



OZDISAN ELECTRONIC A.S.

# 5” 16 bit TFT Digital Driver Board Specification

*TDDB-SSD-5.0-40P-16B-V2*

Doc.Version : 1.0



*Ozdisan Electronic R&D and Technical Support Department*

*Email: [ts@ozdisan.com](mailto:ts@ozdisan.com) Tel: +90 2164201882*

[www.ozdisan.com](http://www.ozdisan.com)



## Product Pictures



TDDB-SSD-5.0-40P-16B-V2

34 Pins 16Bit User MPU interface, 40 pins TFT connection, 40mA Backlight Driver

## Main Parameters

Board supply voltage	3.3V
Backlight supply voltage	2.7V ~ 5.5V
Working current	< 1mA
Working temperature scope	-20°C ~ +60°C
Storage temperature scope	-40°C ~ +70°C

## Controller Information

Built-in SSD1963

SSD1963 is a display controller of 1215K byte frame buffer to support up to 864x480x24bit graphics content. It also equips parallel MCU interfaces in a different bus width to receive graphics data and commands from MCU. Its display interface supports common RAM-less LCD driver of color depth up to 24 bit-per pixel.

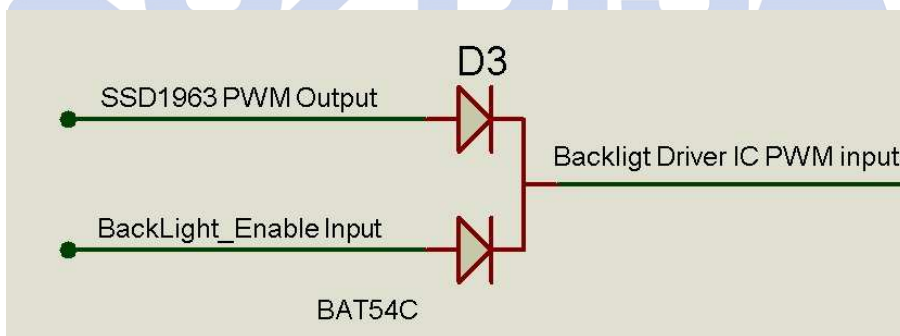


### Data Format

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]
24 bits	1 <sup>st</sup>	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
18 bits	1 <sup>st</sup>							R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0
16 bits (565 format)	1 <sup>st</sup>									R5	R4	R3	R2	R1	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1
16 bits	1 <sup>st</sup>									R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0
	2 <sup>nd</sup>									B7	B6	B5	B4	B3	B2	B1	B0	R7	R6	R5	R4	R3	R2	R1	R0
	3 <sup>rd</sup>									G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
12 bits	1 <sup>st</sup>													R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4
	2 <sup>nd</sup>											G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0		
9 bits	1 <sup>st</sup>																R5	R4	R3	R2	R1	R0	G5	G4	G3
	2 <sup>nd</sup>															G2	G1	G0	B5	B4	B3	B2	B1	B0	
8 bits	1 <sup>st</sup>																	R7	R6	R5	R4	R3	R2	R1	R0
	2 <sup>nd</sup>																	G7	G6	G5	G4	G3	G2	G1	G0
	3 <sup>rd</sup>																	B7	B6	B5	B4	B3	B2	B1	B0

### Backlight

The driver board has 40mA constant current Backlight Driver circuit with maximum 22V output. The backlight can be controlled from MCU Backlight Enable input or SSD1963 PWM output.



Backlight Enable Input	SSD1963 PWM Output	Backlight
0	Off	Off
0	On	Brightness Dimming
1	Off	%100
1	On	%100



### Pin Description for CN1: 34Pin 16 bit input from User's

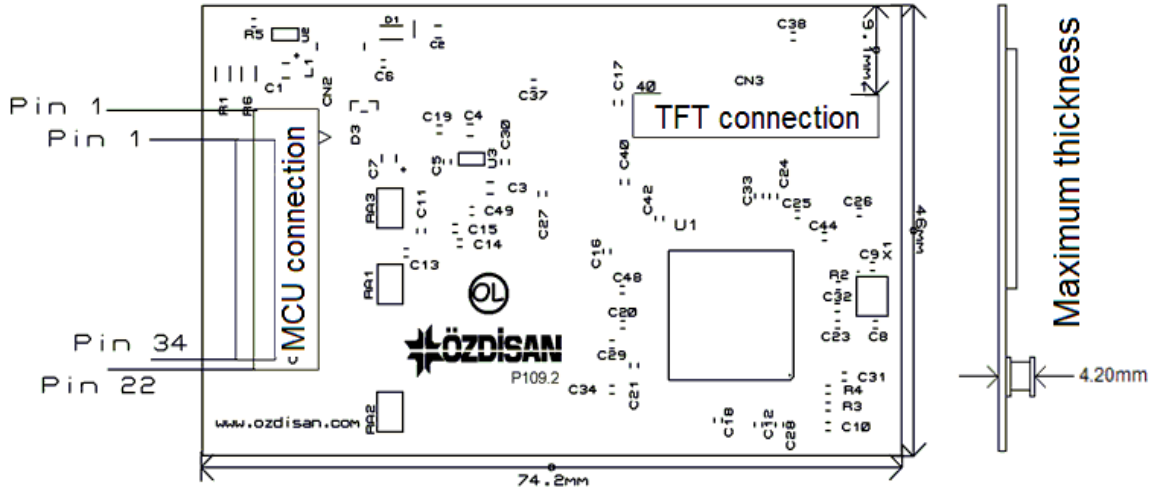
Pin No.	Symbol	Function Description
1	GND	GND
2	GND	GND
3	Vbl	Backlight Supply 2.7V-5.5V
4	Vbl	Backlight Supply 2.7V-5.5V
5	Vcc	Board and TFT Supply 3.3V
6	NC	
7	NC	
8	BLen	Backlight Enable
9	WR	Write Signal-Active Low
10	RD	Read Signal-Active Low
11	RS	Register select (Data Or command)
12	CS	Chip Select Signal- Active Low (Enables data or command sending)
13	TE	Tear Effect
14	RESET	Reset Signal- Active Low
15-30	D0-D15	16 Bit Bidirectional Data bus
31	TP_R	Touch Panel Right
32	TP_B	Touch Panel Bottom
33	TP_L	Touch Panel Left
34	TP_T	Touch Panel Top

### Pin Description for CN3: Pin Connections to TFT Panel

Pin No.	Symbol	Function Description
1	LED-	LED Cathode
2	LED+	LED Anode
3	GND	Power Ground
4	VDD	Power Voltage(3.3V)
5-12	R0-R7	Red Data 0-7
13-20	G0-G7	Green Data 0-7
21-28	B0-B7	Blue Data 0-7
29	GND	Power Ground
30	DCLK	Pixel clock
31	DISP	Display on/off
32	HSYNC	Horizontal sync signal
33	VSYNC	Vertical sync signal
34	DE	Data enable
35	NC	No connection
36	GND	Power Ground
37	X_R	Right electrode
38	Y_B	Bottom electrode
39	X_L	Left electrode
40	Y_T	Top electrode



### Outline Dimensions



### Board Connections





## Other Tools used with the boards

DS1057-03-5A34W4L10E1B: Flat cable with 34 pins 10cm

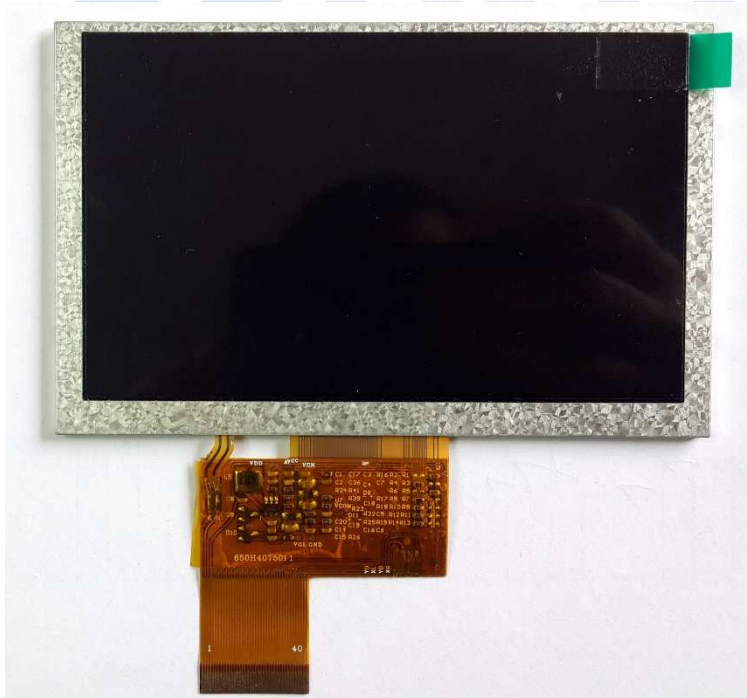


FPC4AMR6-34TNBT-U: Connector with 34 pins



5" TFT panel

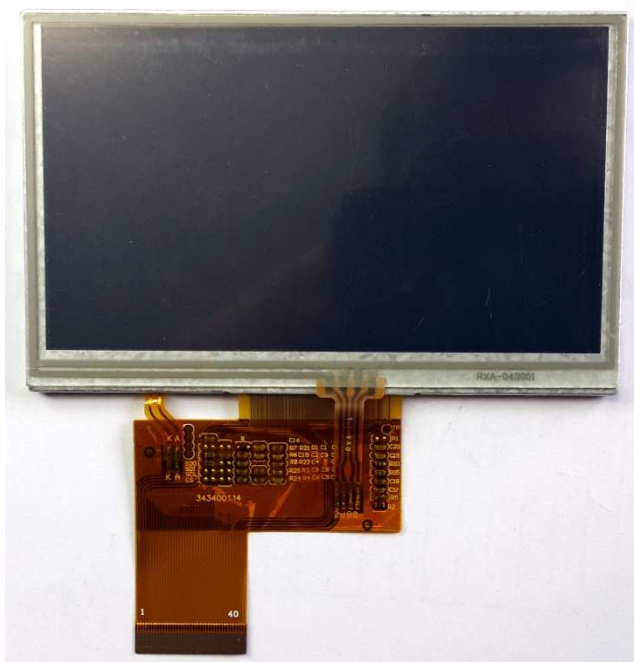
SAT050AT40DHY0-A0 (480X272 pixel), SAT050AT40HHY0-A0(800X480 pixel)





5" TFT panel with touch panel

SAT050AT40DHYO-A0-TP (480X272 pixel), SAT050AT40HHY0-A0-TP (800X480 pixel)



5" TFT panel with bracket

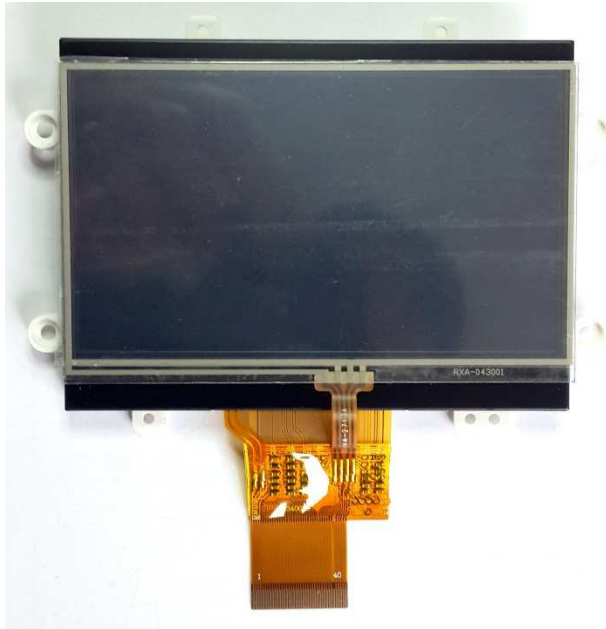
SAT050AT40DHYO-C0 (480X272 pixel)



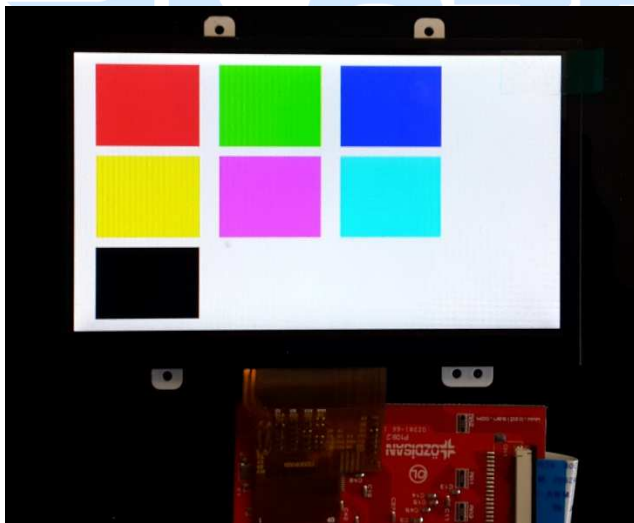


5" TFT panel with bracket and Touch Panel

SAT050AT40DHYO-C0-VSD-TP (480X272 pixel)



**Code Sample for 480X272 resolution**



*TFT screen which is working with this software*

```
//pin defines for STM32 in mikroc compiler
unsigned int TFT_DataPort at GPIOE_ODR;
sbit TFT_RST at GPIOB_ODR.B1;
sbit TFT_RS at GPIOC_ODR.B4;
sbit TFT_CS at GPIOC_ODR.B5;
sbit TFT_RD at GPIOA_ODR.B7;
sbit TFT_WR at GPIOA_ODR.B6;
```





```
sbit TFT_BLED at GPIOA_ODR.B5;
```

```
void Send_TFT_Command(char index)
```

```
{  
    TFT_CS = 0;  
    TFT_RS = 0;  
    TFT_DataPort = index;  
    TFT_WR = 0;  
    asm nop;  
    TFT_WR = 1;  
    TFT_CS = 1;  
}
```

```
void Send_TFT_Data_8(unsigned char index)
```

```
{  
    TFT_CS = 0;  
    TFT_RS = 1;  
    TFT_DataPort = index;  
    TFT_WR = 0;  
    asm nop;  
    TFT_WR = 1;  
    TFT_CS = 1;  
}
```

```
void Send_TFT_Data_16(int index)
```

```
{  
    TFT_CS = 0;  
    TFT_RS = 1;  
    TFT_DataPort = index;  
    /*  
    if data port has 8 bit output  
    TFT_DataH = index>>8;  
    TFT_DataL = index&0x00FF;  
    */  
    TFT_WR = 0;  
    asm nop;
```



```

TFT_WR = 1;
TFT_CS = 1;
}

void init_SSD1963(void)
{
    TFT_RST = 0; // Hold in reset
    TFT_RS = 1; // Enable data access
    TFT_CS = 1; // Disable LCD
    TFT_RD = 1;
    TFT_WR = 1;

    // Release from reset
    Delay_ms(100);
    TFT_RST = 1;
    Delay_ms(100);

    Send_TFT_Command(0x01); //Software Reset
    Send_TFT_Command(0x01); //Software Reset
    Send_TFT_Command(0x01); //Software Reset
    Delay_ms(100);
    Send_TFT_Command(0xE2); //SET PLL freq=110MHz
    Send_TFT_Data_8(0x21); //N=33, 33X10=330Mhz
    Send_TFT_Data_8(0x02); //M=3 330/3=110Mhz
    Send_TFT_Data_8(0x54);
    Delay_ms(100);
    Send_TFT_Command(0xE0); //PLL settings
    Send_TFT_Data_8(0x01); //START PLL
    delay_ms(1); // Wait to let the PLL stable
    Send_TFT_Command(0xE0); //PLL settings
    Send_TFT_Data_8(0x03); //LOCK PLL
    delay_ms(1);

    Send_TFT_Command(0xB0); //LCD MODE Settings
    Send_TFT_Data_8(0x20); //TFT panel data width 24bit, FRC, dithering disable
    Send_TFT_Data_8(0x00); //hsync+Vsync+DEN
    Send_TFT_Data_8(0x01); //horizontal panel size(horizontal lines) HightByte
    Send_TFT_Data_8(0xDF); //LowByte

```



Send\_TFT\_Data\_8(0x01); //vertical panel size(vertical lines) HightByte

Send\_TFT\_Data\_8(0x0F); //SET vertical size LowByte

Send\_TFT\_Data\_8(0x00); //avaible if serial RGB mode is selected.

delay\_ms(1);

Send\_TFT\_Command(0xF0);

Send\_TFT\_Data\_8(0x03); //SSD1963 data input format data 16 bit

delay\_ms(1);

Send\_TFT\_Command(0x3A); //Pixel format

Send\_TFT\_Data\_8(0x60);

delay\_ms(1);

Send\_TFT\_Command(0xE6); //SET PCLK freq=10MHz = 110MHz \* LCDC\_FPR / 2^20

Send\_TFT\_Data\_8(0x01);

Send\_TFT\_Data\_8(0x45);

Send\_TFT\_Data\_8(0x47);

//////////

delay\_ms(1);

Send\_TFT\_Command(0xB4);

Send\_TFT\_Data\_8(0x02); //horizontal total period (display + non-display)-1 highbyte

Send\_TFT\_Data\_8(0x0C); //low byte

Send\_TFT\_Data\_8(0x00); //Horizontal Pulse Width + Horizontal Back Porch highbyte

Send\_TFT\_Data\_8(0x28); //low byte

Send\_TFT\_Data\_8(0x00); //Horizontal Sync Pulse Width

Send\_TFT\_Data\_8(0x00); //Hsync pulse start position

Send\_TFT\_Data\_8(0x00); //lowbyte

Send\_TFT\_Data\_8(0x00); //for serial RGB mode

delay\_ms(1);

Send\_TFT\_Command(0xB6);

Send\_TFT\_Data\_8(0x01); //vertical total period (display + non-display)-1 highbyte

Send\_TFT\_Data\_8(0x21); //low byte

Send\_TFT\_Data\_8(0x00); //vertical Pulse Width + vertical Back Porch highbyte

Send\_TFT\_Data\_8(0x08); //low byte

Send\_TFT\_Data\_8(0x00); //vertical Sync Pulse Width

Send\_TFT\_Data\_8(0x00); //Vsync pulse start position

Send\_TFT\_Data\_8(0x00); //lowbyte



```
delay_ms(1);
```

```
////////////////////////////////////
```

```
Send_TFT_Command(0x36); // Address Mode
```

```
Send_TFT_Data_8(0x00);
```

```
delay_ms(1);
```

```
Send_TFT_Command(0x29); //SET display on
```

```
//backlight PWM setting.
```

```
Send_TFT_Command(0xBE);
```

```
Send_TFT_Data_8(0x01); //PLL clock / (256 * (PWMF[7:0] + 1)) / 256
```

```
Send_TFT_Data_8(0x64); //PWM duty cycle
```

```
Send_TFT_Data_8(0x01); //PWM, DBC enable/disable setting.
```

```
Send_TFT_Data_8(0x00); //DBC manual brightness
```

```
Send_TFT_Data_8(0x00); //DBC minimum brightness
```

```
Send_TFT_Data_8(0x00); //Brightness prescaler
```

```
}
```

```
void draw_rectagle(unsigned int X1,unsigned int X2,unsigned int Y1,unsigned int Y2,unsigned int red,unsigned int green,unsigned int blue)
```

```
{
```

```
unsigned char X_point_1_1;
```

```
unsigned char X_point_1_2;
```

```
unsigned char X_point_2_1;
```

```
unsigned char X_point_2_2;
```

```
unsigned char Y_point_1_1;
```

```
unsigned char Y_point_1_2;
```

```
unsigned char Y_point_2_1;
```

```
unsigned char Y_point_2_2;
```

```
unsigned long temp1;
```

```
unsigned long temp2;
```

```
unsigned long frame_pixel;
```

```
unsigned long pixel_CNT;
```

```
unsigned int color;
```

```
red=red<<11;
```

```
green=green<<5;
```



```
color=blue|green|red;
```

```
temp1= X2-X1+1;
```

```
temp2= Y2-Y1+1;
```

```
frame_pixel=temp1*temp2;
```

```
X_point_1_1=X1>>8;
```

```
X_point_1_2=X1&0x00FF;
```

```
X_point_2_1=X2>>8;
```

```
X_point_2_2=X2&0x00FF;
```

```
Y_point_1_1=Y1>>8;
```

```
Y_point_1_2=Y1&0x00FF;
```

```
Y_point_2_1=Y2>>8;
```

```
Y_point_2_2=Y2&0x00FF;
```

```
Send_TFT_Command(0x2A);//Setup the frame buffer vertical addressing range
```

```
Send_TFT_Data_8(X_point_1_1);
```

```
Send_TFT_Data_8(X_point_1_2);
```

```
Send_TFT_Data_8(X_point_2_1);
```

```
Send_TFT_Data_8(X_point_2_2);
```

```
Send_TFT_Command(0x2B);// Setup the frame buffer horizontal address range
```

```
Send_TFT_Data_8(Y_point_1_1);
```

```
Send_TFT_Data_8(Y_point_1_2);
```

```
Send_TFT_Data_8(Y_point_2_1);
```

```
Send_TFT_Data_8(Y_point_2_2);
```

```
Send_TFT_Command(0x2C);
```

```
pixel_CNT=0;
```

```
while(pixel_CNT<=frame_pixel)
```

```
{
```

```
    Send_TFT_Data_16(color);
```

```
    pixel_CNT++;
```

```
}
```



```

}
void main()
{
  ///pin configurations for STM32 in Mikroc compiler
  GPIO_Digital_Output(&GPIOA_BASE, _GPIO_PINMASK_ALL); // Set PORTB as digital output
  GPIO_Digital_Output(&GPIOB_BASE, _GPIO_PINMASK_ALL); // Set PORTB as digital output
  GPIO_Digital_Output(&GPIOC_BASE, _GPIO_PINMASK_ALL); // Set PORTB as digital output
  GPIO_Digital_Output(&GPIOD_BASE, _GPIO_PINMASK_ALL); // Set PORTB as digital output
  GPIO_Digital_Output(&GPIOE_BASE, _GPIO_PINMASK_ALL); // Set PORTB as digital output

  GPIOA_ODR = 0;
  GPIOB_ODR = 0;
  GPIOC_ODR = 0;
  GPIOD_ODR = 0;
  GPIOE_ODR = 0;
  GPIOA_OSPEEDR=0xFFFFFFFF;
  GPIOB_OSPEEDR=0xFFFFFFFF;
  GPIOC_OSPEEDR=0xFFFFFFFF;
  GPIOD_OSPEEDR=0xFFFFFFFF;
  GPIOE_OSPEEDR=0xFFFFFFFF;

  TFT_BLED=1;
  /*
  TFT_BLED=1 -> backlight %100
  TFT_BLED=0 and SSD1963 PWM on-> backlight=PWM
  TFT_BLED=0 and SSD1963 PWM off-> backlight=off
  */

  init_SSD1963();

  //color depth in 16 bit mode
  //red 0-31
  //green 0-63
  //blue 0-31
  draw_rectagle(0,479,0,271,31,63,31);

  while(1)
  {
    draw_rectagle(20,120,10,90,31,0,0);
  }
}

```



```

draw_rectagle(140,240,10,90,0,63,0);
draw_rectagle(260,360,10,90,0,0,31);
draw_rectagle(20,120,100,180,31,63,0);
draw_rectagle(140,240,100,180,31,0,31);
draw_rectagle(260,360,100,180,0,63,31);
draw_rectagle(20,120,190,260,0,0,0);
}
}

```

### Code Sample for 800X480 resolution



*TFT screen which is working with this software*

```

//pin defines for STM32 in mikro compiler
unsigned int TFT_DataPort at GPIOE_ODR;
sbit TFT_RST at GPIOB_ODR.B1;
sbit TFT_RS at GPIOC_ODR.B4;
sbit TFT_CS at GPIOC_ODR.B5;
sbit TFT_RD at GPIOA_ODR.B7;
sbit TFT_WR at GPIOA_ODR.B6;
sbit TFT_BLED at GPIOA_ODR.B5;
void Send_TFT_Command(char index)
{
TFT_CS = 0;
TFT_RS = 0;
TFT_DataPort = index;
TFT_WR = 0;
asm nop;
TFT_WR = 1;
TFT_CS = 1;

```



```
}

void Send_TFT_Data_8(unsigned char index)
{
    TFT_CS = 0;
    TFT_RS = 1;
    TFT_DataPort = index;
    TFT_WR = 0;
    asm nop;
    TFT_WR = 1;
    TFT_CS = 1;
}

void Send_TFT_Data_16(int index)
{
    TFT_CS = 0;
    TFT_RS = 1;
    TFT_DataPort = index;
    /*
    if data port has 8 bit output
    TFT_DataH = index>>8;
    TFT_DataL = index&0x00FF;
    */
    TFT_WR = 0;
    asm nop;
    TFT_WR = 1;
    TFT_CS = 1;
}

void init_SSD1963(void)
{
    TFT_RST = 0; // Hold in reset
    TFT_RS = 1; // Enable data access
    TFT_CS = 1; // Disable LCD
    TFT_RD = 1;
    TFT_WR = 1;

    // Release from reset
    Delay_ms(100);
    TFT_RST = 1;
    Delay_ms(100);
```





Send\_TFT\_Command(0x01); //Software Reset

Send\_TFT\_Command(0x01); //Software Reset

Send\_TFT\_Command(0x01); //Software Reset

Delay\_ms(100);

Send\_TFT\_Command(0xE2); //SET PLL freq=110MHz

Send\_TFT\_Data\_8(0x21); //N=33, 33X10=330Mhz

Send\_TFT\_Data\_8(0x02); //M=3 330/3=110Mhz

Send\_TFT\_Data\_8(0x54);

Delay\_ms(100);

Send\_TFT\_Command(0xE0); //PLL settings

Send\_TFT\_Data\_8(0x01); //START PLL

delay\_ms(1); // Wait to let the PLL stable

Send\_TFT\_Command(0xE0); //PLL settings

Send\_TFT\_Data\_8(0x03); //LOCK PLL

delay\_ms(1);

Send\_TFT\_Command(0xB0); //LCD MODE Settings

Send\_TFT\_Data\_8(0x20); //TFT panel data width 24bit, FRC, dithering disable

Send\_TFT\_Data\_8(0x00); //hsync+Vsync+DEN

Send\_TFT\_Data\_8(0x03); //horizontal panel size(horizontal lines) HightByte

Send\_TFT\_Data\_8(0x1F); //LowByte

Send\_TFT\_Data\_8(0x01); //vertical panel size(vertical lines) HightByte

Send\_TFT\_Data\_8(0xDF); //SET vertical size LowByte

Send\_TFT\_Data\_8(0x00); //avaible if serial RGB mode is selected.

delay\_ms(1);

Send\_TFT\_Command(0xF0);

Send\_TFT\_Data\_8(0x03); //SSD1963 data input format data 16 bit

delay\_ms(1);

Send\_TFT\_Command(0x3A); //Pixel format

Send\_TFT\_Data\_8(0x60);

delay\_ms(1);

delay\_ms(1);

Send\_TFT\_Command(0xE6); //SET PCLK freq=33MHz = 110MHz \* LCDC\_FPR / 2^20



```

Send_TFT_Data_8(0x04);
Send_TFT_Data_8(0xD7);
Send_TFT_Data_8(0xF9);
delay_ms(1);
Send_TFT_Command(0xB4);
Send_TFT_Data_8(0x04); //horizontal total period (display + non-display)-1 highbyte
Send_TFT_Data_8(0x1F); //low byte
Send_TFT_Data_8(0x00); //Horizontal Pulse Width + Horizontal Back Porch highbyte
Send_TFT_Data_8(0x2E); //low byte
Send_TFT_Data_8(0x00); //Horizontal Sync Pulse Width
Send_TFT_Data_8(0x00); //Hsync pulse start position
Send_TFT_Data_8(0x00); //lowbyte
Send_TFT_Data_8(0x00); //for serial RGB mode
delay_ms(1);
Send_TFT_Command(0xB6);
Send_TFT_Data_8(0x02); //vertical total period (display + non-display)-1 highbyte
Send_TFT_Data_8(0x0C); //low byte
Send_TFT_Data_8(0x00); //vertical Pulse Width + vertical Back Porch highbyte
Send_TFT_Data_8(0x17); //low byte
Send_TFT_Data_8(0x00); //vertical Sync Pulse Width
Send_TFT_Data_8(0x00); //Vsync pulse start position
Send_TFT_Data_8(0x00); //lowbyte
delay_ms(1);
Send_TFT_Command(0x36); // Address Mode
Send_TFT_Data_8(0x00);
delay_ms(1);
Send_TFT_Command(0x29); //SET display on

//backlight PWM setting.
Send_TFT_Command(0xBE);
Send_TFT_Data_8(0x01); //PLL clock / (256 * (PWMF[7:0] + 1)) / 256
Send_TFT_Data_8(0x64); //PWM duty cycle
Send_TFT_Data_8(0x01); //PWM, DBC enable/disable setting.
Send_TFT_Data_8(0x00); //DBC manual brightness
Send_TFT_Data_8(0x00); //DBC minimum brightness
Send_TFT_Data_8(0x00); //Brightness prescaler
}

```



```

void draw_rectangle(unsigned int X1,unsigned int X2,unsigned int Y1,unsigned int Y2,unsigned int red,unsigned int green,unsigned int blue)
{
    unsigned char X_point_1_1;
    unsigned char X_point_1_2;
    unsigned char X_point_2_1;
    unsigned char X_point_2_2;
    unsigned char Y_point_1_1;
    unsigned char Y_point_1_2;
    unsigned char Y_point_2_1;
    unsigned char Y_point_2_2;

    unsigned long temp1;
    unsigned long temp2;
    unsigned long frame_pixel;
    unsigned long pixel_CNT;

    unsigned int color;

    red=red<<11;
    green=green<<5;
    color=blue|green|red;

    temp1= X2-X1+1;
    temp2= Y2-Y1+1;
    frame_pixel=temp1*temp2;

    X_point_1_1=X1>>8;
    X_point_1_2=X1&0x00FF;

    X_point_2_1=X2>>8;
    X_point_2_2=X2&0x00FF;

    Y_point_1_1=Y1>>8;
    Y_point_1_2=Y1&0x00FF;
    Y_point_2_1=Y2>>8;
    Y_point_2_2=Y2&0x00FF;

    Send_TFT_Command(0x2A);//Setup the frame buffer vertical addressing range
    Send_TFT_Data_8(X_point_1_1);

```



```

Send_TFT_Data_8(X_point_1_2);
Send_TFT_Data_8(X_point_2_1);
Send_TFT_Data_8(X_point_2_2);
Send_TFT_Command(0x2B);// Setup the frame buffer horizontal address range
Send_TFT_Data_8(Y_point_1_1);
Send_TFT_Data_8(Y_point_1_2);
Send_TFT_Data_8(Y_point_2_1);
Send_TFT_Data_8(Y_point_2_2);
Send_TFT_Command(0x2C);
pixel_CNT=0;
while(pixel_CNT<=frame_pixel)
{
    Send_TFT_Data_16(color);
    pixel_CNT++;
}
}
void main()
{
    //pin configurations for STM32 in MikroC compiler
    GPIO_Digital_Output(&GPIOA_BASE, _GPIO_PINMASK_ALL); // Set PORTB as digital output
    GPIO_Digital_Output(&GPIOB_BASE, _GPIO_PINMASK_ALL); // Set PORTB as digital output
    GPIO_Digital_Output(&GPIOC_BASE, _GPIO_PINMASK_ALL); // Set PORTB as digital output
    GPIO_Digital_Output(&GPIOD_BASE, _GPIO_PINMASK_ALL); // Set PORTB as digital output
    GPIO_Digital_Output(&GPIOE_BASE, _GPIO_PINMASK_ALL); // Set PORTB as digital output

    GPIOA_ODR = 0;
    GPIOB_ODR = 0;
    GPIOC_ODR = 0;
    GPIOD_ODR = 0;
    GPIOE_ODR = 0;

    GPIOA_OSPEEDR=0xFFFFFFFF;
    GPIOB_OSPEEDR=0xFFFFFFFF;
    GPIOC_OSPEEDR=0xFFFFFFFF;
    GPIOD_OSPEEDR=0xFFFFFFFF;
    GPIOE_OSPEEDR=0xFFFFFFFF;

    TFT_BLED=1;
    /*
    TFT_BLED=1 -> backlight %100
    TFT_BLED=0 and SSD1963 PWM on-> backlight=PWM
    TFT_BLED=0 and SSD1963 PWM off-> backlight=off

```



\*/

```
init_SSD1963();
```

```
//color depth in 16 bit mode
```

```
//red 0-31
```

```
//green 0-63
```

```
//blue 0-31
```

```
draw_rectagle(0,799,0,479,31,63,31);
```

```
while(1)
```

```
{
```

```
draw_rectagle(20,220,10,110,31,0,0);
```

```
draw_rectagle(240,440,10,110,0,63,0);
```

```
draw_rectagle(460,660,10,110,0,0,31);
```

```
draw_rectagle(20,220,130,230,31,63,0);
```

```
draw_rectagle(240,440,130,230,31,0,31);
```

```
draw_rectagle(460,660,130,230,0,63,31);
```

```
draw_rectagle(20,220,250,350,0,0,0);
```

```
}
```

```
}
```

