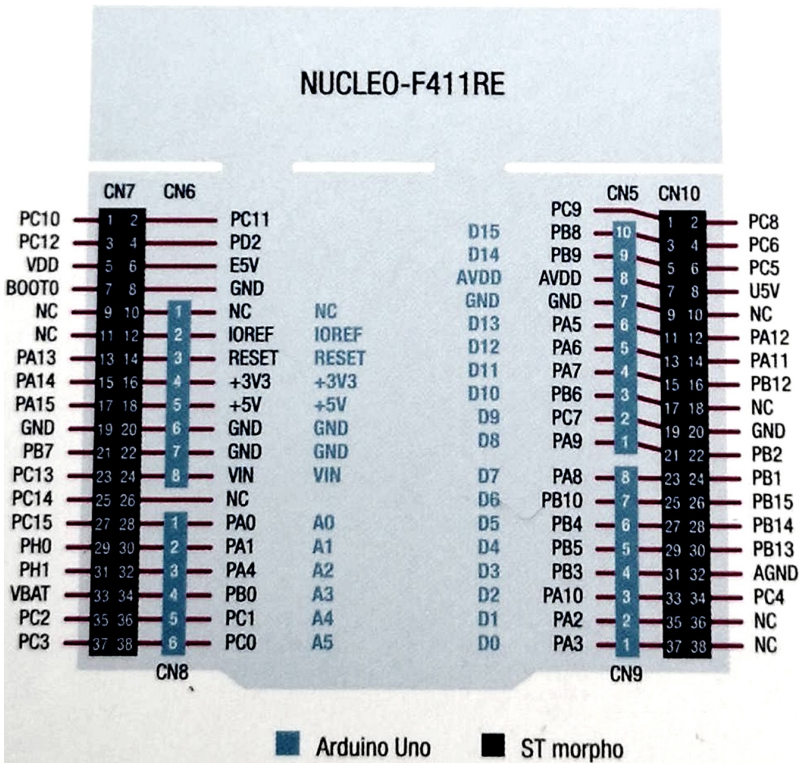
```c
// ---------- Lab2 - 01 & 02 ----------
// blink (on/off) with a period of 0.2 sec. for an on board LED
// blink (on/off) with a period of 0.2 sec. for an external LED

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    HAL_Delay(100);
    HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
    /* USER CODE END WHILE */
    /* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */

// ---------- Lab2 - 03 ----------
// toggle an LED (on/off) with pushing USER push-button. (Debouching is required)

while (1)
{
    // blue button(PC13) is pressed(GPIO_PIN_RESET state)
    if (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET)
    {
        HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
        HAL_Delay(100);
        // debouncing the button
        while (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET)
            ;
    }

    // ---------- Lab2 - 04 ----------
    // communicate UART, serial terminal
    // screen /dev/tty.usbmodem11203 115200, onclr
    // quit: ctrl + A then ctrl + \

    uint8_t ch;
    while (1)
    {
        if (HAL_UART_Receive(&huart2, &ch, 1, 100) == HAL_OK)
        {
            HAL_UART_Transmit(&huart2, &ch, 1, 100);
        }

        // ---------- Lab2 - 05 ----------
        // toggle an LED status (on/off) with commands via serial console.
        // (Type “on” or “off” then press Enter to on or off the LED

        /* Infinite loop */
        /* USER CODE BEGIN WHILE */
        int idx = 0;
        uint8_t ch;
        uint8_t buf[256];
        while (1)
        {
            if (HAL_UART_Receive(&huart2, buf + idx, 1, 1000) == HAL_OK)
            {
                HAL_UART_Transmit(&huart2, buf + idx, 1, 1000);
                if (buf[0] == 'o' && buf[1] == 'n' && buf[2] == '\r' && ch != 1)
                {
                    HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
                    HAL_Delay(100);
                    ch = 1;
                }
                else if (buf[0] == 'o' && buf[1] == 'f' && buf[2] == 'f' && buf[3] == '\r' && ch != 0)
                {
                    HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
                    HAL_Delay(100);
                    ch = 0;
                }
                if (buf[idx] == '\r')
                {
                    idx = 0;
                    buf[2] = '0';
                    buf[3] = '0';
                }
                else
                {
                    idx++;
                }
            }
        }
```

```c
// Skill Test Sec 2 2562
// 1. Make all lights blink every 150 ms (on 50 ms, off 100 ms)
// 2. push button TOGGLE (on 100 ms, off 50 ms)
// 3. hold to all light on and button dont respone anymore
// to push button anymore.

/* USER CODE BEGIN WHILE */

uint8_t state = 1;
uint8_t buttonState = GPIO_PIN_SET;
while (1)
{
    if (state == 1)
    {
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5 | GPIO_PIN_6, GPIO_PIN_SET); // LED on
        HAL_Delay(50);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5 | GPIO_PIN_6, GPIO_PIN_RESET); // LED off
        HAL_Delay(100);

        //            if (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET) {
        //                state++;
        //                // debouncing the button
        //                while (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET);
        //            }
        // Debouncing
        if (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET && buttonState == GPIO_PIN_SET)
        {
            HAL_Delay(50); // Debounce delay
            if (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET)
                state++; // Increment state
        }
        buttonState = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13);
    }
    else if (state == 2)
    {
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5 | GPIO_PIN_6, GPIO_PIN_SET); // LED on
        HAL_Delay(100);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5 | GPIO_PIN_6, GPIO_PIN_RESET); // LED off
        HAL_Delay(50);

        //            if (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET) {
        //                state++;
        //                // debouncing the button
        //                while (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET);
        //            }
        // Debouncing
        if (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET && buttonState == GPIO_PIN_SET)
        {
            HAL_Delay(50); // Debounce delay
            if (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET)
                state++; // Increment state
        }
        buttonState = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13);
    }
    else
    {
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5 | GPIO_PIN_6, GPIO_PIN_SET); // All LEDs on
    }
}

// Debounce function
uint8_t buttonState = GPIO_PIN_SET;
if (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET && buttonState == GPIO_PIN_SET)
{
    HAL_Delay(50); // Debounce delay
    if (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET)
    {
        state++; // Increment state
        // stage controller
        if (state > 2)
            state = 0; // Reset state after the third press
    }
}
buttonState = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13);
```



```c
// ---------- Lab3 - 01 ----------
// blink (on/off) green LED with changeable period by pushing USER push-button
// from periods of 0.2 sec. → 1 sec. → 5 sec. and back to 0.2 sec.
// (using External interrupt/event controller (EXTI) is required)

/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */
uint16_t blinkTime = 200;

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_15);
    HAL_Delay(100);

    if (blinkTime == 200)
        blinkTime = 1000;
    else if (blinkTime == 1000)
        blinkTime = 5000;
    else
        blinkTime = 200;

    __HAL_GPIO_EXTI_CLEAR_IT(GPIO_Pin);
}
/* USER CODE END 0 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
    HAL_Delay(blinkTime);
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */
```

```c
// ---------- skilltest2561 ----------
// Implement the following:
// 1 The indicator LEDs blinks every one second (500ms on, 500ms off) after reset
// 2 Pushing the USER button the first time will change the state such that only one LED is blinking. ( choose any)
// 3 Pushing the USER button the second time will change the state such that only two LEDs are blinking.
// 4 Pushing the USER button the third time will go back to the first state after reset

// Setup: HSE clock, GPIO mode Rising edge, NVIC 2 bit 2

/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */
uint16_t state = 0;

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_15);
    HAL_Delay(100);

    if (state == 0)
        state = 1;
    else if (state == 1)
        state = 2;
    else
        state = 0;

    __HAL_GPIO_EXTI_CLEAR_IT(GPIO_Pin);
}

while (1)
{
    if (state == 0)
    {
        // HAL_GPIO_WritePin: Set Status
        // off: GPIO_PIN_RESET, on: GPIO_PIN_SET
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5 | GPIO_PIN_6, GPIO_PIN_RESET); // All LEDs off
        // debouncing the button
        while (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET)
            ;
    }
    else if (state == 1)
    {
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_RESET); // Turn off LED
        HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
        HAL_Delay(500); // Delay for 500ms
        // debouncing the button
        while (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET)
            ;
    }
    else if (state == 2)
    {
        HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5 | GPIO_PIN_6);
        HAL_Delay(500);
        // debouncing the button
        while (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET)
            ;
    }
```

```c
// ---------- Skill Test Sec 1 2562 ----------
// 1. Make red and green lights blink every ms.
// 2. Use user push button to to make red and green lights blink every ms.
// 3. Hold push button at least
// push button anymore.

/* USER CODE BEGIN WHILE */
uint8_t state = 1;

while (1)
{
    if (state == 1)
    {
        HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5 | GPIO_PIN_6);
        HAL_Delay(75); // Delay for 75ms
        if (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET)
        {
            state++;
            // debouncing the button
            while (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET)
                ;
        }
    }
    else if (state == 2)
    {
        HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5 | GPIO_PIN_6);
        HAL_Delay(15); // Delay for 75ms
        if (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET)
        {
            state++;
            // debouncing the button
            while (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET)
                ;
        }
    }
    else
    {
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5 | GPIO_PIN_6, GPIO_PIN_RESET); // All LEDs off
    }
}
```