Table of Contents

1 Introduction

1.1 Eagle Family of Controllers

The Eagle series of single board computers run on an ARM Cortex-M3 microcontroller with a vast array of peripherals from 10/100 Ethernet to a reprogrammable CPLD. Through example programs and project files this family of single board computers can be developed from concept to production quickly. The Eagle family is available in custom and standard configurations.

1.2 Eagle 100 Overview

The Eagle 100 is a single board computer designed for cost-sensitive control applications that require real-time performance, networking and extensive support of popular peripherals. It delivers 32-bit performance and features at a cost equivalent to legacy 8- and 16-bit controllers. Powered by a Texas Instruments' Stellaris LM3S6918 ARM Cortex-M3 microcontroller, capable of over 60 MIPS, the Eagle 100 can fulfill demanding requirements in monitoring, instrumentation, data acquisition, process control, factory automation and many other applications. An extensive array of peripherals is built-in plus expandability is available using PC/104 I/O cards. Several peripherals can be reconfigured with the programmable logic capabilities of the integrated CPLD.

1.3 Eagle 100 Features

- 50 MHz 32-bit ARM® Cortex?-M3
- 256 KB Flash/64 KB SRAM
- 10/100 Ethernet
- Micro-SD Socket
- LCD Port
- 4x4 Keypad Port
- PC/104 8-bit I/O expansion
- Two RS232 Serial Ports
- I2C, SSI (SPI) Ports
- 8-ch. 10-bit ADC, 4-ch. 10-bit DAC
- Reconfigurable I/O with Xilinx CPLD
- Up to 52 - 5V tolerant GPIOs, 80 without PC/104
- 4 general purpose timers, 1 watchdog timer
- Support for GNU and IAR compilers
- Thumb2 instruction set for smaller object code
- Basic and Python also supported
- +5V@250mA power supply required
- Dimensions: 3.8 inches x 4.4 inches

Figure 1.3: Eagle 100 Hardware Block Diagram

1.4 Eagle 100 Development Kit

The EAGLE 100-20 development kit comes with all of the necessary hardware and software to quickly develop applications. The development kit includes the following:

- 1 ? EAGLE 100 SBC
- 1 ? IAR J-Link Debugger
- 1 ? IAR Embedded Workbench for ARM, 32K Kickstart Version

The Eagle SBC is also available unbundled, without the power supply and debugger.

1.5 Software and Support

Code examples are included with the Eagle 100 to get you started quickly. Applications can run standalone with no operating system or can use a compact real time operating system such as FreeRTOS. You can use popular IDEs together with the GNU and IAR compilers. The microSD card capability simplifies program and data storage. Remote access can be implemented via web or command line interfaces, providing off-site monitoring and maintenance capabilities. The JTAG interface speeds up application development and debugging.

Ports of popular Basic and LUA development tools are available for the Eagle 100 to reduce application development time and simplify integration with code libraries developed for industrial and scientific environments. Using these tools, you can achieve significant functionality in a very short time. These open source tools can be easily extended, allowing a virtually unlimited number of possibilities.

Micromint USA provides free technical support by phone, email, or fax. Technical support emails are usually answered within one business day. Software and documentation updates are available on our website at www.micromint.com. Each product comes with a one year warranty.

NEXT: Getting Started

PREVIOUS: Table of Contents

2 Getting Started

2.1 Steps to Getting Started

Getting started with the Eagle 100 can be done in just 6 steps.

- 1. Run the Preloaded Application
- 2. Install the Eagle Code Examples
- 3. Choose a Compiler and IDE (Integrated Development Environment)
- 4. Install the Chosen Compiler and IDE
- 5. Compile a Code Example
- 6. Download the Code Example to the Eagle 100

2.2 Run the Preloaded Application

The Eagle 100 come from the factory preloaded with an application called enet_io that demonstrates web-based I/O control. The preloaded applications should be run to test the board after receiving it. The full source code for the application is provided on the Eagle's Wiki. http://wiki.micromint.com/index.php/Eagle_Documentation

2.2.1 Eagle SBC enet_io Application

The Eagle Single Board Computer's (SBC) with Ethernet are shipped with the enet_io example application from the TI Stellarisware Library. This example application demonstrates web-based I/O control using the Ethernet controller and the lwIP TCP/IP Stack. DHCP is used to obtain an Ethernet address. If DHCP times out without obtaining an address, a static IP address will be chosen using AutoIP. The address that is selected will be shown on COM1, allowing access to the internal web pages served by the application via a normal web browser. Figure 2.1 shows the web page that will first be displayed.

Figure 2.1: Web interface for the enet_io application

In order to demonstrate this application on an Eagle SBC without the optional USB Debug Port you will need the following software and equipment:

- 1. A Personal Computer (PC) with a serial port or a USB to serial port adapter
- 2. A DB9 to 2x5 adapter cable if using an Eagle SBC where COM1 is a 2x5 pin header
- 3. A null modem cable
- 4. A terminal program such as Putty, HyperTerminal, or TeraTerm
- 5. A web browser
- 6. One Ethernet cable
- 7. One +5V power supply

Please follow these steps to run the enet_io application on an Eagle SBC without the optional USB Debug Port:

- 1. If the Eagle SBC's COM1 port to the computer using the null modem cable. NOTE: Use the 2x5 adapter cable if COM1 is a 2x5 pin header.
- 2. Connect the Eagle SBC's Ethernet port to an Ethernet Network.
- 3. Start and set-up the terminal program with the following settings.
    - ♦ a. Baudrate ? 115200
    - ♦ b. Data Bits ? 8
    - ♦ c. Parity ? None
    - ♦ d. Stop Bits ? 1
    - ♦ e. Flow Control ? None
- 4. Apply power to the board.
- 5. The Eagle SBC will transmit the boards IP address to the terminal
- 6. Open a web browser.
- 7. Type the IP address into the web browsers address bar and the web page in Figure 2.1 should load.
- 8. Click on ?I/O Control Demo 1 (HTTP Requests)?.
- 9. Click on the ?Toggle LED? button to turn the User LED on.

In order to demonstrate the enet_io application on an Eagle SBC with the optional USB Debug Port you will need the following software and equipment:

- 1. A Personal Computer (PC) with a USB port
- 2. A terminal program such as Putty, HyperTerminal, or TeraTerm
- 3. A web browser
- 4. One USB A to micro USB cable (Supplied with Eagle SBC with USB Debug Port)
- 5. One Ethernet cable

Please follow these steps to run the enet_io application on an Eagle SBC with the optional USB Debug Port:

- 1. Make sure two jumpers are set correctly to enable power to come from the Debug port and enable the virtual COM port. The Eagle 50 jumpers are JP1 for power and and JP6 for the virtual COM port.
- 2. Connect the Eagle SBC's Ethernet port to an Ethernet Network.
- 3. Connect the Eagle SBC to the PC by using the USB A to micro USB cable.
- 4. Wait until the USB drivers have been installed. The USB drivers can be found in the Software Updates section on the Eagle's Wiki. http://wiki.micromint.com/index.php/Eagle_Documentation
- 5. Find out what COM port the Eagle SBC's virtual COM port is by looking in Windows Device Manager?s PORTS (COM & LPT). The Eagle SBC's COM port is the one that says Stellaris Virtual COM Port.
- 6. Start and set-up the terminal program for the COM port that the Eagle SBC is on and with the following settings.
    - a. Baudrate ? 115200
    - b. Data Bits ? 8
    - c. Parity ? None
    - d. Stop Bits ? 1
    - e. Flow Control ? None
- 7. Press the Reset button.
- 8. The Eagle SBC will transmit the boards IP address to the terminal.
- 9. Open a web browser.
- 10. Type the IP address into the web browsers address bar and the web page in Figure 2.1 should load.
- 11. Click on ?I/O Control Demo 1 (HTTP Requests)?.
- 12. Click on the ?Toggle LED? button to turn the User LED on.

2.3 Install the Eagle Code Examples

The Eagle Code Examples are generated from Stellarisware and may be downloaded from the Software Updates section of the Software Updates section of the Eagle Wiki. After they are downloaded unzip them into a directory of your choice. Descriptions of the examples can be viewed on the Eagle Examples Page.

2.4 Choose a Compiler and IDE (Integrated Development Environment)

2.4.1 Choosing a Compiler

The Code Examples currently supports the following C and C++ compilers :

- IAR Embedded Workbench for ARM (ewarm) 5.40
- GNU Toolchain (gcc) for ARM 4.4.1 ? CodeSourcery G++ 2010q1
- GNU Toolchain (gcc) for ARM 4.4.3 ? devkitARM 30

The IAR EWARM C/C++ compiler generally produces the smallest code sizes for ARM targets and has excellent integrated debugging capabilities versus the GNU Toolchain. If a GNU chain is used then an IDE needs to be chosen.

2.4.2 Choosing an IDE

An IDE installs when the IAR C/C++ Compiler is installed where the GNU tool chains do not install one. Code::Blocks IDE and the Eclipse IDE are the IDEs currently supported by the Code Examples. Debugging is currently not supported in the Code::Blocks IDE. If a debug environment is needed the Eclipse IDE should be use.

2.5 Install the Chosen Compiler and IDE

2.5.1 Installing EWARM

The IAR EWARM Kickstart Edition is a 32 KB code-sized limited version of the IAR C/C++ compiler and debugger. It can be downloaded from IAR's website.

After downloading the EWARM-KS-CD click on the application to install the IAR Embedded Workbench for ARM. Select the ?Install IAR Embedded Workbench? option from the Applications main menu as shown in Figure 2.4. Follow the instructions in the installation application. We suggest that you use the default directories, and the ?Full? installation option.

Figure 2.4: IAR Kickstart Main Menu

2.5.2 Installing a GNU Compiler and IDE

The Code Examples currently support two GNU tool chains:

- GNU Toolchain (gcc) for ARM 4.4.1 ? EABI CodeSourcery G++ Lite 2010q1
- GNU Toolchain (gcc) for ARM 4.4.3 ? devkitARM 30

2.5.2.1 Installing the Sourcery CodeBench Lite Edition for ARM EABI GNU Compiler

Download the Sourcery G++ Lite 2010q1 for ARM EABI version of Sourcery CodeBench Lite Edition for ARM from Mentor Graphics website. When it is finished downloading click on the application to install it and a screen similar to Figure 2.5 should appear. We suggest that you use the default directories during the installation.

If make is not installed on the computer then perform the following steps:

- 1. Open a command prompt
- 2. Type "cd\program files\codesourcery\sourcery g++ lite\bin"
- 3. Press enter
- 4. Type "copy cs-make.exe make.exe"
- 5. Press enter

Figure 2.5: Sourcery CodeBench Lite Edition for ARM EABI Main Menu

2.5.2.2 Installing the devkitARM GNU Compiler

Installing the devkitARM GNU Compiler can be done in 6 steps.

- 1. Download the devkitARM compiler from the sourceforge website by clickin the following link: http://sourceforge.net/projects/devkitpro/files/devkitARM/
- 2. Create a folder called devkitPro on the C drive.
- 3. Extract the contents into the "C:\devkitPro" folder. It should create a folder called "devkitARM".
- 4. Download the msystools from the sourceforge website by clicking the following link: http://sourceforge.net/projects/mingw/files/MSYS/Base/msys-core/msys-1.0.10/MSYS-1.0.10.exe/download
- 5. Run the msystools installer.

2.5.2.3 Installing the Eclipse IDE

Eclipse is a Java application and has the potential to be run on a wide variety of hardware and operating systems. Eclipse may install on systems with as little as 64MB of memory, however, we recommend to have 1GB of memory or more.

Follow these steps to install the Eclipse IDE:

- 1. Install the desired GNU ARM Toolchain (EABI CodeSourcery G++ Lite 2010q1 or devkitARM 30).
- 2. Be sure the Java Runtime Environment (JRE) is installed.
- 3. The Eclipse SDK includes the Eclipse Platform is provided as an archive and can be downloaded on the following website: http://www.eclipse.org/cdt/downloads.php
- 4. Extract the Eclipse SDK to its desired directory (commonly ?C:\Program Files\?).
- 5. Start Eclipse by double clicking on the "eclipse.exe" file where the Eclipse SDK was extracted to.
- 6. Select the workspace directory. "StellarisWare\boards\eagle" for the Eagle and "Lincoln\CMSIS\projects" for the Lincoln.

Figure 2.6 ? Specifying Workspace in Eclipse

- 7. Click on the "Help" drop down menu and select "Install new software?".
- 8. Expand "CDT Main Features" and check "Eclipse C/C++ Development Tools"

Figure 2.7 ? Installing the CDT plug-in

- 9. Click the "Next" button.
- 10. Copy http://opensource.zylin.com/zylincdt and paste it into the "Work with:" box.
- 11. Click the "Add" button.
- 12. Check "Zylin Embedded CDT".
- 13. Click the "Next" button.
- 14. Copy http://sourceforge.net/projects/gnuarmeclipse/files/Eclipse/updates/ and paste it into the "Work with:" box.
- 15. Click the "Add" button.
- 16. Check "CDT GNU Cross Development Tools".
- 17. Click the "Next" button.

2.5.2.4 Installing the CodeBlocks IDE

Codeblocks is a cross-platform IDE built around wxWidgets, designed to be extensible and configurable. It can be downloaded from the Codeblocks website by clicking the following link: http://www.codeblocks.org/downloads

Install Codeblocks by clicking on the downloaded executable.

Figure 2.8: CodeBlocks IDE installation

2.6 Compile a Code Example

2.6.1 Compiling Code Examples with EWARM

Follow these steps to compile an example using EWARM.

- 1. Open IAR Embedded Workbench for ARM (EWARM).
- 2. Click "File|Open|Workspace..." from the drop down menu.
- 3. Browse to the "StellarisWare\boards\eagle" directory and select the eagle workspace. The work space should look similar to the one in Figure 2.5.
- 4. Right click on the desired project and select "Set as Active".
- 5. Right click on the project and select "Rebuild".

Figure 2.5: Using StellarisWare projects with the IAR EWARM IDE

2.6.2 Compiling Examples with the GNU Toolchain

2.6.2.1 Compiling Examples with the Code::Blocks IDE

Follow these steps to compile an example using the Code::Blocks IDE.

- 1. Open the Code::Blocks IDE.
- 2. Click "File" from the drop down menu.
- 3. Click "Open".
- 4. Browse to the "StellarisWare\boards\eagle" directory and select the eagle workspace. The work space should look similar to the one in Figure 2.6.
- 5. Right click on the desired project and select "Activate project".
- 6. Right click on the project and select build.

Figure 2.6: Using the StellarisWare projects with the CodeBlocks IDE

2.6.2.2 Compiling Examples with the Eclipse IDE

Follow these steps to compile an example using the Eclipse IDE.

- 1. Open the Eclipse IDE.
- 2. When prompted browse to the "StellarisWare\boards\eagle" directory.
- 3. Click 'OK'.
- 4. Press the 'F5' key on the keyboard to refresh the files.
- 5. Close all of the open projects by right clicking on the project and selecting "Close Project".
- 6. Open a project by right clicking on the desired project and selecting "Open Project". If the Sourcery G++ Lite compiler is not installed and devkitARM is installed then click on "Window>Preferences>C/C++>Build>Environtment" and add a variable called "COMPILER" with the value set to "devkitARM".
- 7. Right click on the project and select build.

2.6.2.3 Compiling Examples with the GNU Toolchain from the Command Line

All example programs include a Makefile that allows you to build binary images from the command line using the GNU toolchain. The GNU ?make? utility is installed as part of the GNU toolchain on the Eagle Setup CD. To build an image using the command line, just change to the project directory and execute ?make?. To build the image for blinky you would perform the following:

- 1. Open a command prompt.
- 2. Type ?CD\Program Files\Micromint\Eagle\StellarisWare\boards\eagle\blinky?.
- 3. Press ?Enter?.
- 4. Type ?make?.

- 5. Press ?Enter?.

2.7 Downloading Code Examples to the Eagle 100

There are many options to download firmware to the Eagle SBC. This section covers firmware downloads using a J-Link debugger, USB Debugger, picoJTAG, Ethernet Bootloader, and Serial Bootloader. The method used to download firmware will depend on what hardware is being used.

2.7.1 Downloading Code Examples using EWARM

EWARM can be configured to download firmware using a variety of different debuggers. The following section explains how to set-up EWARM for using the J-Link, optional USB Debugger, and the picoJTAG.

2.7.1.1 Steps for using the J-Link and EWARM

- 1. Select "Projects>Options" from the drop down menu.
- 2. Select the "Debugger" category.
- 3. On the "Setup" tab select ?J-Link/J-Trace?.
- 4. Make sure "Run to" has a check next to it and "main" is written in the text box.
- 5. On the "Download" tab make sure "Verify download" and "Use flash loader(s) are checked
- 6. Click the "OK" button.
- 7. Select "Projects>Download and Debug" from the drop down menu.
- 8. Select "Debug>Go" from the drop down menu.

2.7.1.2 Steps for using the optional USB Debugger on the Eagle 50 and EWARM

- 1. Select "Projects>Options" from the drop down menu.
- 2. Select the "Debugger" category.
- 3. On the "Setup" tab select ?LMI FTDI?.
- 4. Make sure "Run to" has a check next to it and "main" is written in the text box.
- 5. On the "Download" tab make sure "Verify download" and "Use flash loader(s) are checked
- 6. Click the "OK" button.
- 7. Select "Projects>Download and Debug" from the drop down menu.
- 8. Select "Debug>Go" from the drop down menu.

2.7.1.3 Steps for using the picoJTAG and EWARM

Please see the picoJTAGs Wiki: http://wiki.micromint.com/index.php/PicoJTAG_Manual/Getting_Started#IAR_Plugin

2.7.2 Downloading Code Examples using the CoFlash Flash Programmer and the picoJTAG

CooCox CoFlash is a stand-alone Cortex M Flash Programming software for PCs running Microsoft Windows. It can be downloaded from the following website: http://www.coocox.org/CoFlash_Programmer.htm

Follow these steps to program a Cortex M microcontroller using CoFlash and the picoJTAG:

- 1. Open the CoFlash programmer

Figure 2.9: Flash Programmer

- 2. Select picoJTAG as the Adpater
- 3. Select the microcontroller being used
- 4. Select the Command tab
- 5. Browse for the binary or elf file
- 6. Verify that the sector offset and sectors are properly selected. See below figures for further details

Figure 2.10: CoFlash Programmer set-up for Eagle SBC with bootloader

Figure 2.11: CoFlash Programmer set-up for Eagle SBC without bootlader

Figure 2.2: CoFlash Programmer set-up for Lincoln SBC

- 7. Click on the Program button

2.7.3 Downloading Code Examples using the Ethernet Bootloader

The Eagle 50E is shipped with an Ethernet bootloader that can be used to update the firmware on the board from an Ethernet connection. The bootloader uses the BOOTP and TFTP protocols to temporarily acquire an IP address and copy the binary image to the board. The LM Flash Programmer implements a small BOOTP and TFTP server to do this. The LM Flash Programmer can be downloaded from Texas Instruments website. http://www.ti.com/tool/lmflashprogrammer

The bootloader uses the first 8 KB of the flash address space (0x00000000 to 0x00001fff). Programs loaded with the bootloader should be linked to start at address 0x00002000. That is done automatically if you use the ?ewarm/application.icf? or ?gcc/application.ld? linker scripts used in the examples.

You can overwrite the bootloader if you so desire by using the ?ewarm/standalone.icf? or ?gcc/ standalone.ld? linker scripts and downloading your binary image via the JTAG. Currently the bootloader cannot be overwritten via the Ethernet port.

Follow the below steps to use the LM Flash Programmer and the Ethernet bootloader:

- 1. Connect the Eagle 50E?s Ethernet port to the network.
- 2. Apply power to the Eagle 50E.
- 3. Press and hold the Eagle 50E?s ?USER? button.
- 4. Press and release the Eagle 50E?s ?RESET? button.
- 5. Release the ?USER? button. The Eagle 50E?s ?USER LED? should now be blinking approximately once a second.
- 6. Open the LM Flash Programmer
- 7. Select the "Setup" tab and you will see a screen similar to Figure 2.9
- 8. For "Interface" select "Ethernet"
- 9. For "Ethernet Adapter:" select the adapter that is connected to the computers network.
- 10. For ?Client IP Address:? enter an IP address that is on the same subnet as your Ethernet adapter. The last octet must be different than the Ethernet Adapters.
- 11. For ?Client MAC Address:? enter the Eagle 50E?s serial number. It can be found on the bottom of the board and should start with 0021A3. Make sure to separate the octets with dashes (00-21-A3).

- 12. Select the "Program" tab on the LM Flash Programmer software and you will see a screen similar to Figure 2.10.
- 13. Click the ?Browse? button and locate the application to be downloaded.
- 14. Click on the ?Program? button. It generally takes several seconds for the program to start downloading.

Figure 2.9: Ethernet Bootloader Flash programmer configuration

Figure 2.10: Ethernet Bootloader flash programmer download
Ethernet bootloader troubleshooting guide:

- 1. Temporarily turn off the computers firewall. If this works add the LM Flash programmer to the Firewalls Exceptions list.
- 2. Try bypassing the network by connecting the Eagle 50E directly to the computer Ethernet port. A regular CAT5 cable will work. Make sure to close the LM Flash Programmer software and start at step 2 in the above procedure.

2.7.4 Downloading Code Examples using the Serial Bootloader

The Eagle 50 is shipped with a Serial bootloader that can be used in conjuction with the LM Flash Programming software to update the firmware on the board from a Serial connection. The LM Flash Programmer can be downloaded from Texas Instruments website http://www.ti.com/tool/lmflashprogrammer.

To place the bootloader in update mode you need to press and hold the user button while the board is starting up, e.g. via power up or when pressing the reset button. After the board starts up release the user button and you will see the user LED blinking approximately once per second. That indicates the board is ready to receive a firmware update via a Serial connection.

Programs loaded with the bootloader should be linked to start at address 0x00002000. That is done automatically if you use the ?ewarm/application.icf? or ?gcc/application.ld? linker scripts used in the examples. You can overwrite the bootloader if you so desire by using the ?ewarm/standalone.icf? or ?gcc/ standalone.ld? linker scripts and downloading your binary image via the JTAG. Currently the bootloader cannot be overwritten via the Serial port.

Follow the below steps to set up the LM Flash Loader:

- 1. Open the LM Flash Programmer
- 2. Select the "Setup" tab and you will see a screen similar to Figure 2.11
- 3. For "Interface" select "Serial Interface"
- 4. For "COM Port:" select the COM port that is assigned to "Stellaris Virtual COM Port" in the "Device Manager" for the Optioanl USB Debugger or the COM port that you will be using
- 5. The baudrate should be set to 115200
- 6. Set the transfer size to 76
- 7. Select the "Program" tab and you will see a screen similar to Figure 2.12
- 8. Set the "Program Address Offset" to 2000

- 9. Click the "Browse" button and locate the application to be downloaded

Figure 2.11: Serial bootloader flash programmer configuration

Figure 2.12: Serial bootloader flash programmer download
The application can be downloaded using the LM Flash Programmer and the optional USB Debugger by following these steps:

- 1. If you are powering the board via USB put a jumper on JP1
- 2. Make sure there is a jumper on JP6
- 3. Connect the USB cable to the computer and the board
- 4. Press and hold the "USER" pushbutton on the board
- 5. Press and release the "RESET" pushbutton on the board and the LED should start blinking around once a second
- 6. Release the "USER" pushbutton
- 7. Click on the "Program" button

The application can be downloaded using the LM Flash Programmer and COM1 by following these steps:

- 1. If you have an optional USB debugger make sure there is NOT a jumper on JP6
- 2. Connect the serial cable to COM1 (J3)
- 3. Power the board
- 4. Press and hold the "USER" pushbutton on the board
- 5. Press and release the "RESET" pushbutton on the board and the LED should start blinking around once a second
- 6. Release the "USER" pushbutton
- 7. Click on the "Program" button

2.7.5 Downloading Code Examples from the Command Line

The TI Flash Programmer can also be invoked from the command line. The command line allows you to program the Eagle SBC though Ethernet Bootloader, Serial Bootloader, or FTDI JTAG. This section provides a step by step procedure for each programming method. Follow these steps to program the board using the command line and the Ethernet Bootloader. The Eagle 50E and Eagle 100 comes with the Ethernet Bootloader.

- 1. Put the board in bootload mode by hold the user pushbutton while pressing and releasing the reset pushbutton. The user LED will blink when it is in bootload mode.
- 2. Enter the following command line example with your own parameters. Where 192.168.1.201 is the IP address of the computers Ethernet Port and 192.168.1.210 is a free IP address of the network the Eagle SBC is on. 00-21-A3-00-01-02 is the MAC address of the Eagle SBC. The MAC address is also the board?s serial number which is on the bottom of the board.

```
lmflash -i ethernet -n 192.168.1.201,192.168.1.210,00-21-A3-00-01-02 blinky.bin
```

- 3. The command line will give you status while the board is programming.

Follow these steps to program the board using the command line and the Serial Bootloader. The Eagle 50 comes with the Ethernet Bootloader.

- 1. Put the board in bootload mode by hold the user pushbutton while pressing and releasing the reset pushbutton. The user LED will blink when it is in bootload mode.
- 2. Enter the following command line example with your own parameters. Where COM7 is the COM port that is connected to the Eagle 50 SBC

```
lmflash -i serial -d -p COM7 -b 115200 -o 0x2000 -r blinky.bin
```

- 3. The command line will give you status while the board is programming.

Follow these steps to program the board using the command line and the USB Debug Port

- 1. Connect the USB Debug Port to the computers USB port.
- 2. Enter the following command line example with your own parameters

lmflash ?I ftdi ?r ?o 0x2000 blinky.bin 3. The command line will give you status while the board is programming.

To see all options available from the command line, use ?lmflash ?h?. To see other examples of command lines, use ?lmflash ?examples?.

NEXT: Hardware

PREVIOUS: Introduction

3 Hardware

The following image shows where some of the hardware components are located.

Eagle 100 Hardware

3.1 Microcontroller

The Eagle 100 includes a Texas Instruments' Stellaris LM3S6918 microcontroller. This 32-bit ARM Cortex-M3 RISC microcontroller is capable of 50-MHz operation with a Thumb2 instruction set for smaller object code. It has hardware division and single cycle multiplication for fast calculations. The nested vector interrupt controller provides interrupt handling for 33 interrupts with eight levels of priority. Please see the Texas Instruments? LM3S6918 Microcontroller Data Sheet for more information and register definitions.

LM3S6918 key features:

- Internal Memory
  - ♦ 256 kilo-bytes flash
  - ♦ 2-kB flash block protection defined by the user
  - ♦ Flash data programming
  - ♦ User defined and managed flash-protection block
  - ♦ 64 kilo-bytes SRAM
- Timers
  - ♦ Four General Purpose Timer Modules (GPTM)

  Each GPTM can operate independently
  Each module provides two 16-bit timers or one 32-bit timer
  Can be used for Pulse Width Modulation (PWM)
  Can be used to trigger analog to digital conversions

- ♦ System Timer (SysTick)

24-bit clear on write decrementing counter
Uses include RTOS tick timer, high speed alarm timer, or a simple counter

- ♦ ARM FIRM compliant Watchdog Timer

32-bit decrementing counter
Programmable load register
Separate clock with an enable

- 10/100 Ethernet Controller
  - ♦ IEEE 802.3-2002 specification
  - ♦ Half and full duplex for 10 Mbps and 100 Mbps
  - ♦ Automatic MDI/MDI-X cross over correction
  - ♦ Programmable MAC address
  - ♦ Power down and power saving modes
- UART
  - ♦ Two fully programmable 16C550 type UARTs
  - ♦ Separate transmit and receive FIFOs
  - ♦ Baud rates up to 3.125 Mbps
  - ♦ Loopback mode for testing and debugging
- Synchronous Serial Interface (SSI)
  - ♦ Master or slave operation
  - ♦ Separate transmit and receive FIFOs
  - ♦ Freescale SPI, MICROWIRE, or Texas Instruments synchronous serial interfaces
  - ♦ Internal loopback mode for testing and debugging
- I2C Module
  - ♦ Master or slave operation
  - ♦ 100 kbps or 400 kbps transmission speed
  - ♦ Mulitmaster support
  - ♦ 7-bit addressing mode
- Analog-to-Digital Converter
  - ♦ Eight 10-bit channels
  - ♦ Single and differential input configurations
  - ♦ 500k samples per second
  - ♦ Four programmable sampling sequences with conversion result FIFOs
  - ♦ Sequences triggered by software, timers, analog comparators, or GPIO
  - ♦ On-chip temperature sensor
- General Purpose Input or Outputs (GPIO)
  - ♦ 5V tolerant inputs
  - ♦ Every GPIO is capable of edge triggered or level sensitive interrupts
  - ♦ Enable or disable internal weak pull-up or pull-down resistors
  - ♦ Programmable open drain input or outputs
  - ♦ Programmable drive current of 2mA, 4mA, or 8mA
- Reset Sources
  - ♦ Power on reset
  - ♦ Reset pin assertion
  - ♦ Brown out reset
  - ♦ Software reset
  - ♦ Watchdog timer reset
- Additional Features
  - ♦ IEEE 1149.1-1990 compliant Test Access Port (TAP) controller
  - ♦ Debugging via JTAG or Serial Wire interfaces
  - ♦ Programmable PLL for system clock

3.2 CPLD

A Xilinx?s XC9572XL Complex Programmable Logic Device(CPLD) comes standard on the Eagle 100. The CPLD supports in-system programming via an IEEE 1149.1 boundary-scan JTAG. The XC9572XL is a 3.3V CPLD with 5V tolerant pins. The CPLD has 1,600 usable gates and 72 macrocells. For further information please see Xilinx?s XC9500XL High-Performance CPLD Family Data Sheet.

3.3 DAC

The Eagle 100 includes a National Semiconductor?s DAC104S085 general purpose digital-to-analog converter (DAC). The DAC has four channels with a resolution of 10-bit. The output amplifiers allow for a rail-to-rail output swing from 0 to 3.3V. Communication to the

DAC is done through a three wire synchronous serial interface that operates up to 40 MHz. The DAC?s outputs have a settling time of 6µs. It allows for simultaneous output updating. For further information please see National Semiconductor?s DAC104S085 Data Sheet.

NEXT: User Interfaces, Connectors, and Jumpers

PREVIOUS: Getting Started

4 User Interfaces, Connectors, and Jumpers

The following image shows where the connectors, headers, and jumpers are located on the Eagle 100.

4.1 Power Supply

The Eagle 100 requires a regulated +5VDC at 250mA power supply applied to J1. J1 comes standard with a 2.5 mm positive center tapped female power supply jack. It can be populated with a 2 position screw terminal upon request. A diode (D1) will protect the Eagle 100 should polarity of the power supply be reversed. When power is applied LED1 will illuminate.

WARNING: Supply voltages over +5VDC while a LCD is connected may damage the LCD.

Figure 4.1: Power supply connector configurations

4.2 10/100 Ethernet

The LM3S6918 is equipped with a fully-integrated 10/100 Mbps Ethernet Controller. Both the Ethernet Media Access Control (MAC) and Physical (PHY) layers are integrated in the microcontroller. The RJ-45 connector with integrated magnetics and built in LEDs completes the Ethernet sub-system. Please see the LM3S6918 data sheet for further information on the Ethernet controller.

4.3 Serial (COM) Ports

Both Universal Asynchronous Receivers/Transmitters (UARTs) are level shifted to RS-232 levels. UART0 (COM1) reaches the external world through a male DB9 connector. UART1 (COM2) reaches the external world through a 2x5 pin berg header. Please see figure 4.3 for the pin outs of COM1 (J3) and COM2 (J4) connectors. The two serial ports support software handshaking (XON/XOFF). To simplify interfacing to devices using hardware handshaking, a loopback is implemented on the modem control signals, from RTS to CTS and from DTR to CD and DSR. Note that the loopbacks do not provide flow control so software handshaking should be used when proper flow control is desired.

Figure 4.3: COM port connector pin outs

4.4 Micro-SD

The microSD socket (J9) enables micro-secure-digital memory cards to be plugged into the Eagle 100 microcontroller board. The microSD card allows the user the ability of a standard removable media for transferring data to and from the Eagle 100.

4.5 General Purpose Digital Inputs and Outputs

The general purpose digital inputs/outputs (I/O) are broken into two different categories GPIO (General Purpose Input/Outputs) and Extended I/O. The GPIO is accessed directly through the LM3S6918 microcontroller and the Extended I/O is accessed through the CPLD (Complex Programmable Logic Device) via the microcontrollers Synchronous Serial Interface (SSI) port zero.

There are twenty bits of GPIO available on the J2 connector. Please see the pin out for J2 in Figure 4.5. Eight bits are from port B, eight are from port C and four are from port E. NOTE: If the keypad port (J11) is used then port B on the J2 connector should not be used (pins 3 through 10 of J2). Ports B, C, and E have alternate functions other than digital inputs and outputs. Table 4.5 lists the alternate functions and a brief description of the function. For further information on the alternate functions please refer to the LM3S6918 data sheet.

The J2 connector also has the input and output for the hibernation module. Pin 29 is the WAKE input that brings the microcontroller out of hibernation mode when it is asserted. Pin 30 is the HIB output that indicates the microcontroller is in hibernation mode. The power source (VBAT) for the hibernation module can be accessed from Pin 28 of J2.

The extended I/O is accessed through the CPLD. There are four eight bit ports PXA, PXB, PXC, and PXD. The pin out for the extended I/O connector (J12) can be viewed in Figure 4.5. The numbers in parenthesis are the pin numbers for the CPLD. The provided VHDL firmware can be changed to use the extended I/O ports for application specific purposes. NOTE: If the LCD connector (J10) is used then PXD of the extended I/O is no longer available and nothing should be connected to pins 33 through 40 of J12.

J2 Pin#	GPIO	Alternate Function	Brief Description
3	PB0	CCP0	Capture/Compare/Pulse Width Modulation Channel 0
4	PB1	CCP2	Capture/Compare/Pulse Width Modulation Channel 2
5	PB2	I2C 0SCL	Inter-Integrated Circuit Interface bus 0 clock
6	PB3	I2C 0SDA	Inter-Integrated Circuit Interface bus 0 clock
7	PB4	C0-	Analog comparator channel 0 negative input
8	PB5	C1-	Analog comparator channel 1 negative input
9	PB6	C0+	Analog comparator channel 0 positive input
10	PB7	TRST	JTAG Test Reset
13	PC0	TCK/SWCLK	JTAG Test Clock/Serial Wire Debug clock
14	PC1	TMS/SWDIO	JTAG Test Mode Select/Serial Wire Debug Input and Output
15	PC2	TDI	JTAG Test Data Input
16	PC3	TDO/SWO	JTAG Test Data Output and SWO
17	PC4	CCP5	Capture/Compare/Pulse Width Modulation Channel 5
18	PC5	C1+/C0o	Analog comparator channel 1 positive input/Analog comparator channel 0 output
19	PC6	CCP3	Capture/Compare/Pulse Width Modulation Channel 3
20	PC7	CCP4	Capture/Compare/Pulse Width Modulation 4
23	PE0	SSI1Clk	Synchronous Serial Interface bus 1 clock
24	PE1	SSI1Fss	Synchronous Serial Interface bus 1 function slave select

25	PE2	SSI1Rx	Synchronous Serial Interface bus 1 receiver
26	PE3	SSI1Tx	Synchronous Serial Interface bus 1 transmitter

Table 4.5: GPIO alternate functions

Figure 4.5: Extended I/O and GPIO connector pin out (CPLD pin#)

4.6 Keypad

A 4x4 matrix keypad using a 16-pin (2x8) ribbon cable can be connected to port B of the microcontroller through J11. Please see Figure 4.6 for the pin out of the keypad connector. NOTE: If the keypad port (J11) is used then port B on the J2 connector should not be used (pins 3 through 10 of J2).

Figure 4.6: Keypad connector pin out

4.7 Liquid Crystal Display (LCD)

A standard alphanumeric LCD may be connected to J10 through a 32-pin (2x16) ribbon cable. Extended port D is the byte-wide port used for the LCD?s data bus. The LCD?s control signals and backlight are driven by the CPLD as well. The contrast for the LCD may be adjusted by turning potentiometer R33 located next to J10. Please see figure 4.7 for the LCD?s connector pin out. NOTE: If the LCD connector (J10) is used then PXD of the extended I/O is no longer available and nothing should be connected to pins 33 through 40 of J12.

Figure 4.7: LCD connector pin out (CPLD pin#)

4.8 JTAG

The JTAG port can be used for software download and debugging, reducing the need for an in-circuit emulator. For detailed information on the operation of the JTAG port and TAP controller, please refer to IEEE Standard 1149.1-Test Access Port and Boundary-Scan Architecture. JTAG connector

Figure 4.8: JTAG connector pin out

4.9 Analog to Digital Converter (ADC)

The Eagle 100?s eight channels of 10-bit ADC can be connected to through J7. Please see figure 4.9 for the pin out of the ADC connector. The ADC is accessed directly through the LM3S6918 microcontroller. It is capable of 500 kilo-samples/second and can be configured as eight single ended or four differential channels. The ADC can be triggered to read through software, timers, analog comparators, or GPIO. An internal temperature sensor may be read using the ADC module. Please see the LM3S6918 datasheet for further details. ADC connector

Figure 4.9: Analog to Digital connector pin out

4.10 Digital to Analog Converter (DAC)

The Eagle 100?s four channels of 10-bit DAC can be connected to through J8. Please refer to figure 4.10 for the pin out of the DAC connector. The DAC is accessed through the LM3S6918?s SSI0 port. Port G bit 0 is the DAC?s sync input for loading the conversion count into the DAC. The DAC can be updated at a maximum of 150 kHz for 1 channel. The data transfer is 16-bits at 25 MHz which is a total of 640 nS per data transfer but the DAC has a settling time of 6 µS so the total time needed is 6.64 µS. All 4 channels of the DAC can be updated at a maximum of 116 kHz. This is accomplished by sending the data for all 4 channels and updating all of the outputs on the last data transfer.

Figure 4.10: Digital to Analog connector pin out

4.11 Pushbuttons and LED

The Eagle 100 comes standard with a user pushbutton, a reset push button, a user LED and a power LED. The user push button is connected to port A bit 6 with a 10k? pull-up resistor connected to it. The user LED is buffered through the CPLD and can be illuminated by setting port E bit 1 of the LM3S6918.

4.12 CPLD Programming Header

The CPLD comes preprogrammed from the factory with VHDL code that will function with the example programs and certified PC/104 expansion boards. The VHDL firmware can be updated in two ways. The first way is directly through the CPLD?s JTAG port which can be accessed through the combination of JP1 and JP2 as illustrated in Figure 4.12. The second method is by programming the LM3S6918 microcontroller with a utility that will program the CPLD through its SPI0 port. This can be done by placing jumper across pins 1&2, 3&4, 5&6, and 7&8 of JP1 as like in Figure 4.12.

Figure 4.12: CPLD programming jumper pin out (CPLD pin#)

4.13 PC/104 Expansion

The available signals on the PC/104 expansion connector are shown in Figure 4-13. The default VHDL firmware shipped with the Eagle SBC allows access to those signals as extended I/O ports. VHDL firmware is available on the Micromint web site that allows use of certified PC/104 expansion boards with the Eagle SBC. Using the signals for extended I/O will lead to higher I/O performance but the PC/104 mode allows use of off-the-shelf expansion boards. PC/104 connector

Figure 4.13: PC/104 connector pin out (CPLD pin#)

NEXT: References

PREVIOUS: Mechanical and Electrical Characteristics

5 Mechanical and Electrical Characteristics

5.1 Absolute Maximum Ratings

Operating Temperature		**Maximum Voltage**	
Commercial	0ºC to +70ºC	Voltage on J1 with LCD	+5.5 VDC Regulated
Industrial	-40°C to +85°C	Voltage on J1 without LCD	+15 VDC Regulated
Storage Temperature	-50ºC to +125ºC	Voltage on VBAT (JP7&J2)	+3.3 VDC Regulated

The Eagle SBC is currently available for commercial temperature ranges. Contact the Micromint sales department if you require support for industrial temperature ranges.

5.2 Mechanical Dimensions

DIM	Inches	Millimeters	DIM	Inches	Millimeters	DIM	Inches	Millimeters
A	4.4	111.76	I	1.15	29.21	Q	4.15	105.41
B	0.35	8.89	J	0.56	14.224	R	0.125	3.175
C	0.375	9.525	K	2.8	71.12	S	0.2	5.08
D	0.75	19.05	L	0.75	19.05	T	0.4	10.16
E	1.775	45.085	M	3.8	96.52	U	0.775	19.685
F	1.15	29.21	N	0.65	16.51	V	0.55	13.97

G	0.5	12.7	O	1.05	26.67	W	0.2	5.08
H	0.53	13.462	P	3.15	80.01			

Eagle 100 Mechanical Dimensions

NEXT: References

PREVIOUS: User Interfaces, Connectors, and Jumpers

6 References

This section outlines material that may be useful for further reading.

6.1 Documents

LM3S6918 Microcontroller Data Sheet

http://www.ti.com/product/lm3s6918

This data sheet provides reference information for the LM3S6918 microcontroller, describing the functional blocks of the system-on-chip (SoC) device designed around the ARM® Cortex?-M3 core. All MCU registers are described in the data sheet.

6.2 Books

The Definitive Guide to ARM® Cortex-M3 and Cortex-M4 Processors, Third Edition

by Joseph Yiu

ISBN: 0124080820 Publisher: Newnes (November, 2013)

Overview of the processor and instruction set architecture of the ARM® Cortex®-M3 and Cortex®-M4 processors. Several code examples using IAR, Keil, gcc and CooCox CoIDE.

The Designer's Guide to the Cortex-M Processor Family: A Tutorial Approach

by Trevor Martin

ISBN: 0080982964 Publisher: Newnes (May, 2009)

Tutorial-based book giving the key concepts required to develop programs in C with a Cortex M- based processor.

ARM System Developer's Guide: Designing and Optimizing System Software

ISBN: 1558608745 Publisher: Morgan Kaufmann; (March, 2004)

In-depth overview of the ARM architecture with examples that outline impact of programming practices on performance, power and cost.

Circuit Design with VHDL

by Volnei A. Pedroni

ISBN: 0262162245 Publisher: MIT Press (August, 2004)

Concise overview of the VHDL language and design concepts. Includes a large number of complete design examples with illustrative circuit diagrams and simulation results.

The Designers Guide to VHDL, Volume 3, Third Edition

by Peter J. Ashenden

ISBN: 0120887851 Publisher: Morgan Kaufmann (May, 2008)

Starts with the basics of VHDL and builds on to create registers and other logic subsystems. Continues with more complex projects such as a the DLX processor, a basic CPU implemented with VHDL.

6.3 Useful Web Links

Micromint Web Site

http://www.micromint.com/

Product information and software updates for the Eagle SBCs.

Texas Instruments Web Site

http://www.TI.com/

Manuals and application notes for the LM3S6918 microcontroller.

PREVIOUS: Mechanical and Electrical Characteristics