

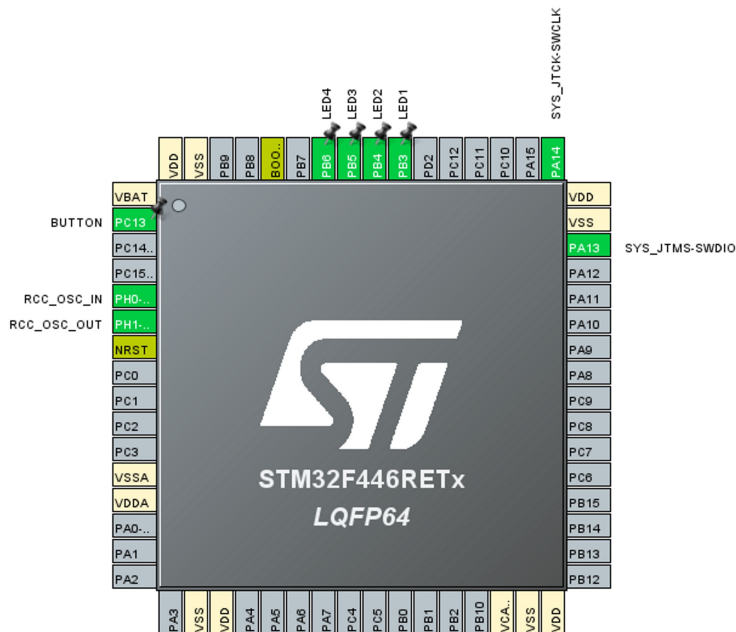
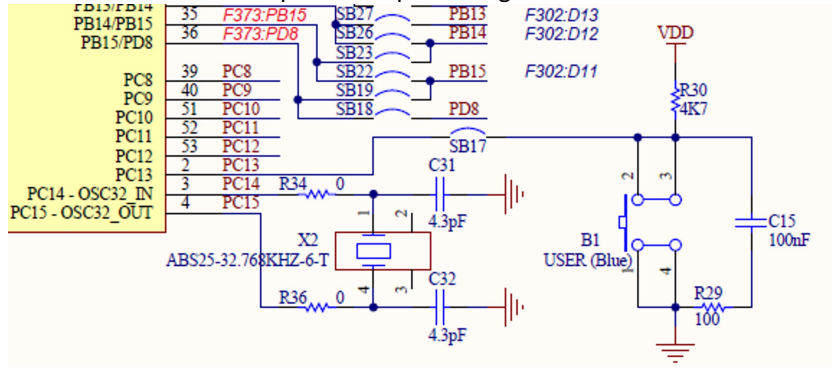
25 Aralık 2021 Cumartesi 00:52

Buton ile Led Yakma

 HAL

Konfigürasyon Kısmı

- Kullanıcı butonu C portun 13.pinine bağlı.



Pin Name	Signal on Pin	GPIO output...	GPIO mode	GPIO Pull-u...	Maximum o...	User Label	Modified
PB3	n/a	Low	Output Push...	No pull-up a...	Very High	LED1	✓
PB4	n/a	Low	Output Push...	No pull-up a...	Very High	LED2	✓
PB5	n/a	Low	Output Push...	No pull-up a...	Very High	LED3	✓
PB6	n/a	Low	Output Push...	No pull-up a...	Very High	LED4	✓
PC13	n/a	n/a	Input mode	Pull-down	n/a	BUTTON	✓

Kod Kısmı

- RCC için bir değişiklik yapmadığımızdan fonksiyon içeriği aynıdır.
- hal_gpio.c kısmından ReadPin fonksiyonlarına ulaşabiliriz.

```

375 GPIO_PinState HAL_GPIO_ReadPin(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin)
376 {
377     GPIO_PinState bitstatus;
378
379     /* Check the parameters */
380     assert_param(IS_GPIO_PIN(GPIO_Pin));
381
382     if((GPIOx->IDR & GPIO_Pin) != (uint32_t)GPIO_PIN_RESET)
383     {
384         bitstatus = GPIO_PIN_SET;
385     }
386     else
387     {
388         bitstatus = GPIO_PIN_RESET;
389     }
390     return bitstatus;
391 }

```

- Buton için count değişkeni atıyoruz.

```

44 /* USER CODE BEGIN PV */
45 int count=0;
46 /* USER CODE END PV */

```

- Okuma yaptığında yani butona bastığımızda if yapısının içine girer ve while ile kullanıcının eli butona basılı olup olmadığını kontrol ederiz. Elini çektiğinde count değerini 1 artırıyor.
- Count değerinin modunu aldığımızda 0 iken RESET, 1 iken SET durumundadır.

```

94 while (1)
95 {
96     /* USER CODE END WHILE */
97
98     /* USER CODE BEGIN 3 */
99     if(HAL_GPIO_ReadPin(GPIOC, BUTTON_Pin))
100     {
101         while(HAL_GPIO_ReadPin(GPIOC, BUTTON_Pin));
102         HAL_Delay(100);
103         count++;
104     }
105
106     if(count % 2 == 1)
107     {
108         HAL_GPIO_WritePin(GPIOB, LED1_Pin | LED2_Pin | LED3_Pin | LED4_Pin,GPIO_PIN_SET);
109     }
110     else
111     {
112         HAL_GPIO_WritePin(GPIOB, LED1_Pin | LED2_Pin | LED3_Pin | LED4_Pin,GPIO_PIN_RESET);
113     }
114 }

```

➤ REGISTER

Konfigürasyon Kısmı

- Öncelikle RCC ve GPIO ayarlarını yapıyoruz.

```

8 void RCC_Config(void)
9 {
10     RCC->CR |= 0x00030000;           //HSEON, HSERDY
11     while(!(RCC->CR & 0x00020000)); //HSERDY
12     RCC->CR |= 0x00080000;           //CSSON
13     RCC->CFGR = 0x00000000;
14     RCC->PLLCFGR |= 0x00400000;       //PLLSRC
15     RCC->PLLCFGR |= 0x00000004;       //PLLM 4
16     RCC->PLLCFGR |= 0x00002A00;       //PLLN 168
17     RCC->PLLCFGR |= 0x00000000;       //PLL P 2
18     RCC->CR |= 0x01000000;           //PLLON
19     while(!(RCC->CR & 0x02000000)); //PLL RDY
20     RCC->CFGR |= 0x00000001;         //SW
21     while(!(RCC->CR & 0x00000001)); //SWS
22 }

```

- GPIO ayarlarını aşağıdaki gibi düzelttik.
- A portunu da kullanacağımızdan A ve D portlarını RCC clocklarını aktif ediyoruz.

```

RCC->AHB1ENR |= 0x00000009;           //A, D clock enable

```

```

24 void GPIO_Config(void)
25 {
26     RCC->AHB1ENR |= 0x00000009;           //A, D clock enable
27
28     GPIOD->MODER |= 0x55000000;           //PD12, PD13, PD14, PD15
29     GPIOD->OTYPER |= 0x00000000;          //Output push-pull
30     GPIOD->OSPEEDR |= 0xFF000000;         //Very high speed
31     GPIOD->PUPDR |= 0x00000000;          //No pull-up, pull-down
32 }

```

Kod Kısmı

- Buton için A0 pini için okuma yapacağız. Burada ilk biti kullanacağız. While döngüsü içinde bir if yapısı içinde eğer okuma yapılırsa while döngüsüne girer ve okuma yani butona basma devam ediyorsa count değişkenini 1 arttırır.
- Buradaki delay ark olayından dolayı kullandık.

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.

```

if(GPIOA->IDR & 0x00000001)
{
    while(GPIOA->IDR & 0x00000001);
    delay(1680000);

    count++;
}

```

- Count ve delay için globalde tanıttık.
- While döngüsünün içi 1 olduğu sürece çalışmaya devam eder ancak 0 olduğunda döngüden çıkıp alt satıra geçer.

```

3 int count = 0;
4
5 void delay(uint32_t time)
6 {
7     while(time--);
8 }

```

- Count değişkenini kalanı 0 ise yani 2.defa basıldığında ledi söndürken kalan 1 olduğunda yani 1.defa basıldığı durumda ledi yakacaktır.

```

57 int main(void)
58 {
59     RCC_Config();
60     SystemCoreClockUpdate();
61
62     GPIO_Config();
63
64     while (1)
65     {
66         if(GPIOA->IDR & 0x00000001)
67         {
68             while(GPIOA->IDR & 0x00000001);
69             delay(1680000);
70
71             count++;
72         }
73         if(count %2 == 0)
74             GPIOD->ODR |= 0x00000000;
75         else
76             GPIOD->ODR |= 0x0000F000;
77     }
78 }

```