Microcomputer-Design-

微机原理课设

basic_program.asm

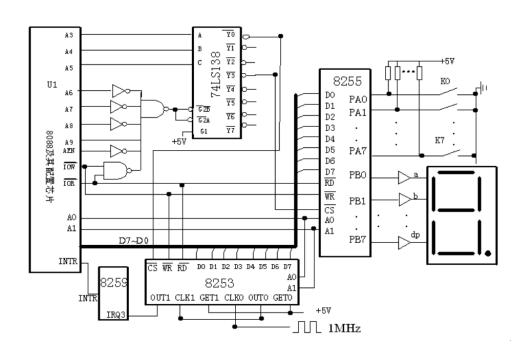
题目描述

通过A端口读入按键值,用数码管轮流显示按键值的高四位和低四位,具体方式为: 高四位显示 1 秒, 低四位显示 0.5 秒。 如此反复循环显示, 如果有键盘按键输入, 则退出程序并返回操作系统。

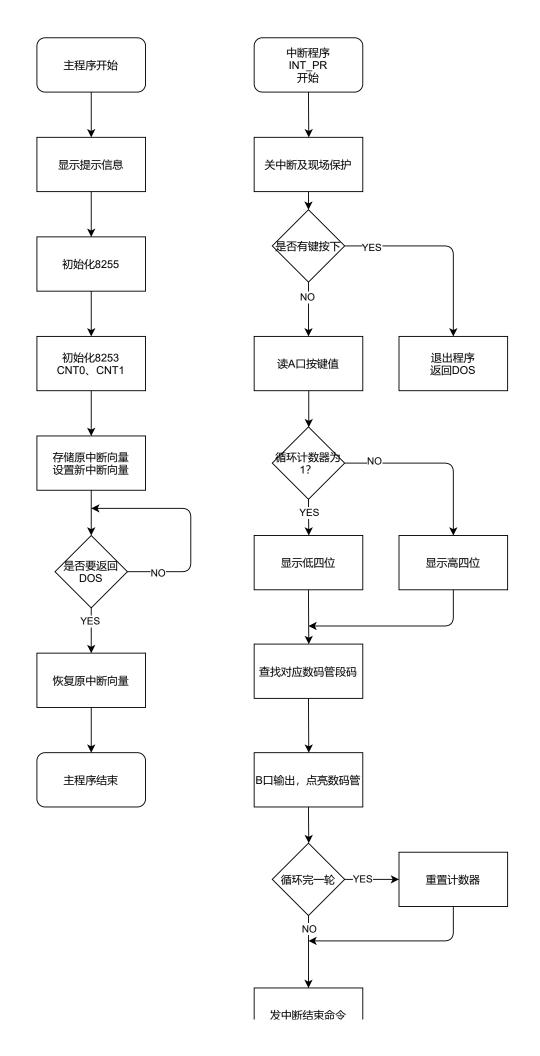
设计思路

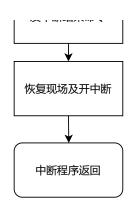
- 1. 程序开始时, 先进行硬件的的初始化设置, 以及中断的相关设置。
- 2. 8253每0.5s输出一个高电平,触发一次中断。设置一个在1-3之间变化的CH用作计数器,用以区分高位显示1s与低位显示0.5s。
- 3. 进入中断程序后,检测是否有键按下,若按下则恢复中断设置,返回DOS。之后读8255PA按键值,确定该显示高位或低位,查表得到数码管段码,在PB输出。
- 4. 不执行中断程序时,循环等待中断。

硬件接线图



流程图





遇到问题及解决方案

1. 问题:中断子程序只能进入一次。

原因: 未发送中断结束命令。

解决:中断退出前,向20H地址写入20H,发结束中断命令。

2. 问题:数码管显示为乱码。

原因:进入中断程序后,数据段段基址发生改变,未能查表成功。

解决:进入中断后,把DS赋值为0FE6H,与主程序一致。

extend_program_1.asm

题目描述

通过实验台上 PS2 接口连接键盘, 要求:

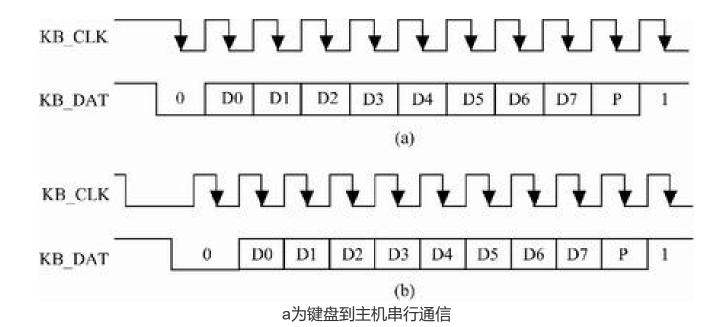
1)输入字符(0~F), 转换成十进制后用数码管(1、2)显示(00~15);

2)用实验台上的数码管 (3、4) 显示输入的次数, 计数范围 0~99。

设计思路

- 1. 主程序开始时, 先进行硬件的的初始化设置, 之后循环进行键盘的读取、转换和数码管显示。
- 2. 编写子函数读取PS2键盘按键编码, 存入数据段。
- 3. 编写子函数将键盘编码转换为按键值(0~F)。
- 4. 编写子函数将16进制数转换为10进制数的十位和个位。具有通用性,字符值和按键次数均可转换。
- 5. 编写子函数进行数码管显示。具有通用性,可指定控制第几个数码管。

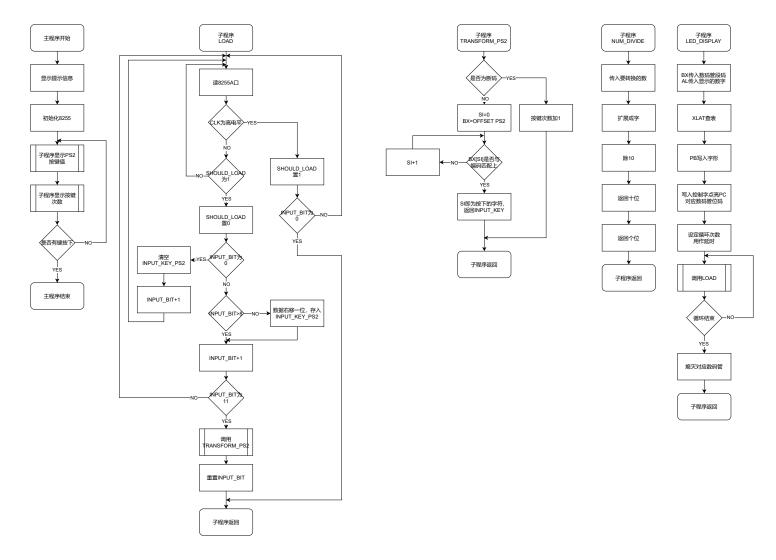
PS2接口时序图



硬件接线表

- 1. 8255
 - 1. PA0----PS2:DATA
 - 2. PA4----PS2:CLK
 - 3. PB----数码管段码
 - 4. PC0-PC3----数码管位码
 - 5. CS----138:0290H-0293H
- 2. PS2
 - 1. CLK----8255:PA4
 - 2. DATA----8255:PA0

流程图



遇到问题及解决方案

1. **问题**:通过PS2接口的CLK触发中断,在中断程序内接收字符编码,接收失败。

原因: CLK跳变太快,8259不能保证每有一个上升沿就进入一次中断。

解决:放弃使用中断,编写普通子程序捕捉CLK的跳变。特别设计对每次传输第一个下降沿的检测:如果没检测到,则直接退出LOAD子程序;如果检测到,就一直在LOAD子程序内循环检测,直到读取完本次传输的所有信息,才退出LOAD子程序。这样可以避免因被数码管显示程序打断而错过脉冲。

2. **问题**:增加数码管显示后,读入的PS2键盘编码有一定几率会出错。

原因:程序分为读键盘和数码管显示两部分,单独执行都不会出错。但数码管显示程序中存在延时,若第一个脉冲来时刚好在延时,则不能被接收,CLK时序被打乱。

解决:把数码管显示的延时程序用多次执行LOAD子程序替代,这样在一次循环中仅有少数指令是用来控制数码管显示的,执行时间很短,不会超过一个CLK脉冲的时间,总能检测到CLK第一个下降沿。

3. 问题:按着按键不抬起,按键次数会一直增加。

原因:按着按键不抬起时,键盘会以一定间隔连续发送编码。

解决:通过检测抬手动作增加按键次数。抬手时,会发送F0和按键编码,收到F0的次数即为按键次数。

extend_program_4.asm

题目描述

利用 8253、8255 和中断系统实现以下功能:

- 1) 使液晶屏实现秒表功能,显示数字 00.00~10.00,每隔 0.01 秒数字变化一次;
- 2) 读入8个开关的状态值,将这两位数字以16进制 (范围 00H~FFH) 在 16x16LED点阵模块上显示。

设计思路

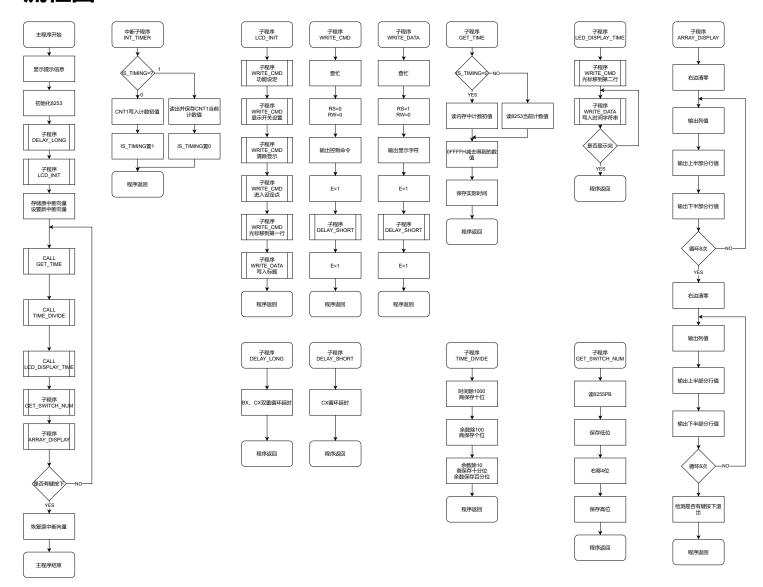
- 1. 在主程序中,需要进行硬件的初始化设置(8253、LCD)、中断向量和中断屏蔽字的更改,之后循 环调用子程序实现功能。第一部分为读取8253时间,将时间换算成十进制,在LCD上显示。第二部 分为读取开关状态,转换为十六进制,在16×16点阵上扫描显示。
- 2. 初始化过程中,8253CNT0采用方式3,CLK0接入1MHz方波,OUT0输出100MHz方波;CNT1采用方式0,CLK1接OUT0,计数初值为FFFFH。LCD初始化依次写入命令30H,0CH,01H,06H,80H。
- 3. 中断程序INT_TIMER:通过单脉冲按钮作为秒表的启动/暂停按键,按下后触发IRQ3中断。通过 IS_TIMING确定是开始还是暂停,每次执行后更改该变量。当开始计时,从数据段装入计数初值, 开始计时;当暂停计时时,锁存并读出当前计数值,保存在数据段的计数初值中,为下一次开始做 准备。
- 4. 获取时间:若正在计时,则锁存并读出CNT1计数值;若暂停计时,则从数据段取出计数初值。用FFFFH减去这个数,即为经过0.01s的个数。之后依次除1000、100、10,分解为四位。覆盖TIME_STRING('xx.xxs')对应位置。
- 5. 显示时间:每次把光标移到92H,依次写入TIME_STRING各字符。
- 6. 16×16点阵显示: 8255读出开关值存到数据段。计算出字模的首地址(OFFSET LED_TABLE + 字符*16),采用列扫描,利用视觉暂留效应显示字符。

硬件接线表

- 1. 8353
 - 1. CLK0----1MHz
 - 2. OUT0----CLK1
 - 3. CS----0290H-0293H
- 2.8255
 - 1. PA----LCD:D0-D7
 - 2. PB----8个开关
 - 3. PC0----LCD:RS
 - 4. PC1----LCD:R/W
 - 5. PC2----LCD:E

- 6. CS----0280H-0283H
- 3.8259
 - 1. IRQ3----启动/暂停单脉冲
- 4. 16*16LED点阵:JX1扩展板
 - 1. 列高8位----02B1H
 - 2. 列低8位----02B0H
 - 3. 行高8位----02B3H
 - 4. 行低8位----02B2H

流程图

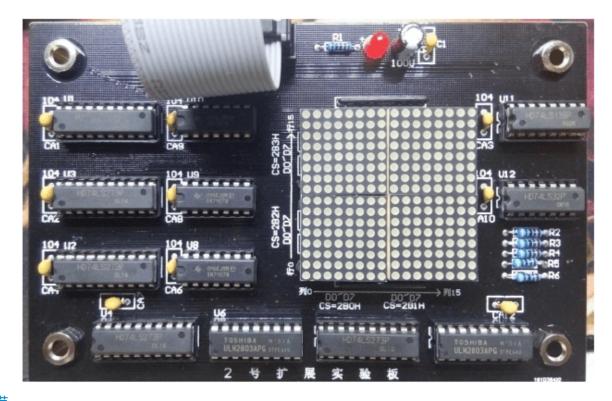


遇到问题及解决方案

1. 问题:

参考资料

PS2 ST7920说明书



16*8字模