

Mikroprozessorpraktikum

Konstantin Bork, Kean Seng Liew, & Oliver Stein, Gruppe A, HWP8

02-02 SysTick

A02-02.1

Erweitern Sie den SysTick_Handler() in der Art, daß die grüne LED wie folgt angesteuert wird. - Die grüne LED soll endlos im Wechsel für 0,5 Sek angeschaltet und danach für 3 Sek ausgeschaltet werden.

Auszug interrupts.c

```
void SysTick_Handler(void)
{
    static unsigned long stc_led = 0;
    static unsigned long stc0 = 0;
    static unsigned long stc1 = 0;
    static unsigned long stc2 = 0;
    stc_led++;
    stc0++;
    stc1++;
    stc2++;

    //=====
    // DW1000 Timeout
    systickcounter += 1;
    if ( stc0 >= 20 )
    {
        //uwbranging_tick();
        stc0 = 0;
    }

    //=====
    // Co0S_SysTick_Handler alle 10ms in Co0s arch.c aufrufen
    // nur Einkommentieren wenn Co0S genutzt wird
    if ( stc1 >= 10 )
    {
        // Co0S_SysTick_Handler();
        stc1 = 0;
    }

    //=====
    // CC3100 alle 50ms Sockets aktualisieren
    if (stc2 >= 5)
    {
```

```

        stc2 = 0;
        if ( (IS_CONNECTED(WiFi_Status)) && (IS_IP_ACQUIRED(WiFi_Status)) &&
(!Stop_CC3100_select) && (!mqtt_run) )
        {
            CC3100_select(); // nur aktiv wenn mit AP verbunden
        }
        else
        {
            _SlNonOsMainLoopTask();
        }
    }

//=====
// SD-Card
sd_card_SysTick_Handler();

//=====
// MQTT
MQTT_SysTickHandler();

//=====
// LED laut Aufgabe schalten
// nach 500mS schalten wir die LED aus
if ( stc_led >= 500 )
{
    LED_GR_OFF;
}

// nach weiteren 3000mS schalten wir sie wieder an
// und setzen den Zähler wieder auf 0
if ( stc_led >= 3500 )
{
    LED_GR_ON;
    stc_led = 0;
}
}

```

A02-02.2

In einigen Fällen wird ein blockierendes Warten benötigt. In dem folgenden Codebeispiel wird eine Variable timer auf einen Wert gesetzt und daraufhin mit einem while(timer) die weitere Programmabarbeitung unterbrochen bis timer den Wert 0 hat. Die Variable timer soll im SysTick_Handler() bis auf 0 heruntergezählt werden.

Erweitern Sie den SysTick_Handler() in der Form, das mittels der Variablen timer Zeitverzögerungen realisiert werden können. Dabei soll der timer=1 für eine Zeitverzögerung von 100mSek stehen. Für eine Zeitverzögerung von einer Sekunde müsste timer=10 gesetzt werden.

Auszug interrupts.c

```
int32_t timer = 0;
```

```

void SysTick_Handler(void)
{
    static unsigned long stc_led = 0;
    static unsigned long stc0 = 0;
    static unsigned long stc1 = 0;
    static unsigned long stc2 = 0;
    static unsigned long stc3 = 0; // ein neuer Zähler für den timer
    stc_led++;
    stc0++;
    stc1++;
    stc2++;
    stc3++; // jede mS setzen wir den timer Zähler um 1 hoch

    //=====
    // DW1000 Timeout
    systickcounter += 1;
    if ( stc0 >= 20 )
    {
        //uwbranging_tick();
        stc0 = 0;
    }

    //=====
    // Co0S_SysTick_Handler alle 10ms in Co0s arch.c aufrufen
    // nur Einkommentieren wenn Co0S genutzt wird
    if ( stc1 >= 10 )
    {
        // Co0S_SysTick_Handler();
        stc1 = 0;
    }

    //=====
    // CC3100 alle 50ms Sockets aktualisieren
    if (stc2 >= 5)
    {
        stc2 = 0;
        if ( (IS_CONNECTED(WiFi_Status)) && (IS_IP_ACQUIRED(WiFi_Status)) &&
(!Stop_CC3100_select) && (!mqtt_run) )
        {
            CC3100_select(); // nur aktiv wenn mit AP verbunden
        }
        else
        {
            _SlNon0sMainLoopTask();
        }
    }

    //=====
    // SD-Card
    sd_card_SysTick_Handler();

    //=====
    // MQTT
    MQTT_SysTickHandler();

    //=====
    // wenn 100ms vergangen sind, dekrementieren wir den timer

```

```
// und setzen den Zähler wieder auf 0
if(stc3 >= 100)
{
    timer--;
    stc3 = 0;
}
}
```