

# Open BMC Development Series (10) Configuring the serial port to output BIOS information

原创

dayuli

Posted on 2021-12-20 11:49:23

Read 4.2k

Collection 14

Likes 5

Category Column: 

BMC

Article Tags: 

Microcontroller

stm32

FPGA Development

bmc

Copyright CC 4.0 BY-SA



BMC This column includes this content

24 articles

Subscribe to our column



This article records the process of solving the problem that the openBMC serial port cannot output BIOS boot information. Through in-depth analysis of the UART controller on the BMC side, the correct device tree configuration was found and the serial port communication failure was solved.

The summary is generated in [C Know](#) , supported by DeepSeek-R1 full version, [go to experience](#)>

I have been working on the serial port problem of open BMC recently. It took me a long time, mainly because I am not familiar with it.

Problem description:

We use Inspur's BMC, and the serial port of the corresponding board can output the BIOS startup process normally. However, when using my branch's BMC, the serial port cannot output the corresponding BIOS startup information.

Analysis process:

From the question, we can see that the problem lies on the BMC side, and the BIOS has completed the redirection settings.

The serial port component on the BMC is UART , and the corresponding serial port controller is described as follows:

## 39 UART Controller (16550)

### 39.1 Overview

AST2500 integrates 5 sets of UART (Universal Asynchronous Receiver/Transmitter) providing serial communication capabilities with other external devices, like another computer using a serial cable based on RS232 protocol. This core is designed to be compatible with the industry standard — 16550 UART. The two sets of UART are equipped with a 16x8 FIFO that can be programmed to be enabled or disabled. The supported baud Rates are also programmable.

Each unit of UART totally implements 12 sets of 32-bit registers, which are listed below, to program the various supported functions including character length selection, baud rate selection, interrupt generation, and parity generation/checking. Each register has its own specific offset value, ranging from 0x00 to 0x14h, to derive its physical address location.

Base Address of UART1 = 0x1E78 3000

Base Address of UART2 = 0x1E78 D000

Base Address of UART3 = 0x1E78 E000

Base Address of UART4 = 0x1E78 F000

Base Address of UART5 = 0x1E78 4000

Register Address of UART = (Base Address of UART) + Offset

UART RBR : Receiving Buffer Register (DLAB = 0)

UART THR : Transmit Holding Register (DLAB = 0)  
 UART IER : Interrupt Enable Register (DLAB = 0)  
 UART IIR : Interrupt Identity Register  
 UART FCR : FIFO Control Register  
 UART LCR : Line Control Register  
 UART MCR : Modem Control Register  
 UART LSR : Line Status Register  
 UART MSR : Modem Status Register  
 UART SCR : Scratch Register  
 UART DLL : Divisor Latch Low Register: (DLAB = 1)  
 UART DLH : Divisor Latch High Register: (DLAB = 1)

### 39.2 Features

- Directly connected to APB bus

*ASPEED Confidential*

*All rights reserved.*

631

May 12, 2017

### ASPEED

**Confidential** ASPEED AST2500/AST2520 A2 Datasheet – V1.6

- Support two UART with full flow control pins (one is with dedicated flow control pins, the other is shared with GPIO pins)
- Separate transmit & receive FIFO buffer (16x8) to reduce CPU interrupts
- Support up to 115.2K baud-rate
- Programmable baud rate generator
- Standard asynchronous communication bits — Stat/Stop/Parity
- Independent masking of transmit FIFO, receive FIFO, receiving timeout and error condition Interrupts
- False start-bit detection
- Line break generation and detection
- Fully programmable serial interface characteristics:
  - 5/6/7/8 data length
  - Even, odd and none parity generation and detection
  - 1/2 stop-bit generation
- Extended diagnostic Loopback Mode allows testing more Modem Control and Auto Flow Control features

ast2500 supports 5 serial ports and 12 registers. So I need to know which serial port is used for directional output of BIOS on the board. After searching, I know it is serial port 1.

After getting the serial port, I checked the device tree and found that uart1 was not added to the device tree, which made the serial port unusable. The first thing I thought of was to enable uart 1, that is, to add in the device tree:

```
&uart1 {    status = "okay";
}
```

After adding, the serial port still cannot output the corresponding BIOS information, but through the jumper (first use Inspur's BMC to start, then jumper to my BMC), I found that the serial port can output.

At this point, it means that there should be some initialization configurations of my serial port that have not been loaded and operated, resulting in the failure to output the startup information. At this point, I can only ask Inspur staff for help and ask them to send the device tree configuration of serial port 1 to see the result.

Finally, I got the device tree information of serial port 1:

```
&uart1 { /* Rear RS-232 connector */    status = "okay";    pinctrl-names = "default";    pinctrl-0 = <&pinctrl_txd1_default        &pinctrl_rxd1_default        &pinctrl_nrts1_default        &pinctrl_ndtr1_default
    &pinctrl_ndsr1_default        &pinctrl_ncts1_default        &pinctrl_ndcd1_default        &pinctrl_nri1_default>; };
```



Added to the device tree, communication is normal. From the configuration, the pin is configured, I don't understand the specific configuration of pinctrl, friends who understand it can leave a message to communicate.  
*at last:*

*Like is a virtue,*  
*Attention is fate,*  
*Collection is for sure.*  
*Reward as you like,*  
*Your encouragement is part of the good in my world, love you guys!*