

Timersystem mit und ohne Interrupt

1. 1-Sekunden Timer mit Overflow-Flag (TOF) Polling

Realisieren Sie einen 1-Sekunden Timer (LED am Port F:1 blinken lassen) indem zwischen jedem Toggeln der LED eine Sekunde gewartet wird. Die 1000 ms Verzögerung soll durch Abfrage (Polling) des Timer Overflow Flags (TOF) von TPM1 realisiert werden.

```

/*****
*** Uebung 6.1
*** 1-Sekunden Timer mit Overflow-Flag (TOF) Polling
*****/
#include "platform.h" /* include peripheral declarations */

/**
 * Application entry point.
 */
void main(void)
{
    // Realisieren Sie einen 1-Sekunden Timer (LED am Port F:1 blinken lassen)
    // indem zwischen jedem Toggeln der LED eine Sekunde gewartet wird.
    // Die 1000 ms Verzögerung soll durch Abfrage (Polling) des
    // Timer Overflow Flags (TOF) von TPM1 realisiert werden.

    // init Port
    PTFDD_PTFDD1 = 1; // PTD1 als Ausgang
    PTFDD_PTFDD2 = 1; // PTD2 als Ausgang
    PTFD = 2; // init LED

    // init TPM
    TPM1SC_PS = 4; // Prescaler: 16
    // (16.3.1 TPM Status and Control Register (TPMxSC), S.281)
    TPM1MOD = 62499; // 1'000'000 : 16 = 62500 - 1
    TPM1SC_TOF = 0; // TOF-Flag zurücksetzen
    TPM1SC_CLKSx = 2; // Bin: 0x10 - Use Fix Syst.Clk - 1MHz (1 micros)
    // --> Timer starten

    for(;;)
    {
        if (TPM1SC_TOF) {
            TPM1SC_TOF = 0; // TOF Flag reset
            PTFD_PTFD1 = !PTFD_PTFD1; // LED R FL
            PTFD_PTFD2 = !PTFD_PTFD2; // LED R FR
        }

        __RESET_WATCHDOG(); /* feeds the dog */
    } /* loop forever */
} // main

```

2. 1-Sekunden Timer mit Overflow-Flag (TOF) Interrupt
Realisieren Sie einen 1-Sekunden Timer (LED am Port F:1 blinken lassen) indem Sie jede Sekunde einen Timer-Overflow Interrupt von TPM1 auslösen lassen.

main.c

```

/*****
*** Uebung 6.2
*** Sekunden Timer mit Overflow-Flag (TOF) Interrupt
*****/
#include "platform.h" /* include peripheral declarations */

/*
 * Interrupt Funktion: LED blinken
 * Interrupt Funktion in Project.prm definieren !!!
 *   >>> Project Settings > Linker Files > Project.prm
 *   >>> VECTOR ADDRESS 0xFFE0 toggleLED          // TPM1 overflow
 */
interrupt void toggleLED(void)
{ // TOF muss nicht abgefragt werden, sonst hätte es ja kein Interrupt gegeben
  TPM1SC_TOF = 0;          // TOF-Flag reset
  PTFD_PTFD1 = !PTFD_PTFD1; // LED R FL
  PTFD_PTFD2 = !PTFD_PTFD2; // LED R FR
}

/**
 * Application entry point.
 */
void main(void)
{
  // Realisieren Sie einen 1-Sekunden Timer (LED am Port F:1 blinken lassen) indem
  // Sie jede Sekunde einen Timer-Overflow Interrupt von TPM1 auslösen lassen.

  // init Port
  PTFDD_PTFDD1 = 1; // PTD1 als Ausgang
  PTFDD_PTFDD2 = 1; // PTD2 als Ausgang
  PTFD = 2;        // init LED

  // init TPM
  TPM1SC_PS = 4;      // Prescaler: 16
                      // (16.3.1 TPM Status and Control Register (TPMxSC), S.281)
  TPM1MOD = 62499;   // 1'000'000 : 16 = 62500 - 1
  TPM1SC_TOF = 0;    // TOF-Flag zurücksetzen
  TPM1SC_TOIE = 1;   // Enable Interrupt
  TPM1SC_CLKSx = 2;  // Bin: 0x10 - Use Fix Syst.Clk - 1MHz (1 micros)
                      // --> Timer starten

  EnableInterrupts; // Enable Global Interrupt

  for(;;)
  {
    // do nothing here - our interrupt does the work...

    RESET_WATCHDOG(); /* feeds the dog */
  } /* loop forever */
} // main

```

project.prm

```

[...]  

VECTOR ADDRESS 0xFFE0 toggleLED          // TPM1 overflow  

[...]
```