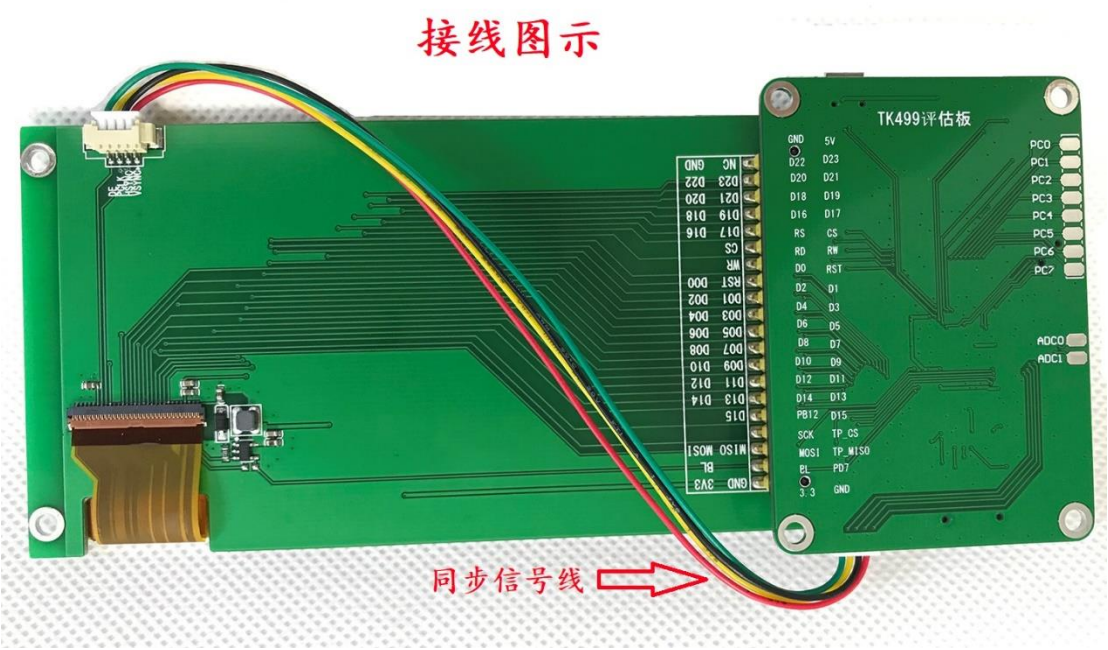
TK043F3211 液晶屏应用说明

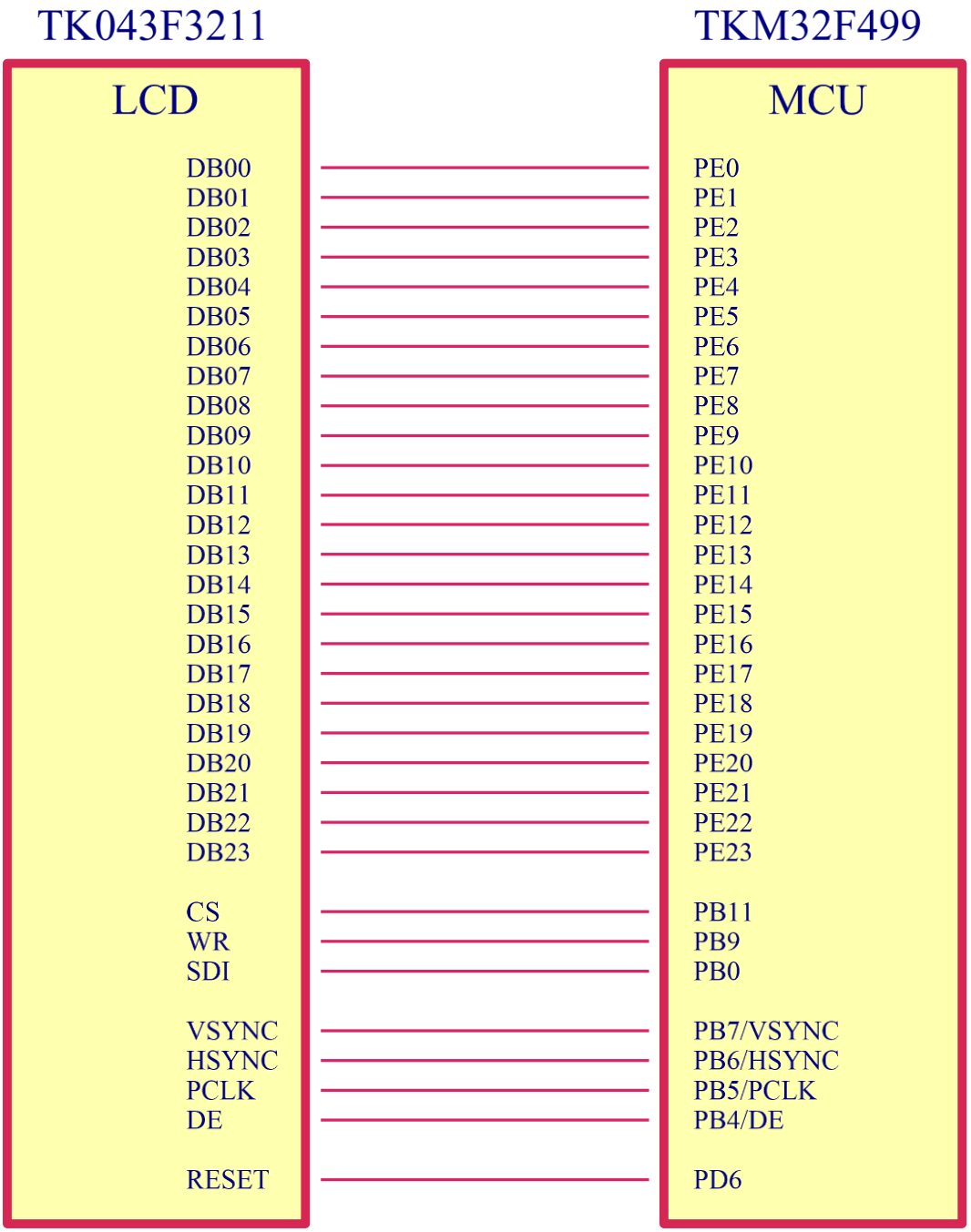


接线方式如上图，注意这根同步信号线比较长，仅是用于演示，产品上用建议尽量缩短这个同步线，最好主控芯片做在转接板上。如果演示过程中有点花屏，请看看这根同步信号线是不是拧在一起了，分散一点就行。

此屏支持最高 RGB888 模式，也就是 1600 万色，可以向下兼容 RGB666 及 RGB565。向下兼容 RGB666 或者 565 时，有几种方法，最简单的是把每组颜色的低位丢弃，高位对齐；高级一点的方式是除高位对齐外，把低位颜色并联到高位上。更加高级的是，通过初始化配置，直接把液晶屏配置成 RGB666 或者 RGB565。

液晶屏初始化默认是 RGB888 模式。

主要信号接线



转接板上的 BL_CTL 是背光控制引脚，高电平控制背光亮。

程序初始化流程

先用 3 个 IO 口模拟 SPI 初始化液晶屏，在程序中，CS= LCD_SPI_CS；WR= SPI_DCLK；SDI= SPI_SDA；下面是模拟 SPI 的三个根函数。

```
void LCD_WriteByteSPI(unsigned char byte)
{
    unsigned char n;
    for(n=0; n<8; n++)
    {
        if(byte&0x80) SPI_SDA(1)
        else SPI_SDA(0)
        byte<<= 1;
        SPI_DCLK(0);
        SPI_DCLK(1);
    }
}
void SPI_WriteComm(u16 CMD)//3 线 8bit  串行接口
{
    LCD_SPI_CS(0);
    LCD_WriteByteSPI(0x70);
    LCD_WriteByteSPI(CMD);
    LCD_SPI_CS(1);
}
void SPI_WriteData(u16 tem_data)
{
    LCD_SPI_CS(0);
    LCD_WriteByteSPI(0x72);
    LCD_WriteByteSPI(tem_data);
    LCD_SPI_CS(1);
}
```

然后把初始化写入液晶屏就可以初始化液晶屏

```
void TK043F3211_init_code(void)
{
    // ================= 4.3 inch  金属框
==================== //
  SPI_WriteComm(0xc0);//Oscillator frequency control
  SPI_WriteData(0x00);//use PLCK as system clock
  SPI_WriteData(0x00);

  SPI_WriteComm(0x20);
  SPI_WriteComm(0x35);
  SPI_WriteData(0x00);
  SPI_WriteData(0xC8);
```

```
SPI_WriteComm(0xb1);
SPI_WriteData(0x06);
SPI_WriteData(0x0F);
SPI_WriteData(0x0F);

SPI_WriteComm(0xb2);
SPI_WriteData(0x00);
SPI_WriteData(0xc8);

SPI_WriteComm(0xb3);
SPI_WriteData(0x01);

SPI_WriteComm(0xb4);
SPI_WriteData(0x04);

SPI_WriteComm(0xb5);
SPI_WriteData(0x10);
SPI_WriteData(0x30);
SPI_WriteData(0x30);
SPI_WriteData(0x00);
SPI_WriteData(0x00);

SPI_WriteComm(0xb6);    //
SPI_WriteData(0x0b);   //0b
SPI_WriteData(0x0f);
SPI_WriteData(0x3c);
SPI_WriteData(0x13);
SPI_WriteData(0x13);
SPI_WriteData(0xe8);

SPI_WriteComm(0xb7);
SPI_WriteData(0x46);
SPI_WriteData(0x06);
SPI_WriteData(0x0c);
SPI_WriteData(0x00);
SPI_WriteData(0x00);


SPI_WriteComm(0xc3); //Power Control 3
SPI_WriteData(0x07);
SPI_WriteData(0x03);
SPI_WriteData(0x04);
SPI_WriteData(0x04);
```

```
SPI_WriteData(0x04);
LCD_delay(40);

SPI_WriteComm(0xc4); //Power Control 4
SPI_WriteData(0x12);//11
SPI_WriteData(0x24);//23
SPI_WriteData(0x18);//12     16
SPI_WriteData(0x18);//12     16
SPI_WriteData(0x02);//05
SPI_WriteData(0x49);//6d   49     //6A
LCD_delay(20);

SPI_WriteComm(0xc5); //Power Control 5
SPI_WriteData(0x69);   //69
LCD_delay(10);

SPI_WriteComm(0xc6); //Power Control 6
SPI_WriteData(0x41);//41 40
SPI_WriteData(0x63);
LCD_delay(1);

SPI_WriteComm(0xd0); //Positive Gamma Curve for Red
SPI_WriteData(0x01);
SPI_WriteData(0x26);
SPI_WriteData(0x71);
SPI_WriteData(0x16);
SPI_WriteData(0x04);
SPI_WriteData(0x03);
SPI_WriteData(0x51);
SPI_WriteData(0x15);
SPI_WriteData(0x04);

SPI_WriteComm(0xd1); //Negative Gamma Curve for Red
SPI_WriteData(0x01);
SPI_WriteData(0x26);
SPI_WriteData(0x71);
SPI_WriteData(0x16);
SPI_WriteData(0x04);
SPI_WriteData(0x03);
SPI_WriteData(0x51);
SPI_WriteData(0x15);
SPI_WriteData(0x04);

SPI_WriteComm(0xd2); //Positive Gamma Curve for Green
```

```
SPI_WriteData(0x01);
SPI_WriteData(0x26);
SPI_WriteData(0x71);
SPI_WriteData(0x16);
SPI_WriteData(0x04);
SPI_WriteData(0x03);
SPI_WriteData(0x51);
SPI_WriteData(0x15);
SPI_WriteData(0x04);

SPI_WriteComm(0xd3); //Negative Gamma Curve for Green
SPI_WriteData(0x01);
SPI_WriteData(0x26);
SPI_WriteData(0x71);
SPI_WriteData(0x16);
SPI_WriteData(0x04);
SPI_WriteData(0x03);
SPI_WriteData(0x51);
SPI_WriteData(0x15);
SPI_WriteData(0x04);

SPI_WriteComm(0xd4);//Positive Gamma Curve for Blue
SPI_WriteData(0x01);
SPI_WriteData(0x26);
SPI_WriteData(0x71);
SPI_WriteData(0x16);
SPI_WriteData(0x04);
SPI_WriteData(0x03);
SPI_WriteData(0x51);
SPI_WriteData(0x15);
SPI_WriteData(0x04);

SPI_WriteComm(0xd5);//Negative Gamma Curve for Blue
SPI_WriteData(0x01);
SPI_WriteData(0x26);
SPI_WriteData(0x71);
SPI_WriteData(0x16);
SPI_WriteData(0x04);
SPI_WriteData(0x03);
SPI_WriteData(0x51);
SPI_WriteData(0x15);
SPI_WriteData(0x04);
```

```
  SPI_WriteComm(0x11); //Sleep Out
  LCD_delay(20);

  SPI_WriteComm(0x29);//Display On
      LCD_delay(10);

      SPI_WriteComm(0x36);
  SPI_WriteData(0x88);

  SPI_WriteComm(0x3a);
  SPI_WriteData(0x77);//16/18/24bit
}
```