

实验三 UART 串行通讯口实验

一. 实验目的

- 了解 UART 串行通讯的工作原理;
- 掌握在 PC 端使用串口调试工具(SerialProV1.04.exe)与实验板进行 UART 通讯的方法;
- 掌握 UART 的堵塞式与非堵塞式数据传输编程方法;
- 掌握中断优先级设置方法及对程序运行的影响。

二. 实验要求

1. 例程 exp3-1.c 完成 UART0 的初始化,并在初始化完成后向主机发送“HELLO,WORLD!”字符串。Task1 完成 UART ECHO,即通过 PC 端串口调试窗口向实验板发送字符串,实验板收到后原样返回;Task2 为跑马灯控制。阅读并理解例程。

➤ 修改程序,将收到的字符改为大写后返回。

2. 例程 exp3-2.c 将实验 3-1 的 Task1 改为非堵塞式方式(即中断方式)接收字符串,并原样返回,以回车换行“`\r\n`”表示一次接收完成。当进行数据接收时,点亮 PN1。全部接收完成后,熄灭 PN1。阅读理解例程,并思考:当 PC 端一次发送字符大于 8 个时,观察串口调试窗口的信息变化并解释原因。

➤ 修改程序结构,将 Task1 移至中断处理程序中(参见三.实验框图),说明两种结构各自的优势。

3. 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。(要求大小写均能适应)

4. 例程exp3-4.c优先级调整实验。

- 任务1,主循环走马灯;
- 任务2, SysTick中断中点亮D1,长按SW1点亮D2,释放熄灭D2;
- 任务3, UART0中断中熄灭D1和D2,长按SW2保持UART接收状态。

调整UART0 的优先级,使之高于SYSTICK 的优先级,并处于抢占式优先。这样当长按 USR_SW2且超级终端有数据传送时, UART0 中断不退出,导致SYSTICK 中断不能进入,任务2无法执行,任务1显示也不再跳变。请阅读并理解例程。

➤ 修改任务优先级,使SysTick优先级大于或等于UART0优先级,或者分组设置为不抢占时,长按USR_SW1和USR_SW2分别会出现怎样的现象,观察并解释任务1~3的执行情况。

5. 请根据上位机的命令作应答。命令格式形如“MAY+01”。其中:

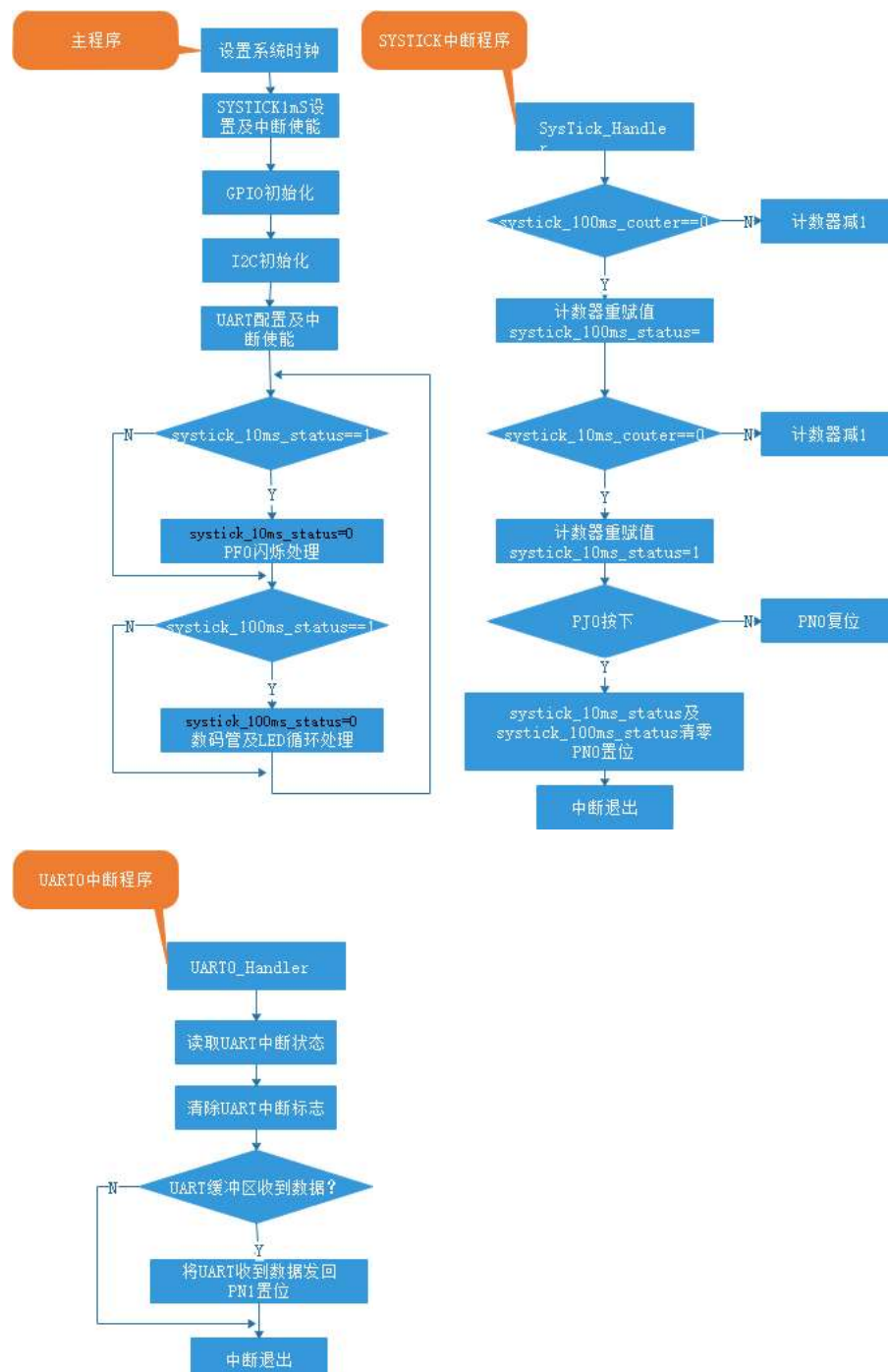
- MAY 为月份, (JAN, FEB, ..., DEC)均为三位。
- +表示加运算符, -表示减运算符, 占 1 位。加减均对月份进行模12运算。
- 01 表示增加或减少量,均为2 位。范围00~11。

如: MAY+01 应该回之以JUN, MAY-06 应该回之以NOV

6. 编程实现模拟时钟程序。底板Reset后从12:00:00开始自动进行1秒计时，并在数码管上以“HH-MM-SS”形式动态显示当前时间。程序运行中可以接收以下三个命令：
- a) PC端发来绝对对时命令，如SET12:56:03 或SET12-56-03，自动将当前时间同步到12:56:03，并回之以当前时间；
 - b) 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；
 - c) PC端发来查询命令，GETTIME，自动回之以当前时间；
- 所有回复的当前时间格式统一为TIME12:56:03，其中TIME 为字符，后续为时间值。

三.实验框图

实验 3-2 的中断驱动程序结构



四.实验结果

应能现场演示实验的效果。

五. 讨论

1. 实验 3-1 中, `if(UARTCharsAvail(UART0_BASE))`此程序的作用。如果没有此行, 会导致什么问题?
2. 实验 3-2 中, 采用中断方式进行 UART 接收时, 为什么要使能 “接收超时” 的中断条件? 若设 RxFIFO 深度分别为 4/8 和 7/8, 则当 PC 端一次性发送 17 个字符时, 至少会产生几次 UART 中断请求?
3. 实验 3-2 中, `void UART0_Handler(void)`为什么没有在主函数前声明? 为什么 UART 和 GPIO 的中断处理程序需要读取并清除中断状态, 而 SYSTICK 不需要?
4. 实验3-4中, 执行语句 `IntPrioritySet(INT_UART0,0x3);` 后, Uart0的实际优先级为多少?