# 附录 A MCS51 汇编语言程序清单

; 软件实验一 存储器块清零 S1.ASM

Block equ 6000h

0000 906000 mov dptr, #Block ; 起始地址0003 7800 mov r0, #0 ; 清 256 字节0005 E4 clr a

Loop:

0006 F0 movx @dptr, a

0007 A3 inc dptr ; 指向下一个地址

0008 D8FC djnz r0, Loop ; 记数减一

000A 02000A ljmp $ end

; 软件实验二 二进制到 BCD 码转换 S2.ASM

; 将 A 拆为三个 BCD 码, 并存入 Result 开始的叁个单元

Result equ 20h

org 0

0000 020015 ljmp Start

BinToBCD:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0003 | 75F064 |  | mov | b, #100 |  |
| 0006 | 84 |  | div | ab |
| 0007 | F520 |  | mov | Result, a | ; 除以 100, 得百位数 |
| 0009  000B  000E  000F | E5F0 75F00A  84  F521 |  | mov mov div mov | a, b b, #10  ab Result+1, a | ; 余数除以 10, 得十位数 |
| 0011 | 85F022 |  | mov | Result+2, b | ; 余数为个位数 |
| 0014 | 22 |  | ret |  |  |
| 0015 | 758140 | Start: | mov | sp, #40h |  |
| 0018 | 747B |  | mov | a, #123 |  |
| 001A  001C | 1103  02001C |  | call ljmp | BinToBCD  $ |  |
|  |  |  | end |  |  |

; 软件实验三 二进制到 ASCII 码转换 S3.ASM

; 将 A 拆为二个 ASCII 码, 并存入 Result 开始的二个单元

Result equ 20h

org 0

0000 020026 ljmp Start

ASCIITab:

0003 303132

|  |  |  |  |
| --- | --- | --- | --- |
| 0006 | 333435 |  | |
| 0009 | 363738 |
| 000C | 394142 |
| 000F | 434445 |
| 0012 | 46 | db | '0123456789ABCDEF' ; 定义数字对应的 ASCII 表 |
| 0013 | 900003 | BinToHex:  mov | dptr, #ASCIITab |
| 0016 | F5F0 | mov | b, a ; 暂存 A |
| 0018 | C4 | swap | a |
| 0019 | 540F | anl | a, #0fh ; 取高四位 |
| 001B | 93 | movc | a, @a+dptr ; 查 ASCII 表 |
| 001C | F520 | mov | Result, a |
| 001E | E5F0 | mov | a, b ; 恢复 A |
| 0020 | 540F | anl | a, #0fh ; 取低四位 |
| 0022 | 93 | movc | a, @a+dptr ; 查 ASCII 表 |
| 0023 | F521 | mov | Result+1, a |
| 0025 | 22 | ret |  |

Start:

0026 758140 mov sp, #40h

0029 741A mov a, #1ah

002B 1113 call BinToHex

002D 02002D ljmp $

end

; 软件实验四 内存块移动 S4.ASM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | ; 移动 | 3000H | --> 4000H, 256 字节 |
| 0000 | 7830 |  | mov | r0, #30h |
| 0002 | 7900 |  | mov | r1, #00h |
| 0004 | 7A40 |  | mov | r2, #40h |
| 0006 | 7B00 |  | mov | r3, #00h |
| 0008  000A | 7F00  8883 | Loop: | mov  mov | r7, #0  dph, r0 |
| 000C  000E  000F  0011 | 8982  E0 8A83  8B82 |  | mov movx mov  mov | dpl, r1 a, @dptr dph, r2  dpl, r3 |
| 0013 | F0 |  | movx | @dptr, a |
| 0014 | B9FF01 |  | cjne | r1, #0ffh, Goon1 |
| 0017  0018 | 08  09 | Goon1: | inc  inc | r0  r1 |
| 0019  001C  001D  001E | BBFF01 0A  0B DFEA | Goon2: | cjne inc  inc djnz | r3, #0ffh, Goon2 r2  r3  r7, Loop |
| 0020 | 020020 |  | ljmp | $ |
|  |  |  | end |  |

; 软件实验五 程序跳转表 S5.ASM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0000 | 02001D |  | org ljmp | 0  Start |
| 0003 | 7400 | Func0: | mov | a, #0 |
| 0005 | 22 |  | ret |  |
| 0006 | 7401 | Func1: | mov | a, #1 |
| 0008 | 22 |  | ret |  |
| 0009 | 7402 | Func2: | mov | a, #2 |
| 000B | 22 |  | ret |  |
| 000C | 7403 | Func3: | mov | a, #3 |
| 000E | 22 |  | ret |  |

FuncEnter:

000F 25E0 add a, acc ; ajmp 为二字节指令, 调用号 X 2 0011 900015 mov dptr, #FuncTab

0014 73 jmp @a+dptr FuncTab:

|  |  |  |  |
| --- | --- | --- | --- |
| 0015 | 0103 | ajmp | Func0 |
| 0017 | 0106 | ajmp | Func1 |
| 0019 | 0109 | ajmp | Func2 |
| 001B | 010C | ajmp | Func3 |

Start:

001D 7400 mov a, #0

001F 110F call FuncEnter

0021 7401 mov a, #1

0023 110F call FuncEnter

0025 7402 mov a, #2

0027 110F call FuncEnter

0029 7403 mov a, #3

002B 110F call FuncEnter

002D 02002D ljmp $

end

; 软件实验六 数据排序 S6.ASM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | Size Array Change | equ equ equ | 10  50h 0 | ; 数据个数  ; 数据起始地址  ; 交换标志 |
| 0000 | 7850 | Sort: | mov | r0, #Array |  |
| 0002 | 7F09 |  | mov | r7, #Size-1 |  |
| 0004  0006 | C200  E6 | Goon: | clr  mov | Change  a, @r0 |  |
| 0007 | FA |  | mov | r2, a |  |
| 0008 | 08 |  | inc | r0 |  |
| 0009  000B  000E | 86F0  B5F002 8008 |  | mov  cjne sjmp | B, @r0  a, B, NotEqual Next |  |
| 0010 | 4006 | NotEqual:  jc | | Next | ; 前小后大, 不交换 |
| 0012 | D200 | setb | | Change | ; 前大后小, 置交换标志 |
| 0014 | C6 | xch | | a, @r0 | ; 交换 |

0015 18 dec r0

0016 C6 xch a, @r0

0017 08 inc r0 Next:

0018 DFEC djnz r7, Goon

001A 2000E3 jb Change, Sort

001D 02001D ljmp $

end

; 硬件实验一 P1 口输入、输出实验 H1A.ASM Loop:

0000 7401 mov a, #01h

0002 7A08 mov r2, #8

Output:

0004 F590 mov P1, a

0006 23 rl a

0007 12000F call Delay

000A DAF8 djnz r2, Output

000C 020000 ljmp Loop

Delay:

000F 7E00 mov r6, #0

0011 7F00 mov r7, #0

DelayLoop:

0013 DEFE djnz r6, DelayLoop

0015 DFFC djnz r7, DelayLoop

0017 22 ret

end

; 硬件实验一 P1 口输入、输出实验 H1B.ASM

KeyLeft equ P1.0 KeyRight equ P1.1 LedLeft equ P1.2 LedRight equ P1.3

0000 D290 SETB KeyLeft

0002 D291 SETB KeyRight Loop:

0004 A290 MOV C,KeyLeft

0006 9292 MOV LedLeft,C

0008 A291 MOV C,KeyRight

000A 9293 MOV Ledright,C

000C 020004 LJMP Loop

END

; 硬件实验二 继电器控制实验 H2.ASM

Output equ P1.0 ; P1.0 port Loop:

0000 C3 clr c

0001 9290 mov Output,c

0003 12000F call Delay

0006 D3 setb c

0007 9290 mov Output,c

0009 12000F call Delay

000C 020000 ljmp Loop

Delay:

000F 7E00 mov r6, #0

0011 7F00 mov r7, #0

DLoop:

0013 DFFE djnz r7, DLoop

0015 DEFC djnz r6, DLoop

0017 22 ret

end

; 硬件实验三 用 74LS245 扩展 I/O 口实验 H3.ASM CS245 equ 8000h

org 0000h

Loop:

0000 908000 mov dptr,#CS245

0003 E0 movx a, @dptr

0004 020000 ljmp Loop

end

; 硬件实验四 用 74LS273 扩展 I/O 口实验 H4.ASM CS273 equ 8000h

0000 908000 mov dptr, #CS273

0003 7401 mov a, #1

Loop:

0005 F0 movx @dptr, a

0006 23 rl a

0007 00 nop

0008 020005 ljmp Loop

end

; 硬件实验五 PWM 转换电压实验 H5.ASM

|  |  |  |  |
| --- | --- | --- | --- |
|  | | ; 输出  ; 输出  ; 输出  OUTPUT | 50 (5:5) 占空比 PWM  10 (1:9) 占空比 PWM  90 (9:1) 占空比 PWM  equ P1.0 |
| 0000 | C290 | Loop: | clr OUTPUT |
| 0002 | 7403 |  | mov A, #3 |
| 0004 | 120011 |  | call Delay |
| 0007 | D290 |  | setb OUTPUT |
| 0009 | 7407 |  | mov a, #7 |
| 000B | 120011 |  | call Delay |
| 000E | 020000 |  | ljmp Loop |
| 0011 | 7800 | Delay: | mov r0, #0 |
| 0013 | D8FE | DLoop: | djnz r0, DLoop |
| 0015 | D5E0FB |  | djnz acc, DLoop |
| 0018 | 22 |  | ret |
|  |  |  | end |

; 硬件实验六 音频控制实验 H6.ASM

Speaker equ P1.0 CBit equ 0

Loop:

0000 C200 clr CBit

0002 850090 mov Speaker,CBit

0005 120013 call Delay

0008 D200 setb CBit

000A 850090 mov Speaker,CBit

000D 120013 call Delay

0010 020000 ljmp Loop

Delay:

0013 7832 mov r0, #50

0015 D8FE djnz r0, $

0017 22 ret

end

; 硬件实验七 8255 输入、输出实验 H7.ASM

mode equ 082h ; 方式 0，PA，PC 输出，PB 输入

PortA equ 0f000h ; Port A PortB equ 0f001h ; Port B PortC equ 0f002h ; Port C CAddr equ 0f003h ; 控制字地址

org 0h

0000 7482 mov a, #mode

0002 90F003 mov dptr, #CAddr

0005 F0 movx @dptr,a ; 输出控制字

0006 02001B jmp EX\_B

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ;实验 1：PortA 输出 | | | | | |
|  |  | EX\_A: |  |  |  |
| 0009 | 7480 |  | mov | a,#80H |  |
| 000B | 90F000 |  | mov | dptr,#PortA |  |
| 000E | 75F008 |  | mov | b,#8H |  |
| 0011 | F0 | OutA: | movx | @dptr,a | ; 输出 PortA |
| 0012 | 03 |  | rr | a | ; 移位 |
| 0013 | 7D01 |  | mov | r5,#1 |  |
| 0015 | 12002B |  | call | delay | ; 延时 |
| 0018 | D5F0F6 |  | djnz | b, OutA |  |
| ;实验 2： PortB 输入 PortA 输出 | | | | | |
|  |  | EX\_B: |  |  |  |
| 001B | 90F001 |  | mov | dptr,#PortB |  |
| 001E | E0 |  | movx | a,@dptr | ; 读入 PortB |
| 001F | 90F000 |  | mov | dptr,#PortA |  |
| 0022 | F0 |  | movx | @dptr,a | ; 输出到 PortA |
| 0023 | 7D02 |  | mov | r5,#2 |  |
| 0025 | 12002B |  | call | delay |  |
| 0028 | 020000 |  | ljmp | 0 |  |
|  |  | delay: |  |  |  |
| 002B | 7F00 |  | mov | r7,#0 |  |
| 002D | DFFE | ddd: | djnz | r7,ddd |  |

002F DEFC djnz r6,ddd

0031 DDFA djnz r5,ddd

0033 22 ret

end

; 硬件实验八 串行数转换并行数实验 H8A.ASM

|  |  |  |  |
| --- | --- | --- | --- |
| 0000 759800 | mov | SCON, #0 | ; 串口方式 0 |
| 0003 74AA | mov | a, #0aah |  |
| 0005 F599 | mov | SBUF, a | ; 输出 55h |
| 0007 3099FD | jnb | ti, $ |  |
| 000A C299 | clr | ti |  |
| 000C 02000C | ljmp | $ |  |
|  | end |  |  |

; 硬件实验八 串行数转换并行数实验 H8B.ASM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | LD | equ | P1.0 |
| DAT | equ | P1.1 |
| CLK | equ | P1.2 |
| 0000 | D292 |  | setb | CLK |
| 0002 | C290 |  | clr | LD |
| 0004 | D290 |  | setb | LD |
| 0006  0008 | 7F08  03 | Next: | mov  rr | r7, #8  a |
| 0009  000B  000D  000F  0011 | A291 92E7 C292 D292 DFF5 |  | mov mov clr setb djnz | c, DAT ACC.7, c CLK  CLK  r7, Next |
| 0013 | 020013 |  | ljmp end | $ |

; 硬件实验九 并行数转换串行数实验 H9A.ASM

|  |  |  |
| --- | --- | --- |
| LD equ | | P1.0 |
| 0000 C290 | clr | LD |
| 0002 D290 | setb | LD |
| 0004 759810 | mov | SCON, #00010000b |
| 0007 3098FD Wait: | jnb | RI, Wait |
| 000A E599 | mov | a, SBUF |
| 000C C298 | clr | ri |
| 000E 02000E | ljmp end | $ |

|  |  |  |
| --- | --- | --- |
| ; 硬件实验九 | 并行数转换串行数实验 | H9B.ASM |
|  | LD equ P1.0  DAT equ P1.1  CLK equ P1.2 |  |
| 0000 D292  0002 C290  0004 D290 | setb CLK  clr LD  setb LD |  |

0006 7F08 mov r7, #8

Next:

0008 03 rr a

0009 A291 mov c, DAT

000B 92E7 mov ACC.7, c

000D C292 clr CLK

000F D292 setb CLK

0011 DFF5 djnz r7, Next

0013 020013 ljmp $ end

|  |  |  |  |
| --- | --- | --- | --- |
| ; 硬件实验十 | 计数器实验 | H10.ASM |  |
| 0000 758905  0003 758C00  0006 758A00  0009 D28C | mov mov mov setb | TMOD, #00000101b TH0, #0  TL0, #0 TR0 | ; 方式 1,记数器  ; 开始记数 |
| 000B 858A90 | Loop:  mov | P1, TL0 | ; 将记数结果送 P1 口 |
| 000E 02000B | ljmp | Loop |  |
|  | end |  |  |

; 硬件实验十一 外部中断实验 H11.ASM LED equ P1.0

LEDBuf equ 0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0000 | 02000E | ljmp  org Interrupt0: | | Start  3 |  |
| 0003 | C0D0 | push | | PSW | ; 保护现场 |
| 0005 | B200 | cpl | | LEDBuf | ; 取反 LED |
| 0007 | A200 | mov | | c, LEDBuf |  |
| 0009 | 9290 | mov | | LED, c |  |
| 000B | D0D0 | pop | | PSW | ; 恢复现场 |
| 000D | 32 | reti | |  |  |
| 000E | C200 | Start:  clr | | LEDBuf |  |
| 0010 | C290 | clr | | LED |  |
| 0012 | 758801 | mov | | TCON, #01h | ; 外部中断 0 下降沿触发 |
| 0015 | 75A881 | mov | | IE, #81h | ; 打开外部中断允许位(EX0)及总中断允许位(EA) |
| 0018 | 020018 | ljmp | | $ |  |
|  |  | end | |  |  |
| ; 硬件实验十二 | | | 定时器实验 H12.ASM | |  |
|  | | | T100us equ 256-50 | | ; 100us 时间常数(6M) |
|  | | | C100us equ 30h | | ; 100us 记数单元 |
|  | | | LEDBuf equ 0 | |  |
|  | | | LED equ P1.0 | |  |
| 0000 020026 | | | ljmp Start | |  |

org 000bh

T0Int:

000B C0D0 push PSW

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 000D | E531 |  | mov | a, C100us+1 |
| 000F | 7002 |  | jnz | Goon |
| 0011  0013 | 1530  1531 | Goon: | dec  dec | C100us  C100us+1 |
| 0015 | E530 |  | mov | a, C100us |
| 0017 | 4531 |  | orl | a, C100us+1 |
| 0019 | 7008 |  | jnz | Exit ; 100us 记数器不为 0, 返回 |
| 001B | 753027 |  | mov | C100us, #high(Tick) |
| 001E  0021 | 753110  B200 | Exit: | mov cpl | C100us+1, #low(Tick)  LEDBuf ; 100us 记数器为 0, 重置记数器  ; 取反 LED |
| 0023 | D0D0 |  | pop | PSW |
| 0025 | 32 |  | reti |  |
| 0026 | 758902 | Start: | mov | TMOD, #02h ; 方式 2, 定时器 |
| 0029  002C | 758CCE  758ACE |  | mov mov | TH0, #t100us TL0, #t100us |
| 002F  0032 | 75A882 D28C |  | mov setb | IE, #10000010b ; EA=1, IT0 = 1  TR0 ; 开始定时 |
| 0034 | C200 |  | clr | LEDBuf |
| 0036 | C290 |  | clr | LED |
| 0038 | 753027 |  | mov | C100us, #high(Tick) |
| 003B | 753110 |  | mov | C100us+1, #low(Tick) |
| 003E | A200 | Loop: | mov | c, LEDBuf |
| 0040 | 9290 |  | mov | LED, c |
| 0042 | 02003E |  | ljmp | Loop |
|  |  |  | end |  |

; 硬件实验十三 D/A 转换实验 H13.ASM CS0832 equ 0a000h

|  |  |  |  |
| --- | --- | --- | --- |
| 0000 | 90A000 | mov | dptr, #CS0832 |
| 0003 | 7400 | mov | a, #0 |
| 0005 | F0 | movx | @dptr, a |
| 0006 | 7440 | mov | a, #40h |
| 0008 | F0 | movx | @dptr, a |
| 0009 | 7480 | mov | a, #80h |
| 000B | F0 | movx | @dptr, a |
| 000C | 74C0 | mov | a, #0c0h |
| 000E | F0 | movx | @dptr, a |
| 000F | 74FF | mov | a, #0ffh |
| 0011 | F0 | movx | @dptr, a |
| 0012 | 020012 | ljmp | $ |
|  |  | end |  |

; 硬件实验十四 A/D 转换实验 H14.ASM

mode equ 082h ; 方式 0，PA，PC 输出，PB 输入

CS0809 equ 8000h

PortA equ 9000h ; Port A PortB equ 9001h ; Port B

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | PortC CAddr | equ equ  org | 9002h ; Port C 9003h ; 控制字地址  0 |
| 0000 | 909003 |  | mov | dptr,#CAddr |
| 0003 | 7482 |  | mov | a,#mode |
| 0005 | F0 |  | movx | @dptr,a |
| 0006 | 908000 |  | mov | dptr, #CS0809 |
| 0009  000B | 7400  F0 |  | mov movx | a, #0  @dptr, a ; 起动 A/D |
| 000C  000E | 7440  D5E0FD |  | mov djnz | a, #40h  ACC, $ ; 延时 > 100us |
| 0011 | E0 |  | movx | a, @dptr ; 读入结果 |
| 0012  0014 | 7F64  909000 | dly: | mov  mov | r7,#100  dptr,#PortA |
| 0017 | F0 |  | movx | @dptr,a |
| 0018  001A | DFFA 020000 |  | djnz ljmp  end | r7,dly ;延时0 |

;硬件实验十五 外部中断实验（急救车与交通灯） H15.ASM

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | Flash STOP | equ 0  equ 1 | ; | LED 状态 |
| SR | equ p1.0 | ; | 南北红灯 |
| SY | equ p1.1 | ; | 南北黄灯 |
| SG | equ P1.2 | ; | 南北绿灯 |
| ER | equ P1.3 | ; | 东西红灯 |
| EY | equ P1.4 | ; | 东西黄灯 |
| EG | equ p1.5 | ; | 东西绿灯 |
| 0000 | 020006 |  | ljmp | Start |  |  |
|  |  |  | org | 3 |  |  |
| 0003 | D201 |  | setb | STOP | ; | 南北, 东西均红灯 |
| 0005 | 32 |  | reti |  |  |  |
| 0006 | 758146 | Start: | mov | SP, #70 |  |  |
| 0009 | 758801 |  | mov | TCON, #01h | ; | down edge IT0 |
| 000C | 75A881 |  | mov | IE, #81h | ; | enable EA, EX0 |
| 000F | D290 |  | setb | SR | ; | 南北, 东西均红灯 |
| 0011 | C291 |  | clr | SY |  |  |
| 0013 | C292 |  | clr | SG |  |  |
| 0015 | D293 |  | setb | ER |  |  |
| 0017 | C294 |  | clr | EY |  |  |
| 0019 | C295 |  | clr | EG |  |  |
| 001B  001D | C201  200168 | Loop: | clr  jb | STOP  STOP, AllRed |  |  |
| 0020 | D290 |  | setb | SR | ; | 南北红灯, 东西绿灯 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0022 | C291 |  | clr | SY |  |
| 0024 | C292 |  | clr | SG |
| 0026 | C293 |  | clr | ER |
| 0028 | C294 |  | clr | EY |
| 002A | D295 |  | setb | EG |
| 002C  002E | 7414  12009E |  | mov call | a, #20  Delay |
| 0031 | 200154 |  | jb | STOP, AllRed |
| 0034 | D290 |  | setb | SR | ; 南北红灯, 东西黄灯闪 |
| 0036 | C291 |  | clr | SY |  |
| 0038 | C292 |  | clr | SG |  |
| 003A | C293 |  | clr | ER |  |
| 003C | C294 |  | clr | EY |  |
| 003E | C295 |  | clr | EG |  |
| 0040 | C200 |  | clr | Flash |  |
| 0042  0044 | 7F09  A200 | Loop1: | mov  mov | r7, #9  c, Flash |  |
| 0046 | 9294 |  | mov | EY, c |  |
| 0048  004A  004D  004F | 7401  12009E B200 DFF3 |  | mov call cpl djnz | a, #1  Delay Flash  r7, Loop1 |  |
| 0051 | 200134 |  | jb | STOP, AllRed |  |
| 0054 | C290 |  | clr | SR | ; 南北绿灯, 东西红灯 |
| 0056 | C291 |  | clr | SY |  |
| 0058 | D292 |  | setb | SG |  |
| 005A | D293 |  | setb | ER |  |
| 005C | C294 |  | clr | EY |  |
| 005E | C295 |  | clr | EG |  |
| 0060 | 7414 |  | mov | a, #20 |  |
| 0062 | 12009E |  | call | Delay |  |
| 0065 | 200120 |  | jb | STOP, AllRed |  |
| 0068 | C290 |  | clr | SR | ; 东西红灯, 南北黄灯闪 |
| 006A | C291 |  | clr | SY |  |
| 006C | C292 |  | clr | SG |  |
| 006E | D293 |  | setb | ER |  |
| 0070 | C294 |  | clr | EY |  |
| 0072 | C295 |  | clr | EG |  |
| 0074 | C200 |  | clr | Flash |  |
| 0076  0078 | 7F09  A200 | Loop2: | mov  mov | r7, #9  c, Flash |  |
| 007A | 9291 |  | mov | SY, c |  |
| 007C  007E  0081 | 7401  12009E B200 |  | mov  call cpl | a, #1  Delay Flash |  |
| 0083 | DFF3 |  | djnz | r7, Loop2 |  |
| 0085 | 02001D |  | ljmp | Loop |  |

AllRed:

|  |  |  |  |
| --- | --- | --- | --- |
| 0088 | D290 | setb | SR |
| 008A | C291 | clr | SY |
| 008C | C292 | clr | SG |
| 008E | D293 | setb | ER |
| 0090 | C294 | clr | EY |
| 0092 | C295 | clr | EG |
| 0094 | C201 | clr | STOP |
| 0096 | 740A | mov | a, #10 |
| 0098 | 12009E | call | Delay |
| 009B | 02001D | ljmp | Loop |

Delay: ; 延时子程序

009E 7980 mov r1, #80H

00A0 7800 mov r0, #0

DelayLoop:

00A2 200107 jb STOP, ExitDelay 00A5 D8FB djnz r0, DelayLoop

00A7 D9F9 djnz r1, DelayLoop

00A9 D5E0F2 djnz ACC, Delay ExitDelay:

00AC 22 ret

end

; 硬件实验十六 八段数码管显示实验 H16.ASM

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | OUTBIT OUTSEG IN | equ equ equ | 08002h  08004h  08001h | ;  ;  ; | 位控制口 段控制口 键盘读入口 | |
| 0000 | 020041 | LEDBuf Num DelayT | equ equ equ  ljmp | 60h 70h 75h  Start |  | ; 显示缓冲  ; 显示的数据  ; | |
| LEDMAP: | | | ; | | | 八段管显示码 | |
| 0003 | 3F065B |  |  |  | |  | |
| 0006 | 4F666D |  |  |  | |  | |
| 0009  000B  000E  0011 | 7D07  7F6F77  7C395E  7971 |  | db  db | 3fh, 06h,  7fh, 6fh, | | 5bh, 4fh, 66h, 6dh, 7dh, 07h  77h, 7ch, 39h, 5eh, 79h, 71h | |
|  |  | Delay: |  | ; | | 延时子程序 | |
| 0013  0015 | 7F00  DFFE | mov DelayLoop:  djnz | | r7, #0  r7, DelayLoop | | | |
| 0017 | DEFC | djnz | | r6, DelayLoop | | | |
| 0019 | 22 | ret | |  | | | |
| DisplayLED: | | | | | | | |
| 001A | 7860 | mov | | r0, #LEDBuf | | |  |
| 001C | 7906 | mov | | r1, #6 | | | ; 共 6 个八段管 |
| 001E | 7A20 | mov | | r2, #00100000b | | | ; 从左边开始显示 |
| 0020 | 908002 | Loop:  mov | | dptr, #OUTBIT | | |  |
| 0023 | 7400 | mov | | a, #0 | | |  |
| 0025 | F0 | movx | | @dptr, a | | | ; 关所有八段管 |
| 0026 | E6 | mov | | a, @r0 | | |  |
| 0027 | 908004 | mov | | dptr, #OUTSEG | | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 002A  002B  002E  002F | F0 908002 EA  F0 |  | movx mov mov movx | @dptr,a  dptr, #OUTBIT a, r2  @dptr, a | ; 显示一位八段管 |
| 0030 | 7E01 |  | mov | r6, #01 |  |
| 0032 | 1113 |  | call | Delay |  |
| 0034 | EA |  | mov | a, r2 | ; 显示下一位 |
| 0035 | 03 |  | rr | a |  |
| 0036 | FA |  | mov | r2, a |  |
| 0037 | 08 |  | inc | r0 |  |
| 0038 | D9E6 |  | djnz | r1, Loop |  |
| 003A | 908002 |  | mov | dptr, #OUTBIT |  |
| 003D  003F  0040 | 7400  F0 22 |  | mov movx ret | a, #0  @dptr, a | ; 关所有八段管 |
| 0041 | 758140 | Start: | mov | sp, #40h |  |
| 0044  0047 | 757000  0570 | MLoop: | mov  inc | Num, #0  Num |  |
| 0049  004B  004D | E570 F5F0  7860 |  | mov mov  mov | a, Num b, a  r0, #LEDBuf |  |
| 004F | E5F0 | FillBuf:  mov | | a, b | |
| 0051 | 540F | anl | | a, #0fh | |
| 0053 | 900003 | mov | | dptr, #LEDMap | |
| 0056 | 93 | movc | | a, @a+dptr ; 数字转换成显示码 | |
| 0057 | F6 | mov | | @r0,a ; 显示在码填入显示缓冲 | |
| 0058 | 08 | inc | | r0 | |
| 0059 | 05F0 | inc | | b | |
| 005B | B866F1 | cjne | | r0, #LEDBuf+6, FillBuf | |
| 005E  0061 | 75751E  111A | mov DispAgain:  call | | DelayT,#30  DisplayLED ; 显示 | |
| 0063 | D575FB | djnz | | DelayT,DispAgain | |
| 0066 | 020047 | ljmp | | MLoop | |

end

; 硬件实验十七 键盘扫描显示实验 H17.ASM

|  |  |  |
| --- | --- | --- |
| OUTBIT | equ 08002h | ; 位控制口 |
| OUTSEG | equ 08004h | ; 段控制口 |
| IN | equ 08001h | ; 键盘读入口 |

LEDBuf equ 60h ; 显示缓冲

0000 0200A3 ljmp Start

LEDMAP: ; 八段管显示码

0003 3F065B

0006 4F666D

0009 7D07 db 3fh, 06h, 5bh, 4fh, 66h, 6dh, 7dh, 07h

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 000B  000E  0011 | 7F6F77  7C395E  7971 |  | db | 7fh, 6fh, 77h, 7ch, 39h, 5eh, 79h, 71h | |
|  |  | Delay: |  | ; 延时子程序 | |
| 0013 | 7F00 | mov | | r7, #0 | |
|  |  | DelayLoop: | |  | |
| 0015 | DFFE | djnz | | r7, DelayLoop | |
| 0017 | DEFC | djnz | | r6, DelayLoop | |
| 0019 | 22 | ret | |  | |
| DisplayLED: | | | | | |
| 001A | 7860 | mov | | r0, #LEDBuf |  |
| 001C | 7906 | mov | | r1, #6 | ; 共 6 个八段管 |
| 001E | 7A20 | mov | | r2, #00100000b | ; 从左边开始显示 |
|  |  | Loop: |  |  |  |
| 0020 | 908002 | mov | | dptr, #OUTBIT |  |
| 0023 | 7400 | mov | | a, #0 |  |
| 0025 | F0 | movx | | @dptr, a | ; 关所有八段管 |
| 0026 | E6 | mov | | a, @r0 |  |
| 0027 | 908004 | mov | | dptr, #OUTSEG |  |
| 002A | F0 | movx | | @dptr, a |  |
| 002B | 908002 | mov | | dptr, #OUTBIT |  |
| 002E | EA | mov | | a, r2 |  |
| 002F | F0 | movx | | @dptr, a | ; 显示一位八段管 |
| 0030 | 7E01 | mov | | r6, #1 |  |
| 0032 | 1113 | call | | Delay |  |
| 0034 | EA | mov | | a, r2 | ; 显示下一位 |
| 0035 | 03 | rr | | a |  |
| 0036 | FA | mov | | r2, a |  |
| 0037 | 08 | inc | | r0 |  |
| 0038 | D9E6 | djnz | | r1, Loop |  |
| 003A | 22 | ret | |  |  |
| TestKey: | | | | | |
| 003B | 908002 | mov | | dptr, #OUTBIT |  |
| 003E | 7400 | mov | | a, #0 |  |
| 0040 | F0 | movx | | @dptr, a | ; 输出线置为 0 |
| 0041 | 908001 | mov | | dptr, #IN |  |
| 0044 | E0 | movx | | a, @dptr | ; 读入键状态 |
| 0045 | F4 | cpl | | a |  |
| 0046 | 540F | anl | | a, #0fh | ; 高四位不用 |
| 0048 | 22 | ret | |  |  |
|  |  | KeyTable: | |  | ; 键码定义 |
| 0049 | 161514 |  | |  |  |
| 004C | FF | db | | 16h, 15h, 14h, | 0ffh |
| 004D | 131211 |  | |  |  |
| 0050 | 10 | db | | 13h, 12h, 11h, | 10h |
| 0051 | 0D0C0B |  | |  |  |
| 0054 | 0A | db | | 0dh, 0ch, 0bh, | 0ah |
| 0055 | 0E0306 |  | |  |  |
| 0058 | 09 | db | | 0eh, 03h, 06h, | 09h |
| 0059 | 0F0205 |  | |  |  |
| 005C | 08 | db | | 0fh, 02h, 05h, | 08h |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 005D | 000104 |  | | | |
| 0060 | 07 | db | 00h, 01h, 04h, 07h | | |
| GetKey: | | | | | |
| 0061 | 908002 | mov | dptr, #OUTBIT | | |
| 0064 | 8583A0 | mov | P2, dph | | |
| 0067 | 7801 | mov | r0, #Low(IN) | | |
| 0069 | 7920 | mov | r1, #00100000b | | |
| 006B | 7A06 | mov | r2, #6 | | |
| KLoop: | | | | | |
| 006D | E9 | mov | a, r1 | ; | 找出键所在列 |
| 006E | F4 | cpl | a |  |  |
| 006F | F0 | movx | @dptr, a |  |  |
| 0070 | F4 | cpl | a |  |  |
| 0071 | 03 | rr | a |  |  |
| 0072 | F9 | mov | r1, a | ; | 下一列 |
| 0073 | E2 | movx | a, @r0 |  |  |
| 0074 | F4 | cpl | a |  |  |
| 0075 | 540F | anl | a, #0fh |  |  |
| 0077 | 7006 | jnz | Goon1 | ; | 该列有键入 |
| 0079 | DAF2 | djnz | r2, KLoop |  |  |
| 007B | 7AFF | mov | r2, #0ffh | ; | 没有键按下, 返回 0ffh |
| 007D | 800F | sjmp | Exit |  |  |
| Goon1: | | | | | |
| 007F | F9 | mov | r1, a | ; | 键值 = 列 X 4 + 行 |
| 0080 | EA | mov | a, r2 |  |  |
| 0081 | 14 | dec | a |  |  |
| 0082 | 23 | rl | a |  |  |
| 0083 | 23 | rl | a |  |  |
| 0084 | FA | mov | r2, a | ; | r2 = (r2-1)\*4 |
| 0085 | E9 | mov | a, r1 | ; | r1 中为读入的行值 |
| 0086 | 7904 | mov | r1, #4 |  |  |
| LoopC: | | | | | |
| 0088 | 13 | rrc | a | ; | 移位找出所在行 |
| 0089 | 4003 | jc | Exit |  |  |
| 008B | 0A | inc | r2 | ; | r2 = r2+ 行值 |
| 008C | D9FA | djnz | r1, LoopC |  |  |

Exit:

008E EA mov a, r2 ; 取出键码

008F 900049 mov dptr, #KeyTable 0092 93 movc a, @a+dptr

0093 FA mov r2, a

WaitRelease:

0094 908002 mov dptr, #OUTBIT ; 等键释放

0097 E4 clr a

0098 F0 movx @dptr, a

0099 7E0A mov r6, #10

009B 1113 call Delay

009D 113B call TestKey

009F 70F3 jnz WaitRelease

00A1 EA mov a, r2

00A2 22 ret

Start:

00A3 758140 mov sp, #40h

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 00A6 | 7560FF |  | mov | LEDBuf+0, #0ffh | | ; 显示 8.8.8.8. |
| 00A9 | 7561FF |  | mov | LEDBuf+1, #0ffh | |  |
| 00AC | 7562FF |  | mov | LEDBuf+2, #0ffh | |  |
| 00AF | 7563FF |  | mov | LEDBuf+3, #0ffh | |  |
| 00B2 | 756400 |  | mov | LEDBuf+4, #0 | |  |
| 00B5 | 756500 |  | mov | LEDBuf+5, #0 | |  |
|  |  | MLoop: |  |  | |  |
| 00B8 | 111A |  | call | DisplayLED | | ; 显示 |
| 00BA | 113B |  | call | TestKey | | ; 有键入? |
| 00BC | 60FA |  | jz | MLoop | | ; 无键入, 继续显示 |
| 00BE | 1161 |  | call | GetKey | | ; 读入键码 |
| 00C0 | 540F |  | anl | a, #0fh | | ; 显示键码 |
| 00C2 | 900003 |  | mov | dptr, #LEDMap | |  |
| 00C5 | 93 |  | movc | a, @a+dptr | |  |
| 00C6 | F565 |  | mov | LEDBuf+5, a | |  |
| 00C8 | 0200B8 |  | ljmp  end | MLoop | |  |
| ; 硬件实验十八 | | | 电子时钟 | | H18.ASM | |

OUTBIT equ 08002h ; 位控制口OUTSEG equ 08004h ; 段控制口IN equ 08001h ; 键盘读入口

LEDBuf equ 60h ; 显 示 缓 冲

Hour equ 40h Minute equ 41h Second equ 42h C100us equ 43h

Tick equ 10000

T100us equ 256-50

0000 020083 ljmp Start

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 000B | C0D0 | T0Int: | org  push | 000bh  PSW |
| 000D | C0E0 |  | push | ACC |
| 000F | E544 |  | mov | a, C100us+1 |
| 0011 | 7002 |  | jnz | Goon |
| 0013  0015 | 1543  1544 | Goon: | dec  dec | C100us  C100us+1 |
| 0017 | E543 |  | mov | a, C100us |
| 0019 | 4544 |  | orl | a, C100us+1 |
| 001B | 7024 |  | jnz | Exit |
| 001D | 754327 |  | mov | C100us, #high(Tick) |
| 0020 | 754410 |  | mov | C100us+1, #low(Tick) |
| 0023 | 0542 |  | inc | Second |
| 0025 | E542 |  | mov | a, Second |
| 0027 | B43C17 |  | cjne | a, #60, Exit |
| 002A | 754200 |  | mov | Second, #0 |
| 002D  002F  0031 | 0541  E541 B43C0D |  | inc mov cjne | Minute a, Minute  a, #60, Exit |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0034 | 754100 |  | mov | Minute, #0 |
| 0037 | 0540 |  | inc | Hour |
| 0039  003B | E540 B41803 |  | mov cjne | a, Hour  a, #24, Exit |
| 003E  0041 | 754000  D0E0 | Exit: | mov  pop | Hour, #0  ACC |
| 0043 | D0D0 |  | pop | PSW |
| 0045 | 32 |  | reti |  |

Delay: ; 延时子程序

0046 7F00 mov r7, #0

DelayLoop:

0048 DFFE djnz r7, DelayLoop

004A DEFC djnz r6, DelayLoop

004C 22 ret

LEDMAP: ; 八段管显示码

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 004D  0050 | 3F065B  4F666D |  | | |
| 0053 | 7D07 | db | 3fh, 06h, 5bh, 4fh, 66h, 6dh, 7dh, 07h | |
| 0055 | 7F6F77 |  |  | |
| 0058  005B | 7C395E  7971 | db | 7fh, 6fh, 77h, 7ch, 39h, 5eh, 79h, 71h | |
| 005D | 7860 | DisplayLED:  mov | r0, #LEDBuf | |
| 005F | 7906 | mov | r1, #6 ; 共 6 个八段管 | |
| 0061 | 7A20 | mov  Loop: | r2, #00100000b ; 从左边开始显示 | |
| 0063 | 908002 | mov | dptr, #OUTBIT |  |
| 0066 | 7400 | mov | a, #0 |  |
| 0068 | F0 | movx | @dptr, a | ; 关所有八段管 |
| 0069 | E6 | mov | a, @r0 |  |
| 006A  006D  006E  0071 | 908004  F0 908002  EA | mov movx mov  mov | dptr, #OUTSEG @dptr,a  dptr, #OUTBIT  a, r2 |  |
| 0072 | F0 | movx | @dptr, a | ; 显示一位八段管 |
| 0073 | 7E01 | mov | r6, #1 |  |
| 0075 | 1146 | call | Delay |  |
| 0077 | EA | mov | a, r2 | ; 显示下一位 |
| 0078 | 03 | rr | a |  |
| 0079 | FA | mov | r2, a |  |
| 007A | 08 | inc | r0 |  |
| 007B | D9E6 | djnz | r1, Loop |  |
| 007D | 22 | ret |  |  |

; =============================================== ToLED:

007E 90004D mov dptr, #LEDMap 0081 93 movc a, @a+dptr

0082 22 ret

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0083 | 758902 | Start: | mov | TMOD, #02h | ; 模式 2, 定时器 |
| 0086 | 758CCE |  | mov | TH0, #T100us |  |
| 0089 | 758ACE |  | mov | TL0, #T100us |  |
| 008C | 75A882 |  | mov | IE, #10000010b | ; EA=1, IT0 = 1 |
| 008F | 754000 | mov | | Hour, #0 | |
| 0092 | 754100 | mov | | Minute, #0 | |
| 0095 | 754200 | mov | | Second, #0 | |
| 0098 | 754327 | mov | | C100us, #high(Tick) | |
| 009B | 754410 | mov | | C100us+1, #low(Tick) | |
| 009E | D28C |  | setb | TR0 ; 启动定时器 0 | |
| 00A0 | E540 | MLoop: | mov | a, Hour | |
| 00A2 | 75F00A |  | mov | b, #10 | |
| 00A5  00A6  00A8  00AA  00AC  00AE  00B0 | 84  117E F560 E5F0 117E  4480  F561 |  | div call mov mov call orl mov | ab ToLED  LEDBuf, a a, b ToLED  a, #80h LEDBuf+1, a | |
| 00B2  00B4  00B7  00B8  00BA  00BC  00BE  00C0  00C2 | E541 75F00A  84  117E F562 E5F0 117E  4480  F563 |  | mov mov div call mov mov call orl mov | a, Minute b, #10  ab ToLED  LEDBuf+2, a a, b  ToLED  a, #80h LEDBuf+3, a | |
| 00C4  00C6  00C9  00CA  00CC  00CE  00D0  00D2 | E542 75F00A  84  117E F564 E5F0 117E F565 |  | mov mov div call mov mov call mov | a, Second b, #10  ab ToLED  LEDBuf+4, a a, b  ToLED LEDBuf+5, a | |
| 00D4  00D6 | 115D  0200A0 |  | call ljmp | DisplayLED MLoop | |
|  |  |  | end |  | |

; 硬件实验十九 单片机串行口通讯实验 H19.ASM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OUTBIT OUTSEG  IN | | equ 08002h equ 08004h  equ 08001h | | ; 位控制口  ; 段控制口  ; 键盘读入口 |
| HasRcv equ | | | 20h.0 | ; 接收标志位 |
| LEDBuf equ | | | 40h | ; 显示缓冲 |
| RCVBuf equ | | | 50H | ; 接收缓冲 |
|  | ORG | | 0000H | |
| 0000 0200D4 | LJMP | | START | |

; 串行口中断程序

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | ORG | 0023H |  |
| 0023 | 309905 |  | JNB | TI,S0\_R |
| 0026 | C299 |  | CLR | TI |
| 0028 | 00 |  | NOP |  |
| 0029 | 8008 |  | SJMP | S0\_RET |
|  |  | S0\_R: |  |  | ; 接收数据 |
| 002B | C298 |  | CLR | RI |  |
| 002D | 859950 |  | MOV | RCVBUF,SBUF | ; 保存数据 |
| 0030 | D200 |  | SETB | HasRcv | ; 提示收到数据 |
| 0032 | 00 |  | NOP |  |  |
|  |  | S0\_RET: |  |  |  |
| 0033 | 32 |  | RETI |  |  |
| LEDMAP: ; 八段管显示码 | | | | | |
| 0034 | 3F065B |  |  |  | |
| 0037 | 4F666D |  |  |  | |
| 003A | 7D07 |  | db | 3fh, 06h, 5bh, 4fh, 66h, 6dh, 7dh, 07h | |
| 003C | 7F6F77 |  |  |  | |
| 003F | 7C395E |  |  |  | |
| 0042 | 7971 |  | db | 7fh, 6fh, 77h, 7ch, 39h, 5eh, 79h, 71h | |
|  |  | Delay: |  | ; 延时子程序 | |
| 0044 | 7F00 | mov | | r7, #0 | |
|  |  | DelayLoop: | |  | |
| 0046 | DFFE | djnz | | r7, DelayLoop | |
| 0048 | DEFC | djnz | | r6, DelayLoop | |
| 004A | 22 | ret | |  | |
| DisplayLED: | | | | | |
| 004B | 7840 | mov | | r0, #LEDBuf |  |
| 004D | 7906 | mov | | r1, #6 | ; 共 6 个八段管 |
| 004F | 7A20 | mov | | r2, #00100000b | ; 从左边开始显示 |
|  |  | Loop: |  |  |  |
| 0051 | 908002 | mov | | dptr, #OUTBIT |  |
| 0054 | 7400 | mov | | a, #0 |  |
| 0056 | F0 | movx | | @dptr, a | ; 关所有八段管 |
| 0057 | E6 | mov | | a, @r0 |  |
| 0058 | 908004 | mov | | dptr, #OUTSEG |  |
| 005B | F0 | movx | | @dptr,a |  |
| 005C | 908002 | mov | | dptr, #OUTBIT |  |
| 005F | EA | mov | | a, r2 |  |
| 0060 | F0 | movx | | @dptr, a | ; 显示一位八段管 |
| 0061 | 7E01 | mov | | r6, #1 |  |
| 0063 | 1144 | call | | Delay |  |
| 0065 | EA | mov | | a, r2 | ; 显示下一位 |
| 0066 | 03 | rr | | a |  |
| 0067 | FA | mov | | r2, a |  |
| 0068 | 08 | inc | | r0 |  |
| 0069 | D9E6 | djnz | | r1, Loop |  |
| 006B | 22 | ret | |  |  |
| TestKey: | | | | | |
| 006C | 908002 | mov | | dptr, #OUTBIT |  |
| 006F | 7400 | mov | | a, #0 |  |
| 0071 | F0 | movx | | @dptr, a | ; 输出线置为 0 |
| 0072 | 908001 | mov | | dptr, #IN |  |
| 0075 | E0 | movx | | a, @dptr | ; 读入键状态 |

0076 F4 cpl a

0077 540F anl a, #0fh ; 高四位不用

0079 22 ret

KeyTable: ; 键码定义

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 007A | 161514 |  | | | |
| 007D | FF | db | 16h, 15h, 14h, 0ffh | | |
| 007E | 131211 |  |  | | |
| 0081 | 10 | db | 13h, 12h, 11h, 10h | | |
| 0082 | 0D0C0B |  |  | | |
| 0085 | 0A | db | 0dh, 0ch, 0bh, 0ah | | |
| 0086 | 0E0306 |  |  | | |
| 0089 | 09 | db | 0eh, 03h, 06h, 09h | | |
| 008A | 0F0205 |  |  | | |
| 008D | 08 | db | 0fh, 02h, 05h, 08h | | |
| 008E | 000104 |  |  | | |
| 0091 | 07 | db | 00h, 01h, 04h, 07h | | |
| GetKey: | | | | | |
| 0092 | 908002 | mov | dptr, #OUTBIT | | |
| 0095 | 8583A0 | mov | P2, dph | | |
| 0098 | 7801 | mov | r0, #Low(IN) | | |
| 009A | 7920 | mov | r1, #00100000b | | |
| 009C | 7A06 | mov | r2, #6 | | |
| KLoop: | | | | | |
| 009E | E9 | mov | a, r1 | ; | 找出键所在列 |
| 009F | F4 | cpl | a |  |  |
| 00A0 | F0 | movx | @dptr, a |  |  |
| 00A1 | F4 | cpl | a |  |  |
| 00A2 | 03 | rr | a |  |  |
| 00A3 | F9 | mov | r1, a | ; | 下一列 |
| 00A4 | E2 | movx | a, @r0 |  |  |
| 00A5 | F4 | cpl | a |  |  |
| 00A6 | 540F | anl | a, #0fh |  |  |
| 00A8 | 7006 | jnz | Goon1 | ; | 该列有键入 |
| 00AA | DAF2 | djnz | r2, KLoop |  |  |
| 00AC | 7AFF | mov | r2, #0ffh | ; | 没有键按下, 返回 0ffh |
| 00AE | 800F | sjmp | Exit |  |  |
| Goon1: | | | | | |
| 00B0 | F9 | mov | r1, a | ; | 键值 = 列 X 4 + 行 |
| 00B1 | EA | mov | a, r2 |  |  |
| 00B2 | 14 | dec | a |  |  |
| 00B3 | 23 | rl | a |  |  |
| 00B4 | 23 | rl | a |  |  |
| 00B5 | FA | mov | r2, a | ; | r2 = (r2-1)\*4 |
| 00B6 | E9 | mov | a, r1 | ; | r1 中为读入的行值 |
| 00B7 | 7904 | mov | r1, #4 |  |  |
| LoopC: | | | | | |
| 00B9 | 13 | rrc | a | ; | 移位找出所在行 |
| 00BA | 4003 | jc | Exit |  |  |
| 00BC | 0A | inc | r2 | ; | r2 = r2+ 行值 |
| 00BD | D9FA | djnz | r1, LoopC |  |  |
| Exit: | | | | | |

00BF EA mov a, r2 ; 取出键码

00C0 90007A mov dptr, #KeyTable 00C3 93 movc a, @a+dptr

00C4 FA mov r2, a

WaitRelease:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 00C5 | 908002 |  | mov | dptr, #OUTBIT ; 等键释放 | | |
| 00C8 | E4 |  | clr | a | | |
| 00C9 | F0 |  | movx | @dptr, a | | |
| 00CA | 7E0A |  | mov | r6, #10 | | |
| 00CC | 1144 |  | call | Delay | | |
| 00CE | 116C |  | call | TestKey | | |
| 00D0 | 70F3 |  | jnz | WaitRelease | | |
| 00D2 | EA |  | mov | a, r2 | | |
| 00D3 | 22 |  | ret |  | | |
|  |  | START: |  |  | | |
| 00D4 | 758160 |  | MOV | SP, #60H | | |
| 00D7 | 75A800 |  | MOV | IE, #0 ; DISABLE ALL INTERRUPT | | |
| 00DA | 758920 |  | MOV | TMOD,#020H ; 定时器 1 工作于方式 2 (8 位重装) | | |
| 00DD | 758DF3 |  | MOV | TH1, #0F3H ; 波特率?2400BPS @ 12MHz | | |
| 00E0 | 758BF3 |  | MOV | TL1, #0F3H | | |
| 00E3 | 53877F |  | ANL | PCON,#07FH ; SMOD 位清零 | | |
| 00E6 | 438780 |  | orl | PCON,#80h | | |
| 00E9 | 759850 |  | MOV | SCON,#050H ; 串行口工作方式设置 | | |
| 00EC | 7540FF |  | MOV | LEDBuf, #0ffh ; 显示 8.8.8.8. | | |
| 00EF | 7541FF |  | mov | LEDBuf+1, #0ffh | | |
| 00F2 | 7542FF |  | mov | LEDBuf+2, #0ffh | | |
| 00F5 | 7543FF |  | mov | LEDBuf+3, #0ffh | | |
| 00F8 | 754400 |  | mov | LEDBuf+4, #0 | | |
| 00FB | 754500 |  | mov | LEDBuf+5, #0 | | |
| 00FE | D28E |  | SETB | TR1 | | |
| 0100 | D2AC |  | SETB | ES | | |
| 0102 | D2AF |  | SETB | EA | | |
|  |  |  | ;mov | sbuf,a | | |
|  |  |  | ;jnb | ti,$ | | |
|  |  | MLoop: |  |  | | |
| 0104 | 20000F |  | jb | HasRcv, RcvData ; 收到数据？ | | |
| 0107 | 114B |  | call | DisplayLED ; 显示 | | |
| 0109 | 116C |  | call | TestKey ; 有键入? | | |
| 010B | 60F7 |  | jz | MLoop ; 无键入, 继续显示 | | |
| 010D | 1192 |  | call | GetKey ; 读入键码 | | |
| 010F | 540F |  | anl | a, #0fh ; 通讯口输出键码 | | |
| 0111 | F599 |  | MOV | SBUF,A | | |
| 0113 | 020104 |  | LJMP | MLoop | | |
| RcvData: | | | | | | |
| 0116 | C200 | clr | | HasRcv | ; | 是 |
| 0118 | E550 | mov | | a, RcvBuf | ; | 显示数据 |
| 011A | F5F0 | mov | | b,a |  |  |
| 011C | 540F | anl | | a,#0fh | ; | 显示低位 |
| 011E | 900034 | mov | | dptr, #LEDMap |  |  |
| 0121 | 93 | movc | | a, @a+dptr |  |  |
| 0122 | F545 | mov | | LEDBuf+5, a |  |  |
| 0124 | E5F0 | mov | | a,b |  |  |
| 0126 | C4 | swap | | a | ; | 显示高位 |
| 0127 | 540F | anl | | a,#0fh |  |  |
| 0129 | 900034 | mov | | dptr, #LEDMap |  |  |
| 012C | 93 | movc | | a, @a+dptr |  |  |
| 012D | F544 | mov | | LEDBuf+4, a |  |  |
| 012F | 020104 | ljmp | | MLoop |  |  |
|  |  | END | |  |  |  |

; 硬件实验二十 打印机控制试验 H20.ASM

; printer

; ASM for MCS51

mode equ 082h

dport equ 08000h ; PA 口，数据口

status equ 08001h ; PB.0 忙状态

ctl equ 08002h ; PC.0 选通控制

contrl equ 08003h ; 8255 控制口

stb equ 0 ; 选通位

busy equ 0 ; 忙状态位

org 0h

0000 02002E jmp start delay:

0003 7E00 mov r6,#0

0005 7F00 mov r7,#0

0007 00 dd: nop

0008 DFFD djnz r7,dd

000A DEFB djnz r6,dd

000C 22 ret

print:

000D C083 push dph

000F C082 push dpl

0011 F5F0 mov b,a

0013 908001 mov dptr, #status rd\_status:

0016 E0 movx a,@dptr ; 读打印机状态

0017 20E0FC jb acc.busy, rd\_status ; 若忙，等待

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 001A | 908000 | mov | dptr,#dport | ; | 数据送出 |
| 001D | E5F0 | mov | a,b |  |  |
| 001F | F0 | movx | @dptr,a |  |  |
| 0020 | 908002 | mov | dptr,#ctl | ; | 输出选通脉冲 |
| 0023 | 7401 | mov | a,#1 |  |  |
| 0025 | F0 | movx | @dptr,a |  |  |
| 0026 | 7400 | mov | a,#0 |  |  |
| 0028 | F0 | movx | @dptr,a |  |  |
| 0029 | D082 | pop | dpl |  |  |
| 002B | D083 | pop | dph |  |  |
| 002D | 22 | ret |  |  |  |
| start: | | | | | |
| 002E | 908003 | mov | dptr, #contrl |  | |
| 0031 | 7482 | mov | a, #mode |  | |
| 0033 | F0 | movx | @dptr,a |  | |
| 0034 | 900043 | mov | dptr, #string | ; 被打印字串 | |
| nextchar: | | | | | |
| 0037 | E4 | clr | a |  | |
| 0038 | 93 | movc | a,@a+dptr |  | |
| 0039 | 7002 | jnz | prtchar | ; 字串结束 | |
| 003B | 80FE | jmp | $ |  | |
| prtchar: | | | | | |
| 003D | 110D | call | print |  | |
| 003F | A3 | inc | dptr | ; 下一字符 | |
| 0040 | 80F5 | jmp | nextchar |  | |

|  |  |  |
| --- | --- | --- |
| 0042 | 00 | nop  string: |
| 0043 | 48656C |  |
| 0046 | 6C6F20 |  |
| 0049 | 576F72 |  |
| 004C | 6C6421 |  |
| 004F | 0D0A | db 'Hello World!',0dh,0ah |
| 0051 | 492063 |  |
| 0054 | 616E20 |  |
| 0057 | 707269 |  |
| 005A | 6E7420 |  |
| 005D | 6E6F77 |  |
| 0060 | 0D0A0C |  |
| 0063 | 0000 | db 'I can print now',0dh,0ah,0ch,0,0 |
|  |  | end |

; 硬件实验二十一 直流电机控制实验 H21.ASM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | mode STATUS PORTA CTL | equ equ equ equ | 082h  08001h  08000h  08003h |  |
| CS0832 | equ | 09000h |
| DC\_P | equ | 0 |
| count0 | equ | 40 |
| count1 | equ | 41 |
| 0000 | 020039 |  | org jmp | 0h start |
| 0003 | 7D0A | delay: | mov | r5,#10 |
| 0005 | 7E00 |  | mov | r6,#0 |
| 0007  0009 | 7F00  DFFE | ddd: | mov  djnz | r7,#0  r7,ddd |
| 000B | DEFC |  | djnz | r6,ddd |
| 000D  000F  0010 | DDFA 22  758901 | read: | djnz ret  mov | r5,ddd  TMOD, #01 | ; 16 位计时 |
| 0013 | C28C |  | clr | TR0 |  |
| 0015 | 758C00 |  | mov | TH0, #0 |  |
| 0018  001B  001E  001F  0022 | 758A00  908001  E0  30E0FC E0 | r\_0: r\_1: | mov mov movx jnb  movx | TL0, #0  dptr, #STATUS a,@dptr acc.DC\_P, r\_0  a,@dptr | ; 等待低电平完 |
| 0023 | 20E0FC |  | jb | acc.DC\_P, r\_1 | ; 等待高电平完 |
| 0026 | D28C |  | setb | TR0 |  |
| 0028 | E0 | r\_2: | movx | a,@dptr |  |
| 0029  002C  002D  0030 | 30E0FC E0 20E0FC  C28C | r\_3: | jnb movx jb  clr | acc.DC\_P, r\_2 a,@dptr acc.DC\_P, r\_3  TR0 | ; 等待低电平完  ; 等待高电平完 |
| 0032 | 858C28 |  | mov | count0,TH0 |  |
| 0035 | 858A29 |  | mov | count1,TL0 |  |
| 0038 | 22 |  | ret |  |  |
|  |  | start: |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0039  003C  003E  003F | 909000  74FF  F0 1103 | mov mov movx  call | dptr, #CS0832 a, #0ffh @dptr, a  delay | ; | 等待电机运转稳定 |
| 0041 | 1110 | call | read | ; | 读取时间 |
| 0043 | 909000 | mov | dptr, #CS0832 | ; | 设断点，观察上次时间 |
| 0046 | 74C0 | mov | a, #0c0h |  |  |
| 0048 | F0 | movx | @dptr, a |  |  |
| 0049 | 1103 | call | delay | ; | 等待电机运转稳定 |
| 004B | 1110 | call | read |  |  |
| 004D | 909000 | mov | dptr, #CS0832 | ; | 设断点，观察上次时间 |
| 0050 | 7440 | mov | a, #040h | ; | 电机反转 |
| 0052 | F0 | movx | @dptr, a |  |  |
| 0053 | 1103 | call | delay | ; | 等待电机运转稳定 |
| 0055 | 1110 | call | read | ; | 读取时间 |
| 0057 | 909000 | mov | dptr, #CS0832 | ; | 设断点，观察上次时间 |
| 005A  005C  005D | 7400  F0 1103 | mov  movx call | a, #00h  @dptr, a delay | ;  ; | 电机反转加速  等待电机运转稳定 |
| 005F | 1110 | call | read |  |  |
| 0061 | 020061 | ljmp | $ | ; | 设断点，观察上次时间 |
|  |  | end |  |  |  |

; 硬件实验二十二 步进电机控制实验 H22.ASM

mode equ 082h contrl equ 08003h ctl equ 08000h Astep equ 01h Bstep equ 02h Cstep equ 04h Dstep equ 08h

dly\_c equ 60h

org 0h

step:

0000 7482 mov a,#mode

0002 908003 mov dptr,#contrl 0005 F0 movx @dptr,a

0006 908002 mov dptr,#8002h

0009 7400 mov a,#0

000B F0 movx @dptr,a

000C 756010 mov dly\_c,#10h

000F 020051 jmp loop1

; 单/双八拍工作方式

loop:

0012 908000 mov dptr, #ctl

0015 7401 mov a,#Astep

0017 F0 movx @dptr,a

0018 12009D call delay

001B 7403 mov a,#Astep+Bstep

001D F0 movx @dptr,a

001E 12009D call delay

0021 7402 mov a,#Bstep

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0023 | F0 |  | movx | @dptr,a |  |
| 0024 | 12009D |  | call | delay |
| 0027 | 7406 |  | mov | a,#Bstep+Cstep |
| 0029 | F0 |  | movx | @dptr,a |
| 002A | 12009D |  | call | delay |
| 002D | 7404 |  | mov | a,#Cstep |
| 002F | F0 |  | movx | @dptr,a |
| 0030 | 12009D |  | call | delay |
| 0033 | 740C |  | mov | a,#Cstep+Dstep |
| 0035 | F0 |  | movx | @dptr,a |
| 0036 | 12009D |  | call | delay |
| 0039 | 7408 |  | mov | a,#Dstep |
| 003B | F0 |  | movx | @dptr,a |
| 003C | 12009D |  | call | delay |
| 003F | 7409 |  | mov | a,#Dstep+Astep |
| 0041 | F0 |  | movx | @dptr,a |
| 0042 | 12009D |  | call | delay |
| 0045 | E560 |  | mov | a, dly\_c |
| 0047 | 14 |  | dec | a | ; 提高转速 |
| 0048 | B40101 |  | cjne | a,#1, nn1 | ; 最快速度 |
| 004B | 04 |  | inc | a |  |
| 004C | F560 | nn1: | mov | dly\_c,a |  |
| 004E | 020012 |  | ljmp | Loop |  |

; 双四拍工作方式

loop1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0051 | 908000 |  | mov | dptr,#ctl |
| 0054 | 7403 |  | mov | a, #Astep+Bstep |
| 0056 | F0 |  | movx | @dptr,a |
| 0057 | 12009D |  | call | delay |
| 005A  005C  005D  0060 | 7406  F0 12009D  740C |  | mov movx call  mov | a, #Bstep+Cstep @dptr,a  delay  a, #Cstep+Dstep |
| 0062 | F0 |  | movx | @dptr,a |
| 0063 | 12009D |  | call | delay |
| 0066 | 7409 |  | mov | a, #Dstep+Astep |
| 0068 | F0 |  | movx | @dptr,a |
| 0069 | 12009D |  | call | delay |
| 006C | E560 |  | mov | a, dly\_c |
| 006E  006F  0072 | 14  B40201 04 |  | dec cjne inc | a  a,#2, nn2 a |
| 0073 | F560 | nn2: | mov | dly\_c,a |
| 0075 | 80DA |  | jmp | loop1 |

; 单四拍工作方式

loop2:

0077 908000 mov dptr,#ctl

007A 7408 mov a,#Dstep

007C F0 movx @dptr,a

007D 12009D call delay

0080 7404 mov a,#Cstep

0082 F0 movx @dptr,a

0083 12009D call delay

0086 7402 mov a,#Bstep

0088 F0 movx @dptr,a

0089 12009D call delay

008C 7401 mov a,#Astep

008E F0 movx @dptr,a

008F 12009D call delay

0092 E560 mov a, dly\_c

0094 14 dec a

0095 B40301 cjne a,#3, nn3

0098 04 inc a

0099 F560 nn3: mov dly\_c,a 009B 80DA jmp loop2

delay:

009D AE60 mov r6,dly\_c dd1:

009F 7F00 mov r7,#0

00A1 DFFE djnz r7,$

00A3 DEFA djnz r6,dd1

00A5 22 ret

end

; 硬件实验二十三 温度传感器试验 H23.ASM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LowTemp  HighTemp | | | equ -99  equ 99 | ; A/D 0  ; A/D 255 |
|  |  | ADPort equ 09000h CurTemp equ 51h | | |
|  |  | OUTBIT equ 08002h  OUTSEG equ 08004h ; 段控制口  IN equ 08001h | | |
|  |  | LEDBuf equ 60h | | |
| 0000 | 0200A0 | ORG 0000H  ljmp Start | | |
| 0003 | 3F065B | LEDMAP: | | |
| 0006 | 4F666D |  | | |
| 0009  000B  000E  0011 | 7D07  7F6F77  7C395E  7971 | db 3fh, 06h, 5bh, 4fh, 66h, 6dh, 7dh, 07h  db 7fh, 6fh, 77h, 7ch, 39h, 5eh, 79h, 71h | | |

Delay:

0013 7F00 mov r7, #0

DelayLoop:

0015 DFFE djnz r7, DelayLoop

0017 DEFC djnz r6, DelayLoop

0019 22 ret

DisplayLED:

001A 7860 mov r0, #LEDBuf

001C 7906 mov r1, #6

001E 7A20 mov r2, #00100000b

Loop:

0020 908002 mov dptr, #OUTBIT

0023 7400 mov a, #0

0025 F0 movx @dptr, a

0026 E6 mov a, @r0

0027 908004 mov dptr, #OUTSEG

002A F0 movx @dptr, a

002B 908002 mov dptr, #OUTBIT

002E EA mov a, r2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 002F | F0 | movx | | @dptr, a | |
| 0030 | 7E01 | mov | | r6, #1 | |
| 0032 | 1113 | call | | Delay | |
| 0034 | EA | mov | | a, r2 | |
| 0035 | 03 | rr | | a | |
| 0036 | FA | mov | | r2, a | |
| 0037 | 08 | inc | | r0 | |
| 0038 | D9E6 | djnz | | r1, Loop | |
| 003A | 908002 | mov | | dptr, #OUTBIT | |
| 003D | 7400 | mov | | a, #0 | |
| 003F | F0 | movx | | @dptr, a | |
| 0040 | 22 | ret | |  | |
| ; ================================  DisplayResult: | | | | | |
| 0041 | E551 |  | mov | a, CurTemp |  |
| 0043 | 30E708 |  | jnb | acc.7, GE0 |  |
| 0046 | 756040 |  | mov | LEDBuf, #40h | ; '-' |
| 0049 | 14 |  | dec | a |  |
| 004A | F4 |  | cpl | a |  |
| 004B | 020051 |  | jmp | Goon |  |
|  |  | GE0: |  |  |  |
| 004E | 756000 |  | mov | LEDBuf, #0 ; ' | ' |
|  |  | Goon: |  |  |  |
| 0051 | 75F00A |  | mov | b, #10 |  |
| 0054 | 84 |  | div | ab |  |
| 0055 | 900003 |  | mov | dptr, #LEDMAP |  |
| 0058 | 93 |  | movc | a, @a+dptr |  |
| 0059 | F561 |  | mov | LEDBuf+1, a |  |
| 005B | E5F0 |  | mov | a, b |  |
| 005D | 93 |  | movc | a, @a+dptr |  |
| 005E | F562 |  | mov | LEDBuf+2, a |  |
| 0060 | 756300 |  | mov | LEDBuf+3, #0 | ; ' ' |
| 0063 | 756400 |  | mov | LEDBuf+4, #0 | ; ' ' |
| 0066 | 756500 |  | mov | LEDBuf+5, #0 | ; ' ' |
| 0069 | 22 |  | ret |  |  |
| ReadAD: | | | | | |
| 006A | 909000 | mov | | dptr, #ADPort |  |
| 006D | E4 | clr | | a |  |
| 006E | F0 | movx | | @dptr, a | ; start A/D |
| 006F | 7450 | mov | | a, #80 |  |
| 0071 | D5E0FD | djnz | | acc, $ | ; delay |
| 0074 | E0 | movx | | a, @dptr |  |
| 0075 | F5F0 | mov | | b,a |  |
| 0077 | 74FF | mov | | a,#0ffh |  |
| 0079 | C3 | clr | | c |  |
| 007A | 95F0 | subb | | a,b |  |
| 007C | 22 | ret | |  |  |
| ReadTemp: | | | | | |
| 007D | 7900 | mov | | r1, #0 | |
| 007F | 7A00 | mov | | r2, #0 | |
| 0081 | 7810 | mov | | r0, #16 | |

r1r2/16

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0083 | 116A | RLoop: | call | ReadAD |
| 0085 | 2A |  | add | a, r2 |
| 0086 | FA |  | mov | r2, a |
| 0087 | 5001 |  | jnc | GN1 |
| 0089  008A | 09  D8F7 | GN1: | inc  djnz | r1  r0, RLoop |
| 008C  008D  008E  0090 | EA C4 540F  C9 |  | mov swap anl  xch | a, r2 a  a, #0fh  a, r1 |
| 0091 | C4 |  | swap | a |
| 0092 | 54F0 |  | anl | a, #0f0h |
| 0094 | 49 |  | orl | a, r1 ; a = |
| 0095 | 75F0C6 | mov | | b, #(HighTemp-LowTemp) |
| 0098 | A4 | mul | | ab |
| 0099  009B  009D  009F | E5F0 249D F551 22 |  | mov add mov ret | a, b ; /256  a, #LowTemp CurTemp, a |
| 00A0 | 758170 | Start: | mov | sp, #70h |
|  |  | MLoop: |  |  |
| 00A3  00A5  00A7  00A9 | 1141  111A  117D  80F8 |  | call call call sjmp | DisplayResult DisplayLED ReadTemp MLoop |
|  |  |  | end |  |

; 硬件实验二十四 液晶显示控制电路 H24.ASM

CWADD1 EQU 08000H ;写指令代码地址（E1） DWADD1 EQU 08001H ;写显示数据地址（E1） CRADD1 EQU 08002H ;读状态字地址（E1） DRADD1 EQU 08003H ;读显示数据地址（E1）

CWADD2 EQU 08004H ;写指令代码地址（E2） DWADD2 EQU 08005H ;写显示数进地址（E2） CRADD2 EQU 08006H ;读状态字地址（E2） DRADD2 EQU 08007H ;读显示数据地址（E2）

PD1 EQU 3DH ;122/2 分成左右两半屏 122x32 COLUMN EQU 30H

PAGE\_ EQU 31H ;页地址寄存器 D1,DO:页地址CODE\_ EQU 32H ;字符代码寄存器

COUNT EQU 33H ;计数器

DIR equ 34h

dtp1 equ 35h

dtp2 equ 36h

dtp3 equ 37h

CTEMP EQU 38H

COM EQU 20H ;指令寄存器DAT EQU 21H ;数据寄存器

ORG 0

0000 020202 jmp main

0003 90F002 mov dptr, #0f002h

0006 7456 mov a,#56h

0008 F0 movx @dptr,a

0009 7400 mov a,#0

000B E0 movx a,@dptr

000C 020202 JMP MAIN

;

; 初始化程序

000F 7520E2 INIT: MOV COM, #0E2H ;复位

0012 12006E LCALL PRO

0015 1200B0 LCALL PR3

0018 7520A4 MOV COM, #0A4H ;关闭休闭状态

001B 12006E LCALL PRO

001E 1200B0 LCALL PR3

0021 7520A9 MOV COM, #0A9H ;设置 1／32 占空比

0024 12006E LCALL PRO

0027 1200B0 LCALL PR3

002A 7520A0 MOV COM, #0A0H ;正向排序设置

002D 12006E LCALL PRO

0030 1200B0 LCALL PR3

0033 7520C0 MOV COM, #0C0H ;设置显示起始行为第一行

0036 12006E LCALL PRO

0039 1200B0 LCALL PR3

003C 7520AF MOV COM, #0AFH ;开显示设置

003F 12006E LCALL PRO

0042 1200B0 LCALL PR3

0045 22 RET

;

; 清屏

0046 7C00 CLEAR: MOV R4,#00H ;页面地址暂存器设置

0048 EC CLEAR1: MOV A,R4 ;取页地址值

0049 44B8 ORL A,#0B8H ;"或"页面地址设置代码

004B F520 MOV COM,A ;页面地址设置

004D 12006E LCALL PRO

0050 1200B0 LCALL PR3

0053 752000 MOV COM,#00H ;列地址设置为"0"

0056 12006E LCALL PRO

0059 1200B0 LCALL PR3

005C 7B50 MOV R3,#50H ;一页清 80 个字节

005E 752100 CLEAR2: MOV DAT,#00H ;显示数据为"0"

0061 120084 LCALL PR1

0064 1200C6 LCALL PR4

0067 DBF5 DJNZ R3,CLEAR2 ;页内字节清零循环

0069 0C INC R4 ;页地址暂存器加一

006A BC04DB CJNE R4,#04H,CLEAR1;RAM 区清零循环

006D 22 RET

;

;1．写指令代码子程序（E1） 006E C082 PRO: PUSH DPL

0070 C083 PUSH DPH

0072 908002 MOV DPTR,#CRADD1 ;设置读状态字地址

0075 E0 PR01: MOVX A,@DPTR ;读状态字

0076 20E7FC JB ACC.7,PR01 ;判"忙"标志为句"0",否再读

0079 908000 MOV DPTR,#CWADD1 ;设置写指令代码地址

007C E520 MOV A,COM ;取指令代码

007E F0 MOVX @DPTR,A ;写指令代码

007F D083 POP DPH

0081 D082 POP DPL

0083 22 RET

;

;2.写显示数据子程序（E1）

0084 C082 PR1: PUSH DPL

0086 C083 PUSH DPH

0088 908002 MOV DPTR,#CRADD1 ;设置读状态字地址

008B E0 PR11: MOVX A,@DPTR ;读状态宇

008C 20E7FC JB ACC.7,PR11 ;判"忙"标志为"0",否再读

008F 908001 MOV DPTR,#DWADD1 ;设置写显示数据地址

0092 E521 MOV A,DAT ;取数据

0094 F0 MOVX @DPTR,A ;写数据

0095 D083 POP DPH

0097 D082 POP DPL

0099 22 RET

;

;3.读显示数据子程序（E1） 009A C082 PR2: PUSH DPL

009C C083 PUSH DPH

009E 908002 MOV DPTR,#CRADD1 ;设置读状态字地址

00A1 E0 PR21: MOVX A,@DPTR ;读状态字

00A2 20E7FC JB ACC.7,PR21 ;判"忙"标志为"0"否,否再读

00A5 908003 MOV DPTR,#DRADD1 ;设置读显示数据地址

00A8 E0 MOVX A,@DPTR ;读数据

00A9 F521 MOV DAT,A ;存数据

00AB D083 POP DPH

00AD D082 POP DPL

00AF 22 RET

;

;4.写指令代码子程序（E2） 00B0 C082 PR3: PUSH DPL

00B2 C083 PUSH DPH

00B4 908006 MOV DPTR,#CRADD2 ;设置读状态字地址

00B7 E0 PR31: MOVX A,@DPTR ;读状态字

00B8 20E7FC JB ACC.7,PR31 ;判"忙"陈志为"0"否,否再读

00BB 908004 MOV DPTR,#CWADD2 ;设置写指令代码地址

00BE E520 MOV A,COM ;取指令代码

00C0 F0 MOVX @DPTR,A ;写指令代码

00C1 D083 POP DPH

00C3 D082 POP DPL

00C5 22 RET

;

; 5.写显示数据子程序（E2） 00C6 C082 PR4: PUSH DPL

00C8 C083 PUSH DPH

00CA 908006 MOV DPTR,#CRADD2 ;设置读状态字地址

00CD E0 PR41: MOVX A,@DPTR ;读状态字

00CE 20E7FC JB ACC.7,PR41 ;判"忙"标志为"0"否,否再读

00D1 908005 MOV DPTR, #DWADD2 ;设置写显示数据地址

00D4 E521 MOV A,DAT ;取数据

00D6 F0 MOVX @DPTR,A ;写数据

00D7 D083 POP DPH

00D9 D082 POP DPL

00DB 22 RET

;

; 6.读显示数据子程序（E2） 00DC C082 PR5: PUSH DPL

00DE C083 PUSH DPH

00E0 908006 MOV DPTR,#CRADD2 ;设置读状态字地址

00E3 E0 PR51: MOVX A,@DPTR ;读状态字

00E4 20E7FC JB ACC.7,PR51 ;判"忙"标志为"0",否再读

00E7 908007 MOV DPTR,#DRADD2 ;设置写显示数据地址

00EA E0 MOVX A,@DPTR ;读数据

00EB F521 MOV DAT,A ;存数据

00ED D083 POP DPH

00EF D082 POP DPL

00F1 22 RET

; 中文显示子程序

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 00F2 | 900182 | CCW\_PR: | MOV | DPTR,#CCTAB | ;确定字符字模块首地址 |
| 00F5 | E532 |  | MOV | A,CODE\_ | ;取代码 |
| 00F7 | 75F020 |  | MOV | B,#20H | ;字模块宽度为 32 个字节 |
| 00FA | A4 |  | MUL | AB | ;代码×32 |
| 00FB | 2582 |  | ADD | A,DPL | ;字符字模块首地址 |
| 00FD | F582 |  | MOV | DPL,A | ;字模库首地址＋代码×32 |
| 00FF | E5F0 |  | MOV | A,B |  |
| 0101 | 3583 |  | ADDC | A,DPH |  |
| 0103 | F583 |  | MOV | DPH,A |  |
| 0105 | C030 |  | PUSH | COLUMN | ;列地址入栈 |
| 0107 | C030 |  | PUSH | COLUMN | ;列地址入栈 |
| 0109 | 753200 |  | MOV | CODE\_,#00H | ;代码寄存器借用为间址寄存器 |
| 010C | 753310 | CCW\_1: | MOV | COUNT,#10H | ;计数器设置为 16 |
| 010F | E531 |  | MOV | A,PAGE\_ | ;读页地址寄存器 |
| 0111 | 5403 |  | ANL | A,#03H | ;取页地址有效值 |
| 0113 | 44B8 |  | ORL | A,#0B8H | ;"或"页地址设置代码 |
| 0115 | F520 |  | MOV | COM,A | ;设置页地址 |
| 0117 | 12006E |  | LCALL | PRo |  |
| 011A | 1200B0 |  | LCALL | PR3 |  |
| 011D | D030 |  | POP | COLUMN | ;取列地址值 |
| 011F | E530 |  | MOV | A,COLUMN | ;读列地址寄存器 |
| 0121 | C3 |  | CLR | C |  |
| 0122 | 943D |  | SUBB | A,#PD1 | ;列地址-模块参数 |
| 0124 | 4008 |  | JC | CCW\_2 | ;＜0 为左半屏显示区域（E1） |
| 0126 | F530 |  | MOV | COLUMN,A | ;≥0 为右半屏显示区域（E2） |
| 0128 | E531 |  | MOV | A,PAGE\_ |  |
| 012A | D2E3 |  | SETB | ACC.3 | ;设置区域标志位。 |
| 012C | F531 |  | MOV | PAGE\_,A | ;"0"为 E1,"1"为 E2 |
| 012E | 853020 | CCW\_2: | MOV | COM,COLUMN | ;设置列地址值 |
| 0131 | E531 |  | MOV | A,PAGE\_ | ;判区域标志以确定设置哪个控制器 |
| 0133 | 30E306 |  | JNB | ACC.3,CCW\_3 |  |
| 0136 | 1200B0 |  | LCALL | PR3 | ;区域 E2 |
| 0139 | 02013F |  | LJMP | CCW\_4 |  |
| 013C | 12006E | CCW\_3: | LCALL | PRO | ;区域 E1 |
| 013F | E532 | CCW\_4: | MOV | A,CODE\_ | ;取间址寄存器值 |
| 0141 | 93 |  | MOVC | A,@A+DPTR | ;取汉字字模数据 |
| 0142 | F521 |  | MOV | DAT,A | ;写数据 |
| 0144 | E531 |  | MOV | A,PAGE\_ |  |
| 0146 | 30E306 |  | JNB | ACC.3,CCW\_5 |  |
| 0149 | 1200C6 |  | LCALL | PR4 | ;区域 E2 |
| 014C | 020152 |  | LJMP | CCW\_6 |  |
| 014F | 120084 | CCW\_5: | LCALL | PR1 | ;区域 E1 |
| 0152 | 0532 | CCW\_6: | INC | CODE\_ | ;间址寄存器加一 |
| 0154 | 0530 |  | INC | COLUMN | ;列地址寄存器加一 |
| 0156 | E530 |  | MOV | A,COLUMN | ;判列地址是否超出区域范围、 |
| 0158 | B43D00 |  | CJNE | A,#PD1,CCW\_7 |  |
| 015B | 400F | CCW\_7: | JC | CCW\_8 | ;未超出则继续 |
| 015D | E531 |  | MOV | A,PAGE\_ | ;超出则判是否在区域 E2 |
| 015F | 20E30A |  | JB | ACC.3,CCW\_8 | ;在区域 E2 则退出 |
| 0162 | D2E3 |  | SETB | ACC.3 | ;在区域 E1 则修改成区域 E2 |
| 0164 | F531 |  | MOV | PAGE\_,A |  |
| 0166 | 752000 |  | MOV | COM,#00H | ;设置区域 E2 列地址为"0" |
| 0169 | 1200B0 |  | LCALL | PR3 |  |
| 016C | D533D0 | CCW\_8: | DJNZ | COUNT,CCW\_4 | ;当页循环 |
| 016F | E531 |  | MOV | A,PAGE\_ | ;读页地址寄存器 |
| 0171 | 20E70D |  | JB | ACC.7,CCW\_9 | ;判完成标志 D7 位,"1"则完成退出 |
| 0174 | 04 |  | INC | A | ;否则页地址加一 |
| 0175 | D2E7 |  | SETB | ACC.7 | ;置完成位为"1" |
| 0177 | C2E3 |  | CLR | ACC.3 |  |
| 0179 | F531 |  | MOV | PAGE\_,A |  |
| 017B | 753210 |  | MOV | CODE\_,#10H | ;间址寄存器设置为 16 |

017E 02010C LJMP CCW\_1 ;大循环

0181 22 CCW\_9: RET

;

;中文字符库CCTAB:

small0:

;db 38h,44h,44h,44h,44h,44h,38h,00h ;0

small1:

;db 10h,30h,50h,10h,10h,10h,7ch,00h ;1 small2:

;db 38h,44h,04h,08h,10h,20h,7ch,00h ;2 small3:

;db 78h,84h,04h,38h,04h,84h,78h,00h ;3

0182 0004E4

0185 242464

0188 B42F DB 000H,004H,0e4H,024H,024H,064H,0b4H,02fH ; 南

018A 24A464

018D 2424E6

0190 0400 DB 024H,0a4H,064H,024H,024H,0e6H,004H,000H

0192 00007F

0195 040505

0198 057F DB 000H,000H,07fH,004H,005H,005H,005H,07fH

019A 050505

019D 25443F

01A0 0000 DB 005H,005H,005H,025H,044H,03fH,000H,000H

01A2 000404

01A5 E42424

01A8 2526 DB 000H,004H,004H,0e4H,024H,024H,025H,026H ; 京

01AA 242424

01AD E40604

01B0 0000 DB 024H,024H,024H,0e4H,006H,004H,000H,000H

01B2 002010

01B5 190D41

01B8 817F DB 000H,020H,010H,019H,00DH,041H,081H,07fH

01BA 010105

01BD 0D3810

01C0 0000 DB 001H,001H,005H,00dH,038H,010H,000H,000H

01C2 008060

01C5 F8070A

01C8 2828 DB 000H,080H,060H,0f8H,007H,00aH,028H,028H ; 伟

01CA 28FF28

01CD 282828

01D0 0000 DB 028H,0ffH,028H,028H,028H,028H,000H,000H

01D2 010000

01D5 FF0001

01D8 0101 DB 001H,000H,000H,0fFH,000H,001H,001H,001H

01DA 01FF01

01DD 21413F

01E0 0000 DB 001H,0ffH,001H,021H,041H,03fH,000H,000H

01E2 080889

01E5 CE2818

01E8 027A DB 008H,008H,089H,0ceH,028H,018H,002H,07aH ; 福

01EA 4A4A4A

01ED 4A4A7A

01F0 0200 DB 04aH,04aH,04aH,04aH,04aH,07aH,002H,000H

01F2 020100

01F5 FF0102

01F8 FF49 DB 002H,001H,000H,0ffH,001H,002H,0ffH,049H

01FA 49497F

01FD 494949

0200 FF00 DB 049H,049H,07fH,049H,049H,049H,0ffH,000H

; .

; 中文演示显示程序段

MAIN:

0202 12000F LCALL INIT

0205 120046 LCALL CLEAR

0208 753800 MOV CTEMP,#0

020B 753400 MOV DIR,#0 AAA:

020E 753102 MOV PAGE\_,#02H

0211 853830 MOV COLUMN,CTEMP

0214 753200 MOV CODE\_,#00H

0217 1200F2 LCALL CCW\_PR

021A 753102 MOV PAGE\_,#02H

021D E538 MOV A,CTEMP

021F 2410 ADD A,#10H

0221 F530 MOV COLUMN,A

0223 753201 MOV CODE\_,#01H

0226 1200F2 LCALL CCW\_PR

0229 753102 MOV PAGE\_,#02H

022C E538 MOV A,CTEMP

022E 2420 ADD A,#20H

0230 F530 MOV COLUMN,A

0232 753202 MOV CODE\_,#02H

0235 1200F2 LCALL CCW\_PR

0238 753102 MOV PAGE\_,#02H

023B E538 MOV A,CTEMP

023D 2430 ADD A,#30H

023F F530 MOV COLUMN,A

0241 753203 MOV CODE\_,#03H

0244 1200F2 LCALL CCW\_PR

0247 020247 LJMP $

024A 1202AC LCALL DELAY

024D 1202AC LCALL DELAY

0250 1202AC LCALL DELAY

0253 E534 MOV A, DIR

0255 B4000D CJNE A,#0, LEFT

0258 0538 INC CTEMP

025A E538 MOV A,CTEMP

025C B43AAF CJNE A,#58, AAA

025F 753401 MOV DIR,#1

0262 02020E LJMP AAA LEFT:

0265 1538 DEC CTEMP

0267 E538 MOV A,CTEMP

0269 B400A2 CJNE A,#0, AAA

026C 753400 MOV DIR,#0

026F 02020E LJMP AAA

;LCALL S\_LEFT

;LCALL S\_UP

0272 80FE SJMP $

动的效

; 示例五 滚动演示程序段

; 定时间隔地有规律地修改显示起始行地址,将产Th显示画面平滑向上或向下滚

; 果。示例程序如下:

0274 7F00 S\_UP: MOV R7,#00H ;向上改动程序

0276 74C0 S\_UPI: MOV A,#0C0H ;显示起始行设置代码

0278 4F ORL A,R7

0279 F520 MOV COM,A

027B 12006E LCALL PRO

027E 1200B0 LCALL PR3

0281 1202AC LCALL DELAY ;调延时子程序

0284 0F INC R7

0285 BF20EE CJNE R7,#20H,S\_UPI

0288 020274 LJMP S\_UP

;

028B 7F1F S\_DOWN: MOV R7,#1FH ;向下滚动程序

028D 7520C0 MOV COM,#0C0H ;显示起始行设置代码

0290 12006E LCALL PRo

0293 1200B0 LCALL PR3

0296 1202AC LCALL DELAY ;调延时子程序

0299 74C0 S\_DOWN1: MOV A,#0C0H ;显示起始行设置代码

029B 4F ORL A,R7

029C F520 MOV COM,A

029E 12006E LCALL PRO

02A1 1200B0 LCALL PR3

02A4 1202AC LCALL DELAY

02A7 DFF0 DJNZ R7,S\_DOWN1

02A9 02028B LJMP S\_DOWN

;

02AC 7E60 DELAY: MOV R6,#060H ;延时子程序

02AE 7D00 MOV R5,#00H

02B0 00 DELAY1: NOP

02B1 DDFD DJNZ R5,DELAY1

02B3 DEFB DJNZ R6,DELAY1

02B5 22 RET

; 硬件实验二十五 电子琴 H25.ASM

mode equ 082h PORTA equ 08000h

CTL equ 08003h

OUTBIT equ 09002h

IN equ 09001h

Pulse equ 55h PulseCNT equ 50h ToneHigh equ 51h ToneLow equ 52h Tone equ 53h KeyBuf equ 54h

Speaker equ PORTA

|  |  |  |
| --- | --- | --- |
| 0000 020090 | ljmp | Start |
|  | org | 000bh |
|  | Timer0Int: |  |
| 000B C0D0 | push | PSW |
| 000D C28C | clr | TR0 |
| 000F 85518C | mov | TH0, ToneHigh |
| 0012 85528A | mov | TL0, ToneLow |
| 0015 D28C | setb | TR0 |
| 0017 E555 | mov | a, Pulse |
| 0019 908000 | mov | dptr, #Speaker |
| 001C F0 | movx | @dptr,a |

|  |  |  |  |
| --- | --- | --- | --- |
| 001D | 0555 | inc | Pulse |
| 001F | 1550 | dec | PulseCNT |
| 0021 | D0D0 | pop | PSW |
| 0023 | 32 | reti |  |
|  |  | ToneTable: |  |
| 0024 | FC42 |  |  |
| 0026 | FCAE |  |  |
| 0028 | FD0A |  |  |
| 002A | FD35 |  |  |
| 002C | FD82 |  |  |
| 002E | FDC8 |  |  |
| 0030 | FE05 | dw | 64578, 64686, 64778, 64821, 64898, 64968, 65029 |
|  |  | TestKey: |  |
| 0032 | 909002 | mov | dptr, #OUTBIT |
| 0035 | 7400 | mov | a, #0 |
| 0037 | F0 | movx | @dptr, a |
| 0038 | 909001 | mov | dptr, #IN |
| 003B | E0 | movx | a, @dptr |
| 003C | F4 | cpl | a |
| 003D | 540F | anl | a, #0fh |
| 003F | 22 | ret |  |
|  |  | KeyTable: |  |
| 0040 | 161514 |  |  |
| 0043 | FF | db | 16h, 15h, 14h, 0ffh |
| 0044 | 131211 |  |  |
| 0047 | 10 | db | 13h, 12h, 11h, 10h |
| 0048 | 0D0C0B |  |  |
| 004B | 0A | db | 0dh, 0ch, 0bh, 0ah |
| 004C | 0E0306 |  |  |
| 004F | 09 | db | 0eh, 03h, 06h, 09h |
| 0050 | 0F0205 |  |  |
| 0053 | 08 | db | 0fh, 02h, 05h, 08h |
| 0054 | 000104 |  |  |
| 0057 | 07 | db | 00h, 01h, 04h, 07h |
|  |  | GetKey: |  |
| 0058 | 909002 | mov | dptr, #OUTBIT |
| 005B | 8583A0 | mov | P2, dph |
| 005E | 7801 | mov | r0, #Low(IN) |
| 0060 | 7920 | mov | r1, #00100000b |
| 0062 | 7A06 | mov | r2, #6 |
|  |  | KLoop: |  |
| 0064 | E9 | mov | a, r1 |
| 0065 | F4 | cpl | a |
| 0066 | F0 | movx | @dptr, a |
| 0067 | F4 | cpl | a |
| 0068 | 03 | rr | a |
| 0069 | F9 | mov | r1, a |
| 006A | E2 | movx | a, @r0 |
| 006B | F4 | cpl | a |
| 006C | 540F | anl | a, #0fh |
| 006E | 7006 | jnz | Goon1 |
| 0070 | DAF2 | djnz | r2, KLoop |
| 0072 | 7AFF | mov | r2, #0ffh |
| 0074 | 800F | sjmp | Exit |
|  |  | Goon1: |  |
| 0076 | F9 | mov | r1, a |
| 0077 | EA | mov | a, r2 |
| 0078 | 14 | dec | a |
| 0079 | 23 | rl | a |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 007A  007B | 23  FA |  | rl mov | a  r2, a ; r2 = (r2-1)\*4 |
| 007C  007D  007F | E9 7904  13 | LoopC: | mov mov  rrc | a, r1 r1, #4  a |
| 0080 | 4003 |  | jc | Exit |
| 0082 | 0A |  | inc | r2 |
| 0083  0085 | D9FA  909002 | Exit: | djnz  mov | r1, LoopC  dptr, #OUTBIT |
| 0088 | E4 |  | clr | a |
| 0089 | F0 |  | movx | @dptr, a |
| 008A | EA |  | mov | a, r2 |
| 008B | 900040 |  | mov | dptr, #KeyTable |
| 008E | 93 |  | movc | a, @a+dptr |
| 008F  0090 | 22  758170 | Start: | ret  mov | sp, #70h |
| 0093 | 7482 |  | mov | a, #mode |
| 0095 | 908003 |  | mov | dptr,#CTL |
| 0098 | F0 |  | movx | @dptr,a |
| 0099  009C  009F  00A2  00A4 | 758901  75A882  755300  1132  60FC | MLoop: | mov mov mov  call jz | TMOD, #01 ; Timer  IE, #82h ; EA=1, IT0 = 1  Tone,#0  TestKey MLoop |
| 00A6  00A8  00AA  00AC  00AE | 1158  F5F0 60F6  54F8  70F2 |  | call mov jz anl jnz | GetKey b, a  MLoop ; = 0, < 1  a, #!7  MLoop ; > 7 |
| 00B0  00B2  00B4 | 15F0 E5F0 23 |  | dec mov rl | b  a, b  a ; a = a\*2 |
| 00B5 | F5F0 |  | mov | b, a |
| 00B7 | 900024 |  | mov | dptr, #ToneTable |
| 00BA  00BB  00BD | 93  F551 F58C |  | movc mov mov | a, @a+dptr ToneHigh, a TH0, a |
| 00BF | E5F0 |  | mov | a, b |
| 00C1 | 04 |  | inc | a |
| 00C2  00C3  00C5 | 93  F552 F58A |  | movc mov mov | a, @a+dptr ToneLow, a TL0, a |
| 00C7  00C9  00CC  00CE  00D0  00D2 | D28C 755064  E550 70FC C28C 0200A2 | Wait: | setb mov  mov jnz clr ljmp | TR0  PulseCNT, #100  a,PulseCNT Wait  TR0  MLoop |
|  |  |  | end |  |

; 硬件实验二十六 空调温度控制实验 H26.ASM

mode equ 082h PORTA equ 08000h

CTL equ 08003h

OUTBIT equ 09002h

OUTSEG equ 09004h ; 段控制口

IN equ 09001h

LEDBuf equ 60h

ORG 0000H

0000 020109 ljmp Start

LEDMAP:

0003 3F065B

0006 4F666D

0009 7D07 db 3fh, 06h, 5bh, 4fh, 66h, 6dh, 7dh, 07h 000B 7F6F77

000E 7C395E

0011 7971 db 7fh, 6fh, 77h, 7ch, 39h, 5eh, 79h, 71h

Delay:

0013 7F00 mov r7, #0

DelayLoop:

0015 DFFE djnz r7, DelayLoop

0017 DEFC djnz r6, DelayLoop

0019 22 ret

DisplayLED:

|  |  |  |  |
| --- | --- | --- | --- |
| 001A | 7860 | mov | r0, #LEDBuf |
| 001C | 7906 | mov | r1, #6 |
| 001E | 7A20 | mov | r2, #00100000b |
| Loop: | | | |
| 0020 | 909002 | mov | dptr, #OUTBIT |
| 0023 | 7400 | mov | a, #0 |
| 0025 | F0 | movx | @dptr, a |
| 0026 | E6 | mov | a, @r0 |
| 0027 | 909004 | mov | dptr, #OUTSEG |
| 002A | F0 | movx | @dptr, a |
| 002B | 909002 | mov | dptr, #OUTBIT |
| 002E | EA | mov | a, r2 |
| 002F | F0 | movx | @dptr, a |
| 0030 | 7E01 | mov | r6, #1 |
| 0032 | 1113 | call | Delay |
| 0034 | EA | mov | a, r2 |
| 0035 | 03 | rr | a |
| 0036 | FA | mov | r2, a |
| 0037 | 08 | inc | r0 |
| 0038 | D9E6 | djnz | r1, Loop |
| 003A | 909002 | mov | dptr, #OUTBIT |
| 003D | 7400 | mov | a, #0 |
| 003F | F0 | movx | @dptr, a |
| 0040 | 22 | ret |  |

TestKey:

|  |  |  |  |
| --- | --- | --- | --- |
| 0041 | 909002 | mov | dptr, #OUTBIT |
| 0044 | 7400 | mov | a, #0 |
| 0046 | F0 | movx | @dptr, a |
| 0047 | 909001 | mov | dptr, #IN |
| 004A | E0 | movx | a, @dptr |
| 004B | F4 | cpl | a |
| 004C | 540F | anl | a, #0fh |
| 004E | 22 | ret |  |
|  |  | KeyTable: |  |
| 004F | 161514 |  |  |
| 0052 | FF | db | 16h, 15h, 14h, 0ffh |
| 0053 | 131211 |  |  |
| 0056 | 10 | db | 13h, 12h, 11h, 10h |
| 0057 | 0D0C0B |  |  |
| 005A | 0A | db | 0dh, 0ch, 0bh, 0ah |
| 005B | 0E0306 |  |  |
| 005E | 09 | db | 0eh, 03h, 06h, 09h |
| 005F | 0F0205 |  |  |
| 0062 | 08 | db | 0fh, 02h, 05h, 08h |
| 0063 | 000104 |  |  |
| 0066 | 07 | db | 00h, 01h, 04h, 07h |
|  |  | GetKey: |  |
| 0067 | 909002 | mov | dptr, #OUTBIT |
| 006A | 8583A0 | mov | P2, dph |
| 006D | 7801 | mov | r0, #Low(IN) |
| 006F | 7920 | mov | r1, #00100000b |
| 0071 | 7A06 | mov | r2, #6 |
|  |  | KLoop: |  |
| 0073 | E9 | mov | a, r1 |
| 0074 | F4 | cpl | a |
| 0075 | F0 | movx | @dptr, a |
| 0076 | F4 | cpl | a |
| 0077 | 03 | rr | a |
| 0078 | F9 | mov | r1, a |
| 0079 | E2 | movx | a, @r0 |
| 007A | F4 | cpl | a |
| 007B | 540F | anl | a, #0fh |
| 007D | 7006 | jnz | Goon1 |
| 007F | DAF2 | djnz | r2, KLoop |
| 0081 | 7AFF | mov | r2, #0ffh |
| 0083 | 800F | sjmp | Exit |
|  |  | Goon1: |  |
| 0085 | F9 | mov | r1, a |
| 0086 | EA | mov | a, r2 |
| 0087 | 14 | dec | a |
| 0088 | 23 | rl | a |
| 0089 | 23 | rl | a |
| 008A | FA | mov | r2, a ; r2 = (r2-1)\*4 |
| 008B | E9 | mov | a, r1 |
| 008C | 7904 | mov | r1, #4 |
|  |  | LoopC: |  |

008E 13 rrc a

008F 4003 jc Exit

0091 0A inc r2

0092 D9FA djnz r1, LoopC Exit:

0094 EA mov a, r2

0095 90004F mov dptr, #KeyTable 0098 93 movc a, @a+dptr

0099 FA mov r2, a WaitRelease:

009A 909002 mov dptr, #OUTBIT

009D E4 clr a

009E F0 movx @dptr, a

009F 7E0A mov r6, #10

00A1 1113 call Delay

00A3 1141 call TestKey

00A5 70F3 jnz WaitRelease

00A7 EA mov a, r2

00A8 22 ret

UP equ 16h ; Next

DOWN equ 15h ; Last LowLimit equ 10

HighLimit equ 30

LowTemp equ -40 ; A/D 0

HighTemp equ 50 ; A/D 255

ADPort equ 0a000h Heat equ 1

Cool equ 2

SetTemp equ 50h CurTemp equ 51h

DisplayResult:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 00A9  00AB | E551 30E708 |  | mov jnb | a, CurTemp acc.7, GE0 |  |
| 00AE | 756040 |  | mov | LEDBuf, #40h | ; '-' |
| 00B1  00B2  00B3  00B6 | 14  F4 0200B9  756000 | GE0: | dec cpl jmp  mov | a a  Goon  LEDBuf, #0 ; ' | ' |
| 00B9 | 75F00A | Goon: | mov | b, #10 |  |
| 00BC | 84 |  | div | ab |  |
| 00BD | 900003 |  | mov | dptr, #LEDMAP |  |
| 00C0  00C1 | 93  F561 |  | movc mov | a, @a+dptr LEDBuf+1, a |  |
| 00C3 | E5F0 |  | mov | a, b |  |
| 00C5  00C6 | 93  F562 |  | movc mov | a, @a+dptr LEDBuf+2, a |  |
| 00C8 | 756300 |  | mov | LEDBuf+3, #0 | ; ' ' |
| 00CB  00CD  00D0 | E550 75F00A  84 |  | mov mov  div | a, SetTemp b, #10  ab |  |
| 00D1  00D2 | 93  F564 |  | movc mov | a, @a+dptr LEDBuf+4, a |  |
| 00D4 | E5F0 |  | mov | a, b |  |
| 00D6  00D7 | 93  F565 |  | movc  mov | a, @a+dptr  LEDBuf+5, a |  |

00D9 22 ret

ReadAD:

00DA 90A000 mov dptr, #ADPort

00DD E4 clr a

00DE F0 movx @dptr, a ; start A/D

00DF 7400 mov a, #0

00E1 D5E0FD djnz acc, $ ; delay

00E4 E0 movx a, @dptr

00E5 22 ret

ReadTemp:

00E6 7900 mov r1, #0

00E8 7A00 mov r2, #0

00EA 7810 mov r0, #16

RLoop:

00EC 11DA call ReadAD

00EE 2A add a, r2

00EF FA mov r2, a

00F0 5001 jnc GN1

00F2 09 inc r1 GN1:

00F3 D8F7 djnz r0, RLoop

00F5 EA mov a, r2

00F6 C4 swap a

00F7 540F anl a, #0fh

00F9 C9 xch a, r1

00FA C4 swap a

00FB 54F0 anl a, #0f0h

00FD 49 orl a, r1 ; a = r1r2/16

00FE 75F05A mov b, #(HighTemp-LowTemp) 0101 A4 mul ab

0102 E5F0 mov a, b ; /256

0104 24D8 add a, #LowTemp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0106 | F551 |  | mov | CurTemp, a |
| 0108  0109 | 22  758170 | Start: | ret  mov | sp, #70h |
| 010C | 7482 |  | mov | a, #mode |
| 010E  0111 | 908003  F0 |  | mov movx | dptr, #CTL @dptr, a |
| 0112  0115 | 755014  1141 | MLoop: | mov  call | SetTemp, #20  TestKey |
| 0117 | 703D |  | jnz | KeyPressed |
| 0119  011B  011D | 11A9  111A  11E6 |  | call call call | DisplayResult DisplayLED ReadTemp |
| 011F  0121 | E551 20E70C |  | mov jb | a, CurTemp acc.7, LE0 |
| 0124 | C3 |  | clr | c |
| 0125 | 8550F0 |  | mov | b, SetTemp |
| 0128  012A  012C  012E | 15F0  15F0  95F0  5008 |  | dec dec subb  jnc | b b  a, b  GN2 |

; 设置 8255 工作方式,PA,PC 输出,PB 输入

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0130 | 7401 | LE0: | mov | a, #Heat |
| 0132 | 908000 |  | mov | dptr, #PORTA |
| 0135 | F0 |  | movx | @dptr, a |
| 0136  0138 | 801C  E551 | GN2: | sjmp  mov | GN4  a, CurTemp |
| 013A  013B  013E  0140 | D3 8550F0  05F0  05F0 |  | setb mov inc  inc | c  b, SetTemp b  b |
| 0142 | 95F0 |  | subb | a, b |
| 0144 | 4008 |  | jc | GN3 |
| 0146 | 7402 |  | mov | a, #Cool |
| 0148  014B  014C  014E | 908000  F0 8006  7400 | GN3: | mov movx sjmp  mov | dptr, #PORTA @dptr, a GN4  a, #0 |
| 0150 | 908000 |  | mov | dptr, #PORTA |
| 0153 | F0 | GN4: | movx | @dptr, a |
| 0154  0156 | 80BF  1167 | sjmp KeyPressed:  call | | MLoop  GetKey |
| 0158 | F5F0 | mov | | b, a |
| 015A | 6415 | xrl | | a, #DOWN |
| 015C | 700A | jnz | | Key0 |
| 015E | E550 | mov | | a, SetTemp |
| 0160 | 640A | xrl | | a, #LowLimit |
| 0162 | 6012 | jz | | Key1 |
| 0164 | 1550 | dec | | SetTemp |
| 0166 | 800E | sjmp | | Key1 |
|  |  | Key0: |  |  |
| 0168 | E5F0 | mov | | a, b |
| 016A | 6416 | xrl | | a, #UP |
| 016C | 7008 | jnz | | Key1 |
| 016E | E550 | mov | | a, SetTemp |
| 0170 | 641E | xrl | | a, #HighLimit |
| 0172 | 6002 | jz | | Key1 |
| 0174 | 0550 | inc | | SetTemp |
|  |  | Key1: |  |  |

0176 809D sjmp MLoop end

# 附录 B MCS51 C 语言程序清单

// 软件实验一 存储器块清零 S1.C

xdata unsigned char Buffer[256] \_at\_ 0x3000; void main()

{

unsigned int index; unsigned char xdata \* ptr;

ptr = &Buffer; // 起始地址

for (index = 0; index <= 255; index++) {

\*ptr++ = 0; // 清 0, 地址加一

}

}

// 软件实验二 二进制到 BCD 码转换 S2.C

// 将 Number 拆为三个 BCD 码, 并存入 Result 数组

void main()

{

unsigned char Result[3]; unsigned char Number;

Number = 123;

Result[0] = Number / 100; // 除以 100, 得百位数Result[1] = (Number 100) / 10; // 余数除以 10, 得十位数Result[2] = Number 10; // 余数为个位数

}

// 软件实验三 二进制到 ASCII 码转换 S3.C

// 将 A 拆为二个 ASCII 码, 并存入 Result 数组

code unsigned char ASCIITable[16] = "0123456789ABCDEF"; // 定义数字对应的 ASCII 表void main()

{

unsigned char Result[2]; unsigned char Number;

Number = 0x1a;

Result[0] = ASCIITable[Number / 16]; // 高四位

Result[1] = ASCIITable[Number & 0xf]; // 低四位

}

// 软件实验四 内存块移动 S4.C

xdata unsigned char Buffer1[256] \_at\_ 0x3000; xdata unsigned char Buffer2[256] \_at\_ 0x4000;

void main()

{

unsigned int index; unsigned char xdata \* ptr1; unsigned char xdata \* ptr2;

ptr1 = &Buffer1; ptr2 = &Buffer2;

for (index=0; index <= 255; index++) {

\*ptr2++ = \*ptr1++;

}

}

// 软件实验五 程序跳转表 S5.C

void Func0() {}

void Func1() {}

void Func2() {}

void Func3() {}

void FuncEnter(unsigned char FuncID)

{

switch (FuncID) {

case 0: Func0(); break; case 1: Func1(); break; case 2: Func2(); break; case 3: Func3(); break; default: break;

}

}

void main()

{

FuncEnter(0); FuncEnter(1); FuncEnter(2); FuncEnter(3); while(1);

}

// 软件实验六 数据排序 S6.C #define Size 10

unsigned char Array[Size];

void main ()

{

bit Change; unsigned char index; unsigned char Temp;

do {

Change = 0;

for (index = 0; index < (Size-1); index++) { if (Array[index] > Array[index+1]) {

Change = 1;

Temp = Array[index]; Array[index] = Array[index+1]; Array[index+1] = Temp;

}

}

} while (Change);

}

// 硬件实验一 P1 口输入、输出实验 H1A.C #include <reg51.h>

void delay()

{

unsigned int i;

for (i=0; i<20000; i++) {}

}

void main()

{

unsigned char index; unsigned char LED;

while (1) {

LED = 1;

for (index=0; index < 8; index++) { P1 = LED;

LED <<= 1;

delay();

}

}

}

// 硬件实验一 P1 口输入、输出实验 H1B.C #include <reg51.h>

sbit KeyLeft = P1^0; sbit KeyRight = P1^1; sbit LEDLeft = P1^2;

sbit LEDRight = P1^3;

void main()

{

while (1) {

LEDLeft = KeyLeft;

LEDRight = KeyRight;

}

}

// 硬件实验二 继电器控制实验 H2.C #include <reg51.h>

sbit Output = P1^0;

void Delay()

{

unsigned int i;

for (i=0; i<20000; i++) ;

}

void main()

{

while (1) {

Output = 0; Delay(); Output = 1; Delay();

}

}

// 硬件实验三 用 74LS245 扩展 I/O 口实验 H3.C xdata unsigned char CS245 \_at\_ 0x8000;

void main()

{

unsigned char b;

while (1) { b = CS245;

}

}

// 硬件实验四 用 74LS273 扩展 I/O 口实验 H4.C xdata unsigned char CS273 \_at\_ 0x8000;

void main()

{

unsigned char i, b;

b = 1;

for (i=0; i<8; i++) { CS273 = b;

b <<= 1;

}

}

// 硬件实验五 PWM 转换电压实验 H5.C

// 输出 50 (5:5) 占空比 PWM

// 输出 10 (1:9) 占空比 PWM

// 输出 90 (9:1) 占空比 PWM #include <reg51.h>

sbit OUTPUT = P1^0;

void Delay(unsigned char CNT)

{

unsigned char i; while (CNT--) {

for (i=0; i<50; i++) ;

}

}

void main()

{

while (1) {

OUTPUT = 0;

Delay(5); OUTPUT = 1;

Delay(5);

}

}

// 硬件实验六 音频控制实验 H6.C #include <reg51.h>

sbit Speaker = P1^0;

void Delay()

{

unsigned char i;

for (i=0; i<50; i++);

}

void main()

{

while (1) {

Speaker = 0; Delay();

Speaker = 1; Delay();

}

}

// 硬件实验七 8255 输入、输出实验 H7.C #define mode 0x82 // 方式 0，PA，PC 输出，PB 输入

xdata unsigned char PortA \_at\_ 0x8000 ; // Port A xdata unsigned char PortB \_at\_ 0x8001 ; // Port B xdata unsigned char PortC \_at\_ 0x8002 ; // Port C xdata unsigned char CAddr \_at\_ 0x8003 ; // 控制字地址

void delay(unsigned char CNT)

{

unsigned int i;

while (CNT-- !=0)

for (i=20000; i !=0; i--);

}

void main()

{

register unsigned char i, dd;

CAddr = mode; // 方式 0, PA,PC 输出, PB 输入

while(1){

dd = 0x80;

for(i = 0; i<8; i++)

{

PortA = dd; // 输出到 PA dd >>= 1; // 移位

delay(1); // 延时

};

dd = PortB; // PB 输入PortA = dd; // 再输出到 PA delay(2);

}

}

// 硬件实验八 串行数转换并行数实验 H8A.C #include <reg51.h>

void Send164(unsigned char b)

{

SCON = 0; // 串口方式 0 SBUF = b; // 输出 b

}

void main()

{

Send164(0x55);

while(1);

}

// 硬件实验八 串行数转换并行数实验 H8B.C #include <reg51.h>

sbit DAT = P1^0; sbit CLK = P1^1;

void Send164(unsigned char b)

{

unsigned char i;

CLK = 1;

for (i=0; i<8; i++) { DAT = (b & 1);

CLK = 0;

CLK = 1;

b >>= 1;

}

}

void main()

{

Send164(0x55);

while(1);

}

// 硬件实验九 并行数转换串行数实验 H9A.C #include <reg51.h>

sbit LD = P1^0;

unsigned char Read165()

{

LD = 0;

LD = 1;

SCON = 0x10;

while (!RI) ; return(SBUF);

}

void main()

{

unsigned char b;

b = Read165();

}

// 硬件实验九 并行数转换串行数实验 H9B.C #include <reg51.h>

sbit LD = P1^0; sbit DAT = P1^1; sbit CLK = P1^2;

unsigned char Read165()

{

unsigned char i, b;

LD = 0;

LD = 1;

b = 0;

for (i=0; i<8; i++) { b >>= 1;

if (DAT) b |= 0x80;

CLK = 0;

CLK = 1;

}

return(b);

}

void main()

{

unsigned char b;

b = Read165();

}

// 硬件实验十 计数器实验 H10.C #include <reg51.h>

void main()

{

TMOD = 0x05; // 方式 1,记数器TH0 = 0;

TL0 = 0;

TR0 = 1; // 开始记数

while (1) P1 = TL0; // 将记数结果送 P1 口

}

//硬件实验十一 外部中断实验 H11.C #include <reg51.h>

sbit LED = P1^0; bit LEDBuf;

void ExtInt0() interrupt 0

{

LEDBuf = !LEDBuf;

LED = LEDBuf;

}

void main()

{

LEDBuf = 0;

LED = 0;

TCON = 0x01; // 外部中断 0 下降沿触发

IE = 0x81; // 打开外部中断允许位(EX0)及总中断允许位(EA) while (1) ;

}

// 硬件实验十二 定时器实验 H12.C #include <reg51.h>

#define Tick 10000 // 10000 x 100us = 1s #define T100us (256-50) // 100us 时间常数(6M)

unsigned int C100us; // 100us 记数单元bit LEDBuf;

sbit LED = P1^0;

void T0Int() interrupt 1

{

C100us--;

if (C100us == 0) {

C100us = Tick; // 100us 记数器为 0, 重置记数器

LEDBuf = !LEDBuf; // 取 反 LED

}

}

void main()

{

TMOD = 0x02; // 方式 2, 定时器

TH0 = T100us;

TL0 = T100us;

IE = 0x82; // EA=1, IT0 = 1 LEDBuf = 0;

LED = 0;

C100us = Tick;

TR0 = 1; // 开始定时

while (1) {

LED = LEDBuf;

}

}

// 硬件实验十三 D/A 转换实验 H13.C xdata unsigned char CS0832 \_at\_ 0x8000;

void Write0832(unsigned char b)

{

CS0832 = b;

}

void main()

{

Write0832(0); Write0832(0x80); Write0832(0xff); while(1);

}

// 硬件实验十四 A/D 转换实验 H14.C

#define mode 0x82

xdata unsigned char CTL \_at\_ 0x9003; xdata unsigned char PA \_at\_ 0x9000; xdata unsigned char CS0809 \_at\_ 0x8000;

unsigned char Read0809()

{

unsigned char i;

CS0809 = 0; // 起 动 A/D for (i=0; i<0x20; i++) ; // 延时 > 100us return(CS0809); // 读入结果

}

main()

{

unsigned char b; CTL = mode; while(1){

b = Read0809(); PA = b;

}

}

//硬件实验十五 外部中断实验（急救车与交通灯） H15.C #include "reg51.h"

#define ON 1

#define OFF 0

sbit SR = 0x90; // 南北红灯sbit SY = 0x91; // 南北黄灯sbit SG = 0x92; // 南北绿灯

sbit ER = 0x93; // 东西红灯sbit EY = 0x94; // 东西黄灯sbit EG = 0x95; // 东西绿灯

bit Flash; // LED 状态

bit STOP;

void StopInt() interrupt 0

{

STOP = 1;

}

void Delay(unsigned char CNT)

{

unsigned int I;

while ((CNT > 0) && !STOP) {

for (I=0; (I < 10000) && !STOP; I++) ; CNT--;

}

}

void main()

{

unsigned char I;

TCON = 0x01; // INT0 下沿中断IE = 0x81; // EA =1, EX0=1 STOP = 0;

SR = ON; // 南北, 东西均红灯

SY = OFF;

SG = OFF;

ER = ON;

EY = OFF;

EG = OFF;

while (1) {

if (STOP) goto AllRed;

SR = ON; // 南北红灯, 东西绿灯

SY = OFF;

SG = OFF;

ER = OFF; EY = OFF; EG = ON;

Delay(20);

if (STOP) goto AllRed;

SR = ON; // 南北红灯, 东西黄灯闪

SY = OFF; SG = OFF;

ER = OFF; EY = OFF; EG = OFF;

Flash = OFF;

for (I=0; I < 9; I++) {

EY = Flash;

Delay(1);

Flash = !Flash;

}

if (STOP) goto AllRed;

SR = OFF; // 南北绿灯, 东西红灯

SY = OFF;

SG = ON;

ER = ON;

EY = OFF;

EG = OFF;

Delay(20);

if (STOP) goto AllRed;

SR = OFF; // 东西红灯, 南北黄灯闪

SY = OFF;

SG = OFF;

ER = ON;

EY = OFF;

EG = OFF;

Flash = OFF;

for (I=0; I < 9; I++) {

SY = Flash;

Delay(1);

Flash = !Flash;

}

AllRed:

if (STOP) {

SR = ON; // 南北, 东西均红灯

SY = OFF;

SG = OFF;

ER = ON;

EY = OFF;

EG = OFF;

STOP = 0;

Delay(10);

}

}

}

//硬件实验十六 八段数码管显示实验 H16.C #define LEDLen 6

xdata unsigned char OUTBIT \_at\_ 0x8002; // 位控制口xdata unsigned char OUTSEG \_at\_ 0x8004; // 段控制口xdata unsigned char IN \_at\_ 0x8001; // 键盘读入口

unsigned char LEDBuf[LEDLen]; // 显示缓冲

code unsigned char LEDMAP[] = { // 八段管显示码0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71

};

void Delay(unsigned char CNT)

{

unsigned char i;

while (CNT-- !=0)

for (i=100; i !=0; i--);

}

void DisplayLED()

{

unsigned char i, j; unsigned char Pos; unsigned char LED;

Pos = 0x20; // 从左边开始显示

for (i = 0; i < LEDLen; i++) { OUTBIT = 0; // 关所有八段管LED = LEDBuf[i];

OUTSEG = LED;

OUTBIT = Pos; // 显示一位八段管

Delay(1);

Pos >>= 1; // 显示下一位

}

OUTBIT = 0; // 关所有八段管

}

void main()

{

unsigned char i = 0; unsigned char j;

while(1) {

LEDBuf[0] = LEDMAP[ i & 0x0f]; LEDBuf[1] = LEDMAP[(i+1) & 0x0f]; LEDBuf[2] = LEDMAP[(i+2) & 0x0f]; LEDBuf[3] = LEDMAP[(i+3) & 0x0f]; LEDBuf[4] = LEDMAP[(i+4) & 0x0f]; LEDBuf[5] = LEDMAP[(i+5) & 0x0f];

i++;

for(j=0; j<30; j++) DisplayLED(); // 延时

}

}

//硬件实验十七 键盘扫描显示实验 H17.C #define LEDLen 6

xdata unsigned char OUTBIT \_at\_ 0x8002; // 位控制口xdata unsigned char OUTSEG \_at\_ 0x8004; // 段控制口xdata unsigned char IN \_at\_ 0x8001; // 键盘读入口

unsigned char LEDBuf[LEDLen]; // 显示缓冲

code unsigned char LEDMAP[] = { // 八段管显示码0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71

};

void Delay(unsigned char CNT)

{

unsigned char i;

while (CNT-- !=0)

for (i=100; i !=0; i--);

}

void DisplayLED()

{

unsigned char i; unsigned char Pos; unsigned char LED;

Pos = 0x20; // 从左边开始显示

for (i = 0; i < LEDLen; i++) { OUTBIT = 0; // 关所有八段管

LED = LEDBuf[i]; OUTSEG = LED;

OUTBIT = Pos; // 显示一位八段管

Delay(1);

Pos >>= 1; // 显示下一位

}

}

code unsigned char KeyTable[] = { // 键码定义

0x16, 0x15, 0x14, 0xff, 0x13, 0x12, 0x11, 0x10,

0x0d, 0x0c, 0x0b, 0x0a, 0x0e, 0x03, 0x06, 0x09,

0x0f, 0x02, 0x05, 0x08,

0x00, 0x01, 0x04, 0x07

};

unsigned char TestKey()

{

OUTBIT = 0; // 输出线置为 0

return (~IN & 0x0f); // 读入键状态(高四位不用)

}

unsigned char GetKey()

{

unsigned char Pos; unsigned char i; unsigned char k;

i = 6;

Pos = 0x20; // 找出键所在列

do {

OUTBIT = ~ Pos;

Pos >>= 1;

k = ~IN & 0x0f;

} while ((--i != 0) && (k == 0));

// 键值 = 列 x 4 + 行

if (k != 0) { i \*= 4;

if (k & 2) i += 1;

else if (k & 4) i += 2;

else if (k & 8) i += 3;

OUTBIT = 0;

do Delay(10); while (TestKey()); // 等键释放

return(KeyTable[i]); // 取出键码

} else return(0xff);

}

void main()

{

LEDBuf[0] = 0xff; LEDBuf[1] = 0xff; LEDBuf[2] = 0xff; LEDBuf[3] = 0xff; LEDBuf[4] = 0x00; LEDBuf[5] = 0x00;

while (1) { DisplayLED();

if (TestKey()) LEDBuf[5] = LEDMAP[GetKey() & 0x0f];

}

}

//硬件实验十八 电子时钟 H18.C #include <reg51.h>

#define LEDLen 6

xdata unsigned char OUTBIT \_at\_ 0x8002; // 位控制口

xdata unsigned char OUTSEG \_at\_ 0x8004; // 段控制口

unsigned char LEDBuf[LEDLen]; // 显示缓冲

code unsigned char LEDMAP[] = { // 八段管显示码0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71

};

void Delay(unsigned char CNT)

{

unsigned char i;

while (CNT-- !=0)

for (i=100; i !=0; i--);

}

void DisplayLED()

{

unsigned char i, j; unsigned char Pos; unsigned char LED;

Pos = 0x20; // 从左边开始显示

for (i = 0; i < LEDLen; i++) { OUTBIT = 0; // 关所有八段管LED = LEDBuf[i];

OUTSEG = LED;

OUTBIT = Pos; // 显示一位八段管

Delay(1);

Pos >>= 1; // 显示下一位

}

}

// ===============================================================================

#define Tick 10000 // 10000 x 100us = 1s #define T100us (256-50) // 100us 时间常数(6M)

unsigned char Hour, Minute, Second; unsigned int C100us; // 100us 记数单元

void T0Int() interrupt 1

{

C100us--;

if (C100us == 0) {

C100us = Tick; // 100us 记数器为 0, 重置记数器

Second++;

if (Second == 60) { Second = 0; Minute++;

if (Minute == 60) { Minute = 0; Hour++;

if (Hour == 24) Hour = 0;

}

}

}

}

void main()

{

TMOD = 0x02; // 方式 2, 定时器

TH0 = T100us;

TL0 = T100us;

IE = 0x82; // EA=1, IT0 = 1

Hour = 0;

Minute = 0;

Second = 0; C100us = Tick;

TR0 = 1; // 启动定时器 0

while (1) {

LEDBuf[0] = LEDMAP[Hour/10]; LEDBuf[1] = LEDMAP[Hour 10] | 0x80; LEDBuf[2] = LEDMAP[Minute/10]; LEDBuf[3] = LEDMAP[Minute 10] | 0x80; LEDBuf[4] = LEDMAP[Second/10]; LEDBuf[5] = LEDMAP[Second 10];

DisplayLED();

}

}

//硬件实验十九 单片机串行口通讯实验 H19.C

#include <reg51.h> #define LEDLen 6

xdata unsigned char OUTBIT \_at\_ 0x8002; // 位控制口xdata unsigned char OUTSEG \_at\_ 0x8004; // 段控制口xdata unsigned char IN \_at\_ 0x8001; // 键盘读入口

unsigned char LEDBuf[LEDLen]; // 显示缓冲

unsigned char RcvBuf; // 接收缓冲

bit HasRcv = 0; // 接收标志

code unsigned char LEDMAP[] = { // 八段管显示码0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71

};

void SerialIO0() interrupt 4

{

if(RI)

{

RI = 0;

RcvBuf = SBUF;

HasRcv = 1;

}else

{

TI = 0;

}

}

void Delay(unsigned char CNT)

{

unsigned char i;

while (CNT-- !=0)

for (i=100; i !=0; i--);

}

void DisplayLED()

{

unsigned char i, j; unsigned char Pos; unsigned char LED;

Pos = 0x20; // 从左边开始显示

for (i = 0; i < LEDLen; i++) { OUTBIT = 0; // 关所有八段管LED = LEDBuf[i];

OUTSEG = LED;

OUTBIT = Pos; // 显示一位八段管

Delay(1);

Pos >>= 1; // 显示下一位

}

}

code unsigned char KeyTable[] = { // 键码定义

0x16, 0x15, 0x14, 0xff, 0x13, 0x12, 0x11, 0x10,

0x0d, 0x0c, 0x0b, 0x0a, 0x0e, 0x03, 0x06, 0x09,

0x0f, 0x02, 0x05, 0x08,

0x00, 0x01, 0x04, 0x07

};

unsigned char TestKey()

{

OUTBIT = 0; // 输出线置为 0

return (~IN & 0x0f); // 读入键状态 (高四位不用)

}

unsigned char GetKey()

{

unsigned char Pos; unsigned char i;

//unsigned char j; unsigned char k;

i = 6;

Pos = 0x20; // 找出键所在列

do {

OUTBIT = ~ Pos;

Pos >>= 1;

k = ~IN & 0x0f;

} while ((--i != 0) && (k == 0));

// 键值 = 列 X 4 + 行

if (k != 0) { i \*= 4;

if (k & 2) i += 1;

else if (k & 4) i += 2;

else if (k & 8) i += 3;

OUTBIT = 0;

do Delay(10); while (TestKey()); // 等键释放

return(KeyTable[i]); // 取出键码

} else return(0xff);

}

void main()

{

IE = 0x00; // DISABLE ALL INTERRUPT

TMOD = 0x20; // 定时器 1 工作于方式 2 (8 位重装) TH1 = 0xF3; // 2400BPS @ 12MHz

TL1 = 0xF3;

PCON&= 0x7F; // SMOD 位清零

SCON = 0x50; // 串行口工作方式设置

LEDBuf[0] = 0xff; LEDBuf[1] = 0xff; LEDBuf[2] = 0xff; LEDBuf[3] = 0xff; LEDBuf[4] = 0x00; LEDBuf[5] = 0x00;

TR1 = 1;

ES = 1;

EA = 1;

HasRcv = 0;

while (1) { if(HasRcv){

LEDBuf[5] = LEDMAP[RcvBuf & 0x0f];

}

DisplayLED();

if (TestKey()) SBUF = GetKey();

}

}

//硬件实验二十 打印机控制试验 H20.C

#define mode 0x82 /\* 8255 工作方式, PA,PC 输出, PB 输入 \*/

xdata unsigned char control \_at\_ 0x8003; /\* 8255 控 制 口 地 址 \*/ xdata unsigned char dport \_at\_ 0x08000; /\* 数据，8255 PA 口地址 \*/

xdata unsigned char status \_at\_ 0x08001; /\* 忙状态, 8255 PB 口地址 \*/ xdata unsigned char ctl \_at\_ 0x08002; /\* 选通控制, 8255 PC 口地址 \*/

#define busy 1 /\* 忙状态位 \*/

code char string[] = {'H','e','l','l','o',' ',

'W','o','r','l','d','!',

'\n',0x0c,0x00};

void print(unsigned char dd)

{

while(status & busy); // 读打印机状态, 若忙，等待

dport = dd; // 数据送出

ctl = 1; // 输出选通脉冲

ctl = 0;

}

void main()

{

char \*pos;

control = mode;

pos = &string; // 被打印字串

while((\*pos)) print( \*pos++); while(1);

}

//硬件实验二十一 直流电机控制实验 H21.C #include <reg52.h>

#define mode 0x82

xdata unsigned char CTL \_at\_ 0x8003; xdata unsigned char status \_at\_ 0x8001; xdata unsigned char CS0832 \_at\_ 0x9000; unsigned int count;

#define DC\_P 1 void delay()

{

unsigned int ddd;

ddd = 50000; // 在 6MHz 约延时 1 秒while(ddd--);

}

unsigned int read()

{

TMOD = 1; // 16 位计时TR0 = 0;

TH0 = 0;

TL0 = 0;

while(!(status & DC\_P)); // 等待低电平完while(status & DC\_P); // 等待高电平完TR0 = 1;

while(!(status & DC\_P)); // 等待低电平完while(status & DC\_P); // 等待高电平完TR0 = 0;

return (TH0\*0x100+TL0);

}

void main()

{

CTL = mode;

CS0832 = 0xff; // 产Th电压控制电机

delay(); // 等待电机运转稳定

count = read(); // 读取时间

CS0832 = 0xc0; // 产Th电压控制电机

delay(); // 等待电机运转稳定

count = read(); // 读取时间

while(1);

}

//硬件实验二十二 步进电机控制实验 H22.C #define mode8255 0x82

xdata unsigned char control \_at\_ 0x8003; xdata unsigned char ctl \_at\_ 0x8000;

#define Astep 0x01 #define Bstep 0x02 #define Cstep 0x04 #define Dstep 0x08

unsigned char dly\_c; void delay()

{

unsigned char tt,cc;

cc = dly\_c; tt = 0x0; do{

do {

}while(--tt);

}while(--cc);

}

void main()

{

unsigned char mode;

control = mode8255; mode = 2;

ctl = 0;

dly\_c = 0x10;

// 单/双八拍工作方式if(mode ==1) while(1)

{

ctl = Astep; delay();

ctl = Astep+Bstep; delay();

ctl = Bstep; delay();

ctl = Bstep+Cstep; delay();

ctl = Cstep; delay();

ctl = Cstep+Dstep; delay();

ctl = Dstep; delay();

ctl = Dstep+Astep; delay();

if(dly\_c>2) dly\_c --;

};

// 双四拍工作方式if(mode == 2) while(1)

{

ctl = Astep+Bstep; delay();

ctl = Bstep+Cstep; delay();

ctl = Cstep+Dstep; delay();

ctl = Dstep+Astep; delay();

if(dly\_c>3) dly\_c --;

};

// 单四拍工作方式if(mode ==3) while(1)

{

ctl = Dstep; delay(); ctl = Cstep; delay(); ctl = Bstep; delay();

ctl = Astep; delay();

if(dly\_c>4) dly\_c --;

}

while(1);

}

//硬件实验二十三 温度传感器试验 H23.C #include <reg51.h>

#define LEDLen 6

#define LowTemp (-99) // A/D 0

#define HighTemp 99 // A/D 255

xdata unsigned char OUTBIT \_at\_ 0x9002;

xdata unsigned char OUTSEG \_at\_ 0x9004; // 段控制口

xdata unsigned char IN \_at\_ 0x9001; xdata unsigned char ADPort \_at\_ 0x8000;

signed char CurTemp;

unsigned char LEDBuf[LEDLen]; code unsigned char LEDMAP[] = {

0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71

};

void Delay(unsigned char CNT)

{

unsigned char i;

do{

i = 100;

do{

}while(--i);

}while (--CNT);

}

void DisplayLED()

{

unsigned char i; unsigned char Pos; unsigned char LED;

// 初始位置为右边

Pos = 0x20;

for (i = 0; i < LEDLen; i++) {

// 关闭所显示位

OUTBIT = 0;

// 取出显示数据

LED = LEDBuf[i]; OUTSEG = LED;

// 输出位选通信号,选择一位 LED 点亮

OUTBIT = Pos;

// 延时

Delay(2);

// 移位,选择下一位 LED 准备点亮

Pos >>= 1;

}

OUTBIT = 0;

}

void DisplayResult()

{

signed char T;

T = CurTemp; if (T < 0) {

//如果温度低于零度,显示'-'号LEDBuf[0] = 0x40;

T = - T;

} else LEDBuf[0] = 0;

//显示温度十位数

LEDBuf[1] = LEDMAP[T / 10];

//显示温度个位数

LEDBuf[2] = LEDMAP[T 10];

LEDBuf[3] = 0;

}

unsigned char ReadAD()

{

unsigned int i;

// 启动 A/D 变换ADPort = 0;

//延时 100us,

for (i=0; i<20; i++);

//得到 A/D 采样值

return (0xff - ADPort);

}

void ReadTemp()

{

unsigned char i; signed int Temp;

Temp = 0;

//采样 16 次,取平均数

for (i=0; i<16; i++) { Temp += ReadAD();

};

//AD 采样值转换为实际温度

CurTemp = (Temp >> 4) \* (HighTemp-LowTemp) / 256 + LowTemp;

}

void main()

{

unsigned char dtimer;

while (1) {

for(dtimer=15; dtimer>0; dtimer--){

// 当前温度和设定温度送显示缓冲

DisplayResult();

// 显示当前温度和设定温度

DisplayLED();

};

// 读入当前温度

ReadTemp();

}

}

//硬件实验二十四 液晶显示控制电路 H24.C #include <reg51.h>

#define PD1 61 // 122/2 分成左右两半屏(122x32)

unsigned char Column;

unsigned char Page\_ ; // 页地址寄存器 D1,DO:页地址unsigned char Code\_ ; // 字符代码寄存器

unsigned char Command; // 指令寄存器

unsigned char LCDData; // 数据寄存器

xdata unsigned char CWADD1 \_at\_ 0x8000; // 写指令代码地址(E1) xdata unsigned char DWADD1 \_at\_ 0x8001; // 写显示数据地址(E1) xdata unsigned char CRADD1 \_at\_ 0x8002; // 读状态字地址(E1) xdata unsigned char DRADD1 \_at\_ 0x8003; // 读显示数据地址(E1)

xdata unsigned char CWADD2 \_at\_ 0x8004; // 写指令代码地址(E2) xdata unsigned char DWADD2 \_at\_ 0x8005; // 写显示数进地址(E2) xdata unsigned char CRADD2 \_at\_ 0x8006; // 读状态字地址(E2) xdata unsigned char DRADD2 \_at\_ 0x8007; // 读显示数据地址(E2)

//英文字模库 8x8 点阵

code unsigned char EETAB[][8]={

{0x38,0x44,0x44,0x44,0x44,0x44,0x38,0x00}, // 0

{0x10,0x30,0x50,0x10,0x10,0x10,0x7c,0x00}, // 1

{0x38,0x44,0x04,0x08,0x10,0x20,0x7c,0x00}, // 2

{0x78,0x84,0x04,0x38,0x04,0x84,0x78,0x00}}; // 3

// 中文字模库 16x16 点阵

code unsigned char CCTAB[][32] ={

{0x00,0x04,0xe4,0x24,0x24,0x64,0xb4,0x2f, // 南

0x24,0xa4,0x64,0x24,0x24,0xe6,0x04,0x00,

0x00,0x00,0x7f,0x04,0x05,0x05,0x05,0x7f,

0x05,0x05,0x05,0x25,0x44,0x3f,0x00,0x00},

{0x00,0x04,0x04,0xe4,0x24,0x24,0x25,0x26, // 京

0x24,0x24,0x24,0xe4,0x06,0x04,0x00,0x00,

0x00,0x20,0x10,0x19,0x0D,0x41,0x81,0x7f,

0x01,0x01,0x05,0x0d,0x38,0x10,0x00,0x00},

{0x00,0x80,0x60,0xf8,0x07,0x0a,0x28,0x28, // 伟

0x28,0xff,0x28,0x28,0x28,0x28,0x00,0x00,

0x01,0x00,0x00,0xfF,0x00,0x01,0x01,0x01,

0x01,0xff,0x01,0x21,0x41,0x3f,0x00,0x00},

{0x08,0x08,0x89,0xce,0x28,0x18,0x02,0x7a, // 福0x4a,0x4a,0x4a,0x4a,0x4a,0x7a,0x02,0x00, 0x02,0x01,0x00,0xff,0x01,0x02,0xff,0x49, 0x49,0x49,0x7f,0x49,0x49,0x49,0xff,0x00}};

// 1.写指令代码子程序(E1) void WriteCommandE1()

{

while(CRADD1 & 0x80); CWADD1 = Command;

}

// 2.写显示数据子程序(E1) void WriteDataE1()

{

while(CRADD1 & 0x80); DWADD1 = LCDData;

}

// 3.读显示数据子程序(E1) void ReadDataE1()

{

while(CRADD1 & 0x80); LCDData = DRADD1;

}

// 4.写指令代码子程序(E2) void WriteCommandE2()

{

while(CRADD2 & 0x80); CWADD2 = Command;

}

// 5.写显示数据子程序(E2) void WriteDataE2()

{

while(CRADD2 & 0x80); DWADD2 = LCDData;

}

// 6.读显示数据子程序(E2) void ReadDataE2()

{

while(CRADD2 & 0x80); LCDData = DRADD2;

}

// 初始化程序

void Init()

{

Command = 0xe2; WriteCommandE1(); WriteCommandE2();

Command = 0xa4; WriteCommandE1(); WriteCommandE2();

Command = 0xa9; WriteCommandE1(); WriteCommandE2();

Command = 0xa0; WriteCommandE1(); WriteCommandE2();

Command = 0xc0; WriteCommandE1(); WriteCommandE2();

Command = 0xaf;

WriteCommandE1();

WriteCommandE2();

}

// 清屏

void Clear()

{

unsigned char i,j;

i = 0;

do {

Command = (i + 0xb8); // 页地址设置

WriteCommandE1(); WriteCommandE2();

Command = 0x00; // 列地址设置为"0"

WriteCommandE1(); WriteCommandE2();

j = 0x50; // 一页清 80 个字节

do {

LCDData = 0x00; // 显示数据为"0" WriteDataE1();

WriteDataE2();

}while(--j !=0); // 页内字节清零循环

}while(++i !=4); // 页地址暂存器加一

// 显示区清零循环

}

// 延时程序

void Delay()

{

unsigned char i,j; i = 20;

j = 0;

do{

do{

}while(--j !=0);

}while( --i != 0 );

}

// 中文显示子程序

void WriteCHN16x16()

{

unsigned char i,j,k;

i = 0;

j = 0;

while(j<2) {

Command = ((Page\_ + j) & 0x03) | 0xb8; // 设置页地址

WriteCommandE1(); WriteCommandE2();

k = Column; // 列地址值

while(k < Column + 16){

if (k < PD1) { // 为左半屏显示区域(E1) Command = k;

WriteCommandE1(); // 设置列地址值LCDData = CCTAB[Code\_][i]; // 取汉字字模数据WriteDataE1(); // 写字模数据

} else{ // 为右半屏显示区域(E2) Command = k-PD1;

WriteCommandE2(); // 设置列地址值LCDData = CCTAB[Code\_][i]; // 取汉字字模数据WriteDataE2(); // 写字模数据

}; i++;

if( ++k >= PD1 \* 2) break; // 列地址是否超出显示范围

} ; j++;

};

}

// 中文显示程序

void main()

{

Init();

Clear();

Page\_ = 0x02; Column = 0x00; Code\_ = 0x00; WriteCHN16x16();

Page\_ = 0x02; Column = 0x10; Code\_ = 0x01; WriteCHN16x16();

Page\_ = 0x02; Column = 0x20; Code\_ = 0x02; WriteCHN16x16();

Page\_ = 0x02; Column = 0x30; Code\_ = 0x03; WriteCHN16x16();

while(1);

}

// 硬件实验二十五 电子琴 H25.C #include <reg51.h>

#define mode 0x82;

xdata unsigned char Control \_at\_ 0x8003; xdata unsigned char Speaker \_at\_ 0x8000;

xdata unsigned char OUTBIT \_at\_ 0x9002; xdata unsigned char OUTSEG \_at\_ 0x9004; xdata unsigned char IN \_at\_ 0x9001;

code unsigned char KeyTable[] = { 0x16, 0x15, 0x14, 0xff,

0x13, 0x12, 0x11, 0x10,

0x0d, 0x0c, 0x0b, 0x0a, 0x0e, 0x03, 0x06, 0x09,

0x0f, 0x02, 0x05, 0x08,

0x00, 0x01, 0x04, 0x07

};

unsigned char TestKey()

{

OUTBIT = 0;

return (~IN & 0x0f);

}

unsigned char GetKey()

{

unsigned char Pos; unsigned char i; unsigned char k;

i = 6;

Pos = 0x20; do {

OUTBIT = ~ Pos;

Pos >>= 1;

k = ~IN & 0x0f;

} while ((--i != 0) && (k == 0));

if (k != 0) { i \*= 4;

if (k & 2) i += 1;

else if (k & 4) i += 2;

else if (k & 8) i += 3;

OUTBIT = 0;

return(KeyTable[i]);

} else return(0xff);

}

/\* ================================================================================ \*/

code unsigned int ToneTable[7] = {

64578, 64686, 64778, 64821, 64898, 64968, 65029

};

unsigned char PulseCNT; unsigned char Pulse; unsigned ToneHigh, ToneLow;

void Timer() interrupt 1

{

TR0 = 0;

TH0 = ToneHigh;

TL0 = ToneLow;

TR0 = 1;

Speaker = Pulse;

Pulse ++;

PulseCNT--;

}

void main()

{

unsigned char Key;

Control = mode;

OUTSEG = 0;

Pulse = 0;

TMOD = 0x01; // 方式 1 , 记时器

IE = 0x82; // EA=1, IT0 = 1

while (1) {

while (! TestKey) ; // 等按键

Key = GetKey();

if ( (Key >= 1) && (Key <= 7) ) { ToneHigh = ToneTable[Key-1] >> 8; ToneLow = ToneTable[Key-1] & 0xff; TH0 = ToneHigh;

TL0 = ToneLow;

TR0 = 1;

PulseCNT = 100;

while (PulseCNT != 0) ; // 发 100 个脉冲

Speaker = 0;

TR0 = 0;

}

}

}

//硬件实验二十六 空调温度控制实验 H26.C #include <reg51.h>

#define LEDLen 6

#define PB1 2

#define PB0 1 #define mode 0x82

xdata unsigned char CTL \_at\_ 0x8003; xdata unsigned char PA \_at\_ 0x8000;

xdata unsigned char OUTBIT \_at\_ 0x9002; // 位控制地址

xdata unsigned char OUTSEG \_at\_ 0x9004; // 段控制地址

xdata unsigned char IN \_at\_ 0x9001;

unsigned char LEDBuf[LEDLen]; code unsigned char LEDMAP[] = {

0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71

};

void Delay(unsigned char CNT)

{

unsigned char i;

while (CNT-- !=0)

for (i=100; i !=0; i--);

}

void DisplayLED()

{

unsigned char i, j; unsigned char Pos; unsigned char LED;

// 初始位置为右边

Pos = 0x20;

for (i = 0; i < LEDLen; i++) {

// 关闭所显示位

OUTBIT = 0;

// 取出显示数据LED = LEDBuf[i]; OUTSEG = LED;

// 输出位选通信号,选择一位 LED 点亮

OUTBIT = Pos;

// 延时

Delay(1);

// 移位,选择下一位 LED 准备点亮

Pos >>= 1;

}

OUTBIT = 0;

}

code unsigned char KeyTable[] = { 0x16, 0x15, 0x14, 0xff,

0x13, 0x12, 0x11, 0x10,

0x0d, 0x0c, 0x0b, 0x0a, 0x0e, 0x03, 0x06, 0x09,

0x0f, 0x02, 0x05, 0x08,

0x00, 0x01, 0x04, 0x07

};

unsigned char TestKey()

{

//输出列扫描信号

OUTBIT = 0;

//读入行信号

return (~IN & 0x0f);

}

unsigned char GetKey()

{

unsigned char Pos; unsigned char i; unsigned char k;

// 共扫描 6 列i = 6;

Pos = 0x20; do {

// 输出列扫描信号

OUTBIT = ~ Pos;

// 移位,准备扫描下一列

Pos >>= 1;

// 读入行值

k = ~IN & 0x0f;

// 当 6 列扫描完或读到行值时,退出

} while ((--i != 0) && (k == 0));

//如果有键输入if (k != 0) {

// 列值 x4 + 行值 为键值

i \*= 4;

if (k & 2) i += 1;

else if (k & 4) i += 2;

else if (k & 8) i += 3;

OUTBIT = 0;

// 等待按键松开

do Delay(10); while (TestKey());

// 根据键值查表得到键码,并返回键码

return(KeyTable[i]);

} else return(0xff);

}

/\* =============================\*/

#define UP 0x16 #define DOWN 0x15

#define LowLimit 10

#define HighLimit 30

#define LowTemp (-40) // A/D 0

#define HighTemp 50 // A/D 255

#define Heat 1

#define Cool 2

xdata unsigned char ADPort \_at\_ 0xa000; signed char SetTemp, CurTemp;

void DisplayResult()

{

signed char T;

T = CurTemp; if (T < 0) {

//如果温度低于零度,显示'-'号LEDBuf[0] = 0x40;

T = - T;

} else LEDBuf[0] = 0;

//显示温度十位数

LEDBuf[1] = LEDMAP[T / 10];

//显示温度个位数

LEDBuf[2] = LEDMAP[T 10];

LEDBuf[3] = 0;

//后两位显示设定温度

LEDBuf[4] = LEDMAP[SetTemp / 10]; LEDBuf[5] = LEDMAP[SetTemp 10];

}

unsigned char ReadAD()

{

unsigned int i;

// 启动 A/D 变换ADPort = 0;

//延时 100us,

for (i=0; i<20; i++) ;

//得到 A/D 采样值

return (ADPort);

}

void ReadTemp()

{

unsigned char i; signed int Temp;

Temp = 0;

//采样 16 次,取平均数

for (i=0; i<16; i++) { Temp += ReadAD();

};

//AD 采样值转换为实际温度

CurTemp = (Temp/16) \* (HighTemp-LowTemp) / 256 + LowTemp;

}

void main()

{

unsigned char Key;

CTL = mode;

//初始设定温度为 20

SetTemp = 20;

while (1) {

while (!TestKey()) {

// 当前温度和设定温度送显示缓冲

DisplayResult();

// 显示当前温度和设定温度

DisplayLED();

// 读入当前温度

ReadTemp();

// 如果当前温度低于设定温度 2 度

// 则开动加热电机

if (CurTemp < (SetTemp - 2)) { PA = Heat ;

} else

// 如果当前温度高于设定温度 2 度

// 则开动致冷电机

if (CurTemp > (SetTemp + 2)) { PA = Cool;

} else {

// 在设定温度+/- 2 度内

// 无需开动电机

PA = 0;

}

}

// 读键盘

Key = GetKey();

if (Key == DOWN) {

// 设定温度向下调低

if (SetTemp > LowLimit) SetTemp--;

} else if (Key == UP) {

// 设定温度向上调高

if (SetTemp < HighLimit) SetTemp++;

}

}

}

// 硬件实验二十七 计算器实验 H27.C

#define LEDLen 6

#define PB1 2

#define PB0 1

xdata unsigned char OUTBIT \_at\_ 0x9002;

xdata unsigned char OUTSEG \_at\_ 0x9004; // 段控制口

xdata unsigned char IN \_at\_ 0x9001;

unsigned char LEDBuf[LEDLen]; unsigned char LEDMAP[] = {

0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71

};

void Delay(unsigned char CNT)

{

unsigned char i;

while (CNT-- !=0)

for (i=100; i !=0; i--);

}

void DisplayLED()

{

unsigned char i, j; unsigned char Pos; unsigned char LED;

Pos = 0x20;

for (i = 0; i < LEDLen; i++) { OUTBIT = 0; // trun off all LED LED = LEDBuf[i];

OUTSEG = LED;

OUTBIT = Pos; // trun on one LED Delay(1);

Pos >>= 1;

}

}

unsigned char KeyTable[] = { 0x16, 0x15, 0x14, 0xff, 0x13, 0x12, 0x11, 0x10,

0x0d, 0x0c, 0x0b, 0x0a, 0x0e, 0x03, 0x06, 0x09,

0x0f, 0x02, 0x05, 0x08,

0x00, 0x01, 0x04, 0x07

};

unsigned char TestKey()

{

OUTBIT = 0;

return (~IN & 0x0f);

}

unsigned char GetKey()

{

unsigned char Pos; unsigned char i; unsigned char k;

i = 6;

Pos = 0x20; do {

OUTBIT = ~ Pos;

Pos >>= 1;

k = ~IN & 0x0f;

} while ((--i != 0) && (k == 0));

if (k != 0) { i \*= 4;

if (k & 2) i += 1;

else if (k & 4) i += 2;

else if (k & 8) i += 3;

OUTBIT = 0;

do Delay(10); while (TestKey());

return(KeyTable[i]);

} else return(0xff);

}

/\* =====================================================================================\*/

#define ADD 0x0a #define SUB 0x0b #define MUL 0x0c #define DIV 0x0d #define EQU 0x0e #define CLR 0x0f

void DisplayResult(signed int Result)

{

unsigned char i;

if (Result >= 0) { // 数值大于 0

// 不显示符号位

LEDBuf[0] = 0;

} else {

// 在符号位显示 ‘-’

LEDBuf[0] = 0x40;

Result = - Result;

}

LEDBuf[1] = 0;

LEDBuf[2] = 0;

LEDBuf[3] = 0;

LEDBuf[4] = 0;

// 数值对 10 取模，所得结果为最低位

LEDBuf[5] = LEDMAP[Result 10]; Result = Result / 10; i = 4;

// 如果数值除以 10 后不等于 0，继续向高位显示while (Result != 0) {

LEDBuf[i--] = LEDMAP[Result 10]; Result = Result / 10;

}

}

void main()

{

signed int Last, Result; unsigned char OP; unsigned char Key;

Last = 0;

Result = 0; OP = ADD;

DisplayResult(Result); while (1) {

while (!TestKey()) DisplayLED();

// 读取键码

Key = GetKey();

if ( (Key >= 0) && (Key <= 9) ) { // 是数字键

// 输入数值

Result = Result\*10 + Key;

// 数值送显示缓冲

DisplayResult(Result);

} else if (Key == CLR) { // 是清零键

// 状态清零

Last = 0;

Result = 0; OP = ADD;

// 数值送显示缓冲

DisplayResult(Result);

} else if ( (Key==ADD) || (Key==SUB) || (Key==MUL) || (Key==DIV) || (Key=EQU) ) {

// 是功能键

switch (OP) { //上一按键是? case ADD:

Result = Last + Result; break;

case SUB:

Result = Last - Result; break;

case MUL:

Result = Last \* Result;

break; case DIV:

Result = Last / Result; break;

}

if (Key == EQU) { // 功能键是‘=’

// 计算最后结果

Last = 0; OP = ADD;

} else { // 是其它功能键

// 保存结果, 保存按键功能

Last = Result;

OP = Key;

}

// 计算值送显示缓冲, DisplayResult(Result);

// 等待下一个数值输入. Result = 0;

}

}

}

# 附录 C MCS96 汇编语言程序清单

; 软件实验一 存储器块清零 S1.ASM

org 2080h Block equ 6000h

2080 A100601E ld bx, #Block ; 起始地址

2084 B10020 ldb cl, #0 ; 清 256 字节

Loop:

2087 C61F00 stb 0, [bx]+

208A E020FA djnz cl, Loop ; 记数减一

208D E7FDFF ljmp $ end

; 软件实验二 二进制到 BCD 码转换 S2.ASM

; 将 AX 拆为三个 BCD 码, 并存入 Result 开始的叁个单元

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Result equ | 30h |
| 2080 | E71F00 | org ljmp | 2080h  Start |
| 2083 | A1640020 | BinToBCD:  ld | cl, #100 |
| 2087 | FE9C201C | divb | ax, cl |
| 208B | B01C30 | ldb | Result, al ; 除以 100, 得百位数 |
| 208E | B01D1C | ldb | al, ah |
| 2091 | B0001D | ldb | ah, 0 |
| 2094 | B10A20 | ldb | cl, #10 |
| 2097 | FE9C201C | divb | ax, cl |
| 209B | B01C31 | ldb | Result+1, al ; 余数除以 10, 得十位数 |
| 209E  20A1 | B01D32 F0 | ldb ret | Result+2, ah ; 余数为个位数 |

Start:

20A2 A1E00018 ld sp, #0e0h

20A6 A17B001C ld ax, #123

20AA 2FD7 call BinToBCD

20AC E7FDFF ljmp $

end

; 软件实验三 二进制到 ASCII 码转换 S3.ASM

; 将 AX 拆为二个 ASCII 码, 并存入 Result 开始的二个单元

Result equ 30h

org 2080h

2080 E72800 ljmp Start

ASCIITab:

2083 30313233

2087 34353637

208B 38394142

208F 43444546 db '0123456789ABCDEF' ; 定义数字对应的 ASCII 表

BinToHex:

2093 B01C1E ldb bl, al

2096 18041E shrb bl, #4

2099 B0001F ldb bh, 0

209C B31F832030 ldb Result, ASCIITab[bx]

20A1 510F1C1E andb bl, al, #0fh

20A5 B31F832031 ldb Result+1, ASCIITab[bx]

20AA F0 ret

Start:

20AB A1E00018 ld sp, #0e0h

20AF A11A001C ld ax, #1ah

20B3 2FDE call BinToHex

20B5 E7FDFF ljmp $

end

; 软件实验四 内存块移动 S4.ASM

; 移动 3000H --> 4000H, 256 字节

org 2080h

2080 A100401C ld ax, #4000h

2084 A100501E ld bx, #5000h

2088 B10020 ldb cl, #0

Loop:

208B B21D21 ldb ch, [ax]+

208E C61F21 stb ch, [bx]+

2091 E020F7 djnz cl, Loop

2094 E7FDFF ljmp $

end

; 软件实验五 程序跳转表 S5.ASM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2080 | E72500 |  | org ljmp | 2080h  Start |
| 2083 | A100001C | Func0: | ld | ax, #0 |
| 2087 | F0 |  | ret |  |
| 2088  208C | A101001C F0 | Func1: | ld ret | ax, #1 |
| 208D  2091 | A102001C F0 | Func2: | ld ret | ax, #2 |
| 2092 | A103001C | Func3: | ld | ax, #3 |
| 2096 | F0 |  | ret |  |
| 2097 | 641C1C | FuncEnter:  add | | ax, ax ; ljmp 为二字节指令, 调用号 X 2 |
| 209A | 65A0201C | add | | ax, #FuncTab |
| 209E  20A0 | E31C  27E1 | br FuncTab:  sjmp | | [ax]  Func0 |
| 20A2 | 27E4 | sjmp | | Func1 |
| 20A4 | 27E7 | sjmp | | Func2 |
| 20A6 | 27EA | sjmp | | Func3 |

Start:

|  |  |  |  |
| --- | --- | --- | --- |
| 20A8 | A1E00018 | ld | sp, #0e0h |
| 20AC | A100001C | ld | ax, #0 |
| 20B0 | 2FE5 | call | FuncEnter |
| 20B2 | A101001C | ld | ax, #1 |
| 20B6 | 2FDF | call | FuncEnter |
| 20B8 | A102001C | ld | ax, #2 |
| 20BC | 2FD9 | call | FuncEnter |
| 20BE | A103001C | ld | ax, #3 |
| 20C2 | 2FD3 | call | FuncEnter |
| 20C4 | E7FDFF | ljmp  end | $ |

; 软件实验六 数据排序 S6.ASM

|  |  |  |  |
| --- | --- | --- | --- |
| Size | equ | 10 | ; 数据个数 |
| Array | equ | 50h | ; 数据起始地址 |
| Change | equ | 60h | ; 交换标志 |

org 2080h

2080 A1E00018 ld sp, #0e0h

2084 B10A20 ldb cl, #Size

2087 A1C4201E ld bx, #DataArr

208B A1500022 ld dx, #Array MoveData:

208F B21F1C ldb al, [bx]+

2092 C6231C stb al, [dx]+

2095 E020F7 djnz cl, MoveData Sort:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2098 | A150001E | ld | bx, #Array | |
| 209C | B10920 | ldb | cl, #Size-1 | |
| 209F | B00060 | ldb | Change, 0 | |
|  |  | Goon: |  | |
| 20A2 | B21F1C | ldb | al, [bx]+ |  |
| 20A5 | 9A1E1C | cmpb | al, [bx] |  |
| 20A8 | DF0F | je | Next |  |
| 20AA | D30D | jnc | Next | ; 前小后大, 不交换 |
| 20AC | B10160 | ldb | Change, #1 | ; 前大后小, 置交换标志 |
| 20AF | B21E1D | ldb | ah, [bx] |  |
| 20B2 | C61E1C | stb | al, [bx] | ; 交换 |
| 20B5 | C71EFF1D | stb | ah, -1[bx] |  |

Next:

20B9 E020E6 djnz cl, Goon

20BC 980060 cmpb Change, 0

20BF D7D7 jne Sort

20C1 E7FDFF ljmp $ DataArr:

|  |  |  |
| --- | --- | --- |
| 20C4 | 05020100 |  |
| 20C8 | 02030806 |  |
| 20CC | 0509 | db 5,2,1,0,2,3,8,6,5,9  end |

; 硬件实验一 P1 口输入、输出实验 H1A.ASM

org 2080h

Loop:

2080 B1011C ldb al, #01h

2083 B10420 ldb cl, #4

Output:

2086 B01C0F ldb IOPORT1, al

2089 19011C shlb al, #1

208C EF0600 call Delay

208F E020F4 djnz cl, Output

2092 E7EBFF ljmp Loop Delay:

2095 A0001E ld bx, 0

DLoop:

2098 E01EFD djnz bl, DLoop

209B E01FFA djnz bh, DLoop

209E F0 ret

end

; 硬件实验一 P1 口输入、输出实验 H1B.ASM

KeyLeft equ 1 ; P1.0 KeyRight equ 2 ; P1.1 LEDLeft equ 4 ; P1.2 LEDRight equ 8 ; P1.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P1Buf | | | equ 40h | |
|  |  | org | | 2080h |
| 2080 | A1E00018 | ld | | sp, #0e0h |
| 2084 | B00F40 | ldb | | P1Buf, IOPORT1 ; 读入 P1 口初值 |
| 2087 | 910340 | orb | | P1Buf, #(KeyLeft+KeyRight) ; 相应输入位置高 |
| 208A | B0400F | ldb | | IOPORT1, P1Buf ; 输出到 P1 口 |
|  |  | Loop: | |  |
| 208D | 51010F1C | andb | | al,IOPORT1, #KeyLeft ; 与运算，结果送 AL |
| 2091 | D705 | jne | | KeyLeftOn ; 判断 P1 相应位是否为高 |
| 2093 | 71FB40 | andb | | P1Buf, #! LEDLeft |
| 2096 | 2003 | sjmp | | Goon1 |
|  |  | KeyLeftOn: | |  |
| 2098 | 910440 | orb | | P1Buf, #LEDLeft |
|  |  | Goon1: |  |  |
| 209B | 51020F1C | andb | | al, IOPORT1, #KeyRight |
| 209F | D705 | jne | | KeyRightOn |
| 20A1 | 71F740 | andb | | P1Buf, #! LEDRight |
| 20A4 | 2003 | sjmp | | Goon2 |
|  |  | KeyRightOn: | |  |
| 20A6 | 910840 | orb | | P1Buf, #LEDRight |
|  |  | Goon2: |  |  |
| 20A9 | B0400F | ldb | | IOPORT1, P1Buf |
| 20AC | 27DF | sjmp | | Loop |
|  |  | end | |  |

; 硬件实验二 继电器控制实验 H2.ASM

Output equ 1 ; P1.0 P1Buf equ 30h

org 2080h

2080 B00F30 ldb P1Buf, IOPORT1

Loop:

2083 71FE30 andb P1Buf, #!Output

2086 B0300F ldb IOPORT1, P1Buf

2089 EF0C00 call Delay

208C 910130 orb P1Buf, #Output

208F B0300F ldb IOPORT1, P1Buf

2092 EF0300 call Delay

2095 E7EBFF ljmp Loop

Delay:

2098 A1000020 ld cx, #0

DLoop:

209C E120FD djnzw cx, DLoop

209F F0 ret

end

; 硬件实验三 用 74LS245 扩展 I/O 口实验 H3.ASM

CS245 equ 8000h

org 2080h

Loop:

2080 B30100801C ldb al, CS245

2085 E7F8FF ljmp Loop

end

; 硬件实验四 用 74LS273 扩展 I/O 口实验 H4.ASM

CS273 equ 8000h

org 2080h

start:

2080 B10820 ldb cl, #8

2083 B1011C ldb al, #1

Loop:

2086 C70100801C stb al, CS273

208B 19011C shlb al, #1

208E EF0500 call delay

2091 E020F2 djnz cl, Loop

2094 27EA sjmp start delay:

2096 A1204E1E ld bx,#20000

209A E11EFD djnzw bx, $

209D F0 ret

end

; 硬件实验五 PWM 转换电压实验 H5.ASM

; 输出 50 (5:5) 占空比 PWM

; 输出 10 (1:9) 占空比 PWM

; 输出 90 (9:1) 占空比 PWM

OUTPUT equ 1 ; P1.0 P1 equ IOPORT1

P1BUf equ 30h

org 2080h

2080 B00F30 ldb P1Buf, P1

Loop:

2083 71FE30 andb P1Buf, #!OUTPUT ; OUTPUT = 0

2086 B0300F ldb P1, P1Buf

2089 B10521 ldb ch, #5

208C EF0F00 call Delay

208F 910130 orb P1Buf, #OUTPUT ; OUTPUT =1

2092 B0300F ldb P1, P1Buf

2095 B10521 ldb ch, #5

2098 EF0300 call Delay

209B E7E5FF ljmp Loop

Delay:

209E B00020 ldb cl, 0

DLoop:

20A1 E120FD djnzw cx, DLoop

20A4 F0 ret

end

; 硬件实验六 音频控制实验 H6.ASM

Speaker equ 1 ; P1.0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | P1 | equ | IOPORT1 |
| P1Buf | equ | 30h |
|  | org | 2080h |
| 2080 | A1E00018 |  | ld | sp, #0e0h |
| 2084  2087 | B00F30  71FE30 | Loop: | ldb  andb | P1Buf, P1  P1Buf, #!Speaker |
| 208A  208D | B0300F EF0C00 |  | ldb call | P1, P1Buf ; Speaker = 0 Delay |
| 2090 | 910130 |  | orb | P1Buf, #Speaker |
| 2093 | B0300F |  | ldb | P1, P1Buf ; Speaker = 1 |
| 2096 | EF0300 |  | call | Delay |
| 2099 | E7EBFF |  | ljmp | Loop |
| 209C | B1C020 | Delay: | ldb | cl, #0c0h |
| 209F  20A2 | E020FD F0 |  | djnz ret | cl, $ |
|  |  |  | end |  |

; 硬件实验七 8255 输入、输出实验 H7.ASM

mode equ 082h ; 方式 0，PA，PC 输出，PB 输入

PortA equ 8000h ; Port A PortB equ 8001h ; Port B PortC equ 8002h ; Port C CAddr equ 8003h ; 控制字地址

org 2080h

2080 A1E00018 ld sp, #0e0h

Start:

2084 B1821C ldb al, #mode

2087 C70103801C stb al,CAddr ; 输出控制字

;实验 1：PortA 输出

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 208C | B1081C |  | ldb | al,#08H |  | |
| 208F | B10421 |  | ldb | ch,#4H |
| 2092 | C70100801C | OutA: | stb | al,PortA | ; | 输出 PortA |
| 2097 | C70102801C |  | stb | al,PortC |  |  |
| 209C | 18011C |  | shrb | al,#1 | ; | 移位 |
| 209F | B1641D |  | ldb | ah,#100 |  |  |
| 20A2 | EF2100 |  | call | delay | ; | 延时 |
| 20A5 | E021EA |  | djnz | ch,OutA |  |  |
| ;实验 2： PortB 输入 PortA 输出 | | | | | | |
| 20A8 | B30101801C | ldb | | al, PortB | ; | 读入 PortB |
| 20AD | C70102801C | stb | | al, PortC |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 20B2 | C70100801C |  | stb | al, PortA ; 输出到 PortA |
| 20B7  20BA  20BD  20C0 | B1C81D EF0900 B1C81D EF0300 |  | ldb call ldb call | ah,#200  delay ah,#200  delay |
| 20C3 | E7BEFF |  | ljmp | Start |
|  |  | delay: |  |  |
| 20C6  20C8  20CB  20CE  20D0 | C81C B1001C E11CFD CC1C F0 | ddd: | push ax ldb al,#0  djnzw ax,ddd pop ax  ret  end | |

|  |  |  |
| --- | --- | --- |
| ; 硬件实验八 | 串行数转换并行数实验 H8A.ASM |  |
|  | org 2080h |
| 2080 912016 | orb IOC1, #20h | ; P2.0 选择为 TXD |
| 2083 B10011  2086 B1E10E  2089 B1840E  208C B15507  208F 3511FD  2092 E7FDFF | ldb SP\_CON, #0  ldb baud\_rate,#0e1h ldb baud\_rate,#084h ldb SBUF, #55h  jbc sp\_stat,5,$ ljmp $  end | ; 串口方式 0  ; 输出 55h |
| ; 硬件实验八 | 串行数转换并行数实验 H8B.ASM |  |

|  |  |  |  |
| --- | --- | --- | --- |
| DAT | equ | 1 | ; P1.0 |
| CLK | equ | 2 | ; P1.1 |

P1 equ IOPORT1

P1Buf equ 30h

org 2080h

2080 B00F30 ldb P1Buf, P1

2083 B1551C ldb al, #055h

Send:

2086 B10820 ldb cl, #8

2089 910230 orb P1Buf, #CLK

208C B0300F ldb P1, P1Buf ; CLK = 1

Loop:

208F 910130 orb P1Buf, #DAT

2092 381C03 jbs al, 0, B1

2095 71FE30 andb P1Buf, #!DAT ; DAT = al bit 0 B1:

2098 71FD30 andb P1Buf, #!CLK ; CLK = 0

209B B0300F ldb P1, P1Buf

209E 910230 orb P1Buf, #CLK ; CLK = 1

20A1 B0300F ldb P1, P1Buf

20A4 18011C shrb al, #1

20A7 E020E5 djnz cl, Loop

20AA E7FDFF ljmp $ end

; 硬件实验九 并行数转换串行数实验 H9A.ASM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| LD165 | | | equ | 1 | |
|  |  | P1 | equ IOPORT1 | |  |
|  |  |  | org 2080h | |  |
| 2080 | A1C00018 |  | ld sp, #0c0h | |  |
| 2084 | B12016 |  | ldb ioc1, #20h | | ; P2.0 设为 TXD 功能 |
| 2087  208A | B1800E B1800E |  | ldb baud\_rate,#80h ldb baud\_rate,#80h | | ; 波特率参数  ; 2400bps @ 6MHz |
| 208D  2090 | 71FE0F  91010F |  | andb P1, #!LD165 orb P1, #LD165 | | ; LD = 0  ; LD = 1 |
| 2093  2096 | B10811  B0111C | Wait: | ldb SP\_CON, #08  ldb al, SP\_STAT | | ; 串口方式 0, REN = 1 |
| 2099  209C  209F | 361CFA B0071C FD |  | jbc al, 6, Wait ldb al, SBUF nop | | ; RI = 1?  ; 输出 55h |
| 20A0 | E7FDFF |  | ljmp $ | |  |
|  |  |  | end | |  |

; 硬件实验九 并行数转换串行数实验 H9B.ASM

|  |  |  |  |
| --- | --- | --- | --- |
| LD165 | equ | 1 | ; P1.0 |
| DAT | equ | 2 | ; P1.1 |
| CLK | equ | 8 | ; P1.2 |

P1 equ IOPORT1

P1Buf equ 30h

org 2080h

2080 B00F30 ldb P1Buf, P1

2083 910A30 orb P1Buf, #DAT+CLK ; DAT = 1, CLK = 1

2086 71FE30 andb P1Buf, #!LD165 ; LD165 = 0

2089 B0300F ldb P1, P1Buf

208C 910130 orb P1Buf, #LD165 ; LD165 = 1

208F B0300F ldb P1, P1Buf

2092 B0001C ldb al, 0

2095 B10820 ldb cl, #8

Next:

2098 18011C shrb al, #1

209B 310F03 jbc P1, 1, B1

209E 91801C orb al, #80h B1:

20A1 71F730 andb P1Buf, #!CLK

20A4 B0300F ldb P1, P1Buf

20A7 910830 orb P1Buf, #CLK

20AA B0300F ldb P1, P1Buf

20AD E020E8 djnz cl, Next

20B0 E7FDFF ljmp $ end

; 硬件实验十一 外部中断实验 H11.ASM

LED equ 1 ; P1.0

P1Buf equ 30h

P1 equ IOPORT1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | org | 200eh |
| 200E | 9720 |  | dcw | EXTINT |
|  |  |  | org | 2080h |
| 2080 | E70000 |  | ljmp | Start |
|  |  | Start: |  |  |
| 2083 | 71FD16 |  | andb | IOC1, #!02h |
| 2086 | B00F30 |  | ldb | P1Buf, P1 |
| 2089 | 71FE30 |  | andb | P1Buf, #!LED |
| 208C | B0300F |  | ldb | P1, P1Buf |
| 208F | B18008 |  | ldb | INT\_MASK, #80h ; enable extint |
| 2092 | FB |  | ei |  |
|  |  | Loop: |  |  |
| 2093 | FD |  | nop |  |
| 2094 | E7FCFF |  | ljmp | Loop |
|  |  | EXTINT: | |  |
| 2097 | F2 | pushf | |  |
| 2098 | 950130 | xorb | | P1Buf, #LED |
| 209B | B0300F | ldb | | P1, P1Buf |
| 209E | F3 | popf | |  |
| 209F | F0 | ret | |  |
|  |  | end | |  |

; 硬件实验十二 定时器实验 H12.ASM

Tick equ 100 ; 100 x 10ms = 1s T10ms equ 65536-7500 ; 10ms 时间常数(6M) C10ms equ 30h ; 10ms 记数单元

P1Buf equ 31h

LED equ 1 ; P1.0 P1 equ IOPORT1

org 2000h

2000 A720 dcw T1INT

org 2080h

Start:

2080 A1E00018 ld sp, #0e0h

2084 B00F31 ldb P1Buf, P1

2087 71FE31 andb P1Buf, #!LED

208A B0310F ldb P1, P1Buf

208D B10F14 ldb WSR, #15

2090 A1B4E20A ld T1, #T10ms

2094 B10014 ldb WSR, #0

2097 B16430 ldb C10ms, #Tick

209A B10416 ldb IOC1, #04 ; enable timer1 interrupt

209D B10108 ldb INT\_MASK, #01h ; enable timer1 interrupt

20A0 FB ei

Loop:

20A1 B0310F ldb P1, P1Buf

20A4 E7FAFF ljmp Loop

T1Int:

20A7 F2 pushf

20A8 B10F14 ldb WSR, #15

|  |  |  |  |
| --- | --- | --- | --- |
| 20AB A1B4E20A  20AF B10014 |  | ld ldb | T1, #T10ms WSR, #0 |
| 20B2 1530  20B4 D706  20B6 B16430  20B9 950131  20BC F3  20BD F0 | Exit: | decb jne ldb xorb  popf ret | C10ms Exit  C10ms, #Tick ; 10ms 记数器为 0, 重置记数器  P1Buf, #LED ; 取 反 LED |
|  |  | end |  |

; 硬件实验十三 D/A 转换实验 H13.ASM

CS0832 equ 08000h

|  |  |  |  |
| --- | --- | --- | --- |
|  | | org | 2080h |
| 2080 | B1001C | ldb | al, #0 |
| 2083 | C70100801C | stb | al, CS0832 |
| 2088 | B1401C | ldb | al, #40h |
| 208B | C70100801C | stb | al, CS0832 |
| 2090 | B1801C | ldb | al, #80h |
| 2093 | C70100801C | stb | al, CS0832 |
| 2098 | B1C01C | ldb | al, #0c0h |
| 209B | C70100801C | stb | al, CS0832 |
| 20A0 | B1FF1C | ldb | al, #0ffh |
| 20A3 | C70100801C | stb | al, CS0832 |
| 20A8 | E7FDFF | ljmp | $ |
|  |  | end |  |

; 硬件实验十四 A/D 转换实验 H14.ASM

mode equ 082h ; 方式 0，PA，PC 输出，PB 输入

PortA equ 8000h ; Port A CAddr equ 8003h ; 控制字地址CS0809 equ 9000h

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | org | 2080h |  |
| 2080 | A1E00018 |  | ld | sp, #0e0h |
| 2084 | B1821C |  | ldb | al, #mode |
| 2087  208C | C70103801C  B0001C | again: | stb  ldb | al,CAddr  al, 0 | ; 输出控制字 |
| 208F | C70100901C |  | stb | al, CS0809 | ; 起动 A/D |
| 2094 | B14020 |  | ldb | cl, #40h |  |
| 2097 | E020FD |  | djnz | cl, $ | ; 延时 > 100us |
| 209A | B30100901C |  | ldb | al, CS0809 | ; 读入结果 |
| 209F | C70100801C |  | stb | al,PortA | ; 输出 PortA |
| 20A4 | E7E5FF |  | ljmp end | again |  |

;硬件实验十五 外部中断实验（急救车与交通灯） H15.ASM

mode equ 082h ; 8255 方式 0，PA，PC 输出，PB 输入

LEDBuf equ 30h STOP equ 31h

LED equ 8000h

CAddr equ 8003h ; 控制字地址

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | SR SY SG  ER | equ 1 ; 南北红灯  equ 2 ; 南北黄灯  equ 4 ; 南北绿灯  equ 8 ; 东西红灯 |
| EY | equ 16 ; 东西黄灯 |
| EG | equ 32 ; 东西绿灯 |
| 200E | 2A21 |  | org dcw | 200eh EXTINT |
|  |  |  | org | 2080h |
| 2080 | A1E00018 |  | ld | sp, #0e0h |
| 2084 | B1821C |  | ldb | al, #mode |
| 2087 | C70103801C |  | stb | al,CAddr ; 输出控制字 |
| 208C  208F  2092 | 71FD16 B18008 FB |  | andb ldb ei | IOC1, #!02h  INT\_MASK, #80h ; enable extint |
| 2093 | B10930 |  | ldb | LEDBuf, #SR + ER ; 南北, 东西均红灯 |
| 2096  209B  209E  20A1  20A4 | C701008030 B10031  393160  B12130 C701008030 | Loop: | stb ldb  jbs ldb stb | LEDBuf, LED STOP, #0  STOP, 1, AllRed  LEDBuf, #SR + EG ; 南北红灯, 东西绿灯  LEDBuf, LED |
| 20A9  20AC | B1141D EF6600 |  | ldb call | ah, #20 Delay |
| 20AF  20B2  20B5 | 39314F B10130 C701008030 |  | jbs ldb stb | STOP, 1, AllRed  LEDBuf, #SR ; 南北红灯, 东西黄灯闪  LEDBuf, LED |
| 20BA  20BD | B10920  951030 | Loop1: | ldb  xorb | cl, #9  LEDBuf, #EY |
| 20C0  20C5  20C8  20CB | C701008030 B1011D EF4A00 E020EF |  | stb ldb call djnz | LEDBuf, LED  ah, #1 Delay cl, Loop1 |
| 20CE  20D1  20D4 | 393130  B10C30 C701008030 |  | jbs ldb stb | STOP, 1, AllRed  LEDBuf, #SG + ER ; 南北绿灯, 东西红灯  LEDBuf, LED |
| 20D9  20DC | B1141D EF3600 |  | ldb call | ah, #20 Delay |
| 20DF  20E2  20E5 | 39311F B10830 C701008030 |  | jbs ldb stb | STOP, 1, AllRed  LEDBuf, #ER ; 东西红灯, 南北黄灯闪  LEDBuf, LED |
| 20EA  20ED | B10920  950230 | Loop2: | ldb  xorb | cl, #9  LEDBuf, #SY |
| 20F0  20F5  20F8  20FB | C701008030 B1011D EF1A00  E020EF |  | stb ldb call  djnz | LEDBuf, LED  ah, #1 Delay cl, Loop2 |

|  |  |  |  |
| --- | --- | --- | --- |
| 20FE | E79DFF | ljmp | Loop |
|  |  | AllRed: |  |
| 2101 | B10930 | ldb | LEDBuf, #SR + ER ; 南北, 东西均红灯 |
| 2104 | C701008030 | stb | LEDBuf, LED |
| 2109 | B10031 | ldb | STOP, #0 |
| 210C | B10A1D | ldb | ah, #10 |
| 210F | EF0300 | call | Delay |
| 2112 | E789FF | ljmp | Loop |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Delay: ; 延时子程序 | |
| 2115 | B0001C | ldb al, 0 | |
| 2118 | C81E | push bx | |
|  |  | DelayLoop: | |
| 211A | A164001E | ld bx,#100 | |
|  |  | DelayWait: | |
| 211E | 393106 | jbs STOP,1, ExitDelay | |
| 2121 | E11EFA | djnzw bx, DelayWait | |
| 2124 | E11CF3 | djnzw ax, DelayLoop | |
|  |  | ExitDelay: | |
| 2127 | CC1E | pop bx | |
| 2129 | F0 | ret | |
|  |  | EXTINT: | |
| 212A | B1FF31 | ldb | STOP, #0ffh |
| 212D | F0 | ret |  |
|  |  | end |  |

; 硬件实验十六 八段数码管显示实验 H16.ASM

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | OUTBIT OUTSEG IN | equ equ equ | 08002h  08004h  08001h | ; 位控制口  ; 段控制口  ; 键盘读入口 | |
| LEDBuf Num DelayT | equ equ equ | 60h 70h 75h | ; 显示缓冲  ; 显示的数据 | |
| 2080 | E74E00 |  | org ljmp | 2080h  Start |  | |
| 2083 | 3F065B4F | LEDMAP: | | ; 八段管显示码 | | |
| 2087  208B  208F | 666D7D07  7F6F777C  395E7971 | db  db | | 3fh, 06h, 5bh, 4fh, 66h, 6dh, 7dh, 07h  7fh, 6fh, 77h, 7ch, 39h, 5eh, 79h, 71h | | |
| 2093 | C81C | Delay: push | | ax | | ; 延时子程序 |
| 2095  2098 | B1001C  E11CFD | ldb DelayLoop:  djnzw | | al, #0  ax, DelayLoop | |  |
| 209B  209D | CC1C F0 | pop ret | | ax | |  |

DisplayLED:

209E A1600022 ld dx, #LEDBuf

20A2 B10620 ldb cl, #6 ; 共 6 个八段管

20A5 B1201D ldb ah, #00100000b ; 从左边开始显示

DLoop:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 20A8 | C701028000 | stb | 0, OUTBIT | ; 关所有八段管 |
| 20AD | B2221C | ldb | al, [dx] |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 20B0 | C70104801C |  | stb | al,OUTSEG |  |
| 20B5 | C70102801D |  | stb | ah, OUTBIT | ; 显示一位八段管 |
| 20BA | C81C |  | push | ax |  |
| 20BC | B1011D |  | ldb | ah, #1 |  |
| 20BF | 2FD2 |  | call | Delay |  |
| 20C1 | CC1C |  | pop | ax |  |
| 20C3 | 18011D |  | shrb | ah, #1 |  |
| 20C6 | 0722 |  | inc | dx |  |
| 20C8 | E020DD |  | djnz | cl, DLoop |  |
| 20CB | C701028000 |  | stb | 0, OUTBIT | ; trun off all LED |
| 20D0 | F0 |  | ret |  |  |
|  |  | Start: |  |  |  |
| 20D1 | A1E00018 |  | ld | sp, #0e0h |  |
| 20D5 | B10070 |  | ldb | Num, #0 |  |
|  |  | MLoop: |  |  |  |
| 20D8 | 1770 |  | incb | Num |  |
| 20DA | B07021 |  | ldb | ch,Num |  |
| 20DD | B1001D |  | ldb | ah,#0 |  |
| 20E0 | B10620 |  | ldb | cl,#6 |  |
| 20E3 | A1600022 |  | ld | dx,#LEDBuf |  |
| FillBuf: | | | | | |
| 20E7 | B0211C | ldb | | al,ch | |
| 20EA | 710F1C | andb | | al,#0fh | |
| 20ED | B31D83201C | ldb | | al,LEDMap[ax] | |
| 20F2 | C6221C | stb | | al,[dx] | |
| 20F5 | 0722 | inc | | dx | |
| 20F7 | 1721 | incb | | ch | |
| 20F9 | E020EB | djnz | | cl, FillBuf | |
| 20FC | B13275 | ldb | | DelayT,#50 | |
|  |  | DispAgain: | |  | |
| 20FF | 2F9D | call | | DisplayLED ; 显示 | |
| 2101 | E075FB | djnz | | DelayT, DispAgain | |
| 2104 | E7D1FF | ljmp | | MLoop | |
|  |  | end | |  | |

; 硬件实验十七 键盘扫描显示实验 H17.ASM

OUTBIT equ 08002h ; 位控制口OUTSEG equ 08004h ; 段控制口 IN equ 08001h ; 键盘读入口

|  |  |  |  |
| --- | --- | --- | --- |
|  | LEDBuf equ | 60h | ; 显示缓冲 |
| 2080 E7D000 | org ljmp | 2080h  Start |  |
|  | LEDMAP: |  | ; 八段管显示码 |
| 2083 3F065B4F  2087 666D7D07  208B 7F6F777C  208F 395E7971 | db db | 3fh, 06h, 5bh, 4fh, 66h, 6dh, 7dh, 07h 7fh, 6fh, 77h, 7ch, 39h, 5eh, 79h, 71h | |

2093 C81C Delay: push ax ; 延时子程序

2095 B1001C ldb al, #0

DelayLoop:

2098 E11CFD djnzw ax, DelayLoop

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 209B  209D | CC1C  F0 | pop  ret | ax | | | | |
| 209E | A1600022 | DisplayLED:  ld | dx, #LEDBuf | | | |  |
| 20A2 | B10620 | ldb | cl, #6 | | | | ; 共 6 个八段管 |
| 20A5  20A8 | B1201D  C701028000 | ldb  DLoop:  stb | ah, #00100000b  0, OUTBIT | | | | ; 从左边开始显示  ; 关所有八段管 |
| 20AD  20B0  20B5 | B2221C C70104801C C70102801D | ldb stb stb | al, [dx] al, OUTSEG ah, OUTBIT | | | | ; 显示一位八段管 |
| 20BA  20BC  20BF  20C1 | C81C B1011D 2FD2 CC1C | push ldb call pop | ax  ah, #1 Delay ax | | | |  |
| 20C3 | 18011D | shrb | ah, #1 | | | |  |
| 20C6  20C8 | 0722  E020DD | inc djnz | dx  cl, DLoop | | | |  |
| 20CB  20D0 | C701028000 F0 | stb ret | 0, OUTBIT | | | | ; trun off all LED |
|  |  | TestKey: |  | | | |  |
| 20D1  20D6  20DB  20DD  20E0 | C701028000 B30101801C 121C  710F1C F0 | stb ldb notb andb ret |  | 0, OUTBIT  al, IN al  al, #0fh | | | ; 输出线置为 0  ; 读入键状态  ; 高四位不用 |
| 20E1 | 161514FF | KeyTable:  db | ; | 键码定义  16h, 15h, 14h, | | | 0ffh |
| 20E5  20E9  20ED  20F1  20F5 | 13121110  0D0C0B0A  0E030609  0F020508  00010407 | db db db db  db |  | 13h, 12h, 11h,  0dh, 0ch, 0bh, 0eh, 03h, 06h, 0fh, 02h, 05h,  00h, 01h, 04h, | | | 10h 0ah 09h 08h  07h |
| 20F9 | B12021 | GetKey:  ldb | ch, | | #00100000b | | |
| 20FC | B10620 | ldb  KLoop: | cl, | | #6 | | |
| 20FF  2101 | 1221  C701028021 | notb stb | ch ch, | | OUTBIT | ; | 找出键所在列 |
| 2106 | 1221 | notb | ch | |  |  |  |
| 2108 | 180121 | shrb | ch, | | #1 |  |  |
| 210B  2110 | B30101801C 121C | ldb notb | al, al | | IN |  |  |
| 2112 | 710F1C | andb | al, | | #0fh |  |  |

2115 D708 jne Goon\_ ; 该列有键入

2117 E020E5 djnz cl, KLoop

211A B1FF20 ldb cl, #0ffh ; 没有键按下, 返回 0ffh

211D 2013 sjmp Exit1 Goon\_:

211F 1520 decb cl

2121 190220 shlb cl, #2 ; 键值 = 列 X 4 + 行

2124 B10421 ldb ch, #4

LoopC:

2127 381C08 jbs al, 0, Exit1

212A 18011C shrb al, #1

212D 1720 incb cl

212F E021F5 djnz ch, LoopC Exit1:

2132 C701028000 stb 0, OUTBIT

2137 1620 extb cx

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 2139  213E | B321E1201C  B01C1E | ldb  ldb | | al, KeyTable[cx] ; 取出键码  bl, al | | |
|  |  | WaitRelease: | |  | |  |
| 2141 | C701028000 | stb | | 0, OUTBIT | | ; 等键释放 |
| 2146 | B10A1D | ldb | | ah, #10 | |  |
| 2149 | 2F48 | call | | Delay | |  |
| 214B | 2F84 | call | | TestKey | |  |
| 214D | D7F2 | jne | | WaitRelease | |  |
| 214F | B01E1C | ldb | | al, bl | |  |
| 2152 | F0 | ret | |  | |  |
|  |  | Start: | |  | |  |
| 2153 | A1E00018 | ld | | sp, #0e0h | |  |
| 2157 | B1FF60 | ldb | | LEDBuf, #0ffh | | ; 显示 8.8.8.8. |
| 215A | B1FF61 | ldb | | LEDBuf+1, #0ffh | |  |
| 215D | B1FF62 | ldb | | LEDBuf+2, #0ffh | |  |
| 2160 | B1FF63 | ldb | | LEDBuf+3, #0ffh | |  |
| 2163 | B10064 | ldb | | LEDBuf+4, #0 | |  |
| 2166 | B10065 | ldb | | LEDBuf+5, #0 | |  |
|  |  | MLoop: | |  | |  |
| 2169 | 2F33 | call | | DisplayLED | ; 显示 | |
| 216B | 2F64 | call | | TestKey | ; 有键入? | |
| 216D | DFFA | je | | MLoop | ; 无键入, 继续显示 | |
| 216F | 2F88 | call | | GetKey | ; 读入键码 | |
| 2171 | 710F1C | andb | | al, #0fh | ; 显示键码 | |
| 2174 | B1001D | ldb | | ah, #0 |  | |
| 2177 | B31D832065 | ldb | | LEDBuf+5, | LEDMap[ax] | |
| 217C | E7EAFF | ljmp | | MLoop |  | |
|  |  | end | |  |  | |
| ; 硬件实验十八 | | 电子时钟 | H18.ASM | | | |

org 2000h

2000 D120 dcw T1Int

org 2080h

2080 E78000 ljmp Start

Hour equ 40h Minute equ 41h Second equ 42h C10ms equ 43h

Tick equ 100

T10ms equ 65536-7500 /2

|  |  |  |
| --- | --- | --- |
| OUTBIT | equ 08002h | ; 位控制口 |
| OUTSEG | equ 08004h | ; 段控制口 |

LEDBuf equ 60h ; 显示缓冲

LEDMAP: ; 八段管显示码

2083 3F065B4F

2087 666D7D07 db 3fh, 06h, 5bh, 4fh, 66h, 6dh, 7dh, 07h 208B 7F6F777C

208F 395E7971 db 7fh, 6fh, 77h, 7ch, 39h, 5eh, 79h, 71h

2093 C81C Delay: push ax ; 延时子程序

2095 B1001C ldb al, #0

DelayLoop:

2098 E11CFD djnzw ax, DelayLoop

209B CC1C pop ax

209D F0 ret

DisplayLED:

209E A1600022 ld dx, #LEDBuf

20A2 B10620 ldb cl, #6 ; 共 6 个八段管

20A5 B1201D ldb ah, #00100000b ; 从左边开始显示

DLoop:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 20A8  20AD  20B0 | C701028000 B2221C  C70104801C | stb ldb  stb | 0, OUTBIT  al, [dx] al, OUTSEG | ; 关所有八段管  ; |
| 20B5 | C70102801D | stb | ah, OUTBIT | ; 显示一位八段管 |
| 20BA  20BC  20BF  20C1 | C81C B1011D 2FD2 CC1C | push ldb call pop | ax  ah, #1 Delay ax |  |
| 20C3 | 18011D | shrb | ah, #1 |  |
| 20C6  20C8 | 0722  E020DD | inc djnz | dx  cl, DLoop |  |
| 20CB  20D0 | C701028000 F0 | stb ret | 0, OUTBIT | ; trun off all LED |

; ==========================================================

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 20D1 | F2 | T1Int: | pushf |  |
| 20D2  20D5  20D9 | B10F14 A15AF10A B10014 |  | ldb ld ldb | wsr, #15 T1, #T10ms  wsr, #0 |
| 20DC  20DE | 1543  D721 |  | decb jne | C10ms Exit |
| 20E0 | B16443 |  | ldb | C10ms, #Tick |
| 20E3  20E5  20E8 | 1742  993C42 D717 |  | incb cmpb jne | Second Second, #60 Exit |
| 20EA | B10042 |  | ldb | Second, #0 |
| 20ED  20EF  20F2 | 1741  993C41 D70D |  | incb cmpb jne | Minute Minute, #60 Exit |
| 20F4 | B10041 |  | ldb | Minute, #0 |
| 20F7 | 1740 |  | incb | Hour |
| 20F9  20FC | 991840  D703 |  | cmpb jne | Hour, #24 Exit |
| 20FE  2101 | B10040  F3 | Exit: | ldb  popf | Hour, #0 |
| 2102 | F0 |  | ret |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2103 | B10F14 | Start: | ldb | WSR, #15 |  |
| 2106  210A | A15AF10A B10014 |  | ld ldb | T1, #T10ms WSR, #0 |
| 210D  2110 | B16443 B10416 |  | ldb ldb | C10ms, #Tick IOC1, #04 |
| 2113 | B10108 |  | ldb | INT\_MASK, #01h | ; enable timer1 interrupt |
| 2116 | FB |  | ei |  |  |
| 2117  211A  211D  2120  2123 | B10F40 B10841 B10042 B16443  B0401C | MLoop: | ldb ldb ldb ldb  ldb | Hour, #15  Minute, #8  Second, #0 C10ms, #Tick  al, Hour |  |
| 2126 | B1001D |  | ldb | ah, #0 |  |
| 2129  212C | B10A1E 9C1E1C |  | ldb divub | bl, #10 ax, bl |  |
| 212F | B01C1E |  | ldb | bl, al |  |
| 2132 | B1001F |  | ldb | bh, #0 |  |
| 2135 | B31F832060 |  | ldb | LEDBuf, LEDMAP[bx] |  |
| 213A  213D  2142 | B01D1E B31F832061 918061 |  | ldb ldb orb | bl, ah  LEDBuf+1, LEDMAP[bx] LEDBuf+1, #80h | ; add '.' |
| 2145 | B0411C |  | ldb | al, Minute |  |
| 2148  214B  214E | B1001D B10A1E 9C1E1C |  | ldb ldb divub | ah, #0  bl, #10 ax, bl |  |
| 2151 | B01C1E |  | ldb | bl, al |  |
| 2154 | B1001F |  | ldb | bh, #0 |  |
| 2157 | B31F832062 |  | ldb | LEDBuf+2, LEDMAP[bx] |  |
| 215C  215F  2164 | B01D1E B31F832063 918063 |  | ldb ldb orb | bl, ah  LEDBuf+3, LEDMAP[bx] LEDBuf+3, #80h | ; add '.' |
| 2167  216A  216D  2170 | B0421C B1001D B10A1E 9C1E1C |  | ldb ldb ldb divub | al, Second ah, #0  bl, #10 ax, bl |  |
| 2173 | B01C1E |  | ldb | bl, al |  |
| 2176 | B1001F |  | ldb | bh, #0 |  |
| 2179 | B31F832064 |  | ldb | LEDBuf+4, LEDMAP[bx] |  |
| 217E  2181 | B01D1E B31F832065 |  | ldb ldb | bl, ah  LEDBuf+5, LEDMAP[bx] |  |
| 2186 | 2F16 |  | call | DisplayLED |  |
| 2188 | E798FF |  | ljmp | MLoop |  |
|  |  |  | end |  |  |

; 硬件实验十九 单片机串行口通讯实验 H19.ASM

OUTBIT equ 08002h ; 位控制口

OUTSEG equ 08004h ; 段控制口

IN equ 08001h ; 键盘读入口

LEDBuf equ 60h ; 显示缓冲RCVBUF equ 70h ; SIO 接收缓冲HasRcv equ 71h ; 接收标志

org 200ch

200C 5321 dcw Sio\_int

org 2080h

2080 E7E300 ljmp Start

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | LEDMAP: |  | ; 八段 | 管显示码 |
| 2083 | 3F065B4F |  | |  |  |
| 2087 | 666D7D07 | db | | 3fh, 06h, 5bh, | 4fh, 66h, 6dh, 7dh, 07h |
| 208B | 7F6F777C |  | |  |  |
| 208F | 395E7971 | db | | 7fh, 6fh, 77h, | 7ch, 39h, 5eh, 79h, 71h |
| 2093 | C81C | Delay: push | | ax | ; 延时子程序 |
| 2095 | B1001C | ldb | | al, #0 |  |
|  |  | DelayLoop: | |  |  |
| 2098 | E11CFD | djnzw | | ax, DelayLoop |  |
| 209B | CC1C | pop | | ax |  |
| 209D | F0 | ret | |  |  |
|  |  | DisplayLED: | |  |  |
| 209E | A1600022 | ld | | dx, #LEDBuf |  |
| 20A2 | B10620 | ldb | | cl, #6 | ; 共 6 个八段管 |
| 20A5 | B1201D | ldb | | ah, #00100000b | ; 从左边开始显示 |
|  |  | DLoop: | |  |  |
| 20A8 | C701028000 | stb | | 0, OUTBIT | ; 关所有八段管 |
| 20AD | B2221C | ldb | | al, [dx] |  |
| 20B0 | C70104801C | stb | | al, OUTSEG |  |
| 20B5 | C70102801D | stb | | ah, OUTBIT | ; 显示一位八段管 |
| 20BA | C81C | push | | ax |  |
| 20BC | B1011D | ldb | | ah, #1 |  |
| 20BF | 2FD2 | call | | Delay |  |
| 20C1 | CC1C | pop | | ax |  |
| 20C3 | 18011D | shrb | | ah, #1 |  |
| 20C6 | 0722 | inc | | dx |  |
| 20C8 | E020DD | djnz | | cl, DLoop |  |
| 20CB | C701028000 | stb | | 0, OUTBIT | ; 关闭所有显示位 |
| 20D0 | F0 | ret | |  |  |
|  |  | TestKey: | |  |  |
| 20D1 | C701028000 | stb |  | 0, OUTBIT | ; 输出线置为 0 |
| 20D6 | B30101801C | ldb |  | al, IN | ; 读入键状态 |
| 20DB | 121C | notb |  | al |  |
| 20DD | 710F1C | andb |  | al, #0fh | ; 高四位不用 |
| 20E0 | F0 | ret |  |  |  |
|  |  | KeyTable: | ; | 键码定义 |  |
| 20E1 | 161514FF | db |  | 16h, 15h, 14h, | 0ffh |
| 20E5 | 13121110 | db |  | 13h, 12h, 11h, | 10h |
| 20E9 | 0D0C0B0A | db |  | 0dh, 0ch, 0bh, | 0ah |
| 20ED | 0E030609 | db |  | 0eh, 03h, 06h, | 09h |
| 20F1 | 0F020508 | db |  | 0fh, 02h, 05h, | 08h |
| 20F5 | 00010407 | db |  | 00h, 01h, 04h, | 07h |

GetKey:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 20F9 | B12021 | ldb | ch, #00100000b | |
| 20FC | B10620 | ldb | cl, #6 | |
|  | KLoop: | |  | |
| 20FF | 1221 | notb | ch | ; 找出键所在列 |
| 2101 | C701028021 | stb | ch, OUTBIT |  |
| 2106 | 1221 | notb | ch |  |
| 2108 | 180121 | shrb | ch, #1 |  |
| 210B | B30101801C | ldb | al, IN |  |
| 2110 | 121C | notb | al |  |
| 2112 | 710F1C | andb | al, #0fh |  |
| 2115 | D708 | jne | Goon\_ | ; 该列有键入 |
| 2117 | E020E5 | djnz | cl, KLoop |  |
| 211A | B1FF20 | ldb | cl, #0ffh | ; 没有键按下, 返回 0ffh |
| 211D | 2013 | sjmp | Exit1 |  |
| Goon\_: | | | | |
| 211F | 1520 | decb | cl |  |
| 2121 | 190220 | shlb | cl, #2 | ; 键值 = 列 X 4 + 行 |
| 2124 | B10421 | ldb | ch, #4 |  |

LoopC:

2127 381C08 jbs al, 0, Exit1

212A 18011C shrb al, #1

212D 1720 incb cl

212F E021F5 djnz ch, LoopC Exit1:

2132 C701028000 stb 0, OUTBIT

2137 1620 extb cx

2139 B321E1201C ldb al, KeyTable[cx] ; 取出键码

213E B01C1E ldb bl, al WaitRelease:

2141 C701028000 stb 0, OUTBIT ; 等键释放

2146 B10A1D ldb ah, #10

2149 2F48 call Delay

214B 2F84 call TestKey

214D D7F2 jne WaitRelease

214F B01E1C ldb al, bl

2152 F0 ret

Sio\_int:

2153 F2 pushf

2154 C81C push ax

2156 B0111C ldb al,sp\_stat

2159 361C06 jbc al,6,s\_ret ; 是发送中断，不读数

215C B00770 ldb RCVBUF,SBUF

215F B10171 ldb HasRcv,#1 s\_ret:

2162 CC1C pop ax

2164 F3 popf

2165 F0 ret

Start:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2166 | A1C00018 | ld | sp, #0c0h |  |
| 216A | B12016 | ldb | ioc1, #20h | ; P2.0 设为 TXD 功能 |
| 216D | B10911 | ldb | sp\_con, #09 | ; 串行模式 1，允许接收 |
| 2170 | B19B0E | ldb | baud\_rate,#9bh | ; 波特率参数 |
| 2173 | B1800E | ldb | baud\_rate,#80h | ; 2400bps @ 6MHz |
| 2176 | B10013 | ldb | 13h,#00 | ; 禁止 TI(2030H)和 RI(2032H)中断 |
| 2179 | B14008 | ldb | int\_mask,#40h | ; 允许串口中断(200CH) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 217C | B1FF60 |  | ldb | LEDBuf, #0ffh | ; | 显示 8.8.8.8. |
| 217F | B10B61 |  | ldb | LEDBuf+1, #0bh |  |  |
| 2182 | B10362 |  | ldb | LEDBuf+2, #03h |  |  |
| 2185 | B1FF63 |  | ldb | LEDBuf+3, #0ffh |  |  |
| 2188 | B10064 |  | ldb | LEDBuf+4, #0 |  |  |
| 218B | B10065 |  | ldb | LEDBuf+5, #0 |  |  |
| 218E | B10071 |  | ldb | HasRcv,#0 | ; |  |
| 2191 | FB |  | ei |  |  |  |
|  |  | MLoop: |  |  |  |  |
| 2192 | 38710E |  | jbs | HasRcv,0,RcvData | ; | 是否收到数据？ |
| 2195 | 2F07 |  | call | DisplayLED | ; | 显示 |
| 2197 | 2F38 |  | call | TestKey | ; | 有键入? |
| 2199 | DFF7 |  | je | MLoop | ; | 无键入, 继续显示 |
| 219B | 2F5C |  | call | GetKey | ; | 读入键码 |
| 219D | B01C07 |  | ldb | SBUF,al | ; | 通讯口输出键码 |
| 21A0 | E7EFFF |  | ljmp | MLoop |  |  |
| RcvData: | | | | | | |
| 21A3 | B10071 | ldb | | HasRcv,#0 | ; | 已收到数据 |
| 21A6 | B0701C | ldb | | al,RcvBuf |  |  |
| 21A9 | 710F1C | andb | | al,#0fh | ; | 显示低 4 位 |
| 21AC | B1001D | ldb | | ah,#0 |  |  |

21AF B31D832065 ldb LEDBuf+5,LEDMap[ax]

21B4 B0701C ldb al,RcvBuf

21B7 18041C shrb al,#4 ; 显示高 4 位

21BA B31D832064 ldb LEDBuf+4,LEDMap[ax]

21BF E7D0FF ljmp MLoop end

; 硬件实验二十 打印机控制试验 H20.ASM

; printer

; ASM for MCS96

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | mode equ | 082h ; 8255 工作方式, PA,PC 输出,PB 输入 | |
| dport equ | 08000h ; 打印数据，8255 PA 口地址 | |
| status equ | 08001h ; 忙状态，8255 PB 口地址 | |
| ctl equ | 08002h ; 选通控制，8255 PC 口地址 | |
| contrl equ | 08003h ; 8255 控制口地址 | |
| busy equ | 0 ; 忙状态位 | |
| org | 2080h | |
| 2080 | E72100 | jmp | start | |
|  |  | print: |  | |
| 2083 | B01C1D | ldb | ah,al | |
|  |  | rd\_status: |  | |
| 2086 | B30101801C | ldb | al, status ; 读打印机状态 | |
| 208B | 381CF8 | jbs | al, busy, rd\_status ; 若忙，等待 | |
| 208E | C70100801D | stb | ah, dport ; 数据送出 | |
| 2093 | B1011C | ldb | al,#1 | |
| 2096 | C70102801C | stb | al, ctl ; 输出选通脉冲 | |
| 209B | B0001C | ldb | al,0 | |
| 209E | C70102801C | stb | al, ctl | |
| 20A3 | F0 | ret |  | |
|  |  | start: |  | |
| 20A4 | B1821C | ldb | al,#mode |  |
| 20A7 | C70103801C | stb | al,contrl | ; 输出 8255 控制字 |
| 20AC | A1BF201E | ld | bx, #string | ; 被打印字串 |

nextchar:

20B0 B21F1C ldb al,[bx]+

20B3 98001C cmpb al,0

20B6 D702 jne prtchar ; 字串结束

20B8 27FE jmp $

prtchar:

20BA 2FC7 call print

20BC 27F2 jmp nextchar

20BE FD nop

string:

20BF 48656C6C

20C3 6F20576F

20C7 726C6421

20CB 0D0A0C00

20CF 00 db 'Hello World!',0dh,0ah,0ch,0,0 end

|  |  |  |
| --- | --- | --- |
| ; 硬件实验二十一 | 直流电机控制实验 H21.ASM  mode equ 082h |  |
|  | STATUS equ 08001h PORTA equ 08000h  CTL equ 08003h |
|  | CS0832 equ 09000h |
|  | DC\_P equ 0  count equ 60 |
| 2080 E74400  2083 B10F1C | org 2080h  jmp start  delay:  ldb al,#15 |
| 2086 A100001E  208A E11EFD | ld bx,#0  ddd:  djnzw bx,ddd |
| 208D E01CFA  2090 F0  2091 B30101801C  2096 301CF8  2099 B30101801C  209E 381CF8 | djnz al,ddd ret  read:  r\_0: ldb al,STATUS  jbc al, DC\_P, r\_0 r\_1: ldb al,STATUS  jbs al, DC\_P, r\_1 | ; 等待低电平完  ; 等待高电平完 |
| 20A1 B10F14  20A4 A100000A  20A8 B10014 | ldb WSR, #15  ld T1, #0  ldb WSR, #0 |  |
| 20AB B30101801C  20B0 301CF8  20B3 B30101801C  20B8 381CF8  20BB A00A3C  20BE A03C1C  20C1 C70100801D  20C6 F0  20C7 B1821C  20CA C70103801C  20CF B1FF1C  20D2 C70100901C  20D7 2FAA | r\_2: ldb al,STATUS  jbc al, DC\_P, r\_2 r\_3: ldb al,STATUS  jbs al, DC\_P, r\_3 ld count,T1  ld ax,count  stb ah,PortA ret  start:  ldb al, #mode  stb al, CTL  loop:  ldb al, #0ffh stb al, CS0832  call delay | ; 等待低电平完  ; 等待高电平完  ; 等待电机运转稳定 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 20D9  20DB  20DE  20E3 | 2FB6  B1D01C C70100901C 2F9E | call  ldb stb call | read  al, #0d0h al, CS0832  delay | ;  ;  ; | 读取时间  设断点，观察上次时间等待电机运转稳定 |
| 20E5 | 2FAA | call | read |  |  |
| 20E7  20EA  20EF | B1801C C70100901C 2F92 | ldb stb call | al, #080h al, CS0832  delay | ;  ; | 设断点，观察上次时间  等待电机运转稳定 |
| 20F1 | B1201C | ldb | al, #020h | ; | 设断点，观察上次时间 |
| 20F4 | C70100901C | stb | al, CS0832 | ; | 电机反转 |
| 20F9 | 2F88 | call | delay | ; | 等待电机运转稳定 |
| 20FB | 2F94 | call | read | ; | 读取时间 |
| 20FD | B1001C | ldb | al, #00h | ; | 设断点，观察上次时间 |
| 2100 | C70100901C | stb | al, CS0832 | ; | 电机反转加速 |
| 2105 | 2F7C | call | delay | ; | 等待电机运转稳定 |
| 2107 | 2F88 | call | read |  |  |
| 2109  210C  2111 | B1201C C70100901C  2F70 | ldb stb  call | al, #020h al, CS0832  delay | ; | 等待电机运转稳定 |
| 2113 | 2F7C | call | read | ; | 读取时间 |
| 2115 | B1D01C | ldb | al, #0d0h | ; | 设断点，观察上次时间 |
| 2118  211D  211F | C70100901C 2F64  2F70 | stb call call | al, CS0832  delay read | ; | 等待电机运转稳定 |
| 2121 | E7ABFF | ljmp | loop | ; | 设断点，观察上次时间 |
|  |  | end |  |  |  |

; 硬件实验二十二 步进电机控制实验 H22.ASM

mode equ 082h ctl equ 08000h contrl equ 08003h

Astep equ 01h Bstep equ 02h Cstep equ 04h Dstep equ 08h

dly\_c equ 60h

org 2080h

step:

2080 B1821C ldb al, #mode

2083 C70103801C stb al, contrl

2088 C701008000 stb 0, ctl

208D B11060 ldb dly\_c,#10h

2090 E70000 jmp loop

; 单/双八拍工作方式

loop:

2093 B1011C ldb al,#Astep

2096 C70100801C stb al,ctl

209B EFD900 call delay

209E B1031C ldb al,#Astep+Bstep

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 20A1 | C70100801C |  | stb | al,ctl |  |
| 20A6 | EFCE00 |  | call | delay |
| 20A9 | B1021C |  | ldb | al,#Bstep |
| 20AC | C70100801C |  | stb | al,ctl |
| 20B1 | EFC300 |  | call | delay |
| 20B4 | B1061C |  | ldb | al,#Bstep+Cstep |
| 20B7 | C70100801C |  | stb | al,ctl |
| 20BC | EFB800 |  | call | delay |
| 20BF | B1041C |  | ldb | al,#Cstep |
| 20C2 | C70100801C |  | stb | al,ctl |
| 20C7 | EFAD00 |  | call | delay |
| 20CA | B10C1C |  | ldb | al,#Cstep+Dstep |
| 20CD | C70100801C |  | stb | al,ctl |
| 20D2 | EFA200 |  | call | delay |
| 20D5 | B1081C |  | ldb | al,#Dstep |
| 20D8 | C70100801C |  | stb | al,ctl |
| 20DD | EF9700 |  | call | delay |
| 20E0 | B1091C |  | ldb | al,#Dstep+Astep |
| 20E3 | C70100801C |  | stb | al,ctl |
| 20E8 | EF8C00 |  | call | delay |
| 20EB | B0601C |  | ldb | al, dly\_c |
| 20EE | 151C |  | decb | al | ; 提高转速 |
| 20F0 | 99011C |  | cmpb | al,#1 |  |
| 20F3 | D702 |  | jne | nn1 | ; 最快速度 |
| 20F5 | 171C |  | incb | al |  |
| 20F7 | B01C60 | nn1: | ldb | dly\_c,al |  |
| 20FA | E796FF |  | ljmp | Loop |  |

; 双四拍工作方式

loop1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 20FD | B1031C |  | ldb | al, #Astep+Bstep |
| 2100 | C70100801C |  | stb | al,ctl |
| 2105 | EF6F00 |  | call | delay |
| 2108  210B  2110 | B1061C C70100801C  EF6400 |  | ldb stb  call | al, #Bstep+Cstep al,ctl  delay |
| 2113 | B10C1C |  | ldb | al, #Cstep+Dstep |
| 2116  211B  211E  2121 | C70100801C EF5900 B1091C  C70100801C |  | stb call ldb  stb | al,ctl delay  al, #Dstep+Astep  al,ctl |
| 2126 | EF4E00 |  | call | delay |
| 2129  212C  212E  2131 | B0601C 151C  99021C  D702 |  | ldb decb cmpb  jne | al, dly\_c al  al, #2  nn2 |
| 2133 | 171C |  | incb | al |
| 2135 | B01C60 | nn2: | ldb | dly\_c,al |
| 2138 | 27C3 |  | jmp | loop1 |

; 单四拍工作方式

loop2:

213A B1081C ldb al,#Dstep

213D C70100801C stb al,ctl

2142 EF3200 call delay

2145 B1041C ldb al,#Cstep

2148 C70100801C stb al,ctl

214D EF2700 call delay

2150 B1021C ldb al,#Bstep

2153 C70100801C stb al,ctl

2158 EF1C00 call delay

215B B1011C ldb al,#Astep

|  |  |  |  |
| --- | --- | --- | --- |
| 215E  2163 | C70100801C EF1100 |  | stb al,ctl call delay |
| 2166 | B0601C |  | ldb al, dly\_c |
| 2169  216B  216E  2170 | 151C  99031C D702 171C |  | decb al cmpb al, #3 jne nn3  incb al |
| 2172 | B01C60 | nn3: | ldb dly\_c,al |
| 2175  2177 | 27C3  B0601D | delay: | jmp loop2  ldb ah,dly\_c |
| 217A  217D | B0001C  E11CFD | dd1: | ldb al,0  djnzw ax,dd1 |
| 2180 | F0 |  | ret end |

; 硬件实验二十三 温度传感器试验 H23.ASM

OUTBIT equ 08002h ; 位控制口OUTSEG equ 08004h ; 段控制口 IN equ 08001h ; 键盘读入口

LEDBuf equ 60h ; 显示缓冲

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2080 | E7B800 | org  ljmp | 2080h  Start | | |
| 2083 | 3F065B4F | LEDMAP: | ; 八段管显示码 | | |
| 2087  208B  208F | 666D7D07  7F6F777C  395E7971 | db  db | 3fh, 06h, 5bh, 4fh, 66h, 6dh, 7dh, 07h  7fh, 6fh, 77h, 7ch, 39h, 5eh, 79h, 71h | | |
| 2093 | C81C | Delay: push | ax | ; 延时子程序 | |
| 2095 | B0001C | ldb DelayLoop: | al, 0 |  | |
| 2098 | E11CFD | djnzw | ax, DelayLoop | | |
| 209B  209D | CC1C F0 | pop ret | ax | | |
| 209E | A1600022 | DisplayLED:  ld | dx, #LEDBuf | | |
| 20A2 | B10620 | ldb | cl, #6 ; 共 6 个八段管 | | |
| 20A5 | B1201D | ldb  DLoop: | ah, #00100000b ; 从左边开始显示 | | |
| 20A8  20AD  20B0  20B5 | C701028000 B2221C C70104801C C70102801D | stb ldb stb stb | 0, OUTBIT  al, [dx] al, OUTSEG ah, OUTBIT | | ; 关所有八段管  ; 显示一位八段管 |
| 20BA  20BC  20BF  20C1 | C81C B1011D 2FD2 CC1C | push ldb call pop | ax  ah, #1 Delay ax | |  |
| 20C3 | 18011D | shrb | ah, #1 | |  |
| 20C6  20C8 | 0722  E020DD | inc djnz | dx  cl, DLoop | |  |
| 20CB | C701028000 | stb | 0, OUTBIT | | ; trun off all LED |

20D0 F0 ret

; ============================================================

|  |  |  |
| --- | --- | --- |
| LowTemp | equ -99 | ; A/D 0 |
| HighTemp  ADPort CurTemp | equ 99  equ 0a000h equ 50h | ; A/D 255 |

DisplayResult:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 20D1  20D4 | B0501C 371C0A |  | ldb jbc | al, CurTemp al, 7, GE0 |  |
| 20D7  20DA  20DC  20DE  20E1 | B14060 151C  121C E70300  B10060 | GE0: | ldb decb notb jmp  ldb | LEDBuf, #40h  al al Goon  LEDBuf, #0 ; | ; '-'  ' ' |
| 20E4 | B10A20 | Goon: | ldb | cl, #10 | |
| 20E7  20EA | B0001D FE9C201C |  | ldb divb | ah, 0 ax, cl | |
| 20EE  20F1  20F4 | B01C1E B0001F B31F832061 |  | ldb ldb ldb | bl, al bh, 0  LEDBuf+1, LEDMAP[bx] | |
| 20F9  20FC | B01D1E B31F832062 |  | ldb ldb | bl, ah  LEDBuf+2, LEDMAP[bx] | |
| 2101 | B10063 |  | ldb | LEDBuf+3, #0 ; ' ' | |

2104 F0 ret

ReadAD:

2105 C70100A000 stb 0, ADPort

210A B00022 ldb dl, 0

210D E022FD djnz dl, $ ; delay

2110 B30100A022 ldb dl, ADPort

2115 B1FF1C ldb al, #0ffh

2118 68221C sub al, dl

211B F0 ret

ReadTemp:

211C A0001E ld bx, 0

211F B11020 ldb cl, #16

RLoop:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2122 | 2FE1 |  | call | ReadAD |
| 2124 | B0001D |  | ldb | ah, 0 |
| 2127  212A | 641C1E E020F5 |  | add djnz | bx, ax cl, RLoop |
| 212D | 08041E |  | shr | bx, #4 |
| 2130 | 7DC61E |  | mulub | bl, #(HighTemp-LowTemp) |
| 2133 | B01F1C |  | ldb | al, bh ; /256 |
| 2136  213A  213B | 559D1C50 F0  A1E00018 | Start: | addb ret  ld | CurTemp, al, #LowTemp  sp, #0e0h |

213F B10060 ldb LEDBuf ,#0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2142 | B10061 |  | ldb | LEDBuf+1 ,#0 |
| 2145 | B10062 |  | ldb | LEDBuf+2 ,#0 |
| 2148 | B10063 |  | ldb | LEDBuf+3 ,#0 |
| 214B | B10064 |  | ldb | LEDBuf+4 ,#0 |
| 214E | B10065 |  | ldb | LEDBuf+5 ,#0 |
|  |  | MLoop: |  |  |
| 2151 | 2F4B |  | call | DisplayLED |
| 2153 | 2FC7 |  | call | ReadTemp |
| 2155 | 2F7A |  | call | DisplayResult |
| 2157 | 27F8 |  | sjmp  end | MLoop |

; 硬件实验二十四 液晶显示控制电路 H24.ASM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | PD1 | EQU | 3DH | ;122/2 分成左右两半屏 122x32 |
| R3 | EQU | 28H |  |
| R4 | EQU | 29H |  |
| R5 | EQU | 2AH |  |
| R6 | EQU | 2BH |  |
| COLUMN | EQU | 30H |  |
| PAGE\_ | EQU | 31H | ;页地址寄存器 D1,DO:页地址 |
| CODE\_ | EQU | 32H | ;字符代码寄存器 |
| COUNT | EQU | 33H | ;计数器 |
| DIR COL1 COL2  dtp3 CTEMP | equ equ equ equ EQU | 34h 35h 36h 37h 38H |  |
| COM | EQU | 39H | ;指令寄存器 |
| DAT | EQU | 3AH | ;数据寄存器 |
| CWADD1 | EQU | 08000H | ;写指令代码地址（E1） |
| DWADD1 | EQU | 08001H | ;写显示数据地址（E1） |
| CRADD1 | EQU | 08002H | ;读状态字地址（E1） |
| DRADD1 | EQU | 08003H | ;读显示数据地址（E1） |
| CWADD2 | EQU | 08004H | ;写指令代码地址（E2） |
| DWADD2 | EQU | 08005H | ;写显示数进地址（E2） |
| CRADD2 | EQU | 08006H | ;读状态字地址（E2） |
| DRADD2 | EQU | 08007H | ;读显示数据地址（E2） |
|  |  |  | ORG | 2080H |  |
| 2080 | E7FD01 |  | JMP | MAIN |  |

;

; 初始化程序

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2083 | B1E239 | INIT: | LDB | COM, #0E2H | ;复位 |
| 2086 | EF6300 |  | LCALL | PRO |  |
| 2089 | EF9300 |  | LCALL | PR3 |  |
| 208C | B1A439 |  | LDB | COM, #0A4H | ;关闭休闭状态 |
| 208F | EF5A00 |  | LCALL | PRO |  |
| 2092 | EF8A00 |  | LCALL | PR3 |  |
| 2095 | B1A939 |  | LDB | COM, #0A9H | ;设置 1／32 占空比 |
| 2098 | EF5100 |  | LCALL | PRO |  |
| 209B | EF8100 |  | LCALL | PR3 |  |
| 209E | B1A039 |  | LDB | COM, #0A0H | ;正向排序设置 |
| 20A1 | EF4800 |  | LCALL | PRO |  |
| 20A4 | EF7800 |  | LCALL | PR3 |  |

20A7 B1C039 LDB COM, #0C0H ;设置显示起始行为第一行

20AA EF3F00 LCALL PRO

20AD EF6F00 LCALL PR3

20B0 B1AF39 LDB COM, #0AFH ;开显示设置

20B3 EF3600 LCALL PRO

20B6 EF6600 LCALL PR3

20B9 F0 RET

;

; 清屏

20BA B10029 CLEAR: LDB R4,#00H ;页面地址暂存器设置

20BD B0291C CLEAR1: LDB AL,R4 ;取页地址值

20C0 91B81C ORB AL,#0B8H ;"或"页面地址设置代码

20C3 B01C39 LDB COM,AL ;页面地址设置

20C6 EF2300 LCALL PRO

20C9 EF5300 LCALL PR3

20CC B10039 LDB COM,#00H ;列地址设置为"0"

20CF EF1A00 LCALL PRO

20D2 EF4A00 LCALL PR3

20D5 B15028 LDB R3,#50H ;一页清 80 个字节

20D8 B1003A CLEAR2: LDB DAT,#00H ;显示数据为"0"

20DB EF1F00 LCALL PR1

20DE EF4F00 LCALL PR4

20E1 E028F4 DJNZ R3,CLEAR2 ;页内字节清零循环

20E4 1729 INCB R4 ;页地址暂存器加一

20E6 990429 CMPB R4,#04

20E9 D7D2 JNE CLEAR1 ;RAM 区清零循环

20EB F0 RET

;

;1．写指令代码子程序（E1） PRO:

20EC B30102801C PR01: LDB AL, CRADD1 ;读状态字

20F1 3F1CF8 JBS AL,7,PR01 ;判"忙"标志为句"0",否再读

20F4 B0391C LDB AL,COM ;取指令代码

20F7 C70100801C STB AL, CWADD1 ;写指令代码

20FC F0 RET

;

;2.写显示数据子程序（E1） PR1:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 20FD | B30102801C | PR11: | LDB | AL,CRADD1 | ;读状态宇 |
| 2102 | 3F1CF8 |  | JBS | AL,7,PR11 | ;判"忙"标志为"0",否再读 |
| 2105 | B03A1C |  | LDB | AL,DAT | ;取数据 |
| 2108 | C70101801C |  | STB | AL,DWADD1 | ;写数据 |
| 210D | F0 |  | RET |  |  |

;

;3.读显示数据子程序（E1） PR2:

210E B30102801C PR21: LDB AL,CRADD1 ;读状态字

2113 3F1CF8 JBS AL,7,PR21 ;判"忙"标志为"0"否,否再读

2116 B30103801C LDB AL,DRADD1 ;读数据

211B B01C3A LDB DAT,AL ;存数据

211E F0 RET

;

;4.写指令代码子程序（E2） PR3:

211F B30106801C PR31: LDB AL,CRADD2 ;读状态字

2124 3F1CF8 JBS AL, 7,PR31 ;判"忙"陈志为"0"否,否再读

2127 B0391C LDB AL,COM ;取指令代码

212A C70104801C STB AL,CWADD2 ;写指令代码

212F F0 RET

;

; 5.写显示数据子程序（E2） PR4:

2130 B30106801C PR41: LDB AL,CRADD2 ;读状态字

;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2135 | 3F1CF8 | JBS | AL,7,PR41 | ;判"忙"标志为"0"否,否再读 |
| 2138 | B03A1C | LDB | AL,DAT | ;取数据 |
| 213B | C70105801C | STB | AL,DWADD2 | ;写数据 |
| 2140 | F0 | RET |  |  |

; 6.读显示数据子程序（E2） PR5:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2141 | B30106801C | PR51: | LDB | AL,CRADD2 | ;读状态字 |
| 2146 | 3F1CF8 |  | JBS | AL, 7,PR51 | ;判"忙"标志为"0",否再读 |
| 2149 | B30107801C |  | LDB | AL,DRADD2 | ;读数据 |
| 214E | B01C3A |  | LDB | DAT,AL | ;存数据 |
| 2151 | F0 |  | RET |  |  |

; 中文显示子程序

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2152 | A100221E | CCW\_PR: | LD | BX,#CCTAB | ;确定字符字模块首地址 |
| 2156 | B0321C |  | LDB | AL,CODE\_ | ;取代码 |
| 2159 | B0001D |  | LDB | AH,0 |  |
| 215C | 7D201C |  | MULUB | AX,#20H | ;代码×32 |
| 215F | 641C1E |  | ADD | BX,AX | ;字模库首地址＋代码×32 |
| 2162 | B03036 |  | LDB | COL2, COLUMN | ;列地址 |
| 2165 | B10035 |  | LDB | COL1,#00H | ;代码寄存器借用为间址寄存器 |
| 2168 | B11033 | CCW\_1: | LDB | COUNT,#10H | ;计数器设置为 16 |
| 216B | B0311C |  | LDB | AL,PAGE\_ | ;读页地址寄存器 |
| 216E | 71031C |  | ANDB | AL,#03H | ;取页地址有效值 |
| 2171 | 91B81C |  | ORB | AL,#0B8H | ;"或"页地址设置代码 |
| 2174 | B01C39 |  | LDB | COM,AL | ;设置页地址 |
| 2177 | EF72FF |  | LCALL | PRO |  |
| 217A | EFA2FF |  | LCALL | PR3 |  |
| 217D | B03630 |  | LDB | COLUMN,COL2 | ;取列地址值 |
| 2180 | B0301C |  | LDB | AL,COLUMN | ;读列地址寄存器 |
| 2183 | 993D1C |  | CMPB | AL,#PD1 |  |
| 2186 | DE0F |  | JLT | CCW\_2 | ;＜0 为左半屏显示区域（E1） |
| 2188 | 793D1C |  | SUBB | AL,#PD1 |  |
| 218B | B01C30 |  | LDB | COLUMN,AL | ;≥0 为右半屏显示区域（E2） |
| 218E  2191 | B0311C  91081C |  | LDB  ;SETB ORB | AL,PAGE\_ ACC.3  AL,#08 | ;设置区域标志位。 |
| 2194 | B01C31 |  | LDB | PAGE\_,AL | ;"0"为 E1,"1"为 E2 |
| 2197 | B03039 | CCW\_2: | LDB | COM,COLUMN | ;设置列地址值 |
| 219A | B0311C |  | LDB | AL,PAGE\_ | ;判区域标志以确定设置哪个控制器 |
| 219D | 331C06 |  | JBC | AL,3,CCW\_3 |  |
| 21A0 | EF7CFF |  | LCALL | PR3 | ;区域 E2 |
| 21A3 | E70300 |  | LJMP | CCW\_4 |  |
| 21A6 | EF43FF | CCW\_3: | LCALL | PRO | ;区域 E1 |
| 21A9 | B0351C | CCW\_4: | LDB | AL,COL1 | ;取间址寄存器值 |
| 21AC | B0001D |  | LDB | AH,0 |  |
| 21AF | 641E1C |  | ADD | AX,BX |  |
| 21B2 | B21C1C |  | LDB | AL,[AX] | ;取汉字字模数据 |
| 21B5 | B01C3A |  | LDB | DAT,AL | ;写数据 |
| 21B8 | B0311C |  | LDB | AL,PAGE\_ |  |
| 21BB | 331C06 |  | JBC | AL,3,CCW\_5 |  |
| 21BE | EF6FFF |  | LCALL | PR4 | ;区域 E2 |
| 21C1 | E70300 |  | LJMP | CCW\_6 |  |
| 21C4 | EF36FF | CCW\_5: | LCALL | PR1 | ;区域 E1 |
| 21C7 | 1735 | CCW\_6: | INCB | COL1 | ;间址寄存器加一 |
| 21C9 | 1730 |  | INCB | COLUMN | ;列地址寄存器加一 |
| 21CB | B0301C |  | LDB | AL,COLUMN | ;判列地址是否超出区域范围、 |
| 21CE | 993D1C |  | CMPB | AL,#PD1 |  |
| 21D1 | DE12 | CCW\_7: | JLT | CCW\_8 | ;未超出则继续 |
| 21D3 | B0311C |  | LDB | AL,PAGE\_ | ;超出则判是否在区域 E2 |
| 21D6 | 3B1C0C |  | JBS | AL, 3,CCW\_8 | ;在区域 E2 则退出 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | ;SETB | ACC.3 | ;在区域 E1 则修改成区域 E2 |
| 21D9 | 91081C |  | ORB | AL,#08 |  |
| 21DC | B01C31 |  | LDB | PAGE\_,AL |  |
| 21DF | B10039 |  | LDB | COM,#00H | ;设置区域 E2 列地址为"0" |
| 21E2 | EF3AFF |  | LCALL | PR3 |  |
| 21E5 | E033C1 | CCW\_8: | DJNZ | COUNT,CCW\_4 | ;当页循环 |
| 21E8 | B0311C |  | LDB | AL,PAGE\_ | ;读页地址寄存器 |
| 21EB | 3F1C11 |  | JBS | AL,7,CCW\_9 | ;判完成标志 D7 位,"1"则完成退出 |
| 21EE | 171C |  | INCB | AL | ;否则页地址加一 |
|  |  |  | ;SETB | ACC.7 | ;置完成位为"1" |
| 21F0 | 91801C |  | ORB | AL,#80H |  |
| 21F3 | 71F71C |  | ANDB | AL,#0F7H | ; CLR ACC.3 |
| 21F6 | B01C31 |  | LDB | PAGE\_,AL |  |
| 21F9 | B11032 |  | LDB | CODE\_,#10H | ;间址寄存器设置为 16 |
| 21FC | E769FF |  | LJMP | CCW\_1 | ;大循环 |
| 21FF | F0 | CCW\_9: | RET |  |  |

;

;中文字符库CCTAB:

small0:

|  |  |  |  |
| --- | --- | --- | --- |
|  | | ;db  small1: | 38h,44h,44h,44h,44h,44h,38h,00h ;0 |
| ;db small2:  ;db small3:  ;db | 10h,30h,50h,10h,10h,10h,7ch,00h ;1  38h,44h,04h,08h,10h,20h,7ch,00h ;2  78h,84h,04h,38h,04h,84h,78h,00h ;3 |
| 2200 | 0004E424 |  |  |
| 2204 | 2464B42F |  | DB 000H,004H,0e4H,024H,024H,064H,0b4H,02fH ; 南 |
| 2208  220C  2210 | 24A46424  24E60400  00007F04 |  | DB 024H,0a4H,064H,024H,024H,0e6H,004H,000H |
| 2214 | 0505057F |  | DB 000H,000H,07fH,004H,005H,005H,005H,07fH |
| 2218  221C | 05050525  443F0000 |  | DB 005H,005H,005H,025H,044H,03fH,000H,000H |
| 2220 | 000404E4 |  |  |
| 2224 | 24242526 |  | DB 000H,004H,004H,0e4H,024H,024H,025H,026H ; 京 |
| 2228  222C | 242424E4  06040000 |  | DB 024H,024H,024H,0e4H,006H,004H,000H,000H |
| 2230 | 00201019 |  |  |
| 2234 | 0D41817F |  | DB 000H,020H,010H,019H,00DH,041H,081H,07fH |
| 2238  223C | 0101050D  38100000 |  | DB 001H,001H,005H,00dH,038H,010H,000H,000H |
| 2240 | 008060F8 |  |  |
| 2244 | 070A2828 |  | DB 000H,080H,060H,0f8H,007H,00aH,028H,028H ; 伟 |
| 2248  224C  2250 | 28FF2828  28280000  010000FF |  | DB 028H,0ffH,028H,028H,028H,028H,000H,000H |
| 2254 | 00010101 |  | DB 001H,000H,000H,0fFH,000H,001H,001H,001H |
| 2258  225C | 01FF0121  413F0000 |  | DB 001H,0ffH,001H,021H,041H,03fH,000H,000H |
| 2260 | 080889CE |  |  |
| 2264 | 2818027A |  | DB 008H,008H,089H,0ceH,028H,018H,002H,07aH ; 福 |
| 2268  226C  2270 | 4A4A4A4A  4A7A0200  020100FF |  | DB 04aH,04aH,04aH,04aH,04aH,07aH,002H,000H |
| 2274 | 0102FF49 |  | DB 002H,001H,000H,0ffH,001H,002H,0ffH,049H |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2278  227C | 49497F49  4949FF00 | DB 049H,049H,07fH,049H,049H,049H,0ffH,000H  ; .  ; 中文演示显示程序段 | | |
| 2280 | B1E018 | MAIN: LDB SP,#0E0H | | |
| 2283 | EFFDFD | LCALL INIT | | |
| 2286 | EF31FE | LCALL CLEAR | | |
| 2289 | B10038 | LDB CTEMP,#0 | | |
| 228C  228F | B10034  B10231 | LDB DIR,#0  AAA:  LDB PAGE\_,#02H | | |
| 2292 | B03830 | LDB COLUMN,CTEMP | | |
| 2295 | B10032 | LDB CODE\_,#00H | | |
| 2298 | EFB7FE | LCALL CCW\_PR | | |
| 229B | B10231 | LDB PAGE\_,#02H | | |
| 229E | B0381C | LDB AL,CTEMP | | |
| 22A1 | 75101C | ADDB AL,#10H | | |
| 22A4 | B01C30 | LDB COLUMN,AL | | |
| 22A7 | B10132 | LDB CODE\_,#01H | | |
| 22AA | EFA5FE | LCALL CCW\_PR | | |
| 22AD | B10231 | LDB PAGE\_,#02H | | |
| 22B0 | B0381C | LDB AL,CTEMP | | |
| 22B3 | 75201C | ADDB AL,#20H | | |
| 22B6 | B01C30 | LDB COLUMN,AL | | |
| 22B9 | B10232 | LDB CODE\_,#02H | | |
| 22BC | EF93FE | LCALL CCW\_PR | | |
| 22BF | B10231 | LDB PAGE\_,#02H | | |
| 22C2 | B0381C | LDB AL,CTEMP | | |
| 22C5 | 75301C | ADDB AL,#30H | | |
| 22C8 | B01C30 | LDB COLUMN,AL | | |
| 22CB | B10332 | LDB CODE\_,#03H | | |
| 22CE | EF81FE | LCALL CCW\_PR | | |
| 22D1 | EF3000 | LCALL DELAY | | |
| 22D4 | EF2D00 | LCALL DELAY | | |
| 22D7 | EF2A00 | LCALL DELAY | | |
| 22DA | B0341C | LDB AL, DIR | | |
| 22DD | 99001C | CMPB AL,#0 | | |
| 22E0 | D710 | JNE LEFT | | |
| 22E2 | 0738 | INC CTEMP | | |
| 22E4 | B0381C | LDB AL,CTEMP | | |
| 22E7 | 993A1C | CMPB AL,#58 | | |
| 22EA | D7A3 | JNE AAA | | |
| 22EC | B10134 | LDB DIR,#1 | | |
| 22EF  22F2 | E79DFF  0538 | LJMP AAA  LEFT:  DEC CTEMP | | |
| 22F4 | B0381C | LDB AL,CTEMP | | |
| 22F7 | 99001C | CMPB AL,#0 | | |
| 22FA | D793 | JNE AAA | | |
| 22FC | B10034 | LDB DIR,#0 | | |
| 22FF | E78DFF | LJMP AAA | | |
| 2302 | 27FE | SJMP $ | | |
| 2304 | B1602B | DELAY: | LDB | R6,#060H ;延时子程序 |
| 2307 | B1002A |  | LDB | R5,#00H |
| 230A | FD | DELAY1: | NOP |  |
| 230B | E02AFC |  | DJNZ | R5,DELAY1 |

;

|  |  |  |
| --- | --- | --- |
| 230E E02BF9  2311 F0 |  | DJNZ R6,DELAY1 RET |
| ; 硬件实验二十五 | 电子琴 | H25.ASM |
|  | OUTBIT IN | equ 08002h equ 08001h |
|  | P1Buf | equ 6eh |

PulseCNT equ 70h Tone equ 72h

P1 equ IOPORT1

Speaker equ 1

org 2000h

2000 9420 dcw T1Int

org 2080h

2080 E79600 ljmp Start

org 2084h

ToneTable: ; 音调表

2084 C8FD05FE

2088 3CFE56FE

208C 85FEA7FE

2090 D3FE

2092 E4FE dcw 64968, 65029, 65084, 65110, 65157, 65191, 65235, 65252

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | T1Int: |  |  | |
| 2094 | F2 |  | pushf |
| 2095 | B10F14 |  | ldb | wsr, #15 |  |
| 2098 | A0720A |  | ld | t1,Tone | ; 重置音调定时常数 |
| 209B  209E | B10014  B06E0F |  | ldb  ldb | wsr, #0  P1, P1Buf | ; 输出脉冲到 P1 口 |

20A1 95016E xorb P1Buf, #Speaker; P1.0 取 反

20A4 1570 decb PulseCNT

20A6 F3 popf

20A7 F0 ret

TestKey:

20A8 C701028000 stb 0, OUTBIT ; 输出线置为 0

20AD B30101801C ldb al, IN ; 读入键状态

20B2 121C notb al

20B4 710F1C andb al, #0fh ; 高四位不用

20B7 F0 ret

KeyTable: ; 键码定义

20B8 161514FF db 16h, 15h, 14h, 0ffh

20BC 13121110 db 13h, 12h, 11h, 10h

20C0 0D0C0B0A db 0dh, 0ch, 0bh, 0ah

20C4 0E030609 db 0eh, 03h, 06h, 09h

20C8 0F020508 db 0fh, 02h, 05h, 08h

20CC 00010407 db 00h, 01h, 04h, 07h

GetKey:

20D0 B12021 ldb ch, #00100000b

20D3 B10620 ldb cl, #6

KLoop:

20D6 1221 notb ch ; 找出键所在列

20D8 C701028021 stb ch, OUTBIT

20DD 1221 notb ch

20DF 180121 shrb ch, #1

20E2 B30101801C ldb al, IN

20E7 121C notb al

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 20E9  20EC  20EE | 710F1C D708 E020E5 |  | andb jne djnz | al, #0fh  Goon\_ ; 该列有键入  cl, KLoop |
| 20F1  20F4  20F6 | B1FF20 2013  1520 | Goon\_: | ldb sjmp  decb | cl, #0ffh ; 没有键按下, 返回 0ffh Exit1  cl |
| 20F8 | 190220 |  | shlb | cl, #2 ; 键值 = 列 X 4 + 行 |
| 20FB  20FE | B10421  381C08 | LoopC: | ldb  jbs | ch, #4  al, 0, Exit1 |
| 2101 | 18011C |  | shrb | al, #1 |
| 2104 | 1720 |  | incb | cl |
| 2106  2109 | E021F5  C701028000 | Exit1: | djnz  stb | ch, LoopC  0, OUTBIT |
| 210E | 1620 |  | extb | cx |
| 2110 | B321B8201C |  | ldb | al, KeyTable[cx] ; 取出键码 |
| 2115 | B01C1E |  | ldb | bl, al |
| 2118  2119 | F0  A1E00018 | Start: | ret  ld | sp, #0e0h |
| 211D | B00F6E |  | ldb | P1Buf, P1 |
| 2120 | 71FE6E |  | andb | P1Buf, #!Speaker |
| 2123 | B06E0F |  | ldb | P1, P1Buf |
| 2126 | B10416 |  | ldb | IOC1, #04 ; 允许定时器中断 |
| 2129  212C | B10108  2F7A | MLoop: | ldb  call | INT\_MASK, #01h ; 开放定时器中断  TestKey ; 是否有键按下 |
| 212E  2131 | 99001C DFF9 |  | cmpb je | al,#0  MLoop ; 无键，等待 |
| 2133 | 2F9B |  | call | GetKey ; 有键，读回键码 |
| 2135 | B01C1D |  | ldb | ah,al |
| 2138  213B  213D  2140 | 99011C D3EF 99081C D9EA |  | cmpb jnc cmpb jh | al,#1  MLoop ; 键值 < 1，无效al, #8  MLoop ; 键值 > 8，无效 |
| 2142 | B1001D |  | ldb | ah, #0 |
| 2145 | 151C |  | decb | al |
| 2147  214A | 09011C A31D842072 |  | shl ld | ax, #1  Tone, ToneTable[ax] ; 查表，得音调定时值 |
| 214F | B10F14 |  | ldb | wsr, #15 |
| 2152 | A0720A |  | ld | t1,Tone ; 音调定时值送定时器 |
| 2155 | B10014 |  | ldb | wsr, #0 |
| 2158 | FB |  | ei | ; 打开中断，允许发脉冲 |
| 2159  215C | B16470  990070 | Wait: | ldb  cmpb | PulseCNT, #100 ; 送出一串脉冲  PulseCNT,#0 |
| 215F | D7FB |  | jne | Wait |
| 2161 | FA |  | di | ; 关闭中断 |
| 2162 | E7C7FF |  | ljmp | MLoop |
|  |  |  | end |  |

; 硬件实验二十六 空调温度控制实验 H26.ASM

|  |  |  |
| --- | --- | --- |
| OUTBIT | equ 08002h | ; 位控制口 |
| OUTSEG | equ 08004h | ; 段控制口 |
| IN | equ 08001h | ; 键盘读入口 |

P1 equ IOPORT1

LEDBuf equ 60h ; 显示缓冲

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 2080 | E75101 | org  ljmp | | 2080h  Start | | |
| 2083 | 3F065B4F | LEDMAP: | | ; 八段管显示码 | | |
| 2087  208B  208F | 666D7D07  7F6F777C  395E7971 | db  db | | 3fh, 06h, 5bh, 4fh, 66h, 6dh, 7dh, 07h  7fh, 6fh, 77h, 7ch, 39h, 5eh, 79h, 71h | | |
| 2093 | C81C | Delay: push | | ax | ; 延时子程序 | |
| 2095 | B0001C | ldb DelayLoop: | | al, 0 |  | |
| 2098 | E11CFD | djnzw | | ax, DelayLoop | | |
| 209B  209D  209E | CC1C F0  A1600022 | pop ret  DisplayLED:  ld | | ax  dx, #LEDBuf | | |
| 20A2 | B10620 | ldb | | cl, #6 ; 共 6 个八段管 | | |
| 20A5 | B1201D | ldb  DLoop: | | ah, #00100000b ; 从左边开始显示 | | |
| 20A8  20AD  20B0  20B5 | C701028000 B2221C C70104801C C70102801D | stb ldb stb stb | | 0, OUTBIT  al, [dx] al, OUTSEG ah, OUTBIT | | ; 关所有八段管  ; 显示一位八段管 |
| 20BA  20BC  20BF  20C1 | C81C B1011D 2FD2 CC1C | push ldb call pop | | ax  ah, #1 Delay ax | |  |
| 20C3 | 18011D | shrb | | ah, #1 | |  |
| 20C6  20C8 | 0722  E020DD | inc djnz | | dx  cl, DLoop | |  |
| 20CB  20D0 | C701028000 F0 | stb ret | | 0, OUTBIT | | ; trun off all LED |
|  |  | TestKey: | |  |  |  |
| 20D1  20D6  20DB  20DD  20E0  20E1  20E5  20E9  20ED  20F1  20F5 | C701028000 B30101801C 121C  710F1C F0  161514FF  13121110  0D0C0B0A  0E030609  0F020508  00010407 | stb ldb notb andb ret  KeyTable:  db db db db db  db | ; | 0, OUTBIT  al, IN al  al, #0fh  键码定义  16h, 15h, 14h,  13h, 12h, 11h,  0dh, 0ch, 0bh, 0eh, 03h, 06h, 0fh, 02h, 05h,  00h, 01h, 04h, | | ; 输出线置为 0  ; 读入键状态  ; 高四位不用  0ffh 10h 0ah 09h 08h  07h |
| 20F9 | B12021 | GetKey:  ldb | ch, #00100000b | | | |
| 20FC | B10620 | ldb | cl, #6 | | | |
|  |  | KLoop: |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 20FF | 1221 |  | notb | ch | ; 找出键所在列 |
| 2101 | C701028021 |  | stb | ch, OUTBIT |  |
| 2106 | 1221 |  | notb | ch |  |
| 2108 | 180121 |  | shrb | ch, #1 |  |
| 210B | B30101801C |  | ldb | al, IN |  |
| 2110 | 121C |  | notb | al |  |
| 2112 | 710F1C |  | andb | al, #0fh |  |
| 2115 | D708 |  | jne | Goon\_ | ; 该列有键入 |
| 2117 | E020E5 |  | djnz | cl, KLoop |  |
| 211A  211D  211F | B1FF20 2013  1520 | Goon\_: | ldb sjmp  decb | cl, #0ffh Exit1  cl | ; 没有键按下, 返回 0ffh |
| 2121 | 190220 |  | shlb | cl, #2 | ; 键值 = 列 X 4 + 行 |
| 2124  2127 | B10421  381C08 | LoopC: | ldb  jbs | ch, #4  al, 0, Exit1 |  |
| 212A | 18011C |  | shrb | al, #1 |  |
| 212D  212F  2132 | 1720  E021F5  C701028000 | Exit1: | incb djnz  stb | cl  ch, LoopC  0, OUTBIT |  |
| 2137 | 1620 |  | extb | cx |  |
| 2139 | B321E1201C | ldb | | al, KeyTable[cx] ; 取出键码 | |
| 213E | B01C1E | ldb | | bl, al | |

WaitRelease:

2141 C701028000 stb 0, OUTBIT ; 等键释放

2146 B10A1D ldb ah, #10

2149 2F48 call Delay

214B 2F84 call TestKey

214D D7F2 jne WaitRelease

214F B01E1C ldb al, bl

2152 F0 ret

; ======================================================

UP equ 16h ; Next

DOWN equ 15h ; Last LowLimit equ 10

HighLimit equ 30

LowTemp equ -40 ; A/D 0

HighTemp equ 50 ; A/D 255

ADPort equ 8000h

Heat equ 1 ; P1.0

Cool equ 2 ; P1.1

SetTemp equ 50h CurTemp equ 51h P1Buf equ 52h

DisplayResult:

2153 B0511C ldb al, CurTemp

2156 371C0A jbc al, 7, GE0

2159 B14060 ldb LEDBuf, #40h ; '-'

215C 151C decb al

215E 121C notb al

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2160 | E70300 | GE0: | jmp | Goon |
| 2163  2166 | B10060  B10A20 | Goon: | ldb  ldb | LEDBuf, #0 ; ' '  cl, #10 |
| 2169  216C | B0001D FE9C201C |  | ldb divb | ah, 0 ax, cl |
| 2170 | B01C1E |  | ldb | bl, al |
| 2173 | B0001F |  | ldb | bh, 0 |
| 2176 | B31F832061 |  | ldb | LEDBuf+1, LEDMAP[bx] |
| 217B  217E | B01D1E B31F832062 |  | ldb ldb | bl, ah  LEDBuf+2, LEDMAP[bx] |
| 2183 | B10063 |  | ldb | LEDBuf+3, #0 ; ' ' |
| 2186 | B0501C |  | ldb | al, SetTemp |
| 2189  218C | B0001D FE9C201C |  | ldb divb | ah, 0 ax, cl |
| 2190 | B01C1E |  | ldb | bl, al |
| 2193 | B0001F |  | ldb | bh, 0 |
| 2196 | B31F832064 |  | ldb | LEDBuf+4, LEDMAP[bx] |
| 219B  219E | B01D1E B31F832065 |  | ldb ldb | bl, ah  LEDBuf+5, LEDMAP[bx] |
| 21A3 | F0 |  | ret |  |
| 21A4 | C701008000 | ReadAD:  stb | | 0, ADPort |

21A9 B00022 ldb dl, 0

21AC E022FD djnz dl, $ ; delay

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 21AF  21B4 | B30100801C F0 | ldb ret  ReadTemp: | | al, ADPort |
| 21B5 | A0001E | ld | | bx, 0 |
| 21B8 | B11020 | ldb  RLoop: | | cl, #16 |
| 21BB  21BD  21C0  21C3 | 2FE7 B0001D 641C1E E020F5 |  | call ldb add djnz | ReadAD ah, 0 bx, ax  cl, RLoop |
| 21C6 | 08041E |  | shr | bx, #4 |
| 21C9  21CC  21CF  21D3  21D4  21D8  21DB  21DD | 7D5A1E B01F1C 55D81C51 F0  A1E00018 B11450  2EF4 D736 | Start: MLoop: | mulub ldb addb ret  ld ldb  call jne | bl, #(HighTemp-LowTemp) al, bh ; /256  CurTemp, al, #LowTemp  sp, #0e0h SetTemp, #20  TestKey KeyPressed |
| 21DF  21E1  21E3 | 2F72  2EBB  2FD0 |  | call call call | DisplayResult DisplayLED ReadTemp |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 21E5 | B0501C |  | ldb | al, SetTemp |
| 21E8 | 151C |  | decb | al |
| 21EA | 151C |  | decb | al |
| 21EC | 981C51 |  | cmpb | CurTemp, al |
| 21EF | D608 |  | jge | GN2 |
| 21F1 | 910152 |  | orb | P1Buf, #Heat |
| 21F4 | 71FD52 |  | andb | P1Buf, #!Cool |
| 21F7 | 2017 |  | sjmp | GN4 |
|  |  | GN2: |  |  |
| 21F9 | B0501C |  | ldb | al, SetTemp |
| 21FC | 071C |  | inc | al |
| 21FE | 071C |  | inc | al |
| 2200 | 981C51 |  | cmpb | CurTemp, al |
| 2203 | DA08 |  | jle | GN3 |
| 2205 | 910252 |  | orb | P1Buf, #Cool |
| 2208 | 71FE52 |  | andb | P1Buf, #!Heat |
| 220B | 2003 |  | sjmp | GN4 |
|  |  | GN3: |  |  |
| 220D | 71FC52 |  | andb | P1Buf, #!(Cool+Heat) |
|  |  | GN4: |  |  |
| 2210 | B0520F |  | ldb | P1, P1Buf |
| 2213 | 27C6 |  | sjmp | MLoop |

KeyPressed:

2215 2EE2 call GetKey

2217 99151C cmpb al, #DOWN

221A D709 jne Key0

221C 990A50 cmpb SetTemp, #LowLimit

221F DF10 je Key1

2221 1550 decb SetTemp

2223 200C sjmp Key1 Key0:

2225 99161C cmpb al, #UP

2228 D707 jne Key1

222A 991E50 cmpb SetTemp, #HighLimit

222D DF02 je Key1

222F 1750 incb SetTemp Key1:

2231 27A8 sjmp MLoop

end

; 硬件实验二十八 用 HSO 方式输出 PWM 波形 H28.ASM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | hi\_time lo\_time | equ equ | 750  3000-hi\_time |  |
| 2006 | 9320 |  | org dcw | 2006h  pwm\_out |
| 2080 | B1C018 |  | org ldb | 2080h  sp,#0c0h |
| 2083 | B13006 |  | ldb | hso\_command,#30h | ; 选择 HSO0 及 T1 |
| 2086  208B  208E  2090 | 45EE020A04 B10808 1109  FB |  | add ldb clrb  ei | hso\_time,t1,#hi\_time int\_mask, #08h int\_pend | ; 定时  ; 允许 HSO 中断 |
| 2091 | 27FE | jmp pwm\_out: | | $ |  |
| 2093 | F2 | pushf | |  |  |
| 2094 | C81C | push | | ax |  |
| 2096 | B0151C | ldb | | al,ios0 | ; 检查 HSO0 状态 |
| 2099 | 301C0A | jbc | | al,0, pwm\_hi |  |

pwm\_lo:

209C B11006 ldb hso\_command,#10h

209F 45CA080A04 add hso\_time,t1,#lo\_time ; 低电平定时常数

20A4 2008 sjmp pwm\_ret pwm\_hi:

20A6 B13006 ldb hso\_command,#30h

20A9 45EE020A04 add hso\_time,t1,#hi\_time ; 高电平定时常数

pwm\_ret:

20AE CC1C pop ax

20B0 F3 popf

20B1 F0 ret

end

; 硬件实验二十九 用 HSI 方式测量脉冲宽度 H29.ASM

|  |  |  |  |
| --- | --- | --- | --- |
|  | | HSI\_TIME equ P1 equ | 4  IOPORT1 |
| org | 2080h |
| 2080 | A1C00018 | ld | sp, #0c0h |
| 2084 | B10115 | ldb | ioc0, #01h ; 允许 HSI.0 输入 |
| 2087 | B10303 | ldb | hsi\_mode, #03h ; HSI.0 方式 3 |
| 208A | FD | nop |  |
| 208B  208C | FD B1010F | nop ldb | p1,#01 ; P10 置高 |
| 208F | B1000F | ldb | p1,#00 ; P10 置低,产Th下降沿 |
| 2092 | A1011522 | ld | dx, #1501h ; 延时 |
| 2096 | E122FD | djnzw dx,$ | |
| 2099 | B1010F | ldb p1,#01 ; P10 置高,产Th上升沿 | |
| 209C | FD | nop | |
| 209D | FD | nop | |
| 209E | B0061C | ldb al,hsi\_status ; 先读下降沿事件状态 | |
| 20A1 | A0041E | ld bx,hsi\_time ; 再读下降沿的时间 | |

20A4 B0061C ldb al,hsi\_status ; 先读上升沿事件状态

20A7 A00420 ld cx,hsi\_time ; 再读上升沿的时间

20AA 481E2022 sub dx, cx, bx ; 相减得脉冲宽度

20AE FD nop ; 在此行设断点

20AF 27FE jmp $

end

; 硬件实验三十 用 HSI 中断方式统计脉冲个数 H30.ASM

hsi\_time equ 4 counter equ cx result equ dx

org 2004h

2004 AC20 dcw hsi1\_int ; HSI 事件中断

org 2008h

2008 A620 dcw hsi0\_int ; HSI.0 脚中断

org 2080h

2080 A1C00018 ld sp,#0c0h

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2084 | B16F16 |  | ldb | ioc1,#6fh | ; HSI 中断且 HSI.2 输入 |
| 2087 | B10003 |  | ldb | hsi\_mode,#0 | ; HSI 方式 0, 8 次跳变为一事件 |
| 208A | B11008 |  | ldb | int\_mask,#10h | ; HSI.0 脚中断 |
| 208D | 910408 |  | orb | int\_mask,#04h | ; HSI 事件中断 |
| 2090 | B11015 |  | ldb | ioc0,#10h | ; 允许 HSI.2 输入 |
| 2093 | 1109 |  | clrb | int\_pend |  |
| 2095 | A1000020 |  | ld | counter,#0 |  |
| 2099 | A1000022 |  | ld | result,#0 |  |
| 209D | FB |  | ei |  |  |
|  |  | wait: |  |  |  |
| 209E | 89000022 |  | cmp | result, #00 |  |
| 20A2 | DFFA |  | je | wait |  |
| 20A4 | 27FE |  | jmp | $ |  |

hsi0\_int:

20A6 F2 pushf

20A7 A02022 ld result, counter ; 读取计数结果

20AA F3 popf

20AB F0 ret

hsi1\_int:

20AC F2 pushf

20AD C81C push ax

20AF B0061C ldb al, hsi\_status ; 读取 HSI 状态

20B2 A00428 ld 28h,hsi\_time ; 读取 HSI 时间

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 20B5  20B8 | 341C02  0720 | h\_ret: | jbc inc | al,4,h\_ret ; 是 HSI.2 事件counter ; 计数加 1 |
| 20BA | CC1C |  | pop | ax |
| 20BC | F3 |  | popf |  |
| 20BD | F0 |  | ret  end |  |

; 硬件实验三十一 计数器实验 H31.ASM

org 2080h

Start:

2080 A1E00018 ld sp, #0e0h

2084 A0000C ld T2, 0

Loop:

2087 A00C1C ld ax, T2 ; 读取 T2CLK 脚上的上/下沿

208A B01C0F ldb IOPORT1, al ; 将计数结果送到 P1 口

208D E7F7FF ljmp Loop

end

; 硬件实验三十二 片内 A/D 转换实验 H32.ASM

ad\_cmd equ 02h ad\_hi equ 03h ad\_lo equ 02h

mode equ 082h ; 方式 0，PA，PC 输出，PB 输入

PortA equ 8000h ; Port A CAddr equ 8003h ; 控制字地址

org 2080h

start:

2080 B1821C ldb al, #mode

2083 C70103801C stb al, CAddr ; 输出控制字

2088 B10802 ldb ad\_cmd ,#08h ; start 0 channel a/d

208B FD nop ; delay

208C FD nop

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 208D  208E  208F | FD FD FD | test: | nop nop nop |  |  |
| 2090 | B0021C |  | ldb | al,ad\_lo |
| 2093 | 3B1CFA |  | jbs | al,3, test | ; Is it converting? |
| 2096 | B0031D |  | ldb | ah,ad\_hi | ; Read high 8 bit |
| 2099 | C70100801D |  | stb | ah, PortA | ; Output high 8 bit to LED |
| 209E | B0021C |  | ldb | al,ad\_lo | ; Read low 2 bit |
| 20A1 | 08061C |  | shr | ax, #6 |  |
| 20A4 | 27DA |  | jmp  end | start |  |

# 附录 D MCS96 C 语言程序清单

// 软件实验一 存储器块清零 S1.C

unsigned char Buffer[256]; #pragma locate(Buffer = 0x6000)

void main()

{

unsigned int index; unsigned char \* ptr;

ptr = &Buffer; // 起始地址

for (index = 0; index <= 255; index++) {

\*ptr++ = 0; // 清 0, 地址加一

}

}

// 软件实验二 二进制到 BCD 码转换 S2.C

// 将 Number 拆为三个 BCD 码, 并存入 Result 数组

void main()

{

unsigned char Result[3]; unsigned char Number;

Number = 123;

Result[0] = Number / 100; // 除以 100, 得百位数Result[1] = (Number 100) / 10; // 余数除以 10, 得十位数Result[2] = Number 10; // 余数为个位数

}

// 软件实验三 二进制到 ASCII 码转换 S3.C

// 将 A 拆为二个 ASCII 码, 并存入 Result 数组

unsigned char const ASCIITable[16] = "0123456789ABCDEF"; // 定义数字对应的 ASCII 表

void main()

{

unsigned char Result[2]; unsigned char Number;

Number = 0x1a;

Result[0] = ASCIITable[Number >> 4 ]; // 高四位

Result[1] = ASCIITable[Number & 0xf]; // 低四位

}

// 软件实验四 内存块移动 S4.C

unsigned char Buffer1[256]; unsigned char Buffer2[256]; #pragma locate(Buffer1=0x4000) #pragma locate(Buffer2=0x4100)

void main()

{

unsigned int index; unsigned char \* ptr1; unsigned char \* ptr2;

ptr1 = &Buffer1[0]; ptr2 = &Buffer2[0];

for (index=0; index <= 255; index++) {

\*ptr2++ = \*ptr1++;

}

}

// 软件实验五 程序跳转表 S5.C

void Func0() {}

void Func1() {}

void Func2() {}

void Func3() {}

void FuncEnter(unsigned char FuncID)

{

switch (FuncID) { case 0:

Func0(); break;

case 1: Func1(); break;

case 2: Func2(); break;

case 3: Func3(); break;

default: break;

}

}

void main()

{

FuncEnter(0);

FuncEnter(1); FuncEnter(2); FuncEnter(3);

}

// 软件实验六 数据排序 S6.C

#define Size 10

const unsigned char DataArr[Size] ={5,2,0,1,3,6,5,8,7,9}; unsigned char Array[Size];

void main ()

{

unsigned char Change; unsigned char index; unsigned char Temp;

for(index = 0; index < Size; index++) Array[index] = DataArr[index];

do {

Change = 0;

for (index = 0; index < (Size-1); index++) { if (Array[index] > Array[index+1]) {

Change = 1;

Temp = Array[index]; Array[index] = Array[index+1]; Array[index+1] = Temp;

}

}

} while (Change);

}

// 硬件实验一 P1 口输入、输出实验 H1A.C #include <kc\_sfrs.h>

void delay()

{

unsigned int i;

for (i=0; i<20000; i++) {}

}

void main()

{

unsigned char index; unsigned char LED;

while (1) {

LED = 1;

for (index=0; index < 8; index++) {

ioport1 = LED;

LED <<= 1;

delay();

}

}

}

// 硬件实验一 P1 口输入、输出实验 H1B.C #include <kc\_sfrs.h>

#define KeyLeft 1

#define KeyRight 2

#define LEDLeft 4

#define LEDRight 8

unsigned char P1Buf; void main()

{

while (1) {

P1Buf = ioport1 | (KeyLeft+KeyRight); ioport1 = P1Buf;

if (ioport1 & KeyLeft ) P1Buf |= LEDLeft ; else P1Buf &= ~LEDLeft ; if (ioport1 & KeyRight) P1Buf |= LEDRight; else P1Buf &= ~LEDRight; ioport1 = P1Buf;

}

}

// 硬件实验二 继电器控制实验 H2.C #include <kc\_sfrs.h>

#define Output 1 // P1.0 unsigned char P1Buf;

void Delay()

{

unsigned int i;

for (i=0; i<20000; i++) ;

}

void main()

{

P1Buf = ioport1; while (1) {

P1Buf &= ~Output; // Output = 0 ioport1 = P1Buf;

Delay();

P1Buf |= Output; // Output = 1 ioport1 = P1Buf;

Delay();

}

}

// 硬件实验三 用 74LS245 扩展 I/O 口实验 H3.C

unsigned char volatile CS245; #pragma locate(CS245 = 0x8000)

void main()

{

unsigned char b;

while (1) { b = CS245;

}

}

// 硬件实验四 用 74LS273 扩展 I/O 口实验 H4.C

unsigned char volatile CS273; #pragma locate(CS273 = 0x8000) void delay()

{

unsigned int i; i = 10000;

do{} while(--i != 0);

}

void main()

{

unsigned char i, b;

do{

b = 1;

for (i=0; i<8; i++) { CS273 = b;

b <<= 1;

delay();

};

}while(1);

}

// 硬件实验五 PWM 转换电压实验 H5.C

// 输出 50 (5:5) 占空比 PWM

// 输出 10 (1:9) 占空比 PWM

// 输出 90 (9:1) 占空比 PWM

#include <kc\_sfrs.h> #define OUTPUT 1 // P1.0

unsigned char P1Buf;

void Delay(unsigned char CNT)

{

unsigned char i; while (CNT--) {

for (i=0; i<50; i++) ;

}

}

void main()

{

P1Buf = ioport1; while (1) {

P1Buf &= ~OUTPUT;

ioport1 = P1Buf;

Delay(5);

P1Buf |= OUTPUT;

ioport1 = P1Buf;

Delay(5);

}

}

// 硬件实验六 音频控制实验 H6.C

#include <kc\_sfrs.h> #define Speaker 1 // P1.0

unsigned char P1Buf; void Delay()

{

unsigned char i;

for (i=0; i<50; i++);

}

void main()

{

P1Buf = ioport1; while (1) {

P1Buf &= ~Speaker; // Speaker = 0 ioport1 = P1Buf;

Delay();

P1Buf |= Speaker; // Speaker = 1 ioport1 = P1Buf;

Delay();

}

}

// 硬件实验七 8255 输入、输出实验 H7.C

#define mode 0x82 // 方式 0，PA，PC 输出，PB 输入

unsigned char PortA, PortB, PortC; unsigned char CAddr, buf;

#pragma locate(PortA = 0x8000) // Port A #pragma locate(PortB = 0x8001) // Port B #pragma locate(PortC = 0x8002) // Port C #pragma locate(CAddr = 0x8003) // 控制字地址

unsigned char i;

void Delay(unsigned int CNT)

{

while (CNT) CNT--;

}

void main()

{

while(1){

CAddr = mode; // 输出控制字

// 实验 1：PortA 输出

buf = 0x80; for(i=0; i<8; i++){

PortA = buf; // 输出到 A 端口

PortC = buf;

buf >>= 1; // 移位

Delay(0x5000); // 延 时

};

// 实验 2： PortB 输入 PortA 输出buf = PortB; // 读 入 PortB PortA = buf; // 输出到 PortA Delay(0x8000);

}

}

// 硬件实验八 串行数转换并行数实验 H8A.C #include <kc\_sfrs.h>

void Send164(unsigned char b)

{

ioc1 = 0x20; // P2.0 选择为 TXD sp\_con = 0; // 串 口 方 式 0 sbuf\_tx = b; // 输出 b

}

void main()

{

Send164(0x55);

// 硬件实验八 串行数转换并行数实验 H8B.C #include <kc\_sfrs.h>

#define DAT 1 // P1.0

#define CLK 2 // P1.1 unsigned char P1Buf;

void Send164(unsigned char b)

{

unsigned char i;

P1Buf |= CLK;

for (i=0; i<8; i++) {

(b & 1 ? P1Buf |= DAT : P1Buf &= ~DAT); P1Buf &= ~CLK;

ioport1 = P1Buf;

P1Buf |= CLK;

ioport1 = P1Buf; b >>= 1;

}

}

void main()

{

P1Buf = ioport1;

Send164(0x55);

}

// 硬件实验九 并行数转换串行数实验 H9A.C #include <kc\_sfrs.h>

#define LD165 1 // P1.0 unsigned char P1Buf;

unsigned char Read165()

{

unsigned char STAT;

P1Buf &= ~LD165;

ioport1 = P1Buf; // LD165 = 0

P1Buf |= LD165;

ioport1 = P1Buf; // LD165 = 1

ioc1 = 0x20; // P2.0 选 择 为 TXD sp\_con = 0x08; // 串口方式 0, REN = 1

do {

STAT = sp\_stat;

} while ( (STAT & 0x40) == 0); // RI = 1? return(sbuf\_rx);

}

void main()

{

unsigned char b;

P1Buf = ioport1; b = Read165(); while(1);

}

// 硬件实验九 并行数转换串行数实验 H9B.C #include <kc\_sfrs.h>

#define LD165 1 // P1.0 #define DAT 2 // P1.1

#define CLK 4 // P1.2 unsigned char P1Buf;

unsigned char Read165()

{

unsigned char i, b;

P1Buf &= !LD165; // LD165 = 0

ioport1 = P1Buf;

P1Buf |= LD165; // LD165 = 1

ioport1 = P1Buf;

b = 0;

for (i=0; i<8; i++) { b >>= 1;

if ((ioport1 & DAT) != 0) b |= 0x80; P1Buf &= ~CLK;

ioport1 = P1Buf; // CLK = 0; P1Buf |= CLK;

ioport1 = P1Buf; // CLK = 1;

}

return(b);

}

void main()

{

unsigned char b;

P1Buf = ioport1; b = Read165();

//硬件实验十一 外部中断实验 H11.C

#include <kc\_sfrs.h> #include <kc\_funcs.h>

#define LED 1 // P1.0 unsigned char P1Buf;

#pragma interrupt(ExtInt=7) void ExtInt()

{

P1Buf ^= LED;

ioport1 = P1Buf;

}

void main()

{

ioc1 &= ~0x02; P1Buf = ioport1; P1Buf &= ~LED; ioport1 = P1Buf;

int\_mask = 0x80; // 打开外部中断允许位

enable(); // 打开总中断允许位

while (1) ;

}

// 硬件实验十二 定时器实验 H12.C #include <kc\_sfrs.h>

#include <kc\_funcs.h>

#define Tick 100 // 100 x 10ms = 1s #define T10ms (65536-7500) // 10ms 时间常数(6M)

#define LED 1 // P1.0 unsigned char C10ms; // 10ms 记数单元unsigned char P1Buf;

#pragma interrupt(T1Int=0) void T1Int()

{

wsr = 15; timer1 = T10ms; wsr = 0;

C10ms--;

if (C10ms == 0) {

C10ms = Tick; // 10ms 记数器为 0, 重置记数器

P1Buf ^= LED; // 取反 LED

}

}

void main()

{

P1Buf = ioport1; P1Buf &= ~LED; ioport1 = P1Buf;

wsr = 15; timer1 = T10ms; wsr = 0;

C10ms = Tick; ioc1 = 0x04;

int\_mask = 0x01; // 打开定时器中断允许位

enable(); // 打开总中断允许位

while (1) { ioport1 = P1Buf;

}

}

// 硬件实验十三 D/A 转换实验 H13.C

unsigned char CS0832;

#pragma locate(CS0832 = 0x8000)

unsigned char SinTbl[9] =

{0x00,0x18,0x30,0x46,0x59,0x69,0x75,0x7c,0x7f};

void Write0832(unsigned char b)

{

CS0832 = b;

}

unsigned char i;

void main()

{

while(1){

for(i=0; i<16; i++) // 产Th锯齿波

Write0832(i\*0x10);

for(i=0; i<8; i++) // 产Th正弦波

Write0832(0x80 + SinTbl[i]); // (0 ~ π/2) 区间的波形

for(i=8; i>0; i--)

Write0832(0x80 + SinTbl[i]); // (π/2 ~ π) 区间的波形

for(i=0; i<8; i++)

Write0832(0x80 - SinTbl[i]); // (π ~ 3π/2) 区间的波形

for(i=8; i>0; i--)

Write0832(0x80 - SinTbl[i]); // (3π/2 ~ 2π) 区间的波形

}

}

// 硬件实验十四 A/D 转换实验 H14.C

#define mode 0x82 // 方式 0，PA，PC 输出，PB 输入

unsigned char PortA; unsigned char CAddr; unsigned char CS0809;

#pragma locate(PortA = 0x8000) // Port A #pragma locate(CAddr = 0x8003) // 控制字地址#pragma locate(CS0809 = 0x9000)

unsigned char Read0809()

{

unsigned char i,j; unsigned int r;

r =0;

for(j=0; j<20; j++){

CS0809 = 0; // 起动 A/D

for (i=0; i<0x40; i++) ; // 延时 > 100us r += CS0809; // 读入结果

}

return(r / 20);

}

void main()

{

unsigned char b;

CAddr = mode; // 输出控制字

do{

b = Read0809();

PortA = b; // 输出到 A 端口

}while(1);

}

//硬件实验十五 外部中断实验（急救车与交通灯） H15.C

#include <kc\_sfrs.h> #include <kc\_funcs.h>

#define SR 1 // 南北红灯#define SY 2 // 南北黄灯#define SG 4 // 南北绿灯

#define ER 8 // 东西红灯#define EY 16 // 东西黄灯#define EG 32 // 东西绿灯

#define mode 0x82 // 8255 方式 0，PA，PC 输出，PB 输入

unsigned char STOP; unsigned char LEDBuf; unsigned char LED;

unsigned char LED; unsigned char CAddr;

#pragma locate(CAddr = 0x8003) // 控制字地址

#pragma locate(LED = 0x8000)

#pragma interrupt(StopInt=7) void StopInt()

{

STOP = 1;

}

void Delay(unsigned char CNT)

{

unsigned int I;

while ((CNT > 0) && !STOP) {

for (I=0; (I < 20000) && !STOP; I++) ; CNT--;

}

}

void main()

{

unsigned char I;

CAddr = mode; // 输出 8255 控制字

ioc1 &= ~0x02;

int\_mask = 0x80; // 打开外部中断允许位enable(); // 打开总中断允许位STOP = 0;

LEDBuf = SR + ER; // 南北, 东西均红灯

LED = LEDBuf;

while (1) {

if (STOP) goto AllRed;

LEDBuf = SR + EG; // 南北红灯, 东西绿灯

LED = LEDBuf;

Delay(10);

if (STOP) goto AllRed;

LEDBuf = SR; // 南北红灯, 东西黄灯闪

LED = LEDBuf;

for (I=0; I < 9; I++) {

Delay(1);

LEDBuf ^= EY;

LED = LEDBuf;

}

if (STOP) goto AllRed;

LEDBuf = SG + ER; // 南北绿灯, 东西红灯

LED = LEDBuf;

Delay(10);

if (STOP) goto AllRed;

LEDBuf = ER; // 东西红灯, 南北黄灯闪

LED = LEDBuf;

for (I=0; I < 9; I++) {

Delay(1);

LEDBuf ^= SY;

LED = LEDBuf;

}

AllRed:

if (STOP) {

LEDBuf = SR + ER; // 南北, 东西均红灯

LED = LEDBuf;

STOP = 0;

Delay(10);

}

}

}

//硬件实验十六 八段数码管显示实验 H16.C #define LEDLen 6

unsigned char volatile OUTBIT; // 位控制口unsigned char volatile OUTSEG; // 段控制口unsigned char volatile IN ; // 键盘读入口

#pragma locate(OUTBIT = 0x8002) #pragma locate(OUTSEG = 0x8004) #pragma locate(IN = 0x8001)

unsigned char LEDBuf[LEDLen]; // 显 示 缓 冲

unsigned char const LEDMAP[] = { // 八段管显示码0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71

};

void Delay(unsigned char CNT)

{

unsigned char i;

while (CNT-- !=0)

for (i=100; i !=0; i--);

}

void DisplayLED()

{

unsigned char i, j, k; unsigned char Pos; unsigned char LED;

Pos = 0x20; // 从左边开始显示

for (i = 0; i < LEDLen; i++) { OUTBIT = 0; // 关所有八段管LED = LEDBuf[i];

OUTSEG = LED;

OUTBIT = Pos; // 显示一位八段管

Delay(1);

Pos >>= 1; // 显示下一位

}

}

void main()

{

unsigned char i = 0; unsigned char j;

while(1) {

LEDBuf[0] = LEDMAP[ i & 0x0f]; LEDBuf[1] = LEDMAP[(i+1) & 0x0f]; LEDBuf[2] = LEDMAP[(i+2) & 0x0f]; LEDBuf[3] = LEDMAP[(i+3) & 0x0f]; LEDBuf[4] = LEDMAP[(i+4) & 0x0f]; LEDBuf[5] = LEDMAP[(i+5) & 0x0f];

i++;

for(j=0; j<50; j++) DisplayLED(); // 延时

}

}

//硬件实验十七 键盘扫描显示实验 H17.C #define LEDLen 6

unsigned char volatile OUTBIT; // 位控制口

unsigned char volatile OUTSEG; // 段控制口

unsigned char volatile IN ; // 键盘读入口

#pragma locate(OUTBIT = 0x8002) #pragma locate(OUTSEG = 0x8004) #pragma locate(IN = 0x8001)

unsigned char LEDBuf[LEDLen]; // 显示缓冲

unsigned char const LEDMAP[] = { // 八段管显示码0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71

};

void Delay(unsigned char CNT)

{

unsigned char i;

while (CNT-- !=0)

for (i=100; i !=0; i--);

}

void DisplayLED()

{

unsigned char i, j, k; unsigned char Pos; unsigned char LED;

Pos = 0x20; // 从左边开始显示

for (i = 0; i < LEDLen; i++) { OUTBIT = 0; // 关所有八段管LED = LEDBuf[i];

OUTSEG = LED;

OUTBIT = Pos; // 显示一位八段管

Delay(1);

Pos >>= 1; // 显示下一位

}

}

unsigned char const KeyTable[] = { // 键码定义

0x16, 0x15, 0x14, 0xff, 0x13, 0x12, 0x11, 0x10,

0x0d, 0x0c, 0x0b, 0x0a, 0x0e, 0x03, 0x06, 0x09,

0x0f, 0x02, 0x05, 0x08,

0x00, 0x01, 0x04, 0x07

};

unsigned char TestKey()

{

OUTBIT = 0; // 输出线置为 0

return (~IN & 0x0f); // 读入键状态 (高四位不用)

}

unsigned char GetKey()

{

unsigned char Pos; unsigned char i; unsigned char j; unsigned char k;

i = 6;

Pos = 0x20; // 找出键所在列

do {

OUTBIT = ~ Pos;

Pos >>= 1;

k = ~IN & 0x0f;

} while ((--i != 0) && (k == 0));

// 键值 = 列 X 4 + 行

if (k != 0) { i \*= 4;

if (k & 2) i += 1;

else if (k & 4) i += 2;

else if (k & 8) i += 3;

OUTBIT = 0;

do Delay(10); while (TestKey()); // 等键释放

return(KeyTable[i]); // 取出键码

} else return(0xff);

}

void main()

{

LEDBuf[0] = 0xff; LEDBuf[1] = 0xff; LEDBuf[2] = 0xff; LEDBuf[3] = 0xff; LEDBuf[4] = 0x00; LEDBuf[5] = 0x00;

while (1) { DisplayLED();

if (TestKey()) LEDBuf[5] = LEDMAP[GetKey() & 0x0f];

}

}

//硬件实验十八 电子时钟 H18.C

#include <kc\_sfrs.h> #include <kc\_funcs.h>

#define LEDLen 6

unsigned char volatile OUTBIT; // 位控制口unsigned char volatile OUTSEG; // 段控制口unsigned char volatile IN ; // 键盘读入口#pragma locate(OUTBIT = 0x8002)

#pragma locate(OUTSEG = 0x8004) #pragma locate(IN = 0x8001)

unsigned char LEDBuf[LEDLen]; // 显示缓冲

unsigned char const LEDMAP[] = { // 八段管显示码0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71

};

void Delay(unsigned char CNT)

{

unsigned char i;

while (CNT-- !=0)

for (i=100; i !=0; i--);

}

void DisplayLED()

{

unsigned char i, j, k; unsigned char Pos; unsigned char LED;

Pos = 0x20; // 从左边开始显示

for (i = 0; i < LEDLen; i++) { OUTBIT = 0; // 关所有八段管LED = LEDBuf[i];

OUTSEG = LED;

OUTBIT = Pos; // 显示一位八段管

Delay(1);

Pos >>= 1; // 显示下一位

}

}

// ===============================================================================

#define Tick 100 // 100 x 10ms = 1s #define T10ms (65536-7500/2) // 100us 时间常数(6M)

unsigned char Hour, Minute, Second; unsigned char C10ms; // 100us 记数单元

#pragma interrupt(T1Int=0) void T1Int()

{

wsr = 15; timer1 = T10ms;

wsr = 0; C10ms--;

if (C10ms == 0) {

C10ms = Tick; // 100us 记数器为 0, 重置记数器

Second++;

if (Second == 60) { Second = 0; Minute++;

if (Minute == 60) { Minute = 0; Hour++;

if (Hour == 24) Hour = 0;

}

}

}

}

void main()

{

wsr = 15; timer1 = T10ms; wsr = 0;

ioc1 = 0x04;

int\_mask = 0x01; // 打开定时器中断允许位

enable(); // 打开总中断允许位

Hour = 0;

Minute = 0;

Second = 0; C10ms = Tick;

while (1) {

LEDBuf[0] = LEDMAP[Hour/10]; LEDBuf[1] = LEDMAP[Hour 10] | 0x80; LEDBuf[2] = LEDMAP[Minute/10]; LEDBuf[3] = LEDMAP[Minute 10] | 0x80; LEDBuf[4] = LEDMAP[Second/10]; LEDBuf[5] = LEDMAP[Second 10];

DisplayLED();

}

}

//硬件实验十九 单片机串行口通讯实验 H19.C

#include <kc\_sfrs.h> #define LEDLen 6

unsigned char volatile OUTBIT; // 位控制口unsigned char volatile OUTSEG; // 段控制口unsigned char volatile IN ; // 键盘读入口

#pragma locate(OUTBIT = 0x8002) // 定义各端口的地址

#pragma locate(OUTSEG = 0x8004) #pragma locate(IN = 0x8001)

#pragma interrupt(Sio\_int=0x200c)// 定义串口中断向量

unsigned char LEDBuf[LEDLen]; // 显示缓冲

unsigned char RCVBUF,HasRcv;

unsigned char const LEDMAP[] = { // 八段管显示码0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71

};

void Delay(unsigned char CNT)

{

unsigned char i;

while (CNT-- !=0)

for (i=100; i !=0; i--);

}

void DisplayLED()

{

unsigned char i, j, k; unsigned char Pos; unsigned char LED;

Pos = 0x20; // 从左边开始显示

for (i = 0; i < LEDLen; i++) { OUTBIT = 0; // 关所有八段管LED = LEDBuf[i];

OUTSEG = LED;

OUTBIT = Pos; // 显示一位八段管

Delay(1);

Pos >>= 1; // 显示下一位

}

}

unsigned char const KeyTable[] = { // 键码定义

0x16, 0x15, 0x14, 0xff, 0x13, 0x12, 0x11, 0x10,

0x0d, 0x0c, 0x0b, 0x0a, 0x0e, 0x03, 0x06, 0x09,

0x0f, 0x02, 0x05, 0x08,

0x00, 0x01, 0x04, 0x07

};

unsigned char TestKey()

{

OUTBIT = 0; // 输出线置为 0

return (~IN & 0x0f); // 读入键状态 (高四位不用)

}

unsigned char GetKey()

{

unsigned char Pos; unsigned char i; unsigned char j; unsigned char k;

i = 6;

Pos = 0x20; // 找出键所在列

do {

OUTBIT = ~ Pos;

Pos >>= 1;

k = ~IN & 0x0f;

} while ((--i != 0) && (k == 0));

// 键值 = 列 X 4 + 行

if (k != 0) { i \*= 4;

if (k & 2) i += 1;

else if (k & 4) i += 2;

else if (k & 8) i += 3;

OUTBIT = 0;

do Delay(10); while (TestKey()); // 等键释放

return(KeyTable[i]); // 取出键码

} else return(0xff);

}

void Sio\_int()

{

if((sp\_stat & 0x40)){ // RI 位是否为 1? RCVBUF = sbuf\_rx;

HasRcv = 1;

}

}

void main()

{

ioc1 = 0x20; // P2.0 设为 TXD 功能

sp\_con = 0x09; // 串口模式 1，允许接收baud\_rate = 0x9b; // 波特率 2400BPS @ 6MHz baud\_rate = 0x80;

int\_mask1 = 0x00; // 禁止 RI 和 TI 中断int\_mask = 0x40; // 允许串口中断(0x200C) HasRcv = 0;

LEDBuf[0] = 0xff;

LEDBuf[1] = 0xff;

LEDBuf[2] = 0xff; LEDBuf[3] = 0xff; LEDBuf[4] = 0x00; LEDBuf[5] = 0x00;

enable() ; // 允许中断

while (1) {

if(!HasRcv){

DisplayLED(); // 显示

if (TestKey()) // 如果有键按下

sbuf\_tx = GetKey(); // 则用串口输出

} else{

HasRcv = 0; // 已收到数据LEDBuf[5] = LEDMAP[RCVBUF & 0x0f]; // 显示低 4 位LEDBuf[4] = LEDMAP[RCVBUF / 0x10]; // 显示高 4 位

}

}

}

//硬件实验二十 打印机控制试验 H20.C

#define mode 0x82 /\* 8255 工作方式，PA,PC 输出,PB 输入\*/ unsigned char volatile dport; /\* 数据, 8255 PA 口地址 \*/

unsigned char volatile status; /\* 忙状态, 8255 PB 口地址 \*/ unsigned char volatile ctl; /\* 选通控制, 8255 PC 口地址 \*/ unsigned char volatile control; /\* 8255 控制口地址 \*/

#pragma locate(dport = 0x08000) #pragma locate(status = 0x08001) #pragma locate(ctl = 0x08002) #pragma locate(control= 0x08003)

/\* 忙状态位 \*/ #define busy 1

const unsigned char string[] = {"Hello World!\n",0x0c,0x00}; void print(unsigned char dd)

{

//while(status & busy); // 读打印机状态, 若忙，等待

dport = dd; // 数据送出

ctl = 1; // 输出选通脉冲

ctl = 0;

}

void main()

{

unsigned char \*pos;

control = mode;

pos = (unsigned char \*) &string; // 被打印字串

while((\*pos)) print( \*pos++); while(1);

}

//硬件实验二十一 直流电机控制实验 H21.C #include <kc\_sfrs.h>

#define mode 0x82 // 8255 方式 0，PA，PC 输出，PB 输入

unsigned char volatile PortA ; // PA 输出转速unsigned char volatile status; // PB 读入脉冲计数unsigned char volatile CAddr ;

unsigned char volatile CS0832;

#pragma locate(CAddr = 0x8003) // 控制字地址

#pragma locate(PortA = 0x8000) #pragma locate(status = 0x8001) #pragma locate(CS0832 = 0x9000)

unsigned int count; #define DC\_P 1

void delay()

{

unsigned int ddd;

ddd = 60000; // 在 6MHz 约延时 1 秒while(ddd--);

}

unsigned int read()

{

unsigned char j ; unsigned long i ;

while(!(status & DC\_P)); // 等待低电平完

while(status & DC\_P); // 等待高电平完

i = 0;

for(j=0; j<4; j++){ // 测量 4 次

wsr = 15;

timer1 = 0; // 定时器 1 清零

wsr = 0;

while(!(status & DC\_P)); // 等待低电平完

while(status & DC\_P); // 等待高电平完

i += timer1;

} ;

PortA = i >> 10;

return (i >> 2); // 取平均值

}

void main()

{

while(1){

|  |  |  |
| --- | --- | --- |
| CS0832 = 0xff;  delay();  count = read(); | //  //  // | 产Th电压控制电机  等待电机运转稳定  读取时间 |
| CS0832 = 0xd0; | // | 产Th电压控制电机 |
| delay(); | // | 等待电机运转稳定 |
| count = read(); | // | 读取时间 |
| CS0832 = 0x80;  delay(); | //  // | 电机停转 |
| CS0832 = 0x20; | // | 产Th电压控制电机 |
| delay(); | // | 等待电机运转稳定 |
| count = read(); | // | 读取时间 |
| CS0832 = 0x00; | // | 产Th电压控制电机 |
| delay(); | // | 等待电机运转稳定 |
| count = read(); | // | 读取时间 |
| CS0832 = 0x20; | // | 产Th电压控制电机 |
| delay(); | // | 等待电机运转稳定 |
| count = read(); | // | 读取时间 |
| CS0832 = 0x80;  delay(); | //  // | 电机停转 |
| CS0832 = 0xd0; | // | 产Th电压控制电机 |
| delay(); | // | 等待电机运转稳定 |
| count = read(); | // | 读取时间 |

}

}

//硬件实验二十二 步进电机控制实验 H22.C

#define mode8255 0x82 // 8255 工作方式, PA,PC 输出, PB 输入

unsigned char volatile ctl; // 8255 PA 口unsigned char volatile contrl; // 8255 工作方式控制#pragma locate(ctl = 0x8000);

#pragma locate(contrl = 0x8003);

#define Astep 0x01 #define Bstep 0x02

#define Cstep 0x04 #define Dstep 0x08

unsigned char dly\_c; void delay()

{

unsigned char tt,cc;

cc = dly\_c; tt = 0x0; do{

do {

}while(--tt);

}while(--cc);

}

void main()

{

unsigned char mode;

contrl = mode8255; mode = 3;

ctl = 0; dly\_c = 0x10;

// 单/双八拍工作方式if(mode ==1) while(1)

{

ctl = Astep; delay();

ctl = Astep+Bstep; delay();

ctl = Bstep; delay();

ctl = Bstep+Cstep; delay();

ctl = Cstep; delay();

ctl = Cstep+Dstep; delay();

ctl = Dstep; delay();

ctl = Dstep+Astep; delay();

if(dly\_c>2) dly\_c --;

};

// 双四拍工作方式

if(mode == 2)

while(1)

{

ctl = Astep+Bstep; delay();

ctl = Bstep+Cstep; delay();

ctl = Cstep+Dstep; delay();

ctl = Dstep+Astep; delay();

if(dly\_c>3) dly\_c --;

};

// 单四拍工作方式if(mode ==3) while(1)

{

ctl = Dstep; delay(); ctl = Cstep; delay(); ctl = Bstep; delay(); ctl = Astep; delay();

if(dly\_c>4) dly\_c --;

}

while(1);

}

//硬件实验二十三 温度传感器试验 H23.C #define LEDLen 6

unsigned char volatile OUTBIT; // 位控制口unsigned char volatile OUTSEG; // 段控制口unsigned char volatile IN ; // 键盘读入口#pragma locate(OUTBIT = 0x8002)

#pragma locate(OUTSEG = 0x8004) #pragma locate(IN = 0x8001)

unsigned char LEDBuf[LEDLen]; // 显示缓冲

unsigned char const LEDMAP[] = { // 八段管显示码0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71

};

void Delay(unsigned char CNT)

{

unsigned char i;

while (CNT-- !=0)

for (i=100; i !=0; i--);

}

void DisplayLED()

{

unsigned char i, j, k; unsigned char Pos; unsigned char LED;

Pos = 0x20; // 从左边开始显示

for (i = 0; i < LEDLen; i++) { OUTBIT = 0; // 关所有八段管LED = LEDBuf[i];

OUTSEG = LED;

OUTBIT = Pos; // 显示一位八段管

Delay(1);

Pos >>= 1; // 显示下一位

}

OUTBIT = 0; // 关所有八段管

}

#include <kc\_sfrs.h>

#define LowTemp (-99) // A/D 0

#define HighTemp 99 // A/D 255

unsigned char ADPort;

#pragma locate (ADPort = 0xa000) signed char CurTemp;

void DisplayResult()

{

signed char T;

T = CurTemp; if (T < 0) {

LEDBuf[0] = 0x40; T = - T;

} else LEDBuf[0] = 0; LEDBuf[1] = LEDMAP[T / 10]; LEDBuf[2] = LEDMAP[T 10];

LEDBuf[3] = 0;

}

unsigned char ReadAD()

{

unsigned int i;

ADPort = 0; // start A/D convert

for (i=0; i<20; i++) ; // delay 100us return (0xff-ADPort);

}

void ReadTemp()

{

unsigned char i; signed int Temp;

Temp = 0;

for (i=0; i<16; i++) { Temp += ReadAD();

}

CurTemp = (Temp/16) \* (HighTemp-LowTemp) / 256 + LowTemp;

}

void main()

{

unsigned char dtimer;

while (1) {

for(dtimer=15; dtimer!=0; dtimer--){

// 当前温度和设定温度送显示缓冲

DisplayResult();

// 显示当前温度和设定温度

DisplayLED();

};

// 读入当前温度

ReadTemp();

}

}

//硬件实验二十四 液晶显示控制电路 H24.C #include <kc\_sfrs.h>

#define PD1 61 // 122/2 分成左右两半屏(122x32)

unsigned char Column;

unsigned char Page\_ ; // 页地址寄存器 D1,DO:页地址unsigned char Code\_ ; // 字符代码寄存器

unsigned char Command; // 指令寄存器

unsigned char LCDData; // 数据寄存器

unsigned char volatile CWADD1; unsigned char volatile DWADD1; unsigned char volatile CRADD1; unsigned char volatile DRADD1; unsigned char volatile CWADD2;

unsigned char volatile DWADD2; unsigned char volatile CRADD2; unsigned char volatile DRADD2;

#pragma locate(CWADD1 = 0x8000) // 写指令代码地址(E1) #pragma locate(DWADD1 = 0x8001) // 写显示数据地址(E1) #pragma locate(CRADD1 = 0x8002) // 读状态字地址(E1) #pragma locate(DRADD1 = 0x8003) // 读显示数据地址(E1)

#pragma locate(CWADD2 = 0x8004) // 写指令代码地址(E2) #pragma locate(DWADD2 = 0x8005) // 写显示数进地址(E2) #pragma locate(CRADD2 = 0x8006) // 读状态字地址(E2) #pragma locate(DRADD2 = 0x8007) // 读显示数据地址(E2)

//英文字模库 8x8 点阵

unsigned char const EETAB[][8]={

{0x38,0x44,0x44,0x44,0x44,0x44,0x38,0x00}, // 0

{0x10,0x30,0x50,0x10,0x10,0x10,0x7c,0x00}, // 1

{0x38,0x44,0x04,0x08,0x10,0x20,0x7c,0x00}, // 2

{0x78,0x84,0x04,0x38,0x04,0x84,0x78,0x00}}; // 3

// 中文字模库 16x16 点阵unsigned char const CCTAB[][32] ={

{0x00,0x04,0xe4,0x24,0x24,0x64,0xb4,0x2f, // 南

0x24,0xa4,0x64,0x24,0x24,0xe6,0x04,0x00,

0x00,0x00,0x7f,0x04,0x05,0x05,0x05,0x7f,

0x05,0x05,0x05,0x25,0x44,0x3f,0x00,0x00},

{0x00,0x04,0x04,0xe4,0x24,0x24,0x25,0x26, // 京

0x24,0x24,0x24,0xe4,0x06,0x04,0x00,0x00,

0x00,0x20,0x10,0x19,0x0D,0x41,0x81,0x7f,

0x01,0x01,0x05,0x0d,0x38,0x10,0x00,0x00},

{0x00,0x80,0x60,0xf8,0x07,0x0a,0x28,0x28, // 伟

0x28,0xff,0x28,0x28,0x28,0x28,0x00,0x00,

0x01,0x00,0x00,0xfF,0x00,0x01,0x01,0x01,

0x01,0xff,0x01,0x21,0x41,0x3f,0x00,0x00},

{0x08,0x08,0x89,0xce,0x28,0x18,0x02,0x7a, // 福0x4a,0x4a,0x4a,0x4a,0x4a,0x7a,0x02,0x00, 0x02,0x01,0x00,0xff,0x01,0x02,0xff,0x49, 0x49,0x49,0x7f,0x49,0x49,0x49,0xff,0x00}};

// 1.写指令代码子程序(E1) void WriteCommandE1()

{

while(CRADD1 & 0x80); CWADD1 = Command;

}

// 2.写显示数据子程序(E1) void WriteDataE1()

{

while(CRADD1 & 0x80); DWADD1 = LCDData;

}

// 3.读显示数据子程序(E1) void ReadDataE1()

{

while(CRADD1 & 0x80); LCDData = DRADD1;

}

// 4.写指令代码子程序(E2) void WriteCommandE2()

{

while(CRADD2 & 0x80); CWADD2 = Command;

}

// 5.写显示数据子程序(E2) void WriteDataE2()

{

while(CRADD2 & 0x80); DWADD2 = LCDData;

}

// 6.读显示数据子程序(E2) void ReadDataE2()

{

while(CRADD2 & 0x80); LCDData = DRADD2;

}

// 初始化程序

void Init()

{

Command = 0xe2; WriteCommandE1(); WriteCommandE2();

Command = 0xa4; WriteCommandE1(); WriteCommandE2();

Command = 0xa9; WriteCommandE1(); WriteCommandE2();

Command = 0xa0; WriteCommandE1(); WriteCommandE2();

Command = 0xc0; WriteCommandE1(); WriteCommandE2();

Command = 0xaf;

WriteCommandE1();

WriteCommandE2();

}

// 清屏

void Clear()

{

unsigned char i,j;

i = 0;

do {

Command = (i + 0xb8); // 页地址设置

WriteCommandE1(); WriteCommandE2();

Command = 0x00; // 列地址设置为"0" WriteCommandE1();

WriteCommandE2();

j = 0x50; // 一页清 80 个字节

do {

LCDData = 0x00; // 显示数据为"0" WriteDataE1();

WriteDataE2();

}while(--j !=0); // 页内字节清零循环

}while(++i !=4); // 页地址暂存器加一

// 显示区清零循环

}

// 延时程序

void Delay()

{

unsigned char i,j; i = 20;

j = 0;

do{

do{

}while(--j !=0);

}while( --i != 0 );

}

// 中文显示子程序

void WriteCHN16x16()

{

unsigned char i,j,k;

i = 0;

j = 0;

while(j<2) {

Command = ((Page\_ + j) & 0x03) | 0xb8; // 设置页地址

WriteCommandE1(); WriteCommandE2();

k = Column; // 列地址值

while(k < Column + 16){

if (k < PD1) { // 为左半屏显示区域(E1) Command = k;

WriteCommandE1(); // 设置列地址值LCDData = CCTAB[Code\_][i]; // 取汉字字模数据WriteDataE1(); // 写字模数据

} else{ // 为右半屏显示区域(E2) Command = k-PD1;

WriteCommandE2(); // 设置列地址值LCDData = CCTAB[Code\_][i]; // 取汉字字模数据WriteDataE2(); // 写字模数据

};

i++;

if( ++k >= PD1 \* 2) break; // 列地址是否超出显示范围

} ; j++;

};

}

// 中文显示程序

void main()

{

Init();

Clear();

Page\_ = 0x02; Column = 0x10; Code\_ = 0x00; WriteCHN16x16();

Page\_ = 0x02; Column = 0x20; Code\_ = 0x01; WriteCHN16x16();

Page\_ = 0x02; Column = 0x30; Code\_ = 0x02; WriteCHN16x16();

Page\_ = 0x02; Column = 0x40;

Code\_ = 0x03; WriteCHN16x16();

while(1);

}

// 硬件实验二十五 电子琴 H25.C #include <kc\_sfrs.h>

#include <kc\_funcs.h>

unsigned char volatile OUTBIT; // 位控制口unsigned char volatile IN ; // 键盘读入口#pragma locate(OUTBIT = 0x8002)

#pragma locate(IN = 0x8001)

unsigned char const KeyTable[] = { // 键码定义

0x16, 0x15, 0x14, 0xff, 0x13, 0x12, 0x11, 0x10,

0x0d, 0x0c, 0x0b, 0x0a, 0x0e, 0x03, 0x06, 0x09,

0x0f, 0x02, 0x05, 0x08,

0x00, 0x01, 0x04, 0x07

};

unsigned char TestKey()

{

OUTBIT = 0; // 输出线置为 0

return (~IN & 0x0f); // 读入键状态 (高四位不用)

}

unsigned char GetKey()

{

unsigned char Pos; unsigned char i; unsigned char j; unsigned char k;

i = 6;

Pos = 0x20; // 找出键所在列

do {

OUTBIT = ~ Pos;

Pos >>= 1;

k = ~IN & 0x0f;

} while ((--i != 0) && (k == 0));

// 键值 = 列 X 4 + 行

if (k != 0) { i \*= 4;

if (k & 2) i += 1;

else if (k & 4) i += 2;

else if (k & 8) i += 3;

OUTBIT = 0;

return(KeyTable[i]); // 取出键码

} else return(0xff);

}

unsigned int const ToneTable[8] =

{ 64968, 65029, 65084, 65110, 65157, 65191, 65235, 65252};

#define Speaker 1 // P1.0 unsigned char PulseCNT; unsigned char P1Buf; unsigned int Tone;

#pragma interrupt(T1Int=0) void T1Int()

{

wsr = 15; timer1 = Tone; wsr = 0;

ioport1 = P1Buf;

P1Buf ^= Speaker;

PulseCNT--;

}

void main()

{

unsigned char Key;

P1Buf = ioport1; P1Buf &= ~Speaker; ioport1 = P1Buf;

ioc1 = 0x04;

int\_mask = 0x01; // 打开定时器中断允许位

while (1) {

while (! TestKey()) ; // 等按键

Key = GetKey();

if ( (Key >= 1) && (Key <= 8) ) { Tone = ToneTable[Key-1] ;

wsr = 15; timer1 = Tone; wsr = 0;

enable(); // 打开总中断允许位

PulseCNT = 100;

while (PulseCNT != 0) ; // 发 100 个脉冲

disable(); // 关闭总中断允许位

}

}

}

//硬件实验二十六 空调温度控制实验 H26.C #define LEDLen 6

unsigned char volatile OUTBIT; // 位控制口unsigned char volatile OUTSEG; // 段控制口unsigned char volatile IN ; // 键盘读入口#pragma locate(OUTBIT = 0x8002)

#pragma locate(OUTSEG = 0x8004) #pragma locate(IN = 0x8001)

unsigned char LEDBuf[LEDLen]; // 显示缓冲

unsigned char const LEDMAP[] = { // 八段管显示码0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71

};

void Delay(unsigned char CNT)

{

unsigned char i;

while (CNT-- !=0)

for (i=100; i !=0; i--);

}

void DisplayLED()

{

unsigned char i, j, k; unsigned char Pos; unsigned char LED;

Pos = 0x20; // 从左边开始显示

for (i = 0; i < LEDLen; i++) { OUTBIT = 0; // 关所有八段管LED = LEDBuf[i];

OUTSEG = LED;

OUTBIT = Pos; // 显示一位八段管

Delay(1);

Pos >>= 1; // 显示下一位

}

OUTBIT = 0; // 关所有八段管

}

unsigned char const KeyTable[] = { // 键码定义

0x16, 0x15, 0x14, 0xff, 0x13, 0x12, 0x11, 0x10,

0x0d, 0x0c, 0x0b, 0x0a, 0x0e, 0x03, 0x06, 0x09,

0x0f, 0x02, 0x05, 0x08,

0x00, 0x01, 0x04, 0x07

};

unsigned char TestKey()

{

OUTBIT = 0; // 输出线置为 0

return (~IN & 0x0f); // 读入键状态 (高四位不用)

}

unsigned char GetKey()

{

unsigned char Pos; unsigned char i; unsigned char j; unsigned char k;

i = 6;

Pos = 0x20; // 找出键所在列

do {

OUTBIT = ~ Pos;

Pos >>= 1;

k = ~IN & 0x0f;

} while ((--i != 0) && (k == 0));

// 键值 = 列 X 4 + 行

if (k != 0) { i \*= 4;

if (k & 2) i += 1;

else if (k & 4) i += 2;

else if (k & 8) i += 3;

OUTBIT = 0;

do Delay(10); while (TestKey()); // 等键释放

return(KeyTable[i]); // 取出键码

} else return(0xff);

}

/\* ============================================================================\*/

#include <kc\_sfrs.h>

#define UP 0x16 #define DOWN 0x15

#define ON 1

#define OFF 0

#define LowLimit 10

#define HighLimit 30

#define LowTemp (-40) // A/D 0

#define HighTemp 50 // A/D 255

unsigned char ADPort;

#pragma locate (ADPort = 0x8000) #define Heat 1 // P1.0

#define Cool 2 // P1.1 signed char P1Buf;

signed char SetTemp, CurTemp; void DisplayResult()

{

signed char T;

T = CurTemp; if (T < 0) {

LEDBuf[0] = 0x40; T = - T;

} else LEDBuf[0] = 0; LEDBuf[1] = LEDMAP[T / 10]; LEDBuf[2] = LEDMAP[T 10];

LEDBuf[3] = 0;

LEDBuf[4] = LEDMAP[SetTemp / 10]; LEDBuf[5] = LEDMAP[SetTemp 10];

}

unsigned char ReadAD()

{

unsigned int i;

ADPort = 0; // start A/D convert

for (i=0; i<20; i++) ; // delay 100us return (ADPort);

}

void ReadTemp()

{

unsigned char i; signed int Temp;

Temp = 0;

for (i=0; i<16; i++) { Temp += ReadAD();

}

CurTemp = (Temp/16) \* (HighTemp-LowTemp) / 256 + LowTemp;

}

void main()

{

unsigned char Key;

P1Buf = ioport1;

SetTemp = 20;

while (1) {

while (!TestKey()) { DisplayResult(); DisplayLED(); ReadTemp();

if (CurTemp < (SetTemp - 2)) { P1Buf |= Heat;

P1Buf &= ~Cool;

} else if (CurTemp > (SetTemp + 2)) { P1Buf &= ~Heat;

P1Buf |= Cool;

} else {

P1Buf &= ~Heat;

P1Buf &= ~Cool;

}

ioport1 = P1Buf;

}

Key = GetKey();

if (Key == DOWN) {

if (SetTemp > LowLimit) SetTemp--;

} else if (Key == UP) {

if (SetTemp < HighLimit) SetTemp++;

}

}

}

// 硬件实验二十七 计算器实验 H27.C #define LEDLen 6

unsigned char volatile OUTBIT; // 位控制口unsigned char volatile OUTSEG; // 段控制口unsigned char volatile IN ; // 键盘读入口#pragma locate(OUTBIT = 0x8002)

#pragma locate(OUTSEG = 0x8004) #pragma locate(IN = 0x8001)

unsigned char LEDBuf[LEDLen]; // 显示缓冲

unsigned char const LEDMAP[] = { // 八段管显示码0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71

};

void Delay(unsigned char CNT)

{

unsigned char i;

while (CNT-- !=0)

for (i=100; i !=0; i--);

}

void DisplayLED()

{

unsigned char i, j, k; unsigned char Pos; unsigned char LED;

Pos = 0x20; // 从左边开始显示

for (i = 0; i < LEDLen; i++) { OUTBIT = 0; // 关所有八段管LED = LEDBuf[i];

OUTSEG = LED;

OUTBIT = Pos; // 显示一位八段管

Delay(1);

Pos >>= 1; // 显示下一位

}

}

unsigned char const KeyTable[] = { // 键码定义

0x16, 0x15, 0x14, 0xff, 0x13, 0x12, 0x11, 0x10,

0x0d, 0x0c, 0x0b, 0x0a, 0x0e, 0x03, 0x06, 0x09,

0x0f, 0x02, 0x05, 0x08,

0x00, 0x01, 0x04, 0x07

};

unsigned char TestKey()

{

OUTBIT = 0; // 输出线置为 0

return (~IN & 0x0f); // 读入键状态 (高四位不用)

}

unsigned char GetKey()

{

unsigned char Pos; unsigned char i; unsigned char j; unsigned char k;

i = 6;

Pos = 0x20; // 找出键所在列

do {

OUTBIT = ~ Pos;

Pos >>= 1;

k = ~IN & 0x0f;

} while ((--i != 0) && (k == 0));

// 键值 = 列 X 4 + 行

if (k != 0) { i \*= 4;

if (k & 2) i += 1;

else if (k & 4) i += 2;

else if (k & 8) i += 3;

OUTBIT = 0;

do Delay(10); while (TestKey()); // 等键释放

return(KeyTable[i]); // 取出键码

} else return(0xff);

}

#define ADD 0x0a #define SUB 0x0b #define MUL 0x0c #define DIV 0x0d #define EQU 0x0e #define CLR 0x0f

void DisplayResult(signed int Result)

{

unsigned char i;

if (Result >= 0) { LEDBuf[0] = 0;

} else {

LEDBuf[0] = 0x40;

Result = - Result;

}

LEDBuf[1] = 0;

LEDBuf[2] = 0;

LEDBuf[3] = 0;

LEDBuf[4] = 0;

LEDBuf[5] = LEDMAP[Result 10]; Result = Result / 10; i = 4;

while (Result != 0) {

LEDBuf[i--] = LEDMAP[Result 10]; Result = Result / 10;

}

}

void main()

{

signed int Last, Result; unsigned char OP; unsigned char Key;

Last = 0;

Result = 0; OP = ADD;

DisplayResult(Result); while (1) {

while (!TestKey()) DisplayLED(); Key = GetKey();

if ((Key >= 0) && (Key <= 9)) { Result = Result\*10 + Key; DisplayResult(Result);

} else if (Key == CLR) { Last = 0;

Result = 0; OP = ADD;

DisplayResult(Result);

} else if ( (Key==ADD) || (Key==SUB) || (Key==MUL) || (Key==DIV) || (Key=EQU) ) { switch (OP) {

case ADD:

Result = Last + Result; break;

case SUB:

Result = Last - Result; break;

case MUL:

Result = Last \* Result; break;

case DIV:

Result = Last / Result; break;

}

if (Key == EQU) { Last = 0;

OP = ADD;

} else {

Last = Result;

OP = Key;

}

DisplayResult(Result);

Result = 0;

}

}

}

// 硬件实验二十八 用 HSO 方式输出 PWM 波形 H28.C

#include <kc\_sfrs.h> #include <kc\_funcs.h>

//定时常数

#define TimerC 2500

//高电平时间#define RatioH 1000

#pragma interrupt(pwm\_out=0x2006) void pwm\_out()

{

if(ios0 & 0x01)

{

hso\_command = 0x10; // 预置低电平

hso\_time = timer1 + RatioH; // 高电平时间宽度

}else

{

hso\_command = 0x30; // 预置高电平

hso\_time = timer1 + TimerC-RatioH; // 低电平时间宽度

}

}

void main ()

{

hso\_command = 0x30; hso\_time = timer1 + 250; int\_mask = 0x08; int\_pend = 0;

enable();

while(1);

}

// 硬件实验二十九 用 HSI 方式测量脉冲宽度 H29.C #include <kc\_sfrs.h>

unsigned int Counter; unsigned char status;

unsigned int first,second,result;

void main()

{

ioc0 = 1; // 允许 HSI.0 输入

hsi\_mode = 3; // HSI.0 方式 3 asm{ nop; nop; nop;};

ioport1 = 1; // P10 置高

ioport1 = 0; // P10 置低,产Th下降沿

Counter = 0x1234;

while(Counter) Counter --; // 延时

//for( Counter=0x1234; Counter>0; Counter--);

ioport1 = 1; // P10 置高

asm{ nop; nop; nop;};

|  |  |  |
| --- | --- | --- |
| status | = hsi\_status; | // 先读下降沿事件状态 |
| first | = hsi\_time; | // 再读下降沿的时间 |
| status | = hsi\_status; | // 先读上升沿事件状态 |
| second | = hsi\_time; | // 再读上升沿的时间 |

result = second - first; // 相减得脉冲宽度

while(1); // 在此行设断点

}

//硬件实验三十 用 HSI 中断方式统计脉冲个数 H30.C #include <kc\_sfrs.h>

unsigned int Counter; unsigned int Result;

#pragma interrupt(hsi1\_int = 0x2004) // HSI 事件中断

#pragma interrupt(hsi0\_int = 0x2008) // HSI.0 脚中断

void hsi0\_int()

{

Result = Counter; // 读取计数结果

}

void hsi1\_int()

{

unsigned char status; unsigned int hsi\_t;

status = hsi\_status; // 读取 HSI 状态hsi\_t = hsi\_time; // 读取 HSI 时间if(status & 0x10) // 如果是 HSI.2 事件

Counter ++; // 计数加 1

}

void main()

{

ioc1 = 0x6f; // HSI 中断且 HSI.2 输入

hsi\_mode = 0; // HSI 方式 0, 8 次跳变为一事件

int\_mask = 0x10; // HSI.0 脚中断

int\_mask|= 0x04; // HSI 事件中断

ioc0 = 0x10; // 允许 HSI.2 输入

int\_pend = 0;

Counter = 0;

Result = 0;

enable();

while(!Result); // 等待结果

while(1);

}

// 硬件实验三十一 计数器实验 H31.C #include <kc\_sfrs.h>

void main()

{

unsigned int TempT2; timer2 = 0;

while (1) {

TempT2 = timer2; // 读取 T2CLK 脚上的上升/下降沿ioport1 = TempT2; // 将记数结果送 P1 口

}

}

// 硬件实验三十二 片内 A/D 转换实验 H32.C

#include <kc\_sfrs.h>

#define mode 0x82 // 8255 方式 0，PA，PC 输出，PB 输入

unsigned char PortA; unsigned char CAddr;

#pragma locate(CAddr = 0x8003) // 控制字地址

#pragma locate(PortA = 0x8000)

unsigned int result; void main()

{

unsigned char i;

CAddr = mode; // 输出 8255 控制字while(1)

{

ad\_command = 0x08; // 启动 A/D 转换

asm{

nop; nop; nop; nop; // 延时，

};

while(ad\_result\_lo & 0x08); // 是否完成转换

i = ad\_result\_hi; // 转换结果高 8 位

PortA = i; // 输出到 LED

result = i << 2; // 转换结果高字节

result+= ad\_result\_lo >> 6; // 转换结果低字节

}

}

# 附录 E 8088 汇编程序清单

; 软件实验一 存储器块清零 S1.ASM

0000 data segment

0000 0100\*(55) Block db 256 dup(55h)

0100 data ends

0000 code segment

assume cs:code, ds:data

|  |  |  |
| --- | --- | --- |
| 0000 |  | start proc near |
| 0000 | B8 0000s | mov ax, data |
| 0003 | 8E D8 | mov ds, ax |
| 0005 | BB 0000r | mov bx, offset Block ; 起始地址 |
| 0008  000B  000B | B9 0100  C6 07 00 | mov cx, 256 ; 清 256 字节  Again:  mov [bx], byte ptr 0 |
| 000E | 43 | inc bx |
| 000F | E2 FA | Loop Again ; 记数减一 |
| 0011 | EB FE | jmp $ |
| 0013 |  | code ends  end start |

; 软件实验二 二进制到 BCD 码转换 S2.ASM

; 将 AX 拆为三个 BCD 码, 并存入 Result 开始的叁个单元

0000 data segment

|  |  |  |  |
| --- | --- | --- | --- |
| 0000 | 03\*(??) | | Result db 3 dup(?) |
| 0003 |  | | data ends |
| 0000 |  | | code segment  assume cs:code, ds:data |
| 0000 |  |  | start proc near |
| 0000 | B8 | 0000s | mov ax, data |
| 0003 | 8E | D8 | mov ds, ax |
| 0005 | B8 | 007B | mov ax, 123 |
| 0008 | B1 | 64 | mov cl, 100 |
| 000A | F6 | F1 | div cl |
| 000C | A2 | 0000r | mov Result, al |
| 000F | 8A | C4 | mov al, ah |
| 0011 | B4 | 00 | mov ah, 0 |
| 0013 | B1 | 0A | mov cl, 10 |
| 0015 | F6 | F1 | div cl |
| 0017 | A2 | 0001r | mov Result+1, al |
| 001A | 88 | 26 0002r | mov Result+2, ah |
| 001E | EB | FE | jmp $ |
| 0020 |  |  | code ends  end start |

; 除以 100, 得百位数

; 余数除以 10, 得十位数

; 余数为个位数

; 软件实验三 二进制到 ASCII 码转换 S3.ASM

; 将 AX 拆为二个 ASCII 码, 并存入 Result 开始的二个单元

0000 data segment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0000 | 02\*(??) | | Result db 2 dup(?) | |
| 0002 |  | | ASCIITab: | |
| 0002 | 30 | 31 32 33 | 34 35 | 36+ db "0123456789ABCDEF" ;定义数字对应的 ASCII 表 |
|  | 37 | 38 39 41 | 42 43 | 44+ |
|  | 45 | 46 |  |  |

0012 data ends

0000 code segment

assume cs:code, ds:data

0000 start proc near 0000 B8 0000s mov ax, data

0003 8E D8 mov ds, ax

0005 B8 001A mov ax, 1ah

0008 8A D8 mov bl, al

000A D0 EB D0 EB D0 EB D0+ shr bl, 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0012 | EB  B7 | 00 |  | mov | bh, 0 |
| 0014 | 8A | A7 | 0002r | mov | ah, byte ptr ASCIITab[bx] |
| 0018 | 88 | 26 | 0000r | mov | Result, ah |
| 001C  001E  0020 | 24  8A  8A | 0F  D8 A7 | 0002r | and  mov mov | al, 0fh bl,al  ah, byte ptr ASCIITab[bx] |
| 0024 | 88 | 26 | 0001r | mov | Result+1,ah |
| 0028 | EB | FE |  | jmp | $ |
| 002A |  |  |  | code | ends |

end start

; 软件实验四 内存块移动 S4.ASM

; 移动 3000H --> 4000H, 256 字节

0000 data segment

0000 0100\*(55) Source db 256 dup(055h)

0100 0100\*(AA) Target db 256 dup(0aah)

0200 data ends

0000 code segment

assume cs:code, ds:data, es:data

0000 start proc near 0000 B8 0000s mov ax, data

0003 8E D8 mov ds, ax

0005 8E C0 mov es, ax

0007 BE 0000r mov si, offset Source

000A BF 0100r mov di, offset Target

000D B9 0100 mov cx, 256

0010 F3> A4 rep movsb

0012 EB FE jmp $

0014 code ends

end start

; 软件实验五 程序跳转表 S5.ASM 0000 code segment

assume cs:code

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0000 |  |  | Func0 | proc near |
| 0000 | B8 | 0000 | mov | ax, 0 |
| 0003 | C3 |  | ret |  |
| 0004 |  |  | Func0 | endp |
| 0004 |  |  | Func1 | proc near |
| 0004 | B8 | 0001 | mov | ax, 1 |
| 0007 | C3 |  | ret |  |
| 0008 |  |  | Func1 | endp |
| 0008 |  |  | Func2 | proc near |
| 0008  000B  000C | B8 C3 | 0002 | mov ret  Func2 | ax, 2  endp |
| 000C  000C | B8 | 0003 | Func3  mov | proc near ax, 3 |
| 000F  0010 | C3 |  | ret Func3 | endp |

0010 FuncEnter proc near

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0010 | 3D | 0000 | cmp | ax,0 |
| 0013 | 75 | 02 | jnz | Next1 |
| 0015 | EB | E9 | jmp | Func0 |
| 0017 |  |  | Next1: |  |
| 0017 | 3D | 0001 | cmp | ax,1 |
| 001A | 75 | 02 | jnz | Next2 |
| 001C  001E  001E | EB  3D | E6  0002 | jmp Next2:  cmp | Func1  ax,2 |
| 0021 | 75 | 02 | jnz | Next3 |
| 0023 | EB | E3 | jmp | Func2 |
| 0025 |  |  | Next3: |  |
| 0025 | 3D | 0003 | cmp | ax,3 |
| 0028 | 75 | 02 | jnz | Next4 |
| 002A  002C  002C | EB  C3 | E0 | jmp Next4:  ret | Func3 |
| 002D |  |  | FuncEnter endp | |
| 002D |  |  | start proc near | |
| 002D | B8 | 0000 | mov ax, 0 | |
| 0030 | E8 | FFDD | call FuncEnter | |
| 0033 | B8 | 0001 | mov ax, 1 | |
| 0036 | E8 | FFD7 | call FuncEnter | |
| 0039 | B8 | 0002 | mov ax, 2 | |
| 003C | E8 | FFD1 | call FuncEnter | |
| 003F | B8 | 0003 | mov ax, 3 | |
| 0042 | E8 | FFCB | call FuncEnter | |
| 0045 | EB | FE | jmp $ | |
| 0047 |  |  | code ends  end start | |

; 软件实验六 数据排序 S6.ASM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0000 | =000A | Len  data | equ  segment | 10 |
| 0000 | 05 02 01 | 00 02 03 08+ Array db 5,2,1,0,2,3,8,6,5,9 | | |
|  | 06 05 09 |  | | |
| 000A  000B | 00 | Change db 0 data ends | | |
| 0000 |  | code segment | | |

assume cs:code, ds:data

0000 start proc near 0000 B8 0000s mov ax, data

0003 8E D8 mov ds, ax 0005 Sort:

0005 BB 0000r mov bx, offset Array

0008 B9 0009 mov cx, Len-1

000B C6 06 000Ar 00 mov Change, 0

0010 Goon:

0010 8A 07 mov al, byte ptr [bx]

0012 43 inc bx

0013 3A 07 cmp al, byte ptr [bx]

0015 7E 0C jng Next ; 前小后大, 不交换

0017 C6 06 000Ar 01 mov Change, 1 ; 前大后小, 置交换标志

001C 8A 27 mov ah, [bx]

001E 88 07 mov [bx] ,al ; 交换

0020 88 67 FF mov [bx-1], ah

0023 Next:

0023 E2 EB loop Goon

0025 80 3E 000Ar 00 cmp Change, 0 002A 75 D9 jne Sort

002C EB FE jmp $ 002E code ends

end start

; 硬件实验三 用 74LS245 扩展 I/O 口实验 H3.ASM

=8000 CS245 equ 08000h

0000 code segment assume cs:code

0000 start proc near

0000 Again:

0000 BA 8000 mov dx, CS245

0003 EC in al, dx

0004 EB FA jmp Again 0006 code ends

end start

; 硬件实验四 用 74LS273 扩展 I/O 口实验 H4.ASM

=8000 CS273 equ 08000h

0000 code segment assume cs:code

|  |  |  |
| --- | --- | --- |
| 0000 |  | start proc near |
| 0000 | B9 0008 | mov cx, 8 |
| 0003 | B0 01 | mov al, 1 |
| 0005 |  | Again: |
| 0005 | BA 8000 | mov dx, CS273 |
| 0008 | EE | out dx,al |
| 0009 | D0 E0 | shl al, 1 |
| 000B | E2 F8 | loop Again |
| 000D | EB FE | jmp $ |
| 000F |  | code ends |

end start

; 硬件实验七 8255 输入、输出实验 H7.ASM

=0082 mode equ 082h ; 方式 0，PA，PC 输出，PB 输入

=8000 PortA equ 8000h ; Port A

=8001 PortB equ 8001h ; Port B

=8002 PortC equ 8002h ; Port C

=8003 CAddr equ 8003h ; 控制字地址

0000 code segment assume cs:code

0000 start proc near

0000 Start:

0000 B0 82 mov al, mode

0002 BA 8003 mov dx, CAddr

0005 EE out dx, al ; 输出控制字

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ;实验 1：PortA 输出 | | | | | |
| 0006 | B0 | 80 | mov | al, 80H | |
| 0008 | B9 | 0008 | mov | cx, 08H | |
| 000B |  |  | OutA: |  | |
| 000B | BA | 8000 | mov | dx, PortA | |
| 000E | EE |  | out | dx, al ; 输出 PortA | |
| 000F | BA | 8002 | mov | dx, PortC | |
| 0012 | EE |  | out | dx, al | |
| 0013 | D0 | E8 | shr | al, 1 ; 移位 | |
| 0015 | B4 | 64 | mov | ah,100 | |
| 0017 | E8 | 0015 | call | delay | ; 延时 |
| 001A | E2 | EF | loop | OutA |  |
| ;实验 2： PortB 输入PortA 输出 | | | | | |
| 001C | BA | 8001 | mov | dx, PortB |  |
| 001F | EC |  | in | al, dx | ; 读入 PortB |
| 0020 | BA | 8002 | mov | dx, PortC |  |
| 0023 | EE |  | out | dx, al |  |
| 0024 | BA | 8000 | mov | dx, PortA |  |
| 0027 | EE |  | out | dx, al | ; 输出到 PortA |
| 0028 | B4 | C8 | mov | ah, 200 |  |
| 002A | E8 | 0002 | call delay | | |
| 002D | EB | D1 | jmp Start | | |
| 002F |  |  | endp | | |

002F delay proc near 002F 50 push ax

0030 B0 00 mov al,0

0032 51 push cx

0033 8B C8 mov cx,ax

0035 E2 FE loop $

0037 59 pop cx

0038 58 pop ax

0039 C3 ret

003A delay endp

003A code ends

end start

; 硬件实验十三 D/A 转换实验 H13.ASM

=8000 CS0832 equ 08000h

0000 code segment assume cs:code

0000 start proc near 0000 B0 00 mov al, 0

0002 BA 8000 mov dx, CS0832

0005 EE out dx, al

0006 B0 80 mov al, 80h

0008 EE out dx, al

0009 B0 FF mov al, 0ffh

000B EE out dx, al

000C EB FE jmp $ 000E start endp

000E code ends

end start

; 硬件实验十四 A/D 转换实验 H14.ASM

=0082 mode equ 082h

=9000 PA equ 09000h

=9003 CTL equ 09003h

=8000 CS0809 equ 08000h

0000 code segment assume cs:code

0000 start proc near

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0000 | B0 82 | mov | al, | mode |  | |
| 0002 | BA 9003 | mov | dx, | CTL |  |  |
| 0005 | EE | out | dx, | al |  |  |
| 0006 |  | again: |  |  |  |  |
| 0006 | B0 00 | mov | al, | 0 |  |  |
| 0008 | BA 8000 | mov | dx, | CS0809 |  |  |
| 000B  000C | EE  B9 0040 | out  mov | dx,  cx, | al  40h | ; | 起动 A/D |

000F E2 FE loop $ ; 延时 > 100us

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0011 | EC | in | al, | dx | ; | 读入结果 |
| 0012 | BA 9000 | mov | dx, | PA |  |  |
| 0015 | EE | out | dx, | al |  |  |

0016 EB EE jmp again 0018 code ends

end start

; 硬件实验十六 八段数码管显示实验 H16.ASM

=8002 OUTBIT equ 08002h ; 位控制口

=8004 OUTSEG equ 08004h ; 段控制口

|  |  |  |
| --- | --- | --- |
| 0000 | data segment | |
| 0000 | 06\*(??) LEDBuf db 6 dup(?) ; 显示缓冲 | |
| 0006 | 01\*(??) Num db 1 dup(?) ; 显示的数据 | |
| 0007 | 01\*(??) DelayT db 1 dup(?) | |
| 0008 | LEDMAP: ; 八段管显示码 | |
| 0008  0010 | 3F 06 5B 4F 66 6D 7D+ db 3fh, 06h, 5bh, 4fh, 66h, 6dh, 7dh, 07h 07  7F 6F 77 7C 39 5E 79+ db 7fh, 6fh, 77h, 7ch, 39h, 5eh, 79h, 71h | |
| 0018 | 71  data ends | |
| 0000 | code segment | |
|  | assume cs:code, ds:data | |
| 0000 |  | Delay proc near |
| 0000 | 50 | push ax ; 延时子程序 |
| 0001 | 51 | push cx |

0002 B0 00 mov al, 0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0004 | 8B | C8 | mov |  | cx,ax |
| 0006 | E2 | FE | loop | $ |  |
| 0008 | 59 |  | pop cx | | |
| 0009  000A  000B | 58  C3 |  | pop ax ret  Delay endp | | |
| 000B |  |  | DisplayLED proc near | | |
| 000B | BB | 0000r | mov bx, offset LEDBuf | | |
| 000E | B1 | 06 | mov cl, 6 ; 共 6 个八段管 | | |
| 0010 | B4 | 20 | mov ah, 00100000b ; 从左边开始显示 | | |
| 0012 |  |  | DLoop: | | |
| 0012 | BA | 8002 | mov dx, OUTBIT | | |
| 0015 | B0 | 00 | mov al, 0 | | |
| 0017 | EE |  | out dx,al ; 关所有八段管 | | |
| 0018 | 8A | 07 | mov al, [bx] | | |
| 001A | BA | 8004 | mov dx, OUTSEG | | |
| 001D | EE |  | out dx,al | | |
| 001E | BA | 8002 | mov dx, OUTBIT | | |
| 0021 | 8A | C4 | mov al, ah | | |
| 0023 | EE |  | out dx, al ; 显示一位八段管 | | |
| 0024 | 50 |  | push ax | | |
| 0025 | B4 | 01 | mov ah, 1 | | |
| 0027  002A | E8 58 | FFD6 | call Delay pop ax | | |
| 002B  002D  002E | D0 43 FE | EC  C9 | shr ah, 1  inc bx  dec cl | | |
| 0030 | 75 | E0 | jnz DLoop | | |
| 0032 | BA | 8002 | mov dx, OUTBIT | | |
| 0035 | B0 | 00 | mov al, 0 | | |
| 0037 | EE |  | out dx,al ; 关所有八段管 | | |
| 0038 | C3 |  | ret | | |

0039 DisplayLED endp

0039 Start proc near

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0039 | B8 | 0000s | mov | ax, data |
| 003C | 8E | D8 | mov | ds, ax |
| 003E | C6 | 06 0006r 00 | mov | Num, 0 |
| 0043 |  |  | MLoop: |  |
| 0043 | FE | 06 0006r | inc | Num |
| 0047 | 8A | 2E 0006r | mov | ch,Num |
| 004B | B4 | 00 | mov | ah,0 |
| 004D | B1 | 06 | mov | cl,6 |
| 004F | BB | 0000r | mov | bx,offset LEDBuf |
| 0052 |  |  | FillBuf: |  |
| 0052 | BE | 0008r | mov | si, offset LEDMap |
| 0055 | 8A | C5 | mov | al,ch |
| 0057 | 24 | 0F | and | al,0fh |
| 0059 | 03 | C6 | add | ax,si |
| 005B | 8B | F0 | mov | si,ax |
| 005D | 8A | 04 | mov | al,[si] ; 数据转换成显示码 |
| 005F | 88 | 07 | mov | [bx], al ; 显示码存入显示缓冲 |
| 0061 | 43 |  | inc | bx |
| 0062 | FE | C5 | inc | ch |
| 0064 | FE | C9 | dec | cl |
| 0066 | 75 | EA | jnz | FillBuf |

0068 C6 06 0007r 14 mov DelayT,20

006D DispAgain:

006D E8 FF9B call DisplayLED ; 显示

0070 FE 0E 0007r dec DelayT

0074 75 F7 jnz DispAgain

0076 EB CB jmp MLoop 0078 Start endp

0078 code ends

end start

; 硬件实验十七 键盘扫描显示实验 H17.ASM

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| =8002  =8004  =8001 | | | | OUTBIT OUTSEG  IN\_KEY | | equ 08002h equ 08004h  equ 08001h | | ; 位控制口  ; 段控制口  ; 键盘读入口 | |
| 0000 |  | | | data segment | | | | | |
| 0000 | 06\*(??) | | | LEDBuf db 6 dup(?) ; 显示缓冲 | | | | | |
| 0006 | 01\*(??) | | | Num db 1 dup(?) ; 显示的数据 | | | | | |
| 0007 | 01\*(??) | | | DelayT db 1 dup(?) | | | | | |
| 0008 |  | | | LEDMAP: ; 八段管显示码 | | | | | |
| 0008  0010 | 3F  07  7F | 06  6F | 5B 4F 66 6D  77 7C 39 5E | | 7D+  79+ | db  db | | 3fh, 06h, 5bh, 4fh, 66h, 6dh,  7fh, 6fh, 77h, 7ch, 39h, 5eh, | 7dh, 07h  79h, 71h |
|  | 71 |  |  | |  |  | |  |  |
| 0018 |  |  |  | KeyTable: | | | ; 键码定义 | | |
| 0018 | 16 | 15 | 14 FF | db | | | 16h, 15h, 14h, 0ffh | | |
| 001C | 13 | 12 | 11 10 | db | | | 13h, 12h, 11h, 10h | | |
| 0020 | 0D | 0C | 0B 0A | db | | | 0dh, 0ch, 0bh, 0ah | | |
| 0024 | 0E | 03 | 06 09 | db | | | 0eh, 03h, 06h, 09h | | |
| 0028 | 0F | 02 | 05 08 | db | | | 0fh, 02h, 05h, 08h | | |
| 002C  0030 | 00 | 01 | 04 07 | db  data ends | | | 00h, 01h, 04h, 07h | | |

0000 code segment

assume cs:code, ds:data

|  |  |  |  |
| --- | --- | --- | --- |
| 0000 |  | Delay proc near |  |
| 0000 | 50 | push ax | ; 延时子程序 |
| 0001 | 51 | push cx |  |

0002 B0 00 mov al, 0

0004 8B C8 mov cx,ax

0006 E2 FE loop $

0008 59 pop cx

0009 58 pop ax

000A C3 ret

000B Delay endp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 000B |  |  | DisplayLED | proc near |
| 000B | BB | 0000r | mov | bx, offset LEDBuf |
| 000E | B1 | 06 | mov | cl, 6 ; 共 6 个八段管 |
| 0010 | B4 | 20 | mov | ah, 00100000b ; 从左边开始显示 |
| 0012 |  |  | DLoop: |  |
| 0012 | BA | 8002 | mov | dx, OUTBIT |
| 0015 | B0 | 00 | mov | al, 0 |
| 0017 | EE |  | out | dx,al ; 关所有八段管 |
| 0018 | 8A | 07 | mov | al, [bx] |
| 001A | BA | 8004 | mov | dx, OUTSEG |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 001D  001E  0021 | EE  BA 8A | 8002  C4 | out  mov mov | dx,al  dx, OUTBIT  al, ah |  |
| 0023 | EE |  | out | dx, al | ; 显示一位八段管 |
| 0024 | 50 |  | push | ax |  |
| 0025 | B4 | 01 | mov | ah, 1 |  |
| 0027  002A | E8 58 | FFD6 | call pop | Delay ax |  |
| 002B  002D  002E | D0 43 FE | EC  C9 | shr inc dec | ah, 1 bx cl |  |
| 0030 | 75 | E0 | jnz | DLoop |  |
| 0032 | BA | 8002 | mov | dx, OUTBIT |  |
| 0035 | B0 | 00 | mov | al, 0 |  |
| 0037 | EE |  | out | dx,al | ; 关所有八段管 |
| 0038 | C3 |  | ret |  |  |
| 0039 |  |  | DisplayLED | endp |  |
| 0039 TestKey proc near | | | | | |
| 0039 | BA | 8002 | mov | dx, OUTBIT |  |
| 003C | B0 | 00 | mov | al, 0 |  |
| 003E | EE |  | out | dx, al | ; 输出线置为 0 |
| 003F | BA | 8001 | mov | dx, IN\_KEY |  |
| 0042 | EC |  | in | al, dx | ; 读入键状态 |
| 0043 | F6 | D0 | not | al |  |
| 0045 | 24 | 0F | and | al, 0fh | ; 高四位不用 |
| 0047 | C3 |  | ret |  |  |
| 0048 |  |  | TestKey endp | | |
| 0048 |  |  | GetKey proc near | | |
| 0048 | B5 | 20 | mov ch, 00100000b | | |
| 004A  004C  004C | B1  BA | 06  8002 | mov cl, 6 KLoop:  mov dx, OUTBIT | | |
| 004F | 8A | C5 | mov al, ch ; 找出键所在列 | | |
| 0051 | F6 | D0 | not al | | |
| 0053 | EE |  | out dx, al | | |
| 0054 | D0 | ED | shr ch, 1 | | |
| 0056 | BA | 8001 | mov dx, IN\_KEY | | |
| 0059  005A | EC F6 | D0 | in al, dx  not al | | |
| 005C | 24 | 0F | and al, 0fh | | |
| 005E | 75 | 09 | jne Goon\_ ; 该列有键入 | | |
| 0060 | FE | C9 | dec cl | | |
| 0062 | 75 | E8 | jnz KLoop | | |

没有键按下, 返回 0ffh

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0064 | B1 | FF |  | mov | cl, 0ffh | ; |  |
| 0066 | EB | 15 | 90 | jmp | Exit1 |  |
| 0069 |  |  |  | Goon\_: |  |  |
| 0069  006B | FE D0 | C9 E1 | D0 E1 | dec shl | cl cl, 2 |  | ; |
| 006F  0071 | B5 | 04 |  | mov LoopC: | ch, 4 |  |  |
| 0071 | A8 | 01 |  | test | al, 1 |  |  |
| 0073 | 75 | 08 |  | jnz | Exit1 |  |  |
| 0075 | D0 | E8 |  | shr | al, 1 |  |  |

键值 = 列 X 4 + 行

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0077 | FE | C1 | inc | cl |
| 0079 | FE | CD | dec | ch |
| 007B  007D  007D | 75  BA | F4  8002 | jnz Exit1:  mov | LoopC  dx, OUTBIT |
| 0080 | B0 | 00 | mov | al, 0 |
| 0082 | EE |  | out | dx, al |
| 0083 | B5 | 00 | mov | ch, 0 |
| 0085 | BB | 0018r | mov | bx, offset KeyTable |
| 0088 | 03 | D9 | add | bx, cx |
| 008A | 8A | 07 | mov | al, [bx] ; 取出键码 |
| 008C | 8A | D8 | mov | bl, al |
| 008E |  |  | WaitRelease: | |
| 008E | BA | 8002 | mov dx, OUTBIT | |
| 0091 | B0 | 00 | mov al, 0 | |
| 0093 | EE |  | out dx, al ; 等键释放 | |
| 0094 | B4 | 0A | mov ah, 10 | |
| 0096 | E8 | FF67 | call Delay | |
| 0099 | E8 | FF9D | call TestKey | |
| 009C | 75 | F0 | jne WaitRelease | |
| 009E  00A0  00A1 | 8A C3 | C3 | mov al, bl ret  GetKey endp | |
| 00A1  00A1 | B8 | 0000s | Start proc near mov ax, data | |
| 00A4 | 8E | D8 | mov ds, ax | |
| 00A6 | C6 | 06 0000r FF | mov LEDBuf, 0ffh ; 显 示 8.8.8.8. | |
| 00AB | C6 | 06 0001r FF | mov LEDBuf+1, 0ffh | |
| 00B0 | C6 | 06 0002r FF | mov LEDBuf+2, 0ffh | |
| 00B5 | C6 | 06 0003r FF | mov LEDBuf+3, 0ffh | |
| 00BA | C6 | 06 0004r 00 | mov LEDBuf+4, 0 | |
| 00BF  00C4  00C4 | C6  E8 | 06 0005r 00  FF44 | mov LEDBuf+5, 0 MLoop:  call DisplayLED ; 显示 | |
| 00C7 | E8 | FF6F | call TestKey ; 有键入? | |
| 00CA | 74 | F8 | je MLoop ; 无键入, 继续显示 | |
| 00CC | E8 | FF79 | call GetKey ; 读入键码 | |
| 00CF | 24 | 0F | and al, 0fh ; 显示键码 | |
| 00D1 | B4 | 00 | mov ah, 0 | |
| 00D3 | BB | 0008r | mov bx, offset LEDMAP | |
| 00D6 | 03 | D8 | add bx, ax | |
| 00D8 | 8A | 07 | mov al, [bx] | |
| 00DA | A2 | 0005r | mov LEDBuf+5, al | |
| 00DD | EB | E5 | jmp MLoop | |
| 00DF  00DF |  |  | Start endp code ends | |

end start

; 硬件实验十八 电子时钟 H18.ASM

;

; 本实验利用 8253 做定时器，用定时器输出的脉冲控制 8259 产Th中断

; 在 8259 中断处理程序中，对时、分、秒进行计数，在等待中断的循

; 环中用 LED 显示时间。

; 8253 用定时器/计数器 1，8253 片选接 CS4，地址为 0C000H。8253 时钟

; 源 CLK1 接分频电路的 F/64 输出。分频器的 Fin 接 4MHz 时钟。8253 的

; GATE1 接 VCC。

; 8259 中断 INT0 接 8253 的 OUT1，片选接 CS5，地址为 0D000H。

; 显示电路的 KEY/LED CS 接 CS0，地址为 08000H。

;

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| =C003  =C000  =C001  =C002 | | | CONTROL COUNT0 COUNT1 COUNT2 | | equ equ equ equ | 0c003h 0c000h 0c001h 0c002h |  | |
| =0082 | | | md8255 | | equ | 82h | ; 8255 工作方式 | |
| =8000 | | | PA8255 | | equ | 8000h | ; 8255 PA 口输出地址 | |
| =8003 | | | CTL8255 | | equ | 8003h | ; 8255 控制口地址 | |
| =8002 | | | OUTBIT | | equ 08002h | | ; 位控制口 | |
| =8004 | | | OUTSEG | | equ 08004h | | ; 段控制口 | |
| =0013 | | | ICW1 | | equ | 00010011b | ; 单片 8259, 上升沿中断, 要写 ICW4 | |
| =0020 | | | ICW2 | | equ | 00100000b | ; 中断号为 20H | |
| =0001 | | | ICW4 | | equ | 00000001b | ; 工作在 8086/88 方式 | |
| =00FE | | | OCW1 | | equ | 11111110b | ; 只响应 INT0 中断 | |
| =D000  =D001 | | | CS8259A CS8259B | | equ equ | 0d000h 0d001h | ; 8259 地址 | |
| 0000 |  | | data segment | | | | | |
| 0000 | 00 | | CNT db 0 | | | | | |
| 0001 | 06\*(??) | | LEDBuf db 6 dup(?) ; 显示缓冲 | | | | | |
| 0007 | 01\*(??) | | Num db 1 dup(?) ; 显示的数据 | | | | | |
| 0008 | 01\*(??) | | DelayT db 1 dup(?) | | | | | |
| 0009 | 00 | | Hour db 0 | | | | | |
| 000A | 00 | | Minute db 0 | | | | | |
| 000B  000C | 00 | | Second db 0  LEDMAP: ; 八段管显示码 | | | | | |
| 000C  0014 | 3F  07  7F | 06 5B 4F 66  6F 77 7C 39 | 6D  5E | 7D+  79+ | db  db | | 3fh, 06h, 5bh, 4fh, 66h, 6dh,  7fh, 6fh, 77h, 7ch, 39h, 5eh, | 7dh, 07h  79h, 71h |
|  | 71 |  |  |  |  | |  |  |

001C data ends

0000 code segment

assume cs:code, ds: data

|  |  |  |
| --- | --- | --- |
| 0000 |  | Delay proc near |
| 0000 | 50 | push ax ; 延时子程序 |
| 0001 | 51 | push cx |

0002 B0 00 mov al, 0

0004 8B C8 mov cx,ax

0006 E2 FE loop $

0008 59 pop cx

0009 58 pop ax

000A C3 ret

000B Delay endp

|  |  |  |  |
| --- | --- | --- | --- |
| 000B |  |  | DisplayLED proc near |
| 000B | BB | 0001r | mov bx, offset LEDBuf |
| 000E | B1 | 06 | mov cl, 6 ; 共 6 个八段管 |
| 0010 | B4 | 20 | mov ah, 00100000b ; 从左边开始显示 |
| 0012 |  |  | DLoop: |
| 0012 | BA | 8002 | mov dx, OUTBIT |
| 0015 | B0 | 00 | mov al, 0 |
| 0017 | EE |  | out dx,al ; 关所有八段管 |
| 0018 | 8A | 07 | mov al, [bx] |
| 001A | BA | 8004 | mov dx, OUTSEG |
| 001D | EE |  | out dx,al |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 001E  0021 | BA  8A | 8002  C4 | mov  mov | dx, OUTBIT  al, ah |  |
| 0023 | EE |  | out | dx, al | ; 显示一位八段管 |
| 0024 | 50 |  | push | ax |  |
| 0025 | B4 | 01 | mov | ah, 1 |  |
| 0027  002A | E8 58 | FFD6 | call pop | Delay ax |  |
| 002B  002D  002E | D0 43  FE | EC  C9 | shr inc  dec | ah, 1 bx  cl |  |
| 0030 | 75 | E0 | jnz | DLoop |  |
| 0032 | BA | 8002 | mov | dx, OUTBIT |  |
| 0035 | B0 | 00 | mov | al, 0 |  |
| 0037 | EE |  | out | dx,al | ; 关所有八段管 |
| 0038 | C3 |  | ret |  |  |
| 0039 |  |  | DisplayLED | endp |  |
| 0039 |  |  | IEnter proc near | | |
| 0039 | 50 |  | push ax | | |
| 003A | 52 |  | push dx | | |
| 003B | FE | 06 000Br | inc Second | | |
| 003F | A0 | 000Br | mov al, Second | | |
| 0042 | 3C | 3C | cmp al, 60 | | |
| 0044 | 75 | 25 | jne Exit | | |
| 0046 | C6 | 06 000Br 00 | mov Second, 0 | | |
| 004B | FE | 06 000Ar | inc Minute | | |
| 004F | A0 | 000Ar | mov al, Minute | | |
| 0052 | 3C | 3C | cmp al, 60 | | |
| 0054 | 75 | 15 | jne Exit | | |
| 0056 | C6 | 06 000Ar 00 | mov Minute, 0 | | |
| 005B | FE | 06 0009r | inc Hour | | |
| 005F | A0 | 0009r | mov al, Hour | | |
| 0062 | 3C | 18 | cmp al, 24 | | |
| 0064 | 75 | 05 | jne Exit | | |
| 0066  006B  006B | C6  BA | 06 0009r 00  D000 | mov Hour, 0 Exit:  mov dx, CS8259A | | |
| 006E  0070 | B0 EE | 20 | mov al, 20h ; 中断服务程序结束指令  out dx, al | | |
| 0071 | 5A |  | pop dx | | |
| 0072 | 58 |  | pop ax | | |
| 0073 | CF |  | iret | | |
| 0074 |  |  | IEnter endp | | |
| 0074 |  |  | IInit proc | | |
| 0074 | BA | D000 | mov dx, CS8259A | | |
| 0077 | B0 | 13 | mov al, ICW1 | | |
| 0079 | EE |  | out dx, al | | |
| 007A | BA | D001 | mov dx, CS8259B | | |
| 007D  007F | B0 EE | 20 | mov al, ICW2  out dx, al | | |
| 0080 | B0 | 01 | mov al, ICW4 | | |
| 0082 | EE |  | out dx, al | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0083 | B0 | FE | mov | | | al, OCW1 | | |
| 0085 | EE |  | out | | | dx, al | | |
| 0086 | C3 |  | ret | | |  | | |
| 0087 |  |  | IInit endp | | |  | | |
| 0087 |  |  | start proc | | | near | | |
| 0087 | B8 | 0000s | mov | | | ax, data | | |
| 008A | 8E | D8 | mov | | | ds, ax | | |
| 008C | C6 | 06 0009r 0E | mov | | | Hour, 14 | | |
| 0091 | C6 | 06 000Ar 15 | mov | | | Minute, 21 | | |
| 0096 | C6 | 06 000Br 00 | mov | | | Second, 0 | | |
| 009B | FA |  | cli | | |  | | |
| 009C | B8 | 0000 | mov | | | ax, 0 | | |
| 009F | 8E | D8 | mov | | | ds, ax | | |
| 00A1 | B0 | 76 | mov | | | al, 76h; 01110110B ; 计数器 1,16 位,方式 3,二进制 | | |
| 00A3 | BA | C003 | mov | | | dx, CONTROL | | |
| 00A6 | EE |  | out | | | dx, al | | |
| 00A7 | B8 | F424 | mov | | | ax, 62500 | | |
| 00AA | BA | C001 | mov | | | dx, COUNT1 | | |
| 00AD  00AE  00B0 | EE 8A EE | C4 | out mov out | | | dx, al ; 计数器低字节  al, ah  dx, al ; 计数器高字节 | | |
| 00B1 | B0 | 82 | mov | | | al, MD8255 | | |
| 00B3 | BA | 8003 | mov | | | dx, CTL8255 | | |
| 00B6 | EE |  | out | | | dx, al | | |
| 00B7 | BB | 0080 | mov | | | bx, 4\*ICW2 ; 中断号 | | |
| 00BA | B8 | 0000s | mov | | | ax, code | | |
| 00BD | D1 E0 | E0 D1 E0 D1 E0 | | D1+ | | shl | ax, 4 | ; x 16 |
| 00C5 | 05 | 0039r | add | | ax, offset IEnter ; 中断入口地址（段地址为 0） | | | |
| 00C8 | 89 | 07 | mov | | [bx], ax | | | |
| 00CA | B8 | 0000 | mov | | ax, 0 | | | |
| 00CD | 43 |  | inc | | bx | | | |
| 00CE | 43 |  | inc | | bx | | | |
| 00CF | 89 | 07 | mov | | [bx], ax ; 代码段地址为 0 | | | |
| 00D1 | E8 | FFA0 | call | | IInit | | | |
| 00D4 | B8 | 0000s | mov | | ax, data | | | |
| 00D7 | 8E | D8 | mov | | ds, ax | | | |
| 00D9  00DE  00E1  00E4  00E5  00E6  00E6  00E9  00EB  00ED | C6 A0 BA EE FB  A0 B4 B1 F6 | 06 0000r 00  0000r  8000  0009r  00  0A F1 | LP: | mov mov mov out sti  mov mov mov div | CNT, 0 ; 计数值初始为 0 al, CNT  dx, PA8255  dx, al  ; 等待中断，并计数。  al, Hour ah, 0  cl, 10 cl | | | |
| 00EF | 8A | EC |  | mov | ch, ah | | | |
| 00F1  00F3 | B4 BB | 00  000Cr |  | mov mov | ah, 0  bx, offset LEDMAP | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 00F6 | 03 | D8 | add | bx, ax |
| 00F8 | 8A | 07 | mov | al, [bx] |
| 00FA | A2 | 0001r | mov | LEDBuf, al |
| 00FD | BB | 000Cr | mov | bx, offset LEDMAP |
| 0100 | 8A | C5 | mov | al, ch |
| 0102 | B4 | 00 | mov | ah, 0 |
| 0104 | 03 | D8 | add | bx, ax |
| 0106 | 8A | 07 | mov | al, [bx] |
| 0108 | 0C | 80 | or | al, 80h |
| 010A | A2 | 0002r | mov | LEDBuf+1, al |
| 010D | A0 | 000Ar | mov | al, Minute |
| 0110 | B4 | 00 | mov | ah, 0 |
| 0112 | B1 | 0A | mov | cl, 10 |
| 0114 | F6 | F1 | div | cl |
| 0116 | 8A | EC | mov | ch, ah |
| 0118 | B4 | 00 | mov | ah, 0 |
| 011A | BB | 000Cr | mov | bx, offset LEDMAP |
| 011D | 03 | D8 | add | bx, ax |
| 011F | 8A | 07 | mov | al, [bx] |
| 0121 | A2 | 0003r | mov | LEDBuf+2, al |
| 0124 | BB | 000Cr | mov | bx, offset LEDMAP |
| 0127 | 8A | C5 | mov | al, ch |
| 0129 | B4 | 00 | mov | ah, 0 |
| 012B | 03 | D8 | add | bx, ax |
| 012D | 8A | 07 | mov | al, [bx] |
| 012F | 0C | 80 | or | al, 80h |
| 0131 | A2 | 0004r | mov | LEDBuf+3, al |
| 0134 | A0 | 000Br | mov | al, Second |
| 0137 | B4 | 00 | mov | ah, 0 |
| 0139 | B1 | 0A | mov | cl, 10 |
| 013B | F6 | F1 | div | cl |
| 013D | 8A | EC | mov | ch, ah |
| 013F | B4 | 00 | mov | ah, 0 |
| 0141 | BB | 000Cr | mov | bx, offset LEDMAP |
| 0144 | 03 | D8 | add | bx, ax |
| 0146 | 8A | 07 | mov | al, [bx] |
| 0148 | A2 | 0005r | mov | LEDBuf+4, al |
| 014B | BB | 000Cr | mov | bx, offset LEDMAP |
| 014E | 8A | C5 | mov | al, ch |
| 0150 | B4 | 00 | mov | ah, 0 |
| 0152 | 03 | D8 | add | bx, ax |
| 0154 | 8A | 07 | mov | al, [bx] |
| 0156 | 0C | 80 | or | al, 80h |
| 0158 | A2 | 0006r | mov | LEDBuf+5, al |
| 015B | E8 | FEAD | call | DisplayLED |
| 015E | EB | 86 | jmp | LP |
| 0160 |  |  | start endp |  |
| 0160 |  |  | code ends |  |

end start

; 硬件实验二十 打印机控制试验 H20.ASM

; printer

; ASM for 8086/8088

=0082 mode equ 082h

=8003 contrl equ 08003h ; 8255 方式控制

=8001 status equ 08001h ; 状态端口地址

=8002 ctl equ 08002h ; 控制端口地址

=8000 dport equ 08000h ; 数据端口地址

=0001 stb equ 01h ; 选通位

=0001 busy equ 01h ; 忙状态位

0000 data segment

0000 48 65 6C 6C 6F 20 57+ string db 'Hello World!',0dh,0ah,0ch,0,0

6F 72 6C 64 21 0D 0A+

0C 00 00

0011 data ends

0000 code segment

assume cs:code, ds:data

|  |  |  |  |
| --- | --- | --- | --- |
| 0000 |  |  | print proc near |
| 0000 | 8A | E0 | mov ah,al |
| 0002 | BA | 8001 | mov dx,status |
| 0005 |  |  | rd\_status: |
| 0005 | EC |  | in al,dx ; 读打印机状态 |
| 0006 | A8 | 01 | test al,busy |
| 0008 | 75 | FB | jnz rd\_status ; 若忙，等待 |
| 000A | BA | 8000 | mov dx,dport ; 数据送出 |
| 000D  000F | 8A EE | C4 | mov al,ah  out dx,al |
| 0010 | BA | 8002 | mov dx,ctl ; 输出选通脉冲 |
| 0013 | B0 | 01 | mov al,stb |
| 0015 | EE |  | out dx,al |
| 0016 | B0 | 00 | mov al,0 |
| 0018 | EE |  | out dx,al |
| 0019  001A | C3 |  | ret print endp |
| 001A  001A | B8 | 0000s | start proc near mov ax, data |
| 001D | 8E | D8 | mov ds, ax |
| 001F  0022 | BE | 0000r | mov si, offset string ; 被打印字串  nextchar: |
| 0022 | AC |  | lodsb |
| 0023 | 3C | 00 | cmp al,0 |
| 0025 | 75 | 02 | jnz prtchar ; 字串结束 |
| 0027 | EB | FE | jmp $ |
| 0029 |  |  | prtchar: |
| 0029 | E8 | FFD4 | call print |
| 002C  002E  002F  002F | EB 90 | F4 | jmp nextchar ; 下一字符  nop  start endp code ends |

end start

; 硬件实验二十一 直流电机控制实验 H21.ASM

|  |  |  |  |
| --- | --- | --- | --- |
| =0082 | mode | equ | 082h |
| =8003 | CONTRL | equ | 08003h |
| =8001 | STATUS | equ | 08001h |
| =9000 | CS0832 | equ | 09000h |

=0001 DC\_P equ 1

0000 code segment assume cs:code

|  |  |  |  |
| --- | --- | --- | --- |
| 0000 |  |  | delay proc near |
| 0000 | 51 |  | push cx |
| 0001 | 50 |  | push ax |
| 0002 | B8 | 0004 | mov ax, 4 |
| 0005 | B9 | 0000 | mov cx, 0 |
| 0008 |  |  | ddd: |
| 0008 | 90 |  | nop |
| 0009 | E2 | FD | loop ddd |
| 000B | 48 |  | dec ax |
| 000C | 75 | FA | jnz ddd |
| 000E | 58 |  | pop ax |
| 000F | 59 |  | pop cx |
| 0010 | C3 |  | ret |
| 0011 |  |  | delay endp |

0011 read proc near

0011 BA 8001 mov dx, STATUS

0014 EC r\_0: in al, dx

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0015 | A8 | 01 | test al, DC\_P | |  |
| 0017 | 74 | FB | jz r\_0 | | ; 等待低电平完 |
| 0019 | EC |  | r\_1: in al,dx | |  |
| 001A | A8 | 01 | test | al, DC\_P | |
| 001C | 75 | FB | jnz r\_1 ; 等待高电平完 | | |
| 001E | B9 | 0000 | mov cx, 0 ; 开始准备计时 | | |
| 0021 | EC |  | r\_2: in al,dx | | |
| 0022 | 41 |  | inc cx ; 计数器加 1 | | |
| 0023 | A8 | 01 | test al, DC\_P | | |
| 0025 | 74 | FA | jz r\_2 ; 测量低电平宽度 | | |
| 0027 | EC |  | r\_3: in al,dx | | |
| 0028 | 41 |  | inc cx | | |
| 0029 | A8 | 01 | test al, DC\_P | | |
| 002B | 75 | FA | jnz r\_3 ; 测量高电平宽度 | | |
| 002D  002F  0030 | 8B C3 | C1 | mov ax, cx ret  read endp | | |
| 0030 |  |  | start proc near | | |
| 0030 | B0 | 82 | mov al,mode | | |
| 0032 | BA | 8003 | mov dx,CONTRL | | |
| 0035 | EE |  | OUT dx,al | | |
| 0036 | BA | 9000 | mov dx,CS0832 | | |
| 0039  003B  003C | B0 EE E8 | FF  FFC1 | mov al,0ffh  out dx,al  call delay ; 等待电机运转稳定 | | |
| 003F | E8 | FFCF | call read ; 读取时间 | | |
| 0042 | BA | 9000 | mov dx,CS0832 ; 设断点，观察上次时间 | | |
| 0045 | B0 | C0 | mov al,0c0h | | |
| 0047 | EE |  | out dx,al | | |
| 0048 | E8 | FFB5 | call delay ; 等待电机运转稳定 | | |
| 004B | E8 | FFC3 | call read | | |
| 004E | BA | 9000 | mov dx,CS0832 ; 设断点，观察上次时间 | | |
| 0051 | B0 | 40 | mov al,040h ; 电机反转 | | |
| 0053 | EE |  | out dx,al | | |
| 0054 | E8 | FFA9 | call delay ; 等待电机运转稳定 | | |
| 0057 | E8 | FFB7 | call read ; 读取时间 | | |
| 005A | BA | 9000 | mov dx,CS0832 ; 设断点，观察上次时间 | | |
| 005D | B0 | 00 | mov al,00h ; 电机反转加速 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 005F  0060 | EE  E8 FF9D | out  call | dx,al  delay | ; 等待电机运转稳定 |
| 0063 | E8 FFAB | call | read |  |
| 0066 | EB FE | jmp | $ | ; 设断点，观察上次时间 |
| 0068 | start endp | | | |
| 0068 | code ends | | | |

end start

; 硬件实验二十二 步进电机控制实验 H22.ASM

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | =0082 | | mode | | equ | 082h |
| =8000 | | ctl | | equ | 08000h |
| =8003 | | contrl | | equ | 08003h |
| =0001 | | Astep | | equ | 01h |
| =0002 | | Bstep | | equ | 02h |
| =0004 | | Cstep | | equ | 04h |
| =0008 | | Dstep | | equ | 08h |
| 0000 |  | | data | | segment |  |
| 0000 | 0000 | | dly\_c | | dw 0 |  |
| 0002 |  | | data | | ends |  |
| 0000 |  | | code | | segment |  |
| assume cs:code, ds:data  0000 start proc near | | | | | | |
| 0000 | B8 | 0000s |  | mov | ax, data | |
| 0003 | 8E | D8 |  | mov | ds, ax | |
| 0005 | BA | 8003 |  | mov | dx, contrl | |
| 0008 | B0 | 82 |  | mov | al, mode | |
| 000A | EE |  |  | out | dx, al | |
| 000B | BA | 8000 |  | mov | dx, ctl | |
| 000E | B0 | 00 |  | mov | al,0 | |
| 0010 | EE |  |  | out | dx,al | |
| 0011 | C7 | 06 0000r | 1000 |  | mov dly\_c,1000h | |
| 0017 | EB | 45 90 |  | jmp | step4 | |
| ; 单/双八拍工作方式 | | | | | | |
| 001A |  |  | step8: | |  | |
| 001A | BA | 8000 | mov | | dx, ctl | |
| 001D | B0 | 01 | mov | | al,Astep | |
| 001F | EE |  | out | | dx,al | |
| 0020 | E8 | 0093 | call | | delay | |
| 0023 | B0 | 03 | mov | | al,Astep+Bstep | |
| 0025 | EE |  | out | | dx,al | |
| 0026 | E8 | 008D | call | | delay | |
| 0029 | B0 | 02 | mov | | al,Bstep | |
| 002B | EE |  | out | | dx,al | |
| 002C | E8 | 0087 | call | | delay | |
| 002F | B0 | 06 | mov | | al,Bstep+Cstep | |
| 0031 | EE |  | out | | dx,al | |
| 0032 | E8 | 0081 | call | | delay | |
| 0035 | B0 | 04 | mov | | al,Cstep | |
| 0037 | EE |  | out | | dx,al | |
| 0038 | E8 | 007B | call | | delay | |
| 003B | B0 | 0C | mov | | al,Cstep+Dstep | |
| 003D | EE |  | out | | dx,al | |
| 003E | E8 | 0075 | call | | delay | |
| 0041 | B0 | 08 | mov | | al,Dstep | |
| 0043 | EE |  | out | | dx,al | |
| 0044 | E8 | 006F | call | | delay | |
| 0047 | B0 | 09 | mov | | al,Dstep+Astep | |
| 0049 | EE |  | out | | dx,al | |

004A E8 0069 call delay

004D A1 0000r mov ax, dly\_c

0050 FE CC dec ah

0052 3D 0100 cmp ax, 100h ; 提高转速

0055 75 02 jne nn1 ; 最快速度

0057 FE C4 inc ah

0059 A3 0000r nn1: mov dly\_c,ax 005C EB BC jmp step8

; 双四拍工作方式

005E step4:

005E BA 8000 mov dx, ctl

0061 B0 03 mov al,Astep+Bstep

0063 EE out dx,al

0064 E8 004F call delay

0067 B0 06 mov al,Bstep+Cstep

0069 EE out dx,al

006A E8 0049 call delay

006D B0 0C mov al,Cstep+Dstep

006F EE out dx,al

0070 E8 0043 call delay

0073 B0 09 mov al,Dstep+Astep

0075 EE out dx,al

0076 E8 003D call delay

0079 A1 0000r mov ax, dly\_c

007C FE CC dec ah

007E 3D 0200 cmp ax, 200h

0081 75 02 jne nn2

0083 FE C4 inc ah

0085 A3 0000r nn2: mov dly\_c,ax

0088 EB D4 jmp step4

; 单四拍工作方式

008A step41:

008A BA 8000 mov dx,ctl

008D B0 08 mov al,Dstep

008F EE out dx,al

0090 E8 0023 call delay

0093 B0 04 mov al,Cstep

0095 EE out dx,al

0096 E8 001D call delay

0099 B0 02 mov al,Bstep

009B EE out dx,al

009C E8 0017 call delay

009F B0 01 mov al,Astep

00A1 EE out dx,al

00A2 E8 0011 call delay

00A5 A1 0000r mov ax, dly\_c

00A8 FE CC dec ah

00AA 3D 0300 cmp ax, 300h

00AD 75 02 jne nn3

00AF FE C4 inc ah

00B1 A3 0000r nn3: mov dly\_c,ax

00B4 EB D4 jmp step41 00B6 start endp

00B6 delay proc near 00B6 51 push cx

00B7 8B 0E 0000r mov cx, dly\_c 00BB dd1:

;nop

00BB E2 FE loop dd1

00BD 59 pop cx

00BE C3 ret

00BF delay endp

00BF code ends

end start

; 硬件实验二十三 温度传感器试验 H23.ASM

|  |  |  |  |
| --- | --- | --- | --- |
| =9000 | ADPort | equ 09000h | ; AD 采样输入片选 |
| =8002 | OUTBIT | equ 08002h | ; 位控制口 |
| =8004 | OUTSEG | equ 08004h | ; 段控制口 |
| =8001 | IN\_KEY | equ 08001h | ; 键盘读入口 |

=-0063 LowTemp equ -99 ; A/D 0

=0063 HighTemp equ 99 ; A/D 255

0000 data segment

0000 06\*(??) LEDBuf db 6 dup(?) ; 显示缓冲

0006 00 CurTemp db 0 ; 当前温度

0007 LEDMAP: ; 八段管显示码

0007 3F 06 5B 4F 66 6D 7D+ db 3fh, 06h, 5bh, 4fh, 66h, 6dh, 7dh, 07h

07

000F 7F 6F 77 7C 39 5E 79+ db 7fh, 6fh, 77h, 7ch, 39h, 5eh, 79h, 71h

71

0017 data ends

0000 code segment

assume cs:code, ds:data

|  |  |  |  |
| --- | --- | --- | --- |
| 0000 |  |  | Delay proc near |
| 0000 | 50 |  | push ax ; 延时子程序 |
| 0001 | 51 |  | push cx |
| 0002 | B0 | 00 | mov al, 0 |
| 0004 | 8B | C8 | mov cx,ax |
| 0006 | E2 | FE | loop $ |
| 0008 | 59 |  | pop cx |
| 0009  000A  000B | 58  C3 |  | pop ax ret  Delay endp |
| 000B |  |  | DisplayLED proc near |
| 000B | BB | 0000r | mov bx, offset LEDBuf |
| 000E | B1 | 06 | mov cl, 6 ; 共 6 个八段管 |
| 0010 | B4 | 20 | mov ah, 00100000b ; 从左边开始显示 |
| 0012 |  |  | DLoop: |
| 0012 | BA | 8002 | mov dx, OUTBIT |
| 0015 | B0 | 00 | mov al, 0 |
| 0017 | EE |  | out dx,al ; 关所有八段管 |
| 0018 | 8A | 07 | mov al, [bx] |
| 001A | BA | 8004 | mov dx, OUTSEG |
| 001D | EE |  | out dx,al |
| 001E | BA | 8002 | mov dx, OUTBIT |
| 0021 | 8A | C4 | mov al, ah |
| 0023 | EE |  | out dx, al ; 显示一位八段管 |
| 0024 | 50 |  | push ax |
| 0025 | B4 | 01 | mov ah, 1 |
| 0027  002A | E8 58 | FFD6 | call Delay pop ax |
| 002B  002D  002E | D0 43  FE | EC  C9 | shr ah, 1  inc bx  dec cl |
| 0030 | 75 | E0 | jnz DLoop |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0032 | BA | 8002 | mov | dx, OUTBIT |  |
| 0035 | B0 | 00 | mov | al, 0 |
| 0037 | EE |  | out | dx,al | ; 关所有八段管 |
| 0038 | C3 |  | ret |  |  |
| 0039 |  |  | DisplayLED | endp |  |

; ===================================

|  |  |  |  |
| --- | --- | --- | --- |
| 0039 |  |  | DisplayResult proc near |
| 0039 | A0 | 0006r | mov al, CurTemp |
| 003C | A8 | 80 | test al, 80h |
| 003E | 74 | 0C | jz GE0 |
| 0040 | C6 | 06 0000r 40 | mov LEDBuf, 40h ; '-' |
| 0045 | FE | C8 | dec al |
| 0047 | F6 | D0 | not al |
| 0049  004C  004C  0051 | EB  C6 | 06 90  06 0000r 00 | jmp Goon GE0:  mov LEDBuf, 0 ; ' '  Goon: |
| 0051 | B1 | 0A | mov cl, 10 |
| 0053 | B4 | 00 | mov ah, 0 |
| 0055 | F6 | F1 | div cl |
| 0057 | 8A | D8 | mov bl, al |
| 0059 | B7 | 00 | mov bh, 0 |
| 005B | 81 | C3 0007r | add bx, offset LEDMAP |
| 005F | 8A | 07 | mov al, [bx] |
| 0061 | A2 | 0001r | mov LEDBuf+1, al |
| 0064 | 8A | DC | mov bl, ah |
| 0066 | B7 | 00 | mov bh, 0 |
| 0068 | 81 | C3 0007r | add bx, offset LEDMAP |
| 006C | 8A | 07 | mov al, [bx] |
| 006E | A2 | 0002r | mov LEDBuf+2, al |
| 0071 | C6 | 06 0003r 00 | mov LEDBuf+3, 0 ; ' ' |
| 0076 | C3 |  | ret |
| 0077 |  |  | DisplayResult endp |
| 0077 |  |  | ReadAD proc near |
| 0077 | BA | 9000 | mov dx, ADPort |
| 007A  007C  007D  007F  007F | B0 EE B0  FE | 00  3C C8 | mov al, 0  out dx, al  mov al, 60  rr:  dec al ; delay |
| 0081 | 75 | FC | jnz rr |
| 0083 | BA | 9000 | mov dx, ADPort |
| 0086 | EC |  | in al, dx |
| 0087 | 8A | D0 | mov dl, al |
| 0089 | B0 | FF | mov al, 0ffh |
| 008B  008D  008E | 2A C3 | C2 | sub al, dl ret  ReadAD endp |
| 008E |  |  | ReadTemp proc near |
| 008E | BB | 0000 | mov bx, 0 |
| 0091 | B1 | 10 | mov cl, 16 |
| 0093 |  |  | RLoop: |
| 0093 | E8 | FFE1 | call ReadAD |
| 0096 | B4 | 00 | mov ah, 0 |
| 0098 | 03 | D8 | add bx, ax |
| 009A | FE | C9 | dec cl |
| 009C | 75 | F5 | jnz RLoop |
| 009E | D1 EB | EB D1 EB D1 | EB D1+ shr bx, 4 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 00A6  00A8  00AA  00AC  00AE  00B0  00B3 | 8B B1 F6 8A  04  A2 C3 | C3 C6 E1 C4 9D  0006r | mov mov mul mov add mov  ret | ax, bx  cl, HighTemp cl  al, ah  al, LowTemp CurTemp, al | - LowTemp  ; /256 |
| 00B4 |  |  | ReadTemp endp | | |
| 00B4  00B4 | B8 | 0000s | Start proc near mov ax, data | | |
| 00B7  00B9  00B9 | 8E  E8 | D8  FF7D | mov ds, ax MLoop:  call DisplayResult | | |
| 00BC | E8 | FF4C | call DisplayLED | | |
| 00BF | E8 | FFCC | call ReadTemp | | |
| 00C2 | EB | F5 | jmp MLoop | | |
| 00C4  00C4 |  |  | start endp code ends | | |

end start

; 硬件实验二十四 液晶显示控制电路 H24.ASM

=8000 CWADD1 EQU 08000H ;写指令代码地址（E1）

=8001 DWADD1 EQU 08001H ;写显示数据地址（E1）

=8002 CRADD1 EQU 08002H ;读状态字地址（E1）

=8003 DRADD1 EQU 08003H ;读显示数据地址（E1）

=8004 CWADD2 EQU 08004H ;写指令代码地址（E2）

=8005 DWADD2 EQU 08005H ;写显示数进地址（E2）

=8006 CRADD2 EQU 08006H ;读状态字地址（E2）

=8007 DRADD2 EQU 08007H ;读显示数据地址（E2）

=003D PD1 EQU 3DH ;122/2 分成左右两半屏 122x32 0000 DATA SEGMENT

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0000 | 0000 | FONT | | | DW | 0000H | ;Font address |
| 0002 | 00 | COLUMN | | | DB | 0H |  |
| 0003 | 00 | PAGE\_ | | | DB | 0H | ;页地址寄存器 D1,DO:页地址 |
| 0004 | 00 | CODE\_ | | | DB | 0H | ;字符代码寄存器 |
| 0005 | 00 | COUNT | | | DB | 0H | ;计数器 |
| 0006 | 00 | DIR | | | DB | 0h |  |
| 0007 | 00 | R4 | | | DB | 0h |  |
| 0008 | 00 | R3 | | | DB | 0h |  |
| 0009 | 00 | COL2 | | | DB | 0h |  |
| 000A | 00 | CTEMP | | | DB | 0H |  |
| 000B | 00 | COM | | | DB | 0H | ;指令寄存器 |
| 000C | 00 | DAT | | | DB | 0H | ;数据寄存器 |
|  |  | ;中文字符库 | | |  |  |  |
| 000D |  | CCTAB: | | |  |  |  |
| 000D | 00 04 | E4 | 24 24 64 | B4+ | DB 000H,004H,0e4H,024H,024H,064H,0b4H,02fH ; 南 | | |
|  | 2F |  |  |  |  | | |
| 0015 | 24 A4 | 64 | 24 24 E6 | 04+ | DB 024H,0a4H,064H,024H,024H,0e6H,004H,000H | | |
|  | 00 |  |  |  |  | | |
| 001D | 00 00 | 7F | 04 05 05 | 05+ | DB 000H,000H,07fH,004H,005H,005H,005H,07fH | | |
|  | 7F |  |  |  |  | | |
| 0025 | 05 05 | 05 | 25 44 3F | 00+ | DB 005H,005H,005H,025H,044H,03fH,000H,000H | | |
|  | 00 |  |  |  |  | | |
| 002D | 00 04 | 04 | E4 24 24 | 25+ | DB 000H,004H,004H,0e4H,024H,024H,025H,026H ; 京 | | |
|  | 26 |  |  |  |  | | |
| 0035 | 24 24 | 24 | E4 06 04 | 00+ | DB 024H,024H,024H,0e4H,006H,004H,000H,000H | | |
|  | 00 |  |  |  |  | | |
| 003D | 00 20 | 10 | 19 0D 41 | 81+ | DB 000H,020H,010H,019H,00DH,041H,081H,07fH | | |

;

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0045 | 7F  01 | 01 | 05 0D 38 10 | | 00+ | DB 001H,001H,005H,00dH,038H,010H,000H,000H | | |
|  | 00 |  |  | |  |  | | |
| 004D | 00 | 80 | 60 F8 07 0A | | 28+ | DB 000H,080H,060H,0f8H,007H,00aH,028H,028H ; 伟 | | |
|  | 28 |  |  | |  |  | | |
| 0055 | 28 | FF | 28 28 28 28 | | 00+ | DB 028H,0ffH,028H,028H,028H,028H,000H,000H | | |
|  | 00 |  |  | |  |  | | |
| 005D | 01 | 00 | 00 FF 00 01 | | 01+ | DB 001H,000H,000H,0fFH,000H,001H,001H,001H | | |
|  | 01 |  |  | |  |  | | |
| 0065 | 01 | FF | 01 21 41 3F | | 00+ | DB 001H,0ffH,001H,021H,041H,03fH,000H,000H | | |
|  | 00 |  |  | |  |  | | |
| 006D  0075 | 08  7A  4A | 08  4A | 89 CE 28 18  4A 4A 4A 7A | | 02+  02+ | DB 008H,008H,089H,0ceH,028H,018H,002H,07aH ; 福  DB 04aH,04aH,04aH,04aH,04aH,07aH,002H,000H | | |
|  | 00 |  |  | |  |  | | |
| 007D | 02 | 01 | 00 FF 01 02 | | FF+ | DB 002H,001H,000H,0ffH,001H,002H,0ffH,049H | | |
|  | 49 |  |  | |  |  | | |
| 0085 | 49 | 49 | 7F 49 49 49 | | FF+ | DB 049H,049H,07fH,049H,049H,049H,0ffH,000H | | |
| 008D | 00 |  | DATA | |  | ENDS | | |
| 0000 |  |  | CODE | |  | SEGMENT | | |
| ASSUME CS:CODE, DS:DATA | | | | | | | | |
| 0000 |  |  | | ; 初始化程序  INIT PROC NEAR | | |  |  |
| 0000 | B4 | E2 | | MOV AH, | | | 0E2H | ;复位 |
| 0002 | E8 | 0068 | | CALL PRO | | |  |  |
| 0005 | E8 | 0094 | | CALL PR3 | | |  |  |
| 0008 | B4 | A4 | | MOV AH, | | | 0A4H | ;关闭休闭状态 |
| 000A  000D  0010 | E8 E8  B4 | 0060  008C A9 | | CALL PRO  CALL PR3  MOV AH, | | | 0A9H | ;设置 1／32 占空比 |
| 0012 | E8 | 0058 | | CALL PRO | | |  |  |
| 0015 | E8 | 0084 | | CALL PR3 | | |  |  |
| 0018 | B4 | A0 | | MOV AH, | | | 0A0H | ;正向排序设置 |
| 001A  001D  0020 | E8 E8  B4 | 0050  007C C0 | | CALL PRO  CALL PR3  MOV AH, | | | 0C0H | ;设置显示起始行为第一行 |
| 0022 | E8 | 0048 | | CALL PRO | | |  |  |
| 0025 | E8 | 0074 | | CALL PR3 | | |  |  |
| 0028 | B4 | AF | | MOV AH, | | | 0AFH | ;开显示设置 |
| 002A  002D  0030 | E8 E8  C3 | 0040  006C | | CALL PRO  CALL PR3 RET | | |  |  |
| 0031 |  |  | | INIT endp | | |  |  |

;

; 清屏

0031 CLEAR PROC NEAR

0031 C6 06 0007r 00 MOV R4, 00H ;页面地址暂存器设置

0036 A0 0007r CLEAR1: MOV AL,R4 ;取页地址值

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0039 | 0C | B8 | OR | AL, | 0B8H ;"或"页面地址设置代码 | |
| 003B | 8A | E0 | MOV | AH,AL | ;页面地址设置 | |
| 003D | E8 | 002D | CALL | PRO |  | |
| 0040 | E8 | 0059 | CALL | PR3 |  | |
| 0043 | B4 | 00 | MOV | AH, | 00H ;列地址设置为"0" | |
| 0045 | E8 | 0025 | CALL | PRO |  | |
| 0048 | E8 | 0051 | CALL | PR3 |  | |
| 004B | C6 | 06 0008r 50 | MOV | R3, | 50H ;一页清 80 个字节 | |
| 0050 | C6 | 06 000Cr 00 | CLEAR2: MOV | | DAT,00H | ;显示数据为"0" |
| 0055 | E8 | 0024 | CALL PR1 | |  |  |
| 0058 | E8 | 0050 | CALL PR4 | |  |  |
| 005B | FE | 0E 0008r | DEC R3 | |  |  |

;页内字节清零循环

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 005F | 75 | EF | JNZ | CLEAR2 |
| 0061 | FE | 06 0007r | INC | R4 |
| 0065 | A0 | 0007r | MOV | AL,R4 |
| 0068 | 3C | 04 | CMP | AL,04H |

;页地址暂存器加一

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 006A | 75 | CA | JNE | CLEAR1 ;RAM 区清零循环 | | |
| 006C | C3 |  | RET |  | | |
| 006D CLEAR ENDP | | | | | | |
| 006D |  |  | ;1．写指令代码子程序（E1） PRO PROC NEAR | | |  |
| 006D | BA | 8002 | MOV DX,CRADD1 | | | ;设置读状态字地址 |
| 0070 | EC |  | PR01: IN AL,DX | | | ;读状态字 |
| 0071 | A8 | 80 | TEST | AL,80H |  | |
| 0073 | 75 | FB | JNZ | PR01 | ;判"忙"标志为句"0",否再读 | |
| 0075 | BA | 8000 | MOV | DX,CWADD1 | ;设置写指令代码地址 | |
| 0078 | 8A | C4 | MOV | AL,AH | ;取指令代码 | |
| 007A | EE |  | OUT | DX,AL | ;写指令代码 | |
| 007B | C3 |  | RET |  |  | |

;

007C PRO ENDP

;

;2.写显示数据子程序（E1）

007C PR1 PROC NEAR

007C BA 8002 MOV DX,CRADD1 ;设置读状态字地址

007F EC PR11: IN AL,DX ;读状态宇

0080 A8 80 TEST AL,80h

0082 75 FB JNZ PR11 ;判"忙"标志为"0",否再读

0084 BA 8001 MOV DX,DWADD1 ;设置写显示数据地址

0087 A0 000Cr MOV AL,DAT ;取数据

008A EE OUT DX,AL ;写数据

008B C3 RET

008C PR1 endp

;

;3.读显示数据子程序（E1）

008C PR2 PROC NEAR

008C BA 8002 MOV DX,CRADD1 ;设置读状态字地址

008F EC PR21: IN AL,DX ;读状态字

0090 A8 80 TEST AL,80H

0092 75 FB JNZ PR21 ;判"忙"标志为"0"否,否再读

0094 BA 8003 MOV DX,DRADD1 ;设置读显示数据地址

0097 EC IN AL,DX ;读数据

0098 A2 000Cr MOV DAT,AL ;存数据

009B C3 RET

009C PR2 ENDP

;

;4.写指令代码子程序（E2）

009C PR3 PROC NEAR

009C BA 8006 MOV DX,CRADD2 ;设置读状态字地址

009F EC PR31: IN AL,DX ;读状态字

00A0 A8 80 TEST AL,80H

00A2 75 FB JNZ PR31 ;判"忙"陈志为"0"否,否再读

00A4 BA 8004 MOV DX,CWADD2 ;设置写指令代码地址

00A7 8A C4 MOV AL,AH ;取指令代码

00A9 EE OUT DX,AL ;写指令代码

00AA C3 RET

00AB PR3 ENDP

;

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 00AB |  |  | ; | 5.写显示数据子程序（E2） PR4 proc near | | |
| 00AB | BA | 8006 |  | mov dx,CRADD2 ;设置读状态字地址 | | |
| 00AE | EC |  |  | PR41: IN al,dx ;读状态字 | | |
| 00AF | A8 | 80 | TEST | | AL,80H |  |
| 00B1 | 75 | FB | JNZ | | PR41 | ;判"忙"标志为"0"否,否再读 |
| 00B3 | BA | 8005 | MOV | | DX,DWADD2 ;设置写显示数据地址 | |

00B6 A0 000Cr MOV AL,DAT ;取数据

00B9 EE OUT dx,al ;写数据

00BA C3 RET

00BB PR4 endp

;

; 6.读显示数据子程序（E2）

00BB PR5 proc near

00BB BA 8006 mov dx,CRADD2 ;设置读状态字地址

00BE EC PR51: IN al,dx ;读状态字

00BF A8 80 TEST AL,80H

00C1 75 FB JNZ PR51 ;判"忙"标志为"0",否再读

00C3 BA 8007 mov dx,DRADD2 ;设置写显示数据地址

00C6 EC IN al,dx ;读数据

00C7 A2 000Cr MOV DAT,AL ;存数据

00CA C3 RET

00CB PR5 endp

;

; 中文显示子程序

00CB CCW\_PR proc near

00CB BA 000Dr mov dx,OFFSET CCTAB ;确定字符字模块首地址

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 00CE | A0 | 0004r | MOV | AL,CODE\_ | ;取代码 |
| 00D1 | B4 | 00 | MOV | AH,0 |  |
| 00D3 | B1 | 05 | MOV | CL,5 | ;字模块宽度为 32 个字节 |
| 00D5 | D3 | E0 | SHL | AX,CL | ;代码×32 |
| 00D7 | 03 | C2 | ADD | AX,DX | ;字符字模块首地址 |
| ;字模库首地址＋代码×32  00D9 A3 0000r MOV FONT,AX | | | | | |

;列地址

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 00DC | A0 | 0002r | MOV | AL,COLUMN |
| 00DF | A2 | 0009r | MOV | COL2,AL |
| 00E2 | C6 | 06 0004r 00 | MOV | CODE\_,00H |

;代码寄存器借用为间址寄存器00E7 C6 06 0005r 10 CCW\_1: MOV COUNT,10H ;计数器设置为 16

;读页地址寄存器

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 00EC | A0 | 0003r | MOV | AL,PAGE\_ |
| 00EF | 24 | 03 | AND | AL,03H |
| 00F1 | 0C | B8 | OR | AL,0B8H |
| 00F3 | 8A | E0 | MOV | ah,AL |
| 00F5 | E8 | FF75 | CALL | PRO |
| 00F8 | E8 | FFA1 | CALL | PR3 |
| 00FB | A0 | 0009r | MOV | AL,COL2 |
| 00FE | A2 | 0002r | MOV | COLUMN,AL |
| 0101 | A0 | 0002r | MOV | AL,COLUMN |

;取页地址有效值

;"或"页地址设置代码

;设置页地址

;取列地址值

;读列地址寄存器

0104 3C 3D CMP AL, PD1

0106 7C 0B JL CCW\_2 ;＜0 为左半屏显示区域（E1）

0108 A2 0002r MOV COLUMN,AL ;≥0 为右半屏显示区域（E2）

010B A0 0003r MOV AL,PAGE\_

010E 0C 08 OR AL,08H ;设置区域标志位。

0110 A2 0003r MOV PAGE\_,AL ;"0"为 E1,"1"为 E2

0113 8A 26 0002r CCW\_2: MOV ah,COLUMN ;设置列地址值

0117 A0 0003r MOV AL,PAGE\_ ;判区域标志以确定设置哪个控制器

011A A8 08 TEST AL,08H

011C 74 06 JZ CCW\_3

011E E8 FF7B CALL PR3 ;区域 E2

0121 EB 04 90 JMP CCW\_4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0124 | E8 | FF46 | CCW\_3: | CALL | PRO | ;区域 E1 |
| 0127 | A0 | 0004r | CCW\_4: | MOV | AL,CODE\_ | ;取间址寄存器值 |
| 012A | B4 | 00 | mov | ah,0 |  |  |
| 012C | 03 | 06 0000r | add | ax,FONT |  |  |
| 0130 | 8B | D8 | mov | bx,ax |  |  |

0132 8A 07 MOV Al,DS:[bx] ;取汉字字模数据

0134 A2 000Cr MOV DAT,Al ;写数据

0137 A0 0003r MOV Al,PAGE\_

013A A8 08 test al,08h

013C 74 06 Jz CCW\_5

013E E8 FF6A CALL PR4 ;区域 E2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0141 | EB | 04 90 | | JMP CCW\_6 | | |
| 0144 | E8 | FF35 | | CCW\_5: CALL PR1 ;区域 E1 | | |
| 0147 | FE | 06 0004r | | CCW\_6: INC CODE\_ ;间址寄存器加一 | | |
| 014B | FE | 06 0002r | | INC COLUMN ;列地址寄存器加一 | | |
| 014F | A0 | 0002r | | MOV Al,COLUMN ;判列地址是否超出区域范围 | | |
| 0152 | 3C | 3D | | cmp al, pd1 | | |
| 0154 | 7C | 11 | | CCW\_7: Jl CCW\_8 ;未超出则继续 | | |
| 0156 | A0 | 0003r | | MOV Al,PAGE\_ ;超出则判是否在区域 E2 | | |
| 0159 | A8 | 08 | | test al, 08h | | |
| 015B | 75 | 0A | | Jnz CCW\_8 ;在区域 E2 则退出 | | |
| 015D | 0C | 08 | | or Al,08h ;在区域 E1 则修改成区域 E2 | | |
| 015F | A2 | 0003r | | MOV PAGE\_,Al | | |
| 0162 | B4 | 00 | | MOV ah,00H ;设置区域 E2 列地址为"0" | | |
| 0164 | E8 | FF35 | | CALL PR3 | | |
| 0167 | FE | 0E 0005r | | CCW\_8: dec count | | |
| 016B | 75 | BA | | JNZ CCW\_4 ;当页循环 | | |
| 016D | A0 | 0003r | | MOV Al,PAGE\_ ;读页地址寄存器 | | |
| 0170 | A8 | 80 | | test al,80h | | |
| 0172 | 75 | 11 | | Jnz CCW\_9 ;判完成标志 D7 位,"1"则完成退出 | | |
| 0174 | FE | C0 | | INC Al ;否则页地址加一 | | |
| 0176 | 0C | 80 | | or al,80h ;置完成位为"1" | | |
| 0178 | 24 | F7 | | and al,0f7h | | |
| 017A | A2 | 0003r | | MOV PAGE\_,Al | | |
| 017D | C6 | 06 0004r 10 | | MOV CODE\_,10H ;间址寄存器设置为 16 | | |
| 0182 | E9 | FF62 | | JMP CCW\_1 ;大循环 | | |
| 0185 | C3 |  | | CCW\_9: RET | | |
| 0186 |  |  | | CCW\_PR endp | | |
| 0186 |  | ; 中 | | 文演示显示程序段  MAIN proc | | near |
| 0186 | B8 | 0000s |  |  | mov | ax, data |
| 0189 | 8E | D8 |  |  | mov | ds, ax |
| 018B | E8 | FE72 |  |  | CALL | INIT |
| 018E | E8 | FEA0 |  |  | CALL | CLEAR |
| 0191 | C6 | 06 000Ar | 00 |  | MOV | CTEMP,0 |
| 0196  019B  019B | C6  C6 | 06 0006r  06 0003r | 00  02 | AAA: | MOV  MOV | DIR,0  PAGE\_,02H |
| 01A0 | A0 | 000Ar |  |  | MOV | AL,CTEMP |
| 01A3 | A2 | 0002r |  |  | MOV | COLUMN,AL |
| 01A6 | C6 | 06 0004r | 00 |  | MOV | CODE\_,00H |
| 01AB | E8 | FF1D |  |  | CALL | CCW\_PR |
| 01AE | C6 | 06 0003r | 02 |  | MOV | PAGE\_,02H |
| 01B3 | A0 | 000Ar |  |  | MOV | Al,CTEMP |
| 01B6 | 04 | 10 |  |  | ADD | Al,10H |
| 01B8 | A2 | 0002r |  |  | MOV | COLUMN,Al |
| 01BB  01C0 | C6 E8 | 06 0004r FF08 | 01 |  | MOV  CAll | CODE\_,01H CCW\_PR |
| 01C3 | C6 | 06 0003r | 02 |  | MOV | PAGE\_,02H |
| 01C8 | A0 | 000Ar |  |  | MOV | Al,CTEMP |
| 01CB | 04 | 20 |  |  | ADD | Al,20H |
| 01CD | A2 | 0002r |  |  | MOV | COLUMN,Al |
| 01D0  01D5 | C6 E8 | 06 0004r FEF3 | 02 |  | MOV  CAll | CODE\_,02H CCW\_PR |
| 01D8 | C6 | 06 0003r | 02 |  | MOV | PAGE\_,02H |
| 01DD | A0 | 000Ar |  |  | MOV | AL,CTEMP |
| 01E0 | 04 | 30 |  |  | ADD | AL,30H |
| 01E2 | A2 | 0002r |  |  | MOV | COLUMN,AL |
| 01E5 | C6 | 06 0004r | 03 |  | MOV | CODE\_,03H |
| 01EA | E8 | FEDE |  |  | CALL | CCW\_PR |

; .

;延时子程序

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 01ED | EB FE JMP $ | | | |
| 01EF |  | MAIN | ENDP |  |
| 01EF |  | DELAY | PROC | NEAR |
| 01EF | C6 06 0007r 60 |  | MOV | R4,060H |
| 01F4 | C6 06 0008r 00 |  | MOV | R3,00H |
| 01F9 | 90 | DELAY1: | NOP |  |
| 01FA | FE 0E 0008r |  | DEC | R3 |
| 01FE | 75 F9 |  | JNZ | DELAY1 |
| 0200 | FE 0E 0007r |  | DEC | R4 |
| 0204 | 75 F3 |  | JNZ | DELAY1 |
| 0206 | C3 |  | RET |  |
| 0207 |  | DELAY | ENDP |  |
| 0207 |  | CODE | ENDS |  |

END main

; 硬件实验二十五 电子琴 H25.ASM

;

; 本实验用 8253 做定时器输出音频信号，控制喇叭发出声音

; 将 8253 的 CLK1 接到 1MHz 脉冲信号上（1MHz 信号可由 4MHz 信号

; 分频后得到），GATE1 接 VCC，OUT1 接喇叭的脉冲输入。

;

=C003 CONTROL equ 0c003h

=C000 COUNT0 equ 0c000h

=C001 COUNT1 equ 0c001h

=C002 COUNT2 equ 0c002h

=8002 OUTBIT equ 08002h

=8001 IN\_KEY equ 08001h

0000 data segment

0000 00 last db 0

0001 KeyTable: ; 键码定义

0001 16 15 14 FF db 16h, 15h, 14h, 0ffh

0005 13 12 11 10 db 13h, 12h, 11h, 10h

0009 0D 0C 0B 0A db 0dh, 0ch, 0bh, 0ah

000D 0E 03 06 09 db 0eh, 03h, 06h, 09h

0011 0F 02 05 08 db 0fh, 02h, 05h, 08h

0015 00 01 04 07 db 00h, 01h, 04h, 07h

0019 ToneTable:

0019 08E1 07E8 070D 06A8 dw 2273, 2024, 1805, 1704

0021 05ED 0549 04B5 0470 dw 1517, 1353, 1205, 1136

0029 data ends

0000 code segment

assume cs:code, ds:data

0000 TestKey proc near

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0000 | BA 8002 | mov | dx, OUTBIT |  |
| 0003 | B0 00 | mov | al, 0 |
| 0005 | EE | out | dx, al | ; 输出线置为 0 |
| 0006 | BA 8001 | mov | dx, IN\_KEY |  |
| 0009  000A  000C  000E | EC  F6 D0  24 0F C3 | in not and  ret | al, dx al  al, 0fh | ; 读入键状态  ; 高四位不用 |
| 000F |  | TestKey endp | | |
| 000F  000F | B5 20 | GetKey proc near  mov ch, 00100000b | | |
| 0011 | B1 06 | mov cl, 6 | | |

0013 KLoop:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0013 | BA | 8002 | | mov | dx, OUTBIT | | | |
| 0016 | 8A | C5 | | mov | al, ch ; 找出键所在列 | | | |
| 0018  001A  001B | F6 EE D0 | D0  ED | | not out shr | al dx, al ch, 1 | | | |
| 001D | BA | 8001 | | mov | dx, IN\_KEY | | | |
| 0020 | EC |  | | in | al, dx | | | |
| 0021 | F6 | D0 | | not | al | | | |
| 0023 | 24 | 0F | | and | al, 0fh | | | |
| 0025 | 75 | 09 | | jne | Goon\_ ; 该列有键入 | | | |
| 0027 | FE | C9 | | dec | cl | | | |
| 0029 | 75 | E8 | | jnz | KLoop | | | |
| 002B | B1 | FF | | mov | cl, 0ffh ; 没有键按下, 返回 0ffh | | | |
| 002D  0030 | EB | 15 | 90 | jmp  Goon\_: | Exit1 |  |  |  |
| 0030 | FE | C9 |  | dec | cl |  |  |  |
| 0032 | D0 | E1 | D0 E1 | shl | cl, 2 | ; | 键值 = 列 X 4 + | 行 |
| 0036 | B5 | 04 |  | mov | ch, 4 |  |  |  |
| 0038 |  |  |  | LoopC: |  |  |  |  |
| 0038 | A8 | 01 |  | test | al, 1 |  |  |  |
| 003A  003C  003E  0040 | 75  D0  FE FE | 08  E8  C1 CD | | jnz Exit1  shr al, 1  inc cl  dec ch | |  | | |
| 0042 | 75 | F4 | | jnz LoopC | |  | | |
| 0044 |  |  | | Exit1: | |  | | |
| 0044 | BA | 8002 | | mov dx, OUTBIT | |  | | |
| 0047 | B0 | 00 | | mov al, 0 | |  | | |
| 0049 | EE |  | | out dx, al | |  | | |
| 004A  004C  004F  0051 | B5 BB 03  8A | 00  0001r D9 07 | | mov ch, 0  mov bx, offset  add bx, cx  mov al, [bx] | | KeyTable  ; 取出键码 | | |
| 0053 | C3 |  | | ret | |  | | |
| 0054 |  |  | | GetKey endp | |  | | |
| 0054 |  |  | | OutTone proc near | |  | | |
| 0054 | 50 |  | | push ax | |  | | |
| 0055 | B0 | 76 | | mov al, 76h | | ; 计数器 1, 16 位二进制,方式 3。 | | |
| 0057 | BA | C003 | | mov dx, CONTROL | |  | | |
| 005A | EE |  | | out dx, al | |  | | |
| 005B  005C  005F  0060 | 58  BA  EE 8A | C001 C4 | | pop ax  mov dx, COUNT1 out dx, al  mov al, ah | |  | | |
| 0062 | EE |  | | out dx, al | |  | | |
| 0063 | C3 |  | | ret | |  | | |
| 0064 |  |  | | OutTone endp | |  | | |
| 0064 |  |  | | delay proc near | | ; | | |
| 0064 | 51 |  | | push cx | |  | | |
| 0065 | B9 | EA60 | | mov cx,60000 | |  | | |
| 0068 | E2 | FE | | loop $ | |  | | |
| 006A  006B  006C | 59  C3 |  | | pop cx ret  delay endp | |  | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 006C  006C | B8 | 0000s | start proc  mov | near  ax, data |  | |
| 006F  0071 | 8E | D8 | mov MLoop: | ds, ax |
| 0071 | E8 | FF8C | call | TestKey | ; 有键入? | |
| 0074 | 74 | FB | jz | MLoop | ; 无键入, 继续显示 | |
| 0076 | E8 | FF96 | call | GetKey | ; 读入键码 | |
| 0079  007B  007D  007F | 3C  7C  3C  7F | 01  F4 08  F0 | cmp jl cmp  jg | al, 1 MLoop al, 8  MLoop | ; 是否在 1~8 范围内 | |
| 0081 | B4 | 00 | mov | ah, 0 | | |
| 0083 | 48 |  | dec | ax | | |
| 0084 | D1 | E0 | shl | ax,1 | | |
| 0086 | 8B | F0 | mov | si, ax | | |
| 0088 | BB | 0019r | mov | bx, offset ToneTable | | |
| 008B | 8B | 00 | mov | ax, [bx+si] ; 查找音频值 | | |
| 008D | E8 | FFC4 | call | OutTone | ; |  |
| 0090 | E8 | FFD1 | call | delay |  | ; 延时 |
| 0093 | B8 | 0002 | mov | ax, 2 |  | ; 关闭发音 |
| 0096 | E8 | FFBB | call | OutTone |  |  |
| 0099 | EB | D6 | jmp | MLoop |  |  |
| 009B | start endp | | | | | |

009B code ends

end start

; 硬件实验二十六 空调温度控制实验 H26.ASM

=8000 ADPort equ 08000h ; AD 采样输入片选

=A000 CS273 equ 0a000h ; 控制输出片选

=FE02 OUTBIT equ 0FE02h ; 位控制口

=FE04 CLK164 equ 0FE04h ; 段控制口(接 164 时钟位)

=FE06 DAT164 equ 0FE06h ; 段控制口(接 164 数据位)

=FE00 IN\_KEY equ 0FE00h ; 键盘读入口

=0016 UP equ 16h ; Next

=0015 DOWN equ 15h ; Last

=000A LowLimit equ 10

=001E HighLimit equ 30

=-0028 LowTemp equ -40 ; A/D 0

=0032 HighTemp equ 50 ; A/D 255

=0001 Heat equ 1 ; 加热控制

=0002 Cool equ 2 ; 致冷控制

0000 data segment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0000 | 06\*(??) | LEDBuf | db |  | 6 dup(?) ; 显示缓冲 |
| 0006 | 00 | SetTemp db |  | 0 | ; 设定温度 |
| 0007 | 00 | CurTemp db |  | 0 | ; 当前温度 |
| 0008 | 00 | P1Buf db |  | 0 | ; 输出缓冲 |

0009 LEDMAP: ; 八段管显示码

0009 3F 06 5B 4F 66 6D 7D+ db 3fh, 06h, 5bh, 4fh, 66h, 6dh, 7dh, 07h

07

0011 7F 6F 77 7C 39 5E 79+ db 7fh, 6fh, 77h, 7ch, 39h, 5eh, 79h, 71h

71

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0019 |  |  | KeyTable: | ; 键码定义 |
| 0019 | 16 15 14 | FF | db | 16h, 15h, 14h, 0ffh |
| 001D | 13 12 11 | 10 | db | 13h, 12h, 11h, 10h |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0021 | 0D | 0C 0B 0A | db | 0dh, 0ch, 0bh, 0ah |
| 0025 | 0E | 03 06 09 | db | 0eh, 03h, 06h, 09h |
| 0029 | 0F | 02 05 08 | db | 0fh, 02h, 05h, 08h |
| 002D | 00 | 01 04 07 | db | 00h, 01h, 04h, 07h |
| 0031 |  |  | data ends |  |

0000 code segment

assume cs:code, ds:data

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0000 |  | Delay proc near | | | |  |
| 0000 | 50 | push ax | | | | ; 延时子程序 |
| 0001 | 51 | push cx | | | |  |
| 0002 | B0 | 00 | mov |  | al, 0 | |
| 0004 | 8B | C8 | mov |  | cx,ax | |
| 0006 | E2 | FE | loop | $ |  | |
| 0008 | 59 |  | pop cx | | | |
| 0009  000A  000B | 58  C3 |  | pop ax ret  Delay endp | | | |
| 000B |  |  | DisplayLED proc near | | | |
| 000B | BB | 0000r | mov bx, offset LEDBuf | | | |
| 000E | B1 | 06 | mov cl, 6 ; 共 6 个八段管 | | | |
| 0010 | B4 | 20 | mov ah, 00100000b ; 从左边开始显示 | | | |
| 0012 |  |  | DLoop: | | | |
| 0012 | BA | FE02 | mov dx, OUTBIT | | | |
| 0015 | B0 | 00 | mov al, 0 | | | |
| 0017 | EE |  | out dx,al ; 关所有八段管 | | | |
| 0018 | 8A | 07 | mov al, [bx] | | | |
| 001A  001C  001C | B5  50 | 08 | mov ch, 8 ; 送 164 DLP:  push ax | | | |
| 001D | B4 | 00 | mov ah, 0 | | | |
| 001F | A8 | 80 | test al, 80h | | | |
| 0021 | 74 | 02 | jz GN1 | | | |
| 0023 | B4 | 01 | mov ah, 1 | | | |
| 0025 |  |  | GN1: | | | |
| 0025 | BA | FE06 | mov dx, DAT164 ; 输出数据位 | | | |
| 0028  002A  002B | 8A  EE BA | C4  FE04 | mov al, ah  out dx, al  mov dx, CLK164 ; 送出时钟脉冲 | | | |
| 002E  0030 | B0 EE | 01 | mov al, 1  out dx, al | | | |
| 0031 | B0 | 00 | mov al, 0 | | | |
| 0033 | EE |  | out dx, al | | | |
| 0034 | 58 |  | pop ax | | | |
| 0035 | D0 | E0 | shl al, 1 | | | |
| 0037 | FE | CD | dec ch | | | |
| 0039 | 75 | E1 | jnz DLP | | | |
| 003B | BA | FE02 | mov dx, OUTBIT | | | |
| 003E  0040 | 8A EE | C4 | mov al, ah  out dx, al ; 显示一位八段管 | | | |
| 0041 | 50 |  | push ax | | | |
| 0042 | B4 | 01 | mov ah, 1 | | | |
| 0044 | E8 | FFB9 | call Delay | | | |
| 0047 | 58 |  | pop ax | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0048 | D0 | EC | shr | ah, 1 | |
| 004A | 43 |  | inc | bx | |
| 004B | FE | C9 | dec | cl | |
| 004D | 75 | C3 | jnz | DLoop | |
| 004F | BA | FE02 | mov | dx, OUTBIT | |
| 0052 | B0 | 00 | mov | al, 0 | |
| 0054 | EE |  | out | dx,al ; 关所有八段管 | |
| 0055 | C3 |  | ret |  | |
| 0056 |  |  | DisplayLED | endp | |
| 0056 TestKey proc near | | | | | |
| 0056 | BA | FE02 | mov | dx, OUTBIT |  |
| 0059 | B0 | 00 | mov | al, 0 |  |
| 005B | EE |  | out | dx, al | ; 输出线置为 0 |
| 005C | BA | FE00 | mov | dx, IN\_KEY |  |
| 005F | EC |  | in | al, dx | ; 读入键状态 |
| 0060 | F6 | D0 | not | al |  |
| 0062 | 24 | 0F | and | al, 0fh | ; 高四位不用 |
| 0064 | C3 |  | ret |  |  |
| 0065 |  |  | TestKey endp | |  |
| 0065 |  |  | GetKey proc near | |  |
| 0065 | B5 | 20 | mov ch, 00100000b | |  |
| 0067 | B1 | 06 | mov cl, 6 | |  |
| 0069 |  |  | KLoop: | |  |
| 0069 | BA | FE02 | mov dx, OUTBIT | |  |
| 006C | 8A | C5 | mov al, ch | | ; 找出键所在列 |
| 006E | F6 | D0 | not al | |  |
| 0070 | EE |  | out dx, al | |  |
| 0071 | D0 | ED | shr ch, 1 | |  |
| 0073 | BA | FE00 | mov dx, IN\_KEY | |  |
| 0076 | EC |  | in al, dx | |  |
| 0077 | F6 | D0 | not al | |  |
| 0079 | 24 | 0F | and al, 0fh | |  |
| 007B | 75 | 09 | jne Goon\_ | | ; 该列有键入 |
| 007D | FE | C9 | dec cl | |  |
| 007F | 75 | E8 | jnz KLoop | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0081 | B1 | FF | mov | | cl, 0ffh ; 没有键按下, 返回 0ffh |
| 0083 | EB | 15 | 90 | jmp | Exit1 |
| 0086 |  |  |  | Goon\_: |  |
| 0086 | FE | C9 |  | dec | cl |
| 0088 | D0 | E1 | D0 E1 | shl | cl, 2 ; 键值 = 列 X 4 + 行 |
| 008C  008E  008E | B5  A8 | 04  01 |  | mov LoopC:  test | ch, 4  al, 1 |
| 0090 | 75 | 08 | | jnz | Exit1 |
| 0092 | D0 | E8 | | shr | al, 1 |
| 0094 | FE | C1 | | inc | cl |
| 0096 | FE | CD | | dec | ch |
| 0098  009A  009A | 75  BA | F4  FE02 | | jnz Exit1:  mov | LoopC  dx, OUTBIT |
| 009D  009F  00A0 | B0 EE  B5 | 00  00 | | mov out  mov | al, 0 dx, al  ch, 0 |
| 00A2 | BB | 0019r | | mov | bx, offset KeyTable |
| 00A5 | 03 | D9 | | add | bx, cx |
| 00A7 | 8A | 07 | | mov | al, [bx] ; 取出键码 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 00A9 | 8A | D8 | mov | bl, al |
| 00AB |  |  | WaitRelease: | |
| 00AB | BA | FE02 | mov dx, OUTBIT | |
| 00AE | B0 | 00 | mov al, 0 | |
| 00B0 | EE |  | out dx, al ; 等键释放 | |
| 00B1 | B4 | 0A | mov ah, 10 | |
| 00B3 | E8 | FF4A | call Delay | |
| 00B6 | E8 | FF9D | call TestKey | |
| 00B9 | 75 | F0 | jne WaitRelease | |
| 00BB | 8A | C3 | mov al, bl | |
| 00BD | C3 |  | ret | |
| 00BE |  |  | GetKey endp | |
| ; =================================== | | | | |
| 00BE |  |  | DisplayResult proc near | |
| 00BE | A0 | 0007r | mov al, CurTemp | |
| 00C1 | A8 | 80 | test al, 80h | |
| 00C3 | 74 | 0C | jz GE0 | |
| 00C5 | C6 | 06 0000r 40 | mov LEDBuf, 40h ; '-' | |
| 00CA | FE | C8 | dec al | |
| 00CC | F6 | D0 | not al | |
| 00CE | EB | 06 90 | jmp Goon | |
| 00D1 |  |  | GE0: | |
| 00D1 | C6 | 06 0000r 00 | mov LEDBuf, 0 ; ' ' | |
| 00D6 |  |  | Goon: | |
| 00D6 | B1 | 0A | mov cl, 10 | |
| 00D8 | B4 | 00 | mov ah, 0 | |
| 00DA | F6 | F1 | div cl | |
| 00DC | 8A | D8 | mov bl, al | |
| 00DE | B7 | 00 | mov bh, 0 | |
| 00E0 | 81 | C3 0009r | add bx, offset LEDMAP | |
| 00E4 | 8A | 07 | mov al, [bx] | |
| 00E6 | A2 | 0001r | mov LEDBuf+1, al | |
| 00E9 | 8A | DC | mov bl, ah | |
| 00EB | B7 | 00 | mov bh, 0 | |
| 00ED | 81 | C3 0009r | add bx, offset LEDMAP | |
| 00F1 | 8A | 07 | mov al, [bx] | |
| 00F3 | A2 | 0002r | mov LEDBuf+2, al | |
| 00F6 | C6 | 06 0003r 00 | mov LEDBuf+3, 0 ; ' ' | |
| 00FB | A0 | 0006r | mov al, SetTemp | |
| 00FE | B4 | 00 | mov ah, 0 | |
| 0100 | B1 | 0A | mov cl, 10 | |
| 0102 | F6 | F1 | div cl | |
| 0104 | 8A | D8 | mov bl, al | |
| 0106 | B7 | 00 | mov bh, 0 | |
| 0108 | 81 | C3 0009r | add bx, offset LEDMAP | |
| 010C | 8A | 07 | mov al, [bx] | |
| 010E | A2 | 0004r | mov LEDBuf+4, al | |
| 0111 | 8A | DC | mov bl, ah | |
| 0113 | B7 | 00 | mov bh, 0 | |
| 0115 | 81 | C3 0009r | add bx, offset LEDMAP | |
| 0119 | 8A | 07 | mov al, [bx] | |
| 011B | A2 | 0005r | mov LEDBuf+5, al | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 011E  011F  011F | C3 |  | | ret DisplayResult endp  ReadAD proc near | | | | | |
| 011F | BA | 8000 | | mov dx, ADPort | | | | | |
| 0122 | B0 | 00 | | mov al, 0 | | | | | |
| 0124 | EE |  | | out dx, al | | | | | |
| 0125 | B0 | 3C | | mov al, 60 | | | | | |
| 0127 |  |  | | rr: | | | | | |
| 0127 | FE | C8 | | dec al ; delay | | | | | |
| 0129 | 75 | FC | | jnz rr | | | | | |
| 012B | BA | 8000 | | mov dx, ADPort | | | | | |
| 012E  012F  0130 | EC C3 |  | | in al, dx ret  ReadAD endp | | | | | |
| 0130 |  |  | | ReadTemp proc near | | | | | |
| 0130 | BB | 0000 | | mov bx, 0 | | | | | |
| 0133 | B1 | 10 | | mov cl, 16 | | | | | |
| 0135 |  |  | | RLoop: | | | | | |
| 0135 | E8 | FFE7 | | call ReadAD | | | | | |
| 0138 | B4 | 00 | | mov ah, 0 | | | | | |
| 013A | 03 | D8 | | add bx, ax | | | | | |
| 013C | FE | C9 | | dec cl | | | | | |
| 013E | 75 | F5 | | jnz RLoop | | | | | |
| 0140 | D1 EB | EB | D1 EB | D1 EB | | D1+ | shr | bx, 4 | |
| 0148  014A  014C  014E  0150 | 8B  B1 F6 8A  04 | C3  5A E1 C4  D8 | | mov  mov mul mov  add | | | ax, bx  cl, HighTemp cl  al, ah  al, LowTemp | | - LowTemp  ; /256 |
| 0152 | A2 | 0007r | | mov | | | CurTemp, al | |  |
| 0155 | C3 |  | | ret | | |  | |  |
| 0156 |  |  | | ReadTemp | | | endp | |  |
| 0156 |  |  | | | Start proc near | | | | |
| 0156 | B8 | 0000s | | | mov ax, data | | | | |
| 0159 | 8E | D8 | | | mov ds, ax | | | | |
| 015B  0160 | C6 | 06 0006r 14 | | | mov SetTemp, 20 MLoop: | | | | |
| 0160 | E8 | FEF3 | | | call TestKey | | | | |
| 0163 | 75 | 4B | | | jne KeyPressed | | | | |

0165 E8 FF56 call DisplayResult

0168 E8 FEA0 call DisplayLED

016B E8 FFC2 call ReadTemp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 016E | A0 | 0006r | mov | al, SetTemp |
| 0171 | FE | C8 | dec | al |
| 0173 | FE | C8 | dec | al |
| 0175 | 38 | 06 0007r | cmp | CurTemp, al |
| 0179 | 7D | 0D | jge | GN2 |
| 017B | 80 | 0E 0008r 01 | or | P1Buf, Heat |
| 0180 | 80 | 26 0008r FD | and | P1Buf, not Cool |
| 0185 | EB | 20 90 | jmp | GN4 |
| 0188 |  |  | GN2: |  |
| 0188 | A0 | 0006r | mov | al, SetTemp |
| 018B | FE | C0 | inc | al |
| 018D | FE | C0 | inc | al |
| 018F | 38 | 06 0007r | cmp | CurTemp, al |
| 0193 | 7E | 0D | jle | GN3 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0195 | 80 | 0E 0008r 02 | | or P1Buf, Cool | |
| 019A | 80 | 26 0008r FE | | and P1Buf, not Heat | |
| 019F  01A2  01A2 | EB  80 | 06 90  26 0008r FC | | jmp GN4 GN3:  and P1Buf, not (Cool+Heat) | |
| 01A7 |  |  | | GN4: | |
| 01A7 | BA | A000 | | mov dx, CS273 | |
| 01AA  01AD  01AE  01B0  01B0 | A0 EE EB  E8 | 0008r  B0 FEB2 | | mov al, P1Buf  out dx, al  jmp MLoop KeyPressed:  call GetKey | |
| 01B3  01B5 | 3C  75 | 15  0E |  | cmp jne | al, DOWN  Key0 |
| 01B7 | 80 | 3E | 0006r 0A | cmp | SetTemp, LowLimit |
| 01BC | 74 | 16 |  | je | Key1 |
| 01BE | FE | 0E | 0006r | dec | SetTemp |
| 01C2  01C5  01C5 | EB  3C | 10  16 | 90 | jmp Key0:  cmp | Key1  al, UP |
| 01C7 | 75 | 0B |  | jne | Key1 |
| 01C9 | 80 | 3E | 0006r 1E | cmp | SetTemp, HighLimit |
| 01CE | 74 | 04 |  | je | Key1 |
| 01D0  01D4  01D4 | FE  EB | 06  8A | 0006r | inc Key1:  jmp | SetTemp  MLoop |
| 01D6 | start | | | | endp |
| 01D6 | code | | | | ends |

end start

; 硬件实验三十三 PWM 转换电压实验 H33.ASM

; 输出 50 (5:5) 占空比 PWM

; 输出 10 (1:9) 占空比 PWM

; 输出 90 (9:1) 占空比 PWM

=0082 mode equ 082h

=9000 PA equ 09000h

=9003 CTL equ 09003h

=0001 hi\_t equ 1

=0009 lo\_t equ 9

0000 code segment assume cs:code

|  |  |  |
| --- | --- | --- |
| 0000 |  | Delay proc near |
| 0000 | 50 | push ax |
| 0001 | 51 | push cx |
| 0002 | B0 00 | mov al, 0 |
| 0004 | D1 E8 | shr ax, 1 |
| 0006 | 8B C8 | mov cx, ax |
| 0008 |  | dloop: |
| 0008 | E2 FE | loop dloop |
| 000A | 59 | pop cx |
| 000B | 58 | pop ax |
| 000C  000D | C3 | ret Delay endp |
| 000D  000D | BA 9003 | start proc near mov dx, CTL |
| 0010 | B0 82 | mov al, mode |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0012 | EE | out | dx, | al |
| 0013 |  | Loop: |  |  |
| 0013 | BA 9000 | mov | dx, | PA |
| 0016 | B0 01 | mov | al, | 1 |
| 0018 | EE | out | dx, | al |
| 0019  001B | B4 01 E8 FFE2 | mov call | ah, Delay | hi\_t |
| 001E | BA 9000 | mov | dx, | PA |
| 0021 | B0 00 | mov | al, | 0 |
| 0023 | EE | out | dx, | al |
| 0024 | B4 09 | mov | ah, | lo\_t |
| 0026 | E8 FFD7 | call | Delay |  |
| 0029 | EB E8 | jmp | Loop |  |
| 002B | start endp | | | |
| 002B | code ends | | | |

end start

; 硬件实验三十四 8253 计数器实验 H34.ASM

=8003 CONTROL equ 08003h

=8000 COUNT0 equ 08000h

=8001 COUNT1 equ 08001h

=8002 COUNT2 equ 08002h

0000 code segment assume cs:code

0000 start proc near

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0000 | B0 | 30 | mov | al, | 30h ; 通道 0，方式 0 |
| 0002 | BA | 8003 | mov | dx, | CONTROL |
| 0005 | EE |  | out | dx, | al |
| 0006 | B0 | 05 | mov | al, | 5 ; 计数器初始值。 |
| 0008 | BA | 8000 | mov | dx, | COUNT0 |
| 000B  000C | EE  B0 | 00 | out  mov | dx,  al, | al ; 低八位  0 |
| 000E  000F  000F | EE  B0 | 00 | out Again:  mov | dx,  al, | al ; 高八位  00000000B ; 锁存计数器值 |
| 0011 | BA | 8003 | mov | dx, | CONTROL |
| 0014 | EE |  | out | dx, | al |
| 0015 | BA | 8000 | mov | dx, | COUNT0 |
| 0018 | EC |  | in | al, | dx ; 读入计数值低八位 |
| 0019  001B  001C | 8A EC 8A | D8  E0 | mov in mov | bl,  al,  ah, | al  dx ; 读入计数值高八位  al |
| 001E | 8A | C3 | mov | al, | bl |
| 0020 | EB | ED | jmp | Again | |
| 0022 |  |  | start endp |  | |
| 0022 |  |  | code ends |  | |

end start

; 硬件实验三十五 8259 外部中断实验 H35.ASM

|  |  |  |  |
| --- | --- | --- | --- |
| =0082 | mode equ | 82h | ; 8255 工作方式 |
| =8000 | PA8255 equ | 8000h | ; 8255 PA 口输出地址 |
| =8003 | CTL8255 equ | 8003h |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | =0013  =0020  =0001  =00FE  =D000  =D001 | ICW1 ICW2 ICW4 OCW1 CS8259A CS8259B | equ equ equ equ equ equ | 00010011b  00100000b  00000001b  11111110b  0d000h 0d001h | ; 单片 8259, 上升沿中断, 要写 ICW4  ; 中断号为 20H  ; 工作在 8086/88 方式  ; 只响应 INT0 中断  ; 8259 地址 |
| 0000 |  | data | segment |  |  |
| 0000 | 00 | CNT | db 0 |  |  |
| 0001 |  | data | ends |  |  |
| 0000 |  | code | segment |  |  |

assume cs:code, ds: data

|  |  |  |
| --- | --- | --- |
| 0000 |  | IEnter proc near |
| 0000 | 50 | push ax |
| 0001 | 52 | push dx |

0002 BA 8000 mov dx, PA8255

0005 FE 06 0000r inc CNT

0009 A0 0000r mov al, CNT

000C EE out dx, al ; 输出计数值

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 000D | BA | D000 | mov | dx, CS8259A |
| 0010 | B0 | 20 | mov | al, 20h ; 中断服务程序结束指令 |
| 0012 | EE |  | out | dx, al |
| 0013 | 5A |  | pop | dx |
| 0014 | 58 |  | pop | ax |
| 0015 | CF |  | iret |  |
| 0016 |  |  | IEnter endp |  |
| 0016 |  |  | IInit proc |  |
| 0016 | BA | D000 | mov | dx, CS8259A |
| 0019  001B | B0 EE | 13 | mov out | al, ICW1  dx, al |
| 001C | BA | D001 | mov | dx, CS8259B |
| 001F  0021 | B0 EE | 20 | mov out | al, ICW2  dx, al |
| 0022 | B0 | 01 | mov | al, ICW4 |
| 0024 | EE |  | out | dx, al |
| 0025 | B0 | FE | mov | al, OCW1 |
| 0027 | EE |  | out | dx, al |
| 0028 | C3 |  | ret |  |
| 0029 |  |  | IInit endp |  |
| 0029 |  |  | start proc | near |
| 0029 | BA | 8003 | mov | dx, CTL8255 |
| 002C  002E | B0 EE | 82 | mov out | al, mode dx, al |
| 002F | FA |  | cli |  |
| 0030 | B8 | 0000 | mov | ax, 0 |
| 0033 | 8E | D8 | mov | ds, ax |
| 0035 | BB | 0080 | mov | bx, 4\*ICW2 ; 中断号 |
| 0038 | B8 | 0000s | mov | ax, code |

003B D1 E0 D1 E0 D1 E0 D1+ shl ax, 4 ; x 16 E0

0043 05 0000r add ax, offset IEnter ; 中断入口地址（段地址为 0）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0046 | 89 | 07 |  | mov | [bx], ax | |  |
| 0048 | B8 | 0000 |  | mov | ax, 0 | |
| 004B | 43 |  |  | inc | bx | |
| 004C | 43 |  |  | inc | bx | |
| 004D | 89 | 07 |  | mov | [bx], ax | | ; 代码段地址为 0 |
| 004F | E8 | FFC4 |  | call | IInit | |  |
| 0052 | B8 | 0000s |  | mov | ax, data | |  |
| 0055 | 8E | D8 |  | mov | ds, ax | |  |
| 0057  005C  005F  0062 | C6 A0 BA  EE | 06 0000r 00  0000r  8000 |  | mov mov mov  out | CNT, 0  al, CNT dx, PA8255  dx, al | | ; 计数值初始为 0 |
| 0063 | FB |  |  | sti |  | |  |
| 0064 |  |  | LP: |  | ; | | 等待中断，并计数。 |
| 0064 | 90 |  | nop | | |  | |
| 0065 | EB | FD | jmp | | | LP | |
| 0067 |  |  | start endp | | |  | |
| 0067 |  |  | code ends | | |  | |

end start

; 硬件实验三十六 8253 定时器实验 H36.ASM

=C003 CONTROL equ 0c003h

=C000 COUNT0 equ 0c000h

=C001 COUNT1 equ 0c001h

=C002 COUNT2 equ 0c002h

0000 code segment assume cs:code

0000 start proc near

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0000 | B0 | 36 | mov | al, | 36h; 00110110B ; 计数器 0,16 位,方式 3,二进制 | |
| 0002 | BA | C003 | mov | dx, | CONTROL | |
| 0005 | EE |  | out | dx, | al | |
| 0006 | B8 | 03E8 | mov | ax, | 1000 |  |
| 0009 | BA | C000 | mov | dx, | COUNT0 |  |
| 000C  000D  000F | EE 8A EE | C4 | out mov out | dx,  al,  dx, | al ah al | ; 计数器低字节  ; 计数器高字节 |
| 0010 | B0 | 76 | mov | al, | 76h; 01110110B ; 计数器 1,16 位,方式 3,二进制 | |
| 0012 | BA | C003 | mov | dx, | CONTROL | |
| 0015 | EE |  | out | dx, | al | |
| 0016 | B8 | 03E8 | mov | ax, | 1000 | |
| 0019 | BA | C001 | mov | dx, | COUNT1 | |
| 001C  001D  001F | EE 8A EE | C4 | out mov out | dx,  al,  dx, | al ; 计数器低字节  ah  al ; 计数器高字节 | |
| 0020 | EB | FE | jmp | $ |  | |
| 0022 |  |  | start endp |  |  | |
| 0022 |  |  | code ends |  |  | |

end start

; 硬件实验三十七 8251A 串行通讯实验 H37.ASM

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| =8002 | | | | OUTBIT | | equ 08002h | | ; 位控制口 | |
| =8004 | | | | OUTSEG | | equ 08004h | | ; 段控制口 | |
| =8001 | | | | IN\_KEY | | equ 08001h | | ; 键盘读入口 | |
| =D000 | | | | CS8251D | | equ 0d000h | | ; 串行通信控制器数据口地址 | |
| =D001 | | | | CS8251C | | equ 0d001h | | ; 串行通信控制器控制口地址 | |
| 0000 |  | | | data segment | | | | | |
| 0000 | 06\*(??) | | | LEDBuf db 6 dup(?) ; 显示缓冲 | | | | | |
| 0006 | 01\*(??) | | | Num db 1 dup(?) ; 显示的数据 | | | | | |
| 0007 | 01\*(??) | | | DelayT db 1 dup(?) | | | | | |
| 0008 | 00 | | | RBuf db 0 | | | | | |
| 0009  000A | 00 | | | TBuf db 0  LEDMAP: ; 八段管显示码 | | | | | |
| 000A  0012 | 3F  07  7F | 06  6F | 5B 4F 66 6D  77 7C 39 5E | | 7D+  79+ | db  db | | 3fh, 06h, 5bh, 4fh, 66h, 6dh,  7fh, 6fh, 77h, 7ch, 39h, 5eh, | 7dh, 07h  79h, 71h |
|  | 71 |  |  | |  |  | |  |  |
| 001A |  |  |  | KeyTable: | | | ; 键码定义 | | |
| 001A | 16 | 15 | 14 FF | db | | | 16h, 15h, 14h, 0ffh | | |
| 001E | 13 | 12 | 11 10 | db | | | 13h, 12h, 11h, 10h | | |
| 0022 | 0D | 0C | 0B 0A | db | | | 0dh, 0ch, 0bh, 0ah | | |
| 0026 | 0E | 03 | 06 09 | db | | | 0eh, 03h, 06h, 09h | | |
| 002A | 0F | 02 | 05 08 | db | | | 0fh, 02h, 05h, 08h | | |
| 002E | 00 | 01 | 04 07 | db | | | 00h, 01h, 04h, 07h | | |

0000 code segment

assume cs:code, ds:data

|  |  |  |  |
| --- | --- | --- | --- |
| 0000 |  | Delay proc near |  |
| 0000 | 50 | push ax | ; 延时子程序 |
| 0001 | 51 | push cx |  |

0002 B0 00 mov al, 0

0004 8B C8 mov cx,ax

0006 E2 FE loop $

|  |  |  |  |
| --- | --- | --- | --- |
| 0008 | 59 |  | pop cx |
| 0009  000A  000B | 58  C3 |  | pop ax ret  Delay endp |
| 000B |  |  | DisplayLED proc near |
| 000B | BB | 0000r | mov bx, offset LEDBuf |
| 000E | B1 | 06 | mov cl, 6 ; 共 6 个八段管 |
| 0010 | B4 | 20 | mov ah, 00100000b ; 从左边开始显示 |
| 0012 |  |  | DLoop: |
| 0012 | BA | 8002 | mov dx, OUTBIT |
| 0015 | B0 | 00 | mov al, 0 |
| 0017 | EE |  | out dx,al ; 关所有八段管 |
| 0018 | 8A | 07 | mov al, [bx] |
| 001A | BA | 8004 | mov dx, OUTSEG |
| 001D | EE |  | out dx, al |
| 001E | BA | 8002 | mov dx, OUTBIT |
| 0021 | 8A | C4 | mov al, ah |
| 0023 | EE |  | out dx, al ; 显示一位八段管 |
| 0024 | 50 |  | push ax |
| 0025 | B4 | 01 | mov ah, 1 |
| 0027 | E8 | FFD6 | call Delay |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 002A | 58 |  | pop | ax | |
| 002B  002D  002E | D0 43 FE | EC  C9 | shr inc dec | ah, 1 bx cl | |
| 0030 | 75 | E0 | jnz | DLoop | |
| 0032 | BA | 8002 | mov | dx, OUTBIT | |
| 0035 | B0 | 00 | mov | al, 0 | |
| 0037 | EE |  | out | dx,al ; 关所有八段管 | |
| 0038 | C3 |  | ret |  | |
| 0039 |  |  | DisplayLED | endp | |
| 0039 TestKey proc near | | | | | |
| 0039 | BA | 8002 | mov | dx, OUTBIT |  |
| 003C | B0 | 00 | mov | al, 0 |  |
| 003E | EE |  | out | dx, al | ; 输出线置为 0 |
| 003F | BA | 8001 | mov | dx, IN\_KEY |  |
| 0042 | EC |  | in | al, dx | ; 读入键状态 |
| 0043 | F6 | D0 | not | al |  |
| 0045 | 24 | 0F | and | al, 0fh | ; 高四位不用 |
| 0047 | C3 |  | ret |  |  |
| 0048 |  |  | TestKey endp | | |
| 0048 |  |  | GetKey proc near | | |
| 0048 | B5 | 20 | mov ch, 00100000b | | |
| 004A  004C  004C | B1  BA | 06  8002 | mov cl, 6 KLoop:  mov dx, OUTBIT | | |
| 004F | 8A | C5 | mov al, ch ; 找出键所在列 | | |
| 0051 | F6 | D0 | not al | | |
| 0053 | EE |  | out dx, al | | |
| 0054 | D0 | ED | shr ch, 1 | | |
| 0056 | BA | 8001 | mov dx, IN\_KEY | | |
| 0059  005A | EC F6 | D0 | in al, dx  not al | | |
| 005C | 24 | 0F | and al, 0fh | | |
| 005E | 75 | 09 | jne Goon\_ ; 该列有键入 | | |
| 0060 | FE | C9 | dec cl | | |
| 0062 | 75 | E8 | jnz KLoop | | |

没有键按下, 返回 0ffh

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0064 | B1 | FF |  | mov | cl, 0ffh | ; |  |
| 0066 | EB | 15 | 90 | jmp | Exit1 |  |
| 0069 |  |  |  | Goon\_: |  |  |
| 0069  006B | FE D0 | C9 E1 | D0 E1 | dec shl | cl cl, 2 |  | ; |
| 006F  0071 | B5 | 04 |  | mov LoopC: | ch, 4 |  |  |
| 0071 | A8 | 01 |  | test | al, 1 |  |  |
| 0073 | 75 | 08 | | jnz | Exit1 | | |
| 0075 | D0 | E8 | | shr | al, 1 | | |
| 0077 | FE | C1 | | inc | cl | | |
| 0079 | FE | CD | | dec | ch | | |
| 007B  007D  007D | 75  BA | F4  8002 | | jnz Exit1:  mov | LoopC  dx, OUTBIT | | |
| 0080 | B0 | 00 | | mov | al, 0 | | |
| 0082 | EE |  | | out | dx, al | | |
| 0083 | B5 | 00 | | mov | ch, 0 | | |

键值 = 列 X 4 + 行

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0085 | BB | 001Ar | mov | bx, offset KeyTable |
| 0088 | 03 | D9 | add | bx, cx |
| 008A | 8A | 07 | mov | al, [bx] ; 取出键码 |
| 008C | 8A | D8 | mov | bl, al |
| 008E |  |  | WaitRelease: | |
| 008E | BA | 8002 | mov dx, OUTBIT | |
| 0091 | B0 | 00 | mov al, 0 | |
| 0093 | EE |  | out dx, al ; 等键释放 | |
| 0094 | B4 | 0A | mov ah, 10 | |
| 0096 | E8 | FF67 | call Delay | |
| 0099 | E8 | FF9D | call TestKey | |
| 009C | 75 | F0 | jne WaitRelease | |
| 009E  00A0  00A1 | 8A C3 | C3 | mov al, bl ret  GetKey endp | |
| 00A1 |  |  | IInit proc near ; 8251 初始化 | |
| 00A1 | BA | D001 | mov dx, CS8251C | |
| 00A4  00A6 | B0 EE | 4F | mov al, 01001111b ; 1 停止位,无校验,8 数据位, x64 out dx, al | |
| 00A7 | B0 | 15 | mov al, 00010101b ; 清出错标志, 允许发送接收 | |
| 00A9  00AA  00AB | EE C3 |  | out dx, al ret  IInit endp | |
| 00AB |  |  | Send proc near ; 串口发送 | |
| 00AB | BA | D001 | mov dx, CS8251C | |
| 00AE  00B0  00B1  00B1  00B2 | B0 EE  EC A8 | 15  01 | mov al, 00010101b ; 清出错,允许发送接收  out dx, al WaitTXD:  in al, dx  test al, 1 ; 发送缓冲是否为空 | |
| 00B4 | 74 | FB | jz WaitTXD | |
| 00B6 | A0 | 0009r | mov al, TBuf ; 取要发送的字 | |
| 00B9 | BA | D000 | mov dx, CS8251D | |
| 00BC  00BD  00BE | EE 51 B9 | FFFF | out dx, al ; 发送  push cx  mov cx,0ffffh | |
| 00C1 | E2 | FE | loop $ | |
| 00C3  00C4  00C5 | 59  C3 |  | pop cx ret  Send endp | |
| 00C5  00C5  00C8  00C8 | BA EC | D001 | Receive proc near ; 串口接收  mov dx, CS8251C WaitRXD:  in al, dx | |
| 00C9 | A8 | 02 | test al, 2 ; 是否已收到一个字 | |
| 00CB | 74 | FB | je WaitRXD | |
| 00CD | BA | D000 | mov dx, CS8251D | |
| 00D0  00D1  00D4  00D5 | EC A2 C3 | 0008r | in al, dx ; 读入  mov RBuf, al ret  Receive endp | |
| 00D5  00D5 | B8 | 0000s | start proc near mov ax, data | |
| 00D8 | 8E | D8 | mov ds, ax | |
| 00DA | E8 | FFC4 | call IInit | |

|  |  |  |  |
| --- | --- | --- | --- |
| 00DD | C6 | 06 0000r FF | mov LEDBuf, 0ffh ; 显 示 8.8.8.8. |
| 00E2 | C6 | 06 0001r FF | mov LEDBuf+1, 0ffh |
| 00E7 | C6 | 06 0002r FF | mov LEDBuf+2, 0ffh |
| 00EC | C6 | 06 0003r FF | mov LEDBuf+3, 0ffh |
| 00F1 | C6 | 06 0004r 00 | mov LEDBuf+4, 0 |
| 00F6  00FB  00FB | C6  E8 | 06 0005r 00  FF0D | mov LEDBuf+5, 0 MLoop:  call DisplayLED ; 显示 |
| 00FE | BA | D001 | mov dx, CS8251C |
| 0101 | EC |  | in al, dx ; 是否接收到一个字 |
| 0102 | A8 | 02 | test al, 2 |
| 0104 | 75 | 12 | jnz RcvData |
| 0106 | E8 | FF30 | call TestKey ; 有键入? |
| 0109 | 74 | F0 | je MLoop ; 无键入, 继续显示 |
| 010B | E8 | FF3A | call GetKey ; 读入键码 |
| 010E | 24 | 0F | and al, 0fh ; 显示键码 |
| 0110 | A2 | 0009r | mov TBuf, al |
| 0113 | E8 | FF95 | call Send |
| 0116 | EB | E3 | jmp MLoop |
| 0118 |  |  | RcvData: |
| 0118 | E8 | FFAA | call Receive ; 读入接收到的字 |
| 011B | A0 | 0008r | mov al, RBuf |
| 011E | 24 | 0F | and al, 0fh ; 只显示低四位 |
| 0120 | B4 | 00 | mov ah, 0 |
| 0122 | BB | 000Ar | mov bx, offset LEDMAP |
| 0125 | 03 | D8 | add bx, ax |
| 0127 | 8A | 07 | mov al, [bx] ; 转换成显示码 |
| 0129 | A2 | 0005r | mov LEDBuf+5, al |
| 012C | EB | CD | jmp MLoop |
| 012E  012E |  |  | Start endp code ends |

end start

# 附录 F 8088 C 语言程序清单

// 软件实验一 存储器块清零 S1.C

/\* #define Buffer 0x3000 \*/ unsigned char Buffer[256];

void main()

{

unsigned int index; unsigned char \* ptr;

ptr = &Buffer; /\* 起始地址 \*/

for (index = 0; index <= 255; index++) {

\*ptr++ = 0xff; /\* 清 0, 地址加一 \*/

}

}

// 软件实验二 二进制到 BCD 码转换 S2.C

/\* 将 Number 拆为三个 BCD 码, 并存入 Result 数组 \*/

void main()

{

unsigned char Result[3]; unsigned char Number;

Number = 234;

Result[0] = Number / 100; /\* 除以 100, 得百位数 \*/ Result[1] = (Number 100) / 10; /\* 余数除以 10, 得十位数 \*/ Result[2] = Number 10; /\* 余数为个位数 \*/

}

// 软件实验三 二进制到 ASCII 码转换 S3.C

/\* 将 A 拆为二个 ASCII 码, 并存入 Result 数组 \*/

unsigned char const ASCIITable[16] = "0123456789ABCDEF";

/\* 定义数字对应的 ASCII 表 \*/ void main()

{

unsigned char Result[2]; unsigned char Number;

Number = 0x1a;

Result[0] = ASCIITable[Number / 16]; /\* 高四位 \*/ Result[1] = ASCIITable[Number & 0xf]; /\* 低四位 \*/

while(1);

}

// 软件实验四 内存块移动 S4.C

unsigned char Buffer1[256]; unsigned char Buffer2[256];

void main()

{

unsigned char \* ptr1; unsigned char \* ptr2; unsigned int index;

ptr1 = &Buffer1[0]; ptr2 = &Buffer2[0];

for (index=0; index <= 255; index++) {

\*ptr2++ = \*ptr1++ - 1;

}

}

// 软件实验五 程序跳转表 S5.C

void Func0() {}

void Func1() {}

void Func2() {}

void Func3() {}

void FuncEnter(unsigned char FuncID)

{

switch (FuncID) { case 0: Func0();

break;

case 1: Func1(); break;

case 2: Func2(); break;

case 3: Func3(); break;

default: break;

}

}

void main()

{

FuncEnter(0); FuncEnter(1); FuncEnter(2); FuncEnter(3);

}

// 软件实验六 数据排序 S6.C #define Size 10

unsigned char Array[Size]=

{5,2,1,0,2,3,8,6,5,9};

void main ()

{

unsigned char Change; unsigned char index; unsigned char Temp;

do {

Change = 0;

for (index = 0; index < (Size-1); index++) { if (Array[index] > Array[index+1]) {

Change = 1;

Temp = Array[index]; Array[index] = Array[index+1]; Array[index+1] = Temp;

}

}

} while (Change); while (1);

}

// 硬件实验三 用 74LS245 扩展 I/O 口实验 H3.C

extern unsigned char IN(unsigned int port);

extern void OUT(unsigned int port, unsigned char v);

#define CS245 0x8000 unsigned char b;

void main()

{

while (1) {

b = IN(CS245); b ++;

}

}

// 硬件实验四 用 74LS273 扩展 I/O 口实验 H4.C

extern unsigned char IN(unsigned int port);

extern void OUT(unsigned int port, unsigned char v); #define CS273 0x8000

void main()

{

unsigned char i, b;

while(1)

{

b = 1;

for (i=0; i<8; i++) { OUT(CS273, b);

b <<= 1;

}

}

}

// 硬件实验七 8255 输入、输出实验 H7.C

/\* 方式 0，PA，PC 输出，PB 输入 \*/ #define mode 0x82

/\* Port A \*/ #define PortA 0x8000

/\* Port B \*/ #define PortB 0x8001

/\* Port C \*/ #define PortC 0x8002

/\* 控制字地址 \*/ #define CAddr 0x8003

extern unsigned char IN(unsigned int port);

extern void OUT(unsigned int port, unsigned char v); unsigned char i,buf;

void Delay(unsigned int CNT)

{

while (CNT) CNT--;

}

void main()

{

while(1){

OUT(CAddr, mode); /\* 输出控制字 \*/

/\* 实验 1：PortA 输出 \*/ buf = 0x80;

for(i=0; i<8; i++){

OUT(PortA, buf); /\* 输出到 A 端口 \*/ buf >>= 1; /\* 移 位 \*/ Delay(0x5000); /\* 延时 \*/

};

/\* 实验 2： PortB 输入 PortA 输出 \*/ buf = IN(PortB); /\* 读入 PortB \*/

OUT(PortA, buf); /\* 输出到 PortA \*/ Delay(0x8000);

}

}

// 硬件实验十三 D/A 转换实验 H13.C #define CS0832 0x8000

extern unsigned char IN(unsigned int port);

extern void OUT(unsigned int port, unsigned char v);

unsigned char SinTbl[9] =

{0x00,0x18,0x30,0x46,0x59,0x69,0x75,0x7c,0x7f};

void Write0832(unsigned char b)

{

OUT(CS0832,b);

}

/\* unsigned char i; \*/

void main()

{

register i;

while(1){

for(i=0; i<16; i++) /\* 产Th锯齿波 \*/ Write0832(i\*0x10);

for(i=0; i<8; i++) /\* 产 Th 正 弦 波 \*/ Write0832(0x80 + SinTbl[i]); /\* (0 ~ π/2) 区间的波形 \*/

for(i=8; i>0; i--)

Write0832(0x80 + SinTbl[i]); /\* (π/2 ~ π) 区间的波形 \*/ for(i=0; i<8; i++)

Write0832(0x80 - SinTbl[i]); /\* (π ~ 3π/2) 区间的波形 \*/ for(i=8; i>0; i--)

Write0832(0x80 - SinTbl[i]); /\* (3π/2 ~ 2π) 区间的波形 \*/

}

}

// 硬件实验十四 A/D 转换实验 H14.C

#define mode 0x82 #define PA 0x9000 #define CTL 0x9003 #define CS0809 0x8000

extern unsigned char IN(unsigned int port);

extern void OUT(unsigned int port, unsigned char v);

unsigned char Read0809()

{

unsigned char i;

OUT(CS0809, 0); /\* 起 动 A/D \*/ for (i=0; i<0x80; i++) ; /\* 延时 > 100us \*/ return(IN(CS0809)); /\* 读入结果 \*/

}

unsigned char b; void main(void)

{

OUT(CTL, mode);

while(1){

b = Read0809(); OUT(PA, b);

}

}

//硬件实验十六 八段数码管显示实验 H16.C

#define LEDLen 6 #define OUTBIT 0x8002 #define OUTSEG 0x8004

extern unsigned char IN(unsigned int port);

extern void OUT(unsigned int port, unsigned char v);

unsigned char LEDBuf[LEDLen]; /\* 显示缓冲 \*/ unsigned char const LEDMAP[] = { /\* 八段管显示码 \*/

0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71

};

void Delay(unsigned char CNT)

{

unsigned char i;

while (CNT-- !=0)

for (i=100; i !=0; i--);

}

void DisplayLED()

{

unsigned char i, j, k; unsigned char Pos; unsigned char LED;

Pos = 0x20; /\* 从左边开始显示 \*/ for (i = 0; i < LEDLen; i++) {

OUT(OUTBIT,0); /\* 关所有八段管 \*/ LED = LEDBuf[i];

OUT(OUTSEG,LED); /\* 输出 LED 段码 \*/

OUT(OUTBIT, Pos); /\* 显示一位八段管 \*/ Delay(1);

Pos >>= 1; /\* 显示下一位 \*/

}

}

void main()

{

unsigned char i = 0; unsigned char j;

while(1) {

LEDBuf[0] = LEDMAP[ i & 0x0f]; LEDBuf[1] = LEDMAP[(i+1) & 0x0f]; LEDBuf[2] = LEDMAP[(i+2) & 0x0f]; LEDBuf[3] = LEDMAP[(i+3) & 0x0f]; LEDBuf[4] = LEDMAP[(i+4) & 0x0f]; LEDBuf[5] = LEDMAP[(i+5) & 0x0f];

i++;

for(j=0; j<20; j++) DisplayLED(); /\* 延时 \*/

}

}

//硬件实验十七 键盘扫描显示实验 H17.C

#define LEDLen 6 #define IN\_KEY 0x9001 #define OUTBIT 0x9002 #define OUTSEG 0x9004

extern unsigned char IN(unsigned int port);

extern void OUT(unsigned int port, unsigned char v);

unsigned char LEDBuf[LEDLen]; /\* 显 示 缓 冲 \*/ unsigned char const LEDMAP[] = { /\* 八段管显示码 \*/

0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71

};

unsigned char const KeyTable[] = { /\* 键码定义 \*/ 0x16, 0x15, 0x14, 0xff,

0x13, 0x12, 0x11, 0x10,

0x0d, 0x0c, 0x0b, 0x0a, 0x0e, 0x03, 0x06, 0x09,

0x0f, 0x02, 0x05, 0x08,

0x00, 0x01, 0x04, 0x07

};

void Delay(unsigned char CNT)

{

unsigned char i;

while (CNT-- !=0)

for (i=100; i !=0; i--);

}

void DisplayLED()

{

unsigned char i, j, k; unsigned char Pos; unsigned char LED;

Pos = 0x20; /\* 从左边开始显示 \*/ for (i = 0; i < LEDLen; i++) {

OUT(OUTBIT,0); /\* 关所有八段管 \*/ LED = LEDBuf[i];

OUT(OUTSEG,LED); /\* 输出 LED 段码 \*/

OUT(OUTBIT, Pos); /\* 显示一位八段管 \*/ Delay(1);

Pos >>= 1; /\* 显示下一位 \*/

}

}

unsigned char TestKey()

{

OUT(OUTBIT, 0); /\* 输出线置为 0 \*/

return (~IN(IN\_KEY) & 0x0f); /\* 读入键状态 (高四位不用) \*/

}

unsigned char GetKey()

{

unsigned char Pos; unsigned char i; unsigned char j; unsigned char k;

i = 6;

Pos = 0x20; /\* 找出键所在列 \*/ do {

OUT(OUTBIT, ~ Pos);

Pos >>= 1;

k = ~IN(IN\_KEY) & 0x0f;

} while ((--i != 0) && (k == 0));

/\* 键值 = 列 X 4 + 行 \*/ if (k != 0) {

i \*= 4;

if (k & 2) i += 1;

else if (k & 4)

i += 2;

else if (k & 8) i += 3;

OUT(OUTBIT, 0);

do Delay(10); while (TestKey()); /\* 等键释放 \*/

return(KeyTable[i]); /\* 取出键码 \*/

} else return(0xff);

}

void main()

{

LEDBuf[0] = 0xff; LEDBuf[1] = 0xff; LEDBuf[2] = 0xff; LEDBuf[3] = 0xff; LEDBuf[4] = 0x00; LEDBuf[5] = 0x00;

while (1) { DisplayLED();

if (TestKey()) LEDBuf[5] = LEDMAP[GetKey() & 0x0f];

}

}

//硬件实验十八 电子时钟 H18.C

/\*

\*; 本实验利用 8253 做定时器，用定时器输出的脉冲控制 8259 产Th中断

\*; 在 8259 中断处理程序中，对时、分、秒进行计数，在等待中断的循

\*; 环中用 LED 显示时间。

\*; 8253 用定时器/计数器 1，8253 片选接 CS4，地址为 0C000H。8253 时钟

\*; 源 CLK1 接分频电路的 F/64 输出。分频器的 Fin 接 4MHz 时钟。8253 的

\*; GATE1 接 VCC。

\*; 8259 中断 INT0 接 8253 的 OUT1，片选接 CS5，地址为 0D000H。

\*; 显示电路的 KEY/LED CS 接 CS0，地址为 08000H。

\*/

#define ICW1 0x13 /\* 单片 8259, 上升沿中断, 要写 ICW4 \*/ #define ICW2 0x20 /\* 中断号为 20H \*/

#define ICW4 0x01 /\* 工作在 8086/88 方式 \*/ #define OCW1 0xfe /\* 只响应 INT0 中断 \*/ #define LEDLen 6

#define CS8259A 0xd000 #define CS8259B 0xd001

#define CONTROL 0xc003 #define COUNT0 0xc000 #define COUNT1 0xc001 #define COUNT2 0xc002

#define OUTBIT 0x8002 #define OUTSEG 0x8004

unsigned char LEDBuf[LEDLen]; /\* 显示缓冲 \*/ unsigned char const LEDMAP[] = { /\* 八段管显示码 \*/

0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71

};

extern unsigned char IN(unsigned int port);

extern void OUT(unsigned int port, unsigned char v); extern void DISABLE(void);

extern void ENABLE(void);

extern void SETINT(unsigned char NO, unsigned int \*ENTER);

unsigned char CNT;

unsigned char Hour, Minute, Second;

void Delay(unsigned char CNT)

{

unsigned char i;

while (CNT-- !=0)

for (i=100; i !=0; i--);

}

void DisplayLED()

{

unsigned char i, j, k; unsigned char Pos; unsigned char LED;

Pos = 0x20; /\* 从左边开始显示 \*/ for (i = 0; i < LEDLen; i++) {

OUT(OUTBIT,0); /\* 关所有八段管 \*/ LED = LEDBuf[i];

OUT(OUTSEG,LED); /\* 输出 LED 段码 \*/

OUT(OUTBIT, Pos); /\* 显示一位八段管 \*/ Delay(1);

Pos >>= 1; /\* 显示下一位 \*/

}

}

interrupt IEnter()

{

Second++;

if (Second == 60) { Second = 0; Minute++;

if (Minute == 60) { Minute = 0; Hour++;

if (Hour == 24) Hour = 0;

}

};

OUT(CS8259A, 0x20); /\* 中断服务程序结束指令 \*/

}

void IInit()

{

OUT(CS8259A, ICW1); OUT(CS8259B, ICW2); OUT(CS8259B, ICW4); OUT(CS8259B, OCW1);

}

void main()

{

OUT(CONTROL, 0x76); /\* 计数器 1,16 位,方式 3,二进制 \*/ OUT(COUNT1, 62500 256); /\* 计数器低字节 \*/

OUT(COUNT1, 62500 / 256); /\* 计数器高字节 \*/

Hour = 0;

Minute = 0;

Second = 0;

DISABLE(); /\* 关闭中断响应 \*/

IInit();

SETINT(ICW2, &IEnter); /\* 中断入口地址 \*/ CNT = 0;

ENABLE(); /\* 打开中断响应 \*/ while (1) {

LEDBuf[0] = LEDMAP[Hour/10];

LEDBuf[1] = LEDMAP[Hour 10] | 0x80; LEDBuf[2] = LEDMAP[Minute/10]; LEDBuf[3] = LEDMAP[Minute 10] | 0x80; LEDBuf[4] = LEDMAP[Second/10]; LEDBuf[5] = LEDMAP[Second 10];

DisplayLED();

}

}

//硬件实验二十 打印机控制试验 H20.C

/\* 状态端口地址 \*/ #define status 0x08004

/\* 控制端口地址 \*/ #define ctl 0x08002

/\* 数据端口地址 \*/ #define dport 0x08001

/\* 忙状态位 \*/ #define busy 4

extern unsigned char IN(unsigned int port);

extern void OUT(unsigned int port, unsigned char v);

unsigned char string[] = {'H','e','l','l','o',' ',

'W','o','r','l','d','!',

'\n',0x0c,0x00};

void print(unsigned char dd)

{

while(IN(status) & busy); /\* 读打印机状态,若忙,等待 \*/ OUT(dport,dd); /\* 数据送出 \*/

OUT(ctl,1); /\* 输出选通脉冲 \*/ OUT(ctl,0);

}

void main()

{

char \*pos;

pos = &string; /\* 被打印字串 \*/ while((\*pos)) print( \*pos++);

while(1);

}

//硬件实验二十一 直流电机控制实验 H21.C

/\* DC motor \*/

/\* C for 8086/8088 \*/ #define mode 0x82 #define CONTRL 0x8003 #define status 0x8001

#define CS0832 0x9000

extern unsigned char IN(unsigned int port);

extern void OUT(unsigned int port, unsigned char v); unsigned int count;

#define DC\_P 1

void delay()

{

unsigned int ddd;

ddd = 50000; /\* 在 6MHz 约延时 1 秒 \*/ while(ddd--);

}

unsigned int read()

{

while(!(IN(status) & DC\_P)); /\* 等待低电平完 \*/ while(IN(status) & DC\_P); /\* 等待高电平完 \*/

count = 0;

while(!(IN(status) & DC\_P)) count ++; /\* 等待低电平完 \*/ while(IN(status) & DC\_P) count++; /\* 等待高电平完 \*/

return (count);

}

void main()

{

|  |  |  |
| --- | --- | --- |
| unsigned int tt;  OUT(CONTRL,mode); |  | |
| OUT(CS0832,0xff); | /\* | 产Th电压控制电机 \*/ |
| delay(); | /\* | 等待电机运转稳定 \*/ |
| tt = read(); | /\* | 读取时间 \*/ |
| OUT(CS0832,0xc0); | /\* | 产Th电压控制电机 \*/ |
| delay(); | /\* | 等待电机运转稳定 \*/ |
| tt = read(); | /\* | 读取时间 \*/ |
| OUT(CS0832,0x40); | /\* | 产Th电压控制电机 \*/ |
| delay(); | /\* | 等待电机运转稳定 \*/ |
| tt = read(); | /\* | 读取时间 \*/ |
| OUT(CS0832,0x00); | /\* | 产Th电压控制电机 \*/ |
| delay(); | /\* | 等待电机运转稳定 \*/ |
| tt = read(); | /\* | 读取时间 \*/ |
| while(1); |  |  |

}

//硬件实验二十二 步进电机控制实验 H22.C #define mode8255 0x82

/\* 8255 工作方式，PA、PC 输出，PB 输入 \*/ #define contrl 0x8003

/\* 步进电机控制脉冲从 8255 的 PA 端口输出 \*/ #define ctl 0x8000

/\* ABCD 各脉冲对应的输出位 \*/ #define Astep 0x01

#define Bstep 0x02 #define Cstep 0x04

#define Dstep 0x08

extern unsigned char IN(unsigned int port);

extern void OUT(unsigned int port, unsigned char v); unsigned char dly\_c;

void delay()

{

unsigned char tt,cc;

cc = dly\_c; do{

tt = 0x40; do {

}while(--tt);

}while(--cc);

}

void main()

{

unsigned char mode;

OUT(contrl,mode8255); mode = 2;

OUT(ctl,0);

dly\_c = 0x10;

/\* 单/双八拍工作方式 \*/ if(mode ==1)

while(1)

{

OUT(ctl,Astep); delay(); OUT(ctl,Astep+Bstep); delay(); OUT(ctl,Bstep); delay(); OUT(ctl,Bstep+Cstep); delay(); OUT(ctl,Cstep); delay(); OUT(ctl,Cstep+Dstep); delay(); OUT(ctl,Dstep); delay(); OUT(ctl,Dstep+Astep); delay();

if(dly\_c>2) dly\_c --;

};

/\* 双四拍工作方式 \*/ if(mode == 2) while(1)

{

OUT(ctl,Astep+Bstep); delay(); OUT(ctl,Bstep+Cstep); delay(); OUT(ctl,Cstep+Dstep); delay(); OUT(ctl,Dstep+Astep); delay();

if(dly\_c>3) dly\_c --;

};

/\* 单四拍工作方式 \*/ if(mode ==3) while(1)

{

OUT(ctl,Dstep); delay(); OUT(ctl,Cstep); delay(); OUT(ctl,Bstep); delay(); OUT(ctl,Astep); delay();

if(dly\_c>4) dly\_c --;

}

while(1);

}

//硬件实验二十三 温度传感器试验 H23.C #define LEDLen 6

#define ADPort 0x9000 /\* AD 采样输入片选 \*/ #define IN\_KEY 0x8001

#define OUTBIT 0x8002 #define OUTSEG 0x8004

extern unsigned char IN(unsigned int port);

extern void OUT(unsigned int port, unsigned char v);

unsigned char LEDBuf[LEDLen]; /\* 显 示 缓 冲 \*/ unsigned char const LEDMAP[] = { /\* 八段管显示码 \*/

0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71

};

void Delay(unsigned char CNT)

{

unsigned char i;

while (CNT-- !=0)

for (i=80; i !=0; i--);

}

void DisplayLED()

{

unsigned char i, j, k; unsigned char Pos; unsigned char LED;

Pos = 0x20; /\* 从左边开始显示 \*/ for (i = 0; i < LEDLen; i++) {

OUT(OUTBIT,0); /\* 关所有八段管 \*/ LED = LEDBuf[i];

OUT(OUTSEG,LED);

OUT(OUTBIT, Pos); /\* 显示一位八段管 \*/ Delay(1);

Pos >>= 1; /\* 显示下一位 \*/

}

OUT(OUTBIT,0); /\* 关所有八段管 \*/

}

/\* ==================================\*/

#define LowTemp (-99) /\* A/D 0 \*/ #define HighTemp 99 /\* A/D 255\*/

signed char CurTemp; void DisplayResult()

{

signed char T;

T = CurTemp; if (T < 0) {

LEDBuf[0] = 0x40; T = - T;

} else LEDBuf[0] = 0; LEDBuf[1] = LEDMAP[T / 10]; LEDBuf[2] = LEDMAP[T 10];

LEDBuf[3] = 0;

}

unsigned char ReadAD()

{

unsigned int i;

OUT(ADPort, 0); /\* start A/D convert \*/ for (i=0; i<10; i++) ; /\* delay 100us \*/ return (0xff - IN(ADPort));

}

void ReadTemp()

{

unsigned char i; signed int Temp;

Temp = 0;

for (i=0; i<16; i++) { Temp += ReadAD();

}

CurTemp = (Temp/16) \* (HighTemp-LowTemp) / 256 + LowTemp;

}

void main()

{

unsigned char dtimer;

while (1) { for(dtimer=10;dtimer>0;dtimer--){

DisplayResult(); DisplayLED();

};

ReadTemp();

}

}

//硬件实验二十四 液晶显示控制电路 H24.C

#define PD1 61 /\* 122/2 分成左右两半屏(122x32) \*/ unsigned char Column;

unsigned char Page\_ ; /\* 页地址寄存器 D1,DO:页地址 \*/ unsigned char Code\_ ; /\* 字符代码寄存器 \*/

unsigned char Command; /\* 指令寄存器 \*/ unsigned char LCDData; /\* 数据寄存器 \*/

#define CWADD1 0x8000 /\* 写指令代码地址(E1) \*/ #define DWADD1 0x8001 /\* 写显示数据地址(E1) \*/ #define CRADD1 0x8002 /\* 读状态字地址(E1) \*/ #define DRADD1 0x8003 /\* 读显示数据地址(E1) \*/

#define CWADD2 0x8004 /\* 写指令代码地址(E2) \*/ #define DWADD2 0x8005 /\* 写显示数进地址(E2) \*/ #define CRADD2 0x8006 /\* 读状态字地址(E2) \*/ #define DRADD2 0x8007 /\* 读显示数据地址(E2) \*/

extern unsigned char IN(unsigned int port);

extern void OUT(unsigned int port, unsigned char v);

/\* 英文字模库 8x8 点 阵 \*/ unsigned char const EETAB[][8]={

{0x38,0x44,0x44,0x44,0x44,0x44,0x38,0x00}, /\* 0 \*/

{0x10,0x30,0x50,0x10,0x10,0x10,0x7c,0x00}, /\* 1 \*/

{0x38,0x44,0x04,0x08,0x10,0x20,0x7c,0x00}, /\* 2 \*/

{0x78,0x84,0x04,0x38,0x04,0x84,0x78,0x00}}; /\* 3 \*/

/\* 中文字模库 16x16 点 阵 \*/ unsigned char const CCTAB[][32] ={

{0x00,0x04,0xe4,0x24,0x24,0x64,0xb4,0x2f, /\* 南 \*/ 0x24,0xa4,0x64,0x24,0x24,0xe6,0x04,0x00,

0x00,0x00,0x7f,0x04,0x05,0x05,0x05,0x7f,

0x05,0x05,0x05,0x25,0x44,0x3f,0x00,0x00},

{0x00,0x04,0x04,0xe4,0x24,0x24,0x25,0x26, /\* 京 \*/ 0x24,0x24,0x24,0xe4,0x06,0x04,0x00,0x00,

0x00,0x20,0x10,0x19,0x0D,0x41,0x81,0x7f,

0x01,0x01,0x05,0x0d,0x38,0x10,0x00,0x00},

{0x00,0x80,0x60,0xf8,0x07,0x0a,0x28,0x28, /\* 伟 \*/ 0x28,0xff,0x28,0x28,0x28,0x28,0x00,0x00,

0x01,0x00,0x00,0xfF,0x00,0x01,0x01,0x01,

0x01,0xff,0x01,0x21,0x41,0x3f,0x00,0x00},

{0x08,0x08,0x89,0xce,0x28,0x18,0x02,0x7a, /\* 福 \*/ 0x4a,0x4a,0x4a,0x4a,0x4a,0x7a,0x02,0x00, 0x02,0x01,0x00,0xff,0x01,0x02,0xff,0x49, 0x49,0x49,0x7f,0x49,0x49,0x49,0xff,0x00}};

/\* 1.写指令代码子程序(E1) \*/ void WriteCommandE1()

{

while(IN(CRADD1) & 0x80); OUT(CWADD1, Command);

}

/\* 2.写显示数据子程序(E1) \*/ void WriteDataE1()

{

while(IN(CRADD1) & 0x80); OUT(DWADD1, LCDData);

}

/\* 3.读显示数据子程序(E1) \*/ void ReadDataE1()

{

while(IN(CRADD1) & 0x80); LCDData = IN(DRADD1);

}

/\* 4.写指令代码子程序(E2) \*/ void WriteCommandE2()

{

while(IN(CRADD2) & 0x80); OUT(CWADD2, Command);

}

/\* 5.写显示数据子程序(E2) \*/ void WriteDataE2()

{

while(IN(CRADD2) & 0x80); OUT(DWADD2, LCDData);

}

/\* 6.读显示数据子程序(E2) \*/ void ReadDataE2()

{

while(IN(CRADD2) & 0x80); LCDData = IN(DRADD2);

}

/\* 初始化程序 \*/ void Init()

{

Command = 0xe2; WriteCommandE1(); WriteCommandE2();

Command = 0xa4; WriteCommandE1(); WriteCommandE2();

Command = 0xa9; WriteCommandE1(); WriteCommandE2();

Command = 0xa0; WriteCommandE1(); WriteCommandE2();

Command = 0xc0; WriteCommandE1(); WriteCommandE2();

Command = 0xaf;

WriteCommandE1();

WriteCommandE2();

}

/\* 清 屏 \*/ void Clear()

{

unsigned char i,j;

i = 0;

do {

Command = (i + 0xb8);/\* 页地址设置 \*/ WriteCommandE1();

WriteCommandE2();

Command = 0x00; /\* 列地址设置为"0" \*/ WriteCommandE1();

WriteCommandE2();

j = 0x50; /\* 一页清 80 个字节 \*/ do {

LCDData = 0x00; /\* 显示数据为"0" \*/ WriteDataE1();

WriteDataE2();

}while(--j !=0); /\* 页内字节清零循环 \*/

}while(++i !=4); /\* 页地址暂存器加一 \*/

/\* 显示区清零循环 \*/

}

/\* 延时程序 \*/ void Delay()

{

unsigned char i,j; i = 20;

j = 0;

do{

do{

}while(--j !=0);

}while( --i != 0 );

}

/\* 中文显示子程序 \*/ void WriteCHN16x16()

{

unsigned char i,j,k;

i = 0;

j = 0;

while(j<2) {

Command = ((Page\_ + j) & 0x03) | 0xb8; /\* 设置页地址 \*/ WriteCommandE1();

WriteCommandE2();

k = Column; /\* 列地址值 \*/ while(k < Column + 16){

if (k < PD1) { /\* 为左半屏显示区域(E1) \*/ Command = k;

WriteCommandE1(); /\* 设置列地址值 \*/ LCDData = CCTAB[Code\_][i]; /\* 取汉字字模数据 \*/ WriteDataE1(); /\* 写字模数据 \*/

} else{ /\* 为右半屏显示区域(E2) \*/ Command = k-PD1;

WriteCommandE2(); /\* 设置列地址值 \*/ LCDData = CCTAB[Code\_][i]; /\* 取汉字字模数据 \*/ WriteDataE2(); /\* 写字模数据 \*/

};

i++;

if( ++k >= PD1 \* 2) break; /\* 列地址是否超出显示范围 \*/

} ; j++;

};

}

/\* 中文显示程序 \*/ void main()

{

Init();

Clear();

Page\_ = 0x02; Column = 0x00; Code\_ = 0x00; WriteCHN16x16();

Page\_ = 0x02; Column = 0x10; Code\_ = 0x01; WriteCHN16x16();

Page\_ = 0x02; Column = 0x20; Code\_ = 0x02; WriteCHN16x16();

Page\_ = 0x02; Column = 0x30; Code\_ = 0x03; WriteCHN16x16();

while(1);

}

// 硬件实验二十五 电子琴 H25.C

/\*

* 本实验用 8253 做定时器输出音频信号，控制喇叭发出声音
* 将 8253 的 CLK1 接到 1MHz 脉冲信号上（1MHz 信号可由 4MHz 信号
* 分频后得到），GATE1 接 VCC，OUT1 接喇叭的脉冲输入。

\*/

#define CONTROL 0xc003 #define COUNT0 0xc000 #define COUNT1 0xc001 #define COUNT2 0xc002

#define IN\_KEY 0x8001 #define OUTBIT 0x8002

extern unsigned char IN(unsigned int port);

extern void OUT(unsigned int port, unsigned char v);

unsigned int const ToneTable[] = {

2273, 2024, 1805, 1704, 1517, 1353, 1205, 1136};

unsigned char const KeyTable[] = { /\* 键码定义 \*/ 0x16, 0x15, 0x14, 0xff,

0x13, 0x12, 0x11, 0x10,

0x0d, 0x0c, 0x0b, 0x0a, 0x0e, 0x03, 0x06, 0x09,

0x0f, 0x02, 0x05, 0x08,

0x00, 0x01, 0x04, 0x07};

unsigned char TestKey()

{

OUT(OUTBIT, 0); /\* 输出线置为 0 \*/

return (~IN(IN\_KEY) & 0x0f); /\* 读入键状态 (高四位不用) \*/

}

unsigned char GetKey()

{

unsigned char Pos; unsigned char i; unsigned char j; unsigned char k;

i = 6;

Pos = 0x20; /\* 找出键所在列 \*/ do {

OUT(OUTBIT, ~ Pos);

Pos >>= 1;

k = ~IN(IN\_KEY) & 0x0f;

} while ((--i != 0) && (k == 0));

/\* 键值 = 列 X 4 + 行 \*/ if (k != 0) {

i \*= 4;

if (k & 2) i += 1;

else if (k & 4)

i += 2;

else if (k & 8) i += 3;

return(KeyTable[i]); /\* 取出键码 \*/

} else return(0xff);

}

void Delay(unsigned char CNT)

{

unsigned char i;

while (CNT-- !=0)

for (i=100; i !=0; i--);

}

void OutTone(unsigned int Tone)

{

OUT(CONTROL, 0x76); /\* 定时器 1,16 位,模式 3, 二进制 \*/ OUT(COUNT1, Tone 256); /\* 计数器低 8 位 \*/

OUT(COUNT1, Tone / 256); /\* 计数器高 8 位 \*/

}

void main()

{

unsigned char Key; unsigned int Tone;

while (1) {

while (! TestKey()) ; /\* 等按键 \*/ Key = GetKey();

if ( (Key >= 1) && (Key <= 8) ) { Tone = ToneTable[Key-1] ; OutTone(Tone);

delay(600); OutTone(2);

}

}

}

//硬件实验二十六 空调温度控制实验 H26.C #define LEDLen 6

#define ADPort 0x8000 /\* AD 采样输入片选 \*/ #define CS273 0xa000 /\* 控制输出片选 \*/

#define IN\_KEY 0xfe00 #define OUTBIT 0xfe02 #define CLK164 0xfe04 #define DAT164 0xfe06

extern unsigned char IN(unsigned int port);

extern void OUT(unsigned int port, unsigned char v);

unsigned char LEDBuf[LEDLen]; /\* 显 示 缓 冲 \*/ unsigned char const LEDMAP[] = { /\* 八段管显示码 \*/

0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71

};

unsigned char const KeyTable[] = { /\* 键码定义 \*/ 0x16, 0x15, 0x14, 0xff,

0x13, 0x12, 0x11, 0x10,

0x0d, 0x0c, 0x0b, 0x0a, 0x0e, 0x03, 0x06, 0x09,

0x0f, 0x02, 0x05, 0x08,

0x00, 0x01, 0x04, 0x07

};

void Delay(unsigned char CNT)

{

unsigned char i; while (CNT-- !=0)

for (i=80; i !=0; i--);

}

void DisplayLED()

{

unsigned char i, j, k; unsigned char Pos; unsigned char LED;

Pos = 0x20; /\* 从左边开始显示 \*/ for (i = 0; i < LEDLen; i++) {

OUT(OUTBIT,0); /\* 关所有八段管 \*/ LED = LEDBuf[i];

for (j = 0; j < 8; j++) { /\* 送 164 \*/ if (LED & 0x80) OUT(DAT164, 1);

else OUT(DAT164, 0);

OUT(CLK164, 1);

OUT(CLK164, 0);

LED <<= 1;

}

OUT(OUTBIT, Pos); /\* 显示一位八段管 \*/ Delay(1);

Pos >>= 1; /\* 显示下一位 \*/

}

OUT(OUTBIT,0); /\* 关所有八段管 \*/

}

unsigned char TestKey()

{

OUT(OUTBIT, 0); /\* 输出线置为 0 \*/

return (~IN(IN\_KEY) & 0x0f); /\* 读入键状态 (高四位不用) \*/

}

unsigned char GetKey()

{

unsigned char Pos; unsigned char i; unsigned char j; unsigned char k;

i = 6;

Pos = 0x20; /\* 找出键所在列 \*/ do {

OUT(OUTBIT, ~ Pos);

Pos >>= 1;

k = ~IN(IN\_KEY) & 0x0f;

} while ((--i != 0) && (k == 0));

/\* 键值 = 列 X 4 + 行 \*/ if (k != 0) {

i \*= 4;

if (k & 2) i += 1;

else if (k & 4) i += 2;

else if (k & 8) i += 3;

OUT(OUTBIT, 0);

do Delay(10); while (TestKey()); /\* 等键释放 \*/

return(KeyTable[i]); /\* 取出键码 \*/

} else return(0xff);

}

/\* ==================================\*/

#define UP 0x16 #define DOWN 0x15

#define ON 1

#define OFF 0

#define LowLimit 10

#define HighLimit 30

#define LowTemp (-40) /\* A/D 0 \*/ #define HighTemp 50 /\* A/D 255\*/

#define Heat 1

#define Cool 2 signed char P1Buf;

signed char SetTemp, CurTemp;

void DisplayResult()

{

signed char T;

T = CurTemp; if (T < 0) {

LEDBuf[0] = 0x40; T = - T;

} else LEDBuf[0] = 0; LEDBuf[1] = LEDMAP[T / 10]; LEDBuf[2] = LEDMAP[T 10];

LEDBuf[3] = 0;

LEDBuf[4] = LEDMAP[SetTemp / 10]; LEDBuf[5] = LEDMAP[SetTemp 10];

}

unsigned char ReadAD()

{

unsigned int i;

OUT(ADPort, 0); /\* start A/D convert \*/ for (i=0; i<10; i++) ; /\* delay 100us \*/ return (IN(ADPort));

}

void ReadTemp()

{

unsigned char i; signed int Temp;

Temp = 0;

for (i=0; i<16; i++) { Temp += ReadAD();

}

CurTemp = (Temp/16) \* (HighTemp-LowTemp) / 256 + LowTemp;

}

void main()

{

unsigned char Key;

P1Buf = 0;

SetTemp = 20;

while (1) {

while (!TestKey()) { DisplayResult(); DisplayLED(); ReadTemp();

if (CurTemp < (SetTemp - 2)) { P1Buf |= Heat;

P1Buf &= ~Cool;

} else if (CurTemp > (SetTemp + 2)) { P1Buf &= ~Heat;

P1Buf |= Cool;

} else {

P1Buf &= ~Heat;

P1Buf &= ~Cool;

}

OUT(CS273, P1Buf);

}

Key = GetKey();

if (Key == DOWN) {

if (SetTemp > LowLimit) SetTemp--;

} else if (Key == UP) {

if (SetTemp < HighLimit) SetTemp++;

}

}

}

// 硬件实验二十七 计算器实验 H27.C

#define LEDLen 6 #define IN\_KEY 0xfe00 #define OUTBIT 0xfe02 #define CLK164 0xfe04 #define DAT164 0xfe06

extern unsigned char IN(unsigned int port);

extern void OUT(unsigned int port, unsigned char v);

unsigned char LEDBuf[LEDLen]; /\* 显 示 缓 冲 \*/ unsigned char const LEDMAP[] = { /\* 八段管显示码 \*/

0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71

};

unsigned char const KeyTable[] = { /\* 键码定义 \*/ 0x16, 0x15, 0x14, 0xff,

0x13, 0x12, 0x11, 0x10,

0x0d, 0x0c, 0x0b, 0x0a, 0x0e, 0x03, 0x06, 0x09,

0x0f, 0x02, 0x05, 0x08,

0x00, 0x01, 0x04, 0x07

};

void Delay(unsigned char CNT)

{

unsigned char i;

while (CNT-- !=0)

for (i=100; i !=0; i--);

}

void DisplayLED()

{

unsigned char i, j, k; unsigned char Pos; unsigned char LED;

Pos = 0x20; /\* 从左边开始显示 \*/ for (i = 0; i < LEDLen; i++) {

OUT(OUTBIT,0); /\* 关所有八段管 \*/ LED = LEDBuf[i];

for (j = 0; j < 8; j++) { /\* 送 164 \*/ if (LED & 0x80) OUT(DAT164, 1);

else OUT(DAT164, 0);

OUT(CLK164, 1);

OUT(CLK164, 0);

LED <<= 1;

}

OUT(OUTBIT, Pos); /\* 显示一位八段管 \*/ Delay(1);

Pos >>= 1; /\* 显示下一位 \*/

}

}

unsigned char TestKey()

{

OUT(OUTBIT, 0); /\* 输出线置为 0 \*/

return (~IN(IN\_KEY) & 0x0f); /\* 读入键状态 (高四位不用) \*/

}

unsigned char GetKey()

{

unsigned char Pos; unsigned char i; unsigned char j; unsigned char k;

i = 6;

Pos = 0x20; /\* 找出键所在列 \*/ do {

OUT(OUTBIT, ~ Pos);

Pos >>= 1;

k = ~IN(IN\_KEY) & 0x0f;

} while ((--i != 0) && (k == 0));

/\* 键值 = 列 X 4 + 行 \*/ if (k != 0) {

i \*= 4;

if (k & 2)

i += 1;

else if (k & 4) i += 2;

else if (k & 8) i += 3;

OUT(OUTBIT, 0);

do Delay(10); while (TestKey()); /\* 等键释放 \*/

return(KeyTable[i]); /\* 取出键码 \*/

} else return(0xff);

}

/\* ====================================\*/

#define ADD 0x0a #define SUB 0x0b #define MUL 0x0c #define DIV 0x0d #define EQU 0x0e #define CLR 0x0f

void DisplayResult(signed int Result)

{

unsigned char i;

if (Result >= 0) { LEDBuf[0] = 0;

} else {

LEDBuf[0] = 0x40;

Result = - Result;

}

LEDBuf[1] = 0;

LEDBuf[2] = 0;

LEDBuf[3] = 0;

LEDBuf[4] = 0;

LEDBuf[5] = LEDMAP[Result 10]; Result = Result / 10; i = 4;

while (Result != 0) {

LEDBuf[i--] = LEDMAP[Result 10]; Result = Result / 10;

}

}

void main()

{

signed int Last, Result; unsigned char OP; unsigned char Key;

Last = 0;

Result = 0; OP = ADD;

DisplayResult(Result); while (1) {

while (!TestKey()) DisplayLED(); Key = GetKey();

if ( (Key >= 0) && (Key <= 9) ) { Result = Result\*10 + Key; DisplayResult(Result);

} else if (Key == CLR) { Last = 0;

Result = 0; OP = ADD;

DisplayResult(Result);

} else if ( (Key==ADD) || (Key==SUB) || (Key==MUL) || (Key==DIV) || (Key==EQU) ) { if (OP == ADD)

Result = Last + Result; else if (OP == SUB)

Result = Last - Result; else if (OP == MUL)

Result = Last \* Result; else if (OP == DIV)

Result = Last / Result;

if (Key == EQU) { Last = 0;

OP = ADD;

} else {

Last = Result;

OP = Key;

}

DisplayResult(Result);

Result = 0;

}

}

}

// 硬件实验三十三 PWM 转换电压实验 H33.C

/\* 输出 50 (5:5) 占空比 PWM \*/

/\* 输出 10 (1:9) 占空比 PWM \*/

/\* 输出 90 (9:1) 占空比 PWM \*/

extern unsigned char IN(unsigned int port);

extern void OUT(unsigned int port, unsigned char v);

#define mode 0x82 #define PA 0x9000 #define CTL 0x9003

void Delay(unsigned char CNT)

{

unsigned char i; while (CNT--) {

for (i=0; i<30; i++) ;

}

}

void main()

{

OUT(CTL,mode);

while (1) {

OUT(PA,1);

Delay(9);

OUT(PA,0);

Delay(1);

}

}

// 硬件实验三十四 8253 计数器实验 H34.C

#define CONTROL 0xc003 #define COUNT0 0xc000 #define COUNT1 0xc001 #define COUNT2 0xc002

extern unsigned char IN(unsigned int port);

extern void OUT(unsigned int port, unsigned char v);

unsigned int i,j; void main()

{

OUT(CONTROL, 0x30); /\* counter0,16bit,mode 3,binary \*/ OUT(COUNT0, 0); /\* low byte of counter \*/

OUT(COUNT0, 0); /\* high byte of counter \*/ while(1){

OUT(CONTROL, 0x0); /\* counter1,16bit,mode 3,binary \*/ i = IN(COUNT0); /\* low byte of counter \*/

i += IN(COUNT0)\*256; /\* high byte of counter \*/ j = i ;

}

}

// 硬件实验三十五 8259 外部中断实验 H35.C

#define PA8255 0x8000 /\* 8255 PA 输出地址 \*/ #define CTL8255 0x8003

#define mode 0x82

#define ICW1 0x13

/\* 单片 8259, 上升沿中断, 要写 ICW4 \*/

#define ICW2 0x20 #define ICW4 0x01 #define OCW1 0xfe

#define CS8259A 0xa000 #define CS8259B 0xa001

/\* 中断号为 20H \*/

/\* 工作在 8086/88 方式 \*/

/\* 只响应 INT0 中断 \*/

extern unsigned char IN(unsigned int port);

extern void OUT(unsigned int port, unsigned char v); extern void DISABLE(void);

extern void ENABLE(void);

extern void SETINT(unsigned char NO, unsigned int \*ENTER); unsigned char CNT;

interrupt IEnter()

{

CNT++; OUT(PA8255, CNT);

OUT(CS8259A, 0x20); /\* 中断服务程序结束指令 \*/

}

void IInit()

{

OUT(CS8259A, ICW1); OUT(CS8259B, ICW2); OUT(CS8259B, ICW4); OUT(CS8259B, OCW1);

}

void main()

{

OUT(CTL8255,mode);

DISABLE(); /\* 关闭中断响应 \*/

IInit();

SETINT(ICW2, &IEnter); /\* 中断入口地址 \*/ CNT = 0;

OUT(PA8255, CNT);

ENABLE(); /\* 打开中断响应 \*/

while (1); /\* 等待中断，并计数。\*/

}

// 硬件实验三十六 8253 定时器实验 H36.C

#define CONTROL 0xc003 #define COUNT0 0xc000 #define COUNT1 0xc001 #define COUNT2 0xc002

extern unsigned char IN(unsigned int port);

extern void OUT(unsigned int port, unsigned char v);

void main()

{

OUT(CONTROL, 0x36); /\* 计数器 0,16 位,方式 3,二进制 \*/ OUT(COUNT0, 10000 256); /\* 计数器低字节 \*/

OUT(COUNT0, 10000 / 256); /\* 计数器高字节 \*/

OUT(CONTROL, 0x76); /\* 计数器 1,16 位,方式 3,二进制 \*/ OUT(COUNT1, 100 ); /\* 计数器低字节 \*/

OUT(COUNT1, 0); /\* 计数器高字节 \*/

while (1) ;

}

// 硬件实验三十七 8251A 串行通讯实验 H37.C

#define LEDLen 6 #define IN\_KEY 0x8001 #define OUTBIT 0x8002 #define OUTSEG 0x8004

#define CS8251D 0xd000 #define CS8251C 0xd001

extern unsigned char IN(unsigned int port);

extern void OUT(unsigned int port, unsigned char v);

unsigned char LEDBuf[LEDLen]; /\* 显 示 缓 冲 \*/ unsigned char const LEDMAP[] = { /\* 八段管显示码 \*/

0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71

};

unsigned char const KeyTable[] = { /\* 键码定义 \*/ 0x16, 0x15, 0x14, 0xff,

0x13, 0x12, 0x11, 0x10,

0x0d, 0x0c, 0x0b, 0x0a, 0x0e, 0x03, 0x06, 0x09,

0x0f, 0x02, 0x05, 0x08,

0x00, 0x01, 0x04, 0x07

};

void Delay(unsigned char CNT)

{

unsigned char i; while (CNT-- !=0)

for (i=100; i !=0; i--);

}

void DisplayLED()

{

unsigned char i, j, k; unsigned char Pos; unsigned char LED;

Pos = 0x20; /\* 从左边开始显示 \*/ for (i = 0; i < LEDLen; i++) {

OUT(OUTBIT,0); /\* 关所有八段管 \*/ LED = LEDBuf[i];

OUT(OUTSEG,LED);

OUT(OUTBIT, Pos); /\* 显示一位八段管 \*/ Delay(1);

Pos >>= 1; /\* 显示下一位 \*/

}

}

unsigned char TestKey()

{

OUT(OUTBIT, 0); /\* 输出线置为 0 \*/

return (~IN(IN\_KEY) & 0x0f); /\* 读入键状态 (高四位不用) \*/

}

unsigned char GetKey()

{

unsigned char Pos; unsigned char i; unsigned char j; unsigned char k;

i = 6;

Pos = 0x20; /\* 找出键所在列 \*/ do {

OUT(OUTBIT, ~ Pos);

Pos >>= 1;

k = ~IN(IN\_KEY) & 0x0f;

} while ((--i != 0) && (k == 0));

/\* 键值 = 列 X 4 + 行 \*/ if (k != 0) {

i \*= 4;

if (k & 2) i += 1;

else if (k & 4) i += 2;

else if (k & 8) i += 3;

OUT(OUTBIT, 0);

do Delay(10); while (TestKey()); /\* 等键释放 \*/

return(KeyTable[i]); /\* 取出键码 \*/

} else return(0xff);

}

void IInit()

{

OUT(CS8251C,0x4f); /\* 1 停止位，无校验，8 数据位 \*/ OUT(CS8251C,0x15); /\* 清错误标志，允许接收发送 \*/

}

void Send(unsigned char TBuf)

{

unsigned ss;

OUT(CS8251C,0x15); /\* 清错误标志，允许接收发送 \*/ while( !(IN(CS8251C) & 0x1)); /\* 发 送 缓 冲 是 否 为 空 \*/ OUT(CS8251D,TBuf); /\* 送出数据库 \*/

}

unsigned char Receive()

{

while( !(IN(CS8251C) & 0x2)); /\* 是否已收到数据 \*/ return (IN(CS8251D)); /\* 读入数据 \*/

}

void main()

{

unsigned char RCVBUF;

IInit(); LEDBuf[0] = 0xff; LEDBuf[1] = 0xff; LEDBuf[2] = 0xff; LEDBuf[3] = 0xff; LEDBuf[4] = 0x00; LEDBuf[5] = 0x00;

while (1) {

DisplayLED(); /\* 显示 \*/

if(!(IN(CS8251C) & 0x2)) /\* 是否收到数据 \*/

{

if (TestKey()) /\* 如果有键按下 \*/

Send(GetKey()); /\* 则用串口输出 \*/

} else{

RCVBUF = Receive(); /\* 已收到数据 \*/ LEDBuf[5] = LEDMAP[RCVBUF & 0x0f]; /\* 显示低 4 位 \*/ LEDBuf[4] = LEDMAP[RCVBUF / 0x10]; /\* 显示高 4 位 \*/

}

}

}