

专业课程设计I报告

（ 2022 / 2023学年 第 二 学期）

题 目： 微机串口通信功能测试

|  |  |
| --- | --- |
| **专 业** | **计算机科学与技术** |
| **学 生 姓 名** | **周承业** |
| **班 级 学 号** | **B20030726** |
| **指 导 教 师** | **李爱群** |
| **指 导 单 位** | **计算机基础教学中心** |
| **日 期** | |  | | --- | | **2023.06.05-2023.06.18** | |

|  |  |  |
| --- | --- | --- |
| **支撑指标点** | **评价准则** | **计分（每项10分）** |
| 课程目标1：通过课程设计，培养学生综合应用汇编语言、计算机接口技术领域专业知识的技能。（20分） | 1、能够掌握汇编语言程序设计、微型计算机接口技术的相关基础知识，并能够针对求解的工程问题，收集资料进行合理的分析与方案设计 |  |
| 2、通过调研，能够选择合适的编程软硬件开发工具和平台，对求解的工程问题进行软硬件设计 |
| 课程目标2：深入理解相关知识，比较各种解决方案的优缺点，解决复杂工程问题的实践创新能力。（20分） | 3、能够选用合适接口芯片，对不同方案优缺点进行比较，给出最优设计方案。 |  |
| 4、能够给出硬件实现具体方案以及变量和算法的设计描述，给出关键算法的流程图或伪代码。 |
| 课程目标3：培养开发工具运用能力，能够利用软硬件开发工具和实验平台进行软硬件开发，培养工程实践能力。（30分） | 3、对汇编语言程序设计、微型计算机接口技术相关知识有一定的掌握，能够完成课题要求的各项任务和指标 |  |
| 4、能够利用计算机软硬件开发工具和实验平台，合理选用常规接口芯片，理解相关算法，并进行硬件连线和程序实现。 |  |
| 5、掌握调试方法与工具，对程序开发过程中出现软硬件问题进行分析、跟踪与调试，并能够进行充分测试 |  |
| 课程目标4：分组完成一次项目设计与开发的全过程，组内成员通过讨论和交流解决课程设计中的难题，在程序验收过程和实验报告中准确阐述课程设计的内容，能够针对具体工程问题指定解决方案，清晰陈述观点和回答问题。（30分） | 6、组内成员之间有一定的团队合作，互通有无 |  |
| 7、能够正确、完整地回答指导教师关于课题的问询，反映其对课题内容，以及相关的工程基础知识具有较好的理解和掌握 |  |
| 8、具备一定的语言表达能力与文字处理能力，能够结合复杂工程问题撰写报告，报告内容和实验数据详实，格式规范 |  |
| 专业课程设计I能力测评总分 | |  |
| **指导教师： 年 月 日** | | |
| **备注：** | | |

**微机串口通信功能测试**

**一、课题内容和要求**

课题内容：运用微机系统串行口知识，进行 PC 系统机的串行口通信功能测试。

基本要求：

（1）通过硬件设计完成两台 PC 机的串口连接；

（2）设计发送和接收两个窗口，分别显示发送和接收的数据。发送的数据从键盘键入，数据接收后送屏幕显示；数据发送采用查询方式，数据接收采用中断方式；

（3）波特率、帧格式可以通过人机交互进行设置；

（4）实物演示时要求讲出程序原理和设计思想；

（5）程序运行良好、界面清晰。

提高要求：

（1）界面色彩鲜明、人机交互友好；

（2）可以发送、接收文件。

**二、课题需求分析**

本课题目标系统的功能框架图如图1所示。

图形用户界面

低可信度描述已自动生成

图1 功能框架图

具体需求如下：

（1）用户可以设置波特率的大小；

（2）用户可以根据需求设置帧格式；

（3）用户可以自己决定发送的消息并将其发送；

（4）在另一台设备上显示消息内容。

**三、课题设计**

1 硬件设计

利用RS-232C接口连接两台机器的串行通信接口。

2 软件设计

（1）主要变量及数据结构

a.用来存放消息记录的缓冲区域

MESG DB 128

DB ?

DB 128 DUP(?)

LENS EQU $-MESG

b.一些提示性语句或绘制界面所需的特殊图像

A1 DB " ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~$"

A2 DB " \*\*\* \* \* \* Internal Loopback \* \* \* \*\*\*\*$"

A3 DB " designer:B20030726 zcy $"

A4 DB " Please input the word:$"

A5 DB " THE output:$"

A6 DB " -----------------------------------$"

A7 DB " $"

A8 DB " ENTER BAUD(like 01 01 and <=01 80 >=00 0C):$";长度45

A9 DB " ENTER FRAME FORMAT(<=3F ):$";长度27

;波特率报错

WRONGB DB " !!! Wrong!ENTER BAUD(like 01 01 and <=01 80 >=00 0C) !!!$"

;帧格式报错

WRONGF DB " !!! Wrong!PlEASE ENTER FRAME FORMAT(<=3F ) !!!$"

c.一些变量地址用来存放输入的字符或作为中间变量

OLD0B DD ?

STRING DB 6,0,6 DUP(0)

;6位BUF,用来存放波特率高位和低位(两个16进制数和一个空格)

STRINGF DB 3,0,3 DUP(0);3位，用来存放帧格式

DATA1 DB 2 DUP(0) ;2位data用于存放转化后的波特率

DATAF DB 1 DUP(0) ;1位dataf用于存放转化后的帧格式

2 主要算法流程

1. 查询式输入：采用查询方式接受数据前，CPU要查询输入数据是否准备好。只有确认了外设已具备了输入或输出的条件后，才能用IN或OUT指令完成数据传送。输入设备准备好数据后，发出输入选通信号，一方面把数据送入数据锁存/缓冲器，另一方面使状态标志触发器置1，状态标志是一位信息，接到CPU数据线的某一位上，假设接D7位。CPU先读取状态端口，查询D7位是否为1，若是，表示输入数据已准备好，然后读取数据口，取走输入数据，同时将状态标志触发器复位。图2为查询式输入程序流程图。

图示

描述已自动生成

图2 查询方式输入程序流程图

（2）字符转换数值的算法：数字字符与数值的转换，以字符‘1’转化成数值01为例：字符‘1’的ASCII码为31H，所以只要在其基础上减去30H，即可得到对应的数值01H。英文数字与数值的转换，以字符‘A’转换成数值0AH为例，字符‘A’的ASCII码为41H，只要在其基础上减去37H，即可得到对应数值0AH。所以对于一个输入，先进性判断，若为字母，先减去07H，再同数字字符一样减去30H。当然，也可以设置成两个分支直接减去37H。代码设计为每次读入并转换后的单个数值都存在对应区域的第四位，所以当我们已经读入高四位的数据后，想读入低四位的数据时，只需将原有的数据左移4位，再加上地位数据即可。该算法流程图如图3所示。

图示

描述已自动生成

图3 字符转换数值程序流程图

（3）中断子程序接受：利用辅串口的0BH型中断向量实现中断子程序。具体的流程如图4所示。接受主程序时要先关中断并将辅串口初始化，之后置换0BH型中断向量并进行相关的判断。接受主程序流程如图5所示。

图示

描述已自动生成

图4 接受中断服务子程序流程图

图示

描述已自动生成

图5 接受主程序流程图

**四、课题实现**

1 硬件连线

利用RS-232C接口连接两台机器的串行通信接口。

2 程序源代码

......

.586

DATA SEGMENT USE16

MESG DB 128

DB ?

DB 128 DUP(?)

LENS EQU $-MESG

OLD0B DD ?

A1 DB " ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~$"

A2 DB " \*\*\* \* \* \* Transfer \* \* \* \*\*\*\*$"

A3 DB " designer:B20030726 zcy $"

A4 DB " Please input the word:$"

A5 DB " THE output:$"

A6 DB " -----------------------------------$"

A7 DB " $"

A8 DB " ENTER BAUD(like 01 01 and <=01 80 >=00 0C):$";长度45

A9 DB " ENTER FRAME FORMAT(<=3F ):$";长度27

;波特率报错

WRONGB DB " !!! Wrong!ENTER BAUD(like 01 01 and <=01 80 >=00 0C) !!!$"

;帧格式报错

WRONGF DB " !!! Wrong!PlEASE ENTER FRAME FORMAT(<=3F ) !!!$"

LINE DB 1

STRING DB 6,0,6 DUP(0) ;6位BUF,用来存放波特率高位和低位(两个16进制数和一个空格)

STRINGF DB 3,0,3 DUP(0) ;3位，用来存放帧格式

DATA1 DB 2 DUP(0) ;2位data用于存放转化后的波特率

DATAF DB 1 DUP(0) ;1位dataf用于存放转化后的帧格式

DATA ENDS

CODE SEGMENT USE16

ASSUME CS:CODE,DS:DATA

BEG:

;界面样式

;设置外窗口属性

MOV AH,7 ;功能：屏幕初始化

MOV AL,0 ;page

MOV BH,70H ;白底黑字

MOV CH,1 ;row左上

MOV CL,2 ;column左上

MOV DH,23 ;row右下

MOV DL,77 ;column右下

INT 10H ;BIOS显示操作中断

;设置内窗口属性

MOV AH,7 ;功能：屏幕初始化

MOV AL,0 ;page

MOV BH,00111111B ;浅青色底白字

MOV CH,2 ;row左上

MOV CL,4 ;column左上

MOV DH,22 ;row右下

MOV DL,75 ;column右下

INT 10H ;BIOS显示操作中断

;设置光标类型(大小)

MOV CH,12 ;光标起始行

MOV CL,13 ;光标结束行

MOV AH,1 ;设置光标类型

INT 10H ;DOS中断：显示操作

;初始化光标位置

MOV DH,2 ;屏幕显示行

MOV DL,4 ;屏幕显示列

MOV BH,0 ;显示页号

MOV AH,2 ;置光标位置

INT 10H ;BIOS显示操作中断

MOV AX,DATA

MOV DS,AX

MOV AH,0 ;设置显示方式

MOV AL,3 ;80\*25彩色文本方式

INT 10H

MOV AH,6

MOV AL,0

MOV CH,0

MOV CL,0

MOV DH,80

MOV DL,80

MOV BH,01110000B ;前四位定义背景色为棕色

INT 10H

显示“~~~~~~~~~~~~~~~~~~~~~~~~~~~~~”

ADD LINE,2

MOV DH,LINE

MOV DL,1

CALL GUANGBIAO

LEA DX,A1

CALL OUTP

ADD LINE,2

MOV DH,LINE

MOV DL,1

CALL GUANGBIAO

LEA DX,A2

CALL OUTP

ADD LINE,2

MOV DH,LINE

MOV DL,8

CALL GUANGBIAO

LEA DX,A3

CALL OUTP

ADD LINE,2

;输入波特率

;显示绿色底色输入框-波特率(PS,要》=300)

MOV AH,6

MOV AL,0

MOV CH,LINE

MOV CL,47 ;当前行，0列到47列(此处为47列)

MOV DH,LINE

MOV DL,52 ;当前行，47列到52列(此处为52列)

MOV BH,00100001B ;底色显示为绿色

INT 10H

;波特率设置

MOV DH,LINE

MOV DL,1

CALL GUANGBIAO

LEA DX,A8

ADD LINE,1 ;显示"ENTER BAUD(like 01 01H and >=0060H):$"

MOV AH,09H

INT 21H

LEA DX,STRING ;读取STRING字符形式的数据

MOV AH,0AH

INT 21H

;==字符转换成数值

;数字字符与数值的转换（以字符‘1’转化成数值01H为例）：

;字符‘1’的ASCII码为31H，所以只要在其基础上减去30H，即可得到对应的数值。

;==英文字符与数值的转换（以字符‘A’转换成数值0AH为例）：

;字符‘A’的ASCII码为41H，只要在其基础上减去37H，即可得到对应数值0AH。

;上述的代码中，若判断是字母，先减去07H，再同数字字符一样减去30H。当然这里可以设计成两个分支，直接减去37H。

;==对上述代码中逻辑左移部分的解释：上述代码中设计成每次读入并转换

;后的单个数值都存在对应区域的第四位，所以当我们已经读入高四位的数

;据后，想读入低四位的数据时，只需将原有的数据左移4位，再加上低位数据即可。这里也可以设计成乘10H后相加（与左移4位效果相同）。

MOV BX,05H ;目标读取的数据个数

MOV SI,02H ;用来指向STRING

MOV DI,00H ;用来指向DATA1

MOV CL,04H

TRANS1: MOV DL,STRING[SI] ;依次读取STRING中的字符

CMP DL,20H ;空格的ASCII码为20H

JE NEXT1 ;若是空格就跳转到NEXT1

CMP DL,0DH ;回车的ASCII码为0DH

JE NEXT1 ;若是回车就跳转到NEXT1

CMP DL,3AH

JB NEXT0 ;若是字母，则减去07H；若是数字，跳转到NEXT0

SUB DL,07H

NEXT0: SUB DL,30H ;减去30H后由ASCII转为数值

SHL DATA1[DI],CL ;逻辑左移4位后相加

ADD DATA1[DI],DL

INC SI

JMP NEXT2

NEXT1: INC DI

;遇到空格或是回车表示一个数据转换完成，此时DI加一，指向下一个存储数值的位置

INC SI

NEXT2: CMP BX,DI ;判断所有数据是否转换完成

JNE TRANS1

;帧格式显示绿色底色输入框-帧格式

MOV AH,6

MOV AL,0

MOV CH,LINE

MOV CL,30 ;当前行，0列到30列(此处为30列)

MOV DH,LINE

MOV DL,31 ;当前行，30列到31列(此处为31列)

MOV BH,00100001B ;底色显示为绿色

INT 10H

;显示“ENTER FRAME FORMAT(like 03):$”

INPUTF:

MOV DH,LINE

MOV DL,1

CALL GUANGBIAO

LEA DX,A9

ADD LINE,1;显示"ENTER FRAME FORMAT(like 03):$"

MOV AH,09H

INT 21H

LEA DX,STRINGF ;读取STRINGF字符形式的数据

MOV AH,0AH

INT 21H

;字符转换成数值

MOV BX,05H ;目标读取的数据个数

MOV SI,02H ;用来指向STRING

MOV DI,00H ;用来指向DATAF

MOV CL,04H

TRANS1F: MOV DL,STRINGF[SI] ;依次读取STRINGF中的字符

CMP DL,20H ;空格的ASCII码为20H

JE NEXT1F ;若是空格就跳转到NEXT1F

CMP DL,0DH ;回车的ASCII码为0DH

JE NEXT1F ;若是回车就跳转到NEXT1F

CMP DL,3AH

JB NEXT0F ;若是字母，则减去07H；若是数字，跳转到NEXT0

SUB DL,07H

NEXT0F: SUB DL,30H ;减去30H后由ASCII转为数值

SHL DATAF[DI],CL ;逻辑左移4位后相加

ADD DATAF[DI],DL

INC SI

JMP NEXT2F

NEXT1F: INC DI

;遇到空格或是回车表示一个数据转换完成，此时DI加一，指向下一个存储数值的位置

INC SI

NEXT2F: CMP BX,DI ;判断所有数据是否转换完成

JNE TRANS1F

;帧格式输入检验

MOV BX,OFFSET DATAF[1]

CMP WORD PTR [BX],3FH ;输入的帧格式如果超过3FH就报错

JA ERRORF

;显示错误提示信息

ERRORF:;帧格式报错

MOV AH,6

MOV AL,0

MOV CH,LINE

MOV CL,4 ;当前行，23列到30列（输入8位字符）

MOV DH,LINE

MOV DL,60

MOV BH,01000001B ;底色显示为红色

INT 10H

MOV DL,1

CALL GUANGBIAO

LEA DX,WRONGF

CALL OUTP

INC LINE

JMP INPUTF

EXIT1: MOV AH,4CH

INT 21H

;显示绿色底色输入框-input the word

MOV AH,6

MOV AL,0

MOV CH,LINE

MOV CL,26 ;当前行，0列到26列(此处为26列)

MOV DH,LINE

MOV DL,75 ;当前行，26列到75列(此处为75列)

MOV BH,00100001B ;底色显示为绿色

INT 10H

;显示“Please input the word:$”

MOV DH,LINE

MOV DL,1

CALL GUANGBIAO

LEA DX,A4

CALL OUTP

ADD LINE,1

;主程序内容

MOV AX,DATA

MOV DS,AX

CLI ;关中断

MOV AH,0AH

MOV DX,OFFSET MESG

INT 21H

MOV AH,2

MOV DL,0AH

INT 21H

;显示青色底色输出框

MOV AH,6

MOV AL,0

MOV CH,LINE

MOV CL,15 ;当前行，0列到15列(此处为15列)

MOV DH,LINE

MOV DL,75 ;当前行，15列到75列(此处为75列)

MOV BH,11100001B ;底色显示为黄色

INT 10H

;显示“The output”

MOV DH,LINE

MOV DL,1

CALL GUANGBIAO

LEA DX,A5

CALL OUTP

ADD LINE,2

;主程序内容

MOV CH,0

CALL I8250 ;主串口初始化

CALL I8259 ;开放主8259A辅串口中断

CALL RD0B ;读中断向量

CALL WR0B ;写中断向量

STI ;开中断

MOV BX,OFFSET MESG+2

MOV CL,MESG+1

SCANT: MOV DX,2FDH

IN AL,DX

TEST AL,20H

JZ SCANT

MOV DX,2F8H

MOV AL,[BX]

OUT DX,AL

INC BX ;计数

MOV DX,0

TWAIT: DEC DX

JNZ TWAIT ;加循环延时来保证中断接收的时间

LOOP SCANT

RETURN:CALL RESET

MOV AH,4CH

INT 21H ;返回 DOS

;往下是子程序

;光标的函数GUANGBIAO

GUANGBIAO PROC

PUSH AX

PUSH BX

MOV AH,2 ;2号功能调用，预置光标的位置

MOV BH,0

INT 10H ;定光标

POP BX

POP AX

RET

GUANGBIAO ENDP

;界面样式显示输出的函数OUTP

OUTP PROC

PUSH AX

MOV AH,9 ;9号功能调用，字符串输出

INT 21H

POP AX

RET

OUTP ENDP

RECEIVE PROC

PUSHA

PUSH DS

MOV AX,DATA

MOV DS,AX

MOV DX,2F8H

IN AL,DX

AND AL,7FH

MOV AH, 2

MOV DL, AL

INT 21H ;屏幕显示

EXIT: MOV AL,20H ;中断结束命令

OUT 20H,AL

POP DS ;恢复现场

POPA

IRET

RECEIVE ENDP

I8250 PROC ;主串口初始化子程序

MOV DX,2FBH

MOV AL,80H

OUT DX,AL ;寻址位置1

MOV DX,2F9H ;设置波特率，此处00，60是1200波特

MOV AL,DATA1[0]

OUT DX,AL ;除数高8位

MOV DX,2F8H

MOV AL,DATA1[1]

OUT DX,AL ;除数高8位

;帧格式，案例为03H

MOV DX,2FBH

MOV AL,DATAF[0]

OUT DX,AL

MOV DX,2F9H

MOV AL,01H ;允许接收中断

OUT DX,AL

MOV DX,2FCH

MOV AL,18H ;内环，8250能送出中断请求

OUT DX,AL

RET

I8250 ENDP

I8259 PROC

IN AL,21H

AND AL,11110111B

OUT 21H,AL ;置中断屏蔽寄存器

RET

I8259 ENDP

RD0B PROC ;保存原来系统的0BH 中断向量

MOV AX,350BH

INT 21H

MOV WORD PTR OLD0B,BX

MOV WORD PTR OLD0B+2,ES

RET

RD0B ENDP

WR0B PROC ;置换0BH型中断向量指向自定义中断服务程序

PUSH DS

MOV AX,CODE

MOV DS,AX

MOV DX,OFFSET RECEIVE

MOV AX,250BH

INT 21H

POP DS

RET

WR0B ENDP

RESET PROC ;恢复系统0B中断向量

IN AL,21H

OR AL,00001000B

OUT 21H,AL

MOV AX,250BH

MOV DX,WORD PTR OLD0B

MOV DS,WORD PTR OLD0B+2

INT 21H

RET

RESET ENDP

CODE ENDS

END BEG

**五、测试数据及其结果分析**

对算法功能、性能以及健壮性评测所使用的运行以及测试数据进行介绍和说明，并给出与输入数据相匹配的算法执行结果，并进行分析。

用户在输入波特率时，若该数值过小则会引起数据传送时的错误，为了解决这个问题我们对波特率进行范围限制。此外用户在设置帧格式时如果不符合规范也会报，因为帧格式的前两位都必须为0以保证通信的正常。若输入错误就会出现提示并要求重新输入，如图6所示。

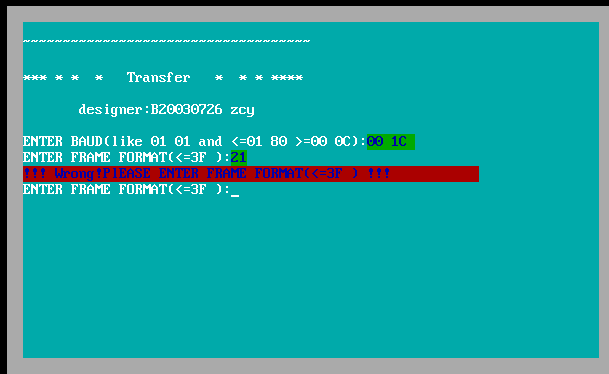


图6 输入数据不合法的例图

当设置波特率位9600波特，帧格式为03H型时，发送数据“123456”，结果如下图所示。

网站, 日程表

中度可信度描述已自动生成

图7 输入合法数据后的正确结果截图

**六、课题完成过程中遇到的问题及解决方法**

问题1：输入字符串后，无法正常显示收到的字符串。接收到的字符串比原来的少且会发生重复。

解决方法：通过一次偶然调整波特率的过程中，我发现了当波特率小于300波特时，就会发生此类问题。于是加上提示字段同时对波特率的取值加上限制，只有输入300-9600波特率时，程序才能正常运行。

问题2：运行程序后，需要用户输入数据，但是运行的结果和理想结果总是不一致。

解决方法：通过询问同学知道了汇编程序输入数字时，程序会将输入的数字转化为相对应的ASCII码值，结果并不是用户所输入的数字。我通过上网查阅，学习了一个有关字符转化的算法，它可以将输入的字符转化为十六进制的数字。

问题3：当初学汇编语言时老师并没有讲解有关界面的相关问题，不知道如何优化界面，搭建出一个优美的背景。

解决方法：上网查询资料，发现可以利用BIOS功能实现界面的布置。查询BIOS命令的使用手册了解各种图形相关的功能搭建出了一个界面。

问题4：由于距离汇编语言的学习已有一个学期，许多基础语法知识都遗忘，如PTR的使用，各种寄存器的寻址方式等等，导致程序刚开始出现大量错误。

解决方法：通过再次回顾ppt以及书本，学习和复习了有关基础知识。

**七、总结600**

为时两周的程序设计周落下帷幕，我的心情也逐渐变得平静，对于这次长达两周的实践周，我的感情是复杂的。从刚开始看题时的

（中文字体：宋体，英文字体：Times New Roman，小四号，不加粗，两端对齐，1.5倍行距，首行缩进2字符，段前段后距为0）

**其他要求：**

1、最后阶段请认真完成本次课程设计报告的电子文档，报告内容不得少于15页。

2、每个学生最后提交的材料：

①课程设计报告的电子文档(起名为：**专业课程设计I报告\_学号.doc)**，单独作为一个文件提交，通过班长、QQ或者E-mail：[liaq@njupt.edu.cn](mailto:liaq@njupt.edu.cn)提交给老师（具体由老师定）；

②压缩文件一个，起名为：**专业课程设计I\_学号.rar(如：专业课程设计I\_B20030701.rar)。**该压缩文件中包含下列几个文件：课程设计报告的电子文档(起名为：**专业课程设计I报告\_学号.doc)**、源程序文件(起名为：**题目名\_学号.asm，如：题目8\_ B20030701.asm)、**可能用到的原始数据文件(如：**stud.dat**)、编译链接后得到的可执行文件(起名为：**题目1\_学号.exe)，通过班长提交。**

③每位同学请打印纸质版，提交给班长，**注意封面是正反两面打印在一页上**。