

# HOWTO: Prepare and boot S32V234 EVB from eMMC

---

S32V234 EVB has 32GB of eMMC memory. This memory can be used as OS drive. In text below is **RED** color used for important notes, **GREEN** for console commands and **BLUE** for filenames. `Courier` font family is used for code/configuration data.

## Requirements

- Prepared SD Card with linux image ([HOWTO: Prepare A SD Card For Linux Boot Of S32V234-EVB Using BSP From VSDK, `https://community.nxp.com/docs/DOC-335023`](#) ) and with **u-boot.s32** file in boot partition.
- Host PC machine with Linux OS, NFS, TFTP server and network connectivity with EVB • NFS shared folder with BSP Linux root file system (the root.tar file located in `[S32DS_Vision]\s32v234_sdk\os\build_content.tar\build_content\v234_linux_build\` )
- tftp server with **Image**, **s32v234-evb.dtb** and **u-boot.s32** files
- putty or other terminal connected to s32v234 EVB (tested with minicom on Linux)

# Procedure

Setup NFS share and TFTP server. Please look at internet for more details about NFS and TFTP

- <https://www.digitalocean.com/community/tutorials/how-to-set-up-an-nfs-mount-on-ubuntu-16-04>
- [How do I install and run a TFTP server? - Ask Ubuntu](https://askubuntu.com/questions/201505/how-do-i-install-and-run-a-tftp-server) (<https://askubuntu.com/questions/201505/how-do-i-install-and-run-a-tftp-server>)

Don't forget add into root file system files **Image**, **s32v234-evb.dtb** and **rootfs.tar** - we will need them later for boot and rootfs partitions. Also make sure that all rootfs files are owned by root.

Here is my entry for **/etc/exports** file for NFS:

```
/rfs 192.168.1.0/24(rw,no_root_squash, sync)
```

and here **/etc/xinetd.d/tftp** file:

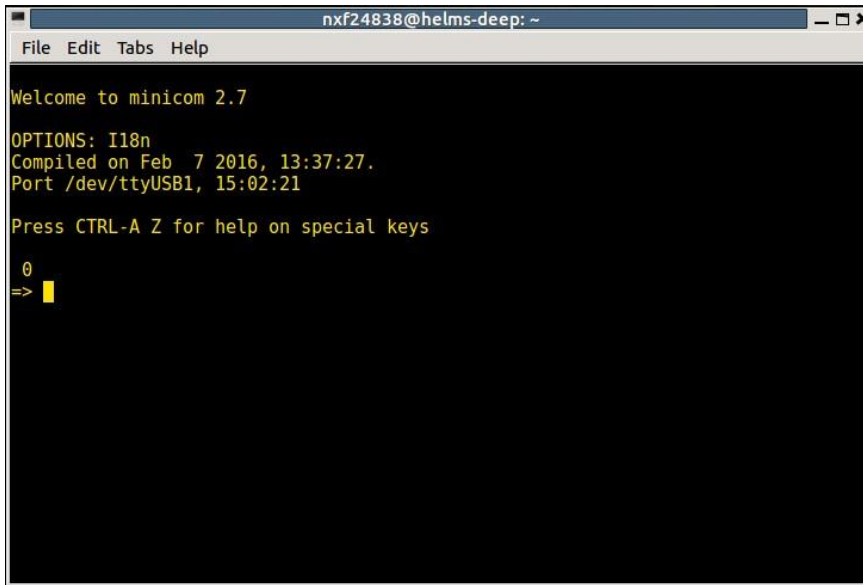
```
service tftp
{
    protocol = udp
    port = 69
    socket_type = dgram
    wait = yes
    user = nobody
    server = /usr/sbin/in.tftpd
    server_args = /tftpboot
    disable = no
}
```

It looks like that in ubuntu is some bug and I have to move tftp files location from /tftpboot to /svr/tftp The tftp file in /etc/xined.d/ remain unchanged. This issue is probably related only to ubuntu.

**Make sure that both servers (nfs,tftp) are accessible from other machine (you can use S32V234 EVB started from SD card for that).**

## HOWTO: Prepare and boot S32V234 EVB from eMMC

I used static IP addresses on PC Host side and EVB. In my case PC has address 192.168.1.1 and EVB 192.168.1.10. You can also use DHCP server - but this is not part of this document. Boot from SD card and stop booting by pressing any key when you see first numbers on terminal window. You can get list of commands by help command.

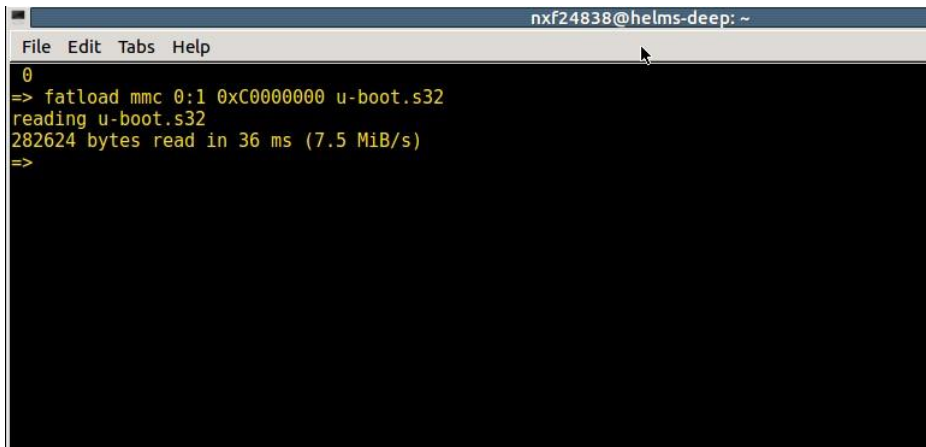


```
nx24838@helms-deep: ~  
File Edit Tabs Help  
Welcome to minicom 2.7  
OPTIONS: I18n  
Compiled on Feb  7 2016, 13:37:27.  
Port /dev/ttyUSB1, 15:02:21  
Press CTRL-A Z for help on special keys  
0  
=> |
```

First - we need to write u-boot.s32 file to eMMC. Unfortunately - there can be active only one storage SD Card or eMMC. We need to copy u-boot.s32 from SD Card to RAM (use RAM address 0xC0000000), deactivate SD Card and connect eMMC. In u-boot console use

**fatload mmc 0:1 0xC0000000 u-boot.s32**

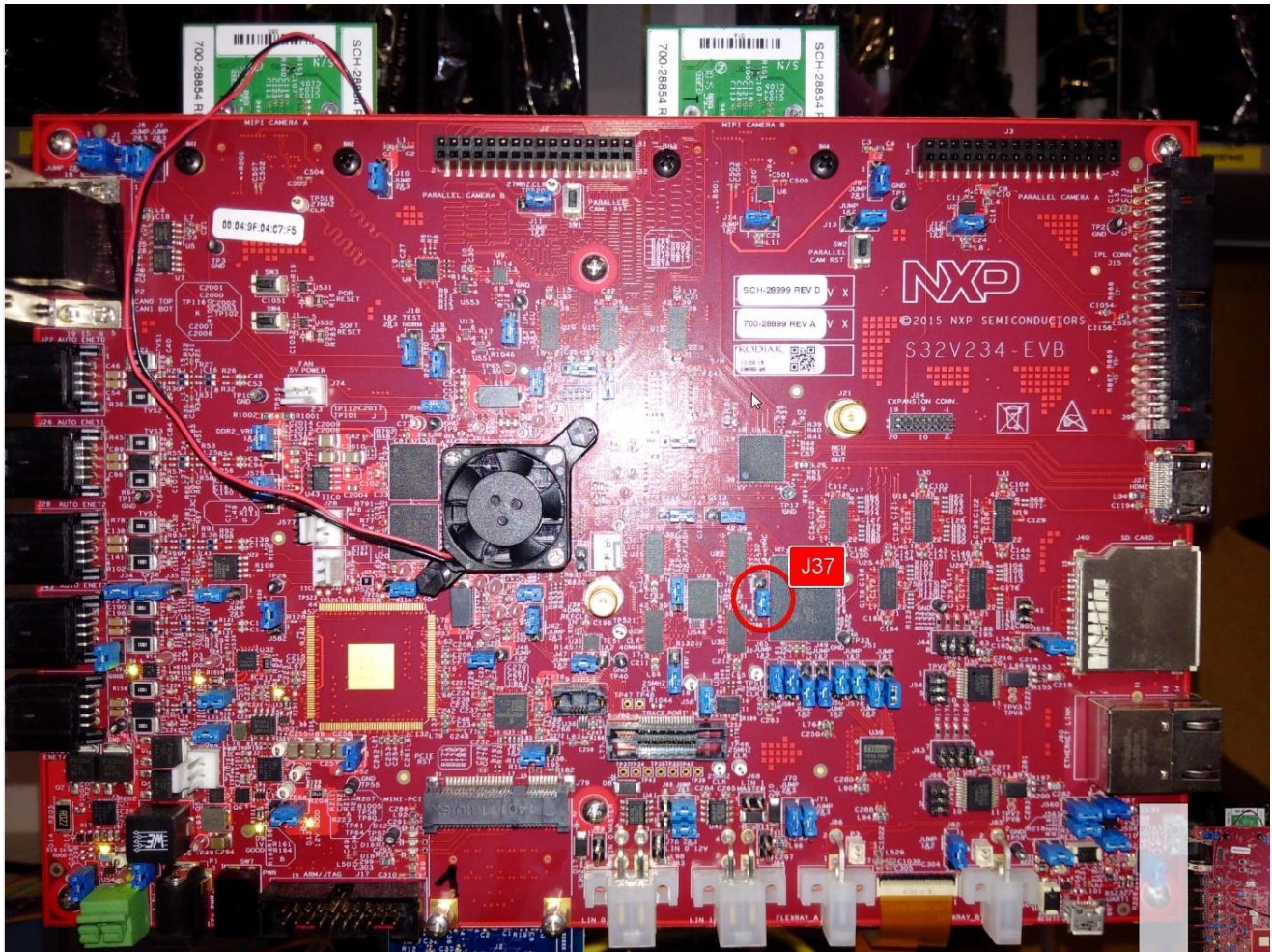
command. Write down the size of loaded file - you will need it later for counting number of sectors. In my case the file size is 282624 (0x45000).



```
nx24838@helms-deep: ~  
File Edit Tabs Help  
0  
=> fatload mmc 0:1 0xC0000000 u-boot.s32  
reading u-boot.s32  
282624 bytes read in 36 ms (7.5 MiB/s)  
=>
```

## HOWTO: Prepare and boot S32V234 EVB from eMMC

Once is u-boot.s32 in the RAM, we can disconnect SD card and connect eMMC by switching J37 jumper from 1-2 to 2-3 (there may be different name for other board versions - but the location is same).



## HOWTO: Prepare and boot S32V234 EVB from eMMC

After switching from SD to eMMC you need to rescan mmc device by command:

### **mmc rescan**

You can verify if eMMC is mapped correctly by

### **mmc info**

```
=> mmc rescan
=> mmc info
Device: FSL_SDHC
Manufacturer ID: fe
OEM: 14e
Name: MMC32
Tran Speed: 52000000
Rd Block Len: 512
MMC version 4.5
High Capacity: Yes
Capacity: 29 GiB
Bus Width: 8-bit
Erase Group Size: 512 KiB
HC WP Group Size: 32 MiB
User Capacity: 29 GiB WRREL
Boot Capacity: 16 MiB ENH
RPMB Capacity: 128 KiB ENH
=>
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7 | VT102 | Offline | ttyUSB1
```

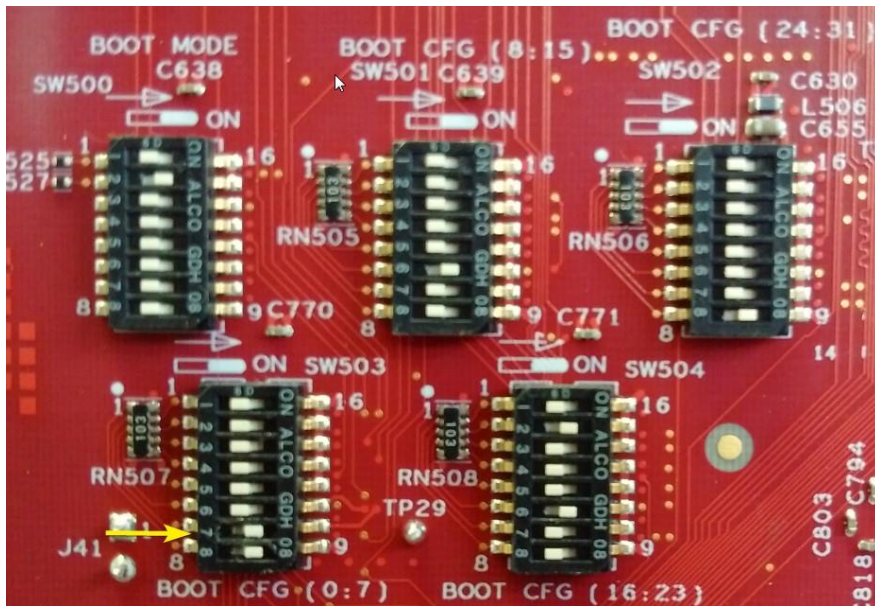
Now we copy u-boot from RAM to emmc. eMMC is located on address 0x1000 - but it is addressed by 512 (0x200) bytes sector size. In this case mmc device address starts on 0x1000/0x200 = 0x08. Number of sectors is (u-boot.s32 filesize) / (sector size) - 0x45000/0x200=0x228. Write u-boot from RAM to eMMC by:

### **mmc write 0xC0000000 0x08 0x228**

```
=> mmc write 0xC0000000 0x08 0x228
MMC write: dev # 0, block # 8, count 552 ... 552 blocks written: OK
=>
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7 | VT102 | Offline | ttyUSB1
```

## HOWTO: Prepare and boot S32V234 EVB from eMMC

Now we can switch boot source from SD to eMMC by switches located on rear side of EVB. The switch name may vary across board version but location is again the same. Turn OFF EVB and remove SD card. For booting from eMMC switch SW503 - BOOT CFG (0:7) 7'th switch from OFF to ON. Switches position for eMMC boot:



## HOWTO: Prepare and boot S32V234 EVB from eMMC

Turn ON EVB and stop again boot in u-boot console. Now we need to configure u-boot for booting from NFS. You can check u-boot variables by

### printenv

command.

Make sure that nfsbootargs has correct EVB ip address, NFS server IP address and path to root file system. In my case - EVB IP is 192.168.1.10, NFS server is 192.168.1.1 and root file system on host PC machine is located in /rfs directory. You can also test network connectivity to PC Host machine by **ping** command.

You can change any of system variables by setenv command. For example - IP address can be changed by:

```
setenv ipaddr -f ipaddr 192.168.1.10
```

Here is printenv output on my EVB:

```
=> printenv
baudrate=115200
boot_fdt=try
boot_mtd=boot1
bootargs=console=ttyLF0,115200 root=/dev/nfs rw ip=192.168.1.10:192.168.1.1::255.255.255.0::eth0:off nfsroot=192.168.1.1:/rfs,nolock
bootcmd=mmc dev ${mmcdev}; if mmc rescan; then if run loadimage; then run mmcboot; else run netboot; fi; else run netboot; fi
bootdelay=3
bootscript=echo Running bootscript from mmc ...; source
console=ttyLF0
ethact=FE
ethaddr=00:1b:c3:12:34:22
fdt_addr=0x82000000
fdt_file=s32v234-evb.dtb
rot_high=0xa0000000
fileaddr=0007ffc0
filesize=52ad
flashboot=echo Booting from flash...; run flashbootargs;cp.b ${kernel_flashaddr} ${loadaddr} ${kernel_maxsize};cp.b ${fdt_flashaddr} ${fdt_addr} ${fdt_maxsize};cp.b ${ramdisk_flashaddr} ${ramdisk_addr} ${ra;
Flashbootargs=setenv bootargs console=${console} root=/dev/ram rw rdinit=/bin/sh;setenv kernel_flashaddr 0x20100000;setenv kernel_maxsize 0xA00000;setenv fdt_flashaddr 0x20B00000;setenv fdt_maxsize 0x100000;
get_cmd=dhcp
image=image
iniIrd_high=0xffffffff
ipaddr=192.168.1.10
jtagboot=echo Booting using jtag...; ${boot_mtd} ${loadaddr} ${ramdisk_addr} ${fdt_addr}
jtagdboot=echo Booting loading Linux with ramdisk from SD...; run loadimage; run loadramdisk; run loadfdt;${boot_mtd} ${loadaddr} ${ramdisk_addr} ${fdt_addr}
loadaddr=0x8007ffc0
loadbootscript=fatload mmc ${mmcdev}:${mmcpart} ${loadaddr} ${script};
loadfdt=fatload mmc ${mmcdev}:${mmcpart} ${fdt_addr} ${fdt_file}
loadimage=fatload mmc ${mmcdev}:${mmcpart} ${loadaddr} ${image}
loadramdisk=fatload mmc ${mmcdev}:${mmcpart} ${ramdisk_addr} ${ramdisk}
loadtftpfdt=tftp ${fdt_addr} ${fdt_file};
loadtftpimage=tftp ${loadaddr} ${image};
mmcargs=setenv bootargs console=${console},${baudrate} root=${mmcroot}
mmcboot=echo Booting from mmc ...; run mmcargs; if test ${boot_fdt} = yes || test ${boot_fdt} = try; then if run loadfdt; then ${boot_mtd} ${loadaddr} - ${fdt_addr}; else if test ${boot_fdt} = try; then ${b;
mmcdev=0
mmcpart=1
mmcroot=/dev/mmcblk0p2 rootwait rw
netargs=setenv bootargs console=${console},${baudrate} root=/dev/nfs ip=dhcp nfsroot=${serverip}:${nfsroot},v3,tcp
netboot=echo Booting from net ...; run netargs; if test ${ip_dyn} = yes; then setenv get_cmd dhcp; else setenv get_cmd tftp; fi; ${get_cmd} ${image}; if test ${boot_fdt} = yes || test ${boot_fdt} = try; the;
netmask=255.255.255.0
nfsboot=echo Booting from net using tftp and nfs...; run nfsbootargs;run loadtftpimage; run loadtftpfdt;${boot_mtd} ${loadaddr} - ${fdt_addr};
nfsbootargs=setenv bootargs console=ttyLF0,115200 root=/dev/nfs rw ip=192.168.1.10:192.168.1.1::255.255.255.0::eth0:off nfsroot=192.168.1.1:/rfs,nolock
ramdisk=rootfs.ulg
ramdisk_addr=0x80000000
script=boot.scr
serverip=192.168.1.1
stderr=serial
stdin=serial
stdout=serial
update_sd_firmware=if test ${ip_dyn} = yes; then setenv get_cmd dhcp; else setenv get_cmd tftp; fi; if mmc dev ${mmcdev}; then if ${get_cmd} ${update_sd_firmware_filename}; then setexpr fw_sz ${filesize} / 1
update_sd_firmware_filename=u-boot.s32
Environment size: 3552/8188 bytes
=>
```

## HOWTO: Prepare and boot S32V234 EVB from eMMC

We are done with configuration - let's boot from NFS by:

**run nfsboot**

It takes a while. At the end you can see login prompt:

```
nxf24838@helms-deep: ~
File Edit Tabs Help
[ 1.293199] fec 40032000.ethernet eth0: Freescale FEC PHY driver [Micrel KSZ9031 Gigabit PHY] (mi_bus:phy_addr=40032000.etherne:07, irq=-1)
[ 1.306561] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
[ 11.303101] Waiting up to 110 more seconds for network.
[ 15.293901] fec 40032000.ethernet eth0: Link is Up - 100Mbps/Full - flow control rx/tx
[ 15.303142] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
[ 15.323246] IP-Config: Complete:
[ 15.326505] device=eth0, hwaddr=00:1b:c3:12:34:22, ipaddr=192.168.1.10, mask=255.255.255.0, gw=255.255.255.255
[ 15.337069] host=192.168.1.10, domain=, nis-domain=(none)
[ 15.342052] bootserver=192.168.1.1, rootserver=192.168.1.1, rootpath=
[ 15.366039] VFS: Mounted root (nfs filesystem) on device 0:15.
[ 15.372802] devtmpfs: mounted
[ 15.375611] Freeing unused kernel memory: 212K (fffffc0006cf000 - fffffc000704000)
[ 15.383184] Freeing alternatives memory: 4K (fffffc000704000 - fffffc000705000)
INIT: version 2.88 booting
Starting udev
udev: Not using udev cache because of changes detected in the following files:
udev: /proc/version /proc/cmdline /proc/devices
udev: lib/udev/rules.d/* etc/udev/rules.d/*
udev: The udev cache will be regenerated. To identify the detected changes,
udev: compare the cached sysconf at /etc/udev/cache.data
udev: against the current sysconf at /dev/shm/udev.cache
[ 16.462628] udevd[581]: starting version 182
[ 17.208929] kjournald starting. Commit interval 5 seconds
[ 17.241874] EXT3-fs (mmcblk0p2): using internal journal
[ 17.242750] FAT-fs (mmcblk0p1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.
[ 17.256732] EXT3-fs (mmcblk0p2): recovery complete
[ 17.261417] EXT3-fs (mmcblk0p2): mounted filesystem with ordered data mode
[ 17.634601] OAL region successfully mapped 5000000@8B000000, Alignment: 0x1000
[ 17.644997] OAL region successfully mapped 5000000@CB000000, Alignment: 0x1000
[ 17.699015] OAL region successfully mapped 3000000@3EB00000, Alignment: 0x8
[ 17.727338] OAL region successfully mapped 1000000@3EB00000, Alignment: 0x8
[ 17.801683] CSI: driver ready -> CSI0 = enabled | CSI1 = enabled.
[ 17.895241] CGD: ready.
[ 17.984370] SEQ: driver ready.
[ 18.103322] FDMA: driver ready in Sequencer-based mode.
[ 18.393540] H264Enc driver ready.
[ 18.700007] random: nonblocking pool is initialized
Populating dev cache
Mon Apr 3 12:59:45 UTC 2017
INIT: Entering runlevel: 5
Configuring network interfaces... ifup skipped for nfsroot interface eth0
run-parts: /etc/network/if-pre-up.d/nfsroot exited with code 1
Starting OpenBSD Secure Shell server: sshd
done.
Starting rpcbind daemon...done.
starting statd: done
Starting network benchmark server: netserver.
NFS daemon support not enabled in kernel
Starting system log daemon...0
Starting kernel log daemon...0
Starting internet superserver: xinetd.

Poky (Yocto Project Reference Distro) 1.8 s32v234evb /dev/ttyLF0
s32v234evb login:
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7 | VT102 | Offline | ttyUSB1
```

Login as root user. Now we need create boot and root file system partition on eMMC by fdisk.



HOWTO: Prepare and boot S32V234 EVB from eMMC

## Boot partition

**fdisk /dev/mmcblk0**

check if there are already some partitions by

**p**

command in fdisk. If there are partitions - delete all of them by

**d**

command. If done - let's create new boot partition with 255 MB size:

**n**

**p**

**1**

**[ENTER]** key for default selection

**+255M**

```
Command (m for help): p
Disk /dev/mmcblk0: 29 GiB, 31138512896 bytes, 60817408 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x18950f33

Device            Boot  Start      End  Sectors  Size Id Type
/dev/mmcblk0p1    2048    524287    522240    255M c W95 FAT32 (LBA)
/dev/mmcblk0p2    524288 60817407 60293120 28.8G 83 Linux

Command (m for help): d
Partition number (1,2, default 2):

Partition 2 has been deleted.

Command (m for help): d
Selected partition 1
Partition 1 has been deleted.

Command (m for help): d
No partition is defined yet!
Could not delete partition 4481025

Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-60817407, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-60817407, default 60817407): +255M

Created a new partition 1 of type 'Linux' and of size 255 MiB.

Command (m for help):
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7 | VT102 | Offline | ttyUSB1
```

HOWTO: Prepare and boot S32V234 EVB from eMMC

## Root FS partition

in fdisk continue with:

n

p

[ENTER] key for default selection

[ENTER] key for default selection

[ENTER] key for default selection

```
Command (m for help): n
Partition type
  p primary (1 primary, 0 extended, 3 free)
  e extended (container for logical partitions)
Select (default p): p
Partition number (2-4, default 2):
First sector (524288-60817407, default 524288):
Last sector, +sectors or +size(K,M,G,T,P) (524288-60817407, default 60817407):
Created a new partition 2 of type 'Linux' and of size 28.8 GiB.
Command (m for help):
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7 | VT102 | Offline | ttyUSB1
```

Change boot partition type from Linux to FAT32.

t

1

c

```
Command (m for help): t
Selected partition 1
Hex code (type L to list all codes): c
If you have created or modified any DOS 6.x partitions, please see the fdisk documentation for additional information.
Changed type of partition 'Linux' to 'W95 FAT32 (LBA)'.
Command (m for help):
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7 | VT102 | Offline | ttyUSB1
```

Write all changes by

w

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read a33iti68 97b .mcbk0: p1 p2
Syncing disks.
```

Create filesystem for boot (vfat) and root fs (ext3) partition:

**mkfs.vfat -n boot /dev/mmcbk0p1**

**mkfs.ext3 -L rootfs /dev/mmcbk0p2**

HOWTO: Prepare and boot S32V234 EVB from eMMC

Now is time for copy some files in new partitions. I created in root home directory boot and rootfs folders:

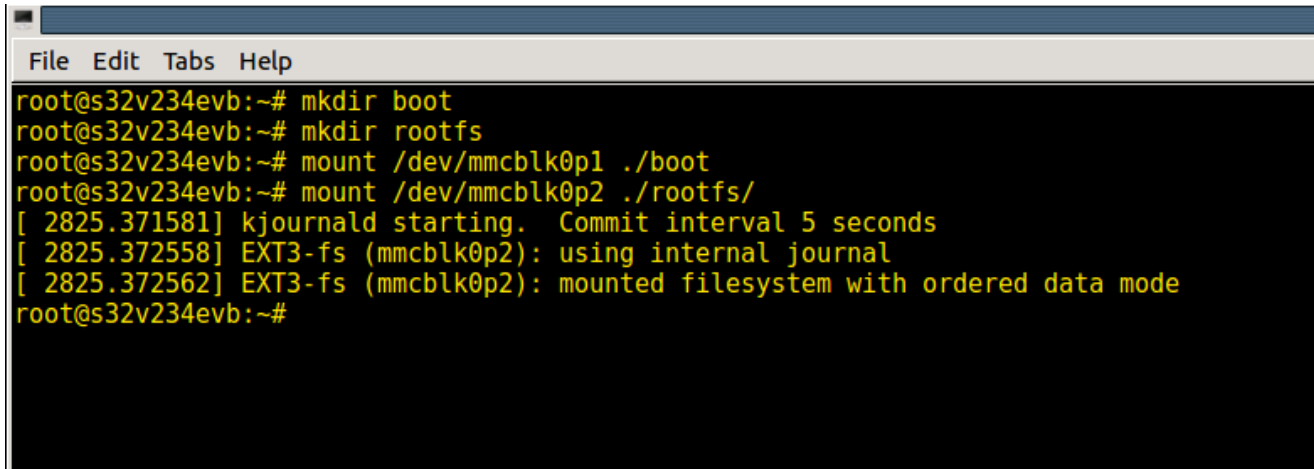
```
cd
```

```
mkdir boot
```

```
mkdir rootfs
```

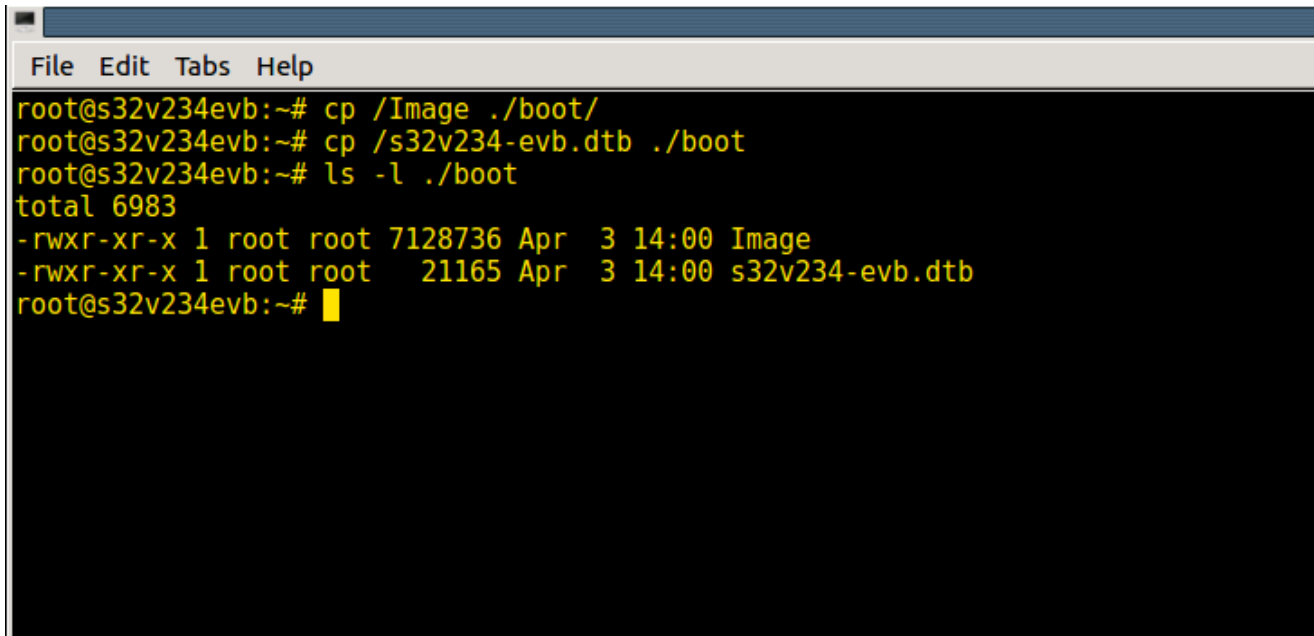
```
mount /dev/mmcblk0p1 ./boot
```

```
mount /dev/mmcblk0p2 ./rootfs
```



```
File Edit Tabs Help
root@s32v234evb:~# mkdir boot
root@s32v234evb:~# mkdir rootfs
root@s32v234evb:~# mount /dev/mmcblk0p1 ./boot
root@s32v234evb:~# mount /dev/mmcblk0p2 ./rootfs/
[ 2825.371581] kjournald starting. Commit interval 5 seconds
[ 2825.372558] EXT3-fs (mmcblk0p2): using internal journal
[ 2825.372562] EXT3-fs (mmcblk0p2): mounted filesystem with ordered data mode
root@s32v234evb:~#
```

Copy [Image](#) and [s32v234-evb.dtb](#) files from root to already mounted mmcblk0p1 partition: `cp /Image ./boot`  
`cp /s32v234-evb.dtb /boot`



```
File Edit Tabs Help
root@s32v234evb:~# cp /Image ./boot/
root@s32v234evb:~# cp /s32v234-evb.dtb ./boot
root@s32v234evb:~# ls -l ./boot
total 6983
-rwxr-xr-x 1 root root 7128736 Apr  3 14:00 Image
-rwxr-xr-x 1 root root  21165 Apr  3 14:00 s32v234-evb.dtb
root@s32v234evb:~# █
```

HOWTO: Prepare and boot S32V234 EVB from eMMC

And now - the final step - untar root file system to mmcblk0p2:

```
tar -xf /rootfs.tar -C ./rootfs
```

```
root@s32v234evb:~# tar -xf /rootfs.tar -C ./rootfs/
root@s32v234evb:~# ls -l ./rootfs/
total 80
drwxr-xr-x  2 root root  4096 Apr  3 14:05 bin
drwxr-xr-x  2 root root  4096 Apr  3 14:05 boot
drwxr-xr-x  2 root root  4096 Aug  4 2016 dev
drwxr-xr-x 31 root root  4096 Apr  3 14:05 etc
drwxr-xr-x  3 root root  4096 Apr  3 14:05 home
drwxr-xr-x  6 root root  4096 Apr  3 14:05 lib
lrwxrwxrwx  1 root root    19 Apr  3 14:05 linuxrc -> /bin/busybox.nosuid
drwx----- 2 root root 16384 Apr  3 13:44 lost+found
drwxr-xr-x  2 root root  4096 Aug  4 2016 media
drwxr-xr-x  2 root root  4096 Aug  4 2016 mnt
drwxr-xr-x  2 root root  4096 Aug  4 2016 proc
drwxr-xr-x  2 root root  4096 Aug  4 2016 run
drwxr-xr-x  2 root root  4096 Apr  3 14:05 s32v234
drwxr-xr-x  2 root root  4096 Apr  3 14:05 sbin
drwxr-xr-x  2 root root  4096 Aug  4 2016 sys
drwxrwxrwt  2 root root  4096 Aug  4 2016 tmp
drwxr-xr-x  9 root root  4096 Apr  3 14:05 usr
drwxr-xr-x  8 root root  4096 Apr  3 14:05 var
root@s32v234evb:~# █
```

## HOWTO: Prepare and boot S32V234 EVB from eMMC

We are done. Disconnect Ethernet cable, reboot and wait for login prompt:

```

nxf24838@helms-deep: ~
File Edit Tabs Help
[ 3.805318] EXT3-fs (mmcblk0p2): recovery complete
[ 3.809753] EXT3-fs (mmcblk0p2): mounted filesystem with ordered data mode
[ 3.816650] VFS: Mounted root (ext3 filesystem) on device 179:2.
[ 3.822606] devtmpfs: mounted
[ 3.825713] Freeing unused kernel memory: 212K (fffffc0006cf000 - fffffc000704000)
[ 3.833278] Freeing alternatives memory: 4K (fffffc000704000 - fffffc000705000)
INIT: version 2.88 booting
Starting udev
[ 4.129721] udevd[574]: starting version 182
[ 4.509859] FAT-fs (mmcblk0p1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.
[ 4.638847] OAL region successfully mapped 5000000@8B000000, Alignment: 0x1000
[ 4.649457] OAL region successfully mapped 5000000@CB000000, Alignment: 0x1000
[ 4.703557] OAL region successfully mapped 300000@3E800000, Alignment: 0x8
[ 4.731898] OAL region successfully mapped 100000@3EB00000, Alignment: 0x8
[ 4.762151] CSI: driver ready -> CSI0 = enabled | CSI1 = enabled.
[ 4.704453] CGD: ready.
[ 4.825107] SED: driver ready.
[ 4.903315] FDMA: driver ready in Sequencer-based mode.
[ 5.033526] H264Enc driver ready.
[ 5.768190] random: dd urandom read with 65 bits of entropy available
Populating dev cache
Mon Apr  3 12:59:45 UTC 2017
Running postinst /etc/rpm-postinsts/100-inetutils-inetd...
Running postinst /etc/rpm-postinsts/101-inetutils-telnetd...
Running postinst /etc/rpm-postinsts/102-inetutils-ftpd...
Running postinst /etc/rpm-postinsts/103-inetutils-tftpd...
update-rc.d: /etc/init.d/run-postinsts exists during rc.d purge (continuing)
Removing any system startup links for run-postinsts ...
INIT: Entering runlevel: 5
Configuring network interfaces... [ 6.803167] fec 40032000.ethernet eth0: Freescale FEC PHY driver [Micrel KS79031 Gigabit PHY] (mii_bus:phy_addr=40032000.ethernet:07, irq=-1)
[ 6.816058] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
udhcpd (v1.23.1) started
Sending discover...
Sending discover...
Sending discover...
No lease, forking to background
done.
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 network benchmark server: netserver.
NFS daemon support not enabled in kernel
Starting system log daemon...0
Starting kernel log daemon...0
Starting internet superserver: xinetd.
[ 23.468046] random: nonblocking pool is initialized

Poky (Yocto Project Reference Distro) 1.8 s32v234evb /dev/ttyLF0
s32v234evb login:
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7 | VT102 | Offline | ttyUSB1
```

# Troubleshooting

## Can't start u-boot console:

You have only about two seconds from power up to interrupt regular boot by pressing key to jump in u-boot console. So - keep trying. Best time for pressing any key is when numbers 2 1 are shown up.

## Can't perform NFS boot:

- check network connectivity between EVB and host PC.
- Try ping host PC from u-boot console. If it doesn't work - check EVB ipaddress by `echo $ipaddr` or `printenv` u-boot command.
- check that nfsbootargs contain correct path/ip address to NFS root file system.
- Check again if your NFS directory is accessible from other machine and it is really s32v234 BSP Linux rootfs. Root fs must not be inside some subfolder.
- check if you can get files from tftp server from other machine. for example `tftp 192.168.1.1 ... get Image`

## Can't perform partitioning of eMMC:

Make sure that `/dev/mmcblk0` is unmounted (in case that there was already some partitions).

## Can't mount partitions on `/dev/mmcblk0`:

Make sure that all files on NFS root file system belongs to root. You can also check boot messages for mount errors related to `/proc` file system.