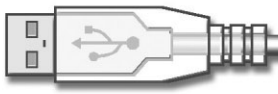

第4章 ARMマイコン基板のCPU内蔵フラッシュROMおよびSRAMへのプログラム・ダウンローダ
マスストレージ・クラスを応用したセカンダリ・ブート・ローダの移植

芥川 克巳

オランダNXP Semiconductors社はUSBセカンダリ・ブート・ローダのサンプル・コードをWebサイト上で提供している．ホスト環境にARMマイコンを接続すると，マスストレージとして認識されフォルダが開く．そこに実行したいユーザ・プログラムをドラッグ&ドロップでコピーしてからARMマイコンを再起動させると，コピーしたユーザ・プログラムの実行が開始されるというブート・ローダである．（筆者）

1. USBセカンダリ・ブート・ローダの概要

● セカンダリ・ブート・ローダとは何か

LPC2388は，ARMマイコン・コアだけでなくフラッシュROMも一つのチップに内蔵しています．そのため，製品のアップデートや不具合修正を行うために，チップ内蔵のフラッシュROMへプログラムを書き込むことが可能です．

チップ内蔵のフラッシュROMへプログラムを書き込むためには，一般的に以下の二つの方法が提供されます．

●In-System Programming（ISP）：ブート・ローダ・ソフトウェアとUART0シリアル・ポートを使用してチップ内のフラッシュROMにプログラムを書き込む方法です．この方法は，ユーザのハードウェア・プラットホーム上にUART0を経由してプログラムを行う機能がある場合に実現可能です．

●In Application Programming（IAP）：エンド・ユーザが作成したアプリケーション・コードでチップ内フラッシュROMの消去や書き込みを行います．セカンダリ・ブート・ローダは，プライマリ・ブート・ローダ（チップ内）で使用される標準UART0以外のほかのチャネルを使用してフラッシュROMへプログラムを書き込みます．

セカンダリ・ブート・ローダは，USBやEthernet，SPI，SSP，CAN，I^2Cなどのデバイスを使用して内蔵フラッシュROMにプログラムを書き込めます．セカンダリ・ブート・ローダは，エンド・ユーザが作成したプログラムをアップデートする方法としてIAPを使用します．

今回はセカンダリ・ブート・ローダの開発でUSBを使用します．以降の各項目では，USBセカンダリ・ブート・ローダの実装について説明を行います．

2. ブート・ローダ設計検討

どのようなブート・ローダにでも，ユーザはセカンダリ・ブート・ローダのために適切なブート・ローダのエントリ・メカニズムを選択する必要があります．それは，専用のハードウェア端子やソフトウェア・ハンドシェークによって行われます．

電源投入時に，セカンダリ・ブート・ローダはエントリ・メカニズムをチェックする必要があります．エントリ・メカニズムが有効である場合，セカンダリ・ブート・ローダが実行されます．エントリ・メカニズムが有効でない場合，処理はユーザ・コードの実行アドレスへジャンプします．通常，プライマリのオンチップ・ブート・ローダは，リセットおよび電源投入後に実行されます．

しかし，プライマリ・ブート・ローダのエントリ・メカニズムは，LPC2388内蔵のCRP（Code Read Protection）を使用して動作をさせないようにすることもできます．

言い替えれば，CRPを設定することでプライマリ・ブート・ローダがデフォルトISPでバイパスされ，リセットおよび電源投入後直ちにセカンダリ・ブート・ローダが実行されるのです．

● エントリ・メカニズム

●専用のハードウェア

セカンダリISPブート・ローダは，エントリが有効かど