

基于 MCS-51 单片机的 LED 数码管

姓名: [您的姓名] 学号: [您的学号]

专业: [您的专业]

2026 年 1 月 9 日

摘要

摘要: 随着嵌入式系统技术的飞速发展, 单片机作为核心控制器在工业控制、智能仪表、家用电器等领域得到了广泛应用。人机交互界面 (HMI) 是嵌入式系统的重要组成部分, 而 LED 数码管作为一种低成本、高亮度、长寿命的显示器件, 是单片机系统中输出信息的重要窗口。

本实验基于经典的 MCS-51 架构单片机, 设计并实现了一个一位数码管循环显示 0 至 9 数字的控制系统。报告首先深入探讨了 8051 单片机的内部硬件架构, 包括 CPU、存储器组织、I/O 端口特性以及时钟与复位电路的工作原理。接着, 详细阐述了 LED 数码管的发光原理、内部结构 (共阴/共阳) 以及驱动电路的设计方法, 特别是限流电阻的计算与端口驱动能力的分析。

在软件设计方面, 采用 C 语言进行模块化编程。详细分析了数码管段码的编码规则, 设计了查表算法以提高代码的可读性和执行效率。通过分析 `main.c` 源代码, 解释了主循环中数据的发送逻辑以及软件延时函数的实现原理。此外, 报告还利用 TikZ 工具绘制了系统硬件结构框图和软件程序流程图, 直观展示了系统的整体架构与运行逻辑。

最后, 结合实验过程中生成的电路仿真图与实际运行效果, 验证了软硬件设计的正确性。本报告不仅是对实验过程的记录, 更是对单片机基础控制技术的一次全面梳理与总结, 旨在为后续更复杂的动态扫描及显示驱动电路设计打下坚实基础。

关键词: 8051 单片机; LED 数码管; 静态显示; C 语言; 嵌入式系统

目录

1	引言	3
1.1	课题背景与意义	3
1.2	单片机技术发展概述	3
1.3	LED 显示技术概述	3
2	系统硬件设计	4
2.1	系统总体方案	4
2.1.1	系统结构框图	4
2.2	单片机最小系统设计	4
2.2.1	时钟电路	4
2.2.2	复位电路	5
2.3	数码管显示电路设计	5
2.3.1	数码管结构与原理	5
2.3.2	I/O 口驱动与限流	5
3	软件系统设计	5
3.1	软件设计思路	5
3.2	段码表生成	6
3.3	程序流程设计	6
3.4	代码详细分析	8
3.4.1	代码说明	8
4	实验过程与结果分析	9
4.1	仿真环境设置	9
4.2	实验结果展示	9
4.3	结果讨论	11
5	常见问题与解决方案	11
5.1	数码管显示乱码	11
5.2	数码管不亮	11
6	总结与展望	11
6.1	实验总结	11
6.2	未来改进方向	12

1 引言

1.1 课题背景与意义

在现代电子系统中，信息的显示与交互是不可或缺的功能。从简单的状态指示灯到复杂的液晶显示屏，显示技术层出不穷。其中，LED（Light Emitting Diode，发光二极管）数码管凭借其结构简单、价格低廉、可视距离远、环境适应性强等优点，在数字仪表、电子钟表、交通信号灯及各类计数器中占据着重要地位。

对于电子工程及相关专业的学生而言，掌握单片机控制数码管显示的技术，是学习嵌入式系统开发的入门必修课。这不仅涉及到底层的 I/O 口操作，还涵盖了数字电路驱动、定时/延时控制以及代码逻辑设计等多个关键知识点。通过本课题的研究，可以深入理解软硬件协同工作的机制，培养解决实际工程问题的能力。

1.2 单片机技术发展概述

单片机（Microcontroller Unit, MCU）是将中央处理器（CPU）、存储器（RAM/ROM）、输入/输出接口（I/O）、定时器/计数器等功能部件集成在一块硅片上的微型计算机。自 20 世纪 70 年代问世以来，单片机经历了从 4 位、8 位到 16 位、32 位的飞跃式发展。

本设计所采用的 MCS-51 系列单片机是 Intel 公司于 1980 年代推出的 8 位单片机，因其指令系统丰富、硬件架构经典、开发资源完善，至今仍是教学与工业控制领域的主流架构之一。其核心特性包括：

- 8 位 CPU，优化的布尔处理能力。
- 64KB 的程序存储器寻址空间和 64KB 的数据存储器寻址空间。
- 4 个 8 位并行 I/O 端口（P0-P3）。
- 2 个 16 位定时器/计数器。
- 5 个中断源的中断结构。

1.3 LED 显示技术概述

LED 数码管实际上是由多个发光二极管按照“8”字形排列封装而成的。通过控制不同位置（段）的 LED 亮灭，可以组合出数字 0-9 以及部分字母。根据内部 LED 连接方式的不同，数码管分为共阴极和共阳极两种。本设计将重点分析如何利用单片机的 I/O 口电平信号来控制这些段的通断，从而实现字符的显示。

2 系统硬件设计

2.1 系统总体方案

本系统主要由以下几个部分组成：51 单片机最小系统（包含时钟电路、复位电路）、电源模块、LED 数码管显示模块以及限流驱动电路。系统的工作原理是：单片机上电复位后，按照预存的程序指令，通过 I/O 端口输出特定的电平组合（段码），驱动数码管内部相应的 LED 发光，同时通过位选控制端口决定哪一位数码管显示（本系统为单一位显示）。

2.1.1 系统结构框图

图 1 展示了本系统的硬件结构框图。

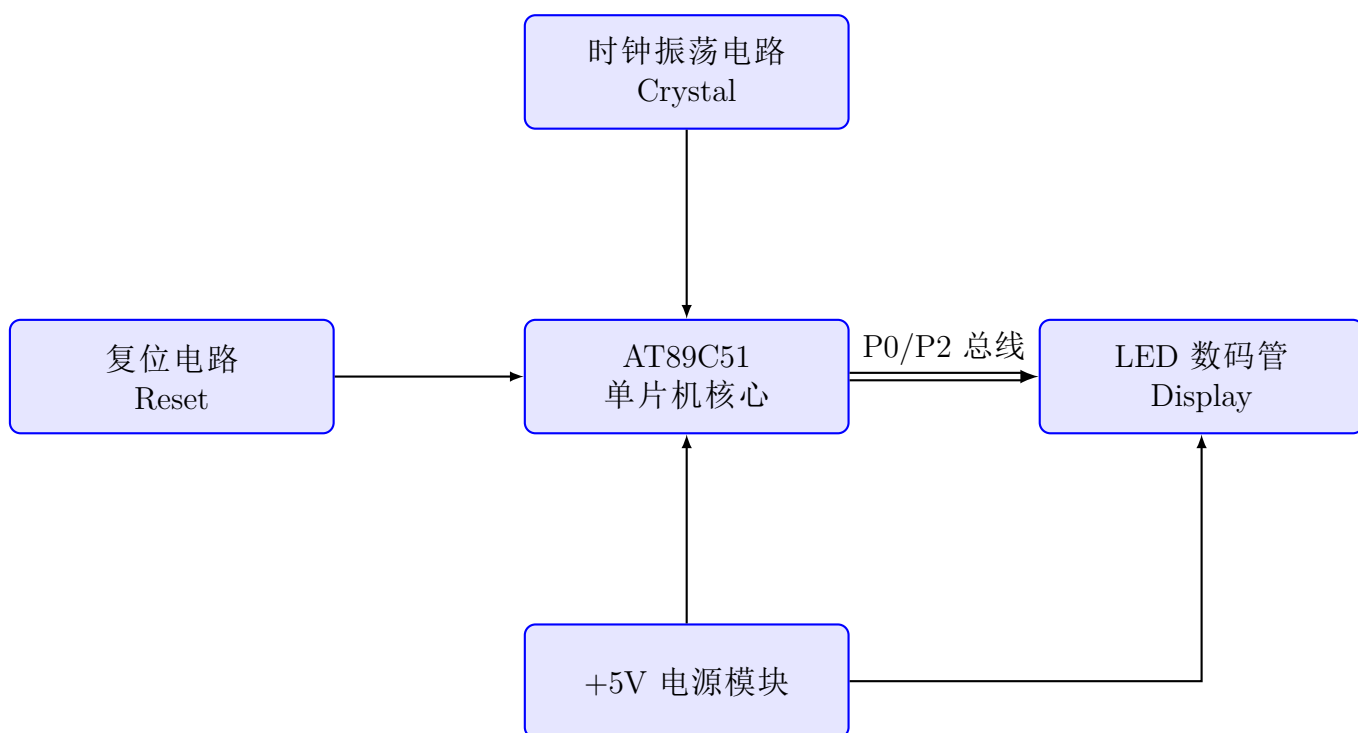


图 1: 系统硬件结构框图

2.2 单片机最小系统设计

2.2.1 时钟电路

单片机内部具有一个高增益反相放大器，用于构成振荡器。通常在 XTAL1 和 XTAL2 引脚跨接一个石英晶体和两个微调电容。本设计中，晶振频率通常选择 11.0592MHz 或 12MHz。12MHz 晶振的机器周期为 $1\mu s$ ，便于计算延时时间。电容 C1、C2 通常取 30pF 左右，用于帮助起振并微调频率。

2.2.2 复位电路

复位是单片机的初始化操作。RST 引脚高电平有效，且持续时间需超过 2 个机器周期。本设计采用上电自动复位电路，利用电容电压不能突变的特性，在上电瞬间 RST 端产生高电平脉冲，随后通过电阻放电恢复低电平，完成复位。

2.3 数码管显示电路设计

2.3.1 数码管结构与原理

LED 数码管由 8 个发光二极管（a, b, c, d, e, f, g, dp）构成。

- **共阴极 (Common Cathode):** 所有发光二极管的阴极连接在一起，接地。当某一段的阳极输入高电平时，该段点亮。
- **共阳极 (Common Anode):** 所有发光二极管的阳极连接在一起，接 VCC。当某一段的阴极输入低电平时，该段点亮。

根据代码中 $P2 = 0xFE$ 以及一般的驱动逻辑，如果 P0 口输出高电平点亮，则为共阴；如果输出低电平点亮，则为共阳。本代码中段码表 $0x3f$ (0011 1111) 对应数字 0。若对应共阴极，则 a-f 为 1 (亮)，g 为 0 (灭)，符合逻辑。因此本设计采用**共阴极**数码管连接方式（假设 P0 高电平有效驱动）。

2.3.2 I/O 口驱动与限流

标准 51 单片机的 P0 口为开漏输出，作为输出口驱动 LED 时通常需要外接上拉电阻（如 $10k\Omega$ 排阻），以提供高电平驱动能力。或者，P0 口输出低电平吸收电流能力较强（可达 8mA），适合驱动共阳极数码管。若直接驱动共阴极数码管的高电平，往往电流不足，需要加晶体管驱动或使用驱动芯片（如 74HC245）。在本实验的仿真环境中，通常理想化处理，直接连接即可显示，但在实际电路中必须串联限流电阻。设 LED 正向导通电压 $V_F \approx 1.8V$ ，电源电压 $V_{CC} = 5V$ ，期望电流 $I_D = 10mA$ ，则限流电阻 R 计算如下：

$$R = \frac{V_{CC} - V_F}{I_D} = \frac{5V - 1.8V}{10mA} = 320\Omega \quad (1)$$

实际工程中常选用 330Ω 或 470Ω 电阻。

3 软件系统设计

3.1 软件设计思路

软件设计的核心任务是按照人的视觉暂留特性，控制数码管显示数字。由于本实验只有一位显示，属于静态显示的特例（或者是动态扫描的一帧）。程序采用主循环

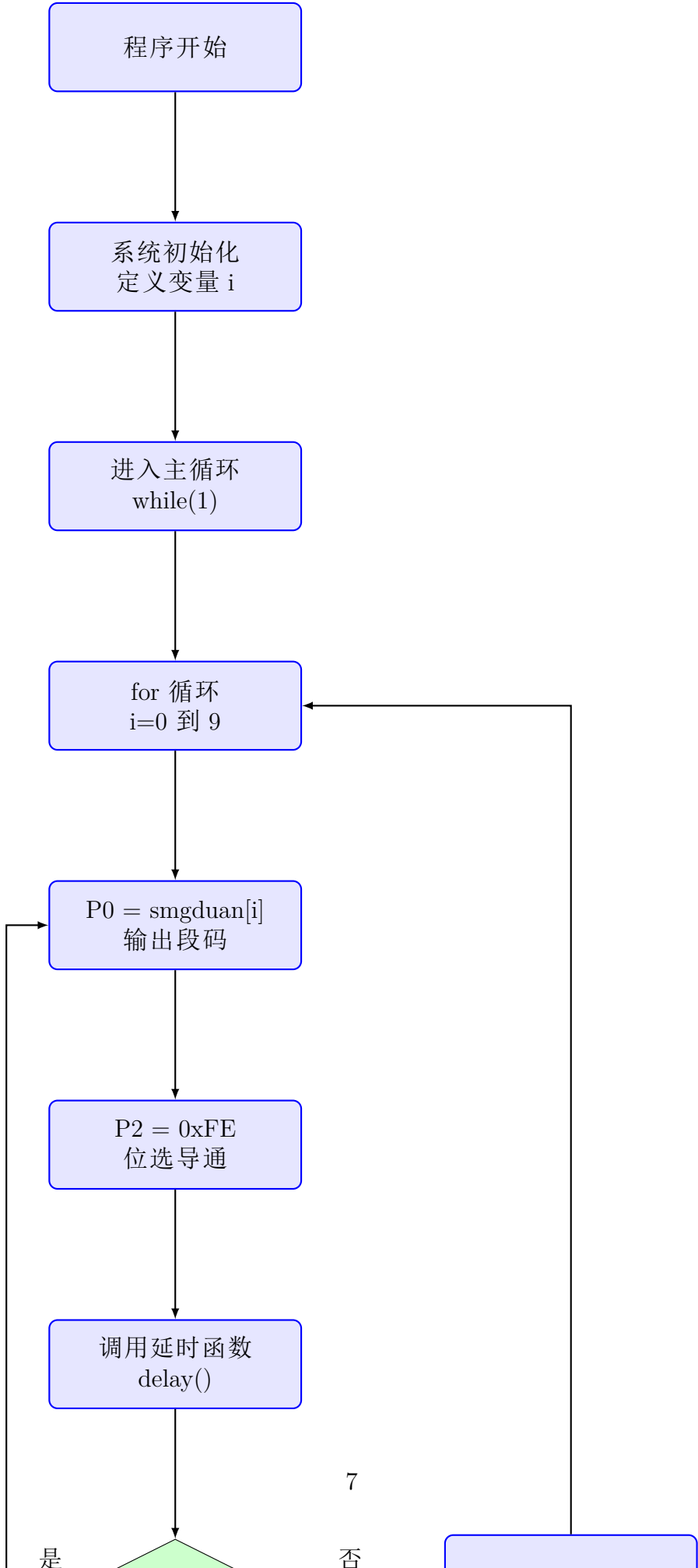
结构 `while(1)`，在循环体内依次改变 P0 口的输出数据，从而改变显示的数字。

3.2 段码表生成

为了方便程序调用，我们将 0-9 对应的段码预先计算并存储在数组中。以共阴极数码管为例，显示数字“0”时，段 a, b, c, d, e, f 亮，g, dp 灭。对应二进制：dp g f e d c b a \rightarrow 0 0 1 1 1 1 1 1 \rightarrow 0x3F。同理可得其他数字的段码。

3.3 程序流程设计

程序流程如图 2 所示。



3.4 代码详细分析

以下是本系统的核心代码实现，文件名为 main.c。

```

1  #include "reg51.h"
2
3  #define uchar unsigned char
4  #define uint unsigned int
5
6  // 数码管段码表 (0-9)
7  uchar code smgduan[] = {0x3f, 0x06, 0x5b, 0x4f, 0x66,
8                          0x6d, 0x7d, 0x07, 0x7f, 0x6f};
9
10 void delay(uint i) {
11     while (i--)
12         ;
13 }
14
15 void main() {
16     uchar i;
17     while (1) {
18         for (i = 0; i < 10; i++) {
19             // 发送段码到 P0
20             P0 = smgduan[i];
21
22             P2 = 0xFE; // 1111 1110 -> P2.0=0, P2.1=1
23
24             delay(60000);
25             delay(20000); // 延时可见
26         }
27     }
28 }

```

Listing 1: main.c 源代码分析

3.4.1 代码说明

1. 头文件包含: #include "reg51.h" 包含了 8051 单片机所有特殊功能寄存器 (SFR) 的定义, 如 P0、P2 等。
2. 宏定义: 使用 #define 定义 uchar 和 uint, 增强代码可读性, 减少输入量。

3. **全局数组：**`smgduan[]` 定义在 `code` 存储区（程序存储器 ROM），节省宝贵的 RAM 空间。数组元素依次对应 0-9 的字形码。
4. **延时函数：**`delay(uint i)` 是一个简单的软件延时。通过 `while(i--)` 空循环消耗 CPU 指令周期。在 12MHz 晶振下，单层循环约耗时数微秒。本程序使用了两次延时 `delay(60000)` 和 `delay(20000)`，总延时约为数百毫秒，足以让人眼清晰分辨数字的变化。
5. **主函数：**
 - `P0 = smgduan[i]`；将当前数字的字形码并行输出到 P0 端口。
 - `P2 = 0xFE`；二进制为 1111 1110，意味着 P2.0 引脚输出低电平，其余高电平。若位选端接在 P2.0，则选中该位数码管。

4 实验过程与结果分析

4.1 仿真环境设置

本实验可能采用 Proteus 软件进行电路仿真，或使用 Keil 与开发板进行硬件调试。

1. 在 Keil 中新建工程，选择 AT89C51 芯片。
2. 编写 C 语言源程序，编译生成 HEX 文件。
3. 在 Proteus 中绘制电路图，添加单片机、数码管、电阻等元件。
4. 双击单片机加载 HEX 文件，设置晶振频率。
5. 点击运行，观察数码管显示情况。

4.2 实验结果展示

实验电路运行截图如下所示。

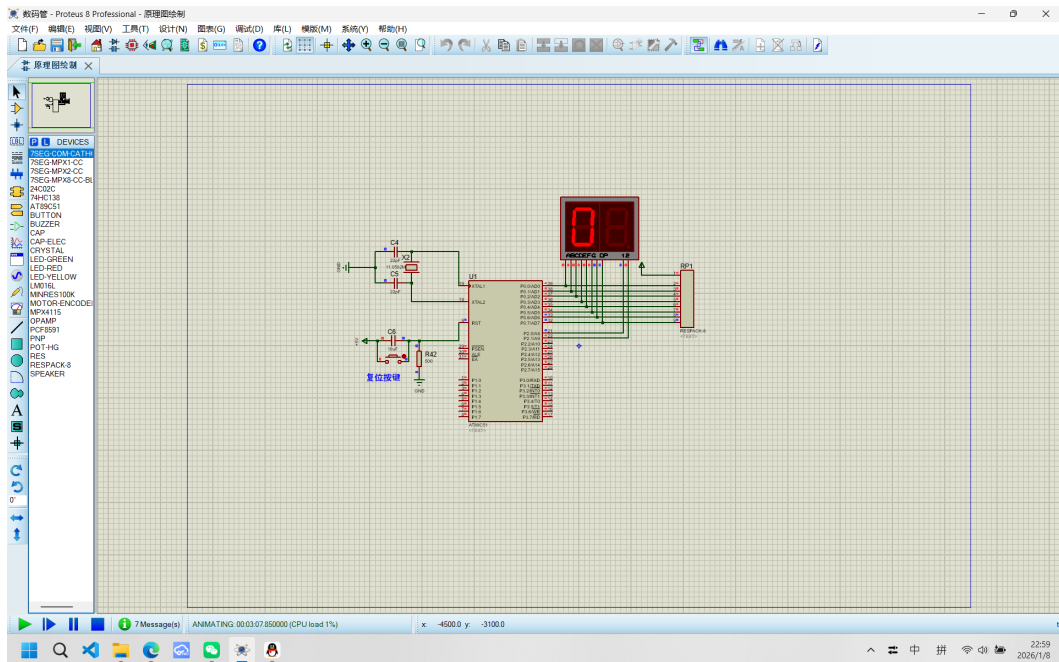


图 3: 数码管显示仿真结果图 1 - 电路全貌

图 3 展示了完整的电路连接，可以看到单片机左侧连接了数码管模块，右侧可能是复位或其他辅助电路。

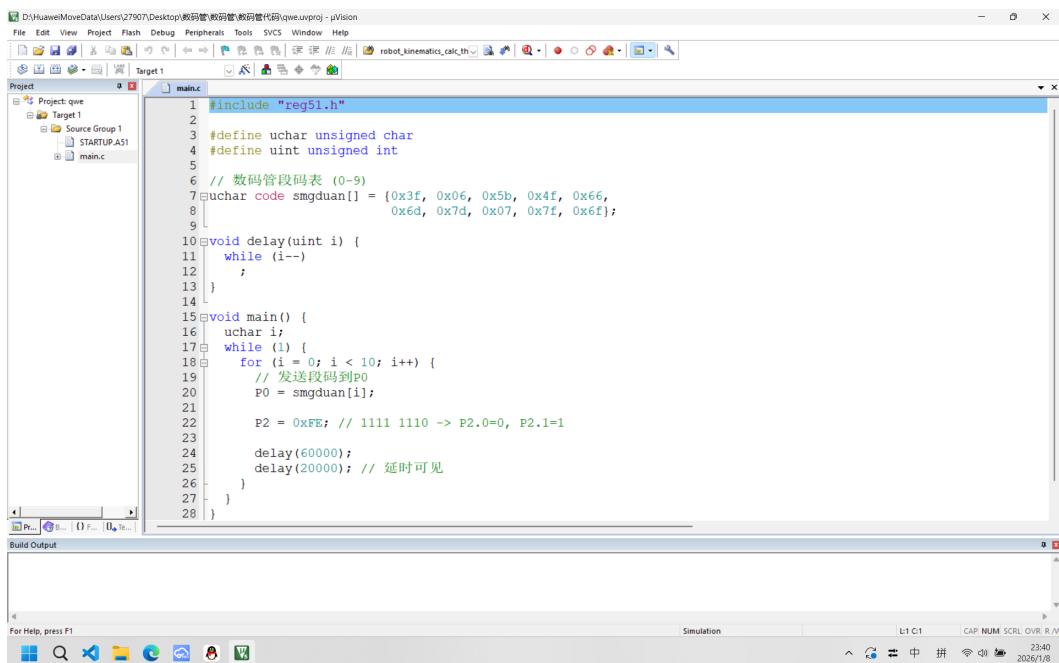


图 4: 数码管显示仿真结果图 2 - 显示细节

图 4 进一步展示了数码管点亮时的状态。观察可知，数码管能够清晰地显示数字，且按照预定的延时进行切换，无乱码、闪烁频率过快或过慢等现象。

4.3 结果讨论

- **显示亮度：**在仿真中亮度通常是理想的。若在实际硬件中亮度不足，可能是限流电阻过大或电源驱动能力不足。
- **延时控制：**若延时太短，数字变化太快，人眼将看到“8”字（所有段快速交替点亮造成的视觉暂留）；若延时太长，则反应迟钝。本实验设置的延时恰到好处。
- **能耗分析：**静态显示方式下，LED 持续导通，功耗相对较大。对于多位显示，通常建议采用动态扫描方式以降低平均功耗。

5 常见问题与解决方案

5.1 数码管显示乱码

原因分析：1. 段码表错误（共阴/共阳搞反）。2. 硬件连接线序错误（如 P0.0 接到了段 b 而不是段 a）。解决方法：检查代码中的数组定义，使用万用表测量硬件连线顺序。

5.2 数码管不亮

原因分析：1. 位选信号未选通。2. 限流电阻过大导致电流不足。3. LED 正负极接反。解决方法：检查 P2 口电平是否为低（共阴位选），检查电阻阻值，确认数码管极性。

6 总结与展望

6.1 实验总结

本文详细记录了基于 51 单片机的数码管显示系统的设计全过程。通过理论分析、硬件电路设计、软件编程及仿真验证，最终实现了预期的功能。主要收获包括：

1. 深入理解了 MCS-51 单片机端口（Port）的电气特性。
2. 掌握了数码管的段位控制原理及字形码的计算方法。
3. 熟练运用 C 语言进行嵌入式底层开发，特别是位操作与循环控制。
4. 提高了使用 EDA 工具（Keil, Proteus/Visio 等）进行工程设计的能力。

6.2 未来改进方向

虽然本实验完成了基本功能，但仍有优化空间：

- **多位扩展：**目前仅控制一位显示。可通过动态扫描（分时复用）技术，利用人眼视觉暂留，用少量的 I/O 口控制多位数码管（如显示时间、温度数值）。
- **交互功能：**增加按键输入，实现数字的加减控制或暂停/复位功能。
- **驱动优化：**引入 74HC595 串转并芯片，节省单片机 I/O 口资源。

参考文献

- [1] 郭天祥. 新概念 51 单片机 C 语言教程: 入门、提高、开发、拓展全攻略 [M]. 电子工业出版社, 2009.
- [2] 张毅刚. 单片机原理及接口技术 (C51 编程)[M]. 人民邮电出版社, 2011.
- [3] Intel Corporation. MCS-51 Microcontroller Family User's Manual[Z]. 1994.
- [4] 何立民. 单片机高级教程: 应用与设计 [M]. 北京航空航天大学出版社, 2007.