

CPUに依存しない開発スタイルを身に付ける

わかって書けば移植性抜群!

歴代の本誌付属
CPUボード全対応!

GCCプログラミング術

第1回 アセンブリ言語は極力使わない! ベクタ・テーブル&
スタートアップ・ルーチンはCで書く

村井 和夫

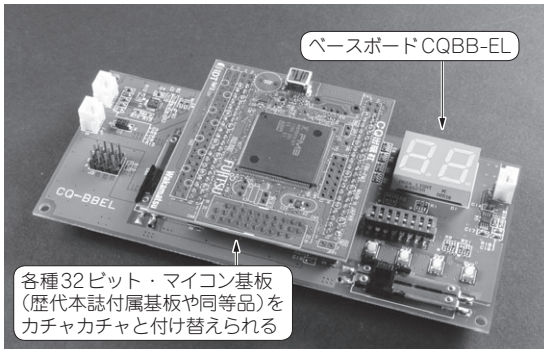


写真1 実験に使ったベースボードCQBB-EL…主要な32ビット・マイコン基板を差し替えられるので今回の実験にピッタリ!
開発元: (株)イーエスピー企画
<http://www.esp.jp/product/detail/cq-bbel.html>
販売元: (株)若松通商
<http://www.wakamatsu-net.com/cgibin/biz/pageshousai.cgi?code=38230043&CATE=3823>

歴代の本誌付属基板を挿し換え可能
(SH-2, V850, FR60, ARM7, SH-2A, RX62N, Cortex-M3)

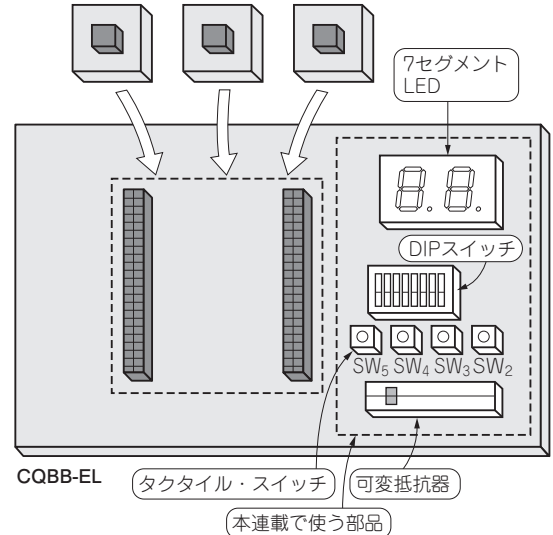


図1 本連載で目指すこと…主要な32ビット・マイコンすべてで(なるべく)共通に動くユーザ・プログラム作りに挑戦してみる

本誌2013年11月号～2014年2月号にかけて連載した「GCCマルチ開発環境の構築」では、複数種類のCPUに対応可能な開発環境GCCの構築方法について解説しました。ここで準備した開発環境を使って、本連載では、アセンブリ記述を最小限にして、CPU依存部を局所化することにより、移植性の高い組み込みソフトウェアを開発してみます(図1)。

実験のターゲット基板としては、CQBB-ELを使います(写真1)。本誌はこれまで、SH-2A, RX62N, V850, ARM7, Cortex-M3などの主要32ビット・マイコン搭載基板を付録にしてきましたが、それらがすべて動かせるので本実験にピッタリです。

題材とする制御プログラムの概要

まず、題材として使用する基板の概要と制御プログラムの概要、ファイルの構成について説明します。制御プログラムのファイルcqbbe1.7zは、本誌のウェブ・サイトからダウンロードできます。

● 実験基板の概要と制御プログラムの仕様

実験基板には図1に示す部品が実装されています。制御プログラムの仕様は以下の通りです。実験基板に搭載されている主要なI/Oと割り込みの確認ができるように、このような仕様としました。

- 電源投入により、DIPスイッチの現在値を、7セグメントLEDの表示の初期値とする。
- SW₂押下のエッジ割り込みにより、状態を「カウントアップ状態」または「カウント停止状態」に交互に変更する。
- タイマ割り込みにより、0.5秒ごとに、7セグメントLEDの表示を更新する。状態が「カウント停止状態」であれば値を点滅表示させ、「カウントアップ状態」であればカウントアップ表示を行う。電源投入時の初期値は「カウント停止状態」である。
- SW₃押下のエッジ割り込みにより、7セグメントLEDの表示値を0にして、状態を「カウント停止