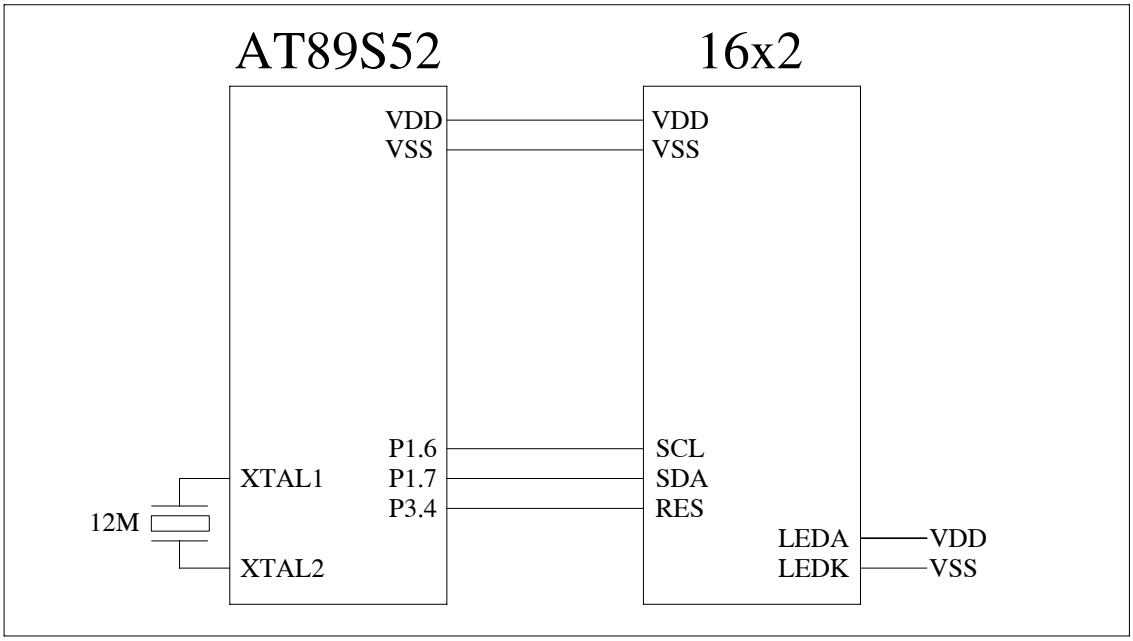
深圳秋田微电子有限公司
SHENZHEN AV-DISPLAY CO.,LTD.

Reference Program for LCD Modules	Version No.	A00
	Type	COG
Remark	16x2(COG) character dot matrix series, with ST7032 or compatible IC	

1. Interface



2. Instruction Code

■ Instructions

There are four categories of instructions that:

- **Designate ST7032 functions, such as display format, data length, etc.**
- **Set internal RAM addresses**
- **Perform data transfer with internal RAM**
- **Others**

➢ instruction table at "Normal mode"

(When "EXT" option pin connect to VDD, the instruction set follow below table)

Instruction	Instruction Code										Description	Instruction Execution Time		
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		OSC= 380KHz	OSC= 540kHz	OSC= 700KHz
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM. and set DDRAM address to "00H" from AC	1.08 ms	0.76 ms	0.59 ms
Return Home	0	0	0	0	0	0	0	0	1	x	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.08 ms	0.76 ms	0.59 ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.	26.3 us	18.5 us	14.3 us
Display ON/OFF	0	0	0	0	0	0	1	D	C	B	D=1:entire display on C=1:cursor on B=1:cursor position on	26.3 us	18.5 us	14.3 us

Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	x	x	S/C and R/L: Set cursor moving and display shift control bit, and the direction, without changing DDRAM data.	26.3 us	18.5 us	14.3 us
Function Set	0	0	0	0	1	DL	N	x	x	x	DL: interface data is 8/4 bits N: number of line is 2/1	26.3 us	18.5 us	14.3 us
Set CGRAM	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter	26.3 us	18.5 us	14.3 us
Set DDRAM address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter	26.3 us	18.5 us	14.3 us
Read Busy flag and address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.	0	0	0
Write data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM)	26.3 us	18.5 us	14.3 us
Read data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM)	26.3 us	18.5 us	14.3 us

Note:
Be sure the ST7032 is not in the busy state (BF = 0) before sending an instruction from the MPU to the ST7032. If an instruction is sent without checking the busy flag, the time between the first instruction and next instruction will take much longer than the instruction time itself. Refer to Instruction Table for the list of each instruction execution time.

➢ instruction table at "Extension mode"

(when "EXT" option pin connect to Vss, the instruction set follow below table)

Instruction	Instruction Code										Description	Instruction Execution Time		
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		OSC= 380KHz	OSC= 540kHz	OSC= 700KHz
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM. and set DDRAM address to "00H" from AC	1.08 ms	0.76 ms	0.59 ms
Return Home	0	0	0	0	0	0	0	0	1	x	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.08 ms	0.76 ms	0.59 ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.	26.3 us	18.5 us	14.3 us
Display ON/OFF	0	0	0	0	0	0	1	D	C	B	D=1:entire display on C=1:cursor on B=1:cursor position on	26.3 us	18.5 us	14.3 us
Function Set	0	0	0	0	1	DL	N	DH	*0	IS	DL: interface data is 8/4 bits N: number of line is 2/1 DH: double height font IS: instruction table select	26.3 us	18.5 us	14.3 us
Set DDRAM address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter	26.3 us	18.5 us	14.3 us
Read Busy flag and address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.	0	0	0
Write data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM/ICONRAM)	26.3 us	18.5 us	14.3 us
Read data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM/ICONRAM)	26.3 us	18.5 us	14.3 us

Note * : this bit is for test command , and must always set to "0"

Instruction table 0(IS=0)														
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	x	x	S/C and R/L: Set cursor moving and display shift control bit, and the direction, without changing DDRAM data.	26.3 us	18.5 us	14.3 us
Set CGRAM	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter	26.3 us	18.5 us	14.3 us

Instruction table 1(IS=1)														
Internal OSC frequency	0	0	0	0	0	1	BS	F2	F1	F0	BS=1:1/4 bias BS=0:1/5 bias F2~0: adjust internal OSC frequency for FR frequency.	26.3 us	18.5 us	14.3 us
Set ICON address	0	0	0	1	0	0	AC3	AC2	AC1	AC0	Set ICON address in address counter.	26.3 us	18.5 us	14.3 us
Power/ICON control/Contrast set	0	0	0	1	0	1	Ion	Bon	C5	C4	Ion: ICON display on/off Bon: set booster circuit on/off C5,C4: Contrast set for internal follower mode.	26.3 us	18.5 us	14.3 us
Follower control	0	0	0	1	1	0	Fon	Rab 2	Rab 1	Rab 0	Fon: set follower circuit on/off Rab2~0: select follower amplified ratio.	26.3 us	18.5 us	14.3 us
Contrast set	0	0	0	1	1	1	C3	C2	C1	C0	Contrast set for internal follower mode.	26.3 us	18.5 us	14.3 us

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;

xchar  send1[8]={0x0e,0x04,0x0e,0x15,0x15,0x0e,0x04,0x0e};
xchar  send2[8]={0x11,0x11,0x11,0x1f,0x11,0x11,0x11,0x00};
xchar  send3[8]={0x0a,0x0a,0x1f,0x0a,0x1f,0x0a,0x0a,0x00};
uchar tempdat;

//-----------------------------------------
void delayms(uint v)
{ while(v!=0)v--;}
//-----------------------------------------
void delay(uint nn)
```

```
{
  uint mm;
  while(nn-->0)
  for(mm=0;mm<1000;mm++) { };
}
//-----------------------------------------
void  start()
{
// I2c  start
  SDA=1;
  delayms(1);
  SCL=1;
  delayms(1);
  SDA=0;
  delayms(3);
}
//-----------------------------------------
void  stop()
{
// I2c stop
  SCL=0;
  delayms(1);
  SDA=0;
  delayms(1);
  SCL=1;
  delayms(1);
  SDA=1;
  delayms(3);
}
//-----------------------------------------
void wcomd(uchar cdat)
{
  uint i;
  for(i=0;i<8;i++)
  {
    SCL=0;
    cdat<<=1;
    SDA=CY;
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();

    SCL=1;
    _nop_();
    _nop_();
    _nop_();
```

```
    _nop_();
    _nop_();
    _nop_();

  }
   SCL=0;
   delayms(3);
   SDA=1;
   SCL=1;
}
//-----------------------------------------
void show(uchar cdat1,uchar cdat2)
{
  unsigned char data n;
  start();
  wcomd(0x7c);
  wcomd(0x80);
  wcomd(0x28);
  wcomd(0x80);
  wcomd(0x80);
  wcomd(0x40);
  for(n=0;n<16;n++)
  { wcomd(cdat1);}
  stop();
  start();
  wcomd(0x7c);
  wcomd(0x80);
  wcomd(0x28);
  wcomd(0x80);
  wcomd(0xc0);
  wcomd(0x40);
  for(n=0;n<16;n++)
  { wcomd(cdat2);}
  stop();
}
//-----------------------------------------
void showicon(uchar dat)
{
  unsigned char data n;
  start();
  wcomd(0x7c);
  wcomd(0x80);
  wcomd(0x39);
  wcomd(0x80);
  wcomd(0x40);
  wcomd(0x40);
  for(n=0;n<16;n++)
  { wcomd(dat);}
  stop();
```

```
}
//-----------------------------------------
void showcgram(uchar *send)
{
  unsigned char data n;
  start();
  wcomd(0x7c);
  wcomd(0x80);
  wcomd(0x28);
  wcomd(0x80);
  wcomd(0x40);
  wcomd(0x40);
  for(n=0;n<32;n++)
  { wcomd(send[n]);}
  stop();
}
//-----------------------------------------
void  initial()
{
  start();
  wcomd(0x7c);    //SET ADDRESS
  wcomd(0x80);    //control byte

  wcomd(0x38);    //funtion set,is=0
  wcomd(0x80);    //control byte

  wcomd(0x06);    //entry mode set
  wcomd(0x80);    //control byte

  wcomd(0x0c);    //display on
  wcomd(0x80);    //control byte

  wcomd(0x01);    //display clear
  delay(5);
  wcomd(0x80);    //control byte

  wcomd(0x39);    //funtion set,is=1
  wcomd(0x80);    //control byte

  wcomd(0x14);    //internal osc frequency,1/5 bias
  wcomd(0x80);    //control byte

  wcomd(0x5f);    //set icon on,booster on,contrast set: D0-D1
  wcomd(0x80);    //control byte

  wcomd(0x6b);    //set follower circuit on,set Ra/Rb:D0-D3
  wcomd(0x80);    //control byte

  wcomd(0x70);    //contrast set:D0-D3
```

```
  wcomd(0x80);    //control byte
  stop();
}
//-----------------------------------------
///////////////  MAIN  ////////////////////////
/////////////////////////////////////////////
void main()
{
  uchar i,j;
  RES=0;
  delay(10);
  RES = 1;
  delay (2);
  initial();
  i=0;j=0;

  while(1)
  {
    //ROM
    showicon(0xff);
    show(0x1c,0x1c);
    delay(70);

    show(0x48,0x48);
    delay(70);

    show(0x23,0x23);
    delay(70);
    //RAM
    showicon(0x00);
    showcgram(send1);
    show(0x00,0x00);
    delay(70);

    showcgram(send2);
    show(0x00,0x00);
    delay(70);

    showcgram(send3);
    show(0x00,0x00);
    delay(70);
  }
}
```