

Interpreting Device Tree Syntax using the Raspberry Pi 3B+'s DT snippets as an example

Source: [DTSpec manual here](#) (ver 0.3-rc2, June 2019, of the specification)

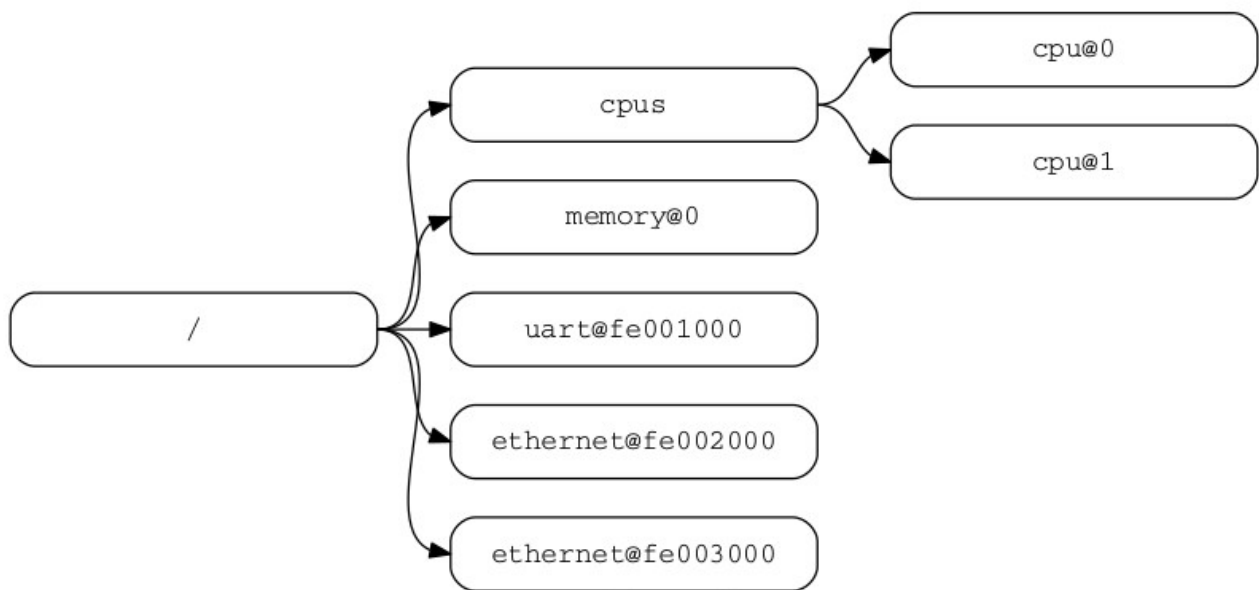
Raspberry Pi - Device Trees, overlays, and parameters

<https://www.raspberrypi.org/documentation/configuration/device-tree.md>

Raspberry Pi 3B+ DTS file location

<https://github.com/raspberrypi/linux/blob/rpi-4.19.y/arch/arm/boot/dts/bcm2837-rpi-3-b-plus.dts>

A few regularly used properties are shown here; for the full details please refer [the DTSpec manual here](#) (ver 0.3-rc2, June 2019, of the specification).



Examples of Node Names

Absolute bare minimum:

a root node '/', one 'cpus/' node, one or more 'memory/' nodes.

model

```
model = "Raspberry Pi 3 Model B Plus Rev 1.3";
```

The model property value is a <string> that specifies the manufacturer's model number of the device.

The recommended format is: "manufacturer,model", where manufacturer is a string describing the name of the manufacturer (such as a stock ticker symbol), and model specifies the model number.

#address-cells and #size-cells

```
soc {  
    ...  
    #address-cells = <0x1>;  
    #size-cells = <0x1>;
```

The #address-cells and #size-cells properties may be used in any device node that has children in the device-tree hierarchy and describes how child device nodes should be addressed. The #address-cells property defines the number of <u32> cells used to encode the address field in a child node's reg property. The #size-cells property defines the number of <u32> cells used to encode the size field in a child node's reg property.

In this example, the #address-cells and #size-cells properties of the soc node are both set to 1. This setting specifies that one cell is required to represent an address and one cell is required to represent the size of nodes that are children of this node.

The serial device reg property necessarily follows this specification set in the parent (soc) node—the address is represented by a single cell (0x4600), and the size is represented by a single cell (0x100).

compatible

```
arm-pmu {  
    interrupts = <0x9>;  
    interrupt-parent = <0x3>;  
    compatible = "arm,armv8-pmu3", "arm,cortex-a7-pmu";  
};
```

The compatible property value consists of one or more strings that define the specific programming model for the device. This list of strings should be used **by a client program for device driver selection**. The property value consists of a concatenated list of null terminated strings, from most specific to most general. They allow a device to express its compatibility with a family of similar devices, potentially allowing a single device driver to match against several devices.

The recommended format is "manufacturer,model", where manufacturer is a string describing the name of the manufacturer (such as a stock ticker symbol), and model specifies the model number.

<< **Rule: most-specific to most-general** ;

start with the most specific match, going on to less specific, more generic matches ; above, the client program will first attempt to use the ARM armv8-pmu3 driver for this device, and if not found, the more generic cortex-a7-pmu driver >>

Another eg. From the root node:

```
/ {  
    #address-cells = <0x1>;  
    memreserve = <0x3b400000 0x4c00000>;
```

```

model = "Raspberry Pi 3 Model B Plus Rev 1.3";
serial-number = "000000000906a583";
#size-cells = <0x1>;
interrupt-parent = <0x1>;
compatible = "raspberrypi,3-model-b-plus", "brcm,bcm2837";
...

```

reg

```

gpiomem {
    compatible = "brcm,bcm2835-gpiomem";
    reg = <0x7e200000 0x1000>;
};

```

Property name: reg

Property value: <prop-encoded-array> encoded as an arbitrary number of (address, length) pairs.

Description:

The reg property describes the address of the device's resources within the address space defined by its parent bus. **Most commonly this means the offsets and lengths of memory-mapped IO register blocks**, but may have a different meaning on some bus types. Addresses in the address space defined by the root node are CPU real addresses.

The value is a <prop-encoded-array>, composed of an arbitrary number of pairs of address and length, <address length>. The number of <u32> cells required to specify the address and length are bus-specific and are specified by the #address-cells and #size-cells properties in the parent of the device node. If the parent node specifies a value of 0 for #size-cells, the length field in the value of reg shall be omitted.

<< In the above example, the Raspberry Pi 3B+ GPIO registers are memory-mapped I/O from 0x7e20 0000 for a length of 0x1000 (4096) bytes. To verify this, we lookup the Broadcom TRM for the BCM2837 SoC and display a snippet of the register table for the GPIO register bank:



BCM2835 ARM Peripherals

The GPIO peripheral has three dedicated interrupt lines. These lines are triggered by the setting of bits in the event detect status register. Each bank has its' own interrupt line with the third line shared between all bits.

The Alternate function table also has the pull state (pull-up/pull-down) which is applied after a power down.

6.1 Register View

The GPIO has 41 registers. All accesses are assumed to be 32-bit.

Address	Field Name	Description	Size	Read/Write
0x 7E20 0000	GPFSEL0	GPIO Function Select 0	32	R/W
0x 7E20 0000	GPFSEL0	GPIO Function Select 0	32	R/W
0x 7E20 0004	GPFSEL1	GPIO Function Select 1	32	R/W
0x 7E20 0008	GPFSEL2	GPIO Function Select 2	32	R/W
0x 7E20 000C	GPFSEL3	GPIO Function Select 3	32	R/W
0x 7E20 0010	GPFSEL4	GPIO Function Select 4	32	R/W
0x 7E20 0014	GPFSEL5	GPIO Function Select 5	32	R/W
0x 7E20 0018	-	Reserved	-	-
0x 7E20 001C	GPSET0	GPIO Pin Output Set 0	32	W

Notice how, just as the DT states, the starting address is 0x7e20 0000.

>>

ranges

```
soc {  
    dma-ranges = <0xc0000000 0x0 0x3f000000>;  
    #address-cells = <0x1>;  
    #size-cells = <0x1>;  
    compatible = "simple-bus";  
    ranges = <0x7e000000 0x3f000000 0x1000000 0x40000000 0x40000000  
0x1000>;  
    phandle = <0x31>;  
    ...
```

Property name: ranges

Value type: <empty> or <prop-encoded-array> encoded as an arbitrary number of (child-bus-address, parent-bus-address, length) triplets.

Description:

The ranges property provides a means of **defining a mapping or translation** between the address space of the bus (the child address space) and the address space of the bus node's parent (the parent address space).

The format of the value of the ranges property is an arbitrary number of triplets of *(child-bus-address, parent-bus-address, length)* ...

chosen

```
chosen {
    linux,initrd-end = <0x434aa99>;
    bootargs = "8250.nr_uarts=1 bcm2708_fb.fbwidth=656
bcm2708_fb.fbheight=416 bcm2708_fb.fbswap=1 vc_mem.mem_base=0x3ec00000
vc_mem.mem_size=0x40000000 net.ifnames=0 dwc_otg.lpm_enable=0
console=ttyS0,115200 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4
elevator=deadline rootwait";
    kaslr-seed = <0x0 0x0>;
    linux,initrd-start = <0x3100000>;
};
```

The /chosen node does not represent a real device in the system but describes parameters chosen or specified by the system firmware at run time. It shall be a child of the root node.

Table 3.4: /chosen Node Properties

Property Name	Usage	Value Type	Definition
bootargs	O	<string>	A string that specifies the boot arguments for the client program. The value could potentially be a null string if no boot arguments are required.
stdout-path	O	<string>	A string that specifies the full path to the node representing the device to be used for boot console output. If the character ":" is present in the value it terminates the path. The value may be an alias. If the stdin-path property is not specified, stdout-path should be assumed to define the input device.
stdin-path	O	<string>	A string that specifies the full path to the node representing the device to be used for boot console input. If the character ":" is present in the value it terminates the path. The value may be an alias.
Usage legend: R=Required, O=Optional, OR=Optional but Recommended, SD=See Definition			

/cpus/cpu* Node Properties

```
cpus {
    #address-cells = <0x1>;
    #size-cells = <0x0>;
    enable-method = "brcm,bcm2836-smp";
    phandle = <0x72>;

    cpu@1 {
        clock-frequency = "SrN";
```

```

        device_type = "cpu";
        compatible = "arm,cortex-a53";
        reg = <0x1>;
        enable-method = "spin-table";
        phandle = <0x2a>;
        cpu-release-addr = <0x0 0xe0>;
    };
...

```

A /cpus node is required for all devicetrees. It does not represent a real device in the system, but acts as a container for child cpu nodes which represent the systems CPUs.

(Pl refer Table 3.6 in the manual).

F.e.:

“status : SD <string> : A standard property describing the state of a CPU. This property shall be present for nodes representing CPUs in a symmetric multiprocessing (SMP) configuration.

For a CPU node the meaning of the "okay" and "disabled" values are as follows:

"okay" : The CPU is running.

"disabled" : The CPU is in a quiescent state.

...

enable-method : SD <stringlist> : Describes the method by which a CPU in a disabled state is enabled. This property is required for CPUs with a status property with a value of "disabled". The value consists of one or more strings that define the method to release this CPU. If a client program recognizes any of the methods, it may use it. ...”

memory

```

memory {
    device_type = "memory";
    reg = <0x0 0x3b400000>;           // 948 MB
};

```

A memory device node is required for all devicetrees and describes the physical memory layout for the system. If a system has multiple ranges of memory, multiple memory nodes can be created, or the ranges can be specified in the reg property of a single memory node.

The unit-name component of the node name (see section 2.2.1) shall be memory.

The client program may access memory not covered by any memory reservations (see section 5.3) using any storage attributes it chooses. However, before changing the storage attributes used to access a real page, the client program is responsible for performing actions required by the architecture and implementation, possibly including flushing the real page from the caches. ...

device_type R <string> Value shall be “memory”

reg : R <prop-encoded-array> : Consists of an arbitrary number of address and size pairs that specify the physical address and size of the memory ranges.