ARM实验（Cortex-M4）

# 实验三 UART串行通讯口及中断优先级实验

### 一. 实验目的

了解UART串行通讯的工作原理

掌握在PC端通过串口调试工具与实验板通过UART通讯的方法

掌握UART的堵塞式与非堵塞式通讯方法

### 二．实验要求

1. 例程为UART0的初始化，实验板在初始化完成后向主机发送“HELLO,WORLD！”字符串。阅读3-1.c并理解。

2. 在实验1的基础上，通过PC端发送字符串，实验板收到后并原样返回。称为UART ECHO。阅读3-2.c并理解。

3. 将实验2改写为非堵塞式方式，即中断方式进行发送与接收。当进行数据接收时，点亮PN1。阅读3-3.c并理解。

4. 请编程实现一个虚拟AT指令集：

当PC端发来AT+CLASS后，实验板回以CLASS######，其中#####为你的班级号

当PC端发来AT+STUDENTCODE后，实验板回以CODE######，其中#####为你的学号

5. 选做。将4改为大小写均能适应。

6. 选做。请编程实现以下三个命令：

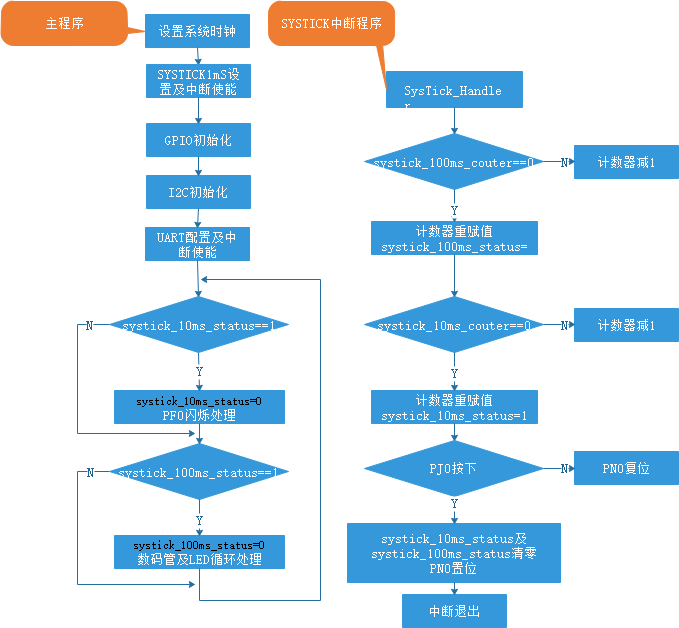
底板运行后自动实现1S计时。

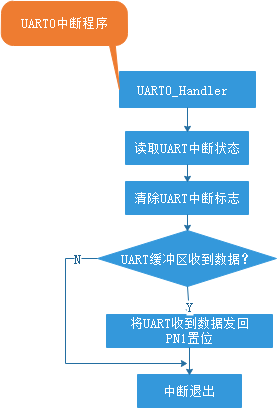
1. PC端发来绝对对时命令，如SET12:56:03或12-56-03，自动将当前时间同步到12:56:03，并回之以当前时间
2. PC端发来相对对时命令，如INC00:00:12，自动将当前时间加12秒，并回之以当前时间
3. PC端发来查询命令，GETTIME，自动回之以当前时间
4. 当前时间格式统一为TIME12:56:03，其中TIME为字符，后续为时间值

7. 优先级调整实验。基于实验3，调整UART0的优先级，使之高于SYSTICK的优先级，并处于抢占式优先，这样当按下USR\_SW2时，UART0不退出，导致SYSTICK中断不能进入，整个系统停滞。显示不再跳变。请阅读3-4.c并理解。

改写程序，将SYSTICK的抢占式优先级高于UART0，从而达到即使USR\_SW2按下时，SYSTICK中断仍然能进入。

### 三.实验框图





### 四.实验结果

### 五. 讨论

1. 实验3-2，if (UARTCharsAvail(UART0\_BASE))此行程序的作用。如果没有此行，会导致什么问题？

2. 实验3-3，void UART0\_Handler(void)为什么没有在主函数声明？

3. 为什么3-3的中断中需要读取中断标志并清除，而SYSTICK不需要

4. 请根据上位机的命令，如“MAY+01”，格式为：

其中MAY为月份，（JAN,FEB,…DEC）均为三位。

+表示加运算符,-表示减运算符，均为1位。

01表示增加或减少量，均为2位。范围00-11

以上均为ASCII码，

MAY+01应该回之以JUNE

MAY-06应该回之以NOV

5. 请根据上位机的命令，如“14:12+05:06”，格式为：

其中14:12为分钟与秒，共5位，包括一个“:”。

+表示加运算符,-表示减运算符，均为1位。

05:06为分钟与秒的变化量，共5位。包括一个“：”，范围00:00~23:59

以上均为ASCII码，

14:12+05:06回之以19:18

六、程序中涉及相关固件库函数说明，

页码为SW-TM4C-DRL-UG-2.1.4.178.pdf中页码。

|  |  |  |
| --- | --- | --- |
| 函数名 | 页码 | 功能说明 |
| GPIOPinTypeUART | 281 | 配置引脚作为UART设备 |
| UARTConfigSetExpClk | 568 | 配置UART，包括地址，波特率，数据格式等 |
| UARTCharPut | 565 | 从指定端口发送一个字节直到发送完成 |
| UARTCharPutNonBlocking | 565 | 从指定端口发送一个字节 |
| UARTIntStatus | 576 | 获取当前中断状态值 |
| UARTIntClear | 574 | 清除UART中断源 |
| UARTCharsAvail | 566 | 确定接收FIFO中是否有字符 |
| SysTickPeriodSet | 532 | 设置SYSTICK计数器的周期值 |
| SysTickEnable | 530 | 使能SYSTICK计数器 |
| SysTickIntEnable | 531 | SYSTICK中断允许 |
| UARTIntEnable | 575 | 启用单个UART中断源 |
| IntEnable | 352 | 使能某个中断 |
| IntPriorityMaskGet | 265 | 获取优先级屏蔽级别 |
| IntPrioritySet | 358 | 设置某个中断的优先级级别 |
| IntPriorityGroupingGet | 356 | 获取中断控制器的优先级分组 |
|  |  |  |