

TouchGFX > Test > Flashing Guides

🙆 Martin Kjeldsen 🐱

Q Search

Configuring STM32F746G-DISCO

Introduction

CubeMX will configure a project without support for the QSPI flash and therefore are the BSP for the QSPI along with the BSP for the touchscreen missing. The linker script also needs to be updated to access the external flash via QSPI. Some changes are therefore needed to run TouchGFX on the STM32F746G-DISCO board. Configuring the CubeMX project to run with TouchGFX is done via the following steps.

- 1. Creating the initial project based on the STM32F746G-DISCO board.
- 2. Setting up QSPI and Adding TouchGFX support in CubeMX.
- 3. Add missing BSP files and update Linker script.
- 4. Configure the IAR project.

Creating the initial project

To create a project based on a STBoard, create a new project by selecting Access to Board Selector under Start My project from STBoard.

File	Window	Help
		New Project
vjects		I need to :
	-	Start My project from MCU
		ACCESS TO MCU SELECTOR
		Start My project from STBoard ACCESS TO BOARD SELECTOR
	दिर	
		ojects

In the Board Selector menu, create a project based on the STM32F746G-DISCO board by following the step below.

Recently viewed articles

Running TouchGFX without an operating system

Using other IDEs with TouchGFX

Installing TouchGFX

Known Issues

TouchGFX HAL Development

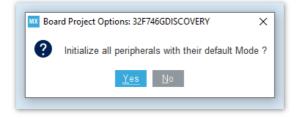
2 Select MCLI family

- 3. Select STM32F746G-DISCO board
- 4. Select "Start Project"

2. JULCE MEDIATING

📩 😨 🔀	ی پ	Features	Large Picture	Date & Resources	Datasheet	🖾 Bay	4.
٩			STMic	roelectronics \$TM32F746G Discovery	Board Support and Examples		
Vendor	,		STUCO FT	Active	Unit Price	(US\$):54.0	
Тури	~		Produc	tis in mass production	Mounted o	Invice: STM32F740NGHa	
CheckUnstreek All CheckUnstreek All CheckUnstreek All CheckUnstreek All CheckUnstreek All CheckUnstreek All STM3260 STM3261 STM3263 STM3263 STM3264	~	e (***	The discovery bit enables Advance connectively as Peatures Peatures Peatures Peatures Per Not Sector Per	-Lunk USB W and BV In measurement r utputs ones	m audio, multi-sensor support, graphics, sec	unity, video and high-speed connectivity	y feadures.
STM3267 2. STM3260 STM32L0 STM32L1		Baaste List: & Hame					
STM3267 2. STM3260 STM3260		Boards List: 6 items 3.	Denemo	we have a start to be	Martinia Status	104 804 61531	X Number Device
STM1287 2. STM1280 STM1280 STM1280 STM12814	>	Boards List: 6 items 3.) Moninera Status Active	Unit Price (253)	Number Device
STM0287 2. STM0200 STM0200 STM020 STM020.0 STM020.4 STM020.4 STM020.4+ STM020.4+	>	3.	_				Mounted Device

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



With the project initialized the project should be saved and the initial code should be generated. This is done by mowing 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

STM32CubeMX	stm3	2F746-disco-CubeMX.ioc:	STM32F746NGHx 32F746GDISCO	VERY				- 🗆
STM32 CubeMX		File	Wind	low	Help	19	f 🖸	y ₂ ★ √
Home	1	STM32F746NGHx	- 32F746GDISCOVERY	1	stm32F746-disco-CubeMX.ioc	- Proje	ct Manager	GENERATE CODE
Pinout 8	& C (onfiguration	Clock Configura	ation	Project Manage	r		Tools
		Project Settings – Project Name stm32F746-disco Project Location C:\CubeMXValida						1.

Configuring QSPI and Adding TouchGFX

support

With the project created and the initial code generated, we are ready to perform the configuring of the QSPI and add TouchGFX to the project.

Configuring the QSPI is done in the following way:

- 1. Select QUADSPI under connectivity.
- 2. Ensure that Bank1 with Quad SPI Lines is selected under QuadSPI Mode.

3. Configure the parameters for the QuadSPI as shown in the picture.

MX STM32CubeMX	(Untitl	ed*: STM32F746NGHx 32F746GDISCOVERY				
STM32 CubeMX		File	Wind	low	Help	
Home	1	STM32F746NGHx - 32F746GDIS	COVERY	1	Untitled - Pinout & Configuration	
	Pinout & Configuration				Clock Configuration	
					Additional Softwares	

https://touchgfx.zendesk.com/hc/en-us/articles/360019884752

Options Q	QUA	DSPI Mode and Configuration	
Categories A->Z		Mode	
	2. QuadSPI Mode Bank1 with Quad SPI Lines	~	
Analog >	Chip Select for Dual bank Disable	~	
Timers >			
Connectivity ~			
Connectivity			
÷			
O CAN1 O CAN2			
🔺 ETH		Configuration	3.
A FMC A I2C1	Reset Configuration		э.
Ø 12C2	🔗 Parameter Settings 🛛 😒 User Constants	📔 🥝 NVIC Settings 📔 😪 DMA Settings 📔 🤡 GPIO Settin	nas
🔥 I2C3	Configure the below parameters :		3-
1. A QUADSPI	Q Search (CrtI+F) ③		0
Ø SPI1	✓ General Parameters		
▲ SPI2 Ø SPI3	Clock Prescaler	1	
Ø SPI3 Ø SPI4	Fifo Threshold	4	
Ø SPI5	Sample Shifting	Sample Shifting Half Cycle	
Ø SPI6 Ø UART4	Flash Size	24	
Ø UART5	Chip Select High Time	6 Cycles	
Ø UART7	Clock Mode	Low	
Ø UART8	Flash ID	Flash ID 1	
▲ USART1 Ø USART2	Dual Flash	Disabled	
Ø USART3			
A USART6			
USB_OTG_FS			

Next, add TouchGFX to the project and configure to work on the STM32F746-DISCO board.

First TouchGFX is added and its parameters are updated by performing the steps below:

- 1. Select GRAPHICS under middlewares.
- 2. Select TouchGFX as the Graphics Framework and Display Parallel Interface using LTDC.
- 3. Change Physical Display Size to 480 x 272.

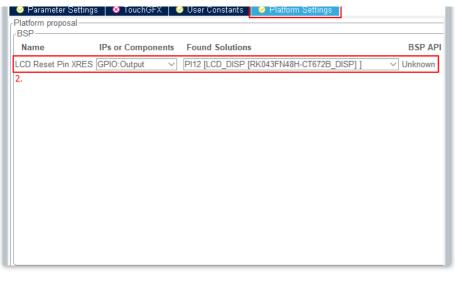
MX STM32CubeMX Untitled*: STM	32F746NGHx 32F746GDISCOVERY			
STM32 CubeMX	File	Window	Help	
Home / STM3	2F746NGHx - 32F746GDISCC	OVERY / U	Intitled - Pinout & Configuration	
Pinou	t & Configuration		Clock Configuration	
				✓ F
Options Q		GRAPHIC	S Mode and Configuration	
Categories A->Z			Mode	
A SDMMC1	2. Graphics Framework Touch	GFX		\sim
Ø SPI1	Display Interface Display Pa		LTDC	~
▲ SPI2 Ø SPI3		5		
Ø SPI3 Ø SPI4				
Ø SPI5				
 Ø SPI6 Ø UART4 				
O UART5				
Ø UART7 Ø UART8				
A USART1				
Ø USART2 Ø USART3			Configuration	
USART6	Reset Configuration			
▲ USB_OTG_FS	😔 Parameter Settings 🛛 😣	TouchGFX 🛛 😏 Us	er Constants 🛛 🔼 Platform Settings 👘	
A USB_OTG_HS	Configure the below parameters			
Multimedia >	Q Search (CrtI+F)	0		
Multimedia /		0		
Security >	✓ Stack Name			
	3. Name ✓ Physical Display Size		TouchGFX	
Computing >	Width		480	
	Height		272	
Middleware ~	✓ Frame Buffer			Ť
\$	LTDC Pixel Format(Set i	n LTDC)	LTDC_PIXEL_FORMAT_RGB565	
 FATES FREERTOS 	Frame Buffer Color Forn	nat	RGB565	
1. S GRAPHICS	Color Frame Buffer Dep		16	
LIBJPEG	Color Frame Buffer Star	t Address(Set in LTDC		
LWIP MBEDTLS	Buffers Count		Double Buffered	
PDM2PCM	 Bitmap Cache Settings Bitmap Cache Address 		0	
	Bitmap Cache Size		0	
✓ USB_HOST	Bitmap Objects Count		0	
	✓ Instrumentation			
Application >	Timer For Instrumentation	on	Timer 1	

The Reset Pin is set up under Platform settings

- 1. Select Platform Settings under Configuration
- 2. Set the LCD Reset Pin to PI12 (LCD_DIPS) by scrolling down in the drop down menu under Found Solutions

Configuration

https://touchgfx.zendesk.com/hc/en-us/articles/360019884752



Finally, we need to tel CubMX where the TouchGFX Designer is located and generated a TouchGFX Designer project.

- 1. Select TouchGFX under Configuration
- 2. Select the installed location for the TouchGFX Designer
- 3. Start the TouchGFX Designer via the button Execute

	Configuration
Reset Configuration	
📀 Parameter Settings 🛛 📀 TouchGFX	🤡 User Constants 🛛 📀 Platform Settings
Configure the below parameters :	
Q Search (CrtI+F) ()	3. Execute
 External application info 	
Name	TouchGFX Designer
2. Version	V4.10.0
✓ Location	
Executable full name	C:\touchgfx\4.10.0\designer\TouchGFXDesigner-4.10.0.exe
✓ Inputs	
Physical Display X Size	480
Physical Display Y Size	272

When selecting "Execute" CubeMX generates the TouchGFX Designer project and opens the project. For testing purpose, add an image as a background and a button, thereby confirming that the project configurations are correct, by testing that images are stored correctly in the external flash and the touch controller is working when programming the board.

Warning

 It is recommended that the "Execute" action is only performed once during project creation, because this step instantiates a new empty TouchGFX application
 resulting in existing work will be overwritten.

Add missing BSP files and update Linker script.

With the BSP files not added for the QSPI and the touchscreen, these files need to be added to the project.

The BSP files are located with the STM32F7Cube firmware. The location can be found by going to Project manager, Use Default Firmware Location.

In the firmware folder, goto to the folder BSP under Drivers, and copy the two folders

2/25/2020

Configuring STM32F746G-DISCO – TouchGFX

"Components" and "STM32746G-Discovery". In the folder for the CubeMX project create a folder named BSP under Drivers and place the two copied folder to the BSP folder.

With the QSPI flash added to the project, the linker script needs to be updated accordingly, by defining the region for the QSPI flash and what to place there

Add the following changes to EWARM\stm32f746xx_flash.icf:

```
define symbol __QSPI_start__ = 0x90000000;
define symbol __QSPI_end__ = 0x901000000;
define region QSPI_region = mem:[from __QSPI_start__ to __QSPI_end__];
place in QSPI_region { section ExtFlashSection};
```

Resulting in a linker script as shown below

```
/*###ICF### Section handled by ICF editor, don't touch! ****/
/*-Editor annotation file-*/
/* IcfEditorFile="$TOOLKIT DIR$\config\ide\IcfEditor\cortex v1 0.xml"
/*-Specials-*/
define symbol ICFEDIT intvec start = 0x08000000;
/*-Memory Regions-*/
define symbol __ICFEDIT_region_ROM_start__ = 0x08000000;
define symbol __ICFEDIT_region_ROM_end__ = 0x080FFFFF;
define symbol __ICFEDIT_region_RAM_start__ = 0x20000000;
define symbol __ICFEDIT_region_RAM_end__ = 0x2004FFFF;
define symbol ICFEDIT region ITCMRAM start = 0x00000000;
define symbol __ICFEDIT_region_ITCMRAM_end_ = 0x00003FFF;
define symbol __QSPI_start__ = 0x90000000;
define symbol QSPI end = 0x90100000;
/*-Sizes-*/
define symbol __ICFEDIT_size_cstack__ = 0x400;
define symbol ICFEDIT size heap = 0x200;
/**** End of ICF editor section. ###ICF###*/
define memory mem with size = 4G;
define region ROM_region = mem:[from __ICFEDIT_region_ROM_start__ to _
define region RAM_region = mem:[from __ICFEDIT_region_RAM_start__ to __
define region ITCMRAM_region = mem:[from __ICFEDIT_region_ITCMRAM_star
define region QSPI_region = mem:[from __QSPI_start__ to __QSPI_end__];
define block CSTACK with alignment = 8, size = __ICFEDIT_size_cstack_
define block HEAP with alignment = 8, size = __ICFEDIT_size_heap__ { }
initialize by copy { readwrite };
do not initialize { section .noinit };
place at address mem:__ICFEDIT_intvec_start__ { readonly section .intve
place in ROM_region { readonly };
place in RAM region { readwrite,
block CSTACK, block HEAP };
place in QSPI_region { section ExtFlashSection};
```

P

Modifying the EWARM Project

- 1. Add components from Cube package to project folder
- 2. Add include paths to EWARM project
- 3. Add source files to EWARM project

Open the IAR project in EWARM -> Project.eww

In the project add a sub-group under Drivers called BSP and add the three files *stm32746g_discovery_c, stm32746g_discovery_qspi.c* and *stm32746g_discovery_ts.c* which is located in STM32F7Cube installation folder, [STM32F7Cube]Drivers/BSP/STM32746G-Discovery and the file *ft5336.c* located in [STM32F7Cube]/Drivers/BSP/Components/ft5336

Add the "STM32746G-Discovery" folder to included directories by opening the options for the project wither by pressing Alt + F7 or selecting options under Project in the top bar.

In the options menu adding the directory is done in the following way.

- 1. Select the category C/C++ Compiler
- 2. Go to the tab Preprocessor
- 3. Press the button labeled ... to open Edit Include Directories
- 4. Scroll down and click on <Click to add> to add a new directory
- 5. Navigate to the "STM32746G-Discovery" folder and select the folder

Add Code

- 1. Open STM32F7TouchController.cpp
- 2. Add include for the touchscreen bsp file inside the "USER CODE" section
 - #include <stm32746g_discovery_ts.h>
- 3. In the STM32F7TouchController::Init remove the comment for "BSP_TS_Init" function inside the "USER CODE" section.
- 4. In the STM32F7TouchController::sampleTouch remove the comment inside the "USER CODE" section.

```
#include "STM32F7TouchController.hpp"
```

```
/* USER CODE BEGIN BSP user includes */
#include <stm32746g_discovery_ts.h>
/* USER CODE END BSP user includes */
extern "C"
{
    uint32_t LCD_GetXSize();
    uint32_t LCD_GetYSize();
}
using namespace touchgfx;
void STM32F7TouchController::init()
{
```

```
/* Add code for touch controller Initialization */
BSP TS Init(LCD GetXSize(), LCD GetYSize());
/* USER CODE END F4TouchController_init */
}
bool STM32F7TouchController::sampleTouch(int32_t& x, int32_t& y)
{
/* USER CODE BEGIN F4TouchController sampleTouch */
TS_StateTypeDef state = { 0 };
BSP_TS_GetState(&state);
if (state.touchDetected)
 {
x = state.touchX[0];
y = state.touchY[0];
return true;
 }
return false;
/* USER CODE END F4TouchController sampleTouch */
}
```

<u>тпт</u> с

Go to main.c to add code for initializing the QSPI.

ODEN CODE DEGTIN I + I OUCHCOHCI OTTEL

Include the QSPI header inside the "USER CODE BEGIN Includes" section by adding

#include <stm32746g_discovery_qspi.h>

The initialization code is added under the *static void MX_QUADSPI_Init(void)* function and inserted in "USER CODE BEGIN QUADSPI_Init 2" section. Add the code below:

```
BSP_QSPI_Init();
```

BSP_QSPI_MemoryMappedMode();
HAL_NVIC_DisableIRQ(QUADSPI_IRQn);

```
MPU_Region_InitTypeDef MPU_InitStruct;
MPU_InitStruct.Enable = MPU_REGION_ENABLE;
MPU_InitStruct.BaseAddress = 0x90000000;
MPU_InitStruct.Size = MPU_REGION_SIZE_256MB;
MPU_InitStruct.AccessPermission = MPU_REGION_FULL_ACCESS;
MPU_InitStruct.IsBufferable = MPU_ACCESS_NOT_BUFFERABLE;
MPU_InitStruct.IsCacheable = MPU_ACCESS_NOT_CACHEABLE;
MPU_InitStruct.IsShareable = MPU_ACCESS_NOT_SHAREABLE;
MPU_InitStruct.Number = MPU_REGION_NUMBER2;
MPU_InitStruct.TypeExtField = MPU_TEX_LEVEL0;
MPU_InitStruct.SubRegionDisable = 0x00;
MPU_InitStruct.DisableExec = MPU_INSTRUCTION_ACCESS_ENABLE;
```

```
HAL_MPU_ConfigRegion(&MPU_InitStruct);
```

2/25/2020	Configuring STM32F746G-DISCO – TouchGFX
/* Contigure the MPU attributes as WT to	r QSPI (used 16Mbytes) */
MPU_InitStruct.Enable = MPU_REGION_ENABL	Ε;
<pre>MPU_InitStruct.BaseAddress = 0x90000000;</pre>	
MPU_InitStruct.Size = MPU_REGION_SIZE_16	MB; /* NOTE! Change this if you
<pre>MPU_InitStruct.AccessPermission = MPU_REG</pre>	GION_FULL_ACCESS;
MPU_InitStruct.IsBufferable = MPU_ACCESS	_NOT_BUFFERABLE;
<pre>MPU_InitStruct.IsCacheable = MPU_ACCESS_0</pre>	CACHEABLE;
MPU_InitStruct.IsShareable = MPU_ACCESS_I	NOT_SHAREABLE;
MPU_InitStruct.Number = MPU_REGION_NUMBE	R3;
MPU_InitStruct.TypeExtField = MPU_TEX_LE	VEL0;
<pre>MPU_InitStruct.SubRegionDisable = 0x00;</pre>	
MPU_InitStruct.DisableExec = MPU_INSTRUC	TION_ACCESS_ENABLE;
HAL_MPU_ConfigRegion(&MPU_InitStruct);	
HAL_MPU_Enable(MPU_PRIVILEGED_DEFAULT);	
•	•

The configuration is done and you should now be able to compile and flash the STM32F746-DISCO board.

Was this article helpful?	(•	1 out of 3 found this helpful	f	9	in

Have more questions? Please create a post on the forum.