

Getting Started with EMAC OpenEmbedded Linux

This guide provides information on getting started using and configuring an EMAC OpenEmbedded Linux system running EMAC OE version 4.0.

Introduction

Operating systems based on the Linux kernel and Open Source user interfaces provide an extremely flexible and reliable operating system environment for embedded applications. The kernel is entirely open source under the GNU Public License and supports a wide range of hardware and peripherals. The user interface typically consists of the GNU core libraries (glibc) and other open source projects which connect the user to the kernel. This makes the task of developing custom drivers and applications very efficient as the source is freely available and a community of developers can be utilized. EMAC can provide GNU/Linux for nearly all of our single board computers.

EMAC OpenEmbedded Linux (also known as EMAC OE Linux) is based on the OpenEmbedded build system and is currently utilized on the majority of EMAC's products. OpenEmbedded allows EMAC to develop a specialized, targeted build entirely from source. This reduces memory footprint, improves performance, and prevents the inclusion of unnecessary libraries and applications. The kernel for EMAC OpenEmbedded Linux is based on vanilla Linux kernel source and patched to add support for the target hardware platform. This includes drivers for peripherals and on-chip features such as GPIO (general purpose input/output), memory devices, audio codecs, and analog-to-digital converters.

Initial Connections

There are several ways to connect to a system running EMAC OE Linux. The preferred connection method depends on the capabilities of the hardware. For example, headless systems (those without a graphical interface) will most likely be connected to by serial port, while systems with a video connection may be accessed using a keyboard and monitor.

Serial Terminal

The most common initial connection to the system will be made using a serial terminal. This requires access to a terminal application such as Minicom under Linux or PuTTY under Microsoft Windows. The serial cable type, console connection, and default communication parameters are listed in Table 1 below.

Table 1: Serial Communication Parameters

Board	Port	Cable Type	Baud	Data Bits	Parity	Stop Bits	Flow Control
iPac-9302	HDR6	NULL Modem	57600	8	None	1	None
PPC-E7	CN1	NULL Modem	57600	8	None	1	None
SoM-200ES with SoM-9307M	CN1	NULL Modem	57600	8	None	1	None
SoM-200ES with SoM-9G45M	COMB	NULL Modem	115200	8	None	1	None
SoM-200ES with SoM-9M10M	COMB	NULL Modem	115200	8	None	1	None
SoM-210ES with SoM-9307M	COMA	NULL Modem	57600	8	None	1	None
SoM-210ES with SoM-9G45M	COMB	NULL Modem	115200	8	None	1	None
SoM-210ES with SoM-9M10M	COMB	NULL Modem	115200	8	None	1	None
SoM-150ES	HDR11	NULL Modem	115200	8	None	1	None
x86 Boards	*	*	*	8	None	1	None

*: These settings vary among x86 systems. Refer to the system manual for the target board for more information. Not all x86 systems have a serial terminal connection enabled by default.

Follow the steps below to establish a connection.

1. Connect a serial cable between the serial port on the workstation and target board and start a terminal application.
2. Setup the communication parameters for your board according to Table 1.
3. Connect power to the target board; boot messages will begin to print within a few seconds.
4. When the board has finished booting a login prompt will appear.

Note that boards that do not have the serial device set as the default kernel console (such as the majority of headed x86 systems) will not display messages over the serial port during boot without further configuration. In these cases the login prompt will appear when the board has finished booting. If a login prompt does not appear, press enter in the serial console application several times.

For login credentials, see System Login below.

Network Connections

This section describes the process of establishing a basic network connection with the board. Once a network connection is established, the system may be accessed through the servers on the board; these include SSH, HTTP, FTP, and TELNET servers.

Local Area Network

To make a connection to a local area network (LAN), simply connect a standard Ethernet patch cable to the board. This may be done before or after the board is powered, as the system is set up to allow hot-plugging of the Ethernet cable.

EMAC OE Linux devices are set to attempt a DHCP network configuration by default. If a DHCP server is available on the network that the system is connected to, it will contact the server to secure an IP address lease.

If an address has been obtained by DHCP, there are several ways of determining what address has been leased. The easiest way to find this information is to connect to the board through another method and use the ifconfig command to determine the address that has been obtained on the eth0 interface. Alternatively, network tools such as packet sniffers can be used to determine what IP address the board has obtained. The local network administration will be able to provide this information as well.

If DHCP fails to obtain an IP address, the system will fall back to the static IP settings in Table 2 below.

Table 2: Default Static IP Settings

IP Address	10.0.2.41
Subnet Mask	255.255.255.0
Default Gateway	10.0.2.1
Broadcast Address	10.0.2.255

Direct Connection

If a LAN connection is not possible or desirable, a direct network connection to the board may be made. To establish a direct connection with the system, use an Ethernet cross-over cable between the workstation and target board. Alternatively, if a cross-over cable is not available, workstation and target board can each be plugged into an Ethernet switch with standard Ethernet Cat-5/6 patch cables without any other devices connected to the switch. In this case, the switch will make the cross-over connections automatically. Once the physical connections have been made, apply power to the board.

If a direct connection has been made to the board and a DHCP server is not running on the development PC, DHCP will fail and the system will fall back to the default IP settings in Table 2 above. The network interface on the workstation will then need to be configured to use the same subnet mask and broadcast address of the board with a different IP address that falls in the same subnetwork (10.0.2.1 for example). See Table 3 below for suggested static network settings.

Table 3: Static Network Settings for Direct Connection

IP Address	10.0.2.1
Subnet Mask	255.255.255.0
Default Gateway	Blank or 10.0.2.1
Broadcast Address	10.0.2.255

To verify that the connection is working correctly, use the ping utility to test that the board responds at address 10.0.2.41.

Once connected, see System Login below.

System Login

EMAC OE Linux allows login from getty, SSH (Secure Shell) and TELNET. Getty uses the serial connection or console while SSH and TELNET utilize a network connection. Warning: using TELNET will expose all data “in the clear” to everyone in the network you are using. This includes your root password and any data you transmit between the system and your computer. EMAC recommends using SSH for all network connections. TELNET should never be used if the system is connected directly to the Internet outside of any firewall or NAT router.

The default root login and password are given in Table 4 below. There is also a normal user on the system with limited system permissions (see Table 4).

Table 4: Default Login Credentials

Username	Password
root	emacs_inc
user	emacs_inc

Note that all EMAC systems ship with the same default passwords. For this reason it is very important that the passwords are changed as soon as possible.

System Configuration

After logging in, some initial configuration of the system may be necessary. Below is a list of recommended common configuration steps.

1. Change the root password and possibly create a new user.
2. Configure the static network settings if DHCP will not be used.
3. Set the date, time, and timezone.
4. Disable the TELNET server if it is not necessary.
5. Configure the SSH server for secure remote login.

Remounting the Root Filesystem

By design, EMAC OE Linux uses a read-only root file system to preserve the life and reduce corruption to the flash either due to too many writes or bad writes during a power failure. Before continuing with any of the listed steps, the root file system will need to be remounted read/write before configuring the system.

At the shell prompt, enter the following command:

```
mount -o remount,rw /
```

Upon completion of the system configuration, remount the root file system read-only. At the shell prompt, enter the following command:

```
mount -o remount,ro /
```

It is possible to change the default behavior so that the root filesystem is mounted read/write by default. While this may be convenient for development, be careful that the system is only shut down using the shutdown, reboot, or halt commands when the root filesystem is mounted read/write. This is especially important with a system using an on-board NAND flash with a JFFS2 filesystem, as write-buffering is required for NAND flash.

To change the default behavior, edit the /etc/fstab file and change the mount options for the root filesystem from ro to rw. The exact entry in the fstab will differ depending on the device being used.

User Account Configuration

Nearly every important file on a Linux system is owned by a single user and group: root. The root account is essential to the security of any Linux system. Having all of the important executable and configurations files on the system owned by root means that only the root user may alter the way the system works. Other users have no permission to execute or change system files, and therefore, they cannot modify or damage the system.

As such, it is critical that the root password be kept secret, and changed as often as is practical. A clumsy or malicious individual with the system's root password could render the entire system inoperative in a matter of seconds. Every EMAC Linux system ships from the factory with the same root password. EMAC recommends you change the root password during your first initial login.

Use the passwd utility to change the root password. See below for an example:

```
root@emac-oe:~# passwd
Changing password for root
New password:
Retype password:
Password for root changed by root
```

To change the password for the user, either login as user and use the passwd utility as shown above, or use the command passwd user when logged in as root.

If desired, additional users may be added to the system. The following example demonstrates the process for creating a user with the name 'test'.

```
root@S0M9G20M:~# adduser test
Changing password for test
New password:
Retype password:
Password for test changed by root
```

Network Configuration

Linux is built with extensive networking capability, making it an ideal choice for a server. EMAC OE Linux includes a large range of networking options. Reliable network connections can be made easily and offer many configuration options to suit the user's needs.

Network configuration is handled using a utility called ifplugd, which manages automatic configuration of network connections. Network configuration options are read from the file /etc/network/interfaces. The default network interface is named eth0.

Default Fallback Settings

By default, the system will attempt to use DHCP for network configuration. If DHCP configuration fails, default static IP settings will be used (see Network ^{*****}Connections). These default settings ^{*****}can be changed by editing the scripts in /etc/udhcpd.d. The default.leasefail file sets defaults in the case of a failed DHCP request. The section ^{*****}

of code below is an excerpt from this file demonstrating the portion of code that can be modified to change the default values.

```

-----
# System defaults for a failed DHCP request.
# IP address
ip="10.0.2.41"
# Network broadcast
broadcast="10.0.2.255"
# Local default gateway (router)
router="10.0.2.1"
# domain contains the domain name
domain="emacinc.com"
# domain IP address
dns="10.0.2.1"
# Network subnet
subnet="255.255.255.0"
-----

```

Static Network Configuration

Depending on the network that the board will be connected to, it is often necessary to configure the board for a static IP address and settings rather than using DHCP. This can be done by editing the file `/etc/network/interfaces` directly. The example below illustrates how to set a static network configuration. See the `resolv.conf` man page (<http://linux.die.net/man/5/resolv.conf>) for more configuration options.

Listing 1: /etc/network/interfaces Static IP Configuration

```

-----
# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)

# The loopback interface
auto lo
iface lo inet loopback

# DHCP automatic IP address disabled
#allow-hotplug eth0
#iface eth0 inet dhcp

# Static IP address
allow-hotplug eth0
iface eth0 inet static
    address 192.168.0.50
    gateway 192.168.0.1
    netmask 255.255.255.0
-----

```

Note that the `allow-hotplug` directive in the example above causes the interface to be configured dynamically by the `ifplugd` daemon.

Once the static IP configuration has been set, the DNS (Domain Name System) resolver configuration should also be changed to match the network that the system will be running on. The resolver configuration is stored in the file `/etc/resolv.conf`. Because `/etc/resolv.conf` must be volatile, it is a link to `/var/run/resolv.conf`, which is stored in a ramdisk. The default configuration in `/etc/default/resolv.conf` is copied into this file on system boot. Edit the file `/etc/default/resolv.conf` for non-volatile configuration changes. The following demonstrates an example resolver configuration. See the `resolv.conf` man page (<http://linux.die.net/man/5/resolv.conf>) for more configuration options.

Listing 2: Example Resolver Configuration

```

-----
nameserver 10.0.2.1
search emacinc.com.
-----

```

Disabling TELNET

Using TELNET will expose all data “in the clear” to everyone on the network. This includes the root password and any data transmitted between the system and other devices. EMAC recommends using the [SSH](#) server for all network sessions, especially if the system is open to the Internet.

To disable the TELNET server, comment out the directive in the inetd configuration file `/etc/inetd.conf`. An example is shown in Listing 3 below. Note that this will not take effect until the system or the inetd server is restarted.

Listing 3: Disabling the Telnet Server in `/etc/inetd.conf`

```
-----
# :STANDARD: These are standard services.
ftp          stream  tcp      nowait  root    /usr/sbin/tcpd  /usr/sbin/ftpd
#telnet      stream  tcp      nowait  root    /usr/sbin/tcpd  /usr/sbin/telnetd
-----
```

Date and Time Configuration

There are two different clocks on a Linux system. The RTC, or hardware clock, is present on most systems and is used to provide a persistent clock using a battery backup and for other purposes like waking the processor up from suspend mode through an alarm interrupt. The hardware clock is generally only accessed to retrieve an initial time when Linux boots and is rarely accessed from a running system. The system clock is implemented in the kernel and is used for timekeeping purposes for the running system. On Linux, time is represented by the number of seconds from the Unix epoch: 00:00:00 UTC on January 1, 1970.

Hardware Clock

The hardware clock can be accessed through the `hwclock` command. Running `hwclock` with no arguments prints the current value stored in the hardware clock. The `–systohc` option sets the value of the hardware clock using the system clock. This command is called by the system shutdown scripts using `/etc/init.d/hwclock.sh`. The `–hctosys` option can be used to set the system clock using the value stored in the hardware clock. This is used by `/etc/init.d/hwclock.sh` to initialize the system clock during the boot process. Note that the hardware clock is generally set to UTC regardless of the local time zone.

On some boards you may receive a message such as `hwclock: RTC_RD_TIME: Invalid or incomplete multibyte or wide character` or a similar message when attempting to read the hardware clock. This generally means that the hardware clock has not yet been set, in which case running the `hwclock –systohc` command will correct the issue.

System Clock

The system clock can be read and set using the `date` command. See the usage information for the `date` command by running `date –help` for information about how to use this command.

During every boot, the system clock is initialized using the hardware clock as described above. If the EMAC OE system has access to the Internet on system boot, it will attempt to synchronize the system clock using the Network Time Protocol (NTP). This is done using the `ntpd` command in the script `/etc/init.d/ntpd`.

Local Timezone

By default, EMAC OE systems are set to UTC time zone. Setting the time zone is done by creating a link from `/etc/localtime` to the desired time zone information file. The most common time zones are installed in the base image; contact EMAC for additional time zone definitions if required. Time zone information files are stored in the `/usr/share/zoneinfo` directory. For example, for Central Standard Time (CST), the file `/usr/share/zoneinfo/America/Chicago` or `/usr/share/zoneinfo/CST6CDT` would be used. This will also allow the system to automatically adjust for Daylight Savings Time.

The example below illustrates the process of setting the time zone. Notice that it is necessary to log out and log

back in for the new timezone setting to take effect. The command below that actually sets the timezone is `ln -s /usr/share/zoneinfo/America/Chicago /etc/localtime`:

```
root@SOM9G20M:~# hwclock
Thu Feb  4 22:54:48 2010  0.000000 seconds
root@SOM9G20M:~# date
Thu Feb  4 22:54:49 UTC 2010
root@SOM9G20M:~# rm -rf /etc/localtime #only necessary if this file exists
root@SOM9G20M:~# ln -s /usr/share/zoneinfo/America/Chicago /etc/localtime
root@SOM9G20M:~# ls -l /etc/localtime
lrwxrwxrwx  1 root  root          35 Feb  4 22:55 /etc/localtime -> /usr/share/zoneinfo/America/Chicago
root@SOM9G20M:~# date
Thu Feb  4 22:55:46 UTC 2010
root@SOM9G20M:~# logout
....
SOM9G20M login: root
Password:
root@SOM9G20M:~# date
Thu Feb  4 16:56:05 CST 2010
root@SOM9G20M:~# hwclock
Thu Feb  4 22:56:08 2010  0.000000 seconds
root@SOM9G20M:~#
```

Graphics

EMAC OE builds are available with graphical applications and libraries on boards that have video capabilities. Systems that come with an LCD, such as panel PCs, are provided with graphical builds by default. Other systems with graphical capabilities, such as headed x86 systems, are configured with a video console by default. Graphical packages for these systems are available on request for a small build and support fee.

The two main Graphical User Interfaces (GUIs) provided by EMAC are X11 with Matchbox Window Manager and Qt4 embedded. X11 is the standard option for most systems and includes the GTK+ libraries GUI application development. Other libraries can be provided upon request as an addon package.

Software Development and Custom Builds

If additional software packages are required, EMAC may be able to provide these for an additional support and installation fee. EMAC has access to thousands of available software projects through the OpenEmbedded build system, most of which can be compiled and installed on any EMAC OE system. Please contact EMAC for information about package availability.

EMAC provides an SDK for EMAC OE systems allowing custom software development. Refer to the EMAC OE software development documentation for more information. EMAC also provides application and custom kernel driver engineering services. Please contact EMAC for more information regarding software engineering services.

Once a custom build has been developed, EMAC can archive this build under a custom part number for future orders. This build will then be pre-programmed prior to shipping. Contact EMAC for more information regarding our archive service.

» import » qt » install » getting_started » eclipse » uboot_image_loading » boot_process » emacs_oe_fact »
linux_start » emacs_oe_getting_started

-
- linux/emacs_oe_getting_started.txt · Last modified: 2010/09/04 11:46 by tstratman
 - Except where otherwise noted, content on this wiki is licensed under the following license: CC Attribution-No Derivative Works 3.0 Unported (cc-by-nd)