

PSoC™ – Programmable System-on-Chip Teil 2

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. Nach der Systembeschreibung und der eines ersten Projektes gehen wir mit dem zweiten Teil in die Praxis und stellen ein passendes Programmiergerät vor, das den Einstieg in die Welt des PSoC™ vereinfacht.

Bequem programmieren

Die PSoC™-Bausteine der Fa. Cypress MicroSystems bieten sowohl dem professionellen Schaltungsentwickler als auch dem ambitionierten Hobby-Elektroniker eine sehr große Anwendungsbandbreite und viele Möglichkeiten zur schnellen Erstellung einer funktionierenden Schaltung auf einem Chip.

Die Besonderheit des Cypress PSoC™ stellen die konfigurierbaren Analog- und Digitalblöcke dar, mit denen der Mikro-

controller um Funktionsblöcke wie Timer, Zähler, Filter, A/D-, D/A-Umsetzer usw. erweitert werden kann. Diese Eigenschaften und Funktionen haben wir im ersten Teil dieses Artikels ausführlich vorgestellt. Obwohl die Programmierung, wie wir sie an unserem ganz einfachen Beispielprojekt gezeigt haben, eine umsichtige Fleißarbeit erfordert, lohnt der Aufwand, denn es entsteht ein besonders kompaktes Schaltungsdesign, das sich von herkömmlichen Lösungen durch den geringen Hardware-Aufwand abhebt. Gerade komplexe und vom Anwender im ultrakompakten Design

geforderte Lösungen sind nur über einen solchen Weg realisierbar. Man denke hier nur an Mini-Messwertaufnehmer mit eigener Intelligenz, die Transpondertechnik, künftige RFID-Anwendungen usw.

Aber ohne ein passendes Programmiergerät ist ein einfacher Einstieg natürlich nicht möglich. Das „PSoC™ MiniProg“ von Cypress MicroSystems ist solch ein einfaches und günstiges Programmiergerät, mit dem die PSoC™-Mikrocontroller direkt in der Zielschaltung programmiert werden können. Dieses Programmiergerät, das mit einer kleinen Entwicklungs-

**Tabelle 2:
ISSP-Programmierschnittstelle**

SDAT	Serieller Datenein-/ausgang
SCLK	Übertragungstakt
X _{Res}	Reset
V _{SS}	Masse
V _{DD}	Betriebsspannung

platine für die ersten Versuche und Experimentierschaltungen mit dem PSoC™-Controller geliefert wird, ist Mittelpunkt unserer Praxisbetrachtungen.

In-System Serial Programming

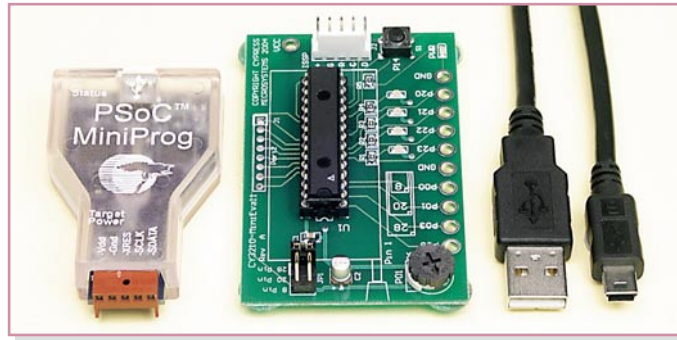
Die Daten werden über eine 5-polige Schnittstelle, deren Belegung in Tabelle 2 aufgeführt ist, vom PC aus in den Baustein übertragen. An dieser Stelle kommt das ISSP™(In-System Serial Programming)-Protokoll [1] zum Einsatz. Das bedeutet, dass die Daten über die Datenleitung (SDAT) zwischen dem Programmiergerät und dem PSoC™-Baustein seriell übertragen werden. Die Übertragungsrichtung dieser Verbindung kann sich im Programmierverlauf auf Anforderung des Programmiers ändern, d. h., es handelt sich beim SDAT-Anschluss um einen Ein-/Ausgang. Den Takt für diese Kommunikation stellt ausschließlich das Programmiergerät über die SCLK-Leitung zur Verfügung.

Die Einleitung des eigentlichen Programmiervorgangs kann hier auf zwei unterschiedliche Arten erfolgen, entweder über den „Power-on-Mode“ oder durch eine vorgegebene Reset-Sequenz über den X_{Res}-Pin (Reset-Mode).

Im „Power-on-Mode“ wird zunächst die Betriebsspannung durch das Programmiergerät eingeschaltet. Danach wartet es so lange, bis die Spannung stabil ist und auf der SDAT-Leitung eine negative Flanke auftritt. Dann kann der PSoC™-Baustein initialisiert werden. Da aber das SDAT-Signal nach dem Anlegen der Betriebsspannung sehr verrauscht sein kann, sind Fehler möglich. Daher sollte man zur Initialisierung den „Reset-Mode“ bevorzugen. Hier wird nach der Stabilisierung der Betriebsspannung vom Programmierer ein kurzer Reset-Impuls erzeugt, nach dem der Initialisierungsbefehl an den Mikrocontroller gesendet werden kann. Zu beachten ist hier allerdings, dass einige 8-polige Bausteine nicht über einen Reset-Pin verfügen – sie sind somit allein über den Power-on-Mode programmierbar.

Vorbereitung der Zielschaltung

Die Zielschaltung mit dem PSoC™-Baustein muss für die Verbindung mit dem Programmiergerät so ausgelegt sein, dass die ISSP™-Schnittstelle frei zugänglich ist.



Ansicht des Lieferumfangs

Im besten Fall sieht man eine Stiftleiste vor, auf die der „MiniProg“ direkt aufgesteckt werden kann.

Für erfolgreiche Datenübertragungen sind bei den Verbindungen mit der Programmierschnittstelle einige wichtige Konventionen zu beachten [2], die im Folgenden zusammengefasst sind. Die Datenleitung wird mit P1(0) und die Taktleitung mit P1(1) des PSoC™-Bausteins verbunden. Dies sind Portpins, die im normalen Betrieb der Schaltung als Ein- oder Ausgang (z. B. zum Einschalten einer LED oder Abfragen eines Tasters) zum Einsatz kommen. Die Verbindung dieser Portpins nach Masse darf minimal einen Widerstand von 1 kOhm und maximal eine Kapazität von 120 pF haben (siehe Abbildung 8). Besteht keine Möglichkeit, diese Werte einzuhalten, leistet eine steckbare Unterbrechung (Jumper o. Ä.) gute Dienste.

Das Reset-Signal (X_{Res}) sollte direkt mit der Programmierschnittstelle verbunden werden, um zu gewährleisten, dass der Programmiermodus verlässlich aktiviert werden kann. Oftmals erfolgt der Reset im Normalbetrieb von einem speziellen Reset-Baustein oder durch einen System-Reset. In einem solchen Fall müssen diese Reset-Quellen über einen Widerstand (1 kOhm) entkoppelt werden (siehe Abbil-

- Lieferumfang
des Programmiergerätes:**
- MiniEvaluation Board, das auch als Programmieradapter benutzt werden kann!
 - MiniProgrammer „PSoC™ MiniProg“
 - CY8C29466-24PXI 28pin DIP Sample
 - PSoC™ Designer CD
 - USB-Kabel
 - User Guide

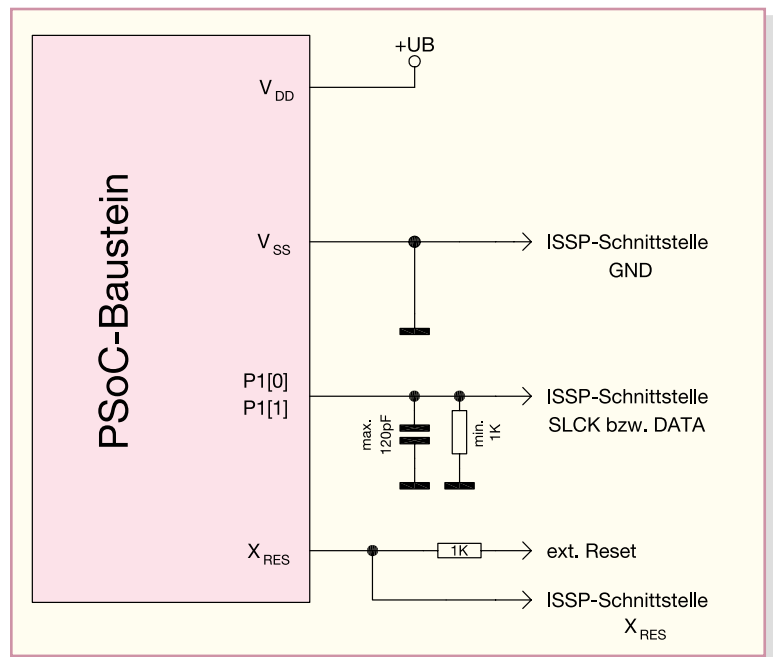
dung 8). Das X_{Res}-Signal bleibt aber trotzdem direkt mit dem entsprechenden Pin des PSoC™-Bausteins verbunden.

Der V_{DD}-Pin sollte nur in dem Fall an die Betriebsspannung des PSoC™ angeschlossen werden, wenn der „Power-on-Mode“ verwendet wird. Der GND-Pin ist in jedem Fall direkt mit der Masse der Zielschaltung zu verbinden.

Das Programmiergerät

Das Programmiergerät „PSoC™ MiniProg“ ist sehr klein und kompakt und unkompliziert in der Anwendung. Bevor man es jedoch einsetzen kann, sind die Entwicklungsumgebung „PSoC™ Designer“ und der „PSoC™ Programmer“ auf dem PC zu installieren. Erst dann wird das Programmiergerät über das mitgelieferte

**Bild 8:
Der
Anschluss
des PSoC™
an die
Programmierschnittstelle**



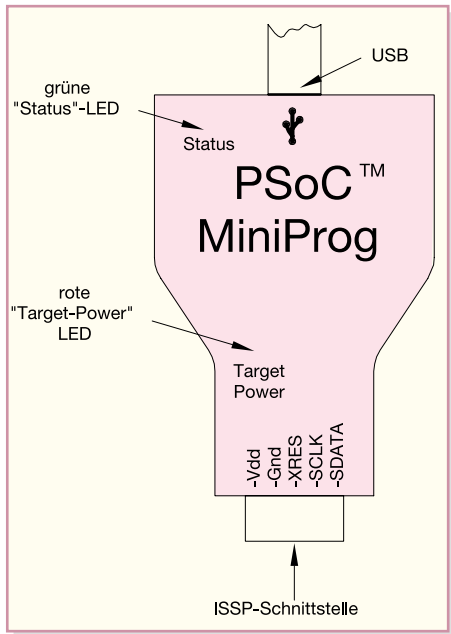


Bild 9: Der PSoC™ MiniProg in der Übersicht. Die Lage der Anzeigen und der Schnittstellen ist hier gut zu sehen.

Jetzt kann man das erstellte Programm über den „Program“-Button in den PSoC™-Baustein des Zielsystems übertragen und direkt testen.

Flexibles System

Die Bausteine sind bis zu 50.000-mal wiederbeschreibbar, so dass man auch kleine Änderungen schnell ausprobieren kann, ohne jedes Mal einen neuen Controller einsetzen zu müssen. So kann man aber auch etwa mit einem „Laborcontroller“ das Programm selbst, aber besonders spätere Änderungen daran, ausgiebig testen, bevor das fertige Programm bzw. dessen Update auf das vielleicht nicht immer für den Entwickler verfügbare Zielsystem übertragen wird. Und jeder Entwickler wird die Möglichkeit schätzen, auf diese Weise sehr schnell Modifikationen und neue Ideen/Features realisieren zu können. Besonders bei der kommerziellen Nutzung bringt dies große Vorteile, kann man doch schnell auf neue Anforderungen reagieren, ohne dass jedes Mal neue Hardware-Kosten entstehen.

Ein mitgeliefertes Testboard, das „MiniEvaluation Board“, vereinfacht den Einstieg in die PSoC™-Welt noch weiter. Auf

dieser kleinen Leiterplatte befinden sich LEDs, ein Potentiometer, ein Taster, eine Stiftleiste zum Aufstecken des MiniProg und ein Sockel für einen PSoC™-Controller. Man kann also einfache Programmierversuche ohne jegliches Löten sofort in einer Schaltung ausführen.

Außerdem kann man dieses Testboard in Verbindung mit dem „MiniProg“ als „Stand-alone“-Programmiergerät für PSoC™-Bausteine verwenden, um diese im o. g. Sinne ohne die entsprechende Zielschaltung beschreiben zu können.

Insgesamt hält der Schaltungsentwickler mit dem PSoC™-System also ein äußerst universell einsetzbares und effizientes Entwicklungswerkzeug in der Hand, das den rein softwaremäßigen Entwurf extrem kompakter Schaltungsdesigns mit implementierter Intelligenz gestattet und dabei eine hohe Flexibilität an den Tag legt. **ELV**

Kabel an einen freien USB-Port des PCs angeschlossen. Ist diese Verbindung in Ordnung, beginnt die grüne „Status“-LED neben der USB-Buchse des MiniProg zu blinken. Sollte der USB-Treiber im Zuge der Installation der Programmer-Software nicht korrekt installiert worden sein, so bleibt diese LED dunkel.

Eine weitere LED („Target Power“) zeigt an, ob am V_{DD}-Ausgang des Programmiergerätes eine Spannung ausgegeben wird, die das Zielsystem vom Programmiergerät aus versorgt.

Jetzt kann die ISSP™-Schnittstelle des MiniProg an die vorbereitete Zielschaltung angeschlossen werden.

Die Lage der LEDs und der Schnittstellen sind aus Abbildung 9 entnehmbar.

Mit der zugehörigen Entwicklungsumgebung, dem „PSoC™ Designer“, werden jetzt die gewünschten Konfigurationen der programmierbaren Analog- und Digitalblöcke vorgenommen sowie das Programm in „Assembler“ oder der Programmiersprache „C“ (optional) erstellt, um somit die Funktionen der Zielschaltung zu definieren. Die Vorgehensweise hierbei ist im ersten Teil dieses Artikels bereits ausführlich erläutert worden.

Die Programmiersoftware (Abbildung 10) ist jetzt aus dem „PSoC™ Designer“ heraus über das Menü „Program → Program Part“ zu starten, dann werden die notwendigen Konfigurationen vorgenommen:

- Port → Programmiergerät auswählen (hier: MiniProg1)
- Device Family → Baustein-Familie (z. B. 27x43)
- Device → Baustein (z. B. CY8C27443)
- Programming mode → Programmiermodus (s. o.)
- Power Device → Spannungsversorgung am V_{DD}-Pin ein-/aus-schalten

Internet:

- [1] In-System Serial Programming (ISSP)™ Protocol
[http://www.cypress.com/
Application note „AN2026a“](http://www.cypress.com/Application_note_„AN2026a“)
- [2] Design for In-System Serial Programming (ISSP)™
[http://www.cypress.com/
Application note „AN2014“](http://www.cypress.com/Application_note_„AN2014“)

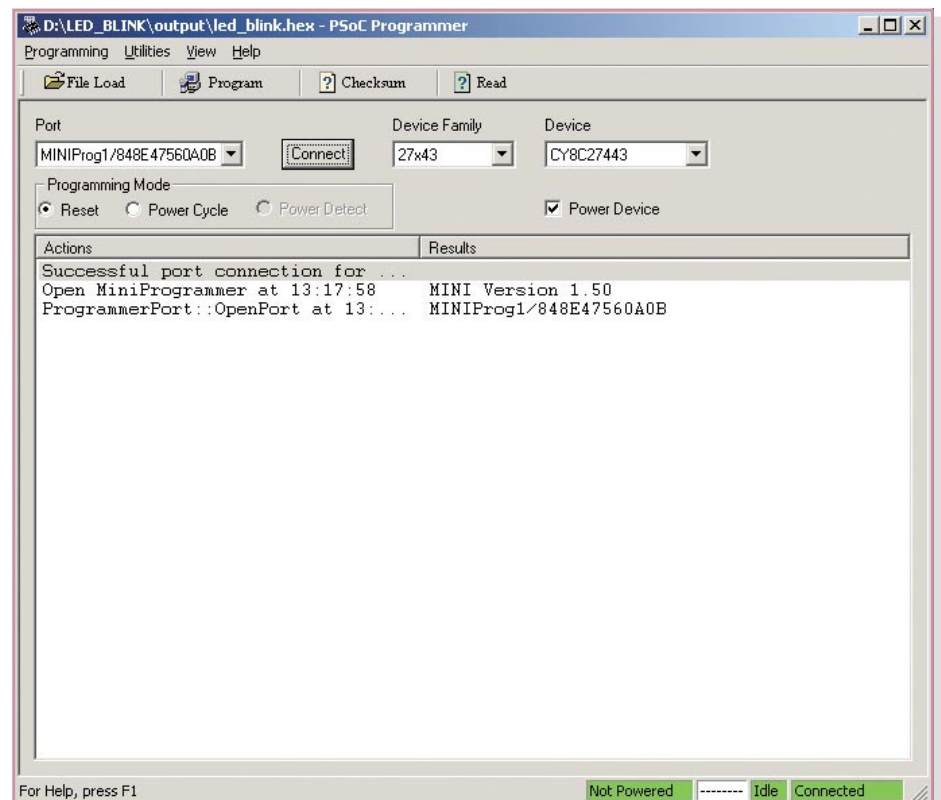


Bild 10: Das Hauptfenster der PSoC™-Programmiersoftware