

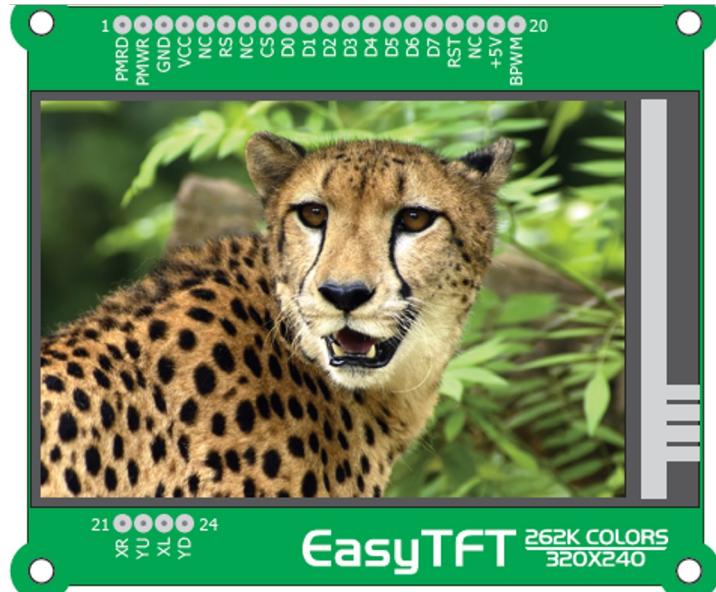


TFT Display & Touch Panel

Prepared by:
Omar Samir

About TFT

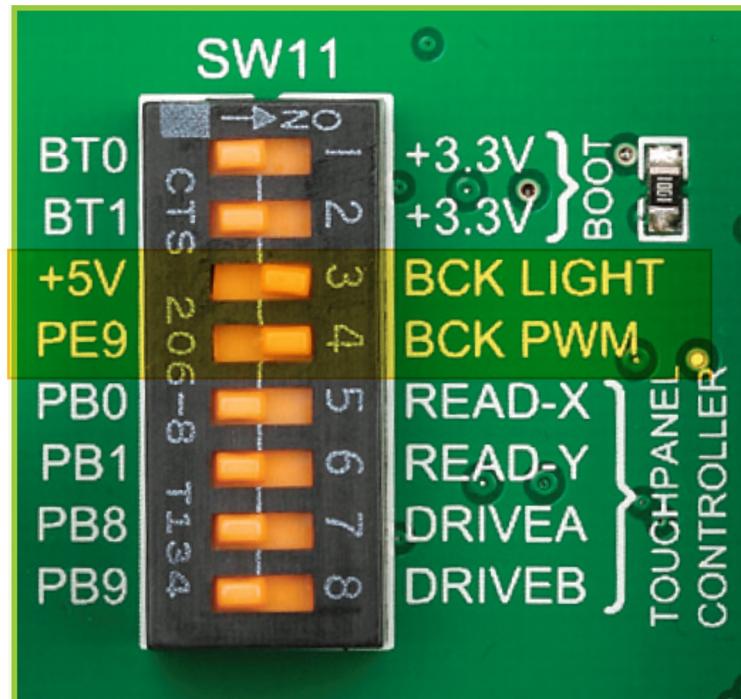
- Color Display
- 320 x 240 pixels
- 262.144 different colors
- Connected to microcontroller PORTE using standard 8080 parallel 8-bit interface
- Brightness control via PWM signal



Enable TFT

Turn on switches

- SW11.3
- SW11.4

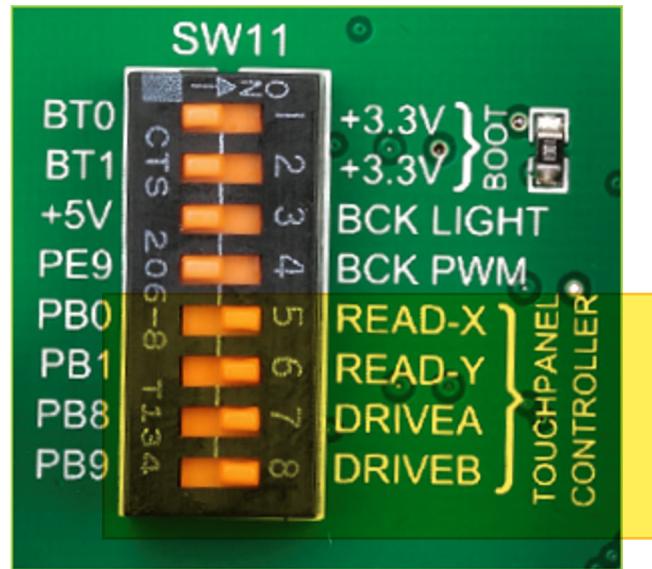


About Touch Panel

- Comes as a part of TFT 320x240 display
- A glass panel whose surface is covered with two layers of resistive material
- When the screen is pressed, the outer layer is pushed onto the inner layer and appropriate controllers can measure that pressure and pinpoint its location
- The pressed coordinates can be converted to X and Y values

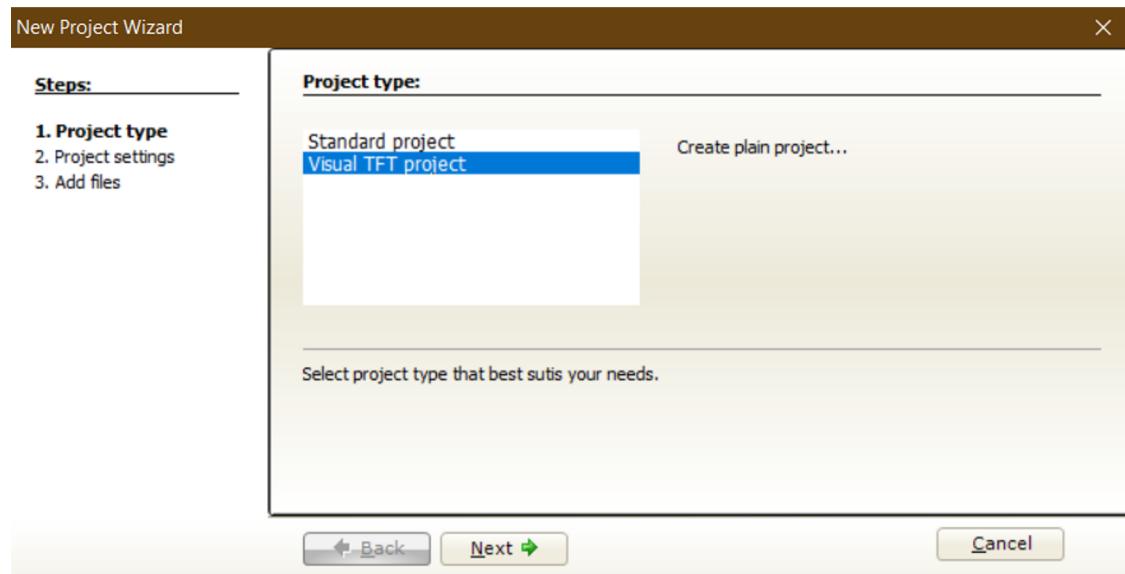
Enable Touch Panel

Enable pins 5, 6, 7, 8 in SW11

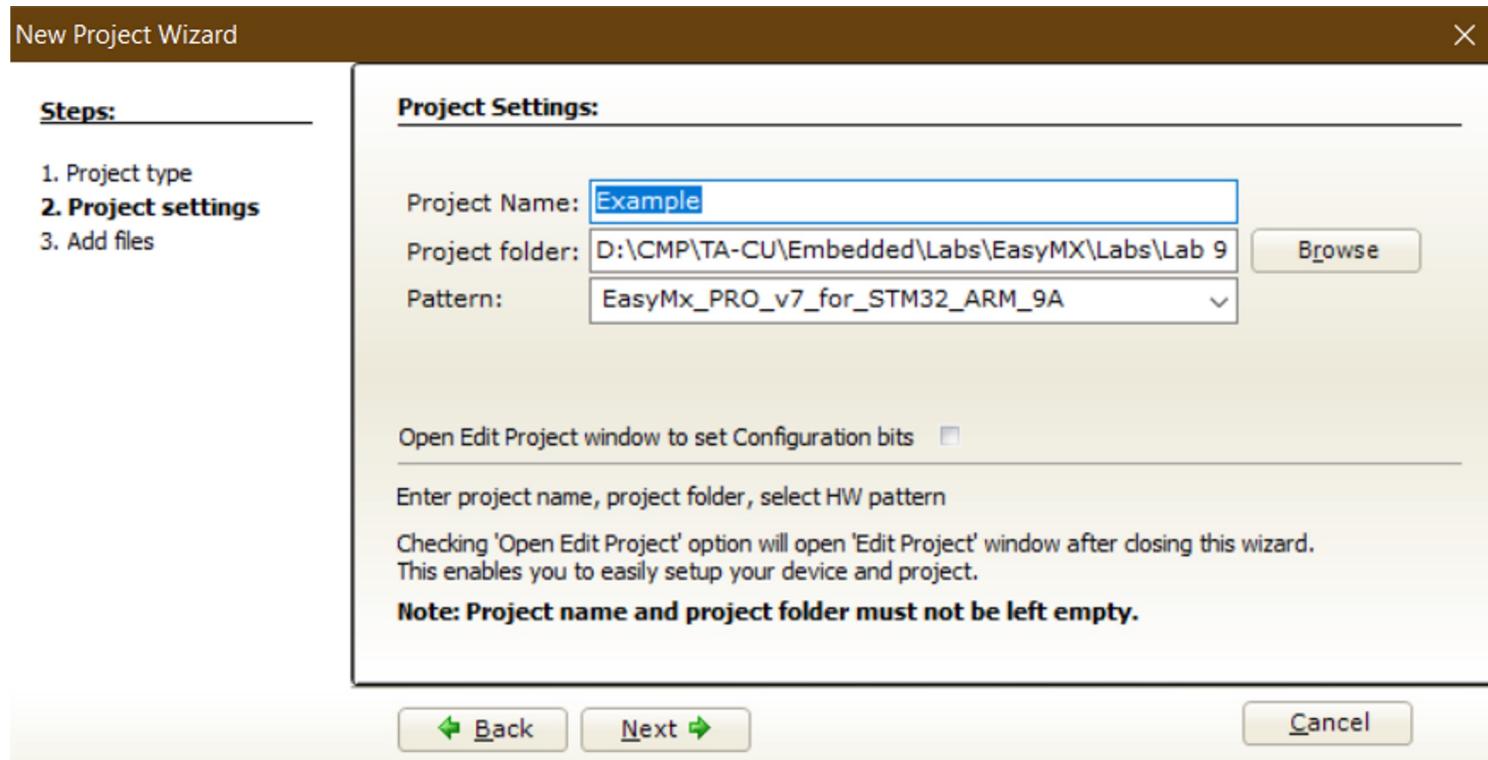


Starting Visual TFT project

1. Open MikroC for ARM
2. Click on New Project
3. Select Visual TFT project
4. Click Next



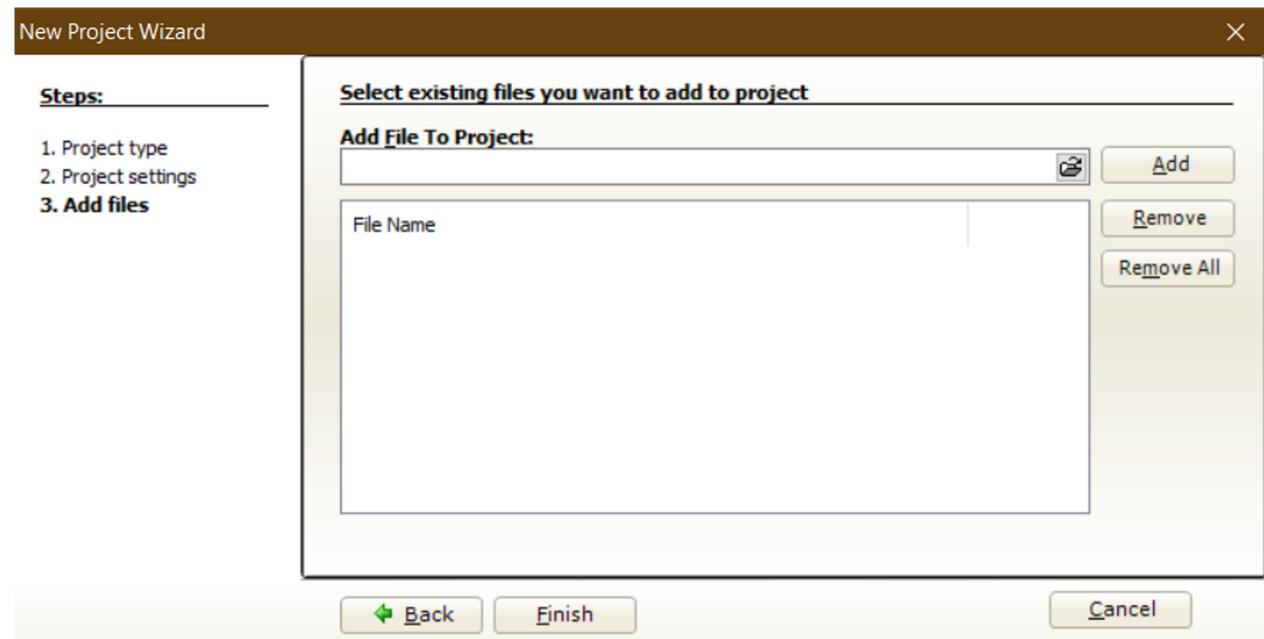
Starting Visual TFT project



Starting Visual TFT project

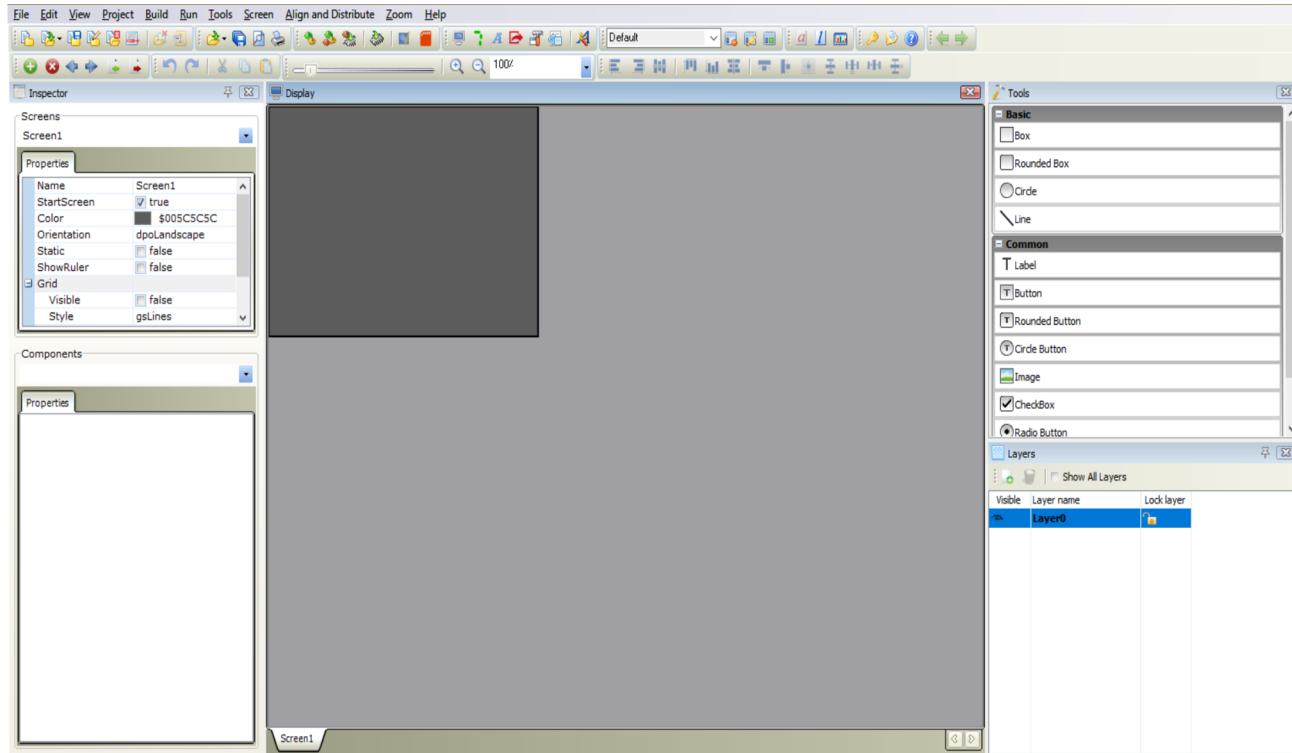
Click Finish

It will generate all required files for the project



Starting Visual TFT project

It will open the Visual TFT window where you can design your TFT display.



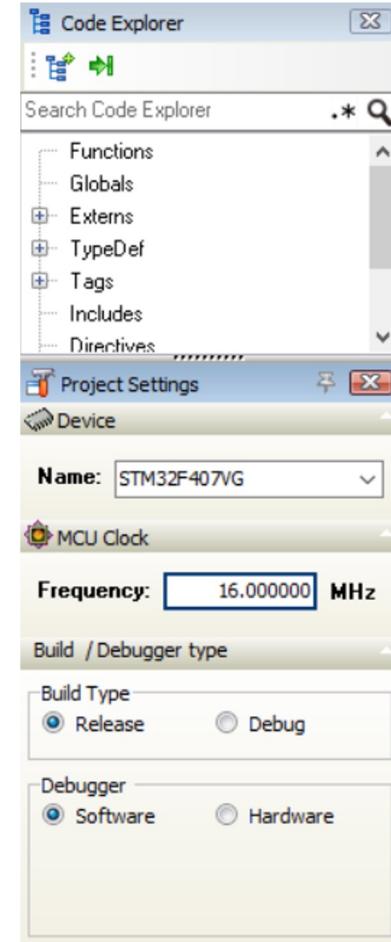
Use **F12** to switch between the Visual TFT window and the code window

Starting Visual TFT project

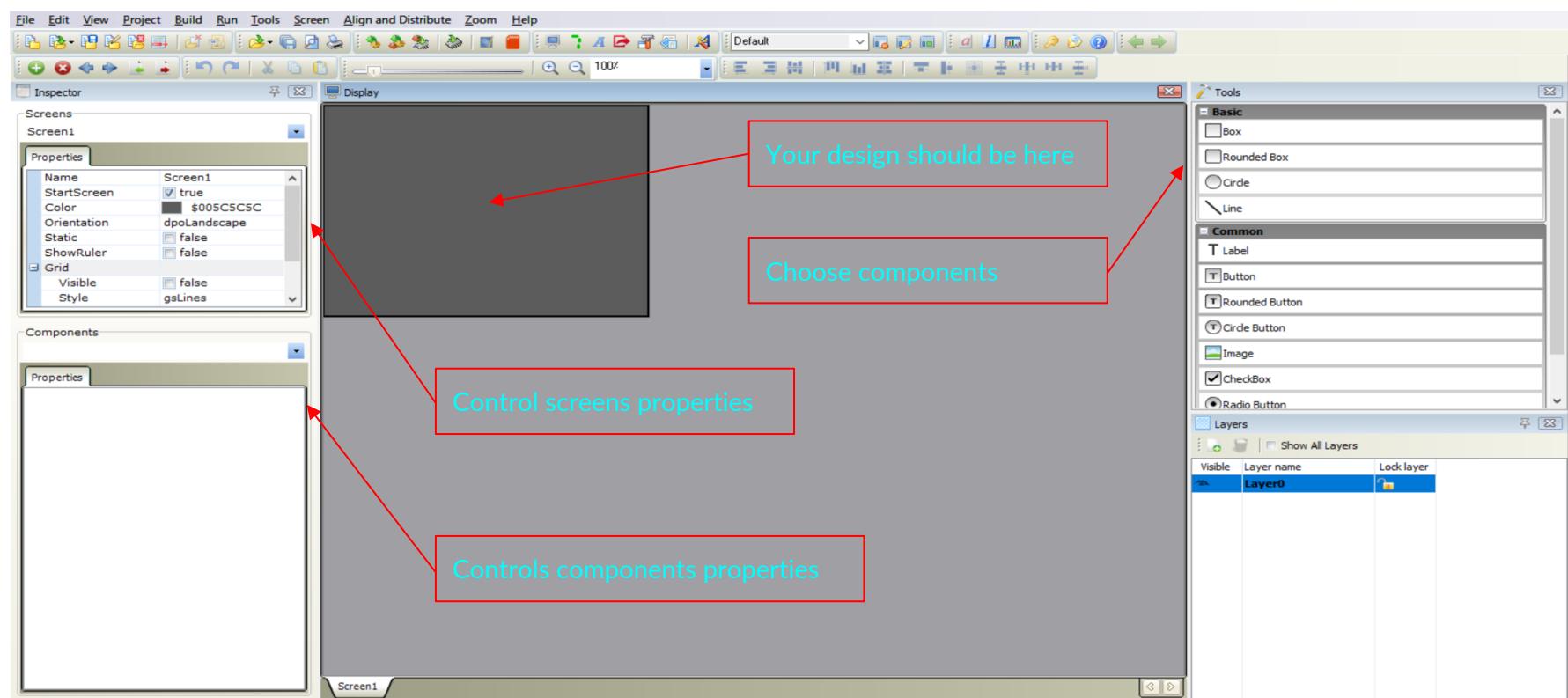
Change the device and MCU Clock in the Project Settings to:

- STM32F407VG
- 16 MHz

By this step we are ready to use the TFT display



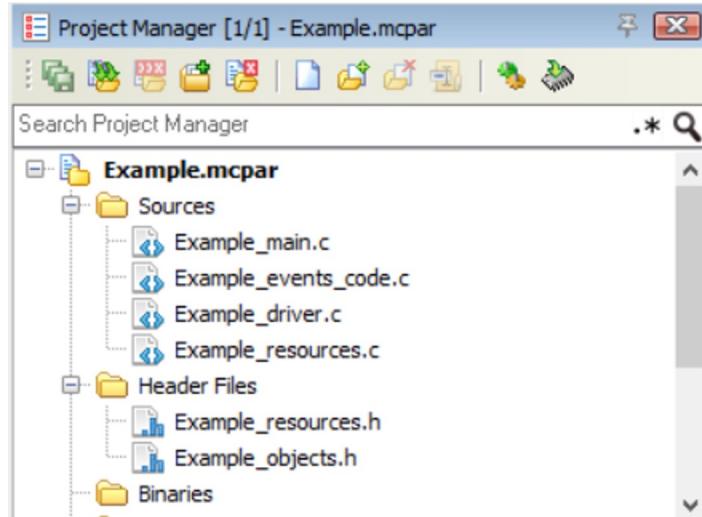
Visual TFT Window



Code Files

Visual TFT project generates all necessary source and header files

It uses MikroC TFT library and Touch Panel library



* _main.c

This file contains the main function.

- Start_TP() does all necessary configurations
- The ordinary while loop of the program
- Check_TP() checks if the touch panel is pressed and does any necessary event handling

```
#include "Example_objects.h"

void main() {
    Start_TP();
    while (1) {
        Check_TP();
    }
}
```

* _driver.c

This file contains all configuration, objects and drawing functions.

- Initialize TFT and touch panel connections
- Initialize the TFT resolution

```
static void InitializeTouchPanel() {
    Init_ADC();
    TFT_Init_ILI9341_8bit(320, 240);

    TP_TFT_Init(320, 240, 8, 9);
    TP_TFT_Set_ADC_Threshold(ADC_THRESHOLD);

    PenDown = 0;
    PressedObject = 0;
    PressedObjectType = -1;
}
```

```
// TFT module connections
sbit TFT_BLED at GPIOE_ODR.B9;
sbit TFT_CS at GPIOE_ODR.B15;
unsigned int TFT_DataPort at GPIOE_ODR;
sbit TFT_RD at GPIOE_ODR.B10;
sbit TFT_RS at GPIOE_ODR.B12;
sbit TFT_RST at GPIOE_ODR.B8;
sbit TFT_WR at GPIOE_ODR.B11;
// End TFT module connections

// Touch Panel module connections
sbit DriveX_Left at GPIOB_ODR.B1;
sbit DriveX_Right at GPIOB_ODR.B8;
sbit DriveY_Up at GPIOB_ODR.B9;
sbit DriveY_Down at GPIOB_ODR.B0;
// End Touch Panel module connections
```

* _driver.c

- Create and initialize Screen Object
- DrawScreen Function

```
////////////////////////////////////////////////////////////////  
TScreen* CurrentScreen;  
  
TScreen           Screen1;  
  
  
static void InitializeObjects() {  
    Screen1.Color          = 0x5AEB;  
    Screen1.Width          = 320;  
    Screen1.Height         = 240;  
    Screen1.ObjectsCount   = 0;  
}
```

```
void DrawScreen(TScreen *aScreen) {  
    unsigned int order;  
    char save_bled;  
  
    object_pressed = 0;  
    order = 0;  
    CurrentScreen = aScreen;  
  
    if ((display_width != CurrentScreen->Width) || (display_height != CurrentScreen->Height)) {  
        save_bled = TFT_BLED;  
        TFT_BLED      = 0;  
        TFT_Init_ILI9341_8bit(CurrentScreen->Width, CurrentScreen->Height);  
        TP_TFT_Init(CurrentScreen->Width, CurrentScreen->Height, 8, 9);  
        TP_TFT_Set_ADC_Threshold(ADC_THRESHOLD);  
        TFT_Fill_Screen(CurrentScreen->Color);  
        display_width = CurrentScreen->Width;  
        display_height = CurrentScreen->Height;  
        TFT_BLED      = save_bled;  
    }  
    else  
        TFT_Fill_Screen(CurrentScreen->Color);  
  
    while (order < CurrentScreen->ObjectsCount) {  
    }  
}
```

* _resources.c

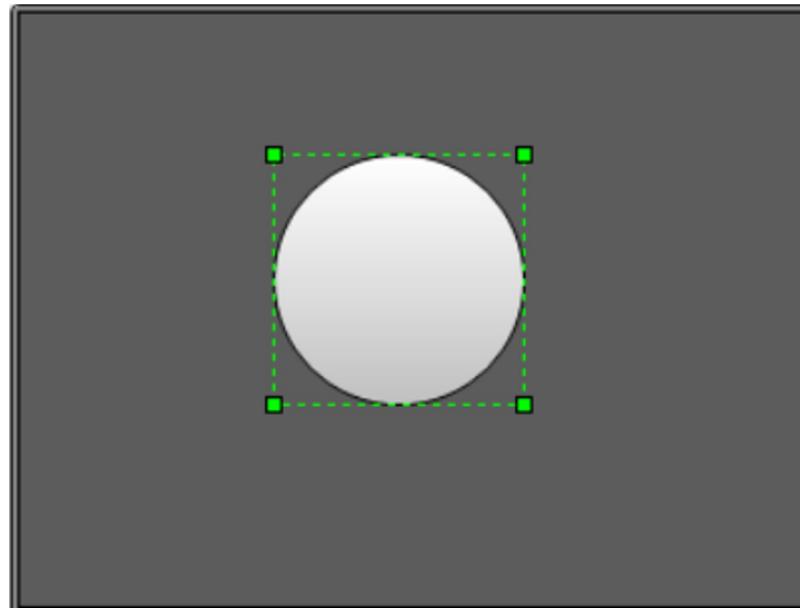
Contains any resources to be displayed like images.

*** _event_codes.c**

Contains event handlers of touch panel to handle pressing, clicking, etc..

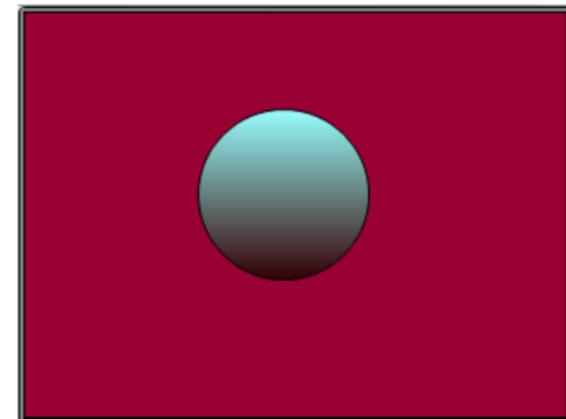
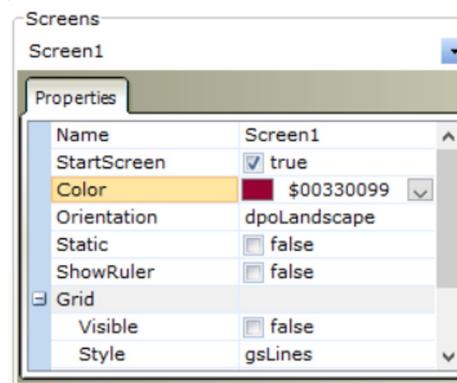
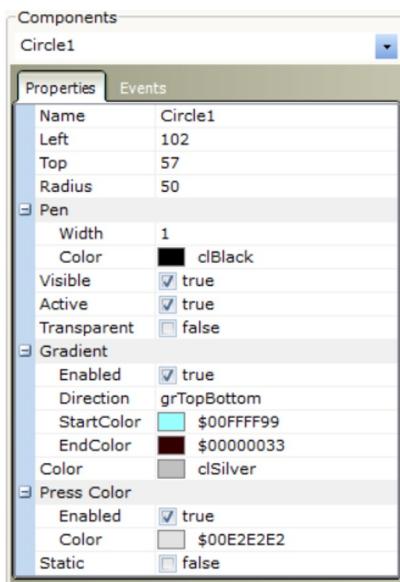
Display Circle Example

In the Visual TFT window drag and drop a circle from the Tools on the display



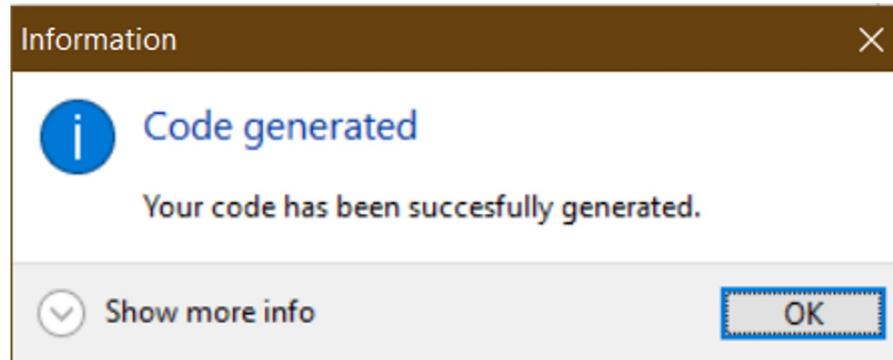
Display Circle Example

- Change the Circle colors from components -> properties
- Change the screen color From screens -> properties



Display Circle Example

Go to code window by pressing F12



Display Circle Example

In *_driver.c you will find all code of the circle added

```
TCircle           Circle1;
TCircle           * const code Screen1_Circles[1]=
{
    &Circle1
};
```

```
Circle1.OwnerScreen      = &Screen1;
Circle1.Order             = 0;
Circle1.Left              = 102;
Circle1.Top               = 57;
Circle1.Radius            = 50;
Circle1.Pen.Width         = 1;
Circle1.Pen.Color          = 0x0000;
Circle1.Visible           = 1;
Circle1.Active             = 1;
Circle1.Transparent        = 1;
Circle1.Gradient           = 1;
Circle1.Gradient.Orientation = 0;
Circle1.Gradient.Start.Color = 0x9FFF;
Circle1.Gradient.End.Color = 0x3000;
Circle1.Color              = 0xC618;
Circle1.PressColEnabled   = 1;
Circle1.Press.Color         = 0xE71C;
```

Display Circle Example

In *_driver.c you will find all code of the circle added

```
void DrawCircle(TCircle *ACircle) {
    if (ACircle->Visible != 0) {
        if (object_pressed == 1) {
            object_pressed = 0;
            TFT_Set_Brush(ACircle->Transparent, ACircle->Press Color, ACircle->Gradient, ACircle->Gradient Orient,
                          ACircle->Gradient_End_Color, ACircle->Gradient_Start_Color);
        }
        else {
            TFT_Set_Brush(ACircle->Transparent, ACircle->Color, ACircle->Gradient, ACircle->Gradient_Orientation,
                          ACircle->Gradient_Start_Color, ACircle->Gradient_End_Color);
        }
        TFT_Set_Pen(ACircle->Pen_Color, ACircle->Pen_Width);
        TFT_Circle(ACircle->Left + ACircle->Radius,
                   ACircle->Top + ACircle->Radius,
                   ACircle->Radius);
    }
}
```

Display Circle Example

- Build and Program the project
- The screen will appear successfully on the TFT

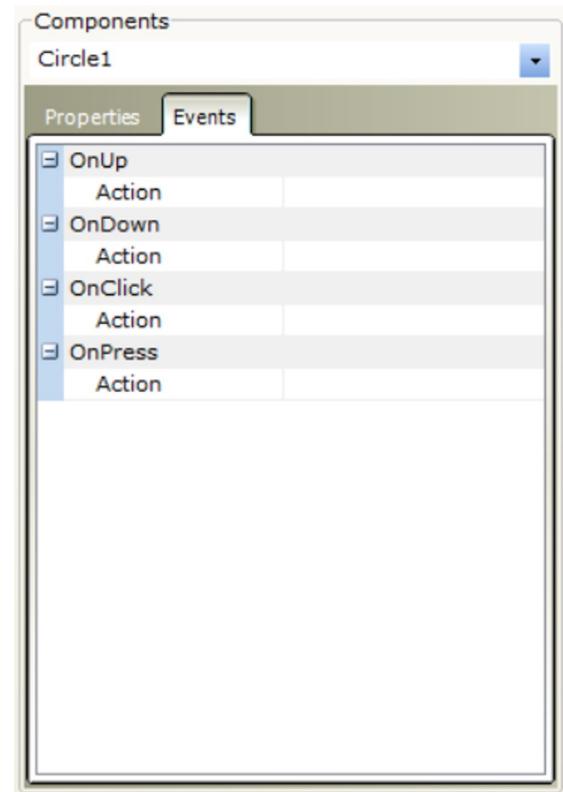
Touch Panel Example

Now lets handle if the Circle is pressed.

- Select the circle in the visual TFT window
- Click on Events in Components

We will work with the following actions

- onClick: when the object is clicked and released
- onPress: when the object is pressed (keep working while pressing)



Touch Panel Example

- Double click beside on the action you need (onClick in our example)
- The project will redirect you to `Example_events_code.c`
- It will create empty function to place your logic in

```
// Event Handlers |  
  
void Circle1OnClick() {  
}  
}
```

Touch Panel Example

- In the circle Handler here we set the visibility of the circle to be zero
- Redraw the screen again to see the effect

```
void Circle1OnClick() {  
    Circle1.Visible = 0;  
    DrawScreen(&Screen1);  
}
```

Touch Panel Example

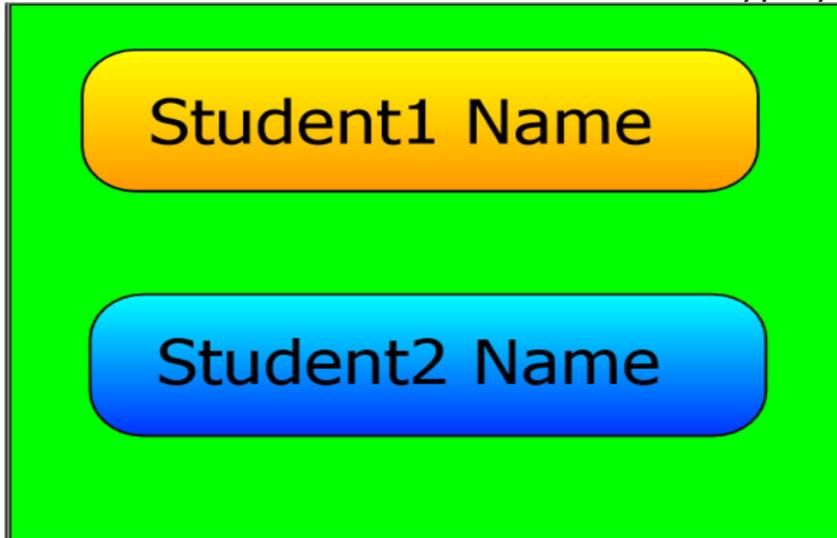
- Build and Program the project
- The screen will appear successfully on the TFT
- When the circle is pressed it disappears



Requirements

Requirement 1

It is required to draw the given screen with your names. The background color should be green. Place your names in round boxes with corner radius = 20. Type your names with font size = 20.



Requirement 2

It is required to display the egyptian flag on the screen. You are given a JPEG image of the eagle. Complete on the same project by adding new screen from the + button in the toolbar.

Steps:

1. Add Red Box
2. Add white Box
3. Add black Box
4. Add the eagle image

Hints:

Check which screen to be called in Start_TP()



Requirement 3

It is required to use PC0 as input pin to switch between Requirement one and two screens. The transition must happen on the rising edge.

Hints:

Use DrawScreen() in the main function to do the switch

Requirement 4

It is required to make your names clickable. When Student1 Name is clicked all PORTD pins are turned on. When Student2 Name is clicked all PORTD pins are turned off.