

Arduino 3.2" Colored TFT Touch LCD Shield

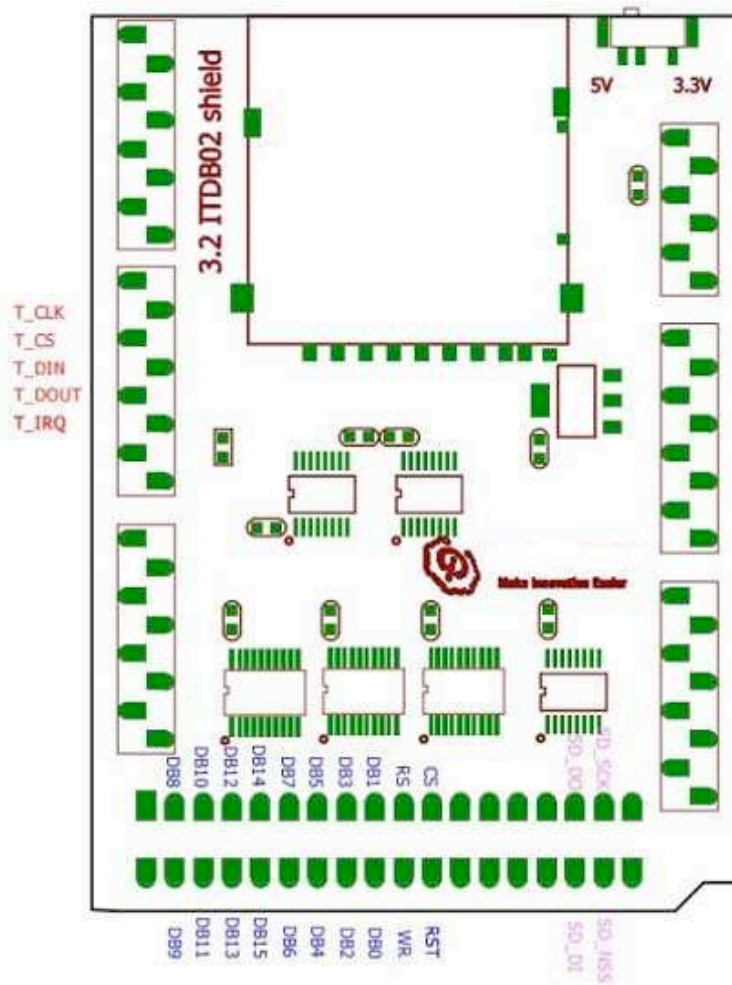


Description

3.2" TFT Touch Shield is an Arduino Mega compatible multicolored TFT display with a 4-wire resistive touch-screen. This display has a resistive touchscreen attached to it already, so you can detect finger presses anywhere on the screen. It is available in an Arduino shield compatible pinout for attachment. The TFT driver is based on SSD1289 driver chip with 8bit data and 4bit control interface.

Features

- Compatible with 3.3/5V operation voltage level
- Compatible with UTFT library
- With SD Card Socket



The Arduino 3.2" TFT Touch module uses the SSD1289 controller, it supports 8-bit data interface.

Arduino Mega PIN	Description
D2	T_IRQ
D3	T_DOUT
D4	T_DIN
D5	T_CS
D6	T_CLK
D22	DB8
D23	DB9
D24	DB10
D25	DB11
D26	DB12
D27	DB13
D28	DB14

D29	DB15
D30	DB7
D31	DB6
D32	DB5
D33	DB4
D34	DB3
D35	DB2
D36	DB1
D37	DB0
D38	RS
D39	WR
D40	CS
D41	RST
D50	SD_MISO
D51	SD_MOSI
D52	SD_SCK
D53	SD_NSS

Software

This shield is compatible with UTFT graphic library for Arduino

<http://henningkarlsen.com/electronics/download.php?f=UTouch.rar>

For touch function library

<http://henningkarlsen.com/electronics/download.php?f=UTouch.rar>

```
/******
```

Pay an attention!

This code is designed for Arduino board.

```
*****/
```

```
#include <SdFat.h>
```

```
Sd2Card card;
```

```
/******
```

```
Define zone
```

```
*****/
```

```
#define RS 38
```

```
#define WR 39
```

```
#define CS 40
```

```
#define RST 41
```

```
#define T_CLK 6
```

```
#define T_CS 5
```

```
#define T_DIN 4
```

```
#define T_DOUT 3
```

```
#define T_IRQ 2
```

```
#define X_CONST 240
```

```
#define Y_CONST 320
```

```
#define PREC_TOUCH_CONST 10
```

```
#define PixSizeX 13.78
```

```
#define PixOffsX 411
```

```
#define PixSizeY 11.01
```

```
#define PixOffsY 378
```

```
/* LCD color */
```

```
#define White      0xFFFF
```

```
#define Black      0x0000
```

```
#define Blue       0x001F
```

```
#define Blue2      0x051F
```

```
#define Red      0xF800
#define Magenta 0xF81F
#define Green   0x07E0
#define Cyan    0x7FFF
#define Yellow  0xFFE0
```

```
/******
```

```
Val Zone
```

```
*****/
```

```
int TP_X,TP_Y;
```

```
/******
```

```
Standard C functions zone
```

```
*****/
```

```
void Write_Command(unsigned int c)
```

```
{
    digitalWrite(CS,LOW);
    digitalWrite(RS,LOW);
    PORTA = c >> 8;
    PORTC = c;
    digitalWrite(WR,LOW);
    digitalWrite(WR,HIGH);
    digitalWrite(CS,HIGH);
}
```

```
void Write_Data(unsigned int c)
```

```
{
    digitalWrite(CS,LOW);
    digitalWrite(RS,HIGH);
    PORTA = c >> 8;
    PORTC = c;
    digitalWrite(WR,LOW);
    digitalWrite(WR,HIGH);
    digitalWrite(CS,HIGH);
}
```

```
void Write_Command_Data(unsigned int cmd,unsigned int dat)
{
    Write_Command(cmd);
    Write_Data(dat);
}
```

```
void Lcd_Init()
{
    pinMode(RS,OUTPUT);
    pinMode(WR,OUTPUT);
    pinMode(CS,OUTPUT);
    pinMode(RST,OUTPUT);

    DDRA = 0xff;
    DDRC = 0xff;

    digitalWrite(RST,HIGH);
    delay(1);
    digitalWrite(RST,LOW);
    delay(1);

    digitalWrite(RST,HIGH);
    digitalWrite(CS,HIGH);
    digitalWrite(WR,HIGH);
    delay(20);

    Write_Command_Data(0x0000,0x0001);
    Write_Command_Data(0x0003,0xA8A4);
    Write_Command_Data(0x000C,0x0000);
    Write_Command_Data(0x000D,0x080C);
    Write_Command_Data(0x000E,0x2B00);
    Write_Command_Data(0x001E,0x00B7);
    Write_Command_Data(0x0001,0x2B3F);
    Write_Command_Data(0x0002,0x0600);
    Write_Command_Data(0x0010,0x0000);
    Write_Command_Data(0x0011,0x6070);
    Write_Command_Data(0x0005,0x0000);
```

```

Write_Command_Data(0x0006,0x0000);
Write_Command_Data(0x0016,0xEF1C);
Write_Command_Data(0x0017,0x0003);
Write_Command_Data(0x0007,0x0233);
Write_Command_Data(0x000B,0x0000);
Write_Command_Data(0x000F,0x0000);
Write_Command_Data(0x0041,0x0000);
Write_Command_Data(0x0042,0x0000);
Write_Command_Data(0x0048,0x0000);
Write_Command_Data(0x0049,0x013F);
Write_Command_Data(0x004A,0x0000);
Write_Command_Data(0x004B,0x0000);
Write_Command_Data(0x0044,0xEF00);
Write_Command_Data(0x0045,0x0000);
Write_Command_Data(0x0046,0x013F);
Write_Command_Data(0x0030,0x0707);
Write_Command_Data(0x0031,0x0204);
Write_Command_Data(0x0032,0x0204);
Write_Command_Data(0x0033,0x0502);
Write_Command_Data(0x0034,0x0507);
Write_Command_Data(0x0035,0x0204);
Write_Command_Data(0x0036,0x0204);
Write_Command_Data(0x0037,0x0502);
Write_Command_Data(0x003A,0x0302);
Write_Command_Data(0x003B,0x0302);
Write_Command_Data(0x0023,0x0000);
Write_Command_Data(0x0024,0x0000);
Write_Command_Data(0x0025,0x8000);
Write_Command_Data(0x004f,0x0000);
Write_Command_Data(0x004e,0x0000);
Write_Command(0x0022);
}
void SetXY(unsigned int x0,unsigned int y0,unsigned int x1,unsigned int y1)
{
Write_Command_Data(0x0044,(x1<<8)+x0);
Write_Command_Data(0x0045,y0);
Write_Command_Data(0x0046,y1);
}

```

```

    Write_Command_Data(0x004e,x0);
    Write_Command_Data(0x004f,y0);
    Write_Command (0x0022);//LCD_WriteCMD(GRAMWR);
}
void Pant(unsigned int color)
{
    int i,j;
    SetXY(0,0,239,319);

    for(i=0;i<320;i++)
    {
        for (j=0;j<240;j++)
        {
            Write_Data(color);
        }
    }
}
void LCD_clear()
{
    unsigned int i,j;
    SetXY(0,0,239,319);
    for(i=0;i<320;i++)
    {
        for(j=0;j<240;j++)
        {
            Write_Data(0x0000);
        }
    }
}

void Touch_Init(void)
{
    pinMode(T_CLK, OUTPUT);
    pinMode(T_CS, OUTPUT);
    pinMode(T_DIN, OUTPUT);
    pinMode(T_DOOUT, INPUT);
    pinMode(T_IRQ, INPUT);
}

```



```

    digitalWrite(T_CS, HIGH);
    digitalWrite(T_CLK, HIGH);
    digitalWrite(T_DIN, HIGH);
    digitalWrite(T_CLK, HIGH);
}

void Touch_WriteData(unsigned char data)
{
    unsigned char temp;
    unsigned char nop;
    unsigned char count;

    temp=data;
    digitalWrite(T_CLK,LOW);

    for(count=0; count<8; count++)
    {
        if(temp & 0x80)
            digitalWrite(T_DIN, HIGH);
        else
            digitalWrite(T_DIN, LOW);
        temp = temp << 1;
        digitalWrite(T_CLK, LOW);
        nop++;
        digitalWrite(T_CLK, HIGH);
        nop++;
    }
}

```

```

unsigned int Touch_ReadData()
{
    unsigned char nop;
    unsigned int data = 0;
    unsigned char count;
    for(count=0; count<12; count++)
    {
        data <<= 1;
    }
}

```

```

        digitalWrite(T_CLK, HIGH);
        nop++;
        digitalWrite(T_CLK, LOW);
        nop++;
        if (digitalRead(T_DOUT))
            data++;
    }
    return(data);
}
void Touch_Read()
{
    unsigned long tx=0;
    unsigned long ty=0;

    digitalWrite(T_CS,LOW);

    for (int i=0; i<PREC_TOUCH_CONST; i++)
    {
        Touch_WriteData(0x90);
        digitalWrite(T_CLK,HIGH);
        digitalWrite(T_CLK,LOW);
        ty+=Touch_ReadData();

        Touch_WriteData(0xD0);
        digitalWrite(T_CLK,HIGH);
        digitalWrite(T_CLK,LOW);
        tx+=Touch_ReadData();
    }

    digitalWrite(T_CS,HIGH);

    TP_X=tx/PREC_TOUCH_CONST;
    TP_Y=ty/PREC_TOUCH_CONST;
}

char Touch_DataAvailable()
{
    char avail;

```

```

    avail = !digitalRead(T_IRQ);
    return avail;
}

int Touch_GetX()
{
    int value;
    value = ((TP_X-PixOffsX)/PixSizeX);
    if (value < 0)
        value = 0;
    return value;
}
int Touch_GetY()
{
    int value;
    value = ((TP_Y-PixOffsY)/PixSizeY);
    if (value < 0)
        value = 0;
    return value;
}
/*****
Arduino functions zone
*****/
void setup()
{
    Lcd_Init();
    if (!card.init(SPI_HALF_SPEED, 53))
    {
        while(1)
        {
            Pant(0xffff);
            Pant(0x0000);
        }
    }
}

Touch_Init();
LCD_clear();
    Pant(0xf800);

```

```
    Pant(0x07e0);
    Pant(0x001f);
    Pant(0xffff);
    Pant(0x0000);
}

void loop()
{
    unsigned int i,j;
    while(Touch_DataAvailable() == 1)
    {
        Touch_Read();
        i = Touch_GetX();
        j = Touch_GetY();
        SetXY(i,i,j,j);
        Write_Data(0xFFFF);
    }
}
```