Table of Contents

1 Introduction

1.1 Electrum Family of Controllers

The Electrum series of single board computers run on an ARM9 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 Electrum family is available in custom and standard configurations.

1.2 Electrum 200 Overview

The Electrum 200 is a single board computer designed for control applications that require high performance, networking and reliable multitasking capabilities. Powered by an Atmel ARM926EJ-S core capable of over 200 MIPS, it can fulfill the most 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 Electrum 200 Features

- 180 MHz 32-bit ARM926EJ-S core
- 1 GB NAND Flash
- MicroSD Socket
- 64 MB SDRAM
- 10/100 Ethernet
- LCD Port, 4x4 Keypad Port
- PC/104 8-bit I/O expansion
- Two RS232 Serial Ports
- I C, SSI (SPI) Ports
- USB Host and device ports
- Optional 8-ch. 12-bit ADC, 4-ch. 12-bit DAC
- Reconfigurable I/O with Xilinx CPLD
- Up to 72 - 5V tolerant GPIOs, 100 without PC/104
- 6 timers, 1 watchdog timer
- Embedded Linux with real-time extensions
- Support for GNU compiler
- Basic, LUA and Python also supported
- +5V power supply required
- Dimensions: 3.8 inches x 4.4 inches

Figure 1.3: Electrum 200 Hardware Block Diagram

1.4 Electrum 200 Development Kit

The Electrum 200-DEV development kit comes with all of the necessary hardware and software to quickly develop applications. The development kit includes the following:

- 1 ? Electrum 200 SBC
- 1 ? +5VDC Positive Center Tap Power Supply

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

1.5 Software and Support

Linux on the Electrum 200 provides a stable and reliable operating system base that can be easily extended using readily available libraries and applications. Linux is loaded from NAND Flash or a microSD card, simplifying software development and distribution. Secure remote access can be implemented via web or command line interfaces, providing off-site monitoring and maintenance capabilities. Many popular Linux applications are available or can be easily ported using the GNU compiler collection as well as popular IDEs including Eclipse and Code::Blocks.

Ports of popular Basic, LUA and Python development tools are available for the Electrum 200 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 http://www.micromint.com. Each product comes with a one year warranty.

NEXT: Getting Started

PREVIOUS: Table of Contents

2 Getting Started

This chapter covers the board and software installation for the Electrum 200.

2.1 Installation

To setup board parameters, a terminal or a PC running a terminal emulator should be connected to the first serial port (COM1) using a null modem cable. Connect a 5 VDC power supply and wait for a login prompt. Upon power up, an Electrum 200 with its default factory configuration will boot from Dataflash and load the kernel and filesystem from NAND. To perform the initial board setup, please login as root and press ENTER when prompted for a password. Change the password using the ?passwd? command.

```
Welcome to Micromint Linux 2.6.28.8 ttyS1
electrum200 login: root
Password:
login[387]: root login on 'ttyS1'

BusyBox v1.13.2 (2009-02-27 17:02:28 EDT) built-in shell (ash)
Enter 'help' for a list of built-in commands.

# passwd root
Changing password for root
New password:
Retype password:
Password for root changed by root
```

The board default Ethernet configuration uses a static address. To change the address or to use DHCP, edit the following lines in /etc/init.d/rcS.

```
# Static IP address
ifconfig eth0 192.168.1.221 netmask 255.255.255.0
route add default gw 192.168.1.253

# Dynamic IP address (DHCP)
#udhcpc -i eth0
```

To enable remote access, generate the SSH keys and remove the comment (#) from the last line in the rcS file.

```
# dropbearkey -t rsa -f /etc/dropbear/dropbear_rsa_host_key
Will output 1024 bit rsa secret key to '/etc/dropbear/dropbear_rsa_host_key'
Generating key, this may take a while...
...
Fingerprint: md5 a1:7f:e9:62:a7:e8:27:fa:ab:db:86:66:b2:93:6f:cb

# dropbearkey -t dss -f /etc/dropbear/dropbear_dss_host_key
Will output 1024 bit dss secret key to '/etc/dropbear/dropbear_dss_host_key'
Generating key, this may take a while...
...
Fingerprint: md5 01:4f:6f:f5:37:41:c5:f4:ea:6b:a6:97:6c:03:10:f7
```

Before powering down, please execute the ?halt? command from the root account to insure a proper shutdown.

```
# halt
The system is going down NOW!
Sending SIGTERM to all processes
Requesting system halt
System halted.
```

Files can be transferred from the PC to the Electrum 200 using an Ethernet network, a USB flash drive or the COM1 serial port. Ethernet uploads can be made via SCP, NFS, FTP (ftpget) or TFTP. Serial uploads can be made via zmodem (rz).

2.2 Toolchain Installation

To develop software for the Electrum 200, the recommended environment is a PC running x86 Linux. The Linux kernel should be 2.6.8 or above and the GCC compiler 4.1.1 or above. Debian or derivatives (including Ubuntu) and RedHat or derivatives (including CentOS and Fedora) are the Linux distributions supported by Micromint. A Tools CD is included with the board that contains the following directories:

Directory	Description
/boot	bootloader, kernel and filesystem files. These can be used to restore the board to its factory installation state.

/docs	Documentation in PDF format.
/drivers	USB drivers.
/licenses	Licenses applicable to software included on the CD.
/src	Board support package (BSP) and kernel source code
/tools	Compiler toolchains and other tools
/utils	General purpose utilities.

Tools CD - Directories
The recommended toolchain is the Sourcery G++ Lite for GNU/Linux from CodeSourcery. A snapshot of this toolchain can be found on the tools directory. The current version of this toolchain can be can be retrieved from the URL below. Select the ?IA32 GNU/Linux TAR? distribution when updating.

http://www.codesourcery.com/downloads/public/gnu_toolchain/arm-none-linux-gnueabi

These are representative commands to mount the CD, install the software on your home directory and add the directory to your PATH. Use the directories, username (myuser) and group (mygroup) applicable to your system.

```
$ mount /dev/cdrom /mnt/cdrom
$ cd ~
$ tar xfj /mnt/cdrom/tools/arm-2008q3-41-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2
$ tar xfj /mnt/cdrom/tools/devkitARM_r25b-i686-linux.tar.bz2
$ tar xfj /mnt/cdrom/tools/electrum200-bsp-2009-03.tar.bz2

$ cd electrum200
$ tar xfj /mnt/cdrom/tools/linux-2.6.28.8-at91.tar.bz2
$ sudo tar xfj /mnt/cdrom/tools/rootfs.arm.tar.bz2
$ sudo chown ?R myuser:mygroup rootfs
$ echo "export PATH=$PATH:~/arm-2008q3/bin;~devkitARM" >> ~/.bashrc
$ unmount /mnt/cdrom
```

To PATH change will be effective on your next login. To confirm that the toolchain is on your PATH and is working properly, try compiling a simple ?Hello? program.

```
$ arm-none-linux-gnueabi-gcc ?version
arm-none-linux-gnueabi-gcc (Sourcery G++ Lite 2008q3-41) 4.3.2
Copyright (C) 2008 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
$ arm-none-linux-gnueabi-gcc -g -march=armv5te -Os -Wall hello.c -o hello

$ file hello
hello: ELF 32-bit LSB executable, ARM, version 1 (SYSV), dynamically, linked (uses shared libs),
for GNU/Linux 2.6.14, not stripped
```

To set the toolchain prefix when compiling user space applications with Sourcery G++ Lite for GNU/Linux use

```
$ export CROSS_COMPILE=arm-none-linux-gnueabi-
```

Some applications may also require changing other Makefile variables to use the toolchain prefix. To test your applications in the Electrum 200, you can export one of your directories via NFS and mount it from the board. Once the application is complete, you can copy the binaries to the flash filesystem. Other alternatives to copy applications to the board include SCP, FTP, TFTP or zmodem.

Other toolchains can be used with the Electrum 200 to fulfill specific application requirements. They are listed on the README file included in the /tools directory of the CD as alternate toolchains. Purposes of alternate toolchains include (but are not limited to) using GPL2 licensing (gcc 4.2.1 or lower) or using uClibc as the system ?C? library. Note that using an alternate toolchain to regenerate a kernel and its operating system utilities requires significant expertise with embedded Linux.

Although x86 Linux is the preferred development environment, it is feasible to develop user space applications on Windows PCs. The recommended toolchain would the ?IA32 Windows Installer? of Sourcery G++ Lite for GNU/Linux. Installation of the Cygwin shell and utilities is also advisable if you intend to use ?make? and other tools normally available under Linux. This will allow you to compile Linux user space executables to use on the Electrum 200. One important limitation when using a Windows PC for development is that many Makefiles for Linux applications assume packages or characteristics of a Linux installation that may not be implemented in Cygwin under Windows. For example, it could be very time consuming to rebuild the Linux kernel or the board bootloader under Windows. If you are restricted to using a Windows PC for development, you may consider VMWare or other virtualization software that allows you to run Linux from Windows.

2.3 Using Remote NFS Filesystems

The kernel in the Electrum 200 can mount its root filesystem via NFS. This allows setup of a directory in a development workstation or server to act as a repository for the target files, saving time during development since files are instantly accessible from the board without a transfer process. Once the development is released, the directory can be converted to a JFFS2 filesystem using ?mkfs.jffs2?.

These are the main steps to use an NFS root filesystem with the Electrum 200:

1. Install the NFS service in your development workstation, if it is not installed already

```
$ apt-get install nfs-kernel-server nfs-common portmap
```

2. Add the ARM root filesystem directory to your /etc/exports file and allow access to your local subnet.

```
/home/electrum200/rootfs 192.168.1.0/24(rw,sync,no_subtree_check,no_root_squash)
```

After changing the exports file, perform an update by executing ?exportfs ?a?.

3. Allow access in /etc/hosts.allow to portmap and NFS daemons to your local subnet.

```
# Allow NFS mounts from local network
portmap: 127.0.0.1 192.168.1.0/255.255.255.0
mountd: 127.0.0.1 192.168.1.0/255.255.255.0
lockd: 127.0.0.1 192.168.1.0/255.255.255.0
rquotad: 127.0.0.1 192.168.1.0/255.255.255.0
statd: 127.0.0.1 192.168.1.0/255.255.255.0
```

4. Change the boot arguments on the board to use NFS. When booting, press any key to go to the U-Boot prompt. These are the arguments to boot the board with 192.168.1.221/24 and mount the root filesystem via NFS from 192.168.1.249.

```
Electrum200> setenv bootargs console=ttyS1,115200 root=/dev/nfs nfsroot=192.168.1.249:/home/electrum200/rootfs
ip=192.168.1.221:192.168.1.249::255.255.255.0::eth0:
Electrum200> saveenv
Electrum200> reset
```

The parameters are as follows:

```
nfsroot=<server_ip>:<mount_directory>
ip = <board_ip>:<server_ip>:<gateway_ip>:<netmask>:<hostname>:<device>:<proto>
```

Note that you can export multiple directories for different purposes, e.g. a development filesystem, a production filesystem, an application-specific filesystem, etc. This can save time during the development process.

For more details on configuring NFS, please visit the following URL or contact Micromint support at support@micromint.com.

http://tldp.org/HOWTO/NFS-HOWTO/server.html

2.4 Updating the Kernel and Root Filesystem

Released versions of the the Electrum 200 Linux kernel and filesystem are included in Released versions of the the Electrum 200 Linux kernel and filesystem are included in the ?boot? directory of the Tools CD. Full sources are included in the board support package (BSP) to allow development of customized kernels and/or filesystems. They can be rebuilt with the Makefile provided.

```
$ cd ~/electrum200
$ make kernel
$ make filesystem
```

The resulting kernel and filesystem will be in the ?boot? subdirectory. These files can be uploaded to the Electrum 200 NAND flash using the U-Boot bootloader. Updating the NAND will require the addresses of the NAND flash partitions shown in Table 2-1.

Start	End	Description	Max Size	Device
0x00000000	0x0003FFFF	Bootstrap	256 KB	/dev/mtdblock0
0x00040000	0x0007FFFF	U-Boot executable	256 KB	/dev/mtdblock1
0x00080000	0x000BFFFF	U-Boot environment	256 KB	/dev/mtdblock2
0x000c0000	0x000FFFFF	U-Boot redundant	256 KB	/dev/mtdblock3
0x00100000	0x001FFFFF	User area	1 MB	/dev/mtdblock4

0x00200000	0x002FFFFF	Linux kernel	2 MB	/dev/mtdblock5
0x00400000	0x3FFFFFFF	Root filesystem	1,020 MB	/dev/mtdblock6

Table 2-1: NAND Flash Memory Map ? 1 GB (Base = 0x40000000)
NOTE: Only the kernel and filesystem partitions are used when booting from Dataflash

To access the bootloader console, connect a terminal or a PC running a terminal emulator should be connected to the first serial port (COM1) using a null modem cable. Upon power up, press any key to prevent the boot process from loading the kernel. Files can be uploaded using a USB flash drive, TFTP, NFS or a serial upload. The following would be representative commands to use a USB flash drive for NAND updates.

```
Electrum200> nand erase

NAND erase: device 0 whole chip
Skipping bad block at  0x02dc0000
Skipping bad block at  0x33b80000
Erasing at 0x3ffc0000 -- 100% complete.
OK

Electrum200> usb start
(Re)start USB...
USB:   scanning bus for devices... 2 USB Device(s) found
       scanning bus for storage devices... 1 Storage Device(s) found

Electrum200> fatload usb 0 0x22000000 uimage-2.6.28.8
 reading uimage-2.6.28.8
...
1400704 bytes read

Electrum200> nand write 0x22000000 0x200000 0x200000

NAND write: device 0 offset 0x200000, size 0x200000
 2097152 bytes written: OK

Electrum200> fatload usb 0 0x22000000 rootfs.arm.jffs2
reading rootfs.arm.jffs2
...
6029312 bytes read

Electrum200> nand write.jffs2 0x22000000 0x400000 0x5c0000

NAND write: device 0 offset 0x400000, size 0x5c0000
 6029312 bytes written: OK

Electrum200> usb stop
stopping USB..

Electrum200> reset
```

The NAND flash erase procedure can report some bad blocks. That is normal. They are mapped during the process and not used for storage. Note that the filesystem size (last parameter in the ?nand write? command) should match the size of the file you generated to insure the filesystem is created properly.

To copy the files from an FTP server, check that the following parameters are setup in the bootloader environment.

```
Electrum200> printenv
bootcmd= console=ttyS1,115200 root=/dev/mtdblock5 mtdparts=at91_nand:128k(bootstrap)ro,256k(uboot)ro,128k(env)ro,
1536k(user),2M(linux),-(root) rw rootfstype=jffs2
bootdelay=1
baudrate=115200
ethact=macb0
ethaddr=00:21:a3:00:00:00
ipaddr=192.168.1.221
netmask=255.255.255.0
serverip=
bootargs=console=ttyS1,115200
stdin=serial
stdout=serial
stderr=serial

Electrum200> setenv ethaddr 00:21:a3:00:00:00
Electrum200> setenv ipaddr 192.168.1.221
Electrum200> setenv netmask 255.255.255.0
Electrum200> setenv serverip 192.168.1.249
Electrum200> saveenv
```

The ethaddr should match the board serial number of the board and the serverip should be the TFTP server address. With this setup, the NAND flash can be updated via TFTP using the following commands.

```
Electrum200> nand erase

NAND erase: device 0 whole chip
Skipping bad block at  0x02dc0000
Skipping bad block at  0x33b80000
Erasing at 0x3ffc0000 -- 100% complete.
OK

Electrum200> usb start
(Re)start USB...
USB:   scanning bus for devices... 2 USB Device(s) found
       scanning bus for storage devices... 1 Storage Device(s) found

Electrum200> tftp 0x22000000 uImage-2.6.28.8
...

Electrum200> nand write 0x22000000 0x200000 0x200000

NAND write: device 0 offset 0x200000, size 0x200000
 2097152 bytes written: OK

Electrum200> tftp 0x22000000 rootfs.arm.jffs2
...

Electrum200> nand write.jffs2 0x22000000 0x400000 0x5c0000

NAND write: device 0 offset 0x400000, size 0x5c0000
 6029312 bytes written: OK

Electrum200> usb stop
stopping USB..

Electrum200> reset
```

To use NFS or serial upload (loady) please consult the U-Boot documentation at the URL below or contact Micromint support (support@micromint.com).

http://www.denx.de/wiki/U-Boot

2.5 Updating the Bootstrap and Bootloader

Released versions of the the Electrum 200 botstrap and bootloader are included in the ?boot? directory of the Tools CD. Full sources are included in the board support package (BSP) to allow development of customized boot features. They can be rebuilt with the Makefile provided.

```
$ cd ~/electrum200
$ make bootstrap
$ make bootloader
```

The resulting kernel and filesystem will be in the ?boot? subdirectory. These files can be uploaded to the Electrum 200 Dataflash using the Atmel AT91 In-System Programmer (ISP) utility included in the ?tools? directory on the CD. The AT91 ISP is currently only released for Windows PCs. A test version of the AT91 ISP for Linux is available on the Atmel web site. Please check the README file in the ?tools? directory of the CD. Updating the Dataflash will require the addresses of the Dataflash partitions shown in Table 2-2.

Start	End	Description	Max Size
0x00000000	0x0000107F	Bootstrap	4,224 >4K
0x00001080	0x000041FFF	U-Boot environment	12,672 >12 KB
0x00004200	0x0003DDFF	U-Boot executable	236,544 >231 KB
0x0003DE00	0x00041FFF	User area	16,896 >16 KB

Table 2-2: DataFlash Memory Map ? 256 KB (CS1 Base = 0xD0000000)
The SAM-BA utility of the AT91 ISP can upload the boot files using a Segger J-Link (JTAG) or a serial port connected to COM2 (DBGU). When updating the Dataflash via the serial port, jumper JP4 must be removed prior to powering up the board. When the AT91 ISP is loaded, a window similar to that of Figure 2-1 will be displayed. Select the desired interface and click the ?Connect? button.

Figure 2-1: Atmel AT91 ISP connection window
For serial port uploads, after the connect process please replace jumper JP4 to allow access to the Dataflash. Once the connection to the board is established, a window similar to that of Figure 2-2 will be displayed. Select the ?Dataflash AT45 DB? tab.

Figure 2-2: Atmel AT91 ISP Dataflash utilities
To perform the upload follow these steps:

1. Select ?Enable Dataflash (SPIO CS1)? from the scripts listbox and click on ?Execute?.

2. Select ?Send boot file? from the scripts listbox and click on ?Execute?. Select the location and filename of the bootstrap file bootstrap-df-1.11.bin.

3. Select the location and filename of the bootloader file u-boot-df-2009.01.bin . Enter 0x4200 on the Address field and click on ?Send File?.

4. Exit SAM-BA and reset the board.

This procedure does not erase the bootloader data saved on the environment partition. That is erased only if you select to completely erase the Dataflash with the SAM-BA ?Erase Dataflash? script. If the Dataflash is erased, after resetting the board, press a key when booting to enter a minimal bootloader environment as indicated in section 2.3 of this manual.

2.6 Memory Maps

The processor memory map and the Linux device names of internal peripherals are shown in Tables 2-3 and 2-4 and Figure 2-3. Majors and minors follow the Linux device name conventions listed on the devices-2.6+.txt file included in the ?docs? directory of the Tools CD. IOCTLs available to user space applications vary according to the driver. Consult Linux device driver references for more details.

Start	End	Description
0x00000000	0x0FFFFFFF	Internal CPU memories
0x20000000	0x2FFFFFFF	SDRAM (32 ? 256 MB)
0x40000000	0x4FFFFFFF	NAND (128 MB ? 2 GB)
0xD0000000	0xDFFFFFFF	DataFlash at CS1
0xF0000000	0xFFFFFFFF	Internal CPU peripherals

Table 2-3: Main Memory Map

Start	Description	Device
0xFFFB0000	COM1 (UART0)	/dev/ttyS0
0xFFFB4000	UART1	
0xFFFB8000	UART2	
0xFFFC4000	Ethernet MAC (EMAC)	eth0
0xFFFC8000	SPI0	
0xFFFCC000	SPI1	
0xFFFFF200	COM2 (DBGU)	/dev/ttyS1
0xFFFFF400	PIOA	
0xFFFFF600	PIOB	
0xFFFFF800	PIOC	

Table 2-4: Internal Peripheral Memory Map

Figure 2-3: AT91SAM9260 Memory Map

NEXT: Hardware

PREVIOUS: Introduction

3 Hardware

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

3.1 Microprocessor

The Electrum 200 includes an AT91SAM9260 microprocessor which is based on the integration of an ARM926EJ-S processor with fast ROM and RAM memories. This 32-bit ARM9 microprocessor is capable of 180-MHz operation. It has a wide range of peripherals including an Ethernet MAC, a USB Device Port, and a USB Host controller. Several standard peripherals are also included, such as USARTs, SPI bus, TWI bus (I2C), timers, and counters. Please see the Atmel Semiconductors ? AT91SAM9260 Microprocessor Data Sheet for more information and register definitions.

AT91SAM9260 key features:

- Internal Memory
  - 32 kilo-bytes internal ROM
  - Two 4-kB internal SRAM
- Timers
  - Two Three-channel 16-bit Timer/Counters
    - Three External Clock Inputs, Two Multi-purpose I/O pins per channel
    - Double PWM Generation, Capture/Waveform Mode, Up/Down Capability
    - High-Drive Capability on Outputs TIOA0, TIOA1, TIOA2
  - Periodic Interval Timer
    - 20-bit interval Timer plus 12-bit Counter
  - Watchdog Timer
    - Key protected, Programmable Only Once
    - Windowed 16-bit counter running on slow clock
  - Real-time Timer
    - 32-bit Free-running Backup Counter with 16-bit Prescaler
- 10/100 Ethernet MAC
  - 28-byte FIFOs
  - Dedicated DMA Channels for Receive and Transmit
- Four Universal Synchronous/Asynchronous Receiver Transmitters (USART)

- ♦ Individual baud rate generator
- ♦ IrDA Infrared Modulation/Demodulation, Manchester Encoding

• Two Master/Slave Serial Peripheral Interface (SPI)
- ♦ 8-to 16-bit programmable data lengths
- ♦ Synchronous Communications

• One Two-wire Interface (TWI) (I2C)
- ♦ Master, multi-master or slave operation

• USB2.0 Full Speed (12 Mbits per second) Device Port
- ♦ On-chip transceiver
- ♦ 2,432-byte configurable DPRAM

• USB2.0 Full Speed (12 Mbits per second) Host Port
- ♦ Integrated FIFOs
- ♦ Dedicated DMA channels

• Three 32-bit Parallel Input/Output Controllers (PIOA, PIOB, PIOC)
- ♦ Input change interrupt on each I/O
- ♦ Individually programmable open-drain and pull-up resistor
- ♦ High current drive I/O lines, up to 16mA each

• Reset Controller
- ♦ Based on power-on reset cell
- ♦ Reset source identification
- ♦ Reset output control

• Additional Features
- ♦ IEEE 1149.1 JTAG Boundry Scan on all digital pins
- ♦ Programmable PLL for system clock

3.2 Data Flash

The Electrum 200 includes a 32k Byte Data Flash. The communication to the Data Flash is through a Serial Peripheral Interface bus (SPI). It has a maximum clock frequency of 66MHz. The Data Flash is used to store the boot-loader for the AT91SAM9260 microprocessor. For further information please see Atmel Semiconductors AT45DB021D Data Sheet.

3.3 SD RAM

The Electrum 200 includes positions for two 32MB SD RAMs for a total of 64MB. Read and write accesses to the SDRAM are burst oriented. Each access starts at a selected location and continue for a programmed number of locations. They have a self refresh mode and a 64mS, 8,192-cycle refresh. For further information please see Micron?s MT48LC16M16A2 Data Sheet.

3.4 NAND Flash

The Electrum 200 comes standard with a 1 GB NAND flash. The NAND Flash is capable of sequential reads in 25nS and can program a page in 220µS. An on-chip control logic automates program and erase operations to maximize cycle endurance. The program/erase endurance is specified at 100,000 cycles. For further information please see Micron Semiconductor?s MT29F4G08AAA Data Sheet.

3.5 PHY

The Electrum 200 includes 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.

3.6 CPLD

A Xilinx?s XC9572XL Complex Programmable Logic Device(CPLD) comes standard with the Electrum 200. 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.

3.7 DAC

A National Semiconductor?s DAC124S085 general purpose digital-to-analog converter (DAC) is an optional feature for the Electrum 200. 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 8.5µs. It allows for simultaneous output updating. For further information please see National Semiconductor?s DAC124S085 Data Sheet.

3.8 ADC

The Electrum 200 has an option to include a National Semiconductor?s ADC128S052 general purpose analog-to-digital converter (ADC). The ADC has eight channels with a resolution of 12-bit. The inputs can range from 0 to 3.3V. Communication to the ADC is done through a three wire synchronous serial interface that operates up to 8 MHz. The ADC?s inputs have a conversion rate up to 500 kSPS. For further information please see National Semiconductor?s ADC128S052 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 Electrum 200.

4.1 Power Supply

The Electrum 200 requires a regulated +5 VDC power supply on connector J1. Typical current requirements are 300 mA with all common peripherals enabled. 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 Electrum 200 should polarity of the power supply be reversed. When power is applied LED1 will illuminate.

4.2 10/100 Ethernet

The Electrum 200 is equipped with a fully-integrated 10/100 Mbps Ethernet port. The Media Access Control (MAC) is implemented in the AT91SAM9260 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 AT91SAM9260 data sheet for the MAC.

4.3 USB Host

The AT91SAM9260 has a fully-integrated USB v2.0 Host port. The host port handles full-speed and low-speed protocols. The port reaches the outside world through J14. J14 is a USB Type A connector. The USB host port controller is fully compliant with the OpenHCI specification. Please see the AT91SAM9260 data sheet for further details on the USB Host port.

4.4 Serial (COM) Ports

A Universal Synchronous Asynchronous Receivers/Transmitters (USART) is level shifted to RS-232 levels. USART0 (COM1) reaches the external world through a male DB9 connector. The Univesal Asynchronous Receiver/Transmitter (UART) Debug Unit (DBGU) is also level shifted to RS-232 levels. The DBGU (COM2/DBGU) reaches the external world through a 2x5 pin berg header. Please see figure 4.4 for the pin outs of COM1 (J3) and COM2/DBGU (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.

4.5 Micro-SD

The microSD socket (J9) enables micro-secure-digital memory cards to be plugged into the Electrum 200 microcontroller board. The microSD card allows the user the ability of a standard removable media for transferring data to and from the Electrum 200.

4.6 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 AT91SAM9260 microprocessor and the Extended I/O is accessed through the CPLD (Complex Programmable Logic Device) via the microcontrollers SPI port one.

There are thirty-three bits of GPIO available on the J2 connector. Please see the pin out for J2 in Figure 4.6. Seven bits are from port A, seventeen are from port B and nine are from port C. Ports A, B, and C have alternate functions other than digital inputs and outputs. Connecting anything to PA0, PA1, and PA2 may cause problems with communications to the Data flash, Micro SD card, 12-bit DAC, and 12-bit ADC because it is a shared SPI bus. Connecting anything to PB0, PB1, and PB2?s may cause problems with communications to the CPLD because it is a shared SPI bus. Table 4.6 lists the alternate functions and a brief description of the function. For further information on the alternate functions please refer to the AT91SAM9260 data sheet.

The J2 connector also has a variety of voltages and a couple of ground pins. The VBAT input is to battery back-up real-time timer. Pin 40 of J2 is where the VBAT input can be accessed and 1.65V to 1.95V may be applied to the pin. Before connecting a voltage to VBAT the 0-ohm resistor R30 will have to be removed and a jumper placed on pins 2 and 3 of JP4.

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 4.6. 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.

Figure 4.6: Extended I/O and GPIO connector pin out (CPLD pin#)

Table 4.6: GPIO Alternate Functions

4.7 Keypad

A 4x4 matrix keypad using a 16-pin (2x8) ribbon cable can be connected to a portion of port B of the microprocessor through J11. Please see Figure 4.7 for the pin out of the keypad connector. If the ports are not used for a keypad they may be used for their alternate functions. Table 4.7 lists the alternate functions for the 8-bits of I/O connected to J11.

Figure 4.7: Keypad connector pin out

J11 Pin#	GPIO	Alternate Function	Bried Description
1	PB24	DTR0	USART0 Data Transmit Ready
2	PB25	R10	USART0 Ring Indicator
3	PB26	RTS0	USART0 Ready to Send
4	PB27	CTS0	USART0 Clear to Send
5	PB28	RTS1	USART1 Ready to Send
6	PB29	CTS1	USART1 Clear to Send
7	PB30	PCK0	Programmable Clock Output 0
8	PB31	PCK1	Programmable Clock Output 1

Table 4.7: Keypad GPIO Alternate Functions

4.8 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 4.8 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 4.8: LCD connector pin out (CPLD pin#)

4.9 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 with boundary scan, please refer to IEEE Standard 1149.1-Test Access Port and Boundary-Scan Architecture

4.10 Pushbuttons and LED

The Electrum 200 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 0 with a 10k? pull-up resistor connected to it. The user LED is connected to port A bit 6 and can be illuminated by CLEARING port A bit 6 of the AT91SAM9260.

4.11 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 CPLD through the microprocessor. A utility provided by Micromint USA will program the CPLD through the AT91SAM9260s SPI1 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 4.11.

Figure 4.11: CPLD programming jumper pin out (CPLD pin#)

4.12 PC/104 Expansion

The available signals on the PC/104 expansion connector are shown in Figure 4.12. 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 Electrum 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.

Figure 4.12: CPLD programming jumper pin out (CPLD pin#Figure 4.12: PC/104 connector pin out (CPLD pin#)

4.13 Analog to Digital Converter (ADC)

The Electrum 200?s optional eight channels of 12-bit ADC can be connected to through J7. Please see figure 4.13 for the pin out of the ADC connector. The ADC is accessed through the AT91SAM9260 SPI0 port. Port C bit 16 is the ADC?s chip select input for reading the conversion counts of the ADC.

Figure 4.13: Analog to Digital connector pin out

4.14 Digtial to Analog Converter(DAC)

The optional four channels of 12-bit DAC can be connected to through J8. Please refer to figure 4.14 for the pin out of the DAC connector. The DAC is accessed through the AT91SAM9260 SPI0 port. Port C bit 17 is the DAC?s sync input for loading the conversion count into the DAC.

Figure 4.14: Digital to Analog connector pin out

4.15 USB Device

The Electrum 200 is optionally equipped with a USB Device Port. The Device port is compliant with the USB V2.0 full-speed device specification. It reaches the outside world through J16. J16 is a micro USB Type AB connector. The USB device port has six endpoints that can be configured in one of several USB transfer types. Please see the AT91SAM9260 data sheet for further details on the USB Host port.

4.16 Option Jumpers

The Electrum 200?s option jumpers are used to set the microprocessor into different modes and have the extra pins that are available on the CPLD. JP3 is used to route power to VDDBU. If R30 is installed on the board then there should not be a jumper on JP3. If R30 is removed and an external power supply is going to be used to power VDDBU then a jumper should be placed on pins 2&3 and the user can apply power for VDDBU on pin 40 of J2. Figure 4.16 contains the pin out for JP3.

Figure 4.16: VDDBU Option Jumper

The chip select signal to the serial dataflash can be disconnected from the microprocessor by removing R32. If the signal needs to be reconnected then a jumper must be placed on JP4. Figure 4.17 contains the pin out of JP4.

Figure 4.17: VDDBU Option Jumper

JP1 is used to be able to put the microprocessor into different modes, to write protect the first 256 pages of the dataflash and to write protect the NAND flash. In order to enable the JTAG boundry scan a jumper must be placed on pins 1&2 before power is applied to the board. If a jumper is placed on pins 3&4 of JP1 before power is applied to the board then the microprocessor will boot from the serial dataflash. If the jumper is left off then the microprocessor will boot from the embedded ROM. If a jumper is placed on pins 5&6 the serial dataflash?s first 256 pages will be write protected. If a jumper is placed on pins 7&8 the NAND flash will be write protected.

Figure 4.18: VDDBU Option Jumper

On pins three through six of J17 are the left over pins from the CPLD. The GCK1, GCK3, GTS2, and GSR signals are not connected to any of the other hardware on the Electrum 200. Pins one and two of J17 are the input and output for the microprocessors shutdown controller. Figure 4.19 contains the pin out of J17.

Figure 4.19: VDDBU Option Jumper

NEXT: Mechanical_and_Electrical_Characteristics

PREVIOUS: Hardware

5 Mechanical and Electrical Characteristics

5.1 Absolute Maximum Ratings

Operating Temperature:

Commercial	0 C to +70 C
Industrial	40 C to +85 C
Storage Temperature	−50 C to +125 C

Maximum Voltage:

Voltage on J1 with LCD	+5.5 VDC Regulated
Voltage on J1 without LCD	+15 VDC Regulated
Voltage on VBAT (J2)	+1.95 VDC Regulated

The Electrum 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

NEXT: References

PREVIOUS: User Interfaces, Connectors, and Jumpers

6 References

6.1 Documents

AT91SAM9260 Microcontroller Data Sheet
http://www.atmel.com/dyn/products/product_card.asp?part_id=3870
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.

XC9500XL High-Performance CPLD Family Data Sheet
http://www.xilinx.com/support/documentation/xc9500xl_data_sheets.htm
This data sheet provides the specifications and general features of the XC9572XL High Performance CPLD.

6.2 Books

Embedded Linux Primer: A Practical Real-World Approach (2nd Edition)

by Christopher Hallinan

ISBN: 0137017839 Publisher: Prentice Hall (November, 2010)

Excellent reference on embedded Linux including processor selection, bootloaders, kernel building, device drivers, file systems and remote debugging.

Linux Device Drivers, 3rd Edition

by Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman

ISBN: 0131679848 Publisher: O'Reilly Media (February, 2005)

Covers the Linux interfaces to hardware and techniques for writing device drivers, including ports, memory management, timers and interrupts. Free online versions are available in HTML and PDF formats.

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

ISBN: 0596005903 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 Electrum SBCs.

Atmel ARM Solutions
http://www.atmel.com/products/AT91/default.asp?family_id=605
Manuals and application notes for the Atmel ARM microcontrollers.

AT91 Community
http://www.at91.com/
Community site for users and developers of AT91 applications.

Meld Embedded Linux Community
http://meld.org/
Meld is a community for embedded Linux developers sponsored by MontaVista Software.

PREVIOUS: Mechanical and Electrical Characteristics