

# 开发者分享 | Petalinux 工程中设备树的介绍

judy 在 周五, 01/29/2021 - 10:19 提交

本文转载自: [XILINX开发者社区微信公众号](#)

本文来自XILINX嵌入式产品应用工程师 Terry Ni

设备树是 Petalinux kernel 的关键组件, 接下来以 2020.1 版本为例, 为大家介绍一下在Xilinx Petalinux 工程中的设备树是如何产生, 配置以及修改的。

## Petalinux 工程中设备树的位置

当我们创建了一个 petalinux 工程后 (以zcu102为例), 在编译的时候, 会自动生成对应的设备树文件。这些设备树的描述信息通常在以下2个工程路径下。

第一个是在 components/plnx\_workspace/device-tree/device-tree 路径下, 这个路径下主要保存的都是基于你导入的hdf/xsa后, 自动生成的设备树信息。通常包括PS的设备树, 时钟的设备树, PL 部分 IP 的设备树 (Xilinx发布的包含驱动的IP), 以及设备树头文件。

```
./components/plnx_workspace/device-tree/device-tree/zynqmp-clk-ccf.dtsi
```

```
./components/plnx_workspace/device-tree/device-tree/zcu102-revc.dtsi
```

```
./components/plnx_workspace/device-tree/device-tree/pl.dtsi
```

```
./components/plnx_workspace/device-tree/device-tree/system-top.dts
```

```
./components/plnx_workspace/device-tree/device-tree/zynqmp.dtsi
```

```
./components/plnx_workspace/device-tree/device-tree/system-conf.dtsi
```

```
./components/plnx_workspace/device-tree/device-tree/pcw.dtsi
```

另一个是在 project-spec/meta-user/recipes-bsp/device-tree/files路径下, 这个路径下主要就是用户基于自动生成的设备树, 自行添加和修改的部分。这部分可以查看 UG1144 第10章 Device Tree Configuration 节的内容。

```
./project-spec/meta-user/recipes-bsp/device-tree/files/openamp.dtsi
```

```
./project-spec/meta-user/recipes-bsp/device-tree/files/system-user.dtsi
```

```
./project-spec/meta-user/recipes-bsp/device-tree/files/pl-custom.dtsi
```

```
./project-spec/meta-user/recipes-bsp/device-tree/files/xen-qemu.dtsi
```

```
./project-spec/meta-user/recipes-bsp/device-tree/files/xen.dtsi
```

需要注意的是, 如果 petalinux-config->DTGsettings->MACHINE\_NAME 采用的是xilinx开发板预设的名称的话 (UG1144DTG Settings 章节中的 Machine name 的说明), 那么会自动生成对应开发板相关的设备树信息。

### Machine Name

For custom boards it is template. For Xilinx® evaluation boards the following values are supported:

ac701-full, ac701-lite, kc705-full, kcu105, zcu1275-revb, zcu1285-reva, zc1751-dc1, zc1751-dc2, zc702, zc706, avnet-ultra96-rev1, zcu100-revc, zcu102-rev1.0, zcu104-revc, zcu106-reva, zcu111-reva, zedboard, vcu118-rev2.0, sp701-rev1.0, zcu216-reva, zcu208-reva.

所以如果使用的是自己的硬件平台，那么尽量不要使用预设的名称，这样可以避免导入多余的设备树信息从而导致编译失败或者最终功能不正确。

## 如何修改设备树

### 1. 基于自动生成的设备树，添加和修改节点。

通过修改./project-spec/meta-user/recipes-bsp/device-tree/files/system-user.dtsi，来实现对之前已经定义的设备树节点中的属性进行变更，也可以添加没有自动生成的设备树节点。

```
/dts-v1/;

/include/ "system-conf.dtsi"

/ {

};

&gem0 {

    phy-handle = <&phy0>;

    ps7_ethernet_0_mdio: mdio {

        phy0: phy@7 {

            compatible = "marvell,88e1116r";

            device_type = "ethernet-phy";

            reg = <7>;

        };

    };

};

gpio-leds {

    compatible = "gpio-leds";

    led-ds23 {

        label = "led-ds23";

        gpios = <&ps7_gpio_0 10 0>;

        default-state = "on";

        linux,default-trigger = "heartbeat";

    };

};
```

有时候为了方便设备树的管理，可以定义多个设备树文件，然后在 `system-user.dtsi` 中将这些设备树文件都包含进去，如下所示。

```
/include/ "system-conf.dtsi"

/include/ "system-user-1.dtsi"

/include/ "system-user-2.dtsi"

/include/ "system-user-3.dtsi"

/ {

};
```

当添加了其他的dtsi文件后，还需要修改`./project-spec/meta-user/recipes-bsp/device-tree/device-tree.bbappend` 文件，将这些 dtsi 文件的路径信息包含进去。如下所示。

```
SRC_URI += "file://system-user.dtsi"

SRC_URI += "file://system-user-1.dtsi"

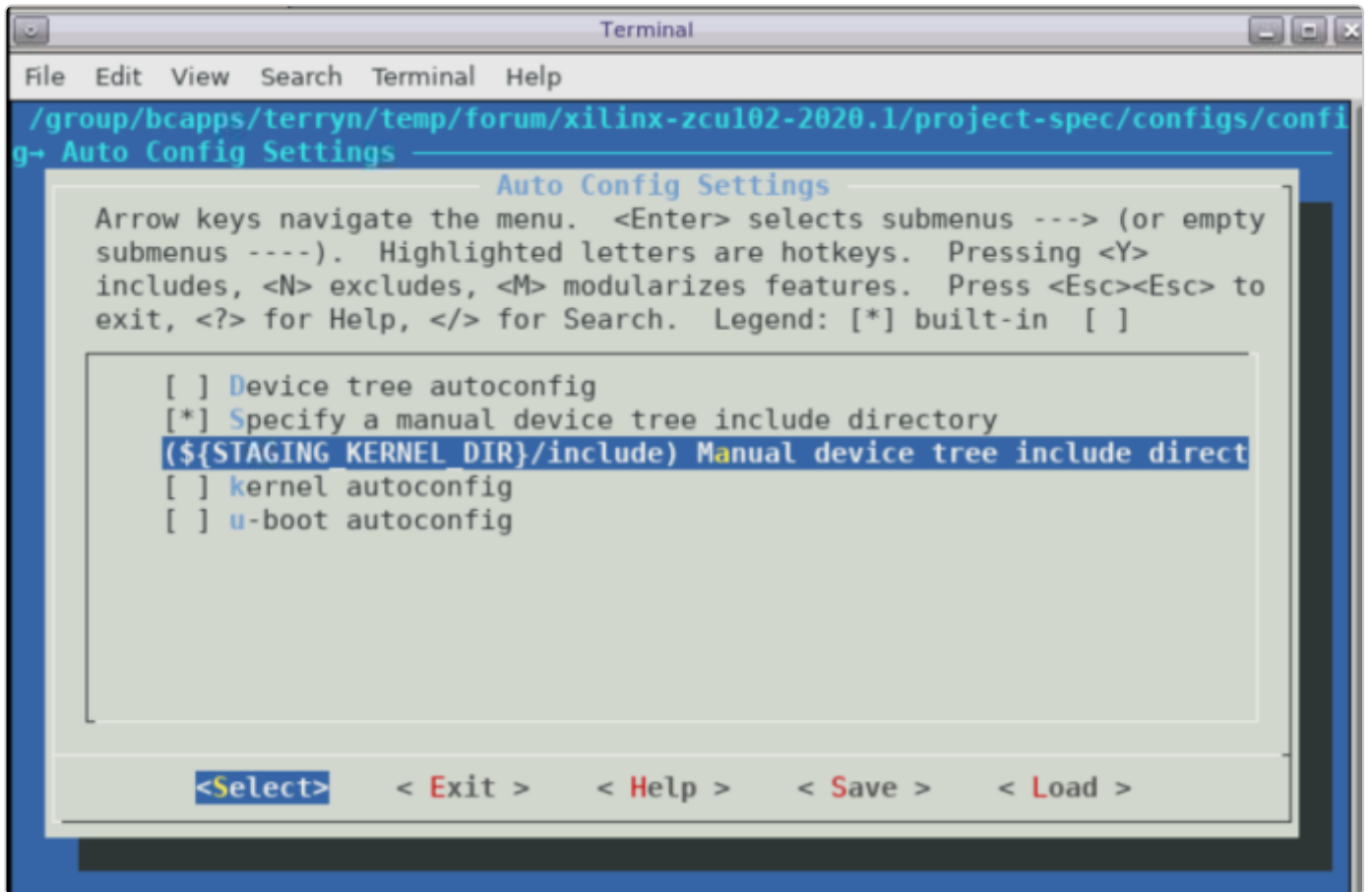
SRC_URI += "file://system-user-2.dtsi"

SRC_URI += "file://system-user-3.dtsi"
```

## 2. 完全使用自己的设备树

我们还可以更改 `petalinux-config-> Auto Config Settings` 中的配置，直接使用自己编写好的设备树。

取消 `Devicetree autoconfig`，使能 `Specify a manual device tree include directory`，然后指定设备树的路径地址。



## 怎么描述设备树

设备树中描述的信息都是要与 kernel 中设备的驱动互相对应的。如果驱动中需要某个参数，但是设备树中并没有包含，很可能就会编译错误或者直接使用了默认值，从而导致工作异常。kernel 的设备驱动中，一般都会包含设备树的描述说明来保证设备树的正确。你可以在 git 上的 kernel 源码中找到响应的设备树说明。以 GPIO 驱动为例。你可以找到相关的设备树描述信息如下。

[https://github.com/Xilinx/linux-xlnx/blob/xlnx\\_rebase\\_v5.4\\_2020.1/Docume...](https://github.com/Xilinx/linux-xlnx/blob/xlnx_rebase_v5.4_2020.1/Docume...)

Example:

```
gpio@e000a000 {
    #gpio-cells = <2>;

    compatible = "xlnx,zynq-gpio-1.0";

    clocks = <&clk 42>;

    gpio-controller;

    interrupt-parent = <&intc>;

    interrupts = <0 20 4>;

    interrupt-controller;

    #interrupt-cells = <2>;

    reg = <0xe000a000 0x1000>;

};
```

另外，你可以查看 xilinx wiki 上 linux driver 的主题页面来获取所有 Xilinx 支持的设备驱动内容，其中包含有各个设备树的例子以及相关配置等说明。

<https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/18841873/Linux+Dri...>

## 反编译设备树

为了最终确认设备树是否正确，我们可以通过 dtc 这个小应用，将 dtb 文件反编译成 dts 设备树文件。这个设备树文件，从 system-top.dts 这个 top 文件开始，把所有包含的设备树内容都整合到了一个 dts 文件中，便于我们确认最终的设备树是否正确。其命令格式如下

```
dtc -I dtb -O dts -o system.dts system.dtb
```