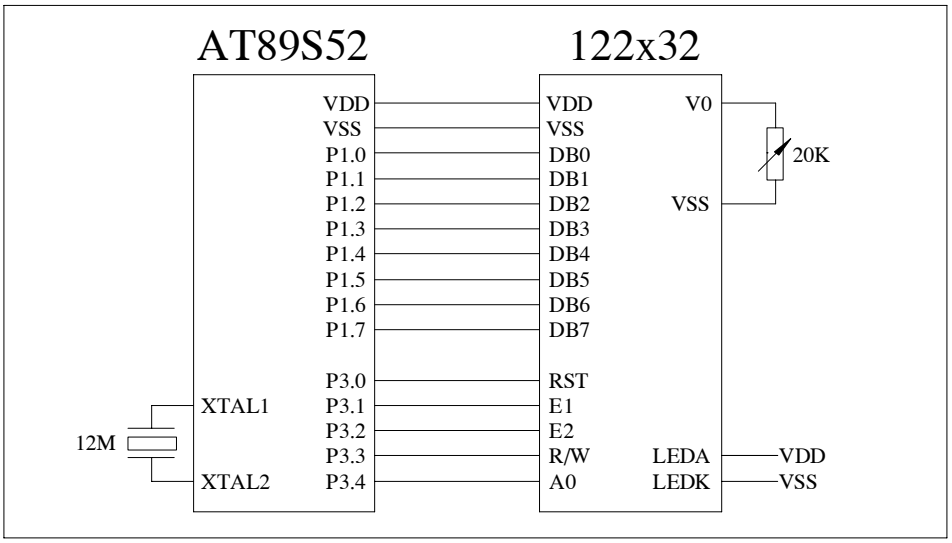
深圳秋田微电子有限公司
SHENZHEN AV-DISPLAY CO.,LTD.

Reference Program for LCD Modules	Version No.	A00
	Type	COB
Remark:	122x32 Graphic dot matrix series, with PT6520 or compatible IC	

1. Interface



2. Instruction Code

DISPLAY COMMANDS

(Based on the 80 port MPU; the RD and WR commands differ for the 68 port MPU)

	Command	RD	WR	A0	D7	D6	D5	D4	D3	D2	D1	D0	Function
1	Display ON/OFF	1	0	0	1	0	1	0	1	1	1	0/1	Switches the entire display ON or OFF, regardless of the Display RAM's data or the internal status. *
2	Display START Line	1	0	0	1	1	0	Display START address (0-31)					Determines the line of RAM data to be displayed at the display's top line (COM0)
3	Page Address Set	1	0	0	1	0	1	1	1	0	Page (0-3)		Sets the page of the Display RAM in the page address register.
4	Column (segment) Address Set	1	0	0	0	Column address (0-79)							Sets the column address of the Display RAM in the column address register.
5	Status Read	0	1	0	BUSY	ACC	ON/OFF	RESET	0	0	0	0	Read the status. Busy 1: Busy (internal processing) 0: Ready status ADC 1: Rightward (forward) output 0: Leftward (reverse) output ON/OFF 1: Display OFF 0: Display ON RESET 1: Resetting. 0: Normal
6	Write Display Data	1	0	1	Write Data								Writes the data on the data bus to RAM
7	Read Display Data	0	1	1	Read Data								Reads data from the Display RAM onto the data bus.

These commands access a previously-specified address of the Display RAM, after which the column address is incremented by one.

8	ADC Select	1	0	0	1	0	1	0	0	0	0	0/1	Used to reverse the correspondence between the Display RAM's column address and segment driver output ports 0: Rightward (forward)output 1: Leftward (reverse) output
9	Static Drive ON/OFF	1	0	0	1	0	1	0	0	1	0	0/1	Selects normal display operation or static all-lit drive display operation. 1: Static drive (power save)* 0: Normal display operation
10	Duty Select	1	0	0	1	0	1	0	1	0	0	0/1	Selects the duty factor for driving LCD cells. 1: 1/32 duty, 0: 1/16 duty
11	Read Modify Write	1	0	0	1	1	1	0	0	0	0	0	Increments column address counter by 1 when display is written. (This is not done when data is read)
12	End	1	0	0	1	1	1	0	1	1	1	0	Cancels the Ready Modify Write mode.
13	Reset	1	0	0	1	1	1	0	0	0	1	0	Resets the display START line to the 1st line in the register. Resets the column address counter to 0 and page address to 0.

Remark: For the detail instruction for the control function, please refer to the related manual data book for controller IC.

3. Reference Program

```
//====== AT89S52 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 RST = P3^0;
sbit E1  = P3^1;
sbit E2  = P3^2;
sbit RW  = P3^3;
sbit A0  = P3^4;
sbit BF_12232  = ACC^7;
xchar xshm[488]={
0xFF,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x81,0x81,0x81,0x81,0x01,0x01,
0x01,0xC1,0xC1,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,
0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,
0x01,0x01,0x01,0x01,0x01,0x81,0x81,0x81,0x01,0x01,0x01,0xC1,0xC1,0x01,0x01,0x01,
0x01,0xC1,0xC1,0x81,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,
0x01,0x01,0xC1,0xC1,0xC1,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,
```

```
0x01,0x01,0x01,0x81,0x81,0x81,0x81,0x81,0x81,0x81,0x81,0x81,0x81,0x81,0x81,0x81,
0x81,0x81,0x81,0x01,0x01,0x01,0x01,0x01,0x01,0xFF,0xFF,0x00,0x00,0x00,0x00,0x10,
0x10,0x12,0x93,0xFF,0xFF,0xFF,0x18,0x58,0x78,0x78,0x80,0xFF,0xFF,0xE0,0x30,0x1C,
0x0C,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,0xFF,0x82,0x82,0x82,0x82,0xFE,
0xFE,0x82,0x82,0x82,0x82,0x82,0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x04,0x86,0xE3,
0xF9,0xF8,0x58,0x5E,0x5E,0x5F,0x5F,0x5F,0xDF,0xE0,0xFE,0xFF,0x0B,0xF8,0xF8,0x08,
0x08,0x00,0x00,0x00,0x00,0x00,0xFC,0xFC,0xFC,0x44,0x44,0x44,0xFF,0xFF,0xFF,0x44,
0x44,0x44,0x44,0xFC,0xFC,0x00,0x00,0x00,0x00,0x00,0x00,0x40,0x40,0x40,0x40,0x40,
0x40,0x40,0x40,0x40,0xF8,0xF8,0x58,0x4C,0x46,0x43,0x41,0x41,0x40,0x40,0x40,0x00,
0x00,0x00,0x00,0xFF,0xFF,0x00,0x00,0x00,0x00,0x10,0x18,0x0E,0x07,0xFF,0xFF,0xFF,
0x07,0x87,0xE0,0x70,0x3F,0x0F,0x03,0x1F,0x3C,0xF0,0xC0,0x80,0x80,0x00,0x00,0x00,
0x00,0x00,0x00,0xFF,0xFF,0x40,0x40,0x40,0x40,0x7F,0x7F,0x40,0x40,0x40,0x40,0x40,
0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x01,0x01,0x00,0xFF,0xFF,0xC0,0x7F,0x3F,0x02,
0x7E,0xFC,0xBB,0xD9,0x63,0x3F,0x3E,0x7F,0xE1,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,
0x1F,0x1F,0x0F,0x04,0x04,0x04,0x7F,0xFF,0xFF,0x04,0x04,0x04,0x04,0x0F,0x0F,0xF0,
0xF0,0x80,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x80,0x80,0x80,0xFF,0xFF,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,0xFF,0x80,
0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x81,0x81,0x81,0x81,0x81,0x80,0x80,0x80,0x80,
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,
0x80,0x80,0x80,0x80,0x81,0x81,0x80,0x80,0x80,0x81,0x81,0x81,0x80,0x80,0x80,0x80,
0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,
0x80,0x80,0x81,0x81,0x81,0x81,0x81,0x81,0x81,0x80,0x80,0x80,0x80,0x80,0x80,0x80,
0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x81,0x81,0x80,0x80,0x80,0x80,0x80,0x80,0x80,
0x80,0x80,0x80,0x80,0x80,0x80,0x80,0xFF
};
//-----------------------------------------
void delayus(uint us)
{
  while(us!=0) us--;
}
//-----------------------------------------
void delayms(uint ms)
{
  uint m;
  while(ms-->0)
  for(m=0;m<80;m++)
  { };
}
//-----------------------------------------
void wcomd(uchar c)
{
  A0=0;RRW=0;P1=c;
}
//-----------------------------------------
void wdata(uchar d)
{
  A0=1;RRW=0;P1=d;
}
```

```
//-----------------------------------------
void initial()
{
  wcomd(0xaf);E1=0;E2=0;E1=1;E2=1; // Display On
  wcomd(0xc0);E1=0;E2=0;E1=1;E2=1; //Set Display Start Line
  wcomd(0xb8);E1=0;E2=0;E1=1;E2=1; //Set Page-Address
  wcomd(0x4f);E1=0;E2=0;E1=1;E2=1; //Set Column-Address
  wcomd(0xa0);E1=0;E2=0;E1=1;E2=1; //ADC select
  wcomd(0xa4);E1=0;E2=0;E1=1;E2=1; //Static drive off
  wcomd(0xa9);E1=0;E2=0;E1=1;E2=1; //1/32 duty
  wcomd(0xe0);E1=0;E2=0;E1=1;E2=1; //Read modify write mode end
  wcomd(0xee);E1=0;E2=0;E1=1;E2=1; //Release from the read modify write mode
" wcomd(0xe2);E1=0;E2=0;E1=1;E2=1; //Reset line,column,counter register to "0
}
//-----------------------------------------
void disp_all(uchar dat1,uchar dat2)
{
  uint i=0,j=0;
  uchar temp;
  temp=0xb8;
  wcomd(0xaf);E1=0;E2=0;E1=1;E2=1;
  for(j=0;j<4;j++)
  {
    wcomd(temp);E1=0;delayms(1);E1=1;
    wcomd(0x00);E1=0;delayms(1);E1=1;
    wcomd(0xc0);E1=0;delayms(1);E1=1;
    for(i=0;i<80;i++)
    {
      wdata(dat1);E1=0;E1=1;
      i++;
      wdata(dat2);E1=0;E1=1;
    }
    wcomd(temp);E2=0;delayms(1);E2=1;
    wcomd(0x00);E2=0;delayms(1);E2=1;
    wcomd(0xc0);E2=0;delayms(1);E2=1;
    for(i=0;i<80;i++)
    {
      wdata(dat2);E2=0;E2=1;
      i++;
      wdata(dat1);E2=0;E2=1;
    }
    temp++;
  }
}
//-----------------------------------------
void disp_bmp(xchar *str)
{
  uint i=0,j=0,k=0;
  uchar temp;
```

```
  temp=0xb8;
  wcomd(0xaf);E1=0;E1=1;
  for(k=0;k<4;k++)
  {
    wcomd(temp);E1=0;delayms(1);E1=1;
    wcomd(0x00);E1=0;delayms(1);E1=1;
    wcomd(0xc0);E1=0;delayms(1);E1=1;
    for(i=0;i<61;i++)
    {
      wdata(str[i+k*122]);E1=0;E1=1;
    }
    temp++;
  }
  temp=0xb8;
  wcomd(0xaf);E2=0;delayms(1);E2=1;
  for(k=0;k<4;k++)
  {
    wcomd(temp);E2=0;delayms(1);E2=1;
    wcomd(0x00);E2=0;delayms(1);E2=1;
    wcomd(0xc0);E2=0;delayms(1);E2=1;
    for(i=0;i<61;i++)
    {
      wdata(str[61+i+k*122]);E2=0;E2=1;
    }
    temp++;
  }
}
//-------------------------------------
///////////////  MAIN  ///////////////////////
////////////////////////////////////////////
void main(void)
{
  RST=0;delay(10);RST=1;delay(2);
  initial();
  while(1)
  {
    disp_bmp(xshm);       delay(150);
    disp_all (0xff,0xff); delay(100);
    disp_all (0x00,0x00); delay(100);
    disp_all (0x55,0xaa); delay(100);
    disp_all (0xaa,0x55); delay(100);
    disp_all (0x55,0x55); delay(100);
    disp_all (0xaa,0xaa); delay(100);
  }
}
```