CYCLONE FX Programmers

User Manual

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.

©2015-2016 P&E Microcomputer Systems, Inc.

ARM and Cortex are registered trademarksof ARM Ltd. or its subsidiaries.

NXP, ColdFire, and Kinetis are registered trademarks of NXP Semiconductors.

Texas Instruments and TI are registered trademarks of Texas Instruments Incorporated.

STMicroelectronics is a registered trademark of STMicroelectronics, Inc.

All other product or service names are the property of their respective owners.

P&E Microcomputer Systems, Inc.
98 Galen St.
Watertown, MA 02472
617-923-0053
http://www.pemicro.com

Manual version: 1.01

April 2017

1 INTRODUCTION

PEmicro's **CYCLONE FX** production programmers are powerful, fast, and feature rich in-circuit programming solutions. PEmicro offers two models which have the same feature set and only vary by the devices supported.

The **CYCLONE_ACP_FX** supports a wide variety of ARM Cortex devices.

The **CYCLONE_UNIVERSAL_FX** supports those ARM Cortex devices as well as the following NXP device families: Kinetis, LPC, S32, Qorivva (MPC5xxx), MPC5xx/8xx, DSC, S12Z, RS08, S08, HC08, HC(S)12(X), and Coldfire.

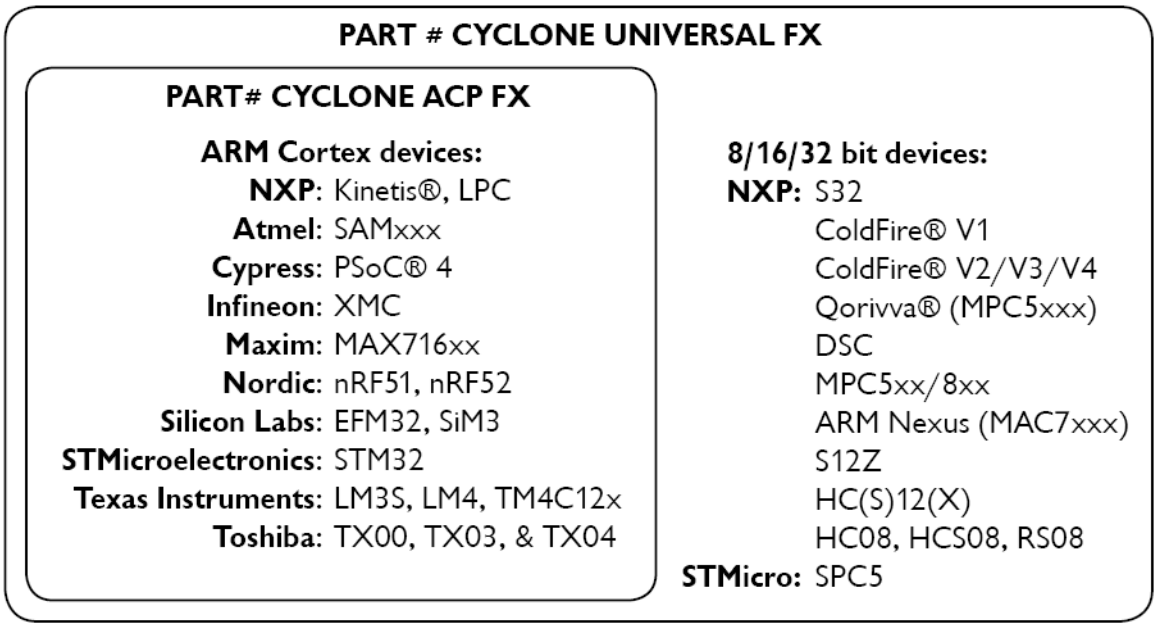


Figure 1-1: CYCLONE FX Supported Architectures

CYCLONE FX programmers are designed to withstand the demands of a production environment. They are Stand-Alone Programmers (SAP) that can be operated manually or used to host automated programming. In manual SAP mode the Cyclone is operated using the touchscreen LCD Menu and/or the Start button. Host-controlled SAP mode, for automated programming, is accomplished using either a command line utility, RS232 protocol, UDP protocol, or the Cyclone Automated Control DLL.

2 QUICK START GUIDE FOR SAP OPERATION

Stand-Alone Programming (SAP) is the most common use of the **CYCLONE FX**. This quick-start guide illustrates how easy it is to begin using the Cyclone for stand-alone programming.

You are encouraged to read this manual in its entirety for a complete description of all features specific to your Cyclone, many of which are beyond the scope of this quick-start guide.

Step 1. Install Software

The first step is to install the accompanying software. This will install all of the applications and drivers that can be used to configure/control the **CYCLONE FX**.

Once the installation is complete and the PC has been rebooted you may begin to configure the Cyclone for SAP operation.

Step 2. Hardware Setup

a. Configure the target power management scheme

Power management is configured by setting jumpers that are revealed by opening the access panel on the Cyclone's left side. The corresponding settings are conveniently illustrated on the rear label of Cyclone. No jumpers are installed by default. You may wish to refer to **Section 3.22 - Target Power Management**.

b. Connect the Cyclone to your PC

Select the appropriate communications interface (Serial, USB or Ethernet) and connect the Cyclone to your PC. If you wish to use the Ethernet port you will need to configure the corresponding network settings before use, either through the touchscreen LCD menu or via the software utility *ConfigureIP*. The Ethernet port will not function properly until this configuration is complete. You may wish to refer to **CHAPTER 9 - ETHERNET CONFIGURATION**.

c. Power up the Cyclone.

Step 3. Create a SAP Image

A SAP image, or Stand-Alone Programming image, is a self-sufficient data object containing the Cyclone and target hardware setup information, programming algorithm, programming sequence, and target data. The Cyclone uses these images to perform SAP operations on target devices. Follow these steps to create a SAP image:

a. Run the *Cyclone Image Creation Utility*

This utility is a GUI designed to help users create architecture/manufacturer-specific SAP images. To run this utility:

From the "Start" menu of your PC, navigate to "All Programs -> PEMicro. From there, select "P&E Cyclone for ARM devices Programmer" or "P&E Cyclone for ARM devices" depending on which specific PEmicro part# you are using, then select -> *P&E Cyclone for ARM devices (*or *P&E Cyclone for ARM devices) Image Creation Utility*. The utility is shown in **Figure 2-1**. Continue with the steps below to create an image.

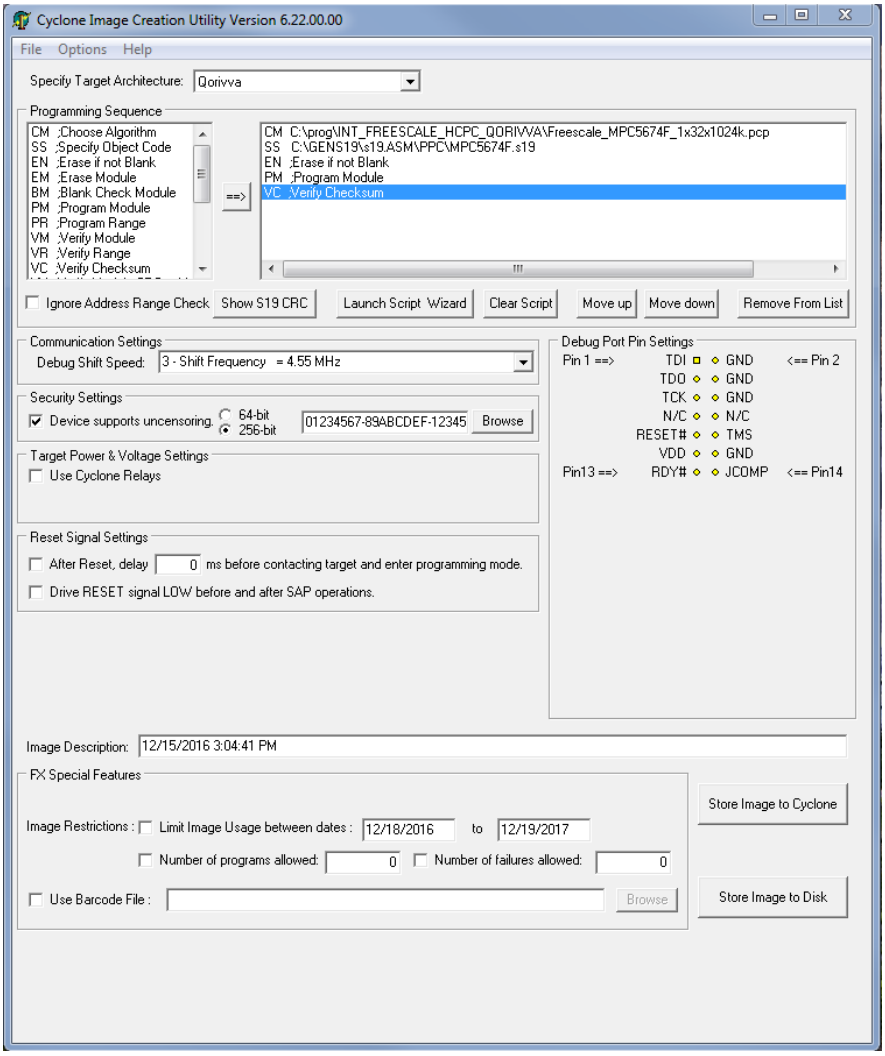


Figure 2-1: Cyclone Image Creation Utility (Qorivva Selected)

b. In the Cyclone Image Creation Utility, select your CPU manufacturer and architecture from their respective drop-down lists.

c. Click the “Launch Script Wizard” button. Follow the pop-up screens to specify a programming algorithm and target object file. The programming algorithm, target object file, and default programming sequence will then show up in the programming sequence listbox.

d. Specify the auxiliary setup and hardware setup, such as Communication Mode, Communication Rate, Target Power, and Voltage Settings.

e. Type an Image Description for your SAP image. The default description is a time stamp.

f. Click the “Store Image to Cyclone” button.

g. Choose the communication interface, select the Cyclone to which the image will be saved, and then click the “Store Image to Cyclone” button. A backend image configuration utility will pop up and store the image information on the Cyclone. Your SAP image has now been created.

Step 4. Execute SAP Image

The SAP image stored on your Cyclone can now be programmed to the target with one button press. Once your target is connected to the Cyclone, press the “Start” button of the Cyclone unit and wait for programming operations to finish. During this process, the LCD screen will show the status of operations. Note that the menu option described in **Section 5.2.3.5.3 - Set Progress**

Details will allow you to set the Cyclone to display either more or less detailed information about the programming process during programming. Eventually the “Success” or “Error” LED will illuminate, and the LCD screen will display the results.

Note: If programming is unsuccessful when using this quick start setup, the user may instead wish to use the included PROG software for their device. The PROG software allows the user to manually walk through the programming procedure step by step, which may help determine which part of setup or programming function is causing difficulty.

3 CYCLONE FX HARDWARE

The following is an overview of the features and interfaces of the **CYCLONE FX** programmers. Many of these interfaces are labeled on the underside of the plastic case.

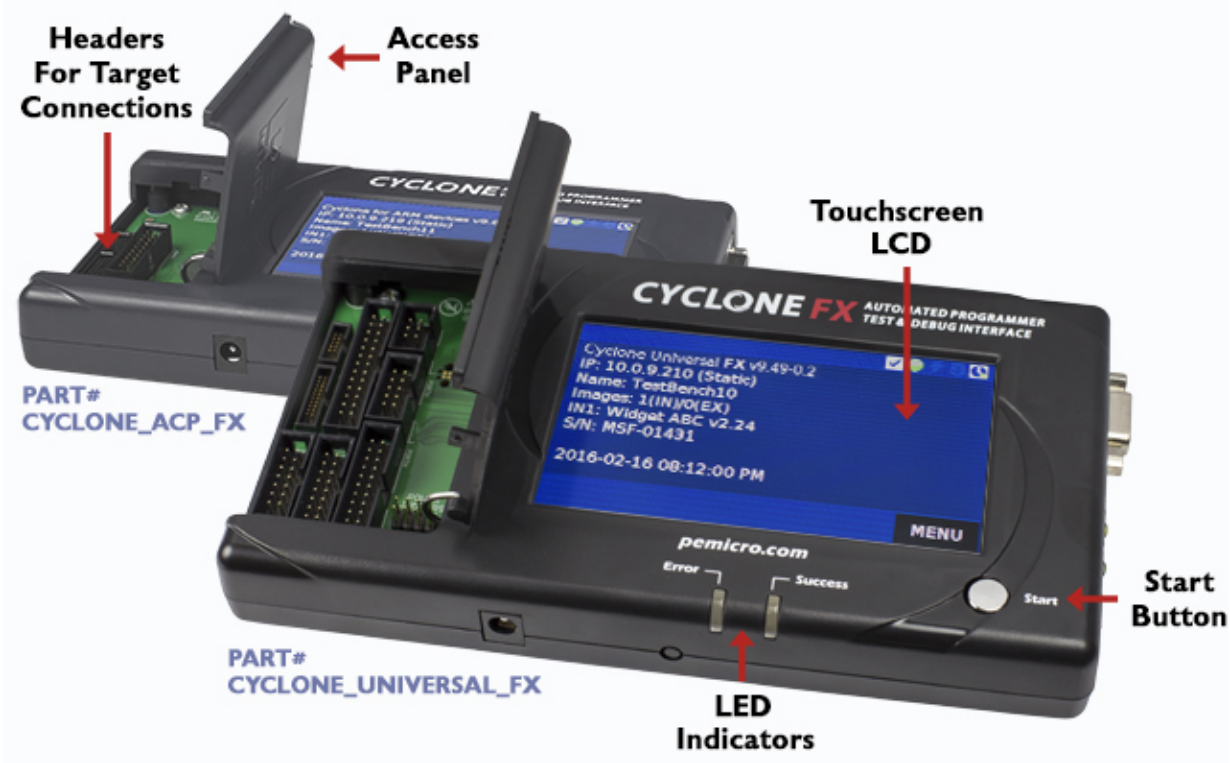


Figure 3-1: CYCLONE FX Top View

3.1 Touchscreen LCD

The LCD Touchscreen displays information about the Cyclone's configuration and the programming process, and also allows the user to navigate the Cyclone's menus. The location of the Touchscreen LCD is shown in **Figure 3-1**.

3.2 LED Indicators

The LED indicators for Error or Success will illuminate depending on the results of the programming process and provide a clear visual indication of the results. The location of the LED Indicators is shown in **Figure 3-1**.

3.3 Start Button

The Start Button can be used to begin the programming process manually, provided that the Cyclone is properly configured. The location of the Start Button is shown in **Figure 3-1**.

3.4 Access Panel

The Access Panel can easily be opened to allow the user to connect/disconnect ribbon cables from the headers, or to configure the Cyclone's Power Jumpers to select one of the available Power Management setups. The location of the Access Panel is shown in **Figure 3-1**; a layout of the headers and jumpers beneath the Access Panel is shown in **Figure 3-5**.

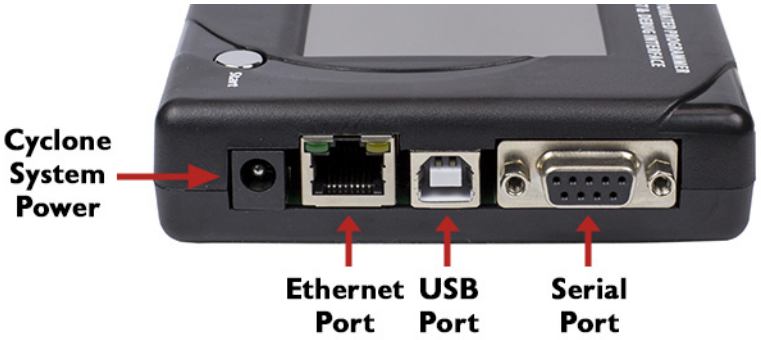


Figure 3-2: CYCLONE FX Right Side View

3.5 Cyclone System Power

The **CYCLONE FX** programmer requires a regulated 6V DC Center Positive power supply with 2.5/5.5mm female plug. Cyclones derive power from the Power Jack located on the right end of the unit. The location of Cyclone System Power is shown in **Figure 3-2**.

3.6 RS232 Communication (Serial Port)

The **CYCLONE FX** provides a DB9 Female connector to communicate with a host computer through the RS232 communication (115200 Baud, 8 Data bits, No parity, 1 Stop bit). The location of the Serial Port is shown in **Figure 3-2**.

3.7 Ethernet Communication

The **CYCLONE FX** provides a standard RJ45 socket to communicate with a host computer through the Ethernet Port (10/100 BaseT). The location of the Ethernet Port is shown in **Figure 3-2**.

3.8 USB Communications

The **CYCLONE FX** provides a USB connector for Universal Serial Bus communications between the Cyclone and the host computer. The **CYCLONE FX** is a USB 2.0 **Full-Speed** compliant device. The location of the USB Port is shown in **Figure 3-2**.

3.9 Electromechanical Relays

Inside the **CYCLONE FX** programmer, two electromechanical relays are used to cycle target power. The specifications of the relays are as following:

Maximum switched power:	30W or 125 VA
Maximum switched current:	1A
Maximum switched voltage:	150VDC or 300VAC
UL Rating:	1A at 30 VDC
	1A at 125 VAC

PEmicro only recommends switching DC voltages up to 24 Volts.

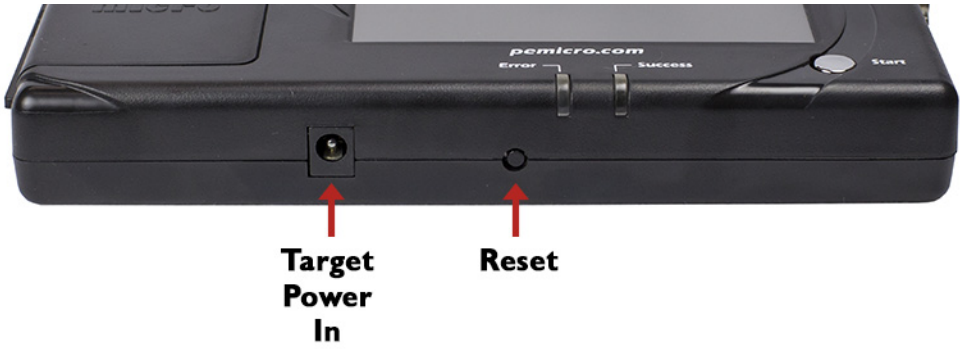


Figure 3-3: CYCLONE FX Front Side View

3.10 Power Connectors

The **CYCLONE FX** programmers provide a Target Power Supply Input Jack and a Target Power Supply Output Jack with 2.5/5.5 mm Pin Diameter. The power jacks are connected or disconnected by two electromechanical relays. When connected, the Center Pin of the Target Power Supply Input Jack is connected to the Center Pin of the Target Power Supply Output Jack. When disconnected, both terminals of the Target Power Supply Output Jack are connected to GND via a 1W, 100 Ohm resistor. The location of Target Power In is shown in **Figure 3-3**, and the location of Target Power Out is shown in **Figure 3-4**.

3.11 Reset Button

The Reset Button performs a hard reset of the Cyclone system. The location of the Reset Button is shown in **Figure 3-3**.

Figure 3-4: CYCLONE FX Rear Side View

3.12 SDHC Port

The SDHC port on the **CYCLONE FX** allows the user to store programming images that are, individually or collectively, larger than the Cyclone's internal memory. It also makes it quicker and more convenient to swap programming images. PEmicro offers certified SDHC cards on our website at pemicro.com. The **CYCLONE FX** supports a minimum of 4GB SDHC card. The location of SDHC Port is shown in **Figure 3-4**.

Programming images are managed on the SD card in exactly the same way as they are in the Cyclone's internal memory. Please see **Section 6.2 - Manage Multiple SAP Images** for more information about using the Manage Images utility.

3.13 USB Expansion Port

The location of the USB Expansion Port is shown in **Figure 3-4**. The USB Expansion Port supports use of a bar code scanner, which can provide the user with helpful features during the programming process. For detailed information on how to use the barcode scanner with a Cyclone FX, please read **Section 10 - USING A BARCODE SCANNER TO SELECT AN IMAGE &**

INITIATE PROGRAMMING.

3.14 Control Expansion Port

The Control Expansion Port is intended for future use and is not currently enabled. The location of the Control Expansion Port is shown in **Figure 3-4**.

3.15 Optional Oscillator (MON08 Only)

CYCLONE FX programmers with MON08 support (PEmicro Part# **CYCLONE_UNIVERSAL_FX** only) provide a software configurable 9.8304MHz or 4.9152 MHz oscillator clock signal to Pin 13 of the MON08 Connector. The user may use this clock signal to overdrive the target RC or crystal circuitry. If this signal is not used, just leave Pin 13 of the target MON08 header unconnected.

Note: Please note that if the target already uses an oscillator as its clock, the Cyclone will NOT be able to overdrive it. The clock should have sufficient drive to be used with a target system even if the target system has an RC circuit or crystal connected.

3.16 Cyclone Time / Real Time Clock

CYCLONE FX programmers are equipped with a Real Time Clock (RTC) module designed to keep accurate timing even when the Cyclone is turned off.

The Date & Time are displayed on the home screen. Date/Time settings can be configured by navigating to the following menu using the touchscreen display:

Main Menu / Configure Cyclone Settings / Configure Time Settings

For more information on the available configuration options, see **Section 5.2.3.3 - Configure Time Settings (Cyclone Time / Real Time Clock)**.

3.17 Power Jumper Settings

The Power Jumpers must be set differently for various power management options that the **CYCLONE FX** offers. If the target is being powered independently of the **CYCLONE FX**, all pins in the Power Jumpers header must instead be left unpopulated. To reveal the Power Jumpers header, lift the access panel on the left end of the **CYCLONE FX**. The location is indicated as Power Jumpers in **Figure 3-5**. Please see **Section 3.22 - Target Power Management** for the correct jumper settings for the Cyclone's power management options. A quick guide to these settings is also located on the underside label of the **CYCLONE FX**.

3.18 Debug Connectors

When purchasing a **CYCLONE FX** programmer, the user is able to choose between two part numbers, each corresponding to a different level of device support. See the sticker on the underside of the Cyclone to determine the PEmicro part# for your specific **CYCLONE FX** programmer.

PEmicro Part# CYCLONE_ACP_FX supports ARM Cortex devices only, so this programmer provides one shrouded, un-keyed, 0.100-inch pitch dual row 0.025-inch square header, and two shrouded, keyed 0.050-inch pitch dual row mini headers.

PEmicro Part# CYCLONE_UNIVERSAL_FX supports ARM Cortex devices and additionally supports target connections to many 8-/16-/32-bit NXP architectures, so this programmer provides six shrouded, un-keyed, 0.100-inch pitch dual row 0.025-inch square headers, and two shrouded, keyed 0.050-inch pitch dual row mini headers.

To reveal the headers and connect/disconnect ribbon cables, lift the access panel on the left end of the Cyclone. Each header is designated for one or more specific target architectures, as indicated in **Figure 3-5**.

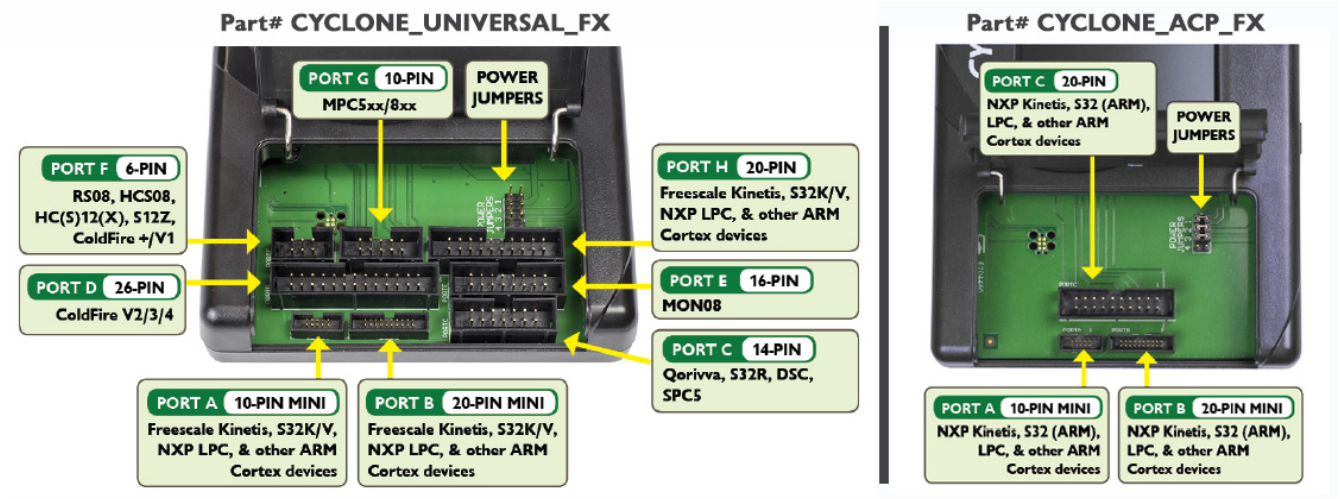


Figure 3-5: Target Headers & Power Jumpers (CYCLONE_UNIVERSAL_FX vs.CYCLONE_ACP_FX)

Mechanical drawings are shown below whose dimensions are representative of the pin size and spacing of these headers.

Note: The number of pins depicted in the mechanical drawings may differ from the Cyclone headers; the drawings are provided simply to demonstrate pin size and spacing.

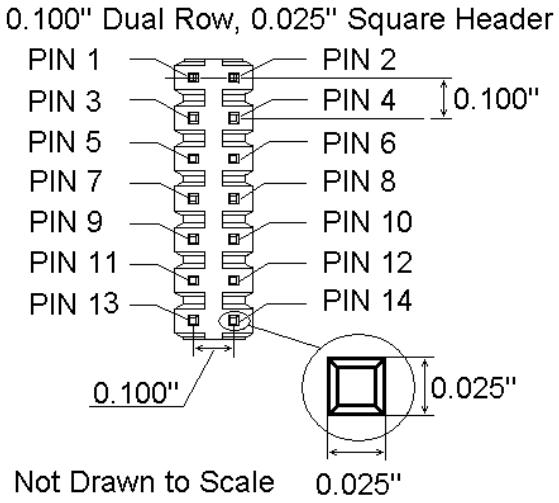


Figure 3-6: 20-Pin Un-Keyed Header Dimensions

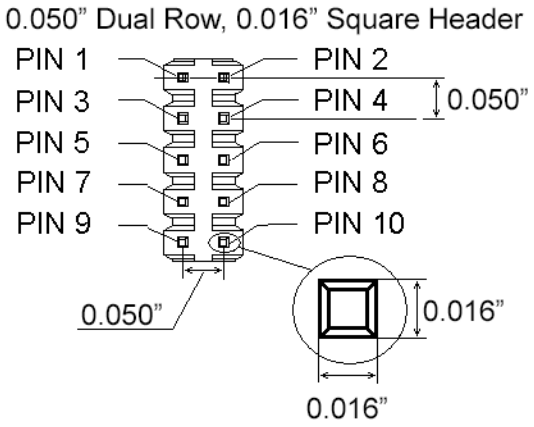


Figure 3-7: Mini 10-Pin and Mini 20-Pin Keyed Header Dimensions

3.19 Target Headers For Part# CYCLONE_ACP_FX

PEmicro Part# **CYCLONE_ACP_FX** features 3 ports labeled A-C.

3.19.1 PORT A: 10-Pin Keyed Mini Connector (Kinetis, S32 (ARM), other PEmicro-Supported ARM devices)

The Cyclone provides a keyed 10-pin 0.050-inch pitch double row connector for ARM targets. The location of the this header is indicated as PORT A in **Figure 3-5**. The 10-pin keyed mini connector pin definitions for JTAG mode are as follows:

10-Pin Keyed Mini Connector JTAG Mode Pin Assignments

PIN 1 -	**TVCC**	**TMS** -	PIN 2
PIN 3 -	**GND**	**TCK** -	PIN 4
PIN 5 -	**GND**	**TDO** -	PIN 6
PIN 7 -	**NC**	**TDI** -	PIN 8
PIN 9 -	**NC**	**RESET** -	PIN 10

CYCLONE FX programmers also support SWD Mode. This replaces the JTAG connection with a clock and single bi-directional data pin.

10-Pin Keyed Mini Connector SWD Mode Pin Assignments

PIN 1 -	**TVCC**	**TMS/SWDIO** -	PIN 2
PIN 3 -	**GND**	**TCK/SWCLK** -	PIN 4
PIN 5 -	**GND**	**NC** -	PIN 6
PIN 7 -	**NC**	**NC** -	PIN 8
PIN 9 -	**NC**	**RESET** -	PIN 10

SWD Mode is selected from the “Communication Mode” drop-down box in the Cyclone Image Creation Utility:

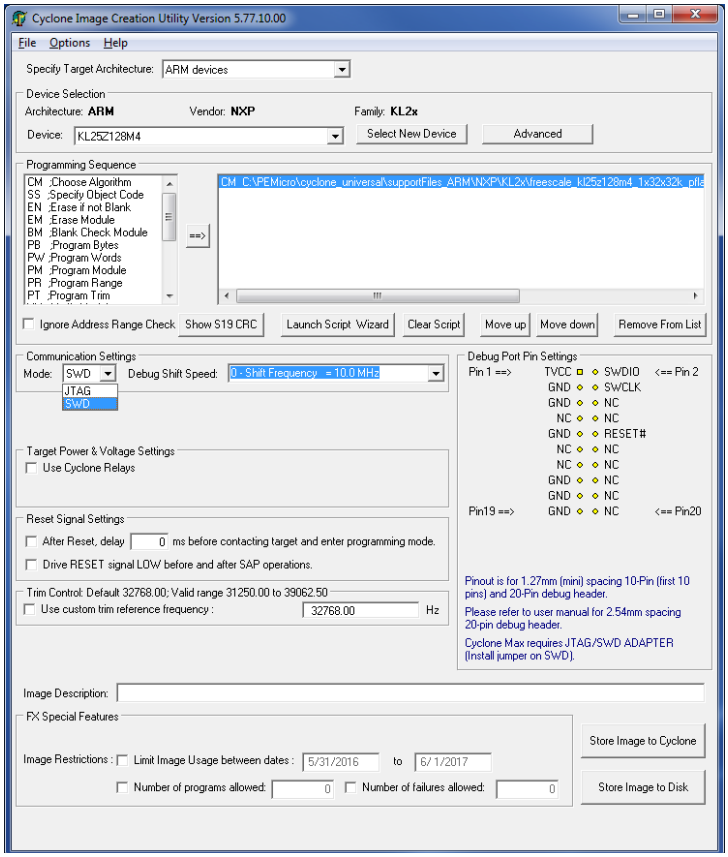


Figure 3-8: Communications Mode Selection

3.19.2 PORT B: 20-Pin Keyed Mini Connector (Kinetis, S32 (ARM), other PEmicro-Supported ARM devices)

The Cyclone provides a keyed 20-pin 0.050-inch pitch double row connector for ARM targets. The location of the this header is indicated as PORT B in **Figure 3-5**. The 20-pin keyed mini connector pin definitions for JTAG mode are as follows:

20-Pin Keyed Mini Connector JTAG Mode Pin Assignments

PIN 1 - **TVCC**	**TMS** - PIN 2
PIN 3 - **GND**	**TCK** - PIN 4
PIN 5 - **GND**	**TDO** - PIN 6
PIN 7 - **NC**	**TDI** - PIN 8
PIN 9 - **NC**	**RESET** - PIN 10
PIN 11 - **NC**	**NC** - PIN 12
PIN 13 - **NC**	**NC** - PIN 14
PIN 15 - **GND**	**NC** - PIN 16
PIN 17 - **GND**	**NC** - PIN 18
PIN 19 - **GND**	**NC** - PIN 20

CYCLONE FX programmers also support SWD Mode. This replaces the JTAG connection with a clock and single bi-directional data pin.

20-Pin Keyed Mini Connector SWD Mode Pin Assignments

PIN 1 - **TVCC**	**TMS/SWDIO** - PIN 2
PIN 3 - **GND**	**TCK/SWCLK** - PIN 4
PIN 5 - **GND**	**NC** - PIN 6
PIN 7 - **NC**	**NC** - PIN 8
PIN 9 - **NC**	**RESET** - PIN 10
PIN 11 - **NC**	**NC** - PIN 12
PIN 13 - **NC**	**NC** - PIN 14

PIN 15 - **GND**	**NC** - PIN 16
PIN 17 - **GND**	**NC** - PIN 18
PIN 19 - **GND**	**NC** - PIN 20

SWD Mode is selected from the “Communication Mode” drop-down box in the Cyclone Image Creation Utility:

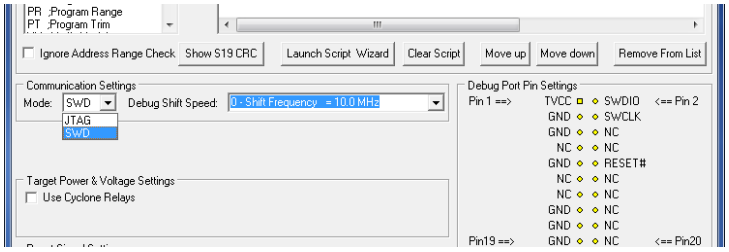


Figure 3-9: Communications Mode Selection

3.19.3 PORT C: 20-Pin Debug Connector (Kinetis, S32 (ARM), other PEmicro-Supported ARM devices)

The Cyclone provides a 20-pin 0.100-inch pitch double row connector for ARM targets. The location of the this header is indicated as **PORT C** in **Figure 3-5**. The 20-pin standard connector pin definitions for JTAG mode are as follows:

20-Pin Standard Connector JTAG Mode Pin Assignments

PIN 1 - **TVCC**	**NC** - PIN 2
PIN 3 - **TRST or NC**	**GND** - PIN 4
PIN 5 - **TDI**	**GND** - PIN 6
PIN 7 - **TMS**	**GND** - PIN 8
PIN 9 - **TCK**	**GND** - PIN 10
PIN 11 - **NC**	**GND** - PIN 12
PIN 13 - **TDO**	**GND** - PIN 14
PIN 15 - **RESET**	**GND** - PIN 16
PIN 17 - **NC**	**GND** - PIN 18
PIN 19 - **NC**	**GND** - PIN 20

CYCLONE FX programmers also support SWD Mode. This replaces the JTAG connection with a clock and single bi-directional data pin.

20-Pin Standard Connector SWD Mode Pin Assignments

PIN 1 - **TVCC**	**NC** - PIN 2
PIN 3 - **TRST or NC**	**GND** - PIN 4
PIN 5 - **NC**	**GND** - PIN 6
PIN 7 - **TMS/SWDIO**	**GND** - PIN 8
PIN 9 - **TCK/SWCLK**	**GND** - PIN 10
PIN 11 - **NC**	**GND** - PIN 12
PIN 13 - **NC**	**GND** - PIN 14
PIN 15 - **RESET**	**GND** - PIN 16
PIN 17 - **NC**	**GND** - PIN 18
PIN 19 - **NC**	**GND** - PIN 20

SWD Mode is selected from the “Communication Mode” drop-down box in the Cyclone Image Creation Utility:

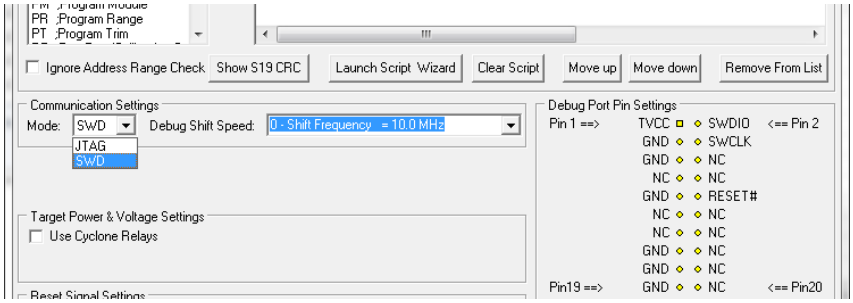


Figure 3-10: Communications Mode Selection

3.20 Target Headers For Part# CYCLONE_UNIVERSAL_FX

PEmicro Part# **CYCLONE_UNIVERSAL_FX** features 6 ports labeled A-H.

3.20.1 PORT A: 10-Pin Keyed Mini Connector (Kinetis, S32 (ARM), other PEmicro-Supported ARM devices)

The Cyclone provides a keyed 10-pin 0.050-inch pitch double row connector for ARM targets. The location of the this header is indicated as PORT A in **Figure 3-5**. The 10-pin keyed mini connector pin definitions for JTAG mode are as follows:

10-Pin Keyed Mini Connector JTAG Mode Pin Assignments

PIN 1 -	**TVCC**	**TMS**	**-** PIN 2
PIN 3 -	**GND**	**TCK**	**-** PIN 4
PIN 5 -	**GND**	**TDO**	**-** PIN 6
PIN 7 -	**NC**	**TDI**	**-** PIN 8
PIN 9 -	**NC**	**RESET**	**-** PIN 10

CYCLONE FX programmers also support SWD Mode. This replaces the JTAG connection with a clock and single bi-directional data pin.

10-Pin Keyed Mini Connector SWD Mode Pin Assignments

PIN 1 -	**TVCC**	**TMS/SWDIO**	**-** PIN 2
PIN 3 -	**GND**	**TCK/SWCLK**	**-** PIN 4
PIN 5 -	**GND**	**NC**	**-** PIN 6
PIN 7 -	**NC**	**NC**	**-** PIN 8
PIN 9 -	**NC**	**RESET**	**-** PIN 10

SWD Mode is selected from the “Communication Mode” drop-down box in the Cyclone Image Creation Utility:

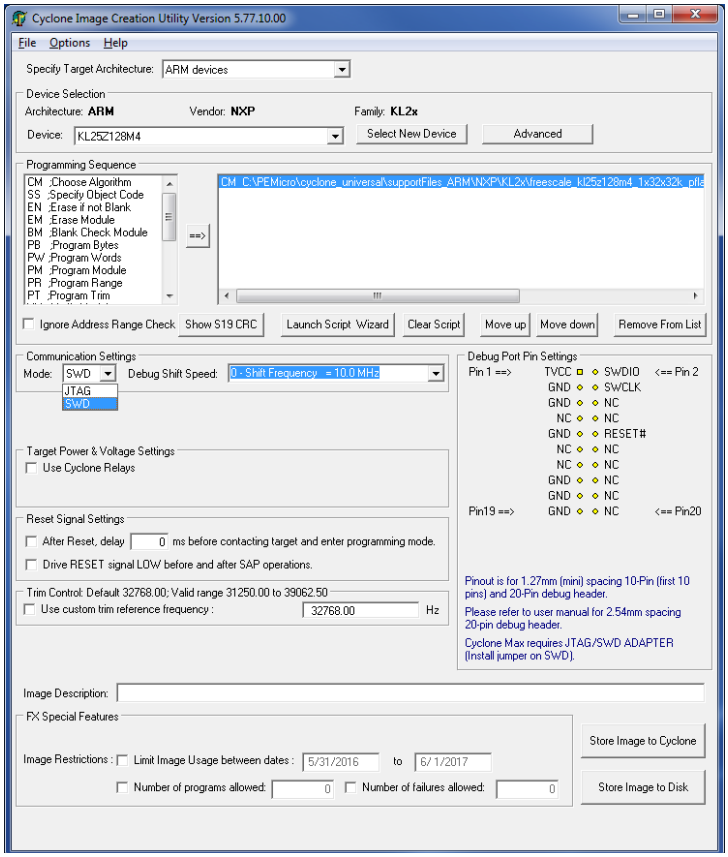


Figure 3-11: Communications Mode Selection

3.20.2 PORT B: 20-Pin Keyed Mini Connector (Kinetis, S32 (ARM), other PEmicro-Supported ARM devices)

The Cyclone provides a keyed 20-pin 0.050-inch pitch double row connector for ARM targets. The location of the this header is indicated as PORT B in **Figure 3-5**. The 20-pin keyed mini connector pin definitions for JTAG mode are as follows:

20-Pin Keyed Mini Connector JTAG Mode Pin Assignments

PIN 1 - **TVCC**	**TMS** - PIN 2
PIN 3 - **GND**	**TCK** - PIN 4
PIN 5 - **GND**	**TDO** - PIN 6
PIN 7 - **NC**	**TDI** - PIN 8
PIN 9 - **NC**	**RESET** - PIN 10
PIN 11 - **NC**	**NC** - PIN 12
PIN 13 - **NC**	**NC** - PIN 14
PIN 15 - **GND**	**NC** - PIN 16
PIN 17 - **GND**	**NC** - PIN 18
PIN 19 - **GND**	**NC** - PIN 20

CYCLONE FX programmers also support SWD Mode. This replaces the JTAG connection with a clock and single bi-directional data pin.

20-Pin Keyed Mini Connector SWD Mode Pin Assignments

PIN 1 - **TVCC**	**TMS/SWDIO** - PIN 2
PIN 3 - **GND**	**TCK/SWCLK** - PIN 4
PIN 5 - **GND**	**NC** - PIN 6
PIN 7 - **NC**	**NC** - PIN 8
PIN 9 - **NC**	**RESET** - PIN 10
PIN 11 - **NC**	**NC** - PIN 12
PIN 13 - **NC**	**NC** - PIN 14

PIN 15 - **GND**	**NC** - PIN 16
PIN 17 - **GND**	**NC** - PIN 18
PIN 19 - **GND**	**NC** - PIN 20

SWD Mode is selected from the “Communication Mode” drop-down box in the Cyclone Image Creation Utility:

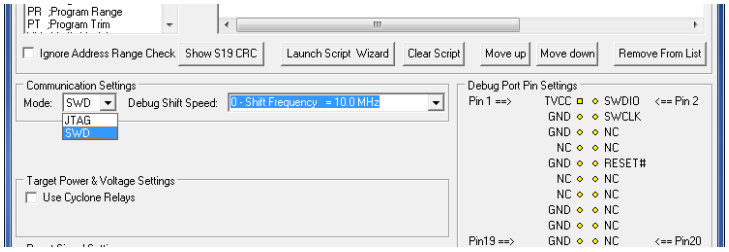


Figure 3-12: Communications Mode Selection

3.20.3 PORT C: 14-Pin Debug Connector (Qorivva, SPC5, DSC, S32 (Power))

The Cyclone provides a standard 14-pin 0.100-inch pitch dual row 0.025-inch square header for Qorivva (MPC5xxx), DSC (MC56F8xxx), S32R, or STMicroelectronics’ SPC5 targets. The location of the this header is indicated as PORT C in **Figure 3-5**.

Qorivva, SPC5, or S32 (Power) 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

DSC Pinout

TDI	**1**	**2**	GND
TDO	**3**	**4**	GND
TCK	**5**	**6**	GND
NC	**7**	**8**	NC/KEY
RESET	**9**	**10**	TMS
VDD	**11**	**12**	GND
NC	**13**	**14**	TRST

3.20.3.1 BERG14-to-MICTOR38 Optional Connector

PEmicro offers a 14-pin BERG to 38-pin MICTOR adapter, sold separately, that may be used on Port C of the **CYCLONE FX**. The PEmicro part number is BERG14-TO-MICTOR38.

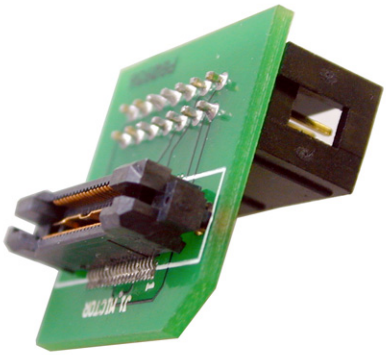


Figure 3-13: BERG14-TO-MICTOR38 Adapter (Sold Separately)

3.20.4 PORT D: 26-Pin Debug Connector (ColdFire V2/3/4)

The Cyclone provides a standard 26-pin 0.100-inch pitch dual row 0.025-inch square header for ColdFire MCF52xx/53xx/54xx family of microprocessors. This port connects to the target hardware using either the ColdFire extension cable for synchronous ColdFire targets such as MCF5272 & MCF5206E (PEmicro part# CABLE-CF-ADAPTER, sold separately), or a standard 26-pin ribbon cable for asynchronous ColdFire targets (included). Please refer to each processor's user manual to identify whether it is a synchronous or asynchronous interface. The location of the this header is indicated as PORT D in **Figure 3-5**.

ColdFire V2/3/4 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:

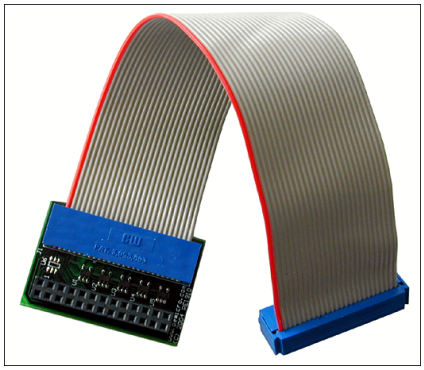


Figure 3-14: ColdFire Extension Cable With Adapter (PEmicro part# CABLE_CF_ADAPTER, for

synchronous ColdFire targets, sold separately)

Figure 3-15: ColdFire Ribbon Cable (for asynchronous ColdFire targets, included with Cyclone)

3.20.5 PORT E: 16-Pin Debug Connector (MON08)

The Cyclone provides a 16-pin 0.100-inch pitch double row connector for MON08 targets. The location of the this header is indicated as PORT E in **Figure 3-5**. The MON08 header adopts the standard pin-out from MON08 debugging with some modifications. The general pin-out is as follows:

MON08 Signals

PIN 1 -	**NC**	**GND**	- PIN 2
PIN 3 -	**NC**	**RST**	- PIN 4
PIN 5 -	**NC**	**IRQ**	- PIN 6
PIN 7 -	**NC**	**MON4**	- PIN 8
PIN 9 -	**NC**	**MON5**	- PIN10
PIN11 -	**NC**	**MON6**	- PIN12
PIN13 -	**OSC**	**MON7**	- PIN14
PIN15 -	**Vout**	**MON8**	- PIN16

3.20.6 PORT F: 6-Pin Debug Connector (RS08, HCS08, HC(S)12(X), S12Z, ColdFire +/V1)

The Cyclone provides a standard 6-pin 0.100-inch pitch dual row 0.025-inch square header for ColdFire V1, S12Z, 68(S)12(X), 68HCS08, and RS08 targets. The location of the this header is indicated as PORT F in **Figure 3-5**. The header uses the NXP standard pin configuration, listed here for reference:

ColdFire V1, 68(S)12(X), 68HCS08, and RS08 Signals

PIN 1 - **BKGD**	**GND** - PIN 2
PIN 3 - **NC**	**RESET** - PIN 4
PIN 5 - **NC**	**TVCC** - PIN 6

S12Z Signals

Note:* indicates optional signal

PIN 1 - **BKGD**	**GND** - PIN 2
PIN 3 - **PDO***	**RESET** - PIN 4
PIN 5 - **PDOCLK***	**TVCC** - PIN 6

3.20.7 PORT G: 10-Pin Debug Connector (Power MPC5xx/8xx)

The Cyclone provides a standard 10-pin 0.100-inch pitch dual row 0.025-inch square header for

Power MPC5xx/8xx BDM targets. The location of the this header is indicated as PORT G in **Figure 3-5**.

Power MPC5xx/8xx BDM Pinout

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

3.20.8 PORT H: 20-Pin Debug Connector (Kinetis, S32 (ARM), other PEmicro-Supported ARM devices)

The Cyclone provides a 20-pin 0.100-inch pitch double row connector for ARM targets. The location of the this header is indicated as **PORT H** in **Figure 3-5**. The 20-pin standard connector pin definitions for JTAG mode are as follows:

20-Pin Standard Connector JTAG Mode Pin Assignments

PIN 1 -	**TVCC**	**NC**	- PIN 2
PIN 3 -	**TRST or NC**	**GND**	- PIN 4
PIN 5 -	**TDI**	**GND**	- PIN 6
PIN 7 -	**TMS**	**GND**	- PIN 8
PIN 9 -	**TCK**	**GND**	- PIN 10
PIN 11 -	**NC**	**GND**	- PIN 12
PIN 13 -	**TDO**	**GND**	- PIN 14
PIN 15 -	**RESET**	**GND**	- PIN 16
PIN 17 -	**NC**	**GND**	- PIN 18
PIN 19 -	**NC**	**GND**	- PIN 20

CYCLONE FX programmers also support SWD Mode. This replaces the JTAG connection with a clock and single bi-directional data pin.

20-Pin Standard Connector SWD Mode Pin Assignments

PIN 1 -	**TVCC**	**NC**	- PIN 2
PIN 3 -	**TRST or NC**	**GND**	- PIN 4
PIN 5 -	**NC**	**GND**	- PIN 6
PIN 7 -	**TMS/SWDIO**	**GND**	- PIN 8
PIN 9 -	**TCK/SWCLK**	**GND**	- PIN 10
PIN 11 -	**NC**	**GND**	- PIN 12
PIN 13 -	**NC**	**GND**	- PIN 14
PIN 15 -	**RESET**	**GND**	- PIN 16
PIN 17 -	**NC**	**GND**	- PIN 18
PIN 19 -	**NC**	**GND**	- PIN 20

SWD Mode is selected from the “Communication Mode” drop-down box in the Cyclone Image Creation Utility:

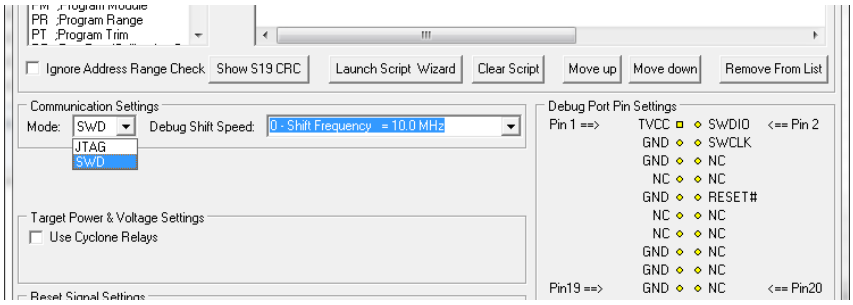


Figure 3-16: Communications Mode Selection

3.21 Ribbon Cable

CYCLONE FX programmers communicate with the target through ribbon cables. The ribbon cables for standard debug connectors have a 0.100-inch centerline dual row socket IDC assembly (not keyed). The ribbon cables for 10- and 20-pin mini debug connectors have a 0.050-inch centerline dual row socket IDC assembly (keyed). The ribbon cables are designed such that the Cyclone's Debug Connector has the same pinout as the Target Header, i.e., Pin 1 of the Cyclone's Debug Connector is connected to Pin 1 of the Target Header. As an example, **Figure 3-17** sketches the connection mechanism (looking down into the sockets) for a 14-pin ribbon cable. Ribbon cables for other supported architectures use a similar scheme, but may have more or fewer pins.

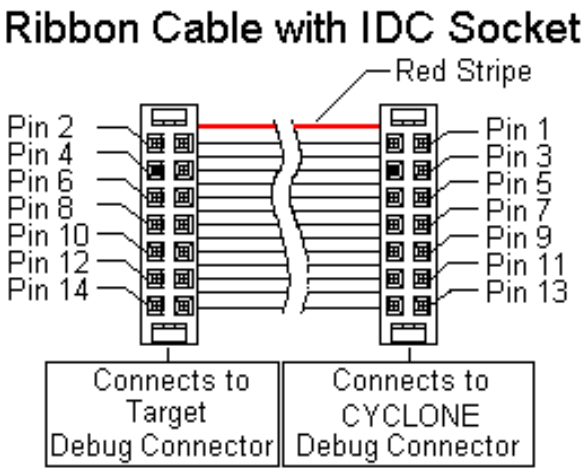


Figure 3-17: Ribbon Cable Example Diagram, When Looking Into IDC Socket

4 TARGET POWER MANAGEMENT

Different target devices may require different power schemes which depend on the design of the target board, target voltages, and even the device architecture. PEmicro has designed the **CYCLONE FX** to be capable of powering a target before, during, and after programming. Power can be sourced at many voltage levels from the Cyclone itself, or sourced by an external power supply and switched by the Cyclone.

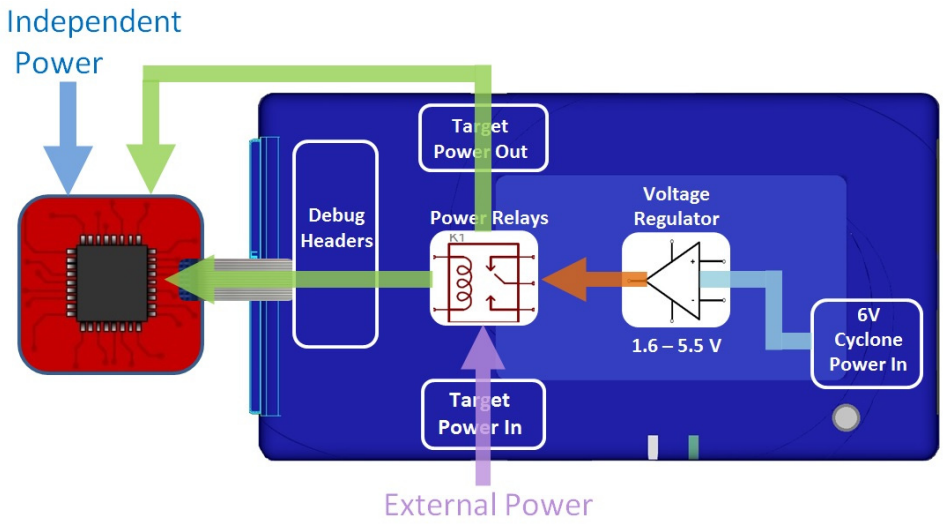


Figure 4-1: Five different paths to power a target

The versatility of the Cyclone power scheme gives the user the utmost flexibility, and includes the following features:

- Provides power through a power jack or through the debug connector
- Provides internally generated voltage from 1.6v-5.5v at up to 500mA
- Switches an external power supply voltage, up to 24V at 1amp
- Selectively powers the target before, during, and after programming
- Powers down the target connections between programming operations
- Uses power switching to aid entry into debug mode for certain targets
- Provides target voltage and current measurement capabilities

If target power is required, each target board may vary where the power is sourced from, externally or internally, and how it is channeled to the target: through the debug header or to a separate connector to the board. Power that is passed through and managed by the Cyclone goes through power relays so it can be power cycled. This is extremely useful because it also allows the power to be off during setup and automatically powered on by the Cyclone for programming. For some devices, the process of entering debug mode requires that the device be powered down and powered back up. Power can also be left in a desired power state, either on or off.

4.1 Cyclone Configuration

There are two different places Power Management is configured and they should be **matched**: first, by the **jumpers** on the **CYCLONE FX**, and second, in the **setup** of the programming image. The Cyclone jumpers are the most important because they are the physical connection to the target. The Cyclone has an easy access panel that reveals debug header connections for a variety of different architectures, and a 2x4 jumper block for configuring power management of the target. The specific location of the jumpers is indicated by the label POWER JUMPERS in **Figure 4-3**. This set of 4 jumpers can be used to set 5 different power management schemes for the target.

Note: If these jumpers are not set correctly, the Cyclone will not function as intended..

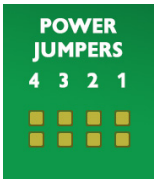
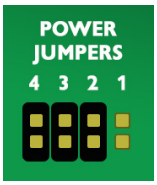
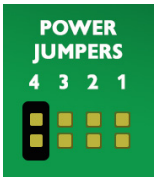
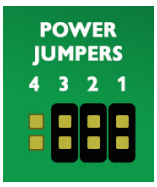
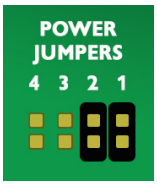
1	Target is powered independently	
2	Power provided externally (center +) and managed by Cyclone, power out to debug ribbon cable.	
3	Power provided externally (center +) and managed by Cyclone, power out to 2.5 mm output jack (center +)	
4	Power provided by Cyclone, power out to debug ribbon cable	
5	Power provided by Cyclone, power out to 2.5 mm output jack (center +)	

Figure 4-2: Cyclone Power Schemes & Corresponding Jumper Settings

The bottom edge of the **CYCLONE FX** has a Power In jack for externally provided power, and the top edge of the Cyclone has Power Out jack, for when power schemes including these are used (see **Figure 4-3**). One of the provided ribbon cables is connected to the appropriate debug header based on the specific target architecture.

Figure 4-3: Cyclone Hardware Features: Power Jumpers and Target Headers

The power settings that are set by the jumpers are a physical connection and take precedence.

After the basic hardware setup, target power and voltage settings are also set in the creation of a SAP (stand-alone programming) image. At a minimum the SAP image contains all the commands to Erase, Program, and Verify a programming image. More sophisticated power selections in the SAP image can control the relays, target voltage, delays, and power down after SAP operations, as shown in the selection dialog.

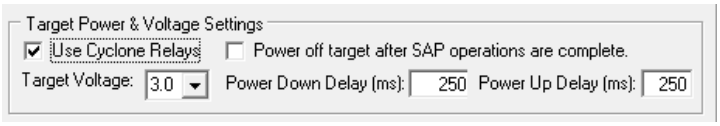


Figure 4-4: Target Power & Voltage Settings

Target voltages (with appropriate jumper settings) in the range of 1.6 to 5.5 volts may be provided. There is also the option to select the internal Cyclone relays to power cycle the Cyclone during programming, and set the length of delays during power up and down. This is extremely useful to make sure the power is off when hooking up the target. Power cycling is especially important for architectures that require it to enter debug mode. The SAP image settings may even be used to turn off the target power once programming is completed, to ensure that the microcontroller is left in a halted state and not running.

4.2 Cyclone Setup

Below is a tutorial that demonstrates how to set up the **CYCLONE FX** in each of the 5 power configurations. A very common configuration is the independently powered target. In this power scenario, the Cyclone will detect and use the power on the target for the appropriate debug communication voltages.

4.2.1 Independently Powered Target

In the simplest and most common scenario, no jumpers are set, so the target is powered independently from the Cyclone. No power is passed through the debug header, just the standard debug signals. The Cyclone automatically detects the target power and sets the debug signals to match.

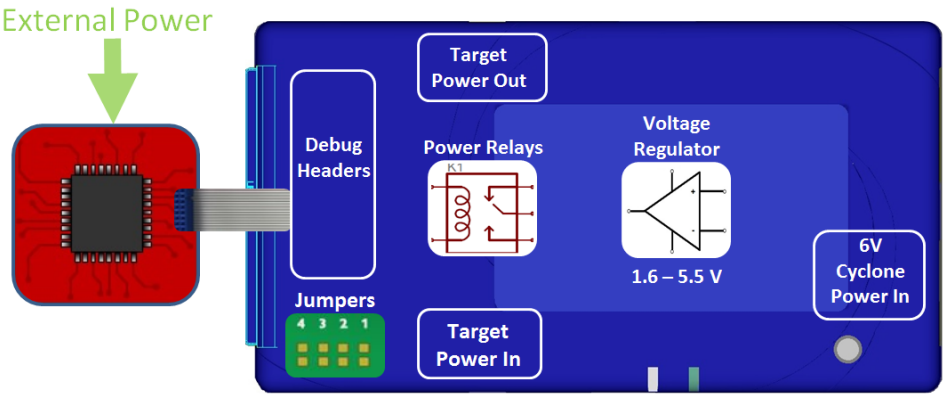


Figure 4-5: Independently Powered Target

4.2.2 Power provided by the Cyclone to the debug cable

It is also possible for the Cyclone to generate power through an internal regulator in the range of 1.6 to 5.5 Volts. In the jumper configuration below, the Cyclone generates the power through a voltage regulator, and passes it through the power relays and out through the debug ribbon cable, which is set up during the SAP image creation. There is only one connection to the target processor which will handle both the communication and the power. In this scenario, external power must not be connected to the Power In jack since it is already being provided.

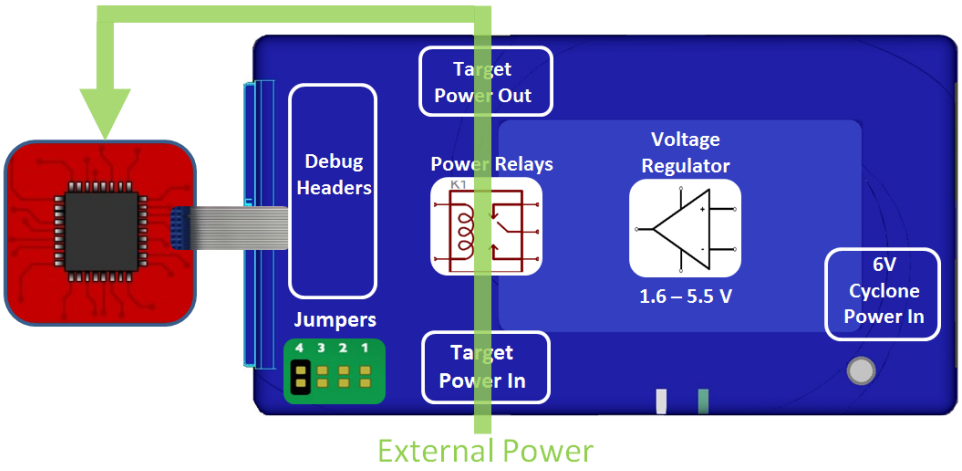


Figure 4-6: Power Provided by the Cyclone to the Debug Cable

4.2.3 External Power passed through the Cyclone and out 2.5 mm barrel port

It is also possible to provide external power, passed through the Cyclone power relays, and back out to be available to power the target board externally. This is useful when the user wants to control the power to the target and the target board has an external power connector. Setting a single jumper will connect the barrel port input connector on the bottom edge of the Cyclone, through the relays, to a matched 2.5 mm barrel port output connector on the top edge of the Cyclone, so that the power can be routed into and back out of the Cyclone.

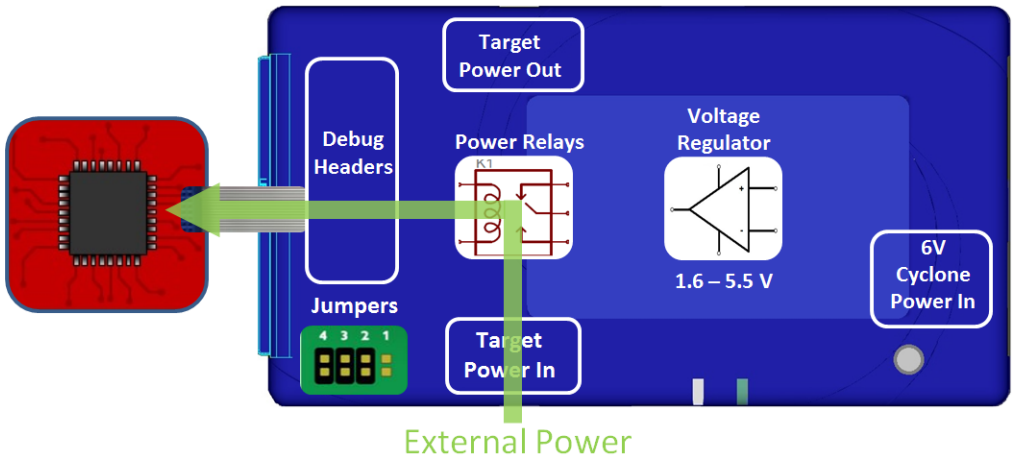


Figure 4-7: External Power Passed Through the Cyclone and Out 2.5 mm Barrel Port

4.2.4 External Power passed through the Cyclone to the debug cable

In a slightly different scenario, the user may wish to provide power to the target through the debug cable. On the bottom edge of the Cyclone is a 2.5 mm Power In port barrel which will pass power through target relays which lets the Cyclone take control of the power cycling during programming. This simple setup requires only an input to the Cyclone and a single ribbon cable connection to the target board that handles both communication and power. The external power provided must be between 1.6 to 5.5 volts.

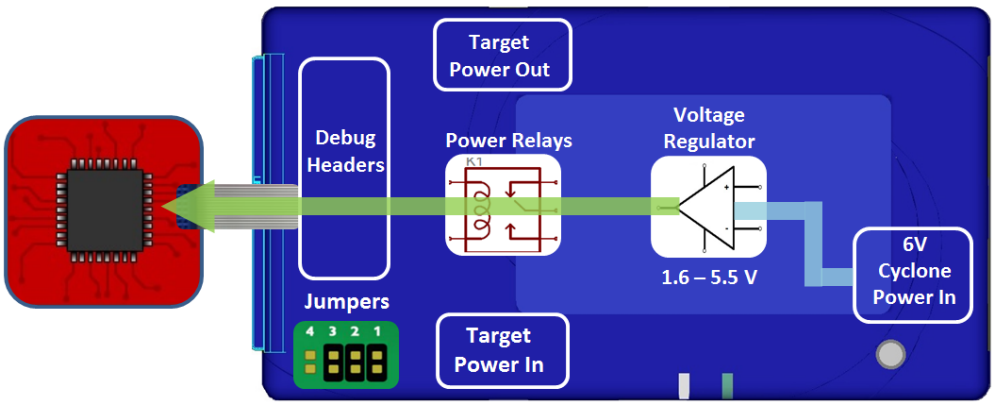


Figure 4-8: External Power Passed Through the Cyclone to the Debug Cable

4.2.5 Power provided by the Cyclone and out 2.5 mm barrel port

In a slightly different scenario, the user may wish to have the Cyclone provide power, but power the target via an external connector on the target. The voltage supplied to the target is determined by the settings in the SAP image. When generating the SAP image the Cyclone relays must be selected as well as the correct voltage level for the target.

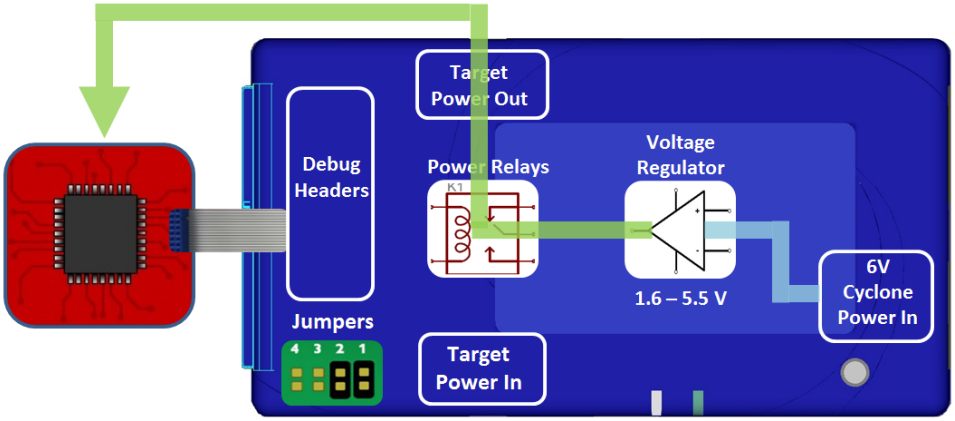


Figure 4-9: Power Provided by the Cyclone and Out 2.5 mm Barrel Port

4.3 Setup Reminders

The most important step when providing power out to a target is to check the Cyclone's **jumper settings** to make sure they match the intended power setup. The jumpers control the power settings which determine how power is supplied, regardless of the SAP image settings. If the jumpers are set for power to be provided through the Cyclone, and the target is externally powered, this is a conflict and may cause damage to the board.

In the case where power is being supplied through the Cyclone and the target is not being powered on, the user should **first check the jumper settings** to make sure they match the intended power setup. Second, the user should check to make sure the SAP image has the 'Use Cyclone Relays' box checked with the appropriate voltage level selected.

5 TOUCHSCREEN LCD MENU

This chapter describes the Cyclone's touchscreen LCD menu. **Figure 5-1** shows an overview of the menu structure.

Note: This menu will change as features are added to the **CYCLONE FX**, so if your menu does not match what is displayed here, please check PEmicro's website, www.pemicro.com, for a user manual containing the latest LCD Menu operations information.

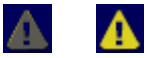
5.1 Home Screen

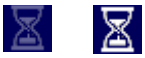
The home screen appears when the **CYCLONE FX** is powered on, or when the  Home button is tapped.

5.1.1 Icons

A row of icons in the upper right corner indicates the status of various attributes of the Cyclone.

Note: The user may tap on the row of icons to view the meaning of each of the currently displayed icons.

Cyclone Unit Status: Ok / Bad 

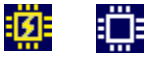
Programming Status: Ready / Busy 

Target Power Relays: On / Off 

USB-To-PC Enumerated: Yes / No 

Real-Time clock Enabled & Working: Yes / No 

Cyclone Power Relays: Closed / Open 

Target Device Is Powered*: Yes / No 

SDHC Memory Card: None / Valid / Unformattted / Reset Cyclone** 

Barcode Scanner: Detected / Not Detected 

***** Target Device Is Powered - "Yes" indicates that the **CYCLONE FX** detects power on the Vcc pin of the target device programming header.

** SDHC Memory Card - "Reset Cyclone" indicates that the Cyclone needs to be reset before the SDHC card will register as Valid. The user can push the Reset button which is located on the front side of the Cyclone, below the LED indicators.

5.1.2 Configurable Display Area

The main area of the home screen can be configured to optionally display the following information, by using the Cyclone IP Configuration Utility (see **Section 9.6.1 - LCD Home Screen Display Selection**):

1. Firmware version of the Cyclone (always shown).
2. IP address assigned to the Cyclone.
3. Name assigned to the Cyclone.
4. Number of programming images in the Cyclone's memory.
5. Name of the selected programming image.
6. First serial number associated with the selected image
7. Current status.
8. Results of the last operation performed.
9. Time and date.

10. Status Window and Main Menu button (always shown).
11. Target voltage and/or current
12. Programming count & limit

5.1.3 Status Window

The status window appears in the lower left corner of the home screen and displays the results of programming operations.

5.1.4 Error Information Icon

When the Cyclone experiences an error during programming operations, the Info icon will appear to the left of the Menu button (or AUX button, if configured).

Info Icon: 

Press the Info icon to view a detailed description of the error.

5.1.5 AUX Button (Appears If Configured)

The Cyclone allows the user to add an Auxiliary (AUX) button to the home screen which will perform a specific function when pressed. The specific function is chosen by the user when the AUX button is configured. The AUX button will appear on the home screen to the left of the “Menu” button, in the lower right corner of the home screen.

Figure 5-1: AUX Button On Home Screen (configured to perform CRC32 function)

For information on how to configure the AUX button, see **Section 5.2.4 - Status**.

5.2 Main Menu

The Main Menu is accessible by pressing the “Menu” button when the Home Screen is displayed. The Main Menu contains the following selections:

Select Programming Image	-	-	-
Current Image Options	Execute Image Function	*Launch Programming*	-
		Verify Data In Target	-
		Toggle Power	-
		Power cycle device to run user code	-
		Validate Image CRC32	-
	Set Image Validation	*Enable Validate IMG*	-
		Disable Validate IMG	-
	Modify Next Serial Number	*Image S/N:*	-
		Decrease Next Serial (will specify "not allowed" if no serial # attached to IMG)	-
		Increase Next Serial (will specify "not allowed" if no serial # attached to IMG)	-
		Current Image ID Selected:	-
		Current Alg ID Selected:	-
		Current CS ID Selected:	-
	Show Current Image Stats (FX Only)	-	-
Configure Cyclone Settings	Edit Cyclone Name	--	-
	Configure Network Settings	Show Current IP Settings	*Current IP Mode*
			IP
			Mask
			Gateway
			MAC Address
		Edit Static IP Settings	*IP*
			Mask
			Gateway
			MAC Address
		Enable/Disable Dynamic IP	-
	Configure Time Settings	Modify Date/Time	Update Time From Internet
			Set Time Zone Hours
			Set Time Zone Minutes
		Set Time-Date Display	*Display Date Only*
			Display Time Only
			Display Date and Time
		Set Date Formatting	*YYYY-MM-DD*
			MM-DD-YYYY
			DD-MM-YYYY
			MM/DD/YYYY
		Set Time Formatting	*HH:MM (24-hour)*
			HH:MM (AM/PM)
			HH:MM:SS (24-hour)
			HH:MM:SS (AM/PM)
	Configure AUX Button	No operation	-
		Perform verify only	-
		Toggle Power	-
		Validate image CRC32	-
		Power cycle device to run user code	-
	Configure Screen	Configure Home Screen	Line 2 Display: [...] Line 8 Display:
		Change Screen Brightness	-
		Calibrate Screen	-
		Set Progress Details	*Show Progress Details*
			Hide Progress Details
			Show Details, Keep Last Progress On
	Configure Storage	Format Internal Storage	-
		Format External SD Card (FX Only)	-
	Configure USB Host Device (FX Only)	Disable	-
		Enable USB Scanner	-
Status	Show Current IP Settings	Current IP Mode	-
		IP	
		Mask	-
		Gateway	-
		MAC Address	-

Figure 5-2: Main Menu Structure

5.2.1 Select Programming Image

Displays a list of the available programming images so that the user may select one for programming. Images in the **CYCLONE FX** internal memory are preceded by the designation **"IN"** and numbered sequentially, i.e., **IN1:** *first image name*, **IN2:** *second image name*. If the SDHC port of the **CYCLONE FX** contains a memory card, any programming images that reside on the SD card will be preceded by the designation **"EX"** in similar fashion.

You may tap the appropriate image to select it. The image name shown is the one specified in the Cyclone Configuration Utility when saving the image to the Cyclone/SD card.

5.2.2 Current Image Options

This menu presents options that allow the user to select or configure programming images on the **CYCLONE FX**.

5.2.2.1 Execute Image Function

Execute Specific SAP Function presents four Stand-Alone Programming functions that you may execute by tapping the function that you wish to execute:

5.2.2.1.1 Launch Programming

This allows the user to execute the programming function. The **CYCLONE FX** will program the target device, if able, using the currently selected programming image. This is functionally equivalent to pressing the Start button.

5.2.2.1.2 Verify Data In Target

Performs a verify function on the data that has been programmed into the target device.

5.2.2.1.3 Toggle Power

Toggles the target power and makes sure all ports are driven to debug mode level.

5.2.2.1.4 Power Cycle Device To Run User Code

Toggles the target power and maintains tri-state mode for all signals.

5.2.2.1.5 Validate Image CRC32

Allows the user to perform a CRC32 validation on the currently selected programming image.

5.2.2.2 Set Image Validation

Allows the user to choose between two validation settings: 1) validate the image each time the Start button is pressed, or 2) do not validate the image.

5.2.2.3 Modify Next Serial Number

Presents options that display the current serial number and allow the user to increase or decrease the next serial number. Tap "Current Image ID Selected" to view/choose the desired programming image; tap "Current Alg ID Selected" to view/choose the desired programming algorithm; use "Current CS ID Selected" to view/choose the desired Choose Serial file. The adjustment buttons will display "Increase Not Allowed" and "Decrease Not Allowed" if the image/algorithm/CS files that the user has selected to do not allow for this operation.

5.2.2.4 Show Current Image Stats (FX Only)

Displays current statistics, if any, for Image Programmed Count & Maximum Allowed, Errors Logged & Maximum Allowed, and Date Range Allowed. These limits can be set in the Image Creation Utility when using the **CYCLONE FX**.

Note: When *Current Image Stats* is displayed as a home screen item, only Image Programmed Count & Maximum Allowed are displayed on the home screen.

5.2.3 Configure Cyclone Settings

Presents options that allow the user to choose to configure the **CYCLONE FX** network settings, time/date settings, and LCD touchscreen display settings, or to set the display to dynamic.

5.2.3.1 Edit Cyclone Name

Allows the user to edit the name of the Cyclone using the on-screen keyboard. Click “Done” to save the new Cyclone name or “Cancel” to exit without saving a new Cyclone name. This name will be displayed on the **CYCLONE FX** home screen if the Cyclone is configured to do so.

5.2.3.2 Configure Network Settings

Presents options that allows the user to view or edit various IP settings, toggle the IP settings between static and dynamic, and re-name the **CYCLONE FX**.

5.2.3.2.1 Show Current IP Settings

This menu allows you to view the **CYCLONE FX** IP address, Mask, and Gateway, and MAC address. You may also tap these entries to edit, as long as the Cyclone is set to Static IP mode.

Dynamic vs. Static

There are two schemes for assigning IP addresses. One is the Static IP addressing mode. This involves the user manually setting the IP address for every device on the network. In this case, it falls to the user to ensure the IPs assigned do not conflict and are within the boundaries of the network. The other is the Dynamic Host Configuration Protocol (DHCP). This involves setting up a separate server to manage the IP addresses. The server is given a list of valid IP addresses for the network. Using a predetermined set of rules, each new device that wishes to connect to the network is given an IP address by the server. This takes the task of managing the validity and uniqueness of IP addresses out of the user's hands and relegates it to the server. **CYCLONE FX** programmers are capable of using either Static IP addressing or DHCP.

5.2.3.2.2 Edit Static IP Settings

This menu allows you to edit the Cyclone’s IP address, Mask, and Gateway, and view the Cyclone’s MAC address. If you are unable to edit these values, you may wish to check to be certain that the Cyclone is not set to Dynamic IP mode.

IP

Edit IP Numbers allows the user to set an IP number for the Cyclone. The current IP number is displayed on the second line. Tap a number to edit and use the touchscreen keyboard to set the new number. When you are finished, hit Done. If you change your mind and decide not to save, hit Cancel to leave the IP number as is and return to the Main Menu.

Mask

Edit IP Mask allows the user to set an IP Mask for the Cyclone. 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.

Gateway

Edit IP Gateway allows the user to set the IP Gateway for the Cyclone. 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.

MAC Address

Show MAC Address displays the current MAC address for the Cyclone.

5.2.3.2.3 Enable/Disable Dynamic IP

Allows the user to toggle the Cyclone configuration between utilizing a Static IP address or a

Dynamic IP address. The user must reset the **CYCLONE FX** after changing from Static to Dynamic or vice-versa. The reset button on the front side of the unit may be used.

5.2.3.3 Configure Time Settings (Cyclone Time / Real Time Clock)

The **CYCLONE FX** is equipped with a Real Time Clock (RTC) module designed to keep accurate timing even when the Cyclone is turned off. The Date & Time are displayed on the home screen.

This menu presents options that allow the user to configure the Cyclone's various date/time/timezone settings, including formatting options.

5.2.3.3.1 Modify Date / Time

1. **Update Time from Internet** - Connects to an SNTP server, fetches the current time, and saves it to the Cyclone. When executed it displays a message that this can freeze the Cyclone for up to 3 minutes – This is due to an invalid ARP response due to a bad gateway configuration. Proper configuration will ensure the problem is resolved. If the network connection is not configured/connected this displays a message that the time failed to update. If it is successful no message is displayed.
2. **Set Time Zone Hours** - Allows you to set the timezone offset, in hours +/-, from GMT time
3. **Set Time Zone Minutes** - Allows you to set the timezone offset, in minutes +/-, from GMT time

5.2.3.3.2 Set Time-Date Display

Allows you set the Cyclone's Time-Date Display to one of the following configurations:

1. Display Date Only
2. Display Time Only
3. Display Date and Time

5.2.3.3.3 Set Date Formatting

Allows you to select how the date is displayed. The options are:

1. YYYY-MM-DD
2. MM-DD-YYYY
3. DD-MM-YYYY
4. MM/DD/YYYY

5.2.3.3.4 Set Time Formatting

Allows you to select how the time is displayed. The options are:

1. HH:MM (24-hour)
2. HH:MM (AM/PM)
3. HH:MM:SS (24-hour)
4. HH:MM:SS (AM/PM)

5.2.3.4 Configure AUX Button

Allows the user to configure an auxiliary (AUX) button which (if configured) will be labeled appropriately and displayed to the left of the Menu button on the Cyclone's touchscreen LCD. When pressed, the AUX button will perform the task for which it has been configured. The options that may be assigned to the AUX button are:

1. No Operation - No operation is assigned to the AUX button and it will not be displayed on

the LCD screen.

2. Perform Verify Only - Verifies the data on the target device against the data in the programming image.
3. Toggle Power - Toggles the Cyclones power relays off/on.
4. Validate Image CRC32 - Validates the CRC32 of the data on the target device against that of the data in the programming image.
5. Power Cycle Device To Run User Code - Toggles the target power and maintains tri-state mode for all signals.

5.2.3.5 Configure Screen

This menu presents options that allow the user to adjust or customize the Cyclone's LCD touchscreen display in various ways.

5.2.3.5.1 Change Screen Brightness

Allows the user to adjust the brightness of the LCD touchscreen. The "Increase" and "Decrease" buttons will raise or lower the brightness level, respectively, in increments of 10%. Brightness can be adjusted from between 100% - 10%. Press "Done" to exit.

5.2.3.5.2 Calibrate Screen

Allows the user to click on specified points on the LCD touchscreen in order to calibrate the accuracy of the touch function. Follow the on-screen instructions.

5.2.3.5.3 Set Progress Details

This configures the display to present more detailed information during the progress of programming, including the specific programming steps that are performed and specific information about the programming and verifying procedure. The user may select "Show Details, Keep Last Progress On," "Show Progress Details," or "Hide Progress Details."

5.2.3.5.4 Configure Home Screen

This menu allows you to choose what information to display on Lines 2-8 of the home screen. Available elements to display consist if information such as: the current IP address, the Cyclone name, the number of images, etc. In this way the user can customize the display to provide the information that they find most useful. There is a separate button for each of Lines 2-8. Tapping on the button for a specific Line brings up a list of elements that you can choose to display on that Line of the home screen. If the list of elements is greater than one page, tap the More button to view the rest of the available elements. Tap the element that you want to display on that line and then tap Done to save your selection.

Note: The **CYCLONE FX** allows one additional option for display: the target device's voltage and/or current.

5.2.3.6 Configure Storage

This menu selection allows the user to format the Cyclone's internal memory. This menu will also allow the user to format an SD card located in the Cyclone's memory expansion slot. Select "Format Internal Storage" or "Format External SD Card". The user will be prompted to ensure that they wish to format the corresponding memory. Tap "Yes" to format, or "Cancel" to go back to the previous menu option without formatting the memory.

5.2.3.7 Configure USB Host Device (FX Only)

This menu selection allows the user to either Enable or Disable a barcode scanner connected to the Cyclone FX's USB Extension Port.

Select **Enable** to provide power to and turn on a connected barcode scanner. The Enable setting is persistent. Once Enabled, the scanner will be powered and turned on whenever the Cyclone is powered. To disable the barcode scanner select **Disable**.

5.2.4 Status

This menu contains a selection that allow the user to view status information regarding various aspects of the Cyclone. This menu will likely be expanded with future updates.

5.2.4.1 Show Current IP Settings

Allows the user to view the Cyclone's Current IP Mode, IP Address, Mask, Gateway, and MAC Address.

6 STAND-ALONE PROGRAMMER CONFIGURATION

The Cyclone may act as a Stand-Alone In-Circuit Programmer. A simple user interface, CREATEIMAGE.EXE, is provided for configuring the Cyclone.

Note: If the user wishes to use a programming image created with an earlier generation Cyclone (such as the Cyclone PRO or MAX, or the Cyclone for ARM devices Rev. A/B) they should first convert the image using the conversion utility described in **CHAPTER 16 - SAP CONVERTER UTILITY**.

6.1 Create A Stand-Alone Programming (SAP) Image

This chapter describes in detail how to configure the **CYCLONE FX** for stand-alone programming using the Cyclone Image Creation Utility, shown in **Figure 6-1**. The **CYCLONE FX** does not require a target to be connected when it is being configured. However, the power of the **CYCLONE FX** must be turned on and one of the communications interfaces must be connected to the **CYCLONE FX** if an image is to be stored on it.

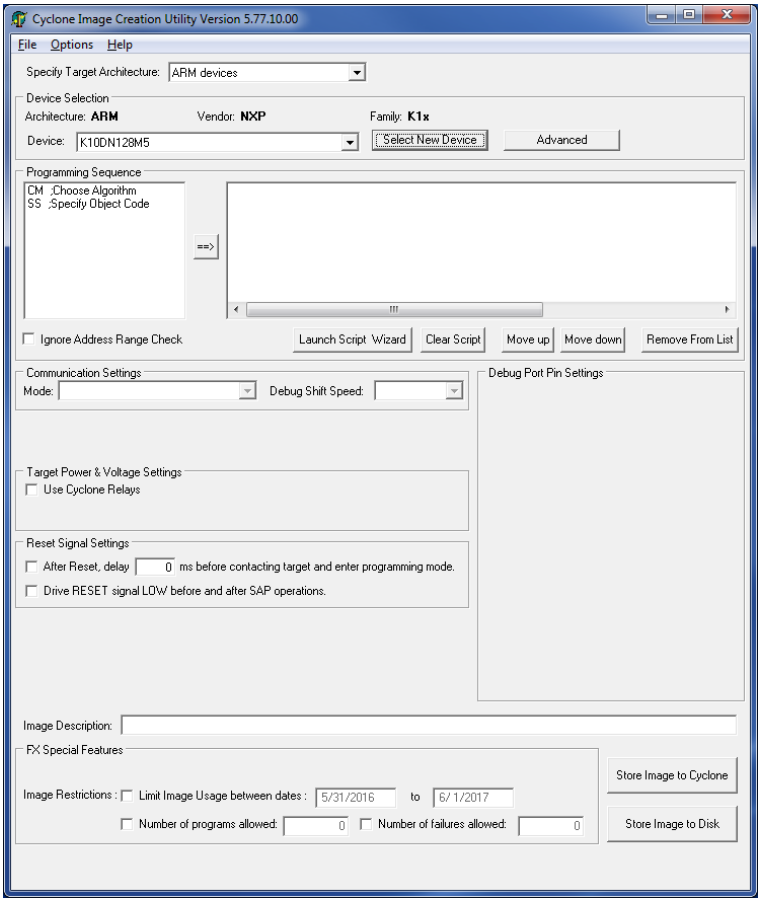


Figure 6-1: Cyclone Image Creation Utility

6.1.1 Specify Target Architecture

CYCLONE FX programmers support ARM Cortex devices from several manufacturers** - including NXP's Kinetis and LPC devices. **If you are using PEmicro Part# CYCLONE_UNIVERSAL_FX**, this Cyclone also supports these 8-16/32-bit architectures: NXP's S32, ColdFire® V2/V3/V4, ColdFire+/V1, MPC5xx/8xx, Qorivva® (MPC5xxx), DSC, ARM® Nexus (MAC7xxx), S12Z, HC(S)12(X), HCS08, HC08, and RS08 devices, as well as STMicroelectronics SPC5 devices.

***For a complete index of PEmicro-supported ARM Cortex devices, please view **pemicro.com/arm**.*

The user may select the CPU Manufacturer from the drop-down list:

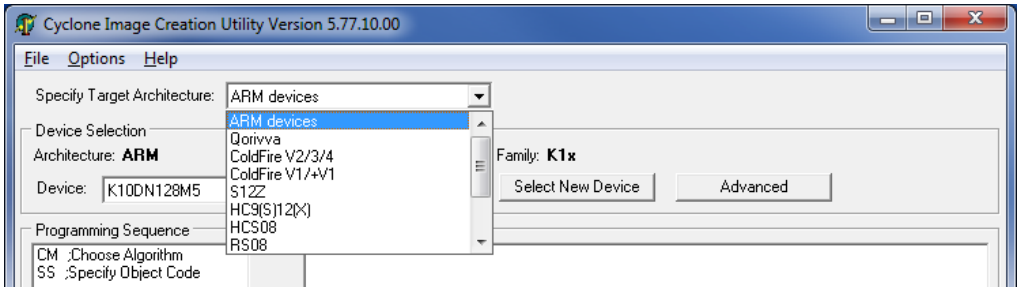


Figure 6-2: CPU Manufacturer Selection

The user may select the appropriate target architecture by clicking on "Select New Device." A Device Selection window will appear.

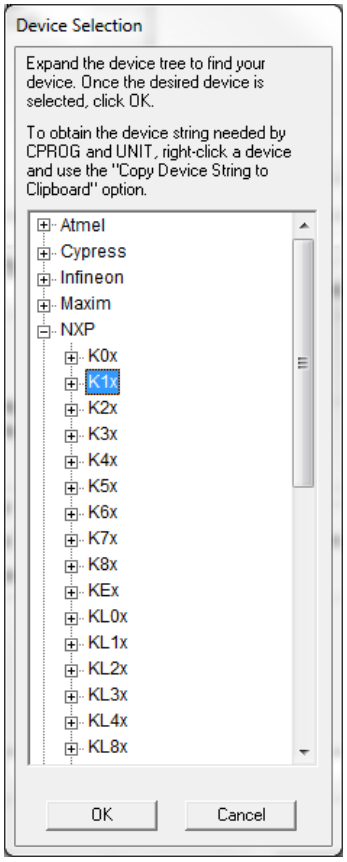


Figure 6-3: Device Selection

6.1.1.1 Security Settings - Qorivva (MPC5xxx) Only

If you selected Qorivva (MPC5xxx), the Cyclone Image Creation Utility will display an area called Security Settings (see **Figure 6-4**). If your Qorivva device supports uncensoring, click the “Device supports uncensoring” checkbox and select the appropriate bit depth for the device’s password (64-bit or 256-bit). The box to the right is where the password must be entered. Optionally you may use the Browse button to navigate to a text file that contains the correct password for the device. The contents of the text file that you select will automatically be used to fill the password text box.

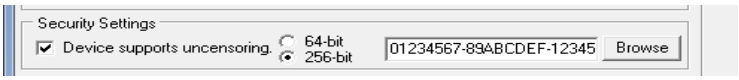


Figure 6-4: Security Settings - Qorivva Only

6.1.2 Specify Programming Script

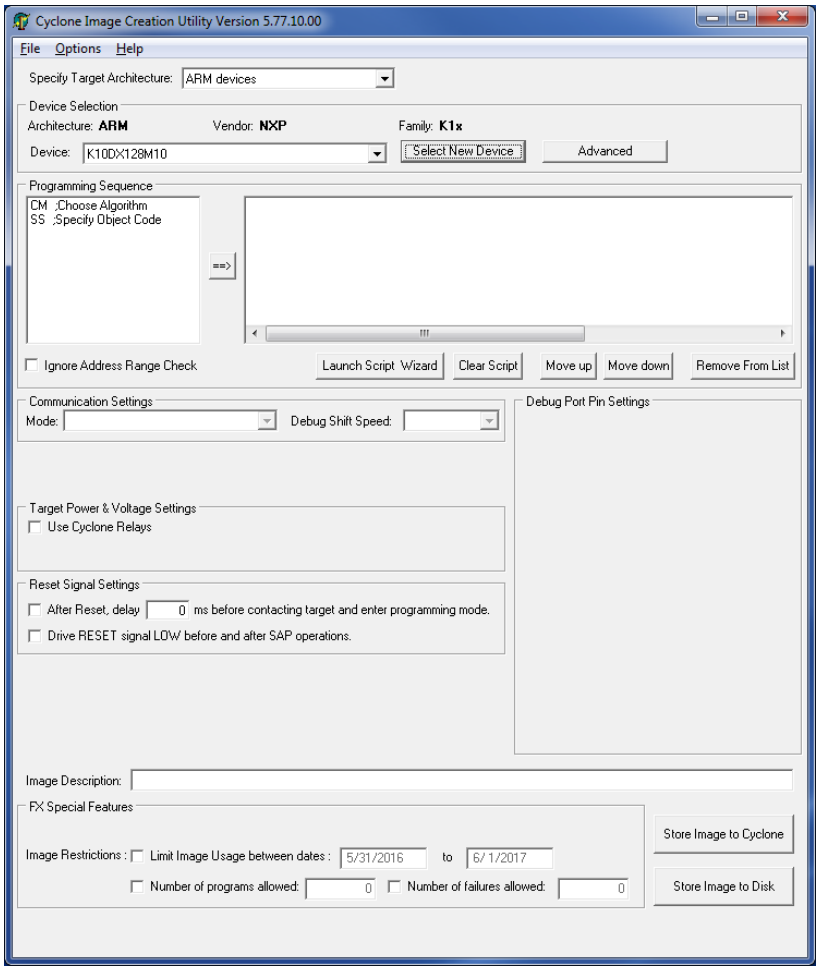


Figure 6-5: Specify Programming Script

This is a two-panel interface. The left panel provides a list of available programming functions. The right panel displays the ordering of the functions.

To specify the programming algorithm for the target, double-click on the Choose Algorithm (CM) function in the left panel. Or, you may highlight it and add it to the right panel using the arrow (->). This opens the Load Programming Algorithm dialog.

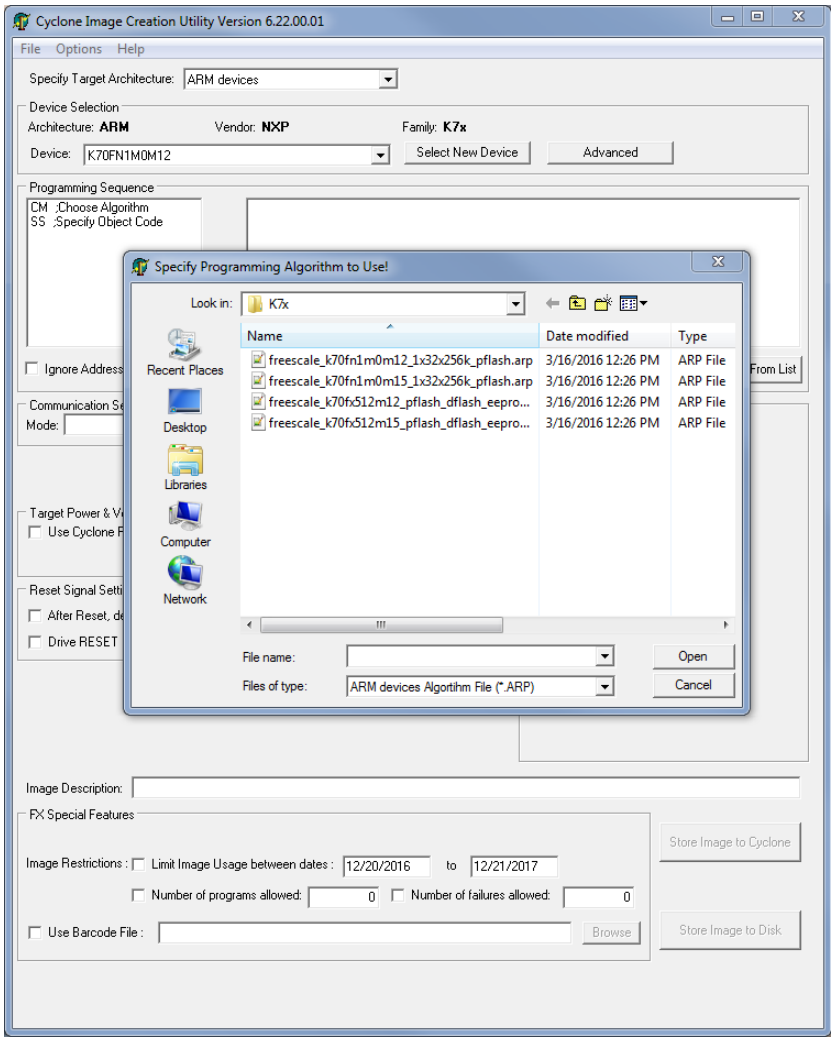


Figure 6-6: Load Programming Algorithm Dialog

Select the programming algorithm that you wish to use.

Similarly, to specify the S-Record to be programmed into the target, double-click on Specify S-Record (SS) in the left panel. This opens a dialog which allows you to select the appropriate S-Record.

Once both the algorithm and S-Record are selected, the full list of programming functions becomes available in the left panel.

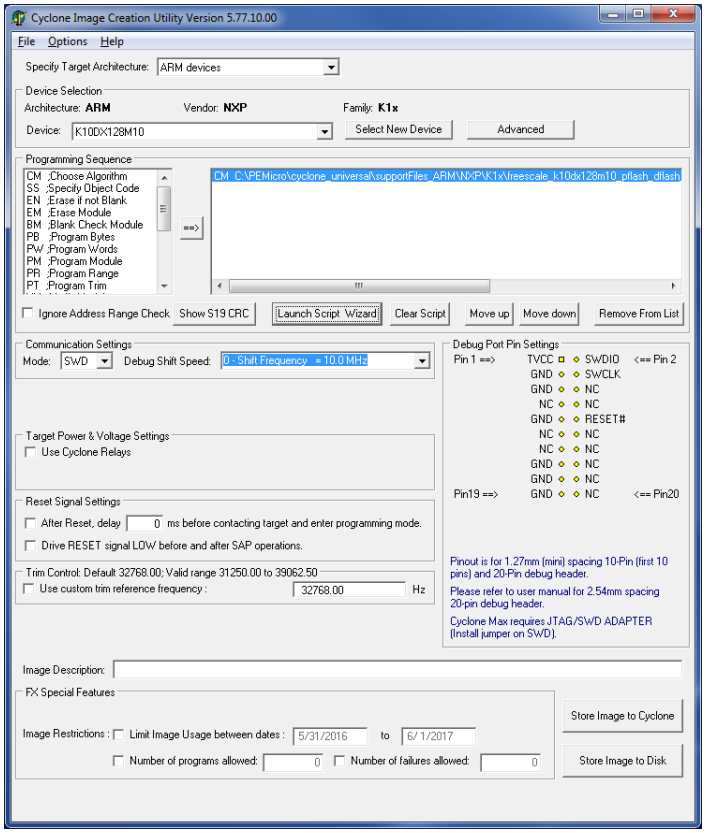


Figure 6-7: Programming Functions Enabled

Next, the user should add additional programming functions to complete the programming script.

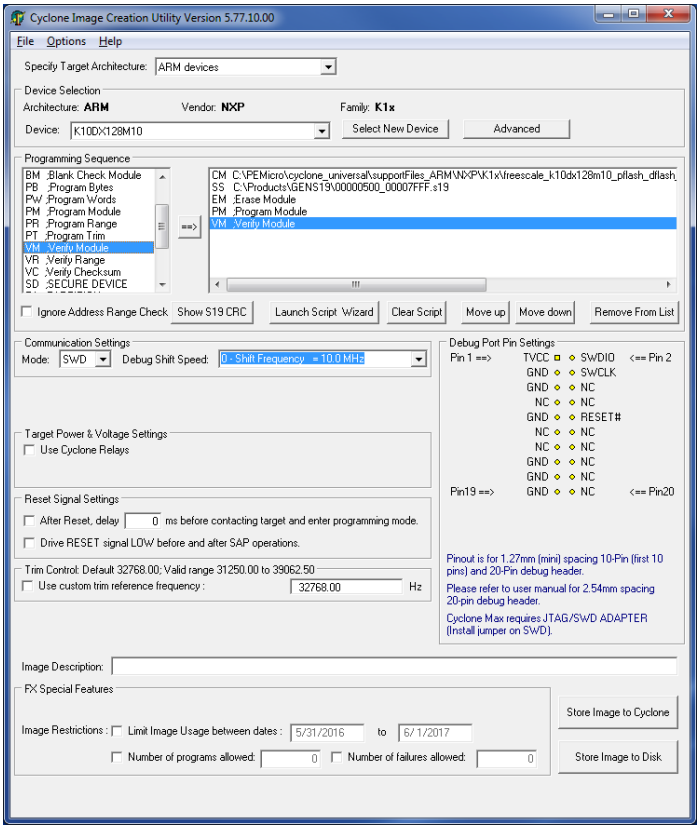


Figure 6-8: Programming Functions Complete

The Launch Script Wizard button prompts the user for a programming module, followed by an S-Record, and creates a default programming script. The user can then modify the programming sequence as needed.

The Clear Script button will remove all programming commands from the right panel.

The Move Up and Move Down buttons allow the user to manually re-sequence the order of the programming commands.

The Remove From List button can be used to remove a selected command from the right panel.

At this point the image can be saved to a disk or to the Cyclone device. For more information, please see **Section 6.1.8 - Store Image To Cyclone**.

6.1.3 Programming Operations

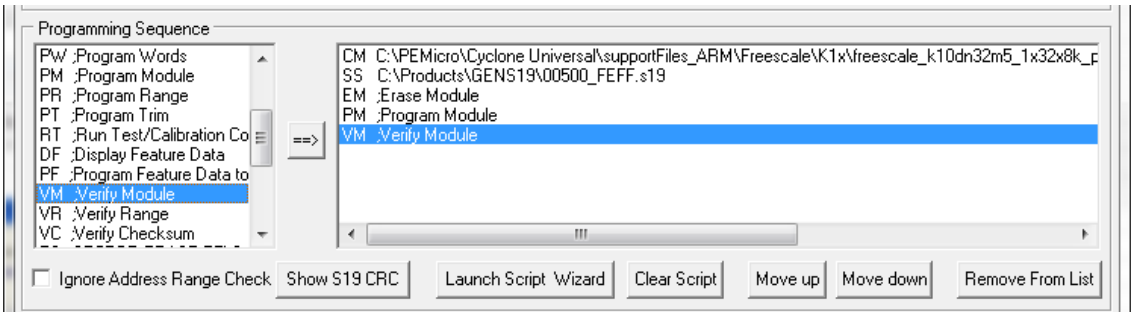


Figure 6-9: Programming Operations Dialog Section

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 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, 3.0, and 4.0 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 will erase the EEPROM/flash on the target device after entering the Monitor Mode or BDM mode.

Blank Check Module

If “Blank Check Module” is checked, the Cyclone will check to see if the flash/EEPROM on the target device is erased.

Program Bytes

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

Program Module

This command will program the selected S-record file into EEPROM/flash. For this command to work, you must have previously selected an S-record file.

Verify Module

This command will verify that the selected S-record file was programmed into the EEPROM/flash. 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. This command is typically much faster than performing a full Verify Module command.

Choose Serial File

This command becomes available once a programming algorithm is selected. It specifies the serial file that holds the serial numbers to be programmed to the target. Please reference **CHAPTER 14.1 - SAP_LAUNCH Introduction** for more information about programming serial numbers.

Program Serial Number

This command becomes available once a programming algorithm is selected. It will instruct the Cyclone to program the serial number to the target once executed. As with other commands, the serial number will not be programmed until the SAP operations are carried out. Please reference **CHAPTER 14.1 - SAP_LAUNCH Introduction** for more information about programming serial numbers.

Note: When using a barcode scanner as part of the programming process, a Barcode Test file must be included with the programming script of the SAP image. The “Use Barcode File” selection is enabled, and the exact file specified, in the FX Special Features section of this window. Please refer to **Section 6.1.7.2 - Use Barcode File**, and for more information on using a barcode scanner with the Cyclone FX please see **CHAPTER 10 - USING A BARCODE SCANNER TO SELECT AN IMAGE & INITIATE PROGRAMMING**.

6.1.4 Communication Mode and Rate Settings

CYCLONE FX programmers support multiple communication modes and communication rates. A user needs to select proper communication mode and rate from the drop down list after programming operations are specified. The debug connector pin definitions are listed for reference.

6.1.5 Target Voltage and Power Settings

A user may elect to use Cyclone to supply power to the target. In this case, the Target Voltage specifies the target MCU I/O voltage level.

The user needs to take into account the power discharge time for the Power Down delay. The reset driver delays, power stabilization time, and the target clock stabilization time should be considered for the Power Up delay.

A checkbox is available for a user to instruct the Cyclone to turn off target power after SAP operations. If unchecked, the target power will remain on.

The user has the option to provide Reset Delay if certain reset monitoring devices are used. The Cyclone will delay for the specified time after allowing the target out of reset.

6.1.6 Image Description

The Cyclone Image Creation Utility allows the user to summarize the purpose of current configuration for future reference. The description will be either programmed into the Cyclone or saved into an encrypted file.

The image description will appear on the touchscreen LCD for image identification. This field will not affect the Cyclone’s operations with the target.

6.1.7 FX Special Features

The **CYCLONE FX** includes additional security and other settings which may be configured here:

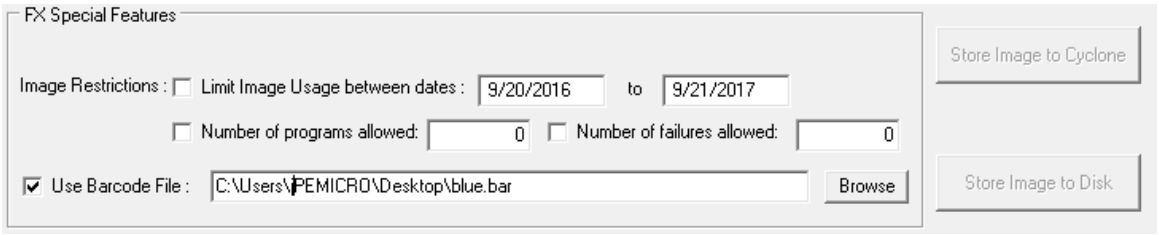


Figure 6-10: FX Special Features

6.1.7.1 Image Restrictions (Enhanced Security Settings)

There are any number of reasons why the user may want to place restrictions on the use of specific programming images on a Cyclone programmer: from added ease when managing production to a desire to protect intellectual property. When using the **CYCLONE FX**, the "FX Special Features" section of the Cyclone Image Creation Utility allows you to specify one or more restrictions and tie them to specific programming images. Even if restricted programming images are deleted from Cyclone's internal memory or an SD card, the Cyclone platform has a persistent memory that continues to tie security restrictions to that programming image. Thus, if an image is removed and re-added to a Cyclone, the image counts are maintained and would continue counting from where it left off. Also, if the SD Card is moved from Cyclone to Cyclone, the count is maintained in both Cyclones as well as the SD Card.

Every time an image is generated by the Cyclone Image Creation utility, it is encoded with a unique image ID number. All counts are stored relative to this unique ID number. So, when an image is regenerated in the Cyclone Image Creation utility, it will have its own counts which will not conflict with the previously generated image, even if the images are otherwise exactly the same. In this way, the user can regenerate an image to allow a new batch of targets to be programmed.

Note: The user may set more than one type of restriction on a programming image. The ability to program the image will be restricted by whichever triggers first. E.g., if the user creates settings to allow 100 programs, and also sets an allowed date range restriction, the ability to program the image will be restricted as soon as the first of these conditions is triggered.

Currently the user may set the following restrictions:

6.1.7.1.1 Limit Image Usage Between Dates

When "Limit Usage Between Dates" is checked and the start and end dates are specified with valid dates (format: DD/MM/YYYY), the Cyclone operator will only be allowed to program the corresponding programming image when the date is on or between the dates specified. The Cyclone has an onboard battery and clock which keeps a clock running even when power to the Cyclone is removed. This clock date is the one used for comparison to the UTD Date specified in the image. The ability to limit programming to a date is useful for making sure that an image will stop working after a period of time. This could be for security purposes, or to make sure that a new and updated image will need to be uploaded to the Cyclone after a period of time (for instance, to not allow a firmware more than a year old to be programmed onto a target).

6.1.7.1.2 Limit Number of Programs Allowed

When "Limit Number of Programs Allowed" is checked and a number is specified in the corresponding box (minimum = 1), the Cyclone operator will only be able to execute a number of successful programming operations of this programming image less than or equal to the number specified. The current programming count can be displayed on the main screen of the Cyclone or it can be seen on the image's statistics page (see **Section 6.1.7.1.4 - Image Restriction Statistics**).

6.1.7.1.3 Limit Number of Failures Allowed

When “Limit Number of Failures Allowed” is checked and a number is specified in the corresponding box (minimum = 1), the Cyclone operator will only be able to execute programming operations on the current image until the maximum number of errors specified has been reached. This restriction exists largely to prevent an operator from intentionally generating an error as part of the programming process in an attempt to circumvent the count restrictions. A recommended limit on this number would be on the order of 5% of the allowed programming counts.

6.1.7.1.4 Image Restriction Statistics

Statistics related to any specified restrictions for the currently selected programming image may viewed by navigating in the touchscreen menu to *Current Image Operations - Show Current Image Stats.* For more information on viewing programming image stats, see **Section 5.2.2.4 - Show Current Image Stats (FX Only)**.

In addition, the statistics for Number of Programs & Maximum Allowed can be set to display on the home screen by navigating in the touchscreen menu to *Configure Cyclone Settings -> Configure Screen -> Configure Home Screen*. For more information on how to configure the Cyclone’s home screen, see **Section 5.2.3.5.4 - Configure Home Screen**.

6.1.7.2 Use Barcode File

PEmicro’s Cyclone FX programmers can be configured to use a bar code scanner (connected to the extension port) as part of the programming process. During setup the user will create a Barcode Test file using the provided Barcode Test Generator utility.

Here, the user should then check the Use Barcode File checkbox if they wish to include a Barcode Test file in the programming script of their SAP image. The user can then Browse to the file they wish to select. This is required when using the bar code scanner as part of the programming process. For more information on how to incorporate a barcode scanner into the Cyclone’s programming process, please see **CHAPTER 10 - USING A BARCODE SCANNER TO SELECT AN IMAGE & INITIATE PROGRAMMING**.FX Special Features

6.1.8 Store Image To Cyclone

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

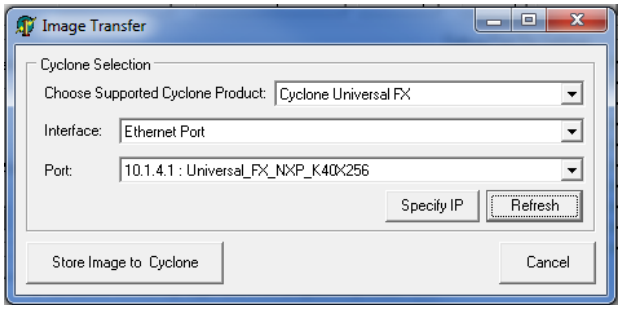


Figure 6-11: Image Management And Transfer Dialog

The Interface drop-down list allows the user to select one of three Serial, USB, or Ethernet interfaces. 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.

6.1.9 Store Image To Disk

“Store Image To Disk” allows the current configuration to be saved onto the hard drive. The image can then be transferred to the Cyclone’s internal flash (or an installed SD card) via the Manage Images Utility.

6.1.10 Save Cyclone Configuration

"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 contents with the file to see if they are the same.

6.1.11 Load Cyclone Configuration

"Load Cyclone Configuration" in the file menu allows the user to load a configuration that has previously been saved in order to create a new image.

6.2 Manage Multiple SAP Images

The Cyclone Image Management Utility, shown in **Figure 6-12**, allows the **CYCLONE FX** to store and manage multiple images in the Cyclone's internal memory, and on any compatible SDHC cards that are loaded into the SDHC port. Once the programming images have been created and saved to the disk using the Create Image utility, they may then be loaded collectively onto the Cyclone.

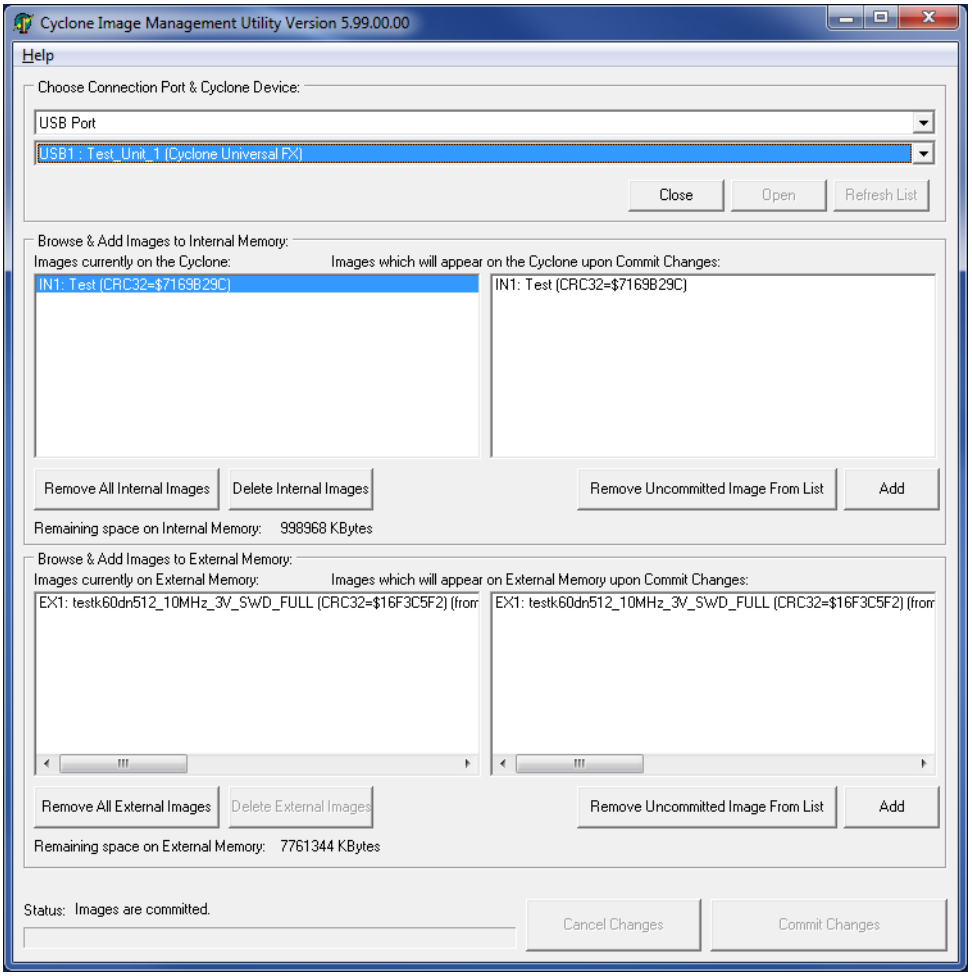


Figure 6-12: Manage Images Utility

Upon opening a selected **CYCLONE FX** in the Image Management Utility, the user is provided in the top left panel with a list of the images currently on the unit's internal memory. A list of images on any installed SDHC card will also be displayed in the bottom left panel. The corresponding Commit Changes panels on the right side are where you may prepare changes that you wish to make to these contents by using the Add and Remove buttons beneath the panels.

6.2.1 Delete Images From Internal/External Memory

Any images that are already stored on the **CYCLONE FX** or installed SD card can be deleted by

selecting the image in the left Current Images panel and then pressing the corresponding "Delete" button or the Delete key on the keyboard. You will be asked to confirm. Once confirmed, the image will be deleted (no need to press the Commit Changes button).

6.2.2 Add/Remove Images From The Commit Changes Panels

The Commit Changes panels to the right can be used to prepare a list of image changes by using the Add/Remove buttons beneath the panels to add and remove images form the list.

The user may also drag and drop image files into the Commit Changes panels on the right.

Note: No actual updates will occur to the Cyclone's internal/external memory or installed SD card until the user selects Commit Changes.

6.2.2.1 Commit Changes

Once the images that you wish to load appear in the panels on the right, you **must** press "Commit Changes" to update the Cyclone accordingly.

Note: Any SAP images that are already stored on older Cyclone models such as the Cyclone PRO, MAX, Renesas, STMicro or Cyclone ACP Rev. A/B (or on a CompactFlash card in one of those units, if applicable) can only be removed by using the appropriate "Remove All Images" button.

7 STAND-ALONE PROGRAMMER MANUAL CONTROL

The **CYCLONE FX** must be configured before it can serve as a Stand-Alone Programmer. The user may manually control the Cyclone via the LCD touchscreen menu and/or the Start button, or via PC software. The target power management schemes remain the same for each control method.

7.1 Operation Via Start Button

There is a Start button on the top of the Cyclone which is used for stand-alone programming. It is specified as follows.

Button	Function
START	Start executing the tasks pre-configured into the Cyclone.

7.1.1 LED Indicators

The Cyclone has two (2) LEDs to indicate the current operation stage.

LED	FUNCTION
Error	The Cyclone failed to execute the functions as instructed.
Success	The Cyclone executed the functions successfully.

7.1.2 Procedure via Start Button / LEDs

The following steps must be followed in order for the Cyclone to operate properly after it has been configured:

1. Turn off the target power supply if the “POWER IN” Jack is adopted.
2. Turn off the Cyclone system power.
3. Set the correct Power Management jumper settings.
4. Connect the target power supply to the “POWER IN” Jack, if applicable.
5. Connect the “POWER OUT” Jack to the target board power, if applicable.
6. Connect the ribbon cable to the target board debug connector.
7. Turn on the Cyclone system power.
8. Turn on the target power supply, if applicable.
9. Press the “START” button on the Cyclone.

When the “Success” LED lights up, you have successfully programmed your target.

7.1.3 Example

After the user programs the contents and procedures into the Cyclone’s on-board flash, the Cyclone may be used as a Stand-Alone Programmer. Suppose the user wants to perform the following instructions for a target device:

1) Erase Module
2) Program Module
3) Verify Module.

If the Cyclone is providing power to the target board, the “Target Power” icon will illuminate on the LCD display.

The Cyclone will then perform the operations. If they are performed successfully, the “Success” LED will be illuminated. One stand-alone programming cycle will have just been completed.

7.2 Operation Via LCD Touchscreen Menu

Once the **CYCLONE FX** is configured for stand-alone programming it may be operated by making selections from the touchscreen LCD menu. This section describes the menu functions that allow the user to easily execute stand-alone programming functions using the touchscreen LCD.

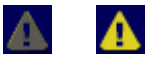
7.3 Home Screen

The home screen appears when the Cyclone is powered on, or when the  Home button is tapped.

7.3.1 Icons

A row of icons in the upper right corner indicates the status of various attributes of the Cyclone.

Note: The user may tap on the row of icons to view the meaning of each of the currently displayed icons.

Cyclone Unit Status: Ok / Bad 

Programming Status: Ready / Busy 

Target Power Relays: On / Off 

USB-To-PC Enumerated: Yes / No 

Real-Time clock Enabled & Working: Yes / No 

Cyclone Power Relays: Closed / Open 

Target Device Is Powered*: Yes / No 

SDHC Memory Card: None / Valid / Unformattted / Reset Cyclone** 

Barcode Scanner: Detected / Not Detected 

* Target Device Is Powered - "Yes" indicates that the **CYCLONE FX** detects power on the Vcc pin of the target device programming header.

** SDHC Memory Card - "Reset Cyclone" indicates that the **CYCLONE FX** needs to be reset before the SDHC card will register as Valid. The user can push the Reset button which is located on the front side of the Cyclone, below the LED indicators.

7.3.2 Configurable Display Area

The main area of the home screen can be configured to optionally display the following information, by using the Cyclone IP Configuration Utility (see **Section 9.6.1 - LCD Home Screen Display Selection**):

1. Firmware version of the Cyclone (always shown).
2. IP address assigned to the Cyclone.
3. Name assigned to the Cyclone.
4. Number of programming images in the Cyclone's memory.
5. Name of the selected programming image.
6. First serial number associated with the selected image
7. Current status.
8. Results of the last operation performed.
9. Time and date.
10. Status Window and Main Menu button (always shown).
11. Target voltage and/or current
12. Programming count & limit

7.4 Status Window

The status window appears in the lower left corner of the home screen and displays the results of programming operations.

7.4.1 Error Information Icon

When the Cyclone experiences an error during programming operations, the Info icon will appear to the left of the Menu button (or AUX button, if configured).

Info Icon: 

Press the Info icon to view a detailed description of the error.

7.4.2 AUX Button (Appears If Configured)

The Cyclone allows the user to add an Auxiliary (AUX) button to the home screen which will perform a specific function when pressed. The specific function is chosen by the user when the AUX button is configured. The AUX button will appear on the home screen to the left of the “Menu” button, in the lower right corner of the home screen.

Figure 7-1: AUX Button On Home Screen (configured for perform CRC32 function)

For information on how to configure the AUX button, see **Section 5.2.4 - Status**.

7.4.3 Main Menu

The Main Menu is accessible by pressing the “Menu” button when the Home Screen is displayed. The Main Menu screen contains four selections. From these, select “Current Image Options.”

Current Image Options	Execute Image Function	*Launch Programming*	-
		Verify Data In Target	-
		Toggle Power	-
		Power cycle device to run user code	-
		Validate Image CRC32	-
	Set Image Validation	*Enable Validate IMG*	-
		Disable Validate IMG	-
	Modify Next Serial Number	*Image S/N:*	-
		Decrease Next Serial (will specify "not allowed" if no serial # attached to IMG)	-
		Increase Next Serial (will specify "not allowed" if no serial # attached to IMG)	-
		Current Image ID Selected:	-
		Current Alg ID Selected:	-
		Current CS ID Selected:	-
	Show Current Image Stats (FX Only)	-	-

Figure 7-2: Touchscreen LCD Menu - Standalone Functions Highlighted

The menu selections in “Current Image Options” will allow the user to execute programming operations, verify data, toggle power, validate the programming image, and modify the upcoming serial number if necessary.

7.4.3.1 Execute Image Function

Execute Specific SAP Function presents four Stand-Alone Programming functions that you may execute by tapping the function that you wish to execute:

7.4.3.1.1 Launch Programming

This allows the user to execute the programming function. The Cyclone will program the target device, if able, using the currently selected programming image. This is functionally equivalent to pressing the Start button.

7.4.3.1.2 Verify Data In Target

Performs a verify function on the data that has been programmed into the target device.

7.4.3.1.3 Toggle Power

Toggles the target power and makes sure all ports are driven to debug mode level.

7.4.3.1.4 Power Cycle Device To Run User Code

Toggles the target power and maintains tri-state mode for all signals.

7.4.3.1.5 Validate Image CRC32

Allows the user to perform a CRC32 validation on the currently selected programming image.

7.4.3.2 Set Image Validation

Allows the user to choose between two validation settings: 1) validate the image each time the Start button is pressed, or 2) do not validate the image.

7.4.3.3 Modify Next Serial Number

Presents options that display the current serial number and allow the user to increase or decrease the next serial number. Tap “Current Image ID Selected” to view/choose the desired programming image; tap “Current Alg ID Selected” to view/choose the desired programming algorithm; use “Current CS ID Selected” to view/choose the desired Choose Serial file. The adjustment buttons will display “Increase Not Allowed” and “Decrease Not Allowed” if the image/algorithm/CS files that the user has selected to do not allow for this operation.

7.4.3.4 Show Current Image Stats

Displays current statistics, if any, for Image Programmed Count & Maximum Allowed, Errors Logged & Maximum Allowed, and Date Range Allowed. These limits can be set in the Image Creation Utility.

Note: When *Current Image Stats* is displayed as a home screen item, only Image Programmed Count & Maximum Allowed are displayed on the home screen.

8 STAND-ALONE PROGRAMMER AUTOMATED CONTROL

Users who wish to automate control of one or more Cyclone units have several options available. This chapter presents a brief overview of those options along with some additional information about each.

8.1 Cyclone Automated Control Package - Overview

Every Cyclone includes the Basic Edition of PEmicro's Cyclone Automated Control Package. PEmicro also offers advanced versions of the control package which may be purchased separately.

8.1.1 Basic Edition

The Basic Edition, included with each **CYCLONE FX** programmer, gives users two options for automating control of the Cyclone.

1. Cyclone Launch Application

   The Cyclone Launch application allows control of **one or more** PEmicro Cyclone units through the usage of simple **batch and script files**.

2. Dynamic Link Library (.DLL)

   The .DLL in the Basic Edition allows custom software applications to control **one** Cyclone unit.

Users wishing to use a .DLL to control more than one Cyclone or who would prefer to use RS232/ Ethernet protocols (e.g., in a non-Windows environment) may purchase the appropriate advanced version of the Cyclone Automated Control Package.

8.1.2 Professional Edition

The Professional Edition is available separately. It contains a dynamic link library **(DLL)** which allows custom software applications to control **up to three** Cyclone units.

8.1.3 Enterprise Edition

The Enterprise Edition is available separately. It contains a dynamic link library **(DLL)** and the ability to use custom software, **RS232 and Ethernet** communication protocols to control an **unlimited** number of Cyclones.

More details about what is included with each edition of the Cyclone Automated Control Package may be found at www.pemicro.com.

8.2 Cyclone Automated Control Package - Details

This section presents brief descriptions of the Cyclone Launch Application, .DLL, and RS232/ Ethernet options that are offered by PEmicro's various Cyclone Automated Control Packages. Detailed operational instructions for these tools are beyond the scope of this manual. For operational instructions, please consult PEmicro's *Cyclone Automated Control Package - Developer's Manual*, which accompanies the Basic Edition of the Cyclone Automated Control Package. This manual may also be downloaded from www.pemicro.com.

8.2.1 Cyclone Launch Application

The Cyclone Launch application is included with every edition of the Cyclone Automated Control Package. It allows a developer to use simple ASCII script files to control Cyclone operations from the PC. Once the script files are configured, a simple batch file can be created to initiate programming operations on one or more Cyclone units. Cyclone Launch also supports features such as stand-alone image maintenance and dynamic data programming, which provides a powerful but easy-to-use interface. The Cyclone Launch application is ideal for getting your automated production environment up and running in a very short amount of time.

8.2.2 .DLL Control

The dynamic link library (DLL) that is included in all editions of the Cyclone Automated Control Package allows you to create an application on the PC that can directly control one (Basic Edition) or more PEmicro Cyclone units. Please see **Section 8.1 - Cyclone Automated Control Package - Overview** for the number of Cyclones that may be controlled by each version of the package.

These interface routines are designed to be compiled into visual and non visual applications running on Windows 95, 98, ME, NT, 2000, XP, Vista, 7, 8, 10. The actual interface routines are located in the “CYCLONE_CONTROL.DLL” 32 bit DLL file. The DLL is callable from almost any 32-bit Windows development environment. Since the way the DLL is called varies depending on the compiler used, you are provided with the DLL interface code and sample applications for each of the following compilers:

Borland Delphi 2.0+ (Pascal) - Visual Application

Microsoft Visual C++ 5.0+ - Visual MFC Application

Microsoft Visual C# 2005+ - Visual Application

These sample applications come with project and workspaces defined for ease of use. Simply open the project/workspace in your compiler and you should be able to build the sample application without any modifications. The sample applications come pre-compiled with ICONS, so you can run them before jumping into the code.

8.2.3 RS232 / Ethernet Communication Protocols

The RS232 and Ethernet Communication protocols included with the Enterprise Edition of the Cyclone Automated Control Package allow a developer to manually send individual command packets to control each Cyclone unit. This is ideal for setups that do not have access to a PC or production environments that do not run Windows-based computers.

9 ETHERNET CONFIGURATION

This section describes the mechanism used by the Cyclone 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.

9.1 Network Architectures

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. 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 the data incoming on one port 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 that 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 type 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, use 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 mechanism, with one very important caveat: it needs to be set up with the appropriate network parameters for the underlying network architecture.

9.2 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 a 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.

Assignation 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 consists of two more IP numbers. The first of these is called the Subnet Mask, and is used to determine whether or not the destination address 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 the "Configuring the Cyclone" section of this manual.

9.3 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, while 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.

9.4 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.

9.4.1 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:

	IP Number	Gateway IP	Subnet Mask
PC1	192.168.100.1	192.168.100.3	255.255.255.0
PC2	192.168.100.2	192.168.100.3	255.255.255.0
CYCLONE	192.168.100.4	192.168.100.3	255.255.255.0
Gateway	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.

9.4.2 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 PEmicro, 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

For more information on programming these IP numbers into the Cyclone device, please see the following section.

9.5 Cyclone IP Setup Via LCD Menu

When the user is connecting the Cyclone via Ethernet, before the connection is established between the Cyclone and the network the menu's Home Screen will display the Cyclone's IP address as 0.0.0.0.

Once a connection has been established, the menu's Home Screen displays the Cyclone's IP address and connection setting (Static or Dynamic).

The Ethernet cable can either be attached at the start of Cyclone startup or connected after setup is complete. The connection with the network will be established when the cable is connected. If the Ethernet cable is disconnected after setup is complete, the user should be able to simply reconnect the cable to reestablish networking. However, depending on the setup of the DHCP server, if the Ethernet cable is left unplugged for a considerable time the IP address may expire and connection will have to be set up once again. This can be accomplished by restarting the Cyclone.

9.5.1 Configure Network Settings

To configure network settings for the Cyclone, navigate to the following Menu location:

Main Menu / Configure Cyclone Settings / Configure Network Settings

The following options will be available under Configure Network Settings:

- Show Current IP Settings
- Edit Static IP Settings
- Enable/Disable Dynamic IP
- Edit Cyclone Name

9.5.1.1 Show Current IP Settings

Show Current IP Settings displays the current IP settings, including:

- Current IP Mode
- IP Number
- Mask
- Gateway

- MAC Address

If you are in Static IP mode, these settings (excluding the MAC address) may be changed by tapping on them. In this case a tap will take you to the Edit menus. If you are in Dynamic IP mode, tapping will show a message that the Cyclone settings cannot be changed.

Dynamic vs. Static

There are two schemes for assigning IP addresses. One is the Static IP addressing mode. This involves the user manually setting the IP address for every device on the network. In this case, it falls to the user to ensure the IPs assigned do not conflict and are within the boundaries of the network. The other is the Dynamic Host Configuration Protocol (DHCP). This involves setting up a separate server to manage the IP addresses. The server is given a list of valid IP addresses for the network. Using a predetermined set of rules, each new device that wishes to connect to the network is given an IP address by the server. This takes the task of managing the validity and uniqueness of IP addresses out of the user's hands and relegates it to the server. **CYCLONE FX** programmers are capable of using either Static IP addressing or DHCP.

Note: The current IP settings may also be viewed/edited by navigating to:

Main Menu / Status / Show Current IP Settings

9.5.1.2 Edit Static IP Settings

This allows editing of IP, Mask, and Gateway in Static IP mode. In the edit dialogs, the user must enter a valid IP address to continue:

Format

xxx.xxx.xxx.xxx

Where:

0 <= xxx <= 255

9.5.1.3 Enable/Disable Dynamic IP

Opens a dialog to toggle the IP settings between Static and Dynamic. Once an option is selected a message is displayed indicating that the Cyclone must be reset for this option to take effect. The reset button on the front side of the Cyclone may be used.

9.6 Cyclone IP Configuration 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 with network parameters.

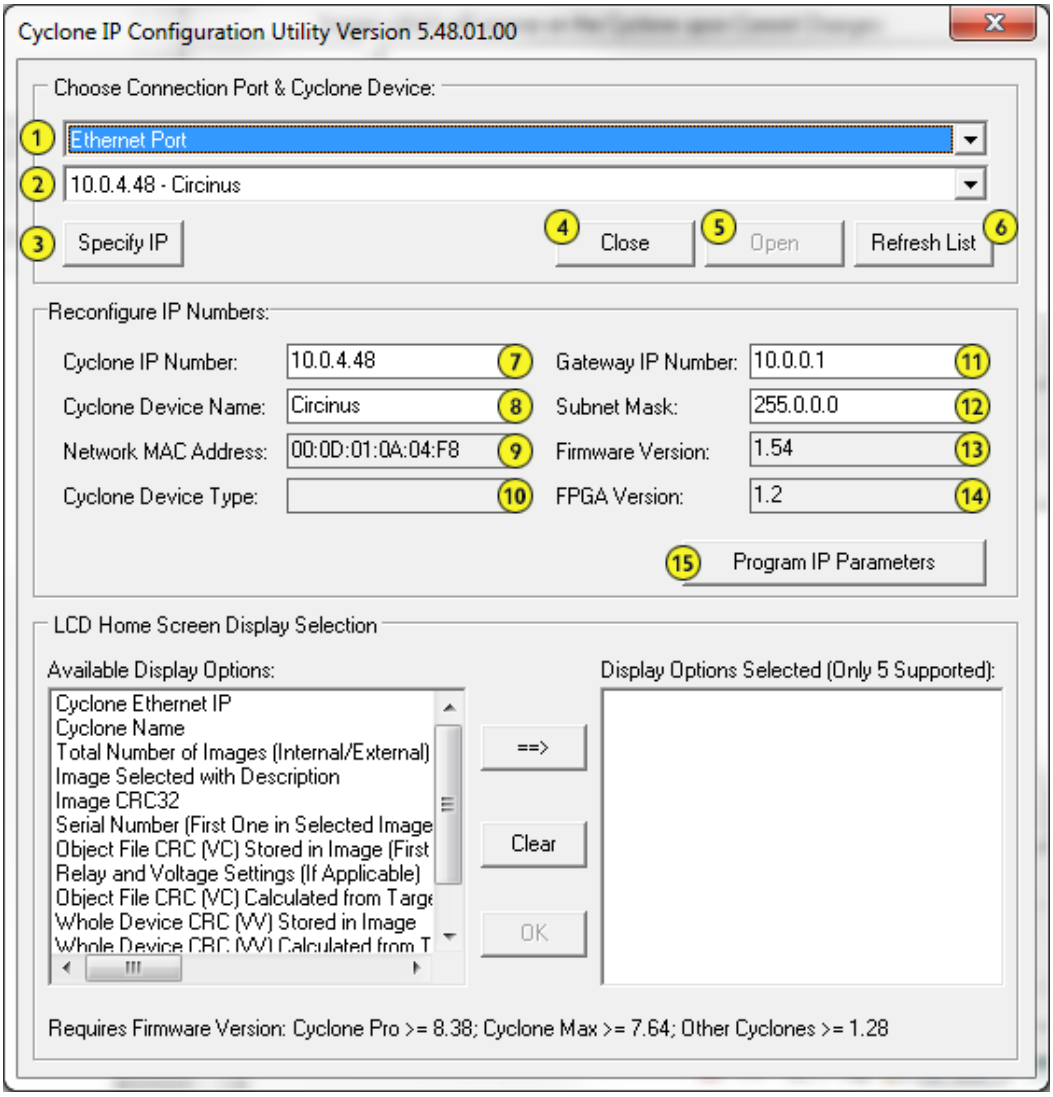


Figure 9-1: Cyclone IP Configuration Utility Default Screen

(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) Specify IP

Opens the **Specify Cyclone IP Address** dialog. This dialog allows the user to manually add and/or remove Cyclone Ethernet IP addresses from the list of IP addresses that is displayed when the user chooses a specific Cyclone from those available. The drop-down list in the “Specify Cyclone Ethernet IP to Remove From Registry” section shows the currently displayed list of IP addresses.

If a Network Card other than the default is desired, the user may set which Network Card to use by specifying the Ethernet IP of the Network Card and clicking “Set Network Card To Use.” Otherwise this field should be left blank.

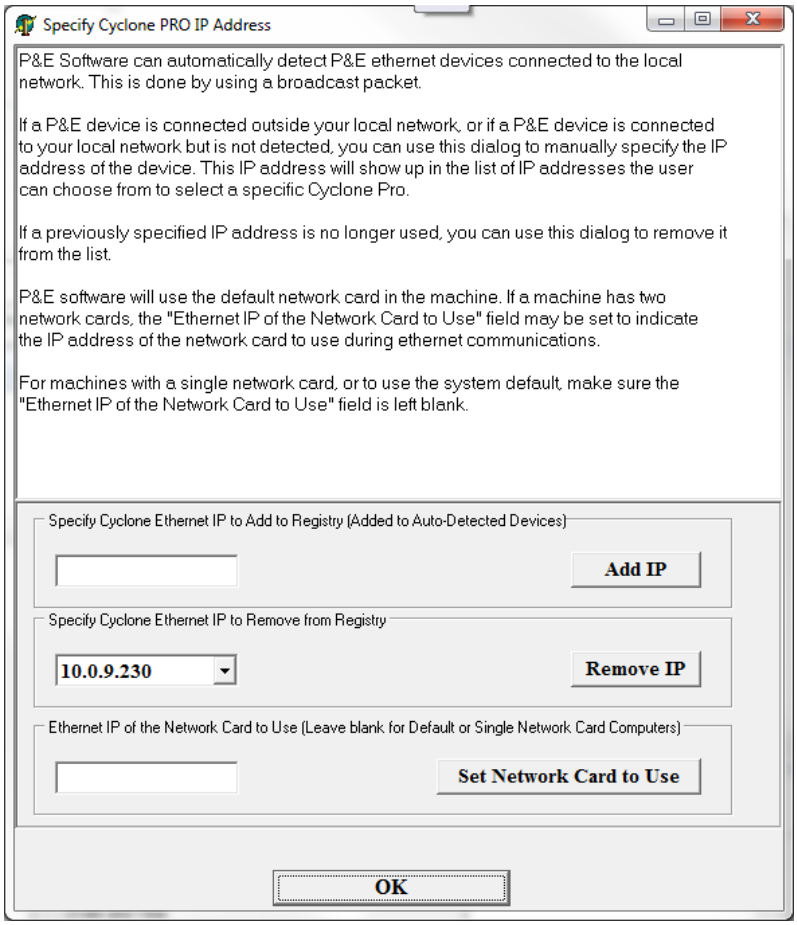


Figure 9-2: Specify Cyclone IP Address Dialog

(4) 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.

(5) 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.

(6) 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.

(7) Cyclone IP Number

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

(8) Cyclone Device Name

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

(9) MAC Address

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

(10) Cyclone Device Type

This displays the type of Cyclone hardware.

(11 Gateway IP Number

The IP number of a gateway on the network.

(12) Subnet Mask

The subnet mask of the network.

(13) Firmware Version

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

(14) FPGA Version

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

(15) Program Cyclone Parameters

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

9.6.1 LCD Home Screen Display Selection

The lower area of the Cyclone IP Configuration Utility contains the LCD Home Screen Display Selection Area. This allows users to configure the Cyclone to display specific items on the Cyclone's home screen. Users may select items from the Available Display Options window and use the arrow button to add them to the Display Options Selected window. Up to 7 items may be selected. Press OK to save the selections, or press Clear to clear them. Information corresponding to the selected items will be displayed on the home screen.

9.7 Using Cyclone IP Configuration Utility To Configure The Cyclone

Before the Cyclone 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 FX** software distribution.

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

1. Connect a Cyclone 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 Configuration Utility starts up with the following screen:

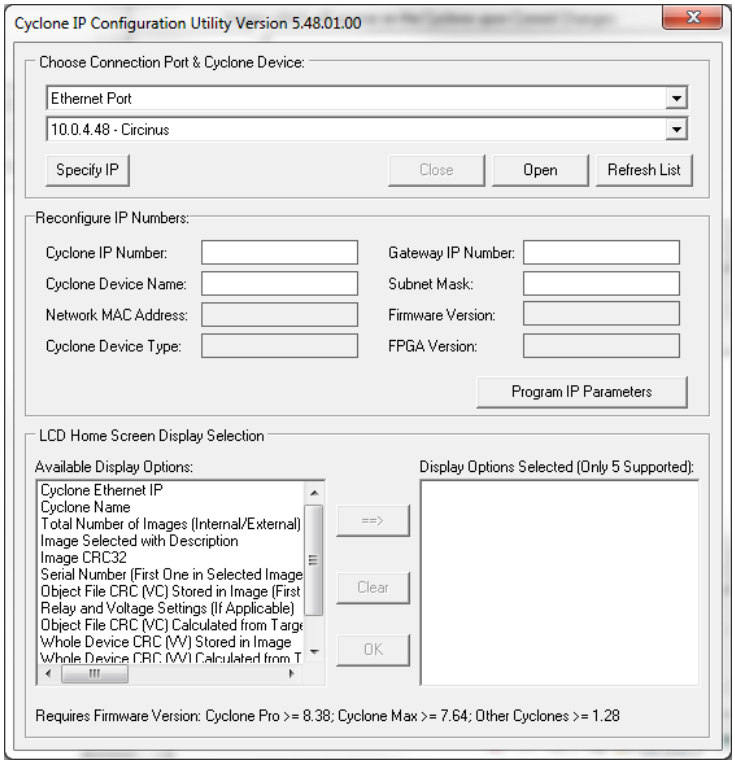


Figure 9-3: Cyclone IP Configuration Utility - Initial Screen

2. Assuming that the Cyclone 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:

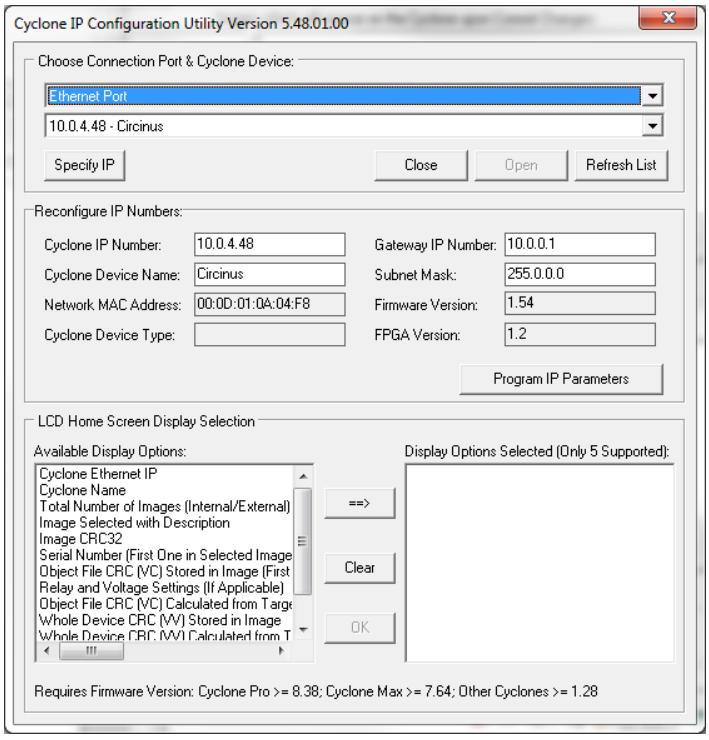


Figure 9-4: Cyclone IP Setup Utility - Continue Setup

3. The Cyclone 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.

10 USING A BARCODE SCANNER TO SELECT AN IMAGE & INITIATE PROGRAMMING

10.1 Introduction

PEmicro's CYCLONE FX programmers are capable of using a barcode scanner during stand-alone programming. Scanned barcodes can be used to automatically select and program a specific SAP (Stand Alone Programmer) image into a target. This means the programming image does not need to be pre-selected before programming. Also, there is no need to hit the start button. Simply scan the bar code and programming will be initiated.

The Cyclone FX scans the barcode and checks all resident images for a barcode match. If exactly one match is found, the image is selected and used to program the target. If no matches are found, or multiple matches are found, an appropriate error is reported. This reduces manual configuration errors especially when large numbers of programming images and product types may be programmed.

Automatic selection and launch of a specific flash programming image based on a scanned barcode can improve the speed and accuracy of production programming, especially when there is a varied product mix being programmed. Barcode scanning improves accuracy by making the process of selecting a programming image fast, automatic, and less vulnerable to user error.

The barcode itself can optionally be programmed into the target device's memory as part of this manufacturing process. The information that the barcode makes available to the device can make it easier to trace products, track product hardware versions, or provide a way to serialize production. This results in a more efficient manufacturing process.

10.2 Scanning Procedure

After the CYCLONE FX is configured (see **Section 10.4 - Enabling Barcode Scanner In Cyclone Menu**), the barcode scanner is connected to the Cyclone's host USB port. The scanner is used in Wedge output mode, which emulates a USB keyboard. When the scanner scans a barcode, it transmits the barcode to the Cyclone. The Cyclone then analyzes the barcode and uses it to select a programming image.

To perform this analysis, all programming images on the Cyclone, in both internal and external memory, are reviewed. Each image may contain a set of rules to determine whether a scanned barcode corresponds to that particular image. This set of barcode analysis rules is referred to as an image's "barcode test." If the barcode passes the test on one and only one of the programming images on the Cyclone, that image is automatically selected and programming is initiated. Otherwise, if the scanned barcode does not pass the barcode test on any image, or if it passes the test on more than one image, an error will occur.

Barcode test rules include: barcode length, character type, specific characters, and numerical ranges. Fixed barcodes or barcodes which are unique to an individual product can be properly analyzed and used to select the appropriate programming image.

A logfile is generated to help the user understand how a scanned barcode auto-selects a specific image on the Cyclone (or yields an error). See **Section 10.7 - Troubleshooting** for more information on this logfile.

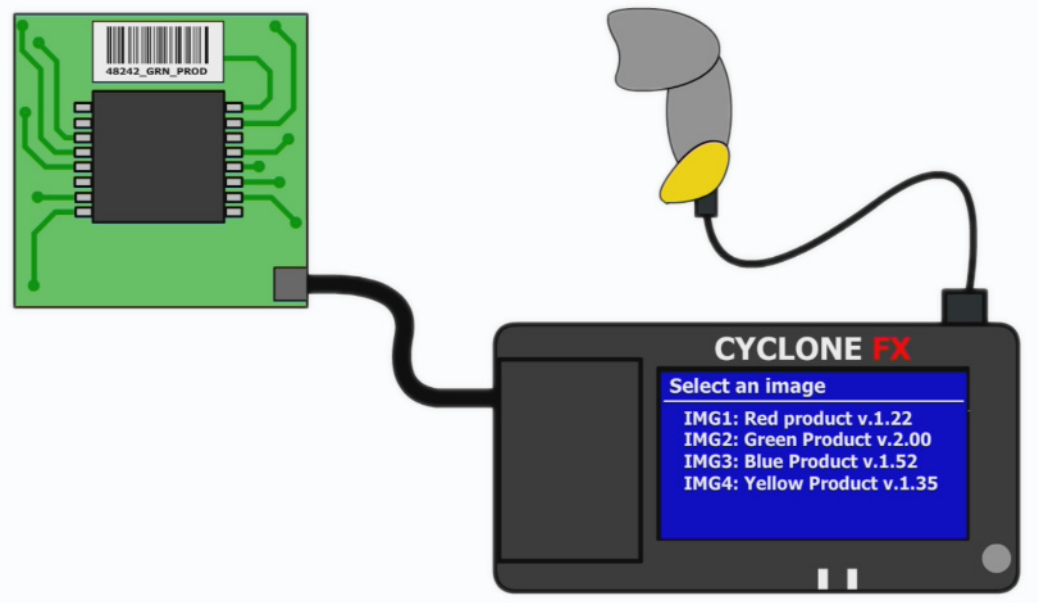


Figure 10-1: CYCLONE FX With Barcode Scanner

10.3 Potential Benefits Of Programming Via Barcode Scan

The use of a barcode scanner as part of the manufacturing process can boost productivity and reduce human error. When the user scans a barcode to select and launch the programming sequence, it frees that user from having to know any details about the product being programmed; point, shoot, and it programs. Programming multiple products with different data can be as simple as connecting each new target board and scanning a barcode that is printed on it.

A manufacturer might, for example, design their products such that every product has a unique barcode printed on it. Part of these barcodes might be a fixed string, and part of them might be changing (for example :WIDGET1-A00143, WIDGET1-A04325, WIDGET1-B03222). In this scenario, the barcode test for the WIDGET1 programming image could check that the barcode starts with the fixed string “WIDGET1-” (product identifier), the next character is an A or B (product version), and that the rest of the barcode is numeric (unique ID). If it passes, the WIDGET1 image would be selected and the target programmed.

This same manufacturer might also set up the programming process for each product to program the barcode itself at a specific point in memory. There are then many ways a device could leverage its barcode number. For example, the running application in the device could check the product version of the barcode to understand whether it is running on hardware revision A or B and make decisions based upon that information. The Unique ID could also be used for this purpose. E.g., if the number is less than 5000, then use high drive strength on the product's port X; otherwise use low drive strength on the product's port X. When connected later to the cloud, it can report its unique identifier back to its home base. Potentially it could use the barcode's Unique ID as part of setting up its MAC address for Ethernet.

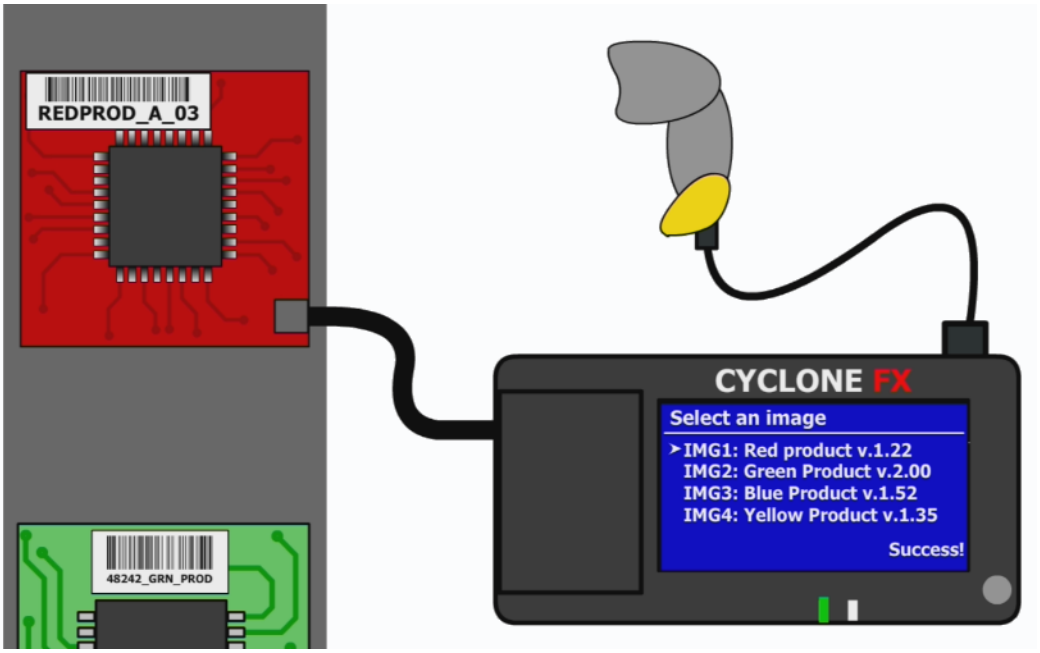


Figure 10-2: Bar Code Scanner On Production Line

In a completely different programming scenario the manufacturer might not want to place barcodes on the products themselves. E.g., it could be that they do not frequently switch from one product to another on the production line. In this case, they might prefer to create a separate set of directions for each product they manufacture which covers programming, test, and casing of the product. They could then include a fixed barcode as part of the programming instructions. The instructions might state that, before starting a production run of this product the barcode should be scanned by the Cyclone programmer. This removes the manual step of the operator choosing which image to use for each product. The Cyclone selects the image automatically by using the barcode in the instructions (and if there is not a unique match, an error is reported). This can help to reduce user error, particularly if the programming images on the Cyclone carry similar names.

10.4 Enabling Barcode Scanner In Cyclone Menu

To use the barcode scanner with a CYCLONE FX, the scanner must first be enabled in the Cyclone menu by navigating to “Configure Cyclone” and then “Configure USB Host Device.” The user should click on “Enable USB Scanner”. The barcode scanner's instructions should be consulted to understand how to put the scanner into Wedge (keyboard emulation) mode, and to make sure that it terminates the barcode with a CR (carriage return) character. The scanner can then be plugged into the USB host mini connector on the side of the Cyclone. A USB-to-USB-Mini adapter may be necessary if the scanner uses a full size USB connector. When plugged into the Cyclone, the scanner will power up and the Scanner Active icon on the Cyclone LCD screen will illuminate.

Barcode Scanner Active icon: 

10.5 Creating A Barcode Test

This section demonstrates one simple example of how to create a barcode test by using the Barcode Test Generator utility. For in-depth instructions on how to use the Barcode Test Generator, please refer to **CHAPTER 10 - BARCODE TEST GENERATION**.

Whenever a programming image is created the user has the option to add a barcode test to the image. PEmicro provides a Barcode Test Generator utility to enable the user to creating these tests. The following steps demonstrate how to create a sample barcode test.

To begin, the user should launch the Barcode Test Generator utility. The user should then write a prototype of the expected barcode in the “Sample(ascii)” section. In this example, the prototype barcode is “P&E MICRO-TEST1-0000”. See **Figure 10-3**. Please note that this example only

demonstrates one simple approach to creating a barcode; there are many methods available.

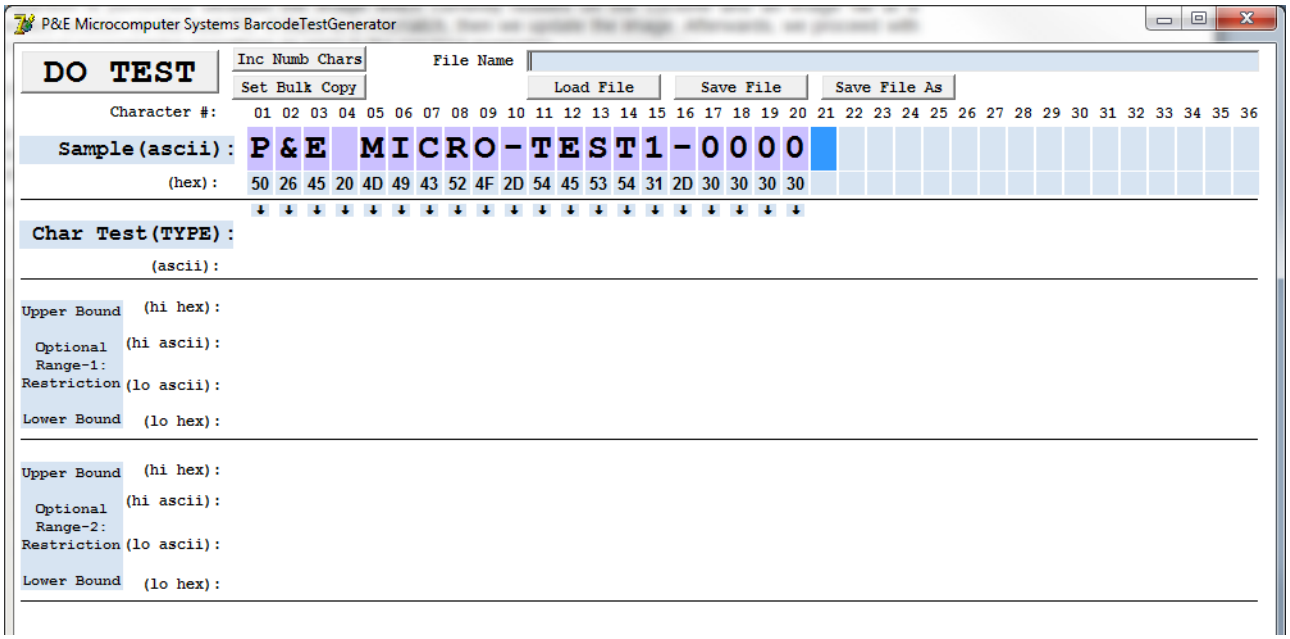


Figure 10-3: Create Prototype Barcode

In this example, the “P&E MICRO” portion of the prototype barcode exists in every bar code the company produces - in this case, header characters that identify the company. Also for this example, a product identifier “TEST1” is in the bar code. Following the identifier, this product also has a 4-digit serial number. This completes the prototype bar code which is representative of this products barcodes.

The next step is to use this prototype barcode to create barcode tests which will be run against scanned barcodes. In this example, the first test should check the header (i.e. "P&E MICRO") at the beginning of the bar code. To enable this, the user should click the small down arrows corresponding to the header section of the barcode. This creates a character test for “P&E MICRO". This string will need to match exactly for the barcode test to pass. The "FC" designation on each character stands for "Fixed Character" with the actual value shown below it. See **Figure 10-4**.

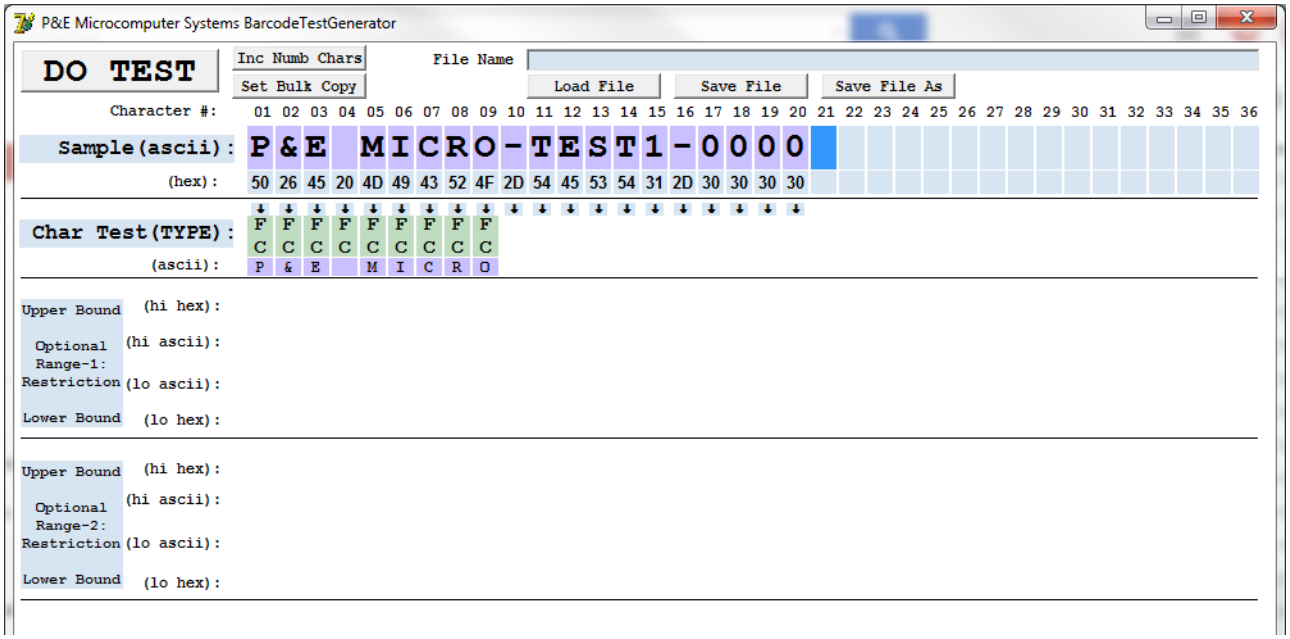


Figure 10-4: Create Char Test For "P&E MICRO"

The next test will check whether the scanned barcode matches the product identifier “TEST1” exactly. The user should click the small down arrows for the "TEST1" characters to bring them into the Char test section. See Figure 5.

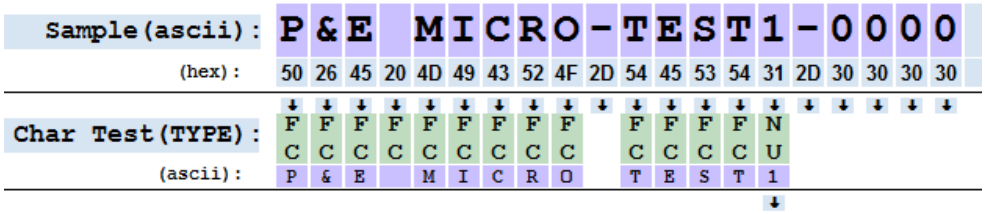


Figure 10-5: Create Char Test For "TEST1"

The "1" in “TEST1” is by default treated as a "Numerical Character" for testing reasons ('NC'). This means it is required to have a value of '0'..'9'. To restrict the value of this character, it could either be change to a fixed character test or we can restrict the range of acceptable values. To do the latter, click on the down arrow below the character test and bring the "1" character into the Range-1 test area. Both the lower and upper bounds for this character can then be changed to “1” so that the match must be exactly the number 1. Refer to Figure 6.

The final step is to create a test for the 4-digit serial number part of the prototype. In this example only 501 units of this product will be manufactured which correspond to a specific programming image, each with a unique serial number from 0000 to 0500. The test should make sure that this limit is not exceeded. Therefore, the user should add “0000” to the Char Test by clicking on the small down arrows for the characters. This should then be converted this into a Range test by clicking on the new arrows to add these four characters to the Range-1 test area.

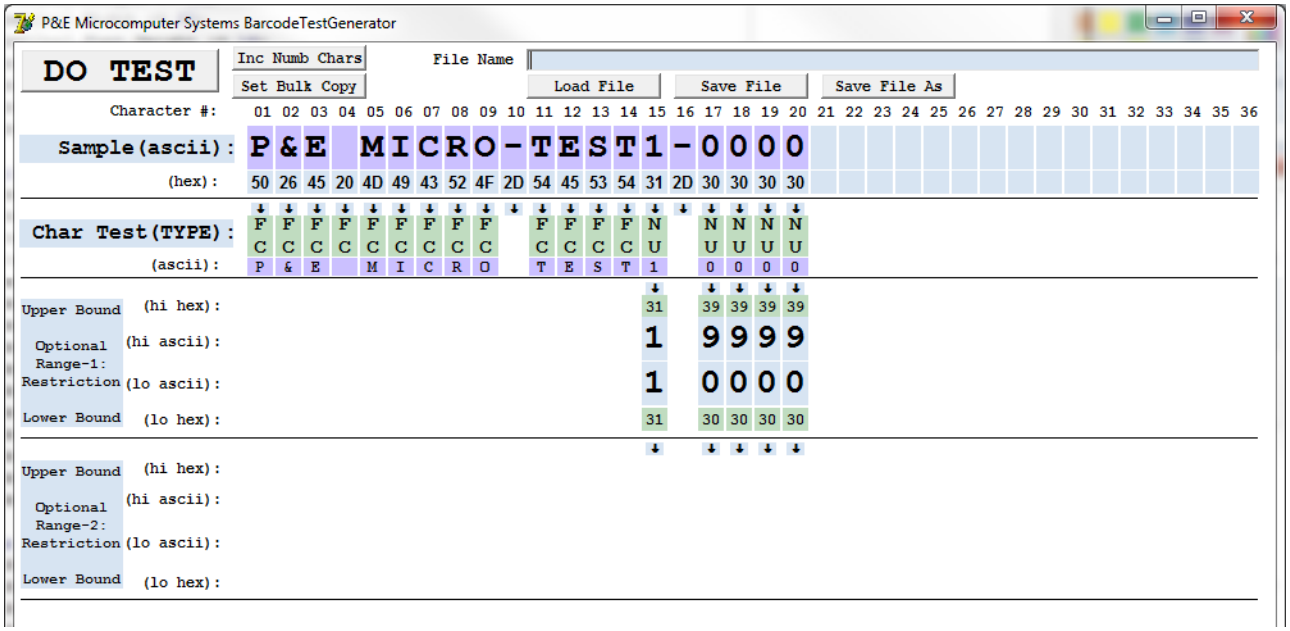


Figure 10-6: Create Range-1 Test

The default for the range test is 0000 to 9999, so after these numerals enter the Range-1 test area, the user should change the upper bound from 9999 to 0500. Any continuous numerical digits are considered to be one numerical value for the purpose of range comparison.

The result in the Barcode Test Generator utility should look like Figure 7:

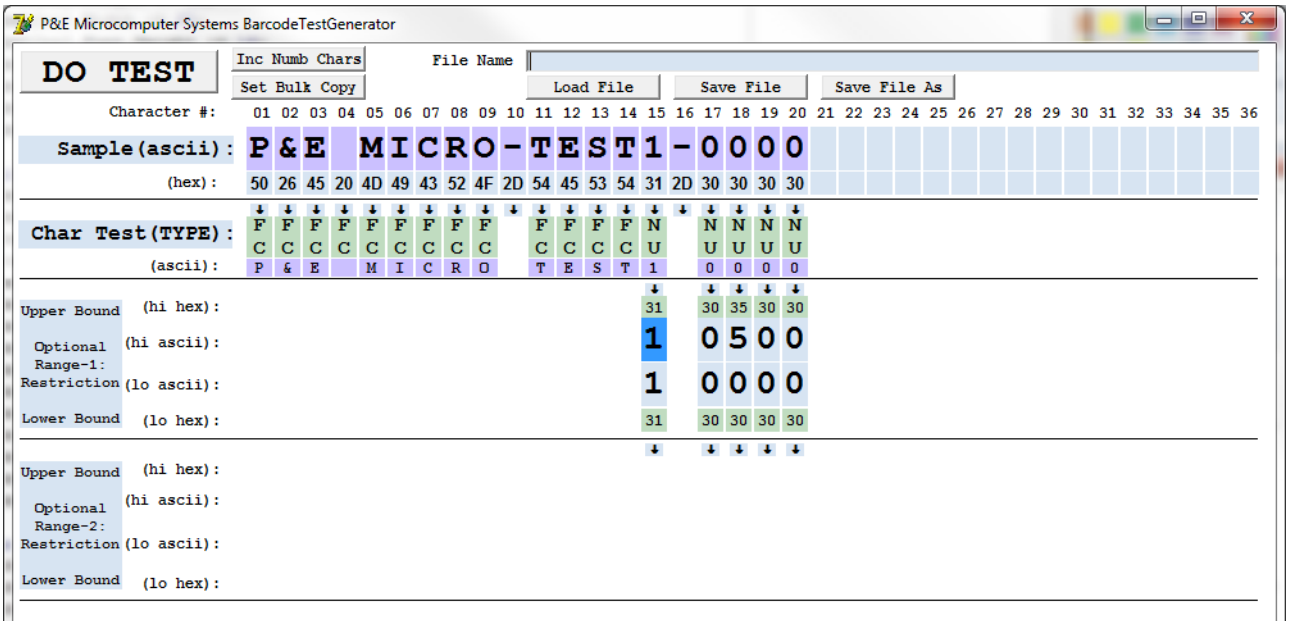


Figure 10-7: Completed Barcode Test

This completes setup of the barcode test. The user should now click on “Save File As” on top and name the test “Test1_barcodetest.bar”. This test is now ready to be added to a programming image using the Cyclone Image Creation utility.

10.6 Adding A Barcode Test Into A Programming Image

Adding the barcode test into an image is done as part of the image creation process. The user should open the Cyclone Image Creation utility and follow the steps to create a programming image. The device should be selected, along with the algorithm, object file, and programming commands. This examples uses the commands EM, PM, VM.

The next step is to add the barcode file. In the “FX Special Features” section at the bottom of the utility, the user should check the “Use Barcode File” checkbox and browse for the appropriate created .bar file. See Figure 8.

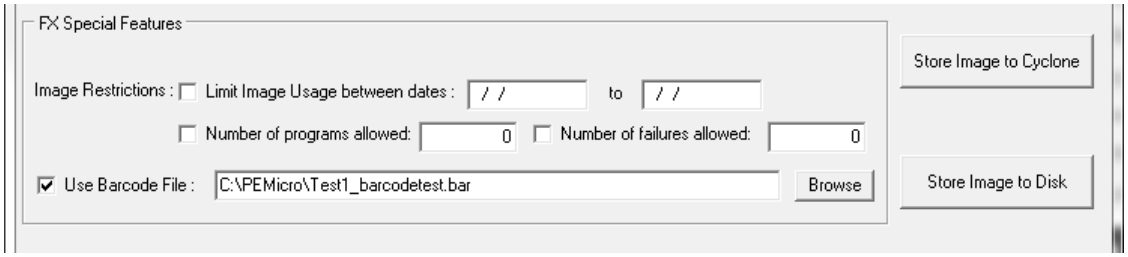


Figure 10-8: FX Special Features (Add Barcode File To Image)

After the programming image is stored in the **CYCLONE FX**, it is ready to use. When a barcode is scanned the Cyclone will encounter this programming image, which will prompt it to test for the exact string “P&E MICRO-TEST1” and for the decimal numbers 0 to 500 on the last four characters of the barcode. If the barcode passes this test (and no other programming images on the Cyclone also pass) the image is executed.

10.7 Troubleshooting

An error will sometimes be generated if more than one image corresponds to the barcode,or if no images correspond to the barcode. The Cyclone includes a way to quickly gain insight into the issue. A log file is created every time the barcode scanner operates and it details the scanned barcode as well as the analysis process used to select the appropriate programming image.

To access the barcode log, navigate in the **CYCLONE FX**'s menu to: Menu->Status->Show Logs->Show Barcode Scanner Log. This Log document contains the details of the last barcode scanner transaction and is overwritten every time the barcode scanner is used.

A sample log file looks like this:

---Scanner Test Log Started---

Scanned barcode: P&E SN: BLUE0138230
5 total number of images in Cyclone

Processing image number 1 (Red Image v1.00)
Image number 1 does not contain a barcode test

Processing image number 2 (Red Image v1.01)
Image number 2 does not contain a barcode test

Processing image number 3 (Widget Rev C)
Image number 3 does not contain a barcode test

Processing image number 4 (Widget Rev D)
Image number 4 does not contain a barcode test

Processing image number 5 (Green Product v7)
Image number 5 passes character test
Image number 5 does not contain a range test
image number 5 passes Barcode Test

One Image passes barcode test
Image number 5 selected to execute

Success, image selected and program command sent

---Scanner Test Log Finished---

11 BARCODE TEST GENERATION

The ability to select a programming image and initiate programming via a barcode scanner requires that a user generate a Barcode Test that will be included in the programming script of the SAP image. In order to create the Barcode test, PEmicro supplies a Barcode Test Generator utility that facilitates creating a Barcode Test which describes the mechanism for calculating whether an input barcode meets the criterion the user is looking for to be considered a match, which can be an exact match or a match that is in a **Range** selected by the user.

The **Char Test(TYPE):** and **Range** test fields are first chosen from selected **Sample** (ASCII) characters. It is intended for the user to enter a whole sample barcode and modify the type and to create appropriate tests, **Char Test(TYPE):** limits the individual characters to types, such as numeric, alphabetic, hexadecimal, etc. Range tests determine if the magnitude of chosen set of Sample input characters viewed as a large number are within a lower and upper bounds. These **Range** tests are used to validate sequence numbers to be within specified bounds. For a test to be valid it must pass any **Char Test(TYPE):** tests and at least one if any **Range** tests. The **Range** tests are actually done on a set of Sub-Ranges defined as being contiguous set of characters and having all the same **Char Test(TYPE):**. **Figure 11-1** shows a typical main screen for the Barcode Test Generator utility.

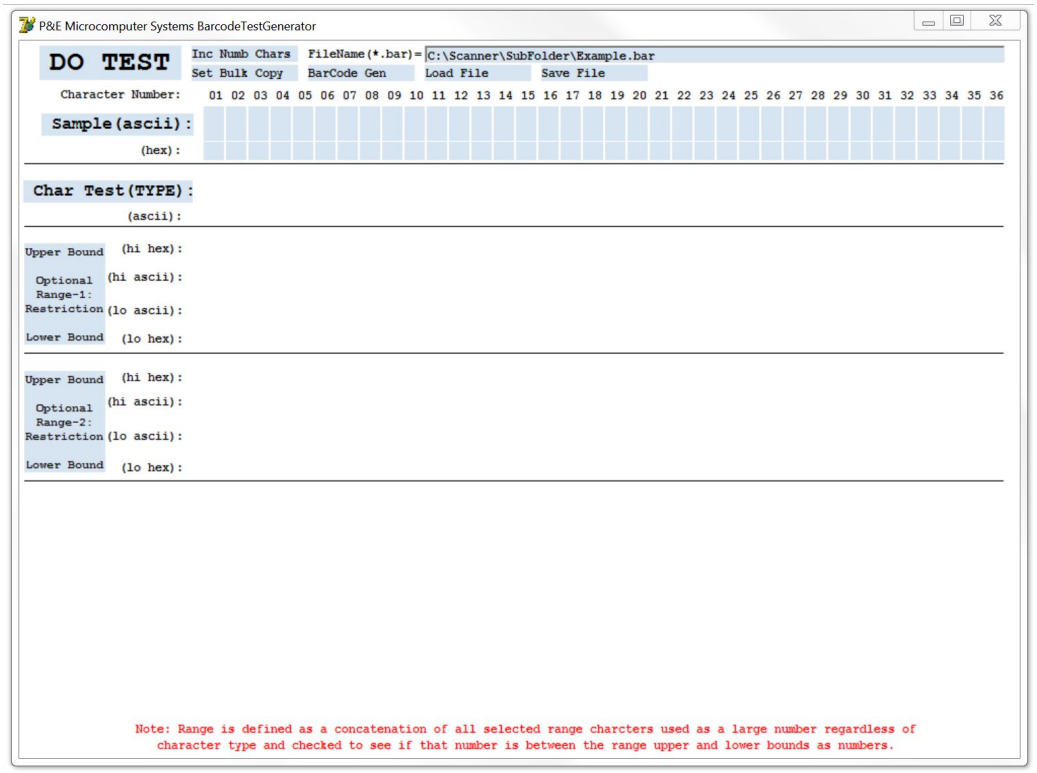


Figure 11-1: Typical Barcode Test Generator Main Screen On Startup

11.1 Barcode Test Creation and Testing Process

A Barcode Test can be easily created and tested using the following 8 simple steps which are explained in paragraphs following the list below:

1. Create a **Sample(ascii)/(hex)** Barcode
2. Select Characters to be tested by copying them to **Char Test(TYPE):**.
3. Set desired type of selected characters from step 2.
4. Copy character types from step 3 to **Range-1** or **Range-2** if Range testing.

5. Adjust range character(s) minimums and maximums of character of Step 4.
6. Press the **DO TEST** button to test the Sample against test setup.
7. Make changes to **Sample**, **Char Test(TYPE)**, **Range-1**, or **Range-2** and repeat steps 6 and 7 until an acceptable test has been created.
8. Save the test to a Barcode Test (.bar) file for later use.

The **Sample** character set is used first as a structural frame for creating a Barcode test and then later as an input field for entering multiple barcode samples to verify the Barcode test works as expected.

11.1.1 Enter a Sample(ascii)/(hex)

Sample characters can be entered either as ascii characters or hexadecimal bytes (2 characters per byte) and in many ways. To start the process, click on either a **Sample(ascii)** box or a **Sample(hex)** box. This makes the box **darker blue** to indicate where the entry cursor is. Once the entry cursor is set characters can be entered into the **Sample** by several methods listed below. When entries are placed in an **Sample(ascii)** box they are also updated in the corresponding **Sample(hex)** box and vice versa:

1. By simply typing in ascii characters or pairs of hexadecimal characters.
2. By copying a string of highlighted characters by Ctrl-C / Ctrl-V method.
3. By scanning in a string of characters starting at the current cursor position.

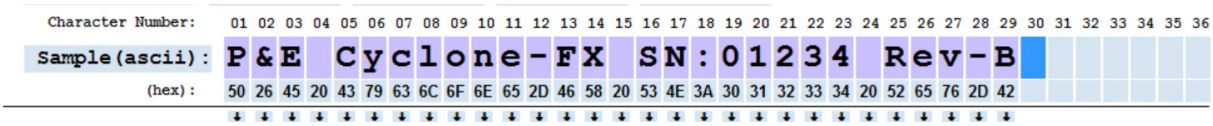


Figure 11-2: Typical Sample Entry Screen Section

Figure 11-1 shows a typical entry screen section. The first line gives the character numbers. The second line shows entered characters and **un-entered characters in ASCII**. The third line shows the **entered characters in hexadecimal**, and the current **cursor** position. The last line shows a set of down arrows which when clicked on selects the characters to be tested.

Sample characters are deleted by double clicking on a **Sample(ascii)** or **Sample(hex)** box . For convenience, the entire set of **Sample** characters can be deleted by double clicking on the left screen **Sample(ascii):** button. As characters are entered, the cursor automatically moves to the right stopping at the last Sample entry box. To add additional character boxes, see section 6.2.

When a characters are entered into a **Sample(asci)** box, the background color of the box is changed to Light Purple. This is done to highlight the difference between a space character and un-entered boxes. At the same time, a down arrow is placed under the **Sample(hex)** box to indicate that the Sample character can be transferred to a **Char Test(Type):** field by clicking on the down arrow.

11.1.1.1 Sample entry by typing in Characters

If the cursor is in a **Sample(ascii)** box typing in characters will place them in that box. The cursor then automatically moved to the next **Sample(ascii)** box. All ASCII characters including control codes can be entered in **Sample(ascii)** boxes. However, if there are characters stored on the desktop, a Ctrl-V will enter them rather than entering a Ctrl-V character. After two **Sample(hex)** box character entries, the cursor automatically proceeds to the next **Sample(hex)** box. Only the hexadecimal characters, (0-9, A-F, a-f), are allowed in a **Sample(hex)** boxes. Hex values are entered high nibble and then the low nibble.

11.1.1.2 Sample entry by copying in a string of highlighted characters

A string of characters can be entered into a series of **Sample(ascii)** boxes starting at the cursor by first highlighting them by a mouse down pass over of the characters, then doing a Ctrl-C and a Ctrl-V. This is a standard Windows System process of placing a string on the desktop and then inserting it somewhere else. However, you cannot do this with characters within the utilities window.

11.1.1.3 Sample entry by scanning in a string of characters

Most barcode scanners can place the scanned characters into the Windows keyboard buffer. Hence, they are automatically entered into a sequence of **Sample(ascii)** boxes starting at the cursor. Make sure when setting up a scanner that it is configured to place the scanned characters in the keyboard buffer.

11.1.1.4 Deleting an entry

The contents of an entry box may be deleted by double-clicking on the entry box.

11.1.2 Copy Characters to be tested to Char Test(TYPE):

When a barcode is scanned during production programming, it is often the case that all the characters do not need to be tested to determine the appropriate standalone programming (SAP) image to be used. Hence, it is necessary to select and copy only those **Sample** characters to be tested.

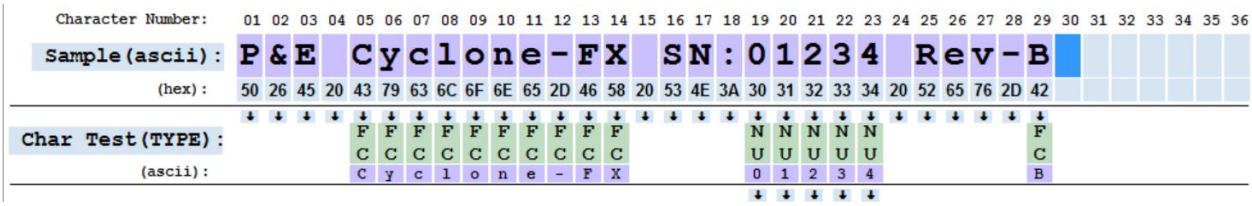


Figure 11-3: Sample Characters copied to Char Test Section

Simply clicking on the down arrows between the **Sample** and **Char Test(TYPE):** Sections of the screen transfers the corresponding character to the test section. The transferred characters are shown as ASCII characters along with an **associated type**. The ASCII characters in the Char Test Section represent transferred value and do not change when the Sample characters are edited. When transferred from the Sample, the associate type is automatically set to either Fixed Character (**FC**) type or if asci characters (0-9) to NUmeric (**NU**) type. All possible character types are defined in **Section 11.2 - Character Types**. Any character types which are represented by a contiguous range of values are indicated by a set on down arrows into the Optional Range Restriction Sections of the display.

Figure 11-3 shows that only the product name “Cyclone-FX”, the serial number (SN) “01234”, and the revision (Rev-) “B” were selected for character testing. The manufacturer “P&E”, blank spaces, and other characters are ignored during character testing.

Note: Each of the fixed characters in ‘Cyclone-FX’ and ‘B’ require an exact match, while each number in ‘01234’ only has to be in the **NU** range of 0-9 to be a match.

All entries in the Char Test field can be deleted by double clicking the **Char Test(TYPE):** button. Individual Char Test entries can be deleted by double clicking anywhere on them.

Often setting just the **Char Test(TYPE):** of tested characters to Numeric (**NU**) or Fixed Character (**FC**) and setting the appropriate ascii value for Fixed Character (**FC**) fields is sufficient to generate a barcode test (although the Barcode Test Generation Utility also allows specification of fairly complex barcodes as well).

11.1.3 Set desired type of selected characters from step 2.2

Once a Char Test characters have been selected, they can be changed to different types than were transferred from the Sample. An individual associated type is edited by clicking on it to set the cursor position. For example, clicking on the Char Test(TYPE): "B" character type opens the window in Figure 2.3a:

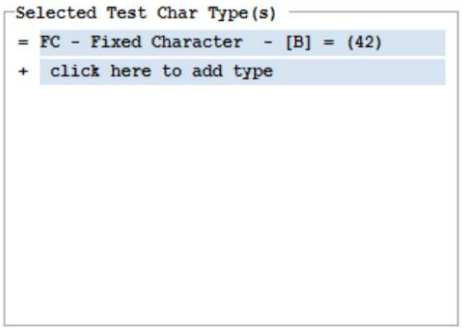


Figure 11-4: Selected Test Char Type(s) Window

11.1.3.1 Adding, Changing and Deleting Char Test(TYPE):

This window shows all the character types associated with the selected `Char Test` character. Shown types can be changed or deleted by clicking on them or types can be added by or deleted clicking on "click here to add type." If clicked, one of the similar windows shown in **Figure 11-5.a**, **Figure 11-5.b,** or **Figure 11-5.c** is opened. Types can also be efficiently copied to other blocks as per **Section 11.6.2 - Ctrl-C / Ctrl-V Char Test(TYPE): and Range Copying**.

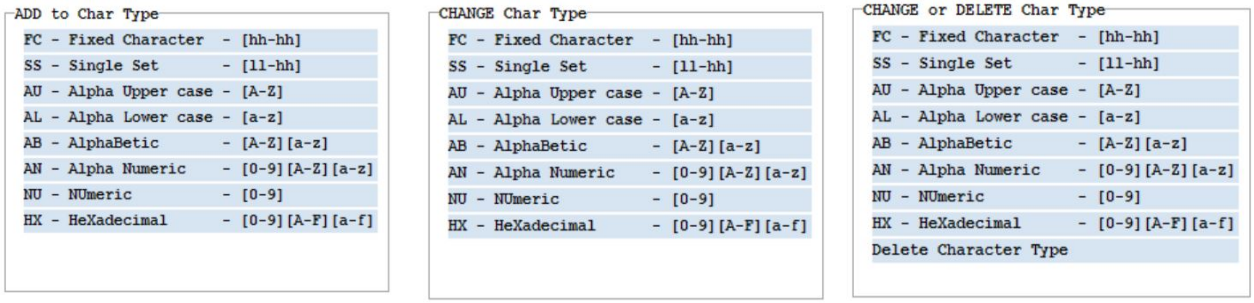


Figure 11-5: a) On Click Add Figure; b) On Click Type Figure; c) On Click Type (multiple types)

Clicking on one of the opened window selections causes that option to be added, changed or deleted.

11.1.3.2 Fixed Character (`FC`) and Single Set (`SS`) Sub-Types

Choosing change either `FC` or `SS` type will open respectively the following windows to allow entry of the numeric values of these types.

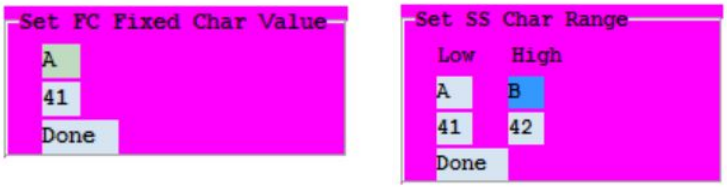


Figure 11-6: a) FC Value; b) SS Values

The `FC` type choice, **Figure 11-6.a**, only requires one value to be entered since a Fixed Character (`FC`) has the same upper and lower values. The Single Set (`SS`) type choice, **Figure 11-6.b**, represents a single contiguous set of any length and hence requires both lower an upper bound

values. These values can be entered either as ASCII characters or pairs of hexadecimal digits. The fuchsia color indicates that some action must be taken before proceeding.

11.1.3.3 Multiple Sub-Types

A given Char Test character can have up to a maximum of 10 sub-types. However, if one selects either numerically adjoining or overlapping sub-type sets, the program automatically compresses the sub-types.

For this example, the “Rev-“ “B” `Char Test(TYPE):` was set to allow the characters “A”, “B”, “D”, or “E”. This is shown in **Figure 11-7.a** as two `SS` sub-types. This could have been done by adding three `FC` types for “A”, “D”, and “E” or two `SS` types [“A”-“B”] and [“D”-“E”]. However, the program would compress the `FC` types to `SS` types as shown. As can be seen in **Figure 11-7.b**, the “Rev-“ `Char Test(TYPE):` information was changed from an `FC` type to a MulTiple (`MT`) type.

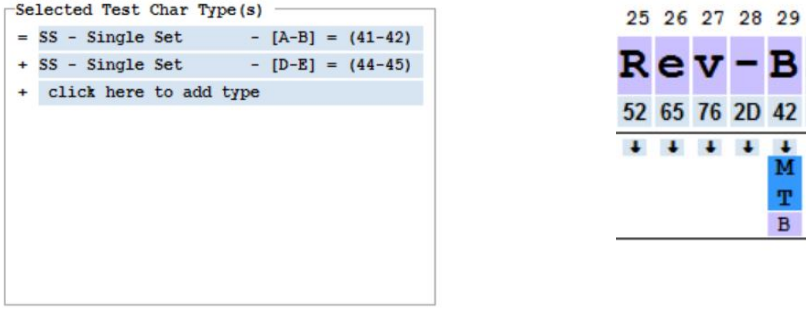


Figure 11-7: a) Compressed Types for “Rev-“ Test Character; b) MT Type Change for “Rev-“ Test Character

11.1.3.4 Type Changing Errors

When initially created, a Char Test type always matches the type of character in the corresponding Sample field. However, changing a Char Test type or a Sample character can case an error. Errors are always indicated by changing appropriate background colors to `red`. For example, if character number 05 type is changed to NUmeric (`NU`) the following error would appear on the screen:

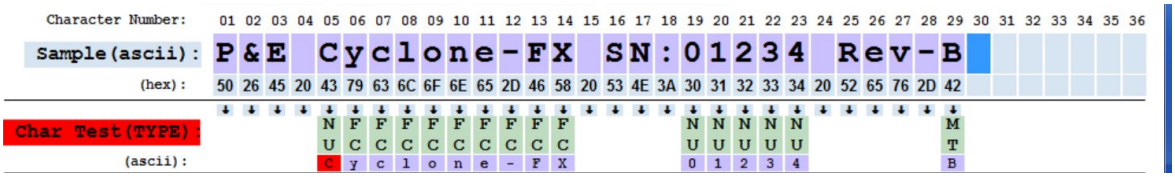


Figure 11-8: Error from invalid type for Sample character 05

Notice that the `Char Test(TYPE):` Button is red to indicate a problem in that field and that the specific type character “C” causing the error is also red. Note that it is not the type character which causes the error but the Sample character which is tested against its type. If multiple type boxes were in error, each of the type boxes would be marked in red. These error conditions can be corrected either by changing the `Char Test(TYPE):` entry or changing the corresponding `Sample` character. Clicking on almost anything con the screen clears all error indications.

11.1.4 Copy character types from step 3 to `Range-1` or `Range-2`.

When Char Test characters were added, down arrows were added only for contiguous types. These down arrows can be used to select which characters from `Char Test(TYPE):` are added to the two possible Optional Range Restrictions, `Range-1` and `Range-2`. In general, range tests should only be added when the Char Type is not sufficient to test the appropriate values.

The characters add to a range are numerically concatenated together to form a single numerical value which will be compared against specified upper and lower bounds tests. This allows for testing to see if the target should be programmed based upon such things as serial numbers. When not all allowable sequences for the types defined are desired. For instance, if a sequence

number from 00000-49999 was desired, but not 50000-99999, a **Range** restriction can be used. If, however, 00000-99999 was fine for the serial number test, a range restriction is not needed and the five characters are simply defined in the **Char Test(TYPE):** as Numerical(**NU**). In most cases, valid range restrictions will probably be restrictions based upon numeric digits. However, any characters: numeric, hexadecimal, alphabetic, control characters, etc. are allowed to form a range.

Down arrows are not provide for a Fixed Character (**FC**) type since it is usually sufficient for them to be tested as **Char Test(TYPE):** characters. Down arrows are only shown for contiguous types since non-contiguous types would cause holes in sequence numbers which can be confusing. To bypass these restrictions, see **Section 11.6.1 - Bulk Copy Option**.

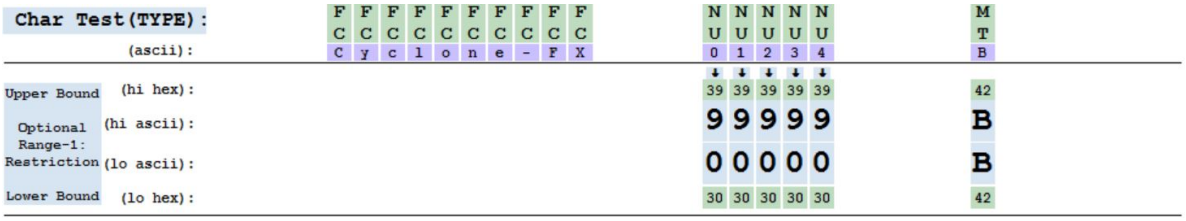


Figure 11-9: Transfers from Char Test to Range-1

In **Figure 11-9**, the NUmeric (**NU**) characters were transferred to the **Range-1** test using the down Arrows. Only types with contiguous character sets can be transferred this way. When contiguous types are transferred, the upper and lower bounds are set to the types maximum and minimum values for convenience. However, it is easy to change them later. The **MT** type was transferred using the bulk copy method of section 6.1 since it is non-contiguous. Because of this, care must be taken about missing value in the Range-1 specified. When non-contiguous sets are transferred, the upper and lower bounds are both set the **Char Test(ascii)** character value. Such values can be easily changed later.

11.1.5 Adjust Range Character Upper and Lower Bounds

To change upper and lower bound values, first click on either a bound (ascii) box or a bound (hex) box to set the cursor there to that box. Typing an asci character or a pair of hexadecimal characters enters them into the appropriate box. Then the cursor is moved to the right one block, circularly over the set of range blocks. Pairs of upper and lower bounds can also be copied to other pairs as describer in Section 6.2. All entries in a range can be deleted by double clicking the Range-1 or the Range-2 buttons. Individual range entries can be deleted by double clicking anywhere on them.

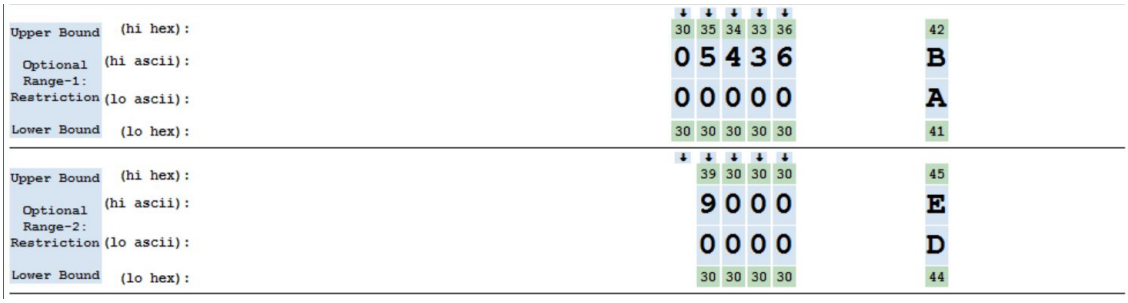


Figure 11-10: Adjusted Range-1 and Range-2 Bounds

Notice in **Figure 11-10**, that **Range-1** numeric values run between 00000 and 05436. In addition, for **Range-1**, the lone **MT** type was adjusted to use only a contiguous range of “A” to “B”. **Range-1** was adjusted to have the numeric values between 0000 and 9000. Also, its lone **MT** type was set to a contiguous range of “D”-“E”. These choices created only contiguous ranges, i.e. no holes in a range. Ranges which are non-contiguous are allowed if necessary but generate confusing results.

When creating range entries, the Upper Bound should be greater than or equal to the lower bound. If the bounds are not entered that way, an error is generated as shown below in **Figure 11-11** by

changing the background color of the appropriate **Range button(s)** and the **bound(s) entry** to fuchsia. This can be fixed by swapping the upper and lower bounds.\

Figure 11-11: Bounds Entry Error

11.1.6 Press the DO TEST button to test Sample(ascii) characters against all tests.

When developing Barcode tests, it is necessary to frequently test to assure that the tests being generated corrects select the appropriate standalone programming (SAP) image. Clicking on the **DO TEST** button, as shown in the **Figure 11-12** control area of this program, causes the **CharTest(TYPE):** and both Range tests to be run if they exist.

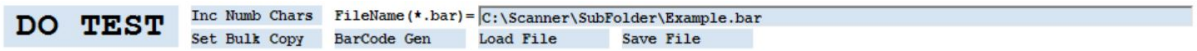


Figure 11-12: Program Control Area

To pass the test, the Char Test and at least one of the Range tests must pass. If the test passed, the, the background color of **DO TEST** is changed to green. It is possible for **DO TEST** to be green even if one of the **Range** tests is red. Shown in Figure 2.6b is the result of changing the numeric characters to “05444” and clicking on **DO TEST** Button.

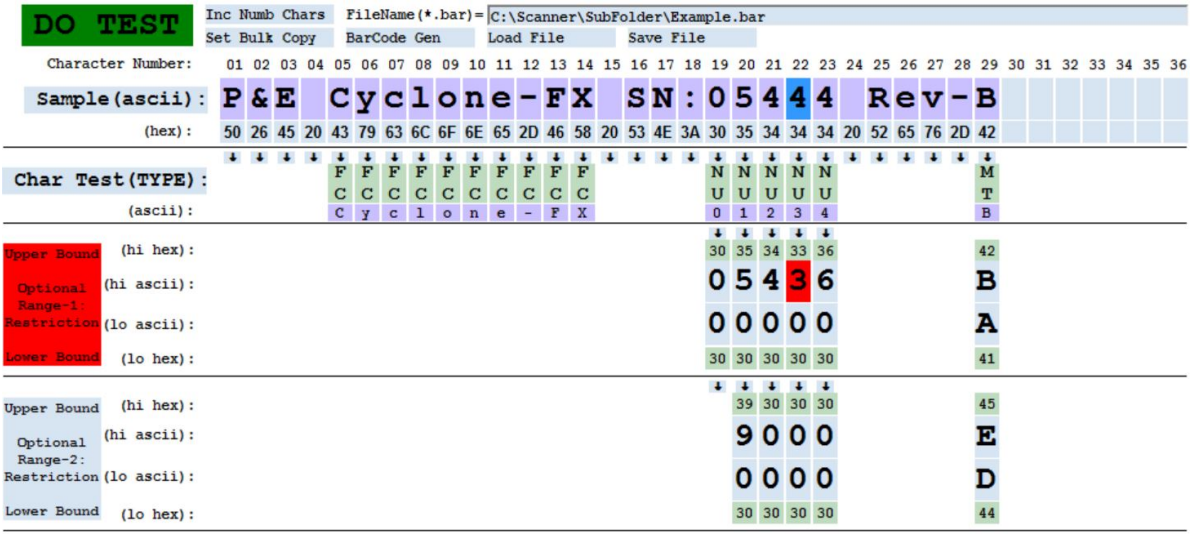


Figure 11-13: Passed Test with only One Range Test Passed

Note that the **DO TEST** button is green indicating that the test passed. Also, note that the **Range-1** button is red which means that that the **Range-1** test failed but didn’t cause the overall test to fail. Additionally, **Range-1** character 22, “3”, failed the upper bound test. See Section 4. On how Range Tests are conducted.

In **Figure 11-13**, the **Range-2** test passed since when the first character tested (character 20, “5”) was greater than the lower bound (“0”) and less than the upper bound (“9”).

Shown in **Figure 11-14** is a test of a slightly different sample which failed on many things.

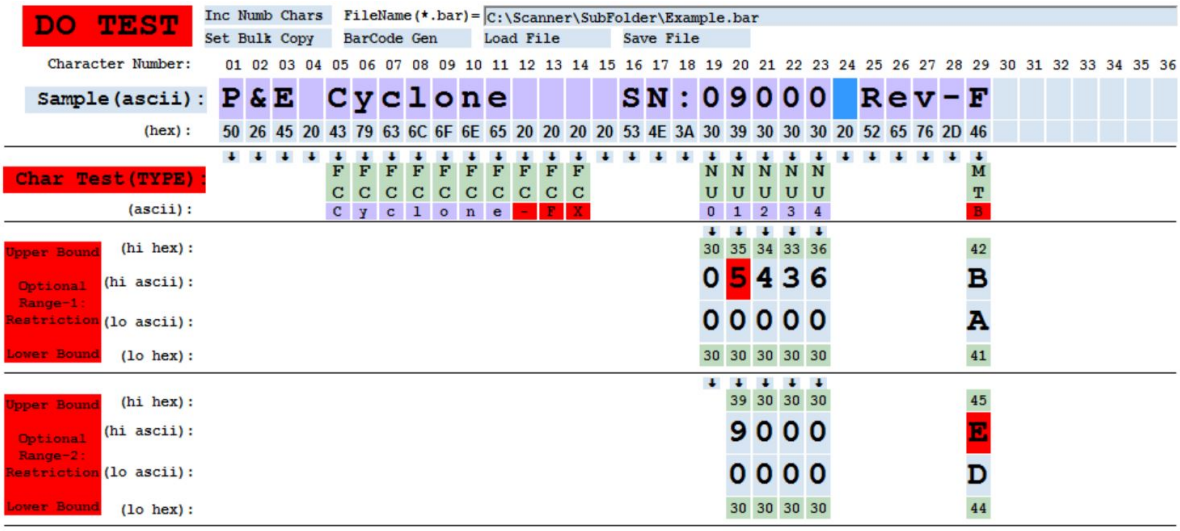


Figure 11-14: Failed Test with Multiple Failures

Clicking on almost anything on the screen clears all error indications.

11.1.7 Make changes to Sample(ascii), Char Test(TYPE):, Range-1 , or Range-2 & Repeat Test

This step represents the start of the iterative process used to generate an appropriate test for the product to be programmed. Try different **Samples**, run **DO TEST**, make changes to the test, etc., until a satisfactory test is found.

11.1.8 Save the test to a Barcode Test (.bar) file for later use.

The tests created can be saved (See **Section 11.5 - Saving and Loading Barcode Test (.bar) Files**.) for later inclusion in a Cyclone FX programming SAP image using PEmicro's image creation utility [1]. The saved tests can also be reloaded for correction or as the base for creating other tests.

11.2 Character Types

Selected characters are tested as part of the **Char Test (TYPE):** to see if they are included in a set of character called a type. Listed in **Table** below are the type sets allowed in this program. Most of the types have common names and abbreviated mnemonics. Some types are contiguous and others are not. Contiguous means that there is only a single lower and a single upper count for the set with no missing values. A special type called MT (Multiple Types) means that is composed of a number of sub-types.

When forming a MulTiple (**MT**) types, the program will automatically compress either adjacent of overlapping sub-types if possible as they are created. This optimization makes for a more efficient type testing process. In addition, when creating Fixed Characters (**FC**) or Single Set (**SS**) types, the program asks for the lower or upper bounds for these sets either as ASCII characters or pairs of hexadecimal digits. **FC** is the same as an **SS** with a bounds range of a single character, i.e., the lower and upper bounds have the same value.

Table K-1. Allowed Character Types

Type	Type Name	Ascii Range	Hex Range	Contiguous
F C	**F**ixed **C**haracter	selected	[hh-hh]	Y
S S	**S**ingle **S**et	selected	[ll-hh]	Y

Table K-1. Allowed Character Types

Type	Type Name	Ascii Range	Hex Range	Contiguous
A U	**A**lphabetic **U**pper case	[A-Z]	[41-5A]	Y
A L	**A**lphabetic **L**ower case	[a-z]	[61-7A]	Y
A B	**A**lpha**B**etic	[A-Z][a-z]	[41-5A] [61-7A]	N
A N	**A**lpha **N**umeric	[0-9][A-Z][a-z]	[30-39] [41-5A] [61-7A]	N
N U	**NU**meric	[0-9]	[30-39]	Y
H X	**H**e**X**adecimal	[0-9][A-F][a-f]	[30-39] [41-46] [61-66]	N
M T	**M**ul**T**iple	Multiples of above	Multiples of above	N

11.2.1 Background Color Codes

Figure 11-15 shows what the color codes typically represent on the program screen:

, , or Clicked to Select or Double-Clicked to delete
Indicates Selected ASCII character values
Indicates Selected hex character values
Indicates a Failure
Indicates a Success
Needs attention to continue or indicates inverted bounds
Marked by Ctrl-C for Ctrl-V copying
Indicates the Cursor position

Figure 11-15: Background Color Codes

11.2.2 Font Color Codes

Figure 11-16 shows what the font colors represent on the program screen:

XXX Normal character font color

XXX Font color selected for Bulk Copying, or to indicate an Option Open Option

Figure 11-16: Font Color Codes

11.3 Enabling/Disabling A Barcode Scanner

To enable a barcode scanner that is connected to the Cyclone FX's USB Extension Port, go to:

Menu -> Configure Cyclone -> Configure USB Host Device -> Enable USB Scanner

This setting is persistent; the scanner will be powered and turned on when the Cyclone is powered, until the scanner is disabled. To disable the barcode scanner, select:

Menu -> Configure Cyclone -> Configure USB Host Device -> Disable

11.4 Ranges and Range Testing

Testing is done in two steps First, if any **`Char Test(TYPE):`** characters are specified a Character Type test is done as indicated in **Section 11.4.1 - Character Type Testing**. Then, if either **`Range-1`** or **`Range-2`** has entries, a Range test is done as indicate in **Section 11.4.2 - Range Testing**.

To pass testing, the **`Char Type(TYPE):`** test must pass if it exists. In addition, at least one Range Test must pass if any exist, If not, errors are indicated on the programs main screen.

11.4.1 Character Type Testing

Each character in **`Sample`** which has a corresponding **`Char Test(TYPE):`** is tested numerically to see if it is within one of the sets for the type specified. Types can have multiple sets of characters. As long as the **`Sample`** character is in one of the sets, the test is valid for that character. If any individual **`Sample`** character test fails, the overall test is failed.

11.4.2 Range Testing

`Ranges` are subdivided into Sub-Ranges which are delineated by contiguous sets of identical **`Char Test(TYPE):`**s. A **`Range`** test consists of testing each Sub-Range. A **`Range`** test passes if all of its' Sub-Range tests pass. Sub-Range testing checks to see if the selected set of Sub-Range **`Sample`** characters treated as a large number fits between a lower and an upper bound. The bounds are formed by numerically concatenating the selected Sub-Range bounds boxes, to form a number representing lower and upper bounds for that Sub-Range.

Sub-Range bounds entry boxes are actually tested against their corresponding **`Sample`** characters on a character by character basis from left to right until all the individual lower and upper bounds for a given **`Range`** are tested:

On lower bounds, starting at the left,	if the corresponding Sample character is:
(a) lower than its lower bound,	lower bound Sub-Range test failed, do next Sub-Range
(b) equal to its lower bound,	next Sub-Range character to the right must be tested
(c) else,	lower bound Sub-Range test passed, do next Sub-Range.

On upper bounds, starting at the left,	if the corresponding Sample character is:
(a) greater than its upper bound,	upper bound Sub-Range test failed, do next Sub-Range
(b) equal to its upper bound,	next Sub-Range character to the right must be tested
(c) else,	upper bound Sub-Range test passed, do next Sub-Range.

This process allows the exact character in the range sequence that caused a range test error if any. This also means that the range is tested as a large concatenated number composed of numerical characters. The test is performed on a character by character basis.

11.5 Saving and Loading Barcode Test (.bar) Files

11.5.1 5.1 Barcode Test (.bar) File Structure

PEmicro Barcode Test Data is stored in JSON Data Interchange Format files [2]. The information in an empty PEmicro JSON Barcode Test file is shown below where blue characters represent the formatting information and red characters represent what data in inserted in the file by the Barcode Test Generator. All data is in hexadecimal to make the files readable since the data represented can be nonprintable asci characters. This empty PEmicro JSON Barcode Test file has been reformatted, spacing wise, to make it more readable. There are many sites on the internet which will reformat JSON files [3].

```
{
  "pesettings": {
    "scannertest": {
      "testchars": " ... hex pairs of character test information ... ",
      "range1": " ... hex pairs of range1 lower and upper bounds ... ",
      "range2": " ... hex pairs of range2 lower and upper bounds ... "
    },
    "scannersave": {
      "samplechars": " ... hex pairs of sample data – not used by Cyclone FX ... ",
      "testcharstrings": "... hex pairs of test data – not used by Cyclone FX ..."
    }
  }
}
```

The unformatted version of an empty PEmicro JSON Barcode Test file looks like:


```
{"pesettings":{"scannertest":{"testchars":"","range1":"","range2":""},"scannersave":{"samplechars":"","testcharstrings":""}}}
```


The unformatted version of the PEmicro JSON Barcode Test file for the examples used in this manual looks like:


```
{"pesettings":{"scannertest":{"testchars":"0501434306017979070163630801 6C6C09016F6F0A016E6E0B0165650C012D2D0D0146460E01585813013039140130391501303916013039170130391D0241424445","range1":"9330301430359530341630339730369D4142","range2":"94303995303096303097303 09D4445"},"scannersave":{"samplechars":"0150022603450420054306790763086C096F0A6E0B650C2D0D460E580F201053114E123A13301431153216331734182019521A651B761C2D1D42","testcharstrings":"0543464303014343067946430301797907634643030163086C464303016C6C096F464303016F6F0A6E464303016E6E0B654643030165650C2D464303012D2D0D4646430301464 60E5846430301585813304E5503073039143 14E5503073039153 24E5503073039163 34E5503073039173 44E55030730391D424D5406024142024445"}}}
```


11.5.2 Saving a Barcode Test (.bar) File

The current information on the program main screen can be saved to a Barcode Test (.bar) file for two reasons. First, so the tests can be used by PEmicro's Cyclone FX, Standalone programmer. Second, so the information can be loaded back into this program for modification, further testing of the tests themselves, or to assist in the creation of new tests.

To save a Barcode Test (.bar) file, click on the **Save File** button in the control area of the main screes.

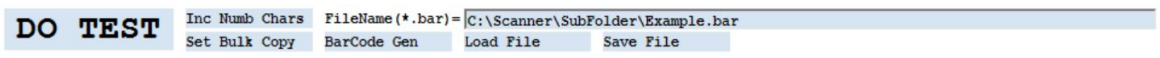


Figure 11-17: Program Control Area

If a filename (editable) exists in the control area, the program tries to save the screen test information to that file. If no filename exists, then the window shown below is opened to search for and/or edit a filename. If Open is clicked, the program attempts to save the test information to that file. On attesting to save a file, the button is recolored respectively on Success or Failure to **Save File** or **Save File**.

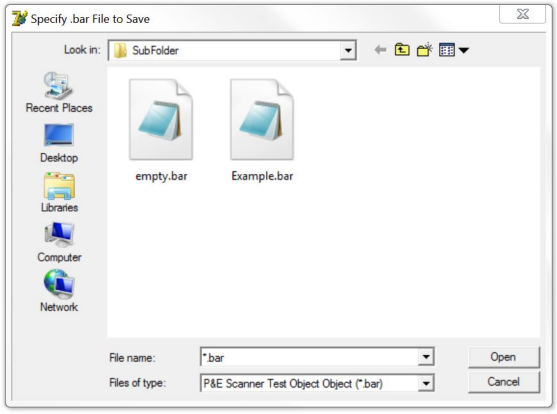


Figure 11-18: Save PEmicro JSON File Window

11.5.3 Loading a Barcode Test (.bar) File

PEmicro JSON Barcode Test (.bar) Information can be loaded into this utility so it can be edited further or to test other barcodes.

To load a Barcode Test (.bar) file, click on the **Load File** button in the control area of the main screen.

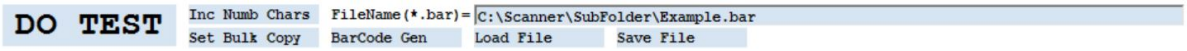


Figure 11-19: Program Control Area

If a filename (editable) exists in the control area, the program tries to load test information from that file. If no filename exists, then the window shown below is opened to search for and/or edit a filename. If Open is clicked, the program attempts to load the test information from that file. On attesting to load a file, the button is recolored respectively on Success or Failure to Load File or LoadFile .

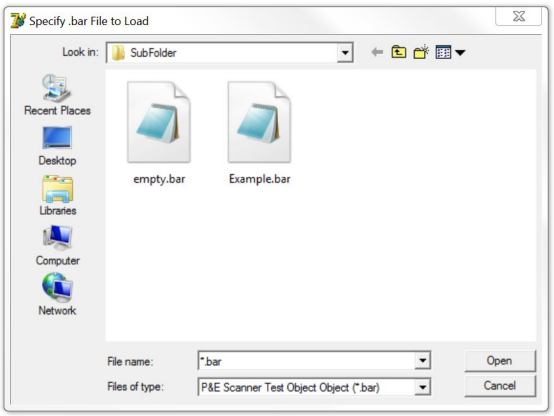


Figure 11-20: Load PEmicro JSON File Window

11.6 Advanced Program Use

11.6.1 Bulk Copy Option

Often, it is required to make many transfers from **Sample** characters to **Char Test(TYPE):** entries or from **Char Test(TYPE):** entries to **Range-1** or **Range-2**. These transfers are usually done by individually clicking on appropriate down arrows. For large barcodes, this can be very time consuming. With Bulk Copy, Sample characters can be selected / deselected by a mouse entry

into a Sample box. Characters which are selected are indicated by changing the font color to red. Once a set of characters has been selected, they are copied into `Char Test(TYPE):` entries by clicking on the `Char Test(TYPE):` button. Bulk Copy turns of the selected character red font highlighting.

In a similar manner, `Char Test(TYPE):` entries can be highlighted and copied into `Range-1` or `Range-2` entries.

This process takes a little practice to get used to but greatly speeds things up. In addition, this process can be used to copy entries which do not have down arrows assigned to them.

11.6.1.1 Turning On Bulk Copy

Clicking on the black text `Set Bulk Copy` button in the program control area enables Bulk Copying. When the Bulk Copy option is selected, the text in the `Set Bulk Copy` button is recolored red as a reminder to turn the option off by clicking the button again.

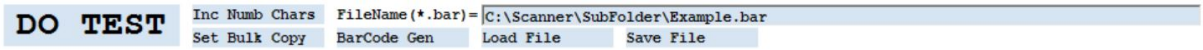


Figure 11-21: Program Control Area

11.6.1.2 Copy Sample Boxes to Char Test

Figure 11-22 shows bulk copy selected `Sample` characters selected as indicated by the red type fonts. After clicking on the `Char Test(TYPE):` button the `Sample` characters are transferred as shown in **Figure 11-23**. Notice that the `Sample` highlighting is turned off after the copy.

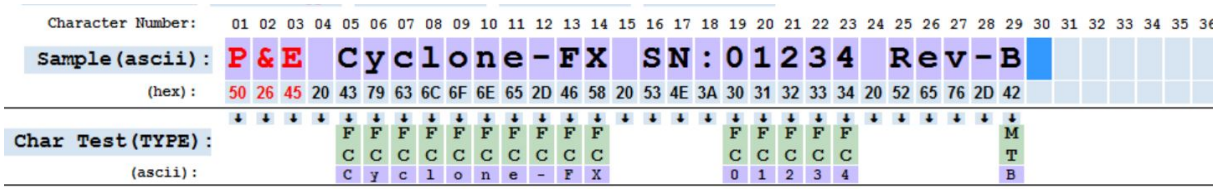


Figure 11-22: Highlighted Sample entries for Bulk Copy to Char Test(TYPE): entries

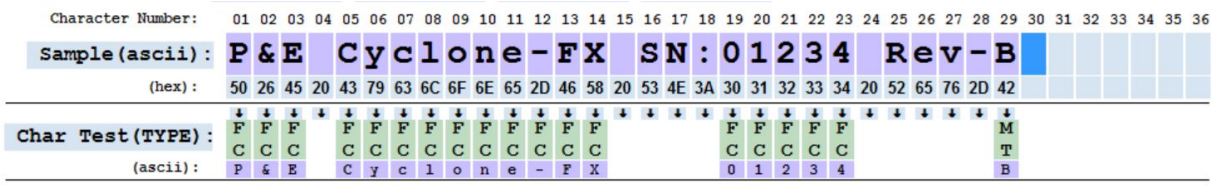


Figure 11-23: After Sample entries Bulk Copy to Char Test(TYPE): entries

11.6.1.3 Copy Char Test Boxes to a Range

Figure 6.1.3a shows bulk copy `Char Test(TYPE):` entries selected as indicated by the red type fonts. After clicking on the `Range-1:` button the `Char Test(TYPE):` entries are transferred as shown in Figure 6.1.3b. Note that `Range-2` entries did not change. However, `Char Test(TYPE):` can be bulk copied to `Range-2` in a similar manner. Notice that the `Char Test(TYPE):` highlighting is turned off after the copy.

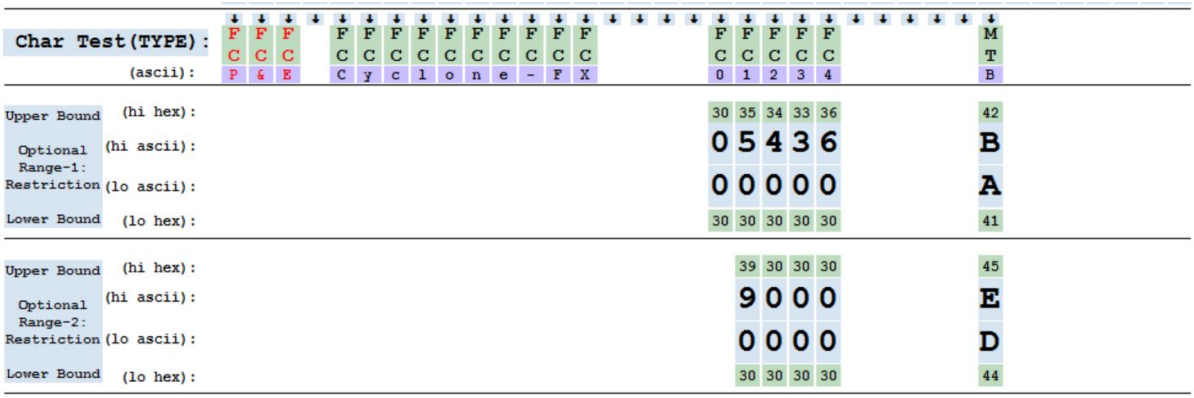


Figure 11-24: Highlighted Char Test(TYPE): entries for Bulk Copy to Range entries

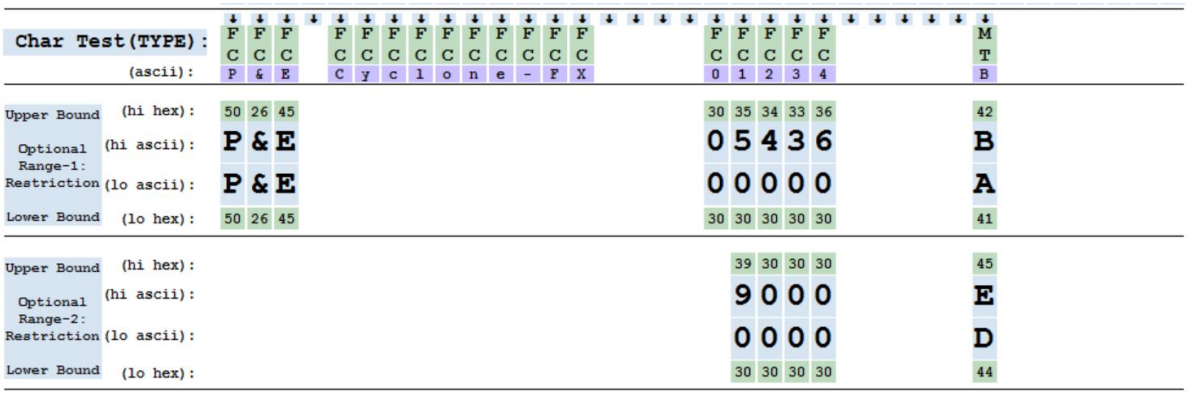


Figure 11-25: After Char Test(TYPE): entries Bulk Copy to Range-1 entries

11.6.1.4 Turning Off Bulk Copy

Clicking on the red text **Set Bulk Copy** button in the program control area disables Bulk Copying. When the Bulk Copy option is deselected, the text in the **Set Bulk Copy** button is recolored to black.

11.6.2 Ctrl-C / Ctrl-V Char Test(TYPE): and Range Copying

This type of copying is used when it is desired to create multiples of the same **Char Test(TYPE):**, **Range-1** or **Range-2** entries. Once an entry has been created in a field, it can be designated as the copy entry that field. To designate an entry, place the cursor on the entry and type a Ctrl-C. The background color for that entry will change to yellow to indicate the designation. Note that only entries with light green boxes can be Ctrl-C designated since ASCII entry boxes would simply accept the Ctrl-C character. To designate a different entry, simply redo the designation process on a different entry.

The designated entry in a given field can then be copied into other existing entries in the same field. To accomplish this, place the cursor on an existing entry in the same field. The information in the designated Ctrl-C entry is copied to the cursor location by pressing Ctrl-V. Only cursor entries with light green boxes can be Ctrl-V copied to since ASCII entry boxes would simply accept the Ctrl-V character. After copying, the cursor is automatically moved right circularly to the next available entry box.

11.6.3 Adding Sample Character Spaces

On startup, this utility if configured to allow for up to 36 character spaces. This number was chosen to fit well on most displays. This number can be increased up to 64 characters. To increase the number of display characters, click on the **Inc Numb Chars** button in the control area of the program as shown in Figure 6.3.

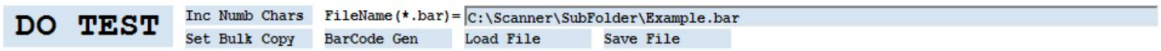


Figure 11-26: Program Control Area

Each time the **Inc Numb Chars** button is clicked, an additional character space is added to the display up to a maximum of 64. Also, the width of the program window is increased to accommodate the added character spaces. The number of display characters is only reset to 36 by restarting the utility.

11.6.4 On Screen Graphical Barcode Generator

Sometimes it is convenient to see what a particular Barcode looks like. To instantiate an on screen barcode display, click on the BarCode Gen button in the program control area.

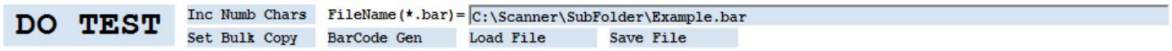


Figure 11-27: Program Control Area

This opens a new window which has an edit box containing "**P&E Microcomputer Systems**", the graphical barcode for this text, and a button Copy to **Sample** and **DO TEST**. Clicking into the edit box allows text to be inserted or edited. Any changes to the edit box are automatically displayed in the graphical barcode which can be scanned. An external target board barcode could be scanned and inserted using a Ctrl-V. The graphical display and the text can be observed to make sure the scanner itself is working correctly.

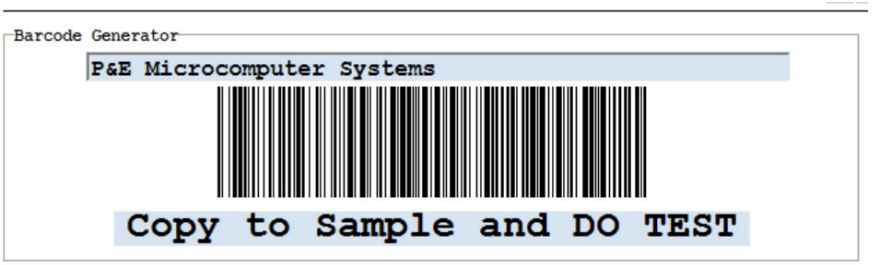


Figure 11-28: Newly opened Barcode Generator Window

Text from the edit box can then be transferred to the main utility screen **Sample** characters and tested against the tests being developed by simply clicking on the **Copy to Sample and DO TEST** button. Any prior **Sample** characters are deleted before the transfer and testing take place.

When this window option is opened, the text in the **BarCode Gen** button is colored red as a reminder to turn the window off by clicking the button again.

11.7 References

[1] Cyclone Universal and Cyclone Universal FX Manual

[2] The JSON Data Interchange Format

[3] JSON Re-formatter

12 SERIAL PORT CONFIGURATION

Standard serial cables may be used for serial port Cyclone configuration.

13 USB PORT CONFIGURATION

Standard USB cables may be used for USB port Cyclone configuration.The user may use network hubs as necessary.

14 SAP_LAUNCH COMMAND-LINE UTILITY

14.1 SAP_LAUNCH Introduction

SAP_LAUNCH is a command line utility developed to assist automated production. A Cyclone must be configured or a SAP image file must be created beforehand.

14.2 SAP_LAUNCH Startup

a)Connect the Cyclone to a PC via RS232, USB, or Ethernet.

b)Connect the Cyclone to the target system via debug port.

c) Power up the Cyclone and the target system.

d) Run the software from a DOS prompt. Command line parameters allowed are:

SAP_LAUNCH [-O log_file] [port=y] [IMAGE=x] [UPDATEIMAGE sap_image_file]

where:

[-O log_file] – Where log_file keeps a log of the operations and result.
It must be the first parameter if used.

[Port=y] - Where the value of y is: (See examples section)

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.

[IMAGE=x] - Where the value of x is the image ID to be executed

[UPDATEIMAGE sap_image_file_name] – Where sap_image_file_name is a previously saved Cyclone SAP image. It is recommended not to use any spaces in the file name and its full path. All current images stored on the Cyclone will be erased. This parameter does not support multiple images.

14.3 SAP_LAUNCH Examples

SAP_LAUNCH Port=COM1

Cyclone is connected to the PC via the COM1 port.

SAP_LAUNCH Port=USB1

Cyclone is connected to the PC via the USB1 port.

SAP_LAUNCH Port=209.61.110.251

Cyclone is connected to a network with IP address of 209.61.110.251

SAP_LAUNCH PORT=USB1 UPDATEIMAGE c:\pemicro\cyclone\file1.sap

Cyclone is connected to the PC via a USB port. The Cyclone will first update its stand alone operation image, then carry out the operations.

14.4 SAP_LAUNCH Sample Batch File

Here is an example of calling the SAP_LAUNCH utility and testing its error code return in a simple batch file. Sample batch files are given for both Windows NT/XP/2000/7/8/10, and Windows 95/98.

Windows NT/XP/2000/7/8/10:

```
@ECHO OFF
SAP_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:

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

14.5 SAP_LAUNCH DOS Error Returns

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

14.5.1 Application Handling-Related Error Codes

160: BM is not pre-configured in the Cyclone.
161: BR is not pre-configured in the Cyclone.
162: EB is not pre-configured in the Cyclone.
163: EW is not pre-configured in the Cyclone.
164: EM is not pre-configured in the Cyclone.
165: PB is not pre-configured in the Cyclone.
166: PW is not pre-configured in the Cyclone.
167: PM is not pre-configured in the Cyclone.
168: VM is not pre-configured in the Cyclone.

169: VR is not pre-configured in the Cyclone.
170: VC is not pre-configured in the Cyclone.
171: USER1 is not pre-configured in the Cyclone.
172: USER2 is not pre-configured in the Cyclone.
173: USER3 is not pre-configured in the Cyclone.
174: USER4 is not pre-configured in the Cyclone.
175: USER5 is not pre-configured in the Cyclone.
176: USER6 is not pre-configured in the Cyclone.
177: Wrong USER function specified.
178: PT is not pre-configured in the Cyclone.

180: Error during power off target.
181: Error during power on target.

190: Wrong 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 is not ready. Please check power and connections.

14.5.2 Additional Error Codes

All other error codes are listed in **CHAPTER 18 - ERROR CODES**.

15 AUTOMATIC SERIAL NUMBER MECHANISM

When producing a microcontroller- or microprocessor-based product, it is often useful to program a unique serial number into the permanent memory (FLASH) of the product.

PEmicro has developed a serial number mechanism to automate this process. Each time you issue a serialization command in the programming software, the current serial number is programmed at a specified address. In addition, the serial number is incremented to the next available serial number and saved for future serialized programming operations.

The Cyclone adopts this automatic serial number mechanism for its stand-alone operations.

15.1 Understanding Serialization

The automatic serial number mechanism supports serial numbers from 1 to 16 bytes in length. Each byte of a serial number ranges between a lower and an upper bound. This approach allows the individual bytes of the serial number to have distinct properties. Some of the forms these properties can take are:

Type	Lower Bound (hex)	Upper Bound (hex)
Constant	Constant	Constant
Binary	00	FF
ASCII Printable	20	7E
ASCII Numeric	30	39
ASCII Upper Case Letter	41	5A
ASCII Lower Case Letter	61	7A
Other	XX	YY

Each serial number and its properties are stored in a separate file. Any file name can be used for the serial number file, however the extension .ser is normally appended because it makes it simpler to locate the file.

A utility called SERIALIZE has been developed to make it easy to create, visualize, edit, and maintain these serial number files.

15.2 Serialize Utility

This section is a modified excerpt from PEmicro's Serialize Help File and explains the Serialize utility in detail.

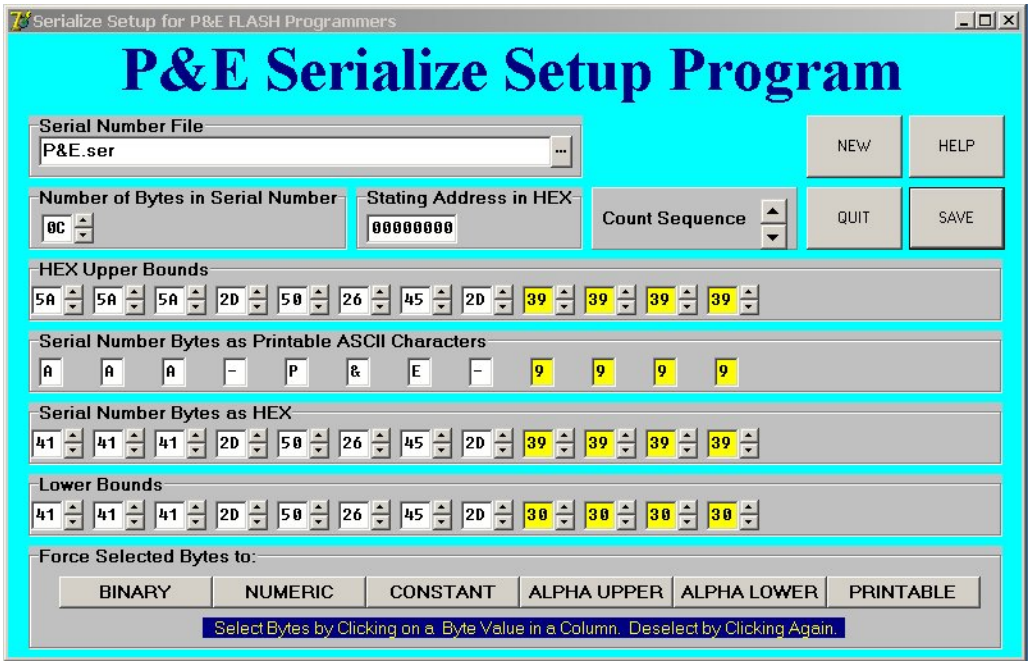


Figure 15-1: Serialize Main Screen

15.2.1 Serial Number File

This edit box shows the currently selected Serial Number File, or else indicates "None Selected". If you try to select a nonexistent file, the selection will revert to "None Selected". On startup the edit box, by default, shows the filename that was in effect the last time the QUIT button was clicked. You can select a new Serial Number File in the following ways:

- Single Click - Lets you directly edit the filename in the edit box. Pressing Enter will check for the existence of the file. If not found, the selected file gets set to "None Selected". If the file exists, the serial number and its properties are displayed on screen.
- Double Click or …Click - Opens a standard file browser and lets you choose from existing files by disk, directory, name, and extension.

15.2.2 Number of Bytes in Serial Number

The up and down arrows let you add or delete bytes for the serial number, max=10 hex (16 base ten), min=1.

- Up Arrow Click - Adds new bytes to the Serial Number. Each byte added appears as a new column in the serial number representation. Added bytes are input as Binary Bytes, i.e. the upper bound is FF and the lower bound is 00.
- Down Arrow Click - Deletes bytes from the right end of the Serial Number. Any previously entered byte properties are lost.

15.2.3 Count Sequence

This window lets you count up or down through the sequencing of the serial number. The serial number is allowed to wrap over the top of the highest serial number or below the lowest serial number. Note that in PEmicro programmers, the serial number can only count up and any attempt to overflow will cause an error.

- Up Arrow Click - Counts the serial number up.
- Down Arrow Click - Counts the serial number down.

15.2.4 Serial Number Bytes as Hex

There is one display column for each byte in the serial number shown as printable ASCII characters. Non-printable ASCII characters are indicated by the small solid block graphic.

- Up Arrow Click - Counts the serial number up.
- Down Arrow Click - Counts the serial number down.

15.2.5 Hex Upper Bounds

There is one display column for each upper bound of the byte in the serial number in hex.

- Up Arrow Click - Increases the upper bound by one with a maximum of FF Hex.
- Down Arrow Click - Decreases the upper bound by one with a minimum of the current serial number byte value.
- Double Click on Hex - Selects or de-selects the byte column. Selected shown in yellow. The serial number byte in this column may then be modified using the buttons at the bottom of the Serialize utility. Please refer to **Section 15.2.11 - BINARY, NUMERIC, CONSTANT, ALPHA UPPER, ALPHA LOWER, and PRINTABLE**.

15.2.6 Hex Lower Bounds

There is one display column for each byte of the lower bound of the serial number in hex.

- Up Arrow Click - Increases the lower bound by one with a maximum of the current serial number byte value.
- Down Arrow Click - Decreases the lower bound by one with a minimum of 00 Hex.

15.2.7 NEW

Instructs the program to start editing a NEW (as yet un-named) serial number file. It will throw away the information for any serial number currently being edited unless that information has been saved (Save Button). The new serial number is initialized with one (1) byte of binary.

15.2.8 SAVE

Instructs the program to save the current serial number being edited into the file name and path shown in the Serial Number File window. If a file name has not been provided, i.e. the window shows None Selected, then an error is displayed in a red window on the screen. If this happens, type in a filename in the window and click Save again.

15.2.9 HELP

Opens the Serialize help system (serialize.hlp file, i.e. this file) for perusal.

15.2.10 QUIT

Turns off the Serialize Program and saves any setup information in the file Serialize.ini. This file will initialize the setup information the next time the program is started. Xing out of the program (top right of screen) does not save the setup info.

15.2.11 BINARY, NUMERIC, CONSTANT, ALPHA UPPER, ALPHA LOWER, and PRINTABLE

These buttons are used to set the properties of selected (colored yellow) bytes of the Serial Number. Individual bytes whose properties you wish to modify are selected or deselected by double-clicking in the Hex Upper Bounds box in the column that corresponds with the values for a particular byte.

15.3 Serialize Utility Example

This example shows:

1. Currently editing file C:\Example.ser
2. Number of bytes in the serial number is 10 Hex (= 16 base ten)
3. Starting address is 0000000 Hex
4. Next Serial number is AAA-P&E-9999 in ASCII

a. First 3 bytes are Upper Case Alphabetic ASCII (AAA)

b. Next 5 bytes are Constants (-P&E-)

c. Last 4 bytes are Numeric ASCII (9999)

4. This provides for a maximum of 6,760,000 (26x26x26x10x10x10x10) serial numbers from AAA-P&E-0000 to ZZZ-P&E-9999.

5. The last 4 bytes of the serial number are selected (colored yellow) so that their properties can be changed using the forced selected byte buttons on the bottom of the screen.

15.4 Using Serial Number File

The command to invoke the serial number file in PEmicro's interactive programming software is "CS Choose Serial File". The command to actually program the serial number to target and automatically increment the serial number afterward is "PS Program Serial Number".

PEmicro's command line software uses the same commands in a command line fashion to invoke the serial number file, initiate its programming, and increment:

```
CS serial_number_file.ser
PS
```

15.5 Serial Number Handling

The **CYCLONE FX** firmware implements the automatic serial number mechanism (see **Section 5.2.2.3 - Modify Next Serial Number**). The same serial number files are used with the Cyclone Image Creation Utility, and the same commands are used to specify the serial number file and initiate serial number programming and incrementation. The serial number data structure is saved in the SAP image. Once a "PS" command is carried out, a serial number is programmed into the target. Only after all operations have been completed successfully does the Cyclone firmware automatically increment the serial number and store it in the Cyclone's flash for internal images (or external CompactFlash for external SAP images).

The CS and PS commands are not present in the Cyclone Image Creation Utility until a valid programming algorithm is specified.

To complement the Cyclone's usage in production environments, the Cyclone supports multiple serial number structures for each programming algorithm block. Each SAP image may contain multiple programming algorithms for every memory module it needs to program, and each programming algorithm block may contain multiple serial number structures. The SAP image sequence below illustrates this briefly:

```
CM algorithm_file_1
SS object_code_1
EM
PM
VC
CS serial_file1.ser
PS
CS serial_file2.ser
PS
CS serial_file_3.ser
PS
CM algorithm_file_2
SS object_code_2
EM
PM
```

```
VC
CS serial_file4.ser
PS
CS serial_file5.ser
PS
```

16 SAP CONVERTER UTILITY

Customers who have used our older Cyclones, such as the Cyclone PRO, Cyclone MAX, Cyclone for ARM devices Rev. A/B, etc., will find that their SAP images for these older generation Cyclones will not work on the newer **CYCLONE** and **CYCLONE FX** programmers. Simply recreating these images for current generation Cyclones could potentially introduce errors and lose information about commands, settings, and configurations.

Therefore, we created the “SAP_Convert_Console.exe” which must be used to convert older generation SAP images into current generation SAP images. Once converted, an image will work not only on **CYCLONE** and **CYCLONE FX** programmers, but it will also remain compatible with the Cyclone for which it was originally created.

SAP_Convert_Console.exe is a Windows command line utility and the software must be run through the Windows Command Prompt. The utility can be found in the same folder as the Cyclone’s software install path.

The command line parameter syntax:

>SAP_Convert_Console [old_SAP_path] [new_SAP_path]

Where:

[old_SAP_path]	The relative or full path to the SAP file. Usually has the .SAP file extension.
[new_SAP_path]	Optional parameter where the user can specify a relative or full path to dump the output of the conversion. If path and file name matches the input, then the output file will replace the input file. If this parameter is not specified, the output will be dumped in the same path as the input file renamed with postfix “_2”. For example if the input is myfile.SAP, then the output will be myfile_2.SAP and will not replace the original input file.

17 TROUBLESHOOTING

This section answers some common questions that should help the user with various aspects of **CYCLONE FX** operation.

What is bootloader mode?

Bootloader Mode is a special running mode of the Cyclone Universal and Cyclone Universal FX in which only limited funtionality of the Cyclone is allowed. In this mode, the Cyclone will allow communication to a PC via USB, ethernet or serial ports. In Bootloader Mode the user can update the Cyclone firmware via the cyclone utilities.

The Bootloader screen will display the version of the bootloader, the version of the internal and external application, the name of the Cyclone and it's IP address.

When do you use bootloader mode?

If the Cyclone ever becomes unresponsive, communication to the PC is not possible via USB, ethernet, or Serial ports and if the cyclone fails to power on.

How do you enter bootloader mode?

You can force the Cyclone into booloader mode witht the following sequence with the Cyclone powered:

- Press the Reset button
- Press the Start button
- Release the Reset button
- Tap the Cyclone LCD screen 3 times
- Release the Start button

18 ERROR CODES

The **CYCLONE FX** will indicate errors using the following codes. Please contact PEmicro if instructed or if you are unsure of the specific meaning of an error code.

18.1 Debug Mode Communication Related Errors

$0001: No target device response.

$0002: Invalid target device response.

$0003: Programming operation canceled.

$0004: Error while waiting for programming operation to complete.

$0005: Error attempting to detect the communication speed.

$0006: Error: Attempt to unsecure the device was unsuccessful.

$0007: An error occurred while entering debug mode.

$0008: Error entering debug mode. The device is secured.

$0009: Error entering debug mode for verification.

$000A: Error writing data to target.

$000B: Error enabling or disabling device for programming.

$000C: Error performing timing test.

$000D: Error finalizing the programming process.

$000E: Error: Vendor hardware is not supported.

$000F: Error generating VPP high voltage.

18.2 SAP Image Handling Related Errors

$0011: No image selected

$0012: Error validating image CRC

$0013: SAP operation was not found. Error: SAP operation pointer not found

$0014: SAP image storage was not initialized

$0015: SAP image transfer error, odd length is not allowed

$0016: SAP image transfer error, invalid start address

$0017: SAP image transfer error while writing to storage

$0018: Error writing the serial number structure storage

$0019: Error writing the menu structure storage

$001A: Error erasing internal memory

$001B: Error: Image requires higher firmware version

$001C: Image version is not supported. Please update firmware.

$001D: Out of RAM memory. Try reset Cyclone.

$001E: SAP image storage failure

$001F: Old SAP image format, not supported.

$0020: Programming image is not accessible

$0031: System reset occurred

$0032: Error system is busy with other operations.

$0033: Error system is busy with too many inquiries.

18.3 SAP Algorithm header Operation Handling Related Errors

$0060: Unsupported SAP image.

$0061: Undefined header operation

$0062: Operation in algorithm header has failed.

18.4 SAP Operation Related Errors

$0080: SAP operation is not supported.

$0082: Target type mismatch

$0083: SAP operation cancelled

$0084: Running algorithm failure

18.5 SAP Blank Check Range and Module Related Errors

$1001: Blank Check is not supported by this algorithm.

$1002: Blank Check algorithm was not found.

$1003: Blank Check operation failed

18.6 SAP Erase Range and Module Related Errors

$2001: Erase error, algorithm not supported

$2002: Erase error, algorithm not found

$2003: Erase error, module failed or cancelled

$2004: Erase error, module failed, target is still secured

$2005: Erase error, module not performed, data is preserved

18.7 SAP Program Byte, Word, and Module Related Errors

$3001: Program error, algorithm not supported

$3002: Program error, algorithm not found

$3003: Program operation failed or was cancelled

$3004: Program operation failed, write protected

$3005: Program error, Data size exceeds the limit

$300A: Error during reading data range, invalid data length

$300B: Error during reading data range, invalid start address

$300C: Error during reading data range, no target power

$300D: Error during programming data range, invalid data length

$300E: Error during programming data range, invalid start address

$300F: Error during programming data range, no target power

$3010: Error reported while running the custom test application (RT) on the target.

$3011: Error displaying feature

$3012: Error programming feature

$3013: Error overlaying feature

$3014: Error: Run Test Operation terminated

$3015: Error: Run Test Operation over character limit 255

$3016: Error run test operation failed

$3017: Error: unable to allocated memory during run test.

$3040: Error: Program may cause the device to be secured permanently

18.8 SAP Verify Checksum Related Errors

$4001: Verify Checksum not supported

$4002: VC failed, invalid algorithm was used

$4003: VC operation failed or was canceled

$4011: VV command not supported

$4012: VV failed, invalid algorithm was used

$4013: VV operation failed or was canceled

18.9 SAP Verify Range and Module Related Errors

$5003: Error during verifying module.

18.10 SAP User Function Related Errors

$6003: Error during user functions.

18.11 SAP Trim Related Errors

$7001: Program Trim operation is not supported

$7003: No target response during a Program Trim operation

$7004: Program Trim error. Trim value is not set

$7007: Program Trim error. Trim value failed

$7008: Trim error. Trim value read failed

$7009: Trim value invalid, value is $00 or $FF

$700A: Trim value is invalid. Trim value is already programmed.

18.12 Unrecoverable Fatal Errors

$8001: Fatal Error: please contact PEmicro.

$8002: Fatal Error: please contact PEmicro.

$8003: Fatal Error: please contact PEmicro.

$8004: Fatal Error: please contact PEmicro.

$8005: Fatal Error: please contact PEmicro.

$8006: Fatal Error: please contact PEmicro.

$8007: Fatal Error: please contact PEmicro.

$8008: Fatal Error: please contact PEmicro.

$8009: Fatal Error: please contact PEmicro.

$800A: Fatal Error: please contact PEmicro.

$800B: Fatal Error: please contact PEmicro.

$800C: Fatal Error: please contact PEmicro.

$800D: Fatal Error: please contact PEmicro.

$800E: Fatal Error: please contact PEmicro.

$800F: Fatal Error: please contact PEmicro.

$8010: Fatal Error: please contact PEmicro.

$8011: Fatal Error: please contact PEmicro.

$8012: Fatal Error: please contact PEmicro.

$8013: Fatal Error: please contact PEmicro.

$8014: Fatal Error: please contact PEmicro.

$8015: Fatal Error: please contact PEmicro.

$8016: Fatal Error: please contact PEmicro.

$8017: Fatal Error: please contact PEmicro.

$8018: Fatal Error: please contact PEmicro.

$8019: Fatal Error: please contact PEmicro.

$801A: Fatal Error: please contact PEmicro.

$801B: Fatal Error: please contact PEmicro.

$8020: Fatal Error: please contact PEmicro.

$8021: Fatal Error: please contact PEmicro.

$8022: Fatal Error: please contact PEmicro.

$8023: Fatal Error: please contact PEmicro.

$8024: Fatal Error: please contact PEmicro.

18.13 Operation Security Related Errors

$9001: Error: Exceeds image specified Program Limit

$9002: Error: Exceeds image specified Error Limit

$9003: Error: Exceeds Image specified Date Range

$9004: This programming image has usage restrictions enabled. Requires Cyclone FX hardware.

$9005: Error: This programming image has barcode enabled. Requires Cyclone FX hardware.

$9006: Error: This programming image has command RT Run Code in Test/Calibration Mode. Requires Cyclone FX hardware.

$9007: Error: This programming image has command DF Display Feature Data. Requires Cyclone FX hardware.

$9008: Error: This programming image has command PF Program Feature Data to Address.Requires Cyclone FX hardware.

$9009: Error: This programming image has command OF Overlay Feature Data over File Data.Requires Cyclone FX hardware.

18.14 External Memory-Related Errors

$A001: Error writing to external memory card

$A002: Error formatting the external memory card

$A003: External memory card was disconnected during use

$A004: External memory card has unsupported format

$A005: External memory card has corrupted data

$A006: Faulty external memory card.

$A007: Failed during internal memory verification

$A008: Failed during external memory card verification

$A009: Error while reading external memory card for image pointer

$A00A: Error: Read-only lock is enabled in the external memory card.

18.15 Serial Number Related Errors

$B001: Error erasing the serial number storage

$B002: Error writing serial number

$B003: Serial number is over the limit, up to 255 can be supported at a time

$B004: Error loading Serial Number structure from reset

$B005: Error during serial number structure update

$B006: Error: Serial Number structure was not found.

$B007: Error: Serial Number structure is invalid

$B008: Error programming Serial Number to target.

$B009: Error obtain Serial Number from storage.

18.16 Download Count Related Errors

$C001: Error erasing the download counts storage

$C002: Error writing the download counts

$C003: Download counts is over the limit, up to 255 can be supported at a time

$C004: Error trying to convert the download counts structure

18.17 System Hardware/Firmware/Logic Recoverable Errors

$D001: Error: Firmware does not exist

$D002: Error: Firmware update is not allowed

$D003: Error: Firmware update has failed

$D004: Error: There is a firmware mismatch during firmware update.

$D005: Error: Voltage calibration failure.

$D006: Error: Cannot either read or write disk

18.18 Barcode Scanner Errors

$E001: The barcode file for this image is invalid or corrupt.

$E002: The barcode scanned does not apply to a programming image.

$E003: The barcode scanned applies to more than one programming image.

$E004: The barcode scanned exceeds the maximum size of 256 characters. Please make sure that the scanner adds a Carriage Return Line Feed at the end of a transmission.

$E005: The barcode scanned includes unsupported characters.