



# Cyclone MAX User Manual



CYCLONEMAXUM  
Rev. 1.04  
October 2007



## Purchase Agreement

P&E Microcomputer Systems, Inc. reserves the right to make changes without further notice to any products herein to improve reliability, function, or design. P&E Microcomputer Systems, Inc. does not assume any liability arising out of the application or use of any product or circuit described herein.

This software and accompanying documentation are protected by United States Copyright law and also by International Treaty provisions. Any use of this software in violation of copyright law or the terms of this agreement will be prosecuted.

All the software described in this document is copyrighted by P&E Microcomputer Systems, Inc. Copyright notices have been included in the software.

P&E Microcomputer Systems authorizes you to make archival copies of the software and documentation for the sole purpose of back-up and protecting your investment from loss. Under no circumstances may you copy this software or documentation for the purpose of distribution to others. Under no conditions may you remove the copyright notices from this software or documentation.

This software may be used by one person on as many computers as that person uses, provided that the software is never used on two computers at the same time. P&E expects that group programming projects making use of this software will purchase a copy of the software and documentation for each user in the group. Contact P&E for volume discounts and site licensing agreements.

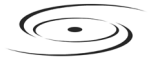
P&E Microcomputer Systems does not assume any liability for the use of this software beyond the original purchase price of the software. In no event will P&E Microcomputer Systems be liable for additional damages, including any lost profits, lost savings or other incidental or consequential damages arising out of the use or inability to use these programs, even if P&E Microcomputer Systems has been advised of the possibility of such damage.

By using this software, you accept the terms of this agreement.

MS-DOS & Windows are registered trademarks of Microsoft Corporation. Freescale is a registered trademark of Freescale, Inc. IBM is a registered trademark of IBM corporation.

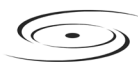
P&E Microcomputer Systems, Inc.  
P.O. Box 2044  
Woburn, MA 01888  
617-353-9206  
<http://www.pemicro.com>

Manual version 1.04



---

1	INTRODUCTION .....	1
2	GETTING STARTED .....	3
2.1	Software Installation .....	3
2.2	Overview Of Cyclone MAX Configuration Utility .....	3
3	CYCLONE MAX HARDWARE .....	6
3.1	Power Source .....	6
3.2	Serial (RS232) Communications Port .....	6
3.3	USB 1.1 Communications Port .....	7
3.4	Ethernet Communications Port .....	7
3.5	PORT A – ColdFire V1 Core .....	8
3.6	PORT B – PowerPC Nexus Interface .....	9
3.7	PORT C – ColdFire Interface & ColdFire Extension Cable .....	11
3.8	PORT D – PowerPC Interface .....	14
3.9	PORT E – ARM Nexus .....	15
3.10	PORT F – Reserved .....	16
3.11	Expansion Port (CompactFlash) .....	16
3.12	Cyclone MAX Buttons .....	17
3.13	LEDs .....	17
3.14	Ribbon Cable with IDC Socket .....	17
4	STAND-ALONE PROGRAMMER CONFIGURATION .....	20
4.1	Target Architectures .....	21
4.2	BDM Shift Clock Delay Constant .....	26
4.3	Specify Programming Algorithm and S-Record .....	26
4.4	Base Address .....	26
4.5	Script Wizard .....	26
4.6	Programming Operations .....	29
4.7	Store Image To Cyclone .....	30
4.8	Save Image/Cyclone Configuration .....	31
4.9	Configuration Via LCD Menu .....	31
5	STAND-ALONE PROGRAMMER MANUAL CONTROL .....	39
5.1	Via Cyclone MAX Buttons .....	39



---

5.2	Operation Procedure via LCD Menu.....	40
6	STAND ALONE PROGRAMMER AUTOMATED CONTROL.....	43
6.1	Automated .EXE Control Of The Cyclone.....	43
6.2	Automated .DLL Control Of The Cyclone .....	46
6.3	Control Of Multiple Cyclone Units.....	55
7	PC-HOSTED DEBUG/PROGRAMMING SOFTWARE.....	57
7.1	Freescall Codewarrior .....	57
7.2	P&E Microcomputer Systems' Software .....	61
8	ETHERNET PORT CONFIGURATION .....	65
8.1	Network Architecture .....	65
8.2	Network Parameters .....	66
8.3	Internet Protocol .....	66
8.4	Connecting the Cyclone Device.....	67
8.5	Cyclone IP Setup Via LCD Menu.....	69
8.6	Cyclone IP Setup Utility User Interface (ConfigureIP) .....	69
8.7	Using IPSetup.exe to configure the Cyclone Max .....	71
8.8	Managing Multiple Images.....	73
9	SERIAL PORT CONFIGURATION.....	75
10	USB PORT CONFIGURATION .....	76



## 1 INTRODUCTION

The Cyclone MAX is a versatile tool that can be used for the programming, debugging, and testing of microprocessor-based hardware. It can be controlled interactively via a Host PC, or operate independently of the PC in stand-alone mode. The Cyclone MAX can also be configured and operated via Serial, USB, and Ethernet communication interfaces. When in stand-alone mode, its programming functions can be initiated directly through user input buttons, or by using those same buttons to make selections from the LCD menu.

The Cyclone MAX is able to debug and program a variety of processor architectures using the target hardware's Background Debug Mode (BDM) or JTAG ports.

### • Supported Architectures

- PowerPC (MPC5xx/MPC8xx)
- PowerPC Nexus (MPC55xx)
- ColdFire (MCF5xxx)
- ARM (MAC7xxx Automotive Family)

In addition to the versatility of being able to debug and program the aforementioned processor families, and the ability to function in stand-alone mode, the Cyclone MAX includes these other features:

### • Multiple Communications Interfaces

- Ethernet 10/100 baseT
- USB 1.1
- Serial Baud 115200, no parity, 8 data bits, 1 stop bit.

### • Onboard Storage

- Onboard storage of 7 Megabytes. Stores the image to be programmed, programming algorithm, and any configuration settings.



- Multiple images may be saved using the onboard storage.

- **Multi-Voltage Operation**

- Automatically detects and caters to target voltages ranging from 1.8 to 5 Volts.

- **LCD Menu**

- Allows simple, menu-based control of the configuration and function of the Cyclone MAX.

In short, the Cyclone MAX is an all-in-one solution for debugging, programming, and testing of your hardware.



---

## 2 GETTING STARTED

Stand-alone programming is probably the most common use of the Cyclone MAX, therefore this section provides a simple quick-start guide to 1) configuring the Cyclone MAX with data, and 2) using it in stand-alone mode to program the flash of a target system. The user is encouraged to read this manual in its entirety for a complete description of the full feature set of the Cyclone MAX.

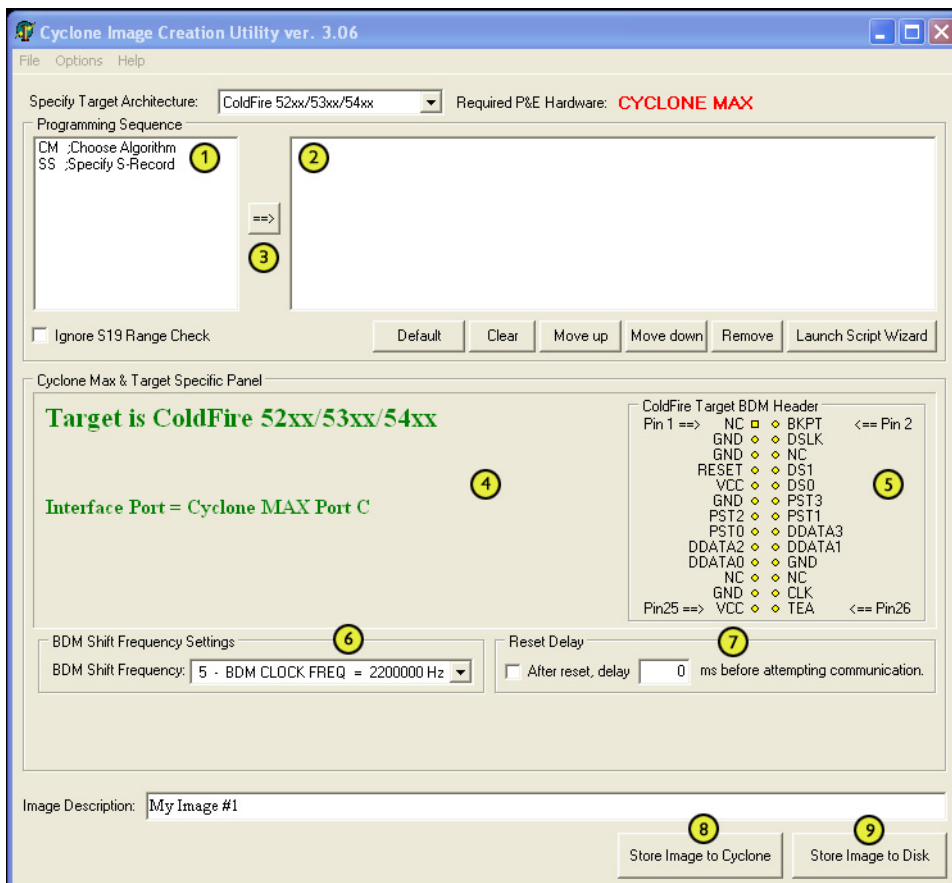
### 2.1 Software Installation

The first step is to install the accompanying software. This will install all of the applications and drivers that will be used to configure/control the Cyclone MAX. Once the installation is complete and the PC has been rebooted you may begin to configure the unit for operation.

### 2.2 Overview Of Cyclone MAX Configuration Utility

If you plan to use the Serial or USB ports for communication, then the Cyclone MAX can be used right out of the box. If you wish to use the Ethernet port, however, you will need to configure the corresponding IP numbers before use. The Ethernet port will not be functional until this configuration is complete. Please read **Section 8 - ETHERNET PORT CONFIGURATION** for more information about the Cyclone IP Setup Utility.

Once you have connected the Cyclone MAX to the PC via either a Serial or a USB cable, you can launch the Cyclone MAX Configuration Utility (CreatelImage.exe) application:



**Figure 2-1: Cyclone Image Creation Utility**

- (1) Provides a list of available functions that the user may select and add to the programming sequence.
- (2) This window is where the programming steps are sequenced.
- (3) This button moves selected programming functions into the sequence window.
- (4) Displays the processor architecture and the corresponding interface





port on the Cyclone MAX.

- (5) Displays the pinout configuration for the selected architecture.
- (6) Allows the user to set the BDM Shift Frequency by selecting from the drop-down box.
- (7) Allows the user to set the Reset Delay.
- (8) The “Store Image To Cyclone” button allows the user to store the current image on the Cyclone MAX.
- (9) The “Store Image To Disk” button allows the user to store the current image on their hard drive. The Manage Image Utility can be used to configure the Cyclone with one or more images.

At this stage you are simply configuring the Cyclone Max with the appropriate data file and programming procedure. Actual programming of the target hardware occurs later.

Once the information required in Areas 1-4 is complete, you may configure the Cyclone Max by clicking the “Store Image to Cyclone MAX” button. **Figure 2-1** indicates that this will be performed over the USB Port, USB1, as displayed in **Area 4**. You may also select the Serial port, or the Ethernet port if it has been configured appropriately.

After the image has been saved to the Cyclone Max, you can program a target by simply pressing the “Start” button on the Cyclone Max. This will begin the sequence that was specified in **Area 3** (in this example: erase, blank check, program, verify programmed data).

The complete functionality of the Cyclone MAX and its accompanying utilities is beyond the scope of this **Getting Started** section. Please read further to learn more about the full feature set and operation of the Cyclone MAX. For a detailed description of the Cyclone MAX Configuration Utility, please refer to **Section 4 - STAND-ALONE PROGRAMMER CONFIGURATION**.



## **3 CYCLONE MAX HARDWARE**

This section describes the various ports and inputs of the Cyclone MAX.

### **3.1 Power Source**

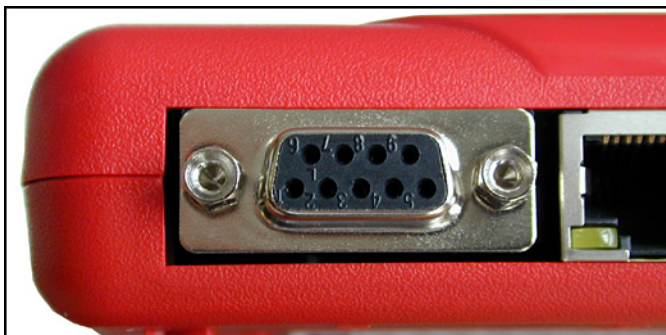
The Cyclone MAX requires a regulated 6V DC Center Positive power supply with 2.5/5.5mm female plug. The Cyclone MAX receives its power from the power jack located on the side of the unit.



**Figure 3-1: Power Jack**

### **3.2 Serial (RS232) Communications Port**

The Cyclone MAX provides a DB9 Female connector for communication with a host computer through the Serial (RS232) Port. Communication takes place at a fixed baud rate of 115200 Baud, 8 Data bits, No Parity, and 1 Stop Bit.

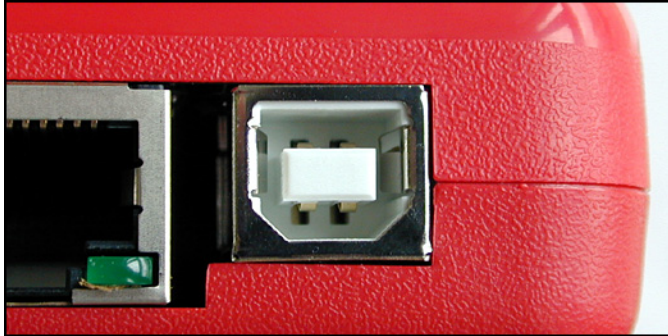


**Figure 3-2: Serial Port**



### **3.3 USB 1.1 Communications Port**

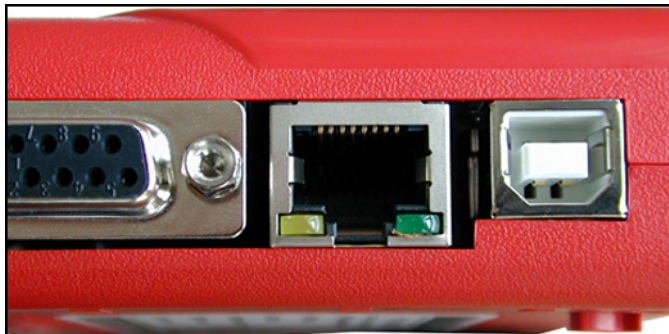
The Cyclone MAX provides a USB connector for communicating with a PC via the Universal Serial Bus port. The Cyclone MAX is a USB 1.1 compliant device.



**Figure 3-3: USB Port**

### **3.4 Ethernet Communications Port**

The Cyclone MAX provides a standard RJ45 socket for communication with a host computer through the Ethernet Port. The Ethernet Port of the Cyclone MAX is a 10/100 baseT port.

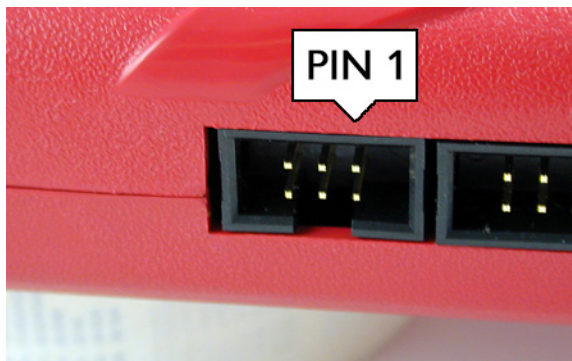


**Figure 3-4: Ethernet Port**



### 3.5 PORT A – ColdFire V1 Core

Port A is used to connect to the ColdFire V1 core family of microprocessors. For the physical dimensions of the connector, please see **Section 3.14 - Ribbon Cable with IDC Socket**.



**Figure 3-5: Port A**

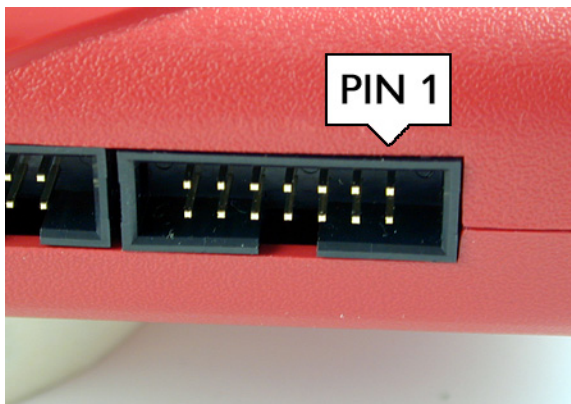
#### **ColdFire V1 Core Pinout**

BKGD	1	2	GND
NC	3	4	RESET
NC	5	6	VDD



### 3.6 PORT B – PowerPC Nexus Interface

Port B can be used to connect to PowerPC Nexus (MPC55xx) targets. For the physical dimensions of the connector, please see **Section 3.14 - Ribbon Cable with IDC Socket**.



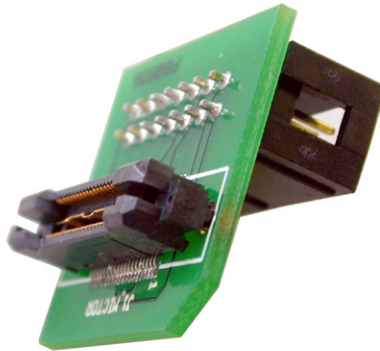
**Figure 3-6: Port B - PowerPC Nexus Interface**

#### PowerPC Nexus Pinout

TDI	1	2	GND
TDO	3	4	GND
TCK	5	6	GND
NC	7	8	NC
RESET	9	10	TMS
VDDE7	11	12	GND
RDY	13	14	JCOMP

#### 3.6.1 BERG14-to-MICTOR38 Optional Connector (PORT B)

P&E offers a 14-pin BERG to 38-pin MICTOR adapter, sold separately, that may be used on Port B of the Cyclone MAX. The P&E part number is BERG14-TO-MICTOR38.

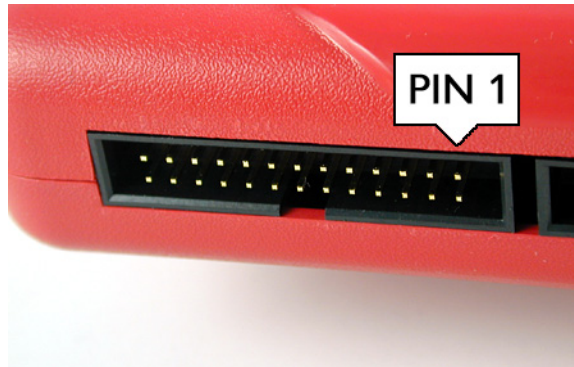


**Figure 3-7: BERG14-TO-MICTOR38 Adapter (Sold Separately)**



### 3.7 PORT C – ColdFire Interface & ColdFire Extension Cable

Port C is the interface for the ColdFire MCF52xx/53xx/54xx family of microprocessors. This port connects to the target hardware using either the ColdFire extension cable (for synchronous ColdFire targets) or a standard 26-pin ribbon cable (for asynchronous ColdFire targets). Both cables are provided in the Cyclone MAX package. Please refer to each processor's user manual to identify whether it is a synchronous or asynchronous interface. If you are not sure, or for test purposes, the synchronous cable will work with all 52xx, 53xx, and 54xx targets. For the physical dimensions of the connector, please see **Section 3.14 - Ribbon Cable with IDC Socket**.



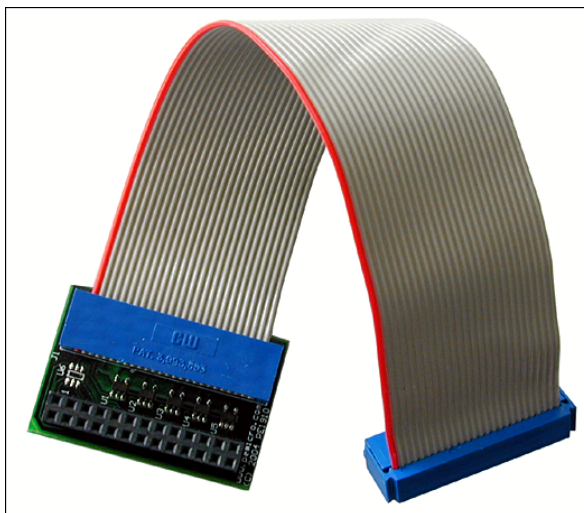
**Figure 3-8: Port C - ColdFire Interface**



**ColdFire Pinout**

N/C	1	2	BKPT
GND	3	4	DSCLK
GND	5	6	N/C
RESET	7	8	DSI
VCC	9	10	DSO
GND	11	12	PST3
PST2	13	14	PST1
PST0	15	16	DDATA3
DDATA2	17	18	DDATA1
DDATA0	19	20	GND
N/C	21	22	N/C
GND	23	24	CLK
VCC	25	26	TEA

The ColdFire extension cables, one for Synchronous targets and one for Asynchronous targets, are pictured below:



**Figure 3-9: ColdFire Extension Cable With Adapter (Synchronous ColdFire Targets)**





**Figure 3-10: ColdFire Ribbon Cable (Asynchronous ColdFire Targets)**



### 3.8 PORT D – PowerPC Interface

Port D can be used to connect to PowerPC BDM (5xx/8xx) targets. For the physical dimensions of the connector, please see **Section 3.14 - Ribbon Cable with IDC Socket**.

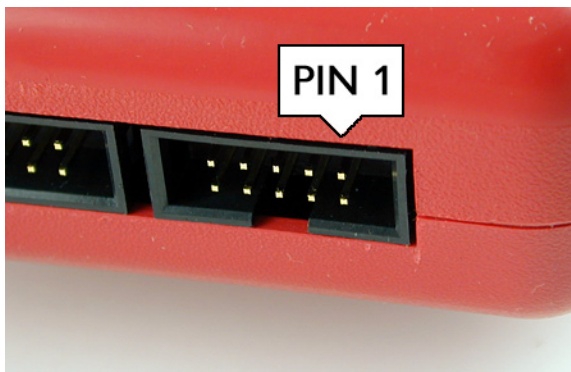


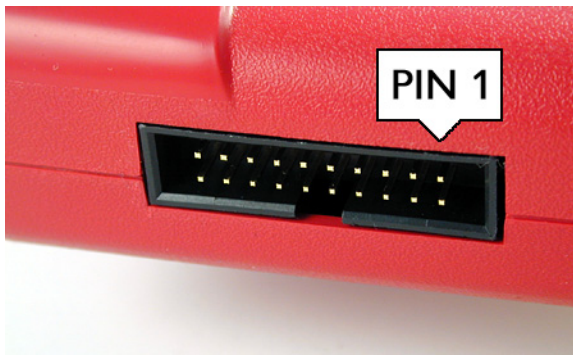
Figure 3-11: Port D - PowerPC Interface

#### PowerPC Pinout

N/C	1	2	SRESET#
GND	3	4	DSCLK
GND	5	6	N/C
HRESET#	7	8	DSDI
VDD	9	10	DSDO

### 3.9 PORT E – ARM Nexus

Port E can be used to connect to ARM (MAC7xxx) targets. For the physical dimensions of the connector, please see **Section 3.14 - Ribbon Cable with IDC Socket**.



**Figure 3-12: Port E- ARM Nexus**

ARM Nexus Pinout			
TVCC	1	2	N/C
TRST#	3	4	GND
TDI	5	6	GND
TMS	7	8	GND
TCK	9	10	GND
N/C	11	12	GND
TDO	13	14	GND
RESET#	15	16	GND
N/C	17	18	GND
N/C	19	20	GND



### **3.10 PORT F – Reserved**

Port F is reserved for future use by P&E. Do not connect to this port.



**Figure 3-13: Port F**

### **3.11 Expansion Port (CompactFlash)**



The CompactFlash Port is for future use by P&E. Do not use this port.



**Figure 3-14: CompactFlash Port (Unimplemented)**

### 3.12 Cyclone MAX Buttons

There are five (5) buttons on the top of the Cyclone MAX which are used for stand-alone programming and to navigate the LCD menus. They are specified as follows.

<u>Button</u>	<u>Function</u>
START / 	Start executing the tasks pre-configured into the Cyclone MAX. <b>Menu Mode:</b> Navigate upwards in LCD menu.
AUX / 	Perform auxiliary function (stand-alone verification). <b>Menu Mode:</b> Navigate downwards in LCD menu.
MENU / [SELECT]	Displays LCD Menu (enters Menu Mode). <b>Menu Mode:</b> Select highlighted item in LCD menu.
CANCEL	Cancel the tasks being executed and go back to the standby state.
RESET	Hardware reset of the Cyclone MAX.

### 3.13 LEDs

Eight LEDs are used to provide programming and Cyclone MAX status:

1. Power – Indicates power has been supplied to Cyclone MAX
2. Ready – The Cyclone MAX is ready to perform programming functions
3. Image – Indicates the presence of one or more images on board.
4. Erase – Erasing of target flash is in progress
5. Program – Programming of target flash is in progress
6. Verify – Verifying of target flash is in progress
7. Error – An error occurred during programming
8. Success – Programming functions completed successfully

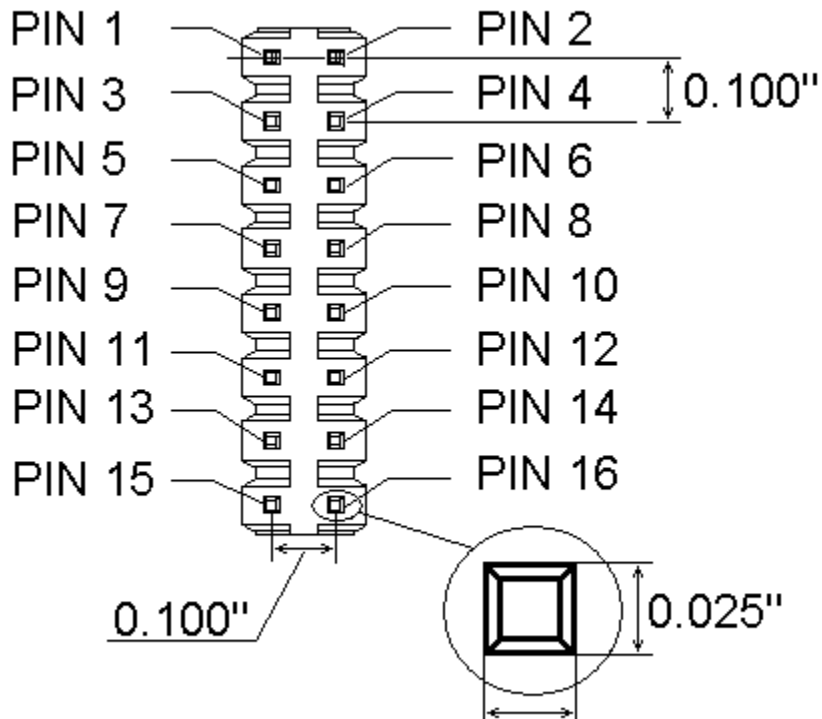
### 3.14 Ribbon Cable with IDC Socket

The Cyclone MAX provides several standard 0.100-inch pitch dual row 0.025-inch square headers (Ports A-F, as described earlier in **Section 3 - CYCLONE MAX HARDWARE**). A typical layout for one such header is shown below. Using these headers, the Cyclone MAX can communicate with



the target through a number of different-sized ribbon cables. Each of the ribbon cables have a 0.100-inch centerline dual row socket IDC assembly (not keyed). The ribbon cables are designed such that the Cyclone MAX Ports and target BDM Ports must have the same pinout, i.e., Pin 1 of a Cyclone MAX Port is connected to Pin 1 of the target BDM header. Here is an example header:

### 0.100" Dual Row, 0.025" Square Header



Not Drawn to Scale 0.025"

**Figure 3-15: Example 16-Pin Header Diagram**

The following notes relate to the pinout labels on Cyclone MAX Ports:



- 
- N/C signifies a No Connect. This pertains to pins that are reserved for future use by Freescale or P&E Microcomputer Systems, and should not be connected.
  - A signal name ending in # (e.g. RESET#) signifies an active-low signal.



## 4 STAND-ALONE PROGRAMMER CONFIGURATION

The Cyclone MAX may act as a stand-alone in-circuit programmer for PowerPC MPC5xx/8xx, PowerPC Nexus MPC55xx, ColdFire MCF5xxx, and ARM MAC7xxx targets. A simple user interface, CreateImage.exe, is provided for configuring the Cyclone MAX.

The Cyclone MAX does not require a target to be connected when it is being configured. However, the Cyclone MAX must be powered on (indicated by the “Ready” LED), and either the PC Serial port, USB port, or Ethernet port must be connected to the Cyclone MAX. **Figure 4-1** shows the configuration dialog with an example configuration. The configuration details are explained below.

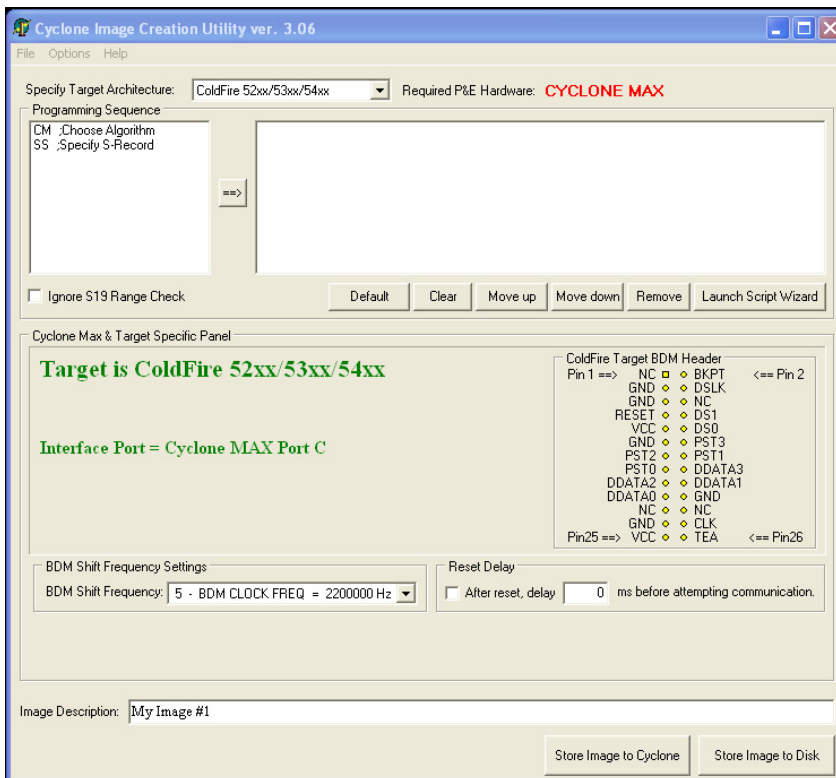


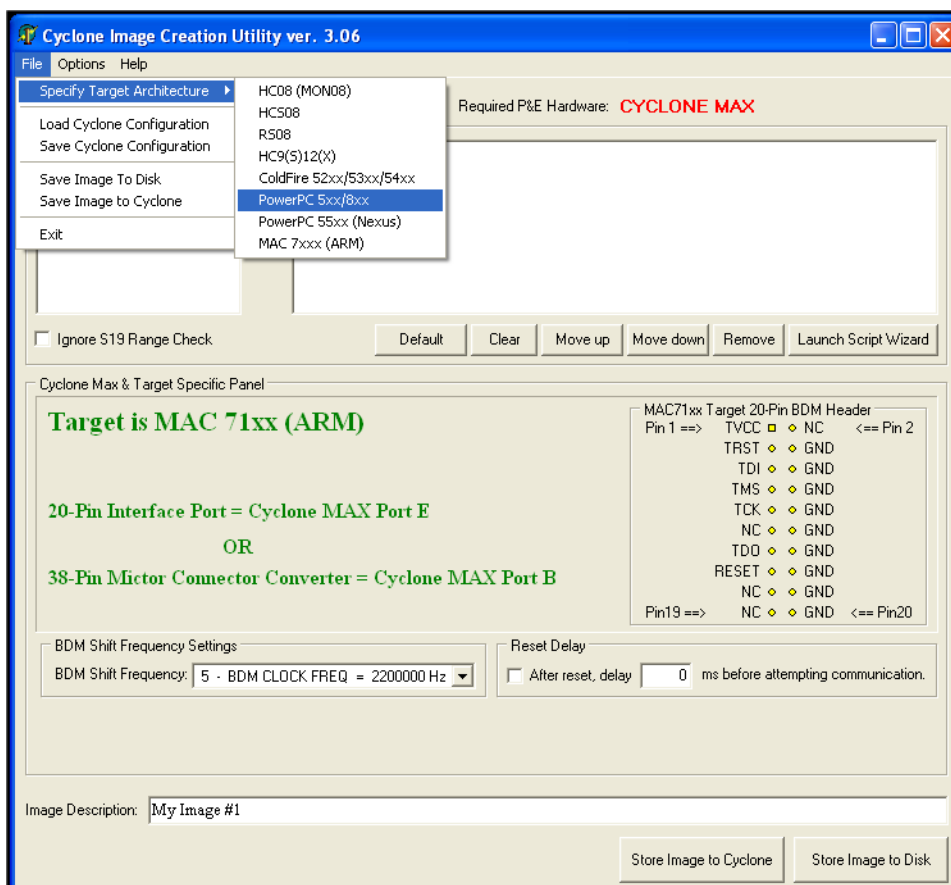
Figure 4-1: Image Creation Utility



## 4.1 Target Architectures

The Cyclone MAX supports the PowerPC MPC5xx/8xx, PowerPC Nexus MPC55xx, ColdFire MCF5xxx, and ARM MAC7xxx architectures.

The user may select the target architecture by clicking the corresponding tab. Alternately, the user may select the target architecture through the File menu. **Figure 4-2** shows the selection.



**Figure 4-2: Target Class Selection via File Menu**



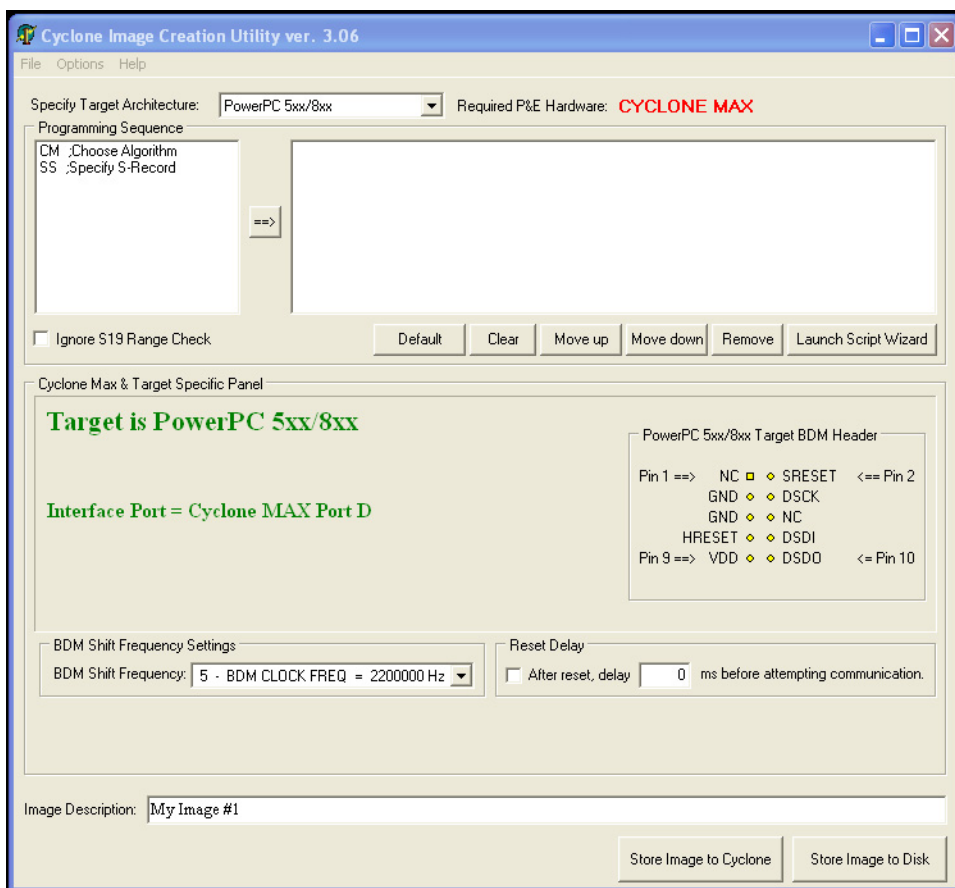
## 4.1.1 PowerPC MPC5xx/8xx

The user may configure the Cyclone MAX to operate on a PowerPC MPC5xx/8xx target by switching to the PowerPC option in the drop-down list.

Alternately, the user may select the PowerPC target through the File menu.

**Figure 4-3** shows the PowerPC specification configuration.

The target BDM header connections are shown on the right-hand side for user reference.



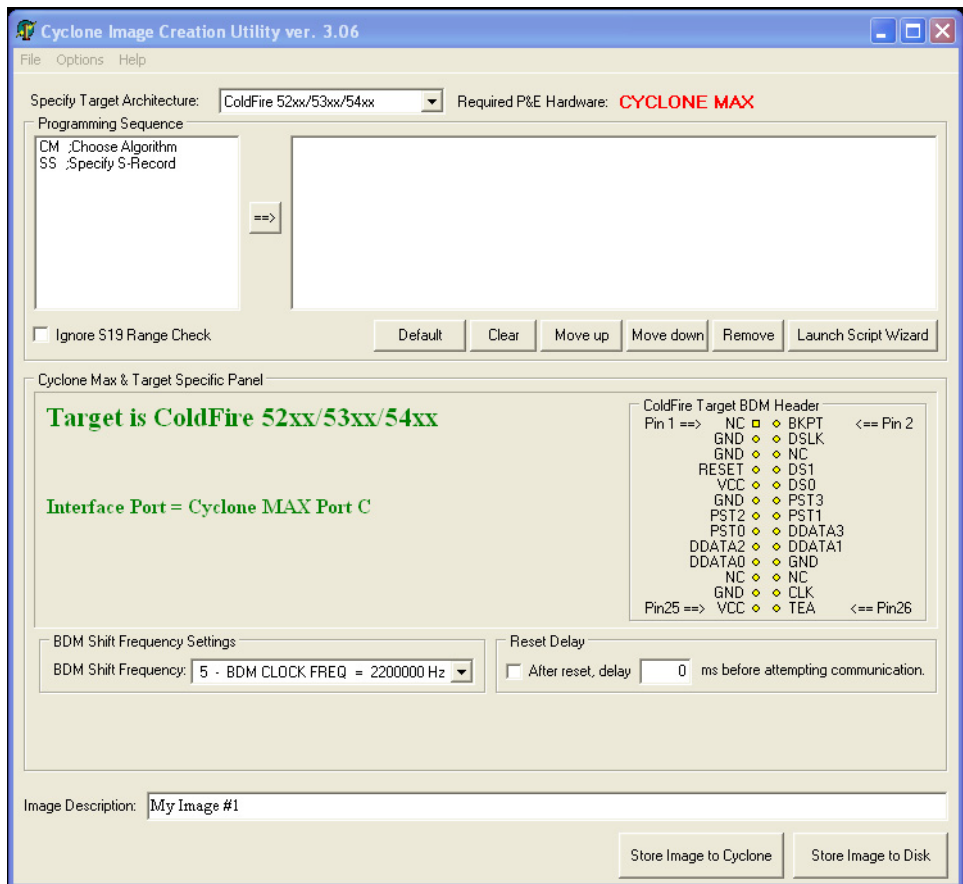
**Figure 4-3: PowerPC Settings**



#### 4.1.2 ColdFire MCF5xxx

The user may configure the Cyclone MAX to operate on a ColdFire MCF5xxx target by switching to the ColdFire MCF5xxx option in the drop-down list. Alternately, the user may select the ColdFire MCF5xxx target through the File menu.

**Figure 4-4** shows the ColdFire MCF5xxx specification configuration. The target BDM header connections are shown on the right-hand side for user reference.



**Figure 4-4: ColdFire Settings**

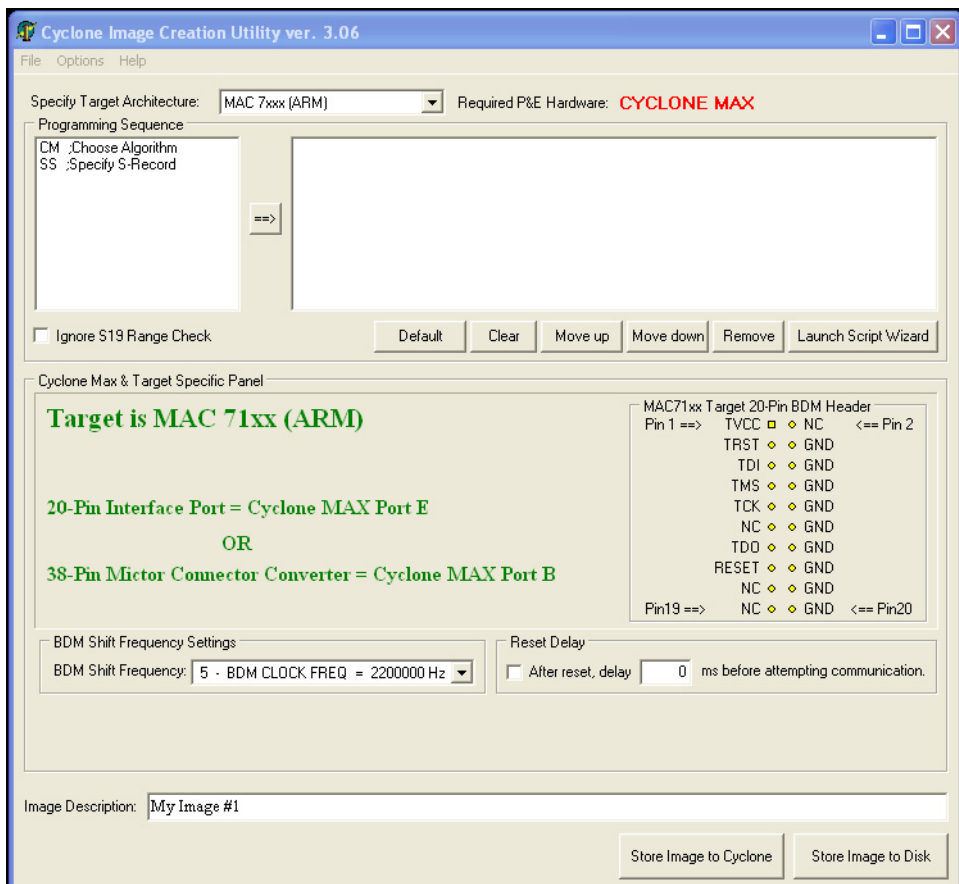


## 4.1.3

### ARM MAC7xxx

The user may configure the Cyclone MAX to operate on an ARM MAC7xxx target by switching to the ARM option in the drop-down list. Alternately, the user may select the ARM MAC7xxx target through the File menu.

**Figure 4-5** shows the ARM MAC7xxx specification configuration. The 20-pin target header connections are shown on the right-hand side for user reference.



**Figure 4-5: ARM Settings**

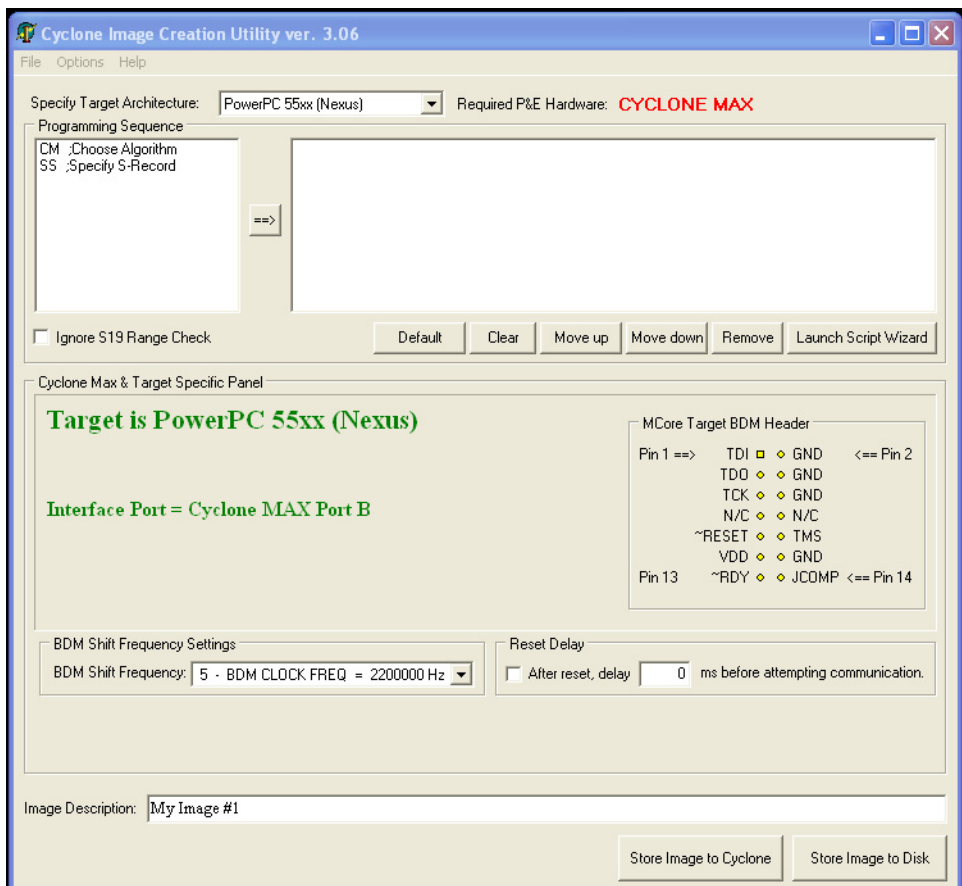
#### 4.1.4 PowerPC Nexus

The user may configure the Cyclone MAX to operate on a PowerPC Nexus target by switching to the PowerPC Nexus option in the drop-down list.

Alternately, the user may select the PowerPC Nexus target through the File menu.

**Figure 4-6** shows the PowerPC Nexus specification configuration.

The 14-pin target header connections are shown on the right-hand side for user reference



**Figure 4-6: PowerPC Nexus Settings**



---

## **4.2 BDM Shift Clock Delay Constant**

The BDM Shift Clock Delay Constant allows the user to set the BDM shift clock speed of the Cyclone MAX. The equation for determining the shifting frequency is:  $50000000 / (5 + 2 \cdot n)$ . This clock cannot generally exceed a div 6 of the processor bus frequency.

## **4.3 Specify Programming Algorithm and S-Record**

To specify the programming algorithm for the target, the user may simply select the CM (Choose Module) command from the Programming Sequence list (1) and add it to the programming steps (2) using the add button (3).

Similarly, to specify the S-Record to be programmed into the target, the user may simply select the SS (Specify S-Record) command from the Programming Sequence list (1) and add it to the programming steps (2) using the add button (3).

See **Figure 2-1** for a visual reference.

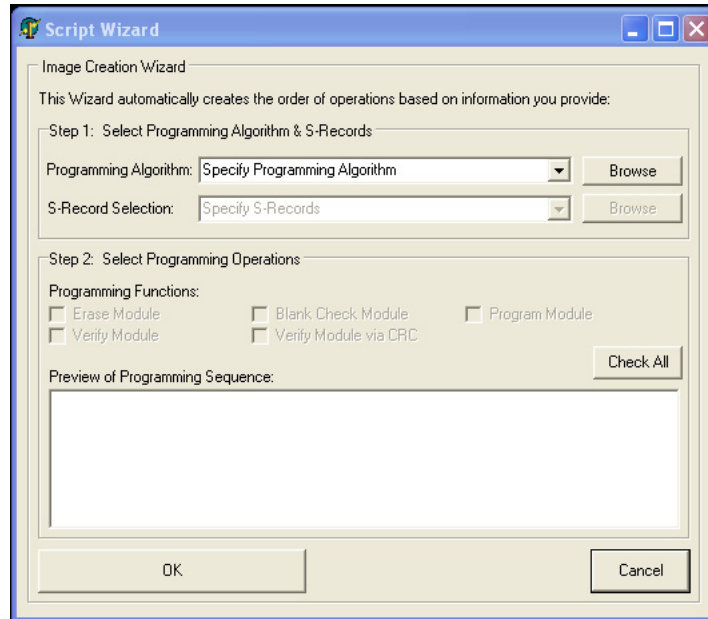
## **4.4 Base Address**

Depending on the programming algorithm, the user may be required to enter the base address for the module. If the programming algorithm requires a base address, the Module Base Address Settings will be displayed. The user must enter the correct base address. Otherwise, the default value (0) shown in the diagram will be used.

## **4.5 Script Wizard**

The Script Wizard is a convenient addition to the Cyclone Image Creation Utility. The Script Wizard allows the user to simply select the programming functions to be executed. The Script Wizard then orders these functions automatically.

To begin, click the Launch Script Wizard button on the right side of the Cyclone Image Creation Utility. This opens the Script Wizard.

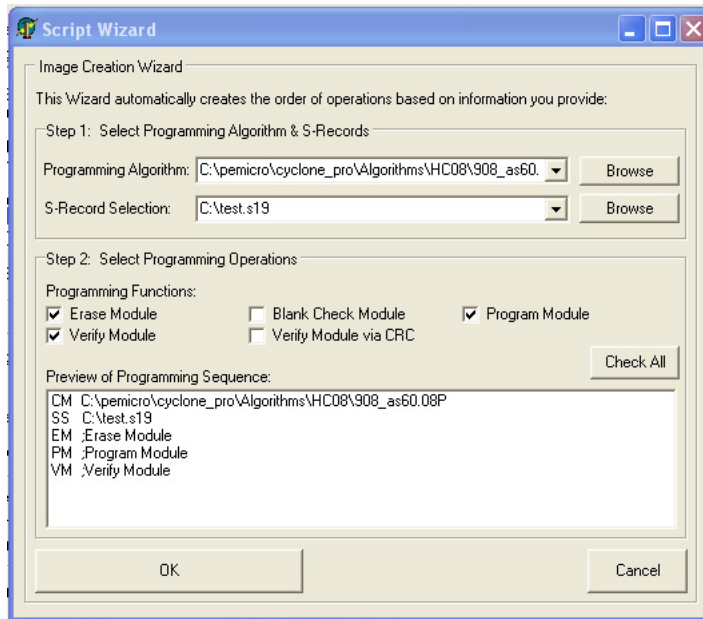


**Figure 4-7: Script Wizard Dialog**

The user must first select a programming algorithm and an S-Record. Click the corresponding Browse buttons to select a file, or use the drop-down list to choose a previously selected file.

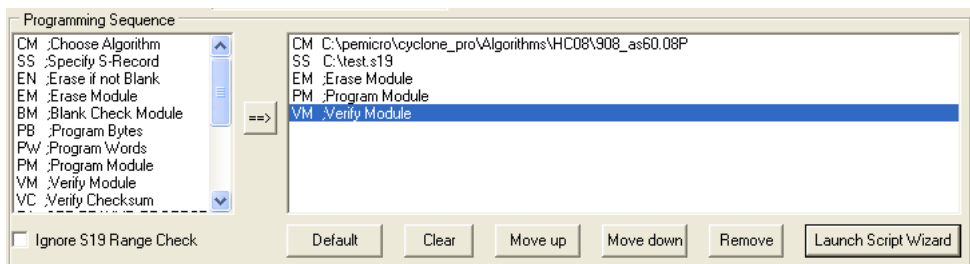
Once this is complete, the programming functions are enabled. At this point the user has only to select the programming steps that are required, with no further attention to the order of operations.

For example, in **Figure 4-8**, three programming functions have been selected and sequenced by the Script Wizard.



**Figure 4-8: Script Wizard Dialog - Programming Functions Selected**

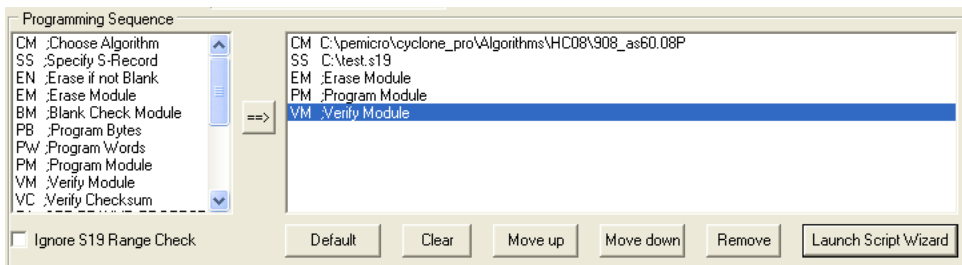
Upon completion, click OK and the script information will appear in the Specify Programming Script dialog.



**Figure 4-9: Results Of Script Wizard Dialog**



## 4.6 Programming Operations



**Figure 4-10: Programming Operations Dialog Section**

In the Programming Sequence field, the user may specify the algorithm, S-Record, and operations to be carried out.

### Choose Module

Presents a list of available programming files. Each programming file contains information on how to program a particular module. Usually, the name of the file indicates what kind of module it relates to.

### Specify S-Record

Asks for the name (and/or path) to a file of Freescale S-records to be used in programming or verifying a module. If the file is not found, an error message is given. The currently-selected file is shown in the S19 file selected window. The programmer accepts S1, S2, and S3 records. All other file records are treated as comments. If you do not specify a file-name extension, a default of .S19 is used. The programmer also supports ELF/Dwarf 2.0 object files.

Your S19 file may contain data for both EEPROM and flash. If you know that your S19 file contains the correct data, “Ignore S19 Range” may be checked. This will cause any out of range errors to be ignored.

### Erase If Not Blank

This command performs a blank check of the module and erases it if it is not blank.

### Erase Module

If “Erase Module” is specified, the Cyclone MAX will perform an “Erase Module” on the target device after entering the Monitor Mode or BDM mode.



### **Blank Check Module**

If “Blank Check Module” is checked, the Cyclone MAX will perform a “Blank Check Module” on the target device.

### **Program Bytes**

Prompts for a starting address, which must be in the module. You are then asked to enter a hexadecimal byte to be programmed into the current location. Clicking the OK button will automatically advance to the next data byte location.

### **Program Words**

Prompts for a starting address, which must be in the module. You are then asked to enter a hexadecimal word to be programmed into the current location. Clicking the OK button will automatically advance to the next data word location.

### **Program Module**

For this command to work, you must have previously selected an S-record file.

### **Verify Module**

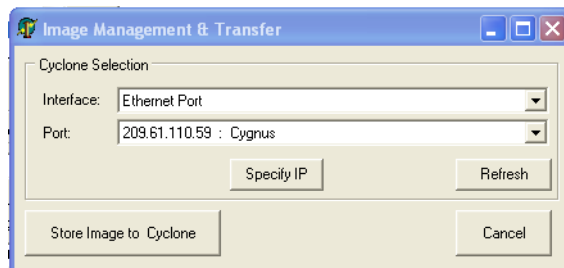
For this command to work, you must have previously selected an S-record file.

### **Verify Checksum**

This command verifies the module content via a CRC calculation.

## **4.7 Store Image To Cyclone**

“Store Image to Cyclone” allows the current configuration to be programmed into the Cyclone MAX. The Cyclone MAX will then be ready for operations.



**Figure 4-11: Image Management And Transfer Dialog**



---

The Interface drop-down list allows the user to select either a serial, USB, or Ethernet communication interface. The Port drop-down list allows the user to select from one of the Cyclones available on that interface. In the case of a Cyclone present on a different network (i.e., not displayed automatically in the Port drop-down list), the user may specify its IP address by using the Specify IP button.

“Store Image to Cyclone” will then store the image on the selected Cyclone.

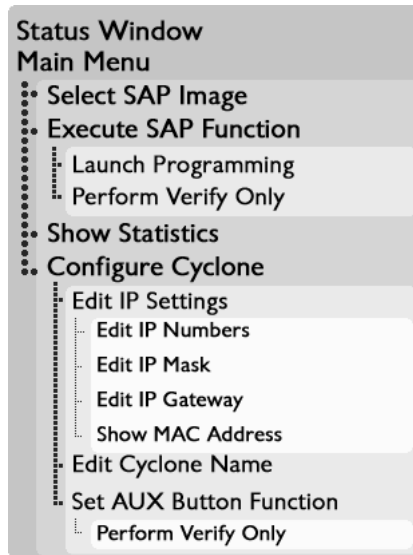
#### **4.8 Save Image/Cyclone Configuration**

“Store Image To Disk” allows the current configuration to be saved onto the hard drive. The image can then be transferred to the Cyclone MAX via the Manage Images Utility.

“Save Cyclone Configuration,” in the file menu, allows the user to save the configuration into a file, which may be used for future reference, e.g., comparing the Cyclone MAX contents with the file to see if they are the same.

#### **4.9 Configuration Via LCD Menu**

The following section describes configuration of the Cyclone MAX using the LCD menus. **Figure 4-12** shows an overview of the menu structure.



**Figure 4-12: Overview Of Cyclone Menu Structure**

## 4.9.1 Status Window

```
Cyclone MAX, v.7.23.00
IP: 200.0.0.0
Name: Manufacturing
PC: none
Total Num. of Imgs: 2
IMG1: MCF5272
Idle....
Programming completed.
```

**Figure 4-13: Status Window**

The status window appears when the Cyclone MAX is powered on. This lists the following information:

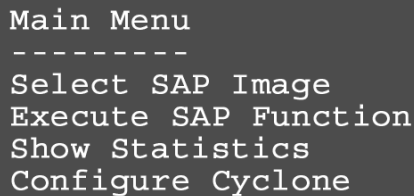
1. The firmware version of the MAX.
2. The IP address assigned to the MAX.



3. The name assigned to the MAX.
4. Name of the PC connected to the MAX.
5. The number of programming images in the MAX's memory.
6. The name of the selected programming image.
7. Current status.
8. Results of the last operation performed.

Hit the MENU/[SELECT] button on the Cyclone MAX to enter LCD menu mode.

#### **4.9.2 Main Menu**



```
Main Menu
-----
Select SAP Image
Execute SAP Function
Show Statistics
Configure Cyclone
```

**Figure 4-14: Main Menu**

The Main Menu is accessible by pressing the MENU/[SELECT] button when the status window is displayed. The Main Menu contains the following selections:

##### **4.9.2.1 Select SAP Image**

Select SAP Image brings up a display listing the images that are stored in the memory of the Cyclone MAX. You may select the appropriate image by using the Up/Down arrows to highlight it, and then pressing the "Select" button. The image name shown is the one specified in the



Cyclone configuration utility when programming the image to the Cyclone.

```
IMG1: ColdFire MCF5272
IMG2: PPC561
```

**Figure 4-15: Select SAP Image**

#### **4.9.2.2 Execute SAP Function**

The Execute SAP Function menu selection is discussed in **Section 5.2 - Operation Procedure via LCD Menu**. Please refer to that section for additional menu information.

#### **4.9.2.3 Show Statistics**

The Show Statistics menu selection is discussed in **Section 5.2 - Operation Procedure via LCD Menu**. Please refer to that section for additional menu information.

#### **4.9.2.4 Configure Cyclone**

```
Configure Cyclone
-----
Edit IP Settings
Edit Cyclone Name
Set AUX Button Func
```

**Figure 4-16: Configure Cyclone**

Configure Cyclone brings up a submenu with three options from which to choose. Selecting Edit IP Settings brings up a submenu with four options:



## Configure Cyclone: Edit IP Settings

```

Edit IP Settings
-----
Edit IP Numbers
Edit IP Mask
Edit IP Gateway
Show MAC Address

```

**Figure 4-17: Configure Cyclone: Edit IP Settings**

### Edit IP Settings: Edit IP Number

Edit IP Number allows the user to set an IP number for the Cyclone MAX. The current IP number is displayed on the second line. Use the Up/Down buttons to scroll through the characters. To select a character, hit the Select button. When you are finished, scroll through the characters until you reach the -> (right-arrow) character. Selecting this character will complete the process. Press the Cancel button at any time to leave the IP number as is and return to the Main Menu.

```

Edit IP Number
(200.0.0.0)
-----
0
Up/Down to scroll
Select to choose
Choose -> when done

```

**Figure 4-18: Edit IP Settings: Edit IP Number**

### Edit IP Settings: Edit IP Mask

Edit IP Mask allows the user to set an IP Mask for the Cyclone MAX. The current IP Mask is displayed on the second line. Use the Up/Down buttons to scroll through the characters. To select a character, hit the Select button. When you are finished, scroll through the characters until you reach



the -> (right-arrow) character. Selecting this character will complete the process. The default IP mask is 255.255.255.0.

```
Edit IP Mask
(0.0.0.0)
-----
0
Up/Down to scroll
Select to choose
Choose -> when done
```

**Figure 4-19: Edit IP Settings: Edit IP Mask**

#### **Edit IP Settings: Edit IP Gateway**

Edit IP Gateway allows the user to set the IP Gateway for the Cyclone MAX. The current IP Gateway is displayed on the second line. Use the Up/Down buttons to scroll through the characters. To select a character, hit the Select button. When you are finished, scroll through the characters until you reach the -> (right-arrow) character. Selecting this character will complete the process.

```
Edit IP Gateway
(209.10.112.201)
-----
0
Up/Down to scroll
Select to choose
Choose -> when done
```

**Figure 4-20: Edit IP Settings: Edit IP Gateway**

#### **Edit IP Settings: Show MAC Address**

Show MAC Address displays the current MAC address for





the Cyclone MAX.

```
Show MAC Address
-----
10-22-33-44-55-0
```

**Figure 4-21: Edit IP Settings: Show MAC Address**

### **Configure Cyclone: Edit Cyclone Name**

Edit Cyclone Name allows the user to set the name for the Cyclone MAX. The current name is displayed on the second line. Use the Up/Down buttons to scroll through the characters. To select a character, hit the Select button. When you are finished, scroll through the characters until you reach the -> (right-arrow) character. Selecting this character will complete the process. Press the Cancel button at any point to leave the name as is and return to the Main Menu.

```
Edit Cyclone Name
(Manufacturing)
-----
A
Up/down to scroll
Select to choose
Choose -> when done
```

**Figure 4-22: Configure Cyclone: Edit Cyclone Name**

### **Configure Cyclone: Set AUX Button Func**

Set AUX Button Func allows the user to assign a function to the AUX button of the Cyclone MAX. If more than one choice is available, highlight the function that you wish to assign to the AUX button and



press the Select button to choose it.

```
Set AUX Button Func
-----
Perform Verify Only
```



**Figure 4-23: Configure Cyclone: Set AUX Button Func**

## 5 STAND-ALONE PROGRAMMER MANUAL CONTROL

The Cyclone MAX must be configured before it may serve as a Stand-Alone Programmer for ColdFire (52xx, 53xx, 54xx), PowerPC (5xx, 8xx), PowerPC Nexus (55xx), and ARM Nexus (MAC7xxx) targets. The user may manually control the Cyclone MAX via the buttons/LEDs, LCD menu, or via PC software. The target power management schemes remain the same for each control method.

### 5.1 Via Cyclone MAX Buttons

There are five (5) buttons on the top of the Cyclone MAX which are used for stand-alone programming and to navigate the LCD menus. They are specified as follows.

<b><u>Button</u></b>	<b><u>Function</u></b>
START / 	Start executing the tasks pre-configured into the Cyclone MAX.
AUX / 	<p><b>Menu Mode:</b> Navigate upward in LCD menu.</p> <p>Perform auxiliary function (stand-alone verification).</p> <p><b>Menu Mode:</b> Navigate downward in LCD menu.</p>
MENU / [SELECT]	<p>Displays LCD Menu (enters Menu Mode).</p> <p><b>Menu Mode:</b> Select highlighted item in LCD menu.</p>
CANCEL	Cancel the tasks being executed and go back to the standby state.
RESET	Hardware reset of the Cyclone MAX.

Pushing the START button launches the stand-alone programming image which is currently selected (i.e., programming is a one-click operation).

#### 5.1.1 Cyclone MAX LED Indicators

The Cyclone MAX has eight (8) LEDs to indicate the current operation stage:

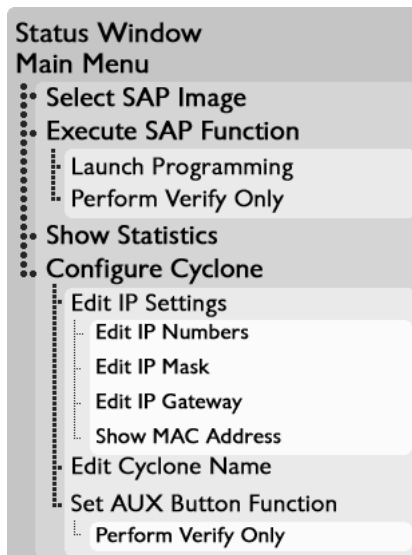
1. Power – Indicates power has been supplied to Cyclone MAX
2. Ready – The Cyclone MAX is ready to perform programming functions
3. Image – Indicates the presence of one or more images on board.
4. Erase – Erasing of target flash is in progress
5. Program – Programming of target flash is in progress



6. Verify – Verifying of target flash is in progress
7. Error – An error occurred during programming
8. Success – Programming functions completed successfully

## 5.2 Operation Procedure via LCD Menu

Rev. B of the Cyclone MAX may be operated by making selections from the LCD menu. This section describes the layout of the menus and the functions that each may be used to perform.



**Figure 5-1: Overview Of Cyclone Menu Structure**



### 5.2.1 Status Window

```
Cyclone MAX, v.7.23.00  
IP: 200.0.0.0  
Name: Manufacturing  
PC: none  
Total Num. of Imgs: 2  
IMG1: MCF5272  
Idle...  
Programming completed.
```

**Figure 5-2: Status Window**

The status window appears when the Cyclone MAX is powered on. This lists the following information:

1. The firmware version of the MAX.
2. The IP address assigned to the MAX.
3. The name assigned to the MAX.
4. Name of the PC connected to the MAX.
5. The number of programming images in the MAX's memory.
6. The name of the selected programming image.
7. Current status.
8. Results of the last operation performed.

### 5.2.2 Main Menu

```
Main Menu  
-----  
Select SAP Image  
Execute SAP Function  
Show Statistics  
Configure Cyclone
```

**Figure 5-3: Main Menu**



The Main Menu is accessible by pressing the “Menu” button when the status window is displayed. The Main Menu contains four selections. This section contains information on Execute SAP Function and Show Statistics. For information on Select SAP Image and Configure Cyclone, please refer to **Section 4.9 - Configuration Via LCD Menu**.

## 5.2.2.1 Execute SAP Function

Execute SAP Function presents two Stand-Alone Programming functions that you may execute by highlighting the function that you wish to perform and pressing the “Select” button.

```
Execute SAP Function
-----
Launch Programming
Perform Verify Only
```

**Figure 5-4: Execute SAP Function**

## 5.2.2.2 Show Statistics

The fourth line (PCIP:) displays the IP address of the last PC to control the Cyclone MAX. The other categories listed are for future use and are not currently implemented.

```
Show Statistics
-----
SerNum:
NumProgs:
ProgsLeft:
PCIP: 0.0.0.0
```

**Figure 5-5: Show Statistics**



---

## **6 STAND ALONE PROGRAMMER AUTOMATED CONTROL**

There are various means of controlling the Cyclone MAX in an automated environment. Included with the Cyclone MAX are:

- 1) The CYCLONEMAX\_LAUNCH utility, which allows control via script files.
- 2) A .DLL which may be used by a host application to control the Cyclone MAX via Ethernet.

In addition, P&E offers the Multi-Cyclone Automated Control SDK separately, which allows control of multiple Cyclones in parallel via serial, USB, or Ethernet.

### **6.1 Automated .EXE Control Of The Cyclone**

The user can use the Cyclone MAX in an automated production environment via PC software called CYCLONEMAX\_LAUNCH.EXE. This utility may be invoked in a batch file to carry out pre-configured operations in the Cyclone MAX.

#### **6.1.1 Startup**

- a) Connect the Cyclone MAX to the PC via Serial Port, USB Port, or Ethernet.
- b) Connect the Cyclone MAX to the target system.
- c) Power up the PC and the target system.
- d) Run the software from the DOS prompt. Allowed command-line parameters are:

**CYCLONEMAX\_LAUNCH [port=y] [image=n]**

- where the value of **y** is:

USB1 - USB Device #1

USB2 - USB Device #2

USB3 - USB Device #3

USB4 - USB Device #4

COM1 - Serial Port 1

COM2 - Serial Port 2

COM3 - Serial Port 3



COM4 - Serial Port 4

### - Ethernet IP address ###. (each # symbol represents a decimal number between 0 and 255).

- and where n designates the index of the image residing on the Cyclone MAX, ranging in value from 1 to 255.

## 6.1.2 Command-Line Parameter Examples

CYCLONEMAX\_LAUNCH Port=COM1

Cyclone MAX is connected to the PC via the COM1 port.

CYCLONEMAX\_LAUNCH Port=USB1

Cyclone MAX is connected to the PC via the USB1 port.

CYCLONEMAX\_LAUNCH Port=USB1 Image=1

Start the first programming image on the Cyclone which is connected to USB1 port.

CYCLONEMAX\_LAUNCH Port=209.61.110.251

Cyclone MAX is connected to a network with IP address of 209.61.110.251

CYCLONEMAX\_LAUNCH Port=209.61.110.251 Image=3

Start the third programming image on the Cyclone which is connected to the network with the IP address of 209.61.110.251.

## 6.1.3 DOS Error Returns

DOS error returns are provided so they may be tested in .BAT files. The error codes used are:

### Application handling related error codes:

190: No command line parameters specified.

191: Specified COM port is not available.

192: Specified USB port is not available. Please make sure the USB port is





available, and the USB cable is connected.

193: Specified Ethernet IP address is incorrect.

199: The Cyclone MAX device is not ready. Please check power and connections.

**Execution related error codes:**

\$0001: Unable to detect target communication speed.

\$0002: Unable to enter debug mode.

\$0003: Operation cancelled by user.

\$0004: Error writing data byte block.

\$0005: Error writing data byte block.

\$0006: Error during execution.

\$0007: Error enabling module.

\$0008: Error enabling module.

\$0009: Error enabling module.

\$000A: Error testing target timing.

\$1001: Blank\_Check\_Word Algorithm is not supported.

\$1002: Blank\_Check\_Byte Algorithm is not available.

\$1003: Error during blank checking device using blank\_check\_byte algorithm.

\$1004: Error during blank checking device using blank\_check\_word algorithm.

\$2003: Error during erasing device.

\$2004: Error during erasing device.

\$3001: Program\_Word Algorithm is not supported.

\$3002: Program\_Byte Algorithm is not available.

\$3003: Error during programming device.

\$5003: Error during verifying module.

\$5004: Error during verifying range.

\$6003: Error during user functions.

\$6004: Error during user functions.



---

#### **6.1.4 Sample Batch File**

Here is an example of a way to call the CYCLONEMAX\_LAUNCH utility and test its error code return in a simple batch file. Sample batch files are given for both Windows NT/2000/2003/Vista and Windows 95/98/ME/XP.

##### **Windows NT/2000/2003/Vista:**

```
CYCLONEMAX_LAUNCH PORT=COM1
if errorlevel 1 goto bad
goto good
:bad
ECHO BAD BAD BAD BAD BAD BAD BAD BAD
:good
ECHO done
```

##### **Windows 95/98/ME/XP:**

```
START /W CYCLONEMAX_LAUNCH PORT=USB1
if errorlevel 1 goto bad
goto good
:bad
ECHO BAD BAD BAD BAD BAD BAD BAD BAD
:good
ECHO done
```

#### **6.1.5 Additional Information**

For more information on CYCLONEMAX\_LAUNCH and how to construct detailed executions for your own devices, please contact P&E.

#### **6.2 Automated .DLL Control Of The Cyclone**

Another automated means of controlling a Cyclone PRO or Cyclone MAX unit is via a Windows .DLL. The .DLL allows a host application to connect to a Cyclone unit via Ethernet and control stand-alone programming operations.

Included with the Cyclone MAX is a .DLL which allows control of a Cyclone unit. Example code and documentation is included which demonstrates use of



the .DLL with a Microsoft Visual C/C++ application and a Borland Delphi application, although the .DLL is not limited to these development environments. Any application which can load and use a .DLL can be used for automated control of a single Cyclone unit.

For controlling multiple Cyclone units in parallel, the required .DLL is available in the “Multi-Cyclone Automated Control SDK.” Please visit [www.pemicro.com](http://www.pemicro.com) for more information about this product.

### 6.2.1 Connecting/Disconnecting from the Cyclone

In order to control a Cyclone PRO or Cyclone MAX unit from a host computer, the host computer must initiate a connection to a Cyclone unit. The *connect\_to\_cyclonepromax\_by\_ip* call opens a connection to the Cyclone unit via Ethernet. The return handle is used by the calling application while controlling the Cyclone unit. The host PC should disconnect from the Cyclone unit when the programming process has completed.

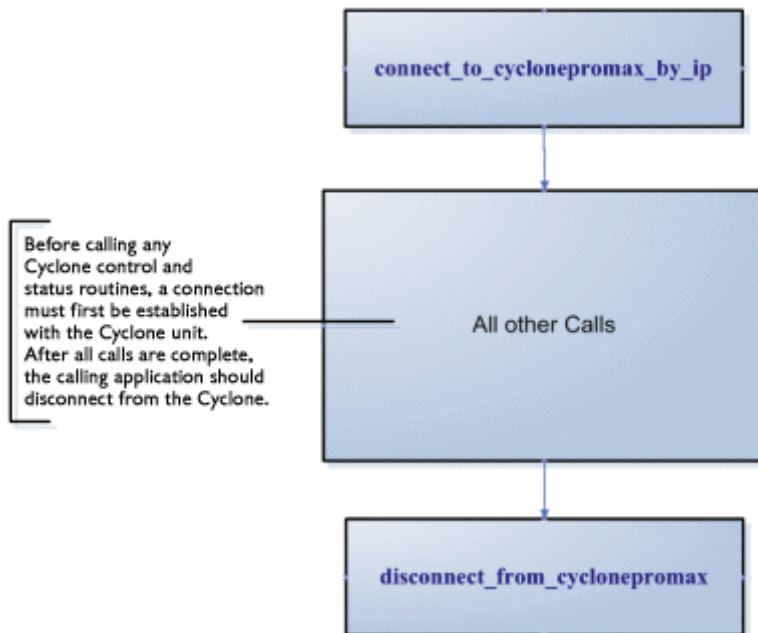


Figure 6-6: Call Sequence Detail



---

### 6.2.1.1      **Call: connect\_to\_cyclonepromax\_by\_ip**

**C/C++ Prototype:**

```
unsigned int connect_to_cyclonepromax_by_ip(char *port_identifier);
```

**Description:**

This function opens a session with a Cyclone PRO or Cyclone MAX by its IP address. The *port\_identifier* parameter is a pointer to a null-terminated character string in the format xxx.xxx.xxx.xxx, where xxx = 0..255. The return value is the handle to the open Cyclone PRO or Cyclone MAX, which must be passed to other routines as a parameter. A return value of 0 indicates a failure to connect to the Cyclone PRO or Cyclone MAX at that address.

### 6.2.1.2      **Call: disconnect\_from\_cyclonepromax**

**C/C++ Prototype:**

```
bool disconnect_from_cyclonepromax(unsigned long cyclonepromaxhandle);
```

**Description:**

When processing is complete, the session with the Cyclone PRO or Cyclone MAX should be terminated. Calling this routine with the appropriate handle terminates the session and allows opening of a different Cyclone PRO/MAX unit.

## 6.2.2      **Controlling Cyclone Programming**

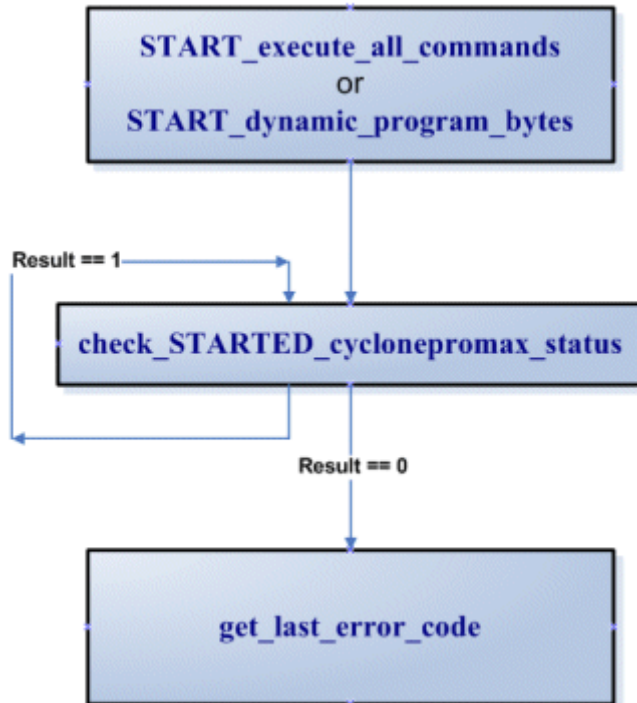
There are two main programming commands which start the programming process on a Cyclone.

The *START\_execute\_all\_commands* call is used to launch a programming image which is stored on the Cyclone unit and has been pre-configured in terms of all the steps to be accomplished during programming. Once started, the *check\_STARTED\_cyclonepromax\_status* routine may be called until it indicates that the Cyclone programming has completed. At this point the *get\_last\_error\_code* routine should be called to determine if an error has occurred or if the programming was successful.

Another command, *START\_dynamic\_program\_bytes*, allows data which has not been pre-configured into the Cyclone unit to be programmed to the target. This is useful for programming the date, a serial number, MAC address, or



other dynamically generated data to the target. The *check\_STARTED\_cyclonepromax\_status* routine is used to wait for programming completion.



**Figure 6-7: Call Sequence Detail**

#### 6.2.2.1 Call: **START\_execute\_all\_commands**

##### **C/C++ Prototype:**

```
bool START_execute_all_commands(unsigned long cyclonepromaxhandle,
unsigned long imagenumber);
```

##### **Description:**

A Cyclone unit may have several independent programming images in non-volatile memory. A programming image contains the programming algorithms,



binary data, and programming sequence. Calling this routine instructs the Cyclone unit to start execution of a particular image. If the Cyclone unit only holds one image, the *imagenumber* should be set to a "1". If the Cyclone unit holds multiple programming images, the valid values for image number range from one up to the number of images in the unit. A result of true indicates that the image has started the programming process. Call the *Check\_STARTED\_CycloneProMax\_Status* command to wait for completion.

## 6.2.2.2 Call: **START\_dynamic\_program\_bytes**

### **C/C++ Prototype:**

```
bool START_dynamic_program_bytes(unsigned long cyclonepromaxhandle,  
unsigned long target_address, unsigned short data_length, char *buffer);
```

### **Description:**

Sometimes, in addition to the large amount of static data being programmed into a target from the Cyclone, it will be advantageous for the calling application to program small sections of unique data dynamically. Examples of this include: date/time, serial number, MAC addresses, and lot numbers. This routine allows the user to specify an address, data length, and pointer to a buffer of data to be programmed into the target. It is valid to be called only after a programming image has been programmed into the target (once *START\_execute\_all\_commands* has completed). A result of true indicates that the image has started the programming process. Call the *Check\_STARTED\_CycloneProMax\_Status* command to wait for completion.

## 6.2.2.3 Call: **check\_STARTED\_cyclonepromax\_status**

### **C/C++ Prototype:**

```
unsigned int check_STARTED_cyclonepromax_status(unsigned long  
cyclonepromaxhandle);
```

### **Description:**

Checks to see if the Cyclone PRO or Cyclone MAX has completed a programming operation started with either the *START\_execute\_all\_commands* or *START\_dynamic\_program\_bytes* routines. The results are:

1 = Currently Programming



0 = Completed (with or without error)

After this call returns the completed value, *get\_last\_error\_code* should be called to determine the programming result.

#### 6.2.2.4 Call: *get\_last\_error\_code*

**C/C++ Prototype:**

unsigned short *get\_last\_error\_code*(unsigned long cyclonepromaxhandle);

**Description:**

Returns the last error code of the Cyclone PRO or Cyclone MAX.

0 = No Error.

#### 6.2.2.5 Call: *get\_last\_error\_addr*

**C/C++ Prototype:**

unsigned int *get\_last\_error\_addr*(unsigned long cyclonepromaxhandle);

**Description:**

If the *get\_last\_error\_code* call returned a non-zero value (i.e., there was an error), this routine can be used to query the address where the error occurred.

### 6.2.3 Configuration / Image Maintenance

These calls are used to perform operations on the Cyclone units such as: reset, check image, erase image, and add new image to the Cyclone. Due to security concerns, the calling application will not read or upload images from the Cyclone unit. All images which are to be verified or downloaded onto the Cyclone must have been generated by the Cyclone Configuration utility.

#### 6.2.3.1 Call: *reset\_cyclonepromax*

**C/C++ Prototype:**

bool *reset\_cyclonepromax*(unsigned long cyclonepromaxhandle, unsigned long reset\_delay\_in\_ms);

**Description:**



---

Used to reset the Cyclone PRO/MAX hardware. The delay should be at least 2500ms. This routine does not need to be called by the calling application and exists for historical reasons.

### 6.2.3.2 Call: `get_image_description`

#### **C/C++ Prototype:**

```
char *get_image_description(unsigned long cyclonepromaxhandle, unsigned long imagenumber);
```

#### **Description:**

Returns the description string of the selected image. For a Cyclone unit with only one programming image, the *imagenumber* value should be set to 1. If the Cyclone unit holds multiple programming images, the valid values for image number range from one up to the number of images in the unit. The result is a pointer to a null-terminated string containing the image description. This description is valid until the next call into the .DLL is made.

### 6.2.3.3 Call: `compare_image_with_file`

#### **C/C++ Prototype:**

```
bool compare_image_with_file(unsigned long cyclonepromaxhandle, char *aFile, unsigned long imagenumber);
```

#### **Description:**

Allows the calling application to compare a stand-alone programming image stored on the PC with an image stored in the Cyclone. This can be used to validate that a programming image in the PC is exactly the same as the one on the Cyclone. The Cyclone images automatically self-verify, but this may be useful to make sure that the version of the image in the Cyclone is what the calling application is expecting. For a Cyclone unit with only one programming image, the *imagenumber* value should be set to 1. If the Cyclone unit holds multiple programming images, the valid values for image number range from one up to the number of images in the unit. A “true” result indicates that the images are the same. The *aFile* pointer should point to a null terminated filename, which is the full pathname of the file.





---

#### 6.2.3.4      **Call: erase\_all\_cyclone\_images**

**C/C++ Prototype:**

```
bool erase_all_cyclone_images(unsigned long cyclonepromaxhandle);
```

**Description:**

This call erases all stand-alone programming images on the Cyclone unit. Generally this call is made before a new image is placed onto the Cyclone by the *add\_image\_to\_cyclone* call. A “true” result indicates that the erasure was successful.

#### 6.2.3.5      **Call: add\_image\_to\_cyclone**

**C/C++ Prototype:**

```
unsigned int add_image_to_cyclone(unsigned long cyclonepromaxhandle,  
char *aFile);
```

**Description:**

This call adds the stand-alone programming image specified with the *aFile* parameter into the Cyclone’s onboard memory. This image must have been created with the Cyclone configuration utility. This call is useful for adding more images into the Cyclone. The Cyclone has limited onboard memory, so the user may wish to erase existing images with the *erase\_all\_cyclone\_images* command before calling this command. The command will return the image number that was just added, or else return a “0” if the operation failed.

#### 6.2.3.6      **Call: toggle\_power\_no\_background\_entrance**

**C/C++ Prototype:**

```
bool toggle_power_no_background_entrance(unsigned long  
cyclonepromaxhandle);
```

**Description:**

Causes the RESET line of the target processor to be driven low, then pulled up high to enter normal operation mode.



## 6.2.3.7 Call: `set_local_machine_ip_number`

### C/C++ Prototype:

```
void set_local_machine_ip_number(char *ip_number);
```

### Description:

If a machine has more than one network card, this routine may be called to indicate the IP address of the network card which should be used during communications. This should be called prior to calling any other routines.

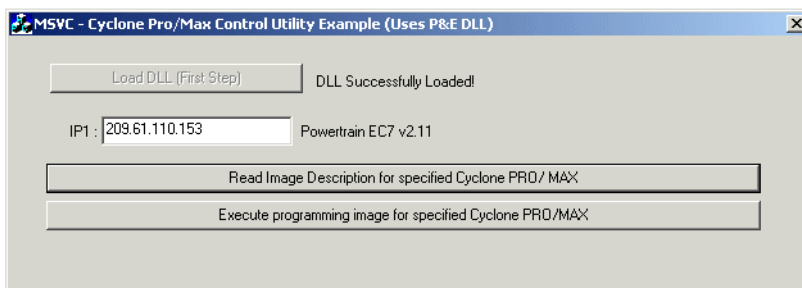
## 6.2.4 DLL Information, Interface Code, and Examples

The documented interface routines are available in the CYCLONEPROMAX\_SAP\_CONTROL.DLL library included in the software install. This .DLL is callable from almost any 32-bit Windows development environment. The method of calling the .DLL varies for different compilers, so .DLL interface code and sample applications are provided for each of the following compilers:

Microsoft Visual C++ 5.0+ - Visual MFC Application

Borland Delphi 4.0+ (Pascal) - Visual Application

The sample applications come with project and workspaces defined for ease of use. Just open the project/workspace in your compiler and you will be able to build the sample application without any modifications.



**Figure 6-8: Cyclone PRO/MAX Control Utility Example**

The sample applications come pre-compiled with ICONS, so you can run



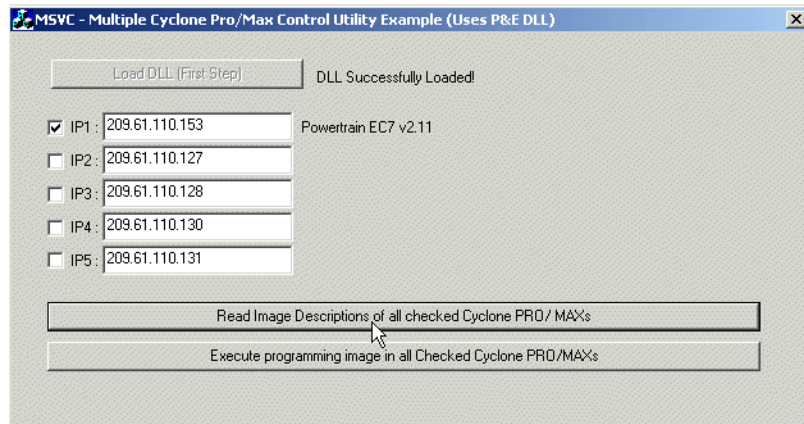
them before jumping into the code. The callable interface routines are defined in:

INSTALLDIR\msvc50\cyclonepromax\_sap\_control.h      - For MSVC 5.0+  
 INSTALLDIR\Delphi20\cyclonepromax\_sap\_control.pas   - For Delphi 4.0+

If you add the above-mentioned interface file (and associated .CPP file for the C++ compilers) to your own application, you will be able to directly access all the Cyclone control routines. Remember that the CYCLONEPROMAX\_SAP\_CONTROL.DLL file has to be in the executable's directory or system directory.

### 6.3 Control Of Multiple Cyclone Units

The utilities included with the Cyclone allow the unit to be updated and controlled in an automated fashion. The user can change the image in the Cyclone and execute the programming images resident on the unit under control from a host computer, or without a computer at all. These utilities/drivers focus on controlling a single Cyclone unit.



**Figure 6-9: Multiple Cyclone PRO/MAX Control Utility Example**

However, it may also be desirable to control multiple Cyclone units simultaneously from a host PC. Since the Cyclone is a self-contained computer system, including all programming algorithms, data, and processing



---

power necessary for programming, the host PC only needs to provide minimal control information. This means that an enormous number of Cyclone units may be controlled from a single PC. P&E offers a software development kit, the Multi-Cyclone Automated Control SDK, which is designed to provide this control information. This kit consists of the following components:

1. A .DLL which allows control of up to 999 Cyclone units in parallel.
2. An .EXE which allows control of up to 999 Cyclone units in parallel.
3. A UDP packet specification for controlling the Cyclone via Ethernet packets.
4. A serial packet specification for controlling the Cyclone via RS232 packets.

This SDK (software development kit) supports both the Cyclone PRO and the Cyclone MAX programmers, meaning that any supported Freescale devices can be programmed in parallel, even if they are different devices with different data. More information may be found at [www.pemicro.com](http://www.pemicro.com).



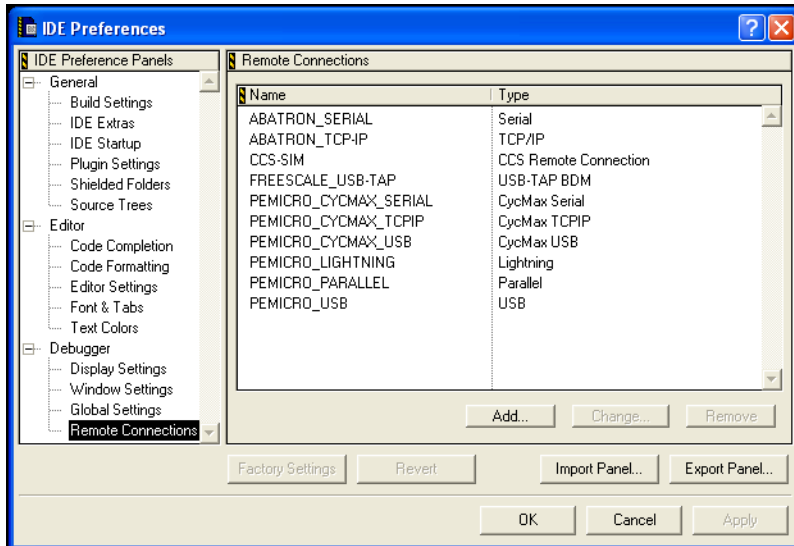
## 7 PC-HOSTED DEBUG/PROGRAMMING SOFTWARE

Software options for interactively programming and debugging PowerPC MPC5xx/8xx, ColdFire MCF5xxx, ARM MAC7xxx MCUs from the PC are sold separately by P&E Microcomputer Systems ([www.pemicro.com](http://www.pemicro.com)). Descriptions of the functionality of some of these applications are listed below.

### 7.1 Freescale Codewarrior

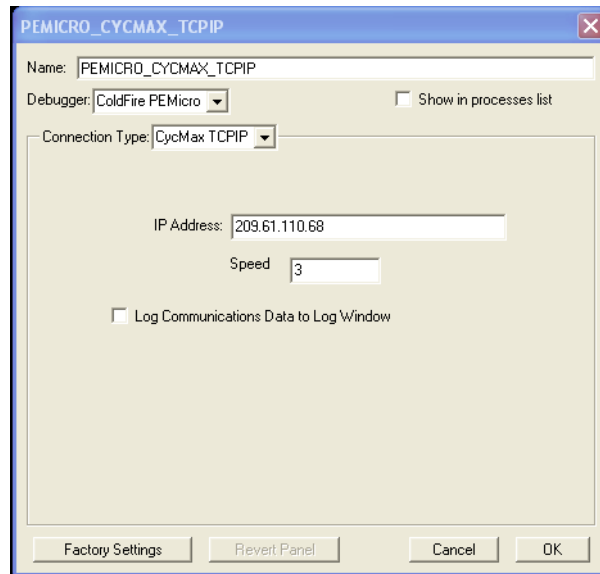
#### 7.1.1 Connecting the Cyclone Max through CodeWarrior

Customers using the Freescale CodeWarrior Development Studio (version 5.7 and above) will now have the capability to interface to target hardware using the Cyclone Max. In order to configure CodeWarrior to do so, the user can select the Preferences option from the Edit menu. This brings up the following dialog box:



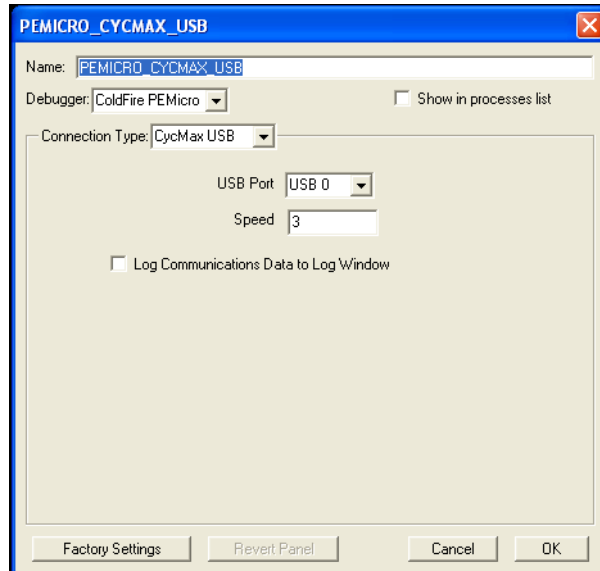
**Figure 7-1: IDE Preferences Dialog**

and the user can then select and configure one of Cyclone Max's Serial, USB, or Ethernet ports.



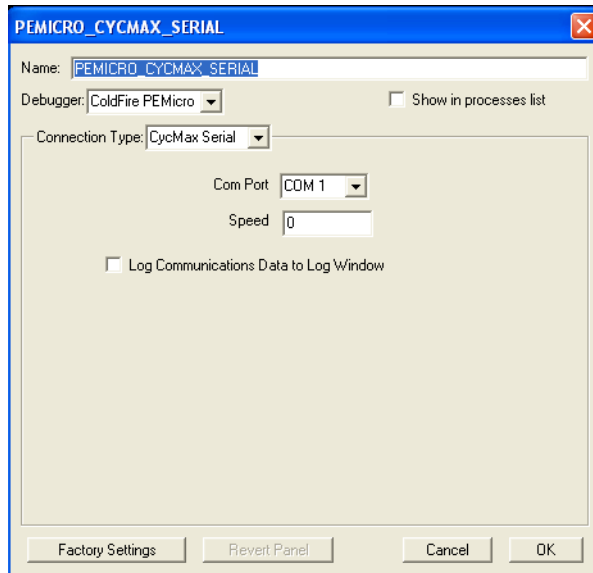
**Figure 7-2: Codewarrior Cyclone MAX Ethernet Configuration**

The image above displays the configuration settings for the Ethernet port of the Cyclone Max. The user can set the IP number, and the BDM Frequency Speed to interface to the target. The higher this value is, the slower the BDM Frequency will be, and it is recommended that for slower targets a higher value be selected. For greater detail on the BDM Frequency Speed please refer to the CodeWarrior Help documentation.



**Figure 7-3: Codewarrior Cyclone MAX USB Port Configuration**

The image above displays the configuration settings for the USB port of the Cyclone Max. The user can select one of multiple Cyclone devices which are connected to the PC, in case more than one Cyclone Max is attached via the USB port.



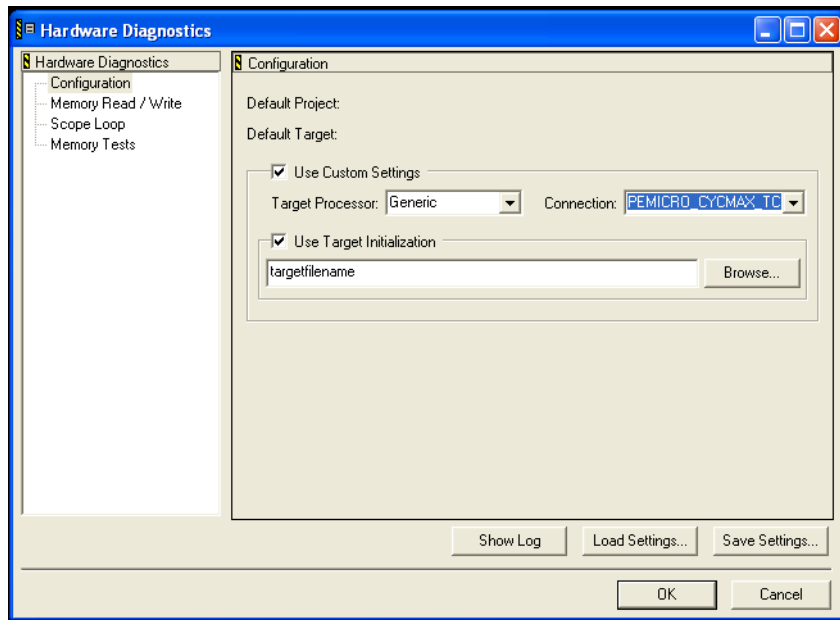
**Figure 7-4: Codewarrior Cyclone MAX USB Port Configuration**

The image above displays the configuration settings for the USB port of the Cyclone Max. The user can select the COM port to which the Cyclone Max is attached.

## 7.1.2 CodeWarrior Hardware Diagnostics

If the user wishes to test the Cyclone Max connected to their PC, they can use the Hardware Diagnostics selection from the Tools menu. This allows the user to run hardware diagnostics and memory tests to ensure proper operation of all devices. For more information on the hardware diagnostic tool, please refer to the CodeWarrior Help documentation.



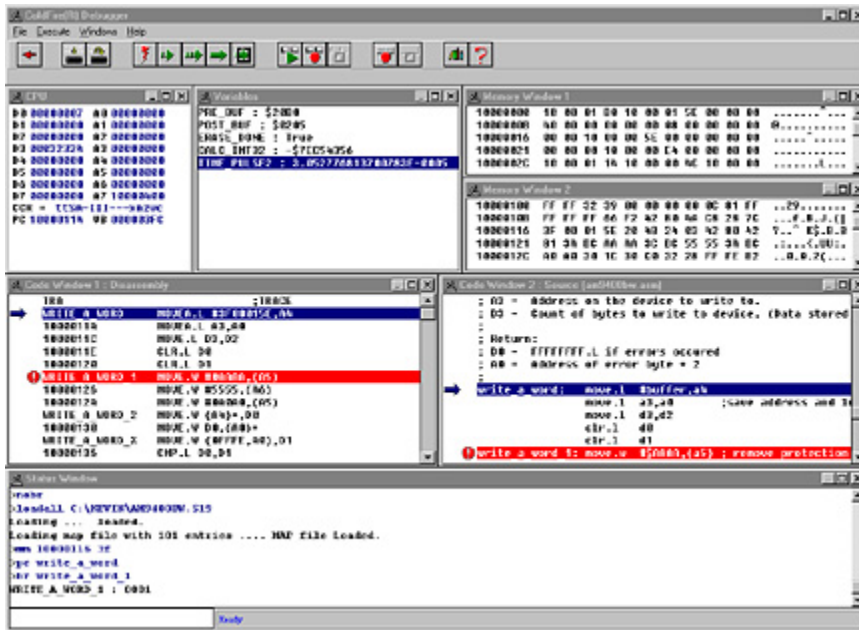


**Figure 7-5: Codewarrior Hardware Diagnostics**

## 7.2 P&E Microcomputer Systems' Software

### 7.2.1 In-Circuit Debugger

The ICD In-Circuit Debugger uses the Cyclone MAX to control the target devices. Separate ICD software is required for each of the PowerPC 5xx/8xx, ColdFire MPC5xxx, or ARM MAC7xxx architectures. With the ICD In-Circuit Debugger you can load code into the on-chip RAM, run code out of RAM or FLASH (already programmed by the In-Circuit Programmer), and set many software breakpoints and a single hardware (in FLASH) breakpoint. The main advantage of using the ICD is that your application runs in real-time at the full bus speed of the processor.



**Figure 7-1: ICDCFZ Debugger Screen Snapshot**

Debugger features include:

- Full-speed in-circuit emulation
- Breakpoints with counters on the Nth execution
- Variables window showing multiple data types
- Real-time execution as well as multiple tracing modes
- Startup and Macro files for automating the debug process
- Context-sensitive help for all commands
- Support for symbolic register files
- Full source-level debugging

## 7.2.2 In-Circuit Programmer

The PROG In-Circuit Programmer is a general-purpose programmer which allows the user to program a PowerPC 5xx/8xx, ColdFire MPC5xxx, or ARM MAC71xxx device with on-chip EEPROM/FLASH, either from an object file



(S-record), or byte by byte.

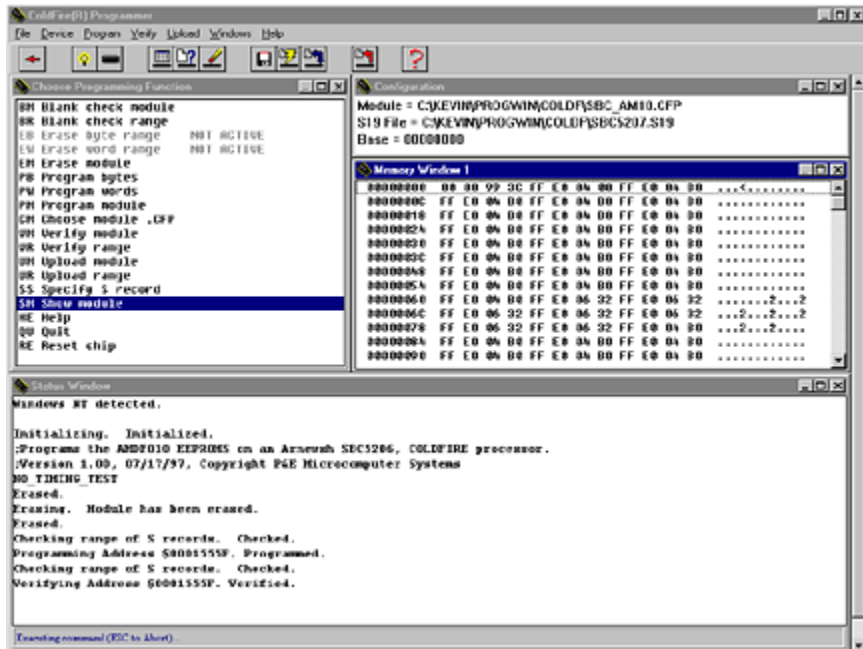


Figure 7-2: PROGCFZ Programmer Screen Snapshot

### 7.2.2.1 Command Line Programmer

CPROG is a command line programmer that allows quick turn-around time for programming target MCUs. The user may create a script file to instruct the software to execute specific commands in sequence. The CPROG software is normally included with P&E's PROG software. Please refer to CPROG documentation for more information.

### 7.2.3 P&E Microcomputer Systems PKG Software Packages

P&E's software packages contain the WinIDE integrated development environment, which pulls together an assembler, in-circuit simulator, flash memory programmer, and in-circuit debugger.



---

## 7.2.4 Latest Updates - P&E Software

The most recent updates of P&E's software products can be requested, after a brief registration, at:

[http://www.pemicro.com/SRS/main\\_screen\\_user.cfm](http://www.pemicro.com/SRS/main_screen_user.cfm).



---

## 8 ETHERNET PORT CONFIGURATION

This chapter describes the mechanism used by the Cyclone MAX device to transact data over an Ethernet network. It primarily focuses on the User Datagram Protocol (UDP), which is a popular method for sending data over a network when the speed of a data transaction is of more concern than the guarantee of its delivery. The Cyclone takes advantage of the UDP protocol's penchant for speed, and adds an extra layer of logic to guarantee the delivery of UDP packets in order to offer a best-of-both-worlds solution.

**Note:** Users who are familiar with ethernet protocol may wish to skip ahead to **Section 8.6 - Cyclone IP Setup Utility User Interface (ConfigureIP)**.

### 8.1 Network Architecture

Before delving into the innards of Ethernet message passing, it is prudent to briefly describe the different network architectures in use today, and how they pertain to the operation of the Cyclone Max. Computers are, of course, connected to one another through intermediary devices in order to form networks. There are several classes of these intermediary devices, but they generally fall into one of the following three groups:

#### Hubs

At the most basic level, computers are connected to one another through a Hub. A Hub is a device with several ports that are used to connect multiple computers together. It is a repeater device – a Hub simply copies incoming data on one of its ports as data outgoing on the other ports. In this manner, if there are four computers connected through a Hub, and if the first computer is sending data to the second computer, then the third and the fourth computers will also receive an identical copy of the data. Hubs are usually used to set up a small Local Area Network (LAN), which may have on the order of 10 to 20 computers.

#### Switches

The aforementioned types of process, where the data is simply replicated onto every available port, quickly becomes inefficient for larger sized networks. For this reason, a larger sized LAN employs the usage of Switches instead of Hubs. A Switch is essentially a smart Hub, in that it limits the input and output of data to the two transacting computers.

#### Routers

Larger networks, such as Wide Area Networks (WANs), or the Internet for



that matter, us progressively more sophisticated devices to transact data. At the core of these devices is the Router, which functions as a switch between networks.

The Cyclone performs irrespective of the connection mechanisms, with one very important caveat: it needs to be set up with the appropriate network parameters for the underlying network architectures.

## 8.2 Network Parameters

A typical network becomes operational not after the physical connections have been established, but after network parameters in the form of IP (Internet Protocol) numbers have been assigned to the individual computers. An IP number is unique string that consists of four numbers ranging between 0 and 255, separated by dots, e.g., 192.168.1.2. Every computer that is on a network needs to have a unique IP number. The computer uses this IP number to identify itself on the network, and also to address the recipient of its data.

Assignment of this IP number is sufficient information to transact data on a simple network connected by a hub. On a more complex network, however, routing information becomes important. The routing information consist of two more IP numbers. The first of these is called the Subnet Mask, and is used to determine whether or not the destination resides on the same subnet (i.e., doesn't need to be forwarded to another network). The other IP number is the Gateway Address, which is the address of the computer that handles forwarding and receiving of packets to and from other networks.

Before first use, the Cyclone needs to be programmed with a unique IP number, the Subnet Mask IP number, and also the default Gateway's IP number. This can be done via the USB or the Serial port, and is described in greater detail in **Section 8.7 - Using IPSetup.exe to configure the Cyclone Max** to configure the Cyclone Max" section of this manual.

## 8.3 Internet Protocol

Once the network has been established, and the IP numbers have been assigned, data can be transacted over a network with one of several protocols. By far the most prevalent protocol is the Transmission Control Protocol (TCP), which runs on top of the Internet Protocol in what is collectively known as the TCP/IP protocol. The TCP/IP protocol was developed by the Department of Defense to connect different computers from



different vendors by a “network of networks,” which has become what is known as the Internet today.

The primary purpose of the TCP/IP protocol was to prevent a complete network outage in the case of a nuclear attack, by automatically rerouting data traffic through the functioning part of the network. As such, the TCP/IP mechanism guaranteed delivery of data packets by introducing a system of acknowledgments and sequence numbers for the data packets. This mechanism, which is good for transacting large amounts of data (such as email or file transfers), is unsuitable in the real-time type environment in which the Cyclone operates. Because the Cyclone needs to transact data as quickly as possible to the target, it takes advantage of TCP/IP’s alternative, the UDP/IP protocol.

Unlike TCP/IP, the UDP/IP protocol is a connectionless, single-packet protocol that sends short data packets at the expense of not guaranteeing their delivery. This makes the UDP/IP protocol efficient in real-time applications such as broadcasting video over the Internet, where the occasional loss of a frame of data is not going to hamper the overall viewing experience. Left unmodified, the UDP/IP, with its lack of guarantees for packet delivery, would be unusable in an environment where the delivery of a single byte of data needs to be guaranteed. The Cyclone firmware adds mechanisms to the UDP/IP protocol, without affecting its underlying efficiency, to guarantee delivery of data packets.

## **8.4 Connecting the Cyclone Device**

There are two methods for establishing a connection between a Cyclone and a PC with an Ethernet cable. The most basic method is to connect the Cyclone directly to a PC, via a cross-over Ethernet cable. However, the more common method is to place the Cyclone and the PC on the same network through a Hub.

### **8.4.1 Connecting the Cyclone to the PC over a network**

The Cyclone was intended for use on a network of multiple computers (and other Cyclones). There are many possible network configurations, and to describe them all is beyond the scope of this document. However, most configurations are a modification of a basic theme, which is that of connecting one or more PCs through a Hub to one or more Cyclones. In order to connect these devices to the Hub, you will need to use the provided straight-through Ethernet cable. The straight-through cable, which is the “standard” Ethernet



cable, is used to connect devices of different types together, such as a PC to a Hub, or a Hub to a Cyclone. At this point it once again becomes necessary to program the Cyclone with valid IP numbers, the process for which is described in greater detail in the following section. However, it is important for the Cyclone and the PCs to have matching Subnet and Gateway IP numbers, and for each to have a unique IP number on the network. An example of a setting for above is as follows:

	<b>IP Number</b>	<b>Gateway IP</b>	<b>Subnet Mask</b>
<b>PC1</b>	192.168.100.1	192.168.100.3	255.255.255.0
<b>PC2</b>	192.168.100.2	192.168.100.3	255.255.255.0
<b>CYCLONE</b>	192.168.100.4	192.168.100.3	255.255.255.0
<b>Gateway</b>	192.168.100.3	192.168.100.3	255.255.255.0

It is important to briefly touch upon the underlying network architecture, which can be a 10Mb (Megabit), 100Mb, 10/100Mb, half-duplex, or a full duplex connection. The details of the underlying network architecture are beyond the scope of this document, but it is sufficient to note that most modern network cards, as well as the Cyclone device, have the capability to configure themselves for the underlying network through the Auto-negotiation mechanism. Auto-negotiation is performed as soon as a network cable is connected to the device, and it sets the operating parameters of the device to match those of the network.

#### **8.4.2 Connecting Cyclone-to-PC via an Ethernet cable**

In order to connect the Cyclone to a PC directly via an Ethernet cable, you need to use what is known as a cross-over cable. A cross-over cable, which is not provided by P&E, is normally used to connect two similar devices such as a PC to a PC, or a Hub to a Hub. It is a cable that has its receive and transmit wires crossed over so that the similar devices can effectively communicate with one another.

With this configuration, it is still important to assign IP numbers to both the PC and the Cyclone device. Although at first glance it may not seem necessary to assign a Gateway address in this configuration, the Cyclone was designed to operate on a network of more than two computers, and therefore it needs to be programmed with a Gateway address.





Assuming the desktop's IP number to be 192.168.100.1, this is an example of the three IP numbers that would need to be programmed into the Cyclone:

	IP Number	Gateway IP	Subnet Mask
PC	192.168.100.1	none	255.255.255.0
CYCLONE	192.168.100.2	192.168.100.1	255.255.255.0

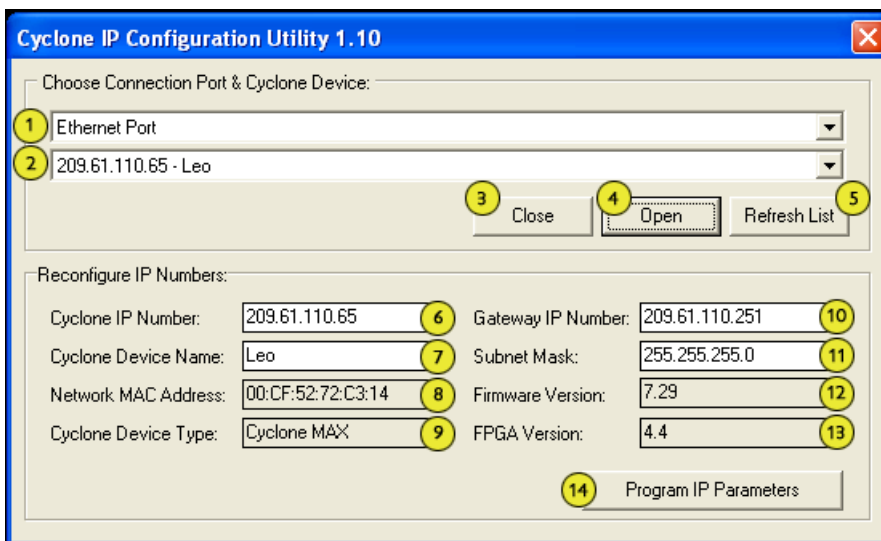
## 8.5 Cyclone IP Setup Via LCD Menu

For instructions on how to configure the Cyclone MAX using the LCD Menu, please see **Section 5.2 - Operation Procedure via LCD Menu**.

## 8.6 Cyclone IP Setup Utility User Interface (ConfigureIP)

Before the Cyclone device transacts data on an Ethernet network, it will need to be configured with the relevant network parameters. The application that provides this capability is the Cyclone IP Setup Utility (ConfigureIP), which can be found as part of the distribution software.

This utility is used to configure the Cyclone MAX with network parameters, and also to update the firmware of the Cyclone MAX.



**Figure 8-1: IPSetup.exe Default Screen**



---

## **(1) Drop-down Box 1**

There are three options available in this drop-down box, of which “Ethernet Port” is displayed. The other options are “Serial Port” and “USB Port”. Changing to any one of the three Ports will list the devices which are found over that specific Port.

## **(2) Drop-down Box 2**

Once one of the three (Serial, USB, or Ethernet) communication interfaces has been selected in the first drop-down box, a list of all available Cyclone devices over that interface will be displayed for selection.

## **(3) Close Button**

The “Close” button is active only when a device has been opened for access. Once a device has been opened for access, it needs to be closed before another device can be opened for access.

## **(4) Open Button**

The “Open” button opens a device for access. This is a required step before changing the parameters on the selected device. Once a device has been selected through the second drop-down box and is opened for access, its information will be displayed at the bottom of the dialog box.

## **(5) Refresh List**

Will refresh the dialog boxes by searching for devices which are currently connected via the Serial or USB interfaces, or are found on the network.

## **(6) Cyclone IP Number**

This is the IP number which will be associated with the Cyclone MAX. It needs to be a unique IP number which can be accessible on the network.

## **(7) Cyclone Device Name**

This is a label which can be used to identify the Cyclone MAX by name, e.g., “John’s Cyclone” or “Manufacturing Floor.”

## **(8) MAC Address**

This is the Media Access Control address, the unique number of an Ethernet device on the network. This is programmed by P&E and cannot be modified.

## **(9) Cyclone Device Type**

This displays the type of Cyclone hardware.



---

**(10) Gateway IP Number**

The IP number of a gateway on the network.

**(11) Subnet Mask**

The subnet mask of the network.

**(12) Firmware Version**

A read-only field which returns information pertaining to the build date and firmware version of the Cyclone MAX device.

**(13) FPGA Version**

A read-only field which returns the hardware version of the FPGA.

**(14) Program Cyclone Parameters**

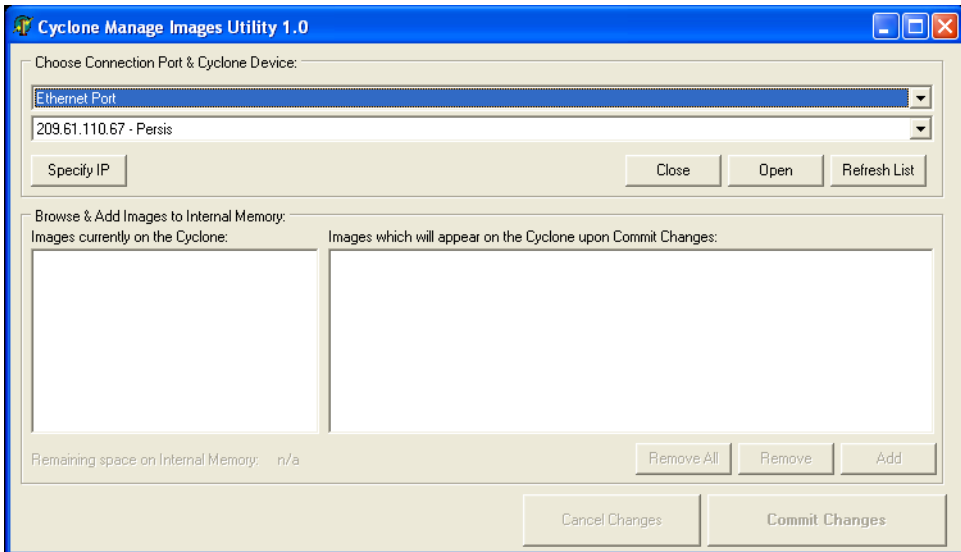
This button saves the information as it appears in the "Reconfigure IP Numbers" area onto the Cyclone device.

## **8.7 Using IPSetup.exe to configure the Cyclone Max**

Before the Cyclone MAX is ready to communicate over an Ethernet network, it will need to be configured with the relevant network parameters. The application that provides this capability is the Cyclone Configuration Utility (IPSetup.exe), and is provided as part of the standard Cyclone Max software distribution.

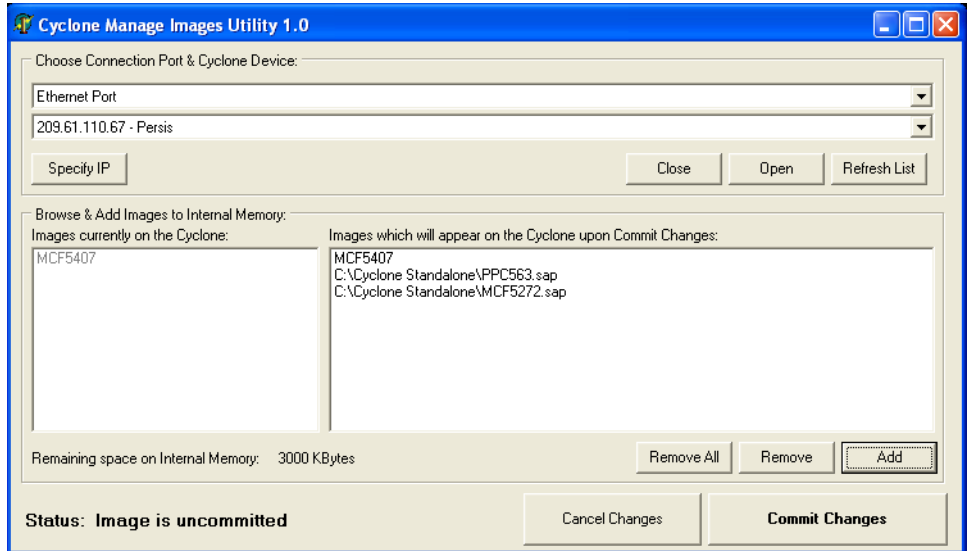
In order to update the network parameters, perform the following steps:

1. Connect a Cyclone MAX to the PC via a serial or a USB cable, and make sure that it is powered before launching the Cyclone Configuration Utility. The Cyclone Max Configuration Utility starts up with the following screen:



**Figure 8-2: Cyclone IP Setup Utility - Initial Screen**

2. Assuming that the Cyclone Max is connected to the COM1 serial port of the PC, switch from “Ethernet Port” to “Serial Port”, at which point the second drop-down box will display COM1. Click “Open” to get a dialog box similar to the following:



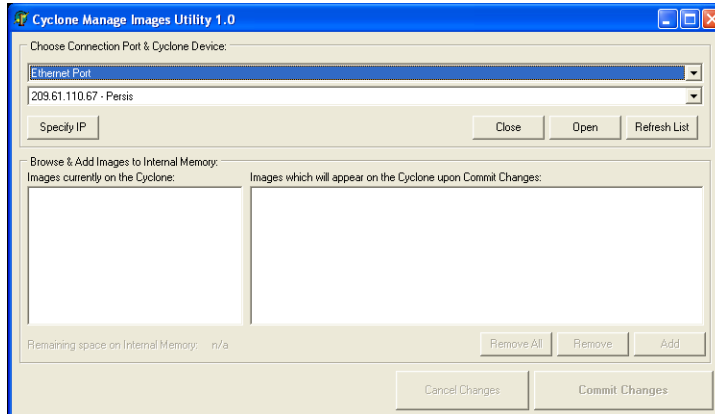
**Figure 8-3: Cyclone IP Setup Utility - Continue Setup**

3. The Cyclone Max now needs to be programmed with IP numbers for the network on which it will operate. The Cyclone IP Number field must contain a unique IP number.

## 8.8 Managing Multiple Images

The Cyclone MAX is able to store and program multiple images onto the target. Once the images have been created and saved to the disk using the CreateImage utility, the ManageImages utility can be used to add a number of images and save them collectively onto the Cyclone MAX.

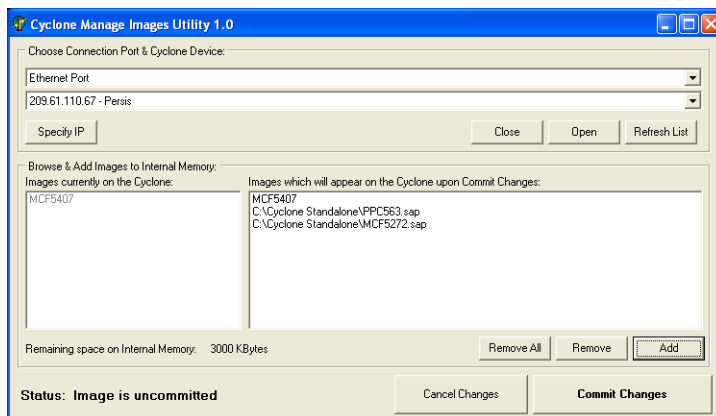
**Figure 8-4** shows the default screen, where no images are displayed:



**Figure 8-4: Manage Images Utility (Default Screen)**

Upon opening a selected Cyclone MAX, the user is provided with a list of the images currently on the unit. This is displayed in the panel to the left, with greyed-out text indicating the current state of the Cyclone MAX. The panel to the right can be used to add or delete additional images. See **Figure 8-5**.

**Note:** Any original images on the Cyclone MAX can only be removed via the “Remove All” button. Changes are not made to the Cyclone MAX until the “Commit Changes” button is pressed.



**Figure 8-5: Manage Images Utility (Multiple Images Displayed)**



---

## **9 SERIAL PORT CONFIGURATION**

The Serial Port of the Cyclone Max can be connected to any standard PC Serial Port. It should be noted that some terminal programs, such as the HyperTerm program are not designed to function with the Cyclone Max. The Cyclone Max operates at a fixed baud rate of 115200 Baud, 8 Data bits, No Parity, and 1 Stop Bit.



---

## **10 USB PORT CONFIGURATION**

The Cyclone Max can be connected to the USB port of a PC, or to a USB HUB through a standard USB cable. Cyclone Max supports USB 1.1.





## ***Freescal e Controller Continuum***

...68HC08/S08/RS08/(S)12(X) ... ColdFire® V1 ... ColdFire® V2/V3/V4 ... PowerPC® ... Nexus® ... ARM® ...



***Cyclone PRO***



***Cyclone MAX***

©2006, P&E Microcomputer Systems, Inc. All rights reserved.

P&E Micrcomputer Systems, Inc.  
P.O. Box 2044  
Woburn, MA 01888

[www.pemicro.com](http://www.pemicro.com)

Freescal e™ and the Freescal e logo are trademarks of Freescal e Semiconductor, Inc.  
All other product or service names are the property of their respective owners.  
© Freescal e Semiconductor, Inc. 2006.