AVR914 : AT90CAN32/64/128用 CAN/UARTに基くﾌﾞｰﾄﾛｰﾀﾞ

8ビット AVR® マイクロ コントローラ

応用記述

1. 要点

■ UART規約
- 物理層として使用されるUART
- Intel HEX形式記録域に基づく
- 自動ﾎﾞｰﾚｰﾄ

■ CAN規約
- 物理層として使用される制御器域網(CAN)
- 再設定可能な7つのISP CAN識別子
- 自動ﾋﾞｯﾄ速度

■ 実装書き込み
- ﾌﾗｯｼｭ ﾒﾓﾘとEEPROMの読み書き
- ﾃﾞﾊﾞｲｽID読み込み
- 完全なﾁｯﾌﾟ消去
- 形態設定ﾊﾞｲﾄの読み書き
- ISP命令からの保護設定
- 遠隔応用開始命令

2. 説明

この資料はUART/CANﾌﾞｰﾄﾛｰﾀﾞの機能だけでなく、ﾁｯﾌﾟ上のﾌﾗｯｼｭ ﾒﾓﾘとEEPROMでの操作を効率的に実行するための直列規約も記述します。

このﾌﾞｰﾄ ﾛｰﾀﾞは"実装書き込み(ISP:In-System Proguramming)"を実装します。ISPはｼｽﾃﾑからﾃﾞﾊﾞｲｽを取り去ることなく、そして予め書き込まれた応用の必要なく、ﾏｲｸﾛ ｺﾝﾄﾛｰﾗのﾁｯﾌﾟ上のﾌﾗｯｼｭ ﾒﾓﾘとEEPROMの書き込みや書き換えを使用者に許します。

CAN/UARTﾌﾞｰﾄ ﾛｰﾀﾞは直列網または直列(信号)線を通してﾎｽﾄとの通信を管理することができます。それはﾁｯﾌﾟ上のﾌﾗｯｼｭ ﾒﾓﾘとEEPROMでのｱｸｾｽや要求した操作を実行することもできます。

3. 資料管理

ﾌﾞｰﾄﾛｰﾀﾞ改訂	変更の目的	ｺﾝﾊﾟｲﾗ版	日付
改訂1.0.0	初回公開	-	2003年6月16日
改訂1.0.1 (7592A)	FLIP 2.4.4用更新 AT90CAN32/64/128用更新 いくつかのﾊﾞｸﾞ修正	ATMEL AVR用 IAR Embedded Workbench 4.11A	2005年10月19日
改訂1.0.1 (7592B)	CAN ID_PROG_START応答での修正		2006年1月5日

本書は一般の方々の便宜のため有志により作成されたもので、ATMEL社とは無関係であることを御承知ください。しおりのはじめにでの内容にご注意ください。

Rev. 7592B-01/06, 7592BJ1-12/13

4. ブートローダ環境

CAN/UART AT90CAN32/64/128ブートローダはチップ上のフラッシュ メモリの"ブートローダ フラッシュ領域"に格納されます。ブートローダの大きさは8Kバイト近くで、故に物理的な"ブートローダ フラッシュ領域"は完全に使用されます。この領域はブートローダのために予約され、応用プログラムの大きさは"応用フラッシュ領域"と等しいか、またはそれ以下でなければなりません(「**表3-1. デバイス メモリ割付** (バイト アドレス指定)」参照)。

表4-1. デバイス メモリ割付 (バイト アドレス指定)

メモリ種別		AT90CAN128	AT90CAN64	AT90CAN32
フラッシュ メモリ	容量	128Kバイト	64Kバイト	32Kバイト
	アドレス範囲	$00000～$1FFFF	$00000～$0FFFF	$00000～$07FFF
"応用フラッシュ領域"	容量	120Kバイト	56Kバイト	24Kバイト
	アドレス範囲	$00000～$1DFFF	$00000～$0DFFF	$00000～$05FFF
"ブートローダ フラッシュ領域"	容量	8Kバイト		
	アドレス範囲	$1E000～$1FFFF	$0E000～$0FFFF	$06000～$07FFF
"ブート リセット アドレス"		$1E000	$0E000	$06000
EEPROM	容量	4Kバイト	2Kバイト	1Kバイト
	アドレス範囲	$0000～$0FFF	$0000～$07FF	$0000～$03FF

注: ブートローダ リセット アドレス"は"BOOTSZ"ヒューズ ビットに依存します。メモリ(フラッシュ メモリ、EEPROMなど)の動きのより多くの詳細についてはデータシートを参照してください。

4.1. 物理的な環境

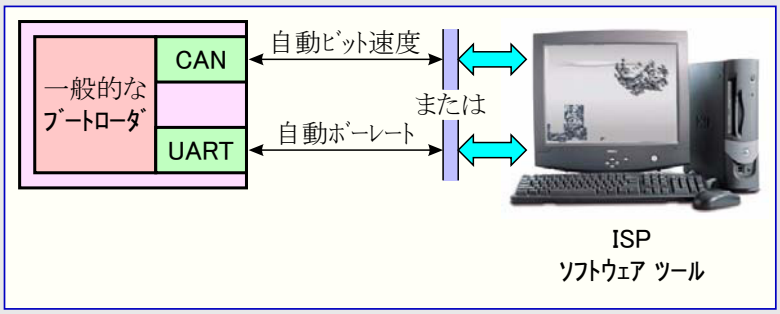
ブートローダは以下を通してホスト(またはPC)とやり取りします。

- CANインターフェース

または

- UARTインターフェース

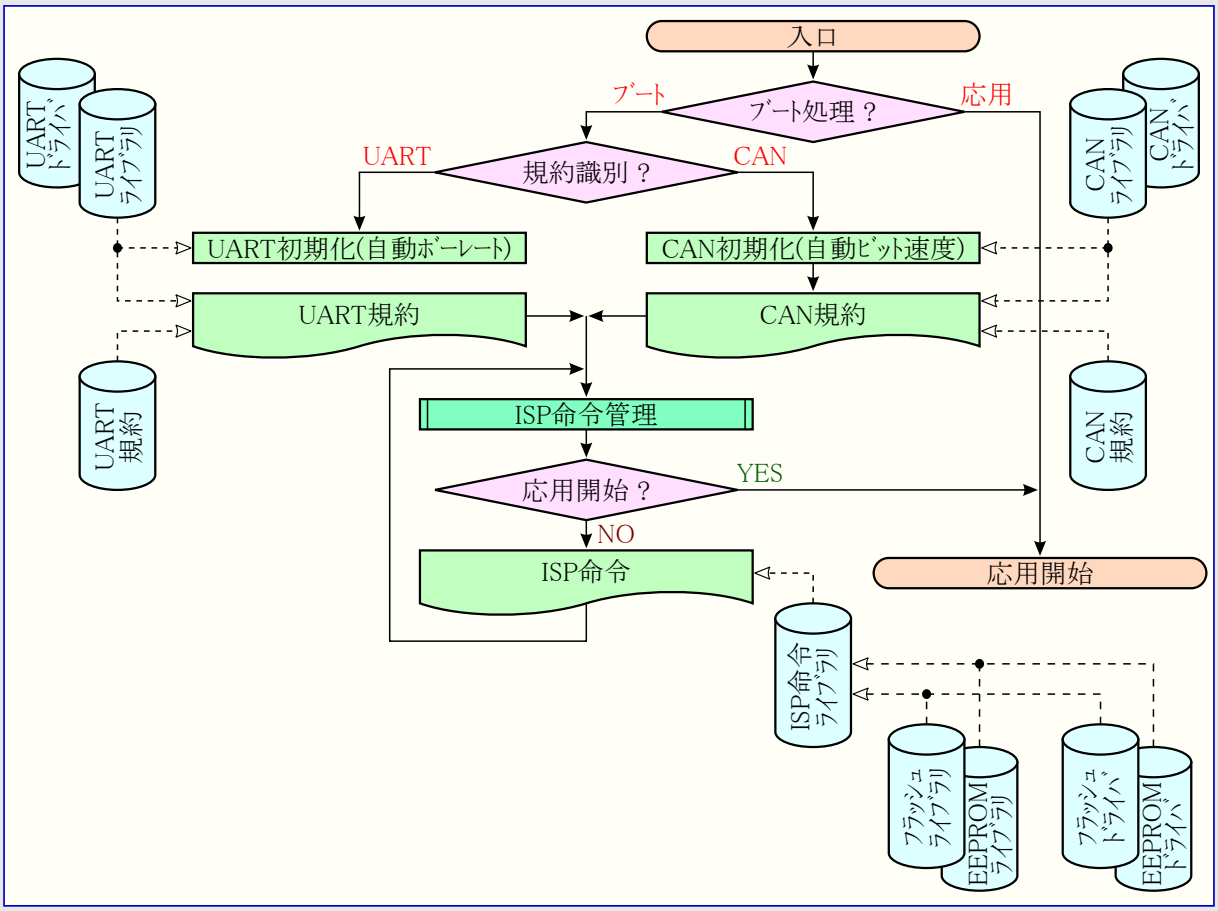
図4-1. 物理的な環境



4.2. ﾌﾞｰﾄﾛｰﾀﾞ説明

4.2.1. 概要

図4-2. ﾌﾞｰﾄﾛｰﾀﾞ構成図



4.2.2. 入口点

1つの"入口点"だけが利用可能で、それはﾌﾞｰﾄ ﾛｰﾀﾞへの入口点です。ﾃﾞﾊﾞｲｽの"BOOTRST"ﾋｭｰｽﾞが設定(0)されなければなりません。ﾘｾｯﾄ後に、ﾃﾞﾊﾞｲｽの"ﾌﾟﾛｸﾞﾗﾑ ｶｳﾝﾀ"は"ﾌﾞｰﾄ ﾘｾｯﾄ ｱﾄﾞﾚｽ"に設定されます(2頁の**表4-1. ﾃﾞﾊﾞｲｽ ﾒﾓﾘ割付**(ﾊﾞｲﾄ ｱﾄﾞﾚｽ指定)参照)。この"入口点"はﾌﾞｰﾄ ﾛｰﾀﾞの"ﾌﾞｰﾄ処理"を初期化します。

4.2.3. ﾌﾞｰﾄ処理

ﾌﾞｰﾄ ﾛｰﾀﾞの"ﾌﾞｰﾄ処理"は応用またはﾌﾞｰﾄ ﾛｰﾀﾞ自身の開始を許します。これは次の2つの変数に依存します。

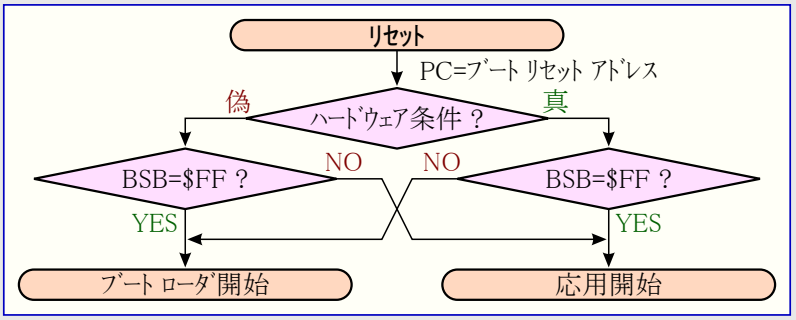
・"ﾊｰﾄﾞｳｪｱ条件"

ﾊｰﾄﾞｳｪｱ条件はﾃﾞﾊﾞｲｽの入力ﾋﾟﾝとそれの活性ﾚﾍﾞﾙ(例:INT0/PIND.0, active low)によって定義されます。これは"config.h"ﾌｧｲﾙで設定されます。

・"ﾌﾞｰﾄ状態ﾊﾞｲﾄ"

ﾌﾞｰﾄ状態ﾊﾞｲﾄ"BSB(Boot Status Byte)"は"ﾌﾞｰﾄ ﾛｰﾀﾞ形態設定ﾒﾓﾘ"に属します(7頁の「**5.4.4.1. ﾌﾞｰﾄ状態ﾊﾞｲﾄ – "BSB"**」項参照)。これの既定値は$FFです。ISP命令はこの値の変更を許します。

図4-3. ﾌﾞｰﾄ処理構成図



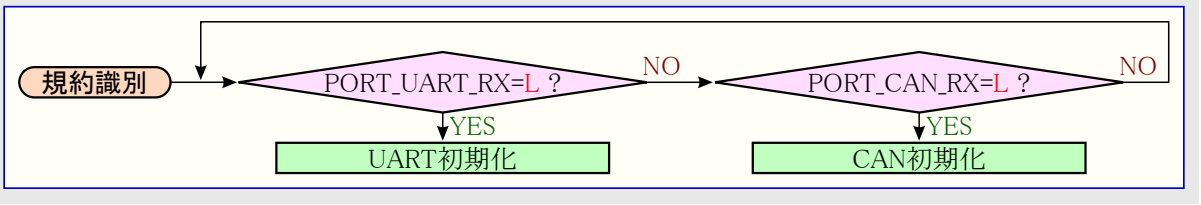
4.2.4. 規約識別

ﾌﾞｰﾄ ﾛｰﾀﾞの"規約識別"は使用される何かの規約、CANまたはUARTの規約を選びます。媒体での活動を検知するために物理(信号)線のﾎﾟｰﾘﾝｸﾞが行われます。それらの線は次の通りです。

- PORT_CAN_RX：RXCAN/PIND.6でﾎﾟｰﾘﾝｸﾞが行われます。
- PORT_UART_RX："config.h"で設定された定義に依存。
  - "USE_UART1"が定義されたなら、ﾎﾟｰﾘﾝｸﾞはRXD0/PINE.0で行われます。
  - "USE_UART2"が定義されたなら、ﾎﾟｰﾘﾝｸﾞはRXD1/PIND.2で行われます。

これらの1つでの最初のLowﾚﾍﾞﾙが対応する周辺機能の初期化を開始します。

図4-4. 規約識別構成図



4.2.5. CAN初期化

ﾎｽﾄとの通信に使用されるCANは以下の形態設定を持ちます。

- **規格**　：CAN形式2.0A (11ﾋﾞｯﾄ識別子)
- **ﾌﾚｰﾑ**　：ﾃﾞｰﾀ ﾌﾚｰﾑ
- **ﾋﾞｯﾄ速度**：特別ﾊﾞｲﾄ(EB:Extra Byte)に依存(8頁の「**特別ﾊﾞｲﾄ - "EB"**」をご覧ください。)
  - "EB"＝$FF：ｿﾌﾄｳｪｱ自動ﾋﾞｯﾄ速度を使用
  - "EB"≠$FF：CANﾋﾞｯﾄ速度設定にﾋﾞｯﾄ ﾀｲﾐﾝｸﾞ制御1～3を使用(8頁の「**ﾋﾞｯﾄ ﾀｲﾐﾝｸﾞ制御1～3 - "BTC1～3"**」をご覧ください。)

初期化処理は各ﾃﾞﾊﾞｲｽ ﾘｾｯﾄ後に実行されなければなりません。ﾎｽﾄは節点(ﾉｰﾄﾞ)を選択するためにﾃﾞｰﾀ ﾌﾚｰﾑを送ることによって通信を始めます。自動ﾋﾞｯﾄ速度の場合、これがCANﾋﾞｯﾄ速度を見つけるため、ﾌﾞｰﾄﾛｰﾀﾞを助けます。CAN規格は異常応答を持つﾌﾚｰﾑは自動的に再送されると言っています。この機能と"聴取"形態に設定されるべきCAN周辺機能の能力が自動ﾋﾞｯﾄ速度によって使用されます。一旦同期ﾌﾚｰﾑがどの異常もなく受信されると、"聴取"形態の開放によって応答で劣性ﾚﾍﾞﾙが印加されます。

ｿﾌﾄｳｪｱ自動ﾋﾞｯﾄ速度はﾃﾞﾊﾞｲｽで設定されたｼｽﾃﾑ ｸﾛｯｸ(CKIO)に従って広範囲のﾎﾞｰﾚｰﾄを支援します("config.h"ﾌｧｲﾙで"FOC"定義参照)。この(自動ﾋﾞｯﾄ速度)機能は多数のCAN節点(ﾉｰﾄﾞ)を持つCAN網で保証されません。

4.2.6. UART初期化

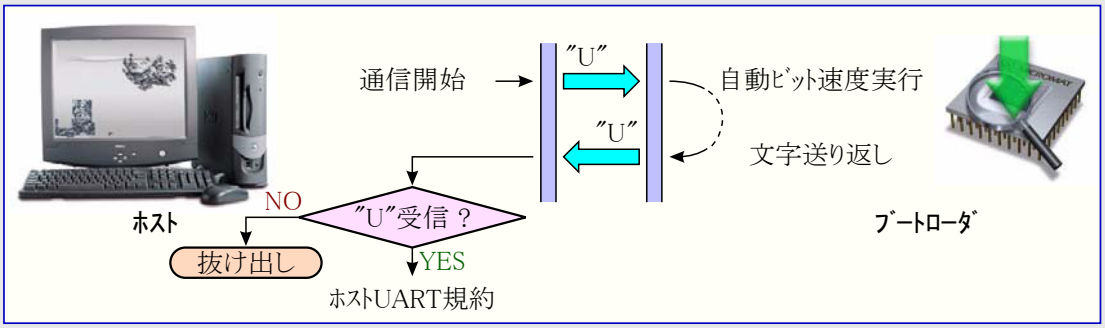
ﾎｽﾄとの通信に使用されるUARTは以下の形態設定を持ちます。

- **文字**　：8ﾋﾞｯﾄのﾃﾞｰﾀ
- **ﾊﾟﾘﾃｨ**　：なし
- **停止ﾋﾞｯﾄ**　：1ﾋﾞｯﾄ
- **流れ制御**　：なし
- **ﾎﾞｰﾚｰﾄ**　：ﾎｽﾄによって選ばれたﾎﾞｰﾚｰﾄを見つけれるために自動ﾎﾞｰﾚｰﾄが実行されます。

初期化処理は各ﾃﾞﾊﾞｲｽ ﾘｾｯﾄ後に実行されなければなりません。ﾎｽﾄはﾎﾞｰﾚｰﾄを見つける(自動ﾎﾞｰﾚｰﾄ)ﾌﾞｰﾄﾛｰﾀﾞを手助けするために同期文字として文字"U"($55)を送ることによって通信を始めます。1つの同期文字だけが送られ、この文字の最後でﾌﾞｰﾄﾛｰﾀﾞは初期化が終わったUARTを持たなければなりません。

ﾌﾞｰﾄﾛｰﾀﾞはﾃﾞﾊﾞｲｽで設定されたｼｽﾃﾑ ｸﾛｯｸ(CKIO)に従って広範囲のﾎﾞｰﾚｰﾄを支援します("config.h"ﾌｧｲﾙで"FOC"定義参照)。

図4-5. UART同期



4.2.7. CAN規約概要

"CAN/UART規約"は直列線上の上位規約です。

これらは本資料の特別な節で記述されます(9頁の「**CAN規約とISP命令**」と13頁の「**UART規約とISP命令**」をご覧ください)。

4.2.8. ISP命令概要

"CAN/UART規約"の各々は"ISP命令"を復号します。"ISP命令"の組は両規約とも明らかに無関係です。

これは本資料の特別な節で記述されます(9頁の「**CAN規約とISP命令**」と13頁の「**UART規約とISP命令**」をご覧ください)。

4.2.9. ﾌﾞｰﾄﾛｰﾀﾞからの出力

ﾌﾞｰﾄﾛｰﾀﾞからの出力はISP命令:"応用開始"を受信した後で実行されます。(9頁の「**CAN規約とISP命令**」と13頁の「**UART規約とISP命令**」をご覧ください)。

5. メモリ空間定義

ﾌﾞｰﾄﾛｰﾀﾞは5つまでの独立したﾒﾓﾘ空間を支援します。それらの各々は低位ｱｸｾｽ規約(ﾄﾞﾗｲﾊﾞ)が異なり得るため、ｺｰﾄﾞ番号(対応する規約領域で報告される値)を受け取ります。

ﾒﾓﾘ空間のｱｸｾｽはﾊﾞｲﾄ ｱｸｾｽ(換言すると、ﾊﾞｲﾄ ｱﾄﾞﾚｽで与えられたｱﾄﾞﾚｽ)です。

表5-1. ﾒﾓﾘ空間ｺｰﾄﾞ番号

空間　(注1)	ｺｰﾄﾞ番号	ｱｸｾｽ
ﾌﾗｯｼｭ ﾒﾓﾘ	0	読み書き
EEPROMﾃﾞｰﾀ ﾒﾓﾘ	1	読み書き
–	2	–
ﾌﾞｰﾄ ﾛｰﾀﾞ情報	3	読み込み専用
ﾌﾞｰﾄ ﾛｰﾀﾞ形態設定	4	読み書き
ﾃﾞﾊﾞｲｽ ﾚｼﾞｽﾀ　(注2)	5	読み込み専用
識票	6	読み込み専用

注1: 時々、識別は物理的ではありません(例:"識票"は"ﾌﾞｰﾄﾛｰﾀﾞ情報"だけでなく、"ﾌﾞｰﾄﾛｰﾀﾞ ﾌﾗｯｼｭ領域"のｺｰﾄﾞの部分でもあります)。

注2: 未だ実装されていません。

5.1. ﾌﾗｯｼｭ ﾒﾓﾘ空間

ﾌﾞｰﾄﾛｰﾀﾞによって管理されるﾌﾗｯｼｭ ﾒﾓﾘ空間はﾃﾞﾊﾞｲｽのﾌﾗｯｼｭ ﾒﾓﾘの部分です。それは"応用ﾌﾗｯｼｭ領域"です。

表5-2. ﾌﾗｯｼｭ ﾒﾓﾘ空間 (ｺｰﾄﾞ番号0)

ﾌﾗｯｼｭ ﾒﾓﾘ空間	AT90CAN128	AT90CAN64	AT90CAN32
容量	120Kﾊﾞｲﾄ	56Kﾊﾞｲﾄ	24Kﾊﾞｲﾄ
ｱﾄﾞﾚｽ範囲	$00000～$1EFFF	$0000～$EFFF	$0000～$6FFF
ﾍﾟｰｼﾞ数　(注)	2	1	1

注: ﾍﾟｰｼﾞ ﾊﾟﾗﾒｰﾀはﾌﾞｰﾄﾛｰﾀﾞとﾃﾞﾊﾞｲｽ自体で異なります。

5.1.1. 読み書き

"ISP読み込み"または"ISP書き込み"命令だけが64Kﾊﾞｲﾄのﾍﾟｰｼﾞ内にﾊﾞｲﾄ ｱﾄﾞﾚｽ指定形態でﾌﾗｯｼｭ ﾒﾓﾘ空間にｱｸｾｽします(**表5-2. ﾌﾗｯｼｭ ﾒﾓﾘ空間** (ｺｰﾄﾞ番号0)参照)。特定のISP命令が異なるﾍﾟｰｼﾞの選択を許します。

読み書き命令が起きる間にｿﾌﾄｳｪｱ保護ﾊﾞｲﾄ"SSB(Software Security Byte)"が設定されている場合、ﾌﾞｰﾄﾛｰﾀﾞは"ﾃﾞﾊﾞｲｽ保護"異常を返します(7頁の「**5.4.4.2. ｿﾌﾄｳｪｱ保護ﾊﾞｲﾄ – "SSB"**」項参照)。

5.1.2. 消去

"ISP消去"命令はﾌﾗｯｼｭ ﾒﾓﾘ空間の完全消去(全ﾊﾞｲﾄ=$FF)です。この操作は例えｿﾌﾄｳｪｱ保護ﾊﾞｲﾄ"SSB"が設定されていても利用可能です。この操作の終わりで、ｿﾌﾄｳｪｱ保護ﾊﾞｲﾄ"SSB"は保護ﾚﾍﾞﾙ0にﾘｾｯﾄされます(7頁の「**5.4.4.2. ｿﾌﾄｳｪｱ保護ﾊﾞｲﾄ – "SSB"**」項)。

5.1.3. 制限

ﾌﾗｯｼｭ ﾒﾓﾘ空間でのISP命令はﾌﾞｰﾄﾛｰﾀﾞに於いて無効です("ﾌﾞｰﾄﾛｰﾀﾞ ﾌﾗｯｼｭ領域"で無効)。

ISP命令に対するﾌﾗｯｼｭ ﾒﾓﾘ空間(ｺｰﾄﾞ番号0)の大きさは「**表5-2. ﾌﾗｯｼｭ ﾒﾓﾘ空間** (ｺｰﾄﾞ番号0)」で与えられます。

5.2. EEPROMﾃﾞｰﾀ ﾒﾓﾘ

ﾌﾞｰﾄﾛｰﾀﾞによって管理されるEEPROMﾃﾞｰﾀ ﾒﾓﾘ空間はﾃﾞﾊﾞｲｽのEEPROMです。

表5-3. EEPROMﾃﾞｰﾀ ﾒﾓﾘ空間 (ｺｰﾄﾞ番号1)

EEPROMﾃﾞｰﾀ ﾒﾓﾘ空間	AT90CAN128	AT90CAN64	AT90CAN32
容量	4Kﾊﾞｲﾄ	2Kﾊﾞｲﾄ	1Kﾊﾞｲﾄ
ｱﾄﾞﾚｽ範囲	$0000～$0FFF	$0000～$07FF	$0000～$03FF
ﾍﾟｰｼﾞ数	(ﾍﾟｰｼﾞなし)		

5.2.1. 読み書き

EEPROMﾃﾞｰﾀ ﾒﾓﾘ空間は不揮発性ﾃﾞｰﾀ ﾒﾓﾘとして使用されます。"ISP読み込み"または"ISP書き込み"命令はこの空間に(ﾍﾟｰｼﾞなしで)ﾊﾞｲﾄ単位でｱｸｾｽします。

読み書き命令が起きる間にｿﾌﾄｳｪｱ保護ﾊﾞｲﾄ"SSB(Software Security Byte)"が設定されている場合、ﾌﾞｰﾄﾛｰﾀﾞは"ﾃﾞﾊﾞｲｽ保護"異常を返します(7頁の「5.4.4.2. ｿﾌﾄｳｪｱ保護ﾊﾞｲﾄ – "SSB"」項参照)。

5.2.2. 消去

"ISP消去"命令はEEPROMﾃﾞｰﾀ ﾒﾓﾘ空間の完全消去(全ﾊﾞｲﾄ=$FF)です。この操作はｿﾌﾄｳｪｱ保護ﾊﾞｲﾄ"SSB"がﾘｾｯﾄされている場合にだけ利用可能です(7頁の「5.4.4.2. ｿﾌﾄｳｪｱ保護ﾊﾞｲﾄ – "SSB"」項)。

5.2.3. 制限

ISP命令に対するEEPROMﾃﾞｰﾀ ﾒﾓﾘ空間(ｺｰﾄﾞ番号1)の大きさは「表5-3. EEPROMﾃﾞｰﾀ ﾒﾓﾘ空間 (ｺｰﾄﾞ番号1)」で与えられます。

5.3. ﾌﾞｰﾄﾛｰﾀﾞ情報

ﾌﾞｰﾄﾛｰﾀﾞによって管理されるﾌﾞｰﾄﾛｰﾀﾞ情報空間はﾌﾞｰﾄﾛｰﾀﾞのｺｰﾄﾞに含まれます。これは"ﾌﾞｰﾄﾛｰﾀﾞ ﾌﾗｯｼｭ領域"内です。

表5-4. ﾌﾞｰﾄﾛｰﾀﾞ情報空間 (ｺｰﾄﾞ番号3)

ﾌﾞｰﾄﾛｰﾀﾞ情報空間		AT90CAN128	AT90CAN64	AT90CAN32
ﾌﾞｰﾄ ﾛｰﾀﾞ改訂	ｱﾄﾞﾚｽ:$00 (読み込み専用)	$01		
ﾌﾞｰﾄ識別1	ｱﾄﾞﾚｽ:$01 (読み込み専用)	$D1		
ﾌﾞｰﾄ識別2	ｱﾄﾞﾚｽ:$02 (読み込み専用)	$D2		
ﾍﾟｰｼﾞ数		(ﾍﾟｰｼﾞなし)		

5.3.1. 読み書き

"ISP読み込み"命令はこの空間に(ﾍﾟｰｼﾞなしで)ﾊﾞｲﾄ単位でｱｸｾｽします。

ｱｸｾｽ保護はこの読み込み専用空間で全く提供されません。

5.3.2. 消去

読み込み専用空間のために当てはまりません。

5.3.3. 制限

ISP命令に対するﾌﾞｰﾄﾛｰﾀﾞ情報空間(ｺｰﾄﾞ番号3)の詳細は「表5-4. ﾌﾞｰﾄﾛｰﾀﾞ情報空間 (ｺｰﾄﾞ番号3)」で与えられます。

5.3.4. ﾌﾞｰﾄﾛｰﾀﾞ情報ﾊﾞｲﾄ説明

5.3.4.1. ﾌﾞｰﾄ改訂

ﾌﾞｰﾄ改訂：読み込み専用ｱﾄﾞﾚｽ=$00、値=$01

5.3.4.2. ﾌﾞｰﾄ識別1,2

ﾌﾞｰﾄ識別1,2：読み込み専用ｱﾄﾞﾚｽ=$01,$02、値=$D1,$D2

5.4. ﾌﾞｰﾄﾛｰﾀﾞ形態設定

ﾌﾞｰﾄﾛｰﾀﾞによって管理されるﾌﾞｰﾄﾛｰﾀﾞ形態設定空間は"ﾌﾞｰﾄﾛｰﾀﾞ ﾌﾗｯｼｭ領域"に含まれます。

表5-5. ﾌﾞｰﾄﾛｰﾀﾞ形態設定空間 (ｺｰﾄﾞ番号4)

ﾌﾞｰﾄﾛｰﾀﾞ形態設定空間			AT90CAN128	AT90CAN64	AT90CAN32
ﾌﾞｰﾄ状態ﾊﾞｲﾄ	"BSB"	ｱﾄﾞﾚｽ:$00	(既定値=$FF)		
ｿﾌﾄｳｪｱ保護ﾊﾞｲﾄ	"SSB"	ｱﾄﾞﾚｽ:$05	(既定値=$FF)		
特別ﾊﾞｲﾄ	"EB"	ｱﾄﾞﾚｽ:$06	(既定値=$FF) (注1)		
ﾋﾞｯﾄ ﾀｲﾐﾝｸﾞ制御1	"BTC1"	ｱﾄﾞﾚｽ:$1C	(既定値=$FF) (注2)		
ﾋﾞｯﾄ ﾀｲﾐﾝｸﾞ制御2	"BTC2"	ｱﾄﾞﾚｽ:$1D	(既定値=$FF) (注2)		
ﾋﾞｯﾄ ﾀｲﾐﾝｸﾞ制御3	"BTC3"	ｱﾄﾞﾚｽ:$1E	(既定値=$FF) (注2)		
節点(ﾉｰﾄﾞ)番号	"NNB"	ｱﾄﾞﾚｽ:$1F	(既定値=$FF) (注3)		
CAN再配置ID区分	"CRIS"	ｱﾄﾞﾚｽ:$20	(既定値=$FF)		
ﾍﾟｰｼﾞ数			(ﾍﾟｰｼﾞなし)		

注1: 効力については8頁の「特別ﾊﾞｲﾄ – "EB"」をご覧ください。
注2: 効力については8頁の「ﾋﾞｯﾄ ﾀｲﾐﾝｸﾞ制御1～3 – "BTC1～3"」をご覧ください。
注3: 効力については8頁の「(CAN)節点(ﾉｰﾄﾞ)番号 – "NNB"」をご覧ください。

5.4.1. 読み書き

"ISP読み込み"命令はこの空間に(ﾍﾟｰｼﾞなしで)ﾊﾞｲﾄ単位でｱｸｾｽします。
ｱｸｾｽ保護はｿﾌﾄｳｪｱ保護ﾊﾞｲﾄでだけ提供されます(「5.4.4.2. ｿﾌﾄｳｪｱ保護ﾊﾞｲﾄ – "SSB"」項参照)。

5.4.2. 消去

"ISP消去"命令はこの空間に対して利用できません。

5.4.3. 制限

ISP命令に対するﾌﾞｰﾄﾛｰﾀﾞ形態設定空間(ｺｰﾄﾞ番号4)の詳細は「表5-5. ﾌﾞｰﾄﾛｰﾀﾞ形態設定空間 (ｺｰﾄﾞ番号4)」で与えられます。

5.4.4. ﾌﾞｰﾄﾛｰﾀﾞ形態設定ﾊﾞｲﾄ説明

5.4.4.1. ﾌﾞｰﾄ状態ﾊﾞｲﾄ – "BSB"

ﾌﾞｰﾄﾛｰﾀﾞのﾌﾞｰﾄ状態ﾊﾞｲﾄは、応用またはﾌﾞｰﾄﾛｰﾀﾞの開始を制御するために、"ﾌﾞｰﾄ処理"(3頁の「4.2.3. ﾌﾞｰﾄ処理」項)で使用されます。ﾊｰﾄﾞｳｪｱ条件が全く設定されない場合、ﾌﾞｰﾄ状態ﾊﾞｲﾄの既定値($FF)はﾌﾞｰﾄﾛｰﾀﾞの開始を強制し、さもなければ(ﾌﾞｰﾄ状態ﾊﾞｲﾄ≠$FF且つﾊｰﾄﾞｳｪｱ条件なし)応用が始まります。

5.4.4.2. ｿﾌﾄｳｪｱ保護ﾊﾞｲﾄ – "SSB"

ﾌﾞｰﾄﾛｰﾀﾞは使用者ｱｸｾｽまたはISPｱｸｾｽから自身と応用を保護するためにｿﾌﾄｳｪｱ保護ﾊﾞｲﾄ"SSB"を持ちます。これはﾌﾗｯｼｭ ﾒﾓﾘとEEPROM空間と自身を保護します。
ｿﾌﾄｳｪｱ保護ﾊﾞｲﾄ"SSB"での"ISP書き込み"命令はより高い優先ﾚﾍﾞﾙだけを書くことができます。3つの保護ﾚﾍﾞﾙがあります。

表5-6. 保護ﾚﾍﾞﾙ

ﾚﾍﾞﾙ	保護	"SSB"	注釈
0	NO_SECURITY	$FF	• これが既定ﾚﾍﾞﾙです。 • ﾚﾍﾞﾙ1またはﾚﾍﾞﾙ2だけがﾚﾍﾞﾙ0を上書きすることができます。
1	WR_SECURITY	$FE	• ﾚﾍﾞﾙ1ではﾌﾗｯｼｭ ﾒﾓﾘとEEPROMの空間で書くことができません。 • ﾌﾞｰﾄﾛｰﾀﾞは異常ﾒｯｾｰｼﾞを返します。 • ﾚﾍﾞﾙ2だけがﾚﾍﾞﾙ1を上書きすることができます。
2	RD_WR_SECURITY	$FC	• ﾌﾗｯｼｭ ﾒﾓﾘとEEPROMの空間での全ての読み書きｱｸｾｽが許されません。 • ﾌﾞｰﾄﾛｰﾀﾞは異常ﾒｯｾｰｼﾞを返します。 • ﾌﾗｯｼｭ ﾒﾓﾘ空間での"ISP消去"命令だけがｿﾌﾄｳｪｱ保護ﾊﾞｲﾄを(ﾚﾍﾞﾙ0に)ﾘｾｯﾄします。

下表はSSBレベルに関して認められた活動を与えます。

表5-7. ソフトウェア保護バイト"SSB"に関して許された活動

ISP命令	NO_SECURITY	WR_SECURITY	RD_WR_SECURITY
フラッシュ メモリ空間消去	許可	許可	許可
EEPROM空間消去	許可	–	–
フラッシュ メモリ空間書き込み	許可	–	–
EEPROM空間書き込み	許可	–	–
フラッシュ メモリ空間読み込み	許可	許可	–
EEPROM空間読み込み	許可	許可	–
("SSB"を除き)ブートローダ形態設定書き込み	許可	–	–
ブートローダ形態設定読み込み	許可	許可	許可
"SSB"書き込み	許可	より高いレベルのみ	–
ブートローダ情報読み込み	許可	許可	許可
識票読み込み	許可	許可	許可
(何れかのメモリの)空白検査	許可	許可	許可
メモリ空間変更	許可	許可	許可

5.4.4.3. 特別バイト – "EB"

特別バイトはCAN初期化を自動ビット速度または固定のCANビット タイミングに切り替えるのに用いられます。

- "EB"＝$FF：ソフトウェア自動ビット速度使用
- "EB"≠$FF：(自動ビット速度ではなく)CAN周辺機能のCANビット タイミング レジスタを設定するのにブートローダ形態設定空間のCANBT1～3バイトを使用

注: 未だ開発されていません。これは将来のブートローダ版で行われるでしょう。

5.4.4.4. ビット タイミング制御1～3 – "BTC1～3"

"EB"≠$FFの時にCAN周辺機能のCANビット タイミング レジスタを設定するのにブートローダ形態設定空間のビット タイミング制御1～3バイト("BTC1","BTC2","BTC3")が使用されます(自動ビット速度ではありません)。

これらのバイトを書く別の方法は9頁の「5.5.4.1. CANBT1～3レジスタ」項で記述されます。

注: 未だ開発されていません。これは将来のブートローダ版で行われるでしょう。

5.4.4.5. (CAN)節点(ノード)番号 – "NNB"

9頁の「CAN規約とISP命令」をご覧ください。

注: 未だ開発されていません。これは将来のブートローダ版で行われるでしょう。

5.4.4.6. CAN再配置ID区分 – "CRIS"

9頁の「CAN規約とISP命令」をご覧ください。

5.5. デバイス レジスタ

ブートローダによって管理されるデバイス レジスタ空間(コード番号5)はデバイスの64個の標準I/Oレジスタと160個の拡張I/Oレジスタです。これらは等価なアセンブリ言語命令によってアクセスされます。

```
LDS     Rxx, REG_ADD
```

ここでREG_ADDは$20(PINA)～$FA(CANMSG)のアドレス範囲です。

5.5.1. 読み書き

"ISP読み込み"命令はこの空間に(ページなしで)バイト単位でアクセスします。

アクセス保護はこの読み込み専用空間で全く提供されません。

5.5.2. 消去

読み込み専用空間のために当てはまりません。

5.5.3. 制限

この空間はビット アドレス指定ではなく、未実装のレジスタは$FFが返ります。

5.5.4. デバイス レジスタ説明

情報については適切なデータシートを参照してください。

5.5.4.1. CANBT1～3ﾚｼﾞｽﾀ

これらが自動ﾋﾞｯﾄ速度を禁止する(EB≠$FF)前に読まれる場合、同時にそれらはﾌﾞｰﾄﾛｰﾀﾞ形態設定空間の"BTC1","BTC2","BTC3"内に複写されます(8頁の「ﾋﾞｯﾄ ﾀｲﾐﾝｸﾞ制御1～3 - "BTC1～3"」をご覧ください)。

注: 未だ開発されていません。これは将来のﾌﾞｰﾄﾛｰﾀﾞ版で行われるでしょう。

5.6. 識票

ﾌﾞｰﾄﾛｰﾀﾞによって管理される識票空間はﾌﾞｰﾄﾛｰﾀﾞのｺｰﾄﾞに含まれます。これは"ﾌﾞｰﾄﾛｰﾀﾞ ﾌﾗｯｼｭ領域"内です。

表5-8. 識票空間 (ｺｰﾄﾞ番号6)

識票空間		AT90CAN128	AT90CAN64	AT90CAN32
製造者符号	ｱﾄﾞﾚｽ:$00 (読み込み専用)	$1E		
系統符号	ｱﾄﾞﾚｽ:$01 (読み込み専用)	$81		
製品名	ｱﾄﾞﾚｽ:$02 (読み込み専用)	$97	$96	$95
製品改訂	ｱﾄﾞﾚｽ:$03 (読み込み専用)	$00		
ﾍﾟｰｼﾞ数		(ﾍﾟｰｼﾞなし)		

5.6.1. 読み書き

"ISP読み込み"命令はこの空間に(ﾍﾟｰｼﾞなしで)ﾊﾞｲﾄ単位でｱｸｾｽします。

ｱｸｾｽ保護はこの読み込み専用空間で全く提供されません。

5.6.2. 消去

読み込み専用空間のために当てはまりません。

5.6.3. 制限

ISP命令に対する識票空間(ｺｰﾄﾞ番号6)の詳細は「表5-8. 識票空間 (ｺｰﾄﾞ番号6)」で与えられます。

6. CAN規約とISP命令

本章はCAN網上のより高位の規約と提携するISP命令の符号化を記述します。

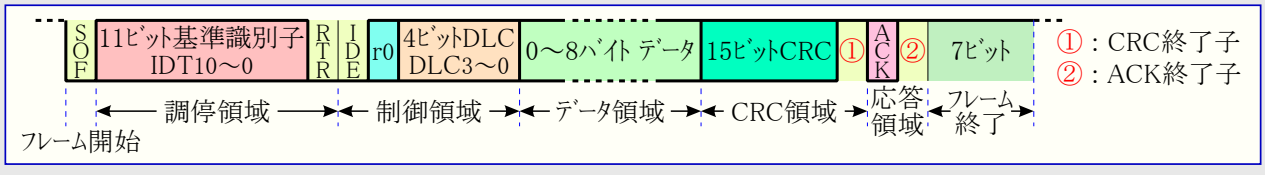
6.1. CANﾌﾚｰﾑ説明

CAN規約は11ﾋﾞｯﾄ識別子を持つCAN 2.0Aとしても知られるCAN標準ﾌﾚｰﾑ(高速用のISO11898と低速用のSIO11519-2参照)を支援します。

"ﾌﾚｰﾑ開始(SOF"で始まるCAN標準ﾌﾚｰﾑ形式内のﾒｯｾｰｼﾞは、ﾃﾞｰﾀ ﾌﾚｰﾑと遠隔ﾌﾚｰﾑと呼ばれるﾃﾞｰﾀ要求ﾌﾚｰﾑとを区別するのに使用される"遠隔送信要求(RTR)"ﾋﾞｯﾄと識別子から成る"調停領域"が後続します。後続する"制御領域"は"識別子拡張(IDE)"ﾋﾞｯﾄと、"ﾃﾞｰﾀ領域"内の後続するﾃﾞｰﾀ ﾊﾞｲﾄ数を示すのに使用される"ﾃﾞｰﾀ長符号(DLC)"を含みます。遠隔ﾌﾚｰﾑではDLCが要求するﾃﾞｰﾀ ﾊﾞｲﾄ数を含みます。後続する"ﾃﾞｰﾀ領域"は8つまでのﾃﾞｰﾀ ﾊﾞｲﾄを保持することができます。ﾌﾚｰﾑの完全性は後続する"巡回冗長検査(CRC)"和によって保証されます。"応答(ACK)領域"はACK間隔とACK終了子で折衷にします。ACK間隔内のﾋﾞｯﾄは劣性ﾋﾞｯﾄとして送られ、この時に正しく受信したﾃﾞｰﾀを持つ受信側によって優性ﾋﾞｯﾄとして上書きされます。

ISP CAN規約はCAN標準ﾃﾞｰﾀ ﾌﾚｰﾑでだけ使用します。

図6-1. CAN標準ﾃﾞｰﾀ ﾌﾚｰﾑ



ISP CAN規約を記述するために、識別子に抽象名が使用されますが、既定値は以下の表現内で与えられます。

表6-1. ISP CAN命令の雛形

識別子 (11ﾋﾞｯﾄ)	長さ (4ﾋﾞｯﾄ)	ﾃﾞｰﾀ[0] ～ ﾃﾞｰﾀ[n-1] (1ﾊﾞｲﾄ)	説明
SYMBLIC_NAME ("CRIS"<<4)+X	n (≦8)	値または意味	命令説明

点対点接続のため、送信CANﾒｯｾｰｼﾞは受信側によってﾊｰﾄﾞｳｪｱ応用が行われるまで繰り返されます。

ﾌﾞｰﾄﾛｰﾀﾞは形態設定が見つけられた場合にだけやって来るCANﾌﾚｰﾑに応答することができます。

この機能は多数のCAN節点(ﾉｰﾄﾞ)を持つ網で保証されません。

6.2. CAN ISP命令ﾃﾞｰﾀ列規約

6.2.1. CAN ISP命令説明

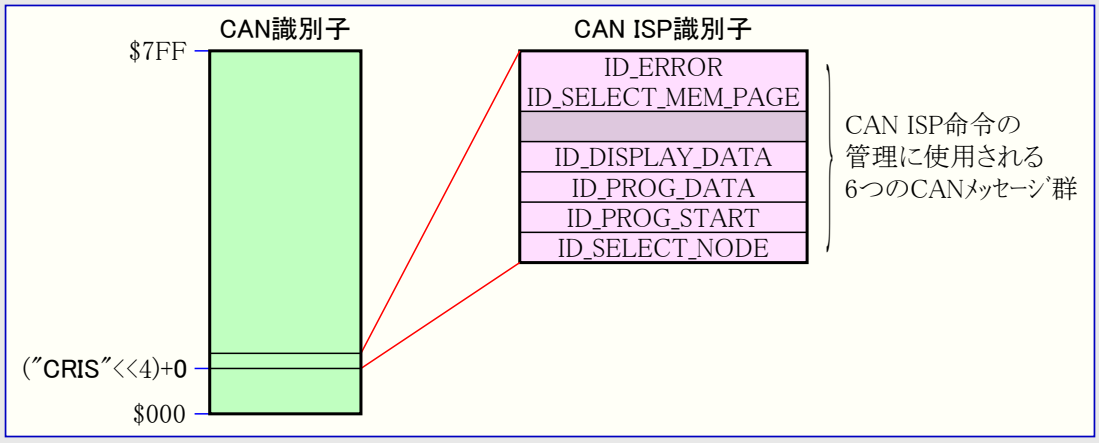
この規約を管理するために多数のCANﾒｯｾｰｼﾞ識別子が定義されています。

表6-2. CAN ISP規約用に定義されたCANﾒｯｾｰｼﾞ識別子

識別子	ISP命令詳細	値
ID_SELECT_NODE	節点(ﾉｰﾄﾞ)との通信路を開く/閉じる	("CRIS"<<4)+0
ID_PROG_START	ﾒﾓﾘ空間のﾌﾟﾛｸﾞﾗﾐﾝｸﾞ開始	("CRIS"<<4)+1
ID_PROG_DATA	ﾒﾓﾘ空間にﾃﾞｰﾀ書き込み	("CRIS"<<4)+2
ID_DISPLAY_DATA	ﾒﾓﾘ空間からﾃﾞｰﾀ読み込み	("CRIS"<<4)+3
ID_START_APPLI	応用開始	("CRIS"<<4)+4
ID_SELECT_MEM_PAGE	ﾒﾓﾘ空間またはﾍﾟｰｼﾞの選択	("CRIS"<<4)+6
ID_ERROR	ﾌﾞｰﾄﾛｰﾀﾞからの異常ﾒｯｾｰｼﾞのみ	

識別子群に関する基礎値を持つ"CRIS"ﾊﾞｲﾄを書くことによってCAN ISP識別子に新しい値を割り当てることが可能です。最大"CRIS"値は$7Fで既定値は$00です。

図6-2. CAN ISP規約に対するCANﾒｯｾｰｼﾞ識別子の再配置

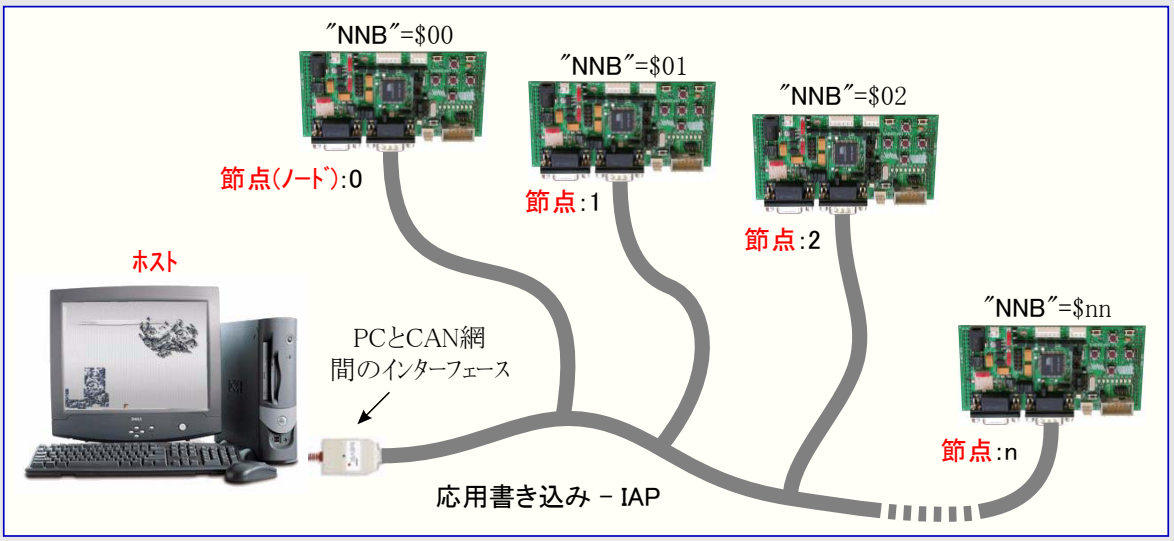


6.2.2. 通信初期化

どんなISP通信を始めるにも先立って装置(CAN節点(ﾉｰﾄﾞ))での通信が開かれなければなりません。装置で通信を開くには、ﾎｽﾄがﾊﾟﾗﾒｰﾀとして渡される節点番号"NNB"を持つ"接続する"CANﾒｯｾｰｼﾞ("ID_SELECT_NODE")を送ります。渡された接点番号が$FFの場合、CANﾌﾞｰﾄﾛｰﾀﾞが通信を受け入れます(図6-3.)。さもなければﾊﾟﾗﾒｰﾀで渡された接点番号は局所"NNB"と等しくなければなりません(図6-4.)。

図6-3. CANﾌﾞｰﾄﾛｰﾀﾞ初回接続

図6-4. CANﾌﾞｰﾄﾛｰﾀﾞ網接続



別の装置で新しい通信を開く前に、現在の装置の通信がそれの接続CANﾒｯｾｰｼﾞ("ID_SELEC_NODE")で閉じられなければなりません。

6.3. CAN ISP命令

6.3.1. CAN節点選択

CAN節点(ﾉｰﾄﾞ)は作業の始めで開かれて終わりで閉じられなければなりません。

6.3.1.1. ﾎｽﾄからのCAN節点選択要求

表6-3. ﾎｽﾄからのCAN節点選択要求

識別子	長さ	ﾃﾞｰﾀ[0]	説明
ID_SELECT_NODE ("CRIS"<<4)+0	1	節点番号("NNB")	指定節点との通信を開くまたは閉じる

6.3.1.2. ﾌﾞｰﾄﾛｰﾀﾞからのCAN節点選択応答

表6-4. ﾌﾞｰﾄﾛｰﾀﾞからのCAN節点選択応答

識別子	長さ	ﾃﾞｰﾀ[0]	ﾃﾞｰﾀ[1]	説明
ID_SELECT_NODE ("CRIS"<<4)+0	2	"ﾌﾞｰﾄﾛｰﾀﾞ改訂"	$00	通信を閉じました。
			$01	通信を開きました。

6.3.2. ﾒﾓﾘ/ﾍﾟｰｼﾞ変更

ﾒﾓﾘ空間と/またはﾍﾟｰｼﾞの変更は1つの命令だけがあり、切り替えはCANﾌﾚｰﾑの"ﾃﾞｰﾀ[0]"によって行われます。

6.3.2.1. ﾎｽﾄからのﾒﾓﾘ/ﾍﾟｰｼﾞ変更要求

表6-5. ﾎｽﾄからのﾒﾓﾘ/ﾍﾟｰｼﾞ変更要求

識別子	長さ	ﾃﾞｰﾀ[0]	ﾃﾞｰﾀ[1]	ﾃﾞｰﾀ[2]	説明
ID_SELECT_MEM_PAGE ("CRIS"<<4)+6	3	$00	ﾒﾓﾘ空間	ﾍﾟｰｼﾞ	活動なし
		$01			ﾒﾓﾘ空間選択
		$02			ﾍﾟｰｼﾞ選択
		$03			ﾒﾓﾘ空間とﾍﾟｰｼﾞを選択

6.3.2.2. ﾌﾞｰﾄﾛｰﾀﾞからのﾒﾓﾘ/ﾍﾟｰｼﾞ変更応答

表6-6. ﾌﾞｰﾄﾛｰﾀﾞからのﾒﾓﾘ/ﾍﾟｰｼﾞ変更応答

識別子	長さ	ﾃﾞｰﾀ[0]	説明
ID_SELECT_MEM_PAGE ("CRIS"<<4)+6	1	$00	選択OK (例え要求ﾌﾚｰﾑで"ﾃﾞｰﾀ[0]"=$00でも)

6.3.3. メモリの読み込み/空白検査

これらの操作は通信に於いて直前に開いた装置でだけ実行されます。この命令は直前に定義されたメモリ空間とページで利用可能です。

読み込みまたは空白検査を開始するには、ホストが"データ[0]"で望む操作、パラメータとして開始アドレスと終了アドレスを持つCANメッセージ("ID_DISPLAY_DATA")を送ります。

6.3.3.1. ホストからのメモリの読み込み/空白検査要求

表6-7. ホストからのメモリの読み込み/空白検査要求

識別子	長さ	データ[0]	データ[1]	データ[2]	データ[3]	データ[4]	説明
ID_DISPLAY_DATA ("CRIS"<<4)+3	5	$00	開始アドレス (上位,下位)		終了アドレス (上位,下位)		選択したメモリ/ページのデータ表示
		$80					選択したメモリ/ページの空白検査

6.3.3.2. ブートローダからの読み込み/空白検査応答

表6-8. ブートローダからのメモリの読み込み/空白検査応答

識別子	長さ	データ[0]	データ[1]	～	データ[7]	説明
ID_DISPLAY_DATA ("CRIS"<<4)+3	最大8	8バイトまでのデータ バイト				データ読み込み
	0	–	–	–	–	空白検査OK
	2	非空白の最初のアドレス		–	–	空白検査で誤り
ID_ERROR ("CRIS"<<4)+6	1	$00	–	–	–	ソフトウェア保護設定異常("データ表示"のみ)

6.3.4. メモリの書き込み/消去

これらの操作は通信に於いて直前に開いた装置でだけ実行されます。これらは次の2つの段階が必要です。

- 最初の段階は書き込みまたは消去の命令に関するアドレス範囲を指示することです。
- 2つ目の段階は書き込みだけに関してデータを送ることです。

書き込み操作を開始するには、ホストが"データ[0]"で望む操作、パラメータとして開始アドレスと終了アドレスを持つ"書き込み開始"CANメッセージ("ID_PROG_START")を送ります。

6.3.4.1. ホストからのメモリの書き込み/消去要求

表6-9. ホストからのメモリの書き込み/消去要求

識別子	長さ	データ[0]	データ[1]	データ[2]	データ[3]	データ[4]	データ[5~7]	説明
ID_PROG_START ("CRIS"<<4)+1	5	$00	開始アドレス (上位,下位)		終了アドレス (上位,下位)		–	選択したメモリ空間/ページ 書き込み開始
	3	$80	$FF	$FF	–	–	–	選択したメモリ空間/ページ消去
ID_PROG_DATA ("CRIS"<<4)+2	最大8	8バイトまでのデータ バイト						書き込むデータ

6.3.4.2. ブートローダからのメモリの書き込み/消去応答

表6-10. ブートローダからのメモリの書き込み/消去応答

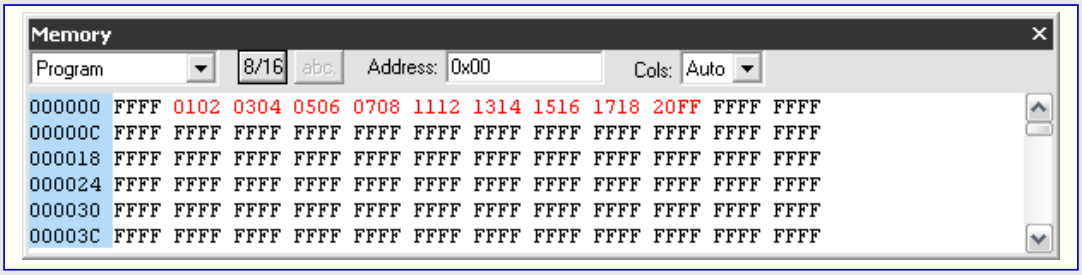
識別子	長さ	データ[0]	説明
ID_PROG_START ("CRIS"<<4)+1	0	–	書き込み初期化、命令OK
	1	$00	消去終了
ID_PROG_DATA ("CRIS"<<4)+2	1	$00	命令OK、転送終了
		$02	命令OK、けれども新しい(他の)データを期待
ID_ERROR ("CRIS"<<4)+6	1	$00	異常 – ソフトウェア保護設定 ("書き込み開始"のみ)

6.3.4.3. メモリ書き込み例

表6-11. メモリ書き込み例

要求/応用	CANメッセージ (16進)			説明
	識別子	長さ	データ[0~7]	
要求(→)	000	1	FF	CAN節点(ノード)選択
応答(←)	000	2	01 01	通信を開く
既定メモリ空間=フラッシュメモリ、既定ページ=0ページ				
要求(→)	001	5	00 00 02 00 12	$0002～$0012番地に書き込み開始
応答(←)	001	0	-	命令OK
要求(→)	002	8	01 02 03 04 05 06 07 08	第1データ列送信
応答(←)	002	1	02	命令OK、新しいデータを期待
要求(→)	002	8	11 12 13 14 15 16 17 18	第2データ列送信
応答(←)	002	1	02	命令OK、新しいデータを期待
要求(→)	002	1	20	第3データ列送信
応答(←)	002	1	00	命令OK、転送終了

図6-5. 直前のメモリ書き込み例の結果



注: AVR Studio®のプログラム用メモリ表示

6.3.5. 応用開始

この操作は通信で直前に開いた装置でだけ実行することができます。

応用を開始するには、ホストが"データ[1]"で選択する"方法"を持つ応用開始CANメッセージを送ります。応用はウォッチドッグ リセットまたはフラッシュ メモリ内のアドレス$0000に飛ぶことによって開始することができます。

ブートローダによって応答は全く返されません。

表6-12. ホストからの応用開始要求

識別子	長さ	データ[0]	データ[1]	データ[2]	データ[3]	説明
ID_START_APPLI ("CRIS"<<4)+4	2	$03	$00	-	-	ウォッチドッグ リセットで応用開始
	4		$01	$00	$00	アドレス$0000へ飛ぶ

7. UART規約とISP命令

本章はUART直列(信号)線上のより高位の規約と提携するISP命令の符号化を記述します。

7.1. UARTフレーム説明

UART規約はIntel拡張HEX形式記録に基づきます。

各記録域はコロン(':')文字のASCII符号の$3Aを含む記録印(RECORD MARK)領域で始まります。各記録域はデータまたは情報のバイト数を指定する記録長(RECORD LENGTH)領域、そして記録域の記録形式(RECORD TYPE)が後続します。1つのデータ バイトが2つのASCII文字によって表現されることに注意してください。記録長(RECORD LENGTH)領域の最大の値は$FFまたは255です。

表7-1. Intel HEX形式フレーム

記録印 (':') (RECORD MARK)	記録長 (RECORD LENGTH)	変位 (OFFSET)	記録形式 (RECORD TYPE)	ﾃﾞｰﾀ/情報 (DATA/INFORMATION)	ﾁｪｯｸｻﾑ (CHECKSUM)
1文字(1ﾊﾞｲﾄ)	1ﾊﾞｲﾄ	2ﾊﾞｲﾄ	1ﾊﾞｲﾄ	nﾊﾞｲﾄ	1ﾊﾞｲﾄ

例: :10E24C00121729FF413950BD0DBCF3395FCCF239504

記録印 記録長 変位 記録形式 ﾃﾞｰﾀ/情報 ﾁｪｯｸｻﾑ

- **記録印**(RECORD MARK) (1ASCIIﾊﾞｲﾄ)

この領域はASCIIのｺﾛﾝ(':')文字の16進符号化$3Aを含みます。

- **記録長**(RECORD LENGTH) (ASCIIを復号したならば1ﾊﾞｲﾄ)

この領域は記録形式(RECORD TYPE)領域に後続するﾃﾞｰﾀ/情報(DATA/INFORMATION)のﾊﾞｲﾄ数を指定します。

- **変位**(OFFSET) (ASCIIを復号したならば2ﾊﾞｲﾄ)

各記録域はﾃﾞｰﾀ ﾊﾞｲﾄの16ﾋﾞｯﾄの設定開始変位を指定する変位(OFFSET)領域を持ち、従ってこの領域はﾃﾞｰﾀ記録域にだけ使用されます。

- **記録形式**(RECORD TYPE) (ASCIIを復号したならば1ﾊﾞｲﾄ)

各記録域はその記録域の記録形式を指定する記録形式(RECORD TYPE)領域を持ちます。記録形式(RECORD TYPE)領域は記録域内の残りの情報を解釈するのに使用されます。

- **ﾃﾞｰﾀ/情報**(DATA/INFORMATION) (ASCIIを復号したならばnﾊﾞｲﾄ)

各記録域は可変(記録長(RECORD LENGTH))長のﾃﾞｰﾀ/情報(DATA/INFORMATION)領域持ちます。これは16進桁の対として符号化される0またはより多くのﾊﾞｲﾄから成ります。ﾃﾞｰﾀの意味は記録形式(RECORD TYPE)に依存します。

- **ﾁｪｯｸｻﾑ**(CHECKSUM) (ASCIIを復号したならば1ﾊﾞｲﾄ)

この領域は記録長(RECORD LENGTH)、変位(OFFSET)、記録形式(RECORD TYPE)、ﾃﾞｰﾀ/情報(DATA/INFORMATION)領域のASCII復号でのﾁｪｯｸｻﾑ(2の補数)を含みます。

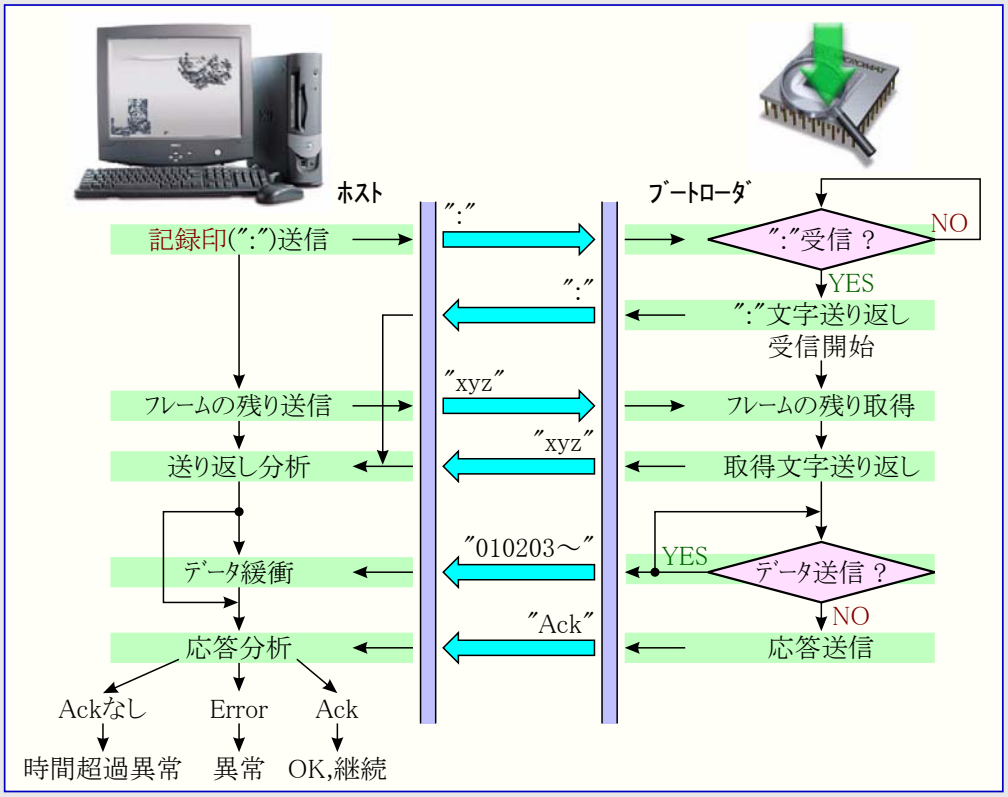
7.2. UART ISP命令ﾃﾞｰﾀ列規約

全てのISP命令は同じ流れを用いて送られます。ﾎｽﾄによって送られた各ﾌﾚｰﾑは最初にﾌﾞｰﾄﾛｰﾀﾞによって送り返されなければなりません。

各命令の流れは以下で終わり得ます。

- "X" : ﾁｪｯｸｻﾑ誤りの場合
- "L" : 読み込み保護設定の場合
- "P" : 書き込み保護設定の場合
- "." : 命令OK(または読み込みﾊﾞｲﾄOKの場合はﾊﾞｲﾄ+".")の場合

図7-1. 命令の流れ概要



7.3. UART ISP命令

7.3.1. ﾒﾓﾘ/ﾍﾟｰｼﾞ変更

ﾒﾓﾘ空間と/またはﾍﾟｰｼﾞを変更するために2つの命令があります。

- **ﾒﾓﾘ選択**　　：ﾒﾓﾘ空間とこのﾒﾓﾘ空間のﾍﾟｰｼﾞを選びます。
- **新ﾍﾟｰｼﾞ選択**：既に選択されているﾒﾓﾘ空間でﾍﾟｰｼﾞを選びます。

7.3.1.1. ﾎｽﾄからのﾒﾓﾘ/ﾍﾟｰｼﾞ変更要求

表7-2. ﾎｽﾄからのﾒﾓﾘ/ﾍﾟｰｼﾞ変更要求

ISP命令要求 (R)		記録長	変位	記録形式	ﾃﾞｰﾀ[0]	ﾃﾞｰﾀ[1]
新ﾍﾟｰｼﾞ選択	":"	$02	開始ｱﾄﾞﾚｽ	$02	上位ﾆﾌﾞﾙ(ﾋﾞｯﾄ7～3)=ﾍﾟｰｼﾞ,下位ﾆﾌﾞﾙ(ﾋﾞｯﾄ3～0)=$0	$00
ﾒﾓﾘ選択	":"	$02	$0000	$04	ﾒﾓﾘ空間符号番号	ﾍﾟｰｼﾞ

7.3.1.2. ﾌﾞｰﾄﾛｰﾀﾞからのﾒﾓﾘ/ﾍﾟｰｼﾞ変更応答

表7-3. ﾌﾞｰﾄﾛｰﾀﾞからのﾒﾓﾘ/ﾍﾟｰｼﾞ変更応答

応答 (A)	文字[0]	文字[1]	文字[2]
命令終了	"."	CR ($0D)	LF ($0A)
不正ﾁｪｯｸｻﾑ	"X"	CR ($0D)	LF ($0A)

7.3.1.3. ﾒﾓﾘ/ﾍﾟｰｼﾞ変更例

表7-4. ﾒﾓﾘ/ﾍﾟｰｼﾞ変更例

ISP命令	要求/応用	ﾌﾚｰﾑ	注釈
ﾒﾓﾘ選択	要求(→)	":020000040100F9"	EEPROM,ﾍﾟｰｼﾞ0選択
	応答(←)	":020000040100F9"+".",CR,LF	命令終了
新ﾍﾟｰｼﾞ選択	要求(→)	"02234502100084"	ﾍﾟｰｼﾞ1,ｱﾄﾞﾚｽ$2345選択 (注)
	応答(←)	"02234502100084"+".",CR,LF	命令終了
ﾒﾓﾘ選択	要求(→)	":20000040001F8"	ﾌﾗｯｼｭ ﾒﾓﾘ,ﾍﾟｰｼﾞ1選択
	応答(←)	":20000040000F8"+"X",CR,LF	ﾁｪｯｸｻﾑ異常

注: ﾍﾟｰｼﾞ容量が64Kﾊﾞｲﾄなので、物理ｱﾄﾞﾚｽは$012345です。

7.3.2. ﾒﾓﾘの読み込み/空白検査

- "ISP **ﾒﾓﾘ読み込み**"命令はﾒﾓﾘ空間のｱﾄﾞﾚｽ範囲の読み込みを許します。
- "ISP **ﾒﾓﾘ空白検査**"命令はﾒﾓﾘ空間のｱﾄﾞﾚｽ範囲の空白検査を許します。

2つの命令は直前に定義されたﾒﾓﾘ空間とﾍﾟｰｼﾞで利用可能です。

7.3.2.1. ﾎｽﾄからのﾒﾓﾘの読み込み/空白検査要求

表7-5. ﾎｽﾄからのﾒﾓﾘの読み込み/空白検査要求

ISP命令要求 (R)		記録長	変位	記録形式	ﾃﾞｰﾀ[0,1] (2ﾊﾞｲﾄ)	ﾃﾞｰﾀ[2,3] (2ﾊﾞｲﾄ)	ﾃﾞｰﾀ[4]
ﾒﾓﾘ読み込み	":"	$05	$0000	$04	開始ｱﾄﾞﾚｽ	終了ｱﾄﾞﾚｽ	$00
ﾒﾓﾘ空白検査	":"	$05	$0000	$04	開始ｱﾄﾞﾚｽ	終了ｱﾄﾞﾚｽ	$01

7.3.2.2. ﾌﾞｰﾄﾛｰﾀﾞからのﾒﾓﾘの読み込み/空白検査応答

表7-6. ﾌﾞｰﾄﾛｰﾀﾞからのﾒﾓﾘの読み込み/空白検査応答

応答 (A)	文字[0～n]			
ﾒﾓﾘ読み込み命令終了	ｱﾄﾞﾚｽ=(16ﾊﾞｲﾄ構成にされた)ﾃﾞｰﾀ,CR,LF			
応答 (A)	**文字[0]**	**文字[1]**	**文字[2]**	**文字[3]**
ﾒﾓﾘ空白検査OK	"."	CR ($0D)	LF ($0A)	-
ﾒﾓﾘ空白検査異常	最初の誤っているｱﾄﾞﾚｽ		CR ($0D)	LF ($0A)
不正ﾁｪｯｸｻﾑ	"X"	CR ($0D)	LF ($0A)	-
異常 - ｿﾌﾄｳｪｱ保護設定 ("ISPﾒﾓﾘ読み込み"のみ)	"L"	CR ($0D)	LF ($0A)	-

7.3.2.3. メモリの読み込み/空白検査例

表7-7. メモリの読み込み/空白検査例

ISP命令	要求/応用	フレーム	注釈
メモリ読み込み	要求(→)	":05000004000300150DF"	選択したメモリとページでアドレス$0003～$0015を読み込み
	応答(←)	":05000004000300150DF",CR,LF "0003=030405～0F101112",CR,LF "0013=131415",CR,LF	データ読み込みと命令終了
メモリ空白検査	要求(→)	":05000004000001001F5"	選択したメモリとページでアドレス$0000～$0100を空白検査
	応答(←)	":05000004000001001F5"+".",CR,LF	メモリ空白検査OK
メモリ読み込み	要求(→)	":05000004001000110D6"	選択したメモリとページでアドレス$0010～$0011を読み込み
	応答(←)	":05000004001000110D6"+"L",CR,LF	保護設定、読み込み中止
メモリ空白検査	要求(→)	":05000004600060020134"	選択したメモリとページでアドレス$6000～$6002を空白検査
	応答(←)	":05000004600060020134"+"6000",CR,LF	アドレス$6000で空白検査失敗
メモリ空白検査	要求(→)	":0500000400005FFF0123"	選択したメモリとページでアドレス$0000～$5FFFを空白検査
	応答(←)	":0500000400005FFF0122"+"X",CR,LF	チェックサム異常

7.3.3. メモリの書き込み/消去

- "ISP **メモリ書き込み**"命令はメモリ空間のアドレス範囲の書き込みを許します。この命令は直前に定義されたメモリ空間とページで利用可能です。
- "ISP **メモリ消去**"命令はメモリ空間の(完全な)消去を許します。この命令は直前に定義されたメモリ空間とページで利用可能です。

7.3.3.1. ホストからのメモリの書き込み/消去要求

表7-8. ホストからのメモリの書き込み/消去要求

ISP命令要求 (R)		記録長	変位	記録形式	データ領域
メモリ書き込み	":"	n	先頭アドレス	$00	nバイト分のデータ
メモリ消去	":"	$05	$0000	$04	$00, $FF, $00, $00, $02

7.3.3.2. ブートローダからのメモリの書き込み/消去応答

表7-9. ブートローダからのメモリの書き込み/消去応答

応答 (A)	文字[0]	文字[1]	文字[2]
命令終了	"."	CR ($0D)	LF ($0A)
不正チェックサム	"X"	CR ($0D)	LF ($0A)
異常 – ソフトウェア保護設定("ISPメモリ書き込み"のみ)	"P"	CR ($0D)	LF ($0A)

7.3.3.3. メモリの書き込み/消去例

表7-10. メモリの書き込み/消去例

ISP命令	要求/応用	フレーム	注釈
メモリ書き込み	要求(→)	":020000001234B8"	選択したメモリとページでアドレス$0000=$12と$0001=$34を書き込み
	応答(←)	":020000001234B8"+".",CR,LF	命令終了
メモリ消去	要求(→)	":0500000400FF000002F6"	選択したメモリを消去
	応答(←)	":0500000400FF000002F6"+".",CR,LF	命令終了
メモリ書き込み	要求(→)	":02000200567821"	選択したメモリとページでアドレス$0002=$56と$0003=$78を書き込み
	応答(←)	":02000200567821"+"P",CR,LF	保護設定、書き込み中止
メモリ消去	要求(→)	":0500000400FF000002F0"	選択したメモリを消去
	応答(←)	":0500000400FF000002F0"+"X",CR,LF	チェックサム異常

7.3.4. 応用開始

ホストはフラッシュ メモリのアドレス$0000へ飛ぶことを発生する応用開始メッセージを送ります。

応答はブートローダによって全く返されません。

表7-11. ホストからの応用開始要求

ISP命令要求 (R)		記録長	変位	記録形式	データ領域
応用開始	":"	$00	$0000	$01	データなし

8. 追補A : "config.h"ファイルの#define

8.1. プロセッサ定義

```
// 全体
#define AVR
#define AT90CAN128  1
#define AT90CAN64   2
#define AT90CAN32   3

// ハードウェア条件 (ブートまたは応用の開始用)
  // DVK90CAN1基板でのINT=INT0またはPD.0 - プルアップでのLow活性
    #define PIN_HWCB      PIND_Bit0
    #define PORT_HWCB     PORTD_Bit0
    #define LEVEL_HWCB    0             // "0"または"1"で活性
    #define PULLUP_HWCB   1             // プルアップ "ON"="1", "OFF"="0"
/*  // DVK90CAN1基板の中央キー= PE.2 プルアップでのLow活性
    #define PIN_HWCB      PINE_Bit2
    #define PORT_HWCB     PORTE_Bit2
    #define LEVEL_HWCB    0             // "0"または"1"で活性
    #define PULLUP_HWCB   1             // プルアップ "ON"="1", "OFF"="0" */
// 応用
#define USE_DEVICE    AT90CAN128
#define USE_UART1
#define FOSC          8000

// 特定定義用切り替え
#ifndef USE_DEVICE
#       error 最初に"config.h"ファイルでUSE_DEVICE AT90CAN128,AT90CAN64,AT90CAN32を定義しなければなりません。
#   elif USE_DEVICE == AT90CAN128
#       define MANUF_ID          0x1E        // 製造者(ATMEL)
#       define FAMILY_CODE       0x97        // 128Kバイトのフラッシュ メモリ
#       define PRODUCT_NAME      0x81        // AT90CAN系列
#       define PRODUCT_REV       0x00        // 改訂0
#       define FLASH_SIZE        0x1FFFF     // バイトでのフラッシュ メモリ量
#       define FLASH_PAGE_SIZE   0x100       // バイトでのフラッシュ ページ容量
#       define BOOT_SIZE         0x2000      // バイトでのブート領域量
#       define EEPROM_SIZE       0x1000      // バイトでのEEPROM量
#   elif USE_DEVICE == AT90CAN64
#       define MANUF_ID          0x1E        // 製造者(ATMEL)
#       define FAMILY_CODE       0x96        // 64Kバイトのフラッシュ メモリ
#       define PRODUCT_NAME      0x81        // AT90CAN系列
#       define PRODUCT_REV       0x00        // 改訂0
#       define FLASH_SIZE        0x0FFFF     // バイトでのフラッシュ メモリ量
#       define FLASH_PAGE_SIZE   0x100       // バイトでのフラッシュ ページ容量
#       define BOOT_SIZE         0x2000      // バイトでのブート領域量
#       define EEPROM_SIZE       0x0800      // バイトでのEEPROM量
#   elif USE_DEVICE == AT90CAN32
#       define MANUF_ID          0x1E        // 製造者(ATMEL)
#       define FAMILY_CODE       0x95        // 32Kバイトのフラッシュ メモリ
#       define PRODUCT_NAME      0x81        // AT90CAN系列
#       define PRODUCT_REV       0x00        // 改訂0
#       define FLASH_SIZE        0x07FFF     // バイトでのフラッシュ メモリ量
#       define FLASH_PAGE_SIZE   0x100       // バイトでのフラッシュ ページ容量
#       define BOOT_SIZE         0x2000      // バイトでのブート領域量
#       define EEPROM_SIZE       0x0400      // バイトでのEEPROM量
```

```
#   else
#       error "config.h"ファイルでUSE_DEVICE定義が参照されていません。
#endif

#ifndef USE_UART1
#       ifndef USE_UART2
#               error "config.h"ファイルでUSE_UART1かUSE_UART2のどちらかを定義しなければなりません。
#       endif
#endif

// ポーリングピン定義
#ifdef USE_UART1
#   define PIN_UART_RX     PINE_Bit0       // UART0用
#   define PORT_UART_TX    PORTE_Bit1      // UART0用
#endif
#ifdef USE_UART2
#   define PIN_UART_RX     PIND_Bit2       // UART1用
#   define PORT_UART_TX    PORTD_Bit3      // UART1用
#endif

#define PIN_CAN_RX          PIND_Bit6
#define PORT_CAN_TX         PORTD_Bit5
```

8.2. UART定義

```
//--------------- UARTライブラリ定義 ---------------
#define UART_AUTOBAUD_EXTERNAL_DETECTION
#define UART_MINIMUM
#define BDR_GENERATOR BRG_TIMER1
#define BAUDRATE      AUTOBAUD
//#define BAUDRATE    19200
#define test_hit()    uart_test_hit()
#define _getkey()     uart_getchar()
#define putchar       uart_putchar
```

8.3. ブートローダ定義

```
//--------------- ブートローダ形態設定 -------------
// UART規約
#define PROTOCOL_DATA                       64
#define GLOBAL_BUFFER_SIZE                  PROTOCOL_DATA+4
#define NB_BYTE_MAX_FOR_DISPLAY_COMMAND     64
#define HEX_SIZE_DISP_PAGE                  16

#define USE_RCS_HEX_PROTOCOL
#define USE_RCS_CAN_PROTOCOL

//------------ ブートローダ識別子定義 ----
#define BOOT_VERSION        0x01    // @00 // Ver 01: JT-18.10.05
#define BOOT_ID1            0xD1    // @01
#define BOOT_ID2            0xD2    // @02

#define MAX_OFFSET_ID       0x7F0

#define NO_SECURITY         0xFF
#define RD_WR_SECURITY      0xFC
#define BSB_DEFAULT         0xFF
#define SSB_DEFAULT         0xFF
#define EB_DEFAULT          0xFF
#define NNB_DEFAULT         0xFF
#define CRIS_DEFAULT        0xFF    // if (offset_id_copy>MAX_OFFSET_ID) offset_id_copy=0;
#define BTC1_DEFAULT        0xFF
#define BTC2_DEFAULT        0xFF
#define BTC3_DEFAULT        0xFF

#define SSB_RD_PROTECTION   0xFC
#define SSB_WR_PROTECTION   0xFE
```

8.4. ﾒﾓﾘ定義

```
//-------- ﾒﾓﾘ定義 ------------------
#define MEM_USER        0
#define MEM_CODE        0
#define MEM_FLASH       0
#define MEM_EEPROM      1
#define MEM_CUSTOM      2
#define MEM_BOOT        3     // ﾌﾞｰﾄ情報
#define MEM_XAF         4     // ﾌﾞｰﾄ形態設定
#define MEM_HW_REG      5
#define MEM_SIGNATURE   6
#define MEM_DEFAULT     MEM_FLASH
#define PAGE_DEFAULT    0x00
```

9. 追補B：CAN規約要約

表9-1. CAN規約要約 – ﾎｽﾄからの要求

ISP命令要求識別子	長さ	ﾃﾞｰﾀ [0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	説明
ID_SELECT_NODE ("CRIS"<<4)+0	1	節点	–	–	–	–	–	–	–	通信を開くまたは閉じる
ID_PROG_START ("CRIS"<<4)+1	5	$00	開始ｱﾄﾞﾚｽ (上位,下位)		終了ｱﾄﾞﾚｽ (上位,下位)		–	–	–	書き込みの初期化
	3	$80	$FF	$FF	–	–	–	–	–	消去
ID_PROG_DATA ("CRIS"<<4)+2	最大8	8ﾊﾞｲﾄまでのﾃﾞｰﾀ ﾊﾞｲﾄ								書き込むﾃﾞｰﾀ
ID_DISPLAY_DATA ("CRIS"<<4)+3	5	$00	開始ｱﾄﾞﾚｽ (上位,下位)		終了ｱﾄﾞﾚｽ (上位,下位)		–	–	–	ﾃﾞｰﾀ表示(読み込み)
		$80								空白検査
ID_START_APPLI ("CRIS"<<4)+4	2	$03	$00	–	–	–	–	–	–	ﾘｾｯﾄで応用開始
	4		$01	$0000						応用開始、0番地へ飛ぶ
ID_SELECT_MEM_PAGE ("CRIS"<<4)+6	3	$00	ﾒﾓﾘ空間	ﾍﾟｰｼﾞ	–	–	–	–	–	活動なし
		$01								ﾒﾓﾘ空間選択
		$02								ﾍﾟｰｼﾞ選択
		$03								ﾒﾓﾘ空間とﾍﾟｰｼﾞを選択

表9-2. CAN規約要約 – ﾌﾞｰﾄﾛｰﾀﾞからの応答

ISP命令応答識別子	長さ	ﾃﾞｰﾀ [0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	説明
ID_SELECT_NODE ("CRIS"<<4)+0	2	ﾌﾞｰﾄﾛｰﾀﾞ改訂	$00	–	–	–	–	–	–	通信を閉じる
			$01							通信を開く
ID_PROG_START ("CRIS"<<4)+1	0	–	–	–	–	–	–	–	–	書き込み初期化命令OK
	1	$00								消去終了
ID_PROG_DATA ("CRIS"<<4)+2	1	$00	–	–	–	–	–	–	–	命令OK、転送の最後
		$02								命令OK、新しいﾃﾞｰﾀを期待
ID_DISPLAY_DATA ("CRIS"<<4)+3	最大8	8ﾊﾞｲﾄまでのﾃﾞｰﾀ ﾊﾞｲﾄ								ﾃﾞｰﾀ読み込み
	0	–	–	–	–	–	–	–	–	ﾘｾｯﾄで応用開始
	2	非空白の最初のｱﾄﾞﾚｽ		–	–	–	–	–	–	空白検査での異常
ID_SELECT_MEM_PAGE またはID_ERROR ("CRIS"<<4)+6	1	$00	–	–	–	–	–	–	–	選択OKまたはｿﾌﾄｳｪｱ保護設定異常

10. 追補C：UART規約要約

表10-1. UART規約要約 – ホストからの要求

ISP命令要求 (R)		記録長	変位	記録形式	データ[0]	データ[1]	データ[2]	データ[3]	データ[4]	～	データ[n-1]
メモリ書き込み	":"	n	先頭アドレス	$00	nバイト分のデータ (n≦255)						
応用開始	":"	$00	$0000	$01	–	–	–	–	–	–	–
新ページ選択	":"	$02	開始アドレス	$02	上=ページ 下=$0	$00	–	–	–	–	–
メモリ選択	":"	$02	$0000	$04	メモリ空間	ページ	–	–	–	–	–
メモリ読み込み	":"	$05	$0000		開始アドレス		終了アドレス		$00	–	–
メモリ空白検査	":"	$05	$0000		開始アドレス		終了アドレス		$01	–	–
メモリ消去	":"	$05	$0000		$00	$FF	$00	$00	$02	–	–

表10-2. UART規約要約 – ブートローダからの応答

応答 (A)	文字[0]	文字[1]	文字[2]	文字[3]	～	文字[n]
命令終了 (OK)	"."	CR ($0D)	LF ($0A)	–	–	–
メモリ読み込み命令終了	アドレス=(16バイト構成にされた)データ,CR,LF					
不正チェックサム	"X"	CR ($0D)	LF ($0A)	–	–	–
メモリ空白検査異常	最初の誤っているアドレス		CR ($0D)	LF ($0A)	–	–
異常 – ソフトウェア保護設定 ("ISPメモリ読み込み"のみ)	"L"	CR ($0D)	LF ($0A)	–	–	–
異常 – ソフトウェア保護設定 ("ISPメモリ書き込み"のみ)	"P"	CR ($0D)	LF ($0A)	–	–	–

本社

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131, USA
TEL 1(408) 441-0311
FAX 1(408) 487-2600

国外営業拠点

Atmel Asia
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
TEL (852) 2245-6100
FAX (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-Yvelines
Cedex
France
TEL (33) 1-30-60-70-00
FAX (33) 1-30-60-71-11

Atmel Japan
104-0033 東京都中央区
新川1-24-8
東熱新川ﾋﾞﾙ 9F
ｱﾄﾒﾙ ｼﾞｬﾊﾟﾝ株式会社
TEL (81) 03-3523-3551
FAX (81) 03-3523-7581

製造拠点

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
TEL 1(408) 441-0311
FAX 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
TEL 1(408) 441-0311
FAX 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3
France
TEL (33) 2-40-18-18-18
FAX (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-42-53-60-00
FAX (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR
Scotland
TEL (44) 1355-803-000
FAX (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn
Germany
TEL (49) 71-31-67-0
FAX (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Biometrics
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex
France
TEL (33) 4-76-58-47-50
FAX (33) 4-76-58-47-60

文献請求
www.atmel.com/literature

本応用記述はATMELのAVR914応用記述(doc7592.pdf Rev.7592B-01/06)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意訳されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

必要と思われる部分には( )内に英語表記や略称などを残す形で表記しています。

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