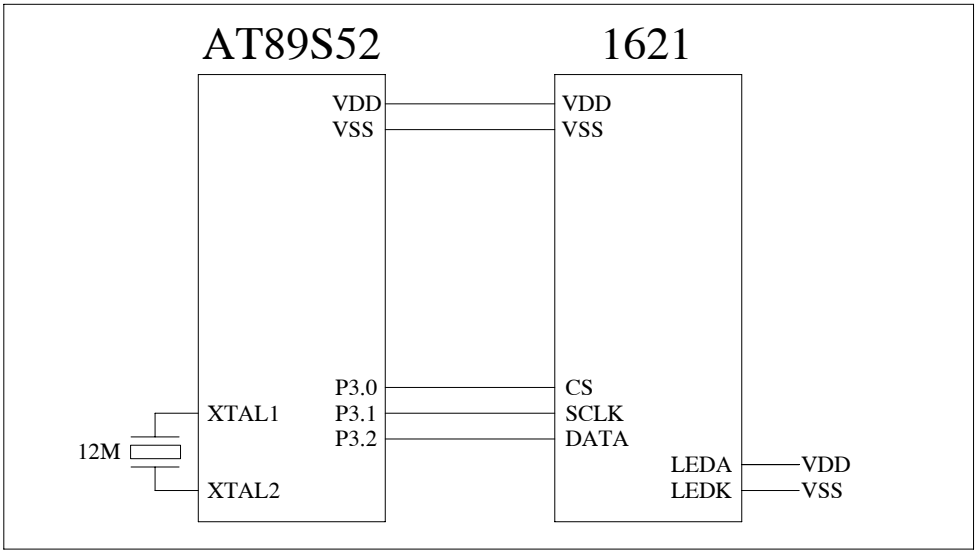
深圳秋田微电子有限公司
SHENZHEN AV-DISPLAY CO.,LTD.

Reference Program for LCD Modules	Version No.	A00
	Type	COB
Remark	Segment Series, with HT1621 or compatible IC	

1. Interface



2. Instruction Code

Command Summary

Name	ID	Command Code	D/C	Function	Def.
READ	1 1 0	A5A4A3A2A1A0D0D1D2D3	D	Read data from the RAM	
WRITE	1 0 1	A5A4A3A2A1A0D0D1D2D3	D	Write data to the RAM	
READ-MODIFY-WRITE	1 0 1	A5A4A3A2A1A0D0D1D2D3	D	READ and WRITE to the RAM	
SYS DIS	1 0 0	0000-0000-X	C	Turn off both system oscillator and LCD bias generator	Yes
SYS EN	1 0 0	0000-0001-X	C	Turn on system oscillator	
LCD OFF	1 0 0	0000-0010-X	C	Turn off LCD bias generator	Yes
LCD ON	1 0 0	0000-0011-X	C	Turn on LCD bias generator	
TIMER DIS	1 0 0	0000-0100-X	C	Disable time base output	
WDT DIS	1 0 0	0000-0101-X	C	Disable WDT time-out flag output	
TIMER EN	1 0 0	0000-0110-X	C	Enable time base output	
WDT EN	1 0 0	0000-0111-X	C	Enable WDT time-out flag output	
TONE OFF	1 0 0	0000-1000-X	C	Turn off tone outputs	Yes
TONE ON	1 0 0	0000-1001-X	C	Turn on tone outputs	
CLR TIMER	1 0 0	0000-11XX-X	C	Clear the contents of time base generator	
CLR WDT	1 0 0	0000-111X-X	C	Clear the contents of WDT stage	

XTAL 32K	**1 0 0**	0001-01XX-X	C	System clock source, crystal oscillator	
RC 256K	**1 0 0**	0001-10XX-X	C	System clock source, on-chip RC oscillator	Yes
EXT 256K	**1 0 0**	0001-11XX-X	C	System clock source, external clock source	
BIAS 1/2	**1 0 0**	0010-abX0-X	C	LCD 1/2 bias option ab=00: 2 commons option ab=01: 3 commons option ab=10: 4 commons option	
BIAS 1/3	**1 0 0**	0010-abX1-X	C	LCD 1/3 bias option ab=00: 2 commons option ab=01: 3 commons option ab=10: 4 commons option	
TONE 4K	**1 0 0**	010X-XXXX-X	C	Tone frequency, 4kHz	
TONE 2K	**1 0 0**	011X-XXXX-X	C	Tone frequency, 2kHz	
$\overline{\text{IRQ}}$ DIS	**1 0 0**	100X-0XXX-X	C	Disable $\overline{\text{IRQ}}$ output	Yes
$\overline{\text{IRQ}}$ EN	**1 0 0**	100X-1XXX-X	C	Enable $\overline{\text{IRQ}}$ output	
F1	**1 0 0**	101X-X000-X	C	Time base/WDT clock output:1Hz The WDT time-out flag after: 4s	
F2	**1 0 0**	101X-X001-X	C	Time base/WDT clock output:2Hz The WDT time-out flag after: 2s	
F4	**1 0 0**	101X-X010-X	C	Time base/WDT clock output:4Hz The WDT time-out flag after: 1s	
F8	**1 0 0**	101X-X011-X	C	Time base/WDT clock output:8Hz The WDT time-out flag after: 1/2 s	
F16	**1 0 0**	101X-X100-X	C	Time base/WDT clock output:16Hz The WDT time-out flag after: 1/4 s	
F32	**1 0 0**	101X-X101-X	C	Time base/WDT clock output:32Hz The WDT time-out flag after: 1/8 s	
F64	**1 0 0**	101X-X110-X	C	Time base/WDT clock output:64Hz The WDT time-out flag after: 1/16 s	
F128	**1 0 0**	101X-X111-X	C	Time base/WDT clock output:128Hz The WDT time-out flag after: 1/32 s	Yes
TEST	**1 0 0**	1110-0000-X	C	Test mode, user don't use.	
NORMAL	**1 0 0**	1110-0011-X	C	Normal mode	Yes

Note: X : Don't care

A5~A0 : RAM addresses

D3~D0 : RAM data

D/C : Data/command mode

Def. : Power on reset default

All the bold forms, namely **1 1 0**, **1 0 1**, and **1 0 0**, are mode commands. Of these, **1 0 0** indicates the command mode ID. If successive commands have been issued, the command mode ID except for the first command will be omitted. The source of the tone frequency and of the time base/WDT clock frequency can be derived from an on-chip 256kHz RC oscillator, a 32.768kHz crystal oscillator, or an external 256kHz clock. Calculation of the frequency is based on the system frequency sources as stated above. It is recommended that the host controller should initialize the HT1621 after power on reset, for power on reset may fail, which in turn leads to the malfunctioning of the HT1621.

Remark: For the detail instruction for the control function, please refer to the related

manual data book for controller IC.

3. Reference Program

```
//====== 89S52 MCU, 12M Oscillator ===========
#include <reg51.h>
#include <intrins.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#define uint unsigned int
#define uchar unsigned char
#define xchar unsigned char code

sbit CS    = P3^0;
sbit SCLK  = P3^1;
sbit DATA  = P3^2;
//-----------------------------------------
void delayms(uint v)
{ while(v!=0)v--;}
//-----------------------------------------
void delay(uint nn)
{
  uint mm;
  while(nn-->0)
  for(mm=0;mm<1000;mm++) { };
}
//-----------------------------------------
void clock()
{
  WWR=0;delayms(2);
  WWR=1;delayms(2);
}
//-----------------------------------------
void wc100()
{
  DATA=1;
  clock();
  DATA=0;
  clock();
  DATA=0;
  clock();
}
//-----------------------------------------
void wc101()
{
  DATA=1;
  clock();
  DATA=0;
  clock();
```

```
  DATA=1;
  clock();
}
//-----------------------------------------
void wcomd(uchar cdat)
{
  uint i;
  for(i=0;i<8;i++)
  {
    cdat<<=1;
    DATA=CY;
    clock();
  }
  DATA=1;
  clock();
}
//-----------------------------------------
void waddress(uchar cadd)
{
  uint i;
  cadd<<=1;cadd<<=1;
  for(i=2;i<8;i++)
  {
    cadd<<=1;
    DATA=CY;
    clock();
  }
}
//-----------------------------------------
void initial()
{
  WWR=1;CS=0;
  wc100();
  wcomd(0x01); //SYS EN
  wcomd(0x03); //LCD ON
  //wcomd(0x14); //CRYSTAL OSCILLATOR
  wcomd(0x18); //RC OSCILLATOR
  wcomd(0x29); //4 COM
  wcomd(0xe3); //NORMAL MODE
  CS=1;
}
//-----------------------------------------
void disp_all_single(uchar dall)
{
  uint i;
  for(i=0;i<32;i++)
  {
    WWR=1;CS=0;
    wc101();
```

```
    waddress(i);
    DATA=dall;clock();
    DATA=dall;clock();
    DATA=dall;clock();
    DATA=dall;clock();
    CS=1;
  }
}
//-----------------------------------------
void disp_seg_single(uint dtime)
{
  uint i;
  for(i=0;i<32;i++)
  {
    WWR=1;CS=0;
    wc101();
    waddress(i);
    DATA=1;clock();
    DATA=0;clock();
    DATA=0;clock();
    DATA=0;clock();
    CS=1;
    delay(dtime);
    WWR=1;CS=0;
    wc101();
    waddress(i);
    DATA=1;clock();
    DATA=1;clock();
    DATA=0;clock();
    DATA=0;clock();
    CS=1;
    delay(dtime);
    WWR=1;CS=0;
    wc101();
    waddress(i);
    DATA=1;clock();
    DATA=1;clock();
    DATA=1;clock();
    DATA=0;clock();
    CS=1;
    delay(dtime);
    WWR=1;CS=0;
    wc101();
    waddress(i);
    DATA=1;clock();
    DATA=1;clock();
    DATA=1;clock();
    DATA=1;clock();
    CS=1;
```

```
    delay(dtime);
  }
}
//-----------------------------------------
///////////////  MAIN  ////////////////////////
/////////////////////////////////////////////
void main(void)
{
  delay(2);
  initial();

  disp_all_single(1);delay(150);
  disp_all_single(0);delay(50);

  while(1)
  {
    disp_seg_single(10);delay(50);
    disp_all_single(0);delay(50);
  }
}
```