**Embedded Linux Boot Sequence**

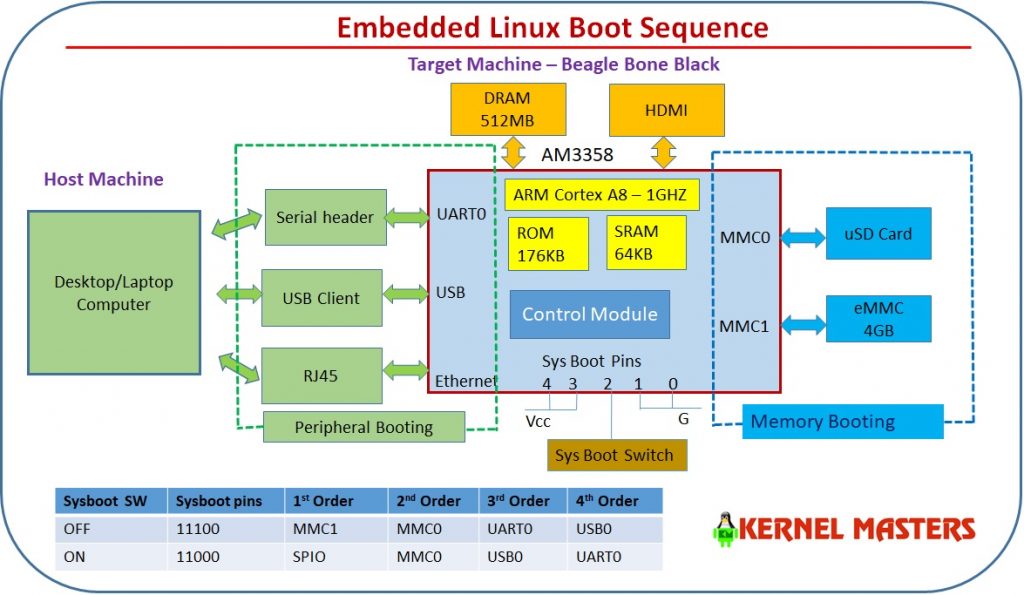
In AM335x the ROM code serves as the bootstrap loader, sometimes referred to as the Initial program Loader (IPL) or the Primary Program Loader (PPL) or ROM Program Loader (RPL).

The booting is completed in two consecutive stages by U-Boot binaries.

1. *The binary for the 1st U-Boot stage is referred to as the Secondary Program Loader (SPL) or the MLO.*
2. *The binary for the 2nd U-Boot stage is simply referred to as U-Boot.* SPL is a non-interactive loader and is a specially built version of U-Boot. It is built concurrently when building U-Boot

**Memory Booting:**Booting the device by starting code stored on permanent memories like flash-memory or memory cards. This process is usually performed after either device cold or warm reset.

**Peripheral Booting:**Booting the device by downloading the executable code over a communication interface like UART, USB or Ethernet. This process is intended for flashing a device.



**Booting the SPL**

The ROM code can load the SPL image from any of the following devices:

1. Memory booting with MMC
2. Peripheral booting with UART

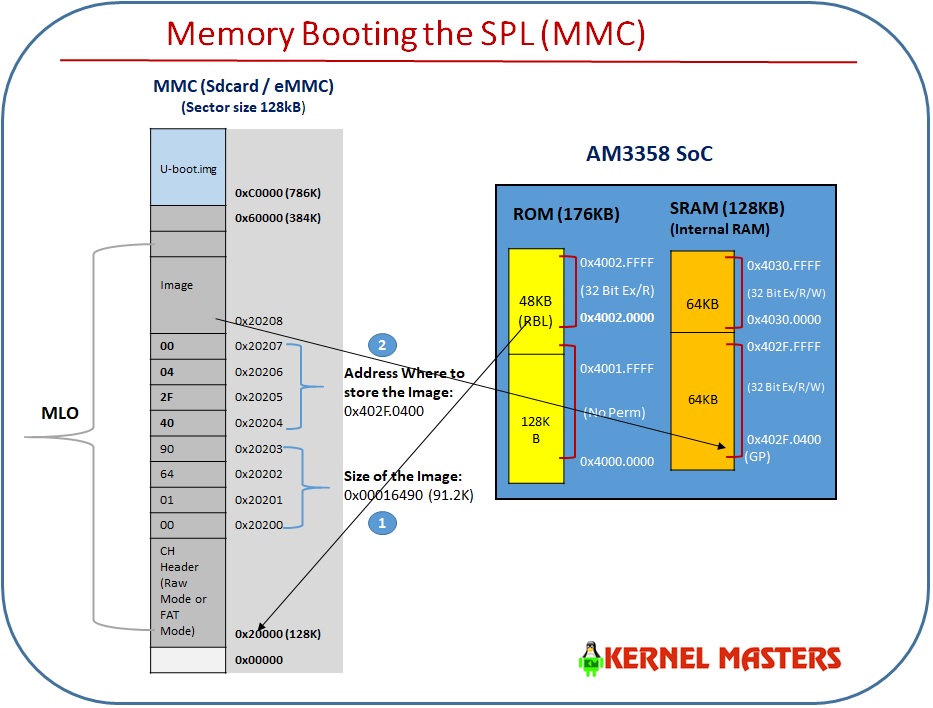
1. Memory Booting with MMC

The image should have the Image header. The image header is of length 8 byte which has the load address (Entry point) and the size of the image to be copied. RBL would copy the image, whose size is given by the length field in the image header, from the device and loads into the internal memory address specified in the load address field of Image header.

When using memory boot a header needs to be attached to the SPL binary indicating the load address and the size of the image. SPI boot additionally requires endian conversion before flashing the image

**The ROM Code supports booting from MMC / SD cards in the following conditions:**

* MMC/SD Cards compliant to the Multimedia Card System Specification and Secure Digital I/O Card Specification of low and high capacities.
* MMC/SD cards connected to MMC0 or MMC1.
* Support for 3.3/1.8 V on MMC0 and MMC1.
* Initial 1-bit MMC Mode, optional 4-bit mode, if device supports it.
* Clock Frequency: identification mode: 400 KHz; data transfer mode up to 10 MHz.
* **File system mode**(FAT12/16/32 supported with or without Master Boot Record), image data is read from a booting file.
* **Raw mode,**image data read directly from sectors in the user area.
  + In raw mode the booting image can be located at one of the four consecutive locations in the main area offset **0x0 / 0x20000 (128KB) / 0x40000 (256KB) / 0x60000 (384KB).**



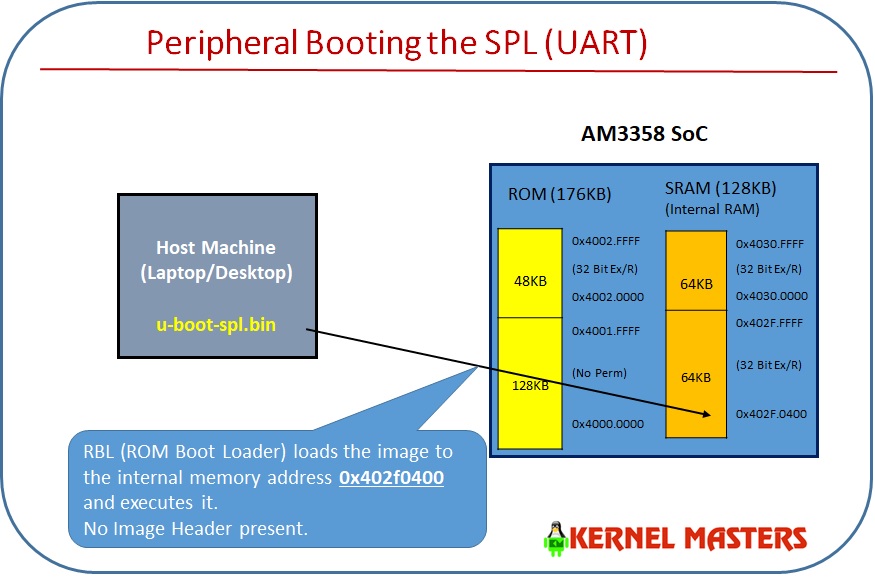
2. Peripheral Booting with UART

RBL loads the image to the internal memory address 0x402f0400 and executes it. No Image Header present.

When using peripheral boot (UART) there can be no header as the load address is fixed.

**The ROM Code supports booting from UART in the following conditions:**

* UART boot uses UART0.
* UART0 is configured to run at 115200 baud, 8-bits, no parity, 1 stop bit and no flow control.
* UART boot uses x-modem client protocol to receive the boot image.
* Utilities like hyperterm, teraterm, minicom can be used on the PC side to download the boot image to the board.
* With x-modem packet size of 1K throughout is roughly about 4KBytes/Sec.
* The ROM code will ping the host 10 times in 3s to start x-modem transfer. If host does not respond, UART boot will timeout.
* Once the transfer has started, if the host does not send any packet for 3s, UART boot will time out.
* If the delay between two consecutive bytes of the same packet is more than 2ms, the host is requested to re-transmit the entire packet again.
* Error checking using the CRC-16 support in x-modem. If an error is detected, the host is requested to re-transmit the packet again.



Bootloader Images:

|  |  |  |  |
| --- | --- | --- | --- |
| Image Name | Image Size | Image Header | Purpose |
| spl/u-boot-spl | 2.2M | No | The binary of SPL ELF Image. |
| spl/u-boot-spl.bin | 91264 Bytes | No | The second-stage bootloader  (a stripped-down version of  u-boot that fits in SRAM). |
| spl/u-boot-spl.map | 227K | No | contains the information for each symbol. |
| MLO | 91784 Bytes | Yes, GP Header | spl/u-boot-spl.bin with a GP image  header prepended to it. |
| U-boot.bin | 474K | No | is the binary compiled U-Boot bootloader |
| u-boot.map | 674K | No | contains the memory map for each symbol |
| U-boot.img | 474K | Yes. GP header | contains u-boot.bin along with an additional  header to be used  by the boot ROM to determine  how and where to load and execute U-Boot. |