

ARMプロセッサで使える汎用JTAGデバッガを自作する

山崎尊永

本稿では、無償ツールと市販部品を使って汎用のARM用JTAGデバッガを自作する。GNUベースなので、生成コードの制約を気にしないで本誌2008年5月号付属ARM基板のプログラムが開発できるほか、さまざまなARMプロセッサにも利用できる。RS-232-CやUARTインターフェースも同時に作成し、開発のさまざまな局面で役立てられる。(編集部)

本誌2008年5月号にSTMicroelectronics社のマイコンSTM32F103を搭載した基板が付属しました。このマイコンは、128Kバイトのフラッシュ・メモリと20KバイトのSRAMを内蔵し、コアとして72MHz動作のARM Cortex-M3を採用したものです。従来のARMマイコンに比べ、Thumb-2という命令体系によりコード効率改善され、割り込み応答性能も良くなっています。せっかく高性能なマイコン基板を入手したので、何かしっかり開発してみたいのが人情というものですね。

ところが、5月号に付属した開発ツールはいずれも評価版で、生成できるコードが16Kバイトや32Kバイトまでに制限されています。せっかくの128Kバイトの内蔵フラッシュ・メモリを生かせません。また、DFU(Device Firmware Update)機能により手軽にフラッシュ・メモリへプログラムをダウンロードできますが、開発環境側からフラッシュ・メモリ上のプログラムをデバッグできません。

コード・サイズ制限は、GNUのGCCを使えば突破できますが、フラッシュ・メモリ上のプログラムをデバッグするにはJTAGデバッガが欲しくなります。本稿では、筆者が製作した図1に示すようなJTAGデバッガ、名付けて

「ARM OCD (On Chip Debugger) myJTAG」を紹介します。また、統合開発環境Eclipseによるデバッグ環境の立ち上げについて解説します。パソコン上のツールはすべて無償で手に入るものを使います。製作に使用した電子部品はすべて東京・秋葉原などの部品販売店で手に入ります。

1. USB接続型JTAGデバッガの製作

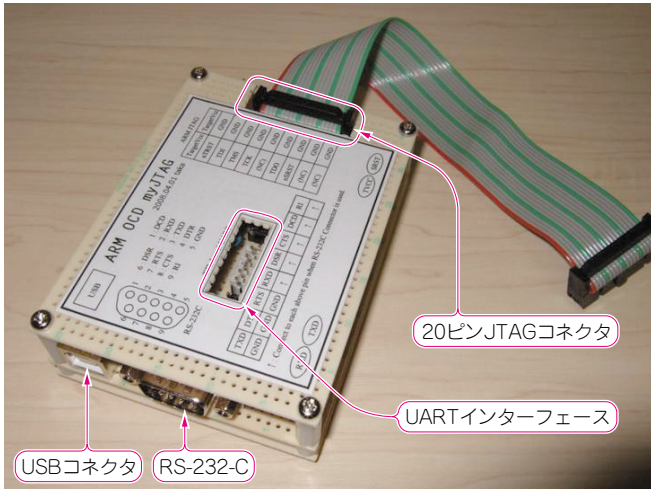
まずは、USB接続型JTAGデバッガ「ARM OCD my JTAG」を製作しましょう。USBからJTAGに変換するデバイスとしてデファクトなのが英国Future Technology Devices International (FTDI)社のFT2232です。これを使ってUSB-JTAG変換機能を実現します。フリーのデバッガ・ソフトウェアとしてはスタンダードな「Open On-chip Debugger」がすでに対応しており、簡単にコア内蔵のオンチップ・デバッガ機能を使えます。

なお、FT2232は2チャンネルを備えています。USB⇔RS-232-C変換機能もおまけで作り込み、TTLレベルのUART信号の接続もできるようにしました。開発の際、なにかと重宝すると思います。試作機の仕様を表1に示します。

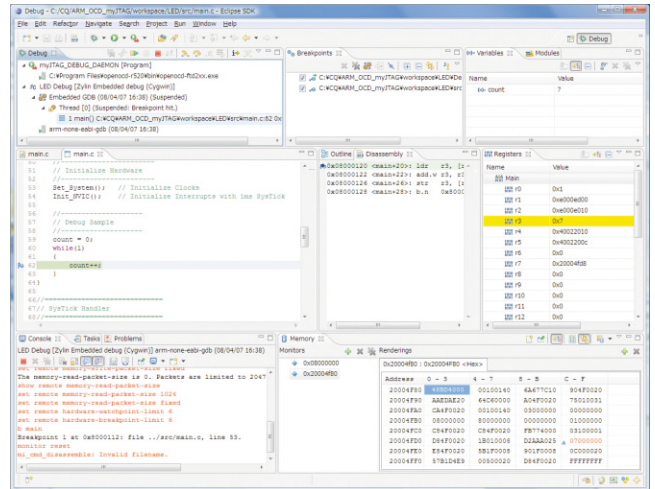
本機は今回のCortex-M3だけでなく、ARM系コアを持つマイコンのほとんどをデバッグできる汎用性の高いものです。1台作っておくと、今後役立つと思います。本機と同等なRS-232-Cポートも含む市販のJTAGデバッガは1万円くらいしますが、自作すると半分以下の値段で完成します。

Keyword

JTAGデバッガ, USB, FT2232, ARM, Cortex-M3, STM32, RS-232-C, SP3243, GNU, GCC, GDB, Open On-chip Debugger, Eclipse



(a) 試作したUSB⇔JTAGインターフェイス「ARM OCD myJTAG」



(b) 統合開発環境Eclipse

図1 「ARM OCD myJTAG」と統合開発環境Eclipse

(a) は手前左側にUSBコネクタがありパソコンと接続する。USB-JTAGインターフェイス機能により、ARM系マイコン標準の20ピンJTAGコネクタに接続できる。USB⇔RS-232-C変換機能も備えており、9ピンのDサブコネクタが見える。上面中央の四角い穴は、TTLレベルUARTインターフェイスである。RS-232-C/UARTは、JTAGと同時使用できる。(b)はオープン・ソースの統合開発環境である。GNU gccとOpen On-Chip Debuggerをベースにして開発する。この環境の上で、ソース・コードの編集、ビルド、フラッシュ・メモリへのダウンロード、フラッシュ・メモリ上のプログラムのソース・レベル・デバッグを行うことができる。

● 部品を集めて製作しよう

本機の回路図を図2に示します。また部品表を表2に示します。部品はすべて東京・秋葉原で入手できました。FT2232は現在、FT2232Dが最新バージョンですが、筆者が入手できたFT2232Lでも全く問題なく使えます。

多くのオープン・ソースの開発ツールは、スイスAmontec社のJTAGデバッグ「JTAGkey」をサポートしています。そこで本機も、FT2232のGPIO経由のリセット関連信号などの論理的な接続方式はJTAGkeyと互換性を持たせてあります。

使用するコネクタのピン配置を写真1と写真2に示します。また本機の外観を写真3と写真4に示します。

U2(シリアルEEPROM)を除くLSIはすべて表面実装タイプなので、ピッチ変換基板を使っています。1.27mmピッチ、0.65mmピッチ、0.5mmピッチと3種類あります(p.78のコラム「表面実装LSIのはんだ付け」を参照)。

USBコネクタCN1とU1 FT2232との間のUSB信号DM/DPの配線は極力短くなるようにします。

JTAG信号は、74VHCタイプの標準ロジックICを経由させます。74VHCタイプはその電源電圧よりも高い入力電圧を印加でき、レベル変換ができます。U4とU5はターゲット基板の電源TV_{CC}を印加しますが、ターゲット基板の電源TV_{CC}は3V系でも5V系でもOKです。本機内にはこのTV_{CC}とUSBケーブルから供給されるV_{CC}(+5V)の2系統の電源

表1 「ARM OCD myJTAG」の仕様

No.	項目	内容
1	使用デバイス	FTDI社 FT2232
2	USB⇔JTAG インターフェース	ARM標準JTAGコネクタ(20ピン)直結可能 TRST/SRSTフル・サポート ターゲット電源検知 ターゲット状態表示LED×2個 (Amontec社JTAGkey準拠)
3	USB⇔RS-232-C インターフェース	Dサブ9ピンコネクタ フル・モデム信号サポート 送受信状態LED×2個 TTLレベルUART信号入出力可能 JTAGと同時利用可能
4	USB	USB 2.0フル・スピード(12Mbps) バス・パワー動作(50mA以下)
5	デバッグ	GDB + Open On-chip Debugger

があるので、間違えないように配線してください。

外部のTTLレベルのUART機器との接続は、SW1のピン・ヘッダ経由となりますが、外部からの信号がU6 SP3243(RS-232Cレベル変換IC)と衝突しないようにジャンパを使っています。TTLレベルUARTを使う場合は、ジャンパ・ピンを取り外します。なお、U6(SP3243)は、/SHUT DOWNを“L”レベルにすると出力信号がハイ・インピーダンスになります。これを使って信号衝突を防ぐ方法もあるので、工夫してみてください。

本機はパソコンやターゲット基板に接続するものなので、相手機器を壊さないため、配線は念入りにチェックしてください。特に電源配線は間違いのないようにしてください。念のため、USBから供給されるV_{CC}ラインには、100mA

