```
/*******************************************************************************
 * Name:    GPIO.c
 * Description: STM32 GPIO on the ARM Cortex M3
 * Version: V1.00
 * Authors: Li Pan
 *
 *******************************************************************************/
#include "stm32f10x.h"
#include "GPIO.h"
#include "CLOCK.h"


void led_IO_GPIOA(void)
{
      //Enable LED PORTA 4 pin for 4 LEDs
            GPIOA->CRH |= GPIO_CRH_MODE9 | GPIO_CRH_MODE10 | GPIO_CRH_MODE11|
GPIO_CRH_MODE12 ;
    GPIOA->CRH &= ~GPIO_CRH_CNF9 & ~GPIO_CRH_CNF10 & ~GPIO_CRH_CNF11 &
~GPIO_CRH_CNF12 ;
}


void led_enable(void)
{
      /*Using the switch control LED*/

      //GPIOA->ODR=(GPIOA->ODR&(~(0xF<<9)))|(read_dip()<<9);

      /*Using the button control LED*/

            GPIOA->ODR=(GPIOA->ODR&(~(0xF<<9)))|(read_bu()<<9);

}

/*****************************
        Check the Switch
*****************************/
uint16_t read_dip(void)
{
      uint16_t sw_val;

      sw_val=((( GPIOA->IDR&(GPIO_IDR_IDR6|GPIO_IDR_IDR7))>>6)|((GPIOC-
>IDR&(GPIO_IDR_IDR10|GPIO_IDR_IDR11))>>8))&0x0F;


      return  sw_val;

}
/*****************************
            Check the Button
*****************************/

uint16_t read_bu(void)
{
```

```
	uint16_t bu_val;

	bu_val=(((GPIOB->IDR&(GPIO_IDR_IDR8|GPIO_IDR_IDR9))>>8)|((GPIOC-
>IDR&(GPIO_IDR_IDR12))>>10)|((GPIOA->IDR&(GPIO_IDR_IDR5))>>2))&0x0F;


	return  bu_val;


}
/*Read the hex value from switches*/
uint16_t read_SW(void)
{
	uint16_t SW_val;

	SW_val=(((GPIOA->IDR&(GPIO_IDR_IDR6))>>3)|((GPIOA-
>IDR&(GPIO_IDR_IDR7))>>5)|((GPIOC->IDR&(GPIO_IDR_IDR10))>>9)|((GPIOC-
>IDR&(GPIO_IDR_IDR11))>>11))&0x0F;


	return  SW_val;

}
/*Pattern for display leds*/
void LED_WigWag(void)
{
	int i=1200000;
	GPIOA->ODR=GPIOA->ODR & 0xfffffcff;
	delay(i);
	GPIOA->ODR=GPIOA->ODR ^ 0x600;
	delay(i);
	GPIOA->ODR=GPIOA->ODR ^ 0xc00;
	delay(i);
	GPIOA->ODR=GPIOA->ODR ^ 0x1800;
	delay(i);

}
 /* check the input from PA5*/
uint32_t read_LSensor(void)
{
	uint32_t Sensor_val;

	Sensor_val=GPIOA->IDR&(GPIO_IDR_IDR5);

	return Sensor_val;

}
 /* check the input from PA4*/
uint32_t read_RSensor(void)
{
	uint32_t Sensor_val;

	Sensor_val=GPIOA->IDR&(GPIO_IDR_IDR4);
```

```
      return Sensor_val;

}
/*initial the Port B pin 12 and 13 for signal lights output*/
void LED(void)
{
            RCC->APB2ENR |=  RCC_APB2ENR_IOPBEN;
            GPIOB->CRH |= GPIO_CRH_MODE12 |GPIO_CRH_MODE13;
            GPIOB->CRH &= ~GPIO_CRH_CNF12 &~GPIO_CRH_CNF13;

}

/*set PB12 and reset PB13*/
void LEFT_LED(void)
{

            GPIOB->BSRR = GPIO_BSRR_BS12;
            GPIOB->BSRR = GPIO_BSRR_BR13;

}
/*reset PB12 and set PB13*/
void RIGHT_LED(void)
{


            GPIOB->BSRR = GPIO_BSRR_BR12;
            GPIOB->BSRR = GPIO_BSRR_BS13;

}
/*reset PB12 and reset PB13*/
void NO_TURN(void)
{


            GPIOB->BSRR = GPIO_BSRR_BR12;
            GPIOB->BSRR = GPIO_BSRR_BR13;

}
/*set PB12 and set PB13*/
void STOP(void)
{

            GPIOB->BSRR = GPIO_BSRR_BS12;
            GPIOB->BSRR = GPIO_BSRR_BS13;

}
```