

合肥工业大学

Hefei University of Technology

电子信息科学与技术 综合设计报告

设计题目： 洗衣机控制电路设计

学生姓名： 钟尤

学 号： 2017218272

班 级： 电信科 17-2 班

组员姓名： 雷炜轩

组员学号： 2017218522

组员班级： 电信科 17-2 班

指导教师： 李春华、殷文斐、姜烨

2020 年 10 月 15 日

摘 要

本次综合设计我们组的题目是洗衣机控制电路设计，题目要求为：实现洗衣机各种不同（正常洗，精洗，脱水，桶自洁等）功能设计。拿到题目后，我们首先分析了以下两个问题：一是通过什么方式来实现这些功能，二是要实现这些功能需要用到哪些模块。最终，我们实现了使用电机模拟洗衣机滚筒的转动，通过计时器控制时间，我们对洗衣机的四种工作模式分别设计了不同的工作时间和不同的电机转速，以此模拟真实的洗衣机情况。并且在 LCD1602 上显示当前洗衣机的工作状态和洗衣机工作剩余时间，能够通过按键设置水位，在模拟洗衣结束后，通过指示灯闪烁和蜂鸣器鸣叫来提示用户洗衣结束。

在完成了以上题目要求的基本功能后，我们增加了刷卡开机的功能：使用 MFRC522 模块，录入我们的学生卡的信息，只有刷卡后才能进入洗衣机功能选择界面。

在本次综合设计过程中，我主要负责硬件及其调试的工作。在明确了我们需要使用的其中一个主要模块是步进电机后，我搜集了关于步进电机的原理图资料、接线方式和使用方法，经过反复调试，最终使电机能够实现定时转动和正传反转交替。

软件方面采用 C 语言编程。通过对单片机程序设计实现对洗衣机系统的整个洗涤程序进行监测、判断和控制以及人机交换。

通过这次设计让我更深入了解单片机基本电路、如何设计控制电路和调试的基本方法，从而锻炼了我学习、设计和开发软、硬件的能力。

关键词：洗衣机；单片机； MFRC522

目 录

1 绪论	1
1.1 课题背景.....	1
1.2 设计任务与要求.....	3
1.3 方案设计思路.....	3
2 系统软硬件设计	3
2.1 系统硬件设计.....	3
2.1.1 硬件总体设计.....	3
2.1.2 五线四象步进电机.....	4
2.1.3 LCD1602 显示屏.....	5
2.1.4 独立按键模块.....	6
2.1.5 继电器模块.....	7
2.1.6 蜂鸣器模块.....	8
2.1.7 MFRC522 模块.....	9
2.2 系统软件设计.....	10
2.3 系统调试.....	11
2.3.1 硬件调试.....	11
2.3.2 软件调试.....	12
2.3.2 调试图片.....	13
 参考文献.....	 14
附录.....	15

插图清单

图 2.1 硬件总体设计电路图.....	4
图 2.2 步进电机原理图.....	4
图 2.3 步进电机时序图.....	5
图 2.4 LCD1602 原理图.....	6
图 2.5 独立按键原理图.....	7
图 2.6 继电器原理图.....	8
图 2.7 蜂鸣器原理图.....	9
图 2.8 MFRC522 原理图.....	10
图 2.9 接通电源前演示图.....	14
图 2.10 显示学号姓名演示图.....	15
图 2.11 模式选择演示图.....	15
图 2.12 设置水位演示图.....	16
图 2.13 电机转动演示图.....	16
图 2.14 洗衣结束提示演示图.....	17

1 绪论

1.1 课题背景

从古到今，洗衣服也都是一项繁重的家务劳动，需要手搓、棒击、冲刷、甩打等。而这些不断重复的简单的体力劳动，留给人的感受常常就是：辛苦劳累。而且在洗衣机出现以前，对于大部分的人毫无乐趣可言。

到了 1874 年，有人发明了用木制手摇洗衣机。它的发明者是美国人比尔·布莱克斯。布莱克斯所发明的洗衣机只是在木筒里装上 6 块叶片，使用用手柄和齿轮传动，构造却极为简单，通过叶片来使衣服在筒内翻转，从而达到是衣物洁净的目的。而这套装置的问世，可以使那些为了冥思苦想提高生活效率的人们很受启发，这开始大大加快了洗衣机的改进过程。

到了 1880 年，美国又发明了蒸气洗衣机，开始使用蒸气动力取代人力。之后，随着水力洗衣机以及内燃机洗衣机的相继出现。到了 1911 年，世界上第一台的电动洗衣机在美国试制成功。而电动洗衣机的问世，则标志着人类家务劳动的自动化开端。

电动洗衣机经过不断的改进与完善，在 1922 年美国玛依塔格公司所研制成功一种新的洗衣方式称之为“搅拌式”的机器搅拌式洗衣机。这种洗衣机的构造是在筒的中心位置装上一个立轴，而立轴的下端有一个装置叫搅拌翼，并通过电动机来带动立轴工作，进行周期性的正反方向的摆动，通过这种方式来驱使衣物和水流之间进行不断翻滚和相互摩擦，以此来去除衣物上的污垢。因搅拌式洗衣机从结构和洗衣功能相对来说更加科学合理和具有更好的洗涤效果，从而受到人们的普遍欢迎。但是不过很快不到 10 年的工夫，美国的本德克斯航空公司又宣布，他们公司所研制成功了第一台的前装式滚筒洗衣机，这种机器可以实现洗涤和漂洗以及脱水功能在同一个滚筒内完成。这就意味着电动洗衣机在型式上又前进了一步，在自动化方向又跃进了一个台阶！直到今天，滚筒式洗衣机仍然在很多地区和国家得到广泛的应用。

随着工业化的加速发展，在世界上各个国家都加快了洗衣机研制的进程。起初英国研制并推出了一种叫做为喷流式洗衣机，它就是靠筒体一侧来运转波轮从而产生强烈涡流，来促使洗涤液和衣物共同在滚筒之内不断翻滚旋转，达到洗净衣物的

效果。到了 1955 年，另具风格、并且至今都在流行的波轮式洗衣机在日本研制出来。此时，搅拌式和滚筒式以及波轮式这三种类型的洗衣机共同占有了洗衣机市场的大部分江山。

到了 20 世纪 60 年代之后，许多发达国家在洗衣机的消费市场开始形成一系列的产品，洗衣机的家庭普及率急速上升。在此期间日本在洗衣机的发展领域备受瞩目。60 年代在日本出现了能带干桶的双桶洗衣机，也就是人们所谓的“半自动型洗衣机”。到了 70 年代，日本又生产出了波轮式套桶全自动洗衣机。而到 70 年代后期随着计算机技术的发展，通过单片机控制(即微电脑控制)的全自动洗衣机横空出世。而到 80 年代，通过所谓的“模糊控制，使洗衣机功能更加完备，操作更加简便，外观造型更加为时尚，洗衣程序更加随人意。进入了 90 年代以后，由于电机调速技术水平的提高，宽范围的转速变换以及调节在洗衣机上的实现，从而诞生了许多的新水流洗衣机。在此以后，在电机驱动技术的发展和提高的基础上，日本又开发和生产出了电机直接驱动式洗衣机，这中洗衣机直接省去了齿轮传动和变速机构，这是洗衣机驱动方式上的一场革命。

但是在中国，由于历史的原因，洗衣机工业的起步非常的晚，一直到 1978 年才开始正式生产家用洗衣机。但是，随着改革开放的不断深化，随着人们认识水平的提高以及经济实力和收入的增长，进入 80 年代后中国的洗衣机工业持续保持着旺盛的发展势头。从 1978 年到 1983 年洗衣机的产量由 400 台急剧飘升到了 365 万台，在此以后大规模的技术引进的热潮在全国各地掀起，大约有数十个厂家不分先后从先进国日本、英国、法国、意大利、澳大利亚等国把洗衣机先进技术引进了数十项。相关技术的大规模引进和吸收甚至创新，从而极大的提升了我国的国产洗衣机的生产能力以及产业素，从而大大缩短了同发达国家之间的技术差距。到了今天，我国洗衣机的年产销量大概占据了世界年产量的四分之一，从而跃居世界首位。除在品种和数量上满足了国内市场之外，又出口到东南亚、欧洲、北美等地区，从而位居国际洗衣机市场的行列。

伴随着科学技术的进一步发展，相信新的更适合人们使用的洗衣机能够给我们的生活带来新的方式。

1.2 设计任务与要求

实现洗衣机各种不同（正常洗，精洗，脱水，桶自洁等）功能设计。

1.3 方案设计思路

目前国内和国外市场上有很多类型的洗衣机，可以采用不同的控制系统，一般都是以单片机以及 PLC 可编程控制器为主。基于实际情况本次设计主要选用 89C51 单片机的各部件的控制要求。以单片机为主体的控制芯片，加上由振荡时钟、输入开关、按键、1602 液晶屏显示器、蜂鸣报警器以及双向可控硅驱动电路和欠压检测保护电路等组成的系统，来形成一个完整的单片机控制系统。对于故障的排除，在检测过机械部件以后，一般来说先检测外围的有关控制电路，等确定正常后，再判断单片机的故障，控制系统接受有关的动作指令，然后发出相应的执行指令，是电机，进水阀，排水阀按照程序的要求通电运行，同时要对洗衣机的工作状态进行检测和显示，以确定工作是否正常，一旦出现异常，则立刻发出命令，并进入报警故障状态。通过系统的各部分功能协调控制洗衣机的工作。

2.1 系统硬件设计

该电路图的核心部件是由单片机控制芯片，再加上由振荡时钟电路、输入开关、按键、8 位 LED 数码管显示器、蜂鸣报警器和双向可控硅驱动电路以及欠压检测保护电路等组成的系统，来形成一个完整的单片机控制系统。通过对单片机编程以实现对洗衣机整个洗涤程序和过程进行监测、判断、控制以及人机交互。

图 2.1 硬件总体设计电路图

该步进电机为一四相步进电机，采用单极性直流电源供电。只要对步进电机的各相绕组按合适的时序通电，就能使步进电机步进转动。图 2.2 是该四相反应式步进

电机工作原理示意图。

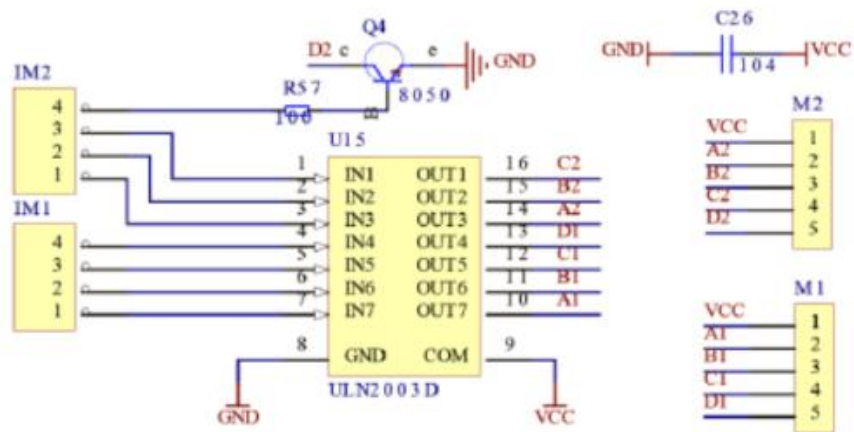


图 2.2 步进电机原理图

开始时，开关 SB 接通电源，SA、SC、SD 断开，B 相磁极和转子 0、3 号齿对齐，同时，转子的 1、4 号齿就和 C、D 相绕组磁极产生错齿，2、5 号齿就和 D、A 相绕组磁极产生错齿。

当开关 SC 接通电源，SB、SA、SD 断开时，由于 C 相绕组的磁力线和 1、4 号齿之间磁力线的作用，使转子转动，1、4 号齿和 C 相绕组的磁极对齐。而 0、3 号齿和 A、B 相绕组产生错齿，2、5 号齿就和 A、D 相绕组磁极产生错齿。依次类推，A、B、C、D 四相绕组轮流供电，则转子会沿着 A、B、C、D 方向转动。

四相步进电机按照通电顺序的不同，可分为单四拍、双四拍、八拍三种工作方式。单四拍与双四拍的步距角相等，但单四拍的转动力矩小。八拍工作方式的步距角是单四拍与双四拍的一半，因此，八拍工作方式既可以保持较高的转动力矩又可以提高控制精度。

单四拍、双四拍与八拍工作方式的电源通电时序与波形分别如图 2.3 所示。

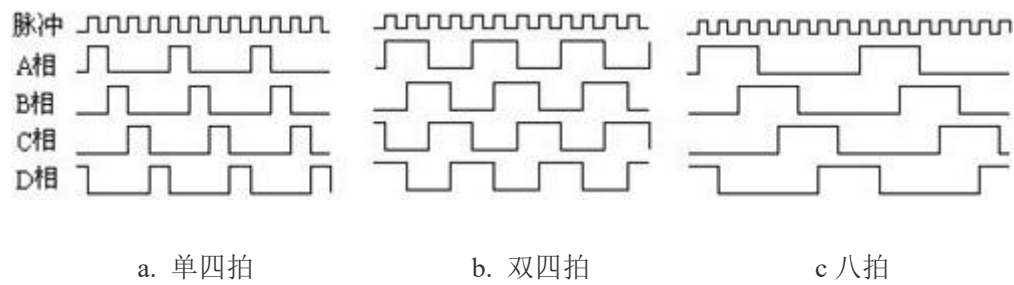


图 2.3 步进电机时序图

2.1.3 LCD1602 显示屏

LCD 技术是把液晶灌入两个列有细槽的平面之间。这两个平面上的槽互相垂直(相交成 90 度)。也就是说,若一个平面上的分子南北向排列,则另一平面上的分子东西向排列,而位于两个平面之间的分子被强迫进入一种 90 度扭转的状态。由于光线顺着分子的排列方向传播,所以光线经过液晶时也被扭转 90 度。但当液晶上加一个电压时,分子便会重新垂直排列,使光线能直射出去,而不发生任何扭转。

LCD 是依赖极化滤光器(片)和光线本身。自然光线是朝四面八方随机发散的。极化滤光器实际是一系列越来越细的平行线。这些线形成一张网,阻断不与这些线平行的所有光线。极化滤光器的线正好与第一个垂直,所以能完全阻断那些已经极化的光线。只有两个滤光器的线完全平行,或者光线本身已扭转至与第二个极化滤光器相匹配,光线才得以穿透。

lcd1602 主要用来显示数字、字母、图形以及少量自定义字符。由于其显示控制简单,性价比高,广泛用于电子表、冰箱、空调、汽车电子仪表等装置。lcd1602 分为带背光和不带背光两种,基控制器大部分为 HD44780,带背光的比不带背光的厚,是否带背光在设计过程中并无差别。

如图 2.4 所示,开机后首先显示我们的学号和姓名,在控制过程中,显示洗衣机当前状态。

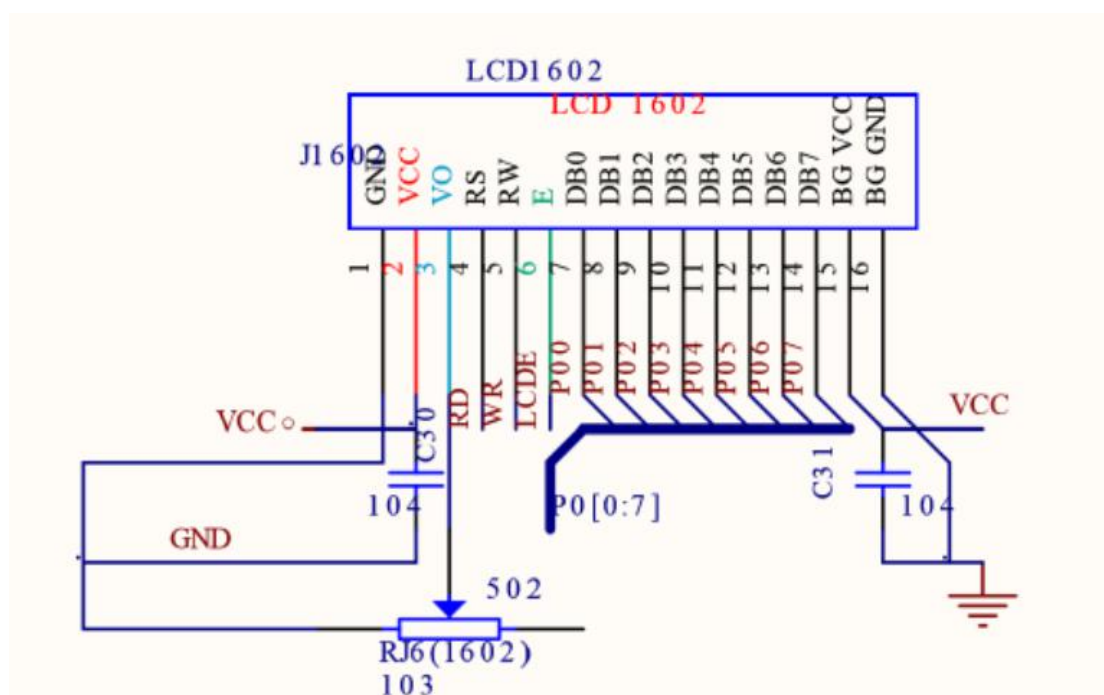


图 2.4 LCD1602 原理图

2.1.4 独立按键

独立按键式直接用 I/O 口线构成的单个按键电路，其特点式每个按键单独占用一根 I/O 口线，每个按键的工作不会影响其他 I/O 口线的状态。独立按键的典型应用如图所示。独立式按键电路配置灵活，软件结构简单，但每个按键必须占用一个 I/O 口线，因此，在按键较多时，I/O 口线浪费较大，不宜采用。独立按键如图 2.5 所示。

独立按键的软件常采用查询式结构。先逐位查询没跟 I/O 口线的输入状态，如某一根 I/O 口线输入为低电平，则可确认该 I/O 口线所对应的按键已按下，然后，再转向该键的功能处理程序。

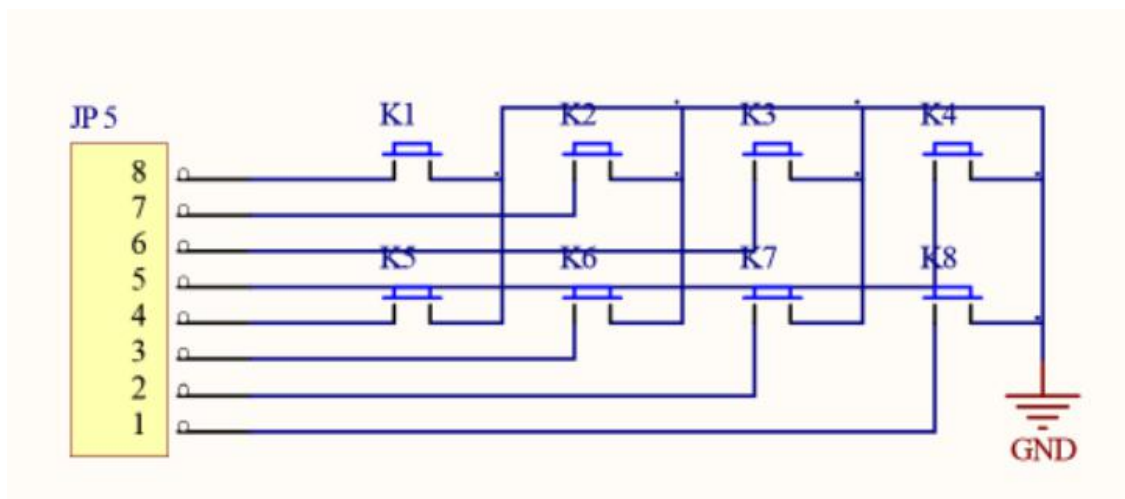


图 2.5 独立按键原理图

2.1.5 继电器模块

继电器是一种电子控制器件，它具有控制系统（又称输入回路）和被控制系统（又称输出回路），通常应用于自动控制电路中，它实际上是用较小的电流去控制较大电流的一种“自动开关”。故在电路中起着自动调节、安全保护、转换电路等作用。

继电器一般都有能反映一定输入变量（如电流、电压、功率、阻抗、频率、温度、压力、速度、光等）的感应机构（输入部分）；有能对被控电路实现“通”、“断”控制的执行机构（输出部分）；在继电器的输入部分和输出部分之间，还有对输入量进行耦合隔离，功能处理和对输出部分进行驱动的中间机构（驱动部分）。

作为控制元件，概括起来，继电器有如下几种作用。

- (1)扩大控制范围：例如，多触点继电器控制信号达到某一定值时，可以按触点组的不同形式，同时换接、开断、接通多路电路。
- (2)放大：例如，灵敏型继电器、中间继电器等，用一个很微小的控制量，可以控制很大功率的电路。
- (3)综合信号：例如，当多个控制信号按规定的形式输入多绕组继电器时，经过比较综合，达到预定的控制效果。
- (4)自动、遥控、监测：例如，自动装置上的继电器与其他电器一起，可以组成程序控制线路，从而实现自动化运行。

继电器一般都有能反映一定输入变量（如电流、电压、功率、阻抗、频率、温度、压力、速度、光等）的感应机构（输入部分）；有能对被控电路实现“通”、“断”控制的执行机构（输出部分）；在继电器的输入部分和输出部分之间，还有对输入量进行耦合隔离，功能处理和对输出部分进行驱动的中间机构（驱动部分）。如图 2.6 所示。

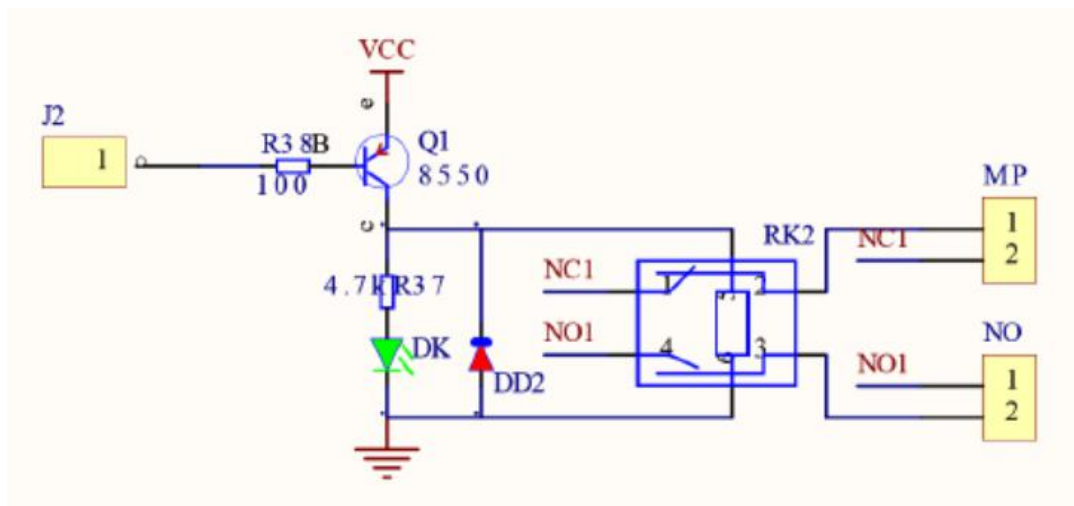


图 2.6 继电器原理图

2.1.6 蜂鸣器模块

使用 I/O 口定时翻转电平驱动蜂鸣器方式的设置比较简单，只需要对波形分析一下。由于驱动的信号刚好为周期 $500\mu\text{s}$ ，占空比为 $1/2\text{duty}$ 的方波，只需要每 $250\mu\text{s}$ 进行一次电平翻转，就可以得到驱动蜂鸣器的方波信号。在程序上，可以使用 TIMER0 来定时，将 TIMER0 的预分频设置为/1，选择 TIMER0 的始终为系统时钟(主振荡器时钟/4)，在 TIMER0 的载入/计数寄存器的高 4 位和低 4 位分别写入 00H 和 06H，就能将 TIMER0 的中断设置为 $250\mu\text{s}$ 。当需要 I/O 口驱动的蜂鸣器鸣叫时，只需要在进入 TIMER0 中断的时候对该 I/O 口的电平进行翻转一次，直到蜂鸣器不需要鸣叫的时候，将 I/O 口的电平设置为低电平即可。不鸣叫时将 I/O 口的输出电平设置为低电平是为了防止漏电。

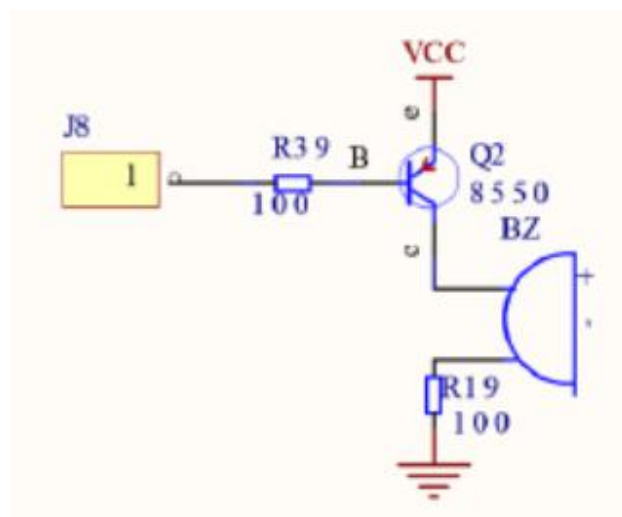


图 2.7 蜂鸣器原理图

2.1.7 MFRC522 模块

MF RC522 是应用于 13.56MHz 非接触式通信中高集成度读写卡系列芯片中的一员。是 NXP 公司针对“三表”应用推出的一款低电压、低成本、体积小的非接触式读写卡芯片，是智能仪表和便携式手持设备研发的较好选择。

MF RC522 利用了先进的调制和解调概念，完全集成了在 13.56MHz 下所有类型的被动非接触式通信方式和协议。支持 ISO14443A 的多层应用。其内部发送器部分可驱动读写器天线与 ISO 14443A/MIFARE 卡和应答机的通信，无需其它的电路。接收器部分提供一个坚固而有效的解调和解码电路，用于处理 ISO14443A 兼容的应答器信号。数字部分处理 ISO14443A 帧和错误检测（奇偶 & CRC）。此外，它还支持快速 CRYPTO1 加密算法，用于验证 MIFARE 系列产品。MFRC522 支持 MIFARE 更高速的非接触式通信，双向数据传输速率高达 424kbit/s。

作为 13.56MHz 高集成度读写卡系列芯片家族的新成员，MF RC522 与 MF RC500 和 MF RC530 有不少相似之处，同时也具备诸多特点和差异。它与主机间的通信采用连线较少的串行通信，且可根据不同的用户需求，选取 SPI、I2C 或串行 UART(类似 RS232)模式之一，有利于减少连线，缩小 PCB 板体积，降低成本。

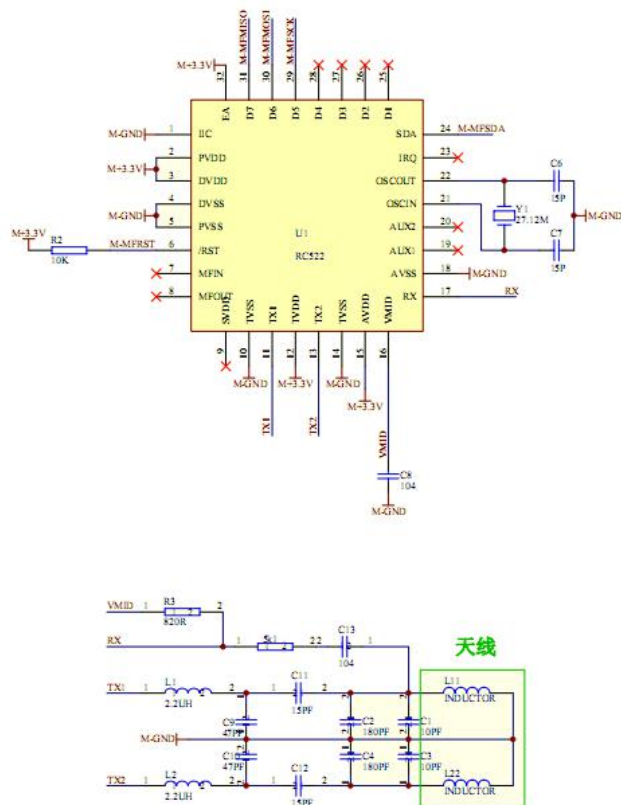


图 2.8 MFRC522 原理图

2.2 系统软件设计

单片机的应用程序，也是根据系统所需要实现功能要求而加以设计的，所以要求能够可靠地达到单片机系统的各项功能要求。应用系统的种类非常多且功能各异，但是一个优秀的应用系统程序应该具备以下特性。

- (1)程序的结构简洁、清晰、流程合理。
- (2)使用模块化程序设计，从而方便用户的编译、调试等操作。
- (3)数据存储区、程序存储区的使用要求合适，从而实现硬件资源的节约这一目的。
- (4)对系统运行状态的管理要标志化，每一个功能程序运行要求、运行状态以及运行结果都必须对状态位进行设置以方便查询。程序的运行、转移、控制都可以根据状态标志来确定。
- (5)所调试以及修改之后的程序能够实现规范化的要求。好处在于便于借鉴以及

交流。

- (6)能够达到软件的抗干扰设计。为是系统能可靠的工作，应该在程序中设置自我诊断功能，每次运行前先执行自诊断程序，用来诊断各个系统的状态参数以及特性特征有没有发生异常现象。

2.3 系统调试

2.3.1 硬件调试

单片机应用系统就是指以单片机为核心部分，加上外设及功能部件以及程序组成的系统，该系统能够实现某项功能。硬件是组成系统的基础，而软件是工作在硬件的基础上通过软件对外设及功能部件进行控制达到系统的功能要求。单片机应用系统的设计的主要过程具体可以分为总体设计、硬件设计、软件设计和可靠性设计(包括软件和硬件方面)以及软硬件调试等几个过程。

(1)总体设计

首先必须要确定本次所设计系统的需要达到的功能要求(即设计者的思路)、相关技术要求等，并且需要进行相应的可行性分析，明确该系统的设计任务。根据该系统任务的要求，在达到技术要求的基础上，再提出多种方案进行选择，对于方案进行详细可靠的分析，并选择其中较为合理的一种方案来架设系统的总体框架。

(2)硬件设计

硬件设计一般包括芯片的选择、主机选择、输入/输出口分配及选择、电源的设计以及抗干扰措施等主要内容。

(3)软件设计

主要是要能够完成各项系统的功能，比如显示、计算和测量等；

(4)系统调试

单片机应用系统的硬件调试一般来说可以分为联机和脱机调试两种方式。

● 脱机调试

在通电之前，先用根据硬件的原理图和装配图，利用万用表等工具，详细的检查系统线路是否正确接线，并且仔细核对元器件的规格、型号以及安装的要求。同时需要特别注意电源的布线，防止电源之间的极性发生错误以及短路，并需要重点检查系统的总线之间是不是存在着短路现象，

或者其他信号线之间短路。

对于排除逻辑上的故障，必须进行单独的调试。在调试好之后，详细检查负载能力、电压值等，如果都符合要求之后才能和系统的各个功能部件相连接。先不接入芯片，通上电，然后对各个插件之上引脚的点位进行详细的检查，并对各点电位情况进行仔细的测量。特别是应该注意单片机芯片的插脚之上的点位是不是发生了异常，如果有电压较高，那么在联机的时候将会使系统损毁。

● 联机调试

一些相对较为明显的硬件故障可以通过脱机调试进行排除。但是也有些故障必须要通过联机调试才能被发现和排除。在联机之前必须首先断开电源，将单片机的仿真头与单片机相连接，再检查仿真器与系统样机之间的电源以及接地是否的情况是否正常。没有问题之后才可以接通电源。

通电之后执行程序，对系统的存储器以及 I/O 端口进行详细的读写操作和逻辑检查。如果不正常，那么可以使用示波器对波形进行观察。通过对波形的观察分析，寻找是什么原因产生的故障，并对故障逐一排除。造成故障的主要原因，有开路、短路、线路连接上的逻辑错误和集成电路失效等。

当系统调试好之后，可以接上其他的外围部件。如显示器、键盘、输出驱动板、D/A, A/D 转换板等，然后再对系统进行进一步调试。

如果在调试的过程当中发现调试的系统工作正常，那么造成这种情况的原因可能有以下几种：系统的负载过大，电源的供电电压不够，电源滤波不完善，联机时线路接触不良等。

2.3.2 软件调试

软件调试和所选用的软件结构以及程序设计水平相关。假如采用模块化程序设计方法，那么需要分别对每个模块子程序分别进行设计和测试。调试子程序时要注意的事项是与实际想结合，对程序的入口和出口条件进行设计。调试的方法一般是采用单步或者对 PC 地址进行设置执行到某一位置，通过对现场、寄存器 RAM 的内容和 I/O 口的数据及状态，来确定程序执行结果是否与预先所设计的要求相同。通过检测与调试可以发现程序中的各种

错误，并且也可以发现系统中的硬件故障、和硬件设计方面的错误。在调试的过程中，需要不断的对软件和硬件进行调整与改进，逐个对每个程序模块进行模拟与调试。

在各个模块调试完成后，需要将所有的功能块联合起来进行总的调试。如果发生故障，那么可能的原因有:用来监控程序的寄存器与缓冲区发生冲突、子程序在运行对现场造成了破坏等。

单步和程序的分段调试以后，这时候应当再进行连线调试。主要原因在于单步运行只能判断程序是否正确，并不能解决时间的精度、CPU 的响应速度等问题。等到所有的调试都完成以后，这是可以通过反复的运行，来观察程序的执行情况，比如稳定性和操作是与原设计要求是不是相符合、操作是不是合乎要求等情况，在必要时需要作一定的改正。

在软件和硬件联和调试完成之后，经过反复的运行之后反应正常，可以将程序固化到 ROM 中。插入单片机插座中之后，本次调试系统工作也就完成。

2.3.3 调试图片

1.接通电源前:

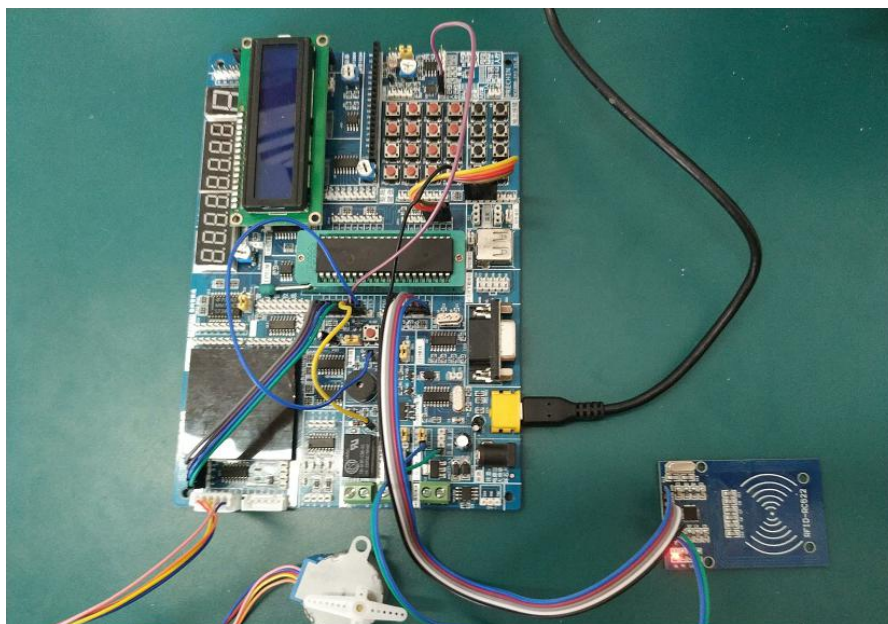


图 2.9 接通电源前演示图

2.刷卡后显示学号姓名:

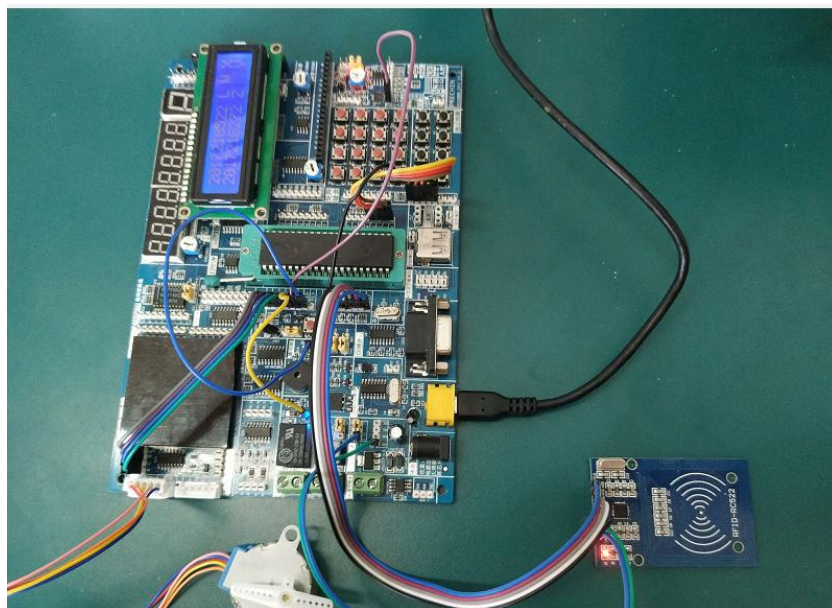


图 2.10 显示学号姓名演示图

3.进入模式选择界面:

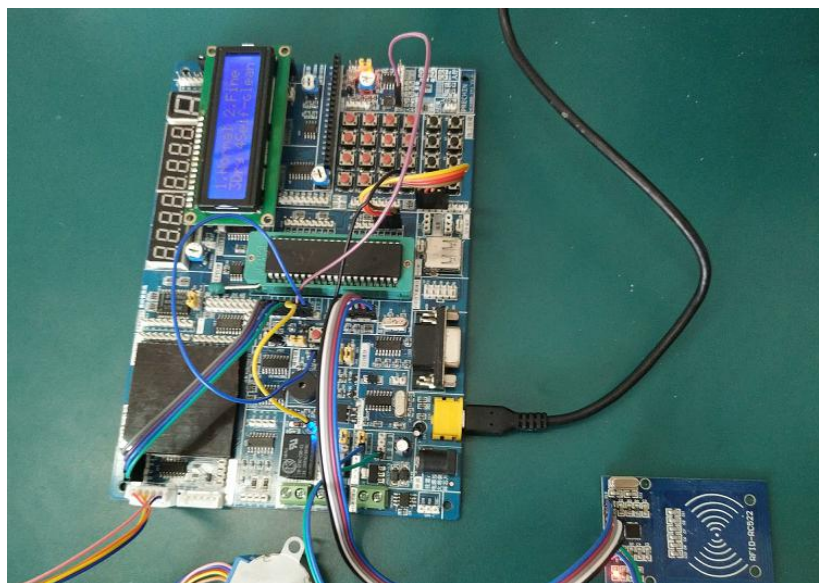


图 2.11 模式选择演示图

4.设置水位及时间:

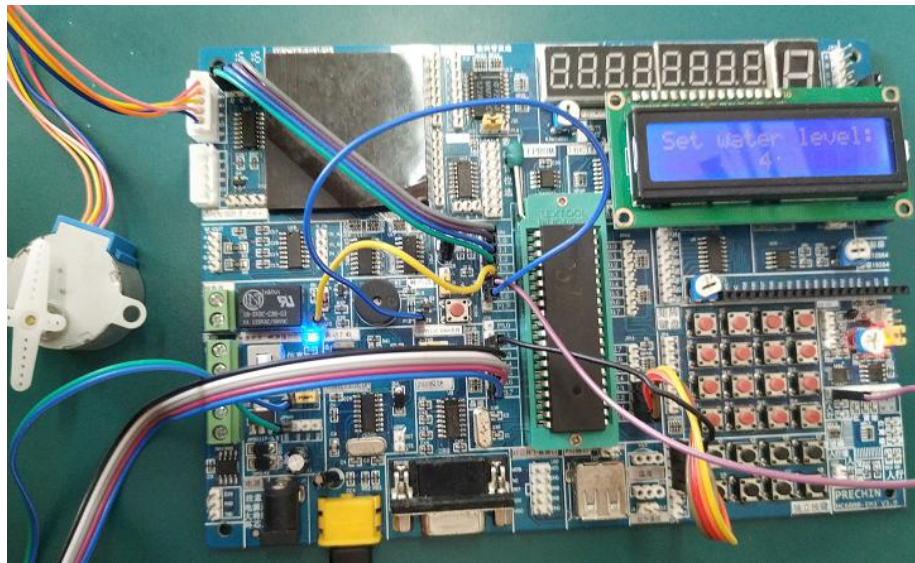


图 2.12 设置水位演示图

5.电机转动并显示剩余时间:

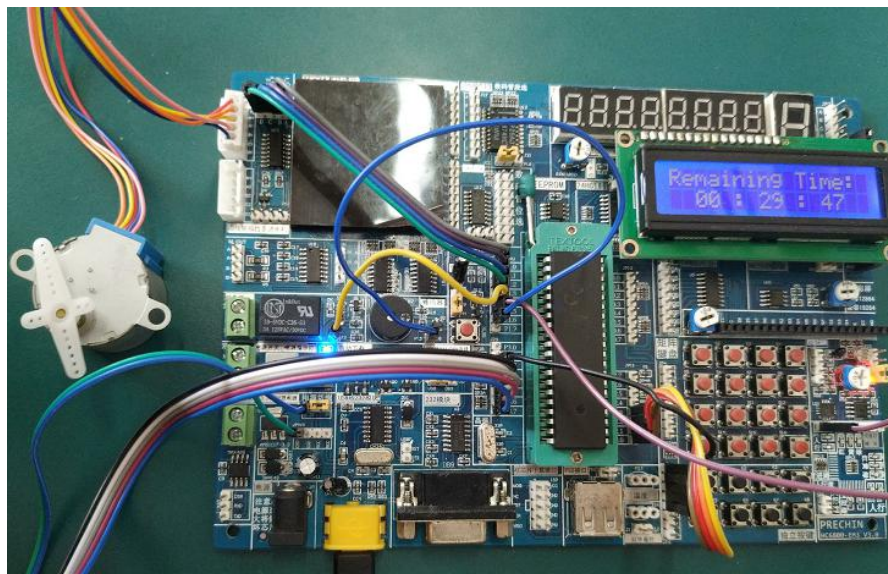


图 2.13 电机转动演示图

6.洗衣结束后指示灯亮、LCD 提示:

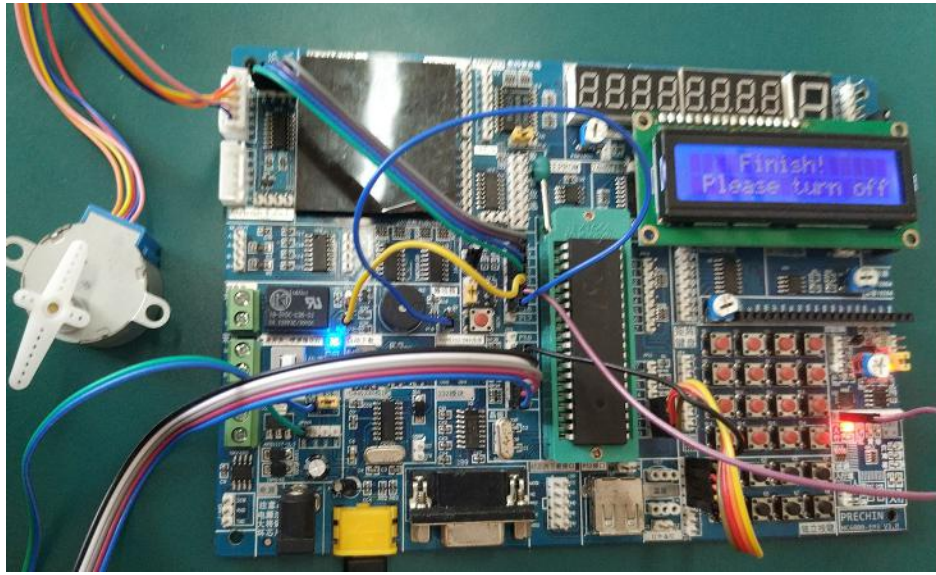


图 2.14 洗衣结束提示演示图

参考文献

- [1] 杜永峰,韦发清.基于单片机的智能洗衣机控制系统设计[J].电子世界,2020(10):153-154.
- [2] 张明德. 基于单片机的全自动洗衣机系统设计[D].安徽大学,2010.
- [3] 林义周.基于 51 单片机的全自动洗衣机控制器[J].福建信息技术教育,2005(02):7-13.
- [4] 史守坤.滚筒洗衣机智能控制系统研究与设计[D].南京理工大学,2012.
- [5] 李远锋.洗衣机发展研究[J].现代职业教育,2017 (3) : 184.
- [6] 克强.用 AT89C2051 单片机制作洗衣机控制电路[J].电子世界,2001(03):39-40+35-59.
- [7] 曾璐,李明.基于 AT89C52 单片机的洗衣机智能控制系统[J].家电科技,2007(02):37-38.
- [8] 杨雪.基于 AT89C51 单片机全自动洗衣机的仿真设计[J].黑龙江科技信息,2013(05):9.
- [9] 秦晓梅,巢明,崔承毅,赵权科,高庆华,王开宇.基于单片机技术的智能洗衣机控制系统设计[J].实验技术与管理,2019,36(10):193-196.
- [10] 黄河,刘福华,王信.基于单片机模糊推理的洗衣机控制系统设计[J].电子设计工程,2012,20(13):186-189+192.
- [11] 边疆.基于 MCS-51 单片机的洗衣机控制系统的实现[J].信息通信,2013(04):73-74.
- [12] 郭运峰,晏长岭.基于单片机的洗衣机控制电路[J].科技资讯,2015,13(16):46-47+49.
- [13] 陈继飞.单片机在全自动洗衣机中的应用[J].家用电器,2000(03):21-22.
- [14] 程阳.单片机实现洗衣机控制系统的工作[J].电子制作,2015(08):63.
- [15] 辽文.洗衣机单片机的管脚功能[J].家用电器,2000(09):18-19.
- [16] 杜永峰,韦发清.基于单片机的智能洗衣机控制系统设计[J].电子世界,2020(10):153-154.
- [17] 丁道宏.电力电子技术[M].北京:航空工业出版社,1999.

附 录

1.主函数部分

```
#include "keyscan.h"

INT8U time=0;
INT8U sec=60;
INT8U count=0;
INT8U pause=0;
INT8U water=0;
INT8U step=0;
INT8U tim=100;

void main (void)  {
    INT8U user_id[16]="User ID:";
    Lcd1602Init();
    TMOD = 0x21;
    ET1 = 0;
    TR1 = 1;
    EA = 1;
    ET0 = 1;
    IT0 = 1;
    EX0 = 1;
    PcdReset();
    PcdAntennaOff();
    PcdAntennaOn();
    Lcd1602ShowStr(0, 0, "2017218522 L W X");
    Lcd1602ShowStr(0, 1, "2017218272 Z   Y");
    delay10ms(100);
    Lcd1602ShowStr(0, 0, "Please swipecard");
    Lcd1602ShowStr(0, 1, " Loading.....  ");
    delay10ms(100);
    PcdFindcard(user_id+8);
```



```
Lcd1602ShowStr(0,0,user_id);
delay10ms(100);
Lcd1602ShowStr(0, 0, " Welcome to use ");
Lcd1602ShowStr(0, 1, "washing machine!");
delay10ms(100);
Lcd1602ShowStr(0, 0, "1.Normal 2.Fine ");
Lcd1602ShowStr(0, 1, "3Dry 4Self-clean");
delay10ms(100);
while(1){
    keyscan();
};
}

void int0() interrupt 0
{
    if(TR0){
        pause=0;
        Lcd1602ShowStr(0, 0, "      Pause      ");
        TR0=0;
        delay10ms(50);
    }
    else{
        pause=1;
        delay10ms(50);
        Lcd1602ShowStr(0, 0, "Remaining Time: ");
        TR0=1;
    }
}

void timer0() interrupt 1 //中断
{
    TL0 = 0xCD;
    TH0 = 0xD4;
```

```
count++;
if(count==tim){
    count=0;
    sec--;
    if(sec%10==1){
        step=!step;
    }
    if(sec==0){
        time--;
        sec=60;
    }
    if(time>60){
        Lcd1602SetCursor(3, 1);
        Lcd1602WriteData((time-1)/60+'0');
    }
    Lcd1602SetCursor(7, 1);
    Lcd1602WriteData((time-1)%60/10+'0');
    Lcd1602WriteData((time-1)%10+'0');
    Lcd1602SetCursor(12, 1);
    Lcd1602WriteData((sec-1)/10+'0');
    Lcd1602WriteData((sec-1)%10+'0');
}
}
```

2.Keyscan 部分

```
#ifndef __KEYSCAN_H__
#define __KEYSCAN_H__

#include "RC522.h"

#define FAST 6
#define MID 11
#define SLOW 16
```

```
extern void _nop_(void);
extern INT8U time;
extern INT8U sec;
extern INT8U count;
extern INT8U pause;
extern INT8U water;
extern INT8U step;
```

```
sbit K1=P2^0;
sbit K2=P2^1;
sbit K3=P2^2;
sbit K4=P2^3;
sbit MA=P1^0;
sbit MB=P1^1;
sbit MC=P1^2;
sbit MD=P1^3;
```

```
void delay(INT8U t);
void delay10ms(INT8U t);
void bee(INT8U mode);
void run(INT8U mode);
void keyscan();
void water_in();
```

```
#endif
```

```
#include "keyscan.h"
```

```
void delay(INT8U z)
{
    INT8U x,y;
    for(x=z;x>0;x--)
        for(y=110;y>0;y--);
}
```

```
}

void m_delay(INT8U t_mode){
    INT8U i,t;
    for(t=0;t<t_mode;t++)
        for(i=0;i<250;i++);
}

void delay10ms(INT8U t){
    INT8U i,j;
    while(t--){
        for(i=0;i<200;i++)
            for(j=0;j<110;j++);
    }
}

void bee(INT8U st){
    INT8U i,j;
    if(st==1){
        Lcd1602ShowStr(0, 0, "    Start!    ");
        Lcd1602ShowStr(0, 0, "                ");
    }
    else if(st==0){
        Lcd1602ShowStr(0, 0, "    Finish!    ");
        Lcd1602ShowStr(0, 1, " Please turn off");
    }
    else
        Lcd1602ShowStr(0, 0, "Remaining Time: ");
    LED=0;
    for(i=0;i<250;i++)
    {
        for(j=0;j<250;j++);
        SPK=~SPK;
    }
    delay10ms(20);
}
```

```
    LED=1;
    delay10ms(20);
    LED=0;
    for(i=0;i<250;i++)
    {
        for(j=0;j<250;j++);
        SPK=~SPK;
    }
    delay10ms(20);
    LED=1;
}

void run(INT8U mode){
    INT8U speed=MID;
    INT8U watt;
    switch(mode){
        case 1:time=30;speed=MID;break;
        case 2:time=60;speed=SLOW;break;
        case 3:time=5;speed=FAST;break;
        case 4:time=10;speed=FAST;break;
    }
    if(mode==1||mode==2){
        Lcd1602ShowStr(0, 0, "Set water level:");
        Lcd1602ShowStr(0, 1, "      0      ");
        delay10ms(100);
        water_in();
    }
    else if(mode==4){
        Lcd1602ShowStr(0, 0, "      Water in      ");
        Lcd1602ShowStr(0, 1, " Please hold on ");
        jdq=0;
        delay10ms(200);
        jdq=1;
    }
}
```

```
}  
bee(1);  
Lcd1602ShowStr(0, 0, "Remaining Time: ");  
Lcd1602ShowStr(0, 1, "  0  :    :    ");  
sec=60;  
Lcd1602SetCursor(3, 1);  
Lcd1602WriteData((time-1)/60+'0');  
Lcd1602SetCursor(7, 1);  
Lcd1602WriteData((time-1)%60/10+'0');  
Lcd1602WriteData((time-1)%10+'0');  
Lcd1602SetCursor(12, 1);  
Lcd1602WriteData((sec-1)/10+'0');  
Lcd1602WriteData((sec-1)%10+'0');  
TR0 = 1;  
pause=1;  
while(time){  
    if(pause){  
        if(step){  
            MA=1;MB=0;MC=0;MD=0;  
            m_delay(speed);  
            MA=1;MB=1;MC=0;MD=0;  
            m_delay(speed);  
            MA=0;MB=1;MC=0;MD=0;  
            m_delay(speed);  
            MA=0;MB=1;MC=1;MD=0;  
            m_delay(speed);  
            MA=0;MB=0;MC=1;MD=0;  
            m_delay(speed);  
            MA=0;MB=0;MC=1;MD=1;  
            m_delay(speed);  
            MA=0;MB=0;MC=0;MD=1;  
            m_delay(speed);  
            MA=1;MB=0;MC=0;MD=1;  
        }  
    }  
}
```

```
        m_delay(speed);
    }
    else{
        MA=1;MB=0;MC=0;MD=1;
        m_delay(speed);
        MA=0;MB=0;MC=0;MD=1;
        m_delay(speed);
        MA=0;MB=0;MC=1;MD=1;
        m_delay(speed);
        MA=0;MB=0;MC=1;MD=0;
        m_delay(speed);
        MA=0;MB=1;MC=1;MD=0;
        m_delay(speed);
        MA=0;MB=1;MC=0;MD=0;
        m_delay(speed);
        MA=1;MB=1;MC=0;MD=0;
        m_delay(speed);
        MA=1;MB=0;MC=0;MD=0;
        m_delay(speed);
    }
}

if((time%10==1&&time!=1&&mode==1)||((time%20==1&&time!=1&&mode==2)){
    Lcd1602ShowStr(0, 0, "    Water out    ");
    jdq=0;
    watt=water;
    while(watt--){
        Lcd1602SetCursor(15, 0);
        Lcd1602WriteData(watt+'1');
        delay10ms(100);
    }
    jdq=1;
    delay10ms(20);
}
```

```
    bee(2);
    Lcd1602ShowStr(0, 0, "    Water in    ");
    jdq=0;
    watt=water;
    while(watt--){
        Lcd1602SetCursor(15, 0);
        Lcd1602WriteData(watt+'1');
        delay10ms(100);
    }
    jdq=1;
    Lcd1602ShowStr(0, 0, "Remaining Time: ");
}
if(time==5&&sec==60&&mode!=3){
    Lcd1602ShowStr(0, 0, "    Water out    ");
    pause=0;
    TR0=0;
    jdq=0;
    watt=water;
    while(watt--){
        Lcd1602SetCursor(15, 0);
        Lcd1602WriteData(watt+'1');
        delay10ms(100);
    }
    jdq=1;
    delay10ms(20);
    bee(2);
    speed=FAST;
    Lcd1602ShowStr(0, 0, "Remaining Time: ");
    sec--;
    pause=1;
    TR0=1;
}
}
```



```
EA=0;
bee(0);
}

void keyscan(){

    if(K1==0){
        delay(5);
        if(K1==0){
            run(1);
            while(!K1);
        }
    }
    if(K2==0){
        delay(5);
        if(K2==0){
            run(2);
            while(!K2);
        }
    }
    if(K3==0){
        delay(5);
        if(K3==0){
            run(3);
            while(!K3);
        }
    }
    if(K4==0){
        delay(5);
        if(K4==0){
            run(4);
            while(!K4);
        }
    }
}
```

```
    }  
}  
  
void water_in(){  
    INT8U watt=0;  
    water=0;  
    while(1){  
        if(K1==0){  
            delay(5);  
            if(K1==0){  
                if(!step){  
                    if(water<9) water++;  
                    Lcd1602SetCursor(7, 1);  
                    Lcd1602WriteData(water+'0');  
                }  
                else{  
                    if(time<180) time+=10;  
                    Lcd1602SetCursor(6, 1);  
                    Lcd1602WriteData((time/100)%10+'0');  
                    Lcd1602WriteData((time/10)%10+'0');  
                }  
                while(!K1);  
            }  
        }  
        if(K2==0){  
            delay(5);  
            if(K2==0){  
                if(!step){  
                    if(water>1) water--;  
                    Lcd1602SetCursor(7, 1);  
                    Lcd1602WriteData(water+'0');  
                }  
                else{  

```

```
        if(time>10) time-=10;
        Lcd1602SetCursor(6, 1);
        Lcd1602WriteData((time/100)%10+'0');
        Lcd1602WriteData((time/10)%10+'0');
    }
    while(!K2);
}
}
if(K3==0){
    delay(5);
    if(K3==0){
        if(!step){
            Lcd1602ShowStr(0, 0, "Set washing time");
            Lcd1602SetCursor(6, 1);
            Lcd1602WriteData((time/100)%10+'0');
            Lcd1602WriteData((time/10)%10+'0');
            Lcd1602WriteData('0');
        }
        else{
            Lcd1602ShowStr(0, 0, "Set water level:");
            Lcd1602ShowStr(0, 1, "      0      ");
            Lcd1602SetCursor(7, 1);
            Lcd1602WriteData(water+'0');
        }
        step=!step;
        while(!K3);
    }
}
if(K4==0){
    delay(5);
    if(K4==0){
        while(!K4);
        break;
    }
}
```

```

    }
}
delay10ms(10);
}
LED=0;
delay10ms(100);
Lcd1602ShowStr(0, 0, "    Water in    ");
Lcd1602ShowStr(0, 1, " Please hold on ");
LED=1;
jdq=0;
watt=water;
while(watt--){
    Lcd1602SetCursor(15, 0);
    Lcd1602WriteData(watt+'1');
    delay10ms(100);
}
jdq=1;
}

```

3.rc522 部分

```

#ifndef __RC522_H__
#define __RC522_H__

```

```

#include "lcd1602.h"

```

```

#include <intrins.h>

```

```

sbit    MF522_SDA  = P3^7 ;    //SDA
sbit    MF522_SCK  = P3^6;      //SCK
sbit    MF522_MOSI = P3^5;      //MOSI
sbit    MF522_MISO = P3^4;      //MISO
sbit    MF522_RST  = P3^3;      //RST
sbit    jdq=P1^4;
sbit    LED=P1^5;
sbit    SPK=P1^6;

```

```

extern INT8U tim;

char PcdReset(void);
char PcdFindcard(unsigned char *pSnr);
void PcdAntennaOn(void);
void PcdAntennaOff(void);
char PcdRequest(unsigned char req_code,unsigned char *pTagType);
char PcdAnticoll(unsigned char *pSnr);
char PcdHalt(void);
char PcdComMF522(unsigned char Command,
                  unsigned char *pInData,
                  unsigned char InLenByte,
                  unsigned char *pOutData,
                  unsigned int *pOutLenBit);
void CalulateCRC(unsigned char *pIndata,unsigned char len,unsigned char
*pOutData);
void WriteRawRC(unsigned char Address,unsigned char value);
unsigned char ReadRawRC(unsigned char Address);
void SetBitMask(unsigned char reg,unsigned char mask);
void ClearBitMask(unsigned char reg,unsigned char mask);

#define PCD_IDLE                0x00                //取消当前命令
#define PCD_AUTHENT             0x0E                //验证密钥
#define PCD_RECEIVE             0x08                //接收数据
#define PCD_TRANSMIT            0x04                //发送数据
#define PCD_TRANSCEIVE          0x0C                //发送并接收数据
#define PCD_RESETPHASE          0x0F                //复位
#define PCD_CALCCRC             0x03                //CRC 计算

#define PICC_REQIDL             0x26                //寻天线区内未进入休眠状态
#define PICC_REQALL            0x52                //寻天线区内全部卡
#define PICC_ANTICOLL1          0x93                //防冲撞

```

#define PICC_ANTICOLL2	0x95	//防冲撞
#define PICC_AUTHENT1A	0x60	//验证 A 密钥
#define PICC_AUTHENT1B	0x61	//验证 B 密钥
#define PICC_READ	0x30	//读块
#define PICC_WRITE	0xA0	//写块
#define PICC_DECREMENT	0xC0	//扣款
#define PICC_INCREMENT	0xC1	//充值
#define PICC_RESTORE	0xC2	//调块数据到缓冲区
#define PICC_TRANSFER	0xB0	//保存缓冲区中数据
#define PICC_HALT	0x50	//休眠
#define DEF_FIFO_LENGTH	64	//FIFO size=64byte
#define RFU00	0x00	
#define CommandReg	0x01	
#define ComIEnReg	0x02	
#define DivIEnReg	0x03	
#define ComIrqReg	0x04	
#define DivIrqReg	0x05	
#define ErrorReg	0x06	
#define Status1Reg	0x07	
#define Status2Reg	0x08	
#define FIFODataReg	0x09	
#define FIFOLevelReg	0x0A	
#define WaterLevelReg	0x0B	
#define ControlReg	0x0C	
#define BitFramingReg	0x0D	
#define CollReg	0x0E	
#define RFU0F	0x0F	
#define RFU10	0x10	
#define ModeReg	0x11	
#define TxModeReg	0x12	
#define RxModeReg	0x13	

#define	TxControlReg	0x14
#define	TxAutoReg	0x15
#define	TxSelReg	0x16
#define	RxSelReg	0x17
#define	RxThresholdReg	0x18
#define	DemodReg	0x19
#define	RFU1A	0x1A
#define	RFU1B	0x1B
#define	MifareReg	0x1C
#define	RFU1D	0x1D
#define	RFU1E	0x1E
#define	SerialSpeedReg	0x1F
#define	RFU20	0x20
#define	CRCResultRegM	0x21
#define	CRCResultRegL	0x22
#define	RFU23	0x23
#define	ModWidthReg	0x24
#define	RFU25	0x25
#define	RFCfgReg	0x26
#define	GsNReg	0x27
#define	CWGSCfgReg	0x28
#define	ModGsCfgReg	0x29
#define	TModeReg	0x2A
#define	TPrescalerReg	0x2B
#define	TReloadRegH	0x2C
#define	TReloadRegL	0x2D
#define	TCounterValueRegH	0x2E
#define	TCounterValueRegL	0x2F
#define	RFU30	0x30
#define	TestSel1Reg	0x31
#define	TestSel2Reg	0x32
#define	TestPinEnReg	0x33
#define	TestPinValueReg	0x34

```
#define TestBusReg      0x35
#define AutoTestReg     0x36
#define VersionReg      0x37
#define AnalogTestReg   0x38
#define TestDAC1Reg     0x39
#define TestDAC2Reg     0x3A
#define TestADCReg      0x3B
#define RFU3C           0x3C
#define RFU3D           0x3D
#define RFU3E           0x3E
#define RFU3F           0x3F

#define MI_OK           0
#define MI_NOTAGERR     (-1)
#define MI_ERR          (-2)

#endif

#include "RC522.h"

#define MAXRLEN 18

void delay2(unsigned int z)
{
    unsigned int x,y;
    for(x=z;x>0;x--)
        for(y=110;y>0;y--);
}

char PcdFindcard(unsigned char *pSnr)
{
    INT8U status,i,j,temp,g_ucTempbuf[20],su[4]={0xc1,0x68,0xf7,0x7b},svip=1;
```



```
while(1)
{
    status = PcdRequest(PICC_REQALL, g_ucTempbuf);//寻卡
    if(status!=MI_OK)
    {
        PcdReset();
        PcdAntennaOff();
        PcdAntennaOn();
        continue;
    }
    status = PcdAnticoll(g_ucTempbuf);//防冲撞
    if (status != MI_OK)
    {
        continue;
    }
    else
    {
        for(i=0;i<4;i++)
        {
            temp=g_ucTempbuf[i];
            if(temp!=su[i])
                svip=0;
            if(temp/16<10)
                *(pSnr+2*i)=temp/16+'0';
            else
                *(pSnr+2*i)=temp/16+'A'-10;
            if(temp%16<10)
                *(pSnr+2*i+1)=temp%16+'0';
            else
                *(pSnr+2*i+1)=temp%16+'A'-10;
            delay2(10);
        }
        if(svip){
            tim=20;
            *(pSnr-4)='V';
```

```

        *(pSnr-2)='P';
    }
    LED=0;
    for(i=0;i<250;i++)
    {
        for(j=0;j<250;j++);
        SPK=~SPK;
    }
    delay2(20);
    LED=1;
    PcdHalt();
    return 1;
}
delay2(200);
}
}

char PcdRequest(unsigned char req_code,unsigned char *pTagType)
{
    char status;
    unsigned int  unLen;
    unsigned char ucComMF522Buf[MAXRLEN];

    ClearBitMask(Status2Reg,0x08);
    WriteRawRC(BitFramingReg,0x07);
    SetBitMask(TxControlReg,0x03);
    ucComMF522Buf[0] = req_code;
    status =
PcdComMF522(PCD_TRANSCEIVE,ucComMF522Buf,1,ucComMF522Buf,&unLen);
    if ((status == MI_OK) && (unLen == 0x10))
    {
        *pTagType      = ucComMF522Buf[0];
        *(pTagType+1) = ucComMF522Buf[1];
    }
}

```

```

else
{
    status = MI_ERR;
}
return status;
}

char PcdAnticoll(unsigned char *pSnr)
{
    char status;
    unsigned char i,snr_check=0;
    unsigned int    unLen;
    unsigned char ucComMF522Buf[MAXRLEN];

    ClearBitMask(Status2Reg,0x08);
    WriteRawRC(BitFramingReg,0x00);
    ClearBitMask(CollReg,0x80);
    ucComMF522Buf[0] = PICC_ANTICOLL1;
    ucComMF522Buf[1] = 0x20;
    status = PcdComMF522(PCD_TRANSCEIVE,ucComMF522Buf,2,ucComMF522Buf,&unLen);
    if (status == MI_OK)
    {
        for (i=0; i<4; i++)
        {
            *(pSnr+i) = ucComMF522Buf[i];
            snr_check ^= ucComMF522Buf[i];
        }
        if (snr_check != ucComMF522Buf[i])
        {
            status = MI_ERR;
        }
    }
    SetBitMask(CollReg,0x80);
    return status;
}

```

```

char PcdHalt(void)
{
    char status;
    unsigned int  unLen;
    unsigned char ucComMF522Buf[MAXRLEN];
    ucComMF522Buf[0] = PICC_HALT;
    ucComMF522Buf[1] = 0;
    CalulateCRC(ucComMF522Buf,2,&ucComMF522Buf[2]);
    status =
PcdComMF522(PCD_TRANSCEIVE,ucComMF522Buf,4,ucComMF522Buf,&unLen);
    return MI_OK;
}

void CalulateCRC(unsigned char *pIndata,unsigned char len,unsigned char
*pOutData)
{
    unsigned char i,n;
    ClearBitMask(DivIrqReg,0x04);
    WriteRawRC(CommandReg,PCD_IDLE);
    SetBitMask(FIFOLevelReg,0x80);
    for (i=0; i<len; i++)
    {   WriteRawRC(FIFODataReg, *(pIndata+i));   }
    WriteRawRC(CommandReg, PCD_CALCCRC);
    i = 0xFF;
    do
    {
        n = ReadRawRC(DivIrqReg);
        i--;
    }
    while ((i!=0) && !(n&0x04));
    pOutData[0] = ReadRawRC(CRCResultRegL);
    pOutData[1] = ReadRawRC(CRCResultRegM);
}

```

```

    }

char PcdReset(void)
{
    //unsigned char i;
    MF522_RST=1;
    _nop_();
    MF522_RST=0;
    _nop_();
    MF522_RST=1;
    delay2(10);
    WriteRawRC(CommandReg,PCD_RESETPHASE);
    _nop_();
    WriteRawRC(ModeReg,0x3D);           //和 Mifare 卡通讯, CRC 初始值
0x6363
    WriteRawRC(TReloadRegL,30);
    WriteRawRC(TReloadRegH,0);
    WriteRawRC(TModeReg,0x8D);
    WriteRawRC(TPrescalerReg,0x3E);
    WriteRawRC(TxAutoReg,0x40);
    return MI_OK;
}

unsigned char ReadRawRC(unsigned char Address)
{
    unsigned char i, ucAddr;
    unsigned char ucResult=0;
    MF522_SCK = 0;
    MF522_SDA = 0;
    ucAddr = ((Address<<1)&0x7E)|0x80;
    for(i=8;i>0;i--)
    {
        MF522_MOSI = ((ucAddr&0x80)==0x80);
    }
}

```

```
MF522_SCK = 1;
ucAddr <<= 1;
MF522_SCK = 0;
}
for(i=8;i>0;i--)
{
MF522_SCK = 1;
ucResult <<= 1;
ucResult|=(bit)MF522_MISO;
MF522_SCK = 0;
}
MF522_SDA = 1;
MF522_SCK = 1;
return ucResult;
}

void WriteRawRC(unsigned char Address, unsigned char value)
{
unsigned char i, ucAddr;
MF522_SCK = 0;
MF522_SDA = 0;
ucAddr = ((Address<<1)&0x7E);
for(i=8;i>0;i--)
{
MF522_MOSI = ((ucAddr&0x80)==0x80);
MF522_SCK = 1;
ucAddr <<= 1;
MF522_SCK = 0;
}
for(i=8;i>0;i--)
{
MF522_MOSI = ((value&0x80)==0x80);
MF522_SCK = 1;
```

```
        value <<= 1;
        MF522_SCK = 0;
    }
    MF522_SDA = 1;
    MF522_SCK = 1;
}

void SetBitMask(unsigned char reg,unsigned char mask)
{
    char tmp = 0x0;
    tmp = ReadRawRC(reg);
    WriteRawRC(reg,tmp | mask);  // set bit mask
}

void ClearBitMask(unsigned char reg,unsigned char mask)
{
    char tmp = 0x0;
    tmp = ReadRawRC(reg);
    WriteRawRC(reg, tmp & ~mask);  // clear bit mask
}

char PcdComMF522(unsigned char Command,
                  unsigned char *pInData,
                  unsigned char InLenByte,
                  unsigned char *pOutData,
                  unsigned int  *pOutLenBit)
{
    char status = MI_ERR;
    unsigned char irqEn    = 0x00;
    unsigned char waitFor = 0x00;
    unsigned char lastBits;
    unsigned char n;
    unsigned int i;
```

```
switch (Command)
{
    case PCD_AUTHENT:
        irqEn    = 0x12;
        waitFor = 0x10;
        break;
    case PCD_TRANSCEIVE:
        irqEn    = 0x77;
        waitFor = 0x30;
        break;
    default:
        break;
}

WriteRawRC(ComIrqReg, irqEn|0x80);
ClearBitMask(ComIrqReg, 0x80);
WriteRawRC(CommandReg, PCD_IDLE);
SetBitMask(FIFOLevelReg, 0x80);
for (i=0; i<InLenByte; i++)
{    WriteRawRC(FIFODataReg, pInData[i]);    }
WriteRawRC(CommandReg, Command);
if (Command == PCD_TRANSCEIVE)
{    SetBitMask(BitFramingReg, 0x80);    }
i = 600; //根据时钟频率调整, 操作 M1 卡最大等待时间 25ms
do
{
    n = ReadRawRC(ComIrqReg);
    i--;
}
while ((i!=0) && !(n&0x01) && !(n&waitFor));
ClearBitMask(BitFramingReg, 0x80);
if (i!=0)
{
```



```
if(!(ReadRawRC(ErrorReg)&0x1B))
{
    status = MI_OK;
    if (n & irqEn & 0x01)
    {
        status = MI_NOTAGERR;
    }
    if (Command == PCD_TRANSCEIVE)
    {
        n = ReadRawRC(FIFOLevelReg);
        lastBits = ReadRawRC(ControlReg) & 0x07;
        if (lastBits)
        {
            *pOutLenBit = (n-1)*8 + lastBits;
        }
        else
        {
            *pOutLenBit = n*8;
        }
        if (n == 0)
        {
            n = 1;
        }
        if (n > MAXRLEN)
        {
            n = MAXRLEN;
        }
        for (i=0; i<n; i++)
        {
            pOutData[i] = ReadRawRC(FIFODataReg);
        }
    }
    else
    {
        status = MI_ERR;
    }
}
SetBitMask(ControlReg,0x80);           // stop timer now
WriteRawRC(CommandReg,PCD_IDLE);
return status;
}

void PcdAntennaOn()
{
    unsigned char i;
    i = ReadRawRC(TxControlReg);
```

```
    if (!(i & 0x03))
    {
        SetBitMask(TxControlReg, 0x03);
    }
}
```

```
void PcdAntennaOff()
{
    ClearBitMask(TxControlReg, 0x03);
}
```

4.lcd1602 部分

```
#ifndef __LCD1602_H__
#define __LCD1602_H__
```

```
#include "reg52.h"
```

```
#define INT8U unsigned char
```

```
#define INT16U unsigned int
```

```
#define LCD1602_DATA_PORT P0
```

```
sbit gLcd1602_E = P2^7; // LCD1602 控制总线的使能信号
```

```
sbit gLcd1602_RW = P2^5; // LCD1602 控制总线的读写选择信号
```

```
sbit gLcd1602_RS = P2^6; // LCD1602 控制总线的数据/命令选择信号
```

```
static void delay5ms(void);
```

```
static void Lcd1602WaitNoBusy(void);
```

```
static void Lcd1602WriteCmd(INT8U cmd);
```

```
void Lcd1602WriteData(INT8U dat);
```

```
void Lcd1602Init(void);
```

```
void Lcd1602SetCursor(INT8U x, INT8U y);
```

```
void Lcd1602ShowStr(INT8U x, INT8U y, INT8U *pStr);
```

```
#endif
```

```
#include "lcd1602.h"

static void Lcd1602WaitNoBusy(void) //忙检测函数，判断 bit7 是 0，允许执行；
1 禁止
{
    INT8U sta = 0;        //
    LCD1602_DATA_PORT = 0xff;
    gLcd1602_RS = 0;
    gLcd1602_RW = 1;
    do
    {
        gLcd1602_E = 1;
        sta = LCD1602_DATA_PORT;
        gLcd1602_E = 0;    //使能，用完就拉低，释放总线
    }while(sta & 0x80);
}

static void Lcd1602WriteCmd(INT8U cmd)
{
    Lcd1602WaitNoBusy();    // 先等待 LCD1602 处于不忙状态

    gLcd1602_E = 0;        // 禁止 LCD
    gLcd1602_RS = 0;        // 选择发送命令模式
    gLcd1602_RW = 0;        // 选择写入模式
    LCD1602_DATA_PORT = cmd;    // 将 1 字节命令字放入 8 位并行数据端口
    gLcd1602_E = 1;        // 使能 LED
    gLcd1602_E = 0;        // 禁止 LCD
}

void Lcd1602WriteData(INT8U dat)
{
    Lcd1602WaitNoBusy();    // 先等待 LCD1602 处于不忙状态

    gLcd1602_E = 0;        // 禁止 LCD
```

```

    gLcd1602_RS = 1;          // 选择发送数据模式
    gLcd1602_RW = 0;          // 选择写入模式
    LCD1602_DATA_PORT = dat;  // 将 1 字节命令字放入 8 位并行数据端口
    gLcd1602_E = 1;           // 使能 LED
    gLcd1602_E = 0;           // 禁止 LCD
}

void Lcd1602Init(void)
{
    Lcd1602WriteCmd(0x38);    // 按照数据手册的初始化时序，先发送 38H
    delay5ms();               // 延时 5ms
    Lcd1602WriteCmd(0x38);    // 按照数据手册的初始化时序，先发送 38H
    delay5ms();               // 延时 5ms
    Lcd1602WriteCmd(0x38);    // 按照数据手册的初始化时序，先发送 38H
    delay5ms();               // 延时 5ms
    Lcd1602WriteCmd(0x38);    // 显示模式设置
    Lcd1602WriteCmd(0x08);    // 关闭显示
    Lcd1602WriteCmd(0x01);    // 清屏（同时清数据指针）
    Lcd1602WriteCmd(0x06);    // 读写后指针自动加 1，写 1 个字符后整屏
显示不移动
    Lcd1602WriteCmd(0x0c);    // 开显示，不显示光标
}

void Lcd1602SetCursor(INT8U x, INT8U y)    // 坐标显示
{
    INT8U addr = 0;

    switch (y)
    {
        case 0:                // 上面一行
            addr = 0x00 + x;    break;
        case 1:                // 下面一行
            addr = 0x40 + x;    break;
    }
}

```

```
        default:
            break;
    }
    Lcd1602WriteCmd(addr | 0x80);
}

void Lcd1602ShowStr(INT8U x, INT8U y, INT8U *pStr)    //显示字符串
{
    Lcd1602SetCursor(x, y);    //当前字符的坐标
    while (*pStr != '\0')
    {
        Lcd1602WriteData(*pStr++);
    }
}

void delay5ms(void)
{
    INT8U i,j;
    for(i=0;i<100;i++)
        for(j=0;j<110;j++);
}
```

指导教师考核意见

（考查学生是否完成电子信息科学与技术综合设计，完成本次电子信息科学与技术综合设计的水平，研究和解决实际问题的意识和能力，学习态度、综合素质、品德纪律等情况）

综合成绩评定（百分制或五级制）：_____（60 分以上为考核通过）

指导教师签名：

年 月 日