

# NAPYDC839GN7y共通CANプログラミング技術資料

※y = 0~3

NO	マイコンパック型名	対応 MCU 名
1	NAPYDC839GN70	μ PD70F3504・μ PD70F3505A
2	NAPYDC839GN71	μ PD70F3506・μ PD70F3507
3	NAPYDC839GN72	μ PD70F3508・μ PD70F3509 μ PD70F3598
4	NAPYDC839GN73	μ PD70F3503

## <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

NAPYDC839GN7y の適用 NETIMPRESS シリーズ 本体は、NETIMPRESS air(AF930)です。

C<sup>o</sup>arNETIMPRESS,G-NETIMPRESS,NETIMPRESS next ではご使用になれません。

作成日:2015年08月12日 初版  
エンベデッドソリューション事業本部 開発部

## 変更履歴

変更日付	変更内容
2015.08.12	新規作成

# 目次

表紙.....	0
変更履歴.....	1
目次.....	2
1. 概要.....	6
1.1 動作条件.....	6
1.2 マイコンパック内容.....	7
2. 用語の定義と略語.....	8
3. 機能概要.....	12
3.1 UCOPシステム構成図.....	12
3.2 ROM.....	13
3.2.1 ROM構成図.....	13
3.2.2 消去ブロックアドレス(ブートスワップモード).....	14
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).....	19
3.3 プログラムエントリーモードフローチャート.....	25
3.4 IBLプログラム概略フローチャート.....	26
4. 初期導入手順.....	27
4.1 書き込み手順フロー.....	27
4.2 設定変更項目.....	28
4.2.1 CANボーレートの変更.....	28
4.2.2 動作クロックの変更.....	28
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.....	28
4.2.4 パスワードチェック領域の変更.....	28
4.2.5 ユーザアプリ領域サム値チェック領域の変更.....	28
4.2.6 ウォッチドッグタイマサービスの変更.....	28
4.2.7 Primary IDの変更.....	29
4.2.8 ステーションアドレスの変更.....	29
4.2.9 CANチャンネルの変更.....	29
4.2.10 内蔵ウォッチドッグタイマの変更.....	29
4.2.11 ブートスワップ機能の変更.....	29
4.2.12 KILLレジスタアドレス設定.....	29
5. UCOP設定変更方法.....	30
5.1 CANボーレート.....	30
5.2 入力クロック周波数.....	31
5.3 クロック逡倍比.....	31
5.4 クロック分周比.....	32
5.5 グローバルクロック選択レジスタ値.....	32
5.6 モジュールビットレートレジスタ値.....	32

5. 7	パスワードチェック領域開始アドレス	33
5. 8	パスワードチェック領域終了アドレス	33
5. 9	ユーザアプリ領域サム値チェック開始アドレス	33
5. 10	ユーザアプリ領域サム値チェック終了アドレス	34
5. 11	I/Oポートサービス対応フラグ	34
5. 12	I/Oポートサービス周期	35
5. 13	I/Oポートサービス用ポート変更方法	35
5. 14	Primary ID	36
5. 15	CAN IDフォーマット設定	39
5. 16	ステーションアドレス	39
5. 17	CANチャンネル番号	40
5. 18	内蔵ウォッチドッグタイマ設定	40
5. 19	内蔵ウォッチドッグタイマクロック設定	40
5. 20	ブートスワップ機能設定フラグ	40
5. 21	KILLレジスタアドレス設定	41
5. 22	Specific Parameter変更方法	41
6.	UCOPシステム概要	45
6. 1	イニシャル・プロセッシング・ルーチン (IPR)	45
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)	46
6. 3	書き込み制御プログラム (WCP)	46
6. 4	書き込みプロセス正常終了判定	48
6. 5	アイデンティファイヤ (CANメッセージID)	49
6. 5. 1	Primary ID	49
6. 5. 2	Secondary ID	49
6. 5. 3	送受信メッセージバッファ	49
6. 5. 4	チャンネル	50
6. 6	ブートスワップ機能	50
6. 7	ステータスレジスタ	51
6. 8	プログラムエントリモード	52
6. 9	u Entry 時ユーザAPL処理項目	53
6. 10	KILLレジスタ	54
6. 11	誤Entry時無限ループ防止機能	54
6. 12	CANポーレート設定時の注意	55
6. 13	ステーションアドレス	55
6. 14	プログラム終了時の処理	55
6. 15	ウォッチドッグタイマ	56
6. 16	IBL処理時間	57
7.	r Entryモード仕様	58
7. 1	概要	58
7. 2	r Entryモード使用方法	59

8. YDC製IBL、WCPの構成 .....	60
9. RAMの使用方法 .....	61
10. CANプロトコル .....	62
10. 1 フレームの種類 .....	62
10. 2 IBL対応コマンド .....	62
10. 3 WCP対応コマンド .....	63
11. 関数一覧 .....	64
11. 1 IBLでの使用関数(y_ibl. cファイルの関数一覧) .....	64
▪ io_service(volatile WORD * sp) 関数 .....	64
▪ init_timer(void) 関数 .....	64
▪ init_mbuf(void) 関数 .....	64
▪ RecData(void) 関数 .....	64
▪ TrmData(BYTE type) 関数 .....	65
▪ TrmReady(void) 関数 .....	65
▪ TrmBusy(void) 関数 .....	65
▪ TrmError(void) 関数 .....	65
▪ exec_reset(void) 関数 .....	65
▪ command_GetCcpVersion(void) 関数 .....	65
▪ command_ExchangeID(void) 関数 .....	65
▪ command_password_check(void) 関数 .....	66
▪ command_ExtSumcheck(void) 関数 .....	66
▪ get_wcp(DWORD * paddr) 関数 .....	66
▪ RecConnect(BYTE tout, WORD * piosdiv) 関数 .....	66
▪ init_CAN(void) 関数 .....	66
▪ uApli_Sum(void) 関数 .....	66
▪ ibl_main(void) 関数 .....	67
▪ ibl_entry(WORD ent) 関数 .....	67
11. 2 WCPでの使用関数(y_wcp. cファイルの関数一覧) .....	68
▪ io_service(volatile WORD * sp) 関数 .....	68
▪ SetFLMDHigh(void) 関数 .....	68
▪ SetFLMDLow(void) 関数 .....	68
▪ RecData(void) 関数 .....	68
▪ TrmData(BYTE type) 関数 .....	68
▪ TrmReady(void) 関数 .....	68
▪ TrmBusy(void) 関数 .....	68
▪ TrmError(void) 関数 .....	69
▪ exec_reset(void) 関数 .....	69
▪ flash_enter(void) 関数 .....	69
▪ flash_exit(void) 関数 .....	69
▪ get_block(u32 addr) 関数 .....	69

▪ Command_Read(void) 関数 .....	69
▪ recvdata(void) 関数 .....	69
▪ Command_Program(void) 関数 .....	70
▪ set_boot_info(void) 関数.....	70
▪ erase_old_boot(void) 関数.....	70
▪ Command_BlockErase(void) 関数 .....	70
▪ Command_ReadSRD(void) 関数 .....	70
▪ Com_ClearSRD(void) 関数.....	70
▪ Command_ExtBlankcheck(void) 関数.....	70
▪ Command_ExtSumcheck(void) 関数 .....	71
▪ wcp_init(void) 関数.....	71
▪ wcp_main(void) 関数.....	71
12. 使用I/Oリソース一覧.....	72
13. 付録.....	73

## 1. 概要

### 1.1 動作条件

項目	内容	ユーザ設定※
対象マイコン	μ PD70F3504、μ PD70F3505A、μ PD70F3506 μ PD70F3507、μ PD70F3508、μ PD70F3509 μ PD70F3598、μ PD70F3503	
書き込み対象アドレス * 1	ROM 容量 256K: #000000h~#03FFFFh ROM 容量 384K: #000000h~#05FFFFh ROM 容量 512K: #000000h~#07FFFFh ROM 容量 1M: #000000h~#0FFFFFFh	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=16MHz、動作=160MHz 逡倍値=10、分周値=なし (デフォルト入力=16MHz、動作=160MHz、逡倍値=10)	可
CAN チャンネル番号	チャンネル 0~1 * 4	可 * 4
モード制御端子	なし (FLMDCNT レジスタにより制御)	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 V3.00.01 * 3	不可
コンパイラバージョン	Renesas 製 CX V1.31 * 3	不可
最適化	IBL,WCP 共に” デフォルト(-Odefault)”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	512byte	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書き込み領域は下記のようになります。

ROM 容量 256K #004000~#03FFFF (248K)

ROM 容量 384K #004000~#05FFFF (376K)

ROM 容量 512K #004000~#07FFFF (504K)

ROM 容量 1M #004000~#0FFFFFF (1016K)

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いたしません。

\* 4 使用できる CAN チャンネル番号は、マイコンにより異なります。

μ PD70F3503 : 0 のみ使用可

μ PD70F3504, μ PD70F3505A, μ PD70F3506, μ PD70F3507 } : 0, 1 使用可  
μ PD70F3508, μ PD70F3509, μ PD70F3598 }

## 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-NAPYDC839GN7y	
CAN 共通プログラミング 技術資料	TR-NAPYDC839GN7y	本書
マイコンパック	NAPYDC839GN70 NAPYDC839GN71 NAPYDC839GN72 NAPYDC839GN73	ROM 容量 1M 版 ROM 容量 512K 版 ROM 容量 384K 版 ROM 容量 256K 版 ・SAMPLE OBJECT FILE 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- NAPYDC839GN70~N73   -Header     -y_ibl.h     -y_init.h   -IBL- NAPYDC839GN70~N73     -user_init.h     -user_apl.c     -user_ipr_start.s     -user_ipr.c     -y_ibl.c     -NAPYDC839GN7x_IBL.dir   -WCP- NAPYDC839GN70~N73     -wcp_start.s     -y_wcp.c     -NAPYDC839GN7x_WCP.dir     -nec_types.h     -FSL.h     -FSL_T05_NEC_R32.a	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 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



## 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 10ms 間 Connect コマンドを待ちます。

この約 10ms 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<sup>※1</sup>」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“物理的なメモリ領域を入れ替える機能”です。普段のブート領域は“ブ  
ロック 0~3”ですが、“ブロック 4~7”をブート領域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 0~3 を書き換えるモードです。

ブートスワップモードにするかどうかは、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

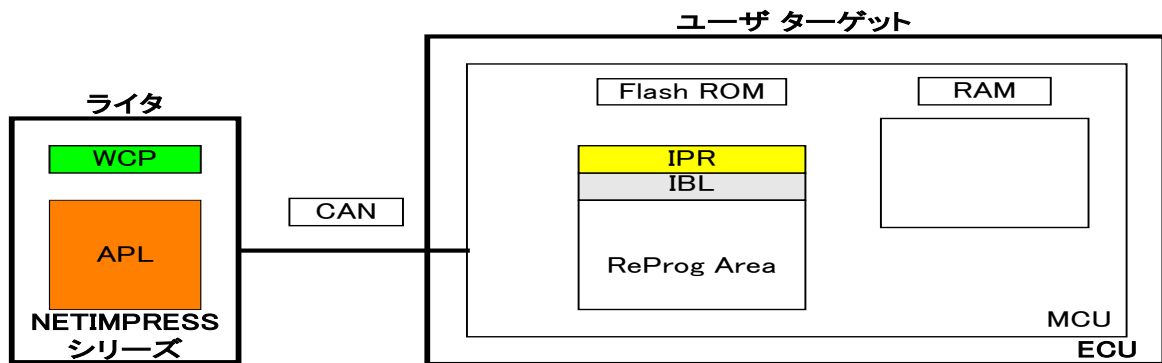
上記、ブートスワップ機能を用いず、ブロック 0~3 を書き換えないモードです。

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

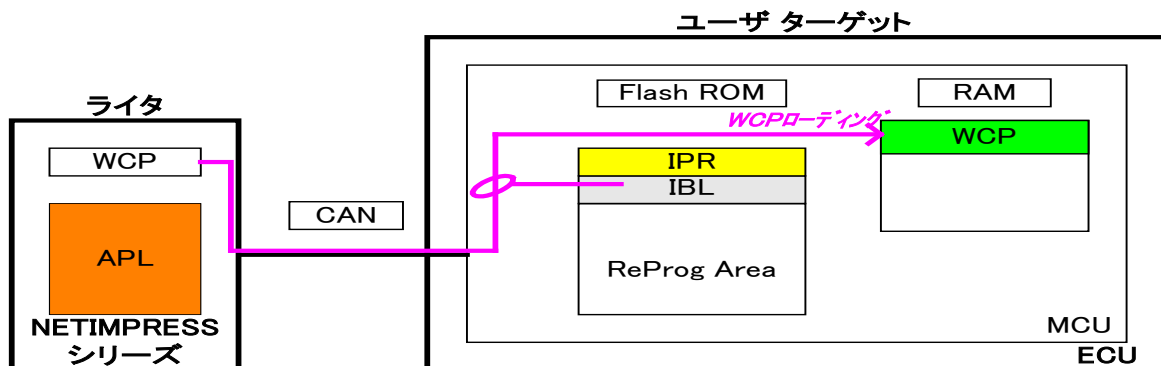
ただし、ブートスワップモードと非ブートスワップモードでは、使用するパラメータファイルも異なりますのでご注意ください。

### 3. 機能概要

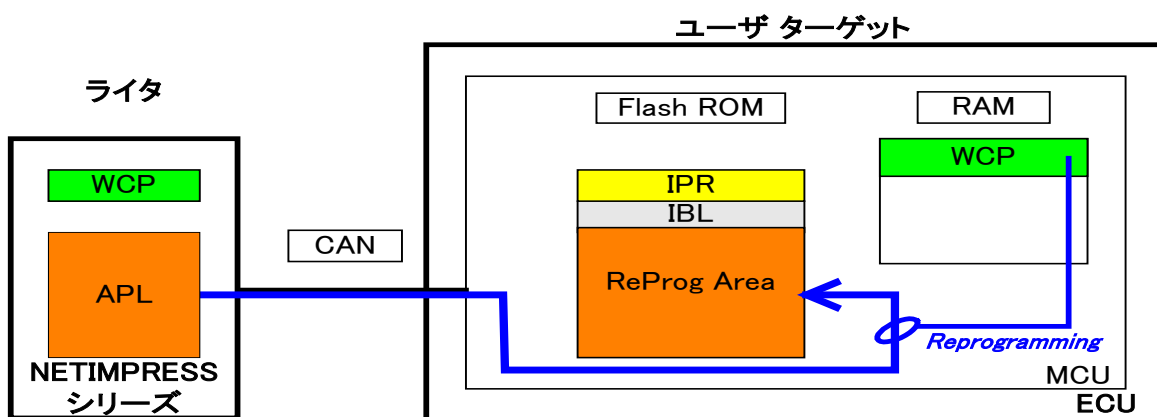
#### 3.1 UCOP システム構成図



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



\* 上記構成図は ROM 容量 384K 版に対するものです。

その他製品に関しましては、ユーザアプリ領域の終了アドレスが異なりますのでご注意ください。

その他製品のユーザアプリ終了アドレスは下記のようになります。

ROM 容量 256K 版 … #03FFFF

ROM 容量 512K 版 … #07FFFF

ROM 容量 1M 版 … #0FFFFFFF

### 3. 2. 2 消去ブロックアドレス(ブートスワップモード)

NAPYDC839GN70 (384K)

NAPYDC839GN71 (512K)

NAPYDC839GN72 (1M)

NAPYDC839GN73 (256K)

グループ°	アドレス
グループ° 0	#000000～#000FFF
グループ° 1	#001000～#001FFF
グループ° 2	#002000～#002FFF
グループ° 3	#003000～#003FFF
グループ° 4	#004000～#004FFF
グループ° 5	#005000～#005FFF
グループ° 6	#006000～#006FFF
グループ° 7	#007000～#007FFF
グループ° 8	#008000～#008FFF
グループ° 9	#009000～#009FFF
グループ° 10	#00A000～#00AFFF
グループ° 11	#00B000～#00BFFF
グループ° 12	#00C000～#00CFFF
グループ° 13	#00D000～#00DFFF
グループ° 14	#00E000～#00EFFF
グループ° 15	#00F000～#00FFFF
グループ° 16	#010000～#010FFF
グループ° 17	#011000～#011FFF
グループ° 18	#012000～#012FFF
グループ° 19	#013000～#013FFF
グループ° 20	#014000～#014FFF
グループ° 21	#015000～#015FFF
グループ° 22	#016000～#016FFF
グループ° 23	#017000～#017FFF
グループ° 24	#018000～#018FFF
グループ° 25	#019000～#019FFF
グループ° 26	#01A000～#01AFFF
グループ° 27	#01B000～#01BFFF
グループ° 28	#01C000～#01CFFF
グループ° 29	#01D000～#01DFFF
グループ° 30	#01E000～#01EFFF
グループ° 31	#01F000～#01FFFF
グループ° 32	#020000～#020FFF
グループ° 33	#021000～#021FFF
グループ° 34	#022000～#022FFF
グループ° 35	#023000～#023FFF
グループ° 36	#024000～#024FFF
グループ° 37	#025000～#025FFF
グループ° 38	#026000～#026FFF
グループ° 39	#027000～#027FFF

グループ 40	#028000～#028FFF
グループ 41	#029000～#029FFF
グループ 42	#02A000～#02AFFF
グループ 43	#02B000～#02BFFF
グループ 44	#02C000～#02CFFF
グループ 45	#02D000～#02DFFF
グループ 46	#02E000～#02EFFF
グループ 47	#02F000～#02FFFF
グループ 48	#030000～#030FFF
グループ 49	#031000～#031FFF
グループ 50	#032000～#032FFF
グループ 51	#033000～#033FFF
グループ 52	#034000～#034FFF
グループ 53	#035000～#035FFF
グループ 54	#036000～#036FFF
グループ 55	#037000～#037FFF
グループ 56	#038000～#038FFF
グループ 57	#039000～#039FFF
グループ 58	#03A000～#03AFFF
グループ 59	#03B000～#03BFFF
グループ 60	#03C000～#03CFFF
グループ 61	#03D000～#03DFFF
グループ 62	#03E000～#03EFFF
グループ 63	#03F000～#03FFFF
グループ 64	#040000～#040FFF
グループ 65	#041000～#041FFF
グループ 66	#042000～#042FFF
グループ 67	#043000～#043FFF
グループ 68	#044000～#044FFF
グループ 69	#045000～#045FFF
グループ 70	#046000～#046FFF
グループ 71	#047000～#047FFF
グループ 72	#048000～#048FFF
グループ 73	#049000～#049FFF
グループ 74	#04A000～#04AFFF
グループ 75	#04B000～#04BFFF
グループ 76	#04C000～#04CFFF
グループ 77	#04D000～#04DFFF
グループ 78	#04E000～#04EFFF
グループ 79	#04F000～#04FFFF
グループ 80	#050000～#050FFF
グループ 81	#051000～#051FFF
グループ 82	#052000～#052FFF
グループ 83	#053000～#053FFF
グループ 84	#054000～#054FFF
グループ 85	#055000～#055FFF
グループ 86	#056000～#056FFF
グループ 87	#057000～#057FFF
グループ 88	#058000～#058FFF
グループ 89	#059000～#059FFF

NAPYDC839GN73 (256K)



グループ 90	#05A000～#05AFFF	
グループ 91	#05B000～#05BFFF	
グループ 92	#05C000～#05CFFF	
グループ 93	#05D000～#05DFFF	
グループ 94	#05E000～#05EFFF	
グループ 95	#05F000～#05FFFF	<u>NAPYDC839GN70 (384K)</u>
グループ 96	#060000～#060FFF	
グループ 97	#061000～#061FFF	
グループ 98	#062000～#062FFF	
グループ 99	#063000～#063FFF	
グループ 100	#064000～#064FFF	
グループ 101	#065000～#065FFF	
グループ 102	#066000～#066FFF	
グループ 103	#067000～#067FFF	
グループ 104	#068000～#068FFF	
グループ 105	#069000～#069FFF	
グループ 106	#06A000～#06AFFF	
グループ 107	#06B000～#06BFFF	
グループ 108	#06C000～#06CFFF	
グループ 109	#06D000～#06DFFF	
グループ 110	#06E000～#06EFFF	
グループ 111	#06F000～#06FFFF	
グループ 112	#070000～#070FFF	
グループ 113	#071000～#071FFF	
グループ 114	#072000～#072FFF	
グループ 115	#073000～#073FFF	
グループ 116	#074000～#074FFF	
グループ 117	#075000～#075FFF	
グループ 118	#076000～#076FFF	
グループ 119	#077000～#077FFF	
グループ 120	#078000～#078FFF	
グループ 121	#079000～#079FFF	
グループ 122	#07A000～#07AFFF	
グループ 123	#07B000～#07BFFF	
グループ 124	#07C000～#07CFFF	
グループ 125	#07D000～#07DFFF	
グループ 126	#07E000～#07EFFF	
グループ 127	#07F000～#07FFFF	<u>NAPYDC839GN71 (512K)</u>
グループ 128	#080000～#080FFF	
グループ 129	#081000～#081FFF	
グループ 130	#082000～#082FFF	
グループ 131	#083000～#083FFF	
グループ 132	#084000～#084FFF	
グループ 133	#085000～#085FFF	
グループ 134	#086000～#086FFF	
グループ 135	#087000～#087FFF	
グループ 136	#088000～#088FFF	
グループ 137	#089000～#089FFF	
グループ 138	#08A000～#08AFFF	
グループ 139	#08B000～#08BFFF	

グループ 140	#08C000～#08CFFF
グループ 141	#08D000～#08DFFF
グループ 142	#08E000～#08EFFF
グループ 143	#08F000～#08FFFF
グループ 144	#090000～#090FFF
グループ 145	#091000～#091FFF
グループ 146	#092000～#092FFF
グループ 147	#093000～#093FFF
グループ 148	#094000～#094FFF
グループ 149	#095000～#095FFF
グループ 150	#096000～#096FFF
グループ 151	#097000～#097FFF
グループ 152	#098000～#098FFF
グループ 153	#099000～#099FFF
グループ 154	#09A000～#09AFFF
グループ 155	#09B000～#09BFFF
グループ 156	#09C000～#09CFFF
グループ 157	#09D000～#09DFFF
グループ 158	#09E000～#09EFFF
グループ 159	#09F000～#09FFFF
グループ 160	#0A0000～#0A0FFF
グループ 161	#0A1000～#0A1FFF
グループ 162	#0A2000～#0A2FFF
グループ 163	#0A3000～#0A3FFF
グループ 164	#0A4000～#0A4FFF
グループ 165	#0A5000～#0A5FFF
グループ 166	#0A6000～#0A6FFF
グループ 167	#0A7000～#0A7FFF
グループ 168	#0A8000～#0A8FFF
グループ 169	#0A9000～#0A9FFF
グループ 170	#0AA000～#0AAFFF
グループ 171	#0AB000～#0ABFFF
グループ 172	#0AC000～#0ACFFF
グループ 173	#0AD000～#0ADFFF
グループ 174	#0AE000～#0AEFFF
グループ 175	#0AF000～#0AFFFF
グループ 176	#0B0000～#0B0FFF
グループ 177	#0B1000～#0B1FFF
グループ 178	#0B2000～#0B2FFF
グループ 179	#0B3000～#0B3FFF
グループ 180	#0B4000～#0B4FFF
グループ 181	#0B5000～#0B5FFF
グループ 182	#0B6000～#0B6FFF
グループ 183	#0B7000～#0B7FFF
グループ 184	#0B8000～#0B8FFF
グループ 185	#0B9000～#0B9FFF
グループ 186	#0BA000～#0BAFFF
グループ 187	#0BB000～#0BBFFF
グループ 188	#0BC000～#0BCFFF
グループ 189	#0BD000～#0BDFFF

グループ 190	#0BE000～#0BEFFF
グループ 191	#0BF000～#0BFFFF
グループ 192	#0C0000～#0C0FFF
グループ 193	#0C1000～#0C1FFF
グループ 194	#0C2000～#0C2FFF
グループ 195	#0C3000～#0C3FFF
グループ 196	#0C4000～#0C4FFF
グループ 197	#0C5000～#0C5FFF
グループ 198	#0C6000～#0C6FFF
グループ 199	#0C7000～#0C7FFF
グループ 200	#0C8000～#0C8FFF
グループ 201	#0C9000～#0C9FFF
グループ 202	#0CA000～#0CAFFF
グループ 203	#0CB000～#0CBFFF
グループ 204	#0CC000～#0CCFFF
グループ 205	#0CD000～#0CDFFF
グループ 206	#0CE000～#0CEFFF
グループ 207	#0CF000～#0CFFFF
グループ 208	#0D0000～#0D0FFF
グループ 209	#0D1000～#0D1FFF
グループ 210	#0D2000～#0D2FFF
グループ 211	#0D3000～#0D3FFF
グループ 212	#0D4000～#0D4FFF
グループ 213	#0D5000～#0D5FFF
グループ 214	#0D6000～#0D6FFF
グループ 215	#0D7000～#0D7FFF
グループ 216	#0D8000～#0D8FFF
グループ 217	#0D9000～#0D9FFF
グループ 218	#0DA000～#0DAFFF
グループ 219	#0DB000～#0DBFFF
グループ 220	#0DC000～#0DCFFF
グループ 221	#0DD000～#0DDFFF
グループ 222	#0DE000～#0DEFFF
グループ 223	#0DF000～#0DFFFF
グループ 224	#0E0000～#0E0FFF
グループ 225	#0E1000～#0E1FFF
グループ 226	#0E2000～#0E2FFF
グループ 227	#0E3000～#0E3FFF
グループ 228	#0E4000～#0E4FFF
グループ 229	#0E5000～#0E5FFF
グループ 230	#0E6000～#0E6FFF
グループ 231	#0E7000～#0E7FFF
グループ 232	#0E8000～#0E8FFF
グループ 233	#0E9000～#0E9FFF
グループ 234	#0EA000～#0EAFFF
グループ 235	#0EB000～#0EBFFF
グループ 236	#0EC000～#0ECFFF
グループ 237	#0ED000～#0EDFFF
グループ 238	#0EE000～#0EEFFF
グループ 239	#0EF000～#0EFFFF

グループ 240	#0F0000～#0F0FFF
グループ 241	#0F1000～#0F1FFF
グループ 242	#0F2000～#0F2FFF
グループ 243	#0F3000～#0F3FFF
グループ 244	#0F4000～#0F4FFF
グループ 245	#0F5000～#0F5FFF
グループ 246	#0F6000～#0F6FFF
グループ 247	#0F7000～#0F7FFF
グループ 248	#0F8000～#0F8FFF
グループ 249	#0F9000～#0F9FFF
グループ 250	#0FA000～#0FAFFF
グループ 251	#0FB000～#0FBFFF
グループ 252	#0FC000～#0FCFFF
グループ 253	#0FD000～#0FDFFF
グループ 254	#0FE000～#0FEFFF
グループ 255	#0FF000～#0FFFFF

NAPYDC839GN72 (1M)

アドレス#000000～#000FFF 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)

NAPYDC839GN70 (384K)

NAPYDC839GN71 (512K)

NAPYDC839GN72 (1M)

NAPYDC839GN73 (256K)

グループ	アドレス
グループ 0	#000000～#000FFF
グループ 1	#001000～#001FFF
グループ 2	#002000～#002FFF
グループ 3	#003000～#003FFF
グループ 4	#004000～#004FFF
グループ 5	#005000～#005FFF
グループ 6	#006000～#006FFF
グループ 7	#007000～#007FFF
グループ 8	#008000～#008FFF
グループ 9	#009000～#009FFF
グループ 10	#00A000～#00AFFF
グループ 11	#00B000～#00BFFF
グループ 12	#00C000～#00CFFF
グループ 13	#00D000～#00DFFF
グループ 14	#00E000～#00EFFF
グループ 15	#00F000～#00FFFF
グループ 16	#010000～#010FFF
グループ 17	#011000～#011FFF
グループ 18	#012000～#012FFF
グループ 19	#013000～#013FFF
グループ 20	#014000～#014FFF

グループ 21	#015000～#015FFF
グループ 22	#016000～#016FFF
グループ 23	#017000～#017FFF
グループ 24	#018000～#018FFF
グループ 25	#019000～#019FFF
グループ 26	#01A000～#01AFFF
グループ 27	#01B000～#01BFFF
グループ 28	#01C000～#01CFFF
グループ 29	#01D000～#01DFFF
グループ 30	#01E000～#01EFFF
グループ 31	#01F000～#01FFFF
グループ 32	#020000～#020FFF
グループ 33	#021000～#021FFF
グループ 34	#022000～#022FFF
グループ 35	#023000～#023FFF
グループ 36	#024000～#024FFF
グループ 37	#025000～#025FFF
グループ 38	#026000～#026FFF
グループ 39	#027000～#027FFF
グループ 40	#028000～#028FFF
グループ 41	#029000～#029FFF
グループ 42	#02A000～#02AFFF
グループ 43	#02B000～#02BFFF
グループ 44	#02C000～#02CFFF
グループ 45	#02D000～#02DFFF
グループ 46	#02E000～#02EFFF
グループ 47	#02F000～#02FFFF
グループ 48	#030000～#030FFF
グループ 49	#031000～#031FFF
グループ 50	#032000～#032FFF
グループ 51	#033000～#033FFF
グループ 52	#034000～#034FFF
グループ 53	#035000～#035FFF
グループ 54	#036000～#036FFF
グループ 55	#037000～#037FFF
グループ 56	#038000～#038FFF
グループ 57	#039000～#039FFF
グループ 58	#03A000～#03AFFF
グループ 59	#03B000～#03BFFF
グループ 60	#03C000～#03CFFF
グループ 61	#03D000～#03DFFF
グループ 62	#03E000～#03EFFF
グループ 63	#03F000～#03FFFF
グループ 64	#040000～#040FFF
グループ 65	#041000～#041FFF
グループ 66	#042000～#042FFF
グループ 67	#043000～#043FFF
グループ 68	#044000～#044FFF
グループ 69	#045000～#045FFF
グループ 70	#046000～#046FFF

NAPYDC839GN73 (256K)

グループ 71	#047000～#047FFF
グループ 72	#048000～#048FFF
グループ 73	#049000～#049FFF
グループ 74	#04A000～#04AFFF
グループ 75	#04B000～#04BFFF
グループ 76	#04C000～#04CFFF
グループ 77	#04D000～#04DFFF
グループ 78	#04E000～#04EFFF
グループ 79	#04F000～#04FFFF
グループ 80	#050000～#050FFF
グループ 81	#051000～#051FFF
グループ 82	#052000～#052FFF
グループ 83	#053000～#053FFF
グループ 84	#054000～#054FFF
グループ 85	#055000～#055FFF
グループ 86	#056000～#056FFF
グループ 87	#057000～#057FFF
グループ 88	#058000～#058FFF
グループ 89	#059000～#059FFF
グループ 90	#05A000～#05AFFF
グループ 91	#05B000～#05BFFF
グループ 92	#05C000～#05CFFF
グループ 93	#05D000～#05DFFF
グループ 94	#05E000～#05EFFF
グループ 95	#05F000～#05FFFF
グループ 96	#060000～#060FFF
グループ 97	#061000～#061FFF
グループ 98	#062000～#062FFF
グループ 99	#063000～#063FFF
グループ 100	#064000～#064FFF
グループ 101	#065000～#065FFF
グループ 102	#066000～#066FFF
グループ 103	#067000～#067FFF
グループ 104	#068000～#068FFF
グループ 105	#069000～#069FFF
グループ 106	#06A000～#06AFFF
グループ 107	#06B000～#06BFFF
グループ 108	#06C000～#06CFFF
グループ 109	#06D000～#06DFFF
グループ 110	#06E000～#06EFFF
グループ 111	#06F000～#06FFFF
グループ 112	#070000～#070FFF
グループ 113	#071000～#071FFF
グループ 114	#072000～#072FFF
グループ 115	#073000～#073FFF
グループ 116	#074000～#074FFF
グループ 117	#075000～#075FFF
グループ 118	#076000～#076FFF
グループ 119	#077000～#077FFF
グループ 120	#078000～#078FFF

NAPYDC839GN70 (384K)

グループ 121	#079000～#079FFF
グループ 122	#07A000～#07AFFF
グループ 123	#07B000～#07BFFF
グループ 124	#07C000～#07CFFF
グループ 125	#07D000～#07DFFF
グループ 126	#07E000～#07EFFF
グループ 127	#07F000～#07FFFF
グループ 128	#080000～#080FFF
グループ 129	#081000～#081FFF
グループ 130	#082000～#082FFF
グループ 131	#083000～#083FFF
グループ 132	#084000～#084FFF
グループ 133	#085000～#085FFF
グループ 134	#086000～#086FFF
グループ 135	#087000～#087FFF
グループ 136	#088000～#088FFF
グループ 137	#089000～#089FFF
グループ 138	#08A000～#08AFFF
グループ 139	#08B000～#08BFFF
グループ 140	#08C000～#08CFFF
グループ 141	#08D000～#08DFFF
グループ 142	#08E000～#08EFFF
グループ 143	#08F000～#08FFFF
グループ 144	#090000～#090FFF
グループ 145	#091000～#091FFF
グループ 146	#092000～#092FFF
グループ 147	#093000～#093FFF
グループ 148	#094000～#094FFF
グループ 149	#095000～#095FFF
グループ 150	#096000～#096FFF
グループ 151	#097000～#097FFF
グループ 152	#098000～#098FFF
グループ 153	#099000～#099FFF
グループ 154	#09A000～#09AFFF
グループ 155	#09B000～#09BFFF
グループ 156	#09C000～#09CFFF
グループ 157	#09D000～#09DFFF
グループ 158	#09E000～#09EFFF
グループ 159	#09F000～#09FFFF
グループ 160	#0A0000～#0A0FFF
グループ 161	#0A1000～#0A1FFF
グループ 162	#0A2000～#0A2FFF
グループ 163	#0A3000～#0A3FFF
グループ 164	#0A4000～#0A4FFF
グループ 165	#0A5000～#0A5FFF
グループ 166	#0A6000～#0A6FFF
グループ 167	#0A7000～#0A7FFF
グループ 168	#0A8000～#0A8FFF
グループ 169	#0A9000～#0A9FFF
グループ 170	#0AA000～#0AAFFF

NAPYDC839GN71 (512K)

グループ 171	#0AB000～#0ABFFF
グループ 172	#0AC000～#0ACFFF
グループ 173	#0AD000～#0ADFFF
グループ 174	#0AE000～#0AEFFF
グループ 175	#0AF000～#0AFFFF
グループ 176	#0B0000～#0B0FFF
グループ 177	#0B1000～#0B1FFF
グループ 178	#0B2000～#0B2FFF
グループ 179	#0B3000～#0B3FFF
グループ 180	#0B4000～#0B4FFF
グループ 181	#0B5000～#0B5FFF
グループ 182	#0B6000～#0B6FFF
グループ 183	#0B7000～#0B7FFF
グループ 184	#0B8000～#0B8FFF
グループ 185	#0B9000～#0B9FFF
グループ 186	#0BA000～#0BAFFF
グループ 187	#0BB000～#0BBFFF
グループ 188	#0BC000～#0BCFFF
グループ 189	#0BD000～#0BDFFF
グループ 190	#0BE000～#0BEFFF
グループ 191	#0BF000～#0BFFFF
グループ 192	#0C0000～#0C0FFF
グループ 193	#0C1000～#0C1FFF
グループ 194	#0C2000～#0C2FFF
グループ 195	#0C3000～#0C3FFF
グループ 196	#0C4000～#0C4FFF
グループ 197	#0C5000～#0C5FFF
グループ 198	#0C6000～#0C6FFF
グループ 199	#0C7000～#0C7FFF
グループ 200	#0C8000～#0C8FFF
グループ 201	#0C9000～#0C9FFF
グループ 202	#0CA000～#0CAFFF
グループ 203	#0CB000～#0CBFFF
グループ 204	#0CC000～#0CCFFF
グループ 205	#0CD000～#0CDFFF
グループ 206	#0CE000～#0CEFFF
グループ 207	#0CF000～#0CFFFF
グループ 208	#0D0000～#0D0FFF
グループ 209	#0D1000～#0D1FFF
グループ 210	#0D2000～#0D2FFF
グループ 211	#0D3000～#0D3FFF
グループ 212	#0D4000～#0D4FFF
グループ 213	#0D5000～#0D5FFF
グループ 214	#0D6000～#0D6FFF
グループ 215	#0D7000～#0D7FFF
グループ 216	#0D8000～#0D8FFF
グループ 217	#0D9000～#0D9FFF
グループ 218	#0DA000～#0DAFFF
グループ 219	#0DB000～#0DBFFF
グループ 220	#0DC000～#0DCFFF



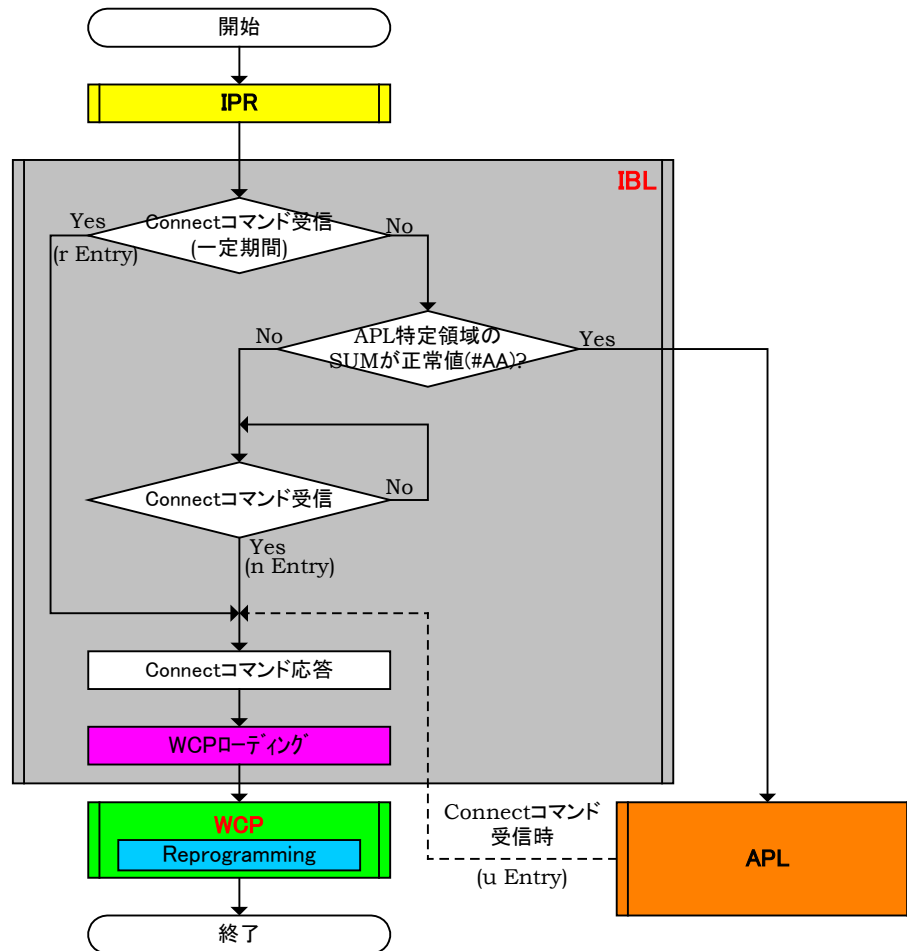
グループ 221	#0DD000～#0DDFFF
グループ 222	#0DE000～#0DEFFF
グループ 223	#0DF000～#0DFFFF
グループ 224	#0E0000～#0E0FFF
グループ 225	#0E1000～#0E1FFF
グループ 226	#0E2000～#0E2FFF
グループ 227	#0E3000～#0E3FFF
グループ 228	#0E4000～#0E4FFF
グループ 229	#0E5000～#0E5FFF
グループ 230	#0E6000～#0E6FFF
グループ 231	#0E7000～#0E7FFF
グループ 232	#0E8000～#0E8FFF
グループ 233	#0E9000～#0E9FFF
グループ 234	#0EA000～#0EAFFF
グループ 235	#0EB000～#0EBFFF
グループ 236	#0EC000～#0ECFFF
グループ 237	#0ED000～#0EDFFF
グループ 238	#0EE000～#0EEFFF
グループ 239	#0EF000～#0EFFFF
グループ 240	#0F0000～#0F0FFF
グループ 241	#0F1000～#0F1FFF
グループ 242	#0F2000～#0F2FFF
グループ 243	#0F3000～#0F3FFF
グループ 244	#0F4000～#0F4FFF
グループ 245	#0F5000～#0F5FFF
グループ 246	#0F6000～#0F6FFF
グループ 247	#0F7000～#0F7FFF
グループ 248	#0F8000～#0F8FFF
グループ 249	#0F9000～#0F9FFF
グループ 250	#0FA000～#0FAFFF
グループ 251	#0FB000～#0FBFFF
グループ 252	#0FC000～#0FCFFF
グループ 253	#0FD000～#0FDFFF
グループ 254	#0FE000～#0FEFFF
グループ 255	#0FF000～#0FFFFFF

NAPYDC839GN72 (1M)

灰色部は IBL、IPR 域で書き換え禁止領域です。

またこの灰色部は APL での RAM エミュレーションも禁止とします。

### 3.3 プログラムエントリー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。

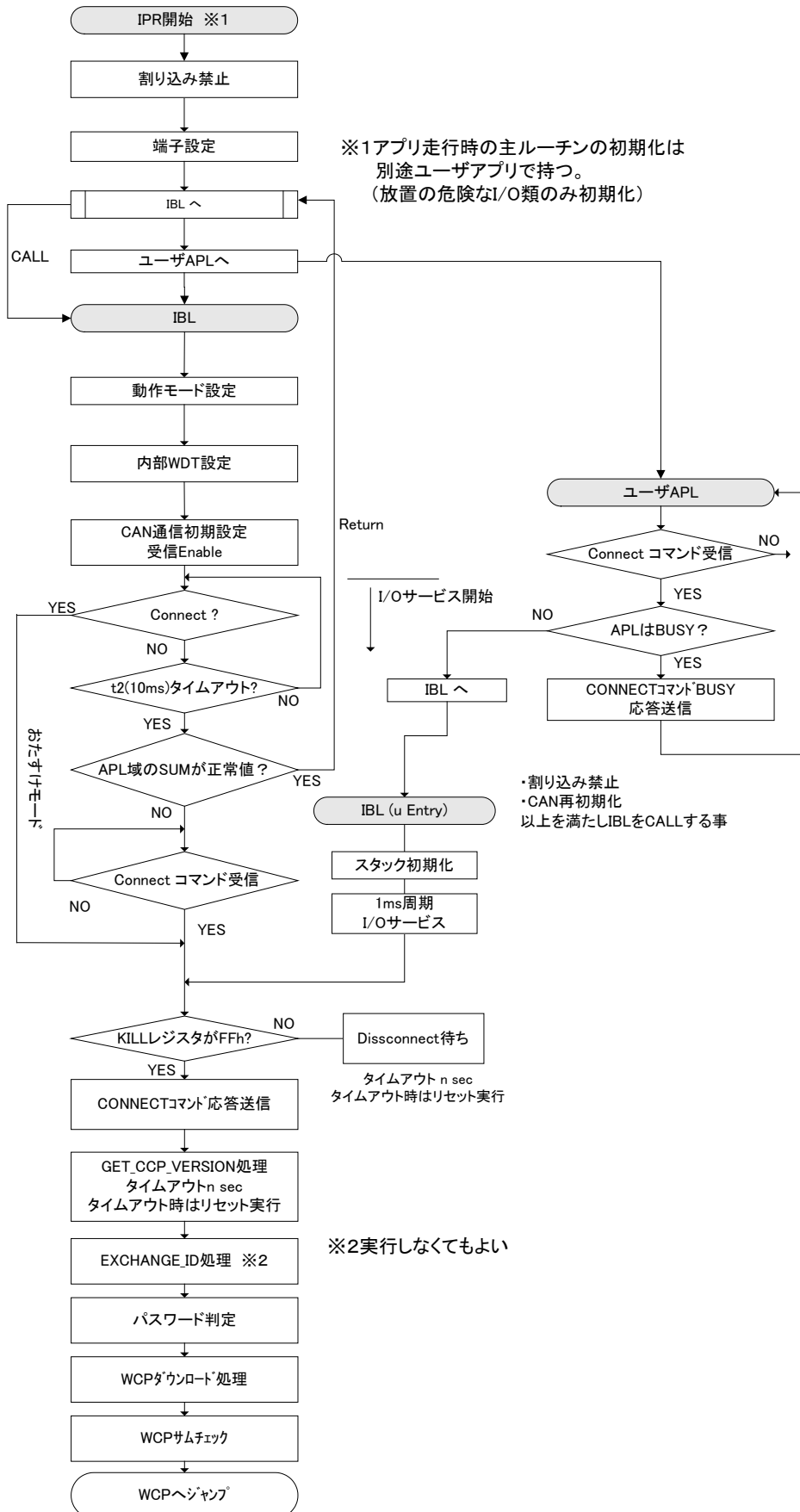
※ライタは Connect コマンド発行後規定時間(25ms)応答がない場合、Disconnect コマンドを発行し、ターゲット MCU との接続を解除します。

※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。

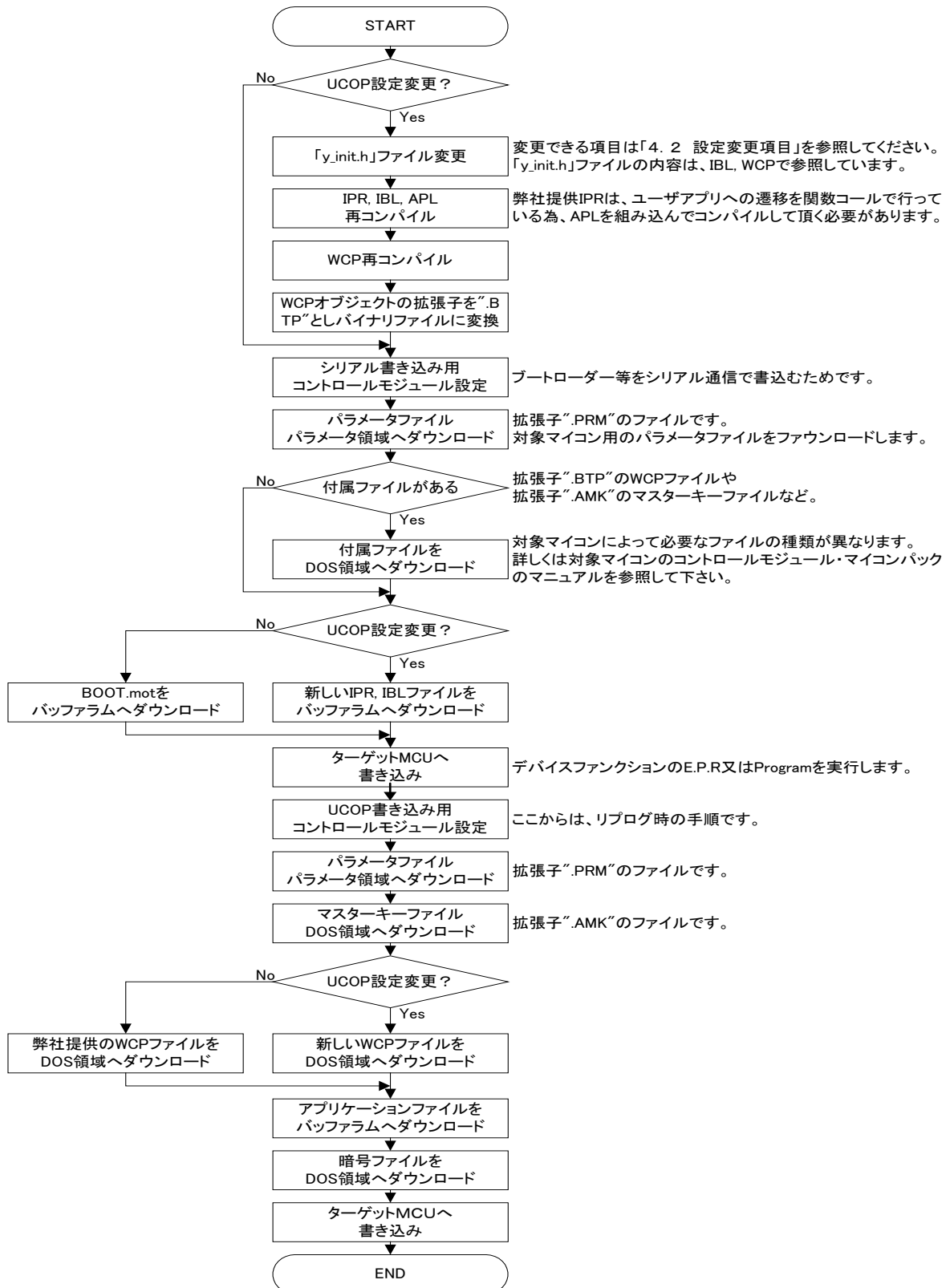
※ターゲット MCU は Disconnect コマンド受信後、リセット状態に戻るものとします。

### 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 グローバルクロック選択レジスタ値」、「5. 6 モジュールビットレートレジスタ値」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 グローバルクロック選択レジスタ値」、「5. 6 モジュールビットレートレジスタ値」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 グローバルクロック選択レジスタ値」、「5. 6 モジュールビットレートレジスタ値」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 グローバルクロック選択レジスタ値」、「5. 6 モジュールビットレートレジスタ値」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 グローバルクロック選択レジスタ値」、「5. 6 モジュールビットレートレジスタ値」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. 7 パスワードチェック領域開始アドレス」、「5. 8 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. 9 ユーザアプリ領域サム値チェック開始アドレス」、「5. 10 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. 11 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. 12 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. 13 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. 14 Primary ID」、「5. 15 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. 16 ステーションアドレス」を参照してください。

#### 4. 2. 9 CANチャネルの変更

「5. 17 CAN チャネルの番号」を参照してください。

#### 4. 2. 10 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クロック設定」を参照してください。

#### 4. 2. 11 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。

#### 4. 2. 12 KILLレジスタアドレス設定

「5. 21 KILL レジスタアドレス設定」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 AF930 操作マニュアルプログラマ編 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

※必ずマイコンのグローバルクロック選択レジスタ値とモジュールビットレートレジスタ値も再計算し、初期設定ファイルの“CAN\_GMCSPRE”と“CAN\_CMBTCTL”も再設定してください。

「5. 5 グローバルクロック選択レジスタ値」と「5. 6 モジュールビットレートレジスタ値」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	0x34	0x 40
500 Kbps	0x 34	0x 41
250 Kbps	0x 34	0x 43
125 Kbps	0x 34	0x 47

※ 「5. 22 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 SETTING”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”の変更とオプションバイトの設定が必要です。

### 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

※必ずマイコンのグローバルクロック選択レジスタ値とモジュールビットレートレジスタ値も再計算し、初期設定ファイルの“CAN\_GMCSPRE”と“CAN\_CMBTCTL”も再設定してください。

「5.5 グローバルクロック選択レジスタ値」と「5.6 モジュールビットレートレジスタ値」をご参照ください。

### ②. オプションバイトの設定方法

本マイコンパックでオプションバイトの設定を行うことはできません。

シリアルライタ等を使用して設定値を変更する必要があります。

マイコンのマニュアルを参照の上、正しい値を設定してください。

## 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

※必ずマイコンのグローバルクロック選択レジスタ値とモジュールビットレートレジスタ値も再計算し、初期設定ファイルの“CAN\_GMCSPRE”と“CAN\_CMBTCTL”も再設定してください。

「5.5 グローバルクロック選択レジスタ値」と「5.6 モジュールビットレートレジスタ値」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「 $\div 2$ 」の場合、2 と設定

※必ずマイコンのグローバルクロック選択レジスタ値とモジュールビットレートレジスタ値も再計算し、初期設定ファイルの“CAN\_GMCSPRE”と“CAN\_CMBTCTL”も再設定してください。

「5. 5 グローバルクロック選択レジスタ値」と「5. 6 モジュールビットレートレジスタ値」をご参照ください。

※ このマイコンパックでは分周比「 $\div 1$ 」固定です。

## 5. 5 グローバルクロック選択レジスタ値

グローバルクロック選択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変更してください。

グローバルクロック選択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\_GMCSPRE”に設定値を設定してください。

グローバルクロック選択レジスタは 8 ビットレジスタです。0~F 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」をご参照ください。

## 5. 6 モジュールビットレートレジスタ値

モジュールビットレート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変更してください。

モジュール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\_CMBTCTL”に設定値を設定してください。

モジュールビットレートレジスタは 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」をご参照ください。

## 5. 7 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になります。

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. 8 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になります。

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. 9 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”とライター側両方の変更が必要です。

注意:この領域はブロック単位で設定する必要があります。設定するアドレスはブロックの先頭アドレスとしてください。

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

※ 「5. 22 Specific Parameter 変更方法」を参照下さい。

## 5. 10 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: この領域はブロック単位で設定する必要があります。設定するアドレスはブロックの先頭アドレスとしてください。

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

※ 「5. 22 Specific Parameter 変更方法」を参照下さい。

## 5. 11 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(6-15. 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. 12 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 ms です。

## 5. 13 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\_ibl.c”、“y\_WCP.c”を変更する必要があります。

- ①. “y\_ibl.c”変更点

・”io\_service”関数

プログラム

```
P0 ^= 0x01;          /* ビット反転 */
```

変更点

I/O サービス用ポートのポートレジスタ値のビット反転を行い、I/O ポートからの出力を反転させています。

I/O ポートサービスを実行するポートのポートレジスタに変更してください。

・”init\_timer”関数

プログラム

```
PMC0 &= 0x7E;      /* ポートモードコントロールレジスタの設定 */  
PM0 &= 0xFE;       /* ポートモードレジスタ 出力モードに設定 */  
P0 &= 0x7E;        /* P00 からローを出力 */
```

変更点

I/O ポートサービスを行うポートの初期化を行っています。

I/O ポートサービスを実行するポートの初期化を行ってください。

・”ibl\_entry”関数

プログラム

```
PMC0 &= 0x7E;      /* ポートモードコントロールレジスタの設定 */  
PM0 &= 0xFE;       /* ポートモードレジスタ 出力モードに設定 */  
P0 &= 0x7E;        /* P00 からローを出力 */
```

```

P0 ^= 0x01;          /* ハルストグル出力 */
mcST()->IOS_Div = 0;
while(mcST()->IOS_Div != 2) io_service(&mcST()->IOS_Div); /* ウェイト */
P0 ^= 0x01;          /* ハルストグル出力 */

```

**変更点**

I/O ポートサービス用ポートの再初期化を行っています。  
 また、1ms 周期で 1 回パルスを出力します。  
 I/O ポートサービスを実行するポートに変更してください。

②. "y\_WCP.c" 変更点

・"io\_service" 関数

**プログラム**

```

P0 ^= 0x01;          /* ビット反転 */

```

**変更点**

I/O サービス用ポートのポートレジスタ値のビット反転を行い、I/O ポートからの出力を反転させています。  
 I/O ポートサービスを実行するポートのポートレジスタに変更してください。

## 5. 14 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。

ライタからマイコンへ送る Primary ID を設定する "ID\_P\_IN" と、マイコンからライタへ送る Primary ID を設定する "ID\_P\_MCU" があります。

必要に応じて "ID\_P\_IN" 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\_IN"、"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
			EXD_ID17	EXD_ID16	EXD_ID15	EXD_ID14	EXD_ID13

Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	EXD_ID7	EXD_ID6	EXD_ID5

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
--------	--------	--------	--------	--------	--------	-------	-------

EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	STD_ID10	STD_ID9	STD_ID8
---------	---------	---------	---------	---------	----------	---------	---------

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	STD_ID2	STD_ID1	STD_ID0

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 31、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\_IN”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\_IN”: 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\_IN”: 0x1F6E5FE9

“ID\_P\_MCU”: 0x1F6E5FEA

※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. 15 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. 22 Specific Parameter 変更方法」を参照下さい。

#0C4～#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\_IN” 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 Parameter のアドレス	#0C4	#0C5	#0C6	#0C7
設定値	0x00	0x00	0x07	0xEA

Specific Parameter 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\_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

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

Specific Parameter 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. 15 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、ライター側で設定されています。

### ①. 初期設定ファイル“y\_init.h”設定変更

“CAN\_ID”にスタンダード・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を設定します。

0:スタンダード・フォーマット使用

1:エクステンデッド・フォーマット使用

### ②. ライター側設定変更

#### 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. 14 Primary ID」を参照してください。

#### ii. ライターでの変更

“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ー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. 16 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”とライター側両方の変更が必要です。

「6. 13 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. ライター側設定変更

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. 22 Specific Parameter 変更方法」を参照下さい。



## 5. 17 CAN チャンネル番号

CAN モジュールのチャンネルが複数存在する場合、どのチャンネルを使用するかを設定します。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\_CHAN”に使用する CAN チャンネルの番号を設定してください。

## 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”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\_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クロック設定

内蔵ウォッチドッグタイマの WDTA モードレジスタ(WDTA0MD)の値を設定します。

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\_MD”にウォッチドッグタイマモードレジスタ(WDTA0MD)の値を設定してください。

## 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 KILL レジスタアドレス設定

KILL レジスタアドレスを変更するには、初期設定ファイル”y\_init.h”を変更する必要があります。

KILL 機能の使用方法は「6.10 KILL レジスタ」を参照下さい。

### ①. 初期設定ファイル”y\_init.h”設定変更

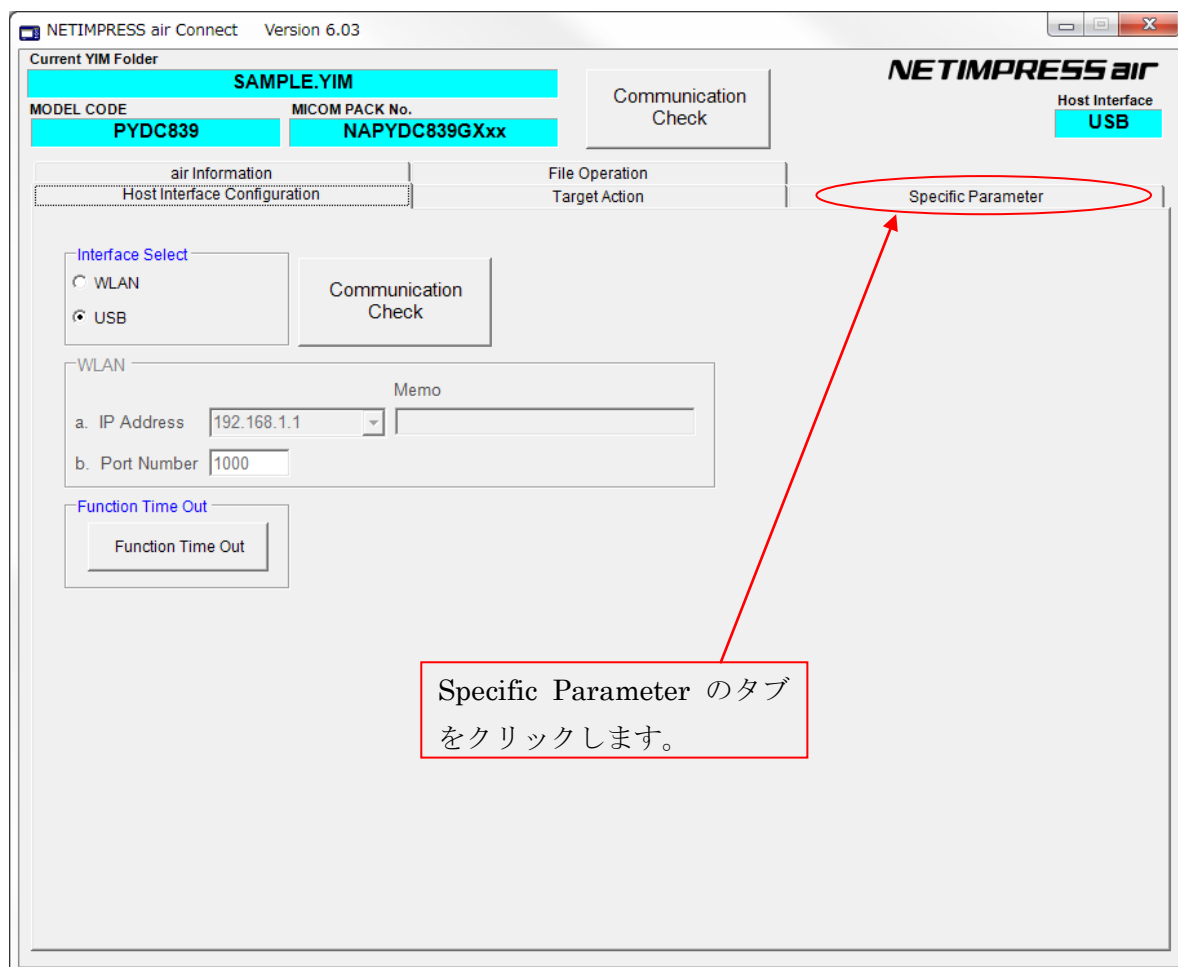
“KILL\_ADDR”に KILL レジスタのアドレスを設定してください。

## 5. 22 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。

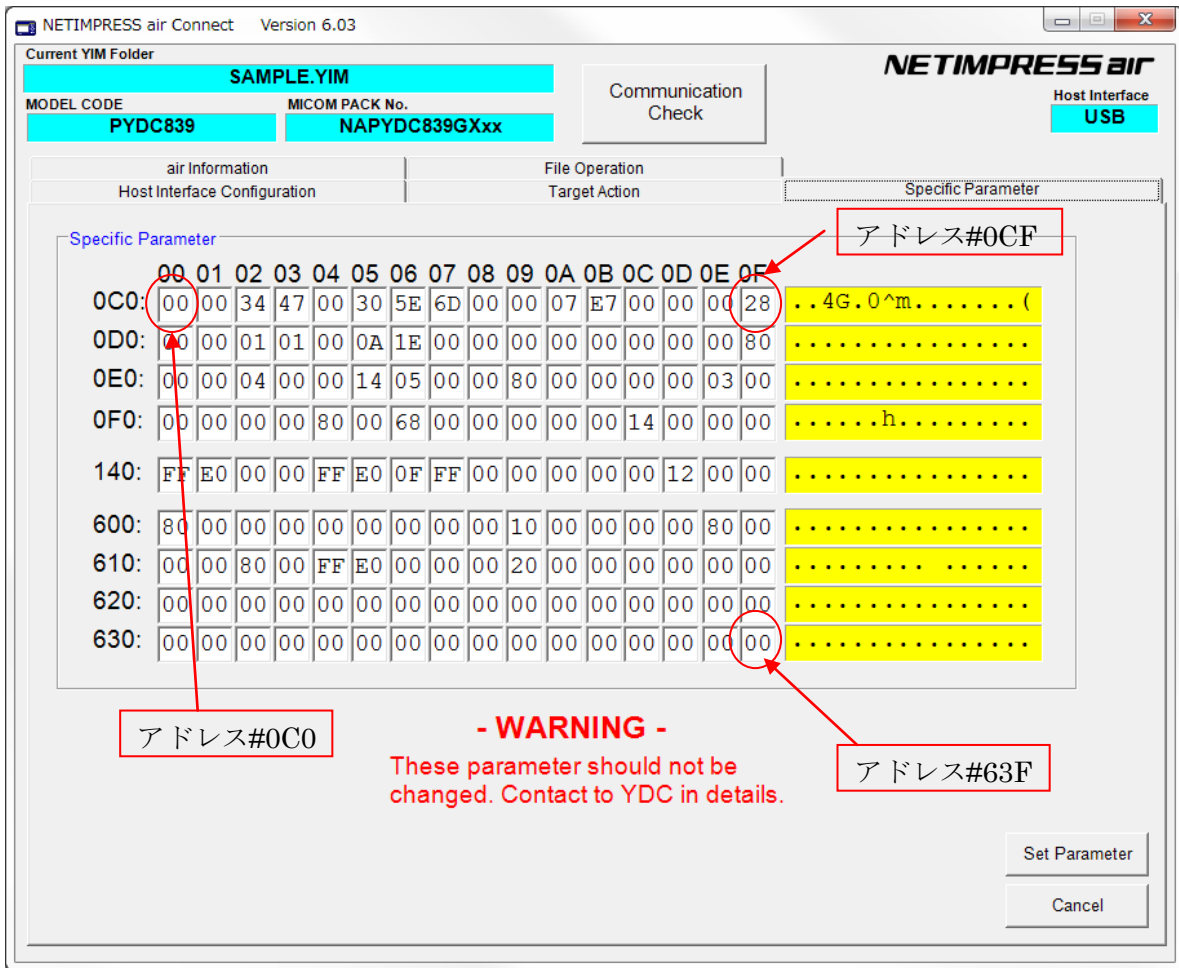
The screenshot shows the NETIMPRESS air Connect software interface. At the top, it displays 'Current YIM Folder: SAMPLE.YIM', 'MODEL CODE: PYDC839', and 'MICOM PACK No.: NAPYDC839GXxx'. The 'Host Interface' is set to 'USB'. A 'Specific Parameter' table is visible, with columns labeled 00 through 0F. A 'Password' dialog box is overlaid on the table, 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	.....

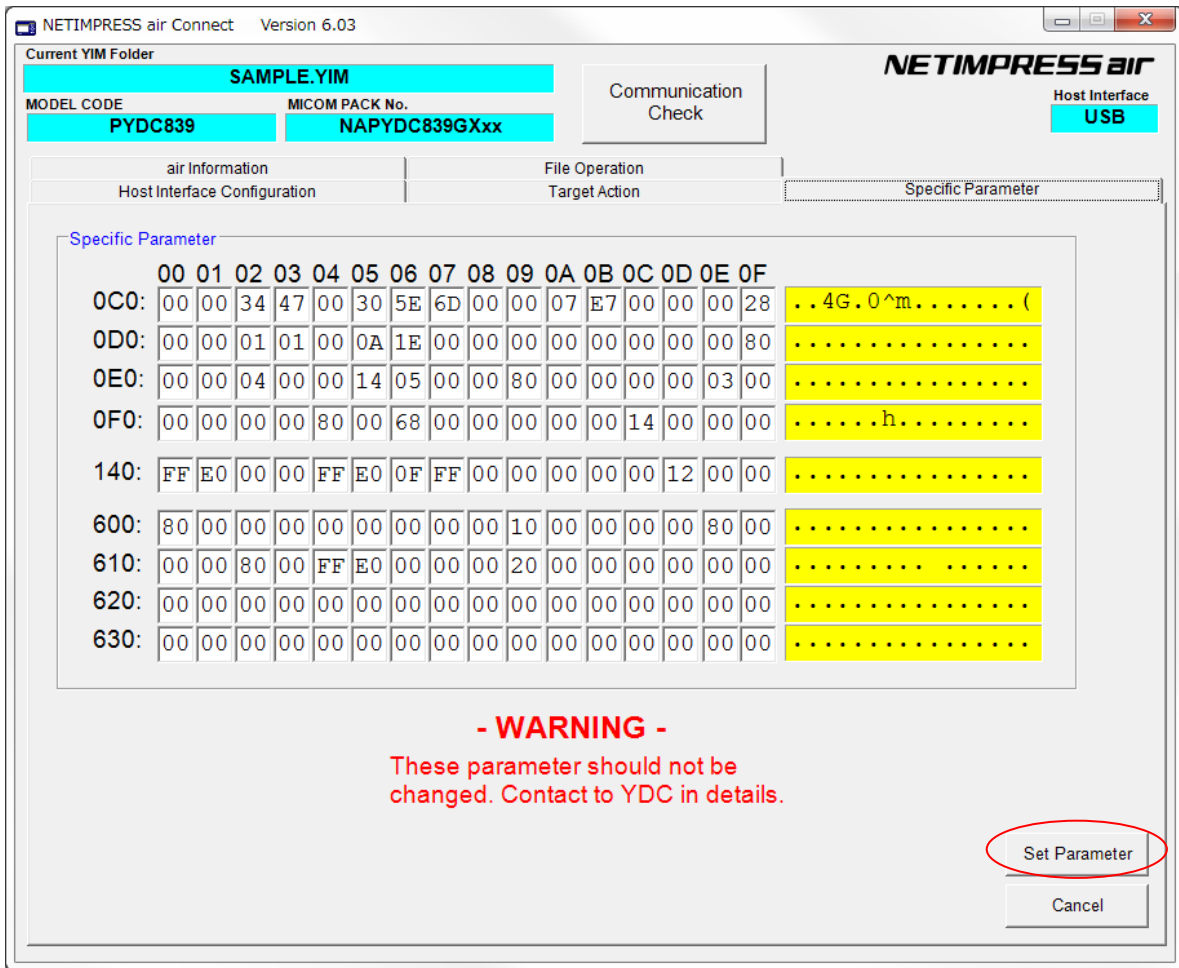
**- WARNING -**  
These parameter should not be changed. Contact to YDC in details.

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”Set 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	#00000800h～#000009FFh (512byte) ※1 (スタートアドレス:#00000800h) ※2	不可
IPR 要初期化項目	■ 割り込み禁止 ■ スタック初期化 ■ 端子の設定	可

※1: 割り込み/例外テーブル領域(#00000000～#000007CF)は IBL、WCP(後述)では使用しておりませんので、IPR 領域が足りない場合、ベースアドレスを変更し割り込み/例外テーブル領域を IPR 領域に割り当てる事ができます。

※2: 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	#000A00h~#003FFFh (13.5Kbyte)	不可

※ 対象マイコン共通とします。

※ 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。

※ デフォルト(-Odefault)の最適化を使用しています。

## 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。

YIM フォルダ内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
対象マイコン共通	#FEDFC000h~#FEDFDFFFh (RAM 8Kbyte)	

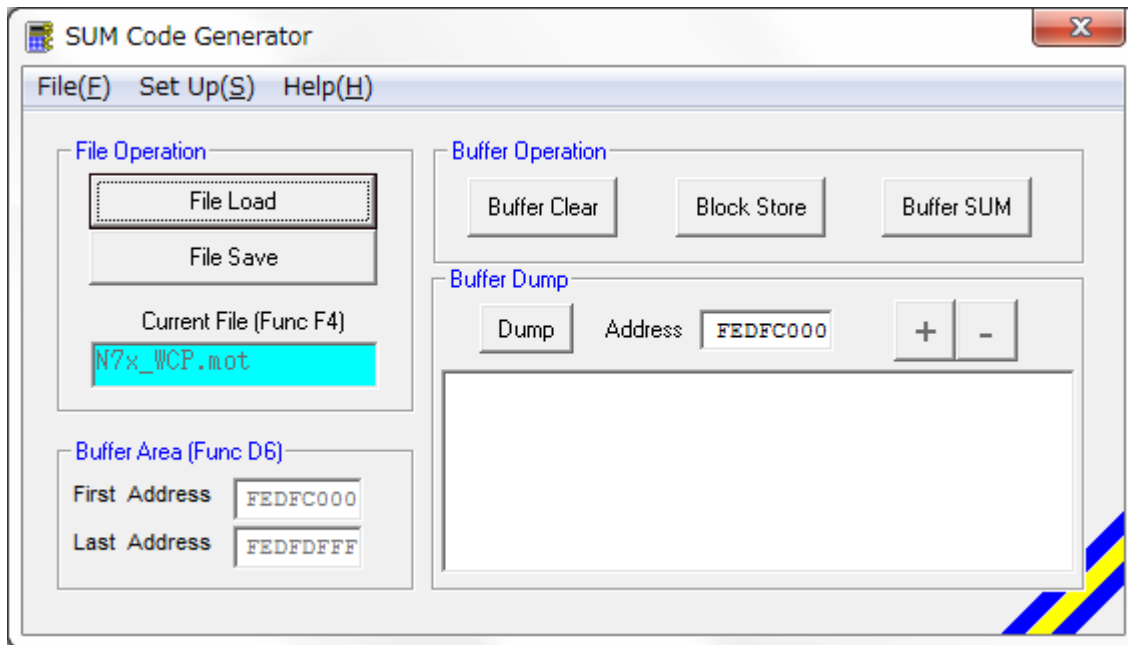
※ デフォルト(-Odefault)の最適化を使用しています。

ソースをコンパイルすると”N7x\_WCP.mot”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#FEDFC000～# FEDFDFFFになります。)

その後、“N7x\_WCP.mot”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. この領域はブロックサイズで規定するものとします。  
ii. 1つの消去ブロックの範囲に設定してください。  
iii. 書き込み禁止領域中には設定しないで下さい。  
iv. ブートスワップモード時、0x1FFFF 以下のアドレス中にユーザ領域サム値チェック領域を設ける場合には、必ず 0x1FFFF を含む領域としてください。

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	#1F000h~#1FFFFh (デフォルト値※)	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. 14 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メッセージバッファ 63 を使用します。
- ・IBLは送信用メッセージバッファとしてメッセージバッファ 62 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。
- ・メッセージバッファ 0～61 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	メールボックス 63	不可
IBL 送信用 メッセージバッファ	メールボックス 62	不可

#### 6. 5. 4 チャンネル

チャンネルが複数存在するマイコンでは、使用する CAN のチャンネルを設定することが可能です。  
初期設定ファイルの CAN\_CHAN の値を設定した後、コンパイルして IBL を書き換えてください。  
(「5. 17 CAN チャンネル番号」参照)

#### 6. 6 ブートスワップ機能

WCP ではマイコンの持つブートスワップ機能を使用しています。(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。)

そのためアドレス#000000～#003FFF に対するプログラミングが可能です。

ただし、アドレス#000000～#003FFF に書き込む場合には、IBL,IPR などのプログラムも共に書いて下さい。

(IBL・IPR を書き込まないと、UCOP を使用しての書き込みができなくなります。)

非スワップモード時にはブート領域(グループ 0～3)に対するプログラミングができませんのでご注意下さい。

ブートモードと非ブート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下さい。

## 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	<input checked="" type="checkbox"/> CAN 通信条件変更不可 ※1 <input checked="" type="checkbox"/> CONNECT コマンド受信 (受信後 20ms 以内(※2)で IBL を CALL する) ※3 <input type="checkbox"/> 内部 WDT ON(任意)	可
Connect コマンド受信後の APL 要初期化項目 (u Entry 時)  ■: 必須 □: 選択	<input checked="" type="checkbox"/> CAN 受信バッファクリア(受信完了状態にする) <input checked="" type="checkbox"/> 割り込み禁止 <input type="checkbox"/> 内部 WDT OFF	可

※1. 常にライタからのコネクトコマンドを受信できるようにしてください

※2. 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

※3. 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

## 6. 10 KILL レジスタ

- ・8bitで構成され、初期設定ファイルにアドレス情報が格納されています。
- ・KILLレジスタがOFFの時(レジスタ値“#FF”)は、コネクトコマンド受信後、コネクト応答をライタに返し、UCOPリプログラムモードを続行します。
- ・KILLレジスタがONの時(レジスタ値“#FF”以外)は、コネクトコマンド受信後、コネクト応答せず、コマンド受信待ち状態になります。  
コマンド受信後、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。
- ・このマイコンパックでは、ブートスワップモードを用い、下記アドレスに直接値を設定することで、KILLレジスタの機能を使用することができます。

注)1度KILLレジスタをONに設定すると、2度とOFFにすることは出来ませんのでご注意ください。

項目	内容	ユーザ設定
KILL レジスタアドレス	#001FF0h	可

## 6. 11 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. 12 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設定します。

グローバルクロック選択レジスタとビットレートレジスタの値を設定してください。

詳細は「5. 5 グローバルクロック選択レジスタ値」と「5. 6 モジュールビットレートレジスタ値」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、動作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. 13 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. 16 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. 14 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。

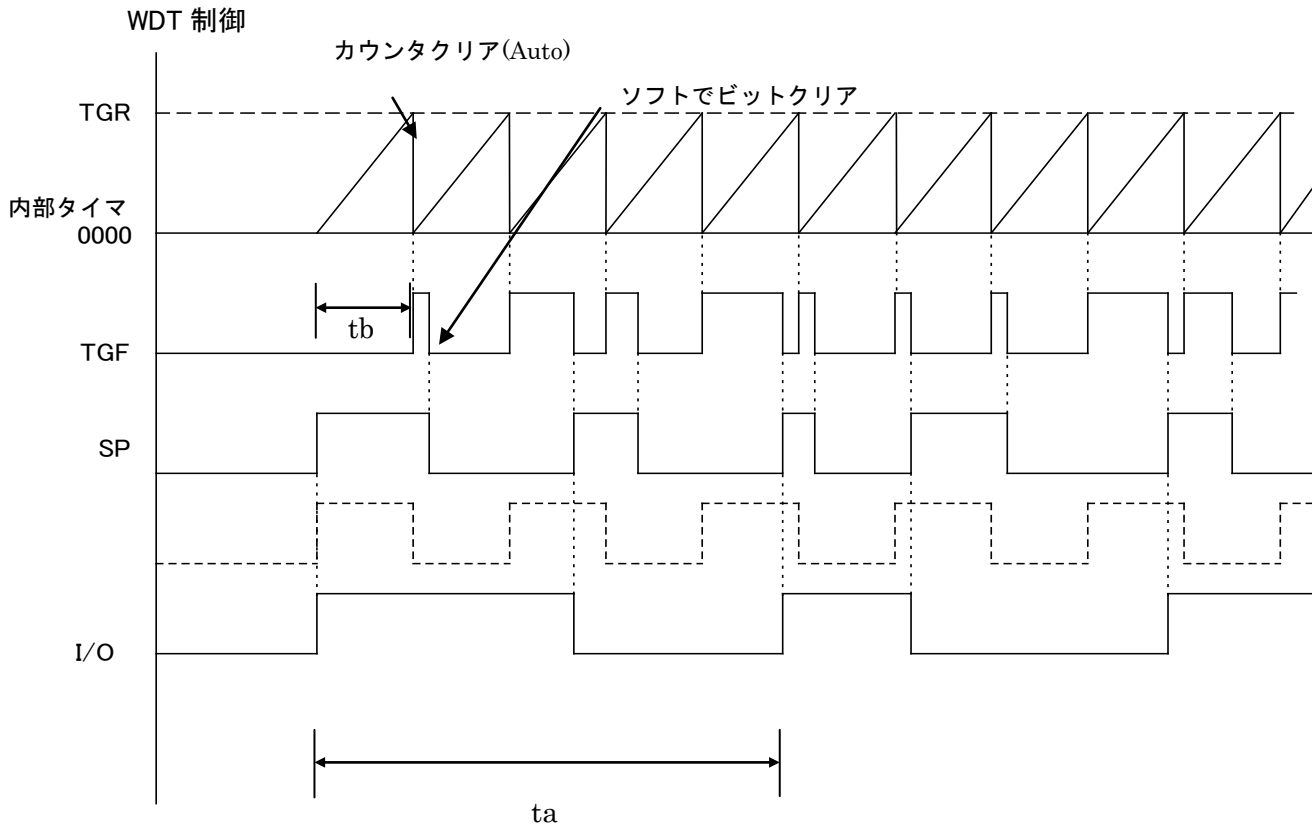
Reset 割り込みが入ったときは、外部回路を含めてリセットスタートします。



## 6. 15 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. 11 I/O ポートサービス対応フラグ」・「5. 12 I/O ポートサービス周期」・「5. 13 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~)

上記図 I/O は SP の2分周値ですが、tb を小さくし分周値を上げることで誤差が小さくなります。

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>TAUA</u>	不可
ta	ta=t1+tb+tc ms (デフォルト t1=6) t1...min 3ms (=tb×6 値) tc...max 3.5ms 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>0</u> bit <u>0</u>	可

※tc はルネサスエレクトロニクス社製セルフライブラリの処理+弊社プログラムの処理にかかる時間で、セルフライブラリ実行中には I/O ポートサービスが行えないためにかかる時間です。

※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. 16 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 ヘリターンするまでの処理時間です。

動作周波数 80MHz で測定

項目	内容	ユーザ設定
APL SUM 域 4Kbyte	12ms	不可
APL SUM 域 1byte	11ms	不可

## 7. r Entryモード仕様

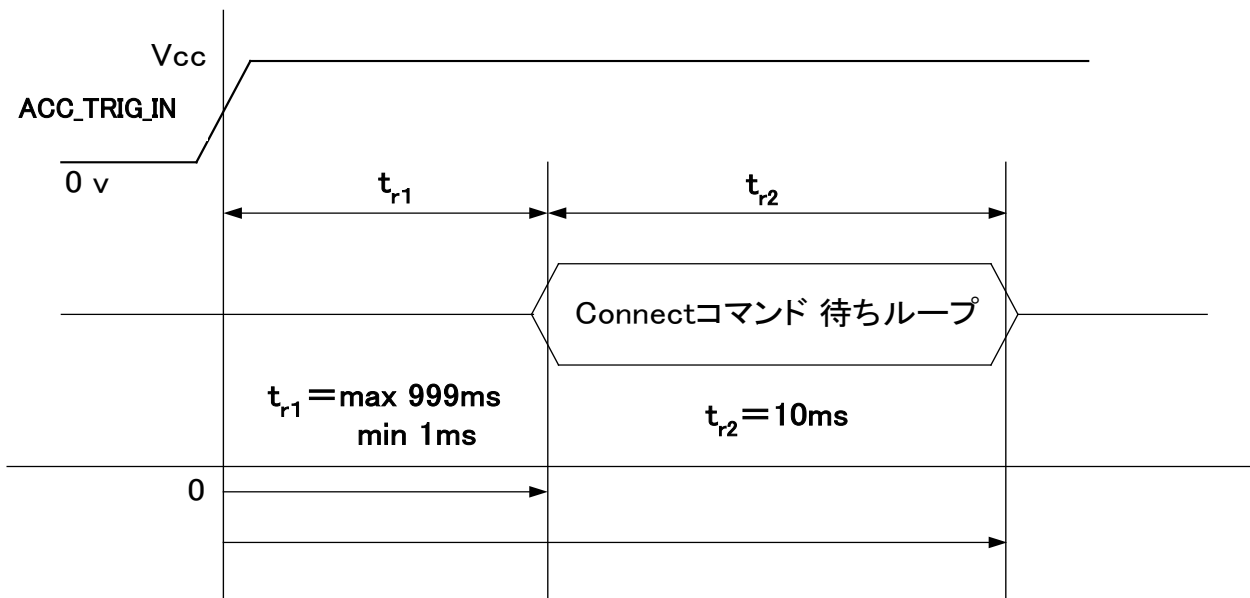
### 7.1 概要

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 10ms間Connectコマンドを待ちます。この約 10ms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の 12V 系マイコン駆動電源電圧ラインに ACC\_TRIG\_IN 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

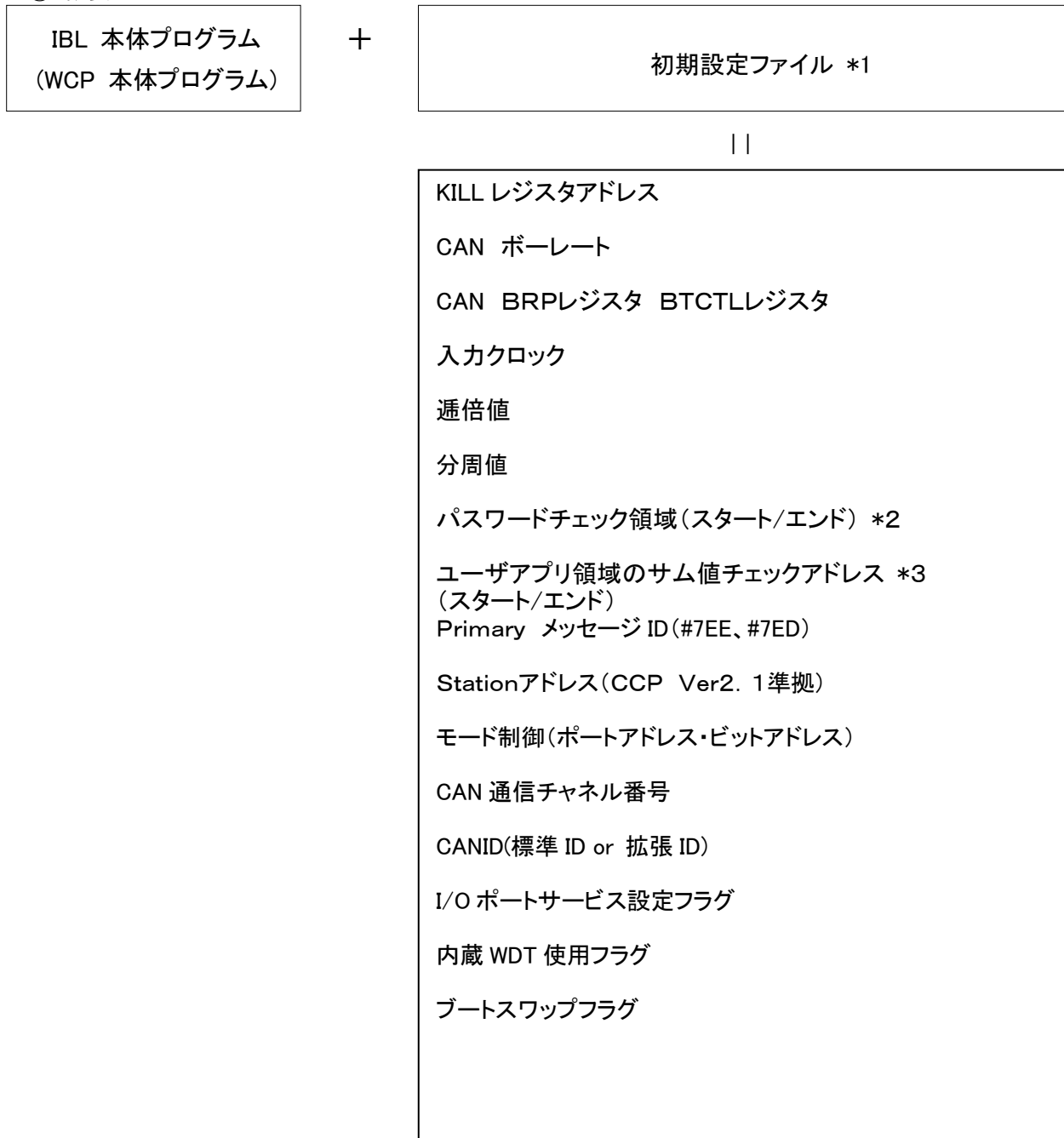
3. ターゲット側の $t_{r1}$ 時間分タイミングを合わせるために、ライター側のConnectコマンド発行タイミングをair ConnectのSpecific Parameterのアドレス#14C,#14Dの 2byteに設定してください。

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	#800h~#8FFh (推奨値)	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内容は保持されています。

## 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(volatile WORD \*sp) 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。 WDT のクリアを行います。
引数	* sp 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\_timer(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。 WDT の初期化を行います。 I/O ポートサービス用のポートを変更したい場合には、この関数内を変更してください。
引数	なし
戻り値	なし

- ・ init\_mbuf(void) 関数

内容	CAN メッセージバッファの初期化をします。
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(BYTE type) 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	type
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(DWORD\* 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, WORD\* piosdiv) 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 I/O サービス用分周カウンタ
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」 station アドレスエラーの場合「2」を返します。

- init\_CAN(void) 関数

内容	CAN 初期設定関数です。 CAN 用ポート、レジスタの初期化を行います。
引数	なし
戻り値	なし

- uApli\_Sum(void) 関数

内容	ユーザアプリ SUM 値計算領域の SUM 値を計算します。
引数	なし
戻り値	SUM 値

・ 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(volatile WORD \*sp) 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。 WDTのクリアを行います。
引数	*sp 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

- ・ SetFLMDHigh(void) 関数

内容	FLMD設定ポートに指定したポートからHigh出力を行います。
引数	なし
戻り値	なし

- ・ SetFLMDLow(void) 関数

内容	FLMD設定ポートに指定したポートからLow出力を行います。
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(BYTE type) 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	type
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とフラッシュプログラミング環境からユーザ環境への遷移を行い、無限ループに入ります。
引数	なし
戻り値	なし

- flash\_enter(void) 関数

内容	FSL_FlashEnv_Activate 関数(セルフライブラリ)を使用し、ユーザ環境からフラッシュプログラミング環境に遷移させます。
引数	なし
戻り値	FlashFLMDCheck 関数戻り値

- flash\_exit(void) 関数

内容	FSL_FlashEnv_Deactivate 関数(セルフライブラリ)を使用し、フラッシュプログラミング環境からユーザ環境に遷移させます。
引数	なし
戻り値	なし

- get\_block(u32 addr) 関数

内容	ブロック番号取得関数です。 引数の値がブロックのラストアドレスの場合、そのブロック番号を返します。
引数	addr アドレス
戻り値	引数のアドレスがブロックのラストアドレスの場合、そのブロック番号を返し、ラストアドレスでない場合には 0xFF を返します。

- 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ー側に送信します。 データ読み出し前に、フラッシュプログラミング環境からユーザ環境に遷移し、読み出し後はフラッシュプログラミング環境に戻します。
引数	なし
戻り値	なし

- recvdata(void) 関数

内容	書き込みデータの受信を行います。
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。 また、セキュリティ情報の書き込みも行います。
引数	なし
戻り値	なし

- set\_boot\_info(void) 関数

内容	ブートブロック情報の設定を行います。 デバイスファンクションの実行時に使用される変数を設定します。
引数	なし
戻り値	なし

- erase\_old\_boot(void) 関数

内容	ブートブロックの消去処理を行います。
引数	なし
戻り値	消去完了フラグ

- 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。 block_erase 関数を Call し、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ー側に送ります。
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。 ブランクチェックは行わず、イレーズ実行後ならば正常値を返し、イレーズが行われていない場合には異常値を返します。
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。 データの読み出しを行う前に、フラッシュプログラミング環境からユーザ環境に 遷移し、読み出し後はフラッシュプログラミング環境に戻します。
引数	なし
戻り値	なし

- wcp\_init(void) 関数

内容	初期化を行います。 セルフプログラミングを行えるよう設定し、FSL_Init()を実行します。
引数	なし
戻り値	0: 正常終了 1: 異常終了

- 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



## 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_0 bit_0	I/O サービスに使用	
MCU 内蔵タイマ	タイマTAUA	サイクルアクセス用	

## 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	0x00001FF0	
CAN ポーレート	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 (BRP)	CAN_BRP	グローバルクロック選択レジスタ値 8BIT	0x04	可
CANビットレートレジスタ	CAN_BRT_DATA	モジュールビットレートレジスタ値	0x0508	可
入力クロック	CLK_EXT	MHz×10で指定 例)10MHZ = 100	80 (8MHz)	可
逡倍値	CLK_MUL	1~n	10	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	0x00000800	可
パスワードチェック領域 (エンド)	PASS_END	32bit アドレス指定	0x000008FF	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_START	32bit アドレス指定	0x0001F000	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	0x0001FFFF	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	0	可
I/O ポートサービス 周期	IOS_PERIOD	ms 単位(3ms~65535ms)	6	可
CANID	CAN_ID	0 or 1 標準 ID:0 拡張 ID:1	0	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	0x00	可
内蔵 WDT クロック	WDT_MD	WDTA0MD レジスタ値	0x77	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	0x01	可
CAN チャンネル番号	CAN_CHAN	0~n チャンネル	0	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	MENU:SUB SETTING CAN AF →TGT ID FMT	“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	MENU:SUB SETTING CAN TGT →AF ID FMT	“STANDERD”/“EXTENDED”を選択	STANDARD	可
CAN 通信ボーレート設定 *1	MENU:SUB SETTING CAN BAURATE SETTING	500Kbps/1Mbps/250Kbps/125Kbps を選択	500K	可
CAN ID 設定 *1	MENU:SUB SETTING CAN ID SET	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	MENU:SUB SETTING KILL WRITE	KILL レジスタを ON に設定します		不可
ブランクチェックモード設定	MENU:SUB SETTING BLANK MODE	“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	#0001F000	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 等で Specific Parameter #144 から 4BYTE 指定する	#0001FFFF	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 Parameter #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。