Agenda

1. Platform configuration
2. **Device tree**
3. ACPI

Module configuration

- module_param()
- Platform configuration
- Device tree
- ACPI

module_param()

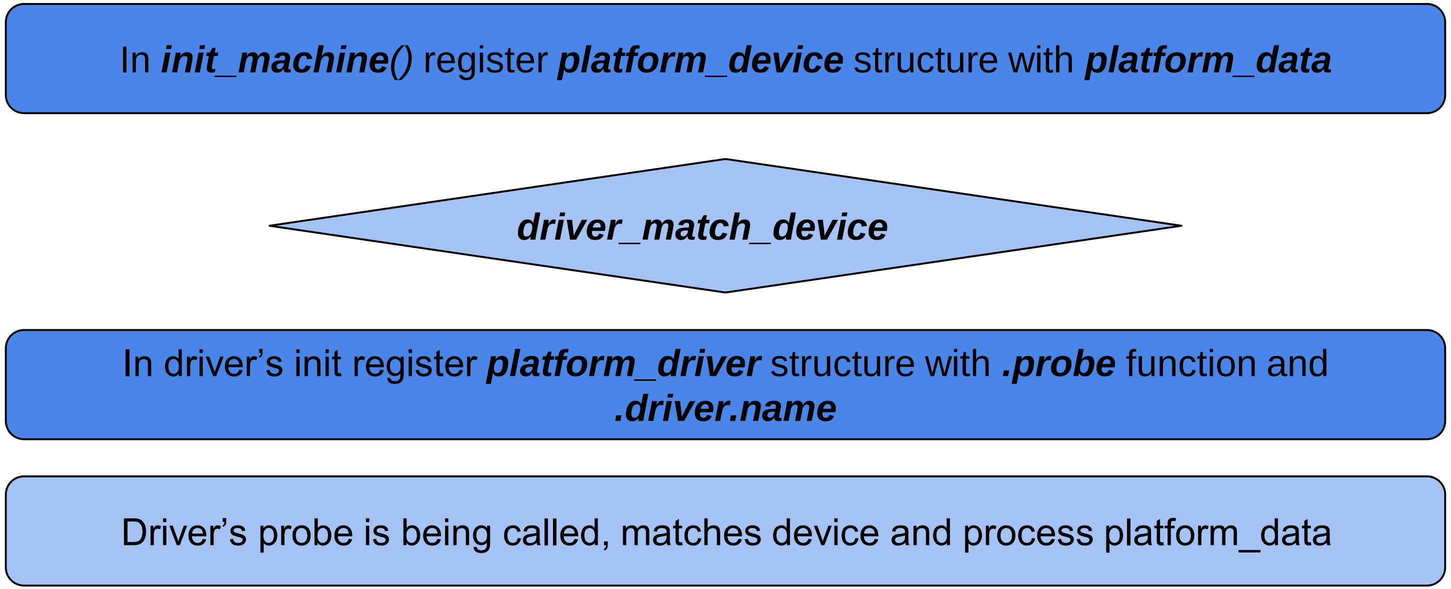
Use the transfer of information in the driver for certain configuration parameters.

First of all, these are the parameters of the modules through the macro **module_param()**.

The value of these parameters can be specified in two ways :

- **while loading (on the command line insmod or modprobe)** https://www.ibm.com/developerworks/ru/library/l-linux_kernel_11/index.html
- **can be specified in Kernel command line** https://landlock.io/linux-doc/landlock-v7/admin-guide/kernel-parameters.html

Driver configuration via platform device



In ***init_machine()*** register ***platform_device*** structure with ***platform_data***

driver_match_device

In driver's init register ***platform_driver*** structure with ***.probe*** function and ***.driver.name***

Driver's probe is being called, matches device and process platform_data

Platform configuration

Platform definitions:

- include/linux/platform_device.h
  - *struct **platform_device***
  - *platform_device_register()*, *platform_add_devices()*
  - *struct **platform_driver***
  - *platform_driver_register()*
- include/linux/device.h
  - *struct **device***
  - *struct **device_driver***

Machine definitions are in */arch/arm/mach-XXX/board-XXXYYY*

- Look at MACHINE_START definition (entry point is *.init_machine()*)

Platform configuration

Consider the example of ARM under the platform Beagle Bone Black (family omap).

linux/arch/arm/mach-omap2/

In / arch is a list of supported architectures. We are interested in the platform Beagle Bone, so choose mach-omap2. https://lxr.missinglinkelectronics.com/linux+v4.9/arch/arm/mach-omap2/.

In mach-omap2, in addition to the general code, the subsystem subsystem (as an example of clock data, I/O ports, etc.), board files

Device Tree

Devicetree provides description of a platform hardware to drivers. (see ePAPR)

Devicetree (open firmware) API:

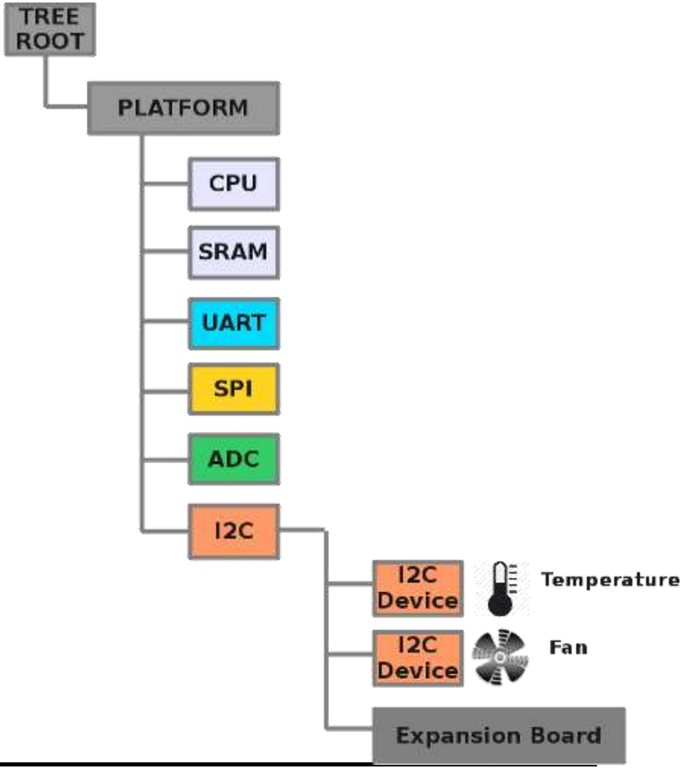
- include/linux/of.h
- include/linux/mod_devicetable.h
    - *struct **of_device_id***
- include/linux/module.h
    - ***MODULE_DEVICE_TABLE***
- drivers/of/

Note: For omap2 DT_MACHINE_START are now moved to arch/arm/mach-omap2/board-generic.c

Device Tree

Device Tree

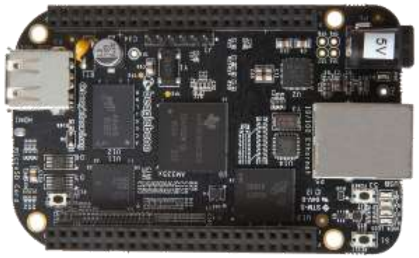
See http://patternagents.com/news/2015/01/28/devicetree-overview.html

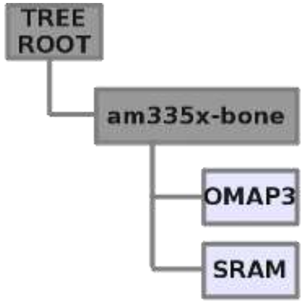


Device Tree

- Name of a node

- The name of a node should be somewhat generic, reflecting the function of the device and not its precise programming Model.
  - adc
  - accelerometer
  - atm
  - audio-codec
  - audio-controller
  - Backlight





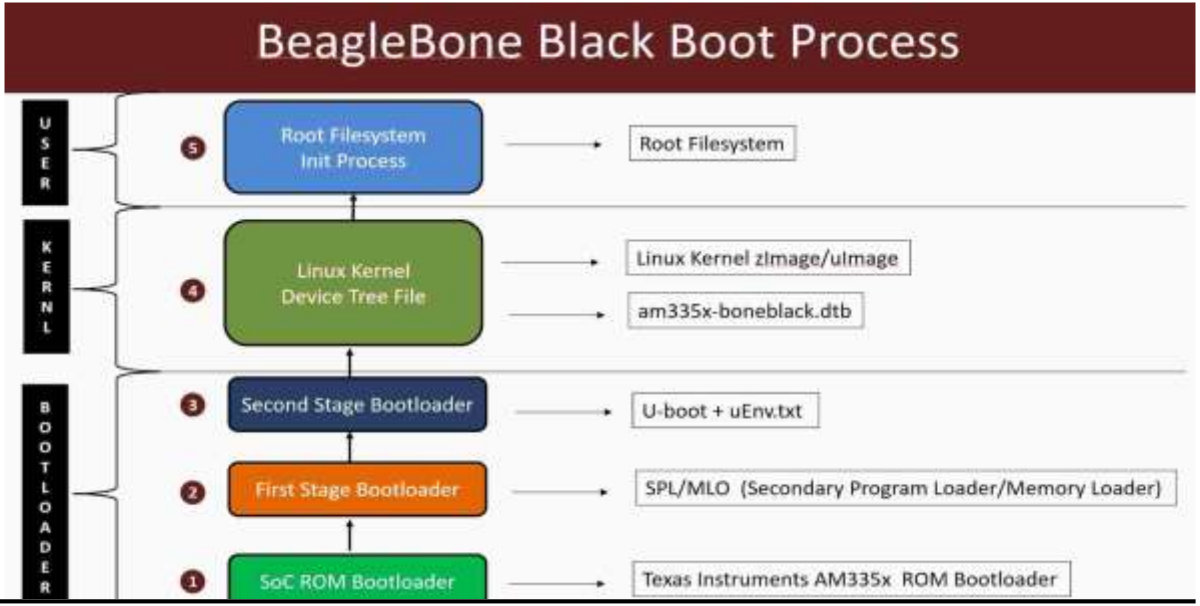
devicetree.org

Device Tree

Device Tree

See https://www.devicetree.org/

- Devicetree Specification 0.2


BeagleBone Black Boot Process
USER
KERNL
BOOTLOADER
5
Root Filesystem Init Process
Root Filesystem
4
Linux Kernel Device Tree File
Linux Kernel zImage/uImage
am335x-boneblack.dtb
3
Second Stage Bootloader
U-boot + uEnv.txt
2
First Stage Bootloader
SPL/MLO (Secondary Program Loader/Memory Loader)
1
SoC ROM Bootloader
Texas Instruments AM335x ROM Bootloader

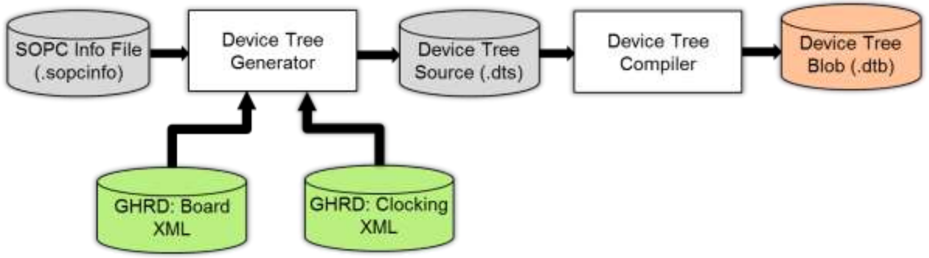
Device Tree - terminology

- **Dtb** - Device Tree Blob
- **Dts** - Device Tree Source
- **Dtbs** - Device Tree Source and Device Tree Blob
- **Device Tree Overlays** - You need a way to describe these optional components using a partial device tree, and then be able to build a complete tree by taking the base DT and adding a number of optional elements. You can do this, and these additional elements are called overlays.
  See https://www.raspberrypi.org/documentation/configuration/device-tree.md

https://elinux.org/Device_Tree_Reference

Installing the Device Tree Compiler

- sudo apt-get install -y **device-tree-compiler**



- dtc -v

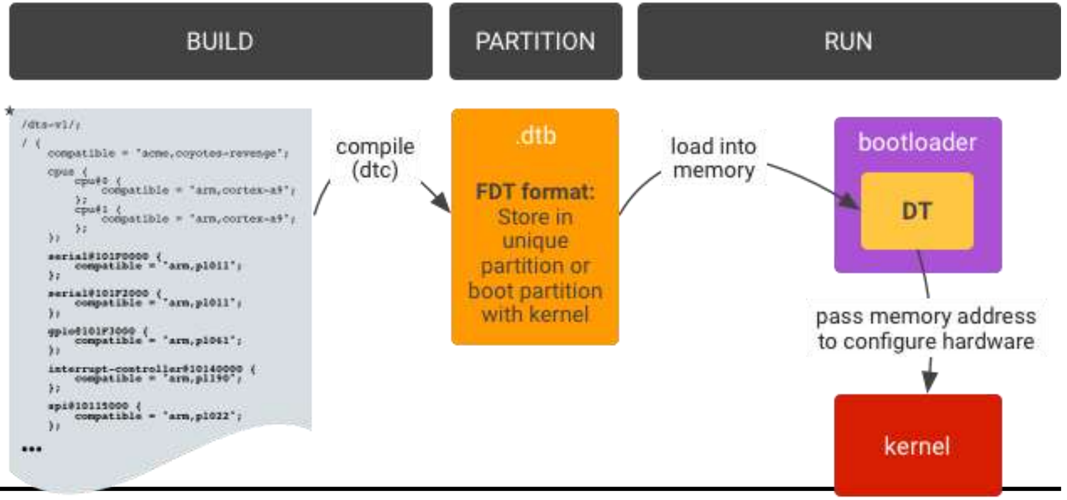
  Version: DTC 1.4.0

- **dtc -O dtb -o outputBLOB.dtb -b 0 inputSOURCE.dts**

Device Tree

To build:

- Use the device tree compiler (dtc) to compile device tree source (.dts) into a device tree blob (.dtb), formatted as a flattened device tree.
- Flash the .dtb file into a bootloader runtime-accessible location

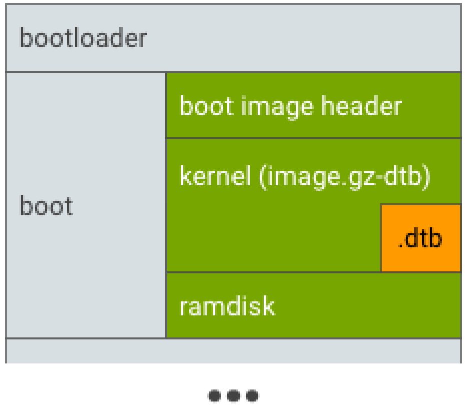

BUILD
PARTITION
RUN
*
compile (dtc)
.dtb
FDT format: Store in unique partition or boot partition with kernel
load into memory
bootloader
DT
pass memory address to configure hardware
kernel

```
/dts-v1/;
/ {
    compatible = "acme,coyotes-revenge";
    cpus {
        cpu@0 {
            compatible = "arm,cortex-a9";
        };
        cpu@1 {
            compatible = "arm,cortex-a9";
        };
    };
    serial@101F0000 {
        compatible = "arm,pl011";
    };
    serial@101F2000 {
        compatible = "arm,pl011";
    };
    gpio@101F3000 {
        compatible = "arm,pl061";
    };
    interrupt-controller@10140000 {
        compatible = "arm,pl190";
    };
    spi@10115000 {
        compatible = "arm,pl022";
    };
...
```

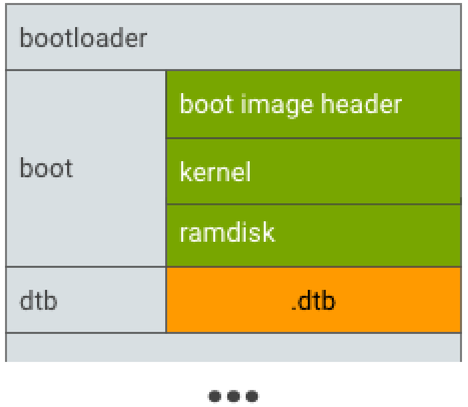
*http://elinux.org/Device_Tree_Usage#Devices

Device Tree

Put .dtb in boot partition by appending to image.gz and passing as "kernel" to mkbootimg



Put .dtb in an unique partition (e.g. dtb partition)



Device tree syntax

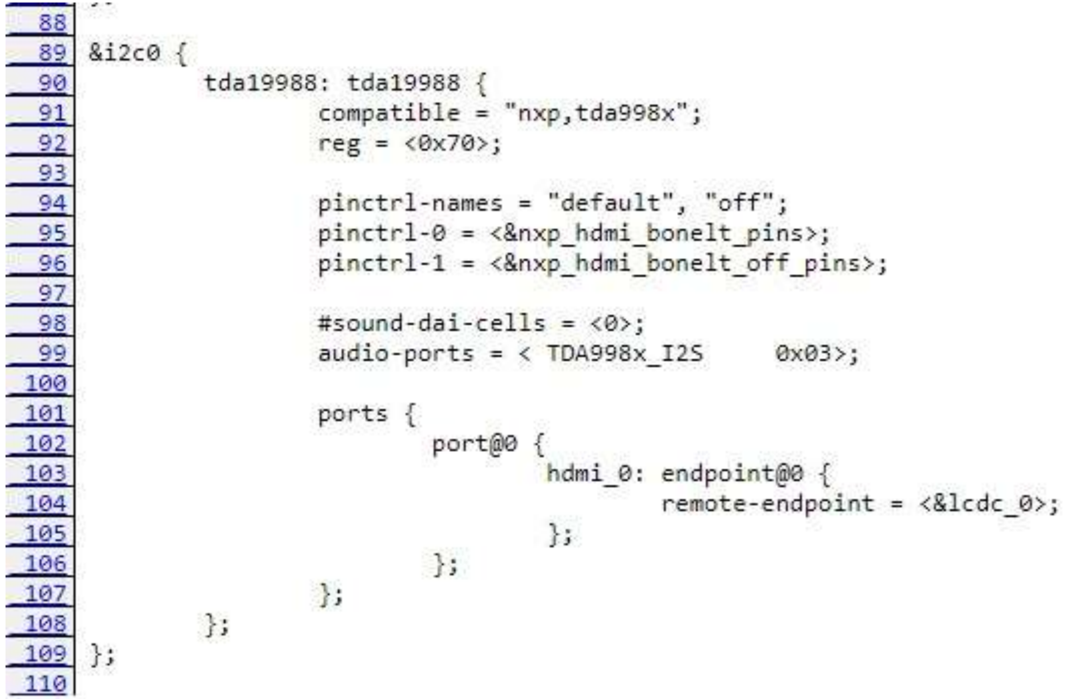
Format:

```
/dts-v1/;

/ {
   node1 {
      a-string-property = "A string";
      a-string-list-property = "first string", "second string";
      // hex is implied in byte arrays. no '0x' prefix is required
      a-byte-data-property = [01 23 34 56];
      child-node1 {
         first-child-property;
         second-child-property = <1>;
         a-string-property = "Hello, world";
      };
      child-node2 {
      };
   };
   node2 {
      an-empty-property;
      a-cell-property = <1 2 3 4>; /* each number (cell) is a uint32 */
      child-node1 {
      };
   };
};
```

Example:

```
dbmdx {
                status = "okay";
                compatible = "dspg,dbmdx-codec";

                qcom,use-pinctrl;
                pinctrl-names = "dbmdx_default",
"dbmdx_sleep";

                pinctrl-0 = <&dbmdx_active>;
                pinctrl-1 = <&dbmdx_sleep>;

                sv-gpio = <&tlmm 42 0>; /* VOICE_INT */
                wakeup-gpio = <&pm8994_mpps 7 0>; /*
VOICE_WAKE */

                /* feature-vqe; */ /* enable VQE */
                /* feature-firmware-overlay; */
                va-firmware-name = "dbmd4_va_fw.bin";
                /* vqe-firmware-name =
"dbmd4_vqe_fw.bin"; */

                master-clk-rate = <32768>;
                /* constant-clk-rate = <32768>; */
                auto_detection = <1>;
                detection_buffer_channels = <0>;
                pcm_streaming_mode = <1>;
                firmware_id = <0xdbd4>;
                use_gpio_for_wakeup = <1>; /* Use
wakeup gpio */

                wakeup_set_value = <0>;  /* Value to write
to wakeup gpio */
```

Driver example



```

&i2c0 {
        tda19988: tda19988 {
                compatible = "nxp,tda998x";
                reg = <0x70>;

                pinctrl-names = "default", "off";
                pinctrl-0 = <&nxp_hdmi_bonelt_pins>;
                pinctrl-1 = <&nxp_hdmi_bonelt_off_pins>;

                #sound-dai-cells = <0>;
                audio-ports = < TDA998x_I2S     0x03>;

                ports {
                        port@0 {
                                hdmi_0: endpoint@0 {
                                        remote-endpoint = <&lcdc_0>;
                                };
                        };
                };
        };
};

```

Links on the topic of module_param()

https://www.ibm.com/developerworks/ru/library/l-linux_kernel_11/index.html

https://landlock.io/linux-doc/landlock-v7/admin-guide/kernel-parameters.html

Links on the topic of platform configuration :
https://lxr.missinglinkelectronics.com/linux+v4.9/arch/arm/mach-omap2/.

Links on the topic of Device Tree

https://lxr.missinglinkelectronics.com/linux+v4.9/arch/arm/boot/dts/

https://lxr.missinglinkelectronics.com/linux+v4.9/Documentation/driver-model/platform.txt

https://lxr.missinglinkelectronics.com/linux+v4.9/Documentation/devicetree/usage-model.txt

https://elinux.org/images/c/cf/Power_ePAPR_APPROVED_v1.1.pdf

Home reading

- *Documentation/driver-model/platform.txt*
- *Documentation/devicetree/*
- ePAPR

Внимание!

- ***Драйвер*** *не должен быть написан с поддержкой только одного конкретного механизма конфигурации.*
- *Так как драйвер собирается с конкретным ядром для конкретной конфигурации ядра и в зависимости от того что в конфигурации ядра включено (мы включим) то и будет использоваться для конфигурации.*
- *Драйвер, при старте может загрузить либо Device tree или ACPI конфигурацию. В ядре какая-то часть может остаться сконфигурирована через таблицы* ***Platform configuration****, задача драйвера попытаться найти свою конфигурацию проверяя, если доступны записи* ***Device tree*** *или* ***ACPI*** *(либо в machine configuration «захардкожено»).*