硬件课程设计报告

课题名称: 电梯

班 级: 计科5班\_\_\_\_\_

姓 名: \_\_\_\_\_ \_秦威\_\_\_\_\_\_

学 号: 08143324\_\_\_\_

指导教师 : \_\_\_\_\_\_\_王凯\_\_\_\_\_\_

2017 年 1月 1日

摘 要

随着科技的发展和时代的进步，电梯这科技发展的产物也越加的深入百姓生活中。本次实验是电梯模拟试验。通过对试验箱上的部分芯片进行连接，再编码，实现电梯的基本功能。

研究的主要内容有：利用8255控制4\*4键盘与LCD显示屏，利用8255控制步进电机正、反转，完成电梯上下方向控制，LED屏显示上行、下行、开门、关门、楼层号显示。

通过本次实验，我已基本完成了上述要求。对8255控制4\*4键盘与LCD显示屏等内容有了深入的理解。实验遇到了一些问题。主要是键盘的输入、8253的延时控制、步进电机的正反转、LCD显示屏的显示。此外，对电梯运行的主体程序的编写也是难点之一。

通过本次实验，我巩固了微机原理课程所学的内容，对电梯运行的原理及实现有了深入的理解，增强了我的编程能力。

【关键词】 电梯 步进电机 LCD显示屏 8255芯片

目 录

[1 绪 论 1](#_Toc357605105)

[1.1问题提出 1](#_Toc357605106)

[1.2 选题意义 1](#_Toc357605107)

[1.3 设计任务与要求 1](#_Toc357605108)

[2 系统设计需求分析 1](#_Toc357605111)

[2.1开发环境及开发平台 1](#_Toc357605112)

[2.2系统设计主要芯片及工作原理 3](#_Toc357605113)

[2.3 系统电路组成及工作原理 1](#_Toc357605114)1

2.4 系统主要设计思想 [1](#_Toc357605117)2

[3 系统总体设计 1](#_Toc357605120)3

[3.1系统电路组成及工作原理 1](#_Toc357605121)3

[3.1.1 系统电路图设计 1](#_Toc357605114)3

[3.1.2系统工作原理 1](#_Toc357605115)4

[3.2系统软件功能 1](#_Toc357605116)6

[3.2.1系统软件功能模块图 1](#_Toc357605116)6

4 [系统软件设计 1](#_Toc357605120)7

[4.1系统软件设计程序流程图 1](#_Toc357605121)7

[4.2 软件代码设计 18](#_Toc357605114)

5[系统分析与测试 3](#_Toc357605115)6

[5.1系统测试方法与结论 3](#_Toc357605116)6

6 [系统运行结果 3](#_Toc357605116)7

7[结论 4](#_Toc357605115)0

8 [设计体会 4](#_Toc357605116)1

参考文献 [4](#_Toc357605116)1

1 绪 论

1.1问题提出

电梯现已广泛应用于百姓的生活，随处可见。那么，电梯是如何运行的呢？如何在HQFC开发板上模拟电梯的功能呢？这些都需要实践来证明。这便是本次实验的出发点。

1.2 选题意义

硬件课程设计选择电梯这个题目是经过深思熟虑的。时代在进步，电梯的应用也越来越广泛，那么其开发，应用及安全应得到更多的关注。想要提高这一产品的性能、安全性等方面，就要从源头上了解它，明白它是怎样工作的，然后再进一步提升。我们在生活中经常用到电梯这种工具，通过实验加深对它的了解是一件很有意义的事，不仅要知道它是怎样运行的，还要知道如何去实现它的功能。此外，此实验在HQFC开发板上用到的芯片很多，对我自己也是一次考验，使我对微机原理所学内容进行更深入的探讨。因此，我选择了这个题目。

1.3 设计任务与要求

利用8255控制4\*4键盘与LCD显示屏，利用8255控制步进电机正、反转，完成电梯上下方向控制，LCD屏显示上行、下行、开门、关门、楼层号显示。

对电梯进行初始化，将目标楼层（l）初始化为1，当前楼层（p）初始化为1,上升序列数组（a[4]）初始化为{0,0,0,0,},下降数组（b[4]）初始化为{0,0,0,0,}。

通过4\*4键盘模拟电梯内外键。对应如下：0--1楼外键上；1--1楼內键；2--2楼内键；3--3楼内键；4--4楼内键；5--2楼外键上；6--2楼外键下；7--3楼外键上；8--3楼外键下；9--4楼外键下；A--关门；B--开门。

步进电机：当电梯上行时逆时针旋转，下降时顺时针旋转。

LCD显示屏显示当前电梯状态，包括上行、下行和当前楼层、目标楼层。

8253将控制延时，每上升或下降一层需要1秒。

电梯用户在所在楼层按下外键使电梯到来，按下内键表明所去目标楼层。

2 系统设计需求分析

2.1开发环境及开发平台

**开发环境：HQFC-A微机单片机接口实验系统**：

在各种计算机外围接口不断推陈出新的今天，USB接口已经成为个人计算机最重要的接口方式之一，USB接口设备的应用也以惊人的速度发展，几乎新型的PC都100%支持USB技术。了解和掌握USB的应用及开发是计算机类、电子类、物理类本科生、大专生的新课题。

TPC-ZK-USB微机接口实验系统正是在这种背景下推出的。该设备在TPC-ZK实验系统上配置了USB接口模块，直接与主机(PC)的USB接口连接，形成了一套完整的USB接口的微机接口实验系统。该系统适应当前高等院校所开设的《微机原理及其应用》和《微机接口技术》这两门课的实验，同时也提供了最新接口USB的实验，使学生在校学习期间不仅有机会接触常规接口，同时有机会接触新型的接口，为学生们今后从事微机开发应用打下基础。

该系统由一块USB总线接口模块、TPC-ZK验系统及集成开发环境软件组成。USB总线接口模块通过USB总线电缆与PC机相连，模块直接插在TPC-ZK实验系统上。其主要特点如下：

1. USB总线接口使用ISP1581/1583 USB2.0高速接口芯片，完全符合USB2.0规范。提供了高速USB下的通信能力，即插即用。

2.满足《微机原理与接口技术》课程教学实验要求。实验台接口集成电路包括：可编程定时器/计数器（8254）、可编程并行接口（8255）、数/模转换器(DAC0832)、模/数转换器(ADC0809)等。外围电路包括：逻辑电平开关、LED显示、七段数码管显示、8X8双色发光二极管点阵及驱动电路、直流电机步进电机及驱动电路、电机测速用光藕电路、继电器及驱动电路、喇叭及驱动电路、键盘显示控制电路等。

3.在USB接口模块上扩展有DMA控制器8237，可以完成微机DMA传送以及USB的DMA传送等实验。

4.开放式结构，模块化设计支持开放实验。实验台上除固定电路外还设有用户扩展实验区。插座引脚都有对应的“自锁紧”插孔，利用这些插孔可以搭试更多的自己设计的实验，方便的进行课程设计。

5.功能强大的软件集成开发环境，支持Win2000;WinXP 等操作系统（不支持WIN98系统）。可以方便的对程序进行编辑、编译、链接和调试，可以查看实验原理图，实验接线，实验程序并进行实验演示。可以增加和删除实验项目。

6.实验程序可以使8086汇编和C语言编程实验。可以对汇编程序和C语言程序进行调试(C语言调试系统需安装了VC软件,因版权不提供该软件).

7.系统还提供：字符、图形液晶显示实验模块；红外收发实验模块；无线通信实验模块；8279键盘显示实验模块等多种扩展实验模块。

8.实验台自备电源，具有电源短路保护确保系统安全。

9.使用USB接口与PC机相连，省却了打开主机箱安装接口卡的麻烦。

**开发平台：HQFC开发板**

TPC-ZK实验系统结构图如下

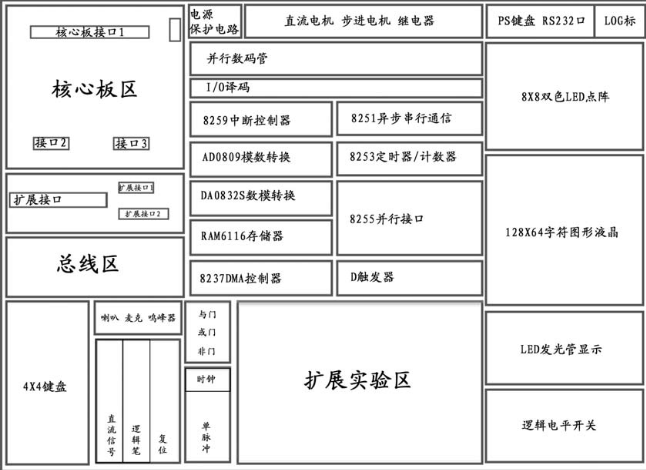


图2-1 TPC-ZK实验系统结构图

2.2系统设计主要芯片及工作原理

**2.2.1 8255芯片**

8255是Intel公司生产的可编程并行I/O接口芯片，有3个8位并行I/O口。具有3个通道3种工作方式的可编程并行接口芯片(40引脚)。 其各口功能可由软件选择，使用灵活，通用性强。8255可作为[单片机](http://baike.so.com/doc/30473-31769.html" \t "http://baike.so.com/doc/_blank)与多种外设连接时的中间接口电路。

8255作为主机与外设的连接芯片，必须提供与[主机](http://baike.so.com/doc/5331327-5566564.html" \t "http://baike.so.com/doc/_blank)相连的3个总线接口，即[数据线](http://baike.so.com/doc/5339043-5574484.html" \t "http://baike.so.com/doc/_blank)、[地址线](http://baike.so.com/doc/2306723-2440032.html" \t "http://baike.so.com/doc/_blank)、[控制线](http://baike.so.com/doc/6678090-6891969.html" \t "http://baike.so.com/doc/_blank)接口。同时必须具有与[外设](http://baike.so.com/doc/325007-344304.html" \t "http://baike.so.com/doc/_blank)连接的接口A、B、C口。由于8255可编程,所以必须具有逻辑控制部分，因而8255内部结构分为3个部分:与CPU连接部分、与外设连接部分、控制部分。

1)与CPU连接部分

根据定义，8255能并行传送8位数据，所以其数据线为8根D0~D7。由于8255具有3个通道A、B、C，所以只要两根地址线就能寻址A、B、C口及控制寄存器，故地址线为两根A0~A1。此外CPU要对8255进行读、写与片选操作，所以控制线为片选、复位、读、写信号。各信号的引脚编号如下:

(1)数据总线DB:编号为D0~D7，用于8255与CPU传送8位数据。

(2)地址总线AB:编号为A0~A1，用于选择A、B、C口与控制寄存器。

(3)控制总线CB:片选信号、复位信号RST、写信号、读信号。当CPU要对8255进行读、写操作时，必须先向8255发片选信号选中8255芯片，然后发读信号或写信号对8255进行读或写数据的操作。

2)与外设接口部分

根据定义，8255有3个通道A、B、C与外设连接，每个通道又有8根线与外设连接，所以8255可以用24根线与外设连接，若进行开关量控制，则8255可同时控制24路开关。各通道的引脚编号如下:

(1)A口:编号为PA0~PA7，用于8255向外设输入输出8位并行数据。

(2)B口:编号为PB0~PB7，用于8255向外设输入输出8位并行数据。

(3)C口:编号为PC0~PC7，用于8255向外设输入输出8位并行数据，当8255工作于应答I/O方式时，C口用于应答信号的通信。

3)控制器部分

8255将3个通道分为两组，即PA0~PA7与PC4~PC7组成A组，PB0~PB7与PC0~PC3组成B组。如图7.5所示，相应的控制器也分为A组控制器与B组控制器，各组控制器的作用如下:

(1)A组控制器:控制A口与上C口的输入与输出。

(2)B组控制器:控制B口与下C口的输入与输出。



图2-2 8255管脚图

引脚功能

RESET:复位输入线，当该输入端处于高电平时，所有内部寄存器(包括控制寄存器)均被清除，所有I/O口均被置成输入方式。

CS:芯片选择信号线，当这个输入引脚为低电平时,即/CS=0时,表示芯片被选中，允许8255与CPU进行通讯;/CS=1时,8255无法与CPU做数据传输.

RD:读信号线，当这个输入引脚为低跳变沿时,即/RD产生一个低脉冲且/CS=0时,允许8255通过数据总线向CPU发送数据或状态信息，即CPU从8255读取信息或数据。

WR:写入信号，当这个输入引脚为低跳变沿时,即/WR产生一个低脉冲且/CS=0时,允许CPU将数据或控制字写入8255。

D0~D7:三态双向数据总线，8255与CPU数据传送的通道，当CPU 执行输入输出指令时，通过它实现8位数据的读/写操作，控制字和状态信息也通过数据总线传送。

8255具有3个相互独立的输入/输出通道端口，用+5V单电源供电，能在以下三种方式下工作。

方式0----基本输入输出方式;方式1----选通输入/出方式;方式2----双向选通

输入/输出方式;

PA0~PA7:端口A输入输出线，一个8位的数据输出锁存器/缓冲器， 一个8位的数据输入锁存器。 工作于三种方式中的任何一种;

PB0~PB7:端口B输入输出线，一个8位的I/O锁存器， 一个8位的输入输出缓冲器。 不能工作于方式二;

PC0~PC7:端口C输入输出线，一个8位的数据输出锁存器/缓冲器， 一个8位的数据输入缓冲器。端口C可以通过工作方式设定而分成2个4位的端口， 每个4位的端口包含一个4位的锁存器，分别与端口A和端口B配合使用，可作为控制信号输出或状态信号输入端口。'不能工作于方式一或二。

A1,A0:地址选择线,用来选择8255的PA口,PB口,PC口和控制寄存器.当A1=0,A0=0时,PA口被选择;当A1=0,A0=1时,PB口被选择;当A1=1,A0=0时,PC口被选择;当A1=1.A0=1时,控制寄存器被选择.

命令字

D7 D6 D5 D4 D3 D2 D1 D0

D7:特征位 D6D5:A组方式00=方式0 ，01=方式1，1X=方式2

D4: A口0=输出，1=输入 D3:C口C7~C4 0=输出，1=输入D2:B组方式0=方式0，1=方式1

D1: B口0=输出，1=输入D0:C口C3~C0 0=输出，1=输入

D7位为"1"时，为方式选择控制字的标识位。

D6、D5位决定A端口的工作方式。

D4位决定A端口工作在输入还是输出方式。

D3位决定C端口高4位PC7~PC4是作为输入端口，还是作为输出端口。

D2位用来选择B端口的工作方式。

D1位决定B端口作为输入还是输出端口。

D0位决定C端口低4位PC3~PC0作出为输入，还是输出。

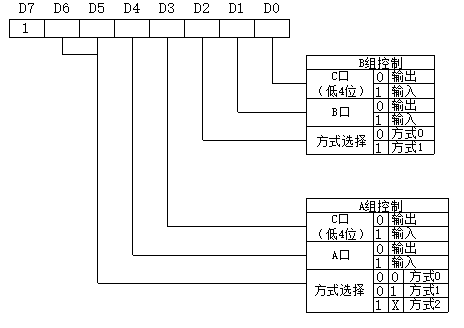


图2-3 8255控制字图示

**2.2.2 4\*4键盘**

在微型机系统中，键盘是一种最常用的外设，它由多个开关组合而成。可以用来制造键盘的按键开关由好多种，最常用的由机械式、薄膜式、电容式和霍尔效应式四种。机械式开关比较便宜，但压键时会产生触点抖动，即在触点可靠地接通前会通断多次，而且长期使用后可靠性会降低。高质量机械式开关的说明约为100万次。薄膜式开关可做成很薄的密封单元，不易受外界潮气或环境污染，常用于微波炉、医疗仪器或电子秤等设备的按键，不同薄膜开关的说明差别很大。电容式开关没有抖动问题，但需要特制电路来测电容的变化，平均寿命约为2000万次。霍尔效用案件是另一种无机械触点的开关，具有很好的密封性，平均寿命高达1亿甚至更高，但开关机制复杂，价格很贵。计算机上用的一般都是机械式开关。

当在键盘上压一个键时，等于在压下一个开关。对于大多数的键盘，按键被拍成行和列的矩阵。

本实验用的是8255C口接键盘。

8255B口接键盘（实际8255的PC0~PC3接键盘的列0～列3，PC5～PC7接键盘的行1～行3），8255CS接地址译码输出的288h~28fh。

识别键盘上的闭合键，通过采用行扫描法或行翻转法：

行扫描法是使键盘上某一行线为低电平，而其余行接高电平，然后读取列值；如果列值中有某位为低电平，则表明行列交点处的键被按下；否则扫描下一行，直到扫完全部的行线为止。

行反转法识别闭合键时，要将行线接一个并行口，先让它工作在输出方式，将列线也接到一个并行口，先让它工作在输入方式；程序通过输出端口向全部行线上送低电平，然后读取列线的值；如果此时有某一键被按下，则必定会使某一列线值为零，程序再对两个并行端口进行方式设置，使行线工作在输入方式，列线工作在输出方式，并且将刚才读到的列线值从列线所接的并行端口输出，再读取行线上的值；那么，在闭合键所在的行线上的值必定为零。这样，当一个键被按下时，必定可以读到一对唯一的行值和列值。

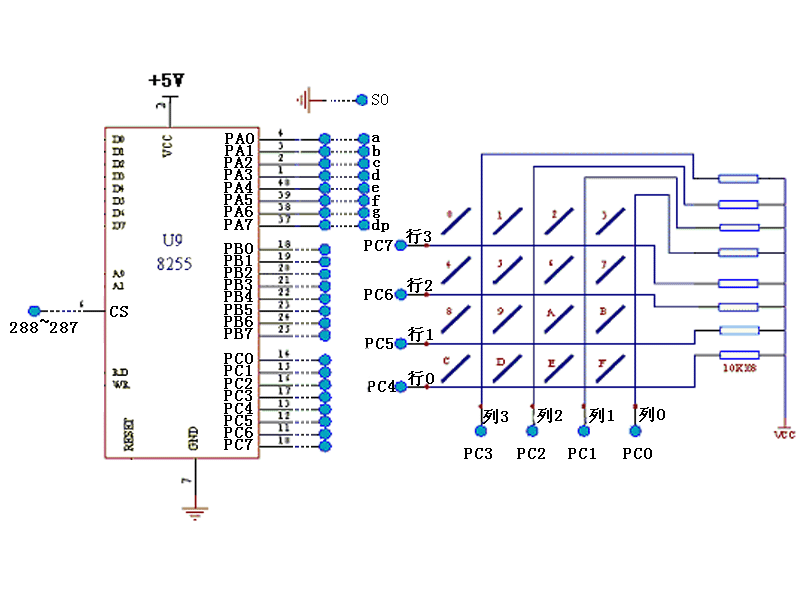
****

图2-4键盘原理图

**2.2.3 LCD显示屏**

ST7920是台湾矽创电子公司生产的中文图形控制芯片,它是一种内置128×64汉字图形点阵的液晶显示控制模块,用于显示汉字及图形｡12864A-1汉字图形点阵液晶显示模块集成ST7920驱动器，可显示汉字及图形，内置8192个中文汉字（16×16点阵）、128个字符（8×16点阵）及64×256点阵显示RAM（GDRAM）。主要技术参数和显示特性：

电源：VDD 3.3V~+5V(内置升压电路，无需负压)；

显示内容：128列× 64行；

显示颜色：黄绿；

显示角度：6：00钟直视；

LCD类型：STN；

与MCU接口：8位或4位并行/3位串行；

2MHZ时钟频率；

汉字显示坐标：



图2-5 128×64液晶显示屏汉字显示坐标表



图2-6 引脚说明

逻辑工作电压(VDD)：4.5～5.5V

电源地(GND)：0V

工作温度(Ta)：0～60℃(常温) / -20～75℃（宽温）

**2.2.4 步进电机**

实验台上使用的步进电机驱动方式为二相励磁方式，BA、BB、BC、BD分别为四个线圈的驱动输入端，输入高电平时，相应线圈通电。当其通电顺序为BA、BB、BC、BD时为一个转动方向；通电顺序为BD、BC、BB、BA时为相反顺序。步进电机驱动电路如下：

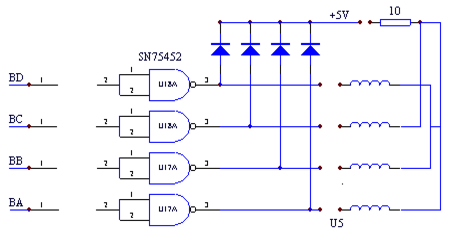


图2-7 步进电机驱动电路

**2.2.5 8253定时器**

8253内部有三个计数器，分别称为计数器0、计数器1和计数器2，他们的机构完全相同。每个计数器的输入和输出都决定于设置在控制寄存器中的控制字，互相之间工作完全独立。每个计数器通过三个引脚和外部联系，一个为时钟输入端CLK，一个为门控信号输入端GATE，另一个为输出端OUT。每个计数器内部有一个8位的控制寄存器，还有一个16位的计数初值寄存器CR、一个计数执行部件CE和一个输出锁存器OL。

执行部件实际上是一个16位的减法计数器，它的起始值就是初值寄存器的值，而初始值寄存器的值是通过程序设置的。输出锁存器的值是通过程序设置的。输出锁存器OL用来锁存计数执行部件CE的内容，从而使CPU可以对此进行读操作。顺便提一下，CR、CE和OL都是16位寄存器，但是也可以作8位寄存器来用。

8253内部结构

8253芯片有24条引脚，封装在双列直插式陶瓷管壳内。

1.[数据总线](http://baike.so.com/doc/6148763-6361949.html" \t "http://baike.so.com/doc/_blank)缓冲器

数据总线缓冲器与系统总线连接，8位双向，与CPU交换信息的通道。这是8253与CPU之间的数据接口，它由8位双向三态缓冲存储器构成，是CPU与8253之间交换信息的必经之路。

[](http://p2.qhmsg.com/t019bb3af0a4926d9d9.jpg)

图2-8 8253端口选择

1. 读/写控制

读/写控制分别连接系统的IOR#和IOW#， 由CPU控制着访问8253的内部通道。接收CPU送入的读/写控制信号， 并完成对芯片内部各功能部件的控制功能， 因此， 它实际上是8253芯片内部的控制器。A1A0:端口选择信号，由CPU输入。8253内部有3个独立的通道，加上控制字寄存器，构成8253芯片的4个端口，CPU可对3个通道进行读/写操作3对控制字寄存器进行写操作。 这4个[端口地址](http://baike.so.com/doc/4068764-4267314.html" \t "http://baike.so.com/doc/_blank)由最低2位地址码A1和A0来选择。如表所示。

3.通道选择

(1) CS#--[片选信号](http://baike.so.com/doc/633061-670010.html" \t "http://baike.so.com/doc/_blank)，由CPU输入，低电平有效，通常由端口地址的高位地址译码形成。

(2) RD#、WR#--读/写控制命令，由CPU输入， 低电平有效。RD#效时，CPU读取由A1A0所选定的通道内计数器的内容。WR#有效时，CPU将计数值写入各个通道的计数器中， 或者是将方式控制字写入控制字寄存器中。CPU对8253的读/写操作。

4.计数通道0~2

每个计数通道内含1个16位的初值寄存器、减1计数器和1个16位的(输出)锁存器。8253内部包含3个功能完全相同的通道，每个通道内部设有一个16位计数器，可进行二进制或[十进制](http://baike.so.com/doc/5333362-5568797.html" \t "http://baike.so.com/doc/_blank)(BCD码)计数。采用二进制计数时， 写入的初值范围为0000H~0FFFFH，最大计数值是0000H，代表65536。 采用BCD码计数时，写入的初值范围为0000~9999，最大计数值是0000，代表10000。与此计数器相对应，每个通道内设有一个16位计数值锁存器。必要时可用来锁存计数值。(特别说明:8253计数器的值先减1再判断是否为0，为0就中断了，所以最大初始值为0，这样减1以后，不为0，所以为最大的，取决于CF标志位)

当某通道用作计数器时，应将要求计数的次数预置到该通道的计数器中、被计数的事件应以脉冲方式从CLK端输入， 每输入一个计数脉冲，计数器内容减"1"，待计数值计到"0"。 OUT端将有输出。表示计数次数到。当某个通道用作[定时器](http://baike.so.com/doc/2784003-2938493.html" \t "http://baike.so.com/doc/_blank)时。 由CLK输入一定频率的[时钟脉冲](http://baike.so.com/doc/955089-1009606.html" \t "http://baike.so.com/doc/_blank)。根据要求定时的时间长短确定所需的计数值。并预置到计数器中，每输入一个时钟脉冲，计数器内容减"1"， 待计数值计到"0"。OUT将有输出，表示定时时间到。允许从CLK输入的[时钟](http://baike.so.com/doc/5407848-5645795.html" \t "http://baike.so.com/doc/_blank)频在1~2MHz范围内。因此，任一通道作计数器用或作定时器用，其内部操作完全相同，区别仅在于前者是由计数脉冲进行减"1"计数。 而后者是内时钟脉冲进行减"1"计数。作计数器时， 要求计数的次数可直接作为计数器的初值预置到减"1"计数器中。作[定时器](http://baike.so.com/doc/2784003-2938493.html" \t "http://baike.so.com/doc/_blank)时， 计数器的初值即定时系数应根据要求定时的时间进行如下运算才能得到:

定时系数=需要定时的时间/[时钟脉冲](http://baike.so.com/doc/955089-1009606.html" \t "http://baike.so.com/doc/_blank)周期

①设置通道:向方式控制字寄存器端口写入方式选择控制字，用于确定要设置的通道及工作方式;

②计数/定时:向通道写入计数值，启动计数操作;

③读取当前的计数值:向指定通道读取当前计数器值时，8253将计数器值存入锁存器，从锁存器向外提供当前的计数器值，计数器则继续作计数操作。

④计数到:当计数器减1为0时，通过引脚OUTi向外输出"到"的[脉冲信号](http://baike.so.com/doc/6297825-6511348.html" \t "http://baike.so.com/doc/_blank)。

计数初值输入存放在初值寄存器中，计数开始或重装入时被复制到计数器中。

锁存器在非锁存状态，其值随计数器的变化而变化;一旦锁存了计数器的当前值，直到锁存器值被读取后才能解除锁存状态。

[](http://p2.qhmsg.com/t01eaddd7e0597e9e01.jpg)

图2-9 控制字格式、通道选择、读/写方式、工作方式、计数数制选择图解

5.方式选择控制字

8253的初始化编程就是对其工作方式的确定。具体实现就是在8253上电后，由CPU向8253的[控制寄存器](http://baike.so.com/doc/4459661-4668293.html" \t "http://baike.so.com/doc/_blank)写入一个控制字，就可以规定8253的工作方式、计数值的长度以及计数所用的数制等，另外根据要求将计数值写入8253的相应通道。

8253的一个方式控制字只决定一个技术通道的工作模式。

8253中各通道可有6种可供选择的工作方式， 以完成定时、计数或脉冲发生器等多种功能。8253的各种工作方式如下:

方式0:计数结束则中断；.方式1:单脉冲发生器；方式2:速率波发生器；方式3:方波发生器；方式4:软件触发方式计数；方式5:硬件触发方式计数。

本实验采用方式2 。

2.3 系统电路组成及工作原理

电路图如下：

288~28f'H

280~287H

步进电机

BA BB BC BD

cs

8253

芯片

out1

cs 8255芯片 PB4 PB5 PB6 PB7

PC4

PC0~PC3

PC5~PC7 PB0 PB1 PB2 PA7~PA0

4\*4键盘

列0~列3

行1~行3

D7~D0

I

RW

D/I

LCD

显示屏

图2-10 系统电路图

工作原理：键盘输入想去的楼层，交给8255处理。调用主程序判断电梯运行状态。8255的PC4接收延时信号，进行延时操作，并输出信号，令异步电机工作，LCD显示屏显示电梯运行信息。

2.4 系统主要设计思想

首先键盘检测按键，改变上升序列数组或下降序列数组。当目标楼层和当前楼层不同时，执行上楼或者下楼操作，每上一层或下一层调用异步电机，延时并显示在LCD显示屏上。当到一个目标楼层时，调用延时：开门-延时-关门。将延时写在步进电机函数里，上或下延时并使异步电机转动，若状态为停，则延时，但步进电机无输入。每进行上升或下降都要在检测是否有键按下。先执行最接近当前楼层的目标楼层的操作。到达后该楼层数组改为0。

3 系统总体设计

3.1系统电路组成及工作原理

**3.1.1系统电路图设计**

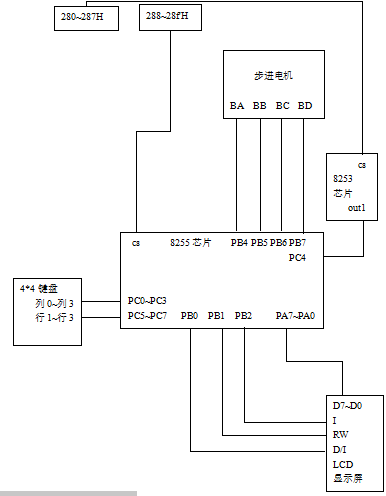


图3-1 系统电路图

工作原理：键盘输入想去的楼层，交给8255处理。调用主程序判断电梯运行状态。8255的PC4接收延时信号，进行延时操作，并输出信号，令异步电机工作，LCD显示屏显示电梯运行信息。

**3.1.2系统工作原理**

LCD显示屏显示原理:

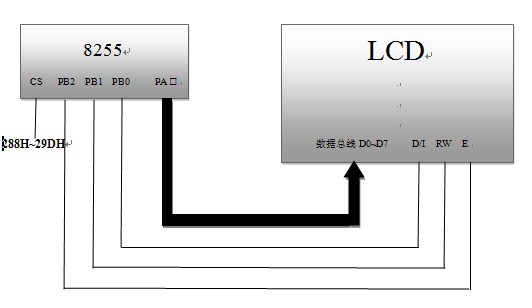


图3-2 LCD与8255连线图

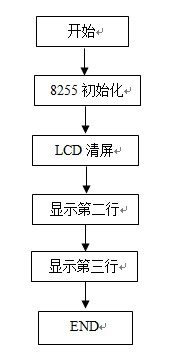


图3-3 LCD液晶屏显示流程图

键盘原理:

键盘的工作原理是用4条I/O线作为行线，4条I/O线作为列线组成的键盘。在行线和列线的每一个交叉点上，设置一个按键。这样键盘中按键的个数是4×4个。这种行列式键盘结构能够有效地提高单片机系统中I/O口的利用率。

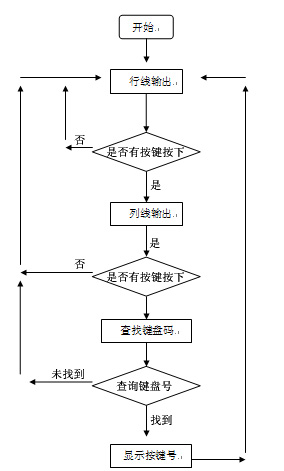
****

图3-4 键盘流程图

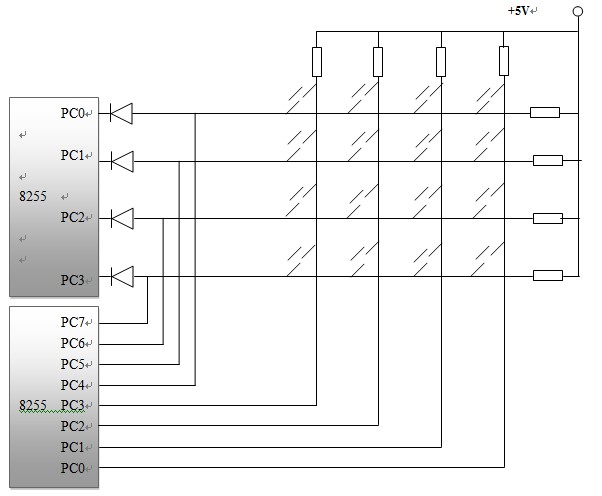
****

图3-5 键盘电路图

步进电机：

步进电机是将电[脉冲](http://baike.so.com/doc/5327352-5562524.html" \t "http://baike.so.com/doc/_blank)信号转变为[角位移](http://baike.so.com/doc/2803889-2959433.html" \t "http://baike.so.com/doc/_blank)或线位移的开环控制元件。在非超载的情况下，电机的转速、停止的位置只取决于脉冲信号的频率和脉冲数，而不受负载变化的影响，当步进驱动器接收到一个脉冲信号，它就驱动步进电机按设定的方向转动一个固定的角度，称为"步距角"，它的旋转是以固定的角度一步一步运行的。可以通过控制脉冲个数来控制角位移量，从而达到准确定位的目的;同时可以通过控制脉冲频率来控制电机转动的速度和加速度，从而达到调速的目的。

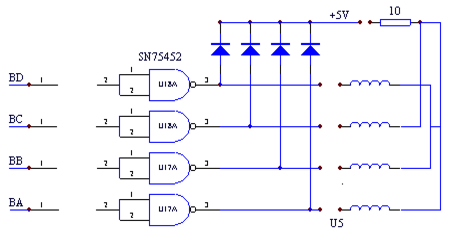


图3-6 步进电机电路图

3.2系统软件功能

**3.2.1系统软件功能模块图**

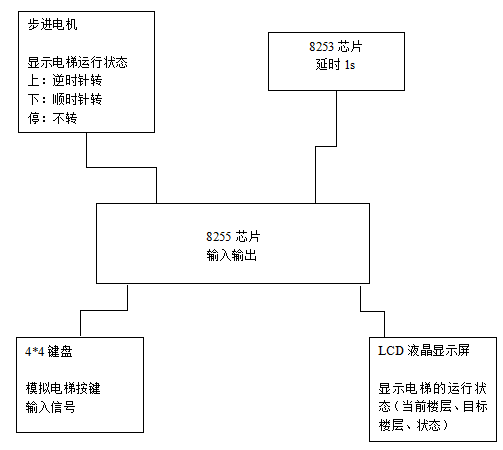


图3-7 系统功能模块图

4系统软件设计

4.1系统软件设计程序流程图

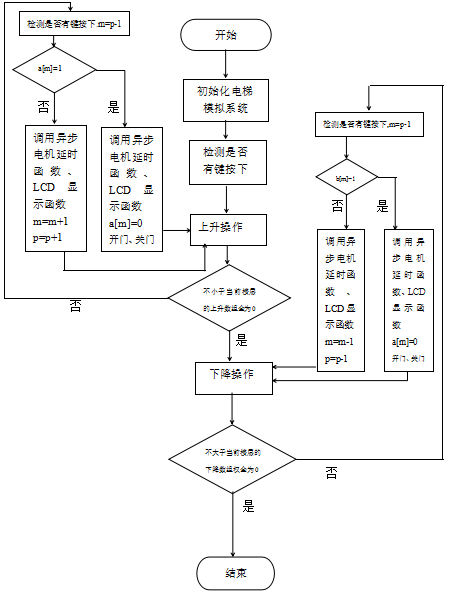


图4-1 总的系统流程图

首先键盘检测按键，改变上升序列数组或下降序列数组。当目标楼层和当前楼层不同时，执行上楼或者下楼操作，每上一层或下一层调用异步电机，延时并显示在LCD显示屏上。当到一个目标楼层时，调用延时：开门-延时-关门。将延时写在步进电机函数里，上或下延时并使异步电机转动，若状态为停，则延时，但步进电机无输入。每进行上升或下降都要在检测是否有键按下。先执行最接近当前楼层的目标楼层的操作。到达后该楼层数组改为0。若键盘无键按下，则重复执行这一流程图。

4.2 软件代码设计

#include<stdio.h>

#include<conio.h>

#include "ApiExusb.h"

#pragma comment(lib,"ApiExusb.lib")

char w1[16]={0xb5,0xc7,0xc2,0xb2,0xc4,0xb1,0xc2,0xb2,0xd7,0xcc,0xcd,0xc9,0xcf,0xbf,0xc3,0xb9};

char w2[16]={0xb1,0xb0,0xa5,0xe3,0xbf,0xea,0xa5,0xe3,0xb4,0xac,0xa3,0xcf,0xc2,0xaa,0xc5,0xd8};

char lcd[4]={0xb1,0xb2,0xb3,0xb4};

int zt=1;

int a[4]={0,0,0,0};

int b[4]={0,0,0,0};

int p=1,l=1,s=0,m=0,g=0;

int buf = 0x33;

void clear();

void cmdsetup();

void datasetup();

void clean1()

{

PortWriteByte(0x28b, 0x81); /\*设置8255的A口C口均为输出\*/

clear();

Sleep(50);

int y;

clear();

PortWriteByte(0x288, 0x84);

cmdsetup();

Sleep(10);

for (y = 0; y<1; y++)

{

PortWriteByte(0x288, 0xA1);

datasetup();

PortWriteByte(0x288, 0xA0);

datasetup();

}

Sleep(10);

}

void clean2()

{

PortWriteByte(0x28b, 0x81); /\*设置8255的A口C口均为输出\*/

clear();

Sleep(50);

int y;

clear();

PortWriteByte(0x288, 0x94);

cmdsetup();

Sleep(10);

for (y = 0; y<2; y++)

{

PortWriteByte(0x288, 0xA1);

datasetup();

PortWriteByte(0x288, 0xA0);

datasetup();

}

Sleep(10);

}

void clean3()

{

PortWriteByte(0x28b, 0x81); /\*设置8255的A口C口均为输出\*/

clear();

Sleep(50);

int y;

clear();

PortWriteByte(0x288, 0x8c);

cmdsetup();

Sleep(10);

for (y = 0; y<1; y++)

{

PortWriteByte(0x288, 0xA1);

datasetup();

PortWriteByte(0x288, 0xA0);

datasetup();

}

Sleep(10);

}

void show()

{

PortWriteByte(0x28b, 0x81); /\*设置8255的A口C口均为输出\*/

clear();

clear();

PortWriteByte(0x288,0x80);

cmdsetup();

//mu

PortWriteByte(0x288,w1[4]);

datasetup();

PortWriteByte(0x288,w2[4]);

datasetup();

//biao

PortWriteByte(0x288,w1[5]);

datasetup();

PortWriteByte(0x288,w2[5]);

datasetup();

//lou

PortWriteByte(0x288,w1[6]);

datasetup();

PortWriteByte(0x288,w2[6]);

datasetup();

//ceng

PortWriteByte(0x288,w1[7]);

datasetup();

PortWriteByte(0x288,w2[7]);

datasetup();

clear();

PortWriteByte(0x288,0x90);

cmdsetup();

//zhuang

PortWriteByte(0x288,w1[8]);

datasetup();

PortWriteByte(0x288,w2[8]);

datasetup();

//tai

PortWriteByte(0x288,w1[9]);

datasetup();

PortWriteByte(0x288,w2[9]);

datasetup();

clear();

PortWriteByte(0x288,0x88);

cmdsetup();

//dang

PortWriteByte(0x288,w1[0]);

datasetup();

PortWriteByte(0x288,w2[0]);

datasetup();

//qian

PortWriteByte(0x288,w1[1]);

datasetup();

PortWriteByte(0x288,w2[1]);

datasetup();

//lou

PortWriteByte(0x288,w1[2]);

datasetup();

PortWriteByte(0x288,w2[2]);

datasetup();

//ceng

PortWriteByte(0x288,w1[3]);

datasetup();

PortWriteByte(0x288,w2[3]);

datasetup();

clear();

PortWriteByte(0x288,0x90);

cmdsetup();

Sleep(10);

}

void zhuangtaixianshihanshu() //显示电梯当前状态

{ clean2();

PortWriteByte(0x28b, 0x81); /\*设置8255的A口C口均为输出\*/

clear();

clear();

PortWriteByte(0x288,0x94);

cmdsetup();

if(zt==0)

{

PortWriteByte(0x288,w1[12]);

datasetup();

PortWriteByte(0x288,w2[12]);

datasetup();

}

else if(zt==1)

{

PortWriteByte(0x288,w1[10]);

datasetup();

PortWriteByte(0x288,w2[10]);

datasetup();

}

else if(zt==2)

{

PortWriteByte(0x288,w1[11]);

datasetup();

PortWriteByte(0x288,w2[11]);

datasetup();

}

}

void open()

{clean2();

PortWriteByte(0x28b, 0x81); /\*设置8255的A口C口均为输出\*/

clear();

clear();

PortWriteByte(0x288,0x94);

cmdsetup();

PortWriteByte(0x288,w1[13]);

datasetup();

PortWriteByte(0x288,w2[13]);

datasetup();

/\* PortWriteByte(0x288,w1[14]);

datasetup();

PortWriteByte(0x288,w2[14]);

datasetup();\*/

}

void close()

{ clean2();

PortWriteByte(0x28b, 0x81); /\*设置8255的A口C口均为输出\*/

clear();

clear();

PortWriteByte(0x288,0x94);

cmdsetup();

PortWriteByte(0x288,w1[15]);

datasetup();

PortWriteByte(0x288,w2[15]);

datasetup();

/\*PortWriteByte(0x288,w1[14]);

datasetup();

PortWriteByte(0x288,w2[14]);

datasetup();\*/

}

void mubiaoloucengxianshihanshu() //显示目标楼层

{

PortWriteByte(0x28b, 0x81); /\*设置8255的A口C口均为输出\*/

clear();

clear();

PortWriteByte(0x288,0x84);

cmdsetup();

PortWriteByte(0x288,0xa3);

datasetup();

PortWriteByte(0x288,lcd[l-1]);

datasetup();

}

void dangqianloucengxianshihanshu() /\*显示当前楼层\*/

{

clean3();

PortWriteByte(0x28b, 0x81); /\*设置8255的A口C口均为输出\*/

clear();

clear();

PortWriteByte(0x288,0x8c);

cmdsetup();

PortWriteByte(0x288,0xa3);

datasetup();

PortWriteByte(0x288,lcd[p-1]);

datasetup();

}

void panduanshangshengshifoujieshu() /\*判断上升是否结束\*/

{

for(int x=p-1;x<4;x++)

{

if(a[x]==0)

{

if(x==3)

{

s=4;

x=4;

}

}

else if(a[x]==1)

{

s=m;

l=x+1;

x=4;

}

}

}

void panduanxiajiangshifoujieshu() //判断下降是否结束

{

for(int y=p-1;y>=0;y--)

{

if(b[y]==0)

{

if(y==0)

{ g=-1;

y=-1;

}

}

else if(b[y]==1)

{

g=m;

l=y+1;

y=-1;

}

}

}

void bujindianji()

{ byte dat;

PortWriteByte(0x283,0x36);/\*通道0控制字，先读写低字节，后读写高字节，方式2，\*/

PortWriteByte(0x280,1000%256);

PortWriteByte(0x280,1000/256);/\*初始化通道0,2000\*/

PortWriteByte(0x283,0x76);/\*通道1控制字，先读写低字节，后读写高字节，方式1，\*/

PortWriteByte(0x281,1000%256);

PortWriteByte(0x281,1000/256);/\*初始化通道1,500\*/

PortWriteByte(0x28b,0x89);

PortReadByte(0x28a,&dat);

BYTE data;

int d;

int m=0;

int buf1;

int buf2;

while(dat&0x10)

{ if (zt==1)

{

d = 0;

}

else if (zt==2)

{

d = 20;

data=0x80;

}

else if(zt==0)

{

d=20;

data=0x00;

}

if (d != 0)

{

Sleep(d);

if (data & 128)

{buf = ((buf&1)<<7)|(buf>>1);}

else

{buf = ((buf&128)>>7)|(buf<<1);}

buf1=buf&0xf0;

buf2=buf1|0x01;

PortWriteByte(0x289,buf2);

PortReadByte(0x28a,&dat);

}

else

{

Sleep(200);

PortWriteByte(0x289,0xf1);

PortReadByte(0x28a,&dat);

}

PortReadByte(0x28a,&dat);

}

}

void clear()

{

PortWriteByte(0x288, 0x0c);

cmdsetup();

}

void cmdsetup()

{

PortWriteByte(0x289, 0x00);

Sleep(1);

PortWriteByte(0x289, 0x04);

Sleep(1);

PortWriteByte(0x289, 0x00);

Sleep(1);

}

void datasetup()

{

PortWriteByte(0x289, 0x01);

Sleep(1);

PortWriteByte(0x289, 0x05);

Sleep(1);

PortWriteByte(0x289, 0x01);

Sleep(1);

}

void main()

{

printf("Start!\n");

if (!Startup()) /\*打开设备\*/

{

printf("ERROR: Open Device Error!\n");

return;

}

while (!kbhit())

{

show();

zhuangtaixianshihanshu();

mubiaoloucengxianshihanshu();

dangqianloucengxianshihanshu();

byte data;

byte i, j;

byte k;

PortWriteByte(0x28b, 0x81); /\*设置8255的A口C口均为输出\*/

clear();

Sleep(50);

PortWriteByte(0x28a, 0x0f);

cmdsetup();

PortReadByte(0x28a, &data);

i = data;

if (i != 0x0f)

{

i = data;

Sleep(50);

PortWriteByte(0x28b, 0x88);

PortWriteByte(0x28a, 0xf0);

PortReadByte(0x28a, &data);

i = i | data;

clear();

PortWriteByte(0x288, 0x91);

cmdsetup();

if (i == 0x77)

{

k = 1;

if(k<p)

{

b[k-1]=1;

}

}

else if (i == 0x7b)

{

k = 1;

if(k<p)

{b[k-1]=1;}

}

else if (i == 0x7d)

{

k = 2;

if(k<p)

{b[k-1]=1;}

else if(k>p)

{a[k-1]=1;}

}

else if (i == 0x7e)

{

k = 3;

if(k<p)

{b[k-1]=1;}

else if(k>p)

{a[k-1]=1;}

}

else if (i == 0xb7)

{

k = 4;

if(k<p)

{b[k-1]=1;}

else if(k>p)

{a[k-1]=1;}

}

else if (i == 0xbb)

{

k = 2;

if(k<p)

{b[k-1]=1;}

else if(k>p)

{a[k-1]=1;}

}

else if (i == 0xbd)

{

k = 2;

if(k<p)

{b[k-1]=1;}

else if(k>p)

{a[k-1]=1;}

}

else if (i == 0xbe)

{

k = 3;

if(k<p)

{b[k-1]=1;}

else if(k>p)

{a[k-1]=1;}

}

else if (i == 0xd7)

{

k = 3;

if(k<p)

{b[k-1]=1;}

else if(k>p)

{a[k-1]=1;}

}

else if (i == 0xdb)

{

k = 4;

if(k>p)

{a[k-1]=1;}

}

else if (i == 0xdd) //A 关门

{

k = 10;

}

else if (i == 0xde) // B 开门

{

k = 11;

}

}

m=p-1;

while(m<=3)

{

byte data;

byte i, j;

byte k;

PortWriteByte(0x28b, 0x81); /\*设置8255的A口C口均为输出\*/

clear();

Sleep(50);

PortWriteByte(0x28a, 0x0f);

cmdsetup();

PortReadByte(0x28a, &data);

i = data;

if (i != 0x0f)

{

i = data;

Sleep(50);

PortWriteByte(0x28b, 0x88);

PortWriteByte(0x28a, 0xf0);

PortReadByte(0x28a, &data);

i = i | data;

printf("%d",i);

clear();

PortWriteByte(0x288, 0x91);

cmdsetup();

if (i == 0x67)

{

k = 1;

if(k<p)

{

b[k-1]=1;

}

}

else if (i == 0x6b)

{

k = 1;

if(k<p)

{b[k-1]=1;}

}

else if (i == 0x6d)

{

k = 2;

if(k<p)

{b[k-1]=1;}

else if(k>p)

{a[k-1]=1;}

}

else if (i == 0x6e)

{

k = 3;

if(k<p)

{b[k-1]=1;}

else if(k>p)

{a[k-1]=1;}

}

else if (i == 0xa7)

{

k = 4;

if(k<p)

{b[k-1]=1;}

else if(k>p)

{a[k-1]=1;}

}

else if (i == 0xab)

{

k = 2;

if(k<p)

{b[k-1]=1;}

else if(k>p)

{a[k-1]=1;}

}

else if (i == 0xad)

{

k = 2;

if(k<p)

{b[k-1]=1;}

else if(k>p)

{a[k-1]=1;}

}

else if (i == 0xae)

{

k = 3;

if(k<p)

{b[k-1]=1;}

else if(k>p)

{a[k-1]=1;}

}

else if (i == 0xc7)

{

k = 3;

if(k<p)

{b[k-1]=1;}

else if(k>p)

{a[k-1]=1;}

}

else if (i == 0xcb)

{

k = 4;

if(k>p)

{a[k-1]=1;}

}

}

panduanshangshengshifoujieshu();

m=s;

if(m>3){break;}

if(a[m]==1)

{

zt=1;

mubiaoloucengxianshihanshu();

dangqianloucengxianshihanshu();

a[m]=0;

zhuangtaixianshihanshu();

if(m=3)

{m=m+1;}

open();

bujindianji();

close();

zhuangtaixianshihanshu();

m=m+1;

}

else if(a[m]==0)

{

zt=2;

mubiaoloucengxianshihanshu();

dangqianloucengxianshihanshu();

p=p+1;

m=m+1;

zhuangtaixianshihanshu();

bujindianji();

}

}

m=p-1;

while(m>=0)

{

byte data;

byte i, j;

byte k;

PortWriteByte(0x28b, 0x81); /\*设置8255的A口C口均为输出\*/

clear();

Sleep(50);

PortWriteByte(0x28a, 0x0f);

cmdsetup();

PortReadByte(0x28a, &data);

i = data;

if (i != 0x0f)

{

i = data;

Sleep(50);

PortWriteByte(0x28b, 0x88);

PortWriteByte(0x28a, 0xf0);

PortReadByte(0x28a, &data);

i = i | data;

printf("%d",i);

clear();

PortWriteByte(0x288, 0x91);

cmdsetup();

if (i == 0x67)

{

k = 1;

if(k<p)

{

b[k-1]=1;

}

}

else if (i == 0x6b)

{

k = 1;

if(k<p)

{b[k-1]=1;}

}

else if (i == 0x6d)

{

k = 2;

if(k<p)

{b[k-1]=1;}

else if(k>p)

{a[k-1]=1;}

}

else if (i == 0x6e)

{

k = 3;

if(k<p)

{b[k-1]=1;}

else if(k>p)

{a[k-1]=1;}

}

else if (i == 0xa7)

{

k = 4;

if(k<p)

{b[k-1]=1;}

else if(k>p)

{a[k-1]=1;}

}

else if (i == 0xab)

{

k = 2;

if(k<p)

{b[k-1]=1;}

else if(k>p)

{a[k-1]=1;}

}

else if (i == 0xad)

{

k = 2;

if(k<p)

{b[k-1]=1;}

else if(k>p)

{a[k-1]=1;}

}

else if (i == 0xae)

{

k = 3;

if(k<p)

{b[k-1]=1;}

else if(k>p)

{a[k-1]=1;}

}

else if (i == 0xc7)

{

k = 3;

if(k<p)

{b[k-1]=1;}

else if(k>p)

{a[k-1]=1;}

}

else if (i == 0xcb)

{

k = 4;

if(k>p)

{a[k-1]=1;}

}

}

panduanxiajiangshifoujieshu();

m=g;

if(m<0)

{break;}

if(b[m]==1)

{

zt=1;

mubiaoloucengxianshihanshu();

dangqianloucengxianshihanshu();

b[m]=0;

zhuangtaixianshihanshu();

open();

bujindianji();

close();

zhuangtaixianshihanshu();

m=m-1;

}

else if(b[m]==0)

{

zt=0;

mubiaoloucengxianshihanshu();

dangqianloucengxianshihanshu();

p=p-1;

m=m-1;

zhuangtaixianshihanshu();

bujindianji();

}

}

}

Cleanup();

}

5系统分析与测试

本实验采用HQFC开发板进行测试。

运行程序，进行以下测试：

1. 运行程序，观察LCD液晶显示屏，看是否对系统进行初始化；
2. 通过4\*4键盘按下一个大于当前楼层的楼层按键，观察LCD液晶显示屏的显 示、异步电机的转动、每上升一层的时间间隔；
3. 通过4\*4键盘按下一个小于当前楼层的楼层按键，观察LCD液晶显示屏的显 示、异步电机的转动、每下降一层的时间间隔；
4. 通过4\*4键盘按下一个介于当前楼层和目标楼层之间的楼层按键，观察LCD显示屏的显示。

结论：对上述测试依次进行，结果符合理论实际，基本实现题目要求，实现对电梯运行的模拟。

6系统运行结果

系统电路连接及运行结果如图所示：

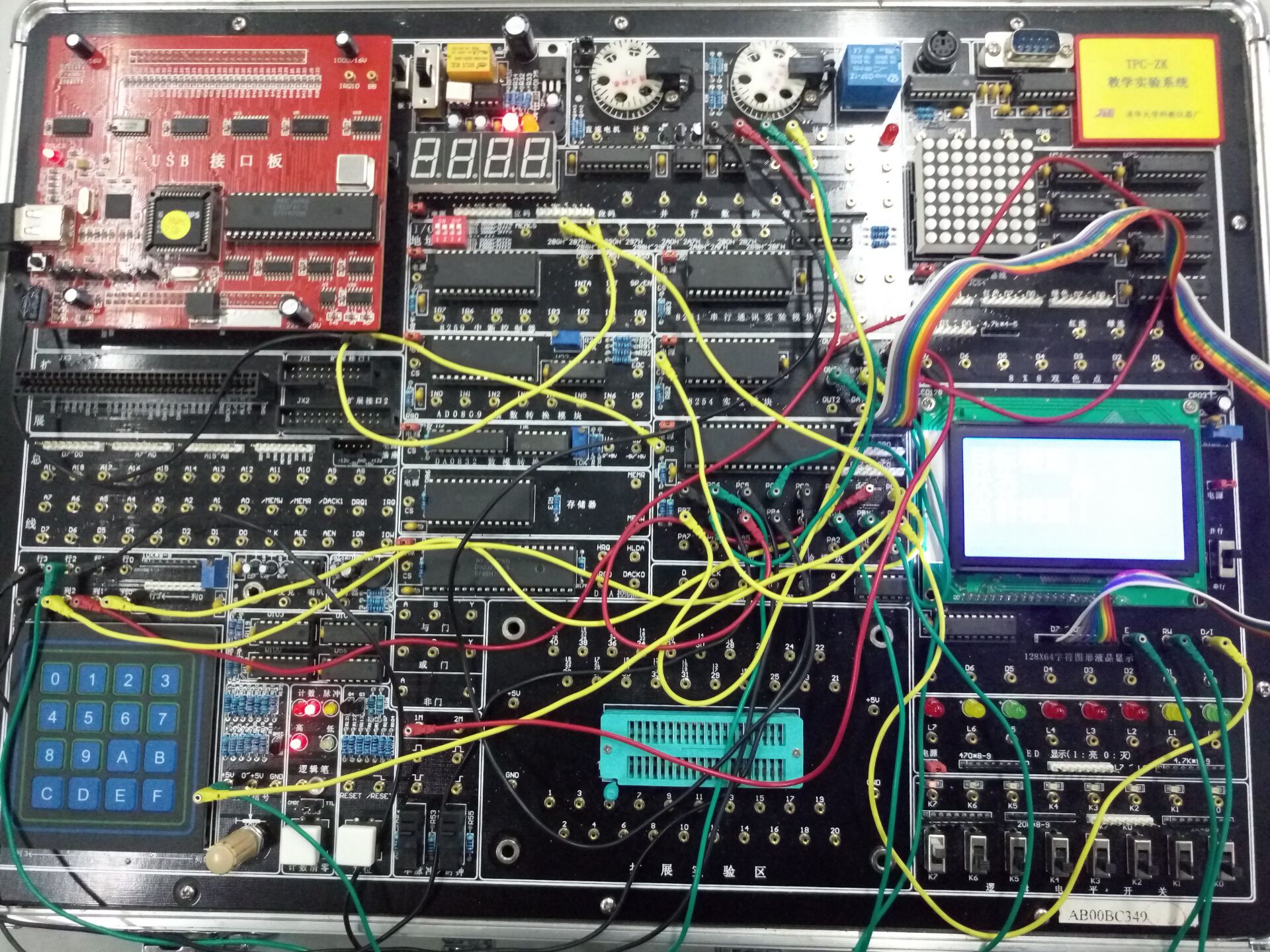


图6-1系统电路连接图



图6-2 系统初始化



图6-3 电梯上升图



图6-4 电梯下降图

检测中途插队情况:

首先电梯前往4楼



图6-5 电梯前往4楼图

中途三楼按下外键，目标楼层改为3楼

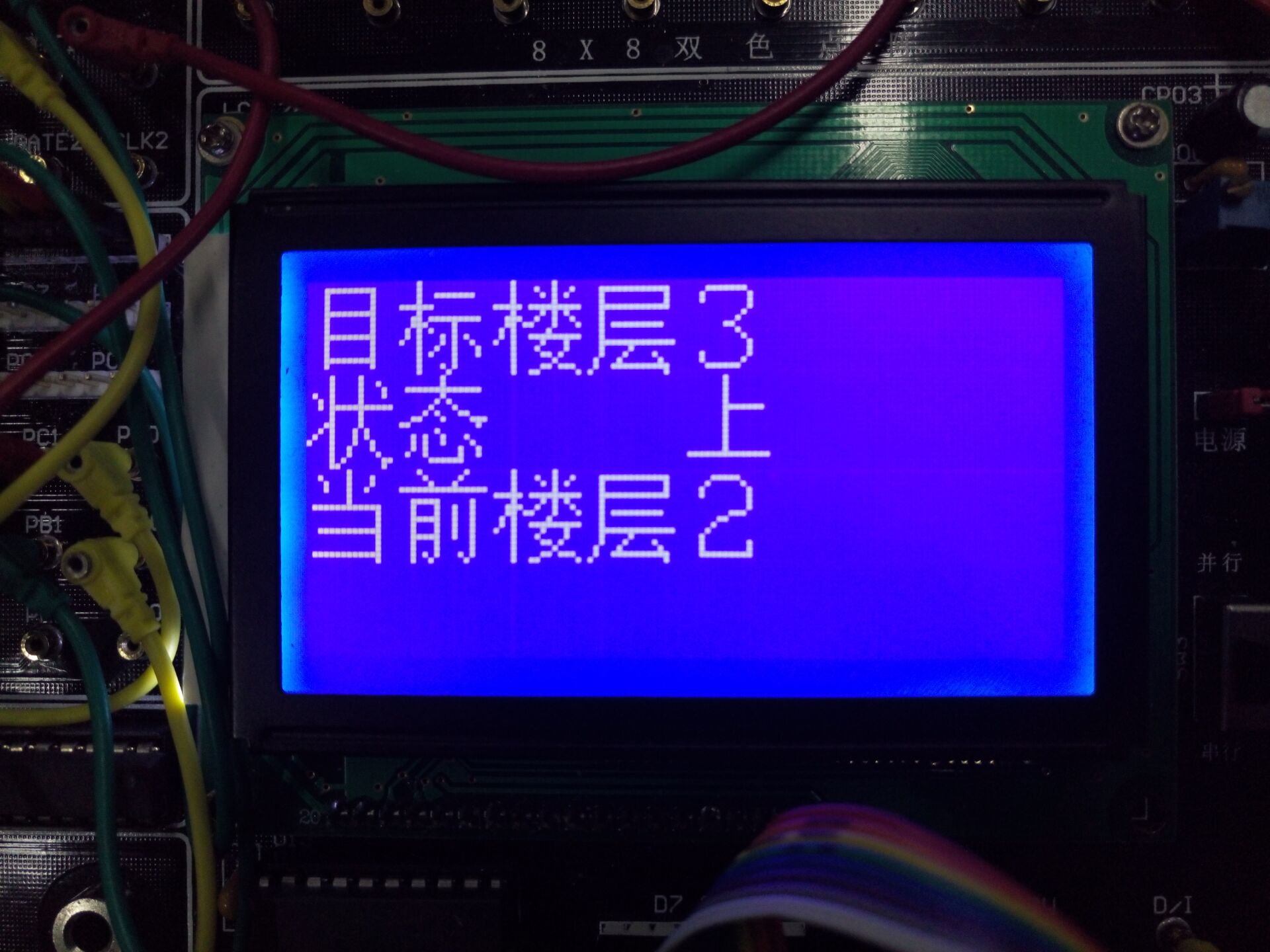


图6-6 目标楼层改变图

经检测，电梯系统运转正常。

7结论

经过一学期的课程，我成功的将电梯题目完成。对题目所要求的功能基本都已实现。

1. 电梯可上下运行，实现电梯的基本功能。

2）利用8253芯片实现1s延时。

3）通过4\*4键盘键入命令。

4）步进电机实现正反转来表示电梯上下状态。

5）用LCD液晶显示屏显示电梯运行状态。

6）用8255芯片控制系统的输入输出。

8设计体会

本次设计可以说是对我能力的一次检验，综合了之前所学的一些学科的内容。我之前的基础不好，实验进度也很慢，但我并没有放弃。由于仅有我一个人选了这个题目，对整个题目的设计均要一个人独立完成，遇到一些难题也只能一个人去查各种资料，默默学习。

对于这个题目，我没有一开始就在硬件上上手，而是先设计整个程序的主体部分，所需要硬件实现的部分先写好函数名而不写具体的函数，然后把程序的主体部分一点点的写出来。这一部分耗时挺久，因为有些地方有点模糊，不好把握。之后是对硬件部分的温习。对相应芯片的演示实验看了许多遍，尽量读透每个细节，然后进行具体的设计。最先开始设计的是4\*4键盘与8255芯片与LCD液晶显示屏之间的通信。这一部分耗时也很久，一直在尝试，但是总会出现一些问题，导致不成功。之后在不断的尝试中学会其中的原理，才发现如此简单。之后进行设计的延时函数。开始是使用的软延时，但是老师说要用8253芯片控制的延时，所以我又开始改进这个地方，还好这个比较简单，没用太多时间。然后添加的步进电机。之前我对步进电机的了解很少，所以设计起来遇到了不小的困难，后来在网上查阅资料和求助老师的情况下才完成了这部分的设计。最后是整个程序的组合。由于前面做的比较好，所以这部分没有花费太多时间。

在设计的时候，很多身边的同学都已经做完了，这给了我很大的精神压力，一度认为自己完不成这个实验设计了。还好，我没有放弃，在我坚持不懈的努力下，终于完成了整个题目。这使我对自己的能力也有了一定程度的肯定。之后一定会继续努力，决不放弃，一直走下去。另外，感谢老师和同学的帮助。

参考文献

【1】微型计算机原理及应用， 郑学坚、 周斌编著， 清华大学出版社，2010 年

【2】HQFC-A系列386EX教师实验指导书