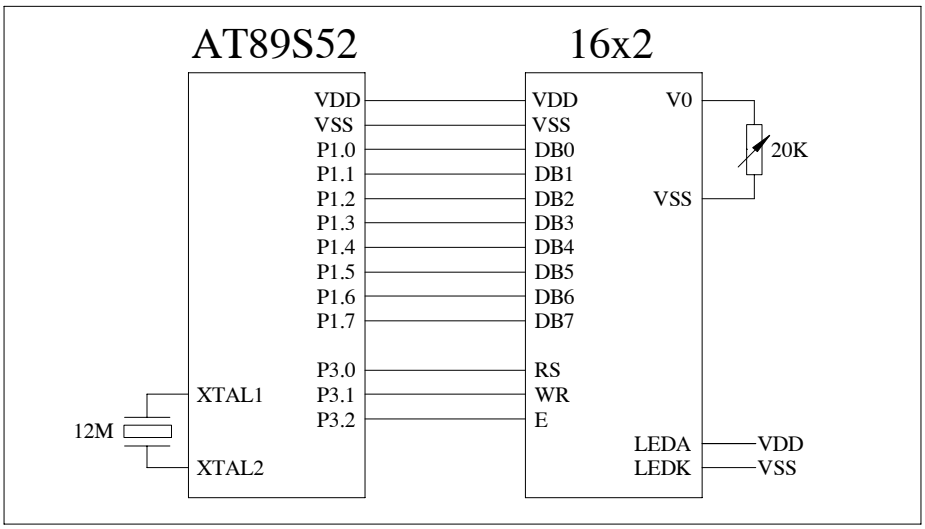
深圳秋田微电子有限公司
SHENZHEN AV-DISPLAY CO.,LTD.

Reference Program for LCD Modules	Version No.	A00
	Type	COB
Remark	16x2 character dot matrix series, with SPLC780D or compatible IC	

1. Interface



2. Instruction Code

Instruction Table

Instruction	Instruction Code										Description	Execution time (Temp = 25℃)		
	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		Fosc= 190KHz	Fosc= 270KHz	Fosc= 350KHz
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM and set DDRAM address to "00H" from AC	2.16ms	1.52ms	1.18ms
Return Home	0	0	0	0	0	0	0	0	1	-	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	2.16ms	1.52ms	1.18ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Assign cursor moving direction and enable the shift of entire display	53μs	38μs	29μs
Display ON/ OFF Control	0	0	0	0	0	0	1	D	C	B	Set display (D), cursor(C), and blinking of cursor(B) on/off control bit.	53μs	38μs	29μs
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	-	-	Set cursor moving and display shift control bit, and the direction, without changing of DDRAM data.	53μs	38μs	29μs

Function Set	0	0	0	0	1	DL	N	F	-	-	Set interface data length (DL: 8-bit/4-bit), numbers of display line (N: 2-line/1-line) and, display font type (F:5x10 dots/5x8 dots)	53μs	38μs	29μs
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.	53μs	38μs	29μs
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter	53μs	38μs	29μs
Read Busy Flag and Address Counter	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.			
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	53μs	38μs	29μs
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM).	53μs	38μs	29μs

Note1: "--": don't care

Note2: In the operation condition under -20℃ ~ 75℃, the maximum execution time for majority of instruction sets is 100us, except two instructions, "Clear Display" and "Return Home", in which maximum execution time can take up to 4.1ms.

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

//-------  16X2 字符液晶 8Bit IO 口定义------
sbit RS  = P3^2;
sbit RRW = P3^3;
sbit E   = P3^4;
sbit BF  = ACC^7;

xchar line1[]={" AV-DISPLAY LCM "};
xchar line2[]={"  CBC016002C07  "};
xchar line3[]={"abcdefghijklmnop"};
xchar line4[]={"ABCDEFGHIJKLMNOP"};
xchar line5[]={"XXXXXXXXXXXXXXXX"};
xchar line6[]={"XXXXXXXXXXXXXXXX"};
```

```
//-----------------------------------------
void delayus(uint us)
{
  while(us!=0) us--;
}
//-----------------------------------------
void delayms(uint ms)
{
  uint m;
  while(ms-->0)
  for(m=0;m<80;m++)
  { };
}
//-----------------------------------------
void delay(uint n)
{
  uint m;
  while(n-->0)
  for(m=0;m<1250;m++)
  { };
}
//-----------------------------------------
void busy()
{
  RS=0;RRW=1;
  do
  {
    P1=0xff;E=1;ACC=P1;E=0;
  }
  while(BF==1);
}
//-----------------------------------------
void wcomdn(uchar cdat)
{
  RRW=0;RS=0;
  P1=cdat;
  E=1;delayus(1);
  E=0;delayus(1);
}
//-----------------------------------------
void wcomd(uchar cdat)
{
  busy();
  RRW=0;RS=0;
  P1=cdat;
  E=1;delayus(1);
  E=0;delayus(1);
}
```

```
//-----------------------------------------
void wdata(uchar ddat)   /
{
  busy();
  RRW=0;RS=1;
  P1=ddat;
  E=1;delayus(1);
  E=0;delayus(1);
}
//-----------------------------------------
void initial()
{
//==== SPLC780D ====
  delayms(20);
  wcomdn(0x38);
  delayms(5);
  wcomdn(0x38);
  delayus(150);
  wcomdn(0x38);

  wcomd(0x38);
  delayus(50);
  wcomd(0x08);
  delayus(50);
  wcomd(0x01);
  delayms(20);
  wcomd(0x06);
  delayus(50);
  wcomd(0x0c);
  delayus(50);
}
//-----------------------------------------
void write_cgram(uchar cgdat1,uchar cgdat2)
{
  uchar ci;
  for(ci=0;ci<8;ci++)
  {
    wcomd(0x40+ci);
    wdata(cgdat1);
    ci++;
    wcomd(0x40+ci);
    wdata(cgdat2);
  }
}
//-----------------------------------------
void display_all(uchar adr,uchar su,uchar length)
{
  uchar i;
```

```
  wcomd(adr);
  for(i=0;i<length;i++)
  { wdata(su);}
  wcomd(adr+0x40);
  for(i=0;i<length;i++)
  { wdata(su);}
}
//-----------------------------------------
void display_str(uchar adr,xchar *str1,xchar *str2,uchar length)
{
  uchar i;
  wcomd(adr);
  for(i=0;i<length;i++)
  { wdata(str1[i]);}
  wcomd(adr+0x40);
  for(i=0;i<length;i++)
  { wdata(str2[i]);}
}
//-----------------------------------------
///////////////  MAIN  ////////////////////////
////////////////////////////////////////////////
void main()
{
  delay(10);
  initial();
  while(1)
  {
    display_str(0x80,line1,line2,16); delay(150);
    display_str(0x80,line3,line4,16); delay(150);
    display_str(0x80,line5,line6,16); delay(150);
    write_cgram(0x00,0x00);
    display_all(0x80,0x00,16); delay(100);
    write_cgram(0xff,0xff);
    display_all(0x80,0x00,16); delay(100);
    write_cgram(0xaa,0x55);
    display_all(0x80,0x00,16); delay(100);
    write_cgram(0x55,0xaa);
    display_all(0x80,0x00,16); delay(100);
    write_cgram(0x55,0x55);
    display_all(0x80,0x00,16); delay(100);
    write_cgram(0xaa,0xaa);
    display_all(0x80,0x00,16); delay(100);
    write_cgram(0xff,0x00);
    display_all(0x80,0x00,16); delay(100);
    write_cgram(0x00,0xff);
    display_all(0x80,0x00,16); delay(100);
  }
}
```