

NAPYDC839GS01 共通 CAN プログラミング技術資料

NO	マイコンパック型名	対応 MCU 名
1	NAPYDC839GS01	SPC570S50(Velvet)
2		
3		
4		

<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. DTS インサイトは、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

NAPYDC839GS01 の適用 NETIMPRESS シリーズ 本体は、NETIMPRESS air(AF930)です。

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

変更履歴

変更日付

2019.03.11

変更内容

新規作成

目次

1. 概要.....	1
1.1 動作条件.....	1
1.2 マイコンパック内容.....	2
2. 用語の定義と略語.....	4
3. 機能概要.....	7
3.1 UCOP システム構成図.....	7
3.2 ROM.....	8
3.2.1 ROM 構成図.....	8
3.2.2 消去ブロックアドレス.....	9
3.3 プログラムエントリーモードフローチャート.....	10
3.4 IBL プログラム概略フローチャート.....	11
4. 初期導入手順.....	12
4.1 書き込み手順フロー.....	12
4.2 設定変更項目.....	13
4.2.1 CANボーレートの変更.....	13
4.2.2 動作クロックの変更.....	13
4.2.3 ビットタイミングパラメータの変更.....	13
4.2.4 パスワードチェック領域の変更.....	13
4.2.5 ユーザアプリ領域サム値チェック領域の変更.....	13
4.2.6 ウォッチドッグタイマサービスの変更.....	13
4.2.7 Primary IDの変更.....	13
4.2.8 ステーションアドレスの変更.....	13
4.2.9 CAN チャンネルの変更.....	13
4.2.10 内蔵ウォッチドッグタイマサービスの変更.....	14
5. UCOP 設定変更方法.....	15
5.1 CAN ボーレート.....	15
5.2 クロック周波数.....	16
5.3 ビットタイミングレジスタ値.....	17
5.4 パスワードチェック領域開始アドレス.....	17
5.5 パスワードチェック領域終了アドレス.....	17
5.6 ユーザアプリ領域サム値チェック開始アドレス.....	18
5.7 ユーザアプリ領域サム値チェック終了アドレス.....	18
5.8 I/Oポートサービス対応フラグ.....	19
5.9 I/Oポートサービス周期.....	19
5.10 I/Oポートサービス用ポート変更方法.....	19
5.11 Primary ID.....	19

5. 12	CAN ID フォーマット設定	22
5. 13	ステーションアドレス	23
5. 14	CAN チャンネル番号	23
5. 15	内蔵ウォッチドッグタイマ設定	23
5. 16	内蔵ウォッチドッグタイマクリア周期設定	24
5. 17	Specific Parameter 変更方法	25
6.	UCOP システム概要	29
6. 1	イニシャル・プロセッシング・ルーチン (IPR)	29
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)	30
6. 3	書き込み制御プログラム (WCP)	30
6. 4	書き込みプロセス正常終了判定	31
6. 5	アイデンティファイヤ (CAN メッセージ ID)	32
6. 5. 1	Primary ID	32
6. 5. 2	Secondary ID	32
6. 5. 3	送受信メッセージバッファ	32
6. 5. 4	チャンネル	33
6. 6	ステータスレジスタ	34
6. 7	プログラムエントリモード	35
6. 8	u Entry 時ユーザ APL 処理項目	36
6. 9	KILL レジスタ	37
6. 10	CAN ボーレート設定時の注意	38
6. 11	ステーションアドレス	38
6. 12	プログラム終了時の処理	38
6. 13	ウォッチドッグタイマ	39
6. 14	IBL 処理時間	40
7.	r Entry モード仕様	41
7. 1	概要	41
7. 2	r Entry モード使用方法	42
8.	YDC製IBL、WCPの構成	43
9.	RAMの使用方法	44
10.	CAN プロトコル	45
10. 1	フレームの種類	45
10. 2	IBL 対応コマンド	45
10. 3	WCP 対応コマンド	46
11.	関数一覧	47
11. 1	IBLでの使用関数 (y_ibl. cファイルの関数一覧)	47
	・ io_service(void) 関数	47
	・ init_pit(void) 関数	47
	・ RecData(void) 関数	47
	・ TrmData(void) 関数	47

▪ TrmReady(void) 関数	48
▪ TrmBusy(void) 関数	48
▪ TrmError(void) 関数	48
▪ exec_reset(void) 関数	48
▪ command_GetCcpVersion(void) 関数	48
▪ command_ExchangeID(void) 関数	48
▪ command_password_check(void) 関数	49
▪ command_ExtSumcheck(void) 関数	49
▪ get_wcp(DWORD * paddr) 関数	49
▪ RecConnect(BYTE tout) 関数	49
▪ ibl_main(void) 関数	49
▪ ibl_entry(WORD ent) 関数	49
11. 2 WCPでの使用関数(y_wcp. cファイルの関数一覧)	50
▪ io_service(void) 関数	50
▪ RecData(void) 関数	50
▪ TrmData(void) 関数	50
▪ TrmReady(void) 関数	50
▪ TrmBusy(void) 関数	50
▪ TrmError(void) 関数	50
▪ exec_reset(void) 関数	50
▪ Command_Read(void) 関数	51
▪ Command_Program(void) 関数	51
▪ Command_BlockErase(void) 関数	51
▪ Command_ReadSRD(void) 関数	51
▪ Command_ClearSRD(void) 関数	51
▪ Command_ExtBlankcheck(void) 関数	51
▪ Command_ExtSumcheck(void) 関数	52
▪ wcp_main(void) 関数	52
12. 使用I/Oリソース一覧	53
13. 付録	54

1. 概要

1.1 動作条件

項目	内容	ユーザ設定※
対象マイコン	SPC570S50(Velvety)	
書き込み対象アドレス	Code Block: #00FC0000h~#0103FFFFh Data Block: #00800000h~#00807FFFh	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
コア動作クロック	入力= 40MHz、動作=80MHz 逡倍値=24、分周値=12	可
CAN チャンネル番号	チャンネル 0~1 (default 0)	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	NETIMPRESS : AZ414 or AZ415	不可
CCP バージョン	Version 2.1 をベースとする *1	不可
開発環境	MULTI v6.1.4 / PPC.V2013.5.4 *2	不可
コンパイラバージョン	↑	不可
最適化	最適化なし	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	128byte	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 CCP を拡張したプロトコルです。完全互換性はありません。

*2 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。

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-NAPYDC839GS01	
CAN 共通プログラミング 技術資料	TR-NAPYDC839GS01	本書
マイコンパック	NAPYDC839GS01	・FDF シート内容 BTP ファイル ^{※1} AMK ファイル KEY ファイル YSM ファイル サンプル APL オブジェクト ^{※2}
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_NAPYDC839GS01 -FC839S2S01. gpj -make_btp. bat -V1500NAPYDC839GS01. BTP -IBL. run -Header -y_ibl. h -y_init. h -user_init. h -peripheral. h -src_IBL -IBL. gpj -standalone_romrun. ld -user_ipr_start. ppc -y_ibl_init. ppc -user_ipr. c -y_ibl. c -user_apl_start. ppc -user_apl. c -src_WCP -WCP. gpj -standalone_ram. ld -y_wcp_init. ppc -y_wcp. c -driver -BlankCheck. c -CheckSum. c -FlashArrayIntegrityCheck. c -FlashArrayIntegrityResume. c -FlashArrayIntegritySuspend. c -FlashCheckStatus. c -FlashErase. c -FlashEraseAlternate. c -FlashInit. c -FlashProgram. c -FlashResume. c	トッププロジェクトファイル BTP作成/バッチファイル CAN リブログ用書込制御プログラム CAN リブログ用ブートロータ 固有値定義ファイル 初期設定ファイル エントリアドレス定義ファイル 周辺デバイス定義ファイル IBL 用サブプロジェクトファイル リンクスクリプトファイル ユーザ IPL スタートアップサンプル IBL スタートアップルーチン ユーザ IPR サンプルファイル IBL/IPR (例) オブジェクト ^{※3} ユーザ APL スタートアップルーチンサンプル APL サンプルファイル WCP 用サブプロジェクトファイル リンクスクリプトファイル WCP スタートアップルーチン WCP 本体 フラッシュ制御ライブラリ (SSD)

	-FlashSuspend.c -GetLock.c -OverPgmProtGetStatus.c -ProgramVerify.c -SetLock.c -ssd_c55.h -ssd_c55_internal.h -ssd_types.h	
ブート(IBL/IPR)書き込み用 マイコンパック	FS719M00 又は FS819M00	・FDFシート内容 IBL、IPR(例)オブジェクト

※1 BTP ファイルは「Header , WCP」フォルダ内のファイルをコンパイルすることにより
作成されるファイルです。

※2 サンプル APL オブジェクトは「Header , IBL」フォルダ内のファイルをコンパイルすることにより
作成される IBL.hex に含まれます。

※3 IBL/IPR (例) オブジェクトは「Header , IBL」フォルダ内のファイルをコンパイルすることにより
作成される IBL.hex ファイルです。

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。

本マイコンパックではご利用いただけません。

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 リプログモードを強制終了するかどうかを判定する機能です。

本マイコンパックではご利用いただけません。

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。

SSD

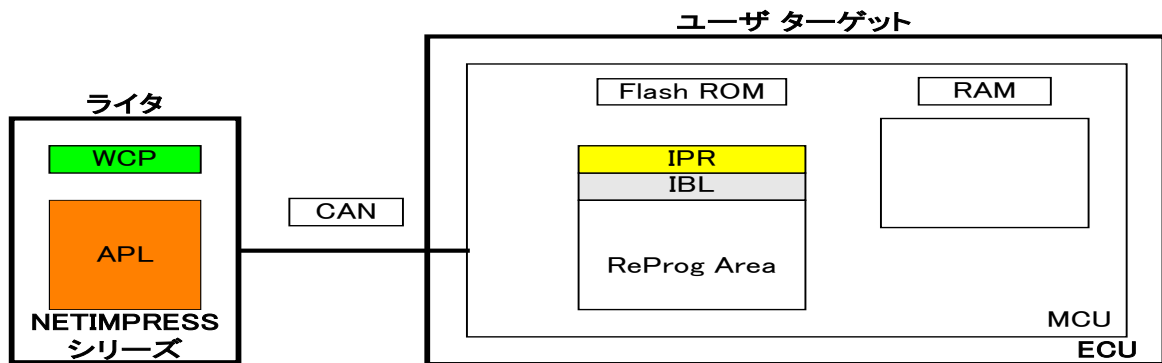
Standard Software Driver の略です。
STマイクロエレクトロニクス社より提供されており、内蔵 Flash に対する書き換え処理を行う API を提供します。
WCP に組み込まれ、消去・書き込みなどを実行します。

BOOT HEADER

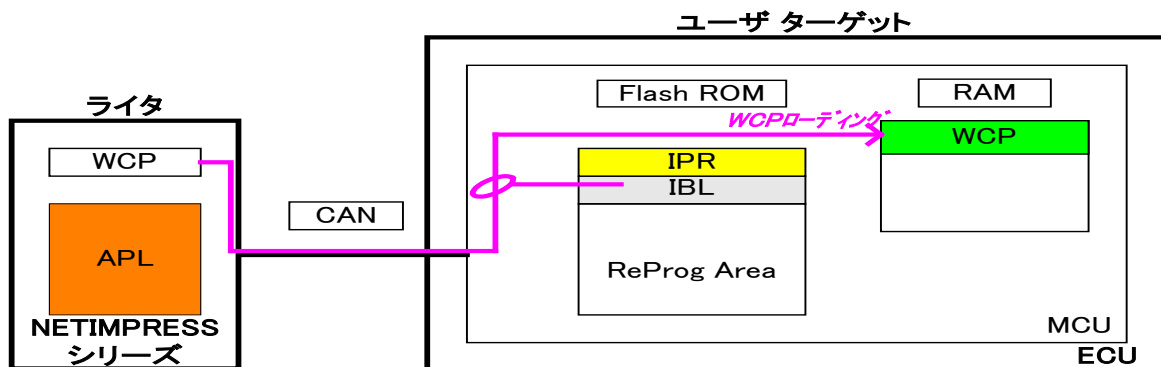
ブートセクタの先頭に配置します。

3. 機能概要

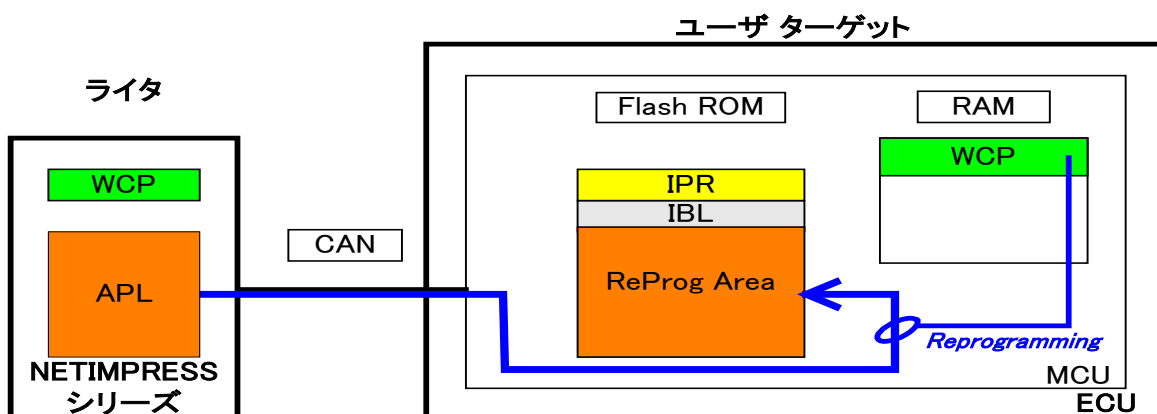
3.1 UCOP システム構成図



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。

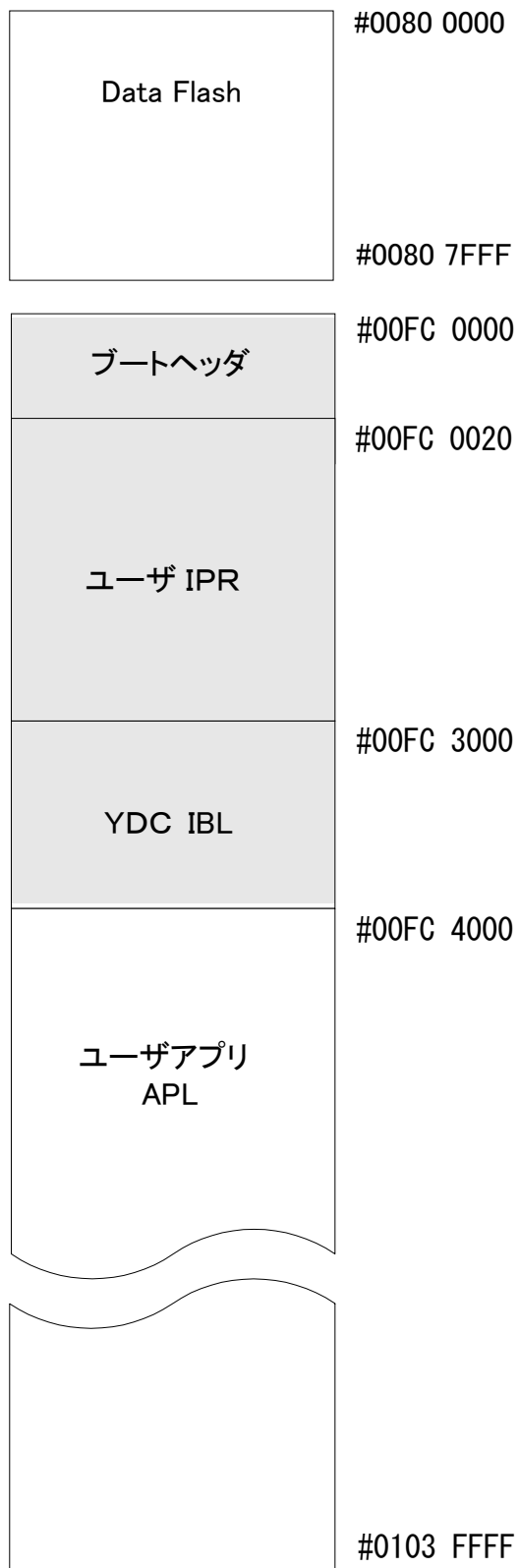


6. ライタは WCP と通信をおこないライターにある APL を ReProg Area に書き込みます

3. 2 ROM

3. 2. 1 ROM構成図

灰色部は書き換え禁止領域。



3. 2. 2 消去ブロックアドレス

Data Flash

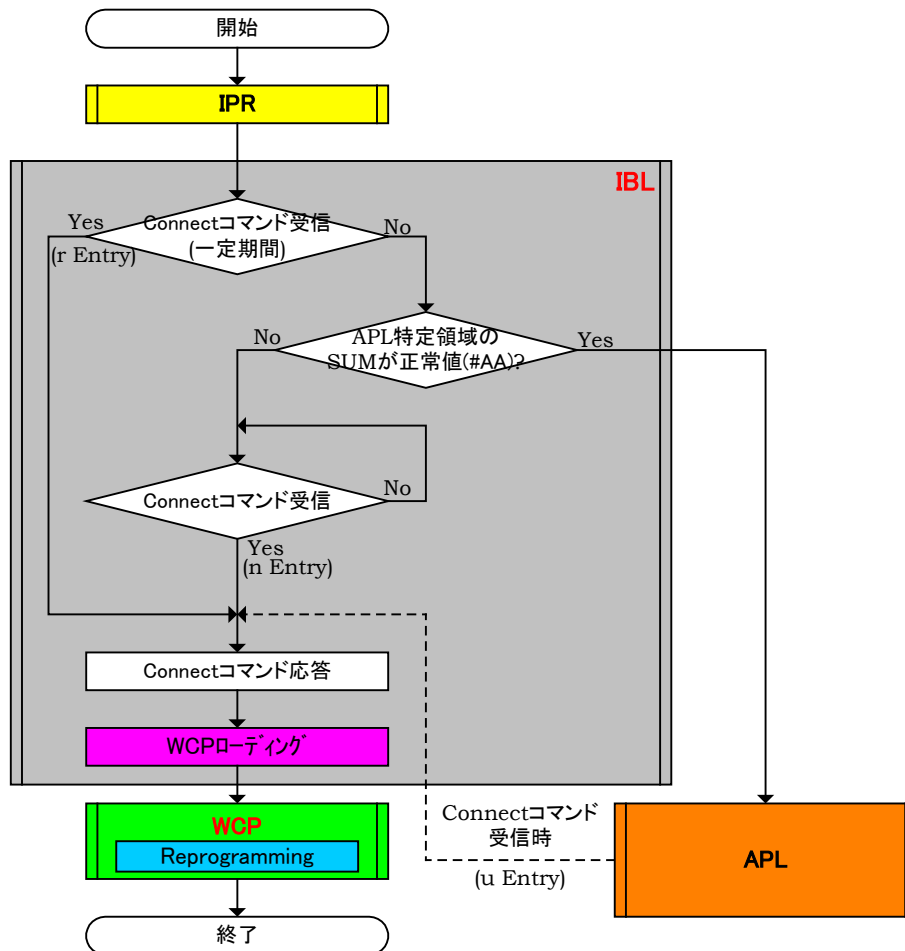
グループ	アドレス
グループ 0	#0080 0000～#0080 1FFF
グループ 1	#0080 2000～#0080 3FFF
グループ 2	#0080 4000～#0080 5FFF
グループ 3	#0080 6000～#0080 7FFF

Code Flash

グループ	アドレス
グループ 4	#00FC 0000～#00FC 3FFF
グループ 5	#00FC 4000～#00FC 7FFF
グループ 6	#00FC 8000～#00FC BFFF
グループ 7	#00FC C000～#00FC FFFF
グループ 8	#00FD 0000～#00FD 7FFF
グループ 9	#00FD 8000～#00FD FFFF
グループ 10	#00FE 0000～#00FE FFFF
グループ 11	#00FF 0000～#00FF FFFF
グループ 12	#0100 0000～#0101 FFFF
グループ 13	#0102 0000～#0103 FFFF

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(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(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。

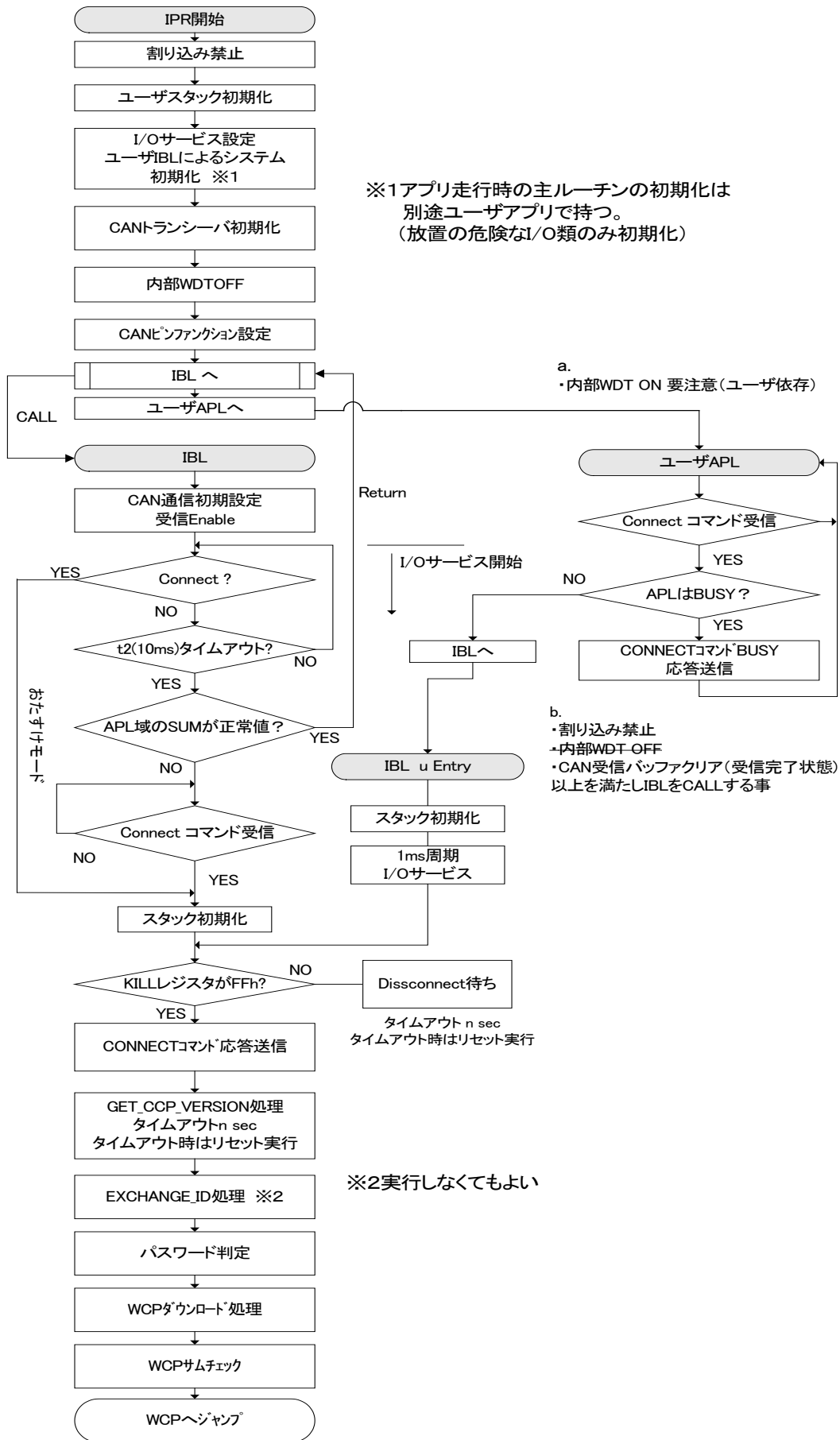
※ライタは Connect コマンド発行後規定時間(25ms)応答がない場合、Disconnect コマンドを発行し、ターゲット MCU との接続を解除します。

※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。

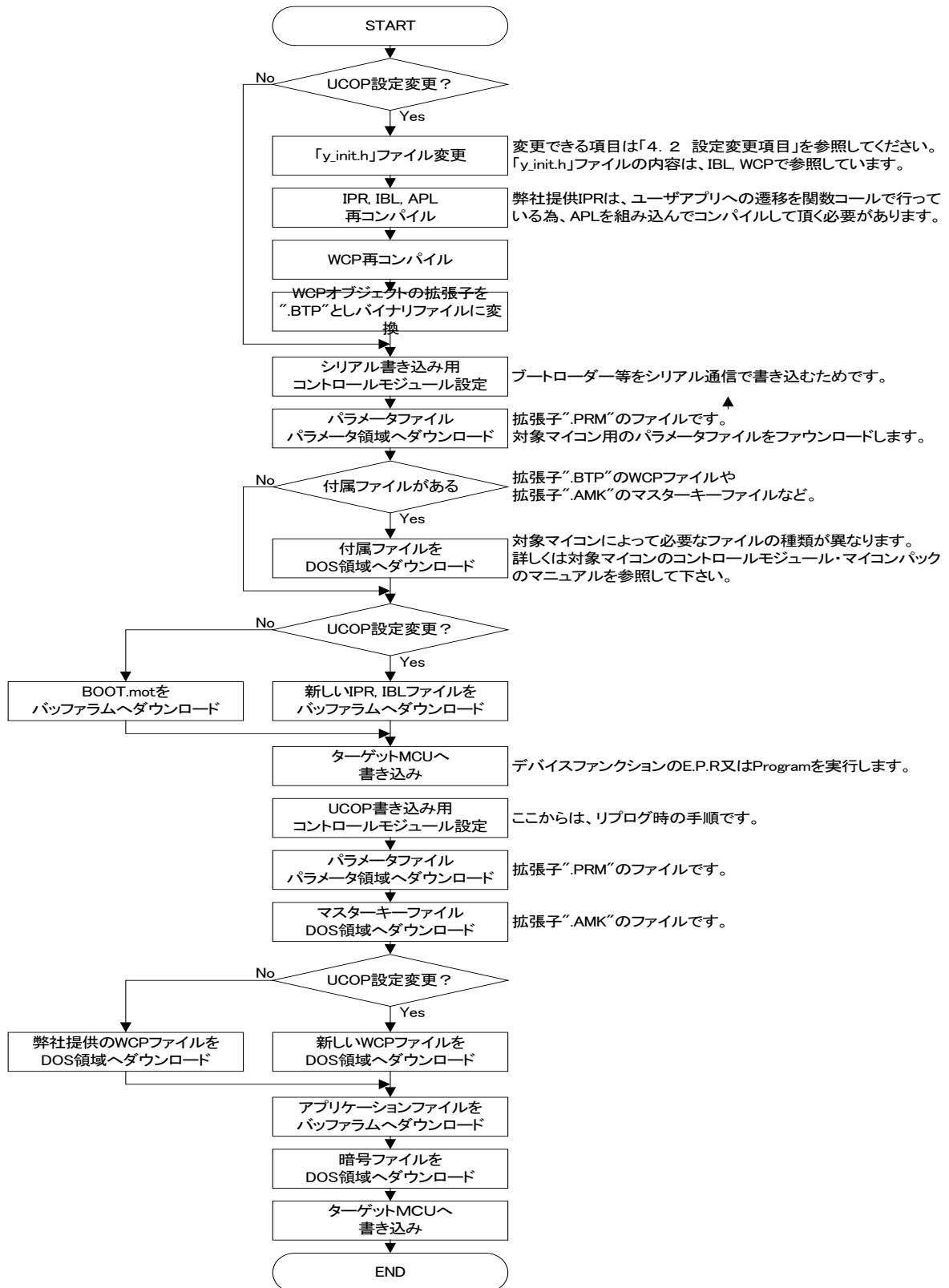
※ターゲット MCU は Disconnect コマンド受信後、リセット状態に戻るものとします。

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. 3 ビットタイミングレジスタ値」を参照してください。

4. 2. 2 動作クロックの変更

「5. 2 クロック周波数」、「5. 3 ビットタイミングレジスタ値」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の変更

「5. 3 ビットタイミングレジスタ値」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. 4 パスワードチェック領域開始アドレス」、「5. 5 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. 6 ユーザアプリ領域サム値チェック開始アドレス」、「5. 7 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. 8 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. 9 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. 10 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. 11 Primary ID」、「5. 12 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. 13 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. 14 CAN チャネル番号」を参照してください。

4. 2. 10 内蔵ウォッチドッグタイマサービスの変更

4. 2. 10. 1 内蔵ウォッチドッグタイマサービス有無の変更

「5. 15 内蔵ウォッチドッグタイマ設定」を参照してください。

4. 2. 10. 2 ウォッチドッグタイマクロックの変更

「5. 16 内蔵ウォッチドッグタイマクリア周期設定」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 の 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本体が FULL KEY モデル(AF620)のものにおいては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タイミングレジスタ値も再計算し、初期設定ファイルの

” CAN_CR_DATA”も再設定してください。

「5. 3 ビットタイミングレジスタ値」をご参照ください。

②. ライター側設定変更

FULL KEY モデルでは Func-83 で変更することが出来ます。

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 25	0x 40
500 Kbps	0x 16	0x 01
250 Kbps	0x 16	0x 03
125 Kbps	0x 16	0x 07

※ 「5. 17 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 クロック周波数

ターゲット MCU の動作クロック周波数を変更した場合には、初期設定ファイル“y_init.h”に反映させる必要があります。

① コアクロックの変更

“CLK_SYS”にコアクロック周波数 MHz 値を 10 倍した値を設定してください。

例)80MHz の場合、800 と設定

② CAN クロックの変更

“CLK_CAN”に CAN モジュールクロック周波数 MHz 値を 10 倍した値を設定してください。

例)20MHz の場合、200 と設定

※必ずマイコンのビットタイミングレジスタ値も再計算し、

初期設定ファイルの” CAN_CFG_DATA”も再設定してください。

「5. 3 ビットタイミングレジスタ値」をご参照ください。

③ PIT クロックの変更

“CLK_PIT”に PIT モジュールクロック周波数 MHz 値を 10 倍した値を設定してください。

例)40MHz の場合、400 と設定

④ SWT クロック

SWT は IRC で動作しますので 16MHz に固定です。変更しないでください。

“CLK_SWT”には IRC クロック周波数として 160 を設定してください。

5.3 ビットタイミングレジスタ値

ビットタイミング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。

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_CR_DATA”に、CAN コントロールレジスタへの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 クロック周波数」をご参照ください。

5.4 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。

5.5 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。

5.6 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。

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 でのみの変更となります。

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「0xFFAABBOO」と設定する場合

Specific Parameter のアドレス	#140	#141	#142	#143
設定値	0xFF	0xAA	0xBB	0x00

※ 「5.17 Specific Parameter 変更方法」を参照下さい。

開始アドレスは256byte 境界となるよう設定して下さい(詳細は 6-4. 参照)

5.7 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。

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 でのみの変更となります。

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「0xFFEEDDDFF」と設定する場合

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

※ 「5.17 Specific Parameter 変更方法」を参照下さい。

終了アドレスはサイズが256byte 区切りとなるよう設定して下さい(詳細は 6-4. 参照)

5. 8 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 (6-13. 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. 9 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. 10 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は、ipr で初期化する必要があります。

また、使用するポート番号を“y_init.h”の“IOS_PORT”に設定してください。

5. 11 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_NI”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_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
			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_NI”のスタンダード・アイデンティファイヤを「7ED」、 “ID_P_MCU” のスタンダード・アイデンティファイヤを「7EE」と設定する場合

“ID_P_NI”: 0x000007ED

“ID_P_MCU”: 0x000007EE

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

“ID_P_NI”: 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. 12 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

FULL KEY モデルでは Func-86 を用いて変更することができます。

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. 17 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 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_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 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. 12 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. 11 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. 13 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. 11 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 でのみの変更となります。

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. 17 Specific Parameter 変更方法」を参照下さい。

5. 14 CAN チャネル番号

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。

①. 初期設定ファイル“y_init.h”設定変更

“CAN_CHAN”に「0」か「1」を設定します。

「0」の場合: チャンネル0を使用

「1」の場合: チャンネル1を使用

5. 15 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対応フラグを変更することでウォッチドッグタイマの有効/無効を設定します。

内蔵ウォッチドッグタイマを使用する場合には、iprで初期化を行ってください。

本サンプルでは内蔵ウォッチドッグタイマを使用しません。

内蔵ウォッチドッグタイマ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_WDT_ON”に「0」か「1」を設定します。

「1」の場合: ウォッチドッグタイマ有効

「0」の場合: ウォッチドッグタイマ無効

※対象マイコンのウォッチドッグタイマのクリア手順には二種類ありますが、本サンプルでは、Fixed service sequence mode に対応しています。

5. 16 内蔵ウォッチドッグタイマクリア周期設定

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。
"IOS_WDT_PERIOD" にミリ秒単位でクリア周期を設定してください。

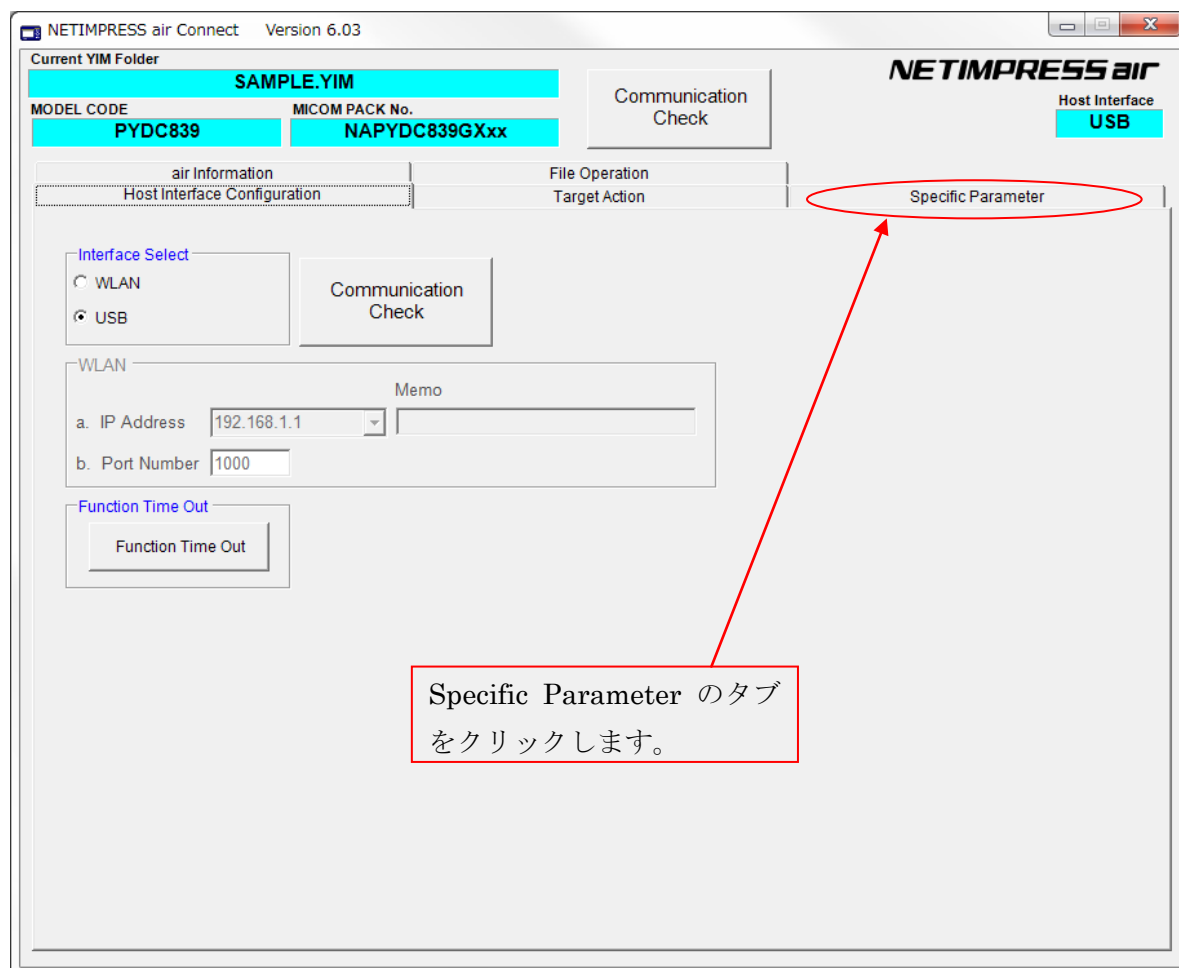
※APL から IBL にジャンプ(uEntry)した後、"IOS_WDT_PERIOD" の設定値に関わらず、1ms 後にウォッチドッグタイマのクリアを行います。

5. 17 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。

NETIMPRESS air Connect Version 6.03

Current YIM Folder: **SAMPLE.YIM**

MODEL CODE: **PYDC839** MICOM PACK No.: **NAPYDC839GXxx**

Communication Check

Host Interface: **USB**

air Information | Host Interface Configuration | File Operation | Target Action | Specific Parameter

Specific Parameter

	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

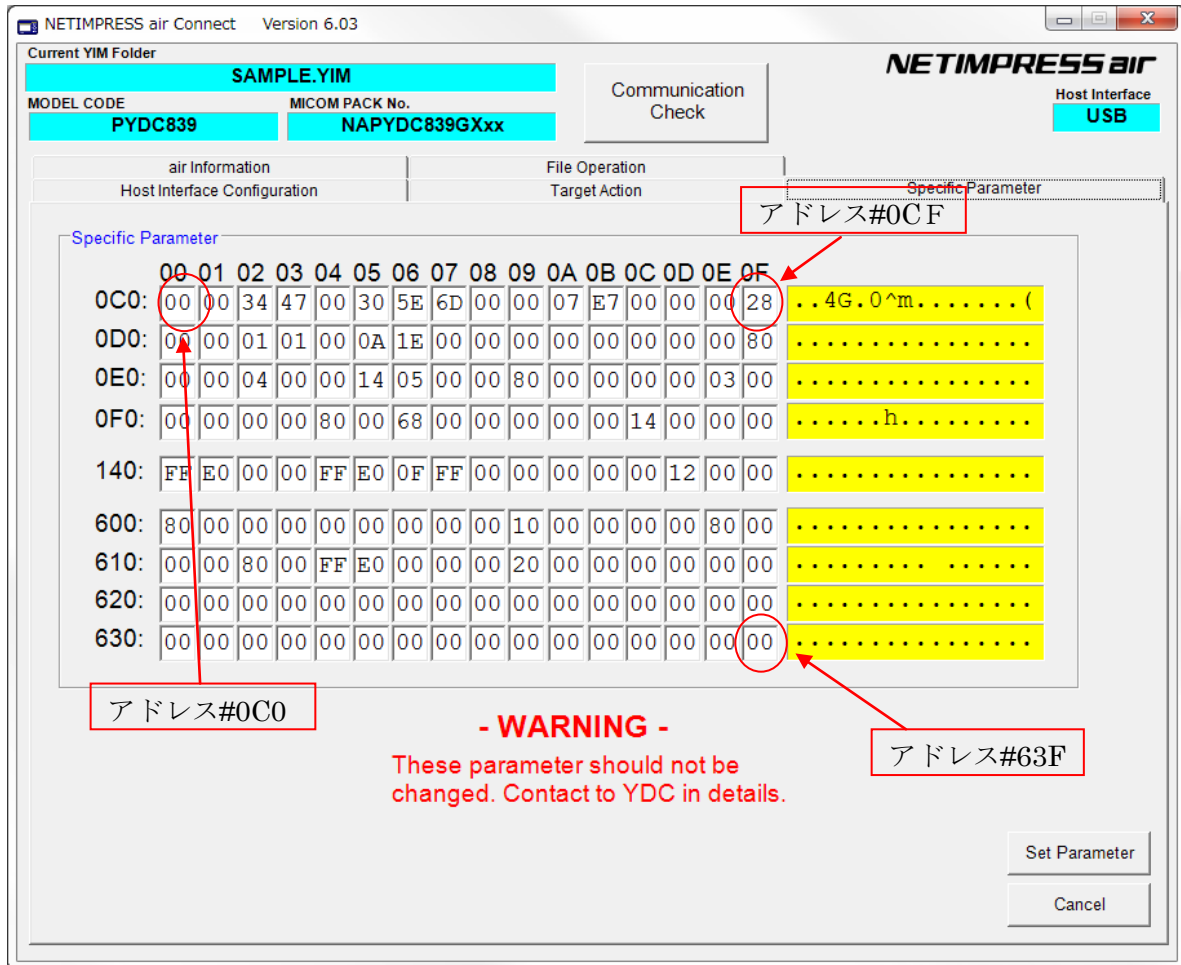
- WARNING -
 These parameter should not be changed. Contact to YDC in details.

Set Parameter

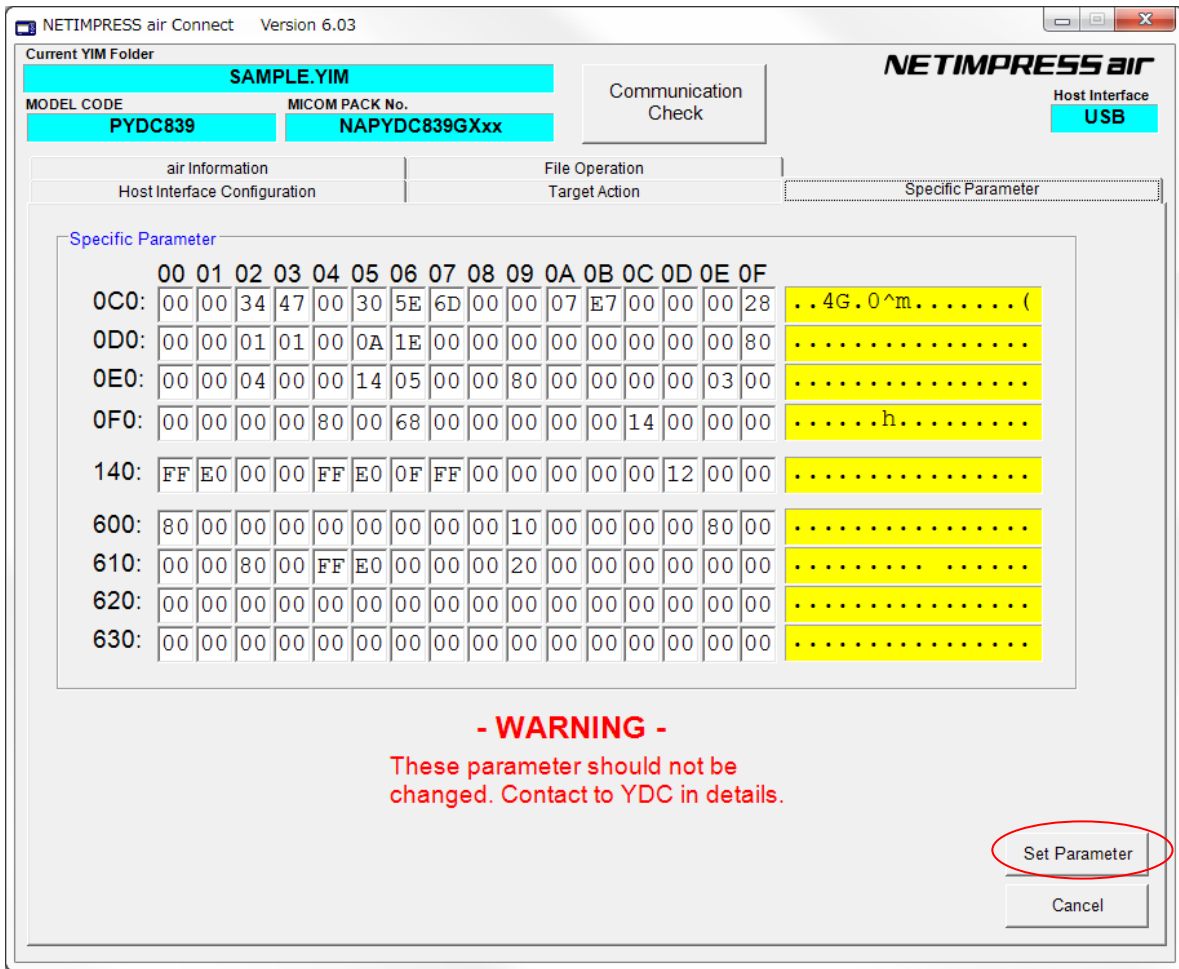
Cancel

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”OK”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。

リセット解除後ブートヘッダ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	#00FC0000h~#00FC2FFFh (スタートアドレス:#0FC00020h)	不可
IPR 要初期化項目	<ul style="list-style-type: none">■ I/O サービスポートの入出力定義■ 割り込み禁止■ スタック初期化■ クロック設定■ CAN ポート設定<input type="checkbox"/> 内蔵 WDT 起動(必要に応じて実施してください)	可

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	#00FC3000~#00FC3FFF	不可
IBL エントリーアドレス From IPR	#00FC3000h	不可
IBL エントリーアドレス From APL	#00FC3010h	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。

※ 最適化レベル なし。

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	#40001000~#40002FFF	不可

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“256byte~4Kbyte”の範囲で規定するものとします。

ii. 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。

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. サイズは256byte 区切りとしてください。

v. 書き込み禁止領域中には設定しないで下さい。

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	#00FDFF00~#00FDFFFF (デフォルト値※)	可

本領域はあくまで正常判定エリアですのでこの領域にはアプリケーションデータを入れないことを推奨します。

※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。

6. 5. 1 Primary ID

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

本マイコンパックでは **Secondary Message ID は使用できません。**

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オブジェクトとして CAN メッセージバッファ 30 を使用します。
- ・IBLは送信用メッセージオブジェクトとして CAN メッセージバッファ 31 を使用します。
- ・ID 完全一致として使用します。
- ・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 メッセージオブジェクト	CAN メッセージバッファ 30	不可
IBL 送信用 メッセージオブジェクト	CAN メッセージバッファ 31	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。なお、使用するピンが変わる場合は、user_ipr.c にてピンのコンフィギュレーションの変更が必要です。

(「5. 14 CAN チャンネル番号」参照)

6.6 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。

6.7 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 (マイコンパック指定)

※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. 8 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	■ CAN 設定条件変更不可 ※1 ■ CONNECT コマンド受信 (受信後 20ms 以内(※2)で IBL を CALL する) ※3	可
Connect コマンド受信後の APL 要初期化項目 (u Entry 時) ■: 必須	■ 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	可

※1. IBL で使用するメッセージバッファはの設定は変更しないで下さい。

(「6. 5. 3 送受信メッセージバッファ」参照)

※2. CONNECT 応答規定は 25ms ですが IBL での応答までの処理が約 2ms 以上要するため
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下さい。

・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 n 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(n)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(n)	3 回	可
リトライ間隔	200ms	可

6.9 KILLレジスタ

本マイコンパックではKILLレジスタの書き換えはできません。

6. 10 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。

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. 3 ビットタイミングレジスタ値」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、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. 11 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 (ライター) 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. 13 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. 12 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。

内蔵ウォッチドッグタイマのクリア動作も行いません。

(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)

6. 13 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外付け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)

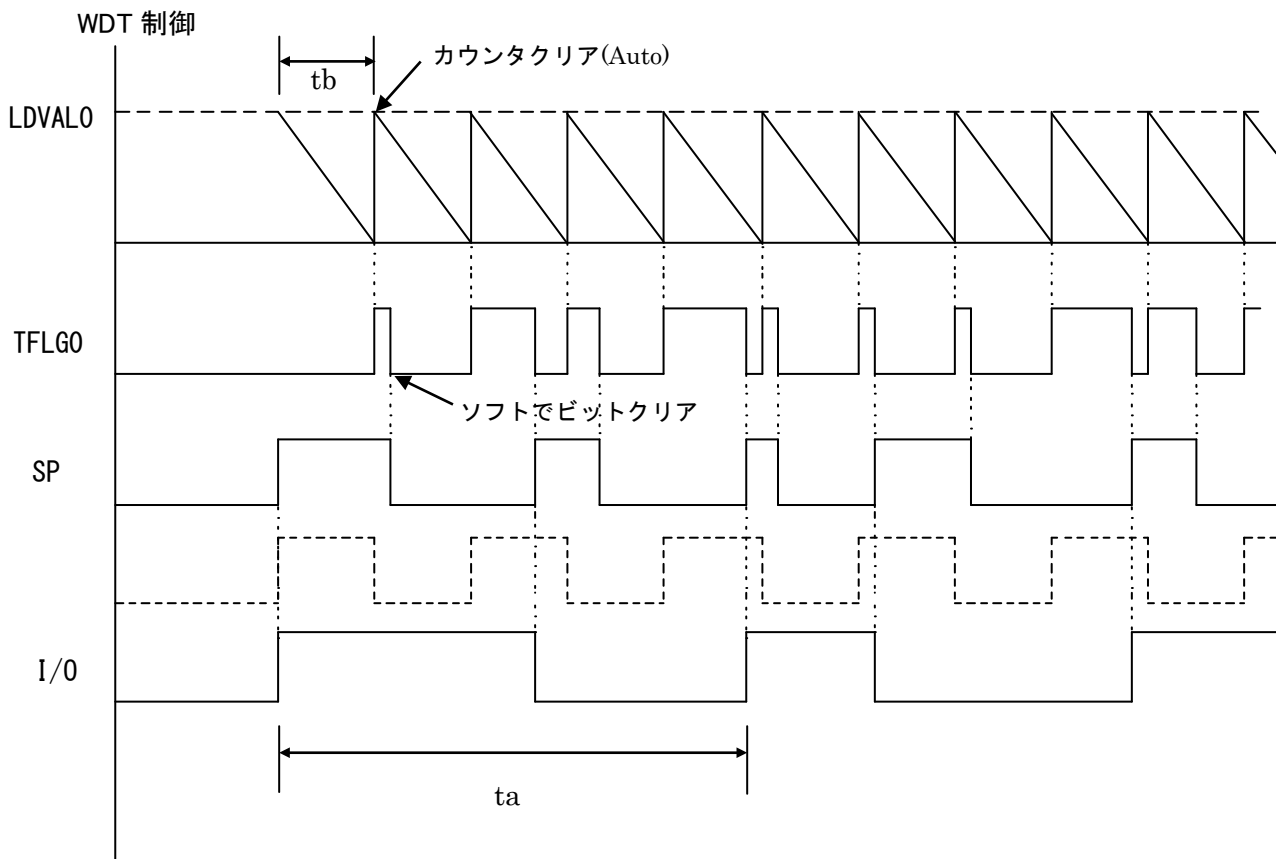
※I/O ポートサービスの有無、アクセス周期変更等は「5. 8 I/O ポートサービス対応フラグ」・「5. 9 I/O ポートサービス周期」・「5. 10 I/O ポートサービス用ポート変更方法」を参照してください。

動作

インターバル時間は t_b (500us)に固定です。

ソフトで TFLG0 を監視、アンダーフローを検出したら t_1 の設定値に応じて分周し、I/O出力をトグルします。割り込みは使用していません。

PIT0の使用は、IBLとWCPの動作中だけです。ユーザーアプリで他の用途にご使用いただくことができます。



LDVAL0: タイマロードレジスタ値

TFLG0: タイマフラグレジスタ

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、 t_b 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>PIT0</u>	不可
ta	ta = t1 + tc ms (デフォルト t1=1) t1...min 1ms t1 の設定は1ms単位とする。	t1 設定可
tb	500usec	
tc	Max: 500usec	
I/O ポート番号		可

※I/O ポートは ①初期化を IPR(ユーザ)が行うこと。

②ビット単位のアクセス可能なポート。

※uEntry時は直ちに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する。

6. 14 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 80 MHz で測定

項目	内容	ユーザ設定
APL SUM 域 4Kbyte	21.2ms	不可
APL SUM 域 256byte	12.6ms	不可

7. r Entryモード仕様

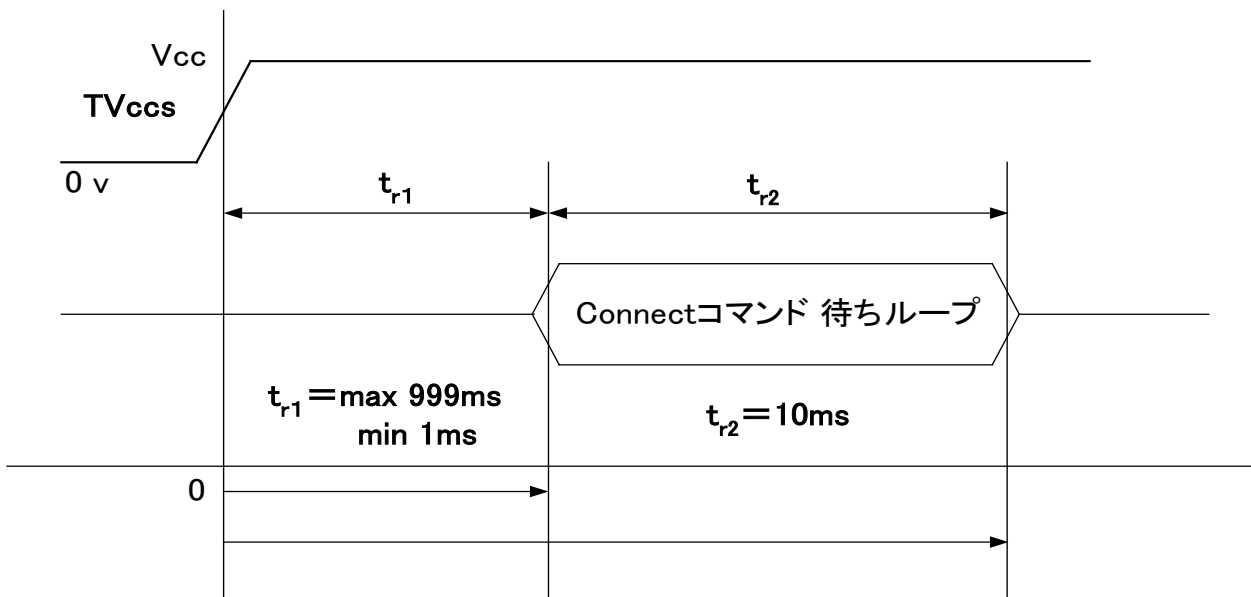
7.1 概要

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、リプログラム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

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

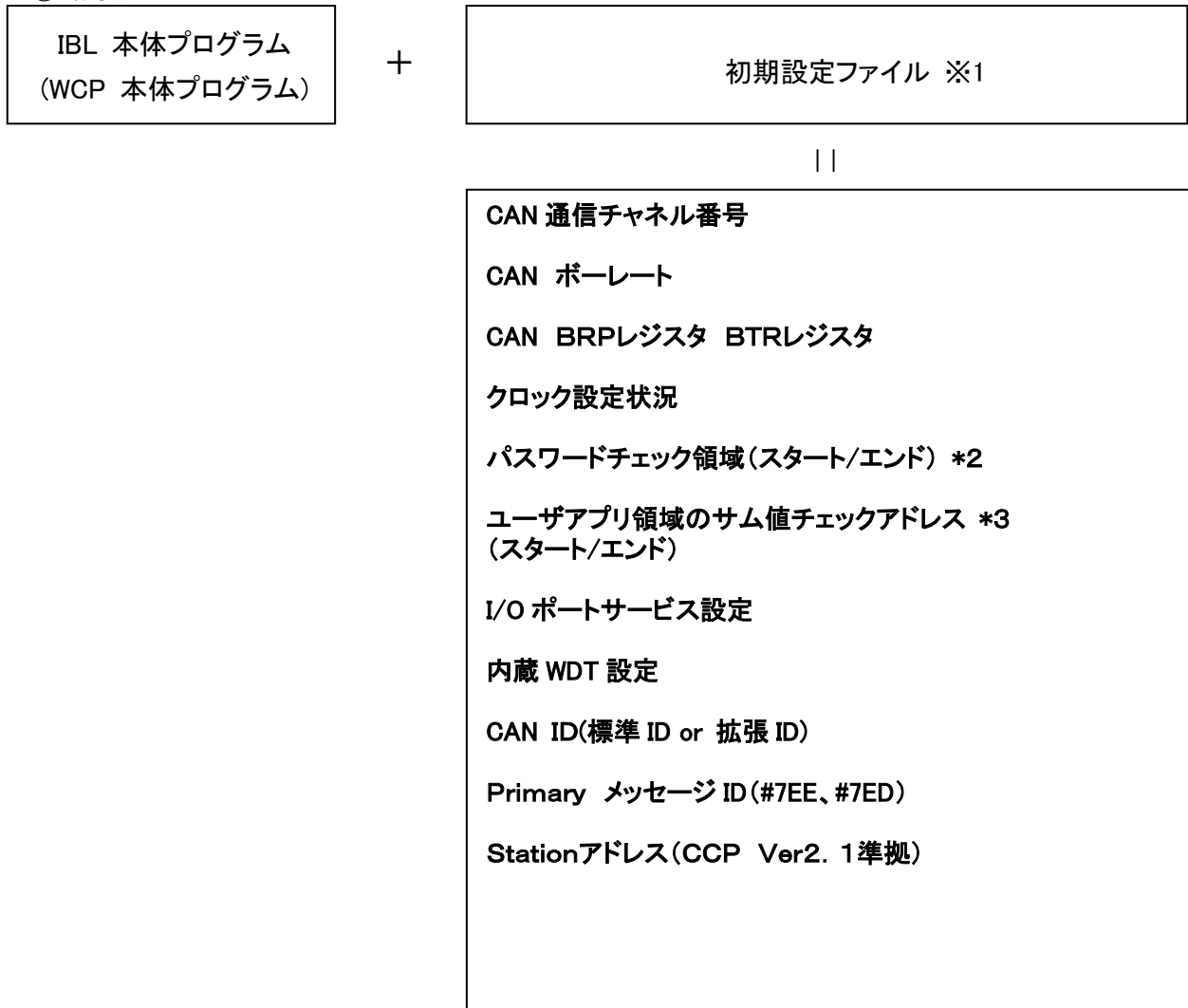
3. ターゲット側の t_{r1} 時間分タイミングを合わせるために、ライター側の Connect コマンド発行タイミングを air Connect の Specific Parameter のアドレス #14C, #14D の 2byte で設定してください。

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	#00FC3FF0h~#00FC3FFFh	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

デバイスファンクション実行に伴い、MCU上のRAM内容は破棄されます。
(RAMのECC対応のため、IPRのスタートアップ時に全領域をクリアします)

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	一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。 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	なし
戻り値	なし

- ・ init_pit(void) 関数

内容	I/O ポートサービス用のタイマの初期化を行います。
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(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) 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
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。 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	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。 無限ループに入ることで WDT によるリセットを行ないます。
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ー側に送信します。
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。 受信データの書き込みを行います。
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。 flash_erase_block 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

- Command_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。 受信したチェック範囲に対し 1byte 単位で読出し 0xFF であるかチェックします。
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ー側に返します。
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。 ライター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	未使用	I/O サービスに使用	可
MCU 内蔵タイマ	PIT0	サイクルアクセス用	不可
CAN 通信ポート	TX = 27 ピン (PC[3]) RX = 26 ピン (PC[2]) ※ピン番号は 64 ピンパッケージの場合	CAN 通信用の端子です	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
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ビットレートレジスタ	CAN_CR_DATA	ビットタイミングレジスタ値	0x03490004	可
コアクロック	CLK_SYS	MHz×10で指定 例)10MHZ = 100	800 (80MHz)	可
CAN クロック	CLK_CAN		200 (20MHz)	可
PIT クロック	CLK_PIT		400 (40MHz)	可
SWT クロック	CLK_SWT		160 (16MHz)	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	0x00FC3FF0
パスワードチェック領域 (エンド)	PASS_END	32bit アドレス指定	0x00FC3FFF	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	0x00FDFF00	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	0x00FDFFFF	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 ポート番号	IOS_PORT	ポート番号	7	可
I/O ポートサービス 周期	IOS_PERIOD	ms 単位(1ms~65535ms)	4	可
内蔵 WDT サービス 対応フラグ	IOS_WDT_ON	0 or 1 0:WDT サービス無し 1:WDT サービス有り	0	可
内蔵 WDT サービス 周期	IOS_WDT_PERIOD	ms 単位(1ms~65535ms)	4	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
CAN チャネル番号	CAN_CHAN	0 チャネル	0	可
CANID フォーマット	CAN_ID	Standard=0 or Extended=1	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	FUNC 81	“STANDARD”/“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	“STANDARD”/“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	0x00FDFF00	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	0x00FDFFFF	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)	- 要調整	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	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	#03 (3 回)	可
NI CAN 終端抵抗	Specific Parameter #CF	AZ990 等で Specific Parameter #CF から 1BYTE 指定する #08: 120Ω #18: 60Ω #28: 無し	#08 (120Ω)	可

※1 初期設定ファイルと同期をとる項目です。