

Requirements for Project Argus DC-SCM BMC Minimum Bootable Device Tree

This document provides the basis for the minimum bootable device tree (DT) to be used on Project Argus DC-SCM 2.0. The minimum bootable DT shall be selected if the DT for the DC-SCM vendor is not found during firmware initialization.

Aim

To enable the BMC boot in a partially functional mode.

The partially functional mode of the BMC should have the following:

- IPMI functionality
 - Host Power On
 - Host Power Off
 - Host Power Reset
- SOL
- KVM
- Redfish/bmcweb firmware update
- Fans (Running at 80%)

Requirements

Source DSTI

Shall include the device tree source for the BMC DC-SCM board (AST2600) i.e. `aspeed-g6.dtsi`.

Model

The model property of the DT shall indicate that the DT is for an unrecognized vendor, according to the following format,

```
model = "cf,DCSCM0xFF"
```

Platform type

Shall specify the platform architecture according to the dt specification - "manufacturer, model". Our DC-SCM platform is based on AST2600 architecture, and its compatibility property should follow that specified by the AST2600 platform,

```
compatible = "aspeed,ast2600"
```

Boot Console Output

Should specify the device to be used for boot console output with a stdout-path property under /chosen, as described in the [Devicetree Specification](#). It should specify earlyprintk to show kernel log messages and also route to a unified ttyS (across different vendors) with a baudrate of 115200 e.g.

```
chosen {
    stdout-path = "some-uart";
    Bootargs = "console=some-tty, 115200 earlyprintk";
};
```

Memory

The /memory node should be specified according to the [Devicetree Specification](#). e.g.

```
memory@80000000 {
    device_type = "memory";
    reg = <some-reg-addr addr-length->;
};
```

Reserved Memory

Should specify the sizes and appropriate compatibility property for the following subnodes: vga_memory, flash_memory, gfx_memory, video_engine_memory.

Firmware Memory Controller

Shall enable the firmware memory controller (FMC) and shall enable flash nodes necessary to read boot code from the FMC controller. Shall use the `openbmc-flash-layout` DT source.

UART

Should enable the necessary UART nodes for host and bmc console output.

KCS

Shall specify the KCS channels needed for uefi bios-ipmi controller interaction and host userspace to bmc interaction over ipmi. e.g.

```
&some-kcs {  
    // KCS for BIOS interaction  
    status = "okay";  
    aspeed,lpc-io-reg = <some reg>;  
};
```

GPIO

The pin convention shall be according to the aspeed-gpio bindings, and shall include reserved PINS to be used across vendors for `host power control`. e.g,

```
/*reserved for gpio setting*/  
&gpio {  
    status = "okay";  
    gpio-line-names =  
        /*some-array*/ "power-button","power-out", "", "", "", "", "", "",  
        /*some-array*/ "reset-button","reset-out", "", "", "", "", "", "",  
    ...  
};
```

HWMON

`iio-channels` for the operation of the `hwmon` to monitor the fans should be specified according to the [ADC-attached Hardware Sensor schema](#); e.g.

```
iio-hwmon {  
    compatible = "iio-hwmon";  
    io-channels = <&adc 1>, <&adc 2>;  
};
```

FAN TACH

Should enable `pwm\_tacho` subnode to support setting the fan speed. The `pwm\_tacho` should specify the default pin configuration node and the fans (as child nodes) available to the chassis. Each child node (i.e. fan) shall include the register for the PWM source port and the fan tach input channel. E.g.

```
&pwm_tacho {
    status = "okay";
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_pwm0_default &pinctrl_tach0_default>;

    someFan@0 {
        reg = <0x00>;
        aspeed,fan-tach-ch = /bits/ 8 <0x00>;
    };
};
```

FLASH

Flash definition to support ABR (dual boot mode for fail-safe).

E.g.:

```
&fmc {
    status = "okay";
    flash@0 {
        status = "okay";
        m25p,fast-read;
        label = "bmc";
        spi-max-frequency = <50000000>;
#include "openbmc-flash-layout-64.dtsi"
    };

    flash@1 {
        status = "okay";
        m25p,fast-read;
        label = "alt-bmc";
        spi-max-frequency = <50000000>;
#include "openbmc-flash-layout-64-alt.dtsi"
    };
};
```