7.4.1　10 位 AD/转换

源程序:SLAVR741.ASM

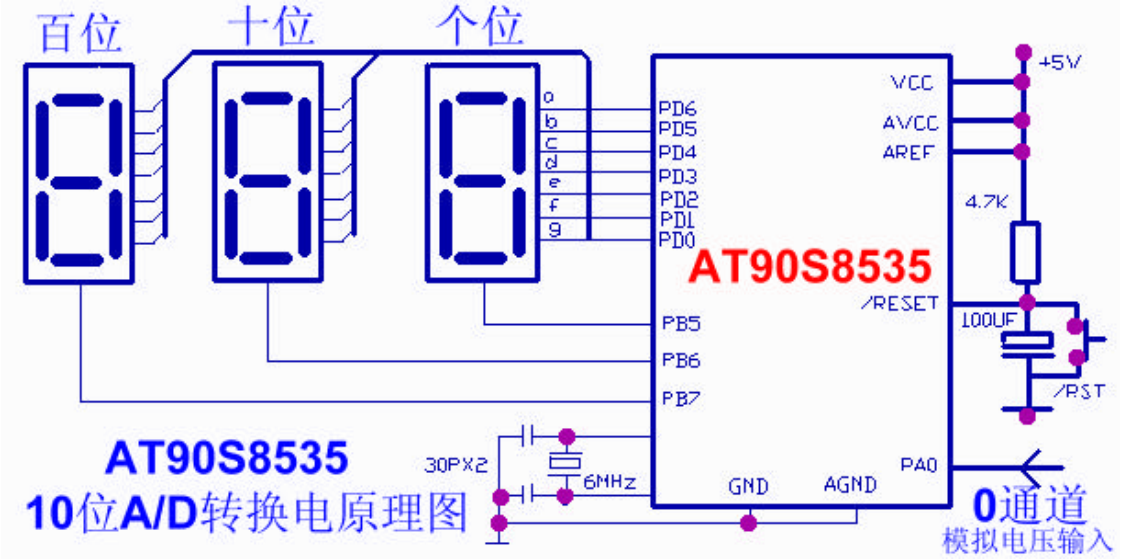
AVR 单片机中 AT90S8535 是功能较强的单片机.有如下特点:

1. AVR RISC 结构
2. AVR—高性能、低功耗 RISC 结构
   —118 条指令——大多数为单指令周期
   —32 个 8 位通用（工作）寄存器
   —工作在 8MHz 时具有 8MIPS 的性能
3. 数据和非易失性程序内存
   —4K/8K 字节的在线可编程 FLASH（擦除次数：1000 次）
   —256/512 字节 SRAM
   —256/512 字节在线可编程 EEPROM（寿命：100000 次）
   —程序加密位
4. 外围（Peripheral）特点
   —两个具有比较模式的可预分频（Prescale）8 位定时器/计数器
   —一个可预分频、具有比较、捕捉和两个 8/9/10 位 PWM 功能的 16 位定时器/计数器
   —片内模拟比较器
   —可编程的看门狗定时器（由片内振荡器生成）
   —8 通道 10 位 ADC
   —全双工 UART
5. 特别的 MCU 特点
   —上电复位电路
   —具有记数功能、有独立振荡器的实时时钟（RTC）
   —低功耗空闲、省电和掉电模式
   —内外部中断源
6. 4MHz、3V、20℃条件下的功耗:
   —工作模式：6.4mA
   —空闲模式：1.9mA
   —掉电模式：<1μA
7. I/O 和封装
   —32 个可编程的 I/O 脚
   —40 脚 PDIP、PLCC 和 TQFP 封装
8. 工作电压
   —2.7V-6.0V（AT90LS4434 和 AT90LS8535）
   —4.0V-6.0V（AT90S4434 和 AT90S8535）
9. 速度
   —0-4MHz（AT90LS4434 和 AT90LS8535）
   —0-8MHz（AT90S4434 和 AT90S8535）

```
;***************AVR 单片机实验测试程序 ************************
;* 标题:AT90S8535 的 10 位 A/D 转换器
;* 版本:      1.0
;* 最后更新日期:2000.08.08
```

```
;*  支援 E-mail:gzsl@sl.com.cn
;*  描述
;*  用 AVR Studio 调试软件窗口观察指令执行变化情况
;*  作者: SL.Z
;*  硬件电路及本程序实测调试通过
;*********************************************************************
```

★ 硬件电路必须按下图连接!

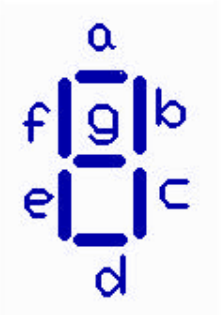



```
.include"8535def.inc"    ;AT90S8535 器件配置文件
.org $0000
    rjmp reset
.org $000e               ;INTER 中断入口地址
    rjmp inter
.defhledbyte=r19         ;存放 ADCH
.def    lledbyte=r18     ;存放 ADCL
.equ    label=$0100      ;字形表首址
.equ    distime=$38      ;显示次数
.def    temp=r16
reset:  ldi temp,$02     ;设堆栈指针$025F
    out sph,temp
    ldi temp,$5f
    out spl,temp
    ldi temp,$ff         ;B 口、D 口为输出
    out ddrb,temp
    out ddrd,temp
    out portd,temp       ;开通 LED 数目管,硬件设定高电平亮
    clr temp
    out ddra,temp        ;A 口为输入
    out porta,temp
main:   clt              ;清 T 标志
```

```
    sei                 ;开中断
    rcall conini        ;调用 A/D 初始化
wait:   brtc wait       ;等待中断
    clt
    ldi r20,distime
start:  ldi zh,high(label*2)
    mov temp,r19
    rcall outpd         ;取出显字形型
    cbi $18,7           ;显示百位 LED
    sbi $18,6           ; 关闭十位 LED
    sbi $18,5           ; 关闭个位 LED
    rcall delay         ;延时
    mov temp,r18
    swap temp
    rcall outpd
    cbi $18,6           ; 显示十位 LED
    sbi $18,7           ; 关闭百位 LED
    sbi $18,5           ; 关闭个位 LED
    rcall delay         ; 延时
    mov temp,r18
    rcall outpd
    cbi $18,5           ; 显示个位 LED
    sbi $18,6           ; 关闭十位 LED
    sbi $18,7           ; 关闭百位 LED
    rcall delay         ; 延时
    dec r20             ;显示次数减 1
    brne start          ;显示次数未完转,为 0 顺执
    rjmp main           ;返回主程序
inter:  in lledbyte,adcl;中断后,读取 ADC 数据寄存器低位数据
    in hledbyte,adch    ;取 ADC 数据寄存器高位数据
    cbi adsc,6          ;停止 ADC 转换
    set                 ;置 T 标志
    reti                ;中断返回
conini: ldi temp,$8d    ;设置 ADC 转换,中断触发,ADC 为单一模式且 32MCU 除频
    out adcsr,temp
    clr temp            ;
    out admux,temp      ;选择 0 通道
    sbi adcsr,6
    ret
delay:  ldi r27,$10     ;延时子程序
delay1: dec r26
    brne delay1
    dec r27
    brne delay1
```

```
    ret
outpd:  andi temp,$0f
    ldi zl,low(label*2)
    add zl,temp
    lpm                 ;取字符
    out portd,r0        ;输出字第
    ret
.cseg
.org $0100          ;字型表首地址
```

h	a	b	c	d	e	f	g	十六进制码	字形
1	1	1	1	1	1	1	0	FEH	0
1	0	1	1	0	0	0	0	B0H	1
1	1	1	0	1	1	0	1	EDH	2
1	1	1	1	1	0	0	1	F9H	3
1	0	1	1	1	0	1	1	B3H	4
1	1	0	1	1	0	1	1	DBH	5
1	1	0	1	1	1	1	1	DFH	6
1	1	1	1	0	0	0	0	F0H	7
1	1	1	1	1	1	1	1	FFH	8
1	1	1	1	0	0	1	1	F3H	9
1	1	1	1	0	1	1	1	F7H	A
1	0	0	1	1	1	1	1	9FH	B
1	1	0	0	1	1	1	0	CEH	C
1	0	1	1	1	1	0	1	BDH	D
1	1	0	0	1	1	1	1	CFH	E
1	1	0	0	0	1	1	1	C7H	F



硬件设定高电平

LED数目管点亮

```
.dw 0xb0fe,0xf9ed,0xdbb3,0xf0df
.dw 0xf3ff,0x9ff7,0xbdce,0xc7cf
```

7.4.2 步进电机控制程序

源程序:SLAVR742.ASM

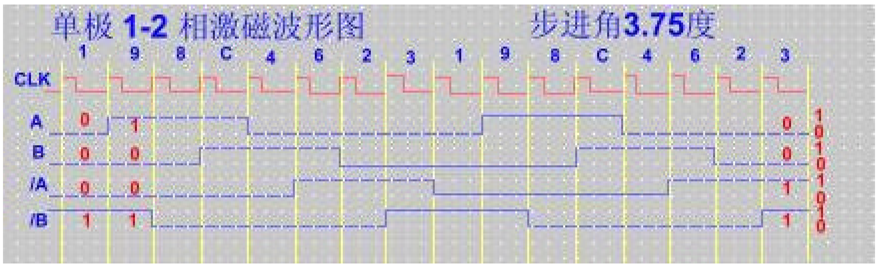
自从六十年代初期步进电机面世以来,在过去几年它的重要性大大提高了。它用来驱动时钟和其他采用指针的仪器,打印机、绘图仪、磁盘光盘驱动器、各种自动控制阀、各种工具,还有机器人等的机械装置。关于马进电机工作原理请参考有关资料。

下面用单极 1-2 相激磁方法步进电机做实验,即 1 极、2 极、1 极、2 极、....极以次循环,如何用单极二相激该方法控制步进电机,由读者或用户自行编制程序实验。

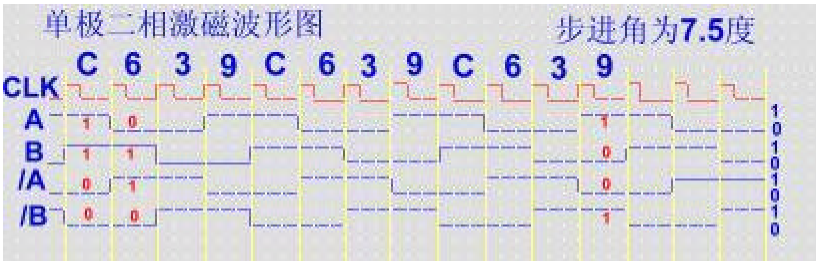
;实验选用 4.5V 步进电机,用 5V 即可,实验时节省一组步进电机驱动电源;

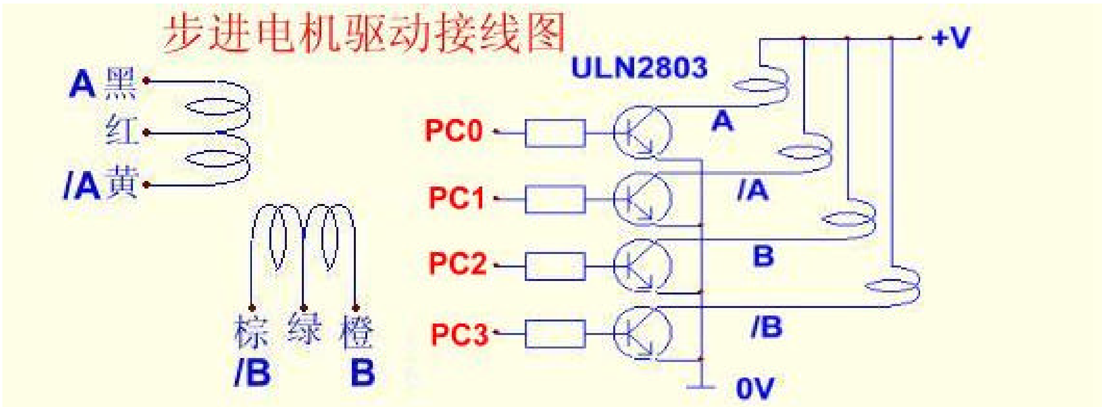
;型号:MA82135; 相数:2 相; 电压:4.5V; 电流/相:0.12A; 电阻欧姆:34Ω/相; 重量:30g

;
;
;
;
;
;
;



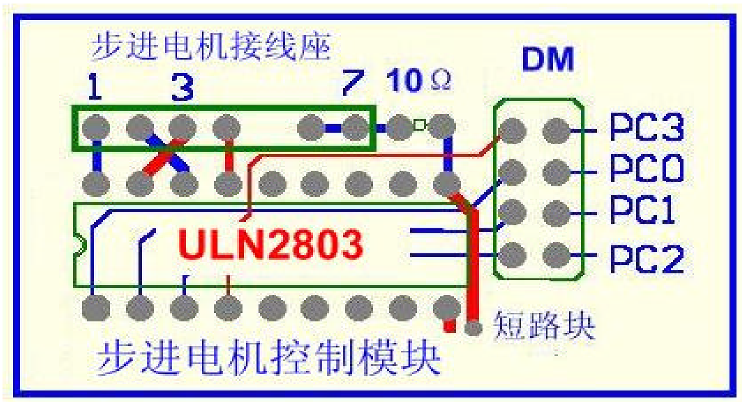
;
;
;
;
;
;
;
;





;
;
;
;
;
;
;
;
;
;

;
;
;
;
;



```
;*****************************************
;*    步进电机控制程序(单极 1-2 相)          *
;*                                         *
;*SLAVR742.ASM                             *
;*use ULN2803 ;使用 PC0-PC3 驱动步进电机     *
;*use 11-17new bord                        *
;*****************************************
.include"8515def.inc"
.def   temp    =r16
.def   dt      =r19
.def   np      =r17
.def   step    =r18
.def   TStep   =r20
.def   cnt     =r21
.equ   turntab=0x0200
.org  $0000
       rjmp   RESET
.cseg
.org  0x010
RESET:
       ldi  temp,low(RAMEND);设堆栈
       out  SPL,temp
       ldi  temp,high(RAMEND)
       out  SPL+1,temp
       ser   TEMP        ;C 口设置为输出
       OUT   ddrc,TEMP
       ldi   zl,low(turntab*2)   ;步进电机旋转资料指针
       ldi   zh,high(turntab*2)
       ldi   np,4
       ldi   temp,$44
       out   portc,temp      ;初始化
       ldi   TStep,$25
       rcall delay
       ldi   cnt,10
       clt
rep:   ldi   step,192
       ldi   TStep,1     ;1--255
       rcall turn
       dec   cnt
       brne  rep
loop:  nop
       rjmp  loop
;*************************************************************
;  t=1  uncircle turn   ;T=1 逆时针转                          *
```

```
;  t=0   circle  turn   ;T=0 顺时针转                          *
;   96  step  a  turn                                        *
;  TStep is  time  of a step  ;                              *
;************************************************************
turn:  brts uncircle;判转向
       inc   np      ;正转
       cpi   np,8
       brne  next
       clr   np
next:  push  zl
       add   zl,np
       lpm
       out   portc,r0
       pop   zl
       rcall delay
       dec   step
       brne  turn
       ret
uncircle:       ;反转
       dec   np
       cpi   np,$ff
       brne  next
       ldi   np,$07
       rjmp  next
delay:  push   TStep;延时子程序
del1:   ldi    dt,70
del2:   push   dt
del3:   dec    dt
        brne   del3
        pop    dt
        dec    dt
        brne   del2
        dec    TStep
        brne   del1
        pop    TStep
        ret
.org   turntab
;       0    1    2    3    4    5    6    7 ;步进电机旋转资料表
.db    0x11,0x99,0x88,0xcc,0x44,0x66,0x22,0x33
```

7.4.3 测脉冲宽度

源程序:SLAVR743.ASM

```
;本程序略加修改可应用于测速,测距,测频率等,用 LED 数目管显示
;本程序在 SL-AVR 开发下载实验器上验证通过
;硬件连接:AT90S8515 的 ICP 作输入,测量输入脉冲宽度时间,以μS/秒计算,
;由六位 LED 十进制显示
.include"8515def.inc"
    rjmp reset
.def   cnt1d =r03              ;cnt1d-cnt5d:存放十进制数
.def   cnt2d =r04              ; 存放形式(压缩 BCD 码)
.def   cnt3d =r05              ;从 cnt1 到 cnt5 依数存放个位、十位.....
.def   cnt4d =r06
.def   cnt5d =r07
.def   tmp1h =r08
.def   tmp2h =r09              ;tmp1h,tmp2h:存放第一次捕捉的 timer1 的值
.def   tim1l =r10
.def   tim1h =r11              ;存放 timer1 溢出的次数
.def   tmp5h =r12
.def   tmp6h =r13              ;tmp5h-tmp8h:存放二次捕捉 timer1 的差值
.def   tmp7h =r14
.def   tmp8h =r15
.def   temp  =r16              ;暂存器
.def   cnt4  =r17
.def   cntn  =r18
.def   cntn1 =r19
.def   tempn =r20              ;利用 tempn 的 d0 位判断是那一次捕捉
.def   tempn1=r21
.cseg
.org 0x03                      ;icp 触发中断向量
icpt1:
    rjmp  captr
.org 0x06                      ;timer1 溢出中断向量
intt1:
    rjmp  calts
.cseg
.org 0x10
captr:                         ;icp 触发中断子程序
    wdr
    sbrc  tempn,  00          ;判断是那一次捕捉
    rjmp  cap2
cap1:                          ;第一次捕捉处理
    in    tmp1h, icr1l
    in    tmp2h, icr1h        ;把捕捉的 timer1 的值放在 tmp1h 和 tmp2h
```

```
    ori   tempn, $01
    ldi   r16,   0b10001000
    out   timsk, r16          ;致 timer1 中断和捕捉中断
    reti
cap2:                         ;第二次捕捉处理
    in    tmp5h, icr1l
    in    tmp6h, icr1h        ;把捕捉的 timer1 的值放在 tmp5h 和 tmp6h
    rcall transd              ;
    rcall htd4                ;把 tmp5h-tmp8h 转成十进制
    ldi  tempn, $00          ;令 tempn d0=0
    rcall clrtmm
    ldi   r16,   0b00000000
    out   timsk, r16          ;除 timer1 中断和致捕捉中断
    ret
calts:                        ;timer1 溢出中断子程序
    inc   tim1l
    breq  ov
    rjmp  b
ov:     inc   tim1h
    brne  b
    set
b:  reti
transd:                          ;二次捕捉 timer1 的差值处理
    clc
    sbc   tmp5h, tmp1h           ;结果放在 tmp5h-tmp8h
    sbc   tmp6h, tmp2h
    brcc  posv
    dec   tim1h
    dec   tim1l
posv:
    mov   tmp7h, tim1l
    mov   tmp8h, tim1h
    ret
reset:
    ldi   temp,  low(ramend)
    out   spl,   temp
    ldi   temp,  high(ramend)      ;设置堆栈
    out   spl+1, temp
    ldi   temp,  $0e                ;设定 wdtcr 中 wde=1
    out   wdtcr, temp
    cli
    ldi   temp,  $ff               ;初始化数码管状态
    out   ddrb,  temp              ;B 口:数码管数据输出
    out   ddrd,  temp              ;D 口:pd0-pd5 为数码管片选
```

```
    out   portd, temp              ;数码管低电平选中
    ldi   temp,  $00
    out   portb, temp              ;共阴极,数码管全灭
    ldi   tempn1,00
    rcall clrtm
    ldi   temp,  $00
    out   tccr1a,temp
    out   tccr1b,temp
    out   tcnt1h,temp              ;装入计数值
    out   tcnt1l,temp              ;tcnt1h=tcht1l=00
    clt
    sei                            ;开中断
    ldi   r16,   0b00001000
    out   timsk, r16
    ldi   r16,   0b11000010
    out   tccr1b,r16               ;开始计数
reptw:
    sbrs  tempn1,00
    rjmp  reptw
    rjmp  reset
clrtm:
    clr   tmp1h
    clr   tmp2h
    clr   tmp5h
    clr   tmp6h
    clr   tmp7h
    clr   tmp8h
clrtmm:
    clr   tim1h
    clr   tim1l
    clr   cnt4
    clt
    ret
htd4:                              ;把 tmp5h-tmp8h 转成十进制子程序
    ldi   cntn,  32
    clr   cnt1d
    clr   cnt2d
    clr   cnt3d
    clr   cnt4d
    clr   cnt5d
    clc
loopd:
    rol   tmp5h
    rol   tmp6h
```

```
    rol   tmp7h
    rol   tmp8h
    rol   cnt1d
    rol   cnt2d
    rol   cnt3d
    rol   cnt4d
    rol   cnt5d
    dec   cntn
    breq  end
    rcall adjn
    rjmp  loopd
end:                        ;在数码管显出十进制数
    ldi   cntn,  $ff
end1:
    ldi   cntn1, $ff
end2:   mov   temp,  cnt1d
    andi  temp,  $0f        ;显示个位
    rcall a
    cbi   portd, 00
    nop
    sbi   portd, 00
    mov   temp,  cnt1d
    andi  temp,  $f0        ;显示十位
    swap  temp
    rcall a
    cbi   portd, 01
    nop
    sbi   portd, 01
    mov   temp,  cnt2d
    andi  temp,  $0f        ;显示百位
    rcall a
    cbi   portd, 02
    nop
    sbi   portd, 02
    mov   temp,  cnt2d
    andi  temp,  $f0        ;显示千位
    rcall a
    swap  temp
    rcall a
    cbi   portd, 03
    nop
    sbi   portd, 03
    mov   temp,  cnt3d
    andi  temp,  $0f        ;显示万位
```

```
    rcall a
    cbi   portd, 04
    nop
    sbi   portd, 04
    mov   temp,  cnt3d
    andi  temp,  $f0        ;显示十万位
    swap  temp
    rcall a
    cbi   portd, 05
    nop
    sbi   portd, 05
    dec   cntn1
    brne  end2
    dec   cntn
    brne  end1
    ldi   tempn1,$01
    ret
a:
    ldi   zh,    high(zk*2)
    ldi   zl,    low(zk*2)
    add   zl,    temp
    lpm
    out   portb, r0
    ret
adjn:
    push  cntn
    mov   cntn,  cnt1d
    rcall adjd1
    mov   cnt1d, cntn
    mov   cntn,  cnt2d
    rcall adjd1
    mov   cnt2d, cntn
    mov   cntn,  cnt3d
    rcall adjd1
    mov   cnt3d, cntn
    mov   cntn,  cnt4d
    rcall adjd1
    mov   cnt4d, cntn
    mov   cntn,  cnt5d
    rcall adjd1
    mov   cnt5d, cntn
    pop   cntn
    ret
adjd1:
```

```
    ldi   tempn, 3
    add   tempn, cntn
    sbrc  tempn, 3
    mov   cntn,  tempn
    ldi   tempn, $30
    add   tempn, cntn
    sbrc  tempn, 7
    mov   cntn,  tempn
    wdr
    ret
.equ    zk=0x0200
.org    zk                          ;字形表
.db     0x03f,0x006,0x05b,0x04f
.db     0x066,0x06d,0x07d,0x007
.db     0x07f,0x06f,0x077,0x07c
.db     0x039,0x05e,0x071
```

7.4.4 LCD 显示 8 字循环

源程序:SLAVR744.ASM

LCD 显示器 1602AT(S)R 简介:

LED 显示器模块 1602AT(S)R 为 2X16 字符。含有 5*10 或 5*7 点 LCD 共 12*16=192 种 CG 显示字形及双组 8 个自由利用软件设定(CGRAM)的 5*8 图点字形,因此除内部固定 192 种字形外,再加上此 16 个可自由设定图字型等共计 208 种字图形如字符代码表所示。因 5*8 个点输入设定故 5 个点仅占用 D4-D0 的 5 位,而 D7-D5 则可为任意值,第八行值为游标地址,因此共八行占八个地址组成一个字形及标示游标地址,总共八个设定字图形,因此占有 $8*8=2^6$ 个地址,CG 地址设定值为 D5-D0。

LCD 引脚功能说明

1. GND:电源地,0V;
2. VCC 电源+5V;
3. VLC:LCD 驱动电压 0V-5V 对比度调节电压;
4. RS 寄存器选择信号;

   RS=0:

   指令寄存器 IR 写入(WRITE);

   (1) 忙(BUSY FLAG)读取(READ);

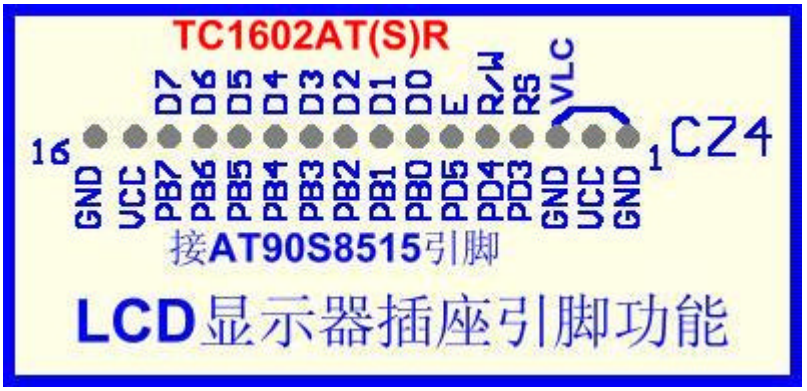
   (2) 地址计数器(ADDRESS COUNTER)AC 读取(READ);

   RS=1:数据寄存器(DATA REGISTER)读取及写入(READ/WRITE);
5. R/W 读/写控制信号(READ/WRITE):R/W=1 读取(READ)、R/W=0 写入(WRITE);
6. E(ENABLE)片使能信号,作写数据控制,下降沿触发;

7~14 脚为 DB0~DB7 八位数据总线,三态双向,若作为 4 位传送时应令:

DL=0,以 DB4-DB7 作传送将 8 位数据分二次传送;

15. 一般不用(空),如有背光 LED,则接 VCC;
16. 一般不用(空),如有背光 LED,则接 GND;



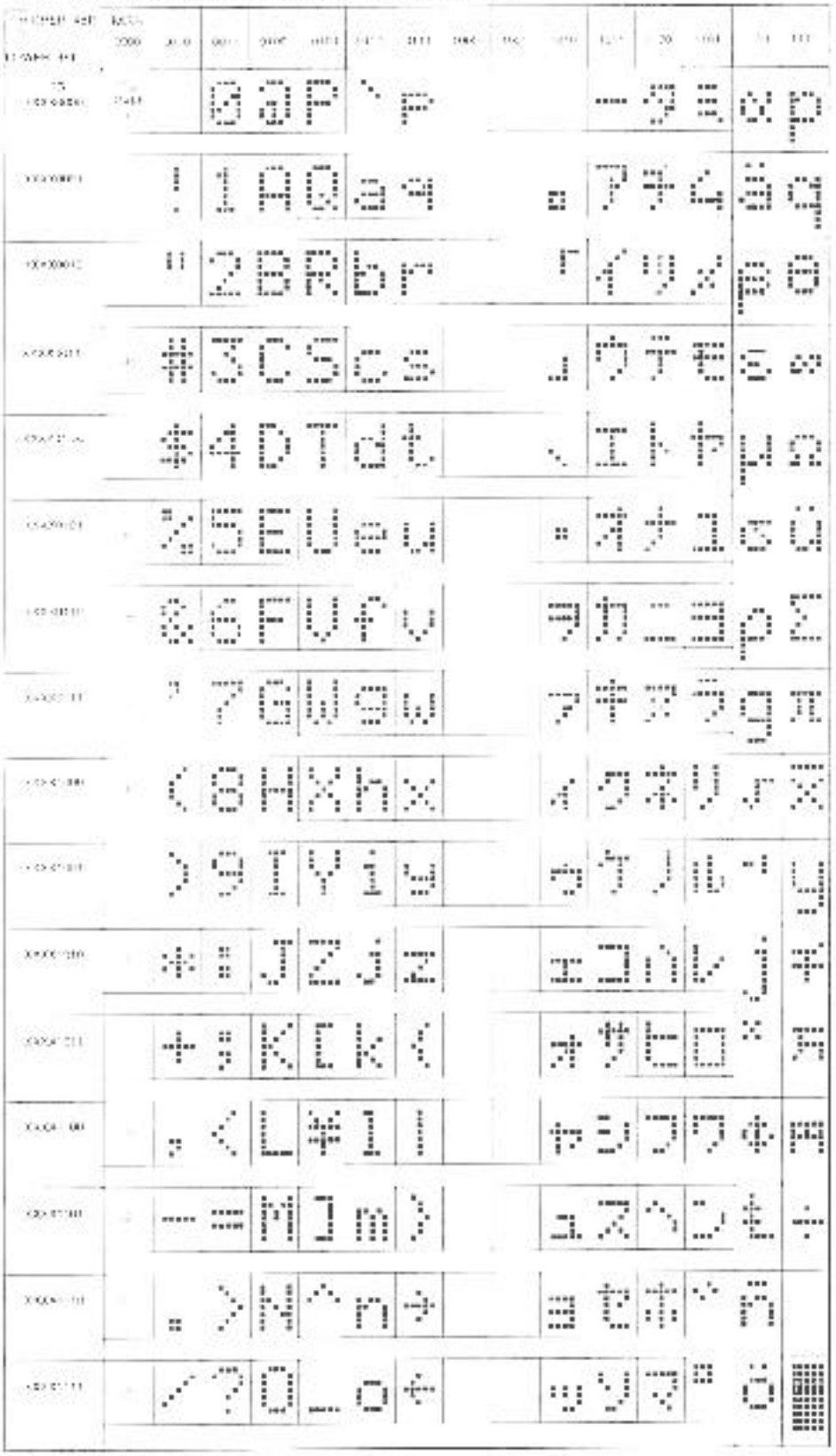
LCDTC1602 CG RAM **字形结构设定输入表**

字形码(DD RAM **数据**									CG RAM **地址**							**字形图样**(CG RAM)**数据**								
7	6	5	4	3	2	1	0		5	4	3	2	1	0		7	6	5	4	3	2	1	0
高位							**低位**		**上位**					**下位**		**上位**			(5*7 **字形**)				**下位**
												0	0	0		*	*	*	1	1	1	1	0
												0	0	1					1	0	0	0	1
												0	1	0					1	0	0	0	1
												0	1	1					1	1	1	1	0
0	0	0	0	*	0	0	0		0	0	0	1	0	0					1	0	1	0	0
												1	0	1					1	0	0	1	0
												1	1	0					1	0	0	0	1
												1	1	1		*	*	*	0	0	0	0	0
												0	0	0		*	*	*	1	0	0	0	1
												0	0	1					0	1	0	1	0
												0	1	0					1	1	1	1	1
												0	1	1					0	0	1	0	0
0	0	0	0	*	0	0	1		0	0	1	1	0	0					1	1	1	1	1
												1	0	1					0	0	1	0	0
												1	1	0					0	0	1	0	0
												1	1	1		*	*	*	0	0	0	0	0
												0	0	0		*	*	*					
												0	0	1									
0	0	0	0	*	1	1	1		1	1	1												
												1	0	1									
												1	1	0									
												1	1	1									

LCD 指令表

指 令	指 令 码										说 明	执行周期 f_{osc}= 250Khz
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
清屏	0	0	0	0	0	0	0	0	0	1	请除屏幕,置 AC 为 0, 光标回位	1.64ms
光标返回	0	0	0	0	0	0	0	0	1	*	DDRAM 地址为 0,显示回原位,DDRAM 内容不变	1.64ms
设置输入方法	0	0	0	0	0	0	0	1	I/D	S	设置光标移动方向并指定显示是否移动	40μs
显示开关	0	0	0	0	0	0	1	D	C	B	设置显示开或关(D),光标开关(C),光标所在字符闪烁(B)	40μs
移位	0	0	0	0	0	1	S/C	R/L	*	*	移动光标及整体显示,同时不改变 DDRAM 内容	40μs
功能设置	0	0	0	0	1	DL	N	F	*	*	设置接口数据(DL)、显示行数(L)、字符字体(F)	40μs
CGRAM 地址设置	0	0	0	1	ACG						设置 CGRAM 地址,设置后发送接收数据	40μs
DDRAM 地址设置	0	0	1	ADD							设置 DDRAM 地址,设置后发送接收数据	40μs
忙标志/读地址计数器	0	1	BF	AC							读忙标志(BF)标志正在执行内部操作并读地址计数器内容	40μs
CGRAM/DDRAM 数据写	1	0	写数据								从 CGRAM 或 DDRAM 写数据	40μs
CGRAM/DDRAM 数据读	1	1	读数据								从 CGRAM 或 DDRAM 读数据	40μs
	I/D=1:增量方式; I/D=0:减量方式 S=1:移位 S/C=1:显示移位; S/C=0 光标移位 R/L=1:右移; R/L=0 左移 DL=1:8 位; DL=0:4 位 N=1:2 行; N=0:1 行 F=1:5*10 字体; F=0:5*7 字体 BF=1:执行内部操作; BF=0:可接收指令										DDRAM:显示数据 RAM CGRAM:字符发生器 RAM ACG:CGRAM 地址 ADD:DDRAM 地址及光标地址 AC: 地址计数器用于 DDRAM 和 CGRAM	执行周期随主频改变而改变例如当 fcp 或 fosc=270 khz 时: 40μs x 250/270= 37μs

字符点阵LCD模块字符代码表



源程序:SLAVR744.ASM

```
.include"8515def.inc"
.def temp=r16
.def temp1=r17
.def temp2=r18
.def cnt=r20
.def cnt1=r21
.org $0000
    rjmp reset
.org $0030
reset:  ldi temp,low(ramend);设置堆栈指针。
    out spl,temp
    ldi temp,high(ramend)
    out sph,temp
    ldi temp,$ff            ;设置 D 口输出，B 口作输入。
    out ddrd,temp
    out portd,temp
    clr temp
    out ddrb,temp
    out portb,temp
    rcall syset             ;调用系统设置。
lp8:clr cnt1                ;循环程序。
lp81:   clr cnt
    ldi temp1,$80           ;设置第一行显示寄存器起址。(第二行为$a8)
    rcall contd
lp82:   cp cnt1,cnt
    brne lp83

    ldi temp1,$38       ;字形 8 的代码为$38。
lp84:   rcall writd
    inc cnt
    cpi cnt,$10
    brne lp82
    ldi temp,$55            ;设置延时常数。
    rcall delay
    inc cnt1
    cpi cnt1,$10
    brne lp81
    rjmp lp8
lp83:   ldi temp1,$20
    rjmp lp84

CONTD:   LDI    TEMP,0B00110000   ;写控制字入 LCD 中
     OUT    PORTD,TEMP
```

```
      RCALL   DELT3
      CBI     PORTD,$05          ;使 E=0 ，LCD 片选有效
      RCALL   DELT3
      SBI     PORTD,$05
BUSYY:    WDR
      SBIC    PINB,$07           ;读取 DB7=PINB7 是否为 0，为 0 则非忙跳过一行
      RJMP    BUSYY              ;DB7=1 为忙，跳回 BUSYY 再等待 DB7=0 以写入
      LDI     TEMP,0b00100000    ;写入数据写入控制字
      OUT     PORTD,TEMP
      RCALL DELT3                ;延时以免 AVR 速度太快而使 LCD 无法工作
      LDI     TEMP,$ff           ;设定 B 口为输出
      OUT     DDRB,TEMP
      OUT     PORTB,TEMP1        ;要写入 LCD 的数据 TEMP1 输出到 PORTB
      WDR
      CBI     PORTD,$05
      RCALL DELT3
      LDI     TEMP,0B00111000
      OUT     PORTD,TEMP
      CLR     TEMP
      OUT     DDRB,TEMP
      OUT     PORTB,TEMP
      RET

WRITD:    LDI TEMP,0B00110000      ;写数据入 LCD 中
      OUT PORTD,TEMP
      RCALL DELT3                ;延时以免 AVR 速度太快而使 LCD 无法工作
      CBI PORTD,$05              ;使 E=0，  LCD 片选有效
      RCALL DELT3
      SBI PORTD,$05
BUZY1:    WDR
      SBIC PINB,$07              ;读取 DB7=PINB7 是否为 0，为 0 则非忙跳过一行
      RJMP BUZY1                 ;DB7=1 为忙，跳回 BUSY1 再等待 DB7=0 以写入
      LDI TEMP,0B00101000      ;写控制字入 LCD 中
      OUT PORTD,TEMP
      OUT PORTB,TEMP1            ;要写入 LCD 的数据 TEMP1 输出到 PORTB
      LDI TEMP,$ff             ;设定 B 口为输入
      OUT DDRB,TEMP
      CBI PORTD,$05              ;使 E=0，  LCD 片选有效
      RCALL DELT3                ;延时以免 AVR 速度太快而使 LCD 无法工作
      LDI TEMP,0B00111000        ;写控制字入 LCD 中
      OUT PORTD,TEMP
      RCALL DELT3
      CLR TEMP                   ;PORTB 为输入
      OUT DDRB,TEMP
```

```
     OUT PORTB,TEMP          ;PORTB 为三态输入
     RET
syset:  ldi temp1,$01        ;清屏设定
    rcall contd
    ldi temp,$50             ;设置时间常数
    rcall delay
    ldi temp1,$38            ;2 行 5*7 显示设定
    rcall contd
    ldi temp1,$06            ;自动增量，显示不移位
    rcall contd
    ldi temp1,$0c            ;字形开关 ON，光标开关 OFF
    rcall contd
    ret
DELT3:  ldi temp2,$24
DT111:  wdr
    dec temp2
    brne dt111
    ret
delay:              ;延时子程序略
```

7.4.5 LED 电脑时钟

源程序:SLAVR745.ASM
硬件连接见:3.3 AVR 单片机开发下载实验器;

本程序若直接按 shife+exec 即从 00:00:00 开始计时。

本程序下载后(或上电后)，LED 显示 00:00:00 等待设置,请您从键盘上输入时、分、秒,要求输入位由小数点作光标提示,输入正确时间后,按执行键(SHIFT+EXEC)执行,电脑钟开始计时。

1. 请你如何修改程序,可当秒表用?
2. 又如何修改程序到点发出报时声?

```
  ;本程序采用 T0，1/1024 分频，设定一次中断为 25MS,40 次中断为 1 秒。
  .include"8515def.inc"
 .def   TEMP  =r16
 .def   TEMP1 =r17
 .def   temp2 =r18
 .def   temp3 =r19
 .def   CNT   =r20
 .def   SCNN  =r21
 .def   KSNI  =r22
 .def   SCNDP =r23
 .def   KEYN  =r24
 .def   cnt1 =r25
 .def   hour=r24
 .def   minute=r22
```

```
 .def   second=r21
 .equ label=$0f00        ;字形表首址
.org $0000
    rjmp reset
.org $007
intt0:  ldi temp,104     ;因 25ms 内差 40us 故补上 40/(1/8)即 320 个 CK。
bu: dec temp             ;因中断需 4CK 这样:4+104*(1+2)+1+1+1+1=320。
    brne bu
    nop
    inc cnt1             ;cnt1 计数 40 次为 1 秒钟。
    ldi temp,256-195     ;计数(256-195)次才产生 1 次中断。
    out tcnt0,temp       ;CK/1024 分频,这样一次中断需 25ms。
    rjmp recog
.org $030
reset:
    ldi temp,low(ramend) ;设置堆栈指针。
    out spl,temp
    ldi temp,high(ramend)
    out sph,temp
    clr r2           ;清工作寄存器。
    clr r3
    clr r4
    clr r5
    clr r6
    clr r7
    clr zh
    clr xh
    clr yh
    clr keyn
    clr second
    clr minute
    clr hour
    clr cnt
    clr yh
    ldi temp,$80
    mov r8,temp      ;R8=$80
    ldi yl,$60       ;设置显示内存地址指针 Y 为$0060.
    rcall disram     ;调用 DISRAM。
    ld temp,y
    ldi temp1,$80
    add temp,temp1
    st y,temp
scanad: ldi temp,$07
    ldi yl,$60
```

```
 scann: rcall scan1      ;调用键扫显示子程序 SCAN1。
    brts scann
 scank: rcall scan1
    brtc scank
    rcall scan1
 scans: nop
        cpi keyn,$10     ;KEYN=$10 转 EXEC。
    brcc exec
    rcall wraddram       ;调用 WRADDRAM。
    dec temp     ;TEMP 减 1。
    cpi temp,$01
    brne scann       ;TEMP=1 则转 SCANN
    rjmp scanad
exec:   mov temp1,r7;把 r7,r6 的两个十进制换成一个十六进制入 hour 中
    mov temp,r6
    rcall dechex
    mov hour,temp
    mov temp1,r5;把 r5,r4 的两个十进制换成一个十六进制入 minute 中
    mov temp,r4
    rcall dechex
    mov minute,temp
    mov temp1,r3;把 r3,r2 的两个十进制换成一个十六进制入 second 中
    mov temp,r2
    rcall dechex
    mov second,temp
    ldi temp,$05     ;T0 设置为 CK/1024 分频。
    out tccr0,temp
    ldi temp,256-195
    out tcnt0,temp       ;装载 T0 时间常数。
    ldi temp,$ff     ;设置 b 口,d 口为输出
    out ddrb,temp
    out ddrd,temp
    sei          ;开中断总开关
    ldi temp,$02
    out timsk,temp       ;允许 t0 中断。
display:rcall disram     ;调用 disram
    clr yh           ;设置显示内存地址指针 Y 为$0060

    ldi yl,$60
    ldi scndp,$df        ;设置扫描显示码 SCNDP 起址 0B11011111.
agdis:  ld r1,y+
    cpi yl,$62
    brne npoint
    add r1,r8
```

```
npoint: cpi yl,$64
    brne next
    add r1,r8
next:   out portb,r1     ;把 R1 送 B 口显示
    out portd,scndp      ;扫亮某个数码管
    sec          ;C=1
    ror scndp        ;右移 SCNDP
    rcall delay      ;延时
    cpi yl,$66
    brne agdis       ;未扫亮最后一位继续
    rjmp display
recog:  cpi cnt1,40      ;40 次中断为 40*25ms=1 秒
    brne inthome     ;40 次中断未到转 inthome
    clr cnt1     ;40 次中断到则清 cnt1
    inc second       ;秒寄存器加 1
    cpi second,60
    brne change      ;秒寄存器未满转 change
    clr second       ;否则清秒寄存器
    inc minute       ;分寄存器加 1
    cpi minute,60
    brne change      ;分寄存器未满转 change
    clr minute       ;否则清分寄存器
    inc hour     ;时寄存器加 1
    cpi hour,24
    brne change      ;时寄存器未满转 change
    clr hour     ;否则清时寄存器
    reti         ;中断返回
change: mov temp,second  ;把 second 中的十六进制转换成二个十进制数存入 r3,r2 中
    rcall hexdec
    mov r3,temp1
    mov r2,temp
    mov temp,minute ;把 minute 中的十六进制转换成二个十进制数存入 r5,r4 中
    rcall hexdec
    mov r5,temp1
    mov r4,temp
    mov temp,hour    ;把 hour 中的十六进制转换成二个十进制数存入 r7,r6 中
    rcall hexdec
    mov r7,temp1
    mov r6,temp
inthome:reti     ;中断返回
hexdec: clr temp1      ;把 temp 中的十六进制转成二个十进制入 temp1,temp 中的子程序
hexdec1:subi temp,10
    brcs negs
    inc temp1
```

```
    rjmp hexdec1
negs:   subi temp,$f6
    ret             ;子程序返回
dechex: push temp   ;把 temp1,temp 的两个十进制数转换成一个十六进制入 temp 中
    ldi temp2,$0a
    clr temp
dechex1:cpi temp1,$00
    breq dh
    dec temp1
    add temp,temp2
    rjmp dechex1
dh: pop temp1
    add temp,temp1
    ret         ;子程序返回
disram: push yl     ;压栈保护
    push zl
    push xl
    ldi zh,high(label*2) ;Z 指针指向字形表首址 label*2
    ldi zl,low(label*2)
    clr xh
    ldi xl,$60
    ldi yl,$07
ramag:  ld temp2,y      ;y 为间址的内容送 temp2
    dec yl
    mov zl,temp2
    lpm
    st x+,r0    ;把 r0 的内容送到$0060-$0065 中
    cpi xl,$66
    brne ramag
    pop xl
    pop zl
    pop yl          ;退栈
    ret     ;子程序返回
wraddram:push temp      ;读键存入显示内存及寄存器中。
    clr zh
    mov zl,temp
    st z,keyn
    ldi zh,high(label*2)
    mov zl,keyn
    lpm
    st y+,r0
    cpi yl,$66
    brne pointc
    ldi yl,$60
```

```
pointc: ld temp2,y
    ldi temp3,$80
    add temp2,temp3
    st y,temp2
    pop temp
    ret     ;子程序返回
delay:  push temp    ;延时子程序
lp1:ldi temp2,$10
lp2:dec temp
    brne lp2
    dec temp2
    brne lp2
    pop temp
    ret     ;子程序返回
 SCAN1:         push xh          ;键盘扫描显示子程序(注释从略,见 7.3.3)。
        PUSH XL
        PUSH TEMP1
        PUSH TEMP
        LDI XL,$60
            SET
            LDI  SCNN,$00
            LDI  SCNDP,0B11011111
            LDI  CNT,$06
            LDI  KSNI,0B11110111
      COL1: LDI  TEMP,$FF
            OUT  DDRb,TEMP
            OUT  DDRC,TEMP
        OUT  PORTC,TEMP
            OUT  DDRd,TEMP
        OUT  PORTd,SCNDP
            LD   R1,X+
            OUT  PORTb,R1
            RCALL  DELAY
            MOV  TEMP,CNT
            SUBI  TEMP,$03
            BRCS  NOSK
            LDI  TEMP1,$04
            LDI  TEMP,0B00001111
            OUT  DDRc,TEMP
            OUT  PORTc,KSNI
            RCALL  DELYT
            IN   TEMP,PINc
            ANDI  TEMP,0B11110000
            SWAP  TEMP
```

```
          KROW: SEC
        ROR  TEMP
            BRCS  NOKEY
            CLT
            MOV   KEYN,SCNN
        SBIS PINd,$07
        ADIW KEYN,$10
         NOKEY: INC  SCNN
            DEC  TEMP1
            BRNE  KROW
            SEC
            ROR  KSNI
          NOSK: SEC
            ROR  SCNDP
            DEC  CNT
            BRNE  COL1
            LDI   TEMP,$FF
            OUT   DDRC,TEMP
        OUT   PORTC,TEMP
        POP TEMP
        POP TEMP1
        POP XL
        pop xh
        RET
         delyt: ldi temp3,$20
       dt31:dec temp3
        brne dt31
        ret

.cseg
.org $0f00              ;字形表
 .dw 0x063f,0x4f5b,0x6d66,0x077d
 .dw 0x6f7f,0x7c77,0x5e39,0x7179
```

7.4.6 测频率

```
;测频率,信号从 AT90S8515 的 ICP 引脚输入,最大值为为 999999Hz/mS
;源程序:SLAVR746.ASM，在 SL-AVR 开发实验器验证通过
.include"8515def.inc"
    rjmp   reset
.def    temp   = r16                ;暂存器
.def    cnt1d  = r17
.def    cnt2d  = r18                ;cnt1、dcnt2d 和 cnt3d 存放结果的十进制
.def    cnt3d  = r19
.def    count  = r20
.def    cnt    = r21
.def    res1   = r22
.def    res2   = r23                ;res1、res2 和 res3 存放结果的十六进制
.def    res3   = r24
.def    dt     = r25
.def    ovfl   = r26
.def    aa     = r27
.org    0x003                       ;icp 触发中断向量
    rjmp   captr
.org    0x007                       ;timer0 触发中断向量
    rjmp   interru
captr:                              ;icp 触发中断子程序
    brts  b
    inc   res1
    ldi   temp,  0b00000101
    out   tccr0, temp           ;开 timer0
    ldi   temp,  0b00001010
    out   timsk, temp           ;致 timer0 中断和捕捉中断
    ldi   temp,  0b11000000
    out   tccr1b,temp
    set
    reti
b:
    set
    inc   res1                  ;开始计数
    brne  c
    inc   res2
    brne  c
    inc   res3
    cpse  res3,  ovfl           ;溢出处理
    rjmp  c
    rjmp  over1
c:
```

```
    ldi   temp,  0b00001010
    out   timsk, temp
    ldi   temp,  0b11000000
    out   tccr1b,temp
    reti
interru:                                ;timer0 溢出中断子程序
    dec   cnt
    breq  over
    ldi   temp,   0b00001010
    out   timsk,  temp
    reti
over:
    rcall htd3
over1:
    rcall sys
    reti
reset:
    ldi   temp,  low(ramend)
    out   spl,   temp
    ldi   temp,  high(ramend)       ;设置堆栈
    out   spl+1, temp
    ldi   temp,  $ff                ;初始化数码管状态
    out   ddrb,  temp               ;B 口:数码管数据输出
    out   ddrd,  temp               ;D 口:pd0-pd5 为数码管片选
    ldi   temp,  $00
    out   portb, temp               ;共阴极,数码管全灭
    out   portd, temp
    ldi   cnt1d, 00
    ldi   cnt2d, 00
    ldi   cnt3d, 00
    sei
    rcall sys
loop:                           ;在数码管显出十进制数
    mov   aa,   cnt1d    ;个位，十位，百位，千位,万位,十万位,
    ......
sys:    ;初始化,16 转 10 子程序,计算结果子程序,字形表等同 7.4.3 测脉冲宽度程序，省略。
    .........
```

7.4.7 测转速

```
; 测转速,信号从 AT90S8515 的 ICP 引脚输入,最大值为为 999999 转/分
;源程序:SLAVR747.ASM,在 SL-AVR 开发实验器验证通过
.include"8515def.inc"
    rjmp reset
```

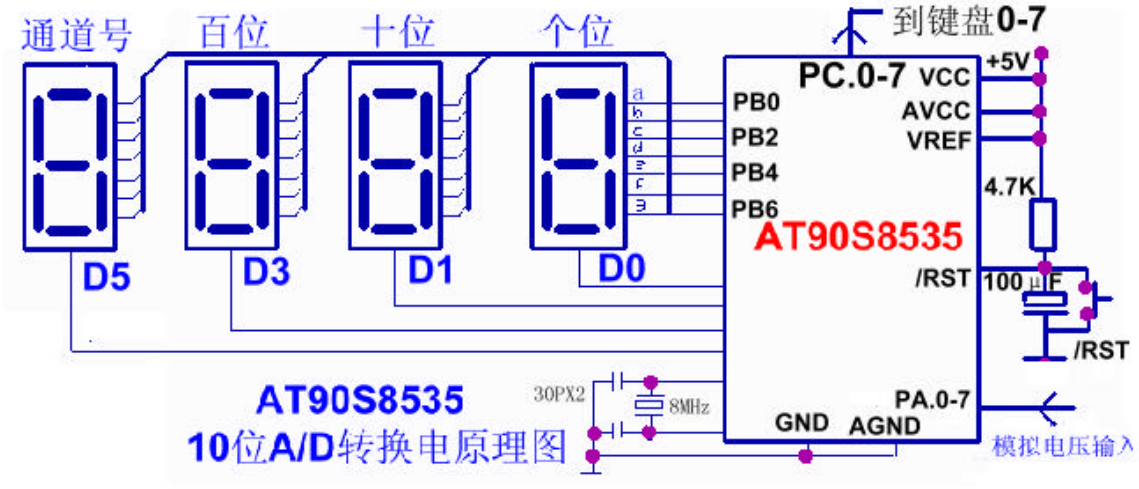
```
.def    temp  = r16          ;暂存器
.def    aa    = r17
.def    cnt   = r18
.def    mc16l = r19          ;mc16l 和 mc16h 存放脉冲个数
.def    mc16h = r20
.def    mp8u  = r21          ;mp8u=30,因为是每 2 秒采样
.def    res1  = r21          ;res1、res2 和 res3 存放结果的十六进制
.def    res2  = r22
.def    res3  = r23
.def    count = r24
.def    cnt1d = r25          ;cnt1、dcnt2d 和 cnt3d 存放结果的十进制
.def    cnt2d = r26
.def    cnt3d = r27
.def    dt    = r28
.def    dt1   = r29
.org 0x003                   ;icp 触发中断向量
icpt1:
    rjmp  captr
.org 0x008
    rjmp  interru
.cseg
.org 0x010
captr:                           ;icp 触发中断子程序
    brts  down
    ldi   temp,  0b00000101
    out   tccr0, temp            ;开 timer0
    ldi   temp,  0b00001010
    out   timsk, temp            ;致 timer0 中断和捕捉中断
    ldi   temp,  0b10000000
    out   tccr1b,temp
    set
    reti
down:                                ;下降沿开始计数
    set
    inc   mc16l
    brne  b
    inc   mc16h
    cpse  mc16h, dt1             ;溢出处理
    rjmp  b
    ldi   mc16h, 00
    ldi   mc16l, 00
    rjmp  over
b:
    ldi   temp,  0b00001010      ;致 timer0 中断和捕捉中断
```

```
    out   timsk, temp
    ldi   temp,  0b10000000
    out   tccr1b,temp
    reti
interru:                        ;timer0 溢出中断子程序
    dec   cnt
    breq  over
    ldi   temp,  0b00001010
    out   timsk, temp
    reti
over:
    rcall conver                ;conver:计算结果子程序
    rcall htd3                  ;htd3:
    rcall sys
    reti

reset:
    ldi   temp,  low(ramend)
    out   spl,   temp
    ldi   temp,  high(ramend)       ;设置堆栈
    out   spl+1, temp
    ldi   temp,  $ff                ;初始化数码管状态
    out   ddrb,  temp               ;B 口:数码管数据输出
    out   ddrd,  temp               ;D 口:pd0-pd5 为数码管片选
    ldi   temp,  $00
    out   portb, temp               ;共阴极,数码管全灭
    out   portd, temp
    ldi   cnt1d, 00
    ldi   cnt2d, 00
    ldi   cnt3d, 00
    sei
    rcall sys
loop:                           ;在数码管显出十进制数
    mov   aa,   cnt1d    ;个位，十位，百位，千位,万位,十万位,
    ......
sys:    ;初始化,16 转 10 子程序,计算结果子程序,字形表等同 7.4.3 测脉冲宽度程序，省略。
    .........
```

7.4.8 AT90S8535 的 A/D 转换

;源程序:SLAVR748.ASM,在 SL-AVR 开发实验器验证通过




```
;用 AT90S8535 作 0-7 通道 A/D 转换,用 LED 显示,左一位(D5)显示通道号,
;右三位(D2-D0)显示转换值(十六进制数 0-3FFH),程序下载即执行,
;自动从 0 通道到 7 通道 A/D 转换扫描显示,当你按下 0-7 任一位数字键,
;该通道显示时间延长一段时间,然后又自动循环显示。
;本程序在 SL-AVR 上调试通过。
;硬件接口: AT90S8535 的 PB.0-7 接 LED 段显示(用短路块短接),
;PD.0-5 接 LED 位显示,用接插线连接,
;PC0-PC7 接键盘线
;PA.0-7 接模拟电压,滑线电位器 A/D VX 端,
;AGND 接地
;最好 AVCC 与 VRBF 间接 1K 电阻,VRBF 到地接 100μF 电解电容,
;AVC 与 VCC 间接一只 100Ω电阻,AVCC 接 104 瓷片电容到地,
;/RST 接上复位按钮,插上 CZ2 到 AT4 下载线,即连通晶振引脚线,
.include  "8535def.inc"
 .org  $0000
     rjmp  reset
 .def   TEMP  =r16
 .def   TEMP1 =r17
 .def   temp2 =r18
 .def   temp3 =r19
 .def   CNT   =r20
 .def   scndp =r21
 .def   KSNI  =r22
 .def   SCNN  =r23
 .def   KEYN  =r24
 .def temp4=r25
 .equ   label =$0f00
 .org 0030
reset:  ldi temp,high(ramend);设置堆栈指针.
    out sph,temp
```

```
    ldi temp,low(ramend)
    out spl,temp
    clr xh          ;设置 x 指针为$0061.
    ldi xl,$61
    clr temp        ;清$0061,$0062 单元.
    st x+,temp
    st x,temp
init:   clr temp2       ;由 0 通道开始.
next:   ldi temp3,$01
next1:  clr temp4
again:  rcall cance ;调用 a/d 转换子程序 cance.
lp: rcall scan1     ;调用键扫显示子程序 scan1.
 scann: rcall scan1
    brtc recog      ;用按键转 recog.
    inc temp4       ;键扫显示次数 temp4 加 1.
    cpi temp4,$ff
    brne again      ;temp4 不等于$ff 转 again.
    dec temp3
    brne again      ;temp3 不等于 0 转 again.
    inc temp2       ;通道代码 temp2 加 1.
    cpi temp2,$08
    brne next       ;8 个通道未结束转下一通道 next.
    rjmp init       ;8 个通道已扫描完再重扫.
recog:  cpi keyn,$08
    brcc next       ;无效键转 next.
    ldi temp3,$04   ;设置有效通道键按下后的循环次数.
    mov temp2,keyn  ;通道数送 temp2.
    rjmp next1
cance:  mov temp,temp2  ;a/d 转换子程序.
    out admux,temp      ;设置通道.
    ldi temp,$86        ;设置 a/d 转换使能且采用 1/64 分频作转换工作频率.
    out adcsr,temp
    sbi adcsr,adsc      ;启动转换.
  loop: sbic adcsr,adsc ;转换结束跳行否则等待.
    rjmp loop
    in r2,adcl      ;把转换结果送 r2.r3.
    in r3,adch
    mov temp,temp2
    rcall wrdisram  ;调用把转换的结果转换成显示代码 wrdisram.
    ret             ;转换结束返回.
wrdisram:clr xh     ;使 x 指针为$0060.
    ldi xl,$60
    rcall fetch     ;调用 fetch.
    st x+,temp      ;把 temp 存入$0060 单元.
```

```
    inc xl
    inc xl
    mov temp,r3
    andi temp,$0f    ;取 r3 的低 4 位.
    rcall fetch      ;取字形代码.
    st x+,temp
    mov temp,r2
    swap temp
    andi temp,$0f    ;取 r2 的高 4 位.
    rcall fetch      ;取字形代码.
    st x+,temp
    mov temp,r2
    andi temp,$0f            ;取 r2 的低 4 位.
    rcall fetch              ;取字形代码.
    st x+,temp
    ret                      ;返回.
fetch:  ldi zh,high(label*2);设置字形表指针 z.
    mov zl,temp
    lpm                      ;取字形.
    mov temp,r0              ;字形码送 temp.
    ret                      ;返回
 SCAN1:         push xh      ;键扫显示子程序。
        PUSH XL               ;将 xl 压入堆栈
        PUSH TEMP3
        PUSH TEMP2
        PUSH TEMP1
        PUSH TEMP
        lDI XL,$60
            SET                      ;T 标志为 1 表示未按键
            LDI  SCNN,$00            ;按键起始扫描码 SCNN 为 00
            LDI  SCNDP,0B11011111    ;令 6 位七段 LED 扫描显示码初始为 11011111
            LDI  CNT,$06             ;七段 LED 共 6 位故 CNT=6 为位数计数
            LDI  KSNI,0B11110111     ;4*4 键盘扫描码 KSNI 初始为 11110111
      COL1: LDI  TEMP,$FF            ;PORTB 设定为输出
            OUT  DDRb,TEMP
            OUT  DDRC,TEMP           ;PORTC 设定为输出
        OUT  PORTC,TEMP
            OUT  DDRd,TEMP           ;PORTD 设定为输出
        OUT  PORTd,SCNDP            ;6 位七段 LED 扫描显示码输出到 PORTD
        CLR  XH
            LD   R1,X+               ;要显示于七段 LED 的间接寄存器 X 中的内容送入 R1 并
令 X 加 1
            OUT  PORTb,R1            ;显示内容输出到 PORTB 以驱动 LED 显示
            RCALL  DELAY             ;调用延时以显示此位数一段时间
```

```
            MOV   TEMP,CNT            ;LED 位数为 6 而按键码行数为 4 故需作 CNT 值检测
            SUBI   TEMP,$03           ;CNT=TEMP 与 3 相减比较
            BRCS   NOSK               ;位数扫描 CNT 超过 3 则 C 为 1 跳到 NOSK 不作按键处理
            LDI   TEMP1,$04           ;一共要检查 4 个按键
            LDI   TEMP,0B00001111     ;设定 PC0-PC3 为输出 PC4-PC7 为输入
            OUT   DDRc,TEMP
            OUT   PORTc,KSNI          ;KSNI 输出到 PORTC 并令 PC7-PC4 为上拉电阻输入态
            RCALL   DELYT             ;调用延时以稳定读取键盘 I/O 输入端
            IN    TEMP,PINc           ;读取 C 口检测 PC7-PC4 看是否有按键低电位输入
            ANDI   TEMP,0B11110000    ;取 TEMP 的高 4 位
            SWAP   TEMP               ;键码顺序为 PC4-PC7 故将 TEMP 的高低 4 位互换成 D0-D3
          KROW: SEC                   ;令 C 标志为 1 以便将键盘码 D0-D3 移到 C 标志位检测
        ROR   TEMP                    ;TEMP 的内容右移 1 位将第一个键码 D0=PC4 移到 C 标志
位检测
            BRCS   NOKEY              ;若有键按下则测到 PC4=D0=0，若 C=1 无按键则转到
NOKEY
            CLT                       ;若 PC4=D0=CF=0 表示有按键令 T=0 表示有按键
            MOV    KEYN,SCNN          ;把按键扫描码 SCNN 送键码 KEYN 中保存
        SBIS PINd,$07
        ADIW KEYN,$10                 ;判定 SHIFT 键是否按下，按下则键值加 10
         NOKEY: INC   SCNN            ;按键扫描码 SCNN 加 1
            DEC   TEMP1               ;扫描读取键数 TEMP1 减 1
            BRNE   KROW               ;每行有 4 个按键如 TEMP1 不为 0 则跳到 KROW 再检测
PC5-PC7
            SEC                       ;此行 4 个键码检测完后令 C 为 1 以方便键盘扫描码 KSNI
内容的移位
            ROR   KSNI                ;键盘扫描码 KSNI=CF=1>11110111 移位以进行下一行按
键扫描
          NOSK: SEC                   ;令进位标志 CF=1
            ROR   SCNDP               ;将扫描显示码 SCNDP 左移作下一位扫描
            DEC   CNT                 ;共需作 6 位数扫描显示故 CNT 减 1
            BRNE   COL1               ;CNT 减 1 不为 0 则跳回 COL1 再作扫描显示及读取键盘
输入
            LDI    TEMP,$FF           ;若已完成全部扫描显示和读取按键则令 TEMP=Off
            OUT    DDRC,TEMP          ; TEMP 输出到 DDRC 设定 PORTC 为输出驱动 LED
        OUT    PORTC,TEMP
        POP TEMP                  ;出栈
        POP TEMP1
        POP TEMP2
        POP TEMP3
        POP XL
        pop xh
        RET               ;子程序返回
      delay:push temp1    ;延时子程序
```

```
        push temp3
        ldi temp1,$10
       dt11:ldi temp3,$20
          dt21:nop
        dec temp3
        brne dt21
        dec temp1
        brne dt11
        pop temp3
        pop temp1
        ret
      delyt:ldi temp3,$20    ;延时子程序
       dt31:dec temp3
        brne dt31
        ret
.cseg
 .org $0f00              ;字形表
  .dw 0x063f,0x4f5b,0x6d66,0x077d
  .dw 0x6f7f,0x7c77,0x5e39,0x7179
```