



图为12v控制电路

其它引脚见程序文件说明，也可参考ATmega8的datsheet, 只要连接无误，将程序编译后烧入at89c51即可

-----以上部分请勿修改！-----

*****ATmega8(L) 简易并行编程器*****

说明：适用于没用烧写ATmega8并行编程器的朋友，初用ATmega8的工程师常在串行编程时写错熔丝位，及加密位，造成不能再串行编程的不便，没并行编程将无法再使用，本人就是因此才特花两天时间做了一个简易的并行编程器(很简单，用万能板搭焊即可)，将的芯片加密位及熔丝恢复出厂默认值；恢复串行编程。

////////////////////////////////////

主控at89c51, 晶振6M

P0口编程数据输入输出，P2编程控制

*****AT89C51与ATmega8(L) 各脚连接*****

P0. 0-PB0, P0. 1-PB1, P0. 2-PB2, P0. 3-PB3, P0. 4-PB4, P0. 5-PB5, P0. 6-PC0

P0. 7-PC1

P2. 0-PD7 (PAGEL), P2. 1-PD6 (XA1), P2. 2-PD5 (XA0), P2. 3-PB6 (XTAL1)

P2. 4-PD4 (BS1), P2. 5-PD3 (_WR), P2. 6-PD2 (_OE), P2. 7-PD0 (RDY/BSY)

P1. 0-PC2 (BS2) P1. 2——控制reset (PC6)的12v电压p1. 2=1时输出12v

////////////////////////////////////

若不用不清楚请参考datsheet;

注：本程序只对reset引脚未被编程为I/O时有用。因时间问题(我目前还没

遇到误设RSTDISBL位)所以没写这段程序。若有需要的发

Email: crd2@21cn.com或fjzwf0068@sina.com.cn给我，需要人多的话我会

抽时间整理出电路图和更完善的单片机控制程序。

制作成功后你就狂烧熔丝和加密位吧!!!!!!!!!!!!!!!!!!!!!!

PAGEL BIT P2. 0 ;flash和E2PROM页装入

XA1 BIT P2. 1 ;XTAL编程动作位1

XA0 BIT P2. 2 ;XTAL编程动作位0

XTAL1 BIT P2. 3 ;晶振输入端

BS1 BIT P2. 4;字节选择1(0: 选低字节, 1: 选高字节)

_WR BIT P2. 5 ;写脉冲

_OE BIT P2. 6 ;输出允许(低有效)

RDY_BSY BIT P2. 7 ;0: 器件正在编程, 1: 器件就绪等待命令

LAMP BIT P1. 3 ;编程指示灯

_RST_12V BIT P1. 2 ;复位端12V电压

KEY BIT P1. 1 ;执行按键

BS2 BIT P1. 0 ;字节选择2(0: 选低字节, 1: 选高字节)

ORG 0000H

AJMP MAIN

ORG 30H

MAIN: MOV SP, #70H

CLR LAMP ;初始化指示

MOV P0, #0FFH

MOV P1, #0FFH

MOV P2, #0FFH

DJNZ R3, \$;延时大于100us

DJNZ R3, \$

DJNZ R3, \$

DJNZ R3, \$

SETB LAMP ;起动脉时完成

LOOP: ;JB KEY, LOOP

DJNZ R3, \$

DJNZ R3, \$

DJNZ R3, \$

DJNZ R3, \$

;JB KEY, LOOP

ACALL PROG_ST ;进入并行编程模式

ACALL CHIP_ERASE ;擦除加密位, 存储器

ACALL WRITE_FUSE ;写熔丝字节

ACALL READ_FUSE ;读熔丝字节/////////可不用(我调试时用的)

CLR _RST_12V

AJMP \$

AJMP LOOP

PROG_ST:

CLR _RST_12V ;RST=0V

CLR BS2

CLR XTAL1 ;XTAL1=0V

MOV R1, #20

PLS: CPL XTAL1 ;产生>6个的XTAL1时钟脉冲触发

NOP

NOP

DJNZ R1, PLS

ANL P2, #0E0H ;PAGEL, XTAL1, XA1, XA0, BS1==0

```

MOV PO, #80H ;送出芯片擦除命令
CLR XTAL1
SETB XTAL1
CLR XTAL1 ;输入正脉冲将命令装入芯片
NOP
SETB _WR
CLR _WR
SETB _WR ;输入一个负脉冲, 开始执行命令擦除芯片
SETB RDY_BSY ;置输入方式
NOP
NOP
JB RDY_BSY, $ ;等待RDY/BSY变高, 表示擦除完成
JNB RDY_BSY, $
SETB LAMP ;指示空闲
DJNZ R3, $
DJNZ R3, $
SETB _RST_12V
RET
;////////////////////写熔丝高低字节////////////////////
WRITE_FUSE:
CLR LAMP ;指示正忙
CLR XA0 ;设置XA1, XA0="10"允许装入命令
SETB XA1
CLR BS1
CLR BS2
MOV PO, #40H ;送出写熔丝命令
CLR XTAL1
SETB XTAL1
CLR XTAL1 ;XTAL1输入正脉冲将命令装入芯片 NOP
SETB XA0 ;设置XA1, XA0="01"允许装入数据
CLR XA1
NOP
CLR BS1 ;BS1, BS2="00"表示熔丝低字节 "10" 表示高字节
CLR BS2
MOV PO, #0E1H; DOBLEVEL=1, BODEN=1, SUT1=1, SUT1=0, SKSEL3=0,
;SKSEL2=0, SKSEL1=0, SKSELO=1(工厂设定值)
CLR XTAL1
SETB XTAL1
CLR XTAL1 ;XTAL1输入正脉冲数据装入芯片
SETB _WR
CLR _WR
SETB _WR ;_WR输入一个负脉冲, 开始执行命令
SETB RDY_BSY ;置输入方式
NOP
NOP
JB RDY_BSY, $ ;等待RDY/BSY变高, 表示完成
JNB RDY_BSY, $
;////////////////写高字节/////////////////
MOV PO, #0D9H ;'D9' 为出厂设定值
CLR XTAL1
SETB XTAL1
CLR XTAL1 ;XTAL1输入正脉冲数据装入芯片
SETB BS1
SETB _WR
CLR _WR
SETB _WR ;_WR输入一个负脉冲, 开始执行命令
SETB RDY_BSY ;置输入方式
NOP
NOP
JB RDY_BSY, $ ;等待RDY/BSY变高, 表示完成
JNB RDY_BSY, $
CLR BS1
DJNZ R3, $
SETB LAMP ;指示空闲
DJNZ R3, $
DJNZ R3, $
RET
;////////////////读熔丝及加密位(这段程序可不用是调试时用的)/////////////////
READ_FUSE:
CLR LAMP ;指示正忙
CLR XA0 ;设置XA1, XA0="10"允许装入命令 SETB XA1
CLR BS1
CLR BS2
MOV PO, #04H ;送出读熔丝和锁定位命令
CLR XTAL1
SETB XTAL1
CLR XTAL1 ;XTAL1输入正脉冲将命令装入芯片
NOP
CLR _OE
CLR BS2 ;读熔丝低字节
CLR BS1
MOV PO, #0FFH
MOV R7, PO
SETB BS2 ;读熔丝高字节
SETB BS1
MOV PO, #0FFH
MOV R6, PO
CLR BS2 ;读加密位
MOV PO, #0FFH
MOV R5, PO
SETB _OE
NOP
SETB LAMP
RET

```