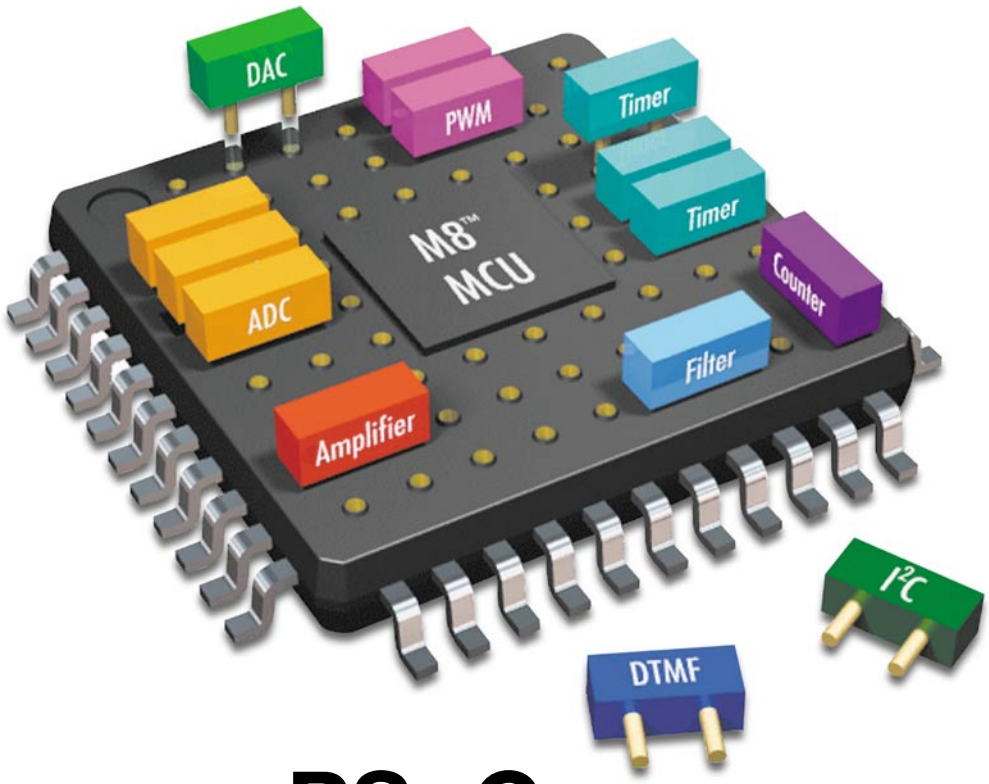


PSoC™ – Programmable System-on-Chip

Teil 1

Die PSoC™-Familie von Cypress Microsystems ist an Flexibilität und Einsatzbreite kaum zu überbieten, befindet sich doch hier ein komplettes System auf einem Chip. Neben einem leistungsfähigen Prozessor und den allgemein aus Mikrocontroller-Konfigurationen bekannten Peripherie-Komponenten wie Watchdog-Timer und Oszillator bietet der PSoC™-Baustein programmierbare analoge und digitale Schaltungsblöcke, so dass der Elektronik-Entwickler hier individuelle A/D-Wandler, Filter usw. erstellen kann. Unser Artikel beschreibt den Aufbau und die Möglichkeiten eines PSoC™-Bausteins und einen Leitfaden zum ersten kleinen Projekt.

Eierlegende Wollmilchsau?

Mikrocontroller sind in fast jedem modernen Gerät zu finden. Sie steuern, regeln, erfassen Eingaben und sind für eine übersichtliche Ausgabe der Ergebnisse verantwortlich. Obwohl die meisten Mikrocontroller bereits über interne digitale Komponenten (Timer, Zähler usw.) verfügen, sind für ein funktionierendes System noch weitere externe Beschaltungen wie z. B. Operationsverstärker, Komparatoren, Filter, A/D-Wandler, D/A-Wandler etc. erforderlich. Ein solches System weist schon einen hohen Grad an Flexibilität auf, der jedoch in einer vorhandenen Schaltung durch die externe Beschaltung des Mikrocontrollers begrenzt wird. Was wäre, wenn diese Flexibilität noch weiter gehen würde und programmierbare digitale und analoge Peripheriekomponenten neben dem Mikrocontroller in einem Chip integriert wären? Man hätte ein komplettes System auf einem Chip und könnte die externe Beschaltung auf das Nötigste, z. B. platzmäßig nicht zu integrierende Kondensatoren oder Bedien- und Einstellelemente, minimieren.

Ein solches System hat Cypress Microsystems mit seiner PSoC™-Familie bereits vor einiger Zeit auf den Markt gebracht.

Tabelle 1: Übersicht der wichtigsten vorhandenen Module

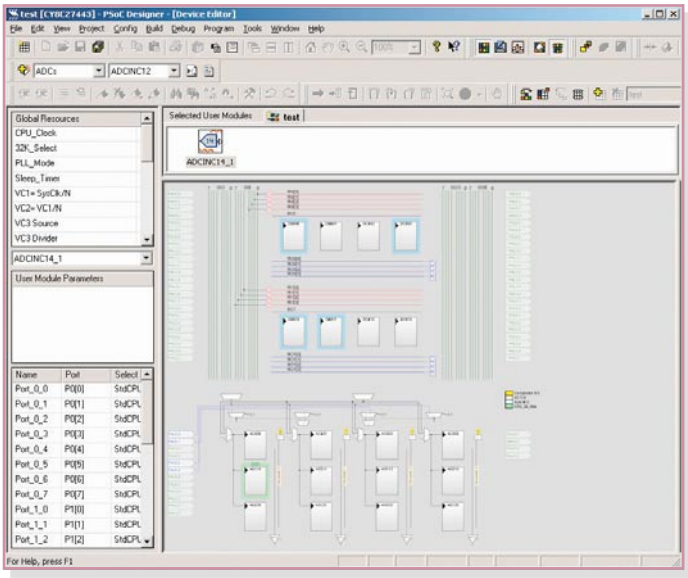
Digitale Module	Analoge Module
• Timer 8, 16, 24, 32 Bit	• Verstärker
• Counter 8, 16, 24, 32 Bit	• Komparatoren
• PWM 8, 16 Bit	• Analog-Digital-Wandler
• I²C Master und Slave	• Digital-Analog-Wandler
• SPI Master und Slave	• Filter
• Full Duplex UART	• DTMF-Tonerzeugung
• IrDA-Empfänger und -Sender	
• CRC-Generator	

Die Abkürzung „PSoC“ steht für „Programmable System-on-Chip“ und verweist auf die Flexibilität, mit der diese Bausteine eingesetzt werden können. Neben den aus einem „normalen“ Mikrocontroller bekannten Elementen wie Timer, UART usw. verfügt der PSoC™-Baustein über zusätzliche, programmierbare analoge und digitale Blöcke, die entsprechend der Anwendung individuell konfigurierbar sind. Eine Übersicht der wichtigsten vorhandenen Module zeigt Tabelle 1. Diese Module können in einem grafischen Editor der kostenlosen Entwicklungsumgebung „PSoC™-Designer“ (siehe Abbildung 1) per „drag & drop“ einfach ausgewählt und auf den entsprechenden Blöcken abgelegt werden. Hierbei kann man die Ein- und Ausgänge des gewählten Moduls einfach in der Ansicht mit den gewünschten Pins des PSoC™-Bausteins oder weiteren Modulen verbinden, so dass tatsächlich ein individueller Mikrocontroller mit zusätzlichen analogen und digitalen Funktionen entsteht. Die Routinen zur Verwendung bestimmter Module (z. B. Analog-Digital-Umsetzer) generiert die Entwicklungsumgebung gleich mit. So sind unterschiedlichste Konfigurationen schnell und einfach erstellbar. Während der Laufzeit kann man per Software sogar zwischen verschiedenen Konfigurationen umschalten, was wir an einem Beispiel betrachten wollen:

Ein Temperaturlogger soll in einem Ferienhaus in festgelegten Abständen die Temperatur aufzeichnen und einmal täglich die erfassten Daten über ein Modem an den heimischen Rechner übertragen.

Um diese Aufgabe zu lösen, werden zwei Konfigurationen für den PSoC™-Baustein erstellt. Die erste besteht aus den Modulen zur Aufbereitung und Messung des Sensorsignals und ist während des gesamten Tages aktiviert. In der zweiten Konfiguration werden die Module zur Ansteuerung für den Modembetrieb ausgewählt und entsprechend verbunden.

Bild 1: Modul- und Verbindungsansicht im PSoC™-Designer



Während des gesamten Tages ist Konfiguration 1 aktiv, erfasst die Temperaturdaten und speichert sie ab. Einmal täglich wird auf Konfiguration 2 umgeschaltet und das Modem aktiviert. Das Gerät wählt die Rufnummer des heimischen Rechners und überträgt die gespeicherten Daten. Danach wird die erste Konfiguration wieder so lange aktiviert, bis die nächste Datenübertragung ansteht.

Die Architektur des PSoC™ stellt dem Anwender also sehr viele Möglichkeiten zur Verfügung. Der unbestrittene Vorteil ist die Flexibilität, aber auch die Einsparmöglichkeit von externen Komponenten sollte nicht außer Acht gelassen werden. Dies ist gerade für den Hobby-Elektroniker sehr nützlich, da man neben der eigentlichen digitalen Steuerschaltung „mal schnell“ einen Verstärker oder Filter „aufbauen“ kann, auch wenn man gerade keinen (passenden) Operationsverstärker zur Hand hat.

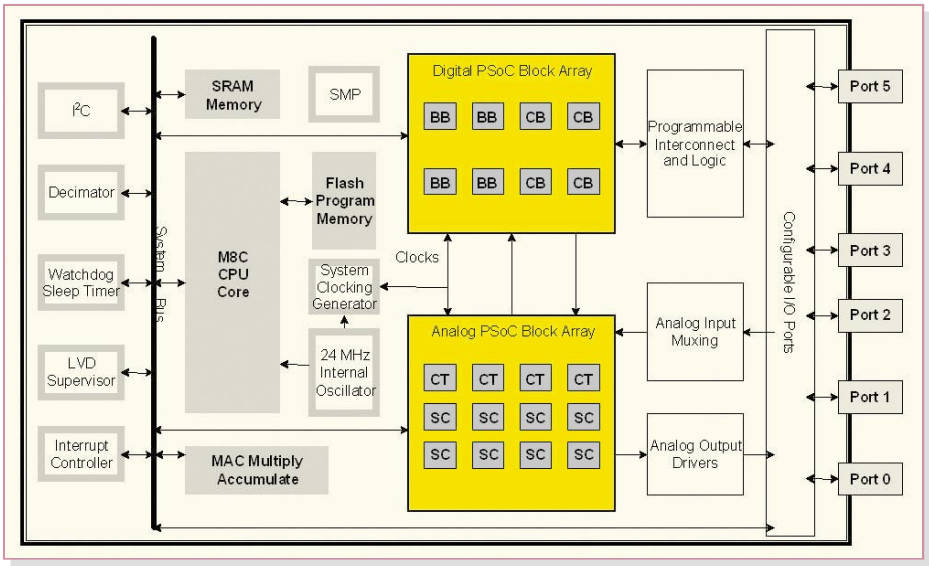


Bild 2: Blockschaltbild eines PSoC™-Bausteins

Der Aufbau – eine Übersicht

Das Blockschaltbild des PSoC™-Bausteins ist in Abbildung 2 zu sehen. Auf den ersten Blick ist zu erkennen, dass neben dem leistungsfähigen 8-Bit-Prozessor (M8C-CPU) zunächst zahlreiche (digitale) Peripheriebausteine vorhanden sind:

- bis zu 32 kB wiederprogrammierbares Flash-ROM
- bis zu 2 kB SRAM
- interner 24-MHz-Oszillator
- Echtzeit-Uhr
- I²C-Schnittstelle
- Hardware-Multiplizierer

Der Clou des PSoC™ sind allerdings die programmierbaren analogen und digitalen PSoC™-Blöcke (gelb hinterlegt), mit denen man die bereits oben aufgezählten und viele weitere Funktionen realisieren kann. Eine Beschreibung des internen Aufbaus und die genaue Funktionsweise dieser Blöcke würde den Rahmen dieses Artikels sprengen und kann den zahlreichen Datenblättern und „Application Notes“ (siehe [1]) entnommen werden.

Die konfigurierbaren Ein-/Ausgabeports des PSoC™ können als Ein- oder Ausgang für die analogen PSoC™-Blöcke, als Ein- oder Ausgang für die digitalen PSoC™-Blöcke oder über den Systembus als digitale Ports für den Prozessor dienen.

Der Einstieg

Der Einstieg in die Welt des PSoC™ sollte für jemanden, der über etwas Erfahrung in der Mikrocontroller-Programmierung verfügt, kein großes Problem darstellen, jedoch ist eine gewisse Einarbeitungszeit zwingend erforderlich, um die Möglichkeiten dieses Bausteins begreifen und anwenden zu können. Hierbei bieten die vielen Informationen auf der PSoC™-Homepage [1] des Herstellers eine gute Hilfe. Es

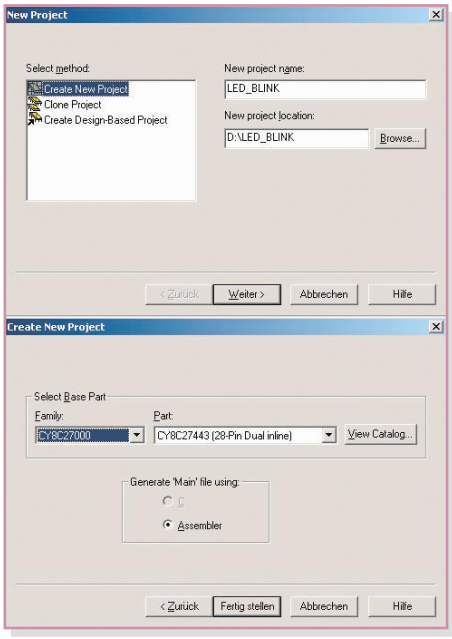


Bild 3: Erstellung eines neuen Projektes

werden neben den Datenblättern für die einzelnen Bausteine zahlreiche „Application Notes“ zur Verfügung gestellt. Hierbei handelt es sich zum Teil um Beschreibungen einzelner wichtiger Punkte, z. B. was man beim Einstieg beachten muss (AN2010: „Getting Started with PSoC“) oder Erläuterungen zum Verständnis der analogen Blöcke (AN2041: „Understanding Switched Capacitor Analog Blocks“).

Den weitaus größeren Teil bilden jedoch konkrete Anwendungsbeispiele, z. B. Bewegungsmelder oder ein Telefonlogger. Aus dem Studium dieser Beispiele erhält man wertvolle Informationen, die bei der Umsetzung des eigenen Projektes sehr hilfreich sind.

Eine weitere effektive Informationsquelle – nicht nur für den Einstieg – ist das „PSoC Customer Forum“ [2]. In diesem Forum sind PSoC™-Anwender, sowohl Einsteiger als auch Profis, aus der ganzen Welt aktiv und versuchen, entsprechende Fragen zu beantworten bzw. Probleme zu lösen. In den meisten Fällen ist die gestellte Frage bereits nach wenigen Stunden beantwortet. Ein Blick in dieses Forum lohnt sich auch dann, wenn man gerade keine konkrete Frage hat, sondern sich über schon vorhandene Lösungen informieren oder Anregungen für eigene Projekte erhalten möchte.

Und schließlich bietet Cypress über den Internet-Support so genannte Tele-Trainings-Module an, die Schritt für Schritt den Weg zur Realisierung eines konkreten Projektes gehen. Die Erläuterung zu den dort herunterladbaren Projekten findet allerdings nach Registrierung per Telefon-Konferenzschaltung in den USA statt, ist für deutsche Anwender also eher kostenintensiv, und sie findet zudem für uns nachts statt. Aber immerhin geben schon die Projektfiles selbst einen erstklassigen Einblick in die Entwicklung von Projekten.

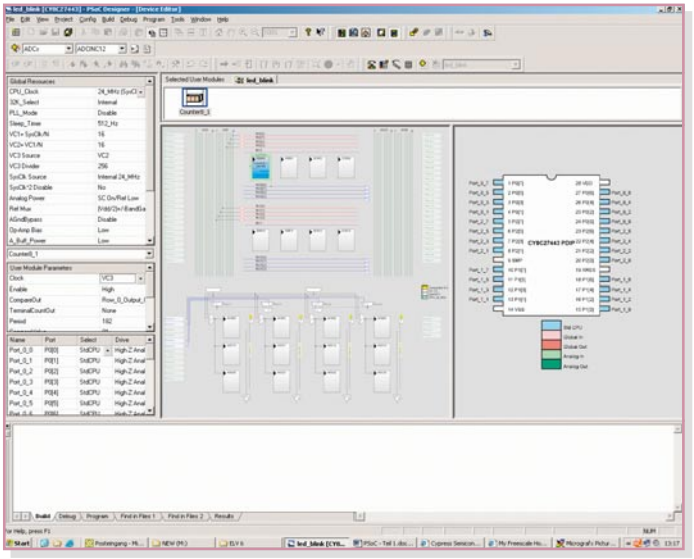


Bild 5: Verbindungsansicht im „Device Editor“

Das erste Projekt

Im Folgenden beschreiben wir die Erstellung eines kleinen ersten Projekts, um den Einstieg zu erleichtern und die ersten Anfangsschwierigkeiten auf ein Minimum zu begrenzen. Hierbei soll mit geringem Aufwand eine LED an Port 2.0 zum Blinken gebracht werden. Bevor man jedoch starten kann, muss die zugehörige Entwicklungsumgebung – der PSoC™-Designer – von der Homepage [1] (Sektion „Software & Drivers“) in der neusten Version heruntergeladen und installiert werden. Danach wird die Software gestartet und ein neues Projekt angelegt (File → New Project → Create New …).

In den daraufhin erscheinenden Fenstern wählt man zunächst den Projektnamen und dann den gewünschten Mikrocontroller aus (Abbildung 3). Nachdem alle Einstellungen über den Button „Fertig stellen“ bestätigt sind, generiert der PSoC™-Designer das Grundgerüst der Applikation und startet darauf die Modulauswahl im „Device Editor“ (Abbildung 4). Links im Fenster sind alle verfügbaren Module aufgelistet, aufgeteilt in mehrere Gruppen. Im mittleren Teil wird das Datenblatt des aktuell ausgewählten Moduls dargestellt. Es beschreibt neben den technischen Angaben auch die Funktionsprototypen, d. h. die Befehle zur Verwendung des Moduls. In den meisten Fällen ist sogar ein kleines Beispielprogramm („Sample Code“) aufgeführt.

Für die blinkende LED wird ein 8-Bit-Zähler benötigt, so dass man im Abschnitt „Counters“ das Modul „Counter8“ mit einem Mausklick auswählt.

Im mittleren Teil ist jetzt die komplette Beschreibung der technischen Eigenschaften, der Parameter sowie der Steuerbefehle (8-Bit-Counter-API) sichtbar. Zur Übernahme dieses Moduls in das Design klickt

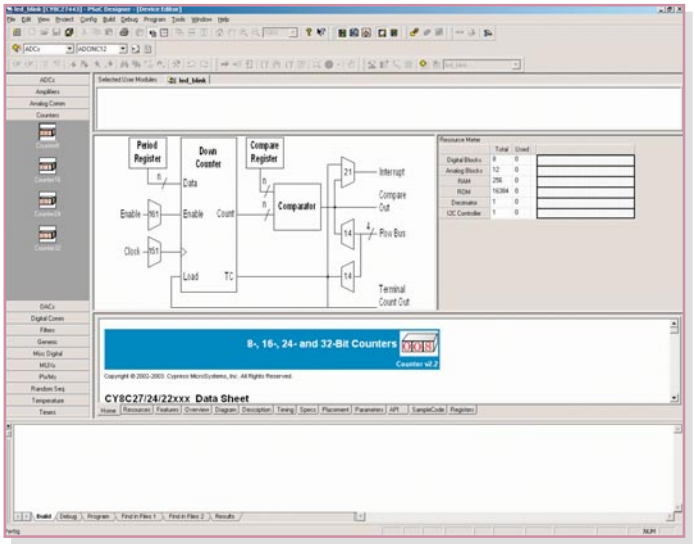


Bild 4: Modulauswahl im „Device Editor“

man mit der rechten Maustaste auf das Icon „Counter8“ und betätigt den Menüpunkt „Select“, daraufhin wird das Modul in der Ansicht „Selected User Modules“ sichtbar. Auf der rechten Seite des Fensters – im „Resource Meter“ – erfolgt die Anzeige der verfügbaren und der verbrauchten Ressourcen des gewählten PSoC™-Bausteins. Man kann hier erkennen, dass der 8-Bit-Counter einen digitalen Block und 67 Byte im ROM für die Ansteuerfunktionen belegt.

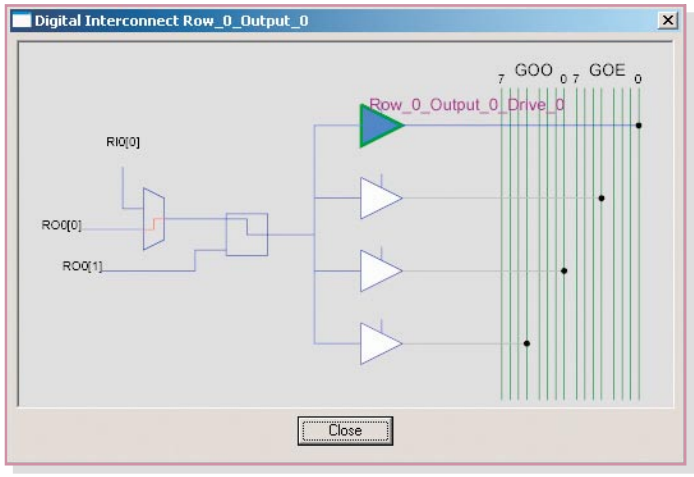
Jetzt müssen die Konfigurationen der globalen Ressourcen und des Zählermoduls vorgenommen werden. Hierzu schaltet man den „Design Editor“ über den Menüpunkt „Config → Interconnect“ in die Verbindungsansicht (Interconnect View, Abbildung 5). Hier wird das Zählermodul zunächst über den Menüpunkt „Config → Place User Module“ platziert. Im Anschluss daran trägt man die erforderlichen Konfigurationen, wie in Abbildung 6 gezeigt, ein. Dabei sind im Bereich „Global Resources“ zunächst der Systemtakt und dann die Teilerwerte der drei internen Taktteiler VC1 bis VC3 festzulegen.

Global Resources	
CPU_Clock	24_MHz (SysClk/1
32K_Select	Internal
PLL_Mode	Disable
Sleep_Timer	512_Hz
VC1= SysClk/N	16
VC2= VC1/N	16
VC3 Source	VC2
VC3 Divider	256
SysClk Source	Internal 24_MHz
SysClk*2 Disable	No
Analog Power	SC On/Ref Low
Ref Mux	(Vdd/2)+/-BandGap
AGndBypass	Disable
Op-Amp Bias	Low
A_Buff_Power	Low
SwitchModePump	OFF
Trip Voltage [LVD (SMP)]	4.64V (5.00V)
LVDThrottleBack	Disable
Supply Voltage	5.0V
Watchdog Enable	Disable

Counter8_1	
User Module Parameters	
Clock	VC3
Enable	High
CompareOut	Row_0_Output_0
TerminalCountOut	None
Period	182
CompareValue	91
CompareType	Less Than Or Equal
InterruptType	Terminal Count
ClockSync	Sync to SysClk
InvertEnable	Normal

Bild 6: Konfiguration

Bild 7: Auswahl des Ausgangstreibers



Im Abschnitt „User Module Parameters“ konfiguriert man den Zähler so, dass der Ausgang des Teilers VC3 die Taktquelle bildet. Der Parameter „Period“ und der Vergleichswert mit „CompareValue“ bilden die Angabe des Endwertes. Entsprechend der Einstellung „CompareType“ wird der Ausgang „CompareOut“ bei einem Zählwert von kleiner oder gleich dem Vergleichswert auf High-, ansonsten auf Low-Pegel gesetzt. Bevor man diesen Ausgang auf ein weiteres Modul bzw. auf einen Ausgang schalten kann, ist er auf eine Zeilenleitung (hier Row_0_Output_0) zu führen („CompareOut“). Diese Leitung wird dann mit einer Leitung des Ausgangsbusses („GlobalOutput“, GOO bzw. GOE) verbunden, indem man den Ausgangsmultiplexer der Zeilenleitung anklickt und im daraufhin erscheinenden Fenster (Abbildung 7) die Treiberstufe 0 auswählt. Zum Abschluss wird die Treiberstufe 0 mit dem Portpin P 2.0 verbunden, indem PORT_2_0 angeklickt und über „Select“ auf „GlobalOutputEven_0“ programmiert wird. Somit ist die Konfiguration des Zählerbausteins beendet und man kann die Applikation über den Menüpunkt „Config → Generate Application“ generieren.

Dann erfolgt der Start des eigentlichen Programmier-Editors, des „Application-Editor“ über den Menüpunkt „Config → Application Editor“ sowie der Aufruf der Datei „main.asm“. Jetzt fügt man unter der Zeile „_main:“ lediglich den Befehl

```
call Counter8_1_Start
```

ein, und die Programmierung ist beendet. Auf den Befehlssatz gehen wir hier nicht weiter ein, da die PSoC™-Homepage hierzu einige übersichtliche Dokumente zur Verfügung stellt (siehe [3] und [4]) und alle Modulbeschreibungen mit entsprechenden Programmierbeispielen ausgestattet sind.

Mittels der Taste „F7“ wird das Projekt nun kompiliert und eine Programmdatei (hex) erzeugt, die man mit einem geeigneten Programmiergerät zum Test in einen PSoC™-Baustein laden kann.

Um ein besseres Verständnis für die Verschaltungs- und Programmiermöglichkeiten der digitalen und analogen Blöcke zu erhalten, kann man nach dem erfolgreichen Test mit diesem Beispielprojekt etwas „spielen“. Ändern Sie einen Parameter und prüfen Sie die Auswirkungen! Verbinden Sie den Ausgang des Zählers mit Port 1.5 anstatt mit Port 2.0!

Dieses Beispielprojekt zeigt den prinzipiellen Ablauf einer auf PSoC™ basierenden Entwicklung und bildet somit einen kleinen Einstieg, um das gesamte Konzept der PSoC™-Familie zu durchblicken und anzuwenden. Bis es jedoch so weit ist, muss der Anwender noch etwas Fleißarbeit investieren, die ihm aber durch den umfangreichen Herstellersupport erleichtert und mit einem ultrakompakten Schaltungsdesign sowie der Möglichkeit zur individuellen Mikrocontrollerlösung honoriert wird.

Im zweiten Teil dieses Artikels wird ein Programmiergerät vorgestellt, mit dem ein einfacher Einstieg in die Welt des PSoC™ möglich ist.

Internet:

[1] Cypress’ PSoC-Homepage
http://www.cypress.com/PSoC

[2] Cypress’ PSoC Consumer Forum
http://www.cypress.com/forums/categories.cfm?catid=3

[3] PSoC™-Designer:
Assembly Language User Guide
http://www.cypress.com/cfuploads/pub/Assembler.book.pdf

[4] Instruction Set Quick Reference
http://www.cypress.com/cfuploads/pub/intructions.pdf