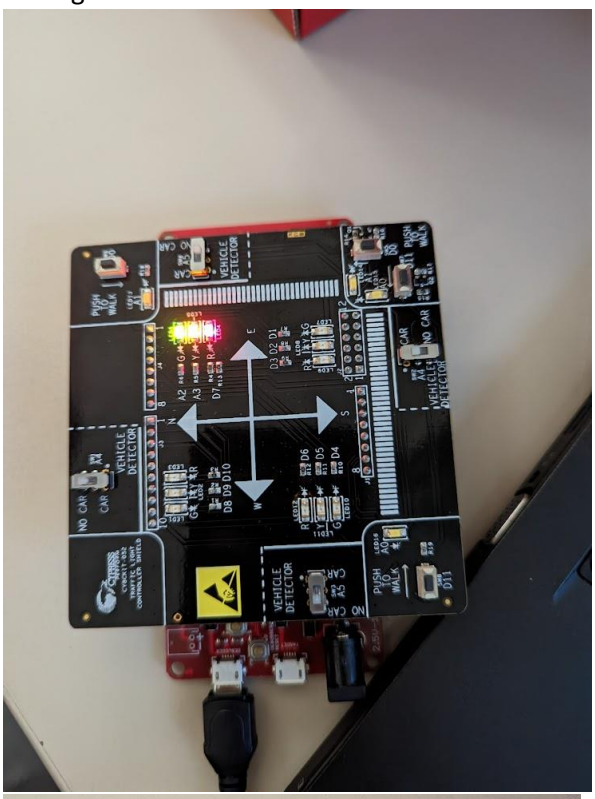
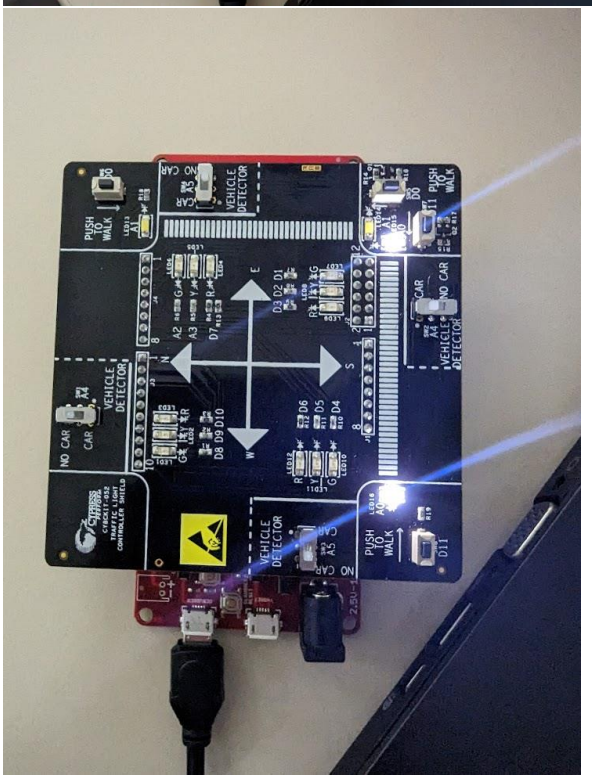
Praktikum 1 MPS
Trung Thieu Quang – 771043

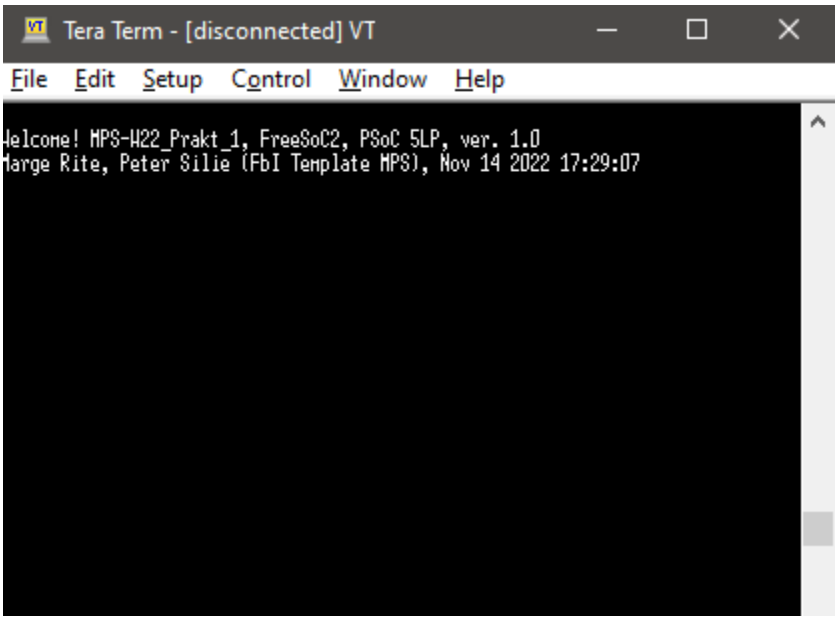
Vorbereitung:

1. Erledigt





2. …



3. …
    a. Um die LEDs R, Y, G zu leuchten, muss PIN_N_R, PIN_N_Y, PIN_N_G zu low gesetzt werden
    b. Um LED_CW zu leuchten, muss PIN_E_CW zu high gesetzt werden
    c. Wenn ein Switch davon gedrückt ist, wird PIN_CWEW zu low gesetzt
4. LED Steuerung:

```
if ( chr != 0 ) {               // da kam ein Zeichen ...
    // ein kleines Menu ...
    // >>> (TODO um Einträg erweitern!) <<<
    //  z.B. grüne LED ein- 'G' und 'g' ausschalten
    switch (chr) {              // Zeichen auswerten
        case 'x':               // 'x' auswerten
            UART_PutString( "... da ist ein <x>'\n\r" );
                                // String Ausgabe, Zeilenabschluss LF CR
            break;
        case 'R':
            Pin_N_R_Write(LED_ON);
            break;
        case 'Y':
            Pin_N_Y_Write(LED_ON);
            break;
        case 'G':
            Pin_N_G_Write(LED_ON);
            break;
        case 'r':
            Pin_N_G_Write(LED_OFF);
            break;
        case 'y':
            Pin_N_G_Write(LED_OFF);
            break;
        case 'g':
            Pin_N_G_Write(LED_OFF);
            break;
        default:
            sprintf( buffer, "erwarte ein <x>, aber <%c> wurde eingegeben\n\r", chr );
            UART_PutString( buffer );
            break;
    } // end switch
    chr = 0;                    // nicht vergessen (mit ISR)
}
```

5. Tasten Abfrage:

   Beim Pin_CWEW_read() == 0 wird kein signal gesendet
   mit der Schleife werden alle Ereignise (z.B blinking) gestoppt und deshalb wird kein Ereignis beim Tastesdrücken verpasst

```
/* +++ Taste CWEW abfragen +++ */
if ( Pin_CWEW_Read() == 0 ) {   // warum auf 0 abfragen? beim 0 wird kein signal gesendet
    while(!Pin_CWEW_Read()) {       // warte auf loslegen
    }
    Pin_E_CW_Write( !Pin_E_CW_Read() ); // toggle
}
```

6. Led Blinking

```
    volatile uint32_t loop;                 // geht mit DEBUG, aber nicht mit RELEASE
                                        // warum???
                                        // Frage, was macht 'volatile'?
    //because in the loop there's nothing to do, so the compiler will try to be "smart"
    // and optimize the code for speed and memory, it will jump over for loop since there's nothing there
    // by using volatile we force the code to do what we want to do
    // => die Schleife wird von Kompiler wegoptimiert
//      Pin_N_R_Write( LED_ON );        // anschalten
//      for ( loop = 0; loop < 1000000; loop++ )    // warten mit Schleife
//          ;
//      Pin_N_R_Write( LED_OFF );       // ausschalten
//      for ( loop = 0; loop < 1000000; loop++ )    // warten mit Schleife
//          ;
```

7. ..
   a. Beim schnellen Drücken an der Taste funktioniert LED_CW nicht. Wegen des Blinkings wird das Drücken-Ereignis verpasst
   b. Wenn man der Button ein bisschen halten und dann loslegen dann funktioniert es noch
   c. Man kann mit einer Schleife lösen, indem man alle andere Ereignisse halt, wenn der Button gedrückt wird