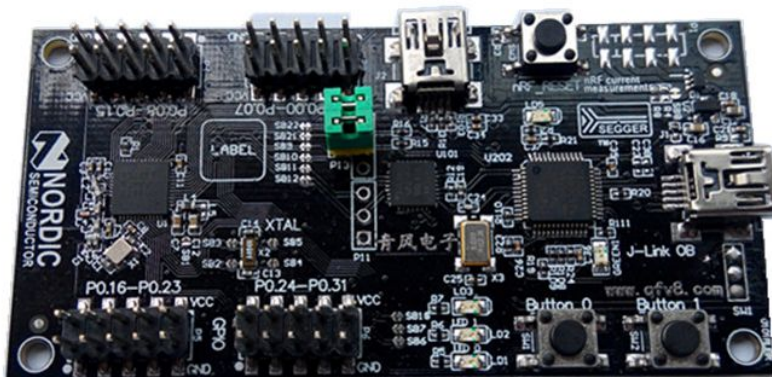


青风带你玩蓝牙 nRF51822 系列教程

-----作者: 青风

出品论坛: www.qfv8.com 青风电子社区

nrf51822蓝牙4.0开发板



青风出品



作者: 青风**出品论坛: www.qfv8.com****淘宝店: <http://qfv5.taobao.com>****QQ 技术群: 346518370****硬件平台: 青云 QY-nRF51822 开发板**

2.7 串口通信的实现

nRF51822 是 cortex m0 内核, 其串口通信也和其他的 cortex m0 内核的处理器相同。并且在官方代码里给出了 uart 的参数配置库函数代码, 这样使用起来非常容易上手。

2.7.1 硬件准备:

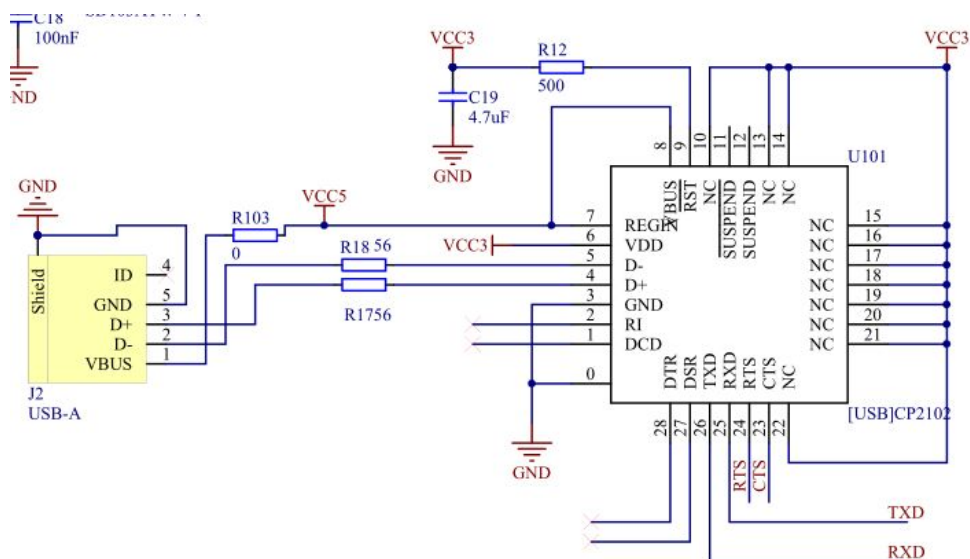
如下图所示: 青云 QY-nRF51822 开发板上, 通过高质量的芯片 CP2102 实现 usb 转串口, 下面是 nrf51822 的芯片串口管脚分配:

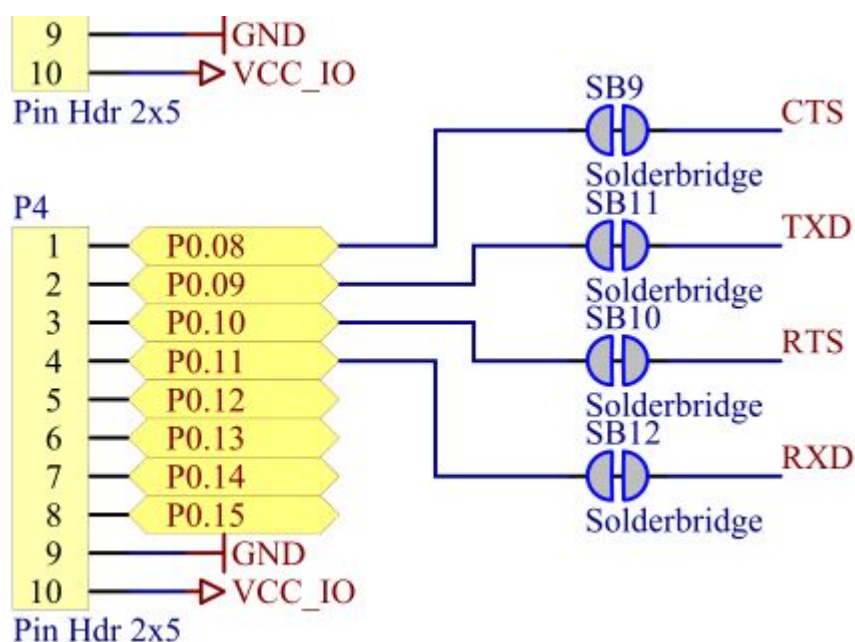
管脚 P0.08 连接 CTS

管脚 P0.09 连接 TXD

管脚 P0.10 连接 RTS

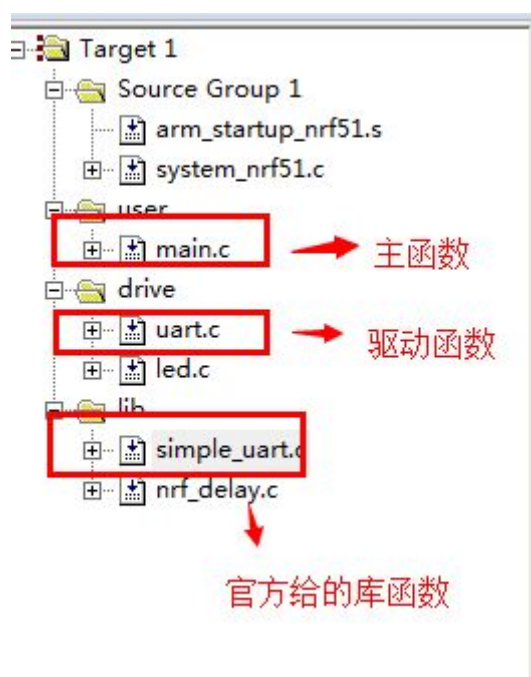
管脚 P0.11 连接 RXD,





2.7.2 软件准备:

在代码文件中,实验七建立了一个演示历程,我们还是采用分层思想,通过官方提供的库文件。打开 user 文件夹中的 uart 工程:



如上图所示：`simple_uart.c` 文件是官方编写好的驱动库文件。我们只需要自己编写 `uart.c` 驱动和主函数 `main.c` 的两个文件就 OK 了，现在我们就来讨论下如何编写 `uart.c` 这个驱动子文件。

下面我们来配置串口，设置代码如下，只使用了 `simple_uart_config()` 一个函数设置对应串口对应的 IO 端口：

```
01. #define RX_PIN_NUMBER 11
02. #define TX_PIN_NUMBER 9
03. #define CTS_PIN_NUMBER 8
04. #define RTS_PIN_NUMBER 10
05.
06. void USART_Configuration(void)//串口初始化函数
07. {
08.     simple_uart_config(RTS_PIN_NUMBER, TX_PIN_NUMBER, CTS_PIN_NUMBER,
09.         RX_PIN_NUMBER, HWFC);
10. }
```

`simple_uart_config()` 函数在官方给出的 `simple_uart.c` 文件中给出了原始定义，我们分析一下这个函数：

```
11. void simple_uart_config( uint8_t rts_pin_number,
12.                           uint8_t txd_pin_number,
13.                           uint8_t cts_pin_number,
14.                           uint8_t rxd_pin_number,
15.                           bool    hwfc)
16. {
17.     nrf_gpio_cfg_output(txd_pin_number);
18.     nrf_gpio_cfg_input(rxd_pin_number, NRF_GPIO_PIN_NOPULL);
19.
20.     NRF_UART0->PSELTXD = txd_pin_number;
21.     NRF_UART0->PSELRXD = rxd_pin_number;
22.     NRF_UART0->PSELCTS = cts_pin_number;
23.     NRF_UART0->PSELRTS = rts_pin_number;
24.
25.     if (hwfc)
26.     {
27.         nrf_gpio_cfg_output(rts_pin_number);
28.         nrf_gpio_cfg_input(cts_pin_number, NRF_GPIO_PIN_NOPULL);
29.         NRF_UART0->CONFIG = (UART_CONFIG_HWFC_Enabled <<
30.             UART_CONFIG_HWFC_Pos);
31.     }
32.     NRF_UART0->BAUDRATE = (UART_BAUDRATE_BAUDRATE_Baud38400 <<
33.         UART_BAUDRATE_BAUDRATE_Pos);
34.     NRF_UART0->ENABLE = (UART_ENABLE_ENABLE_Enabled <<
35.         UART_ENABLE_ENABLE_Pos);
36.     NRF_UART0->TASKS_STARTTX = 1;
37.     NRF_UART0->TASKS_STARTRX = 1;
```

```
35. NRF_UART0->EVENTS_RXDRDY = 0;
36. }
```

我们首先需要配置串口对应的四个管脚，其中 TXD 为输出，RXD 为输入。并且设置如下寄存器确定串口管脚，因为 nrf51822 任意管脚都可以作为串口使用，所有这一点设置很重要：

ERRORSRC	0x480	Error source
ENABLE	0x500	Enable and acquire IOs
PSELRTS	0x508	Pin select for RTS
PSELTXD	0x50C	Pin select for TXD
PSELCTS	0x510	Pin select for CTS
PSELRXD	0x514	Pin select for RXD
RXD	0x518	RXD register
TXD	0x51C	TXD register
BAUDRATE	0x524	Baud rate
CONFIG	0x56C	Configuration of parity and hardware flow

```
37. NRF_UART0->PSELTXD = txd_pin_number;
38. NRF_UART0->PSELRXD = rxd_pin_number;
39. NRF_UART0->PSELCTS = cts_pin_number;
40. NRF_UART0->PSELRTS = rts_pin_number;

    然后启动设置波特率，并且使能串口设置，并且启动串口接收端和发送端：
41. NRF_UART0->CONFIG = (UART_CONFIG_HWFC_Enabled <<
    UART_CONFIG_HWFC_Pos);//配置使能
42. }
43. NRF_UART0->BAUDRATE = (UART_BAUDRATE_BAUDRATE_Baud38400 <<
    UART_BAUDRATE_BAUDRATE_Pos);//波特率设置
44. NRF_UART0->ENABLE = (UART_ENABLE_ENABLE_Enabled <<
    UART_ENABLE_ENABLE_Pos);//串口使能
45. NRF_UART0->TASKS_STARTTX = 1;
46. NRF_UART0->TASKS_STARTRX = 1; //开始串口接收和发送
```

然后重定义 `putc()` 和 `fgetc()` 函数，这样可以使用 `printf` 函数从串口 1 打印输出或者 `scanf` 函数：

```
47. /*****
48. * 函数名 : fputc()
49. * 描述 : 重定义 putc 函数，这样可以使用 printf 函数从串口 1 打印输出
50. * 输入 : None
51. * 输出 : None
52. * 返回 : None
53. * 作者 : 青风
54. * 创建日期: 2014.1.1
55. * 版本 : V1.00
56. *****/
57. int fputc(int ch, FILE *f)
58. {
59.     /* Place your implementation of fputc here */
```

```
60.    /* 发送一个字节的数据 */
61.    simple_uart_put((uint8_t)ch);
62.    /* 等待发送完成 */
63.    return ch;
64. }
65.
66. /*****
67. * 函数名 : fgetc()
68. * 描述 : 重定义 putc 函数, 这样可以使用 printf 函数从串口 1 获取数据
69. * 输入 : None
70. * 输出 : None
71. * 返回 : None
72. * 作者 : 青风
73. * 创建日期: 2014.1.1
74. * 版本 : V1.00
75. *****/
76. int fgetc(FILE *f)
77. {
78.    /* 等待串口 1 输入数据 */
79.    /* 返回接收到的数据 */
80.    return (int)simple_uart_get();
81. }
82.
83.
```

串口发送和接收函数也在 `simple_uart.c` 文件中给出了原始定义, 这个设置比较简单, 配置了基本的接收和发送寄存器:

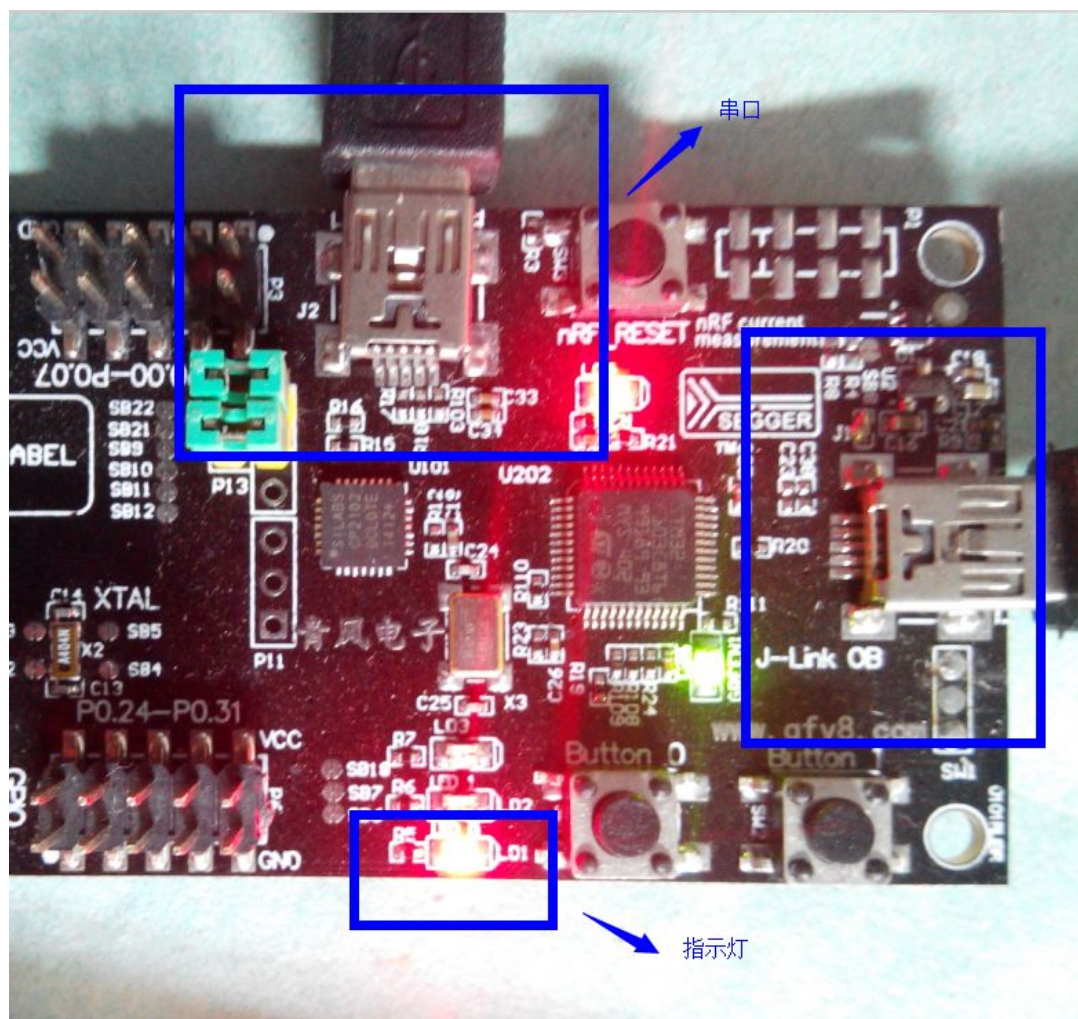
```
84.    //串口发送
85. void simple_uart_put(uint8_t cr)
86. {
87.    NRF_UART0->TXD = (uint8_t)cr;
88.
89.    while (NRF_UART0->EVENTS_TXDRDY!=1)
90.    {
91.        ///等待发送完
92.    }
93.    NRF_UART0->EVENTS_TXDRDY=0;
94. }
95.
96. //串口接收
97. uint8_t simple_uart_get(void)
98. {
99.    while (NRF_UART0->EVENTS_RXDRDY != 1)
100.    {
```

```
101.     ///等待接收到数据
102. }
103.
104. NRF_UART0->EVENTS_RXDRDY = 0;
105. return (uint8_t)NRF_UART0->RXD;//读出数据
106. }
107.
```

那么主函数就是十分的简单了，直接调用我们写好的驱动函数，直接采用 printf 函数打印输出，代码如下所示：

```
108. /***** (C) COPYRIGHT 2014 青风电子 *****/
109. * 文件名   : main
110. * 描述     :
111. * 实验平台: 青云 nRF5188 开发板
112. * 描述     : 串口输出
113. * 作者     : 青风
114. * 店铺     : qfv5.taobao.com
115. *****/
116. #include "nrf51.h"
117. #include "led.h"
118. #include "uart.h"
119. #include "nrf_delay.h"
120.
121. int main(void)
122. {
123.     //
124.     LED_Init();
125.     USART_Configuration();
126.     while (1)
127.     {
128.         LED1_Toggle();
129.         printf(" 2014.1.1 青风!\r\n");
130.         nrf_delay_ms(500);
131.
132.     }
133. }
134.
135.
```

实验下载到青云 nRF51822 开发板后连接 usb 转串口端，如下图所示，然后打开串口调试助手：



实验现象如下，指示灯闪亮表示串口输出进行中，打开串口调试助手，输出要求的内容：

