

目 录

第一章：开发板简介.....	2
1-1. SY_0406 数码显示密码锁时钟开发板的特性简介.....	2
1-2. SY_0406 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一(P89C51RD2xx).....	7
2-4. 在线下载功能的使用二(AT89S52xx).....	7
第三章：开发板用器件资料及说明.....	8
3-1. AT89S51(2).....	8
3-2. P89C51RD2xx.....	9
3-3. AT24Cxx.....	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
4-2. 包装清单.....	28
第五章：其它 5 1 类实验板简介.....	30
5-1. 51DEMO I/O 板简介.....	30
5-2. 89C51 数模转换实验板简介.....	30
5-3. 89C2051 实验板简介.....	31

5-4. 流水灯控制器（12路）32

5-5. ISP 下载线（选配自购件）32

*****公司其它产品简介见软件盘中电子版文件*****

第一章：开发板简介

1-1. SY_0406 数码显示密码锁时钟开发板的特性简介

1. 标准的 8X5X 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. SY_0406 开发板的构成和工作原理

SY_0406 开发板板，带自下载功能。

在采用 PHILIPS 公司的“P89C51RD2”芯片时，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。

在采用了 ATMEAL 公司的“AT89S51/52”芯片时，用户可以通过板上 ISP 插口通过下载电缆（另购下载电缆）与 PC 机的并行口相连即可将 PC 机上的程序代码烧下载到单片机上。此芯片有良好的市场性价比。是单片机初学者的首选。

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，资源共享开放，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，克隆的扩展输出接口（一对一输出），便于用户开发成品时直接插拔。其电路中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, 是单片机初学者的精典范例。也是工程开发人员可随手取及的最佳半成品。

工作原理: SY_0406 开发板是原 SY_0306 板升级更新产品, 仍有的像其它 51 类实验板一样的功能, 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依次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 2.0 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, 并可通过键 “A” / “B” / “C” 分别来调整时钟值; 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(24C01)中, 以备下次开机后使用, EEPROM24C01 是掉电存储密码的设备, 也就是说当密码设定后, 它就存储在芯片中了, 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, 则本单片机系统不预执行它。

4*4 键盘输入部分与 89X5X 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 “83675530”, 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。

通用串口的连接, 在初始状态下, 可接收从 PC 口传来的数字, 并将其收到的数字在 LED 上显示出来; 在开锁状态进入时钟显示下, 会将当前的显示时间通过

串口发送到 PC 机上，供终端显示用。

第二章：开发板使用说明

2-1. 使用简介及入门指导

将 SY_0406 开发板接上电源（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: 00: 00。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[888888]”决定，所以初值为“888888”的密码，或从 EEPROM24C01 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 EEPROM24C01 的 1~5 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：小时位加一。

按一下“B”键：分位加一。

按一下“C”键：秒位加一。

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）

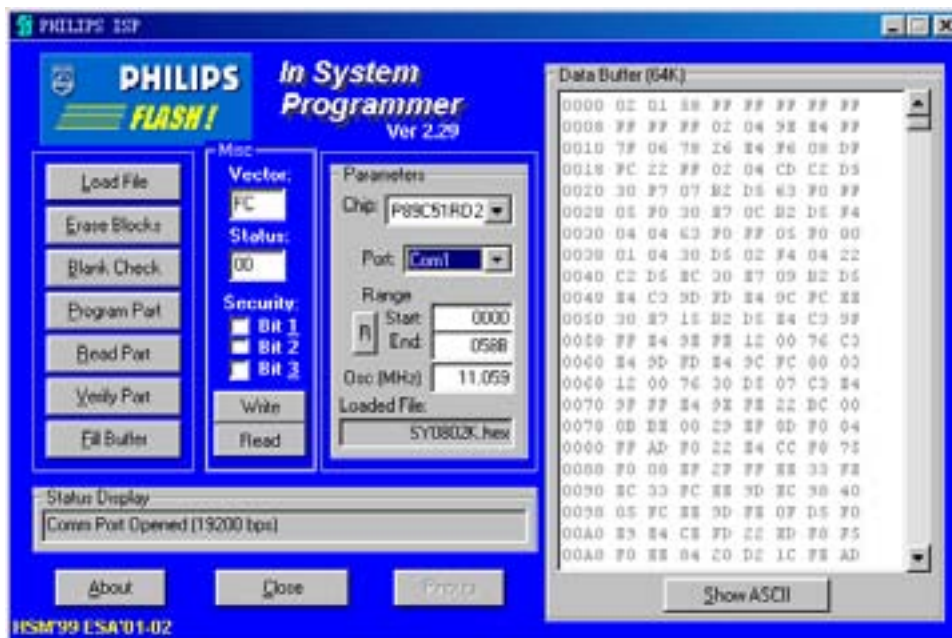
(4) 功能扩展

SY_0406 开发板，预留输出功能控制部分 P2 口，用户可把需保留类控制设备直接连接到 P2 口，即可达到扩展成多路密码控制设备的功能，用户只有输入与设定的密码相符的数字码，才能打开设备，或通过更改程序来达到用不同的密码

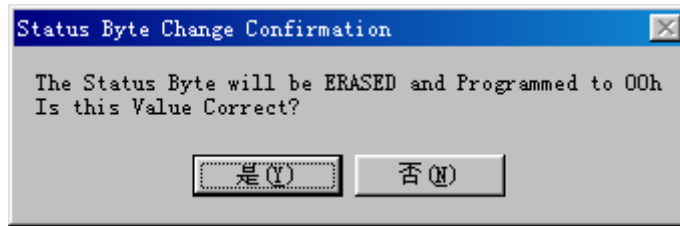
来打开不同的控制输出。因此用户可以用此设备开发成其它电控密码产品或成品。

2-3. 在线下载功能的使用一（P89C51RD2）:

- (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
 - a. 选择烧录的 IC 芯片的型号：P89C51RD2
 - b. 选择通讯接口：COM1-----COM4
 - c. 选择晶体振荡频率：本电路板上是 11.059Mhz
 - d. 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）
 - e. 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。

2-4. 在线下载功能的使用二（AT89S52）:

- (1) 将 Easy52Pro 软件正确安装完成后。
- (2) 下载板上的功能跳线仍为 1-2 短路的运行状态。
- (3) 将下载电缆的连接直接插在 SY_0406 板上的 ISP 接口上，下载电缆的另一端接在 PC 机的并行口上。
- (4) 按图示极性将+5V 电源接在本开发板上，此时板上 Power 指示灯和下载线上 Led 指示灯都亮。
- (5) 运行 Easy51Pro 软件，出现如图所示画面。



(6)

进行下载（烧录）设置

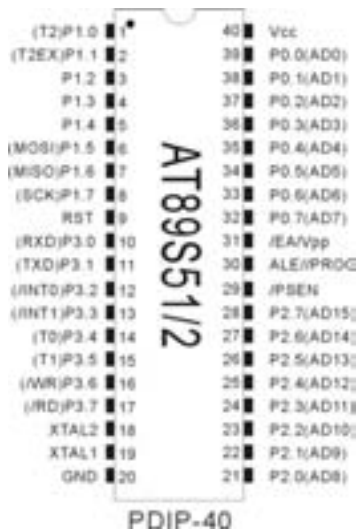
- a. 选择烧录的 IC 芯片的型号：AT89S52
- b. 在设置状态下选择“使用 Easy ISP 下载线”方式。
- c. 退出软件环境后再次进入即可装载烧录代码进行下载烧录芯片。

将开发板复位后即可进入单片机的自运行状态。

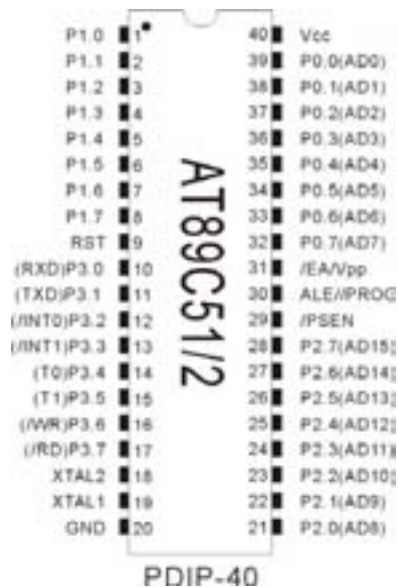
第三章：开发板用器件资料及说明

3—1. AT89S52

芯片平面引脚功能如图：



AT89C51 芯片引脚功能和 P89C51RD2 完全相同如图：



- 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 . 256x8 Bit internal RAM
- 6 . 32 Programmable I/O Lines
- 7 . Three 16-Bit Timer/Cources
- 8 . Six interrupt Sources

9. Programmable Serial channel

1 0. Low Power Idle and Power Down Mod

3—2. P89C51RD2

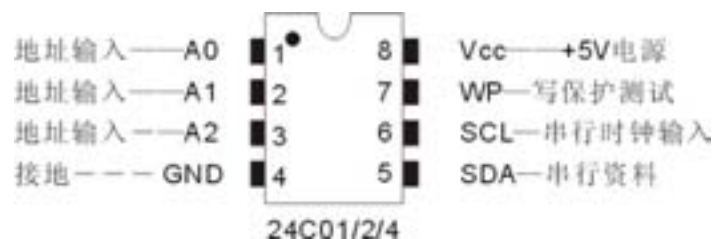
芯片平面引脚功能与上图 AT89C51 相同：（略）

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. AT24C01

芯片平面引脚功能如图：



对 24C01/2/4 的读写操作的方法和使用中应注意的问题。24C04 的 1—3 脚分别为器件编址端 A0、A1、A2（用于与系统中的同类器件编码），4 脚为电源地，5 脚为 IIC 总线的数据线 SDA，6 脚为 I2C 总线的时钟 SCL，7 脚为测试输入端，在系统中接地，8 脚为电源。要正确的对 24C04 进行读写操作首先要了解这两个问题 1、器件寻址方法 2、翻卷现象及处理方法。器件的寻址方法：（24C 系列器件代码为 1010）。

R/W 位为读和写状态位，为 0 时写；为 1 时读。

电路中 IIC 器件采用的是 24C01, 由于 24C01 的容量为 128 字节, 如要对 24C01 进行读操作时, 寻址字节就由: 器件代码 (1010) + 器件 A0、A1、A2 脚的状态 (00) (状态与访问的单元有关, 大于 256 字节时为 1 否则为 0) + R/W (读 0/写 1) 组成。

翻卷现象及处理方法: 在此省略。本着让读者在不了解 IIC 的工作原理的情况下能拿来就用的原则, 简单时序和使用方法如下:

A. 指定地址写 N 个字节:

启动总线->发送器件寻址字节->回答->发送寻址地址->回答->发送第一个数据->回答->发送第二个数据->回答->....->发送第 n-1 个数据->回答->停止总线。

B. 指定地址读 N 个字节: (数据寻址地址放在数据中的第一位字节。)

启动总线->发送器件寻址字节->回答->发送数据寻址地址->回答->停止总线
+启动总线->读第一个数据->回答->读第二个数据->回答->.....->读第 n-1 个数据->回答->停止总线。

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 ”形式字型笔划输出, 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 (本例中只用了六

位), 来依次驱动数码管。

第四章：开发板器件表附件清单

4—1. 调试用源程序(文件名: SY0406.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:SY0406.C****
/****是在 sy0306 的功能上增加了 AT89S52 的在线下载功能
/****IIC 总线芯片改为 24C01
/****串口功能：
// 1.在初始状态时，在 PC 机上向串口发送“A0 01 02 03 04 05 06”会在
//在数码管上显示出"123456"表示接收成功(其中 A0 为识别码)。
// 2.在进入开锁状态时，每隔 1 秒，将数码管的显示向板上串口（PC 机）
//上向发送一次，如“01 02 00 00 00 01” / “01 02 00 00 00 02”——。
/****定时器 0--设定数码管定时扫描。
/****定时器 1--设定时钟显示计时。
/****定时器 2--设定串行通讯波特率。
/****
#include"at89x52.h"
#include<intrins.h>
#define uchar unsigned char
#define uint unsigned int
#define nop _nop_() /* 定义空操作指令 */
sbit SCL=P3^4; /*24C01 CSL 接脚=89C51 T0 P3.4*/
sbit SDA=P3^5; /*24C01 SDA 接脚=89C51 T1 P3.5*/
bit FLAG0=0; /*设定位标号 FLAG0=1 键盘扫描回应*/
bit FLAG1=1; /*设定位标号 20H.1 比较密码回应旗号*/
char m,ptr=0,ptr1=0; /*PTR 键盘扫描指标，ptr1 显示器扫描指标*/
char a1=0,b1=100,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*/
bit TSendAddress(uchar slaaddress,uchar subaddress,uchar * s,uchar no);
/* 向有子地址器件写入 6 字节数据函数 */
bit TRcvAddress(uchar slaaddress,uchar subaddress,uchar * s,uchar no);

```

```

/* 向有子地址器件读取 6 字节数据函数 */
bit acknow; /*应答位 acknow=1 正常响应; acknow=0 未响应 */
static const char tab[16]={0x01,0x02,0x03,0x0c, /*键盘码 00,01,02,03,*/
                          0x04,0x05,0x06,0x0d, /*键盘码 04,05,06,07,*/
                          0x07,0x08,0x09,0x0e, /*键盘码 08,09,0a,0b,*/
                          0x0a,0x00,0x0b,0x0f}; /*键盘码 0c,0d,0e,0f*/
static const char tab1[6]={0x08,0x08,0x08,0x08,0x08,0x08}; /*内定密码"888888"*/
char HOUR=12;MIN=0;SEC=0; /*变量数, 初始时间设定为 12:00:00*/
char disple[8]; /*显示值存放阵列*/
char bufdata[9];
char bufuart[6]; /*串口接收值存放阵列*/
void delay (unsigned int value) /*延时副程式*/
{
    while (value!=0) value--; /*10US 延时*/
}
void BCD(void); /*BCD 码转换副程式*/
void scan(void); /*键盘扫描副程式*/
void clear(void); /*清除按键存放/显示器阵列 disple[]副副程式*/
void open(void); /*开门比较密码副程式*/
void read24c02(); /*读 24c02 的存储值*/
void write24c02(); /*写 24c02 的存储值*/
void IICSAVE(void); /*设定密码并存入 buft[]阵列副程式*/
void disp(void); /*显示存放在 disple[]阵列的密码副程式*/
void tserial(void); /*串口 serial 发送副程式*/
void rserial(void); /*串口 serial 接收副程式*/
void temp(void);
void Start(); /* 起动总线函数 */
void Stop(); /* 结束总线函数 */
void SendByte(uchar c); /* 8951 发数据或地址给 24C01 字节数据发送函数 */
uchar RcvByte(); /* 8951 从 24C01 读数据字节数据接收函数 */
void Ack(bit a); /* 主机 8951 应答子函数 */
void xch(void); /*宣告按键存放/显示器阵列 disple[]右键滚入副程式*/
/*****
main() /*主程式*/
{
    TMOD=0x21; /*TIMER0 工作在案 MODE1,16 位定时器*/

```

```

/*TIMER1 工作在 MODE2,自动重新装载模式*/
/*方式寄存器 TMOD_____*/
/*| 定时器 1 | 定时器 1 |*/
/*| GATE|C/T| M1 | M0 | GATE|C/T| M1 | M0 |*/
/*|_____*/
TH0=(65536-3000)/256; /*设定每隔 3000us 中断扫描一次数码管*/
TL0=(65536-3000)%256;
TR0=1; /*启动 TEMER0*/
T2CON=0x38; /*定时器 2 工作控制寄存器 T2CON_____*/
/*| TF2|EXF2|RCLK|TCLK|EXEN2|TR2|C/T2|CP/RL2|*/
/*|_____*/
TL2=0xfd; /*设定传输波特率 9600, 晶体振荡器=11.059Mhz*/
TH2=0xff; /*设定传输波特率 9600, 晶体振荡器=11.059Mhz*/
RCAP2L=0xdc; /*设定传输波特率 9600, 晶体振荡器=11.059Mhz*/
RCAP2H=0xff; /*设定传输波特率 9600, 晶体振荡器=11.059Mhz*/
TH2=256-(28800/9600); /*设定传输波特率 9600, 晶体振荡器=11.059Mhz*/
/*|波特率 9600 时|SMOD=0|C/T=0|模式=2|自动载入 TH1=FD|*/
/*|波特率 4800 时|SMOD=0|C/T=0|模式=2|自动载入 TH1=FA|*/
/*|波特率 2400 时|SMOD=0|C/T=0|模式=2|自动载入 TH1=F4|*/
/*|波特率 1200 时|SMOD=0|C/T=0|模式=2|自动载入 TH1=E8|*/
TR2=1; /*启动 TEMER2*/
SCON=0x50; /*设定 UART 工作在 MODE3 模式,可传送和接收数据 REN=1*/
/*串行控制寄存器 SCON_____*/
/*| 7 6 5 4 | 3 2 1 0 |*/
/*| SM0| SM1| SM2| REN| TB8| RB8 | TI | RI |*/
/*|_____*/
IE=0x9f; /*8 是允许中断, a 是 TIMER0, TIMER1 中断致能位*/
/*| 7 6 5 4 | 3 2 1 0 |*/
/*| EA | | ET2| ES | ET1| EX1| ET0| EX0 |*/
/*|_____*/
IP=0x10; /*中断优先级寄存器 IP_____*/
/*| | | PT2| PS | PT1| PX1| PT0| PX0|*/
/*|_____*/
read24c02(); /*呼叫读取密码存放在 DATE1[]副程式*/
clear(); /*清除数码管, 显示为 000000 副程式*/
P2_0=1; /*关喇叭灭指示灯*/

```

```

while(1)
{
    scan();          /*呼叫键盘扫描副程式有按 FLAG0 会为 1*/
    if(FLAG0==1) temp(); /*判断有按键输入否? */
    delay(30000);
}
}
/*****
void temp(void)
{
    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*/
    {
        switch(tab[ptr]) /*是,键盘扫描计数器指标至 TAB[]取到的键盘码*/
        {
            case 0x0a:
                write24c02(); /*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*/
                read24c02(); /*呼叫读取密码存放在 DATE1[]副程式*/
                break; /*跳出此循环*/
            case 0x0f: open(); /*是否按“D”,是则呼比较密码开门副程式*/
                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*/
        delay(1000); /*按钮抗机械反弹跳*/
        while(m==P1); /*按钮放开否? */
        delay(1000);
}

```

```

    }
}
/*****/
void xch(void)                /*显示向右移动副程式*/
{
    char c;                  /*宣告变数*/
    for(c=5;c>0;c--)
    {
        disple[c]=disple[c-1];
        /*将显示器 disple[]移位交换, disple[3]->disple[4]->disple[5]...*/
    }
    disple[0]=tab[ptr];      /*新的按键值存入 disple[0]*/
}                             /*返回主程式*/
/*****/
void clear(void)             /*清除显示器 disple[]为 00 副程式*/
{
    char c;                  /*宣告变数*/
    for(c=0;c<6;c++)        /*将 00 存入显示阵列 disple[]*/
    {
        disple[c]=0x00;
    }                       /*返回主程式*/
}
/*****/
void read24c02()            /*读 24c02 的存储值*/
{
    uchar i;

    TRcvAddress(0xa0,0,&bufdata[0],6);

    for(i=0;i<6;i++)
    {
        disple[i]=bufdata[i];
    }
}
/*****/
void write24c02()          /*写 24c02 的存储值*/

```

```

{
    TSendAddress(0xa0,0,disple,6); /* 向有子地址器件写入 6 字节数据函数 */
}
//*****
void tserial(void)                /*串口 serial 发送副程式*/
{
    uchar i=6;
    IE=0x8f;
    {
        while(i>0)
        {
            i--;
            SBUF=disple[i];        /*将显示码的 ASCII 码由 SBUF 发送出去*/
            while(TI!=1);         /*发送完成否? */
            TI=0;                  /*是, 则清除发送完成旗标 TI=0*/
            delay(10);
        }
    }
    IE=0x9f;
}
//*****
void BCD(void)                    /*BCD 码转换副程式*/
{
    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*/
    disple[0]=(SEC%10);          /*取秒的个位数存入 disple[0]*/
    disple[1]=(SEC/10);          /*取秒的十位数存入 disple[1]*/
    disple[2]=(MIN%10);          /*取分的个位数存入 disple[2]*/
    disple[3]=(MIN/10);          /*取分的十位数存入 disple[3]*/
    disple[4]=(HOUR%10);         /*取时的个位数存入 disple[4]*/
    disple[5]=(HOUR/10);         /*取时的十位数存入 disple[5]*/
}

```

```

/*****

```

启动总线函数

函数原型: void Start();

功能: 启动总线,发送 24c02 启动条件.

```

*****

```

```

void Start()                /*启动总线函数 */
{
    SDA=0;
    nop; nop;
    SCL=0;
    SDA=1;                  /*发送起始条件的数据信号*/
    nop;
    SCL=1;
    nop; nop; nop; nop; nop; /*起始条件建立时间大于 4.7us,延时*/
    SDA=0;                  /*发送起始信号*/
    nop; nop; nop; nop; nop; /* 起始条件锁定时间大于 4us*/
    SCL=0;                  /*钳住总线, 准备发送或接收数据 */
    nop; nop;
    SDA=1;
}

```

```

/*****

```

结束总线函数

函数原型: void Stop();

功能: 结束总线,发送 24c02 结束条件.

```

*****

```

```

void Stop()                 /*结束总线函数*/
{
    SCL=0;
    SDA=0;                  /*发送结束条件的数据信号*/
    nop;                    /*发送结束条件的时钟信号*/
    SCL=1;                  /*结束条件建立时间大于 4us*/
    nop; nop; nop; nop; nop;
    SDA=1;                  /*发送总线结束信号*/
    nop; nop; nop; nop;
    SCL=0;
}

```

```

/*****

```

89C51 发数据或地址给 24c02 字节数据发送函数

函数原型: void SendByte(uchar c);

功能: 将数据 c 发送出去,可以是地址,也可以是数据,发完后等待应答,并对此状态位进行操作.(不应答或非应答都使 `acknow=0`)

发送数据正常, `acknow=1`; `acknow=0` 表示被控器无应答或损坏。

```
void SendByte(uchar c)
{
    uchar count;
    for(count=0;count<8;count++)    /*要传送的数据长度为 8 位*/
    {
        if((c<<count)&0x80)
            SDA=1;                /*判断发送位*/
        else
            SDA=0;
        nop;
        SCL=1;                    /*置时钟线为高, 通知被控器开始接收数据位*/
        nop; nop; nop; nop; nop;    /*保证时钟高电平周期大于 4us*/
        SCL=0;
    }
    nop; nop;
    SDA=1;                        /*8 位发送完后释放数据线, 准备接收应答位*/
    nop; nop;
    SCL=1;
    nop; nop; nop;
    if(SDA==1)
        acknow=0;                /* 24c02 无应答 */
    else
        acknow=1;                /* 发送数据正常 */
    SCL=0;
    nop; nop;
}
*****
```

2051 从 24c02 读数据字节数据接收函数

函数原型: uchar RcvByte();

功能: 用来接收从器件传来的数据,并判断总线错误(不发应答信号),

发完后请用应答函数应答从机。

*****/

```

uchar RcvByte()
{
    uchar retc=0;
    uchar count;

    SDA=1; /*置数据线为输入方式*/
    for(count=0;count<8;count++)
    {
        nop;
        SCL=0; /*置时钟线为低,准备接收数据位*/
        nop; nop; nop; nop; nop; /*时钟低电平周期大于 4.7us*/
        SCL=1; /*置时钟线为高使数据线上数据有效*/
        nop; nop;
        retc=retc<<1;
        if(SDA==1)
            retc=retc+1; /*读数据位,接收的数据位放入 retc 中 */
        nop; nop;
    }
    SCL=0;
    nop; nop;
    return(retc);
}

```

*****/

主机 2051 应答子函数

函数原型: void Ack(bit a);

功能: 主控器进行应答信号(可以是应答或非应答信号, 由位参数 a 决定)

*****/

```

void Ack(bit a)
{
    if(a==0)
        SDA=0; /*在此发出应答或非应答信号 */
    else
        SDA=1; /* 应答 a=0 非应答 a=1 */
}

```

```

nop; nop; nop;
SCL=1;
nop; nop; nop; nop; nop;          /*时钟低电平周期大于 4us*/
SCL=0;                            /*清时钟线，钳住 I2C 总线以便继续接收*/
nop; nop;
}
/*****

```

用户接口函数

```

/*****

```

向有子地址器件写入 6 字节数据函数 对应数据资料中的写模式

函数原型: bit TSendAddress(uchar slaaddress,uchar subaddress,uchar * s,uchar no);

功能: 从启动总线到发送从地址，子地址,数据，结束总线的全过程,从器件地址 slaaddress，子地址 subaddress，发送内容是 s 指向的内容,no=字节数

如果返回 1 表示操作成功，否则操作有误。

注意: 使用前必须已结束总线。

```

*****/

```

```

bit TSendAddress(uchar slaaddress,uchar subaddress,uchar * s,uchar no)

```

```

/* 向有子地址器件写入 6 字节数据函数 */

```

```

{
    uchar i;
    Start();          /*启动总线*/
    SendByte(slaaddress); /*发送器件地址*/
    if(acknow==0)
        return(0);
    SendByte(subaddress); /*发送器件子地址*/
    if(acknow==0)
        return(0);
    for(i=0;i<no;i++)
    {
        SendByte(s[i]); /*发送数据*/
        if(acknow==0)
            return(0);
    }
    Stop();          /*结束总线*/
    return(0);
}

```

```

}
/*****
                向有子地址器件读取 4 字节数据函数
函数原型: bit  TRCvAddress(uchar slaaddress,uchar subaddress,uchar * s,uchar no);
功能:         从启动总线到发送从地址, 子地址,读数据, 结束总线的全过程,从器件
                地址 slaaddress, 子地址 subaddress, 读出的内容放入 s, no=字节数
                如果返回 1 表示操作成功, 否则操作有误。
注意:         使用前必须已结束总线。
*****/
bit TRCvAddress(uchar slaaddress,uchar subaddress,uchar * s,uchar no)
{
    uchar i;
    Start();                /*启动总线*/
    SendByte(slaaddress);   /*发送器件从地址*/
    if(acknow==0)
        return(0);
    SendByte(subaddress);  /*发送器件子地址*/
    if(acknow==0)
        return(0);
    Stop();                /*结束总线*/
    Start();               /*重新启动总线*/
    SendByte(slaaddress+1);
    if(acknow==0)
        return(0);
    for(i=0;i<no;i++)
    {
        * s=RcvByte();     /*接收数据*/
        Ack(0);           /*发送应答位*/
        s++;
    }
    * s=RcvByte();        /*接收数据*/
    Ack(1);
    Stop();              /*结束总线*/
    return(1);
}
/*****

```

```

void open(void)                                /*开门比较密码副程式*/
{
    char c;                                    /*宣告变数*/
    FLAG1=1;
    for(c=0;c<6;c++)                            /*比较六个密码*/
    {
        if (disple[c]!=bufdata[c])            /*按键值 disple[]与密码 TAB[]比较*/
            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 1,使绿色指示灯亮*/
        delay(2000);                            /*延时*/
        TMOD=0x11;
        TR1=1;
        IE=0x8a;
        clear();                                /*清除屏幕*/
        BCD();
        // P2_0=1;                              /*P2.0 置 1,使绿色指示灯灭*/
    }
    while (1)
    {
        do
        scan ();                                /*呼叫键盘扫描副程式, 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])
        {
            case 0x0c: SEC++;
                BCD();                            /*是否按"A",是则呼叫 BCD 码转换副程式*/
                break;                            /*跳出此循环*/
            case 0x0d: MIN++;
                BCD();                            /*是否按"B",是则呼叫 BCD 码转换副程式*/
                break;                            /*跳出此循环*/
        }
    }
}

```

```

        case 0x0e: HOUR++;
                BCD(); /*是否按"C", 是则呼叫 BCD 码转换副程式*/
        default:break; /*跳出此循环*/
    } /*跳至 WHILE (1) */
}
}
}
}
/*****
void disp(void) /*显示密码副程式*/
{ char c; /*宣告变数*/
  read24c02(); /*呼叫读取密码存放在 DATE1[]副程式*/
  ptr=0xff;
  for(c=0;c<6;c++) /*将密码存放在阵列 buft[]存入显示器存放阵列 disple[]*/
    {disple[c]=bufdata[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*/
}
/*****
void scan(void) /*扫描键盘副程式*/
{
  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 位元, 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~.4 未接+5V, 而是由指令加载高电平*/
    }
} /*返回主程式*/
/*****
void service_int0 () interrupt 1 using 2 /*设定每隔 3000us 中断扫描一次数码管*/
{
    ptr1++; /*扫描指标加 1*/
    TH0=(65536-3000)/256; /*每隔 4000US 扫描一次*/
    TL0=(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)|disple[ptr1]; /*将扫描值左移至高 4 位元再加上资料码*/
} /*返回主程式*/
/*****
void service_int1 () interrupt 3 using 3 /*TIMER1 中断副程式--秒计时器*/
{
    TH1=(65536-9200)/256; /*重设 TIMER1 计数值*/
    TL1=(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++; /*秒 SEC 加 1*/
        tserial(); /*宣告串口 serial 发送副程式*/
        BCD(); /*呼叫 BCD 码转换副程式*/
    }
    TF1=0;
} /*返回主程式*/
/*****
void uartcom1_in1 () interrupt 4 using 1 /*串行口中断副程式*/

```

```

{  uchar v,i=0;
    while(RI!=1);                /*接收完成否*/
    {
        RI=0;
        v=SBUF;                  /*接收的数据存入 C*/
        if(v==0xa0)              /*是否是识别码"A0"*/
            {
                while(RI!=1);    /*接收完成否*/
                bufuart[i]=v;
                for(i=1;i<7;i++)
                    {
                        while(RI!=1); /*接收完成否*/
                        bufuart[i]=SBUF; /*接收的数据存入 C*/
                        RI=0;          /*清除接收完成标志位 RI*/
                    }
            }
        else    delay(10);
        RI=0;
    }
    disple[0]=bufuart[6];
    disple[1]=bufuart[5];
    disple[2]=bufuart[4];
    disple[3]=bufuart[3];
    disple[4]=bufuart[2];
    disple[5]=bufuart[1];
}                                  /*返回主程式*/
/*****/

```

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) SY_0406.C 的源程序
 - b) SY_0406 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
 - g) Easy51Pro 下载软件
 - h) 串口调试检测软件

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，是检测仿真器端口的最佳工具板。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，可以直接监测到仿真器 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 AC/8-12V 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

5-4. 流水灯控制器（12 路）

外型如下图：



流水灯控制器是用 89C2051 单片机为机芯，可直接使用于广告灯的循环时序控制以及工业现场控制等。将控制器接上电源 DC-+5V 或 DC-+12V 后，面板上侧的绿色指示灯亮，12 路指示灯自动会象流水似的移动闪烁；如需提高或降低流水指示灯的变换速度，只需按住加速（减速）钮，这时可以看到指示灯在做二进制的减法（加法），松开按钮后指示灯的变换速度则会有相应的变化；按住正/反向钮一次，则会自动变换一次流水灯的循环方向一次。

5-5. ISP 下载线（选配自购件）

将 ISP 下载线 25pin 接口端连接在 PC 机的并口（打印口）上，10pin 排线端通过排线插在本开发板上的“AT89S52-ISP”插座上，开发板接上+5 伏电源，下载线上有红色指示灯亮即可启动下载软件进行下载（烧录）AT89S52 芯片。

请选购由深圳得技通研制的“ISP 下载线”。特供配套价每只 45.00 元。

*****公司其它产品简介见软件盘中电子版文件*****