

微處理機實驗期末考結報

第一題

```
%% 考題一
st1=>start: 初始狀態不閃爍
op1=>operation: 調整可變電阻可控制兩燈閃爍快慢
op2=>operation: 按下藍色USER按鈕改變速度
cond1=>condition: 可變電阻順時針或逆時針?
op3=>operation: LED1與LED3快速閃爍
op4=>operation: LED1與LED3慢速閃爍

st1->op1->op2->cond1
cond1(yes)->op3
cond1(no)->op4
```

解題流程：

1. 使用可變電阻來讀取模擬值，通過ADC轉換得到0~4095的數值。
2. 使用PWM信號來控制LED的閃爍，根據可變電阻的數值來調整閃爍間隔。
3. 按下USER按鈕時觸發中斷來改變閃爍速度。按鈕按下後，改變計時的PWM頻率，實現閃爍速度的改變。
4. 使用定時器和中斷來控制時間間隔。

程式碼

main.c -> main() -> while(1)

```
while(1)
{
    HAL_Delay(flag);
    HAL_GPIO_WritePin(LED1_GPIO_Port, LED1_Pin, 0);
    HAL_GPIO_WritePin(LED3_GPIO_Port, LED3_Pin, 1);
    HAL_Delay(flag);
    HAL_GPIO_WritePin(LED1_GPIO_Port, LED1_Pin, 1);
    HAL_GPIO_WritePin(LED3_GPIO_Port, LED3_Pin, 0);
    /* USER CODE END WHILE */

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

main.c -> void HAL_GPIO_EXTI_Callback()

```
/* USER CODE BEGIN 4 */
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
```

```
uint32_t adc_value = HAL_ADC_GetValue(&hadc1);
flag = (adc_value / 4095.0) * 500;
}
```

第二題

```
%% 考題二
st2=>start: 按下藍色USER按鈕
op5=>operation: 字元"A"儲存至EEPROM
cond2=>condition: 三秒內是否未按下?
op6=>operation: 資料顯示於螢幕
op7=>operation: 按下五次則顯示"AAAAA"

st2->op5->cond2
cond2(yes)->op6
cond2(no)->op7->op6
```

解題流程

1. 使用按鈕中斷來觸發存儲“A”到EEPROM中。
2. 利用計時器和中斷來判定三秒內是否按下按鈕，未按下時觸發讀取EEPROM的資料並顯示。
3. 使用I2C總線與EEPROM進行通信，讀寫資料。
4. 按下五次按鈕後，顯示“AAAAA”。

main.c -> main()

```
/* USER CODE BEGIN 2 */
HAL_TIM_Base_Start_IT(&htim6);
/* USER CODE END 2 */
```

main.c -> HAL_GPIO_EXTI_Callback()

```
/* USER CODE BEGIN 4 */
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin) {
    if (GPIO_Pin == button_Pin && HAL_GPIO_ReadPin(button_GPIO_Port, button_Pin)) {
        HAL_Delay(200);
        HAL_GPIO_TogglePin(GPIOB, LED1_Pin);
        __HAL_TIM_SET_COUNTER(&htim6, 0);
        strcat(write_data, "A");

        EEPROM_Write(0, 0, write_data, strlen(write_data));
        HAL_Delay(100);

        HAL_UART_Transmit(&huart3, (uint8_t*)"write data:", strlen("write data:"),
100);
        HAL_UART_Transmit(&huart3, write_data, strlen(write_data), 100);
    }
}
```

```

    HAL_UART_Transmit(&huart3, (uint8_t*)"r\n", strlen("r\n"), 100);
} else {
    __NOP();
}
}

```

main.c -> HAL_TIM_PeriodElapsedCallback()

```

void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim) {
    if(htim == &htim6) {
        HAL_GPIO_TogglePin(GPIOB, LED2_Pin);
        if(strlen(write_data)==0){
            return;
        }
        EEPROM_Read(0, 0, read_data, strlen(write_data));
        HAL_UART_Transmit(&huart3, (uint8_t*)"read data: ", strlen("read data: "),
100);
        HAL_UART_Transmit(&huart3, read_data, strlen(read_data), 100);
        HAL_UART_Transmit(&huart3, (uint8_t*)"r\n", strlen("r\n"), 100);
    }
}
/* USER CODE END 4 */

```

第三題

```

%% 考題三
st3=>start: 按下藍色USER按鈕
op8=>operation: 讀取可變電阻值ADC_value
op9=>operation: 映射ADC_value至0~3數值
op10=>operation: 螢幕顯示ADC_value和flag
op11=>operation: 數值透過LED顯示
op12=>operation: SPI寫入"Ready"
op13=>operation: 讀取SPI資料並判斷正確性
cond3=>condition: 資料正確?
op14=>operation: 重新寫入

st3->op8->op9->op10->op11->op12->op13->cond3
cond3(yes)->op14
cond3(no)->op12

```

解題流程

1. 使用ADC讀取可變電阻的值並將其映射到0~3之間，這樣可以通過LED顯示對應的數值。
2. 使用SPI通信將"Ready"訊息寫入，並讀取回來確認資料正確性。
3. 按下按鈕時觸發讀取可變電阻數值，並顯示映射結果。 main.c -> main()

```

/* USER CODE BEGIN 2 */
HAL_ADC_Start(&hadc1);

/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while(1) {
    if (flag<1){
        HAL_GPIO_WritePin(LED1_GPIO_Port, LED1_Pin, 0);
        HAL_GPIO_WritePin(LED2_GPIO_Port, LED2_Pin, 0);
    }else if (2>flag &&flag>=1){
        HAL_GPIO_WritePin(LED1_GPIO_Port, LED1_Pin, 0);
        HAL_GPIO_WritePin(LED2_GPIO_Port, LED2_Pin, 1);
    }else if (3>flag && flag>=2){
        HAL_GPIO_WritePin(LED1_GPIO_Port, LED1_Pin, 1);
        HAL_GPIO_WritePin(LED2_GPIO_Port, LED2_Pin, 0);
    }else if (flag>=3){
        HAL_GPIO_WritePin(LED1_GPIO_Port, LED1_Pin, 1);
        HAL_GPIO_WritePin(LED2_GPIO_Port, LED2_Pin, 1);
    }
    /* USER CODE END WHILE */
    /* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */

```

main.c -> void HAL_GPIO_EXTI_Callback()

```

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin) {
    // HAL_GPIO_TogglePin(GPIOB, LED2_Pin);
    HAL_Delay(1000);
    if (HAL_ADC_PollForConversion(&hadc1, 1000) == HAL_OK) {
        adc_value = HAL_ADC_GetValue(&hadc1);
        flag = (adc_value / 4095.0) * 3;
        print(adc_value, flag);
    }
}

```