

# Mikroprozessorpraktikum

## Konstantin Bork & Kean Seng Liew, Gruppe A, HWP8

### 07-01 Sleep Mode

#### A07-01.1

Um zu demonstrieren wie hoch das Einsparpotential des Sleep-Mode ist, soll ein Programm entwickelt werden, welches folgende Eigenschaften realisiert: - In einer Endlosschleife soll die Grüne LED im 3 Sekundentakt basierend auf einer waituSek() Funktion blinken. - In der gleichen Endlosschleife soll die Taste1 im Polling abgefragt werden. Wurde die Taste gedrückt soll die grüne LED Eingeschaltet werden und auf die USART2 der Text "Sleep Mode Start" ausgegeben. Danach soll der Sleep-Mode aktiviert werden. - Die Taste2 soll interruptfähig sein, um den STM32 aus dem Sleep-Mode zuholen. In der ISR der Taste2 soll die grüne LED Ausgeschaltet werden. - Unmittelbar nach der Zeile \_\_WFI() binden Sie bitte eine Ausgabe auf die USART2 mit dem Text "Sleep Mode Ende" ein - Testen Sie das Programm und Beobachten Sie den Stromverbrauch. Stimmen die Beobachtungen mit den Angaben zum Stromverbrauch aus dem Datenblatt überein?

Bitte die USART2 Einbindung vor der while(1) Schleife beachten.

#### main.c

```
#include "main.h"
#include "aufgabe.h"

int main(void)
{
    // Initialisierung des Systems und des Clocksystems
    SystemInit();

    // Start der RTC falls diese noch
    // nicht initialisiert war wird
    // die RTC mit der LSE-Taktquelle aktiviert
    start_RTC();

    // Initialisiere die grüne LED
    init_leds();

    // Initialisiere beide Tasten, bei Taste 2 inkl. Interrupt
    init_taste_1();
    init_taste_2_irq();

    // Initialisiere USART2 für die Ausgabe
    init_usart_2_irq();
```

```

init_nvic();

usart_2_print("\r\nSystem Neustart\r\n");

while(1)
{
    wait_mSek(3000);
    GR_LED_Toggle;

    int taste_1_gedruckt = taste_1_gedrueckt();

    if(taste_1_gedruckt == 0)
    {
        GR_LED_ON;
        usart_2_print("\r\nSleep Mode Start\r\n");
        __WFI();
        usart_2_print("\r\nSleep Mode Ende\r\n");
    }
}

```

### Auszug aufgabe.h

```

// Aufgabe A07-01.1
int taste_1_gedrueckt();

```

### Auszug aufgabe.c

```

// Wir verwenden diese Methode, um zu prüfen, ob Taste 1 gedrückt wurde.
// Dabei registrieren wir, wenn das Eingabebit nicht gesetzt ist, und
// geben dann 0 zurück.
int taste_1_gedrueckt()
{
    uint8_t      Byte = 0;
    Byte = GPIO_ReadInputDataBit(GPIOC, GPIO_Pin_8);

    // Für eine bessere Logik sollte hier mit Bit_RESET verglichen werden.
    // Das haben wir allerdings nicht getestet, daher belassen wir es hier
    // bei unserem verwendeten Code.
    if(Byte == Bit_SET)
    {
        return 1;
    }

    return 0;
}

```

### Auszug interrupts.c

```

void EXTI9_5_IRQHandler(void)
{
    //===== Taster2

```

```
if (EXTI_GetITStatus(EXTI_Line5) == SET)
{
    EXTI_ClearFlag(EXTI_Line5);
    EXTI_ClearITPendingBit(EXTI_Line5);
    // Code
    GR_LED_OFF;
}
}
```

Der von uns gemessene Stromverbrauch beträgt zunächst  $3,7V * 180mA = 666mW$ . Im Sleep Mode haben wir einen Stromverbrauch von  $3,7V * 150mA = 555mW$  gemessen, also 30mA bzw. 111mW weniger. Das ist etwas weniger als die angegebenen 40mA, die gespart werden können.