

# Mikroprozessorpraktikum

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

**04-02 WWDG**

**A04-02.1**

Das folgende Beispielprogramm nehmen Sie bitte als Ausgangspunkt für die Bearbeitung der Aufgabenstellung.

```
#include "main.h"

int main(void){

    unsigned char value_watchdog_counter = 0x7f;
    unsigned char window_value = 0x50;
    unsigned char window_value_refresh = 0x50;
    unsigned char cnt_i = 0;
    unsigned char cnt_j = 0;

    SystemInit();

    init_usart_2();

    sprintf(usart2_tx_buffer, "\r\nNeustart\r\n");
    usart_2_print(usart2_tx_buffer);

    RCC_APB1PeriphClockCmd(RCC_APB1Periph_WWDG, ENABLE);

    WWDG_SetPrescaler(WWDG_Prescaler_8);
    WWDG_SetWindowValue(window_value);
    WWDG_Enable(value_watchdog_counter);

    cnt_i = (unsigned char) (value_watchdog_counter + 1);

    while(1){

        cnt_j = (unsigned char) ((WWDG->CR) & 0x7F) ;

        if (cnt_j < cnt_i ) {

            sprintf(usart2_tx_buffer, "i = %u\r\n", cnt_j);
            usart_2_print(usart2_tx_buffer);

            cnt_i = cnt_j;

            if (cnt_i == window_value_refresh ) {

                WWDG_SetCounter(value_watchdog_counter);

                sprintf(usart2_tx_buffer, "##### neu geladen\r\n");
```

```

        usart_2_print(usart2_tx_buffer);

        cnt_i = (unsigned char) (value_watchdog_counter + 1);
    }
}
}
}
}

```

Testen Sie die Funktion des WWDG durch Variation der Variablen `value_watchdog_counter`, `window_value`, `window_value_refresh`. Setzen Sie dazu einen Breakpoint auf die Zeile 18 des obigen Quellcodes. Im Breakpoint können Sie die Variablen verändern ohne das Programm jeweils neu zu Flashen zu müssen. Analysieren Sie für die folgenden Fälle durch die Variation der Variablen das Verhalten des WWDG.

- Bei Zählerwerten von T6:0 in WWDG\_CR größer als W6:0 in WWDG\_CFR ist eine Aktualisierung des Abwärtszählers (WWDG\_CR) verboten (Refresh no allowed) und würde zum Reset führen.
- Bei Zählerwerten von T6:0 in WWDG\_CR kleiner als W6:0 in WWDG\_CFR ist eine Aktualisierung des Abwärtszählers (WWDG\_CR) erlaubt (Refresh window).
- Bei Zählwerten des Abwärtszählers (WWDG\_CR) kleiner 0x40 wird ein RESET ausgelöst.

Dokumentieren Sie die eingestellten Parameter und das Verhalten. Diskutieren Sie mögliche Einsatzszenarien für einen WWDG.

Unser erstes Szenario verwendet folgende Werte und entspricht dem in der Aufgabenstellung zuerst genannten Fall: - `value_watchdog_counter = 0x7f` - `window_value = 0x50` - `window_value_refresh = 0x50`

Das Programm wird bei `i=80` neu geladen.

Unser zweites Szenario entspricht dem zweiten in der Aufgabenstellung beschriebenen Fall und verwendet diese Werte: - `value_watchdog_counter = 0x50` - `window_value = 0x7f` - `window_value_refresh = 0x50`

Zunächst wird das Programm dreimal nach `i=80` neu geladen, ehe bei `i=64` ein Neustart stattfindet.

Im dritten Szenario verwenden wir diese Werte: - `value_watchdog_counter = 0x3c` - `window_value = 0x50` - `window_value_refresh = 0x50`

Das Programm wird durchgängig neu gestartet.

Im letzten Szenario verwenden wir folgende Werte: - `value_watchdog_counter = 0x48` - `window_value = 0x50` - `window_value_refresh = 0x50`

Der Schleifenzähler startet bei `i=72`. Das Programm wird nie neu geladen, allerdings erfolgt bei

i=64 ein Neustart.

Nach unseren Tests kommen wir zu dem Schluss, dass der WWDG am relevantesten für sicherheitskritische Systeme ist, in denen die Firmware möglichst immer in einem sicheren Zustand laufen soll. Bei Problemen in der Firmware sollten diese schnellstmöglich erkannt werden und die Firmware dann wieder in einen sicheren Zustand wechseln.