

实验二 翻转游戏

【实验目的】

1. 熟悉 μ' nSP™ IDE 环境及在该环境下用汇编和 C 语言编写的应用程序
2. 熟悉简单的 μ' nSP™汇编语言指令
3. 掌握 61 板结合 LED 键盘模块程序设计方法

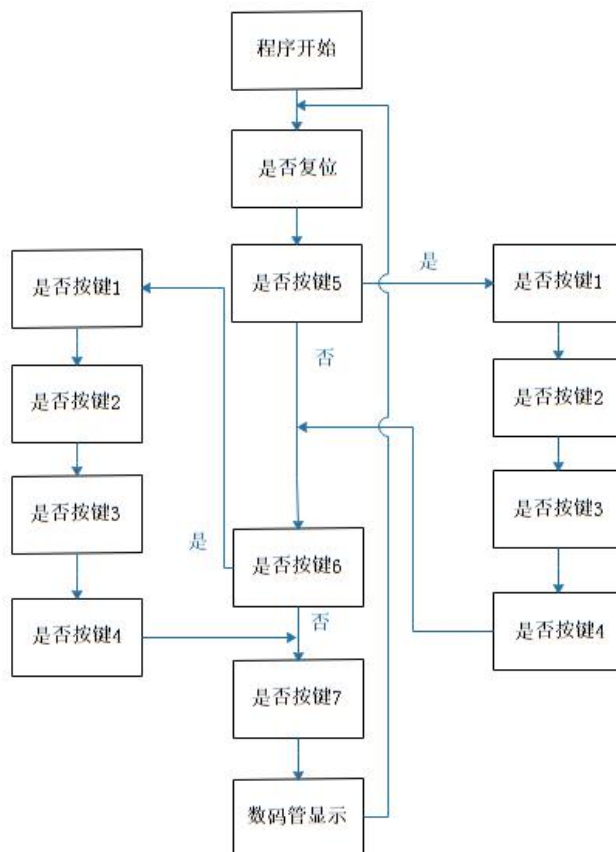
【实验设备】

1. 装有 Windows 系统和 μ' nSP™ IDE 仿真环境的 PC 机一台
2. 61 板一套

【实验说明】

1. 保证 61 板 LED 键盘模块上 KEYTYPE 接在 1*8KEY 的模式下
2. 保证 61 板上 IOA8-15 作为键盘输入口,与 1*8KEY 相连,其中 ROW1 接地; IOB0-7 控制数码管的状态,接 SEG; IOB8-13 分别控制 6 个数码管,接 DIG;

【程序流程图】



【程序调用子程序】

1. 主程序

使用 C 语言编写，具体见代码打包上传文件，各种宏定义见主程序

2. LED 数码管控制子程序

```
//=====//
// 函数名称:  LedDispDig
// 功能描述:  在指定的数码管上显示数字
// 语 法:  void LedDispDig(int Pos,int Dig)
// 输 入:  Pos: 要显示数字的数码管位置,取值范围 1~6
// Dig: 要显示的数字,取值范围 0~9
// 输 出:  无
//=====//

void LedDispDig(int Pos,int Dig)
{
    *P_Watchdog_Clear=1;
    // 初始化 IOB 为同相输出
    *P_IOB_Dir|=LED_SEG+LED_DIG;
    *P_IOB_Attrib|=LED_SEG+LED_DIG;
    *P_IOB_Data=(unsigned)0x0100<<(Pos-1); // 将数字的位置转换为 IOB 高 8 位值,
                                           //选中相应的数码管
    *P_IOB_Data|=DigCode[Dig]; // 将数字转换为编码,作为 IOB 低 8 位输出
}
```

3. 取键值子程序

```
//=====//
// 函数名称:  GetKey
// 功能描述:  等待有键按下并抬起,返回键值,没有去抖处理,一段时间没有按下则返回
//           同时在按键的同时也显示数码管
// 语 法:  unsigned GetKey(void)
// 输 入:  无
// 输 出:  16 位键值
//=====//

unsigned GetKey(int num[6])
{
```

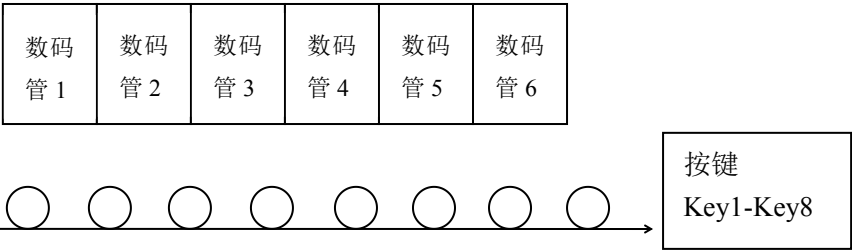
```

unsigned KeyValue, time=0;
int i;
// 初始化 IOA 的相应端口为上拉输入
*P_IOA_Dir&=~KEY_ALL;
*P_IOA_Attrib&=~KEY_ALL;
*P_IOA_Buffer|=KEY_ALL;
//等待有键按下，即有端口变为 0

while(!((*P_IOA_Data&KEY_ALL)^KEY_ALL))
{
    time++;
    if(time>10000)
        break;
    for(i=1;i<=6;i++)
        LedDispDig(i, num[i-1]);
    *P_Watchdog_Clear=1;
}
for(time=0;time<100;time++)
    for(i=1;i<=6;i++)
        LedDispDig(i, num[i-1]);
KeyValue=(*P_IOA_Data&KEY_ALL)^KEY_ALL;
//等待按键抬起
while((*P_IOA_Data&KEY_ALL)^KEY_ALL)
{
    for(i=1;i<=6;i++)
        LedDispDig(i, num[i-1]);
    *P_Watchdog_Clear=1;
}
return KeyValue;
}

```

【补充说明】



1. Key1 控制数码管 1 数字不翻转，数码管 234 翻转
2. Key2 控制数码管 2 数字不翻转，数码管 134 翻转
3. Key3 控制数码管 3 数字不翻转，数码管 124 翻转
4. Key4 控制数码管 4 数字不翻转，数码管 123 翻转
5. Key5 按下之后第一个人开始玩
6. Key6 按下之后第二个人开始玩
7. Key7 按下之后显示两个玩家各操纵了多少步
8. Key8 控制复位
9. 游戏规则：游戏开始时数码管 1234 显示 6666，两名玩家在一定步数之内将四个 6 翻转成 4 个 9，一次只能翻转 3 个数码管的数，谁的翻转次数少谁胜利