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

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

Praktikum 4

Lernziele: PWM (Pulsweitenmodulation) zur Regelung der Helligkeit einer LED. Zeitgesteuerete Ampelsteuerung mit Implementierung eines Zustandsautomaten (finite state machine (FSM))

Für die Bearbeitung der Aufgaben ist ein Termin angesetzt. **Eine gute Vorbereitung ist zwingend erforderlich!**

1. Es soll eine Ampelsteuerung [1] realisiert werden, bei der in Nord-Süd-Richtung und Ost-West-Richtung der Verkehr abwechselnd fließen soll:
   Die Übergänge Rot – Rot-Gelb sowie Grün – Gelb sollen jeweils 1 Sekunde dauern.
   Grün erhält die Richtung West(-Ost) **2** Sekunden, die Richtung Nord-Süd **3** Sekunden.
   Beantworten Sie zuerst die folgenden Fragen.
   a. Wieviele **Ausgaben** hat die FSM? Also LED's die unabhängig voneinander geschaltet werden? (*Hinweis:* fassen Sie die jeweiligen Farben bei Nord-Süd zusammen, wie bei einer richtigen Kreuzung). Fußgänger-Ampeln nicht vergessen!
   b. Wieviele **Zustände** hat die FSM bei den vorgegeben Zeitintervallen?
   c. Zeichnen Sie zuallererst ein **Status-Diagramm** *(remember TGI !)* , am besten mit Bleistift und Papier zwecks schneller Korrektur (**wird *als Vorbereitung verlangt***).

2. *Planen* Sie die effiziente Implementierung! Spart viel Code und mögliche Fehler! (**wird *als Vorbereitung verlangt***)
   a. Zur Erleichterung und Übersicht ersetzen Sie die recht langen Funktionsaufrufe für das Setzen der einzelnen Pins durch kurze, prägnante Macros in einer eigenen Header-Datei, z.B. `Pin_W_R_Write(x)` durch `W_R(x)` in `HAL.h`. (*HAL bedeutet Hardware Abstraction Layer*).
   b. Nutzen Sie für den 1-Sekunden Zeit-Trigger die ISR aus Termin 3. (*Delays, Warteschleifen etc. werden nicht akzeptiert!*)
   c. Planen Sie die FSM sorgfältig:
      i. Wie können der **Zustand** und die Übergänge im Programm dargestellt werden? (Wieviel gibt es?)
      ii. Wie können **Ausgaben** (LEDs) dargestellt werden? Siehe [2].
   d. Auf dem PC ohne PSoC-Board können sie mit *Code::Blocks* o.ä. entsprechenden Code testen!

3. Starten Sie ***PSoC-Creator*** und laden Sie das Projekt Termin 4: MPS21_Prakt_4.
   a. Betrachten Sie das neue Hardware-Design in `TopDesign.cysch`.
   b. Starten Sie ***TeraTerm*** und verbinden Sie mit `COM<xy>` (Datei > Neue Verbindung >Seriell). Beachten Sie die UART-Einstellungen in `TopDesign.cysch`.
   c. Implementieren Sie Ihre Ampelsteuerung fertig, testen Sie und führen Sie vor!

4. **PWM**: betrachten Sie `TopDesign.cysch`.
   a. Auf welchen Wert muss Periode eingestellt werden, um eine Wiederholung von etwa 10ms zu erreichen?
   b. Betrachten Sie die API zur PWM-Komponente. Mit welchem Wert kann die Helligkeit der gelben LED (East) verändert werden?
   c. Verändern Sie die Helligkeit der LED über die Menusteuerung, z.B. '+' und '-'

5. Erweiterungen
   a. Fügen Sie die Fußgänger-Ampeln in Ihre State Machine ein.
   b. Fußgängeranforderung Pin_E_CW über ISR aus Termin 3: Bei Drücken des Buttons (D11 auf dem Board) sollen die Fußgänger schneller weiß bekommen oder länger weiß erhalten.
   c. Die rote LED an einem zweiten PWM-Kanal anschließen (gleiche Komponenente, LED-Pin modifizieren)

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[+)] als *.zip und Ihr Protokoll als *.pdf in Moodle hoch bis **maximal** 1 Woche nach dem Termin.

Bereiten Sie sich auf den Praktikumstermin 4 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 4 hoch.

Viel Spaß und Erfolg

*+) im *.zip bitte **nur** den Ordner mit *.c, *.h und gegebenenfalls Projektdatei*.

[1] Traffic Light Shield https://lernen.h-da.de/mod/resource/view.php?id=608794
[2] es gibt verschiedene Möglichkeiten der Implementierung, z.B.
die Ausgaben in Form eines Feldes zu definieren und dann entsprechend nutzen

```
const static uint8_t lightPattern[][8] = {
      {0, 1, 0,  0, 1, 0,  0, 0}, // state 0
      {0, 0, 0,  0, 0, 0,  0, 0}, // state 1
      // ...
};
```