

Assignment-5

Challenge – 1:

Software user input to hardware action

STM32L475 has usart1 interface directly available on ST-Link. You can use this interface for printing output messages and taking user input if you're not using it for advance trace debugging over st-link.

Enable and configure UART1 as per experiment completed for UART2. Use polling method in while loop.

Write a program to user inputs from UART1. The board functional behavior as per user input should be as following-

=> If user sends number 1 from keyboard on serial terminal - LED1 turns HIGH & prints "LED1 ON"

=> If user sends number 2 from keyboard on serial terminal - both LED1 & LED2 turns HIGH & prints "LED1 & LED 2 ON"

=> If user sends number 3 from keyboard on serial terminal - both LED1 & LED2 switches off & prints "LED1 & LED 2 OFF"



=> Any other user input from terminal - both LED1 & LED2 starts blinking & prints "INVALID INPUT"

Challenge – 2:

Morse Code LED Indicator

Configure uart1 in DMA mode.

Write the function for taking user input as per morsecode valid characters and configure LED1 on STM32L475 to indicate accordingly.

main.c  Ass5ch1.ioc  startup_stm32l475vgtx.s

```
43 DFSDM_Channel_HandleTypeDef hdfsdm1_channel1;
44
45 I2C_HandleTypeDef hi2c2;
46
47 QSPI_HandleTypeDef hqspi;
48
49 SPI_HandleTypeDef hspi3;
50
51 UART_HandleTypeDef huart1;
52 UART_HandleTypeDef huart2;
53 UART_HandleTypeDef huart3;
54
55 PCD_HandleTypeDef hpcd_USB_OTG_FS;
56
57 /* USER CODE BEGIN PV */
58 uint8_t msg1[]="LED1 ON";
59 uint8_t msg2[]="LED1 & LED 2 ON";
60 uint8_t msg3[]="LED1 & LED 2 OFF";
61 uint8_t msg[]="INVALID INPUT";
62 uint8_t count[]="";
63
64 /* USER CODE END PV */
65
66 /* Private function prototypes -----
67 void SystemClock_Config(void);
68 static void MX_GPIO_Init(void);
69 static void MX_DFSDM1_Init(void);
70 static void MX_I2C2_Init(void);
71 static void MX_QUADSPI_Init(void);
72 static void MX_SPI3_Init(void);
73 static void MX_USART1_UART_Init(void);
74 static void MX_USART3_UART_Init(void);
75 static void MX_USB_OTG_FS_PCD_Init(void);
```

<

```

while (1)
{
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    HAL_UART_Receive(&huart1,&count,1,1000);
    if(*count == '1'){
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5,GPIO_PIN_SET);
        HAL_UART_Transmit(&huart1, msg1, sizeof(msg1), 1000);
    /*
    count++;
    */
    }
    else if(*count == '2'){
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5,GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_14,GPIO_PIN_SET);
        HAL_UART_Transmit(&huart1, msg2, sizeof(msg2), 1000);
    /*
    count++;
    */
    }
    else if(*count == '3'){
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5,GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_14,GPIO_PIN_RESET);
        HAL_UART_Transmit(&huart1, msg3, sizeof(msg3), 1000);
    /*
    count++;
    */
    }
    else{
        HAL_GPIO_TogglePin (GPIOA,GPIO_PIN_5);
        HAL_Delay(200);
        HAL_GPIO_TogglePin (GPIOB,GPIO_PIN_14);
        HAL_Delay(200);
        HAL_UART_Transmit(&huart1, msg, sizeof(msg), 1000);
    }

}

```

```

while(1)
{
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    HAL_UART_Transmit(&huart1, "\r\n", 2, 1000);
    HAL_UART_Transmit(&huart1, "-----\r\n",
    HAL_UART_Transmit(&huart1, "Please type your first initial\r\n", 32, 1000);
    HAL_UART_Transmit(&huart1, "Use only capital letter and 1 character:\r\n",

    checkLetter(); // Check the first initial if it is between [A-Z]

    init1 = cmd; // Store the character if it is between [A-Z]
    //
    HAL_UART_Transmit(&huart1, "-----\r\n",
    HAL_UART_Transmit(&huart1, "Please type your second initial\r\n", 33, 1000);
    HAL_UART_Transmit(&huart1, "Use only capital letter and 1 character:\r\n",

    checkLetter(); // Check the second initial if it is between [A-Z]

    init2 = cmd; // Store the character if it is between [A-Z]

    start();
    /* USER CODE END 3 */
}

void start() {
    inits[0] = init1;
    inits[1] = init2;

    HAL_UART_Transmit(&huart1, "-----\r\n", 42, 1000);
    HAL_UART_Transmit(&huart1, "Your initials - ", 16, 1000);
    HAL_UART_Transmit(&huart1, inits, 2, 1000);
    HAL_UART_Transmit(&huart1, " - are in progress..\r\n", 22, 1000);

    ledMorseAlphabet(init1); // Match the first character between A-Z and make the morse

    space(); // 3 morse unit time space between 2 initials
    space();
    space();

    ledMorseAlphabet(init2); // Match the second character between A-Z and make the morse

    space(); // 3 morse unit time space between 2 initials
    space();
    space();
    HAL_UART_Transmit(&huart1, "\r\n", 2, 1000);
    HAL_UART_Transmit(&huart1, "COMPLETED!\r\n", 12, 1000);

    return;
}

```

```

15
16=void dot() {
17    HAL_UART_Transmit(&huart1, " . ", 3, 1000); //Serial output to show user, current sta
18    ledTiming(UNIT_TIME, UNIT_TIME);
19    return;
20
21 }
22
23 |
24
25=void dash() {
26    HAL_UART_Transmit(&huart1, " - ", 3, 1000); //Serial output to show user, current sta
27    ledTiming(3 * UNIT_TIME, UNIT_TIME);
28    return;
29 }
30
31
32=void ledTiming(int onTime, int offTime) {
33
34    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);
35    HAL_Delay(onTime);
36    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
37    HAL_Delay(offTime);
38    return;
39 }
40
41
42
43=void space() {
44    HAL_Delay(UNIT_TIME);
45    HAL_UART_Transmit(&huart1, "   ", 3, 1000); //Serial output to show user, current sta
46    return;
47 }

```

```
void ledMorseAlphabet(uint8_t input) {  
  
    switch (input) {  
        case 'A':  
            dot();space();dash();  
            break;  
        case 'B':  
            dash();space();dot();space();dot();space();dot();  
            break;  
        case 'C':  
            dash();space();dot();space();dash();space();dot();  
            break;  
        case 'D':  
            dash();space();dot();space();dot();space();  
            break;  
        case 'E':  
            dot();  
            break;  
        case 'F':  
            dot();space();dot();space();dash();space();dot();  
            break;  
        case 'G':  
            dash();space();dash();space();dot();  
            break;  
        case 'H':  
            dot();space();dot();space();dot();space();dot();  
            break;  
        case 'I':  
            dot();space();dot();  
            break;  
        case 'J':  
            dot();space();dash();space();dash();space();dash();
```

```

case 'K':
    dash();space();dot();space();dash();
    break;
case 'L':
    dot();space();dash();space();dot();space();dot();
    break;
case 'M':
    dash();space();dash();
case 'N':
    dash();space();dot();
case 'O':
    dash();space();dash();space();dash();space();
    break;
case 'P':
    dot();space();dash();space();dash();space();dot();
    break;
case 'Q':
    dash();space();dash();space();dot();space();dash();
    break;
case 'R':
    dot();space();dash();space();dot();
    break;
case 'S':
    dot();space();dot();space();dot();
    break;
case 'T':
    dash();
    break;
case 'U':
    dot();space();dot();space();dash();
    break;
case 'V':
    dot();dash();dot();dash();dot();dash();dot();dash();
    break;

```

```

        break;
    case 'U':
        dot();space();dot();space();dash();
        break;
    case 'V':
        dot();space();dot();space();dot();space();dash();
        break;
    case 'W':
        dot();space();dash();space();dash();
        break;
    case 'X':
        dash();space();dot();space();dot();space();dash();
        break;
    case 'Y':
        dash();space();dot();space();dash();space();dash();
        break;
    case 'Z':
        dash();space();dash();space();dot();space();dot();
        break;
    default:
        // Error state
        dash();space();dash();space();dash();space();dash();space();dash();s
    }
}

FUNCTION : checkLetter

void checkLetter() {
    while (1) {
        char c;
        if (cmdstate == 1) {
            for (c = 'A'; c <= 'Z'; ++c) {
                if (cmd == c) {
                    HAL_UART_Transmit(&huart1, " is one of your initials\r\n", 26, 1000);

                    //cmdstate is like control bit of if user typed a input.
                    //if it is not this function will be wait inside while loop for user to type.
                    cmdstate = 0;
                    return 0;
                }
            }
            HAL_UART_Transmit(&huart1, " is not valid. Please select between [A-Z]\r\n", 44, 1000);
            HAL_UART_Transmit(&huart1, "-----\r\n", 42, 1000);
            cmdstate = 0;
        }
    }
}

```


File Edit Setup Control Window Help

Please type your first initial

Use only capital letter and 1 character:

S is one of your initials

Please type your second initial

Use only capital letter and 1 character:

Z is one of your initials

Your initials - SZ - are in progress..

COMPLETED!

Please type your first initial

Use only capital letter and 1 character:

a is not valid. Please select between [A-Z]

A is one of your initials

Please type your second initial

Use only capital letter and 1 character:

- is not valid. Please select between [A-Z]

A is one of your initials

Your initials - AA - are in progress..

COM4 - Tera Term VT

File Edit Setup Control Window Help

Please type your first initial

Use only capital letter and 1 character:

a is not valid. Please select between [A-Z]

A is one of your initials

Please type your second initial

Use only capital letter and 1 character:

- is not valid. Please select between [A-Z]

A is one of your initials

Your initials - AA - are in progress..

COMPLETED!

Please type your first initial

Use only capital letter and 1 character: