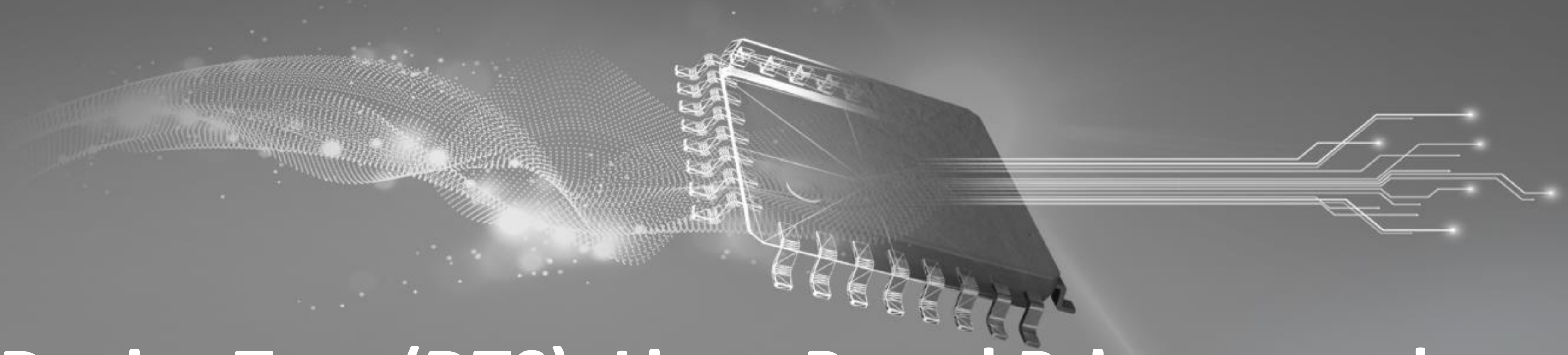


Device Tree (DTS), Linux Board Bring up and Kernel version changing on Embedded systems, a review of some lessons learned processors

Schuyler Patton SW Applications Sitara™ Processors

Texas Instruments

Agenda

- Who am I
- Linux bring up on a new derivative board based on an existing board.
- Discuss boot flow and what does Linux need to boot on a board
- Discuss the Board DTS File and its components
- Introduce the “hello world” concept
- Creating and booting a hello_world.dts
- Lifecycle of the LTS kernel

Schuyler Patton

- Have been working on embedded software with Texas Instruments for 20+ years

- Currently a member of the Sitara Processors Linux Applications team and have been for the past 11 years

- Supporting Linux, networking and board porting on TI SOCs

Target audience for this discussion

- Specifically would to like to address this presentation to those new to Linux and are doing their first board port.

- Anybody interesting in looking at how Linux is “Bound” to an embedded processor on a board to make an embedded system. I will just call the embedded system the board from here.

Level set of the discussion

- This discussion will be about a method to simplify the bring up process for a new board or kernel.

- This is about getting to a minimal number of elements to get a basic boot to a prompt.

- An applications engineer put this presentation together, not a kernel developer. This distinction is sometimes blurred though.

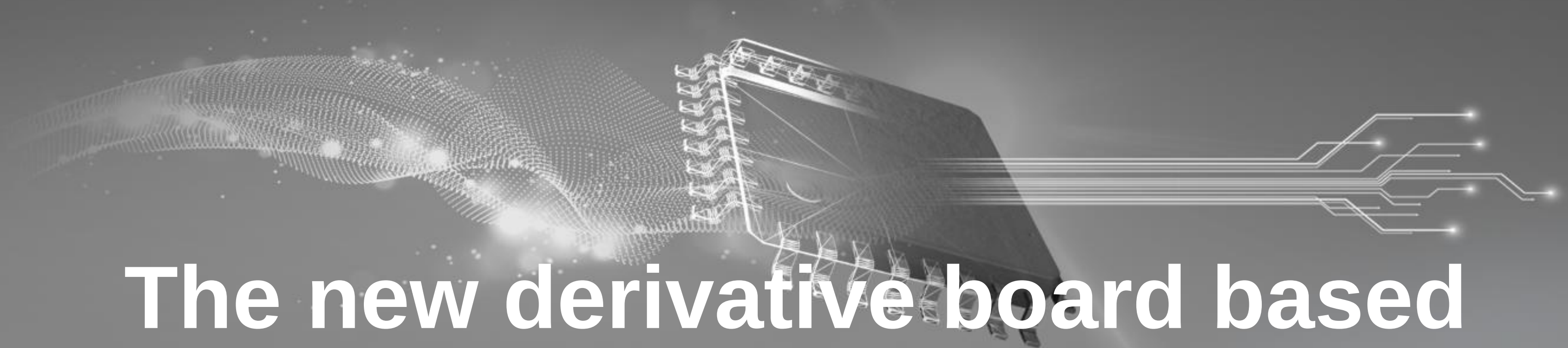
- Perhaps another time and presentation we can discuss additional strategies that the new board is working as intended.

Device Tree Files

- This discussion will be discussing device tree files. This presentation is written with the assumption that the audience has a little bit of background in what the DTS file is.

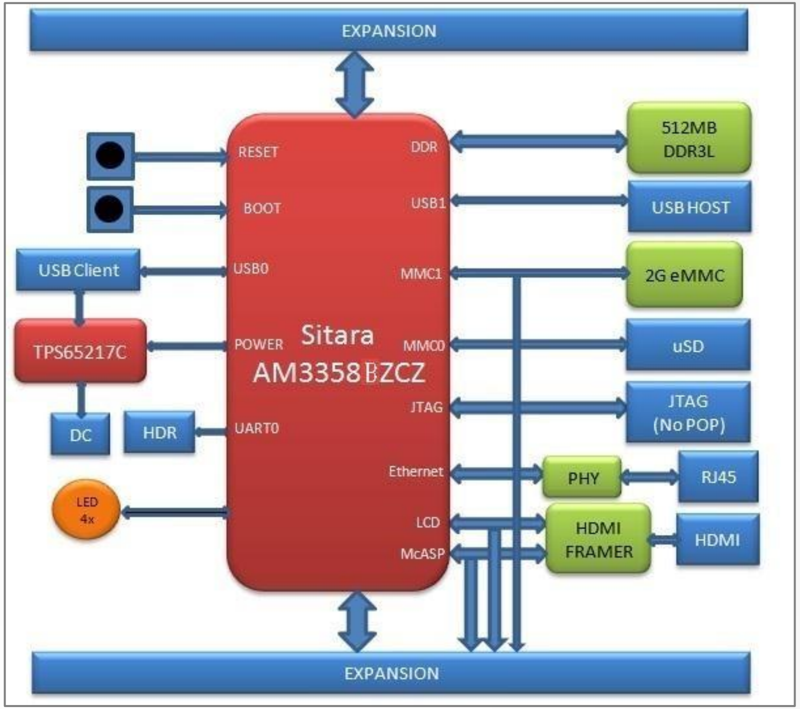
- For a background on Device Tree please look at the tutorial presentation from Free Electrons which is awesome

  - https://elinux.org/images/f/f9/Petazzoni-device-tree-dummies_0.pdf



The new derivative board based on an existing board relationship for Linux bring up

New Derivative Board Based On An Existing Board



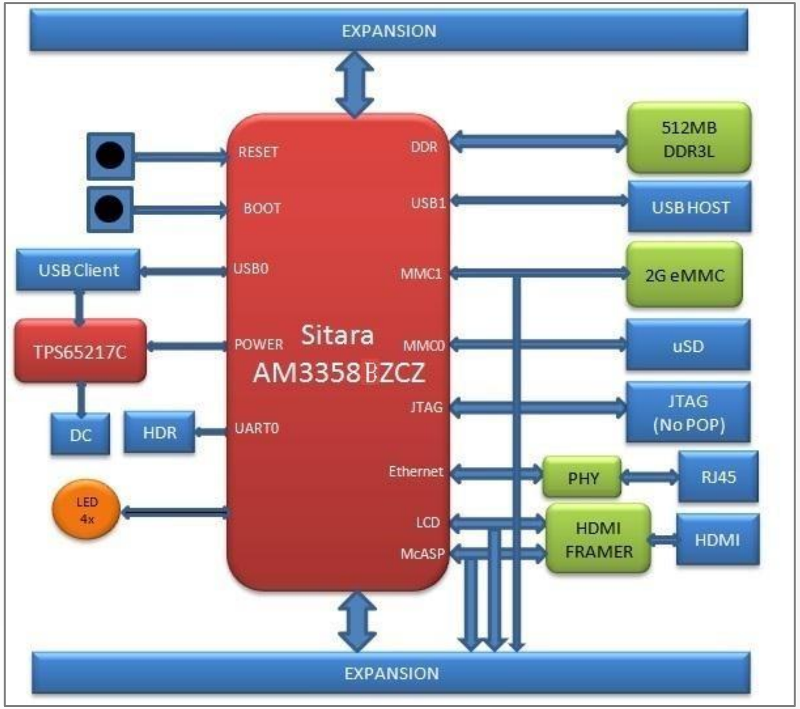
https://github.com/beagleboard/beaglebone-black/wiki/System-Reference-Manual

Linux Community Boards

SOC Vendor Evaluation Boards

Existing Design Board Inventory

New Board Based On An Existing Board



https://github.com/beagleboard/beaglebone-black/wiki/System-Reference-Manual

New Additional Peripherals on new board

PHY

RJ45

UART

UART

Remove these Peripherals on new board

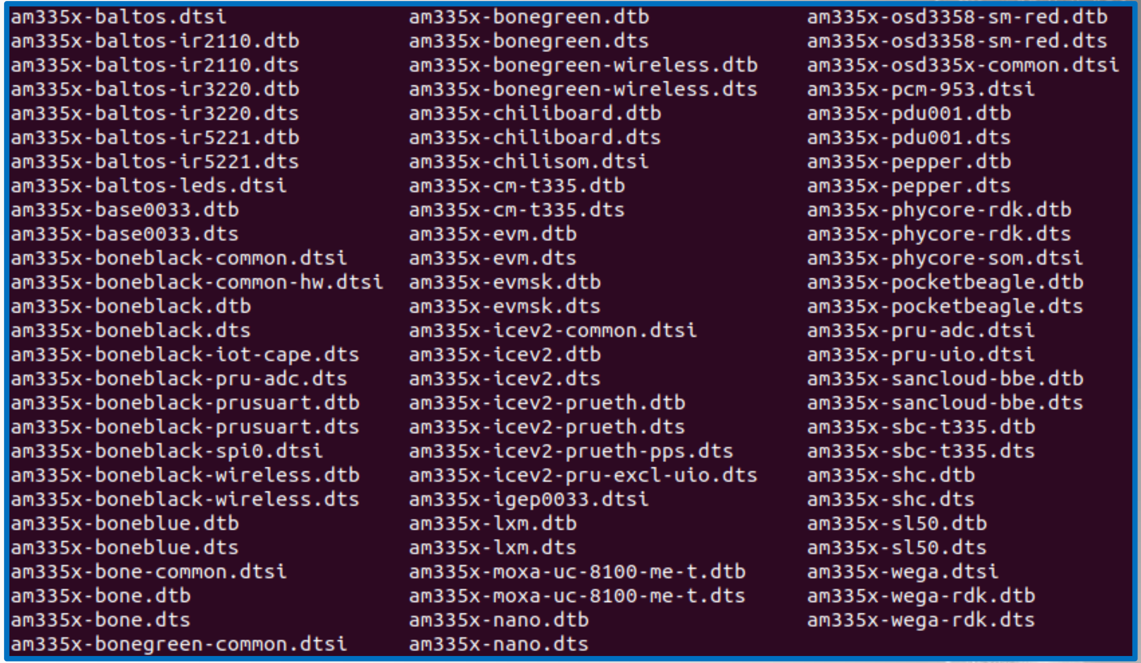
HDMI Framer

HDMI

Existing Board for basing a new design on

The arch/arm/boot/dts/ directory is where the 32bit board dts files are stored

- Leveraging a known good for the new derivative design
- Several Board Vendors shown here.
- This is just a sample of the boards in the directory.
- Going to start with the DTS file of the reference board

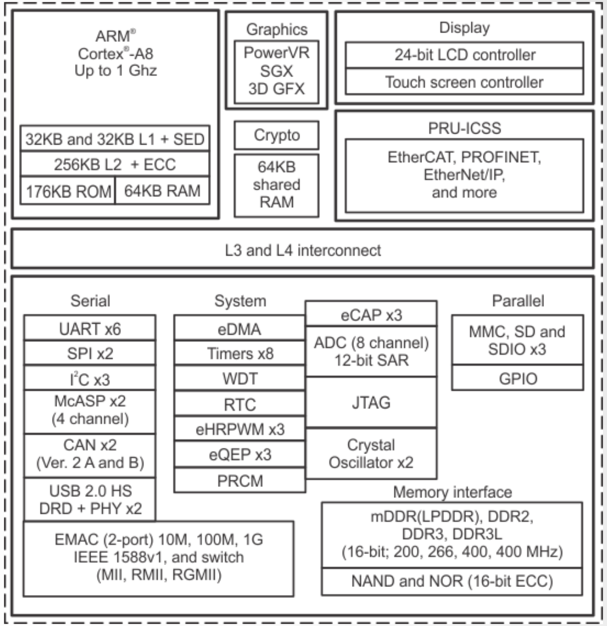


```
am335x-baltos.dtsi               am335x-bonegreen.dtb             am335x-osd3358-sm-red.dtb
am335x-baltos-ir2110.dtb         am335x-bonegreen.dts             am335x-osd3358-sm-red.dts
am335x-baltos-ir2110.dts         am335x-bonegreen-wireless.dtb    am335x-osd335x-common.dtsi
am335x-baltos-ir3220.dtb         am335x-bonegreen-wireless.dts    am335x-pcm-953.dtsi
am335x-baltos-ir3220.dts         am335x-chiliboard.dtb            am335x-pdu001.dtb
am335x-baltos-ir5221.dtb         am335x-chiliboard.dts            am335x-pdu001.dts
am335x-baltos-ir5221.dts         am335x-chilisom.dtsi             am335x-pepper.dtb
am335x-baltos-leds.dtsi          am335x-cm-t335.dtb               am335x-pepper.dts
am335x-base0033.dtb              am335x-cm-t335.dts               am335x-phycore-rdk.dtb
am335x-base0033.dts              am335x-evm.dtb                   am335x-phycore-rdk.dts
am335x-boneblack-common.dtsi     am335x-evm.dts                   am335x-phycore-som.dtsi
am335x-boneblack-common-hw.dtsi  am335x-evmsk.dtb                 am335x-pocketbeagle.dtb
am335x-boneblack.dtb             am335x-evmsk.dts                 am335x-pocketbeagle.dts
am335x-boneblack.dts             am335x-icev2-common.dtsi         am335x-pru-adc.dtsi
am335x-boneblack-iot-cape.dts    am335x-icev2.dtb                 am335x-pru-uio.dtsi
am335x-boneblack-pru-adc.dts     am335x-icev2.dts                 am335x-sancloud-bbe.dtb
am335x-boneblack-prusuart.dtb    am335x-icev2-prueth.dtb          am335x-sancloud-bbe.dts
am335x-boneblack-prusuart.dts    am335x-icev2-prueth.dts          am335x-sbc-t335.dtb
am335x-boneblack-spi0.dtsi       am335x-icev2-prueth-pps.dts      am335x-sbc-t335.dts
am335x-boneblack-wireless.dtb    am335x-icev2-pru-excl-uio.dts    am335x-shc.dtb
am335x-boneblack-wireless.dts    am335x-igep0033.dtsi             am335x-shc.dts
am335x-boneblue.dtb              am335x-lxm.dtb                   am335x-sl50.dtb
am335x-boneblue.dts              am335x-lxm.dts                   am335x-sl50.dts
am335x-bone-common.dtsi          am335x-moxa-uc-8100-me-t.dtb     am335x-wega.dtsi
am335x-bone.dtb                  am335x-moxa-uc-8100-me-t.dts     am335x-wega-rdk.dtb
am335x-bone.dts                  am335x-nano.dtb                  am335x-wega-rdk.dts
am335x-bonegreen-common.dtsi     am335x-nano.dts
```

The arch/arm64/boot/dts/ directory is where the 64bit board dts files are stored

Board dts File – How do you start?

- Will the reference DTS just drop in?
- Completely start over?
- Going to start with the DTS file of the reference board



```
/dts-v1/;
#include “processor.dtsi‟
.
.
#include <...>

/ { model = “Board Base Design X1‟;
.
.
.
}
```

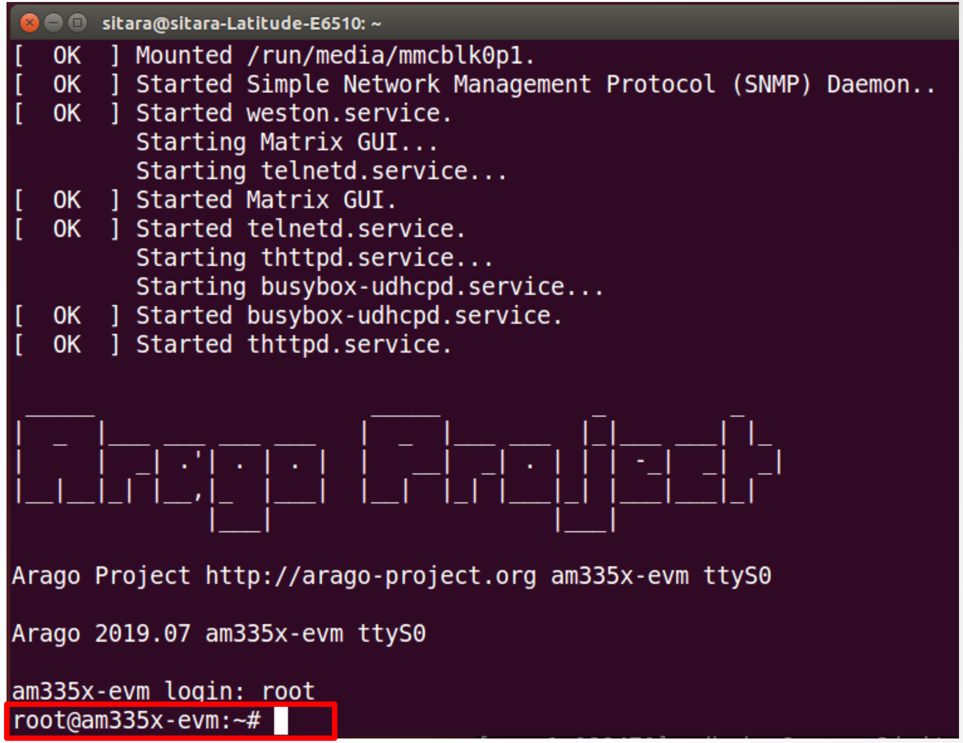
https://github.com/beagleboard/beaglebone-black/wiki/System-Reference-Manual

Reasons for hello_world dts vs. full board dts

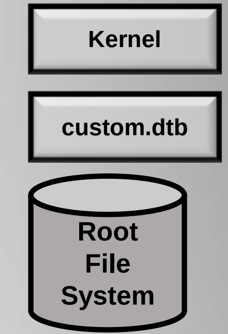
- Developing a minimal working DTS on a known platform eliminates the “is it the board or the DTS debate”
- Board bring up can be challenging so with a functioning minimal set baseline you can concentrate in small steps and will make the bring up process simpler
- By booting to a minimal interface set allows a base platform to use Linux utilities to help with debug
- Debugging a DTS that is not semantically correct is a pain.

- Upgrading kernels will be easier with a minimal baseline DTS as the processor.dtsi file most likely will have significant changes between releases. Just because the DTS file compiles does not mean the kernel will boot cleanly.

What initial success looks like



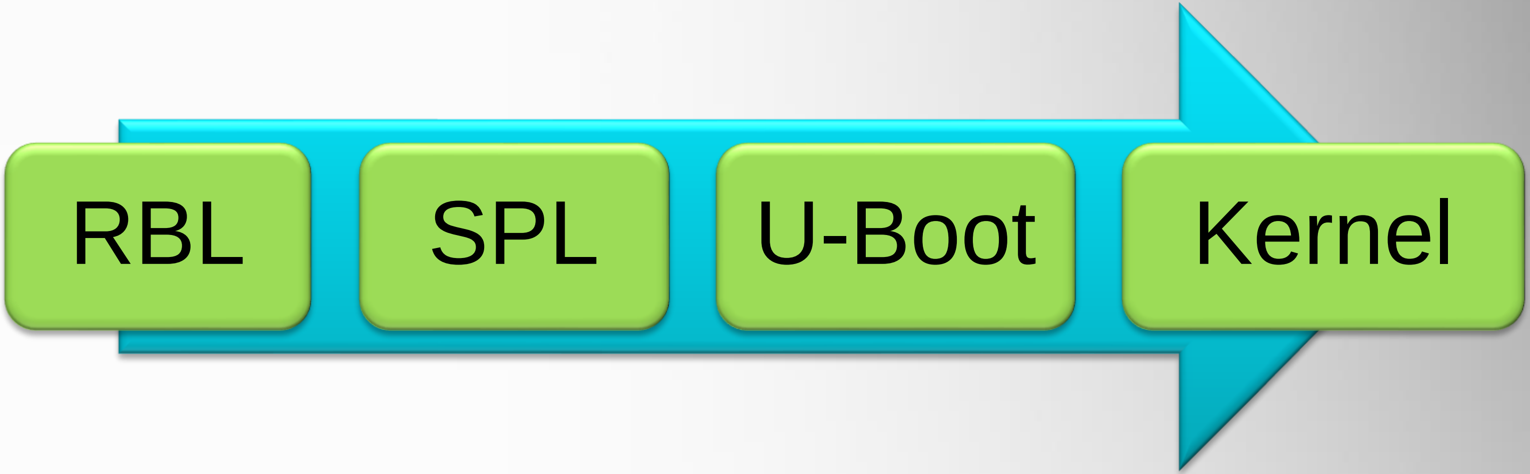
```
[  OK  ] Mounted /run/media/mmcblk0p1.
[  OK  ] Started Simple Network Management Protocol (SNMP) Daemon..
[  OK  ] Started weston.service.
         Starting Matrix GUI...
         Starting telnetd.service...
[  OK  ] Started Matrix GUI.
[  OK  ] Started telnetd.service.
         Starting thttpd.service...
         Starting busybox-udhcpd.service...
[  OK  ] Started busybox-udhcpd.service.
[  OK  ] Started thttpd.service.

 _____                    _____           _         _
|  _  |___ ___ ___ ___   |  _  |___ ___  |_|___ ___| |_
|     |  _| .'| . | . |  |   __|  _| . | | | -_|  _|  _|
|__|__|_| |__,|_  |___|  |__|  |_| |___|_| |___|___|_|
              |___|                    |___|

Arago Project http://arago-project.org am335x-evm ttyS0

Arago 2019.07 am335x-evm ttyS0

am335x-evm login: root
root@am335x-evm:~#
```



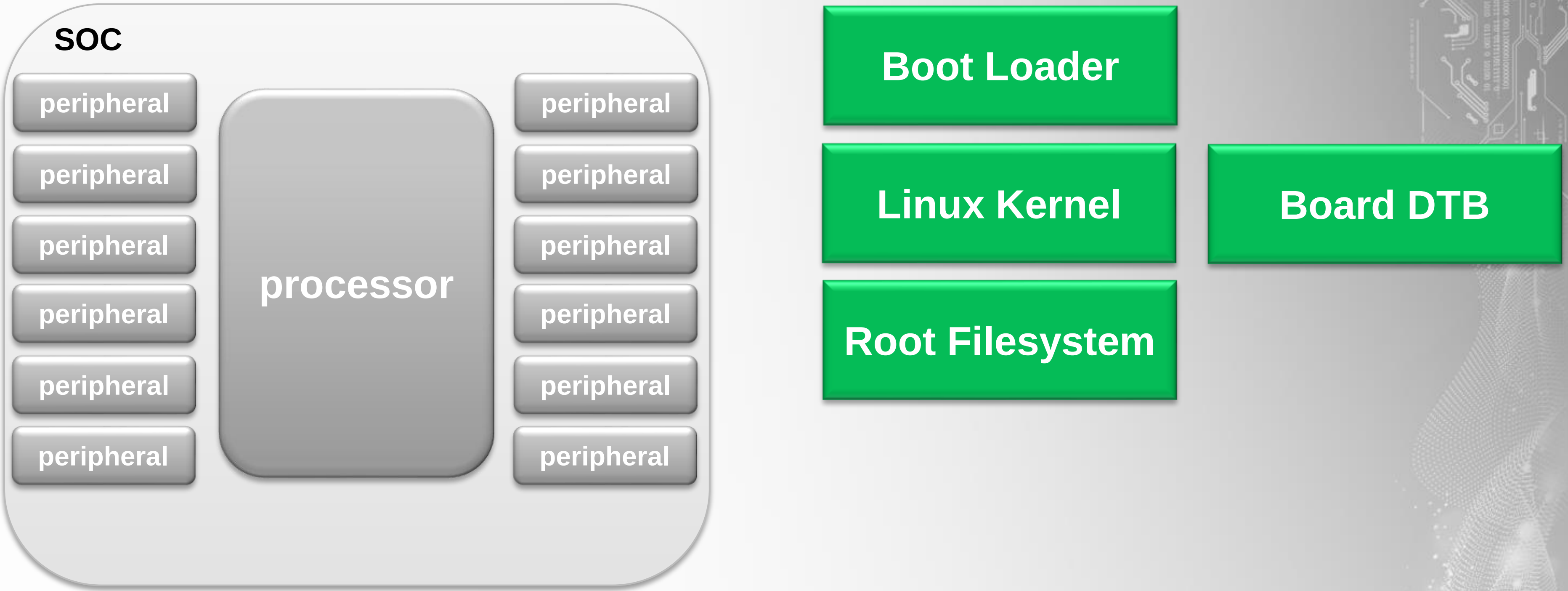
- Booting to a prompt on the console with a minimal DTS.

- Not all the peripherals have to be enabled for the developer to prove that Linux can be booted on the new board.

Quick Review, booting Linux



Elements needed for a board to boot Linux



SOC

peripheral

peripheral

peripheral

peripheral

peripheral

peripheral

processor

peripheral

peripheral

peripheral

peripheral

peripheral

peripheral

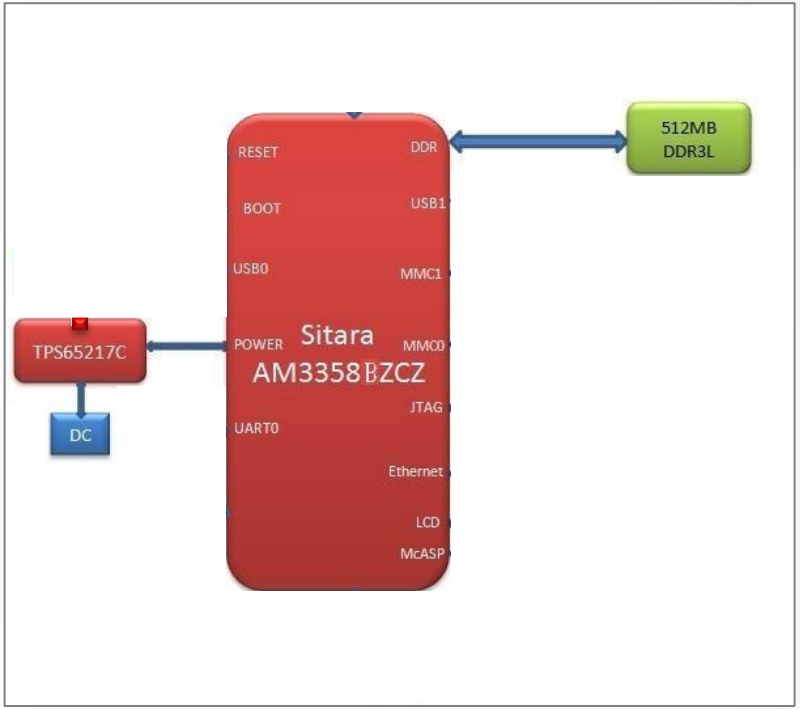
Boot Loader

Linux Kernel

Board DTB

Root Filesystem

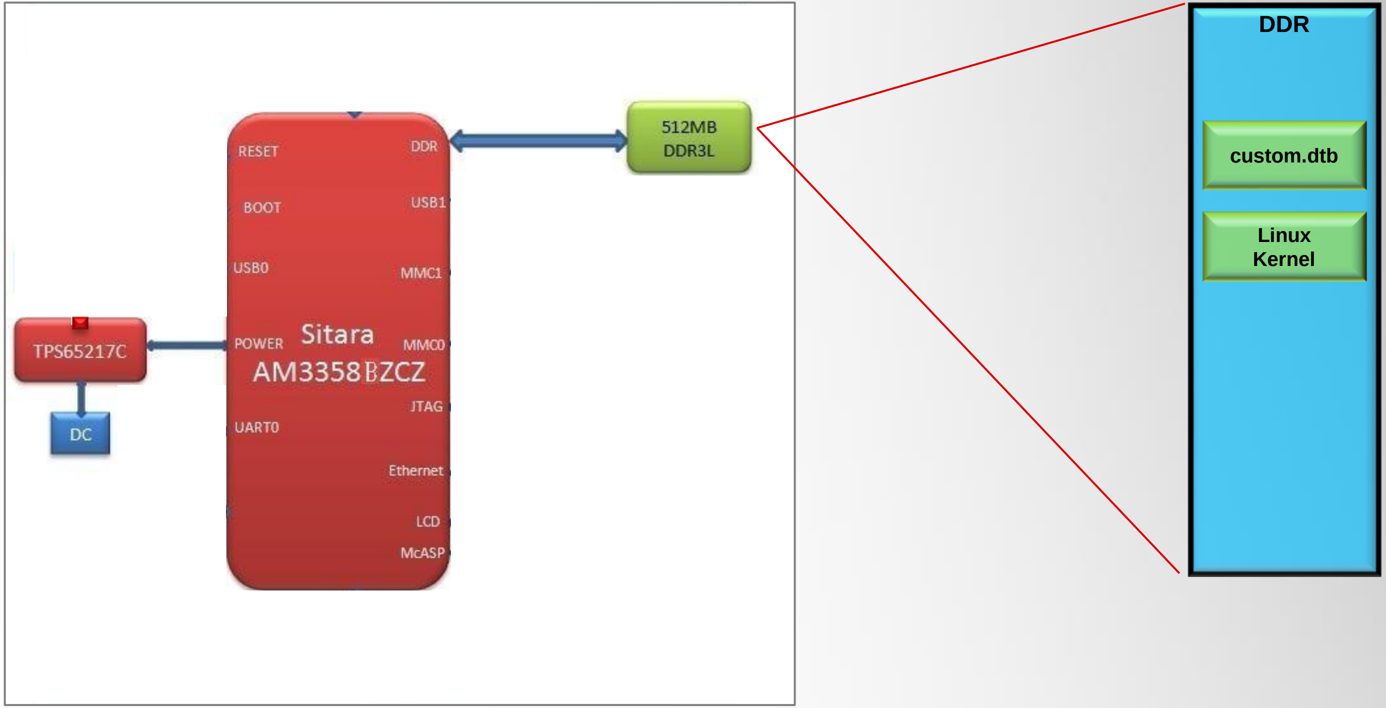
Board state as the boot loader launches Linux



https://github.com/beagleboard/beaglebone-black/wiki/System-Reference-Manual

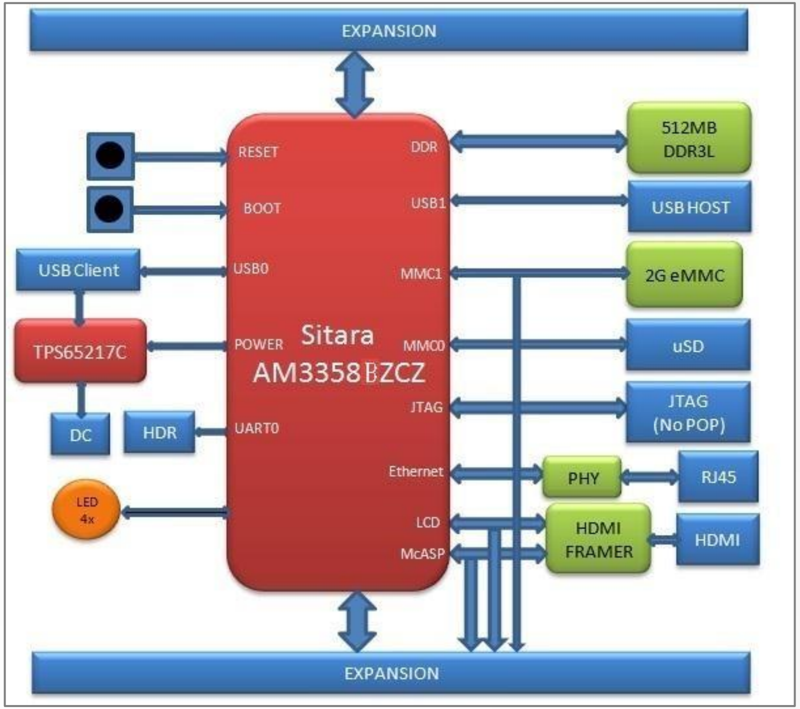
- For this discussion we are expecting that the boot loader (typically U-Boot) has been ported to the board and run and setup a minimal configuration.
- A minimal configuration is that there is power, the processor is operating at a performance point and that DDR has been configured.
- The boot loader loads the Kernel Image and Board DTB to DDR

Board state as the boot loader launches Linux



https://github.com/beagleboard/beaglebone-black/wiki/System-Reference-Manual

DTS file is what “binds” Linux



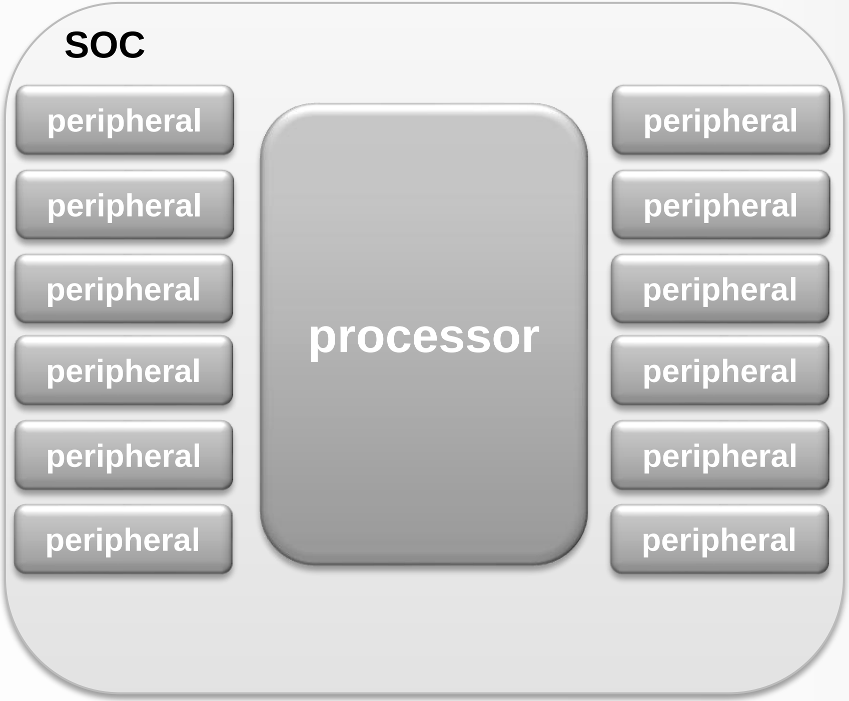
https://github.com/beagleboard/beaglebone-black/wiki/System-Reference-Manual

Board DTS

Devicetree is a data structure describing the hardware components of a particular computer so that the operating system's kernel can use and manage those components, including the CPU or CPUs, the memory, the buses and the peripherals.

https://en.wikipedia.org/wiki/Device_tree

New (Derivative) Board Based On An Existing Board



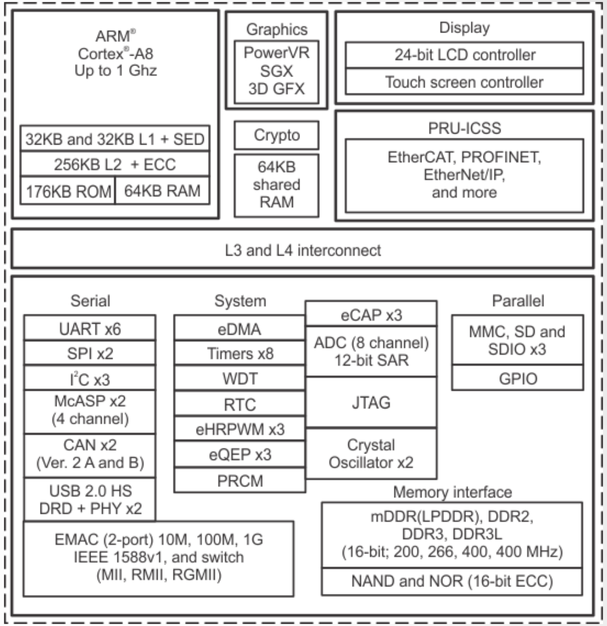
Board DTS

```
/dts-v1/;
#include “processor.dtsi”
#include “base_board_common.dtsi”
#include <...>
.
.
#include <...>

/ { model = “Board Base Design X1”;
.
.
.
}
```

Processor dtsi File

- Contains the definition for the entire SOC.

- Full entitlement of the SOC, all on chip peripherals defined here.



ARM® Cortex®-A8 Up to 1 Ghz

32KB and 32KB L1 + SED

256KB L2 + ECC

176KB ROM

64KB RAM

Graphics

PowerVR SGX 3D GFX

Crypto

64KB shared RAM

Display

24-bit LCD controller

Touch screen controller

PRU-ICSS

EtherCAT, PROFINET, EtherNet/IP, and more

L3 and L4 interconnect

Serial

UART x6

SPI x2

I²C x3

McASP x2 (4 channel)

CAN x2 (Ver. 2 A and B)

USB 2.0 HS DRD + PHY x2

System

eDMA

Timers x8

WDT

RTC

eHRPWM x3

eQEP x3

PRCM

eCAP x3

ADC (8 channel) 12-bit SAR

JTAG

Crystal Oscillator x2

Parallel

MMC, SD and SDIO x3

GPIO

EMAC (2-port) 10M, 100M, 1G IEEE 1588v1, and switch (MII, RMII, RGMII)

Memory interface

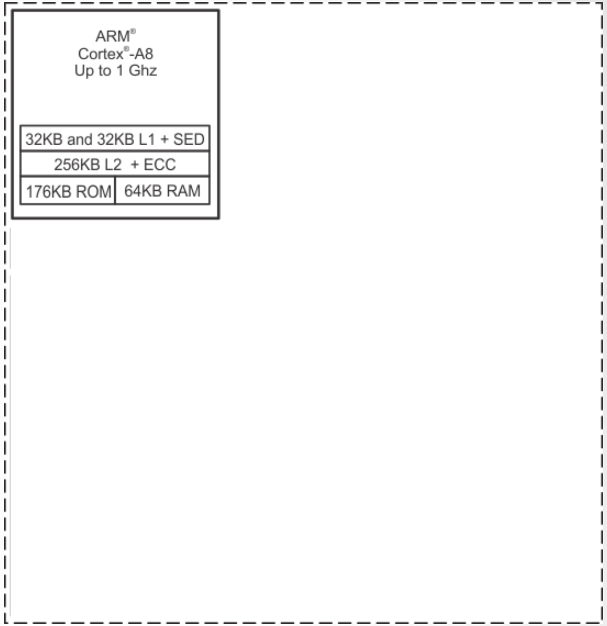
mDDR(LPDDR), DDR2, DDR3, DDR3L (16-bit; 200, 266, 400, 400 MHz)

NAND and NOR (16-bit ECC)

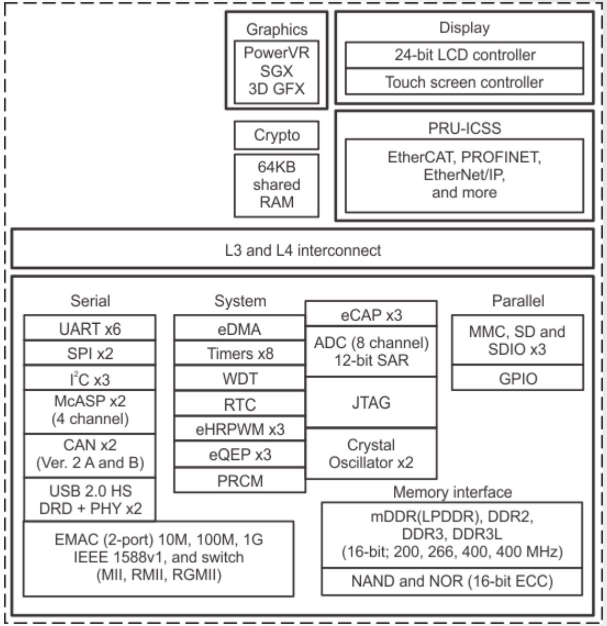
```
/dts-v1/;
#include “processor.dtsi“
.
.
#include <...>

/ { model = “Board Base Design X1“;
.
.
.
}
```

https://github.com/beagleboard/beaglebone-black/wiki/System-Reference-Manual

Processor dtsi File – Processor Architecture



Processor dtsi File – SOC internal modules



https://github.com/beagleboard/beaglebone-black/wiki/System-Reference-Manual

Processor dtsi File – Board Binding

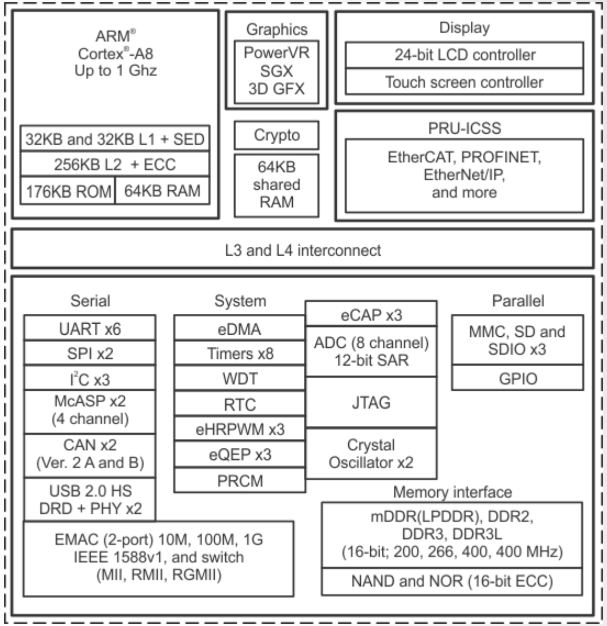
EMAC

UART (console)

I2C

USB

CAN



ARM® Cortex®-A8
Up to 1 Ghz

32KB and 32KB L1 + SED
256KB L2 + ECC
176KB ROM
64KB RAM

Graphics
PowerVR SGX 3D GFX

Crypto

64KB shared RAM

Display
24-bit LCD controller
Touch screen controller

PRU-ICSS
EtherCAT, PROFINET, EtherNet/IP, and more

L3 and L4 interconnect

Serial
UART x6
SPI x2
I²C x3
McASP x2 (4 channel)
CAN x2 (Ver. 2 A and B)
USB 2.0 HS DRD + PHY x2

System
eDMA
Timers x8
WDT
RTC
eHRPWM x3
eQEP x3
PRCM

eCAP x3
ADC (8 channel) 12-bit SAR
JTAG
Crystal Oscillator x2

Parallel
MMC, SD and SDIO x3
GPIO

EMAC (2-port) 10M, 100M, 1G
IEEE 1588v1, and switch
(MII, RMII, RGMII)

Memory interface
mDDR(LPDDR), DDR2, DDR3, DDR3L
(16-bit; 200, 266, 400, 400 MHz)
NAND and NOR (16-bit ECC)

Display

eMMC

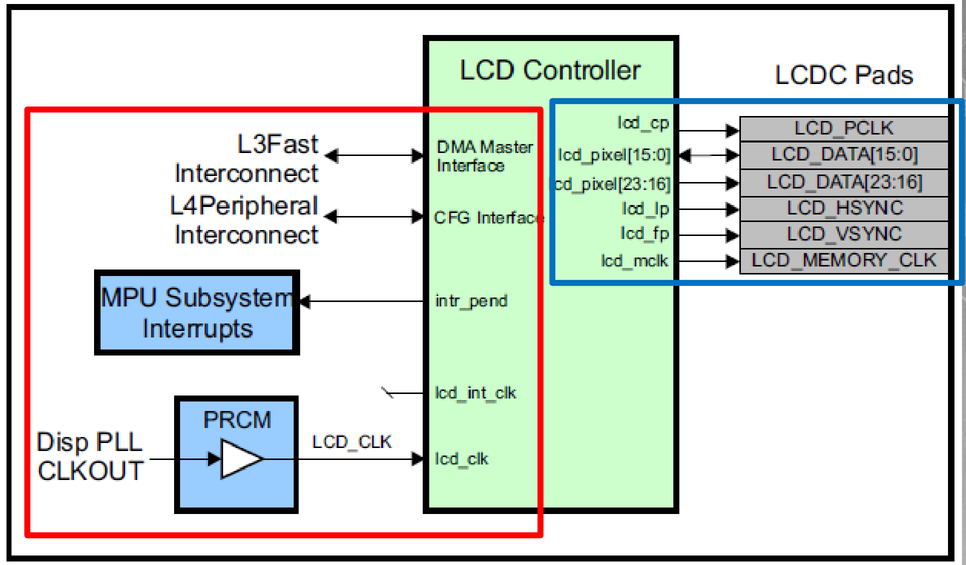
SD Card

DDR

https://github.com/beagleboard/beaglebone-black/wiki/System-Reference-Manual

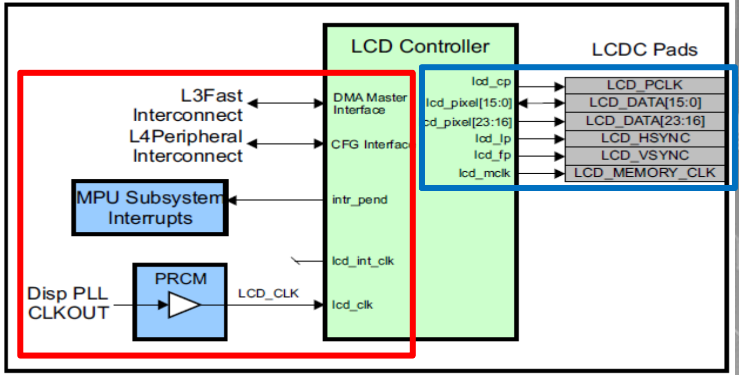
DTS File – Binding a Peripheral to a board

- What is the binding process?
- All these signals in the block diagram need to be accounted for for the driver to function.
- Is the board port developer responsible to identify “all” the settings? No



DTS File – Binding a Peripheral to a board

- Components of a DTS File



Arch

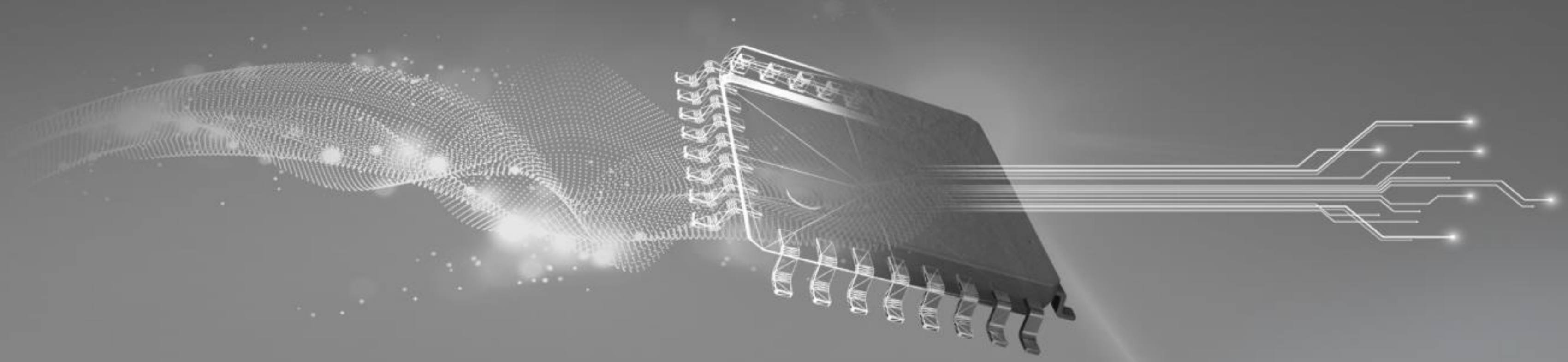
SOC

am33xx.dtsi

dt-bindings/gpio/gpio.h
dt-bindings/pinctrl/am33xx.h
skeleton.dtsi
am33xx-clocks.dtsi

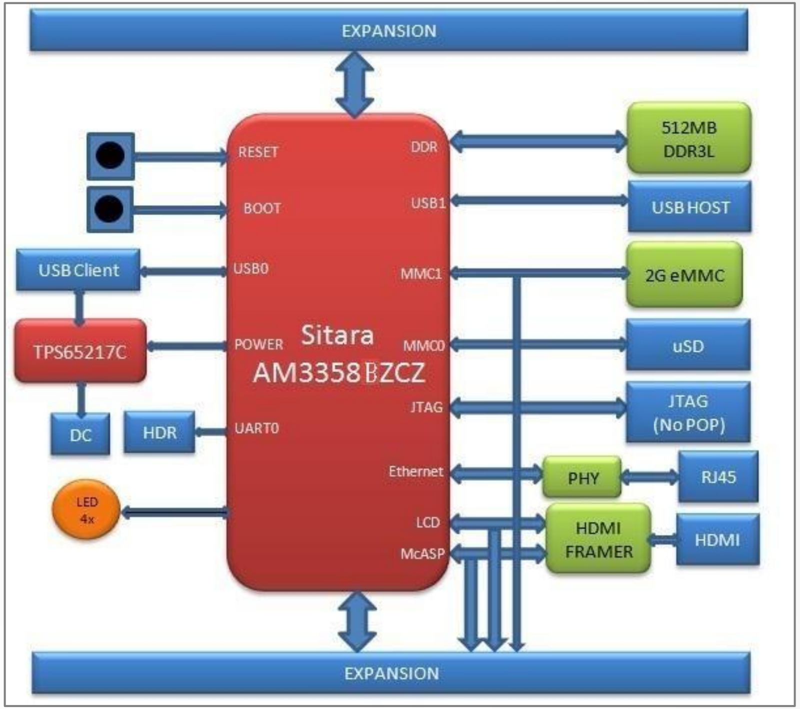
Board

am335x-boneblack.dts
am335x-bone-common.dtsi



Hello World DTS File

DTS file “binds” Linux to a board



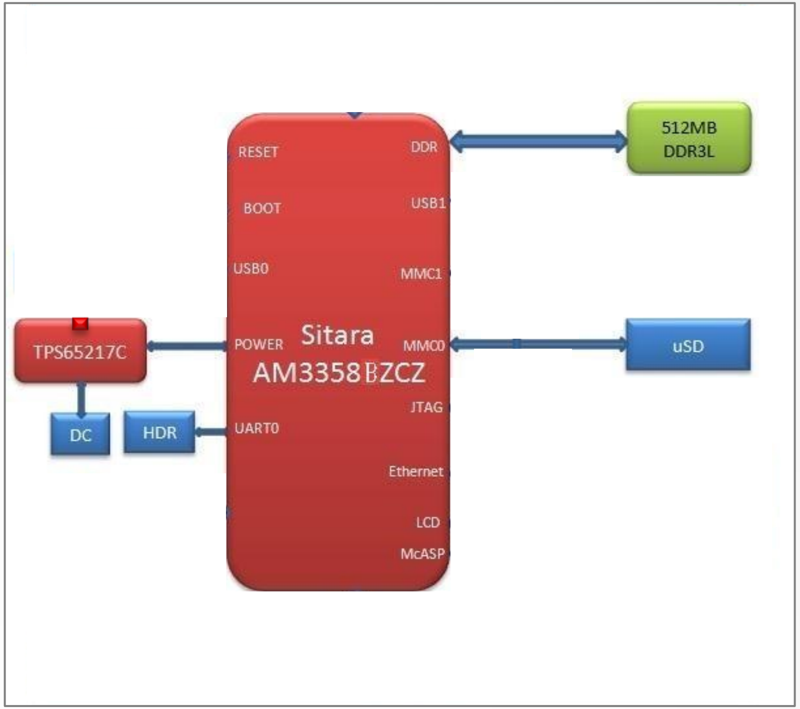
https://github.com/beagleboard/beaglebone-black/wiki/System-Reference-Manual

Board DTS

Devicetree is a data structure describing the hardware components of a particular computer so that the operating system's kernel can use and manage those components, including the CPU or CPUs, the memory, the buses and the peripherals.

https://en.wikipedia.org/wiki/Device_tree

The Hello World DTS File



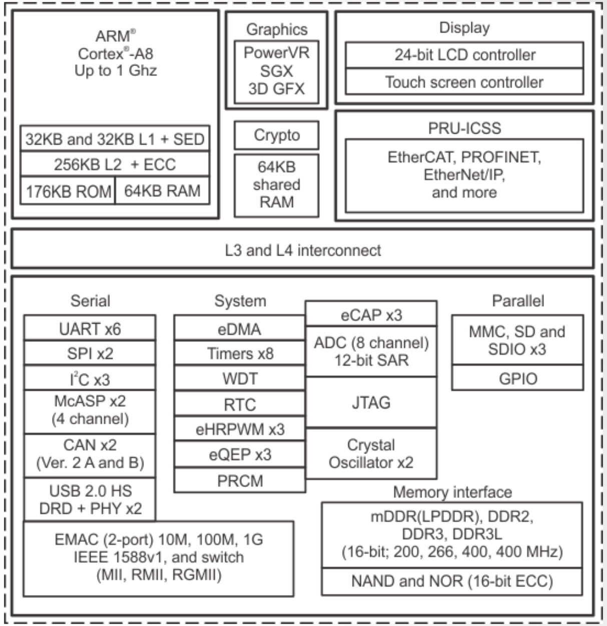
```
main( ) {
     printf("hello, world\n");
}
```

```
/dts-v1/;
#include “processor.dtsi“

/ {
 model = “Hello World“;

};

&uart
&mmc
...
```

https://github.com/beagleboard/beaglebone-black/wiki/System-Reference-Manual

Processor dtsi File – Board Binding

UART (console)



SD Card

DDR

https://github.com/beagleboard/beaglebone-black/wiki/System-Reference-Manual

The Hello World DTS File

existing_reference_board.dts

```
/dts-v1/;
#include “processor.dtsi‟
#include “board-type-common.dtsi‟
#include “base-board-common.dtsi‟
#include “another-common.dtsi‟

/ {
 model = “Hello World‟;

};

&uart
&mmc
&emac
&usb
...
```

hello_world.dts

```
/dts-v1/;
#include “processor.dtsi‟

/ {
 model = “Hello World‟;

};

&uart
&mmc
...
```

Going to draw on reference DTS to make the hello_world.dts

Building the DTS file to a DTB file (blob)

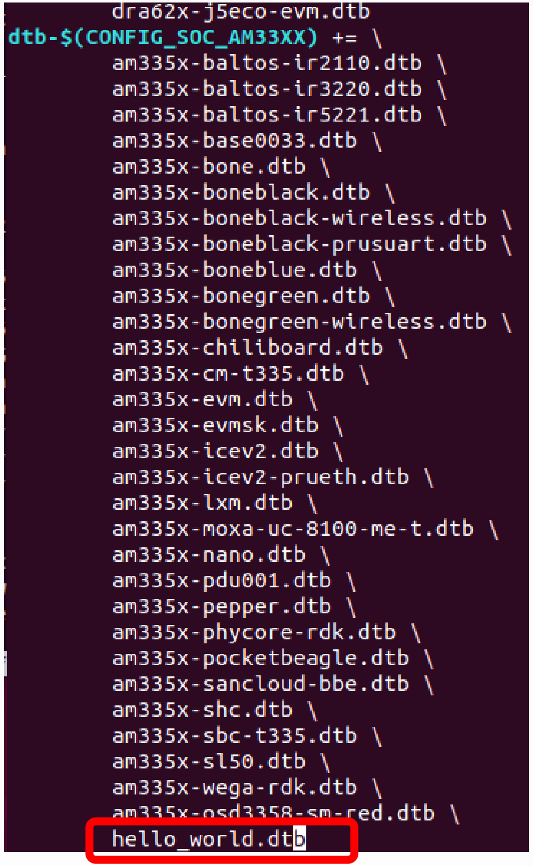
- Using the dtbs build target as part of the kernel make system

```
make  ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- dtbs

CALL    scripts/checksyscalls.sh
DTC     arch/arm/boot/dts/hello_world.dtb
```

- Debug tip, how to “un-compile” the DTS file. The reason is that just because it compiles does not mean it will work. At a minimum this step could reveal incorrect nesting, over writtern values.

```
dtc hello_world.dtb  -O dts -o hello_world_as_compiled.dts
```

Adding new file to the DTB build



```
        dra62x-j5eco-evm.dtb
dtb-$(CONFIG_SOC_AM33XX) += \
        am335x-baltos-ir2110.dtb \
        am335x-baltos-ir3220.dtb \
        am335x-baltos-ir5221.dtb \
        am335x-base0033.dtb \
        am335x-bone.dtb \
        am335x-boneblack.dtb \
        am335x-boneblack-wireless.dtb \
        am335x-boneblack-prusuart.dtb \
        am335x-boneblue.dtb \
        am335x-bonegreen.dtb \
        am335x-bonegreen-wireless.dtb \
        am335x-chiliboard.dtb \
        am335x-cm-t335.dtb \
        am335x-evm.dtb \
        am335x-evmsk.dtb \
        am335x-icev2.dtb \
        am335x-icev2-prueth.dtb \
        am335x-lxm.dtb \
        am335x-moxa-uc-8100-me-t.dtb \
        am335x-nano.dtb \
        am335x-pdu001.dtb \
        am335x-pepper.dtb \
        am335x-phycore-rdk.dtb \
        am335x-pocketbeagle.dtb \
        am335x-sancloud-bbe.dtb \
        am335x-shc.dtb \
        am335x-sbc-t335.dtb \
        am335x-sl50.dtb \
        am335x-wega-rdk.dtb \
        am335x-osd3358-sm-red.dtb \
        hello_world.dtb
```

- Add your file to the Makefile in the config area of the board type you based your board on.

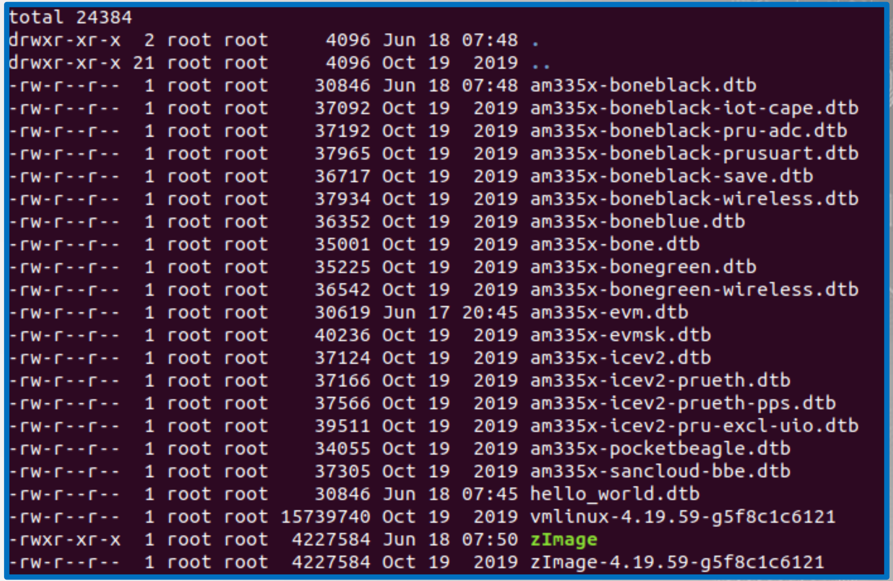
The directory for the Makefile is located here:
(32bit systems) arch/arm/boot/dts
(64bit systems) arch/arm64/boot/dts

Where is the DTB file stored?

- The **/boot** directory in the root filesystem for the board holds the DTB for the board.

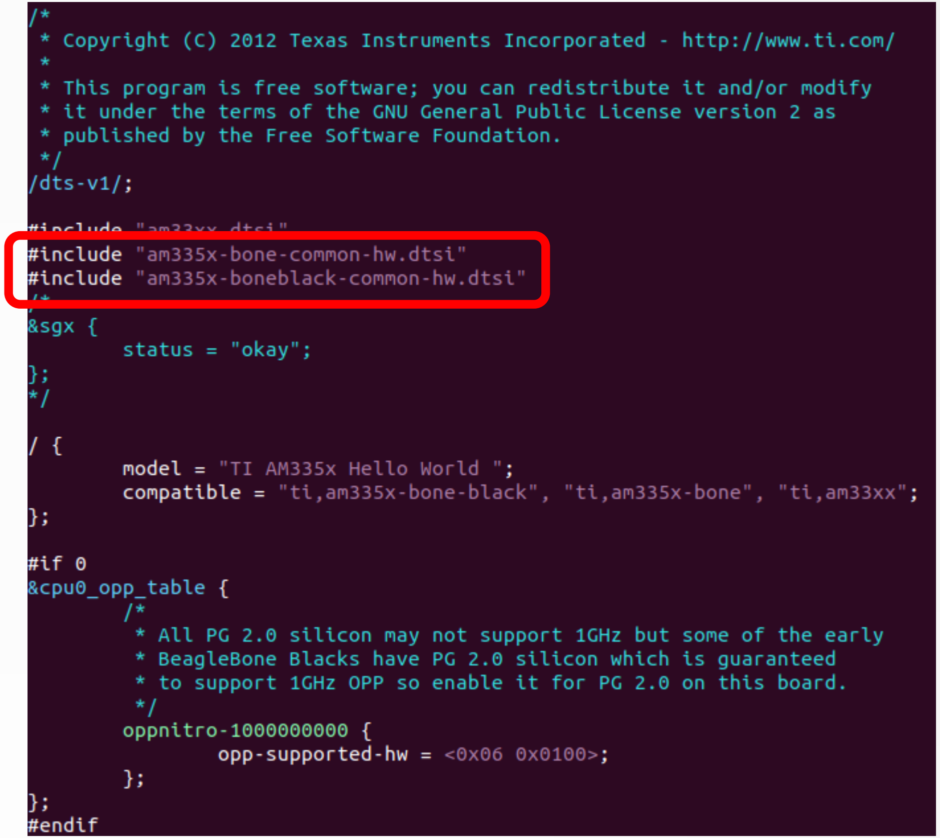
- For this class of SOC U-Boot uses an eeprom to identify the board so the DTB associated with the board can be loaded with the kernel at boot time

- For this test the hello_world.dtb was copied over the am335x_boneblack.dtb to make the development of the new DTB file.



```
total 24384
drwxr-xr-x  2 root root     4096 Jun 18 07:48 .
drwxr-xr-x 21 root root     4096 Oct 19  2019 ..
-rw-r--r--  1 root root    30846 Jun 18 07:48 am335x-boneblack.dtb
-rw-r--r--  1 root root    37092 Oct 19  2019 am335x-boneblack-iot-cape.dtb
-rw-r--r--  1 root root    37192 Oct 19  2019 am335x-boneblack-pru-adc.dtb
-rw-r--r--  1 root root    37965 Oct 19  2019 am335x-boneblack-prusuart.dtb
-rw-r--r--  1 root root    36717 Oct 19  2019 am335x-boneblack-save.dtb
-rw-r--r--  1 root root    37934 Oct 19  2019 am335x-boneblack-wireless.dtb
-rw-r--r--  1 root root    36352 Oct 19  2019 am335x-boneblue.dtb
-rw-r--r--  1 root root    35001 Oct 19  2019 am335x-bone.dtb
-rw-r--r--  1 root root    35225 Oct 19  2019 am335x-bonegreen.dtb
-rw-r--r--  1 root root    36542 Oct 19  2019 am335x-bonegreen-wireless.dtb
-rw-r--r--  1 root root    30619 Jun 17 20:45 am335x-evm.dtb
-rw-r--r--  1 root root    40236 Oct 19  2019 am335x-evmsk.dtb
-rw-r--r--  1 root root    37124 Oct 19  2019 am335x-icev2.dtb
-rw-r--r--  1 root root    37166 Oct 19  2019 am335x-icev2-prueth.dtb
-rw-r--r--  1 root root    37566 Oct 19  2019 am335x-icev2-prueth-pps.dtb
-rw-r--r--  1 root root    39511 Oct 19  2019 am335x-icev2-pru-excl-uio.dtb
-rw-r--r--  1 root root    34055 Oct 19  2019 am335x-pocketbeagle.dtb
-rw-r--r--  1 root root    37305 Oct 19  2019 am335x-sancloud-bbe.dtb
-rw-r--r--  1 root root    30846 Jun 18 07:45 hello_world.dtb
-rw-r--r--  1 root root 15739740 Oct 19  2019 vmlinux-4.19.59-g5f8c1c6121
-rwxr-xr-x  1 root root  4227584 Jun 18 07:50 zImage
-rw-r--r--  1 root root  4227584 Oct 19  2019 zImage-4.19.59-g5f8c1c6121
```

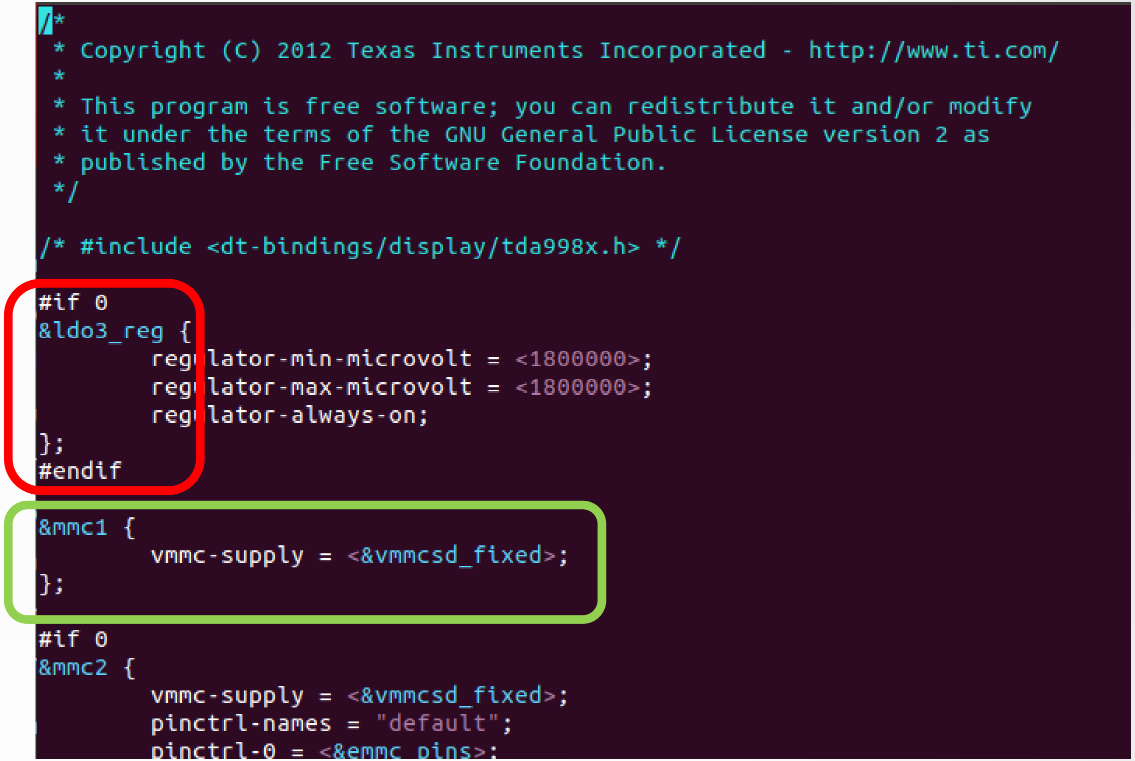
How to make an Hello World DTS



```
/*
 * Copyright (C) 2012 Texas Instruments Incorporated - http://www.ti.com/
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 2 as
 * published by the Free Software Foundation.
 */
/dts-v1/;

#include "am33xx.dtsi"
#include "am335x-bone-common-hw.dtsi"
#include "am335x-boneblack-common-hw.dtsi"
/*
&sgx {
        status = "okay";
};
*/

/ {
        model = "TI AM335x Hello World ";
        compatible = "ti,am335x-bone-black", "ti,am335x-bone", "ti,am33xx";
};

#if 0
&cpu0_opp_table {
        /*
         * All PG 2.0 silicon may not support 1GHz but some of the early
         * BeagleBone Blacks have PG 2.0 silicon which is guaranteed
         * to support 1GHz OPP so enable it for PG 2.0 on this board.
         */
        oppnitro-1000000000 {
                opp-supported-hw = <0x06 0x0100>;
        };
};
#endif
```

- Ideal case is to develop the hello_world.dts on the original base board so you are working from a known good.
- Iteratively remove elements until the boot fails…
- Example here used renamed copies of include files.

How to make an Hello World DTS



```
/*
 * Copyright (C) 2012 Texas Instruments Incorporated - http://www.ti.com/
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 2 as
 * published by the Free Software Foundation.
 */

/* #include <dt-bindings/display/tda998x.h> */

#if 0
&ldo3_reg {
        regulator-min-microvolt = <1800000>;
        regulator-max-microvolt = <1800000>;
        regulator-always-on;
};
#endif

&mmc1 {
        vmmc-supply = <&vmmcsd_fixed>;
};

#if 0
&mmc2 {
        vmmc-supply = <&vmmcsd_fixed>;
        pinctrl-names = "default";
        pinctrl-0 = <&emmc_pins>;
```

- Red box shows isolated node that was not needed to get to a basic boot.
- Green shows what was still needed after hello world DTS was created.
- Process was iterative

HW DTS File

- “Hello World” like minimal board DTS File
- Define the model, memory
- Voltage regulator for SD card for root filesytem
- UART node and supporting pin mux

```
/dts-v1/;
#include "am33xx.dtsi‟
/{
    model = "TI AM335x Hello World ";
    compatible = "ti,am335x-bone-black", "ti,am335x-bone", "ti,am33xx";

    memory@80000000 {
        device_type = "memory";
        reg = <0x80000000 0x10000000>; /* 256 MB */
    };
    chosen {
         stdout-path = &uart0;
    };
    vmmcsd_fixed: fixedregulator0 {
         compatible = "regulator-fixed";
         regulator-name = "vmmcsd_fixed";
         regulator-min-microvolt = <3300000>;
         regulator-max-microvolt = <3300000>;
    };
};

&am33xx_pinmux {
     uart0_pins: pinmux_uart0_pins {
          pinctrl-single,pins = <
              AM33XX_IOPAD(0x970, PIN_INPUT_PULLUP | MUX_MODE0) /*uart0_rxd.uart0_rxd*/
              AM33XX_IOPAD(0x974, PIN_OUTPUT_PULLDOWN | MUX_MODE0) /*uart0_txd.uart0_txd*/
          >;
```

HW DTS File

- mmc1 Pin Mux
- UART node
- mmc1 node

```
 mmc1_pins: pinmux_mmc1_pins {
     pinctrl-single,pins = <
         AM33XX_IOPAD(0x960, PIN_INPUT | MUX_MODE7)         /* spio0_cs1.gpio0_6 */
         AM33XX_IOPAD(0x8fc, PIN_INPUT_PULLUP | MUX_MODE0) /* mmc0_dat0.mmc0_dat0 */
         AM33XX_IOPAD(0x8f8, PIN_INPUT_PULLUP | MUX_MODE0) /* mmc0_dat1.mmc0_dat1 */
         AM33XX_IOPAD(0x8f4, PIN_INPUT_PULLUP | MUX_MODE0) /* mmc0_dat2.mmc0_dat2 */
         AM33XX_IOPAD(0x8f0, PIN_INPUT_PULLUP | MUX_MODE0) /* mmc0_dat3.mmc0_dat3 */
         AM33XX_IOPAD(0x904, PIN_INPUT_PULLUP | MUX_MODE0) /* mmc0_cmd.mmc0_cmd */
         AM33XX_IOPAD(0x900, PIN_INPUT_PULLUP | MUX_MODE0) /* mmc0_clk.mmc0_clk */
      >;
      };
};

&uart0 {
      pinctrl-names = "default";
      pinctrl-0 = <&uart0_pins>;
      status = "okay";
};

&mmc1 {
      status = "okay";
      bus-width = <0x4>;
      pinctrl-names = "default";
      pinctrl-0 = <&mmc1_pins>;
      cd-gpios = <&gpio0 6 GPIO_ACTIVE_LOW>;
      vmmc-supply = <&vmmcsd_fixed>;
};
```

HW DTS File

- Blank Slate, let’s build a “Hello World” like minimal board DTS File, just want to get to a prompt on the console.

```
/dts-v1/;
```

HW DTS File

- “Hello World” like minimal board DTS File
- The first is the defining the Arch/SOC for the custom board. What is the processor the board is based on

```
/dts-v1/;

#include "am33xx.dtsi"
```

Arch/SOC Abstraction

HW DTS File

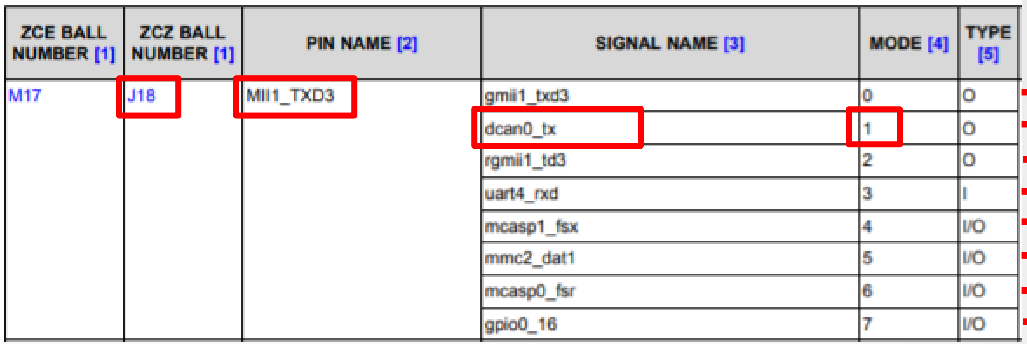
- “Hello World” like minimal board DTS File

- Defining the root node model, compatibility, memory….

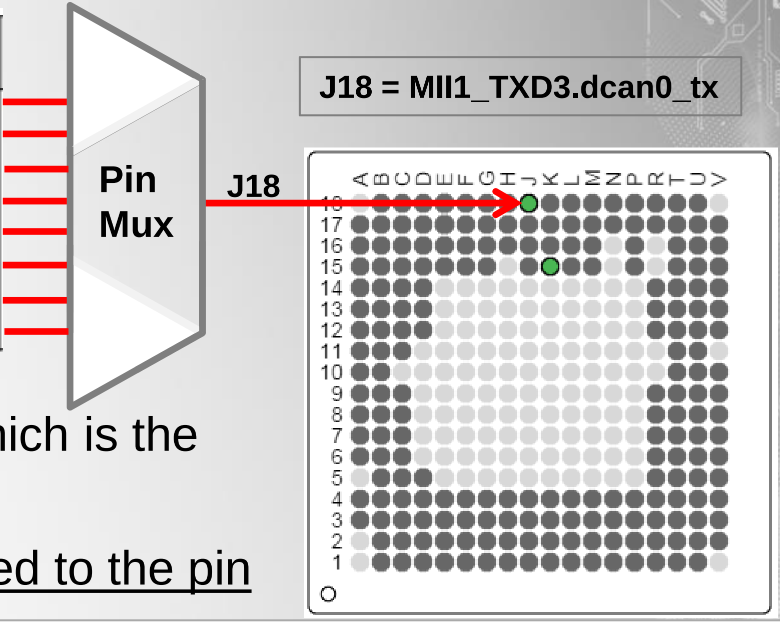
```
/dts-v1/;

#include "am33xx.dtsi"

/ {
     model = "TI AM335x Hello World ";
    compatible = "ti,am335x-bone-black", "ti,am335x-bone", "ti,am33xx";

    memory@80000000 {
        device_type = "memory";
        reg = <0x80000000 0x10000000>; /* 256 MB */
    };
    chosen {
         stdout-path = &uart0;
    };
    vmmcsd_fixed: fixedregulator0 {
         compatible = "regulator-fixed";
         regulator-name = "vmmcsd_fixed";
         regulator-min-microvolt = <3300000>;
         regulator-max-microvolt = <3300000>;
    };
};
```

For SOC that use pin muxes - what is a pin mux?

- Most pins on the SOC have a mux must be set to enable a peripheral access
- Each pin name has several signal names that can accessed by a mux mode



ZCE BALL NUMBER [1]	ZCZ BALL NUMBER [1]	PIN NAME [2]	SIGNAL NAME [3]	MODE [4]	TYPE [5]
M17	J18	MII1_TXD3	gmii1_txd3	0	O
			dcan0_tx	1	O
			rgmii1_td3	2	O
			uart4_rxd	3	I
			mcasp1_fsx	4	I/O
			mmc2_dat1	5	I/O
			mcasp0_fsr	6	I/O
			gpio0_16	7	I/O



J18 = MII1_TXD3.dcan0_tx

- In this example J18 is using mux mode 1 which is the DCAN0 TX signal
- The silicon IP in the DTS has to be connected to the pin

HW DTS File

- “Hello World” like minimal board DTS File

- Defining the pin mux for a peripheral, in this case UART

- <u>On SOCs that use a pin mux for signal routing this is a critical step.</u>

```
&am33xx_pinmux {
   uart0_pins: pinmux_uart0_pins {
          pinctrl-single,pins = <
              AM33XX_IOPAD(0x970, PIN_INPUT_PULLUP | MUX_MODE0) /*uart0_rxd.uart0_rxd*/
              AM33XX_IOPAD(0x974, PIN_OUTPUT_PULLDOWN | MUX_MODE0) /*uart0_txd.uart0_txd*/
          >;
     };

   mmc1_pins: pinmux_mmc1_pins {
     pinctrl-single,pins = <
         AM33XX_IOPAD(0x960, PIN_INPUT | MUX_MODE7)        /* spio0_cs1.gpio0_6 */
         AM33XX_IOPAD(0x8fc, PIN_INPUT_PULLUP | MUX_MODE0) /* mmc0_dat0.mmc0_dat0 */
         AM33XX_IOPAD(0x8f8, PIN_INPUT_PULLUP | MUX_MODE0) /* mmc0_dat1.mmc0_dat1 */
         AM33XX_IOPAD(0x8f4, PIN_INPUT_PULLUP | MUX_MODE0) /* mmc0_dat2.mmc0_dat2 */
         AM33XX_IOPAD(0x8f0, PIN_INPUT_PULLUP | MUX_MODE0) /* mmc0_dat3.mmc0_dat3 */
         AM33XX_IOPAD(0x904, PIN_INPUT_PULLUP | MUX_MODE0) /* mmc0_cmd.mmc0_cmd */
         AM33XX_IOPAD(0x900, PIN_INPUT_PULLUP | MUX_MODE0) /* mmc0_clk.mmc0_clk */
      >;
      };
};
```

HW DTS File

- “Hello World” like minimal board DTS File

- Finally enable the nodes with information specific to the new board

```
&uart0 {
        pinctrl-names = "default";
        pinctrl-0 = <&uart0_pins>;
        status = "okay";
};

&mmc1 {
        status = "okay";
        bus-width = <0x4>;
        pinctrl-names = "default";
        pinctrl-0 = <&mmc1_pins>;
        cd-gpios = <&gpio0 6 GPIO_ACTIVE_LOW>;
        vmmc-supply = <&vmmcsd_fixed>;
};
```

- Notice “&” for uart0, this means information is being appended to the uart0 node that is defined in the processor dtsi. Recommend to never modify the processor dtsi to add board binding information like what is shown here

HW DTS File – What binding was needed

- UART node in the DTS File
- UART node disassembled from the DTB file, significant difference. DTS only needs to define the pinmux and set status to “okay” in this example.
- Remember the debug tip from earlier about reverse compiling the DTB file.

```
&uart0 {
        pinctrl-names = "default";
        pinctrl-0 = <&uart0_pins>;
        status = "okay";
};
```

```
serial@44e09000 {
      compatible = "ti,am3352-uart", "ti,omap3-uart";
      ti,hwmods = "uart1";
      clock-frequency = <0x2dc6c00>;
      reg = <0x44e09000 0x2000>;
      interrupts = <0x48>;
      status = "okay";
      dmas = <0x26 0x1a 0x0 0x26 0x1b 0x0>;
      dma-names = "tx", "rx";
      pinctrl-names = "default";
      pinctrl-0 = <0x2a>;
};
```

HW DTS File – What binding was needed

- UART node in the DTS File
- UART node disassembled from the DTB file, significant difference. DTS only needs to define the pinmux and set status to “okay” in this example.
- Remember the debug tip from earlier about reverse compiling the DTB file.

```
&uart0 {
        pinctrl-names = "default";
        pinctrl-0 = <&uart0_pins>;
        status = "okay";
};
```

```
serial@44e09000 {
        compatible = "ti,am3352-uart", "ti,omap3-uart";
        ti,hwmods = "uart1";
        clock-frequency = <0x2dc6c00>;
        reg = <0x44e09000 0x2000>;
        interrupts = <0x48>;
        status = "okay";
        dmas = <0x26 0x1a 0x0 0x26 0x1b 0x0>;
        dma-names = "tx", "rx";
        pinctrl-names = "default";
        pinctrl-0 = <0x2a>;
};
```

Uart Binding Doc

Documentation/devicetree/bindings/serial/omap_serial.txt

```
OMAP UART controller
Required properties:
- compatible : should be "ti,j721e-uart", "ti,am654-uart"
               for J721E controllers
- compatible : should be "ti,am654-uart" for AM654 controllers
- compatible : should be "ti,omap2-uart" for OMAP2 controllers
- compatible : should be "ti,omap3-uart" for OMAP3 controllers
- compatible : should be "ti,omap4-uart" for OMAP4 controllers
- compatible : should be "ti,am4372-uart" for AM437x controllers
- compatible : should be "ti,am3352-uart" for AM335x controllers
- compatible : should be "ti,dra742-uart" for DRA7x controllers
- reg : address and length of the register space
- interrupts or interrupts-extended :
                          Should contain the uart interrupt
                          specifier or both the interrupt
                          controller phandle and interrupt
                          specifier.
- ti,hwmods : Must be "uart<n>", n being the
                instance number (1-based)
```

```
Optional properties:
- clock-frequency : frequency of the clock input to the UART
- dmas : DMA specifier, consisting of a phandle to the DMA
        controller node and a DMA channel
        number.
- dma-names : "rx" for receive channel,
              "tx" for transmit channel.
- rs485-rts-delay, rs485-rx-during-tx, linux,
- rs485-enabled-at-boot-time: see rs485.txt
- rs485-rts-active-high: drive RTS high when sending
                     (default is low).
- clocks: phandle to the functional clock as per
  Documentation/devicetree/bindings/clock/clock-bindings.txt

Example:
uart4: serial@49042000 {
             compatible = "ti,omap3-uart";
             reg = <0x49042000 0x400>;
             interrupts = <80>;
             dmas = <&sdma 81 &sdma 82>;
             dma-names = "tx", "rx";
             ti,hwmods = "uart4";
             clock-frequency = <48000000>;
         };
```

Uart Binding Doc

Documentation/devicetree/bindings/serial/omap_serial.txt

```
OMAP UART controller
Required properties:
- compatible : should be "ti,j721e-uart", "ti,am654-uart"
               for J721E controllers
- compatible : should be "ti,am654-uart" for AM654 controllers
- compatible : should be "ti,omap2-uart" for OMAP2 controllers
- compatible : should be "ti,omap3-uart" for OMAP3 controllers
- compatible : should be "ti,omap4-uart" for OMAP4 controllers
- compatible : should be "ti,am4372-uart" for AM437x controllers
- compatible : should be "ti,am3352-uart" for AM335x controllers
- compatible : should be "ti,dra742-uart" for DRA7x controllers
- reg : address and length of the register space
- interrupts or interrupts-extended :
                          Should contain the uart interrupt
                          specifier or both the interrupt
                          controller phandle and interrupt
                          specifier.
- ti,hwmods : Must be "uart<n>", n being the
                instance number (1-based)
```

```
Optional properties:
- clock-frequency : frequency of the clock input to the UART
- dmas : DMA specifier, consisting of a phandle to the DMA
         controller node and a DMA channel
         number.
- dma-names : "rx" for receive channel,
              "tx" for transmit channel.
- rs485-rts-delay, rs485-rx-during-tx, linux,
- rs485-enabled-at-boot-time: see rs485.txt
- rs485-rts-active-high: drive RTS high when sending
                         (default is low).
- clocks: phandle to the functional clock as per
  Documentation/devicetree/bindings/clock/clock-bindings.txt

Example:
uart4: serial@49042000 {
             compatible = "ti,omap3-uart";
             reg = <0x49042000 0x400>;
             interrupts = <80>;
             dmas = <&sdma 81 &sdma 82>;
             dma-names = "tx", "rx";
             ti,hwmods = "uart4";
             clock-frequency = <48000000>;
     };
```

Did the hello world work?

```
Starting kernel ...

[    0.000000] Booting Linux on physical CPU 0x0
[    0.000000] Linux version 4.19.59-g5f8c1c6121 (oe-user@oe-host) (gcc version 8.3.0 (GNU
Toolchain for the A-profile Architecture 8.3-2019.03 (arm-rel-8.36))) #1 PREEMPT Sat Oct 19
17:17:25 UTC 2019
[    0.000000] CPU: ARMv7 Processor [413fc082] revision 2 (ARMv7), cr=10c5387d

[    0.000000] OF: fdt: Machine model: TI AM335x Hello World

[    0.000000] Built 1 zonelists, mobility grouping on.  Total pages: 129920
[    0.000000] Kernel command line: console=ttyO0,115200n8 root=PARTUUID=0000000-02 rw
rootfstype=ext4 rootwait
```

Did the hello world work?

```
Starting kernel ...

[    0.000000] Booting Linux on physical CPU 0x0
[    0.000000] Linux version 4.19.59-g5f8c1c6121 (oe-user@oe-host) (gcc version 8.3.0 (GNU
Toolchain for the A-profile Architecture 8.3-2019.03 (arm-rel-8.36))) #1 PREEMPT Sat Oct 19
17:17:25 UTC 2019
[    0.000000] CPU: ARMv7 Processor [413fc082] revision 2 (ARMv7), cr=10c5387d

[    0.000000] OF: fdt: Machine model: TI AM335x Hello World

[    0.000000] Built 1 zonelists, mobility grouping on.  Total pages: 129920
[    0.000000] Kernel command line: console=ttyO0,115200n8 root=PARTUUID=0000000-02 rw
rootfstype=ext4 rootwait
```

Did the hello world work?

```
Starting kernel ...

[    0.000000] Booting Linux on physical CPU 0x0
[    0.000000] Linux version 4.19.59-g5f8c1c6121 (oe-user@oe-host) (gcc version 8.3.0 (GNU
Toolchain for the A-profile Architecture 8.3-2019.03 (arm-rel-8.36))) #1 PREEMPT Sat Oct 19
17:17:25 UTC 2019
[    0.000000] CPU: ARMv7 Processor [413fc082] revision 2 (ARMv7), cr=10c5387d

[    0.000000] OF: fdt: Machine model: TI AM335x Hello World

[    0.000000] Built 1 zonelists, mobility
[    0.000000] Kernel command line: consol
rootfstype=ext4 rootwait
```

```
/{
    model = "TI AM335x Hello World ";
```

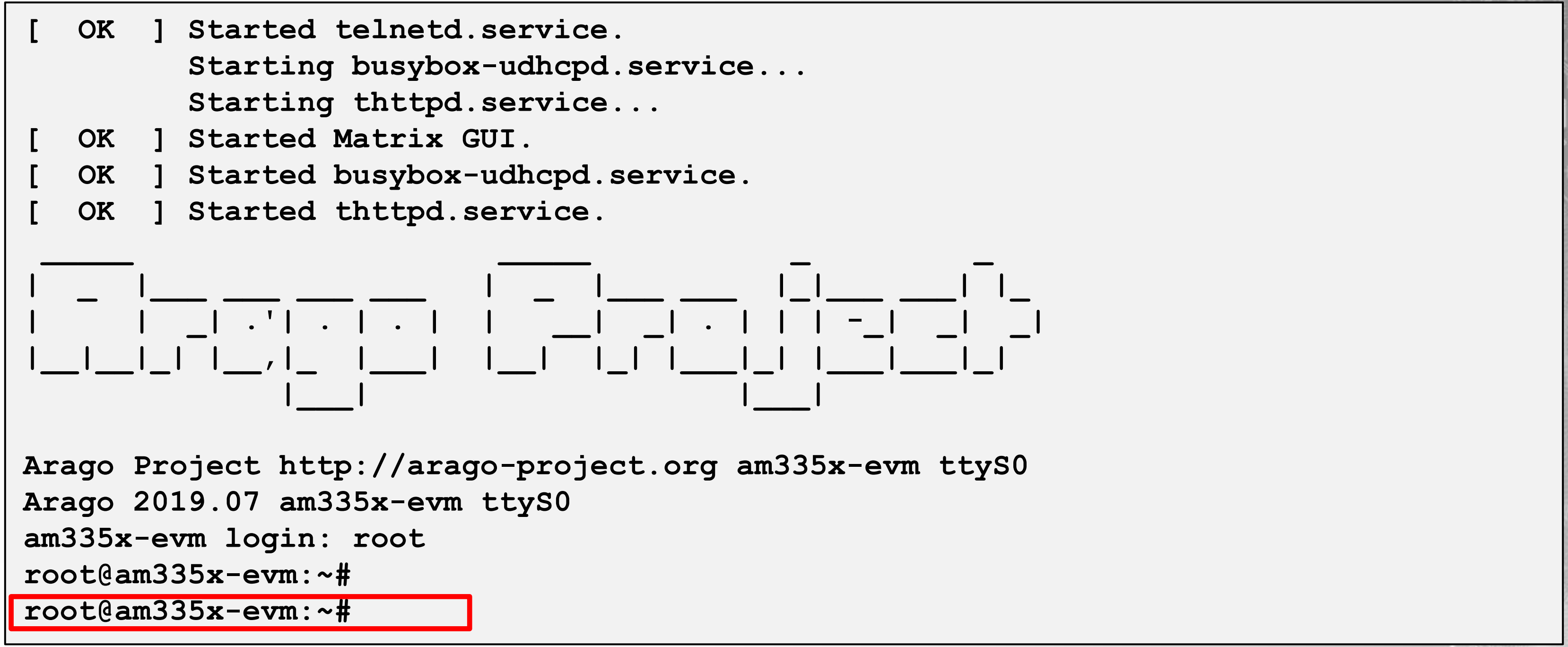
Texas Instruments

Did the hello world work?

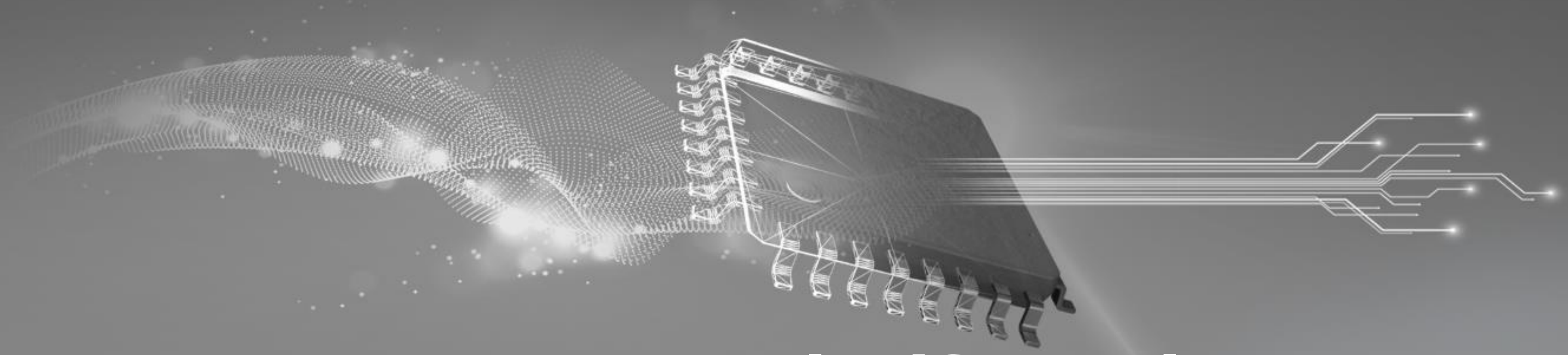
```
[    0.980750] sdhci: Secure Digital Host Controller Interface driver
[    0.987072] sdhci: Copyright(c) Pierre Ossman
[    0.992211] omap_gpio 44e07000.gpio: Could not set line 6 debounce to 200000 microseconds (-
22)
[    1.001011] omap_hsmmc 48060000.mmc: Got CD GPIO
[    1.006261] omap_hsmmc 48060000.mmc: Linked as a consumer to regulator.1

[    1.102726] mmc0: host does not support reading read-only switch, assuming write-enable

[    1.122632] mmc0: new high speed SDHC card at address 0007

[    1.136358] mmcblk0: mmc0:0007 SD08G 7.42 GiB
[    1.143832] mmcblk0: p1 p2

[    1.182949] EXT4-fs (mmcblk0p2): mounted files
[    1.191317] VFS: Mounted root (ext4 filesystem

[    1.217988] Run /sbin/init as init process
```

```
&mmc1 {
        status = "okay";
        bus-width = <0x4>;
        pinctrl-names = "default";
        pinctrl-0 = <&mmc1_pins>;
        cd-gpios = <&gpio0 6 GPIO_ACTIVE_LOW>;
        vmmc-supply = <&vmmcsd_fixed>;
};
```

Did the hello world work?

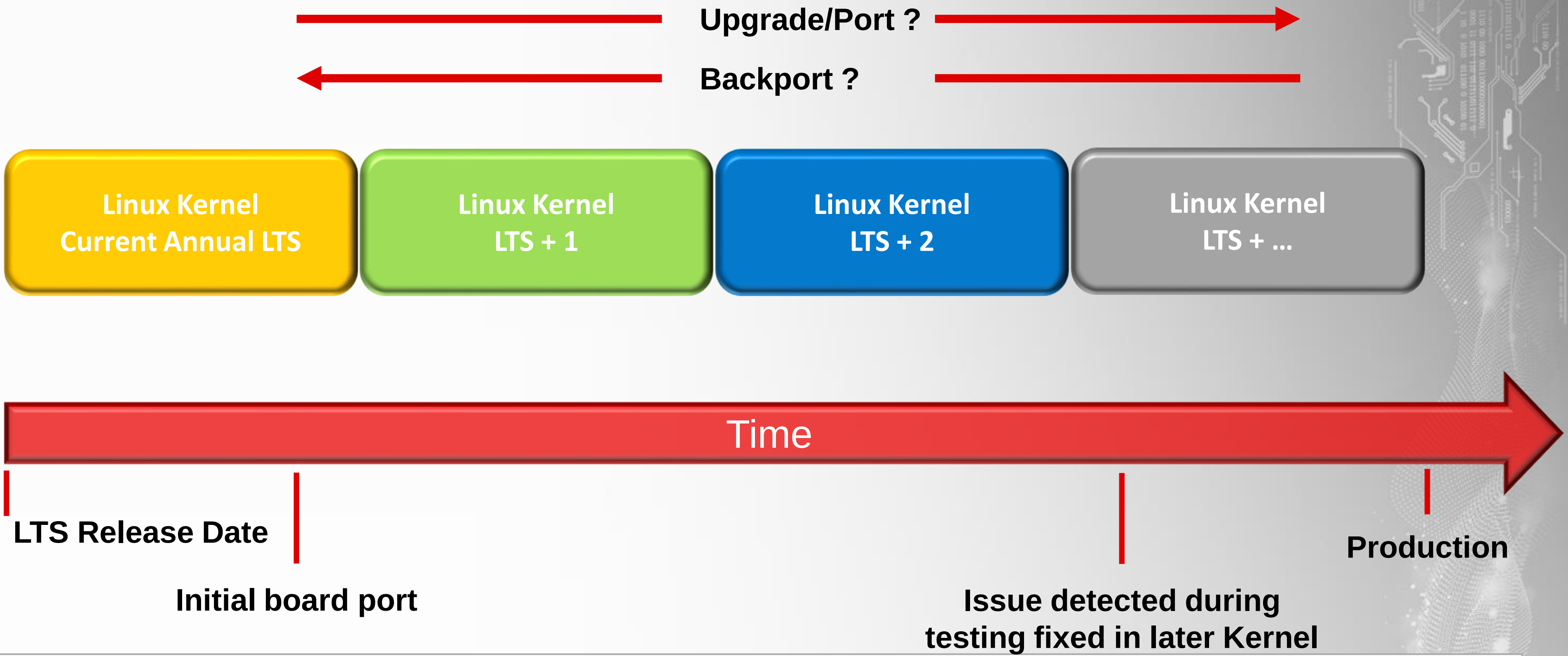


```
[  OK  ] Started telnetd.service.
         Starting busybox-udhcpd.service...
         Starting thttpd.service...
[  OK  ] Started Matrix GUI.
[  OK  ] Started busybox-udhcpd.service.
[  OK  ] Started thttpd.service.

 _____                    _____           _         _
|  _  |___ ___ ___ ___   |  _  |___ ___  |_|___ ___| |_
|     |  _| .'| . | . |  |   __|  _| . | | | -_|  _|  _|
|__|__|_| |__,|_  |___|  |__|  |_| |___|_| |___|___|_|
              |___|                    |___|

Arago Project http://arago-project.org am335x-evm ttyS0
Arago 2019.07 am335x-evm ttyS0
am335x-evm login: root
root@am335x-evm:~#
root@am335x-evm:~#
```



LTS Kernel Lifecycle and the new Board

Linux Board Port – SDK Lifecycle



Upgrade/Port ?

Backport ?

Linux Kernel
Current Annual LTS

Linux Kernel
LTS + 1

Linux Kernel
LTS + 2

Linux Kernel
LTS + …

Time

LTS Release Date

Initial board port

Issue detected during
testing fixed in later Kernel

Production

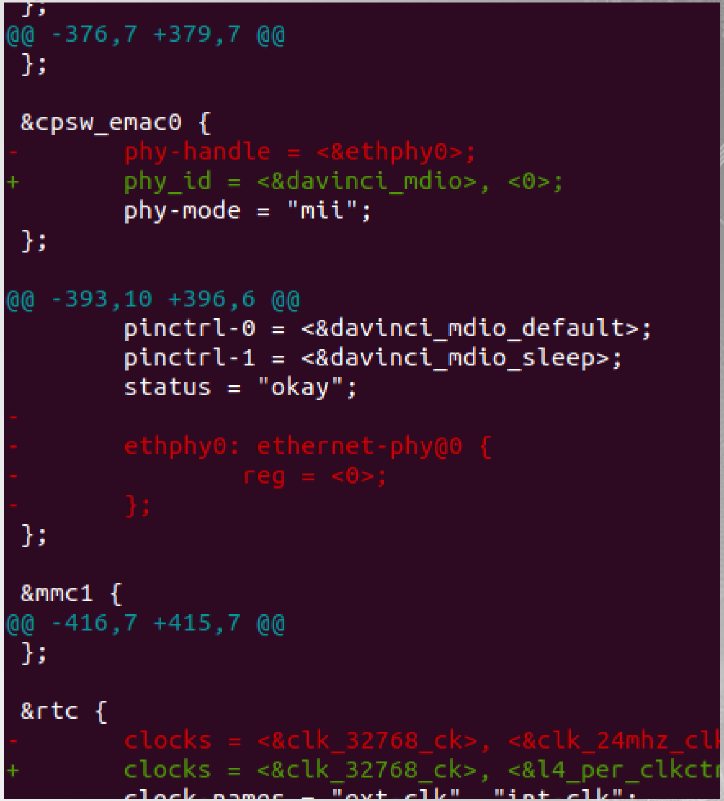
How Processor dtsi files change over time

- In one year there significant changes to am335x-bone-common.dtsi (one of include files for am335x-boneblack.dts) and more importantly the AM335x processor am33xx.dtsi file

- When a board dts is not updated or maintained to kernel version it mostly likely will fail to at least compile.

- Recommend keeping the board hello_world.dts up to date.

- Never modify the processor dtsi file, makes upgrades that much more difficult

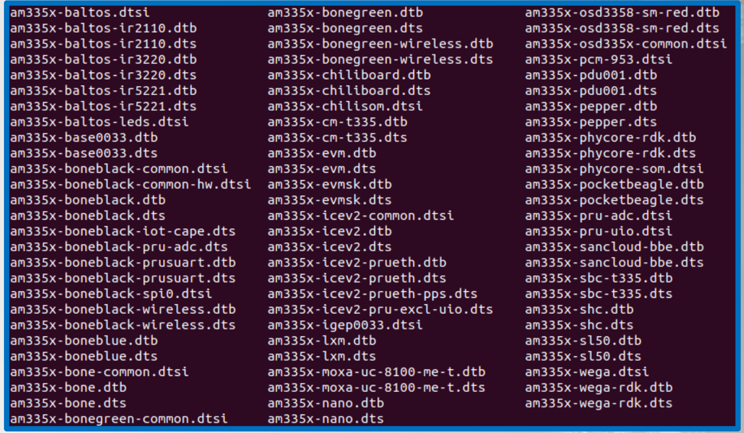


```
 };
@@ -376,7 +379,7 @@
 };

 &cpsw_emac0 {
-        phy-handle = <&ethphy0>;
+        phy_id = <&davinci_mdio>, <0>;
         phy-mode = "mii";
 };

@@ -393,10 +396,6 @@
         pinctrl-0 = <&davinci_mdio_default>;
         pinctrl-1 = <&davinci_mdio_sleep>;
         status = "okay";
-
-        ethphy0: ethernet-phy@0 {
-                reg = <0>;
-        };
 };

 &mmc1 {
@@ -416,7 +415,7 @@
 };

 &rtc {
-        clocks = <&clk_32768_ck>, <&clk_24mhz_clk
+        clocks = <&clk_32768_ck>, <&l4_per_clkctr
         clock-names = "ext-clk", "int-clk";
```

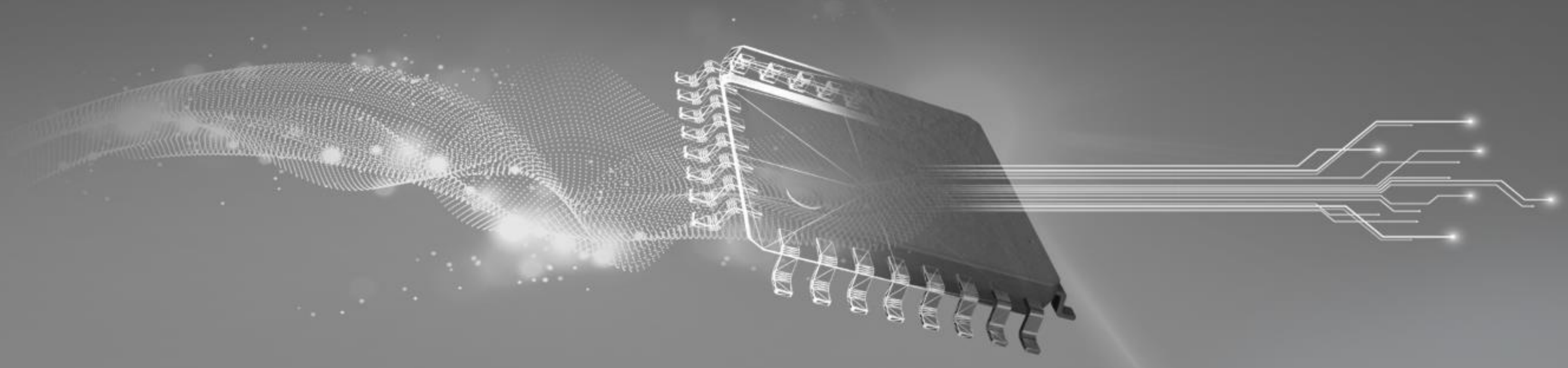
Keep at least the Hello World file current

- Reasons for recommending to at least keeping the board hello_world.dts up to date with the latest kernel.
  - Board Design Refresh
    - (components EOL)
  - Revision of board design
    - (debating upgrading kernels as well)
  - Looking to see if a later driver might fix an issue, don’t have to do a full upgrade.

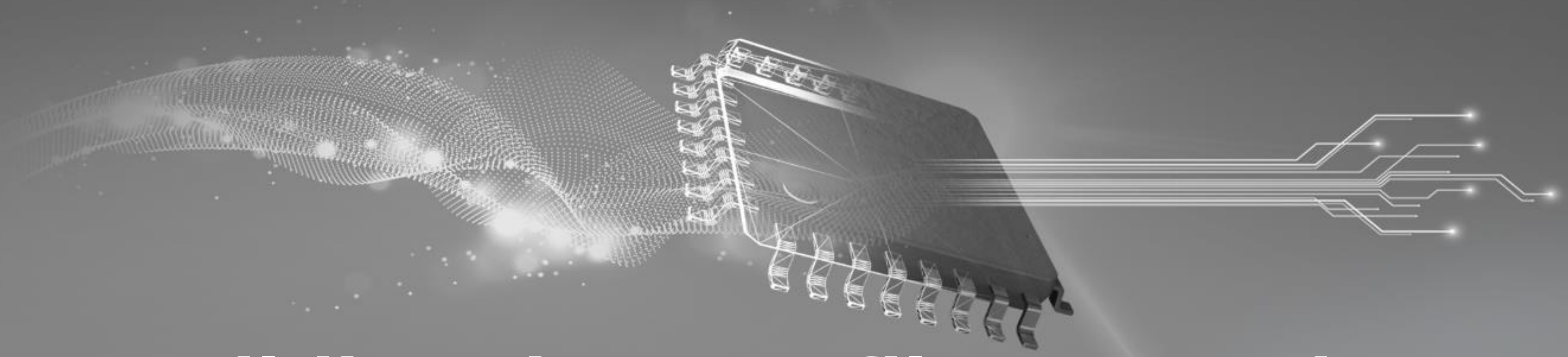
Existing Design Board Inventory



```
am335x-baltos.dtsi               am335x-bonegreen.dtb             am335x-osd3358-sm-red.dtb
am335x-baltos-ir2110.dtb         am335x-bonegreen.dts             am335x-osd3358-sm-red.dts
am335x-baltos-ir2110.dts         am335x-bonegreen-wireless.dtb    am335x-osd335x-common.dtsi
am335x-baltos-ir3220.dtb         am335x-bonegreen-wireless.dts    am335x-pcm-953.dtsi
am335x-baltos-ir3220.dts         am335x-chiliboard.dtb            am335x-pdu001.dtb
am335x-baltos-ir5221.dtb         am335x-chiliboard.dts            am335x-pdu001.dts
am335x-baltos-ir5221.dts         am335x-chilisom.dtsi             am335x-pepper.dtb
am335x-baltos-leds.dtsi          am335x-cm-t335.dtb               am335x-pepper.dts
am335x-base0033.dtb              am335x-cm-t335.dts               am335x-phycore-rdk.dtb
am335x-base0033.dts              am335x-evm.dtb                   am335x-phycore-rdk.dts
am335x-boneblack-common.dtsi     am335x-evm.dts                   am335x-phycore-som.dtsi
am335x-boneblack-common-hw.dtsi  am335x-evmsk.dtb                 am335x-pocketbeagle.dtb
am335x-boneblack.dtb             am335x-evmsk.dts                 am335x-pocketbeagle.dts
am335x-boneblack.dts             am335x-icev2-common.dtsi         am335x-pru-adc.dtsi
am335x-boneblack-iot-cape.dts    am335x-icev2.dtb                 am335x-pru-uio.dtsi
am335x-boneblack-pru-adc.dts     am335x-icev2.dts                 am335x-sancloud-bbe.dtb
am335x-boneblack-prusuart.dtb    am335x-icev2-prueth.dtb          am335x-sancloud-bbe.dts
am335x-boneblack-prusuart.dts    am335x-icev2-prueth.dts          am335x-sbc-t335.dtb
am335x-boneblack-spi0.dtsi       am335x-icev2-prueth-pps.dts      am335x-sbc-t335.dts
am335x-boneblack-wireless.dtb    am335x-icev2-pru-excl-uio.dts    am335x-shc.dtb
am335x-boneblack-wireless.dts    am335x-igep0033.dtsi             am335x-shc.dts
am335x-boneblue.dtb              am335x-lxm.dtb                   am335x-sl50.dtb
am335x-boneblue.dts              am335x-lxm.dts                   am335x-sl50.dts
am335x-bone-common.dtsi          am335x-moxa-uc-8100-me-t.dtb     am335x-wega.dtsi
am335x-bone.dtb                  am335x-moxa-uc-8100-me-t.dts     am335x-wega-rdk.dtb
am335x-bone.dts                  am335x-nano.dtb                  am335x-wega-rdk.dts
am335x-bonegreen-common.dtsi     am335x-nano.dts
```

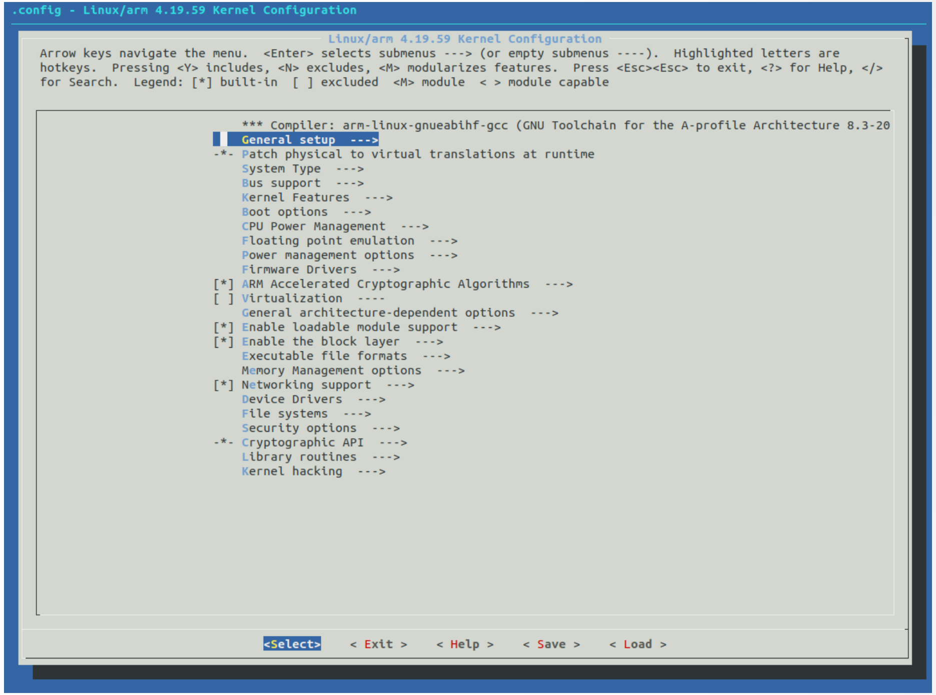


Q/A



Building the root filesystem into the Linux Kernel

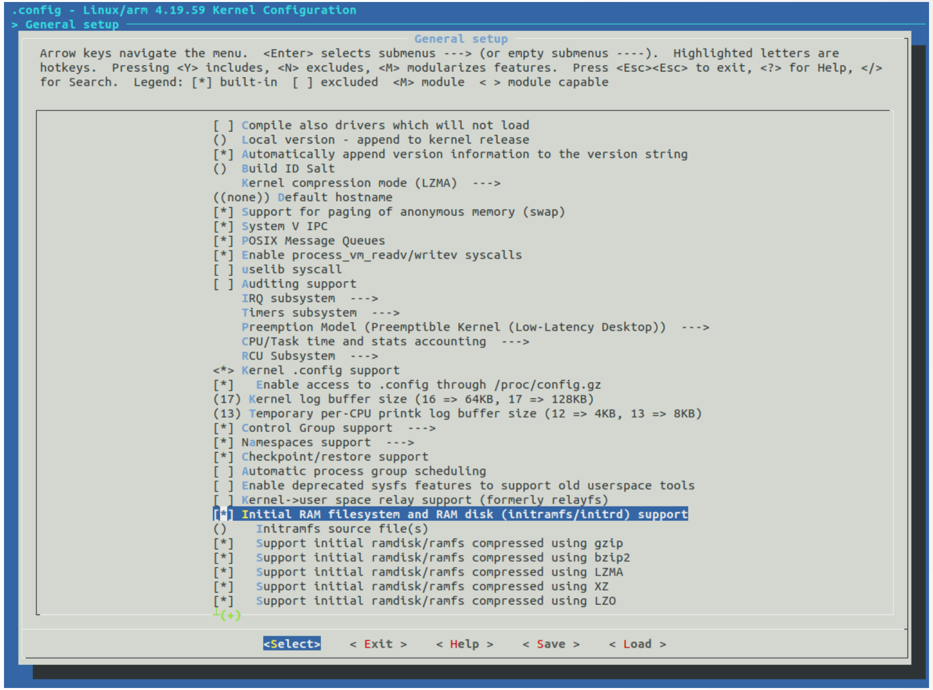
Building the root filesystem into the kernel



```
.config - Linux/arm 4.19.59 Kernel Configuration
                          Linux/arm 4.19.59 Kernel Configuration
 Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted letters are
 hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </>
 for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

              *** Compiler: arm-linux-gnueabihf-gcc (GNU Toolchain for the A-profile Architecture 8.3-20
              General setup  --->
          -*- Patch physical to virtual translations at runtime
              System Type  --->
              Bus support  --->
              Kernel Features  --->
              Boot options  --->
              CPU Power Management  --->
              Floating point emulation  --->
              Power management options  --->
              Firmware Drivers  --->
          [*] ARM Accelerated Cryptographic Algorithms  --->
          [ ] Virtualization  ----
              General architecture-dependent options  --->
          [*] Enable loadable module support  --->
          [*] Enable the block layer  --->
              Executable file formats  --->
              Memory Management options  --->
          [*] Networking support  --->
              Device Drivers  --->
              File systems  --->
              Security options  --->
          -*- Cryptographic API  --->
              Library routines  --->
              Kernel hacking  --->

                 <Select>    < Exit >    < Help >    < Save >    < Load >
```

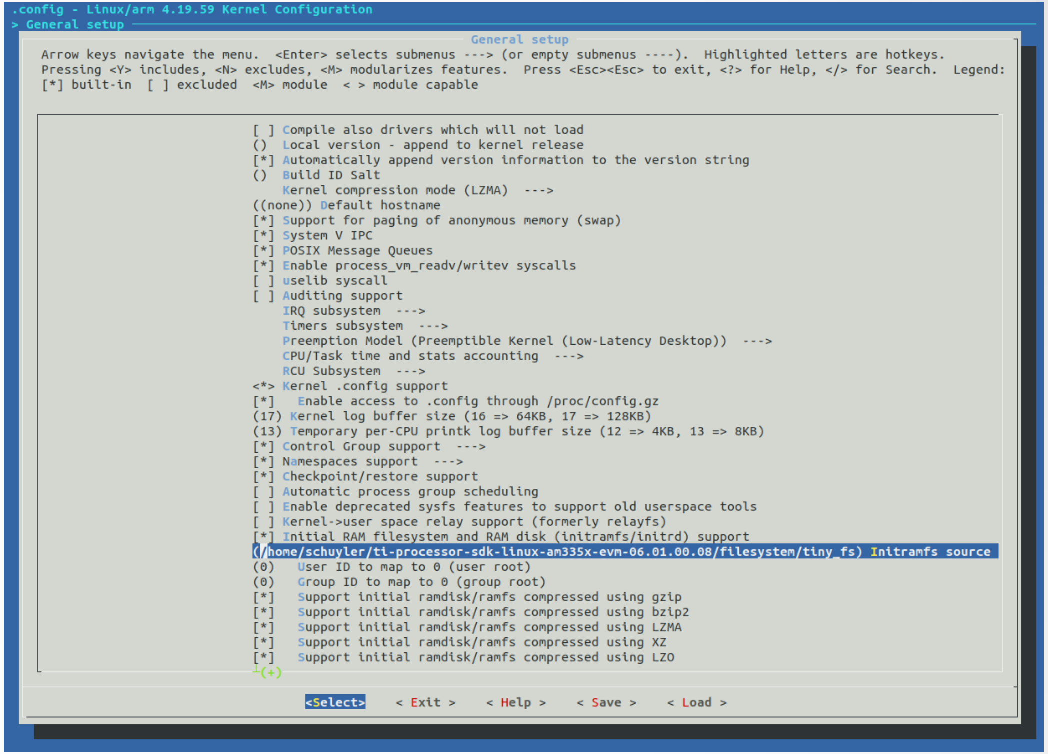
- If there is an issue with the board for the interface that would be used for the root filesystem this technique can be used to bypass the interface, boot the kernel and perhaps debug the problematic interface.
- Need to find a source for the root filesytem, needs to be small.
- Make sure the linux utilities needed for debug are in the filesystem.

Building the root filesystem into the kernel



```
.config - Linux/arm 4.19.59 Kernel Configuration
> General setup
                                  General setup
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted letters are
  hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </>
  for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

                        [ ] Compile also drivers which will not load
                        ()  Local version - append to kernel release
                        [*] Automatically append version information to the version string
                        ()  Build ID Salt
                            Kernel compression mode (LZMA)  --->
                        ((none)) Default hostname
                        [*] Support for paging of anonymous memory (swap)
                        [*] System V IPC
                        [*] POSIX Message Queues
                        [*] Enable process_vm_readv/writev syscalls
                        [ ] uselib syscall
                        [ ] Auditing support
                            IRQ subsystem  --->
                            Timers subsystem  --->
                            Preemption Model (Preemptible Kernel (Low-Latency Desktop))  --->
                            CPU/Task time and stats accounting  --->
                            RCU Subsystem  --->
                        <*> Kernel .config support
                        [*]   Enable access to .config through /proc/config.gz
                        (17) Kernel log buffer size (16 => 64KB, 17 => 128KB)
                        (13) Temporary per-CPU printk log buffer size (12 => 4KB, 13 => 8KB)
                        [*] Control Group support  --->
                        [*] Namespaces support  --->
                        [*] Checkpoint/restore support
                        [ ] Automatic process group scheduling
                        [ ] Enable deprecated sysfs features to support old userspace tools
                        [ ] Kernel->user space relay support (formerly relayfs)
                        [*] Initial RAM filesystem and RAM disk (initramfs/initrd) support
                        ()    Initramfs source file(s)
                        [*]   Support initial ramdisk/ramfs compressed using gzip
                        [*]   Support initial ramdisk/ramfs compressed using bzip2
                        [*]   Support initial ramdisk/ramfs compressed using LZMA
                        [*]   Support initial ramdisk/ramfs compressed using XZ
                        [*]   Support initial ramdisk/ramfs compressed using LZO
                        └(+)

                     <Select>    < Exit >    < Help >    < Save >    < Load >
```

Building the root filesystem into the kernel



```
.config - Linux/arm 4.19.59 Kernel Configuration
> General setup
                                    General setup
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted letters are hotkeys.
  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend:
  [*] built-in  [ ] excluded  <M> module  < > module capable

                            [ ] Compile also drivers which will not load
                            ()  Local version - append to kernel release
                            [*] Automatically append version information to the version string
                            ()  Build ID Salt
                                Kernel compression mode (LZMA)  --->
                            ((none)) Default hostname
                            [*] Support for paging of anonymous memory (swap)
                            [*] System V IPC
                            [*] POSIX Message Queues
                            [*] Enable process_vm_readv/writev syscalls
                            [ ] uselib syscall
                            [ ] Auditing support
                                IRQ subsystem  --->
                                Timers subsystem  --->
                                Preemption Model (Preemptible Kernel (Low-Latency Desktop))  --->
                                CPU/Task time and stats accounting  --->
                                RCU Subsystem  --->
                            <*> Kernel .config support
                            [*]   Enable access to .config through /proc/config.gz
                            (17) Kernel log buffer size (16 => 64KB, 17 => 128KB)
                            (13) Temporary per-CPU printk log buffer size (12 => 4KB, 13 => 8KB)
                            [*] Control Group support  --->
                            [*] Namespaces support  --->
                            [*] Checkpoint/restore support
                            [ ] Automatic process group scheduling
                            [ ] Enable deprecated sysfs features to support old userspace tools
                            [ ] Kernel->user space relay support (formerly relayfs)
                            [*] Initial RAM filesystem and RAM disk (initramfs/initrd) support
                            (/home/schuyler/ti-processor-sdk-linux-am335x-evm-06.01.00.08/filesystem/tiny_fs) Initramfs source
                            (0)   User ID to map to 0 (user root)
                            (0)   Group ID to map to 0 (group root)
                            [*]   Support initial ramdisk/ramfs compressed using gzip
                            [*]   Support initial ramdisk/ramfs compressed using bzip2
                            [*]   Support initial ramdisk/ramfs compressed using LZMA
                            [*]   Support initial ramdisk/ramfs compressed using XZ
                            [*]   Support initial ramdisk/ramfs compressed using LZO
                            (+)

                   <Select>    < Exit >    < Help >    < Save >    < Load >
```

- This increases the size of the kernel so take that into account as the kernel is read into DDR by U-Boot.

Thank You!