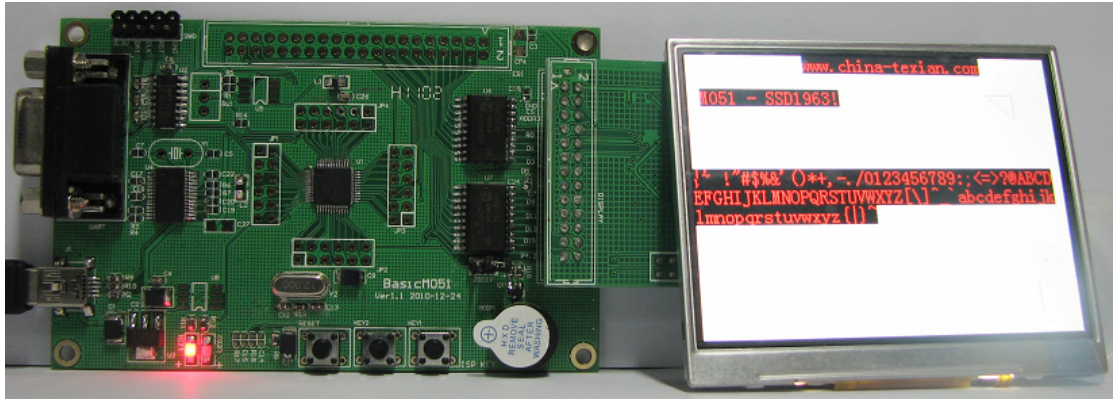


## 如何使用 M051 芯片点亮 SSD1963



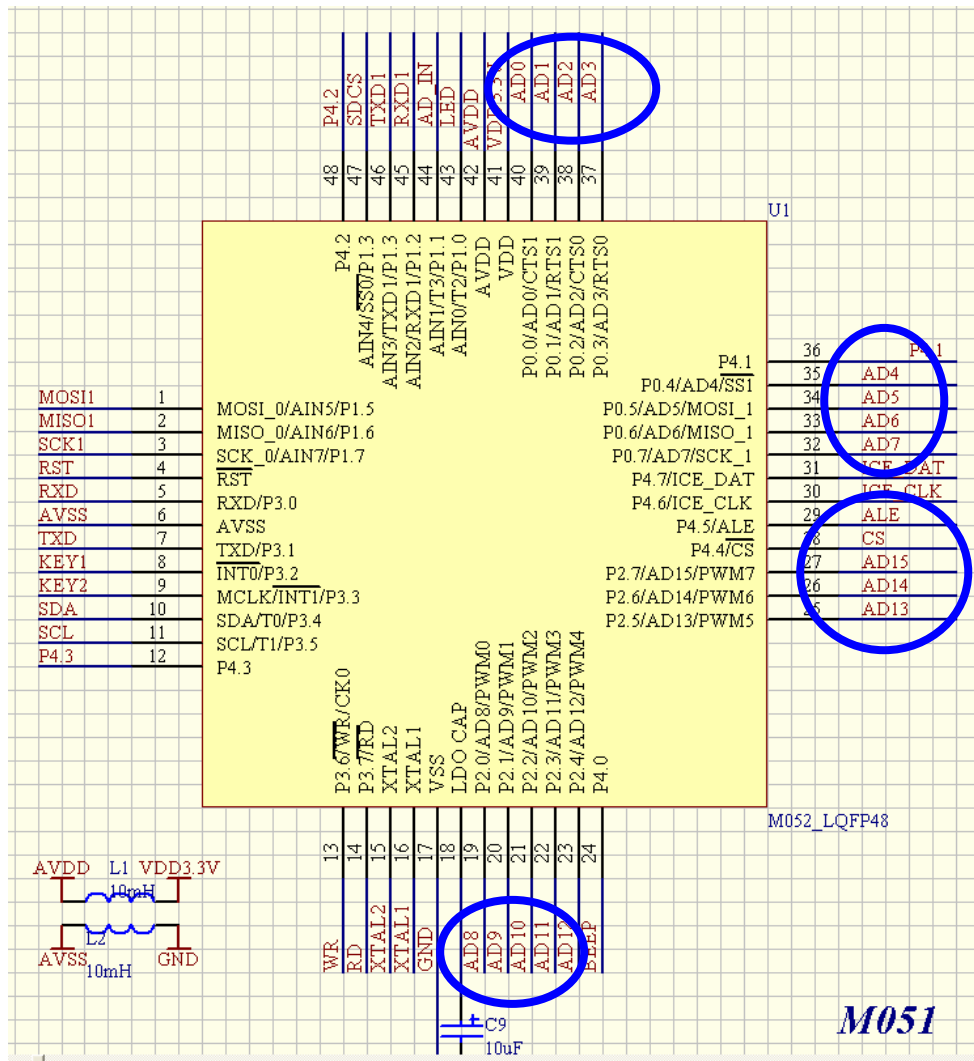
M051 是一款 CORTEX-M0 内核的芯片，其特点就是低功耗低成本，M051 具有的另一特点就是和 MCS-51 在引脚上兼容，使得在设计上大为简化。

SSD1963 是一款 TFT 屏控制器，最大分辨率支持 24 位 864x480，支持 18/24 位 RGB 接口，有 8/9/16/18/24 位 CPU 接口，并具有 4 位可控的 IO。用 SSD1963 来驱动大屏很方便，不需要其他外围电路，程序也非常简单。

用 M051 来驱动 SSD1963 是件很简单的事情，因为 M051 自带了总线驱动，将 SS1963 接在总线上，进行寻址，就可以完成操作。

### 先看硬件上的设计：

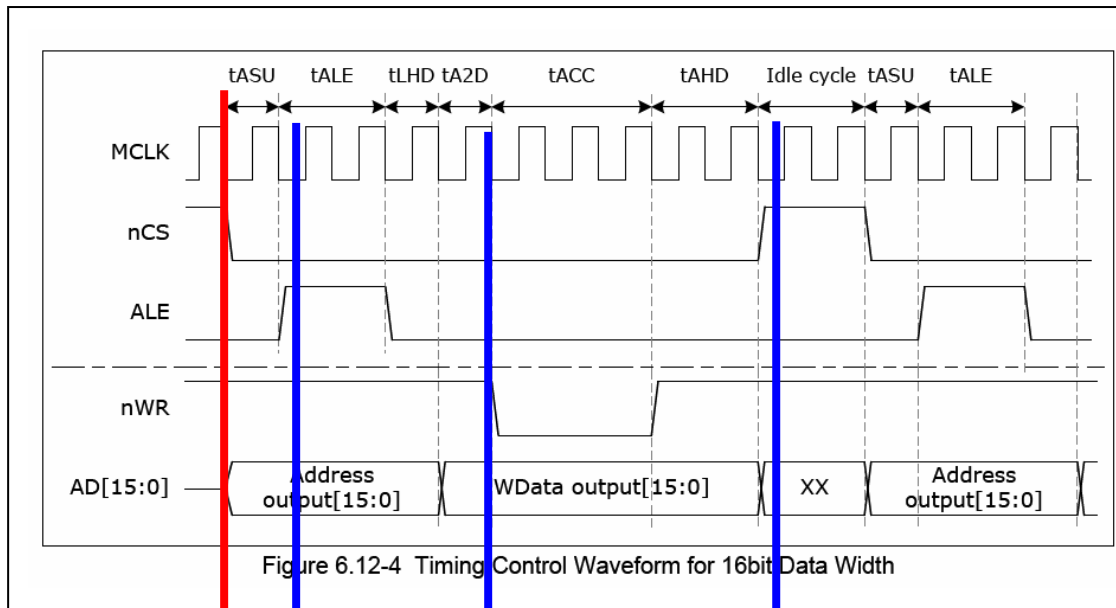
#### (1) M051 部分



M051 具有 16 位总线： 16 根地址线， 16 根数据线， 1 根片选线。 不过 M051 和 51 单片机类似， 地址线和数据线是复用的， 需要使用一个锁存信号来区分这两个信号。 见上图中蓝色圈。

(2) 锁存电路部分





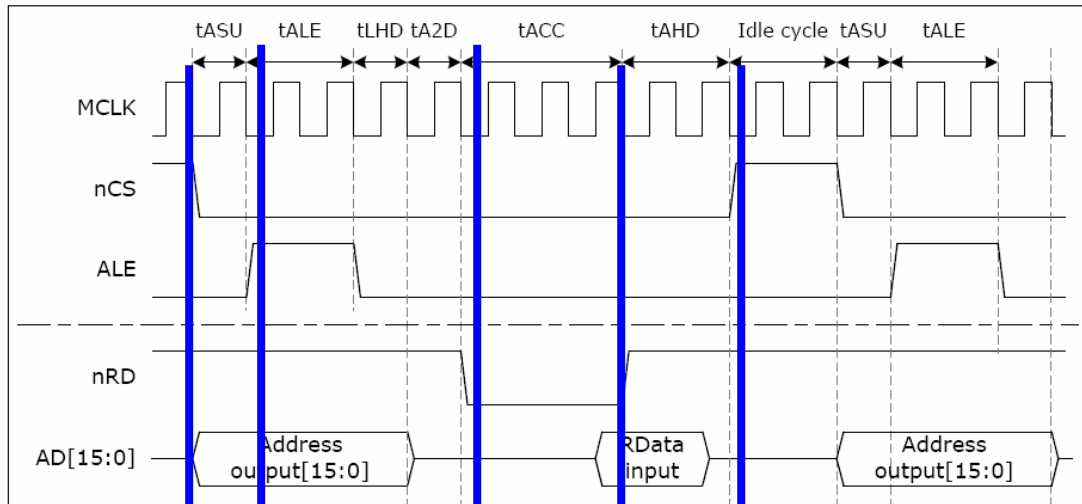
起始阶段：  
CS 被置 0  
ALE 保持 0  
地址数据放在总线 AD[15:0]上

第二阶段：  
CS 保持 0  
ALE 被置 1  
此时锁存器完成输出，地址数据被输出

第三阶段：  
CS 保持 0  
ALE 被拉低  
数据放到总线 AD[15:0]上  
WR 产生低电平

结束阶段：  
CS 被拉高，完成总线操作

M051 在控制读总线时， 先将 CS 置低， 而 ALE 通常为 0， 地址数据放到 AD0—AD15 上， 然后将 ALE 置高， 这样就将地址数据输出到了 573 的输出端， 地址输出完成后， 产生 RD 信号， 然后从总线上读取数据。 请看下图。



起始阶段：  
CS 置 0  
ALE 保持低  
地址数据放到总线 AD[15:0] 上

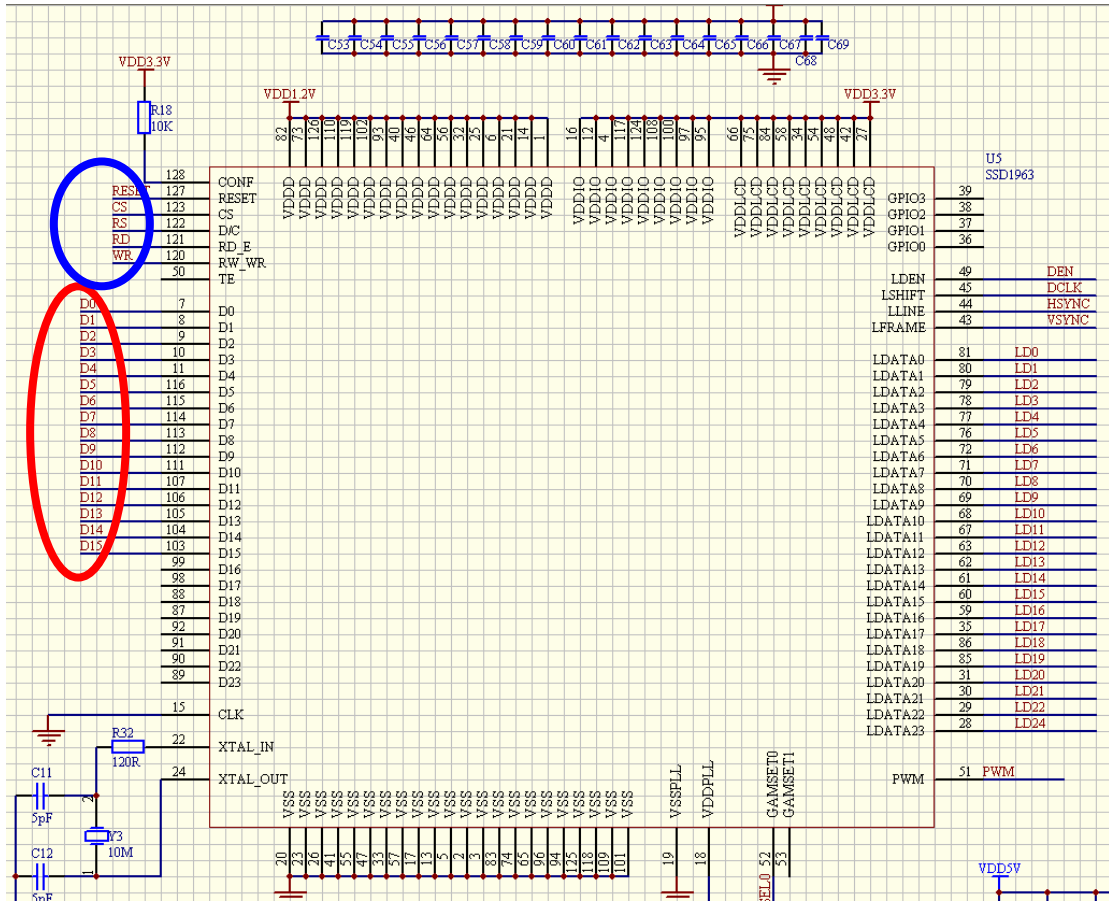
第三阶段：  
CS 保持 0  
ALE 被拉低  
RD 信号产生

第四阶段：  
CS 保持 0  
数据被读取

第二阶段：  
CS 保持 0  
ALE 被拉高  
锁存器完成锁存，地址输出

结束阶段：  
CS 被拉高，完成读操作

(3) SSD1963 的硬件连接



上图是 SSD1963 与 M051 连接的电路图。

图中蓝色圈中为控制线：

RESET 连到一个 IO 上， 就可以用 IO 来控制； 如果连到系统复位信号上， 就可以上电复位

CS 连到 M051 的 CS 脚上， 做片选

RS 连到 573 输出的一个地址线上（ADDR3）， 用来选择对 SSD1963 的操作

WR/RD 连到 M051 的 WR/RD 引脚上， 进行读写控制。

图中红色圈为数据， 直接连到 M051 的数据线上。

SSD1963 的输出端就是 RGB 信号， 包括 24 位数据颜色信号， 4 根同步信号。可以直接连到 RGB 接口的 TFT 屏上。

## 再看软件部分

### (1). 确定 SSD1963 的地址

M051 的外部地址范围是 0x6000 0000 — 0x6001 FFFF。硬件上 SSD1963 的 RS 连到了 ADDR3 上，所以写指令的地址就是 0x6000 0010，写数据的地址就是 0x6000 0000。做如下定义即可访问相关硬件：

```
#define Bank1_LCD_D    ((uint32_t)0x60000010)    //disp Data ADDR
#define Bank1_LCD_C    ((uint32_t)0x60000000)    //disp Reg ADDR
```

### (2)基本的操作函数

//写寄存器地址函数

```
void LCD_WR_REG(unsigned int index)
{
    *(__IO uint16_t *) (Bank1_LCD_C)= index;
}
//写数据函数
```

//写数据函数

```
void LCD_WR_Data(unsigned int val)
{
    *(__IO uint16_t *) (Bank1_LCD_D)= val;
}
// 读数据函数
```

// 读数据函数

```
unsigned int LCD_RD_data(void)
{
    __IO unsigned int a=0;
    a=*(__IO uint16_t *) (Bank1_LCD_D); //L
    return(a);
}
}
```

### (3)初始化操作

SSD1963 的初始化很简单，打开 PLL，并设置好 LCD 的参数即可：

//初始化函数

```
void LCD_Init(void)
{
    LCD_WR_REG(0x00E2); //PLL multiplier, set PLL clock to 120M
    LCD_WR_Data(0x0023); //N=0x36 for 6.5M, 0x23 for 10M crystal
    LCD_WR_Data(0x0002);
    LCD_WR_Data(0x0004);
    LCD_WR_REG(0x00E0); // PLL enable
    LCD_WR_Data(0x0001);
    delay_time(10);
    LCD_WR_REG(0x00E0);
    LCD_WR_Data(0x0003);
    delay_time(10);
}
```

```
LCD_WR_REG(0x0001); // software reset
delay_time(10);
LCD_WR_REG(0x00E6); //PLL setting for PCLK, depends on resolution
LCD_WR_Data(0x0001);
LCD_WR_Data(0x0047);
LCD_WR_Data(0x00be);
```

```
LCD_WR_REG(0x00B0); //LCD SPECIFICATION
LCD_WR_Data(0x0024);
LCD_WR_Data(0x0000);
LCD_WR_Data((HDP>>8)&0X00FF); //Set HDP
LCD_WR_Data(HDP&0X00FF);
LCD_WR_Data((VDP>>8)&0X00FF); //Set VDP
LCD_WR_Data(VDP&0X00FF);
LCD_WR_Data(0x0000);
```

```
LCD_WR_REG(0x00B4); //HSYNC
LCD_WR_Data((HT>>8)&0X00FF); //Set HT
LCD_WR_Data(HT&0X00FF);
LCD_WR_Data((HPS>>8)&0X00FF); //Set HPS
LCD_WR_Data(HPS&0X00FF);
LCD_WR_Data(HPW); //Set HPW
LCD_WR_Data((LPS>>8)&0X00FF); //Set LPS
LCD_WR_Data(LPS&0X00FF);
LCD_WR_Data(0x0000);
```

```
LCD_WR_REG(0x00B6); //VSYNC
LCD_WR_Data((VT>>8)&0X00FF); //Set VT
LCD_WR_Data(VT&0X00FF);
LCD_WR_Data((VPS>>8)&0X00FF); //Set VPS
LCD_WR_Data(VPS&0X00FF);
LCD_WR_Data(VPW); //Set VPW
LCD_WR_Data((FPS>>8)&0X00FF); //Set FPS
LCD_WR_Data(FPS&0X00FF);
LCD_WR_REG(0x0036); //rotation
LCD_WR_Data(0x0000);
```

```
LCD_WR_REG(0x00F0); //pixel data interface
LCD_WR_Data(0x0003);
```

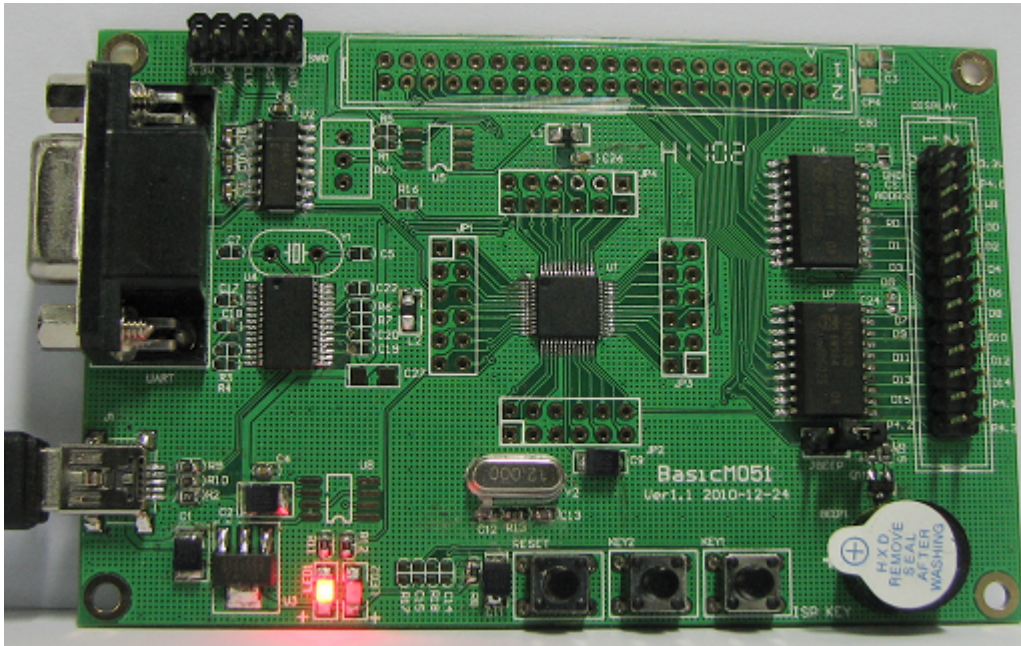
```
delay_time(50);
```

```
LCD_WR_REG(0x0029); //display on
```

```
}
```



以上是 M051 操作 SSD1963 的方法。我的邮箱是 [ldqmoon@163.com](mailto:ldqmoon@163.com)，QQ 是 28773030。欢迎大家来交流。相关的开发板在我的淘宝店上 [texian.taobao.com](http://texian.taobao.com)。



ldqmoon  
2011-2-19

## 付：如何烧录 M051.

M051 中会有两个部分:

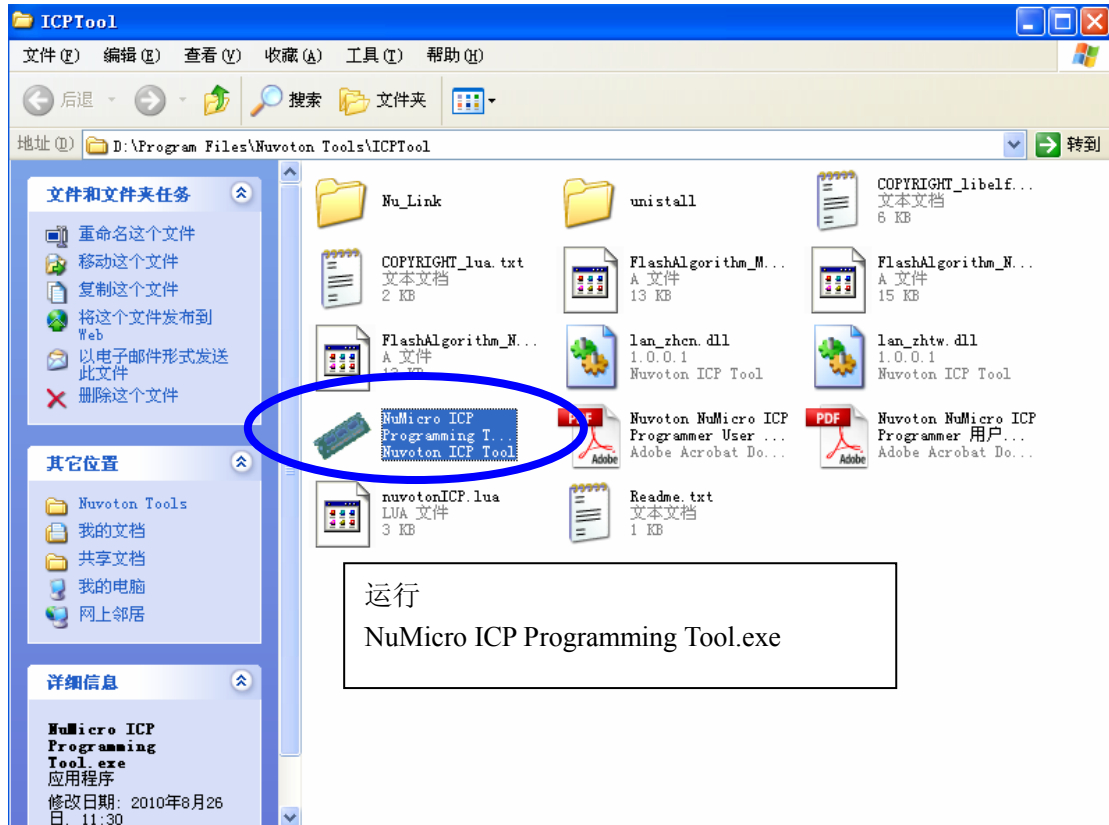
LDROM : 用来做引导用, 用来实现 ISP 功能

APROM0 : 用户程序区.

烧录的时候会用两个软件, 一个是 ICPTool, 主要用 Nulink 来烧录的, 另一个是 ISPTool, 主要是用串口烧录的.

(1) 使用 ICPTool 来烧录.

将 NULINK 连接到开发板上, 使用 ICE 接口. 上电.





nuvoton



选择界面语言 (Select UI language):

- 自动检测 (Auto Detect) -

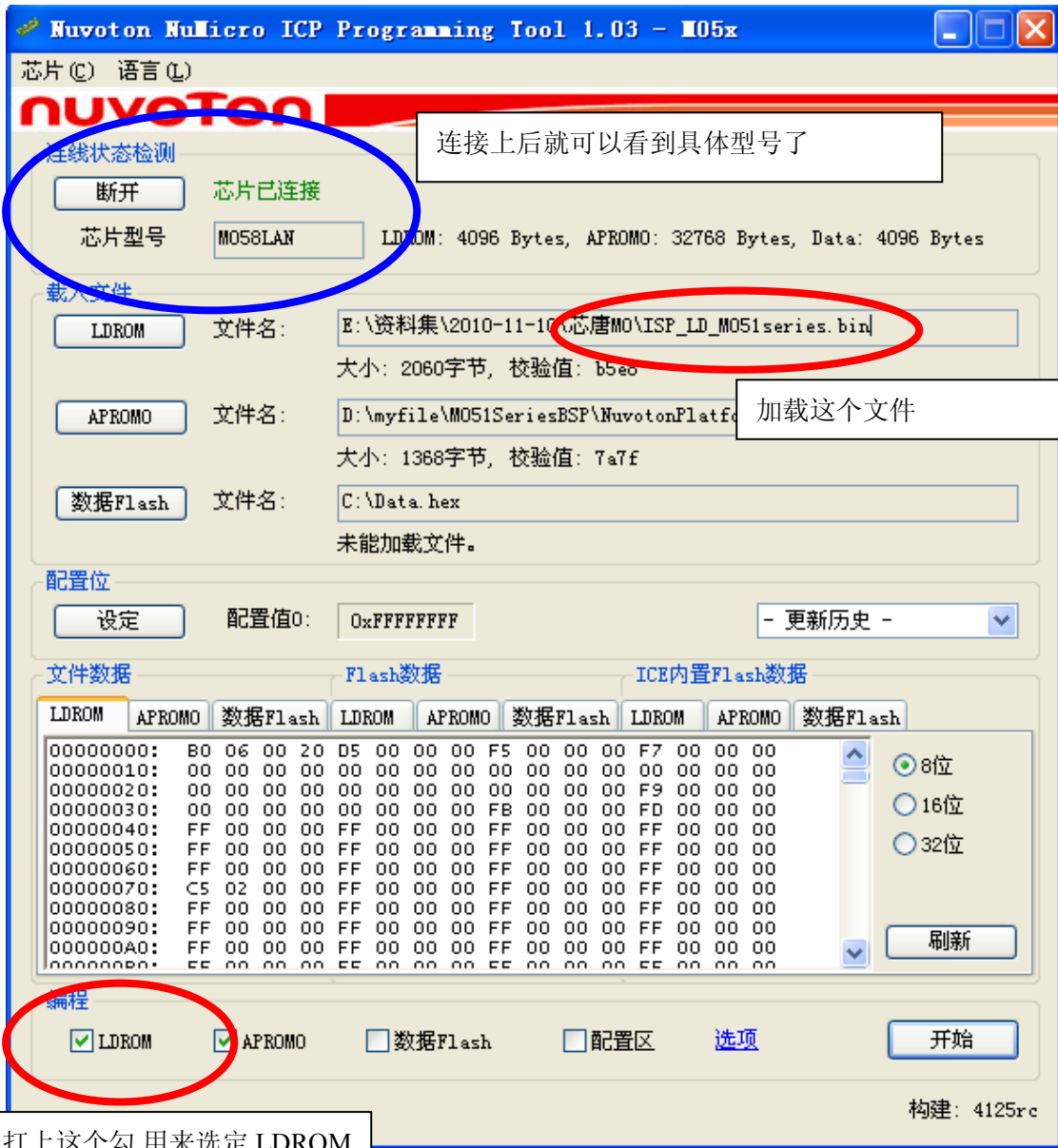
选择目标芯片:

M05x系列

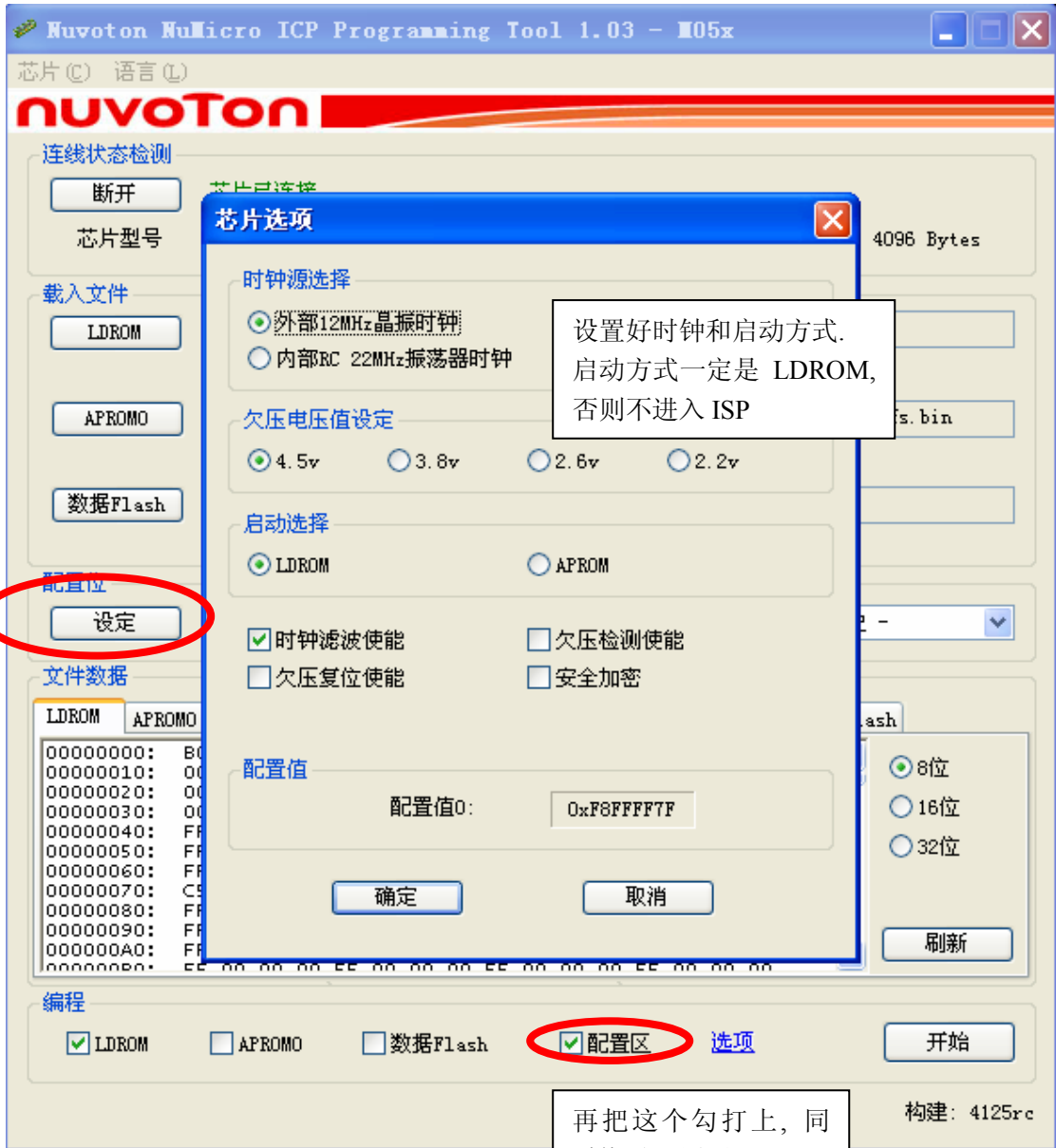
退出

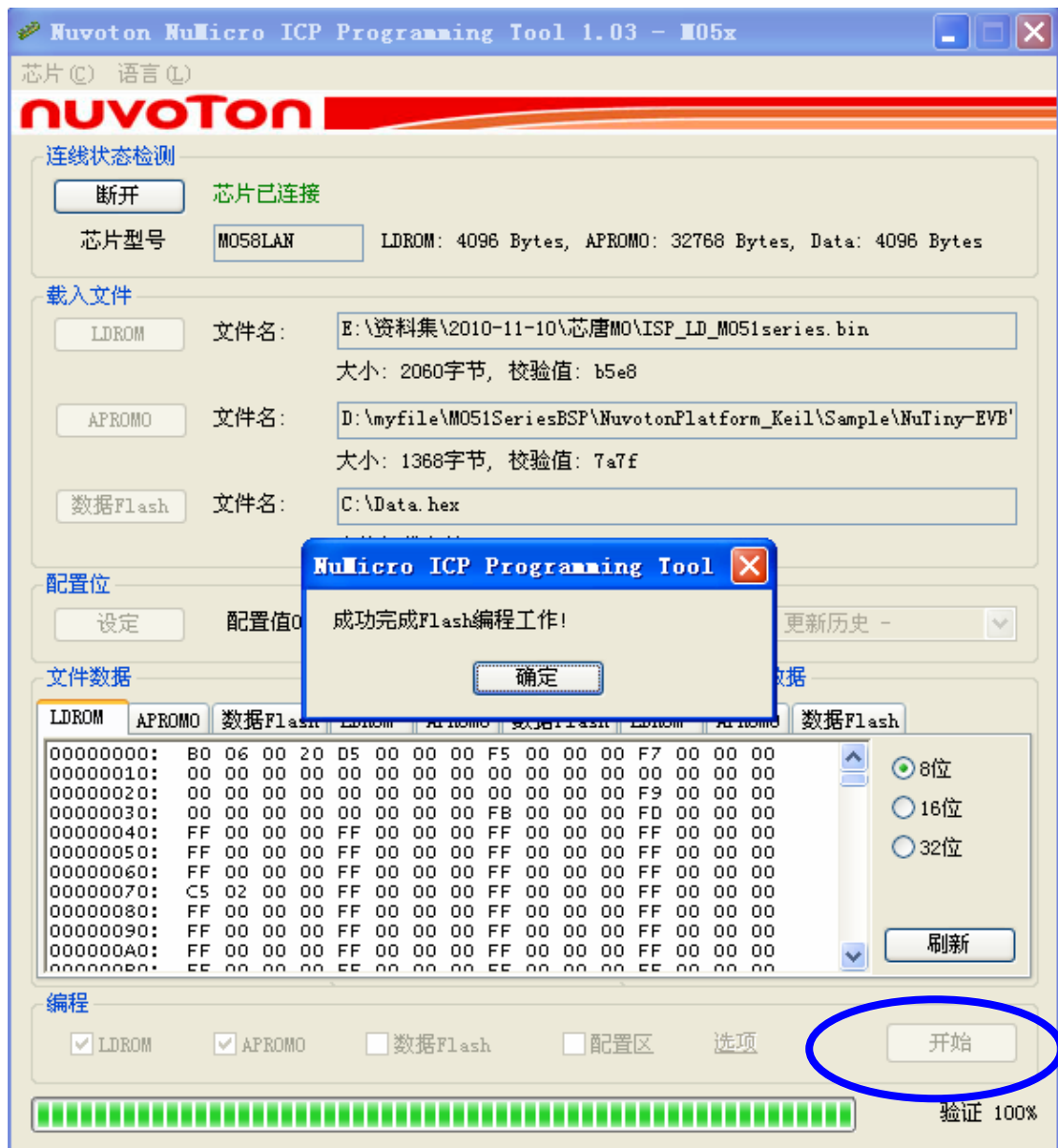
继续>>





打上这个勾,用来选定 LDRUM

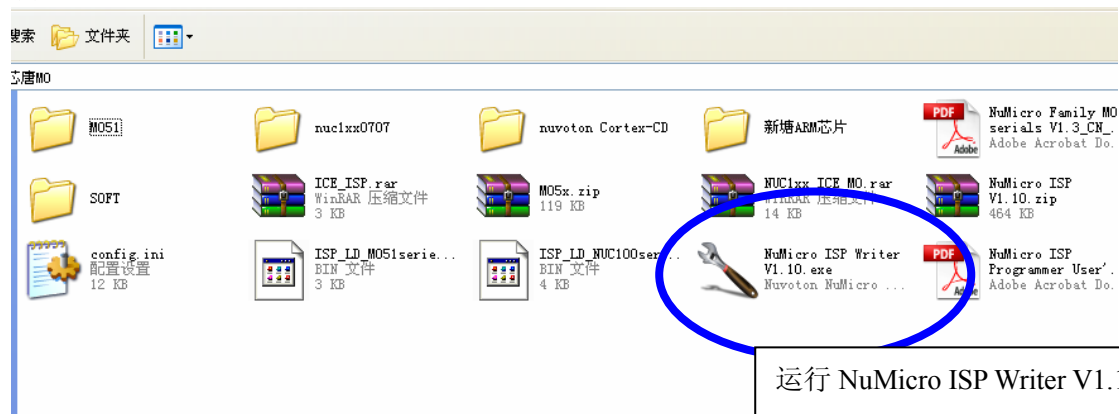




至此, LDR0M 烧录

LDR0M 只需要烧录一次就可以使用 ISP 功能了, 我们出售的开发板, 都已经烧录好了 LDR0M, 只用串口就可以进行开发了.

## (2)使用串口做 ISP 烧录



M051 会在 P3.2 为低,复位时进入 ISP 状态. 所以在连接前, 先按住开发板上的 KEY1, 然后按复位键,就可以进入 ISP 了.

