

Versuch 3 - Einführung in PSoC Creator

Dieser Versuch ist eine Einführung in die integrierte Entwicklungsumgebung (IDE) PSoC Creator und besteht aus zwei Teilen. Dabei werden in den ersten beiden Teilaufgaben alle Schritte bis zum Ziel ausführlich dokumentiert.

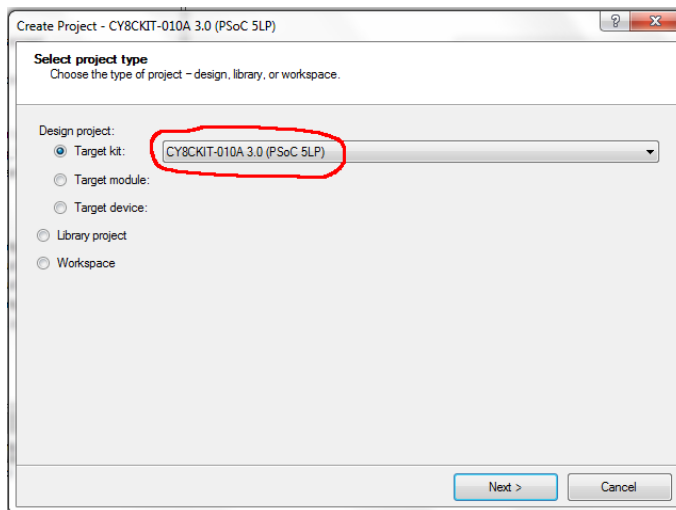
Beginnen werden Sie mit einem Projekt, bei dem Sie lernen, wie Meldungen auf einem alphanumerischen Display ausgeben können. Anschließend realisieren Sie eine zeitabhängige Manipulation eines Digitalausganges, wobei hierfür zwei Möglichkeiten aufgezeigt werden.

Sie sollten diese beiden Anleitungen Punkt für Punkt durcharbeiten. In den weiteren Teilaufgaben und Versuchen werden sich diese Schritte ständig wiederholen.

Teil 1 / Ausgabe auf dem LCD Display


Ziel dieses Aufgabenteils ist es, Sie mit dem LCD Display und dessen Ansteuerung vertraut zu machen.

- Starten Sie das Programm PSoC Creator 4.2
- Wählen Sie „File -> New -> Project“
- Im neu geöffneten Fenster wählen Sie unter „Target Kit“
 - CY8CKIT-010A 3.0 (PSoC 5LP)



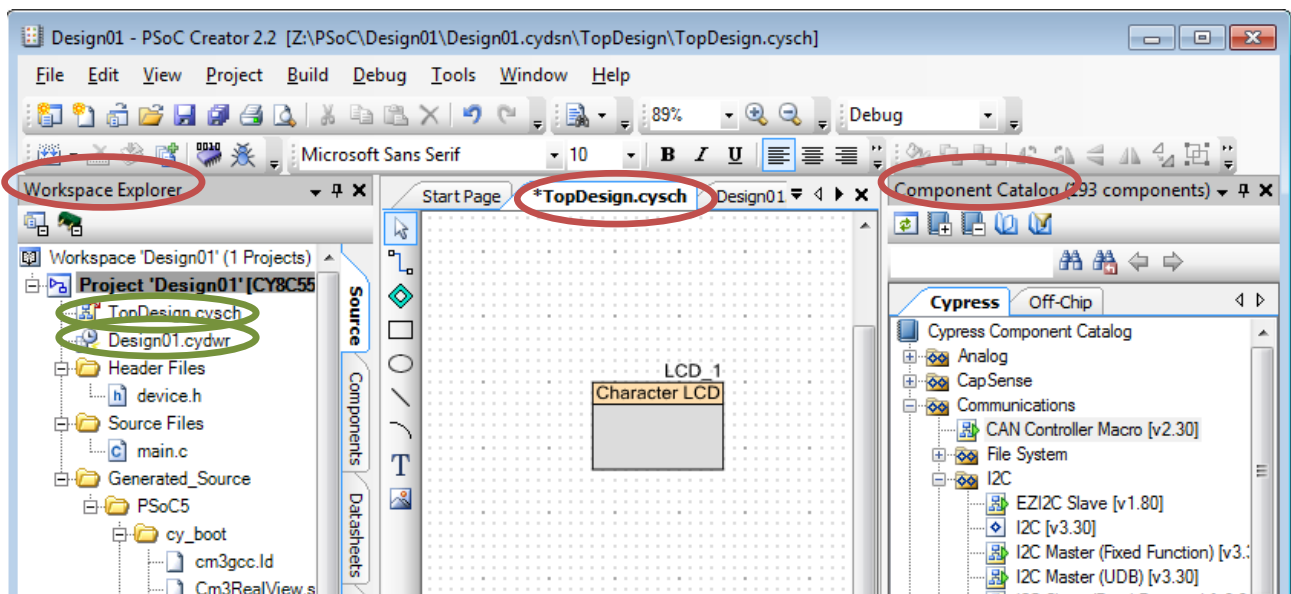
Alternative Möglichkeit:

- Sollte der Punkt „Target Kit“ **nicht** anwählbar sein, dann verfahren Sie wie folgt:
- Wählen sie unter „Target Device“ im ersten Dropdown-Menü „PSoC 5LP“ und im zweiten Dropdown-Menü „CY8C58LP“.

- Klicken Sie auf „Next >“
- Wählen Sie „Empty Schematic“ und dann „Next >“
- Geben Sie unter „Workspace name“ einen Namen ein (z.B. „Ausgabe LCD“)
- Unter „Location“ wählen Sie einen Zielordner für Ihre Datei 

Erstes Projekt

Nach Anlegen des neuen Projekts öffnet sich die *TopDesign.cysch* Datei. Hier können Sie die benötigten Komponenten zusammenstellen und konfigurieren. Für diesen Versuch wählen Sie in der *Component Catalog* Spalte (auf der rechten Seite des Programmfensters) den *Cypress* Tab. Im Unterordner *Display* (oder mit Hilfe der Suchfunktion) wählen Sie *Charakter LCD [v2.20]* und ziehen Sie das Symbol auf die Arbeitsfläche. Beim Doppelklick auf die *Character LCD* erscheint das *Configure* Fenster. Dort können Sie die Komponente konfigurieren und unter *Datasheet* das zugehörige Datenblatt aufrufen. Benennen Sie das Display mit *LCD_1* und drücken Sie anschließend *OK*.



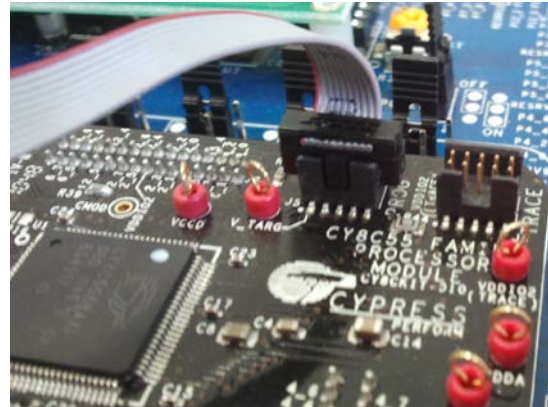
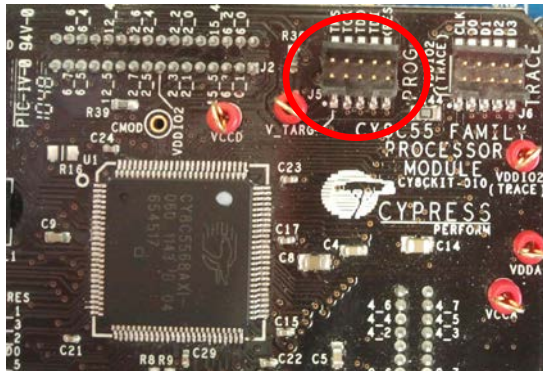
Wählen Sie im *Workspace Explorer* (auf der linken Seite des Programmfensters) den Eintrag „Design Wide Resources“ (*.cydwr). Ordnen Sie im Untermenü „Pins“ dem LCD Display Port *P2[6:0]* zu. Nun öffnen Sie die *main.c* Datei. Hier kann der Programmcode in C eingegeben werden. Im *Datasheet* der Komponenten finden Sie die jeweiligen Funktionen.

LCD_Start(); startet das Display.

Hinweis: Achten Sie immer darauf, dass sie den richtigen Komponentennamen in die Funktion schreiben. In diesem Fall schreiben Sie für die Start Funktion des Displays *LCD_1_Start();*.

Programmieren Sie eine Ausgabe für das LCD Display.

Stellen Sie sicher, dass die *MiniProg3* Komponente an den *PROG* Pins via USB am Computer angeschlossen ist.



Anschließend soll der *J12* Jumper rechts unten vom Display in der *ON* Position sein (die unteren 2 Pins sollen verbunden sein).



Das folgende Bild zeigt ein Beispielprogramm, an dem Sie sich orientieren können. Gerne können Sie auch eigene Ideen verwirklichen.

```

4 void main()
5 {
6
7     LCD_1_Start();           /*Startet das LCD Display*/
8     LCD_1_Position(0,0);     /*Setzt den Cursor auf die Position 0, 0
9                               (erste Zeile, erste Spalte)*/
10    LCD_1_PrintString("Hello"); /*Gibt den String "Hello" aus*/
11    LCD_1_Position(1,6);     /*Setzt den Cursor auf die Position 1, 6
12                               (zweite Zeile, siebte Spalte)*/
13    LCD_1_PrintString("World!"); /*Gibt den String "World!" aus*/
14
15    for(;;)
16    {
17
18        /* Place your application code here. */
19    }
20 }
21
22 /* [] END OF FILE */

```

Beispielprogramm

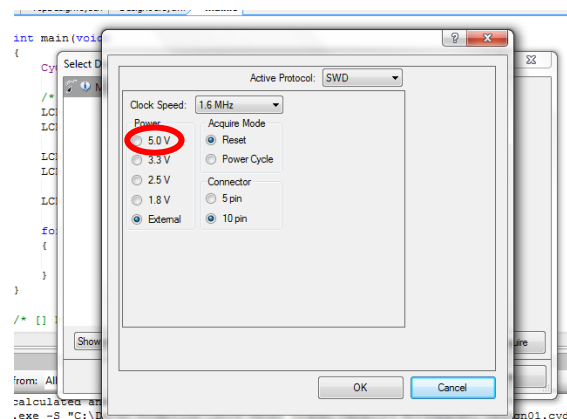
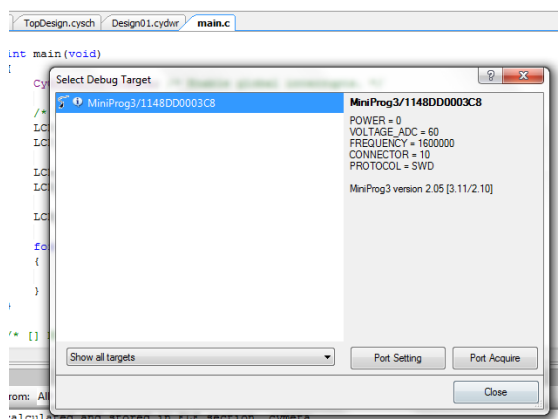
Nach der Eingabe des Programmcodes führen Sie *Build* aus und programmieren Sie anschließend.

Build (Shift+F6)



Program (Ctrl+F5)

Wenn zum ersten Mal versucht wird, nach einem Neustart das PSoC-Device zu programmieren, dann erscheint nach kurzer Zeit das Fenster „Select Debug Target“. Im Praktikum wird das Experimentierboard über das Programmiergerät mit Spannung versorgt. Um diese zu aktivieren klicken Sie auf „Port Setting“ und dann im Unterpunkt „Power“ auf „5.0 V“. Die nun erscheinende Warnmeldung bestätigen Sie mit „Continue“. Zurück im Fenster „Select Debug Target“ klicken Sie nun auf „Port Acquire“ und dann auf „Close“.



Nun können Sie mit einem Klick auf den „Program“-Button das Device programmieren.

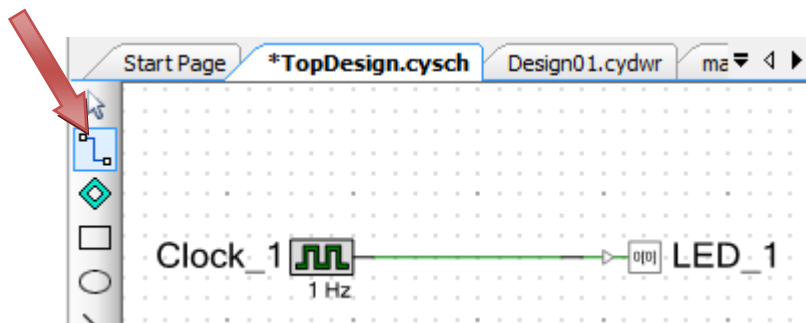
Teil 2 / Blinkende LED

Ziel dieser Aufgabe ist es, eine LED anzusteuern und zum Blinken zu bringen. Es werden zwei mögliche Varianten vorgestellt.

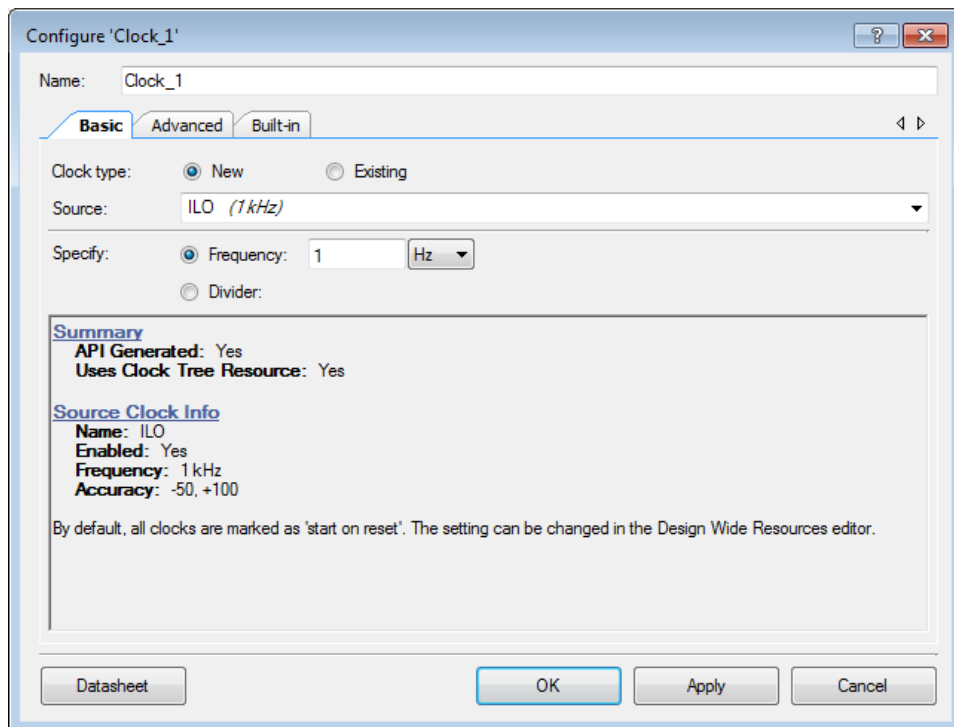
Variante 1

Legen Sie ein neues Projekt an.

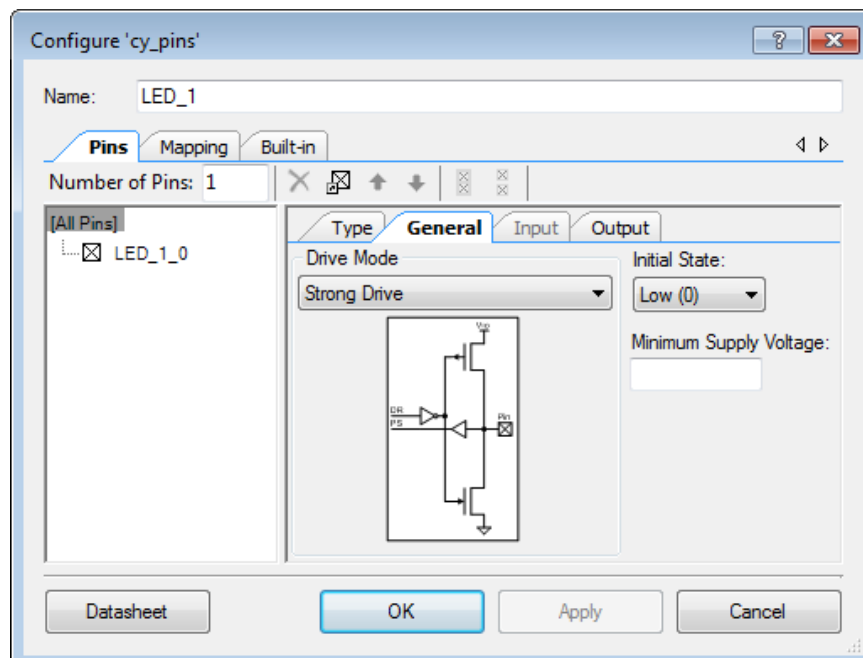
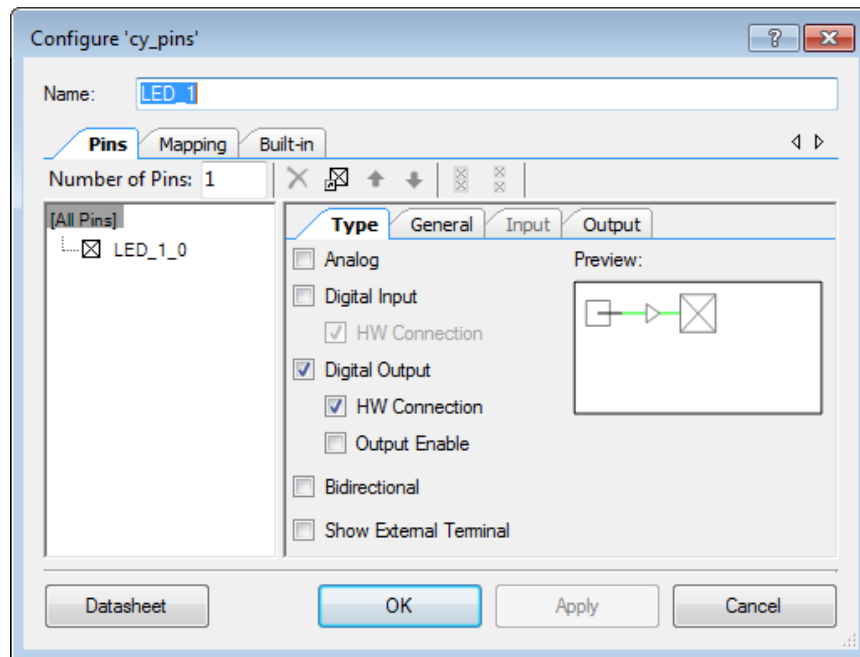
Ziehen Sie in der *TopDesign.cysch* Datei ein *Clock [v2.20]* und ein *Digital Output Pin [v2.20]* auf die Arbeitsfläche. Benennen Sie den *Output Pin* in *LED_1* um und verbinden Sie ihn mit dem *Clock* mit Hilfe des *Wire Tools*.



Konfigurieren Sie die *Clock*-Komponente. Die Einstellungen können Sie dem folgenden Bild entnehmen.



Die Einstellungen für den *LED*-Pin sind die folgenden:



Nun müssen Sie in der *.cydwr*-Datei dem Pin einen Port mit LED zuordnen. Die haben an Pin 4[2] und 4[3] jeweils eine LED zur Verfügung.

In der *main.c*-Datei müssen Sie nun noch die Komponente *Clock_1* starten.

```

2
3  void main()
4  {
5
6      Clock_1_Start();
7
8      for(;;)
9      {
10
11      }
12
13  /* [] END OF FILE */

```

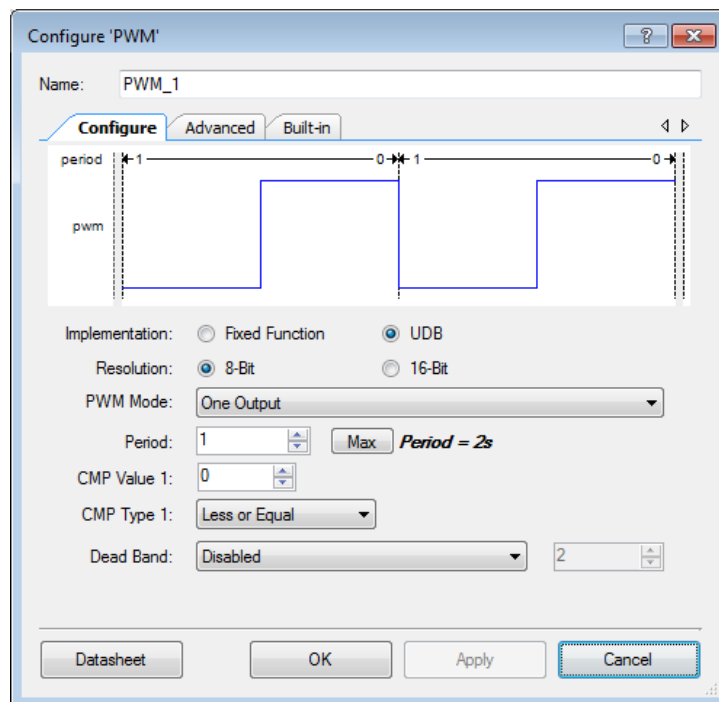
Jetzt können den *Build* durchführen und den PSoC programmieren.

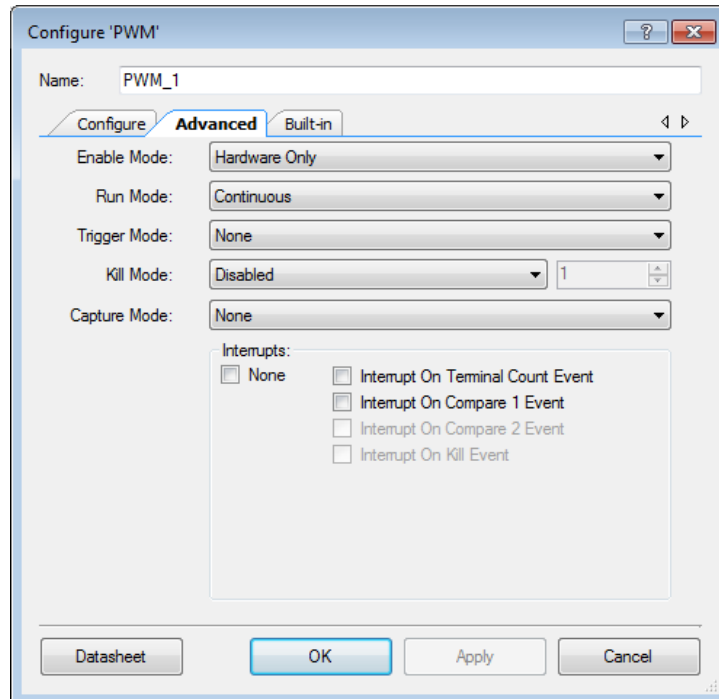
Variante 2

Legen Sie ein neues Projekt an. Sie benötigen folgende Bauteile:

- *PWM[v3.30]*
- *Clock[v2.2]*
- *Digital Output Pin [v2.20]*
- *Logic High '1' und Logic Low '0'*

Konfigurieren Sie nun die Bauteile.





Configure 'PWM'

Name: PWM_1

Configure Advanced Built-in

Enable Mode: Hardware Only

Run Mode: Continuous

Trigger Mode: None

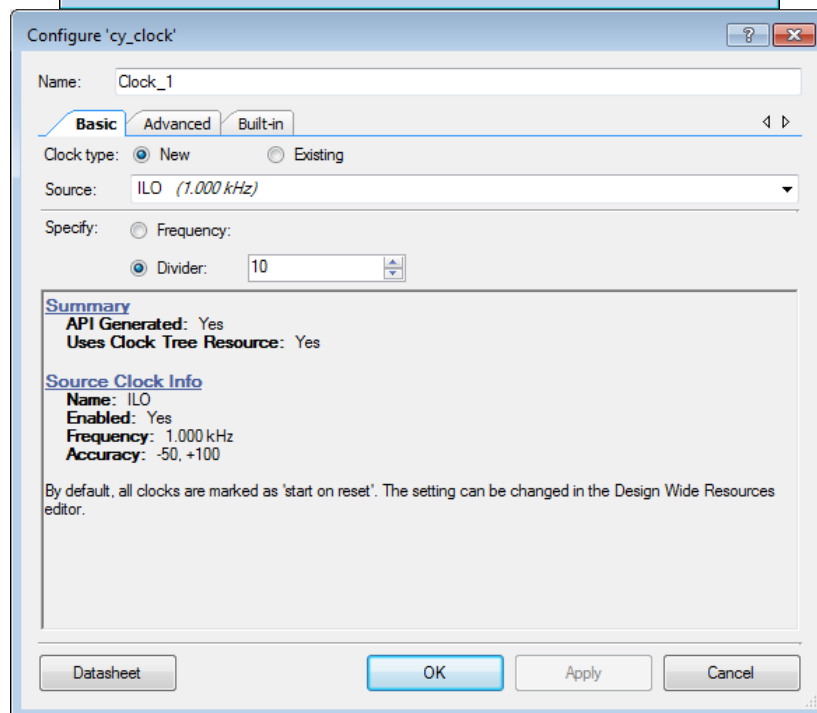
Kill Mode: Disabled 1

Capture Mode: None

Interrupts:

- ☐ None
- ☐ Interrupt On Terminal Count Event
- ☐ Interrupt On Compare 1 Event
- ☐ Interrupt On Compare 2 Event
- ☐ Interrupt On Kill Event

Datasheet OK Apply Cancel



Configure 'cy_clock'

Name: Clock_1

Basic Advanced Built-in

Clock type: ☒ New ☐ Existing

Source: ILO (1.000 kHz)

Specify: ☐ Frequency: ☐ Divider: 10

Summary

API Generated: Yes
Uses Clock Tree Resource: Yes

Source Clock Info

Name: ILO
Enabled: Yes
Frequency: 1.000 kHz
Accuracy: -50, +100

By default, all clocks are marked as 'start on reset'. The setting can be changed in the Design Wide Resources editor.

Datasheet OK Apply Cancel

Verbinden Sie die Bauteile wie folgt:

- *Logic High '1'* auf den *enable*-Eingang des PWM
- *Logic Low '0'* auf den *reset*-Eingang des PWM
- *Clock_1* mit dem *clock* des PWM
- *LED1* mit dem *pwm*-Ausgang

Ordnen Sie in der *.cydwr*-Datei dem LED-Pin einen Hardwareport zu.

In der *main.c*-Datei müssen Sie PWM und Clock starten. Die entsprechenden Befehle finden Sie in den Datenblättern. Im Anschluss führen Sie *Build* und *Programm* aus.

Weitere Aufgaben

Diese Aufgaben sind zum Großteil mit dem Wissen aus den ausführlich beschriebenen Versuchen lösbar, erfordern teils aber auch eine zusätzliche Recherche. Benutzen Sie hierzu die Datenblätter der einzelnen Bausteine im PSoC-Creator.

1. Lassen sie die LEDs abwechselnd blinken.
2. Finden Sie heraus, ab welcher Frequenz das menschliche Auge kein Blinken sondern ein dauerhaftes Leuchten der LED wahrnimmt.
3. Lassen Sie einen Text auf dem LCD-Display von links nach rechts laufen.
4. Im Wechsel: Lassen Sie die linke LED 10x blinken, anschließend die rechte LED 10x blinken.