(0,1,0);   // EW yellow on
        for (int s = 0; s < EW_YELLOW_TIME + 1; s++) {
          display7SEG_NS(NS_RED_TIME - EW_GREEN_TIME - s - 1); // NS red counts 3->0
          display7SEG_EW(EW_YELLOW_TIME - s); // EW yellow counts 2->0
          HAL_Delay(1000);
        }

        // EW RED, North-South GREEN and YELLOW
        setNS(1,0,0);   // NS green on
        setEW(0,0,1);   // EW red on
        for (int s = 0; s < EW_GREEN_TIME; s++) {
          display7SEG_EW(NS_RED_TIME - s - 1); // EW red counts 8->4
          display7SEG_NS(EW_GREEN_TIME - s - 1); // NS green counts 5->1
          HAL_Delay(1000);
        }
        setNS(0,1,0);   // NS yellow on
        for (int s = 0; s < EW_YELLOW_TIME + 1; s++) {
          display7SEG_EW(NS_RED_TIME - EW_GREEN_TIME - s - 1); // EW red counts 3->0
          display7SEG_NS(EW_YELLOW_TIME - s); // NS yellow counts 2->0
```

```
            HAL_Delay(1000);
        }
    }
}
```

4 Bài 5

Integrate the 7SEG-LED to the 4 way traffic light. In this case, the 7SEG-LED is used to display countdown value.

In this exercise, only source code is required to present. The function display7SEG in previous exercise can be reused.

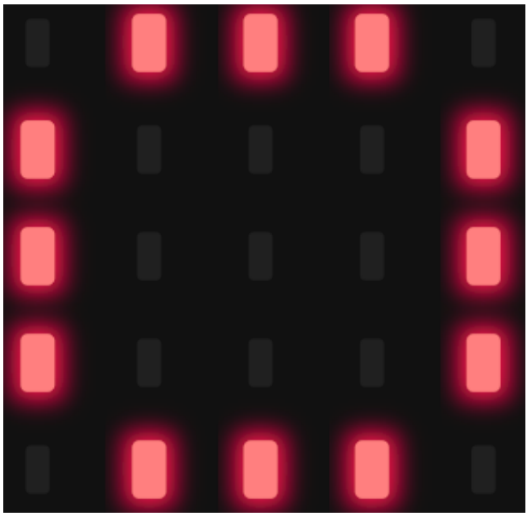
```
// Timing constants
#define NS_RED_TIME   9  // NS red counts 8->0
#define EW_GREEN_TIME 5 // EW green counts 5->0
#define EW_YELLOW_TIME 3 // EW yellow counts 2->0

// GPIO init: configure PA and PB as outputs
static void MX_GPIO_Init(void) {
  GPIO_InitTypeDef GPIO_InitStruct = {0};

  __HAL_RCC_GPIOA_CLK_ENABLE();
  __HAL_RCC_GPIOB_CLK_ENABLE();

  // Traffic LEDs (PA) - set pins OFF initially
  HAL_GPIO_WritePin(L1_GPIO_Port, L1_Pin, GPIO_PIN_SET);
  HAL_GPIO_WritePin(L2_GPIO_Port, L2_Pin, GPIO_PIN_SET);
  HAL_GPIO_WritePin(L3_GPIO_Port, L3_Pin, GPIO_PIN_SET);
  HAL_GPIO_WritePin(L4_GPIO_Port, L4_Pin, GPIO_PIN_SET);
  HAL_GPIO_WritePin(L5_GPIO_Port, L5_Pin, GPIO_PIN_SET);
  HAL_GPIO_WritePin(L6_GPIO_Port, L6_Pin, GPIO_PIN_SET);

  GPIO_InitStruct.Pin = L1_Pin|L2_Pin|L3_Pin|L4_Pin|L5_Pin|L6_Pin;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  HAL_GPIO_Init(L1_GPIO_Port, &GPIO_InitStruct); // all Lx on same port (GPIOA
      )

  // 7-seg NS and EW on PB - set OFF initially
  HAL_GPIO_WritePin(a_GPIO_Port, a_Pin, GPIO_PIN_SET);
  HAL_GPIO_WritePin(b_GPIO_Port, b_Pin, GPIO_PIN_SET);
  HAL_GPIO_WritePin(c_GPIO_Port, c_Pin, GPIO_PIN_SET);
  HAL_GPIO_WritePin(d_GPIO_Port, d_Pin, GPIO_PIN_SET);
  HAL_GPIO_WritePin(e_GPIO_Port, e_Pin, GPIO_PIN_SET);
  HAL_GPIO_WritePin(f_GPIO_Port, f_Pin, GPIO_PIN_SET);
  HAL_GPIO_WritePin(g_GPIO_Port, g_Pin, GPIO_PIN_SET);

  HAL_GPIO_WritePin(A_GPIO_Port, A_Pin, GPIO_PIN_SET);
  HAL_GPIO_WritePin(B_GPIO_Port, B_Pin, GPIO_PIN_SET);
  HAL_GPIO_WritePin(C_GPIO_Port, C_Pin, GPIO_PIN_SET);
  HAL_GPIO_WritePin(D_GPIO_Port, D_Pin, GPIO_PIN_SET);
  HAL_GPIO_WritePin(E_GPIO_Port, E_Pin, GPIO_PIN_SET);
  HAL_GPIO_WritePin(F_GPIO_Port, F_Pin, GPIO_PIN_SET);
  HAL_GPIO_WritePin(G_GPIO_Port, G_Pin, GPIO_PIN_SET);

  GPIO_InitStruct.Pin = a_Pin|b_Pin|c_Pin|d_Pin|e_Pin|f_Pin|g_Pin
                      | A_Pin|B_Pin|C_Pin|D_Pin|E_Pin|F_Pin|G_Pin;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
```

```c
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  HAL_GPIO_Init(a_GPIO_Port, &GPIO_InitStruct); // all segment pins on GPIOB
}

// Light status control function
void setLED(GPIO_TypeDef* port, uint16_t pin, uint8_t on) {
  HAL_GPIO_WritePin(port, pin, on ? GPIO_PIN_RESET : GPIO_PIN_SET);
}

// Set NS lights
void setNS(uint8_t g, uint8_t y, uint8_t r) {
  setLED(L1_GPIO_Port, L1_Pin, g);
  setLED(L2_GPIO_Port, L2_Pin, y);
  setLED(L3_GPIO_Port, L3_Pin, r);
}

// Set EW lights
void setEW(uint8_t g, uint8_t y, uint8_t r) {
  setLED(L4_GPIO_Port, L4_Pin, g);
  setLED(L5_GPIO_Port, L5_Pin, y);
  setLED(L6_GPIO_Port, L6_Pin, r);
}

int main(void) {
    // Turn off traffic LEDs and clear both 7-seg
    setNS(0,0,0);
    setEW(0,0,0);
    clear7SEG_NS();
    clear7SEG_EW();

    while (1) {
        // NS RED, EW GREEN and YELLOW
        setNS(0,0,1);   // NS red on
        setEW(1,0,0);   // EW green on
        for (int s = 0; s < EW_GREEN_TIME; s++) {
          display7SEG_NS(NS_RED_TIME - s - 1); // NS red counts 8->4
          display7SEG_EW(EW_GREEN_TIME - s - 1); // EW green counts 5->1
          HAL_Delay(1000);
        }
        setEW(0,1,0);   // EW yellow on
        for (int s = 0; s < EW_YELLOW_TIME + 1; s++) {
          display7SEG_NS(NS_RED_TIME - EW_GREEN_TIME - s - 1); // NS red counts 3->0
          display7SEG_EW(EW_YELLOW_TIME - s); // EW yellow counts 2->0
          HAL_Delay(1000);
        }

        // EW RED, North-South GREEN and YELLOW
        setNS(1,0,0);   // NS green on
        setEW(0,0,1);   // EW red on
        for (int s = 0; s < EW_GREEN_TIME; s++) {
          display7SEG_EW(NS_RED_TIME - s - 1); // EW red counts 8->4
          display7SEG_NS(EW_GREEN_TIME - s - 1); // NS green counts 5->1
          HAL_Delay(1000);
        }
        setNS(0,1,0);   // NS yellow on
        for (int s = 0; s < EW_YELLOW_TIME + 1; s++) {
          display7SEG_EW(NS_RED_TIME - EW_GREEN_TIME - s - 1); // EW red counts 3->0
          display7SEG_NS(EW_YELLOW_TIME - s); // NS yellow counts 2->0
          HAL_Delay(1000);
        }
    }
}

// Display digit on NS LED-7-seg
void display7SEG_NS(int num) {
  clear7SEG_NS();
  const uint16_t digits[10] = {
    a_Pin|b_Pin|c_Pin|d_Pin|e_Pin|f_Pin,
    b_Pin|c_Pin,
    a_Pin|b_Pin|d_Pin|e_Pin|g_Pin,
    a_Pin|b_Pin|c_Pin|d_Pin|g_Pin,
    b_Pin|c_Pin|f_Pin|g_Pin,
    a_Pin|c_Pin|d_Pin|f_Pin|g_Pin,
    a_Pin|c_Pin|d_Pin|e_Pin|f_Pin|g_Pin,
    a_Pin|b_Pin|c_Pin,
    a_Pin|b_Pin|c_Pin|d_Pin|e_Pin|f_Pin|g_Pin,
    a_Pin|b_Pin|c_Pin|d_Pin|f_Pin|g_Pin
  };
  if (num >= 0 && num <= 9) {
    HAL_GPIO_WritePin(a_GPIO_Port, digits[num], GPIO_PIN_RESET);
  }
}

// Display digit on EW LED-7-seg
void display7SEG_EW(int num) {
  clear7SEG_EW();
  const uint16_t digits[10] = {
    A_Pin|B_Pin|C_Pin|D_Pin|E_Pin|F_Pin,
    B_Pin|C_Pin,
    A_Pin|B_Pin|D_Pin|E_Pin|G_Pin,
    A_Pin|B_Pin|C_Pin|D_Pin|G_Pin,
    B_Pin|C_Pin|F_Pin|G_Pin,
    A_Pin|C_Pin|D_Pin|F_Pin|G_Pin,
    A_Pin|C_Pin|D_Pin|E_Pin|F_Pin|G_Pin,
    A_Pin|B_Pin|C_Pin,
    A_Pin|B_Pin|C_Pin|D_Pin|E_Pin|F_Pin|G_Pin,
    A_Pin|B_Pin|C_Pin|D_Pin|F_Pin|G_Pin
  };
  if (num >= 0 && num <= 9) {
    HAL_GPIO_WritePin(A_GPIO_Port, digits[num], GPIO_PIN_RESET);
  }
}
```

Link Github bài 4 và 5: https://github.com/SangNguyen-232/VXLVDK-LAB1/tree/main/B4%2B5

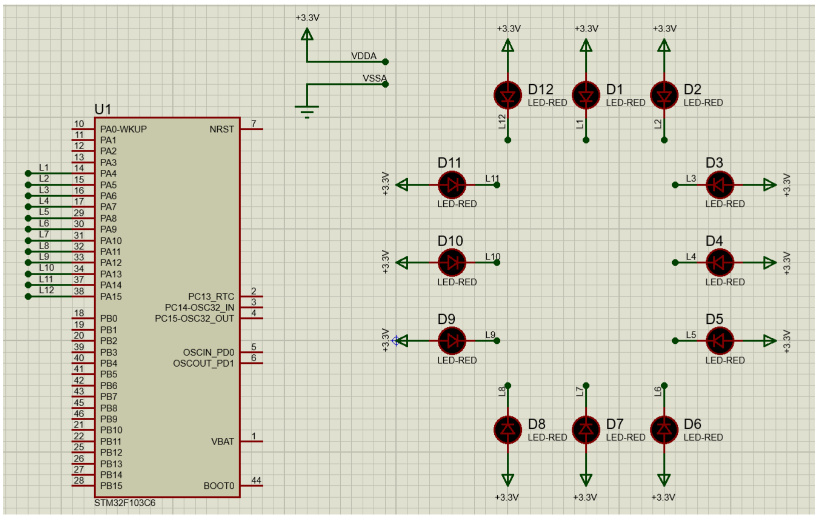
5 Bài 6

In this exercise, a new Proteus schematic is designed to simulate an analog clock, with 12 different number. The connections for 12 LEDs are supposed from PA4 to PA15 of the STM32. The arrangement of 12 LEDs is depicted as follows.



Hình 6: 12 LEDs for an analog clock

Report 1:



Hình 7: Schematic of exercise 6 - 10

Report 2:

```
#define LED_COUNT 12
#define LED_SINK 1

static const uint16_t LED_PINS[LED_COUNT] = {
  Chan_1_Pin, Chan_2_Pin, Chan_3_Pin, Chan_4_Pin,
  Chan_5_Pin, Chan_6_Pin, Chan_7_Pin, Chan_8_Pin,
  Chan_9_Pin, Chan_10_Pin, Chan_11_Pin, Chan_12_Pin
};
```

```c
static GPIO_TypeDef* const LED_PORT = Chan_1_GPIO_Port;
static const GPIO_PinState LED_ON  = (LED_SINK ? GPIO_PIN_RESET : GPIO_PIN_SET);
static const GPIO_PinState LED_OFF = (LED_SINK ? GPIO_PIN_SET : GPIO_PIN_RESET);

static void AllLEDsOff(void) {
  for (int i = 0; i < LED_COUNT; ++i) {
    HAL_GPIO_WritePin(LED_PORT, LED_PINS[i], LED_OFF);
  }
}

static void AllLEDsOn(void) {
  for (int i = 0; i < LED_COUNT; ++i) {
    HAL_GPIO_WritePin(LED_PORT, LED_PINS[i], LED_ON);
  }
}

#define ALL_ON_MS     1000U
#define ALL_OFF_MS    1000U
#define STEP_MS       1000U

  AllLEDsOn();
  HAL_Delay(ALL_ON_MS);
  AllLEDsOff();
  HAL_Delay(ALL_OFF_MS);

while (1) {
    for (int i = 0; i < LED_COUNT; ++i) {
      AllLEDsOff();
      HAL_GPIO_WritePin(LED_PORT, LED_PINS[i], LED_ON);
      HAL_Delay(STEP_MS);
    }

    HAL_GPIO_WritePin(LED_PORT, LED_PINS[LED_COUNT - 1], LED_OFF);
    HAL_Delay(1000);

    AllLEDsOn();
    HAL_Delay(ALL_ON_MS);

    AllLEDsOff();
    HAL_Delay(ALL_OFF_MS);
}
```

6 Bài 7

Implement a function named clearAllClock() to turn off all 12 LEDs. Present the source code of this function.

I used the AllLEDsOff(void) function to turn off all the LEDs and this is done in conjunction with exercise 6.

```c
// Main implementation of the function to turn off all 12 LEDs at once
static void AllLEDsOff(void) {
  for (int i = 0; i < LED_COUNT; ++i) {
    HAL_GPIO_WritePin(LED_PORT, LED_PINS[i], LED_OFF);
  }
}

```

```
// Initialize or initialize all LEDs to off state before entering main loop,
    ensuring all LEDs are off at the same time.
AllLEDsOn();
HAL_Delay(ALL_ON_MS);
AllLEDsOff();
HAL_Delay(ALL_OFF_MS);

while (1) {
  // Turn off all LEDs at once before turning on a specific LED
    for (int i = 0; i < LED_COUNT; ++i) {
      AllLEDsOff();
      HAL_GPIO_WritePin(LED_PORT, LED_PINS[i], LED_ON);
      HAL_Delay(STEP_MS);
    }

    HAL_GPIO_WritePin(LED_PORT, LED_PINS[LED_COUNT - 1], LED_OFF);
    HAL_Delay(1000);

    AllLEDsOn();
    HAL_Delay(ALL_ON_MS);

    //Turn off all LEDs at the same time, used to end an LED running cycle,
        returning the system to the off state before starting a new cycle
    AllLEDsOff();
    HAL_Delay(ALL_OFF_MS);
}
```

7 Bài 8:

Implement a function named setNumberOnClock(int num). The input for this function is from 0 to 11 and an appropriate LED is turn on. Present the source code of this function.

```
// LED_ON / LED_OFF macros
#define LED_ACTIVE_HIGH 0
#if LED_ACTIVE_HIGH
  #define LED_ON(port,pin)  HAL_GPIO_WritePin((port),(pin),GPIO_PIN_SET)
  #define LED_OFF(port,pin) HAL_GPIO_WritePin((port),(pin),GPIO_PIN_RESET)
#else
  #define LED_ON(port,pin)  HAL_GPIO_WritePin((port),(pin),GPIO_PIN_RESET)
  #define LED_OFF(port,pin) HAL_GPIO_WritePin((port),(pin),GPIO_PIN_SET)
#endif

// LED pin and port mapping array
static GPIO_TypeDef* led_ports[12] = {
  LED1_PORT, LED2_PORT, LED3_PORT, LED4_PORT,
  LED5_PORT, LED6_PORT, LED7_PORT, LED8_PORT,
  LED9_PORT, LED10_PORT, LED11_PORT, LED12_PORT
};
static const uint16_t led_pins[12] = {
  Led_1_Pin,  Led_2_Pin,  Led_3_Pin,  Led_4_Pin,
  Led_5_Pin,  Led_6_Pin,  Led_7_Pin,  Led_8_Pin,
  Led_9_Pin,  Led_10_Pin, Led_11_Pin, Led_12_Pin
};

void setNumberOnClock(int num);

// Cycle 1: Turn ON only the LED corresponding to num
void setNumberOnClock(int num) {
  if (num < 0 || num > 11) return;
  for (int i = 0; i < 12; ++i) {
    if (i == num)
        LED_ON(led_ports[i], led_pins[i]);
    else
        LED_OFF(led_ports[i], led_pins[i]);
  }
}

int main(void) {
  // Cycle 1: Run once -> light a single LED
  setNumberOnClock(3);
  HAL_Delay(1000);
  signalCycleEnd(1, 300, 300);
}
```

8 Bài 9:

Implement a function named clearNumberOnClock(int num). The input for this function is from 0 to 11 and an appropriate LED is turn off.

```
// LED_ON / LED_OFF macros
#define LED_ACTIVE_HIGH 0
#if LED_ACTIVE_HIGH
  #define LED_ON(port,pin)  HAL_GPIO_WritePin((port),(pin),GPIO_PIN_SET)
  #define LED_OFF(port,pin) HAL_GPIO_WritePin((port),(pin),GPIO_PIN_RESET)
#else
  #define LED_ON(port,pin)  HAL_GPIO_WritePin((port),(pin),GPIO_PIN_RESET)
  #define LED_OFF(port,pin) HAL_GPIO_WritePin((port),(pin),GPIO_PIN_SET)
#endif

// LED pin and port mapping array
static GPIO_TypeDef* led_ports[12] = {
  LED1_PORT, LED2_PORT, LED3_PORT, LED4_PORT,
  LED5_PORT, LED6_PORT, LED7_PORT, LED8_PORT,
  LED9_PORT, LED10_PORT, LED11_PORT, LED12_PORT
};
static const uint16_t led_pins[12] = {
  Led_1_Pin,  Led_2_Pin,  Led_3_Pin,  Led_4_Pin,
  Led_5_Pin,  Led_6_Pin,  Led_7_Pin,  Led_8_Pin,
  Led_9_Pin,  Led_10_Pin, Led_11_Pin, Led_12_Pin
};

void clearNumberOnClock(int num);

// Cycle 2: Turn OFF only the LED corresponding to num; turn ON all others.

void clearNumberOnClock(int num) {
  if (num < 0 || num > 11) return;
  for (int i = 0; i < 12; ++i) {
    if (i == num)
        LED_OFF(led_ports[i], led_pins[i]);
    else
        LED_ON(led_ports[i], led_pins[i]);
  }
}

int main(void) {
  // Cycle 2: Run once -> clear a single LED
  clearNumberOnClock(7);
  HAL_Delay(1000);
  signalCycleEnd(1, 300, 300);
}
```

9 Bài 10:

Integrate the whole system and use 12 LEDs to display a clock. At a given time, there are only 3 LEDs are turn on for hour, minute and second information.

```
// Cycle 3 step delay in milliseconds
#define CYCLE3_STEP_MS 50

// RTC-like internal time for Cycle 3
static int rtc_hour   = 0;
static int rtc_minute = 0;
static int rtc_second = 0;

void displayClockRealtime(void);

while (1) {
    // DisplayClockRealtime includes the delay and time increment
    displayClockRealtime();
}

void displayClockRealtime(void) {
  int h = (rtc_hour % 12 + 12) % 12;
  int m = ((rtc_minute / 5) % 12 + 12) % 12;
  int s = ((rtc_second / 5) % 12 + 12) % 12;

  // Turn all off first
  for (int i = 0; i < 12; ++i)
    LED_OFF(led_ports[i], led_pins[i]);

  // Turn on hour, minute, second LEDs
  LED_ON(led_ports[h], led_pins[h]);
  LED_ON(led_ports[m], led_pins[m]);
  LED_ON(led_ports[s], led_pins[s]);

  // Wait one second (real)
  HAL_Delay(CYCLE3_STEP_MS);

  // Increment time
  rtc_second++;
  if (rtc_second >= 60) {
    rtc_second = 0;
    rtc_minute++;
  }
  if (rtc_minute >= 60) {
    rtc_minute = 0;
    rtc_hour++;
  }
  if (rtc_hour >= 24) {
    rtc_hour = 0;
  }
}
```

Link Github bài 6 - 10: `https://github.com/SangNguyen-232/VXLVDK-LAB1/tree/main/B6-10`

10 Link Github toàn bộ LAB 1

https://github.com/SangNguyen-232/VXLVDK-LAB1/tree/main