Praktikum 4

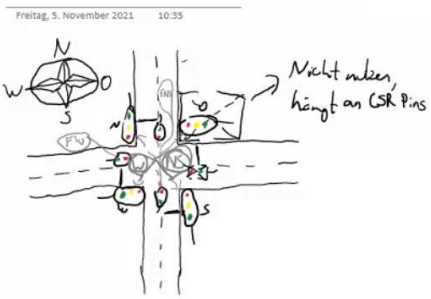
Lernziele: PWM (Pulsweitenmodulation) zur Regelung der Helligkeit einer LED. Zeitgesteuerete. Ampelsteuerung mit Implementierung eines Zustandsautomaten (finite state machine (FSM))

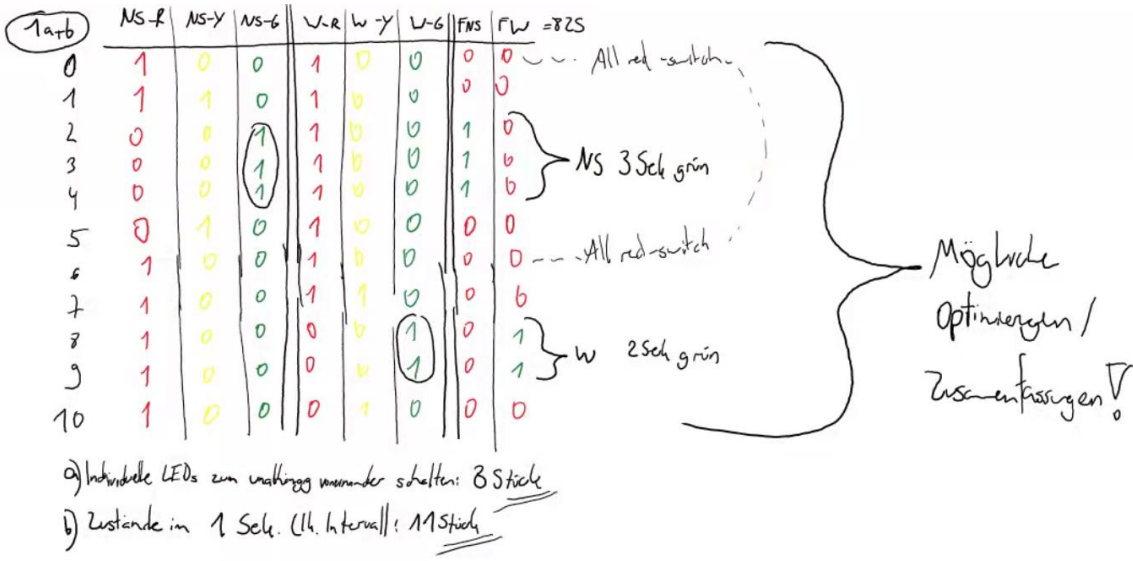
1. Es soll eine Ampelsteuerung realisiert werden, bei der in Nord-Süd-Richtung und Ost-West-Richtung der Verkehr abwechselnd fließen soll:

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.

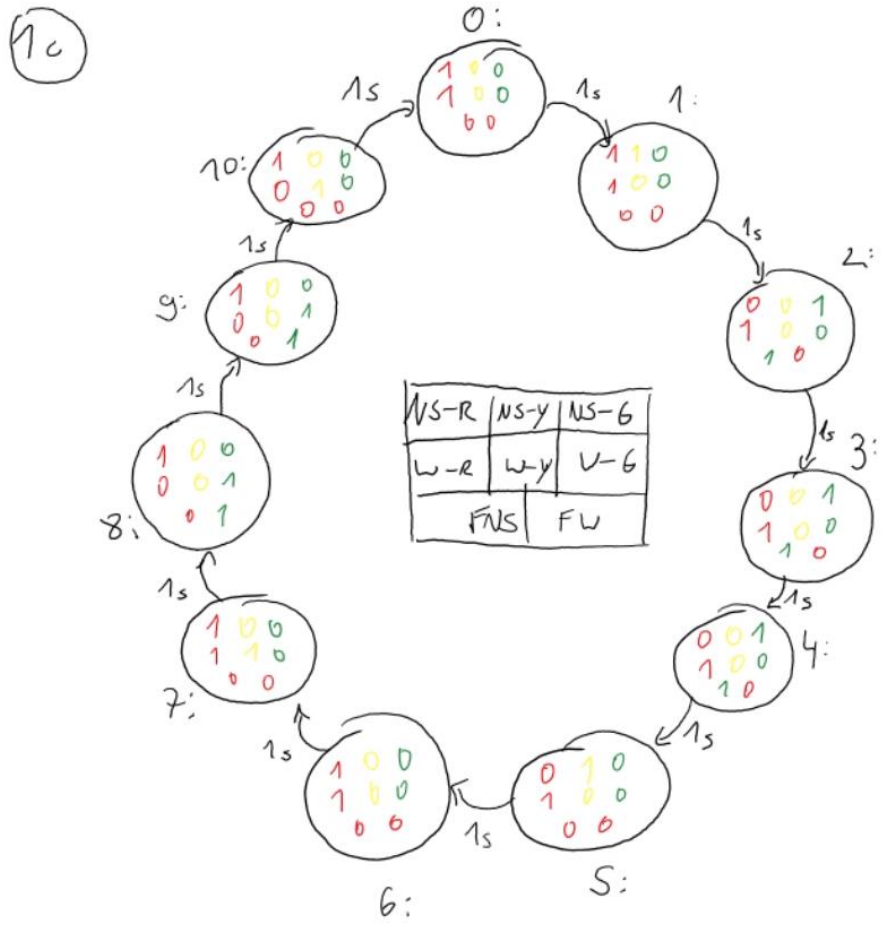
a. Wieviele Ausgaben hat die FSM? Also LED's die unabhängig voneinander geschaltet werden? (Hinweis: fassen Sie die jeweiligen Farben bei Nord-Süd zusammen, wie bei einer richtigen Kreuzung). Fußgänger-Ampeln nicht vergessen!

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).

HAL.h

```
1  #include <stdio.h>
2  #define LED_ON  (0u)                    // LED on (a
3  #define LED_OFF (1u)                    // LED off (
4
5  uint8_t itoLED(int i)
6  {
7      if(i == 0)
8          return LED_OFF;
9      return LED_ON;
10 }
```

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

```
48  /**
49   * Application clock interrupt service routine for isr_Clk every second
50   *
51   * @see fClock
52   */
53  static uint8_t  fsClock = 0;                    // s flag
54  static uint32_t count_ms = 0;                   // ms count since start
55  CY_ISR( IsrAppClk )
56  {
57      count_ms++;                                 // increment ms timestamp
58      if ( (count_ms % 1000) == 0 )               // next 1 s reached
59         fsClock = 1;                             // set flag
60  }
```

```
173     /* initialize CLK */
174     isr_Clk_StartEx( IsrAppClk );               // register application clock
```

```
186     // application loop
187     for(;;)
188     {
189         // change state of intersection every second from CLK interrupt
190         if ( fsClock )
191         {
192             fsClock = 0;                        // reset clock interrupt for the next second
193             handleIntersection();
194         }
```

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!

```
65  // ============================================================================================
66  // ============================== START TRAFFIC LIGHTS LOGIC ==================================
67  // ============================================================================================
68  static int state = 0;
69  // matrix (optimized)
70  // ns-r, ns-y, ns-g, w-r, w-y, w-g, fns, fw
71  static int States[7][8] =
72  {
73      {1,0,0,1,0,0,0,0}, //0,6     0
74      {1,1,0,1,0,0,0,0}, //1       1
75      {0,0,1,1,0,0,1,0}, //2,3,4   2
76      {0,1,0,1,0,0,0,0}, //5       3
77      {1,0,0,1,1,0,0,0}, //7       4
78      {1,0,0,0,0,1,0,1}, //8,9     5
79      {1,0,0,0,1,0,0,0}, //10      6
80  };
```

```
// get array of 8 independent light states (on/off = 1/0) as argument and set traffic lights accordingly
void setLightsFromArray(int *arr)
{
    // traffic lights north/south
    Pin_N_R_Write(itoLED(arr[0]));
    Pin_N_Y_Write(itoLED(arr[1]));
    Pin_N_G_Write(itoLED(arr[2]));

    Pin_S_R_Write(itoLED(arr[0]));
    Pin_S_Y_Write(itoLED(arr[1]));
    Pin_S_G_Write(itoLED(arr[2]));

    // traffic lights west
    Pin_W_R_Write(itoLED(arr[3]));
    Pin_W_Y_Write(itoLED(arr[4]));
    Pin_W_G_Write(itoLED(arr[5]));

    // pedestrian lights north/south and west
    Pin_S_CW_Write(itoLED(arr[6]));
    Pin_E_CW_Write(itoLED(arr[7]));
}
```

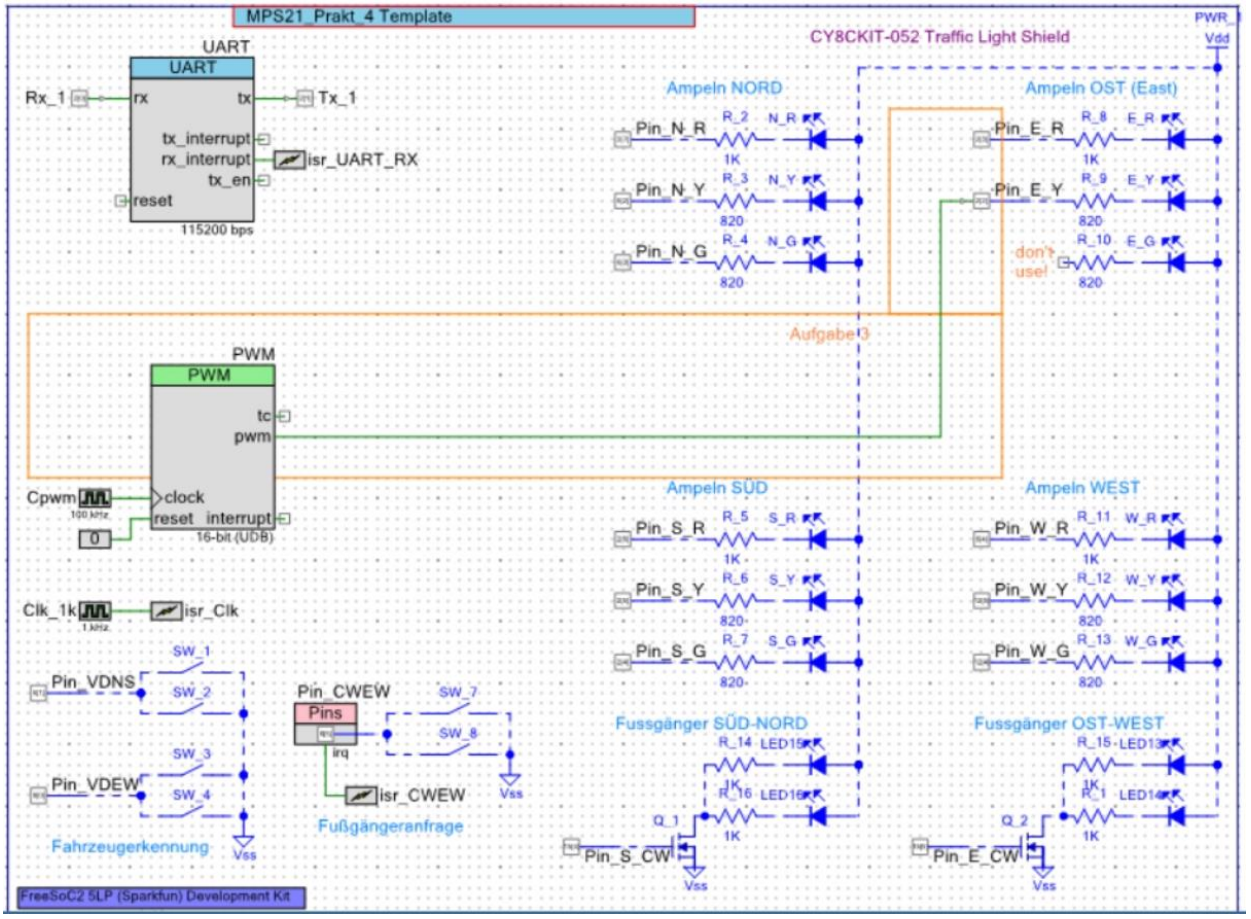
```
// parse intersection-state and pass led-state array to changeLightsFunction
void handleIntersection()
{
    // rotate through the 10 (optimized to 6 array lines) intersection-states and switch traffic leds from according
    switch (state)
    {
        case 0:
        case 6:
            setLightsFromArray(States[0]);
            break;

        case 1:
            setLightsFromArray(States[1]);
            break;

        case 2:
        case 3:
        case 4:
            setLightsFromArray(States[2]);
            if ( fCWEW_Isr )                    // pedestriant wants to cross
            {
                UART_PutString("\n\rPassengers NS pass!\n");
                UART_PutChar(7);
                for(int loop=0;loop<10000000;loop++);
                fCWEW_Isr = 0;
            }
            break;
        case 5:
            setLightsFromArray(States[3]);
            break;

        case 7:
            setLightsFromArray(States[4]);
            break;

        case 8:
        case 9:
            setLightsFromArray(States[5]);
            if ( fCWEW_Isr )                    // pedestriant wants to cross
            {
                UART_PutString("\n\rPassengers W pass!\n");
                UART_PutChar(7);
                for(int loop=0;loop<10000000;loop++);
                fCWEW_Isr = 0;
            }
            break;

        case 10:
            setLightsFromArray(States[6]);
            break;

        default:
        UART_PutString("Something went wrong.");
    }
    // when done increase intersection-state for next iteration (until 10 and restart/rotate from start state)
    state++;
    if (state > 10)
        state = 0;
}
```

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

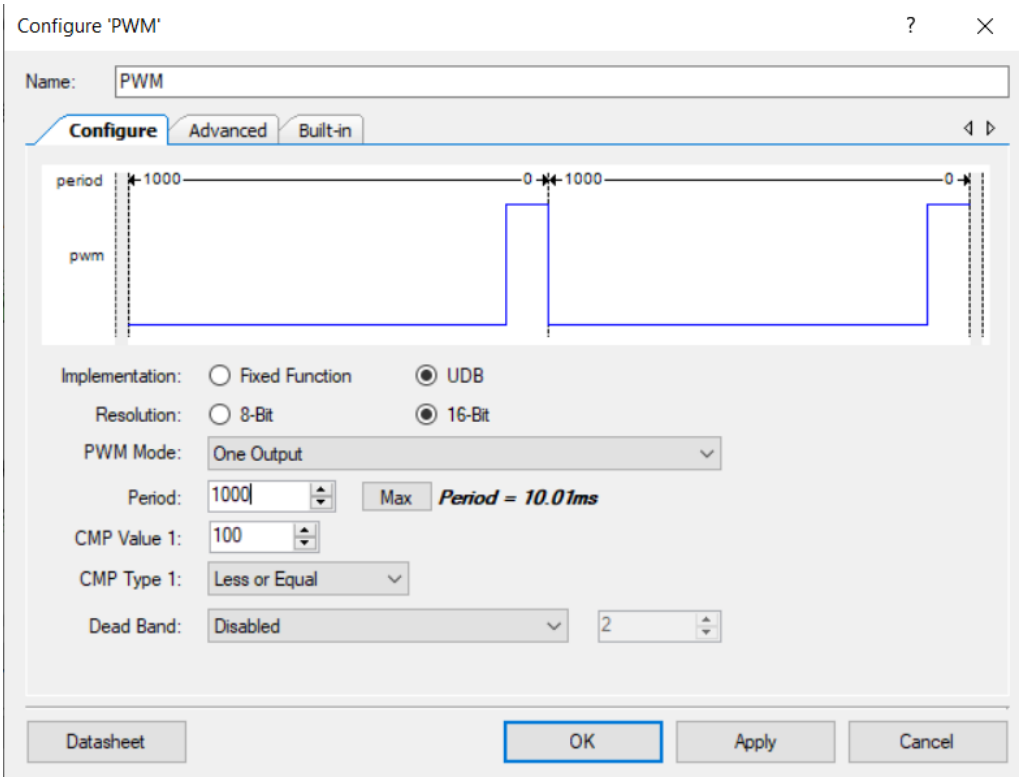
Einstellungen > Terminal:

- Übertragen: CR
- Absenden: CR+LF

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

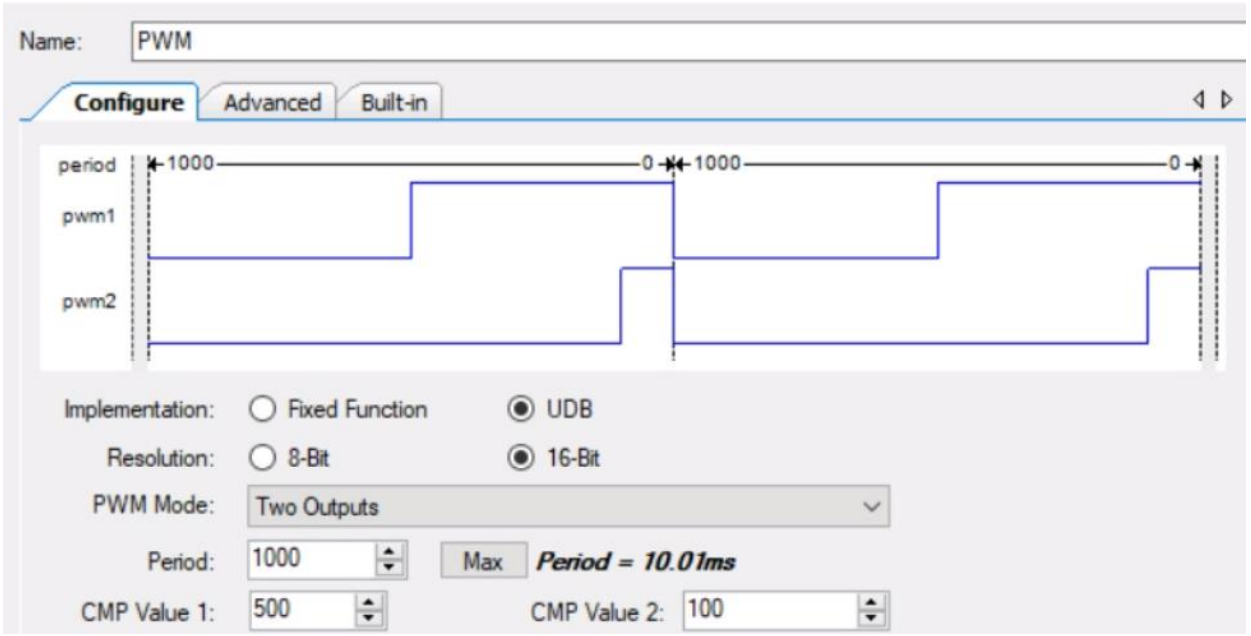
a. Auf welchen Wert muss Periode eingestellt werden, um eine Wiederholung von etwa 10ms zu erreichen?

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 ‘-‘

```
case '+':
    UART_PutString( "\n\rBrighter!\n");
    PWM_WriteCompare1 ( PWM_ReadCompare1() - 50 );
    PWM_WriteCompare2 ( PWM_ReadCompare2() - 50 );
    break;

case '-':
    UART_PutString( "\n\rDarker!\n");
    PWM_WriteCompare1 ( PWM_ReadCompare1() + 50 );
    PWM_WriteCompare2 ( PWM_ReadCompare2() + 50 );
    break;
```

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.

```
/**
 * Interrupt isr_CWSN for button Pin_CWSN interrupt service routine.
 *
 * @see fCWEW_Isr
 */
static uint8_t fCWEW_Isr = 0;                   // flag CW EW button isr, visible within main.c
CY_ISR( MyIsrCWEW )
{
    Pin_CWEW_ClearInterrupt();                  // clear interrupt first
    if (fCWEW_Isr == 1)
        return;
    fCWEW_Isr = 1;                              // set flag
}
        case 2:
        case 3:
        case 4:
            setLightsFromArray(States[2]);
            if ( fCWEW_Isr )                    // pedestriant wants to cross
            {
                UART_PutString("\n\rPassengers NS pass!\n");
                UART_PutChar(7);
                for(int loop=0;loop<10000000;loop++);
                fCWEW_Isr = 0;
            }
            break;
        case 8:
        case 9:
            setLightsFromArray(States[5]);
            if ( fCWEW_Isr )                    // pedestriant wants to cross
            {
                UART_PutString("\n\rPassengers W pass!\n");
                UART_PutChar(7);
                for(int loop=0;loop<10000000;loop++);
                fCWEW_Isr = 0;
            }
            break;
```

c. Die rote LED an einem zweiten PWM-Kanal anschließen (gleiche Komponente, LED-Pin modifizieren)