

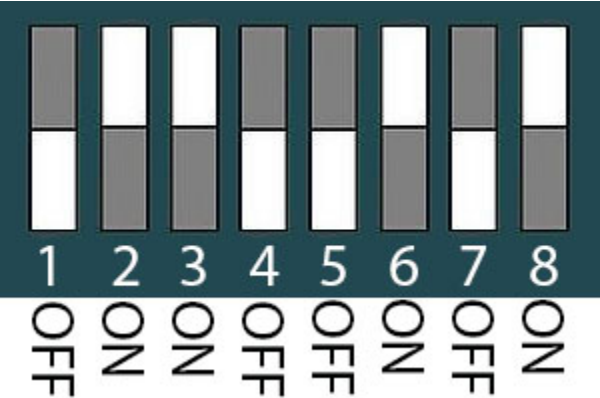
AM57xx: Boot Configuration

The bootloader, one of the key software components included in the BSP, completes the required hardware initializations to download and run operating system images. The boot mode, selected from the S5 dipswitch on the Carrier Board, determines where the hardware looks for the primary bootloader. Set the S5 dipswitch according to which interface you would like to boot from:

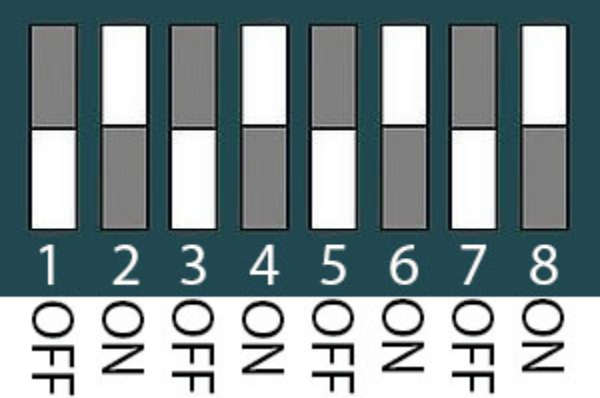
Locate the Boot Switch on your phyCORE-AM57x Carrier Board

Unknown macro: 'scroll-imagemap-macro'

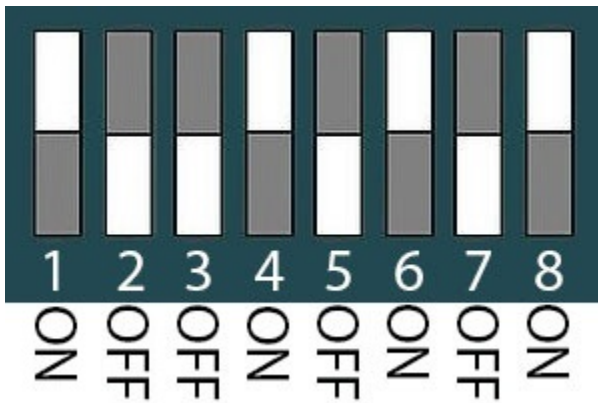
SD Card



eMMC



NAND



Once the boot switch has been set appropriately, press the power button S2 on the phyCORE-AM57x carrier board to power on the board.

Basic Bootloader Settings

After application of power, approximately three seconds are allotted for the user to hit any key which will halt autoboot and enter U-Boot:

```
COM4 - PuTTY
U-Boot SPL 2014.07 (Sep 28 2015 - 13:22:53)
DRA752 ES1.1
reading args
spl_load_image_fat_os: error reading image args, err - -1
reading u-boot.img
reading u-boot.img

U-Boot 2014.07 (Sep 28 2015 - 13:22:53)

CPU   : DRA752 ES1.1
Board: phyCORE-AM57xx
I2C:   ready
DRAM:  2 GiB
MMC:   OMAP SD/MMC: 0, OMAP SD/MMC: 1, OMAP SD/MMC: 2
reading uboot.env

** Unable to read "uboot.env" from mmc0:1 **
Using default environment

SATA link 0 timeout.
AHCI 0001.0300 32 slots 1 ports 3 Gbps 0x1 impl SATA mode
flags: 64bit ncq stag pm led clo only pmp pio slum part ccc apst
scanning bus for devices...
Found 0 device(s).
SCSI: Net:  <ethaddr> not set. Validating first E-fuse MAC
cpsw
Hit any key to stop autoboot:  0
U-Boot#
```



help is a useful tool in U-Boot to show available commands and usage.

Use the following command to verify that all of the environment variables are set as intended:

```
printenv
```

Saving Configurations

After confirming the environment variables are correct, save them and continue on to the next section to set the correct kernel and root filesystem boot location:

```
saveenv
```

Boot Options

The target can be booted from on-board media or from a development host via network. In our standard configuration, the kernel and root filesystem are loaded from SD/MMC.

For booting via network, the development host is connected to the phyCORE-AM57x development kit with a serial cable and via Ethernet; the embedded board boots into the bootloader, then issues a TFTP request on the network and boots the kernel from the TFTP server on the host. Then, after decompressing the kernel into RAM and starting it, the kernel mounts its root filesystem via the NFS server on the host. This method is especially useful for development purposes as it provides a quick turnaround while testing the kernel and root filesystem.

Stand-Alone eMMC Boot

To configure U-boot to boot the kernel from eMMC, modify the *boot_mmc* environment variable and set the *bootcmd* environment variable to make it the default setting:

eMMC boot environment settings

```
setenv boot_mmc 'run findfdt; setenv mmcdev 1;setenv bootpart 1:2;setenv finduuid 'part uuid mmc 1:2 uuid';run
envboot;run mmcboot;setenv mmcdev 0;setenv bootpart 0:2; setenv finduuid 'part uuid mmc 0:2 uuid'; run mmcboot;'
setenv bootcmd 'run boot_mmc'
saveenv
```

Remote Boot

To configure U-Boot to boot the kernel from TFTP and mount the root filesystem from NFS, configure the network as described in the Network Configuration section below and set the *bootcmd* environment variable to make it the default setting:

You can check the target's default environment settings by running the following:

```
printenv
```

The *ethaddr* variable is the MAC id of the target. This is a pre-programmed value which is read from the E-fuse and matches the sticker on the SOM. Set U-boot's network environment variables to match your required network settings:

```
setenv ipaddr ###.###.###.###
setenv serverip ###.###.###.###
setenv gatewayip ###.###.###.###
setenv netmask ###.###.###.###
setenv tftploc <TFTP image location>
setenv rootpath /<NFS mount location>
```

- **ipaddr** - A dedicated IP address for the SOM. This is crucial if TFTP will be used for updating the device's images at any point.
- **serverip** - IP address of the host or another machine. serverip corresponds to where the TFTP directory, if it exists, is located.
- **gatewayip** - Gateway IP for the network. This is only necessary if the TFTP directory is located on another network.
- **netmask** - Netmask for the network: typically 255.255.255.0. This is only necessary if the TFTP directory is located on another network.
- **tftploc (required for TFTP)** - Location of the path to the images on the TFTP server on the host system, setup in Section 4.4.1. Set the variable accordingly by referencing the following examples:

File Path	U-Boot Command
-----------	----------------

/var/lib/tftpboot/PHYTEC/am57xx/PD18.2.0	setenv tftploc PHYTEC
/var/lib/tftpboot	setenv tftploc

- **rootpath (required for NFS)** - Location of the path to the NFS Directory on the host system, set up in Section 4.4.2. Ex: /home/<user>/NFS

Network boot environment settings

```
setenv bootcmd 'run boot_net'
saveenv
```

Stand-Alone SD/MMC Card Boot

By default, the phyCORE-AM57x kit is set up to boot the Linux kernel and root filesystem from SD. If switching from another boot configuration back to SD, modify the *boot_mmc* environment variable and set the *bootcmd* environment variable to make it the default setting:

SD/MMC Card boot environment settings

```
setenv boot_mmc 'run findfdt; setenv mmcdev 0;setenv bootpart 0:2;setenv finduuid 'part uuid mmc 0:2 uuid';run
envboot;run mmcboot;setenv mmcdev 1;setenv bootpart 1:2; setenv finduuid 'part uuid mmc 1:2 uuid'; run mmcboot;'
setenv bootcmd 'run boot_mmc'
saveenv
```

Custom Boot

Unique boot configurations can be created by defining the desired environment variable settings and setting *bootcmd* to run its contents. The following is an example:



Example

Boot the Linux Kernel via TFTP with Root Filesystem on SD:

```
setenv customboot 'tftp ${loadaddr} ${tftploc}${bootfile}; tftp ${fdtaddr} ${tftploc}${fdtfile}; run
mmcargs; bootz ${loadaddr} - ${fdtaddr}'
setenv bootcmd 'run customboot'
saveenv
```