

**NAME**

**ofw\_bus\_is\_compatible**, **ofw\_bus\_is\_compatible\_strict**, **ofw\_bus\_node\_is\_compatible**,  
**ofw\_bus\_search\_compatible** - check device tree nodes for compatibility with drivers

**SYNOPSIS**

```
#include <dev/ofw/openfirm.h>
#include <dev/ofw/ofw_bus.h>
#include <dev/ofw/ofw_bus_subr.h>
```

*int*

```
ofw_bus_is_compatible(device_t dev, const char *compatstr);
```

*int*

```
ofw_bus_is_compatible_strict(device_t dev, const char *compatstr);
```

*int*

```
ofw_bus_node_is_compatible(phandle_t node, const char *compatstr);
```

*const struct ofw\_compat\_data \**

```
ofw_bus_search_compatible(device_t dev, const struct ofw_compat_data *compat);
```

**DESCRIPTION**

The "compatible" property of the device tree node is used to identify the type of the device the node represents. The property is a list of one or more strings that represent hardware types the device is compatible with. The common format for such strings is "vendor,hardware" where "vendor" is an abbreviated name of the manufacturer and "hardware" is a device identifier, for instance, "fsl" for "Freescale" and "imx6ul-i2c" for the I2C controller. More than one string is required for compatibility with older revisions of the driver. If hardware revision B is backward compatible with revision A device tree node can signal this compatibility by providing both "vndr,hrdwrA" and "vndr,hrdwrB" strings in the "compatible" property value. This way older driver can use features available only in revision A, and the new version of the driver can take advantage of revision B feature set.

**ofw\_bus\_is\_compatible()** returns 1 if the *compatstr* value occurs in the "compatible" property list of the device tree node associated with the device *dev*, and 0 otherwise.

**ofw\_bus\_is\_compatible\_strict()** return 1 if the "compatible" property of the device tree node associated with the device *dev* consists of only one string and this string is equal to *compatstr*, and 0 otherwise.

**ofw\_bus\_node\_is\_compatible()** returns 1 if the *compatstr* value occurs in the "compatible" property list of the device tree node *node*, and 0 otherwise.

**ofw\_bus\_search\_compatible()** returns pointer to the first entry of the *compat* table whose *ocd\_str* field occurs in "compatible" property of the device tree node associated with the device *dev*. The *compat* table is an array of struct *ofw\_compat\_data* elements defined as follows:

```
struct ofw_compat_data {
    const char *ocd_str;
    uintptr_t ocd_data;
};
```

The *compat* table must be terminated by the entry with *ocd\_str* set to NULL. If the device tree node is not compatible with any of the entries, the function returns the pointer to the terminating entry.

### EXAMPLES

```
static struct ofw_compat_data compat_data[] = {
    {"arm,hrdwrA",          FEATURE_A},
    {"arm,hrdwrB",          FEATURE_A | FEATURE_B},
    {NULL,                  0}
};
```

```
static int
hrdwr_probe(device_t dev)
{
    ...
    if (!ofw_bus_search_compatible(dev, compat_data)->ocd_data)
        return (ENXIO);
    ...
}
```

```
static int
hrdwr_attach(device_t dev)
{
    ...
    sc = device_get_softc(dev);
    sc->sc_features = ofw_bus_search_compatible(dev, compat_data)->ocd_data;
    ...
}
```

### SEE ALSO

ofw\_bus\_find\_compatible(9)

### AUTHORS

This manual page was written by Oleksandr Tymoshenko.