

实验三：异步串行通信实验

一、实验目的

1. 学习mbed操作系统对异步串行通信的处理方法；
2. 掌握异步串行通信与GPS模块的通信技术；
3. 掌握调试程序的方法

二、实验仪器与设备

Mbed 开发板、连接线、计算机、GPS 模块

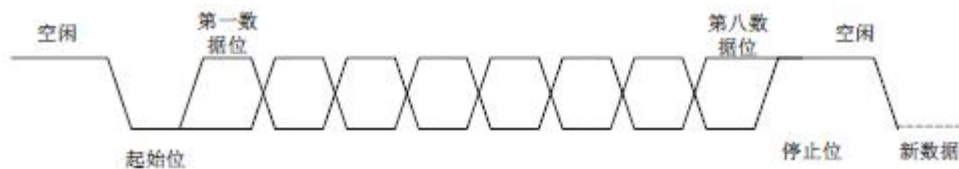
三、预备知识

1、mbed Uart 串行通讯基础及API

Uart (Universal Asynchronous Receiver/Transmitter) 异步串口通讯是各类单片机中最古老又是最常用的通讯方式，在UART 通讯过程中，我们需要使用3 条信号线，即发送 (RX)，接收 (TX) 和地 (GND)，对于收发双方来说，RX 和TX 要交叉连接。由于这三条线中并不带时钟信号，所有在使用时必须约定数据的发送速度，即波特率；另外为了让接收方能够准确地识别出1 个字符，还还需要在发送时添加起始位(1 位)和停止位(1, 1.5, 2 位)。由于UART通讯以字符为传输单位（一般是8 个bit，但也可以是其它，如7 个，9 个），且字符的发送时间是不确定的（异步的），所以我们称它为异步串口通讯。UART 串口通讯具有结构简单，实现方便的优点，但也有传输速度低的缺点。

在具体的应用中，UART 串口有TTL 电平的串口和RS232 电平的串口类中，TTL 电平是高电平为3.3V，低电平为0；而RS232 是负逻辑电平，它定义+5~+12V 为低电平，而-12~-5V 为高电平，对于不加外部电路的单片机而言，它的UART 输出都是TTL 电平，相对RS232 电平来说，通讯距离较短，而且容易受到干扰，但功耗较低，适合1 米以内的短距离数据传输。

下图是UART 数据通讯过程中的时序图：



Uart 串口通讯其实在前面的代码中已经用到过，标准C 语言的printf 函数也被重定义到串口输出上，方便用户的调试，对于mbed 来说，它使用Serial 对象来完成串口的数据收发，它提供的主要方法有：

类名	方法	用途
Serial	Serial(PinName tx, PinName rx, const char *name=NULL);	构造函数, 把 tx,rx 设成 Uart 的输出输入管脚
	void baud(int baudrate);	设置 Uart 的波特率, 默认为 9600
	void format(int bits=8, Parity parity=SerialBase::None, int stop_bits=1);	设置 Uart 传输的格式, 包括一个字长的位数、奇偶检验的方法、停止位的位数, 默认为一个字长为 8 位, 无奇偶检验, 1 位停止位
	int readable();	返回 Uart 是否有数据到达
	int writeable();	返回 Uart 是否还有空间进行数据发送
	void attach(void (*fptr)(void), IrqType type=RxIrq);	设置 Uart 中断时需要执行的用户自定义函数
	void set_flow_control(Flow type, PinName flow1=NC, PinName flow2=NC);	设置 Uart 的流控方法, 其目标是提高数据发送的可靠性, 流控方法有无流控, RTS 流控, CTS 流控, RTSCTS 流控, 后面两个常数为流控的管脚设置
	int getc();	从 Uart 读取一个字符
	int putc(int c);	向 Uart 发送一个字符
	int printf(const char* format, ...)	格式化 Uart 的输出, 参数等同标准 C 的 printf
	int scanf(const char* format, ...);	格式化 Uart 的输入, 参数等同标准 C 的 scanf

2、GPS 模块数据格式

GPS 一般都是采用 Uart 进行通讯的, 它在上电后就会不断输出符合标准的文本信息, 其格式如下, 该标准为 NMEA-0183 协议

NMEA-0183 数据都是以 \$ 开始, 接着是信息类型, 后面是数据, 以逗号分隔开。在 NMEA-0183 标准中一共定义了以下数据类型, 但并不是所有的 GPS 设备都会输出所有的信息类型:

GPGSV: 可见卫星信息
 GPGLL: 地理定位信息
 GPRMC: 推荐最小定位信息
 GPVTG: 地面速度信息
 GPGGA: GPS 定位信息
 GPGSA: 当前卫星信息

其中最重要的就是 GPGGA 和 GPRMC 定位信息, 其格式说明如下:

\$GPGGA, <1>, <2>, <3>, <4>, <5>, <6>, <7>, <8>, <9>, M, <10>, M, <11>, <12>

<1> UTC 时间, hhmmss (时分秒) 格式
 <2> 纬度 ddm. mmmm (度分) 格式 (前面的 0 也将被传输)
 <3> 纬度半球 N (北半球) 或 S (南半球)
 <4> 经度 dddmm. mmmm (度分) 格式 (前面的 0 也将被传输)
 <5> 经度半球 E (东经) 或 W (西经)
 <6> GPS 状态: 0=未定位, 1=非差分定位, 2=差分定位, 6=正在估算
 <7> 正在使用解算位置的卫星数量 (00~12) (前面的 0 也将被传输)
 <8> HDOP 水平精度因子 (0.5~99.9)
 <9> 海拔高度 (-9999.9~99999.9)

<10> 地球椭球面相对大地水准面的高度

<11> 差分时间（从最近一次接收到差分信号开始的秒数，如果不是差分定位将为空）

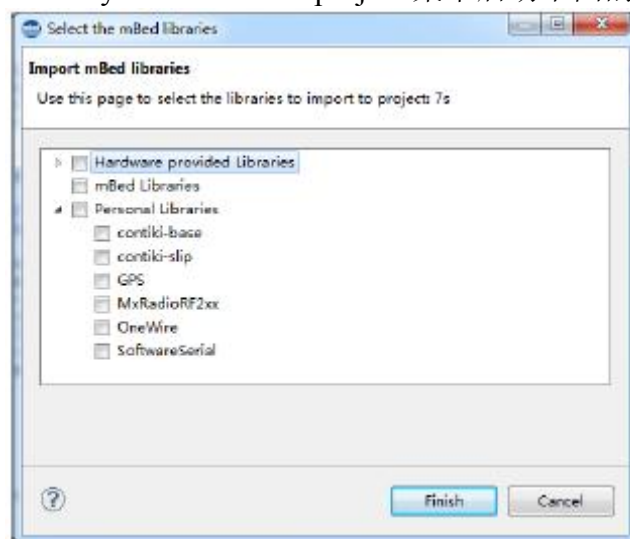
<12> 差分站 ID 号 0000~1023（前面的 0 也将被传输，如果不是差分定位将为空）

3、GPS Mbed 库函数的获取

可以直接通过串口获取数据后进行大量的字符串运算获取，也可以利用现有的库函数，本教材用的是 mbed 网站上 <https://os.mbed.com/users/simon/code/GPS/> 其具体方法如下：

下载此库的压缩包，然后解压到 SMeshStudio 的私有库文件中，默认为 mbed/privatelibrary;

使用 mbed——>Add a library to the selected project 菜单启动下面的界面：



此时选择我们需要的 mbed 扩展库即可，这里是 GPS，点击 Finish 按钮继续即可。

四、 实验内容

（1） 串口输入输出实验：

在Uart串口的双向通讯中，经常涉及到一个用户交互的问题，如当用户执行完一段代码后，经常会说请按任意键或特定键继续，这是，就相当于让程序一直等待，直到程序期待的字符出现，这在mbed中是很容易的，参考代码如下：

```
Serial pc(USBTX,USBRX);
DigitalOut led(LED1);
char username[100];
int userkey;
int main()
{
    pc.printf("Hello World,please enter you name to continue\r\n");
    pc scanf("%s",username);
    pc.printf("You name is %s \r\n",username);
    while (pc.readable())
    pc.getc();
    while (1)
```

```

{
    pc.printf("Hello World,please enter return to continue\r\n");
    userkey=pc.getc();
    if (userkey=='\r')
        break;
    else
        pc.printf("Wrong key,please enter return key to continue \r\n");
}
pc.printf("Right key,good bye \r\n");
}

```

(2) 读取GPS数据实验

使用库函数，测试代码如下：

```

Serial pc(USBTX,USBRX);
GPS mygps(USBTX,USBRX);
int main() {
    mygps.Init();
    while (1)
    {
        mygps.parseData();
        pc.printf("alt:%f lat:%f lon:%f\n",mygps.altitude,mygps.latitude,mygps.longitude);
        wait(1);
    }
}

```

五、实验要求

- 1、认真阅读预备知识和实验内容；
- 2、按图示连接 Mbed 开发板和 GPS 模块；
- 3、编写代码，编译、下载到 mbed 开发板，验证实验效果。
- 4、撰写实验报告。