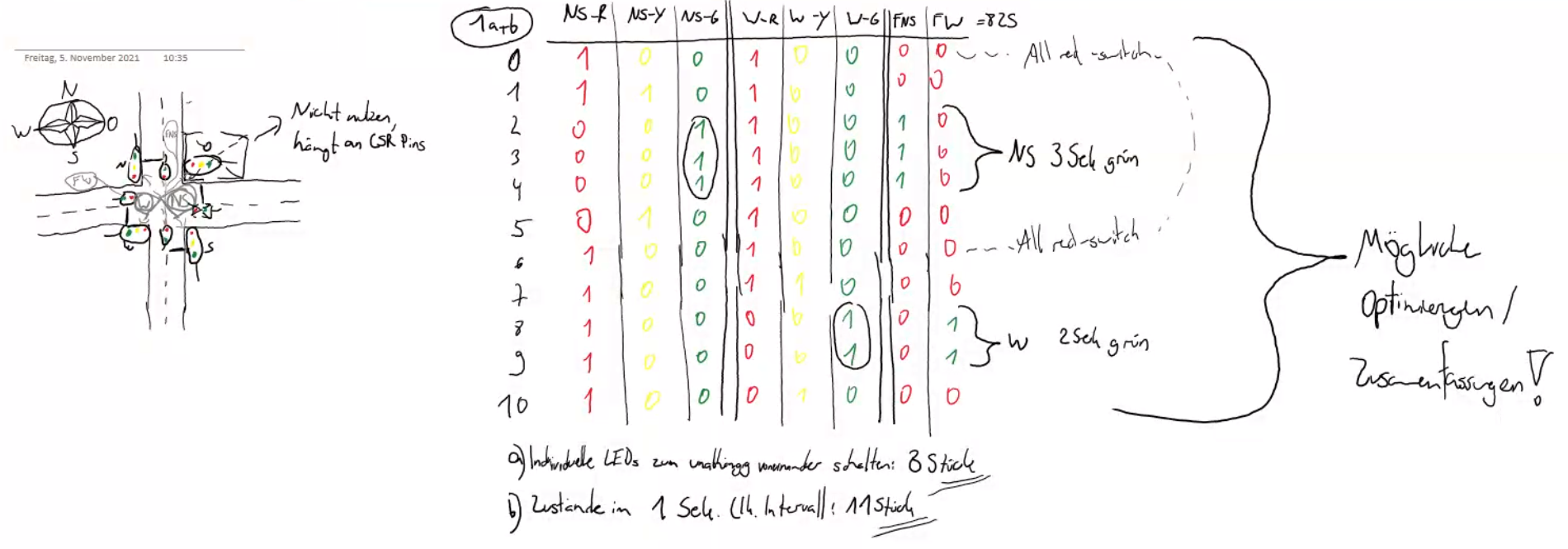
Praktikum 4

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

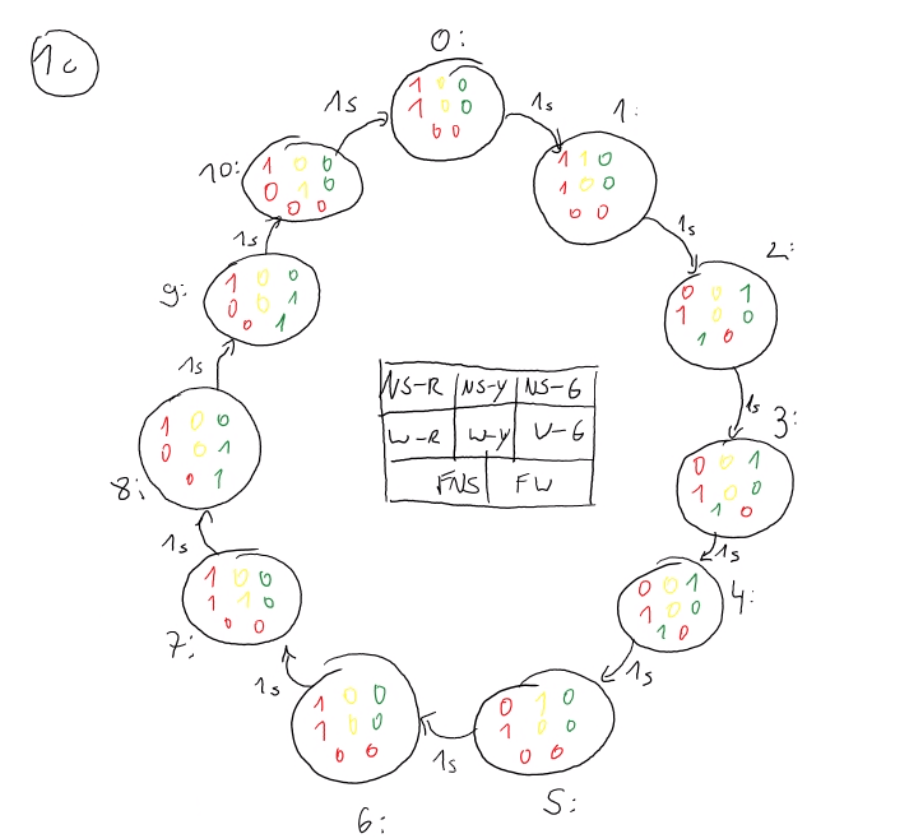
**1. Es soll eine Ampelsteuerung realisiert werden, bei der in Nord-Süd-Richtung und Ost-West-Richtung der Verkehr abwechselnd fließen soll:**

**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.**

**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?**

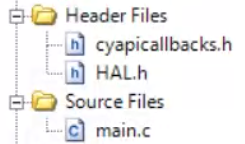
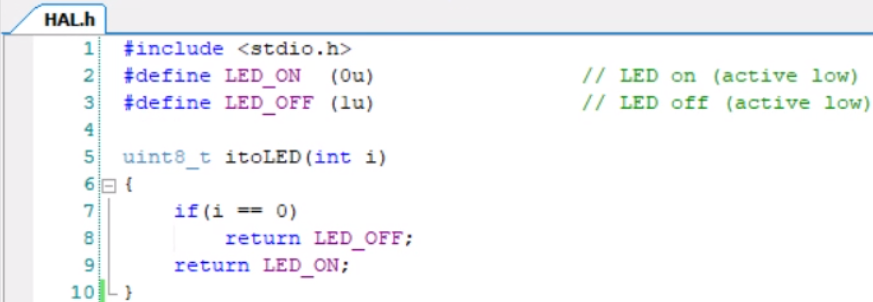
**c. Zeichnen Sie zuallererst ein Status-Diagramm (remember TGI !) , am besten mit Bleistift und Papier zwecks schneller Korrektur.**



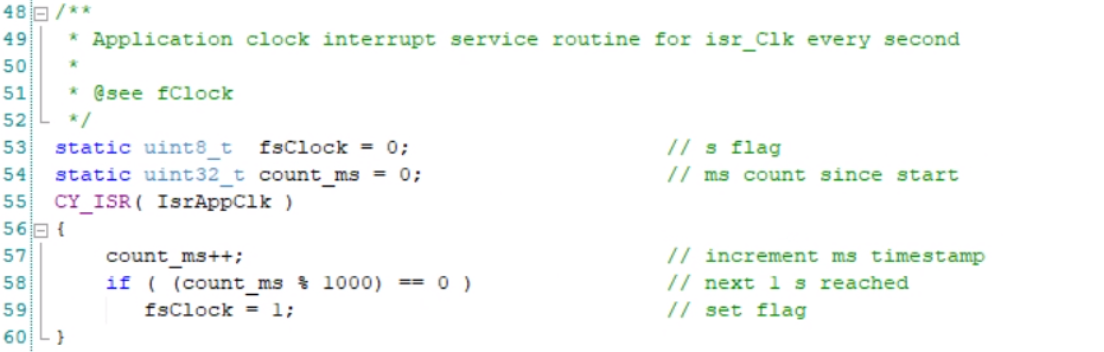
**2. Planen Sie die effiziente Implementierung! Spart viel Code und mögliche Fehler!**

**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.**

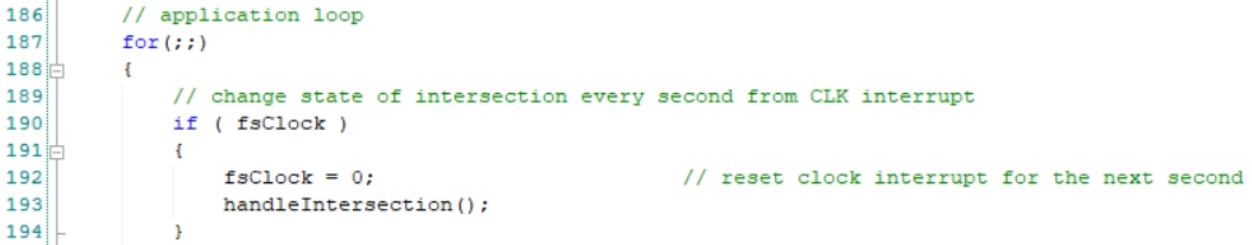
**(HAL bedeutet Hardware Abstraction Layer).**

**b. Nutzen Sie für den 1-Sekunden Zeit-Trigger die ISR aus Termin 3.**

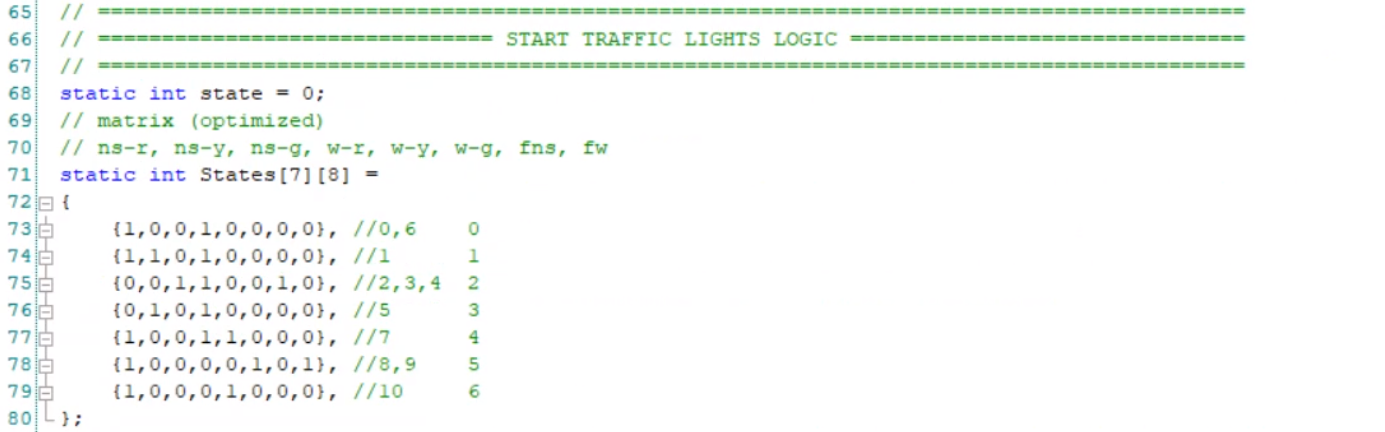


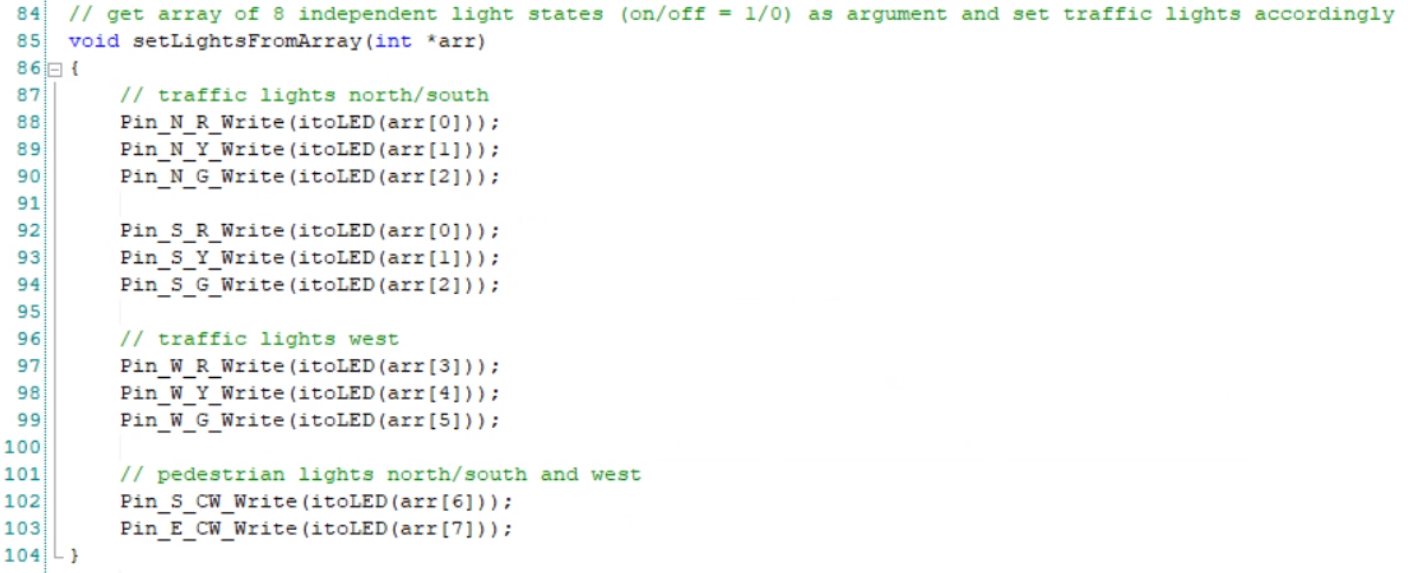


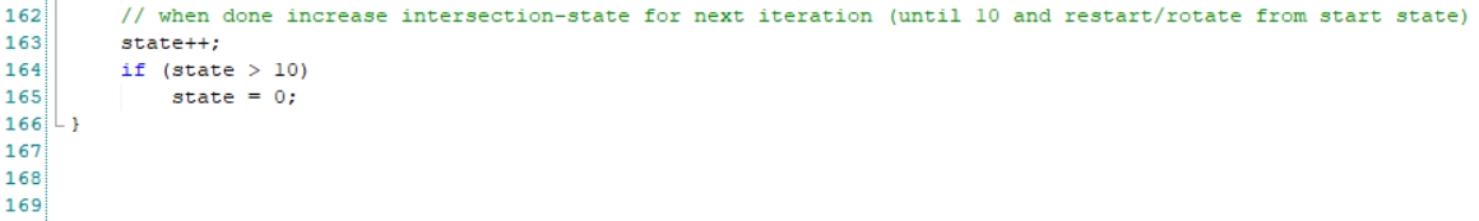
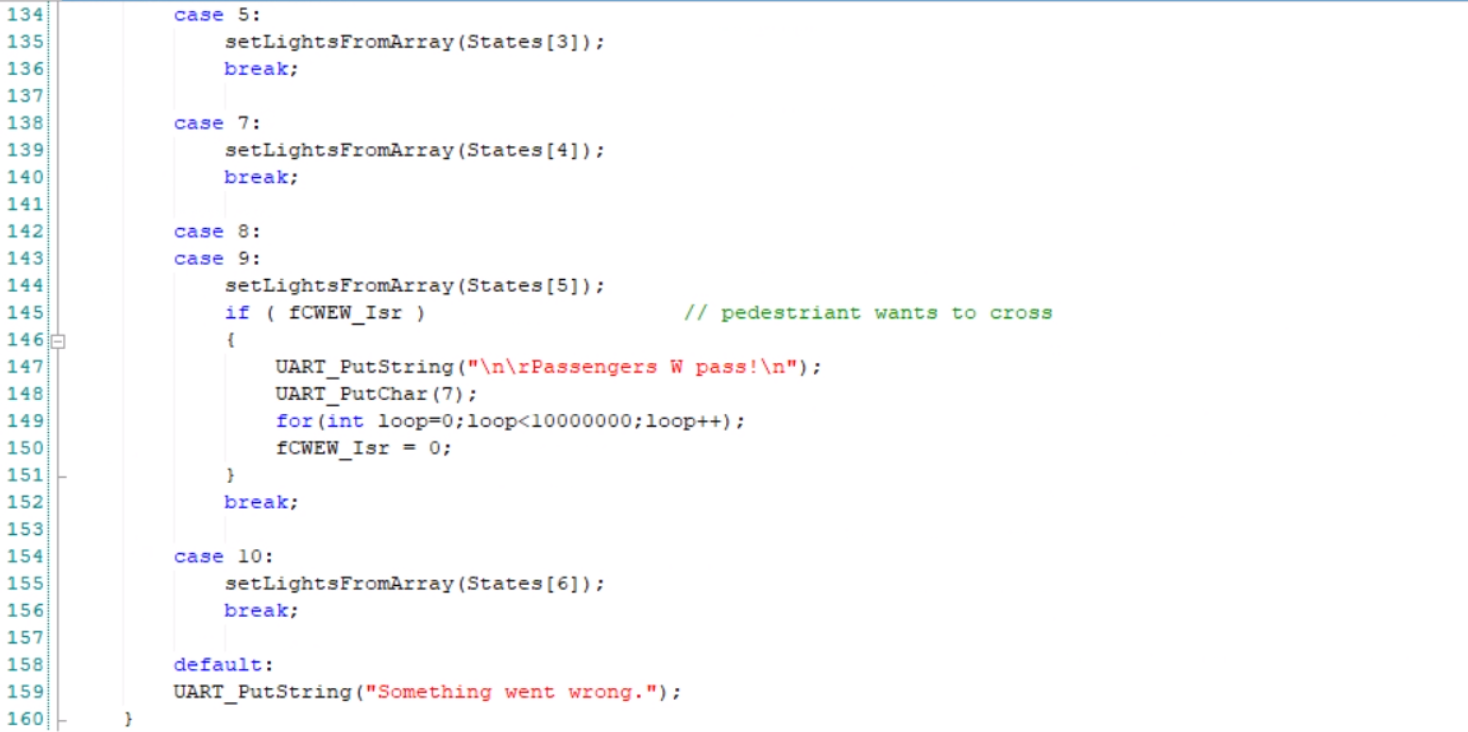
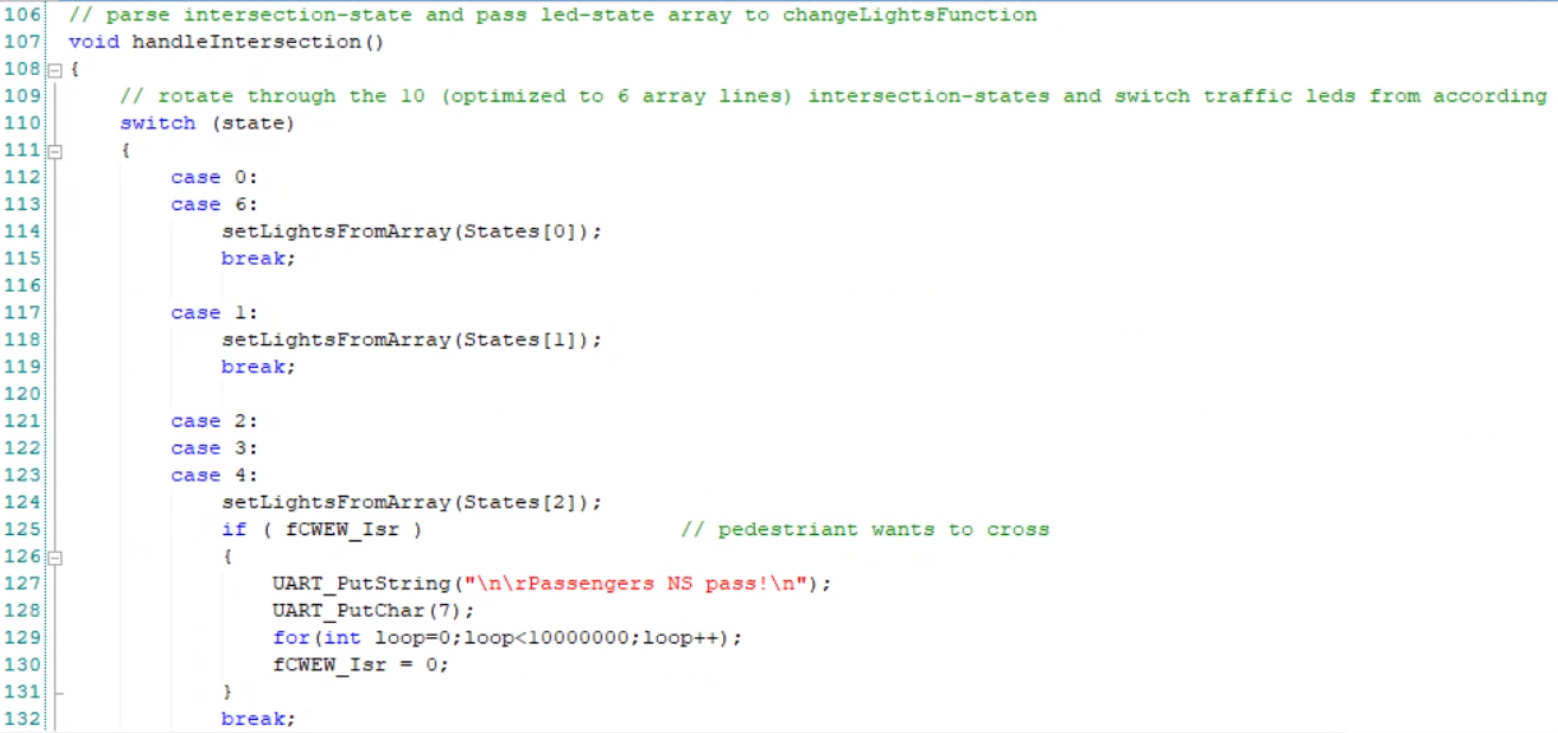


**c. Planen Sie die FSM sorgfältig: Wie können Zustand, Übergänge und Ausgaben dargestellt werden?**

**d. Auf dem PC ohne PSoC-Board können sie mit Code::Blocks o.ä. entsprechenden Code testen!**

****

****

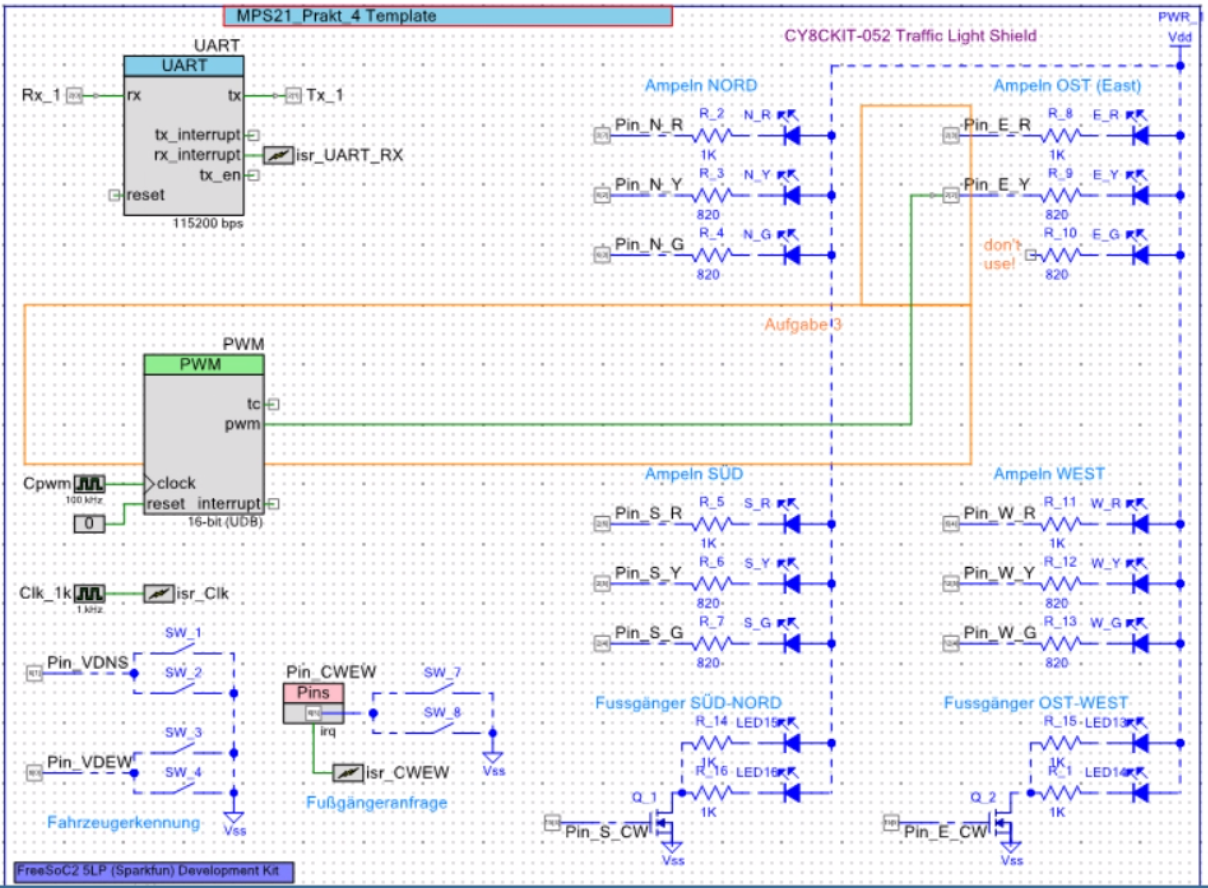
****

**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 UARTEinstellungen in TopDesign.cysch.**

**c. Implementieren Sie Ihre Ampelsteuerung fertig, testen Sie und führen Sie vor!**

****

**Einstellungen > Serieller Port:**

**- speed (baud): 115200  
- data: 8  
- parity: none  
- stop: 1**

**Einstellungen > Terminal:**

**- Übertragen: CR  
- Absenden: CR+LF**

**4. Pulsweitenmodulation (PWM): betrachten Sie TopDesign.cysch.**

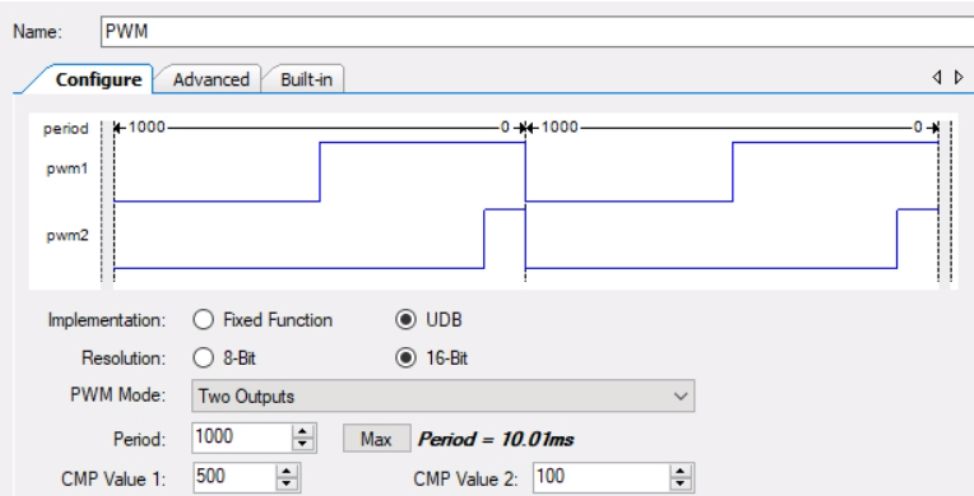
**a. Auf welchen Wert muss Periode eingestellt werden, um eine Wiederholung von etwa 10ms zu erreichen?**

**Periodeneinstellung an PWM-Komponente: 1000**

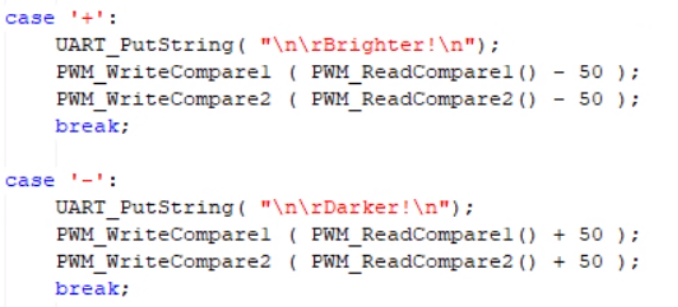
****

**b. Betrachten Sie die API zur PWM-Komponente. Mit welchem Wert kann die Helligkeit der gelben LED (East) verändert werden?**

**Helligkeit kann durch Veränderung der CMP-Werte in PWM verändert werden:**

****

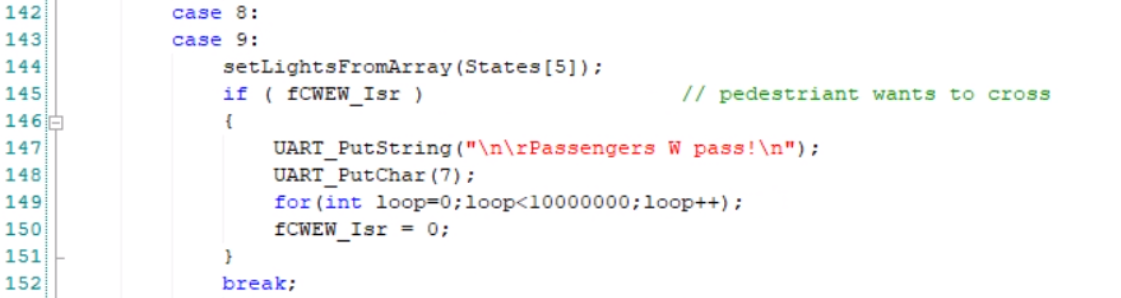
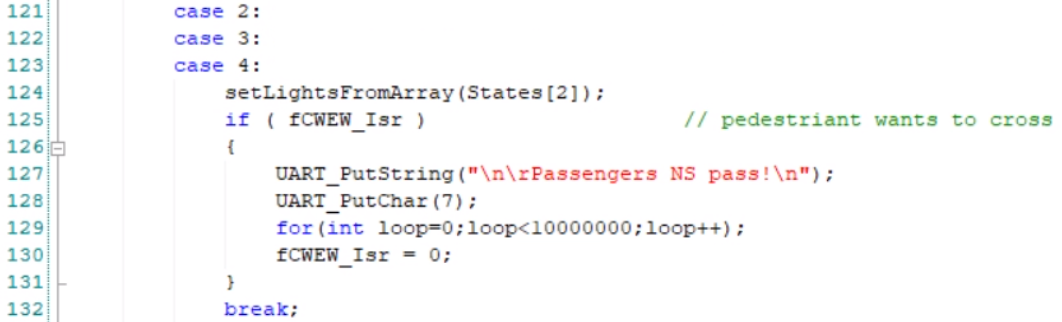
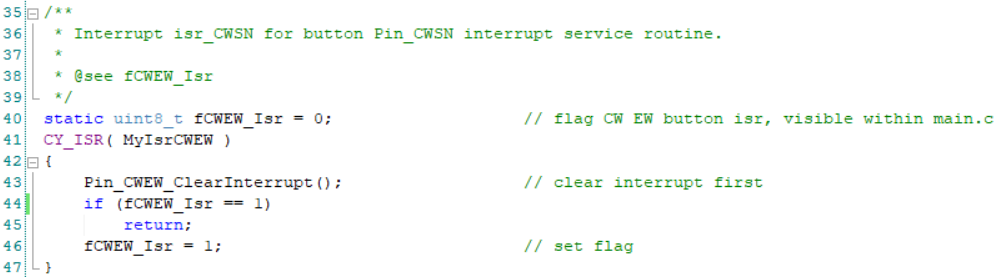
**c. Verändern Sie die Helligkeit der LED über die Menüsteuerung, z.B. ‘+‘ und ‘-‘**

****

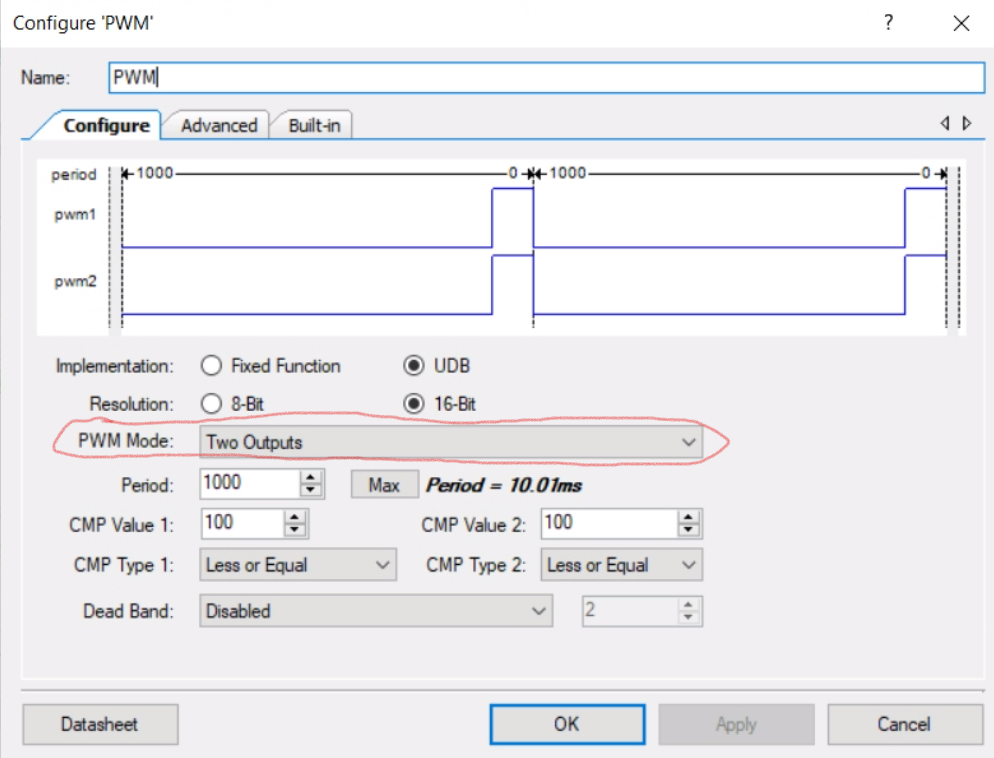
**5. Erweiterungen**

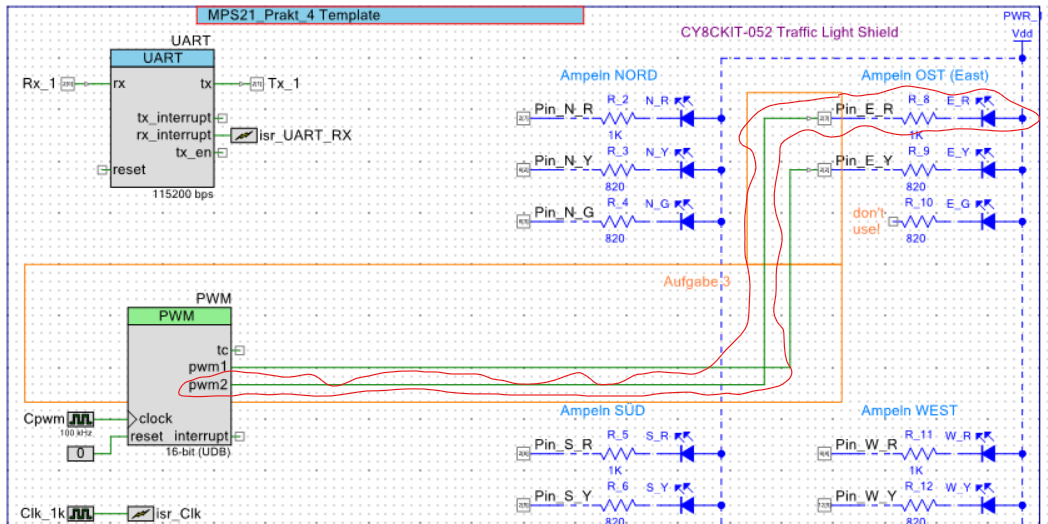
**a. Fügen Sie die Fußgänger-Ampeln in Ihre State Machine ein.**

**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 Komponente, LED-Pin modifizieren)**

****

****