

# BLINKY AND MACROS

## ASSIGNMENT - 02

BASATI SIVAKRISHNA

22976

DESE EPD

### CODE OVERVIEW:

The code runs in interrupt mode. When a push button is pressed the respective part in GPIOHANDLER will get execute and set the global variables accordingly.

Based on the values set by the GPIOHANDLER the variables will control the type of led colour and speed rate of blinking.

### **MACROS:**

```
15
16 #define red_on      GPIO_PORTF_DATA_R = 0x02;
17 #define magenta_on  GPIO_PORTF_DATA_R = 0x06;
18 #define blue_on     GPIO_PORTF_DATA_R = 0x04;
19 #define cyan_on     GPIO_PORTF_DATA_R = 0x0C;
20 #define green_on    GPIO_PORTF_DATA_R = 0x08;
21 #define yellow_on   GPIO_PORTF_DATA_R = 0x0A;
22 #define white_on    GPIO_PORTF_DATA_R = 0x0E;
23 #define led_off      GPIO_PORTF_DATA_R = 0x00;
24 #define offset_delay 1000
25
```

### **Interrupt Handler:**

```
static void GPIOPortF_Handler(void)
{
    if((GPIO_PORTF_MIS_R & 0x10)==0x10)
    {
        GPIO_PORTF_ICR_R = 0x10;
        (led_type > 5) ? (led_type = 0): (led_type++);
    }
    if((GPIO_PORTF_MIS_R & 0x01)==0x01)
    {
        GPIO_PORTF_ICR_R = 0x01;
        (speed_count <= 1) ? (speed_count = 32): (speed_count = speed_count/2);
    }
}
```

**Speed\_count variable is responsible for blinking speed and led\_type is responsible for colour of light to be shown.**

## MAIN FUNCTION:

Main function will be using a switch case to blink an led according to the led\_type, and some delay instructions (controlled by speed\_count variable):

```
2   while(1) {
3       switch(led_type) {
4           case 0:
5               green_on;
5               break;
7           case 1:
3               blue_on;
3               break;
3           case 2:
1               cyan_on;
2               break;
3           case 3:
4               red_on;
5               break;
5           case 4:
7               yellow_on;
3               break;
3           case 5:
3               magenta_on;
1               break;
2           case 6:
3               white_on;
4               break;
5           case 7:
5               white_on;
7               break;
3           default:
3               break;
3       }

    ,
81
82     delayMs(offset_delay / speed_count);
83     if(speed_count != 1) led_off;
84     delayMs(offset_delay / speed_count);
85
86 }
87 }
```

Since we configured pushbuttons as interrupts we don't need to poll check for the Inputs continuously.