ORIENT DISPLAY
Your Total LCD Solution Provider
http://www.orientdisplay.com

SPECIFICATION
FOR
M3 TFT

MODULE NO: AFA480272B-4.3-A-TP
REVISION NO: 00

Customer's Approval:

	SIGNATURE	DATE
PREPARED BY (RD ENGINEER)		
CHECKED BY		
APPROVED BY		

DOCUMENT REVISION HISTORY

VERSION	DATE	DESCRIPTION	CHANGED BY
00	9-28-2010	First issue	Paul
01	12-20-2010	Add 16 data port	Paul

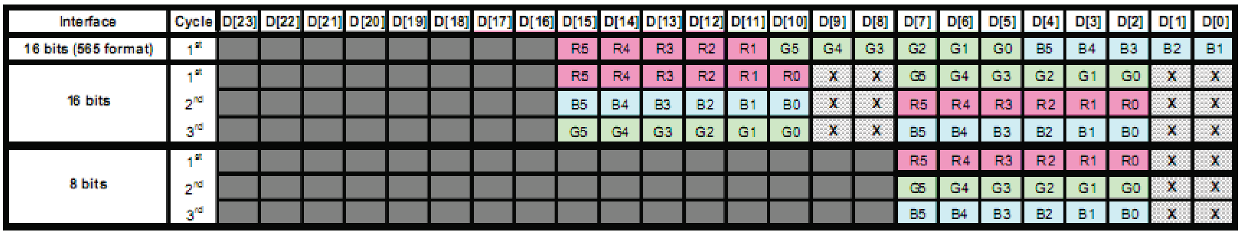
CONTENTS

Summary

This product is the TFT interface control module 8080

User can choose 8-bit data bus and 16-bit data bus work from different interfaces CON1, CON2

You can choose the color mode for the 565 format 16-bit color input (only support 16-bit data bus input), 24-bit color (8-bits, 16-bit data bus input), according to the initialization parameters of different configuration options (specific settings please refer to M7_Bus.h)



Interface	Cycle	D[23]	D[22]	D[21]	D[20]	D[19]	D[18]	D[17]	D[16]	D[15]	D[14]	D[13]	D[12]	D[11]	D[10]	D[9]	D[8]	D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]
16 bits (565 format)	1st									R5	R4	R3	R2	R1	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1
16 bits	1st									R5	R4	R3	R2	R1	R0	X	X	G5	G4	G3	G2	G1	G0	X	X
	2nd									B5	B4	B3	B2	B1	B0	X	X	R5	R4	R3	R2	R1	R0	X	X
	3rd									G5	G4	G3	G2	G1	G0	X	X	B5	B4	B3	B2	B1	B0	X	X
8 bits	1st																	R5	R4	R3	R2	R1	R0	X	X
	2nd																	G5	G4	G3	G2	G1	G0	X	X
	3rd																	B5	B4	B3	B2	B1	B0	X	X

1. Functions & Features

1.1. Format	: 480*272Dots
1.2. LCD mode	: Transmissive
1.3. Viewing direction	: 6 O'clock
1.4. Display color	: Up to 1670K color
1.5. Operation temp	: -20--70℃
1.6. Storage temp	: -30--80℃
1.7. Power supply voltage (VDD)	: 5.0V
1.9. Backlight color	: White(LED)
1.8 LCM Contrast ratio	: 500:1
1.9 LCM Birghtness	: 260cd/m^2(type)

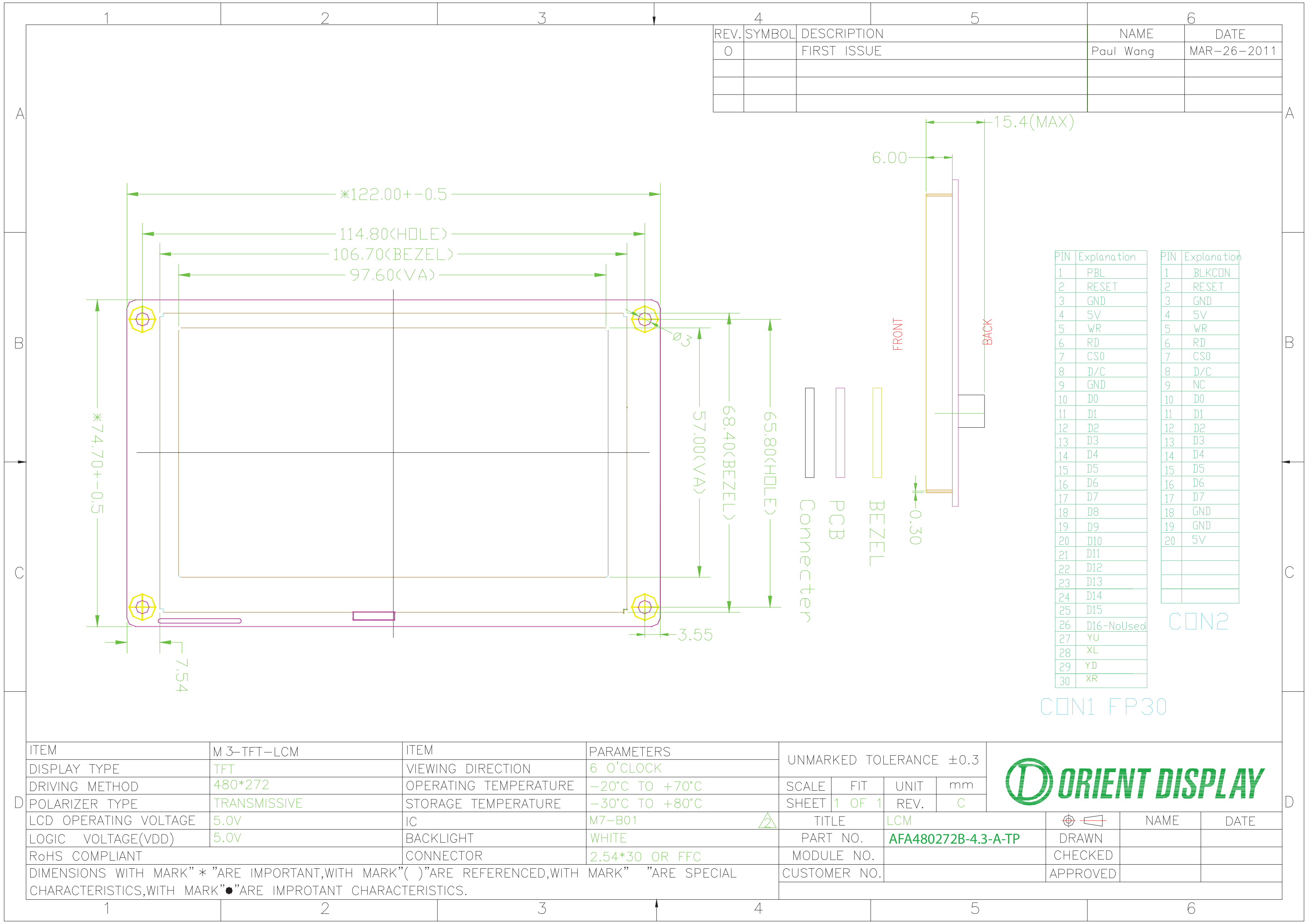
1.10.RoHS standard

1.11.LCM work in 8080 Mode

2. Mechanical Specifications

2.1. Module size	: 122.0mm(L)*75.0mm(W)*16.0max mm (H)
2.2. Viewing area	: 97.6mm(L)*57.0mm(W)
2.3. Dot pitch	: 0.198(W)*0.198(H) mm
2.4. Weight	: Approx.

3. Outline Drawing



REV.	SYMBOL	DESCRIPTION	NAME	DATE
0		FIRST ISSUE	Paul Wang	MAR-26-2011

CON1 FP30

PIN	Explanation
1	PBL
2	RESET
3	GND
4	5V
5	WR
6	RD
7	CS0
8	D/C
9	GND
10	D0
11	D1
12	D2
13	D3
14	D4
15	D5
16	D6
17	D7
18	D8
19	D9
20	D10
21	D11
22	D12
23	D13
24	D14
25	D15
26	D16-NoUsed
27	YU
28	XL
29	YD
30	XR

CON2

PIN	Explanation
1	BLKCON
2	RESET
3	GND
4	5V
5	WR
6	RD
7	CS0
8	D/C
9	NC
10	D0
11	D1
12	D2
13	D3
14	D4
15	D5
16	D6
17	D7
18	GND
19	GND
20	5V

ITEM	M3-TFT-LCM	ITEM	PARAMETERS
DISPLAY TYPE	TFT	VIEWING DIRECTION	6 O'CLOCK
DRIVING METHOD	480*272	OPERATING TEMPERATURE	-20°C TO +70°C
POLARIZER TYPE	TRANSMISSIVE	STORAGE TEMPERATURE	-30°C TO +80°C
LCD OPERATING VOLTAGE	5.0V	IC	M7-B01
LOGIC VOLTAGE(VDD)	5.0V	BACKLIGHT	WHITE
RoHS COMPLIANT		CONNECTOR	2.54*30 OR FFC

DIMENSIONS WITH MARK"*"ARE IMPORTANT,WITH MARK"( )"ARE REFERENCED,WITH MARK" "ARE SPECIAL CHARACTERISTICS,WITH MARK"●"ARE IMPROTANT CHARACTERISTICS.

UNMARKED TOLERANCE ±0.3

SCALE	FIT	UNIT	mm
SHEET	1 OF 1	REV.	C
TITLE	LCM		
PART NO.	AFA480272B-4.3-A-TP		
MODULE NO.			
CUSTOMER NO.			

	NAME	DATE
DRAWN		
CHECKED		
APPROVED		

4. PIN Description

(1) CON2

No.	Symbol	Function
1	PBL	BackLight Control.(H: BackLight On, L: BackLight Off)
2	RESET	RESET
3,18,19	GND	GND
4,20	5V	POWER Supply
5	/WR	TFT Write Signal(Low Active)
6	/RD	TFT Read Signal(Low Active)
7	/CS0	TFT Control Chip enable signal
8	D/C	TFT Data Or Command selection(H:Data L: Command)
9	NC	No USE
10-17	DB0-DB7	Data bus line (TFT)

(2) CON1 FP30

No.	Symbol	Function
1	PBL	BackLight Control.(H: BackLight On, L: BackLight Off)
2	RESET	RESET
3	GND	GND
4	5V	POWER Supply
5	/WR	TFT Write Signal(Low Active)
6	/RD	TFT Read Signal(Low Active)
7	/CS0	TFT Control Chip enable signal
8	D/C	TFT Data Or Command selection(H:Data L: Command)
9	GND	GND
10-26	DB0-DB16	The parallel MCU interface (TFT) (DB16 No Used

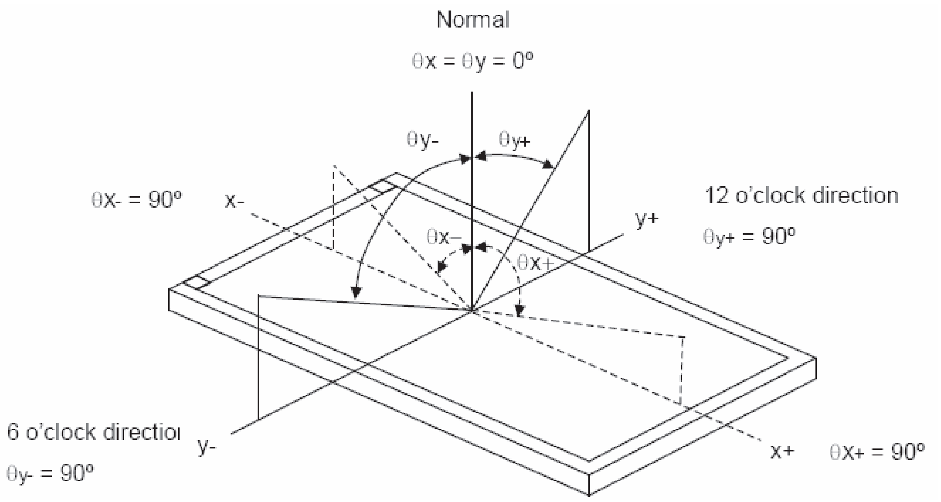
5. Electrical Characteristics

Item	Symbol	MIN	MAX	Unit
Supply Voltage for Logic	VDD	4.8	5.0	V
Supply Current		140	160	mA

6. Electro-optical Characteristics

(VDD=3.3V,Vled=5.0V, Ta = 25℃)

Item	Symbol	Condition	Min	Typ	Max	Unit
Viewing angle (CP 10)	0L	=180(9 o’clock)	60	70	---	
	0P 0T 0B	=00(3 o’clock)	60	70	---	degree
		=90(12 o’clock)	40	50	---	
		=270(6 o’clock)	60	70	---	
Pesponse time	Ton	Normal 0=00	---	10	20	ms
	Toff		---	15	30	ms
Contrast ratio	Cr		450	500	---	---
Luminance	L1		150	250		Cd/m2



7. Control Instruction

7.1 Instruction

Table : Instruction

/CS0	D/C	/WR	/RD	DB0=DB7
H	H/L	H/L	H/L	Command non-active
L	L	L	H	Write Data
L	H	L	H	Write Command (Used DB0=DB7)
L	L	H	L	Command non-active
L	H	H	L	Read Data

7.2 8080 Mode Write Cycle

Table : 8080 Mode Timing

Symbol	Parameter	Min	Typ	Max	Unit
t_{cyc}	Reference Clock Cycle Time	9	-	-	ns
t_{PWCSL}	Pulse width CS# low	1	-	-	t_{CYC}
t_{PWCSH}	Pulse width CS# high	1	-	-	t_{CYC}
t_{FDRD}	First Read Data Delay	5	-	-	t_{CYC}
t_{AS}	Address Setup Time	1	-	-	ns
t_{AH}	Address Hold Time	1	-	-	ns
t_{DSW}	Data Setup Time	4	-	-	ns
t_{DHW}	Data Hold Time	1	-	-	ns
t_{DSR}	Data Access Time	-	-	5	ns
t_{DHR}	Output Hold time	1	-	-	ns

8. Reliability of LCM

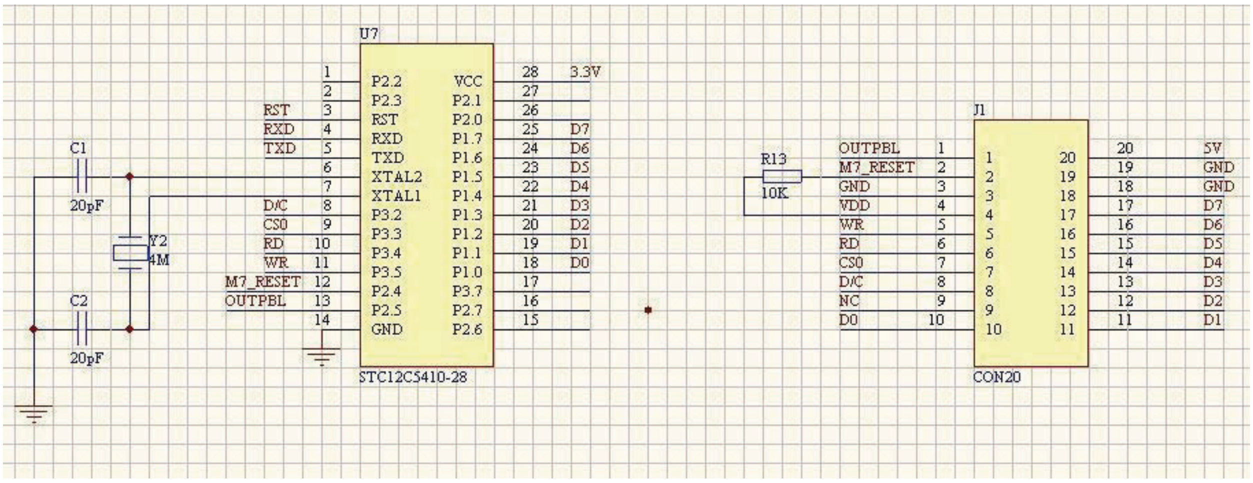
Reliability test condition:

Item	Condition	Time (hrs)	Assessment
High temp. Storage	80°C	48	No abnormalities in functions and appearance
High temp. Operating	70°C	48	
Low temp. Storage	-30°C	48	
Low temp. Operating	-20°C	48	
Humidity	40°C! 90%RH	48	
Temp. Cycle	-20°C — 25°C 70°C	10cycles	

Recovery time should be 24 hours minimum. Moreover, functions, performance and appearance ,etc. shall be free from remarkable deterioration within 50,000 hours under ordinary operating and storage conditions room temperature (20°C), normal humidity (20% RH), and in the area not exposed to direct sun light. The life time is not content the life time of the LED (for the life time of LED which decay only 50%,in the industry the experience value is 50000 hours, but there are notany experimentation data to support this). .

9. Application Example

Figure : APPLICATION CIRCUIT FOR M7-BUS



10. Program initialization Example

(1) M7_Bus.h File

```
/*****************************************************
// Defines the data bus width
*****************************************************/
//#define LCD_DATA_BUS_WIDTH 8                // Definition 8-bit data bus
#define LCD_DATA_BUS_WIDTH 16                     // Definition 16-bit data bus
//#define COLOR_MODULE 16 // Only support 16-bit bus
//#define COLOR_MODULE 24 // support 16-bit bus
/*****************************************************
Define data port
*****************************************************/
#define DATA_PORTL P1
#define DATA_PORTH P2


#define LCD_SCREEN 500
/*****************************************************
// Define the size of the LCD Resolution
*****************************************************/
#if LCD_SCREEN==320
    #define LCD_320
#elif LCD_SCREEN==430
    #define LCD_430
#elif LCD_SCREEN==500
    #define LCD_500
#elif LCD_SCREEN==560
    #define LCD_560
#elif LCD_SCREEN==700
    #define LCD_700
#else
    #define LCD_320
#endif
/*
    above all of that is config for LCD Screen
*/
void ClearScreen(uint color);
/*****************************************************
M7_BUS   IC /128PIN，Used to control read and write the way, if set 0, with 6800 bus MODE ,If set1,
for the 8080 bus mode. The following definition is 8080 mode

*****************************************************/

void M7_BUS_Instruction_Command_8(uchar cmd);
void M7_BUS_Instruction_Data_8(uchar sdata);
void M7_BUS_Data_Input(uint color);
void StartScroll(uint start);


void SetPixel(uint Col,uint Page,uint color); // Shows a phase factor point
uint GetPixel(uint Col,uint Page);// Get a pixel
void M7_BUSInit();

void GrayDisplay();
void ReadM7_BUSSet();
    // Io Config
    #define SetM7_BUSRD() SetBit(P3,4)
    #define ClrM7_BUSRD() ClrBit(P3,4)


    #define SetM7_BUSWR() SetBit(P3,5)
    #define ClrM7_BUSWR() ClrBit(P3,5)

    #define SetM7_BUSCS() SetBit(P3,3)
    #define ClrM7_BUSCS() ClrBit(P3,3)

    #define SetM7_BUSDC() SetBit(P3,2)
    #define ClrM7_BUSDC() ClrBit(P3,2)
    //HS,VS,PCLK and the dot clock pulse polarity     Setting
#ifdef LCD_350
    //LCD   Shift Clock Polarity - Set LCD shift clock active edge polarity, negative edge.
    #define CLKPOL                      (1<<2)
    //6.4MHZ*1024*1024 = 100MHZ * PCLK_VALUE
    #define PCLK_VALUE                      0x10624
    #define DISP_HOR_RESOLUTION       320L
    #define DISP_VER_RESOLUTION       240L


    #define DISP_HOR_PULSE_WIDTH           30
    #define DISP_HOR_BACK_PORCH            38
    #define DISP_HOR_FRONT_PORCH           8


    #define DISP_VER_PULSE_WIDTH           3
    #define DISP_VER_BACK_PORCH            15
    #define DISP_VER_FRONT_PORCH           4

    #define HT (DISP_HOR_RESOLUTION+DISP_HOR_PULSE_WIDTH+DISP_HOR_BACK_PORCH+DISP_HOR_FRONT_PORCH)
    #define HPS (DISP_HOR_PULSE_WIDTH+DISP_HOR_BACK_PORCH)
    #define                                                              VT
(DISP_VER_PULSE_WIDTH+DISP_VER_BACK_PORCH+DISP_VER_FRONT_PORCH+DISP_VER_RESO
LUTION)
    #define VSP (DISP_VER_PULSE_WIDTH+DISP_VER_BACK_PORCH)
#endif
#ifdef LCD_430
    //LCD    Shift Clock Polarity - Set LCD shift clock active edge polarity, negative edge
    #define CLKPOL                    (0<<2)
    //9 MHZ*1024*1024 = 100MHZ * PCLK_VALUE
    #define PCLK_VALUE                     0x170a3
    #define DISP_HOR_RESOLUTION       480
    #define DISP_VER_RESOLUTION       272


    #define DISP_HOR_PULSE_WIDTH          1
    #define DISP_HOR_BACK_PORCH                43
    #define DISP_HOR_FRONT_PORCH          8
    #define DISP_VER_PULSE_WIDTH          4
    #define DISP_VER_BACK_PORCH           10
    #define DISP_VER_FRONT_PORCH          2

    #define                                                              HT
(DISP_HOR_RESOLUTION+DISP_HOR_PULSE_WIDTH+DISP_HOR_BACK_PORCH+DISP_HOR_FRON
T_PORCH)
    #define HPS (DISP_HOR_PULSE_WIDTH+DISP_HOR_BACK_PORCH)
    #define                                                              VT
(DISP_VER_PULSE_WIDTH+DISP_VER_BACK_PORCH+DISP_VER_FRONT_PORCH+DISP_VER_RESO
LUTION)
    #define VSP (DISP_VER_PULSE_WIDTH+DISP_VER_BACK_PORCH)


#endif


#ifdef LCD_500
    //LCD    Shift Clock Polarity - Set LCD shift clock active edge polarity, the effective increase
in the extension.
    #define CLKPOL                    (1<<2)
    //25MHZ*1024*1024 = 100MHZ * PCLK_VALUE
    #define PCLK_VALUE                     0x40fb5
    #define DISP_HOR_RESOLUTION       640L
    #define DISP_VER_RESOLUTION       480L


    #define DISP_HOR_PULSE_WIDTH          10
    #define DISP_HOR_BACK_PORCH             134
    #define DISP_HOR_FRONT_PORCH        16


    #define DISP_VER_PULSE_WIDTH        2
    #define DISP_VER_BACK_PORCH         11
    #define DISP_VER_FRONT_PORCH        32

    #define                                                                      HT
(DISP_HOR_RESOLUTION+DISP_HOR_PULSE_WIDTH+DISP_HOR_BACK_PORCH+DISP_HOR_FRON
T_PORCH)
    #define HPS (DISP_HOR_PULSE_WIDTH+DISP_HOR_BACK_PORCH)
    #define                                                                      VT
(DISP_VER_PULSE_WIDTH+DISP_VER_BACK_PORCH+DISP_VER_FRONT_PORCH+DISP_VER_RESO
LUTION)
    #define VSP (DISP_VER_PULSE_WIDTH+DISP_VER_BACK_PORCH)
#endif

#ifdef LCD_560        //5.6 panel display
    //LCD  Shift Clock Polarity - Set LCD shift clock active edge polarity, the effective increase in
the extension.
.
    #define CLKPOL                      (1<<2)
    //25MHZ*1024*1024 = 100MHZ * PCLK_VALUE
    #define PCLK_VALUE                  0x40fb5
    #define DISP_HOR_RESOLUTION         640
    #define DISP_VER_RESOLUTION         480


    #define DISP_HOR_PULSE_WIDTH        10
    #define DISP_HOR_BACK_PORCH             134
    #define DISP_HOR_FRONT_PORCH        16
    #define DISP_VER_PULSE_WIDTH        2
    #define DISP_VER_BACK_PORCH         11
    #define DISP_VER_FRONT_PORCH        32

    #define                                                                      HT
(DISP_HOR_RESOLUTION+DISP_HOR_PULSE_WIDTH+DISP_HOR_BACK_PORCH+DISP_HOR_FRON
T_PORCH)
    #define HPS (DISP_HOR_PULSE_WIDTH+DISP_HOR_BACK_PORCH)
    #define                                                                      VT
(DISP_VER_PULSE_WIDTH+DISP_VER_BACK_PORCH+DISP_VER_FRONT_PORCH+DISP_VER_RESO
LUTION)
    #define VSP (DISP_VER_PULSE_WIDTH+DISP_VER_BACK_PORCH)
#endif
```

```
#ifdef LCD_700          //7.0'' PANEL
    //LCD    Shift Clock Polarity - Set LCD shift clock active edge polarity, the effective increase
in the extension.

    #define CLKPOL                         (1<<2)
    //pclk=26.2 mhz      30mHZ*1024*1024=100mhz *   PCLK_VALUE
    #define PCLK_VALUE                     0x430E0
    #define DISP_HOR_RESOLUTION        800
    #define DISP_VER_RESOLUTION        480


    #define DISP_HOR_PULSE_WIDTH           128
    #define DISP_HOR_BACK_PORCH                   88
    #define DISP_HOR_FRONT_PORCH           40


    #define DISP_VER_PULSE_WIDTH           3
    #define DISP_VER_BACK_PORCH            25
    #define DISP_VER_FRONT_PORCH           1

    #define                                                                            HT
(DISP_HOR_RESOLUTION+DISP_HOR_PULSE_WIDTH+DISP_HOR_BACK_PORCH+DISP_HOR_FRON
T_PORCH)
    #define HPS (DISP_HOR_PULSE_WIDTH+DISP_HOR_BACK_PORCH)
    #define                                                                            VT
(DISP_VER_PULSE_WIDTH+DISP_VER_BACK_PORCH+DISP_VER_FRONT_PORCH+DISP_VER_RESO
LUTION)
    #define VSP (DISP_VER_PULSE_WIDTH+DISP_VER_BACK_PORCH)
#endif
```

(2) M7_Bus.C File

```
#include "includes.h"

//   NOTE: In the write command to the M7-B01, only with a low 8-bit data lines to transmit data.
// LCD initialization must be 8-bit bus transfers data and commands
void M7_BUS_Instruction_Command_8(uchar cmd)
{

        ClrM7_BUSDC();
        SetM7_BUSRD();

        ClrM7_BUSCS();
        ClrM7_BUSWR();
        DATA_PORTL=cmd;
        SetM7_BUSWR();
        SetM7_BUSCS();
}
void M7_BUS_Instruction_Data_8(uchar sdata)
{
        SetM7_BUSDC();

        SetM7_BUSRD();

        ClrM7_BUSCS();
        ClrM7_BUSWR();
        DATA_PORTL=sdata;

        SetM7_BUSWR();
        SetM7_BUSCS();


}
void M7_BUS_Data_Input(uint color)
{
    static dataflag=0;
    //16-bit data bus，16-bit data bus 16-bit color
    #if   LCD_DATA_BUS_WIDTH==16
        DATA_PORTL=(uchar)color;
        DATA_PORTH=(uchar)(color>>8);
        ClrM7_BUSWR();
        SetM7_BUSWR();

    #elif LCD_DATA_BUS_WIDTH==8
        DATA_PORTL=(uchar)((color>>8)&0x00f8);

        ClrM7_BUSWR();
        SetM7_BUSWR();

        DATA_PORTL=(uchar)((color>>3)&0x00fc);

        ClrM7_BUSWR();
        SetM7_BUSWR();

        DATA_PORTL=(uchar)((color<<3) & 0x00f8);


        ClrM7_BUSWR();
        SetM7_BUSWR();
    #endif
}
void M7_BUSInit()
```

```
{

    M7_BUS_Instruction_Command_8(0x01);       //Reset
    Delay_Us(400);
    Delay_Us(400);

    M7_BUS_Instruction_Command_8(0x01);
    Delay_Us(400);
    Delay_Us(400);

    //enable pll
    M7_BUS_Instruction_Command_8(0xe0);
    Delay_Us(400);
    M7_BUS_Instruction_Data_8(0x01);
    Delay_Us(400);
    M7_BUS_Instruction_Command_8(0xe0);
    Delay_Us(400);
    M7_BUS_Instruction_Data_8(0x03);
    Delay_Us(400);

    //pll: n= 30 m= 3    100mhz
    M7_BUS_Instruction_Command_8(0xe2);
    M7_BUS_Instruction_Data_8(0x1D);        //N= 30 =0X1D+1
    M7_BUS_Instruction_Data_8(0x02);    //M =3        =0X02+1
    //Effectuate the multiplier and divider value
    M7_BUS_Instruction_Data_8(0x04);

    //dotclk setting
    M7_BUS_Instruction_Command_8(0xe6);
    M7_BUS_Instruction_Data_8(PCLK_VALUE>>16);
    M7_BUS_Instruction_Data_8(PCLK_VALUE>>8);
    M7_BUS_Instruction_Data_8(PCLK_VALUE); //

    //panel setting
    M7_BUS_Instruction_Command_8(0xB0);
    // set 18-bit for 7" panel TY700TFT800480
    M7_BUS_Instruction_Data_8(0x38|CLKPOL);//0x10


    M7_BUS_Instruction_Data_8(0x00);                              // set TTL mode
    // Set LCD panel size(DISP_HOR_RESOLUTION = 800) H
    M7_BUS_Instruction_Data_8((DISP_HOR_RESOLUTION-1)>>8);
    // Set LCD panel size(DISP_HOR_RESOLUTION = 800) L
    M7_BUS_Instruction_Data_8(DISP_HOR_RESOLUTION-1);
    // Set LCD panel size(DISP_VER_RESOLUTION = 480) H
```

```
M7_BUS_Instruction_Data_8((DISP_VER_RESOLUTION-1)>>8);
// Set LCD panel size(DISP_VER_RESOLUTION = 480) L
M7_BUS_Instruction_Data_8(DISP_VER_RESOLUTION-1);
M7_BUS_Instruction_Data_8(0x00);//0x00                    // RGB sequence

//pixel format
M7_BUS_Instruction_Command_8(0x3a);
M7_BUS_Instruction_Data_8(0x50);

//hsync setting
M7_BUS_Instruction_Command_8(0xB4);
M7_BUS_Instruction_Data_8((HT-1)>>8);
M7_BUS_Instruction_Data_8(HT-1);
M7_BUS_Instruction_Data_8((HPS-1)>>8);
M7_BUS_Instruction_Data_8(HPS-1);
M7_BUS_Instruction_Data_8(DISP_HOR_PULSE_WIDTH-1);
M7_BUS_Instruction_Data_8(0x00);
M7_BUS_Instruction_Data_8(0x00);
M7_BUS_Instruction_Data_8(0x00);


//vsync setting
M7_BUS_Instruction_Command_8(0xB6);
M7_BUS_Instruction_Data_8((VT-1)>>8);
M7_BUS_Instruction_Data_8(VT-1);
M7_BUS_Instruction_Data_8((VSP-1)>>8);
M7_BUS_Instruction_Data_8(VSP-1);
M7_BUS_Instruction_Data_8(DISP_VER_PULSE_WIDTH-1);
M7_BUS_Instruction_Data_8(0x00);
M7_BUS_Instruction_Data_8(0x00);

//col start_end
M7_BUS_Instruction_Command_8(0x2a);
M7_BUS_Instruction_Data_8(0x00);
M7_BUS_Instruction_Data_8(0x00);
// Set LCD panel size(DISP_HOR_RESOLUTION = 800) H
M7_BUS_Instruction_Data_8((DISP_HOR_RESOLUTION-1)>>8);
// Set LCD panel size(DISP_HOR_RESOLUTION = 800) L
M7_BUS_Instruction_Data_8(DISP_HOR_RESOLUTION-1);
//row start_end
M7_BUS_Instruction_Command_8(0x2b);
M7_BUS_Instruction_Data_8(0x00);
M7_BUS_Instruction_Data_8(0x00);
// Set LCD panel size(DISP_VER_RESOLUTION = 480) H
M7_BUS_Instruction_Data_8((DISP_VER_RESOLUTION-1)>>8);
// Set LCD panel size(DISP_VER_RESOLUTION = 480) L
```

```
    M7_BUS_Instruction_Data_8(DISP_VER_RESOLUTION-1);


    M7_BUS_Instruction_Command_8(0x36);
    M7_BUS_Instruction_Data_8(0x00);

    // SET    M7-BUS IC    8bit Set pixel data interface
    M7_BUS_Instruction_Command_8(0xf0);

    //16-bit data bus as RGB 565 16-bit data transfer
    #if LCD_DATA_BUS_WIDTH==16
        M7_BUS_Instruction_Data_8(0x03);     //03               00 8 bit i     03 16bit

    #endif
    #if LCD_DATA_BUS_WIDTH==8     //8bit bus

        M7_BUS_Instruction_Data_8(0x00);     //03               00 8biti     03 16bit

    #endif
    //display on
    M7_BUS_Instruction_Command_8(0x29);
}
void StartScroll(uint start)
{
    M7_BUS_Instruction_Command_8(0x37);
    M7_BUS_Instruction_Data_8(start>>8);
    M7_BUS_Instruction_Data_8(start);
}
void ClearScreen(uint color)
{
    long i ,j;
    /*****************************************************
    Set the size of the display screen
    *****************************************************/
    M7_BUS_Instruction_Command_8(0x2a);
    M7_BUS_Instruction_Data_8(0x00);
    M7_BUS_Instruction_Data_8(0x00);
    M7_BUS_Instruction_Data_8(((DISP_HOR_RESOLUTION-1)>>8)&0xff);
    M7_BUS_Instruction_Data_8((DISP_HOR_RESOLUTION-1)&0xff);//320
    //row start_end
    M7_BUS_Instruction_Command_8(0x2b);
    M7_BUS_Instruction_Data_8(0x00);
    M7_BUS_Instruction_Data_8(0x00);
    M7_BUS_Instruction_Data_8(((DISP_VER_RESOLUTION-1)>>8)&0xff);
    M7_BUS_Instruction_Data_8((DISP_VER_RESOLUTION-1)&0xff);
    M7_BUS_Instruction_Command_8(0x2c);
```

```
        SetM7_BUSDC();
        SetM7_BUSRD();
        ClrM7_BUSCS();
        for (i=0;i<DISP_HOR_RESOLUTION;i++)
        {
                for (j=0;j<DISP_VER_RESOLUTION;j++)
                {
                        M7_BUS_Data_Input(color);

                }
        }
        SetM7_BUSCS();
}
// Shows part of the picture
void ClearPartColor(uint color,uint startx,uint starty,uint endx,uint endy)
{
        uint i,j;
        M7_BUS_Instruction_Command_8(0x2a);
        M7_BUS_Instruction_Data_8(startx/256);
        M7_BUS_Instruction_Data_8(startx%256);

        M7_BUS_Instruction_Data_8((endx-1)/256);
        M7_BUS_Instruction_Data_8((endx-1)%256);//480

        M7_BUS_Instruction_Command_8(0x2b);
        M7_BUS_Instruction_Data_8(starty/256);
        M7_BUS_Instruction_Data_8(starty%256);

        M7_BUS_Instruction_Data_8((endy-1)/256);
        M7_BUS_Instruction_Data_8((endy-1)%256);//272

        M7_BUS_Instruction_Command_8(0x2c);


        SetM7_BUSDC();
        SetM7_BUSRD();
        ClrM7_BUSCS();

        for(i=starty;i<endy;i++)
        {
                for(j=startx;j<endx;j++)
                {
                        M7_BUS_Data_Input(color);
                }
        }
        SetM7_BUSCS();

}
uint returncolor(uchar R,uchar G,uchar B)
{
        uint R5,G6,B5 ;
R= R&0xf8; //565rgb
        G= G&0xfC; //565rgb
        B= B&0xf8; //565rgb

        R5= R>>3;
        G6= G>>2;
        B5= B>>3;
        return (R5<<11 )|(G6<<5 )| B5;
}
void GrayDisplay()            // Gradation display
{
        uint x,color;
        uchar r,g,b,i;
        x=0;
        r=0;
        g=0;
        b=0;
        for(i=0;i<8;i++)
        {

                color=returncolor(r,g,b);
                ClearPartColor(color,x,0,x+55,136);
                x+=55;
                ClearPartColor(0x0000,x,0,x+5,136);
                x+=5;
                if(r==0)
                {
                        r=0x1f;
                        g=0x1f;
                        b=0x1f;
                }
                else
                {
                        r=r+0x10;
                        g=g+0x10;
                        b=b+0x10;
                }
        }
        x=0;
```

```
    for(i=0;i<8;i++)
    {

        color=returncolor(r,g,b);
        ClearPartColor(color,x,136,x+55,272);
        x+=55;
        ClearPartColor(0x0000,x,136,x+5,272);
        x+=5;

        r=r+0x10;
        g=g+0x10;
        b=b+0x10;
    }
}
```

11.Packing Information

TBD