

I²C[®]によるコードの更新

著者：Aude Richard

はじめに

ADuC702xファミリーには、インサーキット・プログラミング機能があります。標準モデルではUARTインターフェースをこの機能に使用しますが、ADuC7019BCPZ62I、ADuC7020BCPZ62I、ADuC7021BCPZ62I、ADuC7026BSTZ62IのIモデルではインサーキット・プログラミング機能にI²Cインターフェースを使用します。

標準モデルもIモデルも、リセット時にダウンロード・モードに入るにハードウェア・スイッチが必要です（アプリケーション・ノートAN-724とAN-806を参照）。

Iモデルでは、図1に示すように、ソフトウェア・スイッチも使用してダウンロード・モードに入ります。

アプリケーションによっては、P0.0上でハードウェア・スイッチを使用できないことがあります。それでもI²C経由のソフトウェア更新機能は必要です。このアプリケーション・ノートでは、P0.0をグラウンドに固定接続した状態でソフトウェアを更新する方法を説明します。

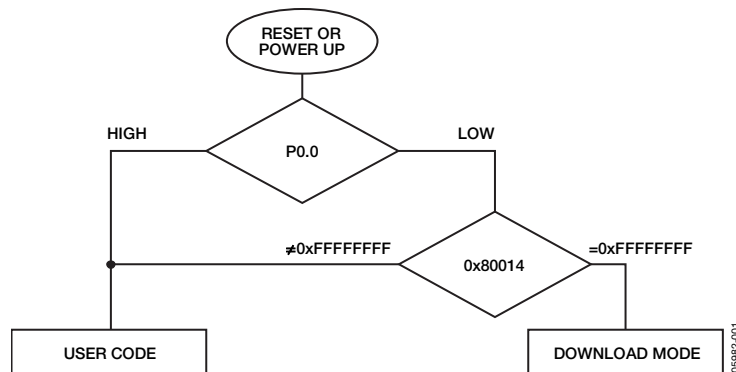


図1. ダウンロード・モードに入る方法

リスク

P0.0をグラウンドに接続するという事は、次のようにアドレス0x80014の内容のみに依存することを意味します。

- アドレス0x80014をクリアすると (=0xFFFFFFFF)、リセット後にダウンロード・モードに入ります。カーネルがJTAGのアクセスから保護されているため、JTAGを使用することはできません。ローダは、I2C0バス (P1.0とP1.1) 上のコマンドを待ちます。ローダが実行コマンドを受信しないと、ユーザ・コードは実行されません (詳細についてはAN-806を参照)。
- アドレス0x80014がクリアしないと、ユーザ・モードに入ります。アドレス0x80014が消去され、電源サイクリングまたはリセットが行われないと、ダウンロード・モードに入れません。

I²CローダのテストとIモデルのコード・デバッグ

I²CからADuC702xにコードをダウンロードするときは、アドレス0x80014を消去する必要があります。I²Cを介してデバイスのコードを常時更新できるようにするためには、次のいずれかを行います。

- アドレス0x80014を常に0xFFFFFFFFにします。そのために、スタートアップ・コードを変更し、I2CWSRを使用してI²Cからコードをダウンロードします。
- JTAGを使用して、フラッシュ/EEメモリを消去します。
- ユーザ・コード内でプロトコルを定義して、フラッシュ/EEメモリのページ0を消去するようにします。

ここでは、最初の2つの方法について説明します。3番目の方法は、「I²Cによるコードの更新」で説明します。

表1. スタートアップ・コードの変更¹

GNU Compiler		Keil Compiler		Address
Vectors		Vectors		
LDR	PC, Reset_Addr	LDR	PC, Reset_Addr	Address 0x80000
LDR	PC, Undef_Addr	LDR	PC, Undef_Addr	Address 0x80004
LDR	PC, SWI_Addr	LDR	PC, SWI_Addr	Address 0x80008
LDR	PC, PAbt_Addr	LDR	PC, PAbt_Addr	Address 0x8000C
LDR	PC, DAbt_Addr	LDR	PC, DAbt_Addr	Address 0x80010
.word	0xFFFFFFFF	DD	0xFFFFFFFF	Address 0x80014
LDR	PC, IRQ_Addr	LDR	PC, IRQ_Addr	Address 0x80018
LDR	PC, FIQ_Addr	LDR	PC, FIQ_Addr	Address 0x8001C

¹ 太字のアドレスは変更が必要です。

スタートアップ・コードの変更

最初の方法では、スタートアップ・コードを変更し、アドレス0x80014を0xFFFFFFFFに強制設定します (表1を参照)。こうすることで、フラッシュ/EEメモリ全体を消去せずに、必要な回数だけダウンロードを行うことができます。この方法では、ハードウェア・スイッチを使用するか、リセット/電源サイクリングのために新しいコードをダウンロードする必要があります。

一括消去コマンドの使用

もう1つの方法では、JTAGを使用してフラッシュ/EEコントローラで一括消去コマンドを入力します。リセット/電源サイクリングのためにユーザ・コードを入力するためにアドレス0x80014を消去しないでください。この方法は開発段階でデバッグを行うときに便利ですが、アプリケーションによっては使用できない場合があります。

次の手順に従って、一括消去コマンドを使用します。

1. JTAGを使ってデバッグ・モードに入り、Flashペリフェラル画面を開きます。
2. 次の順に、一括消去コマンドを入力します。

```
FEEDAT = 0x3CFF;
FEEADR = 0xFFC3;
FEEMOD = FEEMOD | 0x8;
FEEDCON = 0x06;
```

I²C経由でのコードの更新

ユーザ・コードの一部としてI²Cプロトコルを実装して、ページ0を消去することができます。専用アドレスまたはテーブル中の専用の場所を選択してダウンロード・モードに入るフラグとして使用します。SFP参考デザイン・コードに基づき、コンパニオン・コードは0xA0テーブルのアドレス0xFF (255) を使用します。

ホストがコマンドA0 W 0xFF 0x01 Sを送信すると、ページ0が消去され、ソフトウェアのリセットが行われます。アドレス0x80014が消去されると、P0.0がグラウンドに固定接続されているため、自動的にダウンロード・モードになります。これは、コンパニオン・コードのCコードのNoProt.cセクションに表示されます。

サンプル・コードの使用

1. I2CWSDを使用して、NoProt.hexをダウンロードします。
2. Runをクリックします。

ユーザ・コードの更新

1. I²CからコマンドA0 W 0xFF 0x01 Sを送信します。図2に示すように、自動的にダウンロード・モードに入ります。
2. I2CWSDを使用して、更新されたコードをダウンロードします。

この方法では、ページ0が誤って消去されることがあるので注意してください。

フラッシュ保護を用いるもう1つの方法については、「プロトコルの保護の追加」を参照してください。

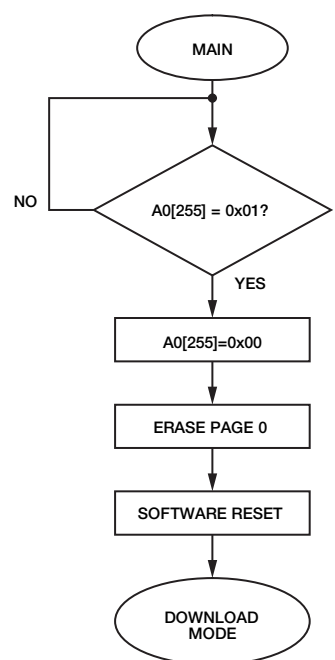


図2. フローチャート

049882-002

プロトコルの保護の追加

プロトコルを強化するために、ページ0を保護できます。フラッシュ/EEメモリの保護は簡単に追加できます。保護を有効にするには、電源サイクリングを行う必要があります。コンパニオン・コードのCコードのProt.cセクションに表示されます。

この例では、以下のアドレスを使用します。

A0[251] FEEADRのMSB
 A0[252] FEEADRのLSB
 A0[253] FEEDATのMSB
 A0[254] FEEDATのLSB
 A0[255] ステータス・フラグ

フラグのステート：

A0[255]=0 (デフォルト設定)
 A0[255]=1はページ0の消去を意味します。
 A0[255]=2はページ0を保護することを意味します。
 A0[255]=3はページ0を保護しないことを意味します。

サンプル・コードの使用

1. I2CWSOを使用して、Prot.hexをダウンロードします。
2. ページ0の保護を行うときは、I²CからコマンドS A0 W FF 02 Pを送信します (図3のメッセージ2を参照)。
3. デバイスの電源サイクリングを実行して、保護を有効にします。

ユーザ・コードの更新

1. ページ0の保護を行わないときは、I²CからコマンドS A0 W FB 12 34 A5 A5 03 Pを送信します (図3のメッセージ4を参照)。
2. デバイスの電源サイクリングを実行します。
3. ページ0を消去するときは、I²CからコマンドS A0 W FF 01 Pを送信します (図3のメッセージ5を参照)。
4. I2CWSOを使用して、更新されたコードをダウンロードします。

コードがどのように強化されたかをチェックするには、ページ0の保護を解除せずにページ消去コマンドを送信します。ユーザ・コードは正常に実行されます。

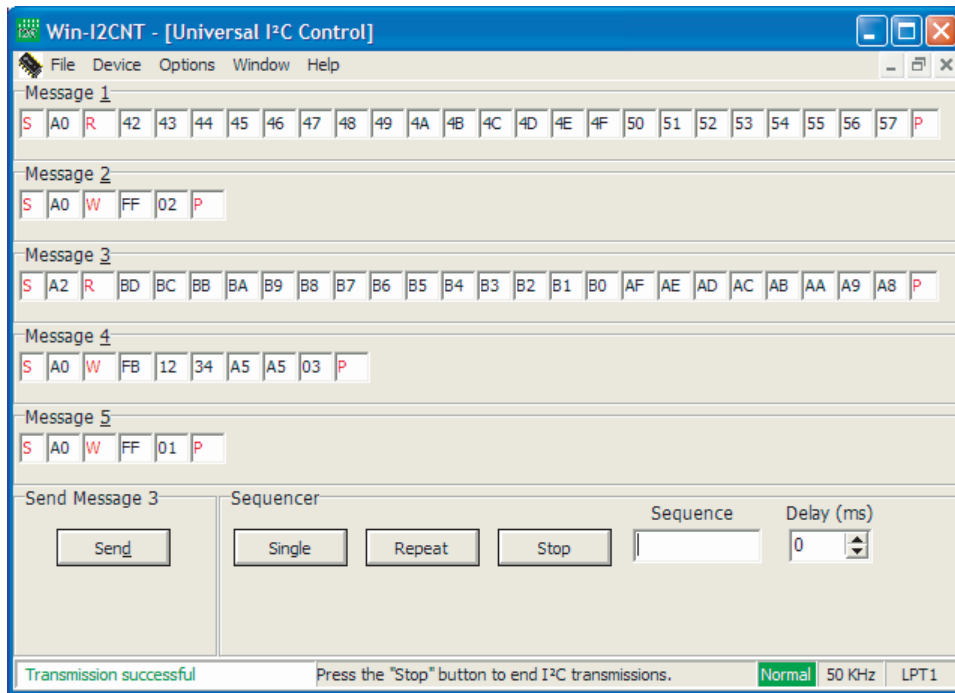


図3. ダウンロード・モードに入る方法