ACPI And Device Trees – Friends Or Foes?

Rafael J. Wysocki

Intel Open Source Technology Center

October 15, 2014
Outline

1. Introduction
   - Origins of ACPI
   - Motivation for Device Trees

2. ACPI vs Device Trees
   - Differences and Analogies
   - Putting It All Together

3. Matching Devices to Drivers
   - Existing Driver Lookup Algorithms
   - Driver Lookup Unification

4. Resources
PC Monoculture (Hardware Compatibility Everywhere)
Differences Appear (Slightly Incompatible Hardware)
Substantial Differences (Not Really Compatible Any More)
Solution: Platform Description Protocol
**ACPI: Advanced Configuration and Power Interface**

Idea (on top of the platform description)

High-level interface for driving platform hardware.
ACPI Complexity

ASL (ACPI Source Language)

AML (ACPI Machine Language)

ACPI Tables

ACPI Definition Blocks

ACPI Namespace

ACPI Objects

Device Description Data

Flat Tables
ARM: Same Architecture, Different Platforms
Solution: Platform Description Protocol
Flattened Device Tree

/ {

  node1 {
    a-string-property = "A string";
    a-string-list-property = "first string", "second string";
    a-cell-property = <1 2 3 4>; /* each number (cell) is a uint32 */
    child-node1 {
      first-child-property;
      second-child-property = <1>;
      a-string-property = "Hello, world";
    };
    child-node2 {
    };
  };

  node2 {
    an-empty-property;
    a-byte-data-property = [0x01 0x23 0x34 0x56];
    child-node1 {
    };
  };
}
ACPI and DT Side-by-Side

ACPI vs Device Trees

October 15, 2014 12 / 22
The Missing Link

_DSD (Device Specific Data) introduced in ACPI 5.1

```
Name (_DSD, Package () {
    ToUUID("<UUID string>"), // Format identifier
    Package () {
        ... // Device data in the given format
    }
}
```

Device Properties UUID: daffd814-6eba-4d8c-8a91-bc9bbf4aa301

```
Name (_DSD, Package () {
    ToUUID("daffd814-6eba-4d8c-8a91-bc9bbf4aa301"),
    Package () {
        Package {"a-string-property", "A string"},
        Package {"a-string-list-property", Package {"first string", "second string"}};
        Package {"a-cell-property", Package {1, 2, 3, 4}};
    }
}
```

Unified Device Properties API

```c
- if (of_property_read_u32(np, "size", &val) == 0 ||
-     of_property_read_u32(np, "at25,byte-len", &val) == 0) {
+ if (device_property_read_u32(dev, "size", &val) == 0 ||
+     device_property_read_u32(dev, "at25,byte-len", &val) == 0) {
     chip->byte_len = val;
   } else {
     dev_err(dev, "Error: missing \"size\" property\n");
     return -ENODEV;
   }

- if (of_property_read_u32(np, "pagesize", &val) == 0 ||
-     of_property_read_u32(np, "at25,page-size", &val) == 0) {
+ if (device_property_read_u32(dev, "pagesize", &val) == 0 ||
+     device_property_read_u32(dev, "at25,page-size", &val) == 0) {
     chip->page_size = (u16)val;
   } else {
     dev_err(dev, "Error: missing \"pagesize\" property\n");
     return -ENODEV;
   }

- if (of_property_read_u32(np, "at25,addr-mode", &val) == 0) {
+ if (device_property_read_u32(dev, "at25,addr-mode", &val) == 0) {
     chip->flags = (u16)val;
   } else {
```
Driver Lookup the DT Way

Device 1
compatible = "name1", "name2"
.of_match_table{.compatible = "name1"}

Driver 1
.of_match_table{.compatible = "name2"}

Device 2
compatible = "nameA", "nameB", "nameC"

Driver 2
.of_match_table{.compatible = "nameC"}

Driver 3
.of_match_table{.compatible = "nameC"}
Driver Lookup the ACPI Way

Device 1
_HID("ID 1")
_CID("ID 2", "ID 3", "ID 4")

Driver 1
.acpi_match_table{"ID 1"}

Device 2
_HID("ID A")
_CID("ID B", "ID C", "ID D")

Driver 2
.acpi_match_table{"ID 2", "ID C"}
ACPI/PNP Device IDs Problem

leds-gpio driver

```c
static const struct of_device_id of_gpio_leds_match[] = {
    { .compatible = "gpio-leds", },
    { },
};
```

gpio_keys_polled driver

```c
static const struct of_device_id gpio_keys_polled_of_match[] = {
    { .compatible = "gpio-keys-polled", },
    { },
};
```

Question
What ACPI/PNP IDs to use with these drivers?
Drivers Using the Uniform Device Properties API

**Device 1**
- `compatible = "name1", "name2"
- `_HID("PRP0001")`

**Driver 1**
- `.of_match_table{.compatible = "name1"}`

**Device 2**
- `_CID("ID 1", "ID 3")`
- `_HID("ID 2")`

**Driver 2**
- `.acpi_match_table{"ID 2"}`
- `.of_match_table{.compatible = "name2"}`

**Device 3**
- `compatible = "nameA", "nameB", "nameC"
- `_HID("PRP0001")`

**Driver 3**
- `.of_match_table{.compatible = "nameC"}`
References


Documentation And Source Code

- http://git.kernel.org/cgit/linux/kernel/git/rafael/linux-pm.git/log/?h=device-properties
- Documentation/devicetree/
- Documentation/acpi/
- drivers/acpi/
- drivers/of/
Thanks!

Thank you for attention!