CML Microcircuits

COMMUNICATION SEMICONDUCTORS

PE0001

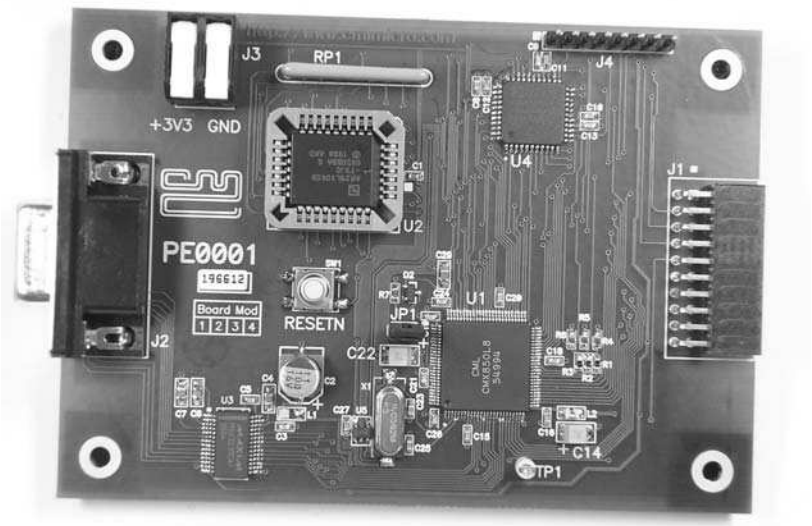
PE0001 Evaluation Kit Interface Card

UM0001/5 May 2007

Provisional Issue

Features

- **Global interface for new generation IC evaluation kits**
- **Target IC C-BUS read, write and general reset operations**
- **8051 µController based operation**
- **Mating interface for wide range of target evkit boards**
- **Power supply from target evkit or standalone**
- **PC GUI, firmware and hardware provided**
- **Function Image™ handling for *FirmASIC®*-based projects**
- **4Mbit of re-programmable Flash program memory**
- **PC control/communications via RS232**



1 Brief Description

The PE0001 interface card is a global interface system for use with evaluation kits for CML's new generation ICs, including *FirmASIC* ® –based products. This greatly simplifies the approach to the evaluation and design-in process.

Supplied with a PC GUI, and in the form of a populated PCB, the PE0001 provides a graphical method of addressing all of the evkit's target IC's on-chip registers via the C-BUS interface.

Based around the operation of CML's CMX850 Communications Controller IC and external Flash program memory, the information generated by the GUI is formatted, timed and delivered to the target IC via its evkit's C-BUS serial interface. The supplied control software can be used to perform C-BUS read, write and general reset operations, run scripting functions and, in the case of *FirmASIC* ® IC evaluations, load a Function Image™. It can also load a Function Image™ into EEPROM on the PE{target} card.

Communications with the evaluating PC GUI are via a serial port using RS232. The Flash memory can be re-programmed in circuit using CML's Flash programming software. Power to the PE0001 can be obtained from the target evkit or standalone from a bench supply.

CONTENTS

It is always recommended that you check for the latest product datasheet version from the Datasheets page of the CML website: [www.cmlmicro.com].

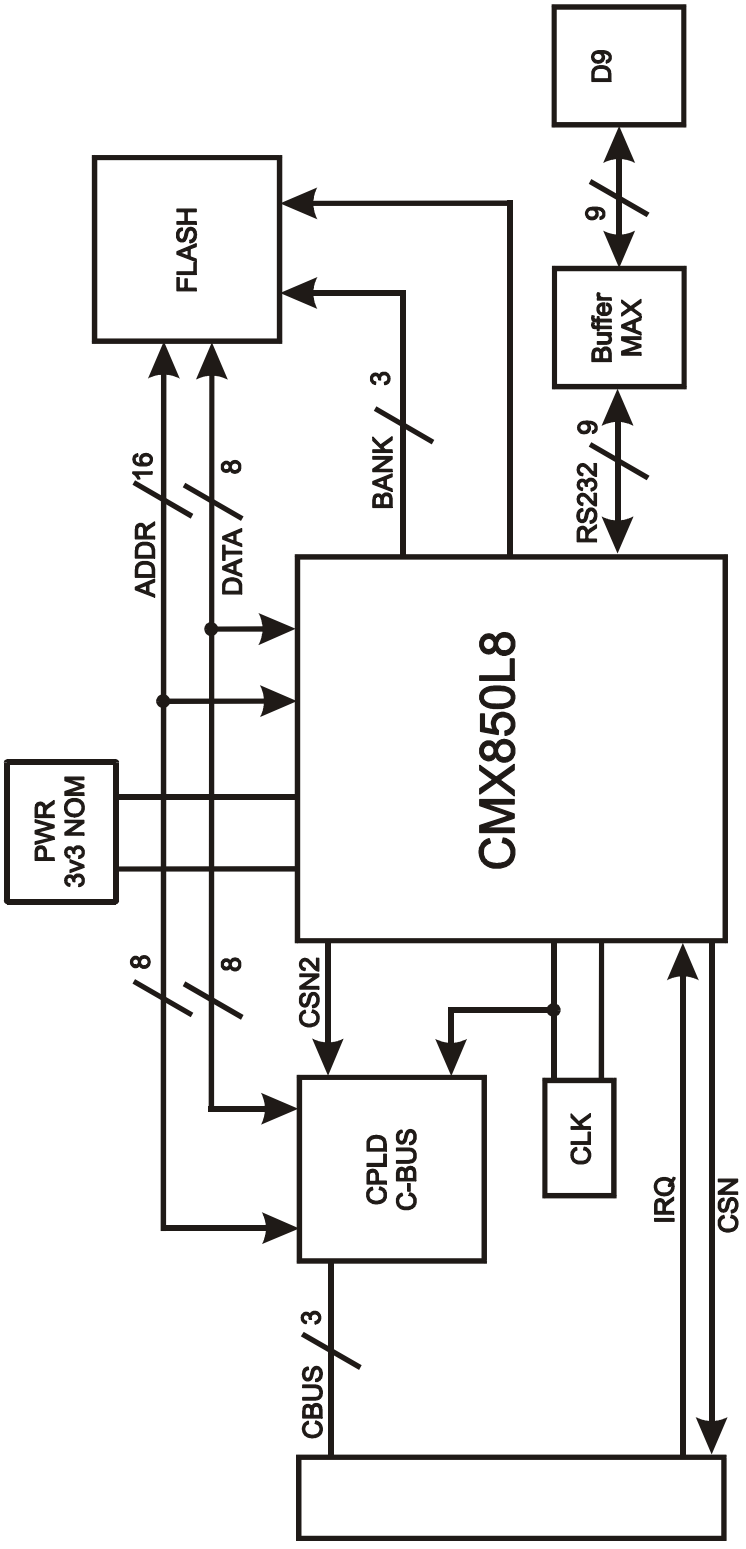


Figure 1 Block Diagram

2. Preliminary Information

2.1 Laboratory Equipment

The following laboratory equipment is needed to use this evaluation kit:

2.1.1 Power Supply

The PE0001 Interface Card is designed to connect to a PE{target} Card, which provides the power for the PE0001 Interface Card.

2.1.2 PC

With the following requirements:

- One of the following operating systems installed - Windows *98, ME, 2000sp4 or XP.*
- Equipped with a serial communications port, capable of supporting a data rate of 19,200baud.
- Minimum screen resolution 800 x 600. Recommended resolution 1024 x 768.

2.1.3 RS232, 9 way DTE to DCE cable

2.2 Precautions

Like most evaluation kits, this product is designed for use in office and laboratory environments. The following practices will help ensure its proper operation.

2.2.1 Static Protection

This product uses low power CMOS circuits which can be damaged by electrostatic discharge. Partially damaged circuits can function erroneously, leading to misleading results. Observe ESD precautions at all times when handling this product.

2.2.2 Contents - Unpacking

Please ensure that you have received all of the items on the separate information sheet (EK0001) and notify CML within 7 working days if the delivery is incomplete.

3. Quick Start

This section provides instructions for users who wish to experiment immediately with the evaluation kit. A fuller description of the kit and its use appears later in this document.

3.1 Setting-Up

- Copy the PC application, 'ES0001xx.EXE', which is downloaded from the CML website following registration, to the hard drive of your host PC.
- Connect the PE0001 Interface Card to the PE{target} Card, via the right angle connector, J1.
- Connect a dc supply to the PE{target} Card and set to the voltage level specified in the relevant PE{target} Card User Manual. The PE0001 Interface Card is powered from the PE{target} Card.
- Attach a 9-way RS232 cable between connector J2 of the PE0001 Interface Card and the serial port of the PC.

3.2 Adjustments

None.

3.3 Operation

- Turn on the power supply.
- Run the PC executable, 'ES0001xx.EXE'.
- Select the relevant serial port number from the drop-down box.
- Press the 'OK' button.
- Press the reset switch (SW1) on the PE0001 board when requested.
- Press the 'OK' button.

If communications between the PC and PE0001 Interface Card are not established an error message will be displayed. Otherwise the kit is now ready to use.

4. Signal Lists

CONNECTOR PINOUT				
Connector Ref.	**Connector Pin No.**	**Signal Name**	**Signal Type**	**Description**
J1	1	IO0	BI	Spare μController I/O
	2	CSN	O/P	C-BUS Chip Select
	3	IO1	BI	Spare μController I/O
	4	CDATA	O/P	C-BUS Command Data
	5	IO2	BI	Spare μController I/O
	6	SCLK	O/P	C-BUS Serial Clock
	7	IO3	BI	Spare μController I/O
	8	RDATA	I/P	C-BUS Reply Data
	9	IO4	BI	Spare μController I/O
	10	IRQN	I/P	C-BUS interrupt request, connected to INT1 of CMX850.
	11,12	GNDD	Power	Digital ground
	13	BOOTEN1	O/P	CMX70xx Hardware boot control
	14	BOOTEN2	O/P	CMX70xx Hardware boot control
	15	RS232/~C-BUS	O/P	CMX70xx Hardware boot control
	16	IO5	BI	Spare μController I/O
	17	IO6	BI	Spare μController I/O
	18	IO7	BI	Spare μController I/O
	19, 20	VDDD	Power	Digital VDD
J2	1	DCD	O/P	9-pin D Type connector – PC DCD
	2	TXD	O/P	9-pin D Type connector – PC TXD
	3	RXD	I/P	9-pin D Type connector – PC RXD
	4	DTR	I/P	9-pin D Type connector – PC DTR
	5	GNDD	Power	9-pin D Type connector – PC GND
	6	DSR	O/P	9-pin D Type connector - PC DSR
	7	RTS	I/P	9-pin D Type connector – PC RTS
	8	CTS	O/P	9-pin D Type connector - PC CTS
	9	RI	O/P	9-pin D Type connector – PC RI

CONNECTOR PINOUT				
Connector Ref.	**Connector Pin No.**	**Signal Name**	**Signal Type**	**Description**
J3	1	GNDD	Power	
	2	VDDD	Power	
J4	1	VDDD	Power	Digital VDD
	2	TDO	O/P	J-TAG, data output.
	3	TDI	I/P	J-TAG, data input
	4, 5	n/c		
	6	TMS	I/P	J-TAG, mode select
	7	GNDD	Power	Digital ground
	8	TCLK	I/P	J-TAG, serial clock

TEST POINTS		
Test Point Ref.	**Default Measurement**	**Description**
TP1	VDDD	INT0, highest priority input to CMX850

JUMPERS			
Link Ref.	**Positions**	**Default Position**	**Description**
JP1	O/C, S/C	O/C	Select μController boot ROM when shorted

SWITCHES		
Switch Ref.	**Default Position**	**Description**
SW1	O/C	Push to reset the CMX850 μController

Notes: I/P = Input
O/P = Output
BI = Bidirectional
O/C = Open circuit
S/C = Short circuit

5. Circuit Schematics and Board Layouts

For clarity, circuit schematics are available as a separate high resolution pdf file. This can be found on the CML website.

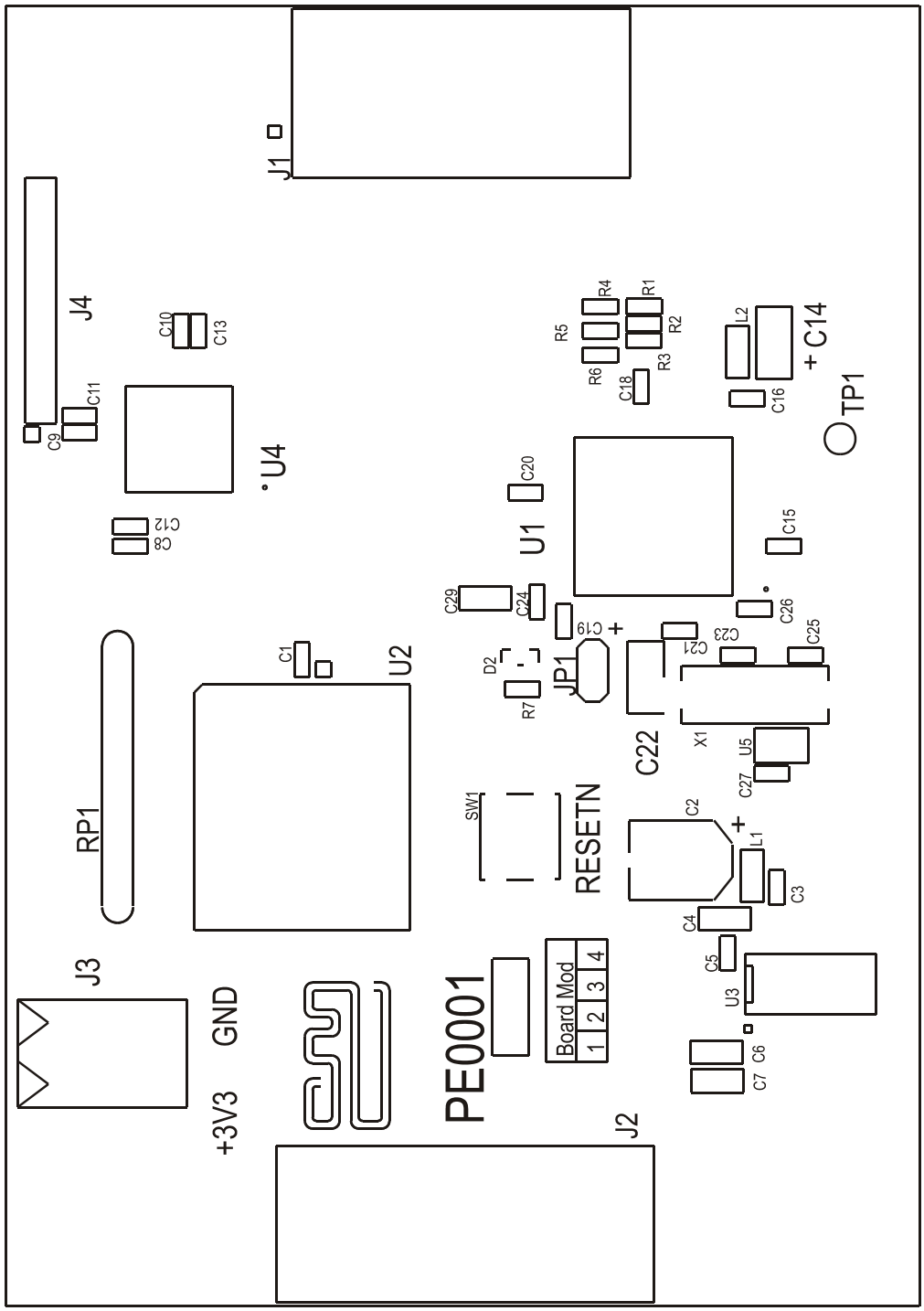


Figure 2 PE0001 Interface Card layout

6. Detailed Description

6.1 Hardware Description

The PE0001 Interface Card is based around the CMX850, 8051 µcontroller. There are 8kBytes of XRAM available on-chip, and 4Mbit of external flash for program memory. The link to a host PC is a standard RS232 serial port. The flash memory can be reprogrammed via the RS232 serial port using the CML 'ESFLASH' tool.

A C-BUS output, J1, is provided for control of the CMX{target} on the selected PE{target} Card. The C-BUS signals Command Data, Reply Data and Serial Clock are interfaced to the µcontroller with some programmable logic implemented in a Xilinx CPLD. This can be reprogrammed via a J-TAG interface on J4.

6.1.1 CPLD C-BUS Implementation

When provided with an appropriate clock signal, the CPLD will perform the translation to/from the data bus of the µcontroller and the C-BUS of the PE{target} Card.

All writes and reads to the CPLD occur via the data bus and must consist of:

1. The assertion of the chip select two and write or read strobes
2. The setting of the 4-bit address bus and 8-bit data bus values
3. The de-assertion of the chip select two and write or read strobes

The C-BUS read/write register is memory mapped and located at address $0x0 in the CSN2 memory block. The MOVX instruction is used for the read and write action, but the MEMCON register of the CMX850 core must be set to direct the MOVX instruction appropriately. See the CMX850 data sheet for more information.

Resetting the CPLD

Before using the CPLD to access the C-BUS it is advisable to reset it first by:

1. Applying the clock signal to the CPLD
2. Creating a positive edge on RESETN
3. Waiting for one C-BUS cycle, i.e. 8 cycles of the CPLD clock

Writing to a C-BUS register

1. The CMX850 should assert CSN, i.e. set port 1, bit 2 (pin 20) low.
2. Perform a write on parallel interface with:
   - The 4-bit address bus = $0
   - The 8-bit data bus = the address of the C-BUS register to be written.
3. The CPLD will then latch the 8-bit C-BUS register address and begin to send it serially down the C-BUS line CDATA.
4. The microprocessor should then wait for the CBUSACTIVE line to de-assert. The CBUSACTIVE line is connected to port 1, bit 3 (pin 21) of the CMX850 µcontroller.
5. For an 8-bit C-BUS register, repeat steps 2-4 with the data bus = the value that is to be written to the register. For a 16-bit C-BUS register, repeat steps 2-4 twice, first with the data bus = the 8 MSBs of the value and then with the data bus = the 8 LSBs of the value to be written.
6. The CMX850 should then de-assert CSN, i.e. set port 1, bit 2 (pin 20) high.

Reading from a C-BUS register

1. The CMX850 should assert CSN, i.e. set port 1, bit 2 (pin 20) low.
2. Perform a write on the parallel interface with:
   - The 4-bit address bus = $0
   - The 8-bit data bus = the address of the C-BUS register to be read.
3. The CPLD will then latch the 8-bit C-BUS register address and begin to send it serially down the C-BUS line CDATA.
4. Wait for the CBUSACTIVE signal to de-assert
5. Repeat steps 2-4 but with the data bus = 0x00, this will cause the CPLD to clock in the register value from the C-BUS data line RDATA.
6. The value of C-BUS register can then be read by performing a read on the parallel interface with the address = $0.
7. For a 16-bit C-BUS register, only the 8 MSBs of the C-BUS register will have been read so steps 2-4 will need to be repeated in order to read the 8 LSBs of the C-BUS register.
8. The CMX850 should then de-assert CSN, i.e. set port 1, bit 2 (pin 20) high.

6.2 Adjustments and Controls

None.

6.3 Embedded Software Description

On power up the µcontroller will monitor the serial communications port until the port becomes active, i.e. the RTS and DTR signals are asserted. When the µcontroller detects the asserted signals, the embedded software identifies itself to the host PC.

The µcontroller software provides the interface between the RS232 serial communications port of the host PC and C-BUS port of the PE{target} Card.

6.3.1 Programming the FLASH ROM

The PE0001 board supports in-situ programming of the onboard FLASH memory using the serial interface and internal BootROM. It requires the use of programming application ‘ESFLASH_xx.EXE’ where ‘xx’ is version ‘14’ or higher.

To enter FLASH programming close jumper JP1 and press the ‘RESETN’ switch, SW1. Connect the serial interface (J2) to a PC able to run the FLASH programming application.

Start the FLASH programming application and select a “Target type selection” of “PE0001 Evaluation board”. Click “Load File” and select a “fat stub download” HEX file which matches the flash memory type fitted to the attached PE0001 evaluation board. (The board is normally fitted with AM29LV040B 70ns FLASH memory, therefore use ‘AM29LV040_0001.hex’). Click “Load File” and select the required FLASH data download file and select ‘FLASH bank 0 download’. The Keil compiler suite produces a single file for code banked projects with the file extension “.Hex”.

Select the PC communications resource which is connected to the PE0001 evaluation board. Press the “Program FLASH” button to start the download sequence to the PE0001 evaluation board. The dialog controls will be greyed and a progress counter will start. If the program prompts with the message “Automatically include ‘thin’ stub code bank in 1 ?” then click the No button.

The first pass of the progress counter is the download of the 'fat' stub to control access to the FLASH memory. The second pass of the progress bar is the download of the FLASH program data. Once both passes of the progress bar are complete the PE0001 FLASH application will inform the user that the download is complete. Remove jumper JP1 and reset the board for normal operation.

6.4 Software Description

A single Windows executable 'ES0001xx.EXE' supports a range of PE{target} Cards. Visit the CML website for the latest information.

When the ES0001xx.EXE application is run, a dialog box is displayed, which is shown in figure 3. The drop-down list enables the required serial port to be selected. The application will automatically determine the enumeration of all COM ports installed on the host PC and display the first valid COM port number within the drop-down list. This COM port number does not necessarily reference the port to which the PE0001 is actually connected. Once the selection is made, clicking the OK button will display a message box, requesting that the reset switch (SW1) on the PE0001 board should be pressed. Once this is done and the OK button is clicked, the selected serial port is configured with the settings shown below and an attempt is made to establish communication between the host PC and PE0001 Interface Card.

Bits per second	19200
Data bits	8
Parity	None
Stop bits	1
Flow control	CTS/RTS

Figure 3 Serial Port Select dialog box

If this is successful the software will assert RTS and DTR signals and monitor the serial port for identification from the embedded software. If communication cannot be established then a message box will be displayed to indicate the failure. Clicking on the 'OK' button will quit the ES0001xx application.

A tabbed dialog box is then displayed, which is the main application dialog. There are four sheets within the tabbed dialog box structure and these are described in the proceeding sections.

6.4.1 The C-BUS Control Tab

This tab provides basic C-BUS read, write and general reset functions. Each character entered into the Address and Data edit boxes is checked to ensure that it is a valid hexadecimal value. The radio buttons select an 8-bit or 16-bit read/write operation. The lengths of the entered values are validated, 2 characters (1 byte) for read or write register addresses and 2 or 4 characters (1 or 2 bytes) for the register write data. The general reset function writes 01_H to the CMX{target}.

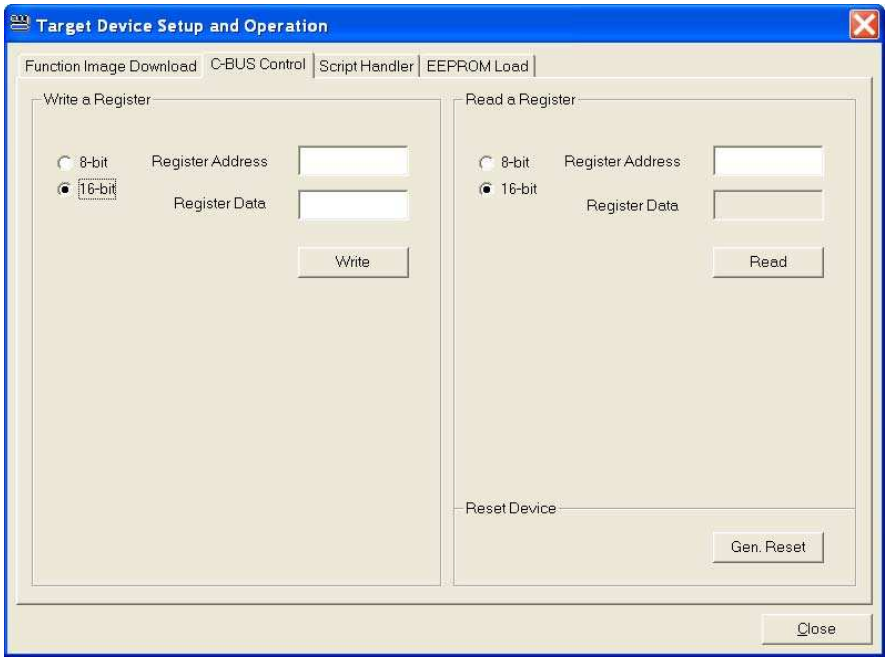


Figure 4 C-BUS Control Tab

6.4.2 The CMX70xx Function Image Download Tab

This tab is used to control the way in which a Function Image is downloaded into the target device. The Function Image can be downloaded directly from EEPROM, which is located on the PE{target} Card into the CMX70xx or it can be downloaded from a file, located on the host PC, via the PE0001 Interface Card.

To load the Function Image™ from EEPROM, either select the checksums and activation code values from the drop-down lists or type them into the relevant edit boxes. Click on the 'Download' button. Note that the lists in the drop-down boxes contain the last two checksums and activation codes used.

Checksum values and activation codes are obtained from the CML website following registration. Any values typed in are checked to ensure that they contain only hexadecimal values and are 8 characters in length. The progress of the download is shown visually on the progress bar and when the download has completed a message box will be displayed indicating if the result of the download operation was successful or not.

It is assumed that the EEPROM has been programmed with the Function Image™ prior to the download. This can be carried out either with the EEPROM in circuit using the ES0001xx

EEPROM Load facility or with the EEPROM out of circuit using a proprietary EEPROM programmer.

To load the Function Image™ from file enter a file name or navigate to the required file name using the 'Browse' button. Select the activation code from the drop-down list or type it into the edit box and click on the 'Download' button. The progress of the download is shown visually on the progress bar and when the download has completed a message box will be displayed indicating if the result of the download operation was successful or not.

The Function Image™ is available from the CML website following registration and is in the form of a 'C' header file. The ES0001xx application software automatically parses the data in this file with no intervention from the user.

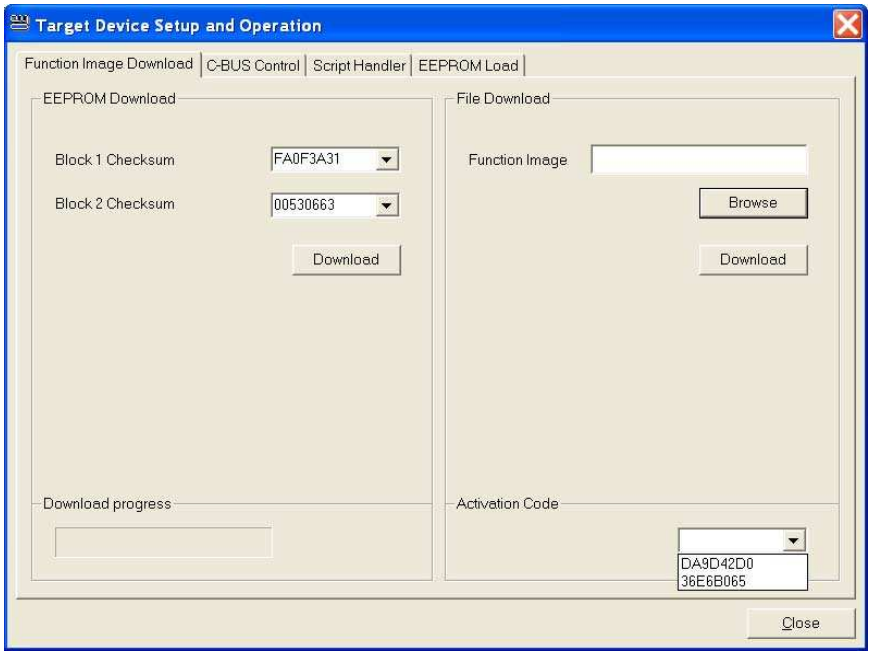


Figure 5 Function Image Download Tab

6.4.3 The Script Handler Tab

The Script Handler tab allows the execution of script files consisting of register read, write, delay and wait on IRQ commands.

The script files should be ASCII text files with each line containing one of the following commands:

- 8 bit register write – *WR8_AA_DD*
- 16 bit register write - *W16_AA_DDDD*
- 8 bit register read – *RD8_AA*
- 16 bit register read - *R16_AA*
- Delay - *DELAY_X..X*
- IRQ - *IRQ*
- IRQ with time-out - *IRQTO_X..X*
- Comment line - *//C..C*

AA is a 2 digit (1 byte), hexadecimal register address, _ is a single space character, *DD* is a 2 or 4 digit (1 or 2 bytes), hexadecimal register value, *X..X* is the duration of the delay time(units of 10ms) and *C..C* is an ASCII string. The delay and IRQ commands stop any following script commands from being executed until the delay or time-out period has passed, or an interrupt has been detected.

The script files can also contain jump commands that cause the execution of the script to branch forward the specified number of lines. There are two types of jump commands, unconditional and conditional. When an unconditional jump is executed the branch is always performed. For conditional jumps the branch is only performed if the condition is true. The jump commands must be in the following format:

- Unconditional jump – *JMP_L...L*
- Jump if equal - *JEQ_AA_VVVV_L...L*
- Jump if < – *JLT_AA_VVVV_L...L*
- Jump if > - *JGT_AA_VVVV_L...L*
- Jump if bit set - *JBS_AA_MMMM_L...L*

L...L is a decimal number representing the number of lines to branch forward in the script. For the conditional jump commands, *AA* is a 2 digit (1 byte), hexadecimal register address from which the read value is compared with *VV*, the result of which determines the jump condition. *VV* is a 2 or 4 digit (1 or 2 bytes), hexadecimal value. When using the jump if bit set command, *MM*, which is a 2 or 4 digit (1 or 2 bytes), hexadecimal value is used as a mask. A logical 'AND' operation is performed with it and the value represented by *AA*. If the result of this logic operation is non-zero then the branch is performed.

Each line must contain only one command and no other characters. An example script file is shown below.

// Line 1 Write address $01 (C-BUS Reset) **WR8 01**

// Line 2 Write 0F60H to register $C0 **W16 C0 0F60**

// Line 3 Jump over line 4 if register $C6 has bit 0 set **JBS C6 0001 1**

// Line 4 250ms delay **DELAY 25**

// Line 5 Jump over line 6 if register $C1 is equal to zero **JEQ C1 0000 1**

// Line 6 Write 0000H to register address $C1 **W16 C1 0000**

// Line 7 Jump over line 8 if register $A9 is greater than 01FFH **JGT A9 01FF 1**

// Line 8 Read register $C6 **R16 C6**

// Line 9 Jump over line 10 if register $A9 is less than 01FFH **JLT A9 01FF 1**

// Line 10 Read register $C6 **R16 C6**

The results window will display the value returned by register read commands and 'IRQ Detected' when an IRQ has occurred after an IRQ command. These results can be saved to an ASCII text file or discarded by clicking on the 'Save Results' or 'Clear Results' buttons, respectively.

The ES0001xx.EXE application will update the displayed register values to reflect those of any register read or write commands contained in an executed script file. When a script file is being

executed the ‘Run Script…’ button will change to be the ‘Stop’ button, the rest of the tab will be disabled and the other tabs cannot be selected. A script file can be executed once, repeatedly until the ‘Stop’ button is pressed or a specified number of times. The execution of a script file can be stopped or paused at any time by clicking the appropriate button.

Details of any PE{target} Card specific controls are included in the relevant PE{target} Card user manual.

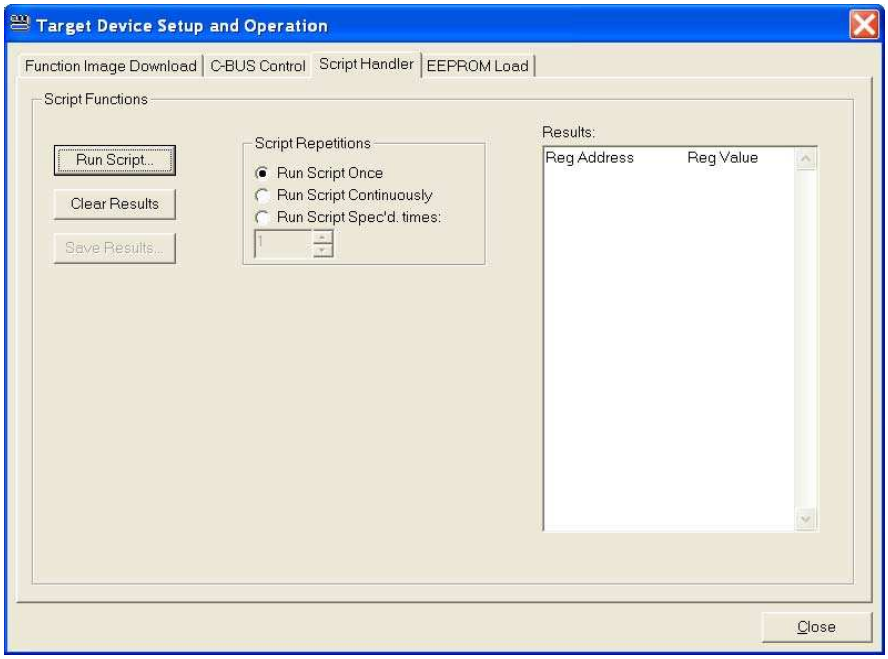


Figure 6 Script Handler Tab

6.4.4 The EEPROM Load Tab

With the EEPROM Load tab the user can load a Function Image™ into an EEPROM on the PE{target} card whilst it is in circuit. This negates the need to remove the EEPROM and have access to a proprietary EEPROM programmer.

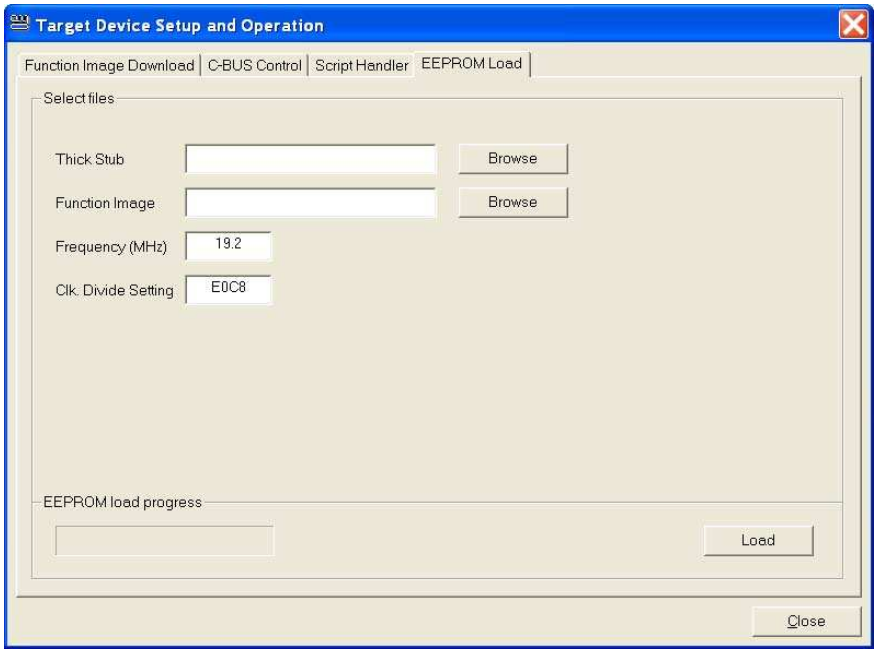


Figure 7 EEPROM Load Tab

To load the EEPROM, enter the file names for the Thick Stub and Function Image™ either by typing the filenames into the respective edit boxes or navigate to the files using the Browse button. The Thick Stub is a specialised Function Image™ and is available from the CML website following registration. Function Images™ are also available from the CML website following registration.

Type a value in units of MHz into the frequency edit box if a frequency other than the default 19.2MHz value is required. The frequency value refers to the external clock/crystal frequency for the CMX[target} device on the PE{target} card. For example, the PE0201 by default uses a 19.2MHz external clock module and the PE0401 uses a 6.144MHz crystal. The Clk. Divide Setting is dependent on the entered frequency value and is calculated by the software, requiring no intervention from the user.

Shortly after pressing the Load button a message box will confirm that the application has loaded the Thick Stub. Click on the message box OK button and the application will proceed to load the Function Image™ into the EEPROM on the PE{target} card. Load progress is shown visually on the progress bar. When the load is complete a message box will be displayed indicating if the result of the load operation was successful or not.

6.5 Evaluation Tests

Details of any PE{target} Card specific tests are included in the relevant PE{target} Card user manual.

6.6 TroubleShooting

After loading a Function Image™ the ES0001xx application writes the activation code that has been typed into the Activation Code edit box to the CMX70xx device. If this code is incorrect for the Function Image™ that has just been loaded the CMX70xx target device will lock up and will not respond to further input from the ES0001xx application. It is recoverable only by power cycling the PE{target} card, closing the ES0001xx application and then restarting the application.

7. Performance Specification

7.1 Electrical Performance

7.1.1 Absolute Maximum Ratings

Exceeding these maximum ratings can result in damage to the Evaluation Kit.

	Min.	Max.	Units
Supply (V_{DDD} - GND_D)	-0.3	4.0	V
Voltage on any connector pin to GND_D	-0.3	V_{DD} + 0.3	V
Current into or out of V_{DDD} and GND_D pins	-50	+50	mA
Current into or out of any other connector pin	-20	+20	mA
Storage Temperature	-10	+70	℃
Operating Temperature	+10	+35	℃

7.1.2 Operating Limits

Correct operation of the Evaluation Kit outside these limits is not implied.

	Notes	Min.	Max.	Units
Supply (V_{DDD} - GND_D)		3.0	3.6	V
Operating Temperature		+10	+35	℃
Xtal Clock Frequency		11.05	11.07	MHz

7.1.3 Operating Characteristics

For the following conditions unless otherwise specified:

Xtal Frequency = 11.0592MHz, Bit Rate = 19.2k bits/sec,
Noise Bandwidth = Bit Rate, V_{DD} = 3.3V, Tamb = +25℃.

	Notes	Min.	Typ.	Max.	Units
DC Parameters					
I_{DD} (not powersaved)	1, 2		76	85	mA
µC Interface					
Input logic "1" level	3, 4	70%			V_{DDD}
Input logic "0" level	3, 4			30%	V_{DDD}
Input leakage current (Vin = 0 to V_{DDD})	3, 4	-1.0		+1.0	µA
Input capacitance	3, 4		10		pF
Output Logic '1' Level - Ports 1, 3, 4, 5 @ I_{OH} = 1 mA		80%	-	-	V_{DDD}
Output Logic '0' Level - Ports 1, 3, 4, 5 @ I_{OL} = -1.5 mA		-	-	0.4	V
Output Logic '1' Level - D7-0, A15-0, WEN, OEN, CSN1, CSN2, CSN3 @ I_{OH} = 2 mA		80%	-	-	V_{DDD}
Output Logic '0' Level - D7-0, A15-0, WEN, OEN, CSN1, CSN2, CSN3 @ I_{OL} = -3 mA		-	-	0.4	V
'Off' state leakage current (Vout = V_{DDD})	5			10	µA
C-BUS clock frequency		-	1.4	-	MHz

Notes:
1. Not including any current drawn from the modem pins by external circuitry.
2. No PE{target} card connected.
3. WRN, RDN, CSN, A0 - A6 pins.
4. D0 - D7 pins.
5. IRQN pin.

CML Microcircuits (UK) Ltd COMMUNICATION SEMICONDUCTORS	CML Microcircuits (USA) Inc. COMMUNICATION SEMICONDUCTORS	CML Microcircuits (Singapore) Pte Ltd COMMUNICATION SEMICONDUCTORS	
		Singapore	China
Tel: +44 (0)1621 875500 **Fax:** +44 (0)1621 875600 **Sales:** sales@cmlmicro.com **Tech Support:** techsupport@cmlmicro.com	**Tel:** +1 336 744 5050 800 638 5577 **Fax:** +1 336 744 5054 **Sales:** us.sales@cmlmicro.com **Tech Support:** us.techsupport@cmlmicro.com	**Tel:** +65 67450426 **Fax:** +65 67452917 **Sales:** sg.sales@cmlmicro.com **Tech Support:** sg.techsupport@cmlmicro.com	**Tel:** +86 21 6317 4107 +86 21 6317 8916 **Fax:** +86 21 6317 0243 **Sales:** cn.sales@cmlmicro.com.cn **Tech Support:** sg.techsupport@cmlmicro.com
- www.cmlmicro.com -			