Mikroprozessorpraktikum

Konstantin Bork & Kean Seng Liew, Gruppe A, HWP8

06-01 Uhr

A06-01.1

> Entwickeln Sie ein Programm, daß im Sekundentakt (z.B. mit wait_uSek(...)) das Datum und die Uhrzeit der RTC ausliest und über die USART2 ausgibt.

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();

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

    init_usart_2_irq();

    RTC_TimeTypeDef  RTC_Time_Aktuell;      //  Zeit
    RTC_DateTypeDef  RTC_Date_Aktuell;      //  Datum

    char data[50] = {0};

    while(1)
    {

        // Datum aus der RTC in das Struct laden und ausgeben
        RTC_GetDate(RTC_Format_BIN, &RTC_Date_Aktuell);
        sprintf(data,"\r\n%.2d-%.2d-%.2d",RTC_Date_Aktuell.RTC_Date,
RTC_Date_Aktuell.RTC_Month, RTC_Date_Aktuell.RTC_Year);
        usart_2_print(data);

        // Zeit aus der RTC in das Struct laden und ausgeben
        RTC_GetTime(RTC_Format_BIN, &RTC_Time_Aktuell);
        sprintf(data,"\r\n%.2d:%.2d:%.2d",RTC_Time_Aktuell.RTC_Hours,
RTC_Time_Aktuell.RTC_Minutes, RTC_Time_Aktuell.RTC_Seconds);
```

```
        usart_2_print(data);

        wait_mSek(1000);
    }
}
```

A06-01.2

> Erweitern Sie die Lösung aus A06-01.1 um die Funktionalität die Uhrzeit und das Datum der RTC über das Terminalprogramm neu zu stellen.

Die Eingabe erfolgt als zusammenhängender String im Format "YYMMDDHHMMSS", also ohne jegliche Trennzeichen. Trennzeichen können leicht akzeptiert werden, indem die Indizes bei der Abfrage des Buffers angepasst werden. Außerdem kann mittels Anpassung der Indizes auch der Wochentag eingestellt werden.

Auszug aufgabe.h

```
// Aufgabe A06-01.2
void zeit_einstellen();
```

Auszug aufgabe.c

```
void zeit_einstellen()
{
    RTC_DateTypeDef RTC_Date_Struct;
    RTC_TimeTypeDef RTC_Time_Struct;

    RTC_DateStructInit(&RTC_Date_Struct);
    RTC_Date_Struct.RTC_Year = (usart2_rx_buffer[0] - '0') * 16 +
(usart2_rx_buffer[1] - '0');
    RTC_Date_Struct.RTC_Month = (usart2_rx_buffer[2] - '0') * 16 +
(usart2_rx_buffer[3] - '0');
    RTC_Date_Struct.RTC_Date = (usart2_rx_buffer[4] - '0') * 16 +
(usart2_rx_buffer[5] - '0');
    RTC_SetDate(RTC_Format_BCD, &RTC_Date_Struct);

    RTC_TimeStructInit(&RTC_Time_Struct);
    RTC_Time_Struct.RTC_Hours = (usart2_rx_buffer[6] - '0') * 16 +
(usart2_rx_buffer[7] - '0');
    RTC_Time_Struct.RTC_Minutes = (usart2_rx_buffer[8] - '0') * 16 +
(usart2_rx_buffer[9] - '0');
    RTC_Time_Struct.RTC_Seconds = (usart2_rx_buffer[10] - '0') * 16 +
(usart2_rx_buffer[11] - '0');
    RTC_SetTime(RTC_Format_BCD, &RTC_Time_Struct);
}
```

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 gelesen wird, stelle die Zeit mit der Eingabe ein
        if (zeichen == '\r')
        {
            zeit_einstellen();
        } else {
            // Ansonsten speichere die Eingabe im Buffer
            usart2_rx_buffer[i] = zeichen;
            i = (i + 1) % 50;
        }
    }
}
```