AN2634

tinyAVR® 0と1系及びmmegaAVR® 0系用ブートローダ

序説

著者: Egil Rotevatn, Microchip Technology Inc.

この応用記述はtinyAVR® 0と1系統及びmegaAVR® 0系統のﾏｲｸﾛ ｺﾝﾄﾛｰﾗ(MCU)が自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞをどう使い得るかを記述します。これは外部書き込み器の必要なしに応用ｺｰﾄﾞをﾌﾗｯｼｭ ﾒﾓﾘに設定することを使用者に許します。応用例はPythonｽｸﾘﾌﾟﾄが走行するPCとUART経由で通信するためにATtiny817 Xplained Pro(ATTINY817-XPRO)ｷｯﾄを使っています。加えて、TWI版のﾌﾞｰﾄﾛｰﾀﾞ応用も利用可能です。

提供されるﾌﾞｰﾄﾛｰﾀﾞ応用例はPythonｽｸﾘﾌﾟﾄは独自ﾌﾞｰﾄﾛｰﾀﾞ応用に対する開始点として適切です。

要点

- ﾌﾗｯｼｭ領域構成設定
- ﾌﾗｯｼｭ ﾒﾓﾘとEEPROMの両ﾒﾓﾘの読み書き
- 読み書き保護
- 自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ用Cｺｰﾄﾞ応用例
- Pythonﾎｽﾄ応用例

目次

1. 関連ﾃﾞﾊﾞｲｽ

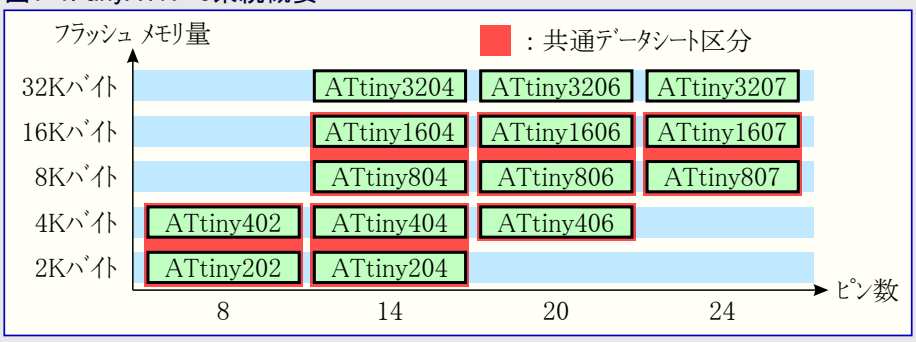
本章はこの資料に関連するﾃﾞﾊﾞｲｽを一覧にします。

1.1. tinyAVR® 0系統

下図はﾋﾟﾝ数の変種とﾒﾓﾘ量を展開してtinyAVR® 0系統ﾃﾞﾊﾞｲｽを示します。

- これらのﾃﾞﾊﾞｲｽが完全にﾋﾟﾝと機能が互換のため、垂直方向移植はｺｰﾄﾞ変更なしで可能です。
- 左への水平方向移植はﾋﾟﾝ数、従って利用可能な機能を減らします。

図1-1. tinyAVR® 0系統概要



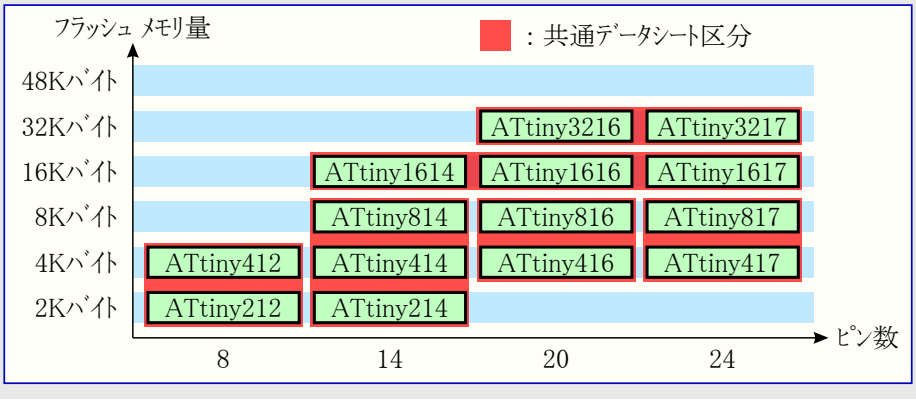
異なるﾌﾗｯｼｭ ﾒﾓﾘ量を持つﾃﾞﾊﾞｲｽは一般的に異なるSRAMとEEPROMの量を持ちます。

1.2. tinyAVR® 1系統

下図はﾋﾟﾝ配置変種とﾒﾓﾘ量を展開してtinyAVR® 1系統ﾃﾞﾊﾞｲｽを示します。

- これらのﾃﾞﾊﾞｲｽがﾋﾟﾝ互換で同じまたはより多くの機能を提供するため、垂直上方向移植はｺｰﾄﾞ変更なしに可能です。下方向移植はより少ない利用可能ないくつかの周辺機能の実体のためにｺｰﾄﾞ変更が必要かもしれません。
- 左への水平方向移植はﾋﾟﾝ数、従って利用可能な機能を減らします。

図1-2. tinyAVR® 1系統概要



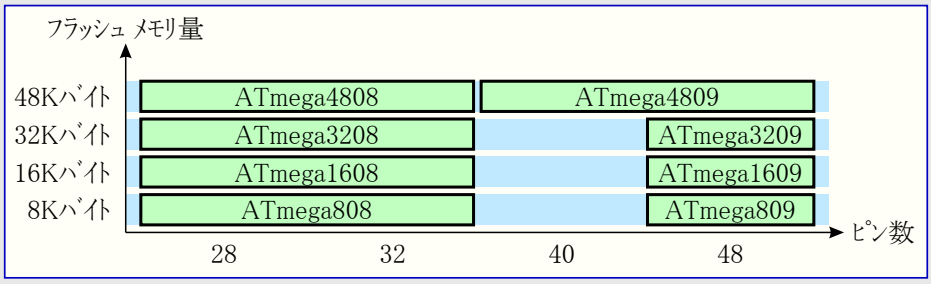
異なるﾌﾗｯｼｭ ﾒﾓﾘ量を持つﾃﾞﾊﾞｲｽは一般的に異なるSRAMとEEPROMの量を持ちます。

1.3. megaAVR® 0系統

下図はﾋﾟﾝ配置変種とﾒﾓﾘ量を展開してmegaAVR® 0系統ﾃﾞﾊﾞｲｽを示します。

- これらのﾃﾞﾊﾞｲｽが完全にﾋﾟﾝと機能が互換のため、垂直方向移植はｺｰﾄﾞ変更なしで可能です。
- 左への水平方向移植はﾋﾟﾝ数、従って利用可能な機能を減らします。

図1-3. megaAVR® 0系統概要



異なるﾌﾗｯｼｭ ﾒﾓﾘ量を持つﾃﾞﾊﾞｲｽは一般的に異なるSRAMとEEPROMの量を持ちます。

2. ﾃﾞﾊﾞｲｽ自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ

tinyAVR® 0と1系統及びmegaAVR® 0系統のMCUでは、ﾌﾗｯｼｭ ﾒﾓﾘ ﾌﾟﾛｸﾞﾗﾐﾝｸﾞが一度に1ﾍﾟｰｼﾞで行われます。ﾌﾗｯｼｭ ﾍﾟｰｼﾞの大きさはﾃﾞﾊﾞｲｽのﾌﾗｯｼｭ ﾒﾓﾘの大きさに依存して64か128ﾊﾞｲﾄのどちらかで、ﾃﾞｰﾀはそれがﾌﾗｯｼｭ ﾒﾓﾘに書かれる前に同じ大きさのﾍﾟｰｼﾞ緩衝部に設定されなければなりません。

ﾍﾟｰｼﾞ緩衝部をﾌﾗｯｼｭ ﾒﾓﾘに書く前に目的対象のﾍﾟｰｼﾞが消去されなければなりません。未消去のﾌﾗｯｼｭ ﾍﾟｰｼﾞへの書き込みはそれの内容を不正にします。NVMCTRL制御A(NVMCTRL.CTRLA)ﾚｼﾞｽﾀにﾍﾟｰｼﾞの消去と書き込み(ERWP:PAGEERASEWRITE)指令を設定することによってﾍﾟｰｼﾞにﾃﾞｰﾀを書くのと同じ時に行うことができます。

以下の手順を用いて、各指令に対してより短いﾌﾟﾛｸﾞﾗﾐﾝｸﾞ時間を許すために独立した2つの操作で消去と書き込みを行うことも可能です。

- ｱﾄﾞﾚｽを構成設定するためにﾍﾟｰｼﾞ内の位置へ見せかけ値を書いてください。
- ﾍﾟｰｼﾞ消去(ER:PAGEERASE)指令を実行してください。
- ﾍﾟｰｼﾞ緩衝部を満たしてください。
- ﾍﾟｰｼﾞ消去(WP:PAGEWRITE)指令を実行してください。

ﾍﾟｰｼﾞ緩衝部はNVMCTRL.CTRLA内の何れかの指令が実行された後で自動的に解消されます。

ﾌﾗｯｼｭ ﾒﾓﾘの語ｱﾄﾞﾚｽ指定はﾘﾄﾙ ｴﾝﾃﾞｨｱﾝ ﾊﾞｲﾄ順を使います。最下位ｱﾄﾞﾚｽ ﾋﾞｯﾄ(ﾋﾞｯﾄ0)が'0'ならば下位ﾊﾞｲﾄがｱｸｾｽされ、'1'ならば上位ﾊﾞｲﾄがｱｸｾｽされます。

NVMCTRL.CTRLAは不慮の変更を防ぐために構成設定変更保護(CCP:Configuration Change Protection)を持ちます。CCPの詳細については関連ﾃﾞﾊﾞｲｽのﾃﾞｰﾀｼｰﾄで「AVR CPU」章を参照してください。指令が終了されたことを確実にするため、NVMCTRL状態(NVMCTRL.STATUS)ﾚｼﾞｽﾀのﾌﾗｯｼｭ ﾒﾓﾘ多忙(FBUSY)の解除(0)を待つことが推奨されます。

注: NVMCTRL.CTRLAでのﾁｯﾌﾟ消去(CHER:CHIPERASE)指令はﾌﾗｯｼｭ ﾒﾓﾘ全体を消去し、故にこれは狙いがﾃﾞﾊﾞｲｽを使えなくすることでない限り、自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ中に実行されてはなりません。

2.1. ﾒﾓﾘ配置

ﾌﾗｯｼｭ ﾒﾓﾘに加えて、MCUによってEEPROMと使用者列の領域を自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞすることができます。本項は位置と領域での違いを説明します。

実際の領域の大きさとｱﾄﾞﾚｽ変位(ｵﾌｾｯﾄ)については関連ﾃﾞﾊﾞｲｽのﾃﾞｰﾀｼｰﾄを参照してください。

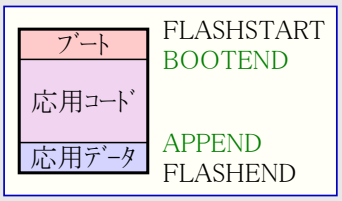
2.1.1. ﾌﾗｯｼｭ ﾒﾓﾘ

ﾌﾗｯｼｭ ﾒﾓﾘはﾌﾞｰﾄ ﾛｰﾀﾞ(BOOT)、応用ｺｰﾄﾞ(APPCODE)、応用ﾃﾞｰﾀ(APPDATA)の3つの領域に分けることができます。これらの領域間の主な違いは次のようなｱｸｾｽ権限です。

- BOOT領域内のｺｰﾄﾞはAPPCODEとAPPDATAに書くことができます。
- APPCODE領域内のｺｰﾄﾞはAPPDATAに書くことができます。
- APPDATA領域内のｺｰﾄﾞはﾌﾗｯｼｭ ﾒﾓﾘやEEPROMに書くことができません。

図2-1.はﾌﾗｯｼｭ ﾒﾓﾘ内でどう管理されるかを示します。

図2-1. ﾌﾗｯｼｭ ﾒﾓﾘ領域



FLASHSTARTはﾌﾟﾛｸﾞﾗﾑ ﾒﾓﾘとしてｱｸｾｽされる時に$0000で、ﾃﾞｰﾀ ﾒﾓﾘ経由でｱｸｾｽされる時に以下の変位(ｵﾌｾｯﾄ)で割り当てられます。

- megaAVR 0系統　　: $4000
- tinyAVR 0及び1系統 : $8000

ｱﾄﾞﾚｽ割り当ては通常のLD/ST系間接命令を用いてｱｸｾｽするために必要とされます。ﾃﾞﾊﾞｲｽ ﾍｯﾀﾞ ﾌｧｲﾙに於いてこの変位はMAPPED_PROGMEM_STARTとして定義され、故にﾃﾞｰﾀ ﾒﾓﾘ経由でﾌﾗｯｼｭ ﾒﾓﾘのｱﾄﾞﾚｽ$0100をｱｸｾｽする場合、そのｱﾄﾞﾚｽ ﾎﾟｲﾝﾀは下の例のように定義することができます。

```
uint8_t *flash_pointer = (uint8_t *) 0x100 + MAPPED_PROGMEM_START;
```

ﾌﾗｯｼｭ ﾒﾓﾘ領域の大きさは256ﾊﾞｲﾄ(128語)段階でBOOTENDとAPPENDのﾋｭｰｽﾞを通して構成設定することができます。以下の表はこれらのﾋｭｰｽﾞが領域をどう構成設定するかを示します。

表2-1. ﾌﾗｯｼｭ ﾒﾓﾘ領域の構成設定

BOOTEND	APPEND	BOOT領域	APPCODE領域	APPDATA領域
0	0	0～FLASHEND	-	-
>0	0	0～BOOTEND×256-1	BOOTEND×256～FLASHEND	-
>0	=BOOTEND	0～BOOTEND×256-1	-	BOOTEND×256～FLASHEND
>0	>BOOTEND	0～BOOTEND×256-1	BOOTEND×256～APPEND×256-1	APPEND×256～FLASHEND

これらのﾋｭｰｽﾞがﾃﾞﾊﾞｲｽ上で意図されるように校正設定されるのを確実にする良い方法はﾌﾞｰﾄﾛｰﾀﾞ ｺｰﾄﾞ ﾌﾟﾛｼﾞｪｸﾄでFUSESﾏｸﾛを使うことです。これはio.hによってｲﾝｸﾙｰﾄﾞされるfuse.hで見つけることができます。

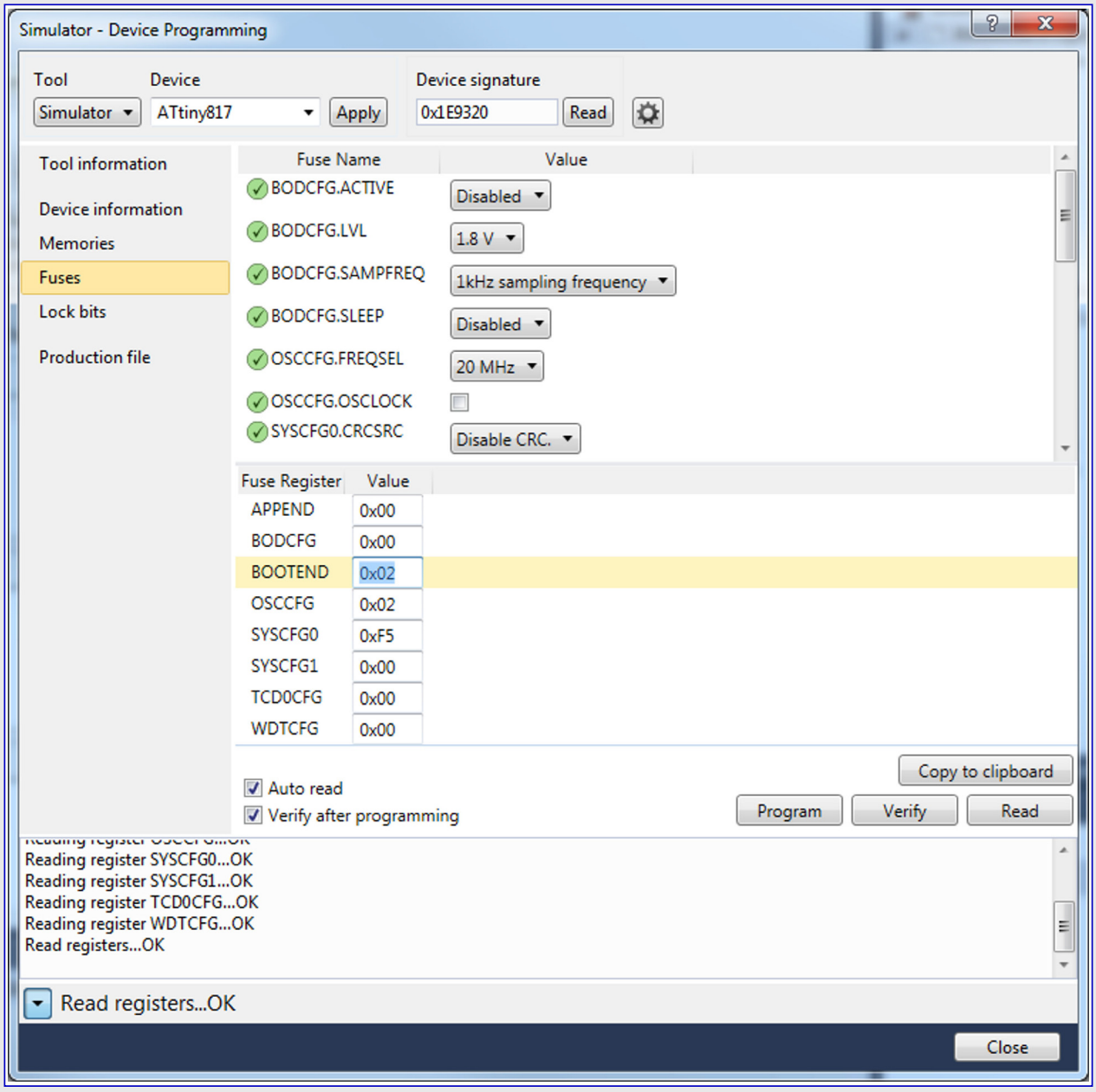
```
#include <avr/io.h>

FUSES = {
    .OSCCFG = FREQSEL_20MHZ_gc,
    .SYSCFG0 = CRCSRC_NOCRC_gc | RSTPINCFG_UPDI_gc,
    .SYSCFG1 = SUT_64MS_gc,
    .APPEND = 0x00,         // 応用ﾃﾞｰﾀ領域禁止
    .BOOTEND = 0x02         // ﾌﾞｰﾄ領域の大きさ = 0x02×256ﾊﾞｲﾄ = 512ﾊﾞｲﾄ
};
```

これはﾋｭｰｽﾞ設定をﾌﾞｰﾄﾛｰﾀﾞ用のelfﾌｧｲﾙにｺﾝﾊﾟｲﾙし、hexﾌｧｲﾙの代わりにこれがﾃﾞﾊﾞｲｽを書くのに使われる場合、このﾋｭｰｽﾞ設定はﾌﾗｯｼｭ ﾒﾓﾘと同時に書かれます。

注: BOOTENDとAPPENDだけでなく、構造体内の全てのﾋｭｰｽﾞ ﾊﾞｲﾄが構成設定されなければなりません。これは省略されたﾋｭｰｽﾞ ﾊﾞｲﾄが$00に設定され、望まれない構成設定を引き起こすかもしれないからです。

ﾃﾞﾊﾞｲｽのﾋｭｰｽﾞは**図2-2.**で示されるように、Device Programming(Ctrl+Shift+P) – Fusesを使い、Atmel® Studio 7.0から直接構成設定することもできます。

図2-2. Atmel Studio 7.0でのBOOTENDとAPPENDのﾋｭｰｽﾞ構成設定



2.1.2. EEPROM

EEPROMは以下の違いを持つﾌﾗｯｼｭ ﾒﾓﾘの応用ﾃﾞｰﾀ領域と同様な独立した領域です。

- ﾒﾓﾘ ｱﾄﾞﾚｽ$1400で開始します。
- ﾍﾟｰｼﾞの大きさはﾌﾗｯｼｭ ﾍﾟｰｼﾞの大きさの半分です。
- EEPROMからｺｰﾄﾞを実行することはできません。
- 単一ﾊﾞｲﾄ読み書きを支援。そのｱﾄﾞﾚｽ位置に対してﾍﾟｰｼﾞ緩衝部に書かれる値だけが消去/書き込みされます。
- ﾌﾗｯｼｭ ﾒﾓﾘと同じ書き込み指令ですが、NVMCTRL.STATUSで違う状態ﾋﾞｯﾄです。

2.1.3. 使用者列

使用者列領域は以下の違いを持つ1つの特別なEEPROMﾍﾟｰｼﾞです。

- ﾒﾓﾘ ｱﾄﾞﾚｽ$1300で開始します。
- ﾁｯﾌﾟ消去によって消去されません。
- 施錠されたﾃﾞﾊﾞｲｽでUPDIを通してｱｸｾｽすることができます。

2.2. ｺﾝﾊﾟｲﾗとﾘﾝｶ

Atmel® Studio 7.0はAVRﾃﾞﾊﾞｲｽに対してCとC++のｺｰﾄﾞをｺﾝﾊﾟｲﾙするのにAVR® GCCを使います。AVR GCCは特にAVRを目的対象にするGNUｺﾝﾊﾟｲﾗ集合(GCC:GNU Compiler Collection)、またはGCCについてAVR特有の何かを参照する時に使われます。

AVR GCCはAtmel® Studio 7.0なしの独立型で使うこともできます。

2.2.1. ブートローダでの標準開始ファイル

AVR GCCによって使われる標準開始ファイルは割り込みベクタ表を含み、AVR CPUとメモリを初期化して'main()'に飛びます。ブートローダによって割り込みが使われないなら、可能な限りコードを小さく保つために開始ファイルを取り去ることができます。

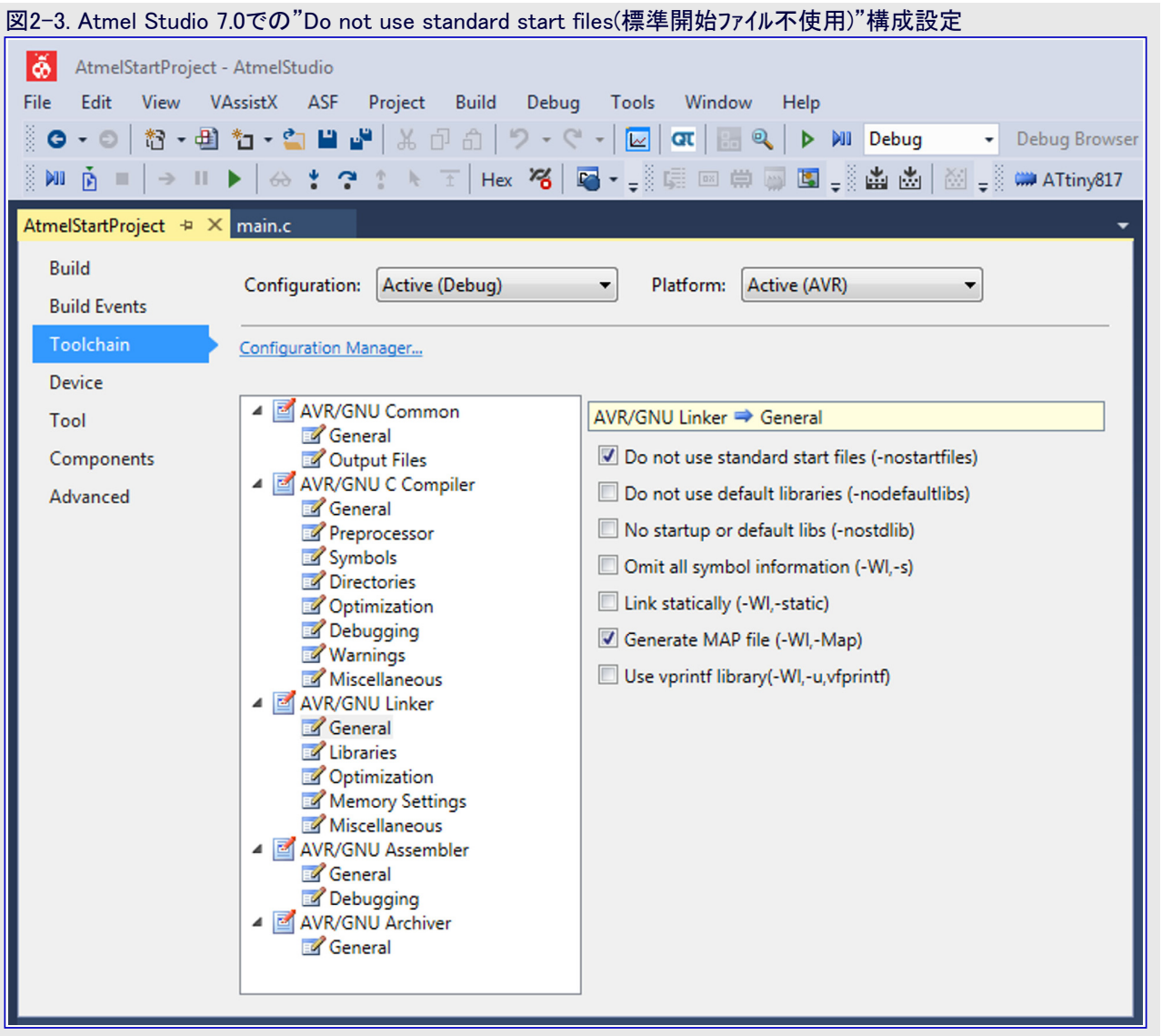
標準開始ファイルが禁止されると、'main()'が呼ばれず、故にデバイスが実行を開始する時に関数が定義されて移行されることが必要です。以下のコード断片はAVR GCCコード プロジェクトのコンストラクタ領域(.ctors)で必要とされる初期化を持つ'boot()'関数例を示します。

```
__attribute__((naked)) __attribute__((section(".ctors"))) void boot(void){
    /* C支援用にシステム初期化 */
    asm volatile("clr r1");

    /* ブートローダ コードで置き換えてください。 */
    while (1)
    {
    }
}
```

関数がCALL/RET命令を使って呼ばれずに始動で移行されるため、コンパイラは関数のプロローグとエピローグを省略することをnaked属性によって指示されます。詳細についてはAVR GCC資料をご覧ください。

AVR GCCではプロジェクトのコンパイル時に-nostartfilesリンカ フラグを設定することによって標準開始フィイルが禁止されます。Atmel® Studio 7.0に於いてこれは**図2-3.**で見えるように、Project Properties(Alt+F7)⇒Toolchain⇒AVR/GNU Linker⇒Generalで見つけることができます。

図2-3. Atmel Studio 7.0での"Do not use standard start files(標準開始ファイル不使用)"構成設定



2.2.2. 応用開始

AVR GCCのﾘﾝｶ ｽｸﾘﾌﾟﾄに対して、ｺﾝﾊﾟｲﾙされた応用ｺｰﾄﾞを置くのにﾌﾗｯｼｭ ﾒﾓﾘ内の何処かを知るため、.textｺｰﾄﾞ領域の開始はﾌﾗｯｼｭ領域の位置に対応して構成設定されなければなりません。入力は語整列され、故におそらく以下の数値が使われます。

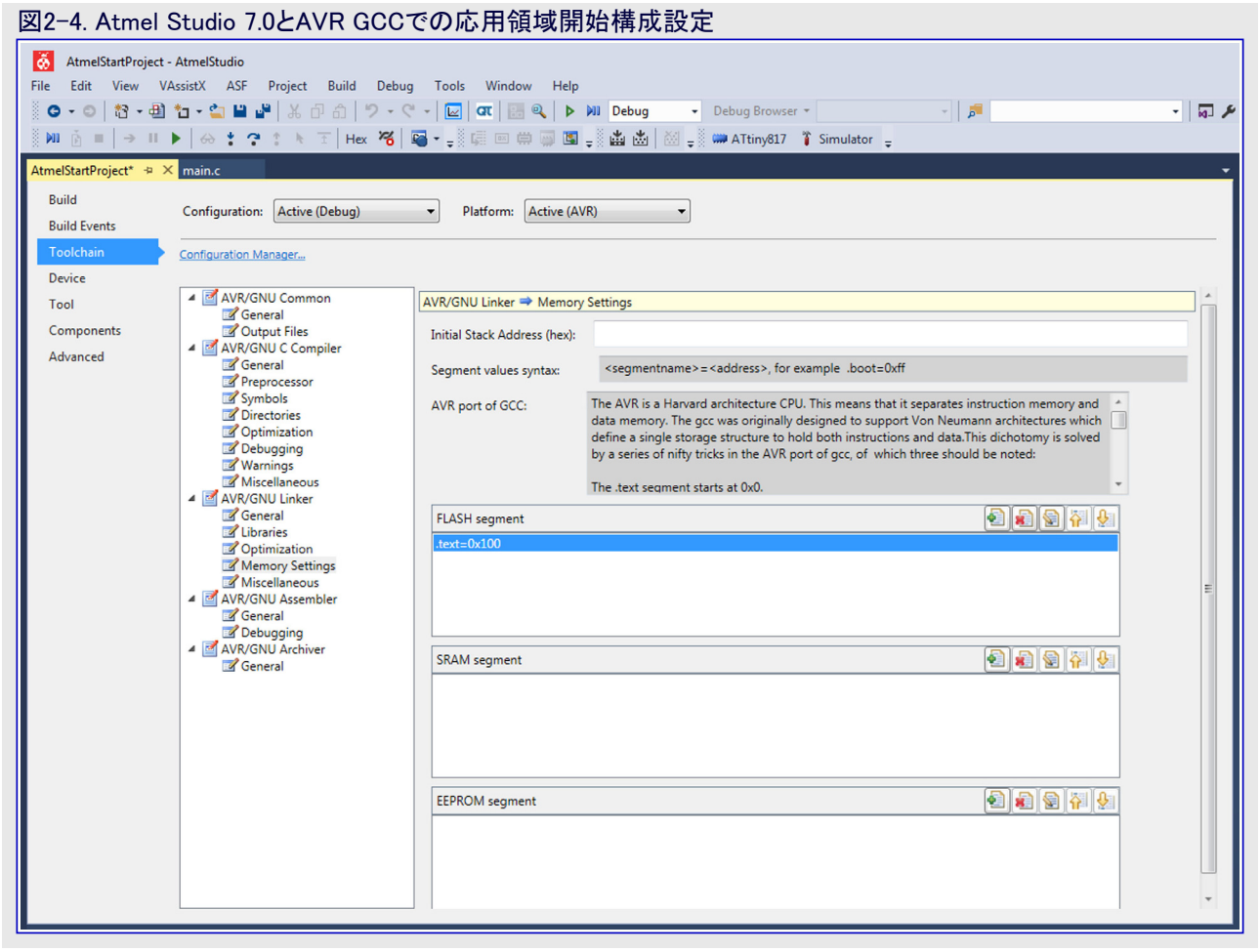
- ﾌﾞｰﾄ開始　　：$0000 (既定)
- 応用ｺｰﾄﾞ開始：BOOTEND×128
- 応用ﾃﾞｰﾀ開始：APPEND×128

例としてBOOTENDﾋｭｰｽﾞの$02設定(256語ﾌﾞｰﾄ容量)を使うと、応用ｺｰﾄﾞ.text領域の再配置は以下のﾘﾝｶ任意選択によって行われます。

```
-Wl,--section-start=.text=0x100
```

Atmel® Studio 7.0では**図2-4.**で示されるように、Project Properties(Alt+F7)⇒Toolchain⇒AVR/GNU Linker⇒Memory Settingsでﾌﾗｯｼｭ ｾｸﾞﾒﾝﾄに.text=0x100を追加することによって再配置を行うことができます。

図2-4. Atmel Studio 7.0とAVR GCCでの応用領域開始構成設定



2.3. ﾒﾓﾘ保護

ｱｸｾｽや書き込みからﾌﾗｯｼｭ ﾒﾓﾘのいくつかまたは全てを保護するには利用可能ないくつかの保護の段階があります。禁止できない保護は「2.1. ﾒﾓﾘ配置」項で記述されるﾌﾗｯｼｭ領域書き込み権です。加えて、安全性を追加するために以下の保護形式を構成設定することができます。

2.3.1. BOOTLOCKとAPCWP

ﾌﾞｰﾄ領域施錠(BOOTLOCK)と応用ｺｰﾄﾞ領域書き込み保護(APCWP)はNVMCTRL制御B(NVMCTRL.CTRLB)に置かれ、走行時書き込み保護に使われます。

BOOTLOCKはBOOT(ﾌﾞｰﾄ領域)からの読み込みｱｸｾｽとｺｰﾄﾞ実行を防ぎます。このﾋﾞｯﾄはBOOTから実行されるｺｰﾄﾞによってだけ設定(1)することができ、ｺｰﾄﾞ実行がBOOTの外へ移る時に活性(有効)にします。BOOTが施錠されると、BOOTから読むどの試みも$00を返し、BOOTから実行されるどの命令も無操作(NOP)命令になります。

APCWPはAPPCODE(応用ｺｰﾄﾞ領域)へのｱｸｾｽを制御します。設定(1)されると、この領域へ書くどの試みも書き込み異常に帰着します。

注: 一旦許可されると、NVMCTRL.CTRLB内のﾋﾞｯﾄはﾘｾｯﾄによってだけ禁止(解除)することができます。

2.3.2. EESAVE

EESAVEはSYSCFG0ﾋｭｰｽﾞ ﾊﾞｲﾄ内のﾋﾞｯﾄの1つです。これはﾁｯﾌﾟ消去中にEEPROMが消去されるか否かを制御します。

注: EESAVEが許可されてﾁｯﾌﾟ消去を動かすことによってﾃﾞﾊﾞｲｽが解錠される場合、EEPROMに存続するﾃﾞｰﾀは読むことができます。

2.3.3. 施錠ﾋﾞｯﾄ

施錠ﾋﾞｯﾄはﾋｭｰｽﾞ、ﾌﾗｯｼｭ ﾒﾓﾘ、EEPROMをｱｸｾｽすることから書き込み器を妨げることができる独立したﾋｭｰｽﾞ内に置かれます。施錠されたﾃﾞﾊﾞｲｽがUPDIでｱｸｾｽされると、使用者にﾃﾞﾊﾞｲｽIDと使用者列のｱｸｾｽとﾁｯﾌﾟ消去の実行を許す、制御と状態の空間だけが利用可能です。

施錠ﾋﾞｯﾄで施錠されたﾃﾞﾊﾞｲｽを解錠するにはﾁｯﾌﾟ消去が実行されなければなりません。

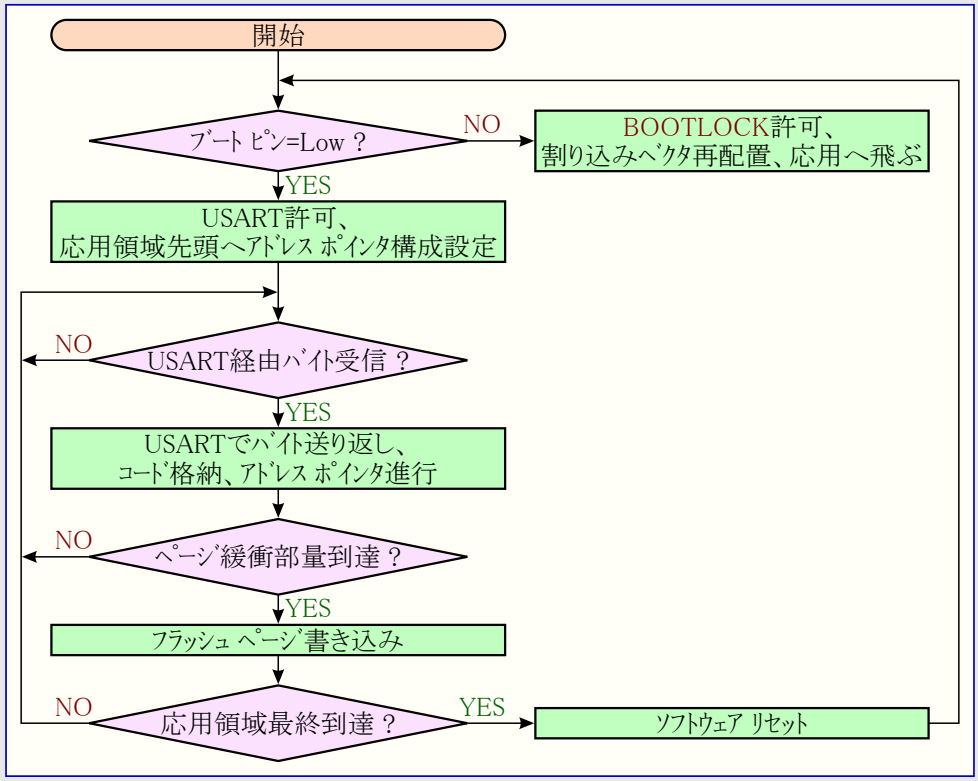
2.4. ﾌﾞｰﾄﾛｰﾀﾞ動作

ﾌﾞｰﾄﾛｰﾀﾞ例の開始でGPIOﾋﾟﾝの状態がﾎﾟｰﾘﾝｸﾞされます。このﾌﾞｰﾄ ﾋﾟﾝがHighなら、BOOTLOCKが許可され、実行はAPPCODEへ飛びます。このﾋﾟﾝがLowなら、ﾌﾞｰﾄﾛｰﾀﾞがUSARTを渡るﾃﾞｰﾀの受信を開始し、そのﾃﾞｰﾀをﾍﾟｰｼﾞ緩衝部に書きます。ﾍﾟｰｼﾞ緩衝部が満たされると、そのﾍﾟｰｼﾞがﾌﾗｯｼｭ ﾒﾓﾘに書かれます。ﾌﾗｯｼｭ ﾒﾓﾘ全体を満たすのに充分なﾃﾞｰﾀが受信された後、ｿﾌﾄｳｪｱ ﾘｾｯﾄが発行され、全ての周辺機能がﾘｾｯﾄします。そして新しい応用を開始することができます。

ﾃﾞﾊﾞｲｽへの最初のﾌﾞｰﾄﾛｰﾀﾞ書き込み時、APPCODEとAPPDATAは空です。この状況に於いて、始動でﾌﾞｰﾄ ﾋﾟﾝがHighの場合、ｺｰﾄﾞ実行は全てのﾊﾞｲﾄが$FFに等しいﾌﾗｯｼｭ領域へ飛びます。これはNOPとして実行されます。実行がﾌﾗｯｼｭ ﾒﾓﾘの最後に達すると、BOOTの先頭に丸められて再びﾌﾞｰﾄﾛｰﾀﾞを実行します。これはﾌﾞｰﾄ ﾋﾟﾝがLowになるまでの繰り返しを生成します。

以下の図はﾌﾞｰﾄﾛｰﾀﾞ動作の流れ構成図です。

図2-5. ﾌﾞｰﾄﾛｰﾀﾞ流れ図



注: ATtiny817 Xplained Proで使われるﾌﾞｰﾄｺｰﾄﾞ例について、ﾌﾞｰﾄ ﾋﾟﾝは外部ﾌﾟﾙｱｯﾌﾟを持つSW1触覚ｽｲｯﾁに接続されます。

応用ｺｰﾄﾞへ飛ぶ前に、応用からﾌﾞｰﾄへのｱｸｾｽを防ぐためにBOOTLOCKが許可されます。

3. ホスト応用

ブートローダ関係性に於いてホストはデバイスに応用コードを送る責任があるシステムです。これは通常、ファームウェア更新を実行する目的のために目的対象デバイスに接続することができるコンピュータやCPU、または同じ回路基板上のCPUホストです。

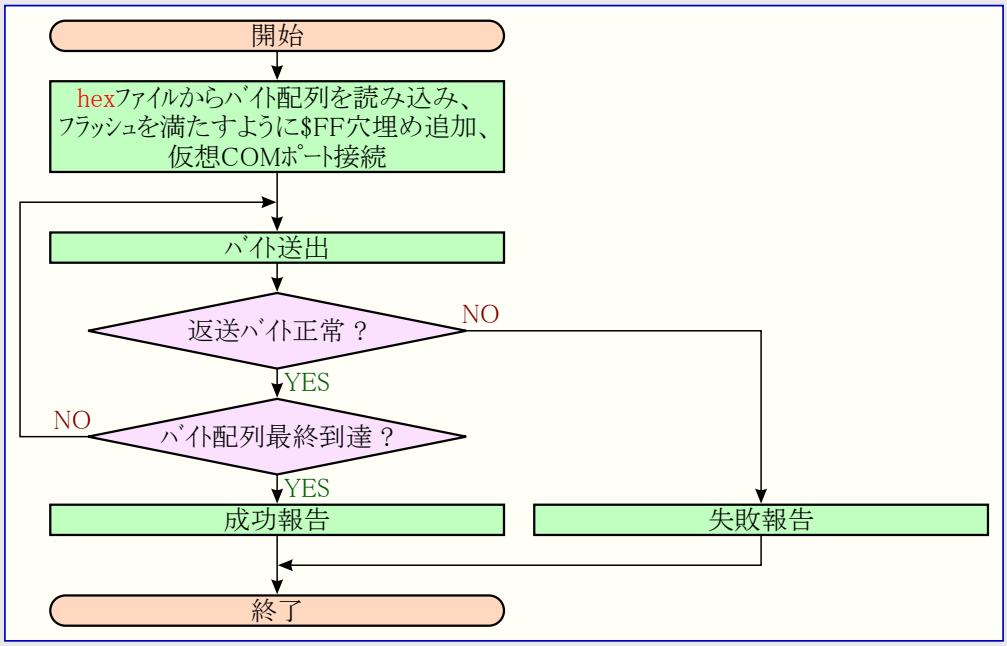
目的対象デバイスと通信することができる限り、ホスト応用の作成方法に殆ど制限はありません。最も簡単なホストは基本的なコマンド行インターフェースだけを持ち、一方でいくつかはいくつかの安全性の層と高度な構成設定を持つ画像使用者インアーフェースを持ちます。

デバイスが無線接続を持つなら、無線(OTA:Over-The-Air)プログラミングも利用可能です。これはスマートフォン応用からのソフトウェア更新機能や、コンピュータや書き込み器に各デバイスを物理的に接続せずに大量のデバイスを更新する他の方法を追加することを容易にします。

3.1. Pythonスクリプト操作

Pythonスクリプト例はブートローダ例を走行しているデバイスに応用hexファイルを設定します。これはデバイスとPC間の橋渡しとしてXplained Pro組み込みデバッガを用いて達成されます。送られた各バイトに対して、データ転送が成功したことを通知するために返しで同じ値が予期されます。ブートローダはAPPCODEを満たすのに充分なデータを予期します。hexファイルがそれを行うのに充分なデータを含まない場合、APPCODEが満たされるまで$FFが送られます。図3-1.はこれがどう動くかを示します。

図3-1. Pythonスクリプト流れ図

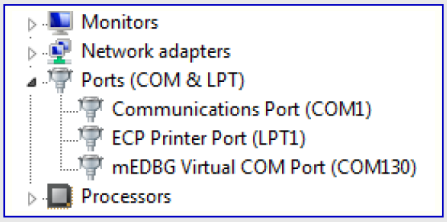


スクリプトを動かすには以下の引数が必要とされます。

1. 書き込むhexファイル。ファイルが同じフォルダでない場合はパスを含めれください。
2. 総フラッシュ メモリ容量。これはバイト配列の大きさと未使用コード空間への$FF穴埋め追加に必要とされます。
3. UART通信に使われる仮想COMポート。
   - これはWindows®に於いてデバイス マネージャーで一覧にされます。ATtiny817 Xplaine d ProはEDBG Virtual COM Port (COMxxx)として一覧にされます。

   注: 仮想COMポートは基板上のATtiny817デバイスのUSARTピン(PB2-TxDとPB3-RxD)に接続されます。
4. 使われるボーレート。ブートローダ例によって使われる既定ボーレートは11520bpsです。

図3-2. mEDBG仮想COMポート



8Kバイトの総フラッシュ メモリ容量を持つCOM130ポートに接続されたATtiny817 Xplained Proに対してスクリプトの走行は応用例の公開版を設定(書き込む)時にことように見えます。

```
python tiny_uploader.py ./App/Release/App.hex 8192 COM130 115200
```

注: スクリプトを開始する前にデバイスをブートローダ動作に置くことを確実にしてください。これはATtiny817 Xplained Proの電源投入またはリセットの間にSW1を押すことによって行われます。

応用書き込み成功後、コマンド ウィンドウはおそらくこのように見えます。

```
c:¥[your_path]>python tiny_uploader.py ./App/Release/App.hex 0x2000 COM130 115200
Uploading 7936 bytes...
100.00%
OK
```

書き込みが失敗した場合、以下の形式のﾒｯｾｰｼﾞが現れます。この例でこれは仮想COMﾎﾟｰﾄが通信時間超過後に$00を返すためです。

```
Uploading 7936 bytes...                          書き込み中 7936ﾊﾞｲﾄ...

Failed at address 0x0100                         ｱﾄﾞﾚｽ$0100で失敗
Value 0x00 echoed, expected 0x19                 予期される$19が値$00で返されました。
```

Python必要条件

ｽｸﾘﾌﾟﾄはPython 2.7.13と3.5.2を支援するように書かれていて、更にそれ以降の版も多分、異常なしに動くでしょう。https://www.python.org/downloads/または好きなPython配給元からPythonをﾀﾞｳﾝﾛｰﾄﾞしてください。

Pythonに加えて、これらの単位部がｲﾝｽﾄｰﾙされることが必要です。

- hexﾌｧｲﾙの解析用、intelhex
- 直列通信用、pyserial
- Python2と3の両方での互換用、future

以下の命令はPython一括指標から依存性を持つ単位部の最新版をｲﾝｽﾄｰﾙします。

```
python -m pip install -U future pyserial intelhex
```

この命令は単位部を最新版へ更新するのにも使うことができます。

4. 機能拡張

ﾌﾞｰﾄﾛｰﾀﾞ例は最も機能的な機能だけを含むﾌﾞｰﾄﾛｰﾀﾞの簡単な実装です。けれども、この実装は沢山の方法で拡張することができます。本章はいくつかの可能な改良を紹介します。

4.1. ﾌﾞｰﾄ動作移行

物理的なﾋﾟﾝの状態はﾃﾞﾊﾞｲｽをﾌﾞｰﾄﾛｰﾀﾞに移行させるための唯一の方法ではなく、度々ﾌﾞｰﾄﾛｰﾀﾞ更新を起動する応用のために必要です。下の例は更新を起動するために使用者列またはEEPROM内の値を調べる関数を示します。

```
static bool is_boot_requested(void)
{
    /* ﾌｧｰﾑｳｪｱからのﾌﾞｰﾄ要求調査 */
    if (USERROW.USERROW31 == 0xEB) {
        /* Clear boot request*/
        USERROW.USERROW31 = 0xff;
        _PROTECTED_WRITE_SPM(NVMCTRL.CTRLA, NVMCTRL_CMD_PAGEERASEWRITE_gc);
        while(NVMCTRL.STATUS & NVMCTRL_EEBUSY_bm);
    }
    /* SW1(PC5)がLowか調査 */
    else if(VPORTC.IN & PIN5_bm) {
        return false;
    }

    return true;
}
```

ﾋﾟﾝをLowに引くことなしにﾌﾞｰﾄ動作へ移行するには使用者列のﾊﾞｲﾄ31が応用または書き込み器のどちらかによって書かれることが必要です。下の例は必要とされる値を書いてﾃﾞﾊﾞｲｽをﾘｾｯﾄする方法を示します。

```
void enter_bootloader(void)
{
    /* ﾌﾞｰﾄ要求書き込み */
    USERROW.USERROW31 = 0xEB;
    _PROTECTED_WRITE_SPM(NVMCTRL.CTRLA, NVMCTRL_CMD_PAGEERASEWRITE_gc);
    while(NVMCTRL.STATUS & NVMCTRL_EEBUSY_bm);

    /* ｼｽﾃﾑ ﾘｾｯﾄ発行 */
    _PROTECTED_WRITE(RSTCTRL.SWRR, RSTCTRL_SWRE_bm);
}
```

これら2つの関数は共に電源OFF/ONと物理ﾋﾟﾝを必要とせずにﾌﾞｰﾄﾛｰﾀﾞ動作へ移行することを可能にします。

4.2. インターフェース

ホスト通信に利用可能なインターフェースは最終応用間で違うかもしれません。ブートローダ例が応用を受信するのにUSART周辺機能の基本的な構成設定を利用すると同時に、これはboot.cで以下の3つのUART関数を置き換えることによって必要とされるように容易に更新することができます。

- 'static void init_uart(void)'
- 'static uint8_t uart_receive(void)'
- 'static void uart_send(uint8_t byte)'

全てのtinyAVR® 0と1系統及びmegaAVR® 0系統のデバイスは直列通信に対して利用可能なUSART、TWI、SPIのハードウェア周辺機能を持ち、入出力ピンは独自デジタル規約にも使うことができます。

利用可能な周辺機能割り込みは割り込みベクタ表をブート領域の先頭に再配置することによってブートローダ コードで使うことができます。これはCPUINT制御A(CPUINT.CTRLA)レジスタの割り込みベクタ選択(IVSEL)ビットを許可(1)することによって行われます。より多くの情報については「AN1982 – tinyAVR® 0と1系及びmegaAVR® 0系での割り込みシステム」をご覧ください。

従装置動作でTWI周辺機能を使うブートローダ コード例も利用可能です。これはこの応用記述に関連する.zipファイルに置かれます。

4.3. データ完全性

デバイスに転送されたコードが正しく受信されることを確実にするため、やって来るデータで巡回冗長検査(CRC:Cyclic Redundancy Check)を使うことができます。これはデータを受信すると同時、またはコードを実行する前に行うことができます。

全てのtinyAVR 1系統とmegaAVR 0系統のデバイスはフラッシュ メモリ内容を確認するのに使うことができる巡回冗長検査メモリ走査(CRCSCAN:Cyclic Redundancy Check Memory Scan)を持ちます。この周辺機能の使用方法のより多くの情報については「AN2521 – tinyAVR® 0と1系及びmegaAVR® 0系でのCRCSCAN」応用記述をご覧ください。

4.4. 機密性

製品とそれの応用コードが複製、模造、または改ざんされないことを保証するために暗号化対策が必要かもしれません。ブートローダでのCryptoAuthentication™実装は正当なコードだけがホストとデバイス間で転送され得ることを保証します。

より多くの情報についてはMicrochipのCryptoAuthentication™サイトを訪ねてください。

5. 参照

以下の参照はこの応用記述によって網羅されるデバイスと話題に関連されています。

- AN1982 – tinyAVR® 0と1系及びmegaAVR® 0系での割り込みシステム:
  – http://www.microchip.com/wwwappnotes/appnotes.aspx?appnote=en603505
- AN2521 – tin yAVR® 1系デバイスでのCRCSCAN:
  – http://www.microchip.com/wwwappnotes/appnotes.aspx?appnote=en599876
- ATtiny817 Xplained Pro使用者の手引き:
  – http://ww1.microchip.com/downloads/en/DeviceDoc/50002684A.pdf
- MicrochipのCryptoAuthentication™サイト:
  – http://www.microchip.com/design-centers/security-ics/cryptoauthentication
- AVR® GCC資料:
  – https://gcc.gnu.org/onlinedocs/gcc/AVR-Function-Attributes.html

6. 改訂履歴

資料改訂	日付	注釈
A	2018年1月	初版資料公開
B	2018年4月	TWIブートローダ コードに対する情報を追加
C	2018年10月	「**関連デバイス**」章の**図1-1.**、**図1-2.**、**図1-3.**を更新。文法と句読法を修正。

Microchipウェブ サイト

Microchipはhttp://www.microchip.com/で当社のウェブ サイト経由でのオンライン支援を提供します。このウェブ サイトはお客様がファイルや情報を容易に利用可能にする手段として使われます。お気に入りのインターネット ブラウザを用いてアクセスすることができ、ウェブ サイトは以下の情報を含みます。

- **製品支援** – データシートと障害情報、応用記述と試供プログラム、設計資源、使用者の手引きとハードウェア支援資料、最新ソフトウェア配布と保管されたソフトウェア
- **全般的な技術支援** – 良くある質問(FAQ)、技術支援要求、オンライン検討グループ、Microchip相談役プログラム員一覧
- **Microshipの事業** – 製品選択器と注文の手引き、最新Microchip報道発表、セミナーとイベントの一覧、Microchip営業所の一覧、代理店と代表する工場

お客様への変更通知サービス

Microchipのお客様通知サービスはMicrochip製品を最新に保つのに役立ちます。加入者は指定した製品系統や興味のある開発ツールに関連する変更、更新、改訂、障害情報がある場合に必ず電子メール通知を受け取ります。

登録するにはhttp://www.microchip.com/でMicrochipのウェブ サイトをアクセスしてください。”Support”下で”Customer Change Notification”をクリックして登録指示に従ってください。

お客様支援

Microchip製品の使用者は以下のいくつかのチャネルを通して支援を受け取ることができます。

- 代理店または販売会社
- 最寄りの営業所
- 現場応用技術者(FAE:Field Aplication Engineer)
- 技術支援

お客様は支援に関してこれらの代理店、販売会社、または現場応用技術者(FAE)に連絡を取るべきです。最寄りの営業所もお客様の手助けに利用できます。営業所と位置の一覧はこの資料の後ろに含まれます。

技術支援はhttp://www.microchip.com/supportでのウェブ サイトを通して利用できます。

Microchipデバイス コード保護機能

Microchipデバイスでの以下のコード保護機能の詳細に注意してください。

- Microchip製品はそれら特定のMicrochipデータシートに含まれる仕様に合致します。
- Microchipは意図した方法と通常条件下で使われる時に、その製品系統が今日の市場でその種類の最も安全な系統の1つであると考えます。
- コード保護機能を破るのに使われる不正でおそらく違法な方法があります。当社の知る限りこれらの方法の全てはMicrochipのデータシートに含まれた動作仕様外の方法でMicrochip製品を使うことが必要です。おそらく、それを行う人は知的財産の窃盗に関与しています。
- Microchipはそれらのコードの完全性について心配されているお客様と共に働きたいと思います。
- Microchipや他のどの半導体製造業者もそれらのコードの安全を保証することはできません。コード保護は当社が製品を”破ることができない”として保証すると言うことを意味しません。

コード保護は常に進化しています。Microchipは当社製品のコード保護機能を継続的に改善することを約束します。Microchipのコード保護機能を破る試みはデジタル ミレニアム著作権法に違反するかもしれません。そのような行為があなたのソフトウェアや他の著作物に不正なアクセスを許す場合、その法律下の救済のために訴権を持つかもしれません。

法的通知

デバイス応用などに関してこの刊行物に含まれる情報は皆さまの便宜のためにだけ提供され、更新によって取り換えられるかもしれません。皆さまの応用が皆さまの仕様に合致するのを保証するのは皆さまの責任です。**Microchipはその条件、品質、性能、商品性、目的適合性を含め、明示的にも黙示的にもその情報に関連して書面または表記された書面または黙示の如何なる表明や保証もしません。**Microchipはこの情報とそれの使用から生じる全責任を否認します。生命維持や安全応用でのMicrochipデバイスの使用は完全に購入者の危険性で、購入者はそのような使用に起因する全ての損害、請求、訴訟、費用からMicrochipを擁護し、補償し、免責にすることに同意します。他に言及されない限り、Microchipのどの知的財産権下でも暗黙的または違う方法で許認可は譲渡されません。

商標

DNVによって認証された品質管理ｼｽﾃﾑ

ISO/TS 16949

Microchipはその世界的な本社、ｱﾘｿﾞﾅ州のﾁｬﾝﾄﾞﾗｰとﾃﾝﾍﾟ、ｵﾚｺﾞﾝ州ｸﾞﾗｼｬﾑの設計とｳｪﾊｰ製造設備とｶﾘﾌｫﾙﾆｱとｲﾝﾄﾞの設計ｾﾝﾀｰに対してISO/TS-16949:2009認証を取得しました。当社の品質ｼｽﾃﾑの処理と手続きはPIC® MCUとdsPIC® DSC、KEELOQ符号飛び回りﾃﾞﾊﾞｲｽ、直列EEPROM、ﾏｲｸﾛ周辺機能、不揮発性ﾒﾓﾘ、ｱﾅﾛｸﾞ製品用です。加えて、開発ｼｽﾃﾑの設計と製造のためのMicrochipの品質ｼｽﾃﾑはISO 9001:2000認証取得です。

日本語© *HERO* 2020.

本応用記述はMicrochipのAN2634応用記述(DS00002634C-2018年10月)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訳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。

世界的な販売とサービス

米国	亜細亜/太平洋	亜細亜/太平洋	欧州
本社 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 技術支援: http://www.microchip.com/support ｳｪﾌﾞｱﾄﾞﾚｽ: www.microchip.com **ｱﾄﾗﾝﾀ** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **ｵｰｽﾁﾝ** TX Tel: 512-257-3370 **ﾎﾞｽﾄﾝ** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **ｼｶｺﾞ** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **ﾀﾞﾗｽ** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **ﾃﾞﾄﾛｲﾄ** Novi, MI Tel: 248-848-4000 **ﾋｭｰｽﾄﾝ** TX Tel: 281-894-5983 **ｲﾝﾃﾞｱﾅﾎﾟﾘｽ** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **ﾛｻﾝｾﾞﾙｽ** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **ﾛｰﾘｰ** NC Tel: 919-844-7510 **ﾆｭｰﾖｰｸ** NY Tel: 631-435-6000 **ｻﾝﾎｾ** CA Tel: 408-735-9110 Tel: 408-436-4270 **ｶﾅﾀﾞ - ﾄﾛﾝﾄ** Tel: 905-695-1980 Fax: 905-695-2078	**ｵｰｽﾄﾗﾘｱ - ｼﾄﾞﾆｰ** Tel: 61-2-9868-6733 **中国 - 北京** Tel: 86-10-8569-7000 **中国 - 成都** Tel: 86-28-8665-5511 **中国 - 重慶** Tel: 86-23-8980-9588 **中国 - 東莞** Tel: 86-769-8702-9880 **中国 - 広州** Tel: 86-20-8755-8029 **中国 - 杭州** Tel: 86-571-8792-8115 **中国 - 香港特別行政区** Tel: 852-2943-5100 **中国 - 南京** Tel: 86-25-8473-2460 **中国 - 青島** Tel: 86-532-8502-7355 **中国 - 上海** Tel: 86-21-3326-8000 **中国 - 瀋陽** Tel: 86-24-2334-2829 **中国 - 深圳** Tel: 86-755-8864-2200 **中国 - 蘇州** Tel: 86-186-6233-1526 **中国 - 武漢** Tel: 86-27-5980-5300 **中国 - 西安** Tel: 86-29-8833-7252 **中国 - 廈門** Tel: 86-592-2388138 **中国 - 珠海** Tel: 86-756-3210040	**ｲﾝﾄﾞ - ﾊﾝｶﾞﾛｰﾙ** Tel: 91-80-3090-4444 **ｲﾝﾄﾞ - ﾆｭｰﾃﾞﾘｰ** Tel: 91-11-4160-8631 **ｲﾝﾄﾞ - ﾌﾟﾈｰ** Tel: 91-20-4121-0141 **日本 - 大阪** Tel: 81-6-6152-7160 **日本 - 東京** Tel: 81-3-6880-3770 **韓国 - 大邱** Tel: 82-53-744-4301 **韓国 - ｿｳﾙ** Tel: 82-2-554-7200 **ﾏﾚｰｼｱ - ｸｱﾗﾙﾝﾌﾟｰﾙ** Tel: 60-3-7651-7906 **ﾏﾚｰｼｱ - ﾍﾟﾅﾝ** Tel: 60-4-227-8870 **ﾌｨﾘﾋﾟﾝ - ﾏﾆﾗ** Tel: 63-2-634-9065 **ｼﾝｶﾞﾎﾟｰﾙ** Tel: 65-6334-8870 **台湾 - 新竹** Tel: 886-3-577-8366 **台湾 - 高雄** Tel: 886-7-213-7830 **台湾 - 台北** Tel: 886-2-2508-8600 **ﾀｲ - ﾊﾞﾝｺｸ** Tel: 66-2-694-1351 **ﾍﾞﾄﾅﾑ - ﾎｰﾁﾐﾝ** Tel: 84-28-5448-2100	**ｵｰｽﾄﾘｱ - ｳﾞｪﾙｽ** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **ﾃﾞﾝﾏｰｸ - ｺﾍﾟﾝﾊｰｹﾞﾝ** Tel: 45-4450-2828 Fax: 45-4485-2829 **ﾌｨﾝﾗﾝﾄﾞ - ｴｽﾎﾟｰ** Tel: 358-9-4520-820 **ﾌﾗﾝｽ - ﾊﾟﾘ** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **ﾄﾞｲﾂ - ｶﾞﾙﾋﾝｸﾞ** Tel: 49-8931-9700 **ﾄﾞｲﾂ - ﾊｰﾝ** Tel: 49-2129-3766400 **ﾄﾞｲﾂ - ﾊｲﾙﾌﾞﾛﾝ** Tel: 49-7131-67-3636 **ﾄﾞｲﾂ - ｶｰﾙｽﾙｰｴ** Tel: 49-721-625370 **ﾄﾞｲﾂ - ﾐｭﾝﾍﾝ** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **ﾄﾞｲﾂ - ﾛｰｾﾞﾝﾊｲﾑ** Tel: 49-8031-354-560 **ｲｽﾗｴﾙ - ﾗｰﾅﾅ** Tel: 972-9-744-7705 **ｲﾀﾘｱ - ﾐﾗﾉ** Tel: 39-0331-742611 Fax: 39-0331-466781 **ｲﾀﾘｱ - ﾊﾟﾄﾞﾊﾞ** Tel: 39-049-7625286 **ｵﾗﾝﾀﾞ - ﾃﾞﾙｰﾈﾝ** Tel: 31-416-690399 Fax: 31-416-690340 **ﾉﾙｳｪｰ - ﾄﾛﾝﾊｲﾑ** Tel: 47-72884388 **ﾎﾟｰﾗﾝﾄﾞ - ﾜﾙｼｬﾜ** Tel: 48-22-3325737 **ﾙｰﾏﾆｱ - ﾌﾞｶﾚｽﾄ** Tel: 40-21-407-87-50 **ｽﾍﾟｲﾝ - ﾏﾄﾞﾘｰﾄﾞ** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **ｽｳｪｰﾃﾞﾝ - ｲｪｰﾃﾎﾞﾘ** Tel: 46-31-704-60-40 **ｽｳｪｰﾃﾞﾝ - ｽﾄｯｸﾎﾙﾑ** Tel: 46-8-5090-4654 **ｲｷﾞﾘｽ - ｳｫｰｷﾝｶﾞﾑ** Tel: 44-118-921-5800 Fax: 44-118-921-5820