Application Notes

UART User Guide
Revision 0.1

Amlogic, Inc.
3930 Freedom Circle
Santa Clara, CA 95054
U.S.A.
www.amlogic.com

Table of Contents

Revision history

Revision	Date	Owner	Changes
0.1	March 11, 2013	Jiamin Miao	Draft

1. Overview

This document tells users how to config Amlogic UART port.

2. Software Operation

2.1 Add aml_uart_device

We should:

- add a platform_device "aml_uart_device" (defined as below) in BSP file:

```
static struct platform_device aml_uart_device = {
   .name       = "mesonuart",
   .id     = -1,
   .num_resources  = 0,
   .resource   = NULL,
   .dev = {
      .platform_data = &aml_uart_plat,
   },
};
```

- add the platform_device "aml_uart_device" into the platform_device array "platform_devs".

2.2 Declare aml_uart_platform structure

The member variable "platform" of aml_uart_device points to the structure variable "aml_uart_plat" (defined as below):

```
static struct aml_uart_platform  __initdata aml_uart_plat = {
   .uart_line[0]   = UART_AO,
   .uart_line[1]   = UART_A,
   .uart_line[2]   = UART_B,
   .uart_line[3]   = UART_C,
   .uart_line[4]   = UART_D,
   .pinmux_uart[0] = (void*)&aml_uart_ao,
   .pinmux_uart[1] = (void*)&aml_uart_a,
   .pinmux_uart[2] = NULL,
   .pinmux_uart[3] = NULL,
   .pinmux_uart[4] = NULL
};
```

<Note>:

- "uart_line" is the macro name of our UART port.
- "pinmux_uart" points to the pinmux_set of our UART port .
- We should validate the pinmux_set to activate an UART.
  For example, if we want to use UART A, we should define its pinmux_set "aml_uart_a" pointed from pinmux_uart[1].
- We can activate several UART as requested.
  For example, we can activate UART AO + UART A, or UART A + UART B, or all those 5 UART, simply by validating the according pinmux_set.

3. Sample

3.1 The flow of adding UART A

- Firstly, since RTS, CTS, RX and TX of UART A are bit[13:10] of pinmux reg4, so we have the below defination (the bold part):

```
static pinmux_item_t uart_pins[] = {
    {
        .reg = PINMUX_REG(AO),
        .setmask = 3 << 11
    },
    {
        .reg = PINMUX_REG(4),
        .setmask = 0xf << 10
    },
    PINMUX_END_ITEM
};
```

- Secondly, we define "aml_uart_a" pointing to uart_pins[1]:

```
static pinmux_set_t aml_uart_a = {
    .chip_select = NULL,
    .pinmux = &uart_pins[1]
};
```

- Finally, we have the pinmux_uart[1] of "aml_uart_plat" point to "aml_uart_a" (the bold part):

```
static struct aml_uart_platform  __initdata aml_uart_plat = {
    .uart_line[0]   = UART_AO,
    .uart_line[1]   = UART_A,
    .uart_line[2]   = UART_B,
    .uart_line[3]   = UART_C,
    .uart_line[4]   = UART_D,

    .pinmux_uart[0] = (void*)&aml_uart_ao,
    .pinmux_uart[1] = (void*)&aml_uart_a,
    .pinmux_uart[2] = NULL,
    .pinmux_uart[3] = NULL,
    .pinmux_uart[4] = NULL
};
```