Week 3 Blinky Study STM32F4429i-DISC

Code description

The program flashing two leds as a chaser and stops flashing the leds when the button is pressed. A global variable (flashLEDs) is used to indicate if the Leds should be flashing and a interrupt attached to the button is used to toggle its state.

In main

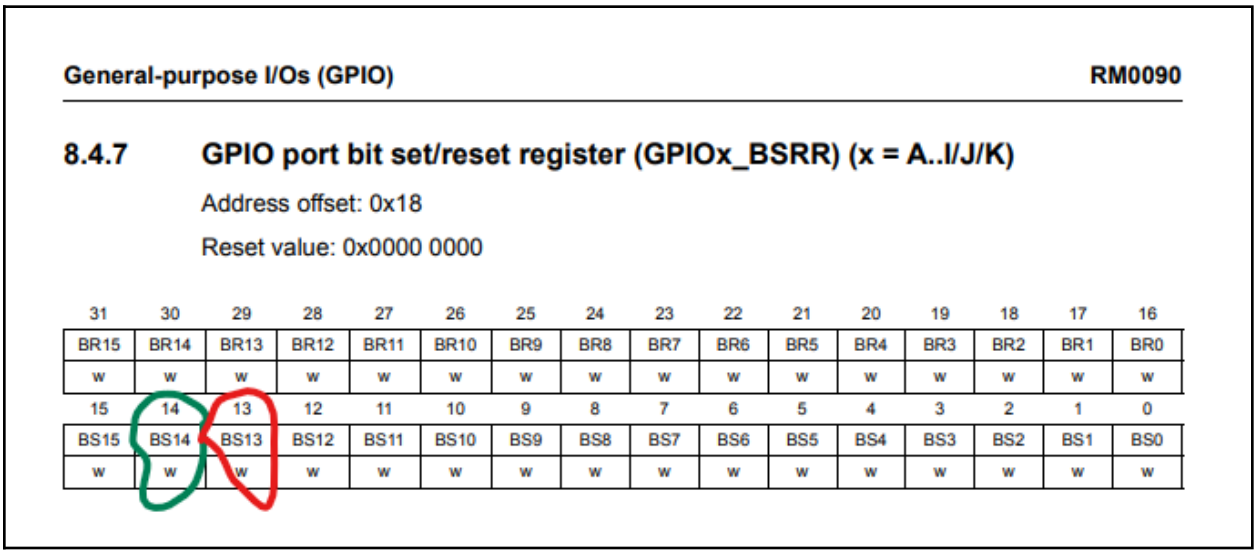
```
while (1)
  {
       if(flashLEDs){
             HAL_GPIO_TogglePin(GPIOG, GPIO_PIN_13);
             HAL_Delay(100);
             HAL_GPIO_TogglePin(GPIOG, GPIO_PIN_14);
       }
       else{
             HAL_GPIO_WritePin(GPIOG, GPIO_PIN_13, 0);
             HAL_GPIO_WritePin(GPIOG, GPIO_PIN_14, 0);
       }
  }
```

In the interrupt handler

```
void EXTI0_IRQHandler(void)
{
  /* USER CODE BEGIN EXTI0_IRQn 0 */
  extern int flashLEDs;
  flashLEDs = flashLEDs ^ 1;
  /* USER CODE END EXTI0_IRQn 0 */
  HAL_GPIO_EXTI_IRQHandler(Button_Pin);
  /* USER CODE BEGIN EXTI0_IRQn 1 */

  /* USER CODE END EXTI0_IRQn 1 */
}
```

What are the hardware registers that cause the LED to turn on and off?



General-purpose I/Os (GPIO) RM0090

8.4.7 GPIO port bit set/reset register (GPIOx_BSRR) (x = A..I/J/K)

Address offset: 0x18

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

In my example i'm using the LEDS attached to PG13 and PG14. The BS14 bit is used to control the output of PG14 while BS13 is used to control PG13, of note is that in order to read the state of these output pins we can not use GPIOG_BSRR as it is write only, we must use GPIOG_ODR. PG13 and PG14 are on port G. The GPIOG_ODR register address is 0x40021814

What are the registers that you read in order to find out the state of the button?

8.4.5 GPIO port input data register (GPIOx_IDR) (x = A..I/J/K)

Address offset: 0x10

Reset value: 0x0000 XXXX (where X means undefined)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **IDRy**: Port input data (y = 0..15)

These bits are read-only and can be accessed in word mode only. They contain the input value of the corresponding I/O port.

The button is attached to pin PA0, which is attached to port A. Therefore we can read its state by using the GPIOA_IDR register (0x40020010), a simpler way is to simply use the HAL library HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0)