Linux Board Port Overview for Sitara AM-Class Devices: AM33x, AM43x, and AM57x

What is a Linux Board Port?

Presentation overview

- What is a Linux board port?
- Linux board port process when migrating from a TI EVM to a custom board
- Linux board port process resources
- Conclusion

Disclaimer

This presentation covers Sitara catalog embedded processors in an overview fashion. While the topics presented generally apply to all the processors, there will be areas that are different between the processors. Please review the processor TRM, Datasheet and TI Processors Linux SDK for specific information and examples.

What is a Linux Board Port?

Section overview: What is a Linux board port?

- The typical perception of porting an OS to a custom board
- Finding an abstraction for board porting
- The base port concept
- TI Processor SDK Linux board port abstraction
- Implementing a Linux board port

Board port questions you might have

Where to start?

Processor initialization code?

How do I build code?

Driver support?

Architecture support?

OS abstraction? (OSAL)

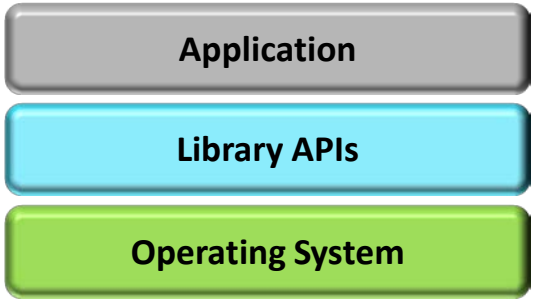
Processor privileged modes?

Interrupt setup?

Finding an abstraction for board porting

- Abstractions have been used for awhile to accelerate porting to a new platform.
- Applications are typically written to this standard to make sure they are portable.
- One example you have heard of:

  IEEE Std 1003.1-2007 Portable Operating System Interface for Unix (POSIX)



POSIX APIs provide an OS abstraction layer

Linux application hardware abstraction

- “Everything is defined as a file in Linux.”
- Files are accessed with the very basic APIs open(), read(), write(), close(), and ioctl() file I/O abstraction.
- Code example: Writing to file in the /home directory on a TI EVM running Linux.

```
#include <stdio.h>

int main () {
FILE *fd;

fd = fopen("/home/root/example.txt","w");
fprintf(fd, "example of a file write\n");
fclose(fd);

}
```

```
root@am335x-evm:/proc# mount
/dev/mmcblk0p2 on / type ext4 (rw,relatime,data=ordered)
```

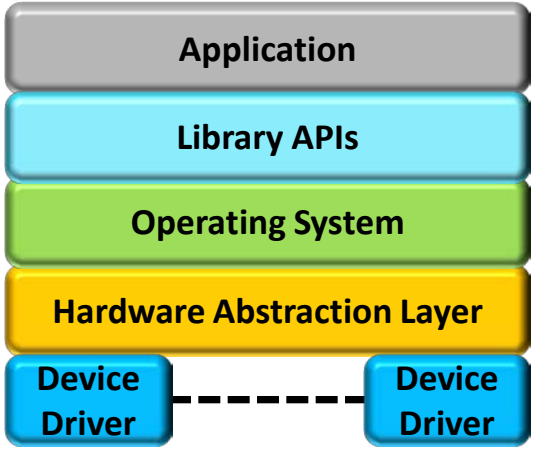
```
root@am335x-evm:~# ls /dev/mmc*
/dev/mmcblk0
/dev/mmcblk0p1
/dev/mmcblk0p2
/dev/mmcblk1boot0
/dev/mmcblk1boot1
/dev/mmcblk1
/dev/mmcblk1p1
/dev/mmcblk1p2
```

SD Card on EVM

- The directory path is abstracted to the mounted root file system, which is abstracted to a particular device.

OS Abstraction Layer (OSAL)

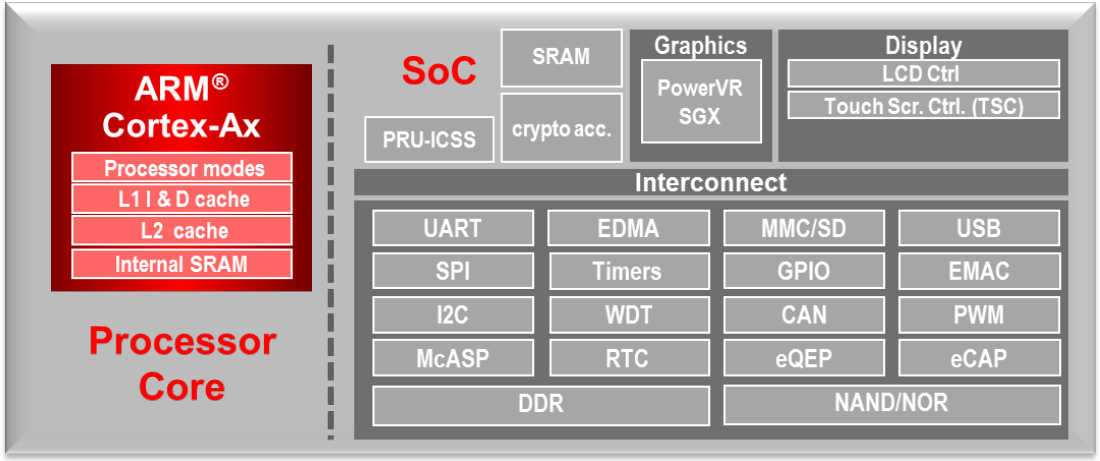
- As with application porting, abstractions are used to accelerate porting an OS to a new platform.
- Some operating systems provide a Hardware Abstraction Layer (HAL), also known as an Operating System Abstraction Layer (OSAL).



Linux does not have a single layer or conventional HAL.

So how then is Linux put on new platforms?

TI processors Linux board port abstraction



- Architecture Support
- Processor Initialization Code
- Processor Privileged Modes
- Interrupt Setup

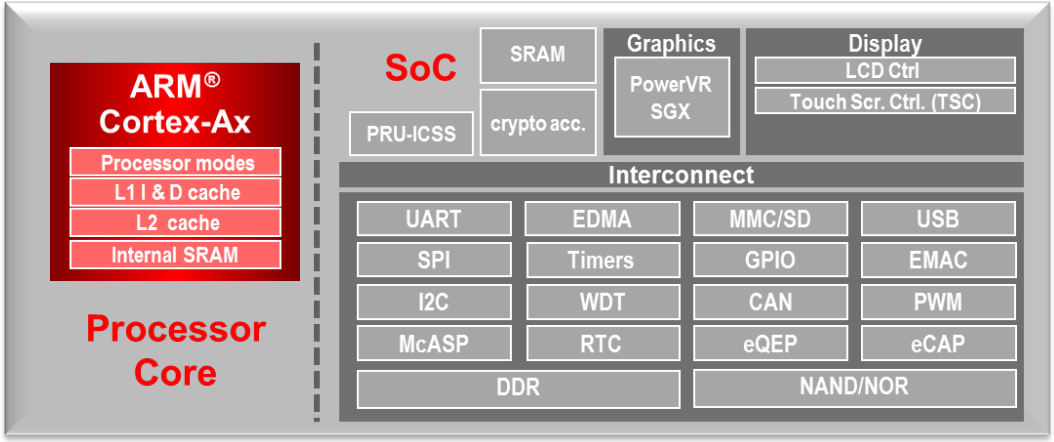
- HW or OS Abstraction Layer (HAL/OSAL)
- Driver Support

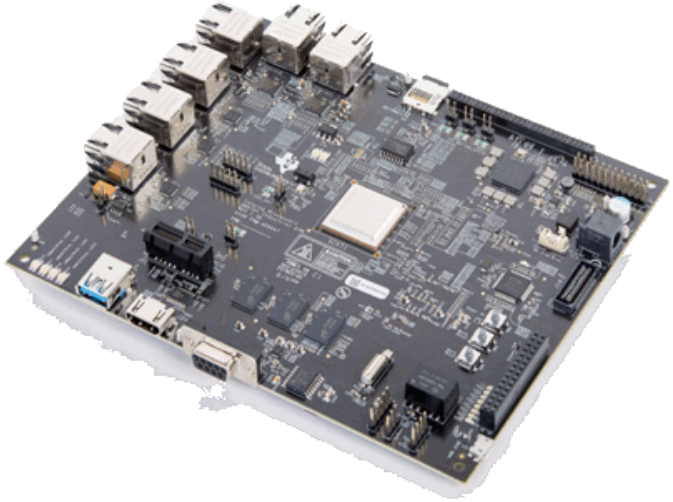
Base port concept

The Base Port = Processor Core + SOC + TI EVM

Known good: This is the starting point for custom boards.

ARCH SOC TI EVM

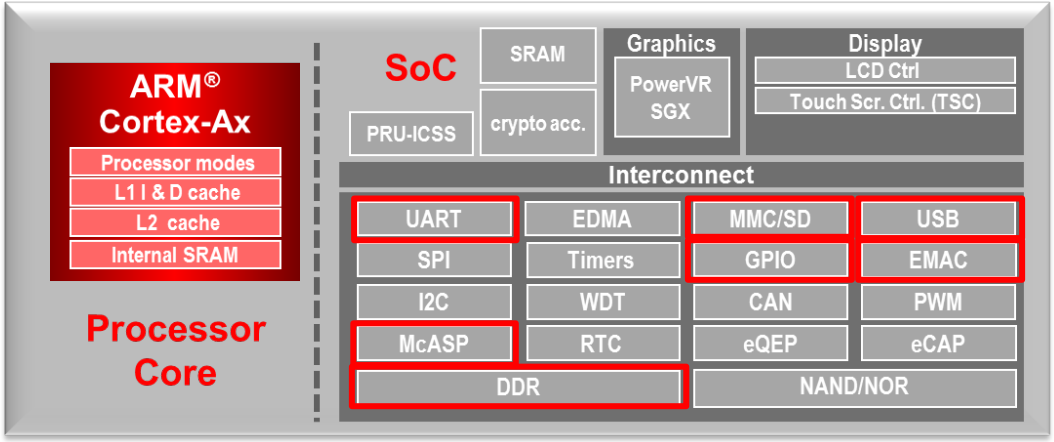


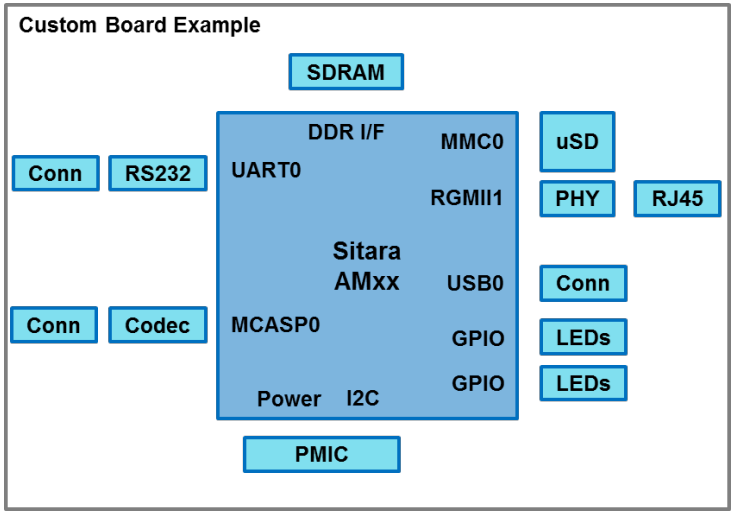


Implementing a Linux board port

Board port = Base port +/- custom board peripheral set and kernel requirements

ARCH SOC Custom board





Section Conclusion: What is a Linux board port?

- The typical perception of porting an OS to a board
- Finding an abstraction for board porting
- TI Processor SDK Linux board port abstraction
- The base port concept
- Implementing a Linux board port

For more information

- Processor SDK Training Series: https://training.ti.com/processor-sdk-training-series
- Linux Board Port for Sitara AM-Class Devices Training Series https://training.ti.com/linux-board-port-sitara-series
- Giving Linux the Boot https://training.ti.com/sitara-arm-processors-boot-camp-giving-linux-boot
- Pin Mux Tool training: https://training.ti.com/pinmux-v4-cloud
- Device Tree Overview https://training.ti.com/debugging-embedded-linux-locate-device-driver
- For questions about this training, refer to the E2E Community Forums for Sitara Processors at http://e2e.ti.com/support/arm/sitara_arm/f/791/t/277411