

**《电气仿真软件》**

**课程设计报告**

|  |  |
| --- | --- |
| **专业** | 自动化 |
| **学生姓名** | 张航 |
| **班级** | B自动化181 |
| **学号** | 180106051026 |
| **指导教师** | 辅晓荣 |
| **完成日期** | 2021年7月9日 |

**盐城工学院电气学院**

**目录**

1.计算机语言与仿真软件的关系-------------------------------------1

2.仿真软件的作用与分类

2.1 Proteus仿真软件的特点-------------------------------------1

2.2 博图仿真软件的特点----------------------------------------2

2.3 Matlab Simulink仿真软件的特点-----------------------------3

3.仿真步骤

3.1单片机仿真的过程步骤---------------------------------------4

3.2 PLC仿真的过程步骤-----------------------------------------5

4.仿真流程图绘制

4.1单片机仿真的流程图绘制-------------------------------------8

4.2单片机仿真的语句分析注释-----------------------------------8

4.3 PLC仿真的实例流程图绘制----------------------------------14

4.4 PLC仿真的梯形图分析注释----------------------------------15

5.电气原理图

5.1单片机电气原理图------------------------------------------22

5.2 PLC电气原理图--------------------------------------------22

6.调试与仿真

6.1单片机仿真------------------------------------------------23

6.2 PLC仿真--------------------------------------------------24

7.小结----------------------------------------------------------26

参考文献--------------------------------------------------------27

附录

单片机：

附录1：程序清单------------------------------------------------28

附录2：设计图纸------------------------------------------------34

附录3：元器件目录表--------------------------------------------35

电气控制与PLC

附录1：装卸料小车程序-------------------------------------------36

附录2.1：装卸料小车PLC原理图-----------------------------------43

附录2.2：装卸料小车流程图---------------------------------------43

附录2.3：装卸料小车状态转移图-----------------------------------44

附录3：元器件清单-----------------------------------------------45

**电气仿真软件课程设计报告**

1. **计算机语言与仿真软件的关系**

计算机语言（Computer Language）指用于人与计算机之间通讯的语言。计算机语言是人与计算机之间传递信息的媒介。计算机系统最大特征是指令通过一种语言传达给机器。为了使电子计算机进行各种工作，就需要有一套用以编写计算机程序的数字、字符和语法规划，由这些字符和语法规则组成计算机各种指令（或各种语句）。这些就是计算机能接受的语言。

C语言是一门面向过程的、抽象化的通用程序设计语言，广泛应用于底层开发。C语言能以简易的方式编译、处理低级存储器。C语言是仅产生少量的机器语言以及不需要任何运行环境支持便能运行的高效率程序设计语言。尽管C语言提供了许多低级处理的功能，但仍然保持着跨平台的特性，以一个标准规格写出的C语言程序可在包括类似嵌入式处理器以及超级计算机等作业平台的许多计算机平台上进行编译。

仿真软件是仿真研究中基于通用计算机语言而又比其更高级的软件系统。仿真软件为仿真研究人员提供了专门用于建模、仿真实验和仿真结果统计、分析、显示的程序语句，研究人员可以不必深入掌握通用计算机高级语言的编程细节和技巧，而用比较习惯或容易掌握的表达方式来描述仿真模型，把主要精力集中在仿真研究上。

**2. 仿真软件的作用与分类**

仿真软件，专门用于仿真的计算机软件。它与仿真硬件同为仿真的技术工具。仿真软件是从50年代中期开始发展起来的。它的发展与仿真应用、算法、计算机和建模等技术的发展相辅相成。1984年出现了第一个以数据库为核心的仿真软件系统，此后又出现采用人工智能技术（专家系统）的仿真软件系统。这个发展趋势将使仿真软件具有更强、更灵活的功能、能面向更广泛的用户。目前比较风行的是虚拟现实仿真软件，比如虚拟现实仿真平台（VR-Platform）

**2.1 Proteus仿真软件的特点**

Proteus软件是英国Labcenter electronics公司出版的EDA工具软件。它不仅具有其它EDA工具软件的仿真功能，还能仿真单片机及外围器件。它是目前最好的仿真单片机及外围器件的工具。虽然目前国内推广刚起步，但已受到单片机爱好者、从事单片机教学的教师、致力于单片机开发应用的科技工作者的青睐。Proteus是世界上著名的EDA工具(仿真软件)，从原理图布图、代码调试到单片机与外围电路协同仿真，一键切换到PCB设计，真正实现了从概念到产品的完整设计。是目前世界上唯一将电路仿真软件、PCB设计软件和虚拟模型仿真软件三合一的设计平台。 使用Proteus 软件进行单片机系统仿真设计, 是虚拟仿真技术和计算机多媒体技术相结合的综合运用，有利于培养学生的电路设计能力及仿真软件的操作能力;在单片机课程设计和全国大学生电子设计竞赛中，我们使用 Proteus 开发环境对学生进行培训，在不需要硬件投入的条件下，学生普遍反映，对单片机的学习比单纯学习书本知识更容易接受，更容易提高。实践证明，在使用 Proteus 进行系统仿真开发成功之后再进行实际制作，能极大提高单片机系统设计效率。因此，Proteus 有较高的推广利用价值。

1. 提供软件调试功能。
2. 提供丰富的外围接口器件及其仿真。RAM，ROM，键盘，马达，LED，LCD，AD/DA，部分SPI器件，部分IIC器件。
3. 提供丰富的虚拟仪器，利用虚拟仪器在仿真过程中可以测量外围电路的特性，培养实际硬件的调试能力。
4. 具有强大的原理图绘制功能。

**2.2博图仿真软件的特点**

西门子PLC在工业控制领域有着广泛的应用，其中新一代的西门子PLC S7-1200系列和西门子PLC S7-1500系列的出现，为用户提供了更高端的配置和新的解决方法。对于新一代的西门子PLC，用户需要使用TIA博途软件STEP7对其进行程序的逻辑设计，通过TIA博途软件WinCC来对控制系统进行上位机界面的设计和组态。TIA博途软件是一个系统，里面包含有多种软件，可以满足用户在不同自动化控制系统中的各种需求。西门子博途软件STEP7，WinCC等为用户提供了强大的功能，用户可以根据项目中的实际需求进行选择。西门子博途软件系列为用户提供了很好的支持，使得用户的程序设计变得简单有效，并可以将整套自动化控制系统集成在一起进行操作和控制，极大的提高了工程效率，降低了维护成本。

1.可用性高

西门子为用户提供了完善的高可用性设计方案，保证工厂中的设备都具有较高的可用性，范围广泛；例如；对控制器进行现场测试，通过带自动事件同步的可靠切换，确保了高可用性。

2. 可靠性高

西门子系统中的产品都具有极佳的品质和较好的耐用性，适用于各种工业环境。产品通过了系统测试，能达到设计的目标水平，并且具有相关的认证。西门子产品同样规定了电磁兼容性等方面特性；

3. 安全性

　　在工业现场中，以太网有着广泛的应用，相应的出现了网络安全的问题。为了对工厂数据进行保护，需要采取大量不同的措施，从而确保PC和控制系统保护各个网络的自动化单元设备。西门子在这里采用单元保护方案，使用交换机系统模块等，提供各种组件以组成良好的保护单元。

**2.3 Matlab Simulink仿真软件的特点**

simulink是用来建模、分析和仿真各种动态系统的交互环境，提供了采用鼠标拖放的方法建立系统框图模型的交互界面。通过simulink提供的丰富的功能块，可以迅速地创建系统的模型，不需要书写一行行代码。

与其他高级程序设计语言相比较，matlab不但在数学语言的表达与解释方面表现出人机交互的高度一致，而且具有如下特点：

1. 基于向量、数组和矩阵的高级程序设计语言。

2. 界面友好、编程效率高。

3. 高级图形和可视化数据处理能力。

4. 广泛解决各学科专业领域内复杂问题的能力。

5. 拥有一个强大的仿真工具——simulink。

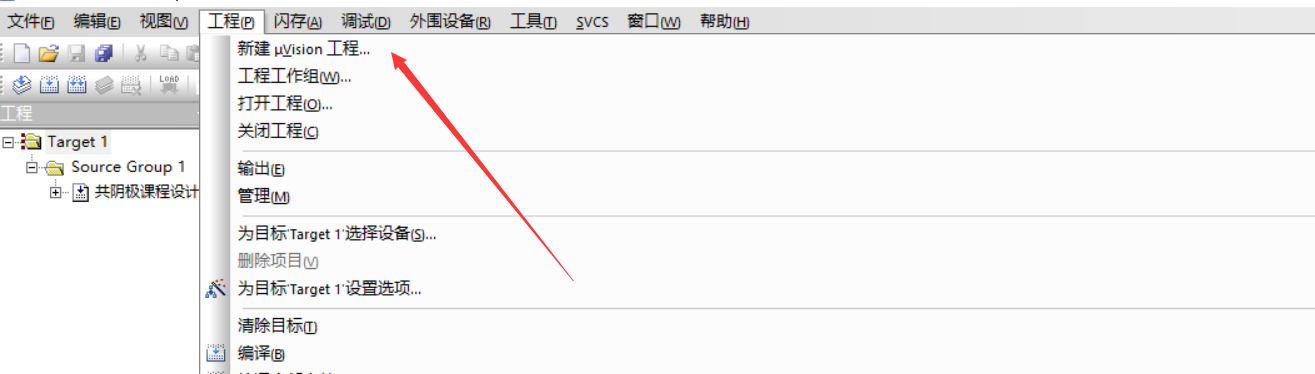
6. 支持科学和工程计算标准的开放式、可扩充结构。

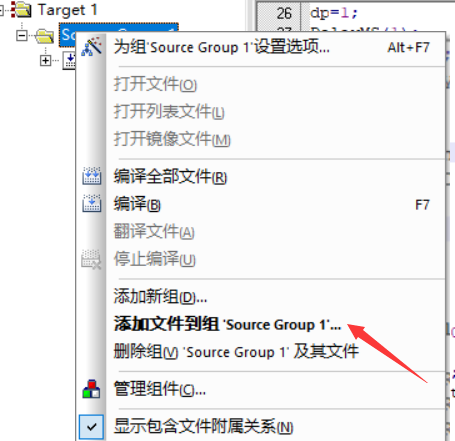
7. 跨平台兼容。

**3. 仿真步骤**

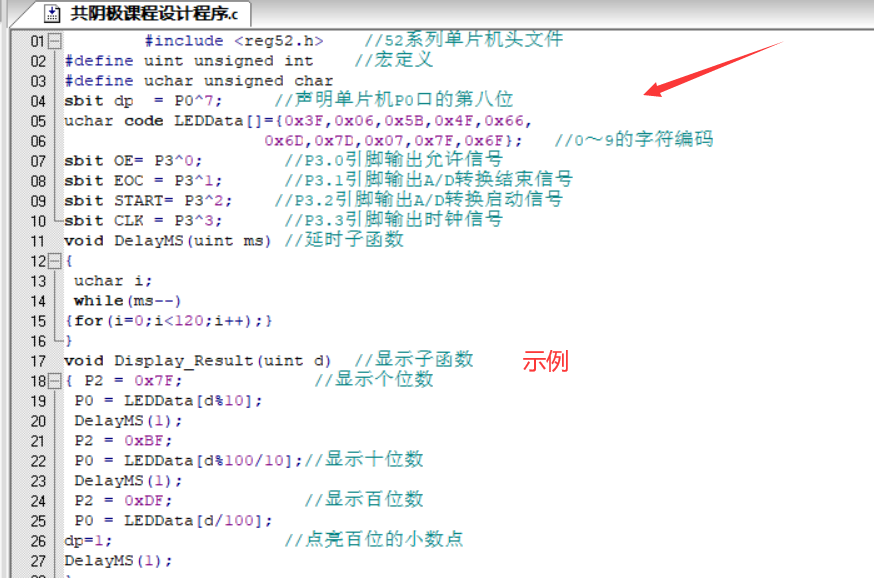
**3.1 单片机仿真的过程步骤**

1、打开keil,点击“工程”，创建新项目。

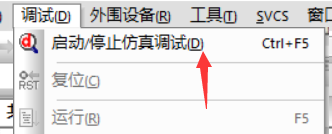


2、新建文件，导入到新建的工程项目中。

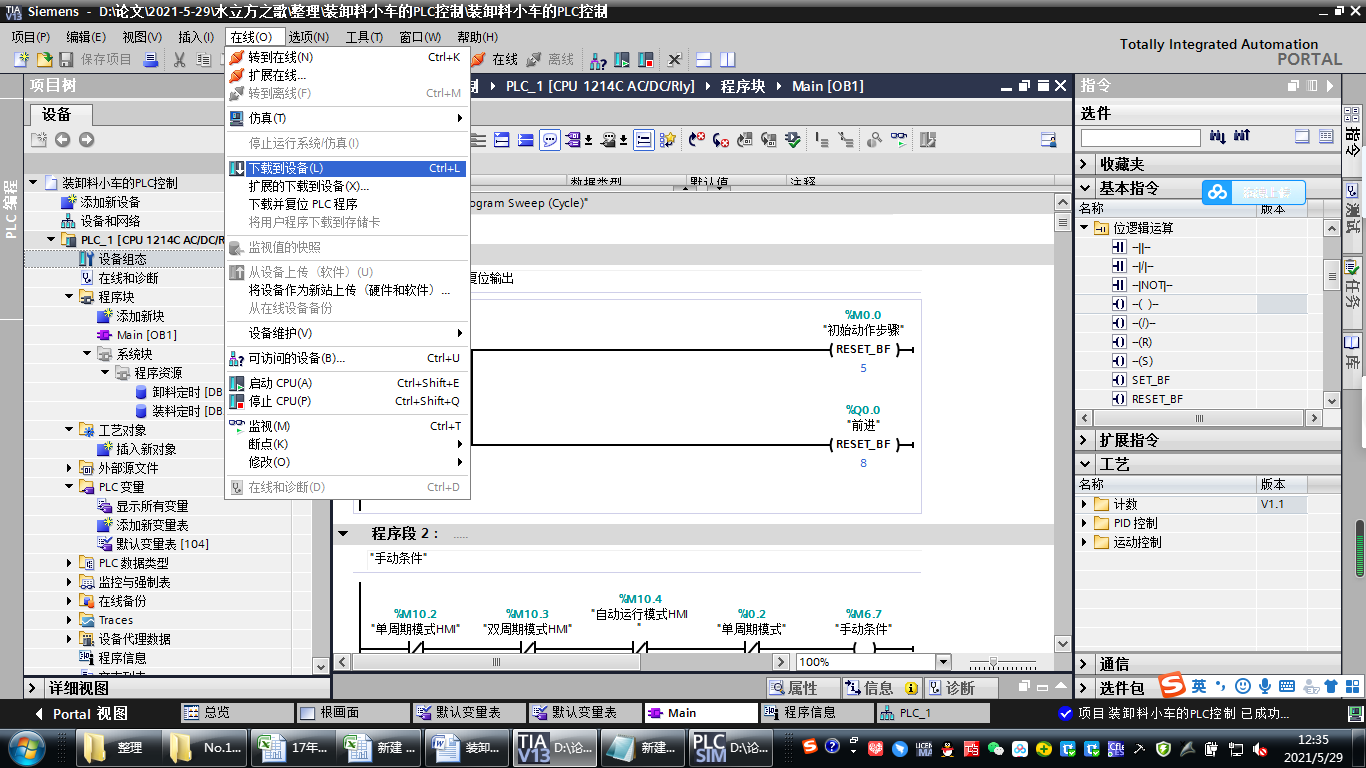


3、在之前的.c文件中，用C语言编写程序（示例如下）。

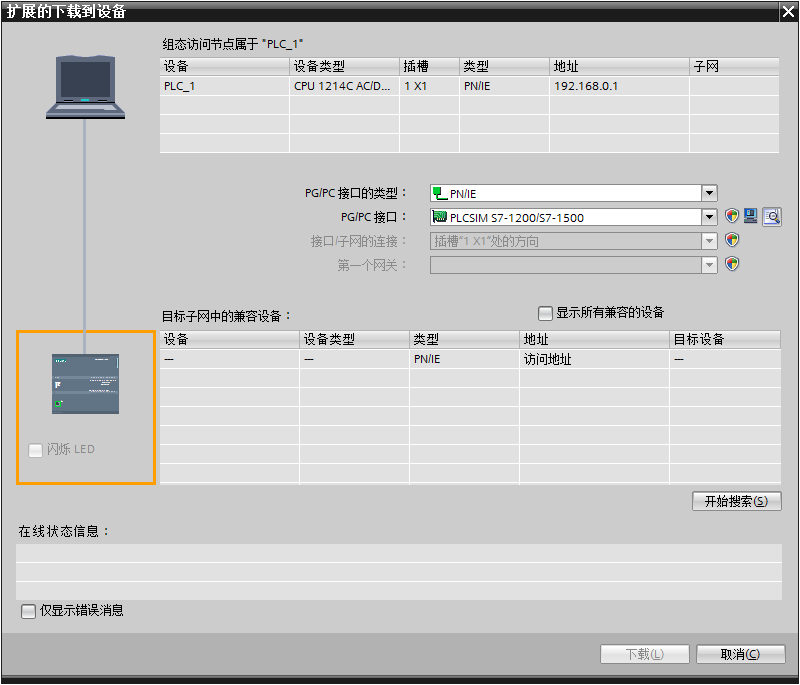
4、点击调试，启动仿真即可仿真单片机的程序

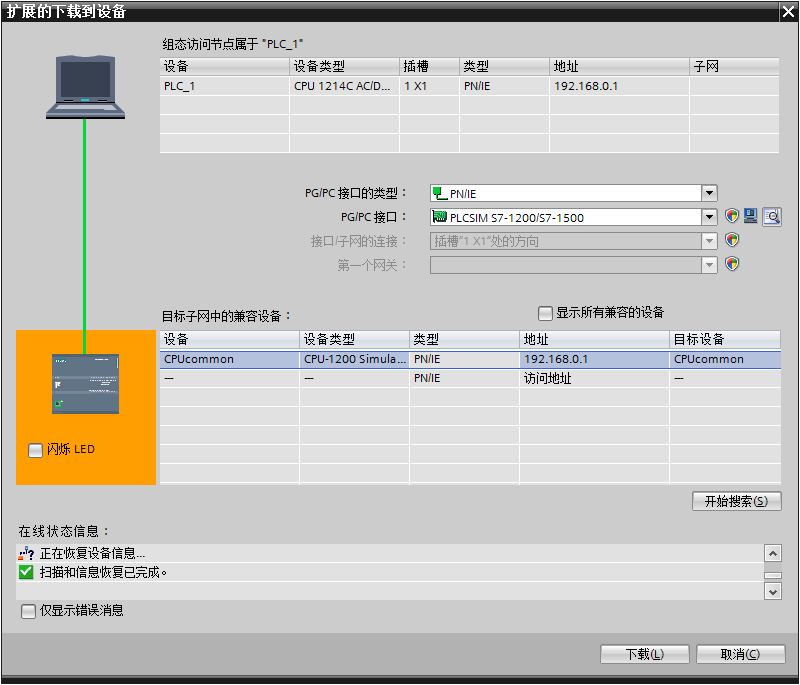


**3.2 PLC仿真的过程步骤**

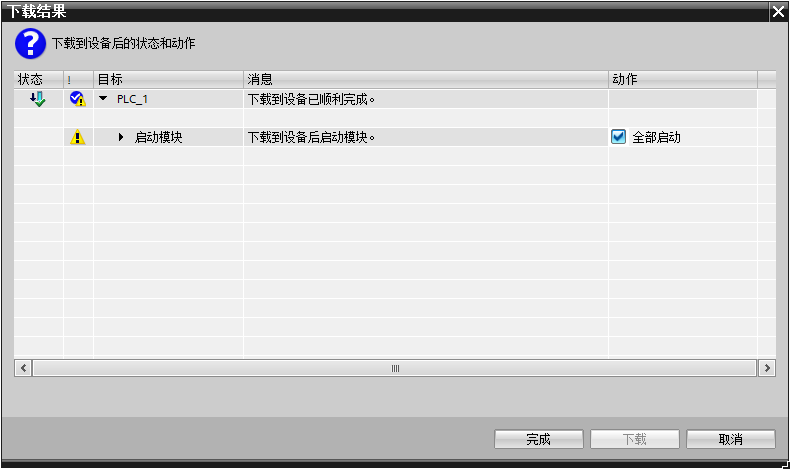
1、启动仿真PLC软件，进行编译，编程软件里点下载到设备，下载设备和程序。

2、打开扩展的下载到设备，选择PG、PC接口为PLCSIM S7-1200,1500点底部的搜索

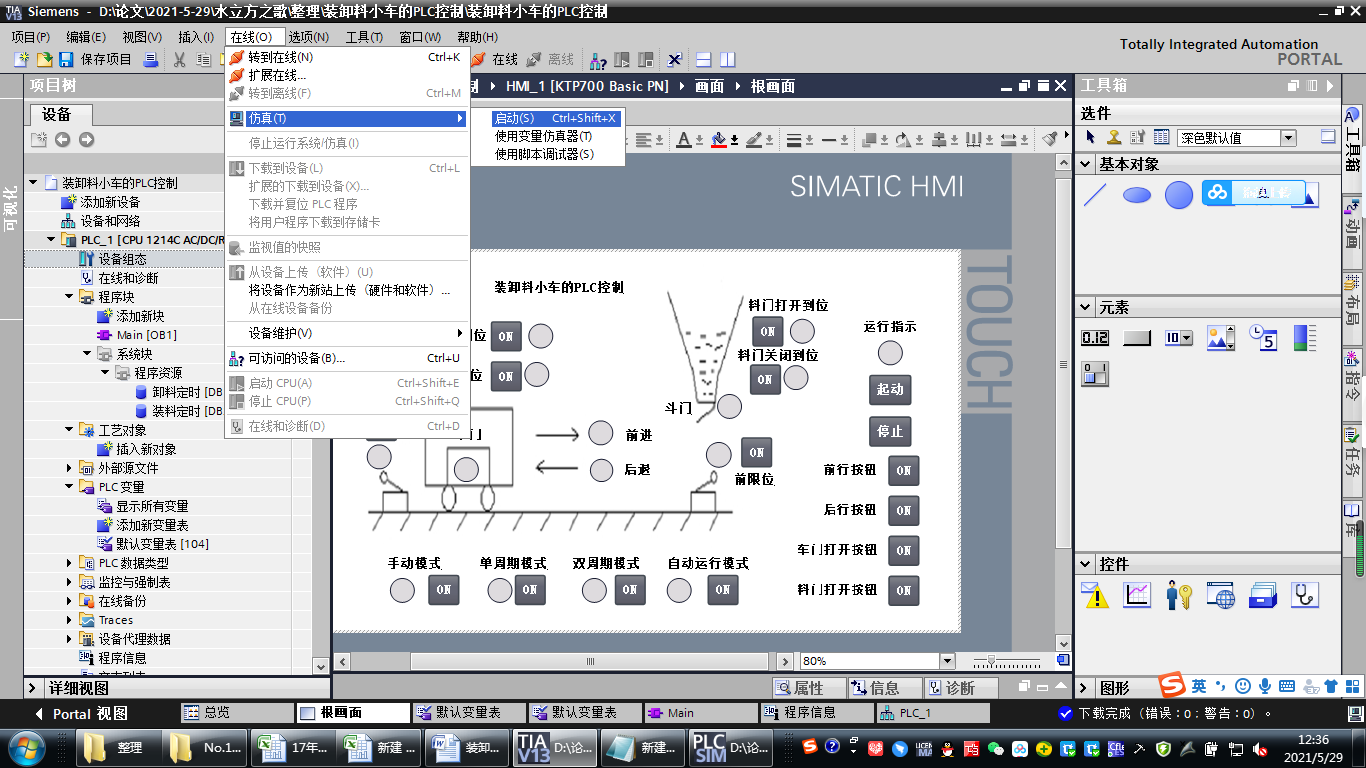


3、找到设备后，点CPU-1200，然后点下载

4、下载前选择全部覆盖，然后点下载

5、下载完成，点全部启动

6、触摸屏编辑画面，点在线菜单，点仿真，点启动，启动仿真触摸屏。



**4. 仿真流程图绘制**

**4.1 单片机仿真的流程图绘制**



**4.2 单片机仿真的语句分析注释**

uchar tCount=0;//一年中每个月的天数，2月的天数由年份决定

uchar MonthsDays[]={0,31,0,31,30,31,30,31,31,30,31,30,31};

uchar \*WEEK[]={"SUN","MON","TUS","WEN","THU","FRI","SAT"};//周日，周一到周六

uchar LCD\_DSY\_BUFFER1[]={"Date 00-00-00 "}; //LCD显示缓冲

uchar LCD\_DSY\_BUFFER2[]={" Time00-00-00 "};

uchar DateTime[7]; //所读取的日期时间

char Adjust\_Index=-1; //当前调节的时间：秒，分，时，日，

uchar Change\_Flag[]="-MHDM-Y";

uchar Read\_LCD\_State();

void LCD\_Busy\_Wait();

void Write\_LCD\_Data(uchar dat);

void Write\_LCD\_Command(uchar cmd);

void Init\_LCD();

void Set\_LCD\_POS(uchar p);

void DelayMS(uint x) //延时程序

{ uchar i;

while(x--) for(i=0;i<120;i++); }

uchar Read\_LCD\_State() //读LCD1602状态

{ uchar state;

RS=0;

RW=1;

EN=1;

DelayMS(1);

state=P0;

EN=0;

DelayMS(1);

return state;

}

void LCD\_Busy\_Wait()

{

while((Read\_LCD\_State()&0x08)==0x80);

DelayMS(5);

}

void Write\_LCD\_Data(uchar dat) //向LCD1602写数据

{

LCD\_Busy\_Wait();

RS=1;

RW=0; //选择写数据模式

EN=0;

P0=dat; //将要写的数据送入数据总线

EN=1; //将使能端置给一个高脉冲

DelayMS(1);

EN=0; //将使能端置0以完成高脉冲

}

void Write\_LCD\_Command(uchar cmd)// 向LCD1602写命令

{

LCD\_Busy\_Wait();

RS=0;

RW=0;//选择写命令模式

EN=0;

P0=cmd;//将要写的命令字送入数据总线

EN=1;//将使能端置给一个高脉冲

DelayMS(1);

EN=0; //将使能端置0以完成高脉冲

}

void Init\_LCD() //LCD初始化

{ Write\_LCD\_Command(0x38); DelayMS(1);

Write\_LCD\_Command(0x01); DelayMS(1);

Write\_LCD\_Command(0x06); DelayMS(1);

Write\_LCD\_Command(0x0C); DelayMS(1);

Display\_LCD\_String(0x00," dian zi ri li ");

DelayMS(2000);

}

void Set\_LCD\_POS(uchar p)

{

Write\_LCD\_Command(p|0x80);

}

void Display\_LCD\_String(uchar p,uchar \*s)

{ uchar i;

Set\_LCD\_POS(p);

for(i=0;i<16;i++)

{ Write\_LCD\_Data(s[i]);

DelayMS(1);}

}

void DS1302\_Write\_Byte(uchar x) //向DS1302写入一字节

{ uchar i;

for(i=0;i<8;i++)

{SDA=x&1;

CLK=1;

CLK=0;

x>>=1;

}

}

uchar DS1302\_Read\_Byte() //从DS1302读取一字节

{ uchar i,b,t;

for(i=0;i<8;i++)

{ b>>=1;

t=SDA;

b|=t<<7;

CLK=1;

CLK=0; }

return b/16\*10+b%16; }

uchar Read\_Data(uchar addr) //从DS1302指定位置读数据

{ uchar dat;

RST=0;

CLK=0;

RST=1;

DS1302\_Write\_Byte(addr);

dat=DS1302\_Read\_Byte();

CLK=1;

RST=0;

return dat;}

void Write\_DS1302(uchar addr,uchar dat) //向DS1302指定位置写入数据

{ CLK=0;

RST=1;

DS1302\_Write\_Byte(addr);

DS1302\_Write\_Byte(dat);

CLK=0;

RST=0; }

void SET\_DS1302() //设置时间

{ uchar i;

Write\_DS1302(0x8e,0x00);

for(i=0;i<7;i++)

{

Write\_DS1302(0x80+2\*i,(DateTime[i]/10<<4|(DateTime[i]%10)));

}

Write\_DS1302(0x8e,0x80);

}

void GetTime() //读取当前时期时间

{ uchar i;

for(i=0;i<7;i++)

{

DateTime[i]=Read\_Data(0x81+2\*i);

}

}

void Format\_DateTime(uchar d,uchar \*a) //时间和日期转换成数字字符

{

a[0]=d/10+'0';a[1]=d%10+'0';

}

uchar isLeapYear(uint y) //判断是否为闰年

{

return (y%4==0&&y%100!=0)||(y%400==0);

}

void RefreshWeekDay() //星期转换

{ uint i,d,w=5;

for(i=2000;i<2000+DateTime[6];i++)

{ d=isLeapYear(i) ? 366 : 365;

w=(w+d)%7; }

d=0;

for(i=1;i<DateTime[4];i++) d+=MonthsDays[i];

d+=DateTime[3];

DateTime[5]=(w+d)%7+1;

}

void DateTime\_Adjust(char x) //年月日时分秒++/--

{ switch (Adjust\_Index)

{ case 6: //年

if(x== 1&&DateTime[6]<99) DateTime[6]++;

if(x==-1&&DateTime[6]>0) DateTime[6]--;

MonthsDays[2]=isLeapYear(2000+DateTime[6])? 29:28;

if(DateTime[3]>MonthsDays[DateTime[4]])

DateTime[3]=MonthsDays[DateTime[4]];

RefreshWeekDay();

break;

case 4: //月

if(x== 1&&DateTime[4]<12) DateTime[4]++;

if(x==-1&&DateTime[4]>1) DateTime[4]--;

MonthsDays[2]=isLeapYear(2000+DateTime[6])? 29:28;

if(DateTime[3]>MonthsDays[DateTime[4]])

DateTime[3]=MonthsDays[DateTime[4]];

RefreshWeekDay();

break;

case 3: //日

MonthsDays[2]=isLeapYear(2000+DateTime[6])? 29:28;

if(x== 1&&DateTime[3]<MonthsDays[DateTime[4]]) DateTime[3]++;

if(x==-1&&DateTime[3]>0) DateTime[3]--;

RefreshWeekDay();

break;

case 2: //时

if(x== 1&&DateTime[2]<23) DateTime[2]++;

if(x==-1&&DateTime[2]>0) DateTime[2]--;

break;

case 1: //秒

if(x== 1&&DateTime[1]<59) DateTime[1]++;

if(x==-1&&DateTime[1]>0) DateTime[1]--;

break;}

}

void T0\_INT() interrupt 1 //定时器0中断，定时器每秒刷新LCD显示

{ TH0=-50000/256;

TL0=-50000%256;

if(++tCount!=2) return;

tCount=0;

Format\_DateTime(DateTime[6],LCD\_DSY\_BUFFER1+5);

Format\_DateTime(DateTime[4],LCD\_DSY\_BUFFER1+8);

Format\_DateTime(DateTime[3],LCD\_DSY\_BUFFER1+11);

strcpy(LCD\_DSY\_BUFFER1+13,WEEK[DateTime[5]-1]);

Format\_DateTime(DateTime[2],LCD\_DSY\_BUFFER2+5);

Format\_DateTime(DateTime[1],LCD\_DSY\_BUFFER2+8);

Format\_DateTime(DateTime[0],LCD\_DSY\_BUFFER2+11);

Display\_LCD\_String(0x00,LCD\_DSY\_BUFFER1);

Display\_LCD\_String(0x40,LCD\_DSY\_BUFFER2); }

void EX\_INT0() interrupt 0 //键盘中断

{ if(K1==0) //选择调整对象：年，月，日，时，分，秒

{ while(K1==0);

if(Adjust\_Index==-1||Adjust\_Index==1) Adjust\_Index=7;

Adjust\_Index--;

if(Adjust\_Index==5) Adjust\_Index=4;

LCD\_DSY\_BUFFER2[13]='[';

LCD\_DSY\_BUFFER2[14]=Change\_Flag[Adjust\_Index];

LCD\_DSY\_BUFFER2[15]=']'; }

else if(K2==0) // 加

{

while (K2==0); DateTime\_Adjust(1);

}

else if (K3==0) // 减

{

while (K3==0); DateTime\_Adjust(-1);

}

else if (K4==0) // 确定

{ while(K4==0);

SET\_DS1302(); //调整后的时间写入DS1302

LCD\_DSY\_BUFFER2[13]=' ';

LCD\_DSY\_BUFFER2[14]=' ';

LCD\_DSY\_BUFFER2[15]=' ';

Adjust\_Index=-1; }

}

void main() //主程序

{ Init\_LCD(); //LCD初始化

IE=0x83; // 允许INT0，T0中断

IP=0x01;

IT0=0x01;

TMOD=0x01;//设置定时器0为工作方式1（M1M0为01）

TH0=-50000/256; //装初值

TL0=-50000%256;

TR0=1;

while(1)

{

if(Adjust\_Index==-1) GetTime();

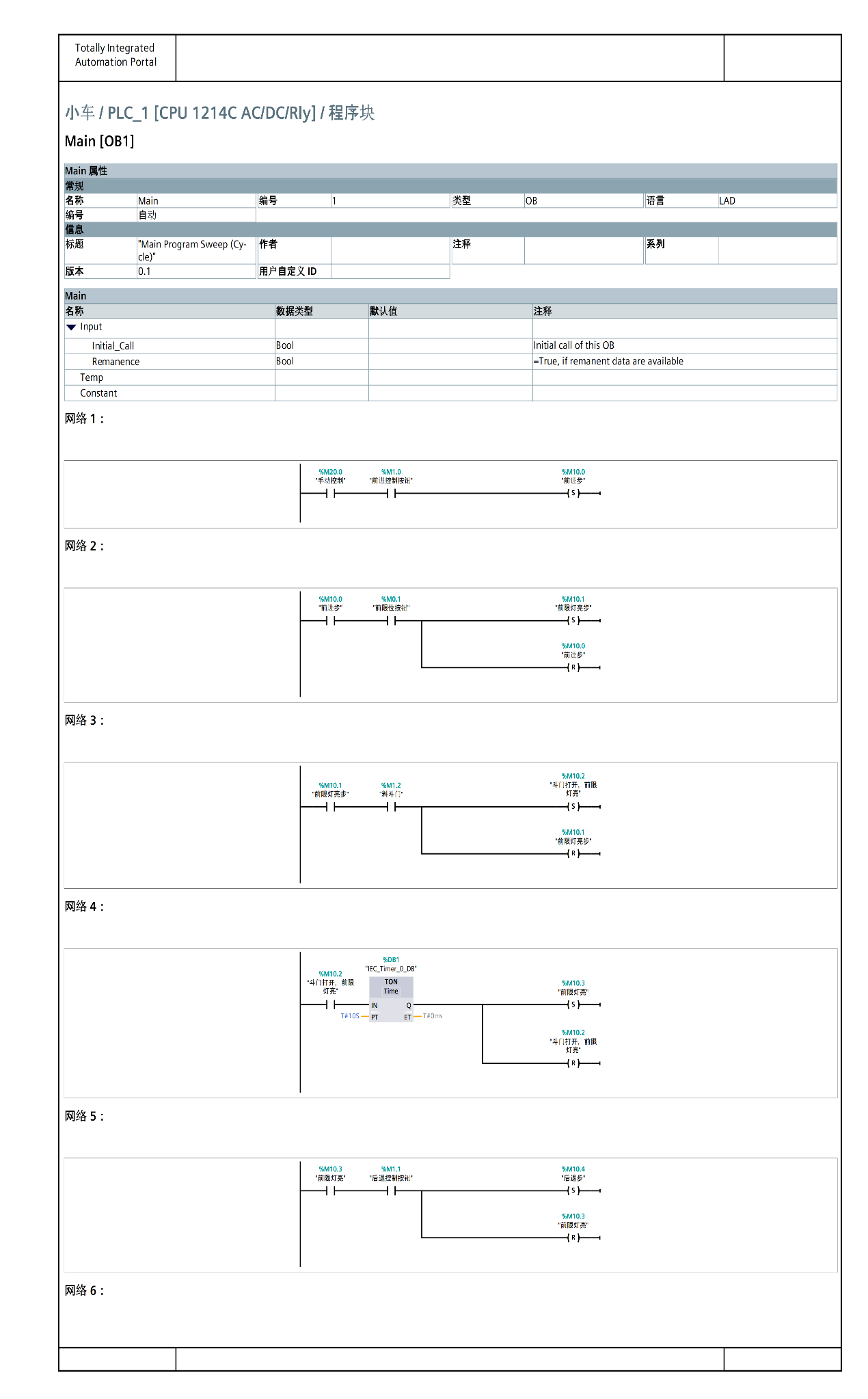
}

}

**4.3 PLC仿真的实例流程图绘制**



**4.4 PLC仿真的梯形图分析注释**



图示

描述已自动生成

箱线图

描述已自动生成

表格

低可信度描述已自动生成

图形用户界面

低可信度描述已自动生成

图示, 箱线图

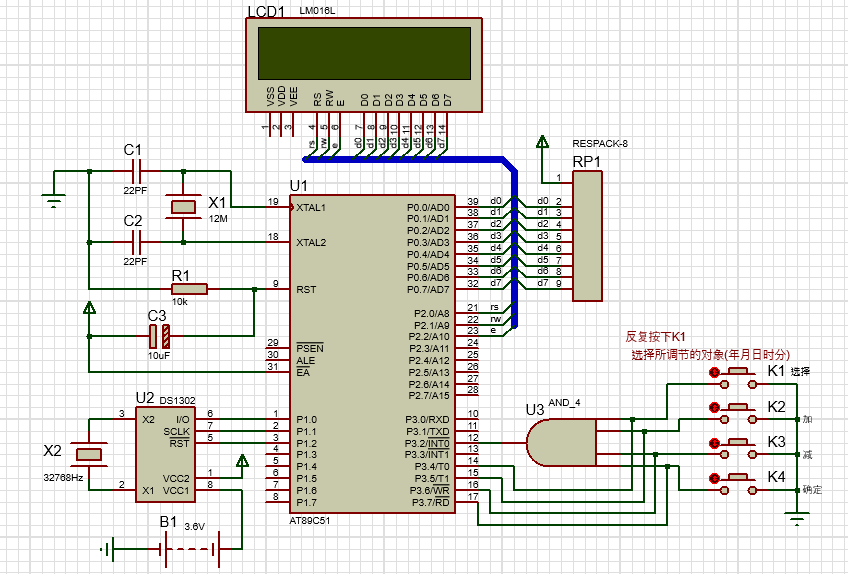
描述已自动生成

图表

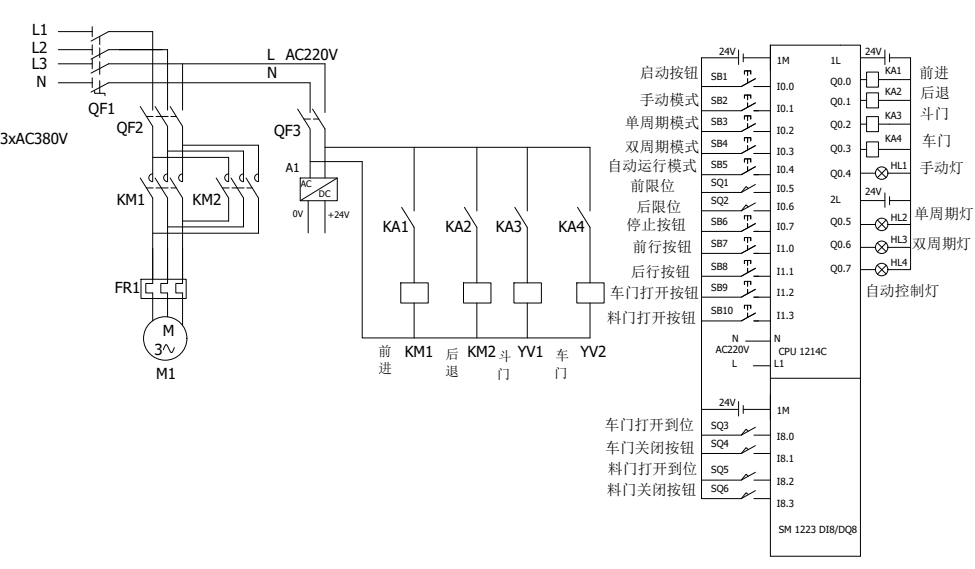
中度可信度描述已自动生成

**5. 电气原理图**

**5.1 单片机电气原理图**

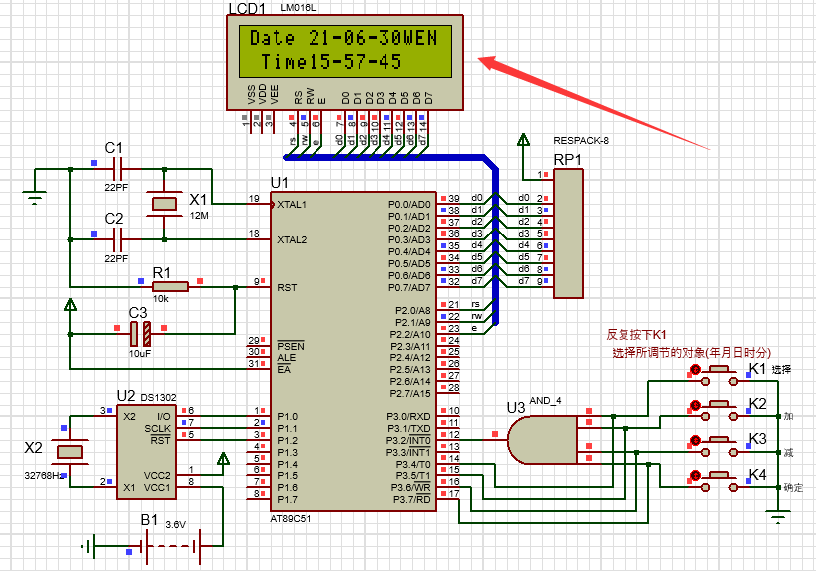
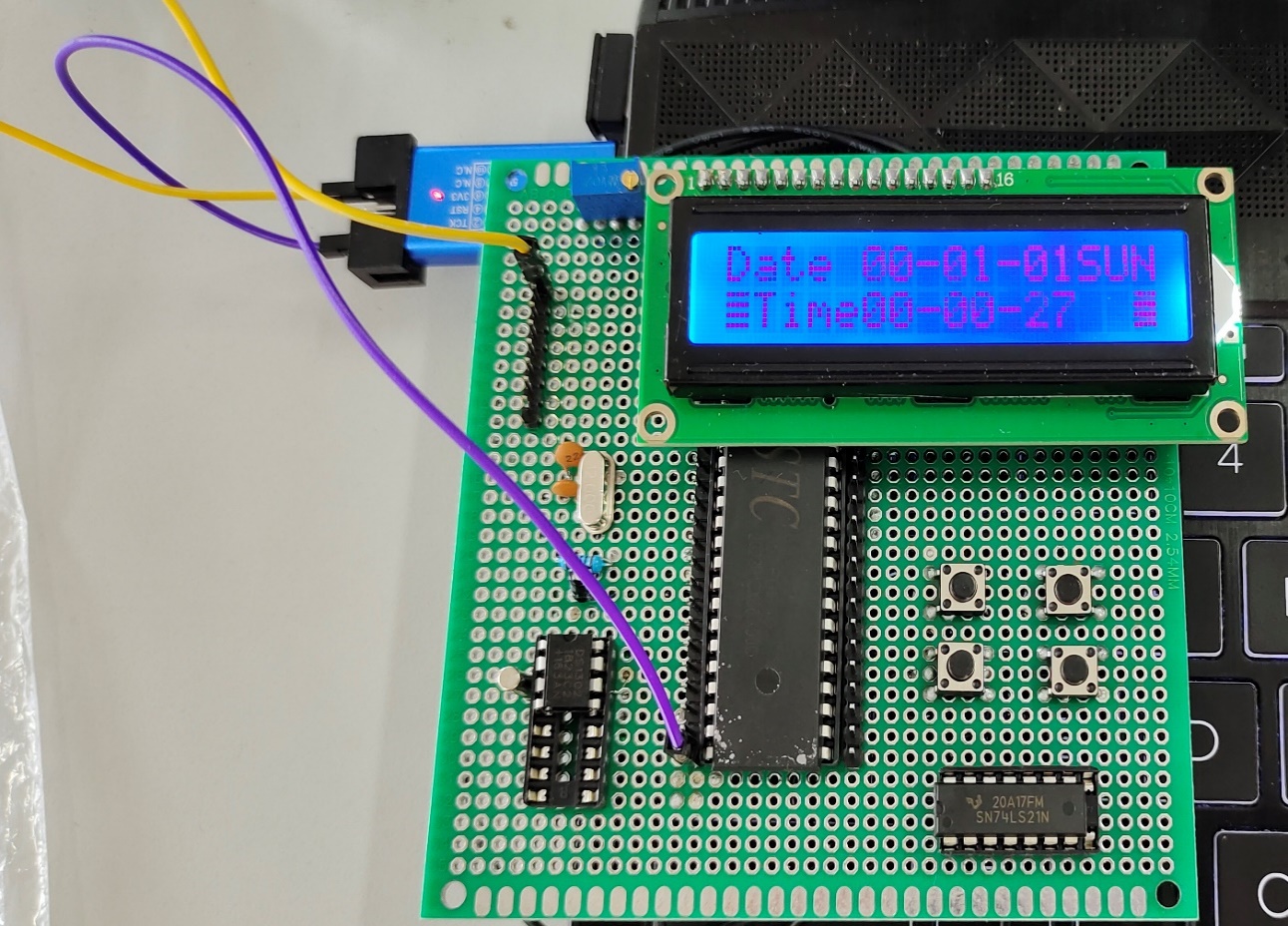


**5.2 PLC电气原理图**

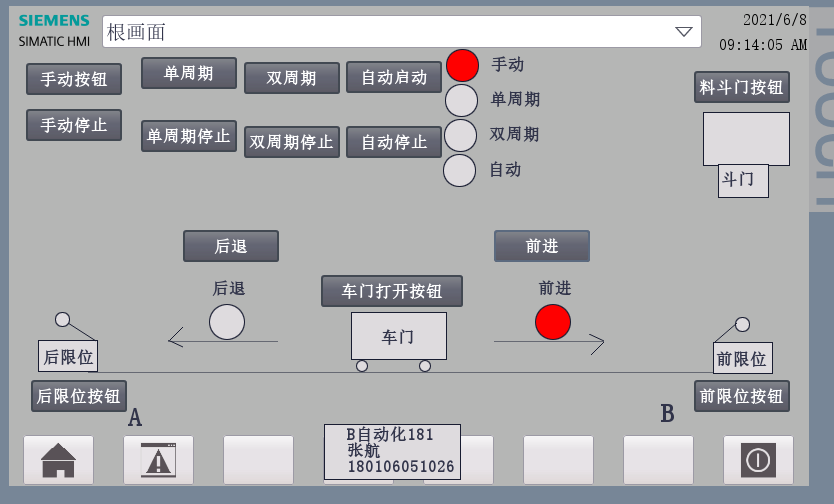


**6. 调试与仿真**

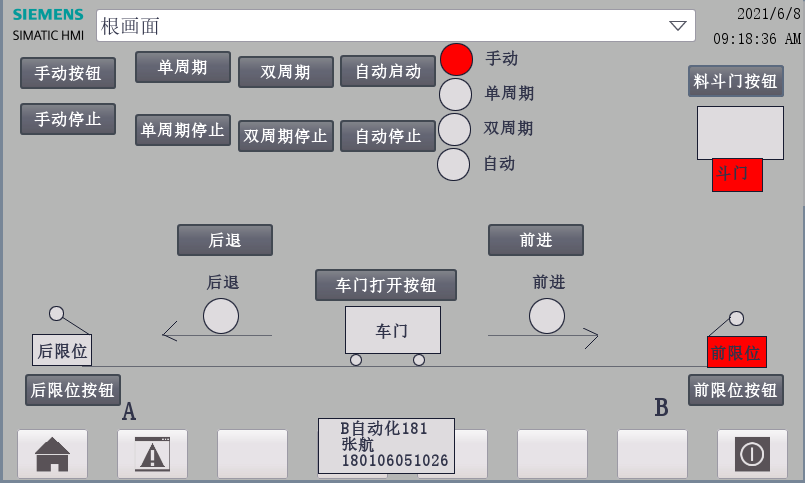
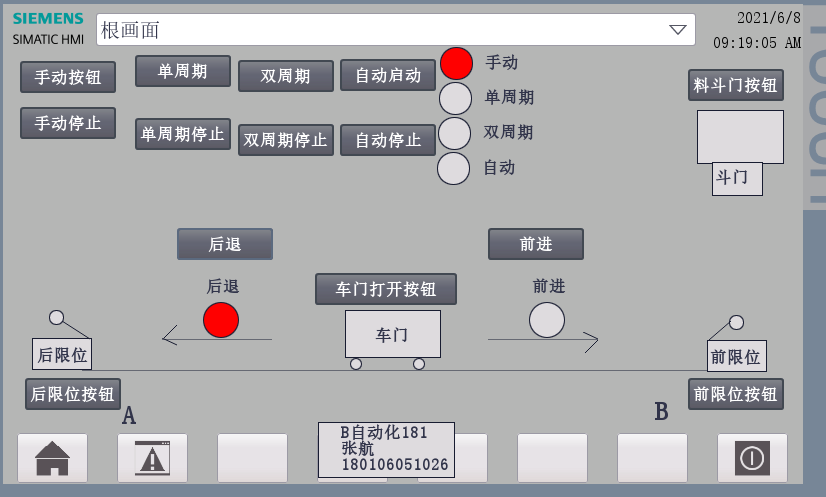
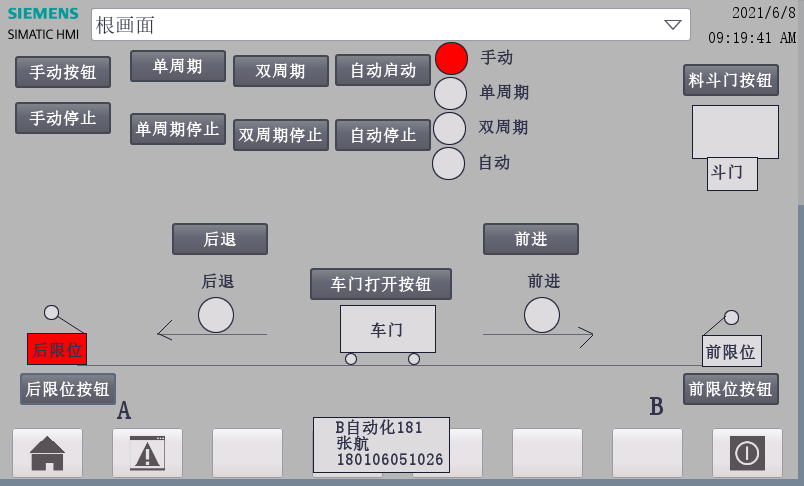
**6.1 单片机仿真**

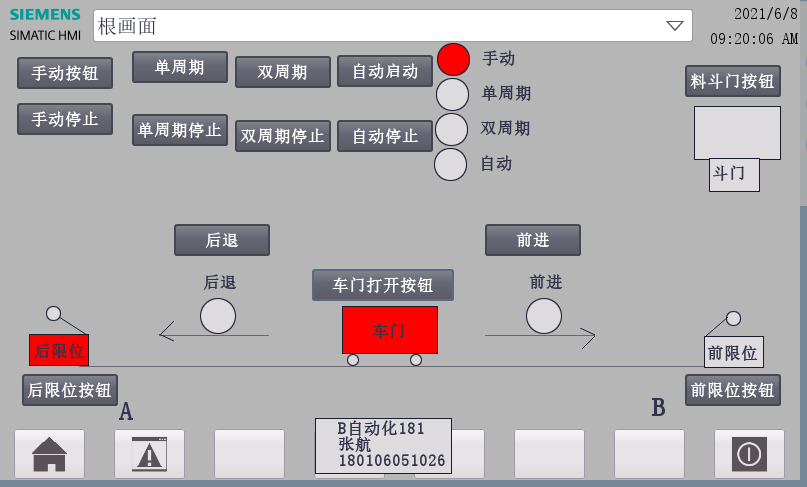


**6.2 PLC仿真**

触摸屏仿真







**7. 小结**

1、本次实验虽然需要设计四种控制卸料小车的模式，但是只要手动模式搞懂了，其它三种都可以很轻松的解决！在触摸屏的设计中我遇到了自己编写的程序出现了很多bug,每次调试完都要再三的修改自己的程序，力求能够做的特别的严谨！

2、在安装博图软件的时候，总是遇到安装的时候会要求重启，只要将自己电脑一段路径的注册表文件删完之后，再次安装就不会出现像这样的问题。刚开始装完软件的时候，自己编写程序完成后点击编译发现软件没有反应，然后我去了西门子的官方论坛去看了看发现很多人也遇到了和我类似的问题，将软件全部卸载掉，然后重装！这时候我特别想说一句：“西门子的软件真的很矫情”，经过重新安装之后，编译功能终于正常了，我于是就开始了我的触摸屏调试的旅程！

3、在使用CAD和visio作图时，自己也遇到了很多问题，比如CAD的很多快捷指令啊，如何将自己画好的图放进老师提供的模板里面，visio如何画好自己本次课程设计的状态转移图，经过自己几天的不断查阅资料，看相关的视频，自己独立画了出来！

4、在单片机电子日历的设计过程中，最让我头疼的莫过于焊板子了，尤其是我这边用到的LCD1602液晶显示屏，它有16个引脚，每个引脚都有自己的功能，这些都是要查阅手册才能知道的，但是在实际过程中，我焊完板子之后，我的显示屏是不亮的，通过查找资料得知，我的背光A,K这两个引脚没有接，接完之后，LCD显示屏成功点亮，但是新的问题有出现了，LCD只亮，但是什么东西都显示不出来，通过询问同学，得加个滑动电阻来调节对比度，加完之后，这次的焊板子之旅成功结束，所有功能和模拟仿真调试的结果相一致！

**参考文献**

【1】 刘瑞新 . 单片机原理及应用教程 . 北京 . 机械工业出版社 ,2003

【2】 李群芳 .单片微型计算机与接口技术 .电子工业出版 ,2012

【3】 审庆阳 .郭庭杰 .单片机的实践与应用 .清华大学出版 ,2002

【4】 沙占友 .王彦朋 .孟志永 .单片机外围电路设计 .北京 .电子工业出版社 ,2003

【5】张兰红，陆广平，仓思雨. 单片机课程设计仿真与实践指导[M].北京：机械工业出版社，2018

【6】陈盛开. 装卸料小车运行的总体框架结构及PLC控制系统设计[J]. 中国高新区, 2018, 000(002):23.

【7】周甫庆. 选矿厂卸料小车的PLC控制设计[J]. 有色金属设计, 1998(02):12-13.

【8】苏方臣, 刘振, 刘波. 基于PLC的皮带卸料小车自动布料控制系统设计[J]. 工业控制计算机, 2015, 000(004):167-168.

【9】缪雪峰. 基于iFIX及PLC的控制系统在选矿布料中的应用[J]. 矿山机械, 2010, 038(005):97-101.

**附录1：程序清单**

#include <reg52.h> //52系列单片机头文件

#include <string.h>

#include<intrins.h>

#define uchar unsigned char //宏定义

#define uint unsigned int

sbit SDA=P1^0; // DS1302数据线

sbit CLK=P1^1; //DS1302时钟线

sbit RST=P1^2; //DS1302复位线

sbit RS=P2^0; //数据命令选择端（H/L）

sbit RW=P2^1; //读写选择端（H/L）

sbit EN=P2^2; //使能信号

sbit K1=P3^4; // 选择

sbit K2=P3^5; // 加

sbit K3=P3^6; // 减

sbit K4=P3^7; // 确定

void Display\_LCD\_String(uchar p,uchar \*s);

uchar tCount=0;//一年中每个月的天数，2月的天数由年份决定

uchar MonthsDays[]={0,31,0,31,30,31,30,31,31,30,31,30,31};

uchar \*WEEK[]={"SUN","MON","TUS","WEN","THU","FRI","SAT"};//周日，周一到周六

uchar LCD\_DSY\_BUFFER1[]={"Date 00-00-00 "}; //LCD显示缓冲

uchar LCD\_DSY\_BUFFER2[]={" Time00-00-00 "};

uchar DateTime[7]; //所读取的日期时间

char Adjust\_Index=-1; //当前调节的时间：秒，分，时，日，

uchar Change\_Flag[]="-MHDM-Y";

uchar Read\_LCD\_State();

void LCD\_Busy\_Wait();

void Write\_LCD\_Data(uchar dat);

void Write\_LCD\_Command(uchar cmd);

void Init\_LCD();

void Set\_LCD\_POS(uchar p);

void DelayMS(uint x) //延时程序

{ uchar i;

while(x--) for(i=0;i<120;i++); }

uchar Read\_LCD\_State() //读LCD1602状态

{ uchar state;

RS=0;

RW=1;

EN=1;

DelayMS(1);

state=P0;

EN=0;

DelayMS(1);

return state;

}

void LCD\_Busy\_Wait()

{

while((Read\_LCD\_State()&0x08)==0x80);

DelayMS(5);

}

void Write\_LCD\_Data(uchar dat) //向LCD1602写数据

{

LCD\_Busy\_Wait();

RS=1;

RW=0; //选择写数据模式

EN=0;

P0=dat; //将要写的数据送入数据总线

EN=1; //将使能端置给一个高脉冲

DelayMS(1);

EN=0; //将使能端置0以完成高脉冲

}

void Write\_LCD\_Command(uchar cmd)// 向LCD1602写命令

{

LCD\_Busy\_Wait();

RS=0;

RW=0;//选择写命令模式

EN=0;

P0=cmd;//将要写的命令字送入数据总线

EN=1;//将使能端置给一个高脉冲

DelayMS(1);

EN=0; //将使能端置0以完成高脉冲

}

void Init\_LCD() //LCD初始化

{ Write\_LCD\_Command(0x38); DelayMS(1);

Write\_LCD\_Command(0x01); DelayMS(1);

Write\_LCD\_Command(0x06); DelayMS(1);

Write\_LCD\_Command(0x0C); DelayMS(1);

Display\_LCD\_String(0x00," dian zi ri li ");

DelayMS(2000);

}

void Set\_LCD\_POS(uchar p)

{

Write\_LCD\_Command(p|0x80);

}

void Display\_LCD\_String(uchar p,uchar \*s)

{ uchar i;

Set\_LCD\_POS(p);

for(i=0;i<16;i++)

{ Write\_LCD\_Data(s[i]);

DelayMS(1);}

}

void DS1302\_Write\_Byte(uchar x) //向DS1302写入一字节

{ uchar i;

for(i=0;i<8;i++)

{SDA=x&1;

CLK=1;

CLK=0;

x>>=1;

}

}

uchar DS1302\_Read\_Byte() //从DS1302读取一字节

{ uchar i,b,t;

for(i=0;i<8;i++)

{ b>>=1;

t=SDA;

b|=t<<7;

CLK=1;

CLK=0; }

return b/16\*10+b%16; }

uchar Read\_Data(uchar addr) //从DS1302指定位置读数据

{ uchar dat;

RST=0;

CLK=0;

RST=1;

DS1302\_Write\_Byte(addr);

dat=DS1302\_Read\_Byte();

CLK=1;

RST=0;

return dat;}

void Write\_DS1302(uchar addr,uchar dat) //向DS1302指定位置写入数据

{ CLK=0;

RST=1;

DS1302\_Write\_Byte(addr);

DS1302\_Write\_Byte(dat);

CLK=0;

RST=0; }

void SET\_DS1302() //设置时间

{ uchar i;

Write\_DS1302(0x8e,0x00);

for(i=0;i<7;i++)

{

Write\_DS1302(0x80+2\*i,(DateTime[i]/10<<4|(DateTime[i]%10)));

}

Write\_DS1302(0x8e,0x80);

}

void GetTime() //读取当前时期时间

{ uchar i;

for(i=0;i<7;i++)

{

DateTime[i]=Read\_Data(0x81+2\*i);

}

}

void Format\_DateTime(uchar d,uchar \*a) //时间和日期转换成数字字符

{

a[0]=d/10+'0';a[1]=d%10+'0';

}

uchar isLeapYear(uint y) //判断是否为闰年

{

return (y%4==0&&y%100!=0)||(y%400==0);

}

void RefreshWeekDay() //星期转换

{ uint i,d,w=5;

for(i=2000;i<2000+DateTime[6];i++)

{ d=isLeapYear(i) ? 366 : 365;

w=(w+d)%7; }

d=0;

for(i=1;i<DateTime[4];i++) d+=MonthsDays[i];

d+=DateTime[3];

DateTime[5]=(w+d)%7+1;

}

void DateTime\_Adjust(char x) //年月日时分秒++/--

{ switch (Adjust\_Index)

{ case 6: //年

if(x== 1&&DateTime[6]<99) DateTime[6]++;

if(x==-1&&DateTime[6]>0) DateTime[6]--;

MonthsDays[2]=isLeapYear(2000+DateTime[6])? 29:28;

if(DateTime[3]>MonthsDays[DateTime[4]])

DateTime[3]=MonthsDays[DateTime[4]];

RefreshWeekDay();

break;

case 4: //月

if(x== 1&&DateTime[4]<12) DateTime[4]++;

if(x==-1&&DateTime[4]>1) DateTime[4]--;

MonthsDays[2]=isLeapYear(2000+DateTime[6])? 29:28;

if(DateTime[3]>MonthsDays[DateTime[4]])

DateTime[3]=MonthsDays[DateTime[4]];

RefreshWeekDay();

break;

case 3: //日

MonthsDays[2]=isLeapYear(2000+DateTime[6])? 29:28;

if(x== 1&&DateTime[3]<MonthsDays[DateTime[4]]) DateTime[3]++;

if(x==-1&&DateTime[3]>0) DateTime[3]--;

RefreshWeekDay();

break;

case 2: //时

if(x== 1&&DateTime[2]<23) DateTime[2]++;

if(x==-1&&DateTime[2]>0) DateTime[2]--;

break;

case 1: //秒

if(x== 1&&DateTime[1]<59) DateTime[1]++;

if(x==-1&&DateTime[1]>0) DateTime[1]--;

break;}

}

void T0\_INT() interrupt 1 //定时器0中断，定时器每秒刷新LCD显示

{ TH0=-50000/256;

TL0=-50000%256;

if(++tCount!=2) return;

tCount=0;

Format\_DateTime(DateTime[6],LCD\_DSY\_BUFFER1+5);

Format\_DateTime(DateTime[4],LCD\_DSY\_BUFFER1+8);

Format\_DateTime(DateTime[3],LCD\_DSY\_BUFFER1+11);

strcpy(LCD\_DSY\_BUFFER1+13,WEEK[DateTime[5]-1]);

Format\_DateTime(DateTime[2],LCD\_DSY\_BUFFER2+5);

Format\_DateTime(DateTime[1],LCD\_DSY\_BUFFER2+8);

Format\_DateTime(DateTime[0],LCD\_DSY\_BUFFER2+11);

Display\_LCD\_String(0x00,LCD\_DSY\_BUFFER1);

Display\_LCD\_String(0x40,LCD\_DSY\_BUFFER2); }

void EX\_INT0() interrupt 0 //键盘中断

{ if(K1==0) //选择调整对象：年，月，日，时，分，秒

{ while(K1==0);

if(Adjust\_Index==-1||Adjust\_Index==1) Adjust\_Index=7;

Adjust\_Index--;

if(Adjust\_Index==5) Adjust\_Index=4;

LCD\_DSY\_BUFFER2[13]='[';

LCD\_DSY\_BUFFER2[14]=Change\_Flag[Adjust\_Index];

LCD\_DSY\_BUFFER2[15]=']'; }

else if(K2==0) // 加

{

while (K2==0); DateTime\_Adjust(1);

}

else if (K3==0) // 减

{

while (K3==0); DateTime\_Adjust(-1);

}

else if (K4==0) // 确定

{ while(K4==0);

SET\_DS1302(); //调整后的时间写入DS1302

LCD\_DSY\_BUFFER2[13]=' ';

LCD\_DSY\_BUFFER2[14]=' ';

LCD\_DSY\_BUFFER2[15]=' ';

Adjust\_Index=-1; }

}

void main() //主程序

{ Init\_LCD(); //LCD初始化

IE=0x83; // 允许INT0，T0中断

IP=0x01;

IT0=0x01;

TMOD=0x01;//设置定时器0为工作方式1（M1M0为01）

TH0=-50000/256; //装初值

TL0=-50000%256;

TR0=1;

while(1)

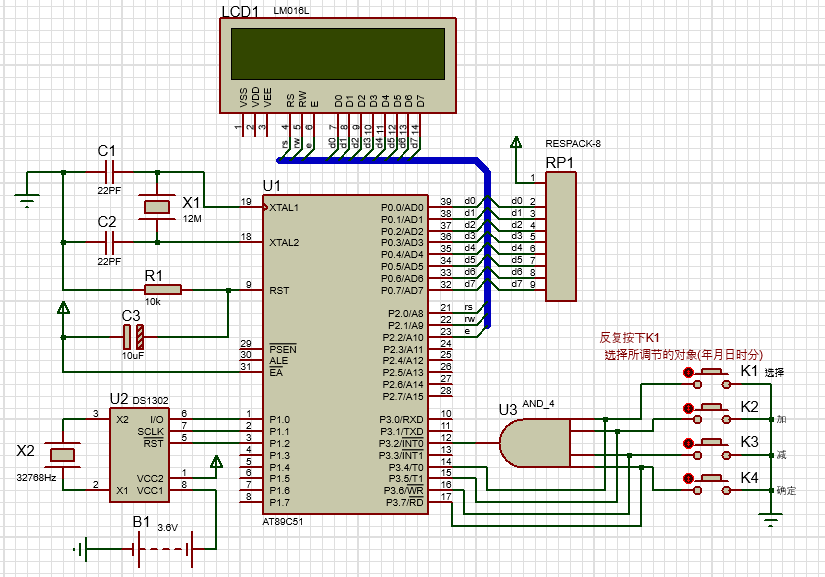
{

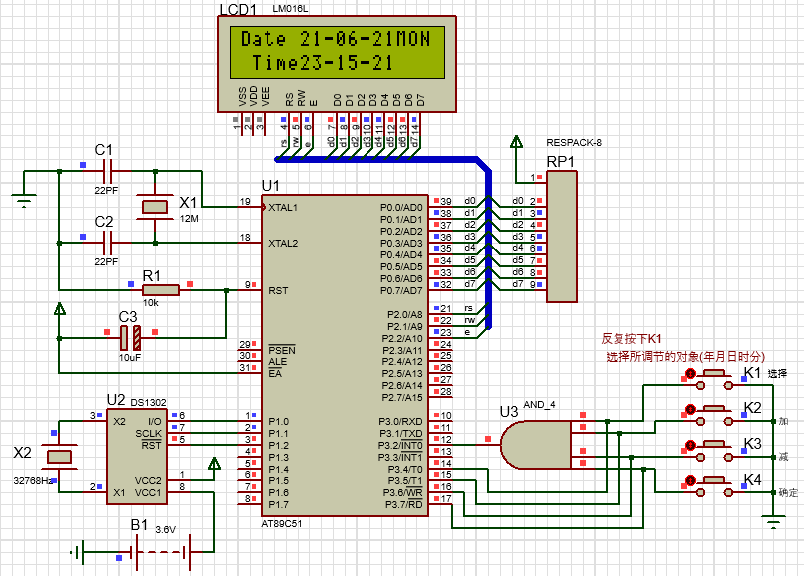
if(Adjust\_Index==-1) GetTime();

}

}

**附录2：设计图纸**





**附录3：元器件目录表**

表1.单片机最小系统元件清单

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 型 号（名称） | 数量 | 作用 | 标注 |
| 1 | STC89C52 | 1 | 单片机 | U1 |
| 2 | 40脚锁座 | 1 | 单片机座子 |  |
| 3 | 22pF瓷片电容 | 2 | 晶振电路用 | C2,C1 |
| 4 | 12MHZ晶振 | 1 | 晶振 | X1 |
| 5 | 10uF电解电容 | 1 | 上电复位 | C3 |
| 6 | 10K电阻 | 1 | 复位电路 | R1 |
| 7 | 4脚按键 | 1 | 按下，单片机实现复位功能 | K5 |

表2.DS1302时钟电路元件清单

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 型 号（名称） | 数量 | 作用 | 标注 |
| 1 | DS1302时钟芯片 | 1 | 时钟计时 | U2 |
| 2 | 32.768kHz晶振 | 1 | 晶振 | X2 |
| 3 | 3.6V电池模块 | 1 | 后备电源 | B1 |

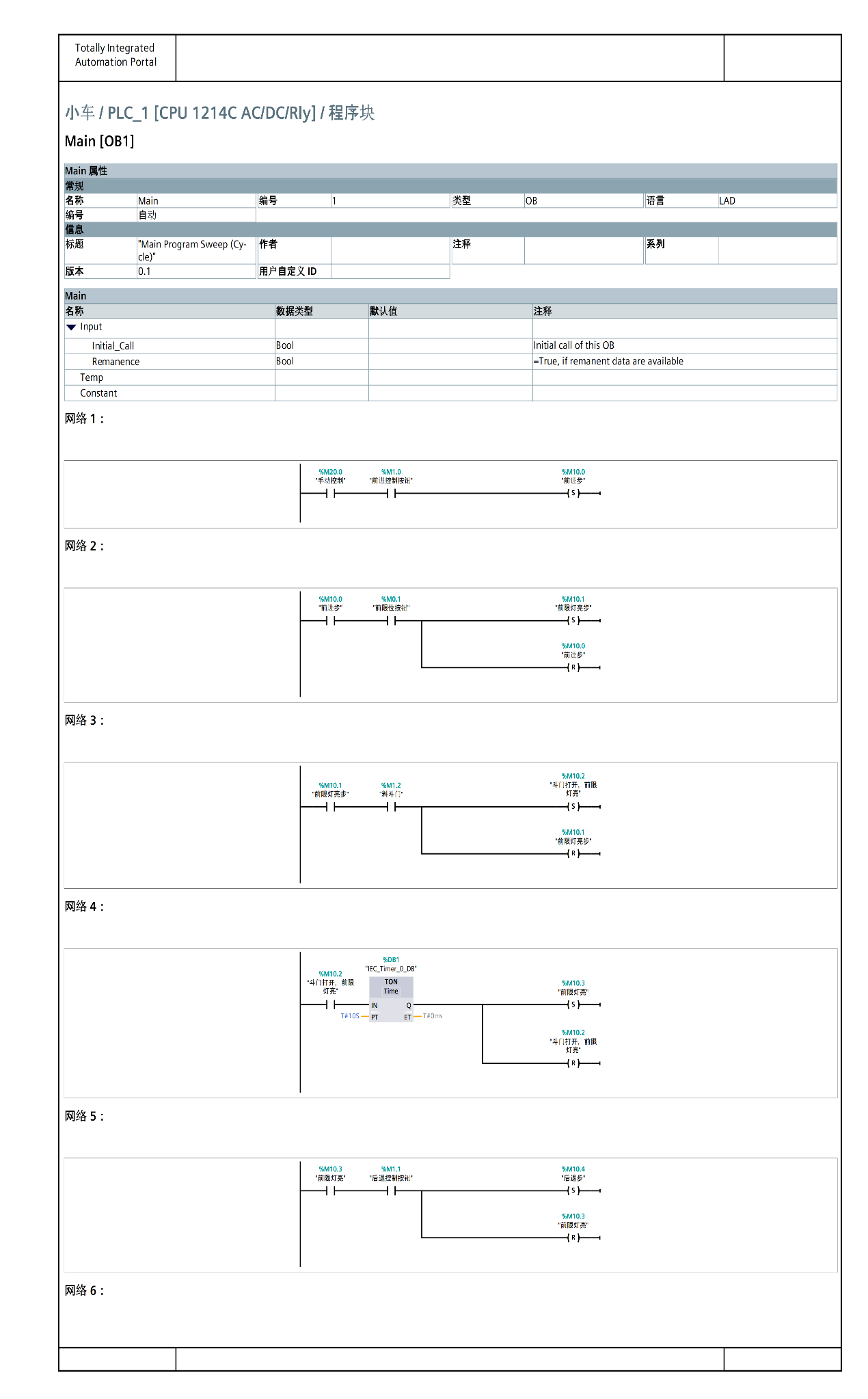
表3.LCD1602液晶显示元件清单

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 型 号（名称） | 数量 | 作用 | 标注 |
| 1 | LCD1602 | 1 | 显示日历 | LM016L |
| 2 | 16脚单排母座 | 1 | 显示屏底座 |  |
| 3 | 9脚排阻 | 1 | 保护电路 | RP1 |

表4.独立按键元件清单

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 型 号（名称） | 数量 | 作用 | 标注 |
| 1 | 74LS21P | 1 | 4输入双与门 | U3 |
| 2 | 4脚按键 | 4 | 实现选择、加、减、确定功能 | K1、K2、K3、K4 |

**附录1：装卸料小车程序清单**



图示

描述已自动生成

箱线图

描述已自动生成

表格

低可信度描述已自动生成

图形用户界面

低可信度描述已自动生成

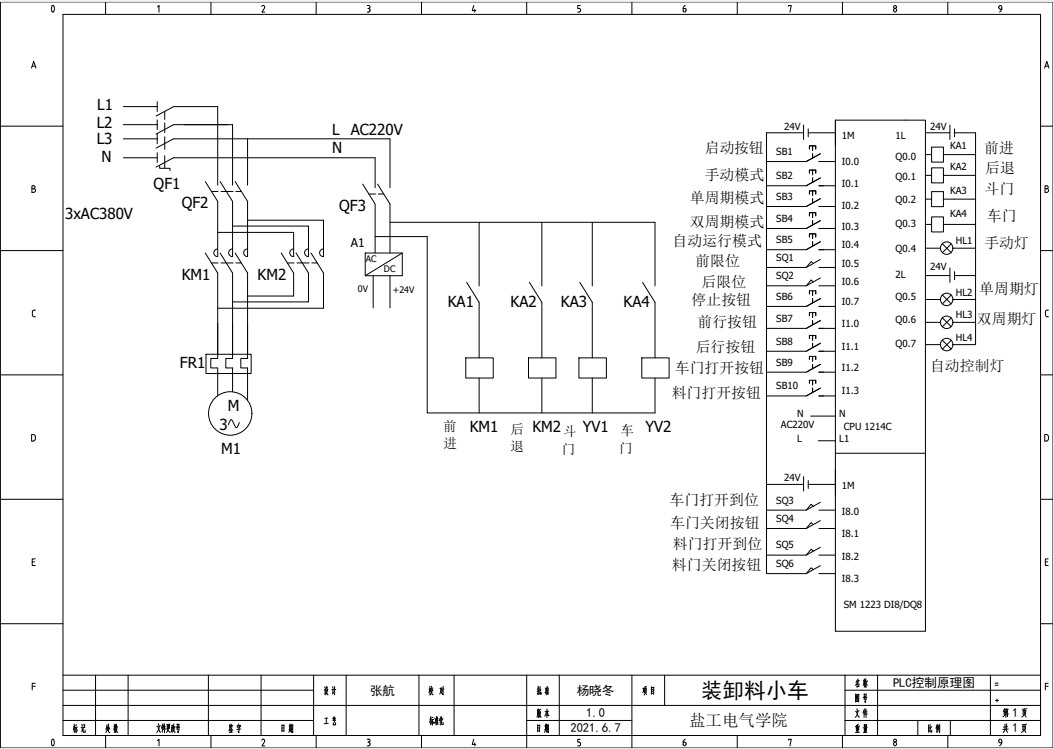
图示, 箱线图

描述已自动生成

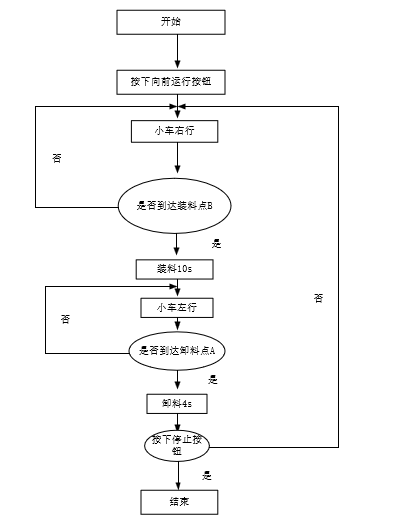
图表

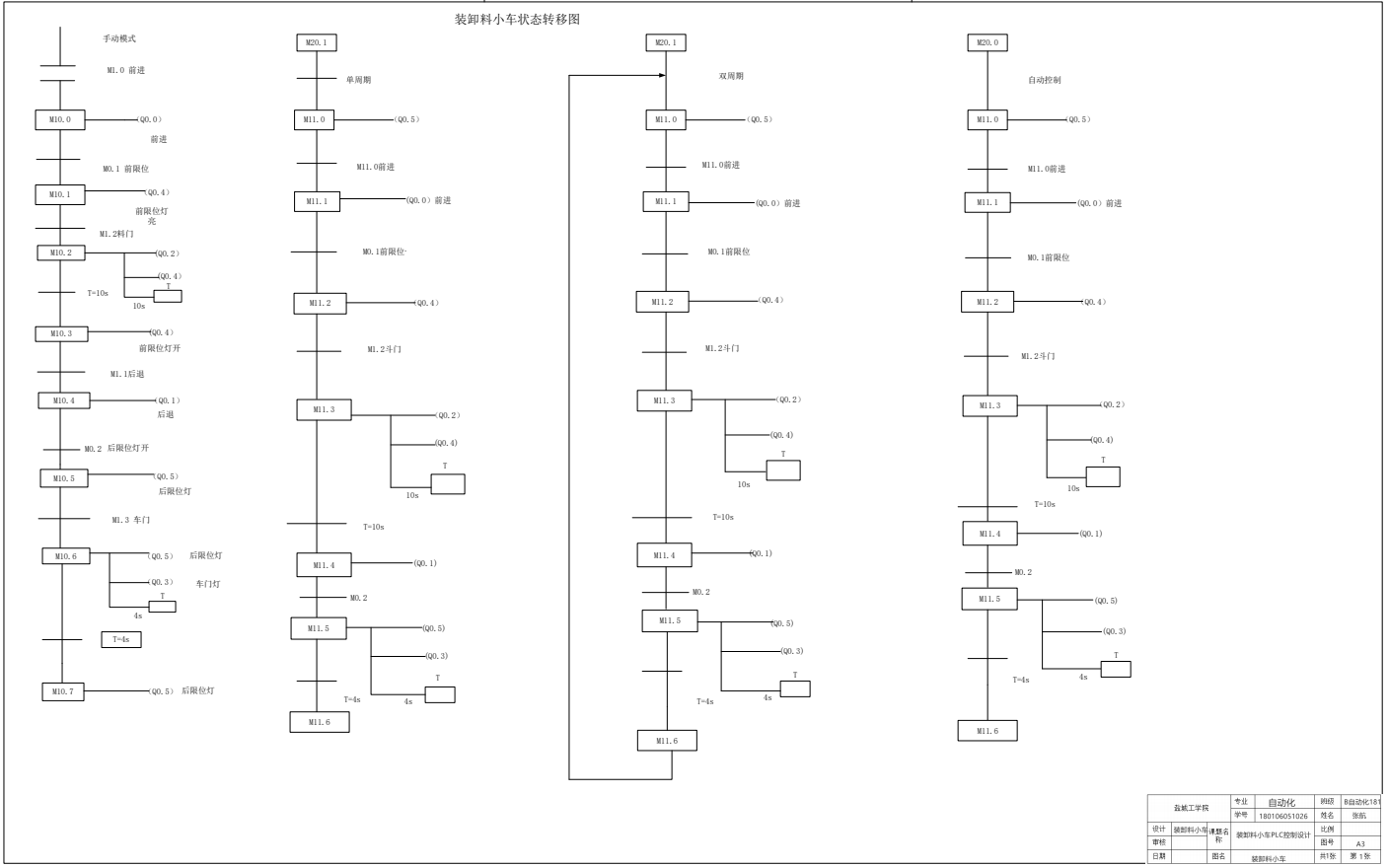
中度可信度描述已自动生成

**附录2.1：PLC原理图**



**附录2.2：装卸料小车流程图**



**附录2.3：装卸料小车状态转移图**

**附录3 元器件目录表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PLC部分 | | | | |
| 名称 | 型号 | 单位 | 数量 | 备注 |
| 电源模块 | L61P-CM | 块 | 1 | AC 100-240 V→DC 5 V 5A |
| CPU模块 | 1214CAC/DC/RLY CPU-S7-1200 | 块 | 1 | 内置以太网接口、14入10出I/O接口 |
| 低压断路器 | DZ47-3P-16A | 个 | 1 | 灭弧方式：磁吹断路器 |
| 施耐德接触器 | CJX2 | 个 | 1 | 额定绝缘电压690V；额定工作电压380V；额定工作电流至20A |
| 熔断器 | RC1A-10 | 个 | 1 | 额定电压：380V；额定电流:10A;工作频率：50Hz |
| 热继电器 | JY-1 | 个 | 1 | 频率50Hz/60Hz；额定电压：380V；额定电流：2A |
| 三相异步电动机 | YR355M1-4 | 台 | 1 | 电机型号：JYZ-12-4；额定频率50Hz;额定功率0.81KW；额定电流2A；转速1410r/min |
| 按钮 | TS1271 | 个 | 5 | 尺寸：12mm\*12mm;圆按钮：带灯轻触开关 |
| 限位开关 | LX32 | 个 | 2 | 工作频率：50Hz;额定电流：5A;额定电压：380V |
| 通电延时继电器 | ST3P-A | 个 | 2 | 额定电压：220V;额定电流：5A；频率：60Hz |
| 电磁阀 | 4V310-10 | 个 | 2 | 额定电压：220V；频率50/60Hz |
| 选择开关 | LW6-2 | 个 | 1 | 频率：50Hz 额定电压：380V；额定电流：5A |