

3 Hello, World

"Hello, world"几乎所有的编程语言入门教材的第一课。

它的起源要追溯到 1972 年，贝尔实验室著名研究员 Brian Kernighan 在撰写“B 语言教程与指导 (Tutorial Introduction to the Language B)”时初次使用（程序），这是目前已知最早的在计算机著作中将 `hello` 和 `world` 一起使用的记录。之后，在 1978 年，他在他和

Dennis Ritchie 合作撰写的 C 语言圣经“The C Programming Language”中，延用了“hello, world”句式，作为开篇第一个程序。在这个程序里，输出的“hello, world”全部是小写，没有感叹号，逗号后有一空格。虽然之后几乎没能流传下来这个最初的格式，但从此用 `hello world` 向世界打招呼成为惯例。

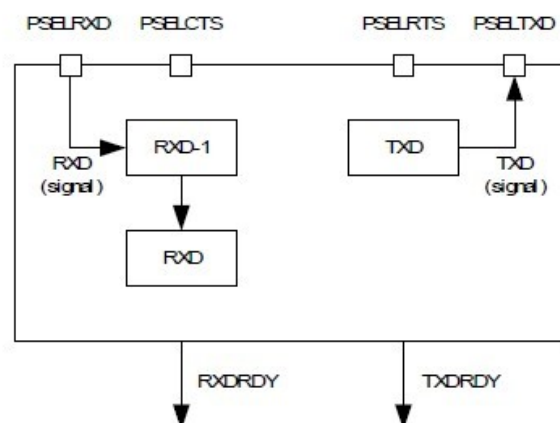
我们的第二课就跟这个“Hello, world”有关。

3.1 nRF51822 的 UART

nRF51822 的串口支持如下特性：

- 全双工（Full-duplex）
- 自动的 flow control
- 奇偶校验并自动产生校验位
- 高达 1Mbps 的带宽

Universal Asynchronous Receiver/Transmitter (UART)



3.1.1 管脚配置

nRF51822 的任何一个 GPIO 都可以用作 UART。这样极大的提高了布线的灵活性，有效的降低了 PCB 的尺寸（或者层数）。

3.1.2 奇偶校验

当使能自动奇偶校验后，校验位可以自送生成。如果不想使用自动方式，还可以由 TXPARITY 寄存器特殊指定。

3.1.3 Error

当结束位没有被正确识别时，将会产生一个 ERROR event。另外，当 RXD 一直被拉低超过一帧数据的长度也会产生错误。

3.1.4 flow control(流控)

CTS 和 RTS 用于控制流量，如果不正确设置将无法显示。请注意相应的串口调试软件中也要根据代码配置开启（或关闭）RTS/CTS 的选项。

3.2 UART 寄存器

Register	Offset	Description
TASKS		
STARTRX	0x000	Start UART receiver
STOPRX	0x004	Stop UART receiver
STARTTX	0x008	Start UART transmitter
STOPTX	0x00C	Stop UART transmitter
EVENTS		
RXDRDY	0x108	Data received in RXD
TXDRDY	0x11C	Data sent from TXD
ERROR	0x124	Error detected
REGISTERS		
INTENSET	0x304	Interrupt enable set register
INTENCLR	0x308	Interrupt enable clear register
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 control

figure 1 UART registers

3.2.1 UART 寄存器简介

3.2.2 波特率寄存器

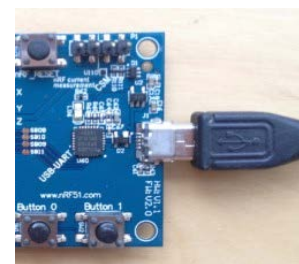
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID (Field ID)	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A				
Reset value	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
ID	RW	Field	Value ID	Value	Description																															
A	RW				Baud-rate																															
			BAUD1200	0x0004F000	1200 baud																															
			BAUD2400	0x0009D000	2400 baud																															
			BAUD4800	0x0013B000	4800 baud																															
			BAUD9600	0x00275000	9600 baud																															
			BAUD14400	0x003B0000	14400 baud																															
			BAUD19200	0x004EA000	19200 baud																															
			BAUD28800	0x0075F000	28800 baud																															
			BAUD38400	0x009D5000	38400 baud																															
			BAUD57600	0x00EBF000	57600 baud																															
			BAUD76800	0x013A9000	76800 baud																															
			BAUD115200	0x01D7E000	115200 baud																															
			BAUD230400	0x03AFB000	230400 baud																															
			BAUD250000	0x04000000	250000 baud																															
			BAUD460800	0x075F7000	460800 baud																															
			BAUD1M	0x10000000	1 megabaud																															

figure 2 UART baudrate registers

3.3 官方 uart 代码样例

3.3.1 硬件连接

请将 Micro USB 连接线（赠品）一端插在 AK board 的“J1”上，另一端插入电脑的 USB 端口。



nRF51822 AK 通过 TX（pin12）和 RX（pin14）收发数据。

安装 SDK 后，源代码位于：

<keil path>\ARM\Device\Nordic\nRF51822\Board\PCA10001\uart_example\

不需要了解代码内容的，编译并下载程序后，可以直接跳到 3.4 小节。



3.3.2 simple_uart_config 函数讲解:

```
void simple_uart_config(
uint8_t rts_pin_number,
uint8_t txd_pin_number,
uint8_t cts_pin_number,
uint8_t rxd_pin_number,
bool hwfc)
{
/* 设置 txd_pin 为输出 */
nrf_gpio_cfg_output(txd_pin_number);
/* 设置 rxd_pin 为输入 */
nrf_gpio_cfg_input(rxd_pin_number, NRF_GPIO_PIN_NOPULL);
/* 设置使用哪个引脚做 RX 和 TX */
NRF_UART0->PSELTXD = txd_pin_number;
NRF_UART0->PSELRXD = rxd_pin_number;
if(hwfc)
{
/* 如果使用硬件流控, 则设置相应引脚 */
nrf_gpio_cfg_output(rts_pin_number);
nrf_gpio_cfg_input(cts_pin_number, NRF_GPIO_PIN_NOPULL);
NRF_UART0->PSELCTS = cts_pin_number;
NRF_UART0->PSELRTS = rts_pin_number;
NRF_UART0->CONFIG = (UART_CONFIG_HWFC_Enabled <<
UART_CONFIG_HWFC_Pos);
}
/* 设置波特率 */
NRF_UART0->BAUDRATE = (UART_BAUDRATE_BAUDRATE_Baud9600 <<
UART_BAUDRATE_BAUDRATE_Pos);
/* 使能串口功能 */
NRF_UART0->ENABLE = (UART_ENABLE_ENABLE_Enabled <<
UART_ENABLE_ENABLE_Pos);
/* 开启相应任务(启动) */
NRF_UART0->TASKS_STARTTX = 1;
NRF_UART0->TASKS_STARTRX = 1;
NRF_UART0->EVENTS_RXDRDY = 0;
}
```

3.3.3 相关宏定义位于头文件"pca10001.h"内：

```
#ifndef PCA10001_H
#define PCA10001_H

#define LED_START      18 #define
LED0      18
#define LED_STOP      22
#define LED1      19
#define LED_PORT      NRF_GPIO_PORT_SELECT_PORT2
#define LED_OFFSET      2

#define BUTTON_START    16 #define
BUTTON0      16 #define
BUTTON_STOP    17 #define BUTTON1
17

#define RX_PIN_NUMBER  14          //for AK!
#define TX_PIN_NUMBER  12          //for AK!
#define CTS_PIN_NUMBER 10
#define RTS_PIN_NUMBER 8
#define HWFC            true

#define BLINKY_STATE_MASK 0x01

#endif
```

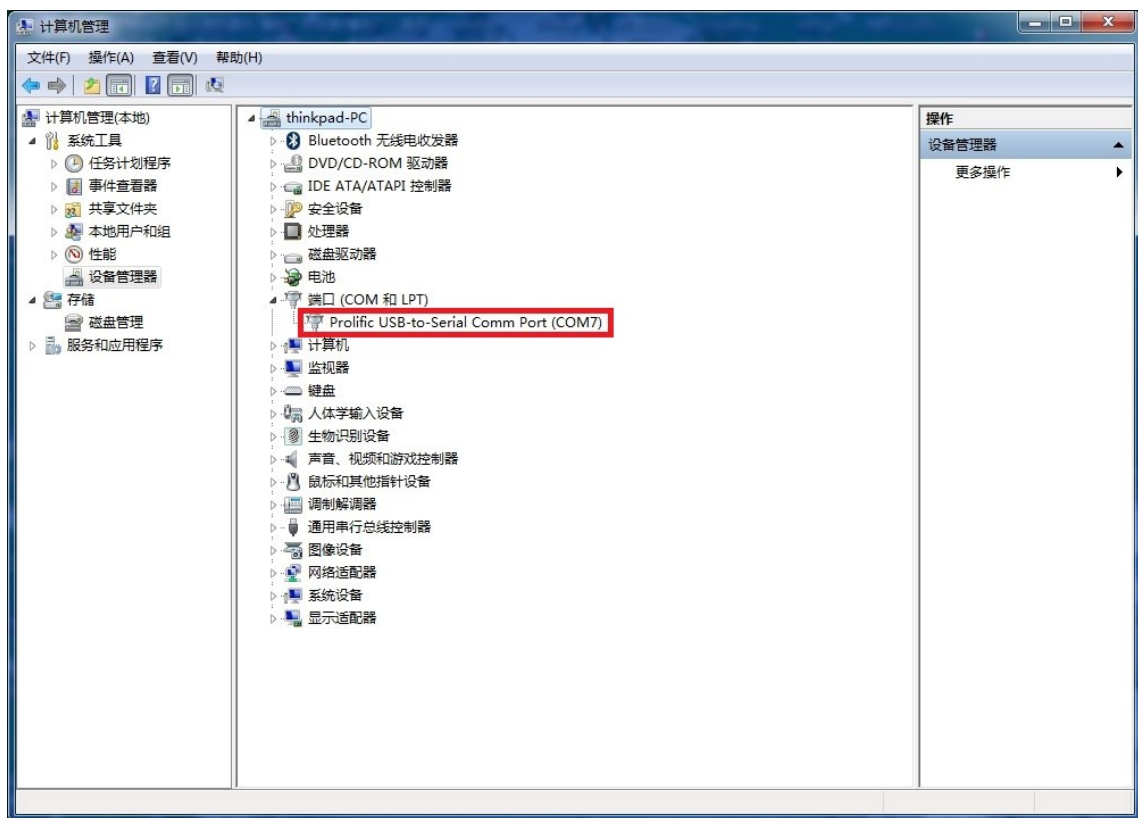
注意：本例程开启 *flow control*，所以设置 *HWFC* 为 *true*。

3.4 打印输出

电脑端串口调试软件可以使用商业化的 **SecureCRT** 或者开源的 **Putty**。下面以 **SecureCRT** 为例，讲解设置流程。

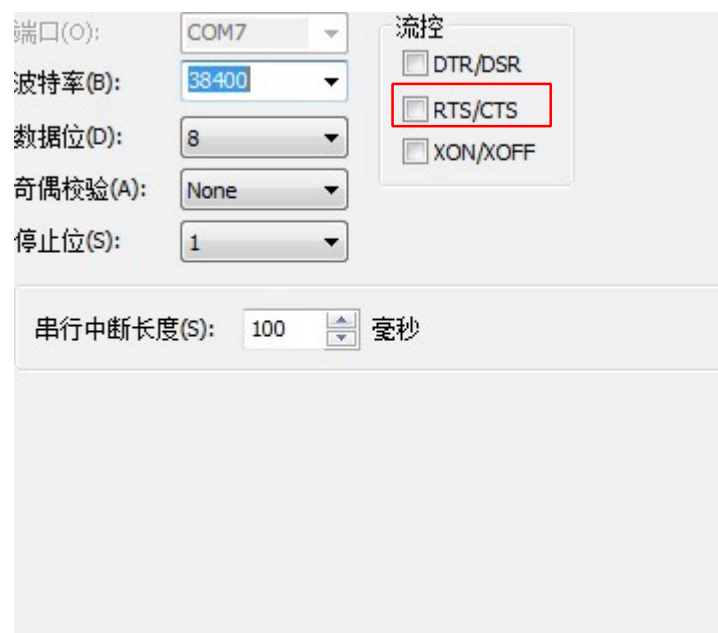
3.4.1 找到虚拟串口

邮件单击“我的电脑”，在弹出菜单中选择“管理”，并在下图弹出窗口左侧选择“设备管理器”。本例中显示，找到的虚拟串口为（COM7）。



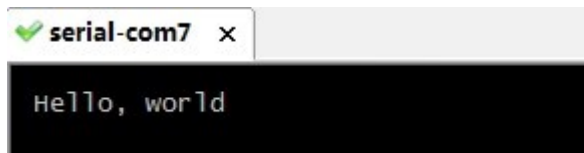
3.4.2 配置 SecureCRT 用 SecureCRT 建立相应串口的连接，本例中

使用（COM7），并进行如下配置（应选中图中的 RTS/CTS 选项）：



3.4.3 输出“Hello, world”

最后，使用 KEIL 下载程序到 nRF51822 AK II 中，串口调试软件会显示如下内容：



本样例还具有回显功能，即敲击键盘后，串口数据会发送到 nRF51822，nRF51822 接收后会回传到串口终端，并在屏幕上显示。

nRF51822 的 Event & Task

nRF51822 中引入了更容易理解的 Event & Task 机制。Task 是命令外设（TWI、I2C、RADIO 之类）执行任务，Event 是外设汇报上来的执行过程中的各种事件（状态）。另外，event 也与 Interrupt（中断）有关。

关于 Task，Event 以及 Interrupt 我们会在后面详细介绍。