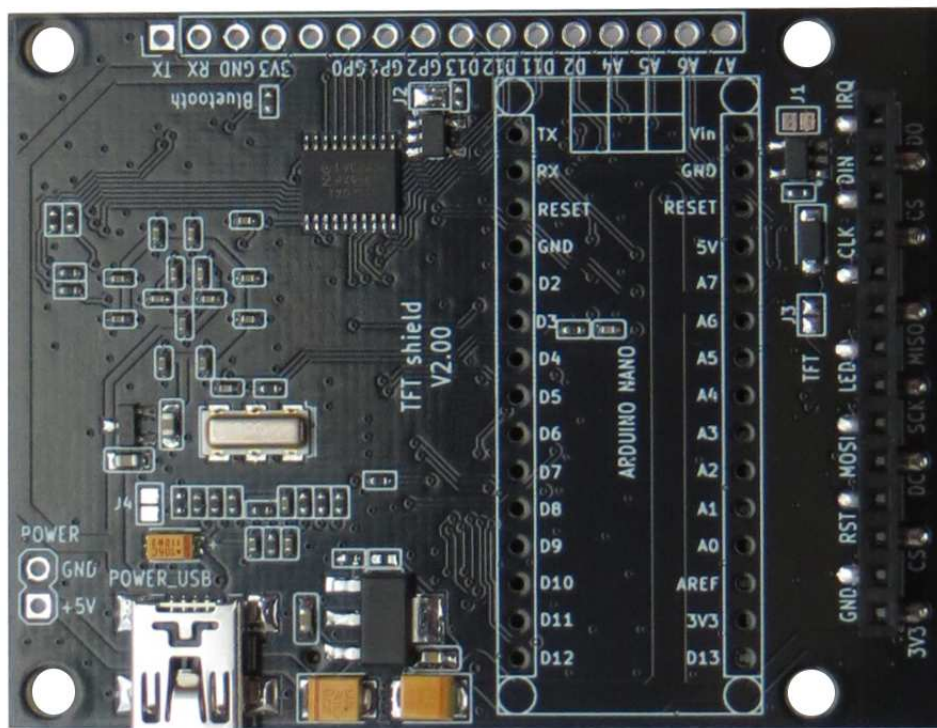


The Arduino Nano TFT shield Tutorial, Reference and Cookbook

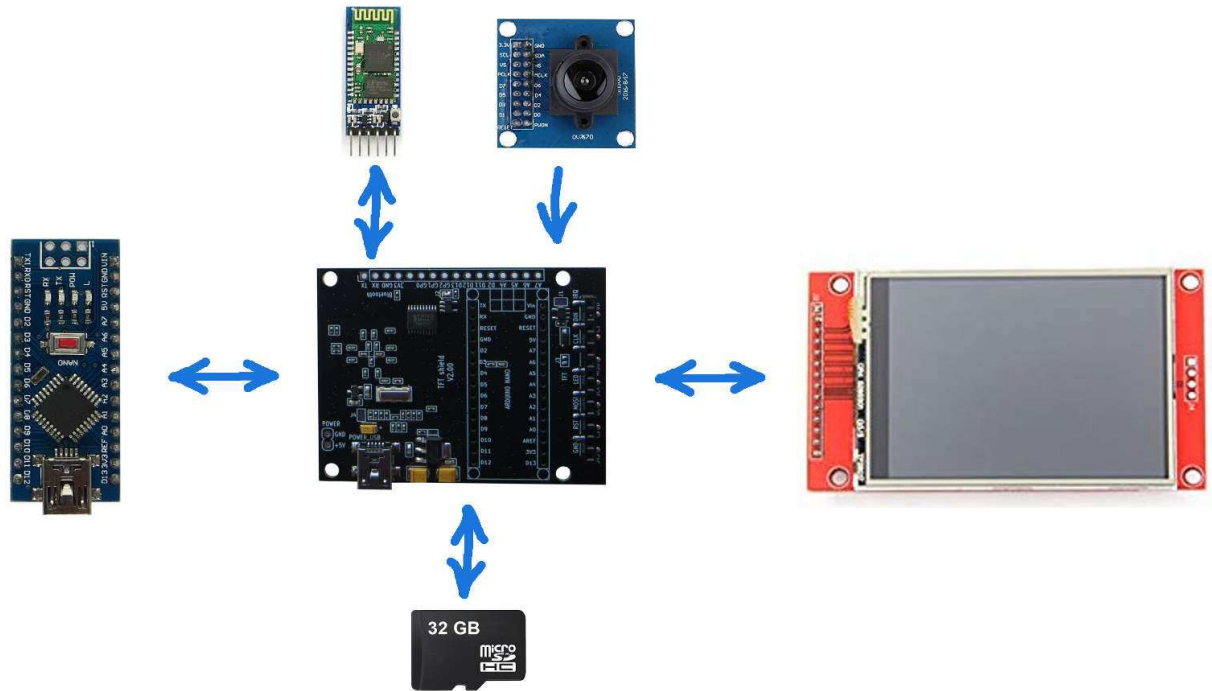
(Preliminary)



Contents

1. Overview	3
Arduino Nano connector	4
5V power connector	4
USB power connector	5
SPI TFT display connector	5
Bluetooth connector and internal 3V power	6
GPIO connector	6
Analog Inputs connector	7
Digital I/O connector	7
Camera connector	8
MicroSD connector	8
2. Quick start	9
2.1 Preparation for Work	9
2.1.1 Installing Libraries	9
2.1.2 Device assembly	10
Arduino Nano board	10
SPI TFT display	11
Camera	12
Fully assembled	13
Power USB port	14
Programming USB port	15
3. Examples	17
3.1 Draw Text and Graphics	17
3.2 Work with Touchscreen	25
3.3 Logic game “Columns”	31
3.4 Camera Initialization	41
3.5 View photos and RGB video on TFT SPI Display	...
3.6 Camera shot photos and save to microSD	...

1. Overview

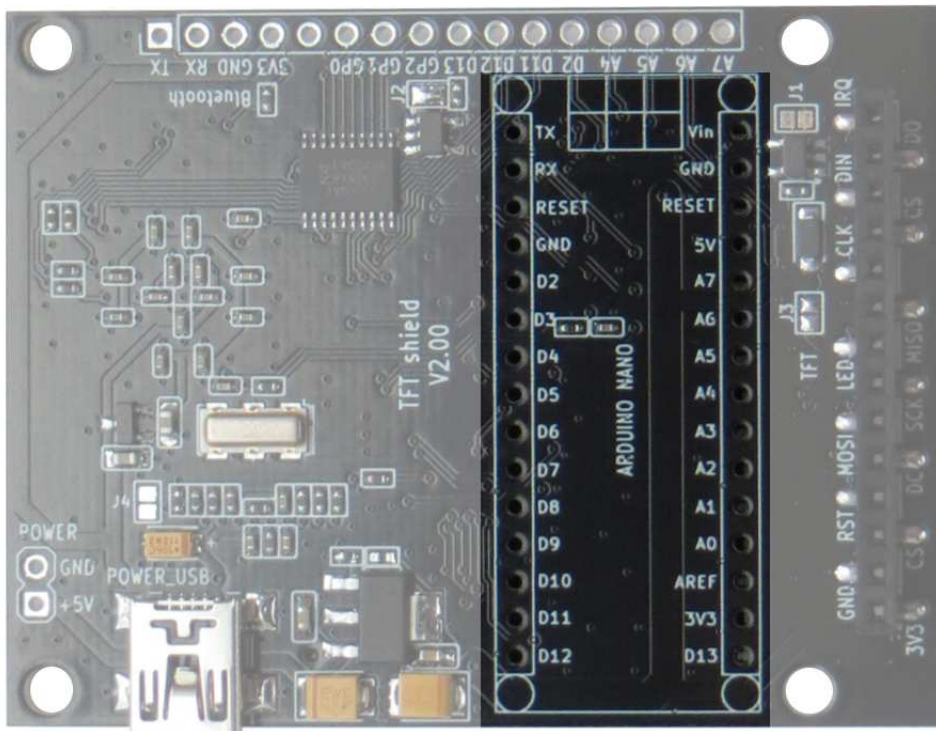


The Arduino Nano TFT Shield is designed to connect several devices to the Arduino Nano, such as:

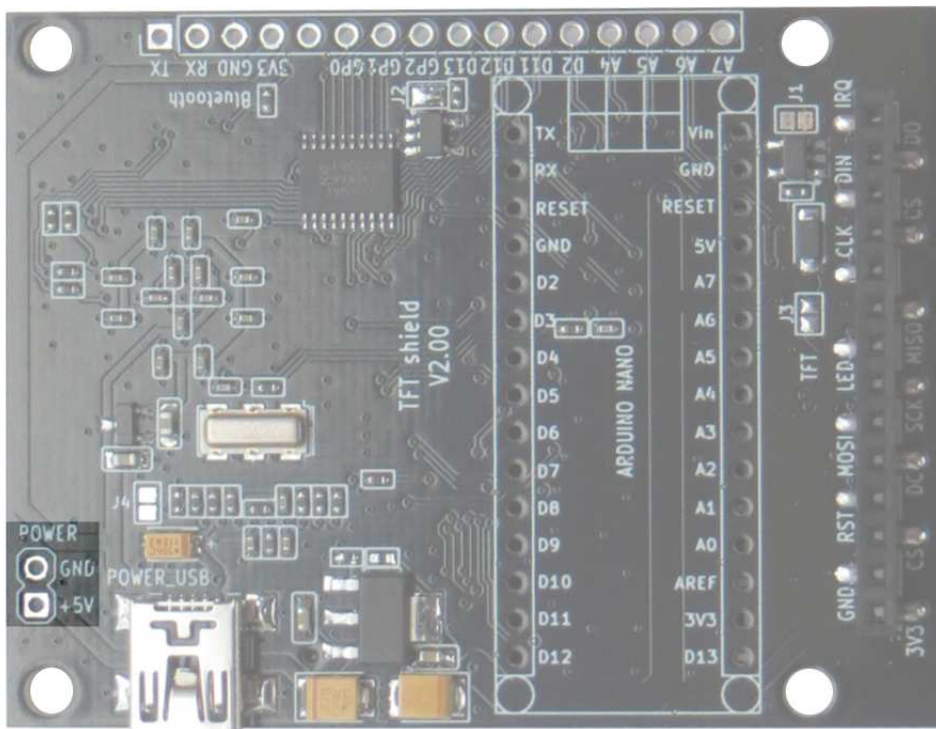
- SPI TFT display (320x240, 2.4 ", 2.8", 3.2 ");
- microSD card;
- Camera (e.g. OV7670, OV2640 etc.);
- Bluetooth module HC-05 (06);

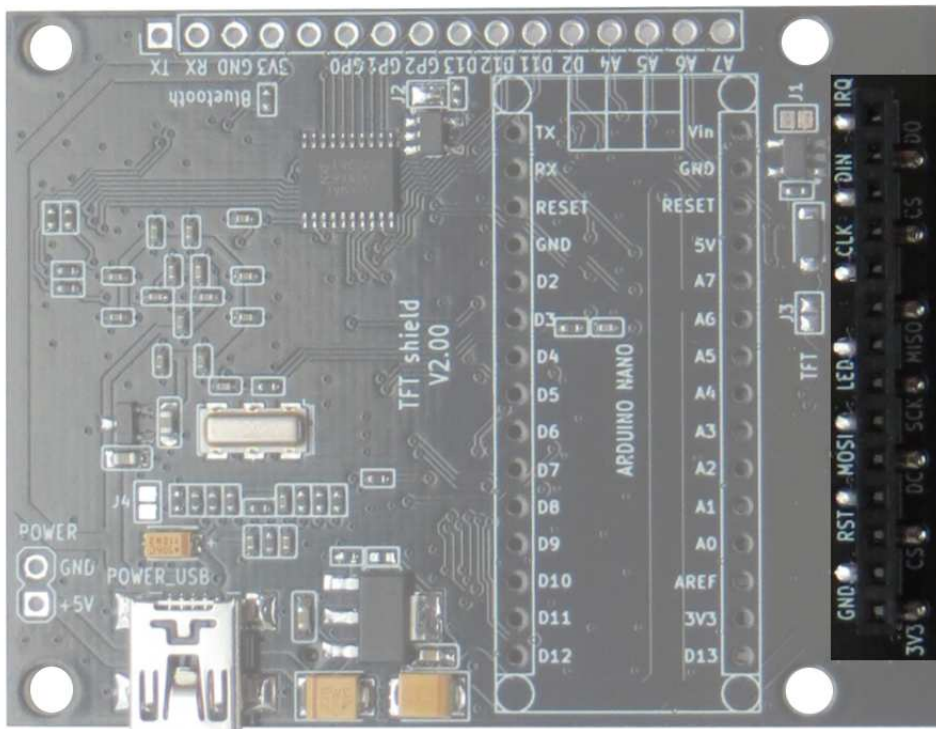
The following connectors are located on the board:

Arduino Nano connector:



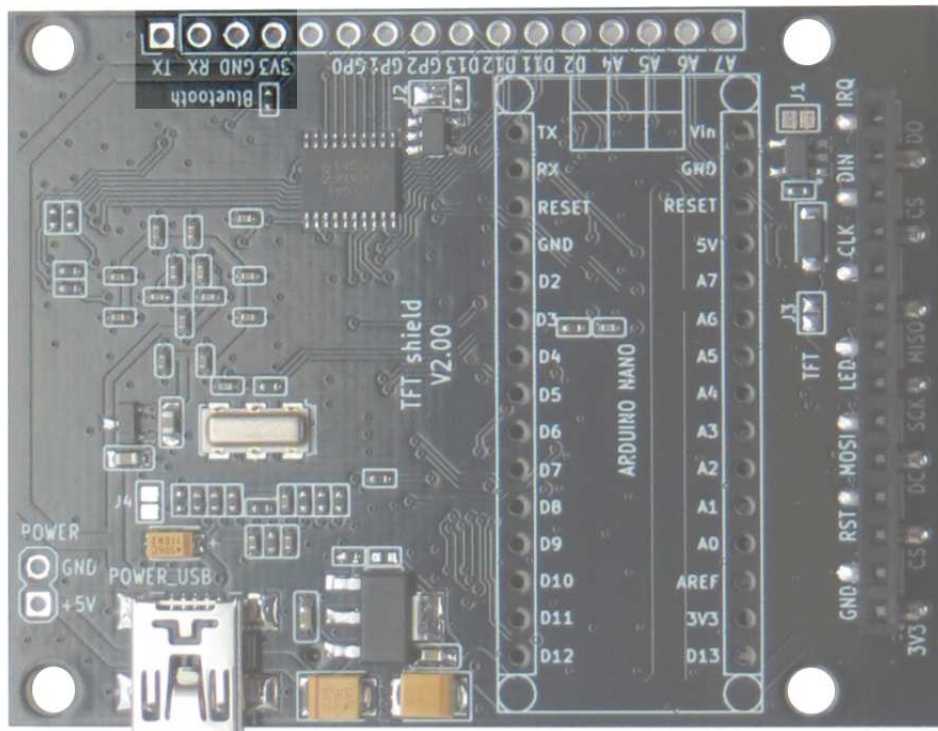
5V power connector:



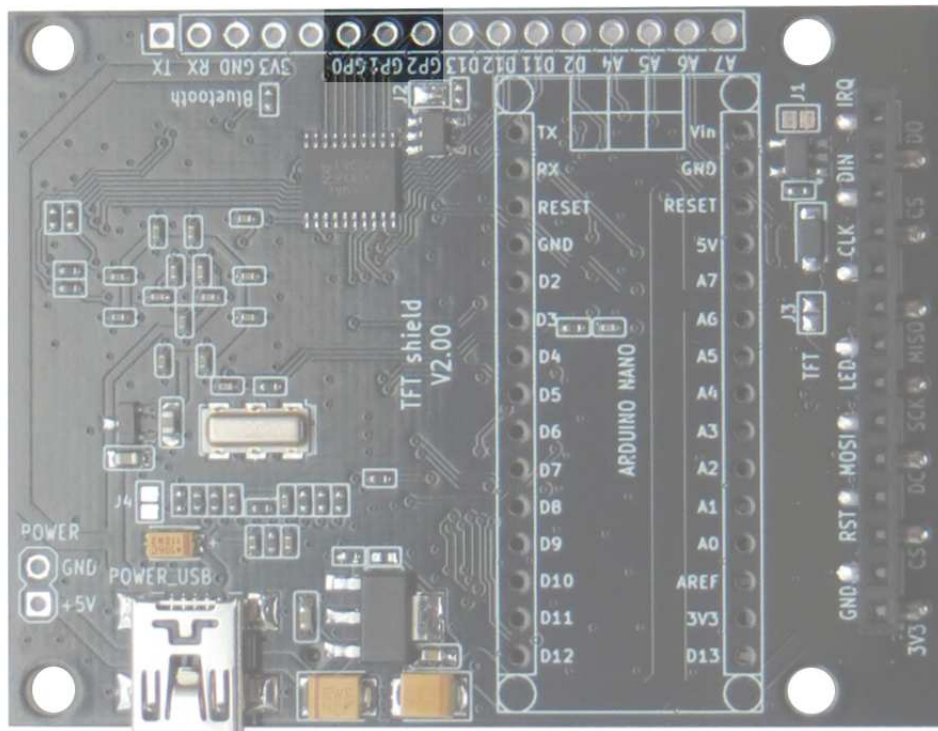


5

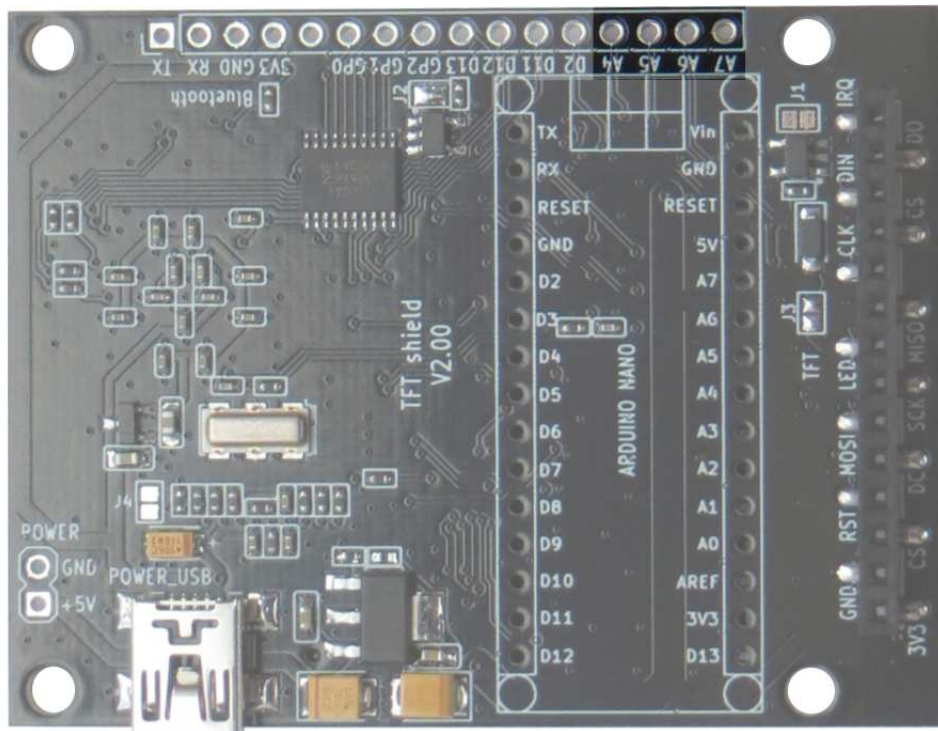
Bluetooth connector and internal 3V power:



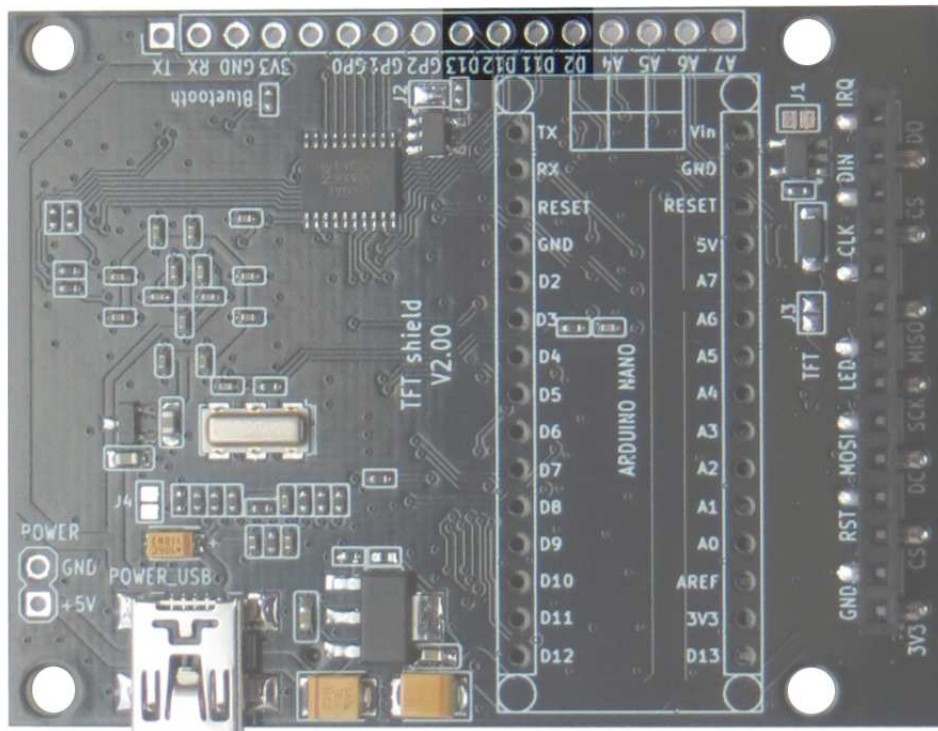
GPIO connector (3V level):



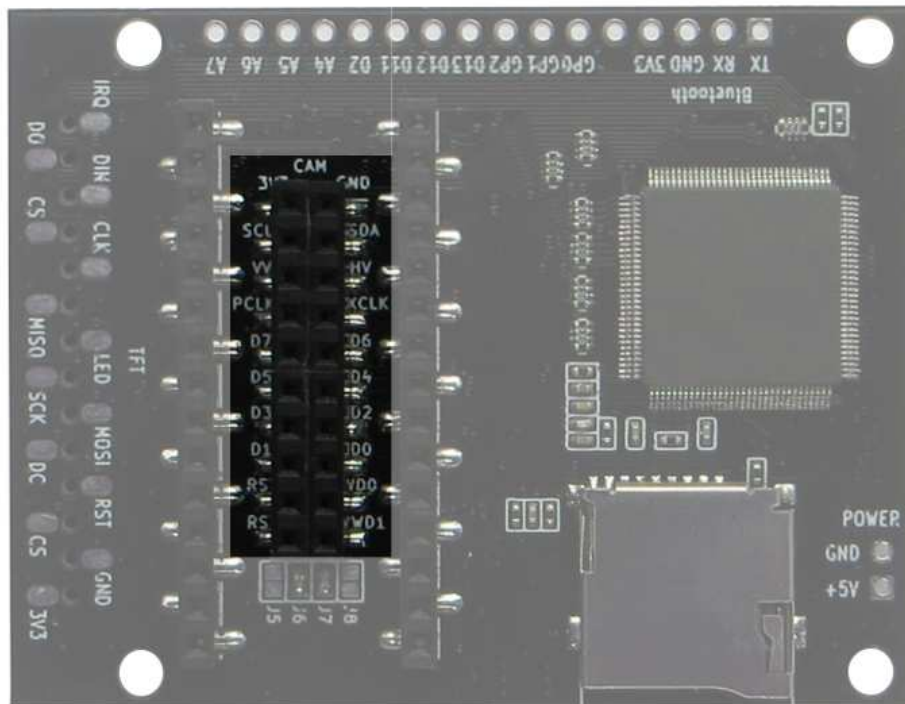
Analog Inputs connector:



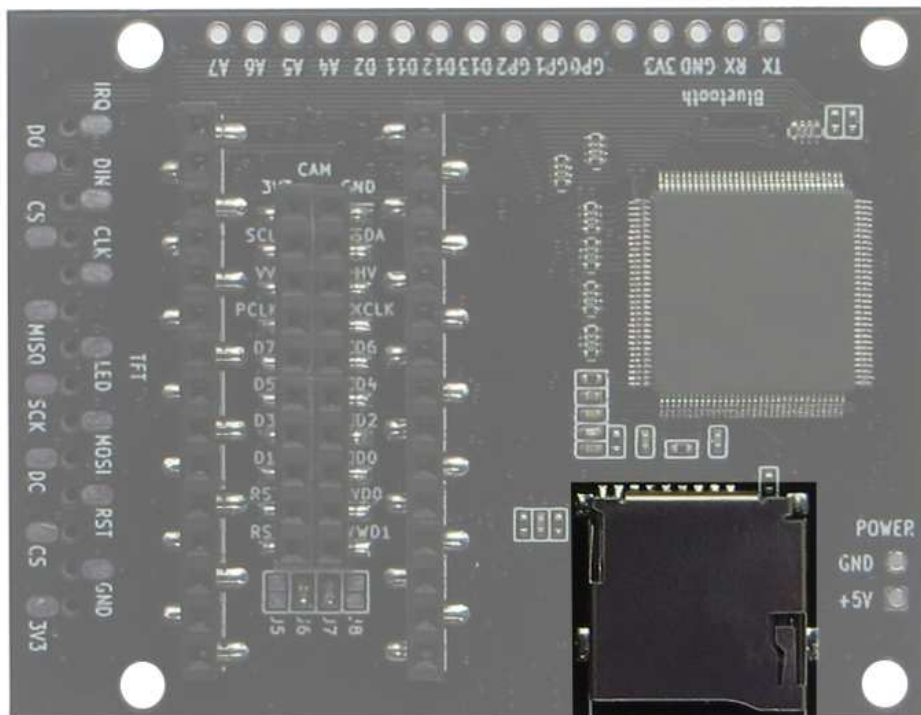
Digital I/O connector:



Camera Connector:



MicroSD Connector:



2. Quickstart

2.1 Preparation for Work

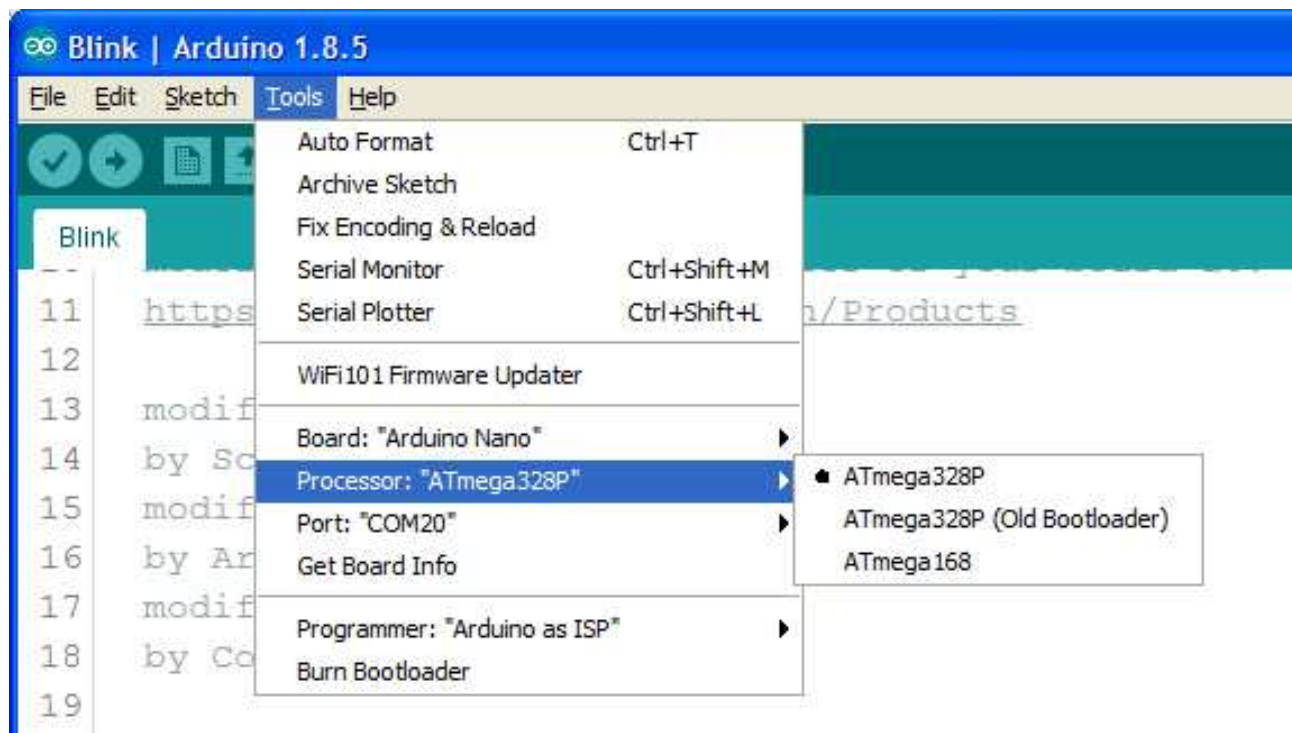
2.1.1 Installing Libraries

To install the libraries you need to download files from the https://github.com/YATFT/TFT_Nano repository and unzip them into the folder /Arduino/libraries/.

To work with text, also install the library "Adafruit_GFX" (from Adafruit) from the repository <https://github.com/adafruit/Adafruit-GFX-Library>.

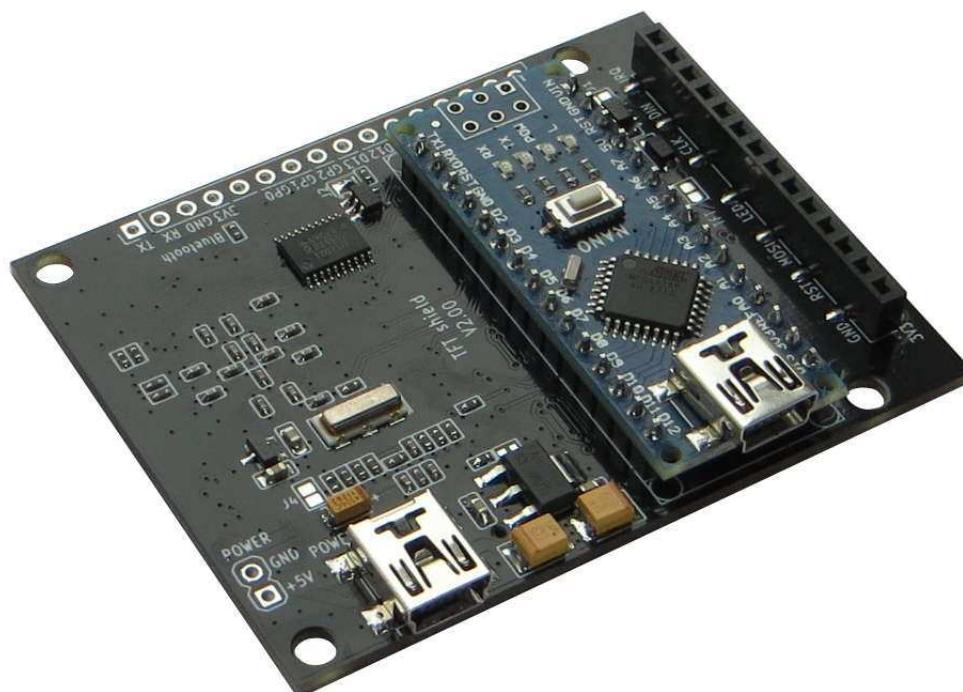
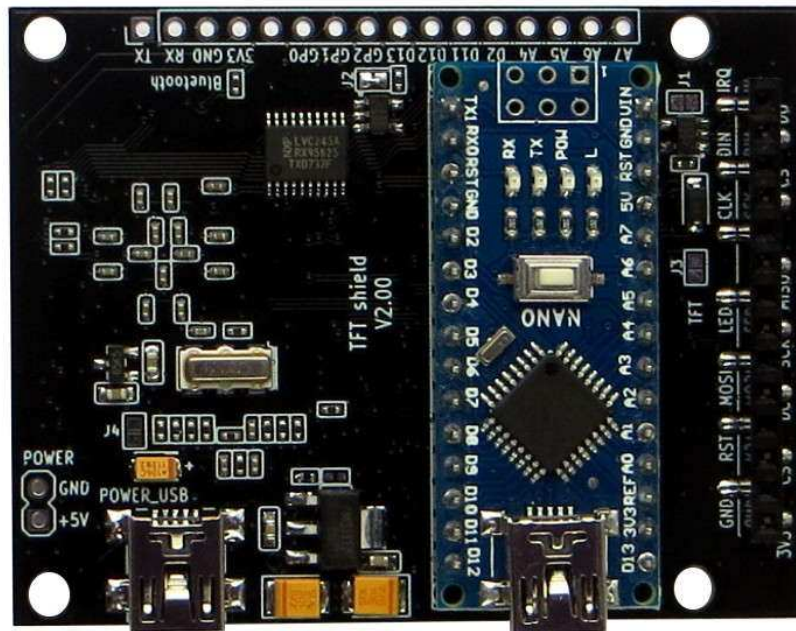
To program a sketch, you must run Arduino IDE.

In the "Tools" menu select a board *Arduino Nano*:

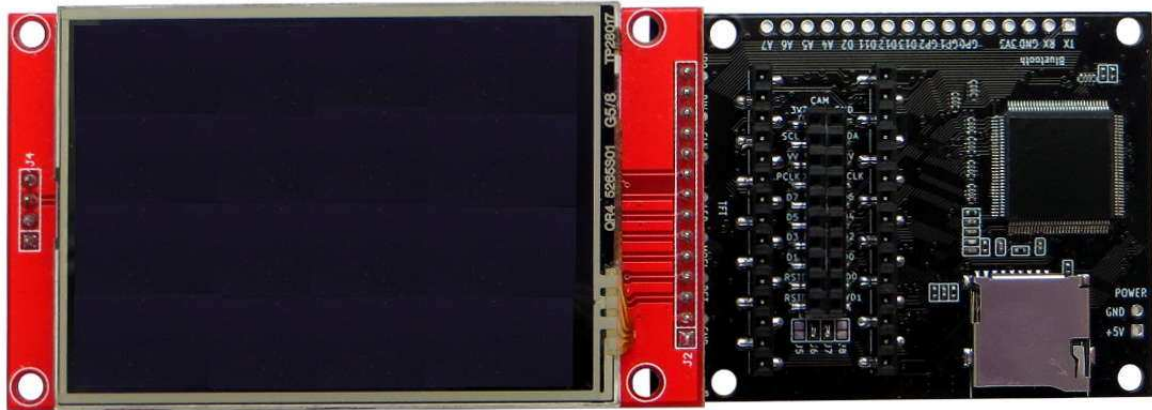


2.1.2 Device assembly

Install the Arduino Nano board on the adapter board, as shown in the figure:



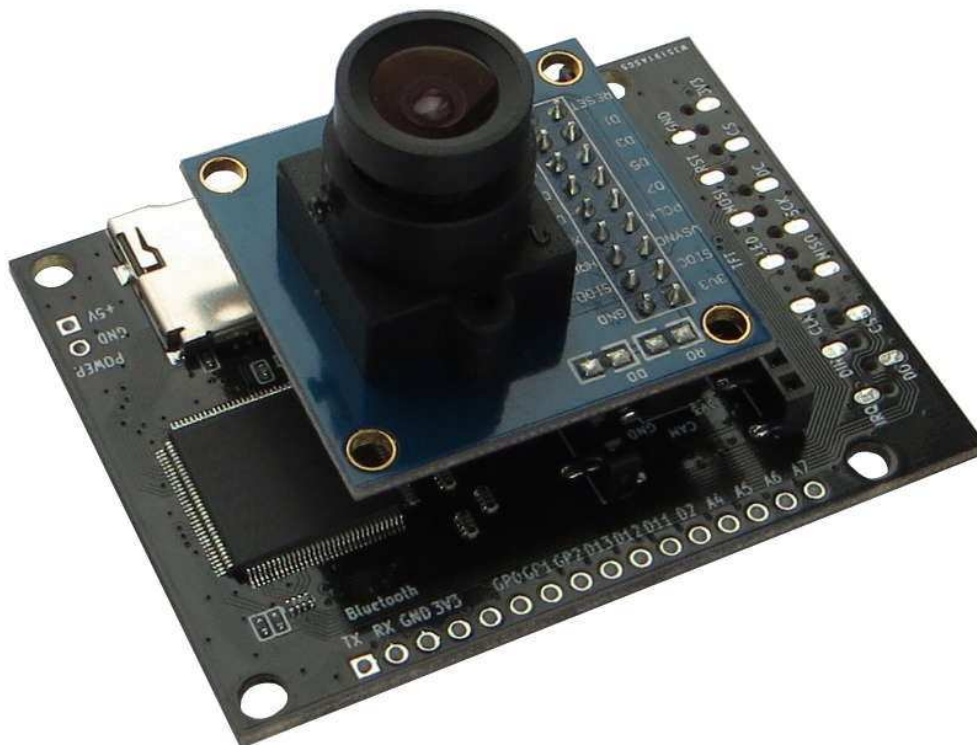
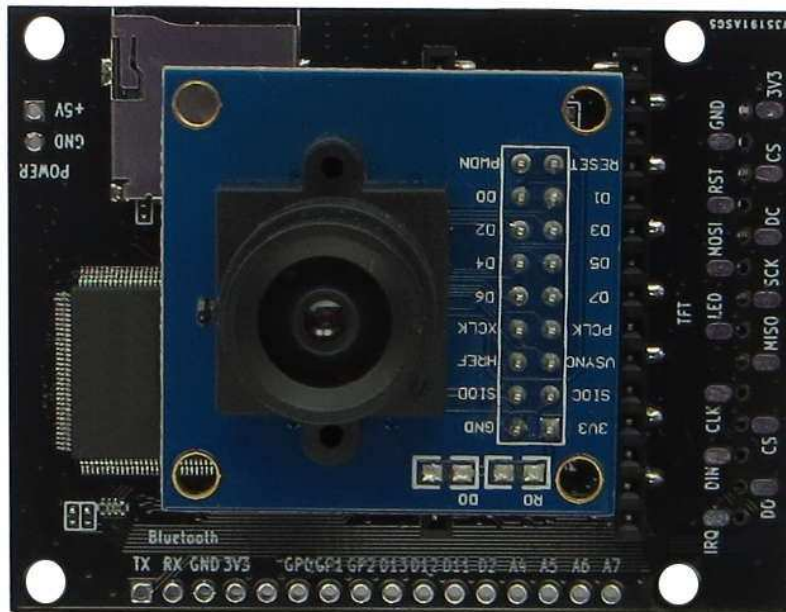
The display can be installed in two ways (as you wish). First way:



The second way to install the display:

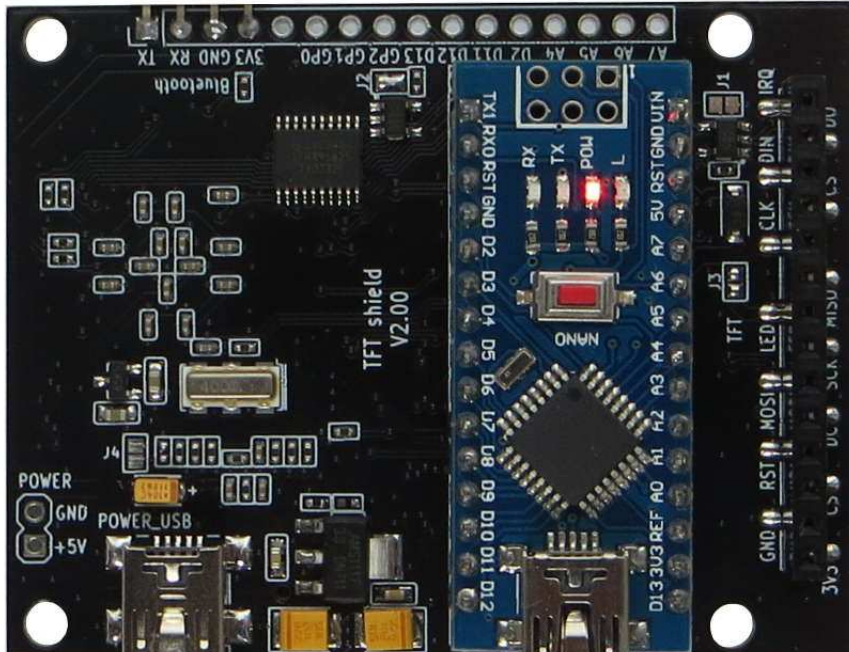


Then (if necessary) you can install the camera, as shown in the figure:

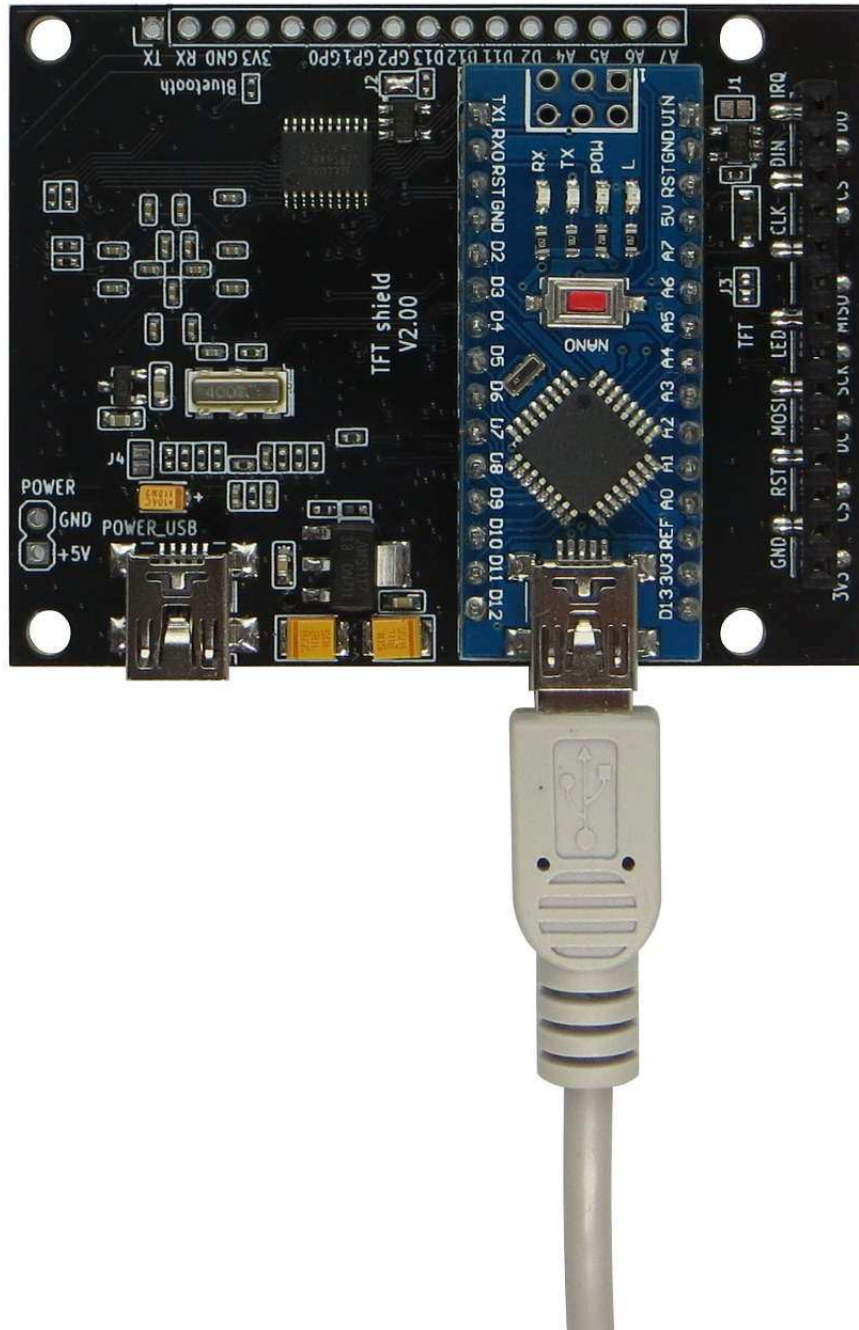


A fully assembled device looks like this:

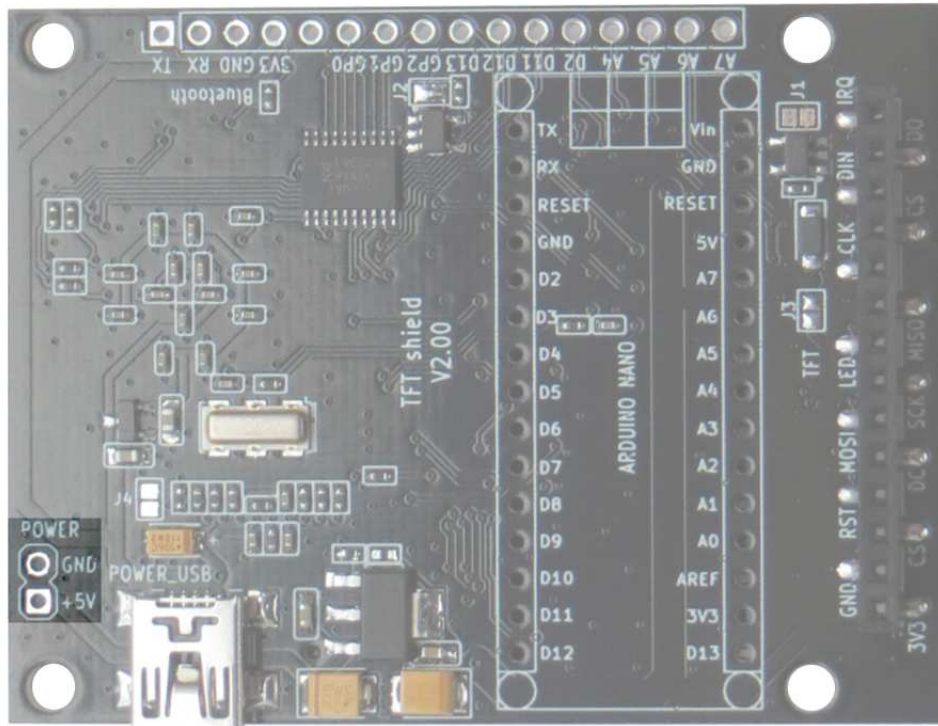




The other for programming:



The current from the USB port is enough to power the device along with the display. Also, 5V power can be supplied to a special power connector:



3. Examples

3.1 Draw Text and Graphics

Open sketch *Example_01.ino*:

```
/*
 * Arduino TFT_shield Example_01
 *
 * Author: Kamil Lee (2019)
 *
 * Comments:
 * This sketch demonstrate text & graphics.
 * Additionally need Adafruit-GFX library
 */

#include <YATFT2.h>           // Hardware-specific library
#include <Adafruit_GFX.h>     // Include Adafruit-GFX library
#include <Fonts/FreeSerif9pt7b.h> // Include Adafruit fonts
#include <Fonts/FreeSerifItalic24pt7b.h>
#include <Fonts/FreeSans24pt7b.h>

YATFT tft(0);

uint32_t total_time;
uint16_t Color[4] = {BRIGHTBLUE, BRIGHTGREEN, BRIGHTRED, BRIGHTYELLOW};
uint16_t Gray[7] = {GRAY0, GRAY1, GRAY2, GRAY3, GRAY4, GRAY5, GRAY6};

/*****/
void ClearScreen (void)
{
    tft.SetColor(BLACK); // Set fone color
    tft.ClearDevice();    // Fill all screen
}

void setup()
{
    Serial.begin(115200); // initialize the serial port
    Serial.println("Arduino TFT_shield Example 1!");
    tft.begin();          // initialize the display
}

void loop()
{
    uint16_t x, y, x2, y2, mask_gray;
```

```

uint16_t i;

ClearScreen();

// Fonts
Serial.print("1) View Fonts (");
total_time = millis();
tft.SetColor(BRIGHTBLUE);
tft.SetFont(NULL);
tft.OutTextXY(5, 5, "Demonstration of work with the TFT display.");
tft.SetColor(BRIGHTGREEN);
tft.SetFont(&FreeSerif9pt7b);
tft.OutTextXY(5, 20, "The example uses fonts from Adafruit.");
tft.SetFont(&FreeSerifItalic24pt7b);
tft.SetColor(BRIGHTCYAN);
tft.OutTextXY(5, 45, "3,5");
tft.SetColor(BRIGHTRED);
tft.OutTextXY(90, 45, "QVGA");
tft.SetColor(BRIGHTMAGENTA);
tft.OutTextXY(230, 45, "disp.");
tft.SetColor(BRIGHTYELLOW);
tft.SetFont(&FreeSans24pt7b);
tft.OutTextXY(5, 100, "A R D U I N O   +   T F T");

tft.SetFont(NULL);
for (i = 0; i < 7; i++)
{
    tft.SetColor(Gray[i]);
    tft.OutTextXY(5, 170+10*i, "Demonstration of work with the TFT display.");
}

PrintTotalTime();

delay(3000);
ClearScreen();

// Circle
Serial.print("2) Draw circle (");
total_time = millis();
tft.SetColor(BRIGHTRED);
for (i = 10; i < GetMaxY()>>1; i += 10) {
    tft.DrawCirc(GetMaxX()>>1, GetMaxY()>>1, i);
}

PrintTotalTime();

delay(1000);

```

```

// DrawFillCircle & DrawFillRect
Serial.print("3) Draw FillCircle and FillRect (");
total_time = millis();
tft.SetColor(BRIGHTRED);
tft.DrawFillCirc(GetMaxX()>>1,GetMaxY()>>1,110);
tft.SetColor(BRIGHTCYAN);
tft.DrawFillRect(GetMaxX()/2-77,GetMaxY()/2-77, GetMaxX()/2+77,GetMaxY()/2+77);
tft.SetColor(BRIGHTGREEN);
tft.DrawFillCirc(GetMaxX()>>1,GetMaxY()>>1,77);
tft.SetColor(BRIGHTMAGENTA);
tft.DrawFillRect(GetMaxX()/2-54,GetMaxY()/2-54, GetMaxX()/2+54,GetMaxY()/2+54);
tft.SetColor(BRIGHTBLUE);
tft.DrawFillCirc(GetMaxX()>>1,GetMaxY()>>1,54);
tft.SetColor(BRIGHTYELLOW);
tft.DrawFillRect(GetMaxX()/2-37,GetMaxY()/2-37, GetMaxX()/2+37,GetMaxY()/2+37);

PrintTotalTime();

delay(1000);
ClearScreen();

// Arc
Serial.print("4) Draw Arc (");
total_time = millis();
ClearScreen();
tft.SetColor(BRIGHTBLUE);
tft.DrawArc((GetMaxX()>>1)-60,(GetMaxY()>>1)-60,(GetMaxX()>>1)+60,
(GetMaxY()>>1)+60,20,30,0xFF);
tft.SetColor(BRIGHTGREEN);
tft.DrawArc((GetMaxX()>>1)-40,(GetMaxY()>>1)-40,(GetMaxX()>>1)+40,
(GetMaxY()>>1)+40,20,30,0xFF);
tft.SetColor(BRIGHTRED);
tft.DrawArc((GetMaxX()>>1)-20,(GetMaxY()>>1)-20,(GetMaxX()>>1)+20,
(GetMaxY()>>1)+20,20,30,0xFF);

PrintTotalTime();

delay(1000);

Serial.print("5) Draw FillBevel (");
total_time = millis();
tft.SetColor(BRIGHTBLUE);
tft.DrawFillBevel((GetMaxX()>>1)-60,(GetMaxY()>>1)-60,(GetMaxX()>>1)+60,
(GetMaxY()>>1)+60,30);
tft.SetColor(BRIGHTGREEN);
tft.DrawFillBevel((GetMaxX()>>1)-40,(GetMaxY()>>1)-40,(GetMaxX()>>1)+40,
(GetMaxY()>>1)+40,30);
tft.SetColor(BRIGHTRED);

```

```

    tft.DrawFillBevel((GetMaxX()>>1)-20,(GetMaxY()>>1)-20,(GetMaxX()>>1)+20,
(GetMaxY()>>1)+20,30);

    PrintTotalTime();

    delay(1000);
    ClearScreen();

    Serial.print("6) Draw Arc (");
    total_time = millis();
    for (i = 0; i < 4; i++) {
        tft.SetColor(Color[i]);
        tft.DrawArc((GetMaxX()>>1),(GetMaxY()>>1)-50,(GetMaxX()>>1),
(GetMaxY()>>1)+50,50,60,0x11<<i);
    }
    for (i = 0; i < 4; i++) {
        tft.SetColor(Color[i]);
        tft.DrawArc((GetMaxX()>>1),(GetMaxY()>>1)-30,(GetMaxX()>>1),
(GetMaxY()>>1)+30,35,45,0x11<<i);
    }
    for (i = 0; i < 4; i++) {
        tft.SetColor(Color[i]);
        tft.DrawArc((GetMaxX()>>1),(GetMaxY()>>1),(GetMaxX()>>1),
(GetMaxY()>>1),20,30,0x11<<i);
    }

    PrintTotalTime();

    delay(1000);
    ClearScreen();

    // Draw 1000 random lines
    Serial.print("7) Draw 1000 random lines (");
    total_time = millis();
    for (i = 0; i < 1000; i++) {
        tft.SetColor(random(65535));
        x = random(GetMaxX());
        y = random(GetMaxY());
        x2 = random(GetMaxX());
        y2 = random(GetMaxY());
        tft.DrawLine(x, y, x2, y2);
    }

    PrintTotalTime();

    delay(1000);

    // ReDraw Fill Screen

```

```

Serial.println("8) ReDraw 10 Fill Screen:");
RedrawFillScreen(BLACK);
RedrawFillScreen(BRIGHTBLUE);
RedrawFillScreen(YELLOW);
RedrawFillScreen(BRIGHTGREEN);
RedrawFillScreen(BRIGHTRED);
RedrawFillScreen(BRIGHTCYAN);
RedrawFillScreen(GREEN);
RedrawFillScreen(BRIGHTYELLOW);
RedrawFillScreen(BRIGHTCYAN);
RedrawFillScreen(BLACK);
}

void PrintTotalTime (void)
{
    total_time = millis() - total_time;
    Serial.print(total_time);
    Serial.println(" ms");
}

void RedrawFillScreen (uint16_t color)
{
    // ReDraw Fill Screen
    Serial.print("ReDraw Fill Screen (");
    total_time = millis();

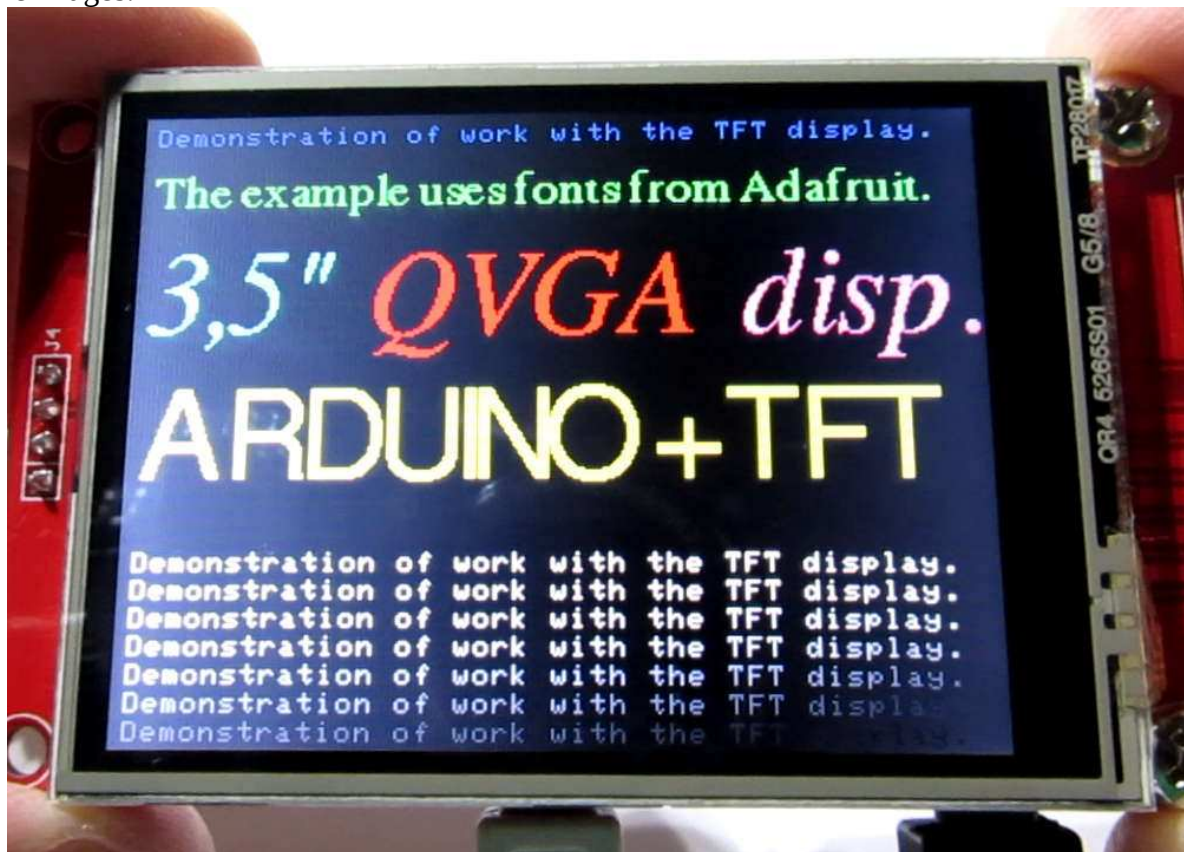
    tft.SetColor(color); // Set fone color
    tft.ClearDevice();    // Fill all screen

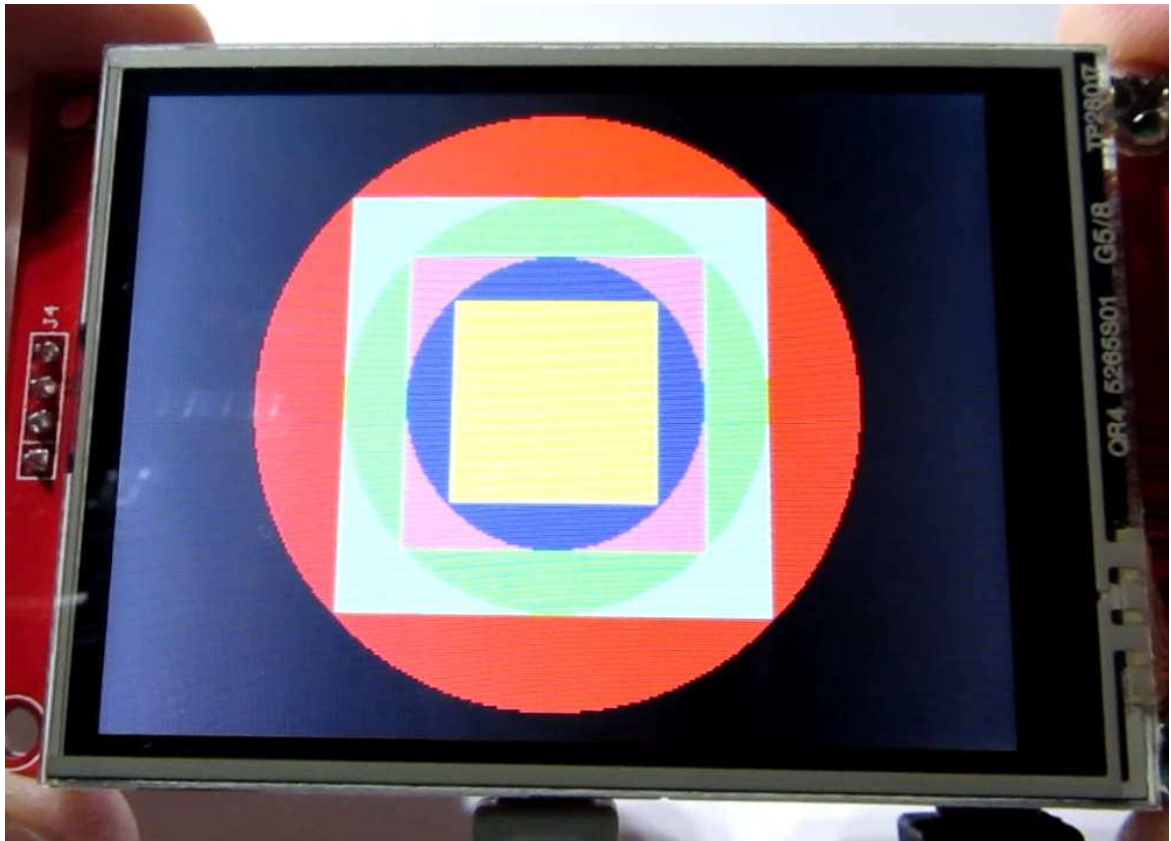
    PrintTotalTime();

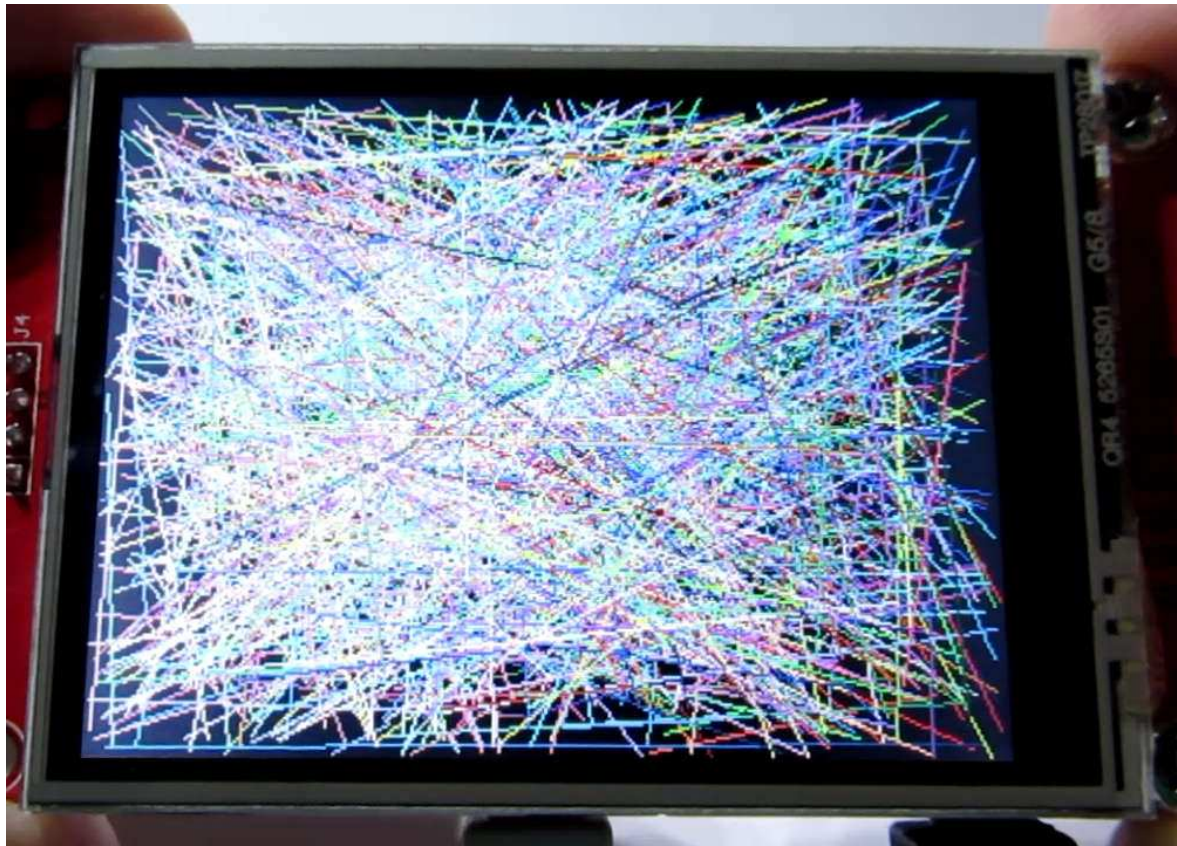
    delay(1000);
}

```

Some images:







3.2 Work with Touchscreen

Open sketch *Example_02.ino*:

```
/*
 * Arduino TFT_shield Example_02
 *
 * Author: Kamil Lee (2018)
 *
 * Comments:
 * This sketch demonstrate text & graphics.
 * Additionally need Adafruit-GFX, XPT2046_Touchscreen, SPI library's
 */

/*****
 * SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT
 * WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING
 * WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY, TITLE,
 * NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE.
 * IN NO EVENT SHALL AUTHOR OR ITS LICENSORS BE LIABLE OR
 * OBLIGATED UNDER CONTRACT, NEGLIGENCE, STRICT LIABILITY,
 * CONTRIBUTION, BREACH OF WARRANTY, OR OTHER LEGAL EQUITABLE
 * THEORY ANY DIRECT OR INDIRECT DAMAGES OR EXPENSES INCLUDING BUT
 * NOT LIMITED TO ANY INCIDENTAL, SPECIAL, INDIRECT, PUNITIVE OR
 * CONSEQUENTIAL DAMAGES, LOST PROFITS OR LOST DATA, COST OF
 * PROCUREMENT OF SUBSTITUTE GOODS, TECHNOLOGY, SERVICES, OR ANY
 * CLAIMS BY THIRD PARTIES (INCLUDING BUT NOT LIMITED TO ANY DEFENSE
 * THEREOF), OR OTHER SIMILAR COSTS.
 *****/
#include <YATFT2.h>           // Hardware-specific library
#include <SPI.h>              // Include SPI library
#include <XPT2046_Touchscreen.h> // Include Touchscreen library
#include <Adafruit_GFX.h>     // Include Adafruit-GFX library
#include <Fonts/FreeSerif9pt7b.h> // Include Adafruit fonts
#include <Fonts/FreeSerifItalic24pt7b.h>
#include <Fonts/FreeSans24pt7b.h>

// Touchscreen: MOSI=11, MISO=12, SCK=13, CS=2
#define CS_PIN A0
XPT2046_Touchscreen ts(CS_PIN, 255);

YATFT tft(0);

#define Y_BAR_TOP (GetMaxY()-50)
#define Y_BAR_BOT GetMaxY()

/*
```

If using the shield, all control and data lines are fixed, and a simpler declaration can optionally be used:

```
*/
uint16_t pos_x[] = {0,0,0,0};
uint16_t pos_y[] = {0,0,0,0};
uint8_t pos_x_cnt = 0;
uint8_t pos_y_cnt = 0;
uint16_t pos_x_mid = 0;
uint16_t pos_y_mid = 0;
uint16_t color_paint = WHITE;
uint8_t buttons = 0;
uint16_t Color[4] = {BRIGHTBLUE, BRIGHTGREEN, BRIGHTRED, BRIGHTYELLOW};
uint16_t Gray[7] = {GRAY0, GRAY1, GRAY2, GRAY3, GRAY4, GRAY5, GRAY6};

/*****/
void ClearScreen (void)
{
    tft.SetColor(BLACK); // Set fone color
    tft.ClearDevice();   // Fill all screen
}

void setup()
{
    Serial.begin(115200); // initialize the serial port
    Serial.println("Arduino TFT_shield Example1!");
    ts.begin();           // Init Touchscreen
    SPI.end();            // Disable SPI for correct work DB2 (SS) pin
    tft.begin();          // initialize the display

    RefreshWindow();
}

void loop()
{
    uint16_t x, y;

    // Touch
    // When the SS pin is set as OUTPUT, it can be used as
    // a general purpose output port (it doesn't influence
    // SPI operations).
    SPI.begin();
    if (ts.touched())
    {
        TS_Point p = ts.getPoint();

        Serial.print(F("Pressure = "));
        Serial.print(p.z);
        Serial.print(F(", x = "));
```

```

Serial.print(p.x);
Serial.print(F(", y = "));
Serial.print(p.y);
Serial.println();
delay(3); // Delay for filtering

SPI.end(); // Disable SPI for correct work DB2 (SS) pin

// Calculate coordinates x, y from code ADC
Serial.print(F("Pressure = "));
Serial.print(p.z);
Serial.print(F(", x = "));
Serial.print(p.x);
Serial.print(F(", "));
Serial.print(x);
Serial.print(F(", y = "));
Serial.print(p.y);
Serial.print(F(", "));
Serial.print(y);
Serial.println();

if (p.x < 350) p.x = 350;
if (p.x > 3850) p.x = 3850;
if (p.y < 250) p.y = 250;
if (p.y > 3850) p.y = 3850;

x = (uint16_t)(320L - ((uint32_t)p.x - 350L)*100L/1094L);
y = (uint16_t)(240L - ((uint32_t)p.y - 250L)*100L/1510L);

// Filtering
pos_x_mid = (pos_x[0] + pos_x[1] + pos_x[2] + pos_x[3])/4;
pos_y_mid = (pos_y[0] + pos_y[1] + pos_y[2] + pos_y[3])/4;
pos_x[pos_x_cnt++] = x;
pos_y[pos_y_cnt++] = y;
pos_x_cnt &= 0x03;
pos_y_cnt &= 0x03;

if (x > (pos_x_mid - 10) && x < (pos_x_mid + 10) && y > (pos_y_mid - 10) && y <
(pos_y_mid + 10)) {

    if (y > Y_BAR_TOP && y < Y_BAR_BOT) {
        if (x < 1*(GetMaxX()+1)/5) { // Touch Bar 1
            color_paint = Color[0];
            RefreshTitle();
        } else
        if (x < 2*(GetMaxX()+1)/5) { // Touch Bar 2
            color_paint = Color[1];
            RefreshTitle();
        }
    }
}

```

```

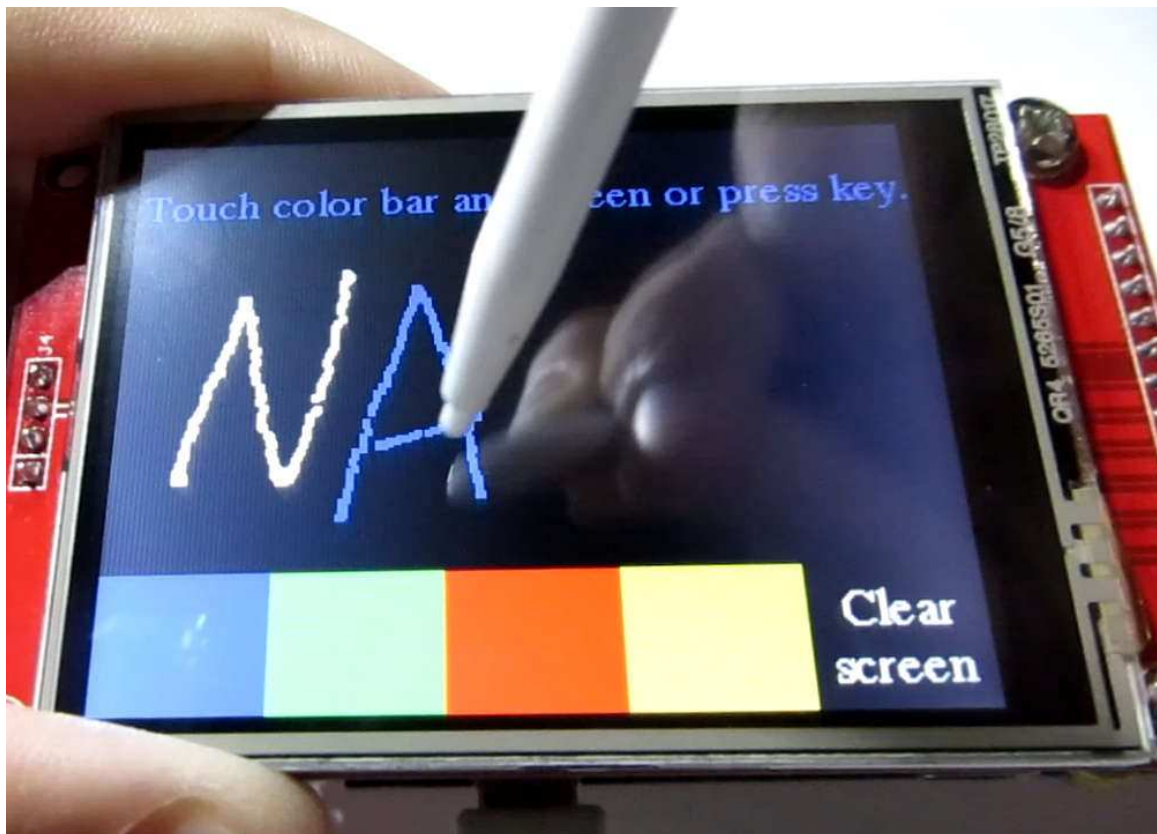
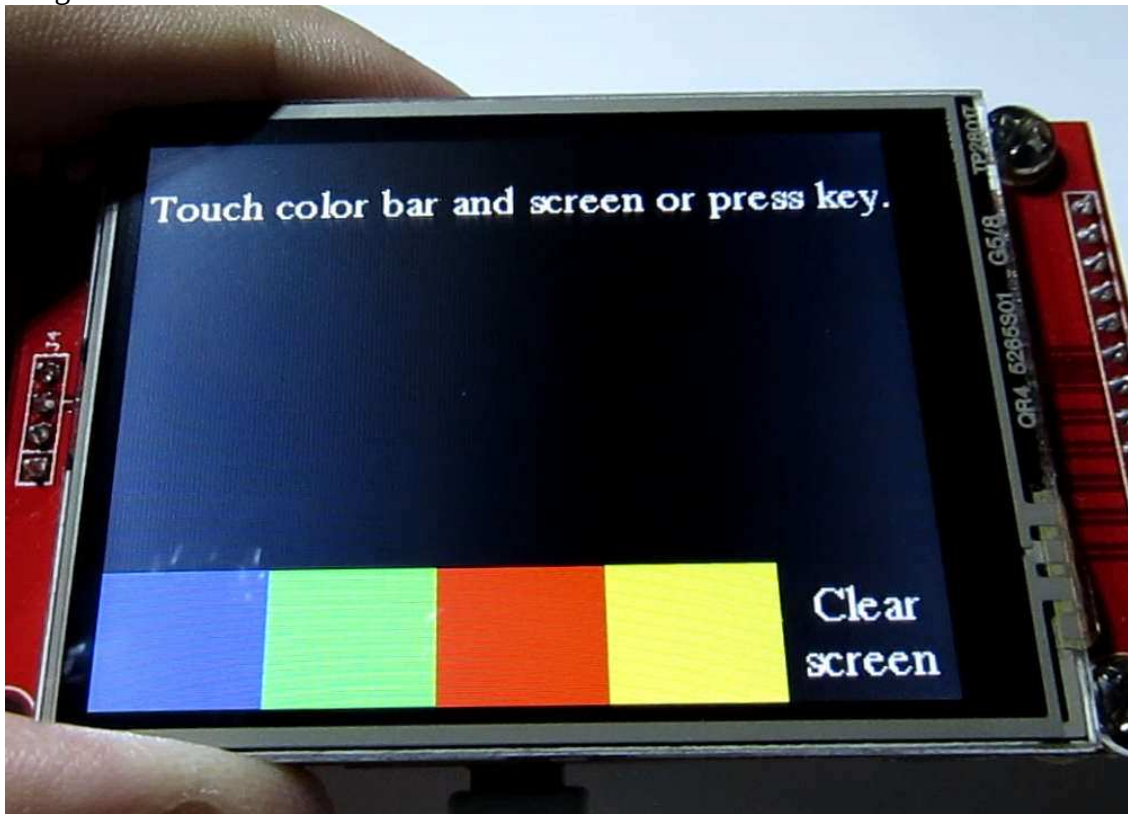
    } else
    if (x < 3*(GetMaxX()+1)/5) { // Touch Bar 3
        color_paint = Color[2];
        RefreshTitle();
    } else
    if (x < 4*(GetMaxX()+1)/5) { // Touch Bar 4
        color_paint = Color[3];
        RefreshTitle();
    } else { // Clear screen
        RefreshWindow();
    }
} else {
    tft.SetColor(color_paint);
    tft.DrawFillRect(x-1, y-1, x+1, y+1);
}
}
}
SPI.end(); // Disable SPI for correct work DB2 (SS) pin
}

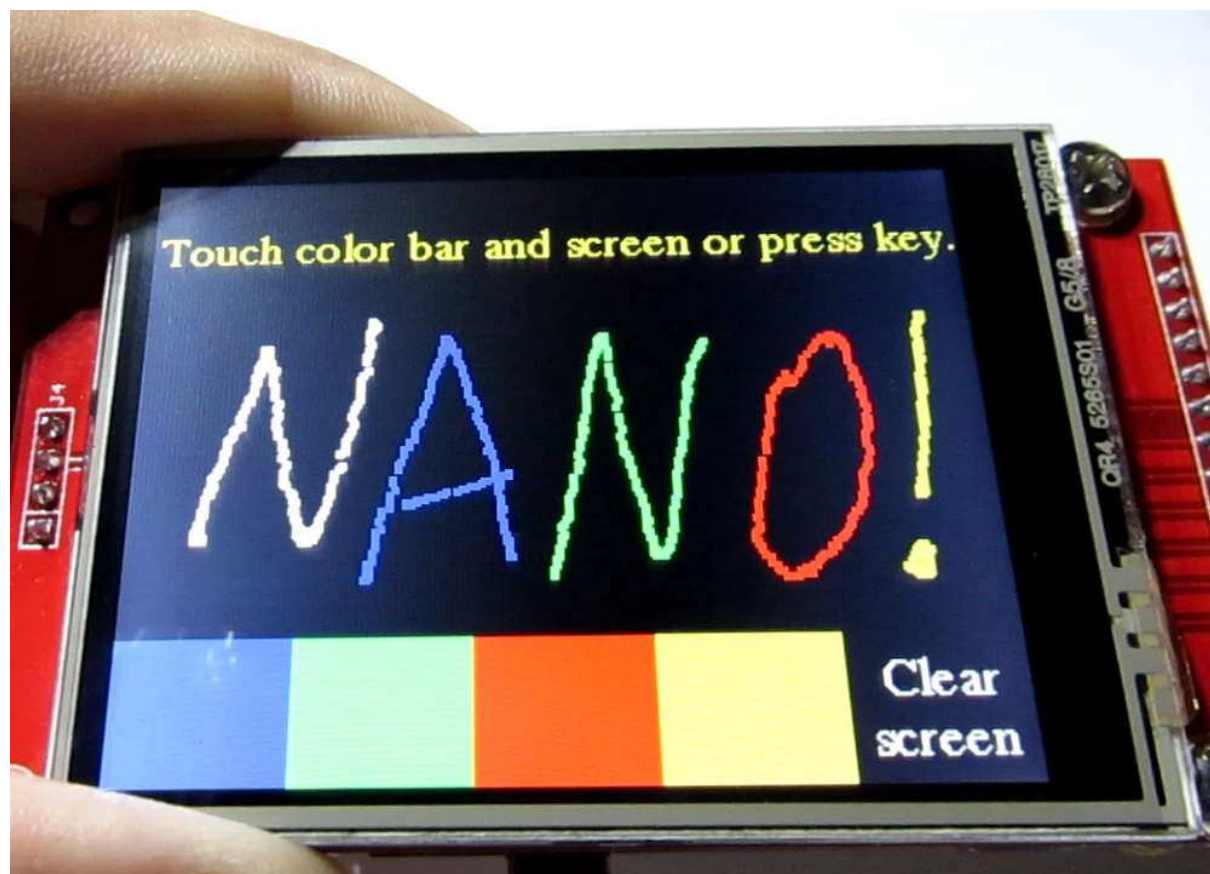
void RefreshWindow(void)
{
    color_paint = WHITE;
    ClearScreen();
    for (uint8_t i = 0; i < 4; i++) {
        tft.SetColor(Color[i]);
        tft.DrawFillRect((i+1)*((GetMaxX()+1)/5), Y_BAR_TOP, (i)*((GetMaxX()+1)/5),
Y_BAR_BOT);
    }
    RefreshTitle();
    tft.SetColor(WHITE);
    tft.OutTextXY(GetMaxX() - 50, GetMaxY() - 45, "Clear");
    tft.OutTextXY(GetMaxX() - 55, GetMaxY() - 25, "screen");
}

void RefreshTitle(void)
{
    tft.SetColor(color_paint);
    tft.SetFont(&FreeSerif9pt7b);
    tft.OutTextXY(3, 20, "Touch color bar and screen or press key.");
}

```

Some images:





3.3 Logic game “Columns”

```
/******
 * SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
 * KIND, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY WARRANTY
 * OF MERCHANTABILITY, TITLE, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR
 * PURPOSE. IN NO EVENT SHALL AUTHOR OR ITS LICENSORS BE LIABLE OR
 * OBLIGATED UNDER CONTRACT, NEGLIGENCE, STRICT LIABILITY, CONTRIBUTION,
 * BREACH OF WARRANTY, OR OTHER LEGAL EQUITABLE THEORY ANY DIRECT OR INDIRECT
 * DAMAGES OR EXPENSES INCLUDING BUT NOT LIMITED TO ANY INCIDENTAL, SPECIAL,
 * INDIRECT, PUNITIVE OR CONSEQUENTIAL DAMAGES, LOST PROFITS OR LOST DATA,
 * COST OF PROCUREMENT OF SUBSTITUTE GOODS, TECHNOLOGY, SERVICES, OR ANY
 * CLAIMS BY THIRD PARTIES (INCLUDING BUT NOT LIMITED TO ANY DEFENSE THEREOF),
 * OR OTHER SIMILAR COSTS.
 *****/
#include <YATFT2.h>
#include <XPT2046_Touchscreen.h>
#include <SPI.h>
#include <Adafruit_GFX.h> // Core graphics library
#include <Fonts/FreeMonoBoldOblique12pt7b.h>
#include <Fonts/FreeSerif9pt7b.h>
#define CS_PIN A0
// MOSI=11, MIS0=12, SCK=13
XPT2046_Touchscreen ts(CS_PIN, 255);
YATFT tft(0);
/*
  If using the shield, all control and data lines are fixed, and
  a simpler declaration can optionally be used:
*/
long randNumber;
#define KEY_EMPTY          0
#define KEY_SWAP           1
#define KEY_FALL           2
#define KEY_LEFT           3
#define KEY_RIGHT          4
#define DISPLAY_MAX_X 320
#define DISPLAY_MAX_Y 240
#define MaxCol             8
#define MaxRow             17
#define SmeX               3
#define SmeY               3
#define razmer             15
#define LL                 250
#define NumCol             6
#define MaxLevel           8
#define PeriodLevel       80
#define DISP_LEFT ((DISPLAY_MAX_X - MaxCol*razmer)/2 - 2)
#define DISP_RIGHT ((DISPLAY_MAX_X + MaxCol*razmer)/2 + 2)
#define DISP_TOP ((DISPLAY_MAX_Y - (MaxRow-4)*razmer)/2 - 2 - 10)
#define DISP_BOT ((DISPLAY_MAX_Y + (MaxRow-4)*razmer)/2 + 2 - 10)
uint8_t MasSt[MaxCol][MaxRow], MasTmp[MaxCol][MaxRow], MasOld[MaxCol][MaxRow],
fignext[3];
uint8_t Level=1, OldLevel, tr, flfirst=1;
uint16_t MasCol[]={WHITE, BLACK, BRIGHTRED, BRIGHTBLUE, BRIGHTGREEN,
BRIGHTYELLOW, BRIGHTMAGENTA, BRIGHTCYAN};
unsigned long Counter,Score=0, TScore=0, Record=0, OldRecord, OldScore,
myrecord;
```

```

uint16_t tempspeed;
bool    fl, Demo=true, myfl=false, Arbeiten=false, FlNew, FlZ=false;
int8_t  VAL, Mp, x,y;
int8_t  mmm [4][2]={{-1,0},{0,-1},{1,0},{0,1}};
uint16_t MasSpeed[MaxLevel]={500,450,400,350,300,250,200,100};
/*****/
void  setup(void)
{
    // initialize the serial port
    Serial.begin(115200);
    Serial.println("TFT_shield_Game1 example!");
    ts.begin(); // Init Touchscreen
    SPI.end();
    tft.begin(); // initialize the display
    randomSeed(analogRead(5));
    tft.SetColor(WHITE);
    tft.ClearDevice();
    tft.SetColor(RED);
    tft.SetFont(&FreeSerif9pt7b);
    tft.OutTextXY( 20,  20, "LEVEL");
    tft.OutTextXY(240,  20, "NEXT");
    tft.OutTextXY( 20,  75, "SCORE");
    tft.OutTextXY( 25, 130, "TOP");
    FlNew = true;
    ViewStacan();
    GetNext();
    delay(100);
    tft.SetColor(BLACK);
    tft.DrawLine(DISP_LEFT + 1, DISP_TOP + 0, DISP_LEFT + 1, DISP_BOT - 1);
    tft.DrawLine(DISP_LEFT + razmer*MaxCol+5-MaxCol, DISP_TOP + 0, DISP_LEFT +
razmer*MaxCol+5-MaxCol, DISP_BOT - 1);
    tft.DrawLine(DISP_LEFT + 1, DISP_BOT - 1, DISP_LEFT + 1+razmer*MaxCol+5-
MaxCol-1, DISP_BOT - 1);
}
void  loop(void)
{
    if (Demo)  ProcDemo();
    else {
        if (Arbeiten) {
            mydelay(tempspeed);
            figmove(0,1);
        } else  if (mypush()==KEY_SWAP)  NewGame();
    }
}
uint8_t  mypush(void)
{
    unsigned long tpr = millis();
    uint8_t res = KEY_EMPTY;
    uint8_t button = 0;
    static uint8_t button_old;
    static uint8_t cnt = 0;
    button = Touch();
    if (button) {
        cnt = 5;
        Serial.print("Scan Button: ");
        Serial.print(button, HEX);
        Serial.println();
        if (button != button_old) {

```

```

        if (button & 0x01) res = KEY_SWAP;
        if (button & 0x02) res = KEY_FALL;
        if (button & 0x04) {};
        if (button & 0x08) res = KEY_LEFT;
        if (button & 0x10) res = KEY_RIGHT;
        button_old = button;
    }
}
if (!cnt) {
    button_old = button;
} else {
    cnt--;
}
return(res);
}
void ViewQuad(uint8_t i,uint8_t j,uint8_t mycolor)
{
    if (j<3) return;
    uint16_t wy = DISP_TOP + SmeY + (j-3)*razmer - j;
    uint16_t wx = DISP_LEFT + SmeX + i*razmer - i;
    if (mycolor!=0) {
        tft.SetColor(BLACK);
        tft.DrawRect(wx, wy, wx+razmer-1, wy+razmer-1);
        tft.SetColor(MasCol[mycolor]);
        tft.DrawFillRect(wx+1, wy+1, wx+1+razmer-3, wy+1+razmer-3);
    } else {
        tft.SetColor(WHITE);
        tft.DrawFillRect(wx+1, wy+0, wx+1+razmer-3, wy+1+razmer-3);
    }
}
void ViewStacan(void)
{
    char myStr2[5];
    uint8_t h = tft.GetTextHeight(&FreeMonoBoldOblique12pt7b);
    tft.SetFont(&FreeMonoBoldOblique12pt7b);
    if (OldScore!=Score || FlNew) {
        sprintf(myStr2,"%05d",Score);
        int16_t w = tft.GetTextWidth(myStr2, &FreeMonoBoldOblique12pt7b);
        tft.SetColor(WHITE);
        tft.DrawFillRect(20,100,20+w+5,100+h);
        tft.SetColor(GREEN);
        tft.OutTextXY(20,100,myStr2);
        OldScore=Score;
    }
    if (OldRecord!=Record || FlNew) {
        sprintf(myStr2,"%05d",Record );
        int16_t w = tft.GetTextWidth(myStr2, &FreeMonoBoldOblique12pt7b);
        tft.SetColor(WHITE);
        tft.DrawFillRect(20,155,20+w+5,155+h);
        tft.SetColor(GREEN);
        tft.OutTextXY(20,155,myStr2);
        OldRecord=Record;
    }
    if (OldLevel!=Level || FlNew) {
        sprintf(myStr2,"%01d",Level );
        int16_t w = tft.GetTextWidth(myStr2, &FreeMonoBoldOblique12pt7b);
        tft.SetColor(WHITE);
        tft.DrawFillRect(25,45,25+w+5,45+h);
    }
}

```

```

        tft.SetColor(GREEN);
        tft.OutTextXY(25,45,myStr2);
        OldLevel=Level;
    }
    FlNew=false;
    for (byte j=3;j<MaxRow;j++)
        for (byte i=0;i<MaxCol;i++)
            if (MasSt[i][j]!=MasOld[i][j]) ViewQuad(i,j,MasSt[i][j]);
    for (byte j=3;j<MaxRow;j++)
        for (byte i=0;i<MaxCol;i++)
            MasOld[i][j]=MasSt[i][j];
}
void ClearMas(byte MasStx[MaxCol][MaxRow])
{
    for (byte j=0;j<MaxRow;j++)
        for (byte i=0;i<MaxCol;i++)
            (MasStx[i][j]=0);
}
void Sosed (int i,int j,int dx,int dy, byte mode)
{
    int nx=i+dx;
    int ny=j+dy;
    if (nx>=0 && ny>=0 && nx<MaxCol && ny<MaxRow && MasSt[nx][ny]==MasSt[i][j])
    {
        if (mode==1) MasTmp[i][j]++;
        else
            if (mode==2 && (MasTmp[nx][ny]>1 || MasTmp[i][j]>2 )) {
                MasTmp[nx][ny]=3;
                MasTmp[i][j]=3;
            } else
                if (mode==3 && MasTmp[nx][ny]==3) {
                    if (MasTmp[i][j]!=3) {
                        MasTmp[i][j]=3;
                        fl=true;
                    }
                }
    }
}
void Sos(int i,int j, byte mode)
{
    for (byte k=0;k<4;k++) Sosed(i,j,mmm[k][0],mmm[k][1],mode);
}
bool FindFull(void)
{
    byte i,j,k; bool res;
    res = false;
    for (byte k=2;k<8;k++) {
        ClearMas(MasTmp);
        for (j=3;j<MaxRow;j++)
            for (i=0;i<MaxCol;i++)
                if (MasSt[i][j]==k) Sos(i,j,1);
        for (j=3;j<MaxRow;j++)
            for (i=0;i<MaxCol;i++)
                if (MasTmp[i][j]>1) Sos(i,j,2);
        do {
            fl=false;
            for (j=3;j<MaxRow;j++)
                for (i=0;i<MaxCol;i++)

```

```

        if (MasTmp[i][j]>0) Sos(i,j,3);
    } while (f1);
    for (j=3;j<MaxRow;j++)
        for (i=0;i<MaxCol;i++)
            if (MasTmp[i][j]==3) {
                MasSt[i][j]=1;
                TScore++;
            }
    }
    if (TScore>0) {
        ViewStacan();
        FlZ=true;
        mydelay(500);
    }
    for (j=0;j<MaxRow;j++)
        for (i=0;i<MaxCol;i++) {
            while (MasSt[i][MaxRow-1-j]==1) {
                for (k=0;k<MaxRow-2-j;k++) MasSt[i][MaxRow-1-k-j]= MasSt[i]
[MaxRow-2-k-j];
                res=true;
            }
        }
    return(res);
}
void  GetNext(void)
{
    byte dx=255;
    byte dy=60;
    x=3;
    y=0;
    for (byte i=0;i<3;i++) {
        //fig[i]=fignext[i];
        if (!Demo) MasSt[x][i]=fignext[i];
        fignext[i]=random(NumCol)+2;
        tft.SetColor(BLACK);
        tft.DrawRect(dx,dy+(razmer-1)*i,dx+razmer, dy+(razmer-1)*(i+1)+1);
        tft.SetColor(MasCol[fignext[i]]);
        tft.DrawFillRect(dx+1,dy+(razmer-1)*i+1, dx+razmer-1, dy+(razmer-
1)*(i+1));
    }
    if (!Demo) {
        Counter++;
        if (Counter==PeriodLevel) {
            Counter=0;
            Level++;
            if (Level>MaxLevel) Level=MaxLevel;
        }
        tempspeed=MasSpeed[Level-1];
    }
}
void  MyScore(void)
{
    TScore=0;
    while(FindFull()) {
        if (TScore>7) Score=Score+TScore+(TScore-8)*2;
        else Score=Score+TScore;
        ViewStacan();
        FlZ=true;
    }
}

```

```

        mydelay(1000);
    }
    FlZ=false;
}
void ProcDemo(void)
{
    Score=0;
    GetNext();
    for (byte j=3;j<MaxRow;j++)
        for (byte i=0;i<MaxCol;i++)
            MasSt[i][j]=random(6)+2;
    ViewStacan();
    mydelay(1000);
    if (!Demo) return;
    MyScore();
    if (Record<Score) Record=Score;
}
void mydelay(int md)
{
    unsigned long startttime=millis();
    while (millis()-startttime < md) {
        VAL=0;
        Mp=mypush();
        if (Mp==KEY_RIGHT) {
            VAL=1;
        } else
        if (Mp==KEY_LEFT) {
            VAL=-1;
        }
        if ((VAL!=0 || Mp!=KEY_EMPTY) && Demo) {
            Demo=false;
            NewGame();
        }
        if (VAL!=0 && figmove(VAL, 0) && !FlZ) {
            for (byte i=0;i<3;i++) {
                MasSt[x+VAL][y+i]=MasSt[x][y+i];
                MasSt[x][y+i]=0;
            }
            ViewStacan();
            if (MasSt[x][y+3]==0) {
                tft.SetColor(WHITE);
                tft.DrawLine(DISP_LEFT+SmeX+x*(razmer-1)+1,
DISP_TOP+SmeY+y*(razmer-1)-3, DISP_LEFT+SmeX+(x+1)*(razmer-1)-1,
DISP_TOP+SmeY+y*(razmer-1)-3);
            }
            x=x+VAL;
        }
        if (Mp==KEY_SWAP && !FlZ) {
            byte aa=MasSt[x][y];
            MasSt[x][y]=MasSt[x][y+2];
            MasSt[x][y+2]=MasSt[x][y+1];
            MasSt[x][y+1]=aa;
            ViewStacan();
        }
        if (Mp==KEY_FALL && !FlZ) tempspeed=50;
    }
}
void NewGame(void)

```

```

{
    for (byte i=0; i<(MaxCol+1); i++) {
        tft.SetColor(BLACK);
        tft.DrawLine(DISP_LEFT+3+(razmer-1)*i, DISP_TOP, DISP_LEFT+3+(razmer-
1)*i, DISP_BOT-3);
    }
    for (byte j=3; j<MaxRow; j++)
        for (byte i=0; i<MaxCol; i++)
            MasOld[i][j]=255;
    Score=0;
    FlNew=true;
    OldScore=Score;
    ClearMas(MasSt);
    Arbeiten=true;
    GetNext();
    Counter=0;
    Level=1;
    tempspeed=MasSpeed[0];
    Record=myrecord;
    ViewStacan();
}
void gameover(void)
{
    Arbeiten=false;
    tft.SetColor(BLACK);
    tft.DrawRect(112, 90, 202, 145);
    tft.SetColor(RED);
    tft.DrawFillRect(113, 91, 201, 144);
    tft.SetFont(&FreeSerif9pt7b);
    tft.SetColor(WHITE);
    tft.OutTextXY(132, 100, "GAME");
    tft.OutTextXY(135, 120, "OVER");
}
bool figmove(int dx, int dy)
{
    bool fff=false;
    if (x+dx<0 || x+dx>MaxCol-1) return(false);
    if (dx!=0) if (MasSt[x+dx][y+dy+2]==0) return(true); else return(false);
    if (dy>0) {
        if (y+dy+2>MaxRow-1 || MasSt[x+dx][y+dy+2]>0) {
            if (y<3) gameover();
            else fff=true;
        } else {
            for (byte i=0; i<3; i++) MasSt[x][y+2-i+dy]=MasSt[x][y+2-i];
            MasSt[x][y]=0;
            y=y+dy;
        }
        if (fff) {
            MyScore();
            GetNext();
        }
        ViewStacan();
    }
    return(true);
}
uint8_t Touch(void)
{
    uint16_t x, y;

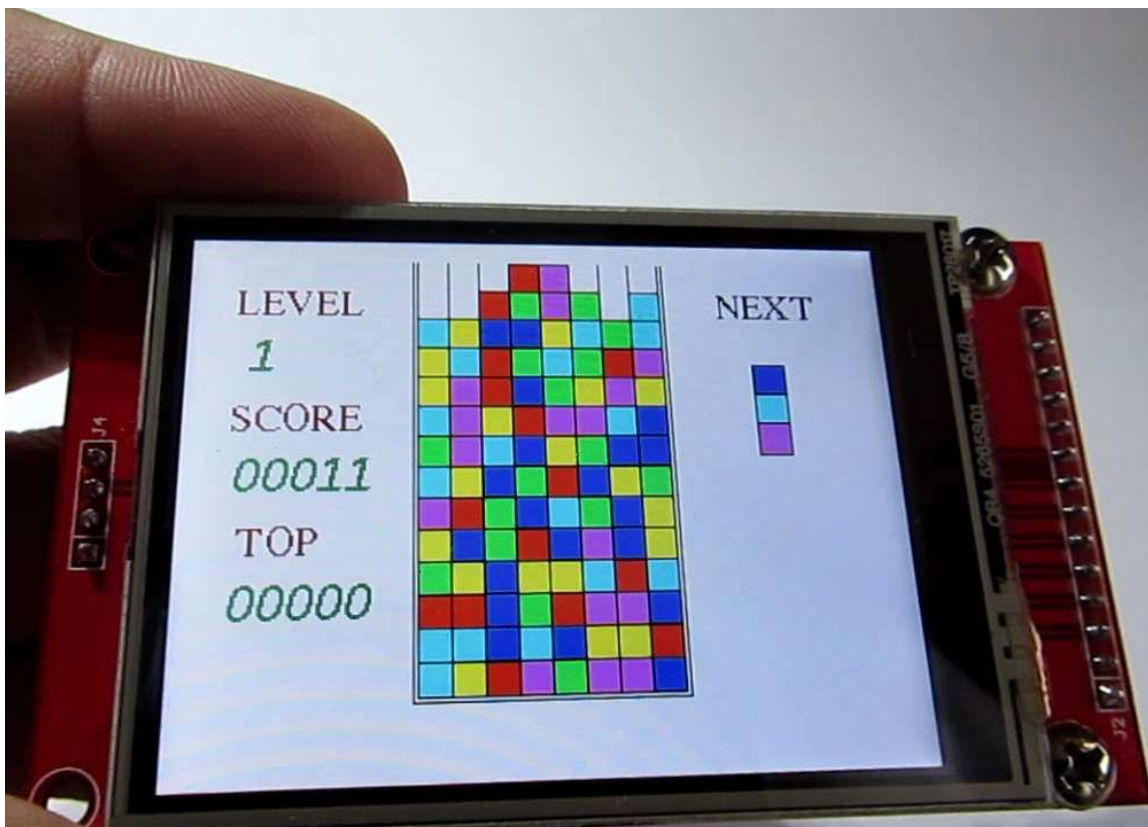
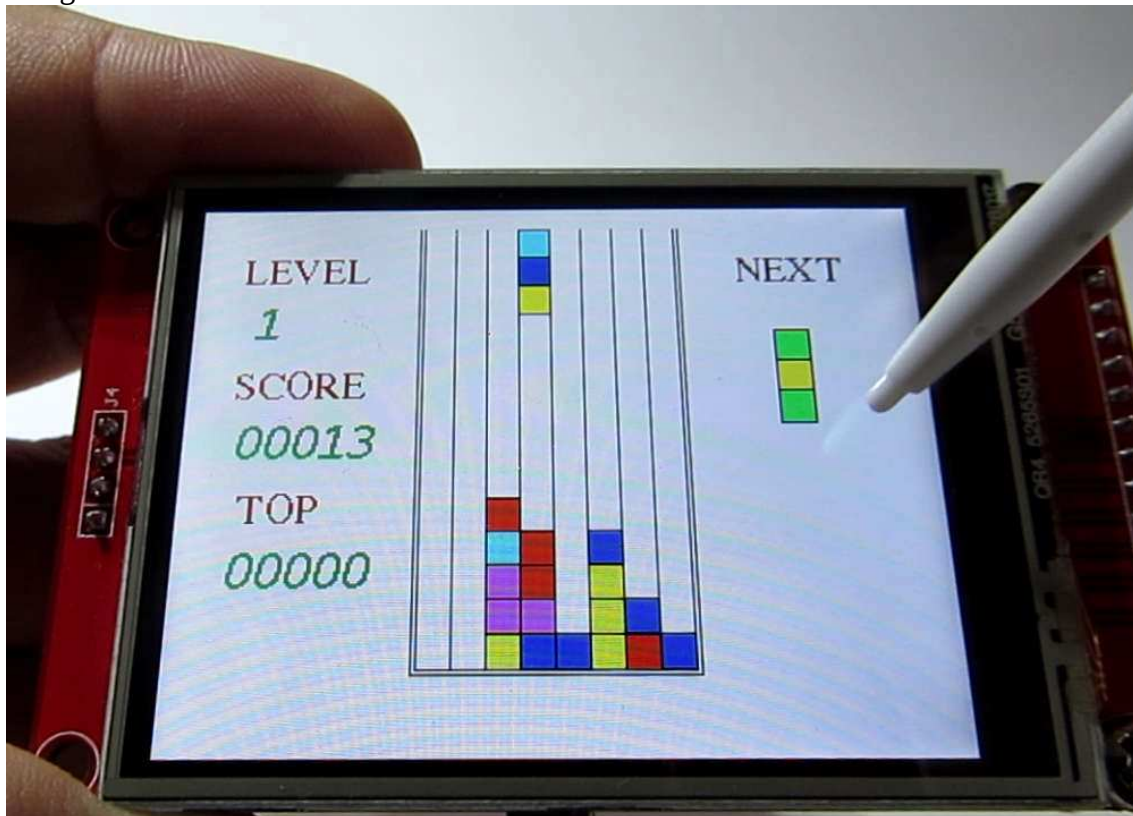
```

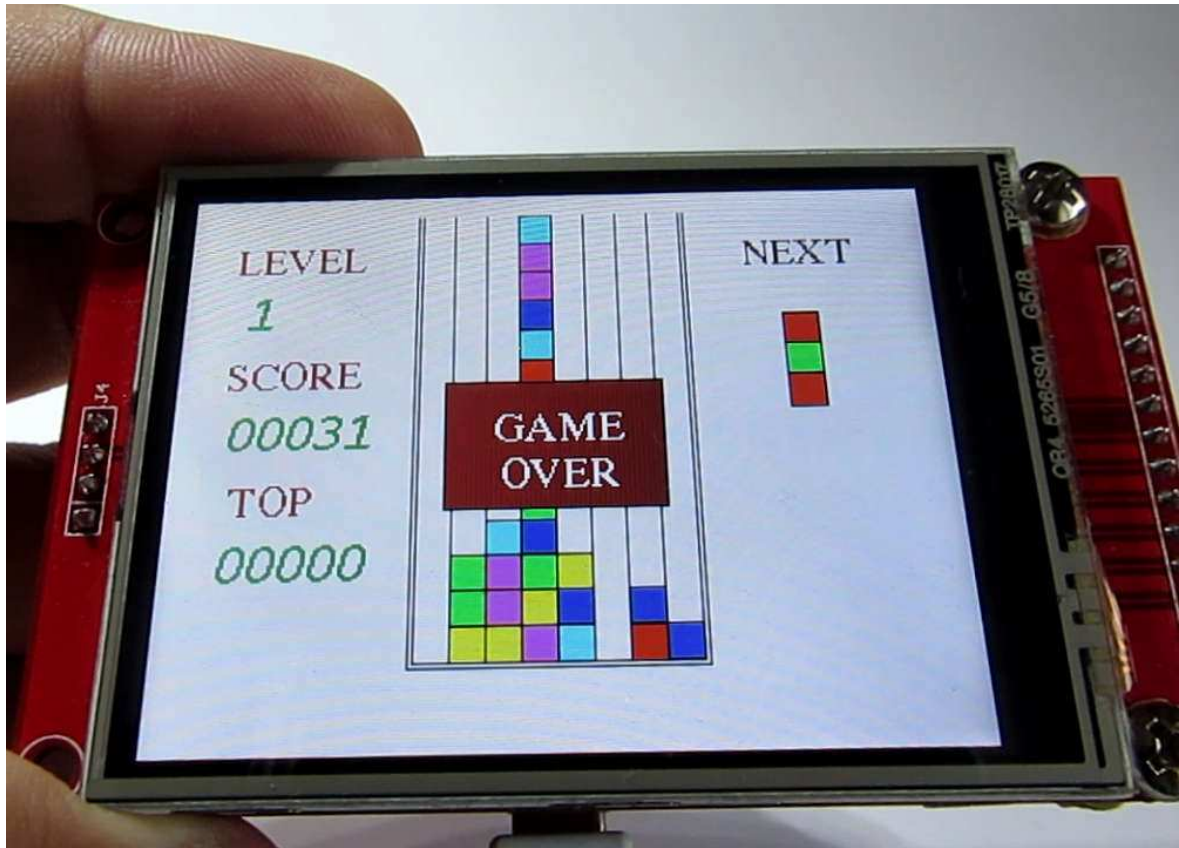
```

uint8_t button = 0;
// Touch
// When the SS pin is set as OUTPUT, it can be used as
// a general purpose output port (it doesn't influence
// SPI operations).
SPI.begin();
if (ts.touched())
{
    TS_Point p = ts.getPoint();
    delay(3); // Delay for filtering
    SPI.end();
    // Calculate coordinates x, y from code ADC
    if (p.x < 450) p.x = 450;
    if (p.y < 500) p.y = 500;
    x = (uint16_t)(320L - ((uint32_t)p.x - 450L)*10L/106L);
    y = (uint16_t)(240L - ((uint32_t)p.y - 500L)*10L/140L);
    if (x < 80) { // Left
        button = 0x08;
    } else
    if (x > 240) { // Right
        button = 0x10;
    } else
    if (y < 120) { // Swap
        button = 0x01;
    } else
        button = 0x02; // Fall
}
SPI.end();
return button;
}

```

Some images:





3.4 Camera Initialization

```
/*
 * Arduino TFT_shield_v1.00 Camera OV7670 initialization example
 *
 * Author: Kamil Lee (2019)
 *
 * Comments: Camera OV7670 initialization
 */

#include <YATFT2.h> // Hardware-specific library
#include <util/yacam.h>

#include "ov7670_regs.h"

YATFT tft(0);
CAM cam;

/*****
void setup() {

    uint8_t state;

    // initialize the serial port
    Serial.begin(115200);
    Serial.println("Camera initialization example!");

    // initialize the display
    tft.begin();

    // Camera OV7670 initialization
    cam.CamInit(&OV7670_VGA[0][0]);
    cam.CamVideoPreview(0, 0, 1, true);
}

void loop()
{
}
```