



Reference Program for LCD Modules

Version No.

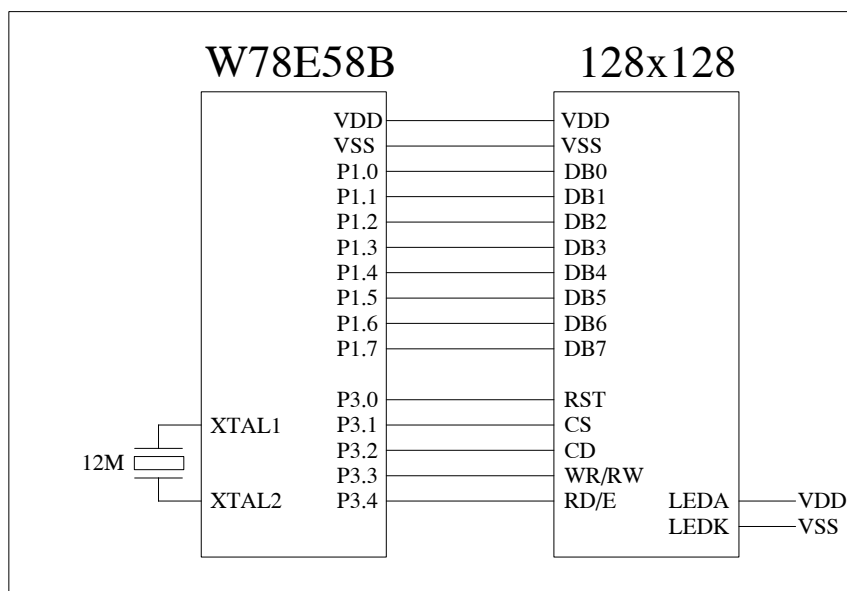
A00

Type

COG

Remark: 128x128 graphic dot matrix series, with UC1617 or compatible IC

1. Interface



2. Instruction Code

COMMAND SUMMARY

The following is a list of host commands supported by UC1617

C/D: 0: Control, 1: Data
W/R: 0: Write Cycle, 1: Read Cycle

Useful Data bits
- Don't Care

	Command	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0	Action	Default
1	Write Data Byte	1	0	#	#	#	#	#	#	#	#	Write 1 byte	N/A
2	Read Data Byte	1	1	#	#	#	#	#	#	#	#	Read 1 byte	N/A
3	Get Status	0	1	1	MX	MY	WA	DE	WS	MD	MS	Get {Status, Ver, PMO, Product Code, PID, MID}	N/A
				Ver	PMO[5:0]					Product Code		PID	MID
4	Set Page_C Address	0	0	0	0	0	#	#	#	#	#	Set CA[4:0]	0H
5	Set Temp. Compensation	0	0	0	0	1	0	0	1	#	#	Set TC[1:0]	00b
6	Set Panel Loading	0	0	0	0	1	0	1	0	#	#	Set PC[1:0]	10b
7	Set Pump Control	0	0	0	0	1	0	1	1	#	#	Set PC[3:2]	11b
8	Set Adv. Program Control (double-byte command)	0	0	0	0	1	1	0	0	R	R	Set APC[R][7:0], R = 0, 1 or 2	N/A
		0	0	#	#	#	#	#	#	#	#		
9	Set Scroll Line LSB	0	0	0	1	0	0	#	#	#	#	Set SL[3:0]	0H
	Set Scroll Line MSB	0	0	0	1	0	1	-	#	#	#	Set SL[6:4]	0H
10	Set Row Address LSB	0	0	0	1	1	0	#	#	#	#	Set RA[3:0]	00H
	Set Row Address MSB	0	0	0	1	1	1	-	#	#	#	Set RA[6:4]	00H
11	Set V _{Bias} Potentiometer (double-byte command)	0	0	1	0	0	0	0	0	0	1	Set PM[7:0]	4EH
		0	0	#	#	#	#	#	#	#	#		
12	Set Partial Display Control	0	0	1	0	0	0	0	1	#	#	Set LC[9:8]	00b: Disable
13	Set RAM Address Control	0	0	1	0	0	0	1	#	#	#	Set AC[2:0]	001b

14	Set Fixed Lines	0	0	1 #	0 #	0 #	1 #	0 #	0 #	0 #	0 #	Set {FLT, FLB}	0	
15	Set Line Rate	0	0	1	0	1	0	0	0	#	#	Set LC[4:3]	00b	
16	Set All-Pixel-ON	0	0	1	0	1	0	0	1	0	#	Set DC[1]	0b	
17	Set Inverse Display	0	0	1	0	1	0	0	1	1	#	Set DC[0]	0b	
18	Set Display Enable	0	0	1	0	1	0	1	1	#	#	Set DC[3:2]	10b	
19	Set LCD Mapping Control	0	0	1	1	0	0	0	#	#	#	Set LC[2:0]	000b	
20	Set N-Line Inversion	0	0	1	1	0	0	1	0	0	0	Set NIV[3:0]	6H	
21	Set LCD Gray Shade	0	0	1	1	0	1	0	#	#	#	Set LC[7:5]	001b	
22	System Reset	0	0	1	1	1	0	0	0	1	0	System Reset	N/A	
23	NOP	0	0	1	1	1	0	0	0	1	1	No operation	N/A	
24	Set Test Control (double-byte command)	0	0	1	1	1	0	0	1	TT		For testing only. Do not use.	N/A	
25	Set LCD Bias Ratio	0	0	1	1	1	0	1	0	#	#	Set BR[1:0]	11b: 11	
26	Reset Cursor Update Mode	0	0	1	1	1	0	1	1	1	0	AC[3]=0, CA=CR	AC[3]=0	
27	Set Cursor Update Mode	0	0	1	1	1	0	1	1	1	1	AC[3]=1, CR=CA	AC[3]=1	
28	Set COM End	0	0	1	1	1	1	0	0	0	1	Set CEN[6:0]	127	
29	Set Partial Display Start	0	0	1	1	1	1	0	0	1	0	Set DST[6:0]	0	
30	Set Partial Display End	0	0	1	1	1	1	0	0	1	1	Set DEN[6:0]	127	
31	Set Window Program Starting Page_C Address	0	0	1	1	1	1	0	1	0	0	Shared with MTP commands	Set WPC0	0
32	Set Window Programming Starting Row Address	0	0	1	1	1	1	0	1	0	1		Set WPP0	0
33	Set Window Programming Ending Page_C Address	0	0	1	1	1	1	0	1	1	0		Set WPC1	31
34	Set Window Programming Ending Row Address	0	0	1	1	1	1	0	1	1	1		Set WPP1	127
35	Enable window program	0	0	1	1	1	1	1	0	0	#	Set AC[4]	0: Disable	
36	Set MTP Operation control	0	0	1	0	1	1	1	0	0	0	Set MTPC[5:0]	10H	
37	Set MTP Write Mask	0	0	1	0	1	1	1	0	0	1	Set MTPM[7:0]	0	
38	Set V _{MTP1} Potentiometer	0	0	1	1	1	1	0	1	0	0	Shared with Window Program commands	Set MTP1	N/A
39	Set V _{MTP2} Potentiometer	0	0	1	1	1	1	0	1	0	1		Set MTP2	
40	Set MTP Write Timer	0	0	1	1	1	1	0	1	1	0		Set MTP3	
41	Set MTP Read Timer	0	0	1	1	1	1	0	1	1	1		Set MTP4	

Notes:

- Any bit patterns other than the commands listed above may result in undefined behavior.
- The interpretation of commands (37)~(41) depends on register MTPC[3].
- Commands (38)~(41) are shared with commands (31)~(34) and have exactly the same code. When MTPC[3]=0, commands (38)~(41) are interpreted as Window Programming commands. When MTPC[3]=1, they are the MTP Control commands.
- MTPM and PM are actually the same register. Only one of the commands (37 or 11) is valid at any time, and it is determined by MTPC[3].
- After MTP-ERASE or MTP-PROGRAM operation, before resuming normal operation, please always
 - Remove TST4 power source,
 - Do a full V_{DD} ON-OFF-ON cycle.

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  
  
//----- 80 -----  
sbit RST = P3^0;  
sbit CS  = P3^1;  
sbit CD  = P3^2;  
sbit WWR = P3^3;  
sbit RRD = P3^4;  
//----- 68 -----  
/*  
sbit RST = P3^0;  
sbit CS  = P3^1;  
sbit CD  = P3^2;  
sbit RW  = P3^3;  
sbit E   = P3^4;  
*/  
//-----  
void delayms(uint v)  
{ while(v!=0)v--;}  
//-----  
void delay(uint nn)  
{  
    uint mm;  
    while(nn-->0)  
        for(mm=0;mm<1000;mm++) { };  
}  
//===== 80 =====  
void wcomd(uchar cdat)  
{  
    RRD=1;  
    CD=0;  
    CS=0;  
    P1=cdat;  
    WWR=0;  
    _nop_();  
    WWR=1;  
    CS=1;  
}
```

```

//-----
void wdata(uchar ddat)
{
    RRD=1;
    CD=1;
    CS=0;
    P1=ddat;
    WWR=0;
    _nop_();
    WWR=1;
    CS=1;
}

//===== 68 =====
/*
void wcomd(uchar cdat)
{
    RW=0;
    CD=0;
    CS=0;
    P1=cdat;
    E=1;
    _nop_();
    E=0;
    CS=1;
}

//-----
void wdata(uchar ddat)
{
    RW=0;
    CD=1;
    CS=0;
    P1=ddat;
    E=1;
    _nop_();
    E=0;
    CS=1;
}

*/
//-----
void initial()
{
    wcomd(0xe2); // SET SYSTEM RESET
    delayms(100);
    wcomd(0x25); // SET TEMP=-0.1%
    wcomd(0xd0); // SET GRAY SHADE
    wcomd(0x28); // <7.2NF
    wcomd(0x2f); // INTERNAL PUMP
    wcomd(0x81); // SET VOP

```

```

wcomd(45);    //
wcomd(0x89); // SET RAM CONTROL
wcomd(0xa3); // SET LINE RATE
wcomd(0xc2); // SET LCD MAP CONTROL
wcomd(0xea); // SET BIAS=1/9
wcomd(0xf1); // CON
wcomd(127);
wcomd(0xAf); // SET DISPLAY ENABLE/4-GRAY
delayms(100);
}
//-----
void display_char1(uchar data0)
{
    int i;
    int j;
    for(i=0;i<128;i++)
    {
        for(j=0;j<32;j++)
        {
            wdata(data0);
        }
    }
}
//-----
void display_char2(uchar data1,uchar data2)
{
    int i;
    int j;
    for(i=0;i<64;i++)
    {
        for(j=0;j<32;j++)
        {
            wdata(data1);
        }
        for(j=0;j<32;j++)
        {
            wdata(data2);
        }
    }
}
//-----
////////// MAIN //////////////////////////////////////
void main(void)
{
    RST=0;delay(10);RST=1;delay(2);
    initial();
    while(1)
    {

```

```
display_char1(0xff);    delay(150);  
display_char1(0x00);    delay(150);  
display_char2(0xcc, 0x33);delay(150);  
display_char2(0x33, 0xcc);delay(150);  
display_char2(0xff, 0x00);delay(150);  
display_char2(0x00, 0xff);delay(150);  
}  
}
```