

NAPYDC839GR08 共通 CAN プログラミング技術資料

対応 MCU 名

R5F10PxJ (x = P,M,L,G)

<ご注意>

下記の利用条件をご了解の上本技術情報をご利用ください。

<本技術情報の利用条件>

1. ホームページ上で公開される共通 CAN プログラミングに関する情報
(以下本技術情報と呼びます)は、あくまでもマイコン導入時の評価・実験
用途として開示されるものであり、生産ライン用プログラマとして応用される
ことを想定していません。
本技術情報を、フラッシュマイコンを組み込んだ製品等の生産用途用として
ご利用になる際は、お客様サイドで本技術情報に関する妥当性を十分検討のうえ
ご利用ください。
2. 横河デジタルコンピュータは、正確な技術情報の開示に努力しますが、
本技術情報の内容について製造責任を負うものではありません。
本技術情報を応用した結果についての責任は、お客様に帰属するものとします。
3. 弊社では、本技術情報を生産用途などに応用するお客様を対象に、本技術
情報に関する技術支援サービス(有償)を行っております。
詳細は、弊社または弊社代理店までお問い合わせください。(日本国内のみ)

ご注意

NAPYDC839GR08 の適用 NETIMPRESS シリーズ本体は、NETIMPRESS air(AF930)です。

C^oarNETIMPRESS,G-NETIMPRESS,NETIMPRESS next ではご使用になれません。

変更履歴

変更日付	変更内容
2015.03.26	新規作成
2015.04.14	誤記修正

目次

変更履歴	1
目次.....	2
1. 概要.....	5
1.1 動作条件.....	5
1.2 マイコンパック内容.....	6
2. 用語の定義と略語.....	7
3. 機能概要	11
3.1 UCOP システム構成図.....	11
3.2 ROM.....	12
3.2.1 ROM 構成図	12
3.2.2 消去ブロックアドレス(ブートスワップモード).....	13
3.2.3 消去ブロックアドレス(非ブートスワップモード時).....	13
3.3 プログラムエントリモードフローチャート.....	14
3.4 IBL プログラム概略フローチャート.....	15
4. 初期導入手順	16
4.1 書き込み手順フロー.....	16
4.2 設定変更項目.....	17
4.2.1 CANボーレートの変更.....	17
4.2.2 動作クロックの変更	17
4.2.3 ビットタイミングパラメータ、ボーレートプリスケーラの変更	17
4.2.4 パスワードチェック領域の変更.....	17
4.2.5 ユーザアプリ領域サム値チェック領域の変更	17
4.2.6 ウォッチドッグタイマサービスの変更	17
4.2.7 Primary IDの変更.....	18
4.2.8 ステーションアドレスの変更.....	18
4.2.9 内蔵ウォッチドッグタイマの変更	18
4.2.10 ブートスワップ機能の変更	18
5. UCOP 設定変更方法.....	19
5.1 CAN ボーレート	19
5.2 入力クロック周波数	20
5.3 クロック逡倍比.....	20
5.4 クロック分周比.....	20
5.5 CAN ビットレートプリスケーラレジスタ値	21
5.6 CAN ビットコンフィグレーションレジスタ値	21
5.7 CAN クロック選択ビット値.....	21
5.8 パスワードチェック領域開始アドレス.....	22

5. 9	パスワードチェック領域終了アドレス.....	22
5. 10	ユーザアプリ領域サム値チェック開始アドレス.....	22
5. 11	ユーザアプリ領域サム値チェック終了アドレス.....	23
5. 12	I/O ポートサービス対応フラグ	23
5. 13	I/O ポートサービス周期	24
5. 14	I/O ポートサービス用ポート変更方法	24
5. 15	Primary ID.....	24
5. 16	CAN ID フォーマット設定.....	27
5. 17	ステーションアドレス.....	28
5. 18	内蔵ウォッチドッグタイマ設定	28
5. 19	内蔵ウォッチドッグタイマ周期設定	28
5. 20	ブートスワップ機能設定フラグ	29
5. 21	Specific Parameter 変更方法	30
6.	UCOP システム概要.....	34
6. 1	イニシャル・プロセッシング・ルーチン (IPR)	34
6. 2	イニシャルブートローダ (IBL) (C 言語プログラム、リロケータブルオブジェクト)	35
6. 3	書き込み制御プログラム (WCP)	35
6. 4	書き込みプロセス正常終了判定.....	37
6. 5	アイデンティファイヤ (CAN メッセージ ID).....	38
6. 5. 1	Primary ID.....	38
6. 5. 2	Secondary ID.....	38
6. 5. 3	送受信メッセージバッファ	38
6. 6	ブートスワップ機能	39
6. 7	ステータスレジスタ.....	40
6. 8	プログラムエントリモード	41
6. 9	u Entry 時ユーザ APL 処理項目	42
6. 10	誤 Entry 時無限ループ防止機能	43
6. 11	CAN ボーレート設定時の注意	44
6. 12	ステーションアドレス.....	44
6. 13	プログラム終了時の処理.....	44
6. 14	ウォッチドッグタイマ.....	45
6. 15	IBL 処理時間	46
7.	r Entry モード仕様.....	47
7. 1	概要	47
7. 2	r Entry モード使用方法	48
8.	YDC製IBL、WCPの構成	49
9.	RAMの使用方法.....	50
10.	CAN プロトコル	51
10. 1	フレームの種類.....	51
10. 2	IBL 対応コマンド.....	51

10.3 WCP 対応コマンド	52
11. 関数一覧	53
11.1 IBLでの使用関数(y_ibl. cファイルの関数一覧)	53
11.2 WCPでの使用関数(y_wcp. cファイルの関数一覧)	57
12. 使用I/Oリソース一覧	63
13. 付録	64

1. 概要

1.1 動作条件

項目	内容	ユーザ設定※
対象マイコン	R5F10PxJ (x=P,M,L,G)	
書き込み対象アドレス	コードフラッシュ 256K: #000000h~#03FFFFh *1 データフラッシュ 8K: #0F1000h~#0F2FFFh	不可
インタフェース	CAN 通信(拡張・標準 ID 対応)	
ボーレート	500Kbps、1Mbps、250Kbps、125Kbps (デフォルト 500Kbps)	可
動作クロック	入力=1MHz~20MHz、動作=1MHz~64MHz 逡倍値=1,4,8 分周値=1 (デフォルト 入力=4MHz、動作=32MHz、逡倍値=8、 分周値=1)	可
CAN チャンネル番号	チャンネル 0	不可
モード制御端子	なし	不可
バンドルファイル	BTP ファイル、YSM ファイル、KEY ファイル AMK ファイル	不可
プローブ	AZ915 AZ916	不可
CCP バージョン	Version 2.1 をベースとする *2	不可
開発環境	Renesas 製 CS+ for CA,CX V3.00.00 *3	不可
コンパイラバージョン	Renesas 製 CA78K0R V1.71 *3	不可
最適化	IBL,WCP 共に”標準(-qx2)”を使用	不可
プログラマ共通仕様	特定領域プロテクト機能(定義体にてアクセス禁止パ ラメータ対応)	
IBL 内のスタックレベル	256byte	不可

※表中の「ユーザ設定」が「可」以外の項目は絶対に設定変更しないで下さい。以降ページも同様です。

*1 ブートスワップモード時の領域です。

IPR,IBL 領域を含みます。その領域には、所定のプログラムを格納してください。

非ブートスワップモード時の領域は下記のようになります。

#000C00~#03FFFF (253K)

非ブートスワップモードは非標準です。詳しくは弊社までお問い合わせください。

*2 CCP を拡張したプロトコルです。完全互換性はありません。

*3 開発環境及びコンパイラバージョンは弊社にて動作確認を行ったバージョンになります。

他のバージョンのものを使用された場合の動作は保証致しかねますのでご注意ください。

1.2 マイコンパック内容

本マイコンパックに関する公開ドキュメント一覧(和文)

項 目	ドキュメント名(ファイル名)	備 考
マイコンパックマニュアル	MNJ-NAPYDC839GR08	
CAN 共通プログラミング 技術資料	TR- NAPYDC839GR08	本書
マイコンパック	NAPYDC839GR08	・FDF シート内容 サンプル APL オブジェクト KEY ファイル YSM ファイル
サンプルプログラム アプリソフト例 (APL) ユーザイニシャライズルーチン (IPR) イニシャルブートローダ (IBL) 書き込み制御プログラム (WCP)	EX- NAPYDC839GR08 -Header -y_ibl.h -y_init.h -IBL- NAPYDC839GR08 -user_init.h -user_apl.c -usr_ipr_init.asm - vect00.asm -user_ipr.c -y_ibl.c -ibl.dr -WCP- NAPYDC839GR08 -y_wcp_init.asm -y_wcp.c -wcp.dr -fsl_types.h -fsl.h -fsl.lib -pfdl_types.h -pfdl.h -pfdl.lib	固有値定義ファイル 初期設定ファイル ユーザ APL/IPR サンプルファイル ユーザ APL サンプルファイル IPR スタートアップルーチンサンプル リセットベクタ設定サンプル ユーザ IPR サンプルファイル CAN リプログ用ブートローダ リンクディレクティブファイル WCP スタートアップルーチン CAN リプログ用書き込み制御プログラム リンクディレクティブファイル セルフライブラリヘッダファイル セルフライブラリヘッダファイル セルフライブラリ データフラッシュライブラリヘッダファイル データフラッシュライブラリヘッダファイル データフラッシュライブラリ

2. 用語の定義と略語

UCOP

Universal CAN Open Protocol の略です。
弊社が提唱した MCU に依存しない CAN 共通プロトコルです。

IPR

Initial Processing Routine の略です。
イニシャル・プロセッシング・ルーチン・プログラムです。
プログラミング上、初期化しなければならない処理を記述いただきます。
お客様サイドでカスタマイズしていただきます。

IBL

Initial Boot Loader の略です。
イニシャル・ブート・ローダ・プログラムです。
プログラミングエントリの判定、書き込み制御プログラム(WCP)の受信
及び内蔵 RAM への書き込みをおこないます。
基本的にはそのままご使用していただけます。

WCP

Write Control Program の略です。
書き込み制御プログラムです。
拡張子が“.BTP”のファイルです。
デバイスに対する消去・書き込み・読み出し等のプログラムが書かれています。
基本的にはそのままご使用していただけます。

APL

アプリケーション・プログラムです。
お客様のアプリケーションプログラムです。

ReProg Area

お客様のアプリケーションプログラムを書き込む ROM エリアです。

UCOP リプログモード

UCOPを利用してアプリケーションプログラムの消去／書き込みを行うモードをUCOPリプログモードと呼びます。
UCOP リプログモードへは3つあるエントリー方法のどれかを通してエントリーします。

r Entry

レスキュー・エントリー

UCOP リプログラムモードに入るエントリー方法の 1 つです。

電源投入後、一定期間(※)経過後、約 10mSec 間 Connect コマンドを待ちます。

この約 10mSec 間に Connect コマンドを受信すると r Entry になります。

※この一定期間は電源投入後 Connect コマンド受信待ちを開始するまでの時間で IPR の処理時間などで時間が変わってきます。

n Entry

ノーマル・エントリー

UCOP リプログラムモードに入るエントリー方法の 1 つです。

IBL 内で Connect コマンドを受信するまで待ちつづけます。

u Entry

ユーザ・エントリー

UCOP リプログラムモードに入るエントリー方法の 1 つです。

APL 内で Connect コマンドを受信した場合のエントリー方法です。

APL 内での Connect コマンド受信方法は、お客様次第です。

Primary ID

初期設定ファイル(y_init.h)の ID_P_NI と ID_P_MCU に設定されている
アイデンティファイヤです。

Secondary ID

UCOP リプログラムモード中に追加したアイデンティファイヤです。

ROM の一部にアイデンティファイヤ登録領域(以下「Secondary ID」という)
を確保し、その領域へ追加したアイデンティファイヤを登録します。

CAN メッセージ ID

CAN プロトコルのフレームにおける、アイデンティファイヤのことです。

KILL レジスタ

UCOP リプログラムモードを強制終了するかどうかを判定する機能です。

ROM の一部を KILL レジスタ領域とします。

KILL レジスタ領域が All FFh でない場合、KILL レジスタ ON となります。

KILL レジスタ領域が All FFh の場合、KILL レジスタ OFF となります。

KILL レジスタ ON 時は、リセット実行処理関数をコールし UCOP リプログラムモードから抜けます。

KILL レジスタ OFF 時は、UCOP リプログラムモードを続行します。

ステーションアドレス

ターゲット毎に2バイト(リトルエンディアン)で設定します。

初期設定ファイル(y_init.h)の CCP_STATION で設定します。

Connect コマンド、Disconnect コマンドのフレームにステーションアドレス情報が入っています。
(UCOP プロトコルのマニュアル参照)

Connect コマンドにおいてアイデンティファイヤ、ステーションアドレスが一致した場合のみ IBL は UCOP リプログモードにエントリーします。

Disconnect コマンドのステーションアドレスは無視します。

パスワードチェック領域

UCOPでは「暗号機能^{※1}」があります。暗号機能においてチェックを行うID数はある領域内において7~256バイト迄で設定します。その領域を「パスワード設定領域」といいます。
この領域はお客様サイドで変更していただくことが可能です。

※1:「暗号機能」については「UCOP_CAN PROGRAMMER」のインストラクション
マニュアルを参照してください。

ユーザアプリ領域サム値チェック

UCOPではIBLにおいて、お客様のアプリケーションプログラムが既にかかれているかどうかをサム値にて判断します。このサム値チェックのことを「ユーザアプリ領域サム値チェック」といいます。

サム値計算領域は”y_init.h”ファイルで変更することが可能です。

ブートスワップ機能

この機能は対象マイコンが持つ、独自の機能です。

ブート・スワップ機能とは“ブートとなるブロックを変更する機能”を言います。つまり、普段のブート領域は“ブロック 0~3”ですが、“ブロック 4~7”をブートにすることができます。ブート領域を変更することにより、書き込み途中でのエラー等で”ブロック 0~3”の内容が不定になっても、“ブロック 4~7”からブート処理が開始され、システムの再起動が不可能になってしまうことを防ぐことができます。

詳細は対象マイコンのマニュアルをご覧ください。

本マイコンパックでは、マイコンが持つこの機能を使用することにより、ブート領域書き換え中の電源瞬断などの予期せぬ事態に対しても、再度書き込みが実行できます。

ブートスワップモード

上記、ブートスワップ機能を用いてブロック 0~7 を書き換えるモードです。

ブートスワップモードかどうかは、初期設定ファイル(y_init.h)で設定することができます。

ただし、ブートスワップモードと非ブートスワップモードでは、使用するパラメータファイルも異なりますのでご注意ください。

非ブートスワップモード

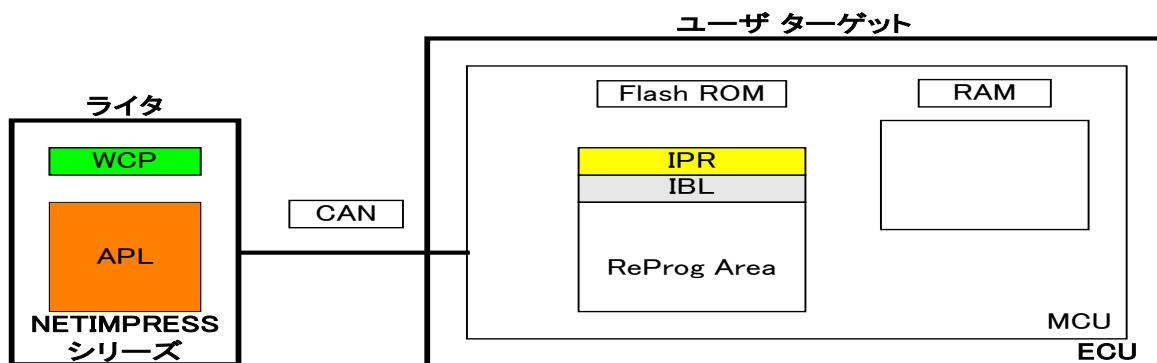
上記、ブートスワップ機能を用いず、ブロック 0~7 を書き換えないモードです。

非ブートスワップモードかどうかは、初期設定ファイル(y_init.h)で設定することができます。

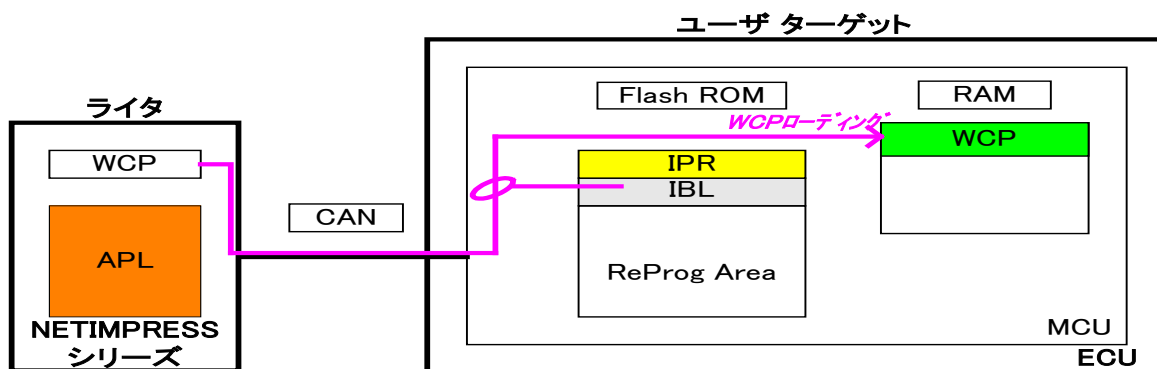
ただし、ブートスワップモードと非ブートスワップモードでは、使用するパラメータファイルも異なりますのでご注意ください。

3. 機能概要

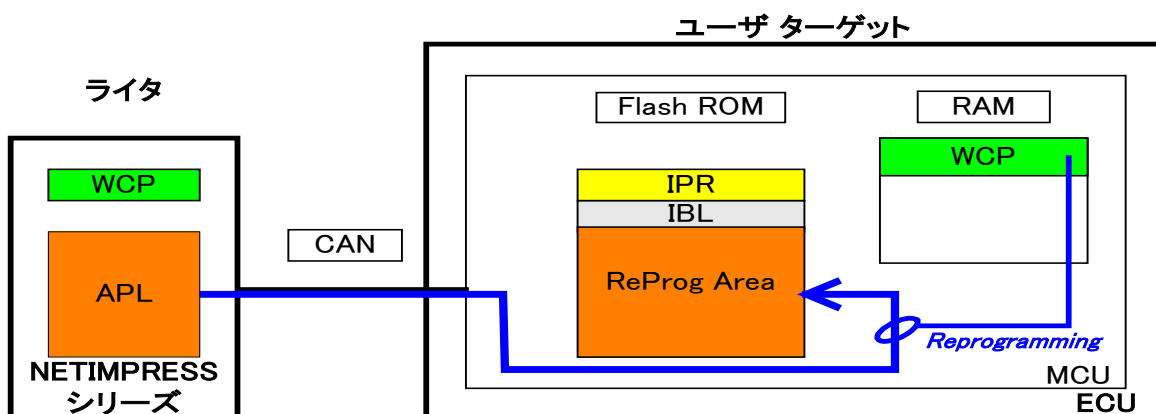
3.1 UCOP システム構成図



1. 予め、IPR と IBL はターゲット MCU の Flash ROM の一部に書き込んでおきます。
2. リセット解除後、IPR において UCOP リプログラムモード実行に際して最低限必要なシステムの初期化を行います。
3. IPR で初期化終了後、制御が IBL へ移行し各エン트리 (r Entry, n Entry, u Entry) のどれかを介してターゲットは UCOP リプログラムモードに入ります。



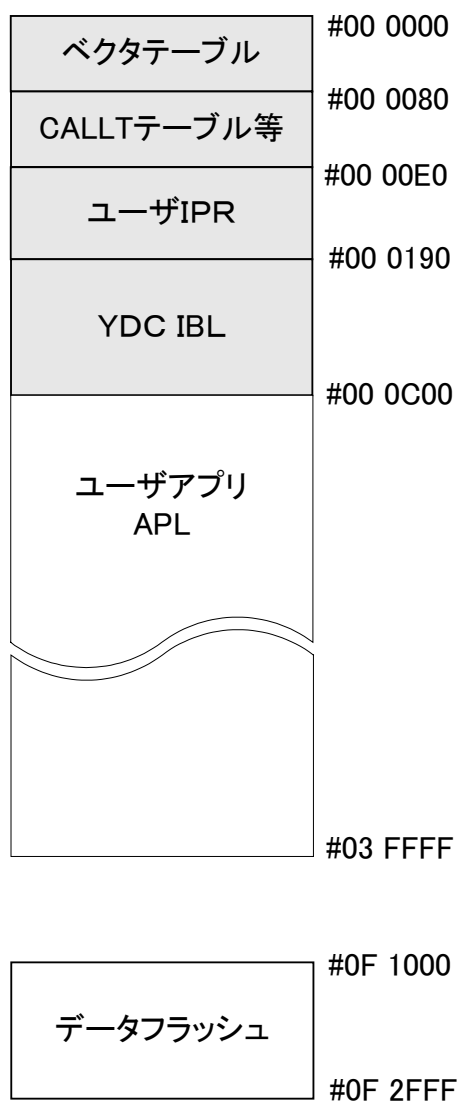
4. ライタは IBL と通信をおこない WCP をターゲット MCU に順次送信します。IBL はライターより受信した WCP をターゲット MCU の内蔵 RAM に書き込みます。
5. WCP を全て内蔵 RAM に書き込んだ後、ターゲット MCU 側の制御は WCP へ移行します。



6. ライタは WCP と通信をおこないライターにある APL を ReProg Area に書き込みます。

3. 2 ROM

3. 2. 1 ROM構成図



3. 2. 2 消去ブロックアドレス(ブートスワップモード)

ブロック	アドレス
ブロック 0	#000000～#0003FF
ブロック 1	#000400～#0007FF
⋮	⋮
ブロック 254	#03F800～#03FBFF
ブロック 255	#03FC00～#03FFFF

データフラッシュブロック	アドレス
データフラッシュブロック 0	#0F1000～#0F13FF
データフラッシュブロック 1	#0F1400～#0F17FF
⋮	⋮
データフラッシュブロック 6	#0F2800～#0F2BFF
データフラッシュブロック 7	#0F2C00～#0F2FFF

アドレス#000000～#001FFF を書き換えるときにはブートスワップ機能を使用します。

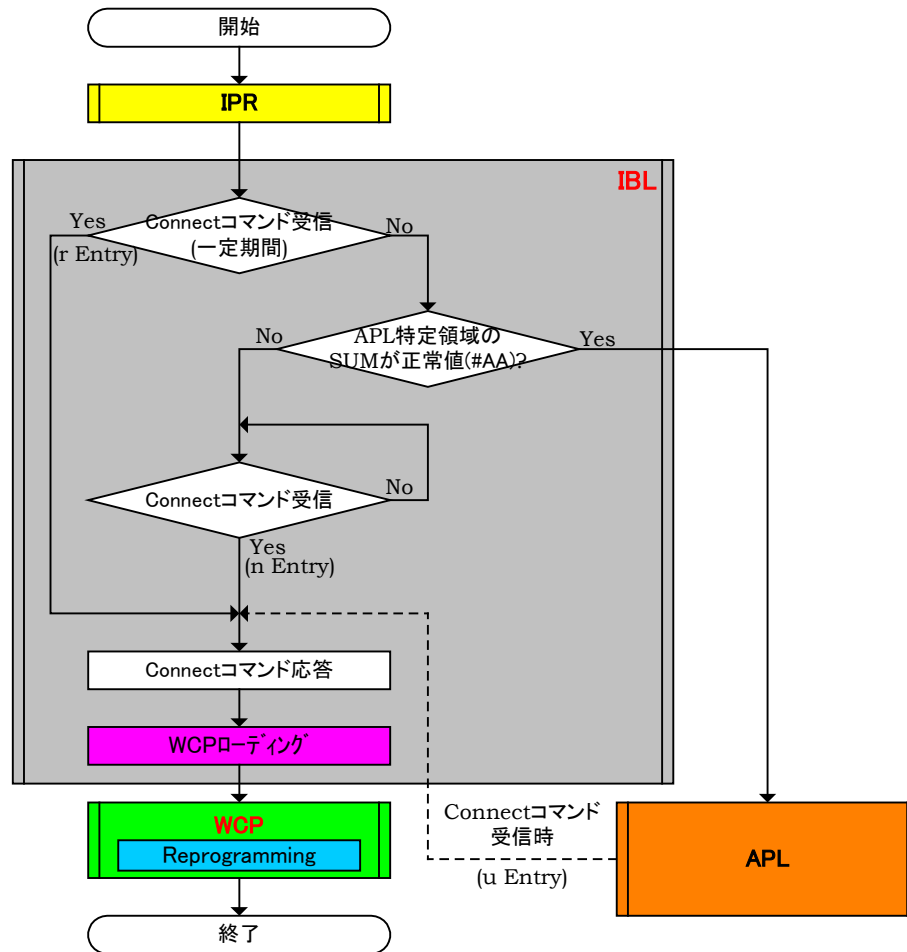
3. 2. 3 消去ブロックアドレス(非ブートスワップモード時)

ブロック	アドレス
ブロック 0	#000000～#0003FF
ブロック 1	#000400～#0007FF
ブロック 2	#000800～#000BFF
ブロック 3	#000C00～#000FFF
ブロック 4	#001000～#0013FF
⋮	⋮
ブロック 254	#03F800～#03FBFF
ブロック 255	#03FC00～#03FFFF

データフラッシュブロック	アドレス
データフラッシュブロック 0	#0F1000～#0F13FF
データフラッシュブロック 1	#0F1400～#0F17FF
⋮	⋮
データフラッシュブロック 6	#0F2800～#0F2BFF
データフラッシュブロック 7	#0F2C00～#0F2FFF

灰色部は IBL、IPR 域で書き換え禁止領域です。

3.3 プログラムエントリモードフローチャート



1. 電源投入後、IPR 処理をおこない一定期間(約 10mS)Connect コマンドを待ちます。この期間内に Connect コマンドを受信しますと r Entry で UCOP リプログモードに遷移します。
2. 一定期間(約 10mS)内に Connect コマンドを受信しなかった場合、ユーザアプリ領域サム値チェックの SUM 値を計算します。SUM 値が#AA ならば APL ヘジジャンプし、お客様のアプリケーションプログラムが実行されます。APL 側で Connect コマンドを受信しますと u Entry で UCOP リプログモードに遷移します。
3. ユーザアプリ領域サム値チェックの SUM 値が#AA 以外ならば、Connect コマンドを受信するまで Connect コマンド受信待ちになります。この状態で Connect コマンドを受信しますと n Entry で UCOP リプログモードに遷移します。

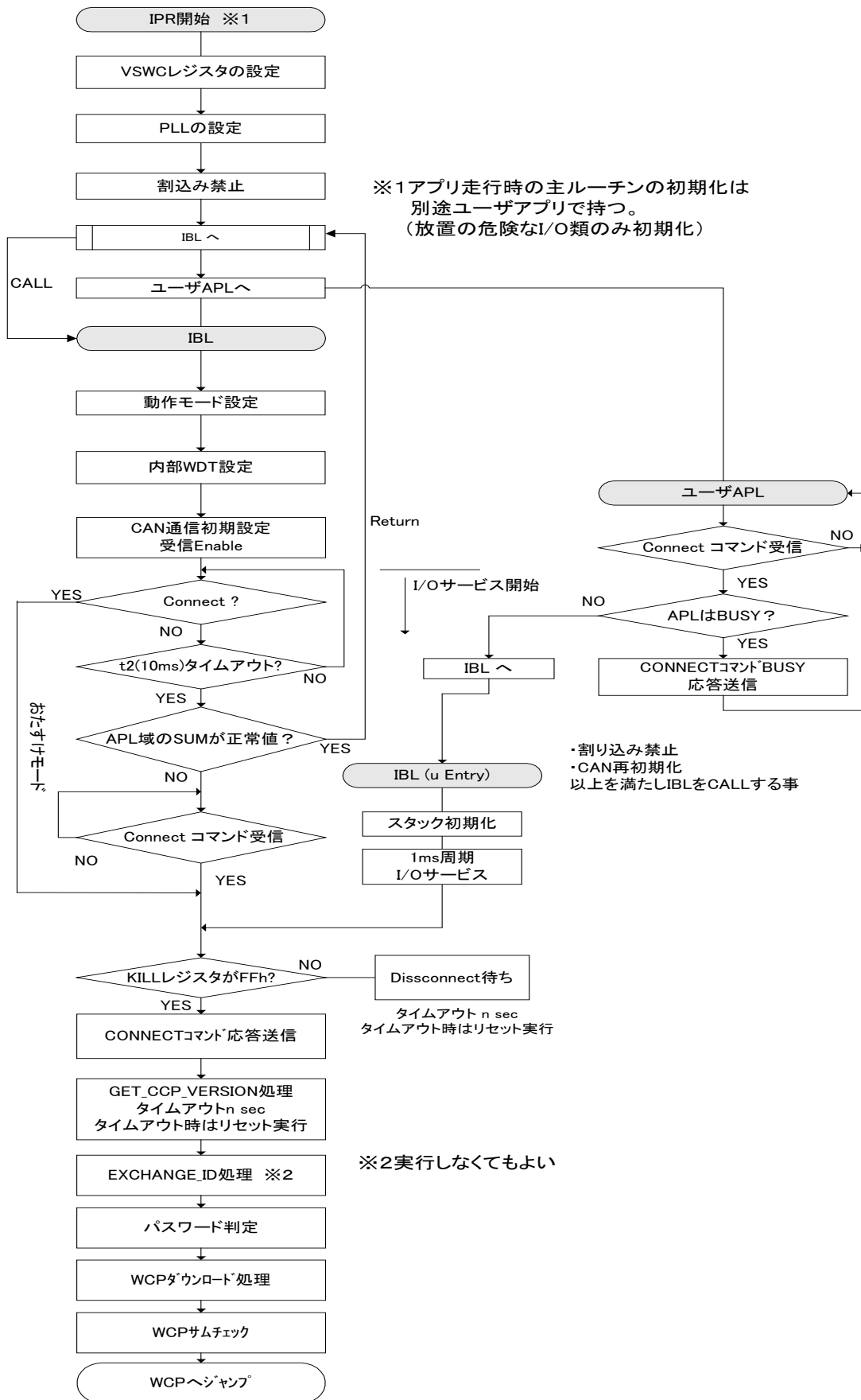
※ライタは Connect コマンド発行後規定時間(25mS)応答がない場合、Disconnect コマンドを発行し、ターゲット MCU との接続を解除します。

※ターゲット MCU は各エントリー方式で UCOP リプログモードに遷移後、Disconnect コマンドを受信するまでライタとは接続状態にあるものとします。

※Disconnect コマンドはデバイスファンクション終了時にライタが発行します。

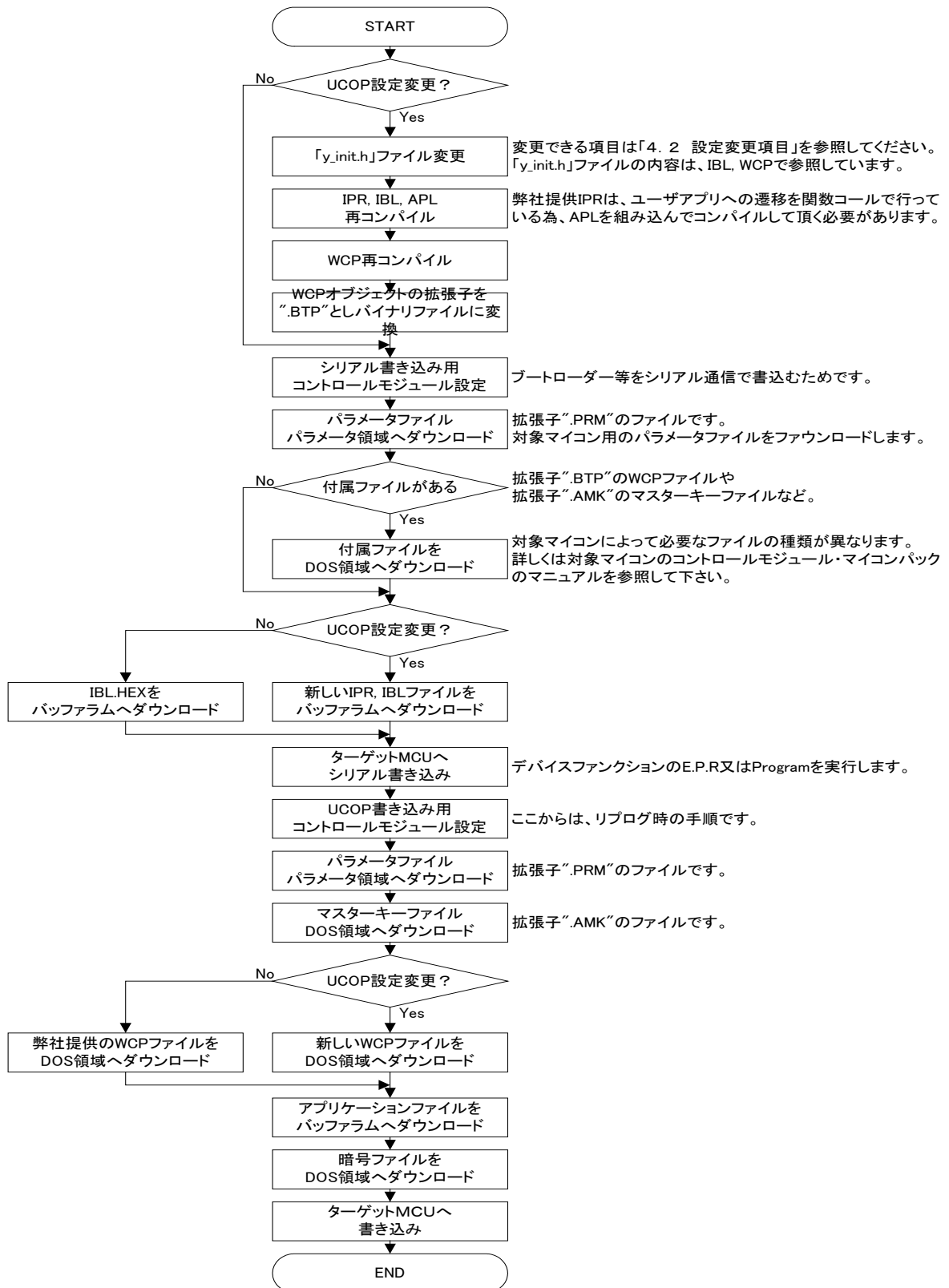
※ターゲット MCU は Disconnect コマンド受信後、リセット状態に戻るものとします。

3. 4 IBLプログラム概略フローチャート



4. 初期導入手順

4.1 書き込み手順フロー



4. 2 設定変更項目

4. 2. 1 CANボーレートの変更

「5. 1 CANボーレート」、「5. 5 ビットレートプリスケアラレジスタ値」、「5. 6 ビットコンフィグレーションレジスタ値」、「5. 7 CAN クロック選択ビット値」を参照してください。

4. 2. 2 動作クロックの変更

4. 2. 2. 1 入力クロック値の変更

「5. 2 入力クロック周波数」、「5. 5 ビットレートプリスケアラレジスタ値」、「5. 6 ビットコンフィグレーションレジスタ値」、「5. 7 CAN クロック選択ビット値」を参照してください。

4. 2. 2. 2 クロック逡倍比の変更

「5. 3 クロック逡倍比」、「5. 5 ビットレートプリスケアラレジスタ値」、「5. 6 ビットコンフィグレーションレジスタ値」、「5. 7 CAN クロック選択ビット値」を参照してください。

4. 2. 2. 3 クロック分周比の変更

「5. 4 クロック分周比」、「5. 5 ビットレートプリスケアラレジスタ値」、「5. 6 ビットコンフィグレーションレジスタ値」、「5. 7 CAN クロック選択ビット値」を参照してください。

4. 2. 3 ビットタイミングパラメータ、ボーレートプリスケアラの変更

「5. 5 ビットレートプリスケアラ値」、「5. 6 ビットコンフィグレーションレジスタ値」、「5. 7 CAN クロック選択ビット値」を参照してください。

4. 2. 4 パスワードチェック領域の変更

「5. 8 パスワードチェック領域開始アドレス」、「5. 9 パスワードチェック領域終了アドレス」を参照してください。

4. 2. 5 ユーザアプリ領域サム値チェック領域の変更

「5. 10 ユーザアプリ領域サム値チェック開始アドレス」、「5. 11 ユーザアプリ領域サム値チェック終了アドレス」を参照してください。

4. 2. 6 ウォッチドッグタイマサービスの変更

4. 2. 6. 1 ウォッチドッグタイマサービス有無の変更

「5. 12 I/O ポートサービス対応フラグ」を参照してください。

4. 2. 6. 2 ウォッチドッグタイマサービス周期の変更

「5. 13 I/O ポートサービス周期」を参照してください。

4. 2. 6. 3 ウォッチドッグタイマサービス用ポートの変更

「5. 14 I/O ポートサービス用ポート変更方法」を参照してください。

4. 2. 7 Primary IDの変更

「5. 15 Primary ID」、「5. 16 CAN ID フォーマット設定」を参照してください。

4. 2. 8 ステーションアドレスの変更

「5. 17 ステーションアドレス」を参照してください。

4. 2. 9 内蔵ウォッチドッグタイマの変更

「5. 18 内蔵ウォッチドッグタイマ設定」、「5. 19 内蔵ウォッチドッグタイマ周期設定」を参照してください。

4. 2. 10 ブートスワップ機能の変更

「5. 20 ブートスワップ機能設定フラグ」を参照してください。

5. UCOP設定変更方法

UCOP の一部の設定は、お客様のシステムに応じて変更していただくことが可能です。

ターゲット MCU 側の各種設定を行っている初期設定ファイル“y_init.h”とライター側両方の変更が必要な項目もあります。

初期設定ファイル“y_init.h”を変更された場合は、「IPR, IBL, WCP」のファイルを再コンパイルしていただく必要があります。

ライター側の変更は AZ990-air Connect を用いて行います。

一部設定につきましてはライターのファンクション機能を用いて変更することが出来ます。

air Connect の詳細な操作方法は air Connect のインストラクションマニュアルをご参照ください。

5. 1 CAN ボーレート

CAN 通信のボーレートを変更するには、初期設定ファイル“y_init.h”とライター側両方の変更が必要です。

①. 初期設定ファイル“y_init.h”設定変更

“CAN_BAUD”にボーレート値に下記の表の設定値を設定してください。

ボーレート値	設定値
1 Mbps	1000
500 Kbps	500
250 Kbps	250
125 Kbps	125

※必ずマイコンのビットコンフィグレーションレジスタ値も再計算し、初期設定ファイルの” CAN_C0CFGL_DATA”と” CAN_C0CFGH_DATA”も再設定してください。

「5. 5 ビットレートプリスケアラレジスタ値」と「5. 6 ビットコンフィグレーションレジスタ値」をご参照ください。

②. ライター側設定変更

i . air Connect での変更

Specific Parameter のアドレス#0C2,0C3 を変更することでボーレートを変更できます。

ボーレート値とアドレス#0C2,0C3 の値との関係は下記の表のようになっています。

ボーレート値	#0C2 の値	#0C3 の値
1 Mbps	0x 34	0x 40
500 Kbps	0x 34	0x 41
250 Kbps	0x 34	0x 43
125 Kbps	0x 34	0x 47

※ 「5. 21 Specific Parameter 変更方法」を参照下さい。

ii. メニューからの変更

“SUB SETTING”メニューの“CAN BAUDRATE SETING”からボーレート変更を行います。
上下キーで設定したいボーレートを選択します。

5.2 入力クロック周波数

ターゲット MCU の入力クロック周波数を変更するには、初期設定ファイル“y_init.h”ファイルの変更が必要です。

①. 初期設定ファイル“y_init.h”設定変更

“CLK_EXT”に入力クロック周波数値を 10 倍した値を設定してください。

例) 10MHz の場合、100 と設定

※必ずマイコンのビットコンフィグレーションレジスタ値も再計算し、初期設定ファイルの
” CAN_C0CFG_L_DATA”と” CAN_C0CFG_H_DATA”も再設定してください。

「5.5 ビットレートプリスケアラレジスタ値」と「5.6 ビットコンフィグレーションレジスタ値」をご参照ください。

5.3 クロック逡倍比

ターゲット MCU に入力されたクロックを PLL 逡倍回路などにより逡倍して動作周波数とする場合、その逡倍比を設定します。

逡倍比を変更するには、初期設定ファイル“y_init.h”の変更が必要です。

①. 初期設定ファイル“y_init.h”設定変更

“CLK_MUL”に逡倍比を設定してください。

例) 逡倍比が「×8」の場合、8 と設定

※必ずマイコンのビットコンフィグレーションレジスタ値も再計算し、初期設定ファイルの
” CAN_C0CFG_L_DATA”と” CAN_C0CFG_H_DATA”も再設定してください。

「5.5 ビットレートプリスケアラレジスタ値」と「5.6 ビットコンフィグレーションレジスタ値」をご参照ください。

5.4 クロック分周比

ターゲット MCU に入力されたクロックを分周回路などにより分周して動作周波数とする場合、その分周比を設定します。

分周比を変更するには、初期設定ファイル“y_init.h”の変更が必要です。

①. 初期設定ファイル“y_init.h”設定変更

“CLK_DIV”に分周比を設定してください。

例)分周比が「÷2」の場合、2と設定

※必ずマイコンのビットコンフィグレーションレジスタ値も再計算し、初期設定ファイルの
”CAN_C0CFGL_DATA”と”CAN_C0CFGH_DATA”も再設定してください。

「5. 5 ビットレートプリスケアラレジスタ値」と「5. 6 ビットコンフィグレーションレジスタ値」をご参照ください。

5. 5 CAN ビットレートプリスケアラレジスタ値

ビットレートプリスケアラレジスタの値を設定します。

ターゲット MCU の動作周波数、CAN ボーレートを変更する場合には変更してください。

ビットレートプリスケアラレジスタの値を変更するには、初期設定ファイル“y_init.h”の変更が必要です。

①. 初期設定ファイル“y_init.h”設定変更

“CAN_C0CFGL_DATA”に設定値を設定してください。

ビットレートプリスケアラレジスタ値には、0~1023(0x0000~0x03FF)までの値を設定してください。

設定値はマイコンのマニュアルを参照して計算してください。

※必要に応じて CAN ボーレートやクロック関連の再設定を行ってください。

※「5. 1 CAN ボーレート」「5. 2 入力クロック周波数」「5. 3 クロック逡倍比」「5. 4 クロック分周比」「5. 7 CAN クロック選択ビット値」をご参照ください。

5. 6 CAN ビットコンフィグレーションレジスタ値

ビットコンフィグレーションレジスタの値を設定します。

ターゲット MCU の動作周波数、CAN ボーレートを変更する場合には変更してください。

ビットレートレジスタの値を変更するには、初期設定ファイル“y_init.h”の変更が必要です。

①. 初期設定ファイル“y_init.h”設定変更

“CAN_C0CFGH_DATA”に設定値を設定してください。

ビットコンフィグレーションレジスタは 16 ビットレジスタです。

設定値はマイコンのマニュアルを参照して計算してください。

※必要に応じて CAN ボーレートやクロック関連の再設定を行ってください。

※「5. 1 CAN ボーレート」「5. 2 入力クロック周波数」「5. 3 クロック逡倍比」「5. 4 クロック分周比」「5. 7 CAN クロック選択ビット値」をご参照ください。

5. 7 CAN クロック選択ビット値

CAN クロック選択ビット(GCFGL レジスタの DCS ビット)値を設定します。(0 or 1)

CAN クロックに使用するクロック源にあわせて変更してください。

CAN クロック選択ビットの値を変更するには、初期設定ファイル“y_init.h”の変更が必要です。

①. 初期設定ファイル“y_init.h”設定変更

“CAN_GCFGL_DCS”に設定値を設定してください。

設定値はマイコンのマニュアルを参照してください。

※必要に応じて CAN ボーレートやクロック関連の再設定を行ってください。

※「5. 1 CAN ボーレート」「5. 2 入力クロック周波数」「5. 3 クロック逡倍比」「5. 4 クロック分周比」「5. 5 ビットレートプリスケアラレジスタ値」「5. 6 ビットコンフィグレーションレジスタ値」をご参照ください。

5. 8 パスワードチェック領域開始アドレス

ReProg Area 内で暗号機能に使用する領域の開始アドレスを変更する場合に設定します。

パスワードチェック領域開始アドレスを変更するには、初期設定ファイル“y_init.h”の変更が必要です。

①. 初期設定ファイル“y_init.h”設定変更

“PASS_START”にパスワードチェック領域の開始アドレスを設定してください。

パスワードチェック領域開始アドレスのデータもパスワードチェックの対象とすることができます。

パスワードチェック領域中の 7byte 以上のデータをチェックします。

パスワードをチェックするデータのサイズが 7byte 未満の場合、エラーになります。

パスワードチェック領域中のすべてのデータをチェックする必要はありません。

5. 9 パスワードチェック領域終了アドレス

ReProg Area 内で暗号機能に使用する領域の終了アドレスを変更する場合に設定します。

パスワードチェック領域終了アドレスを変更するには、初期設定ファイル“y_init.h”の変更が必要です。

①. 初期設定ファイル“y_init.h”設定変更

“PASS_END”にパスワードチェック領域の終了アドレスを設定してください。

パスワードチェック領域終了アドレスのデータもパスワードチェックの対象とすることができます。

パスワードチェック領域中の 7byte 以上のデータをチェックします。

パスワードをチェックするデータのサイズが 7byte 未満の場合、エラーになります。

パスワードチェック領域中のすべてのデータをチェックする必要はありません。

パスワードチェック領域終了アドレスは最低でもパスワードチェック領域開始アドレスから 7byte 分サイズを確保して設定してください。

5. 10 ユーザアプリ領域サム値チェック開始アドレス

書き込みプロセス正常終了判定(6-4. 参照)に使用する“ユーザアプリ領域サム値チェック”領域の開始アドレスを変更する場合に設定します。

ユーザアプリ領域サム値チェック開始アドレスを変更するには、初期設定ファイル“y_init.h”とライタ側両方の変更が必要です。

注意:この領域はコードフラッシュ範囲内で設定する必要があります。

①. 初期設定ファイル“y_init.h”設定変更

“APL_SUM_START”に設定値を設定してください。

32ビットで設定します。

ユーザアプリ領域サム値チェック開始アドレスのデータもサム値演算の対象になります。

②. ライタ側設定変更

air Connect でのみ変更可能です。

Specific Parameter のアドレス#140,141,142,143 を変更することでユーザアプリ領域サム値チェック開始アドレスを変更できます。

例)ユーザアプリ領域サム値チェック開始を「0xFFAABBCC」と設定する場合

Specific Paramete のアドレス	#140	#141	#142	#143
設定値	0xFF	0xAA	0xBB	0xCC

※ 「5. 21 Specific Parameter 変更方法」を参照下さい。

5. 11 ユーザアプリ領域サム値チェック終了アドレス

書き込みプロセス正常終了判定(6-4. 参照)に使用する“ユーザアプリ領域サム値チェック”領域の終了アドレスを変更する場合に設定します。

ユーザアプリ領域サム値チェック終了アドレスを変更するには、初期設定ファイル“y_init.h”とライタ側両方の変更が必要です。

注意:この領域はコードフラッシュ範囲内で設定する必要があります。

①. 初期設定ファイル“y_init.h”設定変更

“APL_SUM_END”に設定値を設定してください。

32ビットで設定してください。

ユーザアプリ領域サム値チェック終了アドレスのデータもサム値演算の対象になります。

②. ライタ側設定変更

air Connect でのみ変更可能です。

Specific Paramete のアドレス#144,145,146,147 を変更することでユーザアプリ領域サム値チェック終了アドレスを変更できます。

例)ユーザアプリ領域サム値チェック開始を「0xFFEEDDCC」と設定する場合

Specific Paramete のアドレス	#144	#145	#146	#147
設定値	0xFF	0xEE	0xDD	0xCC

※ 「5. 21 Specific Paramete 変更方法」を参照下さい。

5. 12 I/O ポートサービス対応フラグ

UCOP では、I/O ポートを制御することにより外部ウォッチドッグタイマの制御を行う仕組みを持っています(6-14. 参照)。

I/O ポートサービス対応フラグを変更することで I/O ポートサービスの有無を設定します。

I/O ポートサービス対応フラグを変更するには、初期設定ファイル“y_init.h”の変更が必要です。

①. 初期設定ファイル“y_init.h”設定変更

“IOS_ON”に「0」か「1」を設定します。

「0」の場合:I/O ポートサービスなし

「1」の場合:I/O ポートサービスあり

5. 13 I/O ポートサービス周期

I/O ポートサービスを行う周期を設定します。

I/O ポートサービス周期を変更するには、初期設定ファイル“y_init.h”の変更が必要です。

①. 初期設定ファイル“y_init.h”設定変更

“IOS_PERIOD”に I/O ポートサービスの周期を設定してください。

周期の単位は msec です。

5. 14 I/O ポートサービス用ポート変更方法

I/O ポートサービス用のポートを変更するためには、“y_init.h”変更する必要があります。

```
•#define IOS_P_PORT      0x0FFF08
```

I/O サービス用ポートのポートレジスタアドレスです。

I/O サービスを実行するポートのアドレスを設定してください。

```
•#define IOS_PD_PORT     0x0FFF28
```

I/O サービス用ポートのポート方向レジスタです。

I/O サービスを実行するポートのアドレスを設定してください。

```
•#define IOS_BIT         0x01
```

I/O サービス用のポートを示す Bit です。

I/O サービスを実行するポートのビットに 1 をセットしてください。

上記の例では、IOS_P_PORT で設定したレジスタのビット 0 に割り当てられているポートを使用します。

5. 15 Primary ID

Primary ID を変更するには初期設定ファイル“y_init.h”の変更が必要です。

ライターからマイコンへ送る Primary ID を設定する“ID_P_NI”と、マイコンからライターへ送る Primary ID を設定する“ID_P_MCU”があります。

必要に応じて“ID_P_NI”や“ID_P_MCU”の変更を行ってください。

また、通信を行うためにはライター側で設定するアイデンティファイヤの変更も必要です。

「6.5 アイデンティファイヤ(CAN メッセージ ID)」をご参照ください。

①. 初期設定ファイル“y_init.h”設定変更

“ID_P_NI”、“ID_P_MCU”に設定値を設定してください。

32ビットで設定してください。

32ビットは下記のように割り当てられています。

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
IDE			STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6

Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
STD_ID5	STD_ID4	STD_ID3	STD_ID2	STD_ID1	STD_ID0	EXD_ID17	EXD_ID16

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0

IDE: アイデンティファイヤ・エクステンションの略です。

Primary ID のフォーマットがスタンダードかエクステンデッドかを識別するためのものです。

0: スタンダード・フォーマット

1: エクステンデッド・フォーマット

EXD_ID17~EXD_ID0

エクステンデッド・アイデンティファイヤを設定します。

STD_ID10 ~STD_ID0

スタンダード・アイデンティファイヤを設定します。

Bit 30、Bit 29: 予約ビット

0を設定してください。

例1) スタンダード・フォーマットの場合

“ID_P_NI”のスタンダード・アイデンティファイヤを「7E9」、「ID_P_MCU」のスタンダード・アイデンティファイヤを「7EA」と設定する場合

“ID_P_NI”: 0x000007E9

“ID_P_MCU”: 0x000007EA

例2) エクステンデッド・フォーマットの場合

“ID_P_IN”のエクステンデッド・アイデンティファイヤを「3EDCB」、スタンダード・アイデンティファイヤを「7E9」、「ID_P_MCU」のエクステンデッド・アイデンティファイヤを「3EDCB」、スタンダード・アイデンティファイヤを「7EA」と設定する場合

“ID_P_NI”: 0x9FA7EDCB

“ID_P_MCU”:0x9FABEDCB

※スタンダード・フォーマットとエクステンデッド・フォーマットのどちらを使用するかは CAN ID フォーマット設定で指定する必要があります。

「5. 16 CAN ID フォーマット設定」を参照してください。

②. ライタ側設定変更

i. air Connect での変更

マイコンからライタへ送るアイデンティファイヤ及びフレームのフォーマットは Specific Parameter のアドレス#0C4,0C5,0C6,0C7 で、ライタからマイコンへ送るアイデンティファイヤ及びフレームのフォーマットは Specific Parameter のアドレス#0C8,0C9,0CA,0CB で変更します。

※ 「5. 21 Specific Parameter 変更方法」を参照下さい。

#0C0～#0C7(#0C8～#0CB)の 32 ビットは下記のように割り当てられています。

#0C4(#0C8)							
IDE			EXD_ID17	EXD_ID16	EXD_ID15	EXD_ID14	EXD_ID13

#0C5(#0C9)							
EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	EXD_ID7	EXD_ID6	EXD_ID5

#0C6(#0CA)							
EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	STD_ID10	STD_ID9	STD_ID8

#0C7(#0CB)							
STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	STD_ID2	STD_ID1	STD_ID0

IDE: アイデンティファイヤ・エクステンションの略です。

Primary ID のフォーマットがスタンダードかエクステンデッドかを識別するためのものです。

0: スタンダード・フォーマット

1: エクステンデッド・フォーマット

EXD_ID17～EXD_ID0

エクステンデッド・アイデンティファイヤを設定します。

STD_ID10 ～STD_ID0

スタンダード・アイデンティファイヤを設定します。

例1) スタンダード・フォーマットの場合

“ID_P_NI”のスタンダード・アイデンティファイヤを「7E9」

“ID_P_MCU”のスタンダード・アイデンティファイヤを「7EA」と設定する場合

Specific Paramete のアドレス	#0C4	#0C5	#0C6	#0C7
設定値	0x00	0x00	0x07	0xEA

Specific Paramete のアドレス	#0C8	#0C9	#0CA	#0CB
設定値	0x00	0x00	0x07	0xE9

例2) エクステンデッド・フォーマットの場合

“ID_P_NI” のエクステンデッド・アイデンティファイヤを「3EDCB」、スタンダード・アイデンティファイヤを「7E9」、「ID_P_MCU」 のエクステンデッド・アイデンティファイヤを「3EDCB」、スタンダード・アイデンティファイヤを「7EA」と設定する場合

Specific Paramete のアドレス	#0C4	#0C5	#0C6	#0C7
設定値	0x9F	0x6E	0x5F	0xEA

Specific Paramete のアドレス	#0C8	#0C9	#0CA	#0CB
設定値	0x9F	0x6E	0x5F	0xE9

ii. メニューからの変更

“SUB SETTING”メニューの“CAN ID SET”からアイデンティファイヤを設定します。
 上下キーで設定するアイデンティファイヤを選択します。
 左右キーでアイデンティファイヤを変更します。

5. 16 CAN ID フォーマット設定

Primary ID のフォーマットとして、スタンダード・フォーマットとエクステンデッド・フォーマットのどちらを使用するかは、初期設定ファイル“y_init.h”と、ライターで行います。

①. 初期設定ファイル“y_init.h”設定変更

“ID_P_NI”、“ID_P_MCU”に ID のフォーマットを設定します。
 詳しくは、「5. 15 Primary ID」をご参照ください。

②. ライタ側設定変更

i. air Connect での変更

Specific Parameter でスタンダード・フォーマットかエクステンデッド・フォーマットかを設定します。
 詳しくは「5. 15 Primary ID」を参照してください。

ii. ライタでの変更

“SUB SETTING”メニューの“CAN AF → TGT ID FMT”、“CAN TGT → AF ID FMT”で設定を変更します。

“CAN AF → TGT ID FMT”はライターからマイコンへ送信するフレームの CAN ID フォーマット設定し

ます。

”CAN TGT -> AF ID FMT”はマイコンからライタへ送信するフレームのCAN IDフォーマット設定します。

上下キーでスタンダードかエクステンデッドかを選択します。

5. 17 ステーションアドレス

ステーションアドレスを変更するには、初期設定ファイル“y_init.h”とライタ側両方の変更が必要です。

「6. 12 ステーションアドレス」をご参照ください。

①. 初期設定ファイル“y_init.h”設定変更

“CCP_STATION”にステーションアドレスを 2byte(リトルエンディアン)で設定してください。

例)ステーションアドレスを「0x0200」の場合

“CCP_STATION”に「0x0002」と設定する。

②. ライタ側設定変更

air Connect でのみ変更可能となります。

Specific Parameter のアドレス#0D8,0D9 に 2byte(リトルエンディアン)で設定してください。

例)ステーションアドレスを「0x0200」の場合

Specific Parameter のアドレス	#0D8	#0D9
設定値	0x00	0x02

※ 「5. 21 Specific Parameter 変更方法」を参照下さい。

5. 18 内蔵ウォッチドッグタイマ設定

内蔵ウォッチドッグタイマを使用するかどうかを設定することができます。

設定を変更するためには、初期設定ファイル“y_init.h”の変更とオプションバイトの設定があります。

①. 初期設定ファイル“y_init.h”設定変更

“WDT_ON”に内蔵ウォッチドッグタイマ使用の有無を設定します。

0:内蔵ウォッチドッグタイマを使用しません。

1:内蔵ウォッチドッグタイマを使用します。

5. 19 内蔵ウォッチドッグタイマ周期設定

内蔵ウォッチドッグタイマのクリア周期を設定することができます。

設定を変更するためには、初期設定ファイル“y_init.h”を変更する必要があります。

①. 初期設定ファイル“y_init.h”設定変更

“WDT_PERIOD”に内蔵ウォッチドッグタイマのクリア周期を設定します。

設定の単位は ms です。

5. 20 ブートスワップ機能設定フラグ

ブートスワップを使用するかどうかを設定することができます。

設定を変更するためには、初期設定ファイル”y_init.h”を変更する必要があります。

※ブートスワップ機能の使用の有無によって、使用するパラメータファイルが異なります。

対応するパラメータファイルをご使用下さい。

①. 初期設定ファイル”y_init.h”設定変更

“BOOT_SP”にブートスワップ機能の使用の有無を設定します。

0:ブートスワップモード不使用

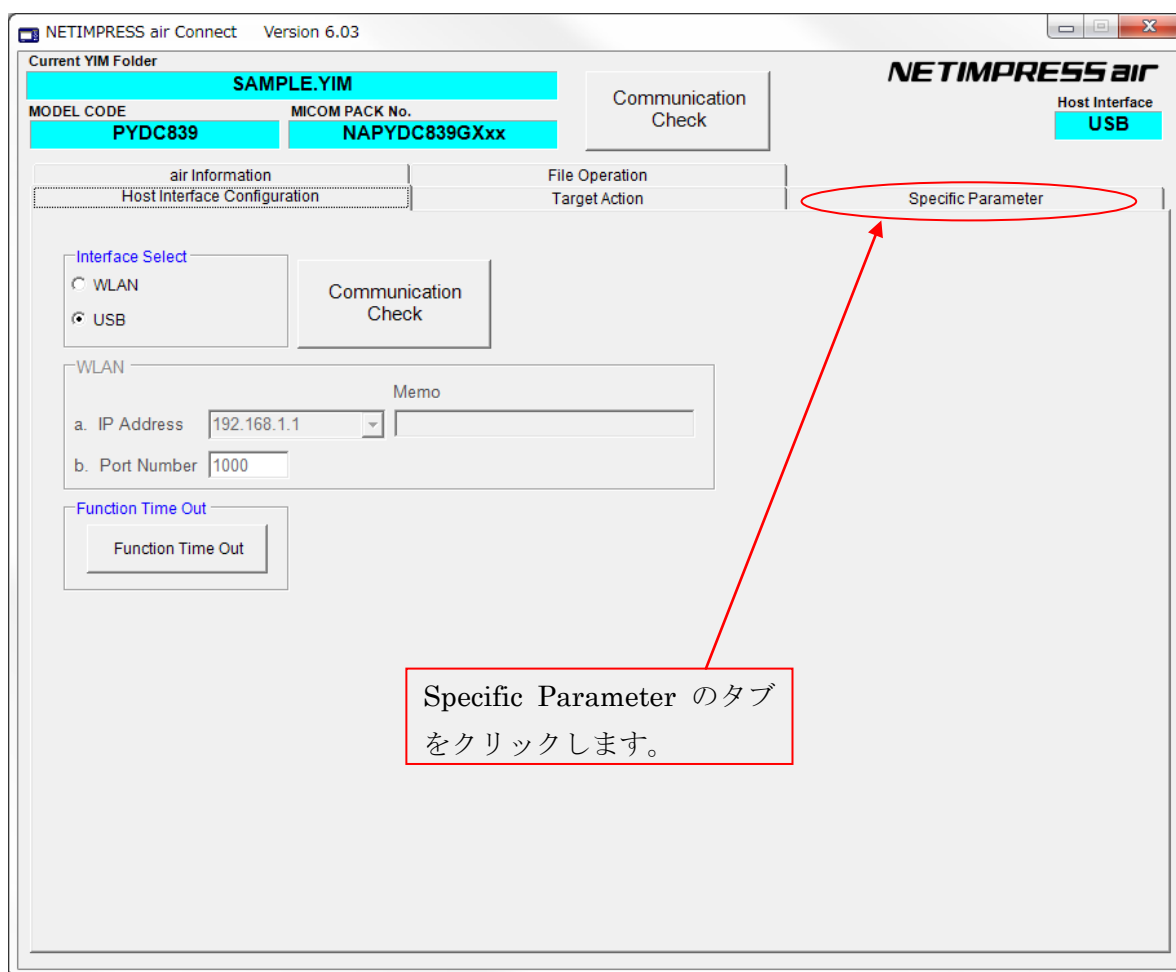
1:ブートスワップモード使用

5. 21 Specific Parameter 変更方法

ここでは、air Connect の Specific Parameter の変更方法を説明します。

まずは、air Connect を起動し、NET IMPRESS と接続してください。

Specific Parameter のタブをクリックし、Specific Parameter を開きます。



パスワードを求めるウィンドウが開きますので、"AF200"と入力し、OK ボタンを押してください。

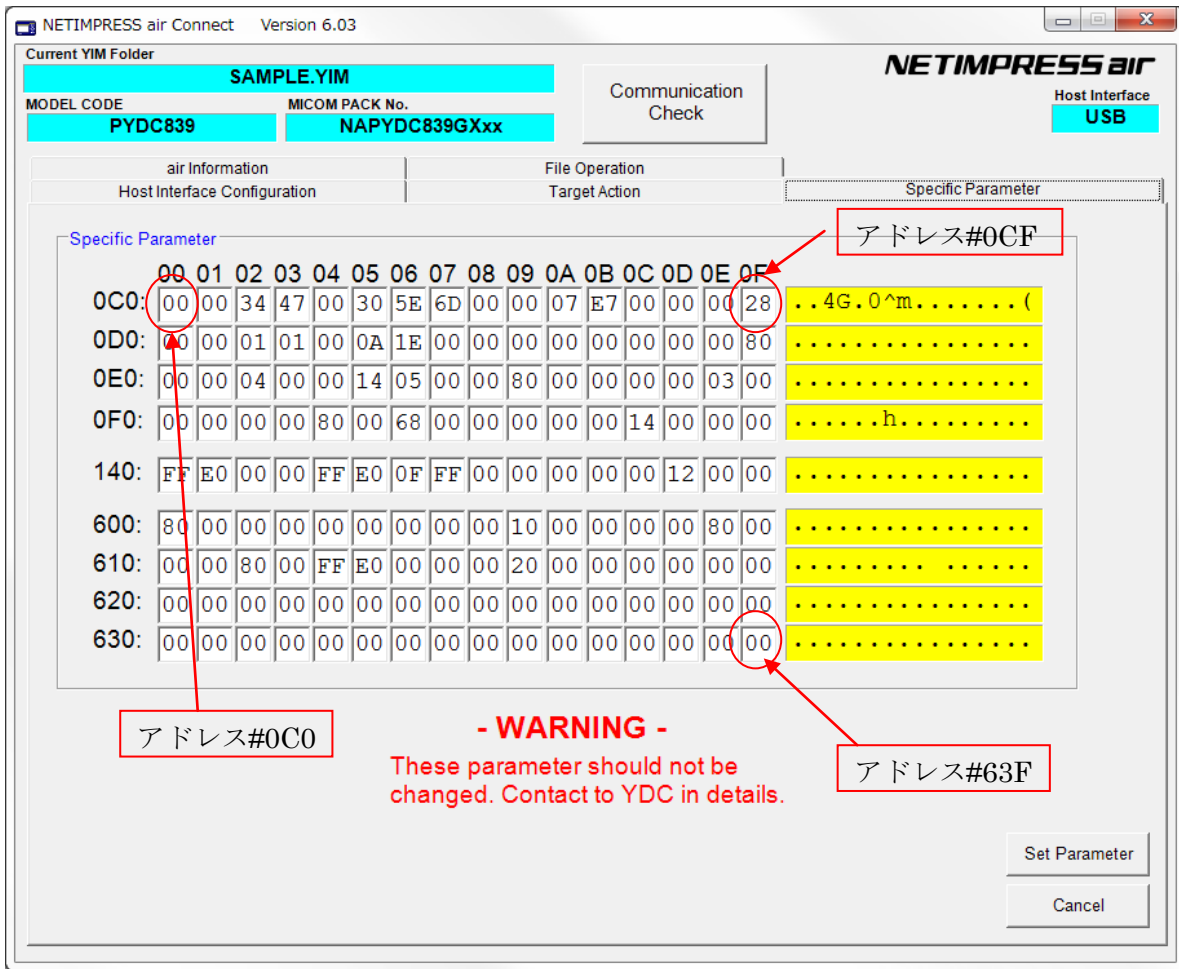
The screenshot shows the NETIMPRESS air Connect software interface. At the top, it displays 'Current YIM Folder: SAMPLE.YIM', 'MODEL CODE: PYDC839', and 'MICOM PACK No.: NAPYDC839GXX'. The 'Host Interface' is set to 'USB'. A 'Specific Parameter' table is visible, with columns labeled 00 through 0F. A 'Password' dialog box is open, with a red circle around the 'Password:' input field and a red arrow pointing to it from a text box. Below the dialog, a red warning message is displayed.

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
0C0:	00	00	34	47	00	30	5E	6D	00	00	07	E7	00	00	00	28	..4G.0^m.....(
0D0:	00	00	01	01	00	0A	1E	00	00	00	00	00	00	00	00	80
0E0:	00	00	04	00	00	14	05	00	00	80	00	00	00	00	03	00
0F0:	00	00	00	00	80	00	68									h.....
140:	FF	E0	00	00	FF	E0	0F									
600:	80	00	00	00	00	00	00									
610:	00	00	80	00	FF	E0	00									
620:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
630:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

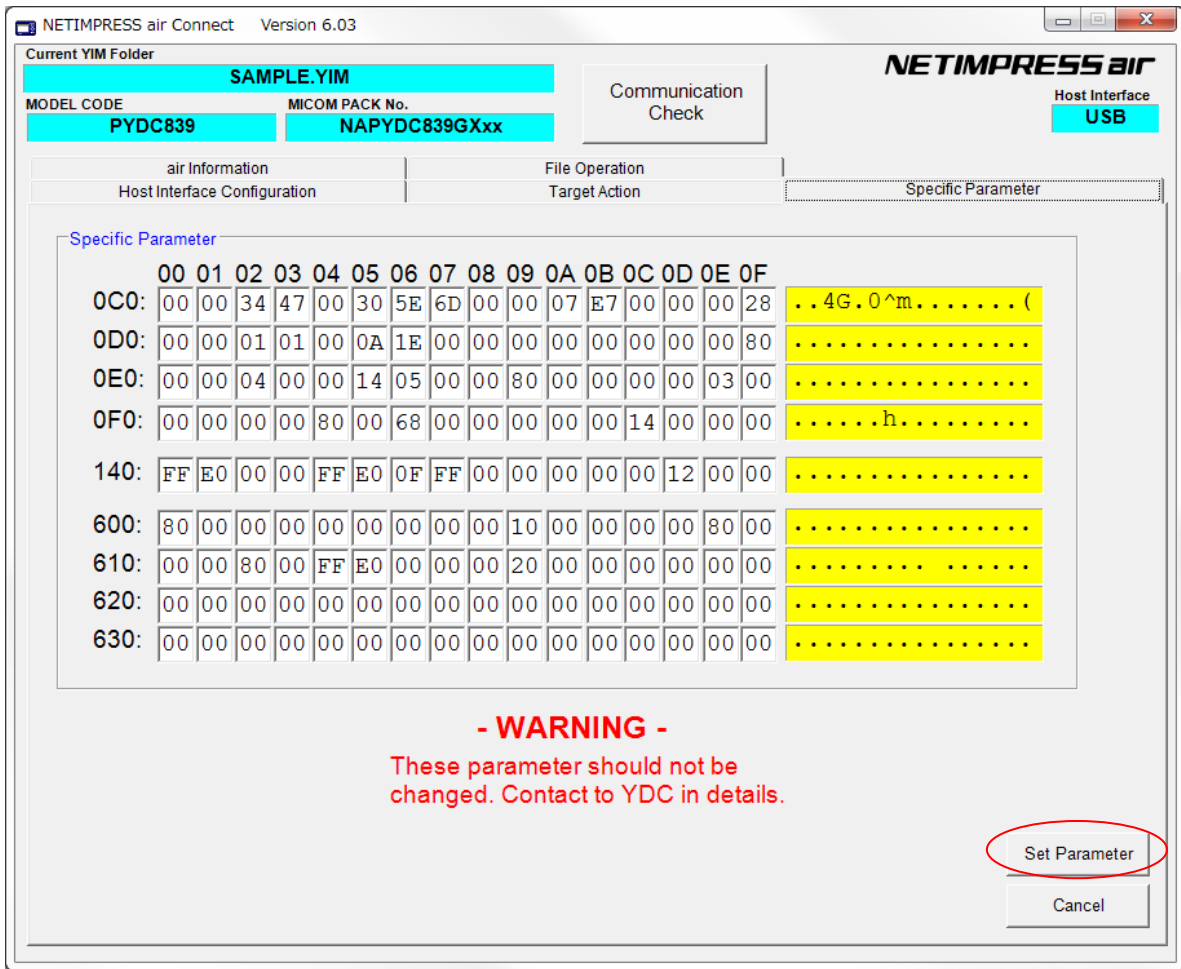
- WARNING -
These parameter should not be changed. Contact to YDC in details.

AF200 と入力します

Specific Parameter の画面が開きますので、任意のアドレスのデータを書き換えます。



パラメータを書き換えましたら、その内容を保存するために”Save Parameter”ボタンをクリックします。



以上で、Specific Parameter の値を設定することができます。

6. UCOPシステム概要

6. 1 イニシャル・プロセッシング・ルーチン (IPR)

(C 言語プログラム、リロケータブルオブジェクト)

ソースファイルは「user_ipr.c」として供給されます。

お客様サイドでカスタマイズして頂きます。

ターゲット MCU 実装前にシリアルライタ等で予め ROM の所定領域に書き込んでおきます。

リセット解除後リセットベクタよりジャンプします。

UCOP リプログラムモード実行に際して最低限必要なシステムの初期化を行うルーチンです。

アプリケーションプログラム実行時に初めて必要となる初期化ルーチンは、これとは別にアプリケーションプログラム中に専用初期化ルーチンを設けその中に配置してください。

項目	内容	ユーザ設定
IPR 配置アドレス	#000000E0h～#0000018Fh (176byte) (スタートアドレス:#000000E0h) ※1	不可
IPR 要初期化項目	■ 割り込み禁止 ■ スタック初期化 ■ クロック逡倍／分周設定	可

※1:リセット解除後、リセットベクタからこのアドレスへ飛びます。

6. 2 イニシャルブートローダ (IBL) (C 言語プログラム、リロケータブルオブジェクト)

ソースファイルは「y_ibl.c」として供給されます。

基本的にそのままでお使いいただけます。

ターゲット MCU 実装前にシリアルライタ等で予め ROM の所定領域に書き込んでおきます。

IPR よりコールされます。

UCOP リプログモード用の CAN 初期化設定、UCOP リプログモードエントリー、WCP の受信及び内蔵 RAM への書き込みを行います。

IBLプログラムを予めマイコンの下記アドレスに配置します。

項目	内容	ユーザ設定
YDC 製 IBL	#000190h~#000BFFh (約 2.6k)	不可

※ IBL ファイルとして供給されます。シリアルライタ等で所定の領域に書き込み後、MCU を実装して下さい。

※ IBL 突入時スタックはイニシャライズされます。

※ 標準の最適化を使用しております。

6. 3 書き込み制御プログラム (WCP)

(C 言語プログラム、リロケータブルオブジェクト)

ソースファイルは「y_wcp.c」として供給されます。

実行ファイルは拡張子が BTP のファイルとして供給されます。

コントロールモジュールの DOS 領域に配置してください。

ライタはあらかじめ ROM 内に組み込まれている IBL と通信を行い、BTP ファイルを順次送信します。

IBL はターゲットの内蔵 RAM へ受信した BTP ファイルを書き込みます。

IBLプログラムとの通信によりマイコンの下記アドレスに配置します。

項目	内容	ユーザ設定
WCP 転送アドレス	#0FE700h~#0FFAFFh (5kbyte)	

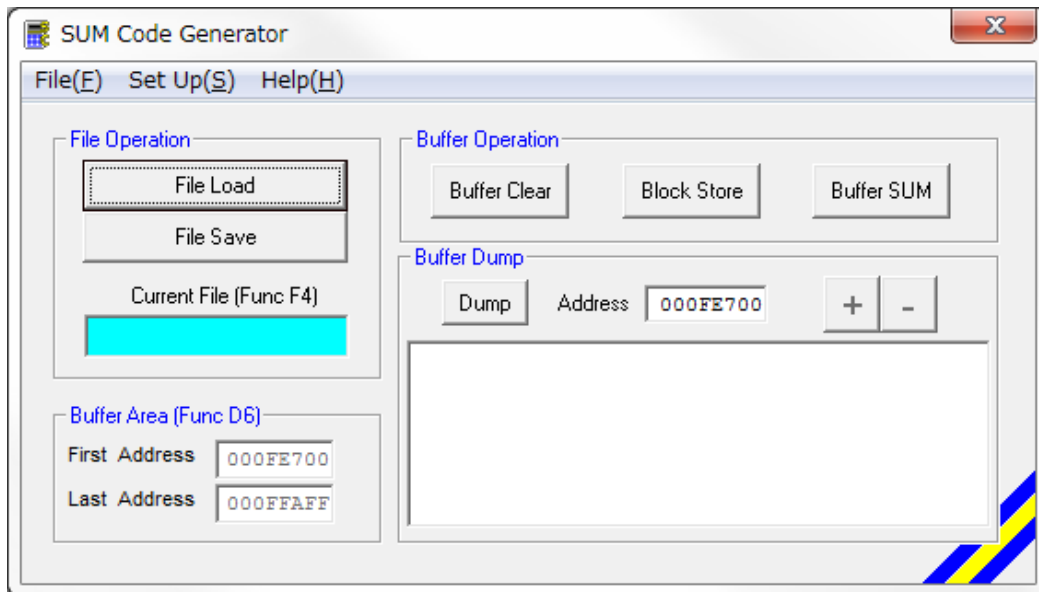
※ 標準の最適化を使用しております。

ソースをコンパイルすると「wcp.hex」ファイルが作成されます。

上記ファイルのフォーマットは INTEL HEX であるため、BTP ファイルのフォーマット形式(バイナリ)に変換する必要があります。

データのフォーマット変換には弊社ソフトの AZ286 をお使いください。

(AZ286 に関しては、弊社サポートにご連絡ください。)



AZ286 にて、バッファのアドレスを WCP 配置アドレスにします。

(今回の場合は#FE700～#FFAFF になります。)

その後、”wcp.hex”ファイルをロードし、セーブ時にバイナリ形式を選択して保存すれば、バイナリ形式のファイルが作成されます。

6. 4 書き込みプロセス正常終了判定

- ①. アプリケーションプログラム (APL) の一部領域を“ユーザアプリ領域サム値チェック”領域として使用し、既に正常な APL が存在しているか否かの判定をします。
- ②. 判定方法は、“ユーザアプリ領域サム値チェック”領域の SUM 値が #AA (8ビット単純加算8ビット比較) の場合、正常に APL が書き込まれていると判断します。
- ③. “ユーザアプリ領域サム値チェック”領域の SUM 値計算は IBL 中で実行され、正常な APL が書き込まれていれば APL へ JUMP し、そうでなければ IBL で Connect コマンド待ちになります。
- ④. 書き込み時にエラーが発生し、“ユーザアプリ領域 SUM 値チェック”領域のみ正常に書き込みが行われ、その他領域に正常データが書かれていない状態になるのを防ぐため、E.P.R 実行に際し下記のように動作します。
 - I. 1番最初に“ユーザアプリ領域サム値チェック”領域を含むブロックを消去します。
 - II. その他ブロックの消去をおこないます。
 - III. “ユーザアプリ領域サム値チェック”領域以外の領域にデータを書き込みます。
 - IV. 1番最後に“ユーザアプリ領域サム値チェック”領域にデータを書き込みます。

注意: i. 1つの消去ブロック内に収まるように設定してください。

ii. 書き込み禁止領域中には設定しないで下さい。

iii. 先頭のアドレスは、書き込みページの先頭(下位 8bit “00h”)とします。

ページ: データ転送のアラインメント単位をここではページと呼びます。

iv. ブートスワップモード時、ブートとして使用している領域を“ユーザアプリ領域サム値チェック”領域として設定しないでください。

項目	内容	ユーザ設定
“ユーザアプリ領域サム値チェック”領域	#002000h~#0023FFh (デフォルト※)	可

※初期設定ファイル(y_init.h)の APL_SUM_START と APL_SUM_END で設定してある値です。

※APL 領域範囲で変更可能です。

6. 5 アイデンティファイヤ(CAN メッセージ ID)

アイデンティファイヤ(以下:CAN メッセージ ID)としてデフォルトとして設定されるものを「Primary ID」と規定します。

※ メッセージ ID はここで規定した Primary ID 以外の他の ID へ変更することができます。

初期設定ファイルにて ID_P_NI, ID_P_MCU の値を変更したあと、コンパイルし、ブートスワップモードで IBL を書き換えてください。(5. 15 Primary ID 参照)

6. 5. 1 Primary ID

項目	内容	ユーザ設定
ライタ→マイコン	7ED	可
マイコン→ライタ	7EE	可

※Primary IDは初期設定ファイル(y_init.h)に登録します。

6. 5. 2 Secondary ID

このマイコンパックは Secondary ID 追加機能を持っておりません。ご注意ください。

6. 5. 3 送受信メッセージバッファ

IBLで使用するメッセージバッファについて規定します。

基本規定

- ・IBLは受信用メッセージバッファとして受信バッファ 15 を使用します。
- ・IBLは送信用メッセージバッファとして送信バッファ 3 を使用します。
- ・IBLは受信用メッセージバッファのマスクは使用しません。
- ・ID 完全一致として使用します。
- ・受信バッファ 0～14 は使用しません。
- ・割り込みは禁止です。

ユーザ APL の使用法

- ・IBL から APL へ移行した再は上記「IBL 規定」の状態です。
- 必要であればIBLで使用しているメッセージバッファの設定を変更し使用しても構いません。
ただし APL から IBL へ移行する場合(u Entry)、メッセージバッファを IBL 規定に戻し、
割り込みは禁止にしてから移行してください。

項目	内容	ユーザ設定
IBL 受信用 メッセージバッファ	受信バッファ 15	不可
IBL 送信用 メッセージバッファ	送信バッファ 3	不可

6. 6 ブートスワップ機能

WCP ではチップの持つブートスワップ機能を使用します。(ブートスワップモード時)

(ブートスワップ機能の詳細はマイコンのマニュアルを参照してください。)

そのためベクタテーブルや IBL,IPR の書き換えが可能です。

(書き込みオブジェクトファイルに IBL や IPR を含める必要があります。IBL や IPR を含めない場合、IBL 領域が書き換わってしまいますので、リプログラムができなくなりますのでご注意ください。)

非ブートスワップモード時にはベクタテーブルや IBL,IPR(ブロック 0~3)の書き換えができませんのでご注意ください。

ブートスワップモードと非ブートスワップモードの切り替えは初期設定ファイルによって行います。

(「5. 20 ブートスワップ機能設定フラグ」参照)

また、2つのモードでは使用するパラメータファイルが異なるのでご注意ください。

6.7 ステータスレジスタ

フラッシュメモリの動作状態やイレーズ、プログラムの正常/エラー終了時の状態を示します。

ステータスレジスタ(SRD)の内容により状態を判断します。

ステータスレジスタの内容をチェックする為、SRD エリアとして WCP 中で占有します。

<ステータスレジスタ(SRD)>

SRD の各ビット	ステータス名	定義	
		“1”	“0”
SR7(bit7)	コマンドビジー	レディ	ビジー
SR6(bit6)	イレーズステータス	エラー終了	正常終了
SR5(bit5)	プログラムステータス	エラー終了	正常終了
SR4(bit4)	リザーブ	---	---
SR3(bit3)	リザーブ	---	---
SR2(bit2)	リザーブ	---	---
SR1(bit1)	データ受信タイムアウト	タイムアウト	正常動作
SR0(bit0)	リザーブ	---	---

(a)SR7 (コマンドビジー)

- ・書き込み動作や消去動作中は”0”に、これらの動作終了とともに”1”にセットされます。

(b)SR6 (イレーズステータス)

- ・消去の動作状況を示し、消去エラーが発生すると”1”にセットされます。
このビットに一旦”1”がセットされると、クリアステータスレジスタ
コマンドを行わない限りリセット(“0”に書き換わる)されません。

(c)SR5 (プログラムステータス)

- ・書き込みの動作状況を示し、書き込みエラーが発生すると”1”にセット
されます。
・このビットに一旦”1”がセットされると、クリアステータスレジスタ
コマンドを行わない限りリセット(“0”に書き換わる)されません。

(d)SR1 (データ受信タイムアウト)

- ・データの受信中にタイムアウトが発生すると”1”にセットされます。
・このビットに一旦”1”がセットされると、クリアステータスレジスタ
コマンドを行わない限りリセット(“0”に書き換わる)されません。

<ステータスレジスタ 1(SRD1)>

- ・8bit で構成され、受信したコマンドフレームのコマンドがセットされます。

6. 8 プログラムエントリーモード

UCOP には次の 3 種のエントリーモードが存在します。

エントリーモード	概 要	使用方法
n Entry	“書き込みプロセス正常終了判定領域”の SUM 値が #AA 以外の場合のエントリーモード	標準
u Entry	“書き込みプロセス正常終了判定領域”の SUM 値が #AA の場合のエントリーモード ※APL上のCAN対応コマンドでエントリーします	標準
r Entry	“書き込みプロセス正常終了判定領域”の SUM 値が #AA かつ u Entry が不可能な場合のエントリーモード	非標準

※各エントリーのフローは「3. 3 プログラムエントリーモードフローチャート」を参照してください。

6.9 u Entry 時ユーザ APL 処理項目

項目	内容	ユーザ設定
APL ヘジャンプ後の 処理項目 ■: 必須 □: 選択	<ul style="list-style-type: none"> ■ CONNECT コマンド受信 (受信後 20ms 以内(※1)で IBL を CALL する) ※2 □ CAN メッセージボックスの設定変更 ■ CAN ボーレートの変更不可 	可
Connect コマンド受信後の APL 要初期化項目 (u Entry 時) ■: 必須	<ul style="list-style-type: none"> ■ CAN 受信バッファクリア(受信完了状態にする) ■ 割り込み禁止 ■ UCOP 用メッセージボックスの設定 ※3 	可

※1. CONNECT 応答規定は 25ms ですが IBL での応答までの処理が約 2ms 程要するため
20ms 以内程度で IBL を CALL して下さい

※2. APL がビジー状態で CONNECT できない場合、ビジー応答を返してください。

・ビジー応答規定

Data1	Data2	Data3	Data4	Data5	Data6	Data7	Data8
FEh	36h	CTR	/	/	/	/	/

CTR=00h 固定、斜線部は don't care です

ビジー応答があった場合 IMPRESS 側は CONNECT リトライを 200ms 間隔で 5 回まで行います。

リトライオーバーした場合“resource/function not available”エラーとなります。

尚、リトライ回数及びその間隔は設定が可能です。

項目	内容	ユーザ設定
リトライ回数	5 回	可
リトライ間隔	200ms	可

※3. ユーザアプリにて CAN のメッセージボックスを変更した場合、IBL や WCP にて CAN の送受信が
正しくできるよう、UCOP で使用する設定に戻してください。

6. 10 誤 Entry 時無限ループ防止機能

何らかの理由で誤って Entry を行った際の無限ループを防止する目的で IBL では規定ポイントにてタイムアウト処理を行います。タイムアウトした場合リセット処理を行います。

・タイムアウト処理を行うポイント

KILL レジスタ ON 時 :Connect 受信後、次コマンド待ち

KILL レジスタ OFF 時 :Connect 応答後、次コマンド待ち

項目	内容	ユーザ設定
タイムアウト値 n sec	3 sec	不可

6. 11 CAN ボーレート設定時の注意

- ・CAN 通信におけるボーレート設定は、MCUに対するレジスタ情報を使用します。
ビットレートプリスケアラレジスタ、ビットレートレジスタ、CAN クロック選択ビットの値を設定してください。
詳細は「5. 5 ビットレートプリスケアラレジスタ値」、「5. 6 ビットレートレジスタ値」、「5. 7 CAN クロック選択ビット値」を参照してください。
- ※コンパイル時に、指定されたレジスタ情報とボーレート値、CAN クロックが矛盾する場合にはコンパイルエラーとなります。

6. 12 ステーションアドレス

- ・16bit 構成で CCP プロトコルを使用します。
 - ・スレーブ側 (ECU) は初期設定ファイルにて設定します。
 - ・マスター側 (ライタ) は IMPRESS モジュールのパラメータファイルにて設定します。
 - ・スレーブ側 (ECU) はステーションアドレス不一致時、エラーを返さず引き続きコネク待ち状態となります。
- (ステーションアドレスの変更方法は「5. 17 ステーションアドレス」を参照してください。)

項目	内容	ユーザ設定
ステーションアドレス	#0000h	可

※Disconnect コマンドについては、MCU はその受信に際してステーションアドレスを無視します。

6. 13 プログラム終了時の処理

WCP、IBL のプログラム終了時の処理について正常終了時、異常終了時、共にメッセージ送信後に Disconnect コマンド待ちとなります。

<Disconnect コマンド受信時の動作>

I/O ポートサービスの停止処理を行いその後永久ループに入ります。

(I/O ポートサービス停止に伴う、MCU 外部からのリセット、または内蔵 WDT のリセットにより ECU のリスタートを行います)

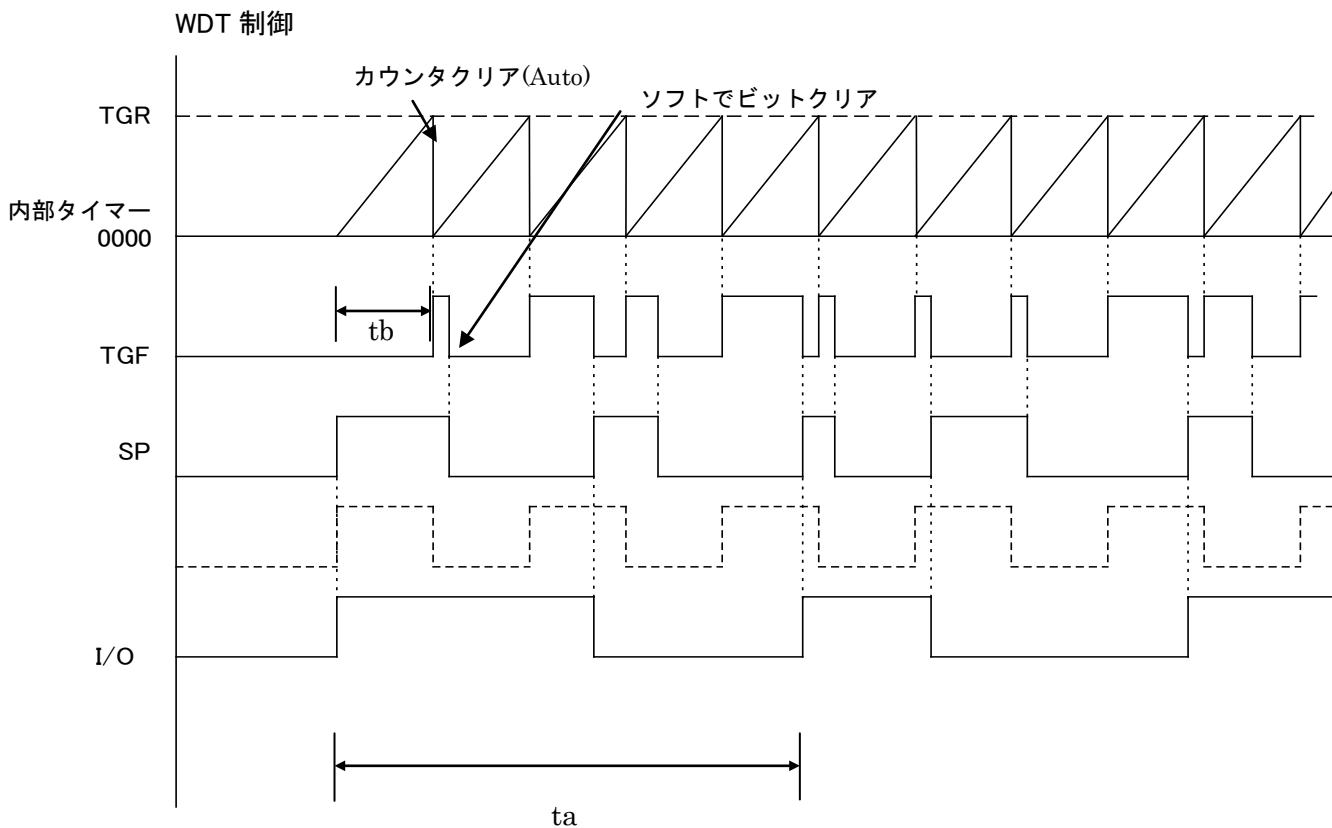
外部から Reset 割り込みが入らない場合、内蔵 WDT 未使用の場合は、電源断まで永久ループに入っています。

Reset 割り込みが入ったときは、外部回路を含めてリセットスタートします。

6. 14 ウォッチドッグタイマ

- ・ I/O ポートを制御することによりウォッチドッグタイマの制御を行います。
- ・ ビット単位の制御可能な出力ポートに対するサイクルアクセス機能を持ちます。
- ・ アクセス周期は初期設定ファイルに設定します。
- ・ 内部タイマーを使用します。(割り込みは使用しません)

※I/O ポートサービスの有無、アクセス周期変更等は「5. 12 I/O ポートサービス対応フラグ」・「5. 13 I/O ポートサービス周期」・「5. 14 I/O ポートサービス用ポート変更方法」を参照してください。



TGR: タイマカウンタ

TGF: タイマオーバーフロービット

SP: ソフトウェアパルス

I/O: I/O 出力パルス(ソフトウェアパルス2分周値~)

仕様

項目	内容	ユーザ設定
使用内蔵タイマ	<input type="checkbox"/> 無し <input checked="" type="checkbox"/> 有: タイマ番号 <u>TAU00</u>	不可
ta	ta=t1 + tb ms (デフォルト t1=8) t1...min 1ms (=tb×2 値) t1 の設定は1ms単位とする。(tb×2 値)	t1 設定可
tb	500 μ sec	
I/O ポート番号	Port <u>8</u> bit <u>0</u>	可

※uEntry時直ちに1ms周期固定で1回パルス出力し通常のパルス動作に移行します。

6. 15 IBL 処理時間

リセット後ユーザ IPR から IBL を CALL し APL SUM 域が正常だった場合の IPR へリターンするまでの処理時間です。

動作周波数 32MHz で測定

項目	内容	ユーザ設定
APL SUM 域 4Kbyte	17ms	不可
APL SUM 域 1byte	10ms	不可

7. r Entryモード仕様

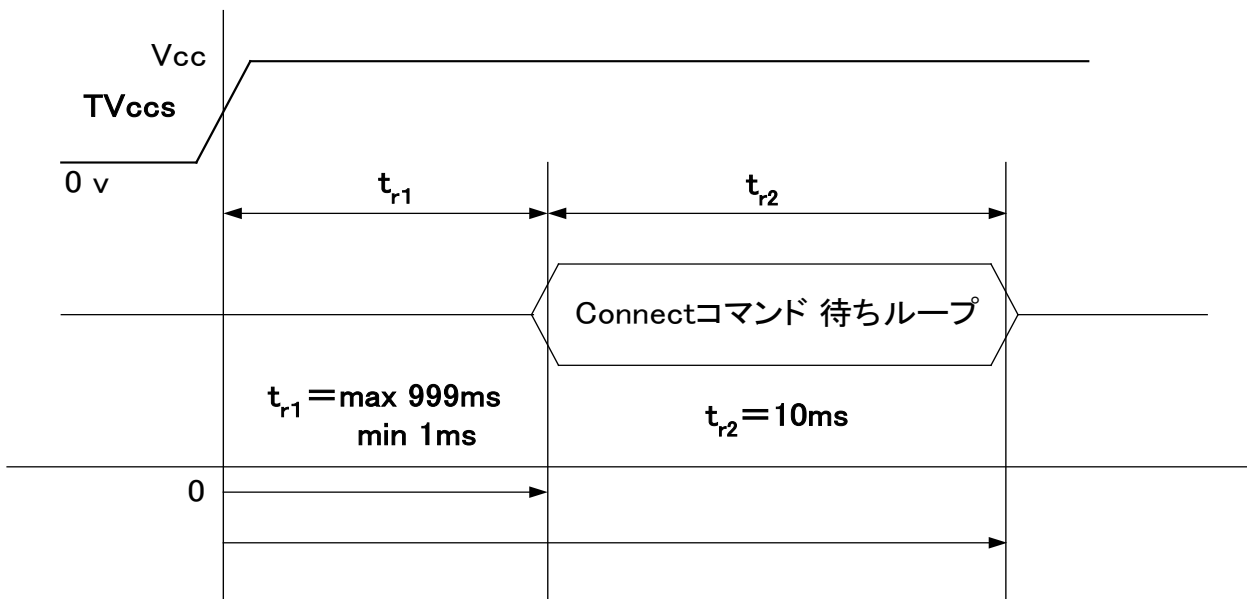
7.1 概要

r Entry は、ユーザアプリケーションが正常にかかっている状態(“書き込みプロセス正常終了判定領域”の SUM 値が#AA)で、u Entry が不可能な場合に使用します。

電源投入後、一定期間 t_{r1} (※)経過後、約 10mSec 間 Connect コマンドを待ちます。この約 10mSec 間に Connect コマンドを受信すると r Entry になります。

ライタ側はターゲットの電源を開始することにより、タイミングを合わせ r Entry モード期間に Connect コマンドを送信するようにします。

※この一定期間はターゲットに電源投入後 Connect コマンド受信待ちを開始するまでの時間で IPR の処理時間などお客様のシステム構成によって時間が変わってきます。



NI はゼロ基点で t_{r1} 後に CONNECT コマンドを発することで、“SUM 一致”の場合も、リプログモードに入り込めます。

※TVccsは電源監視用の信号線です。

項目	内容	ユーザ設定
t_{r1}	1~999ms	可
t_{r2}	10 ms	不可

7. 2 r Entry モード使用方法

r Entry モードの使用には、r Entry モード用マイコンパックが必要となります。

r Entry モード使用の際は、弊社サポートセンタまで、ご連絡ください。

使用手順は以下のようになります。

1. 電源監視用にターゲットマイコンの Vcc ラインに TVccs 信号線を必ず接続してください。
2. air Connect の Specific Parameter のアドレス #14F の値が「0x01」であることを確認してください。

Specific Parameter のアドレス	#14F
設定値	0x01

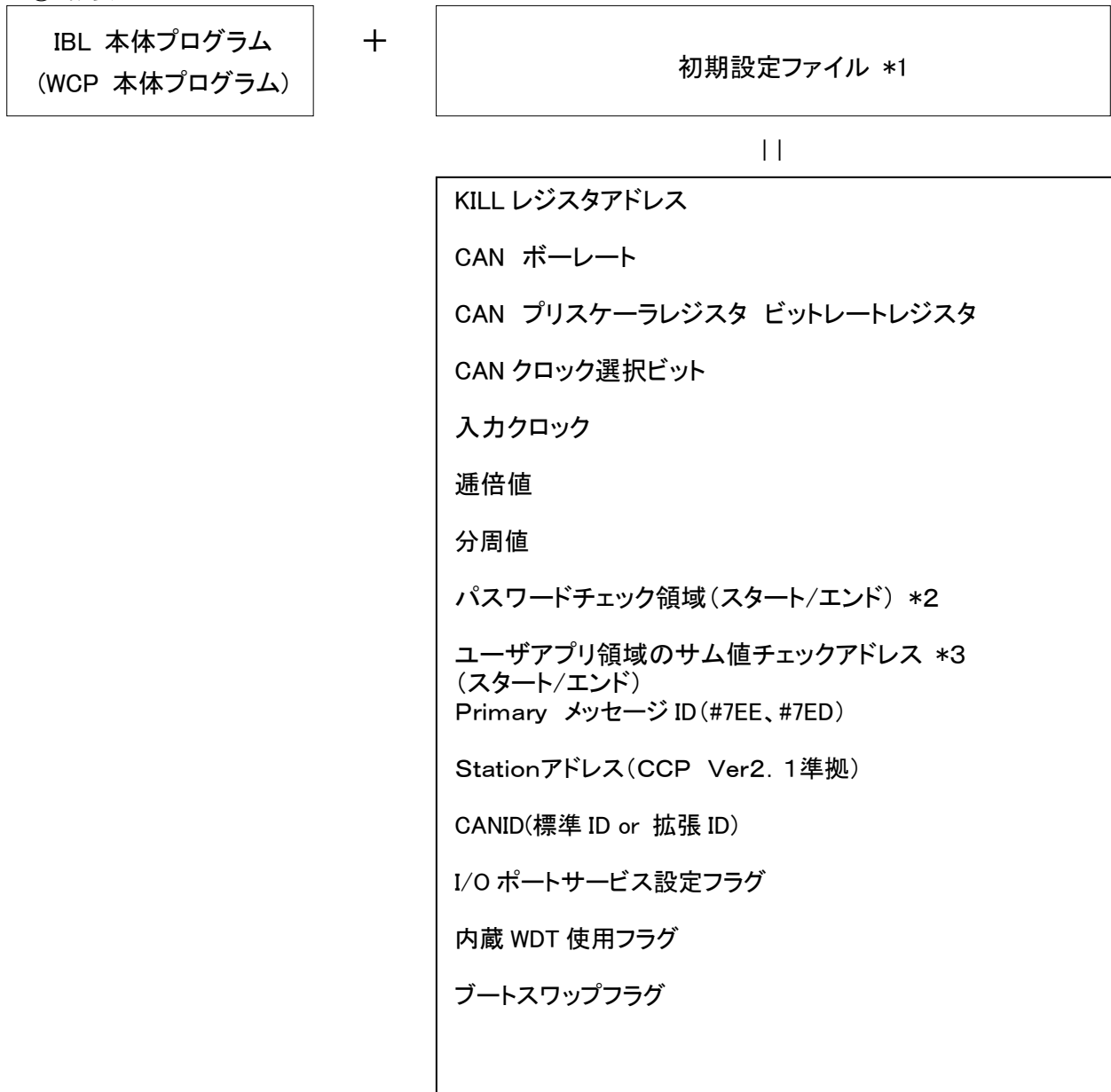
3. ターゲット側の t_{r1} 時間分タイミングを合わせるために、ライター側の Connect コマンド発行タイミングを air Connect の Specific Parameter のアドレス #14C, #14D の 2byte で設定してください。

例) t_{r1} 時間が 5mS の場合

Specific Parameter のアドレス	#14C	#14D
設定値	0x00	0x05

8. YDC製IBL、WCPの構成

① 概要



* 1 初期設定ファイルは、ヘッダファイルとして IBL、WCP にてインクルードされます。

* 2 パスワードチェック領域はユーザアプリ領域中の任意の領域を設定します。

項目	内容	ユーザ設定
パスワードチェック領域	#002F00h~#002FFFh	可

* 3 ユーザアプリ領域のサム値チェックスタートアドレスは、書き込みページの先頭(下位 8bit “00h”)とします。サイズは1byte~4Kbyte とします。

9. RAMの使用方法

書き込み実行後MCU上のRAM内容は保持されておられません。

10. CANプロトコル

UCOP で使用する各コマンドについての詳細は弊社ホームページに掲載しています「UCOP プログラミングコマンド・プロトコル仕様書」をご参照ください。

10.1 フレームの種類

UCOP ではデータフレームを次の6種類に分別して通信を行っています。

- ① コマンドフレーム
- ② ビジィフレーム
- ③ リードフレーム
- ④ ライトフレーム
- ⑤ レディフレーム
- ⑥ エラーフレーム

全てのフレームはスタンダード・フォーマットとエクステンデッド・フォーマットの2つのフレームフォーマットがあります。データフィールドは8バイトの固定長です。

各フレームの役割を下記に示します。

フレーム	フレームの役割
コマンドフレーム	実行するコマンドを指定するフレームです。ライタが MCU へ送信します。
ビジィフレーム	MCU がライタへ受け取ったコマンドを返すフレームです。
リードフレーム	ライタが MCU からデータを受信する為のフレームです。
ライトフレーム	ライタが MCU へデータを送信する為のフレームです。
ターミネータフレーム	ライトフレームが終了したことを知らせる為のフレームです。
レディフレーム	MCU がライタへ処理終了を知らせる為のフレームです。
エラーフレーム	1部のコマンド実行においてエラーが発生した場合や規定外のコマンドを受信した場合に MCU がライタへ送信するフレームです。

10.2 IBL 対応コマンド

IBLでは下記コマンドに対して応答する。

コマンド	内容
Connect	コネクト処理を行います。
Get CCP Version	CCP のバージョンをライタへ送信します。
Exchange ID	予め決められた ID をライタへ送信します。
Pass Word Check	パスワードチェックを行います。
Download	WCP プログラムを受信し内蔵 RAM へ書き込みます。
Extended Sum Read	内蔵 RAM へ書き込んだ WCP のサム値を計算しライタへ送信します。
Disconnect	ディスコネクト処理を行います。

上記以外のコマンドに対してはエラーフレームを発行します。

Disconnect コマンドに関しては常時受信可能とし、ツールからの指示でいつでも、Disconnect・リセットスタートを可能にします。

10.3 WCP 対応コマンド

WCP では下記コマンドに対して応答します。

コマンド	内容
Read	ROM からデータを読み出します。
Program	ROM にデータを書き込みます。
Block Erase	ブロック単位で ROM のデータを消去します。
Read SRD	ステータスレジスタの値をライターへ送信します。
Clear SRD	ステータスレジスタの値を初期化します。
Extended Blank Check	MCU 側でブランクチェックを行います。
Extended Sum Read	MCU 側で SUM 値を計算しライターへ送信します。
Disconnect	ディスコネクト処理を行います。

上記以外のコマンドに対してはエラーフレームを発行します。

Disconnect コマンドに関しては常時受信可能とし、ツールからの指示でいつでも、Disconnect・リセットスタートを可能にします。

11. 関数一覧

この章では、IBL, WCP 内で使用されている関数について説明しています。

ただし、お客様が変更された結果についての責任は、弊社では負いかねますので予めご了承ください。

また、弊社プログラム内ではデータサイズとデータ形式は下記のように取り決めています。

弊社プログラム内データ形式	データ形式	データサイズ
DWORD	unsigned long	32bit データ
WORD	unsigned short	16bit データ
BYTE	unsigned char	8bit データ

11. 1 IBLでの使用関数(y_ibl. cファイルの関数一覧)

- io_service(void) 関数

内容	I/O ポートサービス処理関数です。 引数を設定の上 io_sv 関数をコールします。
引数	なし
戻り値	なし

- io_sv(WORD *sp, WORD *sp_wdt) 関数

内容	I/O ポートサービス処理関数です。 I/O ポートサービスを行います。 内蔵 WDT のクリアを行います。
引数	*sp IO サービス用のソフトウェアパルスカウンタです。 設定したカウンタ値になるまでインクリメントします。 設定したカウンタ値になるとゼロに初期化します。 *sp_wdt 内蔵 WDT 用のソフトウェアパルスカウンタです。 設定したカウンタ値になるまでインクリメントします。 設定したカウンタ値になるとゼロに初期化します。
戻り値	なし

- init_tau00(void) 関数

内容	I/O ポートサービス用のタイマとポートの初期化を行います。
引数	なし
戻り値	なし

- init_can(void) 関数

内容	CAN 設定関数です。 CAN 用ポート、レジスタの初期化を行います。 送信バッファ 3 を送信用、受信バッファ 15 を受信用に設定します。
引数	なし
戻り値	なし

- RecData(void) 関数

内容	コマンドフレーム、ライトフレーム、ターミネータフレーム受信処理関数です。 ディスコネクトコマンドを受信した場合は応答を返し、リセット処理を行います。
引数	なし
戻り値	なし

- TrmData(void) 関数

内容	リードフレーム送信処理関数です。 CAN データの送信を行います。
引数	なし
戻り値	なし

- TrmReady(void) 関数

内容	レディフレーム送信処理関数です。 レディフレームを送信します。
引数	なし
戻り値	なし

- TrmBusy(void) 関数

内容	ビジィフレーム送信処理関数です。 ビジィフレームを送信します。
引数	なし
戻り値	なし

- TrmError(void) 関数

内容	エラーフレーム送信処理関数です。 エラーフレームを送信します。
引数	なし
戻り値	なし

- exec_reset(void) 関数

内容	リセット実行処理関数です。 I/O ポートサービス用タイマのカウントをストップし、無限ループに入ります。
引数	なし
戻り値	なし

- command_GetCcpVersion(void) 関数

内容	CCP バージョン取得処理関数です。 取得したバージョンが「2.1」かどうか判定します。
引数	なし
戻り値	取得したバージョンが「2.1」かどうか判定し 戻り値として、「2.1」の場合 OK(= 0)、「2.1」以外の場合 NG(= -1)を返します。

- command_ExchangeID(void) 関数

内容	Exchange_ID 処理関数です。 予め決められた ID をライター側に送信します。
引数	なし
戻り値	戻り値は、OK(= 0)を返します。

- command_password_check(void) 関数

内容	パスワードチェック処理関数です。 ライターから送られてきた ID と、フラッシュメモリの内容が同じかを判定します。
引数	なし
戻り値	戻り値は、正常の場合 OK(= 0)、エラーの場合 NONE(= 1)を返します。

- command_ExtSumcheck(void) 関数

内容	拡張サムチェック処理関数です。 8bit 単純加算したサム値と 16bit 単純加算したサム値をライター側に返します。
引数	なし
戻り値	なし

- get_wcp(WORD* paddr) 関数

内容	W.C.P データ受信処理関数です。 W.C.P データを受信し内蔵 RAM へ書き込みます。 拡張 SUM チェックコマンドを受信したら、command_ExtSumcheck関数を Call します。
引数	* paddr 内蔵 RAM ジャンプ先アドレスです。
戻り値	正常の場合 OK(= 0)、コマンド異常の場合 NG(= -1)を返します。

・ RecConnect(BYTE tout) 関数

内容	コネクトコマンド受信処理関数です。 r Entry, n Entry においてコネクトコマンドを受信します。
引数	tout コネクトコマンド受信においてタイムアウトの有無を設定します。 = 1 : タイムアウト有り(10ms) = 0 : タイムアウト無し * piosdiv IO サービス用のソフトウェアパルスカウンタです。 * pwdtdiv 内蔵 WDT 用のソフトウェアパルスカウンタです。
戻り値	コネクトコマンドを受信した場合「1」、コネクトコマンドを受信しなかった場合「0」を返します。

・ ibl_main(void) 関数

内容	iblメイン処理関数です。 IPRからコールされるメインルーチンです。 各種初期設定と、コネクトコマンド受信処理を行います。
引数	なし
戻り値	戻り値「1」の場合、r Entry または n Entry で UCOP リプログラムモードへ遷移します。 戻り値「0」の場合、APLへ制御が移ります。

・ ibl_entry(WORD ent) 関数

内容	エントリー処理関数です。 r Entry, n Entry, u Entry 後のメインルーチンです。
引数	ent UCOP リプログラムモードへエントリーしたエントリーモードを判定します。 = 1 : rEntry, nEntry = 0 : uEntry
戻り値	なし

11.2 WCPでの使用関数(y_wcp. cファイルの関数一覧)

- io_service(void) 関数

内容	I/O ポートサービス処理関数です。 引数を設定の上 io_sv 関数をコールします。
引数	なし
戻り値	なし

- io_sv(WORD *sp, WORD *sp_wdt) 関数

内容	I/O ポートサービス処理関数です。 I/O ポートサービスを行います。 内蔵 WDT のクリアを行います。
引数	*sp IO サービス用のソフトウェアパルスカウンタです。 設定したカウンタ値になるまでインクリメントします。 設定したカウンタ値になるとゼロに初期化します。 *sp_wdt 内蔵 WDT 用のソフトウェアパルスカウンタです。 設定したカウンタ値になるまでインクリメントします。 設定したカウンタ値になるとゼロに初期化します。
戻り値	なし

- RecData(void) 関数

内容	コマンドフレーム、ライトフレーム、ターミネータフレーム受信処理関数です。 ディスコネクトコマンドを受信した場合は応答を返し、リセット処理を行います。
引数	なし
戻り値	なし

- TrmData(void) 関数

内容	リードフレーム送信処理関数です。 CAN データを送信します。
引数	なし
戻り値	なし

- TrmReady(void) 関数

内容	レディフレーム送信処理関数です。 レディフレームを送信します。
引数	なし
戻り値	なし

- TrmBusy(void) 関数

内容	ビジィフレーム送信処理関数です。 ビジィフレームを送信します。
引数	なし
戻り値	なし

- TrmError(void) 関数

内容	エラーフレーム送信処理関数です。 エラーフレームを送信します。
引数	なし
戻り値	なし

- exec_reset(void) 関数

内容	リセット実行処理関数です。 I/O ポートサービス用タイマのカウントストップを行い、無限ループに入ります。
引数	なし
戻り値	なし

- Command_Read(void) 関数

内容	リードコマンド処理関数です。 読み出したデータをライタ側に送信します。
引数	なし
戻り値	なし

- Command_Program(void) 関数

内容	プログラムコマンド処理関数です。 書き込み先アドレスと書き込みデータを受信します。 ページ単位で書き込みを行います。 ブート領域書き込み時にはブートスワップ機能を使用します。
引数	なし
戻り値	なし

- Command_BlockErase(void) 関数

内容	ブロックイレーズコマンド処理関数です。 対象アドレスから計算したブロックの消去を行います。
引数	なし
戻り値	なし

- Command_ReadSRD(void) 関数

内容	リードステータスレジスタコマンド処理関数です。 ステータスレジスタの値をライタ側に送ります。
引数	なし
戻り値	なし

- Com_ClearSRD(void) 関数

内容	クリアステータスレジスタコマンド処理関数です。 ステータスレジスタを初期化します。
引数	なし
戻り値	なし

- Command_ExtBlankcheck(void) 関数

内容	拡張ブランクチェックコマンド処理関数です。 指定範囲のブランクチェックを実行し、結果をライタ側に返します。
引数	なし
戻り値	なし

- Command_ExtSumcheck(void) 関数

内容	拡張サムチェック処理関数です。 8bit 単純加算したサム値と 16bit 単純加算したサム値をライタ側に返します。
引数	なし
戻り値	なし

- Command_KILLwrite(void) 関数

内容	KILL レジスタ設定処理関数です。 KILL レジスタアドレスに値を書き込み、エントリできないようにします。 また、書き込み結果をライタに返します。
引数	なし
戻り値	なし

- wcp_main(void) 関数

内容	WCPのメイン関数です。 ライタからのコマンドを受信し、各関数を Call します。
引数	なし
戻り値	なし

- flash_enter(void) 関数

内容	フラッシュ環境開始関数です。 コードフラッシュセルフプログラミングモードに遷移します。
引数	なし
戻り値	コードフラッシュセルフプログラミングモード移行結果

- flash_exit(void) 関数

内容	フラッシュ環境終了関数です。 コードフラッシュセルフプログラミングモードから抜けます。
引数	なし
戻り値	なし

- dflash_enter(void) 関数

内容	フラッシュ環境開始関数です。 データフラッシュセルフプログラミングモードに遷移します。
引数	なし
戻り値	データフラッシュセルフプログラミングモード移行結果

- dflash_exit(void) 関数

内容	フラッシュ環境終了関数です。 データフラッシュセルフプログラミングモードから抜けます。
引数	なし
戻り値	なし

- exec_cblank(fsl_u16 fsl_blocknum) 関数

内容	コードフラッシュに対するブランクチェック処理実行関数です。 ブランクチェックコマンドの発行とステータスチェックを行います。
引数	fsl_blocknum ブランクチェックを実行するブロック番号
戻り値	ブランクチェック結果

- exec_cerase(fsl_u16 fsl_blocknum) 関数

内容	コードフラッシュに対するブロックイレーズ処理実行関数です。 イレーズコマンドの発行とステータスチェックを行います。
引数	fsl_blocknum イレーズを実行するブロック番号
戻り値	イレーズ結果

- exec_cprogram(fsl_u32 fsl_addr, fsl_u08 fsl_word, _near fsl_u08 *fsl_bufaddr) 関数

内容	コードフラッシュに対する書き込み処理実行関数です。 書き込みコマンドの発行とステータスチェックを行います。 書き込み成功時は、内部ベリファイコマンドの発行とステータスチェックも行います。
引数	fsl_addr 書き込み先頭アドレス fsl_word 書き込みワード数(1ワード=4バイト) * fsl_bufaddr 書き込みデータ格納バッファ先頭アドレス
戻り値	書き込み結果

- exec_cbootswap(void) 関数

内容	コードフラッシュに対するブートスワップ処理実行関数です。 ブートスワップコマンドの発行とステータスチェックを行います。 この関数を実行すると、リセット動作なしでブートクラスタが入れ替わられます。
引数	なし
戻り値	ブートスワップ結果

- exec_cstatus(DWORD us) 関数

内容	コードフラッシュに対するステータスチェック処理実行関数です。 BUSY 以外のステータスとなるまで、指定されたインターバルでコードフラッシュ に対してステータスチェックを実行します。
引数	us ステータスチェックのインターバル時間(us 単位)
戻り値	ステータスチェック結果

- exec_dblank(pfdl_u16 pfdl_addr, pfdl_u16 pfdl_size) 関数

内容	データフラッシュに対するブランクチェック処理実行関数です。 ブランクチェックコマンドの発行とステータスチェックを行います。
引数	pfdl_addr ブランクチェック開始アドレス pfdl_size ブランクチェックサイズ
戻り値	ブランクチェック結果

- exec_derase(pfdl_u16 pfdl_block) 関数

内容	データフラッシュに対するブロックイレーズ処理実行関数です。 イレーズコマンドの発行とステータスチェックを行います。
引数	pfdl_block イレーズを実行するブロック番号
戻り値	イレーズ結果

- exec_dprogram(pfdl_u16 pfdl_addr, pfdl_u16 pfdl_size, _near pfdl_u08 *pfdl_bufaddr) 関数

内容	データフラッシュに対する書き込み処理実行関数です。 書き込みコマンドの発行とステータスチェックを行います。 書き込み成功時は、内部ベリファイコマンドの発行とステータスチェックも行い ます。
引数	pfdl_addr 書き込み先頭アドレス pfdl_size 書き込みバイト数 * pfdl_bufaddr 書き込みデータ格納バッファ先頭アドレス
戻り値	書き込み結果

- exec_dread(pfdl_u16 pfdl_addr, pfdl_u16 pfdl_size, _near pfdl_u08 *pfdl_bufaddr) 関数

内容	データフラッシュからのデータ読み出し処理実行関数です。
引数	pfdl_addr 読み出し先頭アドレス pfdl_size 読み出しバイト数 * pfdl_bufaddr 読み出しデータ格納バッファ先頭アドレス
戻り値	データ読み出し結果

- exec_dstatus(DWORD us) 関数

内容	データフラッシュに対するステータスチェック処理実行関数です。 BUSY以外のステータスとなるまで、指定されたインターバルでデータフラッシュ に対してステータスチェックを実行します。
引数	us ステータスチェックのインターバル時間(us 単位)
戻り値	ステータスチェック結果

- add_get (DWORD *addr) 関数

内容	CAN 受信データからアドレス情報を取得します。 4バイトアドレスを取得します。
引数	* addr 取得したアドレス格納先
戻り値	なし

- add_get_ext (DWORD *faddr, DWORD *laddr) 関数

内容	CAN 受信データからアドレス情報を取得します。 開始アドレスと終了アドレスを取得します。 開始アドレスの最下位1バイトは 0x00 固定です。 終了アドレスの最下位1バイトは 0xFF 固定です。
引数	* faddr 取得した開始アドレス格納先 * laddr 取得した終了アドレス格納先
戻り値	なし

- timer_wait(DWORD us) 関数

内容	指定時間のウエイトを実行します。 ウエイト時間のカウントにタイマアレイユニット0のチャンネル7を使用します。
引数	us ウエイト時間(us 単位)
戻り値	なし

12. 使用I/Oリソース一覧

項目	リソース	備考	ユーザ設定
サイクルアクセスポート	Port_8 bit_0	I/O サービスに使用	可
MCU 内蔵タイマ	タイマ TAU00	サイクルアクセス用	不可

13. 付録

① 初期設定ファイル

初期設定ファイル名: y_init.h

項目	初期設定 ファイル定義名	内容	デフォルト値	ユーザ 設定
KILL レジスタアドレス	KILL_ADDR	32bit アドレス指定	0x00000B00	
CAN ボーレート	CAN_BAUD	125~1000 (Kbps 単位) 1000 : 1M 500 : 500K 250 : 250K 125 : 125K	500 (500Kbps)	可
CAN ビットレート プリスケアラレジスタ	CAN_C0CFGL_D ATA	ビットレートプリスケアラレジスタ値 16BIT	0x0001	可
CAN ビットレートレジスタ	CAN_C0CFGH_D ATA	ビットレートレジスタ値	0x0049	可
CAN クロック選択ビット	CAN_GCFGL_DC S	GCFGL レジスタの DCS ビット値	0	可
入カクロック	CLK_EXT	MHz×10で指定 例) 10MHZ = 100	40 (4MHz)	可
逡倍値	CLK_MUL	1~n	8	可
分周値	CLK_DIV	1~n	1	可
パスワードチェック領域 (スタート)	PASS_START	32bit アドレス指定	0x00002F00	可
パスワードチェック領域 (エンド)	PASS_END	32bit アドレス指定	0x00002FFF	可
ユーザアプリ領域の サム値チェックアドレス (スタート)	APL_SUM_STAR T	32bit アドレス指定	0x00002000	可
ユーザアプリ領域の サム値チェックアドレス (エンド)	APL_SUM_END	32bit アドレス指定	0x000023FF	可
I/O ポートサービス 対応フラグ	IOS_ON	0 or 1 0: I/O サービス無し 1: I/O サービス有り	1	可
I/O ポートサービス 周期	IOS_PERIOD	ms 単位(1ms~65535ms)	8	可
Primary メッセージ ID (ライター→マイコン)	ID_P_NI	32bit 値 bit0~bit10: 標準 ID bit11~bit28: 拡張 ID bit31: ID のフォーマット(0:標準、1:拡張)	0x000007ED	可
Primary メッセージ ID (マイコン→ライター)	ID_P_MCU	32bit 値 bit0~bit10: 標準 ID bit11~bit28: 拡張 ID bit31: ID のフォーマット(0:標準、1:拡張)	0x000007EE	可
Stationアドレス	CCP_STATION	16bit 値	0x0000	可
内蔵 WDT 対応フラグ	WDT_ON	0 or 1 0: 内蔵 WDT 未使用 1: 内蔵 WDT 使用	0	可
内蔵 WDT クリア周期	WDT_PERIOD	ms 単位	1	可
ブートスワップ対応フラグ	BOOT_SP	0 or 1 0: 非ブートスワップモード 1: スワップモード	1	可
I/O ポートサービスポートレ ジスタ	IOS_P_PORT	ポートレジスタのアドレスを指定	0x0FFF08	可
I/O ポートサービスポート方 向レジスタ	IOS_PD_PORT	ポート方向レジスタのアドレスを指定	0x0FFF28	可
I/O ポートサービスビット	IOS_BIT	I/O ポートサービスに使用するビットを 1 に設定	0x01	可

②リプログラム時 NET IMPRESS 設定項目(本定義体固有項目)

項目	設定方法	内容	デフォルト値	ユーザ設定
CAN ID(ライター→マイコン)フォーマット設定 *1	FUNC 81	“STANDERD”/“EXTENDED”を選択	STANDARD	可
CAN ID(マイコン→ライター)フォーマット設定 *1	FUNC 82	“STANDERD”/“EXTENDED”を選択	STANDARD	可
CAN 通信ボーレート設定 *1	FUNC 83	500Kbps/1Mbps/250Kbps/125Kbps を選択	500K	可
CAN ID 設定 *1	FUNC 86	ライター→マイコン、マイコン→ライターの標準/拡張ID 設定	ライター→マイコン 標準:#7ED 拡張:#00000 マイコン→ライター 標準:#7EE 拡張:#00000	可
KILL レジスタ ON	FUNC 87	KILL レジスタを ON に設定します		
ブランクチェックモード設定	FUNC 88	“NORMAL BLANK” (NI で読み出しチェック) “EXTENDED BLANK” (マイコン側でチェック)	EXTENDED BLANK	可
パスワードチェック領域	.KEY ファイルにて指定	初期設定ファイルで指定した、パスワードチェック領域スタート～エンドの範囲内に 7byte～255byte で指定		可
CONNECT Station Address *1	Specific Parameter #0D8	AZ990 等で Specific Parameter #0D8 から 2BYTE 指定する	#0000	可
ユーザアプリ領域サムチェックスタートアドレス *1	Specific Parameter #140	AZ990 等で Specific Parameter #140 から 4BYTE 指定する	#00002000	可
ユーザアプリ領域サムチェックエンドアドレス *1	Specific Parameter #144	AZ990 等 Specific Parameter #144 から 4BYTE 指定する	#000023FF	可
r Entry コネクト発行タイミング(5 章 t1)	Specific Parameter #14C	AZ990 等で Specific Parameter #14C から 2BYTE 指定する 1～999ms(1ms 単位)	#0A (10ms)	可
Entry モード	Specific Parameter #14F	AZ990 等で Specific Parameter #14F から 1BYTE 指定する 00: n Entry モード 01: r Entry モード	#00	可
CONNECT ビジー(36h) 応答時リトライ周期	Specific Parameter #E5	AZ990 等で Specific Parameter #E5 から 1BYTE 指定する 0～2550ms(10ms 単位)	#14 (200ms)	可
CONNECT ビジー(36h) 応答時リトライ回数	Specific Paramete #E6	AZ990 等で Specific Parameter #E6 から 1BYTE 指定する 0～255 回	#05 (5 回)	可

* 1 初期設定ファイルと同期をとる項目です。