

MPLAB® ICE 4 インサーキット エミュレータ ユーザガイド

開発ツールユーザへの注意

Important:

どのような文書でも内容は時間が経つにつれ古くなります。本書も例外ではありません。Microchip 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の説明とは異なる場合があります。最新 PDF 文書は Microchip 社のウェブサイト (www.microchip.com/)をご覧ください。

文書は各ページ下部に表記している「DS」番号で識別します。DS 番号のフォーマットは DS<文書番号> <リビジョン>です。<文書番号>は8桁の番号、<リビジョン>はアルファベットの大文字です。

最新情報はツールのヘルプ(onlinedocs.microchip.com/)をご覧ください。



目次

1. 序章 4 1.1 本書の表記規則 4 1.2 推奨参考資料 5 2. 本工ミュレータについて 6 2.1 強み 6 2.2 構成要素 7 2.3 ブロック図 8 2.4 MPLAB ICE 4とMPLAB X IDE 8 3.1 電源とセルフテスト 9 3.2 PCの接続 10 3.3 ターゲットとの接続 10 3.3 ターゲットとの接続 10 3.3 ターゲットとの接続 10 4.1 MPLAB X IDEでのデバッグ 37 4.1 MPLAB X IDEでのデバッグ 37 4.3 PIC MCU/dsPIC DSC - オンチップデバッグ) 37 4.3 PIC MCU/dsPIC DSC - オンチップデバッグ 45 5. 本エミュレータの特長 53 5.1 USB CDC仮想とCOMボート 53 5.2 DGI 53 5.3 データキャプチャどランタイム ウォッチ - PIC MCUとdsPIC DSC 57 5.4 CI/CDのサボート 58 5.5 PIC MCUとdsPIC DSCの計測機能付き トレース 59 5.6 PIC32M MCUの命令 トレース 59 5.7 SAM ITM/SWO ト レース 59 5.8 PCサンブリング - 16ビットPIC MCUのみ 73 5.9 PCプロファイリング - 32ビットMCUのみ 74 5.10 デバッガ ボーリング 76 5.11 Power Monitor	開多	^後 ツール	レユーザへの注意	1
1.1 本書の表記規則 4 1.2 推奨参考資料 5 2. 本工ミュレータについて. 6 2.1 強み 6 2.2 構成要素 7 2.3 ブロック図 8 2.4 MPLAB ICE 4 とMPLAB X IDE 8 3.1 電源とセルフテスト 9 3.2 PCの接続 10 3.3 ターゲットとの接続 15 4.1 MPLAB X IDEでのデパッグ 37 4.2 ATデバイス - OCD(オンチッブデバッグ) 37 4.3 PIC MCU/dsPIC DSC - オンチッブデバッグ 45 5. 本工ミュレータの特長 53 5.1 USB CDC仮想とCOMポート 53 5.2 DGI 53 5.3 データキャブチャとランタイム ウォッチ - PIC MCUとdsPIC DSC 57 5.4 CI/CDのサポート 58 5.5 PIC MCU dsPIC DSCの計測機能村き トレース 59 5.6 PIC 32M MCUの命令 トレース 59 5.7 SAM ITM/SWOトレース 59 5.8 PC ケンブリング - 16ビッ PIC MCUのみ 73 5.9 PC プロアイリング - 32ビットMCUのみ 74 5.10 デバッガ	1.	序章		4
1.2 推奨参考資料 .5 2. 本工ミュレータについて .6 2.1 強み .6 2.2 構成要素 .7 2.3 ブロック図 .8 2.4 MPLAB ICE 4とMPLAB X IDE .8 3.1 電源とセルフテスト .9 3.1 電源とセルフテスト .9 3.1 電源とセルフテスト .9 3.2 PCの接続 .10 3.3 ターゲットとの接続 .15 4.1 MPLAB X IDEでのデパッグ .37 4.2 ATデバイス - OCD(オンチッブデバッグ) .37 4.3 PIC MCU/dsPIC DSC - オンチッブデバッグ) .37 4.3 PIC MCU/dsPIC DSC - オンチッブデバッグ .45 5. 本工ミュレータの特長 .53 5.1 USB CDC仮想SCOMボート .53 5.2 DGI .53 5.3 データキャブチャとランタイム ウォッチ - PIC MCUとdsPIC DSC .57 5.4 CI/CDのサポート .58 5.5 PIC MCU dsPIC DSC の計測機能付き トレース .58 5.6 PIC32M MCUの命令トレース .67 5.8 PCサンブリング - 16ビットPIC MCUのみ .73 5.9 PC プロファイリング - 32ビットMCUのみ .74 5.1 TPower Monitor .77 6.1 最初に確認する項目 .85 6.2 デバッグに失敗する主な理由 .85 6.3 一般的注意事項 .86		1.1	本書の表記規則	4
2. 本エミュレータについて		1.2		5
2 本エミュレータについて 6 2.1 強み 6 2.2 構成要素 7 2.3 ブロック図 8 2.4 MPLAB ICE 4とMPLAB X IDE 8 3.1 電源とセルフテスト 9 3.2 PCの接続 10 3.3 ターゲットとの接続 11 4.1 MPLAB X IDEでのデバッグ 37 4.2 ATデバイス - OCD(オンチップデバッグ) 37 4.3 PIC MCU/dsPIC DSC - オンチップデバッグ 45 5. 本エミュレータの特長 53 5.1 USE CDC仮想COMボート 53 5.2 DGI 53 5.3 データキャプチャとランタイム ウォッチ - PIC MCUとdsPIC DSC 57 5.4 CI/CDのサポート 58 5.5 PIC MCUとdsPIC DSCの計測機能付きトレース 59 5.6 PIC32M MCUの命令トレース 67 5.8 PC プロファイリング - 16ビットPIC MCUのみ 74 5.10 FC コファイリング - 32ビットMCUのみ 74 5.10 Fパッガボーリング 74 5.10 Fパッガボーリング 75 5.11 Power Monitor 77 6.1 最初				
2.1 強み 6 2.2 構成要素 7 2.3 ブロック図 8 2.4 MPLAB ICE 4とMPLAB X IDE 8 2.4 MPLAB ICE 4とMPLAB X IDE 8 3. 接続 9 3.1 電源とセルフテスト 9 3.2 PCの接続 10 3.3 ターゲットとの接続 15 4. 動作 37 4.1 MPLAB X IDEでのデバッグ 37 4.2 ATデパイス - OCD(オンチッブデバッグ) 37 4.3 PIC MCU/dsPIC DSC - オンチッブデバッグ 45 5. 本エミュレータの特長 53 5.1 USB CDC仮想COMポート 53 5.2 DGI 53 5.3 データキャブチャとランタイム ウォッチ - PIC MCUとdsPIC DSC 57 5.4 CI/CDのサポート 53 5.5 PIC MCUとdsPIC DSCの計測機能付きトレース 59 5.6 PIC32M MCUの命令トレース 65 5.7 SAM ITM/SWOトレース 65 5.8 PCサンブリング - 16ビッ PIC MCUのみ 73 5.9 PCプロファイリング - 32ビッ トMCUのみ 76 5.10 デバッガボーリング 70 5.1 最初に確認する項目 65 6.2 デバッグに失敗する主な理由 65 6.3 一般的注意事項 66 6.4 ハードウェアシルエマージェンシーブートファームウェアリカバリ 1 1.1 動作に関するFAQ	2.	本エミ	シュレータについて	6
2.2 構成要素 / 2.3 ブロック図 .8 2.4 MPLAB ICE 4とMPLAB X IDE .8 3. 接続 .9 3.1 電源とセルフテスト .9 3.2 PCの接続 .10 3.3 ターゲットとの接続 .15 4. 動作		2.1	通み	6
2.3 フロック図		2.2	構成要素	7
2.4 MPLAB ICE 4 & MPLAB X IDE		2.3	フロック図	8
3. 接続 9 3.1 電源とセルフテスト 9 3.2 PCの接続 10 3.3 ターゲットとの接続 15 4. 動作 37 4.1 MPLAB X IDEでのデパッグ 37 4.2 ATデパイス - OCD(オンチップデパッグ) 37 4.3 PIC MCU/dsPIC DSC - オンチップデパッグ) 37 4.3 PIC MCU/dsPIC DSC - オンチップデパッグ 45 5. 本エミュレータの特長 53 5.1 USB CDC仮想COMボート 53 5.2 DGI 53 5.3 データキャプチャとランタイム ウォッチ - PIC MCUとdsPIC DSC 57 5.4 Cl/CDのサポート 58 5.5 PIC MCUとdsPIC DSCの計測機能付きトレース 58 5.6 PIC32M MCUの命令トレース 57 5.7 SAM ITM/SWOトレース 67 5.8 PCサンプリング - 16ビットPIC MCUのみ 73 5.9 PC プロファイリング - 32ビットMCUのみ 74 5.10 デパッガポーリング 76 5.11 Power Monitor 77 6. トラブルシュートの最初のステップ 85 6.1 最初に確認する項目 85 6.2 デパッグに失敗する主な理由 85 6.3 一般的注意事項 86 6.4 ハードウェア ツール エマージェンシー ブート ファームウェア リカバリ ユーティ リティ の使い方 86 7.4 動作に関引 るFAQ 88		2.4	MPLAB ICE 4 & MPLAB X IDE	8
3.1<電源とセルフテスト	3.	接続		9
3.2 PCの接続 10 3.3 ターゲットとの接続 15 4. 動作 37 4.1 MPLAB X IDEでのデバッグ 37 4.2 ATデバイス - OCD(オンチップデバッグ) 37 4.3 PIC MCU/dsPIC DSC - オンチップデバッグ 45 5. 本エミュレータの特長 53 5.1 USB CDC仮想COMボート 53 5.2 DGI 53 5.3 データキャプチャとランタイム ウォッチ - PIC MCUとdsPIC DSC 57 5.4 CI/CDのサポート 53 5.5 PIC MCUとdsPIC DSCの計測機能付きトレース 58 5.6 PIC32M MCUの命令トレース 57 5.7 SAM ITM/SWO トレース 67 5.8 PCサンブリング - 16ビットPIC MCUのみ 73 5.9 PC プロファイリング - 32ビットMCUのみ 74 5.10 デバッガボーリング 76 5.11 Power Monitor 77 6. トラブルシュートの最初のステップ 85 6.1 最初に確認する項目 85 6.3 一般的注意事項 86 6.4 ハードウェア ツール エマージェンシー ブート ファームウェアームウェア・ジェンシー ブート ファームウェア・リカバリ ューティリティの使い方 86 7.1 動作に関い方		3.1	電源とセルフテスト	9
3.3 ターゲットとの接続 15 4. 動作 37 4.1 MPLAB X IDEでのデバッグ 37 4.2 ATデバイス - OCD(オンチッブデバッグ) 37 4.3 PIC MCU/dsPIC DSC - オンチッブデバッグ 45 5. 本エミュレータの特長 53 5.1 USB CDC仮想COMポート 53 5.2 DGI 53 5.3 データキャブチャとランタイム ウォッチ - PIC MCUとdsPIC DSC 57 5.4 CI/CDのサポート 53 5.5 PIC MCUとdsPIC DSCの計測機能付きトレース 58 5.6 PIC32M MCUの命令トレース 59 5.6 PIC32M MCUの命令トレース 65 5.7 SAM ITM/SWOトレース 65 5.8 PCサンプリング - 16ビットPIC MCUのみ 73 5.9 PCプロファイリング - 32ビットMCUのみ 74 5.10 デバッガポーリング 76 5.11 Power Monitor 77 6. トラブルシュートの最初のステップ 85 6.1 最初に確認する項目 85 6.2 デバッグにた敗する主な理由 65 6.3 一般的注意事項 66 6.4 ハードウェア ツール エマージェンシー ブート ファームウェアレウェアリカ・ジェ 86 <td></td> <td>3.2</td> <td>PCの接続</td> <td> 10</td>		3.2	PCの接続	10
4. 動作 37 4.1 MPLAB X IDEでのデバッグ 37 4.2 ATデバイス - OCD(オンチップデバッグ) 37 4.3 PIC MCU/dsPIC DSC - オンチップデバッグ 45 5. 本エミュレータの特長 53 5.1 USB CDC仮想COMポート 53 5.2 DGI 53 5.3 データキャプチャとランタイム ウォッチ - PIC MCUとdsPIC DSC 57 5.4 CI/CDのサポート 58 5.5 PIC MCUとdsPIC DSCの計測機能付きトレース 59 5.6 PIC32M MCUの命令トレース 59 5.7 SAM ITM/SWOトレース 65 5.7 SAM ITM/SWOトレース 67 5.8 PCサンプリング - 16ビッ PPIC MCUのみ 74 5.10 デバッガポーリング 76 5.11 Power Monitor 77 6. トラブルシュートの最初のステップ 85 6.1 最初に確認する項目 85 6.2 デバッグに失敗する主な理由 85 6.3 一般的注意事項 86 6.4 ハードウェア ツール エマージェンシー ブート ファームウェア リカバリ 2 ユーティリティの使い方 76 7.1 動作に関するFAQ 88		3.3	ターゲットとの接続	15
4.1 MPLAB X IDEでのデバッグ 37 4.2 ATデバイス - OCD(オンチップデバッグ) 37 4.3 PIC MCU/dsPIC DSC - オンチップデバッグ 45 5. 本エミュレータの特長 53 5.1 USB CDC仮想COMポート 53 5.2 DGI 53 5.3 データキャプチャとランタイム ウォッチ - PIC MCUとdsPIC DSC 57 5.4 CI/CDのサポート 58 5.5 PIC MCUとdsPIC DSCの計測機能付きトレース 59 5.6 PIC32M MCUの命令トレース 65 5.7 SAM ITM/SWOトレース 67 5.8 PCサンプリング - 16ビットPIC MCUのみ 73 5.9 PCプロファイリング - 32ビットMCUのみ 74 5.10 デバッガボーリング 76 5.11 Power Monitor 77 6. トラブルシュートの最初のステップ 85 6.1 最初に確認する項目 85 6.2 デバッグに失敗する主な理由 85 6.3 一般的注意事項 86 6.4 ハードウェア ツール エマージェンシー ブート ファームウェア リカバリ 2-ディリティ リティの使い方 7.4 転行 86 7.5 ホイドウェア ツール エマージェンシー ブート ファームウェア リカバリ 86 7.1 動作	1	勈ル		27
4.1 MP LAB × ND L COD (オンチップデバッグ)	4.	到11	MDLAR Y IDEでのデバッグ	37
4.2 AT I / AT A = OCD(A D J D J J A D J A D D J A D D J A D D J A D D J A D D J A D D J A D D J A D D J A D D J A D D J A D D J A D D J A D D J A D D J A D D J A D D J A D D J A D D J A D D D J A D D J A D D D D		4.1	MFEAD XIDE COULT (チャップ デバッグ)	37
4.3 THE MODULAS HE DISCE - オンディアノアノアイリアノアイリアノアイリアノアイリアノアイリアノアイリアノアイト 53 5. 本エミュレータの特長 53 5.1 USB CDC仮想COMポート 53 5.2 DGI 53 5.3 データキャプチャとランタイム ウォッチ - PIC MCUとdsPIC DSC 57 5.4 CI/CDのサポート 58 5.5 PIC MCUとdsPIC DSCの計測機能付きトレース 59 5.6 PIC32M MCUの命令トレース 65 5.7 SAM ITM/SWOトレース 65 5.8 PCサンプリング - 16ビットPIC MCUのみ 73 5.9 PCプロファイリング - 32ビットMCUのみ 74 5.10 デバッガ ポーリング 76 5.11 Power Monitor 77 6. トラブルシュートの最初のステップ 85 6.1 最初に確認する項目 85 6.2 デバッグに失敗する主な理由 85 6.3 一般的注意事項 86 6.4 ハードウェア ツール エマージェンシー ブート ファームウェア リカバリ 1 ユーティ リティ の使い方 86 7.1 動作に関するFAQ 88		4.2 1 3	$PIC MCI/dePIC DSC = t v \neq y \neq \overline{z} v \neq \overline{z}$	57
5. 本エミュレータの特長 53 5.1 USB CDC仮想COMポート 53 5.2 DGI 53 5.3 データキャプチャとランタイム ウォッチ - PIC MCUとdsPIC DSC 57 5.4 CI/CDのサポート 58 5.5 PIC MCUとdsPIC DSCの計測機能付きトレース 59 5.6 PIC32M MCUの命令トレース 65 5.7 SAM ITM/SWOトレース 67 5.8 PCサンプリング - 16ビットPIC MCUのみ 73 5.9 PCプロファイリング - 32ビットMCUのみ 74 5.10 デバッガ ポーリング 76 5.11 Power Monitor 77 6. トラブルシュートの最初のステップ 85 6.3 一般的注意事項 85 6.3 一般的注意事項 86 6.4 ハードウェア ツール エマージェンシー ブート ファームウェア リカバリ 86 7.1 動作に関するFAQ 88		4.5		40
5.1 USB CDC仮想COMボート 53 5.2 DGI 53 5.3 データキャプチャとランタイム ウォッチ - PIC MCUとdsPIC DSC 57 5.4 CI/CDのサポート 58 5.5 PIC MCUとdsPIC DSCの計測機能付きトレース 59 5.6 PIC32M MCUの命令トレース 65 5.7 SAM ITM/SWOトレース 67 5.8 PCサンプリング - 16ビットPIC MCUのみ 73 5.9 PCプロファイリング - 32ビットMCUのみ 74 5.10 デバッガ ポーリング 72 5.11 Power Monitor 77 6. トラブルシュートの最初のステップ 85 6.1 最初に確認する項目 85 6.2 デバッグに失敗する主な理由 85 6.3 一般的注意事項 86 6.4 ハードウェア ツール エマージェンシー ブート ファームウェア リカバリ 2 ユーティ リティ の使い方 86 7.1 動作に関するFAQ 88	5.	本エミ	ミュレータの特長	53
5.2 DGI 53 5.3 データキャプチャとランタイム ウォッチ - PIC MCUとdsPIC DSC 57 5.4 CI/CDのサポート 58 5.5 PIC MCUとdsPIC DSCの計測機能付きトレース 59 5.6 PIC32M MCUの命令トレース 65 5.7 SAM ITM/SWOトレース 67 5.8 PCサンプリング - 16ビットPIC MCUのみ 73 5.9 PCプロファイリング - 32ビットMCUのみ 74 5.10 デバッガ ポーリング 76 5.11 Power Monitor 77 6. トラブルシュートの最初のステップ 85 6.1 最初に確認する項目 85 6.2 デバッグに失敗する主な理由 85 6.3 一般的注意事項 86 6.4 ハードウェア ツール エマージェンシー ブート ファームウェア リカバリ 2 ユーティ リティ の使い方 86 7.1 動作に関するFAQ 88		5.1	USB CDC仮想COMポート	53
5.3 データキャプチャとランタイム ウォッチ - PIC MCUとdsPIC DSC		5.2	DGI	53
5.4 CI/CDのサポート		5.3	データキャプチャとランタイム ウォッチ - PIC MCUとdsPIC DSC	57
5.5 PIC MCUとdsPIC DSCの計測機能付きトレース 59 5.6 PIC32M MCUの命令トレース 65 5.7 SAM ITM/SWOトレース 67 5.8 PCサンプリング - 16ビットPIC MCUのみ 73 5.9 PCプロファイリング - 32ビットMCUのみ 74 5.10 デバッガ ポーリング 76 5.11 Power Monitor 77 6. トラブルシュートの最初のステップ 85 6.1 最初に確認する項目 85 6.2 デバッグに失敗する主な理由 85 6.3 一般的注意事項 86 6.4 ハードウェア ツール エマージェンシー ブート ファームウェア リカバリ 2 ユーティ リティの使い方 86 7.1 動作に関するFAQ 88		5.4	CI/CDのサポート	58
5.6 PIC32M MCUの命令トレース 65 5.7 SAM ITM/SWOトレース 67 5.8 PCサンプリング - 16ビットPIC MCUのみ 73 5.9 PCプロファイリング - 32ビットMCUのみ 74 5.10 デバッガ ポーリング 76 5.11 Power Monitor 77 6. トラブルシュートの最初のステップ 85 6.1 最初に確認する項目 85 6.2 デバッグに失敗する主な理由 85 6.3 一般的注意事項 86 6.4 ハードウェア ツール エマージェンシー ブート ファームウェア リカバリ 2 ユーティ リティの使い方 86 7.1 動作に関するFAQ 88		5.5	PIC MCUとdsPIC DSCの計測機能付きトレース	59
5.7 SAM ITM/SWOトレース		5.6	PIC32M MCUの命令トレース	65
5.8 PCサンプリング - 16ビットPIC MCUのみ		5.7	SAM ITM/SWOトレース	67
5.9 PCプロファイリング - 32ビットMCUのみ		5.8	PCサンプリング - 16ビットPIC MCUのみ	73
5.10 デバッガ ポーリング 76 5.11 Power Monitor 77 6. トラブルシュートの最初のステップ 85 6.1 最初に確認する項目 85 6.2 デバッグに失敗する主な理由 85 6.3 一般的注意事項 86 6.4 ハードウェア ツール エマージェンシー ブート ファームウェア リカバリ 86 7. よく寄せられる質問(FAQ) 88 7.1 動作に関するFAQ 88		5.9	PCプロファイリング - 32ビットMCUのみ	74
5.11 Power Monitor 77 6. トラブルシュートの最初のステップ 85 6.1 最初に確認する項目 85 6.2 デバッグに失敗する主な理由 85 6.3 一般的注意事項 86 6.4 ハードウェア ツール エマージェンシー ブート ファームウェア リカバリ ユーティリティの使い方 86 7. よく寄せられる質問(FAQ) 88 7.1 動作に関するFAQ 88		5.10	デバッガ ポーリング	76
 6. トラブルシュートの最初のステップ		5.11	Power Monitor	77
6.1 最初に確認する項目 85 6.2 デバッグに失敗する主な理由 85 6.3 一般的注意事項 86 6.4 ハードウェア ツール エマージェンシー ブート ファームウェア リカバリ ユーティリティの使い方 86 7. よく寄せられる質問(FAQ) 88 7.1 動作に関するFAQ 88	6.	トラフ	ブルシュートの最初のステップ	85
 6.2 デバッグに失敗する主な理由		6.1	最初に確認する項目	85
 6.3 一般的注意事項		6.2	デバッグに失敗する主な理由	85
 6.4 ハードウェア ツール エマージェンシー ブート ファームウェア リカバリ ユーティリティの使い方		6.3	一般的注意事項	86
ユーティリティの使い方		6.4	ハードウェア ツール エマージェンシー ブート ファームウェア リカバリ	
 よく寄せられる質問(FAQ)			ユーティリティの使い方	86
7.1 動作に関するFAQ	7.	よく暑	Fせられる質問(FAQ)	88
		7.1	· · · · · · · · · · · · · · ·	88

MPLAB® ICE 4インサーキット エミュレータ ユーザガイド

	7.2	トラブルに関するFAQ	
8	т <i>⇒</i> _	-メッセージ	90
0.	81	エラーメッセージのタイプ	90
	8.2	 一 ジ ジ ジ ご ジ ジ ジ 、 ジ 一 般的な対処方法 	
9.	エミュ	レータの機能のまとめ	
	9.1	エミュレータの選択と切り換え	101
	9.2	エミュレータ オブションの選択	101
	9.3	エミュレータのウィンドウとダイアログ	109
10.	ハード	・ウェア仕様	113
	10.1	エミュレータ ユニット	113
	10.2	電源仕様	113
	10.3	インジケータ ランプ(LED)	113
	10.4	PC接続の仕様	114
	10.5	ターゲット接続の仕様	115
	10.6	リカバリの仕様	119
	10.7	ターゲットボードに関する注意事項	119
11	改訂薩		120
	11.1		
12.	サポー		
	12.1		
	12.2	myMicrochip変更通知サービス	121
Mic	rochip∤	生ウェブサイト	122
製品変更通知サービス			
お客様サポート			
製品識別システム123			
Microchip社のデバイスコード保護機能			
法律上の注意点			
商樗	商標124		
品質	[管理シ	マステム	125
各国	各国の営業所とサービス		

1. 序章

MPLAB[®] ICE 4 の関連文書とサポート情報は以下の項目に記載しています。

1.1 本書の表記規則

本書には以下の表記規則を適用しています。

表 1-1.本書の表記規則

表記	意味	例
Arial、MS ゴシックフォント:		
二重かぎカッコ: 『』	参考資料	『MPLAB [®] IDE ユーザガイド』
太字	テキストの強調	…は唯一のコンパイラです…
角カッコ:[]	ウィンドウ名	[Output]ウィンドウ
	ダイアログ名	[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(>)で区切り、角カッコ ([])で囲んだ下線付きテキスト	メニューパス	[File] > [Save]
角カッコ([])で囲んだ太字の	ダイアログのボタン	[OK]をクリックする
テキスト	タブ	[Power]タブをクリックする
N 'Rnnnn	Verilog 形式の数値(N は総桁数、 R は基数、n は各桁の値)	4'b0010, 2'hF1
山カッコ(< >)で囲んだテキスト	キーボードのキー	<enter>、<f1>を押す</f1></enter>
Courier New フォント		
標準書体の Courier New	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'
斜体の Courier New	変数の引数	^{file.o} (file は有効な任意のファイル 名)
角カッコ:[]	オプションの引数	<pre>mcc18 [options] file [options]</pre>
中カッコとパイプ文字: { }	どちらかの引数を選択する場合 (OR 選択)	errorlevel {0 1}

続き		
表記	意味	例
省略記号:	繰り返されるテキスト	<pre>var_name [, var_name]</pre>
	ユーザが定義するコード	<pre>void main (void) { }</pre>

1.2 推奨参考資料

本書では MPLAB ICE 4 インサーキット エミュレータの使い方を説明します。以下の文書にも役に立つ情報が記載 されています。参考資料として、Microchip 社が提供する以下の文書を推奨します。

『開発ツールの設計注意書』

この文書を最初にお読みください。

この文書には、MPLAB ICE 4 インサーキット エミュレータを使う時に注意すべき情報が記載されています。以下を 参照してください。

開発者ヘルプ: 『開発ツールの設計注意書』(最新版)

PDF: 『各種ツールの設計注意書』(DS51764) MPLAB X IDE オンラインヘルプ/ユーザガイド

このオンラインヘルプは全ての Microchip 社ハードウェア ツール向けの文書です。

これは MPLAB X IDE の広範なヘルプファイルです。これには組み込みシステムの概要、インストール要件、 チュートリアル、プロジェクトの新規作成、ビルドプロパティの設定、コードのデバッグ、コンフィグレーション ビットの設定、ブレークポイントの設定、デバイスのプログラミングに関する詳細を説明しています。このヘルプ ファイルは通常、以下からダウンロードできる PDF 版ユーザガイド(DS50002027)よりも新しい情報を収めていま す。www.microchip.com/mplabx/

MPLAB[®] ICE 4 インサーキット エミュレータのリリースノート

MPLAB ICE 4の使い方に関する最新情報は、MPLAB X IDE ツールバーの[Help] > [Release Notes]を選択してくだ さい。リリースノートには、本書に記載できなかった最新情報と既知の問題を記載しています。

MPLAB ICE 4 In-Circuit Emulator Quick Start Guide Poster (DS50003240)

この文書には、ターゲットボードを使った MPLAB ICE 4 のハードウェアの接続方法とソフトウェアのインストー ル方法を記載しています。 MPLAB[®] ICE 4インサーキット エミュレータ ユーザガイド ^{本エミュレータについて}

2. 本エミュレータについて

MPLAB ICE 4 インサーキット エミュレータ/プログラマ(DV244140)は、Microchip 社の MCU(マイクロコントローラ)、 PIC、dsPIC、AVR[®]、SAM デバイス用に Microchip 社が提供する高速で機能豊富な最新のエミュレーションおよび プログラミング ツールです。

本ツールを使うと、MPLAB X IDE(統合開発環境)の使いやすくパワフルなグラフィカル ユーザ インターフェイスを 使って、デバッグとプログラミングが行えます。

既定の構成では、MPLAB ICE 4 はハイスピード USB 3.0/2.0 インターフェイスで PC と接続します。しかし、Wi-Fi または Ethernet を使って接続する事もできます。

MPLAB ICE 4 は高速 Edge Rate[®]ケーブルを使ってターゲットと接続します。このケーブルの一方の端はエミュレータに、もう一方の端はサポートされているデバイス通信に合わせたアダプタボードに接続されます。

このエミュレータは、特別なデバッガ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と通信するため、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 デバイスが実装している全機能にインタラクティブに アクセスでき、MPLAB X IDE インターフェイスによる設定と変更が可能です。

MPLAB ICE 4 は組み込みプロセッサ向けの高機能デバッガとして設計されており、従来のシステム プロセッサとは 異なる以下の特長を備えています。

- プロセッサを最大速度で動作させる事ができる
- I/O ポートデータ入力が利用できる
- ・ 先進のホスト通信インターフェイス(対応 OS: Windows、Linux、macOS)
- 先進の通信媒体およびプロトコル
- 高速なプログラミングが可能
- ・ モジュール設計(テストと保守が容易)

MPLAB ICE 4 システムは、エミュレータとしてだけでなく量産プログラマとして使う事もできます。

2.1 長所

MPLAB ICE 4 インサーキット エミュレーション システムには以下の長所があります。

特長/機能:

- ハイスピード USB 3.0/2.0、Ethernet、Wi-Fi のいずれかでコンピュータと接続
- 新しいターゲットには Edge Rate ハイスピード同軸ケーブル アセンブリを使って接続。レガシー ターゲットには付属のアダプタボードを使って接続
- MPLAB X IDE または MPLAB IPE を使ってデバイスのプログラミングを実行
- 複数のブレークポイント、ストップウォッチ機能、ソースコード ファイルのデバッグをサポート
- 実際のハードウェア上でのリアルタイム デバッグ
- 内部イベントに基づくブレークポイントの設定
- 内部ファイルレジスタの監視
- フルスピードでのデバッグ
- ピンドライバの設定
- ファームウェア ダウンロードを使ってフィールド アップグレードが可能
- デバイスとツールパックの最新バージョン(www.microchip.com/mplabx/から無償でダウンロード可能)をインストールする事で新しいデバイスのサポートと機能を追加
- LED の輝度制御
- 0~70 ℃の温度レンジで動作

性能/速度:

- 大容量かつ高速のメモリ
- リアルタイム オペレーティング システム(RTOS)

MPLAB[®] ICE 4インサーキット エミュレータ ユーザガイド ^{本エミュレータについて}

- ターゲット デバイス切り換え時にファームウェア ダウンロード遅延が生じない
- 32 ビット MCU は 300 MHz で動作
- バッファメモリは4MB

安全性:

- ターゲットに外付け電源が必要な場合はデバッガからフィードバックを受信
- 1.2~5.5 Vのターゲット電源電圧をサポート
- 9 V DC 電源を使うと最大 1 A までのターゲット電流を安全に供給可能
- プローブドライバに追加した保護回路でターゲットからの電源サージによる故障を防止
- 電圧監視により VDD と VPP を過電圧条件から保護し、全てのラインが過電流保護を備える
- ターゲットへの電圧、設定可能な抵抗値や、方向(プルアップ、プルダウン、どちらもなし)が安全と判定されるまで電源ピンを物理的に絶縁
- プログラミング速度を制御してターゲットボード設計上の問題を回避可能
- 業界標準である CE および RoHS に準拠 RED による試験済み

2.2 構成要素

MPLAB ICE 4000 インサーキット エミュレータ キットは以下の要素で構成されています。

- 長方形の MPLAB ICE 4 ユニット: 丈夫な黒色金属ケースに収納し、アクセントとして LED インジケータバーを 配置(下図参照)。本体側面には USB コネクタ、Ethernet コネクタ、電源コネクタ、通信およびデバッグ用コネ クタ
- ・ 既定のコンピュータ/エミュレータ通信用の Type C USB ケーブル
- ユニットへの給電用9V電源
- ・ ターゲット デバイスに接続するための6インチ 高速 Edge Rate ケーブル
- レガシー ターゲット接続をサポートするための各種アダプタボードと関連ケーブル。キットボックス トレイの下に付いている小さなボックスの中に入っています。つまり、トレイを引き出すと、アダプタボードとケーブルのボックスがあります

図 2-1. エミュレータ ユニット



以下のハードウェアとアクセサリ類も Microchip Purchasing and Client Services(オンラインストア) (www.microshipdirect.com)でご注文頂けます。

• 9 V AC 電源アダプタ(製品番号 AC002014)。詳細は 10.2. 電源仕様を参照)

CAT5e/CAT6 Ethernet ケーブルは他の場所で購入できます。ツメ折れ防止カバーの付いていないケーブルが最も適しています。

2.3 ブロック図

以下に、基本的な MPLAB ICE 4 ユニットの動作と機能のブロック図を示します。



2.4 MPLAB ICE 4 & MPLAB X IDE

MPLAB ICE 4 インサーキット エミュレータを MPLAB X IDE(統合開発環境)と組み合わせて使い、ターゲット アプリケーションを開発できます。

最新バージョンの MPLAB X IDE を MPLAB X IDE のウェブページからダウンロードしてインストールします。本 ユーザガイドとその他の文書はウェブページにもあります。IDE がインストールされたら、以下の操作を実行しま す。

デスクトップのアイコンをクリックして IDE を起動するデスクトップのアイコンをクリックして IDE を起動する新規にプロジェクトを作成するか、既存のプロジェクトを開くハードウェア ツールとして MPLAB ICE 4 を選択するプロジェクト名を右クリックして[Properties]を選択し、[Project Properties]ダイアログを開く。このウィ ンドウを使ってデバッグ、プログラミング、その他の機能のオプションを設定する(9.2. エミュレータオ プションの選択を参照)		
 新規にプロジェクトを作成するか、既存のプロジェクトを開くハードウェア ツールとして MPLAB ICE 4 を選択する プロジェクト名を右クリックして[Properties]を選択し、[Project Properties]ダイアログを開く。このウィ ンドウを使ってデバッグ、プログラミング、その他の機能のオプションを設定する(9.2. エミュレータオ プションの選択を参照) 	MPLAB X IDE VX.XX	デスクトップのアイコンをクリックして IDE を起動する
プロジェクト名を右クリックして[Properties]を選択し、[Project Properties]ダイアログを開く。このウィ ンドウを使ってデバッグ、プログラミング、その他の機能のオプションを設定する(9.2. エミュレータオ プションの選択を参照)		新規にプロジェクトを作成するか、既存のプロジェクトを開くハードウェア ツールとして MPLAB ICE 4 を選択する
	Propert Properties - SCLE.PC Chapters Orena Oren	プロジェクト名を右クリックして[Properties]を選択し、[Project Properties]ダイアログを開く。このウィ ンドウを使ってデバッグ、プログラミング、その他の機能のオプションを設定する(9.2. エミュレータオ プションの選択を参照)

3. 接続

MPLAB ICE 4 インサーキット エミュレータ ハードウェアを設定するには、最初に電源とターゲットをエミュレー タに接続して通信方法を指定します。レガシー ターゲット用にアダプタボードとケーブルが付属しています。

3.1 電源とセルフテスト

MPLAB ICE 4 インサーキット エミュレータと必要に応じてターゲットに電源を供給するには 9 V 外部電源 (AC002014)が**必要**です。この電源はエミュレータ キットに付属しています。3P プラグをコンセントに挿し込み、 もう一方のプラグ先端をエミュレータ上部(下図参照)に接続します。

Note: USB から MPLAB ICE 4 ユニットに給電する事はできません。

図 3-1. MPLAB ICE 4 の電源



電源投入時セルフテスト

MPLAB ICE 4 ユニットは電源投入時に BIST (Built-In Self-Test)を実行します。このテスト中に発生したエラーは MPLAB X IDE の[Output]ウィンドウで報告されます。エラーによっては LED の色でエラーを示す事もあります。

ターゲットへの給電

エミュレータを使ってターゲットに給電できます。詳細は 10.2. 電源仕様を参照してください。

[Project Properties]ウィンドウで以下のオプションを選択します(下図参照)。必要なターゲット電圧も選択します。

図 3-2. ターゲットへの給電の選択

Project Properties - ICE4_LowPower		
Categories: O General O File Indusion/Exclusion	Options for ICE 4 Option categories: Power	Reset
Ornf: [default] O ICE 4 O Loading O Libraries O Building O XC32 (Global Options) O XC32 - as	Power target circuit from ICE 4 Voltage Level	✓ 3.25
○ xc32-gcc ○ xc32-g++ ○ xc32-ld ○ xc32-ar ○ Analysis		
	Option Description If you select an option its description will appear he	re.
Manage Configurations		
Manage Network Tools		
		Cancel Apply Unlock Help

関連リンク

10.2. 電源仕様 8. エラーメッセージ 10.3. インジケータ ランプ(LED)

3.2 PC の接続

MPLAB ICE 4 インサーキット エミュレータは USB3.0/2.0(既定)、Wi-Fi、Ethernet のいずれかを使って PC (と MPLAB X IDE)に接続できます。これらのオプションの詳細は 10.4.PC 接続の仕様を参照してください。

図 3-3. MPLAB ICE 4 の電源と PC の接続

	External
USB 3 or 2	Power WiFi Ethernet
Mi	PLABICE 4 Top of Unit

USB(既定)接続から始めます。次に、[Tools] > [Manage Network Tools]にある[Manage Network Tools]ダイアログで Wi-Fi または Ethernet に切り換えます。詳細は以下のトピックを参照してください。

以下の用途の場合、USB ではなく Wi-Fi か Ethernet 通信を選択します。

- リモートからターゲットにアクセスする:エミュレータとターゲットの場所と PC の場所を分ける事ができます。
- ターゲットを隔離する: 管理された環境が必要なターゲットを PC とは別の場所に配置できます。

関連リンク

10.4.1. USB Type C コネクタ(J1)およびケーブル 10.4.2. Ethernet コネクタ(J6)およびケーブル

3.2.1 既定の USB 接続

PCと MPLAB ICE 4 ユニットの既定の接続方法は USB Type C ケーブルを使った USB 接続です。

Note: MPLAB X IDE の付属プラグインの場合でもスタンドアロン アプリケーションの場合でも、MPLAB Data Visualizer は USB 通信を使っている時にのみエミュレータを検出できます。

Note: トレースには USB 通信のみ使えます。

他の種類の通信で問題が発生した場合、USB に戻してから[MNT (Manage Network Tools)]ダイアログで再度 Wi-Fi または Ethernet に切り換えます。

3.2.2 Wi-Fiと Ethernet - モード

MPLAB ICE 4 インサーキット エミュレータは複数の Wi-Fi および Ethernet 通信モードをサポートしています。

3.2.2.1 Wi-Fi アクセスポイント(WiFi-AP)

WiFi アクセスポイント(WiFi-AP)モードでは、エミュレータが (ルータのように)独自のネットワークを作成し、他 のデバイス (ノート PC 等)をそこに接続できます。Wi-Fi 信号の範囲内にある PC 経由でエミュレータとターゲット にアクセスする場合にこのモードを使います。

MPLAB ICE 4 にはプリセットの固定された SSID およびパスワードが用意されています。

- SSID: 「ICE4_MTIxxxxxxxx」(xxxxxxxxx はツール固有のシリアル番号)
- パスワード:「microchip」

Note: MPLAB ICE 4 を WiFi-AP モードで接続している時、Windows OS を搭載したコンピュータが他の Wi-Fi 接続 から切断される可能性があります。

図 3-4. アクセスポイントとして動作する MPLAB ICE 4



3.2.2.2 Wi-Fi ステーション モード(WiFi-STA)

Wi-Fi ステーション(WiFi-STA)モードは、MPLAB X IDE でユーザが提供したアクセスポイント ルータのパラメータ (SSID とパスワード)を使って自宅/オフィスのネットワークに接続を試みます。エミュレータおよびターゲットと PC との距離を WiFi-AP よりも離す(隔離する)場合にこのモードを使います。

Note: MPLAB ICE 4 は 5.0 GHz ではなく 2.4 GHz をサポートします。

図 3-5. ステーション モードで動作する MPLAB ICE 4



3.2.2.3 Ethernet (Wired/DHCP/APIPA)

MPLAB ICE 4 インサーキット エミュレータを Ethernet に接続する時に DHCP(または APIPA)の IP アドレスを要求 します。

DHCP (Dynamic Host Configuration Protocol)は、ネットワークに接続されているデバイスに動的 IP アドレスを割り 当てるためのものです。APIPA(Automatic Private IP Addressing)は、DHCP サーバと通信できない場合にコン ピュータが自動的に IP アドレスとサブネットマスクを設定できる Windows OS の機能です。



3.2.2.4 Ethernet 静的 IP

静的 IP アドレスはネットワーク デバイスに恒久的に割り当てられる IP アドレスです。



3.2.3 Wi-Fi と Ethernet - 設定とツールの探索

下表の手順に従って必要な Wi-Fi または Ethernet モードを設定して接続を探します。

表 3-1. MPLAB X IDE での Ethernet または Wi-Fi の設定とツールの探索

ステップ	アクション
1	USB ケーブルを使ってエミュレータを PC に接続する
2	MPLAB X IDE で[<u>Tools] > [Manage Network Tools]</u> を選 択する(下図参照)
3	[Network Capable Tools Plugged into USB]でエミュ レータを選択する
4	[Configure Default Connection Type for Selected Tool] で、目的の接続に該当するラジオボタンを選択する Ethernet (Wired/StaticIP): 静的 IP アドレス、サブネッ トマスク、ゲートウェイを入力する
	Wi-Fi STA: 自宅/オフィスのルータのセキュリティ タ イプに応じて SSID、セキュリティ タイプ、パスワー ドを入力する
	[Update Connection Type]をクリックする
5	エミュレータ ユニットから USB ケーブルを外す
6	エミュレータが自動的に再起動し、選択した接続 モードで表示される。起動後、エミュレータは以下 のどちらかで動作する Wi-Fi AP 以外: ネットワーク接続に成功したか、失敗 した(エラーが発生した)かが LED によって表示される
	Wi-Fi AP : Windows OS/macOS/Linux OS の通常の Wi- Fi スキャン処理が、PC 上で利用可能な Wi-Fi ネット ワークをスキャンする
	SSID が「ICE4_MTIxxxxxxx」のツールを探し (xxxxxxxx はツール固有のシリアル番号)、パスワード 「microchip」を使ってそのツールに接続する

続き		
ステップ	アクション	
7	[Manage Network Tools]ダイアログに戻って[Scan]ボタ ンをクリックする。[Active Discovered Network Tools]の 下にエミュレータのリストが表示される。ツールに該当 するチェックボックスを選択してダイアログを閉じる Wi-Fi AP: Windows 10 コンピュータでは「インター ネットなし、セキュリティ保護あり」というメッセー ジが表示されているにもかかわらず、接続されている 事を示す[切断]ボタンが表示される事があります。こ のメッセージは、エミュレータがルータ/AP として接 続されているものの直接接続 (Ethernet)によるもので はない事を意味します。	
8	[Active Discovered Network Tools]にエミュレータが見 つからない場合、[User Specified Network Tools]セク ションに手動で情報を入力できる。ツールの IP アドレ スが必要(ネットワーク管理者に問い合わせるか、静的 IP を割り当てる)	

図 3-6. 最初の USB 接続

Manage Network Tools		
Configure Tools For Network Usage	Selectable Network Tools	
Network Capable Tools Plugged Into USB:	Disable All Network Tool Discovery/Access	
MPLAB ICE 4-BUR123456789	Active Discovered Network Tools	
	Name:	
	Update Name	
	Use Type and Serial Number for Name	
Configure Default Connection Type for Selected Tool:		
() USB		
	Found at IP Address:	
Ethernet (Wired/DHCP/APIAP)		
0.417.45		
O WIFI-AP	Scan	
Ethernet (Wired/StaticIP):		
· · · · · · · · · · · · · · · · · · ·	User-Specified Network Tools	
Static IP Address:	Name	
Gateway:		
	Update Name	
Subnet Mask:	Use Type and Serial Number for Name	
WIFI-STA:		
Network SSID: Security Type:	IP Address:	
Open (None) 🗸	Set	
Update Connection Type		
	New Remove	
	Close	

図 3-7. WiFi-STA の例

🔀 Manage Network Tools			
Configure Tools For Network Usage	Selectable Network Tools		
Network Capable Tools Plugged Into USB:	Disable All Network Tool Discovery/Access		
	r Active Discovered Network Tools		
MPLAD ICE 4-DUR123436763			
	Name:		
	Lindate Name		
Configure Default Connection Type for Selected Tool:	Use Type and Serial Number for Name		
© USB			
Ethernet (Wired/DHCP/APIAP)	Found at IP Address:		
💿 WiFi-AP			
Ethernet (Wired/StaticIP):	Scan		
Static IP Address:			
Cataway	User-Specified Network Tools		
Gateway:	Name:		
Subnet Mask:	Update Name		
	Lice Tune and Serial Number for Name		
WIFI-STA:	ose rype and serial number for hance		
Network SSID: Security Type:			
MyModem WPA/WPA2 Personal -			
Password:	IP Address:		
••••••	Set		
Update Connection Type	New Remove		
	Close		

3.3 ターゲットとの接続

MPLAB ICE 4 インサーキット エミュレータはハイスピード 40 ピン リボンケーブル アセンブリでターゲットに接続します。レガシー ターゲット接続用にオプションのアダプタボードを利用できます。消費電力モニタに使う電流 検出接続もあります。

デバイスと通信の種類と利用可能なアダプタボードについては以下のセクションで説明します。

Note: MPLAB ICE 4 はターゲットに給電できます。詳細は 10.2. 電源仕様を参照してください。[Project Properties] ウィンドウの[ICE 4]カテゴリの[Power]オプション カテゴリでターゲットへの給電方法を選択します。

図 3-8. MPLAB ICE 4 ユニットとターゲットとの接続



3.3.1 ターゲットとの接続のピン配置

下表に、MPLAB ICE 4 ユニットからアダプタボードまたはターゲットボードの 40 ピンコネクタに接続するケーブ ルのターゲット側のピン配置を示します。

表 3-2. ターゲットに対するエミュレータ コネクタのピン機能

ピン	表記	機能	ピン	表記	機能
1	CS- A	電流モニタ	2	CS+ A	電力モニタ
3	CS-B	電流モニタ	4	CS+ B	電力モニタ
5	UTIL SDA	予約済み	6	UTIL SCL	予約済み
7	DGI SPI nCS	DGI SPI nCS, PORT6, TRIG6	8	DGI SPI SCK	DGI SPI SCK, SPI SCK, PORT7, TRIG7
9	DGI SPI MOSI	DGI SPI MOSI, SPI DATA, PORT5, TRIG5	10	DGI SPI MISO	DGI SPI MISO, PORT4, TRIG4
11	3V3	予約済み	12	GND	GND
13	DGI GPIO3	DGI GPIO3, PORT3, TRIG3	14	TRCLK	TRCLK, TRACECLK,
15	DGI GPIO2	DGI GPIO2, PORT2, TRIG2	16	GND	GND
17	DGI GPIO1	DGI GPIO1, PORT1, TRIG1	18	TRDAT3	TRDAT3, TRACEDATA(3)
19	DGI GPIO0	DGI GPIO0, PORT0, TRIG0	20	GND	GND
21	5V0	予約済み	22	TRDAT2	TRDAT2, TRACEDATA(2)
23	DGI VCP RXD	DGI RXD, CICD RXD, VCD RXD	24	GND	GND
25	DGI VCP TXD	DGI TXD, CICD TXD, VCD TXD	26	TRDAT1	TRDAT1, TRACEDATA(1)
27	DGI I2C SDA	DGI I2C SDA	28	GND	GND
29	DGI I2C SCL	DGI I2C SCL	30	TRDAT0	TRDAT0, TRACEDATA(0)
31		TVDD PWR	32	GND	GND
33	TDI IO	TDI IO, TDI, MOSI	34	TMS IO	TMS IO, SWD IO, TMS
35	TPGC IO	TPGC IO, TPGC, SWCLK, TCK, SCK	36	TAUX IO	TAUX IO, AUX, DW, RESET

	続き				
ピン	表記	機能	ピン	表記	機能
37	TV _{PP} IO	TV _{PP} /MCLR, nMCLR, RST	38	TPGD IO	TPGD IO, TPGD, SWO, TDO, MISO, DAT
39			40	TVDD PWR	

関連リンク

10.5.1. 40 ピンエミュレータ ユニットコネクタ(J10) 10.5.2. ターゲット接続ケーブル

3.3.2 アダプタボード

新しいターゲット設計の場合、デバッグ性能を最大限に引き出すために 40 ピンコネクタを使って MPLAB ICE 4 ケーブル アセンブリと直接接続する事を推奨します。ただし、レガシー ターゲット接続用にアダプタパック (AC244140)のアダプタボードを利用できます。アダプタボードの使用はデバッグ性能を低下させる可能性が高く なります。

表 3-3. 利用可能なアダプタボード

アダプタボード	サポートするデバイス	サポートする機能
ICE 4 JTAG アダプタ	SAM MCU	プログラミング、デバッグ
MPLAB ICE 4 ICSP アダプタボード + • ICE 4 AVR JTAG (10 ピン) • ICE 4 AVR (10 ピンミニ) • ICE 4 AVR (6 ピン) • ICE 4 AVR (6 ピンミニ)	AVR MCU	プログラミング、デバッグ
MPLAB ICE 4 ICSP アダプタボード	PIC MCU、dsPIC DSC	プログラミング、デバッグ、 計測機能付きトレース:
		 ネイティブ トレース SPI トレース I/O ポートトレース(追加の接 続が必要)
MPLAB ICE 4 Cortex-M トレース アダプタボード	SAM MCU	トレース: ・ ETM ・ ETB ・ ITM
MPLAB ICE 4 PIC32 トレース アダプタボード	PIC32M MCU	PIC32 命令トレース

3.3.3 SAM MCU - JTAG/SWD アダプタボード

全ての SAM デバイスがプログラミングおよびデバッグ用に SWD (Serial Wire Debug)インターフェイスを実装して います。また、一部の SAM デバイスは同等の機能を持つ JTAG インターフェイスを実装しています。該当デバイ スでサポートされているインターフェイスについてはデバイスのデータシートを参照してください。

レガシー ターゲット接続用に 10 ピン(上部)または 20 ピン(底部)コネクタを備えた「ICE 4 JTAG アダプタボード」 を利用できます。 図 3-9. JTAG/SWD アダプタボード



本アダプタボードはターゲットボードの20ピンコネクタに接続できます(下図参照)。

図 3-10. ターゲットボードへのアダプタの接続



3.3.3.1 JTAG 物理インターフェイス

JTAG インターフェイスは IEEE® 1149.1 規格に準拠した 4 線式 TAP (Test Access Port)コントローラで構成されま す。この IEEE 規格はボードの接続性試験(バウンダリ スキャン)を効率的に行うための業界標準の方法として策定 されました。Microchip 社製 AVR および SAM デバイスはこの機能を拡張し、完全なプログラミングとオンチップ デバッグをサポートします。

このターゲット インターフェイスを MPLAB X IDE で使うには[Project Properties]ウィンドウを開き、[ICE 4]カテゴ リの[Communications]オプション カテゴリで[2-wire JTAG]または[4-wire JTAG]を選択する必要があります。9.2. 「エミュレータ オプションの選択」を参照してください。

図 3-11. JTAG インターフェイスの基本



3.3.3.1.1 SAM JTAG ターゲットへの接続

ICE 4 JTAG アダプタボードを使う MPLAB ICE 4 はレガシー10 ピン 50-mil JTAG 接続とレガシー20 ピン 100-mil JTAG 接続を提供します。

10 ピン 50-mil ヘッダへの直接接続

Arm[®] Cortex[®]デバッグヘッダのピン配置(3.3.3.1.2.「SAM JTAG のピン配置(Cortex-M デバッグコネクタ)」を参照) に準拠したヘッダを持つターゲットボードに直接接続する場合は 10 ピン 50-mil フラットケーブルを使います。

20 ピン 100-mil ヘッダへの直接接続

20 ピン 100-mil ヘッダを実装したターゲットにアダプタボードを接続します。

3.3.3.1.2 SAM JTAG のピン配置(Cortex-M デバッグコネクタ)

JTAG インターフェイスを備えた Microchip 社の SAM を実装するアプリケーション PCB を設計する場合、下図の ピン配置を使う事を推奨します。

図 3-12. SAM JTAG ヘッダのピン配置



表 3-4. SAM JTAG ピンの説明

名称	ピン	表記
ТСК	4	テストクロック(MPLAB ICE 4 からターゲット デバイスへのクロック信号)
TMS	2	テストモード選択(MPLAB ICE 4 からターゲット デバイスへの制御信号)
TDI	8	テストデータ入力(MPLAB ICE 4 からターゲット デバイスに送信したデータ)
TDO	6	テストデータ出力(ターゲット デバイスから MPLAB ICE 4 に送信したデータ)
nRESET	10	リセット (オプション)。MPLAB ICE 4 では使いません。
VTG	1	ターゲット参照電圧。MPLAB ICE 4 では使いません。
GND	3, 5, 9	グランド。MPLAB ICE 4 とターゲット デバイスが同じグランド基準を共有するように全て を接続する必要があります。
KEY	7	AVR コネクタの TRST ピンに内部で接続されます。未接続にしておく事を推奨します。



Tip: ピン1と GND の間にデカップリング コンデンサを必ず入れてください。

3.3.3.2 SAM SWD インターフェイス

Arm SWD インターフェイスは TCK および TMS ピンを利用する JTAG インターフェイスのサブセットです。 3.3.3.1.2. 「SAM JTAG のピン配置(Cortex-M デバッグコネクタ)」を参照してください。

MPLAB ICE 4 は、UART フォーマットの ITM トレースをホスト コンピュータにストリーミングできます。詳細は 5.7.2.1.「SAM ITM トレース」を参照してください。

3.3.3.2.1 SAM SWD ターゲットへの接続

SWD デバイスに接続する場合 10 ピン JTAG コネクタを使えます。

3.3.3.2.2 SAM SWD のピン配置

10 ピン JTAG コネクタの場合:

図 3-13. SAM SWD ヘッダのピン配置



表 3-5. SAM SWD ピンの説明

名称	SAM ポートピン	表記
SWDCLK	4	シリアルワイヤ デバッグクロック
SWDIO	2	シリアルワイヤ デバッグデータ入出力
SWO	6	シリアルワイヤ出力(オプション - 一部のデバイスには実装されていません)
nSRST	10	リセット - MPLAB ICE 4 では使いません
VTG	1	ターゲット参照電圧 - MPLAB ICE 4 では使いません
GND	3, 5, 9	グランド

3.3.4 AVR MCU - アダプタボード

AVR デバイスは各種プログラミングおよびデバッグ インターフェイスを備えています。該当デバイスでサポートされているインターフェイスについてはデバイスのデータシートを参照してください。

レガシー ターゲット接続用に追加の AVR 10 ピンまたは 6 ピン JTAG プラグインボードを接続した MPLAB ICE 4 ICSP アダプタボードを使えます。必要なプラグインボード(下図参照)を折って外し、アダプタボードの 8 ピン SIL コネクタに接続します。

図 3-14. AVR JTAG プラグインボード



3.3.4.1 JTAG 物理インターフェイス

JTAG インターフェイスは IEEE[®] 1149.1 規格に準拠した 4 線式 TAP (Test Access Port)コントローラで構成されま す。この IEEE 規格はボードの接続性試験(バウンダリ スキャン)を効率的に行うための業界標準の方法として策定 されました。Microchip 社製 AVR および SAM デバイスはこの機能を拡張し、完全なプログラミングとオンチップ デバッグをサポートします。

このターゲット インターフェイスを MPLAB X IDE で使うには[Project Properties]ウィンドウを開き、[ICE 4]カテゴ リの[Communications]オプション カテゴリで[2-wire JTAG]または[4-wire JTAG]を選択する必要があります。9.2. 「エミュレータ オプションの選択」を参照してください。 図 3-15.JTAG インターフェイスの基本



3.3.4.1.1 AVR JTAG ターゲットへの接続

MPLAB ICE 4 ICSP アダプタボードと AVR JTAG (10 ピン) アダプタボードを使う MPLAB ICE 4 はレガシー10 ピン 50-mil の JTAG 接続を提供します。

10 ピン 50-mil ヘッダへの直接接続

「AVR JTAG (10 ピン)」アダプタボードの 10 ピン 50-mil プラグイン コネクタを使って、3.3.4.1.2.「AVR JTAGの ピン配置」に示します。

3.3.4.1.2 AVR JTAG のピン配置

JTAG インターフェイスを備えた AVR を含むアプリケーション PCB を設計する場合、下図のピン配置を推奨します。 その他の AVR 接続については 4.2.「AT デバイス - オンチップ デバッグ(OCD)」を参照してください。

図 3-16. AVR JTAG ヘッダのピン配置



表 3-6. AVR JTAG ピンの説明

名称	ピン	表記
тск	1	テストクロック(MPLAB ICE 4 からターゲット デバイスへのクロック信号)。
TMS	5	テストモード選択(MPLAB ICE 4 からターゲット デバイスへの制御信号)。
TDI	9	テストデータ入力(MPLAB ICE 4 からターゲット デバイスに送信したデータ)。
TDO	3	テストデータ出力(ターゲット デバイスから MPLAB ICE 4 に送信したデータ)。
nTRST	8	テストリセット (オプション - 一部の AVR デバイスのみ)。JTAG TAP コントローラのリセット用で す。
nSRST	6	リセット (オプション)。ターゲット デバイスのリセット用です。このピンを接続する事で MPLAB ICE 4 はターゲット デバイスをリセット状態に保持できます。これは特定のシナリオでデ バッグに不可欠となる事があるため、このピンの接続を推奨します。
VTG	4*	ターゲット参照電圧。MPLAB ICE 4 はレベルコンバータへ正確に給電するためにこのピンでター ゲット電圧をサンプリングします。MPLAB ICE 4 は、このモードではこのピンから 1 mA 未満の電 流を消費します。
GND	2, 10	グランド。MPLAB ICE 4 とターゲット デバイスが同じグランド基準を共有するように全てを接続 する必要があります。

.....続き

名称 ピン 表記

* ピン1と GND の間にデカップリング コンデンサを必ず入れてください。

3.3.4.2 AVR SPI 物理インターフェイス

インシステム プログラミングは、ターゲットの Microchip 社 AVR の内部 SPI (Serial Peripheral Interface)を使ってフラッ シュメモリと EEPROM メモリにコードをダウンロードします。これはデバッグ インターフェイスではありません。

3.3.4.2.1 AVR SPI ターゲットへの接続

6 ピン SPI コネクタの推奨ピン配置を 3.3.4.2.2.「AVR SPI のピン配置」に示します。

6 ピン 100-mil SPI ヘッダへの接続

標準の 100-mil SPI ヘッダに接続するには AVR 6 ピン アダプタボードを使います。

6 ピン 50-mil SPI ヘッダへの接続

標準の 50-mil SPI ヘッダに接続するには AVR 6 ピンミニ アダプタボードを使います。



Important:

DWEN (debugWIRE Enable)ヒューズを設定した場合、SPIEN ヒューズを同時に設定しても SPI イン ターフェイスは実質的に無効となります。SPI インターフェイスを再度有効にするには、debugWIRE デバッグ セッション中に「disable debugWIRE」コマンドを発行する必要があります。この方法で debugWIRE を無効にするには、SPIEN ヒューズを設定しておく必要があります。MPLAB X IDE が debugWIRE の無効化に失敗する場合、原因の多くは SPIEN ヒューズを設定していない事によるもの です。この場合、SPIEN ヒューズを設定するために高電圧プログラミング インターフェイスを使う必 要があります。



Info:

Microchip 社の AVR 製品に初めて実装されたインシステム プログラミング インターフェイスである事 から、SPI インターフェイスはしばしば「ISP」と呼ばれます。現在は他のインターフェイスもインシ ステム プログラミングに利用できます。

3.3.4.2.2 AVR SPIのピン配置

SPI インターフェイスを備えた AVR を含むアプリケーション PCB を設計する場合、下図のピン配置を使う必要があります。

図 3-17. SPI ヘッダのピン配置



表 3-7. SPI のピン割り当て

AVR ポートのピン	ターゲットのピン	SPIのピン配置
ピン1 (TCK)	SCK	3
ピン 2 (GND)	GND	6
ピン 3 (TDO)	MISO	1
ピン4 (VTG)	VTG	2
ピン 5 (TMS)		
ピン 6 (nSRST)	/RESET	5
ピン 7(未接続)		

続き			
AVR ポートのピン	ターゲットのピン	SPIのピン配置	
ピン 8 (nTRST)			
ピン 9 (TDI)	MOSI	4	
ピン 10 (GND)			

3.3.4.3 AVR PDI

PDI (Program and Debug Interface)は、デバイスの外部プログラミングと内部デバッグのための Microchip 社独自 のインターフェイスです。PDI 物理インターフェイスは、ターゲット デバイスと双方向の半二重同期通信を行う 2ピン インターフェイスです。

3.3.4.3.1 AVR PDI ターゲットへの接続

6 ピン PDI コネクタの推奨ピン配置を 3.3.4.3.2.「AVR PDI のピン配置」に示します。

6 ピン 100-mil PDI ヘッダへの接続

標準の 100-mil PDI ヘッダに接続するには AVR 6 ピン アダプタボードを使います。

6 ピン 50-mil PDI ヘッダへの接続

標準の 50-mil PDI ヘッダに接続するには AVR 6 ピンミニ アダプタボードを使います。

3.3.4.3.2 AVR PDI のピン配置

PDI インターフェイスを備えた Microchip 社の AVR MCU を実装するアプリケーション PCB を設計する場合、下図のピン配置を使う必要があります。

図 3-18. PDI ヘッダのピン配置



AVR ポートのピン	ターゲットのピン	Microchip STK600 PDI のピン配置
ピン1 (TCK)		
ピン2 (GND)	GND	6
ピン 3 (TDO)	PDI_DATA	1
ピン4 (VTG)	VTG	2
ピン5 (TMS)		
ピン6 (nSRST)	PDI_CLK	5
ピン 7(未接続)		
ピン8 (nTRST)		
ピン 9 (TDI)		
ピン 10 (GND)		

表 3-8. PDI のピン割り当て

3.3.4.4 AVR UPDI

UPDI (Unified Program and Debug Interface)は、デバイスの外部プログラミングと内部デバッグのための Microchip 社独自のインターフェイスです。UPDI は全ての AVR XMEGA[®]デバイスに実装されている PDI 2 線式物理インター フェイスの後継です。UPDI は 1 線式インターフェイスで、プログラミングとデバッグのためにターゲット デバイ スとの間で双方向の半二重非同期通信を行います。

3.3.4.4.1 AVR UPDI ターゲットへの接続

6 ピン UPDI コネクタの推奨ピン配置を 3.3.4.4.2.「AVR UPDI のピン配置」に示します。

6 ピン 100-mil UPDI ヘッダへの接続

標準の 100-mil UPDI ヘッダに接続するには AVR 6 ピン アダプタボードを使います。

6 ピン 50-mil UPDI ヘッダへの接続

標準の 50-mil UPDI ヘッダに接続するには AVR 6 ピンミニ アダプタボードを使います。

3.3.4.4.2 AVR UPDI のピン配置

UPDI インターフェイスを備えた Microchip 社の AVR を実装するアプリケーション PCB を設計する場合、下図の ピン配置を使う必要があります。

図 3-19. UPDI ヘッダのピン配置



表 3-9. UPDI のピン割り当て

AVR ポートのピン	ターゲットのピン	Microchip STK600 UPDI のピン配置
ピン1 (TCK)		
ピン2 (GND)	GND	6
ピン 3 (TDO)	UPDI_DATA	1
ピン4 (VTG)	VTG	2
ピン 5 (TMS)		
ピン6 (nSRST)		
ピン 7(未接続)		
ピン8 (nTRST)		
ピン 9 (TDI)		
ピン 10 (GND)		

3.3.4.5 AVR TPI

TPI は一部の tinyAVR[®]デバイスを対象としたプログラミング専用インターフェイスです。デバッグ インターフェイ スではなく、OCD 機能を実装していない tinyAVR デバイスを対象としています。

3.3.4.5.1 AVR TPI ターゲットへの接続

6 ピン TPI コネクタの推奨ピン配置を 3.3.4.5.2.「AVR TPI のピン配置」に示します。

6 ピン 100-mil TPI ヘッダへの接続

標準の 100-mil TPI ヘッダに接続するには AVR 6 ピン アダプタボードを使います。

6 ピン 50-mil TPI ヘッダへの接続

標準の 50-mil TPI ヘッダに接続するには AVR 6 ピンミニ アダプタボードを使います。

3.3.4.5.2 AVR TPIのピン配置

TPI インターフェイスを備えた AVR を実装するアプリケーション PCB を設計する場合、下図のピン配置を使う必要があります。

図 3-20. TPI ヘッダのピン配置



表 3-10. TPI のピン割り当て

AVR ポートのピン	ターゲットのピン	TPIのピン配置
ピン1 (TCK)	CLOCK	3
ピン 2 (GND)	GND	6
ピン 3 (TDO)	DATA	1
ピン4 (VTG)	VTG	2
ピン 5 (TMS)		
ピン 6 (nSRST)	/RESET	5
ピン 7(未接続)		
ピン 8 (nTRST)		
ピン 9 (TDI)		
ピン 10 (GND)		

3.3.4.6 AVR debugWIRE

debugWIRE インターフェイスは少ピンデバイスで使うために Atmel 社によって開発されました。4 本のピンを使う JTAG インターフェイスとは異なり、debugWIRE は 1 本のピン(/RESET)を使ってデバッガツールと双方向の半二 重非同期通信を行います。

debugWIRE インターフェイスを備えた Atmel 社の AVR を実装するアプリケーション PCB を設計する場合、下図のピン配置を使う必要があります。

図 3-21. debugWIRE (SPI)ヘッダのピン配置



Note:

debugWIRE インターフェイスはプログラミング インターフェイスとしては使えません。これは、ターゲットをプ ログラムするために SPI インターフェイスも利用できなければならない事を意味します(3.3.4.2..「AVR SPI のピン 配置」)を参照)。

debugWIRE を使ってデバッグ セッションを開始すると、debugWIRE インターフェイスを使ってフラッシュがプロ グラムされます。これは工場書き込み用に考慮できるオプションではありません。

debugWIRE イネーブル(DWEN)ヒューズを設定し、ロックビットを未設定にすると、ターゲット デバイス内の debugWIRE システムがアクティブになります。/RESET ピンはワイヤード AND(オープンドレイン)の双方向 I/O ピ ン (プルアップは有効)として設定され、ターゲットとデバッガの間の通信ゲートウェイとして機能します。

3.3.5 PIC MCU - ICSP アダプタボード

MPLAB ICE 4 インサーキット エミュレータは、ICSP[™](インサーキット シリアル プログラミング[™])インターフェイ スを介して PIC MCU(マイクロコントローラ)と dsPIC DSC(デジタルシグナル コントローラ)のデバッグとプログラ ミングをサポートしています。デバッグに加え、この接続は他のエミュレーション機能(PIC 計測機能付きトレース 等)もサポートしています。 レガシー ターゲット接続用に、6 ピンまたは 8 ピン RJ-45 および 8 ピン SIL コネクタを備えた「MPLAB ICE 4 ICSP アダプタボード」を利用できます。

図 3-22. ICSP アダプタボード



図 3-23. ターゲットへのモジュラ接続 To Emulator



図 3-24. ターゲットへの SIL 接続



3.3.5.1 ICSP ターゲット接続

Note: その他の情報および図については、お使いのデバイスのデータシートとインターフェイスのアプリケーションノートを参照してください。

MPLAB ICE 4 インサーキット エミュレータは、RJ-45 コネクタに接続した 6 線または 8 線のモジュラ インター フェイス ケーブルを介してターゲット デバイスに接続します。6 ピンケーブルを使う場合、接続 7 (TMS)および 8 (TDI)は使えません。下図に、ターゲットボードの裏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では、互いに接続されるピンの番号の並びが逆になります。つまり、デバッガ側の 1 ピンはターゲット側の6 ピンまたは8 ピンに接続されます。

図 3-25. ターゲット側標準接続



MPLAB ICE 4 インサーキット エミュレータは、SIL(シングル インライン) コネクタに接続した 6 線または 8 線の モジュラ インターフェイス ケーブルを介してターゲット デバイスに接続します。6 ピンケーブルを使う場合、接 続 7 (TMS)および 8 (TDI)は使えません。下図に、ターゲットボードから見たコネクタのピン番号を示します。

図 3-26. ターゲット側 SIL 接続



3.3.5.2 ICSP のターゲット接続回路

下図に、MPLAB ICE 4 インサーキット エミュレータとターゲットボードの接続回路を示します。この図には、 ターゲットボード上のコネクタからデバイスまでの回路も示しています。VPP/MCLR ラインと VDD の間にプル アップ抵抗 Rpu(通常、約 10 kΩ)を接続する事を推奨します。これにより、このラインを Low にする事によってデ バイスをリセットできます。

図 3-27. ターゲットとの ICSP 接続回路



表 3-11. ターゲット コネクタピンとデバイス I/O の対応

ピン番号	デバイス I/O	表記
1	Vpp/MCLR	本エミュレータは、デバイスをプログラミングおよびデバッグするために Vpp へのアク セスを必要とします。
2	Vdd	エミュレータがターゲットの Vdd を検出する事により、ターゲットの低電圧動作の ためのレベル変換とデバイスの検出が可能になります。エミュレータが Vdd ライン の電圧を検出できない場合、エミュレータはデバイスに接続できません。
3	Vss	ターゲットの Vss はエミュレータによって検出されます。 Note: 本エミュレータは Vss もグランドも提供しません。
4	PGD	本エミュレータは、デバイスをプログラミングおよびデバッグするために、PGD と PGC
5	PGC	へのアクセスを必要とします。

表 3-12. デバイス I/O 向け回路

デバイス I/O	表記
Vpp/nMCLR と Vdd	Vpp/MCLR ラインと Vdd の間にプルアップ抵抗(10 kΩ、typ.)を接続します。これにより、この ラインを Low にする事によってデバイスをリセットできます。
XTAL	エミュレータをデバッガとして使う場合、ターゲット デバイスはオシレータを使って動作中で ある事が必要です。
AVdd、AVss	ターゲット デバイスが AVdd ラインと AVss ラインを備えている場合、 エミュレータが動作するにはその全てを適切なレベルに接続する必要があります。この要件は、 電圧レギュレータピン(例: PIC24FJ MCU の ENVREG/DISVREG)にもあてはまります。
Vdd、Vss、 AVdd、AVss	一般的に、デバイスのデータシートに従って全ての Vdd/AVdd および Vss/AVss ラインを適切 なレベルに接続する事を推奨します。また、Vcap ピンを備えたデバイス(例: PIC18FXXJ)で は、Vcap ピンのできるだけ近くに適切な値のコンデンサを接続します。

3.3.5.3 ICSP のターゲット接続と電源

外部からターゲットへの給電には2通りの方法(外部電源から給電、本エミュレータから給電)があります。

以下では、デバッガ動作に関係する 1 ピン(VVPP/MCLR)、5 ピン(PGC)、4 ピン(PGD)の 3 本のラインについて説 明します。図 3-27.「ターゲットとの標準接続回路」には 2 ピン(VDD)と 3 ピン(VSS)も示しています。

ターゲット デバイスに給電する際は、ターゲットにデバイスの VDD 定格を超える電圧を印加しないでください。

デバイスの VDD の絶対最大定格を超えない事が必要です。絶対最大定格条件を超えて長期間曝露させるとデバイス の信頼性に影響する可能性があります。

デバイス データシートの仕様に示すパラメータから外れた条件でのデバイスの運用は推奨しません。

デバイスに必要な電圧レベルと最大定格は、デバイスのデータシートで確認してください。

3.3.6 SAM MCU - Cortex-M トレース アダプタボード

SAM MCU/MPUは ARM Cortex-A5 プロセッサに基づきます。これらの各プロセッサ ファミリのトレースはそれぞ れ種類が異なります。詳細は以下のセクションで説明します。

レガシー ターゲット接続のために 20 ピンミニ JTAG コネクタを備えた「MPLAB ICE 4 Cortex-M トレース アダプ タボード」を利用できます。

図 3-28. SAM (Cortex M)トレース アダプタボード



関連リンク 5.7. SAM ITM/SWO トレース

3.3.6.1 SAM ITM トレース

システムトレースのために、プロセッサはデータ ウォッチポイントとプロファイリング ユニットと共に ITM (Instrumentation Trace Macrocell)を統合しています。ITM トレースは、選択した SAM マイクロコントローラ (Cortex-M3/M4)にあります。

TPIU(トレースポート インターフェイス ユニット)の1本のピンが SWO(Serial Wire Output)用に指定されます。

3.3.6.2 SAM トレースの物理接続

本アダプタボードは CoreSight[™] 20 命令トレースをサポートしています。

ITM/SWO は以下のようにサポートされます。SWDIO と SWDCLK はデバッグに使い、SWO は ITM 出力に使います。



3.3.6.3 SAM トレースのターゲット接続

CoreSight20 コネクタは標準の JTAG (IEEE 1149.1)モードでも SWD(シリアルワイヤ デバッグ) モードでも使えま す。TPIU 連続モードで最大 4 ビットのパラレルトレースをオプションでキャプチャする事もできます。

表	3-13.	CoreSight20	JTAG ⊐	ネクタ	り	ピン
---	-------	-------------	--------	-----	---	----

ピン	信号名	表記
1	VTREF	VCC_TARGET_P3V3
2	TMS	非アクティブ
3	GND	-
4	тск	非アクティブ
5	GND	-
6	TDO	非アクティブ
9	GND	-
10	nSRST	同期マイクロコントローラのリセット

表 3-14. CoreSight20 シリアルワイヤ デバッグ(非同期トレース)のピン

ピン	信号名	表記
2	SWDIO	シリアルワイヤ データ I/O ピンはデバッグ時にターゲットとの間でシリアルデータの送受信 を行います。ターゲット プロセッサの近くで SWDIO を直列終端処理する事を推奨します。
3	GND	-
4	SWDCLK	SWD クロックピンはデバッグ中にターゲットとの間で入出力したデータをクロック同期します。
5	GND	-
6	TRACESWO	シリアルワイヤ出力ピンを使うと、トレースデータを IDE に提供できます。ターゲット プロセッサの近くで SWO を直列終端処理する事を推奨します。
9	GND	-

ピン	信号名	表記		
10	nSRST	同期マイクロコントローラのリセット。		

表 3-15. CoreSight20 パラレルトレース ソースのピン

ピン	信号名	表記
12	TRACECLK	トレースクロック ピンはトレースデータ信号のサンプリングに必要なクロック信号を IDE に提供します。TRACECLK は PCK3 です。
13	-	-
14	TRACED0	トレースデータ[0-3]ピンは、ターゲットからの TPIU 連続モード トレースデータを IDE に 提供します。
15	グランド	-
16	TRACED1	-
17	グランド	-
18	TRACED2	-
19	グランド	-
20	TRACED3	-

3.3.7 PIC 計測機能付きトレース アダプタボード

選択した PIC MCU によっては、本機をデバッグツールとして選ぶとトレース機能(ネイティブ、SPI、ポートトレース)を使える場合があります。レガシー ターゲット接続でこのタイプのトレースをサポートするために MPLAB ICE 4 ICSP アダプタボードを利用できます。

関連リンク

5.5. PIC MCU と dsPIC DSC の計測機能付きトレース

3.3.7.1 ネイティブトレース接続

ネイティブ トレースの場合は 3.3.5.「PIC MCU - ICSP アダプタボード」に記載されているアダプタボードを使い ます。トレース情報の転送には PGD/PGC/EMUC/EMUD ピンを使います。ただし、デバイスがこの機能をサポート している必要があります。サポートしていない場合、他のトレース方法を使います。

このタイプのトレースの詳細は 5.5.3.1.「ネイティブ トレース」を参照してください。

3.3.7.2 SPI トレースの接続

SPI トレースの場合は 3.3.5.「PIC MCU - ICSP アダプタボード」に記載されている 8 ピン接続を備えたアダプタ ボードを使います。トレース情報の転送には TMS/TDI ピンを使います。ただし、デバイスが SPI 機能をサポートし ている必要があります。サポートしていない場合、他のトレース方法を使います。

このタイプのトレースの詳細は 5.5.3.2. 「SPI トレース」を参照してください。

3.3.7.3 IO ポートトレース接続

IO ポートトレースの場合、3.3.5.「PIC MCU - ICSP アダプタボード」に記載されているアダプタボードではなく 40 ピンケーブルを使ってターゲットに接続する必要があります。ハイスピード ケーブルの選択したピンを IO ポー トとして使うとパラレルトレースが可能です。

このトレース構成ではデータ用に7本、クロック用に1本のラインを使います。PORTxは、全てのピンをトレース に割り当てる事ができる8ピンのポートである事が必要です。このポートは物理的に1つのポートである必要はな く、複数のポートのピンを組み合わせて構成できます。このポートピンは使用中のPGCおよびPGDピンと多重化 できません。

以下のように PORTx を設定します。

- 1. [Projects]ウィンドウ内のプロジェクトを右クリックして[Properties]を選択する
- 2. [Project Properties]ウィンドウ内で[ICE4]カテゴリをクリックする
- 3. [Option categories]ドロップダウン リストから[Trace and Profiling]を選択する
- 4. [Data Collection Selection]で、対象デバイスでサポートされているトレース(例: [User Instrumented Trace])を 選択する
- 5. [Communications Medium]で[I/O Port]を選択する
- 6. [I/O Port Selection]リストからポート設定を選択する下表に、基本的な設定を示します。

Note: 外部トリガは同じターゲットのピンを使います。

ターゲットコネクタのピン	ピン名称	内容	
8	PORT7 ¹	クロック	
7	PORT6	データ	
9	PORT5	データ	
10	PORT4	データ	
13	PORT3	データ	
15	PORT2	データ	
17	PORT1	データ	
19	PORT0	データ	
1 ノイズ低減のために 10 kW のプルダウン抵抗を接続します。			

4.3.2.1.「エミュレータの動作を妨げる回路」で述べたように、ポートピンにはプルアップまたはプルダウン抵抗、 コンデンサ、ダイオードを使ってはいけません。

このタイプのトレースの詳細は 5.5.3.3.「I/O ポートトレース」を参照してください。

3.3.8 PIC32 命令トレース アダプタボード

PIC32 命令トレースは PIC32M MCU でのみ使える機能です。さらに、このトレース機能を備えているのは一部の PIC32M MCU だけです。詳細はデバイスのデータシートで確認してください。

レガシー ターゲット接続用に「MPLAB ICE 4 PIC32 トレース アダプタボード」を利用できます。

図 3-29. PIC32 命令トレース アダプタボード



本アダプタボードは、PIM またはデバイスからデバッグピンにアクセスするために 14 ピン EJTAG コネクタを介し てターゲットボードに接続できます。その後、9 ピン トレースコネクタを使って PIM 上部のトレースポートまたは デバイスのポートピン コネクタに接続できます。

図 3-30. PIC32 PIM に接続するアダプタボード



関連リンク

5.6. PIC32M MCU の命令トレース

3.3.8.1 PIC32 命令トレース - 14 ピン EJTAG の接続

下表に、アダプタボードとターゲットの 14 ピン EJTAG 接続を示します。以下のトピックでトレースポート接続について説明します。

表 3-16. アダプタボードとターゲットとの接続のピン配置

ピン	機能	ピン	機能
1	NC	2	GND
3	TDI	4	GND
5	TPGD	6	GND
7	TMS	8	GND
9	TPGC	10	GND
11	TVPP	12	Кеу
13	NC	14	TVDD_PWR

3.3.8.2 PIC32 命令トレース - PIM のサポート

トレースコネクタ付き PIC32 PIM(プラグイン モジュール)のみが PIC32 命令トレースをサポートします。下表を参照してください。利用可能な PIM の詳細は Microchip 社ウェブサイトを参照してください。

表 3-17. PIC32 PIM のリスト

PIM 番号	PIM 名	トレースコネクタ
MA320001	PIC32MX 100 P QFP to 100 P PIM	 ✓
MA320002	PIC32MX USB PIM	~
MA320002-2	PIC32MX450/470 100-pin to 100-pin USB PIM	~
MA320003	PIC32 CAN-USB PIM	✓

続き		
PIM 番号	PIM 名	トレースコネクタ
MA320011	PIC32MX 1xx/2xx PIM	×
MA320012	PIC32MZ2048EC 100-100 P PIM	 Image: A second s
MA320013	PIC32MX270F256 PIM for Bluetooth Audio Development Kit	×
MA320014	PIC32MX270F256 PIM	×
MA320015	PIC32MX570F512L USB/CAN Explorer 16 PIM	 Image: A second s
MA320016	PIC32MZ Audio 144 P PIM for Bluetooth Audio Development Kit	 Image: A second s
MA320017	PIC32MX270F512L PIM for Bluetooth Audio Development Kit	 Image: A second s
MA320018	PIC32MZ EF Audio 144-pin PIM for Bluetooth Audio Development Kit	 Image: A second s
MA320019	PIC32MZ EF PIM	 Image: A second s
MA320020	PIC32MM0064GPL036 General Purpose PIM	×
MA320021	PIC32MX254F256 PIM for Explorer 16	×
MA320023	PIC32MM0256GPM064 General Purpose PIM	×
MA320024	PIC32MK1024 Motor Control Plug In Module	×

3.3.8.3 PIC32 命令トレース - PIM トレースの接続

アダプタボードを PIC32M ターゲットボードのヘッダに接続しても PIC32M PIM のトレースポートに接続しても同じピン構成にする必要があります。

図 3-31. PIC32 PIM の図







Signal Map J1 J2

•••	
TCLK	TRCLK
TRIG1	TRD0
TRIG2	TRD1
TRIG3	TRD2
TRIG4	TRD3

To PIM Trace Port J1



GND: Blue, TRCLK: Yellow, TRDAT: Green

3.3.8.4 PIC32 命令トレース - ターゲット トレースの接続

お客様の基板で命令トレースを使う場合、以下の準備が必要です。

- 終端直列抵抗を追加する必要があります。詳細は下図を参照してください。
- トレース信号の TRCLK と TRD3:0 を切り離すために、トレースに使う信号の基板配線と負荷に応じて、 取り外し可能な0Ωの抵抗を配置する事を推奨します。

図 3-33. PIC32 PIM のピン接続図



3.3.8.5 PIC32 命令トレース - ターゲット ICSP の接続

代替トレース方法としてターゲットボード上の既存の ICSP コネクタを使えます。以下のラインを PIC32 トレース アダプタボードの 14 ピンコネクタからターゲット ICSP の SIL コネクタに接続します。

PIC32 トレース アダプタボード		ターゲットボードの ICSP SIL (PICkit)インターフェイス		
ピン番号	信 号 名	ピン番号	信号名	
11	TVPP_IO	1	MCLR	
14	TVDD_PWR	2	VDD_PWR	
2/4/6/8/10	GND	3	GND	
5	TPGD_IO	4	PGD	
9	TPGC_IO	5	PGC	

関連リンク

3.3.8.1. PIC32 命令トレース - 14 ピン EJTAG の接続 3.3.5.1. ICSP ターゲット接続
4. 動作

本章では、MPLAB ICE 4 インサーキット エミュレータ システムの動作原理を簡単に説明します。本章の目的は、 本エミュレータを使ってデバッグとプログラミングが行えるようにターゲットボードを設計するための情報を提供 する事です。また、問題が発生してもすぐに解決できるよう、インサーキット デバッグ/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。

4.1 MPLAB X IDE でのデバッグ

[Project Properties]ウィンドウでデバッグ、プログラミング、その他のオプションを設定します。9.2. 「エミュレー

タオプションの選択」を参照してください。オプションを設定したらプロジェクトをデバッグします MPLAB X IDE でアプリケーションをデバッグする詳細な方法は MPLAB X IDE のウェブページまたはオンライン ヘルプ版(onlinedocs.microchip.com/)を参照してください。

4.2 AT デバイス - OCD(オンチップ デバッグ)

オンチップ デバッグ モジュールは、通常はデバッガまたはデバッグアダプタと呼ばれるデバイスを介して開発者 が外部の開発プラットフォームからデバイス上の実行を監視、制御できるシステムです。

OCD システムでは、ターゲット システムの電気的およびタイミング特性を正確に維持しながらアプリケーション を実行し、条件付きまたは手動で実行を停止し、プログラムフローとメモリを調べられます。

ランモード

ランモードではコードの実行は MPLAB ICE 4 から完全に独立しています。MPLAB ICE 4 はターゲット デバイスを 継続的に監視してブレーク条件が発生したかどうかを確認します。ブレーク条件が発生すると、OCD システムが デバッグ インターフェイスを通じてデバイスに問い合わせを行う事でデバイスの内部状態を参照できるようになり ます。

停止モード

ブレークポイントに達するとプログラムの実行が停止しますが、一部の I/O はブレークポイントの発生前と同様に 実行を継続できます。例えば、ブレークポイントに到達した瞬間に USART 送信を開始した場合を想定します。この場合、コアが停止モードであっても、USART はフルスピードで実行を継続して送信を完了します。

ハードウェア ブレークポイント

ターゲット OCD モジュールは、ハードウェアに複数のプログラム カウンタ コンパレータを実装しています。プロ グラム カウンタがコンパレータ レジスタの1つに格納されている値と一致すると、OCD は停止モードに移行しま す。ハードウェア ブレークポイントは OCD モジュール上に専用のハードウェアを必要とするため、利用可能なブ レークポイントの数は、ターゲットに実装された OCD モジュールのサイズによって決まります。通常、このハー ドウェア コンパレータの1つはデバッガが内部で使うために「予約」されています。

ソフトウェア ブレークポイント

ソフトウェア ブレークポイントはターゲット デバイスのプログラムメモリ内に配置された BREAK 命令です。こ の命令が読み込まれるとプログラム実行がブレークし、OCD が停止モードに移行します。実行を継続するには 「start」コマンドを OCD から発行する必要があります。全ての Microchip 社製デバイスが BREAK 命令をサポート する OCD モジュールを備えている訳ではありません。

4.2.1 JTAG/SWD を備えた SAM デバイス

全ての SAM デバイスはプログラミングとデバッグのための SWD インターフェイスを備えています。また、一部 の SAM デバイスは同等の機能を持つ JTAG インターフェイスを備えています。該当デバイスでサポートされてい るインターフェイスについてはデバイスのデータシートを参照してください。

4.2.1.1 ARM CoreSight のコンポーネント

Microchip 社の Arm Cortex-M ベースのマイクロコントローラは CoreSight に準拠した OCD コンポーネントを実装 しています。これらの OCD コンポーネントの機能はデバイスによって異なります。詳細は各デバイスのデータ シートおよび Arm が提供する CoreSight の文書を参照してください。

4.2.1.2 Arm Cortex-M のタイプとデバッグ機能

現在、以下の Arm Cortex-M のタイプが SAM MCU で使われています。特定の SAM MCU で使うタイプを判別する には、該当デバイスのデータシートを参照してください。

Cortex-M のタイプ	デバッグのサポート	トレースの サポート
Cortex-M4、 M4F	デバッグ オプション: 基本的なデバッグ機能としてプロセッサの停止、シング ルステップ、プロセッサコア レジスタへのアクセス、ベクタ捕捉、無制限のソ フトウェア ブレークポイント、システムメモリへのフルアクセスがあります。 各種ブレークポイントおよび 1/4 ウォッチポイント機能も利用できます。	ITM、TPIU
Cortex-M7	Cortex-M7 のデバッグ機能にはプロセッサの停止、シングルステップ、プロ セッサコア レジスタへのアクセス、ベクタ捕捉、無制限のソフトウェア ブ レークポイント、システムメモリへのフルアクセスがあります。プロセッサに は、実装時に設定する 4/8 ハードウェア ブレークポイントと 2/4 ウォッチポイ ントのサポートも含まれます。	データトレース、 ITM、TPIU

4.2.2 tinyAVR、megaAVR[®]、XMEGA デバイス

AVR デバイスは各種プログラミングおよびデバッグ インターフェイスを備えています。該当デバイスでサポートさ れているインターフェイスについてはデバイスのデータシートを参照してください。

- 一部の tinyAVR デバイスは TPI インターフェイスを備えています。TPI はデバイスのプログラミングにのみ使え ます。これらのデバイスはオンチップ デバッグ機能を全く持っていません。
- 一部の tinyAVR デバイスと一部の megaAVR デバイスは、tinyOCD とも呼ばれるオンチップ デバッグシステムに接続する debugWIRE インターフェイスを備えています。
 debugWIRE を実装した全てのデバイスはインシステム プログラミング用の SPI インターフェイスも備えています。
- 一部の tinyAVR デバイスと一部の megaAVR デバイスは、tinyOCD とも呼ばれるオンチップ デバッグシステムに接続する debugWIRE インターフェイスを備えています。debugWIRE を実装した全てのデバイスはインシステム プログラミング用の SPI インターフェイスも備えています。
- 全ての AVR XMEGA デバイスはプログラミングおよびデバッグ用の PDI インターフェイスを備えています。
 一部の AVR XMEGA デバイスは同等の機能を持つ JTAG インターフェイスも備えています。
- 新しい tinyAVR デバイスにはプログラミングとデバッグに使う UPDI インターフェイスがあります。

	UPDI	TPI	SPI	debugWIRE	JTAG	PDI
tinyAVR	新しいデバイス	一部のデバイス	一部のデバイス	一部のデバイス		
megaAVR			全デバイス	一部のデバイス	一部のデバイス	
AVR XMEGA					一部のデバイス	全デバイス

表 4-1. プログラミングおよびデバッグ インターフェイスの概要

4.2.2.1 tinyAVR OCD の機能

新しいデバイス用の tinyAVR OCD は単線式プログラミング/デバッグ インターフェイスである UPDI 物理インター フェイスに基づいています。本 OCD は以下の特長を備えます。

- ・ デバイスアドレス空間(NVM、RAM、I/O)へのメモリマップ アクセス
- 無制限のデバイスクロック周波数
- ・ 無制限数のユーザプログラム ブレークポイント
- 2個のハードウェア ブレークポイント
- 高度な OCD 機能のサポート

- システムレジスタにアクセスしない非侵襲性(non-intrusive)の実行時デバイス監視
- ロック済みデバイスに対するフラッシュの CRC チェックの結果を読み出すためのインターフェイス

古いデバイス用の tinyAVR OCD は debugWIRE に基づいています。OCD 機能の詳細は 4.2.2.4.「debugWIRE OCD の機能」を参照してください。

4.2.2.1.1 TinyX-OCD (UPDI)に固有の注意事項

UPDI データピン(UPDI_DATA)はターゲット AVR デバイス応じて専用ピンにも共用ピンにもなります。共用 UPDI ピンは 12 V 耐圧で、/RESET または GPIO として使うために設定できます。これらの設定におけるこのピンの詳細 な使い方は 3.3.4.4.「AVR UPDI」を参照してください。

CRCSCAN(巡回冗長検査メモリスキャン) モジュールを実装したデバイスでは、デバッグ時にバックグラウンドで 繰り返しモードでこのモジュールを使わないでください。本 OCD モジュールのハードウェア ブレークポイント コ ンパレータのリソースは限られているため、より多くのブレークポイントが必要な場合またはソースレベルのコー ドのステップ実行時にも BREAK 命令がフラッシュに挿入される事があります(ソフトウェア ブレークポイント)。 本 CRC モジュールは、このブレークポイントをフラッシュメモリ コンテンツの破損として誤検出する可能性があ ります。

CRCSCAN モジュールは、ブート前に CRC スキャンを実行するように設定した状態で表示されます。CRC が不一 致の場合、デバイスは起動せずロック状態にあるように見えます。この状態からデバイスを回復させる唯一の方法 は、チップ全体の消去を実行して有効なフラッシュ イメージを書き込むかブート前 CRCSCAN を無効にする事で す(単純なチップ消去では無効な CRC による空のフラッシュとなるため、デバイスはまだブートしません)。この状 態でデバイス消去を実行すると、ソフトウェアのフロントエンドは CRCSCAN ヒューズを自動的に無効にします。

UPDI インターフェイスを使うターゲット アプリケーション PCB を設計する際は、適正動作を得るために以下を考慮する必要があります。

- UPDI ラインのプルアップ抵抗は 10 kΩ以下にしない事。UPDI を使う場合はプルダウン抵抗を使わないでください。または抵抗を取り外してください。UPDI 物理インターフェイスはプッシュプル対応であるため、ラインがアイドル状態(Hi-Z)の時にスタートビットの誤トリガを防ぐために 10KΩ 以上の抵抗によりプルアップしておく必要があります。
- UPDI ピンを RESET ピンとして使う場合、安定化用コンデンサは UPDI インターフェイスの適正動作を妨げる ため、UPDI 使用時には全て取り外す必要があります。
- UPDI ピンを RESET または GPIO ピンとして使う場合、外部ドライバはインターフェイスの適正動作を妨げる 可能性があるため、プログラミングまたはデバッグ時にはライン上の外部ドライバを全て取り外す必要があり ます。

4.2.2.1.2 TPI を備えた AVR デバイス

TPI (Tiny Programming Interface)は、OCD を備えていない tinyAVR デバイスにあります。これらのデバイスはデ バッグできません。TPI はプログラミング専用です。

4.2.2.2 megaAVR OCD の機能

megaAVR OCD は JTAG 物理インターフェイスに基づいています。本 OCD は以下の特長を備えます。

- 完全なプログラムフロー制御
- 全てのレジスタとメモリ領域へのフルアクセス
- 4個のプログラムメモリ (ハードウェア) ブレークポイント(このうち1つは予約済み)
- ハードウェア ブレークポイントを組み合わせてデータ ブレークポイントを形成可能
- 無制限数のプログラム ブレークポイント(BREAK を使用)(ATmega128[A]を除く)

4.2.2.2.1 megaAVR に固有の注意事項

ソフトウェア ブレークポイント

OCD モジュールの初期バージョンを実装しているため、ATmega128[A]はソフトウェア ブレークポイントのための BREAK 命令の使用をサポートしていません。

JTAG クロック

デバッグ セッションを開始する前にソフトウェアのフロントエンドでターゲット クロック周波数を正確に指定す る必要があります。同期上の理由から、信頼性の高いデバッグを行うために JTAG TCK 信号はターゲット クロッ

MPLAB[®] ICE 4インサーキット エミュレータ ユーザガイド 動作

ク周波数の 1/4 未満になる必要があります。JTAG インターフェイスでプログラミングする場合、TCK 周波数は実際に使われるクロック周波数ではなくターゲット デバイスの最大定格周波数の制約を受けます。

内部 RC オシレータを使う場合、周波数はデバイスによって異なる場合があり、温度と Vcc の変化の影響も受けます。ターゲット クロック周波数を指定する際には慎重に判断する必要があります。

OCDEN ヒューズ

megaAVR デバイスをデバッグするには OCDEN ヒューズを設定する必要があります(既定値では OCDEN は未設定 状態です)。これにより、OCD にアクセスして本デバイスをより簡単にデバッグできます。ソフトウェアのフロント エンドにより、OCDEN ヒューズはデバッグ セッション開始時に設定され、セッション終了時に未設定の状態となる ため、OCD モジュールによる不要な電力消費を抑えられます。

JTAGEN ヒューズ

JTAG インターフェイスは JTAGEN ヒューズを使うと有効になります。JTAGEN ヒューズは既定値で設定されてい ます。これにより、JTAG プログラミング インターフェイスにアクセスできます。



Important: JTAGEN ヒューズが意図せず無効になった場合、再度有効にできるのは SPI または高電圧 プログラミング方式を使った場合のみです。

JTAGEN ヒューズが設定されている場合でも、MCU 制御レジスタで JTAG ディセーブル ビットを設定すると、ファー ムウェアで JTAG インターフェイスを無効にできます。この操作ではコードをデバッグできなくなるため、デバッグ セッションを試みる際には行わないでください。デバッグ セッション開始時に Microchip 社の AVR デバイスでこうし たコードが既に実行されている場合、MPLAB ICE 4 は接続中に RESET ラインをアサートします。RESET ラインが正 しく配線されていれば、ターゲット AVR デバイスを強制的にリセットするため、JTAG 接続が可能になります。

JTAG インターフェイスが有効な場合、JTAG ピンを代替ピン機能に使う事はできません。プログラムコードから JTAG ディセーブル ビットを設定するかプログラミング インターフェイスから JTAGEN ヒューズをクリアして JTAG インターフェイスを無効にするまで、これらのピンは JTAG 専用ピンのままです。



Atmel Studio のプログラミング ダイアログとデバッグ オプション ダイアログで[use external reset] チェックボックスに必ずチェックを入れてください。これにより、JTAG ディセーブル ビットを設定 して JTAG インターフェイスを無効にするコードを実行しているデバイスで MPLAB ICE 4 は RESET ラインをアサートし、JTAG インターフェイスを再度有効にできます。

IDR/OCDR イベント

Tip:

IDR (In-out Data Register)は OCDR (On-Chip Debug Register)とも呼ばれ、デバッグ セッション中の停止モードの際 にデバッガが MCU に対して情報を読み書きするために広く使われています。ランモードのアプリケーション プログ ラムがデバッグ中の AVR デバイスの OCDR レジスタに 1 バイトのデータを書き込むと、MPLAB ICE 4 はこの値を 読み出してソフトウェア フロントエンドのメッセージ ウィンドウに表示します。OCDR レジスタは 50 ms ごとに ポーリングされるため、それ以上の頻度で書き込んでも信頼できる結果を得る事はできません。デバッグ中に AVR デバイスの電源が切れると、誤った OCDR イベントが報告される可能性があります。これは、ターゲット電圧が AVR の最小動作電圧を下回っても MPLAB ICE 4 がデバイスをポーリングする可能性があるために起こります。

4.2.2.3 AVR XMEGA OCD の機能

Atmel AVR XMEGA OCD は PDI (Program and Debug Interface)とも呼ばれます。2 つの物理インターフェイス (JTAG および PDI 物理インターフェイス)がデバイス内の同じ OCD 実装へのアクセスを提供します。本 OCD は以下 の特長を備えます。

- ・ 完全なプログラムフロー制御
- 全てのレジスタとメモリ領域へのフルアクセス
- 1個の専用プログラム アドレス コンパレータまたはシンボリック ブレークポイント(予約済み)
- 4個のハードウェアコンパレータ

MPLAB[®] ICE 4インサーキット エミュレータ ユーザガイド

- 無制限数のユーザプログラム ブレークポイント(BREAK 命令を使用)
- 無制限のシステムクロック周波数

4.2.2.3.1 AVR XMEGA に固有の注意事項

OCD とクロック同期

MCU が停止モードに移行すると、OCD クロックが MCU クロックとして使われます。OCD クロックは、JTAG イン ターフェイス使用時は JTAG TCK、PDI インターフェイス使用時は PDI_CLK です。

停止モードでの I/O モジュール

以前の Microchip 社 megaAVR デバイスとは対照的に、XMEGA では I/O モジュールは停止モードで停止します。つまり、USARTの送信が中断されてタイマ(と PWM)が停止します。

ハードウェア ブレークポイント

4 個のハードウェア ブレークポイント コンパレータ(2 個のアドレス コンパレータ、2 個の値コンパレータ)がありま す。以下の制約が課せられています。

- 全てのブレークポイントが同じタイプ (プログラムまたはデータ)である必要があります。
- 全てのデータ ブレークポイントが同じメモリ領域(I/O、SRAM、XRAM)にある必要があります。
- アドレスレンジを使う場合、ブレークポイントは1つしか設定できません。

以下に、設定可能な組み合わせを示します。

- 2個のシングルデータまたはプログラムアドレスのブレークポイント
- 1個のデータまたはプログラム アドレスレンジのブレークポイント
- 2個のシングルデータ アドレスのブレークポイント(単一値の比較による)
- 1個のデータ ブレークポイント (アドレスレンジ、値レンジのどちらかまたは両方)

MPLAB X IDE と Atmel Studio はブレークポイントを設定できない場合にその旨を通知し、その理由を示します。ソ フトウェア ブレークポイントを利用できる場合、データ ブレークポイントがプログラム ブレークポイントよりも優 先されます。

外部リセットと PDI 物理インターフェイス

PDI 物理インターフェイスはリセットラインをクロックとして使います。デバッグ時には RESET ピンのプルアップ抵 抗値を 10 kΩ以上にするか取り外す必要があります。リセット コンデンサは全て取り外す必要があります。その他 の外部リセット要因を切り離す必要があります。

JTAGEN ヒューズ

JTAG インターフェイスは JTAGEN ヒューズを使うと有効になります。JTAGEN ヒューズは既定値で設定されてい ます。これにより、JTAG プログラミング インターフェイスにアクセスできます。



Important: JTAGEN ヒューズが意図せず無効になった場合、再度有効にできるのは PDI 物理イン ターフェイスを使った場合のみです。

JTAGEN ヒューズが設定されている場合でも、MCU 制御レジスタで JTAG ディセーブル ビットを設定すると、 ファームウェアで JTAG インターフェイスを無効にできます。この操作ではコードをデバッグできなくなるため、デ バッグ セッションを試みる際には行わないでください。デバッグ セッション開始時に Microchip 社の AVR デバイス でこうしたコードが既に実行されている場合、MPLAB ICE 4 は接続中に RESET ラインをアサートします。RESET ラインが正しく配線されていれば、ターゲット AVR デバイスを強制的にリセットするため、JTAG 接続が可能にな ります。

JTAG インターフェイスが有効な場合、JTAG ピンを代替ピン機能に使う事はできません。プログラムコードから JTAG ディセーブル ビットを設定するかプログラミング インターフェイスから JTAGEN ヒューズをクリアして JTAG インターフェイスを無効にするまで、これらのピンは JTAG 専用ピンのままです。



Tip:

Atmel Studio のプログラミング ダイアログとデバッグ オプション ダイアログで[use external reset] チェックボックスに必ずチェックを入れてください。これにより、JTAG ディセーブル ビットを設定 して JTAG インターフェイスを無効にするコードを実行しているデバイスで MPLAB ICE 4 は RESET ラインをアサートし、JTAG インターフェイスを再度有効にできます。

ATxmegaA1 rev H 以前のスリープのデバッグ

ATxmegaA1 デバイスの旧バージョンには、デバイスが特定のスリープモードにある時に OCD を有効にできないバ グがありました。OCD を再度有効にするための回避策は 2 つあります。

- MPLAB ICE 4 に移動する。[Tools]メニューの[Options]で[Always activate external Reset when reprogramming device]を有効にする
- チップ消去を実行する

このバグをトリガするスリープモードは以下の通りです。

- ・ パワーダウン
- ・ パワーセーブ
- ・ スタンバイ
- 拡張スタンバイ

4.2.2.4 debugWIRE OCD の機能

debugWIRE OCD は、少ピン AVR デバイス向けに特化して機能を限定した専用 OCD モジュールです。本 OCD は以下の特長を備えます。

- ・ 完全なプログラムフロー制御
- 全てのレジスタとメモリ領域へのフルアクセス
- 無制限のユーザプログラム ブレークポイント(BREAK 命令を使用)
- ターゲット クロックに基づく自動 baud レート設定

4.2.2.4.1 debugWIRE に固有の注意事項

debugWIRE 通信ピン(dW)は物理的に外部リセット(RESET)と同じピンに配置されています。従って、debugWIRE インターフェイスが有効な時は外部リセット要因はサポートされません。

debugWIRE インターフェイスが機能するには、ターゲット デバイス上に debugWIRE イネーブル(DWEN)ヒューズ を設定する必要があります。Microchip 社 AVR デバイスの工場出荷時の既定値では、このヒューズは設定されていま せん。debugWIRE インターフェイス自体を使ってこのヒューズを設定する事はできません。DWEN ヒューズを設定 するには SPI モードを使う必要があります。必要な SPI ピンが接続されていれば、ソフトウェアのフロントエンド が自動的にこれを処理します。ソフトウェアのフロントエンドで SPI プログラミングを使って手動で設定する事もで きます。

- 方法 1: debugWIRE 側でデバッグ セッションの開始を試みます。debugWIRE インターフェイスが有効でない場合、 ソフトウェアのフロントエンドは再試行を提案するか、SPI プログラミングを使って debugWIRE の有効化 を試みます。フル SPI ヘッダが接続されている場合、debugWIRE が有効になり、ターゲットの電源をトグ ルするように指示されます。これはヒューズの変更を有効にするために必要な操作です。
- **方法 2:** SPI モードで Atmel Studio のプログラミング ダイアログを開き、署名が正しいデバイスと一致する事を確認します。DWEN ヒューズにチェックを入れて debugWIRE を有効にします。



Important:

SPIEN ヒューズを設定状態、RSTDISBL ヒューズを未設定状態にしておく事が重要です。この状態に しておかないとデバイスは debugWIRE モードに固定され、DWEN の設定を復帰させるには高電圧プ ログラミングが必要になります。

MPLAB[®] ICE 4インサーキット エミュレータ ユーザガイド 動作

debugWIRE インターフェイスを無効にするには、高電圧プログラミングを使って DWEN ヒューズを未設定状態に します。または、debugWIRE インターフェイスからインターフェイス自体を一時的に無効にします。これにより、 SPIEN ヒューズが設定されている場合は SPI プログラミングを実行できます。



Important:

SPIEN ヒューズが設定状態でない場合、ソフトウェアのフロントエンドはこの操作を完了できないため、高電圧プログラミングを使う必要があります。

MPLAB X IDE では、ターゲット デバイスで debugWIRE が有効になっていて SPI プログラミング セッションが試み られる場合、MPLAB は最初に debugWIRE を無効にするように提案します。Atmel Studio では、デバッグ セッショ ン中に[Debug]メニューから[Disable debugWIRE and Close]メニュー オプションを選択して debugWIRE を手動で無 効にする必要があります。これで debugWIRE が一時的に無効となり、ソフトウェアのフロントエンドは SPI プログ ラミングを使って DWEN ヒューズを未設定状態にします。

DWEN ヒューズを設定する事でクロックシステムの一部は全てのスリープモード中に動作します。これにより、ス リープモード中の AVR の消費電力は増加します。このため、debugWIRE を使わない時は DWEN ヒューズを常に無 効にする必要があります。

debugWIRE を使うターゲット アプリケーション PCB を設計する際は、適正動作を得るために以下を考慮する必要 があります。

- dW/(RESET)ラインのプルアップ抵抗を 10 kΩ未満にしてはなりません。デバッガツールがプルアップ抵抗を 提供するため、debugWIRE 機能にはプルアップ抵抗は不要です。
- 安定化用コンデンサはインターフェイスの適正動作を妨げるため、debugWIRE を使う時は RESET ピンに接続 した安定化用コンデンサを全て取り外す必要があります。
- 外部リセット要因または他のアクティブなドライバはインターフェイスの適正動作を妨げる可能性があるため、 RESET ライン上の外部リセット要因またはアクティブドライバを全て取り除く必要があります。

ターゲット デバイスでロックビットをプログラムしないでください。debugWIRE インターフェイスが適正に動作するにはロックビットがクリアされている事が必要です。

4.2.2.4.2 debugWIRE ソフトウェア ブレークポイント

debugWIRE OCD は、Microchip 社の megaAVR (JTAG) OCD と比較すると大幅にスケールダウンしています。これ は、デバッグ用にユーザが利用できるプログラム カウンタ ブレークポイント コンパレータがない事を意味します。 カーソル位置までの実行およびシングルステップ実行操作のためにこのようなコンパレータが1つ存在しますが、追 加のユーザ ブレークポイントはハードウェアではサポートされていません。

代わりに、デバッガは AVR BREAK 命令を利用する必要があります。この命令はフラッシュに配置でき、実行のた めに読み込まれると、AVR CPU が停止モードに移行します。デバッグ中のブレークポイントをサポートするために、 デバッガはユーザがブレークポイントを要求するポイントで BREAK 命令をフラッシュに挿入する必要があります。 元の命令は後で置き換えるためにキャッシュする必要があります。シングルステップ実行で BREAK 命令があると、 デバッガはプログラムの動作を維持するために元のキャッシュされた命令を実行する必要があります。極端なケース では、BREAK をフラッシュから削除し、後で再配置しなければならない事もあります。

これらのシナリオはブレークポイントからのシングルステップ実行時に明らかな遅延を引き起こし、ターゲット クロック周波数が非常に低い場合にはさらに悪化します。

そのため、可能であれば以下のガイドラインに従う事を推奨します。

- デバッグ中は常にできるだけ高い周波数でターゲットを実行してください。debugWIRE 物理インターフェイス にはターゲット クロックからクロックが供給されます。
- ブレークポイントの追加と削除のたびにターゲット上にフラッシュページを再配置する必要があるため、追加 と削除は最小限の回数で行ってください。
- フラッシュページの書き込み操作の回数を最小限にするために、一度に追加または削除するブレークポイントの数は最小限に抑えてください。
- 可能であればダブルワード命令上にブレークポイントを配置する事は避けてください。

4.2.2.4.3 debugWIRE と DWEN ヒューズの内容

debugWIRE インターフェイスはデバイスの/RESET ピンを制御し、同じくこのピンを必要とする SPI インターフェ イスとは相互排他的です。debugWIRE モジュールを有効および無効にする場合、以下の 2 つの方法のどちらかに従 います。

MPLAB® ICE 4インサーキット エミュレータ ユーザガイド

動作

• ソフトウェアのフロントエンドに処理を任せる(推奨)

• DWEN を手動で設定およびクリアする(上級ユーザ限定の操作のため注意する事)



Important: DWEN を手動で操作する場合、高電圧プログラミングを使わなくても済むように SPIEN ヒューズを設定状態に維持する必要があります。

図 4-1. debugWIRE と DWEN ヒューズの内容



4.2.2.5 高度なデバッグ(AVR JTAG/debugWIRE デバイス)

I/O モジュール

ブレークポイントによってプログラムの実行が停止しても、大半の I/O モジュールは実行を続けます。例: UART の 送信中にブレークポイントに到達した場合、送信が完了し、対応するビットがセットされます。実際のデバイスで は後で発生しますが、TXC(送信完了)フラグが設定され、コードの次のシングルステップで利用可能になります。

以下の2つの例外を除く全てのI/Oモジュールは停止モードで実行を続けます。

- タイマ/カウンタ (ソフトウェアのフロントエンドを使って設定可能)
- ウォッチドッグ タイマ (デバッグ中のリセットを避けるために常に停止)

I/O アクセスのシングルステップ実行

I/O は停止モードで実行を続けるため、タイミングに関する特定の問題を避けるために注意が必要です。例えば、 以下のコードがあるとします。

OUT PORTB, 0xAA IN TEMP, PINB

このコードを通常通りに実行した場合、TEMP レジスタは 0xAA を読み戻しません。これは、IN 操作によってサン プリングされる時点ではデータがまだ物理的にピンにラッチされていないためです。PIN レジスタに正しい値を確 実に入れるために OUT 命令と IN 命令の間に NOP 命令を配置する必要があります。

ただし、OCD の間中この機能をシングルステップ実行する場合、シングルステップ実行中にコアが停止しても I/O はフルスピードで動作しているため、このコードは PIN レジスタに常に 0xAA を提供します。

シングルステップ実行とタイミング

制御信号を有効にしたら、所定のサイクル数以内に特定のレジスタを読み書きする必要があります。停止モードで も I/O クロックと I/O モジュールはフルスピードで実行を続けるため、このようなコードをシングルステップで実行 してもタイミング要件は満たされません。2 つのシングルステップの間に I/O クロックが何百万サイクルも動作して いる可能性があります。

このようなタイミング要件でレジスタを正常に読み書きするには、デバイスをフルスピードで動作させるアトミック動作として読み書きシーケンス全体を実行する必要があります。これには、マクロまたは関数コールを使って コードを実行するか、デバッグ環境でカーソル位置までの実行機能を使います。

16 ビットレジスタへのアクセス

通常、Microchip AVR モジュールは 8 ビットデータバスを通じてアクセス可能な複数の 16 ビットレジスタを実装し ています(例えば 16 ビットタイマの TCNTn)。この 16 ビットレジスタには、2 回の読み出しまたは書き込み動作を 使ってバイト単位でアクセスする必要があります。16 ビットアクセスの途中でブレークしたり、この状況でシング ルステップ実行すると、誤った値になる可能性があります。

制限された I/O レジスタアクセス

一部のレジスタは内容に影響を与えずに読み出す事はできません。このようなレジスタには、読み出しによってクリアされるフラグを格納するものやバッファ付きデータレジスタ(UDR等)があります。OCDデバッグで想定されている非侵入型の性質を維持するために、ソフトウェアのフロントエンドは停止モードではこれらのレジスタの読み出しを避けます。また、一部のレジスタは副作用なしで安全に書き込む事ができません。これらのレジスタは読み出し専用です。以下に例を示します。

- 1 を任意のビットに書き込む事でフラグがクリアされるフラグレジスタ。これらのレジスタは読み出し専用です。
- モジュールの状態に影響を与えずに UDR および SPDR レジスタを読み出す事はできません。これらのレジス タにはアクセスできません。

4.3 PIC MCU/dsPIC DSC - オンチップ デバッグ

オンチップ デバッグ モジュールは、通常は**デバッガ**または**デバッグアダプタ**と呼ばれるデバイスを介して開発者が 外部の開発プラットフォームからデバイス上の実行を監視、制御できるシステムです。OCD システムでは、(シ ミュレータとは異なり)ターゲット システムの電気的およびタイミング特性を正確に維持しながらアプリケーション を実行できます。OCD システムは条件付きまたは手動で実行を停止し、プログラムフローとメモリを調べられます。

PIC MCU(マイクロコントローラ)または dsPIC DSC(デジタルシグナル コントローラ)の場合、一部のデバイスリ ソースをデバッグ用に確保しなければならない場合があります。

4.3.1 エミュレータの基本機能

MPLAB ICE 4 インサーキット エミュレータには以下の基本的なデバッグ機能があります。

4.3.1.1 エミュレーションの開始と停止

MPLAB X IDE 内でアプリケーションをデバッグするには、ソースコードを含むプロジェクトを作成する必要があり ます。プロジェクトを作成した後は、以下の手順でコードをビルドしてデバイスにプログラミングし、これを実行 できます。

MPLAB[®] ICE 4インサーキット エミュレータ ユーザガイド

	デバッグモードでプロジェクト コードをデバッグまたは実行する
	コードの実行を一時停止または停止する
	一時停止または停止の後にコード実行を継続する
	ー時停止/停止したコードに対して1つの命令をステップインまたは実行する。スリープ命令をステップ 実行しないように注意してください。これを行った場合、エミュレーションを再開するにはプロセッサ をリセットする必要があります
ß	一時停止/停止したコードに対して命令をステップオーバーする
	デバッグ セッションを完了する事でコード実行を終了する
	プロセッサをリセットする。デバイスによっては POR/BOR、MCLR、システムリセット等のリセットも 実行できます。

4.3.1.2 プロセッサメモリとソースコードの表示

MPLAB X IDE はデバッグ情報や各種プロセッサメモリ情報を表示する複数のウィンドウを備えています。これらのウィンドウは[Window]メニューから選択できます。これらのウィンドウの詳細な使い方については、MPLAB X IDE のオンラインヘルプを参照してください。

- [Window] > [Target Memory Views] 各種デバイスメモリを表示します。選択したデバイスに応じてプログラ ムメモリ、ファイルレジスタ、コンフィグレーションメモリ等が表示されます。
- [Window] > [Debugging] デバッグ情報を表示します。表示する内容は変数、ウォッチ、コールスタック、ブレークポイント、ストップウォッチ、トレースの中から選択できます。

[Projects]ウィンドウ内で、いずれかのソースコード ファイルをダブルクリックすると、[Files]ウィンドウにその ソースコードが表示されます。このウィンドウには、選択したプロセッサとビルドツールに応じて、色分けされた コードが表示されます。色分けのスタイルを変更するには[Tools] > [Options] > [Fonts & Colors] > [Syntax]をクリッ クします。

エディタの詳細は MPLAB X IDE オンラインヘルプの「Editor」を参照してください。

4.3.1.3 ブレークポイントの使い方

ブレークポイントを使うと、指定した行でコード実行を停止できます。

4.3.1.3.1 ブレークポイントの数

16 ビット PIC/dsPIC デバイスの場合、ブレークポイント、データキャプチャ、ランタイム ウォッチは同じリソー スを使います。従って、実際に使えるブレークポイントの数は、ブレークポイントとトリガの合計数によって制限 されます。

32 ビット PIC デバイスの場合、ブレークポイントとデータキャプチャ、ランタイム ウォッチは別のリソースを使います。従って、使えるブレークポイントの数はトリガの使用数と無関係です。

使用可能または使用中のハードウェア/ソフトウェア ブレークポイントの数は、[Dashboard]ウィンドウ(<u>[Window]></u> [<u>Dashboard]</u>)に表示されます。詳細は MPLAB X IDE ユーザガイドを参照してください。ソフトウェア ブレークポ イントは全てのデバイスで使える訳ではありません。

デバイスあたりのハードウェア ブレークポイントの数、ハードウェア ブレークポイントのスキッド数等、ブレー クポイント動作の制約は、「<u>microchipdeveloper.com サイトの(Debug Limitations - PIC® MCUs)</u>」を参照して ください。

4.3.1.3.2 ハードウェア/ソフトウェア ブレークポイントの選択

ハードウェアまたはソフトウェア ブレークポイントを選択する手順は以下の通りです。

1. [Projects]ウィンドウ内のプロジェクトを選択し、右クリックして[Properties]を選択する

MPLAB[®] ICE 4インサーキット エミュレータ ユーザガイド 動作

- 2. [Project Properties]ダイアログで[ICE4] > [Debug Options]を選択する
- 3. ソフトウェア ブレークポイントを使う場合、[Use software breakpoints]にチェックを入れる。ハードウェア ブレークポイントを使う場合、このチェックを外す

Note: デバッグにソフトウェア ブレークポイントを使うと、デバイスの書き込み耐性に影響します。従って、ソフ トウェア ブレークポイントを使ったデバイスは量産製品に使わない事を推奨します。

下表に、ハードウェア ブレークポイントとソフトウェア ブレークポイントの比較を示します。どちらのブレーク ポイントを使うべきか判断する際の参考にしてください。

表 4-2. ハードウェア ブレークポイントとソフトウェア ブレークポイント

特長	ハードウェア ブレークポイント	ソフトウェア ブレークポイント		
ブレークポイントの数	制限あり	制限なし		
ブレークポイントの書き込み先*	内部デバッグレジスタ	フラッシュ プログラムメモリ		
ブレークポイントの適用先**	プログラムメモリとデータメモリ	プログラムメモリのみ		
ブレークポイントの設定に要する 時間	最小限	オシレータ速度、フラッシュメモリへの書 き込み時間、ページサイズによって決まる		
ブレークポイントのスキッド	ほとんどのデバイスで発生する。 詳細はオンラインヘルプの 「Limitations」を参照してくださ い。	発生しない		
* ブレークポイントに関する情報が書き込まれるデバイス内の場所				
** ブレークポイントが適用されるデバイス機能の種類(ブレークポイントが設定される場所)				

4.3.1.4 ストップウォッチの使用

ストップウォッチを使うと、2つのブレークポイント間の時間を調べられます。

Note: ストップウォッチはブレークポイント リソースを使います。

ストップウォッチの使い方

- 1. ストップウォッチを開始するブレークポイントを追加します。
- 2. ストップウォッチを停止するブレークポイントを追加します。
- 3. <u>[Window] > [Debugging] > [Stopwatch]</u>を選択し、ウィンドウの左にある**[Properties]**アイコンをクリックし、 開始および停止ブレークポイントを選択します。
- 4. 再度プログラムをデバッグすると、ストップウォッチの結果が表示されます。

図 4-2. ストップウォッチの設定

Set Stopwatch Properties	Х				
Start Condition Select a current breakpoint or trigger condition to start the stopwatch					
Line C:\Users\c08227\MPLABXProjects\PIC32MX470CuriosityTricolor.X\main.c:59 $ \smallsetminus$					
Start condition will cause the target device to halt					
Stop Condition Select a current breakpoint or trigger condition to stop the stopwatch Line C:\Users\c08227\MPLABXProjects\PIC32MX470CuriosityTricolor.X\main.c:74 ~					
Stop condition will cause the target device to halt					
OK Cancel Help					

図 4-3. [Stopwatch]ウィンドウの内容

: Stop	watch	₿ %	Watches	Variables	Breakpoints	Output	Tasks
%	Target halt Target halt	ed. Sto ed. Sto	pwatch cycle count = 0 pwatch cycle count = 550	3			
	Cycle count	= 550	3				

[Stopwatch]ウィンドウの左側には以下のアイコンが表示されます。

表 4-3. ストップウォッチ アイコン

アイコン	アイコンテキスト	表記
*	Properties	ストップウォッチのプロパティを設定します。現在のブレークポイントまた はトリガの1つを始点、もう1つを終点として選択します。
Ö	Reset Stopwatch on Run	実行時にストップウォッチの時間を0にリセットします。
	Clear History	[Stopwatch]ウィンドウをクリアします。
٢	Clear Stopwatch	(シミュレータのみ)デバイスリセット後にストップウォッチをリセットします。

4.3.1.5 [Freeze Peripherals]の設定

[Freeze on Halt]を選択すると、選択した周辺モジュールを停止時にフリーズできます。これらの機能の詳細は 9.2.5. 「[Freeze Peripherals]」を参照してください。

4.3.2 ICSP デバッグ

MPLAB ICE 4 インサーキット エミュレータをデバッガとして使うには、2 段階の手順が必要です。まず、アプリ ケーション コードをターゲット デバイスにプログラミングします(MPLAB ICE 4 をプログラマとして使用)。次に、 ターゲットのフラッシュ デバイスが内蔵するインサーキット デバッグ ハードウェアを使ってアプリケーション コードを実行(テスト)します。これら 2 つの手順は、MPLAB X IDE 内での以下の操作に対応します。

MPLAB[®] ICE 4インサーキット エミュレータ ユーザガイド 動作

1. コードをターゲットにプログラミングし、デバッグ機能を有効にする(詳細は次セクション参照)。

2. デバッガを使ってブレークポイントを設定し、コードを実行する。

詳細は MPLAB X IDE オンラインヘルプを参照してください。

ターゲット デバイスにコードを正しくプログラミングできない場合、MPLAB ICE 4 によるデバッグはできません。

下図に、MPLAB ICE4 の内部インターフェイス回路の一部を簡単に示します。この図では Rpu = 10 kΩ (typ.)、 Ric = 4.7 kΩです。

図 4-4. ICSP プログラミングのための接続



プログラミングの場合、ターゲット デバイスにクロックを供給する必要はありませんが、電源は供給する必要があ ります。プログラミング時、デバッガは VPP/MCLR ラインをプログラミング レベルに設定し、PGC にクロックパ ルス、PGD にシリアルデータを出力します。デバイスが正しくプログラミングされた事を確認するため、デバッガ は PGC ヘクロックを供給し、PGD からデータを読み戻します。このシーケンスにより、デバッガとデバイスが正 しく通信している事を確認します。

4.3.2.1 エミュレータの動作を妨げる回路

下図は、デバッグ用のラインに MPLAB ICE 4 インサーキット エミュレータの正常動作を妨げる部品が取り付けら れた状態を示しています。



図 4-5. 不適切な回路部品

To Emulator

特に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わない - 電圧レベルが低下します。

- PGC/PGD にコンデンサを使わない プログラミング/デバッグ中のデータラインとクロックラインの高速な 遷移が妨げられ、プログラミング時間が長くなります。
- MCLR にコンデンサを使わない VPP の高速な遷移が妨げられます。通常は単純なプルアップ抵抗で十分です。
- PGC/PGD にダイオードを使わない デバッガとターゲット デバイス間の双方向通信が妨げられます。

4.3.2.2 デバッグ開始前の準備手順

4.3.2.4.「デバッグに関する要件」を満たしていれば、MPLAB X IDE 内で MPLAB ICE 4 インサーキット エミュ レータをツールとして設定します。[File] > [Project Properties]を選択してダイアログを開き、[Hardware Tool]で **[ICE 4]**をクリックします。

以下の操作を実行できるようになります。

- [Debug] > [Debug Main Project]を選択すると、前述のように ICSP プロトコルに従って、アプリケーション コードがデバイスのメモリに書き込まれます。
- ターゲット デバイスのプログラムメモリの上位領域には、小さな「デバッグ実行プログラム」コードが書き 込まれます。デバッグ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 コードからこの予約済み空間を使えません。デバイスによっては、デバッグ実行プログラム専用の特別なメモ リ空間を備えています。詳細はデバイスのデータシートで確認してください。
- MPLAB X IDE は、ターゲット デバイスの「インサーキット デバッグ」専用レジスタを有効にします。これらにより、デバッガからデバッグ実行プログラムを実行させる事が可能となります。デバイスの予約済みリソースの詳細は 4.3.2.5.「デバッガが使うリソース」を参照してください。
- ターゲット デバイスはデバッグモードで動作します。

4.3.2.3 デバッグの詳細

下図に、デバッグの準備が整った MPLAB ICE 4 インサーキット エミュレーション システムを示します。この図で は Rpu = 10 k Ω (typ.)、Ric = 4.7 k Ω です。

図 4-6. デバッグの準備が整った MPLAB ICE 4 - PIC MCU



通常は、アプリケーション コードが正しく動作するかどうかを確認するために、コードの最初の方にブレークポイントを設定します。MPLAB X IDE のユーザ インターフェイスから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書き込まれます。ブレークポイントのアドレスは、PGC および PGD ピンを介してこれらのレジスタへ直接書き込まれます。

次に、通常は MPLAB X IDE で[<u>Debug] > [Debug Main Project]</u>を選択します。デバッガは、デバッグ実行プログラ ムの実行を指令します。ターゲットはリセットベクタから実行を開始し、プログラム カウンタが内部デバッグレジ スタに書き込まれているブレークポイント アドレスに達すると停止します。

MPLAB[®] ICE 4インサーキット エミュレータ ユーザガイド 動作

ブレークポイント アドレスの命令が実行されるとターゲット デバイスのインサーキット デバッグ メカニズムがト リガされ、デバイスのプログラム カウンタがデバッグ エグゼクティブに渡されます(割り込みに似た動作)。この時 点で、ユーザ アプリケーションの実行は停止します。デバッガは PGC と PGD を介してデバッグ実行プログラムと 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 X IDE に返します。この後、MPLAB X IDE は一連 のクエリをデバッガに向けて送信する事で、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。これらのクエリは、最終的にデバッグ実行プログラムによって実行されます。

デバッグ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同じように動作し、一時変数を保存するため にスタックの一部を使います。何らかの理由(オシレータが動作していない、電源接続の不良、ターゲットボードの 短絡等)でデバイスが動作しない場合、デバッグ実行プログラムは MPLAB ICE 4 と通信できません。この場合、 MPLAB X IDE はエラーメッセージを表示します。

[Debug] > [Pause]を選択する事で、ブレークポイントを設定する事もできます。この場合、PGC ラインと PGD ラ インのトグルによってターゲット デバイスのインサーキット デバッグ メカニズムがトリガされ、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グ実行プログラムへ切り換わります。この場合もターゲット アプリケーションのコード実行が停止し、MPLAB X IDE はデバッガとデバッグ実行プログラム間の通信を使って ターゲット デバイスの状態を取得します。

4.3.2.4 デバッグに関する要件

MPLAB ICE 4 インサーキット エミュレーション システムによるデバッグ (ブレークポイントの設定、レジスタの参照等)では、以下を正しく行う必要があります。

- 本エミュレータは給電され、コンピュータに接続し、MPLAB X IDE ソフトウェアと通信している必要があります。
- ターゲット デバイスは給電されて動作可能(オシレータが動作中)である事が必要です。いかなる場合も、ター ゲット デバイスが動作しない限り MPLAB ICE 4 インサーキット エミュレータによるデバッグは行えません。
- ターゲット デバイスのコンフィグレーション ワードは、以下のように正しく設定しておく必要があります。 コンフィグレーション ワードの設定にはコードを使うか、MPLAB X IDE の[Configuration Bits]ウィンドウで行います。
 - オシレータ コンフィグレーション ビットは、ターゲットで利用可能なオシレータのタイプに応じた値に設 定する必要があります。
 - 既定値でウォッチドッグタイマが有効になるデバイスでは、ウォッチドッグタイマを無効にする必要があります。
 - ターゲット デバイスのコード保護は全種類無効にする必要があります。
 - ターゲット デバイスのテーブル読み出し保護は無効にする必要があります。
- PGC/PGD ペアを複数個備えるデバイスでは、デバイスのコンフィグレーション ワードの設定で正しいペアを 選択する必要があります。これはデバッグに対してのみ適用されます。プログラミングは、どの PGC/PGD ペ アでも正常に動作します。

4.3.2.5 デバッガが使うリソース

ー部のデバイスではデバッグ用にデバイスリソースを使う必要があります。MPLAB ICE 4 インサーキット エミュ レータのデバッガが使うターゲット デバイス上のリソースの詳細については MPLABX IDE メニュー、[Help] > [<u>Release Notes]</u>を参照してください。「Release Notes/Readmes」に関するセクションに加え、「Reserved Resources」のセクションがあります。[Reserved Resources by Device Family and Tool]または[Reserved Resources by Device for All Tools]を選択します。

4.3.2.6 プログラミング

Note: プログラミングの詳細はオンラインヘルプを参照してください。

MPLAB X ICE で MPLAB ICE 4 をプログラマとして使うと、ヘッダボードに実装済みの-ICE/-ICD 以外のデバイス にプログラミングできます。MPLAB ICE 4 をツールとして選択し(ナビゲーション ウィンドウで[Debug Tool ICE 4] をクリックし、メインメニューから[File] > [Project Properties]を選択してダイアログを開き、[Hardware Tool]で [ICE 4]をクリック)、以下の操作を実行します。

 [Run] > [Run Main Project]アイコン(下図参照)を選択すると、アプリケーション コードは ICSP プロトコルに よってデバイスのメモリに書き込まれます。プログラミング中にクロックは不要です。また、コード保護、 ウォッチドッグ タイマの有効化、テーブル読み出し保護等、プロセッサの全てのモードを設定できます。



[Run Main Project]アイコン

- 一部のターゲット デバイスでは、プログラムメモリの上位空間に小さな「プログラム実行プログラム」コードが書き込まれます。
- MPLAB X IDE はターゲット デバイスの「インサーキット デバッグ」専用レジスタと全てのデバッグ機能を 無効にします。これは、ブレークポイントの設定とレジスタの内容の参照または変更ができなくなる事を意 味します。
- ターゲット デバイスはリリースモードで動作します。プログラマとして使う場合、MPLAB ICD 4 インサー キット デバッガは MCLR ラインをトグルする(ターゲット デバイスをリセット/起動する)事しかできません。

4.3.3 PIC32M JTAG によるデバッグ

PIC32M MCU をデバッグ用に設定するには 3.3.8.「PIC32 命令トレース アダプタボード」に記載されているアダプ タボードまたは 40 ピンハイスピード ケーブル (トレースあり/なし)を使います。

4.3.3.1 PIC32M タイプの JTAG

JTAG 方式(PIC32M MCU):

- 2 線式 JTAG: Microchip 社独自の JTAG
- 4 線式 JTAG: 従来型 JTAG (IEEE1149.1)

2 線式 JTAG とは

2線式 JTAG は Microchip 社が定義した独自のデバッグ インターフェイスです。内部的には 4 線式 JTAG デバイス に基づき、外部へは TMS、TDI、TDO JTAG ラインを 1 本のデータラインに多重化して、内部へはそのデータライ ンを逆多重化する 2 線式コンバータを使います。デバイス内に特殊な変換ユニットが必要です。古いデバイスでは、 この変換ユニットはデバイスが特殊な 2 線式プログラミング モードになっている場合にこそ有効になるものの、2 線式プロトコルを使った完全なデバッグは実行できませんでした。

4.3.3.2 PIC32M デバッグ機能

基本的なデバッグ

以下の基本的なデバッグ機能を利用できます。

- Run、Halt、Reset
- Step Into、Step Over

ハードウェア ブレークポイント

HW(ハードウェア) ブレークポイントを利用できます。ブレークポイント数は使用デバイスによって異なります。

• PIC32: 6

ソフトウェア/フラッシュ ブレークポイント

SW(ソフトウェア)または「フラッシュ」ブレークポイントの対応については以下の通りです。

- Linux および Mac OS: フラッシュ ブレークポイントは J-Link BASE モデルではサポートされていません。
- Windows OS: 全ての HW ブレークポイントを使うと、[Flash Breakpoints Notice]ダイアログが開きます。下 表のいずれかのボタンをクリックして処理を進めます。

表 4-4. [Flash Breakpoint Notice]ダイアログ

ボタン	動作
Yes	トライアルベースで無制限数のフラッシュ ブレークポイントを使う
発生しない	無制限数のフラッシュ ブレークポイントをサポートするライセンス取得用のウェブ サイトに移動する
Install existing license	無制限数のフラッシュ ブレークポイントをサポートする既存のライセンスをインス トールする

5. 本エミュレータの特長

基本的なデバッグ機能に加え、本エミュレータは高度なデバッグ機能またはデバッグ機能の実装を可能としています。一部のデバッグ機能は他のデバッグ機能に依存します。

現在のデバイスで特定のデバッグ機能を利用できるかどうかを確認するには、以下の手順を実行します。

- MPLAB X IDE の[Help]メニューで[Release Notes]を選択する
- ・ [Release Notes/Readmes]に加えて[Debug Features Support]を検索する
- ・ [Hardware Tool Debug Features by Device]のリンクをクリックする

下表に、各接続タイプでサポートされるエミュレータの機能を示します。

衣 5-1. サホートされているエミュレータの 特

特長	マイクロコントローラ ファミリ*					
	PIC10/12/16 (8 ビット)	PIC24、dsPIC (16 ビット)	PIC32 (32 ビット)	AVR (8 ビット)	SAM (32 ビット)	
仮想 COM ポート	х	х	х	х	Х	
DGI	х	x	x	х	Х	
基本的なデバッグ	х	х	х	х	Х	
データキャプチャ、 ランタイム ウォッチ	Х	Х	х			
PIC 計測機能付き トレース	x	x				
PIC32M 命令トレース			х			
SAM ITM/SWO トレース					х	
PC サンプリング		Х				
PC プロファイリング			x			
デバッガ ポーリング				х	Х	
電源監視	x		x	x	x	
* - > !					1-	

*ファミリの一部のデバイスではサポートしていません。詳細は「Features by Device」を参照してください。

5.1 USB CDC 仮想 COM ポート

ターゲット UART と USB インターフェイスの間のブリッジを提供し、ターゲットの実際の UART への読み書きア クセスである USB ホスト上の CDC 仮想「COM」を提供します。

CI/CD でコンテナを使う場合、この仮想ポートを使って本エミュレータにアクセスします。

CDC/DGI U(S)ART でサポートされている baud レート: 7200、9600、14400、19200、28800、38400、57600、 76800、115200、230400、460800、921600

5.2 DGI

DGI (Data Gateway Interface)はターゲット MCU からのデータのローレベル転送を処理するためのインターフェイ スです。DGI は各種のツールとオンボード デバッガで利用可能です。

DGI は、設定と通信に同じ API を使うインターフェイスを複数提供します。各インターフェイスは SPI や UART 等の物理通信インターフェイスの抽象化を実装している場合と、タイムスタンプ インターフェイス等の物理通信イ ンターフェイスに直接接続されないサービスを表している場合があります。

MPLAB Data Visualizer は DGI の制御とストリーミングに使うアプリケーションです。このアプリケーションは、 MPLAB X IDE 内からアクセスする事もスタンドアロン プログラムとして使う事もできます。このアプリケーショ ンと DGI の詳細は MPLAB Data Visualizer のウェブページを参照してください。

5.2.1 インターフェイス

DGI の全ての機能は実装されたインターフェイスを軸とします。全てのインターフェイスは同じ USB プロトコル を使いますが、設定パラメータと通信の処理がインターフェイスごとに異なります。詳細は各インターフェイスの セクションを参照してください。DGI デバイスを実装している全てのボードで全てのインターフェイスが使える訳 ではありません。利用可能なインターフェイスは USB プロトコルから読み出す事ができます。

名称	ID	説明
タイムスタンプ	0x00	関連するインターフェイスで受信した全てのイベントにタイムスタンプを付加す るサービス インターフェイスです。
SPI*	0x20	クライアント モードで SPI を介して直接通信します。
UART USART*	0x21	クライアント モードで UART を介して直接通信します。クライアント モードで USART を介して直接通信します。
l ² C*	0x22	クライアント モードで I ² C を介して直接通信します。
GPIO	0x30	GPIO ピンの状態を監視、制御します。
電源	0x40 (データ)	接続した電力計測コプロセッサからデータと同期イベントを受信します。
デバッガ ポーリング	0x50	プログラム カウンタのタイムスタンプ付きサンプルをポーリングする事でデバイ スのプログラム実行を把握できます。 詳細は 5.10.「デバッガ ポーリング」を参照してください。
予約済み	0xFF	インターフェイスがない事を示すための特殊な識別子です。
* 次回のリリースで予算	 定	

表 5-2.インターフェイス一覧

5.2.1.1 タイムスタンプ

タイムスタンプ インターフェイス経由で返されるデータは、タイムスタンプを有効にしたインターフェイスに属す るデータのタイムスタンプ付きパケットのシーケンシャル ストリームです。各パケットの最初のバイトはインター フェイス識別子で、残りのパケットをどのように解析するかを決定します。

タイムスタンプは、サンプリングされて各パケットに埋め込まれる 16 ビットタイマを使っています。タイマティック周波数はタイムスタンプの設定から読み出す事ができます。この値は約 0.5 マイクロ秒の領域内にあります。タイマがオーバーフローすると、このイベントを示すためにパケットがストリームに埋め込まれます。ただし、タイマのオーバーフロー時にデータパケットが埋め込まれている場合、オーバーフロー パケットは埋め込まれません。 代わりに、データパケットのヘッダにオーバーフローが表示されます。

全てのタイムスタンプ付きパケットは、DGI デバイス内のモジュール割り込みから生成され、タイマのオーバーフ ロー割り込みによって割り込む事はできません。これは、タイマをサンプリングして埋め込む前にタイマがオー バーフローする可能性がある事を意味します。このようなイベントに対してタイムスタンプの同期と正確性を維持 するために、パケットはタイマのオーバーフロー ビットも埋め込んでいます。このビットはタイマの後にサンプリ ングされるため、サンプリングされたタイマの値が同期していた場合でも設定される可能性があります。

5.2.1.2 SPI インターフェイス

SPI 信号源は SPI インターフェイスで受信した生の値をストリーミングします。

^{© 2022} Microchip Technology Inc. and its subsidiaries



Important: SPI ハードウェア モジュールはアクティブ Low の CS(チップセレクト)信号を使います。 CS ピンが High の時に送信されたデータは全て無視されます。

SPI インターフェイスは[Data Sources](左)ペインの[**DGI]**セクションの下にあります。SPI 接続を選択すると、このペインの下部に SPI 設定が表示されます。

表 5-3. SPI 設定

フィールド名	值	用途
Char Length	5、6、7、8 ビットのいずれか	各転送のビット数
Mode	 クロックは通常 Low でアイドル、立ち上がり エッジでデータをサンプリング クロックは通常 Low でアイドル、立ち下がり エッジでデータをサンプリング クロックは通常 High でアイドル、立ち下がり エッジでデータをサンプリング クロックは通常 High でアイドル、立ち上がり エッジでデータをサンプリング 	SPI モードでクロック位相とサンプ リングを制御
Force CS Sync	チェックを入れて有効にします。	SPI インターフェイスはチップセレ クト ラインが 2 回切り換わった場合 にのみ有効になります。
Kit-side Timestamping	チェックを入れて有効にします。	ターゲットのタイムスタンプ

5.2.1.3 USART インターフェイス

USART 信号源は USART インターフェイスで受信した生の値をストリーミングします。

[Data Sources](左)ペインで USART 信号源を選択すると、下部に USART 設定が表示されます。

Note: 非同期シリアル プロトコル(DGI USART と CDC 仮想 COM ポート インターフェイスで使う UART プロトコ ル等)で使う **baud レート**は 5.1.「USB CDC 仮想 COM ポート」に記載されています。

表 5-4. USART 設定

フィールド名	値	用途
Baud Rate	0~2000000	非同期モードでの UART インターフェイスの baud レート
Char Length	5、6、7、8 ビットのいずれか	各転送のビット数
Parity	None、Even、Odd、Mark、Spaceの いずれか	通信に使うパリティのタイプ
Stop bits	1、1.5、2 ビットのいずれか	ストップビット数

5.2.1.4 I2C インターフェイス

I2C 信号源は I2C インターフェイスで受信した生の値をストリーミングします。

[I2C Configuration]オプションは左ペインの[DGI]セクションの I2C インターフェイスの下に表示されます。

I2C インターフェイスは[Data Sources](左)ペインの[DGI]セクションの下にあります。I2C 接続を選択すると、このペインの下部に I2C 設定が表示されます。

表 5-5. I2C 設定

フィールド名	値	用途
速度	0	インターフェイスの予測動作速度(Hz)は、クライアント デバイスがタ イミングを調整する場合に役立ちます。最大 400 kHz がサポートされ ます。

MPLAB® ICE 4インサーキット エミュレータ ユーザガイド

本エミュレータの特長

続き					
フィールド名	値	用途			
Address	1	クライアント デバイスのアドレス、7 ビットアドレス: 8~120			
Kit-side Timestamping	チェックを入れて 有効にします。	ターゲットのタイムスタンプ			

5.2.1.5 GPIO インターフェイス

GPIO インターフェイスは、有効にしたデバッグ GPIO ピンのビット値を格納します。ピンが切り換わるたびに符 号なし8ビットデータのパケットが送信されます。

[Data Sources](左)ペインで GPIO インターフェイスを選択すると、下部に GPIO 設定が表示されます。

表 5-6. 設定

フィールド名	値	用途
GPIO 0 Change Triggers Bus Read	ON、OFF	バス読み出しをトリガするために GPIO ピン 0 の変更を監視 します
GPIO 1 Change Triggers Bus Read	ON、OFF	バス読み出しをトリガするために GPIO ピン 1 の変更を監視 します
GPIO 2 Change Triggers Bus Read	ON、OFF	バス読み出しをトリガするために GPIO ピン2の変更を監視 します
GPIO 3 Change Triggers Bus Read	ON、OFF	バス読み出しをトリガするために GPIO ピン3の変更を監視 します



Important: デバッグ GPIO データ源をプロットする時に全ての GPIO がサンプリングされますが、変 更トリガを有効にした GPIO のみが変更時にサンプリングをトリガします。例えば、GPIO n (n = 0,1,2)全てが[GPIO n Change Triggers Bus Read]を無効にし、GPIO 3 がこの機能を有効にしている場 合、GPIO 3 を変更した時にのみ GPIO 値がサンプリングされます。つまり、4 つの GPIO 値が全て読 み出されるのは GPIO 3 を変更した時のみです。

5.2.1.6 Power インターフェイス

Power インターフェイスは接続された回路の消費電力を計測します。

デバッグツール DGI の下にある Power インターフェイスを選択します。[Power Settings]の下にあるコントロール を使ってこのインターフェイスを設定します。

表 5-7. [Power Settings]のコントロール

コントロール	値	用途
Enabled Channels	A、A+B	チャンネル A のみを有効にするか、チャンネル A およ び B を有効にするかを選択します。チャンネル A は常 に有効です。
Lock Channel A in high range*	チェックを外す/入れる	Low レンジへの自動切り換えを防ぐためにチャンネル A を High レンジにロックできます。これにより、レン ジ切り換え時に重要なサンプルを失わずに消費電流の 短いスパイクを検出できます。
Output Voltage in mV	1600 mV~5500 mV または 0	MPLAB ICE 4 はターゲット アプリケーションへの電力 供給に使える可変ターゲット電源を備えています。本 設定はこの電源を有効にしてその出力電圧を制御しま す。0 を選択すると電源は無効になります。
* 将来的な機能。		



Tip: [Power Settings]パネルで[**Apply]**をクリックするまで設定の変更は有効になりません。例えば、 電圧出力を有効にする場合、出力電圧値を設定して[**Apply]**をクリックする必要があります。それにより、電圧出力が実際に有効になり、それに応じて設定されます。



Tip: 既に Low レンジで実行されている場合、チャンネル A のレンジロックはデバッガを強制的に High レンジに戻す事はありません。強制的に変更するのに十分な大きさの電流になるまで待つか、単 純にデバッガを停止してから起動してください。



Tip: 各電力信号の時間プロットはシステムリソースを消費します。性能を向上させるには同時プロット回数を減らしてください

5.3 データキャプチャとランタイム ウォッチ - PIC MCU と dsPIC DSC

データキャプチャはデバイス内蔵のデバッグ機能です。キャプチャデータのアドレスレジスタの値と一致するアドレスを持つ SFR に値が書き込まれると、トリガとデータキャプチャが発生します。データキャプチャには直接アクセスできません。MPLAB Data Visualizer 等のデータキャプチャを使うアプリケーションを選択する必要があります。

ランタイム ウォッチはプログラムの実行に伴い、選択したシンボルが[Watches]ウィンドウ内で変化する事です。 [Watches]ウィンドウ内でランタイムになるシンボルは選択可能です。詳細は MPLAB X IDE ユーザガイドを参照し てください。

8/16 ビットデバイス

全ての 8/16 ビットデバイスがデータキャプチャとランタイム ウォッチ(またはどちらか)をサポートしている訳では ありません。デバイスごとにサポートされている機能のリストは[Help] > [Release Notes]の「Hardware Tool Debug Features by Device」に記載されています。

8 ビット PIC デバイスでウォッチできるのは 8 ビット変数のみ、16 ビット PIC デバイスでウォッチできるのは 16 ビット変数のみです。

8 ビットおよび 16 ビットデバイスの場合、データキャプチャ、ランタイム ウォッチ、ハードウェア ブレークポイン トは同じレジスタ/リソースを使います。例えば、あるシンボルのデータキャプチャ リソースを使う場合、そのシン ボルのハードウェア ブレークポイント リソースまたはランタイム ウォッチリソースは使えません。

32 ビットデバイス

全ての 32 ビットデバイスがデータキャプチャとランタイム ウォッチ(またはどちらか)をサポートしている訳ではあ りません。デバイスごとにサポートされている機能のリストは[Help] > [Release Notes]の「Hardware Tool Debug Features by Device」に記載されています。

PIC32 デバイスの場合、ハードウェア ブレークポイントはデータキャプチャ リソースもランタイム ウォッチリソー スも使いません。しかし、データキャプチャとランタイム ウォッチは同じリソースを使います。従って、あるシン ボルのデータキャプチャ リソースを使う場合、そのシンボルのランタイム ウォッチリソースは使えず、その逆も同 様です。

5.3.1 データキャプチャとストリーミング データ

データキャプチャは、デバイスからのストリーミング データを供給します。データキャプチャは以下の手順で設定 します。

- プロジェクトをビルドする([Projects]ウィンドウでプロジェクト名を右クリックして[Build]を選択)。使用可能 なシンボルを表示するには、プロジェクトをビルドする必要がある。[Project Properties] > [Loading]で、 [Load symbols when programming or building for production (slows process)]にチェックを入れる。
- 2. [Window] > [Debugging] > [Watches]を選択して、[Watches]ウィンドウを開く。
- 3. ウィンドウ内で右クリックして[New Watch]を選択する。[New Watch]ウィンドウで、ウォッチするシンボル または SFR を選択する。[**OK**]をクリックする
- 4. デバッグ セッションを開始する III 。[Watches]ウィンドウで値を確認する。

5.3.2 ランタイム ウォッチとストリーミング データ(PIC32M MCU)

データキャプチャはないものの命令トレースがある PIC32MZ/MK デバイスの場合、トレース メカニズムをランタ イム ウォッチに使えます。トレースコネクタ付きのデバイス PIM がある場合、3.3.8.「PIC32 命令トレース アダプ タボード」に記載されているアダプタボードを使って、ランタイム ウォッチを有効にするためのトレースケーブル を接続できます。デバイスのみの場合、ハイスピード 40 ピン リボンケーブルを使ってボードのコネクタからト レースピンにアクセスできます(3.3.1.「ターゲットとの接続のピン配置」参照)。

ランタイム ウォッチは以下の手順で設定します。

- プロジェクトをビルドする([Projects]ウィンドウでプロジェクト名を右クリックして[Build]を選択)。使用可能 なシンボルを表示するには、プロジェクトをビルドする必要がある。[Project Properties] > [Loading]で、 [Load symbols when programming or building for production (slows process)]にチェックを入れる。
- 2. [Window] > [Debugging] > [Watches]を選択して、[Watches]ウィンドウを開く。
- 3. ウィンドウ内で右クリックして[New Runtime Watch]を選択する。[New Runtime Watch]ウィンドウで、 ウォッチするシンボルまたは SFR を選択する。[OK]をクリックする
- 4. デバッグ セッションを開始する してい (Watches)ウィンドウで値を確認する。

5.3.3 ランタイム ウォッチと[Watches]ウィンドウ

ランタイム ウォッチはプログラムの停止(Halt)中ではなく実行中に、以下のウィンドウに変数の変化を表示します。

- ・ ウォッチ [Window] > [Debugging]メニュー
- ・ メモリ [Window] > [PIC Memory Views]

メニュー ランタイム ウォッチは以下の手順で設定します。

- プロジェクトをビルドする([Projects]ウィンドウでプロジェクト名を右クリックして[Build]を選択)。使用可能 なシンボルを表示するには、プロジェクトをビルドする必要がある。[Project Properties] > [Loading]で、 [Load symbols when programming or building for production (slows process)]にチェックを入れる。
- 2. [Window] > [Debugging] > [Watches]を選択して、[Watches]ウィンドウを開く。
- 3. ウィンドウ内で右クリックして[New Runtime Watch]を選択する。[New Runtime Watch]ウィンドウで、 ウォッチするシンボルまたは SFR を選択する。[**OK**]をクリックする
- 4. デバッグ セッションを開始する [Watches]ウィンドウで変数値の変化を観察する。
- 5. [Memory]ウィンドウで変数の変化を観察するには、一時停止(Halt)して、ウォッチ対象の変数を含む [Memory]ウィンドウを開く。プログラムの実行を再開し、このウィンドウで値の変化を観察する。

5.4 CI/CD のサポート

MPLAB ICE4 インサーキット エミュレータは、実装しているネットワーク通信機能により、CI/CD(継続的インテグ レーション/継続的デリバリ(またはデプロイ))用のハードウェア テストツールとして使えます。MPLAB X IDE で、 CI/CD ウィザードを使って Jenkins-Docker または Docker のみの統合用のファイルを作成します。CI/CD の詳細お よび CI/CD ウィザードの使い方は『CI/CD Wizard in MPLAB X IDE User's Guide』(DS-5000xxxx)または<u>オンライン</u> ヘルプの CI/CD Wizard in MPLAB X IDE User's Guide を参照してください。

ウィザードで Jenkins パイプライン ファイルを作成用に選択した場合、ウィザード画面のいずれかを使ってエミュレータを含める事ができます(下表参照)。

MPLAB ICE 4 をターゲットに接続するには ICE 4 ブレークアウト ボード(AC244141)も必要です。

表く	5-8.	CI/CD	ウィ	ザー	ド	- Hardware	Testing
----	------	-------	----	----	---	------------	---------

オプション	説明
Enable Hardware Testing	テストツールとしてエミュレータを有効にします。
Configuration to Build and Run on ICE4	エミュレータを使うプロジェクト設定([default]か[one dedicated to hardware use]か)を 選択します。

本エミュレータの特長

続き	続き				
オプション	説明				
IP Address of ICE4	使うエミュレータの IP アドレスを入力します。エミュレータの IP アドレスが分から ない場合、ipconfig 等のツールを使うと、エミュレータが接続されているシステ ムを検索してアドレスを確認できます。				
Enable MPLAB Code Coverage	MPLAB Code Coverage 機能を有効にします。この機能を使うには MPLAB Analysis Tool Suite ライセンスが必要です。 Note: カバレッジ レポートを有効にした場合、Jenkins サーバーで Code Coverage API プラグインを利用できる必要があります。				
Scan Output for Unity Test Results	設定が Unity Test Runner をビルドし、ビルドジョブが結果出力に基づいてレポート を作成する場合に有効にします。Unity テストの記述方法は『Unity - Getting Started』を参照してください。				

5.5 PIC MCU と dsPIC DSC の計測機能付きトレース

このセクションでは、使用可能な計測機能付きトレースのタイプと、PIC MCU(PIC マイクロコントローラ)と dsPIC DSC(dsPIC デジタルシグナル コントローラ)でそれらを使う方法を説明します。

5.5.1 トレースの要件

PIC 8 および 16 ビットデバイスでトレースを使うには以下が必要です。

- MPLAB X IDE
- ・ MPLAB XC8 C コンパイラまたは MPLAB XC16 C コンパイラ
- トレースをサポートするデバイスへのデバッグおよびトレース接続を備えたターゲットボード
- 3.3.5.「PIC MCU ICSP アダプタボード」に記載されているアダプタボードまたはハイスピード 40 ピン リボ ンケーブルへのコネクタを備えたターゲットボード

トレースの制約は以下の通りです。

- 8 ビットおよび 16 ビットデバイスでは、インライン アセンブリコード(C コード内のアセンブリコード)はトレースできません。
- デバイスごとのその他の制約は[Help] > [Help Contents] > [Hardware Tool Reference] > [Limitations]を参照して ください。
- トレースタイプごとのその他の制約は 5.5.3.「トレースの方法」を参照してください。

5.5.2 トレースの仕組み

MPLAB ICE 4 インサーキット エミュレータのトレース(計測機能付きトレース)は、基本的なトレース情報を提供す るソリューションです。TRACE()および LOG()マクロを使うと、アプリケーション実行中にプログラム位置または 変数値を MPLAB X IDE に送れます。それらの情報は、アプリケーションを停止した後で[Trace]ウィンドウで確認 できます。これらのマクロは手動で入力するか、エディタ内で右クリックしてコンテクスト メニューから選択でき ます。変数値を記録するには、コンテクスト メニューで選択する前にその変数を選択しておく必要があります。

図 5-1. LOG マクロを挿入する例



トレースの方法と関連オプションは、[Project Properties]ダイアログの[ICE4]カテゴリ、[Trace and Profiling]オプ ション カテゴリページに表示されます。[Communications Medium]に表示される選択肢は、ネイティブ トレース ([Native]) (PGC/PGD 通信ラインを使用)、SPI トレース([SPI])、I/O ポートトレース([I/O Port])です。全製品で全タ イプを使える訳ではありません。デバイスによってサポートしているトレースが異なります。計測機能付きト レース ライブラリは、PIC18F MCU デバイスではC言語とアセンブリ言語のプロジェクトをサポートしますが、 16 ビットデバイスではC言語のプロジェクトのみをサポートします。

図 5-2. [Project Properties]ダイアログ - トレースの選択

ategories:	Continue for ICE 4				
General File Inclusion/Exclusion	Option categories: Trace and Profilir	Options for I/LE 4 Option categories: Trace and Profiling			
- • Conf: [default] - • ICE 4	Data Collection Selection	User Instrumented Trace	~		
- O Loading	Data File Path and Name	defmplabxtrace.log			
	Data File Maximum Size (bytes)	10000000			
	Data Buffer Maximum Size (bytes)	(N/A)	01/A) 01/A) 01/A)		
- O XC16 (Global Options)	Time Stamp or Summary Profile Data	(N/A)			
— ○ xc16-as	Stall CPU When Trace Buffer is Full	(N/A)			
- 0 xc16-gcc	Disable Trace Macros				
- o xc16-ar	Communications Medium	Native	~		
 Analysis 	Timer Selection (Not Used by Application	on Code) (N/A)			
Conf: DebugConfiguration Simulator	Timer Interrupt Priority	(N/A)			
- O Loading		N. 1. V			
— Ubraries	Option Description				
	If you select an option its descr	ription will appear here.			
Manage Configurations					
Manage Network Tools					

送信されるトレース情報とログ情報は、どのトレースタイプを使っても同じです。TRACE()の場合、64~127 の 間の値を 1 つ送信します。この番号に基づいて生成されたラベルがコード内に自動的に挿入されるため、MPLAB X IDE は値の送信元になったトレースバッファ内の位置を識別できます。LOG()の場合、2 バイトのヘッダに続き、 記録された変数の値を送信します。ヘッダの 1 バイト目は変数のタイプを示す 0~63 の間の値です。2 バイト目 は変数の送信元の位置を示します。この位置は 0~127 の間の値で表します

TRACE () または LOG () の呼び出し中、割り込みは無効です。これは割り込みレベルのトレースまたはログ命令文が、アプリケーション レベルで既に実行中であるトレースまたはログ命令文と干渉しないようにするためです。

これは、高優先度の割り込み内の命令文によって低優先度の割り込み内の命令文が破損する事を防ぐ場合の状況に似ています。

5.5.3 トレースの方法

計測機能付きトレースの方法は 3 つあります。いずれも言語ツールのバージョンに依存し、MPLAB X IDE にリア ルタイムでデータをストリーム送信します。下表に各トレース方法の長所と短所、通信に関する情報をまとめます。

トレースの 方法	速度	コードサイズ への影響	リアルタイム 動作	ピンの使用	必要な デバイス機能	ハイスピード通信
ネイティブ トレース	高速	大	近い	なし	内蔵デバッグ	15 MIPS を超える ¹
SPIトレース	より 高速	中	より近い	SPIピン	SPI	あり ¹
l/O ポート トレース	最も 高速	小	最も近い	デバッグコネクタ の 9 ピン	なし	あり ²
1. 実際の速度はレイアウトやノイズ等の要素で変化します。						

2. デバイスポートからエミュレータ デバッグコネクタ(8 ピンポート+ 1 ピンクロック)への接続が必要です。

5.5.3.1 ネイティブ トレース

ネイティブ トレースは PGD/PGC ピンまたは EMUC/EMUD ピンを介して伝達する情報に使えます。この 2 線式イン ターフェイスはトレースマクロ形式を使います(5.5.5.「MPLAB X IDE でのトレースの設定」参照)。ネイティブ ト レースには 6 または 8 ピン ICSP 接続を使えます。

8/16 ビットデバイスのネイティブ トレースでは、クロック速度([Project Properties]ダイアログ、[ICE4]カテゴリ、 [Clock]オプション カテゴリ)を入力する必要があります。

ネイティブ トレースを使う場合、データキャプチャは使えません。これらの機能は同じデバイスリソースを使うためです。ブレークポイントは使えますが、ネイティブ トレース機能がブレークポイントを 1 つ使う事に注意してください。この状況は[Device Debug Resource]ツールバーには反映されません。

データキャプチャ トリガを使う場合、ネイティブ トレースを無効にする必要があります(5.5.8.「トレースの無効 化」参照)。

関連リンク

7.2.3.ネイティブ トレース中に手動でプログラムを停止すると最後のトレースレコードが消えてしまいました。どう したのでしょうか。

5.5.3.2 SPI トレース

SPI トレースは 8 ピン ICSP 接続でのみ使えます。トレースのクロックとデータはピン 7(DAT)とピン 8(CLK)を介して供給します。この機能を使う場合、デバイスを高速動作させるべきではありません。

これらのピンをトレースに使う場合、これらのピンに多重化された機能は使えません。

割り当て変更可能なペリフェラルピンを備えたデバイスの場合、SPI トレースマクロは PPS レジスタにアクセスせず、どの機能がどのピンに割り当てられているか知る必要はありません。マクロは MPLAB X IDE で選択された SPI1 または SPI2 に書き込みます。

SPIトレースではクロックスピード([Project Properties] > [ICE4] > [Clock]オプション カテゴリ)の入力は不要です。

SPI インターフェイスは、トレースマクロ フォーマットを使います(5.5.5.「MPLAB X IDE でのトレースの設定」参照)。

5.5.3.3 I/O ポートトレース

I/O ポートトレースは、標準通信でも高速通信でも使えます。トレースのクロックとデータは、デバイスの 8 ピン I/O ポートからエミュレータのロジックプローブ コネクタ経由で供給します。

I/O ポートトレースでは 8 ピン全てを使います。このポートは使用中の PGC および PGD ピンと多重化できません。 従って、デバイスのデータシートを確認して、未初期化/既定値のポートピンの状態を判断し、必要に応じて変更し ます。

ハードウェア接続については 3.3.7.3.「I/O ポートトレース接続」を参照してください。

ポート インターフェイスはトレースマクロ フォーマットを使います(5.5.5.「MPLAB X IDE でのトレースの設定」参照)。

5.5.4 ハードウェア

準備

- 1. PCと MPLAB ICE 4 の間に USB 接続を使います。他の通信タイプはトレースをサポートしていません。
- 2. [Native]、[SPI]、[I/O PORT]のいずれかのトレースをサポートしているデバイスを見つけます(<u>[Help] ></u> [<u>Release Notes] > [Debug Features Support] > [Hardware Tool Debug Features by Device]</u>を参照)。
- 3.3.5「PIC MCU ICSP アダプタボード」に記載されているアダプタボードを使う場合、エミュレータと ターゲットの通信用コネクタを備えたターゲットボードを設計します。または、デバッグおよびトレース用 のピンを備えたハイスピード 40 ピン リボンケーブルに接続できるターゲットボードを設計します(3.3.1. 「ターゲットとの接続のピン配置」参照)。
 Note: I/O ポートトレースでは、通信とポートピンへのアクセスにハイスピード 40 ピン リボンケーブルを使 う必要があります。
- 4. SPI トレースでは 7 (DAT)および 8 (CLK)ピンを使います。従って、これらのピンに多重化された他の機能は 使えません。I/O ポートトレースでは、トレースに使うポートは使用中の PGC および PGD ピンと多重化で きません。

ハードウェアの準備

お客様の基板で PIC 計測トレース機能を使う場合、以下の手順を実行します。

- 1. ターゲットボードに電源が供給されていない事を確認する
- 2. エミュレータまたはアダプタボードとターゲットボード上の通信コネクタを通信ケーブルで接続する
- 3. ターゲットに電源を供給する

5.5.5 MPLAB X IDE でのトレースの設定

MPLAB ICE 4 インサーキット エミュレータのトレース用に MPLAB X IDE を設定するには、以下の手順を実行します。

- 1. プロジェクト名を右クリックして[Properties]を選択する。[Project Properties]ダイアログ内の[Categories]の 下で[ICE4]をクリックする
- 2. [Option Categories]で[Clock]を選択するデータキャプチャとトレースでは、エミュレータが命令サイクルス ピードを把握している必要があります。
- 3. [Option categories]で[Trace and Profiling]を選択する
- 4. [Data Collection Selection]で[User Instrumented Trace]を選択する
- 5. [Communications Medium]で[Native]、[I/O PORT]、[SPI]トレースのいずれかを選択する
- 6. トレース関連のオプションを全て設定する(9.2.6.「トレースとプロファイリング」参照)。
- 7. **[OK]**をクリックする

5.5.6 トレースの実行

- 1. アプリケーションをデバッグ実行する([Debug] > [Debug Project])
- 2. アプリケーションを一時停止する
- [Trace]ウィンドウでトレースデータを確認する([Window] > [Debugging] > [Trace])。各 TRACE マクロについて、マクロに続くコード行を通過するごとに、そのコード行が[Trace]ウィンドウに表示されます。各_LOGマクロについては、マクロに続くコード行を通過するごとに、そのコード行内で選択した変数が[Trace]ウィンドウに表示されます。
 Note: 複数のコード行または変数をトレースする場合、トレースする各行/変数の前にマクロを配置する必要があります。

以上の手順を、トレースポイントを変更するごとに実行します。

5.5.7 トレースのヒント

コード内で___TRACE および__LOG マクロを使う場合、以下の点を検討してください。

 アプリケーションの1つの領域に注目して、[Trace]ウィンドウ内に「フロー」が形成されるように__TRACE および__LOGマクロを配置します。これで実行フローをたどれるようになり、存在しないトレースポイントま たは不正なトレースポイント、またはトレースフローの突然の中断を手がかりに、アプリケーションをデバッ グできます。 __TRACE および __LOG マクロをコード内で条件文と併用するとデバッグに役立ちます。例: 変数がある値に達した時に記録(ログ)を開始する場合

```
If(var > 5)
{
  LOG(ID, var)
}
```

可能ならば、将来のデバッグに備えて___TRACE および__LOG マクロをコード内に残しておきます([Project Properties] > [ICE4] > [Trace]で[Disable Trace Macros]を選択)。

5.5.8 トレースの無効化

トレースのデータ収集を一時的に OFF にするには、以下の手順を実行します。

- 1. [File] > [Project Properties] > [ICE4] > [Trace and Profiling]を選択する
- 2. [Disable Trace Macros]にチェックを入れる
- 3. [OK]をクリックする

トレース機能を完全に無効にするには、以下の手順を実行します。

- 1. コードからトレースとログのマクロを全て削除する。
- 2. [File] > [Project Properties] > [ICE4] > [Trace and Profiling]を選択する
- 3. [Data Collection Selection]で[Off]を選択する
- 4. **[OK]**をクリックする

5.5.9 リソース使用量の例

以下は説明のための例です。実際の結果はコンパイラ/アセンブラのバージョン、コマンドライン オプション、 MPLAB X IDE のバージョン、記録するデータ変数のサイズ、割り込みの状態、使用するデバイス等で異なります。 全ての例に引数の設定、関数呼び出し、戻り時間(サイクル数)が含まれています。

PIC18FXxxJ MCU の例は、割り込み優先度設定なしでコンパイル/アセンブルしています(ライブラリサイズに 30 命令追加)。優先割り込みを使う場合は 57 命令、割り込みを使わない場合は 15 命令追加されます。

dsPIC33F DSC の例は、memcpy()の16 ビットのライブラリサイズで指定される9命令を示しています。

表 5-9. PIC18FxxJ MCU の 4 MHz (1 MIPS)動作(アセンブリ言語のプロジェクト)

	ネイティブ	SPI	I/O ポート
ライブラリサイズ(命令数)	23 + 30	37 + 30	25 + 30
使用する GPR(バイト単位)	8	6	6
TRACE (id) の命令サイクル	80	54	42
LOG(id, BYTE)の命令サイクル	168	90	57

表 5-10. PIC18FxxJ MCU の 40 MHz (10 MIPS)動作(C 言語のプロジェクト)

	ネイティブ	SPI	I/O ポート
ライブラリサイズ(命令数)	75 + 30	87 + 30	112 + 30
使用する GPR(バイト単位)	10	8	8
TRACE (id) の命令サイクル	79	71	55
LOG(id, INT) の命令サイクル	225	169	162

表 5-11. dsPIC33F DSC の 10 MIPS 動作(C 言語のプロジェクト)

	ネイティブ	SPI	I/O ポート
ライブラリサイズ(命令数)	87 + 9	92 + 9	93 + 9

続き			
	ネイティブ	SPI	I/O ポート
使用する GPR(バイト単位)	18	14	0
TRACE (id) の命令サイクル	80	53	32
LOG(id, INT)の命令サイクル	212	124	106

dsPIC33Fの 16 MIPS 動作(C 言語のプロジェクト)

表 5-12. dsPIC33F DSC の 16 MIPS 動作(C 言語のプロジェクト)

	ネイティブ	SPI	I/O ポート
TRACE (id) の命令サイクル	88	53	32
LOG(id, INT)の命令サイクル	227	138	106

表 5-13. dsPIC33F DSC の 34 MIPS 動作(C 言語のプロジェクト)

	ネイティブ	SPI	I/O ポート
TRACE(id)の命令サイクル	100	53	32
LOG(id, INT) の命令サイクル	251	152	106

5.5.10 トレース/ログ ID 番号の詳細

MPLAB X IDE はトレースまたはログのマクロに自動的に ID 番号を割り当てます。IDE 内部での番号付けの方法は 以下の通りです。

トレースポイントは 64 ヶ所、ログポイントも 64 ヶ所設定できます。これらの上限はポートトレース(8 ビット)に よって決まります。ビット 7 はクロックとして使い、フラグとして使うビット 6 でトレースレコード(1)かログレ コード(0)かを示します。

トレースレコードの場合、下位ビットはトレース番号を表します(nnnnnn)。正規のトレース番号は 0~63 であり、 これにトレースフラグをセットする必要があると言えますが、フラグを番号の一部として、有効な番号は 64~127 であると言った方が分かりやすいでしょう。

クロック	1	n	n	n	n	n	n
ビット7	ビット6						ビット0

ログレコードの場合、下位ビットはログ番号を表します(nnnnn)。

クロック	0	n	n	n	n	n	n
ビット7	ビット6						ビット0

5.5.11 トレースに関するクイック リファレンス

MPLAB ICE 4 インサーキット エミュレータのトレース機能を初めて使う場合、完全に理解するためにトレースに 関するセクション全体を読む事を推奨します。本セクションは、トレースに関するクイック リファレンスとして 使ってください。

- 1. [File] > [Project Properties] > [ICE4] > [Trace]を選択する。トレースを有効化し、トレース方法を選択してオ プションを設定する
- 2. [Window] > [Debugging] > [Trace]を選択して、トレースデータを表示する[Trace]ウィンドウを開く
- 3. コードを右クリックして、トレースマクロ(__TRACE、__LOG)を挿入する
- 4. プロジェクトをデバッグ実行する

5.6 PIC32M MCU の命令トレース

このセクションでは、PIC32M デバイスのトレースとその使い方を説明します。命令トレースは全ての PIC32M デ バイスで使える訳ではありません。[<u>Help</u>] > [<u>Release Notes</u>] > [<u>Debug Features Support</u>] > [<u>Hardware Tool Debug</u> <u>Features by Device</u>]を参照してください。

5.6.1 命令トレースの要件

PIC32M 32 ビットデバイスでトレースを使うには、以下が必要です。

- MPLAB X IDE および MPLAB XC32/32++ C コンパイラ
- トレースをサポートするデバイスとトレースポートを備えた PIC32 PIM(プラグイン モジュール)またはトレー スをサポートするデバイスへのデバッグおよびトレース接続を備えたターゲットボード
- MPLAB ICE 4 PIC32 トレース アダプタボードと 12 インチ トレースケーブルまたはハイスピード 40 ピン リ ボンケーブルへのコネクタを備えたターゲットボード

5.6.2 命令トレースの動作

PIC32 命令トレースは MIPS32 iFlowtrace[™]メカニズムを使います。これは非侵入型のハードウェア命令トレースで す。このトレースを使うと、デバイスが実行する全ての命令をキャプチャできます。トレースデータは TRCLK と TRD3:0 ピンを介してデバイスからエミュレータに送信されます。エミュレータはこのデータを、リングバッファ のように動作する PC 上のトレースバッファにストリーム送信します。

トレースデータの量の上限は、トレースバッファのサイズで決まります。このバッファは最大に設定しても、すぐ ー杯になります。従って、何をキャプチャすべきか慎重に検討する事が必要です。

[<u>Project Properties] > [ICE4] > [Trace and Profiling]</u>でトレースを有効化してオプションを設定します。以下の設定が 可能です。

- [Data selection] トレースの有効/無効化とトレースタイプの選択
- ・ [Data file path and name] トレースファイルの保存場所
- ・ [Data file maximum size] トレースファイルのサイズ
- ・ [Data Buffer maximum size] トレースバッファのサイズ

外部トレースバッファの最大サイズは 22 MB です。この外部バッファが一杯になっても実行は停止しません。

9.2.6.「トレースとプロファイリング」を参照してください。

図 5-3. 命令トレースのオプション

Project Properties - PIC32MX360F512LFLP		
Categories: General File Inclusion/Exclusion Conf: [default] Cef4 Conf: default] Cef4 Conf: default] Cef4 Cef4 Conf: default] Cef4 Conf: default] Cef4 Conf: default] Cef4 Cef4 Cef4 Conf: default] Cef4 Cef4 Conf: default] Cef4	Options for ICE 4 Option categories: Irrace and Profiling Data Collection Selection Data File Path and Name Data File Maximum Size (bytes) Data Buffer Maximum Size (bytes) Time Stamp or Summary Profile Data Stall CPU When Trace Buffer is Full Option Description	Reset Instruction Trace/Profiling defmplabxtrace.log 1000000 546000 (N/A)
Manage Configurations	OK Cancel	Apply Unlock Help

5.6.3 ハードウェア

準備

- 1. PC と MPLAB ICE 4 の間に USB 接続を使います。他の通信タイプはトレースをサポートしていません。
- 2. PIC32M 命令トレースをサポートするデバイスを見つけます(<u>[Help] > [Release Notes] > [Debug Features</u> <u>Support] > [Hardware Tool Debug Features by Device]</u>を参照)。
- 3.3.8.「PIC32 命令トレース アダプタボード」に記載されているアダプタボードを使う場合、エミュレータ とターゲットの通信用コネクタとトレースピン用コネクタを備えたターゲットボードを設計します。または、 デバッグおよびトレース用のピンを備えたハイスピード ケーブルに接続できるターゲットボードを設計しま す(3.3.1.「ターゲットとの接続のピン配置」参照)。
- トレースでは TRCLK および TRD3:0 ピンを使います。従って、これらのピンに多重化された他の機能は使えません。
 PIC32MX360F512L の場合、多重化された機能は RG14:12 と RA7:6 です。詳細はデバイスのデータシートで確認してください。

ハードウェアの準備 - PIM 上の PIC32M MCU

PIM で PIC32M 命令トレースを使う場合、以下の手順を実行します。

- 1. 電源が供給されていない状態のターゲットボードに PIM を挿入する
- 2. エミュレータまたはアダプタボードとターゲットボード上の通信コネクタを通信ケーブルで接続する
- 3. アダプタボードを使う場合、アダプタボードと PIM 上のトレースコネクタをトレースケーブルで接続する
- 4. ターゲットに電源を供給する

MPLAB® ICE 4インサーキット エミュレータ ユーザガイド

本エミュレータの特長

ハードウェアの準備 - ターゲット上の PIC32M MCU

お客様の基板で PIC32M 命令トレースを使う場合、以下の手順を実行します。

- 1. ターゲットボードに電源が供給されていない事を確認する
- 2. エミュレータまたはアダプタボードとターゲットボード上の通信コネクタを通信ケーブルで接続する
- アダプタボードを使う場合、アダプタボードとターゲットボード上のトレースコネクタをトレースケーブル で接続する
- 4. ターゲットに電源を供給する

5.6.4 MPLAB X IDE での命令トレースの設定

MPLAB ICE 4 インサーキット エミュレータのトレース用に MPLAB X IDE を設定するには、以下の手順を実行します。

- 1. プロジェクト名の上で右クリックして[Properties]を選択する。[Project Properties]ダイアログ内の [Categories]の下で[ICE4]をクリックする
- 2. [Option categories]で[Trace and Profiling]を選択する
- 3. [Data Collection Selection]で[Instruction Trace/Profiling]を選択する
- 4. トレース関連のオプションを全て設定する
- 5. [OK]をクリックする

デバッグ実行におけるトレース機能は、トレースバッファが一杯になってもロールオーバーして、プログラムが停止するまでデータを供給し続けます。

5.6.5 命令トレースデータの表示

トレースを有効にしてコードを実行すると、エミュレータがトレースデータを収集します。デバイスが停止すると、 トレースデータがデコードされ[Trace]ウィンドウに表示されます(<u>[Window] > [Debugging] > [Trace]</u>)。

5.7 SAM ITM/SWO トレース

ITM (Instrumentation Trace Macrocell)トレースは SWO (Serial Wire Output)を使って UART フォーマットのデータ を出力します。ITM トレースは全てのデバイスで使える訳ではありません。[Help] > [Release Notes] > [Debug Features Support] > [Hardware Tool Debug Features by Device]を参照してください。

5.7.1 ITM トレースの要件

SAM デバイスでトレースを使うには、以下が必要です。

- MPLAB X IDE および MPLAB XC32/32++ C コンパイラ
- トレースをサポートするデバイスへのデバッグおよびトレース接続を備えたターゲットボード
- 3.3.6.「SAM MCU Cortex-M トレース アダプタボード」に記載されているアダプタボードまたはハイスピー ド 40 ピン リボンケーブルへのコネクタを備えたターゲットボード

5.7.2 ITM/SWO トレースの動作

CoreSight ITM (Instrumentation Trace Macrocell)ブロックはソフトウェア アプリケーションで駆動するトレース ソースです。サポートするコードは SWIT (SoftWare Instrumentation Trace)を生成します。また、このブロックは 大まかなタイムスタンプ機能も提供します。以下に、このブロックの主な用途を示します。

- printf スタイルのデバッグをサポート
- OS およびアプリケーション イベントをトレース
- システム診断情報を送出

5.7.2.1 SAM ITM トレース

MPLAB ICE 4 は UART フォーマットの ITM トレースをホスト コンピュータにストリーミングできます。トレース は 10 ピンヘッダの TRACE/SWO ピン(JTAG TDO ピン)でキャプチャされます。データは MPLAB ICE 4 の内部で バッファリングされ、トレース インターフェイスを介してホスト コンピュータに送信されます。この時の信頼でき る最大データレートは約 4 MB/s です。 **Note:** MPLAB X IDE で ITM baud レートを設定できます。5.7.4.「MPLAB X IDE での ITM の設定」を参照してください。

図 5-4. ITM のブロック図



ITM は以下のサブブロックを含みます。

Timestamp タイムスタンプ パケットを生成

Sync control ITM 同期機能

Arbiter 同期、タイムスタンプ、SWIT パケットの間を調停

FIFO ATB First-In First-Out(先入先出)

Emitter ATB で登録されたエミッタ

データは APB インターフェイスを使ってスティミュラス レジスタに書き込まれます。このデータは SWIT パケット として ATB インターフェイスで送信されます。

5.7.2.2 SAM SWO(シリアルワイヤ出力)

CoreSight SWO は、オンチップ トレースデータと TPA(トレースポート アナライザ)によってキャプチャされる データストリームをつなぐブリッジとして機能するトレースデータ ドレインです。TPIU のようなデバイスとして 完全な TPIU 機能の一部をサポートし、ゲート数を最小限に抑えてシンプルなデバッグ ソリューションを実現しま す。

TPIU と比較すると、SWO には以下の特長があります。

- ・ フォーマッタがない
- ・ パターン ジェネレータがない
- 8ビットATB入力
- ・ 同期トレース出力がない(つまり、TRACEDATA、TRACECTL、TRACECLK ピンがない)
- フラッシュをサポートしていない(不要なため)
- トリガをサポートしていない
- 外部入出力がない(EXTCTLIN と EXTCTLOUT が実装されていない)

図 5-5. SWO のブロック図



5.7.3 ハードウェア

準備

- 1. PCと MPLAB ICE 4 の間に USB 接続を使います。他の通信タイプはトレースをサポートしていません。
- 2. ITM トレースをサポートするデバイスを見つけます([Help] > [Release Notes] > [Debug Features Support] > [Hardware Tool Debug Features by Device]を参照)。
- 3.3.3.「SAM MCU JTAG/SWD アダプタボード」に記載されているアダプタボードを使う場合、エミュレータとターゲットの通信用コネクタとトレースピン用コネクタを備えたターゲットボードを設計します。 または、デバッグおよびトレース用のピンを備えたハイスピード ケーブルに接続できるターゲットボードを設計します(3.3.1.「ターゲットとの接続のピン配置」参照)。
- 4. トレースを使う場合、TRACESWO ピンを使います。従って、そのピンに多重化された他の機能は使えません。

ハードウェアの準備

ITM/SWO 機能の使用手順は以下の通りです。

- 1. ターゲットボードに電源が供給されていない事を確認する
- 2. エミュレータまたはアダプタボードとターゲットボード上の通信コネクタを通信ケーブルで接続する
- アダプタボードを使う場合、アダプタボードとターゲットボード上のトレースコネクタをトレースケーブル で接続する
- 4. ターゲットに電源を供給する

5.7.4 MPLAB X IDE での ITM の設定

MPLAB ICE 4 インサーキット エミュレータの ITM 用に MPLAB X IDE を設定するには、以下の手順を実行します。

- 1. [Projects]ウィンドウでプロジェクト名を右クリックして[Properties]を選択する
- 2. [Project Properties]ダイアログ内の[Categories]の下で[ICE4]をクリックする
- 3. 以下のセクションの説明に従ってクロックとトレースを設定する

5.7.4.1 クロックの設定

[Option categories]で[Clock]を選択します。ターゲットのランタイム周波数を入力します。

MPLAB® ICE 4インサーキット エミュレータ ユーザガイド

本エミュレータの特長

Note: このオプションはクロックを設定するのではなく、ランタイム ウォッチ、データキャプチャ、トレースに使う値 をエミュレータに通知します。

Project Properties - SAME70_swo_2			×
Categories: General File Inclusion/Exclusion	Options for ICE 4 Option categories: Clock	~	Reset
Conf: [Default] ○ ICE 4 ○ Loading ○ Libraries ○ Building ○ Arm GCC (Global Options) ○ arm-as ○ arm-as-pre ○ arm-gcc ○ arm-g++ ○ arm-d ○ arm-ar	Target run-time dock fequency (in MHz)	300	
	Option Description		
Manage Configurations			
Manage Network Tools			
		OK Cancel Apply	Unlock Help

5.7.4.2 ITM トレースの設定

[Option categories]で[Trace and Profiling]を選択する

- 1. [Data Collection Selection]で[ITM Trace]を選択する
- 2. [ITM baud rate]を選択して SWO 速度を指定する。クロックとこの値を使って SWO プリスケーラ値を決定 する

Note: 本エミュレータが使える SWO baud レートは 512 KHz、1 MHz、2 MHz、4 MHz に限られます。

Note: クロック切り換えがアプリケーションで必要な場合、SWO 機能が求められる想定クロックレートに 合わせて SWO baud レートを設定する事を推奨します。

- 3. 一部の SAM デバイスでは、ITM の追加設定用に.ini ファイルを追加する必要があります。5.7.6.「ソフト ウェアの実装」を参照してください。
- 4. 必要に応じてロギング設定を変更する
- 5. 完了したら、[**OK**]をクリックする

	Options for ICE 4		
 General 			
File Indusion/Exclusion Conft [Default]	Option categories: Trace and Profiling	V	Reset
- • ICE 4	Data Collection Selection	ITM Trace	· · · · ·
- Loading	ITM baud rate	.5 MHz	
- O Libraries	ETM Setup File Path and Name	useroperationsITMSAME70.ini	
Arm GCC (Global Options)	Data File Path and Name	defmplabxtrace.log	
— 🍳 arm-as	Data File Maximum Size (bytes)	10000000	PER
- o arm-as-pre	Data Buffer Maximum Size (bytes)	546000	
 → arm-g++ 	Time Stamp or Summary Profile Data	(N/A)	
- o arm-ld	Stall CPU When Trace Buffer is Full	(04/A)	
	Option Description		
Manage Configurations			

5.7.5 ITM データの表示

デバッグ実行におけるトレース機能は、トレースバッファが一杯になってもロールオーバーして、プログラムが 停止するまでデータを供給し続けます。トレースデータの使用方法または表示方法はアプリケーションによって 決まります。

図 5-6. ITM 表示の出力例

and the second se	istory 📆 🔯			DIAL RIGIO		Wa		
2606	STATIC IN	INF mint 32	T TTM Set	ndChar (uint32 t c		ini mi ().		
2607		arne arnoor		denar (arnoor)			
2608	IF LUTTM	TCR & TTM	TCR TTME	NA Mek) I= OIII.) 44	/*	TTM enab	1ed */	
2609	((TTM	STER & HUL	TON TIME) != 0111.)	1 /*	TTM Port	žū enabled	*1
2610	1	FILM & LOD		/	, ,		to cumpted	8 - C
2611	while (TM-SPORT FOR	1 132 ==	0111.)				
2612	i i	in stori (oc	1.402	001)				
E)	NOP	0.						
2614	1	() /						
2615	TTM->PO	= 8" (IIO) TS	(mint8 t	hch:				
2616	1	x1[00].40 -	(arneo_c)	, cu,				
2617	Terurn (c							
2618	1							
2619	1							
0015								
28.20								
2620	/**							
2620	/**							
2620 2621	/**		(
2620 2621 Ariables	Call Stack	Breakpoints	Output	TTM Display - port 0 ×				
Ze20 2621 Zariables Outp LOHEI LOHEI LLOHE HELLO OHELI LOHEI ELLOF	Call Stack ut Format: ASCI LLOHELLOHELLOH HELLOHELLOHELLOH ELLOHELLOH	Breakpoints	Output t number DHELLOHELL LLOHELLOHELL LOHELLOHELLOHE LLOHELLOHE	TTM Display - port 0 × OHELLOHELLOHELLOHELLOHELLOHELLOHELLOHEL	OHELLOHELL OHELLOHELLOHELLOHELLOHELLOHEL	DESCRIPTION OHELLOHELL LICHELLOHELL OHELLOHELL UHELLOHELLOHE LICHELLOHELLOHE LICHELLOHELLOHELLOHELLOHELLOHELLOHELLOHELL	CHELLOHELLOHEL CHELLOHELLOHEL HELLOHELLOHELLOHE ELLOHELLOH	LOHELLO LOHELLO DHELLOHEL DHELLOHEL LOHELLO DHELLOHEL DHELLOHEL DHELLO
Ariables Outpo LOHEI LOHEI LLOHE HELLO OHELI LOHEI ELLOH OHELI LOHEI LLOHEI LLOHEI LLOHEI	Call Stack ut Format: ASCI DEDORCEDURES LLOHELLOHELLOH HELLOHELLOHELLOH ELLOHELLOH	Breakpoints	Output t number DHELLOHELL LLOHELLOHEL LLOHELLOHEL LOHELLOHEL	TTM Display - port 0 × OHELLOHELLOHELLOHELLOHELLOHELLOHELLOHEL	OHELLOHELL OHELLOHELLOHELLOHELLOHELLOHEL	CORPTONES	CONCLOSE CONCLOSE CHELLOHELLOHEL CHELLOHELLOHELLOHE ELLOHELLOHELLOHE CHELLOHELLOHELLOHE HELLOHELLOHELLOHE HELLOHELLO	LOHELLO LOHELLO DHELLOHEL JOHELLO LOHELLO DHELDO DHE DHELDO DHELDO DHELDO DHE DHE DHE DHE DHE DHE DHE DHE DHE DHE

9.3.2.ITM ウィンドウと関連ダイアログ

5.7.6 ソフトウェアの実装

SAM E70 および SAM E54 デバイスは標準的な Arm デバイスとは異なるクロック機構(PLL)を実装しています。 従って、特定のデバイス設定が必要です。これには.ini ファイルを使います。このファイルは MDK-ARM リセット 時に読み込まれます。

useroperationsITMSAME70.ini

```
; Trace Clock Setup
; WDWORD (0x400E064C, 0x4); // Select Master clock for ITM/ETM
write,0x400E064C, 4
;PMC->PMC_SCER = PMC_SCER_PCK3; // Enable PCK3
write,0x400E0600,0x800
useroperationsITMSAME54.ini
```

; Trace Clock Setup ; Enable ITM/ETM Peripheral Generic clock and set it to Master Clock write,0x40001D3C, 0x40

; Configure PB30 to SWO - GPIO PORT MUX write,0x410080BF,0x07
; Configure Pullups for PB30 write,0x410080DE,0x41

5.8 PC サンプリング - 16 ビット PIC MCU のみ

PC サンプリングとは、C コードを検証して関数の処理にかかる時間の割合を調べるための方法です。この情報に よってプログラム実行のどこに時間がかかるのかが明らかになるため、コードを最適化できます。

PC サンプリングではデバイスタイマを設定してプログラム実行のサンプルを取得し、結果を[PC Profiling]ウィンドウに表示します。

PC プロファイリングと PC サンプリングは似ています。詳細は 5.9.「PC プロファイリング - 32 ビット MCU のみ」を参照してください。

図 5-7. PC サンプリング - 16 ビットデバイス

; PC	PC Profiling - dsPIC33EP_PCSamp 🛛 🕸 🕴 Variables				
	Total Samples = 7999				
<mark>0</mark> ु जिल्ल	PC Sampling				
	Function	Sample Count	100.00 %	%	
	subrA	2244	28.05		
	subE_nocalls	2248	28.10		
	subrB	1623	20.29		
	main	1223	15.29		
	subrC	397	4.96		
	subrD	264	3.30		

5.8.1 要件

現状では、PC サンプリングを使う場合、以下に対応したプロジェクトを設定する必要があります

- サポートしているデバイス:
 - PIC24F、PIC24EP
 - dsPIC33FJ、dsPIC33E
- ・ MPLAB XC16 C コンパイラ (バージョン 1.10 以上)

5.8.2 クロックの設定

クロックは以下の手順で設定します。

- 1. [Project Properties]ウィンドウを開く([File] > [Project Properties])
- <u>[ICE4] > [Clock]</u>に移動する(この値は、ターゲットの実際の速度(コードのコンフィグレーション ビットで設定)と一致させます)
- 3. **[Apply]**をクリックする

5.8.3 サンプリングの設定

サンプリングは以下の手順で設定します。

- 1. [ICE4] > [Trace and Profiling]を選択する。[Data Collection Selection]で[PC Sampling]を選択する
- このウィンドウでデータファイルとタイマを設定する(9.2.6.「トレースとプロファイリング」参照)
 Note: 選択しているタイマがユーザプログラム内で使われていない事を確認します。
- 3. **[OK]**をクリックする

図 5-8. PC サンプリング - 選択と設定

ategories:	Options for ICE 4		
General General File Inclusion/Exclusion	Option categories: Trace and Profiling	~	Reset
- • ICE 4	Data Collection Selection	PC Sampling	~ /
- • Loading	Data File Path and Name	defmplabxtrace.log	
- • Libraries	Data File Maximum Size (bytes)	1000000	
Building XC16	Data Buffer Maximum Size (bytes)	(N/A)	
 XC16 (Global Options) 	Time Stamp or Summary Profile Data	(N/A)	
- • xc16-as	Stall CPU When Trace Buffer is Full	(N/A)	
- • xc16-gcc	Disable Trace Macros	(N/A)	
- • xc16-ar	Communications Medium	Native	~
Analysis	Timer Selection (Not Used by Application Code)	1	~
Conf: DebugConfiguration	Timer Internant Priority	4	~
O Simulator O Loading O Libraries	Option Description		
O Building O XC16 O XC16 (Global Options) O xc16-as	If you select an option its description will	appear here.	
Manage Configurations			
Manage Network Tools			

5.8.4 操作

データは以下の手順で生成します。

- 1. [Window] > [Debugging] > [PC Profiling]を選択する([PC Profiling]ウィンドウが表示されます)
- 2. コードを実行してから停止させる
- 3. ウィンドウ内のサンプリング データを確認する(データは停止中にのみ表示されます)
- ウィンドウ内を右クリックしてコンテクストメニューを表示させ、データをクリアするか再度読み込むかを 選択する

このデータは Code Profiling プラグインでも表示できます。このプラグインは Microchip Gallery (gallery.microchip.com/)で購入できます。

5.9 PC プロファイリング - 32 ビット MCU のみ

PC プロファイリングとは、C コードを検証して関数の処理にかかる時間の割合を調べるための方法です。この情報によってプログラム実行のどこに時間がかかるのかが明らかになるため、コードを最適化手法の参考とできます。

PC プロファイリングでは、全てのプログラム カウンタ(PC)トレースサンプルがトレース (データ) バッファから取 得され、プロファイリングが実行されます。このデータは[PC Profiling]ウィンドウに表示されます。この例では、 トレースバッファは 769,230 個の PC サンプル データポイントを収めています。このうち 99,014 個(12.87%)は main()関数、202,839 個(26.37%)は subrA()関数に対応しています。

PC プロファイリングと PC サンプリングは似ています。詳細は 5.8.「PC サンプリング - 16 ビット PIC MCU のみ」を参照してください。

本エミュレータの特長

図 5-9. PC プロファイリング - 32 ビットデバイス

: PC	PC Profiling - PIC32MX360F512LFLP 🛛 🕸 🗄 Variables			
	Total Samples = 769230			
<mark>0</mark> 54	PC Sampling			
-	Function	Sample Count	100.00 %	%
	subrA	202839	26.37	
	subrB	202836	26.37	
	subE_nocalls	202835	26.37	
	main	99014	12.87	
	subrD	61705	8.02	
	I			

5.9.1 要件

現状では PC プロファイリングを使う場合、サポートされている以下のデバイスに対応したプロジェクトを設定する 必要があります。

データキャプチャ付き PIC32MX - [Help] > [Release Notes] > [Debug Features Support] > [Hardware Tool Debug Features by Device]を参照してください。
 Note: 他の PIC32 デバイスはサポートしていません。

5.9.2 プロファイリングの設定

プロファイリングは以下の手順で設定します。

- 1. PIC32 命令トレース用にハードウェアを設定する(3.3.8.「PIC32 命令トレース アダプタボード」参照)
- 2. [Project Properties]ウィンドウを開く([File] > [Project Properties])
- 3. [ICE4] > [Trace and Profiling]を選択する。[Data Collection Selection]で[Instruction Trace/Profiling]を選択する
- 4. このウィンドウでデータファイルを設定する(9.2.6.「トレースとプロファイリング」参照)[OK]をクリックする

図 5-10. PC プロファイリング - 選択と設定

Project Properties - ICE4_PIC32MX_PCProfilir	ng	×
Categories: General File Indusion/Exclusion	Options for ICE 4 Option categories: Trace and Profiling	Reset
 Conf. [Gefault] ICE 4 Loading Libraries Building XC32 (Global Options) XC32-as XC32-gcc XC32-gc XC32-d XC32-ar Analysis 	Data Collection Selection Data File Path and Name Data File Maximum Size (bytes) Data Buffer Maximum Size (bytes) Time Stamp or Summary Profile Data Stall CPU When Trace Buffer is Full Option Description If you select an option its description will appear her	Instruction Trace/Profiling defmplabxtrace.log 1000000 546000 (N/A) Ø e.
Manage Configurations		
Manage Network Tools		
	C	K Cancel Apply Unlock Help

5.9.3 操作

データは以下の手順で生成します。

- 1. [Window] > [Debugging] > [PC Profiling]を選択する([PC Profiling]ウィンドウが表示されます)
- 2. コードを実行してから一時停止/停止する
- 3. ウィンドウ内のプロファイリング データを確認する(データは停止中にのみ表示されます)
- ウィンドウ内を右クリックしてコンテクスト メニューを表示させ、データをクリアするか再度読み込むかを 選択する

このデータは Code Profiling プラグインでも表示できます。このプラグインは Microchip Gallery (gallery.microchip.com/)で購入できます。

5.10 デバッガ ポーリング

ターゲット デバイスとのアクティブなデバッグ セッション中に、PC(プログラム カウンタ)をできるだけ速く繰り返してポーリングするように MPLAB Data Visualizer から MPLAB ICE 4 に指示できます。この操作では高い割合の PC サンプリングまたはコードトレースを得られませんが、コードと電力の相関関係で想定よりも多くの電力を消費するコードの領域を見つけ出す際に役立ちます。例として下図を参照してください。

MPLAB Data Visualizer の詳細は MPLAB Data Visualizer のウェブページを参照してください。



5.10.1 要件

現状では、デバッグ ポーリングを使う場合、サポートされている以下のデバイスに対応したプロジェクトを設定す る必要があります。

- AVR-8 ビットデバイス(UPDI インターフェイス)
- SAM-32 ビットデバイス(SWD インターフェイス)

5.10.2 操作

デバッガ ポーリングは、SAM 32 ビットデバイスの SWD インターフェイスと AVR 8 ビットデバイスの UPDI イン ターフェイスを使って内部プログラム カウンタの場所にアクセスします。タイムスタンプ付きプログラムカウンタ アドレスのサンプルからデバイスのプログラム実行状況の洞察を得ることができます。

Note: デバッガ ポーリングは、MPLAB X IDE 内から MPLAB Data Visualizer 🎬 を実行した場合にのみ有効です。 これにより、MPLAB Data Visualizer は MPLAB X IDE バックエンドを介してデバイス上のデバッグシステムにアク セスできます。

Note: デバッガ ポーリングではデバッガが実行されている事が求められます。つまり MPLAB X IDE で[Debug Project]を選択する必要があります。

5.11 **Power Monitor**

MPLAB ICE 4 インサーキット エミュレータは電力監視に使えます。電力監視とは電流値等の電力データをキャプ チャする事です。電力監視機能は、独立した 2 つの電流検出チャンネルを使って回路設計の消費電力を計測、最適 化します。

電力監視を利用できる Microchip 社製デバイスは PIC、dsPIC、AVR、SAM MCU です。

本エミュレータは MPLAB Data Visualizer と連携して電力データのプロットを提供します。MPLAB X IDE v5.50 以降、MPLAB DV プラグインは MPLAB X IDE に含まれています。スタンドアロン版を利用する事もできます。

Note: MPLAB Data Visualizer には USB 通信のみを使えます。

このソフトウェアの詳細は MPLAB Data Visualizer の製品ページを参照してください。

5.11.1 電力監視の使い方

MPLAB ICE 4 ユニットの 40 ピンコネクタの横に 6 ピン電流検出コネクタがあります。以下も参照してください。 10.5.4.電流検出モジュール

図 5-11. 電流検出接続 - 前から見た図



以下に、各種の電力計測の接続例を示します。全ての設定をサポートしていないデバイスもあります。

図 5-12. MCU 電力の計測



図 5-13. MCU とボードの電力を個別に計測する場合



5.11.2 詳細なユースケース

電力監視のユースケースは MPLAB Data Visualizer の文書に記載されています。使うハードウェアは Atmel Power Debugger ですが、設定は MPLAB ICE 4 インサーキット エミュレータで必要なものと似ています。

以下に、MPLAB ICE 4 を使った単純なユースケースを示します。

5.11.2.1 低消費電力アプリケーション

MPLAB ICE 4 を MPLAB X IDE と MPLAB Data Visualizer と共に使って、一般的な低消費電力アプリケーションの 消費電力を計測、解析、把握、最適化するために本エミュレータをどのように使えるかを学びます。DGI (Data Gateway Interface)を使ったコード計測の例もあります。

5.11.2.1.1 要件

この例に従って作業するには、以下が必要です。

- MPLAB X IDE v6.00 以上をインストールしたコンピュータ。最新の MPLAB Data Visualizer プラグインを IDE から利用可能。
- Microchip 社の MPLAB ICE 4 キット (ケーブルが付属)
- Microchip 社の PIC32MX470 Curiosity ボードまたは類似のターゲットボード
- ・ ピンヘッダ: 1 個の 6 ピン 100-mil ICSP ヘッダ
- 基本的なはんだ付け装置を利用可能

5.11.2.1.2 ハードウェア

始める前にハードウェアに多少の変更を加える必要があります。下図に、PIC32MX470 Curiosity ボードで変更が必要な箇所を示します。



PIC32MX470 Curiosity ボードで MPLAB ICE 4 を使うには、ICSP ヘッダエリア(J16)にピンヘッダーを実装する必要があります。

()	
كما	l
	٢

To do: Curiosity ボードに 6 ピン ICSP ヘッダを取り付けます。

PIC32MX470 Curiosity ボードはプログラマ/デバッガ(PK4oB)を実装済みです。そのため、MPLAB ICE 4 をプログラ マ/デバッガとして使う場合はオンボード デバッガを無効にする必要があります。

To do: ジャンパ J2 を外してオンボード デバッガを取り除きます。

MPLAB ICE 4 を使って PIC32MX470 Curiosity ボードに給電します。これは後でソフトウェアで設定します。どち らの USB 接続もボードへの給電には使わないため、電源入力ジャンパ(J8)は外部の+5 V 電源を選択する必要があり ます。



To do: J8 ジャンパをポジション 5-3(P/S)に移動します。



5.11.2.1.3 接続

ハードウェアの変更が完了したら、MPLAB ICE 4 ターゲット ケーブル アセンブラを PIC32MX470 Curiosity ボードに 実装した ICSP ヘッダに接続する事で MPLAB ICE 4 をターゲットに接続します。



To do: ハイスピード ケーブルの一端をエミュレータ ユニットの 40 ピンコネクタに接続します。ICE ケーブルの他端を MPLAB ICE 4 ICSP アダプタボードに接続します。アダプタボード上の 8 ピン SIL コネクタをターゲットボードの ICSP ヘッダに接続します。下図のように、ヘッダピン 1 が SIL コネクタの最初のソケットに入るように配置します。



ターゲットの消費電流を計測するには、外部電源からの電流が本エミュレータの電流検出 A ピンからターゲットボード上の5V外部電源ヘッダへと流れる必要があります。



To do: 電源を電流検出コネクタ CS+ A と GND に接続します。その後で CS- A を 5 V ヘッダの+ピンに 接続します。-ピンの接続を電流検出 GND ピンに戻します。

設定は以下のようになります。



関連リンク 10.5.4.電流検出モジュール

5.11.2.1.4 プロジェクトの作成

MPLAB X IDE で**[New Project]**ボタン 🛄 をクリックして新しいスタンドアロン プロジェクトを設定します。

- 1. デバイス: PIC32MX470F512H
- 2. ツール: MPLAB ICE 4
- 3. コンパイラ: MPLAB XC32 v2.50

新しいプロジェクトで、プロジェクト ウィンドウの[Source Files]を右クリックして[<u>new] > [main.c]</u>を選択します。 その次に以下のコードをエディタで入力して保存します。

変化の見える電流をターゲットボード上で生成するには、ターゲットがアクティブに何かを実行している必要が あります。以下のターゲット PIC32MX470F512H 用コードはボード上の LED を点滅させます。

// PIC32MX470F512H Configuration Bit Settings // 'C' source line config statements // DEVCFG3 #pragma config USERID = 0xFFFF // Enter Hexadecimal value (Enter Hexadecimal value) #pragma config FSRSSEL = PRIORITY 7 // Shadow Register Set Priority Select (SRS Priority 7) #pragma config PMDL1WAY = // Peripheral Module Disable Configuration (Allow ON only one reconfiguration) #pragma // Peripheral Pin Select Configuration (Allow only config IOL1WAY = ON one reconfiguration) #pragma // USB USID Selection (Controlled by the USB Module) config FUSBIDIO = ON // USB VBUS ON Selection (Controlled by USB Module) #pragma config FVBUSONIO = ON // DEVCFG2 // PLL Input Divider (12x Divider) #pragma config FPLLIDIV = DIV 12 // PLL Multiplier (24x Multiplier) #pragma config FPLLMUL = MUL $\overline{2}4$ // USB PLL Input Divider (12x Divider) #pragma config UPLLIDIV = DIV 12 // USB PLL Enable (Disabled and Bypassed) #pragma config UPLLEN = OFF // System PLL Output Clock Divider (PLL Divide by 256) #pragma config FPLLODIV = DIV_256

// DEVCFG1

```
#pragma config FNOSC = FRCDIV
                                         // Oscillator Selection Bits (Fast RC Osc w/Div-bv-N
(FRCDIV))
#pragma config FSOSCEN = ON
                                         // Secondary Oscillator Enable (Enabled)
#pragma config IESO = OFF
                                         // Internal/External Switch Over (Disabled)
                                         // Primary Oscillator Configuration (Primary osc
#pragma config POSCMOD = OFF
disabled)
#pragma config OSCIOFNC = OFF
                                         // CLKO Output Signal Active on the OSCO Pin
(Disabled)
#pragma config FPBDIV = DIV 8
                                         // Peripheral Clock Divisor (Pb Clk is Sys Clk/8)
#pragma config FCKSM = CSDCMD
                                         // Clock Switching and Monitor Selection (Clock
Switch Disable, FSCM Disabled)
#pragma config WDTPS = PS1048576
#pragma config WINDIS = OFF
                                         // Watchdog Timer Postscaler (1:1048576)
                                         // Watchdog Timer Window Enable (Watchdog Timer is in
Non-Window Mode)
                                         // Watchdog Timer Enable (WDT Disabled (SWDTEN Bit
#pragma config FWDTEN = OFF
Controls))
                                         // Watchdog Timer Window Size (Window Size is 25%)
#pragma config FWDTWINSZ = WINSZ 25
// DEVCFG0
#pragma config DEBUG = OFF
                                         // Background Debugger Enable (Debugger is Disabled)
#pragma config JTAGEN = OFF
                                         // JTAG Enable (JTAG Disabled)
#pragma config ICESEL = ICS PGx2
                                         // ICE/ICD Comm Channel Select (Communicate on PGEC2/
PGED2)
#pragma config PWP = OFF
                                         // Program Flash Write Protect (Disable)
#pragma config BWP = OFF
                                         // Boot Flash Write Protect bit (Protection Disabled)
#pragma config CP = OFF
                                         // Code Protect (Protection Disabled)
// #pragma config statements should precede project file includes.
// Use project enums instead of #define for ON and OFF.
#include <xc.h>
// Three LEDs: Red, Green, Yellow
#define LEDS ON 0xD0
#define LEDS OFF 0x00
void delay (void)
    int n = 50000;
    while(n>0) {n--;}
int main(void) {
    // Port E access
    TRISE = 0x0000; // set all port bits to be output
    while(1) {
        // delay value change
        delav();
        LATE = LEDS ON; // write to port latch
        // delay value change
        delav();
        LATE = LEDS OFF; // write to port latch
    }
    return -1:
}
```

5.11.2.1.5 ソフトウェアの設定

[Project Properties](プロジェクト名を右クリックして[Properties]を選択)で、エミュレータがターゲットに給電していない事を確認します。

Project Properties - pic//f.lcd.uch curiority	nwm_rah_led		~
Project Properties - pic24f-lcd-usb-curiosity-	pwm-rgb-ied		X
Categories: General File Inclusion/Exclusion	Options for ICE 4		Reset
Conf: [default]	Bower target circuit from ICE 4		
ICE 4	Power target circuit from ICE 4		
	Voltage Level	3.25	~
 → Building → XC16 → XC16 (Global Options) → xc16-as → xc16-gcc → xc16-ld → xc16-ar → Analysis 	Option Description		
Manage Configurations Manage Network Tools	If you select an option its description will	appear here.	
Planage retwork roots			
		OK Cancel Apply	Unlock Help

[Make and Program Device]ボタン をクリックすると、コードがビルドされてターゲット デバイスに書き込まれます。

トラブルシュート:

- プロジェクトのビルドに失敗した場合、コードを完全にコピーして貼り付けた事を確認します。また、出力 ウィンドウに表示されるエラーメッセージを参照します。
- MPLAB X IDE とエミュレータまたはターゲットとの接続に問題がある場合は接続を確認します。

5.11.2.1.6 MPLAB Data Visualizer の表示

MPLAB Data Visualizer は MPLAB X IDE 内から開く事もスタンドアロン アプリケーションとして開く事もできます。

MPLAB X IDE で開く場合は[Window] > [Debugging] > [Data Visualizer]を選択します。Data Visualizer が開くと、 電流検出が使われている状態で MPLAB ICE 4 の[DGI]リストの下に[Power]選択項目があります。これをクリック すると、[Power Settings]コントロールが表示されます。このユースケースでは、エミュレータからの電力供給は使 われていません([Output Voltage in mV]が 0)。

図 5-14. MPLAB ICE 4 の[DGI]のオプション



全ての電源をプロットするには、ドロップダウン矢印をクリックして以下のアクションを選択します。プロットデータのストリーミングが始まります。



トラブルシュートの最初のステップ

6. トラブルシュートの最初のステップ

MPLAB ICE 4 インサーキット エミュレータの動作に問題が生じた場合、以下のセクションを参照してください。 MPLAB ICE 4 ユニットのライトの意味は 10.3.「インジケータ ランプ(LED)」を参照してください。

6.1 最初に確認する項目

- 1. 使っているデバイスを確認します。比較的最近リリースされたデバイスを使う場合、MPLAB X IDE を新し いパック(DFP/TP)バージョンに更新する事が必要な場合があります。
- 2. **Microchip 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。**通信接続に関する 抵抗とコンデンサのガイドラインに適合していますか。3.3.「ターゲットとの接続」を参照してください。
- 3. **ターゲットに給電していますか。**本エミュレータは、9 V の外部電源で給電している時にのみターゲットに 給電できます。詳細は 10.2 電源仕様を参照してください。
- 4. USB ハブを使っていますか。ハブは電源に供給されていますか。問題が解決しない場合、ハブを使わずに エミュレータを PC に直接接続してください。
- 5. **本エミュレータに付属の通信ケーブルを使っていますか。**これよりも長いケーブルを使っている場合、通信 エラーが生じる可能性があります。長いケーブルが必要な場合は別の通信タイプを検討してください。3.2. 「PCの接続」を参照してください。
- 6. **デバッガに付属する USB ケーブルを使っていますか。**その他の USB ケーブルの場合、品質が劣る、長過ぎる、USB 通信をサポートしていない等の可能性があります。

6.2 デバッグに失敗する主な理由

- 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。 外部オシレータを使っている場合、内部オシレータを試してみます。内部 PLL を使っている場合、PLL の設 定が正しい事を確認してください。
- 2. **ターゲットボードに給電されていない:** エミュレータから給電している場合、ターゲットまたはエミュレー タへの電源ケーブルの接続を確認してください。
- 3. **VDD 電圧が不適切である:** VDD 電圧がデバイス仕様外です。詳細はデバイスのプログラミング仕様を参照 してください。
- 4. **物理的に接続されていない:** デバッガとコンピュータ間またはデバッガとターゲットボード間が物理的に接続されていません。通信ケーブルを確認してください。
- 5. **通信が遮断された:** 何らかの理由で PC とエミュレータの通信が中断しました。MPLAB IDE でエミュレータ を再接続してください。
- デバイスが取り付けられていない: デバイスがターゲットボードに正しく挿入されていません。エミュレー タが正しく接続されておりターゲットボードに電源が供給されていても、デバイスが存在しないか正常に挿 入されていない場合、以下のメッセージが表示されます。 Target Device ID (0x0) does not match expected Device ID (0x%x) (ターゲット デバイスの ID (0x0)がデバイ ス ID (0x%x)と一致しません) %x はデバイス ID です。
- 7. **デバイスがコード保護されている:** 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。
- デバイスがデバッグ回路を持っていない: 量産デバイスはデバッグ機能を実装していない可能性があります。
 必要に応じてプロセッサ拡張パック(DS50001292)またはエミュレータ拡張パック(DS50002243)を使ってください。
- アプリケーション コードが破損している: ターゲット アプリケーションに破損またはエラーがあります。
 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、ターゲットのパワーオン リセットを実行します。
- 10. **プログラミング ピンの設定が不適切である:** PGC/PGD ピンペアがコンフィグレーション ビットで正しく プログラミングされていません(複数の PGC/PGD ピンペアを備えたデバイスの場合)。
- 11. **追加設定が必要である:**他の設定がデバッグと競合しています。ターゲットのコード実行を妨げるようなコンフィグレーションが設定されている場合、エミュレータによるデバッグモードでのコード実行もできません。

MPLAB[®] ICE 4インサーキット エミュレータ ユーザガイド トラブルシュートの最初のステップ

- 12. ブラウンアウト電圧が不適切である: ブラウンアウト検出電圧が動作電圧 VVDD よりも高く設定されていま す。この場合、デバイスはリセット状態になるためデバッグできません。
- 13. 接続が不適切である: 適切な通信接続は 3.3.「ターゲットとの接続」のガイドラインを参照してください。
- 14. **無効な要求である:** エミュレータは、常に要求に応えられる訳ではない。例えばターゲット アプリケーションが実行中の場合、エミュレータはブレークポイントを設定できません。

6.3 一般的注意事項

- プログラミングに関する一般的な問題の可能性もあります。アイコンを使って実行モードに切り換え、なる べくシンプルなアプリケーション(例: LED 点滅プログラム)でターゲットへの書き込みテストを行います。 このアプリケーションが動作しない場合、ターゲットの設定に何らかの問題があると考えられます。
- ターゲット デバイスが何らかの損傷(例:過電流)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少なくありません。ターゲットボードを交換して再試行します。
- 3. エミュレータの動作を確認し、アプリケーションを正しくセットアップしてください。詳細は 4. 「動作」を 参照してください。
- 回路に対してプログラム速度設定が速過ぎる可能性があります。MPLAB X IDE で[File] > [Project Properties] を選択して[Categories]で[ICE4]を選択し、[Program Options]の[Program Speed]ドロップダウン メニューか ら適切な速度を選択してください。既定値は「Normal」です。

図 6-1. [Program Speed]のオプション

Categories:	Options for ICE4		
General General File Inclusion/Exclusion Conf: [default]	Option categories: Program Options	•	Reset
- • ICE4	Erase All Before Program		
Loading	LED Brightness Setting	5	•
Cubranes Building	PGC Configuration	pull down	•
E • XC16 (Global Options)	PGC resistor value (kohms)	4.7	
- • xc16-as	PGD Configuration	pull down	•
e xc16-gcc	PGD resistor value (kohms)	4.7	
• xc16-ar	Program Speed	Normal	•

5. デバッガが正常に動作しておらず、ファームウェアのダウンロードまたはデバッガの再プログラミングが必要な場合があります。以下のセクションを参照して適切な操作を判断してください。

6.4 ハードウェア ツール エマージェンシー ブート ファームウェア リカバリ ユーティリティ の使い方

<u> へ caution</u> このユーティリティは、ハードウェア ツール ブート ファームウェアを工場出荷時状態に戻す場合に使 います。ハードウェア ツールがどのマシンでも動作しない場合にのみ使います。

まれにデバッガをリカバリブート モード(再プログラム)に移行させる事が必要な場合があります。例えば、デバッガ がコンピュータに接続されている時に以下の状況が発生する場合です。

- ・ デバッガの LED が点灯しない
- LED がシアン色である
- 「ブートロードモードの呼び出し方」の手順が成功しなかった

Important: エマージェンシー リカバリ ユーティリティを MPLAB ICE 4 で使うには MPLAB X IDE v6.00 以上が必要です。

メインメニュー オプション[Debug] > [Hardware Tool Emergency Boot Firmware Recovery]を選択し、MPLAB X IDE に記載されている手順を慎重に実行します。

トラブルシュートの最初のステップ

図 6-2. エマージェンシー ユーティリティの選択



手順が成功した場合、リカバリ ウィザードが成功した事を示す画面を表示します。MPLAB ICE 4 は動作準備が整 い、MPLAB X IDE と通信できる状態です。手順が失敗した場合、再試行します。それでも失敗する場合、 Microchip 社サポート(support.microchip.com)にお問い合わせください。

よく寄せられる質問(FAQ)

7. よく寄せられる質問(FAQ)

以下のセクションでは、MPLAB ICE 4 インサーキット エミュレーション システムについてよく寄せられる質問と その回答を記載してします。

7.1 動作に関する FAQ

MPLAB ICE 4 インサーキット エミュレータ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。 MPLAB ICE 4 は ICSP および他のターゲット 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。本エ ミュレータは専用メモリに書き込んだデバッグ実行プログラムを使います。レガシー8 ビット PIC デバイスの場合、 デバッグ実行プログラムはプログラムメモリに書き込まれます。

デバッグ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は低下しません。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。

MPLAB X IDE インターフェイスと MPLAB ICE 4 インサーキット エミュレータの組み合わせは、どのようにして 従来のデバッグツールよりも多くの機能を実現しているのですか。

MPLAB ICE 4 は、メモリの専用領域に書き込まれたデバッグ実行プログラムを使って通信します。このデバッグ実 行プログラムによって通信効率が向上します。本デバッガは FPGA、大容量 SRAM バッファ(1Mx8)、ハイスピード USB インターフェイスを備えています。プログラムメモリ イメージをダウンロードして SRAM に格納するため、 高速なプログラミングが可能です。デバッガ内の FPGA は、ターゲット デバイスのインサーキット デバッガ モ ジュールとの連携を高速化するアクセラレータとして機能します。

従来のデバッガでは、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送出する必要がありました が、MPLAB ICE 4 インサーキット エミュレータでも同じですか。例えば、フラグが High になったら停止する、 という動きはできますか。

従来のデバッガでは、監視用に専用のデバッガチップ(-ME)を使います。MPLAB ICE 4 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。MPLAB ICE 4 では、外部ブレークポイントを使うのではなく、デバッグ エンジンが内蔵するブレークポイント回路を使います。つまりバスアクセスやブレーク条件はデバイスの内部論理 で監視されています。

MPLAB ICE 4 インサーキット エミュレータでは複合ブレークポイントを使えますか。

はい。特定のデータメモリ アドレスに格納された値に基づいてブレークする事ができます。複数のイベントが発生 した後にブレークするシーケンス化ブレークポイントも使えます。しかし、シーケンスは 2 つしか使えません。 AND 条件と PASS カウントも使えます。

標準ケーブルには何か制約がありますか。

標準 ICSP-RJ11 ケーブルの最大クロック周波数は約 15 MHz です。デバッグ中のデバイス検証動作は CPU クロックレートに関係なく 15 MHz 未満で行われます。しかし、一部の先進機能は CPU バスサイクルに同期して実行されます(例:計測機能付きトレース、データ収集)。データ収集中に CPU が定格(例:40 MIPS)で動作すると、ケーブル上のクロックレートは 15 MHz を超えます。この場合、トレースおよびデータキャプチャ動作の信頼性が確保できず、標準 ICSP-RJ11 ケーブルは使えません。

MPLAB ICE 4 インサーキット エミュレータによってコードの実行速度は低下しますか。 MPLAB ICE 4 は命令サイクルを消費しません。データの出力はデバイス内のステートマシンが実行します。

dsPIC DSC デバイス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。

MPLAB ICE 4 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。

よく寄せられる質問(FAQ)

7.2 トラブルに関する FAQ

MPLAB ICE 4 が想定通りに動作しない、または全く動作しない場合、以下のセクションを参照してください。「エ ラーメッセージ」も参照してください。

7.2.1 コンピュータが省電力/休止モードに移行した後、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。

本エミュレータを(特にデバッガとして)長時間使う場合、お使いのコンピュータ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。[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。これにより、全ての USB サブシステム コンポーネントで全ての通信を維持できます。

7.2.2 デバイスのプログラミング後にベリファイで失敗します。これはプログラミングの問題ですか。

[Run Main Project] を選択した場合、デバイスはプログラミング直後に自動的にコード実行を開始します。 従って、コード実行によってフラッシュメモリが書き換えられる場合、ベリファイが失敗する事があります。プログ

ラミング後のコード実行を防ぐには、[Hold in Reset] 🛀 を選択します。

7.2.3 ネイティブ トレース中に手動でプログラムを停止すると最後のトレースレコードが消えてしまいました。 どうしたのでしょうか。

手動での停止は非同期のため、最後のデータが欠落する可能性があります。実行と停止を再度試してください。また は、ブレークポイントを使ってコードを停止してください。

7.2.4 16 ビットデバイスの周辺モジュールに[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、本エミュレータは周辺モジュール制御レジスタの予約済みビット(通常は bit 14 または 5)を Freeze ビットとして使います。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)を使います。

7.2.5 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。どのようにすれば原因を特定できま すか。

以下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特定する。
- 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、以下のような 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);
    }
;
```

アサートを使う例:

```
ASSERT (IPL==7)
```

8. エラーメッセージ

MPLAB ICE 4 インサーキット エミュレータは各種のエラーメッセージを生成します。エラーメッセージには、特別な対処が必要なもの、情報を提供するだけのもの、一般的な対処法で解決できるものがあります。エラーメッ セージの下に説明がある場合はそれらをお読みください。説明を読んでも問題を解決できない場合、または説明が ない場合、以下を参照してください。

8.1 エラーメッセージのタイプ

以下のセクションでは、選択したエラーメッセージを論理的なカテゴリにグループ化して解決策を提案します。最 後のセクションでは全てのエラーメッセージを記載します。

8.1.1 インストール ファイルの破損/期限切れエラー

Failed to download firmware(ファームウェアのダウンロードに失敗しました)

HEX ファイルが存在する場合の対処法:

- 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可能性があります。MPLAB X IDE を再インストールして ください。

HEX ファイルが存在しない場合の対処法:

• MPLAB X IDE を再インストールしてください。

Unable to download debug executive(デバッグ エグゼクティブをダウンロードできません)

デバッグ時にこのエラーが表示された場合の対処法:

- 1. MPLAB ICE 4 のデバッグツールとしての選択を解除する。
- 2. プロジェクトを閉じてから MPLAB X IDE を終了する。
- 3. MPLAB X IDE を再起動して再度プロジェクトを開く。
- 4. デバッグツールとして MPLAB ICE 4 を選択し、再度ターゲット デバイスのデバッグを試みる。

Unable to download program executive(書き込み実行プログラムをダウンロードできません)

プログラミング時にこのエラーが表示された場合の対処法:

- 1. MPLAB ICE 4 のプログラマとしての選択を解除する。
- 2. プロジェクトを閉じてから MPLAB X IDE を終了する。
- 3. MPLAB X IDE を再起動して再度プロジェクトを開く。
- 4. プログラマとして MPLAB ICE 4 を選択し、再度ターゲット デバイスの書き込みを試みる。

これらの対処法で問題が解決しない場合、「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。

8.1.2 デバッグ障害エラー

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 デバイスはデバッグの準備が完了していません。コンフィグレーション ビットの設定を確認し、デバイスをプログラミングしてからデバッグしてください。

デバイスをプログラミングする前に実行(Run)を試みると、このメッセージが表示されます。実行(Run)を試みると このメッセージが表示される場合、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る場合の対 処法は以下の通りです。

デバイスはコード保護されています。

デバイスのコード保護が有効になっている場合、コードの読み書きと変更(読み出し、書き込み、ブランクチェック、 ベリファイ)はできません。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ます(<u>[Windows] ></u> [Target Memory Views] > [Configuration Bits])。

MPLAB[®] ICE 4インサーキット エミュレータ ユーザガイド ^{エラーメッセージ}

コード保護を無効にするには、該当するコンフィグレーション ビット (デバイスのデータシート参照)をコード内ま たは[Configuration Bits]ウィンドウ内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 グラミングします。

これらの対処法で問題が解決しない場合、「デバッガとターゲット間の通信エラーの対処方法」と「デバッグ障害の対処方法」を参照してください。

8.1.3 その他のエラー

ICE 4 is busy.Please wait for the current operation to finish (ICE 4 がビジーです。実行中の動作が終了するまで 待機してください)

- 待機する。本デバッガが実行中のアプリケーション タスクを終了した後に、本デバッガの選択解除を再度試みる。
- 2. 遅択解除を再度試みる。
- 3. 本デバッガとコンピュータ間の接続ケーブルを抜いた後に、本デバッガの選択解除を再度試みる。
- 4. MPLAB X IDE を終了する。

8.1.4 エラーメッセージの一覧

表 8-1. エラーメッセージ (アルファベット順)

AP_VER=Algorithm Plugin Version (アルゴリズム プラグイン バージョン)

AREAS_TO_PROGRAM=The following memory area(s) will be programmed: (以下のメモリ領域をプログラムします:)

AREAS_TO_READ=The following memory area(s) will be read: (以下のメモリ領域を読み出します:)

AREAS_TO_VERIFY=The following memory area(s) will be verified: (以下のメモリ領域をベリファイします:)

BLANK_CHECK_COMPLETE=Blank check complete, device is blank. (ブランクチェック完了、デバイスはブラン クです)

BLANK_CHECKING=Blank Checking... (ブランクチェック実行中...)

BOOT_CONFIG_MEMORY=boot config memory. (ブートコンフィグ メモリ)

BOOT_VER=Boot Version. (ブートバージョン)

BOOTFLASH=boot flash. (ブートフラッシュ)

BP_CANT_B_DELETED_WHEN_RUNNING=software breakpoints cannot be removed while the target is running. The selected breakpoint will be removed the next time the target halts. (ターゲット動作中にソフトウェア ブレークポイントを削除する事はできません。選択したブレークポイントは、次のターゲット停止時に削除されます)

CANT_CREATE_CONTROLLER=Unable to find the tool controller class. (ツール コントローラ クラスが見つかりません)

CANT_FIND_FILE=Unable to locate file %s. (ファイル%s が見つかりません)

CANT_OP_BELOW_LVPTHRESH=The voltage level selected %f, is below the minimum erase voltage of %f.The operation cannot continue at this voltage level. (選択した電圧レベル%f は最小消去電圧%f を下回っています。この 電圧レベルでは動作を継続できません)

CANT_PRESERVE_PGM_MEM=Unable to preserve program memory: Invalid range Start = %08x, End = %08x. (プログラムメモリを保存できません。無効なレンジ: 開始値 = %08x、終了値 = %08x)

CANT_READ_REGISTERS=Unable to read target register(s). (ターゲット レジスタを読み出せません)

CANT_READ_SERIALNUM=Unable to read the device serial number. (デバイスシリアル番号を読み出せません)

CANT_REMOVE_SWPS_BUSY=The ICE 4 is currently busy and cannot remove software breakpoints at this time. (ICD 4 は現在ビジーです。ソフトウェア ブレークポイントは削除できません)

エラーメッセージ

CHECK_4_HIGH_VOLTAGE_VPP=CAUTION: Check that the device selected in MPLAB IDE (%s) is the same one that is physically attached to the debug tool.Selecting a 5V device when a 3.3V device is connected can result in damage to the device when the debugger checks the device ID.Do you wish to continue? (注意: MPLAB IDE (%s)で 選択したデバイスとデバッグツールに物理的に接続したデバイスが同じである事を確認してください。3.3 V デバ イスを接続した状態で 5 V デバイスを選択すると、デバッガがデバイス ID をチェックする際にデバイスが損傷する 可能性があります)

CHECK_PGM_SPEED=You have set the program speed to %s.The circuit on your board may require you to slow the speed down.Please change the setting in the tool properties to low and try the operation again. (プログラム速度 は%s に設定されています。現状の回路では速度を下げる必要があるようです。ツールプロパティで設定を Low に 変更し、再試行してください)

COMM_PROTOCOL_ERROR=A communication error with the debug tool has occurred. The tool will be reset and should re-enumerate shortly. (デバッグツールとの通信エラーが発生しました。ツールはリセットされ、すぐにエニュメレートし直す必要があります)

COMMAND_TIME_OUT=ICE 4 has timeout out waiting for a response to command %02x. (ICD 4 は、コマンド%02x への応答を待機中にタイムアウトしました)

CONFIGURATION=configuration. (コンフィグレーション)

CONFIGURATION_MEMORY=configuration memory. (コンフィグレーション メモリ)

CONNECTION_FAILED=Connection Failed. (接続に失敗しました)

CORRUPTED_STREAMING_DATA=Invalid streaming data has been detected.Run time watch or trace data may no longer be valid.It is recommended that you restart your debug session. (無効なストリーミング データが検出さ れました。ランタイム ウォッチまたはトレースデータは無効になった可能性があります。デバッグ セッションを リスタートする事を推奨します)

CPM_TO_TARGET_FAILED=An exception occurred during ControlPointMediator.ToTarget(). (ControlPointMediator.ToTarget()実行中に例外が発生しました)

DATA_FLASH_MEMORY=Data Flash memory. (データ フラッシュメモリ)

DATA_FLASH=data flash. (データフラッシュ)

DEBUG_INFO_PGM_FAILED=Could not enter debug mode because programming the debug information failed.Invalid combinations of config bits may cause this problem. (デバッグ情報のプログラミングに失敗したためデ バッグモードに移行できません。コンフィグレーション ビットの無効な組み合わせが原因である可能性がありま す)

DEBUG_READ_INFO=Reading the device while in debug mode may take a long time due to the target oscillator speed.Reducing the range that you'd like to read (under the ICE 4 project properties) can mitigate the situation.The abort operation can be used to terminate the read operation if necessary. (ターゲットのオシレータ速度により、 デバッグモード中のデバイスの読み出しに時間がかかる可能性があります。(ICD 4 のプロジェクト プロパティで) 読み出すレンジを狭めると時間を短縮できます。必要であれば中止して読み出し動作を終了できます)

DEVICE_ID_REVISION=Device Id Revision. (デバイス ID リビジョン)

DEVICE_ID=Device Id. (デバイス ID)

DEVID_MISMATCH=Target Device ID (0x%x) is an Invalid Device ID.Please check your connections to the Target Device. (ターゲット デバイス ID (0x%x)は無効なデバイス ID です。ターゲット デバイスとの接続を確認してください)

DISCONNECT_WHILE_BUSY=The tool was disconnected while it was busy. (ツールが動作中に切断されました)

EEDATA_MEMORY=EEData memory. (EEData メモリ)

EEDATA=EEData.

EMULATION_MEMORY_READ_WRITE_ERROR=An error occurred while trying to read/write MPLAB's emulation memory: Address=%08x. (MPLAB エミュレーション メモリの読み書き中にエラーが発生しました: アドレス =%08x)

END=end. (終了)

ENSURE_SELF_TEST_READY=Please ensure the RJ-11 cable is connected to the test board before continuing. (続行する前に、RJ-11 ケーブルがテストボードに接続されている事を確認してください)

ENSURE_SELF_TEST_READY=Please ensure the RJ-11 cable is connected to the test board before continuing.Would you like to continue? (続行する前に、RJ-11 ケーブルがテストボードに接続されている事を確認 してください。続行しますか)

ENV_ID_GROUP=Device Identification. (デバイスの識別情報)

ERASE_COMPLETE=Erase successful. (消去に成功しました)

ERASING=Erasing... (消去中...)

FAILED_2_PGM_DEVICE=Failed to program device. (デバイスのプログラミングに失敗しました)

FAILED_CREATING_COM=Unable create communications object (RI4Com). (通信オブジェクト(RI4Com)を作成で きません)

FAILED_CREATING_DEBUGGER_MODULES=Initialization failed: Failed creating the debugger module. (初期化に 失敗しました: デバッガ モジュールの作成に失敗しました)

FAILED_ESTABLISHING_COMMUNICATION=Unable to establish tool communications. (ツールの通信を確立できません)

FAILED_GETTING_DBG_EXEC=A problem occurred while trying to load the debug executive. (デバッグ実行プログ ラム読み込み中に問題が発生しました)

FAILED_GETTING_DEVICE_INFO=Initialization failed: Failed while retrieving device database (.pic) information. (初期化に失敗しました: デバイス データベース(.pic)情報の取得に失敗しました)

FAILED_GETTING_EMU_INFO=Initialization failed: Failed getting emulation database information. (初期化に失敗し ました: エミュレーション データベース情報の取得に失敗しました)

FAILED_GETTING_HEADER_INFO=Initialization failed: Failed getting header database information. (初期化に失敗 しました: ヘッダデータベース情報の取得に失敗しました)

FAILED_GETTING_PGM_EXEC=A problem occurred while trying to load the program executive. (プログラム実行 プログラム読み込み中に問題が発生しました)

FAILED_GETTING_TEX=Unable to obtain the ToolExecMediator. (ToolExecMediator を取得できません)

FAILED_GETTING_TOOL_INFO=Initialization failed: Failed while retrieving tool database (.ri4) information. (初期化 に失敗しました: ツールデータベース(.ri4)情報の取得に失敗しました)

FAILED_INITING_DATABASE=Initialization failed: Unable to initialize the tool database object. (初期化に失敗しました: ツールデータベース オブジェクトを初期化できません)

FAILED_INITING_DEBUGHANDLER=Initialization failed: Unable to initialize the DebugHandler object. (初期化に失敗しました: DebugHandler オブジェクトを初期化できません)

FAILED_PARSING_FILE=Failed to parse firmware file: %s.

FAILED_READING_EMULATION_REGS=Failed to read emulation memory. (エミュレーション メモリの読み出し に失敗しました)

FAILED_READING_MPLAB_MEMORY=Unable to read %s memory from %0x08 to %0x08. (%0x08 から%0x08 に%s メモリを読み出せません)

FAILED_SETTING_SHADOWS=Failed to properly set shadow registers. (シャドウレジスタの適切な設定に失敗しました)

FAILED_SETTING_XMIT_EVENTS=Unable to synchronize run time data semiphores. (ランタイムデータ セマフォ を同期できません)

FAILED_STEPPING=Failed while stepping the target. (ターゲットのステップ動作に失敗しました)

FAILED_TO_GET_DEVID=Failed to get Device ID.Please make sure the target device is attached and try the operation again. (デバイス ID の取得に失敗しました。ターゲット デバイスが接続されている事を確認し、操作を 再試行してください)

FAILED_TO_INIT_TOOL=Failed to initialize ICE 4. (ICD 4 の初期化に失敗しました)

FAILED_UPDATING_BP=Failed to update breakpoint:\nFile: %s\naddress: %08x. (ブレークポイントの更新に失敗 しました: \n ファイル: %s\n アドレス: %08x)

FAILED_UPDATING_FIRMWARE=Failed to properly update the firmware. (ファームウェアの適切な更新に失敗し ました)

FILE_REGISTER=file register. (ファイルレジスタ)

MPLAB[®] ICE 4インサーキット エミュレータ ユーザガイド ^{エラーメッセージ}

FIRMWARE_DOWNLOAD_TIMEOUT=ICE 4 timeout out during the firmware download process. (ファームウェア ダウンロード処理中に ICD 4 がタイムアウトしました)

FLASH_DATA_MEMORY=Flash data memory. (フラッシュデータ メモリ)

FLASH_DATA=flash data. (フラッシュデータ)

FPGA_VER=FPGA Version. (FPGA バージョン)

FRCINDEBUG_NEEDS_CLOCKSWITCHING=To use FRC in debug mode the clock switching configuration bits setting must be enabled.Please enable clock switching and retry the requested operation. (デバッグモードで FRC を使うには、クロック切り換えコンフィグレーション ビットの設定を有効にする必要があります。クロック切り換 えを有効にし、要求動作を再試行してください)

FW_DOESNT_SUPPORT_DYNBP=The current ICE 4 firmware does not support setting run time breakpoints for the selected device.Please download firmware version %02x.%02x.%02x or higher. (現在の ICD 4 ファームウェア は、選択されたデバイスに対するランタイム ブレークポイントの設定をサポートしていません。ファームウェア バージョン%02x.%02x.%02x 以降をダウンロードしてください)

GOOD_ID_MISMATCH=Target Device ID (0x%x) is a valid Device ID but does not match the expected Device ID (0x%x) as selected. (ターゲット デバイス ID (0x%x)は有効なデバイス ID ですが、選択した要求デバイス ID (0x%x)と一致しません)

HALTING=Halting... (停止中...)

HIGH=High.

HOLDMCLR_FAILED=Hold in reset failed. (リセット維持に失敗しました)

IDS_SELF_TEST_PASSED=ICE4 is functioning properly. If you are still having problems with your target circuit please check the Target Board Considerations section of the online help. (ICD 4 は正常に動作しています。ターゲット回路の 問題が解決しない場合、オンラインヘルプの[Target Board Considerations]セクションを確認してください)

IDS_ST_CLKREAD_ERR=Test interface PGC clock line read failure. (テストインターフェイス PGC クロックラインの読み出しエラーです)

IDS_ST_CLKREAD_NO_TEST=Test interface PGC clock line read not tested. (テストインターフェイス PGC クロックラインの読み出しテストはしていません)

IDS_ST_CLKREAD_SUCCESS=Test interface PGC clock line read succeeded. (テストインターフェイス PGC クロックラインの読み出しに成功しました)

IDS_ST_CLKWRITE_ERR=Test interface PGC clock line write failure. Please ensure that the tester is properly connected. (テストインターフェイス PGC クロックラインの書き込みエラーです。テスタが正しく接続されている 事を確認してください)

IDS_ST_CLKWRITE_NO_TEST=Test interface PGC clock line write not tested. (テストインターフェイス PGC クロックラインの書き込みテストはしていません)

IDS_ST_CLKWRITE_SUCCESS=Test interface PGC clock line write succeeded. (テストインターフェイス PGC クロックラインの書き込みに成功しました)

IDS_ST_DATREAD_ERR=Test interface PGD data line read failure. (テストインターフェイス PGD データラインの 読み出しエラーです)

IDS_ST_DATREAD_NO_TEST=Test interface PGD data line read not tested. (テストインターフェイス PGD デー タラインの読み出しテストはしていません)

IDS_ST_DATREAD_SUCCESS=Test interface PGD data line read succeeded. (テストインターフェイス PGD デー タラインの読み出しに成功しました)

IDS_ST_DATWRITE_ERR=Test interface PGD data line write failure. (テストインターフェイス PGD データライン の書き込みエラーです)

IDS_ST_DATWRITE_NO_TEST=Test interface PGD data line write not tested. (テストインターフェイス PGD デー タラインの書き込みテストはしていません)

IDS_ST_DATWRITE_SUCCESS=Test interface PGD data line write succeeded. (テストインターフェイス PGD データラインの書き込みに成功しました)

IDS_ST_LVP_ERR=Test interface LVP control line failure. (テストインターフェイス LVP 制御ラインのエラーです)

IDS_ST_LVP_NO_TEST=Test interface LVP control line not tested. (テストインターフェイス LVP 制御ラインのテ ストはしていません)

IDS_ST_LVP_SUCCESS=Test interface LVP control line test succeeded. (テストインターフェイス LVP 制御ライン のテストに成功しました)

IDS_ST_MCLR_ERR=Test interface MCLR level failure. (テストインターフェイス MCLR レベルエラーです)

IDS_ST_MCLR_NO_TEST=Test interface MCLR level not tested. (テストインターフェイス MCLR レベルのテスト はしていません)

IDS_ST_MCLR_SUCCESS=Test interface MCLR level test succeeded. (テストインターフェイス MCLR レベルテ ストに成功しました)

IDS_TEST_NOT_COMPLETED=Interface test could not be completed.Please contact your local FAE/CAE to SAR the unit. (インターフェイス テストを完了できませんでした。修理については販売店または Microchip 社にお問い合わせください)

INCOMPATIBLE_FW=The ICE 4 firmware in not compatible with the current version of MPLAB X software. (ICE 4 ファームウェアが MPLAB X ソフトウェアの現バージョンと互換ではありません)

INVALID_ADDRESS=The operation cannot proceed because the %s address is outside the devices address range of 0x%08x - 0x%08x. (%s アドレスがデバイスアドレス レンジ 0x%08x - 0x%08x を外れているため、動作 を実行できません)

MEM_RANGE_ERROR_BAD_END_ADDR=Invalid program range end address %s received.Please check the manual program ranges on the debug tool's, "Memories to Program" property page. (プログラムレンジで無効な終 了アドレス%s を受信しました。デバッグツールの[Memories to Program]プロパティページで手動プログラムレン ジを確認してください)

MEM_RANGE_ERROR_BAD_START_ADDR=Invalid program range start address %s received.Please check the manual program ranges on the debug tool's, "Memories to Program" property page. (プログラムレンジで無効な開始アドレス%s を受信しました。デバッグツールの[Memories to Program]プロパティページで手動プログラムレンジを確認してください)

MEM_RANGE_ERROR_END_LESSTHAN_START=Invalid program range received: end address %s < start address %s.Please check the manual program ranges on the debug tool's, "Memories to Program" property page. (無効なプログラムレンジを受信しました: 終了アドレス%s < 開始アドレス%s。デバッグツールの[Memories to Program]プロパティページで手動プログラムレンジを確認してください)

MEM_RANGE_ERROR_ENDADDR_NOT_ALIGNED=Invalid program range received: end address %s is not aligned on a proper 0x%x address boundary.Please check the manual program ranges on the debug tool's, "Me-mories to Program" property page. (無効なプログラムレンジを受信しました: 終了アドレス%s が正しい 0x%x アドレス境界にアラインメントされていません。デバッグツールの[Memories to Program]プロパティページ で手動プログラムレンジを確認してください)

MEM_RANGE_ERROR_STARTADDR_NOT_ALIGNED=Invalid program range received: start address %s is not aligned on a proper 0x%x address boundary.Please check the manual program ranges on the debug tool's, "Memories to Program" property page. (無効なプログラムレンジを受信しました: 開始アドレス%s が正しい 0x%x アドレス境界にアラインメントされていません。デバッグツールの[Memories to Program]プロパティページで手動 プログラムレンジを確認してください)

MEM_RANGE_ERROR_UNKNOWN=An unknown error has occurred while trying to validate the user entered memory ranges. (ユーザが入力したメモリ領域検証中に不明なエラーが発生しました)

MEM_RANGE_ERROR_WRONG_DATABASE=Unable to access data object while validating user entered memory ranges. (ユーザが入力したメモリ領域検証中にデータオブジェクトにアクセスできませんでした)

MEM_RANGE_OUT_OF_BOUNDS=The selected program range, %s, does not fall within the proper range for the memory area selected.Please check the manual program ranges on the debug tool's, "Memories to Program" p-roperty page. (選択したプログラムレンジ%s が、選択したメモリ領域のレンジを外れています。デバッグツールの[Memories to Program]プロパティページで手動プログラムレンジを確認してください)

MEM_RANGE_STRING_MALFORMED=The memory range(s) entered on the, "Memories to Program" property page (%s) is not formatted properly. ([Memories to Program]プロパティページで入力したメモリ領域(%s)が正しく フォーマットされていません)

MISSING_BOOT_CONFIG_PARAMETER=Unable to find boot config start/end address in database. (データベース 内にブート コンフィグレーションの開始/終了アドレスが見つかりません)

MUST_SET_LVPBIT_WITH_LVP=The low voltage programming feature requires the LVP configuration bit to be enabled on the target device.Please enable this configuration bit and try the operation again. (低電圧プログラミング 機能では、ターゲット デバイスで LVP コンフィグレーション ビットを有効にする必要があります。このコンフィ グレーション ビットを有効にしてから再試行してください)

NEW_FIRMWARE=Now Downloading new Firmware for target device: %s (ターゲット デバイス: %s の新しい ファームウェアをダウンロード中です)

NMMR=NMMR

NO_DYNAMIC_BP_SUPPORT_AT_ALL=The current device does not support the ability to set breakpoints while the devices is running. The breakpoint will be applied prior to the next time you run the device. (ご使用のデバイス は、動作中のブレークポイント設定機能をサポートしていません。ブレークポイントは次のデバイス動作前に適用されます)

NO_PGM_HANDLER=Cannot program software breakpoints.The program handler has not been initialized. (ソフト ウェア ブレークポイントをプログラムできません。プログラムハンドラが初期化されていません)

NORMAL=Normal.

OP_FAILED_FROM_CP=The requested operation failed because the device is code protected. (デバイスがコード 保護されているため動作は失敗しました)

OpenIDE-Module-Name=ICE 4

OPERATION_NOT_SUPPORTED=This operation is not supported for the selected device. (この動作は選択したデ バイスではサポートされていません)

OUTPUTWIN_TITLE=ICE 4.

PERIPHERAL=Peripheral. (周辺モジュール)

POWER_ERROR_NO_9V=The configuration is set for the tool to provide power to the target but the 9V power jack is not detected. Please ensure the external 9V barrel jack is connected to the tool. (ツールからターゲットに電力を 供給するように設定されていますが、9V 電源ジャックが検出されませんでした。ツールに外部9V ジャックが接 続されている事を確認してください)

POWER_ERROR_NO_POWER_SRC=The configuration is set for the target board to supply its own power but no voltage has been detected on VDD.Please ensure you have your target powered up and try again. (外部電源から ターゲットボードに給電するように設定されていますが、VDD に電圧が検出されませんでした。ターゲットに電力 が供給されている事を確認してから再試行してください)

POWER_ERROR_POWER_SRC_CONFLICT=The configuration is set for the tool to provide power to the target but there is voltage already detected on VDD. This is a conflict. Please ensure your target is not supplying voltage to the tool and try again. (ツールからターゲットに電力を供給するように設定されていますが、VDD には既に電圧 が検出されました。ターゲットからツールに電力を供給していない事を確認してから再試行してください)

POWER_ERROR_SLOW_DISCHARGE= There seems to be excessive capacitance on VDD causing a slower system discharge and shutdown.Consider minimizing overall capacitance loading or use power from your target to avoid discharge delays. (VDD の静電容量が大き過ぎるためにシステムの放電遅延とシャットダウンが発生している ようです。総静電容量負荷を抑えるか、ターゲットから給電して放電遅延を回避してください)

POWER_ERROR_UNKNOWN=An unknown power error has occurred. (不明な電源エラーが発生しました)

POWER_ERROR_VDD_TOO_HIGH=The VDD voltage desired is out of range. It exceeds the maximum voltage of 5.5V. (要求 VDD 電圧がレンジ外です。最大電圧 5.5 V を上回っています)

POWER_ERROR_VDD_TOO_LOW=The VDD voltage desired is out of range. It is below the minimum voltage of 1.5V. (要求 VDD 電圧がレンジ外です。最小電圧 1.5 V を下回っています)

POWER_ERROR_VPP_TOO_HIGH=The VPP voltage desired is out of range. It exceeds the maximum voltage of 14.2V. (要求 VPP 電圧がレンジ外です。最大電圧 14.2 V を上回っています)

POWER_ERROR_VPP_TOO_LOW=The VPP voltage desired is out of range. It is below the minimum voltage of 1.5V. (要求 VPP 電圧がレンジ外です。最小電圧 1.5 V を下回っています)

PRESERVE_MEM_RANGE_ERROR_BAD_END_ADDR=Invalid preserve range end address %s received.Please check the manual program ranges on the debug tool's, "Memories to Program" property page. (無効な保存領域終了 アドレス%s を受信しました。デバッグツールの[Memories to Program]プロパティページで手動プログラムレンジ を確認してください)

PRESERVE_MEM_RANGE_ERROR_BAD_START_ADDR=Invalid preserve range start address %s received.Please check the manual program ranges on the debug tool's, "Memories to Program" property page. (無効 な保存領域開始アドレス%s を受信しました。デバッグツールの[Memories to Program]プロパティページで手動プログラムレンジを確認してください)

PRESERVE_MEM_RANGE_ERROR_END_LESSTHAN_START=Invalid preserve range received: end address %s < start address %s.Please check the manual program ranges on the debug tool's, "Memories to Program" property page. (無効な保存領域を受信しました: 終了アドレス%s < 開始アドレス%s。デバッグツールの[Memories to Program] プロパティページで手動プログラムレンジを確認してください)

PRESERVE_MEM_RANGE_ERROR_ENDADDR_NOT_ALIGNED=Invalid preserve range received: end address %s is not aligned on a proper 0x%x address boundary.Please check the manual program ranges on the debug tool's, "Memories to Program" property page. (無効な保存領域を受信しました: 終了アドレス%s が正しい 0x%x アドレス境界にアラインメントされていません。デバッグツールの[Memories to Program]プロパティページで手動 プログラムレンジを確認してください)

PRESERVE_MEM_RANGE_ERROR_STARTADDR_NOT_ALIGNED=Invalid preserve range received: start address %s is not aligned on a proper 0x%x address boundary.Please check the manual program ranges on the debug tool's, "Memories to Program" property page. (無効な保存領域を受信しました: 開始アドレス%s が正しい 0x%x アドレス境界にアラインメントされていません。デバッグツールの[Memories to Program]プロパティページで手動プログラムレンジを確認してください)

PRESERVE_MEM_RANGE_ERROR_UNKNOWN=An unknown error has occurred while trying to validate the user entered preserve ranges. (ユーザが入力した保存領域検証中に不明なエラーが発生しました)

PRESERVE_MEM_RANGE_ERROR_WRONG_DATABASE=Unable to access data object while validating user entered memory ranges. (ユーザが入力したメモリ領域検証中にデータオブジェクトにアクセスできません)

PRESERVE_MEM_RANGE_MEM_NOT_SELECTED=You have selected to preserve an area of memory but have not selected to program that area.Please check the preserved ranges on the debug tool's, "Memories to Program" property page, and make sure that any preserved memory is also designated to be programmed. (メモ リ領域の保存を選択しましたが、その領域へのプログラムを選択していません。デバッグツールの[Memories to Program]プロパティページで保存領域を確認し、保存メモリのプログラムも指定しているか確認してください)

PRESERVE_MEM_RANGE_OUT_OF_BOUNDS=The selected preserve range, %s, does not fall within the proper range for the memory area selected.Please check the manual program ranges on the debug tool's, "Memories to Program" property page. (選択した保存領域%s が、選択したメモリ領域のレンジから外れています。デバッグツールの[Memories to Program]プロパティページで手動プログラムレンジを確認してください)

PRESERVE_MEM_RANGE_STRING_MALFORMED=The preserve memory range(s) entered on the, "Memories to Program" property page (%s) is not formatted properly. ([Memories to Program]プロパティページで入力した保存メモリ領域(%s)が正しくフォーマットされていません)

PRESERVE_MEM_RANGE_WONT_BE_PROGRAMMED=Some or all of the preserve memory ranges (%s) entered on the, "Memories to Program" property page, do not fall under the indicated program range(s) (%s) for the memory selected. Please check the preserved ranges on the debug tool's, "Memories to Program" property page. ([Memories to Program]プロパティページで入力した一部または全ての保存メモリ領域(%s)が、選択したメモ リ領域の指定プログラム領域(%s)から外れています。デバッグツールの[Memories to Program]プロパティページで 保存レンジを確認してください)

PROGRAM_COMPLETE=Programming/Verify complete. (プログラミング/ベリファイが完了しました)

PROGRAM_MEMORY=program memory. (プログラムメモリ)

PROGRAM=program. (プログラム)

PROGRAMMING_DID_NOT_COMPLETE=Programming did not complete. (プログラミングは完了しませんでした)

READ_COMPLETE=Read complete. (読み出しは完了しました)

READ_DID_NOT_COMPLETE=Read did not complete. (読み出しは完了しませんでした)

RELEASEMCLR_FAILED=Release from reset failed. (リセットからのリリースに失敗しました)

REMOVING_SWBPS_COMPLETE=Removing software breakpoints complete. (ソフトウェア ブレークポイントの 消去が完了しました)

REMOVING_SWBPS=Removing software breakpoints... (ソフトウェア ブレークポイント消去中...)

RESET_FAILED=Failed to reset the device. (デバイスのリセットに失敗しました)

RESETTING=Resetting... (リセット中...)

エラーメッセージ

RUN_INTERRUPT_THREAD_SYNCH_ERROR=An internal run error has occurred. It is advised that you restart your debug session. You may continue running but certain run time features may no longer work properly. (内部実行エラーが発生しました。デバッグ セッションをリスタートする事を推奨します。動作は続行可能ですが、一部のランタイム機能が正常に機能しなくなる可能性があります)

RUN_TARGET_FAILED=Unable to run the target device. (ターゲット デバイスを動作させる事ができません)

RUNNING=Running. (動作中)

SD_RESULT_NO_ERROR=Empty Trace File result (空のトレースファイル結果)

SERIAL_NUM=Serial Number:\n (シリアル番号:\n)

SETTING_SWBPS=Setting software breakpoints...... (ソフトウェア ブレークポイント設定中......)

STACK=stack. (スタック)

START_AND_END_ADDR=start address = 0x%x, end address = 0x%x. (開始アドレス = 0x%x、終了アドレス = 0x%x)

START=start. (開始)

TARGET_DETECTED=Target voltage detected. (ターゲット電圧を検出しました)

TARGET_FOUND=Target device %s found. (ターゲット デバイス%s が見つかりました)

TARGET_HALTED=Target Halted. (ターゲットが停止しました)

TARGET_NOT_READY_4_DEBUG=The target device is not ready for debugging. Please check your configuration bit settings and program the device before proceeding. The most common causes for this failure are oscillator and/or PGC/PGD settings. (ターゲット デバイスはデバッグの準備が完了していません。コンフィグレーション ビットの設定を確認し、デバイスをプログラミングしてからデバッグしてください。一般的に、このエラーはオ シレータと PGC/PGD の両方またはどちらか一方の設定が原因で発生します)

TARGET_VDD=Target VDD: (ターゲット VDD:)

TEST=test. (テスト)

TOOL_IS_BUSY=ICE 4 is busy.Please wait for the current operation to finish. (ICD 4 がビジーです。実行中の動作 が終了するまで待機してください)

TOOL_VDD=VDD:

TOOL VPP=VPP:

UNABLE_TO_OBTAIN_RESET_VECTOR=ICE 4 was unable to retrieve the reset vector address. This indicates that no _reset symbol has been defined and may prevent the device from starting up properly. (ICD 4 はリセット ベクタアドレスを取得できませんでした。_reset シンボルが未定義でデバイスの正常起動が妨げられている可能 性があります)

UNKNOWN_MEMTYPE=Unknown memory type. (不明なメモリタイプ)

UNLOAD_WHILE_BUSY=ICE 4 was unloaded while still busy.Please unplug and reconnect the USB cable before using ICE 4 again. (ICD 4 がまだビジーの間にアンロードされました。USB ケーブルを一度抜いて挿し直してくだ さい)

UPDATING_APP=Updating firmware application... (ファームウェア アプリケーション更新中...)

UPDATING_BOOTLOADER=Updating firmware bootloader... (ファームウェア ブートローダ更新中...)

UPDATING_FPGA=Updating firmware FPGA... (ファームウェア FPGA 更新中...)

USE_LVP_PROGRAMMING=NOTE: If you would like to program this device using low voltage programming, select Cancel on this dialog. Then go to the ICE4 node of the project properties and check the Enable Low Voltage Programming check box of the Program Options Option Category pane (low voltage programming is not valid for debugging operations). (低電圧プログラミングを使ってこのデバイスをプログラムする場合、このダイアログで「Cancel」を選択します。その後、プロジェクト プロパティの ICE4 ノードで[Program Options Option Category]ペインの[Enable Low Voltage Programming]チェックボックスにチェックを入れます(低電圧プログラミンがデバッグ動作に対して有効になっていません))

USERID_MEMORY=User Id Memory. (ユーザ ID メモリ)

USERID=user Id. (ユーザ ID)

MPLAB[®] ICE 4インサーキット エミュレータ ユーザガイド ^{エラーメッセージ}

VERIFY_COMPLETE=Verification successful. (ベリファイが成功しました)

VERIFY_FAILED=Verify failed. (ベリファイに失敗しました)

VERSIONS=Versions. (バージョン)

VOLTAGES=Voltages. (電圧)

WOULD_YOU_LIKE_TO_CONTINUE=Would you like to continue? (続行しますか)

8.2 一般的な対処方法

以下のセクションの一般的な対処法で問題を解決できる場合があります。

8.2.1 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、以下を確認してください。

- 1. [Debug] > [Reset]をクリックしませんでしたか。これによって読み書きエラーが発生する場合があります。
- 2. 同じ操作を繰り返します(一過性エラーの可能性があるため)。
- 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します。デバイスに必要な 電圧レベルは、デバイスのデータシートで確認します。
- 4. デバッガとターゲット間が正しく接続されている事(PGC と PGD が接続されている事)を確認します。
- 5. 書き込みエラーの場合、デバッガ用の[Program Options]で[Erase all before Program]にチェックが入っている 事を確認します(「プログラム」参照)。
- 6. ケーブルの長さが適切である事を確認します。

8.2.2 デバッガとターゲット間の通信エラーの対処方法

MPLAB ICE 4 インサーキット エミュレータとターゲット デバイスが通信していない可能性があります。

- 1. [Debug] > [Reset]を選択してから同じ操作を再試行します。
- 2. ケーブルの長さが適切である事を確認します。

8.2.3 デバッガとコンピュータ間の通信エラーの対処方法

MPLAB ICE 4 インサーキット エミュレータと MPLAB X IDE が通信していない可能性があります。

- 1. デバッガとコンピュータ間の接続ケーブルを一度抜いて挿し直します。
- 2. デバッガに再接続します。
- 3. 同じ操作を繰り返します(一過性エラーの可能性があるため)。
- 4. MPLAB X IDE にインストールされているパック(DPF/TP)のバージョンがターゲット デバイスに対応していな い可能性があります。更新されたパックのインストール方法は MPLAB X IDE の文書を参照してください。
- 5. コンピュータの USB ポートに問題があるかもしれません。「USB ポート通信エラーの対処方法」を参照して ください。

8.2.4 インストール ファイルが破損している場合の対処方法

ほとんどの場合、この問題は MPLAB X 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 をコンピュータからアンインストールします。
- 2. 最新バージョンの MPLAB X IDE を再インストールします。
- 3. それでも解決しない場合は Microchip 社にお問い合わせください。

8.2.5 USB ポート通信エラーの対処方法

ほとんどの場合、このエラーは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。

MPLAB[®] ICE 4インサーキット エミュレータ ユーザガイド ^{エラーメッセージ}

- 1. MPLAB ICE 4 インサーキット エミュレータに再接続します。
- 2. デバッガがコンピュータの適切な USB ポートに物理的に接続されている事を確認します。
- デバッガのオプションで適切な USB ポートが選択されている事を確認します(「デバッガ オプションの選択」参照)。
- 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します。
- 5. USB ハブを使う場合、電源が供給されている事を確認します。
- 6. USB ドライバがインストールされている事を確認します。

8.2.6 デバッグ障害の対処方法

MPLAB ICE 4 インサーキット エミュレータでデバッグを実行できない場合、各種の原因が考えられます(「トラブル シューティング」参照)。

8.2.7 内部エラーの対処方法

内部エラーは想定外のエラーであり、通常は発生しません。これらは主に Microchip 社内の開発で使います。多くの 場合、インストール ファイルの破損が原因です(「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一時的な不足によって発生する場合もあります。

- 1. システムを再起動してメモリを解放します。
- 2. HDD 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します。

それでも解決しない場合は Microchip 社にお問い合わせください。

MPLAB[®] ICE 4インサーキット エミュレータ ユーザガイド エミュレータの機能のまとめ

9. エミュレータの機能のまとめ

MPLAB[®] ICE 4 インサーキット エミュレータの機能を以下にまとめます。

9.1 エミュレータの選択と切り換え

プロジェクトで使うエミュレータの選択と切り換えには[Project Properties]ダイアログを使います。コンピュータ に複数の MPLAB ICE 4インサーキット エミュレータを接続している場合にのみ、エミュレータの切り換えが可能 です。MPLAB X IDE はシリアル番号でエミュレータを区別します。

プロジェクトで使うエミュレータを選択または切り換えるための手順は以下の通りです。

- 1. 以下の方法のどれか1つにより[Project Properties]ダイアログを開く
 - a. [Projects]ウィンドウでプロジェクト名をクリックして[File] > [Project Properties]を選択する
 - b. [Projects]ウィンドウでプロジェクト名を右クリックして[Properties]を選択する
- 2. [Categories]で[Conf: [default]]をクリックする
- 3. [Hardware Tools]に表示される[ICE 4]の中から、プロジェクトで使うエミュレータの SN(シリアル番号)をクリックする

9.2 エミュレータ オプションの選択

[Project Properties]ダイアログのエミュレータ プロパティページで、エミュレータ オプションを設定します。

- 1. 以下の方法のどれか1つにより[Project Properties]ダイアログを開く
 - a. [Projects]ウィンドウでプロジェクト名をクリックして[File] > [Project Properties]を選択する
 - b. [Projects]ウィンドウ内でプロジェクト名を右クリックして、[Properties]を選択する
- 2. [Categories]で[ICE 4]をクリックする
- [Options categories]からプロパティページを選択する。オプションをクリックすると、下のテキストボック スに説明が表示される。オプションを変更するには、オプション名の右側をクリックする。 Note: 表示されるオプションはデバイスによって異なる可能性があります。

エミュレータの機能のまとめ

図 9-1. [Project Properties] - ICE 4 のオプション

Project Properties - ICE4_PIC16F18877					— ×
Categories:	Options for ICE 4				
© File Indusion/Exclusion	Option categories:	Memories to Program			Reset
• • ICE 4	Auto select memorie	Debug Options		E to Select Memories	•
·····	Configuration Memo	Program Options ICE 4 Tool Options			=
O Building	EEPROM	Freeze Peripherals Power			
XC8 Global Options	Program Memory	Tool pack selection	1		
Code Coverage	Program Memory Ra	nge(s) (Enter ranges in h	0-7fff		
-	Option Description				¥
Manage Configurations					
Manage Network Tools					
		OK C	Cancel	Apply Unlock	Help

9.2.1 [Memories to Program]

プログラミングするターゲットのメモリを選択します。

[Program Options]の[Erase All Before Program]を選択している場合、デバイスの全メモリ領域はプログラミング 前に消去されます。消去後一部のメモリのみを選択してプログラミングするには、特定のメモリタイプにチェック を入れます。各種タイプのメモリの値を保存するには、そのメモリタイプにチェックを入れます。特定のメモリタ イプを保護する場合、[Preserve Memory]にチェックを入れると現在の内容が消去前にバッファに書き込まれ、 [Memory]にチェックを入れると消去後にそのメモリに書き戻されます(Memory はメモリのタイプ(EEPROM 等)で す)。

表 9-1.	[Memories to	Program]オフ	プション カテゴリ
--------	--------------	------------	-----------

[Auto select memories and ranges]	[Allow ICE 4 to Select Memories] - ユーザが選択したデバイスと既定値設定に基づ いて、エミュレータがプログラミングする内容を判断します。 [Manually select memories and ranges] - プログラミングするメモリのタイプとレ ンジを選択します(以下参照)。
[Configuration Memory] (Debug モードでは常に プログラミングされます)	チェックを入れると、コンフィグレーション メモリを Release モードでプログラミ ングします。デュアル パーティション デバイスの場合、パーティション 2 用に別の 値を選択できます。
[Memory]	チェックを入れると、メモリをプログラミングします。メモリタイプは以下から選 択できます。[Instruction RAM]、[Flash Data]、[Data Flash]、[EEPROM]、[ID]、 [Boot Flash]、[Auxiliary]
[Preserve Program Memory Start (hex)] [Preserve Program Memory End (hex)]*	メモリ領域の開始アドレスと終了アドレスを 16 進数で指定します。メモリタイプは 以下から選択できます。[Instruction RAM]
[Program Memory]	チェックを入れると、以下で指定したターゲット プログラムメモリ範囲に対してプ ログラミングを実行します。

エミュレータの機能のまとめ

[Program Memory Start (hex)] [Program Memory End (hex)]	プログラミング、読み出し、検証を行うプログラムメモリ領域の開始アドレスと終 了アドレスを 16 進数で指定します。 Note: ここで指定したアドレスレンジ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消去します。	
[Preserve Program Memory]	チェックを入れると、下の[Preserve Program Memory Start (hex)]と[Preserve Program Memory End (hex)]で指定したターゲット プログラムメモリ領域がプ ログラミングから保護されます。コード保護が有効になっていない事を確認し ます。	
[Preserve Program Memory Start (hex)] [Preserve Program Memory End (hex)]	プログラミング、読み出し、検証時に保護するターゲット プログラムメモリ領域の 開始アドレスと終了アドレスを 16 進数で指定します。 このメモリ領域の内容は、ターゲットから読み出されて既存の MPLAB X IDE メモリ にマージされます。	
[Preserve Memory]	チェックを入れると、メモリを保存して再プログラミングを実行します。メモリタ イプは以下から選択できます。[Instruction RAM]、[Flash Data]、[Data Flash]、 [EEPROM]、[ID]、[Boot Flash]、[Auxiliary] コード保護が有効になっていない事を確認します。	
[Preserve Memory Range Start (hex)] [Preserve Memory Range End (hex)]*	プログラミング、読み出し、検証時に保護するターゲットメモリ領域の開始アドレ スと終了アドレスを 16 進数で指定します。メモリタイプは以下から選択できます。 [Instruction RAM]、[Flash Data]、[Data Flash]、[EEPROM]、[Boot Flash]、 [Auxiliary] このメモリ領域の内容は、ターゲットから読み出されて既存の MPLAB X IDE メモリ にマージされます。	
	コード保護が有効になっていない事を確認します。	
* レンジが正しくないためにプログラミング エラーが発生した場合、レンジが利用可能な/残りのデバイスメモリ領		

9.2.2 [Debug Options]

プロジェクト デバイスがこの機能に対応している場合、デバッグ オプションを選択します。

表 9-2. [Debug Options]オプション カテゴリ

[Debug startup]	システム設定は <u>[Tools] > [Options] > [Embedded] > [Generic Settings]</u> にありますが、ここで変 更できるものもあります。[Use system settings]、[Run]、[Halt at main]、[Halt at reset vector]
[Debug reset]	システム設定は[<u>Tools</u>] > [<u>Options</u>] > [<u>Embedded</u>] > [<u>Generic Settings</u>]にありますが、ここで変 更できるものもあります。[Use system settings]、[Main]、[ResetVector]
[Use Software Breakpoints]	ソフトウェア ブレークポイントを使う場合、チェックを入れます。 ハードウェア ブレークポイントを使う場合、チェックを外します。どちらのブレークポイン トが適当かは、下表を参照してください。
[Use Simultaneous Debug]	プロジェクトがマルチコア同時デバッグ セッションの一部である事を示す場合、チェックを 入れます。

表 9-3. ソフトウェア ブレークポイントとハードウェア ブレークポイントの比較

機能	ソフトウェア ブレークポイント	ハードウェア ブレークポイント
ブレークポイントの数	制限なし	制限あり
ブレークポイントの 書き込み先	プログラムメモリ	デバッグレジスタ
ブレークポイントの設定に 要する時間	オシレータ速度に依存 - 数分かかる場合があり ます	最小
スキッド	なし	あり

Note: デバッグにソフトウェア ブレークポイントを使うと、デバイスの書き込み耐性に影響します。従って、ソフトウェア ブレークポイントを使ったデバイスは量産製品に使わない事を推奨します。

エミュレータの機能のまとめ

9.2.3 [Program Options]

プログラミング前にメモリ全体を消去するか、コードをマージするかを選択します。

表 9-4. [Program Options]オプション カテゴリ

[Erase All Before Program]	チェックを入れると、プログラミング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プログラミングする場合を除き、このボックスに は必ずチェックを入れます。チェックを入れないとデバイスは消去されず、デバイス上で既 存のコードと新しいコードがマージされます。
[Do not erase auxiliary memory]	補助メモリをサポートするデバイスにのみ適用されます。 プログラミング時に消去しない補助メモリにチェックを入れます。 補助メモリをプログラミング時に消去する場合、チェックを外します。

9.2.4 [ICE 4 Tool Options]

MPLAB ICE 4 固有のオプションを設定します。

表 9-5. [ICE 4 Tool Options]カテゴリ

[Programming Mode Entry]	プログラミング モード エントリタイプを選択します。 ・ [Use high voltage program mode entry] ・ [Use low voltage program mode entry]
[LED Brightness Setting]	ユニットの LED の明るさを 1(最も暗い)~10(最も明るい)で設定します。
[PGC Configuration]	プログラミング クロックピンの設定。 ・ [none] ・ [pull up] ・ [pull down]
[PGC resistor value (kohms)]	上記で[pull up]または[pull down]を選択した場合、抵抗値を 0~50 kΩの範囲で入力し ます。
[PGD Configuration]	プログラミング データピンの設定。 ・ [none] ・ [pull up] ・ [pull down]
[PGD resistor value (kohms)]	上記で[pull up]または[pull down]を選択した場合、抵抗値を 0~50 kΩの範囲で入力し ます。

9.2.5 [Freeze Peripherals]

コード実行停止時に周辺モジュールをフリーズさせるかどうかを選択します。選択肢はデバイスによって異なります。

表 9-6. [Freeze Peripherals]オプション カテゴリ

[Freeze Peripherals]	実行停止時に全ての周辺モジュールをフリーズさせる場合、チェックを入れます。 実行停止時に周辺モジュールをフリーズさせない場合、チェックを外します。
	このオフションは PIC12/16/18 MCU に適用されます。
[Peripheral Freeze Enable Peripheral List]	実行停止時にフリーズさせる周辺モジュールを選択する場合、チェックを入れま す。実行停止時に周辺モジュールをフリーズさせない場合、チェックを外します。
	このオプションは AC244066 に適用されます。
[Peripheral List]	実行停止時に周辺モジュール「Peripheral」をフリーズさせる場合、チェックを入 れます。実行停止時に周辺モジュール「Peripheral」をフリーズさせない場合、 チェックを外します。
	このオプションは 16 ビットおよび 32 ビット MCU に適用されます。

エミュレータの機能のまとめ

PIC12/16/18 MCU デバイスの場合

[Freeze Peripherals]チェックボックスにチェックを入れると、実行停止時に全ての周辺モジュールがフリーズしま す。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、デバッガから制御する事はできません。 そのような周辺モジュールはチェックを入れても停止しません。

dsPIC、PIC24、PIC32 デバイスの場合

リストの周辺モジュールにチェックを入れると、実行停止時にその周辺モジュールがフリーズします。目的の周辺 モジュールがリストにない場合、[Freeze All Other Peripherals]にチェックを入れます。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、デバッガから制御する事はできません。そのような周辺モジュールは チェックを入れても停止しません。

9.2.6 [Trace and Profiling]

デバイスによっては、デバッグ時にトレース、PC サンプリング/プロファイリング、その他のデータ収集機能を使える場合があります。これらの機能は以下の方法で有効化し、設定します。

8 ビットおよび 16 ビットデバイスの場合

このページで利用可能なオプションは、プロジェクト デバイスのトレース/プロファイリング機能によって異なります。

表 9-7. [Trace and Profiling]オプション カテゴリ

[Data Collection Selection]	 データ収集を有効化/無効化します。 Off - ターゲットデータを収集しません。 User Instrumented Trace PC Sampling Power Monitor (Target Power Sampling)
[Data File Path and Name]	 データの保存に使うファイルのパスまたは名前を設定します。 ファイル名を入力する(プロジェクトに対する相対パス) - 推奨 パスとファイル名を入力する(絶対パス) ファイルを参照し、[Absolute]を選択し、そのファイルを選択し、[Save]をクリックする(絶対パス)
	Rote: ファイルを選択する際は、[Relative]を選択しない事が必要です。[Relative]を選 択すると、MPLAB X IDE はファイルを見つける事ができません。デバッグを実行した 時に、パスが見つからないという警告メッセージが表示されます。
[Data File Maximum Size (bytes)]	データファイルの最大サイズを設定します。 ターゲット電源のサンプリングには、サンプルあたり 12 バイトまたは 18 バイト(PC データ含む)を使います。
	選択したトレースタイプに応じて、ファイルサイズは 12 バイトまたは 18 バイトの倍数 に調整される場合があります。他のトレースデータ タイプでは、これ以外のレコードバ イト サイズが使えます。
[Data Buffer Maximum Size (bytes)]	データバッファの最大サイズを設定します。最大値はエミュレータが実装している 54,600 バイトです。 ターゲットの動作中にトレース/サンプリング データがメモリ内にバッファリングされ る場合、トレースまたはサンプルの各エントリサイズは、トレース/サンプルタイプと 使用するデバイスおよびツールによって異なります。通常、このバッファをなるべく 大きくする事を推奨します。
	例えば、命令トレースを備えた拡張 PIC16 は各メモリ内エントリに対して 1~3 バイ トを使います。さらに、各エントリが 13 バイトの ICE4 命令トレースエントリを生成 します。データファイル サイズの説明に示した通り、通常、メモリ内の各レコードは トレースデータ ファイルのエントリ行に変換されます(トレース/サンプリング ファイ ルのエントリサイズについてのデータファイル サイズの説明参照)。
[Stall CPU When Trace Buffer is Full]	トレースバッファが一杯になったら実行を停止します。上記オプションでバッファサイ ズを設定します。

エミュレータの機能のまとめ

[User Instrumented Trace]の項目		
[Disable Trace Macros]	チェックを入れるとトレースマクロが無効になり、外すと有効になります。 トレースを無効にするには、全てのマクロを削除して[Data Collection Selection]の[Off]を 選択します。	
[Communications Medium]	デバイスによっては、トレース方式として[Native]、[l/O Port]、[SPI]のいずれかを選択で きます。	
[I/O Port Selection]	I/O ポートトレースに使うデバイスのポートを指定します。 使えるポートが表示されます。	
[SPI Selection]	SPI トレースに使うデバイスの SPI ピンを指定します。使えるポートが表示されます。	
[PC Sampling]の項目		
[Timer Selection] (アプリケーション コードでは使いませ	PC サンプルの計数に使うデバイスのタイマを選択します。 Note: 選択したタイマは PC サンプリング専用となり、アプリケーションでは使えなく なります。	
<i>N</i> ₀)	Note: 選択できるタイマは 1 つだけです。2 つのタイマを組み合わせて 32 ビットタ イマにする事はできません。32 ビットタイマペアの一方を使うと、そのペア全体が 32 ビットタイマとして動作しなくなります。	
[Timer Interrupt Priority]	タイマの割り込み優先度を選択します。 Note: アプリケーションで設定済みの他の優先度よりも高い優先度を選択します。そう しないと他の優先度がサンプリングの優先度を上回るため、サンプルを取得できなくな ります。	
[Timer Interval]	サンプリング間隔を入力します。 整数値で指定します。	
	データをキャプチャできない場合、現在のサンプリング間隔ではサンプルを取得できて いない可能性があります。サンプリング間隔の単位選択と値を調整してみてください。 例えば 1 millisecond だった場合、990 milliseconds を試します。	
[Timer Interval Units]	サンプリング間隔の単位を以下より選択します。 microseconds milliseconds seconds instruction cycles 	

32 ビットデバイスの場合

このページで利用可能なオプションは、プロジェクト デバイスのトレース/プロファイリング機能によって異なります。

表 9-8. [Trace and Profiling]オプション カテゴリ

[Data Collection Selection]	データ収集を有効化/無効化します。 • Off - ターゲットデータを収集しません。 • Instruction Trace/Profiling • User Instrumented Trace • Power Monitor (Target Power Sampling)
[Data File Path and Name]	 データの保存に使うファイルのパスまたは名前を設定します。 ファイル名を入力する(プロジェクトに対する相対パス) - 推奨 パスとファイル名を入力する(絶対パス) ファイルを参照し、[Absolute]を選択し、そのファイルを選択し、[Save]をクリックする(絶対パス) Note: ファイルを選択する際は、[Relative]を選択しない事が必要です。[Relative]を選択すると、MPLAB X IDE はファイルを見つける事ができません。デバッグを実行した時に、パスが見つからないという警告メッセージが表示されます。

エミュレータの機能のまとめ

[Data File Maximum Size (bytes)]	データファイルの最大サイズを設定します。 エミュレータを使う場合、トレースデータ ファイル内の各命令トレースデータ行には 13 バイト必要です。	
	ターゲット電源のサンプリングには、サンプルあたり 12 バイトまたは 18 バイト(PC デー タ含む)を使います。	
	選択したトレースタイプに応じて、ファイルサイズは 12 バイトまたは 18 バイトの倍数に 調整される場合があります。他のトレースデータ タイプでは、上記以外のレコードバイト サイズを使用できます。	
[Data Buffer Maximum Size (bytes)]	データバッファの最大サイズを設定します。最大値はエミュレータが実装している 54,600 バイトです。 ターゲットの動作中にトレース/サンプリング データがメモリ内にバッファリングされる 場合、トレースまたはサンプルの各エントリサイズは、トレース/サンプルタイプと使用 するデバイスおよびツールによって異なります。通常、このバッファをなるべく大きく する事を推奨します。 例えば PIC32 命令トレースは「フレーム」あたり8バイトを使い、さらに 50 を超える 13 バイトの ICE 4 命令トレースエントリをトレースファイルに生成します。	
[User Instrumented Trace]の項目		
[Disable Trace Macros]	チェックを入れるとトレースマクロが無効になり、外すと有効になります。 トレースを無効にするには、全てのマクロを削除して[Data Collection Selection]の[Off]を 選択します。	
[Communications Medium]	デバイスによっては、トレース方式として[Native]、[I/O Port]、[SPI]のいずれかを選択で きます。ネイティブ	

9.2.7 [Power]

エミュレータからターゲットに給電するかどうかを選択します。

表 9-9. [Power]オプション カテゴリ

[Power Target Circuit	エミュレータからの電源を使う場合はチェックを入れます。
from ICE 4]	独立した電源からターゲットに給電する場合はチェックを外します。
[Voltage Level]	上の[Power target circuit from ICE 4]オプションにチェックを入れた場合に、エミュ レータからターゲットに供給する Vdd (2.375~5.5 V)を選択します。

9.2.8 [Clock]

[Clock]オプション カテゴリではランタイム クロック(命令)の速度を入力します。このオプションは速度を設定する のではなく、ランタイム ウォッチ、データキャプチャ、トレースに使う値をエミュレータに通知します。

表 9-10. [Clock]オプション カテゴリ

[Use FRC in Debug mode] (dsPIC33E/F と PIC24E/F/H のみ)	デバッグ中のクロックとして、アプリケーションが指定するオシレータ の代わりに、デバイスの FRC(高速内部 RC)を使います。これはアプリ ケーション クロックが低速の場合に便利な機能です。 このボックスにチェックを入れると、アプリケーションは低速で動作し ますが、デバッグはより高速な FRC 速度で実行できます。
	この設定を変更した後は、再プログラミングが必要です。
	Note: フリーズしていない周辺モジュールは、デバッグ中に FRC 速度で 動作します。

エミュレータの機能のまとめ

[Target run-time instruction speed]	選択した[Speed unit]に対する値を入力します。 例 1 : PIC24 MCU を使い、ターゲット クロック オシレータが 32 MHz (HS)で動作する場合、命令速度 = 32 MHz/2= 16 MIPS
	例 2 : PIC18F8722 MCU を使い、ターゲット クロック オシレータに 10 MHz (HS)と PLL (x4 = 40 MHz)を使う場合、命令速度 = 40 MHz/4 = 10 MIPS
[Instruction speed units]	以下のどちらかを選択します。 KIPS - 毎秒 1000 (10 ³)命令 MIPS - 毎秒 100 万(10 ⁶)命令

9.2.9 [Communications]

MPLAB ICE 4 とターゲットとの間で使う通信のタイプを選択します。プロジェクト デバイスによって表示される 通信オプションが異なります。以下に複数の例を挙げます。

詳細は3.3.「ターゲットとの接続」と4.「動作」を参照してください。

表 9-11. [Communications]オプション - ATtiny MCU

[Interface]	UPDI
[Speed]	変更する既定値を選択します。 この行をクリックすると、利用可能な速度値の範囲が[Option Description]の下に表示されます。
[High Voltage Activation Mode]	[No High Voltage]、[Simple High Voltage Pulse]、[User Power Toggle] この行をクリックすると、選択内容の定義が[Option Description]の下に表示されます。「UPDI High-Voltage Activation Information」も参照してください。

表 9-12. [Communications]オプション - ATmega MCU

[Interface]	debugWIREISP
[Speed]	変更する既定値を選択します。 この行をクリックすると、利用可能な速度値の範囲が[Option Description]の下に表示されます。「Debugging with debugWIRE for AVR [®] MCUs」も参照してください。

表 9-13. [Communications]オプション - SAM MCU

[Interface]	・ JTAG ・ SWD - JTAG のサブセット
[Speed]	変更する既定値を選択します。 この行をクリックすると、利用可能な速度値の範囲が[Option Description]の下に表示されます。

表 9-14. [Communications]オプション - PIC32M MCU

[JTAG Method]	JTAG 2-wire
	JTAG 4-wire

9.2.10 [Tool Pack Selection]

最新のツールパックを使うか、プロジェクト デバイスをサポートする別のバージョンを使うかを選択します。
エミュレータの機能のまとめ

表 9-15. [Tool Pack Selection]オプション カテゴリ

[Tool pack update options]	[Use latest installed tool pack] (推奨) - インストールされているツールパックの最新バー ジョンを使います。[Use specific tool pack] - その他の利用可能なツールパック バー ジョンのリストから選択します。
[Specifically selected version]	ここをクリックすると、ツールパックのバージョンを選択するダイアログが開きます。

9.3 エミュレータのウィンドウとダイアログ

以下のトピックに示すウィンドウとダイアログは、主にエミュレータまたはその他のエミュレータ関連デバッグ ツールで使うものです。

9.3.1 [Trace]ウィンドウと関連ダイアログ

[Trace]ウィンドウは、トレース結果を表示します。このウィンドウは、エミュレータとシミュレータで使えます。

図 9-2. [Trace]ウィンドウ

Wat	ches	Outp	out	Trace	- MyXC32Proj	ect ≋	Variable	s C	all Stac	k Breakpoints			6
	Line		Addı	ress	Op	Lab	el	Instru	action	1			
22	52356		9D00	0150	1440FFFC		F	SNE VO	, Z	•			
_	52357		9D00	0154	00000000		1	IOP				1	
11	52358		9D00	0144	3C02BF80		1	UI VO	, ✓	Symbolic Mode			=
_	52359		9D00	0148	8C420610		I	W V0,		Auto Decode 16 bit			- 100
Q	52360		9D00	014C	2C420190		5	SLTIU	v				
	52361		9D00	0150	1440FFFC		I	SNE VO	,	Go To	. •		
	52362		9D00	0154	00000000		1	IOP	-	Go To Source Line			
-	52363		9D00	0144	3C02BF80		I	UI VO	,	Clear Trace File			
♣	52364		9D00	0148	8C420610	-	I	W V0,	-	Clear Trace File			
	52365		9D00	014C	2C420190		5	SLTIU	v	Reload View			
	52366		9D00	0150	1440FFFC		E	SNE VO	,	Find			
	52367		9D00	0154	00000000		1	IOP					
	52368		9D00	0144	3C02BF80		I	UI VO	,	Output To File			
	52369		9D00	0148	8C420610		1	W V0,	-	Print			
	52370		9D00	014C	2C420190		5	SLTIU	v	Adjust Table Column			
	E0271		0000	0150	14405550		τ	ME TO		Aujust Table Column			
			My)	XC32Proj	iect (Build, Loa	H,)	2	debugg	er halte	d 3		134:1	1

上記ウィンドウのコラム内で右クリックすると、機能リストを含むコンテクスト メニューが表示されます。これら の機能の詳細は『MPLAB X IDE ユーザガイド』またはヘルプファイル、「MPLAB X IDE のウィンドウとダイアロ グ」、「[Trace]ウィンドウ」を参照してください。

関連リンク

5.5. PIC MCU と dsPIC DSC の計測機能付きトレース 5.6. PIC32M MCU の命令トレース

9.3.2 [ITM]ウィンドウと関連ダイアログ

SAM ITM (Instruction Trace Microcell)は、printf スタイルのデバッグをサポートする UART フォーマットのトレースを生成します。

エミュレータの機能のまとめ

Variables	Call Stack	Breakpoints	Output	ITM Display - port 0 ×	
Cutput	Format: ASCI	I 🗸 🗌 Show por	t number		
LOHELI	OHELLOHELLOH	FLLOHELLOHELL	OHELLOHELI	OHELLOHELLOHELLOHELLO	OHELLOHELLOHELLOHELLOHELLOHELLOHELLOHEL
ELLOHE	LLOHELLOHELI	OHELLOHELLOHE	LLOHELLOHE	LLOHELLOHELLOHELLOHE	LOHELLOHELLOHELLOHELLOHELLOHELLOHELLOHE
OHELLO	HELLOHELLOHE	LLOHELLOHELLO	HELLOHELLO	HELLOHELLOHELLOHELLO	HELLOHELLOHELLOHELLOHELLOHELLOHELLOHELL
LLOHEL	LOHELLOHELLO	HELLOHELLOHEL	LOHELLOHEI	LOHELLOHELLOHELLOHEL	LOHELLOHELLOHELLOHELLOHELLOHELLOHELLOHE
HELLOH	ELLOHELLOHEI	LOHELLOHELLOH	ELLOHELLOH	ELLOHELLOHELLOHELLOHI	ELLOHELLOHELLOHELLOHELLOHELLOHELLOHELLO
LOHELI	OHELLOHELLOH	ELLOHELLOHELL	OHELLOHELI	OHELLOHELLOHELLOHELLO	OHELLOHELLOHELLOHELLOHELLOHELLOHELLOHEL
ELLOHE	LLOHELLOHELL	OHELLOHELLOHE	LLOHELLOHE	LLOHELLOHELLOHELLOHE	LLOHELLOHELLOHELLOHELLOHELLOHELLOHEL
OHELLO	HELLOHELLOHE	LLOHELLOHELLO	HELLOHELLO	HELLOHELLOHELLOHELLOH	HELLOHELLOHELLOHELLOHELLOHELLOHELLOHELL
LLOHEI	LOHELLOHELLO	HELLOHELLOHEL	LOHELLOHEI	LOHELLOHELLOHELLOHEL	LOHELLOHELLOHELLOHELLOHELLOHELLOHELLOHE
HELLOH	ELLOHELLOHEL	LOHELLOHELLOH	ELLOHELLOH	ELLOHELLOHELLOHELLOH	ELLOHELLOHELLOHELLOHELLOHELLOHELLOHELLO
LOHELI	OHELLOHELLOH	ELLOHELLOHELL	OHELLOHELI	OHELLOHELLOHELLOHELLO	OHELLOHELLOHELLOHELLOHELLOHELLOHELLOHEL
ELLOHE	LLOHELLOHELI	OHELLOHELLOHE	LLOHELLOHE	LLOHELLOHELLOHELLOHE	LLOHELLOHELLOHELLOHELLOHELLOHELLOHEL
OHELLO	HELLOHELLOHE	LLOHELLOHELLO	HELLOHELLO	HELLOHELLOHELLOHELLOH	HELLOHELLOHELLOHELLOHELLOHELLOHELLOH
LLOHEI	LOHELLOHELLO	HELLOHELLOHEL	LOHELLOHEL	LOHELLOHELLOHELLOHEL	LOHELLOHELLOHELLOHELLOHELLOHELLOHELLOHE
HELLOH	ELLOHELLOHEI	LOHELLOHELLOH	ELLOHELLOH	ELLOHELLOHELLOHELLOH	ELLOHELLOHELLOHELLOHELLOHELLOHELLOHELLO
LOHELI	OHELLOHELLOH	ELLOHELLOHELL	OHELLOHELI	OHELLOHELLOHELLOHELLO	OHELLOHELLOHELLOHELLOHELLOHELLOHELLOHEL
ELLOHE	LLOHELLOHELI	OHELLOHELLOHE	LLOHELLOHE	LLOHELLOHELLOHELLOHE	LLOHELLOHELLOHELLOHELLOHELLOHELLOHEL
OHELLO	HELLOHELLOHE	LLOHELLOHELLO	HELLOHELLO	HELLOHELLOHELLOHELLOH	HELLOHELLOHELLOHELLOHELLOHELLOHELLOH
LLOHEI	LOHELLOHELLO	HELLOHELLOHEL	LOHELLOHEI	LOHELLOHELLOHELLOHEL	LOHELLOHELLOHELLOHELLOHELLOHELLOHELLOHE
HELLOH	ELLOHELLOHEI	LOHELLOHELLOH	ELLOHELLOH	ELLOHELLOHELLOHELLOH	ELLOHELLOHELLOHELLOHELLOHELLOHELLOHELLO
LOHELI	OHELLOHELLOH	ELLOHELLOHELL	OHELLOHELI	OHELLOHELLOHELLOHELLO	OHELLOHELLOHELLOHELLOHELLOHELLOHELLOHEL
ELLOHE	LLOHELLOHELI	OHELLOHELLOHE	LLOHELLOHE	LLOHELLOHELLOHELLOHE	LLOHELLOHELLOHELLOHELLOHELLOHELLOHEL
OHELLO	HELLOHELLOHE	LLOHELLOHELLO	HELLOHELLO	HELLOHELLOHELLOHELLO	HELLOHELLOHELLOHELLOHELLOHELLOHELLOHELL
LLOHEI	LOHELLOHELLO	HELLOHELLOHEL	LOHELLOHEI	LOHELLOHELLOHELLOHEL	LOHELLOHELLOHELLOHELLOHELLOHELLOHELLOHE
HELLOH	ELLOHELLOHEI	LOHELLOHELLOH	ELLOHELLOH	ELLOHELLOHELLOHELLOH	ELLOHELLOHELLOHELLOHELLOHELLOHELLOHELLO
LOHELI	OHELLOHELLOH	ELLOHELLOHELL	OHELLOHELI	OHELLOHELLOHELLOHELLO	OHELLOHELLOHELLOHELLOHELLOHELLOHELLO
ELLOHE	LLOHELLOHELI	OHELLOHELLOHE	LLOHELLOHE	LLOHELLOHELLOHELLOHE	LLOHELLOHELLOHELLOHELLOHELLOHELLOHEL
OHELLO	HELLOHELLOHE	LLOHELLOHELLO	HELLOHELLO	HELLOHELLOHELLOHELLOH	HELLOHELLOHELLOHELLOHELLOHELLOHELLOHELL
LLOHEI	LOHELLOHELLO	HELLOHELLOHEL	LOHELLOHEI	LOHELLOHELLOHELLOHEL	LOHELLOHELLOHELLOHELLOHELLOHELLOHELLOHE

エミュレータの機能のまとめ

図 9-4. ITM の[Select Ports]ダイアログ

Notifications Output ITM Display - port 0 ×	
Output Format: ASCII V Show port number	
No de Pleas Select Ports	×
To ad Select Ports	
0 7 8 15 16 23 24 31	j
Output Settings History Size: 50000	
OK Cance	I

関連リンク 5.7.2.1. SAM ITM トレース MPLAB[®] ICE 4インサーキット エミュレータ ユーザガイド ハードウェア仕様

10. ハードウェア仕様

以下では、MPLAB ICE 4 ユニットと関連ハードウェアについて詳細に説明します。

10.1 エミュレータ ユニット

MPLAB ICE 4 インサーキット エミュレータ ユニットの分類は以下の通りです。

- ・ ラボ環境では Class A デバイス
- 家庭やオフィス環境では Category C デバイス
- 動作温度レンジ: 0~70℃

10.2 電源仕様

外部電源コネクタ(J2)

コネクタは 9 V DC +/- 5%センタープラス 2.5 mm ピン内径/5.5 mm 外径です。

この接続部は、9V-AC 電源アダプタ(製品番号 AC002014)、110~220 V 汎用電源によりエミュレータに給電し、最大1Aの電力をターゲット アプリケーションに供給するように設計されています。

関連リンク

3.1. 電源とセルフテスト

10.3 インジケータ ランプ(LED)

MPLAB ICE 4 ユニットの上部には対向する 2 本のライトパイプがあり、それぞれ独立した LED で発光します。



本デバッガの起動シーケンスは以下の通りです。

- 1. 紫 約3秒間点灯します。
- 2. 青 デバッガがパワーオン セルフテストを実行中に約2秒間点滅します。
- 3. 青-点灯します。デバッガは動作準備が整っています。

下表にインジケータライトの見方を示します。

表 10-1. LED とブートローダ エラーの説明

LED 1	LED 2	説明			
通常動作モード					
青	青	電源接続済み、スタンバイ状態、 ネットワーク接続準備完了			
白	青	ネットワーク接続済み			

ハードウェア仕様

続き					
LED 1	LED 2	説明			
白、遅い点滅	青	MPLAB X IDE/MPLAB IPE が ネットワークを介して ICE 4 との 通信を開始			
赤	青	ネットワーク接続障害/エラー			
橙	青	[Power target circuit from ICE 4]に チェックが入っている場合			
禄	青	[Power target circuit from ICE 4]に チェックが入っていない場合			
緑、遅い点滅	青	DGI接続済み			
紫	紫	ブートローダが動作中です。			
黄	黄	デバッガはビジーです。			
赤	赤	動作不良です。			
ブートローダ エラー					
紫	赤、遅い点滅	デバッガのシリアル EEPROM に アクセスできません。			
紫	赤、速い点滅	ブートローダ API コマンドを処理で きません。			
白、速い点滅	白、速い点滅	ツール ファームウェアでランタイム 例外が発生しました。			

10.4 PC 接続の仕様

MPLAB ICE 4 インサーキット エミュレータは以下のいずれかの方法を使って PC(と MPLAB X IDE)に接続できます。

接続タイプ	接続の詳細	プログラミングと デバッグ	トレース
USB Type-C(既定値)	SS USB 3.0、HS USB 2.0	USB 2.0 最大 480 Mbps	USB 3.0 最大 4.8 Gbps
Wi-Fi	直接またはアクセスポイント 経由	最大 28 Mbps	なし
Ethernet	直接またはネットワーク経由	最大 100 Mbps	なし

MPLAB ICE 4 ユニットに備わっている通信用のコネクタについて以下の項目で説明します。

10.4.1 USB Type C コネクタ(J1)およびケーブル

デバッガとコンピュータ間の USB 3.0 SS または USB 2.0 通信用に Type C USB コネクタおよびケーブルが付属しています。



10.4.2 Ethernet コネクタ(J6)およびケーブル

8 ピン RJ-45 コネクタと標準 Ethernet CAT5e/CAT6 ケーブルを使うと、MPLAB ICE 4 ユニットと PC(と MPLAB X IDE)の間で Ethernet 通信が可能です。コネクタは LAN アクティビティを示す LED を 2 つ備えています。Ethernet ケーブルは、コネクタにしっかり接続できるようにツメ折れ防止カバーがないものを推奨します。

Note: Ethernet ケーブルは MPLAB ICE 4 キットに付属していません。

図 10-1. MPLAB ICE 4 上の Ethernet コネクタ



表 10-2. Ethernet コネクタのピン配置

ピン番号	機能
1	TX+
2	TX-
3	RX+
4	EGND
5	EGND
6	RX-
7	EGND
8	EGND
FGND: 笹休グランド	·

表 10-3. コネクタ LED

LED の位置	LED の色	LED の機能
左上	緑	LAN ACT
右上	黄	LAN LINK

10.5 ターゲット接続の仕様

MPLAB ICE 4 ユニットには、ターゲットとの通信用に以下のコネクタが用意されています。

10.5.1 40 ピンエミュレータ ユニットコネクタ(J10)

この 40 ピンコネクタはターゲットとの高速通信を支援する為、Edge Rate ハイスピード同軸ケーブルアッセンブリ を用いて接続可能としています。 図 10-2. MPLAB ICE 4 上の 40 ピンコネクタ



図 10-3. エミュレータ コネクタのピン配置



表 10-4. エミュレータ ユニット上のエミュレータ コネクタピン機能

ピン	説明	機能	ピン	説明	機能
1	TVDD PWR	TVDD PWR	2	TVDD PWR	TVDD PWR
3	TVPP IO	TVPP/MCLR、nMCLR、RST	4	TPGD IO	TPGD IO、TPGD、SWO、TDO、 MISO、DAT
5	TPGC IO	TPGC IO、TPGC、SWCLK、 TCK、SCK	6	TAUX IO	TAUX IO、AUX、DW、RESET
7	TDI IO	TDI IO、TDI、MOSI	8	TMS IO	TMS IO、SWD IO、TMS
9	TVDD PWR	TVDD PWR	10	GND	GND
11	DGI I2C SCL	DGI I2C SCL	12	TRDAT0	TRDAT0、TRACEDATA(0)
13	DGI I2C SDA	DGI I2C SDA	14	GND	GND
15	DGI VCP TXD	DGI TXD、CICD TXD、VCD TXD	16	TRDAT1	TRDAT1、TRACEDATA(1)
17	DGI VCP RXD	DGI RXD、CICD RXD、VCD RXD	18	GND	GND
19	5V0	予約済み	20	TRDAT2	TRDAT2、TRACEDATA(2)
21	DGI GPIO0	DGI GPIO0、PORT0、TRIG0	22	GND	GND
23	DGI GPIO1	DGI GPIO1、PORT1、TRIG1	24	TRDAT3	TRDAT3、TRACEDATA(3)

ハードウェア仕様

	続き					
ピン	説明	機能	ピン	説明	機能	
25	DGI GPIO2	DGI GPIO2、PORT2、TRIG2	26	GND	GND	
27	DGI GPIO3	DGI GPIO3、PORT3、TRIG3	28	TRCLK	TRCLK、TRACECLK、	
29	3V3	予約済み	30	GND	GND	
31	DGI SPI MOSI	DGI SPI MOSI、SPI DATA、 PORT5、TRIG5	32	DGI SPI MISO	DGI SPI MISO、PORT4、TRIG4	
33	DGI SPI nCS	DGI SPI nCS、PORT6、TRIG6	34	DGI SPI SCK	DGI SPI SCK、SPI SCK、 PORT7、TRIG7	
35	UTIL SDA	予約済み	36	UTIL SCL	予約済み	
37	CS-B	Power Monitor	38	CS+ B	Power Monitor	
39	CS- A	Power Monitor	40	CS+ A	Power Monitor	

10.5.2 ターゲットの接続ケーブル

MPLAB ICE 4 インサーキット エミュレータは、40 ピン リボンケーブル アセンブリに接続し、デバイス固有のアダ プタボードを介してレガシー ターゲットに接続するか新しいターゲットに直接接続します。これにより、デバイス に対応した通信を実現します。



表 10-5. ケーブル アセンブリの仕様

仕様	説明
メーカー	Samtec
ケーブル番号	ERCD-020-08.00-TED-TED-3-B
コネクタ番号	ERF8-020-05.0-S-DV-L
アセンブリタイプ	0.8mm Edge Rate ハイスピード同軸ケーブル アセンブリ
アセンブリ長	9.125 インチ/23 cm

10.5.3 アダプタボードとケーブル

レガシー ターゲット接続をサポートするため、MPLAB ICE 4 キット(DV244140)とアダプタパック(AC244140)でア ダプタボードを提供しています。アダプタボードからレガシー ターゲットに接続するケーブルも MPLAB ICE 4 キットとアダプタパックに含まれています。

表 10-6. アダプタボード上のコネクタとケーブル

アダプタボード	コネクタ	ケーブル
ICE4 JTAG アダプタ	20 ピン JTAG	ターゲットボードへ接続
	10 ピンミニ*	2x5 IDC ソケット、両極性対応
MPLAB ICE 4 ICSP アダプタボード +	8-ピン SIL +	ケーブルなし。アダプタボードはターゲット
・ ICE 4 AVR JTAG (10 ピン)	・ 10 ピン	ボード上の対応ソケットに接続します。
・ ICE 4 AVR (10 ピンミニ)	・ 10 ピンミニ	
・ ICE 4 AVR (6 ピン)	・ 6ピン	
・ ICE 4 AVR (6 ピンミニ)	・ 6ピンミニ	

ハードウェア仕様

続き				
アダプタボード	コネクタ	ケーブル		
MPLAB ICE 4 ICSP アダプタボード	8 ピン RJ-45	6" 6 ピン モジュラケーブル、6" 8 ピン モ ジュラケーブル		
	8 ピン SIL	6"インライン ケーブル		
MPLAB ICE 4 Cortex-M トレース アダプタ ボード	20 ピンミニ	3" 20 ピンカスタム 2 列ソケット アセンブリ		
MPLAB ICE 4 PIC32 トレース アダプタ ボード	14 ピン DIL	10" 14 ピン 2 列ソケット IDC アセンブリ		
	9ピントレース	N/A		

* キットに含まれる 10 ピン JTAG/SWD ケーブル(P/N CAB1013)には、ロック付き嵌合シュラウドへの取り付け を妨げるストレイン リリーフが付いている場合があります。ケーブルの使用手順は以下の通りです。

- 1. ケーブルのストレイン リリーフを取り外し、コネクタ本体の余分なプラスチックを削り取る
- ボードとシュラウドのない嵌合コネクタにケーブルを挿入するか、シュラウドの端のプラスチックを取り 除く
- 3. デモボードを使う場合、代替 ARM デバッグアダプタとケーブル(20 ピン SWD トレース、20 ピン JTAG/ARM)を使う

新しいケーブルを入手するには、Microchip 社のサポートに問い合わせるか、Samtec FFSD-05-D-06.00-01-N または Digi-key SAM8218-ND を購入してください。

関連リンク

3.3.2. アダプタボード

10.5.4 電流検出モジュール

電源監視で使う電流検出モジュールについて説明します。

電流検出モジュールの MPLAB ICE 4 アナログ フロントエンドには、Channel A と Channel B と呼ばれる 2 つの チャンネルがあります。どちらのチャンネルも同様のメカニズムで動作しますが、2 つのチャンネルは対称では なく、それぞれのチャンネルの用途は異なります。

どちらのチャンネルも MCP3464 の独立した ADC チャンネルに接続します。この ADC モジュールでは、どちら のチャンネルでも最大サンプリング レート 1 kHz(1000 sps)で+ve の電流計測のみが可能です。

図 10-4. MPLAB ICE 4 上の電流検出コネクタ



表 10-7. チャンネルの仕様

チャンネルの電流と電圧	分解能	フルスケール
Channel A - 電流	0.6086 uA/ステップ	20 mA*
Channel A - 電圧	0.2087 mV/ステップ	6.8 V
Channel B - 電流	3.043 uA/ステップ	100mA

続き				
チャンネルの電流と電圧	分解能	フルスケール		
Channel B - 電圧	0.2087 mV/ステップ	6.8 V		
* アセンブリ#10-10203-R5 を実装したユニットでは Channel A は最大 20 mA に制限されます。Channel A の電 流レンジ全体を 1 A に拡大するには、CS+ A 端子と CS- A 端子の間に外部シャント抵抗(0.01Ω)を配置します。				

MPLAB Data Visualizer を使ってレンジ計測を行う場合、Channel B を代わりに使う事ができます。

10.6 リカバリの仕様

MPLAB ICE 4 ユニットをリカバリモードにする (デバイスリセットとフラッシュ消去を実行する)には、ユニット 底面の穴からペーパークリップ等を使ってハードウェア リセットスイッチを作動させます(下図参照)。

Hardware Tool Emergency Boot Firmware Recoveryのソフトウェアを使って、本ユニットをリカバリモードにする 事もできます。

図 10-5. リカバリスイッチの場所



10.7 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、デバイスとアプリケーションの要件を満たす電源を供給する必要があります。

デバイス データシートの電気的特性に記載している「絶対最大定格」を超える条件は、デバイスに恒久的な損傷を 生じる可能性があります。これはストレス定格です。本書の動作表に示す条件外でのデバイス運用は想定していま せん。絶対最大定格条件を超えて長期間曝露させるとデバイスの信頼性に影響を及ぼす可能性があります。

デバッガはターゲットの電圧を検出します。VDD_TGT には 182 kΩの負荷を接続しています。

デバッガとターゲット間の通信方法によっては、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。 詳細は以下のセクションを参照してください。

- ターゲットの接続回路
- デバッガの正常動作を妨げる回路

11. 改訂履歴

11.1 リビジョンA (2022 年 1 月) 本書の初版のリリース。

12. サポート

サポートの問題については以下のセクションを参照してください。

12.1 保証登録

ツールをオンラインで登録するには www.microchip.com/mysoftware にアクセスします。myMicrochip アカウントを お持ちでない場合、このリンクでアカウントを登録できます。アカウントを既にお持ちの場合、サインインして [Register Hardware Tool]をクリックします。

ツールをオンラインでご登録頂いたお客様には新製品情報をお届けします。ソフトウェアのマイナーリリースは弊 社ウェブサイトで提供しております。

12.2 myMicrochip 変更通知サービス

Microchip 社の変更通知サービスは、お客様に Microchip 社製品の最新情報をお届けする配信サービスです。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。

www.microchip.com/pcn からサービスに登録し、変更通知の配信をご希望になる製品カテゴリをお選びください。 よく寄せられる質問(FAQ)と登録の詳細はウェブページに掲載されています。

Microchip 社ウェブサイト

Microchip 社はウェブサイト(www.microchip.com)を通してオンライン サポートを提供しています。当ウェブサイト では、お客様に役立つ情報やファイルを簡単に見つけ出せます。以下を含む各種の情報をご覧になれます。

- 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
- 技術サポート FAQ(よく寄せられる質問)、技術サポートのご依頼、オンライン ディスカッション グループ、 Microchip 社のデザイン パートナー プログラムおよびメンバーリスト
- **ご注文とお問い合わせ** 製品セレクタと注文ガイド、最新プレスリリース、セミナー/イベントの一覧、お問い 合わせ先(営業所/正規代理店)の一覧

製品変更通知サービス

Microchip 社の製品変更通知サービスは、お客様に Microchip 社製品の最新情報をお届けする配信サービスです。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、リビジョン、エラッタ情報をいち早くメールにて お知らせします。

http://www.microchip.com/pcn にアクセスし、登録手続きをしてください。

お客様サポート

Microchip 社製品をお使いのお客様は、以下のチャンネルからサポートをご利用頂けます。

- 正規代理店
- 技術サポート

サポートは販売代理店にお問い合わせください。各地の営業所もご利用になれます。本書の最後のページに各国の 営業所の一覧を記載しています。

技術サポートは以下のウェブページからもご利用頂けます。 www.microchip.com/support

製品識別システム

ご注文や製品の価格、納期につきましては弊社または正規代理店にお問い合わせください。

_<u>製品番号</u> [X]⁽¹⁾ - X /XX XXX ↓ ↓ ↓ ↓ ↓ デパイス テープ& 温度レンジ パッケージ パターン リール オプション

デバイス:	PIC16F18313、PIC16LF18313、PIC16F18323、PIC16LF18323	
テープ&リール オプション:	空白	= 標準梱包(チューブまたはトレイ)
	Т	= テープ&リール(1)
温度レンジ:	1	=-40~+85 ℃ (産業用温度レンジ)
	E	= -40~+125 ℃ (拡張温度レンジ)
パッケージ: ⁽²⁾	JQ	= UQFN
	Р	= PDIP
	ST	= TSSOP
	SL	= SOIC-14
	SN	= SOIC-8
	RF	= UDFN
パターン:	QTP、SQTP、その他のコード等(または空白)	

例:

- PIC16LF18313- I/P 産業用温度レンジ、PDIP パッケージ
- PIC16F18313- E/SS 拡張温度レンジ、SSOP パッケージ

Note:

- テープ&リールの識別情報は、カタログの製品番号説明に記載しています。これは製品の注文時に使う識別 情報であり、デバイスのパッケージには印刷していません。テープ&リールが選択できるパッケージの在庫/ 供給状況は、Microchip 社正規代理店にお問い合わせください。
- 2. 小型パッケージオプションがご利用になれる場合があります。小型パッケージについては www.microchip.com/packaging をご覧になるか、弊社正規代理店までお問い合わせください。

Microchip 社のデバイスコード保護機能

Microchip 社製品のコード保護機能について以下の点にご注意ください。

- Microchip社製品は、該当するMicrochip 社データシートに記載の仕様を満たしています。
- Microchip社では、通常の条件ならびに動作仕様書の仕様に従って使った場合、Microchip 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。
- Microchip社はその知的財産権を重視し、積極的に保護しています。Microchip 社製品のコード保護機能の侵害は 固く禁じられており、デジタル ミレニアム著作権法に違反します。
- Microchip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、Microchip 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化しています。Microchip 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。

法律上の注意点

本書および本書に記載されている情報は、Microchip 社製品を設計、テスト、お客様のアプリケーションと統合する 目的を含め、Microchip 社製品に対してのみ使う事ができます。それ以外の方法でこの情報を使う事はこれらの条項 に違反します。デバイス アプリケーションの情報は、ユーザの便宜のためにのみ提供されるものであり、更新に よって変更となる事があります。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。 その他のサポートはMicrochip 社正規代理店にお問い合わせ頂くか、https://www.microchip.com/en-us/support/ design-help/client-supportservicesをご覧ください。

Microchip 社は本書の情報を「現状のまま」で提供しています。Microchip 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、非侵害性、商品性、特定目的への適合性の暗黙的保 証、または状態、品質、性能に関する保証をはじめとするいかなる類の表明も保証も行いません。

いかなる場合もMicrochip 社は、本情報またはその使用に関連する間接的、特殊的、懲罰的、偶発的または必然的損 失、損害、費用、経費のいかんにかかわらず、またMicrochip 社がそのような損害が生じる可能性について報告を受 けていた場合あるいは損害が予測可能であった場合でも、一切の責任を負いません。法律で認められる最大限の範囲 を適用しようとも、本情報またはその使用に関連する一切の申し立てに対するMicrochip 社の責任限度額は、使用者 が当該情報に関連してMicrochip 社に直接支払った額を超えません。

Microchip 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Microchip社の製品を使う事は 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、 Microchip 社は擁護され、免責され、損害をうけない事に同意するものとします。特に明記しない場合、暗黙的ある いは明示的を問わず、Microchip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

Microchip 社の名称とロゴ、Microchip ロゴ、Adaptec、AVR、AVRロゴ、AVR Freaks、BesTime、BitCloud、 CryptoMemory、CryptoRF、dsPIC、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、 maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemi ロゴ、MOST、MOST ロゴ、MPLAB、 OptoLyzer、PIC、picoPower、PICSTART、PIC32 ロゴ、PolarFire、Prochip Designer、QTouch、SAM-BA、 SenGenuity、SpyNIC、SST、SST ロゴ、SuperFlash、Symmetricom、SyncServer、Tachyon、TimeSource、tinyAVR、 UNI/O、Vectron、XMEGA は米国とその他の国におけるMicrochip TechnologyIncorporated の登録商標です。

AgileSwitch、APT、ClockWorks、The Embedded Control SolutionsCompany、EtherSynch、Flashtec、Hyper Speed Control、HyperLightLoad、Libero、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、 ProASIC Plus、ProASIC Plus ロゴ、Quiet-Wire、SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、 TimePictra、TimeProvider、TrueTime、ZL は米国におけるMicrochip Technology Incorporated の登録商標です。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、Augmented Switching、BlueSky、BodyCom、Clockstudio、CodeGuard、CryptoAuthentication、CryptoAutomotive、 CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、 Espresso T1S、EtherGREEN、GridTime、IdealBridge、In-Circuit Serial Programming、ICSP、INICnet、Intelligent Paralleling、IntelliMOS、Inter-Chip Connectivity、JitterBlocker、Knob-on-Display、KoD、maxCrypto、maxView、 memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certified ロゴ、MPLIB、MPLINK、MultiTRAK、NetDetach、 Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PowerSmart、PureSilicon、QMatrix、 REAL ICE、RippleBlocker、RTAX、RTG4、SAM-ICE、Serial Quad I/O、simpleMAP、SimpliPHY、SmartBuffer、 SmartHLS、SMART-I.S.、storClad、SQI、SuperSwitcher、SuperSwitcher II、Switchtec、SynchroPHY、 TotalEndurance、Trusted Time、TSHARC、USBCheck、VariSense、VectorBlox、VeriPHY、ViewSpan、 WiperLock、XpressConnect、ZENAは米国とその他の国におけるMicrochip Technology Incorporated の商標です。

SQTP は米国におけるMicrochip Technology Incorporated のサービスマークです。

Adaptec ロゴ、Frequency on Demand、Silicon Storage Technology、Symmcom はその他の国におけるMicrochip Technology Incorporatedの登録商標です。

GestIC は、その他の国におけるMicrochip Technology Germany II GmbH & Co. KG (Microchip Technology Incorporated の子会社)の登録商標です。

その他の商標は各社に帰属します。 © 2022, Microchip Technology Incorporated and its subsidiaries. All Rights Reserved. ISBN: 978-1-6683-0234-7

品質管理システム

Microchip社の品質管理システムについてはwww.microchip.com/qualityをご覧ください。



各国の営業所とサービス

南北アメリカ

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

オースティン、TX Tel: 512-257-3370

ポストン Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088

シカゴ Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075

ダラス Addison, TX Tel: 972-818-7423

Fax: 972-818-2924 **デトロイト** Novi, MI

Tel: 248-848-4000 ヒューストン、TX

Tel: 281-894-5983 インディアナポリス Noblesville, IN

Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380

ロサンゼルス

Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800

ローリー、NC Tel: 919-844-7510

ニューヨーク、NY Tel: 631-435-6000 サンノゼ、CA

Tel: 408-735-9110 Tel: 408-436-4270 カナダ・トロント Tel: 905-695-1980 Fax: 905-695-2078 オーストラリア - シドニー Tel: 61-2-9868-6733 中国 - 北京 Tel: 86-10 -8569-7000 中国 - 成都

アジア/太平洋

Tel: 86-28-8665-5511 **中国 - 重慶** Tel: 86-23-8980-9588

中国 - 東莞 Tel: 86-769-8702-9880 **中国 - 広州**

Tel: 86-20-8755-8029 **中国 - 杭州**

Tel: 86-571-8792-8115 中国 - 香港SAR

Tel: 852-2943-5100 **中国 - 南京**

Tel: 86-25-8473-2460 **中国 - 青島**

Tel: 86-532-8502-7355 **中国 - 上海** Tel: 86-21-3326-8000

中国 - 瀋陽 Tel: 86-24-2334-2829

中国 - 深圳 Tel: 86-755-8864-2200 **中国 - 蘇州**

Tel: 86-186-6233-1526

中国 - 武漢 Tel: 86-27-5980-5300 **中国 - 西安**

Tel: 86-29-8833-7252 **中国 - 厦門**

Tel: 86-592-2388138 **中国 - 珠海** Tel: 86-756-3210040 **アジア/太平洋** インド - バンガロール Tel: 91-80-3090-4444 インド - ニューデリー Tel: 91-11-4160-8631 インド - プネ Tel: 91-20-4121-0141

日本 - 大阪 Tel: 81-6-6152-7160 **日本 - 東京** Tel: 81-3-6880-3770 **韓国 - 大邱** Tel: 82-53-744-4301

韓国 - ソウル Tel: 82-2-554-7200 マレーシア - クアラルンプール Tel: 60-3-7651-7906 マレーシア - ペナン Tel: 60-4-227-8870 フィリピン - マニラ Tel: 63-2-634-9065

シンガポール Tel: 65-6334-8870 **台湾 - 新竹** Tel: 886-3-577-8366

台湾 - 高雄 Tel: 886-7-213-7830 **台湾 - 台北**

Tel: 886-2-2508-8600 タイ - バンコク Tel: 66-2-694-1351 ベトナム - ホーチミン Tel: 84-28-5448-2100 Fax: 43-7242-2244-393 デンマーク - コペンハーゲン Tel: 45-4485-5910 Fax: 45-4485-2829 フィンランド - エスポー Tel: 358-9-4520-820 フランス - パリ Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 ドイツ - ガーヒンク Tel: 49-8931-9700 ドイツ - ハーン Tel: 49-2129-3766400 ドイツ - ハイルブロン Tel: 49-7131-72400 ドイツ - カールスルーエ Tel: 49-721-625370 ドイツ - ミュンヘン

オーストリア - ヴェルス

Tel: 43-7242-2244-39

欧州

Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 ドイツ - ローゼンハイム

Tel: 49-8031-354-560 イスラエル - ラーナナ Tel: 972-9-744-7705

イタリア - ミラノ Tel: 39-0331-742611 Fax: 39-0331-466781

イタリア - パドヴァ Tel: 39-049-7625286

オランダ - ドリューネン Tel: 31-416-690399

Fax: 31-416-690340 ノルウェー - トロンハイム Tel: 47-7288-4388 ポーランド - ワルシャワ Tel: 48-22-3325737

ルーマニア - ブカレスト Tel: 40-21-407-87-50

スペイン - マドリッド Tel: 34-91-708-08-90 Fax: 34-91-708-08-91

スウェーデン - ヨーテボリ Tel: 46-31-704-60-40 スウェーデン - ストックホルム Tel: 46-8-5090-4654 イギリス - ウォーキンガム Tel: 44-118-921-5800 Fax: 44-118-921-5820