串口教程

——疯壳·开发板系列

Wolverine-Team

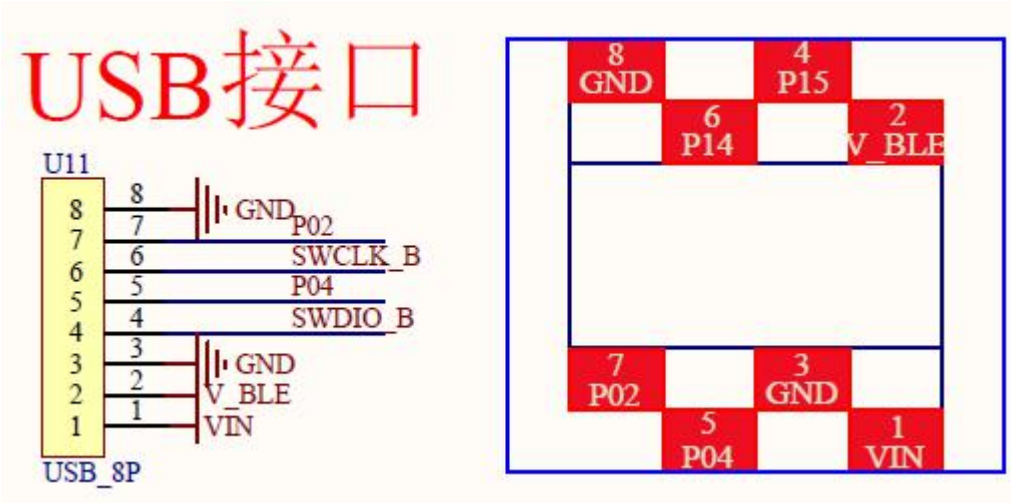
2015/7/24

目录

官网地址：http://www.fengke.club
购买链接：http://shop115904315.taobao.com/
官方 QQ 群：193836402

第一节　串口硬件电路

串口通过 USB 口及手环下载调试线引出，连接 MCU 的 P02 与 P04 引脚，如下图所示：



第二节 UART 寄存器

2.1 UART 介绍

这个 UART 符合工业标准 16550，并且可以和外围设备串行通信。主设备（CPU）通过 APB 总线将数据写进 UART 并且被转换成串行格式并且发送到目标设备。串行数据也可以通过 UART 被接收存储之后，主设备读取接收的数据。

UART 模块不支持 DMA，但是它有内部 FIFOs，并且支持硬件流控制信号（RTS,CTS,DTR,DSR）。

UART 模块有 16 字节的发送和接收 FIFO；支持硬件流控制（CTS/RTS）；影子寄存器来减少软件开销并且有可编程的软件复位；发送寄存器为空的中断模式；IrDA 1.0 SIR 模式支持低功耗模式；可编程的字节属性、校验位和停止位（1,1.5,2）；可以断开通信及检测通信线是否断开；中断优先级的识别；可编程的串行通信波特率。

2.2 UART 寄存器

UART 相关的寄存器比较多，所以我们只介绍常用的寄存器，其它的可以参考官方数据手册 DA14580_DS_v3.1.pdf，位于目录：..\WT 开发板\硬件资料。

2.2.1 接收缓冲寄存器

Table 81: UART_RBR_THR_DLL_REG (0x50001000)

Bit	Mode	Symbol	Description	Reset
15:8	-	-	Reserved	0x0
7:0	R/W	RBR_THR_DLL	Receive Buffer Register: This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overrun error will also occur. Transmit Holding Register: This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If FIFO's are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If FIFO's are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost. Divisor Latch (Low): This register makes up the lower 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set. The output baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor) Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications will occur. Also, once the DLL is set, at least 8 clock cycles of the slowest DW_apb_uart clock should be allowed to pass before transmitting or receiving data.	0x0

15:8 位：保留不使用；

7:0 位：接收缓存寄存器。

2.2.2 中断使能寄存器

Table 82: UART_IER_DLH_REG (0x50001004)

Bit	Mode	Symbol	Description	Reset
15:8	-	-	Reserved	0x0
7	R/W	PTIME_DLH7	Interrupt Enable Register: PTIME, Programmable THRE Interrupt Mode Enable. This is used to enable/disable the generation of THRE Interrupt. 0 = disabled 1 = enabled Divisor Latch (High): Bit[7] of the 8 bit DLH register.	0x0
6:4	-	-	Reserved	0x0
3	R/W	EDSSI_DLH3	Interrupt Enable Register: EDSSI, Enable Modem Status Interrupt. This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. 0 = disabled 1 = enabled Divisor Latch (High): Bit[3] of the 8 bit DLH register	0x0
2	R/W	ELSI_DHL2	Interrupt Enable Register: ELSI, Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. 0 = disabled 1 = enabled Divisor Latch (High): Bit[2] of the 8 bit DLH register.	0x0
1	R/W	ETBEI_DLH1	Interrupt Enable Register: ETBEI, Enable Transmit Holding Register Empty Interrupt. This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt. 0 = disabled 1 = enabled Divisor Latch (High): Bit[1] of the 8 bit DLH register.	0x0
0	R/W	ERBFI_DLH0	Interrupt Enable Register: ERBFI, Enable Received Data Available Interrupt. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFO's enabled). These are the second highest priority interrupts. 0 = disabled 1 = enabled Divisor Latch (High): Bit[0] of the 8 bit DLH register.	0x0

15:8 位：保留不使用；
7 位：发送空中断使能位；
6:4 位：保留不使用；
3 位：调制状态中断使能位；
2 位：接收状态中断使能位；
1 位：发送结束中断使能位；
0 位：数据接收完使能位。

2.2.3 中断识别寄存器/FIFO 控制寄存器

Table 83: UART_IIR_FCR_REG (0x50001008)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	IIR_FCR	Interrupt Identification Register, reading this register; FIFO Control Register, writing to this register. Interrupt Identification Register: Bits[7:6], FIFO's Enabled (or FIFOSE): This is used to indicate whether the FIFO's are enabled or disabled. 00 = disabled. 11 = enabled. Bits[3:0], Interrupt ID (or IID): This indicates the highest priority pending interrupt which can be one of the following types: 0000 = modem status. 0001 = no interrupt pending. 0010 = THR empty. 0100 = received data available. 0110 = receiver line status. 0111 = busy detect. 1100 = character timeout. Bits[7:6], RCVR Trigger (or RT):. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt will be generated. In auto flow control mode it is used to determine when the rts_n signal will be de-asserted. It also determines when the dma_rx_req_n signal will be asserted when in certain modes of operation. The following trigger levels are supported: 00 = 1 character in the FIFO 01 = FIFO 1/4 full 10 = FIFO 1/2 full 11 = FIFO 2 less than full Bits[5:4], TX Empty Trigger (or TET): This is used to select the empty threshold level at which the THRE Interrupts will be generated when the mode is active. It also determines when the dma_tx_req_n signal will be asserted when in certain modes of operation. The following trigger levels are supported: 00 = FIFO empty 01 = 2 characters in the FIFO 10 = FIFO 1/4 full 11 = FIFO 1/2 full Bit[3], DMA Mode (or DMAM): This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals. 0 = mode 0 1 = mode 1 Bit[2], XMIT FIFO Reset (or XFIFOR): This resets the control portion of the transmit FIFO and treats the FIFO as empty. Note that this bit is 'self-clearing' and it is not necessary to clear this bit. Bit[1], RCVR FIFO Reset (or RFIFOR): This resets the control portion of the receive FIFO and treats the FIFO as empty. Note that this bit is 'self-clearing' and it is not necessary to clear this bit. Bit[0], FIFO Enable (or FIFOE): This enables/disables the transmit (XMIT) and receive (RCVR) FIFO's. Whenever the value of this bit is changed both the XMIT and RCVR controller portion of FIFO's will be reset.	0x0

15:0 位：写则为 FIFO 控制寄存器，读则为中断识别寄存器。

2.2.4 通信线控制寄存器

Table 84: UART_LCR_REG (0x5000100C)

Bit	Mode	Symbol	Description	Reset
15:8	-	-	Reserved	0x0
7	R/W	UART_DLAB	Divisor Latch Access Bit. This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers.	0x0
6	R/W	UART_BC	Break Control Bit. This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If active (MCR[6] set to one) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low.	0x0
5	-	-	Reserved	0x0
4	R/W	UART_EPS	Even Parity Select. This is used to select between even and odd parity, when parity is enabled (PEN set to one). If set to one, an even number of logic 1s is transmitted or checked. If set to zero, an odd number of logic 1s is transmitted or checked.	0x0
3	R/W	UART_PEN	Parity Enable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. 0 = parity disabled 1 = parity enabled	0x0
2	R/W	UART_STOP	Number of stop bits. This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to zero, one stop bit is transmitted in the serial data. If set to one and the data bits are set to 5 (LCR[1:0] set to zero) one and a half stop bits is transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit. 0 = 1 stop bit 1 = 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit	0x0
1:0	R/W	UART_DLS	Data Length Select. This is used to select the number of data bits per character that the peripheral transmits and receives. The number of bit that may be selected areas follows: 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits	0x0

15:8 位：保留不使用；
7 位：波特率使能位；
6 位：停止控制位；
5 位：保留不使用；
4 位：奇偶校验选择位；
3 位：校验位；
2 位：停止位；
1:0 位：数据长度。

2.2.5 调制器控制寄存器

Table 85: UART_MCR_REG (0x50001010)

Bit	Mode	Symbol	Description	Reset
15:7	-	-	Reserved	0x0

6	R/W	UART_SIRE	SIR Mode Enable. This is used to enable/disable the IrDA SIR Mode features as described in "IrDA 1.0 SIR Protocol" on page 53. 0 = IrDA SIR Mode disabled 1 = IrDA SIR Mode enabled	0x0
5	R/W	UART_AFCE	Auto Flow Control Enable. Writeable only when AFCE_MODE == Enabled, always readable. When FIFOs are enabled and the Auto Flow Control Enable (AFCE) bit is set, Auto Flow Control features are enabled as described in "Auto Flow Control" on page 58. 0 = Auto Flow Control Mode disabled 1 = Auto Flow Control Mode enabled	0x0
4	R/W	UART_LB	LoopBack Bit. This is used to put the UART into a diagnostic mode for test purposes. If operating in UART mode (SIR_MODE not active, MCR[6] set to zero), data on the sout line is held high, while serial data output is looped back to the sin line, internally. In this mode all the interrupts are fully functional. Also, in loopback mode, the modem control inputs (dsr_n, cts_n, ri_n, dod_n) are disconnected and the modem control outputs (dtr_n, rts_n, out1_n, out2_n) are looped back to the inputs, internally. If operating in infrared mode (SIR_MODE active, MCR[6] set to one), data on the sir_out_n line is held low, while serial data output is inverted and looped back to the sir_in line.	0x0
3	R/W	UART_OUT2	OUT2. This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n, that is: 0 = out2_n de-asserted (logic 1) 1 = out2_n asserted (logic 0) Note that in Loopback mode (MCR[4] set to one), the out2_n output is held inactive high while the value of this location is internally looped back to an input.	0x0
2	R/W	UART_OUT1	OUT1. This is used to directly control the user-designated Output1 (out1_n) output. The value written to this location is inverted and driven out on out1_n, that is: 0 = out1_n de-asserted (logic 1) 1 = out1_n asserted (logic 0) Note that in Loopback mode (MCR[4] set to one), the out1_n output is held inactive high while the value of this location is internally looped back to an input.	0x0
1	R/W	UART_RTS	Request to Send. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] set to zero), the rts_n signal is set low by programming MCR[1] (RTS) to a high.In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] set to one) and FIFOs enable (FCR[0] set to one), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). The rts_n signal is de-asserted when MCR[1] is set low. Note that in Loopback mode (MCR[4] set to one), the rts_n output is held inactive high while the value of this location is internally looped back to an input.	0x0
0	-	-	Reserved	0x0

15:7 位：保留不使用；
6 位：红外通信使能位；
5 位：自动流控制使能位；
4 位：回路位，用于测试；
3 位：用户指定输出 2；
2 位：用户指定输出 1；

1 位：发送请求；
0 位：保留不使用。

2.2.6 通信线状态寄存器

Table 86: UART_LSR_REG (0x50001014)

Bit	Mode	Symbol	Description	Reset
15:8	-	-	Reserved	0x0
7	R	UART_RFE	Receiver FIFO Error bit. This bit is only relevant when FIFOs are enabled (FCR[0] set to one). This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO. 0 = no error in RX FIFO 1 = error in RX FIFO This bit is cleared when the LSR is read and the character with the error is at the top of the receiver FIFO and there are no subsequent errors in the FIFO.	0x0
6	R	UART_TEMT	Transmitter Empty bit. If FIFOs enabled (FCR[0] set to one), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If FIFOs are disabled, this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty.	0x1
5	R	UART_THRE	Transmit Holding Register Empty bit. If THRE mode is disabled (IER[7] set to zero) and regardless of FIFO's being implemented/enabled or not, this bit indicates that the THR or TX FIFO is empty. This bit is set whenever data is transferred from the THR or TX FIFO to the transmitter shift register and no new data has been written to the THR or TX FIFO. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled. If both modes are active (IER[7] set to one and FCR[0] set to one respectively), the functionality is switched to indicate the transmitter FIFO is full, and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting.	0x1

4	R	UART_B1	Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data. If in UART mode (SIR_MODE == Disabled), it is set whenever the serial input, sin, is held in a logic '0' state for longer than the sum of start time + data bits + parity + stop bits. If in infrared mode (SIR_MODE == Enabled), it is set whenever the serial input, sir_in, is continuously pulsed to logic '0' for longer than the sum of start time + data bits + parity + stop bits. A break condition on serial input causes one and only one character, consisting of all zeros, to be received by the UART. In the FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the LSR clears the BI bit. In the non-FIFO mode, the BI indication occurs immediately and persists until the LSR is read.	0x0
3	R	UART_FE	Framing Error bit. This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data. In the FIFO mode, since the framing error is associated with a character received, it is revealed when the character with the framing error is at the top of the FIFO. When a framing error occurs, the UART tries to resynchronize. It does this by assuming that the error was due to the start bit of the next character and then continues receiving the other bit i.e. data, and/or parity and stop. It should be noted that the Framing Error (FE) bit (LSR[3]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 0 = no framing error 1 = framing error Reading the LSR clears the FE bit.	0x0
2	R	UART_PE	Parity Error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set. In the FIFO mode, since the parity error is associated with a character received, it is revealed when the character with the parity error arrives at the top of the FIFO. It should be noted that the Parity Error (PE) bit (LSR[2]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 0 = no parity error 1 = parity error Reading the LSR clears the PE bit.	0x0
1	R	UART_OE	Overrun error bit. This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read. In the non-FIFO mode, the OE bit is set when a new character arrives in the receiver before the previous character was read from the RBR. When this happens, the data in the RBR is overwritten. In the FIFO mode, an overrun error occurs when the FIFO is full and a new character arrives at the receiver. The data in the FIFO is retained and the data in the receive shift register is lost. 0 = no overrun error 1 = overrun error Reading the LSR clears the OE bit.	0x0
0	R	UART_DR	Data Ready bit. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 0 = no data ready 1 = data ready This bit is cleared when the RBR is read in non-FIFO mode, or when the receiver FIFO is empty, in FIFO mode.	0x0

15:8 位：保留不使用；

7 位：接收 FIFO 错误位；

6 位：发送器空位；

5 位：发送保持寄存器空位；
4 位：异常停止中断位；
3 位：帧错误位；
2 位：校验错误位；
1 位：溢出错误位；
0 位：数据准备好位。

2.3 寄存器配置讲解

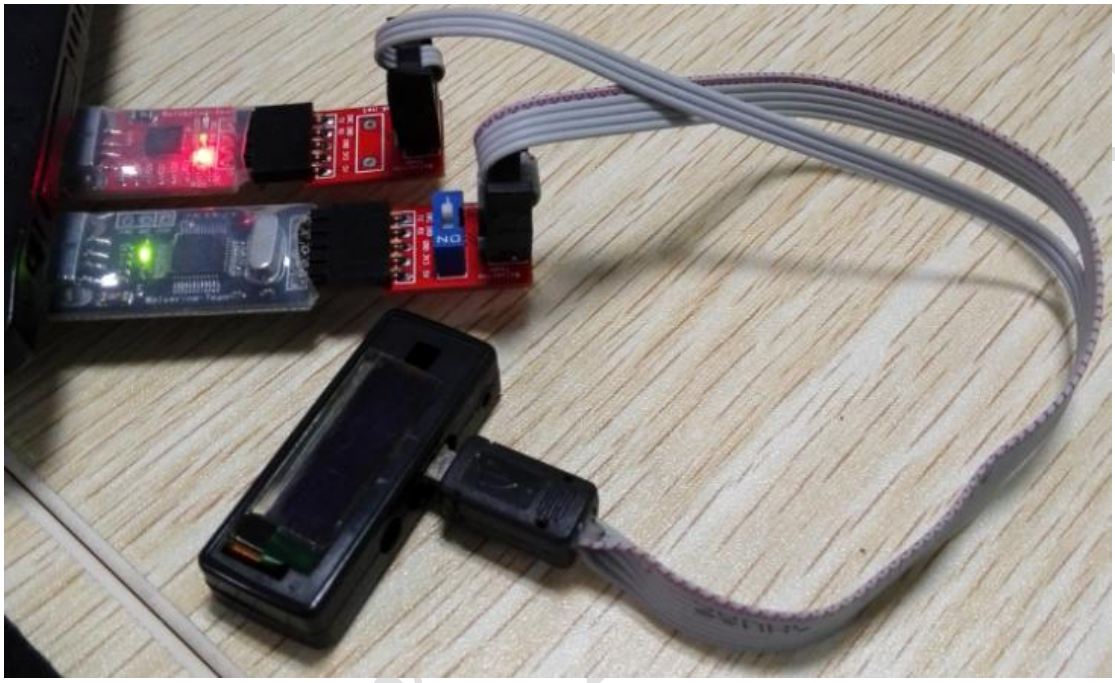
```
#define CLK_PER_REG                 (* ( volatile uint16*)0x50000004)
#define UART_RBR_THR_DLL_REG        (* ( volatile uint16*)0x50001000)
#define UART_IER_DLH_REG            (* ( volatile uint16*)0x50001004)
#define UART_IIR_FCR_REG            (* ( volatile uint16*)0x50001008)
#define UART_LCR_REG                (* ( volatile uint16*)0x5000100C)
#define UART_MCR_REG                (* ( volatile uint16*)0x50001010)
#define UART_LSR_REG                (* ( volatile uint16*)0x50001014)
#define UART_MSR_REG                (* ( volatile uint16*)0x50001018)
#define UART_SCR_REG                (* ( volatile uint16*)0x5000101C)
```

启动 UART1 模块的时钟 CLK_PER_REG |= 0x0080;
串口初始化寄存器配置：
首先使能波特率配置 UART_LCR_REG = 0X08;
配置波特率 UART_IER_DLH_REG = 0; UART_RBR_THR_DLL_REG = 9;
8 位数据，无校验，1 位停止位 UART_LCR_REG = 0X03;
关闭红外通信功能 UART_MCR_REG = 0X00;
使能 FIFO， UART_IIR_FCR_REG = 0X01;
关闭中断 UART_IER_DLH_REG = 0X00;
接收一个字节，等待接收完成 while((UART_LSR_REG&0x01)==0);读取接收数据 rx_data = UART_RBR_THR_DLL_REG;
发送一个字节 0x55，等待发送为空 while((UART_LSR_REG&0x20)==0);填充发送数据 UART_RBR_THR_DLL_REG = 0xaa;

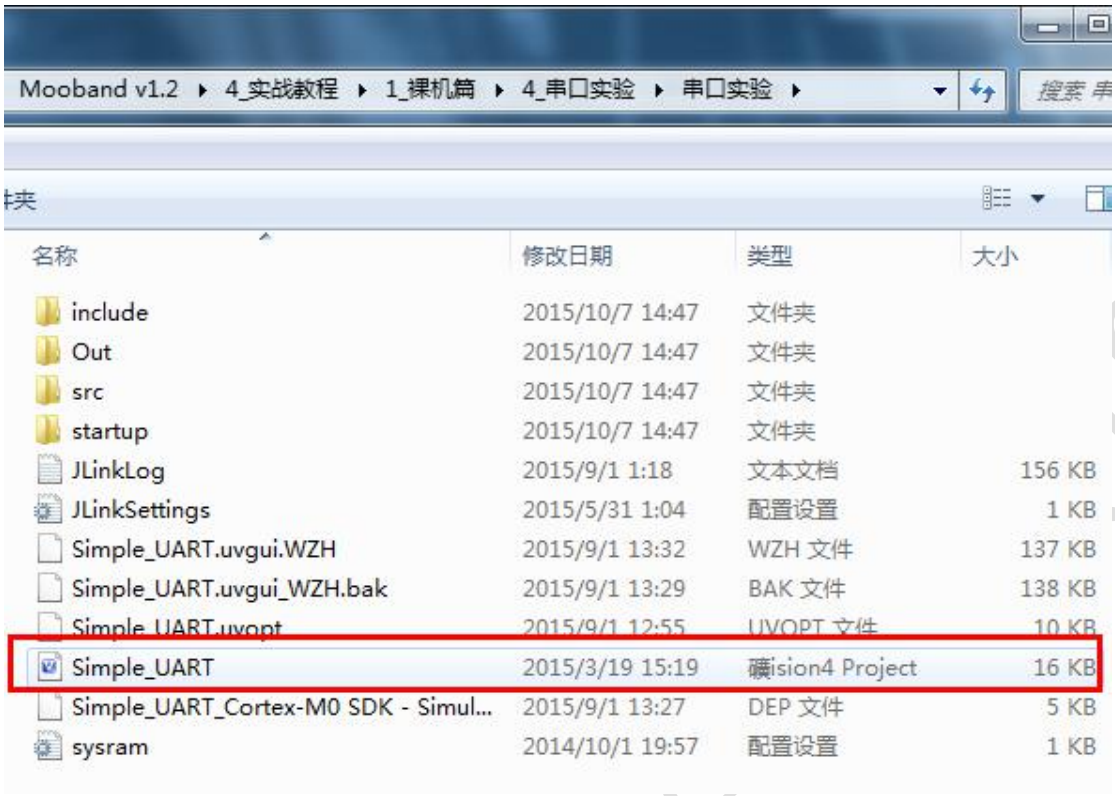
第三节 UART 实验

实验需要使用的模块有：带屏手环，Jlink 调试工具，USB 转串模块，一根手环下载调试线。

将 JLINK 通过下载调试线连接到手环的 USB 调试接口，JLINK 插在有拨码开关的一端，注意丝印标注一一对应，将 JLINK 插上电脑的 USB 口。将 USB 转串模块插在手环现在调试线的另一端，注意丝印标注一一对应，然后将 USB 转串模块插在电脑的 USB 接口。如下图所示：



打开串口实验的 Keil 工程 Simple_UART.uvproj，位于目录：..\ 4_实战教程\1_裸机篇\4_串口实验\串口实验，如下图所示：



打开串口调试助手连接串口，波特率为 115200。打开 KEIL 工程之后，编译代码，点击 DEBUG，然后点击全速运行，就可以看到串口调试助手打印出的信息，发送一个字符，则返回 RX:+发送的字符，如下图所示：

```
************************************************************************

*************************     金刚狼团队     *************************

*************************     Mooband v1.2     *************************

*********************     官网:www.mooband.net     *********************

***************     淘宝:shop115904315.taobao.com     ****************

******************      官方Q Q群: 193836402      ********************

************************************************************************

**************************      UART TEST      **************************

************************************************************************

q

RX:q
6

RX:6
```