RICO-3288
UART

Edit by Jason Wang

UART

1. Introduction¶

The Universal Asynchronous Receiver/Transmitter (UART) is used for serial communication with a peripheral, modem (data carrier equipment, DCE) or data set. Data is written from a master (CPU) over the APB bus to the UART and it is converted to serial form and transmitted to the destination device. Serial data is also received by the UART and stored for the master (CPU) to read back.

UART Controller supports the following features:

- AMBA APB interface – Allows for easy integration into a Synthesizable Components for AMBA 2 implementation
- Support interrupt interface to interrupt controller
- Contain two 64Bytes FIFOs for data receive and transmit
- Programmable serial data baud rate as calculated by the following: baud rate = (serial clock frequency)/(16×divisor)
- UART_BB/UART_BT/UART_GPS/UART_EXP support auto flow-control, UART_DBG do not support auto flow-control
- UART_DBG support IrDA 1.0 SIR mode with up to 115.2 Kbaud data rate
- UART_BB/UART_BT/UART_GPS/UART_EXP are in peripheral subsystem, UART_DBG is in bus subsystem

2. How to Use¶

Configuration Steps¶

Here we use uart4 as an example.

Create DTS Node¶

The DTS node have already been created in file kernel/arch/arm/boot/dts/rk3288.dtsi, shown as following:

```
uart_exp: serial@ff1c0000 {
        compatible = "rockchip,serial";
        reg = <0xff1c0000 0x100>;
        interrupts = <GIC_SPI 59 IRQ_TYPE_LEVEL_HIGH>;
        clock-frequency = <24000000>;
```

```
        clocks = <&clk_uart4>, <&clk_gates6 12>;
        clock-names = "sclk_uart", "pclk_uart";
        reg-shift = <2>;
        reg-io-width = <4>;
        dmas = <&pdma1 9>, <&pdma1 10>;
        #dma-cells = <2>;
        pinctrl-names = "default";
        pinctrl-0 = <&uart4_xfer &uart4_cts &uart4_rts>;
        status = "disabled";
    };
```

Note: uart_exp is defined in “aliases” node as: serial4 = &uart_exp; The only thing you need to do is adding following code in file kernel/arch/arm/boot/dts/rk3288-tb_8846.dts:

```
&uart_exp {
        status = "okay";
        dma-names = "!tx", "!rx";
        pinctrl-0 = <&uart4_xfer &uart4_cts>;
};
```

Compile and Flash Kernel¶

Turn on CONFIG_SERIAL_ROCKCHIP in kernel configuration, which will add corresponding drivers/tty/serial/rk_serial.c to kernel. Then compile the kernel as followed:

```
make rk3288-tb_8846.img
```