

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»

Институт компьютерных наук и технологий

Высшая школа искусственного интеллекта

Дисциплина:
ПРОГРАММИРОВАНИЕ МИКРОКОНТРОЛЛЕРОВ

ОТЧЕТ
По лабораторной работе №3
Тема: События и прерывания

Обучающийся гр. 3530201/10001

Нгуен Куок Дат

Руководитель _____ Вербова Наталья Михайловна

Санкт-Петербург 2022

Содержание

1	Цель и постановка задачи	2
1.1	Цель работы	2
1.2	Постановка задачи	2
2	Выполнение задания	3
2.1	Код программы	3
3	Вывод	8

1 Цель и постановка задачи

1.1 Цель работы

Ознакомится с основными приемами работы с документацией при составлении программ для микроконтроллеров.

1.2 Постановка задачи

Разработать программу для микроконтроллера (МК) STM32F200 мигающую светодиодом PG7 и регистрирующую и обрабатывающую с разным приоритетом замыкание кнопок “WAKEUP” и “USER”. При нажатии кнопки “WAKEUP” на некоторое время должен подключаться светодиод PG6, а при нажатии кнопки “USER” светодиод PG8

2 Выполнение задания

2.1 Код программы

```
//WAKEUP -> PG6, USER ->PG8
#include "stm32f2xx.h"
#include "stm32f2xx_hal_gpio.h"
#define TIME_DELAY 200000
unsigned int j;
void delay(){
    for (j = 0 ; j < 10; j++){
    }
void PG7(){
    while (1){
        GPIOG->ODR |= (1ul << 7);
        for (j = 0; j < TIME_DELAY; j++) {};
        GPIOG->ODR &= ~(1ul << 7);
        for (j = 0; j < TIME_DELAY; j++) {};
    }
}
void PG7off(){
        GPIOG->ODR &= ~(1ul << 7);
}
void PG6off(){
        GPIOG->ODR &= ~(1ul << 6);
```

```

}

void PG8off(){

    GPIOG->ODR &= ~(1ul << 8);

}

void PG6(){

    while(1){

        GPIOG->ODR |= (1ul << 6);

        for (j = 0; j < TIME_DELAY; j++) {};

        GPIOG->ODR &= ~(1ul << 6);

        for (j = 0; j < TIME_DELAY; j++) {};

    }

}

void PG8(){

    while (1){

        GPIOG->ODR |= (1ul << 8);

        for (j = 0; j < TIME_DELAY; j++) {};

        GPIOG->ODR &= ~(1ul << 8);

        for (j = 0; j < TIME_DELAY; j++) {};

    }

}

int main()

```

```

{

    int i;

    i=0;

    PG7off(); PG6off(); PG8off();

    RCC->AHB1ENR |= 1ul<<6;

    RCC->AHB1ENR |= 1ul;

    RCC->AHB1ENR |= 1ul <<2;

    delay();

    //PG7 out

    GPIOG->MODER = (GPIOG->MODER & ~(1ul<<15)) | 1ul<<14;

    GPIOG->OSPEEDR = (GPIOG->OSPEEDR | (1ul<<15)) | (1ul<<14);

    //PG8 out

    GPIOG->MODER = (GPIOG->MODER & ~(1ul<<17)) | 1ul<<16;

    GPIOG->OSPEEDR = (GPIOG->OSPEEDR | (1ul<<17)) | (1ul<<16);

    //PG6 out

    GPIOG->MODER = (GPIOG->MODER & ~(1ul<<13)) | 1ul<<12;

    GPIOG->OSPEEDR = (GPIOG->OSPEEDR | (1ul<<13)) | (1ul<<12);

    delay();

    //PA0 in

    GPIOA->MODER = (GPIOA->MODER & ~(1ul<<1)) & ~1ul;

    //PG15 in

    GPIOG->MODER = (GPIOG->MODER & ~(1ul<<31)) & ~(1ul<<30);

```

```

//SPEED

//delay 4
    delay();

//tac, 14-th bit, enable SYSCFG
RCC->APB2ENR = RCC->APB2ENR | (1ul<<14);

//setup the EXTI Wakeup event for bit G15 and A0
//unmask
EXTI->IMR |= 1ul;
EXTI->IMR |= (1ul<<15);

//falling trigger for input PG15 /PC13
EXTI->RTSR = EXTI->RTSR & ~(1ul<<15);
EXTI->FTSR = EXTI->FTSR | (1ul<<15);

//rising trigger for input PA0
EXTI->FTSR = EXTI->FTSR & ~1ul;
EXTI->RTSR = EXTI->RTSR | 1ul ;

//pending reg
EXTI->PR |= 1ul;
EXTI->PR |= (1ul<<15);

//config external interrupt
//EXTI15, port G
SYSCFG->EXTICR[3] |= 0x6000 ;

//EXTI0, port A
SYSCFG->EXTICR[0] |= 0x0000;

```

```

        for (i = 0; i < 10; i++) {}

    //NVIC setup
    NVIC_SetPriorityGrouping(4); //2 preempt (bb00)
    //set priority
    NVIC_SetPriority(EXTI15_10_IRQn,4);
    NVIC_SetPriority(EXTIO_IRQn, 2);
    //here the PA0 can preempt PG15
    // Event handler interrupts blinking process of PG7
    NVIC_EnableIRQ(EXTI15_10_IRQn);
    NVIC_EnableIRQ(EXTIO_IRQn);
//PG7();

    for(;;){
        PG7();
    }
}

void EXTI0_IRQHandler(){
    //
    PG8off();
    PG6();
    // reset event
    EXTI->PR |= (1ul);
}

```



```

void EXTI15_10_IRQHandler(){
    PG6off();
    PG8();
    //reset event
    EXTI->PR |= (1ul<<15);
}

```

3 Вывод

По настройке приоритет **EXTI15** больше чем у **EXTI0**, т.е. **EXTI0** может упредить **EXTI15**.

PG7 бесконечно мигается, он только перестает когда одно прерывание пришел.