



Reference Program for LCD Modules

Version No.

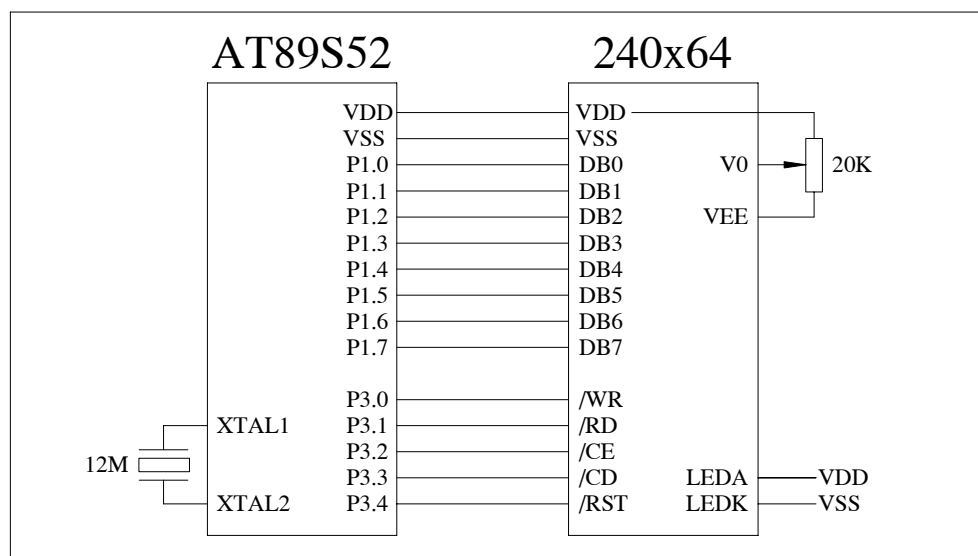
A00

Type

COB

Remark: 240x64 graphic dot matrix series, with T6963 or compatible IC

1. Interface



2. Instruction Code

COMMAND DEFINITIONS

COMMAND	CODE	D1	D2	FUNCTION
REGISTERS SETTING	00100001	X address	Y address	Set Cursor Pointer
	00100010	Data	00H	Set Offset Register
	00100100	Low address	High address	Set Address Pointer
SET CONTROL WORD	01000000	Low address	High address	Set Text Home Address
	01000001	Columns	00H	Set Text Area
	01000010	Low address	High address	Set Graphic Home Address
	01000011	Columns	00H	Set Graphic Area
MODE SET	1000X000	—	—	OR mode
	1000X001	—	—	EXOR mode
	1000X011	—	—	AND mode
	1000X100	—	—	Text Attribute mode
	10000XXX	—	—	Internal CG ROM mode
	10001XXX	—	—	External CG RAM mode
DISPLAY MODE	10010000	—	—	Display off
	1001XX10	—	—	Cursor on, blink off
	1001XX11	—	—	Cursor on, blink on
	100101XX	—	—	Text on, graphic off
	100110XX	—	—	Text off, graphic on
	100111XX	—	—	Text on, graphic on
CURSOR PATTERN SELECT	10100000	—	—	1-line cursor
	10100001	—	—	2-line cursor
	10100010	—	—	3-line cursor
	10100011	—	—	4-line cursor
	10100100	—	—	5-line cursor
	10100101	—	—	6-line cursor
	10100110	—	—	7-line cursor
	10100111	—	—	8-line cursor

COMMAND	CODE	D1	D2	FUNCTION
DATA AUTO READ / WRITE	10110000	—	—	Set Data Auto Write
	10110001	—	—	Set Data Auto Read
	10110010	—	—	Auto Reset
DATA READ / WRITE	11000000	Data	—	Data Write and Increment ADP
	11000001	—	—	Data Read and Increment ADP
	11000010	Data	—	Data Write and Decrement ADP
	11000011	—	—	Data Read and Decrement ADP
	11000100	Data	—	Data Write and Nonvariable ADP
	11000101	—	—	Data Read and Nonvariable ADP
SCREEN PEEK	11100000	—	—	Screen Peek
SCREEN COPY	11101000			Screen Copy
BIT SET / RESET	11110XXX	—	—	Bit Reset
	11111XXX	—	—	Bit Set
	1111X000	—	—	Bit 0 (LSB)
	1111X001	—	—	Bit 1
	1111X010	—	—	Bit 2
	1111X011	—	—	Bit 3
	1111X100	—	—	Bit 4
	1111X101	—	—	Bit 5
	1111X110	—	—	Bit 6
	1111X111	—	—	Bit 7 (MSB)

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

```
//===== 89S52 MCU, 12M Oscillator =====
```

[illegible]

[illegible]

[illegible]


```

}
//-----
void wdata2(d1,d2)
uint d1,d2;
{
    uint temdat1=0,temdat2=0,temp=0;
    temp=d1+30*d2;
    temdat1=temp%256;
    temdat2=temp/256;
    rbusy01();
    wdata1(temdat1);
    wdata1(temdat2);
}
//-----
void wcode(x,y,z)
uint x,y,z;
{
    wdata2(x,y);
    wcomd(z);
}
//-----
void initial()
{
    wcode(0x00,0x00,0x40);
    wcode(0x00,0x00,0x42);
    wcode(30,0x00,0x41);
    wcode(30,0x00,0x43);
    wcomd(0x80);
    wcode(0x02,0x00,0x22);
}
//-----
void disp_all(uint xx,uint yy)
{
    uint i=0,j=0;
    wcomd(0x98);
    wcode(0,0,0x24);
    wcomd(0xb0);
    for(j=0;j<32;j++)
    {
        for(i=0;i<30;i++)
        { wdata1(xx);}
        for(i=0;i<30;i++)
        { wdata1(yy);}
    }
    wcomd(0xb2);
}
//-----
void disp_bmp(uint gwzx,uint gwzy,xchar *str,uint gnum)
{

```

```

uint i=0, j=0;
wcomd(0x98);
wcode(gwzx, gwzy, 0x24);
wcomd(0xb0);
for(i=0; i<gnum; i++)
{ wdata1(str[i]);}
wcomd(0xb2);
}

//-----
////////// MAIN //////////
//////////

void main(void)
{
    RST=0;delayms(10);delay(5);
    RST=1;delay(5);
    initial();
    disp_all(0x00, 0x00);
    while(1)
    {
        disp_bmp(0, 0, xshm, 1920);delay(150);
        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);
        disp_all(0xaa, 0xaa);    delay(100);
        disp_all(0x55, 0x55);    delay(100);
    }
}

```