```
;program for AGM1232G series using SED1520F0a
;CS1=p3.3 (LT side Enable line)
;CS2=p3.4 (RT side Enable line)
;RES=LOW (using 80 port)
;A0=p3.2   (lo=command hi=data)
;RD=P3.5


      org 00H
      Nop
      Nop
      Nop
      Nop
      Nop
init:       mov a, #0E2H            ;Reset column address, display
            acall ltstrobe          ;startline, page address counter=0.
            mov a, #0AfH            ;display=On
            lcall ltstrobe
            mov a, #0C0H            ;Starts on first line
            lcall ltstrobe
            mov a, #0A4H            ;static driver=Off
            lcall ltstrobe
            mov a, #0A9H            ;duty cycle=1/32
            lcall ltstrobe
            mov a, #0A0H            ;ADC=forward
            lcall ltstrobe
            mov a, #0EEH            ;Read Modified Write=End
            lcall ltstrobe
            lcall clrsc             ;Clear screen routine
            mov R3, #0
            mov R4, #0B8H

nxpg:       mov a, R4
            lcall ltstrobe
            mov a, #0
            lcall ltstrobe
            lcall gen_0
            lcall gen_1
            lcall gen_2
            lcall gen_3
            lcall gen_4
            lcall gen_5
            lcall gen_6
            lcall gen_7
            inc R4
            inc R3
            cjne R3, #4,nxpg
            lcall delay
            mov R3, #0
            mov R4, #0B8H

nxpg1:      mov a, R4
            lcall ltstrobe
            mov a, #0
            lcall ltstrobe
            lcall gen_A
            lcall gen_A
            lcall gen_A
            lcall gen_A
            lcall gen_A
            lcall gen_A
            lcall gen_A
            lcall gen_A
            inc R4
            inc R3
            cjne R3, #4,nxpg1
            lcall delay
            mov R3, #0
            mov R4, #0B8H

nxpg2:      mov a, R4
            lcall ltstrobe
            mov a, #0
            lcall ltstrobe
            lcall gen_B
            lcall gen_B
            lcall gen_B
            lcall gen_B
            lcall gen_B
            lcall gen_B
            lcall gen_B
            lcall gen_B
            inc R4
            inc R3
            cjne R3, #4,nxpg2
            lcall delay
            mov R3, #0
            mov R4, #0B8H

nxpg3:      mov a, R4
            lcall ltstrobe
            mov a, #0
            lcall ltstrobe
            lcall gen_C
            lcall gen_C
            lcall gen_C
            lcall gen_C
            lcall gen_C
            lcall gen_C
            lcall gen_C
            lcall gen_C
            inc R4
            inc R3
            cjne R3, #4,nxpg3
            lcall delay
            ajmp init

stop:       sjmp stop

clrsc:      mov R4, #0B8H           ;loop to clear screen.

nxC:        mov a, R4   ;Starts at page 0
            lcall ltstrobe
            mov R5, #00H    ;Starts at column 0
```

```
nxR:        mov a, R5
            lcall ltstrobe
            mov a, #00H             ;Load 0's to data bus
            lcall ltstrobe1
            inc R5                  ;Next column
            cjne R5, #61, nxR
            inc R4
            cjne R4, #0B8H+4, nxC
            ret

ltstrobe:   lcall status            ; Execute Instruction
            setb p3.5         ;RD=1
            clr p3.2          ;A=0
            setb p3.3               ;CS1=1
            setb p3.4               ;CS2=1
            mov p1, a               ;Execute instruction
            clr p3.3                ;CS1=0
            clr p3.4                ;CS2=0
            clr p3.5          ;RD=0
            ret

ltstrobe1:  nop               ; write to both sides
            setb p3.5         ;RD=1
            setb p3.2               ;A=1
            setb p3.3               ;CS1=1
            setb p3.4               ;CS2=1
            mov p1, a               ;Write display data
            clr p3.3                ;CS1=0
            clr p3.4                ;CS2=0
            clr p3.2                ;A=0
            clr p3.5          ;RD=0
            ret

ltstrobe2:  nop                     ;write to left side
            setb p3.5               ;RD=1
            setb p3.2               ;A=1
            setb p3.3               ;CS1=1
            mov p1, a               ;Write display data
            clr p3.3                ;CS1=0
            clr p3.2                ;A=0
            clr p3.5          ;RD=0
            ret

ltstrobe3:  nop                     ;write to right side
            setb p3.5               ;RD=1
            setb p3.2               ;A=1
            setb p3.4               ;CS2=1
            mov p1, a               ;Write display data
            clr p3.4                ;CS2=0
            clr p3.2                ;A=0
            clr p3.5          ;RD=0
            ret

delay:            mov R7, #32H

cyc2:             mov R6, #0F0H
```

```
cyc1:             mov R5, #0F0H

cyc3:             djnz R5, cyc3
                  djnz R6, cyc1
                  djnz R7, cyc2
            ret

status:     mov R1, a                 ;save data
            clr P3.2                  ; A0=0 for read
            setb P3.6                 ; wr=1
            setb P3.3                 ;Left side enabled
            mov P1, #0FFH             ; load input

stat1:      clr P3.5                  ; rd=0
            nop
            nop
            nop
            mov a, P1                 ; move status
            anl a, #80H               ;check bit 8
            setb P3.5                 ; rd=1
            jnz stat1                 ; jump if A not 0
            clr P3.3                  ;
            mov a, R1
            ret

gen_0:            mov a, #3EH
            lcall ltstrobe1
            mov a, #7FH
            lcall ltstrobe1
            mov a, #71H
            lcall ltstrobe1
            mov a, #59H
            lcall ltstrobe1
            mov a, #4DH
            lcall ltstrobe1
            mov a, #7FH
            lcall ltstrobe1
            mov a, #3EH
            lcall ltstrobe1
            mov a, #00H
            lcall ltstrobe1
            ret

gen_1:            mov a, #40H
            lcall ltstrobe1
            mov a, #42H
            lcall ltstrobe1
            mov a, #7FH
            lcall ltstrobe1
            mov a, #7FH
            lcall ltstrobe1
            mov a, #40H
            lcall ltstrobe1
            mov a, #40H
            lcall ltstrobe1
            mov a, #00H
            lcall ltstrobe1
            ret

gen_2:            mov a, #62H
            lcall ltstrobe1
            mov a, #73H
            lcall ltstrobe1
            mov a, #59H
            lcall ltstrobe1
            mov a, #49H
            lcall ltstrobe1
            mov a, #6FH
            lcall ltstrobe1
            mov a, #66H
            lcall ltstrobe1
            mov a, #00H
            lcall ltstrobe1
            mov a, #00H
            lcall ltstrobe1
            ret

gen_3:            mov a, #22H
            lcall ltstrobe1
            mov a, #63H
            lcall ltstrobe1
            mov a, #49H
            lcall ltstrobe1
            mov a, #49H
            lcall ltstrobe1
            mov a, #7FH
            lcall ltstrobe1
            mov a, #36H
            lcall ltstrobe1
            mov a, #00H
            lcall ltstrobe1
            ret

gen_4:            mov a, #18H
            lcall ltstrobe1
            mov a, #1CH
            lcall ltstrobe1
            mov a, #16H
            lcall ltstrobe1
            mov a, #53H
            lcall ltstrobe1
            mov a, #7FH
            lcall ltstrobe1
            mov a, #7FH
            lcall ltstrobe1
            mov a, #50H
            lcall ltstrobe1
            mov a, #00H
            lcall ltstrobe1
            ret

gen_5:            mov a, #27H
            lcall ltstrobe1
            mov a, #67H
            lcall ltstrobe1
            mov a, #45H
            lcall ltstrobe1
            mov a, #45H
            lcall ltstrobe1
            mov a, #7DH
            lcall ltstrobe1
            mov a, #39H
            lcall ltstrobe1
            mov a, #00H
            lcall ltstrobe1
            mov a, #00H
            lcall ltstrobe1
            ret

gen_6:            mov a, #3CH
            lcall ltstrobe1
            mov a, #7EH
            lcall ltstrobe1
            mov a, #4BH
            lcall ltstrobe1
            mov a, #49H
            lcall ltstrobe1
            mov a, #79H
            lcall ltstrobe1
            mov a, #30H
            lcall ltstrobe1
            mov a, #00H
            lcall ltstrobe1
            ret

gen_7:            mov a, #03H
            lcall ltstrobe1
            mov a, #03H
            lcall ltstrobe1
            mov a, #71H
            lcall ltstrobe1
            mov a, #79H
            lcall ltstrobe1
            mov a, #0FH
            lcall ltstrobe1
            mov a, #07H
            lcall ltstrobe1
            mov a, #00H
            lcall ltstrobe1
            mov a, #00H
            lcall ltstrobe1
            ret

gen_A:            mov a, #0AAH
            lcall ltstrobe2
            mov a, #55H
            lcall ltstrobe3
            mov a, #55H
            lcall ltstrobe2
            mov a, #0AAH
            lcall ltstrobe3
            mov a, #0AAH
            lcall ltstrobe2
            mov a, #55H
            lcall ltstrobe3
            mov a, #55H
            lcall ltstrobe2
            mov a, #0AAH
            lcall ltstrobe3
            mov a, #0AAH
            lcall ltstrobe2
            mov a, #55H
            lcall ltstrobe3
            mov a, #55H
            lcall ltstrobe2
            mov a, #0AAH
            lcall ltstrobe3
            mov a, #0AAH
            lcall ltstrobe2
            mov a, #55H
            lcall ltstrobe3
            mov a, #55H
            lcall ltstrobe2
            mov a, #0AAH
            lcall ltstrobe3
            ret

gen_B:            mov a, #55H
            lcall ltstrobe2
            mov a, #0AAH
            lcall ltstrobe3
            mov a, #0AAH
            lcall ltstrobe2
            mov a, #55H
            lcall ltstrobe3
            mov a, #55H
            lcall ltstrobe2
            mov a, #0AAH
            lcall ltstrobe3
            mov a, #0AAH
            lcall ltstrobe2
            mov a, #55H
            lcall ltstrobe3
            mov a, #55H
            lcall ltstrobe2
            mov a, #0AAH
            lcall ltstrobe3
            mov a, #0AAH
            lcall ltstrobe2
            mov a, #55H
            lcall ltstrobe3
            mov a, #55H
            lcall ltstrobe2
            mov a, #0AAH
            lcall ltstrobe3
            mov a, #0AAH
            lcall ltstrobe2
            mov a, #55H
            lcall ltstrobe3
            ret

gen_C:            mov a, #0FFH
            lcall ltstrobe1
            mov a, #0FFH
            lcall ltstrobe1
            mov a, #0FFH
            lcall ltstrobe1
            mov a, #0FFH
            lcall ltstrobe1
            mov a, #0FFH
            lcall ltstrobe1
            mov a, #0FFH
            lcall ltstrobe1
            mov a, #0FFH
            lcall ltstrobe1
            mov a, #0FFH
            lcall ltstrobe1
            ret


END
```