Around revision [900], MutekH got support for Flattened device trees.

Rationale

Flattened device trees (FDT) are useful to get a list of all available hardware where no hardware-based enumeration exists (aka PnP, like PCI, USB, ... provides).

FDT provides normalized representation of the hardware platform without adding specific initialization code.

The normalization comes from IEEE1275 (aka Open Firmware); while Open Firmware also defined heavy things (like a Forth interpreter), we only use the FDT information.

Using FDT is mainstream, andis also used by Linux and BSD, supported is U-Boot and other bootloaders.

Implementation

In MutekH, FDT is handled through an hardware enumerator device driver, it behaves like the other enumerators (PCI, ISAPnP).

Drivers may export themselves as FDT-aware, and define which device name string to match. For instance, the following subtree defines a tty device:

```
tty@0 {
    device_type = "soclib:tty";
        tty_count = <1>;
        reg = <0x90600000 0x10>;
        icudev = &{/icu@0};
        irq = <1>;
};
```

In turn, the SoCLib tty driver declares itself (in source:trunk/mutekh/drivers/device/char/tty-soclib/tty-soclib.c#L146) as:

```
static const struct devenum_ident_s     tty_soclib_ids[] =
{
        DEVENUM_FDTNAME_ENTRY("soclib:tty", 0, 0),
        { 0 }
};

const struct driver_s   tty_soclib_drv =
{
  .class                = device_class_char,
  .id_table             = tty_soclib_ids,
  .f_init               = tty_soclib_init,
  .f_cleanup            = tty_soclib_cleanup,
  .f_irq                = tty_soclib_irq,
  .f.chr = {
    .f_request          = tty_soclib_request,
  }
};
```

This will make the FDT enumerator use the correct driver, matching `"soclib:tty"`