Table of Contents

1 Introduction

1.1 Lincoln Family of Controllers

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

1.2 Lincoln 60 Overview

The Lincoln 60 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 NXP ARM Cortex-M3 microcontroller, capable of operating at 120-MHz, the Lincoln 60 can fulfill demanding requirements in monitoring, instrumentation, data acquisition, process control, factory automation, DSP and many other applications. The 60E adds 100 Mbps Ethernet for high performance networking.

1.3 Lincoln 60 Features

- 120 MHz 32-bit ARM® Cortex?-M3
- 512 KB Flash/64 KB SRAM
- 10/100 Ethernet (60E Only)
- Micro-SD Socket
- Two RS232 Serial Ports
- I2C, SSI (SPI) Ports
- 8-ch. 12-bit ADC
- Up to 63 - 5V tolerant GPIOs, (up to 34 GPIOs on 50E)
- 4 general purpose timers, 1 watchdog timer
- Support for GNU and IAR compilers
- Thumb2 instruction set for smaller object code
- Basic and Lua also supported
- +5V@115mA (225mA 50E) typically
- Dimensions: 3.94inches x 2.83 inches (100mm x 72mm)

1.4 Lincoln 60 Options

- 4-ch. 12-bit DAC
- RS485 Serial Port
- 4x4 Keypad Port
- LCD Port
- USB Debug Port

1.5 Software and Support

Code examples are included with the Lincoln 60 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 and Nuttx. 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 Lincoln 60 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 Lincoln 60/60E can be done in just 6 steps.

- 1. Run the Preloaded Application
- 2. Install the Lincoln Code Examples
- 3. Choose a Compiler and IDE (Integrated Development Environment)
- 4. Install the Chosen Compiler and IDE
- 5. Compiling Code Examples
- 6. Downloading Code Examples to the Lincoln 60/60E

2.2 Run the Preloaded Firmware

The Lincoln 60E and Lincoln 60 come from the factory preloaded with applications. The Lincoln 60E has an Ethernet application called emac_uip that demonstrates a web-based interface. The Lincoln 60 has a micro SD card application called sd_card that demonstrates accessing a micro SD card through a serial command line. The full source code for both applications is provided on the Lincoln Tools Disk. The preloaded applications should be run to test the board after receiving it.

2.2.1 Lincoln 60E emac_uip Application

The Lincoln 60E SBC is shipped with the emac_uip example application that demonstrates a web-based interface using the Ethernet controller and the uIP TCP/IP Stack. DHCP is used to obtain an Ethernet address. Figure 2.1 shows the web page that will first be displayed.

Figure 2.1: Web interface for the emac_uip application
In order to demonstrate this application on an Lincoln 60E 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
- 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 emac_uip application on an Lincoln 60E without the optional USB Debug Port:

- 1. Make sure JP1 is set to JCK.
- 2. Connect the Lincoln 60Es COM1 port to the PC by using the 2x5 adapter cable and the null modem cable.
- 3. Connect the Lincoln 60Es Ethernet port to an Ethernet Network.
- 4. 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
- 5. Apply power to the board.
- 6. The Lincoln 60E will transmit the boards IP address to the terminal
- 7. Open a web browser.
- 8. Type the IP address into the web browsers address bar and the web page in Figure 2.1 should load.

In order to demonstrate the emac_uip application on an Lincoln 60E 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 1x5 female header (Supplied with Lincoln 60E with USB Debug Port)
- 5. One Ethernet cable

Please follow these steps to run the emac_uip application on an Lincoln 60E with the optional USB Debug Port:

- 1. Make sure JP1 is set to DBG.
- 2. Make sure JP3 and JP4 are set on pins 2 and 3 (FTDI).
- 3. Connect the Lincoln 60Es Ethernet port to an Ethernet Network.
- 4. Connect the Lincoln 60E to the PC by using the USB A to 1x5 female header.
- 5. Wait until the USB drivers have been installed. The USB drivers can be found on the Lincoln Tools Disk.

- 6. Find out what COM port the Lincoln 60E is by looking in Windows Device Manager?s PORTS (COM & LPT). The Lincoln 60Es COM port is the one that says Micromint Virtual COM Port.
- 7. Start and set-up the terminal program for the COM port that the Lincoln 60E 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
- 8. Press the Reset button.
- 9. The Lincoln 60E will transmit the boards IP address to the terminal.
- 10. Open a web browser.
- 11. Type the IP address into the web browsers address bar and the web page in Figure 2.1 should load.

2.2.2 Lincoln 60 sd_card Application

The Lincoln 60 SBC is shipped with the sd_card example application from the NXP CMSIS Library. This example application demonstrates reading a file system from a micro SD card. It makes use of FatFs, a FAT file system driver. The application provides a simple command console via the Lincoln 60?s COM1 serial port for issuing commands to view and navigate the file system on the SD card. Figure 2.2 shows what is displayed on a terminal program when the application is run.

Figure 2.2: Serial interface for the sd_card application
In order to demonstrate the sd_card application on an Lincoln 60 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
- 3. A null modem cable
- 4. A terminal program such as Putty, HyperTerminal, or TeraTerm
- 5. One micro SD card
- 6. One +5V power supply

Please follow these steps to run the sd_card application on an Lincoln 60 without the optional USB Debug Port:

- 1. Make sure JP1 is set to JCK.
- 2. Connect the Lincoln 60s COM1 port to the PC by using the 2x5 adapter cable and the null modem cable.
- 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. Insert the micro SD card into the micro SD socket on the Lincoln 60.
- 5. Apply power to the board.
- 6. Please refer to Figure 2.2 to see what the Lincoln 60 will transmit the terminal.
- 7. Type "?" then press enter for the command list.

In order to demonstrate the sd_card application on an Lincoln 60 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. One USB A to 1x5 header cable (Supplied with Lincoln 60E with USB Debug Port)
- 4. One micro SD card

Please follow these steps to run the sd_card application on an Lincoln 60 with the optional USB Debug Port:

- 1. Make sure JP1 is set to JCK.
- 2. Make sure JP3 and JP4 are set on pins 2 and 3 (FTDI).
- 3. Connect the Lincoln 60 to the PC by using the USB A to 1x5 header cable.
- 4. Wait until the USB drivers have been installed. The USB drivers can be found on the Lincoln Tools Disk.
- 5. Find out what COM port the Lincoln 60 is by looking in Windows Device Manager?s PORTS (COM & LPT). The Lincoln 60Es COM port is the one that says Micromint Virtual COM Port.
- 6. Start and set-up the terminal program for the COM port that the Lincoln 60 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. Please refer to Figure 2.2 to see what the Lincoln 60 will transmit the terminal.
- 9. Type "?" then press enter for the command list.

2.3 Install the Lincoln Code Examples

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

2.6.1 Compiling Code Examples with EWARM

To load the CMSIS workspace open the IAR IDE then "File>Open>Workspace" and browse to "%USERPROFILE%\Projects\Micromint\Lincoln" and select the "lincoln.eww" file. A screen similar to that in Figure 2.5 should load. Select the Blinky project by clicking on the drop down menu. Be sure to select the board that you are using. In the case of Figure 2.5 the Lincoln 60E was being used. To rebuild the project select the "Make" button on the toolbar. You can also right click on the project name to select "Make" or "Rebuild". This will build a binary (.BIN) image file in the "%USERPROFILE%\Projects\Micromint\Lincoln\CMSIS\projects\blinky\lincoln-all\ewarm\Exe" directory of the project. To load the application on the board please see the Firmware Download Procedures section in this manual. If the build and download is successful, the user LED on the Lincoln SBC will start blinking.

Figure 2.5: Using CMSIS projects with the IAR EWARM IDE

2.6.2 Compiling Code Examples with the GNU Toolchain

2.6.2.1 Compiling Code Examples with the Code::Blocks IDE

2.6.2.2 Compileing Code Exampels with the Eclipse IDE

To load the CMSIS workspace in the Eclipse IDE, select the workspace on the "%USERPROFILE%\My Documents\Projects\Micromint\Lincoln\CMSIS\projects" subdirectory when Eclipse IDE starts. You will see a screen similar to that on Figure 2.6. Select the Blinky project by right clicking the project name and selecting "Open Project". The first time the Lincoln BSP is installed the files need to be refreshed. To refresh the files just simply press F5. The default compiler for the Lincoln BSP is Sourcery G++ Lite. If you installed devkitARM instead then the compiler will have to be switched. To change the compiler to devkitARM click on "Window>Preferences>C/C++>Build>Environtment" and add a variable called "COMPILER" with the value set to "devkitARM". This will have to be done on all of the example projects. To build the project select the "build" button on the toolbar. You can also right click on the project name to select "Build" . This will build a binary (.BIN) image file in the "board type\gcc\" directory of the project. For example the binary for the blinky project using the Lincoln 60E is in the "%USERPROFILE%\Projects\Micromint\Lincoln\CMSIS\projects\blinky\lincoln-all\gcc" directory if the Lincoln BSP was installed in the default directory. To load the application on the board please see the Firmware Download Procedures section in this manual. If the build and download is successful, the user LED on the Lincoln SBC will start blinking.

Figure 2.6: Using the CMSIS projects with the Eclipse IDE

2.6.2.3 Compiling Applications 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 Lincoln Tools Disk. 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 %USERPROFILE%\Projects\Micromint\Lincoln\CMSIS\projects\blinky".
- 3. Press "Enter".
- 4. Type "make".
- 5. Press "Enter".

2.7 Downloading Code Examples to the Lincoln 60/60E

There are several options to download firmware to the Lincoln SBC. This section covers firmware downloads using a J-Link debugger or a USB Debugger. The method used to download firmware will depend on what hardware is being used.

2.7.1 Downloading Code Examples using EWARM

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 Debugger or picoJTAG and EWARM

The optional Debugger has the same set-up as the picoJTAG. Please see the picoJTAGs Wiki on how to download code examples toe the Lincoln 60 or 60E: 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 optional Debugger or 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 Flash Magic

The Lincoln 60 and Lincoln 60E can be used with the Flash Magic software to program the board. This software can be downloaded from http://www.flashmagictool.com/. After installing it, select Programs> Flash Magic> Flash Magic from your Windows Start menu. Go to the "Setup" tab to see a screen similar to that in Figure 2.7. On "Device" select "LPC1769". Set the baudrate to 115200 bps, ISP interface and 12 MHz oscillator.

The application's HEX file can be downloaded using the FlashMagic using the optional USB Debugger by following these steps:

- 1. Select the COM port that corresponds to the USB Debugger
- 2. Make sure JP3 and JP4 are set to pins 2 and 3
- 3. Browse for the hex file that will be programmed into the Lincoln 60
- 4. Press and hold the 'USER' button on the Lincoln 60
- 5. Press and release the 'RESET' button on the Lincoln 60
- 6. Release the 'USER' button.
- 7. Click the "Start" button on Flash Magic
- 8. Press and release the 'RESET' button on the Lincoln 60 to start running the application

The application's HEX file can be downloaded using the FlashMagic using COM1 by following these steps:

- 1. Select the COM port that corresponds to the one connected to COM1
- 2. Make sure JP3 and JP4 are set to pins 1 and 2
- 3. Browse for the hex file that will be programmed into the Lincoln 60
- 4. Press and hold the 'USER' button on the Lincoln 60
- 5. Press and release the 'RESET' button on the Lincoln 60
- 6. Release the 'USER' button.
- 7. Click the "Start" button on Flash Magic
- 8. Press and release the 'RESET' button on the Lincoln 60 to start running the application

Figure 2.7: Flash Magic download

2.8 Restoring the Factory Default Firmware

NEXT: Hardware

PREVIOUS: Introduction

3 Hardware

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

Lincoln 60 Hardware

3.1 Microcontroller

The Lincoln 60 includes a NXP LPC1769 microcontroller. These 32-bit ARM Cortex-M3 RISC microcontroller are capable of 120-MHz operation with a Thumb2 instruction set for smaller object code. It uses a Harvard architecture with separate local instruction and data buses as well as a separate peripherals bus. Please see NXP?s LPC1769 Microcontroller Data Sheet for more information and register definitions.

LPC1769 key features:

- Internal Memory
    - 512 kilo-bytes flash
    - Flash memory accelerator
    - In-System Programming (ISP) and In-Application Programming (IAP)
    - 64 kilo-bytes SRAM
    - Memory Protection Unit (MPU)
- General Purpose DMA controller (GPDMA)
    - Eight channels that support unidirectional transfers
    - Memory-to-memory, memory-to-peripheral, and peripheral-to-memory transfers
    - Supports SSP, I2S, UART, A\D Converter, and D/A Converter peripherals
    - Supports 8-bit, 16-bit, and 32-bit wide transactions
    - Internal four-word FIFO per channel
- General Purpose 32-bit Timer/Counters
    - Four channels
    - Programmable 32-bit Prescaler
    - Eight capture inputs
    - Ten compare outputs
    - Generate DMA requests
- Additional Timers
    - Watchdog Timer
    - System Tick Timer
    - Repetitive Interrupt Timer (RIT)
- Ethernet MAC
    - Fully compliant with IEEE standard 802.3
    - VLAN frame support
    - Independent transmit and receive buffers memory mapped to shared SRAM

- ♦ Dedicated DMA controller
- ♦ Wake-on-LAN power management support

• USB 2.0 Controller
- ♦ Host, Device, or On-The-Go (OTG)
- ♦ Integrated PHY
- ♦ Dedicated DMA

• UART
- ♦ Data sizes of 5, 6, 7, and 8 bits
- ♦ Parity generation and checking: odd, even mark, space or none
- ♦ One or two stop bits
- ♦ 16 byte receive and transmit FIFOs
- ♦ Baud rate generator, including fractional rate divider
- ♦ Auto-baud capability
- ♦ Supports DMA for both transmit and receive

• Synchronous Peripheral Interface (SPI)
- ♦ Synchronous, Serial, Full Duplex Communication
- ♦ SPI master or slave
- ♦ 8 to 16 bits per transfer

• Synchronous Serial Port (SSP)
- ♦ Compatible with Motorola SPI, 4-wire IT SSI, and National Semiconductors Microwire
- ♦ Master or slave operation
- ♦ 8 frame FIFOs for both transmit and receive
- ♦ 4 to 16 bit data frame
- ♦ DMA transfers supported

• I2C Bus
- ♦ Three channels
- ♦ Master, Slave, or Master/Slave
- ♦ Programmable clock

• I2S Bus
- ♦ Master or slave operation
- ♦ Capable of handling 8-bit, 16-bit, and 32-bit word sizes
- ♦ Mono and stereo audio supported
- ♦ Two DMA requests, controlled by buffer levels.

• Analog-to-Digital Converter
- ♦ Eight 12-bit channels
- ♦ 12-bit conversion rate of 200 kHz
- ♦ Burst conversion mode for single or multiple inputs
- ♦ DMA transfer supported

• Digital-to-Analog Converter
- ♦ One 10-bit channel
- ♦ Resistor string architecture
- ♦ Buffered output
- ♦ Maximum update rate of 1 MHz
- ♦ DMA transfer supported

• General Purpose Input or Outputs (GPIO)
- ♦ 5V tolerant inputs
- ♦ Enable or disable internal pull-up or pull-down resistors
- ♦ Programmable open drain mode
- ♦ Port 0 and Port 2 can provide a single interrupt for any combination of port pins

• Reset Sources
- ♦ Power on reset
- ♦ Reset pin assertion
- ♦ Brown out reset
- ♦ Watchdog timer reset

• Additional Features
- ♦ Debugging via JTAG or Serial Wire interfaces
- ♦ Code Read Protection (CRP) with different security levels
- ♦ Unique device serial number for identification
- ♦ Programmable PLL for system clock

Lincoln 60 Memory Map

3.2 Power Supply

The Lincoln 60's power supply is designed with Diodes Incorporated's AP6015 high efficiency step-down switching power supply. The power supplies input voltage can come from the Lincoln 60's USB device port, optional USB debug port, or J1 connector. A diode (D1) will protect the Lincoln 60 should polarity of the power supply be reversed on the J1 connector. The power supply is capable of outputting 3.3 VDC with a maximum output current of 800 mA. It has an input operating voltage range of 3.9 VDC to 5.8 VDC. Most of the circuitry on the Lincoln 60 operates from the 3.3V output with typical current requirements for the Lincoln 60 of ???mA and ???mA for the Lincoln 60E. A USB device being powered from the Lincoln 60's USB Host port, the LCD, and the optional CAN driver are the only devices that need more than 3.3 VDC to operate. A diode drop of 0.3 VDC and a maximum of 2.0 A through D1 must be taken into consideration when selecting a power supply for J1.

3.3 PHY

The Lincoln 60E includes a Micrel KSZ8041NL 10 Base-T/100 Base-TX Physical Layer Transceiver (PHY). The PHY provides MII/RMII interface to transmit and receive data. It has HP Auto MDI/MDI-X to eliminate the need to differentiate between crossover and straight-through cables. For further information please see Micrel?s KSZ8041NL Data Sheet.

3.4 RS-232 Driver

The Lincoln 60 includes RS-232 drivers that consist of two transmitters, two line receivers, and a dual charge-pump Circuit. The driver provides the electrical interface between the microcontroller?s UART and the serial-port connector. The devices operates at data signaling rates up to 500 kbps.

3.5 DAC (Optional)

The Lincoln 60 optionally comes with a National Semiconductor?s DAC124S085 general purpose digital-to-analog converter (DAC). The DAC has four channels with a resolution of 12-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.

3.6 CAN Driver (Optional)

The Lincoln 60 optionally comes with a CAN transceiver from Maxim Semiconductors. It provides the interface between the microcontroller?s CAN port and the CAN port?s connector. The MAX3059ASA+ CAN transceiver had a maximum data rate of 1Mbps. It features three different modes of operation high speed, slope control, and shutdown. For further information please see Maxim?s MAX3058-MAX3059 Data Sheet.

3.7 RS-485 Driver (Optional)

The Lincoln 60 optionally comes with a RS-485 driver half-duplex transceiver. It provides the electrical interface between the microcontroller?s UART and the RS-485 port?s screw terminal. The half duplex transceiver is suitable for data rates up to 10Mbps over long twisted pairs.

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 Lincoln 60.

Lincoln 60 User Interfaces, Connectors, and Jumpers

4.1 Power Supply

The main power for the Lincoln 60 SBC can be from the J1 connector, the USB device port, or by the debugger (external or onboard) depending on how JP1 is set. Figure 4.1 is the pin-out for JP1 and shows the board being powered by J1. The typical current requirements are 60 mA for the Lincoln 60 and 160 mA for the Lincoln 60E.

Figure 4.1: Power supply selection jumper

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. Figure 4.2 shows the different J1 configurations. A diode (D1) will protect the Lincoln 60 should polarity of the power supply be reversed on the J1 connector. The protection diode has a voltage drop of 0.3V and is limited to a maximum of 2 amperes through it. The specifications for D1 must be taken into consideration when selecting a power supply while using the LCD port, USB Host port, and CAN port.

Figure 4.2: Power supply connector configurations

The Lincoln 60 may also be powered through the USB Device port (J16) by changing the jumper on JP1 to pins 1 and 2. Current draw will be limited to the USB host port that is supplying the power. Most computers? USB host ports allow a maximum current draw of 500mA. Setting JP1 to pins 2 and 3 will allow the board to be powered through the optional USB debug port (J15) or through the JTAG connector (J5). The maximum available current for the board will be limited to the USB host port that is supplying the power to the optional USB debug port. A maximum of 500mA total current draw is allowed when powering the board through the JTAG connector or is limited by the device supplying the power. The J-Link for example only allows 300mA of current.

4.2 10/100 Ethernet (60E Only)

The Lincoln 60E is equipped with a fully-integrated 10/100 Mbps Ethernet port. The Media Access Control (MAC) is implemented in the LPC1769 and the Physical (PHY) layer is implemented with Micrel?s KSZ8041NL. J6 is the RJ-45 connector and it has integrated magnetics and LEDs completes the Ethernet sub-system. Please see the KSZ8041NL data sheet for further information on the PHY and the LPC1769 data sheet for the MAC.

4.3 Serial (COM) Ports

The LPC1769 has a total of 4 Universal Asynchronous Receivers/Transmitters (UART). Two of the UARTs are are level shifted to RS-232 levels. UART0 (COM1) and UART3 (COM2) can be accessed 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) and are considered to be Data Terminal Equipment(DTE). In order to communicate to a Personal Computer a null modem cable is required. 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 Ports Connector Pin Outs

The transmitter for COM1 is port 0 bit 2 and the receiver is port 0 bit 3 on the LPC1769 microcontroller. If the optional USB Debug port is populated UART0 may be configured to communicate over USB by placing moving jumpers JP3 and JP4 to pins 2&3. UART0 can be accessed through the USB Debug Ports 1x5 pin header (J15). See figure 4.4 for shows the jumper settings selecting RS-232 levels or for COM1. The transmitter for COM2 is port 0 bit 0 and the receiver is port 0 bit 1 on the LPC1769.

Figure 4.4: COM1 Driver Selection Jumpers

A third UART is level shifted to RS-485 levels if the I/O plus option is populated. UART1 (COM3) can be accessed through screw terminals T1. The RS-485 transmitter is enabled by making port 2 bit 7, on the LPC1769, a logic 1. The transmitter for COM3 is port 2 bit 0 and the receiver is port 2 bit 1 on the LPC1769. The LPC1769's UART1 supports RS-485 modes of operation. Please consult the LPC1769 User's Manual for further details. The RS-485 network can be terminated with a 120 ohm resistor by placing a jumper on JP5. See figure 4.3 for the pin out of COM3.

The LPC1769 also has two Controller Area Network (CAN) ports avaialble. The I/O plus option populates a CAN tranciever for the CAN1 controller (COM4). COM4 can be accessed through a 2x5 pin berg header. See Figure 4.3 for the pin out of the COM4 (J13) connector. The transmitter for COM4 is port 0 bit 22 and the receiver is port 0 bit 21 on the LPC1769. The CAN network can be terminated with a 120 ohm resistor by placing a jumper on JP6.

4.4 General Purpose Digital Inputs and Outputs

There are seventy total bits of GPIO available on the Lincoln 60 and forty-four bits available on the Lincoln 60E. Twenty-six bits of GPIO are available on J2 for both the Lincoln 60 and Lincoln 60E. Twenty-seven bits of GPIO are available on J12 only for the Lincoln 60. J12 is not populated when the board is configured as a Lincoln 60E due to most of the GPIO being used for the Ethernet PHY signals. Nine bits of GPIO are available on J18 but it is not populated because the GPIO are used for on board perphrials such as COM2, USB Host, USB Device, User LEDs, User Pushbutton, and microSD. Eight bits of GPIO are available on J7. Please see the ADC/GPIO section for further details. Figure 4.5 has the pin outs for J2, J12, and J18.

Figure 4.5: GPIO Connector Pin Outs

Some of the GPIO on J2 and J12 have alternate functions other than digital inputs and outputs and are shared with some of the hardware on the board. Table 4.1 lists the alternate functions, the hardware it is shared with, and a brief description of the alternate function for connector J2. Table 4.2 lists the alternate functions, the hardware it is shared with, and a brief description of the alternate function for connector J14. Table 4.3 lists the alternate functions, the hardware it is shared with, and a brief description of the alternate function for connector J18. For further information on the alternate functions please refer to the LPC1769 data sheet.

J2 Pin#	Signal	Alternate Functions	Notes	Shared Hardware
1	GND		Digital Ground	
2	VCC		3.3 VDC	
3	P0.4	I2SRX_CLK/RD2/CAP2.0	I2S Receive Clock/CAN2 Receive/Timer 2 Capture Input 0	
4	P0.5	I2SRX_WS/TD2/CAP2.1	I2S Receive Word Select/CAN2 Transmit/Timer 2 Capture Input 1	
5	P0.6	I2SRX_SDA/SSEL1/MAT2.0	I2S Receive Data/SSP1 Slave Select/Timer 2 Match Output 0	
6	P0.7	I2STX_CLK/SCK1/MAT2.1	I2S Transmit Clock/SSP1 Serial Clock/Timer 2 Match Output 1	
7	P0.8	I2STX_WS/MISO1/MAT2.2	I2S Transmit Word Select/SSP1 Master In Slave Out/Timer 2 Match Output 2	
8	P0.9	I2STX_SDA/MOSI1/MAT2.4	I2S Transmit Data/SSP1 Master Out Slave /Timer 2 Match Output 3	
9	P0.10	TXD2/SDA2/MAT3.0	UART 2 Transmitter/I2C2 Data/Timer 3 Match Output 0	
10	P0.11	RXD2/SCL2/MAT3.1	UART2 Receiver/I2C2 Clock/Timer 3 Match Output 1	
11	P0.27	SDA0	I2C0 Data	Keypad(J11)
12	P0.28	SCL0	I2C0 Clock	Keypad(J11)
13	P1.22	MC0B/MAT1.0	Motor Control PWM0 Ouput B/Timer 1 Match Output 0	LCD(J10)
14	P1.23	MCFB1/PWM1.4/MISO0	Motor Control PWM1 Feedback Input/PWM1 Output 4 output/Master In slave Out	LCD(J10)
15	P1.24	MCFB2/PWM1.5/MOSI0	Motor Control PWM2 Feedback Input/PWM1 Output 5/SSP0 Master Out Slave In	LCD(J10)
16	P1.25	MC1A/MAT1.1	Motor Control PWM1 Output A/Timer 1 Match Output 1	LCD(J10)
17	P1.26	MC1B/PWM1.6/CAP0.0	Motor Control PWM1 Output B/PWM1 Output 6/Timer 0 Capture Input 0	LCD(J10)
18	P1.27	CLKOUT/CAP0.1	Clock Output Pin/Timer 0 Capture Input 1	LCD(J10)
19	P1.28	MC2A1.0/PCAP1.1/MAT0.0	Motor Control PWM2 Output A/PWM1 Capture Input 0/Timer 0 Match Output 0	LCD(J10)
20	P1.29	MC2B/PCAP1.1/MAT0.1	Motor Control PWM2 Output B/PWM1 Capture Input 1/Timer 0 Match Output 1	LCD(J10)
21	P2.2	PWM1.3/CTS1/TRACEDATA.3	PWM1 Output 3/UART1 Clear to Send/Trace Dat bit 3	Keypad(J11)
22	P2.3	PWM1.4/DCD1/TRACEDATA.2	PWM1 Output 4/UART1 Data Carrier Detect/Trace Dat bit 2	Keypad(J11)
23	P2.4	PWM1.5/DSR1/TRACEDATA.1	PWM1 Output 5/UART1 Data Set Ready/Trace Dat bit 1	Keypad(J11)

24	P2.5	PWM1.6/DTR1/TRACEDATA.0	PWM1 Output 6/UART1 Data Terminal Ready/Trace Dat bit 0	Keypad(J11)
25	P2.11	EINT1/I2STX_CLK	External Interrupt 1/I2S Transmit Clock	
26	P2.12	EINT2/I2STX_WS	External Interrupt 2/I2S Transmit Word Select	
27	P4.28	RX_MCLK/MAT2.0/TXD3	I2S Receive Master Clock/Timer 2 Capture Input 0/UART3 Transmitter	
28	P4.29	TX_MCLK/MAT2.1/RXD3	I2S Transmit Master Clock/Timer 2 Capture Input 1/UART3 Receiver	
29	RST		Microcontroller Reset	
30	+5V		+5 VDC	

Table 4.1: Alternate functions for GPIO on J2

J12 Pin#	Signal	Alternate Functions	Notes	Shared Hardware
1	GND		Digital Ground	
2	VCC		3.3 VDC	
3	P0.15	TXD1/SCK0/SCK	UART1 Transmitter/SSP0 Clock/SPI Clock	microSD/DAC
4	P0.17	CTS1/MISO0/MISO	UART1 Clear to Send/SSP0 Master In Slave Out/SPI Master In Slave Out	microSD
5	P0.18	DCD1/MOSI0/MOSI	UART1 Data Carrier Detect/SSP0 Master Out Slave In/SPI Master Out Slave In	microSD/DAC
6	P0.21[1]	RI1/RD1	UART1 Ring Indicator/CAN1 Receiver	CAN
7	P1.0	ENET_TXD0	Ethernet Transmit Data 0	Ethernet PHY
8	P0.22	RTS1/TD1	UART1 Request to Send/CAN1 Transmitter	CAN
9	P1.1	ENET_TXD1	Ethernet transmit data 1	Ethernet PHY
10	P1.4	ENET_TX_EN	Ethernet transmit data enable	Ethernet PHY
11	P1.8	ENET_CRS	Ethernet carrier sense	Ethernet PHY
12	P1.9	ENET_RXD0	Ethernet receive data	Ethernet PHY
13	P1.15	ENET_REF_CLK	Ethernet reference clock	Ethernet PHY
14	P1.10	ENET_RXD1		Ethernet PHY
15	P1.17	ENET_MDIO	Ethernet MIIM data input and output	Ethernet PHY
16	P1.14	ENET_RX_ER	Ethernet receive error	Ethernet PHY
17	P0.19	DSR1/SDA1	UART1 Data Set Ready/I2C1 Data	Keypad(J11)
18	P1.16	ENET_MDC	Ethernet MIIM clock	Ethernet PHY
19	P0.20	DTR1/SCL1	UART1 Data Terminal Ready/I2C1 Clock	Keypad(J11)
20	P1.19	MC0A/CAP1.1	Motor Control PWM0 Output A/Timer 1 Capture Input 1	LCD(J10)
21	P1.20	MCFB0/PWM1.2/SCK0		LCD(J10)

			Motor Control PWM0 Feedback Input/PWM1 Output 2/SSP0 Clock	
22	P1.21	MCABORT/PWM1.3/SSEL0	Motor Control PWM Emergency Abort/PWM1 Output 3/SSP0 Slave Select	LCD(J10)
23	P2.0	PWM1.1/TXD1	PWM1 Output 1/UART1 Transmitter	COM3 (RS485)
24	P2.1(2)	PWM1.2/RXD1	PWM1 Output 2/UART1 Receiver	COM3 (RS485)
25	P2.6	PCAP1.0/RI1/TRACECLK	PWM1 Capture Input 0/UART1 Ring Indicator/Trace Clock	LCD(J10)
26	P2.7	RD2/RTS1	CAN2 Receiver/UART1 Request to Send	COM3 (RS485)
27	P2.8	TD2/TXD2	CAN2 Transmitter/UART2 Transmitter	DAC
28	VBAT		Supply pin for LPC1769\'s RTC	Coin Battery
29	P2.13	EINT3/I2STX_SDA	External Interrupt 3/I2S Transmit Data	
30	P3.25	MAT0.0/PWM1.2	Timer 0 Match Output 0/ PWM1 Output 2	User Buzzer

Table 4.2: GPIO alternate functions for J12

(1) - P0.21 should not be used as a GPIO when the CAN tranciever (U7) is populated due to the recevier driving the pin.

(2) - P2.1 should not be used as a GPIO when the RS-485 tranceiver (U5) is populated due to the receiver driving the pin.

J18 Pin#	Signal	Alternate Functions	Notes	Shared Hardware
1	GND		Digital Ground	
2	VCC		3.3 VDC	
3	P0.1	TD1/RXD3/SCL1	CAN1 Transmitter/UART3 Receiver/I2C1 Clock	COM2
4	P0.0	RD1/TXD3/SDA1	CAN1 Receiver/UART3 Transmitter/I2C1 Data	COM2
5	P0.16	RXD1/SSEL0/SSEL	UART1 Receiver/SSP0 Slave Select/ SPI Slave Select	microSD
6	P2.9	USB_CONNECT/RXD2	USB Device Soft Connect Feature/UART2 Receiver	USB Device
7	P1.18	USB_UP_LED/PWM1.1/CAP1.0	USB Good Link LED Indicator/PWM1 Output 1/Timer 1 Capture Input 0	USER LED1
8	P0.30	USB_D-	USB D- Line	USB Host/Device
9	P0.29	USB_D+	USB D+ Line	USB Host/Device
10	P2.10	EINT0/NMI	External Interrupt 0/Non-maskable Interrupt	USER PB
11	EN		Switching Power Supply Enable	
12	P3.26	STCLK/MAT0.1/PWM1.3	System Tick Timer Clock Input/Timer 0 Match Output 1/PWM1 Output 3	USER LED2

Table 4.3: GPIO alternate functions for J18

4.5 ADC/GPIO

The Lincoln 60?s eight channels of 12-bit Analog to Digital Converter (ADC) can be connected to through J7. See figure 4.6 for the pin out of the ADC connector. The ADC is accessed directly through the LPC1769 microcontroller. It is capable of 200k samples/second and can be triggered to read through software, timers, or GPIO. Each channel of the ADC are multiplexed with digital inputs and outputs. Table 4.4 lists the alternate functions, the hardware it is shared with, and a brief description of the alternate function for connector J7. AD0.6 and AD0.7 are used for COM1. If an application needs to have eight channels of ADC then COM1 can be disconnected from J7 by removing R68 and R69 from the bottom of the board by JP3 and JP4.

Figure 4.6: Analog to Digital connector pin out

J7 Pin#	Signal	Alternate Functions	Notes	Shared Hardware
1	P0.23	AD0.0/I2SRX_CLK	Analog to Digital Converter 0 Channel 0/I2S Receive Clock	
2	P1.30	AD0.4/VBUS	Analog to Digital Converter 0 Channel 4/Monitors Presence of USB Device Power	
3	P0.24	AD0.1/I2SRX_WS/CAP3.1	Analog to Digital Converter 0 Channel 1/I2S Receive Word Select/Timer 3 Capture Input 1	
4	P1.31	AD0.5/SCK1	Analog to Digital Converter 0 Channel 5/SSP1 Clock	
5	P0.25	AD0.2/TXD3	Analog to Digital Converter 0 Channel 2/UART3 Transmitter	
6	P0.3	AD0.6/RXD0	Analog to Digital Converter 0 Channel 6/UART0 Receiver	COM1
7	P0.26	AD0.3/AOUT/RXD3	Analog to Digital Converter 0 Channel 3/DAC Output/UART3 Receiver	
8	P0.2	AD0.7/TXD0	Analog to Digital Converter 0 Channel 7/UART0 Transmitter	COM1
9	AGND		Analog Ground	
10	AGND		Analog Ground	

Table 4.4: GPIO alternate functions for J7

4.6 DAC (Option)

The Lincoln 60?s four channels of 12-bit DAC can be connected to through J8. Please refer to figure 4.10 for the pin out of the DAC connector. It is capable of outputting voltages between 0 and 3.3V. The DAC is accessed through the LPC1769's SSP0 port. Port 2 bit 8 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 50 MHz which is a total of 320 nS per data transfer but the DAC has a settling time of 6 µS so the total time needed is 6.32 µS. All 4 channels of the DAC can be updated at a maximum of 137 kHz. This is accomplished by sending the data for all 4 channels and updating all of the outputs on the last data transfer.

Figure 4.7: Digital to Analog connector pin out

4.7 Keypad (Option)

A 4x4 matrix keypad using a 16-pin (2x8) ribbon cable can be connected to port 1 and port 2 of the microcontroller through J11. Figure 4.8 is the pin out for the keypad connector.

Figure 4.8: Keypad connector pin out

4.8 Liquid Crystal Display (LCD)(Optional)

A standard alphanumeric LCD may be connected to J10 through a 32-pin (2x16) ribbon cable. Port 1.22 through P1.29 are used for the LCD?s data bus. The LCD?s control signals are connected to P1.19 through P1.21 the microcontroller. P1.19 controls the LCDs register select, P1.20 controls the read/write signal and P1.21 controls the enable signal. The backlight is controlled by P2.6. If port 2 bit 6 is logic 1 then the backlight will be illuminated. The contrast for the LCD may be adjusted by turning potentiometer R12 located next to J10. Please see figure 4.9 for the LCD?s connector pin out. AZ Displays ACM2004D series is recommended for use with the Lincoln 60/60E. http://www.azdisplays.com/index.php?id=Character_Modules&product=c2004d

Figure 4.9: LCD connector pin out

4.9 Micro-SD

The microSD socket (J9) enables micro-secure-digital memory cards to be plugged into the Lincoln 60 microcontroller board. The microSD card allows the user the ability of a standard removable media for transferring data to and from the Lincoln 60. The LPC1769 interfaces to the microSD card through the SSP0 port.

4.10 Pushbuttons and LED

The Lincoln 60 comes standard with a user pushbutton, a reset push button, and two user LEDs. The user push button is connected to port 2 bit 10 with a 22k-ohm pull-up resistor connected to it. User LED1 can be illuminated by clearing port 1 bit 18 of the LPC1769. User LED2 can be illuminated by clearing port 3 bit 26.

The LPC1769 has a serial bootloader in ROM when it is shipped from the factory. The serial bootloader can be initiated by pressing and holding the user pushbutton while pressing and releasing the reset button. This allows the user to update the application through COM1.

4.11 Piezo Buzzer (Option)

The I/O options include a Piezo buzzer that can be used for audible alarms. The buzzer needs a square wave output from Port 3 bit 25 on the LPC1769 in order to generate an alarm. A 4 kHz square wave will generate 70db. P3.25 can be configured as output 2 for PWM1 to generate the square wave needed without creating overhead for the MCU.

4.12 Battery Backed RTC (Option)

The Lincoln 60's microcontroller has a built in real-time clock calendar that can be battery backed by supplying 2.1 VDC to 3.3 VDC to the VBAT pin on the LPC1769. A coin cell battery holder is populated on the bottom of the board with the I/O plus options and holds a CR2032 series coin cell. Power is only drawn from the battery when the power is off to the Lincoln 60.

4.13 JTAG (J5) and USB Debug Port (Option)

The JTAG port (J5) 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. Figure 4.10 shows the pin out for J5.

Figure 4.10A: JTAG connector pin out Board Revision A

Figure 4.10B: JTAG connector pin out Board Revision B

4.14 USB Debug Port (Option) Board Revision A

Debugging and program updates can also be performed using the optional USB debug port accessible through a 1x5 berg header (J15). The USB debug port is supported by the following development tools:

- IAR EWARM IDE
- OpenOCD

The USB debug port also allows UART0 to communicate over USB. Please see section 4.3 for further information. Figure 4.11 shows the pin out for the JTAG connector when the USB Debug option is populated. Figure 4.12 shows the pin out for the USB debug port's connector.

Figure 4.11: JTAG connector pin out with Debug Port Populated Rev A

Figure 4.12: USB Debug connector pin out

4.15 MBED HDK/CMSIS DAP (Option) Board Revision B

The MBED HDK is powered by NXP's LPC11U35 microcontroller. The MBED HDK is compliant with the USB V2.0 full-speed device specification. It's connector is a micro USB Type AB and is designated J15.

The MBED HDK also allows UART0 to communicate over USB. Please see section 4.3 for further information.

4.16 Default Jumper Settings

Figure 4.13 and 4.14 show the default jumper configuration for the Lincoln 60. All boards without the optional USB debug port will come from the factory with the jumpers set like Figure 4.13. The USB Host port will be enabled and power is expected to come from J1. When the USB debug port is populated the jumpers will be set like Figure 4.14. The USB Host port will be enabled, power is expected to come from J1, and COM1 will be available through J3.

Figure 4.13: Lincoln 60 Base Configuration Default Jumper Settings

Figure 4.14: Lincoln 60 with USB Debug Default Jumper Settings

NEXT: Mechanical and Electrical Characteristics

PREVIOUS: Hardware

5 Mechanical and Electrical Characteristics

5.1 Absolute Minimum and Maximum Ratings

Characteristic	Minimum	Maximum	Unit
Voltage on J1 without LCD	3.9	5.8	VDC
Voltage on J1 with LCD	4.8	5.8	VDC
Voltage on VBAT (B1 or J12)	2.1	3.6	VDC
Voltage on ADC	0.0	3.3	VDC
Voltage on Digital Input	0.0	5.0	VDC
Operating Temperature	0	70	ºC
Storage Temperature	-50	125	ºC

The Lincoln 60 SBC is currently available for commercial temperature ranges. Contact the Micromint sales department if you require support for industrial temperature ranges.

5.2 Supply Current Versus System Clock Frequency

Test Conditions: Vin (J1) = 5.63V, VCC = 3.3V

5.3 Mechanical Dimensions

Below is the physical dimensions for the Lincoln 60. The mounting holes will accept a #4 size screw.

DIM	Inches	Millimeters	DIM	Inches	Millimeters	DIM	Inches	Millimeters	DIM	Inches	Millimeters
A	3.94	100.0	**G**	0.266	6.75	**M**	0.71	18.03	**S**	2.835	72.0
B	0.116	2.95	**H**	0.275	6.98	**N**	0.58	14.73	**T**	0.115	2.92
C	0.851	21.61	**I**	0.124	3.14	**O**	0.369	9.37	**U**	0.895	22.73
D	0.567	14.4	**J**	0.4	10.16	**P**	0.117	2.97	**V**	0.3	7.62
E	0.828	21.03	**K**	0.265	6.74	**Q**	0.12	3.04	**W**	0.608	15.44
F	0.2	5.08	**L**	0.63	16.0	**R**	0.950	24.13	**X**	0.76	19.3

Lincoln 60 Mechanical Dimensions

DIM	Inches	Millimeters	DIM	Inches	Millimeters	DIM	Inches	Millimeters	DIM	Inches	Millimeters
A	0.33234	8.4415	**F**	0.588	14.935	**K**	0.261	6.63	**P**	0.136	3.45
B	0.3543	9.0	**G**	0.072	1.8171	**L**	0.06	1.524	**Q**	0.269	6.83
C	0.117	2.97	**H**	0.56693	14.40	**M**	0.545	13.843	**R**	0.177	4.5
D	0.345	8.763	**I**	0.072	1.83	**N**	0.1	2.54	**S**	0.256	6.5
E	0.599	15.21	**J**	0.63	16.0	**O**	0.288	7.313			

Lincoln 60 Suggested Openings

NEXT: References

PREVIOUS: User Interfaces, Connectors, and Jumpers

6 References

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

6.1 Documents

LPC1769 Microcontroller Data Sheet

http://www.nxp.com/documents/data_sheet/LPC1769_68_67_66_65_64.pdf

This data sheet provides reference information for the NXP1769 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.

IEEE 1149.1 JTAG Standard

http://www.jtag.com/en/Learn/Standards/IEEE_1149.1

This website provides reference information for the IEEE 1149.1 JTAG standard. It describes the standard and has a couple of tutorials demonstrating the standard.

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.

6.3 Useful Web Links

Micromint Web Site

http://www.micromint.com/

Product information and software updates for the Lincoln SBCs.

NXP?s Web Site

http://www.nxp.com/

Manuals, Erratas, and application notes for the LPC1769.

NEXT: Appendix A - Updating CMSIS-DAP Firmware

PREVIOUS: Mechanical and Electrical Characteristics

7 Appendix A

7.1 Appendix A - Updating CMSIS-DAP Firmware

- 1. Locate the updated CMSIS-DAP binary file
- 2. Boot the MBED HDK into CRP DISABLED mode by shorting JP3 while connecting the USB1 Cable. A

  drive should appear.

- 3. Program the Interface firmware by deleting the firmware.bin file and copying the new binary to the CRP DISABLED disk.
- 4. Reboot the Interface by disconnecting and reconnecting the USB1 cable without JP3 shorted
- 5. Verify that the MBED disk comes up
- 6. Finished!

PREVIOUS: References