



AN247

A CAN Bootloader for PIC18F CAN Microcontrollers

アプリケーションノート AN247 完全日本語訳（19頁）中の抜粋 サンプル

(Microchip 社の原典の工業所有権表示は次頁に掲載。)

日本語訳文の著作権はテック・ハンゾウガネ諸橋義明に帰属します。

Japanese Translation CopyRights

©2008 Tech・Hanzougane Yoshiaki Morohashi

尚、日本語訳文から派生する利用者のいかなる不利益もテック・ハンゾウガネ諸橋義明は責任を負いません。

Note the following details of the code protection feature on PICmicro[®] MCUs.

- The PICmicro family meets the specifications contained in the Microchip Data Sheet.
- Microchip believes that its family of PICmicro microcontrollers is one of the most secure products of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the PICmicro microcontroller in a manner outside the operating specifications contained in the data sheet. The person doing so may be engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable".
- Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our product.

If you have any further questions about this matter, please contact the local sales office nearest to you.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Trademarks


The Microchip name and logo, the Microchip logo, FilterLab, KEELOQ, MPLAB, PIC, PICmicro, PICMASTER, PICSTART, PRO MATE, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

dsPIC, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microID, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, MXDEV, PICC, PICDEM, PICDEM.net, rPIC, Select Mode and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A.

Serialized Quick Term Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2001, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.



Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro[®] 8-bit MCUs, KEELOQ[®] code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.



AN247

A CAN Bootloader for PIC18F CAN Microcontrollers

*Author: Ross M. Fosler
Microchip Technology Inc.*

イントロダクション

マイクロチップの拡張フラッシュマイクロコントローラデバイス中にビルトインされた多くの特徴に自己プログラムするプログラムメモリの機能があります。この非常に有用な特徴はユーザーにブートローディング動作を行なう能力を与えるために計画的に含まれました。PIC18F458のようなデバイスは指定された「ブートブロック」、特にブートロードファームウェアに割り当てられたブロード可能プログラムメモリの小セクションで設計されますこのアプリケーションノートはCANモジュールでマイクロコントローラのPIC18Fファミリーのために単純なブートローダー実装を実演します。最小コードスペースを必要とする間に、この実装のゴールは最大のパフォーマンスと機能性を強調することです。ユーザーのためにCANを開発するシステムを可能にしました、それはもっとレベルが高いネットワークプロトコルでいっそう複雑な、そして注文製のシステムを開発するために使われることができる低レベルのフレームワークを提供します。

CANバス上のフィールドプログラミングに対する考慮

一つのデバイスにおいてフラッシュテクノロジーと強固なネットワークコミュニケーション能力の組み合わせはネットワーク上のプログラム可能性を非常に望ましいオプションにします。

しかしながら、これは孤立して一つのフラッシュデバイスをプログラムするためにブートローダーを使うようないっそう典型的な使用から非常に異なったチャレンジをCANバスネットワーク上でブートローディングするようにします。鍵となる問題のいくつかがネットワークプログラミング上にあると考えましょう。

一つの、あるいはグループプログラミング

CANバスネットワーク上にブートローディング能力を提供することは若干の慎重さがいります。例えば、多くのノードを持っているシステムがいくつかのノードで同一のファームウェアを持っていてもよいです。CANバスに乗ったすべてのノードがすべてデータを渡しているのを見ることができますから、シングルバスでこれらの同一のノードをプログラムすることはいっそう効率的であることができます。しかしながら、ノードあるいは多くのノードが特有な他の場合で、デバイスをプログラムすることは公開のピアツーピア通信にのみ必要です。これは最もシンプルなプログラミングシステムであり得ます、なぜならプログラミングソースはすべての知能を含んで、自由にターゲットメモリを操作することができたからです。直接ターゲットメモリを操作すること、手作業でデータをベリファイすることはCANバス上で重要な時間をとるから、この欠点は効率の欠如です。オペレーションをいっそう効率的にするために、プログラミングターゲットはセルフベリフィケーションのように、いくらかの知能を与えることができました。これは半分でCANバス上の本質的に時間をカットして、コミュニケーションを一方向けにするでしょう。全体的に、最良の節約は類似の、モジュールファームウェアのシステム上ですべてのノードを設計することです。それぞれのノードがただそのタスクのために必要とされるモジュールにだけを使うことができました、しかしノードの全グループは同時に更新されることができました。

いくつかのノードは使われない存在するファームウェアを持っていてもよいですから、ここの犠牲はプログラムのメモリアーバードです。

実行中のシステムをプログラミング

面白い状況がアクティブでそして機能しているシステムでブートルディングしています。この場合、ノードの1つ以上がそれらのファームウェアを更新するためにオフラインにされます、それでもなお全部のシステムの機能性は完全にディスエーブルではありません。これはもちろんターゲットノードあるいはノードがネットワークで結ばれたシステムの他の部分からいくつかの機能的な独立を持っていることを必要とします。アクティブなシステムにプログラムするとき対処すべきプライオリティ問題があります。例えば、何のプライオリティがシステムで重要なコミュニケーションに影響を与えないでブートルダーに与えられることができますか？ もしそれらの通常のアプリケーションを走らせている他のノードより高いプライオリティはブートルダーが走っているノードに与えられるなら、データがプログラミングターゲットに流出転送されているとき、受信はデータのために時間を取ってもよいです。それで、データ送信あるいは受信のために比較的低い反応時間(レイテンシー)を必要とするクリティカルなシステムが予期されたように機能し損ねてもよいです。反対の状況で、あまりにも低いプライオリティでプログラミングターゲットを割り当てることは、ただプログラミングソースとターゲットがデータを渡すためにアイドルバスを待っているというだけの理由で、極めて長いプログラミング時間に向かうことができました。アクティブなネットワークで、計画はプログラミングに十分なバス時間を提供するために必要です。1つの解決はただブートルダープログラミングオペレーションに比較的高い優先順位を与えて、プログラミングソースがCANバス上でデータの流れ転送している間に、他のアプリケーションの時間を「注入する」よう設計することです。それで、バス時間は常に利用可能で、プログラミングソースによって制御されています。注意深い計画でさえ、安全性がバス論争の結果として実際に危うくされる状態になるかもしれない。これらの場合、最も良い選択は「コンフィギュレーション」モード中にすべてのノードをネットワーク上に置いて、すべてのシステム機能をシャットダウンすることです。

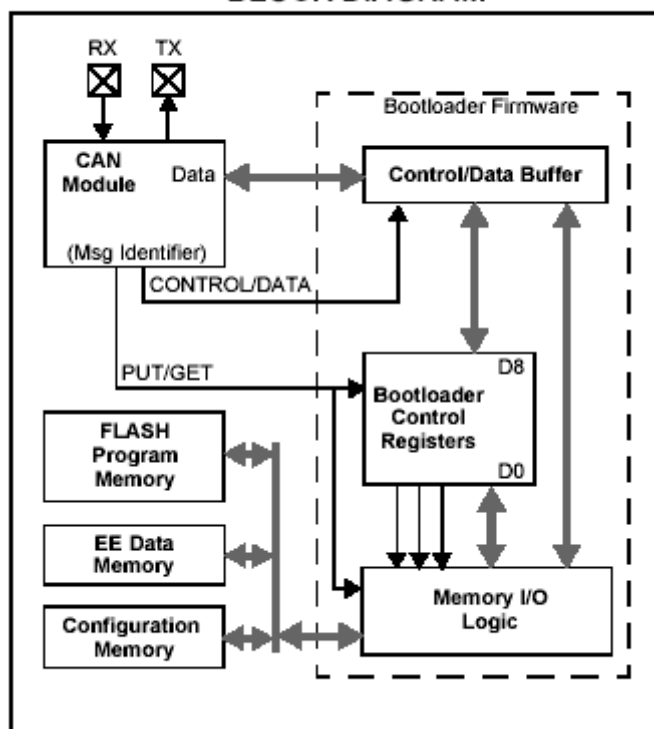
ブートモードエントリー

ブートモードエントリーはイベントによって決定されます。これはリセットデバイスの後に1つ以上のボタンを押すようなハードウェアイベントであり得ました。ブートモードに入るデバイスを示すのは同じくデータの特別なセットのような、ネットワークイベントであり得ました。1つの例が CAN ID中に直接マップされるネットワークブートIDです。それでキーは、特定のターゲットインフォメーションとともに、CANフレームのデータフィールドに埋め込まれることができました。キーインフォメーションは1つ以上のノードをブートモードに入れられました。

ブートルダーファームウェアの基本動作概観

CANブートルダーのオペレーションの概観が図1で示されます。CANメッセージアイデンテファイアとデータがCANモジュールを通して受信されます。アイデンテファイアでの1ビットがPUTあるいはGETデータを示すかに使われます。もう1つはメッセージがプログラムされるデータあるいはブートルダーコントロールインフォメーションと解釈されるはずであるかどうか決定するために使われます。自動的にデータを書くことはメモリ(フラッシュ、データEEPROM、あるいはコンフィギュレーションメモリ)に書き込む適切な機能呼び出します。コントロールレジスタに書くことはブートルダーのオペレーションを設定します。ブートルダーはオペレーション2つの相互に排他的なモードの1つをサポートするビルトタイムに構成されることができます。Pモード(あるいはPUTのみ)は、決してマイクロコントローラーのみがソースに「トークバック」ではなく、PUTコマンドを受け入れます。PGモードで、PUTとGET両方のコマンドが、ソースと共にターゲットのメモリから読み込んで、書くことを許して、受け入れられます。もっと詳細な説明が以降の項で提供されます。

FIGURE 1: BOOTLOADER FUNCTIONAL BLOCK DIAGRAM



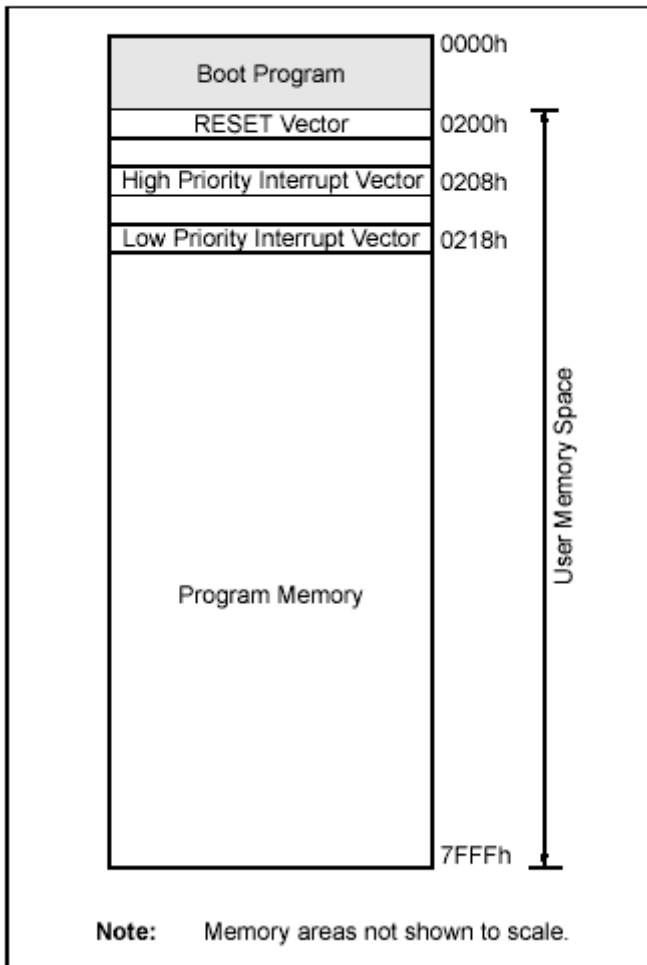
メモリ構成

プログラムメモリ使用法

現在、PIC18Fデバイスがブートブロックとしてプログラムメモリの最初の512バイトを予約します。将来のデバイスが、これらのデバイスをアプリケーション必要要件に依存して、これを拡張してもよいです。

このブートローダーは推薦されたオプションを使ってメモリの512バイト(あるいは256ワード)の現在の指名されたブートブロックを占めるよう設計されます。注意してください、しかしながら、いくつかのコンパイル時間オプションがブートブロックを越えてブートローダーを増大することができます。図2がPIC18F458のメモリーマップを示します。ブートエリアは偶然にブートプログラムに上書きすることを妨げるために守られるコードであり得ます。

FIGURE 2: PROGRAM MEMORY MAP OF THE PIC18F458



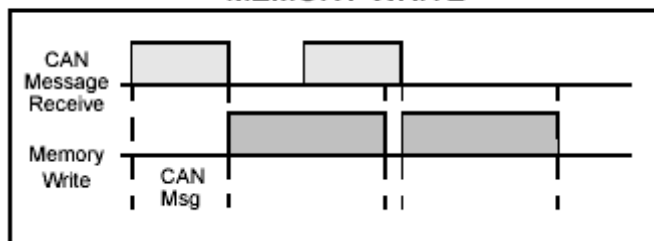
異なり、一度に1バイト書き込まれます。コンフィギュレーションビットが書き込まれる前に自動的に消されますから、イレース制御ビットはコンフィギュレーションメモリ上に影響を持たないでしょう。コンフィギュレーション設定へのアクセスを持つことは非常に強力です;それは同じく潜在的に非常に危険です。例えば、システムが20MHzクリスタルHSモードでランするよう意図されると想定してください。もしブートローダーがオシレーター設定をLPモードに変えるなら、システムは作用することを終えるでしょう - 基本的にブートローダーを含めて、不適当に1ビットを変えることによって、システムはつづがされました。

いくつかのコンフィギュレーションビットがノーマルモードで一つの方向ビットであることを指摘することは同じく重要です;それらはただ1つのステートに変えられるだけで、元に戻されることができません。コンフィギュレーションレジスタ5Lと5Hでのコード保護ビットは良い例です。もしのコード保護タイプが1ブロックでイネーブルであるなら、デバイスプログラマーなしでディスエーブルであるはずがありません。本質的に、ブートローダーはコード保護を無効にすることができません。デバイスID (アドレス3FFFFEhと3FFFFFh)は同じくプログラムメモリであると思われる。それらがアクセスされることできる間に、しかしながらそれらはリードオンリーであって、そして変えられることができません。

書き込み待ち時間(レーテンシー)

データを書くとき、プログラミングソースがプログラミングオペレーションを完了するために待たなくてはならない特定の時間があります。幸いに、CANモジュールは実際に受信されたデータをバッファに入れます;そのために受信することは実際にメモリ書出し操作(図6)とオーバーラップすることができます。一般に、それはプログラムメモリ書出し操作のためにおよそ2ミリ秒を要します、他方EEDATAはおよそ4ミリ秒を要します。すべてのPIC18Fデバイスが同じ時間仕様を持っているわけではありません、それで特定のデバイスが使われる書き込み時間をベリファイすることは重要です。

FIGURE 6: CAN RECEIVE VS. MEMORY WRITE



書き込みコード

ブートローダーはアプリケーションがブートローダーは何をしているかについて極めて少しの関心を払って開発されないことを意味する別個の構成要素として機能します。これはそれがそうであるべきであるようにです;イベントがブートオペレーションを始めるまで、ブートローダーは休眠中のコードであるべきです。理想的な環境下でブートローダーコードが決してアプリケーションの意図された通常のエネーションの間にランしているべきではありません。存在するブートローダーでアプリケーションを開発するとき、いくつかの基本的な原則が念頭におかれなくてはなりません。

アセンブリで書く

アセンブリを書き加えるとき、ブートブロックと新しいベクトルは考慮されなくてはなりません。モジュールコードで、これは一般にただリンクスクリプトファイルをプロジェクトと交換することについての問題です。例が付録Cで与えられます。もし絶対番

地がコードセクションに割り当てられるなら、アドレスはブートブロック上にどこかにポイントしなくてはなりません。絶対のアセンブリを書く人たちのために、思い出すために必要なすべては新しいRESETベクトルが200hであるということです、そして割り込みベクトルは208hと218hです。ブートローダー以外のコードがブートブロックに存在するべきではありません。

Cで書く

MPLAB C18 CコンパイラをアプリケーションのためにPIC18Fファームウェアを開発するために使うとき、標準的なスタートアップオブジェクト(c018.o あるいは c018i.o)は新しいRESETベクトルで再構築されなくてはなりません。モジュールのアセンブリのように、リンカーファイルは保護されたブートブロックと新しいベクトルを含むように変えられなくてはなりません。付録Cが例リンカーファイルを見せます。他のコンパイラのユーザーがどのようにスタートアップコードとベクトルを変えるべきか決定するためにコンパイラのソフトウェアユーザーガイドに問い合わせるべきです。

ブートローダーのリエントリ

もし必要性がアプリケーション(そして通常それはします)からブートモードに再び入るためにあるなら、データEEPROMの最後のロケーションがFFhにセットされなくてはなりません。例1のコードはこれがどのようにアプリケーションでされるかもしれないか実演します。ブートローダーがリセットされた条件を想定しますから、リセットされた指示が、最後のロケーションを設定した後で、始められるべきです。

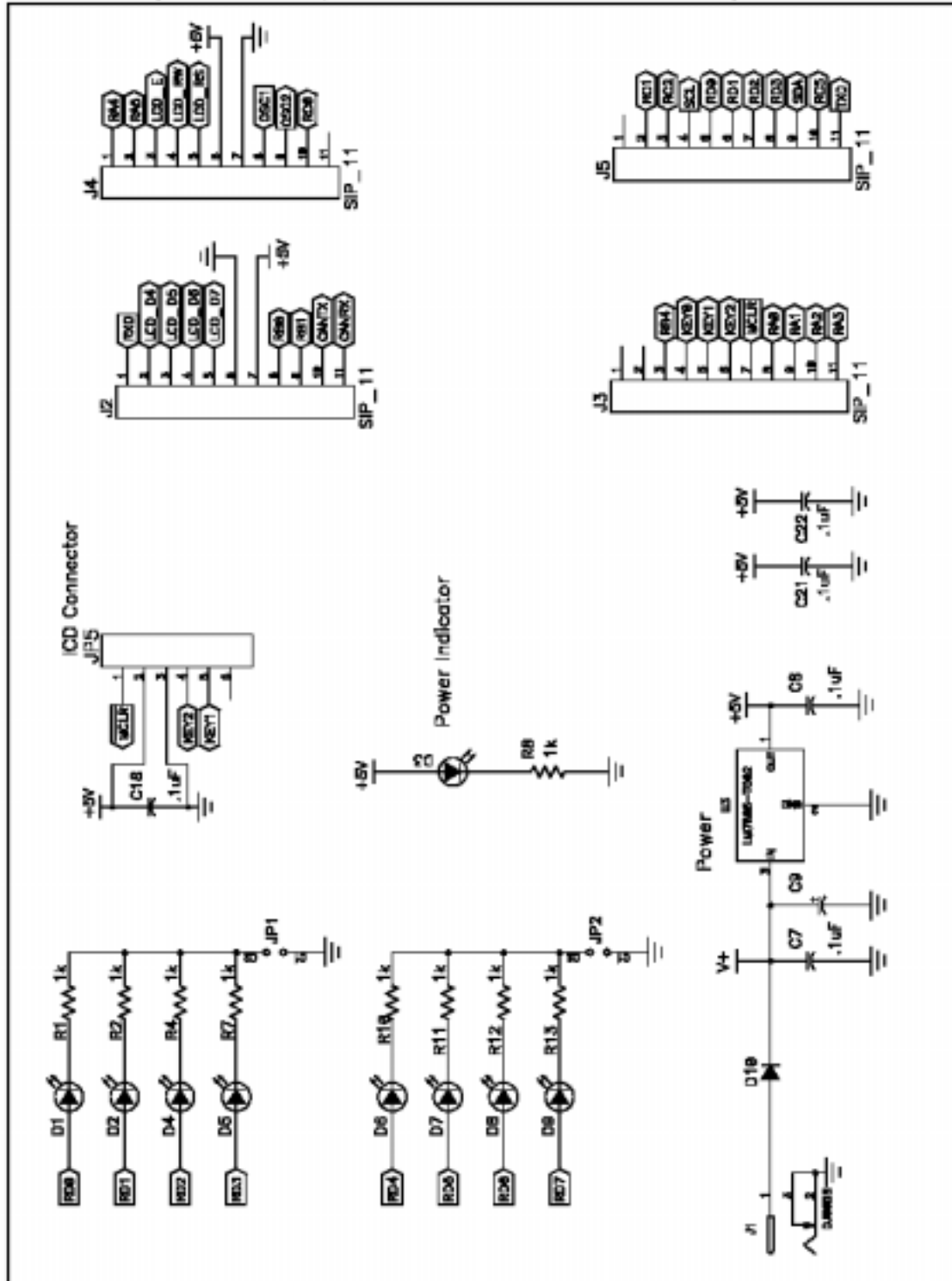
EXAMPLE 1: SETTING THE LAST LOCATION OF THE DATA MEMORY

```
SETF    EEADR      ; Point to the last byte
SETF    EEADRH
SETF    EEDATA     ; Bootmode control byte
MOVLW  b'00000100 ; Setup for EEData
MOVWF   EECON1
MOVLW  0x55       ; Unlock
MOVWF   EECON2
MOVLW  0xAA
MOVWF   EECON2
BSF    EECON1, WR ; Start the write
NOP
BTFSC  EECON1, WR ; Wait
BRA    $ - 2
RESET
```

デバッグング

たいていの状態のために、MPLAB ICD 2あるいはICEデバイスでアプリケーションをデバッグするためにメモリにブートローダーファームウェアを持っていることは必要ではありません。しかしながら、分岐文が新しい指定されたベクトルにもっていくべきハードウェアベクトルに挿入されなくてはなりません。正確にブートローダーエントリーにスタートアップタイミング合致を持っていることは同じく有用であることができます。アプリケーションの開発が終了しているとき、分岐を削除してプロジェクトを再構築するか、あるいはブートブロック上にただメモリだけを書き出してください。このコードはそれでそれらのファームウェアを更新している人たちに配布されることができます。

FIGURE A-2: BOOTLOADER HARDWARE INTERFACE FOR CAN NETWORKS
(POWER SUPPLY, DISPLAYS AND CONNECTION HEADERS)



詳細はホームページ <http://www5b.biglobe.ne.jp/~tekhanzo/>

をご確認ください。

不許複製

アプリケーションノート AN247 完全日本語訳（19頁）中の抜粋 サンプル

Japanese Translation Copy Rights©2004-2007 Tech - Hanzougane Yoshiaki Morohashi

発行 2009年1月28日 初版発行

翻訳者 諸橋 義明

発行元 テック・ハンゾウガネ(個人事業者)

〒940-0213

新潟県長岡市栃尾山田町 6-53

TEL 0258 (53) 0082 E-mail tekhanzo@mta.biglobe.ne.jp

HP アドレス <http://www5b.biglobe.ne.jp/~tekhanzo/>
