

TouchGFX > Knowledge Base > Porting TouchGFX

🔍 Search

## Configuring STM32F429I-DISCO

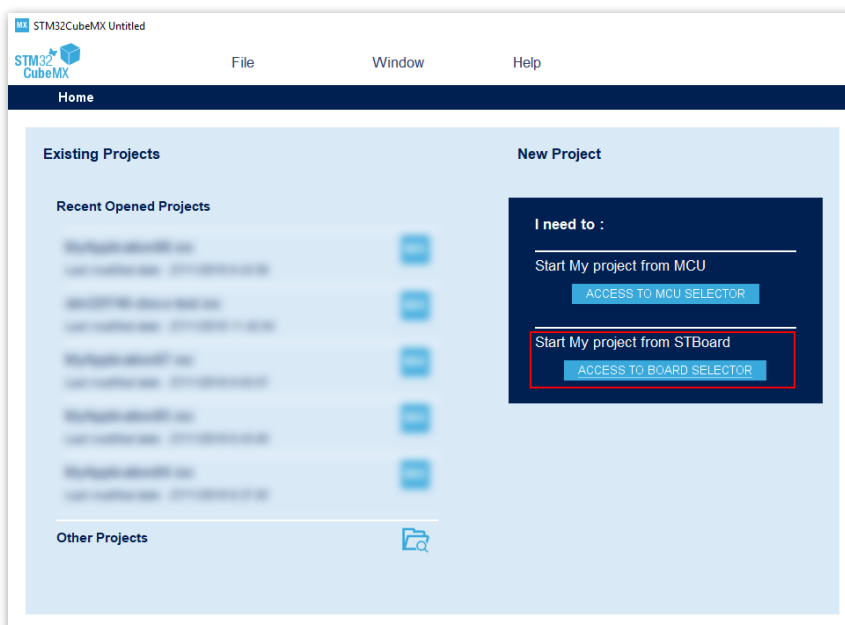
### Introduction

This article shows you how to generate and configure a TouchGFX project for the STM32F429I-DISCO board using CubeMX and the TouchGFX Designer through the following steps:

- Create the initial project
- Graphics framework and display interface configuration
- Clock configuration
- TouchGFX application bootstrapping
- Touch Controller configuration

### Creating the initial project

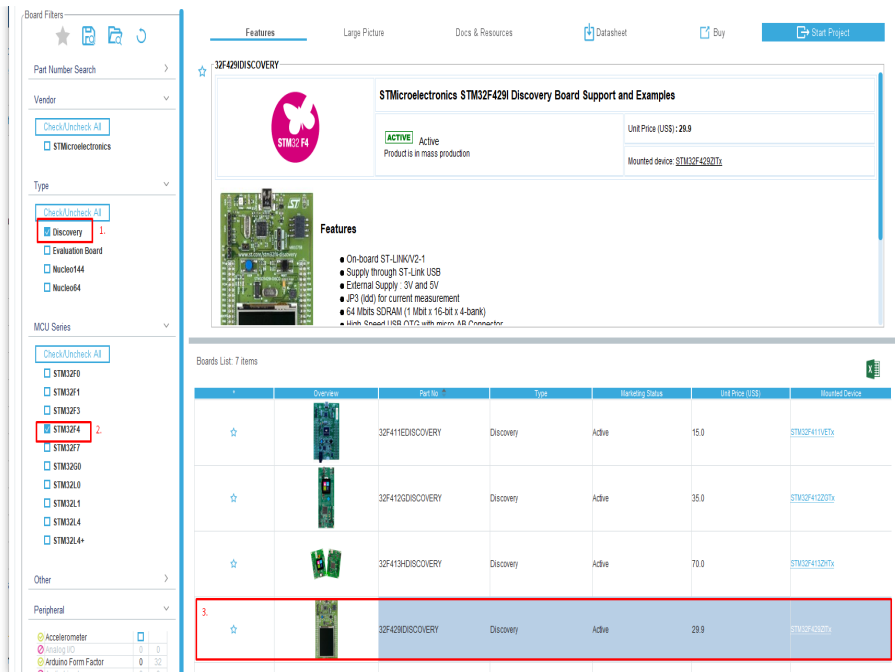
Create a new project by using the board selector by selecting "Access to Board Selector".



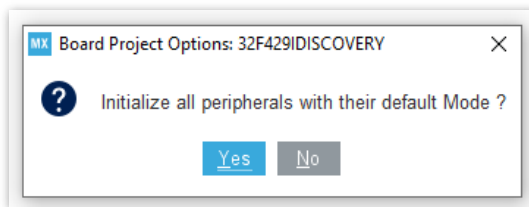
In the Board Selector menu, create a project based on the STM32F429I-DISCO board by following the step below.

1. Select type of board
2. Select STM32F4 MCU family
3. Select STM32F429I-DISCO board
4. Select "Start Project"





After selecting "Start Project" a pop-up asking to Initialize all peripherals to default mode appears. Select yes to initialize all peripherals to their default mode.

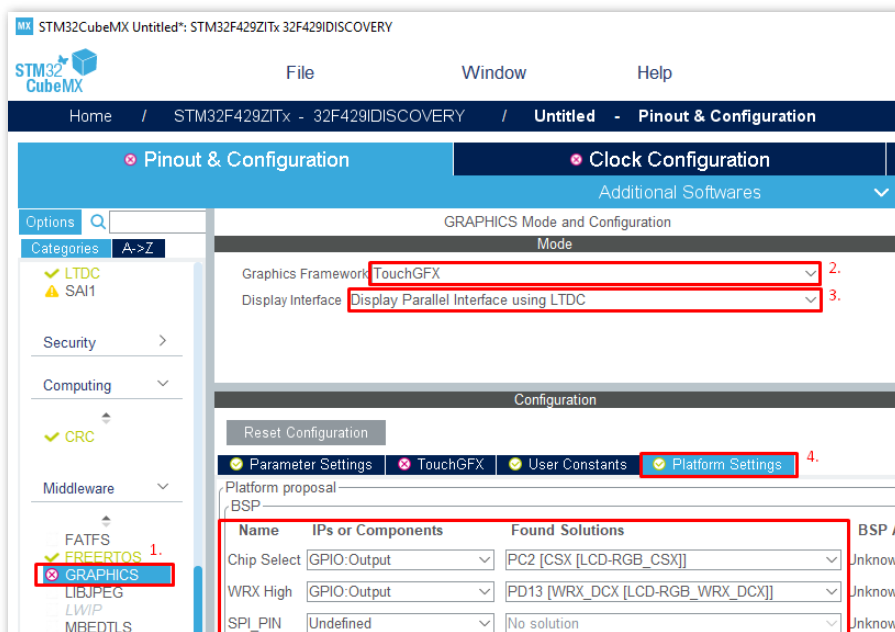


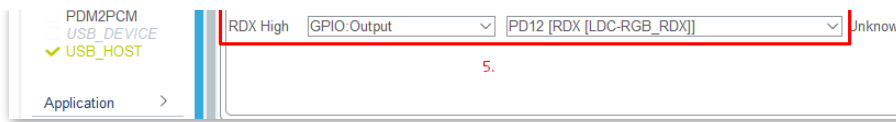
With the project initialized the project should be saved and the initial code should be generated. This is done by moving to the Project Manager tab and do the following:

1. Set the name for the project
2. Use the Generate Code button to save the project and generate the initial code

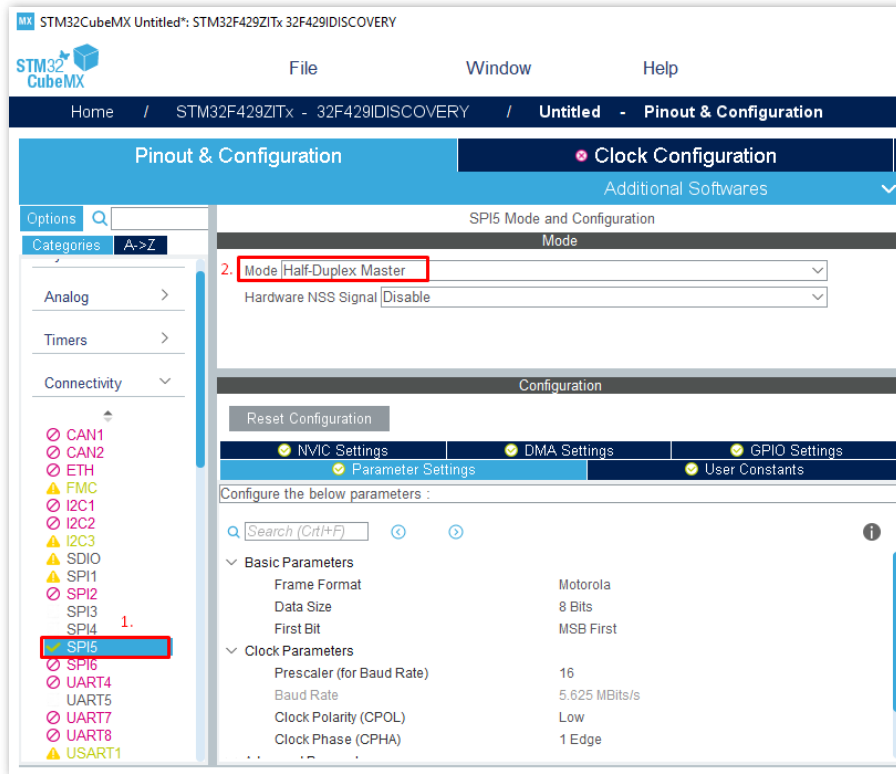
## Graphics and Display Interface

Configure the Graphics middleware (TouchGFX) and display interface platform settings as outlined in the below screenshot.

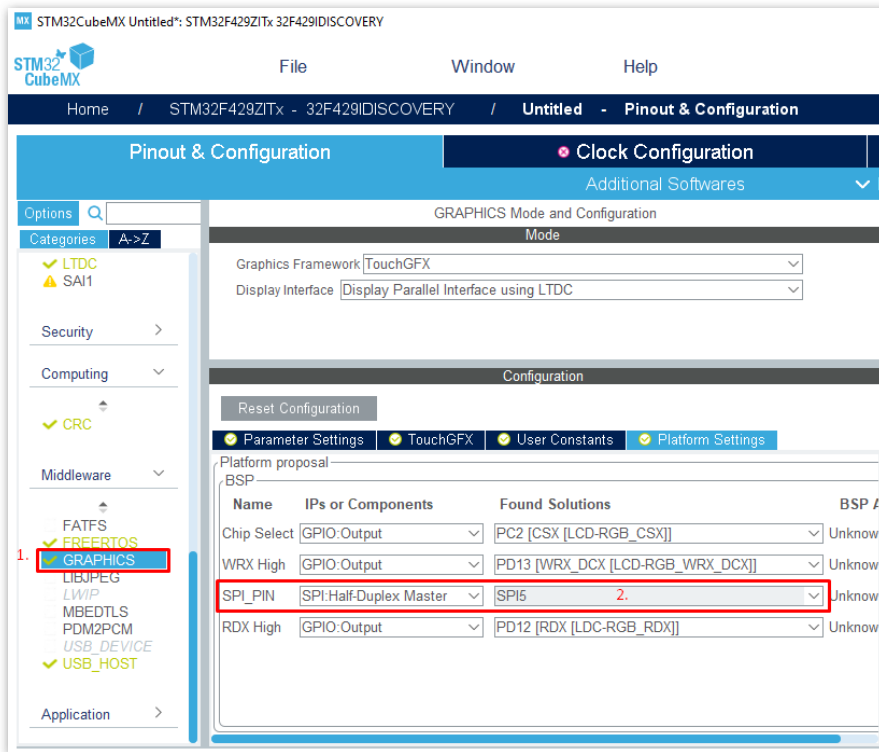




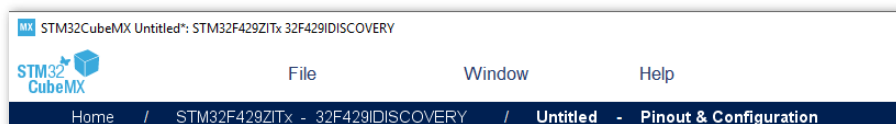
To configure the remaining option "SPI\_PIN", configure SPI5 in half-duplex mode.

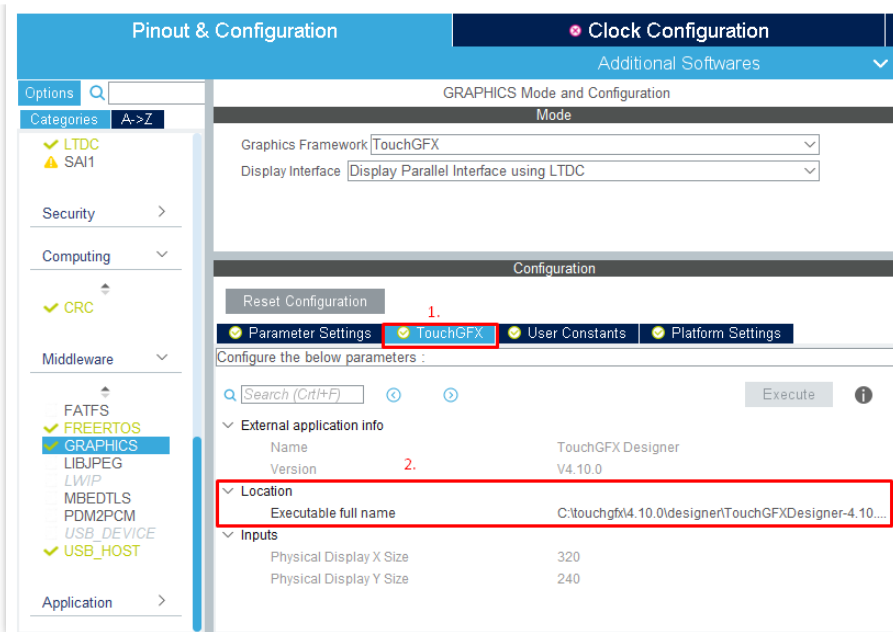


Return to the Graphics middleware settings and configure the SPI\_PIN option to use SPI5.

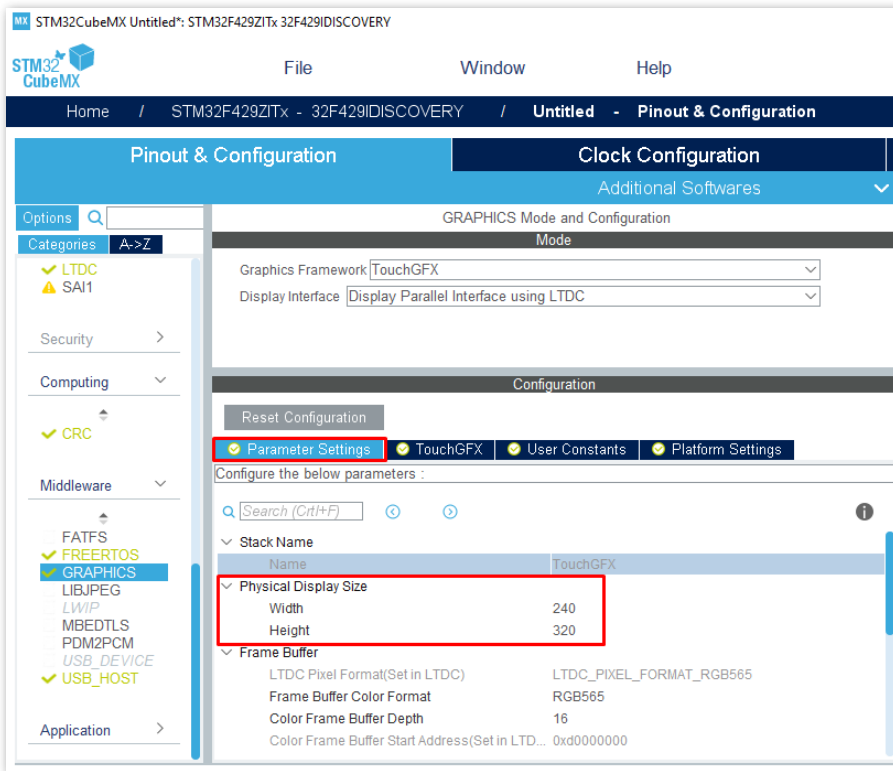


Let CubeMX know where the TouchGFX 4.10.0 designer is located.



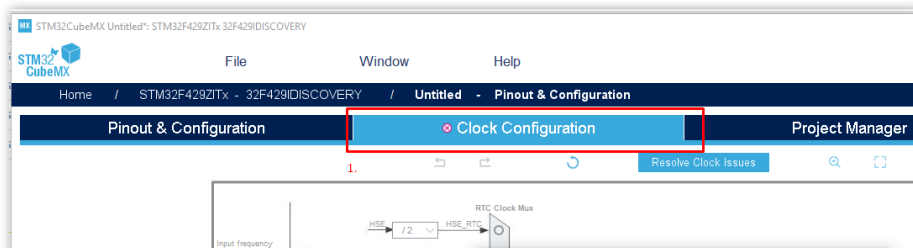


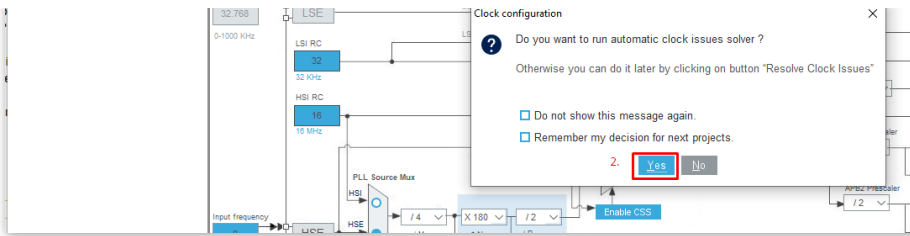
The native resolution of the display must be changed from **320,240** to **240, 320**. CubeMX configures the project to run in RGB565 (16 bpp) mode.



## Clock Config

Allow clock configurator to resolve an issue with one of the clock settings.

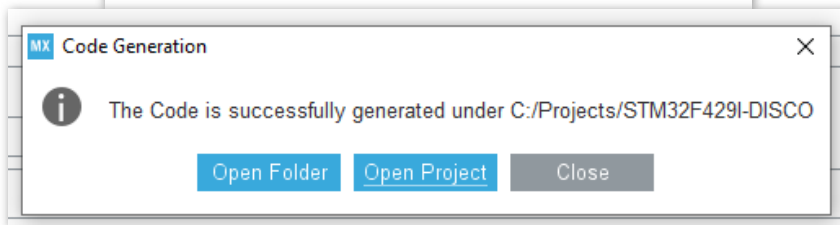
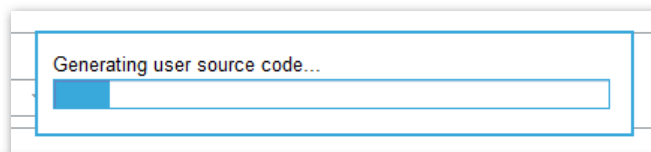
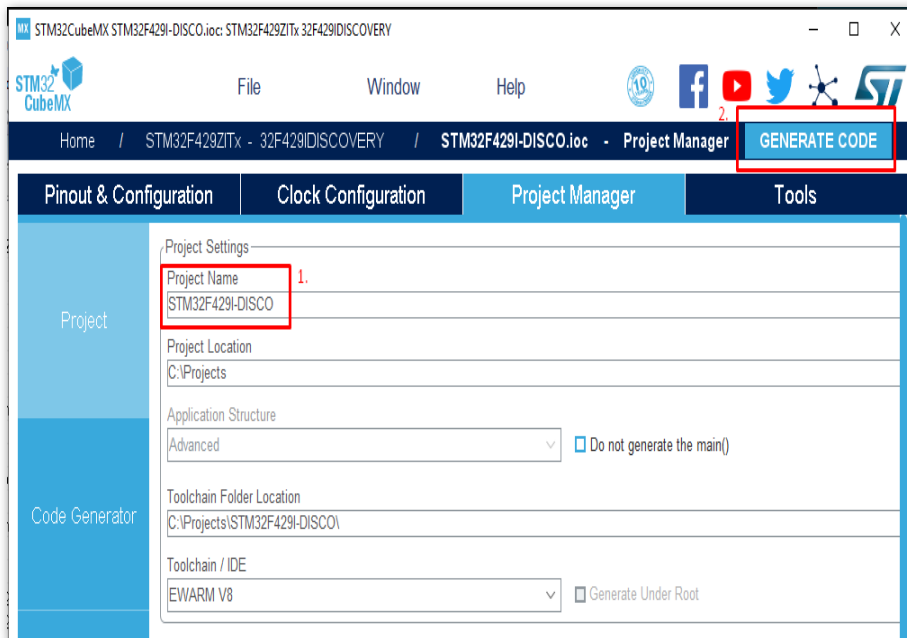




Once completed, change the LCD-TFT clock setting to 6 Mhz.

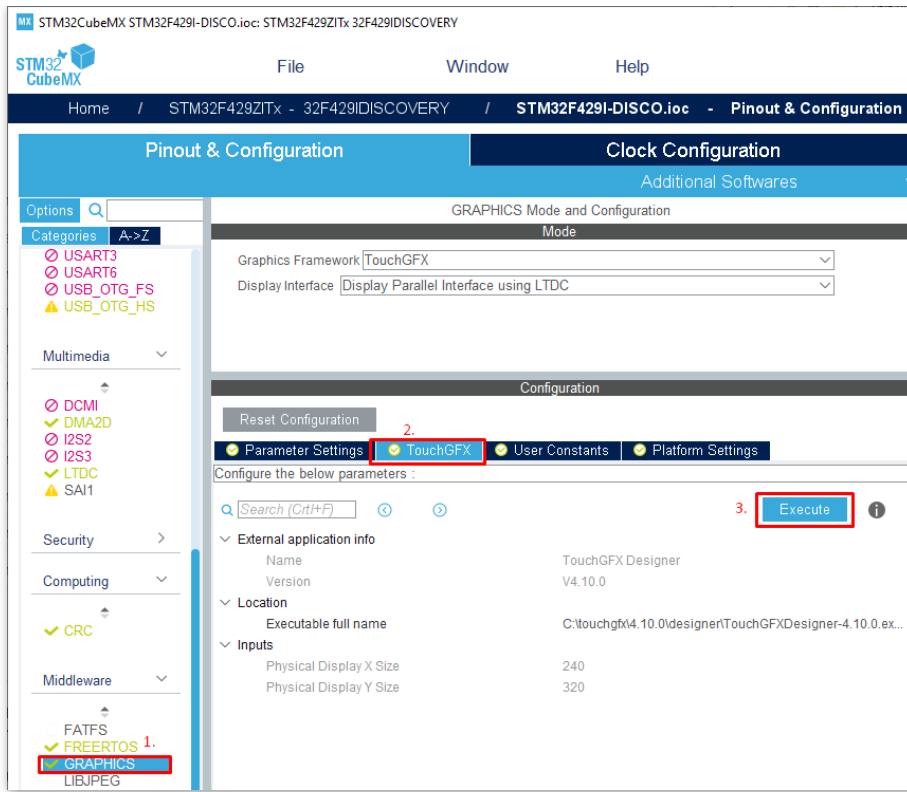
## Generate the project

After giving the project a name, generate the project. By default CubeMX will generate an EWARM V8 project using latest version of Cube Drivers.



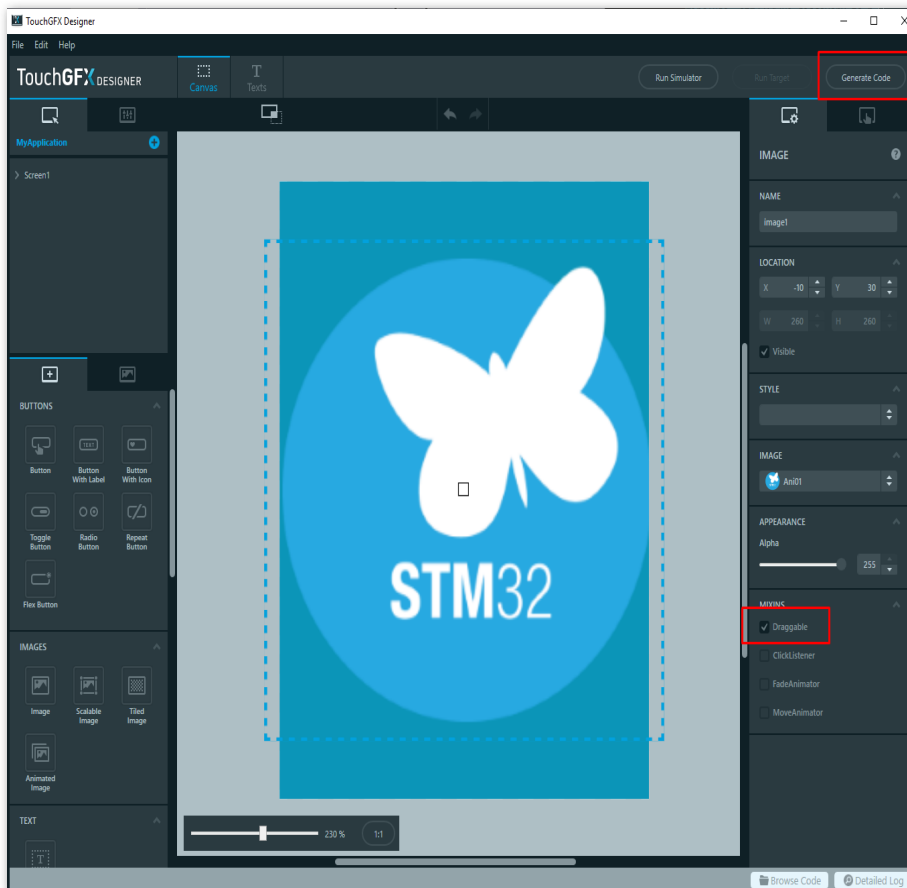
## Bootstrapping the TouchGFX application

Now that the CubeMX configuration is completed, press the "Execute" button to launch the project in the TouchGFX Designer.

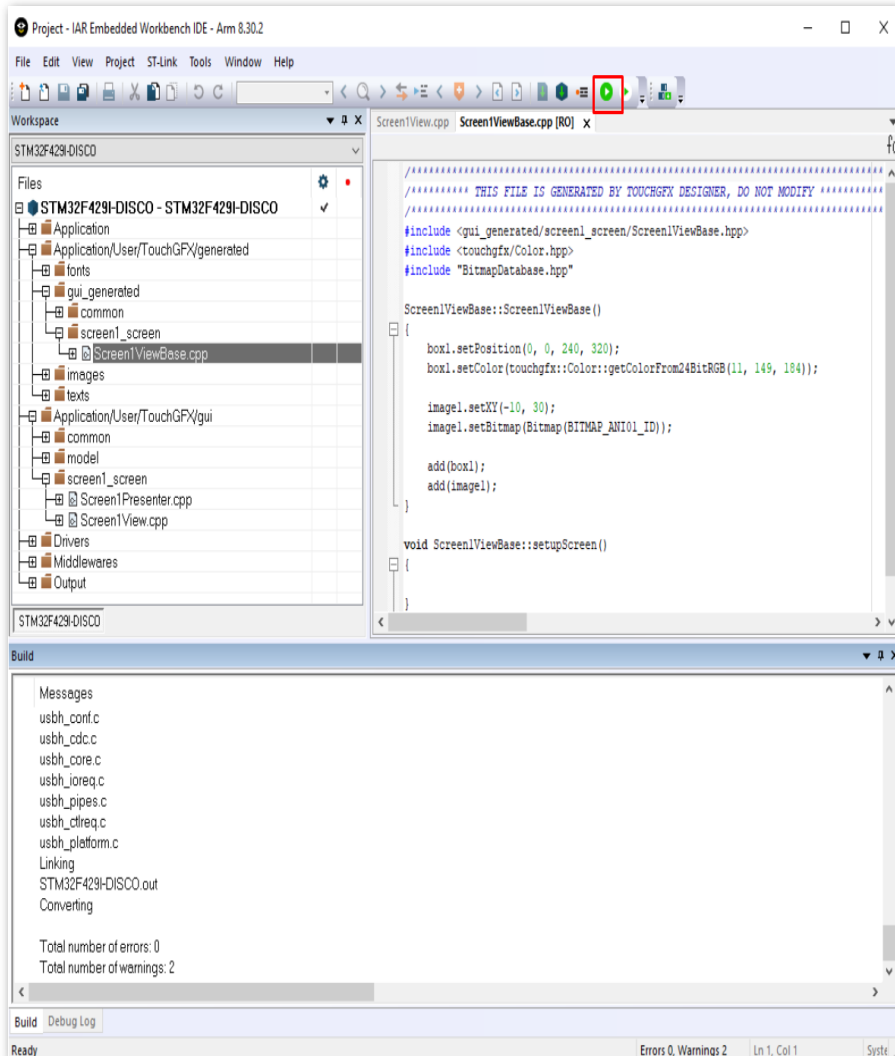


Upon loading the project the designer will bootstrap your application with required framework files. Code generation (Screen definitions, among other things) happens automatically but you can press the "Generate Code" button as you wish.

To verify the configuration add a "Draggable" element to the canvas. Save your project and close the designer.

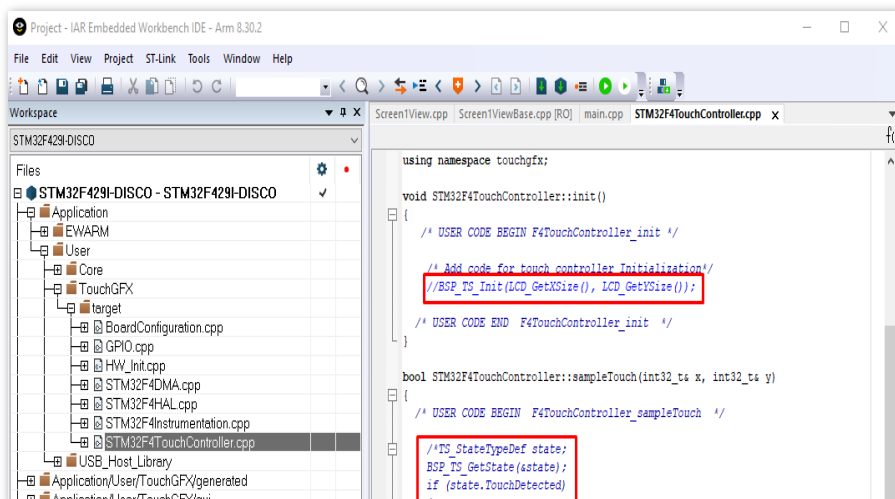


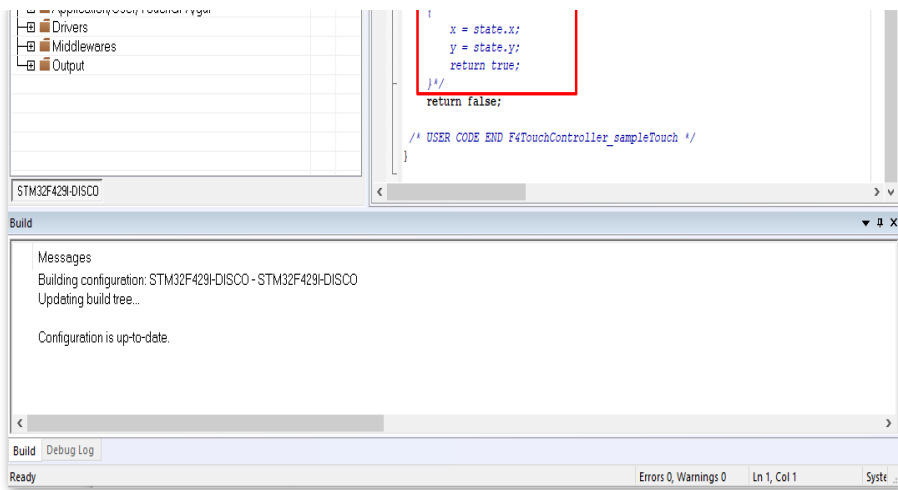
Open the EWARM project (EWARM/Project.eww) with IAR 8 and compile the project (F7). Notice the generated code for screen1 as defined through the Designer. Pressing **"Download and Debug"** will program attached STM32F429I-DISCO board and reset the board to ultimately display your application. The next section shows you how to support for the touchcontroller.



## Touch Controller

*The file STM32F4TouchController.cpp already comes with some commented user code.*

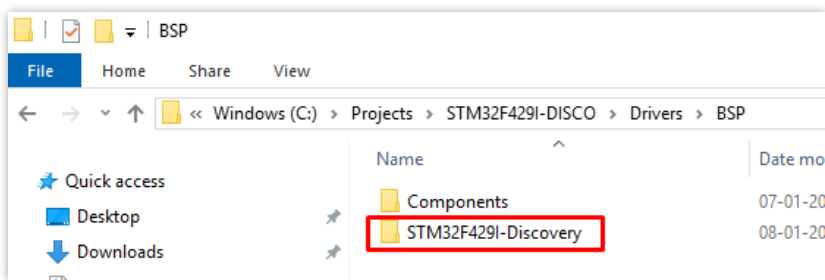




CubeMX does not copy necessary STM32F429I-DISCO Cube driver pack files and this must be done manually.

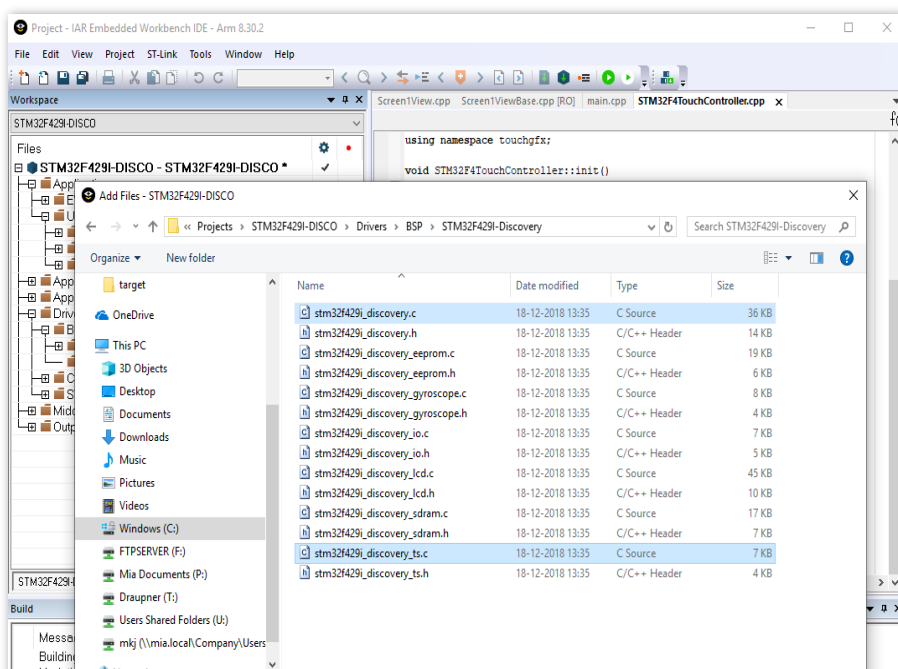
In conclusion, touch support is added by:

1. Uncommenting the touch controller code (See screenshot above).
2. Adding the relevant touch controller code files to your project manually. On Windows the Cube Driver pack used by CubeMX can be located at C:\Users\  
<user>\STM32Cube\Repository\STM32Cube\_FW\_F4\_Vx.y.z\Drivers\BSP\STM32F429I-Discovery. Copy the entire folder to your project at STM32F429I-DISCO\Drivers\BSP (you may need more drivers).

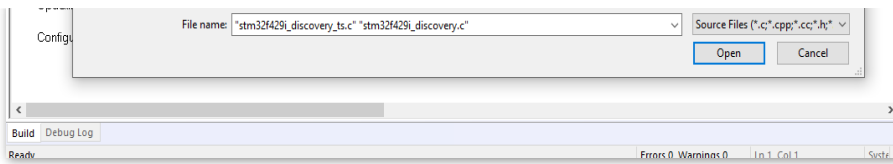


Add the following files to your project (The code for the stmpe811 touch controller is already present):

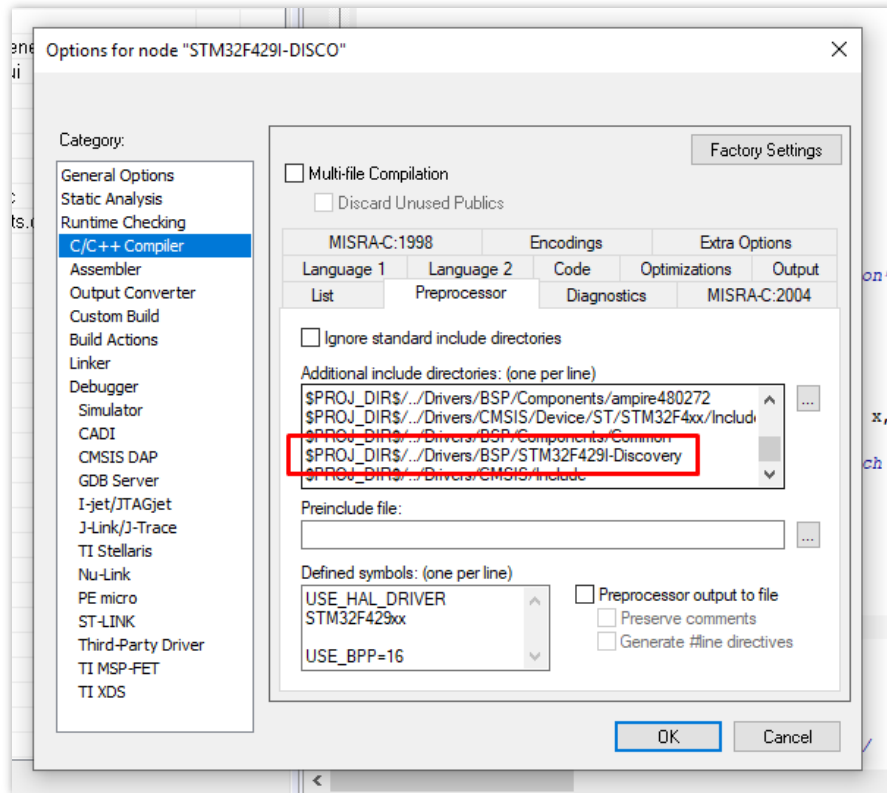
1. stm32f429i\_discovery.c
2. stm32f429i\_discovery\_ts.c



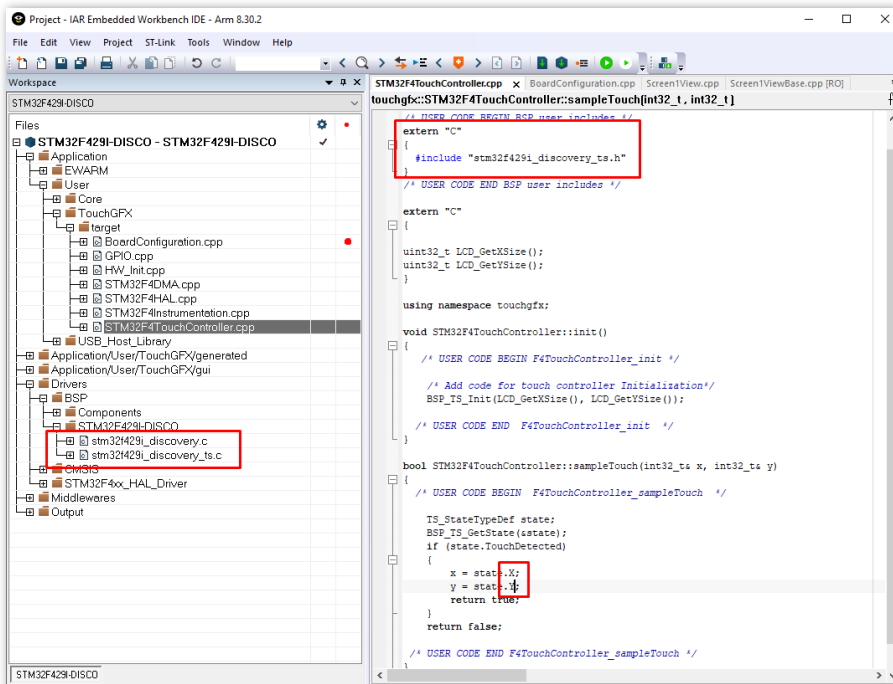




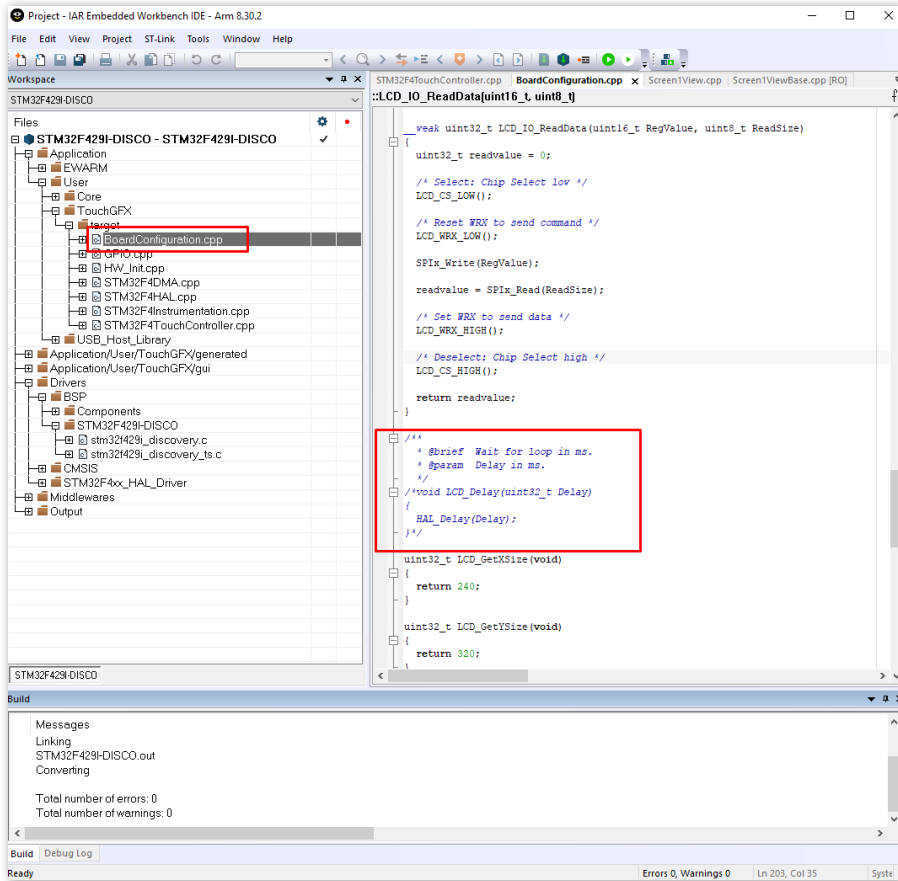
Add this location to the projects include paths:



Include the file `stm32f429i_discovery_ts.h` in `STM32F4TouchController.cpp`. The `TS_StateTypeDef` members "x" and "y" must be capitalized.



What remains now is resolving a duplicate definition for `LCD_Delay()` since this method is now also defined in `stm32f429i_discovery.o`. Commenting or removing the one inside `BoardConfiguration.cpp` should let you compile and link with 0 errors.



Programming the board now should allow you to drag the draggable element.



Was this article helpful?   2 out of 2 found this helpful

Have more questions? Please create a post on the [forum](#).