



MICROCHIP

PICKit™ 3 プログラマ / デバッガ ユーザガイド

注意：この日本語版文書は参考資料としてご利用ください。最新情報は必ずオリジナルの英語版をご参照します。

マイクロチップ社製デバイスのコード保護機能に関して以下の点にご注意ください。

- マイクロチップ社製品は、該当するマイクロチップ社データシートに記載の仕様を満たしています。
- マイクロチップ社では、通常の条件ならびに仕様に従って使用した場合、マイクロチップ社製品のセキュリティ レベルは、現在市場に流通している同種製品の中でも最も高度であると考えています。
- しかし、コード保護機能を解除するための不正かつ違法な方法が存在する事もまた事実です。弊社の理解では、こうした手法はマイクロチップ社データシートにある動作仕様書以外の方法でマイクロチップ社製品を使用する事になります。このような行為は知的所有権の侵害に該当する可能性が非常に高いと言えます。
- マイクロチップ社は、コードの保全性に懸念を抱いているお客様と連携し、対応策に取り組んでいきます。
- マイクロチップ社を含む全ての半導体メーカーで、自社のコードのセキュリティを完全に保証できる企業はありません。コード保護機能とは、マイクロチップ社が製品を「解読不能」として保証するものではありません。

コード保護機能は常に進歩しています。マイクロチップ社では、常に製品のコード保護機能の改善に取り組んでいます。マイクロチップ社のコード保護機能の侵害は、デジタル ミレニアム著作権法に違反します。そのような行為によってソフトウェアまたはその他の著作物に不正なアクセスを受けた場合、デジタル ミレニアム著作権法の定めるところにより損害賠償訴訟を起こす権利があります。

本書に記載されているデバイス アプリケーション等に関する情報は、ユーザの便宜のためにのみ提供されているものであり、更新によって無効とされる事があります。お客様のアプリケーションが仕様を満たす事を保証する責任は、お客様にあります。マイクロチップ社は、明示的、暗黙的、書面、口頭、法定のいずれであるかを問わず、本書に記載されている情報に関して、状態、品質、性能、商品性、特定目的への適合性をはじめとする、いかなる類の表明も保証も行いません。マイクロチップ社は、本書の情報およびその使用に起因する一切の責任を否認します。マイクロチップ社の明示的な書面による承認なしに、生命維持装置あるいは生命安全用途にマイクロチップ社の製品を使用する事は全て購入者のリスクとし、また購入者はこれによって発生したあらゆる損害、クレーム、訴訟、費用に関して、マイクロチップ社は擁護され、免責され、損害を受けない事に同意するものとします。暗黙的あるいは明示的を問わず、マイクロチップ社が知的財産権を保有しているライセンスは一切譲渡されません。

商標

マイクロチップ社の名称とロゴ、Microchip ロゴ、dsPIC、KEELOQ、KEELOQ ロゴ、MPLAB、PIC、PICmicro、PICSTART、PIC³² ロゴ、rfPIC、UNI/O は、米国およびその他の国におけるマイクロチップ・テクノロジー社の登録商標です。

FilterLab、Hampshire、HI-TECH C、Linear Active Thermistor、MXDEV、MXLAB、SEEVAL、Embedded Control Solutions Company は、米国におけるマイクロチップ・テクノロジー社の登録商標です。

Analog-for-the-Digital Age、Application Maestro、chipKIT、chipKIT ロゴ、CodeGuard、dsPICDEM、dsPICDEM.net、dsPICworks、dsSPEAK、ECAN、ECONOMONITOR、FanSense、HI-TIDE、In-Circuit Serial Programming、ICSP、Mindi、MiWi、MPASM、MPLAB Certified ロゴ、MPLIB、MPLINK、mTouch、Omniscient Code Generation、PICC、PICC-18、PICDEM、PICDEM.net、PICkit、REAL ICE、rFLAB、Select Mode、Total Endurance、TSHARC、UniWinDriver、WiperLock、ZENA は、米国およびその他の国におけるマイクロチップ・テクノロジー社の商標です。

SQTP は、米国におけるマイクロチップ・テクノロジー社のサービスマークです。

その他、本書に記載されている商標は各社に帰属します。

© 2012, Microchip Technology Incorporated, All Rights Reserved.

ISBN: 978-1-61341-558-0

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949 ==

マイクロチップ社では、Chandler および Tempe (アリゾナ州)、Gresham (オレゴン州) の本部、設計部およびウェハ製造工場そしてカリフォルニア州とインドのデザインセンターが ISO/TS-16949:2009 認証を取得しています。マイクロチップ社の品質システムプロセスおよび手順は、PIC[®] MCU および dsPIC[®] DSC、KEELOQ[®] コードホッピングデバイス、シリアルEEPROM、マイクロペリフェラル、不揮発性メモリ、アナログ製品に採用されています。さらに、開発システムの設計と製造に関するマイクロチップ社の品質システムは ISO 9001:2000 認証を取得しています。

目次

序章	5
Chapter 1. 概要	
1.1 はじめに	13
1.2 PICkit 3 プログラマ / デバッガについて	13
1.3 PICkit 3 プログラマ / デバッガの利点	15
1.4 PICkit 3 プログラマ / デバッガの構成要素	16
Chapter 2. 動作原理	
2.1 はじめに	17
2.2 PICkit 3 と PICkit 2 の違い	17
2.3 デバッガとターゲットの通信	17
2.4 通信の接続	19
2.5 デバッグ	21
2.6 デバッグのための要件	22
2.7 プログラミング	24
2.8 デバッガが使うリソース	24
Chapter 3. インストール	
3.1 はじめに	25
3.2 ソフトウェアのインストール	25
3.3 ターゲットの接続	25
3.4 ターゲットボードの設定	26
3.5 MPLAB® IDE の設定	27
Chapter 4. 全般的な設定	
4.1 はじめに	29
4.2 MPLAB IDE ソフトウェアの起動	29
4.3 プロジェクトの作成	29
4.4 プロジェクトの表示	30
4.5 プロジェクトのビルド	30
4.6 コンフィグレーション ビットの設定	30
4.7 デバッガまたはプログラマとしての設定	30
4.8 デバッガ / プログラマの制約	31
Chapter 5. PICkit 3 Debug Express	
5.1 はじめに	33
5.2 PICkit 3 Debug Express キットの内容	33
5.3 ハードウェアとソフトウェアのインストール	33

PICkit™ 3 ユーザーガイド

Chapter 6. PICkit 3 Programmer-To-Go

6.1	はじめに.....	35
6.2	PICkit 3 Programmer-To-Go の USB 電源.....	35
6.3	PICkit 3 Programmer-To-Go 対応デバイス.....	36
6.4	PICkit 3 で Programmer-To-Go 機能を使うための設定.....	37
6.5	PICkit 3 の Programmer-To-Go 機能の使い方.....	38
6.6	Programmer-To-Go モードの終了.....	40

Chapter 7. トラブルシュート

7.1	はじめに.....	43
7.2	最初に確認する 5 つの項目.....	43
7.3	デバッグに失敗する主な理由.....	43
7.4	その他の確認事項.....	44

Chapter 8. よく寄せられる質問 (FAQ)

8.1	はじめに.....	45
8.2	動作に関する FAQ.....	45
8.3	不具合に関する FAQ.....	46

Chapter 9. エラーメッセージ

9.1	はじめに.....	49
9.2	特定の状況で表示されるエラーメッセージ.....	49
9.3	一般的な対処方法.....	53

Chapter 10. デバッグ機能一覧

10.1	はじめに.....	57
10.2	デバッグに関する機能.....	57
10.3	デバッグに関するダイアログ/ウィンドウ.....	59
10.4	プログラミングに関する機能.....	62
10.5	[Settings] ダイアログ.....	63

補遺 A. ハードウェア仕様

A.1	はじめに.....	69
A.2	ハイライト.....	69
A.3	適合宣言書.....	69
A.4	USB ポート / 電源.....	70
A.5	PICkit 3 プログラマ / デバッグ.....	70
A.6	標準通信ハードウェア.....	71
A.7	ターゲットボードに関する注意事項.....	73

補遺 B. PICkit 3 の回路図

用語集.....	77
----------	----

索引.....	97
---------	----

各国の営業所とサービス.....	100
------------------	-----

序章

お客様へのご注意

全ての文書の内容は時間と共に古くなります。本書も例外ではありません。マイクロチップ社のツールとマニュアルはお客様のニーズを満たすために常に改良を重ねており、実際のダイアログやツールの内容が本書に記載されているものと異なる場合があります。

最新の文書は弊社ウェブサイト (www.microchip.com) をご覧ください。

文書は「DS」番号で識別します。この識別番号は、各ページのフッタ部分、ページ番号の前に記載しています。DS 番号の表記規則は「DSXXXXA」で、「XXXX」が文書番号、「A」が文書のリビジョンレベルを表しています。

開発ツールについての最新情報は、MPLAB® IDE のオンラインヘルプをご覧ください。[Help] メニューを選択して、次に [Topics] を選択すると、利用できるオンラインヘルプファイルのリストが表示されます。

はじめに

ここでは、PICKit™ 3 プログラマ / デバッガを使用する前に知っておくと便利な一般的な情報について説明します。この章の内容は以下の通りです。

- 本書の構成
- 本書の表記規則
- 保証登録
- 推奨参考資料
- マイクロチップ社のウェブサイト
- 開発システムのお客様向け通知サービス
- カスタマ サポート
- 改訂履歴

本書の構成

本書では、PICkit 3 を開発ツールとして使用し、ターゲットボードのファームウェアのエミュレーションとデバッグを行う方法を説明します。本書の構成は以下の通りです。

- **Chapter 1. 概要** – PICkit 3 プログラマ / デバッグの概要を説明します。
- **Chapter 2. 動作原理** – PICkit 3 プログラマ / デバッグの動作原理を簡単に説明します。
- **Chapter 3. インストール** – PICkit 3 プログラマ / デバッグのインストール方法を説明します。
- **Chapter 4. 全般的な設定** – PICkit 3 プログラマ / デバッグを使ってデバイスをプログラムする手順について説明します。
- **Chapter 5. PICkit 3 Debug Express** – PICkit™ 3 Debug Express の基本的な使用方法を説明します。
- **Chapter 6. PICkit 3 Programmer-To-Go** – PC に接続せず PICkit 3 を使ってデバイスをプログラムする方法を説明します。
- **Chapter 7. トラブルシュート** – トラブルシュートの手順を説明します。最初に確認する項目とデバッグに失敗する一般的な原因を示します。
- **Chapter 8. よく寄せられる質問 (FAQ)** – 一般的な問題の解決方法を説明します。
- **Chapter 9. エラーメッセージ** – エラーメッセージとそれらに対する一般的な対処方法を説明します。
- **Chapter 10. デバッグ機能一覧** – 利用可能なデバッグ機能をまとめています。
- **補遺 A. ハードウェア仕様** – PICkit 3 のハードウェア仕様と電氣的仕様を詳しく説明します。
- **補遺 B. PICkit 3 の回路図** – PICkit 3 プログラマ / デバッグのハードウェア回路図を示します。

本書の表記規則

本書では以下の表記規則を使用します。

本書の表記規則

説明	意味	例
ゴシックフォント:		
斜体	参考資料	MPLAB [®] IDE User's Guide
	強調文字	... は 唯一 のコンパイラです ...
角カッコ: []	ウィンドウ	[Output] ウィンドウ
	ダイアログ	[Settings] ダイアログ
	メニューの選択肢	[Enable Programmer] を選択
かぎカッコ: 「」	ウィンドウまたはダイアログのフィールド名	「Save project before build」
右山カッコ (>) を使用し、角カッコで囲まれた下線付き斜体テキスト	メニューパス	[File]>[Save]
太字で角カッコに囲まれたテキスト	ダイアログのボタン	[OK] をクリックする。
	タブ	[Power] タブをクリックする。
山カッコで囲まれたテキスト: <>	キーボードのキー	<Enter>、<F1> を押す。
Courier New フォント:		
プレーン (斜体、太字でない)	サンプル ソースコード	#define START
	ファイル名	autoexec.bat
	ファイルパス	c:\mcc18\h
	キーワード	_asm, _endasm, static
	コマンドライン オプション	-Opa+, -Opa-
	ビット値	0, 1
	定数	0xFF, 'A'
斜体	変数の引数	<i>file.o</i> (<i>file</i> は有効なファイル名)
角カッコ: []	オプションの引数	mpasmwin [オプション] file [オプション]
中カッコとパイプ文字: {}	いずれかの引数を選択する場合 (OR 選択)	errorlevel {0 1}
省略記号: ...	繰り返されるテキスト	var_name [, var_name...]
	ユーザが定義するコード	void main (void) { ... }

保証登録

同封の保証登録カードにご記入いただき、お早めにご郵送ください。この保証登録カードを送付されたお客様には、製品のアップデート情報をお送りします。ソフトウェアのマイナー リリースは弊社ウェブサイトを提供しております。

推奨参考資料

本書は、PICkit 3 の使用方法を説明したものです。他にも便利な文書がありますので、以下に一覧でご紹介します。以下に記載したマイクロチップ社の文書を参考資料としてお読みになる事をお勧めします。

『In-Circuit Debugger Design Advisory』 (DS51764)

まず、この文書を最初にお読みください。この文書には、PICkit 3 を使用する際に注意すべき重要な情報が記載されています。

『44-Pin Demo Board User's Guide』 (DS41296)

44 ピンデモボードを開発ツールとして使ってターゲットボードのファームウェアのエミュレーションとデバッグを行う方法については、この文書を参照してください。

『Low Pin Count Demo Board User's Guide』 (DS51556)

マイクロチップ社の少ピンデバイス (8/14/20 ピン) を使う方法については、この文書を参照してください。この文書には一連のチュートリアルも含まれます。

『MPLAB® IDE User's Guide/Help』 (DS51519)

MPLAB IDE (Integrated Development Environment) ソフトウェアのインストールと機能の詳細は、この文書を参照してください。オンラインヘルプ版もあります。

『In-Circuit Serial Programmer™ (ICSP™) Guide』 (DS30277)

この文書では、ICSP プログラミングを成功させるための設計ガイドラインを紹介しています。ハードウェア設計向けアプリケーションノートと ICSP プログラミング仕様も含まれます。

『MPASM™ Assembler, MPLINK™ Object Linker, MPLIB™ Object Librarian User's Guide』 (DS33014)

マイクロチップ社の PIC® MCU アセンブラ (MPASM アセンブラ)、リンカ (MPLINK リンカ)、ライブラリアン (MPLIB ライブラリアン) の使用方法を説明しています。

PICkit™ 3 Debug Express の Readme

PICkit 3 Debug Express の使用に関する最新情報は、MPLAB IDE のインストール先ディレクトリの Readmes サブディレクトリにある「Readme for PICkit 3.htm」ファイル (HTML ファイル) を参照してください。Readme ファイルには、ユーザガイドに反映されていない最新情報と既知の問題が記載されています。

『PICkit™ 3 Debug Express C18 Lessons』

PICkit 3 Debug Express と PIC18 MCU 用の MPLAB C コンパイラの使用法を手順を追って説明したチュートリアルです。MPLAB IDE の CD-ROM に収録されていますが、マイクロチップ社ウェブサイトからも入手できます。

Readme ファイル

その他のツールの使用についての最新情報は、MPLAB IDE のインストール先ディレクトリの Readmes サブディレクトリにある各ツールの Readme ファイルを参照してください。Readme ファイルには、ユーザガイドに反映されていない最新情報と既知の問題が記載されています。

マイクロチップ社のウェブサイト

マイクロチップ社は、弊社が運営する WWW サイト (www.microchip.com) を通してオンラインサポートを提供しています。このウェブサイトを活用する事で、ファイルと情報を簡単に入手できます。お好みのインターネットブラウザを使って、下記の内容をご覧ください。

- **製品サポート** – データシートとエラッタ、アプリケーションノートとサンプルプログラム、設計リソース、ユーザガイドとハードウェアサポート文書、最新のソフトウェアと過去のソフトウェア
- **一般的な技術サポート** – よく寄せられる質問 (FAQ)、技術サポートリクエスト、オンラインディスカッショングループ、マイクロチップ社コンサルタントプログラムメンバーの一覧
- **マイクロチップ社の事業** – 製品セレクトと注文のガイド、マイクロチップ社の最新プレスリリース、セミナーとイベントの一覧、マイクロチップ社の各営業所、販売代理店、工場の一覧

開発システムのお客様向け通知サービス

マイクロチップ社のお客様向け通知サービスにて、常にお客様にマイクロチップ社製品の最新情報を提供いたします。ご興味のある製品ファミリまたは開発ツールに関する変更、更新、エラッタ情報をいち早くメールにてお知らせします。

マイクロチップ社のウェブサイト www.microchip.com にアクセスし、[Customer Change Notification] からご登録ください。

開発システム製品のグループカテゴリは、以下の通りです。

- **コンパイラ** – マイクロチップ社の C コンパイラ、アセンブラ、リンカ、その他の言語ツールの最新情報です。これには、MPLAB C コンパイラ全製品、MPLAB アセンブラ全製品 (MPASM アセンブラを含む)、MPLAB リンカ全製品 (MPLINK オブジェクトリンカを含む)、MPLAB ライブラリアン全製品 (MPLIB オブジェクトライブラリアンを含む) が含まれます。
- **エミュレータ** – マイクロチップ社のインサーキットエミュレータの最新情報です。これには、MPLAB REAL ICE、MPLAB ICE 2000 インサーキットエミュレータが含まれます。
- **インサーキットデバッガ** – マイクロチップ社のインサーキットデバッガに関する最新情報です。これには、MPLAB ICD 3 インサーキットデバッガと PICKit™ 2/3 Debug Express が含まれます。
- **MPLAB® IDE** – 開発システムツール向け Windows® 統合開発環境であるマイクロチップ社の MPLAB IDE の最新情報です。これには、MPLAB IDE、MPLAB IDE プロジェクトマネージャ、MPLAB エディタ、MPLAB SIM シミュレータ、一般的な編集とデバッグの機能が含まれます。
- **プログラマ** – マイクロチップ社のプログラマの最新情報です。これには、MPLAB REAL ICE インサーキットエミュレータ、MPLAB ICD 3 インサーキットデバッガ、MPLAB PM3 デバイスプログラマ等の量産プログラマが含まれます。また、PICSTART® Plus、PICKit 1/2/3 等、非量産向けの開発プログラマも含まれます。

カスタマ サポート

マイクロチップ社製品のお客様は、下記のチャンネルからサポートをご利用頂けます。

- 販売代理店または販売担当者
- 各地の営業所
- フィールド アプリケーション エンジニア (FAE)
- 技術サポート

サポートは販売代理店、販売担当者、フィールド アプリケーション エンジニア (FAE) までお問い合わせください。各地の営業所もご利用頂けます。各営業所と所在地の一覧は、本書の巻末に記載しています。最新の営業所一覧は、弊社ウェブサイトでご確認ください。

技術サポートは下記のウェブサイトからもご利用頂けます：
<http://support.microchip.com>

本書の内容に誤りやその他ご意見がありましたら、docerrors@microchip.com までメールでお寄せください。

改訂履歴

リビジョン A (2009 年 1 月)

本書の初版

リビジョン B (2010 年 7 月)

セクション 1.5 「デバイスと機能のサポート」を削除。この内容は、MPLAB IDE の PICkit 3 オンラインヘルプにてご確認ください。

セクション 2.4.4 「デバッガから電源を供給する場合」の 3 番目の段落を差し換え。

セクション 2.7 「プログラミング」に、デバイスのプログラム方法を 3 つ追加。

第 6 章 「PICkit 3 Programmer-To-Go」を追加。

表 10-6 「[Program Memory] タブのオプション」に 「Use high voltage on MCLR」の項目を追加。

セクション 10.5.8 「[Settings] ダイアログの [Power] タブ」に 「Power target circuit from PICkit 3」オプションの説明文を追加。

セクション 10.5.10 「[Setting] ダイアログの [Programmer-to-go] タブ」を追加。

第 11 章 「ハードウェア仕様」を 「補遺 A ハードウェア仕様」に変更。

第 1 部 – 入門編

Chapter 1. 概要	13
Chapter 2. 動作原理	17
Chapter 3. インストール	25
Chapter 4. 全般的な設定	29
Chapter 5. PICKit 3 Debug Express	33

PICKit™ 3 ユーザガイド

NOTES:

Chapter 1. 概要

1.1 はじめに

ここでは、PICKit 3 プログラマ / デバッガ システムの概要を説明します。

- PICKit 3 プログラマ / デバッガについて
- PICKit 3 プログラマ / デバッガの利点
- PICKit 3 プログラマ / デバッガの構成要素

1.2 PICKit 3 プログラマ / デバッガについて

PICKit 3 プログラマ / デバッガ (図 1-1 参照) はシンプルかつ低コストなインサーキット デバッガで、Windows® で動作する MPLAB IDE (v8.20 以上) ソフトウェアを使って操作します。PICKit 3 プログラマ / デバッガは、開発エンジニアにとって欠かせないツールの 1 つです。ソフトウェアの開発からハードウェアの統合まで、幅広い用途に対応します。

PICKit 3 プログラマ / デバッガは、インサーキット シリアル プログラミング™ (ICSP™) と拡張 ICSP の 2 線式シリアル インターフェイスを使って、マイクロチップ社の PIC®MCU と dsPIC® DSC 向けのハードウェア / ソフトウェアの開発を行えるデバッグシステムです。PICKit 3 プログラマ / デバッグシステムは、デバッガとしてだけでなく開発プログラマとしても使用できます。量産プログラマとしての使用は想定していません。

このデバッグシステムはデバッグ専用のチップではなくエミュレーション回路を内蔵したデバイスを使うため、実際のデバイスと同じようにコードを実行します。ターゲット デバイスで利用できる機能は全てインタラクティブに利用でき、MPLAB IDE インターフェイスを利用して設定や変更が行えます。

PICKit 3 デバッガは、デバッグ機能を内蔵した組み込みプロセッサのエミュレーション用に開発されました。PICKit 3 の主な特長は以下の通りです。

- Windows 標準ドライバでフルスピード USB をサポート
- リアルタイム実行
- プロセッサの最大動作速度でデバッグが行える
- 過電圧 / 短絡監視機能を内蔵
- 低電圧 ~ 5 V (レンジ: 1.8 ~ 5 V)
- LED インジケータ (POWER、ACTIVE、STATUS)
- マイクロコントローラのプログラムメモリとデータメモリの読み書き
- 全種類のメモリ (EEPROM、ID、コンフィグレーション、プログラム) の消去とベリファイ
- ブレークポイントでの周辺モジュールのフリーズ

図 1-1: PICKit™ 3 MCU プログラマ / デバッガ



1.2.1 ストラップホール

プログラマ本体にストラップを取り付けて使用できます。

1.2.2 USB ポート

USB mini-B コネクタタイプの USB ポートです。付属の USB ケーブルを使って PICKit 3 を PC に接続します。

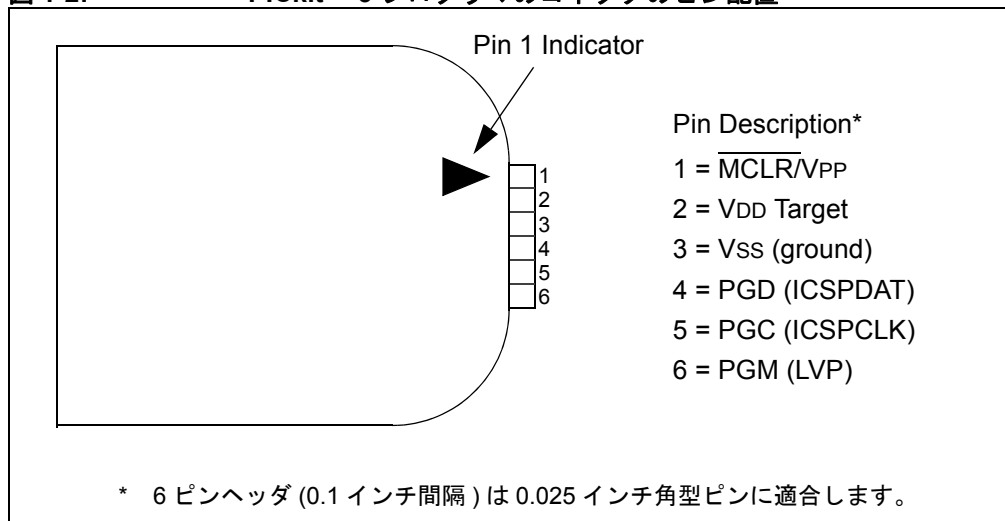
1.2.3 1 ピンマーカ

コネクタの向きを合わせる目印として、1 ピンの位置を示しています。

1.2.4 プログラミング コネクタ

ターゲット デバイスと接続するための 6 ピンヘッダ (0.100 インチ間隔) です。ピン配置は図 1-2 を参照してください。

図 1-2: PICkit™ 3 プログラマのコネクタのピン配置



Note: シリアル EEPROM デバイスのプログラミング時は、プログラミングコネクタのピン機能が異なります。ピン配置の詳細は、MPLAB IDE に付属している PICkit 3 の ReadMe ファイル ([Help]>[Readme]) を参照してください。

1.2.5 インジケータ LED

インジケータ LED は、PICkit 3 のステータスを表示します。

1. **POWER** (緑) – PICkit 3 に USB ポートから電源が供給されています。
2. **ACTIVE** (青) – PICkit 3 が PC の USB ポートに接続されて通信リンクがアクティブです。
3. **STATUS:**
 - ビジー (黄) – プログラミング等の動作を実行中です。
 - エラー (赤) – エラーが発生しました。

1.2.6 プッシュボタン

このプッシュボタンは、Programmer-To-Go 機能で使います (Chapter 6. 「PICkit 3 Programmer-To-Go」参照)。

1.3 PICkit 3 プログラマ / デバッガの利点

PICkit 3 プログラマ / デバッガには以下の利点があります。

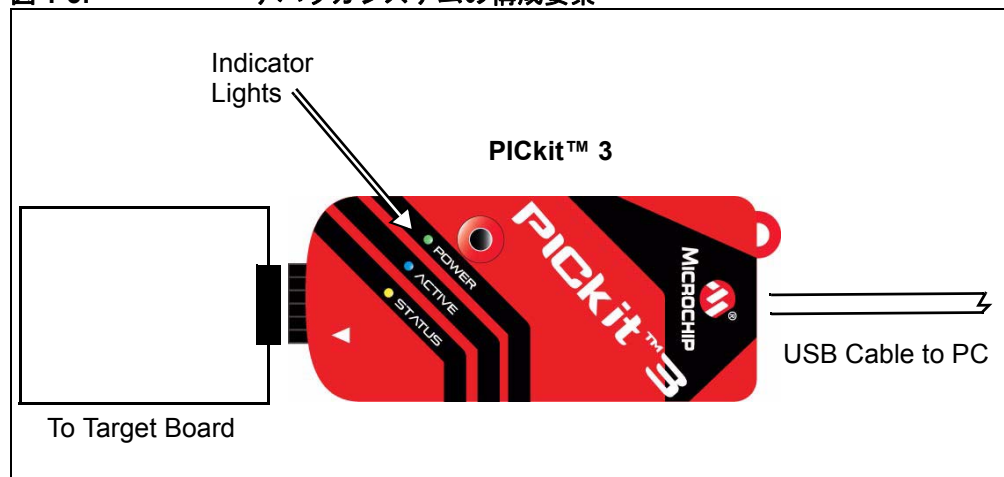
- 実際のハードウェアでアプリケーションをリアルタイム デバッグできる
- ハードウェア ブレークポイント使ってデバッグできる
- 内部イベントに基づいたブレークポイントが設定できる
- 内部ファイルレジスタを監視できる
- フルスピードでエミュレートできる
- デバイスがプログラムできる

1.4 PICKit 3 プログラマ / デバッガの構成要素

PICKit 3 プログラマ / デバッガシステムには以下のものが含まれます。

1. PICKit 3 (POWER、ACTIVITY、STATUS インジケータ付き)
2. USB ケーブル (デバッガと PC 間の通信と、デバッガへの電源供給用)
3. MPLAB IDE ソフトウェアと文書を収録した CD-ROM

図 1-3: デバッガシステムの構成要素



この他、以下のオプションもご購入頂けます。

- 以下を含む PICKit 3 Debug Express Kit:
 - PIC18F45K20 MCU を実装した 44 ピン デモボード
 - PIC18 MCU 用 MPLAB C コンパイラ (無償版)
 - 分かりやすいレッスンとチュートリアル
 - その他のソフトウェア ユーティリティ、サンプルコード (ソースコード付き)、文書一式
- 変換ソケット
- ICD ヘッダ
- MPLAB IDE プロセッサ拡張キット

Chapter 2. 動作原理

2.1 はじめに

ここでは、PICKit 3 プログラマ / デバッガシステムの動作のしくみについて簡単に説明します。ここでの目的は、PICKit 3 プログラマ / デバッガシステムを使ってエミュレーションとプログラミングを行えるよう、ターゲットボードの設計に必要な情報を提供する事にあります。また、万一問題が発生してもすぐに解決できるように、インサーキット エミュレーションおよびプログラミングの基本的な動作原理についても説明します。

- PICKit 3 と PICKit 2 の違い
- デバッガとターゲットの通信
- 通信の接続
- デバッグ
- デバッグのための要件
- プログラミング
- デバッガが使うリソース

2.2 PICKit 3 と PICKit 2 の違い

PICKit 3 プログラマ / デバッガシステムは、PICKit 2 インサーキット デバッガシステムと多くの点で似ています。これら 2 つのデバッガの似ている点は以下の通りです。

- PC から USB ケーブルで電源を供給する
- 供給する電源電圧値を設定できる

PICKit 2 にはなく、PICKit 3 で追加された機能は以下の通りです。

- 拡張 EE プログラム イメージ空間 (512 KB)
- 正確な参照電圧
- 電圧レンジが広がった (1.8 ~ 5 V VDD、1.8 ~ 14 V VPP)

2.3 デバッガとターゲットの通信

ここからは、デバッガシステムの構成について説明します。

CAUTION

PICKit 3 からターゲットに電源を供給している間はケーブル / ピンを抜き差ししないでください。

標準 ICSP による通信

PICKit 3 デバッガシステムは、標準 ICSP 通信によるプログラミングとデバッグが可能です。使用する 6 ピン接続は、PICKit 2 プログラマ / デバッガシステムで使っていたものと同じです。

PICKit™ 3 ユーザガイド

モジュラケーブルは、(1) ターゲット デバイスがターゲットボードに直接実装されている場合はターゲットの対応ソケットに挿入し (図 2-1)、(2) ターゲットボードに標準のアダプタ / ヘッダボード コンボ (プロセッサ パックとして提供) を接続している場合はアダプタ / ヘッダボードのソケットに挿入します (図 2-2)。

Note: 旧式のヘッダボードは 8 ピンコネクタではなく 6 ピンの RJ-11 コネクタを使っているため、AC164110 ICSP アダプタを使ってデバッグと接続します。

標準通信の詳細は、補遺 A.「ハードウェア仕様」を参照してください。

図 2-1: 標準のデバッグシステム — ICE 回路内蔵デバイスの場合

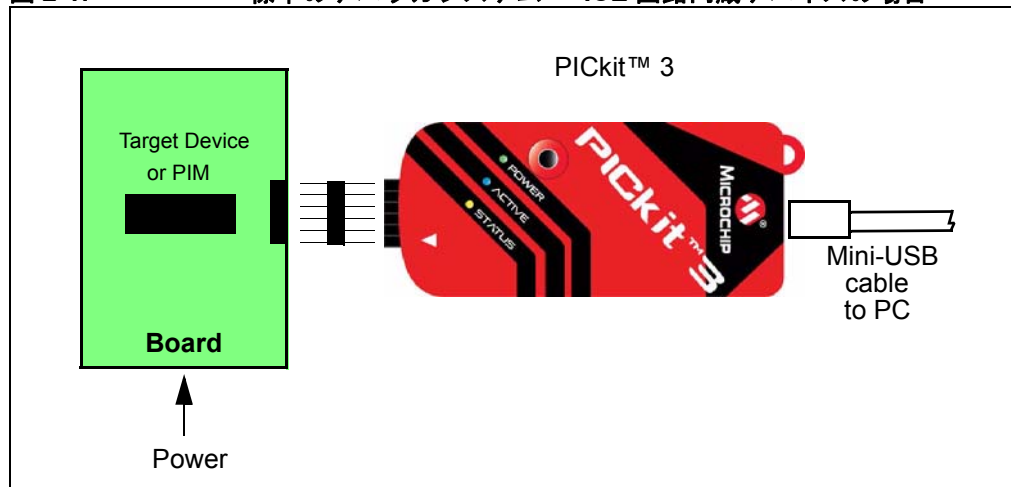
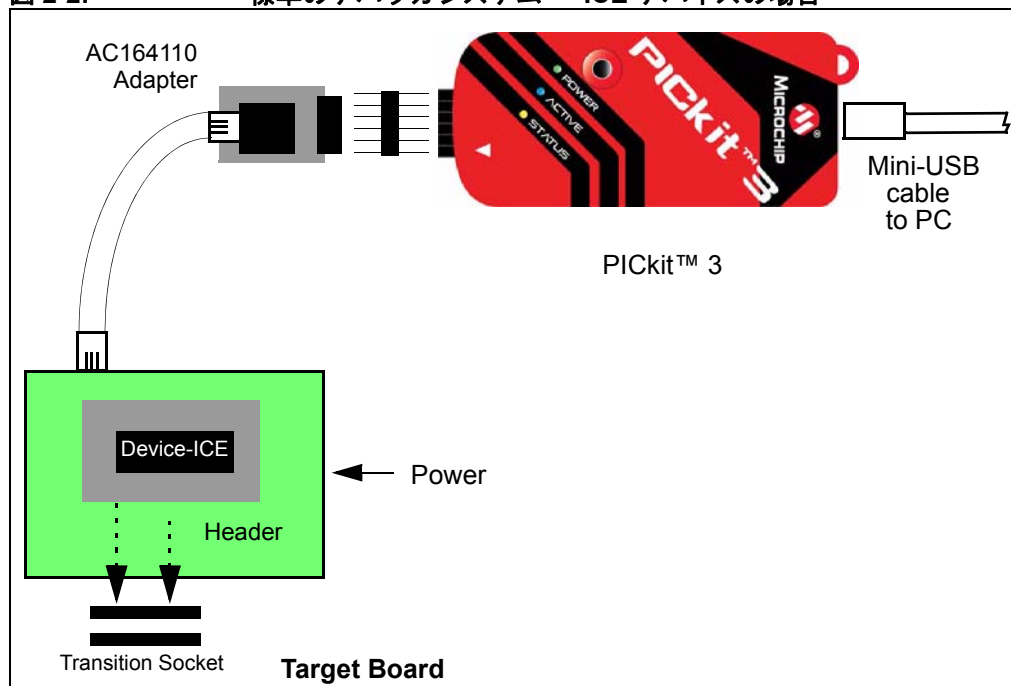


図 2-2: 標準のデバッグシステム — ICE デバイスの場合



2.4 通信の接続

2.4.1 ターゲットとの通信

2.4.1.1 シングル インライン コネクタを使う場合

6 ピンのインライン コネクタを使って PICKit 3 プログラマ / デバッガとターゲットボードのコネクタを接続します (図 2-1 参照)。表 2-1 と A.6「標準通信ハードウェア」も参照してください。

表 2-1: ターゲット側コネクタのピン配置

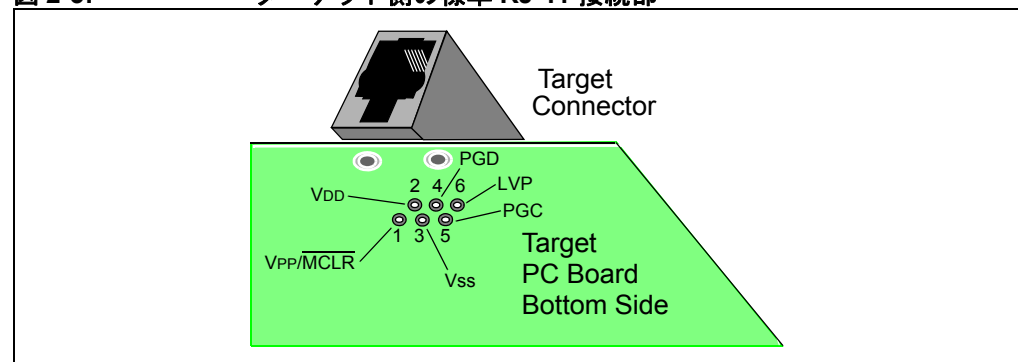
コネクタのピン番号	マイクロコントローラのピン
1	MCLR/VPP
2	VDD
3	グラウンド
4	PGD (ICSPDAT)
5	PGC (ICSPCLK)
6	PGM (LVP)

2.4.1.2 アダプタを使う場合

PICKit 3 プログラマ / デバッガとターゲット デバイスの間に AC164110 アダプタを使って、6 芯のモジュラ インターフェイス ケーブルで接続します。図 2-3 にターゲットボードのはんだ面から見たコネクタのピン番号を示します。

Note: デバッガとターゲットをケーブルで接続すると、反対のピン番号どうしが結線されます。つまり、ケーブルの片方の 1 ピンは、もう片方の 6 ピンに接続されます。(A.6.2.3「モジュラケーブルの仕様」参照)。

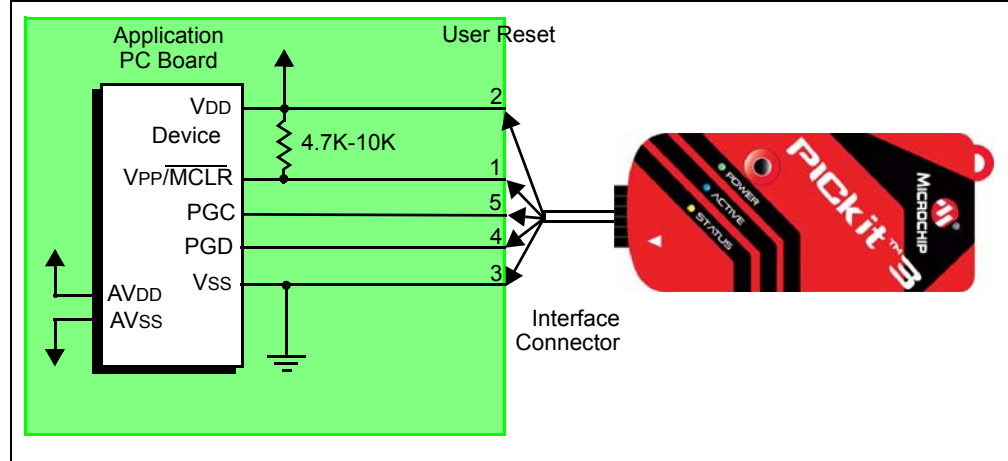
図 2-3: ターゲット側の標準 RJ-11 接続部



2.4.2 ターゲットとの接続回路

図 2-4 に PICKit 3 プログラマ / デバッガとターゲットボードを接続する際の配線を示します。この図には、コネクタからターゲットのプリント基板上のデバイスへの配線も示されています。VPP/MCLR ラインから VDD ヘブルアップ抵抗 (通常、約 10 kΩ) を接続しておく事を推奨します。こうすると、このラインを Low にするとデバイスをリセットできます。

図 2-4: ターゲット回路との標準接続



2.4.3 ターゲットの自己給電

以下の説明では、1 ピン (VPP/MCLR)、5 ピン (PGC)、4 ピン (PGD) の 3 つのラインのみがアクティブで、基本的なデバッグの動作に関係します。参考までに図 2-4 には、2 ピン (VDD) と 3 ピン (VSS) も示しています。PICKit 3 では、ターゲットデバイスの電源をデバッガから供給する構成と、ターゲット側に外付けの電源を使用する構成の 2 種類が可能です。

ターゲットアプリケーションの外付け電源を使用する方法を推奨します。この構成では、デバッガがターゲットの VDD を検出して、ターゲットが低電圧動作の場合にレベル変換を実行します。デバッガが VDD ライン (インターフェイスコネクタの 2 ピン) の電圧を検出できない場合、デバッガは動作しません。

2.4.4 デバッガから電源を供給する場合

デバッガの内部電源は 30 mA までに制限されています。この電源供給方法は、デバイスの VDD が他のアプリケーション回路から分離されて独立プログラミングを行えるようなきわめて小規模のアプリケーションには適していますが、USB バスパワーとして PC から流れる電流が大きくなるため、一般的な用途には推奨できません。

デバイスによっては AVDD ラインと AVSS ラインがない場合もありますが、これらのラインがターゲットデバイスにある場合、デバッガが動作するにはこれらを全て適切なレベルに接続する必要があります。フローティングのままにする事はできません。

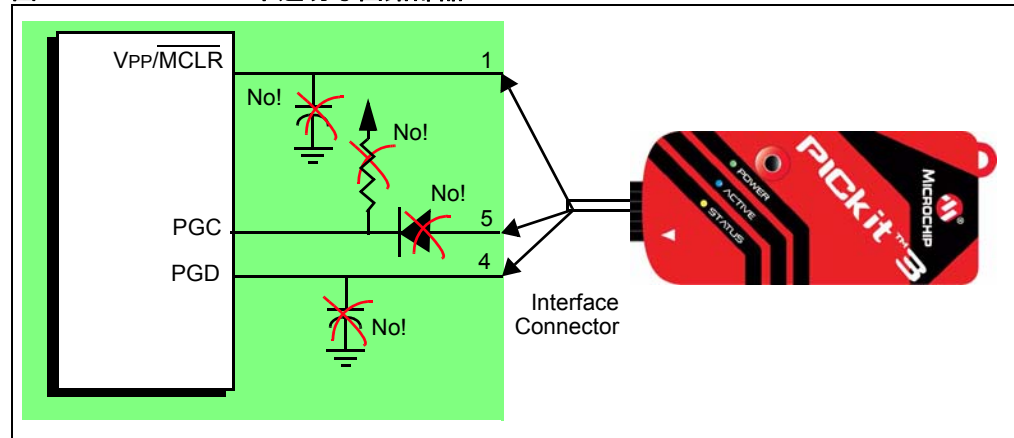
また、VCAP ラインのあるデバイス (PIC18FXXJ 等) は、適切なコンデンサまたはレベルに接続する必要があります。

Note: 配線は非常にシンプルです。問題が発生する場合、これらの重要なラインに他の接続または部品を取り付けているために PICKit 3 プログラマ / デバッガの正常な動作が妨げられている場合がほとんどです。詳細については、次のセクションで説明します。

2.4.5 デバッガの正常動作を妨げる回路

図 2-5 は、デバッガのアクティブなラインに PICkit 3 デバッガシステムの正常動作を妨げる部品を取り付けた様子を示したものです。

図 2-5: 不適切な回路部品



具体的には、以下のガイドラインに従う必要があります。

- PGC/PGD にプルアップ抵抗を使用しない — これらのラインはデバッガ内部に 4.7 kΩ のプルダウン抵抗があるため、プルアップ抵抗を使用すると電圧レベルが低下します。
- PGC/PGD にコンデンサを使用しない — コンデンサを使用すると、プログラミングとデバッグの通信の際にデータ/クロックラインの高速遷移が妨げられます。
- MCLR にコンデンサを使用しない — VPP の高速遷移が妨げられます。通常、シンプルなプルアップ抵抗で十分です。
- PGC/PGD にダイオードを使用しない — デバッガとターゲット デバイスの双方向通信が行えなくなります。

2.5 デバッグ

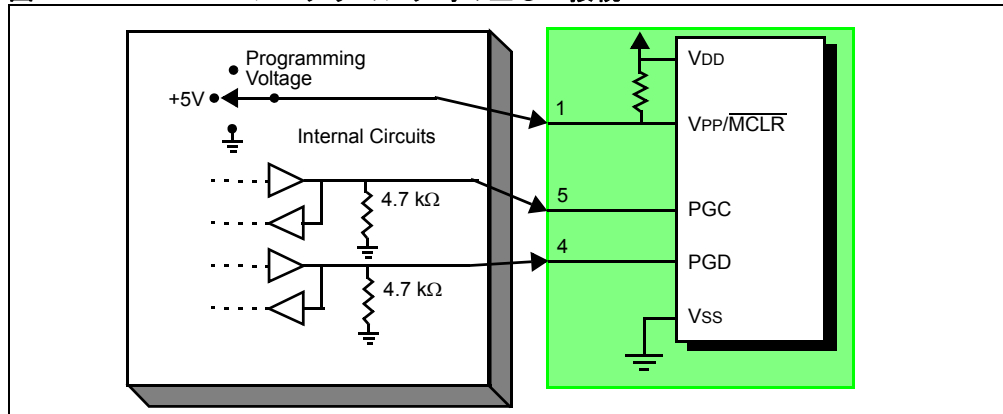
PICkit 3 プログラマ / デバッガシステムをデバッガとして使用するには、2 段階の手順があります。まず、アプリケーションをターゲット デバイスにプログラムします (通常は PICkit 3 をプログラマとして使います)。次に、ターゲットのフラッシュ デバイスに内蔵されたインサーキット デバッグ ハードウェアを使ってアプリケーションプログラムを実行、テストします。この 2 つの手順は、MPLAB IDE での以下の操作に対応しています。

1. コードをターゲット デバイスにプログラムし、専用のデバッグ機能を有効にする (次セクション参照)。
2. デバッガを使ってブレークポイントを設定し、実行する。

ターゲット デバイスに正しくプログラムできないと、PICkit 3 プログラマ / デバッガでデバッグが行えません。

図 2-6 に、プログラミングに必要な基本の配線を示します。これは図 2-4 と同じものですが、分かりやすくするためにデバッガの VDD ラインと VSS ラインは省略しています。

図 2-6: プログラミング時の正しい接続



この図は、PICKit 3 プログラマ/デバッガの内部インターフェイス回路の一部を簡単に示したものです。プログラミング時はターゲット デバイスにクロックは不要ですが、電源は供給する必要があります。プログラミングを行う際、デバッガは VPP/MCLR をプログラミング レベルにし、PGC にクロックパルス、PGD にシリアルデータを送出します。ターゲット デバイスへのプログラムが正しく完了した事を確認するために、PGC にクロックを送出し、PGD からデータを読み出します。これは、ターゲット デバイスの ICSP プロトコルに従って行われます。

2.6 デバッグのための要件

PICKit 3 プログラマ/デバッガシステムでデバッグ (ブレークポイントの設定、レジスタの参照等) を行うために、正しく実施する必要のある重要な項目は以下の通りです。

- デバッガが PC に接続されている必要があります。USB ケーブルを利用して PC からデバッガに電源を供給する事、USB ケーブルを利用して MPLAB IDE ソフトウェアとデバッガが通信を行う事が必要です (詳細は Chapter 3. 「インストール」 参照)。
- 図に示したように、モジュラ インターフェイス ケーブル (または同等品) を使ってデバッガがターゲット デバイスの VPP、PGC、PGD ピンに接続されている必要があります。VSS と VDD もデバッガとターゲット デバイスの間で接続しておく必要があります。
- ターゲット デバイスに電源が供給されており、オシレータが正しく動作している必要があります。何らかの理由でターゲット デバイスが動作しない場合、PICKit 3 プログラマ/デバッガでデバッグを行う事はできません。
- ターゲット デバイスのコンフィグレーション ワードが以下に示すように正しくプログラムされている必要があります。
 - ターゲットの設計に応じてオシレータのコンフィグレーション ビットを RC、XT 等に正しく設定しておく必要があります。
 - ウォッチドッグ タイマが既定値で有効になっているデバイスでは、無効にしておく必要があります。
 - ターゲット デバイスのコード保護は無効にしておく必要があります。
 - ターゲット デバイスのテーブル読み出し保護は無効にしておく必要があります。
- PGM (LVP) は無効にしておく必要があります。

上記の条件が満たされている事を確認したら、次の項に進んでください。

- デバッグ開始までの操作手順
- デバッグの詳細

2.6.1 デバッグ開始までの操作手順

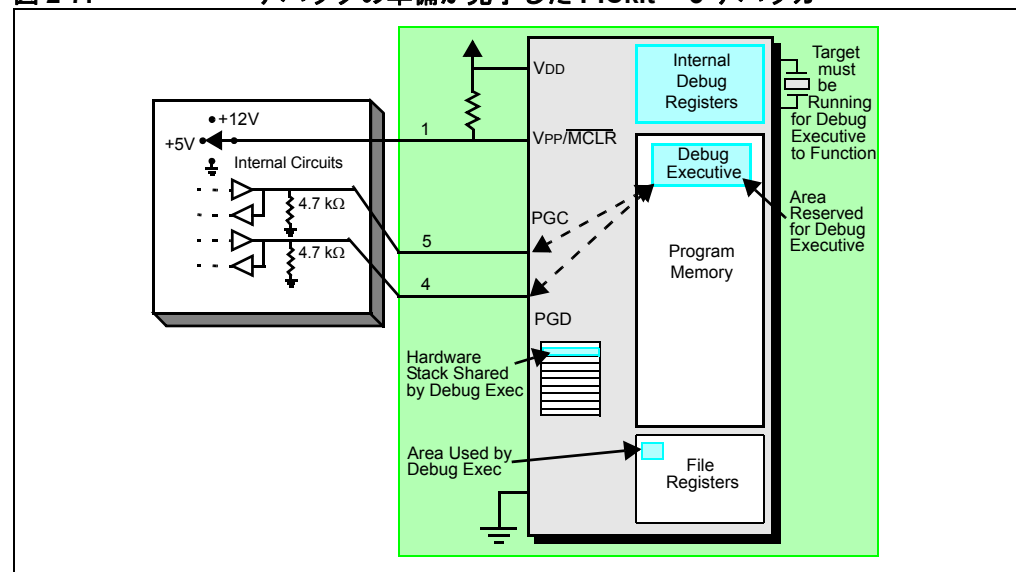
前セクションでデバッグ実行のための要件に記載した要件が満たされている場合、MPLAB IDE のメニューで [Debugger]>[Select Tool]>[PICkit 3] の順に選択し、現在のデバッガとして PICkit 3 プログラマ / デバッガを設定すると、以下の操作が実行できます。

- [Project]>[Build Configuration]>[Debug] の順に選択し、アプリケーションコードをコンパイル / アセンブルする。
- [Debugger]>[Program] の順に選択し、前述の ICSP プロトコルを利用してアプリケーションコードをデバイスのメモリに書き込む。
- MPLAB IDE によって、サイズの小さい「デバッガ実行プログラム」がターゲットデバイスのプログラムメモリの上位領域に自動的に読み込まれる。デバッガ実行プログラムはプログラムメモリに常駐する必要があるため、アプリケーションプログラムはこの予約領域を使用する事はできません。一部のデバイスには、デバッガ実行プログラム専用のメモリ領域が用意されたものもあります。詳細はデバイスのデータシートで確認してください。
- ターゲットデバイスの「インサーキット デバッグ」専用レジスタが有効になる。これにより、デバッガからデバッガ実行プログラムを有効にする事ができます。
- VPP/MCLR ラインを Low に保つ事によって、ターゲット デバイスはリセット状態に保持される。

2.6.2 デバッグの詳細

図 2-7 に、デバッグの準備が完了した PICkit 3 プログラマ / デバッガシステムを示します。

図 2-7: デバッグの準備が完了した PICkit™ 3 デバッガ



一般に、アプリケーションプログラムが正しく動作するかどうかを確認するには、プログラムコードの最初の方にブレークポイントを設定します。MPLAB IDE のユーザーインターフェイスでブレークポイントを設定すると、そのアドレスがターゲットデバイス内部のデバッグ専用レジスタに保存されます。PGC と PGD に対するコマンドは、これらのレジスタと直接通信してブレークポイントのアドレスを設定します。

次は通常、MPLAB IDE で [Debugger]>[Run] の順にクリックするか、または [Run] アイコン (右向きの矢印) をクリックします。すると、デバッガからの命令によってデバッガ実行プログラムが実行されます。ターゲットはリセットベクタから動作を開始し、プログラムカウンタが内部デバッグレジスタに保存されたブレークポイントのアドレスに達すると実行が停止します。

ブレークポイントのアドレスの命令が実行されると、割り込みと同じようなメカニズムでターゲットデバイスのインサーキットデバッグメカニズムが「起動」してデバイスのプログラムカウンタをデバッガ実行プログラムに転送し、ユーザのアプリケーションは停止します。デバッガは PGC と PGD を介してデバッガ実行プログラ

ムと通信し、ブレークポイントのステータス情報を取得して MPLAB IDE に返します。この後、MPLAB IDE は一連の問い合わせをデバッガに送信し、ファイルレジスタの内容や CPU の状態等、ターゲット デバイスに関する情報を取得します。これらの問い合わせは、最終的にはデバッガ実行プログラムによって実行されます。

デバッガ実行プログラムは、プログラムメモリ内のアプリケーションとまったく同じように動作します。一時変数を保存するためにスタックの一部を使用します。オシレータがない、電源接続が不良、ターゲットボードの短絡等、何らかの理由でデバイスが動作しない場合、デバッグ実行プログラムと PICkit 3 プログラマ / デバッガが通信できず、MPLAB IDE にエラーメッセージが表示されます。

ブレークポイントを取得するもう 1 つの方法は、MPLAB IDE の **[Halt]** ボタン ([Run] アイコンの矢印の右にある一時停止のアイコン) をクリックする事です。すると、PGC ラインと PGD ラインが切り換わり、ターゲット デバイスのインサーキット デバッグ メカニズムの働きにより、プログラム カウンタがプログラム メモリ内のユーザコードからデバッガ実行プログラムに切り換わります。この場合もターゲット アプリケーションのプログラム実行が停止し、MPLAB IDE はデバッガとデバッガ実行プログラムの間で行われる通信を利用してターゲット デバイスの状態を問い合わせます。

2.7 プログラミング

PICkit 3 を使ってデバイスをプログラムするには、以下の 3 つの方法があります。

- PICkit 3 と PC を接続し、MPLAB IDE を使う。
- MPLAB IDE で設定後、PICkit 3 の Programmer-To-Go 機能を使う (詳細は **Chapter 6. 「PICkit 3 Programmer-To-Go」** 参照)。
- PICkit 3 プログラマ アプリケーション (MPLAB IDE を使用せずに PICkit 3 でデバイスをプログラムするためのソフトウェア) を利用する (詳細は『PICkit™ 3 Programmer Application User's Guide』参照)。

PICkit 3 プログラマ / デバッガをプログラマとして使うと、実際のデバイス (すなわち、ヘッダボードに装着された -ICE/-ICD 以外のデバイス) に対してプログラムできます。MPLAB IDE のメニューで **[Programmer]>[Select Programmer]** の順にクリックして「PICkit 3」を選択した後、MPLAB IDE のツールバーで **[Build Configuration]** リストボックスから「Release」を選択してアプリケーション コードのコンパイル / アセンブルを実行します。または、**[Project]>[Build Configuration]>[Release]** の順にクリックしても「Release」に設定できます。

デバッガをプログラマとして使う場合、デバッグ機能は全て OFF になるか解除されます。**[Programmer]>[Program]** 順に選択してデバイスをプログラムすると、MPLAB IDE によってインサーキット デバッグ レジスタが無効にされるため、PICkit 3 プログラマ / デバッガはターゲット アプリケーション コードとコンフィグレーション ビット (さらに、EEPROM 装備のデバイスで選択した場合は EEPROM データ) のみをターゲット デバイスにプログラムします。デバッガ実行プログラムは読み込まれません。プログラマとして使う場合、デバッガは MCLR ラインをトグルしてターゲットをリセット / 起動する事しか行えません。ブレークポイントの設定、レジスタ内容の参照または変更はできません。

PICkit 3 プログラマ / デバッガシステムは、ICSP を利用してターゲット デバイスをプログラムします。VPP、PGC、PGD ラインは、前述した通りに接続してください。プログラミング時にはクロックは不要です。また、コード保護、ウォッチドッグ タイマ、テーブル読み出し保護を含め、プロセッサの全てのモードをプログラミングできます。

2.8 デバッガが使うリソース

各デバイスで PICkit 3 プログラマ / デバッガが使用するリソースの一覧は、MPLAB IDE のオンラインヘルプ ファイルを参照してください。

Chapter 3. インストール

3.1 はじめに

ここでは、PICKit 3 プログラマ / デバッガシステムのインストール方法について説明します。

- ソフトウェアのインストール
- ターゲットの接続
- ターゲットボードの設定
- MPLAB® IDE の設定

3.2 ソフトウェアのインストール

MPLAB IDE ソフトウェアをインストールするには、まず最新の MPLAB IDE インストール実行ファイル (MPxxxxxx.exe、xxxxxx は MPLAB IDE のバージョン) をマイクロチップ社のウェブサイト (www.microchip.com) からダウンロードするか、MPLAB IDE の CD-ROM (DS51123) を入手します。次に、このインストール実行ファイルを実行し、画面の指示に従って MPLAB IDE をインストールします。

Note: PICKit 3 プログラマ / デバッガを使用するには、MPLAB IDE v8.20 以上が必要です。

3.3 ターゲットの接続

PICKit 3 プログラマ / デバッガには、ターゲットとの通信方法を選択するための接続回路が内蔵されています。詳細と接続図は、2.3「デバッガとターゲットの通信」を参照してください。

1. USB / 電源ケーブルを接続する。
2. デバッガとターゲットを通信ケーブルで接続する (RJ11 プラグを使う場合) か、または 6 ピンインラインヘッダに直接接続する。

図 3-1: 通信ケーブルと USB / 電源ケーブルの接続



3.4 ターゲットボードの設定

3.4.1 量産デバイスを使う場合

量産デバイスを使う場合、デバッグをターゲットボードに直接接続します。PICkit 3 プログラマ/デバッグでエミュレーションを行うには、ターゲットボード上のデバイスがデバッグ回路を内蔵している必要があります。必要なデバッグ回路を内蔵しているかどうかは、各デバイスのデータシートで確認してください(デバッグ回路を内蔵するデバイスには「バックグラウンド デバッグ イネーブル」コンフィギュレーションビットがあります)。

Note: 将来的には、ICD をサポートする回路を内蔵したデバイスも使用可能になる予定です。

ターゲットボードには、デバッグで使用する通信方法に対応したコネクタが必要です。接続の詳細は、2.3「デバッグとターゲットの通信」の「標準 ICSP による通信」を参照してください。

3.4.2 ICE デバイスを使う場合

ICE デバイスを使う場合、ICE ヘッドボードが必要です。このヘッドボードは、特定のデバイスまたはデバイスファミリのエミュレーションに必要なハードウェアを実装しています。ICE ヘッドの詳細は、『Header Board Specification』(DS51292)を参照してください。

Note: 将来的には、ICD デバイス(デバイス名 -ICD)を搭載した ICD ヘッドボードも使用可能になる予定です。

ICE ヘッドをターゲットボードに接続するには、変換ソケットを使用します。共通のヘッドを各種の表面実装パッケージに接続できるように、各種変換ソケットを用意しています。変換ソケットの詳細は、『Transition Socket Specification』(DS51194)を参照してください。

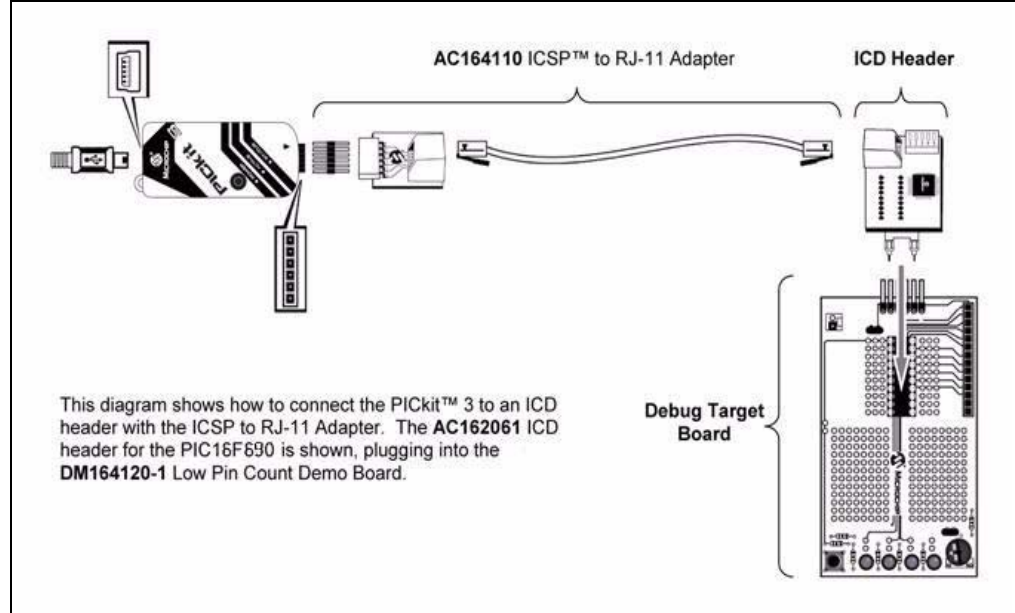
ヘッドボードのレイアウトは、ヘッドまたはプロセッサ拡張パックごとに異なります。接続の詳細は、2.3「デバッグとターゲットの通信」の「標準 ICSP による通信」を参照してください。

3.4.3 ICD ヘッドを使う場合

PIC MCU のベースライン全製品と一部のミッドレンジ製品の場合、デバッグ機能を利用するには特別な「-ICD」デバイスを実装したデバッグヘッドボードが必要です。これらのデバイスの種類と必要な ICD ヘッドボードの製品番号は、『Header Board Specification』(DS51292)を参照してください。ヘッドボードの仕様は PICkit 3 の CD-ROM に収録している他、オンライン (www.microchip.com) でも公開しています。

ICD ヘッドボードは「-ICD」デバイスを実装しており、量産デバイスの代わりにターゲットボードで使用します。しかし、ほとんどのヘッドボードにはデバッグ用の RJ-11 コネクタが付いており、RJ-11/ICSP アダプタキット (AC164110) を使って PICkit 3 に接続します。図 3-2 に、PIC18F45K20 の ICD ヘッド (AC162061) と AC164110 アダプタキット、少ピン デモボードを使った場合の接続を示します。

図 3-2: ICD ヘッダボードの使用



多くのミッドレンジ製品、PIC18、16 ビット製品には ICD ヘッダは不要です。ICSP プログラミング接続で直接デバッグが行えます。

3.4.4 ターゲットへの電源供給

以下に、電源供給方法に関する要点を記します。

- USB 接続を使う場合、PICkit 3 の電源は PC から供給できますが、電流量に制限があり、小型のターゲットボードに最大 30 mA (V_{DD} 1.8 ~ 5 V) しか供給できません。
- 推奨するのは、ターゲットで V_{DD} を自給する方法です。この方法だと、より大きな電流が得られます。もう 1 つの利点として、プラグ & プレイによるターゲット検出機能が継承される点が挙げられます。つまり、MPLAB IDE がターゲットとデバイスを検出すると、その内容が [Output] ウィンドウに表示されます。

Note: ターゲット電圧は、ICSP インターフェイスのドライバに対してのみ電源を供給します。PICkit 3 には電源供給を行いません。PICkit 3 の電源は USB ポートからのみ供給されます。

PICkit 3 とターゲットを適切なケーブルで接続します (3.3「ターゲットの接続」参照)。次に、ターゲットの電源を投入します。PICkit 3 からターゲットに電源を供給する場合、10.5.8「[Settings] ダイアログの [Power] タブ」の説明を参照してください。

3.5 MPLAB® IDE の設定

ハードウェアを接続して電源を投入したら、MPLAB IDE で PICkit 3 プログラマ / デバッガを使用できるように設定します。

一部のデバイスでは、コンフィグレーションビットで通信チャンネルを選択する必要があります (PGC1/EMUC1、PGD1/EMUD1 等)。ここで選択したピンが、デバイスに物理的に接続されているピンと同じであることを確認してください。

プロジェクトを設定して PICkit 3 の使用を開始する手順については、Chapter 4.「全般的な設定」を参照してください。

NOTES:

Chapter 4. 全般的な設定

4.1 はじめに

ここでは、PICKit 3 プログラマ / デバッガで作業を始める方法を説明します。

- MPLAB IDE ソフトウェアの起動
- プロジェクトの作成
- プロジェクトの表示
- プロジェクトのビルド
- コンフィグレーション ビットの設定
- デバッガまたはプログラマとしての設定
- デバッガ / プログラマの制約

4.2 MPLAB IDE ソフトウェアの起動

MPLAB IDE ソフトウェアのインストールが完了したら (3.2「ソフトウェアのインストール」)、以下のいずれかの方法で起動します。

- [スタート]>[すべてのプログラム]>[Microchip]>[MPLAB IDE vx.xx]>[MPLAB IDE] の順に選択する (「vx.xx」はバージョン番号)。
- デスクトップにある MPLAB IDE のアイコンをダブルクリックする。
- MPLAB IDE のインストール先ディレクトリの `mplab_ide\core` サブディレクトリにある `mplab.exe` を実行する。

MPLAB IDE ソフトウェアの使用法の詳細は以下の文書を参照してください。

- 『MPLAB® IDE User's Guide』 (DS51519) — MPLAB IDE の使用法を詳しく解説したガイドです。
- オンラインヘルプ ファイル — MPLAB IDE と PICKit 3 プログラマ / デバッガに関する最新情報を記載しています。
- Readme ファイル — 各リリース直前に発生した情報を `Readme for MPLAB IDE.txt` と `Readme for PICKit 3 Debugger.txt` に記載しています。これらのファイルは、どちらも MPLAB IDE のインストール先ディレクトリの `Readmes` サブディレクトリにあります。

4.3 プロジェクトの作成

新規プロジェクトを簡単に作成するには、[Project]>[Project Wizard] の順にクリックしてプロジェクト ウィザードを使用します。プロジェクト ウィザードの指示に従って操作すると、新規プロジェクトを作成し、そのプロジェクトのビルドに使用する言語ツールを選択できます。ウィザードの手順に従ってソースファイル、ライブラリ等を追加すると、プロジェクト ウィンドウの各種「ノード」に追加されます。ウィザードの使用法の詳細は MPLAB IDE のマニュアルを参照してください。基本的な手順は以下の通りです。

- デバイスを選択する (例 : PIC18F45K20)。
- 言語ツールスイートを選択する (Microchip C Compiler Toolsuite 等)。
- プロジェクト名を指定する。
- アプリケーション ファイルを追加する (`program.c`、`support.s`、`counter.asm` 等)。

Note: プロジェクトにカスタム リンカスクリプトがない場合、プロジェクト マネージャによって適切なリンカスクリプトが自動的に選択されます。

4.4 プロジェクトの表示

プロジェクトウィザードによるプロジェクトの作成が完了すると、プロジェクトと関連ファイルがプロジェクトウィンドウに表示されます。プロジェクトウィンドウでツリーの任意の行を右クリックするとショートカットメニューが表示され、ファイルの追加や削除を行えます。

プロジェクトウィンドウの使用法の詳細は MPLAB IDE のマニュアルを参照してください。

4.5 プロジェクトのビルド

プロジェクトを作成したら、アプリケーションをビルドする必要があります。ビルドを行うと、アプリケーションのオブジェクト (hex) コードが作成されます。このオブジェクトコードを、PICkit 3 プログラマ / デバッガを使ってターゲットに書き込みます。

ビルドのオプションを設定するには、[Project]>[Build Options]>[Project] の順にクリックします。

Note: PICkit 3 をデバッガとして使う場合、プロジェクトマネージャ ツールバー ([View]>[Toolbars]>[Project Manager]) のドロップダウン リストで「Debug」を選択し、プログラマとして使う場合、「Release」を選択します。

設定が完了したら、[Project]>[Build All] の順にクリックしてプロジェクトをビルドします。

4.6 コンフィグレーションビットの設定

デバイスのコンフィグレーションビットはコード中でも設定できませんが、MPLAB IDE の [Configuration Bits] ウィンドウで設定することもできます。

[Configure]>[Configuration Bits] の順にクリックします。[Settings] 欄の文字列をクリックすると、設定を変更できます。

ここでは、いくつかのコンフィグレーションビットについて簡単に説明します。

- **Watchdog Timer Enable** – ほとんどのデバイスでウォッチドッグ タイマは初期状態で有効です。通常は、このビットを無効にしておくといでしょう。
- **Comm Channel Select** – 一部のデバイスでは、デバイスの通信チャンネルを選択する必要があります (例: PGC1/EMUC1、PGD1/EMUD1)。ここで選択したピンが、デバイスに物理的に接続されているピンと同じであることを確認します。
- **Oscillator** – ターゲットのオシレータと同じ設定を選択します。

4.7 デバッガまたはプログラマとしての設定

PICkit 3 プログラマ / デバッガをデバッガとして選択するには、[Debugger]>[Select Tool]>[PICkit 3] の順にクリックします。デバッガを選択すると、[Debugger] メニューと MPLAB IDE のツールバーにはデバッガの機能が表示されます。また、[Output] ウィンドウが開き、[PICkit 3] タブに PICkit 3 のステータスおよび通信に関するメッセージが表示されます。詳細は、10.2「デバッグに関する機能」と 10.3「デバッグに関するダイアログ / ウィンドウ」を参照してください。

PICkit 3 プログラマ / デバッガをプログラマとして選択するには、[Programmer]>[Select Programmer]>[PICkit 3] の順にクリックします。プログラマを選択すると、[Programmer] メニューと MPLAB IDE のツールバーにはプログラマの機能が表示されます。また、[Output] ウィンドウが開き、[PICkit 3] タブに ICE のステータスおよび通信に関するメッセージが表示されます。詳細は、10.4「プログラミングに関する機能」を参照してください。

オプションの設定は、[Debugger]>[Settings] または [Programmer]>[Settings] の順にクリックして [Settings] ダイアログ (10.5「[Settings] ダイアログ」) で行います。

エラーが発生した場合、以下を参照してください。

- Chapter 9.「エラーメッセージ」
- Chapter 8.「よく寄せられる質問 (FAQ)」

4.8 デバッガ / プログラムの制約

個別のデバイスに対するデバッガの制約については、MPLAB IDE の PICkit 3 オンライン ヘルプ ファイルを参照してください ([Help]>[Topics]>[PICkit 3] の順にクリックして、[OK] をクリック)。

PICKit™ 3 ユーザガイド

NOTES:

Chapter 5. PICkit 3 Debug Express

5.1 はじめに

PICkit 3 Debug Express キットを MPLAB IDE アプリケーションと組み合わせると、プログラムの実行、停止、シングルステップ実行を行えます。複数のブレークポイントの設定、プロセッサのリセットも可能です。プロセッサを停止させて、レジスタの内容を確認 / 変更する事もできます。

MPLAB IDE の使用方法の詳細は、以下のマニュアルを参照してください。

- 『MPLAB® IDE User's Guide』 (DS51519)
- MPLAB® IDE オンラインヘルプ

5.2 PICkit 3 DEBUG EXPRESS キットの内容

PICkit 3 Debug Express キット (DV164131) には、以下のものが含まれます。

1. PICkit 3 プログラマ / デバッグ
2. USB ケーブル
3. デバイス実装済みの 44 ピン デモボード *
4. MPLAB IDE CD-ROM
5. PICkit 3 Debug Express C18 のレッスン (CD-ROM 収録のチュートリアル)

* Explorer 16 ボードを使用したデバッグも可能です。

5.3 ハードウェアとソフトウェアのインストール

PICkit 3 のハードウェアとソフトウェアをまだインストールしていない場合、**Chapter 3. 「インストール」** の手順に従ってインストールします。

Note: PICkit 3 Debug Express を使用するには、MPLAB IDE バージョン 8.20 以上が必要です。

5.3.1 予約リソース

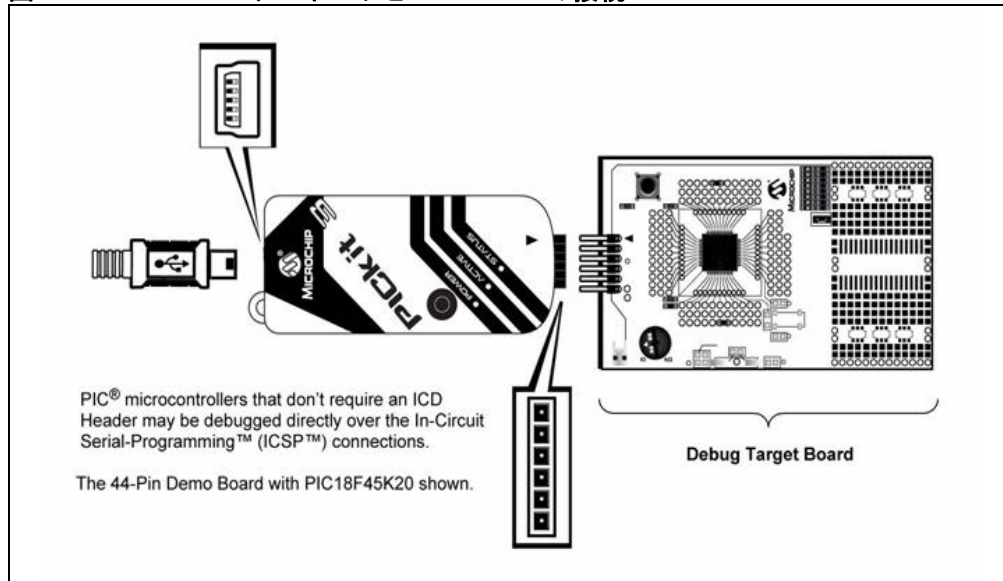
ICD デバイス内蔵のインサーキット デバッグ機能とデバッグの ICSP 機能を使うため、PICkit 3 Debug Express はデバッグ実行時、内部リソースの一部を使います。

インサーキット デバッグ時に必要なデバイスリソースの詳細は、MPLAB IDE のメニューで **[Help]>[Topics]** の順に選択して、MPLAB PICkit 3 ヘルプで見ることができます。「Resources Used By MPLAB PICkit 3」の項に記載されているデバイスの予約リソースを PICkit 3 Debug Express も使用します。

5.3.2 デモボードの接続

44 ピン デモボードに実装されている PIC18F45K20 は、図 5-1 に示すようにデモボードを PICkit 3 に接続するだけでデバッグできます。

図 5-1: デモボードと PICKit™ 3 の接続



5.3.3 コンフィグレーションビットと Debug Express

ICD ヘッダなしで直接デバッグが可能な PIC マイクロコントローラには、コンフィグレーションワードに DEBUG ビットがあり、このビットで PIC マイクロコントローラのデバッグモードの有効/無効を切り換えます。

MPLAB IDE で PICKit 3 Debug Express を使う場合、このビットは自動的にセットされるため、ソースコードのコンフィグレーション設定では明示的に指定しないでください。

CAUTION

通常は、DEBUG コンフィグレーションビットの値をソースコードのコンフィグレーション設定で指定しないでください。指定すると、このデバッグ以外でデバイスをプログラムする際にこのビットがアサートされる可能性があります。このビットがアサートされるとアプリケーション回路でデバイスが正しく動作しないか、または全く動作しなくなります。

PIC24 や dsPIC33 ファミリー等、多くの 16 ビット PIC マイクロコントローラには、複数の ICSP プログラミング/デバッグポートピン (PGC1/EMUC1 と PGD1/EMUD1、PGC2/EMUC2 と PGD2/EMUD2 等) があります。プログラミングの場合は任意の ICSP ポートを使用できますが、デバッグの場合は一度にある 1 つのポートだけがアクティブになります。このアクティブな EMU ポートはデバイスのコンフィグレーションビットで設定します。アクティブポートの設定が、PICKit 3 が接続されている EMU ポートに一致していないと、デバイスはデバッグモードに移行できません。MPLAB IDE の [Configuration Bits] ダイアログでは、これらのビットを [Comm Channel Select] ビットと表示します。

Chapter 6. PICKit 3 Programmer-To-Go

6.1 はじめに

PICKit 3 Programmer-To-Go を使うと、PIC MCU のメモリイメージを PICKit 3 本体にダウンロードして、後で PIC MCU にプログラムできます。つまり、この機能を使うとソフトウェアまたは PC がなくてもデバイスをプログラムできます。必要なのは PICKit 3 の USB 電源のみです。

Note: PICKit 3 にはプログラマとデバッグの機能がありますが、Programming-To-Go 機能を利用する場合はプログラマの機能のみが利用できます。Programming-To-Go ではデバッグ機能は利用できません。

本セクションでは、以下の内容について説明します。

- PICKit 3 Programmer-To-Go の USB 電源
- PICKit 3 Programmer-To-Go 対応デバイス
- PICKit 3 で Programmer-To-Go 機能を使うための設定
- PICKit 3 の Programmer-To-Go 機能の使い方
- Programmer-To-Go モードの終了

6.2 PICKit 3 PROGRAMMER-TO-GO の USB 電源

PICKit 3 プログラマ ハードウェアは、ターゲットの ICSP コネクタの VDD ピンから供給される電源だけでは動作しません。従って、PICKit 3 本体上部の USB mini-B ポートから 5 V 電源を供給する必要があります。これには、以下のようにいくつかの方法があります。

- PC の USB ポートまたは USB ハブのポート (バスパワーのみ使用、USB 通信は不要)
- 携帯型機器の USB ホストポート
- USB mini-B コネクタ対応の USB 電源アダプタ / 充電器 (自動車のシガレット ライター電源または家庭用コンセントから給電)
- 携帯電話 / モバイル機器用の USB mini-B コネクタ出力付きポータブル充電器 / 電源
- レギュレートされた 5 V を PICKit 3 の USB ポートに供給するカスタム バッテリーパック

6.2.1 電源要件

使用する USB 電源は以下の最小条件を満たす必要があります。

- PICKit 3 に 100 mA 以上の電流を供給できる事
- レギュレートされた 4.5 ~ 5.5 V の電圧を安定的に出力できる事

Note 1: ほとんどのバッテリー内蔵型のポータブル充電器 / 電源は、内部のバッテリー電圧が低下して出力電圧が 4.5 V を下回った時にその事を示す手段がありません。従ってバッテリーの残り容量には十分注意し、PICKit 3 に必ず 4.5 V 以上の電圧が供給されるようにしてください。

2: バッテリー駆動型の電源は使用しない時は PICKit 3 から外しておきます。PICKit 3 に接続したままにしておくとバッテリーが消耗します。

6.3 PICKit 3 PROGRAMMER-TO-GO 対応デバイス

下記のファミリのうち、PICKit 3 と MPLAB IDE がサポートするデバイスは全て Programmer-To-Go 機能に対応しています。表 6-1 に、サポートされるデバイスファミリとそれぞれのプログラムメモリの制約を示します。

表 6-1: PROGRAMMER-TO-GO 対応デバイス

デバイスファミリ	対応デバイス
ベースライン	全て ¹
ミッドレンジ	全て ¹
PIC18F	全て ¹
PIC18 J シリーズ	全て ¹
PIC18 K シリーズ	全て ¹
PIC24	全て ^{1,2}
dsPIC33	全て ^{1,2}
dsPIC30	全て ¹
dsPIC30 SMPS	全て ¹
PIC32MX	全て ³

Note 1: このファミリのうち、MPLAB IDE でサポートされているデバイスは全てサポートされます。MPLAB IDE でサポートされているデバイスの一覧は、メニューで [\[Help\]>\[ReadMe\]](#) の順にクリックして確認してください。メモリ容量の大きいデバイスはサポートされています。

2: PICKit 3 の Programmer-To-Go 機能では、これらのデバイスの拡張 ICSP (Programming Executive) は使用できません。これらのデバイスで PICKit 3 の Programmer-To-Go 機能を利用する場合、通常の ICSP でプログラミングが行われます。

3: MPLAB IDE v8.53 以上でサポートされています。

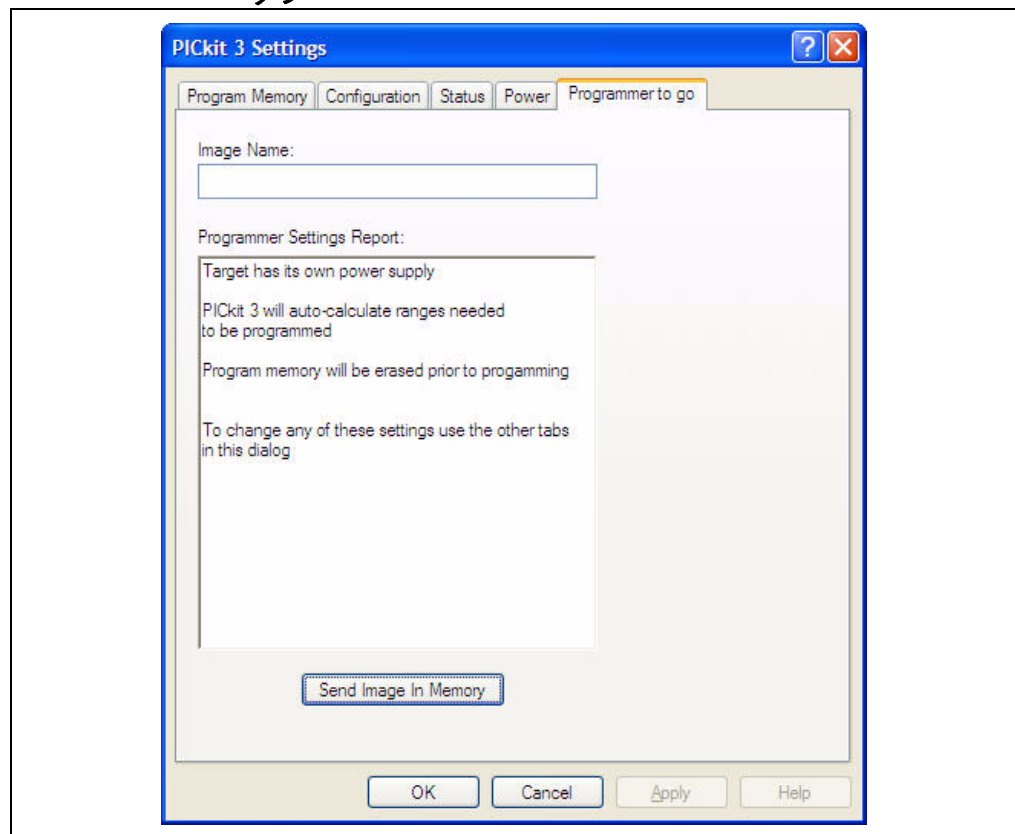
6.4 PICKit 3 で PROGRAMMER-TO-GO 機能を使うための設定

メモリイメージを PICKit 3 にダウンロードして Programmer-To-Go 機能を利用する前に、PICKit 3 プログラマのソフトウェア オプションとバッファを Programmer-To-Go 用に設定する必要があります。実際には、必要なオプションを全て設定してまずソフトウェアからターゲット デバイスにプログラミングを実行し、デバイスに正しくプログラミングが行える事を確認してから Programmer-To-Go 用にイメージをダウンロードする事を推奨します。MPLAB IDE と PICKit 3 を使ってデバイスをプログラムする方法については、= Chapter 4.「全般的な設定」を参照してください。

6.4.1 [PICKit 3 Settings] ダイアログの [Programmer to go] タブ

MPLAB IDE のメニューで [Programmer]>[Programmer to go Tab] の順に選択します (図 6-1)。

図 6-1: [PICKit 3 SETTINGS] ダイアログの [PROGRAMMER TO GO] タブ

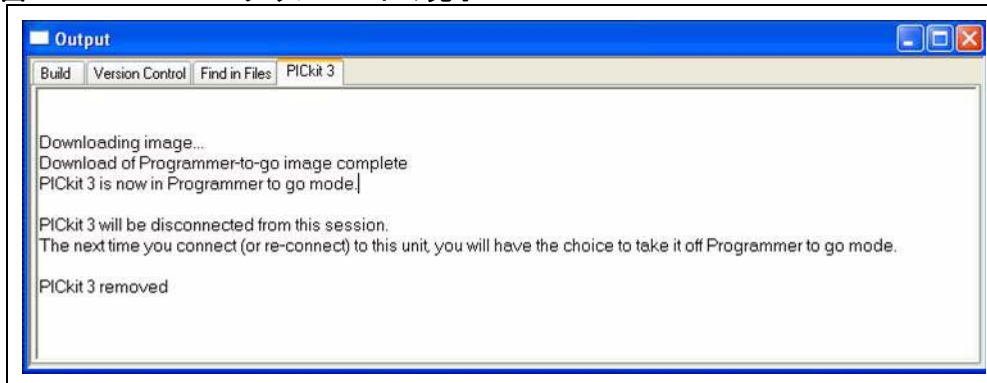


- [Image Name] フィールドには、プログラミング イメージに使う名前を入力します。
- [Programmer Setting Report] パネルには、デバイスのプログラミングに関して選択した設定が表示されます。設定の変更は、[PICKit 3 Settings] ダイアログの対応するタブで行います。
- **[Send Image In Memory]** をクリックすると、PICKit 3 へのイメージ転送が始まります。イメージが PICKit 3 に格納されたら、MPLAB IDE または PC がなくてもデバイスをプログラムできます。PICKit 3 のボタンを押すだけです。

6.4.2 PICKit 3 へのダウンロードの完了

[Send Image In Memory] をクリックした後、ダウンロードが完了すると [Output] ウィンドウにメッセージ (図 6-2) が表示されます。PICKit 3 本体の「ACTIVE」LED が点滅していれば、Programmer-To-Go モードへの移行が完了して書き込み準備が整った事を示します。

図 6-2: ダウンロードの完了



PC の USB ポートから PICKit 3 を取り外します。次に USB ケーブルから PICKit 3 に電源を供給すると「ACTIVE」LED が 1 回点滅します。これは、PICKit 3 が Programmer-To-Go で起動した事を示します。

6.5 PICKit 3 の PROGRAMMER-TO-GO 機能の使い方

PICKit 3 で Programmer-To-Go 機能を使うための設定が完了したら、以下の手順でターゲット デバイスをプログラムします。

1. **6.2 「PICKit 3 Programmer-To-Go の USB 電源」** で説明した方法で、PICKit 3 に USB 電源を接続する。
2. PICKit 3 本体の「POWER」LED が点灯している事を確認する。PICKit 3 が Programmer-To-Go モードになり、プログラムする準備が完了すると、「ACTIVE」LED が 1 回点滅する。
3. PICKit 3 の ICSP コネクタをターゲットに接続する。PICKit 3 から電源を供給しない場合、ターゲットに適切に電源が供給されている事を確認する。
4. PICKit 3 のプッシュボタンを押すと、プログラム動作が開始する。

プログラム動作中は、PICKit 3 の「STATUS」LED がオレンジ色に変わり、動作が終了するまで常時点灯します。

プログラム動作が終了すると、PICKit 3 の LED がエラーの有無を示します。成功した場合は「STATUS」LED が緑に点灯します。プログラムエラーがあった場合は赤に点灯します。表 6-2 に LED 表示と意味を示します。

表 6-2: PROGRAMMER-TO-GO 動作後の LED 表示と意味

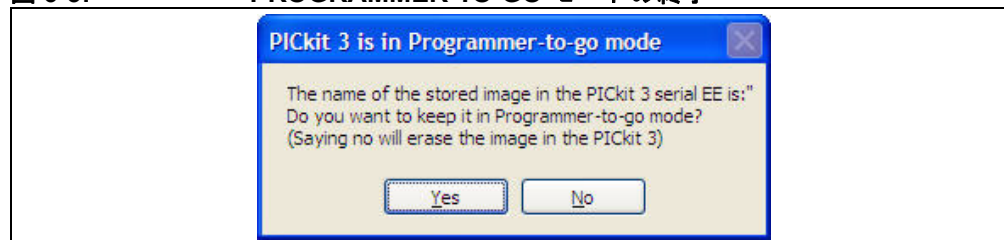
LED のステータス		意味	
ACTIVE LED	STATUS LED	意味	説明
1 回点滅 (青)	緑	成功 / 準備完了	プログラム動作中にエラーは発生しませんでした。PICkit 3 の Programmer-To-Go 機能で再度書き込みが可能です。
消灯	赤 - 間隔をあげず繰り返し点滅: ●●●●●●●●.....	VDD/VPP エラー	PICkit 3 が VDD または VPP の電圧を所定の値に設定できませんでした。PICkit 3 が VDD を供給していない場合、VPP エラーです。VDD と VPP の詳細は、 2.4「通信の接続」と 3.4.4「ターゲットへの電源供給」 を参照してください。
消灯	赤 - 2 回ずつ繰り返し点滅: ●●●●●●●●.....	デバイス ID エラー	PICkit 3 がターゲットから予測しないデバイス ID を受信しました。PICkit 3 の Programmer-To-Go 設定時に選択したデバイスとターゲットデバイスが一致する事を確認します。ICSP 接続に問題があり、PICkit 3 がターゲットと通信できていない可能性もあります。ベースライン デバイスではこのエラーは発生しません。
消灯	赤 - 3 回ずつ繰り返し点滅: ●●●●●●●●.....	ベリファイエラー	プログラム後にターゲットのベリファイに失敗しました。ターゲットの VDD が最小要件を満たしている事を確かめます。ベースライン デバイスの場合、このエラーは ICSP 通信の問題を示す可能性があります。
消灯	赤 - 4 回ずつ繰り返し点滅: ●●●● ●●●●.....	内部エラー	Programmer-To-Go で予測しない内部エラーが発生しました。このエラーが繰り返し発生する場合、PICkit 3 へのダウンロードをやり直します。

Note: PICkit 3 のプッシュボタンを押すとエラーコードがクリアされ、新しいプログラム動作を開始できます。

6.6 PROGRAMMER-TO-GO モードの終了

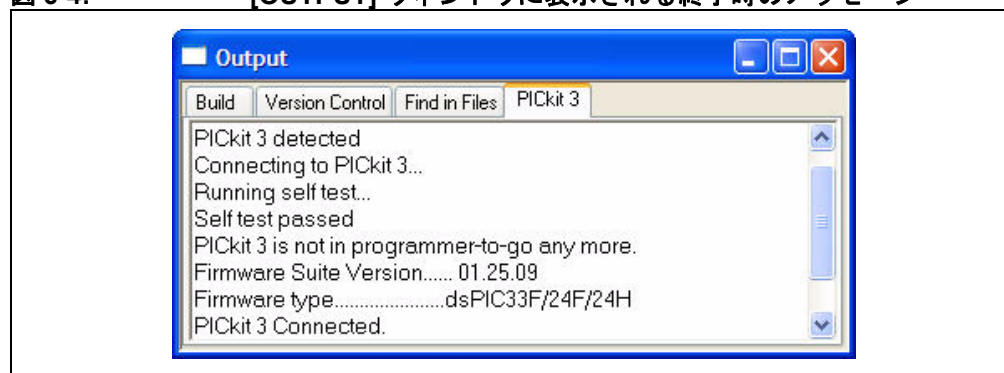
Programmer-To-Go モードを終了するには、PICKit 3 を PC の USB ポートに接続し、MPLAB IDE と通信します。以下のメッセージが表示され、いずれかを選択します。

図 6-3: PROGRAMMER-TO-GO モードの終了



[Yes] をクリックするとイメージが保持され、[No] をクリックすると消去されます。[No] をクリックした場合は [Output] ウィンドウに図 6-4 のようなメッセージが表示され、Programmer-To-Go モードが終了します。

図 6-4: [OUTPUT] ウィンドウに表示される終了時のメッセージ





第 2 部 – トラブルシュート

Chapter 7. トラブルシュート.....	43
Chapter 8. よく寄せられる質問 (FAQ).....	45
Chapter 9. エラーメッセージ.....	49

PICKit™ 3 ユーザガイド

NOTES:

Chapter 7. トラブルシュート

7.1 はじめに

PICKit 3 プログラマ / デバッガの操作で困った事がある場合、まずこちらの内容を確認してください。

- 最初に確認する 5 つの項目
- デバッグに失敗する主な理由
- その他の確認事項

7.2 最初に確認する 5 つの項目

1. 使用デバイスを確認してください。比較的新しいデバイスの場合、MPLAB IDE のバージョンアップが必要となる事があります。旧バージョンの MPLAB IDE では動作確認のとれていないデバイスがあるため、注意してください。
2. マイクロチップ社のデモボードを使っていますか。それとも自作のボードを使っていますか。通信の接続に必要な抵抗とコンデンサに関するガイドラインが守られていますか (**Chapter 2.「動作原理」**参照)。
3. ターゲットに電源が供給されていますか。30 mA を超える場合、デバッガからターゲットに電源を供給できません。
4. USB ハブを使っていますか。ハブに電源は供給されていますか。それでも問題が解決しない場合、ハブを使用せずデバッガを PC に直接接続してください。
5. デバッガに付属の標準通信ケーブル (RJ-11) を使っていますか。これよりも長いケーブルを使った場合、通信エラーが起こる可能性があります。

7.3 デバッグに失敗する主な理由

1. オシレータが動作していない。オシレータに関するコンフィグレーション ビットの設定を確認してください。
2. ターゲットボードに電源が供給されていない。電源ケーブルの接続を確認してください。
3. デバッガと PC またはターゲットボードが物理的に接続されていない。通信ケーブルの接続を確認してください。
4. デバイスのコード保護が有効になっている。コード保護に関するコンフィグレーション ビットの設定を確認してください。
5. Release モードでプロジェクトの再ビルドを行おうとしている。プロジェクト マネージャ ツールバーの [Build Configuration] ドロップダウン リストで **[Debug]** を選択してからプロジェクトを再ビルドしてください。
6. MPLAB IDE で PICKit 3 がデバッガではなくプログラマとして選択されている。
7. デバッガと PC の通信が中断された。MPLAB IDE でデバッガに再接続します。
8. ターゲット アプリケーションに破損またはエラーがある。ターゲット アプリケーションを再ビルドしてプログラミングし直してください。その後で、ターゲットでパワーオンリセットを実行します。
9. 他のコンフィグレーション設定が干渉してデバッグが行えない。ターゲットがコードを実行できないようなコンフィグレーション設定が行われていると、デバッガはコードをデバッグモードにする事ができません。
10. デバッガは、要求された動作を常に実行できるわけではありません。例えばターゲット アプリケーションが実行中の場合、デバッガはブレークポイントを設定できません。

7.4 その他の確認事項

1. 一過性のエラーの場合もあります。同じ操作を繰り返してみてください。
2. プログラミングそのものに問題がある事も考えられます。プログラマモードに切り換えて、なるべくシンプルなアプリケーション (LED 点滅プログラム等) でターゲットへの書き込みテストを行ってください。このプログラムが動作しない場合、ターゲットのセットアップに問題がある事になります。
3. ターゲット デバイスが何らかの損傷 (過電流等) を受けた可能性があります。開発環境では、電子部品に悪影響が及ぶ事がよくあります。ターゲット デバイスを別のものに交換してみてください。
4. マイクロチップ社では、ほとんどのマイクロコントローラをサポートしたデモボードを用意しています。PICkit 3 プログラマ / デバッガが正しく動作するかどうかを検証するには、正常動作が確認済みのこれらのアプリケーションを使用するのも 1 つの方法です。
5. デバッガの動作原理を確認して、アプリケーションを正しくセットアップしてください (Chapter 2.「動作原理」参照)。
6. それでも解決しない場合はマイクロチップ社までお問い合わせください。

Chapter 8. よく寄せられる質問 (FAQ)

8.1 はじめに

ここでは、PICKit 3 プログラマ / デバッガシステムについてよく寄せられる質問とその回答を紹介します。

- 動作に関する FAQ
- 不具合に関する FAQ

8.2 動作に関する FAQ

- PICKit 3 プログラマ / デバッガは通信にデバイス内のどの機能を利用しているのですか。
PICKit 3 プログラマ / デバッガは、ICSP インターフェイスを介してフラッシュデバイスと通信します。そして、プログラムメモリまたはテストメモリにダウンロードしたデバッガ実行プログラムを使います。
- デバッガ実行プログラムを動作させるとプロセッサのスループットにどのような影響がありますか。
デバッガ実行プログラムは実行モードでは動作しないため、コード実行時にスループットの低下はありません。つまり、デバッガがターゲット デバイスの実行サイクルを「奪う」事はありません。
- PICKit 3 プログラマ / デバッガと他のインサーキット エミュレータ / デバッガにはどのような違いがありますか。
2.2 「PICKit 3 と PICKit 2 の違い」を参照してください。
- MPLAB ICE 2000/4000 デバッガでは、データに対して複雑なトリガを実行するにはそのデータをバスに転送する必要があります。PICKit 3 プログラマ / デバッガでもその必要がありますか。例えば、フラグが High になった時に停止する事はできますか。
MPLAB ICE 2000/4000 デバッガは、専用のデバッガチップ (-ME) を使って監視を行います。PICKit 3 プログラマ / デバッガには -ME チップはないため、外部からバスを監視する事はありません。PICKit 3 プログラマ / デバッガでは、外部ブレークポイントを使用するのではなく、デバッグエンジンに内蔵されたブレークポイント回路を使用します。つまりバスとブレークポイント ロジックはデバイス内部で監視します。
- PICKit 3 プログラマ / デバッガでは MPLAB ICE 2000/4000 のように複合ブレークポイントを利用できますか。
いいえ。しかし、特定のデータメモリ アドレスに格納された値またはプログラムアドレスに基づいてブレークを実行する事はできます。詳細は、10.3.1 「[Breakpoints] ダイアログ」を参照してください。
- PICKit 3 は光学的または電氣的に絶縁されていますか。
いいえ。現在のシステムにフロート電圧または高電圧 (120V) を印加する事はできません。

- **標準ケーブルには何か制約はありますか。**
標準の ICSP RJ-11 ケーブルは、15 Mbps を超えるクロック速度には対応していません。
- **PICkit 3 によってプログラムの実行速度は低下しますか。**
いいえ。デバイスはデータシートの仕様通りの速度で動作します。
- **dsPIC DSC を任意の速度で動作させてデバッグを行う事はできますか。**
PICkit 3 は、デバイスのデータシートに記載されている任意の速度でデバイスを動作させてデバッグを実行できます。
- **6 ピン (LVP ピン) はどのような働きをしますか。**
6 ピンは、LVP (Low-Voltage Programming) 接続用に予約されています。

8.3 不具合に関する FAQ

- **PC が省電力 / 休止モードになった後、デバッガが動作しなくなりました。どうしたのでしょうか。**
デバッガを (特にデバッガとして) 長時間使う場合、お使いの PC のオペレーティングシステムの電源オプション設定画面で休止モードを無効にしておいてください。Windows XP の場合、[電源オプションのプロパティ] ダイアログ ボックスの [休止状態] タブで [休止状態を有効にする] のチェックを外します。こうすると、全ての USB サブシステム コンポーネントで全通信を維持できます。
- **周辺モジュールに [Freeze on Halt] を設定していないのに突然フリーズしてしまいます。なぜでしょうか。**
dsPIC30F/33F と PIC24F/H の場合、デバッガは周辺制御レジスタの予約ビット (通常は bit 14 または 5) を Freeze ビットとして使用します。レジスタ全体に書き込みを実行した時に、このビットが上書きされた可能性があります (このビットはデバッグモードではユーザアクセス可能です)。
この問題を防ぐには、レジスタ全体を書き換える命令 (MOV) ではなく、アプリケーションで変更が必要なビットだけを書き換える命令 (BTS、BTC) を使用します。
- **16 ビットデバイスを使用中に、予期しないリセットが発生しました。どのようにして原因を調べれば良いでしょうか。**
以下の点を確認してください。
 - RCON レジスタを確認してリセット要因を調べる。
 - 割り込みサービスルーチン (ISR) でトラップ / 割り込みを処理する。例えば、下記のような trap.c コードを挿入する。

```
void __attribute__((__interrupt__)) _OscillatorFail(void);
:
void __attribute__((__interrupt__)) _AltOscillatorFail(void);
:
void __attribute__((__interrupt__)) _OscillatorFail(void)
{
    INTCON1bits.OSCFAIL = 0;           //Clear the trap flag
    while (1);
}
:
void __attribute__((__interrupt__)) _AltOscillatorFail(void)
{
    INTCON1bits.OSCFAIL = 0;
    while (1);
}
:
```
 - アサートを使用する。

- コードのデバッグが完了しました。デバイスをプログラムしましたが、動作しません。何が原因でしょうか。

以下の点を確認してください。

- デバッガをプログラマとして選択してヘッダボードへの書き込みを行いませんでしたか。ヘッダボードに搭載されているのは -ICE/-ICD バージョンのデバイスで、実際のデバイスとは動作が異なる事があります。デバッガをプログラマとして使って書き込めるのは、通常のデバイスのみです。通常のデバイスにも ICE/ICD 回路を内蔵したデバイスがありますが、ヘッダボードに搭載されている特殊な -ICE/-ICD デバイスとは別のものです。
 - PICkit 3 をデバッガとして選択したままデバイスにプログラミングを行いませんでしたか。PICkit 3 をデバッガとして選択してプログラミングを行うと、デバッガ実行プログラムがプログラムメモリに書き込まれ、デバッグ用のその他のデバイス機能がセットアップされます (2.6.1「デバッグ開始までの操作手順」を参照)。完成した (リリース用) コードをプログラムする場合、PICkit 3 をプログラマに設定する必要があります。
 - [Build Configuration] ドロップダウン リストまたは [Project] メニューで「Release」を選択しましたか。完成した (リリース用) コードを書き込むには、この操作が必要です。プロジェクトを再ビルドしてデバイスにもう一度書き込みを行い、コードが動作するかどうか試してみてください。
- この FAQ に記載されていない問題が発生しました。どうすれば良いでしょうか。以下のセクションを参照してください。
- 2.8「デバッガが使うリソース」
 - 9.2「特定の状況で表示されるエラーメッセージ」
 - 9.3「一般的な対処方法」

NOTES:

Chapter 9. エラーメッセージ

9.1 はじめに

PICKit 3 プログラマ / デバッガで表示されるエラーメッセージには多くの種類があります。特定の状況で表示されるエラーメッセージもありますが、一般的な対処法で解決するものもあります。

- 特定の状況で表示されるエラーメッセージ
- 一般的な対処方法

9.2 特定の状況で表示されるエラーメッセージ

以下に、PICKit 3 プログラマ / デバッガのエラーメッセージをエラーコードの番号順に示します。

Note: 現時点では、エラーメッセージにエラーコードが表示されない場合があります。ヘルプの [Search] タブでメッセージを検索して、強調表示された項目を参照してください。

以下のエラーメッセージで「%x」(変数)と表記された部分は、実際のエラーメッセージでは状況に応じた文字列が表示されます。

PK3Err0001: Failed while writing to program memory. (プログラムメモリへの書き込みに失敗しました。)

PK3Err0002: Failed while writing to EEPROM. (EEPROM への書き込みに失敗しました。)

PK3Err0003: Failed while writing to configuration memory. (コンフィギュレーションメモリへの書き込みに失敗しました。)

9.3.1「読み書きエラーの対処方法」を参照してください。

PK3Err0005: PICKit 3 is currently busy and cannot be unloaded at this time.

(PICKit 3 は現在ビジーで選択を解除できません。)

PICKit 3 をデバッガまたはプログラマとしての指定から解除しようとしてこのエラーが表示された場合、以下の解決方法を試してください。

1. デバッガが現在実行中のタスクが終了するまで待つ。その後、もう一度デバッガの選択解除を試みる。
2. [Halt] を選択して実行中のアプリケーションを停止する。その後、もう一度デバッガの選択解除を試みる。
3. デバッガとPCの接続ケーブルを抜く。その後、もう一度デバッガの選択解除を試みる。
4. MPLAB IDE を終了する。

PK3Err0006: Failed while writing to user ID memory. (ユーザ ID メモリへの書き込みに失敗しました。)

PK3Err0007: Failed while reading program memory. (プログラムメモリの読み出しに失敗しました。)

PK3Err0008: Failed while reading EEPROM. (EEPROM の読み出しに失敗しました。)

PK3Err0009: Failed while reading configuration memory. (コンフィギュレーションメモリの読み出しに失敗しました。)

PK3Err0010: Failed while reading user ID memory. (ユーザ ID メモリの読み出しに失敗しました。)

9.3.1「読み書きエラーの対処方法」を参照してください。

PK3Err0011: Bulk erase failed. (バルク消去に失敗しました。)

9.3.1「読み書きエラーの対処方法」を参照してください。

それでも解決しない場合、別のデバイスで試してください。

PK3Err0012: Download debug exec failed (デバッグ実行プログラムのダウンロードに失敗しました。)

[Debugger] メニューから書き込みを実行しようとしてこのエラーが表示された場合、以下の解決方法を試してください。

1. PICkit 3 をデバッグとしての指定から解除する。
2. プロジェクトを閉じて MPLAB IDE を終了する。
3. MPLAB IDE を再起動してもう一度プロジェクトを開く。
4. もう一度 PICkit 3 をデバッグとして指定し、ターゲット デバイスへの書き込みを実行する。

それでも解決しない場合、9.3.4「インストール ファイルが破損している場合の対処方法」を参照してください。

PK3Err0013: NMMR register write failed. (NMMR レジスタへの書き込みに失敗しました。)

PK3Err0014: File register write failed. (ファイルレジスタへの書き込みに失敗しました。)

9.3.2「デバッグとターゲットの通信エラーの対処方法」を参照してください。

PK3Err0015: Data transfer was unsuccessful. %d byte(s) expected, %d byte(s) transferred. (データ転送に失敗しました。%d バイト中、%d バイト転送されました。)

9.3.3「デバッグと PC の通信エラーの対処方法」を参照してください。

PK3Err0016: Cannot transmit. PICkit 3 not found. (転送できません。PICkit 3 が見つかりません。)

デバッグが PC に正しく接続されていません。

PK3Err0017: File register read failed. (ファイルレジスタの読み出しに失敗しました。)

PK3Err0018: NMMR register read failed. (NMMR レジスタの読み出しに失敗しました。)

PK3Err0019: Failed while reading emulation registers. (エミュレーション レジスタの読み出しに失敗しました。)

PK3Err0020: Failed while writing emulation registers. (エミュレーション レジスタの書き込みに失敗しました。)

9.3.2「デバッグとターゲットの通信エラーの対処方法」を参照してください。

PK3Err0021: Command not echoed properly. Sent %x, received %x. (コマンドのエコーが正しくありません。送信 %x、受信 %x)

PK3Err0022: Failed to get PICkit 3 version information. (PICkit 3 のバージョン情報の取得に失敗しました。)

PK3Err0024: Download RS failed. (RS のダウンロードに失敗しました。)

PK3Err0025: Download AP failed. (AP のダウンロードに失敗しました。)

9.3.3「デバッグと PC の通信エラーの対処方法」を参照してください。

PK3Err0026: Download program exec failed. (EXEC プログラムのダウンロードに失敗しました。)

[Debugger] メニューから書き込みを実行しようとしてこのエラーが表示された場合、以下の解決方法を試してください。

1. PICkit 3 をデバッグとしての指定から解除する。
2. プロジェクトを閉じて MPLAB IDE を終了する。
3. MPLAB IDE を再起動してもう一度プロジェクトを開く。
4. もう一度 PICkit 3 をデバッグとして指定し、ターゲット デバイスへの書き込みを実行する。

それでも解決しない場合、9.3.4「インストール ファイルが破損している場合の対処方法」を参照してください。

PK3Err0027: Bulk transfer failed due to invalid checksum. (チェックサム エラーによりバルク転送に失敗しました。)

9.3.3「デバッグと PC の通信エラーの対処方法」を参照してください。

また、使っているケーブルの長さが適切である事を確認してください。

PK3Err0028: Download device database failed. (デバイス データベースのダウンロードに失敗しました。)

このエラーが表示された場合、以下の解決方法を試してください。

1. もう一度ダウンロードする。(一過性のエラーの場合があるため。)
2. 手動でダウンロードする。[Debugger]>[Settings]の順に選択して、[Configuration] タブを選択し、[Manual Download] をクリックする。最も数の大きい .jam ファイルを選択して [Open] をクリックする。

PK3Err0029: Communication failure. Unexpected command echo response %x received from ICD 3. (通信エラー。PICkit 3 から予期しないコマンドエコー %x が返されました。)

9.3.3「デバッガと PC の通信エラーの対処方法」を参照してください。

PK3Err0030: Unable to read/find firmware File %s. (ファームウェア ファイル %s が読み出せないか、見つかりません。)

Hex ファイルが存在する場合、以下の解決方法を試してください。

- 再接続してもう一度試してみる。
- それでも解決しない場合、ファイルが破損している場合があります。MPLAB IDE を再インストールする。

Hex ファイルが存在しない場合、以下の解決方法を試してください。

- MPLAB IDE を再インストールする。

PK3Err0031: Failed to get PC. (PC の取得に失敗しました。)

PK3Err0032: Failed to set PC. (PC の設定に失敗しました。)

9.3.2「デバッガとターゲットの通信エラーの対処方法」を参照してください。

PK3Err0033: %d bytes expected, %d bytes received. (%d バイト中、%d バイト受信しました。)

9.3.3「デバッガと PC の通信エラーの対処方法」を参照してください。

PK3Err0034: This version of MPLAB IDE does not support hardware revision %06x. Please upgrade to the latest version of MPLAB IDE before continuing.

(このバージョンの MPLAB IDE はハードウェア リビジョン %06x をサポートしていません。最新バージョンの MPLAB IDE にアップグレードしてから作業を続けてください。)

最新の MPLAB IDE を www.microchip.com から入手してください。

PK3Err0035: Failed to get Device ID. (デバイス ID の取得に失敗しました。)

9.3.1「読み書きエラーの対処方法」を参照してください。

PK3Err0036: MPLAB IDE has lost communication with PICkit 3. (MPLAB IDE と PICkit 3 の通信が切断されました。)

9.3.3「デバッガと PC の通信エラーの対処方法」を参照してください。

PK3Err0037: Timed out waiting for response from PICkit 3. (PICkit 3 からの応答待機中にタイムアウトになりました。)

PK3Err0038: Failed to initialize PICkit 3. (PICkit 3 の初期化に失敗しました。)

PK3Err0039: PICkit 3 self-test failed. (PICkit 3 のセルフテストに失敗しました。)

このエラーは、デバッガが応答していない場合に表示されます。

1. デバッガと PC の接続ケーブルを一旦抜いて挿し直す。
2. MPLAB IDE でデバッガに再接続する。
3. それでも解決しない場合はマイクロチップ社までお問い合わせください。

PK3Err0040: The target device is not ready for debugging. Please check your Configuration bit settings and program the device before proceeding.

(ターゲット デバイス側でデバッグの準備ができていません。コンフィグレーション ビットの設定を確認してデバイスをもう一度プログラムしてから作業を続けてください。)

このメッセージが表示されるのは、デバイスをプログラムせずに実行しようとした場合です。=このような場合、またはデバイスのプログラミング直後にこのメッセージが表示された場合、9.3.6「デバッグエラーの対処方法」を参照してください。

PK3Err0043: RS download failed. (RS のダウンロードに失敗しました。)

PK3Err0044: AP download failed. (AP のダウンロードに失敗しました。)

PK3Err0045: You must connect to a target device to use PICkit 3.

(PICkit 3 を使用するには、ターゲット デバイスに接続する必要があります。)

電源が未検出です。

1. デバッガとターゲットの間で VDD と GND が接続されている事を確認する。
2. ターゲットに電源が供給されている事を確認する。
3. デバッガが検出できる十分な電力がターゲットに供給されている事を確認する
(補遺 A.「ハードウェア仕様」参照)。

PK3Err0046: An error occurred while trying to read the stopwatch count. The stopwatch count may not be accurate.

(ストップウォッチのカウンtr読み出し中にエラーが発生しました。ストップウォッチのカウンtr値が正確でない可能性があります。)

9.3.2「デバッガとターゲットの通信エラーの対処方法」を参照してください。

PK3Err0047: Bootloader download failed. (ブートローダのダウンロードに失敗しました。)

9.3.3「デバッガと PC の通信エラーの対処方法」を参照してください。

PK3Err0052: The current PICkit 3 hardware version %x, is out of date. This version of MPLAB IDE will support only version %x or higher.

(現在の PICkit 3 のハードウェアバージョン %x は旧バージョンです。お使いの MPLAB IDE はバージョン %x 以上のみサポートします。)

最新ファームウェアのダウンロードを確認するダイアログで [Cancel] をクリックしませんでしたか。もしそうなら、最新ファームウェアをすぐにダウンロードしてください。[Debugger]>[Settings] の順に選択して、[Configuration] タブを選択し、[Manual Download] をクリックします。最も大きい数字の .jam ファイルを選択して [Open] をクリックします。

ダウンロードするファイルが見つからない場合や、上記手順でも解決しない場合 (ファイルが破損している可能性があります) は、最新バージョンの MPLAB IDE を入手してインストールしてください。最新の MPLAB IDE を www.microchip.com から入手してください。

PK3Err0053: Unable to get PICkit 3 protocol versions. (PICkit 3 のプロトコルバージョンの取得に失敗しました。)

9.3.3「デバッガと PC の通信エラーの対処方法」を参照してください。

PK3Err0054: MPLAB IDE's PICkit 3 protocol definitions are out of date. You must upgrade MPLAB IDE to continue. (MPLAB IDE の PICkit 3 プロトコル定義は旧バージョンです。MPLAB IDE をアップグレードしてから作業を続けてください。)

最新の MPLAB IDE を www.microchip.com から入手してください。

PK3Err0055: Unable to set firmware suite version. (ファームウェアスイートのバージョンを設定できません。)

PK3Err0056: Unable to get voltages from PICkit 3. (PICkit 3 から電圧を取得できません。)

9.3.3「デバッガと PC の通信エラーの対処方法」を参照してください。

PK3Err0068: Failed while writing to boot Flash memory. (ブートフラッシュメモリへの書き込みに失敗しました。)

PK3Err0069: Failed while reading boot Flash memory. (ブートフラッシュメモリの読み出しに失敗しました。)

PK3Err0070: Failed while writing peripheral memory. (周辺メモリへの書き込みに失敗しました。)

PK3Err0071: Failed while reading peripheral memory. (周辺メモリの読み出しに失敗しました。)

9.3.1「読み書きエラーの対処方法」を参照してください。

PK3Err0072: Unable to send freeze peripheral information. (周辺モジュールのフリーズ情報を送信できません。)

PK3Err0073: Device is code-protected. (デバイスがコード保護されています。)

操作 (読み出し、書き込み、ブランクチェック、ベリファイ) を行おうとしたデバイスのコード保護が有効になっており、コードの読み出しまたは変更が行えません。コード保護に関するコンフィグレーションビットの設定を確認してください。

コード保護を無効にするには、デバイスのデータシートを参照して該当するコンフィグレーションビットをコード内または [Configuration Bits] ウィンドウ ([Configure]>[Configuration Bits]) でセットまたはクリアします。その後で、デバイス全体を消去してから再プログラムします。

PK3Err0075: Unable to set power. (電源をセットできません。)

PICkit 3 からターゲットに電源を供給できません。

PK3Err0080: Failed setting shadow bit(s). (シャドウビットの設定に失敗しました。)

9.3 一般的な対処方法

以下、問題が発生した場合の一般的な対処方法を紹介します。

- 読み書きエラーの対処方法
- デバッガとターゲットの通信エラーの対処方法
- デバッガと PC の通信エラーの対処方法
- インストール ファイルが破損している場合の対処方法
- USB ポート通信エラーの対処方法
- デバッグエラーの対処方法
- 内部エラーの対処方法

9.3.1 読み書きエラーの対処方法

読み書きエラーが発生した場合、以下の事項を確認してください。

1. [Abort] をクリックしたかどうか (この操作で読み書きエラーが発生する事があるため)。
2. 同じ操作を繰り返してみる (一過性のエラーの場合があるため)。
3. ターゲットに電源が供給されており、デバイスの電圧レベルが適正である事を確認する (デバイスの適正電圧は、各デバイスのデータシートを参照してください。)
4. デバッガとターゲットの接続が正しい事 (PGC と PGD が接続されている事) を確認する。
5. 書き込みエラーの場合、[Settings] ダイアログの [Program Memory] タブで [Erase all before Program] にチェックが入っている事を確認する。
6. 使っているケーブルの長さが適切である事を確認する。

9.3.2 デバッガとターゲットの通信エラーの対処方法

PICkit 3 プログラマ / デバッガとターゲット デバイスの同期がとれていません。

1. [Reset] を選択してから同じ操作を再試行する。
2. 使っているケーブルの長さが適切である事を確認する。

9.3.3 デバッガと PC の通信エラーの対処方法

PICkit 3 プログラマ / デバッガと MPLAB IDE の同期がとれていません。

1. デバッガと PC の接続ケーブルを一旦抜いて挿し直す。
2. デバッガに再接続する。
3. 同じ操作を繰り返してみる (一過性のエラーの場合があるため)。
4. インストールされている MPLAB IDE のバージョンが、PICkit 3 プログラマ / デバッガに読み込まれているファームウェアのバージョンに対応していない可能性がある。9.3.4 「インストール ファイルが破損している場合の対処方法」の手順に従う。

9.3.4 インストール ファイルが破損している場合の対処方法

この問題は、ほとんどの場合、MPLAB IDE のインストールが不完全であるか、インストールしたファイルが破損した事が原因で発生します。

1. 全てのバージョンの MPLAB IDE を PC からアンインストールする。
2. 使用するバージョンの MPLAB IDE を再インストールする。
3. それでも解決しない場合はマイクロチップ社までお問い合わせください。

9.3.5 USB ポート通信エラーの対処方法

この問題は、ほとんどの場合、通信ポートに問題があるか、存在しない通信ポートを指定している事が原因で発生します。

1. PICkit 3 プログラマ / デバッガに再接続する。
2. デバッガが物理的に PC の適切な USB ポートに接続されている事を確認する。
3. デバッガの [Settings] で適切な USB ポートが選択されている事を確認する。
4. 指定した USB ポートを他のデバイスが使っていない事を確認する。
5. USB ハブを使う場合、電源が供給されている事を確認する。
6. USB ドライバが読み込まれている事を確認する。

9.3.6 デバッグエラーの対処方法

PICkit 3 プログラマ / デバッガでデバッグを実行できない場合、これには多くの原因が考えられます。7.3「デバッグに失敗する主な理由」と 7.4「その他の確認事項」を参照してください。

9.3.7 内部エラーの対処方法

内部エラーは予測できず、通常は発生すべきものではありません。これらは主にマイクロチップ社内の開発で使われます。

主な原因として、インストール ファイルの破損が考えられます (9.3.4「インストール ファイルが破損している場合の対処方法」参照)。

また、システム リソースの減少が原因の事もあります。

1. システムを再起動してメモリを解放する。
2. ハードディスク ドライブに十分な空き容量がある事、過度なフラグメンテーションが発生していない事を確認する。

それでも解決しない場合はマイクロチップ社までお問い合わせください。



第 3 部 – リファレンス

Chapter 10. デバッガ機能一覧.....	57
補遺 A. ハードウェア仕様.....	69
補遺 B. PICKit 3 の回路図.....	75

NOTES:

Chapter 10. デバッグ機能一覧

10.1 はじめに

ここでは、MPLAB IDE のメニュー、ウィンドウ、ダイアログで表示される PICKit 3 プログラマ / デバッグの機能をまとめます。

- デバッグに関する機能
- デバッグに関するダイアログ / ウィンドウ
- プログラミングに関する機能
- [Settings] ダイアログ

10.2 デバッグに関する機能

[Debugger] メニューで PICKit 3 を選択すると、MPLAB IDE の機能に以下の項目が追加されます。

- [Debugger] メニュー — ドロップダウン メニューに項目が追加されます。
- デバッグに関する右クリック メニュー — コンテキスト メニューに項目が追加されます。
- ツールバー / ステータス バー — メニューバーの下にツールバーが表示されます。ステータスバーにも追加の情報が表示されます。

10.2.1 [Debugger] メニュー

[Run] (F9 キー)

ブレークポイントまで、または [Halt] を選択するまでプログラムコードを実行します。コード実行は、現在のプログラム カウンタ (ステータスバーに表示) から開始します。現在のプログラム カウンタ位置は、[Program Memory] ウィンドウのポインタでも確認できます。プログラム実行中は、一部の機能が無効になります。

[Animate]

[Animate] を実行すると、デバッグが 1 ステップずつ実行し、画面に表示されるレジスタの値もそれに合わせて更新されます。

Animate は Run よりも実行速度が遅い一方で、[Special Function Register] ウィンドウまたは [Watch] ウィンドウでレジスタの値の変化を確認できます。

Animate を停止するには、メニューで [Debugger]>[Halt] の順にクリックするか、ツールバーの [Halt] をクリックするか、<F5> キーを押します。

[Halt] (F5 キー)

[Halt] を実行すると、プログラムコードの実行が停止します。[Halt] をクリックすると、ステータス情報が更新されます。

[Step Into] (F7 キー)

プログラムコードを 1 ステップずつ実行します。

アセンブリコードの場合、1 命令 (1 サイクル命令または複数サイクル命令) を実行して停止します。1 命令を実行するたびに、全てのウィンドウが更新されます。

Cコードの場合、コードを1行(アセンブリ命令レベルでは複数の命令に相当する事もあります)ずつ実行して停止します。実行が完了するたびに、全てのウィンドウが更新されます。

Note: [Step Into] では SLEEP 命令を実行しないでください。

[Step Over] (F8 キー)

現在のプログラムカウンタ位置の命令を実行します。CALL 命令があると、呼び出されたサブルーチンを実行後、CALL 命令の次のアドレスで停止します。Step Over が長時間終了しない場合、またはハングしたと思われる場合、[Halt] をクリックします。

[Step Out]

利用できません。

[Reset]>[Processor Reset] (F6 キー)

ターゲットプロセッサにリセットシーケンスを送信します。MCLR が発行され、プログラムカウンタはリセットベクタに戻ります。

[Breakpoints]

[Breakpoints] ダイアログを開きます (10.3.1 「[Breakpoints] ダイアログ」参照)。ハードウェアブレークポイントを使うと、コードの指定行でコード実行を停止できます。このダイアログで複数のブレークポイントを設定します。

Note: コード行を右クリックまたはダブルクリックすると、その行に単純ブレークポイントを設定できます。

[Program]

コードをターゲットデバイスにダウンロードします。

[Read]

ターゲットのメモリを読み出します。情報は MPLAB IDE にアップロードされます。

[Erase Flash Device]

フラッシュメモリ全体を消去します。

[Abort Operation]

プログラミング操作(プログラム、読み出し等)を中止します。操作を中止すると、デバイスの状態は未知となります。

[Reconnect]

PC と PICkit 3 プログラマ / デバッガの通信再開を試みます。接続の進行状況は、[Output] ダイアログの [PICkit 3] タブに表示されます。

[Settings]

PICkit 3 の [Settings] ダイアログを開きます (10.5 「[Settings] ダイアログ」参照)。プログラムとファームウェアのオプションを設定します。

10.2.2 デバッグに関する右クリック メニュー

プログラムメモリやソースコード ファイル等のコード表示画面を右クリックすると、デバッグに関する以下のメニューオプションが表示されます。ここに記載されていないメニューオプションは、MPLAB IDE ヘルプまたは MPLAB エディタのヘルプを参照してください。

[Set Breakpoint]

現在選択している行にブレークポイントを設定 (または解除) します。

[Breakpoints]

全てのブレークポイントを一括して削除、有効化、無効化します。

[Run To Cursor]

現在のカーソル位置までプログラムを実行します。以前の [Run to Here] と同じです。

[Set PC at Cursor]

現在のカーソル位置にプログラム カウンタ (PC) をセットします。

[Center Debug Location]

現在 PC がセットされている行をウィンドウの中央に表示します。

10.2.3 ツールバー / ステータス バー

PICkit 3 プログラマ / デバッグをデバッグとして指定すると、MPLAB IDE に以下のツールバーが表示されます。

- 基本的なデバッグ ツールバー ([Run]、[Halt]、[Animate]、[Step Into]、[Step Over]、[Step Out]、[Reset]、[Breakpoints])
- シンプルなプログラム ツールバー ([Program]、[Read]、[Erase Flash Device])

MPLAB IDE デスクトップの一番下にあるステータスバーには、現在選択されているデバッグ (PICkit 3) とその他の開発に関する情報が表示されます。ステータスバーの詳細は、MPLAB IDE のオンラインヘルプを参照してください。

10.3 デバッグに関するダイアログ / ウィンドウ

10.2「デバッグに関する機能」に記載したメニュー項目をクリックすると、デバッグに関する以下のダイアログまたはウィンドウが開きます。

- [Breakpoints] ダイアログ
 - [Set Breakpoint] ダイアログ
 - [Stopwatch] ダイアログ
 - [Event Breakpoints] ダイアログ

10.3.1 [Breakpoints] ダイアログ

ブレークポイントを設定するには、**[Debugger]>[Breakpoints]** の順にクリックします。

このダイアログでは、各種ブレークポイントを設定できます。**[Add Breakpoint]** をクリックすると、ブレークポイントの一覧に新規ブレークポイントを追加できます。選択したデバイスによっては、より高度なブレークポイント機能に関するボタンが表示される事があります。

10.3.1.1 [BREAKPOINTS] ダイアログ ウィンドウ

このウィンドウには、各ブレークポイントに関する情報が表示されます。

表 10-1: [BREAKPOINTS] ダイアログ ウィンドウ

コントロール	機能
[Breakpoint Type]	ブレークポイントの種類 (プログラムまたはデータ)
[Address]	ブレークポイント位置のアドレス (16 進数)
[File Line #/Symbol Name]	ファイル名とブレークポイント位置の行番号
[Enabled]	チェックしてブレークポイントを有効化

このウィンドウにブレークポイントを追加した後、ブレークポイントを右クリックすると、以下のメニューオプションが表示されます。

- [Delete] — 選択したブレークポイントを削除します。
- [Edit/View] — [Set Breakpoint] ダイアログを開きます。
- [Delete All] — 一覧に表示されたブレークポイントを全て削除します。
- [Disable All] — 一覧に表示されたブレークポイントを全て無効にします。

10.3.1.2 [BREAKPOINTS] ダイアログのボタン

これらのボタンを使ってブレークポイントを追加したり、ブレーク条件を詳しく設定したりできます。また、ブレークポイントとトリガにストップウォッチを組み合わせる事もできます。

Note: 表示されるボタンは、選択したデバイスによって決まります。

表 10-2: [BREAKPOINTS] ダイアログのボタン

コントロール	機能	関連するダイアログ
[Add Breakpoint]	ブレークポイントの追加	10.3.2 「[Set Breakpoint] ダイアログ」
[Stopwatch]	ストップウォッチの設定	10.3.3 「[Stopwatch] ダイアログ」
[Event Breakpoints]	イベント ブレークポイントの設定	10.3.4 「[Event Breakpoints] ダイアログ」

10.3.2 [Set Breakpoint] ダイアログ

[Breakpoints] ダイアログで **[Add Breakpoint]** をクリックすると、このダイアログが表示されます。

[Breakpoints] ダイアログに追加するブレークポイントをここで設定します。

10.3.2.1 [PROGRAM MEMORY] タブ

プログラムメモリのブレークポイントをここで設定します。

表 10-3: プログラムメモリのブレークポイント

コントロール	機能
[Address]	ブレークポイントのアドレス (16 進数)
[Breakpoint Type]	プログラムメモリ ブレークポイントの種類です。テーブル読み書きの詳細は、各デバイスのデータシートを参照してください。 [Program Memory Execution]: 上記アドレスの実行でブレークします。 [TBLRD Program Memory]: 上記アドレスに対するテーブル読み出しでブレークします。 [TBLWT Program Memory]: 上記アドレスに対するテーブル書き込みでブレークします。
[Pass Count]	ブレーク条件にパスカウントを設定します。 [Always break]: [Breakpoint type] で指定したイベントで常にブレークします。 [Break occurs Count instructions after Event]: [Breakpoint type] で指定したイベントの後、[Count] で指定した数 (0 ~ 255) の命令を実行してからブレークします。 [Event must occur Count times]: [Breakpoint type] で指定したイベントが [Count] で指定した回数 (0 ~ 255) 発生するとブレークします。

10.3.2.2 [DATA MEMORY] タブ

データメモリのブレークポイントをここで設定します。

表 10-4: データメモリのブレークポイント

コントロール	機能
[Address]	ブレークポイントのアドレス (16 進数)
[Breakpoint Type]	データメモリ ブレークポイントの種類です。Xバス読み書きの詳細は、各デバイスのデータシートを参照してください。 [X Bus Read]: 上記アドレスに対する Xバス読み出しでブレークします。 [X Bus Read Specific Byte]: 上記アドレスの Xバス読み出しの値が [Specific Value] で指定したバイト値ならブレークします。 [X Bus Read Specific Word]: 上記アドレスの Xバス読み出しの値が [Specific Value] で指定したワード値ならブレークします。 [X Bus Write]: 上記アドレスに対する Xバス書き込みでブレークします。 [X Bus Write Specific Byte]: 上記アドレスの Xバス書き込みの値が [Specific Value] で指定したバイト値ならブレークします。 [X Bus Write Specific Word]: 上記アドレスの Xバス書き込みの値が [Specific Value] で指定したワード値ならブレークします。
[Pass Count]	ブレーク条件にパスカウントを設定します。 [Always break]: [Breakpoint type] で指定したイベントで常にブレークします。 [Break occurs Count instructions after Event]: [Breakpoint type] で指定したイベントの後、[Count] で指定した数 (0 ~ 255) の命令を実行してからブレークします。 [Event must occur Count times]: [Breakpoint type] で指定したイベントが [Count] で指定した回数 (0 ~ 255) 発生するとブレークします。

10.3.3 [Stopwatch] ダイアログ

[Breakpoints] ダイアログで [Stopwatch] をクリックすると、ストップウォッチ設定ダイアログが表示されます。

ブレークポイントとストップウォッチを併用すると、コード実行の時間を計測できます。ストップウォッチを使うと、2つのブレークポイント/トリガ条件の間の経過時間を計測できます。ストップウォッチの値は 10 進数です。

表 10-5: ストップウォッチの設定

コントロール	機能
[Start Condition]	ストップウォッチの開始条件となるブレークポイントまたはトリガを選択します。あらかじめ [Breakpoints] ダイアログに追加済みのブレークポイント/トリガの中から選択できます。開始条件をクリアするには、[None] を選択します。 [Start condition will cause the target device to halt] にチェックを入れると、この条件でプログラム実行が停止します。
[Stop Condition]	ストップウォッチの終了条件となるブレークポイントまたはトリガを選択します。先に [Breakpoints] ダイアログに追加したブレークポイント/トリガの中から選択できます。終了条件をクリアするには、[None] を選択します。 [Stop condition will cause the target device to halt] にチェックを入れると、この条件でプログラム実行が停止します。
[Reset stopwatch on run]	プログラムを実行するたびにストップウォッチの値を 0 にリセットします。

コードにブレークポイントを設定するには、以下のいずれかの操作を行います。

- コードの任意の行をダブルクリックまたは右クリックして、個別にブレークポイントを設定する。
- [Debugger]>[Breakpoints] の順にクリックして [Breakpoints] ダイアログを開き、複数のブレークポイントとその条件を設定する (詳細は 10.3.1 「[Breakpoints] ダイアログ」参照)。

2つのブレークポイント間の時間を計測するには、以下の手順でストップウォッチを使用します。

1. [Breakpoints] ダイアログを開く ([Debugger]>[Breakpoints])。
2. [Breakpoints] ダイアログで [Stopwatch] をクリックして [Stopwatch] ダイアログを開く。
3. [Start Condition] のドロップダウン リストから、開始条件のブレークポイントを選択する。必要に応じて [Start condition will cause the target device to halt] にチェックを入れる。
4. [Stop Condition] のドロップダウン リストから、終了条件のブレークポイントを選択する。必要に応じて [Stop condition will cause the target device to halt] にチェックを入れる。
5. 必要に応じて [Reset stopwatch on run] にチェックを入れる。
6. [OK] をクリックする。

10.3.4 [Event Breakpoints] ダイアログ

[Breakpoints] ダイアログで [Event Breakpoints] をクリックすると、このダイアログが表示されます。

プログラム実行を必ず停止する条件をここで選択します。

- [Break on Watchdog Timer] — ウォッチドッグ タイマのタイムアウトで必ず停止します。コンフィグレーション ビットでウォッチドッグ タイマが有効に設定されている事を確認してください。
- [Break on SLEEP instruction] — プログラム内で SLEEP 命令が出現したら実行を停止します。

10.4 プログラミングに関する機能

[Programmer] メニューで PICKit 3 プログラマ / デバッガを選択すると、以下の MPLAB IDE 機能にプログラムに関する項目が追加されます。

- [Programmer] メニュー

- ツールバー / ステータスバー

10.4.1 [Programmer] メニュー

[Program]

指定したメモリ領域 (プログラムメモリ、コンフィグレーション ビット、ID ロケーション、EEPROM データ) にプログラミングを実行します。プログラミングのオプションは [Settings] ダイアログを参照してください。

[Verify]

指定したメモリ領域 (プログラムメモリ、コンフィグレーション ビット、ID ロケーション、EEPROM データ) に書き込んだ内容を検証します。

[Read]

指定したメモリ領域 (プログラムメモリ、コンフィグレーション ビット、ID ロケーション、EEPROM データ) から読み出しを実行します。読み出しのオプションは [Settings] ダイアログを参照してください。

[Blank Check All]

デバイスのメモリが全て消去されてブランクである事を確認します。

[Erase Flash Device]

フラッシュメモリ全体を消去します。

[Abort Operation]

プログラミング操作 (プログラム、読み出し等) を中止します。操作を中止すると、デバイスの状態は未知となります。

[Reconnect]

PC と PICkit 3 プログラマ / デバッガの通信再開を試みます。接続の進行状況は、[Output] ダイアログの [PICkit 3] タブに表示されます。

[Settings]

[Programmer] ダイアログを開きます (10.5 「[Settings] ダイアログ」参照)。プログラムとファームウェアのオプションを設定します。

10.4.2 ツールバー / ステータスバー

PICkit 3 プログラマ / デバッガをプログラマとして指定すると、MPLAB IDE に以下のツールバーが表示されます。

- 基本的なプログラム ツールバー ([Program]、[Read]、[Verify]、[Erase Flash Device]、[Blank Check All])。

MPLAB IDE デスクトップの一番下にあるステータスバーには、現在選択されているプログラマ (PICkit 3) とその他プログラミングに関する情報が表示されます。ステータスバーの表示内容は、MPLAB IDE のオンラインヘルプを参照してください。

10.5 [SETTINGS] ダイアログ

[Debugger]>[Settings] または [Programmer]>[Settings] の順にクリックすると [Settings] ダイアログが開き、PICkit 3 プログラマ / デバッガを設定できます。

Note: 表示されるタブの内容は、選択したデバイスと、PICkit 3 をデバッガとプログラマのいずれとして選択したかによって決まります。

- [Settings] ダイアログの [Program Memory] タブ
- [Settings] ダイアログの [Configuration] タブ
- [Settings] ダイアログの [Freeze on Halt] タブ

- [Settings] ダイアログの [Status] タブ
- [Settings] ダイアログの [Clock] タブ
- [Settings] ダイアログの [Secure Segment] タブ
- [Settings] ダイアログの [Warning] タブ
- [Settings] ダイアログの [Power] タブ
- [Settings] ダイアログの [Limitations] タブ
- [Settings] ダイアログの [Programmer-to-go] タブ

10.5.1 [Settings] ダイアログの [Program Memory] タブ

このタブを使うと、デバッグ / プログラミングのオプションを設定できます。

表 10-6: [PROGRAM MEMORY] タブのオプション

[Allow PICkit 3 to select memories and ranges]	こちらを選択すると、選択済みのデバイスと既定値の設定に基づいて、PICkit 3 がプログラムするメモリとアドレス範囲を決めます。
[Manually select memories and ranges]	こちらを選択すると、選択済みのデバイスのメモリ、プログラムメモリ範囲、プログラム オプションをユーザが決める必要があります。
[Memories:]	
[Program]	チェックを入れると、プログラムメモリをターゲットに書き込みます。
[Configuration]	チェックを入れると、コンフィグレーション ビットをターゲットに書き込みます。 Note: このメモリは、PICkit 3 をデバッグとして選択すると常に書き込まれます。
[EEPROM]	チェックを入れると、ターゲットが EEPROM メモリを実装している場合、消去してから書き込みを実行します。チェックを外すと、ターゲットの EEPROM メモリを消去します。
[ID]	チェックを入れると、ID メモリをターゲットに書き込みます。
[Boot Flash]	サポートされている場合、チェックを入れるとターゲットのブートメモリに書き込みます。
[Program Memory Range:]	
[Start]、[End]	書き込み、読み出し、ベリファイを行うプログラムメモリのアドレス範囲を開始アドレスと終了アドレス (いずれも 16 進数) で指定します。 終了アドレスが正しくないためプログラミング エラーが発生した場合、再接続後に終了アドレスを訂正して書き込みを再度実行する必要があります。 Note: ここで指定したアドレス範囲は消去機能には適用されません。消去機能はデバイスの全データを対象に実行されます。
[Program Options:]	
[Erase all before Program]	チェックを入れると、書き込みを開始する前にメモリ全体を消去します。 新しいデバイスまたは消去済みデバイスに書き込む場合を除き、このボックスには必ずチェックを入れてください。チェックを入れない場合はデバイスが消去されず、デバイスに既存のコードと新しいプログラムコードがマージされます。
[Preserve Program Memory Range]	チェックを入れると、メモリ範囲が保持されます。 [Start]、[End] – 保持するプログラムメモリの範囲を開始アドレスと終了アドレス (いずれも 16 進数) で指定します。

表 10-6: [PROGRAM MEMORY] タブのオプション (続き)

[Use high voltage on MCLR]	<p>PIC24FXXKAXXX の場合： ピンを通常の入力ピンまたは MCLR リセットに設定する際に高電圧を使う場合、チェックを入れます。 低電圧の MCLR リセットの場合、チェックを外します。</p> <p>Notes:</p> <ol style="list-style-type: none"> 1. コンフィグレーション設定が「MCLR ピン = 有効、RA5 入力ピン = 無効」の場合、低電圧エントリを使用できます ([Use high voltage on MCLR] オプションのチェックを外す)。 2. コンフィグレーション設定が「RA5 入力ピン = 有効、MCLR = 無効」の場合、[Use high voltage on MCLR] オプションにチェックを入れる必要があります。 3. コンフィグレーション設定で「MCLR ピン イネーブルビット」をプログラムする場合、[Use high voltage on MCLR] オプションにチェックを入れる必要があります。 4. 「-ICE」ヘッダを使う場合、このチェックボックスの設定は無視されます。
[Automatically]	
[Program after successful build]	アプリケーション コードのビルドが正常に完了したら、このコードをデバイスに書き込みます。
[Run after successful program]	このオプションはデバッグモードでのみ有効です。アプリケーション コードをターゲット デバイスに正しく書き込んだ後、コードを実行します。

10.5.2 [Settings] ダイアログの [Configuration] タブ

このタブでは、デバッグの動作を設定します。

表 10-7: [CONFIGURATION] タブの項目

[Download Firmware]	ファームウェア ダウンロードのオプションを設定します。
[Auto Download Latest Firmware]	チェックを入れると、ターゲット デバイスの最新バージョンのファームウェアを自動的にダウンロードできます (推奨)。
[Manual Download]	ターゲット デバイスにダウンロードするファームウェア ファイルを手動で選択します。

10.5.3 [Settings] ダイアログの [Freeze on Halt] タブ

このタブでは、Halt 時にフリーズする周辺モジュールを選択します。

PIC12/16 MCU

デバイスの全ての周辺モジュールに対して停止時のフリーズの有効 / 無効を設定するには、[Freeze on Halt] チェックボックスの ON/OFF を切り換えます。チェックを入れても特定の周辺モジュールが停止しない場合、一部の周辺モジュールは Freeze on Halt 機能をサポートしておらず、エミュレータでは制御できないため注意してください。

デバイスによっては、Freeze on Halt をサポートする周辺モジュールに制約がある場合があります。制約に関する詳細は、[Limitations] ボタンをクリックして確認します。

PIC18 MCU

デバイスの全ての周辺モジュールに対して停止時のフリーズの有効 / 無効を設定するには、[Freeze on Halt] チェックボックスの ON/OFF を切り換えます。チェックを入れても特定の周辺モジュールが停止しない場合、一部の周辺モジュールは Freeze on Halt 機能をサポートしておらず、デバッグでは制御できないため注意してください。

dsPIC30F/33F、PIC24F/H、PIC32MX

[Peripherals to Freeze on Halt] リストで、Halt 時にフリーズしたい周辺モジュールにチェックを入れます。チェックを外した周辺モジュールは、プログラム実行が停止しても動作を継続します。該当の周辺モジュールがリストにない場合、[All Other Peripherals] にチェックを入れます。チェックを入れても特定の周辺モジュールが停止しない場合、一部の周辺モジュールは Freeze on Halt 機能をサポートしておらず、デバッグでは制御できないため注意してください。

[All Other Peripherals] を含め、全ての周辺モジュールを選択するには、[Check All] をクリックします。[All Other Peripherals] を含め、全ての周辺モジュールの選択を解除するには、[Uncheck All] をクリックします。

10.5.4 [Settings] ダイアログの [Status] タブ

このタブでは、PICkit 3 システムのステータスを確認できます。

表 10-8: [STATUS] タブの項目

[Versions]	
[Firmware Suite Version]	デバッグのファームウェアスイートのバージョンです。ファームウェアスイートは以下の 3 つの項目で構成されます。
[Algorithm Plug-in Version]	デバッグのアルゴリズム プラグインのバージョンです。選択したデバイスに、ターゲットに装着したデバイスをサポートするアルゴリズムを使用します。
[OS Version]	デバッグのオペレーティング システムのバージョンです。
[Voltages]	
[PICkit™ 3 VPP]	デバッグの VPP です。
[PICkit 3 VDD]	デバッグの VDD です。
[Refresh Voltages]	[Status] タブに表示される電圧値は、このタブがアクティブになった時点のものです。表示内容を更新したい場合、このボタンをクリックします。

10.5.5 [Settings] ダイアログの [Clock] タブ

このタブでは、デバッグクロック モードを選択します。[FRC in Debug mode] にチェックを入れると、デバッグ中は高速 RC クロックを使用します。

デバイスのエミュレーションはプロセッサの動作速度で行われます。プロセッサの動作周波数が低い場合 (32 kHz 等)、ステップ実行、[Watch] ウィンドウのリフレッシュ等、全ての動作が遅くなります。しかし、プロセッサによっては高速 RC オシレータを内蔵しているものもあります。[FRC in Debug mode] オプションにチェックを入れると、アプリケーション (通常の動作速度) の停止後に高速オシレータに切り換わり、エミュレーションと周辺モジュール (フリーズしていない場合) をより高速で実行できます。

10.5.6 [Settings] ダイアログの [Secure Segment] タブ

CodeGuard™ セキュリティ対応デバイスは、このタブでセキュア セグメントのプロパティを設定します。

CodeGuard セキュリティ機能の詳細は、マイクロチップ社ウェブサイトで公開されている 16 ビットデバイス向けリファレンス マニュアルの『Section 1. CodeGuard Security』(DS70180)、dsPIC33F/PIC24H および dsPIC30F のプログラミング仕様を参照してください。

表 10-9: [SECURE SEGMENT] タブのオプション

[Full Chip Programming]	クリックすると、プログラムメモリの全てのセグメントに対する書き込みが選択されます。
[Segment Programming]	クリックすると、セグメント プログラミングが選択されます。以下のいずれかを選択します。 - [Boot, secure and general segments] - [Secure and general segments] - [General segment only]

10.5.7 [Settings] ダイアログの [Warning] タブ

このタブには、PICkit 3 プログラマ / デバッグの全ての警告が一覧表示されます。

- 警告の項目にチェックを入れると、その警告が有効になります。警告は [Output] ウィンドウに表示されます。
- 警告の項目のチェックを外すと、その警告は無効になります。

エラーとは異なり、警告が発生してもプロジェクトのビルドは中止されません。エラーメッセージが出力された場合、**Chapter 9. 「エラーメッセージ」**を参照してください。

10.5.8 [Settings] ダイアログの [Power] タブ

このタブでは、電源オプションを設定します。

- チェックボックスをクリックして、[Power target circuit from PICkit 3] の有効 / 無効を切り換えます。
この項目にチェックを入れると、ツールバーの [Power on/off] ボタンが有効になります。このボタンは、初期状態では「電源 ON」の状態です。クリックするたびに ON と OFF が切り換わります。ON の時にクリックすると OFF になり、OFF の時にクリックすると ON になります。[Power target circuit from PICkit 3] のチェックを外すと、[Power on/off] ボタンは無効な状態に戻ります。
ワークスペースを開くと、最後にそのワークスペースを保存した時の状態で開きます。
- スライダーを調整して電圧を設定します。スライダーを動かすと、フィールドの数値が変化します。

10.5.9 [Settings] ダイアログの [Limitations] タブ

このタブには、選択したデバイスの全般的な制約が表示されます。[Details] をクリックすると、制約の詳細が表示されます。

10.5.10 [Settings] ダイアログの [Programmer-to-go] タブ

このタブでは、プログラミング イメージに名前を付けて PICkit 3 のメモリに送信できます。

表 10-10: [PROGRAMMER-TO-GO] タブのオプション

[Image Name]	プログラミング イメージの名前を入力します。
[Programmer Setting Report]	デバイスのプログラミングに関して選択した設定を表示します。設定を変更する場合、[PICkit 3 Settings] ダイアログの適切なタブを使います。
[Send Image In Memory]	MPLAB IDE から PICkit 3 ヘイメージ転送を実行します。

補遺 A. ハードウェア仕様

A.1 はじめに

ここでは、PICKit 3 プログラマ / デバッガシステムのハードウェア仕様と電氣的仕様について詳しく説明します。

A.2 ハイライト

この章の内容は以下の通りです。

- 適合宣言書
- USB ポート / 電源
- PICKit 3 プログラマ / デバッガ
- 標準通信ハードウェア
- ターゲットボードに関する注意事項

A.3 適合宣言書

当社

Microchip Technology Inc.
2355 W. Chandler Blvd.
Chandler, Arizona 85224-6199
USA

は、以下の製品

PICKit 3 プログラマ / デバッガ

が、操作マニュアルに記載された制限事項を遵守した場合に以下の規格に適合する事を宣言します。

規格 : EN61010-1 試験所用電気機器
Microchip Technology, Inc.

日付 : 2009 年 1 月

PICKit 3 プログラマ / デバッガの使用に関する重要な注意事項

PICKit 3 プログラマ / デバッガは、その特殊な性質により、通常より高いレベルの電磁放射を発生して無線機器等あらゆる種類の機器の動作に影響を与える事があるので注意してください。

従って、上記欧州規格に適合するためには、以下の制限事項を遵守する必要があります。

1. 本開発システムの使用は工業地域 (または同等地域) のみに限定する事。
2. 無線受信機やテレビ等、電磁波の影響を受ける可能性のある機器から 20 メートル以内の場所で本システムを使用しない事。

PICKit™ 3 ユーザガイド

A.4 USB ポート / 電源

PICKit 3 プログラマ / デバッガは、mini USB 2.0 ポートを介してホスト PC に接続します。USB コネクタはポッドの上部にあります。

本システムは USB インターフェイスを利用してファームウェアを再読み込みできます。

システムの電源は USB インターフェイスから供給されます。このデバッガは USB 仕様の「ハイパワー システム」に分類され、デバッガモードとプログラマモードのいずれの場合も 100 mA をやや上回る電力を USB から供給する必要があります。

Note: PICKit 3 プログラマ / デバッガの電源は USB 接続から供給します。ターゲットボードの電源は原則として自己給電とします。ただし、ターゲットの消費電流が 30 mA 未満の場合に限り、PICKit 3 から電源を供給する事もできます。

ケーブル長 — PC とデバッガを接続するケーブルは、正常動作に適した長さのものをキットに同梱しています。

セルフパワーハブ — USB ハブを使う場合、バスパワーではなくセルフパワータイプを使います。PC キーボード内蔵の USB ポートでは、デバッガの動作に十分な電力が得られません。

PC の休止 / 省電力モード — USB 接続による PC とデバッガの通信を確保するため、休止モード等 PC の省電力モードを無効にします。

A.5 PICKIT 3 プログラマ / デバッガ

デバッガの筐体はメインボードを格納しており、USB コネクタとシングルインラインコネクタが付いています。デバッガの筐体にはインジケータ ライト (LED) もあります。

A.5.1 メインボード

メインボードには、インターフェイス プロセッサ、USB 2.0 インターフェイス、SPI、エミュレーション デバイスへのプログラミング用のシリアル EE、内蔵フラッシュ、インジケータ LED が実装されています。

A.5.2 インジケータ ライト (LED)

インジケータ ライトの意味は以下の通りです。

LED	色	説明
POWER	緑	電源を投入またはターゲットを接続すると点灯します。
ACTIVE	青	PICKit™ 3 と PC の接続が確立されている時、またはコマンド送受信時に点灯します。
STATUS	緑	デバッガが正常に動作しており、スタンバイ状態の時に点灯します。
	黄	動作中 (ビジー) の時に点灯します。
	赤	デバッガにエラーが発生すると点灯します。

A.6 標準通信ハードウェア

デバッガとターゲットの標準通信 (2.3「デバッガとターゲットの通信」の「標準 ICSP による通信」) には、RJ-11 コネクタ付きのアダプタを使用します。

ヘッダボード使用時に標準通信を行うには、デバイス専用のプロセッサパック (-ICE/ICD デバイスを装着した 8 ピンコネクタのヘッダボードと標準アダプタボードのセット) が必要になる事があります。

Note: 旧式のヘッダボードは 8 ピンコネクタではなく 6 ピンの RJ-11 コネクタを使っているため、デバッガと直接接続できます。

利用可能なヘッダボードの詳細は、『Header Board Specification』(DS51292) を参照してください。

A.6.1 標準通信

標準通信は、ターゲット プロセッサとのメイン インターフェイスとなるものです。この中には、ターゲット デバイスへの接続とプログラミングに必要な高電圧 (VPP)、VDD 検出ライン、クロックおよびデータ接続が含まれます。

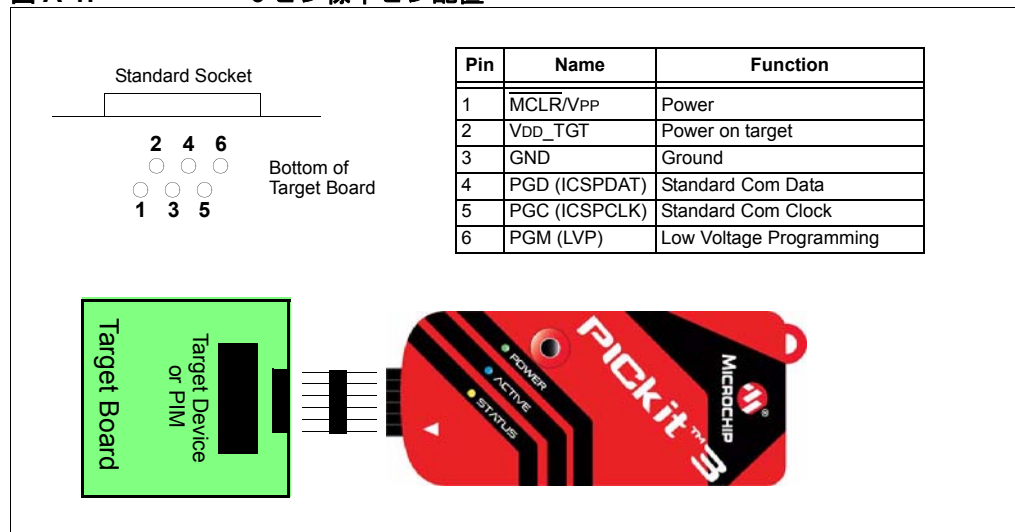
VPP の高電圧ラインは、エミュレーション プロセッサの電圧要件に応じて 1.8 ~ 14 V の電圧を生成できます。

VDD 検出ラインには、ターゲット プロセッサからの電流が流れます。

クロックとデータの接続インターフェイスには、以下の特長があります。

- 高インピーダンス モードのクロックおよびデータ信号 (PICkit 3 プログラマ / デバッガシステムに電源が供給されていない場合を含む)
- ターゲット システムの不良または不適切な接続による過電圧からクロックおよびデータ信号を保護
- プロトタイプまたはターゲット システムの電氣的短絡による過電流からクロックおよびデータ信号を保護

図 A-1: 6 ピン標準ピン配置



A.6.2 モジュラケーブルとコネクタ

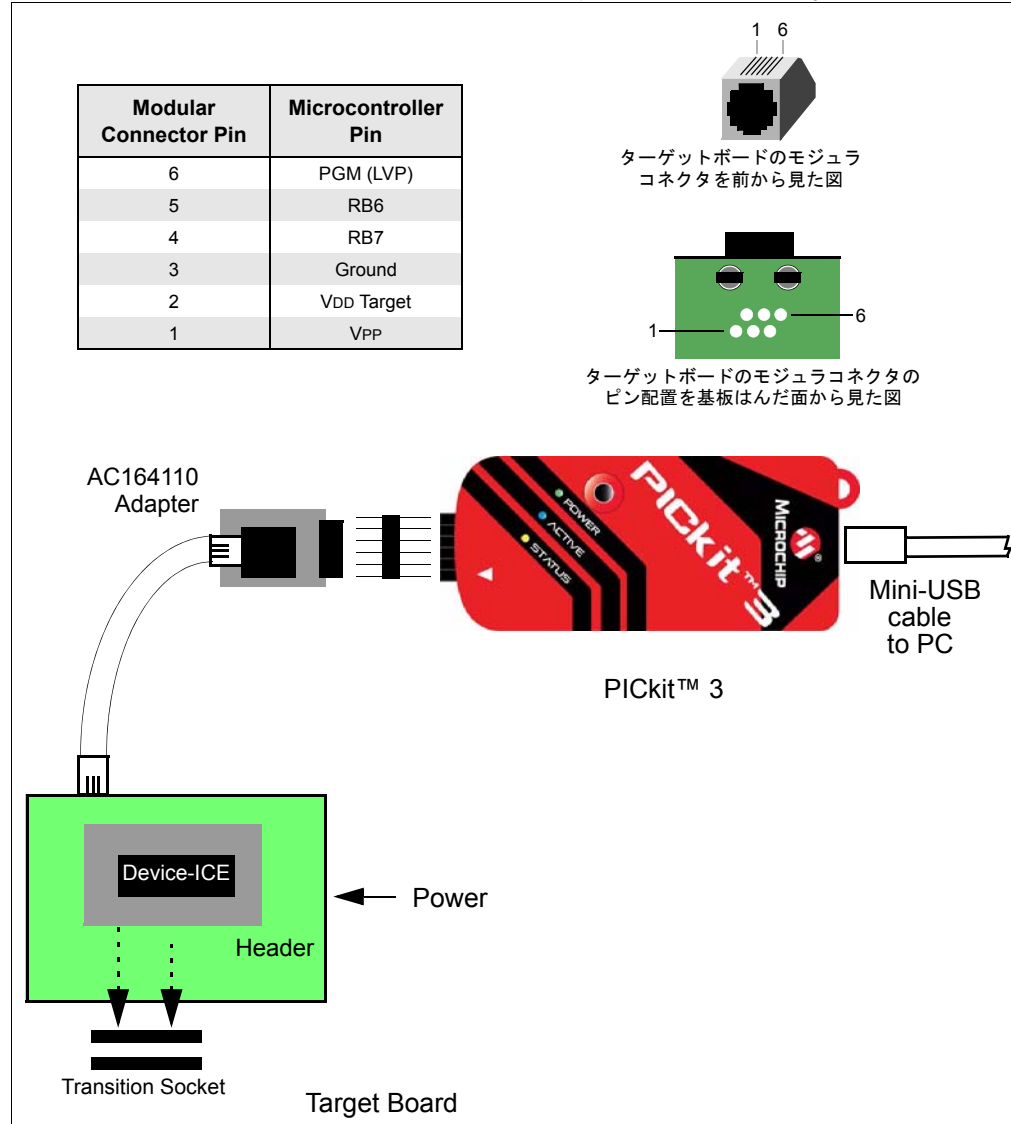
標準通信では、デバッガとターゲットアプリケーションをモジュラケーブルで接続します。以下に、このケーブルとコネクタの仕様を示します。

A.6.2.1 モジュラコネクタの仕様

- メーカー名、製品番号 — AMP Incorporated、555165-1
- 販売代理店名、製品番号 — Digi-Key、A9031ND

図 A-2 の表に、アプリケーション側のモジュラコネクタの各ピンとマイクロコントローラのピンの対応を示します。この構成では、ICD の機能を全て利用できます。

図 A-2: ターゲットボードのモジュラコネクタのピン配置



A.6.2.2 モジュラプラグの仕様

- メーカー名、製品番号 — AMP Incorporated、5-554710-3
- 販売代理店名、製品番号 — Digi-Key、A9117ND

A.6.2.3 モジュラケーブルの仕様

- メーカー名、製品番号 — マイクロチップ テクノロジー社、07-00024

A.7 ターゲットボードに関する注意事項

ターゲットボードには、選択したデバイス (1.8 ~ 5.0 V) とアプリケーションの要件に合わせて電源を供給してください。

デバッガとターゲットの通信の方法によっては、ターゲットボードの回路に関して何点かの注意事項があります。詳しくは以下のセクションを参照してください。

- 2.4.2「ターゲットとの接続回路」
- 2.4.5「デバッガの正常動作を妨げる回路」

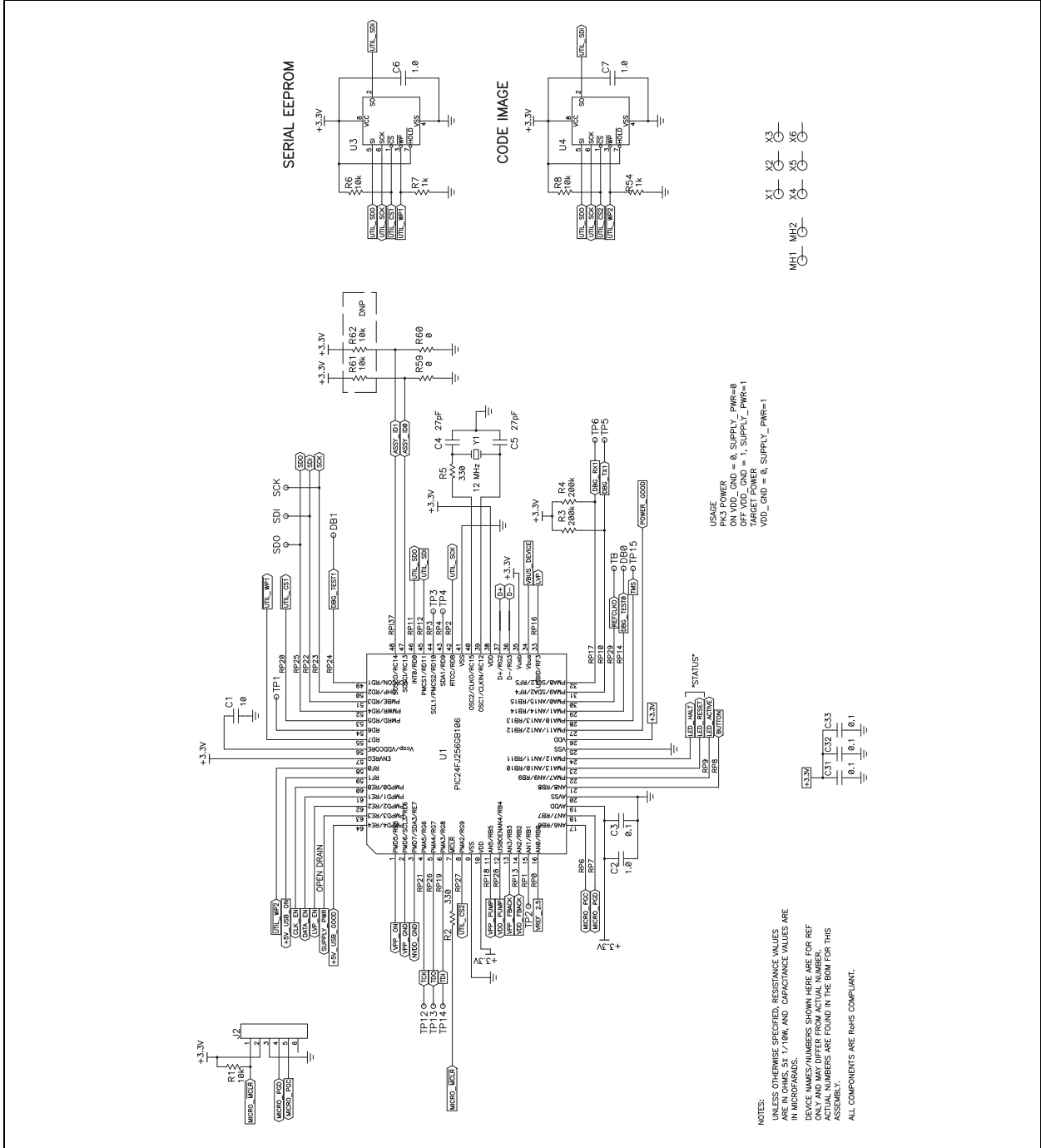
PICKit™ 3 ユーザガイド

NOTES:

補遺 B. PICKit 3 の回路図

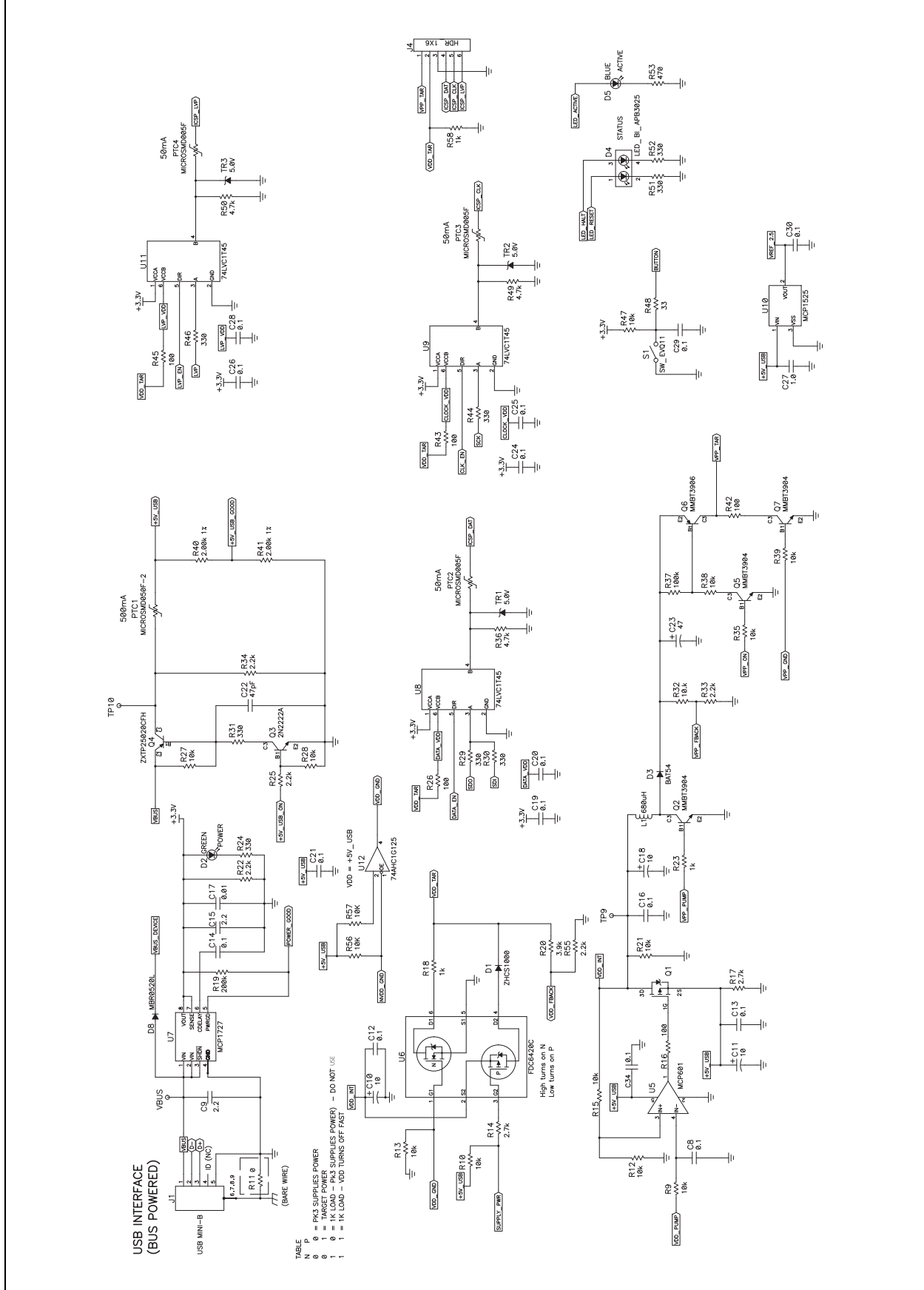
ここでは、PICKit 3 開発プログラマ / デバッガの回路図を示します。デモボードの回路図は、各ボードのユーザガイドを参照してください。

図 B-1: PICKit™ 3 の回路図 (1/2 ページ)



PICKit™ 3 ユーザガイド

図 B-2: PICKit™ 3 の回路図 (2/2 ページ)



用語集

16 進数 (Hexadecimal)

0～9の数字とA～F(またはa～f)のアルファベットを使用する、16を底とした記数法。A～Fで10進数の10～15を表現する。一番右の桁が1の位、次の桁が16の位、その次の桁が $16^2 = 256$ の位を表す。

1対1のプロジェクト/ワークスペース モデル (One-to-One Project-Workspace Model)

1つのワークスペースで1つのプロジェクトを扱う。MPLAB IDEにおけるアプリケーション開発の最も一般的な構成。[\[Configure\]>\[Settings\]](#)の順に選択して[\[Projects\]](#)タブで[\[Use one-to-one project-workspace model\]](#)にチェックを入れて指定する。

2 進数 (Binary)

0と1の数字を使用する、2を底とした記数法。一番右の桁が1の位、次の桁が2の位、その次の桁が $2^2 = 4$ の位を表す。

8 進数 (Octal)

0～7の数字のみ使用する、8を底とした記数法。一番右の桁が1の位、次の桁が8の位、その次の桁が $8^2 = 64$ の位を表す。

[Watch] ウィンドウ (Watch Window)

ウォッチ変数の一覧が表示され、ブレークポイントで毎回表示が更新されるウィンドウ。

AND 条件ブレークポイント (ANDed Breakpoints)

プログラム実行を停止するために設定するAND条件(ブレークポイント1とブレークポイント2が同時に発生した場合のみプログラム実行を停止する)。AND条件で実行が停止するのは、データメモリのブレークポイントとプログラムメモリのブレークポイントが同時に発生した場合のみ。

ANSI

American National Standards Institute (米国規格協会)の略。米国における標準規格の策定と承認を行う団体。

ASCII

American Standard Code for Information Interchangeの略。7桁の2進数で1つの文字を表現する文字セットエンコード方式。大文字、小文字、数字、記号、制御文字等が含まれる。

Clean

MPLAB IDEの[Project]メニューの項目。アクティブなプロジェクトのオブジェクトファイル、Hexファイル、デバッグファイル等、全ての間接ファイルが削除される。これらのファイルは、プロジェクトのビルド時に他のファイルから再構築される。

COFF

Common Object File Formatの略。このフォーマットのオブジェクトファイルには、マシンコードの他、デバッグ等に関する情報が含まれる。

CPU

「中央演算処理装置」を参照。

C 言語 (C)

簡潔な表現、現代的な制御フローとデータ構造、豊富に用意された演算子等の特長とする汎用プログラミング言語。

DSC

「デジタル シグナル コントローラ」を参照。

DSP

「デジタル シグナル プロセッサ」を参照。

dsPIC DSC (dsPIC DSCs)

dsPIC デジタル シグナル コントローラの略。マイクロチップ社の DSC ファミリの総称。

DWARF

Debug With Arbitrary Record Format の略。ELF ファイルのデバッグ情報フォーマット。

EEPROM

Electrically Erasable Programmable Read Only Memory の略。電氣的に消去可能なタイプの PROM。データの書き込みと消去はバイト単位で行われる。EEPROM の内容は電源を OFF にしても保持される。

ELF

Executable and Linking Format の略。この形式のオブジェクト ファイルにはマシンコードが含まれる。デバッグその他の情報は DWARF で指定する。ELF/DWARF の方が COFF よりも最適化したコードのデバッグに適している。

EPROM

Erasable Programmable Read Only Memory の略。再書き込みが行えるタイプの ROM で、消去は紫外線照射によって行うものが主流。

FNOP

Forced No Operation の略。Forced NOP サイクルは、2 サイクル命令の 2 サイクル目で発生する。PIC マイクロコントローラのアーキテクチャはパイプライン構造となっており、現在の命令を実行中に物理アドレス空間の次の命令がプリフェッチされる。しかし、現在の命令によってプログラム カウンタが変化した場合、プリフェッチした命令は明示的に無視され、Forced NOP サイクルが発生する。

GPR

General Purpose Register (汎用レジスタ) の略。デバイスのデータメモリ (RAM) のうち、汎用目的に使える部分。

Halt

プログラム実行を停止する事。Halt を実行する事は、ブレークポイントで停止する事と同じ。

Hex コード (Hex Code)

実行可能な命令を 16 進数形式のコードで保存したもの。Hex コードは Hex ファイルに保存される。

Hex ファイル (Hex File)

デバイスに書き込み可能な 16 進数形式のアドレスと値 (Hex コード) を記述した ASCII ファイル。

ICD

In-Circuit Debugger (インサーキット デバッガ) の略。MPLAB ICD と PICkit (Debug Express を含む) はマイクロチップ社の ICD。

ICE

In-Circuit Emulator (インサーキット エミュレータ) の略。MPLAB ICE 2000 と MPLAB ICE 4000 システムはマイクロチップ社の従来型 ICE。MPLAB REAL ICE システムはマイクロチップ社の次世代 ICE。

ICSP™

In-Circuit Serial Programming (インサーキット シリアル プログラミング) の略。マイクロチップ社製の組み込みデバイスをシリアル通信を利用して最小限のデバイスピンでプログラミングする方法。

IDE

Integrated Development Environment (統合開発環境) の略。マイクロチップ社の MPLAB IDE 等。

IEEE

Institute of Electrical and Electronics Engineers の略。

IRQ

「割り込み要求」を参照。

ISO

「国際標準化機構 (International Organization for Standardization)」を参照。

ISR

「割り込みサービスルーチン」を参照。

LVDS

Low Voltage Differential Signaling の略。銅線を使って、低ノイズ、低消費電力、低振幅でデータを高速伝送 (Gbps) する方法。

通常の I/O とはいくつかの点で異なる。

通常のデジタル I/O は 5 V を High (2 進数の「1」)、0 V を Low (2 進数の「0」) として動作する。一方、差動伝送方式では 3 つ目の電圧値 (-5 V) を使用する。これにより、符号化に使用できるレベルが 1 つ増え、最大データ転送レートが向上する。

データ転送レートが向上すると、信号線の数が少なくすみ、UW (Ultra Wide) および UW-2/3 SCSI ハードディスクでは 68 芯しか使用しない。これらのデバイスでは、短距離で高いデータ転送レートが要求される。標準の I/O 転送方式では、SCSI ハードディスクドライブの接続に 68 芯よりもはるかに多くの信号線が必要になる。

使用する電圧も標準の 5 V より低く、3.3 V または 1.5 V を使用する。

LVDS では、2 本の信号線を使って逆位相の信号を送信する。この方式ではノイズが 2 つの信号線とも同じレベルで伝送されるため、ノイズを簡単かつ効果的にフィルタリングできる。

標準の I/O シグナリングでは、データストレージは実際の電圧レベルに依存する。電圧レベルは信号線の長さによって影響を受ける (信号線が長いと抵抗が増大し、電圧が低下する)。これに対し LVDS では、データストレージは実際の電圧レベルでなく正と負の電圧値によってのみ区別する。従って、長い信号線でもクリアで安定したデータストリームを維持した伝送が可能。

出典 : <http://www.webopedia.com/TERM/L/LVDS.html>

Makefile

プロジェクトの Make に関する指示をファイルにエクスポートしたもの。このファイルは、MPLAB IDE 以外の環境で make コマンドを実行してプロジェクトをビルドする際に使用する。

Makefile をエクスポートするには、[Project]>[Build Options]>[Project] の順にクリックして、[Directories] タブの [Build Directory Policy] で [Assemble/Compile/Link in the project directory] を選択しておく必要がある。

Make Project

アプリケーションを再ビルドするコマンド。前回の完全なコンパイル後に変更されたソースファイルのみを再コンパイルする。

MCU

Microcontroller Unit の略。マイクロコントローラの事。「 μ C」と表記する事もある。

MPASM™ アセンブラ (MPASM Assembler)

PIC マイクロコントローラ、KEELOQ® デバイス、マイクロチップ社のメモリデバイスに対応したマイクロチップ社の再配置可能なマクロアセンブラ。

MPLAB (言語ツール名) for (デバイス名) (MPLAB Language Tool for Device)

特定のデバイスに対応したマイクロチップ社の C コンパイラ、アセンブラ、リンカ。言語ツールは、アプリケーションで使用するデバイスに対応したものを選択する必要がある。例えば PIC18 MCU 用の C コードを作成する場合は「MPLAB C Compiler for PIC18 MCU」を使用する。

MPLAB ICD

MPLAB IDE と組み合わせて使用するマイクロチップ社のインサーキット デバugg。ICD では、デバugg回路を内蔵したフラッシュ デバイスがサポートされる。ICE の中心的なコンポーネントはポッドと呼ばれる。ポッド、ヘッダボード(「デバイス名-ICD」を搭載)、ターゲットボード、ケーブル、MPLAB IDE ソフトウェアを組み合わせて完全なシステムが構成される。

MPLAB ICE 2000/4000

新規デザインには推奨しません。「MPLAB REAL ICE インサーキット エミュレータ」を参照。

MPLAB IDE と組み合わせて使用するマイクロチップ社の従来型インサーキット エミュレータ。MPLAB ICE 2000 では 8 ビット PIC MCU がサポートされる。MPLAB ICE 4000 では PIC18F、PIC24 MCU、dsPIC DSC がサポートされる。ICE の中心的なコンポーネントはポッドと呼ばれる。ポッド、プロセッサモジュール、ケーブル、MPLAB IDE ソフトウェアを組み合わせて完全なシステムが構成される。

MPLAB IDE

マイクロチップ社の統合開発環境。エディタ、プロジェクトマネージャ、シミュレータが付属する。

MPLAB PM3

マイクロチップ社提供のデバイス プログラマ。PIC18 マイクロコントローラと dsPIC デジタル シグナル コントローラの書き込みに対応。MPLAB IDE との併用も、単体での使用も可能。PRO MATE II の後継製品。

MPLAB REAL ICE™ インサーキット エミュレータ (MPLAB REAL ICE™ In-Circuit Emulator)

MPLAB IDE と組み合わせて使用するマイクロチップ社の次世代インサーキット エミュレータ。PIC MCU と dsPIC DSC をサポートする。ICE の中心的なコンポーネントはポッドと呼ばれる。ポッド、ドライバ(および場合によってはレシーバ)カード、ケーブル、MPLAB IDE ソフトウェアを組み合わせて完全なシステムが構成される。

MPLAB SIM

MPLAB IDE と組み合わせて使用するマイクロチップ社のシミュレータで、PIC MCU と dsPIC DSC に対応する。

MPLAB Starter Kit for (デバイス名) (MPLAB Starter Kit for Device)

特定のデバイスでの作業を開始する上で必要となるものを全てセットにしたマイクロチップ社のスタータキット。既製のアプリケーションの動作を確認した後で、一部を変更してカスタム アプリケーションとしてデバuggとプログラムを行う。

MPLIB™ オブジェクト ライブラリアン (MPLIB™ Object Librarian)

MPLAB IDE と組み合わせて使用するマイクロチップ社のライブラリアン。MPLIB ライブラリアンは、MPASM アセンブラ (mpasm または mpasmwin v2.0) または MPLAB C18 C コンパイラで作成した COFF オブジェクト モジュールに使用するオブジェクト ライブラリアン。

MPLINK™ オブジェクト リンカ (MPLINK™ Object Linker)

マイクロチップ社の MPASM アセンブラと C18 C コンパイラに対応したオブジェクト リンカ。マイクロチップ社の MPLIB ライブラリアンとの併用も可能。MPLAB IDE に統合して使用できるように設計されているが、MPLAB IDE 以外の環境でも使用できる。

MRU

Most Recently Used の略。最近使用したファイルおよびウィンドウのこと。MPLAB IDE のメインメニューで選択できる。

NOP

No Operation の略。実行してもプログラム カウンタが進むだけで何も動作を行わない命令。

OTP

One Time Programmable の略。パッケージに窓のない EPROM デバイス。EPROM を消去するには紫外線照射が必要なため、パッケージに窓のあるデバイスしか消去できない。

PC

パーソナル コンピュータまたはプログラム カウンタの略。

PICKit 1/2/3 (PICKit 1, 2, and 3)

マイクロチップ社の開発用デバイス プログラマで、Debug Express によるデバッグ機能を備える。サポートしているデバイスの種類は、各ツールの Readme ファイルを参照。

PIC MCU (PIC MCUs)

マイクロチップ社の全てのマイクロコントローラ ファミリの総称。

PICSTART Plus

マイクロチップ社提供の開発用デバイス プログラマ。8/14/28/40 ピンの PIC マイクロコントローラのプログラミングに対応。書き込みには必ず MPLAB IDE ソフトウェアを使用する。

PRO MATE II

生産終了品です。「MPLAB PM3」を参照してください。

マイクロチップ社提供のデバイス プログラマ。ほとんどの PIC マイクロコントローラ、メモリ、KEELOQ デバイスへの書き込みに対応。MPLAB IDE との併用も、単体での使用も可能。

PWM 信号 (PWM Signals)

パルス幅変調 (Pulse Width Modulation) 信号。一部の PIC MCU には周辺モジュールとして PWM が内蔵されている。

RAM

Random Access Memory の略。データメモリ。任意の順にメモリ内の情報にアクセスできる。

ROM

Read Only Memory の略。プログラムメモリ。メモリの内容を変更できない。

Run

エミュレータを Halt から解放するコマンド。エミュレータはアプリケーション コードを実行し、I/O に対してリアルタイムに変更、応答を行う。

SFR

「特殊機能レジスタ」を参照。

Single Step

コードを 1 命令ずつ実行するコマンド。1 命令を実行するたびに、MPLAB IDE のレジスタ ウィンドウ、ウォッチ変数、ステータス ディスプレイの表示が更新されるため、命令実行を解析してデバッグできる。C コンパイラのソースコードもシングルステップ実行できるが、その場合は 1 命令ずつ実行されるのではなく、高級言語の C で記述されたコードの 1 行から生成される全てのアセンブリレベル命令がシングルステップで実行される。

Serialized Quick Turn Programming

デバイス プログラムでマイクロコントローラをプログラムする際に、各デバイスに異なるシリアル番号を書き込めるようにする機能。エントリコード、パスワード、ID 番号等を書き込む目的で使用する。

SQTP

「Serialized Quick Turn Programming」を参照。

Step Into

Single Step と同じコマンド。Step Over とは異なり、Step Into では CALL 命令によって呼び出されるサブルーチンもステップ実行される。

Step Out

現在ステップ実行中のサブルーチンから抜け出すためのコマンド。このコマンドを実行すると、サブルーチンの残りのコードが全て実行され、サブルーチンのリターンアドレスで実行が停止する。

Step Over

Step Over を実行すると、サブルーチン内はステップ実行せずにデバッグできる。Step Over では、CALL 命令があると CALL の次の命令にブレークポイントが設定される。何らかの理由により、サブルーチンが無限ループになる等、正しくリターンしない場合、次のブレークポイントには到達しない。CALL 命令の処理以外は、Step Over コマンドと Single Step コマンドは同じ。

USB

Universal Serial Bus の略。2 本のシリアル伝送線で PC と外部周辺機器の通信を行う外部周辺インターフェイス規格。USB 1.0/1.1 でサポートされるデータ転送レートは 12 Mbps。USB 2.0 (Hi-Speed USB) は最大 480 Mbps のデータレートをサポートしている。

WDT

「ウォッチドッグ タイマ」を参照。

アーカイバ (Archiver)

ライブラリを作成、操作するためのツール。

アーカイブ (Archive)

再配置可能なオブジェクト モジュールを集めたもの。複数のソースファイルをオブジェクト ファイルにアセンブルした後、アーカイバを使ってこれらオブジェクト ファイルを 1 つのライブラリ ファイルにまとめると生成される。ライブラリをオブジェクト モジュールや他のライブラリとリンクすると、実行コードが生成される。

アクセス エントリ ポイント (Access Entry Points)

リンク時に定義されていない可能性のある関数に、セグメントの境界を越えて制御を渡すための手段。ブートセグメントとセキュア アプリケーション セグメントを別々にリンクする方法も提供する。

アクセスメモリ (Access Memory)

PIC18 のみ – PIC18 でバンク セレクト レジスタ (BSR) の設定にかかわらずアクセスできる特殊なレジスタ。

アセンブラ (Assembler)

アセンブリ言語のソースコードをマシンコードに変換する言語ツール。

アセンブリ言語 (Assembly Language)

2 進数のマシンコードをシンボル表現で記述したプログラミング言語。

アップロード (Upload)

エミュレータやプログラマ等のツールからホスト PC へ、またはターゲットボードからエミュレータへデータを転送する事。

アドレス (Address)

メモリ内の位置を一意に特定する値。

アプリケーション (Application)

PIC[®] マイクロコントローラで制御されるソフトウェアとハードウェアを組み合わせたもの。

アルファベット文字 (Alphabetic Character)

アルファベットの小文字と大文字の総称 (a, b, ..., z, A, B, ..., Z)。

イベント (Event)

アドレス、データ、パスカウント、外部入力、サイクルタイプ (フェッチ、R/W)、タイムスタンプ等、バスサイクルを記述したもの。トリガ、ブレイクポイント、割り込みを記述するために使用する。

入れ子の深さ (Nesting Depth)

マクロに他のマクロを含める事のできる階層の数。

インポート (Import)

Hex ファイル等の外部ソースから MPLAB IDE にデータを取り込む事。

ウォッチドッグ タイマ (Watchdog Timer)

PIC マイクロコントローラに内蔵されたタイマの 1 つで、ユーザが設定した期間が経過するとプロセッサをリセットする。WDT の有効化 / 無効化、設定はコンフィグレーション ビットで行う。

ウォッチ変数 (Watch Variable)

デバッグ セッション中に [Watch] ウィンドウで観察できる変数。

英数字 (Alphanumeric)

アルファベット文字と 0 ~ 9 の 10 進数の数字の総称。

永続データ (Persistent Data)

クリアも初期化も行えないデータ。デバイスをリセットしてもアプリケーションがデータを保持できるようにするために使用する。

エクスポート (Export)

MPLAB IDE のデータを標準フォーマットで外部に出力する事。

エピローグ (Epilogue)

コンパイラで生成したコードのうち、スタック領域の割り当て解除、レジスタの復帰、ランタイムモデルで指定したその他のマシン固有の要件を実行するコード部分。エピローグは関数のユーザコードの後、関数リターンの直前に実行される。

エミュレーション (Emulation)

エミュレーション メモリに読み込んだソフトウェアを、あたかもマイクロコントローラに読み込んだファームウェアのように実行する事。

エミュレーション メモリ (Emulation Memory)

エミュレータに内蔵されたプログラムメモリ。

エミュレータ (Emulator)

エミュレーションを実行するハードウェア。

エミュレータ システム (Emulator System)

MPLAB ICE 2000 および MPLAB ICE 4000 の場合、エミュレータ システムはポッド、プロセッサ モジュール、デバイスアダプタ、ターゲットボード、ケーブル、MPLAB IDE ソフトウェアで構成される。MPLAB REAL ICE の場合、ポッド、ドライバ (および場合によってはレシーバ) カード、ターゲットボード、ケーブル、MPLAB IDE ソフトウェアで構成される。

エラー (Errors)

プログラムの処理を継続できない問題が発生するとエラーとして報告される。可能な場合、エラーは問題が発生したソースファイル名と行番号を特定する。

エラーファイル (Error File)

言語ツールから出力されたエラーメッセージと診断結果を記録したファイル。

演算子 (Operators)

定義可能な式を構成する際に使用される「+」や「-」等の記号。各演算子に割り当てられた優先順位に基づいて式が評価される。

エンディアン (Endianness)

マルチバイト オブジェクトにおけるバイトの並び順。

オフチップメモリ (Off-Chip Memory)

PIC18 で選択できるメモリオプション。ターゲットボード上のメモリを使用するか、または全てのプログラムメモリをエミュレータから供給する。
[Options]>[Development Mode] の順にクリックして **[Memory]** タブでオフチップメモリの選択を行う。

オブジェクト コード (Object Code)

アセンブラまたはコンパイラによって生成されるマシンコード。

オブジェクト ファイル (Object File)

マシンコードを含むファイル。デバッグ情報を含む事もある。そのまま実行できるものと、他のオブジェクト ファイル (ライブラリ等) とリンクしてから完全な実行プログラムを生成する再配置可能形式のものがある。

オブジェクト ファイル ディレクティブ (Object File Directives)

オブジェクト ファイル作成時にのみ使用するディレクティブ。

オペコード (Opcodes)

Operational Code の略。「ニーモニック」を参照。

拡張マイクロコントローラ モード (Extended Microcontroller Mode)

拡張マイクロコントローラ モードでは、内蔵プログラムメモリと外部メモリの両方が利用できる。プログラムメモリのアドレスが PIC18 の内部メモリ空間より大きい場合、自動的に外部メモリの実行に切り換わる。

拡張モード (Extended Mode)

コンパイラの動作モードの1つ。拡張命令 (ADDFSR、ADDULNK、CALLW、MOVSF、MOVSS、PUSHL、SUBFSR、SUBULNK) とリテラル オフセットによるインデックス アドレス指定を利用できる。

環境 (Environment)

IDE – アプリケーション開発に特化したデスクトップ環境。

MPLAB PM3 – デバイスのプログラミングに関する設定ファイルを保存したフォルダ。このフォルダを SD/MMC カードに転送できる。

緩和 (Relaxation)

ある命令を、機能が同じでよりサイズの小さい命令に変換する事。コードサイズを抑えるために便利である。最新の MPLAB ASM30 には、CALL 命令を RCALL 命令に緩和する機能がある。この変換は、現在の命令から +/- 32k 命令ワード以内にあるシンボルを呼び出す場合に行われる。

外部 RAM (External RAM)

オフチップの読み書き可能なメモリ。

外部シンボル (External Symbol)

外部リンケージを持つ識別子のシンボル。参照の場合と定義の場合がある。

外部シンボル解決 (External Symbol Resolution)

リンカが全ての入力モジュールの外部シンボル定義を1つにまとめ、全ての外部シンボル参照を解決しようとするプロセス。外部シンボル参照に対応する定義が存在しない場合、リンカエラーとなる。

外部入力ライン (External Input Line)

外部信号に基づいてイベントを設定するための外部入力信号ロジック プロブ ライン (TRIGIN)。

外部ラベル (External Label)

外部リンケージを持つラベル。

外部リンケージ (External Linkage)

関数や変数が、それ自身が定義されたモジュールの外部から参照できる場合、外部リンケージを持つという。

記憶域クラス (Storage Class)

指定されたオブジェクトを格納する記憶場所の持続期間を決定する。

記憶域修飾子 (Storage Qualifier)

宣言されるオブジェクトの特別な属性を示す (const 等)。

基数 (Radix)

アドレスを指定する際の記数法 (16 進法、10 進法) の底。

警告 (Warning)

IDE – デバイス、ソフトウェア ファイル、装置に物理的な損傷を与える可能性のある状況で、ユーザに注意を促すために表示されるメッセージ。

ALU30、C30 – 問題となる可能性のある状態を警告として報告するが、処理は停止されない。MPLAB C30 の警告メッセージではソースファイル名と行番号が報告されるが、エラーメッセージと区別するために「warning:」の文字列も付加される。

高級言語 (High Level Language)

プログラムを記述するための言語で、プロセッサから見てアセンブリよりも遠い位置関係にあるもの。

校正メモリ (Calibration Memory)

PIC マイクロコントローラの内蔵 RC オシレータやその他の周辺モジュールの校正値を格納するための特殊機能レジスタ。

国際標準化機構 (International Organization for Standardization)

コンピューティングや通信をはじめ、多くのテクノロジーとビジネス関連の標準規格の策定を行っている団体。

コマンドライン インターフェイス (Command Line Interface)

プログラムとユーザのやり取りをテキストの入出力だけで行う方法。

コンパイラ (Compiler)

高級言語で記述されたソースファイルをマシンコードに変換するプログラム。

コンフィグレーション ビット (Configuration Bits)

PIC マイクロコントローラの動作モードを設定するために書き込む専用ビット。コンフィグレーション ビットは事前プログラミングされている場合とされていない場合がある。

再入可能 (Reentrant)

1つの関数を複数呼び出して同時に実行できる事。直接または間接再帰、あるいは割り込み処理中の実行によって起こる事がある。

再帰 (Recursion)

定義した関数やマクロがそれ自身を呼び出す事。再帰マクロを作成する際は、再帰から抜けずに無限ループとなりやすいため注意が必要。

再帰呼び出し (Recursive Calls)

直接または間接的に自分自身を呼び出す関数。

再配置 (Relocation)

リンクが絶対アドレスを再配置可能セクションに割り当てる事。再配置可能セクション内のシンボルは全て新しいアドレスに更新される。

再配置可能 (Relocatable)

アドレスが固定されたメモリ番地に割り当てられていないオブジェクト。

再配置可能セクション (Relocatable Section)

ALU30 – アドレスが固定されていない (絶対アドレスでない) セクション。再配置可能セクションには、再配置と呼ばれるプロセスによって、リンクがアドレスを割り当てる。

左辺値 (L-value)

検査または変更が可能なオブジェクトを指し示す式。左辺値は代入演算子の左側で使用する。

シーケンス ブレークポイント (Sequenced Breakpoints)

シーケンスで発生するブレークポイント。ブレークポイントのシーケンス実行はボトムアップ方式で行われる。つまり、シーケンスの最後のブレークポイントが最初に発生する。

シェル (Shell)

MPASM アセンブラにおいて、マクロアセンブラへの入力を行うためのプロンプトインターフェイス。MPASM アセンブラには DOS 用シェルと Windows 用シェルの2種類がある。

式 (Expressions)

算術演算子または論理演算子で区切った定数または記号の組み合わせ。

識別子 (Identifier)

関数または変数の名前。

システム ウィンドウ コントロール (System Window Control)

ウィンドウや一部のダイアログの左上隅にあるコントロール。通常、このコントロールをクリックすると、[最小化]、[最大化]、[閉じる]等のメニュー項目がポップアップ表示される。

シナリオ (Scenario)

MPLAB SIM シミュレータにおいて、ステイミュラス制御を具体的に設定したもの。

シミュレータ (Simulator)

デバイスの動作をモデリングするソフトウェア プログラム。

修飾子 (Qualifier)

パスカウンタで使用する、または複合トリガにおける次の動作前のイベントとして使用するアドレスまたはアドレス範囲。

初期化済みデータ (Initialized Data)

初期値を指定して定義されたデータ。C では、

```
int myVar=5;
```

として定義した変数は初期化済みデータセクションに格納される。

シンボル (Symbol)

プログラムを構成する各種の要素を記述する汎用のメカニズム。関数名、変数名、セクション名、ファイル名、struct/enum/union タグ名等がある。MPLAB IDE では、主に変数名、関数名、アセンブリラベルをシンボルと呼ぶ。リンク実行後は、シンボルの値はメモリ内の値となる。

シンボル (絶対シンボル) (Symbol, Absolute)

アセンブリの .equ ディレクティブによる定義等、即値を表す。

実行可能コード (Executable Code)

読み込んで実行できる形式のソフトウェア。

条件付きアセンブリ (Conditional Assembly)

アセンブリ言語で、ある特定の式のアセンブル時の値に基づいて含まれたり除外されたりするコード。

条件付きコンパイル (Conditional Compilation)

プログラムの一部を、プリプロセッサ ディレクティブで指定した特定の定数式が真の場合のみコンパイルする事。

推奨しない機能 (Deprecated Features)

後方互換性のためにサポートされているだけで現在は使用されておらず、いずれ廃止される事が決まっている機能。

スキッド (Skid)

ハードウェア ブレークポイントを使ってプロセッサを停止する場合、ブレークポイントからさらに1つ以上の命令を実行してプロセッサが停止する事がある。ブレークポイントの後に実行される命令の数をスキッドと呼ぶ。

スキュー (Skew)

命令実行に関する情報は、異なる複数のタイミングでプロセッサバスに現れる。例えば、実行されるオペコードは直前の命令の実行時にフェッチとしてバスに現れる。ソースデータのアドレスと値、およびデスティネーションデータのアドレスは、オペコードが実際に実行される時にバスに現れる。デスティネーションデータの値は次の命令の実行時にバスに現れる。トレースバッファには、1インスタンスでバス

上に存在する情報がキャプチャされる。従って、トレースバッファの 1 エントリには 3 つの命令の実行情報が含まれる。1 つの命令実行で、ある情報から次の情報までにキャプチャされるサイクル数をスキューと呼ぶ。

スタック (ソフトウェア スタック) (Stack, Software)

アプリケーションがリターンアドレス、関数パラメータ、ローカル変数を保存するのに使用するメモリ。高級言語でコードを開発する場合、このメモリは通常コンパイラによって管理される。

スタック (ハードウェア スタック) (Stack, Hardware)

PIC マイクロコントローラで関数呼び出しを行う時にリターンアドレスを格納する場所。

スタティック RAM (SRAM) (Static RAM or SRAM)

Static Random Access Memory の略。ターゲットボード上の読み書き可能なプログラムメモリ。頻繁に書き換える必要のないプログラムを書き込む。

ステータスバー (Status Bar)

MPLAB IDE ウィンドウの一番下にあるバーで、カーソル位置、開発モードとデバイス、アクティブなツールバー等に関する情報が表示される。

スティミュラス (Stimulus)

シミュレータへの入力。すなわち、外部信号に対する応答をシミュレートするために生成されるデータ。通常、このデータはテキストファイルにアクションのリストとして記述される。スティミュラスの種類には、非同期、同期 (ピン)、クロック動作、レジスタがある。

ストップウォッチ (Stopwatch)

実行サイクルを測定するためのカウンタ。

制御ディレクティブ (Control Directives)

アセンブリ言語コード内で使用するディレクティブで、指定した式のアセンブル時の値に基づいてコードを含めるか除外するかを決定する。

セクション (Section)

特定のメモリ番地にあるアプリケーションの一部。

セクション属性 (Section Attribute)

セクションの特徴を表す情報 (access セクション等)。

絶対セクション (Absolute Section)

リンカによって変更されない固定 (絶対) アドレスを持つセクション。

ソースコード (Source Code)

プログラマ (人間) が記述したコンピュータ プログラム。プログラミング言語で記述されたソースコードは、マシンコードに変換して実行するか、またはインタプリタによって実行される。

ソースファイル (Source File)

ソースコードを記述した ASCII テキストファイル。

相互参照ファイル (Cross Reference File)

シンボルテーブルとそのシンボルを参照するファイルリストを参照するファイル。シンボルが定義されている場合、リストの最初のファイルがシンボル定義の位置となる。残りのファイルにはシンボルへの参照が含まれる。

属性 (Attribute)

C プログラムの変数または関数の特徴を表す情報で、マシン固有の特性を記述する目的で使用する。

属性 (セクション属性) (Attribute, Section)

「executable」、「readonly」、「data」等、セクションの特徴を表す情報。アセンブラの `.section` ディレクティブでフラグとして指定できる。

ターゲット (Target)

ユーザ ハードウェアの事。

ターゲット アプリケーション (Target Application)

ターゲットボードに読み込んだソフトウェア。

ターゲット プロセッサ (Target Processor)

ターゲット アプリケーション ボードで使用されているマイクロコントローラ デバイス。

ターゲットボード (Target Board)

ターゲット アプリケーションを構成する回路とプログラマブルなデバイス。

ダウンロード (Download)

ホストから別のデバイス (エミュレータ、プログラマ、ターゲットボード等) にデータを送信する事。

致命的エラー (Fatal Error)

コンパイルがただちに停止するようなエラー。メッセージも出力されない。

中央演算処理装置 (Central Processing Unit)

デバイス内で、実行する正しい命令をフェッチし、デコードして実行する装置。必要に応じて、算術論理演算装置 (ALU) と組み合わせて命令実行を完了する。プログラマメモリのアドレスバス、データメモリのアドレスバス、スタックへのアクセスを制御する。

ツールバー (Tool Bar)

MPLAB IDE の機能を実行するためのボタン (アイコン) を縦または横に並べたもの。

テンプレート (Template)

後でファイルに挿入するために作成するテキスト行。MPLAB エディタでは、テンプレートはテンプレートファイルに保存される。

データ ディレクティブ (Data Directives)

アセンブラによって行われるプログラムメモリまたはデータメモリの割り当てを制御するディレクティブ。データ項目をシンボル (意味のある名前) を使って参照する手段としても使われる。

データメモリ (Data Memory)

マイクロチップ社の MCU と DSC では、データメモリ (RAM) は汎用レジスタ (GPR) と特殊機能レジスタ (SFR) で構成される。EEPROM データメモリを内蔵したデバイスもある。

ディレクティブ (Directives)

言語ツールの動作を制御するためにソースコードに記述するステートメント。

デジタル シグナル コントローラ (Digital Signal Controller)

デジタル信号処理機能を搭載したマイクロコントローラ。マイクロチップ社の dsPIC DSC デバイス等。

デジタル シグナル プロセッサ (Digital Signal Processor)

デジタル信号処理用に設計されたマイクロプロセッサ。

デジタル信号処理 (Digital Signal Processing)

デジタル信号をコンピュータで処理する事。通常は、アナログ信号 (音声または画像) をデジタル形式に変換 (サンプリング) して処理する事をいう。

デバイス プログラマ (Device Programmer)

マイクロコントローラ等、電気的に書き込み可能な半導体デバイスにプログラミングを行うためのツール。

デバッガ (Debugger)

デバッグを実行するハードウェア。

デバッガシステム (Debugger System)

デバッガ本体、プロセッサ モジュール、デバイスアダプタ、ターゲットボード、ケーブル、MPLAB IDE ソフトウェアで構成されるシステム。

デバッグ情報 (Debugging Information)

コンパイラとアセンブラでこのオプションを選択すると、アプリケーションコードのデバッグに使える各種レベルの情報が出力される。デバッグ オプションの選択の詳細はコンパイラまたはアセンブラのマニュアルを参照。

特殊機能レジスタ (Special Function Registers)

I/O プロセッサ機能、I/O ステータス、タイマ等の各種モードや周辺モジュールを制御するレジスタ専用使用するデータメモリ (RAM) 領域。

匿名構造体 (Anonymous Structure)

C30 – 名前のない構造体。

C18 – C 共用体のメンバーで、名前のない構造体。匿名構造体のメンバーは、その構造体を包含している共用体のメンバーと同じようにアクセスできる。例えば以下のサンプルコードでは、hi と lo は共用体 `caster` に含まれる匿名構造体のメンバーである。

```
union castaway
{
    int intval;
    struct {
        char lo; //accessible as caster.lo
        char hi; //accessible as caster.hi
    };
} caster;
```

トライグラフ (Trigraphs)

「??」で始まる 3 文字のシーケンス。ISO C で定義されており、1 つの文字に置換される。

トリガ出力 (Trigger Output)

任意のアドレスまたはアドレス範囲で生成でき、トレースおよびブレイクポイント設定から独立したエミュレータ出力信号の事。トリガ出力ポイントはいくつでも設定できる。

トレース (Trace)

プログラム実行のログを記録するエミュレータまたはシミュレータの機能。エミュレータはプログラム実行のログをトレースバッファに記録し、これを MPLAB IDE のトレース ウィンドウにアップロードする。

トレースマクロ (Trace Macro)

エミュレータ データからのトレース情報を提供するマクロ。これはソフトウェアトレースのため、トレースを利用するには、マクロをコードに追加し、コードを再コンパイルまたは再アセンブルし、ターゲット デバイスにこのコードをプログラムする必要がある。

トレースメモリ (Trace Memory)

エミュレータに内蔵されたトレース用のメモリ。トレースバッファとも呼ばれる。

内部リンケージ (Internal Linkage)

関数や変数が、それ自身が定義されたモジュールの外部からアクセスできない場合、内部リンケージを持つという。

ニーモニック (Mnemonics)

マシンコードと 1 対 1 で対応したテキスト命令。オペコードとも呼ばれる。

ネイティブ データサイズ (Native Data Size)

ネイティブ トレースの場合、[Watch] ウィンドウで使用する変数のサイズは選択したデバイスのデータメモリと同じサイズ (PIC18 の場合は同じバイトサイズ、16 ビットデバイスの場合は同じワードサイズ) である必要がある。

ノード (Node)

MPLAB IDE のプロジェクトを構成するコンポーネント。

パスカウンタ (Pass Counter)

イベント (特定のアドレスの命令を実行する等) が発生するたびに値をデクリメントするカウンタ。パスカウンタの値がゼロになると、イベントの条件が満たされる。パスカウンタはブレイクログック、トレースログック、複合トリガダイアログの任意のシーケンシャル イベントに割り当てられる。

パワーオン リセット エミュレーション (Power-on-Reset Emulation)

データ RAM 領域にランダムな値を書き込んで、初回電源投入時の RAM の非初期化値をシミュレートするソフトウェア ランダム化処理。

ヒープ (Heap)

動的メモリ割り当てに使われるメモリ領域。メモリブロックの割り当てと解放は実行時に任意の順序で行われる。

非拡張モード (Non-Extended Mode)

コンパイラの動作モードの 1 つ。拡張命令またはリテラル オフセットによるインデックス アドレス指定を利用しない。

非初期化データ (Uninitialized Data)

初期値なしで定義されたデータ。C では、

```
int myVar;
```

として定義した変数は非初期化データ セクションに格納される。

非同期 (Asynchronously)

複数のイベントが同時には発生しない事。一般に、プロセッサ実行中の任意の時点で発生する割り込みに言及する際に使う。

非同期スティミュラス (Asynchronous Stimulus)

シミュレータ デバイスへの外部入力をシミュレートするために生成されるデータ。

非リアルタイム (Non Real Time)

ブレークポイントで停止中、またはシングルステップ実行中のプロセッサ、あるいはシミュレータ モードで動作中の MPLAB IDE を指す。

ビルド (Build)

全てのソースファイルをコンパイルおよびリンクしてアプリケーションを作成する事。

ファイル レジスタ (File Registers)

汎用レジスタ (GPR) と特殊機能レジスタ (SFR) で構成される内蔵のデータメモリ。

ファントムバイト (Phantom Byte)

dsPIC アーキテクチャで、24 ビット命令ワードを 32 ビット命令ワードと見なし扱った場合に使用する未実装バイト。dsPIC の hex ファイルに見られる。

フィルタ (Filter)

トレース ディスプレイまたはデータファイルにどのデータを含めるか / 除外するかを選択するもの。

不揮発性ストレージ (Non-Volatile Storage)

電源を OFF にしても内容が失われないストレージ デバイス。

フラッシュ (Flash)

データの書き込みと消去をバイト単位ではなくブロック単位で行えるタイプの EEPROM。

フリースタANDING (Free-Standing)

複素数型を使っておらず、ライブラリ (ANSI C89 規格第 7 節) で規定する機能の使用が標準ヘッダ (<float.h>、<iso646.h>、<limits.h>、<stdarg.h>、<stdbool.h>、<stddef.h>、<stdint.h>) の内容のみに限定されている厳密な規格合致プログラムを受理する処理系。

フレームポインタ (Frame Pointer)

スタックベースの引数とスタックベースのローカル変数の境界となるスタック番地を指し示すポインタ。ここを基準にすると、現在の関数のローカル変数やその他の値に容易にアクセスできる。

ブックマーク (Bookmarks)

ファイル内の特定の行に簡単な操作でアクセスできるようにする機能。

ブックマークの有効 / 無効の切り替え、前後のブックマークへの移動、すべてのブックマークのクリア等、ブックマークの管理は [Edit] メニューで [Bookmarks] をクリックして行う。

ブレークポイント (Breakpoint)

ハードウェア ブレークポイント : 実行するとファームウェアの実行が停止するイベント。
ソフトウェア ブレークポイント : ファームウェアの実行が停止するアドレス。通常、特別な Break 命令によって実行が停止される。

プラグイン (Plug-ins)

MPLAB IDE では、標準コンポーネントにプラグイン モジュールを追加する事によって、各種ソフトウェア / ハードウェア ツールに対応する。一部のプラグイン ツールは、[Tools] メニューから利用できる。

プリAGMA (Pragma)

特定のコンパイラにとって意味を持つディレクティブ。一般に、実装で定義した情報をコンパイラに伝達するために使われる。MPLAB C30 は属性を利用してこの情報を伝達する。

プログラム カウンタ (Program Counter)

現在実行中の命令のアドレスを格納した場所。

プログラム カウンタ ユニット (Program Counter Unit)

ALU30 – プログラムメモリのレイアウトを概念的に表現したもの。プログラムカウンタは 1 命令ワードで 2 つインクリメントする。実行可能セクションでは、2 プログラムカウンタ ユニットは 3 バイトに相当する。読み出し専用セクションでは、2 プログラムカウンタ ユニットは 2 バイトに相当する。

プログラムメモリ (Program Memory)

IDE – デバイス内で命令が保存されるメモリ領域。また、エミュレータまたはシミュレータにダウンロードしたターゲット アプリケーションのファームウェアを格納するメモリ領域もプログラムメモリと呼ばれる。

ALU30/C30 – デバイス内で命令が保存されるメモリ領域。

プロジェクト (Project)

アプリケーションのビルドに必要なファイル (ソースコードやリンカ スクリプト ファイル等)一式と、各種ビルドツールやビルドオプションとの関連付けをまとめたもの。

プロトタイプ システム (Prototype System)

ユーザのターゲット アプリケーションまたはターゲットボードの事。

プロファイル (Profile)

MPLAB SIM シミュレータにおいて、実行したスティミュラスをレジスタ別に一覧表示したもの。

プロローグ (Prologue)

コンパイラで生成したコードのうち、スタック領域の割り当て、レジスタの退避、ランタイムモデルで指定したその他のマシン固有の要件を実行するコード部分。プロローグは、関数のユーザコードの前に実行される。

ベクタ (Vector)

リセットまたは割り込みが発生した時にアプリケーションのジャンプ先となるメモリ番地。

ホスト PC (PC Host)

サポートされた Windows オペレーティング システムが動作する PC。

ポッド (Pod)

MPLAB REAL ICE システムの場合：ヘッダボードまたはターゲットボードに装着された ICE デバイス用のエミュレーション制御回路を内蔵。ICE デバイスとは、ICE 回路を内蔵した標準デバイス、または標準デバイスの特別な ICE バージョン (デバイス名 -ICE) を指す。

MPLAB ICD の場合：ヘッダボードまたはターゲットボードに装着された ICD デバイス用のデバッグ制御回路を内蔵。ICD デバイスとは、ICD 回路を内蔵した標準デバイス、または標準デバイスの特別な ICD バージョン (デバイス名 -ICD) を指す。

MPLAB ICE 2000/4000 の場合：エミュレーション メモリ、トレースメモリ、イベントおよびサイクルタイム、トレース / ブレークポイント ロジックを含む外付けのエミュレータ本体。

マイクロコントローラ (Microcontroller)

CPU、RAM、プログラムメモリ、I/O ポート、タイマ等、多くの機能が統合されたチップ。

マイクロコントローラ モード (Microcontroller Mode)

PIC18 マイクロコントローラで設定可能なプログラムメモリ構成の 1 つ。マイクロコントローラ モードでは、内部実行のみが許可される。つまり、マイクロコントローラ モードでは内蔵プログラムメモリしか利用できない。

マイクロプロセッサ モード (Microprocessor Mode)

PIC18 マイクロコントローラで設定可能なプログラムメモリ構成の 1 つ。マイクロプロセッサ モードでは、内蔵プログラムメモリは使用されない。プログラムメモリ全体が外部にマッピングされる。

マクロ (Macro)

マクロ命令。一連の命令シーケンスを短い名前 で表現した命令。

マクロ ディレクティブ (Macro Directives)

マクロ定義の中で実行とデータ割り当てを制御するディレクティブ。

マシンコード (Machine Code)

コンピュータ プログラムをプロセッサが実際に読み出して解釈できる形式で表現したもの。2進数のマシンコードで記述されたプログラムは、マシン命令のシーケンス (命令間にデータを挟むこともある) で構成される。ある特定のプロセッサで使用できる全ての命令の集合を「命令セット」という。

マシン語 (Machine Language)

ある CPU が翻訳を必要とせず実行できる命令の集合。

未割り当てセクション (Unassigned Section)

リンカのコマンドファイルで特定のターゲット メモリ ブロックに割り当てられていないセクション。リンカは、未割り当てセクションを割り当てるターゲット メモリ ブロックを検出する必要がある。

命令 (Instructions)

CPU に対して特定の演算を実行するように指示するビット列。演算の対象となるデータを含める事もできる。

命令セット (Instruction Set)

特定のプロセッサが理解できるマシン語命令の集合。

メッセージ (Message)

言語ツールの動作に問題が発生したことを知らせる文字列。メッセージが表示されても処理は停止しない。

メモリモデル (Memory Model)

C30 – アプリケーションで利用可能なメモリを表現したもの。

C18 – プログラムメモリを指し示すポインタのサイズに関する規定を記述したもの。

優先順位 (Precedence)

式の評価順を定義した規則。

読み出し専用メモリ (Read Only Memory)

恒久的に保存されているデータへの高速アクセスが可能なメモリ ハードウェア。ただし、データの追加や変更は不可。

ライブラリ (Library)

「アーカイブ」を参照。

ライブラリアン (Librarian)

「アーカイバ」を参照。

ランタイムモデル (Run-time Model)

ターゲット アーキテクチャのリソースの使用を記述したもの。

リアルタイム (Real Time)

インサーキット エミュレータまたはデバッガが Halt 状態から解放されると、プロセッサの実行はリアルタイム モードとなり、通常のチップと全く同じ挙動をする。リアルタイム モードでは、エミュレータのリアルタイム トレース バッファが有効になり、選択した全てのサイクルを常時キャプチャする。また、全てのブレーク ロジックが有効になる。インサーキット エミュレータまたはデバッガでは、有効なブレークポイントで停止するか、またはユーザによって実行が停止されるまでプロセッサはリアルタイム で動作する。

シミュレータでは、ホスト CPU でシミュレート可能な最大速度でマイクロコントローラの命令を実行する事をリアルタイムと呼ぶ。

リアルタイム ウォッチ (Real-Time Watch)

アプリケーション実行中に [Watch] ウィンドウで変数の値がリアルタイムに変化する事。リアルタイム ウォッチの設定方法は、各ツールのマニュアルを参照。リアルタイム ウォッチをサポートしていないツールもある。

リスティング ディレクティブ (Listing Directives)

アセンブラのリスティング ファイルのフォーマットを制御するディレクティブ。タイトルや改ページ指示等、リスティング ファイルに関する各種の設定を行う。

リスティング ファイル (Listing File)

ソースファイルにある各 C ソース ステートメント、アセンブリ命令、アセンブラ ディレクティブ、マクロに対して生成されたマシン コードを記述した ASCII テキストファイル。

リトル エンディアン (Little Endian)

マルチバイト データで最下位バイト (LSB) を最下位アドレスに格納するデータ並び順方式。

量産プログラマ (Production Programmer)

デバイスのプログラムを高速に行えるようにリソースを強化したプログラマ。各種電圧レベルでのプログラミングに対応し、プログラミング仕様にも完全に準拠している。量産環境ではアプリケーション回路が組立ラインにとどまる時間をなるべく短くする必要があるため、デバイスへの書き込み時間が特に重視される。

MPLAB PM3、MPLAB REAL ICE インサーキット エミュレータ、MPLAB ICD 3 等、マイクロチップ社の量産プログラマはこうした要求の厳しい環境にも耐えられるよう特に堅牢な設計が採用されている。

一部のハイエンドツールには、さらに専用のアクセサリが用意されている。例えば MPLAB REAL ICE には、通信と ICSP (インサーキット シリアル プログラミング) のプロセスをさらに高速化する Performance Pak が提供されている。MPLAB PM3 プログラマには、各種デバイスを直接サポートするための交換可能ソケットモジュールが提供されている。

リンカ (Linker)

オブジェクト ファイルとライブラリを結合し、モジュール間の参照を解決して実行可能コードを生成する言語ツール。

リンカ スクリプト ファイル (Linker Script Files)

リンカのコマンド ファイル。リンカのオプションを定義し、ターゲット プラットフォームで利用可能なメモリを記述する。

ループバック テスト ボード (Loop-Back Test Board)

MPLAB REAL ICE インサーキット エミュレータの動作をテストするために用いる。

レイテンシ (Latency)

イベントが発生してからその応答までの時間の長さ。

ロー (raw) データ (Raw Data)

あるセクションに関連付けられたコードまたはデータを 2 進数で表現したもの。

ローカルラベル (Local Label)

マクロ内で LOCAL ディレクティブで定義されたラベル。ローカルラベルは、マクロの同一インスタンス内でのみ有効。すなわち、LOCAL として宣言されたシンボルやラベルには、ENDM マクロ以降はアクセスできない。

ロジックプローブ (Logic Probes)

マイクロチップ社製エミュレータには、最大 14 のロジックプローブを接続できるものがある。ロジックプローブは、外部トレース入力、トリガ出力信号、+5 V、共通グラウンドを提供する。

ワークスペース (Workspace)

MPLAB IDE において、デバイスやデバッグツール / プログラムの選択内容、起動中のウィンドウとその位置等、IDE の各種設定を保存したもの。

ワークブック (Workbook)

MPLAB SIM シミュレータにおいて、SCL スティミュラスの生成に関する設定を保存したもの。

割り当てセクション (Assigned Section)

リンカのコマンドファイルで特定のターゲット メモリ ブロックに割り当てられたセクション。

割り込み (Interrupt)

CPU に対する信号の一種。この信号が発生すると、現在動作中のアプリケーションの実行を一時停止し、制御を割り込みサービスルーチン (ISR) に渡してイベントを処理する。ISR の実行が完了すると、通常の実行が再開される。

割り込みサービスルーチン (Interrupt Service Routine)

ALU30、C18、C30 – 割り込みを処理する関数。

IDE – 割り込みが発生すると実行されるユーザ作成コード。通常、発生した割り込みの種類によってプログラム メモリ内の異なる位置のコードが実行される。

割り込みハンドラ (Interrupt Handler)

割り込み発生時に専用のコードを実行するルーチン。

割り込みベクタ (Interrupt Vector)

割り込みサービスルーチンまたは割り込みハンドラのアドレス。

割り込み要求 (Interrupt Request)

プロセッサの通常の命令実行を一時的に停止し、割り込みハンドラルーチンの実行開始を要求するイベント。プロセッサによっては複数の割り込み要求イベントを持ち、優先度の異なる割り込みを処理できるものもある。

索引

数字		PK3Err0006	49
44 ピン デモボード	33	PK3Err0007	49
A		PK3Err0008	49
[Abort Operation]	58, 63	PK3Err0009	49
[Animate]	57	PK3Err0010	49
B		PK3Err0011	50
[Blank Check]	63	PK3Err0012	50
[Breakpoints]		PK3Err0013	50
設定	58	PK3Err0014	50
ダイアログ	59	PK3Err0015	50
有効化	59	PK3Err0016	50
[Build Configuration]	24, 43	PK3Err0017	50
C		PK3Err0018	50
CD-ROM	16	PK3Err0019	50
CodeGuard セキュリティ	67	PK3Err0020	50
D		PK3Err0021	50
Debug Express	33	PK3Err0022	50
[Debug Read]	58	PK3Err0023	50
[Download Firmware]	65	PK3Err0024	50
E		PK3Err0025	50
[Erase]	63	PK3Err0026	50
[Erase Flash Device]	58	PK3Err0027	50
F		PK3Err0028	51
[Freeze on Halt]	46	PK3Err0029	51
H		PK3Err0030	51
[Halt]	57	PK3Err0031	51
I		PK3Err0032	51
ICD ヘッダ	16	PK3Err0033	51
ICSP	22, 23, 24, 71	PK3Err0034	51
ICSPCLK	71	PK3Err0035	51
ICSPDAT	71	PK3Err0036	51
L		PK3Err0037	51
LED	15, 70	PK3Err0038	51
M		PK3Err0039	51
MPLAB IDE	25	PK3Err0040	51
P		PK3Err0041	52
PC、省電力	46, 70	PK3Err0043	52
PGC	19, 20, 21, 22, 23	PK3Err0044	52
PGD	19, 20, 21, 22, 23	PK3Err0045	52
PICKIT 3 Debug Express Kit	16	PK3Err0046	52
PICKIT 3 について	13	PK3Err0047	52
PICKIT 3 の構成要素	16	PK3Err0052	52
PIM	18	PK3Err0053	52
PK3Err0001	49	PK3Err0054	52
PK3Err0002	49	PK3Err0055	52
PK3Err0003	49	PK3Err0056	52
		PK3Err0063	52
		PK3Err0068	52
		PK3Err0069	52
		PK3Err0070	52
		PK3Err0071	52
		PK3Err0072	52
		PK3Err0073	53
		PK3Err0075	53
		[Program]	58, 63

PICKit™ 3 ユーザガイド

R

[Read]	58, 63
Readme	8
[Reconnect]	58, 63
[Reset]	
プロセス	58

S

[Secure Segment]	67
------------------------	----

U

USB	70, 82
ケーブル	16
ハブ	70
USB ポート	14

V

Vcap	20
Vdd	19, 20, 21, 22
[Verify]	63
Vpp	19, 20, 21, 22, 23
Vss	19, 20, 21, 22

い

一般的な対処方法	53
インジケータ ライト	70
インターネット アドレス、マイクロチップ社	9

う

ウェブサイト、マイクロチップ社	9
ウォッチドッグ タイマ	22, 83

お

お客様向け通知サービス	9
-------------------	---

か

カスタマ サポート	10
-----------------	----

き

休止モード	46, 70
-------------	--------

け

ケーブル	
長さ	70, 73

こ

コード保護	22
コネクタ、6 ピン	14
コンデンサ	20, 21
コンフィグレーション ビット	22, 30

さ

参考資料、推奨	8
---------------	---

し

省電力モード	46, 70
--------------	--------

す

ステップ	57
------------	----

た

ターゲット デバイス	22
ターゲットとの接続	
回路	19
標準	19
不適切な回路	21
耐久性、カードガイド	70

て

抵抗	21
テーブル読み出し保護	22
デバイスによる予約リソース	24
デバッグの正常動作を妨げる回路	21
デバッグ	
実行プログラム	23
レジスタ	24
デバッグに失敗する主な理由	43
デバッグモード	
操作手順	23

と

ドライバボード	
標準	71

は

ハブ、USB	70
--------------	----

ひ

標準 ICSP による通信	17
標準通信	
接続	19
ドライバボード	71

ふ

ファームウェアのダウンロード	65
プルアップ	21
プロジェクト ウィザード	30
プロセスサ拡張キット	16

へ

変換ソケット	16
仕様	26

ほ

保証登録	8
本書	
構成	6
表記	7

も

モジュラ インターフェイス ケーブル	22
--------------------------	----

NOTES:

各国の営業所とサービス

北米

本社

2355 West Chandler Blvd.
Chandler, AZ 85224-6199

Tel: 480-792-7200

Fax: 480-792-7277

技術サポート:

<http://www.microchip.com/support>

URL:

www.microchip.com

アトランタ

Duluth, GA

Tel: 678-957-9614

Fax: 678-957-1455

ボストン

Westborough, MA

Tel: 774-760-0087

Fax: 774-760-0088

シカゴ

Itasca, IL

Tel: 630-285-0071

Fax: 630-285-0075

クリーブランド

Independence, OH

Tel: 216-447-0464

Fax: 216-447-0643

ダラス

Addison, TX

Tel: 972-818-7423

Fax: 972-818-2924

デトロイト

Farmington Hills, MI

Tel: 248-538-2250

Fax: 248-538-2260

インディアナポリス

Noblesville, IN

Tel: 317-773-8323

Fax: 317-773-5453

ロサンゼルス

Mission Viejo, CA

Tel: 949-462-9523

Fax: 949-462-9608

サンタクララ

Santa Clara, CA

Tel: 408-961-6444

Fax: 408-961-6445

トロント

Mississauga, Ontario,
Canada

Tel: 905-673-0699

Fax: 905-673-6509

アジア / 太平洋

アジア太平洋支社

Suites 3707-14, 37th Floor

Tower 6, The Gateway

Harbour City, Kowloon

Hong Kong

Tel: 852-2401-1200

Fax: 852-2401-3431

オーストラリア - シドニー

Tel: 61-2-9868-6733

Fax: 61-2-9868-6755

中国 - 北京

Tel: 86-10-8569-7000

Fax: 86-10-8528-2104

中国 - 成都

Tel: 86-28-8665-5511

Fax: 86-28-8665-7889

中国 - 重慶

Tel: 86-23-8980-9588

Fax: 86-23-8980-9500

中国 - 杭州

Tel: 86-571-2819-3187

Fax: 86-571-2819-3189

中国 - 香港 SAR

Tel: 852-2401-1200

Fax: 852-2401-3431

中国 - 南京

Tel: 86-25-8473-2460

Fax: 86-25-8473-2470

中国 - 青島

Tel: 86-532-8502-7355

Fax: 86-532-8502-7205

中国 - 上海

Tel: 86-21-5407-5533

Fax: 86-21-5407-5066

中国 - 瀋陽

Tel: 86-24-2334-2829

Fax: 86-24-2334-2393

中国 - 深圳

Tel: 86-755-8203-2660

Fax: 86-755-8203-1760

中国 - 武漢

Tel: 86-27-5980-5300

Fax: 86-27-5980-5118

中国 - 西安

Tel: 86-29-8833-7252

Fax: 86-29-8833-7256

中国 - 厦門

Tel: 86-592-2388138

Fax: 86-592-2388130

中国 - 珠海

Tel: 86-756-3210040

Fax: 86-756-3210049

アジア / 太平洋

インド - バンガロール

Tel: 91-80-3090-4444

Fax: 91-80-3090-4123

インド - ニューデリー

Tel: 91-11-4160-8631

Fax: 91-11-4160-8632

インド - プネ

Tel: 91-20-2566-1512

Fax: 91-20-2566-1513

日本 - 大阪

Tel: 81-66-152-7160

Fax: 81-66-152-9310

日本 - 横浜

Tel: 81-45-471-6166

Fax: 81-45-471-6122

韓国 - 大邱

Tel: 82-53-744-4301

Fax: 82-53-744-4302

韓国 - ソウル

Tel: 82-2-554-7200

Fax: 82-2-558-5932 または

82-2-558-5934

マレーシア - クアラルンプール

Tel: 60-3-6201-9857

Fax: 60-3-6201-9859

マレーシア - ペナン

Tel: 60-4-227-8870

Fax: 60-4-227-4068

フィリピン - マニラ

Tel: 63-2-634-9065

Fax: 63-2-634-9069

シンガポール

Tel: 65-6334-8870

Fax: 65-6334-8850

台湾 - 新竹

Tel: 886-3-5778-366

Fax: 886-3-5770-955

台湾 - 高雄

Tel: 886-7-536-4818

Fax: 886-7-330-9305

台湾 - 台北

Tel: 886-2-2500-6610

Fax: 886-2-2508-0102

タイ - バンコク

Tel: 66-2-694-1351

Fax: 66-2-694-1350

ヨーロッパ

オーストリア - ヴェルス

Tel: 43-7242-2244-39

Fax: 43-7242-2244-393

デンマーク - コペンハーゲン

Tel: 45-4450-2828

Fax: 45-4485-2829

フランス - パリ

Tel: 33-1-69-53-63-20

Fax: 33-1-69-30-90-79

ドイツ - ミュンヘン

Tel: 49-89-627-144-0

Fax: 49-89-627-144-44

イタリア - ミラノ

Tel: 39-0331-742611

Fax: 39-0331-466781

オランダ - ドリユーン

Tel: 31-416-690399

Fax: 31-416-690340

スペイン - マドリッド

Tel: 34-91-708-08-90

Fax: 34-91-708-08-91

イギリス - ウォーキングム

Tel: 44-118-921-5869

Fax: 44-118-921-5820