

Topic:	Getting started with the Example Packages CubeIDE
Author:	PARA
Date:	30.11.2021

1. Introduction

This document will show you what you need and how you set everything up so you can start, run and perhaps modify the example packages provided by EBS-SYSTART. The following steps are required for all our example packages and should be executed before you begin to work with the code.

2. Needed Hardware

Additional to your computer you will need the following devices to use our packages:

- 2in1-display-module by EBS-SYSTART
- ST-LINK/V2 with USB-A to USB-Mini-B cable
- 20 pin ribbon cable + DB2STL adapter by EBS-SYSTART
 - Alternative: 5 jumper wires (female/female)
- 12 V power supply, which can supply 400 mA
 - For connecting to the power socket of display module you need the “2P Pluggable Terminal Block” by Harting (manufacturer number: 14310213101000)

3. Setting Up the Display Module

The display modules need to be connected to a 12 V power supply.

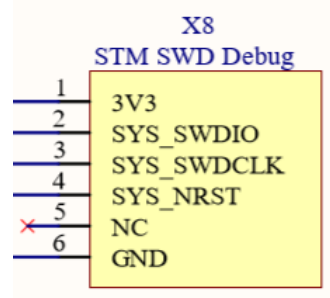
3.1 Connecting the Display Module with ST-LINK/V2

The ST-LINK/V2 is the debugger and programmer for the STM32 microcontrollers. It is used for flashing and debugging your code on the microcontroller. You need to install two drivers. STSW-LINK007 is the application used to upgrade the firmware of the ST-LINK. STSW-LINK009 is the USB driver. Install these before you go on. They can easily be found on the STMicroelectronics homepage (<https://www.st.com>) by typing in the name in the search bar.

After you have successfully installed them, you can connect the JTAG/SWD target connector of the ST-LINK/V2 with the JTAG/SWD pins of the display module. The connector consists of 20 pins but only 5 of them are needed. Pin 1, 7, 9, 15 and 20. If you use our DB2STL adapter you can just plug the 20 pin ribbon cable in the ST-LINK and the adapter and connect the adapter to the JTAG/SWD pins of the display board. Make sure that you match Pin 1 which is marked with a dot.

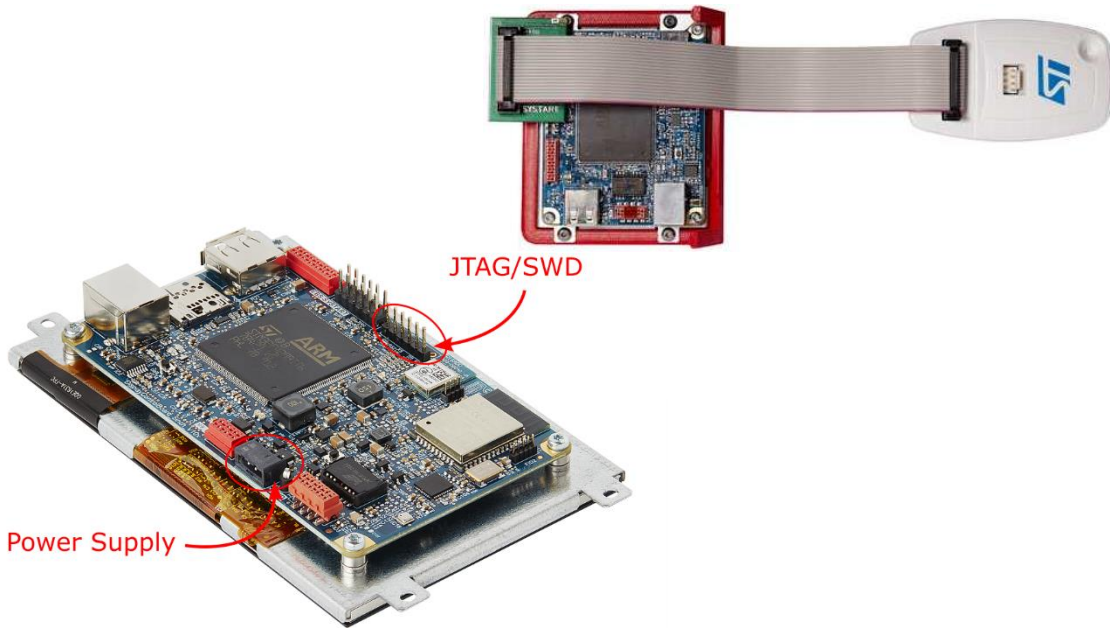
19	17	15	13	11	9	7	5	3	1	
		NRST			SWD CLK	SWD IO			VCC 3V3	
GND	20	18	16	14	12	10	8	6	4	2

Alternatively, you can use 5 individual female/female jumper wires and connect each pin individually. Connect these pins with the corresponding pins of the JTAG/SWD connector on the display module using the jumper wires. Note, that only five of the six pins on the display board are connected.



At last, you need to connect the ST-LINK with the USB-cable to your computer.

On the following picture you can see the plugs and pins for power supply and the JTAG/SWD connection. The picture shows the back of the 5 inch display module. For the 3.5 and 4.3 inch module the position of the connectors may differ a bit, but should be found easily.



4. Preparing your PC-Workspace

There are two ways you can continue. First one is to directly flash the provided binaries for demonstration purposes (4.2). Or you can import the provided source code and tinker around with it (4.3 and 4.4).

4.1 Needed Software

- STM32CubeIDE
- STM32CubeProgrammer (only for flashing binaries)

4.2 Flashing a provided binary file

Install the STM32CubeProgrammer ([Quicklink](#)). Start the program and connect your display module. Click the tab *Open file*, select the fitting binary (according to the display size) in the folder 00_bin and click the *Download* button. After programming you must reset the display, either through the Reset button or switching the power off and on.

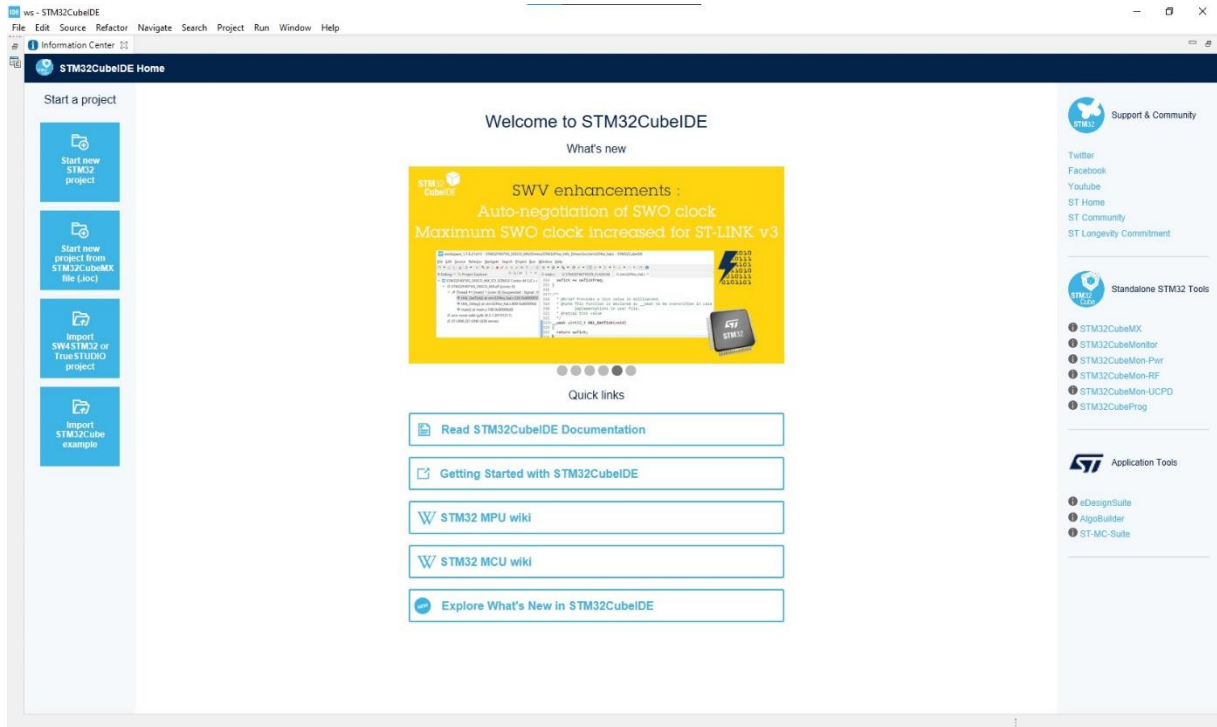
You can also choose the Erasing and Programming Tab, click on browse and chose the correct .bin File. If you now click on *Start Programm...* The MCU is programmed and reset automatically

4.3 Setting up CubeIDE

Since we are going to use STM32CubeIDE as our development environment, you need to install this free software on your computer. You can download the installer for the software and your preferred operating system on the ST.com page. ([Quicklink](#)). Together with the STM32CubeIDE the Program CubeMX is installed.

Note: All Example Packages were programmed and tested with CubeIDE Version 1.7.0 and CubeMX Version 6.3.0

After installation start CubeIDE. With every start CubeIDE will ask you to choose a so-called workspace. This is the location, where your STM32 projects will be stored. You can choose whatever you like. Afterwards you will see the Welcome Window. You can close this tab.



Now we can start to create a new project.

4.4 Importing the example packages

For creation of a new project click *File* → *New* → *Create a New STM32 Project from an existing STM32CubeMX Configuration File (.ioc)*.

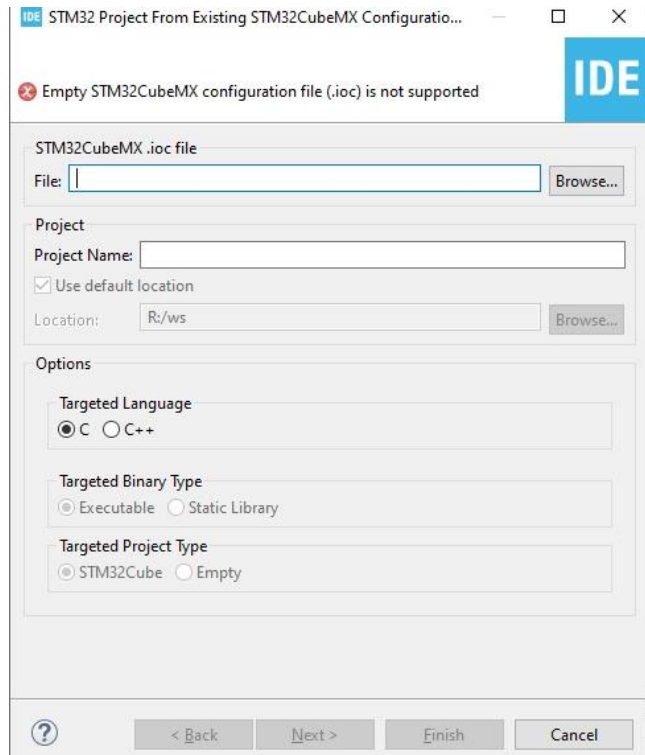
A wizard will pop up to guide you through the process.

To select your .ioc file, click *Browse...*

Then select the file that belongs to your display size, in the 00_ioc folder.

To continue click *Finish*.

Note: If you want to import another project e.g., project *CubeIDE Touchart* pick the right .ioc file in the 04_ioc folder.



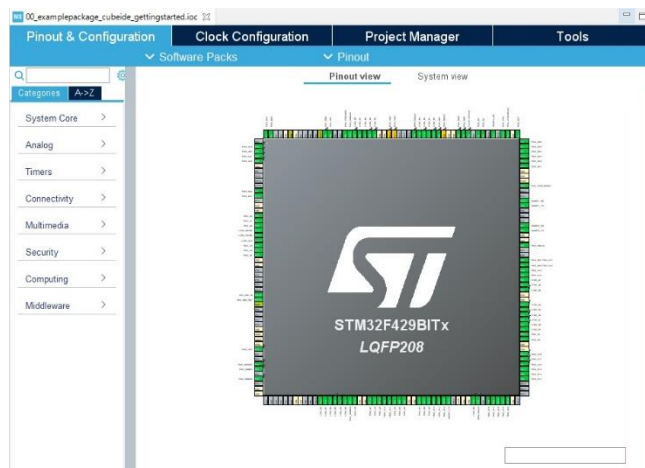
The Device Configuration Tool now starts. Wait for it to finish the code framework creation process.

When it's finished you should see the pin configuration of the STM32.

Now copy folder "Core" of the zip archive provided by www.21display.com into the project directory in your workspace.

Note: In the package "03_digitalpictureframe" you also have to copy the folders "FATFS" and "LIBJPEG".

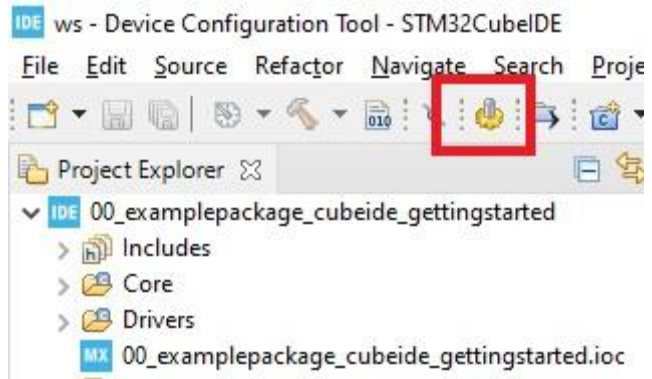
Important: Overwrite the files before you generate your code (press on the cogwheel).



When finished, click on the small yellow cogwheel icon in the menu bar.

The device configuration Tool begins to update the code.

After the Tool is done, you have to update one line of code as seen in the next chapter.



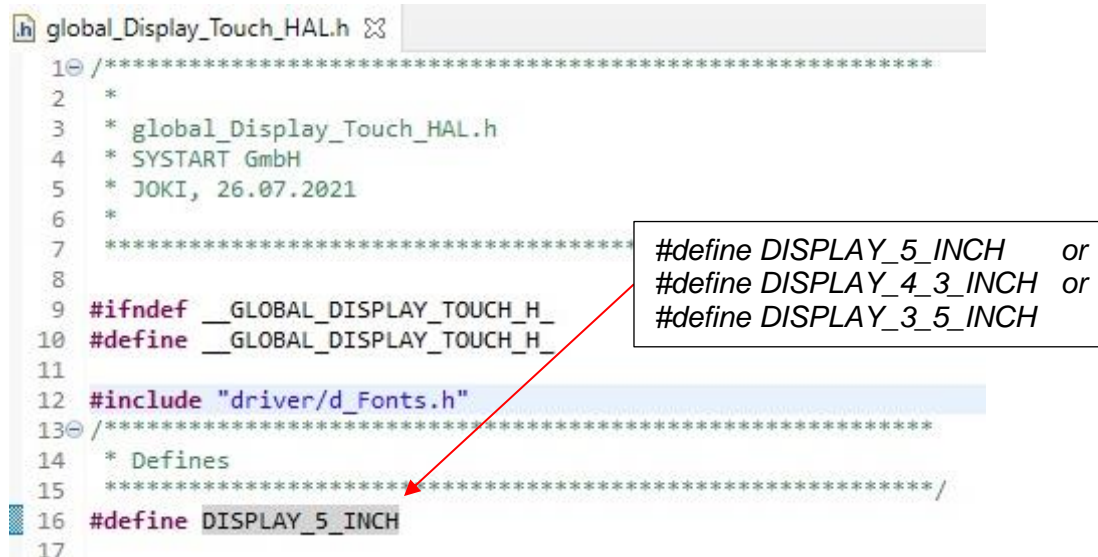
5. Hardware-specific Configuration of the Example Code

The obvious difference between the three different display modules is the display size (3.5", 4.3" and 5"). This needs to be written directly as a define in the code so that everything will be drawn in the right proportions. The drivers need these defines too, to configure the display correctly when making changes to the layers/buffers etc.

Go to the file *Core/Inc/global_Display_Touch_HAL.h*.

Here, you just need to modify the `#define DISPLAY_5_INCH` command in line 16. Depending on which display you have, you need to replace the macro with:

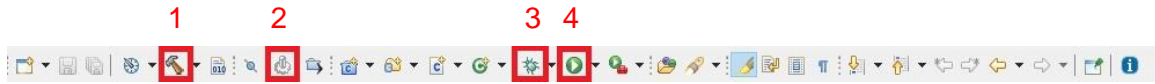
```
#define DISPLAY_5_INCH
#define DISPLAY_4_3_INCH
#define DISPLAY_3_5_INCH
```



Now everything is set up and you can continue to build the project.

6. Running Your Code

In case you haven't worked with CubelIDE before, we will give you a short summary on how you build and flash your project onto our display module.

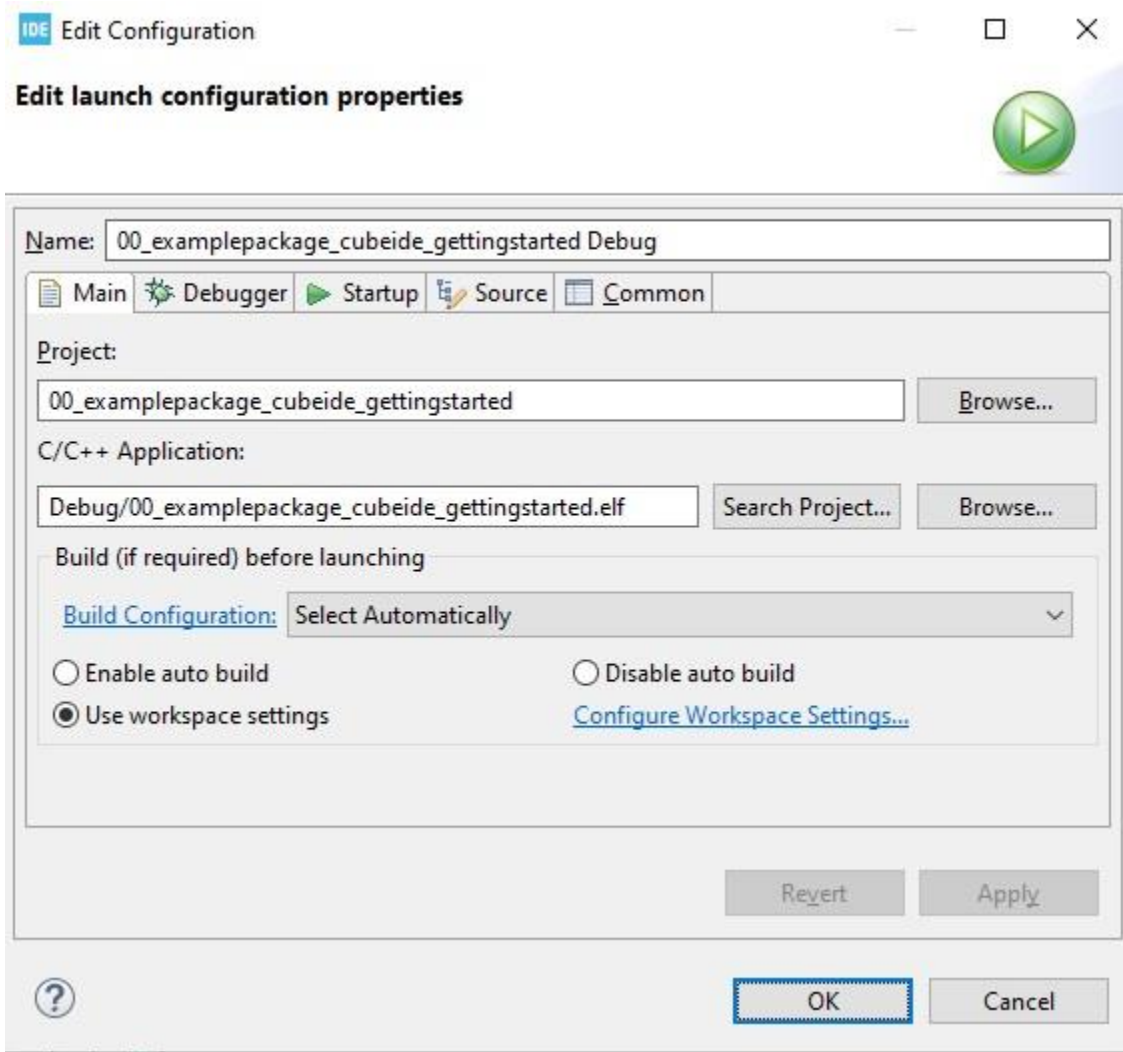


If you click the *Build*-Button (1.), your selected project will be compiled. This is a good way to check if there are any errors, like syntax errors, in your code. You can open a Dropdown Menu if you press on the little Arrow on the right side of the button where you can select building in *Debug* or in *Release* configuration.

The cogwheel button (2.) is needed if you want to change something in the .ioc file and update the CubelIDE generated code sections.

The button with the bug icon (3.) is for debugging your code, you will be prompted to configure your debug launch configuration, normally everything should work fine with the defaults.

To simply flash your code onto the microcontroller of the board, you need to press the *Run*-Button (4.). Make sure that everything, especially the ST-LINK and the power supply, are connected properly, like it is described in section 3, and that the 12 V power is switched on. When you run your project the first time, a window will pop up and ask you to select a way to run your project. Click OK.

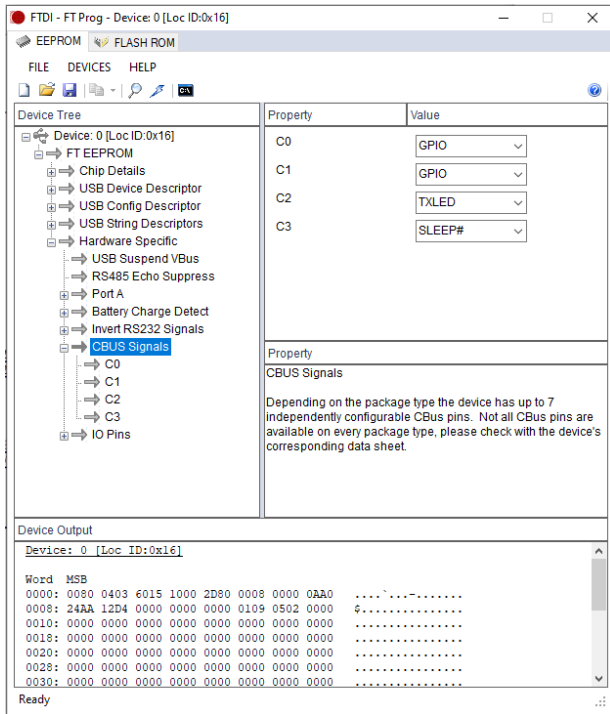


7. Troubleshoot

7.1 Black Display

If you want to start one of the examples but the display remains dark, this can have different causes. First of all, you should make sure, that the display board is supplied with power. Check if you made the necessary changes in the code depending on your display size, like it is described in section 5.

Disconnect all unnecessary peripherals, e.g. USB cables. If a USB cable connected to the USB-Type B socket seems to be the problem, this can be caused by the FTDI-chip which connects the USB-socket with the microcontroller. You may need to configure this chip. To do this, you should download the free EEPROM programming utility FT_PROG by FTDI (<https://ftdichip.com/utilities/>). Connect your computer with a USB-Type-A to USB-Type-B cable to the display board and make sure that the board is connected to its power supply. Start FT_PROG and click on the button with the magnifying glass. This will start the scan process for connected devices. Then select *Hardware Specific -> CBUS Signals*. Now check if values for the properties C0 and C1 on the right panel are set to *GPIO*. If not, change this. Afterwards click on the lightning button and click *Program* to start the programming process. After the programming has ended successfully, turn the power off, wait a few seconds and turn it on again. The FTDI-chip should now be configured properly.



Another reason for the black screen could be your ST-LINK programmer. If you have it connected with your display board but the ST-LINK is **not** connected to your PC, the microcontroller is kept in reset mode. Connect the ST-LINK with the PC or disconnect it from the display board.