Long Jumping Linux 2.6...5.10

Parthiban Nallathambi

Linumiz
Parthiban Nallathambi, Linumiz

- Embedded Linux and Zephyr RTOS development, consulting, training
- Embedded Linux development: BSP, u-boot, Linux Kernel, Yocto Project, Buildroot
- Zephyr: SoC, Board support, drivers

www.linumiz.com

Living in Berlin, Germany
Agenda

- Why to migrate?
- What can be upgraded?
- How to upgrade?
- Key factors to consider
- Build system
Assumptions

➢ All the IP blocks are supported in U-Boot, Linux Kernel

➢ No coding required or minor changes

➢ Custom Kernel out-of-tree drivers and migration aren't considered

➢ Knowledge about devicetree is assumed and isn't discussed extensively
Why to migrate?
New technology and hardware
New technology and hardware

## Product Lead Time Estimates

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Ship Date Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>4,000</td>
<td>3/11/2059</td>
</tr>
</tbody>
</table>

Ship dates are approximate and subject to change.
Chip and Hardware shortage

- Newer SoC/SoM/Board aren't available
- Peripheral obsolete (e.g. NAND, NOR flashes)
- Migrate hardware based on existing/available SoC's
Are we on the same page?

- Is this migration/upgrade valid for my product portfolio?
- How long it will take? Timeline?
Product lifecycle
Modified product lifecycle

- Gen3: Production from 2005 to 2025
- Gen3+: Production from 2010 to 2030
- Gen4: Development from 2015 to 2025, Production from 2025 to 2035

Timeline:
- 2005
- 2010
- 2015
- 2020
- 2025
- 2030
- 2035
Customer cases

- Retain new (Gen4) code base of middleware and application software
- Obsolete connectivity technology (e.g. 3G shutdown)
- Time to market – switching to new family vs retain the same family SoC
  - Cost – Development, Certification
  - Production timeline
How to upgrade?
What can be upgraded? (Gen3 ➔ Gen3+)

- Linux Kernel
- Root filesystem
- Over the Air
  - New network component
  - Larger memory
- Bootloader
  - U-boot
Steps to migrate

➢ One step at a time

➢ Add basic blocks (Clock, Timer, UART) and make it bootable

➢ Enable only needed features in Kconfig

➢ Consider out-of-tree modules

➢ Enable security
Kernel

➢ Platform identification

➢ Create devicetree for the board

✓ Less code duplication across platforms

✓ Memory configuration

✓ Device population

➢ No/very less custom changes when devicetree is used

➢ Migrate defconfig for the board
Moving from **MACHINE_START** to **setup_machine_fdt()** to find the **machine_desc[]**
/*
 * Set of macros to define architecture features.
 * This is built into a table by the linker.
 */
#define MACHINE_START(_type, _name) \
static const struct machine_desc __mach_desc_##_type \
__used \
__attribute__((__section__(".arch.info.init"))) = { \
    .name = _name,
}
#define MACHINE_END

MACHINE_START(AT91SAM9260_CTEK, "Atmel AT91SAM9260-CTEK")
/* Maintainer: Atmel */
    .init_time = at91sam926x_pit_init,
    .map_io = at91_map_io,
    .handle_irq = at91_aic_handle_irq,
    .init_early = ek_init_early,
    .init_irq = at91_init_irq_default,
    .init_machine = ek_board_init,
MACHINE_END
Devicetree for board

`+/dts-v1/;
+    #include "at91sam9260.dtsi"
+    #include <dt-bindings/input/input.h>
+    +
+    +/ {
+    +    model = "CTEK G6200";
+    +    compatible = "atmel,at91sam9260ctek", "atmel,at91sam9260", "atmel,at91sam9";
+    +`
UART for custom peripheral

```c
static struct platform_device serial_device = {
    .name = "serial8250",
    .id = 0,
    .dev = {
        .platform_data = serial_platform_data,
    },
};
```

/* USART0 on ttyS1. (Rx, Tx, CTS, RTS, DTR, DSR, DCD, RI) */
at91_register_uart(AT91SAM9260_ID_US0, 1, ATMEL_UART_CTS | ATMEL_UART_RTS |
                   ATMEL_UART_DTR | ATMEL_UART_DSR | ATMEL_UART_DCD |
                   ATMEL_UART_RI);

/* USART1 on ttyS2. (Rx, Tx, RTS, CTS) */
at91_register_uart(AT91SAM9260_ID_US1, 2, ATMEL_UART_CTS | ATMEL_UART_RTS);
at91_add_device_serial();
mcu: serial@fffb0000 {
    compatible = "atmel,at91sam9260-usart";
    reg = <0xffffb0000 0x200>
    interrupts = <6 IRQ_TYPE_LEVEL_HIGH 5>; 
    atmel,use-dma-rx;
    atmel,use-dma-tx;
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_usart0>;
    clocks = <&pmc PMC_TYPE_PERIPHERAL 6>;
    clock-names = "uart";
    status = "okay";
};
From 8dd19e8cc050c37fe105052424e46e4a23423057 Mon Sep 17 00:00:00 2001
From: Parthiban Nallathambi <parthiban@linumiz.com>
Date: Sat, 7 May 2022 13:26:30 +0000
Subject: [PATCH] arm: boot: add device tree for ctek g6200

add initial device tree for ctek g6200 based on at91sam9260
evaluation kit

Signed-off-by: Parthiban Nallathambi <parthiban@linumiz.com>
---
arch/arm/boot/dts/Makefile | 1 +
arch/arm/boot/dts/at91sam9260_g6200.dts | 253 +++++++++++++++++++++++++++++++
2 files changed, 254 insertions(+)
create mode 100644 arch/arm/boot/dts/at91sam9260_g6200.dts
diff --git a/arch/arm/boot/dts/Makefile b/arch/arm/boot/dts/Makefile
index 7e8151681597..9ea40b75cf2e 100644
--- a/arch/arm/boot/dts/Makefile
+++ b/arch/arm/boot/dts/Makefile
@@ -18,6 +18,7 @@ dtb-$(CONFIG_SOC_AT91SAM9) += \
tny_a9260.db 
     usb_a9260.db 
     at91sam9260ek.db 
+    at91sam9260_g6200.db 
     at91sam9261ek.db 
     at91sam9263ek.db 
     at91-sam9_19260.db 

I2C: Userspace driver to IIO

➢ Removed / Dropped I2C custom driver in userspace

➢ Moved to Industrial IO

```c
&i2c0 {
    #address-cells = <1>;
    #size-cells = <0>;

    light-sensor@60 {
        compatible = "vishay,vcnl4035";
        reg = <0x60>;
        interrupt-parent = &gpio4;
        interrupts = <11 IRQ_TYPE_LEVEL_LOW>;
    }
};
```
defconfig migration

- Review all the features in 5.10 SLTS
- Add basics and append (only) needed features
- Migration step by step
- Storage space limitation and Kernel size
  - Strip & Compression - lzo
U-Boot

➢ Porting is very simple, If you have,
  ✓ SoC support
  ✓ Support for all IP blocks
  ✓ Board based on this SoC

➢ If not,
  ✓ Custom power sequence
  ✓ Custom drive handling

➢ Migration consideration
  ✓ Kconfig – from header files
  ✓ Driver model
```c
int board_late_init(void)
{
    u32 type = get_board_type();

    switch (type) {
        case BOARD_A:
            env_set("bootconf", "conf@board-a.dtb");
            break;
        case BOARD_B:
            env_set("bootconf", "conf@board-b.dtb");
            break;
        default:
            printf("Unknown board %d\n", type);
            break;
    }

    return 0;
}
```

`bootm ${fit_addr}#${bootconf}`
U-Boot: Kernel image type

- bootz vs bootm / zImage vs uImage vs fitImage support
- Loading kernel from different storage medium (with different filesystem)
Key factors
Unexpected deviation

➢ Software time deviation

➢ Change of hardware components e.g. NAND

➢ Porting custom drivers

➢ Corner cases (Panics and OOPS)
Software time deviation

- Boot time impact
- initramfs

✓ compiling most kernel parts are modules

```
setenv bootargs console=ttyS0,115200 rootfstype=ramfs root=/dev/ram0
```
New partition from vendor

```
nand read ${loadaddr} ${new_offset}
setenv bootargs console=ttyS0,115200 ubi.mtd=5 root=ubi:root${ubibootvol}
mtdparts=atmel_nand:10K(bootstrap),512K(modem),512K(U-boot),64K(config),2M(Kernel),128M(RFS),- (Data) rw rootfstype=ubifs
```
NAND Page size

 Partition size adjustment

- 5 Pages (2K x 5)
  - Modem memory (512K)
  - First stage (10K)

- 3 Pages (4K x 3)
  - Modem memory (512K)
  - 2K
  - First stage (10K)
Porting custom driver

- Migration effort for compiler compatibility
- Porting to newer API and ABI
  - Memory reservation for error logging – moved to devicetree without any custom memory reservation in the kernel/driver
  - Platform device to device tree
Build system
Yocto project: machine conf

Ref: https://docs.zephyrproject.org/latest/_images/hierarchy.png
Cont.,

```plaintext
# @TYPE: Machine
# @Name: CTEK AT91SAM9260 G6200 NAND
# @DESCRIPTION: Machine configuration for CTEK AT91SAM9260 G6200 board
require include/sam9x60.inc

DISTRO_FEATURES_DEFAULT = ""
DISTRO_FEATURES = "ipv4"
DISTRO_FEATURES:remove = "pulseaudio x11 alsa"
BAD_RECOMMENDATIONS += "udev-hwdb"

DEFAULTTUNE ?= "armv5te"
require conf/machine/include/arm/arch-armv5-dsp.inc

TUNEVALID[arm926ejs] = "Enable arm926ejs specific processor optimizations"
TUNE_CCARGS .= "${@.utils.contains('TUNE_FEATURES', 'arm926ejs', \\n  '-mcpu=arm926ejs', '', d)}"

MACHINEOVERRIDES = "${@.utils.contains('TUNE_FEATURES', 'arm926ejs', \\n  'armv5:', '', d)}"
AVAILTUNES += "arm926ejs"
ARMPKGARCH:tune-arm926ejs = "arm926ejs"

# Atmel AT91SAM9 defaults
require conf/machine/include/soc-family.inc
require conf/machine/include/arm/armv5\tune-arm926ejs.inc

SOC_FAMILY = "at91sam9"
SERIAL_CONSOLES ?= "115200;ttyS0"
```
Benefits

- Benefited with new features (migration)
  - Power Management: pm_runtime
  - Distributed Switch Architecture (DSA) with devicetree based device model
  - Virtualization: Enabled the way to use Embedded Containers like Pantcor/Pantavisor
  - Over the Air using SWUpdate and RAUC (with Hawkbit)
  - USB: GadgetFS to ConfigFS with libusbg
  - squashfs and lifetime of NAND, NOR storage
  - Overcome 2.6.x CVE's

[2] Porting U-Boot and Linux on New ARM Boards -
https://www.youtube.com/watch?v=5EOsdYkvq-Q
Questions?

Parthiban Nallathambi
parthiban@linumiz.com