

@CROシリーズ

---

*ACRO741-00*

---

スタートアップガイド



## 目 次

|        |                        |    |
|--------|------------------------|----|
| 1.     | 概要.....                | 1  |
| 1.1.   | 動作環境.....              | 1  |
| 1.2.   | アプリケーション.....          | 1  |
| 2.     | インストール手順の概要.....       | 2  |
| 3.     | インストール手順.....          | 3  |
| 3.1.   | コンパイラのインストール.....      | 3  |
| 3.1.1. | インストール時における注意事項.....   | 3  |
| 3.2.   | サポートソフトウェアのインストール..... | 3  |
| 3.3.   | デバイスドライバのインストール.....   | 3  |
| 3.4.   | @CROボード実装.....         | 3  |
| 3.5.   | PC接続.....              | 4  |
| 3.6.   | @CROボード動作確認.....       | 9  |
| 3.7.   | エラーについて.....           | 10 |
| 4.     | インストール後のフォルダ構造.....    | 11 |



## 1. 概要

本書は、ACRO741-00をはじめてご利用いただく際、ACRO741-00の導入から動作確認までの作業を容易に進めるための案内役をするものです。

ACRO741-00の導入には、専用のコンパイラ、ボード付属のツール類、ライブラリ等のインストールや、ボードの動作確認等の作業が必要になります。本書では、その全てについて、順を追って説明しています。

### 1.1. 動作環境

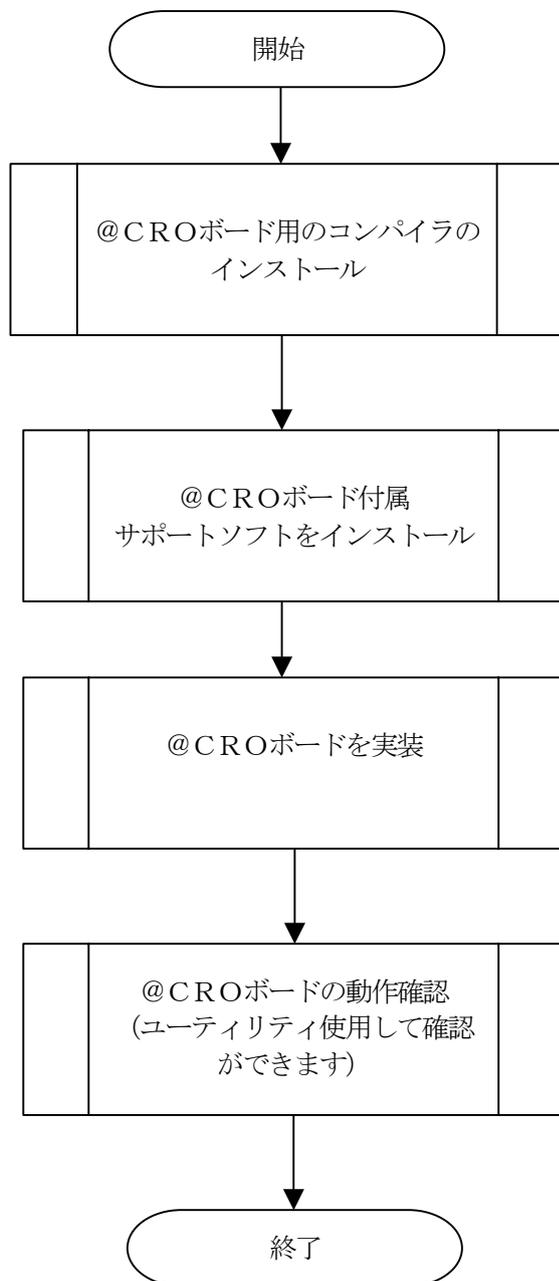
1. OS : Windows XP (SP2, SP3) / Windows Vista(32bit) / Windows 7(32bit)
2. メモリ : XP:256MB 以上、Vista:512MB 以上、7 : 1GB 以上
3. 必要ディスクサイズ : 10MB
4. ホスト通信 : USB1.1 以上 (推奨 2.0)

### 1.2. アプリケーション

1. コンパイラ : FreeScale 社製 CodeWarrior V8.5 以上

## 2. インストール手順の概要

ACRO741-00の導入に必要な作業は以下の通りです。



### 3. インストール手順

#### 3.1. コンパイラのインストール

FreeScale 社製 Code Warrior をインストールしてください。

##### 3.1.1. インストール時における注意事項

- 1) インストールの際フォルダ名に空白を含むと、MATLAB 対応ライブラリ (弊社製) のビルド時にエラーとなりますので、名称に空白を含まないフォルダへインストールしてください。

(例) <誤> “c:\Program Files\Metrowerks\CodeWarrior PwerPC comm. V8.5”  
<正> “c:\Metrowerks\CodeWarriorPowerPCcomm”

#### 3.2. サポートソフトウェアのインストール

ここでは、@CROボード付属のサポートソフトウェアをインストールします。

- 1) CD-ROMドライブに、ボード付属の“@CROシリーズサポートソフト”ディスクを挿入します。
- 2) 自動でセットアッププログラムが起動します。  
(自動起動しない場合は、CD-ROM内 setup.exe を起動してください。)
- 3) セットアッププログラムが起動されたら、セットアッププログラムの指示に従ってセットアップを完了します。

#### 3.3. デバイスドライバのインストール (Ver1.01.00 以降)

次に、USB デバイスドライバをインストールします。

- 1) インストールフォルダ\Driver\Setup.exe をダブルクリックし実行してください。
- 2) ガイドに従ってインストールします。

#### 3.4. @CROボード実装

次に、@CROボードを実装します。

- 1) ACRO741-00のDSWを設定します。(“ACRO741-00 ハードウェアマニュアル”参照)
- 2) ACRO741-00をセットします。I/Oボードも必要に応じてセットしてください。実装形態(PC内部に実装時はPCの電源、拡張ラックに実装時は拡張ラックの電源)にあわせて電源をOFFにして作業をしてください。  
[注] ACRO741-00とI/Oボードは専用ケーブルにて接続してください。
- 3) PC又は拡張ラックの電源をONします。通信形態によりドライバのインストール要求がありますので、インストールしてください。(3.5. PC接続 参照)

### 3.インストール手順

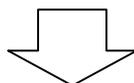
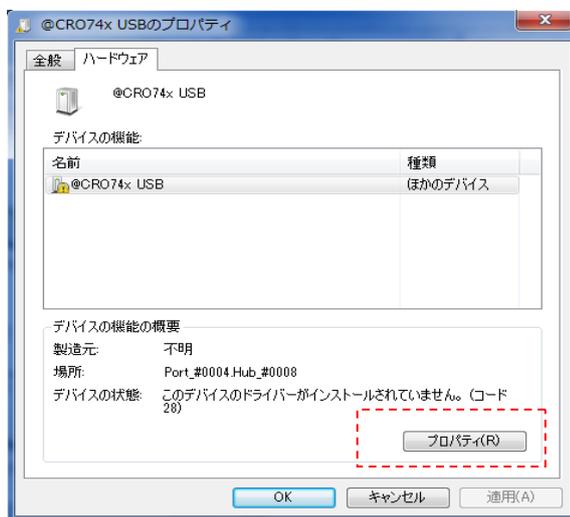
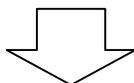
#### 3.5. PC接続

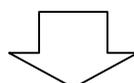
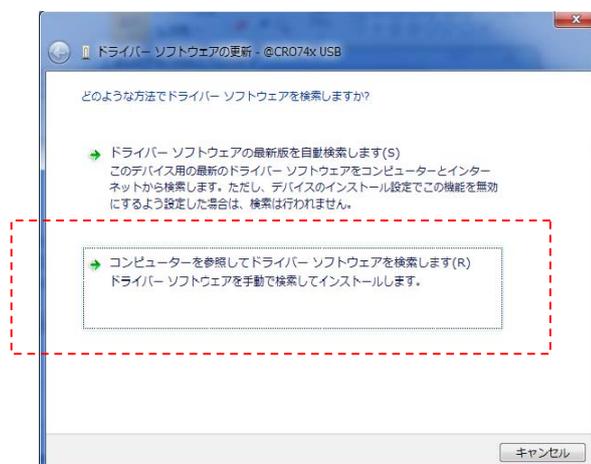
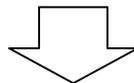
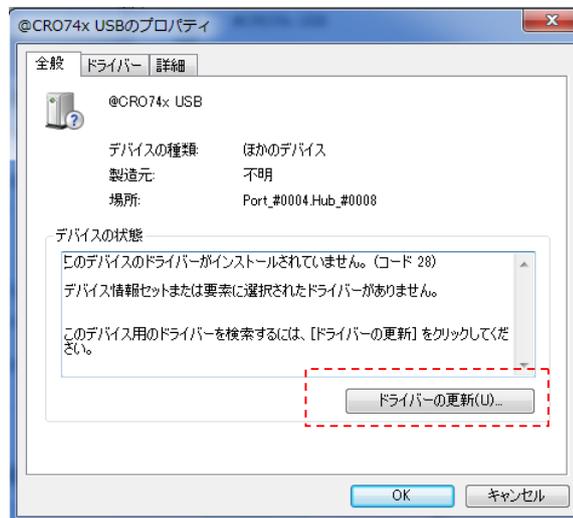
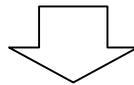
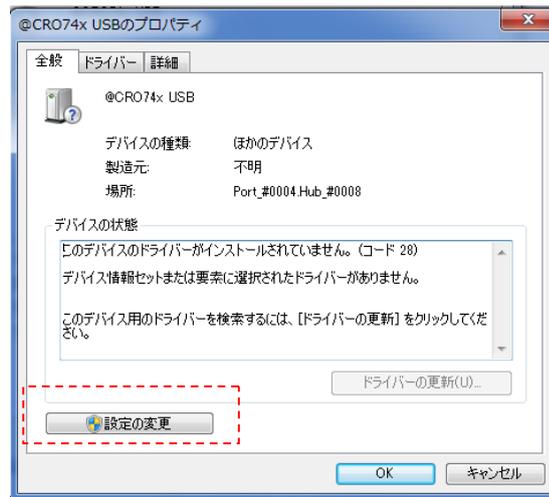
@CRO741の電源ONにし、USBケーブルをホスト(PC)と接続すると下記のようなデバイスドライバインストールガイドが表示されますので、サポートソフトインストールにコピーされているドライバを使用してインストールしてください。ドライバは、”Windows システムフォルダ¥system32¥drivers”にAP74XUSB.sysがコピーされています。

なお、“3.3. デバイスドライバのインストール”にて予めドライバがインストールされている場合は、インストールガイドなしでドライバが自動インストールされます。

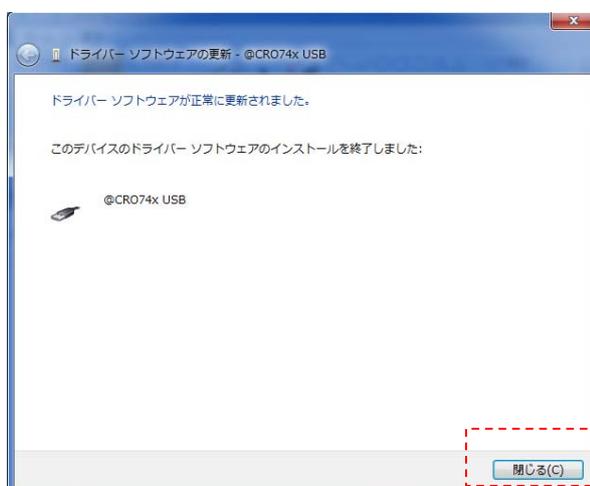
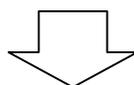
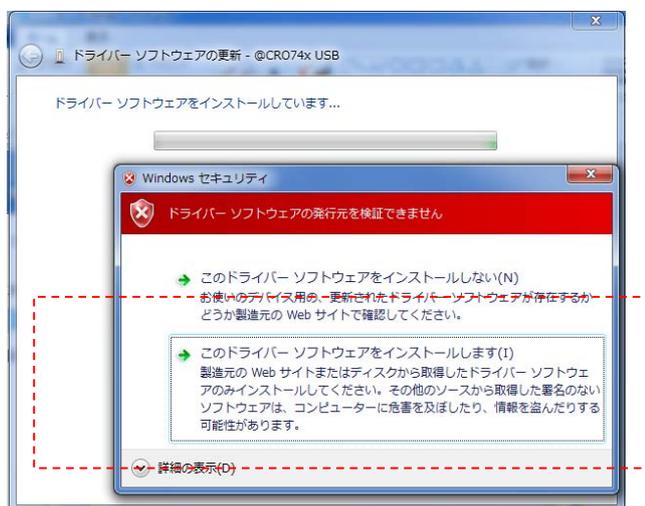
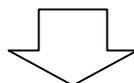
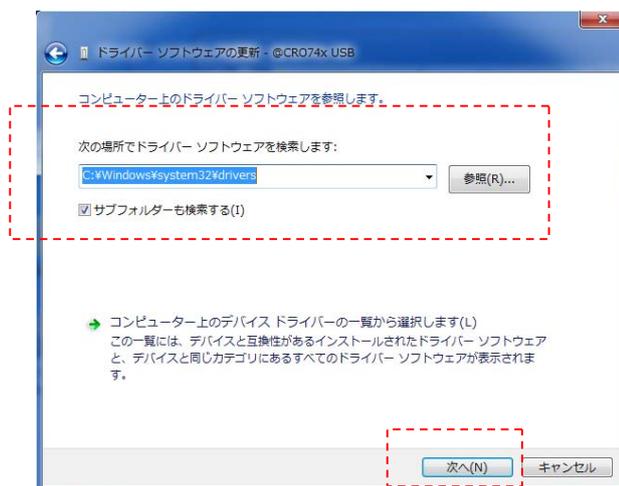
#### 【Windows 7】

- ※  を選択し進めてください。
1. スタート→デバイスとプリンタを選択してください。
  2. “@CRO74xUSB”を選択しダブルクリックしてください。

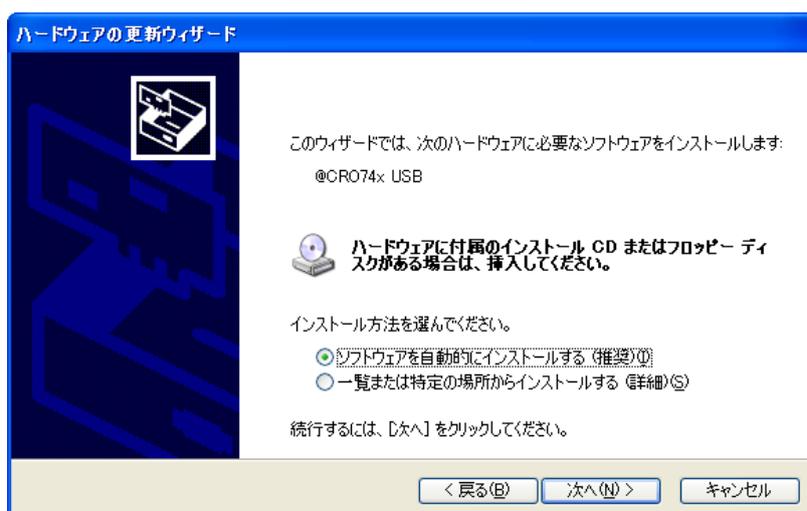
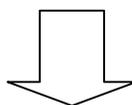
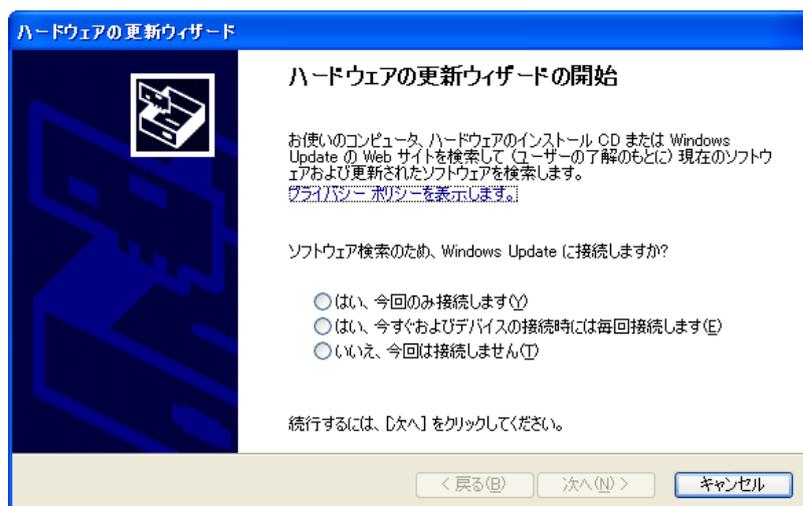




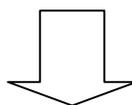
### 3. インストール手順

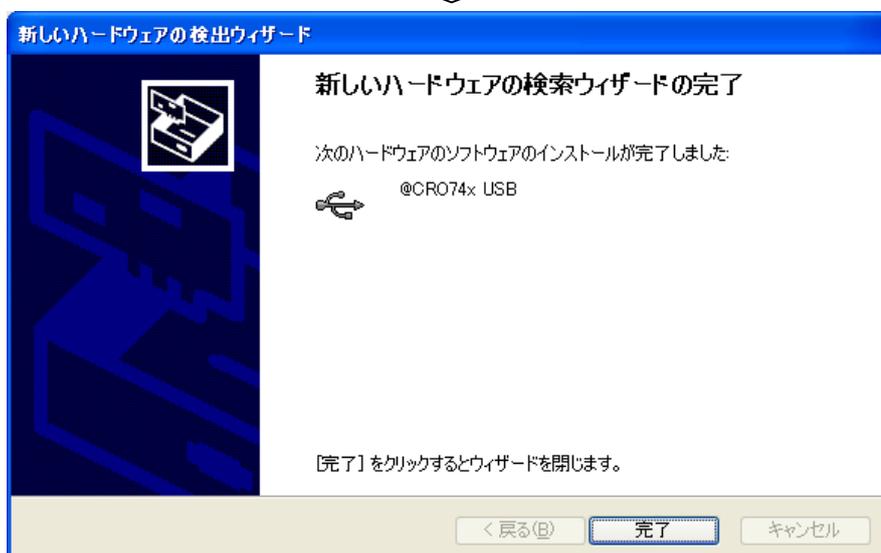
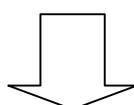
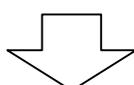
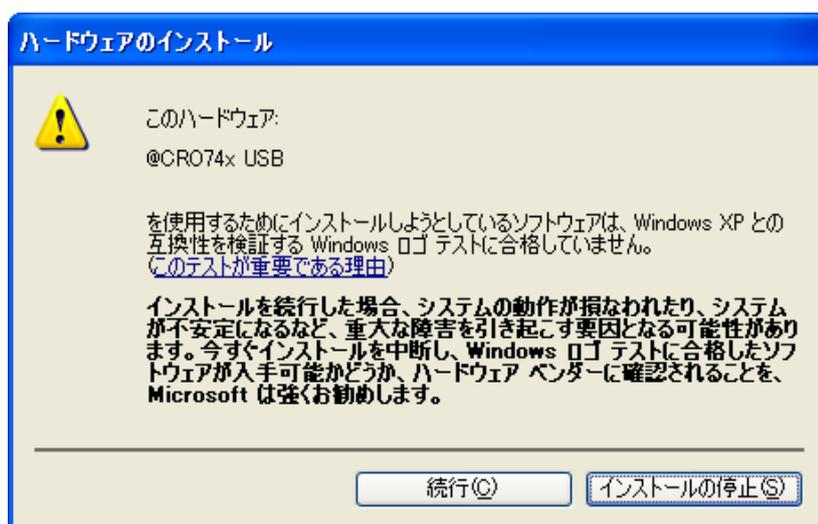


## 【Windows XP、Windows Vista】



\* ドライバが検出されないエラーが表示された場合は、”Windows システムフォルダ”¥system32¥drivers”を指定してください。



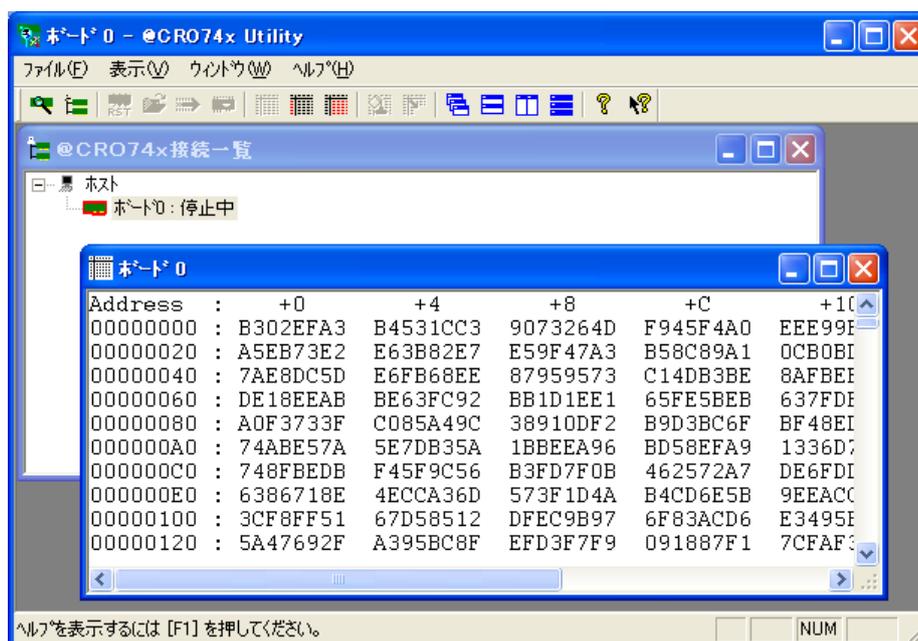


## 3.6. @CROボード動作確認

次に、@CROボードの動作確認をします。3.2でインストールした“ACRO741-00ユーティリティ”にて動作確認をしてください。

- 1) 「スタート」→「すべてのプログラム」→「@CRO74X Support Soft Ver .xx.xx.xx」→「@CRO74Xユーティリティ」を選択して、ソフトを起動してください。「操作」→「メモリの表示」を選択し、メモリ内容の表示をします。「図1ユーティリティ」のように何らかのデータが表示されれば動作確認OKです。

図 1 エラーダイアログ



#### 3.7. エラーについて

- 1) 前回実行時のエラー等で異常終了をすると、以下の“図2エラーダイアログ”が表示されます。この場合は、「スタート」→「すべてのプログラム」→「@CRO Support Soft Ver .xx.xx.xx」→「@CRO74X リジューム」を実行してください。実行が完了すると“図3リジューム完了ダイアログ”が表示されますので、**Enter** キーを押して終了してください。リジュームにて復旧しない場合は、一度@CRO ボードの電源を **OFF** にしてください。

図 2 エラーダイアログ

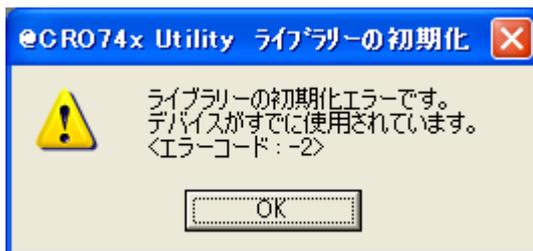
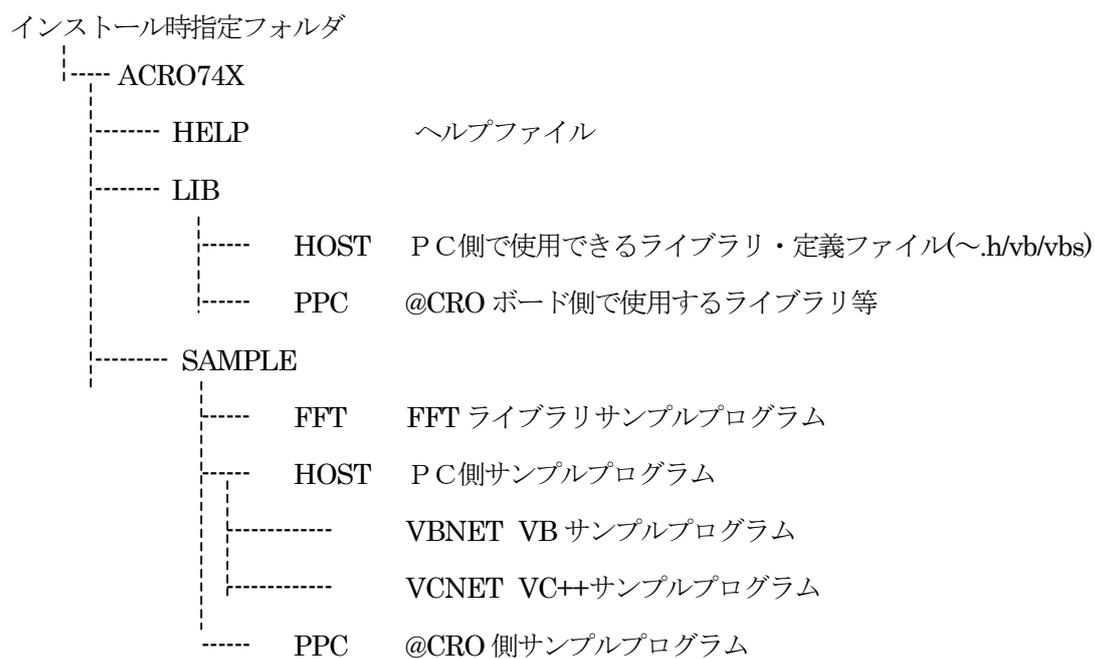


図 3 リジューム完了ダイアログ



- 4) ユーティリティにて、動作確認の NG の場合は“Can Not read from @CRO74x memory ”等エラー表示がされます。その際は、@CRO ボードの DSW、USB 接続等の確認をお願いします。（“ACRO741-00 ハードウェアマニュアル”参照）

## 4. インストール後のフォルダ構造



C:\Windows\System32\drivers(システムフォルダ\drivers)

- AP74xUSB.sys        デバイスドライバ
- AP74xUSB.inf        デバイスドライバインストール用ファイル



## 【 改 訂 履 歴 】

| 改訂番号 | 改訂日付       | 改訂内容             |
|------|------------|------------------|
| 初版   | 2010.5.24  | 初版               |
| 2版   | 2011.08.05 | デバイスドライバインストール追加 |
|      |            |                  |

- ・本製品及び本マニュアルの内容は製品の改良のため予告無しに変更される事がありますのでご了承下さい。
- ・本製品及び本マニュアルの内容は著作権法により保護されています。許可なく複製する事はできません。

**ACRO741-00**  
**スタートアップガイド**

**中部電機株式会社**

〒440-0004 愛知県豊橋市忠興3丁目2-8

TEL <0532>61-9566 FAX <0532>63-1081

URL : <http://www.chubu-el.co.jp>

E-mail : [csg@chubu-el.co.jp](mailto:csg@chubu-el.co.jp)

2010.05 第1版

2011.08 第2版

@CROシリーズ

ACRO741-00

ハードウェアユーザーズ・マニュアル



# 目 次

|        |                                 |    |
|--------|---------------------------------|----|
| 1.     | 仕 様.....                        | 1  |
| 1.1.   | 標準仕様.....                       | 1  |
| 1.2.   | 拡張 I/O.....                     | 2  |
| 1.3.   | その他.....                        | 2  |
| 2.     | ハードウェア.....                     | 3  |
| 2.1.   | ディップスイッチとショートピンの初期設定.....       | 3  |
| 2.2.   | コネクタ、ディップスイッチの配置.....           | 5  |
| 2.3.   | 入出力信号の接続例.....                  | 6  |
| 2.3.1. | パラレル入力 (TTLレベル).....            | 6  |
| 2.3.2. | パラレル出力 (TTLレベル).....            | 6  |
| 2.3.3. | PWM (TTLレベル出力).....             | 6  |
| 2.3.4. | カウンタ A, B, Z相 (RS422レベル入力)..... | 6  |
| 2.3.5. | カウンタ 外部入力指定ビット (TTLレベル入力).....  | 7  |
| 2.3.6. | 高速用CAN.....                     | 7  |
| 2.3.7. | 低速用CAN.....                     | 7  |
| 3.     | ホストパソコンと本ボードへの接続.....           | 8  |
| 3.1.   | USB接続.....                      | 8  |
| 4.     | メモリマップ.....                     | 8  |
| 4.1.   | メモリサイズ.....                     | 8  |
| 4.2.   | メモリーマップ.....                    | 9  |
| 4.3.   | 標準メモリー.....                     | 10 |
| 4.4.   | ユーザー用BOOT ROM.....              | 10 |
| 4.5.   | レジスタ.....                       | 10 |
| 4.5.1. | 制御レジスタ (アドレス 8F000008h).....    | 10 |
| 4.5.2. | 機種コードレジスタ (アドレス 8F00000Ch)..... | 12 |
| 4.6.   | 割り込み.....                       | 12 |
| 4.7.   | ソフトリセット.....                    | 13 |
| 4.8.   | ウォッチドックリセット.....                | 13 |
| 4.9.   | 拡張バス.....                       | 13 |
| 4.9.1. | データボード.....                     | 14 |
| 4.9.2. | 拡張 I/Oボード.....                  | 15 |
| 5.     | 動作設定に関するスイッチの設定.....            | 17 |
| 5.1.   | ボード番号、モード設定 DSW1.....           | 17 |
| 5.1.1. | ボード番号の設定.....                   | 18 |
| 5.1.2. | ホスト接続.....                      | 18 |
| 5.1.3. | 起動動作モード.....                    | 18 |
| 5.2.   | ユーザーROM書き込み設定 DSW2.....         | 19 |
| 5.3.   | ハードリセット SW1.....                | 19 |
| 6.     | 標準実装 I/O.....                   | 20 |
| 6.1.   | 精度と機能.....                      | 20 |
| 6.2.   | PIO 入力と出力.....                  | 21 |
| 6.2.1. | PIO 8bit入力.....                 | 21 |
| 6.2.2. | PIO 8bit出力.....                 | 21 |

|        |   |    |
|--------|---|----|
| 6.2.3. | PIOのメモリマップ                              | 21 |
| 6.3.   | PWM                                     | 22 |
| 6.3.1. | 3相モードと単相モード                             | 22 |
| 6.3.2. | 出力波形                                    | 23 |
| 6.3.3. | 同期パルス出力                                 | 26 |
| 6.3.4. | 設定データの反映タイミング                           | 26 |
| 6.3.5. | 初期設定                                    | 27 |
| 6.3.6. | PWMのメモリマップと制御レジスタ                       | 29 |
| 6.4.   | カウンタ                                    | 36 |
| 6.4.1. | カウンタのアップ・ダウン                            | 36 |
| 6.4.2. | カウンタの通倍                                 | 36 |
| 6.4.3. | カウンタリセット                                | 37 |
| 6.4.4. | 初期設定                                    | 38 |
| 6.4.5. | カウンターのメモリマップと制御レジスタ                     | 40 |
| 7.     | 通信ポート                                   | 44 |
| 7.1.   | USB通信                                   | 44 |
| 7.2.   | CAN通信                                   | 44 |
| 7.3.   | RS232C通信                                | 44 |
| 8.     | ホストコンピュータより本ボードを分離して動作させる方法 (スタンドアロン機能) | 45 |
| 9.     | MetroWerks社製 CodeWarrior JTAG使用         | 45 |
| 10.    | コネクタのピン配列とピン配置                          | 46 |
| 10.1.  | 標準実装 I/O 入出力コネクタ (CN12) のピン配列とピン配置      | 46 |
| 10.2.  | CANコネクタ (CN13) のピン配置とピン配置               | 47 |
| 10.3.  | RS232Cコネクタ (CN14) のピン配置とピン配置            | 47 |
| 11.    | 添付品                                     | 48 |

## 図 目 次

|         |                           |    |
|---------|---------------------------|----|
| 図 2-1   | ディップスイッチの出荷時設定.....       | 3  |
| 図 2-2   | 出荷時のショートピンの設定.....        | 4  |
| 図 2-3   | 配置図.....                  | 5  |
| 図 2-4   | パラレル入力回路.....             | 6  |
| 図 2-5   | パラレル出力回路.....             | 6  |
| 図 2-6   | PWM出力回路.....              | 6  |
| 図 2-7   | カウンタ入力回路.....             | 6  |
| 図 2-8   | 外部指定入力.....               | 7  |
| 図 2-9   | 高速用CAN 終端抵抗.....          | 7  |
| 図 2-10  | 低速用CAN 終端抵抗.....          | 7  |
| 図 5-1   | DSW1の設定.....              | 17 |
| 図 5-2   | DSW2の設定.....              | 19 |
| 図 6-1   | 3相PWM出力の回路構成.....         | 22 |
| 図 6-2   | 単相PWM出力の回路構成.....         | 22 |
| 図 6-3   | PWM出力波形 フルブリッジの正出力.....   | 23 |
| 図 6-4   | PWM出力波形 フルブリッジの逆出力.....   | 24 |
| 図 6-5   | PWM出力波形 ハーフブリッジの正出力.....  | 24 |
| 図 6-6   | PWM出力波形 ハーフブリッジの逆出力.....  | 25 |
| 図 6-7   | 同期パルス出力波形.....            | 26 |
| 図 6-8   | D u t y ・レジスタ.....        | 31 |
| 図 6-9   | 同期パルス出力・レジスタ.....         | 33 |
| 図 6-10  | AB相入力とカウントの関係.....        | 36 |
| 図 6-11  | カウンタの逡倍.....              | 36 |
| 図 6-12  | ABZ相のカウントリセット条件.....      | 37 |
| 図 1 0-1 | CN1 2のコネクタを挿入方向からみた図..... | 46 |
| 図 1 0-2 | CN1 3のコネクタを挿入方向からみた図..... | 47 |
| 図 1 0-3 | CN1 4のコネクタを挿入方向からみた図..... | 47 |

## 表 目 次

|        |  |    |
|--------|--|----|
| 表 4-1  | メモリサイズ.....                              | 8  |
| 表 4-2  | CPU (PPC) から見たメモリマップ.....                | 9  |
| 表 4-3  | 8F000008 bit割付.....                      | 10 |
| 表 4-4  | ボード番号.....                               | 11 |
| 表 4-5  | 機種コード.....                               | 12 |
| 表 4-6  | 割り込み受付.....                              | 12 |
| 表 4-7  | 拡張バス (8240 0000h~83FF FFFFh) 割付アドレス..... | 13 |
| 表 4-8  | CN1 0のピン配置.....                          | 14 |
| 表 4-9  | PPCと拡張I/Oボードのアドレス対応表.....                | 15 |
| 表 4-10 | CN1 1のピン配置.....                          | 16 |
| 表 5-1  | DSW1.....                                | 17 |
| 表 5-2  | DSW1 1-4ビットの設定.....                      | 18 |
| 表 5-3  | DSW2.....                                | 19 |
| 表 6-1  | PIOメモリーマップ.....                          | 21 |
| 表 6-2  | モード選択、出力許可・レジスタ.....                     | 27 |
| 表 6-3  | 設定データ反映タイミング・レジスタ.....                   | 27 |

|        |                   |    |
|--------|-------------------|----|
| 表 6-4  | PWMのメモリマップ (1)    | 29 |
| 表 6-5  | PWMのメモリマップ (2)    | 30 |
| 表 6-6  | モード選択、出力許可・レジスタ   | 34 |
| 表 6-7  | 設定データ反映タイミング・レジスタ | 35 |
| 表 6-8  | カウンタリセット条件        | 37 |
| 表 6-9  | カウンタ逡倍選択・レジスタ     | 38 |
| 表 6-10 | カウンタリセット許可・レジスタ   | 39 |
| 表 6-11 | カウンタのメモリマップ       | 40 |
| 表 6-12 | カウンタリセット許可・レジスタ   | 41 |
| 表 6-13 | カウンタリセット条件        | 41 |
| 表 6-14 | カウンタ逡倍選択・レジスタ     | 42 |
| 表 6-15 | 外部指定ビット入力の状態・レジスタ | 43 |
| 表 7-1  | CAN 終端抵抗          | 44 |
| 表 7-2  | 低速用CAN バッテリー電圧設定  | 44 |
| 表 8-1  | ブートROM書き込み可       | 45 |
| 表 8-2  | BOOT ROM起動        | 45 |
| 表 10-1 | CN12のピン配列         | 46 |
| 表 10-2 | CN13のピン配置         | 47 |
| 表 10-3 | CN14のピン配置         | 47 |
| 表 11-1 | 添付品一覧             | 48 |

## 1.仕様

## 1.1.標準仕様

- CPU MPC7457 1.25GHz を搭載。
- メモリ
- |          |                                     |
|----------|-------------------------------------|
| L1 キャッシュ | : 32K/32Kbyte                       |
| L2 キャッシュ | : 512Kbyte                          |
| メモリ      | : SC 内部 RAM 256Kbyte                |
|          | : DPSRAM 64Kbyte×2(USB, CAN 用)      |
|          | : プライベートメモリー 2Mbyte                 |
|          | : DDR SDRAM 512Mbyte (オプション 1Gbyte) |
- I/O
- |               |                              |
|---------------|------------------------------|
| ユーザー用 BOOTROM | : FROM 4Mbyte                |
| パラレル入力        | : 8bit 1ch TTLレベル            |
| パラレル出力        | : 8bit 1ch TTLレベル            |
| アップダウンカウンタ    | : 32bit 4ch RS422レベル 5MHz    |
| PWM           | : 3相×3ch か単相×4ch 24Hz~120KHz |
- 通信ポート
- |        |                     |
|--------|---------------------|
| RS232c | : 1ch               |
| CAN    | : 2ch 高速用、低速用 各 1ch |
| USB    | : 1ch               |
- 並列処理機能 16台までのマルチプロセッシング可能。
- スタンドアロン機能  
 ホストコンピュータから分離して、パワーオンスタートさせることができる。  
 BOOT用ユーザー FLASH ROM : 4Mbyte 標準実装。
- 拡張バスインターフェース
- |           |                                |
|-----------|--------------------------------|
| アドレス信号    | : 24bit                        |
| データ信号     | : 32bit                        |
| 制御信号      | : 3本 *EXTWR、*EXTSTRB、*EXTRESET |
| 割り込み      | : INT1・INT2・INT3               |
| 拡張ボード接続台数 | : 最大4台                         |

ホストインターフェース

- ホスト通信 USB
- JTAGコネクタ  
 Metrowerks社製JTAGのコネクタあり。
- 電源電圧 DC5V±5%(7A)、DC3.3V±5%(7A)、DC12V±10%(0.1A)
- 動作温度 5~40℃
- 湿度 30~80% 結露しない事

## 1. 仕様

---

### 1.2. 拡張I/O

- ターボポート 増設A/D・D/AボードはACR0741-01が使用できます。
- 拡張I/O 拡張I/OボードはACR0741-06、ADSP324シリーズが使用できます。
- 拡張バスケーブル 拡張I/Oボードを使用する時には必要となります。(ACR0741-15)

### 1.3. その他

【注意】 ソフト開発にはMetrowerks社のCコンパイラを推奨しています。

## 2. ハードウェア

## 2.1. ディップスイッチとショートピンの初期設定

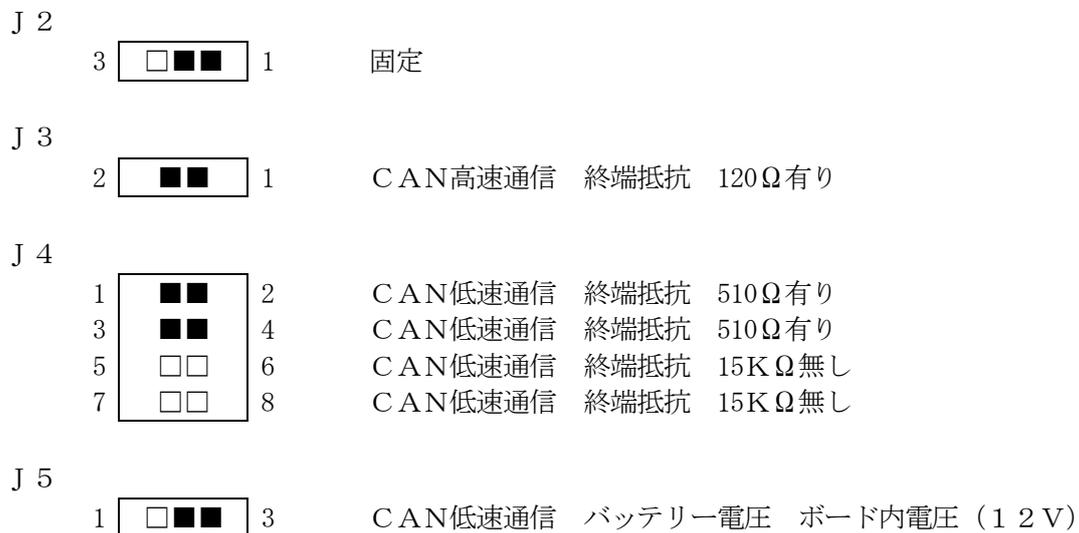
本ボードにはディップスイッチが2個【DSW1】【DSW2】と ショートピンが8個【J2～J6】【J8～J10】があります。

本ボードをお使いになる前にかならず使用するシステムに対応した設定を行なって下さい。尚、出荷時の設定状態を下記に示します。



図 2-1 ディップスイッチの出荷時設定

■ ■ : ショート    □ □ : オープン



## 2. ハードウェア

---

システムクロック 133MHz

J6

|    |                          |                          |    |    |
|----|--------------------------|--------------------------|----|----|
| 1  | <input type="checkbox"/> | <input type="checkbox"/> | 2  | 固定 |
| 3  | <input type="checkbox"/> | <input type="checkbox"/> | 4  | 固定 |
| 5  | <input type="checkbox"/> | <input type="checkbox"/> | 6  | 固定 |
| 7  | <input type="checkbox"/> | <input type="checkbox"/> | 8  | 固定 |
| 9  | <input type="checkbox"/> | <input type="checkbox"/> | 10 | 固定 |
| 11 | <input type="checkbox"/> | <input type="checkbox"/> | 12 | 固定 |
| 13 | <input type="checkbox"/> | <input type="checkbox"/> | 14 | 固定 |

システムクロック 100MHz

J6

|    |                                     |                                     |    |    |
|----|-------------------------------------|-------------------------------------|----|----|
| 1  | <input type="checkbox"/>            | <input type="checkbox"/>            | 2  | 固定 |
| 3  | <input type="checkbox"/>            | <input type="checkbox"/>            | 4  | 固定 |
| 5  | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 6  | 固定 |
| 7  | <input type="checkbox"/>            | <input type="checkbox"/>            | 8  | 固定 |
| 9  | <input type="checkbox"/>            | <input type="checkbox"/>            | 10 | 固定 |
| 11 | <input type="checkbox"/>            | <input type="checkbox"/>            | 12 | 固定 |
| 13 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 14 | 固定 |

J8

|   |                                     |                                     |   |    |
|---|-------------------------------------|-------------------------------------|---|----|
| 1 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 2 | 固定 |
| 3 | <input type="checkbox"/>            | <input type="checkbox"/>            | 4 | 固定 |

J9

|   |                                     |                                     |                          |   |    |
|---|-------------------------------------|-------------------------------------|--------------------------|---|----|
| 1 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | 2 | 固定 |
|---|-------------------------------------|-------------------------------------|--------------------------|---|----|

図 2-2 出荷時のショートピンの設定

## 2.2. コネクタ、ディップスイッチの配置

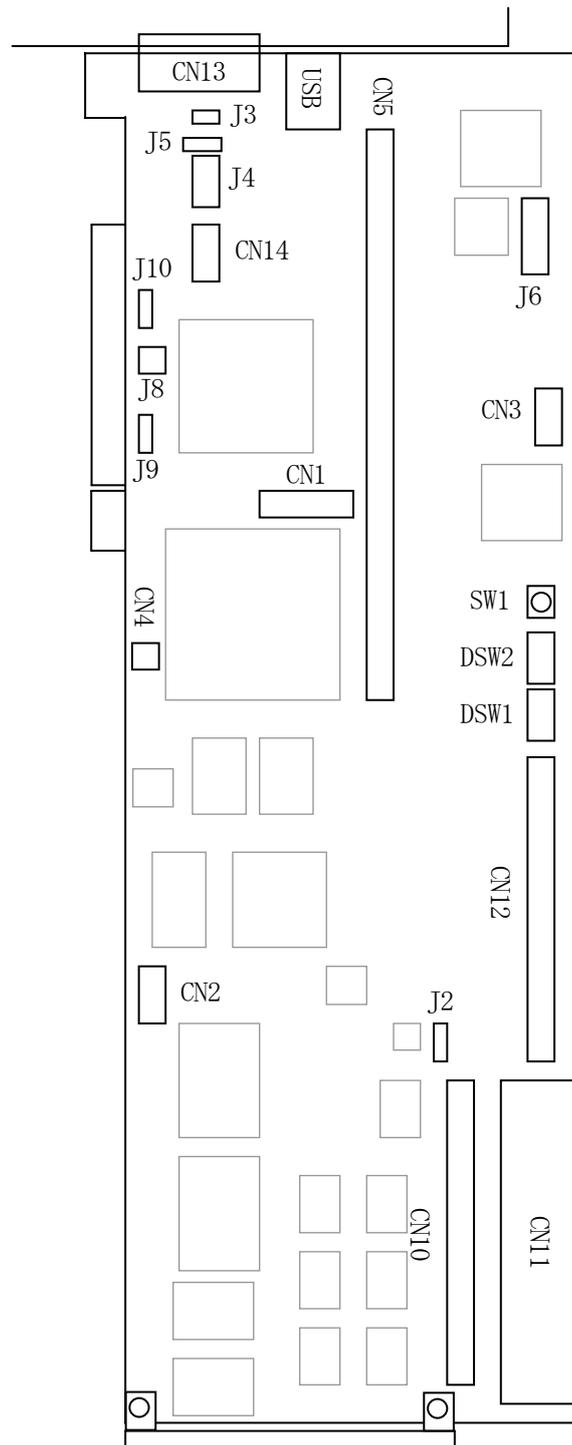


図 2-3 配置図

2.3. 入出力信号の接続例

2.3.1. パラレル入力 (TTLレベル)

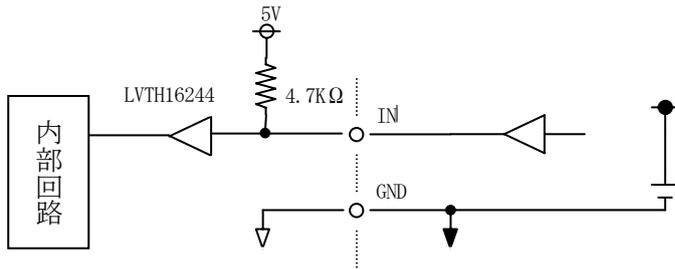


図 2-4 パラレル入力回路

2.3.2. パラレル出力 (TTLレベル)

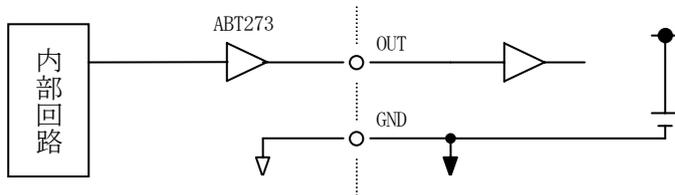


図 2-5 パラレル出力回路

2.3.3. PWM (TTLレベル出力)

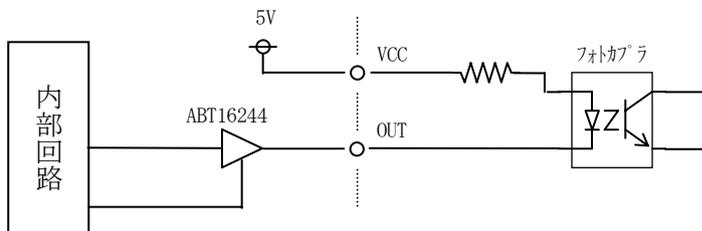


図 2-6 PWM出力回路

2.3.4. カウンタ A, B, Z相 (RS422レベル入力)

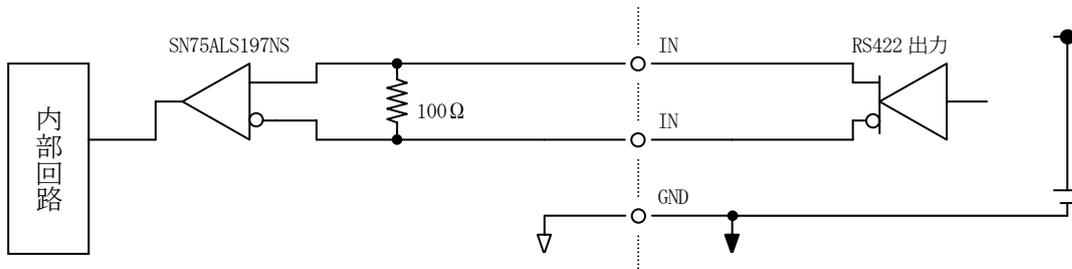


図 2-7 カウンタ入力回路

## 2.3.5. カウンタ 外部入力指定ビット (TTLレベル入力)

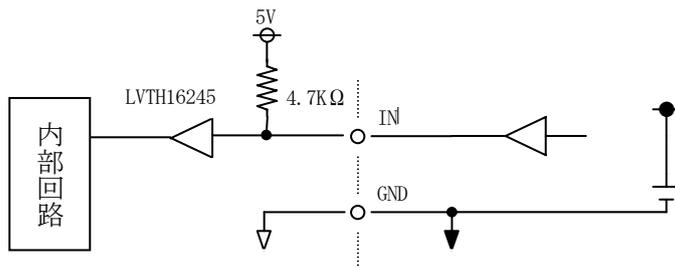


図 2-8 外部指定入力

## 2.3.6. 高速用 CAN

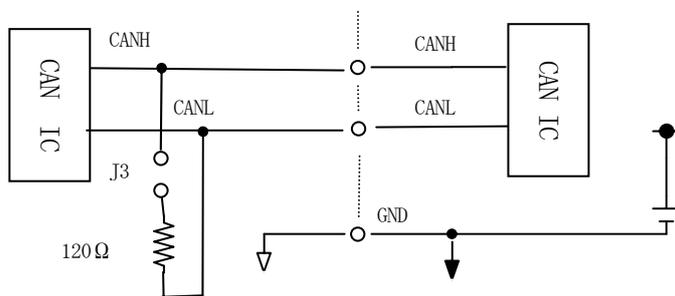


図 2-9 高速用 CAN 終端抵抗

## 2.3.7. 低速用 CAN

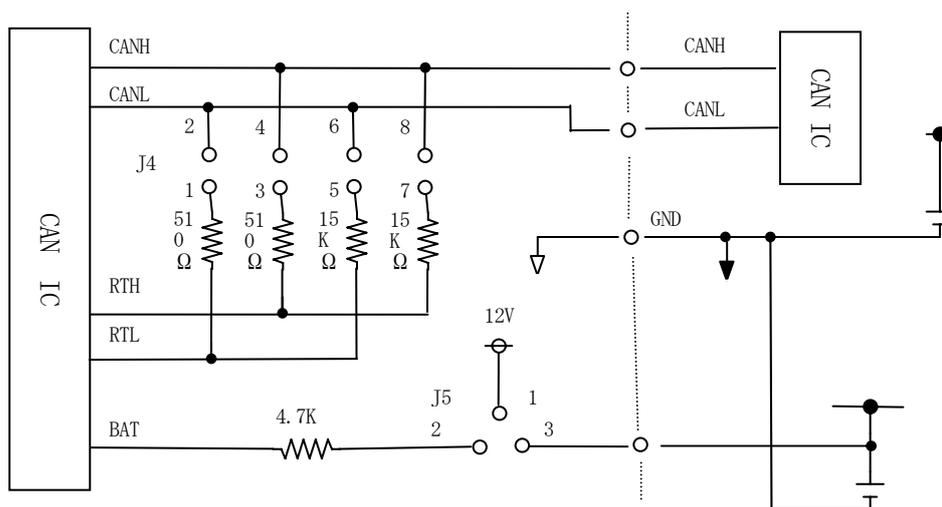


図 2-10 低速用 CAN 終端抵抗

### 3. ホストパソコンと本ボードへの接続

#### 3.1. USB接続

ホストパソコンはUSB通信を使って本ボードのメモリー空間をアクセスすることができます。USBケーブルを使って、パソコンと本ボードを接続してください。本ボードはPCIバスからは電源を供給しているだけなので、PCIバスから認識することは出来ません。

### 4. メモリマップ

#### 4.1. メモリサイズ

本ボードのメモリー空間は4Gbyteあります。主なメモリーサイズを記載します。

| 領域名             | サイズ       |
|-----------------|-----------|
| Integrated SRAM | 256Kbyte  |
| DDR SRAM        | 512Mbyte  |
| フラッシュメモリー       | 2Mbyte    |
| DPSRAM          | 64Kbyte×2 |
| BOOT・ROM        | 4Mbyte    |
| 拡張バス            | 128Mbyte  |

\*DDR SRAMはオプションで1Gbyteになります。

表 4-1 メモリサイズ

## 4.2. メモリーマップ

表にメモリマップを示します。

| メモリアドレス               | メモリ内容                           |
|-----------------------|---------------------------------|
| 0000 0000h~0003 FFFFh | Integrated SRAM(256Kbyte)       |
| 2000 0000h~3FFF FFFFh | 標準 DDR SDRAM(512Mbyte)          |
| 4000 0000h~5FFF FFFFh | オプション DDR SDRAM(512Mbyte)       |
| 6000 0000h~601F FFFFh | プライベートメモリー(2Mbyte)              |
| 8000 0000h~8023 FFFFh | 空き領域                            |
| 8240 0000h~83FF FFFFh | 拡張バス(128Mbyte)                  |
| 8400 0000h~8EFF FFFFh | 空き領域                            |
| 8F00 0000h            | 標準実装 I/O PIO IN                 |
| 8F00 0004h            | 標準実装 I/O PIO OUT                |
| 8F00 0008h            | 制御レジスタ (ポート番号、動作モード)            |
| 8F00 000Ch            | 機種コードレジスタ                       |
| 8F00 0010h            | PPC ソフトリセット                     |
| 8F00 0014h            | ウォッチドックリセット                     |
| 8F00 0018h            |                                 |
|                       |                                 |
| 8F00 0200h~8F00 027Fh | PWM                             |
| 8F00 0280h~8F00 02FFh | カウンタ                            |
| 8F00 0300h~8F00 03FFh | 予備                              |
|                       |                                 |
| 9000 0000h~9003 FFFFh | 空き領域                            |
| 9004 0000h~90FF FFFFh | 空き領域                            |
| 9100 0000h~911F FFFFh | 空き領域                            |
| 9120 0000h~91FF FFFFh | 空き領域                            |
| 9200 0000h~923F FFFFh | ユーザー用 BOOT ROM(4Mbyte)          |
| 9240 0000h~92FF FFFFh | 空き領域                            |
| 9300 0000h~9300 7FFFh | USB 用 DPSRAM(64Kbyte)           |
| 9300 8000h~9300 FFFFh | CAN 用 DPSRAM(64Kbyte)           |
| 9301 0000h~93FF FFFFh | 空き領域                            |
|                       |                                 |
|                       |                                 |
|                       |                                 |
| F100 0000h~F100 FFFFh | Internal Registers(64Kbyte)     |
| FFC0 0000h~FFFF FFFFh | 起動用 BOOT ROM(4Mbyte) ユーザー書き込み禁止 |

表 4-2 CPU (PPC) から見たメモリマップ

**【注意】** 弊社ユーティリティソフトウェアを使用する場合は  
 0000 0000h~0000 03FFh はリセット、割り込みベクタ  
 0000 0400h~0000 2FFFh はプログラム領域  
 0000 3000h~0000 34FFh はデータ領域  
 としてユーティリティソフトウェアが使用するのでユーザーは使用しないで下さい。

## 4. メモリマップ

### 4.3. 標準メモリー

本ボードには、“Integrated SRAM” “DDR SDRAM” “プライベートメモリー” “DPSRAM” の5種類のメモリーが実装されています。

### 4.4. ユーザー用 BOOT ROM

スタンドアロンで動作させる時の立ち上げ用ROM領域です。4MbyteのROMを搭載しています。BOOT・ROM用の書き込み回路が搭載されています。ROMはフラッシュメモリを使用しているため 本ボードより書き込みが可能です。ホストからは付属ソフトのユーティリティによって書き込み可能です。

### 4.5. レジスタ

現在のボードの設定状態を本レジスタを読むことによって知る事が出来ます。

#### 4.5.1. 制御レジスタ (アドレス 8F000008h)

| BIT | 8F00 0008H bit割付 |                 |
|-----|------------------|-----------------|
| 31  | 未定義              |                 |
|     | 未定義              |                 |
| 18  | 未定義              |                 |
| 17  | 未定義              |                 |
| 16  | ドーターボード無し【0】     | 有り【1】           |
| 15  | DSW2-8 OFF【0】    | ON【1】           |
| 14  | DSW2-7 OFF【0】    | ON【1】           |
| 13  | DSW2-6 OFF【0】    | ON【1】           |
| 12  | DSW2-5 OFF【0】    | ON【1】           |
| 11  | DSW2-4 OFF【0】    | ON【1】           |
| 10  | DSW2-3 OFF【0】    | ON【1】           |
| 9   | DSW2-2 OFF【0】    | ON【1】           |
| 8   | ユーザーROM書き込み禁止【0】 | 書き込み可能【1】       |
| 7   | ホスト立ち上げ【0】       | BOOT ROM立ち上げ【1】 |
| 6   | 常時【1】            | ホストとのUSB通信【1】   |
| 5   | ホストと接続する【0】      | ホストと接続しない【1】    |
| 4   | DSW1-5 OFF【0】    | ON【1】           |
| 3   | ボード番号 $2^3$      |                 |
| 2   | ボード番号 $2^2$      |                 |
| 1   | ボード番号 $2^1$      |                 |
| 0   | ボード番号 $2^0$      |                 |

表 4-3 8F000008 bit割付

ビット0～3はボード番号を表します。ボード番号は0～15まで有ります。  
 ボード番号は下記表を参照して下さい。

| ビット    | 0     | 1     | 2     | 3     |
|--------|-------|-------|-------|-------|
| ボード No | $2^0$ | $2^1$ | $2^2$ | $2^3$ |
| 0      | OFF   | OFF   | OFF   | OFF   |
| 1      | O N   | OFF   | OFF   | OFF   |
| 2      | OFF   | O N   | OFF   | OFF   |
| 3      | O N   | O N   | OFF   | OFF   |
| 4      | OFF   | OFF   | O N   | OFF   |
| 5      | O N   | OFF   | O N   | OFF   |
| 6      | OFF   | O N   | O N   | OFF   |
| 7      | O N   | O N   | O N   | OFF   |
| 8      | OFF   | OFF   | OFF   | O N   |
| 9      | O N   | OFF   | OFF   | O N   |
| 10     | OFF   | O N   | OFF   | O N   |
| 11     | O N   | O N   | OFF   | O N   |
| 12     | OFF   | OFF   | O N   | O N   |
| 13     | O N   | OFF   | O N   | O N   |
| 14     | OFF   | O N   | O N   | O N   |
| 15     | O N   | O N   | O N   | O N   |

表 4-4 ボード番号

- 2) ビット5はホストとの接続状態を表します。  
 “0”：接続可能  
 “1”：接続不可  
 (本ボードはUSB通信によってホストコンピュータと接続します。)
- 3) ビット7は起動動作モードを表します。  
 “0”：ホスト立ち上げ  
 “1”：BOOT ROM 立ち上げ (スタンドアロン起動)
- 4) ビット8はユーザーROMにプログラムを書き込む状態を表します。  
 “0”：書き込み不可  
 “1”：書き込み可能
- 5) ビット16はドータボードの有無を表します。  
 “0”：ACRO741-01 ドータボード無し  
 “1”：ACRO741-01 ドータボード有り

## 4. メモリマップ

### 4.5.2. 機種コードレジスタ (アドレス 8F0000Ch)

機種コードレジスタからACROシリーズの機種コードを読みとることが可能です。主にソースレベルデバッガ、MATLAB対応ライブラリなどの開発用のソフトウェアで使用されます。

32bitデータの下位16bitが有効であり各機種のコードを示しています。この機種コードは8F0000Chに割り付けられています。

| アドレス：8400 000Ch |                     |
|-----------------|---------------------|
| コード (4bit)      | 機種名                 |
| 00000001h       | ACRO741-00 (100MHz) |
| 00000002h       | ACRO741-00 (133MHz) |

表 4-5 機種コード

### 4.6. 割り込み

標準実装 I/O から本ボードに対し割り込みを要求

標準実装 I/O から割り込みを要求することができます。割り込みは1つの割り込み受付ラインがあり表 4-6 の様になっています。信号はCN12-88ピン「I\_D7」表 10-1 に割り付けられています。

拡張 I/O ボードから本ボードに対し割り込みを要求

拡張 I/O ボードは本ボード対し割り込みを要求することができます。割り込みは3つの割り込み受付ラインがあり表 4-6 の様になっています。信号はCN11-3, 5, 7ピン「INT3」「INT2」「INT1」表 4-10 に割り付けられています。

ドータボードから本ボードに対し割り込みを要求

ドータボードは本ボード対し割り込みを要求することができます。割り込みは3つの割り込み受付ラインがあり表 4-6 の様になっています。信号はCN10-65, 64, 63ピン「INT31」「INT21」「INT11」表 4-8 に割り付けられています

割り込みが受け付けられるとそれぞれに対応したベクタアドレスより割り込み処理開始番地を取り出しその番地にプログラムが移ります。

| 割り込み名 | 内 容                            |
|-------|--------------------------------|
| MPP13 | 標準実装 I/O からの割り込み要求 (I_D7)      |
| MPP12 | ドータボードからの割り込み要求3 (標準 INT31)    |
| MPP11 | ドータボードからの割り込み要求2 (INT21)       |
| MPP10 | ドータボードからの割り込み要求1 (INT11)       |
| MPP9  | 拡張 I/O ボードからの割り込み要求3 (標準 INT3) |
| MPP8  | 拡張 I/O ボードからの割り込み要求2 (INT2)    |
| MPP7  | 拡張 I/O ボードからの割り込み要求1 (INT1)    |
| MPP6  | CANからの割り込み要求 (使用不可)            |
| MPP5  | ホストコンピュータからの割り込み要求 (使用不可)      |

表 4-6 割り込み受付

#### 4.7. ソフトリセット

PPCにソフトリセットをかけます。

8F000010HをライトするとPPCにソフトリセットがかかります。

ソフトリセットの詳細内容は“MPC7457 ユーザーズマニュアル”(freescalar社)を参照して下さい。

#### 4.8. ウォッチドックリセット

ウォッチドック出力「\*WD\_OUT」CN12-95ピン表 10-1をリセットします。

MV6436 のレジスタ WDC (F100B410h) でウォッチドックをリセットしてから8F000014hをライトすると、ウォッチドック出力をリセットします。

#### 4.9. 拡張バス

拡張バスは、CN10用のドータボードとCN11用の拡張I/Oボードを制御するための拡張バスでPPCから直接制御が可能です。

82400000h～83FFFFFFhの空間が割り付けられ、CN10とCN11に出力されておりCN10はドータボードを1枚、CN11は拡張I/Oボードを4枚まで接続することができます。

| PPC アドレス              | I/O ボードアドレス              | I/O ボードアドレス      |
|-----------------------|--------------------------|------------------|
| 8240 0000h～8240 7FFFh | ISA 拡張ボード用 (ADSP324-**等) | 900000h～901FFFh  |
| 8240 8000h～8240 80FFh | ドータボード用 (@CR0741-01)     | 902000h～90203Fh  |
| 8240 8100h～8240 BFFFh | ドータボード (予備)              | 902040h～902FFFFh |
| 8240 C000h～8240 C7FFh | PCI 拡張ボード (@CR0741-06)   | 903000h～9031FFh  |
| 8240 C800h～8240 FFFFh | PCI 拡張ボード (予備)           | 903200h～903FFFh  |
| 8241 0000h～8243 FFEFh | 予備拡張ボード                  | 904000h～90FFFBh  |
| 8243 FFF0h            | 空き                       | 90FFFCh          |
| 8243 FFF4h            | 空き                       | 90FFFDh          |
| 8243 FFF8h            | ドータボード ID 読み出し           | 90FFFEh          |
| 8243 FFFCh            | 割り込みフラグ (共通)             | 90FFFFh          |
| 8244 0000h～83FF FFFFh | 未使用                      | 910000h～9FFFFFFh |

表 4-7 拡張バス (8240 0000h～83FF FFFFh) 割付アドレス

## 4. メモリマップ

### 4.9.1. ドータボード

ドータボードの詳しいメモリ割付は、ドータボードのハードウェア・マニュアルを参照して下さい。ここではCN10のピン割付のみ記載します。

| No | 信号名  | No | 信号名  | No | 信号名      | No  | 信号名       |
|----|------|----|------|----|----------|-----|-----------|
| 01 | XD00 | 02 | XD01 | 51 | XA18     | 52  | XA19      |
| 03 | XD02 | 04 | XD03 | 53 | XA20     | 54  | XA21      |
| 05 | XD04 | 06 | XD05 | 55 | XA22     | 56  | XA23      |
| 07 | XD06 | 08 | XD07 | 57 | *1       | 58  | *EXTWR    |
| 09 | XD08 | 10 | XD09 | 59 | *EXTSTRB | 60  | TOUT      |
| 11 | XD10 | 12 | XD11 | 61 | *1       | 62  | EXTDB1    |
| 13 | XD12 | 14 | XD13 | 63 | *INT11   | 64  | *INT21    |
| 15 | XD14 | 16 | XD15 | 65 | *INT31   | 66  | *EXTRESET |
| 17 | XD16 | 18 | XD17 | 67 | GND      | 68  | GND       |
| 19 | XD18 | 20 | XD19 | 69 | GND      | 70  | GND       |
| 21 | XD20 | 22 | XD21 | 71 | GND      | 72  | GND       |
| 23 | XD22 | 24 | XD23 | 73 | GND      | 74  | GND       |
| 25 | XD24 | 26 | XD25 | 75 | GND      | 76  | GND       |
| 27 | XD26 | 28 | XD27 | 77 | N.C      | 78  | N.C       |
| 29 | XD28 | 30 | XD29 | 79 | N.C      | 80  | N.C       |
| 31 | XD30 | 32 | XD31 | 81 | N.C      | 82  | N.C       |
| 33 | XA00 | 34 | XA01 | 83 | N.C      | 84  | N.C       |
| 35 | XA02 | 36 | XA03 | 85 | +3.3V    | 86  | +3.3V     |
| 37 | XA04 | 38 | XA05 | 87 | +3.3V    | 88  | +3.3V     |
| 39 | XA06 | 40 | XA07 | 89 | +3.3V    | 90  | +3.3V     |
| 41 | XA08 | 42 | XA09 | 91 | +3.3V    | 92  | +3.3V     |
| 43 | XA10 | 44 | XA11 | 93 | +5V      | 94  | +5V       |
| 45 | XA12 | 46 | XA13 | 95 | +5V      | 96  | +5V       |
| 47 | XA14 | 48 | XA15 | 97 | +5V      | 98  | +5V       |
| 49 | XA16 | 50 | XA17 | 99 | +5V      | 100 | +5V       |

表 4-8 CN10のピン配置

【注】 \* は負論理を表しています。  
\*1 は未使用ですが使用してはいけません。

- XD00 : バスインインターフェイスの32ビットデータバス
- XA00 : バスインインターフェイスの24ビットアドレスバス  
XA00~XA23には900000h~9FFFFFFhが出力されます。
- \*EXTSTRB : 外部のデバイスに対するストロブです。Loの時XD\*\*, XA\*\*有効
- \*EXTWR : 外部のデバイスに対するリード/ライトです
- TOUT : タイマクロック。TOUTからタイマ0で発生されたパルス出力です。
- \*INT00 : 外部割り込み。ローアクティブ。
- \*EXTRESET : PPCがリセットされた時、ローになります。

## 4.9.2. 拡張I/Oボード

拡張I/Oボードのメモリ割付（拡張I/Oボードのハードウェア・マニュアル参照）は900000h～90FFFFhに割り当てられているのでアドレス変換が必要になります。アドレス変換は本ボードが行うのでユーザーは、82400000h～8243FFFFhをアクセスして下さい。下記にアドレスの対応を示します。

CN11のアドレスには82400000h～83FFFFFFhをアクセスすると900000h～9FFFFFFhが出力されます。

拡張I/Oボードを使用するには別売のケーブル(ACR0741-15)が必要です。

| PPCボードアドレス | 拡張I/Oボードアドレス |
|------------|--------------|
| 82400000h  | 900000h      |
| 82400004h  | 900001h      |
| 82400008h  | 900002h      |
| ・          | ・            |
| ・          | ・            |
| 8243FFFC h | 90FFFF h     |
| ・          | ・            |
| ・          | ・            |
| 827FFFF8 h | 9FFFFFFE h   |
| 827FFFFC h | 9FFFFFFF h   |

表 4-9 PPCと拡張I/Oボードのアドレス対応表

4. メモリマップ

| No | 信号名       | No | 信号名 | No | 信号名  | No  | 信号名  |
|----|-----------|----|-----|----|------|-----|------|
| 01 | *EXTRESET | 02 | GND | 51 | XA08 | 52  | GND  |
| 03 | *INT3     | 04 | GND | 53 | XA07 | 54  | GND  |
| 05 | *INT2     | 06 | GND | 55 | XA06 | 56  | GND  |
| 07 | *INT1     | 08 | GND | 57 | XA05 | 58  | GND  |
| 09 | *1        | 10 | GND | 59 | XA04 | 60  | GND  |
| 11 | *1        | 12 | GND | 61 | XA03 | 62  | GND  |
| 13 | TCLK      | 14 | GND | 63 | XA02 | 64  | GND  |
| 15 | *EXTSTRB  | 16 | GND | 65 | XA01 | 66  | GND  |
| 17 | *EXTWR    | 18 | GND | 67 | XA00 | 68  | GND  |
| 19 | *1        | 20 | GND | 69 | XD31 | 70  | XD30 |
| 21 | XA23      | 22 | GND | 71 | XD29 | 72  | XD28 |
| 23 | XA22      | 24 | GND | 73 | XD27 | 74  | XD26 |
| 25 | XA21      | 26 | GND | 75 | XD25 | 76  | XD24 |
| 27 | XA20      | 28 | GND | 77 | XD23 | 78  | XD22 |
| 29 | XA19      | 30 | GND | 79 | XD21 | 80  | XD20 |
| 31 | XA18      | 32 | GND | 81 | XD19 | 82  | XD18 |
| 33 | XA17      | 34 | GND | 83 | XD17 | 84  | XD16 |
| 35 | XA16      | 36 | GND | 85 | XD15 | 86  | XD14 |
| 37 | XA15      | 38 | GND | 87 | XD13 | 88  | XD12 |
| 39 | XA14      | 40 | GND | 89 | XD11 | 90  | XD10 |
| 41 | XA13      | 42 | GND | 91 | XD09 | 92  | XD08 |
| 43 | XA12      | 44 | GND | 93 | XD07 | 94  | XD06 |
| 45 | XA11      | 46 | GND | 95 | XD05 | 96  | XD04 |
| 47 | XA10      | 48 | GND | 97 | XD03 | 98  | XD02 |
| 49 | XA09      | 50 | GND | 99 | XD01 | 100 | XD00 |

表 4-10 CN11のピン配置

【注】 \* は負論理を表しています。  
\*1 は未使用ですが使用してはいけません。

- XD00 : バスインインターフェイスの32ビットデータバス
- XA00 : バスインインターフェイスの24ビットアドレスバス  
XA00~XA23には900000h~9FFFFFFhが出力されます。
- \*EXTSTRB : 外部のデバイスに対するストローブです。Loの時XD\*\*、XA\*\*有効
- \*EXTWR : 外部のデバイスに対するリード/ライトです
- TOUT : タイマクロック。TOUTからタイマ0で発生されたパルス出力です。
- \*INT00 : 外部割り込み。ローアクティブ。
- \*EXTRESET : DSPがリセットされた時、ローになります。

## 5. 動作設定に関するスイッチの設定

## 5.1. ボード番号、モード設定 DSW1

本ボードは1台のホストコンピュータから16枚まで制御する事が出来ます。  
 そのためボード毎につきボード番号を設定します。設定はDSW1-1～4にて設定します。  
 ホストパソコンへの切り離しはDSW1-6で設定します。  
 「ON」でホストパソコンと切り離します。  
 BOOT・ROMによる立ち上げ（スタンドアロン機能）はDSW1-8にて設定します。  
 「ON」でBOOT・ROMによる立ち上げです。（スタンドアロン機能）

| DSW1 |             |         |               |
|------|-------------|---------|---------------|
| DSW1 | 信号名         | OFF【0】  | ON【1】         |
| 8    | 起動モード       | ホスト立ち上げ | BOOT ROM 立ち上げ |
| 7    | 未使用         |         |               |
| 6    | ホスト接続モード    | 接続      | 切り離し          |
| 5    | 未使用         |         |               |
| 4    | ボードNo $2^3$ | 0       | $2^3$         |
| 3    | ボードNo $2^2$ | 0       | $2^2$         |
| 2    | ボードNo $2^1$ | 0       | $2^1$         |
| 1    | ボードNo $2^0$ | 0       | $2^0$         |

表 5-1 DSW1

例 ボード番号0 :                      ホスト接続 :                      ホスト立ち上げ（出荷時）

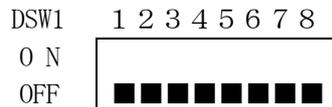


図 5-1 DSW1の設定



## 5.2. ユーザーROM書き込み設定 DSW2

ROM書き込みは、ディップスイッチDSW2-1を「ON」で書き込み許可に変更した後、付属のソフトウェアのユーティリティを使って書き込みを行って下さい。  
ユーザー用ROM 4Mbyteのフラッシュメモリが実装されています。

| DSW2 |             |        |       |
|------|-------------|--------|-------|
| DSW2 | 信号名         | OFF【0】 | ON【1】 |
| 8    | 未使用         |        |       |
| 7    | 未使用         |        |       |
| 6    | 未使用         |        |       |
| 5    | 未使用         |        |       |
| 4    | 未使用         |        |       |
| 3    | 未使用         |        |       |
| 2    | 未使用         |        |       |
| 1    | ユーザーROM書き込み | 禁止     | 許可    |

表 5-3 DSW2

例 ユーザーROM書き込み禁止（出荷時）

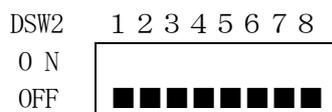


図 5-2 DSW2の設定

## 5.3. ハードリセット SW1

ボードなどが異常動作（暴走など）となり、本ボードの制御が不能になったときなど、外部よりリセット信号を入力することで本ボードを初期化状態にすることができます。本ボードへリセット入力を行うときは、SW1を押してください。

## 6. 標準実装 I/O

本ボードには、8 bit のパラレル入力が 1 CH、8 bit のパラレル出力が 1 CH、32 bit アップダウンカウンタが 4 CH、3 相×3 CH か単相×4 CH の PWM が搭載されています。

## 6.1. 精度と機能

## □パラレル入力部

- ①点数 8 bit × 1 ch
- ②信号レベル TTLレベル (終端 4.7kΩでプルアップ)
- ③割り込み信号 I<sub>LD</sub>7はPPCへの割り込みも可能  
(割り込みレジスタ有効時)

## □パラレル出力部

- ①点数 8 bit × 1 ch
- ②信号レベル TTLレベル

## □アップダウンカウンタ入力部

- ①点数 32 bit 4 ch
- ②信号レベル RS422レベル・90度位相差パルス (A, B, Z相) 入力
- ③最高入力周波数 5MHz
- ④通倍 各CH毎に1, 2, 4通倍選択
- ⑤カウンタリセット Z相入力とソフトウェアによるプリセット

## □PWM出力部

- ①点数 3相×3CH 又は単相×4CH(ソフトウェア選択)
- ②信号レベル TTLレベル
- ③キャリア周波数 24Hz～120kHz (分解能<sup>1</sup>/24MHz)
- ④デューティ比 0～100% (分解能 キャリア周波数に依存)
- ⑤デッドタイム 0～2msec (分解能<sup>1</sup>/24MHz)

## 6.2. PIO 入力と出力

### 6.2.1. PIO 8bit入力

8bit入力データ（ビット0～7）は【8F000000h】にマップされています。データはこのアドレスからのデータ読み込みによって行われます。入力インターフェイスはTTLレベル(LVTH16244)です。(図 2-4 参照) CN12に割り付けられています。(表 10-1参照)

| ビット | 31 | ～ | 8 | 7    | 6    | ～ | 1    | 0    |
|-----|----|---|---|------|------|---|------|------|
| 信号名 | *  | ～ | * | I_D7 | I_D6 | ～ | I_D1 | I_D0 |

\* は不定（ビット8～31のデータは不定です）

### 6.2.2. PIO 8bit出力

8bit出力データ（ビット0～7）は【8F000004h】にマップされています。データはこのアドレスへのデータ書き込みによって行われます。このポートを読み出すと、現在の出力状態がわかります。出力インターフェイスはTTLレベル(SN74ABT273)です。(図 2-5 参照) CN12に割り付けられています。(表 10-1参照)

| ビット | 31 | ～ | 8 | 7    | 6    | ～ | 1    | 0    |
|-----|----|---|---|------|------|---|------|------|
| 信号名 | *  |   | * | O_D7 | O_D6 |   | O_D1 | O_D0 |

\* の出力データは無効です。読み出しデータは“0”です。

### 6.2.3. PIOのメモリマップ

| アドレス       | WRITE     | READ        |
|------------|-----------|-------------|
| 8F00 0000h |           | パラレル入力データ   |
| 8F00 0004h | パラレル出力データ | 左記データ値の読み出し |

表 6-1 PIOメモリーマップ

## 6.3. PWM

## 6.3.1. 3相モードと単相モード

本ボードのPWM出力回路は、3相モードと単相モードがあります。

## 1) 3相モードの出力

下図のような回路構成で3相PWM出力を行うことができます。  
出力インターフェイスはTTLレベル (SN74ABT16244) です。(図 2-6 参照)

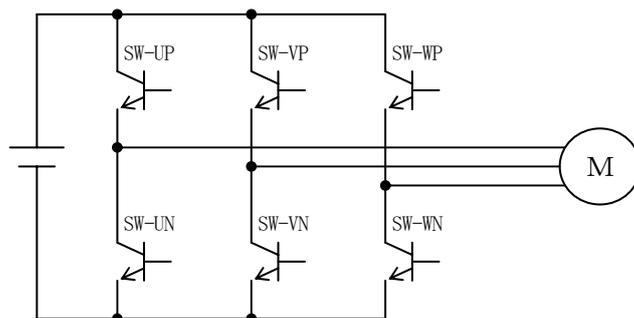


図 6-1 3相PWM出力の回路構成

## 2) 単相モードの出力

下図のような回路構成で単相PWM (Hブリッジ回路) 出力を行うことができます。  
出力インターフェイスはTTLレベル (SN74ABT16244) です。(図 2-6 参照)

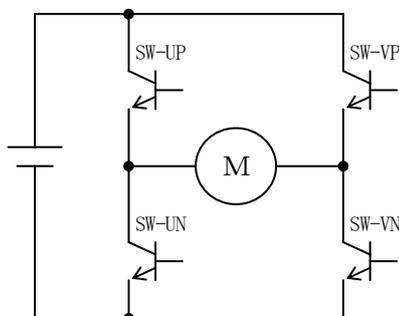
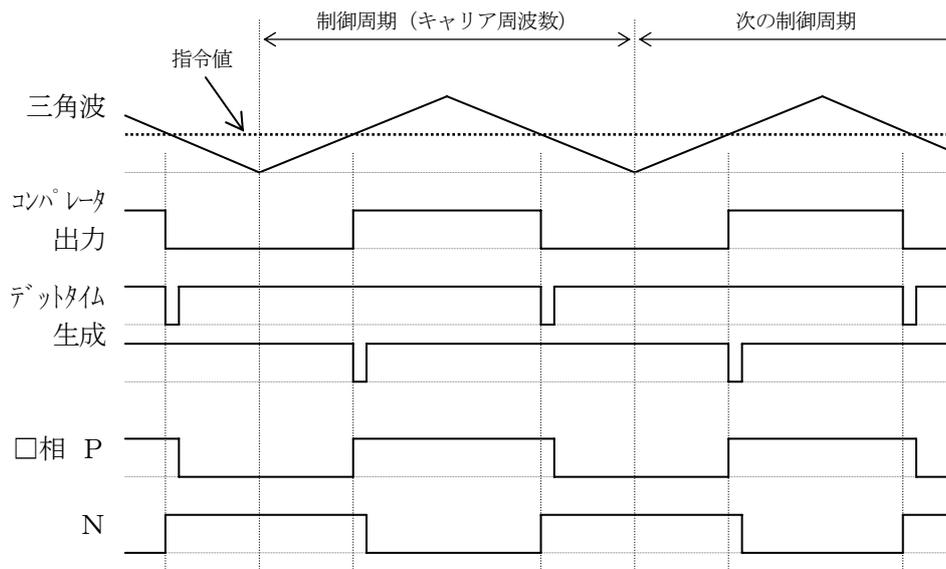


図 6-2 単相PWM出力の回路構成

### 6.3.2. 出力波形

本ボードのPWM出力波形は  
 フルブリッジの正出力、フルブリッジの逆出力  
 ハーフブリッジの正出力、ハーフブリッジの逆出力  
 の4通りの出力が可能です。それぞれの出力波形は以下のとおりです。

#### 1) フルブリッジの正出力



※各相の出力は負論理。

図 6-3 PWM出力波形 フルブリッジの正出力

2) フルブリッジの逆出力

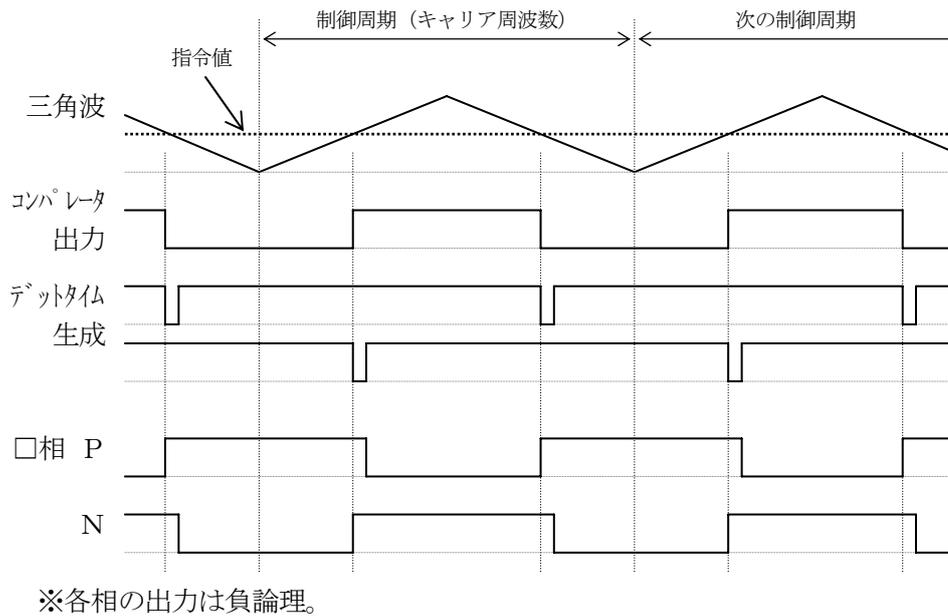


図 6-4 PWM出力波形 フルブリッジの逆出力

3) ハーフブリッジの正出力

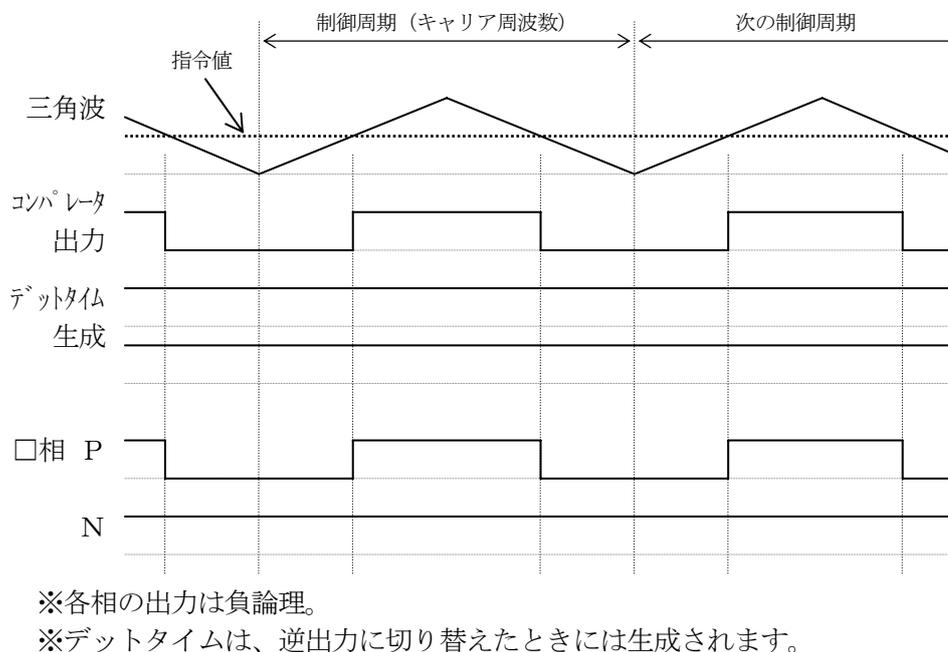
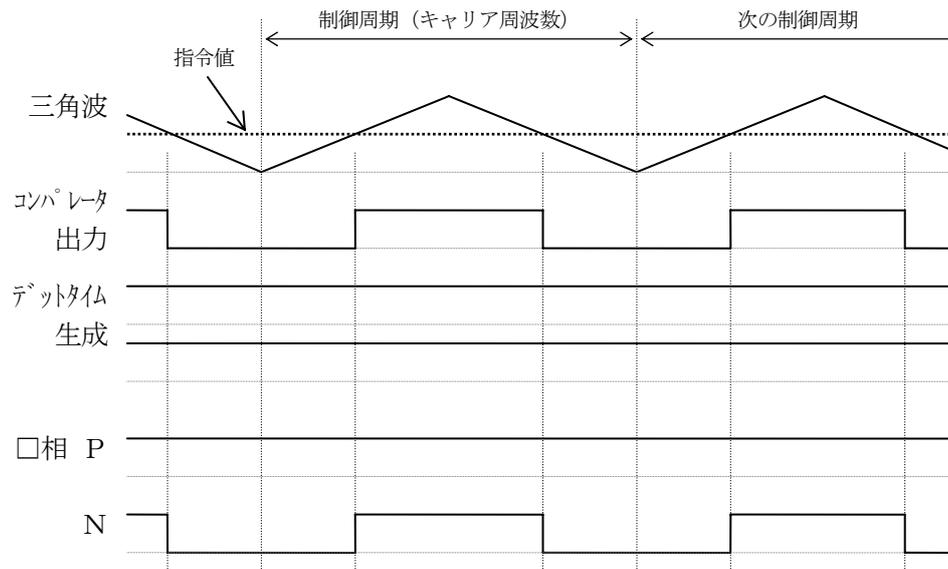


図 6-5 PWM出力波形 ハーフブリッジの正出力

## 4) ハーフブリッジの逆出力



※各相の出力は負論理。

※デットタイムは、正出力に切り替えたときには生成されます。

図 6-6 PWM出力波形 ハーフブリッジの逆出力

## 6.3.3. 同期パルス出力

本ボードのPWM波形生成回路では、制御周期（キャリア周波数）に同期したパルス出力が可能です。このパルス信号は、A/D変換開始のトリガに使用するなどPWM制御周期に同期した制御に利用できます。この同期パルス出力はPWM制御周期毎（1周期に1出力）から数周期毎（2～255周期に1出力）の出力設定が可能です。（キャリア周波数設定と同期パルス出力設定を行えば、出力許可と関係なく同期パルスが出力されます）

パルス出力幅はPWM制御周期の1/2幅で出力され、負論理・正論理の出力切り替えが可能です。

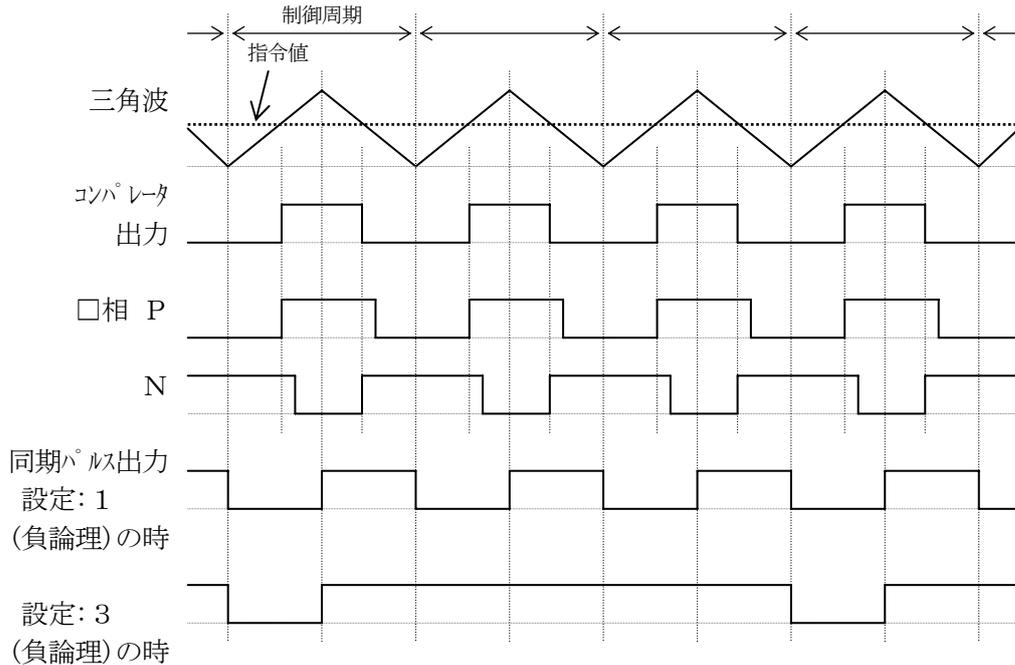


図 6-7 同期パルス出力波形

## 6.3.4. 設定データの反映タイミング

本ボードのPWM波形生成回路では、設定されたキャリア周波数データとDutyデータを反映するタイミングを周期に同期させるか、書き込み時に反映するかを選択できます。

## 1) 周期に同期して反映

「図 6-3 PWM出力波形 フルブリッジの正出力」の制御周期の途中でデータが書き込まれても波形生成回路には反映されず（出力波形には影響はなく）、次の制御周期へ切り替わったときにデータが反映されますので安定した出力が得られます。但し、最大1周期分の制御遅れが発生します。

## 2) 書き込み時に反映

データの書き込みがおこなわれた時点ですぐ波形生成回路に反映されます。データの書き込みが行われたときの制御周期での出力は意図しない波形となる場合がありますが、早く制御を切り替えることができます。

### 6.3.5. 初期設定

この機能を使用するときの初期設定する必要のあるレジスタには

1. モード選択、出力許可
2. 設定データ反映タイミング
3. デットタイム

などがあり、機能内容の選択のために初期設定を行います。

なお、初期状態では、モード【3相モード】、出力許可【未許可】、設定データ反映タイミング【周期に同期】、デットタイム【0】となっています。

#### 1) モード選択、出力許可・レジスタの設定

モード選択、出力許可・レジスタは、PWM出力モード【3相モード】【単相モード】の選択とチャンネル毎の【出力許可】【出力未許可】の選択を行うことができます。電源投入時、このレジスタはイニシャルリセットされ【bit 0】～【bit 4】はすべて【0】になっています。

| モード選択、出力許可<br>レジスタ | チャンネル  | 0     | 1     |
|--------------------|--------|-------|-------|
| bit 4              | チャンネル3 | 出力未許可 | 出力許可  |
| bit 3              | チャンネル2 | 出力未許可 | 出力許可  |
| bit 2              | チャンネル1 | 出力未許可 | 出力許可  |
| bit 1              | チャンネル0 | 出力未許可 | 出力許可  |
| bit 0              | 共通     | 3相モード | 単相モード |

表 6-2 モード選択、出力許可・レジスタ

#### 2) 設定データ反映タイミング・レジスタの設定

設定データ反映タイミング・レジスタは、データ一括書き込み指令・レジスタへの書き込みによって制御プロセスへ反映された、キャリア周波数と各Dutyデータを波形生成回路に反映するタイミング【周期に同期】【書き込み時】の選択を行うことができます。電源投入時、このレジスタはイニシャルリセットされ【bit 1】～【bit 4】はすべて【0】になっています。

| 設定データ反映タイミング<br>レジスタ | チャンネル  | 0     | 1     |
|----------------------|--------|-------|-------|
| bit 4                | チャンネル3 | 周期に同期 | 書き込み時 |
| bit 3                | チャンネル2 | 周期に同期 | 書き込み時 |
| bit 2                | チャンネル1 | 周期に同期 | 書き込み時 |
| bit 1                | チャンネル0 | 周期に同期 | 書き込み時 |
| bit 0                | —      | —     | —     |

表 6-3 設定データ反映タイミング・レジスタ

### 3) デットタイム・レジスタの設定

デットタイム・レジスタは、チャンネル毎のデットタイムの設定を行うことができます。電源投入時、このレジスタはイニシャルリセットされ【0】になっています。

【注意】デットタイム・レジスタへの設定を【1】以上にして動作させてください。

### 4) 同期パルス出力・レジスタの設定

同期パルス出力・レジスタはチャンネル毎の同期パルス出力の設定を行うことができます。電源投入時、このレジスタはイニシャルリセットされ【0】になっています。

## 6.3.6. PWMのメモリマップと制御レジスタ

この機能を使用するときのソフトウェアによる制御は、

1. キャリア周波数の設定
2. Dutyの設定
3. データ一括書き込み指令
4. デットタイムの設定
5. モード選択、出力許可の設定
6. 設定データ反映タイミングの設定

等があります。

## 1) メモリマップ

| アドレス       | WRITE                  | READ |
|------------|------------------------|------|
| 8F00 0200h | PWM CH0<br>キャリア周波数     |      |
| 8F00 0204h | PWM CH0<br>U相 Duty     |      |
| 8F00 0208h | PWM CH0<br>V相 Duty     |      |
| 8F00 020Ch | PWM CH0<br>W相 Duty     |      |
| 8F00 0210h | PWM CH0<br>データ一括書き込み指令 |      |
| 8F00 0214h | PWM CH1<br>キャリア周波数     |      |
| 8F00 0218h | PWM CH1<br>U相 Duty     |      |
| 8F00 021Ch | PWM CH1<br>V相 Duty     |      |
| 8F00 0220h | PWM CH1<br>W相 Duty     |      |
| 8F00 0224h | PWM CH1<br>データ一括書き込み指令 |      |
| 8F00 0228h | PWM CH2<br>キャリア周波数     |      |
| 8F00 022Ch | PWM CH2<br>U相 Duty     |      |
| 8F00 0230h | PWM CH2<br>V相 Duty     |      |
| 8F00 0234h | PWM CH2<br>W相 Duty     |      |
| 8F00 0238h | PWM CH2<br>データ一括書き込み指令 |      |
| 8F00 023Ch |                        |      |

表 6-4 PWMのメモリマップ (1)

| アドレス        | WRITE                  | READ       |
|-------------|------------------------|------------|
| 8F00 0240 h | PWM CH3<br>キャリア周波数     |            |
| 8F00 0244 h | PWM CH3<br>U相 Duty     |            |
| 8F00 0248 h | PWM CH3<br>V相 Duty     |            |
| 8F00 024C h | PWM CH3<br>データ一括書き込み指令 |            |
| 8F00 0250 h | PWM CH0<br>デットタイム      |            |
| 8F00 0254 h | PWM CH1<br>デットタイム      |            |
| 8F00 0258 h | PWM CH2<br>デットタイム      |            |
| 8F00 025C h | PWM CH3<br>デットタイム      |            |
| 8F00 0260 h | PWM CH0<br>同期パルス出力     |            |
| 8F00 0264 h | PWM CH1<br>同期パルス出力     |            |
| 8F00 0268 h | PWM CH2<br>同期パルス出力     |            |
| 8F00 026C h | PWM CH3<br>同期パルス出力     |            |
| 8F00 0270 h | PWM<br>モード選択、出力許可      | 左記設定値の読み出し |
| 8F00 0274 h | PWM<br>設定データ反映タイミング    | 左記設定値の読み出し |
| 8F00 0278 h |                        |            |
| 8F00 027C h |                        |            |

表 6-5 PWMのメモリマップ (2)

## 2) キャリア周波数・レジスタの設定

キャリア周波数・レジスタはチャンネル(0~3)毎に、【8F000200h、8F000214h、8F000228h、8F000240h】にマップされています。このレジスタは、  
 $(1 \div \text{設定したいキャリア周波数 (Hz)}) \div 41.66 \text{ (nSec)} \text{ (}^{1/24\text{MHz}}\text{)}$   
 で算出したパルス数を設定してください。

設定範囲は、200 ~ 1,000,000 (有効ビット: 下位20ビット) です。

(120k ~ 24Hz)

このレジスタへの設定は、設定データ反映タイミング・レジスタの設定によって波形生成回路への反映タイミングが異なり、出力波形に多少の影響がありますので注意してください。

## 3) U相Duty・レジスタの設定

U相Duty・レジスタはチャンネル(0~3)毎に、【8F000204h、8F000218h、8F00022Ch、8F000244h】にマップされています。このレジスタは、

キャリア周波数設定値 × 設定したいDuty比 (%) / 100 で算出したパルス数を設定してください。

設定範囲は、1,000,000 ~ 2,0 (有効ビット: 下位20ビット) です。

また、最上位ビットが正出力/逆出力選択フラグビットになっています。最上位ビットが【0】のとき、正出力となります。最上位ビットが【1】のとき、逆出力となります。

正出力、逆出力については、「6.3.2 出力波形」を参照してください。

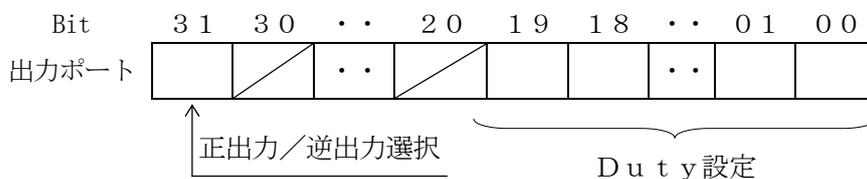


図 6-8 Duty・レジスタ

このレジスタへの設定は、設定データ反映タイミング・レジスタの設定によって波形生成回路への反映タイミングが異なり、出力波形に多少の影響がありますので注意してください。

モード選択が【3相モード】のとき、チャンネル3の設定は無効です。

## 4) V相Duty・レジスタの設定

V相Duty・レジスタはチャンネル(0~3)毎に、【8F000208h、8F00021Ch、8F000230h、8F000248h】にマップされています。

設定値算出、設定範囲等は「0 3) U相Duty・レジスタの設定」と同じです。

モード選択が【3相モード】のとき、チャンネル3の設定は無効です。

### 5) W相Duty・レジスタの設定

W相Duty・レジスタはチャンネル（0～2）毎に、【8F00020Ch、8F000220h、8F000234h】にマップされています。設定値算出、設定範囲等は「0 3) U相Duty・レジスタの設定」と同じです。モード選択が【単相モード】のとき、このレジスタの設定は無効になります。

### 6) データ一括書き込み指令・レジスタ

データ一括書き込み指令・レジスタはチャンネル（0～3）毎に、【8F000210h、8F000224h、8F000238h、8F00024Ch】にマップされています。このレジスタに書き込みを行うことによりキャリア周波数・レジスタとDuty・レジスタのデータを制御プロセスに同時に反映することができます。キャリア周波数、各相Duty設定データの書き込み途中で制御プロセスへデータが反映されてしまうと、意図しない出力波形になる可能性があります。それを回避するために、PPCからのデータ書き込みで制御プロセスの前段にあるメモリへデータを記憶し、データ一括書き込み指令・レジスタへの書き込み指令で全データを一括して制御プロセスへ反映します。その後、設定データ反映タイミング・レジスタへの設定により、各データは波形生成回路へ反映されます。

### 7) デットタイム・レジスタの設定

デットタイム・レジスタはチャンネル（0～3）毎に、【8F000250h～8F00025Ch】にマップされています。このレジスタは、
$$\text{設定したい時間 (nSec)} \div 41.66 \text{ (nSec)} \left( \frac{1}{2.4\text{MHz}} \right)$$
 で算出したパルス数を設定してください。設定範囲は、0 ～ 48,000（有効ビット：下位16ビット）です。  
(0 ～ 2mSec)

【注意】デットタイム・レジスタへの設定を【1】以上にして動作させてください。

## 8) 同期パルス出力・レジスタの設定

同期パルス出力・レジスタはチャンネル（0～3）毎に、【8F000260h～8F00026Ch】にマップされています。このレジスタは、同期パルスを出力する場合、PWM制御周期の何回に一回出力するかを設定してください。同期パルスを出力しない場合、【0】を設定してください。

（キャリア周波数設定と同期パルス出力設定を行えば、出力許可と関係なく同期パルスが出力されます。）

設定範囲は、0，1～255（有効ビット：下位8ビット）です。

また、最上位ビットが負論理出力／正論理出力選択フラグビットになっています。最上位ビットが【0】のとき、負論理出力となります。最上位ビットが【1】のとき、正論理出力となります。

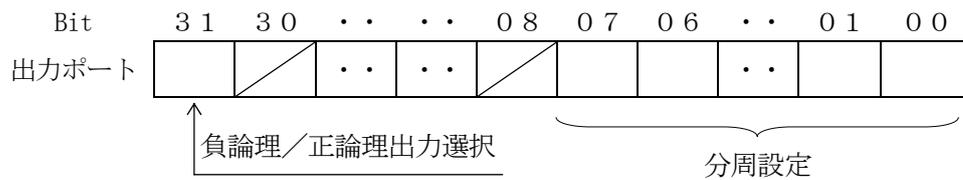


図 6-9 同期パルス出力・レジスタ

## 9) モード選択、出力許可・レジスタの設定

モード選択、出力許可・レジスタは、【8F000270h】にマップされています。このレジスタは、PWM出力モードの選択と出力チャンネルの選択を行うことができます。モード選択、出力許可・レジスタは【bit0】～【bit8】までの9ビットで構成され、【bit9】～【bit31】までは使用しません。

モード選択、出力許可・レジスタ【bit0】が【0】のとき、【3相モード】となりチャンネル0～2が有効になります。モード選択、出力許可・レジスタ【bit0】が【1】のとき、【単相モード】となりチャンネル0～3が有効になります。

モード選択、出力許可・レジスタ【bit1】～【bit4】は各チャンネルの出力許可設定です。各チャンネルの設定が【0】のとき、【出力未許可】となります。各チャンネルの設定が【1】のとき、【出力許可】となります。各チャンネルの出力を未許可にすると、制御プロセス内のカウンタなどはリセットされ、出力がOFF（【Hi】レベル）になります。チャンネル3の設定は、モード選択が【単相モード】のときに有効です。

モード選択、出力許可・レジスタ【bit5】～【bit8】は各チャンネルの出力モード設定です。各チャンネルの設定が【0】のとき、【フルブリッジ出力】となります。各チャンネルの設定が【1】のとき、【ハーフブリッジ出力】となります。

フルブリッジ出力、ハーフブリッジ出力については、「6.3.2 出力波形」を参照してください。

| モード選択、出力許可<br>レジスタ | チャンネル  | 0        | 1         |
|--------------------|--------|----------|-----------|
| bit8               | チャンネル3 | フルブリッジ出力 | ハーフブリッジ出力 |
| bit7               | チャンネル2 | フルブリッジ出力 | ハーフブリッジ出力 |
| bit6               | チャンネル1 | フルブリッジ出力 | ハーフブリッジ出力 |
| bit5               | チャンネル0 | フルブリッジ出力 | ハーフブリッジ出力 |
| bit4               | チャンネル3 | 出力未許可    | 出力許可      |
| bit3               | チャンネル2 | 出力未許可    | 出力許可      |
| bit2               | チャンネル1 | 出力未許可    | 出力許可      |
| bit1               | チャンネル0 | 出力未許可    | 出力許可      |
| bit0               | 共通     | 3相モード    | 単相モード     |

表 6-6 モード選択、出力許可・レジスタ

## 10) 設定データ反映タイミング・レジスタの設定

設定データ反映タイミング・レジスタは、【8F000274h】にマップされています。このレジスタは、データ一括書き込み指令・レジスタへの書き込みによって制御プロセスへ反映された、キャリア周波数と各Dutyデータを波形生成回路に反映するタイミング周期に同期／書き込み時の選択を行うことができます。設定データ反映タイミング・レジスタは【bit1】～【bit4】までの4ビットで構成され、【bit0】と【bit5】～【bit31】までは使用しません。

周期に同期は、設定された制御周期の切り替わりに同期して設定データを反映します。設定されたデータは、次の制御のはじめに反映されますので安定した出力が得られますが、最大1周期（キャリア周波数・レジスタに設定された周期）の制御遅れが発生します。

書き込み時は、書き込みがおこなわれた時点で設定データを反映します。

書き込みが行われたときの制御周期での出力は意図しない波形となる場合がありますが、早く制御を切り替えることができます。

| 設定データ反映タイミング<br>レジスタ | チャンネル  | 0     | 1     |
|----------------------|--------|-------|-------|
| bit4                 | チャンネル3 | 周期に同期 | 書き込み時 |
| bit3                 | チャンネル2 | 周期に同期 | 書き込み時 |
| bit2                 | チャンネル1 | 周期に同期 | 書き込み時 |
| bit1                 | チャンネル0 | 周期に同期 | 書き込み時 |
| bit0                 | —      | —     | —     |

表 6-7 設定データ反映タイミング・レジスタ

6.4. カウンタ

6.4.1. カウンタのアップ・ダウン

本ボードのカウンタの値は、A相がB相に対して進み位相のときにアップカウントします。

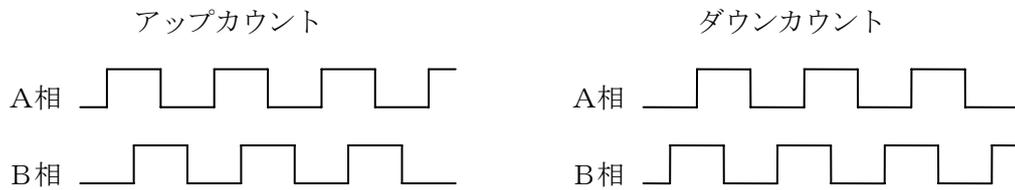


図 6-10 AB相入力とカウントの関係

6.4.2. カウンタの通倍

本ボードのカウンタの通倍は、下記のようにになっています。

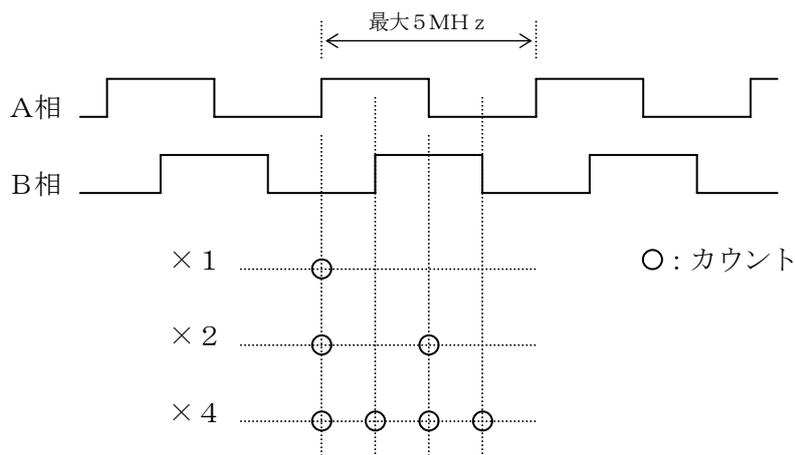


図 6-11 カウンタの通倍

## 6.4.3. カウンタリセット

本ボードのカウンタ回路は外部よりカウンタ毎にリセットがかかることができるようになって  
います。リセットさせるには、カウンタリセット・レジスタに【1】を設定し、外部入力指定ビ  
ットが【1】（外部入力指定端子をGNDレベルにする）で、Z相が【Hi】、B相が【Lo】の  
ときのA相の立ち上がり又は立ち下がりでゼロリセットされます。

外部入力指定ビットはTTLレベル（SN74LVTH16245）です。（図 2-7 2-8 参照）

| チャンネル  | カウンタリセット<br>許可 レジスタ | 外部入力<br>指定ビット    | Z相   | B相   | A相               |
|--------|---------------------|------------------|------|------|------------------|
| チャンネル3 | bit 4<br>【1】        | CN12-25ピン<br>【1】 | 【Hi】 | 【Lo】 | 立ち上がり又は<br>立ち下がり |
| チャンネル2 | bit 3<br>【1】        | CN12-26ピン<br>【1】 | 【Hi】 | 【Lo】 | 立ち上がり又は<br>立ち下がり |
| チャンネル1 | bit 2<br>【1】        | CN12-27ピン<br>【1】 | 【Hi】 | 【Lo】 | 立ち上がり又は<br>立ち下がり |
| チャンネル0 | bit 1<br>【1】        | CN12-28ピン<br>【1】 | 【Hi】 | 【Lo】 | 立ち上がり又は<br>立ち下がり |

表 6-8 カウンタリセット条件

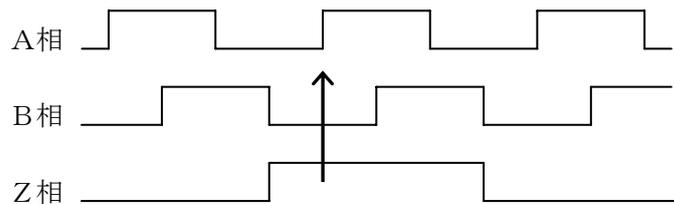


図 6-12 ABZ相のカウンタリセット条件

## 6.4.4. 初期設定

この機能を使用するときの初期設定する必要があるレジスタには

1. 通倍選択
2. リセット許可

などがあり、機能内容の選択のために初期設定を行います。

なお、初期状態では、通倍選択【×4】、リセット許可【未許可】となっています。

## 1) カウンタ通倍選択・レジスタの設定

カウンタ通倍選択・レジスタは、チャンネル毎のカウント通倍【×1】、【×2】、【×4】の選択を行うことができます。電源投入時、このレジスタはイニシャルリセットされ【bit0】～【bit15】はすべて【0】になっています。

カウンタの通倍の詳細につきましては、「6.4.2 カウンタの通倍」を参照してください。

| モード選択レジスタ | チャンネル  | ×4 | ×1 | ×2 |
|-----------|--------|----|----|----|
| bit15     | —      | —  | —  |    |
| bit14     | —      | —  | —  |    |
| bit13     | チャンネル3 | 0  | 0  | 1  |
| bit12     |        | 0  | 1  | 0  |
| bit11     | —      | —  | —  |    |
| bit10     | —      | —  | —  |    |
| bit9      | チャンネル2 | 0  | 0  | 1  |
| bit8      |        | 0  | 1  | 0  |
| bit7      | —      | —  | —  |    |
| bit6      | —      | —  | —  |    |
| bit5      | チャンネル1 | 0  | 0  | 1  |
| bit4      |        | 0  | 1  | 0  |
| bit3      | —      | —  | —  |    |
| bit2      | —      | —  | —  |    |
| bit1      | チャンネル0 | 0  | 0  | 1  |
| bit0      |        | 0  | 1  | 0  |

表 6-9 カウンタ通倍選択・レジスタ

## 2) カウンタ リセット許可・レジスタの設定

カウンタ リセット許可・レジスタは、チャンネル毎のカウンタリセットの【許可】【未許可】の選択を行うことができます。電源投入時、このレジスタはイニシャルリセットされ【b i t 1】～【b i t 4】はすべて【0】になっています。

| カウンタリセット許可 レジスタ | チャンネル  | 0   | 1  |
|-----------------|--------|-----|----|
| b i t 4         | チャンネル3 | 未許可 | 許可 |
| b i t 3         | チャンネル2 | 未許可 | 許可 |
| b i t 2         | チャンネル1 | 未許可 | 許可 |
| b i t 1         | チャンネル0 | 未許可 | 許可 |
| b i t 0         | —      | —   | —  |

表 6-10 カウンタリセット許可・レジスタ

## 6.4.5. カウンタのメモリマップと制御レジスタ

この機能を使用するときのソフトウェアによる制御は、

1. カウンタ値の読み出し
2. カウンタ値のプリセット
3. カウンタリセット許可の設定
4. カウンタ逡倍の設定

等があります。

## 1) カウンタのメモリマップ

| アドレス        | WRITE                 | READ                  |
|-------------|-----------------------|-----------------------|
| 8F00 0280 h |                       | カウンタ CH0<br>カウンタ値読み出し |
| 8F00 0284 h |                       | カウンタ CH1<br>カウンタ値読み出し |
| 8F00 0288 h |                       | カウンタ CH2<br>カウンタ値読み出し |
| 8F00 028C h |                       | カウンタ CH3<br>カウンタ値読み出し |
| 8F00 0290 h |                       |                       |
| 8F00 0294 h |                       |                       |
| 8F00 0298 h |                       |                       |
| 8F00 029C h |                       |                       |
| 8F00 02A0 h | カウンタ CH0<br>カウンタプリセット |                       |
| 8F00 02A4 h | カウンタ CH1<br>カウンタプリセット |                       |
| 8F00 02A8 h | カウンタ CH2<br>カウンタプリセット |                       |
| 8F00 02AC h | カウンタ CH3<br>カウンタプリセット |                       |
| 8F00 02B0 h | カウンタ<br>カウンタリセット許可    | (左記設定値の読み出し)          |
| 8F00 02B4 h | カウンタ<br>カウンタ逡倍選択      | (左記設定値の読み出し)          |
| 8F00 02B8 h |                       |                       |
| 8F00 02BC h |                       | カウンタ<br>外部入力指定ビットの状態  |

表 6-11 カウンタのメモリマップ

## 2) カウンタデータの読み出し

32 bit カウンタデータの読み出しはチャンネル (0~3) 毎に【8F000280h~8F00028Ch】にマップされています。このレジスタは、外部からの90°位相の異なる2つの入力パルスによってアップ・ダウンカウントされたデータを読み出すことができます。

## 3) カウント プリセット

32 bit カウンタデータのプリセットはチャンネル (0~3) 毎に【8F0002A0h~8F0002ACh】にマップされています。このレジスタにデータを書き込むことにより任意のデータにすることができます。

## 4) カウンタ リセット許可・レジスタの設定

カウンタリセット・レジスタは、【8F0002B0h】にマップされています。このレジスタは、チャンネル毎のカウンタリセットの 許可/未許可の選択を行うことができます。カウンタリセット・レジスタは【bit1】~【bit4】までの4ビットで構成され、【bit0】と【bit5】~【bit31】までは使用しません。

実際にリセットさせるには、カウンタリセット・レジスタに【1】を設定し、外部入力指定ビットが【1】(入力端子をGNDレベルにする)で、Z相が【Hi】、B相が【Lo】のときのA相の立ち上がり又は立ち下がりでゼロリセットされます。

| カウンタリセット許可 レジスタ | チャンネル  | 0   | 1  |
|-----------------|--------|-----|----|
| bit4            | チャンネル3 | 未許可 | 許可 |
| bit3            | チャンネル2 | 未許可 | 許可 |
| bit2            | チャンネル1 | 未許可 | 許可 |
| bit1            | チャンネル0 | 未許可 | 許可 |
| bit0            | —      | —   | —  |

表 6-12 カウンタリセット許可・レジスタ

| チャンネル  | カウンタリセット許可 レジスタ | 外部入力指定ビット        | Z相   | B相   | A相           |
|--------|-----------------|------------------|------|------|--------------|
| チャンネル3 | bit4<br>【1】     | CN12-25ピン<br>【1】 | 【Hi】 | 【Lo】 | 立ち上がり又は立ち下がり |
| チャンネル2 | bit3<br>【1】     | CN12-26ピン<br>【1】 | 【Hi】 | 【Lo】 | 立ち上がり又は立ち下がり |
| チャンネル1 | bit2<br>【1】     | CN12-27ピン<br>【1】 | 【Hi】 | 【Lo】 | 立ち上がり又は立ち下がり |
| チャンネル0 | bit1<br>【1】     | CN12-28ピン<br>【1】 | 【Hi】 | 【Lo】 | 立ち上がり又は立ち下がり |

表 6-13 カウンタリセット条件

## 5) カウンタ逡倍選択・レジスタの設定

カウンタ逡倍選択・レジスタは、【8F0002B4h】にマップされています。チャンネル毎のカウンタ逡倍の選択を行うことができます。カウンタ逡倍選択・レジスタは【bit0】～【bit15】までの16ビットで構成され、【bit16】～【bit31】までは使用しません。選択できる範囲は【×1】、【×2】、【×4】（デフォルトは【×4】）の3種類から選択することができます。

カウンタの逡倍の詳細につきましては、「6.4.2 カウンタの逡倍」を参照してください。

| モード選択レジスタ | チャンネル  | ×4 | ×1 | ×2 |
|-----------|--------|----|----|----|
| bit15     | —      | —  | —  |    |
| bit14     | —      | —  | —  |    |
| bit13     | チャンネル3 | 0  | 0  | 1  |
| bit12     |        | 0  | 1  | 0  |
| bit11     | —      | —  | —  |    |
| bit10     | —      | —  | —  |    |
| bit9      | チャンネル2 | 0  | 0  | 1  |
| bit8      |        | 0  | 1  | 0  |
| bit7      | —      | —  | —  |    |
| bit6      | —      | —  | —  |    |
| bit5      | チャンネル1 | 0  | 0  | 1  |
| bit4      |        | 0  | 1  | 0  |
| bit3      | —      | —  | —  |    |
| bit2      | —      | —  | —  |    |
| bit1      | チャンネル0 | 0  | 0  | 1  |
| bit0      |        | 0  | 1  | 0  |

表 6-14 カウンタ逡倍選択・レジスタ

## 6) 外部指定ビット入力の状態・レジスタの設定

外部指定ビット入力の状態・レジスタは、【8F0002BCh】にマップされています。このレジスタは、各チャンネルの外部指定ビットの入力状態を読み出すことができます。外部指定ビット入力の状態・レジスタは【bit1】～【bit4】までの4ビットで構成され、【bit0】と【bit5】～【bit31】までは不定です。

各bitが【0】のときは対応する信号の状態が【OFF】で、各bitが【1】のときは対応する信号の状態が【ON】です。

信号状態【OFF】：入力端子をHiレベル（TTLレベル）かオープンにする

信号状態【ON】：入力端子をLoレベル（TTLレベル）かGND接続にする

図 2-8 を参照して下さい

| カン列セット許可<br>レジスタ | チャンネル  | 外部入力指定ビット | 0   | 1  |
|------------------|--------|-----------|-----|----|
| bit4             | チャンネル3 | CN12-28ピン | OFF | ON |
| bit3             | チャンネル2 | CN12-27ピン | OFF | ON |
| bit2             | チャンネル1 | CN12-26ピン | OFF | ON |
| bit1             | チャンネル0 | CN12-25ピン | OFF | ON |
| bit0             | —      | —         | —   | —  |

表 6-15 外部指定ビット入力の状態・レジスタ

## 7. 通信ポート

本ボードには3つの通信ポートがあります。

1. USB通信
2. CAN通信
3. RS232C通信

### 7.1. USB通信

USB通信はホストコンピュータとの通信に使用します。正面にあるUSBコネクタに付属のUSBコネクタを使って接続してください。

### 7.2. CAN通信

CAN通信は高速通信、低速通信の2種類があります。  
 高速通信の終端抵抗値は120Ωです。図 7-9 参照  
 低速通信の終端抵抗値は510Ωと15KΩの2種類があります。図 7-10 参照  
 終端抵抗の有無の設定はジャンパーピンで出来ます。

| 通信モード | 抵抗値 (Ω) | ジャンパー設定      | 終端抵抗有り | 終端抵抗無し |
|-------|---------|--------------|--------|--------|
| 高速設定  | 120     | J3 1-2ピン     | ショート   | オープン   |
| 低速設定  | 510     | J4 1-2と3-4ピン | ショート   | オープン   |
| 低速設定  | 15K     | J4 5-6と7-8ピン | ショート   | オープン   |

表 7-1 CAN 終端抵抗

低速通信の バッテリー電圧は J5によってボード内電圧 (12V) と外部電圧の切り替えが出来ます。外部電圧は CN13-7ピン (低速用CAN\_BAT) に割り当てられています。  
 図 7-10 参照

| バッテリー電圧         | J5         |
|-----------------|------------|
| ボード内部電圧 12V     | 1-2ピン ショート |
| 外部電圧 (CN13-7ピン) | 2-3ピン ショート |

表 7-2 低速用CAN バッテリー電圧設定

CAN通信は高速通信、低速通信が出来ます。DサブコネクタCN13を使用し接続します。  
 ピン配置は表 10-2を参照して下さい。  
 またCAN通信について詳しくは “741-00 ハードマニュアル CAN 通信機能編.doc” を参照して下さい。

### 7.3. RS232C通信

RS232C通信は CN14を使用して接続します。ピン配置は表 7-3を参照して下さい

## 8. ホストコンピュータより本ボードを分離して動作させる方法 (スタンドアロン機能)

ホストコンピュータよりACRO741-00を分離して動作させる。(スタンドアロン機能)  
ユーザーBOOTメモリに書かれているプログラムを他の高速メモリーに転送してプログラムを開始させることができます。よってホストコンピュータが不要になります。

ROM書き込みは、ディップスイッチDSW2-1を書き込み許可【ON】に変更した後、付属のソフトウェアのユーティリティを使って書き込みを行って下さい。

1. ホストコンピュータからプログラムをブートROMに書き込みます。  
下記の設定にして下さい。  
書き込み方は、ソフトウェア・ユーザーズ・マニュアルを参照してください。

|        |                         |
|--------|-------------------------|
| DSW1-6 | OFF<br>ホストコンピュータと接続     |
| DSW1-8 | OFF<br>ホストコンピュータより起動    |
| DSW2-1 | ON<br>ユーザーBOOT・ROM書き込み可 |

表 8-1 ブートROM書き込み可

2. BOOT・ROMからプログラムを転送し起動します。  
下記の設定にして電源を入れてください。

|        |                           |
|--------|---------------------------|
| DSW1-6 | ON<br>ホストコンピュータと切り離し      |
| DSW1-8 | ON<br>ユーザーBOOT・ROMより起動    |
| DSW2-1 | OFF<br>ユーザーBOOT・ROM書き込み不可 |

表 8-2 BOOT ROM起動

## 9. Metrowerks社製 CodeWarrior JTAG使用

本ボードはMetrowerks社製JTAGが使用できるようにJTAG用のコネクタ(CN1)が付いています。CN1にJTAGコネクタを差し込んで使用してください。  
使用方法は CodeWarrior プロジェクト外等作成手順.doc を参照して下さい。

10. コネクタのピン配列とピン配置

10.1. 標準実装 I/O 入出力コネクタ (CN12) のピン配列とピン配置

| No | 信号名        | No | 信号名        | No | 信号名       | No  | 信号名             |
|----|------------|----|------------|----|-----------|-----|-----------------|
| 01 | CNT0_A+    | 02 | CNT0_A-    | 51 | GND       | 52  | PWM2_VP         |
| 03 | CNT0_B+    | 04 | CNT0_B-    | 53 | PWM2_VN   | 54  | GND             |
| 05 | CNT0_Z+    | 06 | CNT0_Z-    | 55 | PWM2_WP   | 56  | PWM2_WN         |
| 07 | CNT1_A+    | 08 | CNT1_A-    | 57 | GND       | 58  | PWM0_CK         |
| 09 | CNT1_B+    | 10 | CNT1_B-    | 59 | PWM0_CK   | 60  | PWM2_CK         |
| 11 | CNT1_Z+    | 12 | CNT1_Z-    | 61 | PWM3_CK   | 62  | GND             |
| 13 | CNT2_A+    | 14 | CNT2_A-    | 63 | N. C      | 64  | N. C            |
| 15 | CNT2_B+    | 16 | CNT2_B-    | 65 | N. C      | 66  | N. C            |
| 17 | CNT2_Z+    | 18 | CNT2_Z-    | 67 | N. C      | 68  | N. C            |
| 19 | CNT3_A+    | 20 | CNT3_A-    | 69 | N. C      | 70  | N. C            |
| 21 | CNT3_B+    | 22 | CNT3_B-    | 71 | O_D0      | 72  | O_D1            |
| 23 | CNT3_Z+    | 24 | CNT3_Z-    | 73 | O_D2      | 74  | O_D3            |
| 25 | CNT0_RESET | 26 | CNT1_RESET | 75 | O_D4      | 76  | O_D5            |
| 27 | CNT2_RESET | 28 | CNT3_RESET | 77 | O_D6      | 78  | O_D7            |
| 29 | GND        | 30 | GND        | 79 | GND       | 80  | GND             |
| 31 | PWM0_UP    | 32 | PWM0_UN    | 81 | I_D0      | 82  | I_D1            |
| 33 | GND        | 34 | PWM0_VP    | 83 | I_D2      | 84  | I_D3            |
| 35 | PWM0_VN    | 36 | GND        | 85 | I_D4      | 86  | I_D5            |
| 37 | PWM0_WP    | 38 | PWM0_WN    | 87 | I_D6      | 88  | I_D7            |
| 39 | GND        | 40 | PWM1_UP    | 89 | GND       | 90  | GND             |
| 41 | PWM1_UN    | 42 | GND        | 91 | N. C      | 92  | N. C            |
| 43 | PWM1_VP    | 44 | PWM1_VN    | 93 | *1        | 94  | *1              |
| 45 | GND        | 46 | PWM1_WP    | 95 | *WD_OUT   | 96  | *EXSYSRESET_OUT |
| 47 | PWM1_WN    | 48 | GND        | 97 | *1        | 98  | *1              |
| 49 | PWM2_UP    | 50 | PWM2_UN    | 99 | *EXSTALON | 100 | *EXSYSRESET_IN  |

表 10-1 CN12のピン配列

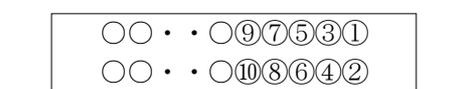


図 10-1 CN12のコネクタを挿入方向からみた図

## 10.2. CANコネクタ (CN13) のピン配置とピン配置

| No | 信号名         | No | 信号名         |
|----|-------------|----|-------------|
| 01 | 高速用 CAN_HI  | 02 | 高速用 CAN_LO  |
| 03 | GND         | 04 | N.C         |
| 05 | 低速用 CAN_HI  | 06 | 低速用 CAN_LO  |
| 07 | 低速用 CAN_BAT | 08 | 低速用 CAN_INH |
| 09 | GND         |    |             |

表 10-2 CN13のピン配置

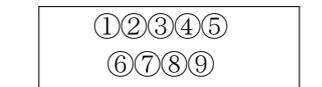


図 10-2 CN13のコネクタを挿入方向からみた図

## 10.3. RS232Cコネクタ (CN14) のピン配置とピン配置

| No | 信号名     | No | 信号名      |
|----|---------|----|----------|
| 01 | N.C     | 02 | DSR (*1) |
| 03 | TXD     | 04 | RTS      |
| 05 | RXD     | 06 | CTS      |
| 07 | DTR(*1) | 08 | N.C      |
| 09 | GND     | 10 | N.C      |

表 10-3 CN14のピン配置

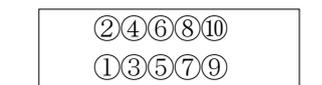


図 10-3 CN14のコネクタを挿入方向からみた図

**【注意】** \*1のDSRとDTRは本ボード内でショートさせて有ります。

## 1 1. 添付品

---

### 1 1. 添付品

| 品 名              | 型 式                   | 数 量 | メーカ  |
|------------------|-----------------------|-----|------|
| USBケーブル          | USB2R01GY             | 1   | アーベル |
| ケーブル付きコネクタ       | 8825E-100-175-157S-G0 | 1   | ケル   |
| コネクタ             | DE-9SF-N              | 1   | JAE  |
| ジャンクションシールド      | DE-C4-J6              | 1   | JAE  |
| ケーブル付コネクタ (コネクタ) | PS-10SEN-D4P1-1C      | 1   | JAE  |
| 上記ケーブル           | 1.5m                  | 1   |      |
| ストレーンリリーフ        | PS-SRN10              | 1   | JAE  |

表 1 1-1 添付品一覧

## 【 改 訂 履 歴 】

| 改訂番号 | 改訂日付       | 改訂内容        |
|------|------------|-------------|
| 初版   | 2010.05.12 | 初版          |
| 第2版  | 2011.09.15 | オプション1Gメモ対応 |
|      |            |             |
|      |            |             |
|      |            |             |
|      |            |             |

- ・本製品及び本マニュアルの内容は製品の改良のため予告無しに変更される事がありますのでご了承下さい。
- ・本製品及び本マニュアルの内容は著作権法により保護されています。許可なく複製する事はできません。

**ACRO741-00**  
ハードウェア・ユーザズ・マニュアル

**中部電機株式会社**

〒440-0004 愛知県豊橋市忠興3丁目2-8

TEL <0532>61-9566 FAX <0532>63-1081

URL : <http://www.chubu-el.co.jp>

E-mail : [csg@chubu-el.co.jp](mailto:csg@chubu-el.co.jp)

2011.09 第2版発行

@CROシリーズ

# ACRO741-00

ソフトウェアユーザーズ・マニュアル



## 目 次

|        |  |    |
|--------|--|----|
| 1.     | ボードコントロールライブラリ                           | 1  |
| 1.1.   | 概要                                       | 1  |
| 1.2.   | 使用可能アプリケーション                             | 1  |
| 1.3.   | 関数一覧                                     | 2  |
| 1.3.1. | ライブラリー制御関数                               | 2  |
| 1.3.2. | ボード関連関数                                  | 2  |
| 1.3.3. | ボード制御関数                                  | 2  |
| 1.3.4. | アップ/ダウンロード関数                             | 2  |
| 1.3.5. | 特殊制御関数                                   | 2  |
| 1.3.6. | オフライン関数                                  | 2  |
| 1.4.   | 関数リファレンス (アルファベット順)                      | 3  |
| 1)     | AP74X_CloseDevice (デバイスドライバクローズ)         | 3  |
| 2)     | AP74X_GenInt (ホストからボードへ割込み発生)            | 4  |
| 3)     | AP74X_GetArray (@CROボードからデータ読み込み)        | 5  |
| 4)     | AP74X_GetChar (ボードからデータ読み込み)             | 6  |
| 5)     | AP74X_GetExtSize (実装されているSDSRAMの容量読み込み)  | 7  |
| 6)     | AP74X_GetMem (ボードから1ワードデータ読み込み)          | 8  |
| 7)     | AP74X_GetSymbole (シンボル情報取得)              | 9  |
| 8)     | AP74X_GetVersion (デバイス情報の取得) ※USB接続時のみ有効 | 10 |
| 9)     | AP74X_Load (プログラムロード)                    | 11 |
| 10)    | AP74X_OpenDevice (デバイスドライバオープン)          | 12 |
| 11)    | AP74X_PutArray (ボードへデータ書き込み)             | 13 |
| 12)    | AP74X_PutChar (ボードへデータ書き込み)              | 14 |
| 13)    | AP74X_PutMem (ボードへ1ワードデータ書き込み)           | 15 |
| 14)    | AP74X_ResetBd (ボードリセット)                  | 16 |
| 15)    | AP74X_Resume (デバイスドライバの強制解放)             | 17 |
| 16)    | AP74X_Run (プログラム実行)                      | 18 |
| 17)    | AP74X_SelBd (ボード選択)                      | 19 |
| 18)    | AP74X_Valid (ボード実装確認)                    | 20 |
| 1.5.   | エラーコード表                                  | 21 |

|        |                                       |    |
|--------|---------------------------------------|----|
| 2.     | FFTライブラリ .....                        | 22 |
| 2.1.   | 概要 .....                              | 22 |
| 2.2.   | 使用可能アプリケーション .....                    | 22 |
| 2.3.   | フォルダ構成 .....                          | 22 |
| 2.4.   | FFT計算ライブラリ関数リファレンス (アルファベット順) .....   | 23 |
| 1)     | Lib_ffftend (終了処理) .....              | 23 |
| 2)     | Lib_ffftinit (初期処理) .....             | 23 |
| 3)     | Lib_ffft1 (1次元FFT計算) .....            | 23 |
| 4)     | Lib_ffft2 (2次元FFT計算) .....            | 24 |
| 5)     | Lib_ifft1 (1次元IFFT計算) .....           | 24 |
| 6)     | Lib_ifft2 (2次元IFFT計算) .....           | 24 |
| 2.5.   | サンプルプログラム説明 .....                     | 25 |
| 2.5.1. | 概要 .....                              | 25 |
| 2.5.2. | 構成 .....                              | 25 |
| 2.5.3. | 内容 .....                              | 25 |
| 2.6.   | サンプルプログラム説明 (別途オプションソフトウェア) .....     | 26 |
| 2.6.1. | 概要 .....                              | 26 |
| 2.6.2. | 構成 .....                              | 26 |
| 2.6.3. | ホスト側サンプル用DLL関数リファレンス (アルファベット順) ..... | 26 |
| 1)     | AP74X_ffftend (終了処理) .....            | 26 |
| 2)     | AP74X_ffftinit (初期処理) .....           | 26 |
| 3)     | AP74X_ffft (1次元FFT) .....             | 27 |
| 4)     | AP74X_ifft (1次元IFFT) .....            | 27 |
| 5)     | AP74X_ffft2 (2次元FFT) .....            | 28 |
| 6)     | AP74X_ifft2 (2次元IFFT) .....           | 28 |
| 2.6.4. | ホスト側サンプルアプリケーション .....                | 29 |

## 1. ボードコントロールライブラリ

### 1.1. 概要

ボードコントロールライブラリは、ACRO741-00に標準で添付されているソフトウェアです。このソフトウェアは、ACRO741-00ボードを、ユーザーアプリケーションから操作するための橋渡しを行うソフトウェアで、ACRO741-00にユーザープログラムをダウンロードしたり、メモリ操作を行うために利用します。

### 1.2. 使用可能アプリケーション

このボードコントロールライブラリは、Microsoft 社製 Visual Studio V6.0/.NET/2005 VC++/VBにて使用できます。

### 1.3. 関数一覧

#### 1.3.1. ライブラリー制御関数

|  |    |
|--|----|
| AP74X_OpenDevice (デバイスドライバーオープン) ..... | 12 |
| AP74X_CloseDevice (デバイスドライバクローズ) ..... | 3  |
| AP74X_Resume (デバイスドライバーの強制解放) .....    | 17 |

#### 1.3.2. ボード関連関数

|   |    |
|---|----|
| AP74X_SelBd (ボード選択) .....                     | 19 |
| AP74X_Valid (ボード実装確認) .....                   | 20 |
| AP74X_GetExtSize (実装されているSDSRAMの容量読み込み) ..... | 7  |

#### 1.3.3. ボード制御関数

|                               |    |
|-------------------------------|----|
| AP74X_Run (プログラム実行) .....     | 18 |
| AP74X_ResetBd (ボードリセット) ..... | 16 |

#### 1.3.4. アップ/ダウンロード関数

|   |    |
|---|----|
| AP74X_GetMem (ボードから1ワードデータ読み込み) .....   | 8  |
| AP74X_GetArray (@CROボードからデータ読み込み) ..... | 5  |
| AP74X_PutMem (ボードへ1ワードデータ書き込み) .....    | 15 |
| AP74X_PutArray (ボードへデータ書き込み) .....      | 13 |
| AP74X_GetChar (ボードからデータ読み込み) .....      | 6  |
| AP74X_PutChar (ボードへデータ書き込み) .....       | 14 |
| AP74X_Load (プログラムロード) .....             | 11 |

#### 1.3.5. 特殊制御関数

|  |    |
|--|----|
| AP74X_GenInt (ホストからボードへ割り込み発生) .....           | 4  |
| AP74X_GetVersion (デバイス情報の取得) ※USB接続時のみ有効 ..... | 10 |

#### 1.3.6. オフライン関数

|                                   |   |
|-----------------------------------|---|
| AP74X_GetSymbole (シンボル情報取得) ..... | 9 |
|-----------------------------------|---|

#### 1.4. 関数リファレンス (アルファベット順)

この関数リファレンスは、Microsoft 社製 VisualStudio VC++/VB での使用方法について説明しています。VB については、Microsoft 社製 VisualStudio のバージョンによりデータ型の長さがことなりますので、注意してください。

VC++の場合、関数名称の先頭に ‘\_’ が有る関数は現在のボード (AP74X\_SelBd で選択したボード) を対象に実行される関数です。

VB の場合、関数名称の先頭に ‘S’ が有る関数は現在のボード (AP74X\_SelBd で選択したボード) を対象に実行される関数です。

##### 1) AP74X\_CloseDevice (デバイスドライバクローズ)

**【機能】** デバイスドライバ等を解放します。ユーザープログラム終了直前にこの関数を実行してください。この関数を実行しないで終了した場合、次回デバイスオープン (AP74X\_OpenDevice) が実行できません。  
もし、未実行でプログラムを終了した場合は、「リジューム」 (AP74X\_Resume) でデバイスを開放してください。

**【引数】** なし

**【書式】** VC : void AP74X\_CloseDevice(void)  
VB (Ver6.0) : Sub AP74X\_CloseDevice()  
VB (.NET 以降) : Sub AP74X\_CloseDevice()

**【戻値】** なし

**【参考】** AP74X\_OpenDevice

2) AP74X\_GenInt (ホストからボードへ割り込み発生)

**【機能】** ホストからボードへ割り込みを発生させます。  
ホストからボードへの同期に利用できます。  
この割り込みは、ハードウェアによる割り込みです。

**【引数】** bdn0                      ボード番号

**【書式】** VC                              : int              AP74X\_GenInt(int bdn0)  
VB (Ver6.0)                            : Function      AP74X\_GenInt(ByVal bdn0 As Long) As Long  
VB (.NET以降)                         : Function      AP74X\_GenInt(ByVal bdn0 As Integer) As Integer

**【戻値】** 0              正常終了  
0以外      異常終了

## 3) AP74X\_GetArray (@CROボードからデータ読み込み)

【機能】 ホストはボードからデータを読み込みます。データ長は任意のサイズで、単位はワード（4バイト）で指定します。

【引数】

|      |              |
|------|--------------|
| bdno | ボード番号        |
| src  | 転送元(ボード)アドレス |
| len  | 読み込みワード数     |
| dest | 受信(ホスト)アドレス  |

【書式】 VC :

```
int AP74X_GetArray( int bdno, int src, int len, void *dest)
int _AP74X_GetArray( uint src, int len, void *dest)
```

VB (Ver6.0) :

```
Function AP74X_GetArray( ByVal bdno As Long, ByVal src As Long,
                        ByVal len As Long, ByRef dest As Any) As Long
Function SAP74X_GetArray( ByVal src As Long,
                        ByVal len As Long, ByRef dest As Any) As Long
```

VB (.NET 以降) :

```
Function AP74X_GetArray( ByVal bdno As Integer, ByVal src As Integer,
                        ByVal len As Integer, ByRef dest As Integer) As Integer
Function SAP74X_GetArray( ByVal src As Integer,
                        ByVal len As Integer, ByRef dest As Integer) As Integer
```

【戻値】 0 正常終了  
 -1 パラメータエラー  
 -2 読み込みエラー  
 その他 別紙エラーコード表参照 (P.21)

【参考】 AP74X\_PutArray, AP74X\_GetMem

【使用例】

```
VC      int    buffer[0x100];
        if( AP74X_GetArray(0, 0x1000, 0x100, buffer ) {
            printf( “データ取得エラー\n” );
        }
VB (Ver6.0) Dim buffer(&H100) As Long
            If AP74X_GetArray(0, &H1000, &H100, buffer) Then
                Print “データ取得エラー”
            End If
```

#### 4) AP74X\_GetChar (ボードからデータ読み込み)

**【機能】** ホストはボードから Char データを読み込みます。データ長は任意のサイズで、単位はバイトで指定します。

**【引数】**

|      |              |
|------|--------------|
| bdno | ボード番号        |
| src  | 転送元(ボード)アドレス |
| len  | 読み込みバイト数     |
| dest | 受信(ホスト)アドレス  |

**【書式】 VC :**

```
int AP74X_GetChar(int bdno, uint src, int len, char *dest)
int _AP74X_GetChar(uint src, int len, char *dest)
```

**VB (Ver6.0) :**

```
Function AP74X_GetChar(ByVal bdno As Long, ByVal src As Long,
                        ByVal len As Long, ByVal dest As Byte) As Long
Function SAP74X_GetChar(ByVal src As Long,
                        ByVal len As Long, ByVal dest As Byte) As Long
```

**VB (.NET 以降) :**

```
Function AP74X_GetChar(ByVal bdno As Integer, ByVal src As Integer,
                        ByVal len As Integer, ByVal dest As Byte) As Integer
Function SAP74X_GetChar(ByVal src As Integer,
                        ByVal len As Integer, ByVal dest As Byte) As Integer
```

**【戻値】** 0 正常終了  
-1 パラメータエラー  
-2 読み込みエラー  
その他 別紙エラーコード表参照 (P. 21)

**【参考】** AP74X\_PutChar, AP74X\_GetMem

**【使用例】**

```
VC
char buffer[0x100+1];
if( AP74X_GetChar(0, 0x1000, 0x100, buffer ) {
    printf( “データ取得エラー\n” );
}

VB (Ver6.0)
Dim buffer(&H100) As Byte
If AP74X_GetChar(0, &H1000, &H100, buffer(0)) Then
    Print “データ取得エラー”
End If
```

## 5) AP74X\_GetExtSize (実装されているSDSRAMの容量読み込み)

【機能】 ボードに実装されているSDSRAMの容量(サイズ)を読み込みます。この関数はAP74X\_OpenDevice()関数を実行した後に有効となります。

【引数】 bdno            ボード番号  
size                サイズ

【書式】 VC :

```
int AP74X_GetExtSize(int bdno, uint *size)
int _AP74X_GetExtSize(uint *size)
```

VB (Ver6.0) :

```
Function AP74X_GetExtSize(ByVal bdno as Long, ByRef size as Long) As Long
Function SAP74X_GetExtSize(ByRef size as Long) As Long
```

VB (.NET 以降) :

```
Function AP74X_GetExtSize(ByVal bdno as Integer, ByRef size as Integer) As Integer
Function SAP74X_GetExtSize(ByRef size as Integer) As Integer
```

【戻値】 0            正常終了  
-1            パラメータエラー

【使用例】

```
VC            unsigned int    size;
              if(AP74X_GetExtSize(0, &size) {
                  printf( "サイズ取得エラー\n" );
              }
VB (Ver6.0) Dim size As Long
              If AP74X_GetExtSize(0, size) Then
                  Print "サイズ取得エラー"
              End If
```

6) AP74X\_GetMem (ボードから1ワードデータ読み込み)

【機能】ホストはボードから1ワード(4バイト)データを読み込みます。

【引数】 bdno            ボード番号  
         src            転送元(ボード)アドレス  
         dest           受信(ホスト)アドレス

【書式】 VC :

```
int AP74X_GetMem(int bdno, uint src, void *dest)
int _AP74X_GetMem(uint src, void *dest)
```

VB (Ver6.0) :

```
Function AP74X_GetMem(ByVal bdno as Long, ByVal src as Long,
                     ByRef dest as Any) As Long
Function SAP74X_GetMem(ByVal src as Long, ByRef dest as Any) As Long
```

VB (.NET 以降) :

```
Function AP74X_GetMem(ByVal bdno as Integer, ByVal src as Integer,
                     ByRef dest as Any) As Integer
Function SAP74X_GetMem(ByVal src as Integer, ByRef dest as Any) As Integer
```

【戻値】 0            正常終了  
         -1           パラメータエラー  
         その他 別紙エラーコード表参照 (P. 21)

【参考】 AP74X\_PutMem, AP74X\_GetArray

【使用例】

```
VC            float    buffer;
              if(AP74X_GetMem(0, 0x1000, &buffer){
                  printf( "データ取得エラー\n" );
              }
VB (Ver6.0) Dim buffer As Long
              If AP74X_GetMem(0, &H1000, buffer) Then
                  Print "データ取得エラー"
              End If
```

## 7) AP74X\_GetSymbole (シンボル情報取得)

【機能】 プログラム実行ファイル (ELFファイル) より、シンボルアドレス情報を取得します。

【引数】 fname           ファイル名  
          sname        シンボル名  
          address      アドレス

【書式】 VC :

```
int AP74X_GetSymbole(const char *fname, const char *sname, uint*address)
```

VB (Ver6.0) :

```
Sub AP74X_GetSymbole(ByRef fname As String, ByRef sname As String,  
                    ByRef address As Long) As Long
```

VB (.NET 以降) :

```
Sub AP74X_GetSymbole(ByRef fname As String, ByRef sname As String,  
                    ByRef address As Integer) As Integer
```

【戻値】 0     正常終了  
         -1    パラメータエラー  
         -2    ファイル異常

【備考】

プログラムRUN前にホストより、@CROプログラムのグローバル変数へ値を設定して使用する場合は、グローバル変数を定義する際、ゼロ以外の値で初期値セットをしてください。(変数が data セクションに定義されるように)初期値セットなし、又はゼロで初期値セットをする (変数が bss セクションに定義される) とプログラムRUN時にゼロセットされます。

8) AP74X\_GetVersion (デバイス情報の取得) ※USB 接続時のみ有効

【機能】 USBデバイスの情報を取得します。

【引数】 bdno                    ボード番号  
          ver                    バージョン番号

【書式】 VC :

```
int AP74X_GetVersion(int bdno, uint *ver)
```

VB (Ver6.0) :

```
Function AP74X_GetVersion(ByVal bdno As Long, ByRef *ver As Long) As Long
```

VB (.NET 以降) :

```
Function AP74X_GetVersion(ByVal bdno As Integer, ByRef *ver As Integer) As Integer
```

【戻値】 0     正常終了  
         -1     パラメータエラー  
         -2     該当ボードが見つかりません

【使用例】

```
VC           uint     *verno;  
             int     ret;  
             ret = AP74X_GetVersion(0, &verno);  
             switch(ret) {  
                  case -1: printf( "パラメータエラー\n" )  
                              break;  
                  case -1: printf( "該当ボードなし\n" )  
                              break;  
              }  
             }
```

```
VB (Ver6.0) Dim verno As Long  
           Dim ret As Long  
           ret = AP74X_GetVersion(0, verno)  
           Select Case ret  
              Case -1: Print "パラメータエラー"  
              Case -2: Print "該当ボードなし"  
           End Select
```

## 9) AP74X\_Load (プログラムロード)

**【機能】** プログラムファイルをボードへダウンロードします。ロード可能なファイルはリンカーが出力した実行形式モジュールです。

**【引数】** bdno            ボード番号  
filename            ファイル名

**【書式】 VC :**

```
int AP74X_Load(int bdno, const char *filename)
int _AP74X_Load(const char *filename)
```

**VB (Ver6.0) :**

```
Function AP74X_Load(ByVal bdno As Long, ByVal filename as String) As Long
Function SA74X_Load(ByVal filename as String) As Long
```

**VB (.NET 以降) :**

```
Function AP74X_Load(ByVal bdno As Long, ByVal filename as String) As Integer
Function SA74X_Load(ByVal filename as String) As Integer
```

**【戻値】** 0    正常終了  
-1    パラメータエラー  
-2    ファイルが見つからない  
-3    ファイル異常  
-4    ダウンロード異常

**【使用例】**

```
VC            if( AP74X_Load(0, "UserPg.elf" )){
                      printf( "ロードエラー\n" );
                      }
VB (Ver6.0) If AP74X_Load(0, "UserPg.elf" ) Then
                      Print "ロードエラー"
                      End If
```

10) AP74X\_OpenDevice (デバイスドライバーオープン)

**【機能】** デバイスドライバのオープンとライブラリの初期化を行います。ライブラリの機能を使用するに先立って、この関数を必ず実行してください。

**【引数】** なし

**【書式】** VC : int AP74X\_OpenDevice(void)  
VB (Ver6.0) : Function AP74X\_OpenDevice() As Long  
VB (.NET以降) : Function AP74X\_OpenDevice() As Integer

**【戻値】** 0 正常終了  
-1 デバイスドライバがオープンできません  
-2 すでにデバイスがオープンされています  
(前回 AP74X\_CloseDevice を実行せずに終了した。「リジューム」を実行して下さい)  
-3 ボードが1枚も見つかりません

**【参考】** AP74X\_CloseDevice, AP74X\_Resume

## 11) AP74X\_PutArray (ボードヘデータ書き込み)

【機能】 ホストからボードヘデータを書き込みます。データ長は任意サイズで、ワード（4バイト）単位です。

【引数】

|      |                  |
|------|------------------|
| bdno | ボード番号            |
| src  | 転送先(ボード)アドレス     |
| len  | 書き込みワード数         |
| dest | 転送元(ホスト)アドレス     |
| chk  | ベリファイ有無(0:無 1:有) |

## 【書式】 VC :

```
int AP74X_PutArray(int bdno, uint src, int len, void *dest, int chk)
int _AP74X_PutArray(uint src, int len, void *dest, int chk)
```

## VB (Ver6.0) :

```
Function AP74X_PutArray(ByVal bdno As Long, ByVal src As Long,
                        ByVal len As Long, ByRef dest As Any,
                        ByVal chk As Long) As Long
Function SAP74X_PutArray(ByVal src As Long,
                        ByVal len As Long, ByRef dest As Any,
                        ByVal chk As Long) As Long
```

## VB (.NET 以降) :

```
Function AP74X_PutArray(ByVal bdno As Integer, ByVal src As Integer,
                        ByVal len As Integer, ByRef dest As Any,
                        ByVal chk As Integer) As Integer
Function SAP74X_PutArray(ByVal src As Integer,
                        ByVal len As Integer, ByRef dest As Any,
                        ByVal chk As Integer) As Integer
```

【戻値】 0 正常終了  
 -1 パラメータエラー  
 -2 書き込みエラー  
 その他 別紙エラーコード表参照 (P.21)

【参考】 AP74X\_PutMem, AP74X\_GetArray

## 【使用例】

```
VC      int    buffer[0x100];
        if(AP74X_PutArray(0, 0x1000, 0x100, buffer){
            printf( "データ転送エラー\n" );
```

```
VB (Ver6.0) Dim    buffer(&H100) As Long
            If AP74X_PutArray(0, &H1000, &H100, buffer) Then
                Print "データ転送エラー"
            End If
```

## 12) AP74X\_PutChar (ボードへデータ書き込み)

**【機能】** ホストからボードへデータを書き込みます。データ長は任意サイズで、バイト単位です。

**【引数】**

|      |                  |
|------|------------------|
| bdno | ボード番号            |
| src  | 転送先(ボード)アドレス     |
| len  | 書き込みバイト数         |
| dest | 転送元(ホスト)アドレス     |
| chk  | ベリファイ有無(0:無 1:有) |

**【書式】 VC :**

```
int AP74X_PutChar(int bdno, uint src, int len, char *dest, int chk)
int _AP74X_PutChar(uint src, int len, char *dest, int chk)
```

**VB (Ver6.0) :**

```
Function AP74X_PutChar(ByVal bdno As Long, ByVal src As Long,
    ByVal len As Long, ByVal dest As Byte,
    ByVal chk As Long) As Long
Function SAP74X_PutChar(ByVal src As Long,
    ByVal len As Long, ByVal dest As Byte,
    ByVal chk As Long) As Long
```

**VB (.NET 以降) :**

```
Function AP74X_PutChar(ByVal bdno As Integer, ByVal src As Integer,
    ByVal len As Integer, ByVal dest As Byte,
    ByVal chk As Integer) As Integer
Function SAP74X_PutChar(ByVal src As Integer,
    ByVal len As Integer, ByVal dest As Byte,
    ByVal chk As Integer) As Integer
```

**【戻値】** 0 正常終了  
-1 パラメータエラー  
-2 書き込みエラー  
その他 別紙エラーコード表参照 (P. 21)

**【注意】**

実際に PowerPC への書き込みは、アドレスアライメント 4 バイトで、4 バイト単位でおこないます。

**【参考】** AP74X\_GetChar, AP74X\_PutMem

**【使用例】**

```
VC
char buffer[0x100+1];
if(AP74X_PutChar(0, 0x1000, 0x100, buffer){
    printf(“データ転送エラー\n”);
}
VB (Ver6.0)
Dim buffer(&H100) As Byte
If AP74X_PutChar(0, &H1000, &H100, buffer(0)) Then
    Print “データ転送エラー”
End If
```

**13) AP74X\_PutMem (ボードへ1ワードデータ書き込み)**

**【機能】** ホストからボードへ1ワード（4バイト）データを書き込みます。

**【引数】**

|      |                  |
|------|------------------|
| bdno | ボード番号            |
| src  | 転送先(ボード)アドレス     |
| dest | 転送元(ホスト)アドレス     |
| chk  | ベリファイ有無(0:無 1:有) |

**【書式】 VC :**

```
int AP74X_PutMem(int bdno, uint src, void *dest, int chk)
int _AP74X_PutMem(uint src, void *dest, int chk)
```

**VB (Ver6.0) :**

```
Function AP74X_PutChar (ByVal bdno As Long, ByVal src As Long,
                        ByRef dest As Any, ByVal chk As Long) As Long
Function SAP74X_PutChar (ByVal src As Long,
                        ByRef dest As Any, ByVal chk As Long) As Long
```

**VB (.NET 以降) :**

```
Function AP74X_PutChar (ByVal bdno As Integer, ByVal src As Integer,
                        ByRef dest As Any, ByVal chk As Integer) As Integer
Function SAP74X_PutChar (ByVal src As Integer,
                        ByRef dest As Any, ByVal chk As Integer) As Integer
```

**【戻値】**

|     |                     |
|-----|---------------------|
| 0   | 正常終了                |
| -1  | パラメータエラー            |
| -2  | 書き込みエラー             |
| その他 | 別紙エラーコード表参照 (P. 21) |

**【参考】** AP74X\_GetMem, AP74X\_PutArray

**【使用例】**

```
VC
int buffer;
if(AP74X_PutMem(0, 0x1000, &buffer){
    printf(“データ転送エラー\n”);
}

VB (Ver6.0)
Dim buffer As Long
If AP74X_PutArray(0, &H1000, buffer) Then
    Print “データ転送エラー”
End If
```

14) AP74X\_ResetBd (ボードリセット)

【機能】 ボードを初期化します。通常は、AP74X\_OpenDevice 関数にて行っています。

【引数】 bdnno                    ボード番号

【書式】 VC                    : int    AP74X\_ResetBd(int bdnno)  
                                int    \_AP74X\_ResetBd(void)  
VB (Ver6.0)                   : Function AP74X\_ResetBd(ByVal bdnno As Long) As Long  
                                Function    SAP74X\_ResetBd() As Long  
VB (.NET以降)                 : Function AP74X\_ResetBd(ByVal bdnno As Integer) As Integer  
                                Function    SAP74X\_ResetBd() As Integer

【戻値】 0                    正常終了  
          -1                  パラメータエラー  
          上記以外            その他エラー

## 15) AP74X\_Resume (デバイスドライバーの強制解放)

【機能】 ユーザープログラムの異常終了などで、ロックされてしまったデバイスドライバーを強制的に開放します。

【引数】 なし

【書式】 VC : int AP74X\_Resume(void)  
VB (Ver6.0) : Function AP74X\_Resume() As Long  
VB (.NET 以降) : Function AP74X\_Resume() As Integer

【戻値】 0 正常終了  
-1 異常終了

## 【使用例】

```
VC      if(AP74X_Resume()){  
                printf(“デバイスの強制開放エラー\n”);  
        }  
VB (Ver6.0) If AP74X_Resume Then  
                Print “デバイスの強制開放エラー”  
End If
```

16) AP74X\_Run (プログラム実行)

【機能】 指定ボードを実行状態にします。

【引数】 bdno                    ボード番号

【書式】 VC                    : int AP74X\_Run(int bdno)  
VB (Ver6.0)                : Function AP74X\_Run(ByVal bdno As Long) As Long  
VB (.NET以降)             : Function AP74X\_Run(ByVal bdno As Integer) As Integer

【戻値】 0            正常終了  
         -1           パラメータエラー

【使用例】

## 17) AP74X\_SelBd (ボード選択)

【機能】 指定ボードを選択状態にします。

【引数】 bdno                    ボード番号

【書式】 VC                    : int AP74X\_SelBd(int bdno)  
VB (Ver6.0)                   : Function AP74X\_SelBd(ByVal bdno As Long) As Long  
VB (.NET以降)                 : Function AP74X\_SelBd(ByVal bdno As Integer) As Integer

【戻値】 0        正常終了  
         -1       パラメータエラー  
         -2       選択状態にできませんでした  
                  指定したボードが実装されていません

【使用例】

### 18) AP74X\_Valid (ボード実装確認)

**【機能】** 指定されたボードが実装されているかどうか確認します。ボード実装の可否は AP74X\_OpenDevice デバイスオープン時に確認をした情報で行います。

**【引数】** bdno                    ボード番号

**【書式】** VC                    : int AP74X\_Valid(int bdno)  
VB (Ver6.0)                : Function AP74X\_Valid(ByVal bdno As Long) As Long  
VB (.NET以降)             : Function AP74X\_Valid(ByVal bdno As Integer) As Integer

**【戻値】** 0            ボードは実装されていません  
          1            ボードは実装されています  
          -1           パラメータエラー

**【参考】** AP74X\_OpenDevice

#### **【使用例】**

```
VC            If (AP74X_Valid(0) {  
                  printf( “ボードは実装されています” );  
                  }  
VB (Ver6.0)   If AP74X_Valid(0) Then  
                  Print “ボードは実装されています”  
                  End If
```



## 2. FFTライブラリ

### 2.1. 概要

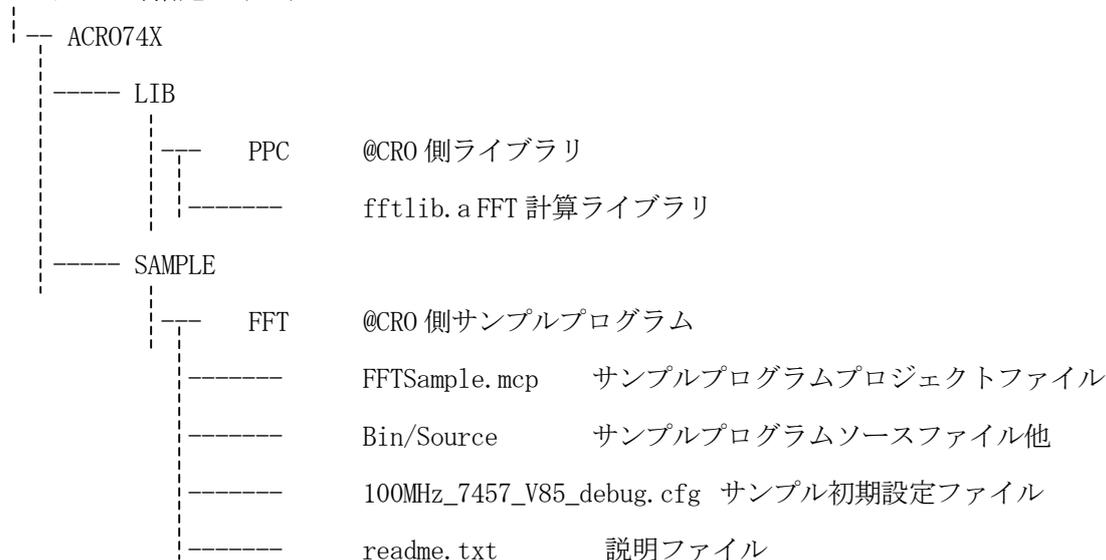
FFTライブラリは、ACRO741-00にてFFT計算を実行するターゲット側のライブラリです。ホスト（PC）からターゲット側の計算ライブラリを操作するライブラリは、別途オプションソフトウェアとなります。

### 2.2. 使用可能アプリケーション

ライブラリは FreeScale 社製 CodeWarrior V8.5にて作成しています。  
このライブラリを使用して、ホスト（PC）側よりデータを受け渡す、Microsoft 社製 Visual Studio 2005 VC++にて作成してる、別途オプションソフトウェアもあります。詳細につきましては、お問い合わせください。

### 2.3. フォルダ構成

インストール時指定フォルダ



## 2.4. FFT 計算ライブラリ関数リファレンス (アルファベット順)

この関数は、@CRO ボードにてFFTを計算するライブラリ関数です。

### 1) Lib\_ffftend (終了処理)

**【機能】** FFTライブラリの終了処理として、Lib\_ffftinit()にて確保した領域の解放等を行います。

**【引数】** なし

**【書式】** void Lib\_ffftend(void)

**【戻値】** なし

**【参考】** Lib\_ffftinit

### 2) Lib\_ffftinit (初期処理)

**【機能】** FFTライブラリの初期処理として、計算領域の確保等を行います。この関数を実行したら必ず終了前にLib\_ffftend()にて終了処理を実行してください。

**【引数】**

|       |                   |
|-------|-------------------|
| rdata | 実数部データの先頭アドレス     |
| idata | 虚数部データの先頭アドレス     |
| n     | FFT点数 (2の累乗であること) |
| dim   | FFTの次元 (1又は2)     |

**【書式】** int Lib\_ffftinit(float \*\*rdata, float \*\*idata, int n, int dim)

**【戻値】** 0 正常終了  
-1 引数エラー  
-2 領域確保エラー

**【参考】** Lib\_ffftend

### 3) Lib\_ffft1 (1次元FFT計算)

**【機能】** Lib\_ffftinit()にて設定した内容で、1次元のFFT計算を実行します。データはLib\_ffftinit()で確保した実数部・虚数部のデータアドレスにて計算し、結果は同じアドレスに返されます。

**【引数】** なし

**【書式】** void Lib\_ffft1(void)

**【戻値】** なし

**【参考】** Lib\_ifft1

### 4) Lib\_fft2 (2次元FFT計算)

【機能】 Lib\_ffftinit()にて設定した内容で、2次元のFFT計算を実行します。データは Lib\_ffftinit()で確保した実数部・虚数部のデータアドレスにて計算し、結果は同じアドレスに返されます。

【引数】 なし

【書式】 void Lib\_fft2(void)

【戻値】 なし

【参考】 Lib\_ifft2

### 5) Lib\_ifft1 (1次元IFFT計算)

【機能】 Lib\_ffftinit()にて設定した内容で、1次元のIFFT計算を実行します。データは Lib\_ffftinit()で指定した実数部・虚数部のデータを下に計算し、結果は同じアドレスに返されますので、計算前のデータは関数実行後破棄されます。

【引数】 なし

【書式】 void Lib\_ifft1(void)

【戻値】 なし

【参考】 Lib\_ffft1

### 6) Lib\_ifft2 (2次元IFFT計算)

【機能】 Lib\_ffftinit()にて設定した内容で、2次元のIFFT計算を実行します。データは Lib\_ffftinit()で指定した実数部・虚数部のデータを下に計算し、結果は同じアドレスに返されますので、計算前のデータは関数実行後破棄されます。

【引数】 なし

【書式】 void Lib\_ifft2(void)

【戻値】 なし

【参考】 Lib\_fft2

## 2.5. サンプルプログラム説明

### 2.5.1. 概要

このサンプルプログラムは、FFT計算ライブラリをターゲット (@CRO) にて実行するためのサンプルソフトウェアです。

### 2.5.2. 構成

FreeScale 社製 CodeWarrior V8.5 にて作成しています @CRO 側サンプルアプリケーションから構成されています。

### 2.5.3. 内容

このサンプルは、CodeWarrior にて JTAG を使用した場合のサンプルプログラムです。FFTSample.mcp がプロジェクトファイルになります。JTAG, イニシャルファイル設定を行ってから実行してください。

変数:hz で指定した周波数のサイン波データを FFT/IFFT 処理するサンプルプログラムです。結果は、\*ar,\*ai へ配列データとして格納されます。また、FFT点数は、変数:n で指定してください。

### 2.6. サンプルプログラム説明 (別途オプションソフトウェア)

#### 2.6.1. 概要

このサンプルプログラムは、F F T計算ライブラリをホスト ( P C ) から実行するためのサンプルソフトウェアです。

#### 2.6.2. 構成

Microsoft 社製 Visual Studio 2005 VC++にて作成していますホスト側サンプルアプリケーションとD L L、FreeScale 社製CodeWarrior V8.5にて作成しています@ C R O側サンプルアプリケーションから構成されています。

#### 2.6.3. ホスト側サンプル用D L L関数リファレンス (アルファベット順)

この関数は、F F Tライブラリサンプルソフトウェアのホスト側サンプルを支援するためのD L L関数です。

##### 1) AP74X\_ffftend (終了処理)

**【機能】** F F Tライブラリの終了処理、Lib\_ffftend()を@ C R O側サンプルアプリケーションを使用して実行します。

**【引数】** なし

**【書式】** void AP74X\_ffftend(void)

**【戻値】** なし

**【参考】** AP74X\_ffftinit

##### 2) AP74X\_ffftinit (初期処理)

**【機能】** F F Tライブラリの初期処理、Lib\_ffftinit()を@ C R O側サンプルアプリケーションを使用して実行します。

**【引数】** n                    F F T点数 ( 2 の累乗にて指定)  
dim                    F F T次元 ( 1 又は 2 )  
fname                @ C R O側サンプルアプリケーション実行ファイル名 (フルパスにて指定)

**【書式】** int AP74X\_ffftinit(int n, int dim, char \*fname)

**【戻値】** 0        正常終了  
1        デバイスドライバ初期化エラー  
2        @ C R O側サンプルアプリケーション実行エラー  
-1       引数エラー  
-2       領域確保エラー

**【参考】** AP74X\_ffftend

### 3) AP74X\_fft (1次元FFT)

【機能】FFTライブラリの1次元FFT計算、Lib\_fft1()を@CRO側サンプルアプリケーションを使用して実行します。

【引数】 in            計算データ先頭アドレス  
      rdata        FFT計算後の実数部データ先頭アドレス  
      idata        FFT計算後の虚数部データ先頭アドレス

【書式】 int AP74X\_fft(float \*in, float \*rdata, float \*idata)

【戻値】 0        正常終了

【参考】 AP74X\_ifft

### 4) AP74X\_ifft (1次元IFFT)

【機能】FFTライブラリの1次元IFFT計算、Lib\_ifft1()を@CRO側サンプルアプリケーションを使用して実行します。

【引数】 in\_rl        FFT計算結果の実数部データ先頭アドレス  
      in\_im        FFT計算結果の虚数部データ先頭アドレス  
      rdata        IFFT計算後の実数部データ先頭アドレス  
      idata        IFFT計算後の虚数部データ先頭アドレス

【書式】 int AP74X\_ifft(float \*in\_rl, float in\_im, float \*rdata, float \*idata)

【戻値】 0        正常終了

【参考】 AP74X\_fft

### 5) AP74X\_fft2 (2次元FFT)

**【機能】** FFTライブラリの2次元FFT計算、Lib\_fft2()を@CRO側サンプルアプリケーションを使用して実行します。

**【引数】** in            計算データ先頭アドレス  
rdata            FFT計算後の実数部データ先頭アドレス  
idata            FFT計算後の虚数部データ先頭アドレス

**【書式】** int AP74X\_fft2(short \*in, float \*rdata, float \*idata)

**【戻値】** 0        正常終了

**【参考】** AP74X\_ifft2

### 6) AP74X\_ifft2 (2次元IFFT)

**【機能】** FFTライブラリの2次元IFFT計算、Lib\_ifft2()を@CRO側サンプルアプリケーションを使用して実行します。

**【引数】** in\_rl            FFT計算結果の実数部データ先頭アドレス  
in\_im            FFT計算結果の虚数部データ先頭アドレス  
rdata            IFFT計算後の実数部データ先頭アドレス  
idata            IFFT計算後の虚数部データ先頭アドレス

**【書式】** int AP74X\_ifft2(float \*in\_rl, float in\_im, float \*rdata, float \*idata)

**【戻値】** 0        正常終了

**【参考】** AP74X\_fft2

## 2.6.4. ホスト側サンプルアプリケーション

このアプリケーションは、Microsoft 社製 VisualStudio2005 VB にて作成されています。@CRO側のサンプルアプリケーションと下記の領域をハンドリング領域として予め予約設定して実行されます。このアドレスは任意に設定できますので、仕様にあわせて変更できます。内容についてはアプリケーションプログラムを参照してください。

ハンドリング予約領域)

トップアドレス 0x91000000

データ構造

|       |      |              |
|-------|------|--------------|
| int   | STB; | ストローブ        |
| int   | ACK; | アクノリッジ       |
| int   | FLG; | フラグ          |
| int   | DAT; | データ          |
| float | *ar; | 実数部データ先頭アドレス |
| float | *ai; | 虚数部データ先頭アドレス |
| int   | n;   | FFT点数        |
| int   | dim; | FFT次元        |
| int   | ret; | 戻り値          |



## 【 改 訂 履 歴 】

| 改訂番号 | 改訂日付       | 改訂内容 |
|------|------------|------|
| 初版   | 2010.03.05 | 初版   |
|      |            |      |
|      |            |      |

- ・本製品及び本マニュアルの内容は製品の改良のため予告無しに変更される事がありますのでご了承下さい。
- ・本製品及び本マニュアルの内容は著作権法により保護されています。許可なく複製する事はできません。

ACRO741-00  
ソフトウェア・ユーザズ・マニュアル

中部電機株式会社

〒440-0004 愛知県豊橋市忠興3丁目2-8

TEL <0532>61-9566 FAX <0532>63-1081

URL : <http://www.chubu-el.co.jp>

E-mail : [csg@chubu-el.co.jp](mailto:csg@chubu-el.co.jp)

2010. 3 第1版発行