微机系统与应用课程设计

## 课题一 交通信号灯自动控制模拟指示系统

**一、课程设计目的**

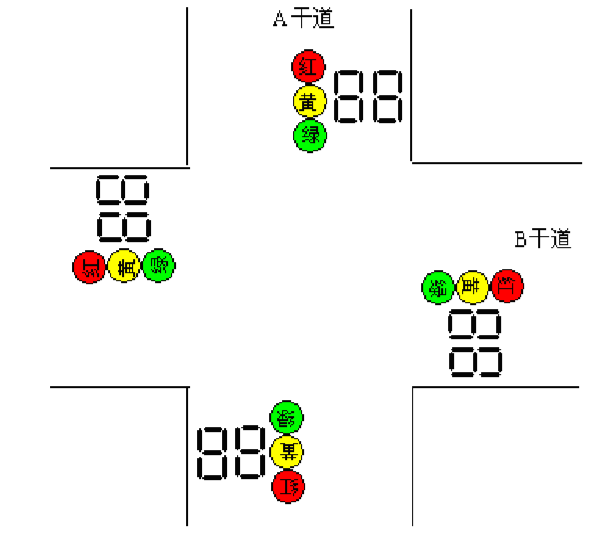
1、掌握CPU与各芯片管脚连接方法，提高接口扩展硬件电路的连接能力。

2、加深对定时器/计数器和并行接口芯片的工作方式和编程方法的理解。

3、掌握交通信号灯自动控制系统的设计思路和实现方法。

**二、课程设计内容**

设计并实现十字路口通信号自动控制模拟指示系统。设该路口由A、B两条通行干道相交而成，四个路口各设一组红、黄、绿三色信号灯，用两位数码管作倒计时显示。



**交通灯模拟控制图**

**三、系统功能与设计要求**

**1、基本功能要求**

（1）以秒为计时单位，两位数码管以十进制递减计数显示通行（绿灯）剩余时间，在递减计数回零瞬间转换。十字路口交通灯的变化规律及控制时序：

① 南北口的绿灯、东西路口的红灯同时亮30秒，同时南北路口数码管递减显示绿灯剩余时间30，29，28……0 秒。

② 南北路口的黄灯闪烁 5 秒钟，同时东西路口的红灯继续亮。

③ 南北路口的红灯、东西路口的绿灯同时亮30秒，同时东西路口数码管递减显示绿灯剩余时间30，29，28……0秒。

④ 南北路口的红灯继续亮，同时东西路口的黄灯闪烁5秒钟。

⑤ 转① 重复。

（2）通过键盘可以对红、黄、绿三色信号灯所亮时间在0～99秒内任意设定。

（3）十字路口的通行起始状态可自行设定，系统启动后自动运行，按“Q”键退出。

**2、发挥部分**

（1）增加人工干预模式。在特殊情况下可通过人工干预，手动控制A、B道路交通灯的切换时间，并可以随时切换为自动运行模式。

（2）增加夜间控制功能，交通灯在进入夜间模式后，A、B两个干道上红、绿灯均不亮，黄色信号灯闪烁显示。

（3）增加两位红色信号灯倒计时显示。使系统同时显示A、B干道的红绿灯（通行/等待）时间。

**四、设计思路**

交通信号灯的亮灭时间及数码管显示时间可以通过计数/定时器（8254）来控制，8254的时钟源采用时钟信号发生器与分频电路提供，通过计算获得计数初值。按照需要设定工作方式。交通信号灯及数码管可以采用系统提供的相应模块，控制可以通过 8255可编程并行接口，如：PA口控制红黄绿交通灯的亮灭，PB口和 PC口控制时间显示数码管的段，位。也可使用基本并行 I/O 接口（74LS244，74LS273）。人工干预及夜间控制可以采用F4区的拨动开关K5~K8进行模拟控制。

**五、操作步骤**

1、提交设计方案

包括：控制功能实现方案，编程语言硬件设计方案及硬件连线图，硬件支持环境和软件支持的环境要求

2、方案审核

系统设计方案经过指导老师审核后，方可以实施操作。

3、硬件系统实现

按照硬件设计方案及硬件连线图连线，经过指导老师认可后，加电并观察是否正确。 如果发现异常，立刻关掉电源，查明原因，解决问题后，再加电观察。

4、按照功能要求编程，运行，调试

**六、 撰写课程设计报告及其内容**

1、应用系统设计方案

2、系统测试结果

3、课程设计中遇到的问题及解决办法

4、写出体会与建议

**七、课程设计实验环境：**

1、硬件配置：

微机一台 （Pentium 4）

微机接口技术实验箱 一个

连接电缆 一条

万用表 一块

微机接口技术实验讲义 一本

连接导线

2、软件环境：

Windows XP/2000/Win 7/Win10平台

TASM、Turbo C 编译器

**八、评分标准**

|  |  |  |
| --- | --- | --- |
|  | 项 目 | 得分  （100） |
| 基本要求 | 设计与总结报告：方案比较、设计与论证，理论分析与计算，电路图及有关设计文件，测试方法、数据及结果分析。 | 20 |
| 实际制作完成情况 | 40 |
| 发挥部分 | 完成第（1）项 | 10 |
| 完成第（2）项 | 10 |
| 完成第（3）项 | 20 |

总分：60 分~70 分 “及格”；70~80 分“中等”； 80~90 分 “良好”；90 分以上“优秀”

**参考设计**

**一、实验原理图**









**二、实验步骤**

1、主机连线说明：

|  |  |  |
| --- | --- | --- |
| D3区：CS、A0、A1 | —— | A3区：CS1、A0、A1 |
| D3区：PC0、PC1 | —— | F5区：KL1、KL2 |
| D3区：JP20、B、C | —— | F5区：A、B、C |
| D3区：PC4、PC5、PC6 | —— | F4区：K5、K6、K7 |
| C4区：CS、A0、A1 | —— | A3区：CS2、A0、A1 |
| C4区：GATE | —— | C1区：VCC |
| C4区：CLK0 | —— | B2区：62.5K |
| C4区：OUT0 | —— | B3区：IR0 |
| B3区：CS、A0 | —— | A3区：CS3、A0 |
| B3区：INT、INTA | —— | A3区：INTR、INTA |
| D5区：CS、A0、A1、A2 | —— | A3区：CS4、A0、A1、A2 |
| D5区：SIN、SOUT | —— | D7区：RXD、TXD |
| B4区：JP57(D0..D7) | —— | A3区：JP42(D0..D7) |
| B4区(I/O)：BLE、BHE | —— | A3区：BLE、BHE |
| B4区(I/O)：RD、WR | —— | A3区：IOR、IOW |
| B4区(I/O)：CS273 | —— | A3区：CS5 |
| B4区(I/O)：JP51(O0..O7) | —— | F4区：JP18 |

D7区通过串口线与微机相连。

K5：东西向强制；K6：南北向强制；K7：晚间模式；低电平有效。

2、K5、K6、K7拨向1，运行程序，观察实验现象

3、K5拨向0，东西向立即变为绿灯，南北向变为红灯，数码管停止显示；

K6拨向0，南北向立即变为绿灯，东西向变为红灯，数码管停止显示；

K7拨向0，进入夜间模式，东西向、南北向黄灯闪烁，数码管停止显示；

**4\*4键盘说明：**

0-9数字键；

A：东西向绿灯秒数；B：绿灯闪烁秒数；C：黄灯秒数

D：南北向绿灯秒数；E：绿灯闪烁秒数；F：黄灯秒数

按A-F键，设置时间常数，再按同一按键，保存设定的时间常数。

程序见文件夹RGLED）

## 课题二 步进电机开环控制系统设计

**一、课程设计目的**

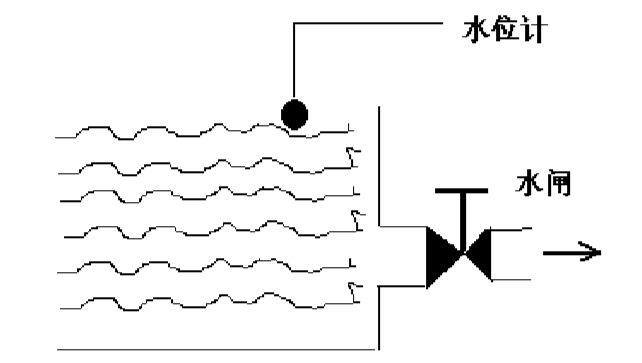
1、掌握微机系统总线与各芯片管脚连接方法，提高接口扩展硬件电路的连接能力。

2、加深对A/D和并行接口芯片的工作方式和编程方法的理解。

3、搞懂步进电机的工作原理及控制方式，掌握开环控制系统的设计思路和实现方法。

**二、课程设计的内容**

手动调节电位器旋钮（0v~5v），通过ADC0809模拟输入水库水位0米~50米,CPU收到水位信号后，根据水位高度控制步进电机（水闸）进行调节。



**水库水位模拟控制示意图**

**三、系统功能与设计要求**

1、基本功能要求

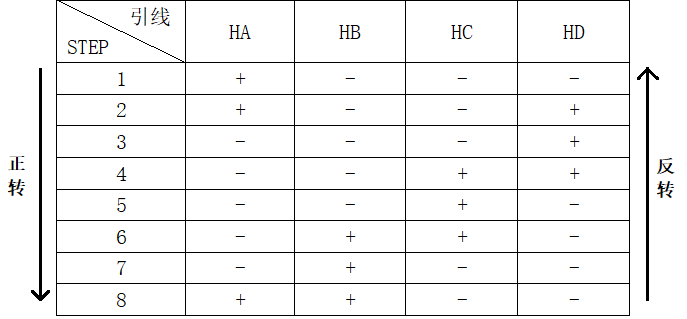
手动调节电位器旋钮，步进电机根据水位实时调节水闸。设水闸全部打开需要逆时针旋 转10圈 （10 x 360°）度。随着上游进入水库的水流量变化，水库水位不断变化（手动调节电位器旋钮），每到一定高度，步进电机顺时针（关）或逆时针（开）旋转一定的角度调

节水闸开启程度，从而控制水库水位在 10~50 米之间。调节精度控制在±5%，调节规律如下：

**水位与水闸控制开启控制表**

|  |  |  |
| --- | --- | --- |
| 水位高度 H（米） | 水闸开启程度（%） | 指示灯 |
| 0≤H< 10 | 0% （0 圈） | 关闸蓄水，水位超低报警（黄 led 闪烁） |
| 10≤H< 20 | 20%（2 圈） | 正常调节（绿 LED 亮） |
| 20≤H< 30 | 40%（4 圈） | 正常调节（绿 LED 亮） |
| 30≤H< 40 | 60%（6 圈） | 正常调节（绿 LED 亮） |
| 40≤H< 50 | 80%（8 圈） | 正常调节（绿 LED 亮） |
| H≥50 | 100%（10 圈） | 开闸泄洪，水位超高报警（红LED闪烁） |

步进电机采用四相八步控制，开关顺序如图



**步进电机控制图**

**注：当实验结束要立即关闭电源，否则一直停留在某一相上会使电机发热。**

**2、发挥部分**

（1）增加速度调节功能。水位在10~40米期间，步进电机中速转动，水位低于10（水位过低）或高于40米（水位过高）时，步进电机高速转动。

（2）增加实时水位显示。用数码管DLED高两位显示当前水位（00~50 米）。

（3）增加水闸开启程度显示。用数码管DLED低两位实时显示水闸开启程度（00~10 圈）。

**四、设计思路**

手动调节电位器旋钮(0~5V)，模拟水库水位变化（0~50米），电位器的输出接入ADC0809的一个输入通道。通过编程将模拟量实时采集并转变为对应的数字量，CPU通过查询数字量可计算出实际的水位。如果水位过低（或过高）则关闭（或打开）水闸，如果在正常范围内，根据调节规律查出相应的水闸开启程度，并与上次开启程度比较，计算出调节步进电机的角度、方向（步进电机的步进角为 xx 度)，然后控制步进电机进行相应转动，调节水闸，从而达到控制水位的目的。LED 指示灯可以通过使用基本并行 I/O 接口（74LS273）控制显示，DLED 数码管可以采用系统提供的相应模块，控制可以通过 8255 可编程并行接口，如：PA口的底四位（PA0~PA3）控制步进电机，PB 口和 PC 口的低四位（PC0~PC3）控制数码管的段、位。

**五、操作步骤**

1、提交设计方案

包括：控制功能实现方案，编程语言硬件设计方案及硬件连线图，硬件支持环境和软件支持的环境要求

2、方案审核

系统设计方案经过指导老师审核后，方可以实施操作。

3、硬件系统实现 按照硬件设计方案及硬件连线图连线，经过指导老师认可后，加电并观察是否正确。 如果发现异常，立刻关掉电源，查明原因，解决问题后，再加电观察。

4、按照功能要求编程，运行，调试

**六、 撰写课程设计报告及其内容**

1、应用系统设计方案

2、系统测试结果

3、课程设计中遇到的问题及解决办法

4、写出体会与建议

**七、课程设计实验环境：**

1、硬件配置：

微机一台 （Pentium 4）

微机接口技术实验箱 一个

连接电缆 一条

万用表 一块

微机接口技术实验讲义 一本

连接导线

2、软件环境：

Windows XP/2000/Win 7/Win 10 平台

TASM、Turbo C 编译器

**八、评分标准**

|  |  |  |
| --- | --- | --- |
|  | 项 目 | 得分  （100） |
| 基本要求 | 设计与总结报告：方案比较、设计与论证，理论分析与计算，电路图及有关设计文件，测试方法、数据及结果分析。 | 20 |
| 实际制作完成情况 | 50 |
| 发挥部分 | 完成第（1）项 | 10 |
| 完成第（2）项 | 10 |
| 完成第（3）项 | 10 |

总分：60 分~70 分 “及格”；70~80 分“中等”； 80~90 分 “良好”；90 分以上“优秀”

**参考设计**

**一、实验原理图**











**二、实验步骤**

1、主机连线说明：

|  |  |  |
| --- | --- | --- |
| A3区：CS1、A3、A3、A3 | —— | F7区：G2A、A、B、C |
| C1区：VCC、GND | —— | F7区：G1、G2B |
| D3区：CS | —— | F7区：Y1 |
| D3区：A0、A1 | —— | A3区：A0、A1 |
| D3区：B、C | —— | F5区：B、C |
| D3区：PC0、PC1、PC2 | —— | F4区：LED1、LED2、LED3 |
| D3区：PC4、PC5、PC6、PC7 | —— | D1区：A、B、C、D |
| C4区：CS | —— | F7区：Y8 |
| C4区：A0、A1 | —— | A3区：A0、A1 |
| C4区：GATE | —— | C1区：VCC |
| C4区：CLK0 | —— | B3区：1M |
| C4区：OUT0 | —— | B3区：IR0 |
| B3区：CS、A0 | —— | A3区：CS3、A0 |
| B3区：INT、INTA | —— | A3区：INTR、INTA |
| D5区：CS、A0、A1、A2 | —— | A3区：CS4、A0、A1、A2 |
| D5区：SIN、SOUT | —— | D7区：RXD、TXD |
| C2区：CS、ADDA、ADDB、ADDC | —— | A3区：CS2、A0、A1、A2 |
| C2区：EOC | —— | B3区：IR7 |
| C2区：CLK | —— | B2区：500K |
| C2区：IN0 | —— | F6区：0-5v |

2、编写、调试、运行程序，调节F6区的电位器，观察实验现象，是否达到设计要求。

完整程序见文件夹WLC）

## 课题三 数字时钟和自动报时系统设计

**一、课程设计目的**

1、了解以80X86访问总线设备的原理，提高接口扩展硬件电路的连接能力。

2、了解数字时钟和语音报时的工作原理，提高应用系统的设计与调试的综合能力

3、加深对定时器/计数器、并行接口芯片和语音芯片的硬件电路及软件编程的工作方式和编程方法的理解。

**二、 课程设计内容**

设计数字时钟，能够动态时间显示，整点语音报时。利用定时器产生精确的时钟源，进行数据的处理后，控制并行接口进行DLED的动态扫描，正确的显示时钟；利用语音芯片ISD1420进行编程录音，录制整点报时音。当系统运行后，数码管显示时分秒，每到整点，喇叭播放报时音。

**三、系统功能与设计要求**

1、基本功能要求

（1）系统显示界面：

a) 在DLED数码管上进行实时刷新显示。

b) 使用8255扫描数码管，显示出时分秒，小时、分钟分别为两位显示，秒使用LED3 的DP显示，0.5秒亮，0.5秒灭。

（2）自动报时要求：

a) 通过编程实现语音录制，播放，可以随时修改报时音内容。

b）报时音以24时制录制，24个报时音以连续地址存放在ISD1420芯片内，录音存放顺序如下表所示。

**报时录音存放地址表**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ISD1420  地址 | 时长  (ms) | 报时音 | ISD1420  地址 | 时长  (ms) | 报时音 | ISD1420  地址 | 时长  (ms) | 报时音 |
| 00 | 50 | 零 | 24 | 50 | 九 | 56 | 80 | 十八 |
| 04 | 50 | 一 | 28 | 50 | 十 | 5C | 80 | 十九 |
| 08 | 50 | 二 | 2C | 80 | 十一 | 62 | 80 | 二十 |
| 0C | 50 | 三 | 32 | 80 | 十二 | 68 | 80 | 二十一 |
| 10 | 50 | 四 | 38 | 80 | 十三 | 6E | 80 | 二十二 |
| 14 | 50 | 五 | 3E | 80 | 十四 | 74 | 80 | 二十三 |
| 18 | 50 | 六 | 44 | 80 | 十五 | 7A | 160 | 点整 |
| 1C | 50 | 七 | 4A | 80 | 十六 | 83 | 400 | 欢迎语 |
| 20 | 50 | 八 | 50 | 80 | 十七 |  |  |  |

2、发挥部分

（1）增加“语音提醒”功能，自行设计提醒时间和播放的语音内容。打开“语音提醒”功能后，数字钟运行到设定的时刻，喇叭发声提醒事先录制的语音。

（2）增加“秒表计时”功能，启动该功能后可以在 DLED 上进行 00.00~99.99 秒计时。

（3）增加“时区转换功”能，根据设定的时区表随时转换为相应时区的时刻。

**四、设计思路**

系统设计可以从数字时钟和自动报时两部分考虑，数字时钟部分可以利用CLOCK分频器提供的信号作为定时/计数器8254的时钟源，通过8254产生0.5 秒精确时钟，监控程序通过查询获取时钟，根据时钟计算出时分秒，然后驱动8位数码管实时显示出时间；自动报时主要完成语音录制和播放功能，可以通过并行接口8255 控制ISD1420的 REC、PLAYE、PLAYL端口，然后编写录音和放音子程序，供监控程序调用。系统可以通过8255可编程并行接口PA口控制录音、播放，PB口和PC口控制时间显示数码管的段、位。基本I/O74LS244作为定时/计数器 8254 时钟源监控端口。

**五、操作步骤**

1、提交设计方案

包括：控制功能实现方案，编程语言，硬件设计方案及硬件连线图，硬件支持环境和软件支持的环境要求

2、方案审核

系统设计方案经过指导老师审核后，方可以实施操作。

3、硬件系统实现

按照硬件设计方案及硬件连线图连线，经过指导老师认可后，加电并观察是否正确。 如果发现异常，立刻关掉电源，查明原因，解决问题后，再加电观察。

4、按照功能要求编程，运行，调试

**六、 撰写课程设计报告及其内容**

1、应用系统设计方案

2、系统测试结果

3、课程设计中遇到的问题及解决办法

4、写出体会与建议

**七、课程设计实验环境：**

1、硬件配置：

微机一台 （Pentium 4）

微机接口技术实验箱 一个

连接电缆 一条

万用表 一块

微机接口技术实验讲义 一本

连接导线

2、软件环境：

Windows XP/2000/Win 7 平台

**八、评分标准**

|  |  |  |
| --- | --- | --- |
|  | 项 目 | 得分  （100） |
| 基本要求 | 设计与总结报告：方案比较、设计与论证，理论分析与计算，电路图及有关设计文件，测试方法、数据及结果分析。 | 20 |
| 实际制作完成情况 | 50 |
| 发挥部分 | 完成第（1）项 | 10 |
| 完成第（2）项 | 10 |
| 完成第（3）项 | 10 |

总分：60 分~70 分 “及格”；70~80 分“中等”； 80~90 分 “良好”；90 分以上“优秀”

**参考设计**

**一、实验原理图**













**二、实验步骤**

1、主机连线说明：

|  |  |  |
| --- | --- | --- |
| D3区：CS、A0、A1 | —— | A3区：CS1、A0、A1 |
| D3区：PC0、PC1 | —— | F5区：KL1、KL2 |
| D3区：JP20、B、C | —— | F5区：A、B、C |
| D3区：PC4、PC5、PC6 | —— | B1区：REC、PLAYE、PLAYL |
| C4区：CS、A0、A1 | —— | A3区：CS2、A0、A1 |
| C4区：GATE | —— | C1区：VCC |
| C4区：CLK0 | —— | B3区：1M |
| C4区：OUT0 | —— | B3区：IR0 |
| B3区：CS、A0 | —— | A3区：CS3、A0 |
| B3区：INT、INTA | —— | A3区：INTR、INTA |
| D5区：CS、A0、A1、A2 | —— | A3区：CS4、A0、A1、A2 |
| D5区：SIN、SOUT | —— | D7区：RXD、TXD |
| B4区：JP57(D0..D7) | —— | A3区：JP42(D0..D7) |
| B4区(I/O)：BLE、BHE | —— | A3区：BLE、BHE |
| B4区(I/O)：RD、WR | —— | A3区：IOR、IOW |
| B4区(I/O)：CS273 | —— | A3区：CS5 |
| B4区(I/O)：JP51(O0..O7) | —— | B1区：JP5 |

2、运行程序，观察实验现象，是否达到设计要求。

4\*4键盘功能定义：

B：放一小段欢迎语音

C：录放音

00-23：0-23语音 24：点整 25：一小段欢迎语音

D：录音 E：放音 F：退出

D：选择时区

0：北京 1：东京 2：巴黎 3：纽约

E：秒表

E:开始、停止 F：退出

F：设置时间

## 课题四 常用函数波形发生器的设计

**一、课程设计目的**

1、掌握微机系统总线与各芯片管脚连接方法，提高接口扩展硬件电路的连接能力。

2、加深对和D/A芯片的工作方式的理解，学会编写程序，使 D/A 转换输出函数的方法。

3、掌握通过A/D转换进行实时数据采集与处理的设计思路和实现方法。

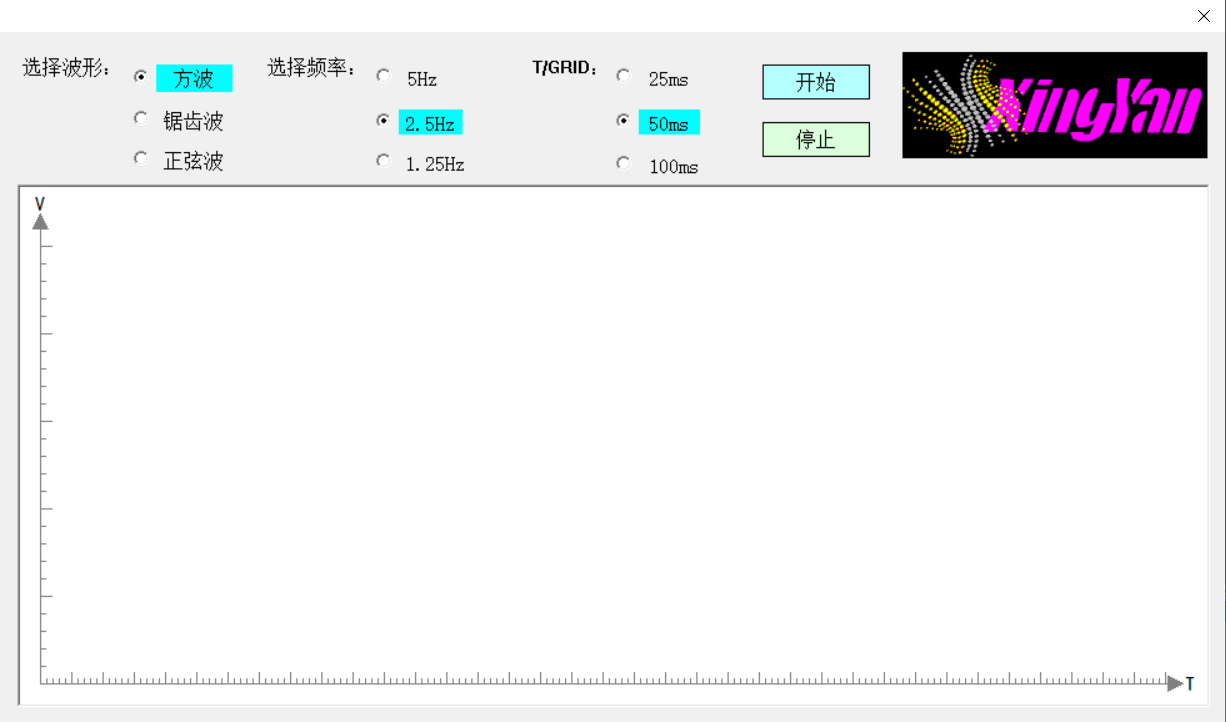
**二、课程设计的内容**

微机通过DAC0832接口（OUT）输出常用函数波形（方波、锯齿波、正弦波等），该波形作为ADC0809模拟信号输入连到IN0，ADC0809的输出送到数据总线，设计连接硬件线路，然后编写波形发生和数据采集程序，将采集的数据（波形）显示在计算机屏幕上。

**三、系统功能与设计要求**

1、基本功能要求

具有友好的人机交互界面，通过鼠标选择相关按钮或菜单对系统进行控制，采集的数据（波形）要以图形的方式实时显示在计算机屏幕上。屏幕显示参考下图：



**常用函数波形发生器屏幕显示图**

屏幕显示参考以上图形，也可以自行设计界面风格，通过按钮（或下拉菜单）选择要产生的波形，点击“开始”后开始产生波形，并进行数据采集，采集到的数据按照图形坐标显示在计算机屏幕上，点击“停止”后立刻停止输出，重新选择波形后，再次点击“开始”即显示新的波形。

2．发挥部分

（1） 增加频率调节功能。使输出到屏幕上的波形随频率值变化。

（2） 增加PWM信号，占空比可调。缺省为50%。

（3） 增加一路信号采集，旋钮电位器的输出端接到ADC0809的IN1，地址信号ADDA、ADDB、ADDC分别接 K6,K7,K8开关，波动开关状态为000或001，分别选择0通道或1信号输出到计算机屏幕上显示。

**四、设计思路**

微机利用软件编程，通过DAC0832数模转换输出端口 OUT2 输出 0~5V 的连续模拟电压（波形），然后将此模拟电压作为ADC0809模数转换的一路输入信号接到IN0 端口，地址信号ADDA、ADDB、ADDC 分别接 K6、K7、K8开关，选择0通道（000），输出的八位数字信号输出到数据总线，控制信号接PB口和PC口，通过采集程序实时读取采集到的数据，并通过RS232传送给微机，最后在计算机屏幕上以图形方式显示出来。

注意事项：DAC0832和 ADC0809的采样频率应小于640KHz。

**五、操作步骤**

1、提交设计方案

包括：控制功能实现方案，编程语言，硬件设计方案及硬件连线图，硬件支持环境和软件支持的环境要求。

2、方案审核

系统设计方案经过指导老师审核后，方可以实施操作。

3、硬件系统实现

按照硬件设计方案及硬件连线图连线，经过指导老师认可后，加电并观察是否正确。 如果发现异常，立刻关掉电源，查明原因，解决问题后，再加电观察。

4、按照功能要求编程，运行，调试

**六、撰写课程设计报告及其内容**

1、应用系统设计方案

2、系统测试结果

3、课程设计中遇到的问题及解决办法

4、写出体会与建议

**七、课程设计实验环境：**

1、硬件配置：

微机一台 （Pentium 4） 轻快PDF阅读器

微机接口技术实验箱 一个

连接电缆 一条

万用表 一块

微机接口技术实验讲义 一本

连接导线

2、软件环境：

Windows XP/2000/Win 7/Win 10平台

TASM、Turbo C 编译器

**八、评分标准**

|  |  |  |
| --- | --- | --- |
|  | 项 目 | 得分  （100） |
| 基本要求 | 设计与总结报告：方案比较、设计与论证，理论分析与计算，电路图及有关设计文件，测试方法、数据及结果分析。 | 20 |
| 实际制作完成情况 | 50 |
| 发挥部分 | 完成第（1）项 | 10 |
| 完成第（2）项 | 10 |
| 完成第（3）项 | 10 |

总分：60 分~70 分 “及格”；70~80 分“中等”； 80~90 分 “良好”；90 分以上“优秀”

**参考设计**

**一、实验原理图**







**二、实验步骤**

1、主机连线说明：

|  |  |  |
| --- | --- | --- |
| C4区：CS、A0、A1 | —— | A3区：CS2、A0、A1 |
| C4区：GATE | —— | C1区：VCC |
| C4区：CLK0 | —— | B3区：1M |
| C4区：OUT0 | —— | B3区：IR0 |
| B3区：CS、A0 | —— | A3区：CS3、A0 |
| B3区：INT、INTA | —— | A3区：INTR、INTA |
| D5区：CS、A0、A1、A2 | —— | A3区：CS4、A0、A1、A2 |
| D5区：SIN、SOUT | —— | D7区：RXD、TXD |
| C2区：CS、ADDA、ADDB、ADDC | —— | A3区：CS1、A0、A1、A2 |
| C2区：CLK | —— | B2区：500K |
| C2区：EOC | —— | B3区：IR7 |
| C2区：IN0 | —— | D2区：OUT |
| A3区：CS5 | —— | D2区：CS |

2、编写、运行程序，观察实验现象，是否达到设计要求。**示程序**（完整程序见文件夹ADDA）

**通信协议**

（1）串口通信波特率38400bps，数据格式为8数据位,1停止位,偶校验

（2）微机发出“开始”命令

|  |  |
| --- | --- |
| **顺序** | **数据** |
| 第一个字节 | 55H |
| 第二个字节 | AAH |
| 第三个字节 | 00H |
| 第四个字节 | 00--方波 01--锯齿波 02--正弦波 |
| 第五个字节 | 01--1ms 02--2ms 04--4ms（AD多少时间采样一次） |
| 第六个字节 | 01--1ms 02--2ms 04--4ms（DA多少时间转换一次） |
| 第七个字节 | 显示一屏，需要多少个数据（低字节） |
| 第八个字节 | 显示一屏，需要多少个数据（高字节） |

（3）微机发出“停止”命令

|  |  |
| --- | --- |
| **顺序** | **数据** |
| 第一个字节 | 55H |
| 第二个字节 | AAH |
| 第三个字节 | 01H |

接收三个字节后，向微机发送00H

（4）X86向微机发送AD、DA数据

|  |  |
| --- | --- |
| **顺序** | **数据** |
| 第一个字节 | 55H |
| 第二个字节 | AAH |
| 第三个字节 | AD转换值 |
| 第四个字节 | DA值 |
| 第五个字节 | AD转换值（下一次） |
| 第六个字节 | DA值（下一次） |
| …… | ……(微机发出“停止命令”，结束数据发送) |

## 课题五 点阵式LCD动态显示系统设计

**一、课程设计目的**

1、了解点阵式液晶显示器的硬件接口电路、控制原理和方法。

2、掌握点阵英文、汉字和图形的字模提取和显示方法。

3、掌握点阵式LCD动态显示程序的设计思路和实现方法。

**二、课程设计的内容**

根据128\*64点阵液晶显示控制器说明书,认真阅读和理解LCD的硬件接口电路、控制原理，并根据ASCII码、汉字码以及图形的提取和显示方法,设计连接硬件线路，然后编写公交车报站监控程序，使汽车运行过程中，液晶显示屏在司机的控制下，按照行车路线用汉字动态显示下一站的站名。

**三、系统功能与设计要求**

1、基本功能要求

公交车报站器的控制键盘参考下图：

|  |  |  |
| --- | --- | --- |
| 上/下行  （发挥1） | 进一站 | 出 站 |
| 广 告  （发挥2） | 退一站 | 进 站 |

**公交车报站控制器示意图**

设某路公交车共有8站（站名自定），车从起点站开出后，按【出站】键，液晶屏幕显示下一站的站名，如“下一站 钟楼”，当行驶到站时，按【进站】键，液晶屏幕显示该站到达，如：“钟楼 到了”，再次按【出站】键，液晶屏幕显示钟楼下一站的站名，如此循环，直到终点站结束。每按一次键，显示一个整屏，显示字体、字号及格式自定，要求美观清晰。 在运行过程中，可以重复按【进一站】或【退一站】键，随时调整当前站的站名，调整后，当再次按【出站】或【进站】键，则从调整后的站名开始继续向下循环显示。

2、发挥部分

（1）增加【上/下行】按键功能，按一次【上/下行】键，系统反方向（下行）显示

站名，再按一次后，系统正方向（上行）显示站名。

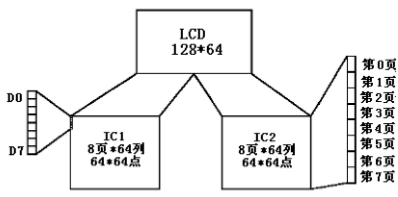
（2）增加插播广告功能，车在运行中，按【广告】键，屏幕显示广告信息，信息中含有自行设计的点阵图案。

（3）当按【出站】键后，从右向左滚动显示下一站的内容，直到按下【进站】键。

**四、设计思路**

公交车报站控制器的显示部分使用128\*64点阵LCD液晶显示，按键采用4X4矩阵键盘模拟。LCD的数据线可以直接与CPU数据总线连接，控制信号CS1/2、RW、RS可以与地址线 A2、A1、A0连接，并连接IOR、IOW信号，通过 8255可编程并行接口的 PA 口和 PC 口控制4X4矩阵键盘，进行动态扫描，测得某键按下后即可执行相关功能，完成显示。

LCD液晶显示器原理图及连接图如下：



**LCD 液晶显示原理图**



**LCD 液晶电路连接图**

ASCII字符的字模可选8x16，每个ASCII字符占用16 字节，汉字字模可选16x16，每个汉字占32字节，图形点阵根据图案自己设计。汉字字模可以采用如ZIMO221、Win-TC等软件取得。

需要注意的是：LCD显示是按列纵向存放的，取模时注意。

**五、操作步骤**

1、提交设计方案

包括：控制功能实现方案，编程语言，硬件设计方案及硬件连线图，硬件支持环境和软件支持的环境要求。

2、方案审核

系统设计方案经过指导老师审核后，方可以实施操作。

3、硬件系统实现 按照硬件设计方案及硬件连线图连线，经过指导老师认可后，加电并观察是否正确。如果发现异常，立刻关掉电源，查明原因，解决问题后，再加电观察。

4、按照功能要求编程，运行，调试

**六、 撰写课程设计报告及其内容**

1、应用系统设计方案

2、系统测试结果

3、课程设计中遇到的问题及解决办法

4、写出体会与建议

**七、课程设计实验环境：**

1、硬件配置：

微机一台 （Pentium 4）

微机接口技术实验箱 一个

连接电缆 一条

万用表 一块

微机接口技术实验讲义 一本

连接导线

2、软件环境：

Windows XP/2000/Win 7/Win10 平台

Visual C++ 6.0 编译器

**八、评分标准**

|  |  |  |
| --- | --- | --- |
|  | 项 目 | 得分  （100） |
| 基本要求 | 设计与总结报告：方案比较、设计与论证，理论分析与计算，电路图及有关设计文件，测试方法、数据及结果分析。 | 20 |
| 实际制作完成情况 | 50 |
| 发挥部分 | 完成第（1）项 | 10 |
| 完成第（2）项 | 10 |
| 完成第（3）项 | 10 |

总分：60 分~70 分 “及格”；70~80 分“中等”； 80~90 分 “良好”；90 分以上“优秀”

**参考设计**

**一、实验原理图**







**二、实验步骤**

1、主机连线说明：

|  |  |  |
| --- | --- | --- |
| D3区：CS、A0、A1 | —— | A3区：CS1、A0、A1 |
| D3区：JP23(PA) | —— | F5区：A |
| C1区：GND | —— | F5区：KL1 |
| A1区：CS、RW、RS、CS1/2 | —— | A3区：CS2、A0、A1、A2 |

2、运行程序，观察实验现象，是否达到设计要求

4\*4键盘：

0：上/下行 1：进一站 2：出站

4：广告 5：退一站 6：进站

## 课题六 键盘电子乐器演奏程序设计

**一、课程设计目的**

1、了解利用8254定时器产生不同音符的原理及方法。

2、加深对计算机键盘扫描程序和执行过程的理解。

3、初步掌握键盘电子乐器演奏程序的设计方法。

**二、课程设计的内容**

设计一个可以通过微机键盘、鼠标演奏不同音乐的控制系统，系统通过按下微机键盘或鼠标点击琴键模仿电子琴键驱动实验箱上的蜂鸣器发声，实现演奏音乐的功能。

**三、系统功能与设计要求**

1、基本功能要求

能够演奏C调包含高、中、低音的不同节拍的乐曲，音调与频率的对应关系如下：

中 音：1--523Hz, 2--575Hz, 3--659Hz, 4--698Hz, 5--784Hz, 6--880Hz, 7--988Hz

低8度音：基本音频率/2,例如低音1为523/2=261.5Hz

高8度音：基本音频率x2，例如高音1为523x2=1046Hz

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **低音音符** | **频率（Hz）** | **中音音符** | **频率（Hz）** | **高音音符** | **频率（Hz）** |
| 1DO | 131 | DO | 262 | DO1 | 523 |
| 1DOr | 139 | DOr | 277 | DO1r | 554 |
| 1RE | 147 | RE | 294 | RE1 | 579 |
| 1REr | 155 | REr | 311 | RE1r | 621 |
| 1MI | 165 | MI | 330 | MI1 | 651 |
| 1FA | 175 | FA | 349 | FA1 | 695 |
| 1FAr | 185 | FAr | 370 | FA1r | 740 |
| 1SOL | 196 | SOL | 392 | SOL1 | 784 |
| 1SOLr | 207 | SOLr | 416 | SOL1r | 830 |
| 1LA | 220 | LA | 440 | LA1 | 880 |
| 1LAr | 233 | LAr | 466 | LA1r | 932 |
| 1SI | 247 | SI | 492 | SI1 | 983 |

每个音的长短由音符的节拍来控制。在一张乐谱中，经常会看到：1=C 3/4、1=G 4/4等，这里1=C、1=G表示乐谱的曲调，它是用来确定基音的；3/4、4/4就是用来表示节拍的。以3/4为例，它表示乐谱中以四分音符为节拍，每一节拍有三拍。

通常一拍的时长为400-500毫秒。以一拍的时长为400为例，当四分音符为节拍时，四分音符的时长就为400毫秒，八分音符的时长就为200毫秒，16分音符的时长为100毫秒。

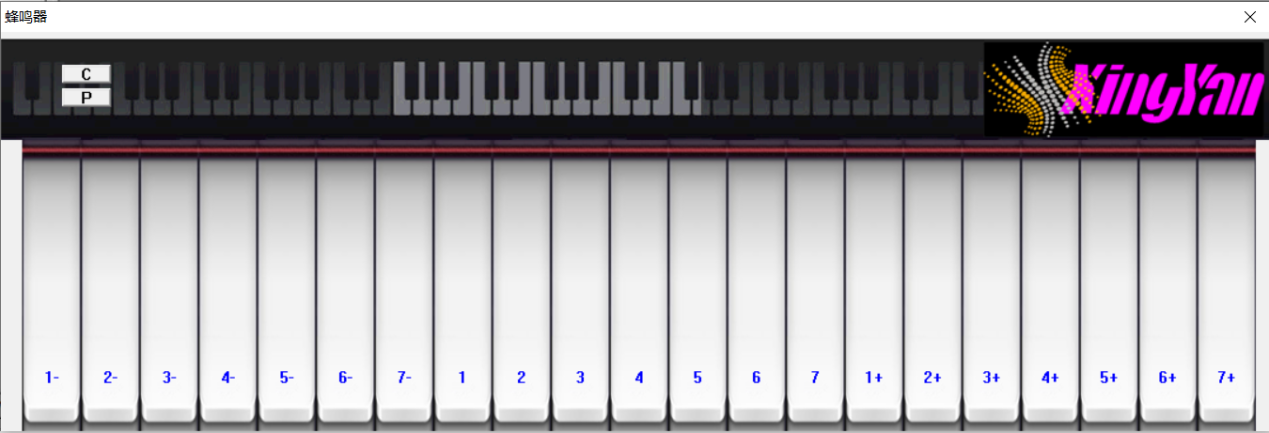
微机键盘与电子琴键盘的对应关系为：高音依次为Q、W、E、R、T、Y、U，中音依次为 A、S、D、F、G、H、J，低音依次为 Z、X、C、V、B、N、M。

2、发挥部分

（1）增加自动演奏乐曲功能。当按下P键后，自动播放一首乐曲，曲长大于20秒。

（2）增加琴键图形显示功能。屏幕上显示 21个琴键（高中低音），当按下某键后屏幕上的对应琴键有所反映。

（3）增加一个音调D，当从C调改为D调后，演奏音按照D调(比 C高一个调)发音。



**四、设计思路**

一首乐曲由若干音符组成，一个音符对应一个频率，将与一个频率对应的计数初值写入计数器就可以产生相应的频率，计算公式如下：

计数初值 = CLK0(输入频率)/OUT0（输出频率）

当音符对应的频率确定后，还需要知道每个音符演奏的时间。我们知道，音符的演奏时间是由节拍控制的，分为一拍、半拍、1/4 拍、1/8……，如果在一首乐曲中，音符演奏的最短时间是1/8拍，我们就将1/8拍作为一个最短时间单位1，那么1/4 拍单位时间就是2，1/2拍就是4，一拍就是8，假定一拍的时间为1s,那么 1/2、1/4、1/8拍的时间就是0.5s、0.25s、0.125s。

音符的演奏时间 = 单位时间 × N ( N为调式参数，影响音乐的节奏)

编程首先应该对计数器初始化，然后扫描键盘，根据扫描结果选择对应的频率，输出到实验箱的音频电路输入端口，并调用延时子程序控制节拍。

**五、操作步骤**

1、提交设计方案

包括：控制功能实现方案，编程语言，硬件设计方案及硬件连线图，硬件支持环境和软件支持的环境要求。

2、方案审核

系统设计方案经过指导老师审核后，方可以实施操作。

3、硬件系统实现

按照硬件设计方案及硬件连线图连线，经过指导老师认可后，加电并观察是否正确。如果发现异常，立刻关掉电源，查明原因，解决问题后，再加电观察。

4、按照功能要求编程，运行，调试

**六、 撰写课程设计报告及其内容**

1、应用系统设计方案

2、系统测试结果

3、课程设计中遇到的问题及解决办法

4、写出体会与建议

**七、课程设计实验环境：**

1、硬件配置：

微机一台 （Pentium 4）

微机接口技术实验箱 一个

连接电缆 一条

万用表 一块

微机接口技术实验讲义 一本

连接导线

2、软件环境：

Windows XP/2000/Win 7/Win 10平台

TASM、Turbo C 编译器

**八、评分标准**

|  |  |  |
| --- | --- | --- |
|  | 项 目 | 得分  （100） |
| 基本要求 | 设计与总结报告：方案比较、设计与论证，理论分析与计算，电路图及有关设计文件，测试方法、数据及结果分析。 | 20 |
| 实际制作完成情况 | 50 |
| 发挥部分 | 完成第（1）项 | 10 |
| 完成第（2）项 | 10 |
| 完成第（3）项 | 10 |

总分：60 分~70 分 “及格”；70~80 分“中等”； 80~90 分 “良好”；90 分以上“优秀”

**参考设计**

**一、实验原理图**





**二、实验步骤**

1、主机连线说明：

|  |  |  |
| --- | --- | --- |
| C4区：CS、A0、A1 | —— | A3区：CS2、A0、A1 |
| C4区：GATE | —— | C1区：VCC |
| C4区：CLK0、CLK1 | —— | B3区：1M、1M |
| C4区：OUT0 | —— | F8区：Ctrl |
| C4区：OUT1 | —— | B3区：IR0 |
| B3区：CS、A0 | —— | A3区：CS3、A0 |
| B3区：INT、INTA | —— | A3区：INTR、INTA |
| D5区：CS、A0、A1、A2 | —— | A3区：CS4、A0、A1、A2 |
| D5区：SIN、SOUT | —— | D7区：RXD、TXD |

2、编写、运行程序，观察实验现象，是否达到设计要求**示程序**（完整程序见文件夹Buzzer）

**通信协议**

（1）串口通信波特率4800bps，数据格式为8数据位,1停止位,偶校验

（2）向微机发出一组数据

|  |  |
| --- | --- |
| **顺序** | **数据** |
| 第一个字节 | 55H |
| 第二个字节 | AAH |
| 第三个字节 | 00H：琴键恢复（自动演奏结束）  01-21：对应琴键1-..7+（自动演奏时该键按下）  ‘R’：向上位机软件发出复位命令 |

（3）微机向X86发出一组数据

|  |  |
| --- | --- |
| **顺序** | **数据** |
| 第一个字节 | 55H |
| 第二个字节 | AAH |
| 第三个字节 | 'P'：自动演奏一曲  'D'：D调  'C'：C调  'R'：复位  1-21：对应琴键1-..7+（根据音调，发出该琴键对应的声音） |

## 课题七 直流电机转速测量与控制系统设计

**一、课程设计目的**

1、掌握CPU与各芯片管脚连接方法，提高接口扩展硬件电路的连接能力。

2、了解光电开关的工作原理，电机转速的测量与控制的基本原理。

3、了解闭环控制系统的组成原理。掌握电机转速测量与控制系统的构成方法。

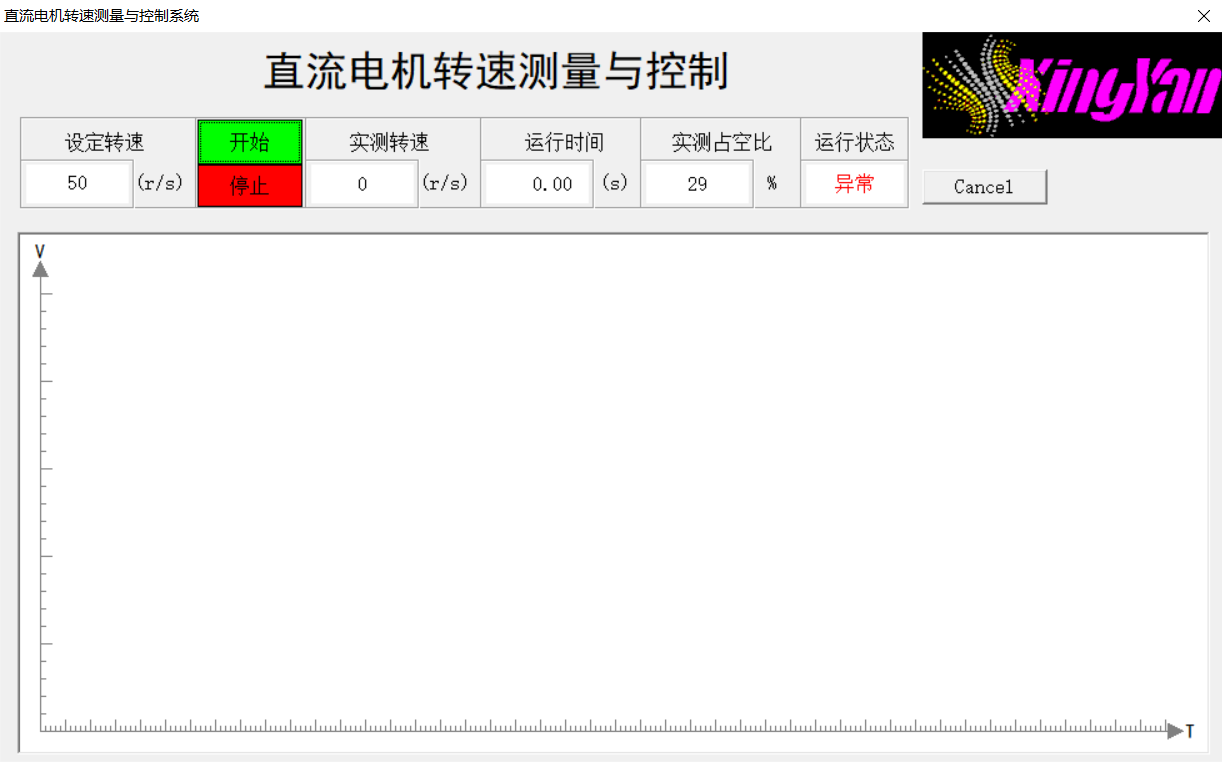
**二、课程设计的内容**

设计一个对直流电机转速测量与转速控制的闭环控制系统。微机控制中心在监控界面上设置电机转速。电机转速测量利用霍尔传感器电路产生转速脉冲，定时/计数电路通过脉冲计数获得转速参量。电机转速调整采用PWM（脉宽调节）方法，控制中心采样到电机转速参量，算得转速值同预定转速设置值进行比较，若不相同，则调整控制转速脉冲的占空比，来达到调速的目的。（占空比=脉冲宽度/脉冲周期）

三、系统功能与设计要求

1、基本功能要求

具有友好的人机交互界面，通过鼠标或键盘，选择相关按钮对系统进行监控，监控系统具有转速参数设置窗口、采样的电机转速数据显示窗口和强行干预系统运行的按钮或相应功能选择菜单。监控程序用查询（或中断）方式获取转速数据。



**直流电机转速测量与控制监控界面图**

屏幕监控界面参考上面界面图，也可以自行设计界面风格。首先自行设定转速（小于

50r/s）,点击“启动”后电机开始运转，同时进行数据采集，采集到的数据（实测转速和运行时间）实时显示在相关位置，点击“停止”后立刻停止电机转动，同时停止采集数据的输出，重新设定转速波形后，再次点击“启动”即显示新数据，测试完成后，填写下表：

**直流电机转速控制表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 设定转速 | 40r/s | 35r/s | 30r/s | 20r/s | 10r/s | 5r/s |
| 实测转速 |  |  |  |  |  |  |
| 调整时间 |  |  |  |  |  |  |
| 误 差 |  |  |  |  |  |  |
| 实际观察情况如：调节快慢，波动大小等 |  |  |  |  |  |  |

2、发挥部分

（1）增加实测占空比显示功能，实时显示占空比的调节变化数据（0% - 100%）。

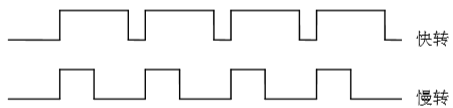
（2）增加电机运行状态显示，当电机转速进入相对稳定状态后，实测转速与设定转速误差在 10%内显示“正常”，否则显示“异常”。

（3）增加实时图形数据显示功能，该功能打开后，可以在屏幕上用图形的方式显示电机的设定转速，实测转速，运行时间等信息,直观看到系统运行情况。

**四、设计思路**

用数字信号来控制模拟量的应用很多，PWM就是其中一种方法。直流电机的转速可以由加到（CTRL）口的脉冲占空比决定，正向占空比越大则转速越快，反之则越慢，其原理图

如下：



控制电机转速的PWM 脉冲

转速测量原理

|  |  |  |
| --- | --- | --- |
| G205_2 | G205_1 | 2 |
| 图1 强反射 | 图2 弱反射 | 图3转盘 |

本转速测量实验采用反射式光电开关，通过计数转盘通断光电开关产生的脉冲，计算出转速

(1) 反射式光开关工作原理：光电开关发射光，射到测量物体上，如果强反射，如图1，光电开关接收到反射回来的光，则产生高电平1；弱反射，如图2，光电开关接收不到反射回来的光，则产生弱电平0。

(2) 转速测量方法：转速测量用的转盘在下表面做成如图3样子的转盘，白部分为强反射区，黑部分为弱反射区，转盘每转一圈，产生4个脉冲，每1/4秒计数出脉冲数，即得到每秒的转速。



**直流电机连接电路图**

利用实验系统资源可以设计一个直流电机转速闭环控制系统。利用带锁存的I/O接口电路(如 8255,74LS273，D/A-DA0832)输出控制电机转速的脉冲。采样转速用光电开关提供电机转速脉冲。利用定时/计数电路对电机转速脉冲计数。微机可从定时/计数电路中获得电机转速数值，并产生控制电机转速的PWM脉冲。

**五、操作步骤**

1、提交设计方案

包括：控制功能实现方案，编程语言，硬件设计方案及硬件连线图，硬件支持环境和软件支持的环境要求。

2、方案审核

系统设计方案经过指导老师审核后，方可以实施操作。

3、硬件系统实现 按照硬件设计方案及硬件连线图连线，经过指导老师认可后，加电并观察是否正确。如果发现异常，立刻关掉电源，查明原因，解决问题后，再加电观察。

4、按照功能要求编程，运行，调试

**六、 撰写课程设计报告及其内容**

1、应用系统设计方案

2、系统测试结果

3、课程设计中遇到的问题及解决办法

4、写出体会与建议

**七、课程设计实验环境：**

1、硬件配置：

微机一台 （Pentium 4）

微机接口技术实验箱 一个

连接电缆 一条

万用表 一块

微机接口技术实验讲义 一本

连接导线

2、软件环境：

Windows XP/2000/Win 7/Win 10平台

TASM、Turbo C 编译器

**八、评分标准**

|  |  |  |
| --- | --- | --- |
|  | 项 目 | 得分  （100） |
| 基本要求 | 设计与总结报告：方案比较、设计与论证，理论分析与计算，电路图及有关设计文件，测试方法、数据及结果分析。 | 20 |
| 实际制作完成情况 | 50 |
| 发挥部分 | 完成第（1）项 | 10 |
| 完成第（2）项 | 10 |
| 完成第（3）项 | 10 |

总分：60 分~70 分 “及格”；70~80 分“中等”； 80~90 分 “良好”；90 分以上“优秀”

**参考设计**

**一、实验原理图**











**二、实验步骤**

1、主机连线说明：

|  |  |  |
| --- | --- | --- |
| D3区：CS、A0、A1 | —— | A3区：CS1、A0、A1 |
| D3区：PC0、PC1 | —— | F5区：KL1、KL2 |
| D3区：JP20、B、C | —— | F5区：A、B、C |
| C4区：CS、A0、A1 | —— | A3区：CS2、A0、A1 |
| C4区：GATE | —— | C1区：VCC |
| C4区：CLK0 | —— | B3区：31250 |
| C4区：OUT0 | —— | B3区：IR0 |
| C4区：CLK1 | —— | B3区：1M |
| B3区：CS、A0 | —— | A3区：CS3、A0 |
| B3区：INT、INTA | —— | A3区：INTR、INTA |
| D5区：CS、A0、A1、A2 | —— | A3区：CS4、A0、A1、A2 |
| D5区：SIN、SOUT | —— | D7区：RXD、TXD |
| D2区：CS | —— | A3区：CS5 |
| D2区：OUT | —— | F3区：IN1 |
| E1区：CTRL | —— | F3区：OUT1 |
| E1区：REV | —— | B3区：IR7 |

2、编写、运行程序，观察实验现象，是否达到设计要求**示程序**（完整程序见文件夹Motor）

**通信协议**

（1）串口通信波特率4800bps，数据格式为8数据位,1停止位,偶校验

（2）微机发出“开始”命令

|  |  |
| --- | --- |
| **顺序** | **数据** |
| 第一个字节 | 55H |
| 第二个字节 | AAH |
| 第三个字节 | 00H |
| 第四个字节 | XX—设定转速 |

（3）微机发出“停止”命令

|  |  |
| --- | --- |
| **顺序** | **数据** |
| 第一个字节 | 55H |
| 第二个字节 | AAH |
| 第三个字节 | 01H |

接收三个字节后，向微机发送00H

（4）X86向微机发送转速等数据

|  |  |
| --- | --- |
| **顺序** | **数据** |
| 第一个字节 | 55H |
| 第二个字节 | AAH |
| 第三个字节 | 当前转速 |
| 第四个字节 | DA值 |

## 课题八 矩阵式键盘数字密码锁设计

**一、课程设计目的**

1、掌握微机系统总线与各芯片管脚连接方法，提高接口扩展硬件电路的连接能力。

2、初步掌握键盘扫描，密码修改和计时报警程序的编写方法。

3、掌握通过矩阵式键盘扫描实现密码锁功能的设计思路和实现方法。

**二、课程设计的内容**

根据设定好的密码，采用4x4矩阵键盘实现密码的输入功能。当密码输入正确之后，锁就打开（绿灯亮），10秒之后，锁自动关闭（红灯亮）；如果连续输入三次密码不正确，就锁定按键5秒钟，同时发出报警(黄灯闪)，5秒后，解除按键锁定，恢复密码输入。

**三、系统功能与设计要求**

1、基本功能要求

1）具有开锁、修改用户密码等基本的密码锁功能。

2）对于超过3次密码密码错误，锁定键盘5秒，系统报警。5秒后解除锁定。

3）通过LCD字符液晶和LED指示灯（红、绿、黄）实时显示相关信息。

4）用户密码为6位数字，显示采用“\*”号表示。

5）密码锁键盘设计合理，功能完善，方便用户使用。



**数字密码锁操作键盘**

数字密码锁操作键盘参考上面设定，也可以自行设计键盘。用户初始密码为“123456”，系统加电运行后，密码锁初始状态为常闭（红灯亮），用户可以选择开锁或修改密码。

如果选择开锁就按“Open”键，系统提示输入密码，输入用户密码+“#”键后，如果密码正确，就打开锁（绿灯亮），系统等待10秒，然后重新关闭密码锁，若密码错，提示重新输入，连续三次错误，提示警告词同时报警（黄灯闪），锁定键盘5秒，然后重新进入初始状态。

如果选择修改密码就按”Modify Secret”键，系统提示输入旧密码，输入旧密码+“#”键后，如果正确，系统提示输入新密码，输入新密码+“#”后，新密码起效，重新进入初始 状态；如果旧密码错，不能修改密码，密码锁直接进入初始状态。

2、发挥部分

（1）增加管理员（Admin）功能，如果用户忘记密码可向管理员求助。管理员密码为8位数字（系统内预先已设定），管理员按“Admin”键后，系统提示输入管理员密码，输入管理员密码+“#”键后，如果正确，系统自动恢复用户初始密码为“123456”；如果错误，程序停止运行,系统退出。

（2）增加组合键（Shift）功能，使系统的密码选择范围除数字外，另外增加 a、b、c、d、e、f、g、h、i、j等10个字母。数字直接输入，字母“Shift”+“字母”。

（3）增加输入清除功能（Delete），在输入密码过程中，如果中途按下“Del”键，则清除前面的输入，开始重新输入密码。

**四、设计思路**

延时时间可以通过计数/定时器（8253）来控制。8253的时钟源采用时钟信号发生器与分频电路提供，通过计算获得计数初值，按照设定的工作方式获得精确的延时时间。密码锁 键盘采用4x4矩阵键盘，液晶显示LCD和LED灯可以采用系统提供的相应模块。控制可以通过8255可编程并行接口，如：PB 口和PC口低二位控制键盘扫描，PC 口高四位控制LED灯，也可使用基本并行I/O接口（74LS245，74LS273）。编程通过键盘扫描实时处理键盘输入数据，完成密码锁相关操作。

**五、操作步骤**

1、提交设计方案

包括：控制功能实现方案，编程语言，硬件设计方案及硬件连线图，硬件支持环境和软件支持的环境要求。

2、方案审核

系统设计方案经过指导老师审核后，方可以实施操作。

3、硬件系统实现

按照硬件设计方案及硬件连线图连线，经过指导老师认可后，加电并观察是否正确。如果发现异常，立刻关掉电源，查明原因，解决问题后，再加电观察。

4、按照功能要求编程，运行，调试

**六、 撰写课程设计报告及其内容**

1、应用系统设计方案

2、系统测试结果

3、课程设计中遇到的问题及解决办法

4、写出体会与建议

**七、课程设计实验环境：**

1、硬件配置：

微机一台 （Pentium 4）

微机接口技术实验箱 一个

连接电缆 一条

万用表 一块

微机接口技术实验讲义 一本

连接导线

2、软件环境：

Windows XP/2000/Win 7/Win 10平台

TASM、Turbo C 编译器

**八、评分标准**

|  |  |  |
| --- | --- | --- |
|  | 项 目 | 得分  （100） |
| 基本要求 | 设计与总结报告：方案比较、设计与论证，理论分析与计算，电路图及有关设计文件，测试方法、数据及结果分析。 | 20 |
| 实际制作完成情况 | 50 |
| 发挥部分 | 完成第（1）项 | 10 |
| 完成第（2）项 | 10 |
| 完成第（3）项 | 10 |

总分：60 分~70 分 “及格”；70~80 分“中等”； 80~90 分 “良好”；90 分以上“优秀”

**参考设计**

**一、实验原理图**











**二、实验步骤**

1、主机连线说明：

|  |  |  |
| --- | --- | --- |
| A3区：CS1、A3、A3、A3 | —— | F7区：G2A、A、B、C |
| C1区：VCC、GND | —— | F7区：G1、G2B |
| D3区：CS | —— | F7区：Y1 |
| D3区：A0、A1 | —— | A3区：A0、A1 |
| D3区：JP20(PB) | —— | F5区：A |
| D3区：PC0、PC1 | —— | F5区：KL1、KL2 |
| C4区：CS | —— | F7区：Y8 |
| C4区：A0、A1 | —— | A3区：A0、A1 |
| C4区：GATE | —— | C1区：VCC |
| C4区：CLK0 | —— | B3区：1M |
| C4区：OUT0 | —— | B3区：IR0 |
| B3区：CS、A0 | —— | A3区：CS3、A0 |
| B3区：INT、INTA | —— | A3区：INTR、INTA |
| A1区：CS、RW、RS、CS1/2 | —— | A3区：CS2、A0、A1、A2 |
| D5区：CS、A0、A1、A2 | —— | A3区：CS4、A0、A1、A2 |
| D5区：SIN、SOUT | —— | D7区：RXD、TXD |
| D3区：PC4、PC5、PC6 | —— | F4区：LED1、LED2、LED3 |

2、编写、运行程序，观察实验现象，是否达到设计要求

**通信协议**

（1）串口通信波特率4800bps，数据格式为8数据位,1停止位,偶校验

（2）微机向X86发送数据

|  |  |
| --- | --- |
| **顺序** | **数据** |
| 第一个字节 | 55H |
| 第二个字节 | AAH |
| 第三个字节 | 00H-09H：数字键  0AH：“Modify Secret”键  0BH：“OPEN”键  0CH：“Admin”键  0DH：“#”键  0EH：“DEL”键  ‘a’..‘i’:字母键  20H-绿色LED亮 21H-黄色LED亮 22H-红色LED亮 |

（3）X86向微机发送数据

|  |  |
| --- | --- |
| **顺序** | **数据** |
| 第一个字节 | 55H |
| 第二个字节 | AAH |
| 第三个字节 | 00H-09H：数字键  0AH：“Modify Secret”键  0BH：“OPEN”键  0CH：“Admin”键  0DH：“#”键  0EH：“DEL”键  ‘a’..‘i’:字母键  20H-绿色LED亮 21H-黄色LED亮 22H-红色LED亮 |