实验五:通用异步收发器（UART）实验

1．实验目的

- 掌握 UART 外设的操作原理和编程。
- 学习使用 UART 进行多机通讯。

2．实验设备

- 硬件：PC 机一台
  Mini2440 ARM 实验板一套
  J-link 仿真器一套
- 软件：WindowsXP 系统，Keil uVision 4.0 集成开发环境

3．实验内容

（1）使用 C 语言编写 UART 基本收发数据程序，进行 2 个实验板之间的数据收发测试。

（2）用两个实验板模拟嵌入式控制系统中的数据采集/控制实验，其中一个实验板模拟数据采集模块，将通过 UART 返回数据；另一块实验板模拟控制系统的主机，通过 UART 采集数据，并通过 UART 发出控制指令。

4．实验预习要求

(1)学习 UART 相关的原理概念；

(2)查阅 S3C2440 芯片手册，了解 UART0 结构和原理。

5．实验步骤

(1)参考实验四的方法和步骤建立工程。启动 Keil uVision，新建一个工程 ex05。不需要系统提供的 Startup 文件。建立汇编源文件 ex05.s，编写实验程序，然后添加到工程中。设置工程选项，存储器映射。设置工程调试选项。建立仿真初始化文件 RAM.ini。

(2)建立 C 语言源文件 main.c，编写实验程序，然后添加到工程中。

(3)使用交叉串口电缆连接两个实验板，见图 5-1。

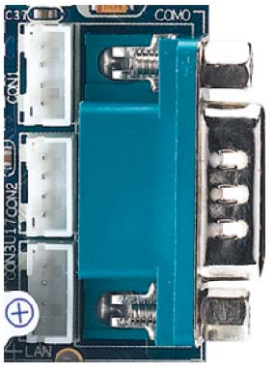

图 5-1 Mini2440 ARM 实验板上的串口 0 接口

(4)编译程序，使用仿真器在目标板上调试运行程序，使用单步、设置断点，

观察程序执行时，收发数据的值。

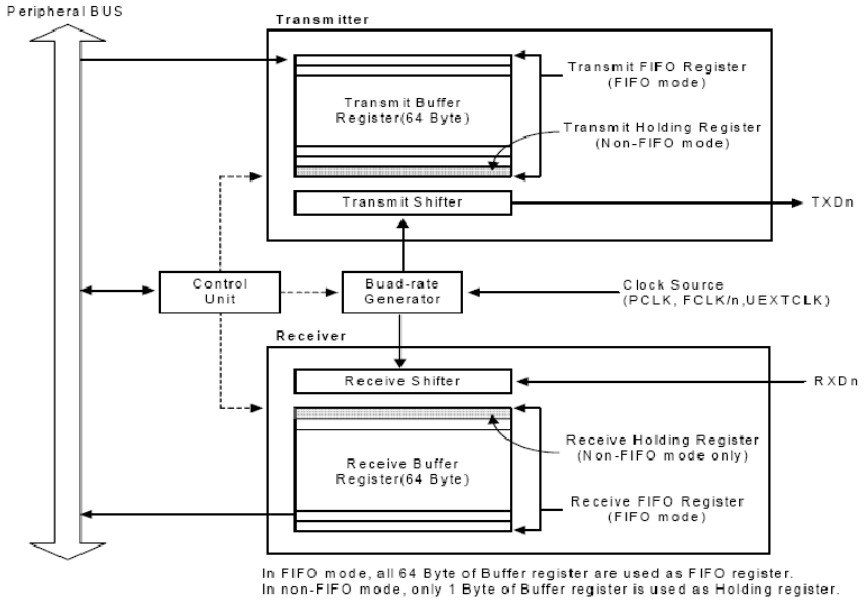

图 5-2 S3C2440UART 结构图

参考：通用异步收发器控制寄存器

UART CONTROL REGISTER

There are three UART control registers including UCON0, UCON1 and UCON2 in the UART block.

Register	Address	R/W	Description	Reset Value
UCON0	0x50000004	R/W	UART channel 0 control register	0x00

Clock Selection	[11:10]	Select PCLK, UEXTCLK or FCLK/n for the UART baud rate. UBRDIVn = (int)(selected clock / (baudrate x 16)) –1 00, 10 = PCLK 01 = UEXTCLK 11 = FCLK/n (If you would select FCLK/n, you should add the code of “NOTE” after selecting or deselecting the FCLK/n.)	0

Tx Interrupt Type	[9]	Interrupt request type. 0 = Pulse (Interrupt is requested as soon as the Tx buffer becomes empty in Non-FIFO mode or reaches Tx FIFO Trigger Level in FIFO mode.) 1 = Level (Interrupt is requested while Tx buffer is empty in Non-FIFO mode or reaches Tx FIFO Trigger Level in FIFO mode.)	0
Rx Interrupt Type	[8]	Interrupt request type. 0 = Pulse (Interrupt is requested the instant Rx buffer receives the data in Non-FIFO mode or reaches Rx FIFO Trigger Level in FIFO mode.) 1 = Level (Interrupt is requested while Rx buffer is receiving data in Non-FIFO mode or reaches Rx FIFO Trigger Level in FIFO mode.)	0
Rx Time Out Enable	[7]	Enable/Disable Rx time out interrupt when UART FIFO is enabled. The interrupt is a receive interrupt. 0 = Disable 1 = Enable	0
Rx Error Status Interrupt Enable	[6]	Enable the UART to generate an interrupt upon an exception, such as a break, frame error, parity error, or overrun error during a receive operation. 0 = Do not generate receive error status interrupt. 1 = Generate receive error status interrupt.	0
Loopback Mode	[5]	Setting loopback bit to 1 causes the UART to enter the loopback mode. This mode is provided for test purposes only. 0 = Normal operation 1 = Loopback mode	0
Send Break Signal	[4]	Setting this bit causes the UART to send a break during 1 frame time. This bit is automatically cleared after sending the break signal. 0 = Normal transmit 1 = Send break signal	0

Transmit Mode	[3:2]	Determine which function is currently able to write Tx data to the UART transmit buffer register. (UART Tx Enable/Disable) 00 = Disable 01 = Interrupt request or polling mode 10 = DMA0 request (Only for UART0), DMA3 request (Only for UART2) 11 = DMA1 request (Only for UART1)	00
Receive Mode	[1:0]	Determine which function is currently able to read data from UART receive buffer register. (UART Rx Enable/Disable) 00 = Disable 01 = Interrupt request or polling mode 10 = DMA0 request (Only for UART0), DMA3 request (Only for UART2) 11 = DMA1 request (Only for UART1)	00

UART TX/RX STATUS REGISTER

There are three UART Tx/Rx status registers including UTRSTAT0, UTRSTAT1 and UTRSTAT2 in the UART block.

Register	Address	R/W	Description	Reset Value
UTRSTAT0	0x50000010	R	UART channel 0 Tx/Rx status register	0x6

UTRSTATn	Bit	Description	Initial State
Transmitter empty	[2]	Set to 1 automatically when the transmit buffer register has no valid data to transmit and the transmit shift register is empty. 0 = Not empty 1 = Transmitter (transmit buffer & shifter register) empty	1
Transmit buffer empty	[1]	Set to 1 automatically when transmit buffer register is empty. 0 =The buffer register is not empty 1 = Empty (In Non-FIFO mode, Interrupt or DMA is requested. In FIFO mode, Interrupt or DMA is requested, when Tx FIFO Trigger Level is set to 00 (Empty)) If the UART uses the FIFO, users should check Tx FIFO Count bits and Tx FIFO Full bit in the UFSTAT register instead of this bit.	1
Receive buffer data ready	[0]	Set to 1 automatically whenever receive buffer register contains valid data, received over the RXDn port. 0 = Empty 1 = The buffer register has a received data (In Non-FIFO mode, Interrupt or DMA is requested) If the UART uses the FIFO, users should check Rx FIFO Count bits and Rx FIFO Full bit in the UFSTAT register instead of this bit.	0

UART TRANSMIT BUFFER REGISTER (HOLDING REGISTER & FIFO REGISTER)

There are three UART transmit buffer registers including UTXH0, UTXH1 and UTXH2 in the UART block. UTXHn has an 8-bit data for transmission data.

Register	Address	R/W	Description	Reset Value
UTXH0	0x50000020(L) 0x50000023(B)	W (by byte)	UART channel 0 transmit buffer register	-

UTXHn	Bit	Description	Initial State
TXDATAn	[7:0]	Transmit data for UARTn	-

UART RECEIVE BUFFER REGISTER (HOLDING REGISTER & FIFO REGISTER)

There are three UART receive buffer registers including URXH0, URXH1 and URXH2 in the UART block. URXHn has an 8-bit data for received data.

Register	Address	R/W	Description	Reset Value
URXH0	0x50000024(L) 0x50000027(B)	R (by byte)	UART channel 0 receive buffer register	-

URXHn	Bit	Description	Initial State
RXDATAn	[7:0]	Receive data for UARTn	-

UART BAUD RATE DIVISOR REGISTER

There are three UART baud rate divisor registers including UBRDIV0, UBRDIV1 and UBRDIV2 in the UART block. The value stored in the baud rate divisor register (UBRDIVn), is used to determine the serial Tx/Rx clock rate (baud rate) as follows:

UBRDIVn = (int)(UART clock / (buad rate x 16)) –1

(UART clock : PCLK, FCLK/n or UEXTCLK)

Where, UBRDIVn should be from 1 to (2^{16}-1), but can be set zero only using the UEXTCLK which should be smaller than PCLK.

For example, if the baud-rate is 115200 bps and UART clock is 40 MHz, UBRDIVn is:

UBRDIVn = (int)(40000000 / (115200 x 16)) -1
= (int)(21.7) -1 [round to the nearest whole number]
= 22 -1 = 21

Register	Address	R/W	Description	Reset Value
UBRDIV0	0x50000028	R/W	Baud rate divisior register 0	-

UBRDIV n	Bit	Description	Initial State
UBRDIV	[15:0]	Baud rate division value UBRDIVn > 0 Using the UEXTCLK as input clock, UBRDIVn can be set '0'.	-

6. 实验参考程序

C 语言实验的参考程序见程序清单 5。

程序清单 4.1 UART 实验参考程序

```
// Uart0

#define WrUTXH0(ch)  (*(volatile unsigned char *)0x50000020)=(unsigned char)(ch)
#define RdURXH0()    (*(volatile unsigned char *)0x50000024)

#define rULCON0      (*(volatile unsigned *)0x50000000)  //UART 0 Line control
#define rUCON0       (*(volatile unsigned *)0x50000004)  //UART 0 Control
#define rUFCON0      (*(volatile unsigned *)0x50000008)  //UART 0 FIFO control
#define rUMCON0      (*(volatile unsigned *)0x5000000c)  //UART 0 Modem control
#define rUTRSTAT0    (*(volatile unsigned *)0x50000010)  //UART 0 Tx/Rx status
#define rUERSTAT0    (*(volatile unsigned *)0x50000014)  //UART 0 Rx error status
#define rUFSTAT0     (*(volatile unsigned *)0x50000018)  //UART 0 FIFO status
#define rUMSTAT0     (*(volatile unsigned *)0x5000001c)  //UART 0 Modem status
```

```
#define rUBRDIV0    (*(volatile unsigned *)0x50000028)  //UART 0 Baud rate divisor

#define rGPHCON    (*(volatile unsigned *)0x56000070)   //Port H control
#define rGPHUP     (*(volatile unsigned *)0x56000078)   //Pull-up control H

//PCLK:12MHz
#define PCLK  12000000
//( (int)(pclk/16./baud+0.5) -1 )
#define baud_value 12    //57600

void Uart_Init()
{
    int i;

    rUFCON0 = 0x0;   //UART channel 0 FIFO control register, FIFO disable
    rUMCON0 = 0x0;   //UART chaneel 0 MODEM control register, AFC disable
    rULCON0 = 0x3;   //Line control register : Normal,No parity,1 stop,8 bits
     //    [10]  [9]   [8]    [7]    [6]    [5]    [4]      [3:2]      [1:0]
     // Clock Sel,  Tx Int,  Rx Int, Rx Time Out, Rx err, Loop-back, Send break,  Transmit Mode, Receive Mode
     //     0          1       0    ,     0          1        0           0     ,     01          01
     //   PCLK       Level    Pulse    Disable    Generate  Normal      Normal          Interrupt or Polling
    rUCON0  = 0x245;   // Control register
    rUBRDIV0= baud_value;   //Baud rate divisior register 0

    for(i=0;i<100;i++)
        ;

    rGPHCON |= 0xaa;//use GPH port as uart0
    rGPHUP  =0x0f;//the pull up function is disabled
}

void Uart_SendByte(int data)
{
    WrUTXH0(data);
}

char Uart_Getch(void)
{
    while(!(rUTRSTAT0 & 0x1))   //Receive data ready
        ;
```

```
    return RdURXH0();
}

main()
{
    char c = 'a';

    Uart_Init();

    Uart_SendByte(c);

    while(1)
    {
        c = Uart_Getch();
        c++;
        Uart_SendByte(c);
    }

}
```

7. 思　考

(1)UART 的 FIFO 对改进通讯性能和可靠性有什么作用？

(2)两个实验板互连通讯，波特率设置满足什么关系时可以正确通讯？

8.选作内容

(1) 使用一个实验板上的按键控制另一个实验板上的 LED 指示灯状态。

(2) 实验板 A 的模数转换器 ADC 采集电压（0～3.3v）数据发给实验板 B，当电压超过正常范围（1～2.5v）时，实验板 B 发出报警信号（4 个 LED 指示灯闪烁）。