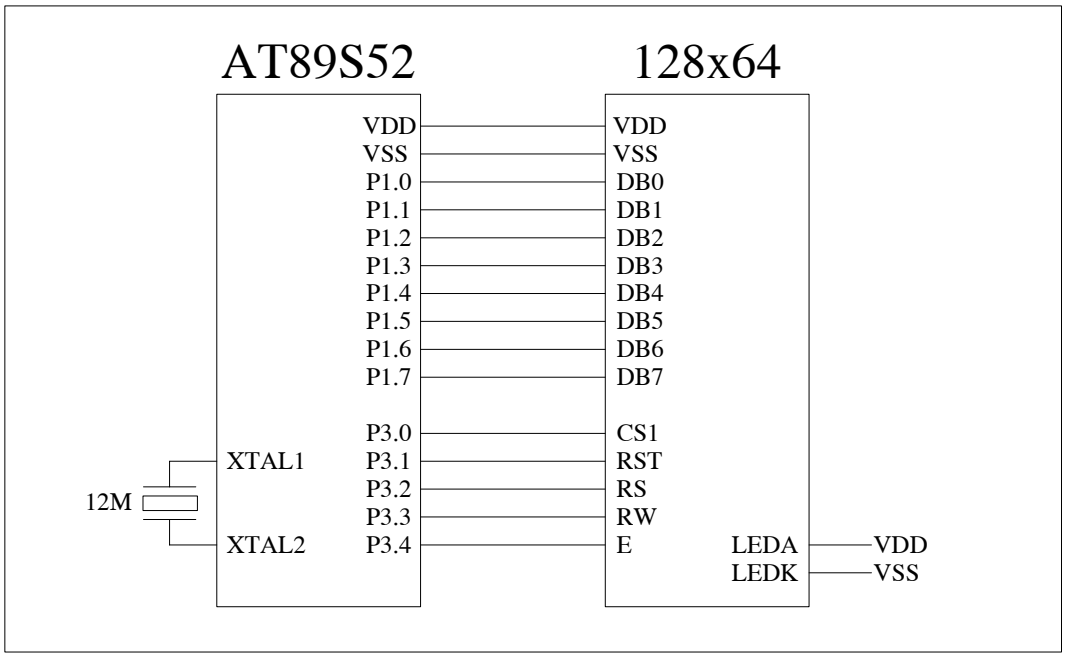
深圳秋田微电子有限公司
SHENZHEN AV-DISPLAY CO.,LTD.

Reference Program for LCD Modules	Version No.	A00
	Type	COG
Remark:	128x64 Graphic Type LCM, with ST7565P or compatible IC	

1. Interface



2. Instruction Code

Instruction	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Function
Display ON/OFF	L	L	L	L	H	H	H	H	H	L/H	Controls the display on or off. Internal status and display RAM data is not affected. L:OFF, H:ON
Set Address (Y address)	L	L	L	H	Y address (0~63)						Sets the Y address in the Y address counter.
Set Page (X address)	L	L	H	L	H	H	H	Page (0~7)			Sets the X address at the X address register.
Display Start Line (Z address)	L	L	H	H	Display start line (0~63)						Indicates the display data RAM displayed at the top of the screen.
Status Read	L	H	BUSY	L	ON/OFF	RESET	L	L	L	L	Read status. BUSY L: Ready H: In operation ON/OFF L: Display ON H: Display OFF RESET L: Normal H: Reset
Write Display Data	H	L	Write Data								Writes data (DB0:7) into display data RAM. After writing instruction, Y address is increased by 1 automatically.
Read Display Data	H	H	Read Data								Reads data (DB0:7) from display data RAM to the data bus.

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

//////6800///////
sbit CS1 = P3^0;
sbit RST = P3^1;
sbit RS  = P3^2;
sbit RW  = P3^3;
sbit E   = P3^4;
//------------------------------------
xchar xshm[1024]={
0xFF,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,
0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,
0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,
0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,
0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,
0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,
0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,
0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0xFF,
0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,
0xFF,0x00,0x00,0x00,0x00,0x20,0x20,0x20,0xF0,0xF0,0x10,0x18,0x18,0x10,0x00,0x00,
0x00,0xFC,0xFC,0xF8,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xE0,0xE0,
0x40,0x40,0x40,0x40,0x40,0x40,0xC0,0xC0,0x40,0x40,0x40,0x40,0x40,0x40,0xE0,0xE0,
0xE0,0x00,0x00,0x00,0x00,0x80,0xF0,0x70,0x30,0x00,0x40,0xC0,0xC0,0xF8,0xF8,0xF0,
0xC0,0xC0,0xC0,0x00,0x00,0xF8,0xF8,0x30,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xF8,0xF8,0xF0,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,
0x20,0x20,0x20,0x20,0x20,0xA0,0xE0,0xE0,0x70,0x70,0x20,0x00,0x00,0x00,0x00,0xFF,
```

```
0xFF,0x00,0x00,0x04,0x04,0x04,0xC4,0xF4,0xFF,0xFF,0xC4,0xC6,0x86,0x36,0x3C,0x3F,
0x03,0xFF,0xFF,0xFF,0xF0,0x1C,0x0E,0x07,0x03,0x01,0x00,0x00,0x00,0x00,0xFF,0xFF,
0x40,0x40,0x40,0x40,0x40,0x40,0xFF,0xFF,0x40,0x40,0x40,0x40,0x40,0x40,0xFF,0xFF,
0xFF,0x00,0x00,0x02,0x03,0xC1,0xF0,0xFC,0xFE,0x0E,0x04,0x47,0x4F,0x4F,0x47,0x4F,
0x6F,0x2F,0xC7,0xF0,0xFF,0xFF,0x05,0x04,0xFC,0xFC,0x0E,0x06,0x04,0x00,0x00,0xFF,
0xFF,0xFF,0x42,0x42,0x42,0x42,0x42,0xFF,0xFF,0xFF,0x42,0x42,0x42,0x42,0x42,0xFF,
0xFF,0xFF,0x00,0x00,0x00,0x40,0x40,0x40,0x40,0x40,0x40,0x40,0x40,0x40,0x40,0x40,
0xFC,0xFC,0xFE,0x4A,0x43,0x41,0x40,0x40,0x40,0x60,0x60,0x60,0x60,0x40,0x00,0xFF,
0xFF,0x00,0x20,0x30,0x1C,0x0F,0x03,0x01,0xFF,0xFF,0x00,0x01,0x03,0x83,0xC0,0xF0,
0x7E,0x1F,0x07,0x01,0x0F,0x3E,0xF8,0xE0,0xC0,0x80,0x00,0x00,0x00,0x00,0xFF,0xFF,
0x80,0x80,0x80,0x80,0x80,0x80,0xFF,0xFF,0x80,0x80,0x80,0x80,0x80,0x80,0xFF,0xFF,
0xFF,0x00,0x00,0x02,0x03,0x01,0x00,0xFF,0xFF,0x00,0x00,0xFF,0xFF,0x1E,0x02,0xFE,
0xFE,0xFE,0x30,0x10,0x80,0xDF,0x7F,0xFF,0xFF,0x83,0x00,0x00,0x00,0x00,0x00,0x3F,
0x3F,0x1F,0x08,0x08,0x08,0x08,0x08,0xFF,0xFF,0xFF,0x08,0x08,0x08,0x08,0x08,0x1F,
0x1F,0x0F,0xE0,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0xFF,0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,
0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x03,0x06,0x06,0x03,0x01,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x03,0x03,0x01,0x01,0x00,0x00,0x01,0x01,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x01,
0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x07,0x07,0x02,0x03,0x01,0x00,0x08,0x08,0x0C,
0x04,0x06,0x03,0x03,0x01,0x00,0x00,0x00,0x01,0x03,0x03,0x02,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x07,0x07,0x04,0x04,0x04,0x04,0x04,0x04,
0x04,0x04,0x07,0x07,0x07,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x03,0x02,0x06,0x06,
0x07,0x07,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,
0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,
0xFF,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,
0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,
0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,
0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,
0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,
0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,
0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,
0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0xFF
};
//-----------------------------------------------
void delayms(uint v)
{ while(v!=0)v--;}
//-----------------------------------------------
void delay(uint nn)
{
  uint mm;
  while(nn-->0)
```

```
  for(mm=0;mm<1000;mm++) { };
}
//-----------------------------------------
void delayms(uint ms)
{
  uint mm;
  for(mm=0;mm<ms;mm++) { }
}
//-----------------------------------------
void delay(uint nnn)
{
  uint mmm;
  while(nnn-->0)
  for(mmm=0;mmm<1000;mmm++) { }
}
//-----------------------------------------
void wcomd(uchar c)
{
  RW=0;RS=0;CS1=0;P1=c;E=1;delayms(2);E=0;CS1=1;
}
//-----------------------------------------
void wdata(uchar d)
{
  RW=0;RS=1;CS1=0;P1=d;E=1;delayms(2);E=0;CS1=1;
}
//-----------------------------------------
void initial()
{
  wcomd(0xae);
  wcomd(0xa6);
  wcomd(0xa1);
  wcomd(0xc8);
  wcomd(0xa2);
  wcomd(0x2f);
  wcomd(0x26);
  wcomd(0x81);
  wcomd(0x24);
  wcomd(0xaf);
}
//-----------------------------------------
void disp_all(uint xx,uint yy)
{
  uint i=0;
  uchar temp;
  for(temp=0xb0;temp<0xb8;temp++)
  {
    wcomd(temp);
    wcomd(0x10);wcomd(0x04);
    for(i=0;i<64;i++)
```

```
    { wdata(xx);wdata(yy);}
  }
}


//-----------------------------------------
void disp_bmp(xchar *str)
{
  uint ib=0,jb=0;
  uchar temp;
  temp=0xb0;
  for(jb=0;jb<8;jb++)
  {
    wcomd(temp);
    wcomd(0x10);wcomd(0x00);
    for(ib=0;ib<128;ib++)
    {
      wdata(str[ib+jb*128]);
    }
    temp++;
  }
}
//-----------------------------------------
///////////////  MAIN  ////////////////////////
////////////////////////////////////////////////
void main(void)
{
  RST=0;;delay(10);RST=1;delay(2);
  initial();
  disp_all(0x00,0x00);
  while(1)
  {
    disp_all(0xff,0xff);  delay(100);
    disp_all(0x00,0x00);  delay(100);
    disp_all(0xaa,0x55);  delay(100);
    disp_all(0x55,0xaa);  delay(100);
    disp_all(0xff,0x00);  delay(100);
    disp_all(0x00,0xff);  delay(100);
  }
}
```