

Name	Matrikel	Anmerkungen
Datum	Raster (z.B. Mo-2x)	Testat/Datum

Legende: V: Vorbereitung, D: Durchführung, P: Protokoll/Dokumentation, T: Testat

## Praktikum 2

*Lernziele: Verarbeitung von Ereignissen als Interrupts oder direkt in der Hardware (SoC)*

Für die Bearbeitung der Aufgaben ist ein Termin angesetzt.

- Starten Sie **PSoC-Creator** und laden Sie das Projekt Termin 2: MPS\_W22\_Prakt\_2.
  - Betrachten Sie das neue Hardware-Design in `TopDesign.cysch`.  
(*Nicht erschrecken*: enthält jetzt alle Funktionen, die mit dem Traffic-Light-Shield und zusammen mit der UART möglich sind. Nur ein Teil wird Termin 3 benötigt.)
  - Starten Sie **TeraTerm** und verbinden Sie mit  
`COM<xy>` (Datei > Neue Verbindung > Seriell). Beachten Sie die UART-Einstellungen in `TopDesign.cysch`.
  - Betrachten Sie die HW-Verbindung von `FreqDiv_Clk (div)` mit `Pin_E_Y`. Welche Funktion hat `FreqDiv_Clk (div)` und wie ist `Pin_E_Y` konfiguriert?
  - Welche Funktion erfüllen die Komponenten `isr_UART_RX`, `isr_Clk`, und `isr_CWEW` ?
  - Was bewirkt `CySoftwareReset()` ?

Beantworten Sie als *Vorbereitung* alle Fragen und dokumentieren Sie später im Protokoll.

- Betrachten Sie `Pin_E_Y`
  - Muss für die Funktion der LED `E_Y` etwas im Code konfiguriert und/oder initialisiert werden?
  - Mit welcher Frequenz blinkt die LED?
  - Wird das Blinken durch das Programm *main* in irgend einer Weise beeinflusst? (Testen Sie dies, z.B. durch eine Delay im Programm)
- UART-Interrupt
  - Welches Ereignis löst den UART-Interrupt aus? (*Hinweis*: UART-Komponente öffnen, Reiter *advanced*)
  - Wo im Code ist die ISR (bereits vollständig) implementiert, und wo wird sie registriert?
  - Das Menu-Template reagiert jetzt nur auf den Interrupt, erklären Sie den Code
  - Erweitern Sie später das Menu nach Bedarf für die folgenden Aufgaben oder für Ihre Tests.
- Button-Ereignis aus Interrupt `MyIsrCWEW`
  - Wo im Code ist die ISR (bereits vollständig) implementiert, und wo wird sie registriert?
  - Toggeln Sie die LED `Pin_E_CW` wenn der Button gedrückt wird. Erweitern Sie dazu den vorgegebenen Code.
  - Freiwillig*: geben Sie ein Tonsignal auf Ihrem Terminal aus, wenn die Taste gedrückt wird. *Hinweis*: ASCII-Tabelle, Steuerzeichen!

5. Timer-Ereignis `isr_Clk`
    - a. Wie häufig wird dieser Interrupt ausgelöst?
    - b. Wie könnte man die vergangene Zeit seit dem Start des Programms messen? Zeigen Sie diese über das Menu an.
    - c. Setzen Sie in der ISR alle Sekunden ein Flag, welches Sie in Ihrem main-Programm auswerten. Zeigen Sie die seit Programmstart vergangene Zeit kontinuierlich in *ein und derselben* Terminalzeile an. *Hinweis: ASCII-Steuerzeichen!*
    - d. *Freiwillig*: zeigen Sie die Zeit aus 5.c. in Minuten und Sekunden an (mm:ss).
    - e. Wie oft - schätzen Sie – wird die Endlosschleife in main pro Sekunde durchlaufen? Messen Sie! (Dazu ggf. c. und d. auskommentieren)
  6. Kommentieren Sie – gegebenenfalls nach dem Praktikum zu Hause – Ihren Code. Archivieren Sie Ihr Projekt zu Ihrem späteren Gebrauch.
  7. Schreiben Sie ein kurzes Protokoll und fassen Sie Ihre Erkenntnisse zusammen und fügen Sie die jeweiligen Codeabschnitte hinzu. Laden Sie Ihren Code<sup>+)</sup> als \*.zip und Ihr Protokoll als \*.pdf in Moodle hoch bis **maximal** 1 Woche nach dem Termin.
- 

Bereiten Sie sich auf den Praktikumstermin 2 so vor, dass die Zeit zur Durchführung während des Termins sicher ausreicht. (*Lesen Sie bitte die Aufgabenstellung und Begleitmaterial vor dem Praktikumstermin.*)

*Die Themen und Erkenntnisse aus diesem Praktikum werden im Lauf des Semesters weiter benötigt! Arbeiten und dokumentieren Sie Ihre Ergebnisse sorgfältig!*

Die Teilaufgaben sind schriftlich zu dokumentieren. Laden Sie Ihr Protokoll wie in 7. beschrieben zu Termins 2 hoch.

Viel Spaß und Erfolg

<sup>+)</sup> im \*.zip bitte **nur** den Ordner mit \*.c, \*.h und gegebenenfalls Projektdatei.