



Novatek DTS User Guide

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.

With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

Table of Content

Novatek DTS User Guide.....	1
<i>Table of Content</i>	2
1 DTS 簡介	3
2 DTS 語法	3
2.1 DTS 中常用符號	4
2.2 DTS 中常用屬性	4
3 DTS 目錄結構	5
4 Sample Codes	7
4.1 Kernel Space	7
4.2 User Space	8

1 DTS 簡介

DTS 是 Device Tree Source 的縮寫，用來描述設備的硬件細節。通過 DTC (Device Tree Compiler)，可以將 DTS 文件轉化成 DTB (Device Tree binary file) 文件。



在系統啟動的過程中會將保存在外的 DTB 文件拷貝到內存中，然後將 DTB 的起始地址傳遞給內核。當需要對不同的平台實現支持時，只須替換相應 DTS 文件，不必改動內核。

DTB 也可以以檔案的形式存在 root-fs 裡，作為 configure file 供一般 application 透過 libfdt 來讀取使用。

2 DTS 語法

Device Tree 是一種節點 (node) 和屬性 (property) 的屬性結構，屬性由鍵值對 (name & value) 組成，節點中可以包含子節點和屬性。

Device Tree 有以下 5 個特點：

1. 每個設備樹文件都有一根節點，每一個設備都是一個節點
2. 節點間可以嵌套，形成父子關係，便於描述設備間的關係
3. 設備的屬性使用鍵值對 (key-value) 描述
4. 每個屬性的描述都以 ; 結束
5. 可自定義屬性

如下是一個 Device Tree 的基本範例：

● Example

```

/ { 根節點 (root node)
    cpus {  cpus 是節點名，是 / 的子節點
        #address-cells = <1>;
        #size-cells = <0>;
    }
}
  
```

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.

With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

```

cpu0: cpu@0 {
    device_type = "cpu";    device_type 是屬性名
    compatible = "arm,cortex-a7";
    reg = <0x0>;
    next-level-cache = <&L2>;
    clock-frequency = <799000000>;
};

};

};

```

2.1 DTS 中常用符號

符号	含义
/	根节点
@	如果设备有地址，则由此符号指定
&	引用节点
:	冒号前的 label 是为了方便引用给节点起的别名，此 label 一般使用为&label
"	引号中的为字符串，字符串数组："string1","string2","string3"
<>	尖括号中的为 32 位整形数字，整形数组<12 3 4>
[]	方括号中的为 32 位十六进制数，十六进制数据[0x11 0x12 0x13] 其中 0x 可省略
#	#并不表示注释。如 #address-cells ，#size-cells 用来决定 reg 属性的格式
,	属性名称中可以包含逗号

2.2 DTS 中常用屬性

1. Compatible

該屬性以 "<manufacturer>,<model>" 的形式使用，例如：compatible = "arm,cortex-a7"; 其包含製造商以避免命名空間衝突。其用於driver和device的匹配。

2. #xxx -cells

xxx 表示某個屬性，cell 表示一個 32 位無符號整數，xxx-cells 表示將使用多少個 cell

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.

With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

描述 xxx 屬性。在 reg 和 region 相關屬性介紹中會有相應的舉例。

3. reg

該屬性表示地址，#address-cells 表示有多少個 cell 用於表述地址，#size-cells 表示有多少個 cell 用於表示地址空間大小。舉例如下：

```
hdal-memory {
    #address-cells = <1>;
    #size-cells = <1>;
    media { reg = <0x12800000 0x0C800000>; };
    dsp2 { reg = <0x1F000000 0x00800000>; };
    dsp1 { reg = <0x1F800000 0x00800000>; };
};
```

- #address-cells = <1>：表示 reg 用一個 cell 表示地址，即 0x12800000 表示 hda1-memory 節點的起始地址。
- #size-cells = <1>：表示 reg 用一個 cell 表示地址空間的大小，即 0x0C800000 表示 hda1-memory 節點佔有的空間大小為 0x0C800000 字節。

3 DTS 目錄結構

這裡指的是 kernel space 裡給系統用的 dts，一個 .dts 文件對應於一個 machine，而 SOC 共用的部分或者多個 machine 共同的部分則被存放於 .dtsi 文件中，可以使用 include 將 .dtsi 文件包含進 .dts 文件。以 IPCam SDK 的 DTS 目錄為例：

```
/na51000_linux_sdk/configs
.
├── cfg_EMMC_EVB
│   ├── nvt-na51000-audio.dtsi
│   ├── nvt-na51000-basic.dtsi
│   ├── nvt-na51000-display.dtsi
│   ├── nvt-na51000-gpio.dtsi
│   ├── nvt-na51000-i2c.dtsi
│   ├── nvt-na51000-info.dtsi
│   ├── nvt-na51000-media.dtsi
│   ├── nvt-na51000-mem-tbl.dtsi
│   ├── nvt-na51000-nvtpack.dtsi
│   └── nvt-na51000-peri.dtsi
```

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.

With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

```

|   ├── nvt-na51000-smp-evb.dts
|   ├── nvt-na51000-storage-partition.dtsi
|   └── nvt-na51000-top.dtsi
└── cfg_IPCAM1_EVB
    ├── nvt-na51000-audio.dtsi
    ├── nvt-na51000-basic.dtsi
    ├── nvt-na51000-display.dtsi
    ├── nvt-na51000-gpio.dtsi
    ├── nvt-na51000-i2c.dtsi
    ├── nvt-na51000-info.dtsi
    ├── nvt-na51000-media.dtsi
    ├── nvt-na51000-mem-tbl.dtsi
    ├── nvt-na51000-nvtpack.dtsi
    ├── nvt-na51000-peri.dtsi
    ├── nvt-na51000-smp-evb.dts
    ├── nvt-na51000-storage-partition.dtsi
    └── nvt-na51000-top.dtsi
└── Makefile

```

在 `na51000_linux_sdk/` folder 使用如下命令即可生成 `nvt-na51000-smp-evb.dtb` 文件 (under `na51000_linux_sdk/output/`)。

```
make cfg
```

`nvt-na51000-smp-evb.dts` 文件只是將其它的 `.dtsi` 文件包含進來，其內容如下：

```

/*
 * Novatek Ltd. NA51000 BSP part of dts
 *
 * Cortex-A53 MPCore
 *
 */

/dts-v1/;
#include <dt-bindings/gpio/nvt-gpio.h>
#include "nvt-na51000-peri.dtsi" /* engine register definition */
#include "nvt-na51000-top.dtsi" /* pinmux, made from tool */
#include "nvt-na51000-i2c.dtsi" /* i2c */

```

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.

With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

```
#include "nvt-na51000-gpio.dtsi" /* gpio, made from tool */
#include "nvt-na51000-audio.dtsi" /* peripheral: audio */
#include "nvt-na51000-display.dtsi" /* peripheral: display */
#include "nvt-na51000-media.dtsi" /* ko files relation */
#include "nvt-na51000-mem-tbl.dtsi" /* memory partition */
#include "nvt-na51000-storage-partition.dtsi" /* flash partition */
#include "nvt-na51000-info.dtsi" /* used for turnkey information */
#include "nvt-na51000-nvtpack.dtsi" /* used for turnkey to make all-in-one bin */
```

4 Sample Codes

4.1 Kernel Space

如何在 kernel space 讀取系統用的 dtb, 使用的 APIs 是 linux-kernel/include/linux/of.h 而不是 libfdt, 以下為一般網路上都可找得到的範例:

```
static void access_dt(void)
{
    /* device node path - check it from /proc/device-tree/ */
    char *path = "/path/to/BB/child_2";
    struct device_node *dt_node;
    const u8 *prop = NULL;
    int ret;

    dt_node = of_find_node_by_path(path);
    if (!dt_node) {
        printk(KERN_ERR "Failed to find node by path: %s.\n");
    } else {
        printk(KERN_INFO "Found the node for %s.\n", path);
        prop = of_get_property(dt_node, "property 2", &ret);
        if (!prop) {
            //You are still in trouble!
        } else {
```

```
//You have got property 2 of BB!  
  
}  
  
}  
  
}
```

4.2 User Space

在 SDK 裡, *na51000_linux_code/sample/fdt/fdt_test.c* 展現的是如何讀取存在於 root-fs 中的 dtb, 與系統用的 dtb 無關.

```
int main(void)  
{  
    const char dtb_path[] = "/etc/application.dtb";  
    ...  
    const void *nodep; /* property node pointer */  
    struct fdt_header fdt_hdr = {0};  
  
    // load dtb to memory  
    fp = fopen(dtb_path, "rb");  
    ...  
    fread(&fdt_hdr, sizeof(fdt_hdr), 1, fp);  
    fseek(fp, 0, SEEK_SET);  
  
    int fdt_size = fdt_totalsize(&fdt_hdr);  
    buf = (unsigned char *)malloc(fdt_totalsize(&fdt_hdr));  
  
    fread(buf, fdt_size, 1, fp);  
    fclose(fp);  
  
    // read /sensor@1/cfg  
    int nodeoffset = fdt_path_offset(buf, "/sensor@1");  
    ...  
    nodep = fdt_getprop(buf, nodeoffset, "cfg", &len);  
    ...  
    const char *p_cfg = (const char *)nodep;  
    printf("cfg: ");  
    for (i = 0; i < len; i++) {  
        printf("%s ", p_cfg);  
        i += strlen(p_cfg);  
        p_cfg += strlen(p_cfg) + 1; // +1 for skip \0  
    }  
    printf("\n");  
  
    // read /sensor@1/sie  
    nodep = fdt_getprop(buf, nodeoffset, "sie", &len);  
    ...  
    int n = len / sizeof(unsigned int);
```

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.

With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.


```
unsigned int *p_sie = (unsigned int *)nodep;
printf("sie: ");
for (i = 0; i < n; i++) {
    printf("%u ", be32_to_cpu(p_sie[i]));
}
printf("\n");

free(buf);
return 0;
}
```