

天津职业技术师范大学

Tianjin University of Technology and Education

期末设计

基于 STM32 的自动售货机

Vending machine based on STM32

专 业:	自动化 (电气技术教育)
班级学号:	电气 2141-0203214108
学生姓名:	王煜涵
指导教师:	李会艳

二〇二二年六月

天津职业技术师范大学本科生期末设计

基于 STM32 的自动售货机

Vending machine based on STM32

专业班级:	电气 2141
学生姓名:	王煜涵
指导教师:	李会艳
学 院:	电气工程与自动化学院

2022 年 6 月

摘 要

自动贩卖机是能够根据投入的钱币自动付货的机器，它是商业自动化的常用设备，不受时间、地点的限制，能节省人力、方便交易。现代的自动售货机的种类、结构和功能依出售的商品而异。钱币装置是售货机的核心，其主要功能是确认投入钱币的金额，如果投入的金额达到购买物品的数值即发出售货信号，并进行找零。自动售货机自动输出;顾客所要的商品。因此，自动售货机在商业，食品行业越来越普及，更方便消费者购买自己所需的商品。使用方便可靠、节省人力的自动售货机将在社会生活中成为一种趋势。

本文详细介绍了自动售货机系统以 STM32F103C8T6 芯片为核心，运用 C 语言在 Keil 软件上进行编写，编译，在 Proteus 上进行仿真并通过选择矩阵按键，OLED 屏幕显示和串口通信，实现自动售货机的货物选择，投币显示，出货，找零等功能。本文并详细介绍了自动售货机系统的设计方案、硬件选择、软件规划和编写。重点描述了自动售货机的工作原理、系统设计和软件编程

关键词：自动售货机；STM32F103C8T6；串口通信；OLED 屏幕

ABSTRACT

Vending machine is a machine that can automatically pay goods according to the coins invested, it is a commonly used equipment for commercial automation, not limited by time and place, can save manpower and facilitate transactions. The type, structure, and function of modern vending machines vary depending on the goods sold. The numismatic device is the core of the vending machine, and its main function is to confirm the amount of coins invested, and if the amount of money invested reaches the value of the purchased item, it will send a signal to sell the goods and make change. Vending machine automatic output; The goods that the customer wants. Therefore, vending machines are becoming more and more popular in the commercial and food industries, making it more convenient for consumers to buy the goods they need. Easy to use, reliable, labor-saving vending machines will become a trend in social life.

This article details the vending machine system with STM32F103C8T6 chip as the core, using the C language to write on Keil software, compile, simulate on Proteus and through the selection matrix button, OLED screen display and serial port communication, to achieve the vending machine's cargo selection, coin display, shipment, change and other functions. This article also details the design, hardware selection, software planning, and writing of vending machine systems. The focus is on how vending machines work, system design, and software programming

Key Words: Vending machines; STM32F103C8T6; Serial communication; OLED screen

目 录

1 引言.....	1
1.1 课题背景.....	1
2 设计方案.....	1
2.1 设计要求.....	1
2.2 软件与硬件整体设计框图.....	2
3 硬件设计.....	3
3.1 元器件选型.....	3
3.1.1 主控芯片选型.....	3
3.1.2 输入模块选型.....	4
3.1.3 人机交互显示模块选型.....	4
3.1.4 电机驱动器选型.....	4
3.2 硬件电路设计.....	5
3.2.1 电源电路设计.....	5
3.2.2 主控芯片电路设计.....	7
3.2.3 串口通信电路设计.....	7
3.2.4 矩阵键盘电路设计.....	8
3.2.5 整体电路与 PCB 设计图.....	8
4.软件程序设计.....	10
4.1 程序设计思路.....	10
4.1.2.HAL 库介绍.....	10
4.1.1 状态机介绍.....	11
4.1.2 I2C 通信介绍.....	11
4.2 编程软件介绍.....	12
4.2.1 STM32CubeMX 介绍.....	13
4.2.2 KEIL 软件介绍.....	14
4.3 程序功能要求.....	15
4.4 矩阵键盘程序编写.....	16
4.4.1 矩阵键盘程序流程.....	16
4.4.2 矩阵键盘程序.....	16
4.4 OLED 屏幕程序编写.....	21
4.4.1 OLED 屏幕介绍.....	22
4.4.2 OLED 屏幕程序介绍与工作流程.....	22
4.4.3 OLED 程序编写.....	24

4.5 串口通信程序	26
4.5.1 串口通信工作流程	26
4.5.2 串口通信编写	27
4.6 主程序流程编写	29
4.6.1 主要逻辑程序编写	30
5 电路与程序功能仿真	32
5.1 软件介绍	32
5.1.1 仿真软件 Proteus 介绍	32
5.1.2 虚拟串口软件 Configure Virtual Serial Port Driver 介绍	33
5.2 仿真图的绘制	34
5.2.1 各主要模块仿真细节图	35
5.3 Virtual Serial Port Driver 虚拟串口软件设置	38
5.4 Proteus 仿真设置与仿真过程	39
5.4.1 Proteus 仿真设置	39
5.4.2 仿真过程	41
6 结论	46

1 引言

1.1 课题背景

自动售货机又称自动售货亭是能根据投入的钱币自动付货的机器,是商业化常用设备。一般的自动售货机系统由货币识别系统, 按键系统, 出货系统等组成。

公元 1 世纪, 希腊人希罗制造的自动出售圣水的装置是世界上最早的自动售货机。1925 年美国研制出售香烟的自动售货机, 此后又出现了出售邮票、车票的各种现代自动售货机。现代自动售货机的种类、结构和功能依出售的物品而异, 主要有食品、饮料、香烟、邮票, 车票日用品等自动售货机。自动售货机是一种全新的商业零售形式, 20 世纪 70 年代自日本和欧美发展起来。它又被称为 24 小时营业的微型超市。在日本, 70% 的罐装饮料是通过自动售货机售出的。全球著名饮料商可口可乐公司在全世界就布有 50 万台饮料自动售货机。但是在中国市场, 自动售货机出售的商品一般比超市、便利店的贵。但在美国, 日本这些自动售货机已经十分发达的国家, 自动售货机出售商品的价格是浮动的, 跟机器所处位置、周边商业环境相关。在美国, 自动售货机的竞争对手主要是便利店, 与便利店为邻的自动售货机出售的商品往往以价格便宜取胜。南开大学市场营销学韩德昌教授指出, 自动售货机具有技术含量高销售方式新、市场潜力大、商品货真价实等优点, 决定了它广阔的发展前景。而日, 网络与电子购物等新型消费方式相通, 能创造巨大的商机。在天津南开戈德股份有限公司的展览台前看到, 只要顾客用手机拨打自动售货机上显示的一个号码, 就可以通过手机付费的方式买到东西, 顾客只需月底支付手机账单。专家预测, 未来几年中国市场上的自动售货机将每年增加 5 到 10 万台。中国沿海比较富裕的地区, 将大约有 3.5 亿人口经常使用自动售货机。

2 设计方案

2.1 设计要求

设计一台销售袋装零食或罐装饮品的自动售货机, 具有硬币或纸币识别功

能，币值累加，自动售货，自动找零等功能。

基本设计要求：

- ①：可识别一元，五元，十元，二十元，五十元，一百元纸币（硬币），该功能使用串口通信的方式模拟识别。
- ②：有货物选择按键，有货物选择指示灯用于提示已选则货物。
- ③：具有找零功能和出货功能。
- ④：可以使用按键打开柜门便于补货。

2.2 软件与硬件整体设计框图

根据对自动售货机设计要求分析，完整的售货系统主要包括以下几个模块：商品选择模块（按键输入模块），显示模块，物品选择指示模块，电机与电机驱动模块与主控 CPU（Central Processing Unit，中央处理器）。

根据设计要求对程序部分进行设计分析，本程序需要对多个模块进行交互设计，多个模块需要与主控芯片进行串行或并行通信。总体设计框图如图 2-1 所示。

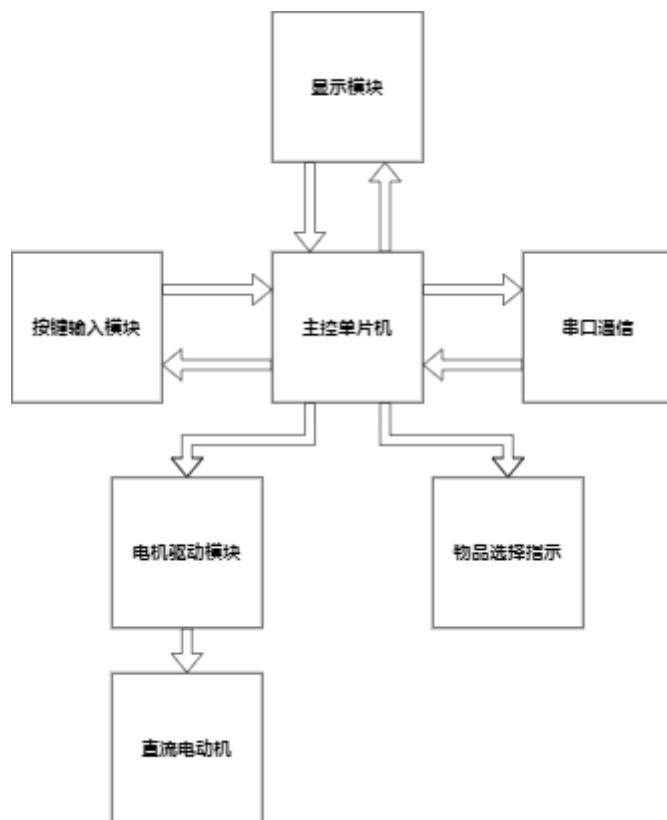


图 2-1 硬件设计连接框图

3 硬件设计

自动售货机硬件电路设计与元件选型主要考虑一些因素：

(1)：首先考虑的是设计成本，设计产品想要有一个好的利润必须要控制好成本问题，多以在保证硬件电路可以稳定运行的情况下选择的主频高且价格低廉，运行功耗小的元器件。

(2)：考虑到目前市面上的产品趋于智能化，可扩展化，所以根据这一趋势需要选择可扩展性强，兼容性好的驱动器与控制器，所以我采用了意法半导体公司的 STM32 系列单片机作为该次设计的主控芯片

3.1 元器件选型

3.1.1 主控芯片选型

方案一：采用 AT89C52 系列单片机作为主控芯片。AT89C52 是一个低电压，高性能 CMOS 8 位单片机，片内含 8k bytes 的可反复擦写的 Flash 只读程序存储器和 256 bytes 的随机存取数据存储器（RAM），器件采用 ATMEL 公司的高密度、非易失性存储技术生产，兼容标准 MCS-51 指令系统，片内置通用 8 位中央处理器和 Flash 存储单元，AT89C52 单片机在电子行业中有着广泛的应用。AT89C52 有 40 个引脚，32 个外部双向输入/输出(I/O)端口，同时内含 2 个外中断口，3 个 16 位可编程定时计数器,2 个全双工串行通信口，2 个读写口线，AT89C52 可以按照常规方法进行编程，也可以在线编程。其将通用的微处理器和 Flash 存储器结合在一起，特别是可反复擦写的 Flash 存储器可有效地降低开发成本。

方案二：采用 STM32F103C8T6 单片机作为主控芯片。STM32F103 系列单片机基于 Cortex-M3 芯片具有 72 MHz CPU 的速度和高达 1 MB 的闪存。包含电动机控制外围设备以及 CAN 和 USB 全速接口。STM32 系列 ARM Cortex-M3 32 位的闪存微控制器工作时具有低功率、低电压，并结合了实时功能的极佳性能。封装类型系列可用于您的嵌入式应用。MCU 体系结构具有一个易于使用的 STM32 平台，可用于包括电动机驱动；PC 和游戏；HVAC 和工业应用在内的应用。32 位 RISC 引脚到引脚软件兼容 SRAM 高达 96 Kb 闪存高达 1MB 电源：2

V 至 3.6 V 温度范围: -40 至+85° C 或-40 至+105° C STM32F1 系列 32 位 ARM®Cortex®-M3 微控制器, STMicroelectronics 32 位的闪存微控制器的 STM32 系列基于 ARM Cortex M3 核心, 为嵌入式应用特别开发的核心。STM32 系列得益于 Cortex-M3 体系结构增强功能, 包括为传达改进性能而设置的 Thumb-2 指令, 具有更好的编码密度, 对中断更快的反应, 所有的均和领先的工业功耗相接合。出色的实时表现卓越功效和新型的外围设备最大程度的集成式多引脚, 外围设备和软件兼容性。STM32F103C8T6 是一个中密度性能线, 配有 ARM Cortex-M3 32 位微控制器, LQFP 封装, 它结合了高性能的 ARM Cortex-M 内核, 运行频率可达 72MHz, 以及高速度内嵌存储器, 增强范围的强化输入/输出和外部连接至两个 APB 总线。STM32F103C8T6 具有 12 位模数转换器, 计时器, PWM 计时器, 标准和高级通讯接口, 一套全面的省电模式允许设计者设计低功耗应用。

通过上述两种方案对比可发现 STM32F103C8T6 具有更好性能与更好的扩展性能和相对丰富的总线, 所以本次设计我们将采用 STM32F103C8T6 作为主控芯片。

3.1.2 输入模块选型

输入模块将采用按键输入的方式来实现与单片机的交互。

方案一: 采用独立按键的设计, 使用独立按键需要将每一个按键连接至单片机, 本次设计需要使用 16 个独立按键。

方案二: 采用矩阵按键的设计, 4*4 矩阵键盘, 同样满足本次 16 个按键的需求。

对比上述两种方案, 方案一独立按键需要将每一个按键连接至 I/O 点, 需要占用 16 个 I/O 口, 但选用的单片机仅有 32 个可用 I/O, 这样造成了大量的 I/O 占用, 方案二矩阵按键的设计方式, 同样 16 个按键仅需 8 个 I/O 口便可实现与单片机的交互, 所以本次设计选用方案二矩阵按键的方式与单片机进行交互。

3.1.3 人机交互显示模块选型

自动售货机多需要满足 24 小时工作, 所以本次设计将采用 OLED 显示屏与 LED 灯 共同作为人机交互显示模块, 以提示使用者当前选择的商品与当前的使用状态。

3.1.4 电机驱动器选型

方案一: 采用 L298N 电机驱动器。L298N 就是 L298 的立式封装, 源自意

法半导体集团旗下品牌产品，是一款可接受高电压、大电流双路全桥式电机驱动芯片，工作电压可达 46V，输出电流最高可至 4A，采用 Multiwatt 15 脚封装，接受标准 TTL 逻辑电平信号，具有两个使能控制端，在不受输入信号影响的情况下通过板载跳帽插拔的方式，动态调整电路运作方式，有一个逻辑电源输入端，通过内置的稳压芯片 78MO5，使内部逻辑电路部分在低电压下工作，也可以对外输出逻辑电压 5V，为了避免稳压芯片损坏，当使用大于 12V 驱动电压时，务必使用外置的 5V 接口独立供电。L298N 通过控制主控芯片上的 I/O 输入端，直接通过电源来调节输出电压，即可实现电机的正转、反转、停止，由于电路简单，使用方便，通常情况下 L298N 可直接驱动继电器（四路）、螺线管、电磁阀、直流电机（两台）以及步进电机（一台两相或四相）。主要特点是：1. 发热量低 2. 抗干扰能力强 3. 驱动能力强（高电压、大电流）4. 可靠性高（使用大容量滤波电容，续流保护二极管可过热自断和反馈检测）5. 工作电压高（最高可至 46V）6. 输出电流大（瞬间峰值电流可达 3A，持续工作电流为 2A）7. 额定功率 25W（电压 X 电流）。

方案二：采用 TB6612FNG 电机驱动器。TB6612FNG 是东芝半导体公司生产的一款直流电机驱动器件，它具有大电流 MOSFET-H 桥结构，双通道电路输出，可同时驱动 2 个电机。TB6612FNG 每通道输出最高 1.2 A 的连续驱动电流，启动峰值电流达 2A/3.2 A(连续脉冲/单脉冲);4 种电机控制模式：正转/反转/制动/停止;PWM 支持频率高达 100 kHz;待机状态;片内低压检测电路与热停机保护电路;工作温度：-20~85℃;SSOP24 小型贴片封装。

对比上述方案，L298N 具有更低廉的价格并可以达到与 TB6612FNG 相同的功能，所以本次项目使用 L298N 作为电机驱动器。

3.2 硬件电路设计

3.2.1 电源电路设计

由于本次电路设计需要考虑 L298N 电机驱动器的 12V 供电问题所以在电源设计上采用多级降压的设计，即两个降压芯片分别对输入电压进行降压处理，以实现宽电压，电流输入并不影响单片机所需工作电压的情况下控制 L298N。

1. TPS563200:

TPS563200 采用 6 引脚 SOT-23 封装的简单易用型 2A 和 3A 同步降压转换器。此器件被优化为使用尽可能少的外部组件即可运行，并且可以实现

低待机电流。输入电压范围：4.5V 至 17V，输出电压范围：0.76V 至 7V。
TPS563200 电路原理图如图 3-1 所示。

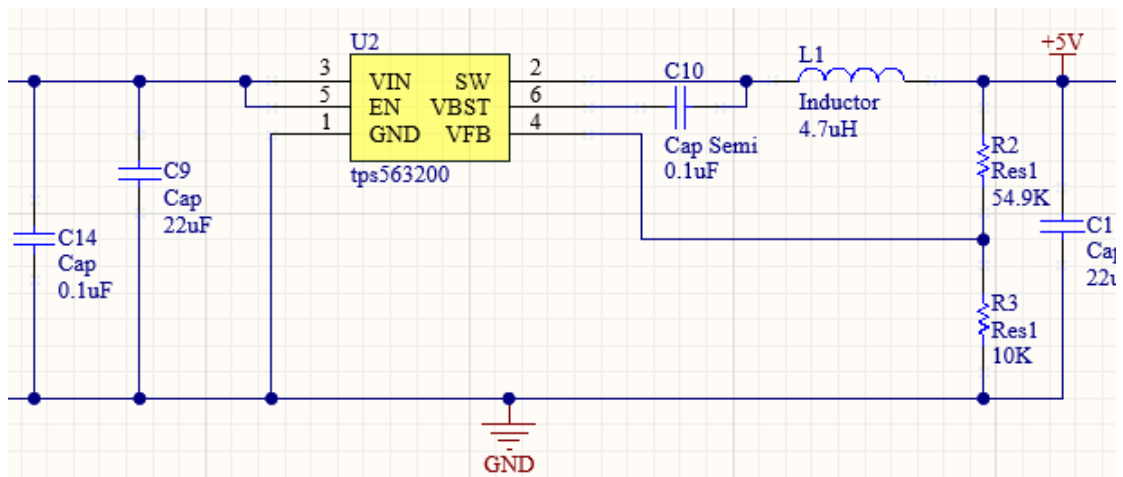


图 3-1 TPS563200 原理图

2. ASM1117:

AMS1117 系列稳压器有可调版与多种固定电压版，设计用于提供 1A 输出电流且工作压差可低至 1V。在最大输出电流时，AMS1117 器件的最小压差保证不超过 1.3V，并随负载电流的减小而逐渐降低。AMS1117 的片上微调把基准电压调整到 1.5% 的误差以内，而且电流限制也得到了调整，以尽量减少因稳压器和电源电路超载而造成的压力。AMS1117 器件引脚上兼容其他三端 SCSII 稳压器，提供适用贴片安装的 SOT-223，8 引脚 SOIC，和 TO-252(DPAK)塑料封装。AMS1117 参数 AMS1117 基本参数 输出电流 (A) 1 输出电压：1.2V 1.5V 1.8V 2.5V 2.85V 3.3V 5.0V。ASM1117 电路原理图如图 3-2 所示

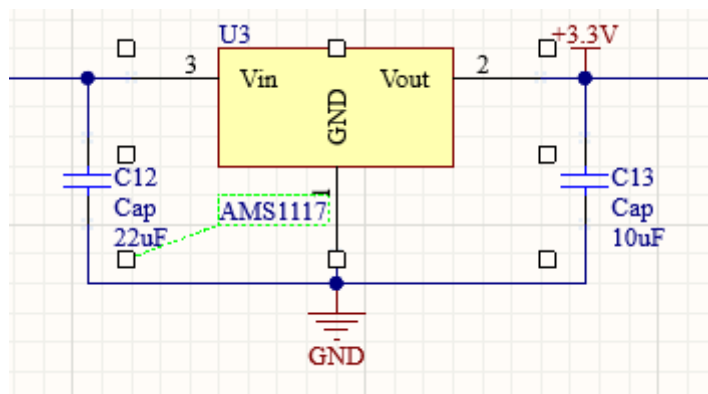


图 3-2 ASM1117 原理图

3.2.2 主控芯片电路设计

STM32F103 需工作在 3.3V 电压下，故我们需要从 ASM1117 的输出端引出 3.3V 电压以供给主控芯片，并需设计其下载电路与复位电路设计电路如图 3-3 所示。

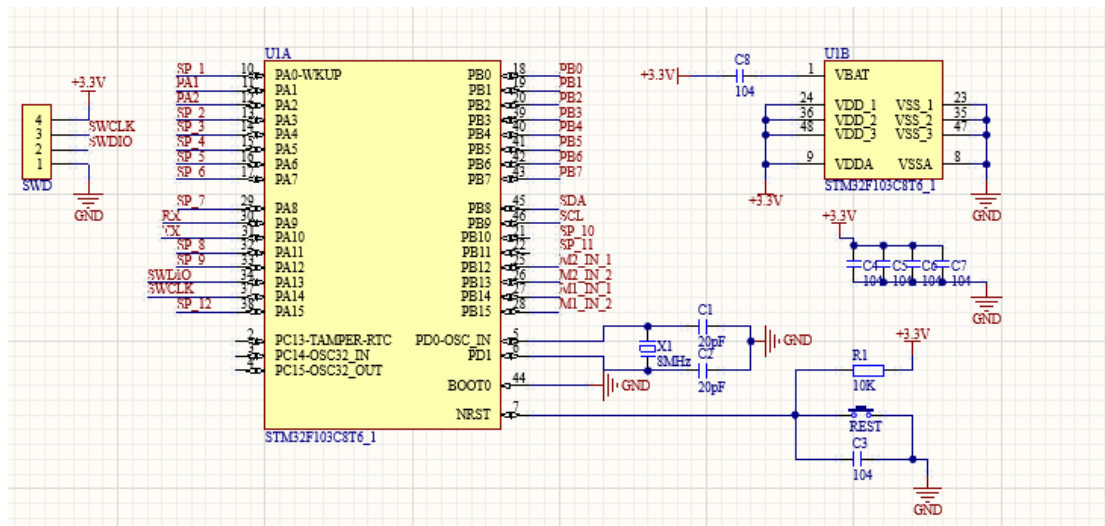


图 3-3 STM32F103 电路图

3.2.3 串口通信电路设计

本次项目设计以串口通信来模拟对钱币面额的识别，故我们选用 TTL 电平的 CH340 通信芯片以用来和串口助手进行通信，电路设计如图 3-4 所示。

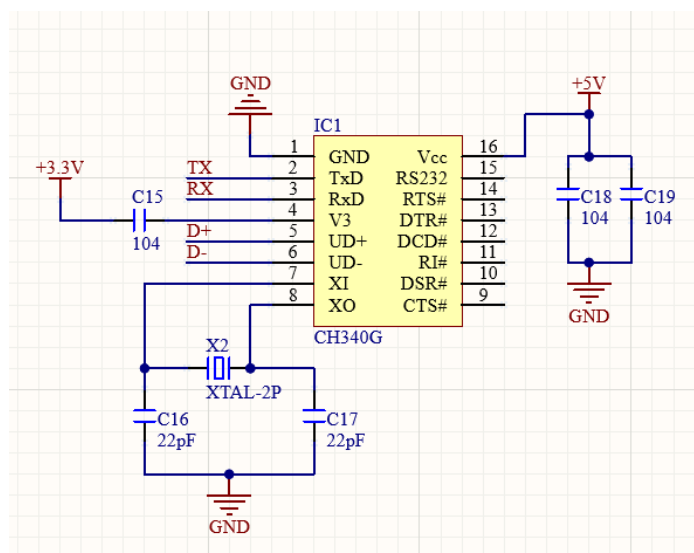


图 3-4 CH340 电路图

3.2.4 矩阵键盘电路设计

矩阵键盘本质是使用 8 个 io 口来进行 16 个按键的控制读取，可以减小 io 口的使用，用 4 条 I/O 线作为行线，4 条 I/O 线作为列线组成的键盘。在行线和列线的每个交叉点上，设置一个按键。而这样的按键中按键的个数是 4×4 个。这样的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。节约单片机的资源，其本质和独立按键类似，就是进行逐行扫描和逐列扫描，然后判断是第几行第几列个按键，进而进行整体按键值得确定，电路设计如图 3-5 所示。

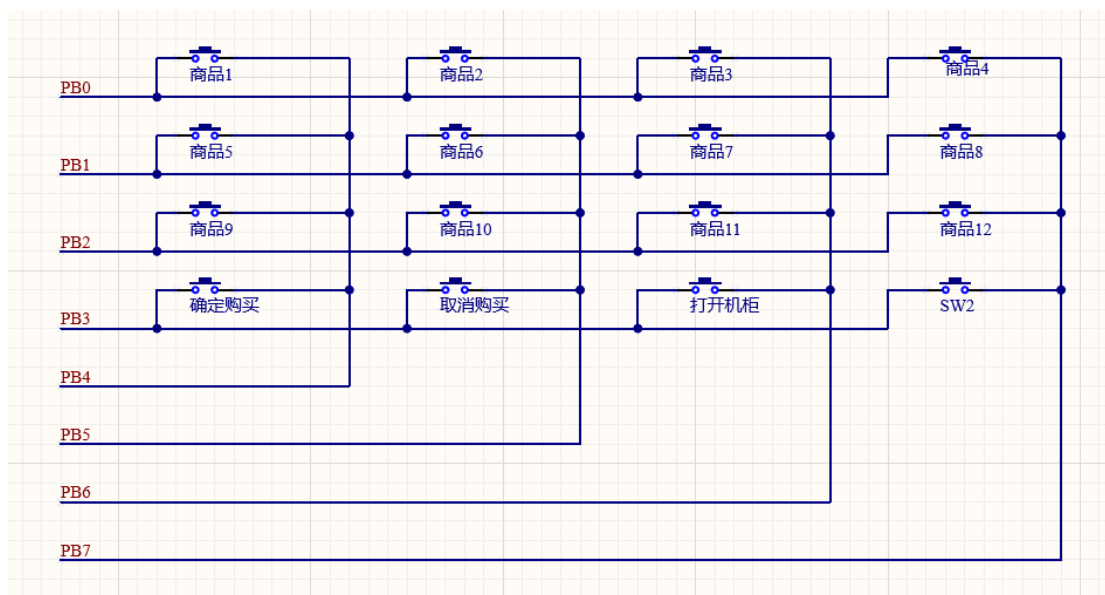


图 3-5 矩阵键盘电路图

3.2.5 整体电路与 PCB 设计图

本次项目电路图与 PCB 设计均由 Altium Designer 软件设计完成，项目整体硬件电路图与 PCB 图 3D 预览图如图 3-6，图 3-7，图 3-8 所示。

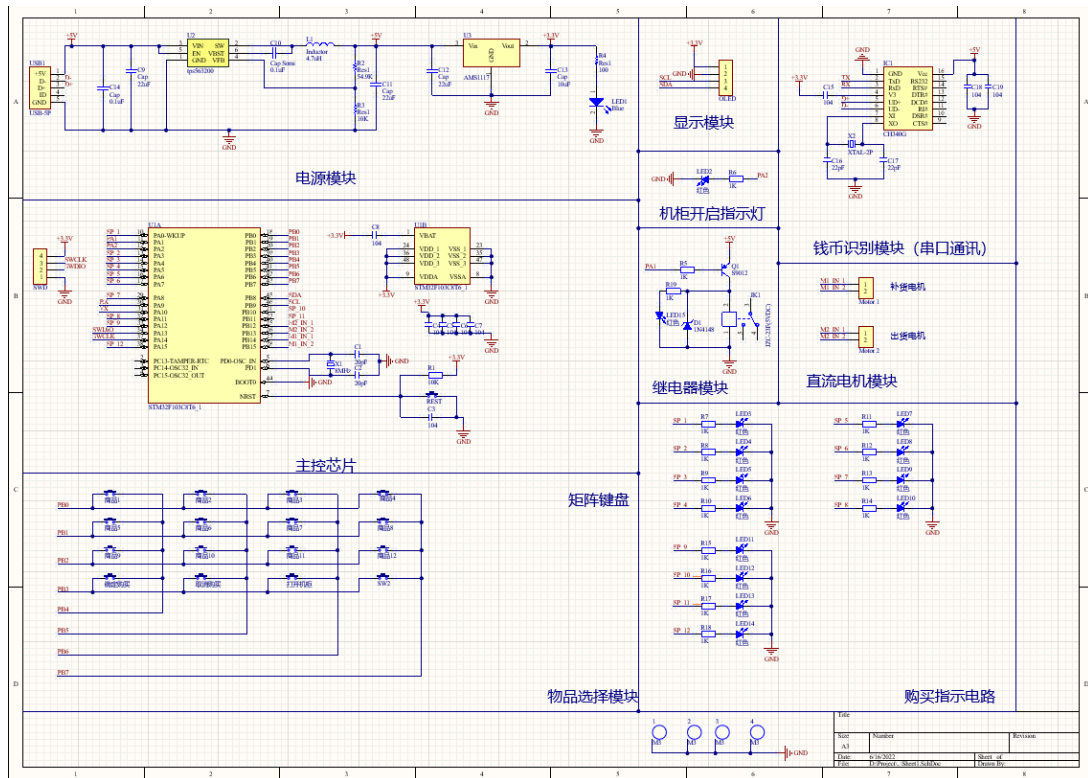


图 3-6 整体电路原理图

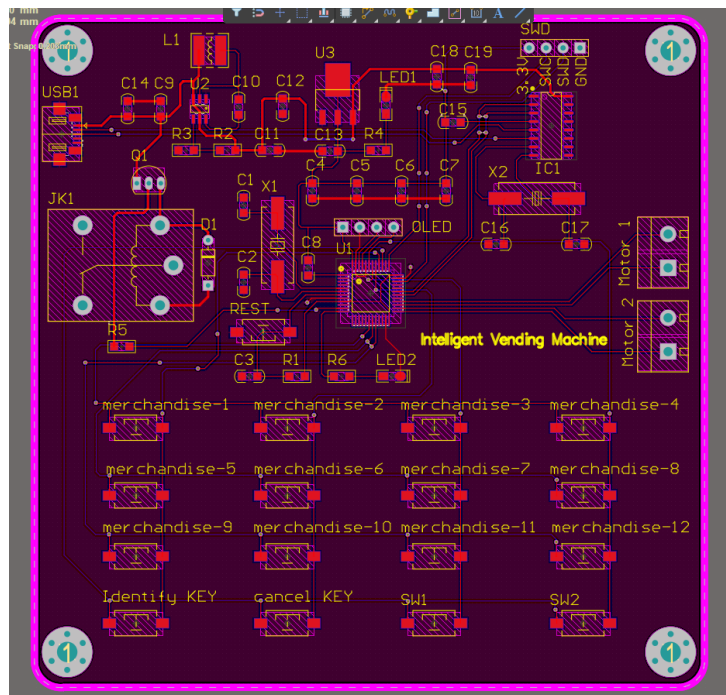


图 3-7 PCB 设计图

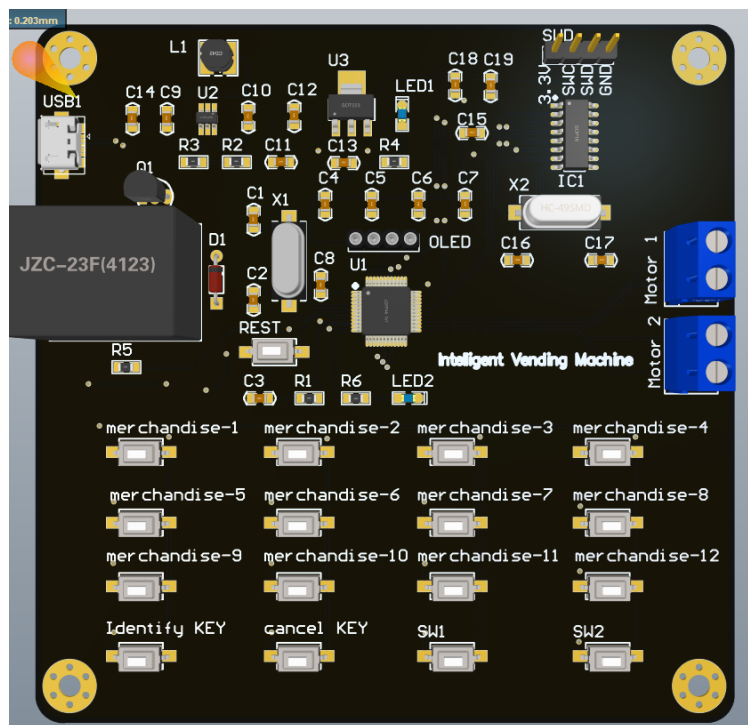


图 3-8 PCB3D 预览图

4.软件程序设计

4.1 程序设计思路

本次项目程序设计将使用 MDK 软件进行编写，并使用 ST 官方的 HAL 程序库进行辅助进行。程序整体将使用状态机的方式实现，OLED 模块将使用 I2C 的通信方式与单片机通信。

4.1.2.HAL 库介绍

HAL 是 Hardware Abstraction Layer 的缩写，中文名称是：硬件抽象层，HAL 库工程一般使用 Cube 软件来生成工程。HAL 库是 ST 公司为 STM32 的 MCU 最新推出的抽象层嵌入式软件，更方便的实现跨 STM32 产品的最大可移植性。优势就是不需要开发工程师再关注所用 MCU 型号，只需要专注所以要的功能软件开发工作。而且是未来主推的方向，正在不断的推出更新。HAL 库的推出的同时，也加入了很多第三方的中间件，有 RTOS，USB，TCP/IP 和图形等等。和标准库对比起来，STM32 的 HAL 库更加的抽象，ST 最终的目的是

要实现在 STM32 系列 MCU 之间无缝移植，甚至在其他 MCU 也能实现快速移植。

4.1.1 状态机介绍

有限状态机，（英语：Finite-state machine, FSM），又称有限状态自动机，简称状态机，是表示有限个状态以及在这些状态之间的转移和动作等行为的数学模型。其作用主要是描述对象在它的生命周期内所经历的状态序列，以及如何响应来自外界的各种事件。在计算机科学中，有限状态机被广泛用于建模应用行为、硬件电路系统设计、软件工程，编译器、网络协议、和计算与语言的研究。有限状态机（Finite-state machine）是一个非常有用的模型，可以模拟世界上大部分事物。它是一个数学模型。是一个抽象机器，在任何时候都可以处于有限数量的状态之一。响应某些外部输入，FSM 可以从一个状态转换到另一个状态；从一种状态到另一种状态的变化称为过渡。状态机的行为可以在现代社会中的许多设备中观察到，这些设备根据它们呈现的事件序列执行预定的一系列行为。例如自动售货机，当存放适当的硬币组合时分配产品，当车辆等待时改变顺序的交通灯等。

4.1.2 I2C 通信介绍

随着大规模集成电路技术的发展，把 CPU 和一个单独工作系统所必需的 ROM、RAM、I/O 端口、A/D、D/A 等外围电路集成在一个单片内而制成的单片机或微控制器愈来愈方便。目前，世界上许多公司生产单片机，品种很多。其中包括各种字长的 CPU，各种容量的 ROM、RAM 以及功能各异的 I/O 接口电路等等，但是，单片机的品种规格仍然有限，所以只能选用某种单片机来进行扩展。扩展的方法有两种：一种是并行总线，另一种是串行总线。由于串行总线的连线少，结构简单，往往不用专门的母板和插座而直接用导线连接各个设备。因此，采用串行线可大大简化系统的硬件设计。PHILIPS 公司早在十几年前就推出了 I2C 串行总线，利用该总线可实现多主机系统所需的裁决和高速设备同步等功能。因此，这是一种高性能的串行总线。

飞利浦电子公司日前推出新型二选一 I2C 主选择器，可以使两个 I2C 主设备中的任何一个与共享资源连接，广泛适用于从 MP3 播放器到服务器等计算、通信和网络应用领域，从而使制造商和终端用户从中获益。PCA9541 可以使两个 I2C 主设备在互不连接的情况下与同一个从设备相连接，从而简化了设计的复杂性。此外，新产品以单器件替代了 I2C 多个主设备应用中的多个芯片，有

效节省了系统成本。

I2C 串行总线一般有两根信号线，一根是双向的数据线 SDA，另一根是时钟线 SCL。所有接到 I2C 总线设备上的串行数据 SDA 都接到总线的 SDA 上，各设备的时钟线 SCL 接到总线的 SCL 上。为了避免总线信号的混乱，要求各设备连接到总线的输出端时必须是漏极开路（OD）输出或集电极开路（OC）输出。设备上的串行数据线 SDA 接口电路应该是双向的，输出电路用于向总线上发送数据，输入电路用于接收总线上的数据。而串行时钟线也应是双向的，作为控制总线数据传送的主机，一方面要通过 SCL 输出电路发送时钟信号，另一方面还要检测总线上的 SCL 电平，以决定什么时候发送下一个时钟脉冲电平；作为接受主机命令的从机，要按总线上的 SCL 信号发出或接收 SDA 上的信号，也可以向 SCL 线发出低电平信号以延长总线时钟信号周期。总线空闲时，因各设备都是开漏输出，上拉电阻 R_p 使 SDA 和 SCL 线都保持高电平。任一设备输出的低电平都将使相应的总线信号线变低，也就是说：各设备的 SDA 是“与”关系，SCL 也是“与”关系。总线对设备接口电路的制造工艺和电平都没有特殊的要求（NMOS、CMOS 都可以兼容）。在 I2C 总线上的数据传送率可高达每秒十万位，高速方式时在每秒四十万位以上。另外，总线上允许连接的设备数以其电容量不超过 400pF 为限。总线的运行（数据传输）由主机控制。所谓主机是指启动数据的传送（发出启动信号）、发出时钟信号以及传送结束时发出停止信号的设备，通常主机都是微处理器。被主机寻访的设备称为从机。为了进行通讯，每个接到 I2C 总线的设备都有一个唯一的地址，以便于主机寻访。主机和从机的数据传送，可以由主机发送数据到从机，也可以由从机发到主机。凡是发送数据到总线的设备称为发送器，从总线上接收数据的设备被称为接受器。

I2C 总线上允许连接多个微处理器以及各种外围设备，如存储器、LED 及 LCD 驱动器、A/D 及 D/A 转换器等。为了保证数据可靠地传送，任一时刻总线只能由某一台主机控制，各微处理器应该在总线空闲时发送启动数据，为了妥善解决多台微处理器同时发送启动数据的传送（总线控制权）冲突，以及决定由哪一台微处理器控制总线的问题，I2C 总线允许连接不同传送速率的设备。多台设备之间时钟信号的同步过程称为同步化。

4.2 编程软件介绍

4.2.1 STM32CubeMX 介绍

STM32CubeMX 是 ST 官方出的一款针对 ST 的 MCU/MPU 跨平台的图形化工具，支持在 Linux、MacOS、Window 系统下开发，支持 ST 的全系列产品目前包括：STM32L0，STM32L1，STM32L4，STM32L5，STM32F0，STM32F1，STM32F2，STM32F3，STM32F4，STM32F7，STM32G0，STM32G4，STM32H7，STM32WB，STM32WL，STM32MP1，其对接的底层接口是 HAL 库，STM32CubeMx 除了集成 MCU/MPU 的硬件抽象层，另外还集成了像 RTOS，文件系统，USB，网络，显示，嵌入式 AI 等中间件，这样开发者就能够很轻松的完成 MCU/MPU 的底层驱动的配置，留出更多精力开发上层功能逻辑，能够更进一步提高了嵌入式开发效率。软件界面如图 4-1 所示。

STM32CubeMX 软件的特点：

- ①：集成了 ST 的每一款型号的 MCU/MPU 的可配置的图形界面，能够自动提示
- ②：IO 冲突并且对于复用 IO 可自动分配；
- ③：具有动态验证的时钟树；
- ④：能够很方便的使用所集成的中间件；
- ⑤：能够估算 MCU/MPU 在不同主频运行下的功耗；
- ⑥：能够输出不同编译器的工程，比如能够直接生成 MDK、EWARM、STM32CubeIDE、MakeFile 等工程；



图 4-1 STM32CubeMX 软件界面

4.2.2 KEIL 软件介绍

Keil C51 是美国 Keil Software 公司出品的 51 系列兼容单片机 C 语言软件开发系统，与汇编相比，C 语言在功能上、结构性、可读性、可维护性上有明显的优势，因而易学易用。Keil 提供了包括 C 编译器、宏汇编、链接器、库管理和一个功能强大的仿真调试器等在内的完整开发方案，通过一个集成开发环境（ μ Vision）将这些部分组合在一起。运行 Keil 软件需要 WIN98、NT、WIN2000、WINXP 等操作系统。Keil 界面如图 4-2 所示。

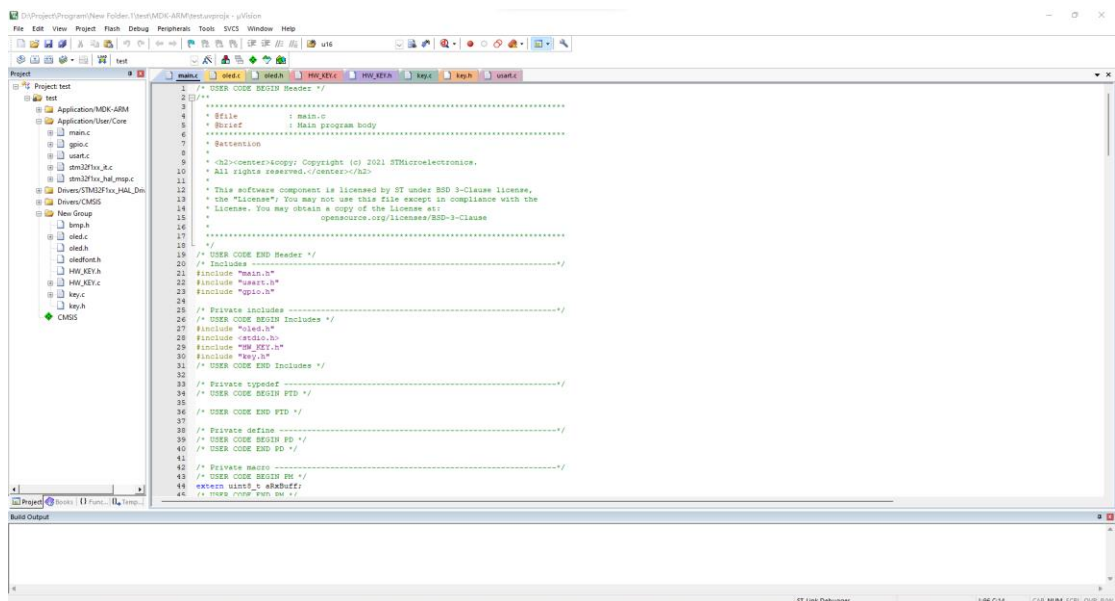


图 4-2 Keil 界面

4.3 程序功能要求

本项目程序通过矩阵按键下达指令给单片机进行判断所选择的商品，并通过 OLED 显示屏与物品选择指示灯来提示用户所选择的商品，商品数量与总金额，当用户按下确定购买下，系统会等待付款，等待付款期间可随时取消本次购物，付款以串口的方式模拟付款，将十六进制码通过上位机串口助手发送给单片机，当单片机收到付款指令后会判断是否满足消费金额，若付款小于消费金额则系统继续等待付款，若付款金额大于消费金额，则系统会自动找零，并在屏幕上显示出找零金额，同时，继电器吸合代表找零，两个电机分别转动代表出货与补货。系统运行流程图如图 4-3 所示。

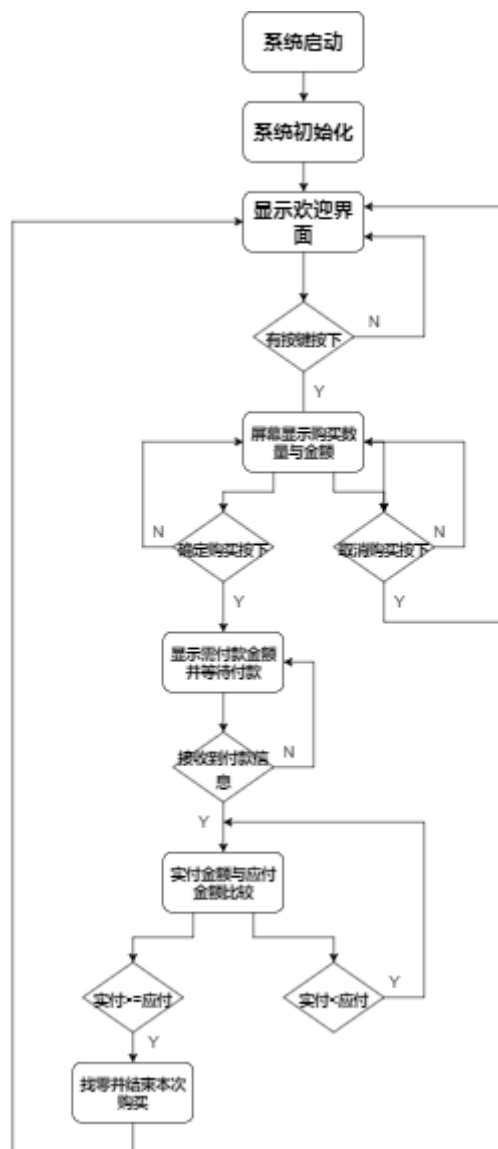


图 4-3 系统运行流程图

4.4 矩阵键盘程序编写

4.4.1 矩阵键盘程序流程

矩阵键盘为本次设计的主要输入端口，一共由十六个按键组成，矩阵按键分为四行四列，每一个按键具有一个独立的功能，本次矩阵按键程序的设计运用了状态机的方式编写，即按下一个按键进入一种状态，矩阵按键流程图如图 4-4 所示。



图 4-4 矩阵键盘流程图

4.4.2 矩阵键盘程序

① 行列扫描程序：

```

static unsigned char MB_KEY_RowScan(void)
{
    // 读行扫描状态
    Key_Row[0] = HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_0)<<3;
    Key_Row[0] = Key_Row[0] | (HAL_GPIO_ReadPin(GPIOB,
    GPIO_PIN_1)<<2);
    Key_Row[0] = Key_Row[0] | (HAL_GPIO_ReadPin(GPIOB,
    GPIO_PIN_2)<<1);

```

```
Key_Row[0] = Key_Row[0] | (HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_3));
```

```
if(Key_Row[0] != 0x0f)
{
    // 消抖
    HAL_Delay(10);
    if(Key_Row[0] != 0x0f)
    {
        //printf("Key_Row_DATA = 0x%x\r\n",Key_Row[0]);
        switch(Key_Row[0])
        {
            case 0x07:
                return 1;
            case 0x0b:
                return 2;
            case 0x0d:
                return 3;
            case 0x0e:
                return 4;
        }
    }
    else
        return 0;
}
else
    return 0;
}
```

② 矩阵键盘位号判断:

```
static unsigned char MB_KEY_Scan(void)
```

```
{
    unsigned char Key_Num = 0;
    unsigned char Key_RowResult = 0;

    MK_COL0_LOW;
    if((Key_RowResult = MB_KEY_RowScan()) != 0)
    {
        while(MB_KEY_RowScan() != 0); // 消抖
        Key_Num = 0 + Key_RowResult;
    }
    MK_COL0_HIGH;

    MK_COL1_LOW;
    if((Key_RowResult = MB_KEY_RowScan()) != 0)
    {
        while(MB_KEY_RowScan() != 0); // 消抖
        Key_Num = 4 + Key_RowResult;
    }
    MK_COL1_HIGH;

    MK_COL2_LOW;
    if((Key_RowResult = MB_KEY_RowScan()) != 0)
    {
        while(MB_KEY_RowScan() != 0); // 消抖
        Key_Num = 8 + Key_RowResult;
    }
    MK_COL2_HIGH;

    MK_COL3_LOW;
```



```
if((Key_RowResault = MB_KEY_RowScan()) != 0)
{
    while(MB_KEY_RowScan() != 0); // 消抖
    Key_Num = 12 + Key_RowResault;
}
MK_COL3_HIGH;

return Key_Num;
}
```

③ 矩阵键盘状态机程序：

```
static void MB_KEY_Funcation(void)
{
    unsigned char Key_Code,f=0;
    Key_Code = MB_KEY_Scan();
    switch(Key_Code)
    {
        case 1:
            MK_KeyUserCode_1();
            //printf("1\r\n");
            break;

        case 2:
            MK_KeyUserCode_5();
            //printf("2\r\n");
            break;

        case 3:
            MK_KeyUserCode_9();
            //printf("3\r\n");
            break;
    }
}
```

```
case 4:
    MK_KeyUserCode_identify();
    //printf("4\r\n");
break;
```

```
case 5:
    MK_KeyUserCode_2();
    //printf("5\r\n");
break;
```

```
case 6:
    MK_KeyUserCode_6();
    //printf("6\r\n");
break;
```

```
case 7:
    MK_KeyUserCode_10();
    //printf("7\r\n");
break;
```

```
case 8:
    MK_KeyUserCode_cancel();
    //printf("8\r\n");
break;
```

```
case 9:
    MK_KeyUserCode_3();
    //printf("9\r\n");
break;
```

```
case 10:
    MK_KeyUserCode_7();
    //printf("10\r\n");
```

```
break;

case 11:
    MK_KeyUserCode_11();
    //printf("11\r\n");
break;
case 12:
    LED_open(1);
    //printf("12\r\n");
break;

case 13:
    MK_KeyUserCode_4();
    //printf("13\r\n");
break;

case 14:
    MK_KeyUserCode_8();
    //printf("14\r\n");
break;

case 15:
    MK_KeyUserCode_12();
    //printf("15\r\n");
break;

case 16:
    printf("16\r\n");
break;
}
}
```

4.4 OLED 屏幕程序编写

4.4.1 OLED 屏幕介绍

OLED，即有机发光二极管(Organic Light Emitting Diode)。OLED 由于同时具备自发光，不需背光源、对比度高、厚度薄、视角广、反应速度快、可用于挠曲性面板、使用温度范围广、构造及制程较简单等优异之特性，被认为是下一代的平面显示器新兴应用技术。LCD 都需要背光，而 OLED 不需要，因为它是自发光的。这样同样的显示 OLED 效果要来得好一些。以目前的技术，OLED 的尺寸还难以大型化，但是分辨率确可以做到很高。在此我们使用的是中景园电子的 0.96 寸 OLED 显示屏。

该屏有以下特点：

- ①：0.96 寸 OLED 有黄蓝，白，蓝三种颜色可选;其中黄蓝是屏上 1/4 部分为黄光，下 3/4 为蓝;而且是固定区域显示固定颜色，颜色和显示区域均不能修改
- ②：分辨率为 128*64。
- ③：多种接口方式; OLED 裸屏总共种接口包括: 6800、8080 两种并行接口方式、3 线或 4 线的串行 SPI 接口方式、IIC 接口方式，这五种接口是通过屏上的 BSO~BS2 来配置的。

4.4.2 OLED 屏幕程序介绍与工作流程

本次选用的 OLED 屏幕通信方式为 I2C 通信，根据屏幕手册对 I2C 地址描述根据 DC 引脚电平的不同，地址为 0111100 和 0111101 两种，通常我们设置 DC 引脚接地，所以作为 I2C 从机的七位地址为 0111100；又因为我们与驱动 IC 交互时，都是主机发送命令或者数据到 IC，也就是只有写数据，没有读数据，所以从机地址为 0x78。OLED 读数据通信模式方式如图 4-5 所示。OLED 初始化流程如图 4-6 所示。

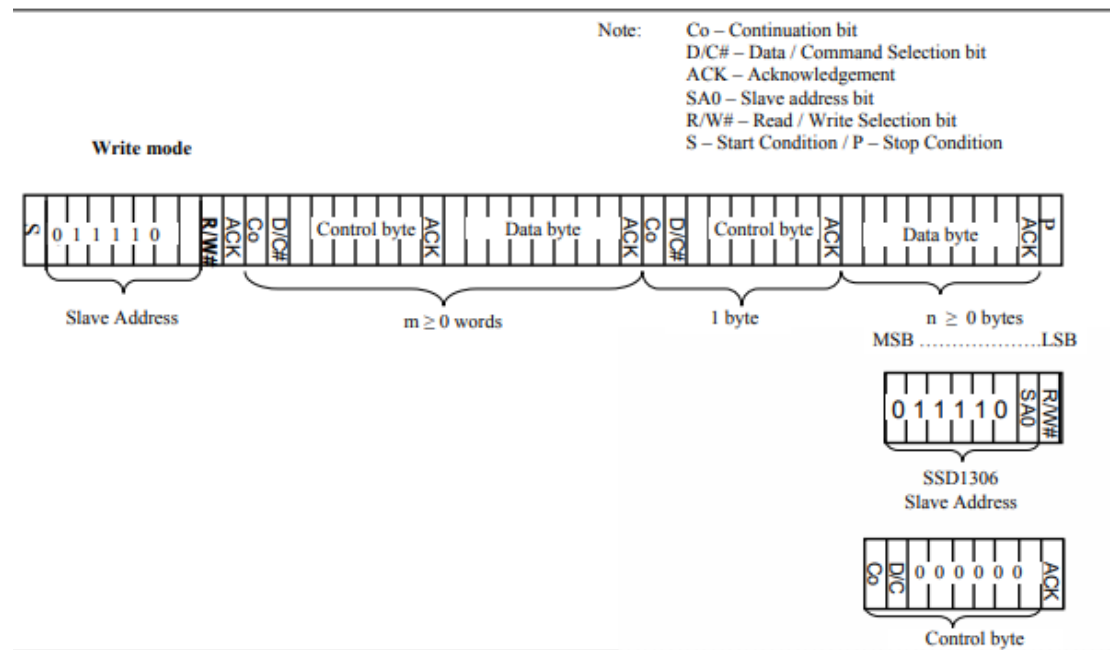


图 4-5 OLED 读写模式通信

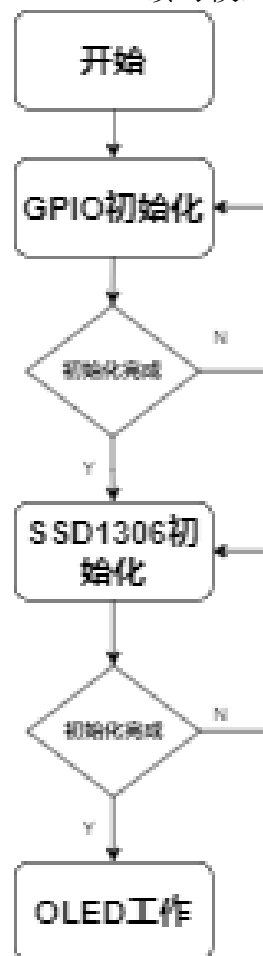


图 4-6 OLED 初始化流程图

4.4.3 OLED 程序编写

① OLED 读写程序:

//写入一个字节

```
void Send_Byte(uint8_t dat)
```

```
{
    uint8_t i;
    for(i=0;i<8;i++)
    {
        OLED_SCLK_Clr();//将时钟信号设置为低电平
        if(dat&0x80)//将 dat 的 8 位从最高位依次写入
        {
            OLED_SDIN_Set();
        }
        else
        {
            OLED_SDIN_Clr();
        }
        OLED_SCLK_Set();//将时钟信号设置为高电平
        OLED_SCLK_Clr();//将时钟信号设置为低电平
        dat<<=1;
    }
}
```

//发送一个字节

//向 SSD1306 写入一个字节。

//mode:数据/命令标志 0,表示命令;1,表示数据;

```
void OLED_WR_Byte(uint8_t dat,uint8_t mode)
```

```
{
    I2C_Start();
    Send_Byte(0x78);
    I2C_WaitAck();
    if(mode){Send_Byte(0x40);}
}
```

```

else{Send_Byte(0x00);}
    I2C_WaitAck();
    Send_Byte(dat);
    I2C_WaitAck();
    I2C_Stop();
}

```

② OLED 初始化程序:

//OLED 的初始化

```
void OLED_Init(void)
```

```

{
    OLED_WR_Byte(0xAE,OLED_CMD);/--turn off oled panel
    OLED_WR_Byte(0x00,OLED_CMD);/---set low column address
    OLED_WR_Byte(0x10,OLED_CMD);/---set high column address
        OLED_WR_Byte(0x40,OLED_CMD);/--set start line address Set Mapping
RAM Display Start Line (0x00~0x3F)
    OLED_WR_Byte(0x81,OLED_CMD);/--set contrast control register
    OLED_WR_Byte(0xCF,OLED_CMD);// Set SEG Output Current Brightness
    OLED_WR_Byte(0xA1,OLED_CMD);/--Set SEG/Column Mapping 0xa0 左
右反置 0xa1 正常
    OLED_WR_Byte(0xC8,OLED_CMD);//Set COM/Row Scan Direction 0xc0 上
下反置 0xc8 正常
    OLED_WR_Byte(0xA6,OLED_CMD);/--set normal display
    OLED_WR_Byte(0xA8,OLED_CMD);/--set multiplex ratio(1 to 64)
    OLED_WR_Byte(0x3f,OLED_CMD);/--1/64 duty
    OLED_WR_Byte(0xD3,OLED_CMD);/--set display offset Shift Mapping RAM
Counter (0x00~0x3F)
    OLED_WR_Byte(0x00,OLED_CMD);/--not offset
    OLED_WR_Byte(0xd5,OLED_CMD);/--set display clock divide ratio/oscillator
frequency
    OLED_WR_Byte(0x80,OLED_CMD);/--set divide ratio, Set Clock as 100
Frames/Sec
    OLED_WR_Byte(0xD9,OLED_CMD);/--set pre-charge period

```

```

OLED_WR_Byte(0xF1,OLED_CMD);//Set Pre-Charge as 15 Clocks &
Discharge as 1 Clock
OLED_WR_Byte(0xDA,OLED_CMD);//--set com pins hardware configuration
OLED_WR_Byte(0x12,OLED_CMD);
OLED_WR_Byte(0xDB,OLED_CMD);//--set vcomh
OLED_WR_Byte(0x40,OLED_CMD);//Set VCOM Deselect Level
OLED_WR_Byte(0x20,OLED_CMD);//-Set Page Addressing Mode
(0x00/0x01/0x02)
OLED_WR_Byte(0x02,OLED_CMD);//
OLED_WR_Byte(0x8D,OLED_CMD);//--set Charge Pump enable/disable
OLED_WR_Byte(0x14,OLED_CMD);//--set(0x10) disable
OLED_WR_Byte(0xA4,OLED_CMD);// Disable Entire Display On (0xa4/0xa5)
OLED_WR_Byte(0xA6,OLED_CMD);// Disable Inverse Display On (0xa6/a7)
OLED_WR_Byte(0xAF,OLED_CMD);
OLED_Clear();
}

```

4.5 串口通信程序

4.5.1 串口通信工作流程

本次项目采用串口通信的方式，使用 STM32 自带串口，在全双工模式下。串口通信(Serial Communication)，是指外设和计算机间，通过数据信号线、地线、控制线等，按位进行传输数据的一种通讯方式。这种通信方式使用的数据线少，在远距离通信中可以节约通信成本，但其传输速度比并行传输低。

由于 CPU 与接口之间按并行方式传输，接口与外设之间按串行方式传输，因此，在串行接口中，必须要有 " 接收移位寄存器 "（串→并）和 " 发送移位寄存器 "（并→串）。在数据输入过程中，数据 1 位 1 位地从外设进入接口的 " 接收移位寄存器 "，当 " 接收移位寄存器 " 中已接收完 1 个字符的各位后，数据就从 " 接收移位寄存器 " 进入 " 数据输入寄存器 "。CPU 从 " 数据输入寄存器 " 中读取接收到的字符。（并行读取，即 D7~D0 同时被读至累加器中）。" 接收移位寄存器 " 的移位速度由 " 接收时钟 " 确定。在数据输出过程中，CPU 把要输出的字符（并行地）送入 " 数据输出寄存器 "，" 数据输出寄存器 " 的内容传输到 " 发送移位寄存器 "，然后由 " 发送移位寄存器 " 移位，把数据 1 位 1 位地送到外

设."发送移位寄存器"的移位速度由"发送时钟"确定.接口中的"控制寄存器"用来容纳 CPU 送给此接口的各种控制信息,这些控制信息决定接口的工作方式."状态寄存器"的各位称为"状态位",每一个状态位都可以用来指示数据传输过程中的状态或某种错误.例如,用状态寄存器的 D5 位为"1"表示"数据输出寄存器"空,用 D0 位表示"数据输入寄存器满",用 D2 位表示"奇偶检验错"等,能够完成上述"串并"转换功能的电路,通常称为"通用异步收发器"(UART: Universal Asynchronous Receiver and Transmitter)。本次串口工作流程如图 4-7 所示。



图 4-7 串口通信流程图

4.5.2 串口通信编写

由于本次项目使用的是 ST 官方工具 STM32CubeMX 所生成配置代码,故无需二次编写串口初始化程序,这里我们直接编写串口中断处理程序。编写串口处理函数前需先定义串口通信协议,协议如表 4-1 所示。

表 4-1

串口通信协议		
包头（1byte）	所示含义	备注
0x01	代表支付 一元	
0x05	代表支付五元	
0x0A	代表支付十元	
0x14	代表支付二十元	
0x32	代表支付五十元	
0x64	代表支付一百元	
包尾（1byte）		
0x0D	包尾结束位	
0x0A	包尾结束位	

串口处理程序如下所示：

```

/*
  串口终端程序
*/
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
    int i=0;
    for(i=0;i<3;i++)
    {
        Rx_Data[i] = aRxBuff; // 0x64 0x0d 0x0a
    }
    switch(Rx_Data[0]) //判断串口接收到的数据
    {
        case money_1: //1 yuan
            total_price+=1;
            break;
        case money_5: //5 yuan
            total_price+=5;
            break;
        case money_10: //10 yuan
            total_price+=10;
            break;
    }
}

```

```
        case money_20: //20 yuan
            total_price+=20;
        break;
        case money_50: //50 yuan
            total_price+=50;
        break;
        case money_100: //100 yuan
            total_price+=100;
        break;
        default:
            printf("输入错误，请重新输入\r\n");
            total_price+=0;
        break;
    }
    printf("已收款: %d\r\n",total_price);
    HAL_UART_Receive_IT(&huart1,(uint8_t *)&aRxBuff, 1);
}
```

4.6 主程序流程编写

在主要处理函数中我们要实现对自动售货机的整体功能实现，即上文所叙述的主要功能：需在接收到购买指令后对记录下所购买的商品并亮起相应的指示灯，需要具备对零钱找零功能，并将当前售货机所处状态实时通过 OLED 屏幕反馈给用户实现人机交互，具体流程如图 4-7 所示。

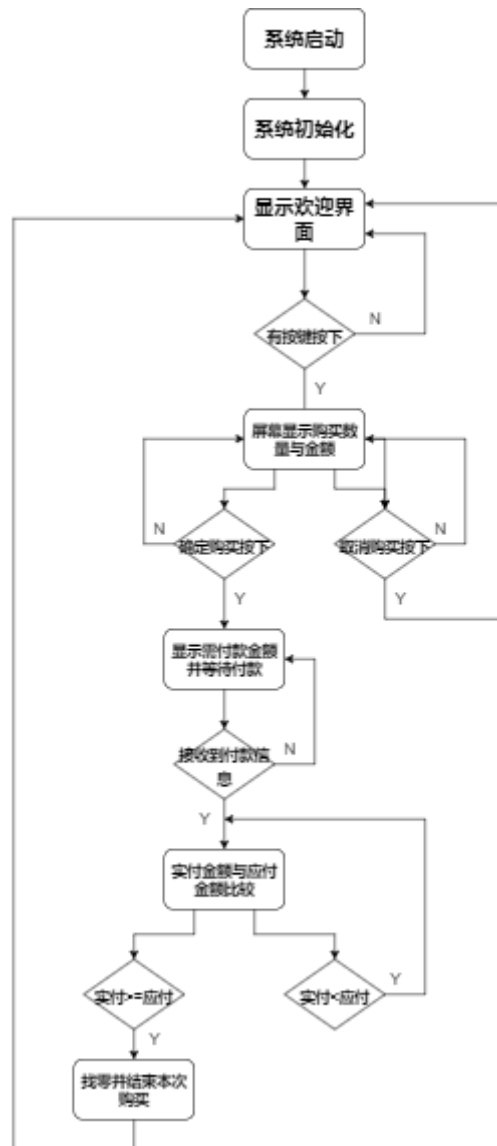


图 4-7 自动贩卖机工作流程

4.6.1 主要逻辑程序编写

```

/*
    主逻辑函数
*/
void test(void)
{

    if(total_price!=0&&price!=0) //付款金额与消费金额均不为 0 时进入循环
    {

```

```

collect_money(); //显示找零界面
if(total_price>=price) //付款金额大于等于消费金额
{
    collect_money(); //显示找零界面
    HAL_GPIO_WritePin(GPIOA ,GPIO_PIN_1 ,GPIO_PIN_RESET ); //
继电器吸合
    HAL_Delay(2000); //延时 2s
    HAL_GPIO_WritePin(GPIOA ,GPIO_PIN_1 ,GPIO_PIN_SET ); //继电
器断开
    HAL_Delay(50); //延时 50ms
    Motor_1(); //电机一启动，表示送出货物
    HAL_Delay(30000); // 延时 30s
    Motor_2(); //电机二启动，表示补充货物
    HAL_Delay(50); //延时 50ms
    change_money=total_price-price; //找零金额=付款金额-消费金额

    OLED_Clear(); //OLED 清屏函数
    OLED_ShowChinese(31,16,25,16);
    OLED_ShowChinese(47,16,26,16);
    OLED_ShowNum(63,16,change_money,2,16); //显示找零金额
    OLED_ShowChinese(79,16,16,16);
    OLED_Refresh();

    HAL_Delay(1000); //延时 1s 显示
    total_price=0; //清空付款金额
    price=0; //清空消费金额
    ZS_LED_OFF();

    /* 智能售货机
        请选择物品*/
    OLED_Clear();
    OLED_ShowChinese(25,0,0,16);
    OLED_ShowChinese(41,0,1,16);

```

```
OLED_ShowChinese(57,0,2,16);
OLED_ShowChinese(73,0,3,16);
OLED_ShowChinese(89,0,4,16);
OLED_ShowChinese(25,25,9,16);
OLED_ShowChinese(41,25,10,16);
OLED_ShowChinese(57,25,11,16);
OLED_ShowChinese(73,25,12,16);
OLED_ShowChinese(89,25,13,16);

OLED_Refresh();
}
}
}
```

5 电路与程序功能仿真

本次项目使用电子电路仿真软件 Proteus 与虚拟串口软件 Configure Virtual Serial Port Driver 完成仿真实验。

5.1 软件介绍

5.1.1 仿真软件 Proteus 介绍

Proteus 软件是英国 Lab Center 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。它是比较好的仿真单片机及外围器件的工具。虽然国内推广刚起步，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 是英国著名的 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，其处理器模型支持 8051、HC11、PIC10/12/16/18/24/30/DSPIC33、AVR、ARM、8086 和 MSP430 等，2010 年又增加了 Cortex 和 DSP 系列处理器，并持续增加其他系列处理器模型。在编译方

面，它也支持 IAR、Keil 和 MATLAB 等多种编译器。

1. Proteus 软件具有其它 EDA 工具软件的功能。这些功能是：

- ①：原理布图
- ②：PCB 自动或人工布线
- ③：SPICE 电路仿真

2. 革命性的特点：

① 互动的电路仿真：

用户甚至可以实时采用诸如 RAM，ROM，键盘，马达，LED，LCD，AD/DA，部分 SPI 器件，部分 IIC 器件。

② 仿真处理器及其外围电路：

可以仿真 51 系列、AVR、PIC、ARM、等常用主流单片机。还可以直接在基于原理图的虚拟原型上编程，再配合显示及输出，能看到运行后输入输出的效果。配合系统配置的虚拟逻辑分析仪、示波器等，Proteus 建立了完备的电子设计开发环境。

5.1.2 虚拟串口软件 Configure Virtual Serial Port Driver 介绍

Virtual Serial Port Driver 是基于「虚拟串口驱动程序」功能的高效一体化软件。该程序有助于创建虚拟串口捆绑并设置自定义端口参数，这使得该解决方案非常适合各种使用场景。Virtual Serial Port Driver 为您提供以最方便、最简单的方式管理实际和虚拟 COM 端口的功能。Virtual Serial Port Driver 具有以下特点。

1. 串口分割：

Virtual Serial Port Driver 允许将物理串口（例如 COM 1）分割成多个出现在系统中的虚拟通信端口（COM2, COM3, COM4, 等），作为硬件串行接口的精确副本。此外，该软件还提供了在多个应用程序之间共享硬件串口。数个串口应用程序将能够访问相同的 COM 端口和插入到它的设备。

2. 结合 COM 端口：

该软件解决方案帮助您将多个真正的 COM 端口链接到一个虚拟串口。连接到此虚拟 COM 端口，串行应用程序将同时从所有连接的实际 COM 端口接收数据。简而言之，使用此高级功能，您将能够从多个设备重定向数据到一个应用程序。

3.创建捆绑连接:

软件提供的这一独特选项使您可以创建和管理复杂的串口。这意味着串行数据可以通过无限数量的物理和虚拟 COM 接口传入和传出。通过「IN」端的 COM 端口接收,串行数据将被重定向到捆绑的「OUT」端的每个 COM 端口,反之亦然。以上为捆绑的每一端(In/Out)设置主端口,便于管理硬件控制线的过程。

4.自动切换端口:

如果您的应用程序设计为使用有限数量的端口,则可以使用 Switcher (Virtual Serial Port Driver PRO 的特殊功能)。使用此选项,可以将多个物理端口连接到一个虚拟端口(可以选择将所有端口都命名为同一个名称),然后在所需的应用程序中创建一个到此虚拟 COM 端口的切换器。程序何时需要串口连接,切换器都会检测连接端口列表中未占用的 COM 端口,并将其提供给应用程序。这样,你的程序将自动和任何空闲的串口通信,就好像它总是相同的端口。

5.合并 COM 端口:

The new advanced feature of Virtual Serial Port Driver 的高级功能使您可以将任意数量的物理和虚拟 COM 端口添加到未分为「IN」和「OUT」端的单个串口捆绑。发送到一个端口的所有内容都将被复制到该捆绑的所有其他实际和虚拟端口。由于此功能,每个设备/应用程序都可以同时与多个设备/应用程序交换数据。

5.2 仿真图的绘制

本次项目仿真图采用模块化的方式绘制,将各个元器件分为多个模块,仿真图绘制如图 5-1 所示。

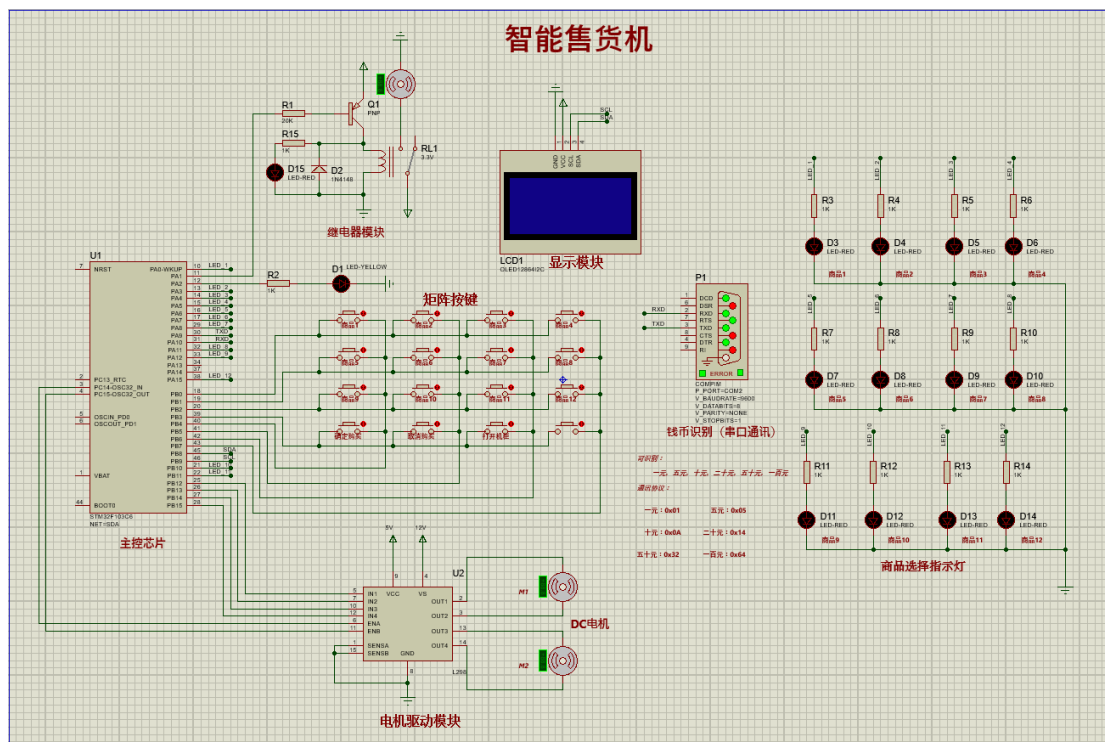


图 5-1 仿真图

5.2.1 各个主要模块仿真细节图

主控芯片如图 5-2 所示，矩阵键盘如图 5-3 所示，商品选择指示灯如图 5-4 所示，电机驱动模块如图 5-5 所示，OLED 显示模块如图 5-6 所示。

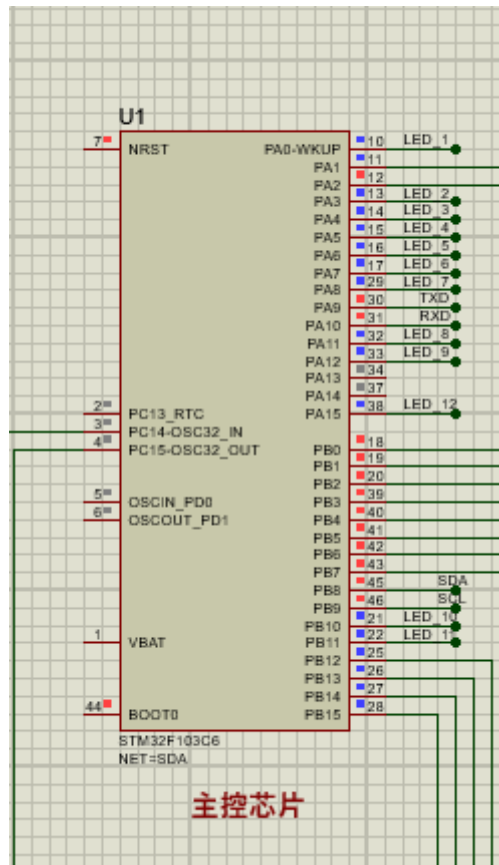


图 5-2 主控芯片仿真图

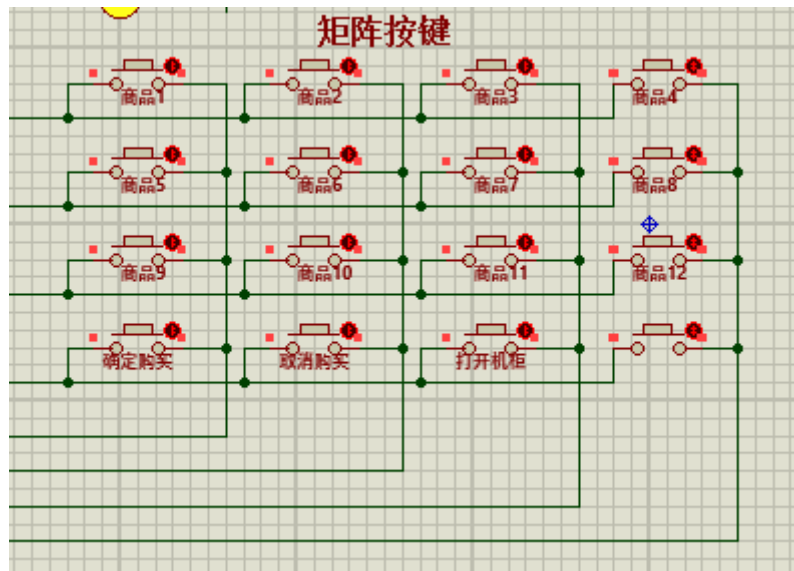


图 5-3 矩阵键盘仿真

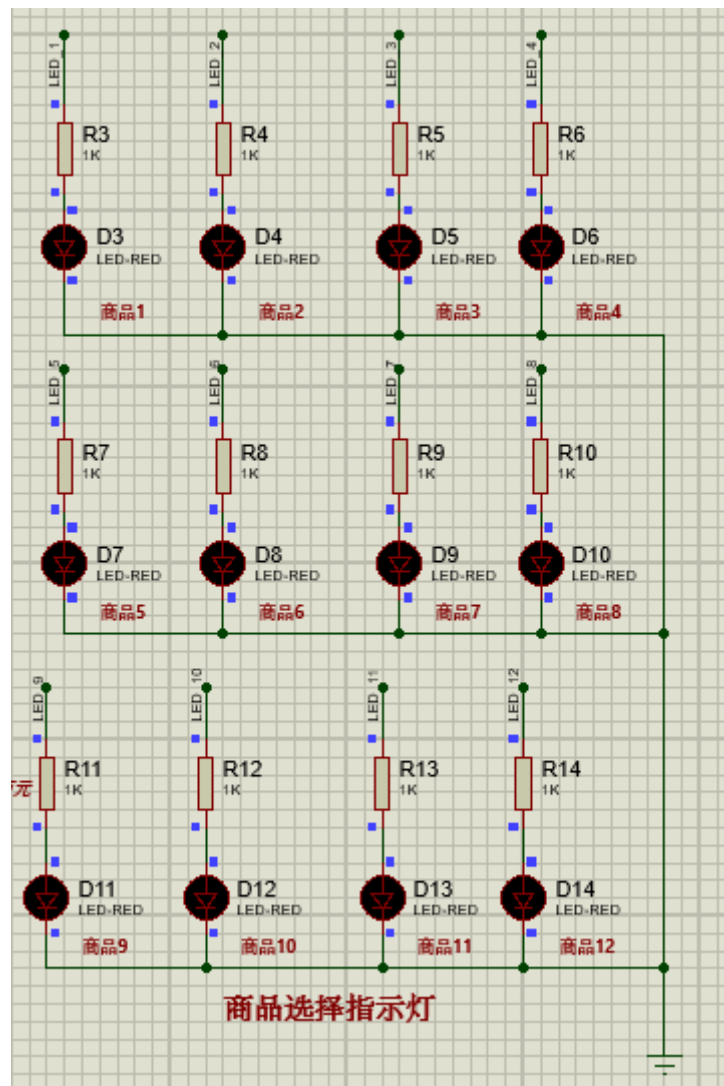


图 5-4 商品指示仿真图

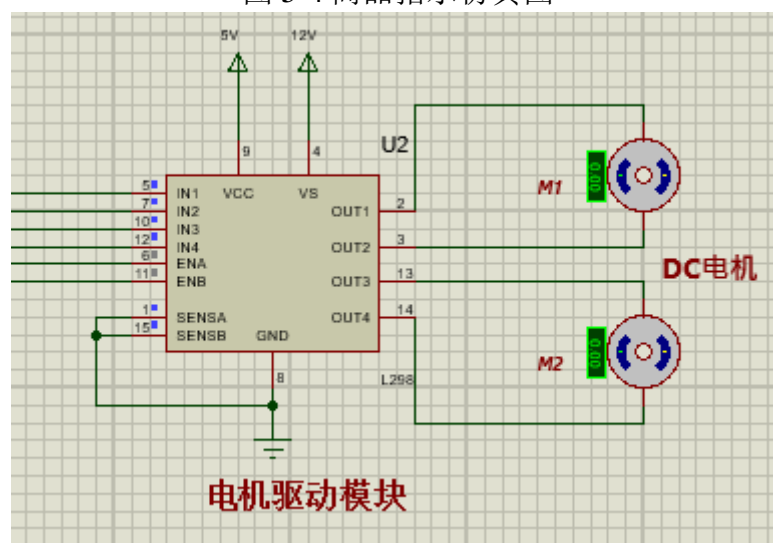


图 5-5 电机驱动仿真图

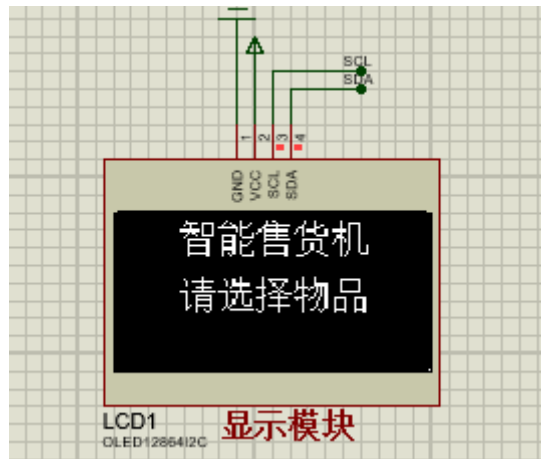


图 5-6 OLED 显示模块仿真

5.3 Virtual Serial Port Driver 虚拟串口软件设置

Virtual Serial Port Drive 软件设置起来非常的简单仅需点击软件界面上的 Add pair 便可添加一对串口 COM1 和 COM2，未添加虚拟串口时如图 5-7 所示，添加完毕后如图 5-8 所示。

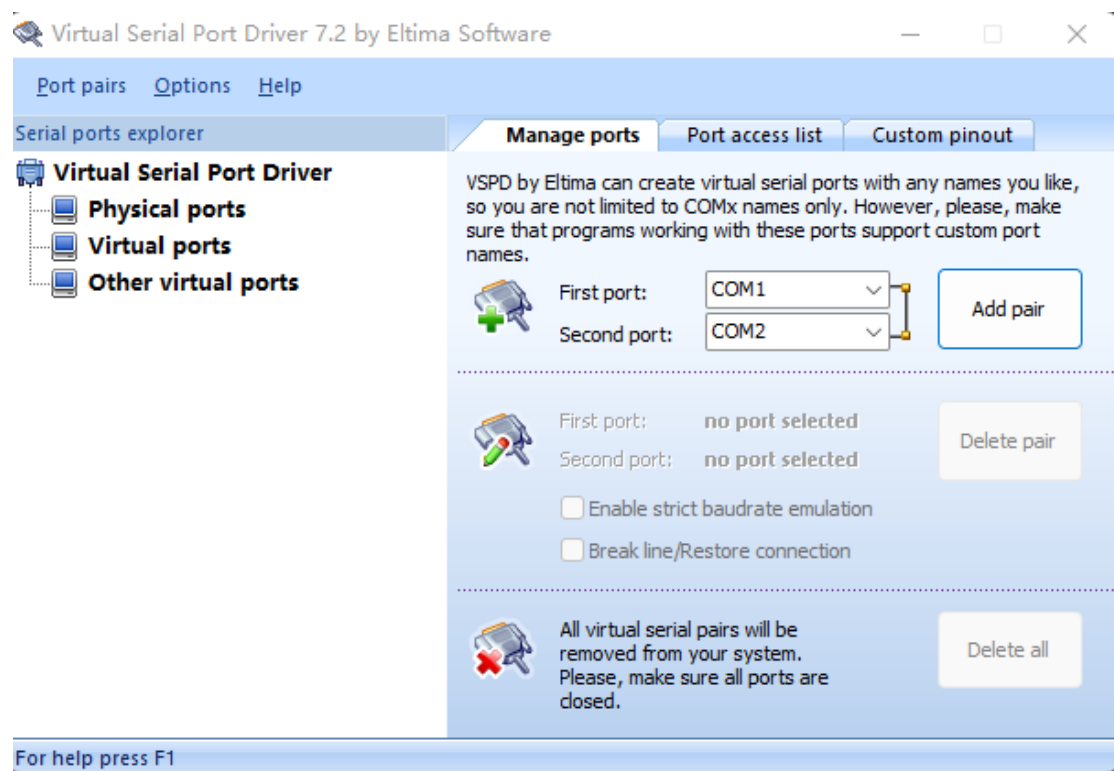


图 5-8 未添加虚拟串口

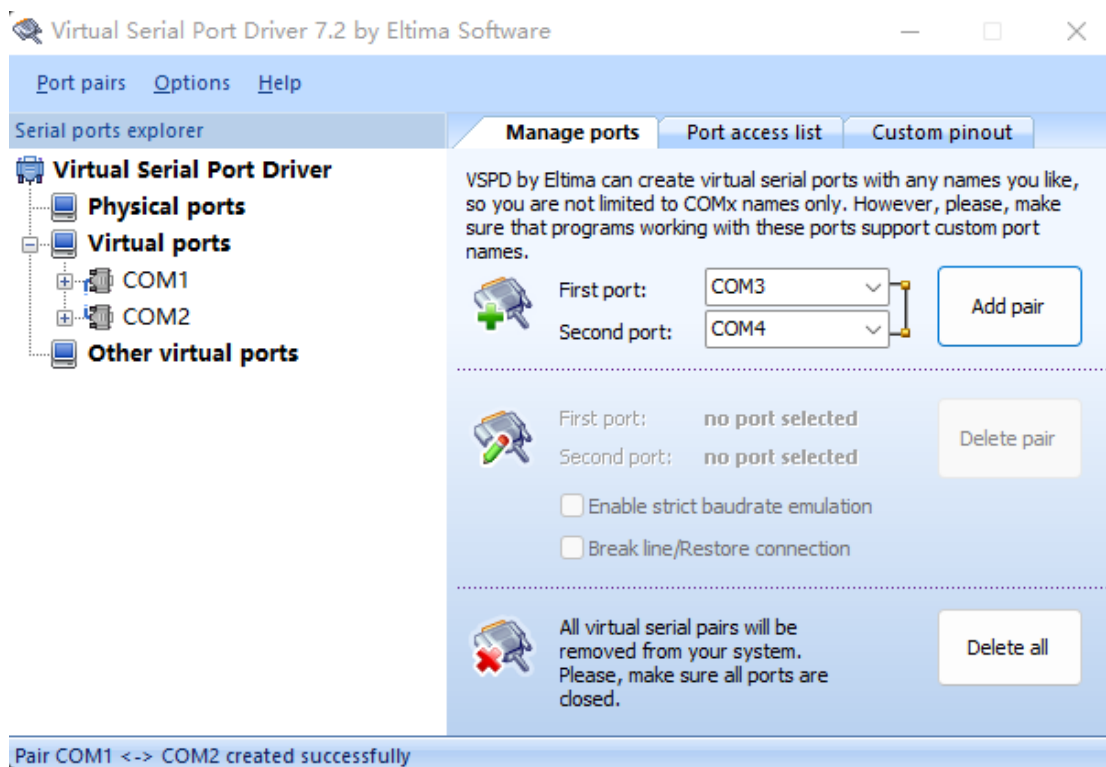


图 5-9 虚拟串口以添加

完成虚拟串口的添加设置后便可开启与 Proteus 仿真软件的串口通信了。

5.4 Proteus 仿真设置与仿真过程

5.4.1 Proteus 仿真设置

在 Proteus 仿真中我们需要为主控芯片添加已经由 Keil 软件编译好的 hex 文件，添加过程如下，双击主控芯片，会显示出主控芯片的配置窗口，如图 5-10 所示。

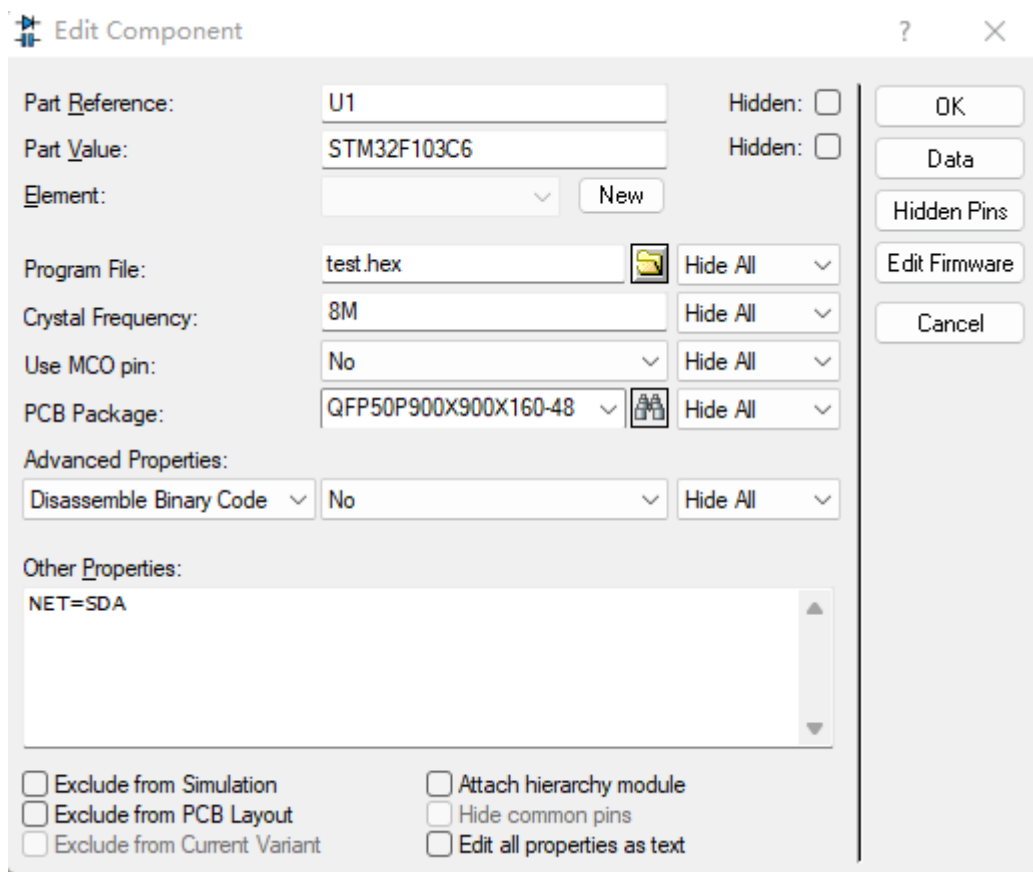


图 5-10 主控芯片配置界面

如图 5-10 中所示，在 Program File 选择框中添加以编译好的 hex 文件，并在 Crystal Frequency 选择框中设置单片机仿真主频为 8M。

配置好主控芯片后还需对仿真中的串口进行配置，双击串口元件出现串口配置界面如图 5-11 所示。

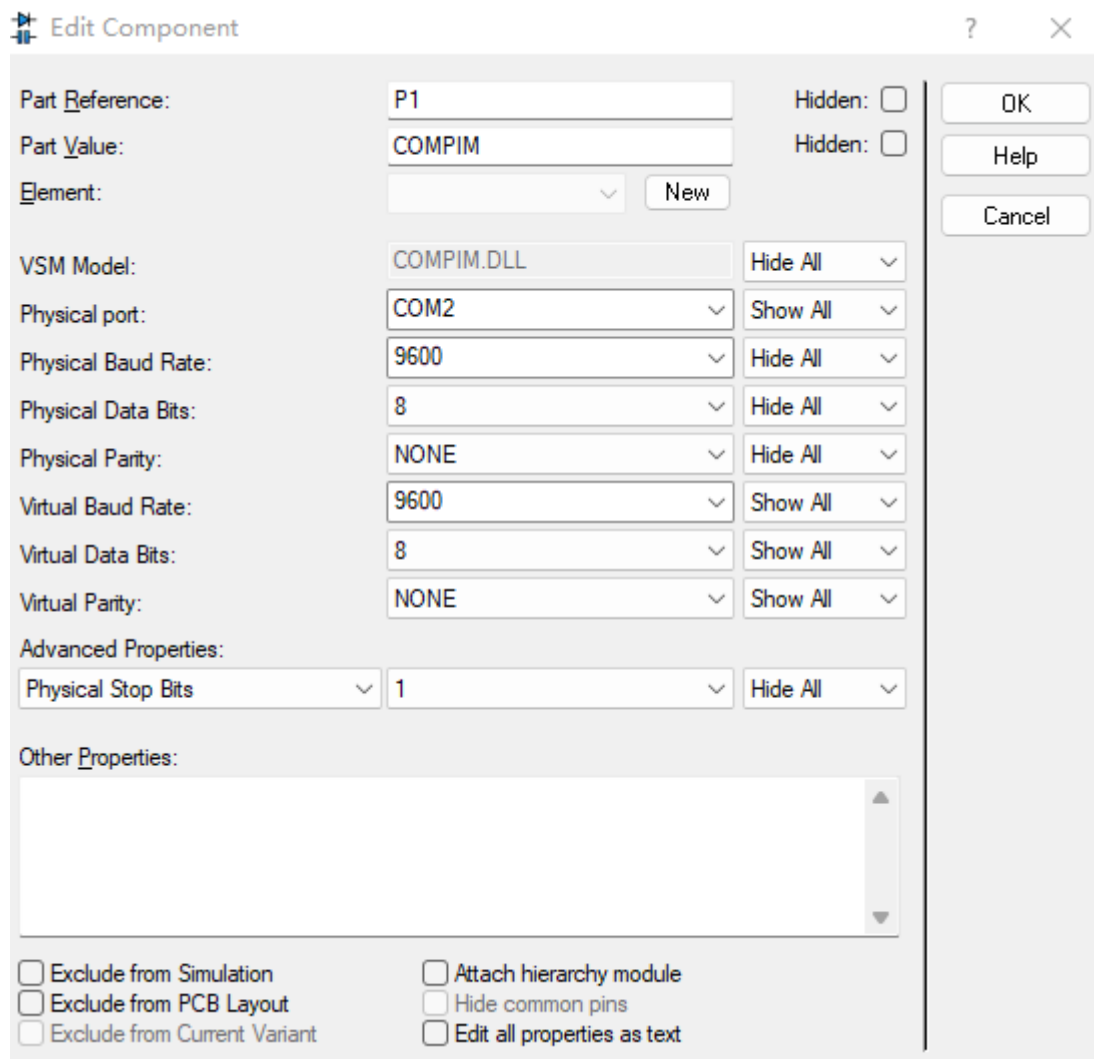


图 5-11 串口配置界面

如图 5-11 中所示，需要将串口波特率与端口号的设置为上图，点击 Physical port 选择框选择 COM2 为通信端口号，点击 Physical Baud Rate 选择框，选择 9600，将通信波特率设置为 9600。至此完成所有仿真所需的设置。

5.4.2 仿真过程

启动仿真，OLED 显示屏出现汉字表示系统成功启动，如图 5-12 所示。

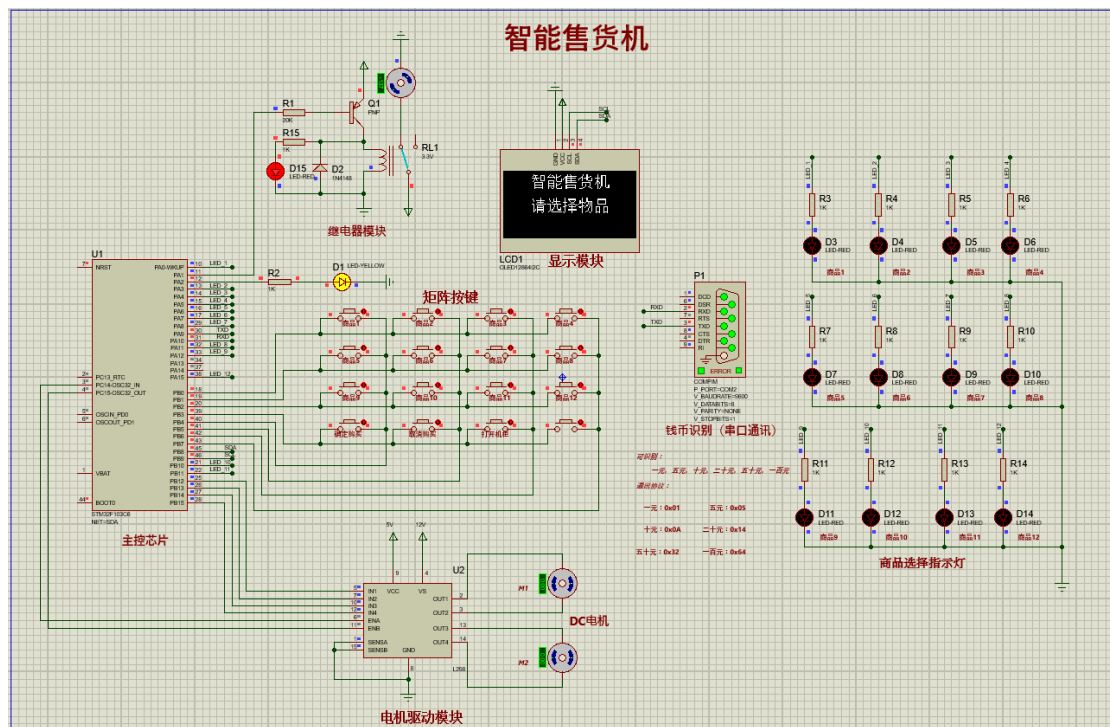


图 5-12 仿真启动

随机按下商品按键，OLED 显示屏会显示所选商品价格，商品选择指示灯也会亮起，表示商品已被选中，如图 5-13 所示。

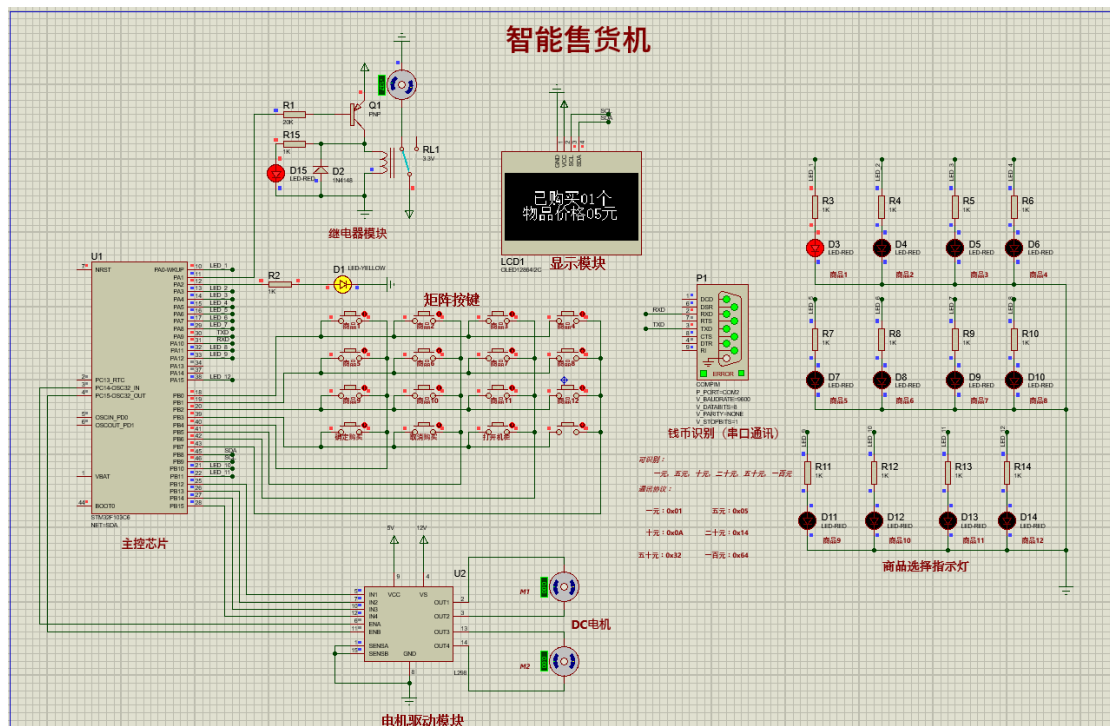


图 5-13 选中商品

选择商品完成后按下确定购买按键，此时 OLED 显示屏会显示出请付款与

商品价格，如图 5-14 所示。

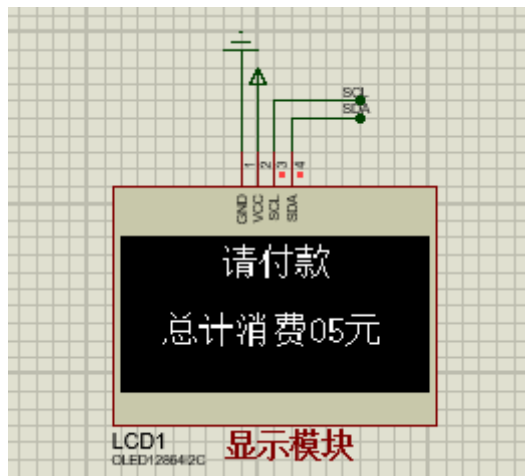


图 5-14 付款显示

若此时按下取消购买按键，便可取消购物，返回主界面，重新进行商品的选择与购买。出现付款界面后，我们通过串口助手发送十六进制指令，以表示付款，发送如图 5-15 所示。

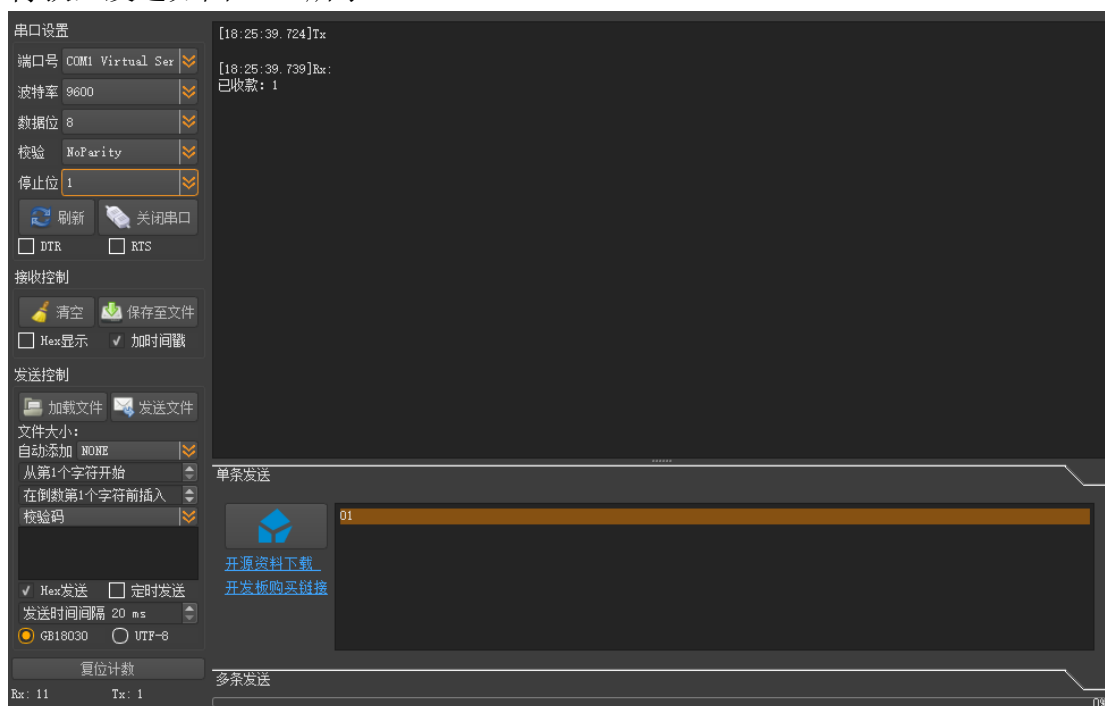


图 5-15 串口发送

此时发送的为 0x01，表示付款一元，付款金额小于实际货品金额，OLED 显示屏会显示商品价格与已付款金额，如图 5-16 所示。

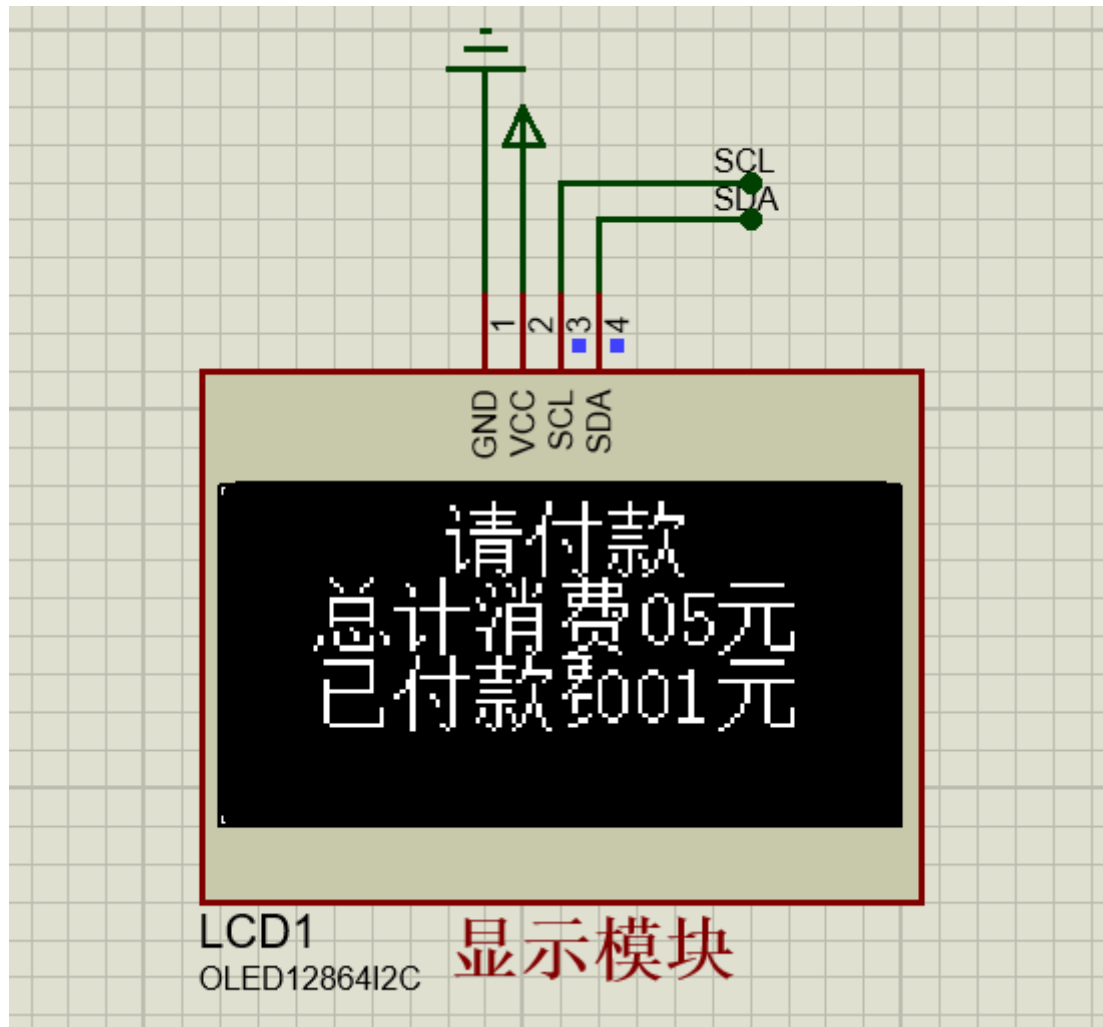


图 5-16 已付款金额显示

此时可以继续通过串口助手发送十六进制指令，当付款金额大于等于商品金额时，OLED 显示屏会显示找零，如图 5-17 所示。

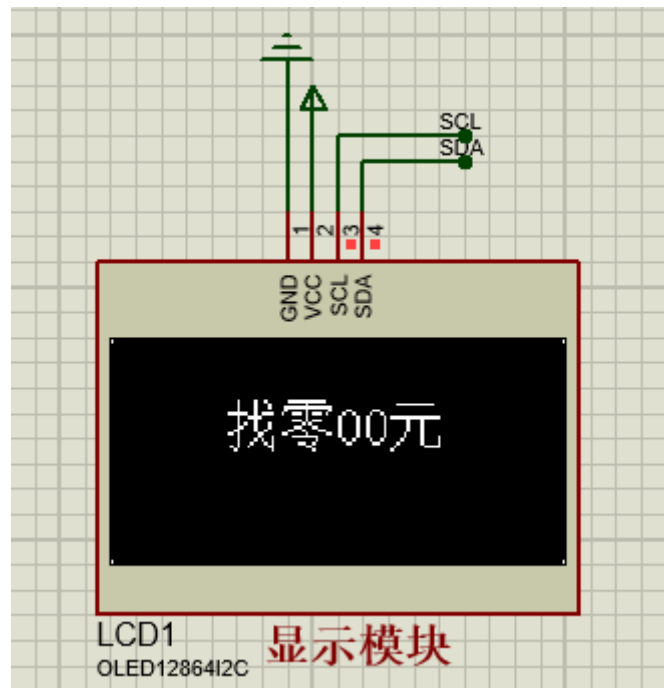


图 5-17 找零显示

找零完成后，出货电机与补货电机依次转动，表示出货，出货完成后电机停止转动，结束一次工作流程。电机转动如图 5-18 所示。

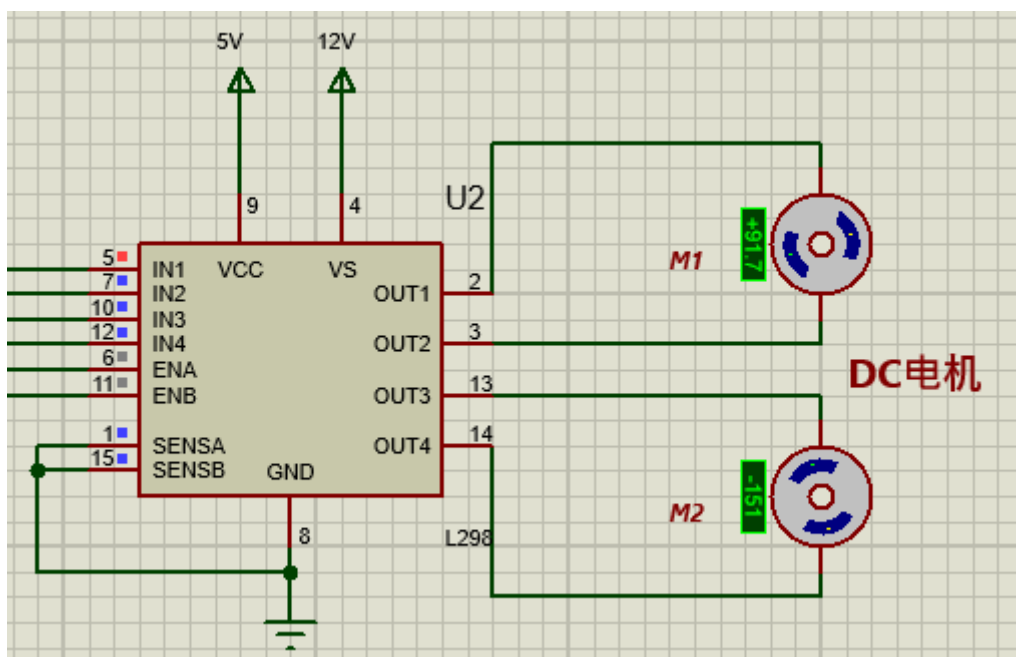


图 5-18 出货电机与补货电机

6 结论

在当代的社会中，自动售货机深受广大学生和上班族的欢迎，很多城市的公共场所里面都放置有自动售货机，自动售货机也为人们提供了很多便利。本次首先从自动售货机的功能要求开始设计，通过确定其所需功能继而确定其控制要求，然后确定输入输出，通过与其他控制的对比，选择单片机进行控制，设计流程图及设计程序。

在完成期末设计的过程中，我复习了单片机，数字电路模拟电路的相关知识，自己查找本文所需的单片机的特点，深入了解了单片机的功能与外围电路，产生的问题也通过与同学的讨论解决，得到了很大进步。本次期末设计将自动售货机分为几个模块：货币识别模块，货物选择模块，电机驱动出货模块等。由于字数有限，并没对自动售货机的智能化做更多的研究，但我相信，关于自动售货机的智能化将成为今后的主要研究方向。