

电子线路设计训练专题实验 2 实验报告

西安交通大学



## 基于 C8051F020 智能控制器的 游戏设计实验报告

自动化 94—胡欣盈—2194323176

# 目 录

一、游戏简介.....	3
二、实验设备.....	3
三、硬件结构.....	4
3.1 C8051F020.....	5
3.1.1 C8051F020 单片机简介.....	5
3.1.2 C8051F020 单片机特性.....	6
3.1.3 C8051F020 引脚说明.....	6
3.2 LCD 显示电路.....	8
3.2.1 字符显示 RAM（DDRAM）.....	9
3.2.2 绘图 RAM（GDRAM）.....	9
3.3 按键电路.....	10
3.4 LED 显示电路.....	11
四、软件实现.....	12
4.1 功能介绍.....	12
4.2 系统流程图.....	12
4.3 主要函数代码.....	13
4.3.1 LED 显示.....	13
4.3.2 按键中断.....	13
4.3.3 LCD 显示.....	13
五、结果展示与分析.....	14
六、实验总结.....	16
七、参考文献.....	16
八、附录.....	17

# “接樱花”小游戏设计

## 一、游戏简介

“接樱花”小游戏是一个基于 C8051F020 智能控制器实验平台设计的单机游戏。程序下载启动后，会在 LCD 屏幕上显示主菜单，内含游戏名字与简易规则（按键盘中间键开始，左右键移动），按中间键开始后计时器开始 10 秒倒计时，计时显示在第一组 LED 上，LCD 屏幕中会出现小人和樱花图片，按左右键可以移动小人，樱花会缓慢下降，若在樱花落下前使小人与花对齐（即接到樱花）可计一分，得分会显示在第二组 LED 上，樱花落下则视为一局结束，总局数会显示在第三组 LED 上。10 秒倒计时结束后，游戏自动结束并回到主菜单。

游戏设有以下 3 大板块：

1. LCD 显示区：LCD 用来显示开始和结束的菜单，游戏过程中显示人与樱花的位置关系。
2. 键盘控制区：3 个按键分别被赋予了移动人物和开始的功能。
3. LED 提示区：3 组 LED 分别被赋予了展示倒计时、展示当前得分、展示当前总局数的功能。

## 二、实验设备

基于 C8051F020 的智能控制器、计算机、Keil C51 编程软件

### 三、硬件结构

本次实验所用硬件以基于 C8051F020 的智能控制器为主，它是以 C8051F020 单片机为核心部件，增加了按键模块、数码管和液晶屏显示模块、AD 和 DA 输入输出信号调理电路等。智能控制器系统框图如图 3-1 所示，智能控制器实验板实物如图 3-2 所示。

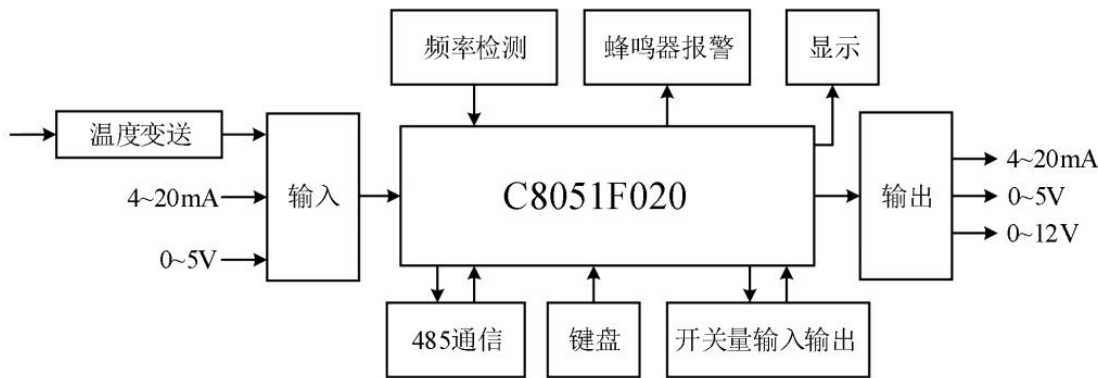


图 3-1 智能控制器系统框图

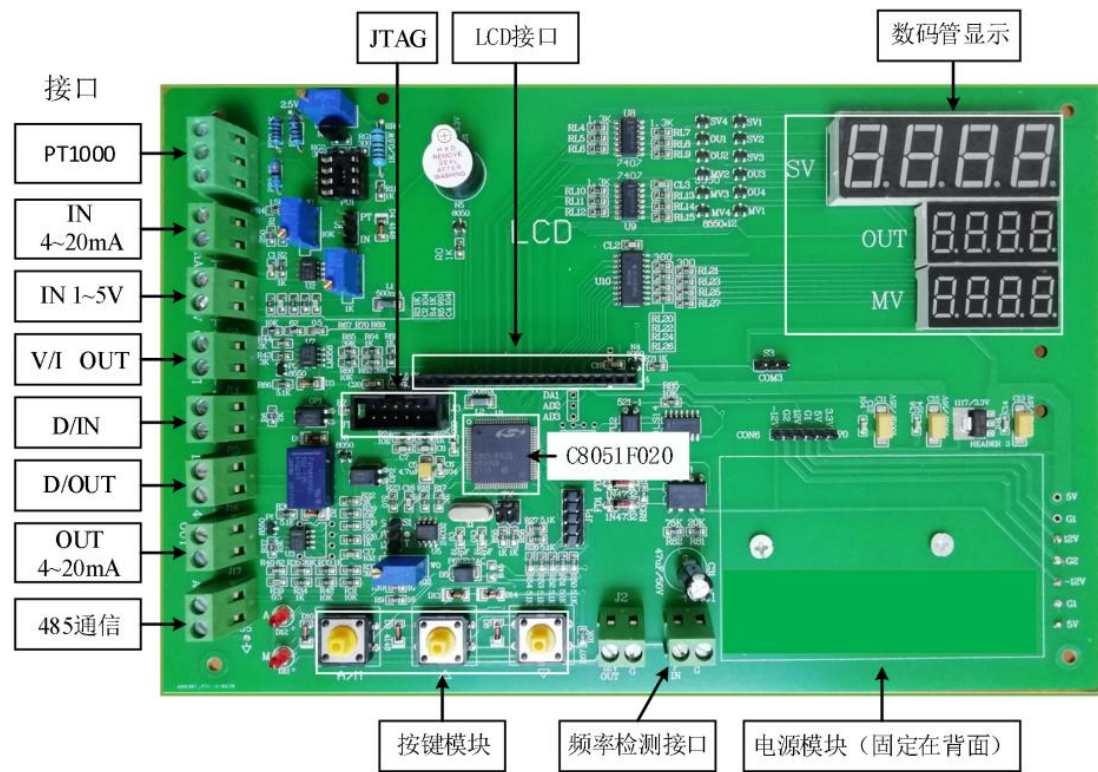


图 3-2 智能控制器实物板



### 3.1.2 C8051F020 单片机特性

- ① 高速、流水线结构的 8051 兼容的 CIP-51 内核。
- ② 全速、非侵入式的在系统调试接口 JTAG。
- ③ 64K 字节可在系统编程的 FLASH 存储器。
- ④ 4352 (4096+256) 字节的片内 RAM。
- ⑤ 可寻址 64K 字节地址空间的外部数据存储器接口。
- ⑥ 真正 12 位 100ksps 的 8 通道 ADC0, 带 PGA 和模拟多路开关。
- ⑦ 真正 8 位 500ksps 的 ADC1, 带 PGA 和 8 通道模拟多路开关。
- ⑧ 两个 12 位 DAC, 具有可编程数据更新方式。54
- ⑨ 硬件实现的 SPI、SMBus/I2C 和两个 UART 串行接口。
- ⑩ 5 个通用的 16 位定时器。
- ⑪ 具有 5 个捕捉/比较模块的可编程计数器/定时器阵列 (PCA)。
- ⑫ 片内看门狗定时器、VDD 监视器和温度传感器。
- ⑬ 具有片内 VDD 监视器、看门狗定时器和时钟振荡器的

C8051F020 是真正能独立工作的片上系统。

### 3.1.3 C8051F020 引脚说明

基于 C8051F020 单片机的特性, 智能控制器在设计时, 选取该单片机作为控制、计算、显示的核心部件。C8051F020 单片机有 100 个引脚, 封装为 TQFP-100, 其引脚如图 3-4 所示。C8051F020 单片机低端口 (P0、P1、P2、P3) 既可以按位寻址, 也可以按字节寻址, 高端口 (P4、P5、P6、P7) 只能按字节寻址, 所有引脚都可以被配置为开漏或推挽输出方式。C8051F020 单片机有大量的数字资源需要通过 P0、

P1、P2 和 P3 端口才能使用。P0、P1、P2 和 P3 中的每个引脚即可定义为通用的 I/O 端口引脚,也可以分配给一个数字外设或功能(例如:UART0 或 INT1)。这种资源分配的灵活性是通过使用优先权交叉开关实现的。

交叉开关按优先权顺序将端口 P0、P1、P2、P3 引脚分配给单片机的数字外设（UART、SMBus、PCA、定时器等），端口引脚的分配顺序是从 P0.0 开始可以一直分配到 P3.7。当交叉开关配置寄存器 XBR0、XBR1 和 XBR2 中外设的对应允许位被设置为逻辑 ‘1’ 时，交叉开关将端口引脚分配给外设。因为 UART0 有最高优先权，所以当 UART0EN 位被设置为逻辑 ‘1’ 时，其引脚将总是被分配到 P0.0 和 P0.1。被交叉开关分配的端口引脚输出状态受使用数字外设的控制，向端口寄存器（或相关端口位）写入时，对引脚的状态没有影响，但在执行读-修改-写的读周期，所读的值是端口数据寄存器的内容，而不是端口引脚的状态。因为交叉开关寄存器影响外设的引脚，所以在外设被配置前，由系统的端口初始化代码配置。一旦端口在初始化时进行了交叉开关配置，则在程序运行过程中，不再对其重新编程。

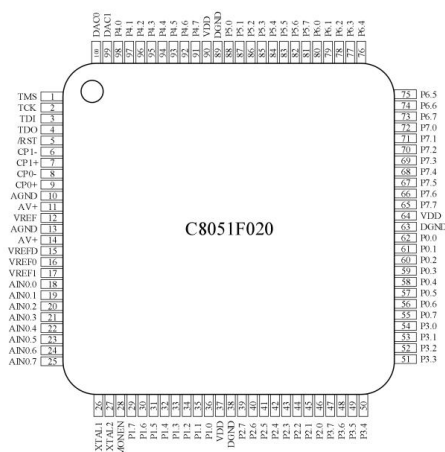


图 3-4 C8051F020 引脚图

3.2 LCD 显示电路

图 3-5 液晶显示电路中，智能控制器选用 HS12864-15B 汉字图形型液晶，带中文字库。液晶显示采用串口通信模式，可以显示字母、数字符号、中文字型及图形，具有绘图及文字画面混合显示功能。该液晶共有 20 个引脚，E、RW、RS 分别接在单片机的 P1.3、P1.4、P1.5 引脚，引脚说明如图 3-6 所示，没有列出的引脚是空接状态。P1.3、P1.4、P1.5 引脚在系统端口初始化时被设置为推挽模式，一旦端口在初始化时进行了交叉开关配置，则在程序运行过程中，端口不能进行修改。RW 引脚（P1.4）在端口初始化时被配置成推挽输出，则不能读取 LCD 返回的数据。如果需要液晶显示温度变化曲线，软件编程时设置一个“虚拟屏幕”，对数据处理后，再刷新到真实的液晶屏显示。

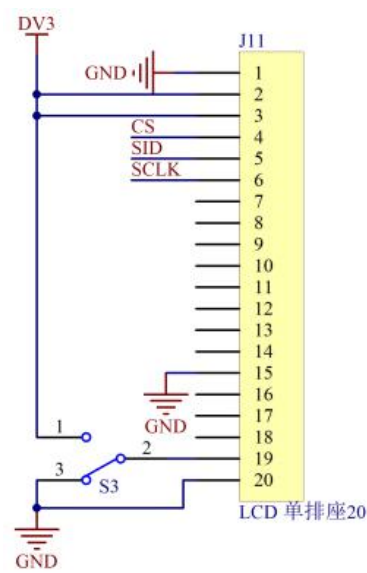


图 3-5 液晶显示电路

管脚号	名称	电平	功能
1	GND	0V	电源地
2	VCC	+5V	模块电源输入
3	V0	-	对比度（亮度）调整
4	RS	H/L	片选端，高电平有效
5	RW	H/L	串行数据线
6	E	H/L	串行时钟输入
15	PSB	L	L:串口方式
17	RST	H/L	复位端，低电平有效
19	A	VDD	背光电压
20	K	GND	背光源负端0V

图 3-6 12864-15B 液晶引脚说明



3.2.1 字符显示 RAM（DDRAM）

HS12864-15B 液晶的控制 IC 为 ST7920。ST7920 内置 2M 位中文字型 ROM(CGROM)，总共提供 8192 个中文字型(16x16 点阵)，16K 位半宽字型 ROM(HCGROM)，总共提供 126 个符号字型(16x8 点阵)，64x16 位字型产生 RAM(CGRAM)，用来用户自定义字型。显示中文字符时将 16 位数据送入 DDRAM 中，先写高 8 位(D16~D8)，再写低 8 位(D7~D0)，可显示 4 行，每行显示 8 个汉字，共显示 32 个汉字。DDRAM 在液晶模块中的地址为 80H~9FH，字符显示的 RAM 地址与字符在屏幕上的显示区域是一一对应的关系，如图 3-7 所示。

	列 1	列 2	列 3	列 4	列 5	列 6	列 7	列 8
行 1	80H	81H	82H	83H	84H	85H	86H	87H
行 2	90H	91H	92H	93H	94H	95H	96H	97H
行 3	88H	89H	8AH	8BH	8CH	8DH	8EH	8FH
行 4	98H	99H	9AH	9BH	9CH	9DH	9EH	9FH

图 3-7 字符显示的 DDRAM 地址与显示区域对应关系

3.2.2 绘图 RAM（GDRAM）

绘图 GDRAM 由扩充指令进行设置。横坐标将 128 点分为 16 点一列，共 8 列，纵坐标将 64 点分为 64 行。写入数据时，先写入垂直地址，再写入水平地址，最后连续写入两字节 8bit 数据，先高 8 位，后低 8 位。在编程显示图片时，先将其调整成合适尺寸的图片，然后通过取模软件对图片取模，保存成点阵数组。显示时，数据以长度为[1024]大小的数组形式送入 GDRAM，数组内每个元素为一个字节，即 8 位二进制数。

### 3.3 按键电路

智能控制器的按键电路，如图 3-8 所示。三个按键信号 A8、A9、A10 分别接在 C5051F020 单片机的 P5.0、P5.1、P5.2 引脚，中断信号 INT1 接在 P0.3 引脚。按键由外部中断信号触发，低电平有效，按键按下触发中断，进入按键中断服务程序，完成一定功能后再回到中断前正在执行的程序。

INT1 中断信号和数码管 P5 的低三位复用，假设某一按键按下，P5 对应的位置为逻辑‘0’，此时这条线路形成通路，才能将低电平信号和 INT1 连通。如果在按键扫描时，让 P5 的三个端口状态轮流为逻辑‘0’，读取 P5 端口的值，就可以建立 P5 的端口和三个按键一一对应的关系。共阳极数码管显示的位选是轮流使能的过程，故将数码管的位选端口与 INT1 的端口复用。

INT1 端口对应的 P0.3 端口，在应用前需要由交叉开关进行设置。交叉开关是 C8051F020 中端口配置的特色，可以通过交叉开关寄存器来控制端口为普通的数据输入输出端口，还是具有特定功能的端口。P0、P1 等的全部或部分端口被交叉开关配置成推挽模式，以保证正常输出。

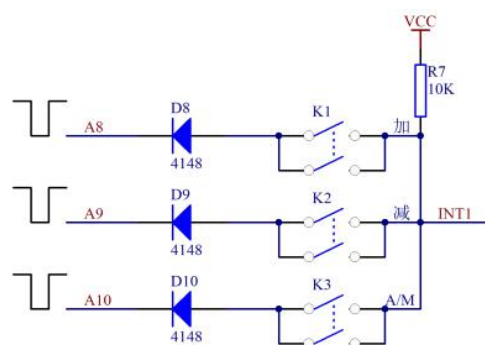


图 3-8 按键电路

### 3.4 LED 显示电路

智能控制器设置了三组四位数码管，其中一组数码管的显示电路，如图 3-9 所示。三组数码管的段选信号 Q0~Q7，通过 74HC245 分别接在单片机的 P7.0~P7.7 引脚，三组共 12 位数码管的位选信号 LED21~LED24、LED25~LED28、LED29~LED32，通过 7407 驱动器分别接在单片机的 P5.0~P5.3、P5.4~P5.7、P6.0~P6.3 引脚。数码管采用动态扫描显示方式，数码管为共阳极接法，位选信号为逻辑‘0’表示该位对应的数码管被选中，数码管显示内容由段选信号决定，利用余辉效应可以分时复用 P7 端口，来点亮特定的数码管，显示对应的数字。在实际应用中，采用两种方式可以达成余晖的效果。一种方式是在程序中设置延时，来制造余晖的效果。另一种方式是程序在一次循环中需要执行很多指令的情况下，这样即使设置了延迟，程序执行的时间往往远超过期望的延迟，则余晖的效果会变成闪烁，严重影响数码管的显示。在这种情况下，可以考虑采用定时器中断进行特定周期的触发以达成稳定显示。

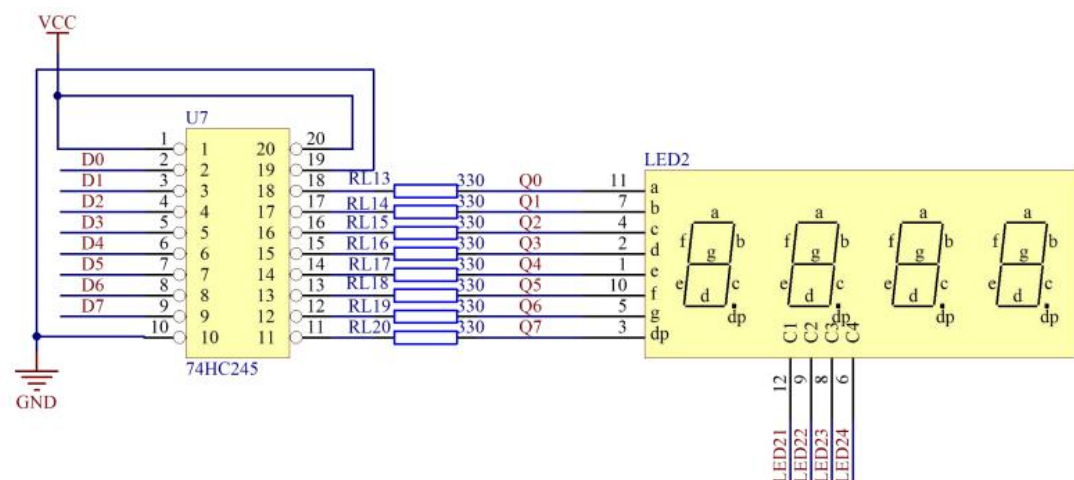


图 3-9 数码管显示电路

## 四、软件实现

### 4.1 功能介绍

- 1、LCD 显示初始菜单
- 2、按键控制游戏开始与人物移动方向
- 3、第一组 LED 显示游戏倒计时
- 4、第二组 LED 显示当前游戏得分
- 5、第三组 LED 显示当前游戏总局数

### 4.2 系统流程图

在软件设计中程序分别完成 LCD 初始化及显示、按键输入检测、倒计时运算、得分计算、局数统计、LED 显示等功能。对主程序进行初始化,其他程序选择模块化的方式实现。首先对每个模块进行调度,再逐一加入主程序中,最后完成整个软件部分的设计。系统流程图如图 4-1 所示。

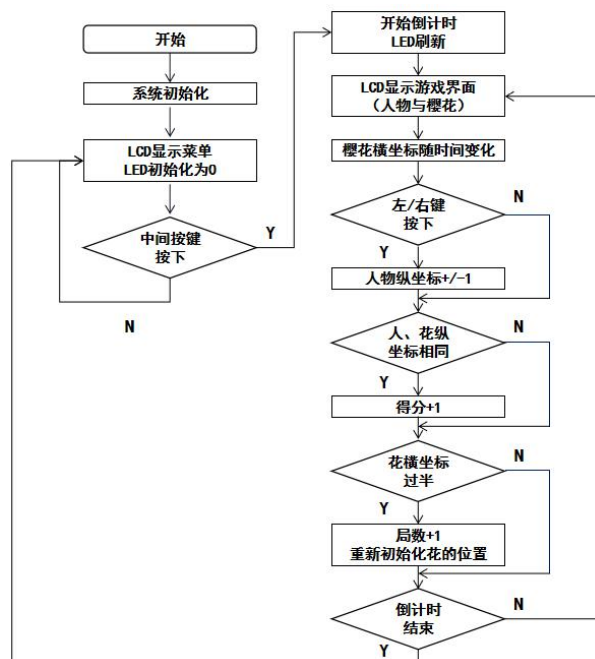


图 4-1 系统流程图

### 4.3 主要函数代码

本设计程序采用 C 语言编程，程序分为多个模块，由主程序、LED 显示和 LCD 系列程序等模块组成。具体程序见附件，此处简要说明各模块函数功能。

#### 4.3.1 LED 显示

在主程序文件中定义了 Time\_num、zongshu、defen 这 3 个 unsigned int 类型的全局变量分别用于倒计时、记录总局数、记录当前得分。在 led.c 文件中外部引用这三个变量，并使三组 LED 数码管分别显示其数值即可。通过修改例程中的 LedDispNum() 函数可实现该功能，即将 3 个 4 位数分别存在对应的 3 个 4 位数组中，通过片选可将数组中的每一个数赋予对应的数码管，使数码管在主程序中不断刷新即可实现 3 组 LED 显示。

#### 4.3.2 按键中断

在主程序文件中定义了按键中断对应的函数 INT1\_ISR(void) 如图 4-2 所示，其中变量 ren\_posi 代表人物的列坐标，%8 的操作使人在到达左右边界时可从另一边出现，即 LCD 左右边界相连。

#### 4.3.3 LCD 显示

通过取模软件对图片取模，在 lcdkeyphoto.c 文件中定义了樱花和人的图像点阵数组，如图 4-3 所示。通过定义 ImageShow( void ) 函数可实现花的位置随时间改变而下降，定义 ImageShow\_ren( void ) 函数则可以实现通过键盘左右键控制人的左右位置移动。

```

void INT1_ISR(void) interrupt 2
{
    Delay_ms(100);

    switch(P5)
    {
        case 0xfb:    //左键
            ren_posi=ren_posi-1;
            ren_posi=(ren_posi)%8;
            ImageShow_ren();
            break;

        case 0xfd:    //中间键
            TR0 = 1;  // 开始计时
            Time_num = 1000;
            LcmClear();
            ren_posi=4;
            defen=0;
            zongshu=0;
            ImageShow_ren();
            ImageShow();
            break;

        case 0xfe:    //右键
            ren_posi=ren_posi+1;
            ren_posi=(ren_posi)%8;
            ImageShow_ren();
            break;
    }
}

```

图 4-2 按键中断函数 INT1\_ISR(void)

```

//constant definition
const unsigned char code flower[1024] = { /* 0X00,0X01,0X10,0X00,0X10,0X00, */
0X00,0X00,0X00,0X00,0X00,0X80,0X02,0X40,0X02,0X00,0X18,0X18,0X20,0X04,0X00,0X04,
0X10,0X00,0X00,0X10,0X00,0X00,0X08,0X00,0X00,0X0A,0X70,0X00,0X00,0X00,
};

const unsigned char code ren[1024]= { /* 0X00,0X01,0X10,0X00,0X10,0X00, */
0X00,0X00,0X06,0X00,0X06,0X00,0X32,0X00,0X18,0X60,0X07,0XC0,0X03,0X00,0X03,0X80,
0X01,0X80,0X00,0XC0,0X00,0XF0,0X00,0XA0,0X00,0X80,0X01,0X80,0X01,0X80,0X00,0X00,
};

```

图 4-3 樱花和人的图像点阵数组

## 五、结果展示与分析

游戏初始化后在 LCD 屏幕上显示主菜单，内含游戏名字与简易规则，LED 初始化全部赋 0 值，如图 5-1 所示，按中间键开始游戏后进入 10 秒倒计时，按左右键移动人物接住正在下落的樱花，接到积 1



分，未接到不积分，三组 LED 显示分别显示当前所剩游戏时间、当前得分、当前总共进行游戏局数，如图 5-2 所示。倒计时结束后，LCD 再次回到初始菜单界面，LED 赋 0 值，如图 5-1 所示。

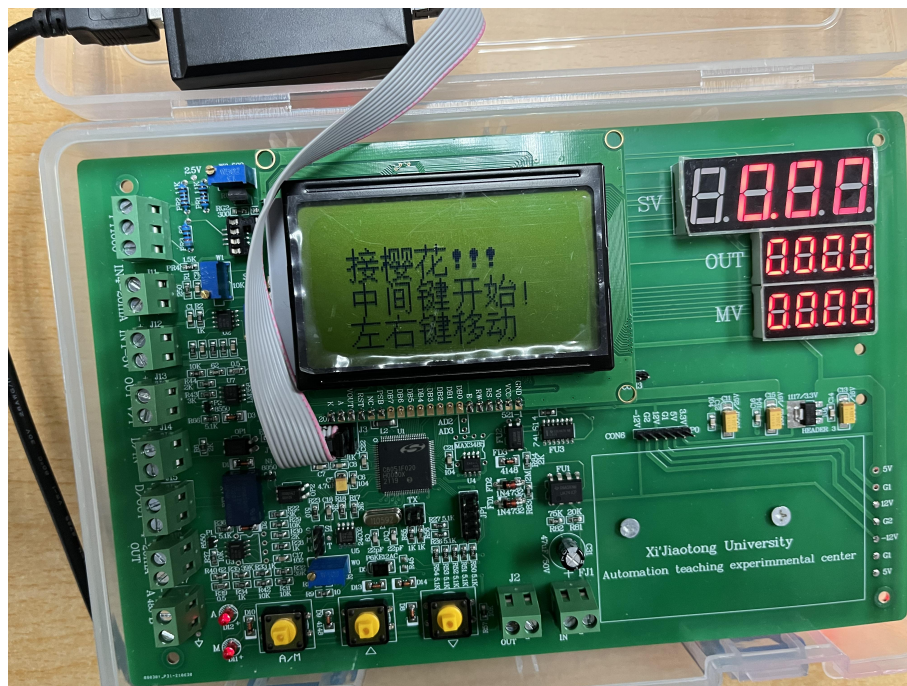


图 5-1 游戏初始菜单界面

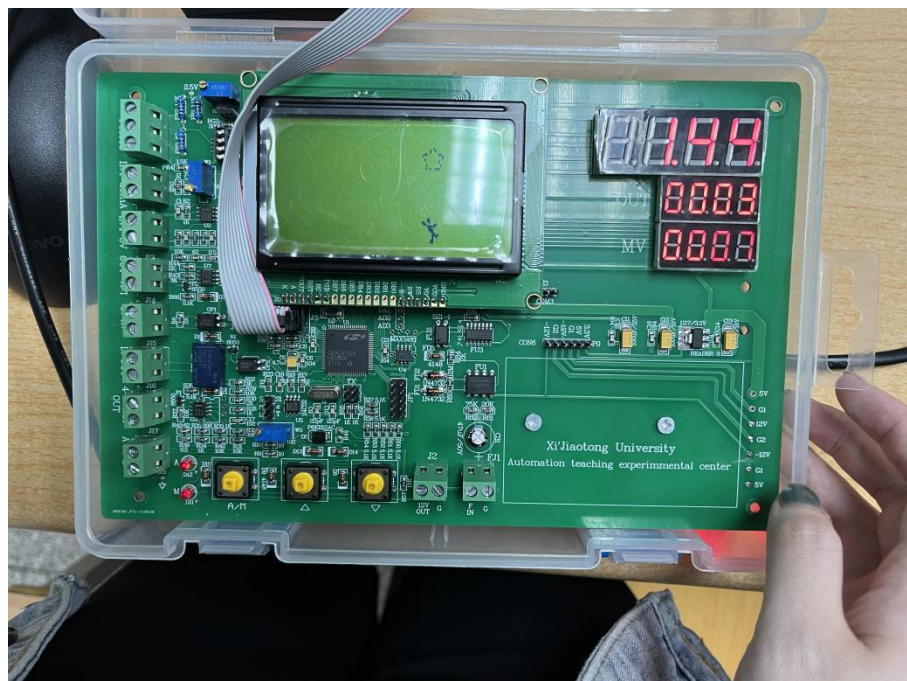


图 5-2 游戏进行界面

## 六、实验总结

通过本次实验，很大程度的提高了我的理解力和动手能力，也拓宽了知识面。在图书馆和网上查阅资料使我了解了更多的资料，也方便了我们以后的使用。在查询的资料中让我对单片机有了更深入的了解，能把学到的知识用活，而不只局限于理论方面。

学习例程让我对整个系统运行与引脚定义有了大致的了解，在自己进行系统编程时，我也学习了例程的方法，充分使用了结构化的思想。这样一来，因为语句较少，程序调试也比较方便，功能模块可以逐一地调试，充分体现了结构化编程的优势。当每个模块都完成时，将其功能互相整合就完成了整体的设计。

经历了从最初的学习例程到设计出小游戏的过程，我从根本上提高了对专业的认识及兴趣，对于我们工科学生来说，学习这些对我们以后的工作有巨大帮助。同时也感谢刘老师在实验中的悉心指导与耐心讲解，让我获益匪浅，也在实践中得以学习与成长。

## 七、参考文献

[1]刘美兰，刘瑞玲，刘源等，电子线路设计训练实验教程. 西安交通大学出版社.

[2]杨国林, c 语言程序设计[J]. 内蒙古大学出版社. 2001. 9

[3]郭天祥, 新概念 51 单片机 C 语言程宁[J]. 电子工业出版社 -2009. 1



## 八、附录



电子线路设计专题  
实验2任务1自动化