Dust Networks

Eterna™ Serial Programmer Guide

For use with the following hardware, evaluation or development kits (see "Supported Products" for more details):

DC9010A or DC9010B

Table of Contents

About This Guide

This document describes the use of the Eterna Serial Programmer hardware (DC9010A or DC9010B, two similar versions referred as DC9010 in this document[1]) and the Eterna Serial Programming software application (ESP). The current version of this document may be found on Linear Technology website. The ESP software application is available at the following link: http://www.linear.com/docs/43081.

Audience

This document is intended for system developers, hardware designers, and software developers.

Background

The Eterna SoC includes 512kB of flash memory similar to general purpose microcontroller devices. Eterna also includes a programming interface that may be enabled by asserting both the RESETn and FLASH_P_ENn signals. This interface provides the fastest method to program images on Eterna based products. All LTC5800 devices will ship with the flash erased and must be programmed to enable operation. While some previous LTP products shipped pre-programmed, all current LTP products must be programmed prior to operation: the combination of application software (mote or manager) and fuse table determines the devices role (e.g. LTP5901-IPMA, IPRA, or IBRB). The DC9010 and ESP software have been optimized for fast programming of Eterna.

Eterna requires the use of a 20 MHz crystal oscillator. Eterna includes an internal capacitor array for centering (trimming) the 20 MHz crystal reference for each layout. To insure proper operation, the correct value for the load trim must be determined for each new design. The DC9010 must be used to determine the correct load trim value for a particular layout.

Supported Products

The DC9010 hardware and the ESP software will support all products based upon the Eterna platform. The current list of supported products is:

- LTC5800
- LTP5900
- LTP5901
- LTP5902

---

[1] See Hardware Versions on page 35

Any future products based on the Eterna platform will also be supported by the DC9010 programmer and ESP software.

The ESP software may also be used with a DC2274, other Eterna based evaluation boards connected to the DC9006 or future Eterna devices similarly accessed via an FTDI USB-to-serial converter.

Related Documents

The following related documents are available:

Eterna Integration Guide

Eterna Board Specific Parameter Configuration Guide

Eterna Serial Programmer Board Files

Conventions and Terminology

This guide uses the following text conventions:

- `Computer type` indicates information that you enter, such as a URL.
- **Bold type** indicates buttons, fields, and menu commands.
- *Italic type* is used to introduce a new term.
- **Note:** Notes provide more detailed information about concepts.
- **Caution:** Cautions advise about actions that might result in loss of data.
- **Warning:** Warnings advise about actions that might cause physical harm to the hardware or your person.

Revision History

Revision	Date	Description
040-0110 rev 1	2/9/2012	Initial Release
040-0110 rev 2	12/05/12	Added Background, Programmer Hardware and crystal characterization procedure. Corrected read with offset command call.
040-0110 rev 3	3/3/13	Add Eterna Flash Emulator Section Update Target Crystal Characterization
040-0110 rev 4	5/3/13	Added histograms for target crystal calibration
040-0110 rev 5	5/30/13	Editorial changes to target crystal calibration language.
040-0110 rev 6	8/23/13	Add reference to DC9010 on front page Add mxtal usage restriction (radio off)
040-0110 rev 7	11/12/13	Add Hardware Versions section Update document formatting and block diagram
040-0110 rev 8	01/13/15	Added Troubleshooting section Added section on Using an External Supply Voltage
040-0110 rev 9	04/20/15	Update for USB 3.0 (ESP version 1.1.1 and higher)
040-0110 rev 10	09/23/15	Addendum: Fuse Table and 20 MHz crystal change
040-0110 rev 11	09/23/16	Adjust information related to locked devices Add manufacturer part number of connectors Correct maximum programming frequency

Getting Started

Introduction

The DC9010 Eterna Serial Programmer consists of an enclosed circuit board with a USB micro-B interface and a 2x5 2mm ribbon cable.

The DC9010 allows in-circuit access to an Eterna mote-on-a-chip installed with the programming header recommended in the Eterna Integration Guide. An LTP5900 can be programmed directly on the DC9010 by removing the cover.

The DC9010 is used in conjunction with the Eterna Serial Programmer (ESP) software utility described in this document.

Figure 1   DC9010 Eterna Serial Programmer

System Requirements

- PC running Windows
- USB 2.0 or USB 3.0[2] port

---

[2] Refer to page 24 for details.

Hardware Setup

Connect the DC9010 to the Windows PC via USB and connect the target to the DC9010 via the ribbon cable.
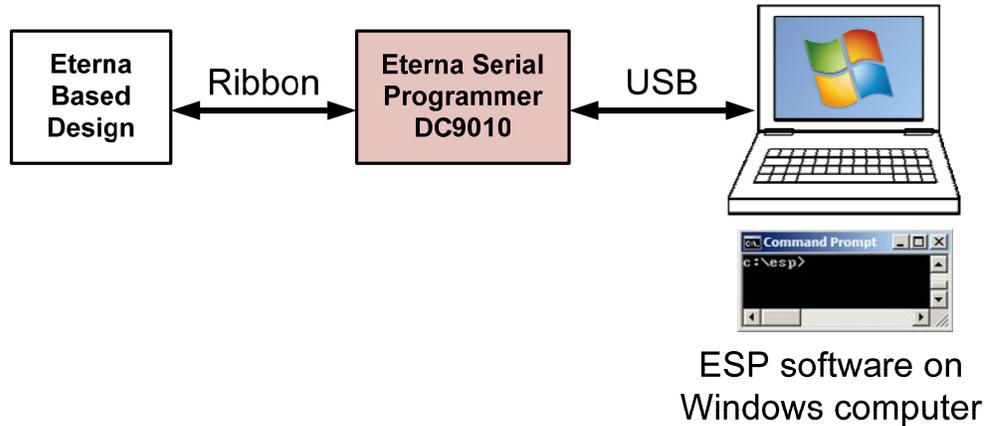

Figure 2   Programming Configuration

Software Installation

ESP software is distributed as a .zip archive and does not require installation. ESP software can be downloaded from http://www.linear.com/dust_programmer. To install, unarchive all files into a directory (e.g. C:\esp).   ESP software calls FTDI, http://www.ftdichip.com/, drivers that are required for operation. The FTDI drivers can be found at http://www.ftdichip.com/Drivers/D2XX.htm and are referred to by FTDI as "D2XX Drivers". ESP software has been tested against D2XX Drivers revision 2.08.14.

The utility should be executed from the directory where you placed the files.

Setup

The Eterna Serial Programming solution is comprised of the Windows ESP.exe application, which in turn use the drivers supported by Future Technology Devices International (FTDI) to interface to the Eterna Serial Programmer via USB. Pairing of USB hardware to drivers is most easily accomplished in most systems by connecting the hardware and following the Windows driver installation instructions.

FTDI hardware solutions are very common and as such there is a reasonable chance that the required drivers have already been installed in a system. It should also be noted that as the drivers are generic to many solutions, for some of the ESP commands (e.g. those involving locked parts), installation of FTDI Virtual COM port (VCP) drivers is required in addition to the D2XX drivers. For Windows, both types of drivers are bundled together. The Eterna Serial Programmer's target COM port is a required argument to these commands.

Note that the pairing of the FTDI hardware is done to a specific USB port on a system. Changing of the USB port used to pair the Eterna Serial Programmer to the FTDI driver

will result in having to reinstall the driver and additional manual notation of the Eterna Serial Programmer's target COM ports.

To pair the Eterna Serial Programmer to the FTDI driver on a system:

1) Connect the USB cable between the Eterna Serial Programmer and the system.

   If four new COM ports appear in the device manager, go to step 6.

   If the Found New Hardware Wizard appears, go to step 2.

   If the Found New Hardware Wizard does not appear, do the following:

   a. Ensure that the port is functional, and that the device is connected correctly. If the Wizard still does not appear, open the Windows Device Eterna Serial Programmer to see how Windows has recognized the device.

   b. If a new "Question Mark Icon" appears , right-click the device and select Update Driver. This displays the Found New Hardware Wizard.

   c. Go to step 3.

2) In the Wizard, click the option to "Install from a list or specific location," and click Next.

3) Select the box to "Include this location in the search." Then, use the Browse button to navigate to the directory where ESP and the associated drivers have been stored , and click Next.
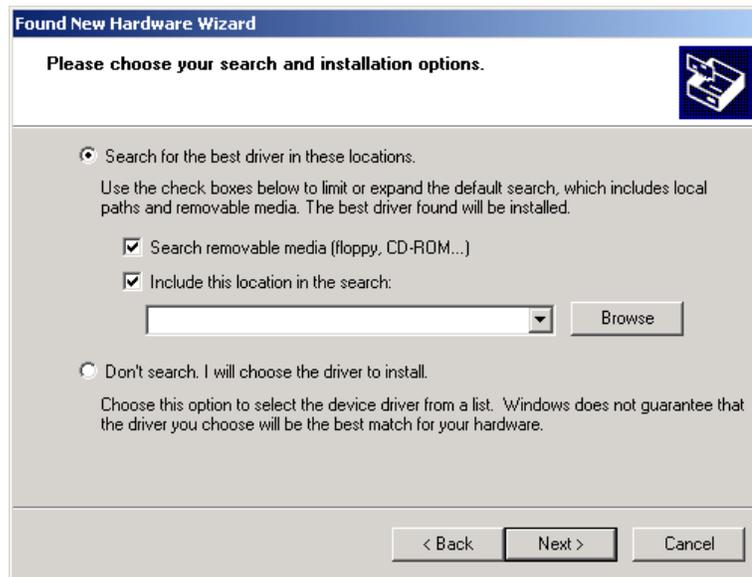
4) After the Wizard installs the software, click **Finish**.

5) When the Found New Hardware Wizard reappears, repeat steps 1 through 4 to continue the installation. Repeat these steps each time the Wizard appears.

Because of the way Windows works, you may be prompted to go through the Wizard up to eight times to complete the installation and mapping of the USB port. The Eterna Serial Programmer will install a total of four virtual serial ports, along with the USB drivers to control them.

6) When the installation and mapping of the USB ports is complete, open the Device Named Eterna Serial Programmer to find out the COM port numbers that have been assigned to the virtual serial ports. The third COM port number listed will be the COM port used by the ESP application for communications with the Eterna Serial Programmer.

a. Make a note of the third COM port identifier.

For example, if the new ports are COM3, COM4, COM5, and COM6, the PORT parameter used in some ESP commands and to initiate crystal characterization will be on COM5.

7) Configure the following Advanced Settings for each of the four new COM ports:

a. **Right-click** on a COM port and click **Properties**.

b. Click the **Port Settings** tab, and then click **Advanced**.

c. Deselect the **Serial Enumerator** option, and click **OK**.

d. Click **OK** to return to the Device Manager.

e. Repeat this step for each of the four new COM ports. When you are finished, close the Device Manager.
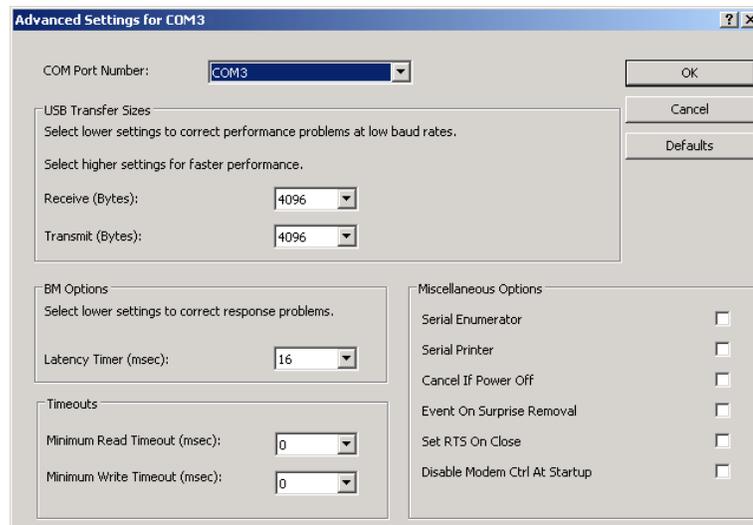
Commands

Commands are issued via a Windows Command prompt. The commonly used command options are documented below – see the Help for how to view a complete list of supported commands.

Help

For a complete list of all the ESP options, enter:

```
C:\...\ESP\ESP
```

Erase

The entire flash is 512 KB in size. That flash is organized in 256 pages of 2 KB each. The pages are numbered 0 through 255. Each page has a starting address that is represented in hexadecimal. So, page 0 starts at address 0x0. Page 1 starts at 0x800, and so on.

To erase the entire 512 KB of flash, enter:

```
C:\...\ESP\ESP –E
```

To erase a select number of pages of flash, enter:

```
C:\...\ESP\ESP –e OFFSET PAGES
```

Where OFFSET is in hexadecimal with no leading 0x and must be in multiples of 800 hexadecimal (2 KB) and PAGES is the decimal number of page(s) to be erased. For example, to erase various flash image components described in the following Flash Image Structure section, enter:

`C:\...\ESP\ESP –e 0 1`	to only erase the Fuse Table (one page starting at address offset 0x0)
`C:\...\ESP\ESP –e 800 1`	to only erase the Partition Table (one page starting at address offset 0x800)
`C:\...\ESP\ESP –e 1000 237`	to only erase the Main Executable (237 pages starting at offset 0x1000)
`C:\...\ESP\ESP –e 77800 17`	to only erase the Loader (17 pages starting at 0x77800)

Read (unlocked device)

To read the entire 512 KB of flash and store the image in a file, enter:

```
C:\...\ESP\ESP –r FILENAME
```

To read the entire a subset of flash and store the image in a file, enter:

```
C:\...\ESP\ESP –R FILENAME OFFSET BYTES
```

Where OFFSET and BYTES are in hexadecimal with no leading 0x characters.

Read (locked device)

A device can be locked with the Hardware Lock Key (see *040-0109 Eterna Board Specific Parameter Configuration Guide* for details). A locked device prevents access to all internal memory including the flash. To read a locked device enter:

```
C:\...\ESP\ESP –U FILENAME  KKKKKKKK PORT
```

Where KKKKKKKK is the Hardware Lock Key and PORT is the PC's third USB Serial Port assigned to the programmer (eg. for COM10, PORT would be 10). This will temporarily unlock the device and read the flash and store the entire 512 KB in a file called FILENAME. This will not modify the locked state of the device. The device will remain locked for any user who does not have the correct Hardware Lock Key. Note that UARTC0_RX can always be used for unlocking a device, regardless of how the pin is configured for use in the product.

Program with Verify

To program an image enter:

```
C:\...\ESP\ESP –P FILENAME OFFSET
```

Where OFFSET is in hexadecimal with no leading 0x and must be in multiples of 800 hexadecimal (2 KB).   The device will only program the number of bytes from the starting OFFSET to OFFSET + size(FILENAME). If the verification is successful ESP will report on a new line:

```
Verify: PASS
```

If the verification is unsuccessful ESP will report on a new line:

```
Verify: FAIL
```

Followed by a line indicating which address failed, the expected value and the value read from the failing location.

Verify

To verify an image enter:

```
C:\...\ESP\ESP –V FILENAME OFFSET
```

Where OFFSET is in hexadecimal with no leading 0x and must be in multiples of 800 hexadecimal (2 KB).   The device will only verify the number of bytes from the starting OFFSET to OFFSET + size(FILENAME). If verification is unsuccessful ESP will report the first difference. If the verification is successful ESP will report:

```
Verify: PASS
```

If the verification is unsuccessful ESP will report on a new line:

```
Verify: FAIL
```

Followed by a line indicating which address failed, the expected value and the value read from the failing location.

Unlock (persistent)

A device can be locked with the Hardware Lock Key (see *040-0109 Eterna Board Specific Parameter Configuration Guide* for details). A locked device prevents access to all internal memory including the flash. **This command will alter part of the first page in flash, erasing the lock key and invalidating the contents of the fuse table following the lock key in the process.** To unlock a locked device enter:

```
C:\...\ESP\ESP –u KKKKKKKK PORT
```

Where KKKKKKKK is the Hardware Lock Key and PORT is the PC's third USB Serial Port assigned to the programmer. For example, if 0x1234abcd was enabled as Hardware Lock Key in the Fuse Table, and the 3rd programmer port is COM10, the following command unlocks the part:

```
C:\...\ESP\ESP –u 1234abcd 10
```

Once a device is permanently unlocked, if using the part or locking is again required afterwards, the device will have to be reprogrammed with a fuse table containing the desired Hardware Lock Key (as described in the *040-0109 Eterna Board Specific Parameter Configuration Guide*).

Programmer Identification

When multiple programmers are connected to a single computer, ESP requires a specific programmer to be identified.

-i LOCID

When the programmer is connected to a USB 2.0 port, the –i LOCID parameter may be used. The LOCID argument is given by the esp –l command for each of the connected programmer. The LOCID of the device listed as "B" must be used.

-n DEVNAME

When the programmer is connected to a USB 2.0 or USB3.0 port, the –n DEVNAME parameter should be used. The DEVNAME argument is given by "devString" in the esp –l command. The DEVNAME of the device listed as "B" must be used.

For example, the following reads a complete flash image into a file named image.bin from a Eterna mote-on-a-chip connected to "ETERNA SERIAL PROGRAMMER B".

```
C:\...\ESP\ESP –L
ESP, Eterna SPI Programmer, version 1.1.1-7
FTCSPI.dll = 2.1.2.2
locID[0] = 0x0, devString = Dust Interface Board A
locID[1] = 0x0, devString = ETERNA SERIAL
PROGRAMMER A
locID[2] = 0x0, devString = Dust Interface Board B
locID[3] = 0x0, devString = ETERNA SERIAL
PROGRAMMER B

C:\...\ESP\ESP -n "ETERNA SERIAL PROGRAMMER B" -r
image.bin
```

-t TYPE

This command is not used in normal operation. Only if the devString has been changed from the factory settings (as shown on page 27), the –t TYPE argument must be given to ESP in order to identify which type of programming board is connected. The TYPE is a numeric argument corresponding to each board supported by ESP:

```
-t TYPE  : Specify target TYPE, default 0.
           3 = Dust Interface Board (DC9006)
           4 = Eterna Serial Programmer (DC9010)
           5 = DC2274 with Memory
           6 = DC2274 without Memory
```

For example, the following reads a complete flash image from a LTC5800 connected to a renamed programmer "DC9010-PROGRAMMER-1 B" using type 4.

```
C:\...\ESP\ESP -L
ESP, Eterna SPI Programmer, version 1.1.1-7
FTCSPI.dll = 2.1.2.2
locID[0] = 0x0, devString = DC9010-PROGRAMMER-1 A
locID[2] = 0x0, devString = DC9010-PROGRAMMER-1 B

C:\...\ESP\ESP -t 4 -n "DC9010-PROGRAMMER-1 B" -r
image.bin
```

Flash Image Structure

The flash image of an Eterna mote is constructed with multiple individual binary files called image components which are divided as shown in Figure 3.
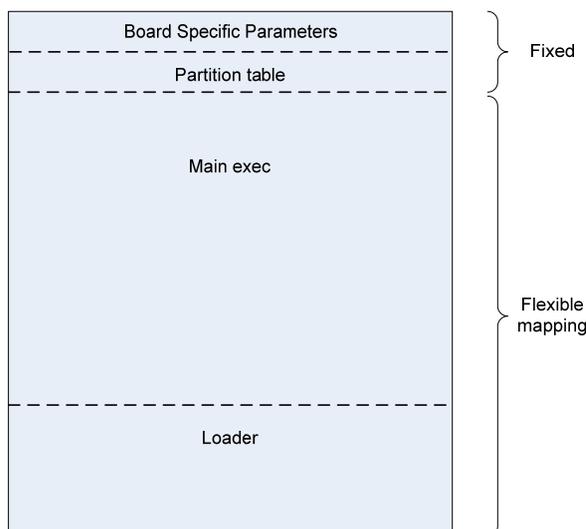

Figure 3   Eterna Mote Flash Image and Components

Users creating an image for the first time will need to load four individual images:

Image Component Name	Start Address (Hex)	Description
Board Specific Parameters or "Fuse Table"	0	2 KB image containing hardware and software configuration settings. This file is required for proper operation and is created via the fuseTable.exe application described in Eterna Board Specific Parameter Configuration Guide.
Partition Table	800	Defines the location of the elements in the Flexible mapping portion of Eterna's image, including the Main Executable and Loader. The partition table is currently agnostic to the particular product software variants.
Main Executable	1000	The main executable image. Each variant of the Eterna product family will have a corresponding software image.
Loader[3]	77800	The loader manages handling of completed Over The Air Programming (OTAP) images and starting the Main Executable image. The loader is currently agnostic to the particular product software variants.

---

[3] The Loader is only required for mote devices, managers do not require a Loader

For LTC5800 devices users will need to create the board specific parameters (fuse table) image - see the Eterna Board Specific Parameter Configuration Guide for details. Access to the Partition Table, Main Executable and Loader images should be obtained through your myLinear account.

A full image consists of the complete flash content with all components. A full image may be copied from one device to a second device (of the same design and layout). The full image may be created by reading the entire 512 kB image of an already programmed device with the following command:

```
C:\...\ESP\ESP –r FullImage.bin
```

Flash Image Programming

To program all four of the image components onto an **erased** Eterna mote-on-a-chip via ESP enter:

```
C:\...\ESP\ESP –P FuseTable.bin 0
C:\...\ESP\ESP –P PartitionTable.bin 800
C:\...\ESP\ESP –P Main.bin 1000
C:\...\ESP\ESP –P Loader.bin 77800
```

To program a full image onto an **erased** device enter:

```
C:\...\ESP\ESP –P FullImage.bin 0
```

Troubleshooting

USB Device Presence

The first step is to determine the presence of the desired device.

ESP -L shows a listing of connected devices, with a location ID and string description. Each device should show two location IDs, one with an 'A' string, and one with a 'B' string. ESP operates on the device with location ID associated with the 'B' string.

For example: 'Dust Interface Board A', and 'Dust Interface Board B'. Supported devices are:

- Dust Interface Board;
- Dust Huron;
- ETERNA SERIAL PROGRAMMER; and,
- ETERNA USB MANAGER.

USB 2.0 or USB 3.0

Early versions of ESP.exe (prior to version1.1.1) only support USB 2.0, due to missing Location ID references returned by software drivers from USB 3.0 ports.

USB 3.0 is now supported in version 1.1.1 and higher of ESP.exe with the following restrictions:

- A single ESP device is connected and its Device Name is the factory default; or,
- Multiple devices which have distinct Device Names (Note DC9010A and DC9010B are shipped with the device name set to the factory default, so use of multiple DC9010A/B programmers on the same system on USB 3.0 ports will require configuration of each DC9010A/B programmer.

If multiple devices with the same Device Name need to be connected to the USB 3.0 ports of a system (for example multiple DC9010), refer to Device Name (USB 3.0 on page 24.

Multiple Devices

Multiple devices are supported by using the location ID and the -i option or the –n option, as described on page 24..

HSP_ERR_NODEV indicates that the desired device can not be found, which can occur when the location ID is not entered correctly.

HSP_ERR_OPENDEV indicates that the desired device may not be opened, possibly due to the device already being opened. Try closing other applications that may be using the resource, such as an open terminal on one of the ports associated with the FTDI interface for example.

Accessing Eterna Flash

Once an appropriate device is established, the -r command can be used to obtain additional information.

If an unlocked powered ETERNA is not present, the HSP_ERR_FLASHID error is returned. If a locked ETERNA is present, the HSP_ERR_FLASHID error is returned for all commands except the unlocking commands –u and –U. If an unlocking command is attempted on an unlocked part, the HSP_ERR_FLASH_ID error is returned.

If the read is successful, all other valid ESP commands should function.

HSP_ERR_BADPARAM indicates a parameter out of range, such as an address outside of 0x80000.

Erase Flash before Programming

When programming an image, the corresponding flash memory must have been previously erased. Production LTC5800 parts are shipped by Linear with the flash erased. Production LTP (PCB) parts are shipped by Linear with the flash images loaded.

Other Common Issues

Other ESP errors are usually internal errors associated with the installation, which includes: ESP.exe, FTCSPI.dll (HSP_ERR_LOADLIB), FTD2XX.dll, and the FTDI driver.

Error Codes

The following table describes common error codes and numbers.

Error Code	Number	Description
HSP_ERR_OPENDEV	3	USB device already open or internal error
HSP_ERR_NODEV	4	USB device not found (when not using -i option)
HSP_ERR_LOADLIB	7	FTDI/FTCSPI installation error
HSP_ERR_FLASHID	12	ETERNA FlashID not found; target not present or locked
HSP_ERR_MAXNUMDEV	17	USB configuration error, too many high speed devices found (64 units max)
HSP_ERR_BADTYPE	18	Incorrect target type passed via the -t option
HSP_ERR_NOCOM	19	Invalid COM port or already open; Fail to open COM port in case of an unlock command
HSP_ERR_NOTUNLOCKED	22	Failed to unlock ETERNA part, invalid key, or no powered ETERNA part present.
HSP_ERR_BADPARAM	26	Parameter error, incorrect address, bytes or pages value

Eterna Flash Emulator

The Eterna Serial Programmer Software uses the Eterna Flash Emulator, part of the Eterna integrated circuit, which supports a command protocol over IPCS that allows manipulation of the main pages on the Eterna Flash. The following describes the details of the Eterna Flash Emulator protocol.

If a device can is locked with the Hardware Lock Key (see *040-0109 Eterna Board Specific Parameter Configuration Guide* for details), it will prevent access to all internal memory including the flash. A locked device may be unlocked using UARTC0_RX, UARTC1_RX and UART_RX as described in the CLI (unlock command) section below.

Notes:

- UARTC0_RX, UARTC1_RX and UART_RX can always be used for unlocking a device, regardless of how the pin is configured for use in the product.

- On some operating systems, e.g. Linux, it may not be possible to load both D2XX and VCP drivers simultaneously. See the FTDI website for driver support.

A typical sequence in pseudo code for a simple example follows:

```
EnterFlashEmulator();      // assert FLASH_P_ENn,
                 // RESETn high 1ms, RESETn low,
                 // 10ms wait
BulkErase();     // SPI transaction to send Bulk Erase
command
WaitForRdy();    // SPI reads to poll STATUS until ready
ProgramFlash(    // SPI writes, in 16KB chunks max0x0,
                 // start of flash
fullImage,       // pointer to buffer with desired image
 sizeof(fullImage) );
WaitForRdy();
ReadFlash(       // SPI reads
0x0,
                 buf,
                 sizeof(fullImage));
Verify(          // Verify image
buf,
fullImage,
sizeof(fullImage));
```

IPCS (SPI Slave)

For parts that are not locked, the Eterna Flash emulator uses the IPCS bus in conjunction with the RESETn and FLASH_P_ENn signals. To enter the flash emulator, assert both RESETn and FLASH_P_ENn (both low), and then after 10ms, the host communicate with the emulator via the bus, at a maximum rate of 3.33 MHz.
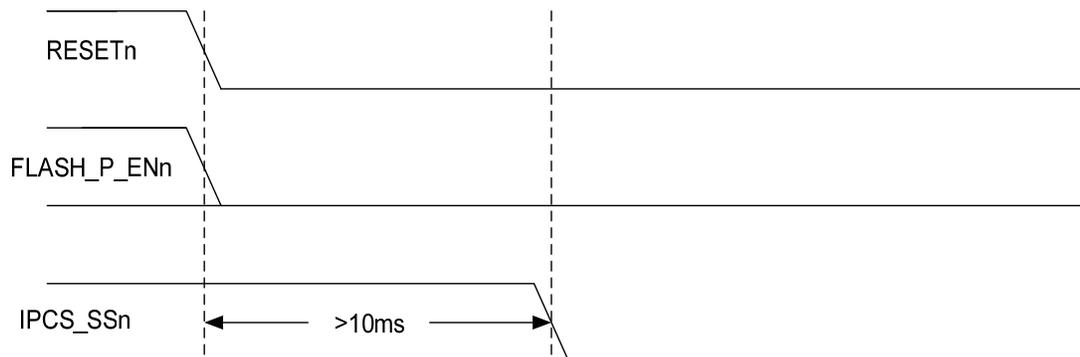
Operation is controlled by a protocol mastered externally. Commands starts with the falling edge of IPCS_SSn followed by an appropriate 8-bit instruction, listed below, followed by appropriate address and/or data. While IPCS_SSn is low, data from IPCS_MOSI is clocked from the master on the rising edge of IPCS_SCK, MSB first.

As appropriate, status and read data from the device is clocked out on IPCS_MISO on the falling edge of IPCS_SCK, for sampling by the master on the rising edge.
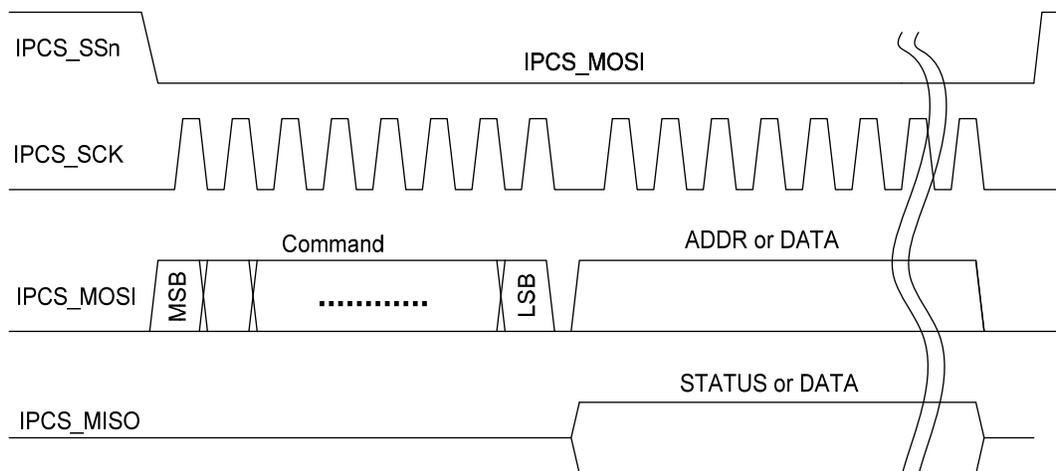
CLI (unlock command)

The Eterna flash emulator supports unlocking locked parts via either the UARTC0_RX, UARTC1_RX or UART_RX signals operating as UART receive pin with no flow control. The UART settings are 9600 baud, 8 bits, no parity, 1 stop bit. If a part is locked, the UART RX signals can be used to unlock the part using an 8 byte sequence consisting of a 4 byte cookie and the 4 byte key that was used to lock the part.

The format of the 8 byte unlock command is:

{0x12, 0x43, 0x87, 0x56, key[7:0], key[15:8], key[23:16], key[31:24]}.

If the correct sequence is received on the first try, the flash emulator will temporarily unlock the part, allowing the flash emulator commands listed above to be executed. If the correct sequence is not received, the flash emulator will stop executing, requiring a reset to restart. The flash emulator will listen for the command on the UARTC0_RX, UARTC1_RX, and UART_RX signals. The first UART to receive a byte will be used for the rest of the command.

If a persistent unlock of the part is desired, the normal flash emulator commands can be used to read out the 1st main flash page that contains the fuse table, modify it appropriately (i.e. remove the hw lock key entry), erase the 1st page and reprogram it with the modified data.

Command Summary

The emulator supports the following SPI commands:

- 0x9F Read ID
- 0xD7 Read Status
- 0x03 Continuous Read (Low Frequency)
- 0x50 Page Erase
- 0xC7 Bulk Erase
- 0x44 Buffer Write and Program

Command values other than those defined above can lead to undefined behavior.

The emulator supports the following CLI command:

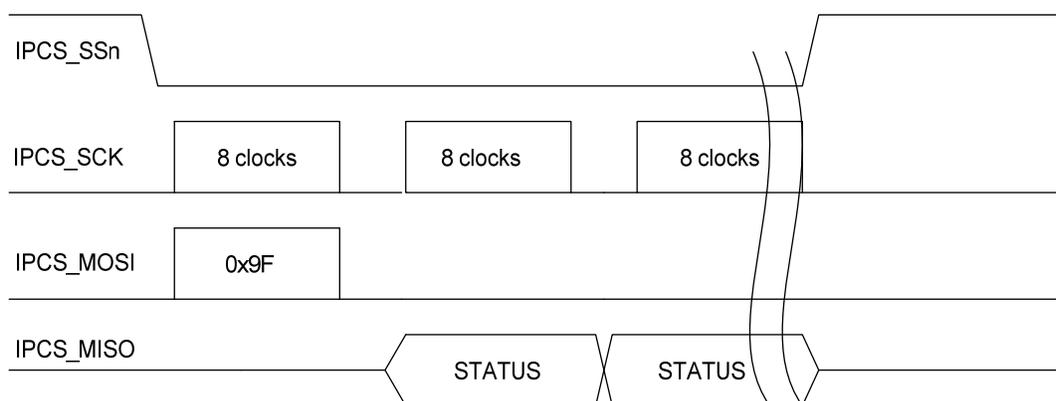- Unlock, see CLI (unlock command) section above.

Read ID

The Read ID command will cause the device to clock out the two byte value 0x1F24.

IPCS_SSn				
IPCS_SCK	8 clocks	8 clocks	8 clocks	
IPCS_MOSI	0x9F			
IPCS_MISO		0x1F	0x24	
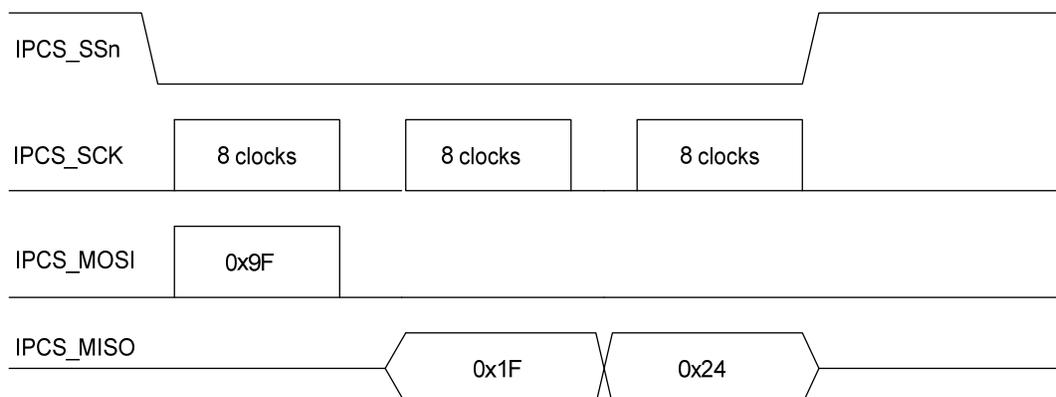
Read Status

The Read Status command will continually output a status byte.

IPCS_SSn				
IPCS_SCK	8 clocks	8 clocks	8 clocks	
IPCS_MOSI	0x9F			
IPCS_MISO		STATUS	STATUS	

The status byte has the following format:

STATUS Byte

MSB

RDY	0	0	1	1	1	X	X

The RDY bit is true (high) if the device is ready for the next operation, and the RDY bit is false (low) if the device has not completed the previous command, such as an erase. The 2 LSBs are reserved for future use, and their values are undefined.

Continuous Read

The Continuous Read command takes an additional 3 bytes of address after the command, and is followed by data from the relative address (more on address below).
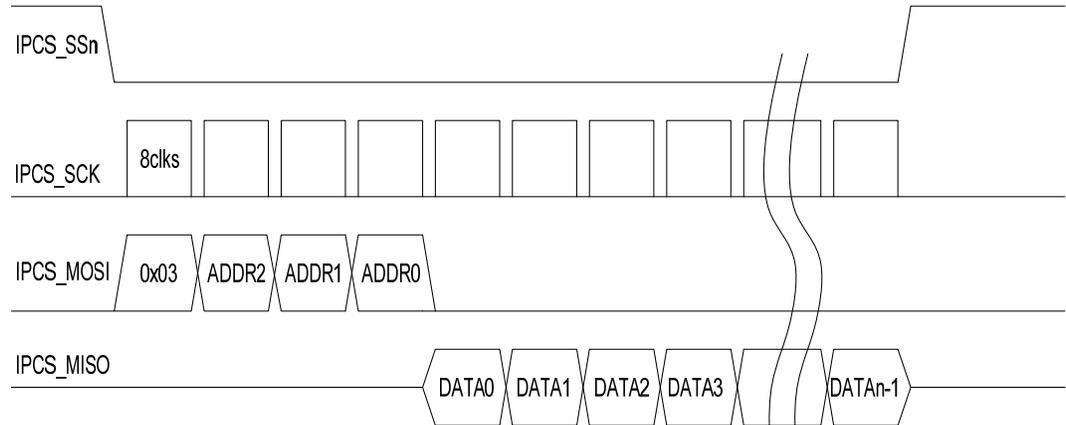

The 3 bytes of address is a byte address sent MSBs first. 0 represents the 1st byte address in Eterna Flash. The range of valid addresses is **0x000000** to **0x07FFFF**.

Address

MSB

ADDR2	ADDR1	ADDR0

The data is returned starting from the byte address provided, incrementing byte address for each additional byte read before the cycle is terminated.

Page Erase

The Page Erase command erases the 2048 byte page pointed to by the 3 byte address (the lower 11 bits are ignored).
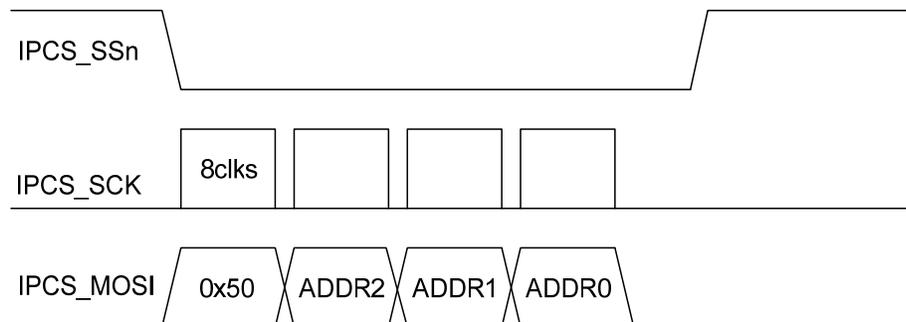
After issuing the Page Erase command, the host must insure that the erase operation is complete before proceeding by polling the Read Status Command, or waiting at least 40 ms.

Bulk Erase

The Bulk Erase command erases all 512 kBytes of Eterna flash. The Bulk Erase command must be followed by a 3 byte key: 0x94809A.

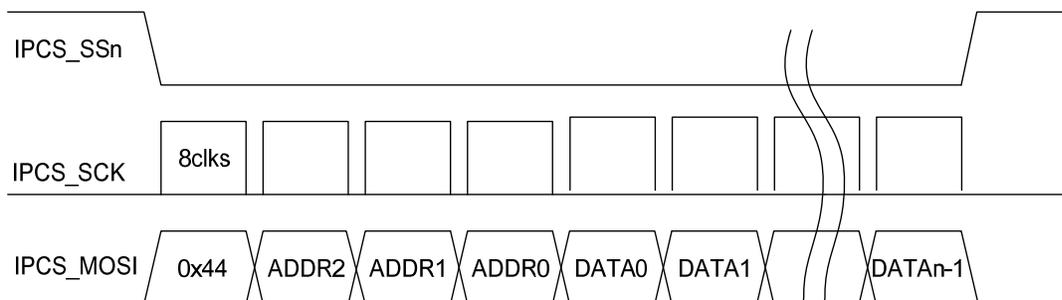| IPCS_SSn				
IPCS_SCK	8clks			
IPCS_MOSI	0xC7	0x94	0x80	0x9A
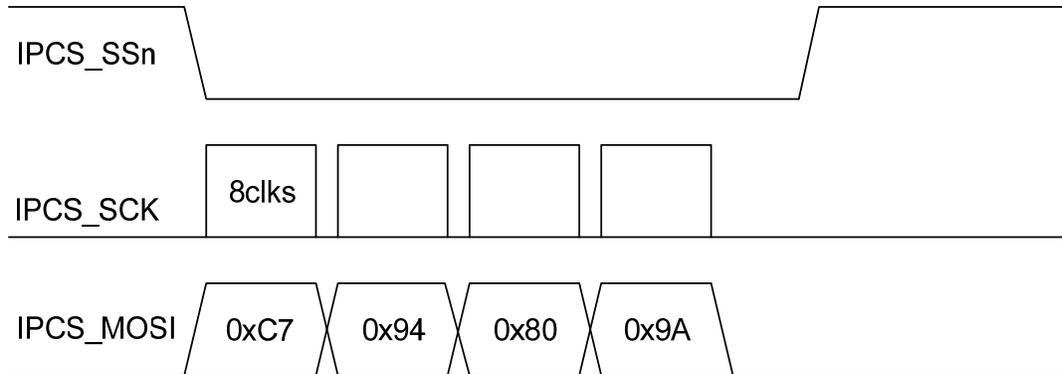
After issuing the Bulk Erase command, the host must insure that the erase operation is complete before proceeding by polling the Read Status Command, or waiting at least 40 ms.

Buffer Write

Buffer Program

Buffer Write and Program

The Buffer Write and Program command programs the passed data to the address provided. Writes must be integral 32-bits of data, and address must be 32-bit aligned. Maximum write length is 16KB.

IPCS_SSn							
IPCS_SCK	8clks						
IPCS_MOSI	0x44	ADDR2	ADDR1	ADDR0	DATA0	DATA1	DATAn-1

After issuing the Buffer Write and Program command, the host must insure that the erase operation is complete before preceding by polling the Read Status Command, or waiting at least 40 us times the number of words written.

DC9010 for Manufacturing

One or more DC9010s can be enabled to run in parallel on a single PC to create a manufacturing programming station.

Creating a Full Image

In order to accelerate programming, it is recommended to create a single image for manufacturing, as follows:

1) Erase the contents of Eterna's flash

2) Load all components of the flash's image as described in the **Error! Reference source not found.** section of this document

3) Configure any parameters via Eterna's API to the values needed for manufacture

4) Read Eterna's flash content to a 512 KB file (using the ESP –r command)

The binary image created from reading the flash content now represents a single image that can be loaded using the program with verify command with an offset of zero (refer to the example in the following section).

For network security purposes it is strongly recommended that the Hardware Lock Key Board Specific Parameter be enabled and set. It is essential that this value be tracked because once the device is locked, the only way to unlock the device is to present the correct Hardware Lock Key.

It is also recommended that manufacturers obtain a unique Manufacturer's ID from Linear Technology and assign the Manufacturer's ID along with a Board ID and Board Revision to each LTC5800 based design.

Programming the Device

With a single programmer, the device may be erased and programmed with the commands described in the Commands section on page 5:

```
C:\...\ESP\ESP –E
C:\...\ESP\ESP –P FILENAME 0
```

Where FILENAME is the full image created per the prior section. The ESP –E command is only required if the device is not blank or already erased.

Multiple Programmer Setup

When multiple programmers are connected, the ESP command requires a specific programmer to be identified as follows.

Location ID (USB 2.0 only)

To determine the Location ID for a specific DC9010 enter the following after connecting each DC9010:

```
C:\...\ESP\ESP –L
```

A list similar to the following will be presented (if USB 2.0 is used):

```
locID[0] = 0x2121, devString = Eterna Serial
Programmer A
locID[1] = 0x2122, devString = Eterna Serial
Programmer B
```

Use the B port's location ID with the –i option with any of the commands to direct the command to a specific ESP. For example to program with verify to the ESP device identified above one, enter:

```
C:\...\ESP\ESP –i 2122 –P FILENAME OFFSET
```
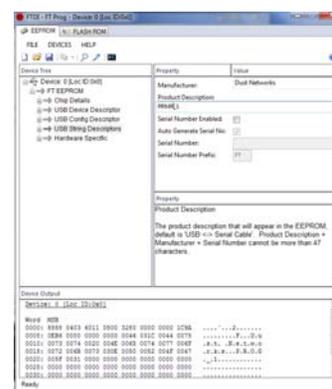
Device Name (USB 3.0 and ESP version 1.1.1 and higher)

With USB 3.0, the ESP command line option "–i LOCID" will not function as intended due to the FTDI drivers do not returning a valid Location ID. When multiple DC9010 are connected to a system via USB 3.0, the Device Name of each device must be written with a unique value. The following setup procedure details this programming step.

1. Program the multiple DC9010s one at a time with a devString (Device Name) unique to each DC9010 (i.e. "PROG1","PROG2", …):

   a) Download and install FT_PROG, an EEPROM Programming Utility available from FTDI website: http://www.ftdichip.com/Support/Utilities.htm.

   b) In the FT_PROG window, select "SCAN AND PARSE" in the "DEVICES" menu. Select "USB String Descriptors" and change the "Product Description" field to a unique value of your choice.
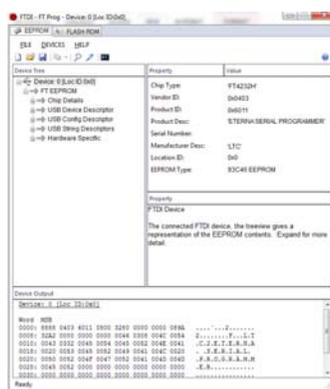
FT_PROG Window after "Scan and Parse"   Change "Product Description" field

   c) Once reprogrammed, the multiple devices will be listed as follows with the ESP –L command:

```
C:\>ESP –L
ESP, Eterna SPI Programmer, version 1.1.1-3
FTCSPI.dll = 2.1.2.2
locID[0] = 0x0, devString = PROG2 A
locID[1] = 0x0, devString = PROG1 A
locID[2] = 0x0, devString = PROG2 B
```

```
locID[3] = 0x0, devString = PROG1 B
```

2. Connect all devices to the computer and use the new Device Name to identify each device with ESP –n (include the " B" returned by esp –l). It is also recommended to include the –t option with the type number corresponding to the device (type 4 for the DC9010, see esp –h for the list of types).

```
C:\...\ESP\ESP –t 4 –n "PROG1 B" –P FILENAME OFFSET
```

DC9010 for Hardware Development

Crystal Load Capacitance Characterization

Eterna Integrated Crystal Load Capacitance and Board Support Parameters

Most modern crystal oscillators require two capacitors, often called load capacitors, to oscillate at the desired frequency. The LTC5800 family of products includes on-chip load capacitors for both the 32kHz and the 20 MHz crystal oscillators. The load capacitance for the 32kHz oscillator is fixed at a standard value of 12.5pF while the load capacitance for the 20 MHz crystal is adjustable via the "20 MHz load trim (0x13)" board support parameter as described in the Board Specific Configuration Guide. The adjustable load capacitors are used to center the resonating frequency of the 20 MHz oscillator. Because the 20 MHz oscillator is used as the radio frequency reference, the load capacitance must be trimmed to account for variations in printed-circuit board layout and dielectric stack-up. The DC9010 is used for characterizing the 20 MHz frequency distribution as described below. The characterization procedure is performed on a sample of PCBs during product development to allow correct generation of the "20 MHz load trim (0x13)"; it is not necessary to trim every part in production.

Load Capacitance Characterization with the DC9010

The DC9010 includes a calibrated timing reference signal that may be used to characterize the target board layout. The DC9010 is calibrated at the time of manufacture and should be calibrated annually to maintain accuracy. See the Crystal Timing Reference & Calibration Procedure section of this document for how to calibrate the DC9010 reference. The CLI command mxtal is used to characterize an LTC5800 based design, comparing the DC9010 reference against Eterna's crystal oscillator. The resulting characterized values should be used when creating the specific Board Specific Parameter binary image for each LTC5800 based design.

Characterization is performed by issuing the `mxtal` command over the target's **CLI port**, not via the ESP software. Use a terminal program, such as Tera Term, set to 9600 baud, 8-bit, no parity, 1 stop bit and no flow control, on the serial port identified in step 6) of the ESP setup. During the characterization of devices the actual value of the "20 MHz load trim (0x13)" board support parameter does not matter. After loading the product code image and a valid fuse table, enter the following command on the terminal:

```
> mxtal trim
```

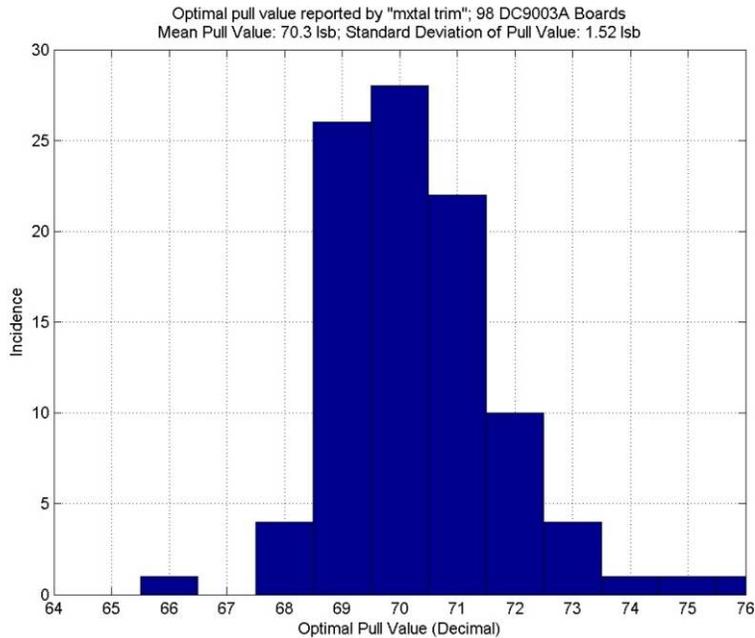After several seconds, Eterna will respond with approximately the following:

```
Optimal pullVal for this board=72 yields PPM
err=15/16
```

The above output indicates

    i)   that this particular unit performs optimally with a "20 MHz load trim (0x13)" (e.g. pullVal or pull value) equal to 72 decimal and

ii) were this value to be used in the fuse table the center frequency would be trimmed to within 15 / 16 's of a ppm (e.g. 0.9375 ppm) of desired.

Crystal characterization should be performed over a statistically meaningful number of targets to account for unit-to-unit variation in crystals and load. As an example of the complete characterization process, we will now find the correct "20 MHz load trim (0x13)" for the DC9003A hardware. We start with a sample size of ninety-eight DC9003A units and run mxtal trim on the motes. The crystal pull values are tabulated and the results are plotted in the histogram below:

Optimal pull value reported by "mxtal trim"; 98 DC9003A Boards
Mean Pull Value: 70.3 lsb; Standard Deviation of Pull Value: 1.52 lsb

The above histogram shows the pull value that centers the 20 MHz frequency for 98 DC9003A boards. The histogram includes the effects of variability due to crystal frequency tolerance, due to LTC5800 load-capacitance variation, and due to variation in PCB-trace capacitance. This distribution is indicative of the variation one should expect when the LTC5800 layout guidelines are followed and may be used to select a starting sample size for characterization. However, the variability in the above plot is a function of board manufacturing tolerances and layout so the number of characterization units required may be different than the above data suggest for other designs. In general, the number of units that should be characterized to attain 0.5 LSB (Least Significant Bit) accuracy may be estimated to be at least:
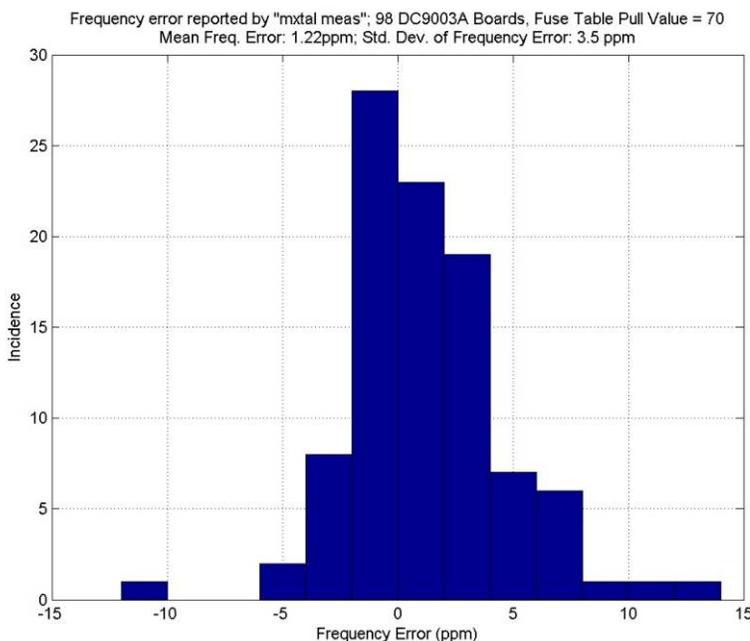
$$N = (\sigma / 0.5 \text{ LSB})^2$$

Where σ is the sample standard deviation of the pull values in LSB, and N is the total number of units needing to be characterized. From the histogram the standard deviation of frequency offset is 1.52 LSB which suggests, for this product, an absolute minimum of about 10 units need to be characterized and averaged to yield a pull value within 0.5 LSB of desired. Since the above equation is an approximation, it is a good idea to double the number of parts measured to offset errors in the approximation. So in this case, based upon the data from DC9003A implementation a conservative approach would be to use 20 parts to determine the "20 MHz load trim (0x13)".

We also note, from the above histogram, that the mean of the optimal pull value is 70.3 which is rounded to 70 because the fuse table field "20 MHz load trim (0x13)" is an integer. Now, to verify proper functionality, we generate a fuse table with "20 MHz load trim (0x13)" set to 70, and retest the units with the command:

```
> mxtal meas
pullVal used for measurement=70, PPM err=93/16
>
```

Where the "pullVal" indicates the correct "20 MHz load trim (0x13)" has been used and the PPM err of 93/16 shows the frequency error = 5.1825 PPM. Retesting the units with the correct fuse table ensures that the fuse table was generated correctly and shows the spread of the frequency errors using the value of the "20 MHz load trim (0x13)" to be loaded onto production boards.

Frequency error reported by "mxtal meas"; 98 DC9003A Boards, Fuse Table Pull Value = 70
Mean Freq. Error: 1.22ppm; Std. Dev. of Frequency Error: 3.5 ppm

The histogram above shows the results of the mxtal meas-reported 20 MHz frequency error at room temperature reported by mxtal trim for 98 DC9003A motes with 20 MHz load trim equal to 70. The histogram represents the trimmed unit-to-unit frequency variation. This distribution is indicative of the frequency variation one should expect of finished products in production. Note that the mean is not exactly zero which is principally because the pull value was rounded from 70.3 to 70. This distribution is well centered, has a single peak and bell-curve shaped. Thus a 20 MHz load trim equal to 70 is correct for this board.

Note: The crystal timing reference signal generator requires annual calibration from a GPS 1 Hz (PPS) clock source using the two SMA crystal timing reference maintenance interfaces and the on-board trim DIP switches. See Crystal Timing Reference & Calibration Procedure the section for details.

Note: This command may only be used when the mote's radio is not active, i.e. in the *slave* mode and prior to joining the network. After using this command, reboot the mote to continue normal operation.

DC9010 Specification

Functional Block Diagram

The following diagram depicts the internal circuitry of the DC9010. Detail data such as schematics, bill of material or circuit board layout are available online (Eterna Serial Programmer Board Files (Schematics...)).
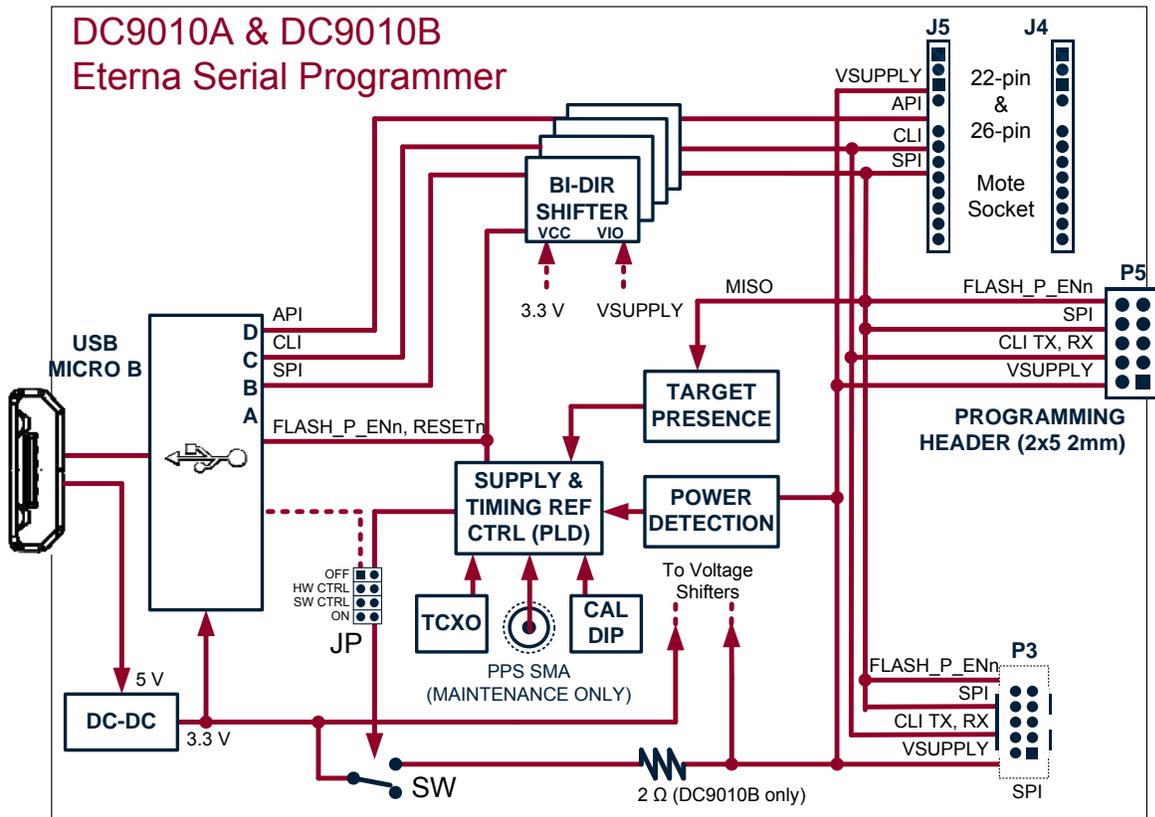
Features

Programming

The DC9010 allows programming of Eterna with the following interfaces:

- a 2x5 2mm header (externally accessible),
- a 0.050" programming header (internal),
- a 22/26-pin through-hole mote socket (internal).

Supply Voltage

The DC9010 accommodates either self-powered or unpowered targets. The full voltage range of self-powered devices is supported (2.1V to 3.76V). Unpowered targets are provided a supply voltage of approximately 3V.

At the start of programming (upon assertion of FLASH_P_ENn) or when Eterna's serial interface is accessed (UARTC0_RX), the DC9010 senses the target voltage and provide power to the target if no voltage is detected.

The automatic power selection mechanism may be overridden with a 2-mm board jumper if required (P1). The P1 jumper is only accessible when the enclosure cover is removed. P1 jumper settings are as follows.

P1 Jumper	Power Setting Description
None	Automatic target power selection
"ON"	Unpowered target (DC9010 provides VSUPPLY)
"OFF"	Self-powered target

Using an External Supply Voltage

The DC9010B features a header (P8, 3 pins, .100" pitch) located inside the enclosure near the "Target" connector. P8 is connected to the supply pins of all the target interfaces (J5, P5, P3…).

P8 Jumper	Description
1	GND
2	TARGET
3	VSUPPLY

In order to use an external power supply, set the P1 jumper in the "OFF" position and connect P8 pins 1 and 3 to the external power supply.

Serial Interfaces

The DC9010 PC interface consists of USB Serial Ports mapped to the Eterna's SPI, CLI and API interfaces.

ESP will operate over SPI for flash access and one of the CLI or API ports for device locking and unlocking.

Additionally any PC terminal application may open the available DC9010 USB Serial Ports, thus independently accessing Eterna's serial interfaces.

Visual Indicator

The DC9010 top LED flickers GREEN during programming and blinks GREEN at 4Hz rate when the crystal timing reference is provided (default).

Hardware Versions

DC9010A

The DC9010A is the initial version of the programmer. It includes all features listed above.

The DC9010A has been discontinued.

DC9010B

The DC9010B is an incremental enhancement to the DC9010A: it includes a simple limiter to the target current (2 ohm series resistance, see functional block diagram).

The DC9010B may be used with unpowered high capacitance targets such as energy harvesting boards with large storage capacitors.

Board Outline and Connector Pinout

Serial Programming Header (P5, right angle external 2mm, Molex 87833-1020)

Pin #	Signal	Direction	Pin #	Signal	Direction
1	IPCS_SSn	O	2	FLASH_P_ENn	O
3	IPCS _SCK	O	4	IPCS _MOSI	O
5	IPCS _MISO	I	6	RESETn	O
7	VSUPPLY	-	8	GND	-
9	UARTC0_TX	I	10	UARTC0_RX	O

22/26-Pin Mote Socket (J4 & J5, internal 2mm, Samtec SQT-113-01-L-S-005)

Pin #	J5 Signals	Direction	Pin #	J4 Signals	Direction
1	UARTC0_RX	O	1	LPTIMERn	I/O
2	UARTC0_TX	I	2	SLEEPn	I/O
3	GND	-	3	RESETn	O
4	VSUPPLY	-	4	IPCS_SSn	O
5	KEY	-	5	KEY	-
6	UART_RX	O	6	IPCS _MISO	I
7	UART_TX	I	7	IPCS _MOSI	O
8	UART_RX_RTSn	I	8	IPCS _SCK	O
9	UART_TX_RTSn	O	9	TCK	O
10	UART_RX_CTSn	O	10	TDO	I
11	UART_TX_CTSn	I	11	TDI	O
12	TIMEn	I/O	12	TMS	O
13	MODE	I/O	13	FLASH_P_ENn	O

Small Programming Header (P3, internal .050", Samtec FTSH-105-01-F-DV-K)

Pin #	Signal	Direction	Pin #	Signal	Direction
1	IPCS_SSn	O	2	FLASH_P_ENn	O
3	IPCS _SCK	O	4	IPCS _MOSI	O
5	IPCS _MISO	I	6	RESETn	O
7	VSUPPLY	-	8	GND	-
9	UARTC0_TX	I	10	UARTC0_RX	O

Reference Calibration Connectors (J2, J3, SMA, TE Connectivity 5-1814832-1)

Pin #	J2, PPS Input	Direction	Pin #	J3, Calibration Output	Direction
Signal	REF_IN	I	Signal	TEST_OUT	O
Shield	GND	-	Shield	GND	-

Crystal Timing Reference & Calibration Procedure

Crystal Timing Reference, 4PPS

The DC9010 features a crystal timing reference signal also referred to as "4PPS". The 4PPS signal used for device calibration using the standard programming interface header.

The 4PPS signal consists of four pulses per second with precise ¼-second edge placement; each pulse is typically one eighth of a second.

The DC9010's CPLD derives the crystal timing reference signal from its on-board temperature compensated crystal oscillator 2.5ppm TCXO. The DC9010's CPLD implements a multi-bit counter, allowing for less than 0.5ppm resolution adjustment of the crystal timing reference signal.

The REF_IN signal shall be disconnected when the 4PPS is used to calibrate an Eterna product.

4PPS Calibration

An on-board 8-bits switch is available for of calibration and controls the offset of the multi-bit counter. With 0x80 (switches closed or ON = 0) creating a 0ppm offset from the on-board TCXO.

TCXO offsets are calibrated using a 1 PPS signal from a GPS source, connected to J2 (REF_IN), J3 (TEST_OUT) is a copy of the 4PPS signal. During the calibration of the DC9010 the 1PPS reference resets the 4PPS signal output.

DC9010 – Crystal Timing Reference Calibration
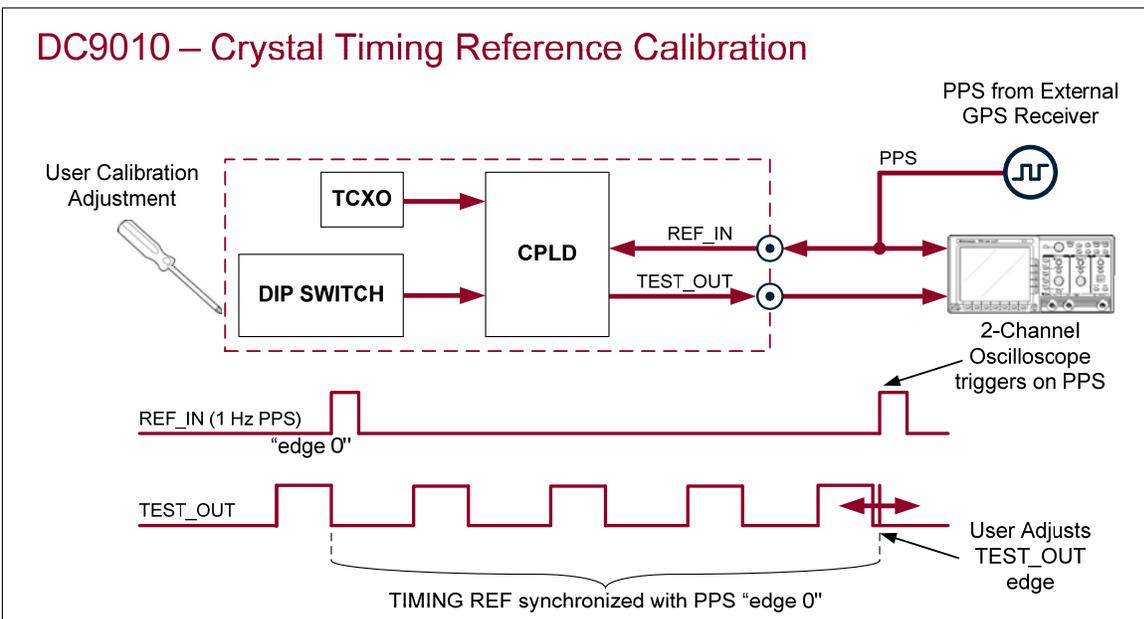
The purpose of the calibration is to set the offset 8-bit switch such that the 4PPS falling edge aligns with the PPS reference rising edge. With the calibration set correctly, and the scope zoomed into ~1us range (1us is 1ppm of 1s), the 4PPS falling edge can be observed slowly moving on the left of the reference rising edge, resetting periodically,

and moving in that same direction again. One LSB change in the offset switch value will result in 400ns change in the 4PPS edge placement.

To calibrate:

1. Pry open the DC9010 enclosure using the bottom slots and remove the top cover. Connect the DC9010 to a PC and allow for the temperature to stabilized to the environment for 30 min

2. Connect the DC9010 REF_IN SMA input to the channel 1 of the oscilloscope (set the scope input impedance to DC and high-Z, not 50 ohm) and then to a GPS 1PPS reference

3. Connect the DC9010 TEST_OUT SMA output to the channel 2 of the oscilloscope with a 50 ohm coax of the same length as the REF_IN connection to the oscilloscope

4. Start with a lower than nominal setting such as 0x70 (ON position is 0, OFF position is 1, pin 1 is LSB). Zoom in on PPS rising edge trigger (~1us/div) and observe the falling edge of the 4PPS signal. The falling edge of a valid 4PPS pulse should be on the left of the rising edge of the reference. A valid 4PPS pulse is a large pulse, rather than the reset narrow pulse (which is an artifact of the calibration).

5. Increment the setting value. The scope captures below show the 4PPS falling edge getting closer as the value is incremented. An LSB change corresponds to 100ns 4PPS period, or 0.4ppm. As the 4th pulse is used for alignment, 1 LSB results in 400ns shift over 1 second.
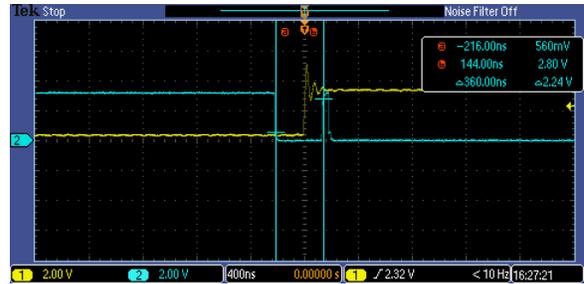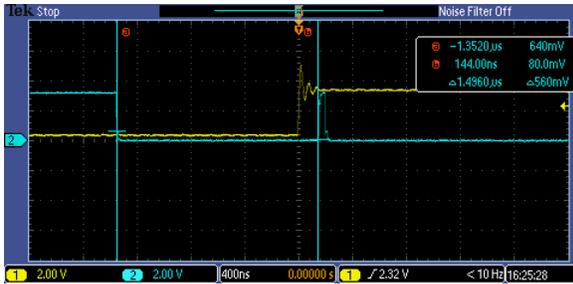
6. Calibration is complete once the falling edge of a large 4PPS pulse passes the REF_IN trigger edge and start merging into the reset pulse.
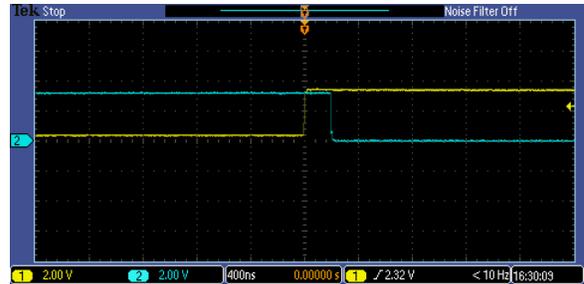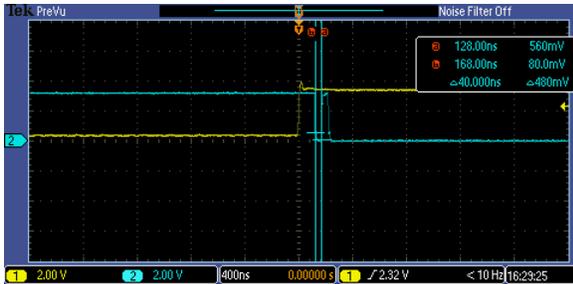
7. Disconnect the REF_IN signal from the DC9010 and verify that the 4PPS drift against the PPS reference is minimal.

Specifications

Specifications are at 25°C and 5V input supply voltage unless otherwise noted and may change without notice.

Electrical Parameters

Parameter	Min	Typ	Max	Units	Comments
Input supply voltage	4.5	5	5.5	V	From USB
Input supply current (w/out target)		120		mA	
Eterna programming current		15		mA	
Target supply voltage	2.4	3		V	
Maximum target load current			200	mA	
Timing reference calibration resolution		0.4		ppm	

Timing reference stability
    Over operating temperature
    Aging
		+/- 2.5
		+/- 1
				ppm
				ppm/year	

Environmental Parameters

Parameter	Min	Typ	Max	Units	Comments
Operating temperature	0		50	°C	
Storage temperature	-40		85	°C	
Humidity	10		90	% RH	Non-condensing
Altitude	0		3000	m	

Physical Parameters

Parameter	Min	Typ	Max	Units	Comments
Dimensions		24 x 100 x 52		mm^3	
Weight		80		g	
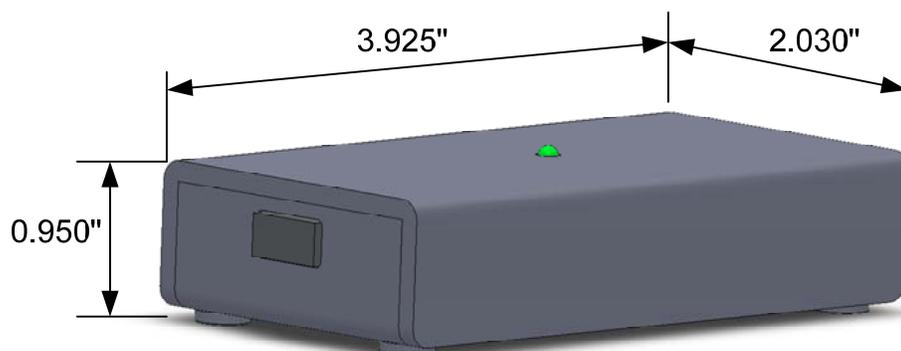
Addendum

Crystal changes on Printed Circuit Assembly (PCA) and Demo Circuits

Starting with Work Order ("WO") number D603679, PCA products and Demo Circuits containing the LTC5800 feature different crystals. The following image illustrates the differences on an LTP5901 PCA product.

New (WO# > D603679)   Former (WO# < D603679)
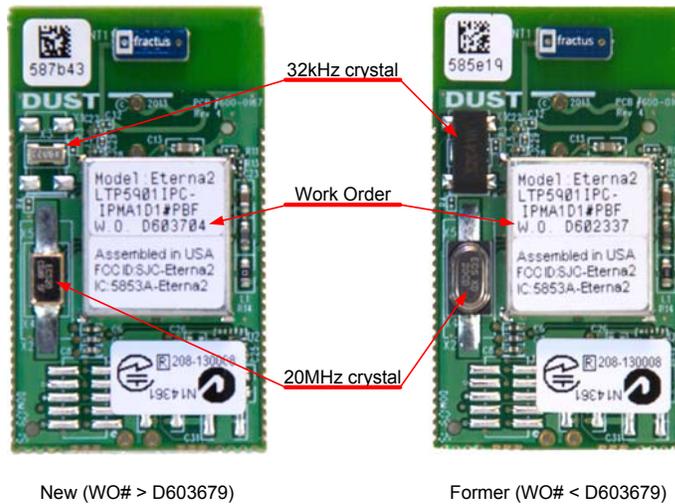
Board Specific Parameters ("Fuse Table") associated with the version of the 20 MHz crystal

If a PCA Product (LTP59xx) or a Demo Circuit (DCxxxx) is reprogrammed after it leaves the factory, attention should be given to Board Specific Parameters.

Linear Technology provides binary images of the Board Specific Parameters called "Fuse Tables" for each PCA Product and Demo Circuit. Fuse Table files are named after the product and an internal part number of the form 680-xxxx-yyyy, where xxxx is a sequential number specific to the PCA or Demo Circuit and yyyy is a version number.

PCA products and Demo Circuits[4] based on LTP5901 and LTP5902 with WO greater than D603679 shall include Board Specific Parameters with version -0003 or newer.

LTP5900 products with WO greater than D603679 shall include Board Specific Parameters with version -0005 or newer.

Other Demo Circuits such as the DC9000, DC9001, DC9003 and DC9007 are all built prior to WO D603679 and may be programmed with any of the respective mote or manager recommended Board Specific Parameters.

---

[4] Demo Circuits based on LTP5901 or LTP5902 include DC9018, DC9021, DC2126 and DC2274 series

Board Specific Parameters ("Fuse Table") for PCA with WO# D603679 or newer (greater)

PCA Products or Demo Circuits for WO number equal or greater than D603679 shall be programmed with the Fuse Tables shown below (or newer).

Current Products	Fuse Table
LTP5900IPC-WHMA	FT-LTP5900-WHMA-M13-9600-115K-680-0204-0005REV1.bin
LTP5901IPC-IPMA [5]	FT-LTP5901-IPMA-M4-115K-680-0237-0003REV1.bin
	FT-LTP5901-IPRA-M4-115K-680-0238-0003REV1.bin
	FT-LTP5901-IPRB-MEM-128K-M4-115K-680-0301-0003REV1.bin
LTP5901IPC-WHMA	FT-LTP5901-WHMA-M4-115K-680-0236-0003REV1.bin
LTP5902IPC-IPMA [5]	FT-LTP5902-IPMA-M4-115K-680-0241-0003REV1.bin
	FT-LTP5902-IPRA-M4-115K-680-0242-0003REV1.bin
	FT-LTP5902-IPRB-MEM-128K-M4-115K-680-0302-0003REV1.bin
LTP5902IPC-WHMA	FT-LTP5902-WHMA-M4-115K-680-0240-0003REV1.bin

Legacy Products [6]	Fuse Table (if WO is D603679 or greater)
LTP5900IPC-WHMA***	FT-LTP5900-WHMA-M13-9600-115K-680-0204-0005REV1.bin
LTP5901IPC-IPMA***	FT-LTP5901-IPMA-M4-115K-680-0237-0003REV1.bin
LTP5901IPC-IPRA***	FT-LTP5901-IPRA-M4-115K-680-0238-0003REV1.bin
LTP5901IPC-IPRB***	FT-LTP5901-IPRB-MEM-128K-M4-115K-680-0301-0003REV1.bin
LTP5901IPC-IPRC***	FT-LTP5901-IPRC-MEM-128K-M4-115K-680-0305-0003REV1.bin
LTP5901IPC-WHMA***	FT-LTP5901-WHMA-M4-115K-680-0236-0003REV1.bin
LTP5902IPC-IPMA***	FT-LTP5902-IPMA-M4-115K-680-0241-0003REV1.bin
LTP5902IPC-IPRA***	FT-LTP5902-IPRA-M4-115K-680-0242-0003REV1.bin
LTP5902IPC-IPRB***	FT-LTP5902-IPRB-MEM-128K-M4-115K-680-0302-0003REV1.bin
LTP5902IPC-IPRC***	FT-LTP5902-IPRC-MEM-128K-M4-115K-680-0306-0003REV1.bin
LTP5902IPC-WHMA***	FT-LTP5902-WHMA-M4-115K-680-0240-0003REV1.bin

Demo Circuits	Fuse Table (if WO is D603679 or greater)
DC2126A	FT-LTP5901-IPMA-M4-115K-680-0237-0003REV1.bin

---

[5] LTP5901IPC-IPMA and LTP5902IPC-IPMA products may be programmed as a mote, 32-mote manager (which does not require external memory) or 100-mote manager (which requires external memory). The programmed software image shall include the Main Executable corresponding to the mote or manager function and the Fuse Table specified in this table. The Fuse Tables are respectively named after -IPMA, -IPRA and -IPRB, for mote, manager and manager with memory.

[6] LTP5900, LTP5901 and LTP5902 based products were originally shipped pre-programmed at the factory; these product versions are still available but referred to as Legacy Products. In this column, "***" represents a three digit alphanumeric field signifying the pre-programmed software revision (e.g. "LTP5901IPC-IPMA**1D1**"). If a Legacy Product is re-programmed and its WO number is greater than D603679, the Fuse Table specified in this table or newer is required.

Demo Circuits	Fuse Table (if WO is D603679 or greater)
DC2274A-A	FT-DC2274A-MEM-128K-M4-115K-680-0383-0003REV1.bin
DC9018A-B	FT-DC9018A-MOTE-M4-115K-680-0379-0003REV1.bin
DC9018A-C	FT-DC9018A-MOTE-M4-115K-680-0379-0003REV1.bin
DC9018B-B	FT-DC9018B-MOTE-M4-115K-680-0380-0003REV1.bin
DC9018B-C	FT-DC9018B-MOTE-M4-115K-680-0380-0003REV1.bin
DC9020A	FT-DC9020A-MANAGER-MEM-128K-M4-115K-680-0381-0003REV1.bin
DC9020B	FT-DC9020B-MANAGER-MEM-128K-M4-115K-680-0382-0003REV1.bin

Board Specific Parameters ("Fuse Table") for PCA with WO# older than D603679

The following details the latest compatible Fuse Table for older PCA Products or Demo Circuits with WO number smaller than D603679.

Legacy Products [6]	Fuse Table (if WO is smaller than D603679)
LTP5900IPC-WHMA***	FT-LTP5900-WHMA-M13-9600-115K-680-0204-0004REV1.bin
LTP5901IPC-IPMA***	FT-LTP5901-IPMA-M4-115K-680-0237-0002REV1.bin
LTP5901IPC-IPRA***	FT-LTP5901-IPRA-M4-115K-680-0238-0002REV1.bin
LTP5901IPC-IPRB***	FT-LTP5901-IPRB-MEM-128K-M4-115K-680-0301-0002REV1.bin
LTP5901IPC-IPRC***	FT-LTP5901-IPRC-MEM-128K-M4-115K-680-0305-0002REV1.bin
LTP5901IPC-WHMA***	FT-LTP5901-WHMA-M4-115K-680-0236-0002REV1.bin
LTP5902IPC-IPMA***	FT-LTP5902-IPMA-M4-115K-680-0241-0002REV1.bin
LTP5902IPC-IPRA***	FT-LTP5902-IPRA-M4-115K-680-0242-0002REV1.bin
LTP5902IPC-IPRB***	FT-LTP5902-IPRB-MEM-128K-M4-115K-680-0302-0002REV1.bin
LTP5902IPC-IPRC***	FT-LTP5902-IPRC-MEM-128K-M4-115K-680-0306-0002REV1.bin
LTP5902IPC-WHMA***	FT-LTP5902-WHMA-M4-115K-680-0240-0002REV1.bin

Older Demo Circuits	Fuse Table (if WO is smaller than D603679)
DC2126A	FT-LTP5901-IPMA-M4-115K-680-0237-0002REV1.bin
DC2274A-A	FT-DC2274A-MEM-128K-M4-115K-680-0383-0002REV1.bin
DC9001A	FT-DC9003A-M4-115K-680-0222-0004REV1.bin
DC9001B	FT-DC9011A-M4-115K-MEM256k-680-0258-0003REV1.bin
DC9003A-B	FT-DC9003A-M4-115K-680-0222-0004REV1.bin
DC9003A-C	FT-DC9003A-M4-115K-680-0222-0004REV1.bin
DC9018A-B	FT-DC9018A-MOTE-M4-115K-680-0379-0002REV1.bin
DC9018A-C	FT-DC9018A-MOTE-M4-115K-680-0379-0002REV1.bin
DC9018B-B	FT-DC9018B-MOTE-M4-115K-680-0380-0002REV1.bin
DC9018B-C	FT-DC9018B-MOTE-M4-115K-680-0380-0002REV1.bin
DC9020A	FT-DC9020A-MANAGER-MEM-128K-M4-115K-680-0381-0002REV1.bin
DC9020B	FT-DC9020B-MANAGER-MEM-128K-M4-115K-680-0382-0002REV1.bin

Document Status	Product Status	Definition
Advanced Information	Planned or under development	This document contains the design specifications for product development. Dust Networks reserves the right to change specifications in any manner without notice.
Preliminary	Engineering samples and pre-production prototypes	This document contains preliminary data; supplementary data will be published at a later date. Dust Networks reserves the right to make changes at any time without notice in order to improve design and supply the best possible product. The product is not fully qualified at this point.
No identification noted	Full Production	This document contains the final specifications. Dust Networks reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
Obsolete	Not in production	This document contains specifications for a product that has been discontinued by Dust Networks. The document is printed for reference information only.

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Programmers - Processor Based *category:*

Click to view products by Analog Devices *manufacturer:*

Other Similar products are found below :

5.05.10 TPG100004 X2S-FP-X APM32PROG ECC111429EU UMFTPD2A MIKROPROG FOR 8051 JTAG HS2 PROGRAMMING CABLE JTAG-SMT2-NC SM PROGRAMMING MODULE MIKROPROG FOR AVR MIKROPROG FOR PIC,DSPIC AND PIC32 MIKROPROG FOR STM32 MIKROPROG FOR TIVA ZL20PRG AVR-ISP500-TINY GP-ARM DFR0116 PGM-08702 ACNPROG PGM-07834 XUP USB-JTAG PROGRAMMING CABLE CK-USB-04A REVELPROG-IS FLASHPRO-2000-STD GANGPRO-ARM-1V CODEGRIP FOR ARM CODEGRIP FOR STM32 CODEGRIP FOR TIVA FLASHPRO-430-CC FLASHPRO-430-LJ FLASHPRO-430-STD FLASHPRO-ARM(X2S) FLASHPRO-ARM-1V(XS) GANGPRO-430(XS) GANGPRO-ARM-1V(XS) AVR-ISP500-ISO AVR-JTAG-USB-A 462 MIKROPROG FOR MSP432 JTAG USB CABLE PROGRAMMER FOR CMT 2548 46 VA800A-PROG CY8CKIT-005 FlashPro-CC-STD FLASHPRO-X REP430F USB-MSP430-FPA-LJ JTAG-SMT3-NC PROGRAMMING MODULE