

CCSLOAD Programmer Control Software

インストールレーション

1. CCSLOAD を CD、又は、ダウンロードしたものからインストールして下さい。:

<http://www.ccsinfo.com/downloads.php> から CCSLOAD (Windows Version)

又は、

CD-ROM の ccsload フォルダのなかの setup_ccsload.exe

これらのソフトウェアのデフォルトのディレクトリは CCS のコンパイラと同じディレクトリです。もし、CCS IDE コンパイラをご使用の場合は、共通ファイルが同じバージョンを使用する様に同じディレクトリを使用して下さい。

※**DVD/CD** が古い可能性がありますので、**CCS** より直接実行ファイルをダウンロード/保存してその実行ファイルをデフォルトでインストールされることを強くお勧めします。

CCSLOAD のダウンロード

http://www.ccsinfo.com/downloads/setup_ccsload5.exe

又は、

<http://www.ccsinfo.com/downloads.php>

CCSLOAD Programmer Control Software for Windows の **Download** ボタン

2.次にお使いのデバッガ/プログラマに合わせて USB ドライバをインストールします。

※**DVD/CD** が古い可能性がありますので、**CCS** より直接実行ファイルをダウンロード/保存してその実行ファイルをデフォルトでインストールされることを強くお勧めします。

Windows 用 USB ドライバーのダウンロード

http://www.ccsinfo.com/downloads/setup_usbdrivers.exe

又は、

<http://www.ccsinfo.com/downloads.php>

USB Drivers USB drivers for all CCS programmers/debuggers の **Download** ボタン

※**USB** ドライバをインストールする前にデバッガ/プログラマは接続しないで下さい。

CCSLOAD のアイコン  をクリックして CCSLOAD を立ち上げ状態を確認しておいて下さい。

ICD-U

キットに含まれるもの:

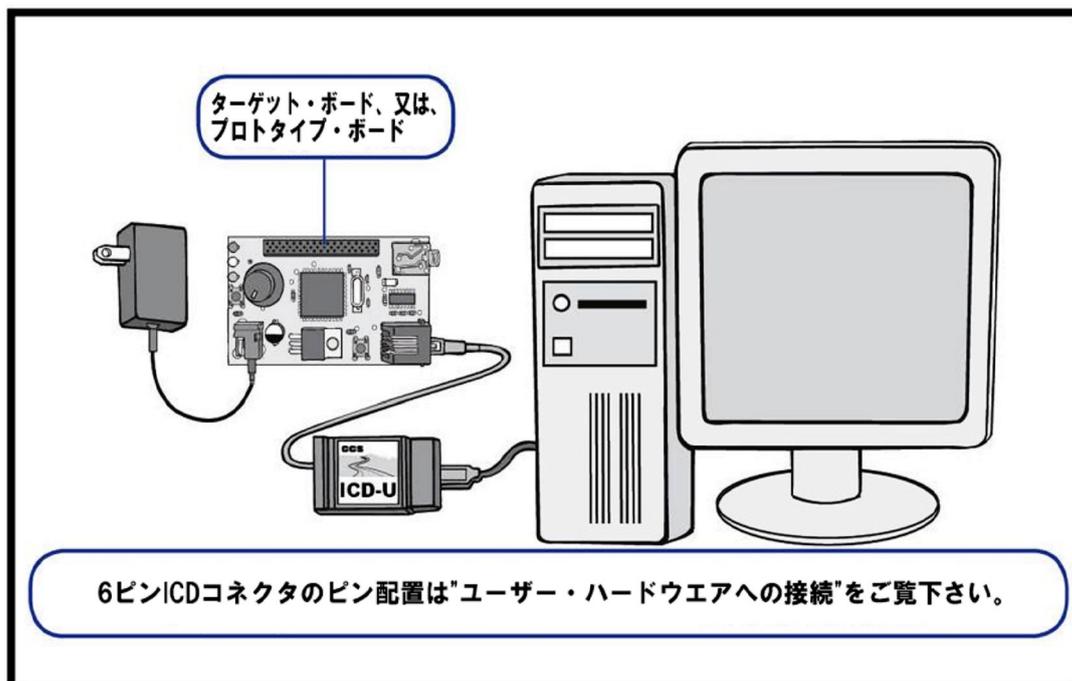
- ICD-U ユニット
- ターゲット・ボード/プロトタイピング・ボードへのモジュラー・ケーブル
- USB⇄PC ケーブル

ハードウェア・インストール:

1. Windows ドライバーと CCSLOAD をインストールして下さい。
CCSLOAD Programmer Control Software for Windows ->
http://www.ccsinfo.com/downloads/setup_ccsload5.exe
USB Drivers USB drivers for all CCS programmers/debuggers ->
http://www.ccsinfo.com/downloads/setup_usbdrivers.exe
2. ICD-U とターゲット/プロトタイピング・ボードにモジュラー・ケーブルを接続して下さい。
3. ターゲット、又は、プロトタイピング・ボードの電源を入れて下さい。外部電源、又は、ICD-U40/64 はジャンパー装着、ICD-U80 は CCSLOAD からセット
4. デスクトップ・アイコンから CCSLOAD.EXE ソフトウェアを起動
ノート: ICD-U40/U64/U80 は USB バスから電源供給されます。ICD-U40/U64 からターゲットへ電源供給したい場合は ICD ユニットにジャンパーを装着して下さい。ICD-U80 はジャンパーは不要、CCSLOAD や IDE コンパイラからデバッガ経由で電源供給することが出来ます。

ICD ファームウェア:

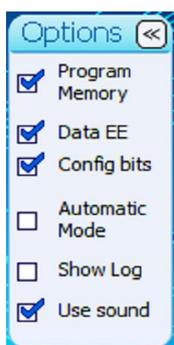
ICD ユニットは最新の CCS ファームウェアで出荷されていますが、CCSLOAD のバージョンアップ等によりファームウェアのアップグレードは Diagnostics[ダイアグノスティック]ウィンドウから Update Firmware をクリックすることでアップデートして下さい。



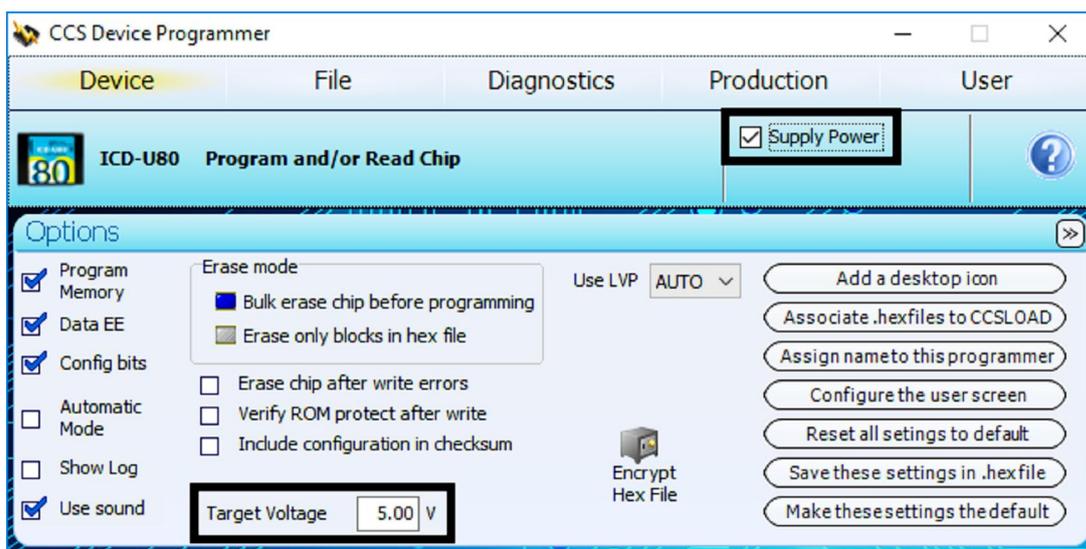
ICD-U64 から 5V 又は、3.3V をターゲットへ供給:
下記のようにジャンパーを装着して下さい:



CCSLOAD を経由して ICD-U80 からターゲットへ 5V 又は、3.3V を供給:
下の画面のように CCSLOAD ウィンドウのトップの"Supply Power"ボックスにチェックを入れます。オプション・パネルを展開して Target Voltage[ターゲットの電圧値](1.5V - 5.5V)を設定します。



ここをクリックと以下の画面が現れます。



LOAD-n-GO

キットに含まれるもの:

- LOAD-n-GO プログラマ・ユニット
- ターゲット・ボード/プロトタイピング・ボードへのモジュラー・ケーブル
- USB⇄PC ケーブル

ハードウェア・インストール:

1. Windows ドライバーと CCSLOAD をインストールして下さい。
CCSLOAD Programmer Control Software for Windows ->
http://www.ccsinfo.com/downloads/setup_ccsload5.exe
USB Drivers USB drivers for all CCS programmers/debuggers ->
http://www.ccsinfo.com/downloads/setup_usbdrivers.exe
2. LOAD-n-GO とターゲット/プロトタイピング・ボードにモジュラー・ケーブルを接続して下さい。
3. デスクトップ・アイコンから CCSLOAD.EXE ソフトウェアを起動

LOAD-n-GO ファームウェア:

LOAD-n-GO ユニットは最新の CCS ファームウェアで出荷されていますが、CCSLOAD のバージョンアップ等によりファームウェアのアップグレードは Diagnostics[ダイアグノスティック]ウィンドウから Update Firmware をクリックすることでアップデートして下さい。

LOAD-n-GO ヘプログラムをダウンロード:

1. 付属の USB ケーブルを使用して LOAD-n-GO ユニートを PC に接続します。
2. CCSLOAD を起動します。画面の右上の領域から、4つのメモリ・バンクのいずれかを選択します。これはプログラムの行き先バンクとして対応します。
3. ファイル・ブラウザを使用して選択したメモリ・バンクに配置するプログラムを選択します。ファイルとターゲット・デバイスが表示されていることを確認します。
4. %Write to Mem+ボタンをクリックします。(各メモリ・バンクは最大 512KB のプログラムを保持することができます。プログラムが大きくなると追加のメモリ・バンクが必要になります)。
5. 最大 4 つのプログラムに対してこれらの手順を繰り返します。ノート:メモリ・バンクをプログラムすると以前に格納された内容が上書きされます。

リモート・プログラミングの使用:

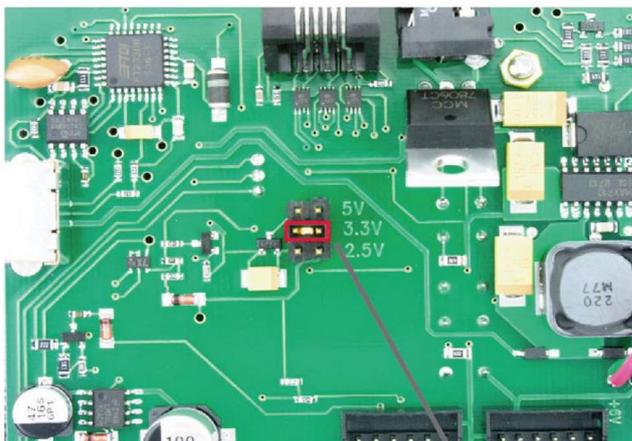
1. 付属のモジュラー・ケーブルを使用してユニットからターゲット・ボードに接続します。
2. %On/Off+ボタンを押して下さい。ユニットに電源供給されたとき緑色に点灯します。
3. %Program Select+ ボタンを使用して目的のメモリ・バンクを選択します。選択されたバンクを示すために 4 つのプログラム LED の 1 つが点灯します。
4. プログラムをするために %Load+ボタンを押します。ステータス LED がプログラムのローディングを示すために点滅します。
5. プログラムが完了すると、LED は点滅を停止し成功すると緑色に変わります。転送が失敗した場合は赤色になります。
6. モジュラー・ケーブルを外します。ターゲット・ボードのプログラムが実行されます。
7. 必要に応じて繰り返して下さい。ターゲットに必要なプログラムが選択されていることを確認して下さい。

パワー・オプション:

- LOAD-n-GO は 4 AA バッテリーを使用出来ます。
- バッテリー電源が供給されている場合、5 分間使用しないと自動シャットオフ機能を使用してバッテリーを節約します。

- 電源を切ってもプログラムは消去されません。
- LOAD-n-GO はシャント・ジャンパを LOAD-n-GO 基板に挿入することによりターゲットボードに 5V、3.3V、又は、2.5V を供給するために使用できます。これらのジャンパーにアクセスするにはケース背面の 4 本のネジを外してケースを開く必要があります。これらのジャンパーの位置は回路基板にマークされています。
 - エンクロジヤの底からカバーは外します。
 - 短絡ジャンパ(例：3M 929950-00)を下の例のように水平位置に取り付けます。

**注意；このジャンパはターゲットに電源供給されていないことを確認してインストールして下さい。



3.3V の所にジャンパを入れた状態

インサーキット・シリアル・プログラミングとデバッグのために使用:

1. USB ケーブル経由で LOAD-n-GO ユニットを PC に接続します。ターゲット・ボードに電源が供給され、そして、モジュラー・ケーブルでユニットに接続されていなければいけません。
2. CCSLOAD を起動し、メイン・ドロップ・ダウン・ボックスから%**CSP+** を選択して下さい。
3. ファイル・ブラウザを使用して選択したメモリ・バンクに配置するプログラムを選択します。ファイルとターゲット・デバイスが表示されていることを確認します。
4. ターゲットのプログラミングを開始するには%**Burn Chip+** ボタンをクリック
5. LOAD-n-GO は CCS C IDE コンパイラを使用してリアルタイム・アドバンスド・デバッグを行う場合、インサーキット・デバッグとしても機能します。

赤色 **LED** の点滅に対するエラー・コード::

2. ベリフィケーション失敗。
3. ターゲットの PIC が検知されない。
4. メモリー・スロットにプログラムするための有効な HEX ファイルがありません。
5. 重大なエラー; CCS のテクニカル・サポートに連絡して下さい。
6. 重大なエラー; CCS のテクニカル・サポートに連絡して下さい。
7. このスロットにユーザ定義の書き込み回数が残っていない場合、プログラミング・コマンドは無視されます。

Mach X

キットに含まれるもの:

- Mach X プログラマ・ユニット
- ターゲット・ボード/プロトタイピング・ボードへのモジュラー・ケーブル
- USB⇄PC ケーブル

ハードウェア・インストール:

1. Windows ドライバーと CCSLOAD をインストールして下さい。
CCSLOAD Programmer Control Software for Windows ->
http://www.ccsinfo.com/downloads/setup_ccsload5.exe
USB Drivers USB drivers for all CCS programmers/debuggers ->
http://www.ccsinfo.com/downloads/setup_usbdrivers.exe
2. Mach X を USB ケーブル経由で PC に接続して下さい。

ZIF ソケット・プログラミング:

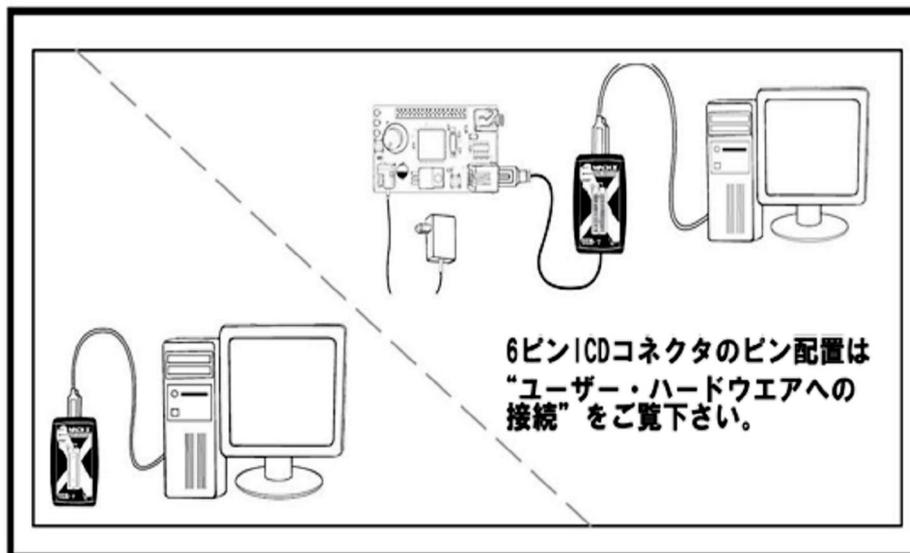
3. ラッチを引いてデバイスを 40 ピン ZIF ソケットにピン 1 をソケットの左上隅に置きます。

インサーキット・プログラミング:

3. モジュラー・ケーブルでターゲット・ボードに接続し、そして、必要な場合は電源供給します。

4. デスクトップ・アイコンから CCSLOAD.EXE ソフトウェアを起動

ノート: Mach X は USB バスからもターゲットに電源供給することが出来ます。CCSLoad ソフトウェアを使ってコンフィギヤして下さい。



Mach X ファームウェア:

Mach X ユニットは最新の CCS ファームウェアで出荷されていますが、CCSLOAD のバージョンアップ等によりファームウェアのアップグレードは Diagnostics[ダイアグノスティック]ウィンドウから Update Firmware をクリックすることでアップデートして下さい。

Prime8

キットに含まれるもの:

- Prime8 プロダクション・プログラマ・ユニット
- 8本のPrime8からターゲット・ボードへのモジュラー・ケーブル
- USB⇄PCケーブル
- ACアダプタ

ハードウェア・インストール:

1. Windows ドライバーと CCSLOAD をインストールして下さい。
CCSLOAD Programmer Control Software for Windows ->
http://www.ccsinfo.com/downloads/setup_ccsload5.exe
USB Drivers USB drivers for all CCS programmers/debuggers ->
http://www.ccsinfo.com/downloads/setup_usbdrivers.exe
2. Prime8 ユニットの USB ケーブル経由で PC に接続して下さい。
3. AC アダプタを Prime8 に接続してから電源へプラグして下さい。

スタンドアローン・モードの使用:

- スタンドアローンモードは Prime8 が PC に接続されていない場合です。
- LCD と押しボタンを使用してユニットを制御します。
- スタンドアローン・モードでは最大 8 つのジョブを保持できます。各ジョブは対象ファイル、シリアル番号と電圧などの構成情報を保持します。ジョブは読み取り専用にすることが出来ます。
- スタンドアローンには手動モードもあります - オンラインで読むことができないジョブ

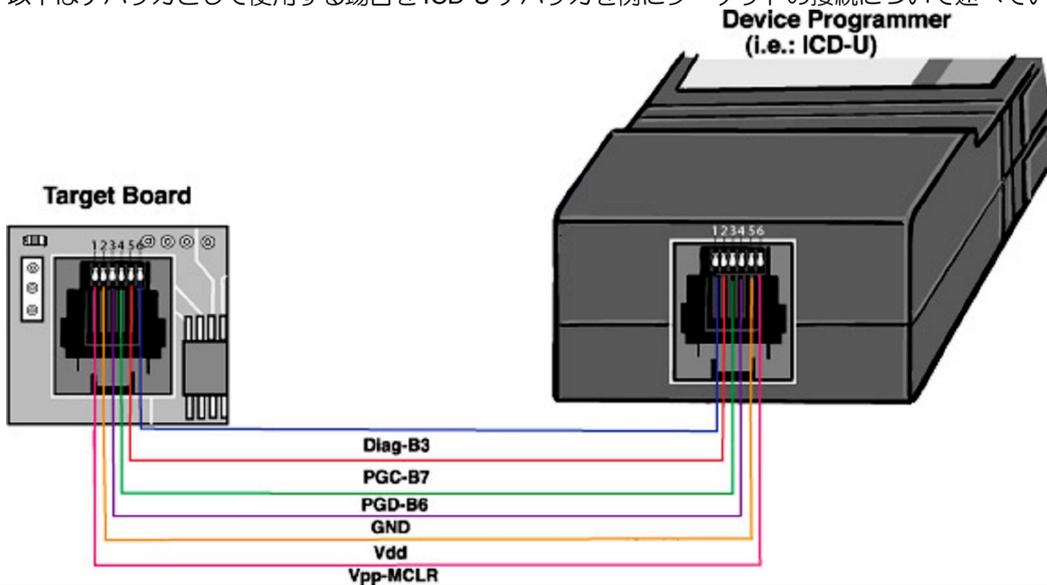
CCS プログラマ・コントロール・ソフトウェア(**CCSLOAD**) モードの使用:

- CCS プログラマ制御ソフトウェア (ccsload.exe) を実行します。Prime8 のプログラミング設定は CCSLoad ソフトウェアの最上部にあります。
- ~~Prime8~~ Memoryq チェック・ボックスは読み取り/書き込みが行われる場所を制御します:
 - このボックスにチェックを入れると、下のラジオ・ボタンで選択された内部メモリ・バンクが読み書きされます。別のラジオ・ボタンを選択するとどのバンクが読み書きされるかが変更されます。メモリ・バンクは 512KB です。プログラムが 512KB より大きい場合は、複数のメモリ・バンクを使用します。バンクに書き込むと前の内容が消去されます。
 - このボックスのチェックを外すと、Prime8 はギャングモードになります。追加のチェック・ボックスとステータス・インジケータが 8 つのデバイス・スロットのそれぞれに表示されます。同時に複数のスロットに書き込むには、必要なすべてのスロットのチェック・ボックスをオンにします。チェック・ボックス 1 は slot 1 を選択し、チェック・ボックス 2 は slot 2、以下同様に slot 8 まで選択します。
- ~~Prime8~~ Memory boxq ボックスがチェックにされていない場合(ギャングモード)の時 ~~Supply Power~~q チェック・ボックスが表示されます。このボックスは Prime8 の電源を制御します:
 - このボックスにチェックマークを付けると、リモート・ターゲットは Prime8 によって電源が供給され個々のターゲットはそれぞれの電源を必要としません。Prime8 から供給される Vdd は、CCSLoad ソフトウェアの ~~Options~~q パネルで設定することが出来ます。
 - このボックスのチェックを外されている時、リモート・ターゲットはそれぞれ電源給電されなければいけません。

ユーザー・ハードウェアの接続

PC の USB ポートをご使用下さい。ハブからは使用しないで下さい。

Mach-X を ZIF ソケット・プログラマとして使用する場合は下記の様に **ZIF Socket** を指定して下さい。Mach-X をデバッガとして使用する場合は、Mach-X から電源供給する場合は **Mach-X supplies power** を選択、外部電源を使用の場合は **Target powered** にチェックを入れて下さい。以下はデバッガとして使用する場合を ICD-U デバッガを例にターゲットの接続について述べています。

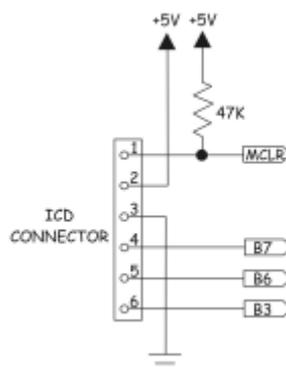
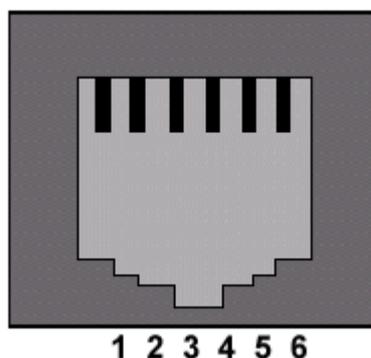


ICD ピン	ターゲット・ピン	ターゲット・ピン
1	6	ターゲットの PIC [®] MCU 上の B3- これはオプションです。CCS IDE コンパイラでの C 言語リアルタイム・デバッグ・モニターのために使用されます。
2	5	(ICSP クロック)ターゲット PIC [®] MCU の PGC(多くのデバイスで B6)
3	4	(ICSP データ) ターゲット PIC [®] MCU の PGD(多くのデバイスで B7)
4	3	グラウンド
5	2	ターゲット Vdd +5V ターゲットから ICD. ICD はこのピン(5V)から電源供給され、そして、ICD は信号(3V-5V)をプルアップするために使用します。
6	1	MCLR - ターゲット PIC [®] MCU の MCLR/VPP ピンに接続され、そして、47K 抵抗でターゲット・ボード上で+5V にプルアップされます。ICD はチップのプログラミング中これを 13V で駆動します。MCLR ではコンデンザーは使用しないで下さい。プログラミングに問題を起こすことがあります。

ICD のソケットはプログラマ・ユニット上のピンアウトです。ターゲットのソケットはターゲット/プロトタイプ・ボードのピンアウト; このピンアウトは CCS から供給されている RJ12 ケーブルを使うことで対応されます。

ノート:

1. Mach X, ICD-U80 と Prime8 はターゲットに電源供給することが出来ます。勿論、ターゲットに外部電源を供給することも出来ます。Debug configure オプションで正しいオプションを選択して下さい。
2. プログラムと/又は、イン-サーキット・デバッグでは 2 つの I/O ピン(PGC,PGD)はイン-サーキットのためにリザーブして下さい。もし、デバッグされない場合は、それらのピンはターゲット回路のために使用出来ますが、しかし、プログラミング中にターゲット回路がハイ・インピーダンスを持っていることに注意して下さい。
3. CCS のプログラムは低電圧プログラミング・モードを使用しません。C プログラムのヒューズ設定は NOLVP にセットされてなければいけません。
4. ターゲット・チップのオシレータはデバッグとして動作するために動作(発信)していなければいけません。プログラミングはオシレータなしでおこなうことが出来ます。
5. B3 ピンはオプションです。そして、プログラミングには使用しないで下さい。しかし、デバッグのモニタ機能は B3 を使用します。モニターを使わないでプログラムとデバッグ、そして、B3 をターゲット・ハードウェアに割り当てることは可能です。この場合は B3 をユニットのコネクターへ接続しないで下さい。もし、モニター機能を使用しない場合は、ユーザー・ストリームを Configure Tab でディスエーブル[Configure Tab Enable Userstream False]にすることが出来ます。そして、1-6 の接続は問題にはなりません。古いバージョンのソフトウェアでは Configure Tab を使ってユーザー・ストリーム[Enable Userstream]をディスエーブル[False にする]出来ませんので、ピンは常にハイにプルアップする必要があります。B3 はすべての PIC[®] MCU ピンをこの機能のために使うことが推奨されています。#use rs232 を debugger にすると monitor port に表示するのに初期値では B3 を用いる様設定されています。詳しくはコンパイラのヘルプ・ファイルを見てください。
6. MCLR ピンはプログラミングとデバッグに使用されますが、プログラミング中の電圧は 13V です。ターゲット側の MCLR 回路のプルアップ抵抗は 47k オームの範囲を推奨します。5V への 47K の抵抗は 13V に対するアイソレーションとしては十分です。しかし、もし、何かが MCLR ピンに接続されている場合は、13V が障害や干渉をしないように気をつけて下さい。
7. 重要：各ユニットからのターゲット・ケーブルはピン番号が反転されますので MCLR 信号は各ユニットの 6 番ピンであり、それはターゲット・ピンの 1 番ピンに接続されます。
8. ケーブルの長さは 30cm 以下でなければいけません。



ある種のチップは PGC と PGD のために B6, B7 を使用しません。 例えば：

チップ	B6 の代わり	B7 の代わり
PIC12F629	GP1	GP0
PIC12F675	GP1	GP0
PIC12F683	GP1	GP0
PIC16F630	RA1	RA0
PIC16F676	RA1	RA0

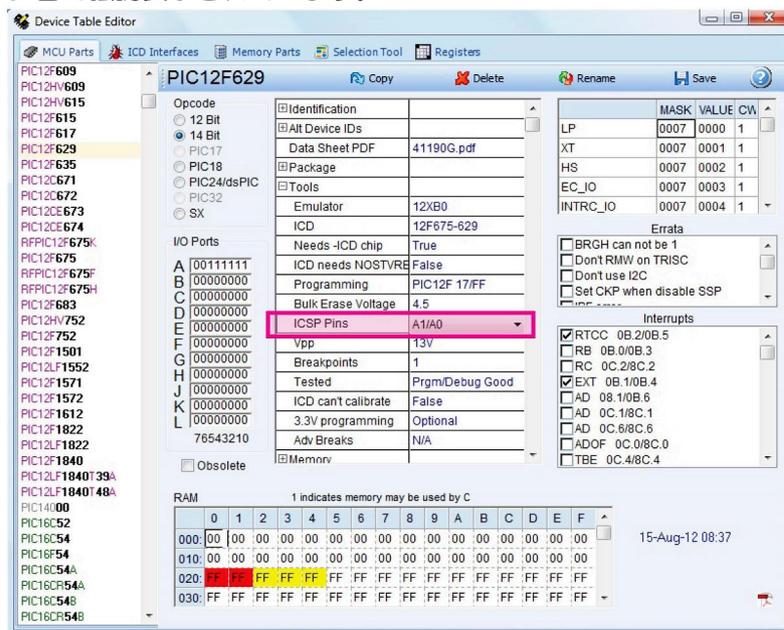
PIC16F684	RA1	RA0
PIC16F688	RA1	RA0

下記のチップは標準のチップでデバッグ機能を持っていないチップです。デバッグのためには特定の ICD チップが必要です。ICD チップは標準のチップより多い数のピンを持っています。
※下記は参考です。正確にはデータシートで確認して下さい。

PIC10F200/2/4/6	PIC16F505-ICD	PIC16F627A/628A	PIC16F648A-ICD
PIC10F220/2	PIC16F506-ICD	PIC16F630	PIC16F676-ICD
PIC12F508/509	PIC16F505-ICD	PIC16F631	PIC16F690-ICD
PIC12F510	PIC16F506-ICD	PIC16F636	PIC16F636-ICD
PIC12F519	PIC16F526-ICD	PIC16F639	PIC16F636-ICD
PIC12F609/HV609	PIC16F616-ICD	PIC16F648A	PIC16F648A-ICD
PIC12F615/HV615	PIC16F616-ICD	PIC16F676	PIC16F676-ICD
PIC12F629	PIC12F675-ICD	PIC16F677	PIC16F690-ICD
PIC12F635	PIC16F636-ICD	PIC16F684	PIC16F684-ICD
PIC12F675	PIC12F675-ICD	PIC16F685/687	PIC16F690-ICD
PIC12F683	PIC12F683-ICD	PIC16F688	PIC16F688-ICD
PIC16F505	PIC16F505-ICD	PIC16F689/690	PIC16F690-ICD
PIC16F506	PIC16F506-ICD	PIC16F716	PIC16F716-ICD
PIC16F526	PIC16F526-ICD	PIC16F785/HV785	PIC16F785-ICD
PIC16F610/HV610	PIC16F616-ICD	PIC18F13K50	PIC18F14K50-ICE
PIC16F616/HV616	PIC16F616-ICD	PIC18F14K50	PIC18F14K50-ICE
		PIC18LF13K50	PIC18LF14K50-ICE

CCS コンパイラ・デバイス・データベース[Compiler Device Database]:

PICC インストール・ディレクトリにある実行可能ファイル "chipedit.exe"を開きます。左列リストでサポートされている全てのデバイスからターゲット・デバイスを選択します。ICSP Pins"を見つけるためには、データベースエントリの"Tools"セクションを展開して下さい。下にピンク色で強調表示されています。



ノート：記載されているピンは ICSPCLK/ICSPDAT(PGC/PGD)の形式です。デバイスにこれらのピンの複数のセットがある場合、ペアは分離されます。

ノート：デバイスによっては GPx(GP0, GP1 等)のピン名を用語規則でデータシートで使用するものがあります。このような場合、CCS コンパイラは A0, A1 等(GP0 = A0、GP1 = A1 ...)を使用します。

データシートの使用

デバイスのデータシートは常に正しいピンを記載していますので、最新のデータシート、エラッタ等で確認されること強くお勧めします。

デバッグ機能のないデバイス:

標準バージョンのデバッグ機能を持たないデバイスもあります。デバッグには特定の ICD バージョンのチップが必要です。マイクロチップ社はこれらのチップを備えたアダプタを販売しています。*デバッガのデバイス・サポートで Debug 欄に X² とあるデバイスがそれに相当します。

Windows ユーザー・インターフェース

CCSLOAD ソフトウェアには 5 つの主要なユーザー・インターフェースがあります:

- **Devices** (Alt-M)
- **File** (Alt-F)
- **Diagnostics** (Alt-D)
- **Production** (Alt-P)
- **User** (Alt-U)

これらは画面のトップのインターフェース名をクリックすることで選択出来ます。

メニューの左上にソフトウェアによって現在制御されているデバイス・プログラムの画像と名前が表示されます。

画面の下のステータス・バーには 5 つのセクションを持っています:

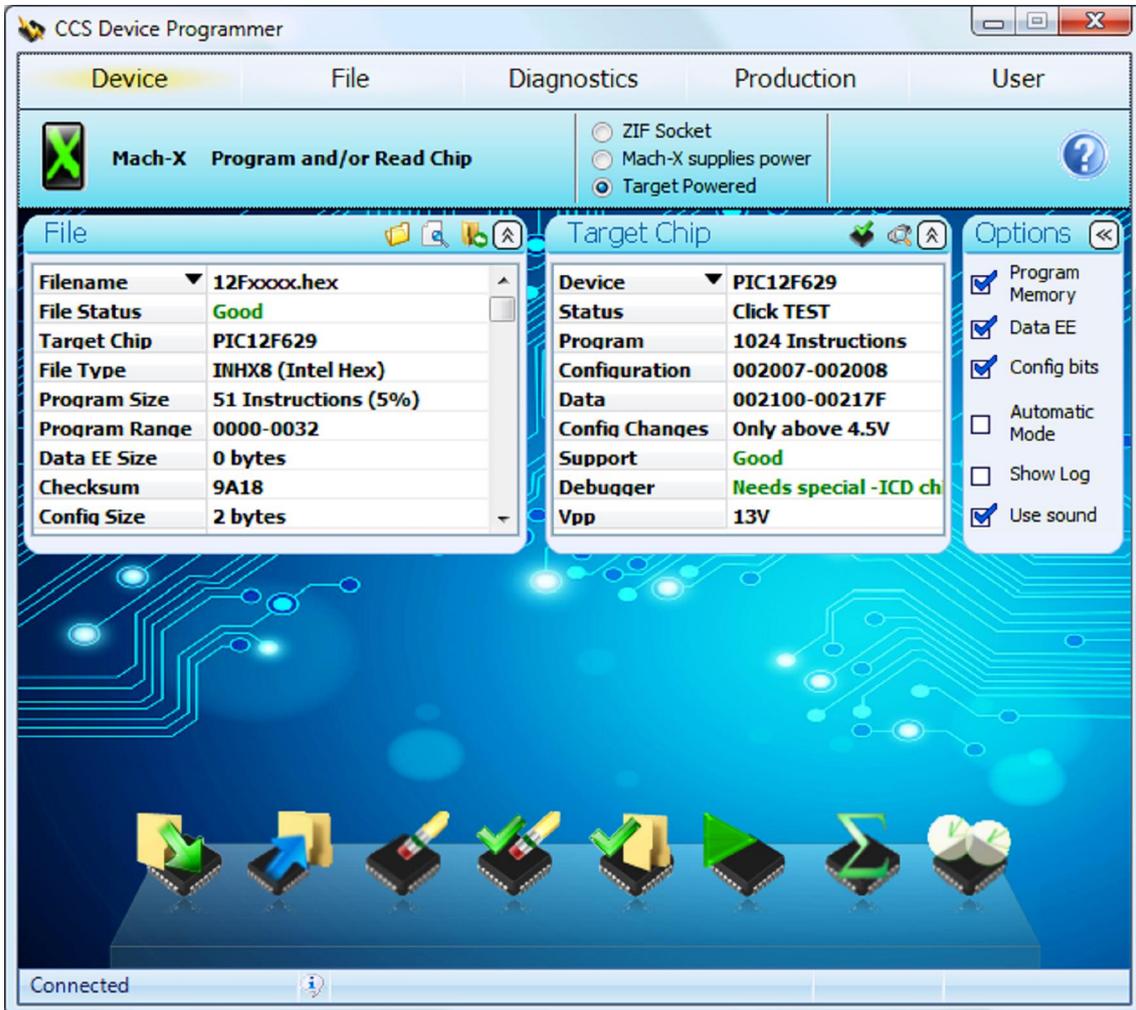
- 最後の操作の状態
- 詳細情報アイコン。このアイコンが表示された場合にこのアイコンをクリックすると最後の操作に関する詳細情報が表示されます。白いアイコンはエラーの説明を示し青色のアイコンは検証結果を示します。
- 中央の領域は現在の操作の進行状況を示すために使用されます。
- 最も右側の 2 つのパネルはパスと失敗の数を示します。これらのカウントを変更するにはテキストをクリックします。



右側上部の X をクリックすることでソフトウェアを終了します。

Device ウィンドウ

ファイルとチップの選択、そして、プログラミング、イレース、読み込み等のための基本となるコントロール・ウィンドウです。



これは主要な制御ウィンドウです。

- **File** - このパネルはターゲットにプログラムするファイルを選択したり、又は、ターゲットの読み取りを行うときに書き込むファイルを選択できます。
 - Load File、又は、Browse Directory(最初の2つのアイコン)を使用してPCのファイルを参照できます。
 - Load Previous File(最後のアイコン)、又は、パネルのFilenameをクリックしますと以前に使用したファイルのリストが表示されます。ファイルが選択された後、ファイルに付いての特定の情報がテーブル内にリストされます。



ファイルのロード



ディレクトリをブラウズ



前のファイルをロード

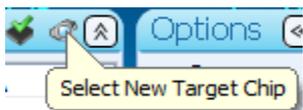
- Target Chip Panel** . このパネルはターゲット・チップが表示されます。ファイルが CCS コンパイラによって作成された場合、又は、ファイルにチップ名が指定されていまずとファイルが選択されるとターゲットが自動的に更新されます。"Select New Target Chip"アイコンをクリックするか、又は、パネルの"Device"をクリックすることでチップを手動で選択します。ターゲットが選択された時、チップに関する特定の情報がテーブルにリストされます。"Test Target Chip"アイコン(又は、Ctrl+T)を使用してターゲットが正しく接続されていることを確認します。



ターゲット・チップをテスト -> Click しますと以下の様に画面が変わりターゲット・チップの Device ID や Revision を見ることが出来ます。

Target Chip	
Device	PIC16F877A
Status	Click TEST
Program	8192 Instructions
Configuration	002007-002008
Data	002100-0021FF
Config Changes	Only above 4.5V
Support	Good
Debugger	Good
Vpp	13V

Target Chip	
Device	PIC16F877A
Status	Good
Device ID	0E20
Revision	07
Program	8192 Instructions
Configuration	002007-002008
Data	002100-0021FF
Config Changes	Only above 4.5V
Support	Good



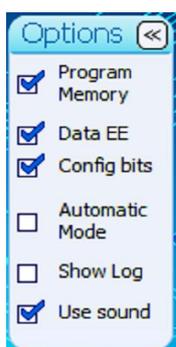
新しいターゲット・チップを選択

- Programmer Specific Area** - デバイス・プログラマに応じてメニュー・バーに設定するオプションがあります。例えば、LOAD-n-GO、又は、Mach X の電源オプションでメモリ・バンク番号を設定します。
- Message Area** - File と Target Chip パネルの下の領域でポップアップ・エラーメッセージと確認メッセージが表示されます。メッセージは短時間だけ表示されます。メッセージによってはクリックすると詳細情報が表示されます。"More Information"アイコンをクリックすると同じ情報を見ることが出来ます。

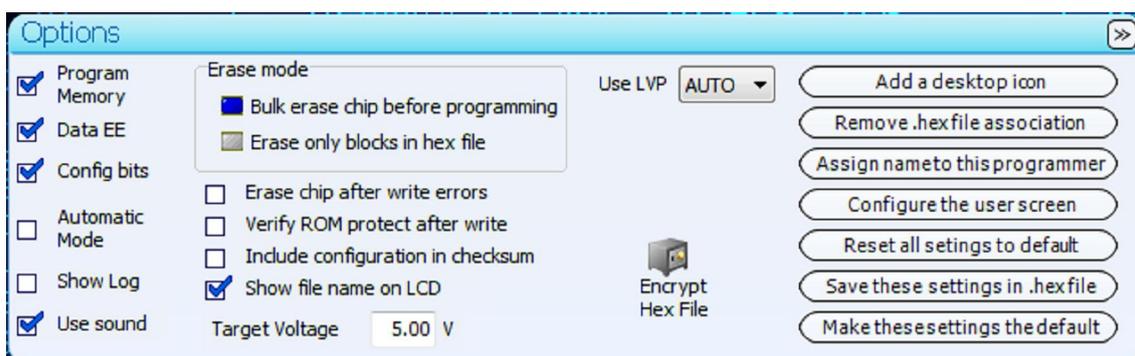


- Operational Buttons** . これらの機能の操作は Options パネルの設定の影響を受けることに注意して下さい。例えば、Options パネルでプログラム・メモリのみが選択されている場合、"Erase target Chip"はプログラム・メモリのみを消去します。

 <p>Write to Chip</p>	<p>ターゲット・チップのプログラム</p>	<p>F => Chip (Ctrl-W)</p>
 <p>Read from Chip</p>	<p>チップからファイルに読み込み</p>	<p>Chip+ => File (Ctrl-R)</p>
 <p>Erase Chip</p>	<p>ターゲット・チップをイレース</p>	<p>(Ctrl-E)</p>
 <p>Verify Chip Blank</p>	<p>チップがイレースされているかをベリファイ</p>	<p>(Ctrl-B)</p>
 <p>Verify Chip=File</p>	<p>チップとファイルがマッチしているかをベリファイ</p>	<p>(Ctrl-V)</p>
 <p>Run Target Program</p>	<p>ターゲット・チップ上でプログラムの実行</p>	<p>(Ctrl-H)</p>
 <p>Stop Target Program</p>	<p>ターゲット・チップ上でプログラムの実行を停止</p>	
 <p>Checksum Chip</p>	<p>チップをファイルに読み込み読み取ったものからチェックサムを計算</p>	
<p>Calibrate</p>	<p>プログラム・メモリに校正值を持ったチップの場合のみに現れます。 この機能はカリブレーション定数を再計算しメモリに保存します。ノート：これらのチップに対してCCSLOAD ソフトウェアは書き込み又は、イレースでこの定数を変えません。</p>	<p>(Ctrl+C)</p>



ここをクリックと以下の画面が現れます。



Options Panel . 各種 CCSLOAD とプログラミング・セッティングをセット。"More Options" アイコン("<<" icon)は上記の様にパネルを展開します。

Settings Panel - ターゲット・チップへの読み込み/書き込み/イレース操作に影響する各種オプションをセット。これらのセッティングの初期値は下記の順番でソースからのものとなります。:

1. CCSLOAD デフォルト値
2. .ini ファイルに保存された最後の設定
3. コマンド・ラインで指定された設定
4. .HEX ファイルで指定された設定
5. セッティング画面でユーザーにより変更された設定

- 例えば、ヘキサ・ファイルの設定はコマンド・ラインでの設定より優先されます。
- ヘキサ・ファイルへの新しい設定を保存するには “Save these settings in hex file” をクリックして下さい。
- .ini ファイルのデフォルトに対して新しい設定を保存するには “Make the default” をクリックして下さい。

設定:	選択された場合:
Program Memory	プログラム・メモリは全ての操作に含まれます。
Data EE	データ・メモリは全ての操作に含まれます。
Config Bits	コンフィギュレーション・ヒューズは操作に含まれます。
Bulk erase chip before programming	高速のチップ・イレースは毎回書き込み操作時の前に全てのメモリをクリアするために使用されます。
Erase only blocks in hex file	hex ファイル内のメモリのみ選択された時に消去されます。これはブートローダやコンフィギュレーションデータを ROM に保存するために使用されます。
Erase chip after write	プログラミング中にエラーが発生した場合、チップは消去

errors	されます。
Verify ROM protect after write	余分なステップはチップのプロテクトされた領域を読み込み返すことができないことを確かめるために実行されます。言い換えれば、プロテクト機能はプログラムが確実に読まれないように安全を確認するためにテストされています。確認メッセージはチップがこのテストに合格したことを示すために現れます。
Include configuration in checksum	画面上のチェックサムはコンフィギュレーション・メモリーを含んでいます。これはチェックサムにビットが含まれていないのを確信したところで Microchip のアルゴリズムを使用して行われます。
Use sound	パス/フェイル・ノイズは各操作の後に PC のスピーカーを通して聞こえます。
Target Voltage	ターゲット Vdd を供給できるプログラマにのみ適用され、供給ターゲット Vdd の選択が行われます。ここで設定される電圧が使用されます。
Show file name on LCD	このオプションは Prime8 プログラマにのみ表示されます。これはメモリ・スロットをプログラマがそのファイルのファイル名が画面、又は、デフォルトのテキストに表示されるかどうかを設定します。
Automatic Mode	このオプションは他のボタンを押す必要をなくすことで同じタイプの複数のチップを連続してより迅速にプログラミングすることが出来ます。ターゲットチップをプログラマに接続するためのプロンプトが表示されます。CCSLOAD がターゲットを検出した時、自動的にプログラミングが開始されます。完了したら、プログラムされたターゲットの接続を切断し別のターゲットを接続するよう促すメッセージが表示されます。このオプションが無効にされるまでこのサイクルを繰り返します。
Show Log	最後に実行した操作のログエントリを示すパネルを表示

セッティング画面に追加して次のボタンがコンフィギュレーション操作のためにあります。:

Add a desktop icon	新しいアイコンをデスクトップに置きます。アイコンは特定のプログラマに接続するか、又は、CCSLOAD プログラムで特定のウィンドウを開くように設定できます。
Assign name to this programmer	各プログラマはユニークなシリアル番号を持っています。このオプションは同じ PC に複数のプログラマが接続されて使用される時にそれぞれのプログラマに名前を与えることが出来ます。
Associate/Un-associate .HEX files	関連付けられた時、ウィンドウズのディレクトリ内で HEX ファイル上でダブル・クリックすることでこのソフトウェアを開始することが出来ます。
Configure user screen	このボタンをクリックするとユーザー画面が現れます。そこでボタンが持つべき機能を指定するためにどのボタンの上でもクリックすること

	が出来ます。また、イメージを変更するためにそのイメージ上でクリックすることも出来ます。
Reset to default settings	これは全てのセッティングを CCS のデフォルト値にセットします。

File ウィンドウ

このオプションは .HEX ファイルを見る事と編集が出来ます。変更された後は SAVE ボタンを使って同じ .HEX ファイルに保存したり SAVE AS ボタンで新しいファイルとして保存することが出来ます。HEX ファイルの 4 つのセクションが別々に編集出来ます。

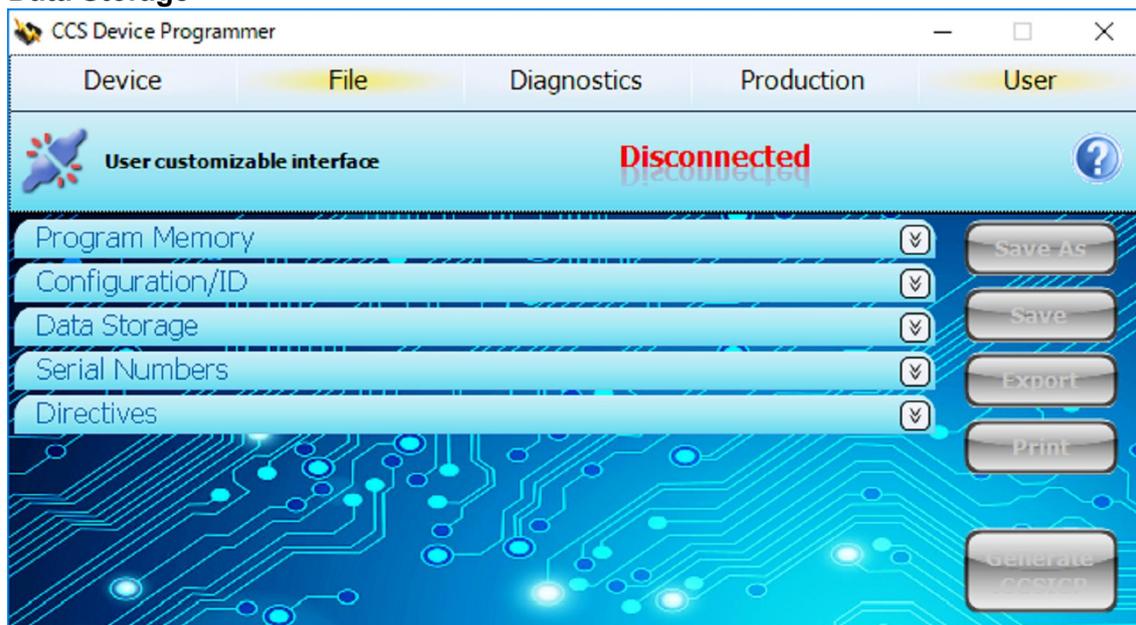
Program Memory

Configuration

Serial Numbers

Directives

Data Storage



Program Memory[プログラム・メモリ]

アドレス・セッティング - ヘキサ・ファイル内のアドレスは8ビット・アドレスですが、殆どのPIC® デバイス・チップは16ビット・アドレスを使用します。ビューはアドレス表示を何れの方法にもセットすることが出来ます。デフォルトでは選択されたパーツに対するデータシートにあるアドレス方法で表示されます。

グルーピング - ヘキサ・ファイルでは各バイトは別個に(int8)扱われますが、プログラム・メモリは殆どのチップがワード(又は、ダブル・ワード)で構成されます。

この選択はバイトの分類を指定することが出来ます。デフォルトは選択されたチップによるデフォルトです。バイト・オーダーはビューでデータシートに合致するように変更されます。例えば、int16 グルーピングが選択された時、4バイトのPIC16 設定: 11 22 33 44 は 2211 4433 として表示されます。

プログラム・メモリ値を変えるためには変更する数値の上でクリックし新しい数値を入れて下さい。

テーブルに新しいセルを加えるためには insert key を使用

The screenshot shows the CCS Device Programmer software interface. The main window is titled "CCS Device Programmer" and has a menu bar with "Device", "File", "Diagnostics", "Production", and "User". Below the menu bar, there is a "Review / Edit Hex file contents" button and several radio buttons for "ZIF Socket", "Mach-X supplies power", and "Target Powered". The main area is titled "Program Memory" and contains a table of memory addresses and their corresponding hex values. The table has columns for addresses 0, 2, 4, 6, 8, A, C, and E. The values are displayed in hex, with some cells containing "0042". To the right of the table, there are controls for "Address" (int8, int16) and "Grouping" (int8, int16, int32). Below the table, there are buttons for "Save As", "Save", "Export", and "Print". At the bottom, there is a status bar showing "Target Programmed", a file path "C:\Evan\LED Test Programs\Tests\18F4xxx.hex", and "Pass = 3".

	0	2	4	6	8	A	C	E
000000	EF15	F000	6AEA	0E05	6EE9	50EF	E00D	0E06
000010	6E01	6A00	2E00	D7FE	2E01	D7FB	0E7B	6E00
000020	2E00	D7FE	2EEF	D7F3	0012	6AF8	9ED0	50C1
000030	0BC0	090F	6EC1	0E07	6EB4	0EC8	6E05	DPE2
000040	9A92	9A89	0EC8	6E05	DFDD	9A92	8A89	9893
000050	988A	0EC8	6E05	DFD6	9893	888A	9A93	9A8A
000060	0EC8	6E05	DFCF	9A93	8A8A	D7E7	0003	0042
000070	0042	0042	0042	0042	0042	0042	0042	0042
000080	0042	0042	0042	0042	0042	0042	0042	0042
000090	0042	0042	0042	0042	0042	0042	0042	0042
0000A0	0042	0042	0042	0042	0042	0042	0042	0042
0000B0	0042	0042	0042	0042	0042	0042	0042	0042
0000C0	0042	0042	0042	0042	0042	0042	0042	0042
0000D0	0042	0042	0042	0042	0042	0042	0042	0042
0000E0	0042	0042	0042	0042	0042	0042	0042	0042
0000F0	0042	0042	0042	0042	0042	0042	0042	0042
000100	0042	0042	0042	0042	0042	0042	0042	0042
000110	0042	0042	0042	0042	0042	0042	0042	0042

Save ショートカットは Ctrl-S

Save-As ショートカットは Ctrl-A

Print ショートカットは Ctrl-P

Configuration I/D [コンフィギュレーション]

この画面の中央エリアは全てのコンフィギュレーション・オプションを持っています。選択はこのエリアで変更します。

コンフィギュレーションのためのヘキサ値は右パネルにリストされています。変更されると、どの場所でも、両方の場所で値がアップデートされます。

チップ ID ワード(もし、チップがそれを持っている場合)もこの画面で表示されます。

High speed Osc (> 4mhz for PCM/PCH) (> 10mh)

- Fail-safe dock monitor enabled
- Internal External Switch Over mode enabled
- Power Up Timer

Reset when brownout detected

Brownout reset at 2.1V

- Watch Dog Timer

Watch Dog Timer uses 1:32768 Postscale

CCP2 input/output multiplexed with RC1

- PORTB pins are configured as analog input channels on RES
- Timer1 configured for low-power operation
- Master Clear pin enabled
- Stack full/underflow will cause reset

Config	
1	C200
2	1E1E
3	8700
4	0081
5	C00F
6	E00F

ID	

Save As

Save

Export

Print

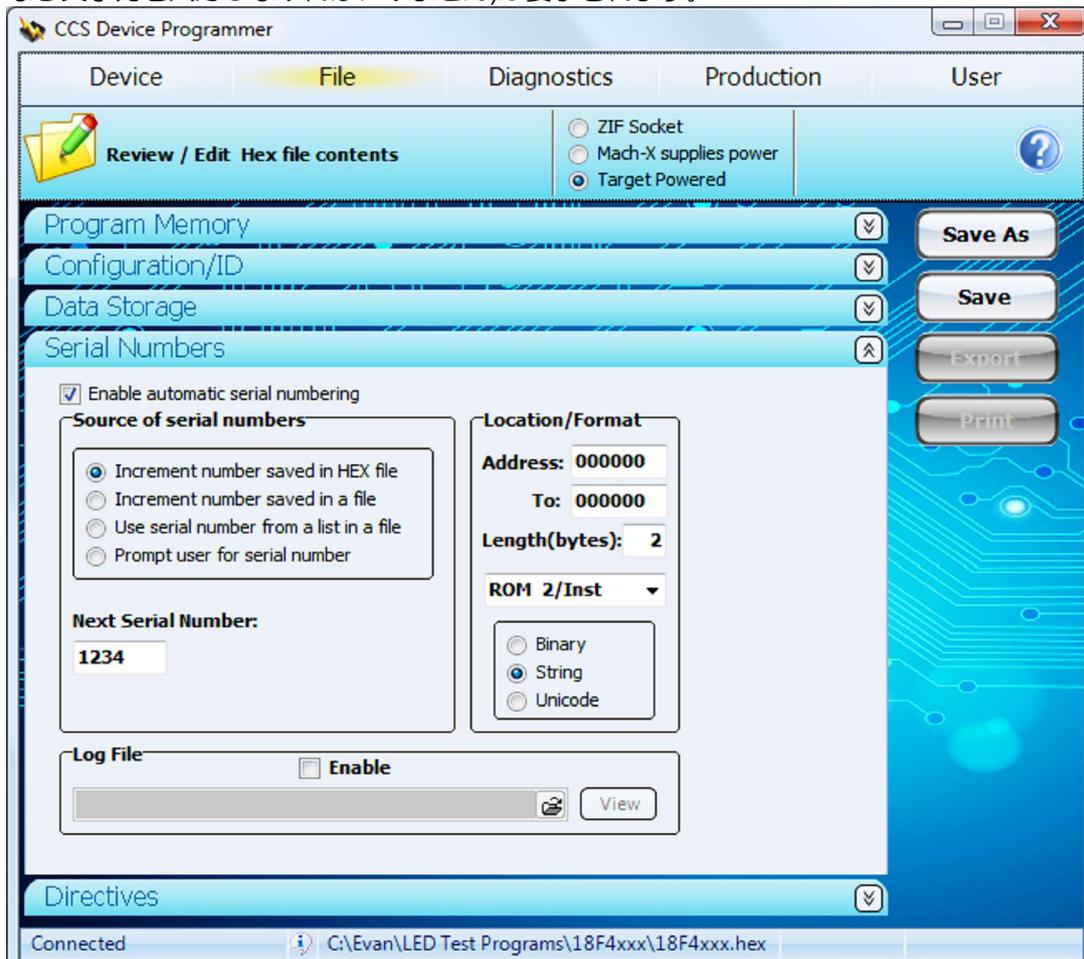
Target Programmed C:\Evan\LED Test Programs\Tests\18F4xxx.hex Pass = 3

Serial Numbers[シリアル番号]

Cレベルでのシリアル・ナンバーリングは#SERIALIZE ディレクティブを使ってセット・アップすること出来ます。

シリアル・ナンバーリングを有効にするには、+Automatic Serial Numbering” チェック・ボックスにチェックを入れて下さい。

シリアル番号のためのソースを選択すると、付加情報(例えば、ソースが入力を促すプロンプトなら文字列を入力しなければいけません)が要求されます。



・ **Increment number saved in HEX file**[ヘキサ・ファイルに保存された番号をインクリメント]
最初はヘキサ・ファイルのシリアル番号値を使ってロードしますとヘキサ・ファイルがアップデートされシリアル番号値を1ずつインクリメントします。

・ **Increment number saved in other file**[他のファイルに保存された番号をインクリメント 4]
シリアル番号は他のファイル(シリアル番号を持った1行のファイルである必要があります。)から取り出します。プログラマはファイルをアップデートし、シリアル番号値を1ずつインクリメントします。

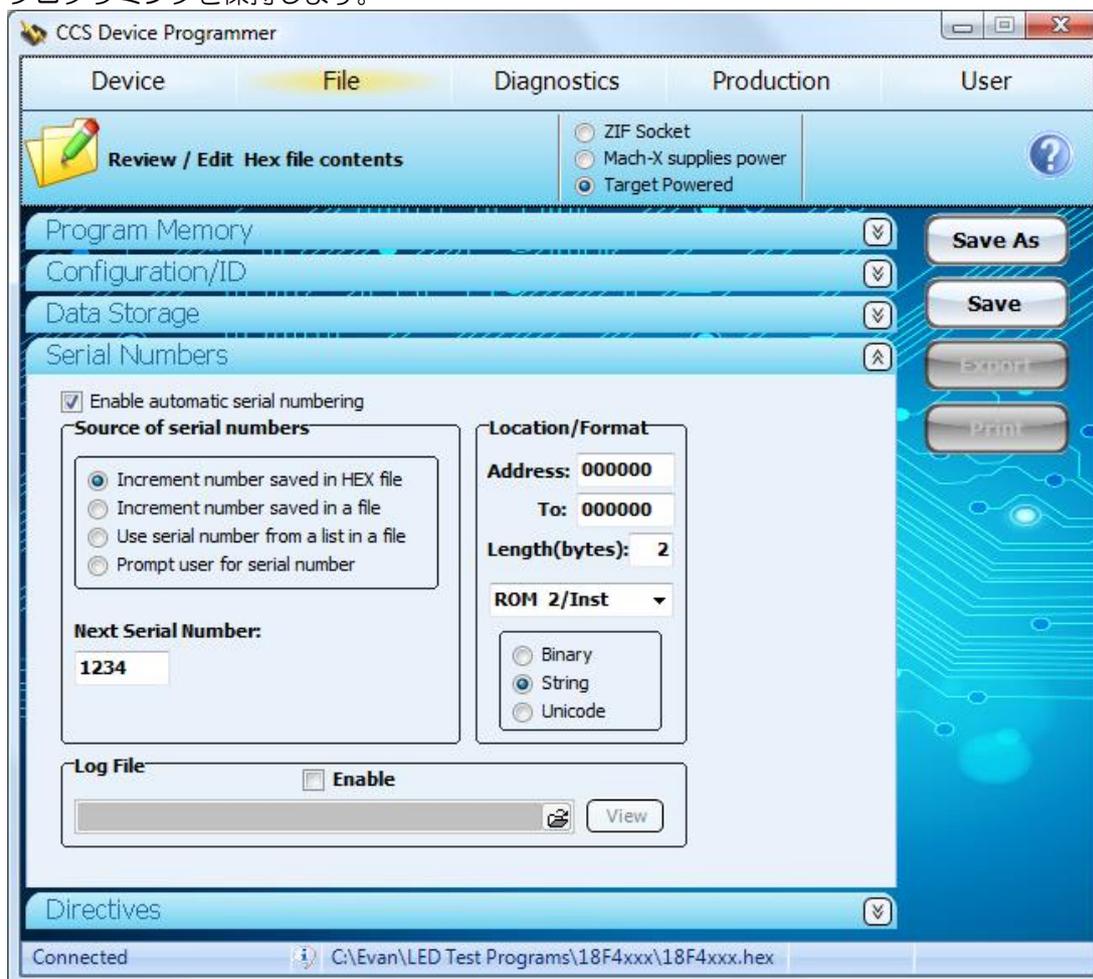
・ **Use serial number from list in other file**[他のファイルのリストからシリアル番号を使う]
シリアル番号は使用されるべき全てのシリアル番号を持っている他のファイル(ファイルは行毎にシリアル番号を持っている必要があります。)から取り出します。プログラマは最初の行を読

んで、そして、ファイルからの行を削除します。このオプションはマイクロチップ SQTP フォーマット・ファイルでも動作します。

・ **Prompt user for a serial number on each burn**

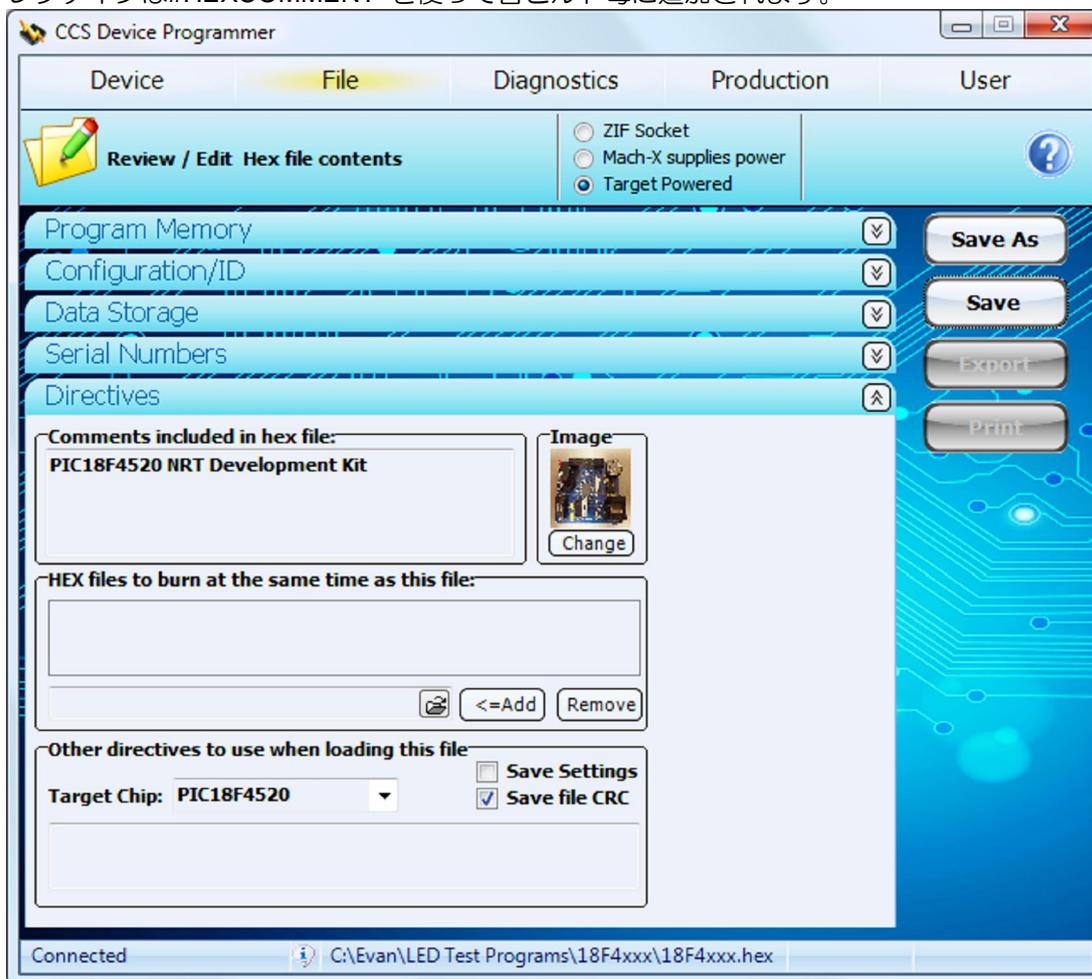
各書き込みでシリアル番号を、ユーザーにプロンプト表示で入力を促します。

もし、ログ・ファイル・オプションをチェックしますと、ログがこのヘキサ・ファイルで全てのプログラミングを保持します。



Directives[ディレクティブ]

CCS はデバイスをプログラミングするのを助けるためにオプションなディレクティブの数を保存するためにヘキサ・ファイル内にコメントを使っております。Cレベルではこれらのディレクティブは#HEXCOMMENT を使って各ビルド毎に追加されます。



サポートされているディレクティブ:

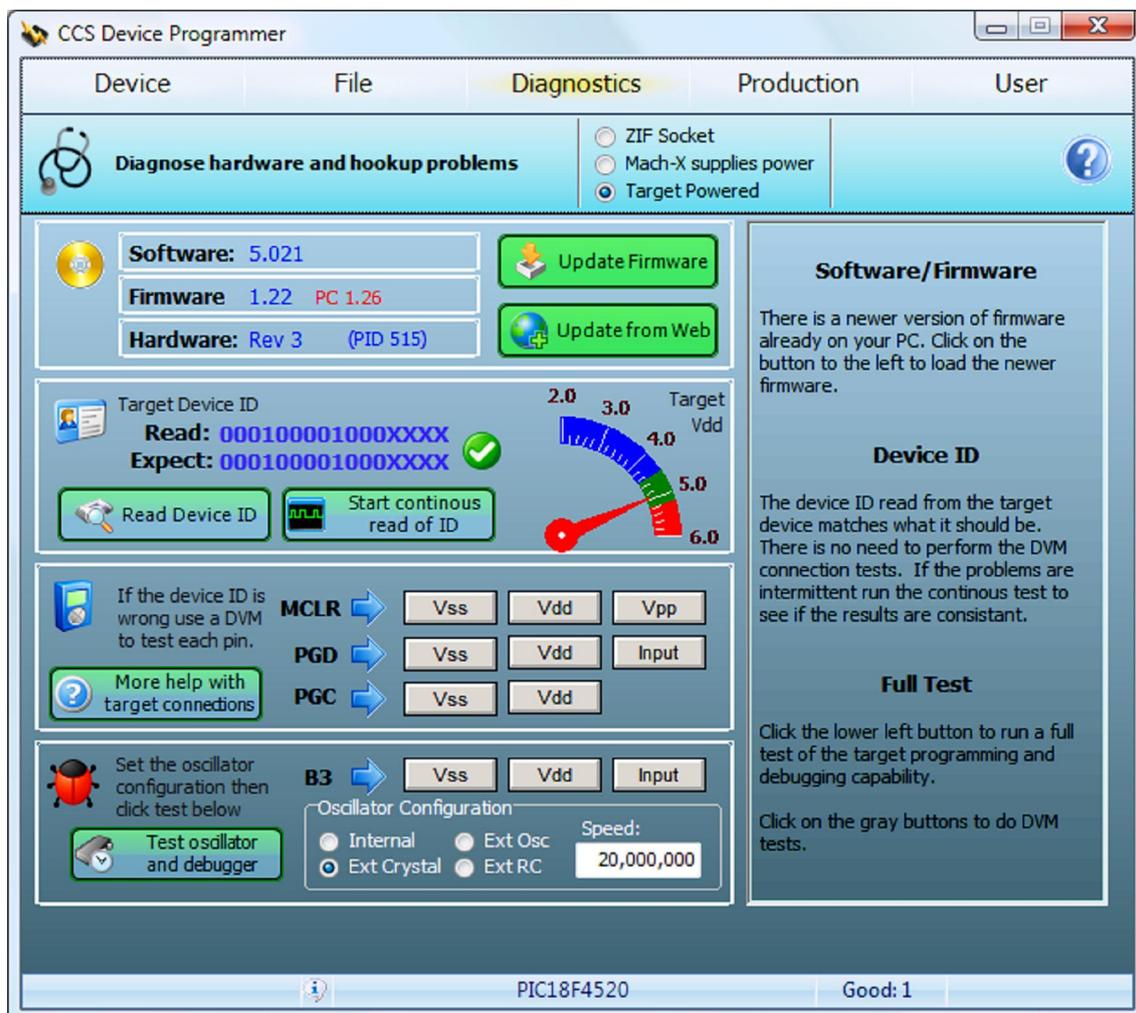
Comments	ここにリストされているものは全てデバイスがプログラムされる度にポップアップとして現れます。
Image	このイメージはプロダクション・ファイル・リストに現れます。イメージの標準サイズは 48x48 です。” change” をクリックして変更して下さい。このイメージはヘキサ・ファイルには置かれずにイメージにリンクされるだけです。
Links(hex files to burn)	このリスト・ファイルはこのファイルがプログラムされる時にプログラムされる必要があります。これはブート・ファイル、又は、別個のヘキサ・ファイルに置かれる定数テーブルで使用されます。ファイルを追加するにはファイル・アイコン上でクリックし、そして、ADD をクリックして下さい。ボックス下に現れているだけでは不十分ですので、ADD をクリックしなければいけません。
Target Chip	これは使用するターゲット・チップです。
Save Settings	セッティング・ウィンドウからのセッティングにチェックされた時はヘキサ・ファイルに保存されます。これらのセッテ

	ィングはヘキサ・ファイルがロードされた時は常に再保存されます。
Save CRC	チェックされますと、ファイル CRC、作成日付と変更された日付がファイルの終わりに行として加えられます。この CRC はすべてのデータ、コメント、及び、ディレクティブを含んでいます。CRC ミスマッチはデバイス・ウィンドウに赤で特定されます。
Other Directives	ここにはプログラムされる度にポップ・アップに示されたくないコメントがリストされるかも知れません。他のディレクティブ・エリアはターゲット・チップの下です。

File Windows ファイル・ウィンドウ

Diagnostics Windows [ダイアグノスティック・ウィンドウ]

このウィンドウはコフィギュレーションの問題を特定するために使われます。このウィンドウは5つの主なエリアに分かれています。右側はソフトウェアがどのような問題を検知したかによりノートとヒントが変わります。



- バージョン・チェック . トップ・パネルは青色文字で現在のバージョン、そして、古い場合は

赤色で新しいバージョンが表示されています。デバイス・プログラマに PC のディレクトリから新しいファームウェアをロードするためのボタンがあります。他のボタンはウェブから PC のディレクトへ新しいソフトウェア/ファームウェアをダウンロードします。

- デバイス ID チェック . 中央のパネルはターゲットとの通信が正しいかをチェックします。これを正しく動作するためにはデバイス・ウィンドウで正しいチップが選択されている必要が有ります。試みはパーツからデバイス ID を読みます、そして、予想されたデバイス ID とそれを比較します。それらがマッチしますと緑のチェックマークが示され、ミスマッチですと赤い X が表示されます。テストを再実行するには READ DEVICE ID ボタンをクリックして下さい。
- 大きな針はターゲット電圧を示しています。もし、これが悪いとターゲットへの接続トラブルが考えられ、これは Device ID の読み込みがうまく行っていないことを意味します。

- もし、デバイス ID が問題が無くても、プログラミングにまだ問題がある場合は、連続してデバイス ID の読み込み込むために continuous ボタンを試して下さい。成否のスコアは右下に表示されます。また、これはクロックとデータ・ピンがスコープでチェックするための良い機会です。
- DVM チェック . もし、デバイス ID を読むことが出来ない場合は、ターゲットとデバイス・プログラムの接続をチェックするためにこれらのテストを使って下さい。其々の 3 本のピンには、各ピンの可能な電圧にピンを設定し、そして、DVM でチップでのその電圧についてチェックして下さい。
- これは接続の問題かターゲット・ボードへのローディングの問題を区別するのに使用出来ます。
- デバッガ・テスト - デバイス ID を読み取ることが出来る場合デバッグをテストするためのオプションが提供されます。ボタンがクリックされた時良好なテスト・プログラムがダウンロードされ、検証され、実行されます。プログラムは停止され、そして、それが正しく機能していることを確かめるためにシングル・ステップされます。テスト結果は測定されたクロック速度で示されます。また、テストに使用する C ファイルを生成するオプションもあります。

Production Window[プロダクション・ウィンドウ]

このウィンドウはファイルの選択とプログラミングを簡単にするためにグリッド内にファイルのグループを表示するのに使用します。ファイルは単一のディレクトリの全てを表示します。ディレクトリには実際にファイルがある必要はありません。むしろショート・カットが実際のファイルを示します。

Device	File	Diagnostics	Production	User
12Fxxxx	PIC12F629	23-Dec-15 10:07		
16F6xx	PIC16F689	03-Dec-15 11:20		
16Fxxxx	PIC16F1937	18-Dec-15 13:58		
18F4xxx	PIC18F4520	NRT 18F4520 Development Kit 23-Dec-15 12:30		
18FxxJxx	PIC18F67J60	23-Dec-15 12:25		
18FxxKxx	PIC18F45K22	18-Dec-15 13:39		
24FJxxxGAxxx	PIC24FJ128GA00	09-Oct-15 13:48		
30Fxxxx	DSPIC30F4012	18-Dec-15 14:22		
33EPxxxMUxxx	DSPIC33E	18-Dec-15 14:06		

Start/Stop として Ctrl-S を使用.

- グリッド内の各行はプログラムされようとしているファイルを表しています。 行を選択し、そのファイルをプログラムするためには **START** をクリックします。列は下記を示しています：
 - Filename
 - Target chip
 - Comments and creation date
 - Image associated with this file.
- ハンズフリー・チェックボックス . チェックにされますとソフトウェアは PC で何もせずにデバイス・プログラマが接続された後にターゲットがプログラムされる特別なモードで作動します。例えば：Hands-Free mode にチェックを入れ、**START** をクリックします。この時にプラグインされるべきターゲットを聞いてくるメッセージが現れます。ターゲットにプラグされます。プログラミング中であることを示すメッセージが現れ、

そして、完了しますと PC からピープ音(成功、又は、失敗のピープ)を発します。この時、メッセージがターゲットの非接続するように言います。接続をはずしますと、メッセージは新しいターゲット・ボードを装着ためにメッセージが戻ります。このプロセスが PC で何もせずにプログラムする必要がある全てのボードで繰り返すことができます。

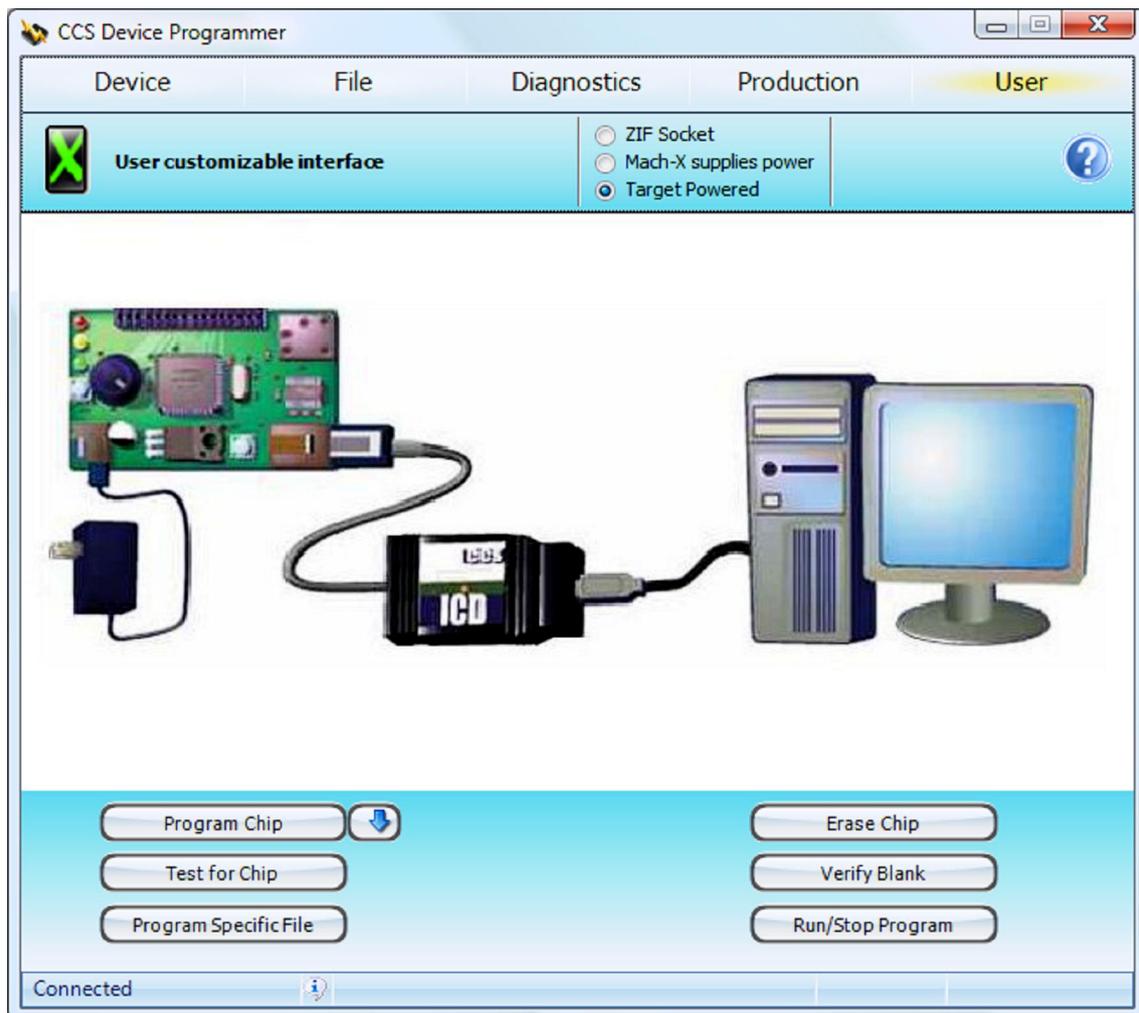
- リンク・ボタン作成 . このボタンをクリックして、他のディレクトリに位置するファイルにディレクトリへのショート・カットを作成します。
- ディレクト・ボタンをセット . 新しいディレクトリが表示されるように選択するのに使用します。ディレクトリは.ini ファイルに保存されます。そして、これは CCSLOAD コマンド・ラインからでもディレクトリを指定できます。
- セル(右クリックメニューからセルを選ぶ)の上をダブル・クリックしますと下記の特別な操作が出来ます:
- ファイル名コラム . 表示されたファイルがロードされているデバイス・ウィンドウへの切り替え
- デバイス・コラム . それがマッチしているかを確認するためにターゲット・チップをテスト
- コメント・コラム . ヘキサ・ファイルにコメントを変更、又は、追加します。
- イメージ・コラム - ヘキサ・ファイルにイメージを変更、又は、追加します。

User Window[ユーザー・ウィンドウ]

このウィンドウはユーザーによってカスタマイズすることが出来ます。ウィンドウ上には特定の機能の実行したり、又は、特定のファイルをロードするための9つのボタンがあります。

このページのイメージはユーザーが指定することも出来ます。このウィンドウの機能性は多くのユーザが古いICDソフトウェアをまねることも出来ます。

Customized User Screen[ユーザー画面のカスタマイズ] . ユーザーのためのユニークなセッティングは Options パネルを使って行われます。



この画面のショートカット・キーは：

F1	F4	F7
F2	F5	F8
F3	F6	F9

コマンド-ラインの使用方法

全てのコマンドライン・オプションは + 又は、 - で始めて下さい。ウィンドウズ・ユーザーはオプションの開始に / を使用、そして、リナックスのユーザーは ¥ を使用。もし、= がオプションに有る場合、先行文字はオプションです。何れの場合も先行文字の選択は操作に影響しません。

次のオプションが操作のセッティングを指定します。如何なるモードでも使用することが出来ます。:

@filename	表示されたファイルから追加のコマンド・ライン・ディレクトィブを読みます
filename	現在のワーキング・ファイル名をセット
AREAS=pc AREAS=ALL	ライト、リードとイレースする領域を指定。P, D と C の如何なる組み合わせをプログラム、データとコンフィギュレーションに使用することが出来ます。 例; AREAS=PC
DEBUGLOG	問題の解決のためにデバッグ・ログ・ファイルの診断を始めます
DEVICE=name	現在のワーキング・デバイス名をセット (ターゲット)
DIR=name	ワーキング・ディレクトリをセット。 ファイル名はこのディレクトリに関連づけられ、そして、このディレクトリはプロダクション・ウィンドウで使用されます
? Help Help = Browser	ヘルプ・ファイルを呼び出します。 リナックスでは Mozilla ブラウザーが他のブラウザがここに名前を指定するのに使用されます
MODE=BULK_ERASE MODE=MIN_ERASE	書き込み操作に必要な全領域、又は、その部分何れかをイレース
MODE=CHECKSUM_CONFIG MODE=NO_CHECKSUM_CONFIG	有効にされた時、スクリーンに示されていたチェックサムは構成メモリを含みます。これはあるビットがチェックサムに含まれていないのを確認するために Microchip アルゴリズムが使用されます
MODE=ERASE_ON_ERROR MODE=NO_ERASE_ON_ERROR	有効にされた時、プログラミングが失敗した後にチップはイレースされます。しかし、パーツにプロテクトされていないコードが残されるのを防ぐために時間が掛ります。プログラミングが失敗するときにはそれに注意してください、ヒューズが書かれていない構成を保護してください。ノート: プログラミングが失敗したときプロテクト・コンフィギュレーション・ヒューズは書き込まれません。
MODE=SOUND MODE=NO_SOUND	有効にした時、各操作の後 PC のスピーカーを通して PASS/FAIL のノイズで知らせます
MODE=VERIFY_PROTECT MODE=NO_VERIFY_PROTECT	有効にされた時、チップのプロテクトされた領域が読みだされないかをベリファイするために付加的なステップが実行されます。言い換えれば、保護機能がプログラムが確実に読み出しから保護されているかテストされています。確認メッセージはチップがこのテストに合格した時は常に表示されます
PORT=COMxx PORT=USBxxxx PORT=COM* PORT=USB*	デバイス・プログラムのポートを見るためにセットされます。COMxx は COM3 の様に、COM* は全部の COM ポート。USB の xxxx はシリアル番号、又は、プログラムの名前
POWER=TARGET	

POWER=ZIF POWER=ICSP	
SAVE	ini ファイルをデフォルトとして全てのセッティングをセーブ
SLOT=n SLOT=n, n, n... SLOT=ALL SLOT=MEMORYn	複数のディスティネーションを持っているデバイス・プログラマとして下記の様に行き先を指定します。SLOT=n は Prime-8 の 1-8。Memory1 から Memory4 は Prime-8 と Load-N-Go の内部メモリ。Sloat=ALL は全 8 つの Prime-8 のスロットを意味します
SN=xxxx SN=xxxx++ SN=xxxx,yyyy,zzzz	シリアル番号のためにプロンプトする代わりにこのオプションは前もってシリアル番号を指定します。これは通常、ユーザーがプロンプトされるシリアル番号にのみ適用されます。プログラムが実行される度に++記法で数を増加させます。カンマでセパレートされたリストは、その都度、リストから 1 つの数を取ります
VERBOSE	コマンドラインで CCSLOAD を使用する時に追加のエラー情報を有効にします。これは GUI で見ることの出来るステータス・バーの情報ボタンと同じです。
VERSION	ソフトウェアのバージョン表示
VOLTAGE=n.nn	それがサポートされているデバイス・プログラマのターゲットの Vdd 出力電圧をセット
WAIT NOWAIT	ウィンドウでコマンドラインが使用される場合、ユーザーが OK をクリックする迄ユニット・モードはウィンドウがアップされます。デフォルトでは待ちません。エラー全く無い場合は待ちます。
WINDOW=DEVICE WINDOW=SETTINGS WINDOW=FILE WINDOW=DIAGNOSTICS WINDOW=PRODUCTION WINDOW=USER WINDOWONLY=-	示されたウィンドウを開いてプログラムを開始 WINDOWONLY が使用されているときは他のウィンドウに切り替えることが出来ません。

CCSLOAD プログラムを実行する時、下記のコマンド・オプションは、通常ユーザー・インタフェースの表示とならないで、代わりにポップアップ・ウィンドウにより、進捗状況と結果が表示されます。コマンドは整然とした状態で実行され、そして設定はコマンドの間で変化するかもしれません。また、これらのコマンドは CCSLOADER プログラムのすべてで有効です。

CALIBRATE	サポートしているパーツにおいて、内部オシレータをカリブレートします。
CHECKFILE=filename	コマンド・ラインの使用のため指定された hex ファイルの情報を表示します。データの範囲とエラーは GUI ノートブックでの様に表示されます。
CLOSE	開いた CCSLOAD プログラムを終了。"Stay Open"をご覧ください。
ERASE	ターゲット・チップをイレース。
HALT	ターゲットの動作を停止 (MCLR はローに保持されます)。
READ=filename	ターゲット・チップを読み込み、そして、内容を指定された hex ファイルに書き込みます。
RUN	プログラムの動作を開始(MCLR は Vdd にセットされます)。
SETUP	PCW IDE に CCSLOAD へのリンクをインストール。コマンド・ラインの残りは無視されます。

STAYOPEN	プログラムとデバイス・プログラムの接続状態を開いたままにします。CCSLOAD、又は、CCSLOADERに関連したコマンドに対して、開いているプログラムがある時は、時間を節約するために開いているプログラムに対して発行されます。
TEST TEST=device	デバイス・プログラマとの接続をテスト、そして、オプションでターゲットとの接続もテストします。
UPGRADE=filename	デバイス・プログラマに新しいファームウェアをインストール。filename は通常の状態では CCS .FW ファイルでなければいけません。
UPGRADES�AVE=filename	デバイス・プログラマのスレーブ・プロセッサに新しいファームウェアをインストール。filename は通常の状態では CCS .FW ファイルでなければいけません。*これは内部にマルチ・プロセッサを持ったデバイス・プログラマにのみ対応しています。
VERIFY=BLANK	ターゲット・チップがブランクであるかをベリファイします。
VERIFY=filename	ターゲット・チップが指定されたファイル名とマッチするかベリファイします。
WRITE=filename	指定されたファイル名でターゲット・チップをプログラムします。

次のコマンドは以前のバージョンとの互換性のために提供されています：

オプション	同じ:
devicename	DEVICE=devicename
0	POWER=ZIF
1	POWER=ICSP
2	POWER=TARGET
S	PORT=COM*
SX	PORT=COMx
U	PORT=USB*
Uxxxx	PORT=USBxxxx
RAfilename	AREAS=PDC READ=filename
RDfilename	AREAS=D READ=filename
Tfilename	WRITE=filename +RUN
WS	WINDOW=PRODUCTION
WC	WINDOW=DEVICE
+ICSP	POWER=TARGET
+ICSP_VDD	POWER=ICSP
+ZIF	POWER=ZIF

ウィンドウズ終了コード：

0 - 全てが GOOD

1. デバイス・プログラマと接続出来ませんでした

4. 操作失敗

例：

CCSLOAD DIR="c:\picprojects¥" Window=Production

CCSLOAD +USBmaster

CCSLOAD AREAS=PDC WRITE="c:\picprojects¥myfile.hex" +RUN

HEX ファイル・フォーマット

使用される hex ファイル形式は一部のマイクロチップと CCS 拡張機能を備えた標準 Intel 形式です。標準フォーマットはコロンで始まるデータレコードとセミコロンで始まるコメント行で構成されています。Intel が定義した 6 つのレコードのうち PIC hex ファイルは次の 3 種類のレコード(それぞれ CR/LF、又は、LF でターミネートされた)のみをサポートします。

```
:LLAAA00DD..DDCC      データ行(type 0)
:02000004PP00CC      上位アドレス・ビットをセット(type 4)
:00000001FF          ファイルの終了(type 1)
```

LL DD hex ペアの数
AAAA アドレスの下位 16 ビット(最初の DD バイトのアドレス)
DD..DD データバイトのリスト。ワード命令のためにデータは LSB ファーストで保存されます。
PP アドレスの Bits 16-23
CC チェックサム。ライン上の各 hex ペアを合計し CC はその 2 の補数です。

このフォーマットは INHX8 フォーマットと呼ばれ、ファイルに type 4 のレコードがない場合、INHX32 ファイル(32bits アドレス指定)です。

アドレスはバイト・アドレスですが多くのデータシートはワードアドレスを使用してメモリを参照します。これらのパーツではデータシートのアドレスに 2 を掛けて hex ファイルのアドレスを取得します。

Microchip extensions:

チップごとにマイクロチップのプログラミング仕様にはコンフィギュレーション・ヒューズ、データ EEPROM 値とユーザ ID データ用のデータを定義するためにプログラム・メモリ以外の特別なアドレス範囲が識別されています。

多くの部分で HEX ファイルに未使用のビットやバイトがあることに注意してください。例えば、PIC24 の各パーツでは各命令は 24 ビットですが余分な未使用バイトが各命令に含まれ HEX ファイルで 32 ビットになります。

CCS extensions:

- CCS ソフトウェアはパーツがプログラミングされるたびに最初のデータ行の前のコメントをユーザーに表示するものとして取り扱います。
- EOF 行の後のコメント行にはターゲット・チップのパーツ番号が含まれている場合があります。
- ;BURNTIME、及び/又は、;SSUNICODE で開始されたコメント行はシリアル番号情報とともに表示されることがあります。
- ;SETTINGS、及び/又は、;IMAGE で開始されたコメント行は CCSLOAD プログラム(ターゲット電圧等の様な)により使用されたセッティングとともに表示されることがあります。
- ;LINKHEX で開始されたコメント行はプログラミング時に使用する必要がある別の hex ファイルを示すために表示されることがあります。
- コメント行はこの行の前のファイル内のすべての印刷可能文字の 16 ビット CRC を指定する;CRC を開始する最後の行として表示されることがあります。この行は作成され変更された日付、又は、プログラマの名前も含まれます。

すべての権利は国際条例で保護されています。作品に関して、製作者の許可なく写真模写・記録・テーピングや情報検索システムに取り入れることや電子機器・グラフィック・メカニズムの複製やコピーを行うことをいかなる場合も禁じます。

注意

本書の内容は予告なしに変更されることがありますのでご了承ください。
本書の内容については万全を記して作成しましたが、万一誤りなど、お気づきの点がございましたら、ご連絡をお願い致します。本書に記載されている使用の結果につきましては、2項にかかわらず当社は責任を負いかねますのでご了承ください。本書と異なる使用をされたり、あるいは本書に記載されていない使用をされた場合の結果について当社は責任を負いません。本書を無断で複写、引用、配布することは禁じられております。

テクニカル・サポートは E-mail でお願い申し上げます。

(有)データダイナミクス/テクニカル・サポート

E-mail アドレス: support@datadynamics.co.jp

ご質問の前に下記の **FAQ** をご覧下さい。

コンパイラ関連 FAQ: <http://www.datadynamics.co.jp/ccsc/faq.html>

デバッグ関連 FAQ: http://www.datadynamics.co.jp/ccsc/icd_faq.html