Quick Start Guide

Linux S32V_BSP23.1

11-Nov-2020

Contents

1 Introduction 3

2 Linux BSP installation 4
- 2.1 Getting the Yocto image . . . 4
- 2.2 Build Linux BSP . . . 5

3 Board setup 7
- 3.1 S32V234 EVB . . . 7
  - 3.1.1 Jumper settings . . . 7
  - 3.1.2 Setup the SD Card . . . 8
  - 3.1.3 Booting . . . 8
- 3.2 S32V234 EVB SCH-28899 . . . 10
  - 3.2.1 Jumper settings . . . 10
  - 3.2.2 Setup the SD Card . . . 11
  - 3.2.3 Booting . . . 11
- 3.3 S32V234 PCIe . . . 13
  - 3.3.1 Jumper settings . . . 13
  - 3.3.2 Setup the SD Card . . . 13
  - 3.3.3 Booting . . . 13
- 3.4 Other boards . . . 14

1 Introduction

This document explains how to build and install the NXP Linux BSP, where BSP stands for Board Support Package, on S32V234 board and platform. All steps needed to get the S32V234 board and platform running are detailed, including board dip switch settings, step to download an OS image, and instructions on configuring and using the U- Boot bootloader.

2 Linux BSP installation

2.1 Getting the Yocto image

All the steps described below have been run and validated on Ubuntu-18.04 LTS (native or through a virtual machine). It is then recommended to install Ubuntu-18.04 LTS before going through the following sections.

To get the BSP you need to have **repo** installed and its prerequisites. This only needs to be done once.

Update the package manager:

```
sudo <pkg-mgr> update
```

Install dependencies:

- python 2.x - 2.6 or newer:

  ```
  sudo <pkg-mgr> install python
  ```

- git 1.8.3 or newer:

  ```
  sudo <pkg-mgr> install git
  ```

- curl:

  ```
  sudo <pkg-mgr> install curl
  ```

  where <pkg-mgr> is the package manager for your distribution (apt-get or apt for Debian/Ubuntu, yum or dnf for CentOS/Fedora, zypper for Suse).

Install the **repo** utility:

```
mkdir ~/bin
curl http://commondatastorage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
chmod a+x ~/bin/repo
PATH=${PATH}:~/bin
```

Next, download the Yocto Project Environment into your directory:

```
mkdir fsl-auto-yocto-bsp
cd fsl-auto-yocto-bsp
repo init -u https://source.codeaurora.org/external/autobsps32/auto_yocto_bsp -b release/S32V_BSP23.1
repo sync
```

This will download the sources for the latest NXP Auto Linux BSP (from the branch release/S32V_BSP23.1), structured ontop of the Yocto rocko release and upstream NXP QorIQ SDK.

2.2 Build Linux BSP

Enter the directory fsl-auto-yocto-bsp and follow below instructions (available in the *README* file).

Note:

A Yocto build needs at least 50GB of free space and takes a lot of time (2 to 10 hours, depending on the system configuration). It is recommended to use a powerful system with many cores and a fast storage media (SSD is recommended). The recommended RAM size is 8 GB.

If the machine chosen was s32v234evb, after the successful finalization of the build, the results will be placed in the build_s32v234evb_release/tmp/deploy/images/s32v234evb directory. Results for other machines will be placed in similar directories, but named according to the machine name.

This release includes support for:

- Machines: s32v234bbmini, s32v234evb28899, s32v234evb, s32v234pcie, s32v234tmdp, s32v234evbubuntu, s32v234sbc, s32v234ccpb
- images: fsl-image-base, fsl-image-auto, fsl-image-ubuntu, fsl-image-ubuntu-base, fsl-image-ubuntu-ros

In case of building Ubuntu target images, this release includes support for versions 16.04.6 LTS and 18.04.5 LTS.

To build the Linux BSP, please follow the steps:

1. First time setup

   ```
   ./sources/meta-alb/scripts/host-prepare.sh
   ```

2. Creating Build Directories and Testing the Installation

   Now you can create a build directory in the SDK root with:

   ```
   source nxp-setup-alb.sh -m <machine>
   ```

   For example, machine is s32v234evb:

   ```
   source nxp-setup-alb.sh -m s32v234evb
   ```

If targeting the EVB SCH-29288 board.

For PCIE, TMDP, BlueBox Mini, Microsys SBC-S32V234, s32v234ccpb boards use s32v234pcie, s32v234tmdp, s32v234bbmini, s32v234sbc, s32v234ccpb respectively. For the EVB SCH-28899 board, use s32v234evb28899 instead.

When this is done, a “bitbake <imagename>”, e.g.,

```
bitbake fsl-image-auto
```

would be enough to completely build u-boot, kernel, modules, and a rootfs ready to be deployed. Look for a build result in <builddirectory>/tmp/deploy/images/.

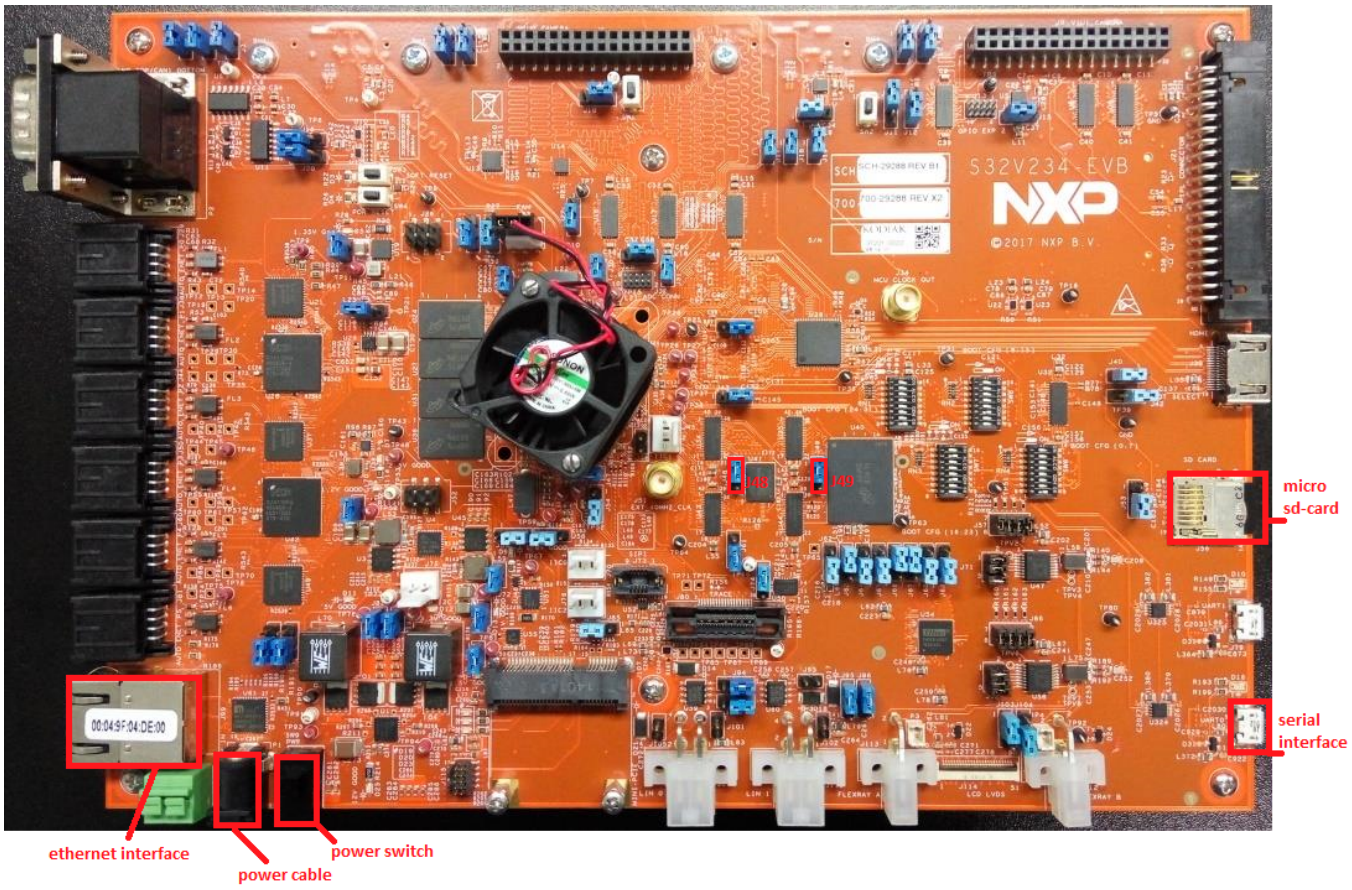
In case of Ubuntu images, by default Ubuntu target version is 18.04.5.
If targeting Ubuntu-16.04 images, add the following line in `<builddirectory>/conf/local.conf` file.

```
UBUNTU_TARGET_VERSION = "16.04.6"
```

Then run “bitbake <imagename>”, where <imagename> is any of the following: fsl-image-ubuntu, fsl-image-ubuntu-base, fsl-image-ubuntu-ros.

3 Board setup

3.1 S32V234 EVB



3.1.1 Jumper settings

SD booting is selected if the jumper on J48 is in the 1-2 position and jumper J49 is in the 1-2 position.
DIP Switches configuration for SD Card boot:

Switch	SW5	SW6	SW7	SW8
ON	8	6	2, 6	8

Switches in groups SW5-SW8 not specified explicitly in the table above should be set to the OFF position

3.1.2 Setup the SD Card

The .sdcard format creates an image with all necessary partitions and loads the bootloader, kernel and rootfs to this image.

Ensure that any partitions on the card are properly unmounted before writing the card image, or you may have a corrupted card image in the end. Also ensure to properly "sync" the filesystem before ejecting the card to ensure all data has been written.

In this example, the device node assigned is DEVSD (a block is 1kB large).

If DEVSD is /dev/sdb, we should execute the following command:

```
export DEVSD=/dev/sdb
```

You can just low level copy the data on this file to the SD Card device using dd as on the following command example:

```
sudo dd if=<image name>.sdcard of=${DEVSD} bs=1M && sync
```

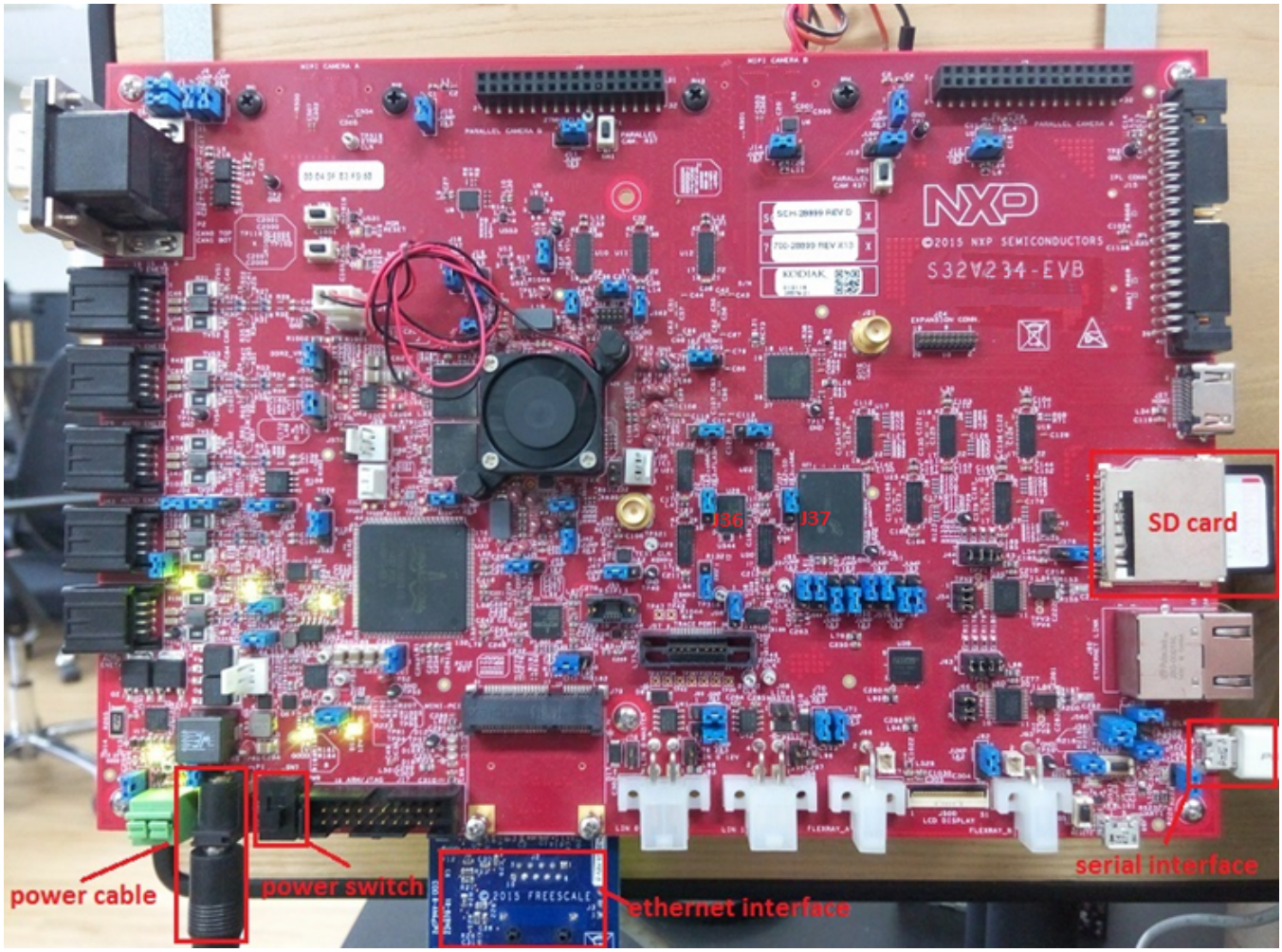
Note:
Yocto supports the generation of both uImage and Image kernel binary formats, but only the 'Image' format works with the current configuration of U-Boot. It is not expected that uImage support will be restored in the future.

3.1.3 Booting

1. Insert a bootable SD Card into the microSD Card slot (J56) on the top of the board.
2. Configure the boot switches to boot from SD.
3. Connect the micro USB cable from a free usb port on your host PC to the J100 connector labeled "RS232/UART0".
4. Start your favorite terminal software (such as Minicom or TeraTerm) on your host PC and configure it for 115200 baud, 8 data bits, no parity, and 1 stop bit with no handshake.
5. Plug power supply (P1) and turn on the power switch (SW9)
6. On powering on, you will instantly see Power LED D25 light up. You will also start to see console output on your terminal window. If everything was done correctly the board should boot completely into Linux. The default login account is root with an empty password.

```
U-Boot 2018.07+gecddad2 (May 22 2019 - 14:11:34 +0000)

CPU:   NXP S32V234 V2.0 at 1000 MHz
Reset cause: unknown reset
Board: s32v234evb29288
I2C:   ready
DRAM:  2 GiB
All (4) cores are up.
MMC:   FSL_SDHC: 0
Loading Environment from MMC... OK

Forcing PCIe to Gen1 operation
phy link never came up
DEBUG_R0: 0x004d2d00, DEBUG_R1: 0x08200000
In:    serial
Out:   serial
Err:   serial
The security module (CSE3) is disabled.
Net:
Warning: System is using default MAC address. Please set a new value
FEC
Hit any key to stop autoboot:  0
switch to partitions #0, OK
mmc0 is current device
8771592 bytes read in 411 ms (20.4 MiB/s)
Booting from mmc ...
22761 bytes read in 15 ms (1.4 MiB/s)
## Flattened Device Tree blob at 82000000
   Booting using the fdt blob at 0x82000000
   reserving fdt memory region: addr=80000000 size=10000
   Loading Device Tree to 000000009fff7000, end 000000009ffff8e8 ... OK

Starting kernel ...

[    0.000000] Booting Linux on physical CPU 0x0
[    0.000000] Linux version 4.19.31-rt18+g9e46b41 (oe-user@oe-host) (gcc version 6.3.1 20170509 (Linaro GCC 6.3-2017.06~dev))
[    0.000000] Boot CPU: AArch64 Processor [410fd034]
[    0.000000] Machine model: Freescale S32V234
.....
Starting rpcbind daemon...done.
starting statd: done
Starting atd: OK
Starting DHCP server: .
starting DNS forwarder and DHCP server: dnsmasq... done.
Starting network benchmark server: netserver.
exportfs: can't open /etc/exports for reading
NFS daemon support not enabled in kernel
Starting internet superserver: xinetd.

Auto Linux BSP 1.0 s32v234evb /dev/ttyLF0

s32v234evb login:
```

3.2 S32V234 EVB SCH-28899



3.2.1 Jumper settings

SD booting is selected if the jumper on J36 is in the 1-2 position and jumper J37 is in the 1-2 position.
DIP Switches configuration for SD Card boot:

Switch	SW500	SW501	SW502	SW503	SW504
ON	2	6	8	8	2, 6

Switches in groups SW500-504 not specified explicitly in the table above should be set to the OFF position

3.2.2 Setup the SD Card

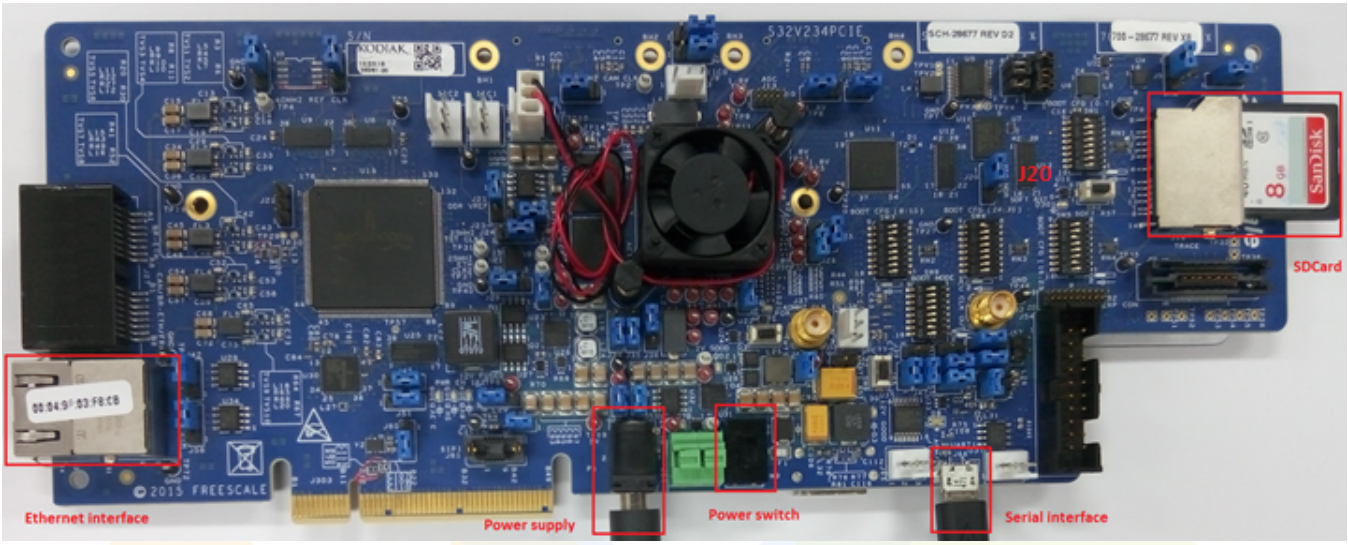
For the S32V234 EVB SCH-28899 board, the same instructions as for the S32V234 EVB board applied. See details in chapter 3.1.2.

3.2.3 Booting

1. Insert a bootable SD Card into the microSD Card slot (J40) on the top of the board.
2. Configure the boot switches to boot from SD.
3. Connect the micro USB cable from a free usb port on your host PC to the J75 connector labeled “RS232/UART0”.
4. Start your favorite terminal software (such as Minicom or TeraTerm) on your host PC and configure it for 115200 baud, 8 data bits, no parity, and 1 stop bit with no handshake.
5. Plug power supply (P1) and turn on the power switch (SW7)
6. On powering on, you will instantly see Power LED D13 light up. You will also start to see console output on your terminal window. If everything was done correctly the board should boot completely into Linux. The default login account is root with an empty password.

```
U-Boot 2018.07+gecddad2 (May 22 2019 - 14:11:34 +0000)

CPU:   NXP S32V234 V1.0 at 1000 MHz
Reset cause: Software Functional reset
Board: s32v234evb28899
I2C:   ready
DRAM:  256 MiB
All (4) cores are up.
MMC:   FSL_SDHC: 0
Loading Environment from MMC... OK
Setting PCIE Vendor and Device ID
Forcing PCIe to Gen1 operation
phy link never came up
DEBUG_R0: 0x0006a200, DEBUG_R1: 0x08200000
In:    serial
Out:   serial
Err:   serial
The security module (CSE3) is disabled.
Net:   FEC
Warning: System is using default MAC address. Please set a new value
Hit any key to stop autoboot: 0
.......................
Starting kernel ...
[    0.000000] Booting Linux on physical CPU 0x0
[    0.000000] Linux version 4.19.31-rt18+g9e46b41 (oe-user@oe-host) (gcc version 6.3.1 20170509 (Linaro GCC 6.3-2017.06~dev))
[    0.000000] Boot CPU: AArch64 Processor [410fd034]
[    0.000000] Machine model: Freescale S32V234
.......................
Starting OpenBSD Secure Shell server: sshd
  generating ssh RSA key...
  generating ssh ECDSA key...
  generating ssh DSA key...
  generating ssh ED25519 key...
done.
Starting rpcbind daemon...done.
starting statd: done
Starting atd: OK
Starting DHCP server: .
starting DNS forwarder and DHCP server: dnsmasq... done.
Starting network benchmark server: netserver.
exportfs: can't open /etc/exports for reading
NFS daemon support not enabled in kernel
Starting internet superserver: xinetd.
Auto Linux BSP 1.0 s32v234evb28899 /dev/ttyLF0
s32v234evb28899 login:
```

3.3 S32V234 PCIe



3.3.1 Jumper settings

SD booting is selected if the jumper on J20 is in the 1-2 position.
DIP Switches configuration for SD Card boot:

Switch	SW1	SW3	SW4	SW5	SW6
ON	8	6	8	2, 6	2

Switches in groups SW1,3-6 not specified explicitly in the table above should be set to the OFF position

3.3.2 Setup the SD Card

For the S32V234 PCIe board, the same instructions as for the S32V234 EVB board applied. See details in chapter 3.1.2.

3.3.3 Booting

1. Insert a bootable SD Card into the microSD Card slot (J19) on the top of the board.
2. Configure the boot switches to boot from SD.
3. Connect the micro USB cable from a free usb port on your host PC to the J66 connector labeled “UART1”.
4. Start your favorite terminal software (such as Minicom or TeraTerm) on your host PC and configure it for 115200 baud,
5. 8 data bits, no parity, and 1 stop bit with no handshake.

6. Plug power supply (P1) and turn on the power switch (SW9)
7. On powering on, you will instantly see Power LED D3 light up. You will also start to see console output on your terminal window. If everything was done correctly the board should boot completely into Linux. The default login account is root with an empty password.

```
U-Boot 2018.07+gecddad2 (May 22 2019 - 14:11:34 +0000)

CPU:   NXP S32V234 V1.0 at 1000 MHz
Reset cause: Software Functional reset
Board: s32v234pcie
I2C:   ready
DRAM:  256 MiB
All (4) cores are up.
MMC:   FSL_SDHC: 0
Loading Environment from MMC... OK
Setting PCIE Vendor and Device ID
Forcing PCIe to Gen1 operation
phy link never came up
DEBUG_R0: 0x0004d900, DEBUG_R1: 0x08200000
In:    serial
Out:   serial
Err:   serial
The security module (CSE3) is disabled.
Net:   FEC
Hit any key to stop autoboot: 0

Starting kernel ...
[    0.000000] Booting Linux on physical CPU 0x0
[    0.000000] Linux version 4.19.31-rt18+g9e46b41 (oe-user@oe-host) (gcc version 6.3.1 20170509 (Linaro GCC 6.3-2017.06~dev))
[    0.000000] Boot CPU: AArch64 Processor [410fd034]
[    0.000000] Machine model: Freescale S32V234
.................
Starting system message bus: dbus.
Starting OpenBSD Secure Shell server: sshd
  generating ssh RSA key...
  generating ssh ECDSA key...
  generating ssh DSA key...
  generating ssh ED25519 key...
done.
Starting rpcbind daemon...done.
starting statd: done
Starting atd: OK
Starting DHCP server: .
starting DNS forwarder and DHCP server: dnsmasq... done.
Starting network benchmark server: netserver.
exportfs: can't open /etc/exports for reading
NFS daemon support not enabled in kernel
Starting internet superserver: xinetd.
Auto Linux BSP 1.0 s32v234pcie /dev/ttyLF1
s32v234pcie login:
```

3.4 Other boards

For instructions on starting up other boards, please see the User Manual.