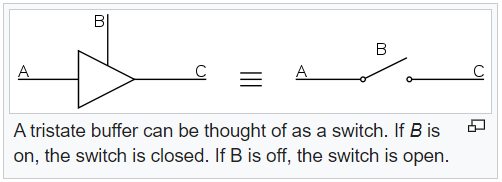
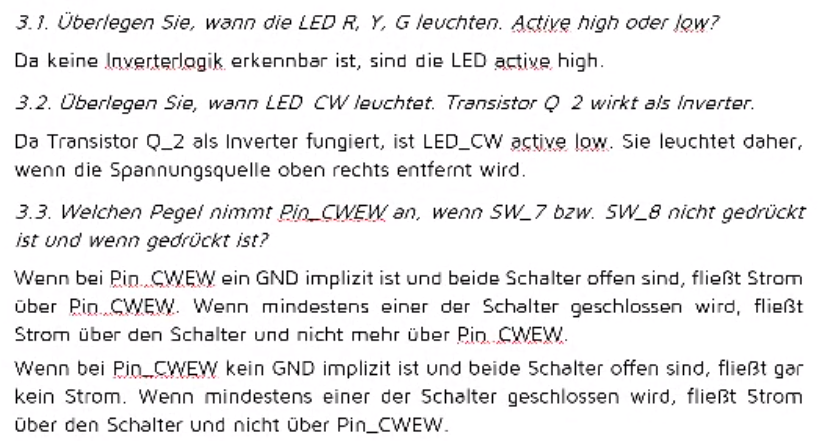
**MPS – Praktikum Termin 2 – Sebastian Zill(769544)**

*Lernziele: PSoC Creator einrichten, Evaluierungs-Board anschließen und programmieren, Ein-*

*Ausgabe über PC-Terminal. Uart, I/O-Peripherie. C-Schlüsselwort volatile*

1. Starten Sie ***PSoC-Creator*** und laden Sie das Projekt Termin 2: MPS21\_Prakt\_2.
2. Schließen Sie das *FreeSoC2*-Entwicklungsboard an und warten Sie ab bis alle Treiber installiert sind. (kann etwas dauern).
3. Starten Sie ***TeraTerm*** und verbinden Sie mit COM<xy> (Datei > Neue Verbindung >Seriell)  
   Es ist der COM<xy>: KitProg USB-UART(COM<xy>)
4. Einstellungen > Serieller Port: baud: 115200, data: 8, parity: none, stop: 1
5. Einstellungen > Terminal Übertrage: CR, Absenden: CR+LF
6. Projekt MPS21\_Prakt\_2.zip entpacken. Doppelclick auf *MPS21\_Prakt\_2.cyprj*
7. PSoC Creator: Build > Generate Application. (muss fehlerfrei sein)
8. Creator: Build <your project> (muss fehlerfrei sein)
9. *(geht nur mit Board!)* Creator: Debug > Program. Program wird auf Board übertragen, TeraTerm-Fenster sollte Begrüßungstext anzeigen.
10. Erzeugen Sie eine einfache Bildschirmausgabe:  
    Formatierte Ausgabe (von Zahlen, char, strings, …)  
    *Hinweis*: es hat sich als ressourcenschonender für den PSoC erwiesen, statt des  
    bekannten printf( “<fmt>“, var, […] );  
    eine formatierte Ausgabe in einen genügen großen string **buffer** zu machen.  
    **sprintf( buffer, ( “<fmt>“, var, […] );** und diesen dann mit  
    **UART\_PutString( buffer );** auszugeben
11. Vorbereitung: Betrachten Sie *TopDesign*. Fenster zeigt Schaltplan mit Komponenten: UART (Pins 2[0] und 2[1] für RX und TX), vier LEDs (Pin\_N\_R 2[7], Pin\_N\_Y 6[2], Pin\_N\_G 6[3] und Pin\_E\_CW 15[5]) als Output und einen Button (Pin\_CWEW 6[5]) als Input.
12. Überlegen Sie, wann die LEDs R, Y, G leuchten? Active high oder low?  
    **Active-high, da nach Schaltplan die Diode direkt eine Versorgungsspannung von PWR\_1 erhält und der Strom durch einen 1K / 820 Ohm zu GRND fließt. (keine Inverter Logik)**
13. Überlegen Sie, wann LED\_CW leuchtet? Transistor Q\_2 wirkt als Inverter!  
    **Da der Schaltkreis erst geschlossen ist, wenn der Knopf E\_CW den Transistor öffnet, immer dann wenn der Knopf gedrückt wird.**
14. Welchen Pegel nimmt Pin\_CWEW an, wenn SW\_7 bzw. SW\_8 nicht gedrückt ist und wenn gedrückt ist.  
      
    **Gedrückt: LOW (Schaltkreis geschlossen - Positiv fließt nach GND)  
    Nicht Gedrückt: HIGH (Schaltkreis nicht geschlossen - Positiv von isr\_CWEW bleibt)**

**Angenommen auf den Pins liegt ebenfalls ein GND…**

****

1. Lesen vom Zeichen vom Terminal, Menu, Schalten der LED(s) über Terminaleingaben
2. Lesen Sie Zeichen von der UART: **UART\_GetChar()**
3. Schreiben Sie ein *einfaches* Menu, welches durch die Eingabezeichen gesteuert wird.
4. Steuern Sie mit den Zeichen das Verhalten der LED(s) (s. 3, An- u. Ausschalten)
5. Taste CWEW abfragen und Aktion, toggeln der weißen LEDs überprüfen
6. Wird nur unzuverlässig funktionieren, weil die Taste prellt!  
   (*Prellen bedeutet, dass der mechanische Kontakt nicht sofort zuverlässig den neuen Zustand einnimmt, und anfangs mehrmals zurückspringt*)
7. Wie könnte man per Software entprellen?
8. Aktivieren Sie das Blinken der roten LED durch Entfernen der Kommentarzeichen ‚//‘. Funktion: *an … warten … aus … warten*. Das Warten wird mit Schleifen realisiert (*nicht ändern!*).
9. Kompilieren Sie mit **Debug** und testen Sie Funktion.
10. Kompilieren Sie jetzt mit **Release** und testen Sie Funktion. Warum geht es jetzt nicht mehr? (es scheint, als ob die Schleifen nicht ausgeführt werden)
11. Erinnern Sie sich an Rechnerarchitektur und Optimierung des Codes durch den Compiler. (wird die Variabel ‚loop‘ denn gebraucht?)  
    Recherchieren Sie in C-Referenzen nach ‚volatile‘. Verwenden und testen Sie!
12. Testen Sie jetzt mit 6. nochmal die Funktion der Keyboard-Eingabe und des Buttons
13. Sind Sie mit der Schnelligkeit der Reaktion auf Terminal-Eingabe zufrieden?
14. Funktioniert der Button überhaupt noch zuverlässig?
15. Warum ist das so? *Könnte man so eine Lösung erfolgreich verkaufen?*