# 实验4.7 7279键盘扫描及动态LED显示实验

姓名：孟麟芝 学号：201800121050 实验时间：2020.10.10

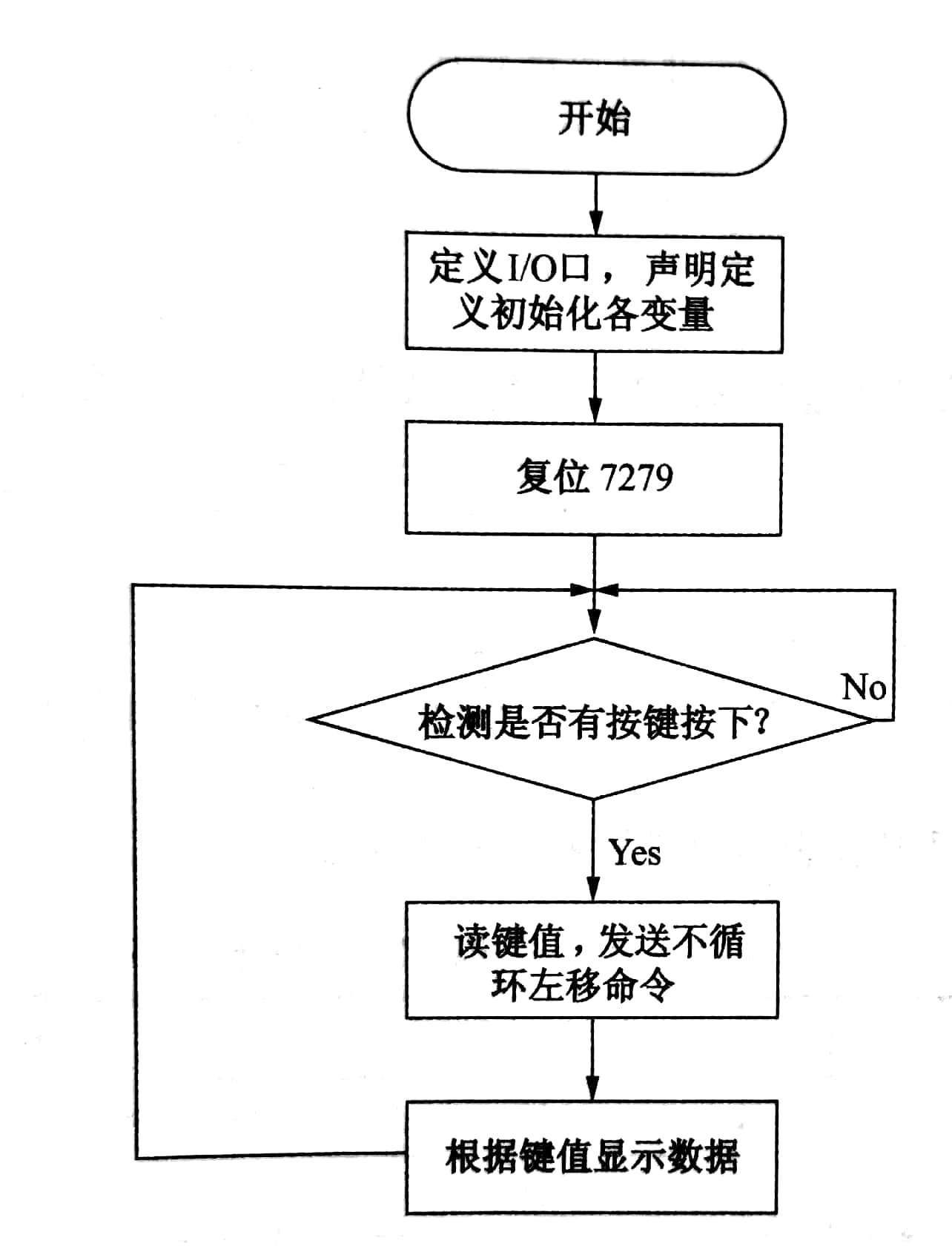
## 实验要求

本实验利用7279进行键盘扫描及动态LED数码管的显示控制，实验原理图如图4.17所示。要求按下某个按键时，所按按键对应的字符显示在最右端LED数码管上，再次按键，原来的内容左移一位，新按下的字符显示在最右端，要注意，左移不是循环左移。

## 实验电路连接

实验线路连接及原理图参见课本P228和P229。

## 实验原理



## 实验过程

（1）关闭实验箱电源，将MCU板插在母版上，按照表4.6将硬件连接好。

（2）在仿真器断电的情况下将仿真头插在MCU板的MCU插座上。将仿真器与PC机的通信口连接好，打开实验箱和仿真器的电源。

（3）建立并运行工程，检测实验结果是否正确。

## 实验源程序

#include <reg51.h> //\*\*\* 函数定义 \*\*\*

#include <stdio.h>

#define uchar unsigned char

#define uint unsigned int

#define RESET 0xa4   // 复位指令

#define RL 0xa1      //左移指令

#define DECODE1 0xc8 //译码方式1

#define READ 0x15    //读键盘指令

void delay1(void);   // 长延时

void delay2(void);

void write7279(uchar, uchar); // 写入到7279

uchar read7279(uchar);        // 从7279读出

void sendByte(uchar);         // 发送一个字节

uchar receiveByte(void);      // 接收一个字节

uchar keyTable[] = {0x1b, 0x13, 0x0b, 0x03, 0x1a, 0x12, 0x0a, 0x02, 0x19, 0x11, 0x09, 0x01, 0x18, 0x10, 0x08, 0x00};

//\*\*\* 变量及I/O口定义 \*\*\*

sbit cs = P1 ^ 0;  // cs 连接于 P1.0

sbit clk = P1 ^ 1; // clk 连接于 P1.1

sbit dio = P1 ^ 2; // dio 连接于 P1.2

sbit key = P1 ^ 3; // key 连接于 P1.3

void main()

{

    uchar key, i, num;

    sendByte(0xa4); //全部复位指令

    while (1)

    {

        if (key == 0)

        { //如果按键按下

            sendByte(0x15);      //读键盘指令

            key = receiveByte(); //接收键盘数据

            for (i = 0; i < 16; i++)

            {

                if (key == keyTable[i])

                {

                    num = i;

                    break;

                }

            }

            sendByte(0xa1);

            write7279(0xc8, num);

            while (key == 0)

                ;

        }

    }

}

void write7279(uchar cmd, uchar num)

{

    sendByte(cmd);

    sendByte(num);

}

uchar read7279(uchar command)

{

    sendByte(command);

    return (receiveByte());

}

void sendByte(unsigned char out\_byte)

{

    unsigned char i;

    cs = 0; //芯片使能

    delay1();

    for (i = 0; i < 8; i++) //分8次移入数据

    {

        if (out\_byte & 0x80) //先传高位

        {

            dio = 1;

        }

        else

        {

            dio = 0;

        }

        clk = 1;

        delay2();

        clk = 0;

        delay2();

        out\_byte = out\_byte \* 2; //数据左移

    }

    dio = 0;

}

uchar receiveByte(void)

{

    uchar i, in\_byte;

    dio = 1; //设置传输口打开

    delay1();

    for (i = 0; i < 8; i++) //分8次读入数据 高位在前

    {

        clk = 1;

        delay2();

        in\_byte = in\_byte \* 2; //数据左移

        if (dio)

        {

            in\_byte = in\_byte | 0x01;

        }

        clk = 0;

        delay2();

    }

    dio = 0;

    return (in\_byte);

}

void delay1(void)

{

    uchar i;

    for (i = 0; i < 0x30; i++)

        ;

}

void delay2(void)

{

    uchar i;

    for (i = 0; i < 8; i++)

        ;

}