twam.info

My computer, physics, electronics & photography blog

U-Boot on BeagleBone Black

Posted on August 23, 2014

Understanding the boot up mechanism of the BeagleBone Black is important to be able to modify it. As we later want to change the Linux Kernel itself we need to know how it is started by the BeagleBone.

The AM335x processor on the BeagleBone Black has many configurable options where it can boot from which are all documented in Chapter 26 of the AM335x ARM® Cortex™-A8 Microprocessors Technical Reference Manual. The BeagleBone Black provides by alternative boot sequences which are selectable by the boot switch (S2). In default mode (S2 not pressed) it tries to boot from

1. MMC1 (onboard eMMC),
2. MMC0 (microSD),
3. UART0,
4. USB0.

Usually it will find something in the onboard eMMC and boot from there. If S2 is pressed during power-up the boot sequence is changed to

1. SPI0,
2. MMC0 (microSD),
3. UART0,
4. USB0.

As there is usually nothing bootable found on SPI0 it will boot from the microSD card. The onboard eMMC or an external microSD card have to be formatted in a special way for the AM335x processor to find its boot file, which is described very good on the MMC boot format wiki page at TI. If the AM335x processor finds a valid formatted MMC it searchings for a file named MLO on the first partition and if it is found it boots from that file.

This is very U-Boot kicks in. U-Boot is a very versatile boot loader which can be used on the BeagleBone Black. U-Boot provides this MLO file as a second-stage boot-loader which then loads the actual U-Boot which has to be provided as a file named u-boot.bin in the same directory. U-Boot itself will then look for a file named uEnv.txt for further configuration and then act upon it.

Formatting a microSD card

Now as we have the theoretical background, let's try it our self. The following steps were all done on a (virtual) Ubuntu Linux system. Our microSD card is connected to /dev/sdb as has a size of 16 GB. **The content of the SD card fill be deleted during this procedure.** Instead of calculating the exact partitions sizes needed as explained by the MMC boot format wiki page we use a script which does everything for use. Download, make it executable and run it by

```
wget http://dev.gentoo.org/~armin76/arm/beaglebone/mkcard.sh
chmod +x mkcard.sh
sudo ./mkcard.sh /dev/sdb
```

Be sure to have /dev/sdb pointing to your microSD card and not anything else! In the output of the script you can see the partition table it created and in our case it was a
primary FAT23 partition of 72261 blocks in size and a Linux ext4 partition of 15478627 blocks in size:

```
1024+0 records in
1024+0 records out
1048576 bytes (1.0 MB) copied, 0.447124 s, 2.3 MB/s
Disk /dev/sdb doesn't contain a valid partition table
DISK SIZE - 15931539456 bytes
CYLINDERS - 1936
Checking that no-one is using this disk right now ...
OK

Disk /dev/sdb: 1936 cylinders, 255 heads, 63 sectors/track

sfdisk: ERROR: sector 0 does not have an msdos signature
 /dev/sdb: unrecognized partition table type
Old situation:
No partitions found
New situation:
Units = cylinders of 8225280 bytes, blocks of 1024 bytes, counting fro

   Device Boot Start     End   #cyls    #blocks   Id  System
/dev/sdb1   *      0+      8       9-     72261    c  W95 FAT32 (LBA)
/dev/sdb2          9    1935    1927   15478627+  83  Linux
/dev/sdb3          0       -       0          0    0  Empty
/dev/sdb4          0       -       0          0    0  Empty
Successfully wrote the new partition table

Re-reading the partition table ...

If you created or changed a DOS partition, /dev/foo7, say, then use dd
```

```
to zero the first 512 bytes:  dd if=/dev/zero of=/dev/foo7 bs=512 coun
(See fdisk(8).)
umount: /dev/sdb1: not mounted
mkfs.fat 3.0.26 (2014-03-07)
mkfs.fat: warning - lowercase labels might not work properly with DOS
umount: /dev/sdb2: not mounted
mke2fs 1.42.9 (4-Feb-2014)
Filesystem label=rootfs
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
969136 inodes, 3869656 blocks
193482 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=3963617280
119 block groups
32768 blocks per group, 32768 fragments per group
8144 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632,

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

Now you can mount the boot partition manually or replug the microSD card as most systems will mount it then automatically.

Compiling U-Boot

The next step is to compile U-Boot. As we need to cross compile this for the BeagleBone Black we need a suitable compiler. On Ubuntu this can be installed relatively easy by

```
sudo apt-get install gcc-arm-linux-gnueabihf
```

Now we clone the git repository, create a default configuration and cross compile it for the BeagleBone Black by

```
git clone git://git.denx.de/u-boot.git && cd u-boot
make am335x_boneblack_defconfig
ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- make
```

We should get a MLO and u-boot.bin file in the current directory and can copy those onto the boot partition of microSD card.

Configuring U-Boot

To configure U-Boot we create a file uEnv.txt on the boot partition and fill it, e.g. with

```
console=ttyO0,115200n8
ipaddr=192.168.23.2
serverip=192.168.23.1
rootpath=/exports/rootfs
netargs=setenv bootargs console=${console} ${optargs} root=/dev/nfs nf
netboot=echo Booting from network ...; tftp ${loadaddr} ${bootfile}; t
uenvcmd=run netboot
```

to instruct U-Boot to get the Linux Kernel and device tree via TFTP. The details are depending on your configuration :).

Booting ...

Finally we can now boot from our microSD card. To see the actual boot process a RS232 cable like the TTL-232R-3V3 from FTDI is very handy. Just plug it to the J1 connector and open a serial terminal of your choice. Plug in the microSD card and hold down the S2 boot switch to force booting from the microSD card. You will see hopefully something like

```
U-Boot SPL 2014.10-rc1 (Aug 22 2014 - 19:20:25)
reading u-boot.img
reading u-boot.img


U-Boot 2014.10-rc1 (Aug 22 2014 - 19:20:25)

I2C:   ready
DRAM:  512 MiB
MMC:   OMAP SD/MMC: 0, OMAP SD/MMC: 1
Using default environment

Net:    not set. Validating first E-fuse MAC
cpsw, usb_ether
Hit any key to stop autoboot:  0
switch to partitions #0, OK
mmc0 is current device
SD/MMC found on device 0
reading uEnv.txt
454 bytes read in 4 ms (110.4 KiB/s)
Loaded environment from uEnv.txt
Importing environment from mmc ...
Running uenvcmd ...
Booting from network ...
cpsw Waiting for PHY auto negotiation to complete. done
link up on port 0, speed 100, full duplex
```

```
Using cpsw device
TFTP from server 192.168.23.36; our IP address is 192.168.23.30
Filename 'zImage'.
Load address: 0x82000000
Loading: #############################################################
         #############################################################
         #############################################################
         #############################################################
         #############################################################
         #############################################################
         #############################################################
         #############################################################
         #############################################################
         #############################################################
         #############################################################
         #############################################################
         #############################################################
         #############################################################
         #############################################################
         #############################################################
         #############################################################
         #############################################################
         #############################################################
         ####
         1.5 MiB/s
done
Bytes transferred = 6342632 (60c7e8 hex)
link up on port 0, speed 100, full duplex
Using cpsw device
TFTP from server 192.168.23.36; our IP address is 192.168.23.30
Filename 'am335x-boneblack.dtb'.
Load address: 0x88000000
Loading: #######
         1.2 MiB/s
done
Bytes transferred = 31882 (7c8a hex)
Kernel image @ 0x82000000 [ 0x000000 - 0x60c7e8 ]
## Flattened Device Tree blob at 88000000
   Booting using the fdt blob at 0x88000000
   Loading Device Tree to 8fff5000, end 8ffffc89 ... OK

Starting kernel ...
```

where we first can see of the U-Boot SPL (MLO file) and then afterwards U-Boot itself. Then it reads the uEnv.txt configuration file and acts upon it.

Always boot from microSD

If you always want to boot from your microSD card you can invalidate the boot partition on the onboard eMMC. Then the AM335x processor will always fail back to the second boot option in the default order which is the microSD card. This can be done easily by booting your favourite Linux distro on the BeagleBone Black and issue an

```
sudo dd if=/dev/zero of=/dev/mmcblk1 bs=1024 count=1024
```

Be warned: This will wipe your boot partition on the eMMC!

This entry was posted in BeagleBone Black and tagged BeagleBone Black, U-Boot by twam. Bookmark the permalink [http://www.twam.info/hardware/beaglebone-black/u-boot-on-beaglebone-black] .

About the author

My name is Tobias Müller. I'm interested in computers, physics, electronics and photography. more …

0 Tweet Share Pin it G+1 0

38 THOUGHTS ON "U-BOOT ON BEAGLEBONE BLACK"

Jay Bhukhanwala
on November 7, 2014 at 07:20 said:

Thanks for the wonderful article.

I have a dumb question though (more out of laziness than anything else):

What exactly helps to determine, that MMC1 is onboard eMMC, whereas MMC0 is microSD ?
Is it the BBB Reference Manual ? Where exactly do I find that info ?

twam
on November 7, 2014 at 09:44 said:

5.3.3 in the BeagleBone Black System Reference Manual states that MMC1 is the eMMC.

Javi
on November 24, 2014 at 12:02 said:

Hi, I have a question. It's possible disable boot from a microSD at the same way that you explain "Always boot from microSD"?

Thank you!

> twam
> on November 24, 2014 at 12:12 said:
>
> 'Always boot from microSD' is realized by not putting anything bootable on the eMMC. As long as you have noth bootable on the microSD card it won't boot from there. Also the eMMC is always prefered as long as you don't push the the boot button.
>
> If you want to disable booting from microSSD independent of the boot button or the content of the eMMC you have to alter the config pins.
>
> > Renan
> > on July 14, 2015 at 01:52 said:
> >
> > How can I edit the config pins, and which ones should I change to disable booting from sd card reader?
> >
> > > twam
> > > on July 14, 2015 at 07:36 said:
> > >
> > > You have to 'edit' them with a soldering iron.

Shashwat
on February 1, 2015 at 10:24 said:

Hi All,

I have just started learning embedded linux and wants to have some hand's on experience of embedded devices bringup from scratch.

I have a BBB, and have tried multiple steps to bring up the board but they are very confusing and the result was 0 most of the time.

Can anyone please help me out and guide me on how to bring up embedded devices from scratch.

Thanks in advance
🙂

Thaj
on February 15, 2015 at 19:33 said:

Please follow the steps here
https://eewiki.net/display/linuxonarm/BeagleBone+Black

Carl
on February 3, 2015 at 00:55 said:

Excellent explanation!
Different from your suggestion, I have modified the uEnv.txt on the eMMC so that I can boot via network (NFS) or uSD (using invalid uboot command). My question is: once you boot via NFS or uSD, how can you mount and access the eMMC so that the uEnv.txt on eMMC is modified to boot from eMMC again? I know you can reflash the eMMC, but that will be an overkill and really time-consuming as I have to switch from eMMC and NFS booting often.

twam
on February 5, 2015 at 17:58 said:

You can mount the /dev/mmcblk1p1 partition to your preferred location and change the uEnv.txt file.

Samuel
on February 22, 2015 at 18:27 said:

Hi !
I'm currently struggling doing exactly that : how do you boot from eMMC if the NFS server (or the DHCP server) is down as a faisafe ?
You spoke of invalid u-boot command ... ?

thanks

twam
on March 2, 2015 at 19:46 said:

I didn't mention any invalid u-boot commands. I just said that the if you have a complete invalid boot section on the eMMC, the hardware built-in boot loader of the AM335x processor will fall back to its second boot option, which in that case is the SD card.

Your plan might be possible using u-boot scripting, but I haven't tried this and I doubt that it's worth the effort. You want NFS boot usually for development and then you're pretty sure that your DHCP and NFS server is there. If not, fix it! 😉 If you're using your BeagleBone Black in some kind of production environment you usually want to boot from eMMC or micro SD. If have have to boot from NFS for some reason then you should make sure that it's always there. Having a fallback on microSD would net you also the keep your settings and everything in sync between NFS and microSD/eMMC root.

Thaj
on February 15, 2015 at 19:44 said:

Hey. Thanks Man. This is really helpful.

I just have one doubt. There is MLO, u-boot.bin and uEnv.txt on eMMC already that comes by default with BBB.

Can I just modify the uEnv.txt on eMMC to boot a kernel from microSD ? What i mean is can I just put the FS, Kernel zImage and dtb file on microSD and just modify the uEnv.txt in eMMC to point this kernel of microSD, Will that work? I am asking beacuse I dont want to corrupt the uboot on eMMC, but still i want to boot a custom kernel for device driver development.

twam
on February 19, 2015 at 20:40 said:

Sure, that is possible.

Maurice Sun
on July 10, 2015 at 10:02 said:

Hi,
does anyone who made it successfully can share with your howto?
Thanks.

Ron
on June 4, 2015 at 19:17 said:

Thaj,
I would really like to know if you were able to modify the uEnv.txt file and see the parameters you added to it. I have been racking my brains out on this one.

Juan Benavides
on February 23, 2015 at 21:27 said:

Hello Tobias, do you know if you can access the boot partition /boot/uboot/ to read/write files?
I want both the host PC and the BBB to be able to read/write files in the same directory at /boot/uboot/
With 17% of space left it seems like it should be possible but I'm having problems syncing the files. In other words I can create a file from the Host PC, and the BBB can see it right away. But, if I try the opposite, that is creating a file from the BBB, the Host PC doesn't see it. I tried the command sync to no avail.

```
Filesystem 1K-blocks Used Available Use% Mounted on
/dev/mmcblk0p1 98094 80886 17208 83% /boot/uboot
```

twam
on March 2, 2015 at 19:40 said:

This depends on the system you have booted. If this shares the /boot partition with your host PC this is of course possible. You could also share the partition or folder via your favorite network protocol (NFS, SMB, …)

Viswanath.B
on March 7, 2015 at 11:50 said:

I am newbie to Beaglebone(Black). I am trying to boot it through eMMC without any SDcard. Boot seems to proceed to an extent and stops. Following is the copy of boot steps, can any one suggest if I am missing any thing here.

```
U-Boot 2013.04-dirty (Jul 10 2013 - 14:02:53)

I2C: ready
DRAM: 512 MiB
WARNING: Caches not enabled
NAND: No NAND device found!!!
0 MiB
MMC: OMAP SD/MMC: 0, OMAP SD/MMC: 1
*** Warning - readenv() failed, using default environment

musb-hdrc: ConfigData=0xde (UTMI-8, dyn FIFOs, HB-ISO Rx, HB-ISO Tx, SoftConn)
musb-hdrc: MHDRC RTL version 2.0
musb-hdrc: setup fifo_mode 4
musb-hdrc: 28/31 max ep, 16384/16384 memory
USB Peripheral mode controller at 47401000 using PIO, IRQ 0
musb-hdrc: ConfigData=0xde (UTMI-8, dyn FIFOs, HB-ISO Rx, HB-ISO Tx, SoftConn)
musb-hdrc: MHDRC RTL version 2.0
musb-hdrc: setup fifo_mode 4
musb-hdrc: 28/31 max ep, 16384/16384 memory
USB Host mode controller at 47401800 using PIO, IRQ 0
Net: not set. Validating first E-fuse MAC
cpsw, usb_ether
```

```
Hit any key to stop autoboot: 0
gpio: pin 53 (gpio 53) value is 1
Card did not respond to voltage select!
mmc0(part 0) is current device
Card did not respond to voltage select!
No micro SD card found, setting mmcdev to 1
mmc_send_cmd : timeout: No status update
mmc1(part 0) is current device
mmc_send_cmd : timeout: No status update
gpio: pin 54 (gpio 54) value is 1
SD/MMC found on device 1
reading uEnv.txt
26 bytes read in 3 ms (7.8 KiB/s)
Loaded environment from uEnv.txt
Importing environment from mmc ...
gpio: pin 55 (gpio 55) value is 1
4385024 bytes read in 765 ms (5.5 MiB/s)
gpio: pin 56 (gpio 56) value is 1
24808 bytes read in 52 ms (465.8 KiB/s)
Booting from mmc ...
## Booting kernel from Legacy Image at 80007fc0 ...
Image Name: Angstrom/3.8.13/beaglebone
Image Type: ARM Linux Kernel Image (uncompressed)
Data Size: 4384960 Bytes = 4.2 MiB
Load Address: 80008000
Entry Point: 80008000
Verifying Checksum ... OK
## Flattened Device Tree blob at 80f80000
Booting using the fdt blob at 0x80f80000
XIP Kernel Image ... OK
OK
Using Device Tree in place at 80f80000, end 80f890e7

Starting kernel ...

Uncompressing Linux... done, booting the kernel.
[ 0.196410] omap2_mbox_probe: platform not supported
[ 0.206977] tps65217-bl tps65217-bl: no platform data provided
[ 0.283534] bone-capemgr bone_capemgr.8: slot #0: No cape found
[ 0.320638] bone-capemgr bone_capemgr.8: slot #1: No cape found
[ 0.357748] bone-capemgr bone_capemgr.8: slot #2: No cape found
[ 0.394856] bone-capemgr bone_capemgr.8: slot #3: No cape found
[ 0.414585] bone-capemgr bone_capemgr.8: slot #6: BB-BONELT-HDMIN conflict P8.45
(#5:BB-BONELT-HDMI)
[ 0.424195] bone-capemgr bone_capemgr.8: slot #6: Failed verification
[ 0.444657] omap_hsmmc mmc.4: of_parse_phandle_with_args of 'reset' failed
[ 0.451953] bone-capemgr bone_capemgr.8: loader: failed to load slot-6 BB-BONELT-HDMIN:00A0 (prio 2)
[ 0.518398] pinctrl-single 44e10800.pinmux: pin 44e10854 already requested by 44e10800.pinmux; cannot claim for gpio-leds.7
[ 0.530134] pinctrl-single 44e10800.pinmux: pin-21 (gpio-leds.7) status -22
[ 0.537459] pinctrl-single 44e10800.pinmux: could not request pin 21 on device pinctrl-single
```

Maurice Sun
on July 10, 2015 at 09:56 said:

Hi everyone,
Except for erasing the eMMC's boot section, can anyone share the procedures about how we can configure or modify the boot loader(u-boot) to decide we want to boot from uSD card?

Appreciate!

Brian Tremaine
on November 13, 2015 at 18:04 said:

I'm using a BBB to profile the performance of an algorithm that will go into a AM3358 board. I'm not getting the integer performance I expect with a 1Ghz processor. Is there a way to verify if the cache is enabled on the BBB? I've searched for days and see some references to UBOOT turning cache off but I haven't seen anything on code changes to enable cache.

Thanks,
Brian

Reza
on February 2, 2016 at 22:46 said:

Dear All
I'm beginner with beaglebone. Maybe my question is ridiculous for you…
Some forums noted the following instruction to convert a SD Card image to eMMC flasher:
In /boot/uEnv.txt:
##enable BBB: eMMC Flasher:
#cmdline=init=/opt/scripts/tools/eMMC/init-eMMC-flasher-v3.sh
Change to:
##enable BBB: eMMC Flasher:
cmdline=init=/opt/scripts/tools/eMMC/init-eMMC-flasher-v3.sh

How should I reach to "/boot/uEnv.txt" from a windows computer and delete the # from the mentioned line?!
Regards

twam
on February 3, 2016 at 17:57 said:

Just connect the SD Card to your Windows PC (e.g. by a USB SD Card reader). The Filesystem of /boot should appear as a drive as it is FAT formatted.

Rajesh.D
on February 14, 2016 at 13:28 said:

I am new in bbb, when the time of booting from sd card I got this error..(timeout error).

```
TFTP from server 192.168.0.101; our IP address is 192.168.0.101
Filename 'uImage'.
Load address: 0x80200000
Loading: cpsw Waiting for PHY auto negotiation to complete......... TIMEOUT !
cpsw Waiting for PHY auto negotiation to complete......... TIMEOUT !
cpsw Waiting for PHY auto negotiation to complete......... TIMEOUT !
T cpsw Waiting for PHY auto negotiation to complete......... TIMEOUT !
cpsw Waiting for PHY auto negotiation to complete......... TIMEOUT !
cpsw Waiting for PHY auto negotiation to complete.
```

twam
on February 14, 2016 at 13:31 said:

Looks like you're having at least two problems:

> Your IP and the IP of your TFTP server are the same. And for some reason auto negotiation does not complete. Maybe your cable is not plugged in or the other side does not understand auto negotiation.

eric
on March 3, 2016 at 12:26 said:

Hi,
This is the most clear document I have ever seen these 5 days.
Thank you.

Jakub
on March 6, 2016 at 16:00 said:

Hello,

I have a problem with the boot BBB if you connect to pin P8 (40 41 42 43) UNL motor driver board will not boot. I think it's bootable control pins used for HDMI. I have HDMI banned in uEnv.txt. If I connect UNL to boot everything works. What should I do?

I apologize for my English.

g3blv
on April 3, 2016 at 15:54 said:

Hi Tobias,

I'm running Ubuntu 14.04 on a BBB. I'm booting from a MircroSD card. It works fine when I'm holding down the S2 button but I would like to have it always booting from the MicroSD card. I've cleared the MMC partion with "sudo dd if=/dev/zero of=/dev/mmcblk1 bs=1024 count=1024" but I doesn't boot up. It still boots up fine from the MicroSD card by holding down the S2 button. Are there any other configurations I need to do to make it boot from the MircoSD card?

twam
on April 3, 2016 at 18:24 said:

Usually this should work? Did you try to read the data from the eMMC and check if it is really all 0?

g3blv
on April 4, 2016 at 07:46 said:

Here is the output from lsblk, mmcblk1 beeing the eMMC and mmcblk0 being the MicroSD card. Checking in GParted I can see that mmcblk1 is "unallocated".

```
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
mmcblk1boot0 179:16 0 1M 1 disk
mmcblk1boot1 179:24 0 1M 1 disk
mmcblk0 179:0 0 7.4G 0 disk
├─mmcblk0p1 179:1 0 1M 0 part /boot/uboot
└─mmcblk0p2 179:2 0 7.4G 0 part /
mmcblk1 179:8 0 1.8G 0 disk
```

twam
on April 4, 2016 at 17:24 said:

Maybe you have to wipe the mmcblk1bootX partitions as well. I do not remember seeing them on my BBB, but maybe you have a newer kernel with some special support for those (see https://git.kernel.org/cgit/linux/kernel/git/stable/linux-stable.git/tree/Documentation/mmc/mmc-dev-parts.txt?id=refs/tags/v4.2.4).

Raphael
on April 13, 2016 at 11:43 said:

Hi Tobias,

First, let me congratulate you for the quality of your post. I'm facing a question and I need your opinion if possible.

I've in hand a BBB bought in december 2014. It works fine and boot from the Onboard eMMC (Debian 7). Recently, I've downloaded a new Debian ISO and flashed it on an external SD (Debian 8).
If I insert the SD in the BBB, the board boots from the SD, and not on the eMMC as it should be or explained in yout article:
1. MMC1 (onboard eMMC),
2.MMC0 (microSD),
3.UART0,
4.USB0.

I mention that i do not use the S2 switch. If I remove the SD, it boots from the eMMC.

As you explained, it should boot by defaut from the eMMC, Did I miss Something ?

Thanks.

twam
on April 13, 2016 at 19:57 said:

The BBB should load U-Boot from the eMMC, but the config file there could make U-Boot try to boot a Linux from your SD card. Then it would look like everything is booting from SD card.

Mich
on May 25, 2016 at 03:34 said:

With no SD card, my BBB boots from eMMC just fine.

With a SD card my BBB boots from eMMC, unless I press S2 (in which case it correctly boots from the SD card.)

I'm trying to boot from the SD card without pressing S2, and without disturbing the eMMC partition (so that it can still boot from eMMC in case no SD card is inserted.)

Can this be done by editing the (right now empty) file 'uEnv.txt' in the 'boot' directory of /dev/mmcblk0p1 ?

mmcdev=0
bootpart=0:1
mmcroot=/dev/mmcblk0p1 ro

...but it's not working. Any help is welcome!

My SD card hold Ubuntu Snappy, so:

mmcblk0p1: fa32 boot partition
mmcblk0p2: ext4 system 'A'
mmcblk0p3: ext4 system 'B'
mmcblk0p4: ext4 'writable'

Mich
on May 25, 2016 at 03:36 said:

Do I need to do more than just edit 'uEnv.txt' on the SD card, like do I need to copy MLO or a uboot.img onto the SD card too?

Mich
on May 25, 2016 at 03:55 said:

By 'doesnt work' I mean that the boot sequence continues booting eMMC as if the lines in the SD card's uEnv.txt (mmcdev=0 ...) don't exist.

suraj rajput
on June 17, 2016 at 07:36 said:

Hi all,
Am new to BBB at start am getting this kind of error plz help guys what to do

```
U-Boot SPL 2016.05 (Jun 15 2016 - 14:47:02)
Trying to boot from MMC2
reading args
spl_load_image_fat_os: error reading image args, err - -1
reading u-boot.img
spl_load_image_fat: error reading image u-boot.img, err - -1
Failed to mount ext2 filesystem...
spl_load_image_ext_os: ext4fs mount err - 0
Failed to mount ext2 filesystem...
spl_load_image_ext: ext4fs mount err - 0
```

Fred Wright
on July 28, 2016 at 22:43 said:

Correction:

With S2 pressed, the boot order is:

1. SPI0,
2. MMC0 (microSD),
3. USB0,
4. UART0.

(from table 26-7)