

序説

この応用記述は低容量ブートローダを通してAtmel[®] ATtiny104の自己プログラミング機能を実演します。Atmel ATtiny104はAVR[®]強化RISC構造に基づく低電力CMOS 8ビット マイクロ コントローラです。

この応用記述と共に入手可能なzipファイルはブートローダhexファイル、zipファイルとしてのプロジェクト、Pythonスクリプトファイル、例応用のhexファイルを含みます。このブートローダは応用コードを受け取るための通信インターフェースとしてUSARTを使用します。PC上で走るPythonスクリプトはCOMポートを使用して応用コードを送るのに使用されます。

特徴

- 低容量のためのアセンブリ言語コード
- デバイスにhexファイルを送るためのPythonスクリプト
- データ転送のためのUSART

目次

序説	1
特徴	1
1. 事前必要条件	3
2. ATtiny104での自己プログラミング	3
3. ブートローダ操作	3
3.1. Pythonスクリプト	4
3.2. ブートローダを使う方法	4
4. 参照	4
5. 改訂履歴	4

1. 事前必要条件

- ATtiny104デバイスとATtiny104 Xplained Nanoキットの熟知
- <https://www.python.org/downloads/>で入手可能なPython 2.7.8がPCにインストールされるべきです。
- Pythonアドオン単位のpyserialがPCにインストールされるべきです。 <https://pip.pypa.io/en/latest/installing/>で提供される段階に従うことによってpipツールをインストールしてください。このpipツールはpip install pyserial命令を実行することによってpyserialをインストールするのに使用することができます。
- Atmel Studio 7.0.790またはそれ以降版
- 自己プログラミングを許可するためにATtiny104デバイスのSELFPROGENヒューズは'0'を書かれるべきです。これはAtmel StudioでメニューのTools⇒Device Programming⇒Fusesを使用して行うことができます。

2. ATtiny104での自己プログラミング

ATtiny104では、フラッシュメモリがデータメモリ空間に割り当てられます。割り当てられたプログラムメモリはデータメモリ空間のバイトアドレス\$4000で始まります。プログラムはプログラムメモリ空間のアドレス\$0000から開始して実行されますが、データメモリ空間経由でアクセスされる時は\$4000から始まるアドレスで指定されなければなりません。アプリケーションは関節アドレス指定を持つ取得(load)命令を使用してデータメモリ空間の割り当てられた位置からNVMを読むことができます。フラッシュメモリはデータメモリ空間の割り当てられた位置から一度に1バイトを読むことができます。読み込み動作について、語アドレス内の上位または下位のバイトを選ぶのに最下位ビット(ビット0)が使用されます。このビットが0の場合、下位バイトが読まれ、1の場合は上位バイトが読まれます。

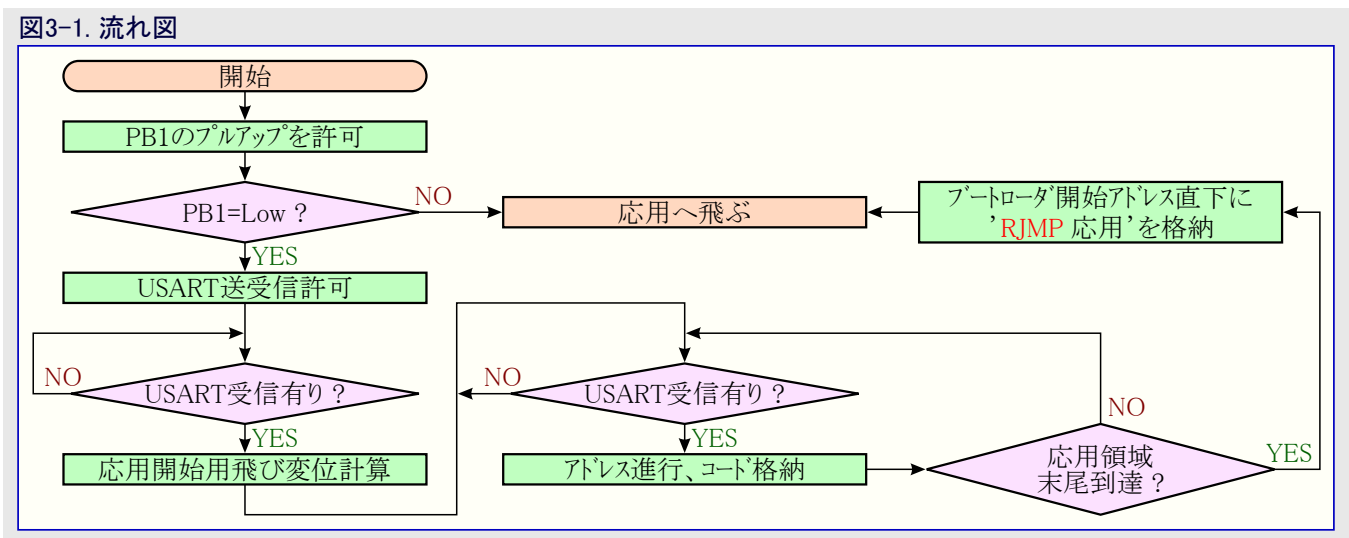
フラッシュメモリは語単位で書かれます。フラッシュメモリの語を書く前に、フラッシュメモリの目標位置が消去されなければなりません。未消去のフラッシュメモリの語書き込みはその内容が不正になるでしょう。チップ消去開始に先立って、不揮発性メモリ指令(NVMCMD)レジスタはチップ消去(CHIP_ERASE)指令を設定されなければなりません。消去動作を開始するには、フラッシュコード領域に属する語位置の上位バイトに偽装バイトが書かれなければなりません。

フラッシュメモリは書き込みに対して語アクセスされ、データ空間はデータメモリ空間に割り当てられたフラッシュメモリをアクセスするのにバイトアドレス指定を使用します。従ってフラッシュメモリに正しい順、即ち上位バイトの前に下位バイトを、で語を書くことが重要です。最初に、アドレスに下位バイトが書かれ、それは一時緩衝部に格納されます。その後、上位バイト書き込みがフラッシュ語緩衝部へ上位バイトと下位バイトの両方を設定し、フラッシュメモリへの書き込み操作を開始します。正しいCPU動作を保証するため、自己プログラミングを起動するST命令の後には2つのNOP命令が挿入されるべきです。

3. ブートローダ操作

このブートローダはアプリケーションコードを受け取るのにUSARTを使用します。この応用記述と共に入手可能なPythonに基づくツールはhexファイルをbinファイルに変換し、各バイトをATtiny104デバイスに送ります。デバイス上で走行するブートローダはやって来るバイトに対してUSART周辺機能のポーリングを保ち、それらを語単位でフラッシュメモリに書き込みます。

次図はブートローダの流れ構成図を示します。



注: 主応用の開始アドレスはPythonスクリプトからブートローダへ送られます。その後、ブートローダは無条件分岐を実行するための変位を計算してブートローダの開始直下の位置にRJMP命令を配置します。アプリケーションコードがフラッシュメモリに書き込まれた後、ブートローダはブートローダの開始直下の位置へ飛んでアプリケーションへの無条件分岐を実行します。

3.1. Pythonスクリプト

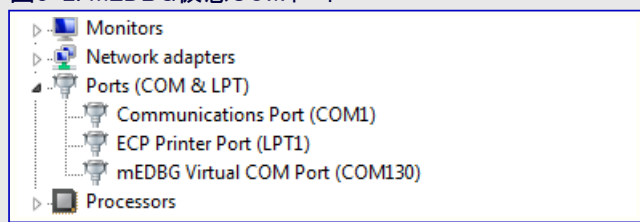
この応用記述と共に供給されるPythonスクリプトは、`sys`、`pyserial`、`argparse`の3つのアドイン単位部を使用します。それは応用hexファイルを解析し、指定されたCOMポート経由でコードをブートローダに送ります。それがフラッシュメモリの4語を保存する可能性を与えるため、このブートローダで使用されるボーレートは62500です。これはデバイスが1MHzの既定クロック周波数で走行する時に可能です。この場合、USARTボーレートレジスタ(`UBRRn`)を既定値(`$00`)に保つことは概ね0%誤差で62500のボーレートを与えます。ATtiny104で走行するブートローダは各バイトをPCに送り返します。受信したバイトが送ったバイトと同じでない場合、Pythonスクリプトは異常を表示して終了します。

3.2. ブートローダを使う方法

ATtiny104 Xplained Nanoでブートローダを使うには、

- 書き込み器としてmEDBGを使用してAtmel Studio経由でATtiny104デバイスにブートローダhexファイルを設定して(書き込んで)ください。
- SW0**がブートローダ移行ピンです。ブートローダを活性にするために**SW0**を押して基板を電源OFF/ONしてください。
- ATtiny104 Xplained Nano用の仮想COMポートを確認してください。これはWindows® PC上の**デバイス マネージャ**内で**mEDBG Virtual COM Port (COMxxx)**として一覧されます。

図3-2. mEDBG仮想COMポート

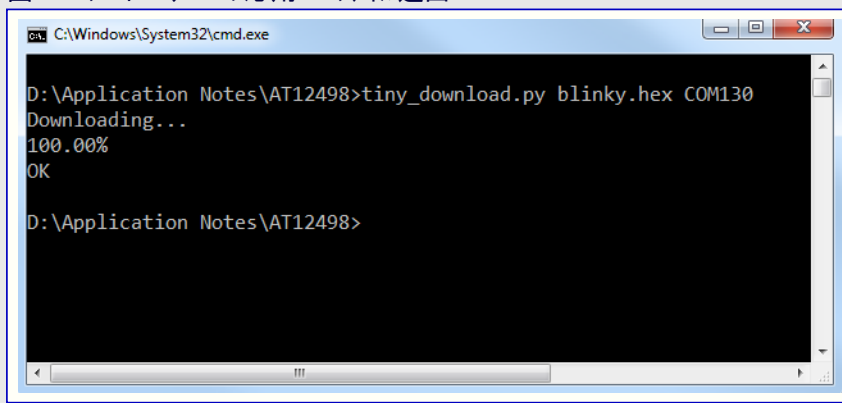


注: mEDBGによって提供される仮想COMポートは基板上のATtiny104のUSARTピン(PB2-TXDとPB3-RXD)に接続されます。

- 以下のようにコマンドプロンプトから**tiny_download.py**を走らせてください。ダウンロード(転送/書き込み)の進捗がコマンドプロンプトで表示されます。この応用記述と共に提供される応用例(**blinky.hex**)は固定間隔でATtiny104 Xplained Nano上のLEDを点滅します。

- `tiny_download.py blinky.hex COMxx`

図3-3. ブートローダへの応用hexファイル送出



- ダウンロード(転送/書き込み)成功時、応用が走行を開始します。これはATtiny104 Xplained Nano上でLEDが点滅することによって示されます。

4. 参照

- ATtiny104データシート
- Atmel AT12489:Atmel ATtiny102とATtiny104での開始に際して
- AVRアセンブラ命令
- ATtiny104 Xplained Nano

5. 改訂履歴

資料改訂	日付	注釈
42736A	2016年6月	初版資料公開

Atmel®, Atmelロゴとそれらの組み合わせ、Enabling Unlimited Possibilities®, AVR®, QTouch®とその他は米国及び他の国に於けるAtmel Corporationの登録商標または商標です。Windows®は米国及び他の国に於けるMicrosoft Corporationの登録商標です。ARM®はARM Ltdの登録商標です。他の用語と製品名は一般的に他の商標です。

お断り: 本資料内の情報はAtmel製品と関連して提供されています。本資料またはAtmel製品の販売と関連して承諾される何れの知的所有権も禁反言あるいはその逆によって明示的または暗示的に承諾されるものではありません。Atmelのウェブサイトに位置する販売の条件とAtmelの定義での詳しい説明を除いて、商品性、特定目的に関する適合性、または適合性の暗黙保証に制限せず、Atmelはそれらを含むその製品に関連する暗示的、明示的または法令による如何なる保証も否認し、何ら責任がないと認識します。たとえAtmelがそのような損害賠償の可能性を進言されたとしても、本資料を使用できない、または使用以外で発生する(情報の損失、事業中断、または利益と損失に関する制限なしの損害賠償を含み)直接、間接、必然、偶然、特別、または付随して起こる如何なる損害賠償に対しても決してAtmelに責任がないでしょう。Atmelは本資料の内容の正確さまたは完全性に関して断言または保証を行わず、予告なしでいつでも製品内容と仕様の変更を行う権利を保留します。Atmelはここに含まれた情報を更新することに対してどんな公約も行いません。特に別の方法で提供されなければ、Atmel製品は車載応用に対して適当ではなく、使用されるべきではありません。Atmel製品は延命または生命維持を意図した応用での部品としての使用に対して意図、認定、または保証されません。

安全重視、軍用、車載応用のお断り: Atmel製品はAtmelが提供する特別に書かれた承諾を除き、そのような製品の機能不全が著しく人に危害を加えたり死に至らしめることがかなり予期されるどんな応用(“安全重視応用”)に対しても設計されず、またそれらとの接続にも使用されません。安全重視応用は限定なしで、生命維持装置とシステム、核施設と武器システムの操作用の装置やシステムを含みます。Atmelによって軍用等級として特に明確に示される以外、Atmel製品は軍用や航空宇宙の応用や環境のために設計も意図もされていません。Atmelによって車載等級として特に明確に示される以外、Atmel製品は車載応用での使用のために設計も意図もされていません。

© HERO 2016.

本応用記述はAtmelのAT12498応用記述(Rev.42736A-06/2016)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意識されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

必要と思われる部分には()内に英語表記や略称などを残す形で表記しています。

青字の部分はリンクとなっています。一般的に赤字の0,1は論理0,1を表します。その他の赤字は重要な部分を表します。