

GMT130-V1.0

1.3 inch TFT Module 240×240 ST7789



1.3-Inch TFT Module (Model GMT130-V1.0): Experience vibrant visuals with our 1.3-inch TFT LCD display module. This full-color screen boasts a resolution of 240×240 pixels, displaying a stunning palette of 262K colors. The module's compact outline dimensions of 27.5×39.0×11.3 mm house an active area of 23.4×23.4 mm, with pixel sizes of 0.0975×0.0975 mm. It's powered by the reliable ST7789 driver IC and interfaces seamlessly through 4-wire SPI. This versatile module operates in temperatures ranging from -20°C to 70°C, with a storage temperature range of -30°C to 80°C. With just 7 pins for assembly through soldering, it's a powerful and compact solution for your display needs.

Vivid Visuals, Compact Design: Our GMT130-V1.0 TFT module delivers vivid full-color graphics and sharp details on its 1.3-inch screen. Its compact outline dimensions make it ideal for space-constrained applications. With a resolution of 240×240 pixels and a color palette of 262K, it ensures your content pops with vibrant colors and precision. The module, driven by the ST7789 IC, interfaces through a 4-wire SPI connection for seamless integration into your projects.

Whether you're crafting wearables, handheld devices, or prototypes, this module thrives in various environments, from -20°C to 70°C.

Robust Performance, Easy Integration: With its 7-pin assembly through soldering, the GMT130-V1.0 TFT module offers reliable performance and straightforward integration. Its pixel-perfect 240×240 resolution, 262K color depth, and ST7789 driver IC combine to deliver exceptional visuals. This compact display operates effortlessly in temperatures ranging from -20°C to 70°C, with a wide storage temperature range of -30°C to 80°C, ensuring durability in various conditions. Its 4-wire SPI interface simplifies connectivity, making it an excellent choice for a wide range of applications where compact design and vibrant visuals are essential.

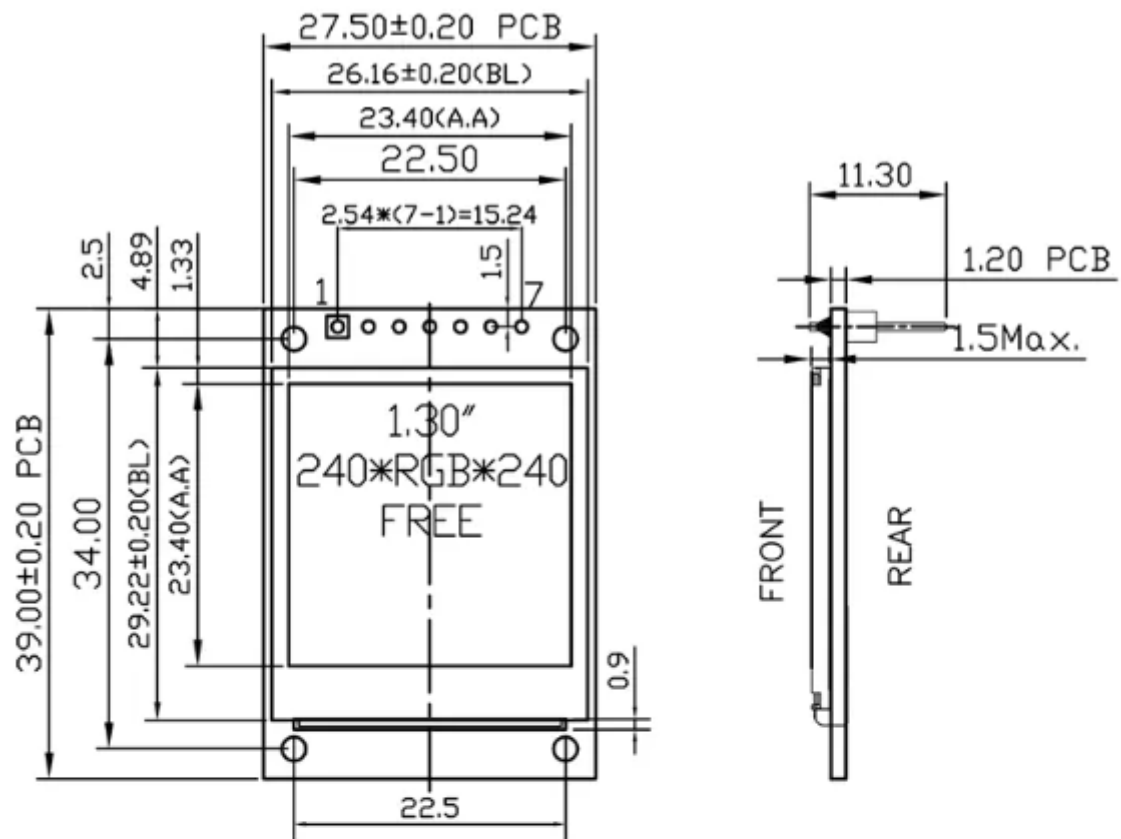
Product parameter

Diagonal Screen Size:	1.3 inch tft 13 lcd display screen module
Color type:	Full Color
Model number:	GMT130-V1.0
Resolutions:	240xRGBx240
Color:	262K
Outline dimension(mm):	27.50x39.00x11.30
Active area(mm):	23.40x23.40
Pixel size(mm):	0.0975x0.0975
Driver IC:	ST7789
Interface:	4-wire SPI
Operating temperature:	-20°C-70°C
Storage Temperature:	-30°C-80°C
Pins:	7Pins
Assemble:	Soldering

Module Connections

Connection to the display is via a 7-pin header.

1. GND Ground of Logic Circuit
2. VCC Power Supply for Logic
3. SCK Serial clock input.
4. SDA Serial data input.
5. RES Reset
6. DC: Data/Command
7. BLK: Backlight control switch



Pin Description

Pin No.	Symbol
1	GND
2	VCC
3	SCK
4	SDA
5	RES
6	DC
7	BLK

Install Arduino ST7789 GMT130-V1.0.Test Program

```
#include<reg51.h>

#include<absacc.h>

#include<intrins.h>

#include<string.h>

#include<font.h>

#define uchar unsigned char

#define uint unsigned int

#ifdef MCU_STC12

sfr P3M1 = 0xB1;

sfr P3M0 = 0xB2;

#endif

sbit BL = P3^5;

sbit SCL = P3^0;

sbit SDA = P3^1;

sbit RS = P3^3;

sbit reset = P3^2;

#define RED 0xf800

#define GREEN 0x07e0

#define BLUE 0x001f

#define WHITE 0xffff
```

```

#define BLACK          0x0000

#define YELLOW 0xFFE0

#define GRAY0  0xEF7D

#define GRAY1  0x8410

#define GRAY2  0x4208

void delay_ms(uint time)

    {

        uint i,j;

        for(i=0;i<time;i++)

            for(j=0;j<25;j++);

    }

void SPI_WriteData(uchar Data)

    {

        unsigned char i=0;

        for(i=8;i>0;i--)

            {

                if(Data&0x80)

                    SDA=1;

                else SDA=0;

                SCL=0;

                SCL=1;

                Data<<=1;

            }

    }

void Lcd_WriteIndex(uchar Data)

```

```
    {  
  
        RS=0;  
        SPI_WriteData(Data);  
  
    }  
  
void Lcd_WriteData(uchar Data)  
  
    {  
  
        RS=1;  
        SPI_WriteData(Data);  
  
    }  
  
void LCD_WriteData_16Bit(unsigned int Data)  
  
    {  
  
        RS=1;  
        SPI_WriteData(Data>>8);  
        SPI_WriteData(Data);  
  
    }  
  
void Reset()  
  
    {
```

```
        reset=0;

        delay_ms(10);

        reset=1;

        delay_ms(10);

    }

void lcd_initial()

{
    Reset();

    delay_ms(12);

    Lcd_WriteIndex(0x36);

    Lcd_WriteData(0x00);

    Lcd_WriteIndex(0x3A);

    Lcd_WriteData(0x03);

    Lcd_WriteIndex(0xB2);

    Lcd_WriteData(0x0C);

    Lcd_WriteData(0x0C);

    Lcd_WriteData(0x00);

    Lcd_WriteData(0x33);

    Lcd_WriteData(0x33);

    Lcd_WriteIndex(0xB7);

    Lcd_WriteData(0x35);

    Lcd_WriteIndex(0xBB);

    Lcd_WriteData(0x19);

    Lcd_WriteIndex(0xC0);

    Lcd_WriteData(0x2C);

    Lcd_WriteIndex(0xC2);
```



```
Lcd_WriteData(0x01);  
  
Lcd_WriteIndex(0xC3);  
  
Lcd_WriteData(0x12);  
  
Lcd_WriteIndex(0xC4);  
  
Lcd_WriteData(0x20);  
  
Lcd_WriteIndex(0xC6);  
  
Lcd_WriteData(0x0F);  
  
Lcd_WriteIndex(0xD0);  
  
Lcd_WriteData(0xA4);  
  
Lcd_WriteData(0xA1);  
  
Lcd_WriteIndex(0xE0);  
  
Lcd_WriteData(0xD0);  
  
Lcd_WriteData(0x04);  
  
Lcd_WriteData(0x0D);  
  
Lcd_WriteData(0x11);  
  
Lcd_WriteData(0x13);  
  
Lcd_WriteData(0x2B);  
  
Lcd_WriteData(0x3F);  
  
Lcd_WriteData(0x54);  
  
Lcd_WriteData(0x4C);  
  
Lcd_WriteData(0x18);  
  
Lcd_WriteData(0x0D);  
  
Lcd_WriteData(0x0B);  
  
Lcd_WriteData(0x1F);  
  
Lcd_WriteData(0x23);  
  
Lcd_WriteIndex(0xE1);
```

```
Lcd_WriteData(0xD0);  
  
Lcd_WriteData(0x04);  
  
Lcd_WriteData(0x0C);  
  
Lcd_WriteData(0x11);  
  
Lcd_WriteData(0x13);  
  
Lcd_WriteData(0x2C);  
  
Lcd_WriteData(0x3F);  
  
Lcd_WriteData(0x44);  
  
Lcd_WriteData(0x51);  
  
Lcd_WriteData(0x2F);  
  
Lcd_WriteData(0x1F);  
  
Lcd_WriteData(0x1F);  
  
Lcd_WriteData(0x20);  
  
Lcd_WriteData(0x23);  
  
Lcd_WriteIndex(0x21);  
  
Lcd_WriteIndex(0x11);  
  
Lcd_WriteIndex(0x29);  
  
    }
```

```
void Lcd_SetRegion(unsigned int x_start,unsigned int  
    y_start,unsigned int x_end,unsigned int y_end)
```

```
{  
  
    Lcd_WriteIndex(0x2a);  
  
    Lcd_WriteData(0x00);  
  
    Lcd_WriteData(x_start);  
  
    Lcd_WriteData(0x00);
```

```
Lcd_WriteData(x_end);  
  
Lcd_WriteIndex(0x2b);  
  
Lcd_WriteData(0x00);  
  
Lcd_WriteData(y_start);  
  
Lcd_WriteData(0x00);  
  
Lcd_WriteData(y_end);  
  
Lcd_WriteIndex(0x2c);  
  
}
```

```
void dsp_12colour(char color)  
  
{  
  
    uchar i,j;  
  
Lcd_SetRegion(0,0,240-1,240-1);  
  
    RS=1;  
  
    for (i=0;i<240;i++)  
  
        for (j=0;j<120;j++)  
  
            {  
  
                SPI_WriteData(color);  
  
  
                SPI_WriteData(color);  
  
  
                SPI_WriteData(color);  
  
            }  
  
}
```

```
void Fast_DrawFont_GBK16(uint x, uint y, uint fc, uint bc, uchar *s)  
  
{
```

```

        unsigned char i,j;

        unsigned short k;

        uint HZnum;

        HZnum=sizeof(hz16)/sizeof(typFNT_GBK16);

        while(*s)

            {

                if((*s) >= 128)

                    {

                        for (k=0;k<HZnum;k++)

                            {

                                if

                                ((hz16[k].Index[0]==*(s))&&(hz16[k].Index[1]==*(s+1)))

                                    {

                                        Lcd_SetRegion(x,y,x+16-1,y+16-1);

                                        for(i=0;i<16*2;i++)

                                            {

                                                for(j=0;j<8;j++)

                                                    {

                                                        if(hz16[k].Msk[i]&(0x80>>j))

                                                            LCD_WriteData_16Bit(fc);

                                                        else

                                                            {

                                                                if (fc!=bc) LCD_WriteData_16Bit(bc);

                                                            }

                                                    }

                                                }

                                            }

                                        }

                            }

                    }

            }

```

```

        }
    }
    s+=2;x+=16;
}

else
    s+=1;

}
}

void Font_Test(void)
{
    Lcd_WriteIndex(0x3A);

    Lcd_WriteData(0x05);

    Fast_DrawFont_GBK16(8,10,BLUE,YELLOW, "Testing");

    Fast_DrawFont_GBK16(0,30,RED,WHITE, "Goldenmorning
        Electronic");

    Fast_DrawFont_GBK16(16,50,BLUE,WHITE, "Thank you");

    Fast_DrawFont_GBK16(8,70,WHITE,BLUE, "For");

    Fast_DrawFont_GBK16(8,90,WHITE,BLUE, "Your Order");

    delay_ms(10);

}

void dsp_single_colour(int color)
{

```

```

        uchar i,j;

Lcd_SetRegion(0,0,240-1,240-1);

        for (i=0;i<240;i++)

        for (j=0;j<240;j++)

        LCD_WriteData_16Bit(color);

        }

        main()

        {

        #ifndef MCU_STC12

P3M1 &= ~(1<<2), P3M0 |= (1<<2);

        #endif

        lcd_initial();

        BL=1;

        while(1)

        {

        dsp_12colour(0xff);

        delay_ms(10);

        dsp_12colour(0x00);

        delay_ms(10);

        Font_Test();

        delay_ms(10);

        dsp_single_colour(RED);

        delay_ms(10);

        dsp_single_colour(GREEN);

        delay_ms(10);

        dsp_single_colour(BLUE);

```

```
        delay_ms(10);  
    Lcd_WriteIndex(0x3A);  
    Lcd_WriteData(0x03);  
    }  
}
```