



# Winbond 单片机 ISP 编程

## 完全攻略手册



Version : 0.01

Update : 2006.12.7



编写：陈文

恒森科技版权所有



## 目录

- 1、简介
- 2、ISP 编程硬件设计
- 3、如何烧录 LD ROM 程序
- 4、ISP 编程详细流程
- 5、软件模式进入 ISP 编程
- 6、需要烧录配置位使能硬件 ISP 的型号列表
- 7、FAQ
- 8、用户反馈途径

HENGSEN

## 1、简介

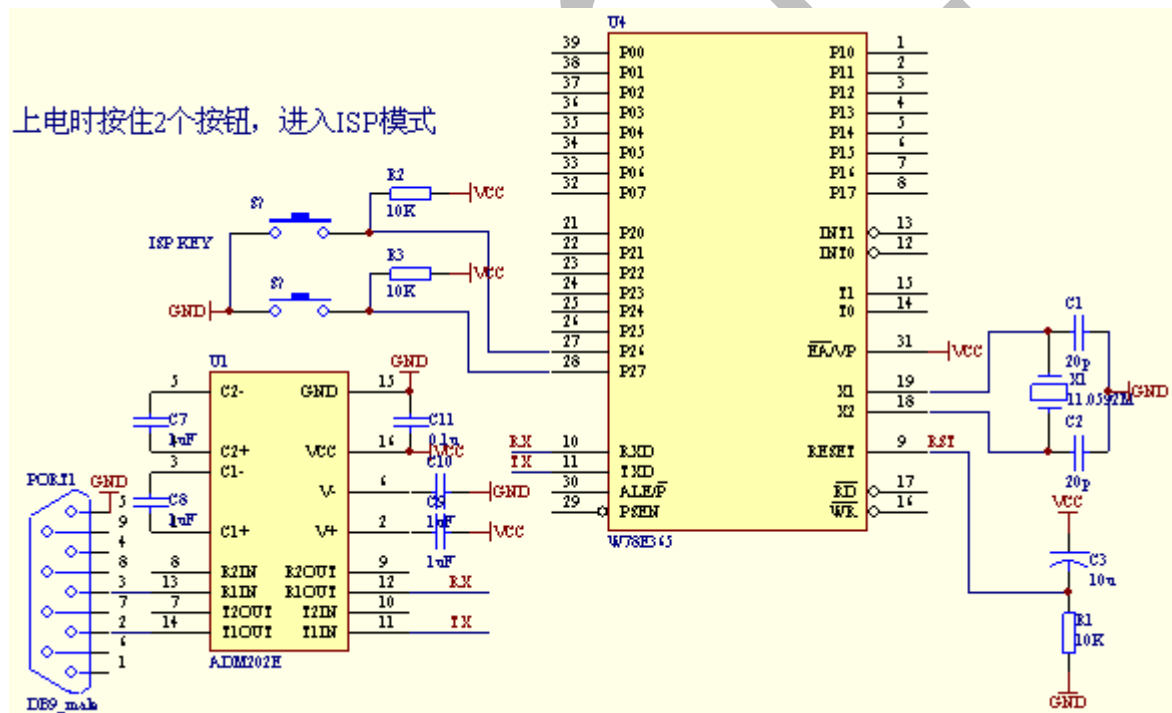
使用传统编程器对单片机重新编程会带来诸多不便，特别是已经安装在 PCB 板的芯片，Winbond 是最早使用 ISP (在系统编程) 技术的公司之一。从 W78E58B 后续产品开始，Winbond 的 W78E、W77E、W79E 系列单片机都具有 ISP 功能。使用 ISP 技术，用户将无需拆卸芯片就可以通过 RS232 对单片机进行反复编程，对产品开发和产品升级都有重要作用，对远程升级的实现也成为可能。

具有 ISP 功能的单片机内部有 2 个 ROM 空间，以 W78E516 为例，内部有一个 64K 的 APROM 和一个 4K 的 LDROM，其中 APROM 用来存放用户程序代码，LDROM 用来存放 ISP 引导码。需特别说明的是，新的单片机内部 LDROM 是没有程序的，用户必须自行用通用编程器烧录引导码。

Winbond 单片机进入 ISP 模式有 2 种方式，一种是硬件模式，在上电时手动拉低 P2.6/P2.7 或 P4.3 直接进入 ISP 模式；另一种是软件模式，通过 APROM 的程序设置寄存器进入 ISP 模式。

## 2、ISP 编程硬件设计

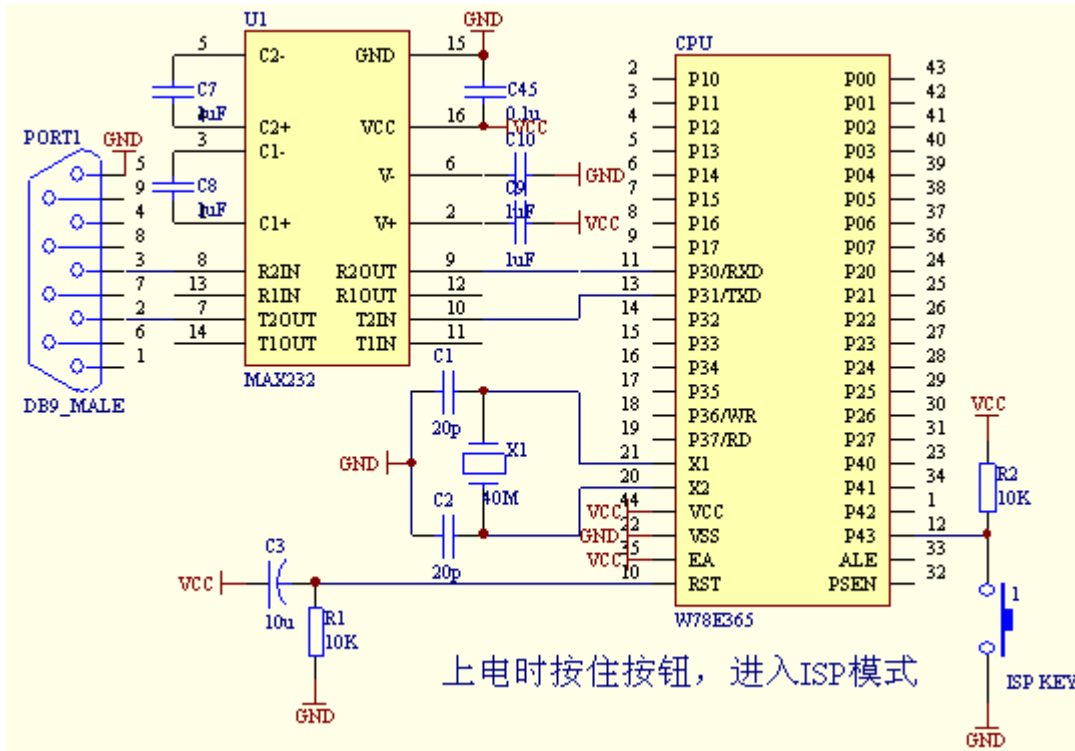
2.1、对于 DIP 封装的单片机，硬件使能 ISP 的管脚是 P2.6/P2.7，硬件设计如下图：



Note : 1、P2.6/P2.7 建议加上拉电阻，防止上电时误进入 ISP 模式；

2、对晶振无特殊要求，自动识别波特率；

2.2、对于 PLCC 封装的单片机，可以使用 P2.6/P2.7，也可以使用一个管脚 P4.3，硬件设计如下：



- Note :
- 1、P4.3 建议加上拉电阻，防止上电时误进入 ISP 模式；
  - 2、对晶振无特殊要求，自动识别波特率；
  - 2.3、对于某些特殊型号的说明：  
W79E201：ISP 使能管脚是 P4.0。

### 3、如何烧录 LD ROM 程序

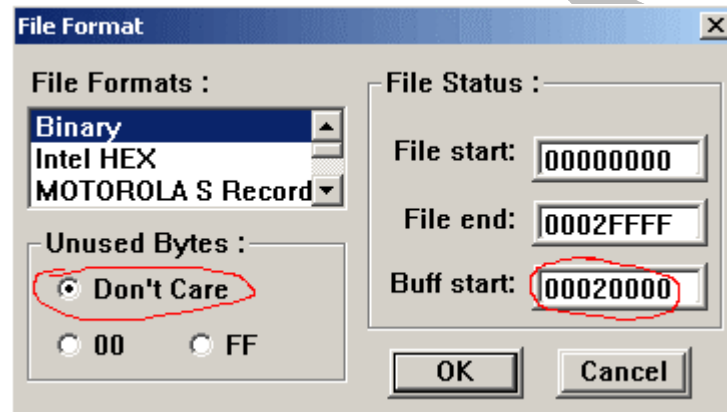
3.1、Winbond ISP 单片机的 LDROM 程序空间起始地址有 3 个，如下表：

型 号	APROM 地址	LDROM 地址
W78(L)E58B	0000 ~ 7FFF	8000 ~ 8FFF
W78E858 W79E649	0000 ~ 7FFF	10000 ~ 10FFF
W79(L)E516 W78(L)E65 W78(L)E62 W78(L)E365 W78ERD2 W78IRD2 W77(L)E516	0000 ~ FFFF	10000 ~ 10FFF
W77(L)E532 W79(L)E532 W79(L)E632 W79(L)E548 W79(L)E648 W79E201	10000 ~ 1FFFF       0000 ~ 3FFF	20000 ~ 20FFF       10000 ~ 10FFF

Note : 1、用编程器烧录 LDROM 程序时，大部分编程器是表中地址，对 W78E58 可能有所不同，以编程器提示为准。如河洛 ALL-11 编程器为例，选中 W77E532,提示如下地址：

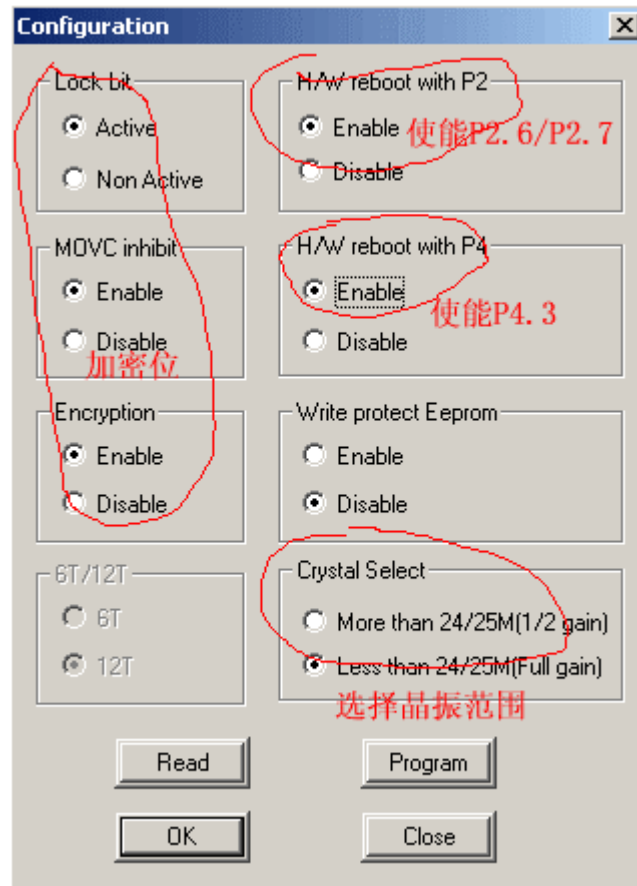


3 . 2、调入 LDROM 的程序（在 LD 目录下的 LDU40910.BIN 文件,版本不同文件名不同）



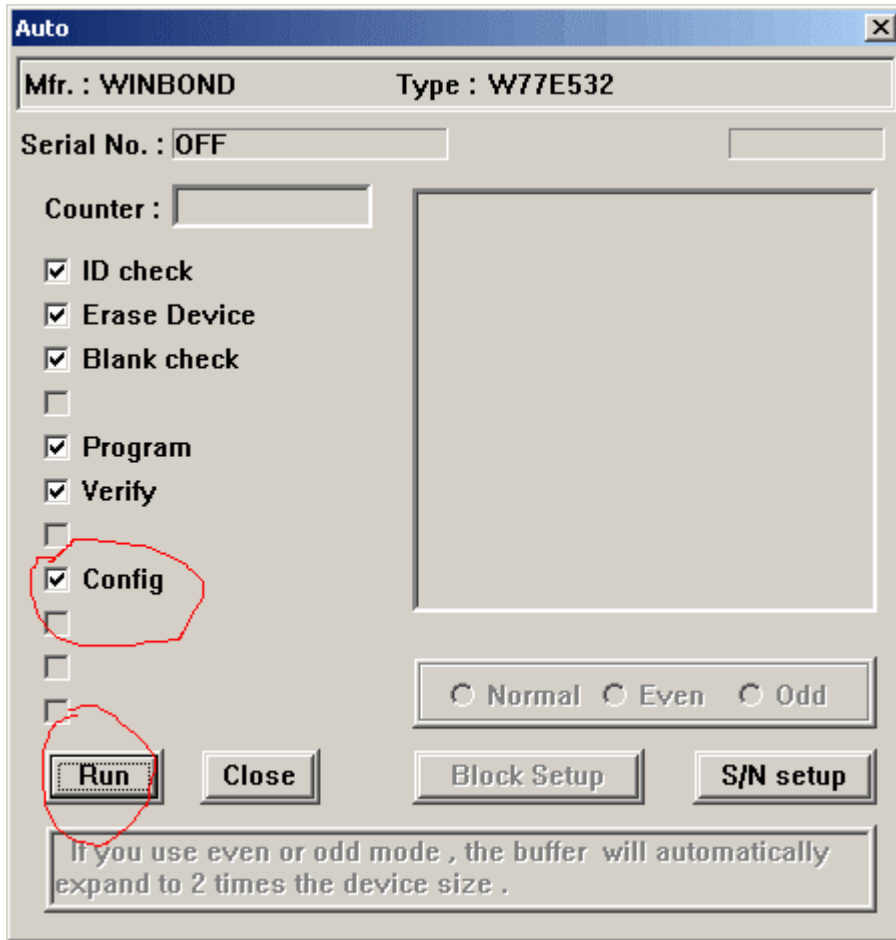
对照上表，W77E532 的起始地址是 20000，如果是 W78E365，则是 10000。如果是 APROM 和 LDROM 的程序一起烧录进单片机，操作方法是先调入 APROM 程序，再调入 LDROM 程序，注意此时 Don't Care 必须选中，否则会清除 APROM 程序。

3 . 3、设置单片机配置位、加密位等。ISP 编程时不擦除加密位，也不能设置加密位，所以需要加密的芯片必须在烧录 LDROM 程序时设置加密位。有些型号需要设置配置位才能使能 ISP 功能管脚，如 W77E532 为例：



对于 DIP 封装，只需使能 P2 口，对于 PLCC 或 TQFP 封装，可以使能 P2，也可以使能 P4，也可以 2 个都使能。一般推荐只使能一个。有些型号如 W78E365，则无设置使能位，硬件已经使能，不用设置。

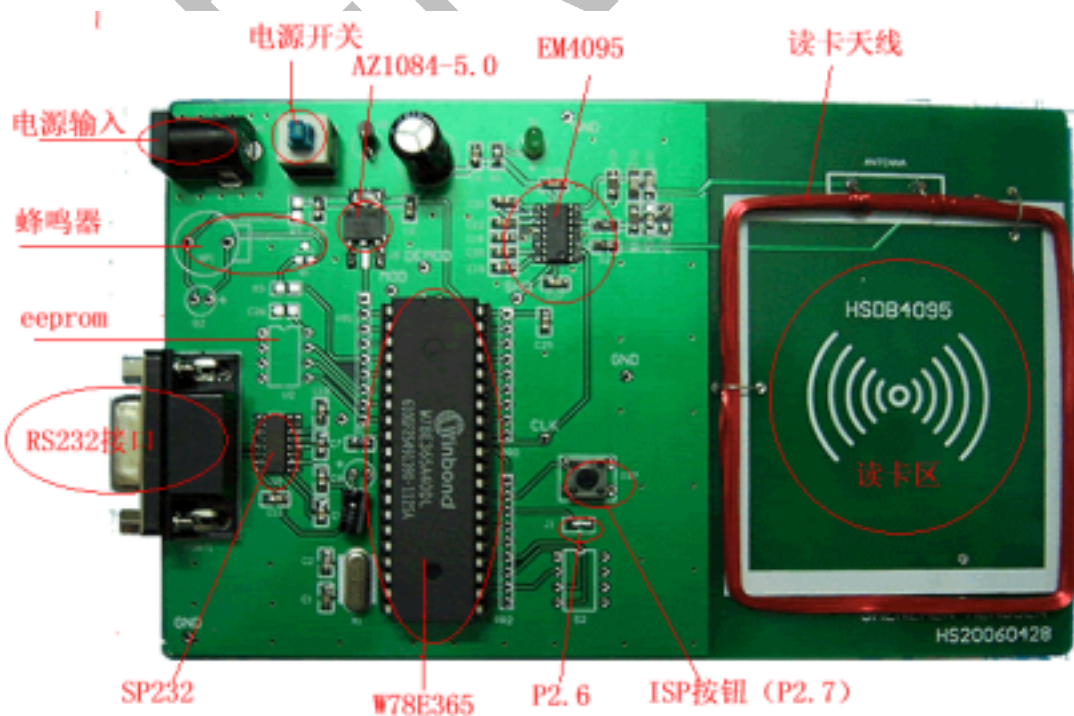
3.4、执行烧录，烧录完毕后可以进行 ISP 联机编程了。



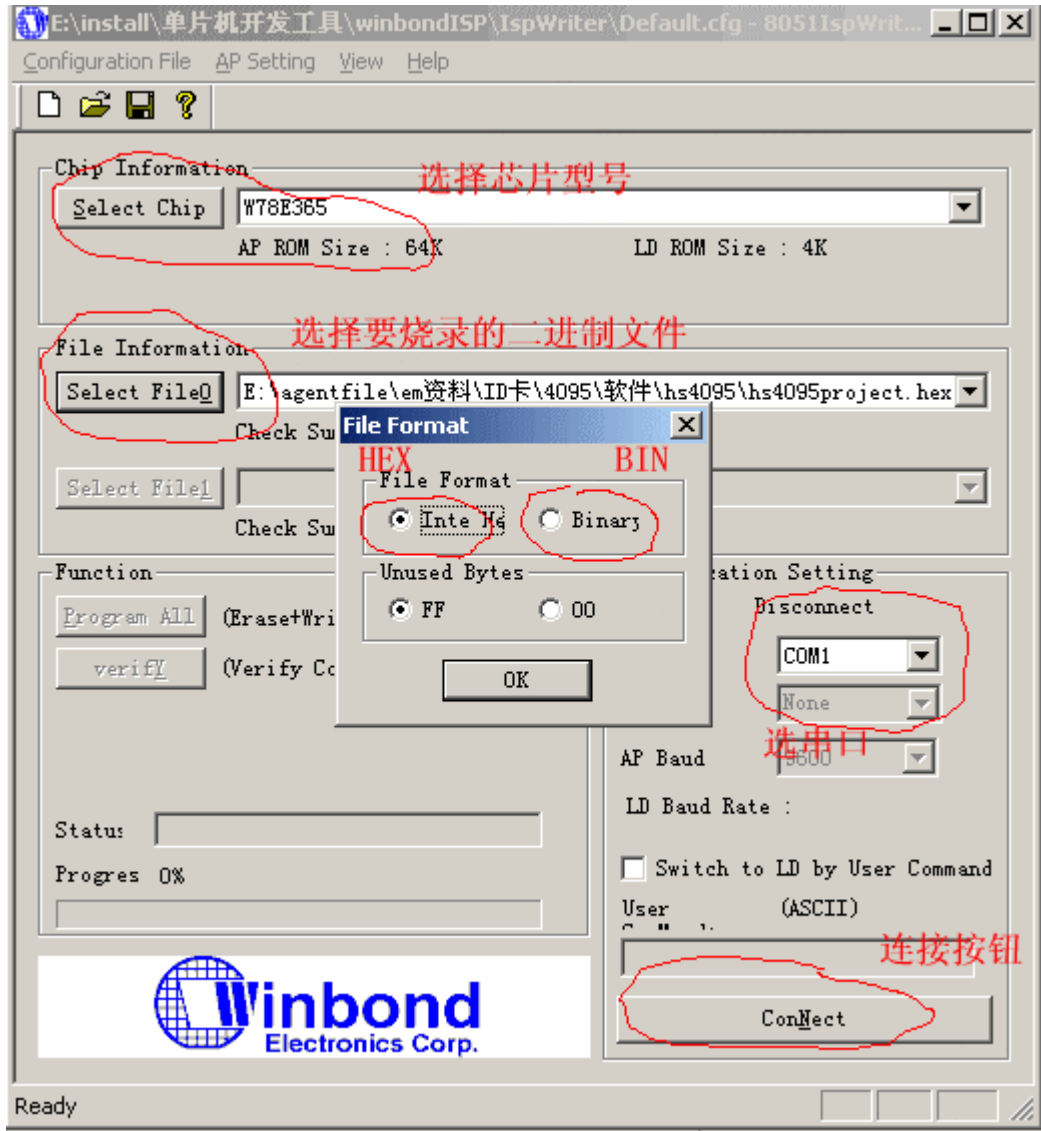
必须选中 config，执行 RUN。

#### 4、ISP 编程详细流程

以 W78E365 为例，介绍实现 ISP 编程的详细流程



4.1、启动 8051IspWriter 软件，选择芯片 W78E365，打开要烧录的二进制文件（HEX 或 BIN），选择与开发板连接的串口，自适应波特率，不用选择波特率。



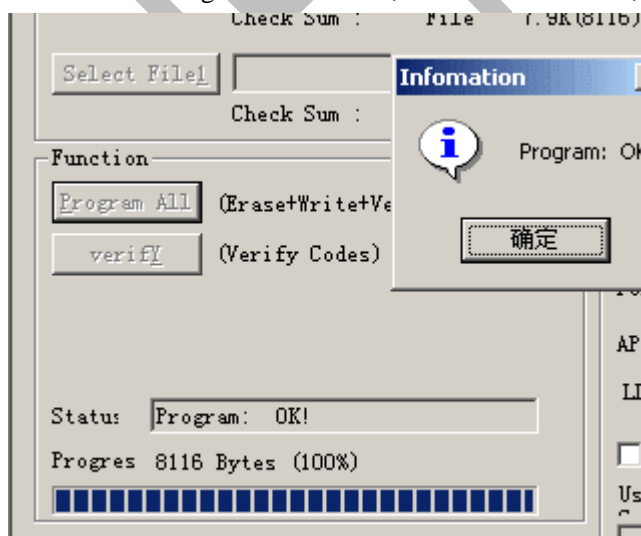
4.2、关闭开发板电源，按下开发板上 ISP 小按键，同时打开电源，W78E365 被强制进入 ISP 模式。

4.3、按下 ISP 软件上的 ConNect 按钮，如成功连接单片机，将出现如下界面：





4.4、按下 Program ALL 按钮，对 W78E365 编程，编程完成显示以下界面：



4.5、如出现连接失败（打开串口失败），请检查：

- 检查有无别的软件占用串口；

- 重新对 W78E365 上电操作（执行步骤 4.3）；
- DIP 封装的 W78E365 硬件进入 ISP 模式的管脚是 P2.6/P2.7，PLCC 和 PQFP 封装可以是 P4.3；因为这个原因，建议客户对这 2 个 I/O 加 10K 的上拉电阻，增强抗干扰能力。
- 检查 W78E365 是否能与 PC 可靠通讯；
- 检查目标板有无外部的看门狗芯片；
- 可以编写一个简单的测试程序，如驱动一个 I/O 口的高低电平，在烧录 LDROM 的同时，把测试程序烧录进 APROM，如 I/O 有高低电平驱动，说明没有进入 ISP 模式，检查配置位是否需要设置，按钮是否能拉低电平。如 I/O 时高电平，则已经进入了 LDROM，检查 LDROM 引导码烧录是否正确，地址是否正确，串口通讯是否正常。

## 5、软件模式进入 ISP 编程

5.1、在某些应用系统里，不能够手动进入 ISP，这时就需要用到软件 ISP 编程。程序首先在 APROM 运行，需要进入 LDROM 时，设置寄存器，通过一段软件实现进入 LDROM 区执行 ISP 程序。可以用串口接收特定的数据流判断是否要进入 ISP，也可以通过系统的键盘进入（一般用组合键，单键或简单的组合键比较容易误进入）。

5.2、以串口数据流为例，下面是 APROM 程序范例，用户自行修改嵌入自己的 APROM 程序。一般占用串口、定时器、小部分的 RAM 和 ROM 空间。

```

unsigned char CodeTotalNumber = 8-1;
//          W   B   1   2   3   4   5   6
char code UserCode[] = {0x57,0x42,0x31,0x32,0x33,0x34,0x35,0x36}; //串口数据流
/*****/
//          ISP Mode APROM File
/*****/
sfr P2 = 0xA0;
sbit P2_4 = P2^4; //驱动外部一个 LED 观测是否进入 ISP 程序
sfr sbuf = 0x99;
sfr pcon = 0x87;
sfr scon = 0x98;
sbit ti = scon^1;
sbit ri = scon^0;
sfr ie = 0xA8;
sbit ex0 = ie^0;
sbit et0 = ie^1;
sbit ex1 = ie^2;
sbit et1 = ie^3;
sbit es = ie^4;
sbit et2 = ie^5;
sbit ea = ie^7;
sfr ip = 0xB8;
sbit px0 = ip^0;
sbit pt0 = ip^1;
sbit px1 = ip^2;
sbit pt1 = ip^3;
    
```

```

sbit ps          = ip^4;
sbit pt2        = ip^5;
sfr psw         = 0xd0;
sbit p          = psw^0;
sbit ov         = psw^2;
sbit rs0        = psw^3;
sbit rs1        = psw^4;
sbit f0         = psw^5;
sbit ac         = psw^6;
sbit c          = psw^7;
sfr tcon       = 0x88;
sbit it0        = tcon^0;
sbit ie0        = tcon^1;
sbit it1        = tcon^2;
sbit ie1        = tcon^3;
sbit tr0        = tcon^4;
sbit tf0        = tcon^5;
sbit tr1        = tcon^6;
sbit tf1        = tcon^7;
sfr t2con      = 0xc8;
sbit cprl2     = t2con^0;
sbit ct2       = t2con^1;
sbit tr2       = t2con^2;
sbit exen2     = t2con^3;
sbit tclk      = t2con^4;
sbit relk      = t2con^5;
sbit exf2      = t2con^6;
sbit tf2       = t2con^7;
sfr tmod       = 0x89;
sfr th0        = 0x8c;
sfr tl0        = 0x8a;
sfr th1        = 0x8d;
sfr tl1        = 0x8b;
sfr th2        = 0xcd;
sfr tl2        = 0xcc;
sfr rcap2h     = 0xcb;
sfr rcap2l     = 0xca;
sfr CHPCON     = 0xbf;
sfr CHPENR     = 0xf6;
sfr SFRAL      = 0xc4;
sfr SFRAH      = 0xc5;
sfr SFRFD      = 0xc6;
sfr SFRCN      = 0xc7;
sfr turbo_TA   = 0xc7;

```

```

sfr turbo_SFRAL          = 0xac;
sfr turbo_SFRAH          = 0xad;
sfr turbo_SFRFD          = 0xae;
sfr turbo_SFRCN          = 0xaf;
sfr turbo_CHPCON         = 0x9f;
sfr turbo_CKCON          = 0x8e;          // clock control
sfr turbo_T2MOD          = 0xc9;          // clock control
bit jmpLDROM             = 0x20;
bit TimerOut             = 0x21;

```

```

void InitialPowerON(void);
void ChangeAPtoLD(void);
void RecDataTimer(void);
/*****/
void Timer0ISR (void) interrupt 1 using 1
{
    tr0=0;
    tf0=0;
}
/*****/
void SerialInt(void) interrupt 4 using 3
{
    char i,j,code_a;
    if(ri==0)
    { ti=0; }
    else
    {
        jmpLDROM=1;
        i=0x00;
        while(1)
        {
            j=sbuf;
            code_a=UserCode[i];
            if(j!=code_a)
            {
                jmpLDROM=0;
                goto ERROR;
            }
            while(i==CodeTotalNumber)
                goto OK;
            RecDataTimer();
            if(TimerOut==1)
            {
                jmpLDROM=0;

```

```

        goto ERROR;
    }
    i++;
}
}
OK: ;
ERROR: ;
ri=0;
tr0=0;
}
/*****/
void main(void)
{
int i;

InitialPowerON();
jmpLDROM=0;

while(1)
{

P2_4 = ~P2_4;
if(jmpLDROM==1)
{
    es=0;
    ti=0;
    sbuf=0x00;           //Command OK Code
    while(ti==0)
    { ; }
    ti=0;
    es=1;
    ChangeAPtoLD();
}
for(i=0;i<8000;i++);
}
}
/*****/
void InitialPowerON(void)
{
    turbo_CKCON=0x01;
    ie=0x00;
    et0=1;           // timer0 interrupt enable
    es=1;           // RS-232 interrupt enable
    ea=1;           // all interrupt enable

```

```

// Timer2BaudRate();
scon=0x50; // mode=mode1 ,ren=enable
t2con=0x30; // Timer2=BaudRate CLK

rcap2h = 0xff; //Fosc = 11.0592M
rcap2l = 0xdc; //BaudRate = 9600

th2=rcap2h;
tl2=rcap2l;
tr2=1;
}
/*****/
void RecDataTimer(void)
{
//unsigned int delay;
char delay;
ri=0;
TimerOut=0;

tmod=0x11;
tr0=1;
for(delay=50; delay>0; delay--)
{
//11.0592M
th0 = 0xfc; // th0=(65536-(1000*(Fosc/12000000)))/256; // delay x 1ms
tl0 = 0x66; // tl0=65536-(1000*(Fosc/12000000)); // delay x 1ms
for(tf0=0; tf0==0; )
{
while(ri==1)
{ goto OK ; }
}
}
TimerOut=1;
OK: ;
}
/*****/
void ChangeAPtoLD(void)
{
CHPENR=0x87; // write enable
CHPENR=0x59; // write enable
CHPCON=0x03; // bank=4k,flash mode
tl0=0xfe;
th0=0xff;
tr0=1;
}

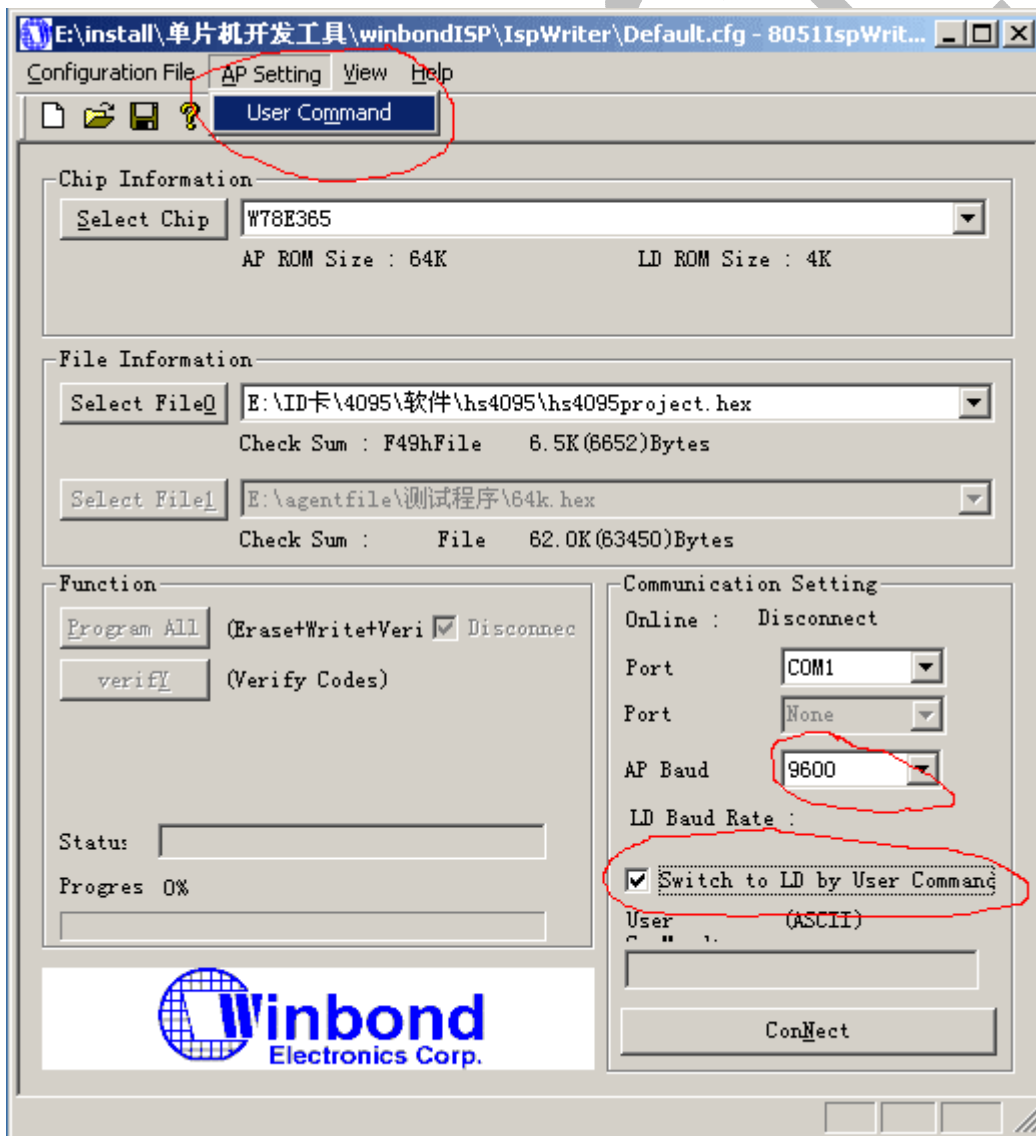
```

```

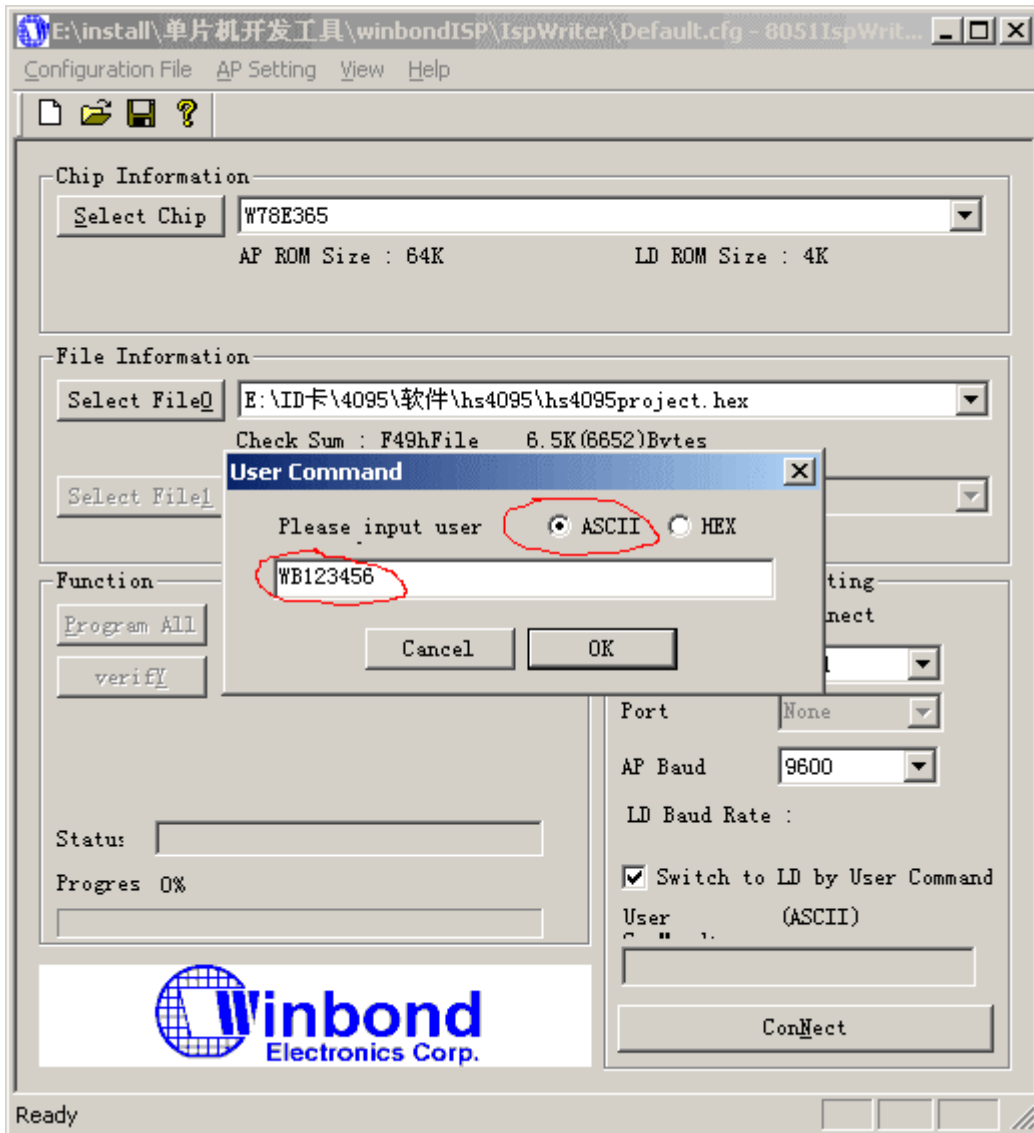
pcon=pcon|0x01;          // enter idle mode
;
;
;
;
        turbo_TA=0xaa;          // write enable
        turbo_TA=0x55;          // write enable
        turbo_CHPCON=0x03;      // bank=4k,flash mode
        tl0=0xfe;
        th0=0xff;
        tr0=1;
        pcon=pcon|0x01;          // enter idle mode
;
;
;
}

```

5.3、ISP 软件控制界面：



输入命令 WB23456 :



输入命令后，按 ConNect 按钮，后续的操作与硬件 ISP 相同。

6、需要烧录配置位使能硬件 ISP 的型号列表

W77(L)E516(A),W77(L)E532(A),W78ERD2,W78IRD2,W79E201,W79(L)E5329A),W79(L)E549(A),  
W79(L)E632(A),W79(L)E648(A),W79(L)E649(A),W79(L)E658,W79(L)E659

7、FAQ

7.1、Q：串口通讯正常的系统，MAX813L 看门狗复位电路，硬件模式连接不上 ISP？

A：看门狗电路影响了 ISP 引导程序运行，解决办法：暂时去掉看门狗电路或在 ISP 引导码加入看门狗喂狗程序语句，在论坛 <http://www.hengsen.cn/form/index.asp> 上有成功案例。

7.2、Q：W78E365 芯片出厂有无烧录 ISP 引导程序？

A：没有。用户自行烧录。ISP 引导程序在 isp 软件的文件夹 LD 目录下。

8、用户反馈途径

WINBOND 用户在使用 ISP 功能有任何问题或本文档有任何错误请联系：

Email：chenwen@hengsen.cn