Group 12 Lab 04

Question :

Write a program where the main loop monitors the state of a GPIO input pin (SW1) and turns on RED LED whenever SW1 is pressed (and turns off RED LED whenever SW1 is released). The interrupt handler should control the blue LED and toggle it every 500ms i.e. create a 1Hz blink rate of the blue LED.

Solution :

Setting Up =>

Setting up Port F registers and Systick registers
STRELOAD value determines the speed at which led toggles.

```
    SYSCTL_RCGCGPIO_R = 0x20; // This enables clock for the GPIO Port F
Register
    GPIO_PORTF_LOCK_R = 0x4C4F434B;
    GPIO_PORTF_CR_R = 0xFF;
    GPIO_PORTF_DIR_R = 0x0E;
    GPIO_PORTF_PUR_R = 0x11;
    GPIO_PORTF_DEN_R = 0x1F;
    STRELOAD = 500*CLOCK_MHZ*1000;
    STCTRL |= (CLKINT | ENABLE | STINT);
```

Red Led Working =>

Code for RED LED turning on when SW1 is pressed and off when released
We check if button is pressed by 1st bit of GPIO data and turn on red led by setting 2nd bit.
When released 2nd bit is cleared.

```
    while(1){
        switch(GPIO_PORTF_DATA_R & 0x11)
        {
        case 0x11:
            GPIO_PORTF_DATA_R &= 0xFFFFFFFD;
            break;
        case 0x10:
```

```
            GPIO_PORTF_DATA_R &= 0xFFFFFFFD;
            break;
        case 0x01:
            GPIO_PORTF_DATA_R |= 0x02;
            break;
        }
    };
```

Blue Led Toggle Working =>

Set up SysTickInterruptHandler in startup file by defining it as extern and replacing the IntDefaultHandler with SysTickHandler

```
extern void SysTickHandler(void);
```

Now define the handler to toggle blue led in the .c file

```
void SysTickHandler(void){
    GPIO_PORTF_DATA_R ^= 0x04;
}
```