

目 录

第一章：开发板简介.....	2
1—1. P8X5X 密码锁时钟开发板的特性简介.....	2
1—2. P8X5X 密码锁时钟开发板的构成和工作原理.....	3
第二章：开发板使用说明.....	4
2—1. 使用简介及入门指导.....	4
2—2. 键盘规划及编码规划，键盘功能，功能扩展.....	5
2—3. 在线下载功能的使用.....	7
第三章：开发板用器件资料及说明.....	8
3—1. AT89C51.....	8
3—2. P89C51RD2.....	9
3—3. AT93C46.....	10
3—4. 74LS47.....	11
3—5. 74LS138.....	11
第四章：开发板器件表附件清单.....	11
4—1. 调试用源程序.....	11
4—2. 原理图.....	附录插页
4—3. 包装清单.....	26
第五章：其它 5 1 类实验板简介.....	26
5—1. 51DEMO I/O 板简介.....	27
5—2. 89C51 数模转换实验板简介.....	28
5—3. 89C2051 实验板简介.....	28

第一章：开发板简介

1-1. P8X5X 密码锁时钟开发板的特性简介

1. 标准的 P8X5X 应用电路设计。
2. 自带程序的在线烧录（自下载）功能。
3. 带 6 位数码管显示 LED。
4. 自带标准 RS232 接口连接电路。
5. 有掉电密码不消失之功能串行 EPROM 应用。
6. 自带 4*4 标准键盘输入，便于学习者掌握键盘输入和程序编写。
7. 有一路声（光）显示的控制输出指示，并可直接在板上的 P2 口上再扩展七路输出控制。
8. 预留扩展空间及接口，可直接驱动或控制用户设备，便于用户直接开发品。

1-2. P8X5X 电子密码锁时钟开发板的构成和工作原理

P8X5X 电子密码锁时钟板，采用了 PHILPS 公司的 P89C51RD2 可在线下载的新科技芯片，用户可以通过板上的跳线设置，直接通过 RS232 口将用户在 PC 机上的程序代码烧录到单片机上直接演示，免去再购置烧录器的重复投资。

本产品集单片机的最小系统应用于一体，在设计其电路的过程中，力求简洁明了，资源共享，方便实用；为单片机的二次开发和初学者提供极大的方便。并预留单片机所有接口的克隆扩展输出接口（一对一输出），便于用户开发成品时直接插拔。

其包含典型的电源电路，复位启动电路，晶体振荡时钟电路，键盘输入电路，RS232 串行通讯电路，串行 EPROM 的在线擦写（可随时更改信息密码），LED 数码管扫描显示电路等，是单片机初学者的最佳范例极品。也是工程开发人员可随手取及的最佳半成品。

工作原理： P8X5X 电子密码锁时钟板像其它 51 类实验板一样，上电复位后进入启动工作状态，程序自动将 LED 数码管上电显示并且清零，等待接收指令状态，当在键盘上输入六位数字的数码时，单片机通过键盘输入口（即定为 P1 口），将键盘的数据读入处理后，将其对应值通过 74LS47 传送数据到 LED 数码管显示，并经 74LS138 译码控制器译码后以扫描的方式逐个连续点亮六个数码管；在此同时单片机还要判断所收到的键盘码是开锁码？如是，则将 P 2.7 口置低电平，此时绿色指示灯亮（或喇叭响），表示给出一个开锁动作的信号，此时程序自动进入时钟显示状态，时钟初始显示为“12:00:00”可直接作为时钟显示；如不是则再判断是否是要更改密码？如是则将键盘的对应值储存于 EEPROM(93c46)中，以备下次开机后使用，EEPROM93C46 是掉电存储密码的设备，也就是说当密码设定后，它就存储在 EEPROM93C46 中了，单片机掉电后密码不会消失；如判断读入的数值均非单片机预设的指令码，则本单片机系统不理它。

4*4 键盘输入部分与 P8X5X 单片机的 P1 口相联接，也是以扫描的方式来读取键位值；而 P2 口是预留功能控制输出部分。

单片机初始内设定密码为“367553”，开机时读取哪一种密码，由 93C46 地址“0”所放的位来决定，若存入位置为“1”，表示 EEPROM 中已存在密码，须到 EEPROM 中读取；若存入不是“1”表示 EEPROM 中未存放密码，必须读取单片机内设定密码。内定密码只是提供给未设定密码的新板产品。

第二章：开发板使用说明

2-1. 使用简介及入门指导

将 P8X5X 电子密码锁时钟板接上电源（DC+5V），电源接通后，红色的电源指示灯点亮，数码管显示应为“000000”，系统正常。输入已设定密码后，再按“D 键”，密码锁驱动部份被打开，此时输出指示绿灯闪烁（或喇叭响），表示锁已被打开；几秒后锁又被关闭（绿灯灭或喇叭不响），单板机自动进入时钟计时状态，数码管显示“120000”，意为 12 点 0 分 0 秒。可以通过按键“C”“B”“A”分别调整“时”“分”“秒”。

显示已存密码: 按住“C”键即可读到已存的密码，初始密码由 8X51 程序中的“TAB[360368]”决定，所以初值为“360368”的密码，或从 EEPROM93C46 中读取上一次已更改的密码；放开则显示“000000”。

清除输入显示: 按“#”键为清除显示器，数码管显示即为“000000”。

设置（更改）密码: 欲更改密码时，在清除显示后，先键入新密码，然后按“*”键，即可建立新密码，并存入

EEPROM93C46 的 1~3 地址中，以备下次启动时用。

进入时钟显示时设置:

按一下“A”键：秒位加一。

按一下“B”键：分位加一。

按一下“C”键：小时位加一。

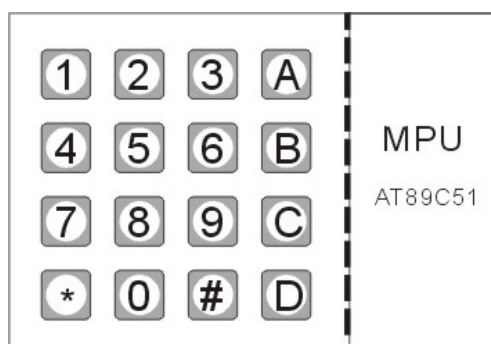
以上键均可连续按不断加一直到校正为止。

如要回到初始密码锁状态只需按一下复位键（Reset）

系统重新启动即可。

2-2. 键盘规划及编码规划，键盘功能，功能扩展

（1） 键盘规划如图：



（2） 编码规划如图：

01	02	03	0C
04	05	06	0D
07	08	09	0E
0A	00	0B	0F

（3） 键盘功能

1. “1--9” 数字键为输入密码键，输入密码有效位每次为 6 位数字。
2. “*” 符号键，为密码输入的确认键。

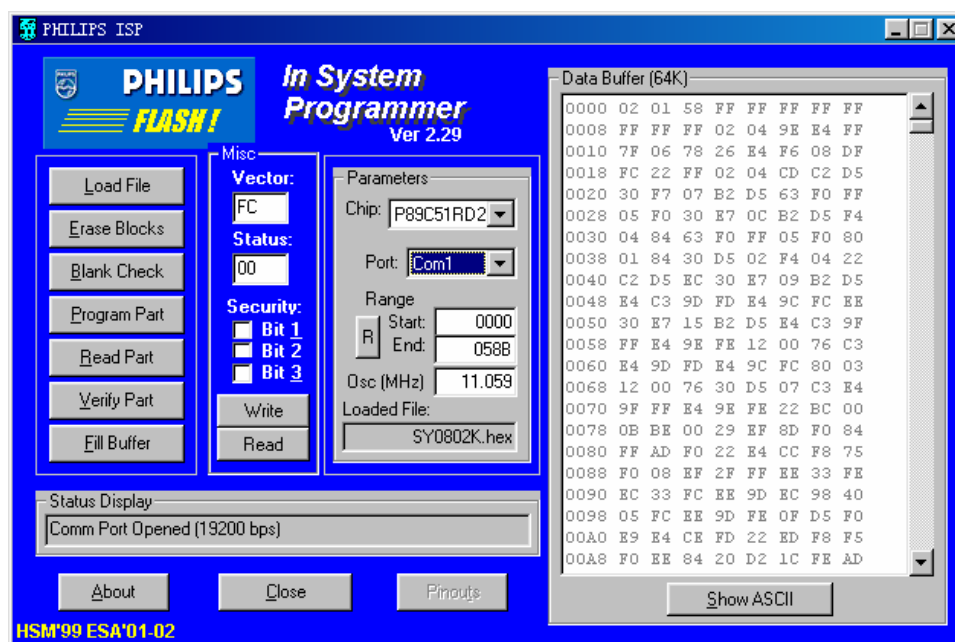
3. “#” 符号键，为显示屏清除键。按下后清除数码管为“000000”。
4. “D” 符号键，为开锁确认键。密码输入正确后按下此键，既可打开驱动，绿色灯闪亮或喇叭响作为指示。
5. “C” 符号键，为密码显示键。按下后即可显示开机密码。
6. “A” “B” “C” 符号键，为机动保留键。用户可在源程序中设定其功能使用。（在时钟显示状态下可调时分秒）

（4） 功能扩展

P8X5X 电子密码锁时钟板，预留输出功能控制部分 P2 口，用户可把须保留类控制设备直接连接到 P2 口，即可扩展多路密码控制设备的功能，同样达到，用户只有输入与设定\密码相符的密码，才能打开设备。因此用户可以用此设备开发成其它电控密码产品或成品。

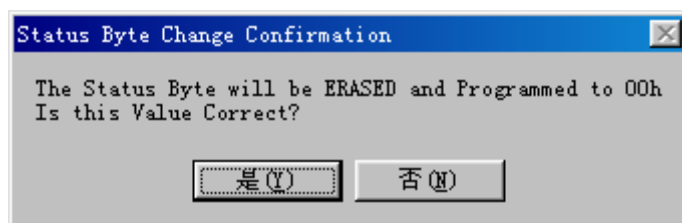
2—3. 在线下载功能的使用：

- （1）将 WINISP 软件正确安装完成后。
- （2）将本下载板的串口与 PC 串口直接连接。
- （3）将下载板上的功能跳线设为 2-3 短路（1-2 短路为运行，2-3 短路为下载）
- （4）按图示极性将+5V 电源加在本下载板上，此时 Power 指示灯亮。
- （5）运行 WINISP 软件，出现如图所示画面。



(6) 进行下载（烧录）设置

- 选择烧录的 IC 芯片的型号：P89C51RD2
- 选择通讯接口：COM1-----COM4
- 选择晶体振荡频率：本电路板上是 11.059Mhz
- 设置芯片状态设定：Status=00（只要在第一次烧录时设定一次即可）
- 写入芯片状态的设定值：点击“Write”键，出现如下画面时请选“是”。



(7) 在“Load File”下装入用户的程序代码（***.HEX 或 ***.BIN）

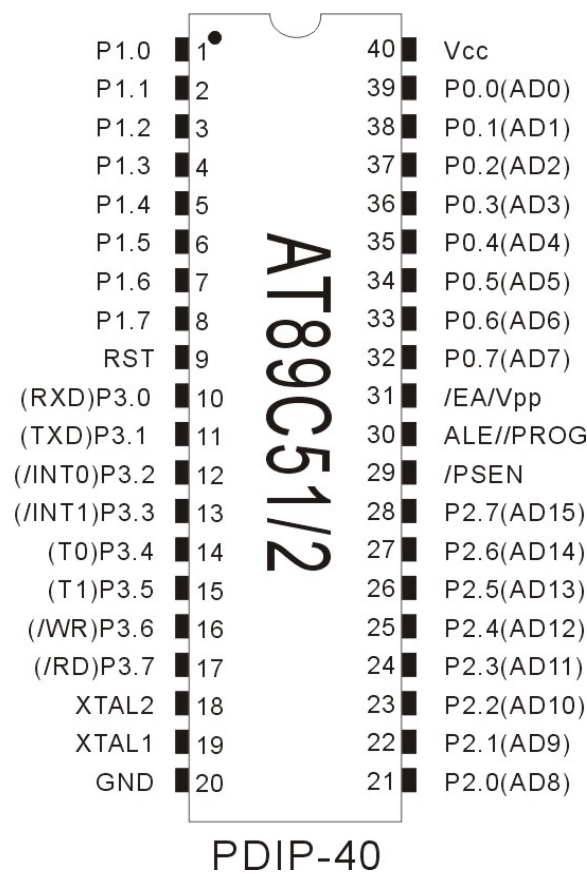
(8) 在“Erase Blocks”下擦除芯片，可选择全擦除或部份擦除（如是新空芯片此项可免除）。

- (9) 在“Program Part”烧录（下载）已装入的用户的程序代码到 P89C51RD 中。
- (10) 当出现“Flash Programming”提示时，恭喜您已掌握了此类芯片的下载功能，烧录成功。
- (11) 将下载板上的功能跳线转设为 1-2 短路，复位后即可进入单片机的自运行。

第三章：开发板用器件资料及说明

3—1. AT89C51

芯片平面引脚功能如图：



- 1 . Compatibie with MCS-51 Products
- 2 . 4 kbytes of in-System Reprogrammable Flash Memory

- 3 . Fully Staic Operation:0 Hz to 24 MHz
- 4 . Three-Level Program Memory Lock
- 5 . 128x8 Bit internal RAM
- 6 . 32 Programmable I/O Lines
- 7 . Two 16-Bit Timer/Cources
- 8 . Six interrupt Sources
- 9 . Programmable Serial channel
- 1 0 . Low Power Idle and Power Down Mod

3—2. P89C51RD2

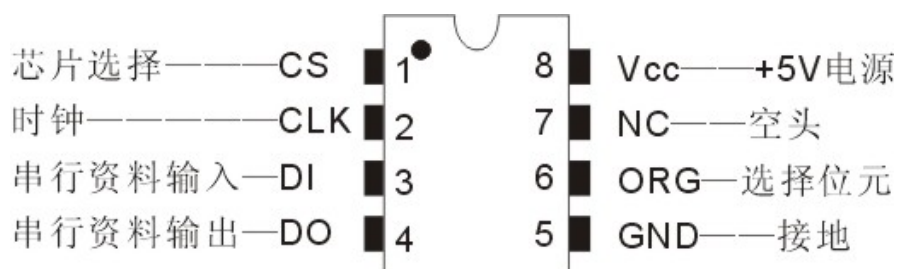
芯片平面引脚功能与上图 AT89C51 相同：（略）

PHILIPS 公司的 P89C51RD2 芯片具有并行可编程 62KB 非易失性存储器，可实现对器件的串行在线编程（ISP）和在应用中编程（IAP）；片内 ROM 中出厂时已固化有加载驱动程序，允许 ISP 通过 UART 将程序代码装如 FLASH 存储器中，而在用户的 FLASH 代码中则不需要加载程序，便于与 AT89C51 系列的兼容替代。

P89C51RD2 芯片是 6 个时钟周期为一个机器周期，因此，其运行速度是普通 51 芯片的二倍（设定可选择）；片内增加 64KB 的 EEPROM 和 1KB 的 RAM；双 DPTR 指针和新增看门狗定时器；该器件是 8051 的派生产品指令系统与 80C51 完全相同。

3—3. AT93C46

芯片平面引脚功能如图：



1. 单 5 伏电源供电。
2. DIP 8 封装。
3. 64 X 16 (ORG 为 HI) 或 128 X 8 (ORG 为 LO)。
4. 全芯片的擦除及单字节的擦除。
5. 时钟频率最高为 250 KHz。

指令格式说明:

1. 读 (READ): 当下达 10XXXXXX 指令后, 将使地址 (XXXXXX) 的资料, 在 CLK=HI, 由 DO 输出。
2. 写 (WRITE): 在写入资料前, 须先下达致能 (EWEN) 指令, 然后下达 01XXXXXX 指令后, 在 CLK 为 HI 时, 会把资料码写入指令的地址 (XXXXXX)。而 DO=0 时, 表示还在进行烧录, 烧录完成时, DO 会转为 HI, 写入动作完成后, 必须再下达写除能 (EWDS) 指令。
3. 清除 (ERASE): 下达清除指令 11XXXXXX, 会将地址 (XXXXXX) 的资料清除。
4. 写致能 (EWEN): 下达 0011XXXX 指令后, 才可进行写入 (WRITE) 动作。
5. 写除能 (EWDS): 下达 0000XXXX 指令后, 才可重复进

行写入（WRITE）动作。

6．芯片清除（ERAL）：下达 0010XXXX 指令后，全部除能。

7．芯片写入（WRAL）：下达 0001XXXX 指令后，全部写入“0”。

3—4. 74LS47

译码电路，将“A0——A3”的十六进制码转换成数码管的“A,B,C,D,E,F,G,H”形式输出，直接驱动数码管。

3—5. 74LS138

数位译码电路，将“A,B,C”的三线地址的八种状态转换成“Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7”分别对应的八位独立输出的形式，来轮番依次驱动数码管。

第四章：开发板器件表附件清单

4—1. 调试用源程序(文件名:sy0306.c)

```
*****
```

```
/**此程序为调试通过的源程序，由南京赛博电子有限公司提供**/
```

```
/**filename:SY0306.C**/
```

```
*****
```

```
#include"at89x51.h"
```

```
#define read    0    /*93c46 读取的识别码 READ=0*/
```

```
#define write   2    /*93c46 写入的识别码 WRITE=2*/
```

```
#define ewen    4    /*93C46 写致能的识别码 EWEN=4*/
```

```
#define ewds 6 /*93C46 写除能的识别码 EWDS=6*/

#define cs RD /*93C46 CS 接脚=8051 RD P3.7*/

#define clk WR /*93C46 CLK 接脚=8051 WR P3.6*/

#define di T1 /*93C46 DI 接脚=8051 T1 P3.5*/

#define d0 T0 /*93C46 DO 接脚=8051 T0 P3.4*/

bit FLAG0=0; /*设定位旗号 20H.0 键盘扫描回应旗号*/

bit FLAG1=1; /*设定位旗号 20H.1 比较密码回应旗号*/

char ADR46,CH,CL,m,ptr,ptr1=0; /*ADR46, 93C46 地址, CH 高位组, CL
低位组, PTR 键盘扫描指标, PTR1 显示器扫描指标*/

char a1=0,b1=100,ptr=0,stand=0,m=0; /*宣告变数 a1 为 TIMER0 的中断次
数, ptr 为 TIMER1 扫描数码管指标*/

static const char tab[16]={0x01,0x02,0x03,0x0c, /*键盘码*/
0x04,0x05,0x06,0x0d,
0x07,0x08,0x09,0x0e,
0x0a,0x00,0x0b,0x0f};

static const char tab1[6]={0x03,0x06,0x07,0x05,0x05,0x03};

/*内定密码"3675530"*/

char HOUR=12;MIN=0;SEC=0; /*宣告变数, 初始时间设定为 08:00:00*/

char DATA[16]; /*按键值存放阵列*/

char data1[6]; /*密码存放阵列*/

char data2[6]; /*宣告存放时, 分, 秒阵列*/

void delay (unsigned int value) /*延时子程序*/

{
```

```
        while (value!=0) value--; /*10US 延时*/

    }

void BCD(void);          /*宣告 BCD 码转换子程序*/
void scan(void);         /*宣告键盘扫描子程序*/
void to9346(char c);     /*宣告 TO93C46 子程序*/
void SDT46(char c);      /*宣告串入子程序*/
RDT46();                 /*宣告串出子程序*/
void clear(void);        /*宣告清除按键存放/显示器阵列 DATA[]副子程序*/
void xch(void);          /*宣告按键存放/显示器阵列 DATA[]右键滚入子程序*/
void open(void);         /*宣告开门比较密码子程序*/
void input(void);        /*宣告读取密码并存入 DATA1[]阵列子程序*/
void set(void);          /*宣告设定密码并存入 DATA1[]阵列子程序*/
void disp(void);         /*宣告显示存放在 DATA1[]阵列的密码子程序*/

*****

main()                   /*主程序*/
{
    P3=0x13;             /*令 93C46 DO=1*/
    TMOD=0x11;           /*TIMER1 工作在案 MODE1*/
    TH1=(65536-3000)/256; /*设定每隔 3000US 扫描一次*/
    TL1=(65536-3000)%256;
    TH0=(65536-10000)/256; /*TIMER0 计数初值设定*/
    TL0=(65536-10000)%256;
    IE=0x88;             /*8 是允许中断，a 是 TIMER0，TIMER1 中断致能位*/
```

```
TR0=1;          /*启动 TEMER0*/

TR1=1;          /*启动 TEMER1*/

BCD();          /*呼叫 BCD 十进制转换子程序，转换起始内定时间*/

input();        /*呼叫读取密码存放在 DATE1[]子程序*/

clear();        /*呼叫清除显示器为 000000 子程序*/

while (1)

{

    do

    scan ();      /*呼叫键盘扫描子程序有按 FLAG0 会为 1*/

    while (FLAG0!=1);    /*判断有按键输入否？*/

    if(tab[ptr]==0x0e) disp(); /*如果是按 “C” 则呼叫显示密码 DISP*/

    else          /*否则往下执行*/

    {

        delay(1000);    /*按钮抗机械反弹跳*/

        while(m==P1);    /*按钮放开否？*/

        delay(1000);

        switch(tab[ptr]) /*是则测试 PTR 键盘扫描计数器指标至 TAB[]取

到的键盘码*/

        {

            case 0x0a: set();    /*是否按 “*” 是则呼叫设定密码 SET*/

                break;          /*跳出此循环*/

            case 0x0b: clear(); /*是否按 “#” 是则呼叫清除显示器 CLEAR*/

                break;          /*跳出此循环*/
```

```
        case 0x0c:    /*是否按“A”，未规划键待用户自己定义*/
            break;    /*跳出此循环*/

        case 0x0d:    /*是否按“B”，未规划键待用户自己定义*/
            break;    /*跳出此循环*/

        case 0x0e:    /*是否按“C”，已侦测过*/
            break;    /*跳出此循环*/

        case 0x0f: open(); /*是否按“D”，是则呼比较密码开门子程序*/
            break;    /*跳出此循环*/

        default:    xch(); /*以上均不是则为数字键呼叫 XCH 作右键滚入*/
            break;    /*跳出此循环*/

    }                /*跳出 SWITCH*/

}

}                /*跳至 WHILE (1) */

}

*****

void BCD(void)        /*BCD 码转换子程序*/
{
    char c1;          /**/

    c1=SEC;            /*除 60 取余数为 SEC 秒*/

    SEC=c1%60;         /*除 60 取余数为 SEC 秒*/

    c1=MIN+c1/60;       /*"分"加上秒的进位*/

    MIN=c1%60;          /*除 60 取余数为 MIN 分*/

    HOUR=(HOUR+c1/60)%24; /*"时"加上分的进位除 24 取余数为 HOUR 时*/
```

```
DATA[0]=(SEC%10);    /*取秒的个位数存入 data2[0]*/
DATA[1]=(SEC/10);    /*取秒的十位数存入 data2[1]*/
DATA[2]=(MIN%10);    /*取分的个位数存入 data2[2]*/
DATA[3]=(MIN/10);    /*取分的十位数存入 data2[3]*/
DATA[4]=(HOUR%10);   /*取时的个位数存入 data2[4]*/
DATA[5]=(HOUR/10);   /*取时的十位数存入 data2[5]*/

}
```

```
void to9346(char c)
```

```
{
    char cl;

    cs=1;di=1;clk=1;    /*起始位 DI=1，CS=1，93C46 致能*/

    delay(3);

    clk=0;

    delay(3);

    switch(c)            /*侦测 93C46 识别码的指令*/
    {
        case 0: cl=ADR46+0x80; /*0=READ 读指令 10 (A5-A0) =地址+运算码*/

            SDT46(cl);        /*呼叫串入子程序写指令至 93C46*/

            CH=RDT46();        /*呼叫串出子程序传回读取的高位组*/

            CL=RDT46();        /*呼叫串出子程序传回读取的低位组*/

            cs=0;            /*93C46 除能*/

            break;          /*跳出此循环*/
    }
}
```



```
case 2: cl=ADR46+0x40; /*2=WRITE 写指令 01 (A5-A0) =地址+运算码*/
        SDT46(cl);      /*呼叫串入子程序写指令至 93C46*/
        SDT46(CH);      /*呼叫串出子程序传回读取的高位组*/
        SDT46(CL);      /*呼叫串出子程序传回读取的低位组*/
        cs=0;           /*93C46 除能*/
        break;          /*跳出此循环*/
case 4: SDT46(0x30);     /*4=EWEN 写致能指令 0011XXXX*/
        cs=0;           /*93C46 除能*/
        break;          /*跳出此循环*/
case 6: SDT46(0x00);     /*6=EWDS 写除能指令 0000XXXX*/
        cs=0;           /*93C46 除能*/
        break;          /*跳出此循环*/
    }
}
```

```
void SDT46(char c)      /*串入子程序*/
{
    char c1;
    for(c1=0;c1<8;c1++) /*8 位*/
    {
        if ((c&0x80)!=0) di=1; /*侦测 C 的 BIT7 并写入 DI*/
        else di=0;
        c=c<<1;               /*资料码左移一位, 准备串入下一个位*/
    }
}
```

```
        clk=1;                /*产生时钟 CLK*/

        delay(2);

        clk=0;

        delay(2);

    }

}

RDT46()                        /*串出子程序*/

{

    char c1,c2=0;

    for(c1=0;c1<8;c1++)        /*8 位*/

    {

        clk=1;                /*产生时钟 CLK*/

        delay(2);

        clk=0;

        delay(2);

        c2=c2<<1;             /*读取存放暂存器资料左移一位*/

        if(d0==1) c2=c2|0x01;  /*如 DO=1，则 C2 BIT0 存入 1*/

        else c2=c2&0xfe;      /*否则 C2 BIT0 存入 0*/

    }

    return(c2);                /*将读取值传回给呼叫程序*/

}

*****

void clear(void)                /*清除显示 DATA[]为 00 子程序*/
```

```
{  
    char c;                /*宣告变数*/  
    for(c=0;c<6;c++)        /*将 00 存入显示阵列 DATA[]*/  
    {  
        DATA[c]=0x00;  
    }                    /*返回主程序*/  
}  
  
*****  
  
void open(void)            /*开锁比较密码子程序*/  
{  
    char c;                /*宣告变数*/  
    FLAG1=1;                /*设开门旗号为 1*/  
    for(c=0;c<6;c++)        /*比较六个密码*/  
    {  
        if (DATA[c]!=data1[c]) /*按键值 DATA[]与密码 TAB[]比较是否相等*/  
            FLAG1=0;  
            if (FLAG1==0)        /*FLAG1=0 表示不相等*/  
                break;        /*跳出此循环*/  
    }  
    if(FLAG1==1) /*如 FLAG1=1 表示密码相等,令电锁 P2.0 置 0 动作*/  
    {  
        P2_0=0;                /*P2.0 置 0,使绿色指示灯亮*/  
        delay(2000);            /*延时*/  
    }
```

```
clear();          /*清除屏幕*/

IE=0x8a;  /*8 是允许中断，a 是 TIMER0，TIMER1 中断致能位*/

BCD();          /*呼叫 BCD 十进制转换子程序，转换起始内定时间*/

while (1)

{

    do

    scan ();          /*呼叫键盘扫描子程序，有按时 FLAG0 会为 1*/

    while (FLAG0!=1); /*判断有按键输入否？无时返回 scan()*/

    delay(1000);      /*按钮防抖动延时，抗机械反弹跳*/

    while(m==P1);     /*按钮放开否？*/

    switch(tab[ptr])  /*是则测试 PTR 键盘扫描计数器指标至 TAB[]取
到的键盘码*/

    {

        case 0x0c: SEC++;

            BCD();      /*是否按"A",是则呼叫 BCD 码转换子程序*/

            break;      /*跳出此循环*/

        case 0x0d: MIN++;

            BCD();      /*是否按"B",是则呼叫 BCD 码转换子程序*/

            break;      /*跳出此循环*/

        case 0x0e: HOUR++;

            BCD();      /*是否按"C", 是则呼叫 BCD 码转换子程序*/

            default:break; /*跳出此循环*/

    }

    /*跳至 WHILE (1) */
```

```
    }  
  
}  
  
}  
  
*****  
  
void input(void)          /*读取密码子程序*/  
{  
  
    char c,c1=0;  
  
    ADR46=0;              /*读 93C46 0 地址的值*/  
  
    to9346(read);  
  
    if(CH==1)             /*如为 1, 表示 93C46 存有密码,读取 93C46 的密码*/  
    {  
  
        for(c=1;c<4;c++)  /*读 1~3 地址六个密码*/  
        {  
  
            ADR46=c;       /*93C46 地址*/  
  
            to9346(read);  /*呼叫读取 93C46*/  
  
            data1[c1]=CH;  /*读取高位组存入密码阵列 DATA1[]*/  
  
            c1++;  
  
            data1[c1]=CL;  /*读取低位组存入密码阵列 DATA1[]*/  
  
            c1++;  
  
        }  
  
    }  
  
    else                  /*否则 93C46 未存有密码,至 TAB1[]读取内定密码*/  
    {
```

```
        for(c=0;c<6;c++) /*读取 6 个内定密码存入密码 DATA1[]中*/
            data1[c]=tab1[5-c];
    }
}

*****

void set(void)
{
    char c,c1=0;

    for(c=1;c<4;c++) /*写入 93C46 1~3 地址 6 个密码*/
    {
        ADR46=c; /*93C46 地址*/
        to9346(ewen); /*93C46 写致能*/
        data1[c1]=DATA[c1]; /*按键阵列 DATA[]存入密码阵列 DATA1[]*/
        CH=DATA[c1]; /*按键值存入准备写入 93C46 高位组*/
        c1++;
        data1[c1]=DATA[c1]; /*按键阵列 DATA[]存入密码阵列 DATA1[]*/
        CL=DATA[c1]; /*按键值存入准备写入 93C46 低位组*/
        c1++;
        to9346(write); /*写入 93C46*/
        to9346(ewds); /*93C46 写除能*/
        delay(1000); /*延时 1000MS*/
    }

    ADR46=0; /*93C46 0 地址写入 01*/
```

```
        to9346(ewen);    /*93C46 写致能*/

        CH=0x01;    /*按键值存入准备写入 93C46 高位 01*/

        CL=0x00;    /*按键值存入准备写入 93C46 低位组 00*/

        to9346(write);    /*写入 93C46*/

        to9346(ewds);    /*93C46 写除能*/

        delay(1000);    /*延时 100MS*/

        clear();    /*清除荧屏显示*/

    }

    *****

void disp(void)    /*显示密码子程序*/

{

    char c;    /*宣告变数*/

    ptr=0xff;

    for(c=0;c<6;c++)    /*将密码存放在阵列 DATA[] 存如显示器存放阵列

DATA[]*/

    {

        DATA[c]=data1[c];

    }

    while(m==P1);    /*判断按钮放开否，没有则在此等待*/

    clear();    /*有则呼叫清除显示器*/

}    /*返回主程序*/

    *****

void scan(void)    /*扫描键盘子程序*/
```

```
{  
    char a1=0xf7,i;          /*A1=0XF7 列扫描初值，I 行*/  
        FLAG0=0;            /*设按键回应旗号为 0，键盘扫描计数指标为 0*/  
        ptr=0;  
    for(i=0;i<4;i++)        /*键盘 4 个扫描列*/  
    {  
        P1=a1;              /*列扫描输出，读入 P1 存入 M，以便侦测行与侦  
测按键是否放开*/  
        m=P1;  
        switch(m&0xf0)      /*取行的高 4 位，侦测那一行被按*/  
        {  
        case 0x70: ptr=i*4;  /*第一行被按否？是则扫描指标=列 X4*/  
            FLAG0=1;        /*是则设 FLAG0=1 表有按键输入*/  
            break;          /*跳出此循环*/  
        case 0xb0: ptr=i*4+1; /*第二行被按否？是则扫描指标=列 X4+1*/  
            FLAG0=1;        /*是则设 FLAG0=1 表有按键输入*/  
            break;          /*跳出此循环*/  
        case 0xd0: ptr=i*4+2; /*第三行被按否？是则扫描指标=列 X4+2*/  
            FLAG0=1;        /*是则设 FLAG0=1 表有按键输入*/  
            break;          /*跳出此循环*/  
        case 0xe0: ptr=i*4+3; /*第四行被按否？是则扫描指标=列 X4+3*/  
            FLAG0=1;        /*是则设 FLAG0=1 表有按键输入*/  
        default: break;     /*跳出此循环*/  
    }
```



```
        }

        if(FLAG0==1)break;    /*不为 1，则扫描列右移，扫描下一列*/

        a1=a1>>1|0x80;    /*高位补 1，由于 P1.7~P1.4 未接+5V，而是由指
令加载高电平*/

    }

}    /*返回主程序*/

*****

void service_int0 () interrupt 1 using 1    /*TIMER0 中断子程序--秒计时器*/
{

    TH0=(65536-9200)/256;    /*重设 TIMER0 计数值*/

    TL0=(65536-9200)%256;

    b1--;    /*中断次数减 1*/

    if (b1==0)    /*中断次数完成否，是则表 1 秒到了*/

    {

        b1=100;    /*重设中断次数*/

        SEC+=1;    /*秒 SEC 加 1*/

        BCD();    /*呼叫 BCD 码转换子程序*/

    }

}    /*返回主程序*/

*****

void service_int1 () interrupt 3 using 2    /*TIMER1 中断子程序*/
{

    char ptr1;    /*宣告扫描指标变数*/
```

```
TH1=(65536-3000)/256;          /*每隔 4000US 扫描一次*/  
TL1=(65536-3000)%256;  
while (ptr1>5) ptr1=0;          /*PTR1>5 表示 6 个数码管都已经扫描过*/  
P0=(ptr1<<4)|DATA[ptr1];        /*将扫描值左移至高 4 位再加上资料码*/  
    ptr1++;                      /*扫描指标加 1*/  
}  
                                /*返回主程序*/
```

4—2. 电路原理图

请见附录（一）

4—3. 结构示意图

请见附录（二）

4—4. 包装清单

- a. 主机板 * 1 块
- b. 使用说明书 * 1 本
- c. 软件盘 * 1 张（包含软件）
 - a) SY0306.C 的源程序
 - b) SY-0306 板电源理图
 - c) UltraEdit 编辑软件
 - d) A51,C51 的编译连接软件
 - e) MEDWIN 仿真软件
 - f) WINISP 下载软件

第五章：赛博其它 51 类实验板简介

5—1. 51DEMO I/O 板简介

51XXDEMO 板结构简单,使用方便,51XXDEMO 板的 P0, P1, P2, P3 各口已直接联接上四个七段 LEO 数码管显示模块,并配有演示示范程序,将板上单片机取下,插上 51 系列仿真器的仿真头后,可以直接测试仿真器 P0, P1, P2。P3 端口的输出状态。是用户用来检测仿真器输出端口的最好工具,板上并扩展了液晶显示的模式。适用于初学者及数学试验和直接用于产品开发。此产品还提供了方便实用的双电源供电插座,DC+5V 或 AC8-12V 电源均可。详情请见该产品的资料说明。

5—2. A/D-8X51 数模转换开发实验板简介

A/D-8X51 模数转换板像其它 51 类的开发实验板一样,上电复位后进入启动工作状态,程序自动的控制 AD0804 模数转换器,将 0—5 伏的模拟电压转换成数字电压,供 8X51 单片机处理后送 LED 数码管显示;在本案中是采用了 AD590 的温度传感器将变化的温度转换成变化的电压,再将电压的变化值转换成对应的温度显示出来;如测到的温度在设定的上下限之间,绿色的正常指示灯亮,反之温度大于或小于设定温度的上下限,红色(HI/LOW)指示灯将分别闪亮,表示报警。

板上的配制有:

- a. 标准的 8X51 应用电路设计。
- b. 带 4 位数码管显示 LED,可显示 3 位数电压或转换成的温度。
- c. 自设 4 个功能设置按键,或用于上下(加减)调整之

用。在板上分别为是 MODE, SET, UP, DOWN。

- d. 有方便实用的双电源供电插座, DC+5V 或 AC9V-12V 电源均可。
- e. 有三路 LED 显示的控制输出指示, 并可直接在板上再扩展输出控制。
- f. 预留 8X51 所有端口的连接插口, 便于用户二次开发成成品。

详情请见该产品的资料说明 (型号为: SY-1032)。

5-3. 89C2051 实验板简介

AT89C2051 实验板, 输入电压 DC8-12V 工作电源, 自带极性定位及稳定电路, 采用 RS232, 9 针 LCD 显示器插座为用户使用 LCD 显示器监视系统状态提供了方便; CON2 为 AT89C2051 管脚标准信号引出, 着为扩展 AT89C2051 的外部接口提供了方便, LED1-1ED8 为实验板演示程序用指示灯, 通电后 LED 为一闪一闪的顺序发光状况, 如果插上 LCD 显示器则会显示相应的字符。该实验板在开发产品, 教学实验等方面有很的的用途, 此产品具体性能如下:

1. 与 MC-51 产品完全兼容
2. 串行接口输出
3. LCD 显示接口输出
4. AT89C2051 标准管脚引出
5. DC8V-12V 工作电源

*****公司简介见软件盘中电子板文件*****