

Mikroprozessorpraktikum

Konstantin Bork & Kean Seng Liew, Gruppe A, HWP8

05-02 Interrupt intern

A05-02.1

Die main() für diese Aufgabe soll aus einer Initialisierungssequenz mit einer anschließenden leeren Endlosschleife bestehen. In dieser Aufgabe soll die USART2 interruptfähig für den Empfangsmodus gemacht werden. Der Interrupt Handler für die USART2 mit entsprechenden Verarbeitungsfunktionen ausgestattet werden. Der Interrupt Handler der USART2 soll folgende Funktionalität bieten: - Ein empfangenes Zeichen "1" lässt die grüne LED im 1 Sekundentakt blinken - Ein empfangenes Zeichen "4" lässt die grüne LED im 4 Sekundentakt blinken - Ein empfangenes Zeichen "s" schaltet die grüne LED dauerhaft aus

main.c

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

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

    // SysTick initialisieren
    // jede ms erfolgt dann der Aufruf
    // des Handlers fuer den Interrupt SysTick_IRQn
    InitSysTick();

    // Initialisiere die grüne LED
    init_leds();

    // Initialisiere beide Tasten
    init_taste_1_irq();
    init_taste_2_irq();

    init_usart_2_irq();

    init_nvic();

    while(1)
    {

    }
}
```

Auszug aufgabe.c

```
// Initialisiere USART2 für Empfang von Zeichen
void init_usart_2()
{
    GPIO_InitTypeDef GPIO_InitStructure;
    USART_InitTypeDef USART_InitStructure;

    // Taktsystem für die USART2 freigeben
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_USART2, ENABLE);

    // GPIO Port A Taktsystem freigeben
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA, ENABLE);

    // USART2 TX an PA2 mit Alternativfunktion Konfigurieren
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2 | GPIO_Pin_3;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP ;
    GPIO_Init(GPIOA, &GPIO_InitStructure);

    // USART2 TX mit PA2 verbinden
    GPIO_PinAFConfig(GPIOA, GPIO_PinSource2, GPIO_AF_USART2);
    GPIO_PinAFConfig(GPIOA, GPIO_PinSource3, GPIO_AF_USART2);
    // Datenprotokoll der USART einstellen
    USART_InitStructure.USART_BaudRate = 921600;
    USART_InitStructure.USART_WordLength = USART_WordLength_8b;
    USART_InitStructure.USART_StopBits = USART_StopBits_1;
    USART_InitStructure.USART_Parity = USART_Parity_No;
    USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None;
    USART_InitStructure.USART_Mode = USART_Mode_Tx | USART_Mode_Rx;
    USART_Init(USART2, &USART_InitStructure);

    // USART2 freigeben
    USART_Cmd(USART2, ENABLE); // enable USART2
}

// Initialisiere die USART2 inklusive zugehörigem Interrupt
void init_usart_2_irq()
{
    init_usart_2();

    NVIC_InitTypeDef NVIC_InitStructure;

    NVIC_InitStructure.NVIC_IRQChannel = USART2_IRQn;
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
    NVIC_Init(&NVIC_InitStructure);

    // Die Freigabe des zugehörigen Interrupts sieht wie folgt aus:
    USART_ClearITPendingBit(USART2, USART_IT_RXNE);
    USART_ITConfig(USART2, USART_IT_RXNE, ENABLE);
}
```

```
}
```

A05-02.2

In der nun folgenden Aufgabenstellung soll eine beliebige Zeichenkette über ein Terminalprogramm am PC eingegeben werden. Am Ende der Zeichenkette erfolgt der Abschluß der Eingabe mit der Enter-Taste. Das Terminalprogramm sendet für Enter-Taste das Steuerzeichen für Wagenrücklauf. Die Zeichenkette soll Zeichen für Zeichen im Interrupt empfangen werden und in einen Empfangspuffer eingetragen werden. Die Länge der Zeichenkette wird anhand des empfangenen Steuerzeichens für Wagenrücklauf bestimmt. Die Zeichenkette und die Länge sollen vom 3D-SRLD Board zum PC über die USART zurückgeschickt werden. Danach kann eine erneute Eingabe einer Zeichenkette erfolgen.

Auszug interrupts.c

```
void USART2_IRQHandler(void)
{
    char zeichen;

    // RxD - Empfangsinterrupt
    if (USART_GetITStatus(USART2, USART_IT_RXNE) != RESET)
    {
        zeichen = (char)USART_ReceiveData(USART2);
        // Wenn der Wagenrücklauf empfangen wird, beende die aktuelle Zählung,
        // gebe die Eingabe inkl. ihrer Länge aus und führe einen Reset der
        // Werte durch
        if (zeichen == '\r')
        {
            sprintf(usart2_tx_buffer, "%s,%d\n", usart2_rx_buffer, length);
            usart_2_print(usart2_tx_buffer);
            empty_buffers();
            length = 0;
            i = 0;
        } else {
            // Speichere das aktuelle Zeichen im Buffer und zähle es
            usart2_rx_buffer[i] = zeichen;
            length++;
            i = (i + 1) % 50;
        }
    }
}
```

main.c und aufgabe.c werden von der vorherigen Aufgabe übernommen.