



基于STM32CubeMX的GUI应用开发 ——TouchGFX版

May, 2019



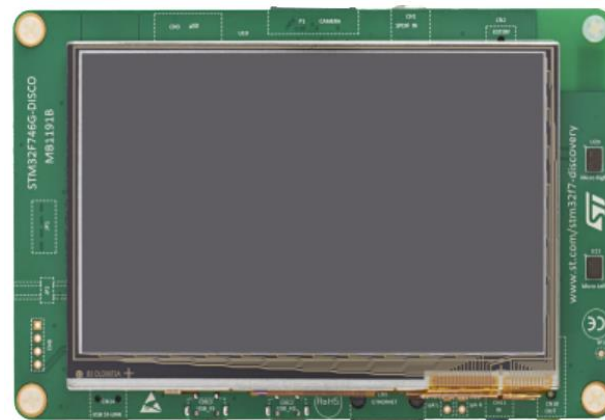
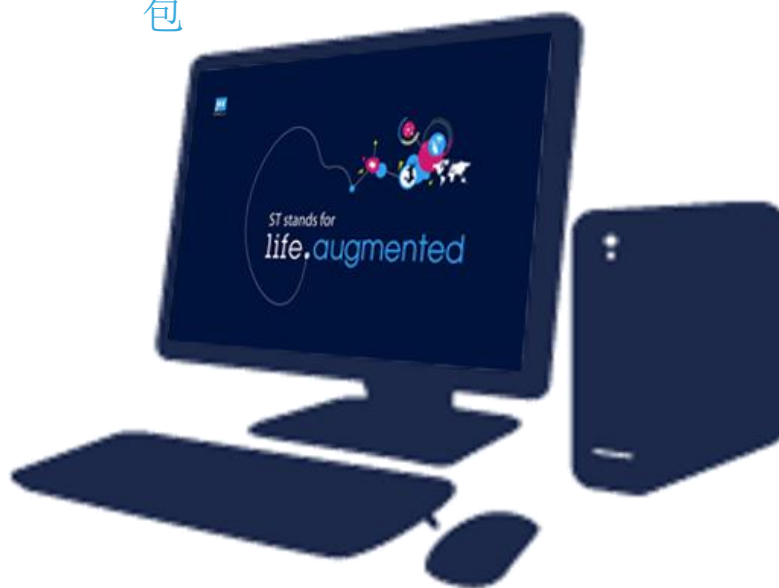
- 熟悉STM32CubeMX中配置FMC、LTDC、FreeRTOS、GRAPHICS
- 了解如何使用STM32CubeMX创建包含TouchGFX的工程
- 了解TouchGFX Designer如何实现HelloWorld的显示
- 利用STM32CubeMX开发TouchGFX应用

• 主机

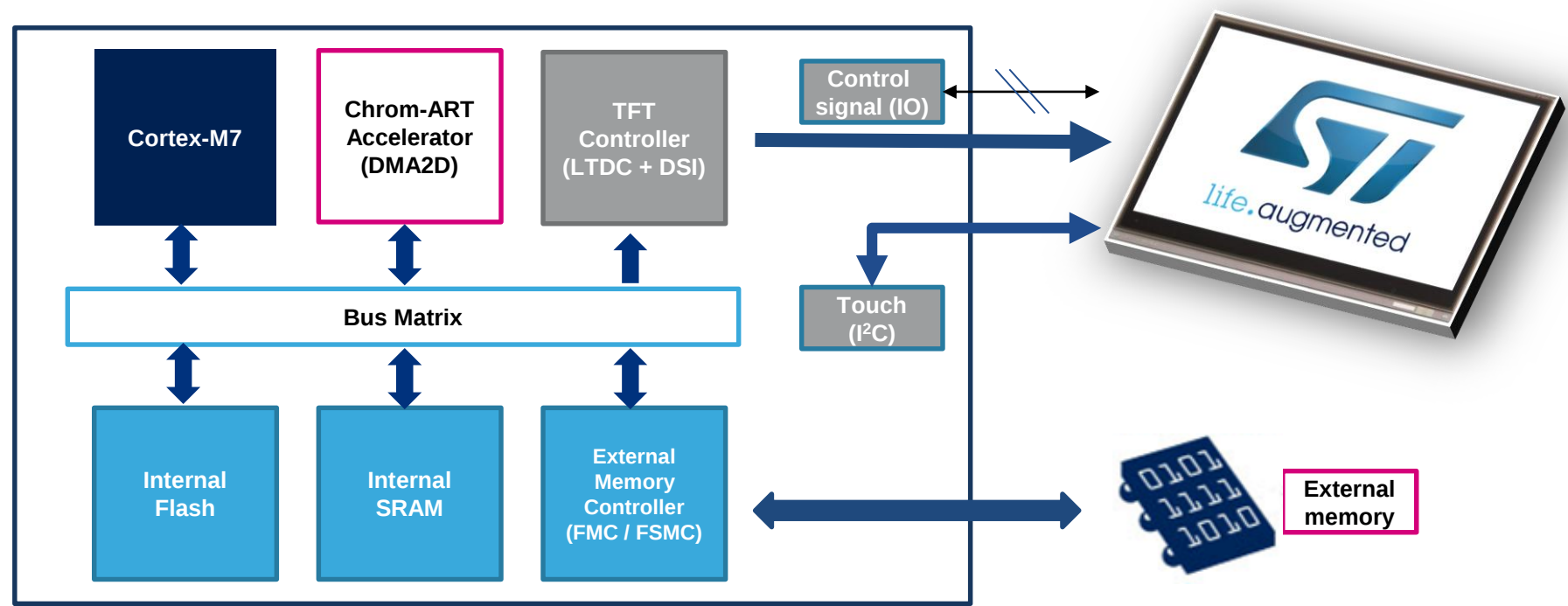
- 安装STM32CubeMX 5.1.0
- 安装STM32CubeProgrammer或者ST-Link Utility
- 安装对应的IDE（IAR8.32）
- TouchGFX designer 4.10.0
- 在STM32CubeMX中载入STM32Cube F7 1.15.0 软件包

• GUI板

- STM32F746G-DISCO（演示用）
 - STM32F746NGH6
 - 272x480 LCD屏
 - 外扩SDRAM



• 应用框架



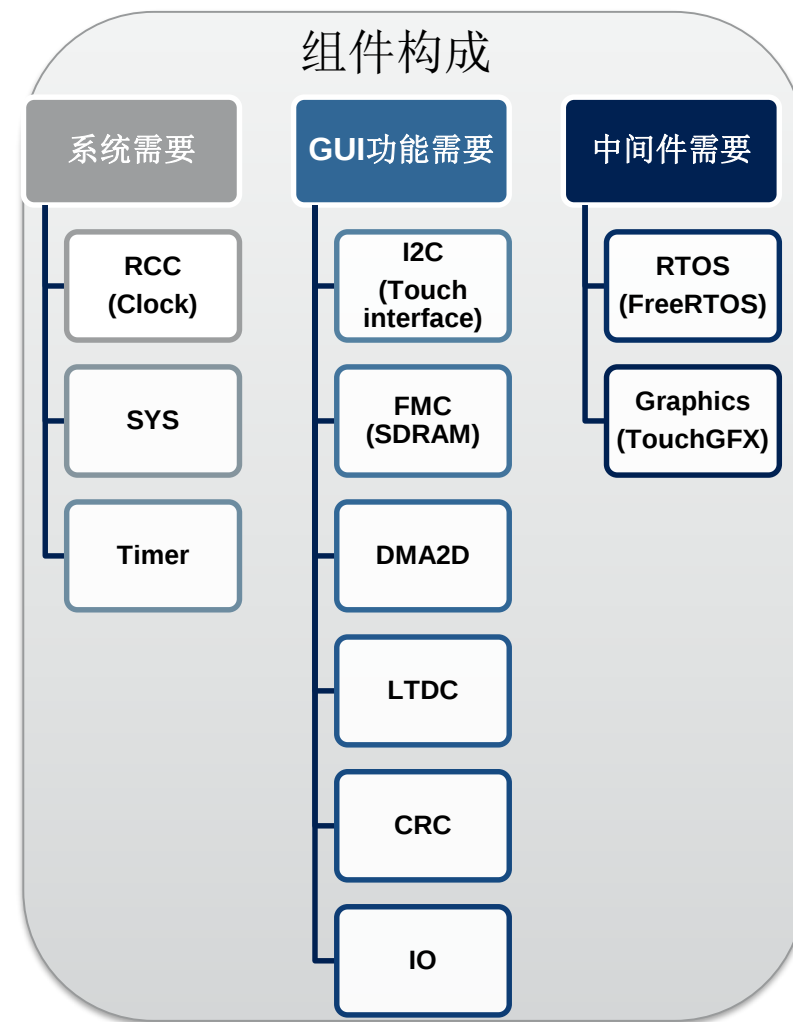


实现示例（续）

6

- IP组件（系统需要）

- 利用STM32CubeMX开发TouchGFX应用，在STM32CubeMX中的主要工作，集中在对使用的IP组件，根据应用需要、硬件连接、器件支持特性等进行配置。
- 演示例中，以STM32F746-DISCO板作为目标板，需要用到组件构成如右图所示。





- 演示例中，开发流程如下所示。
- STM32CubeMX覆盖了红色框中部分。

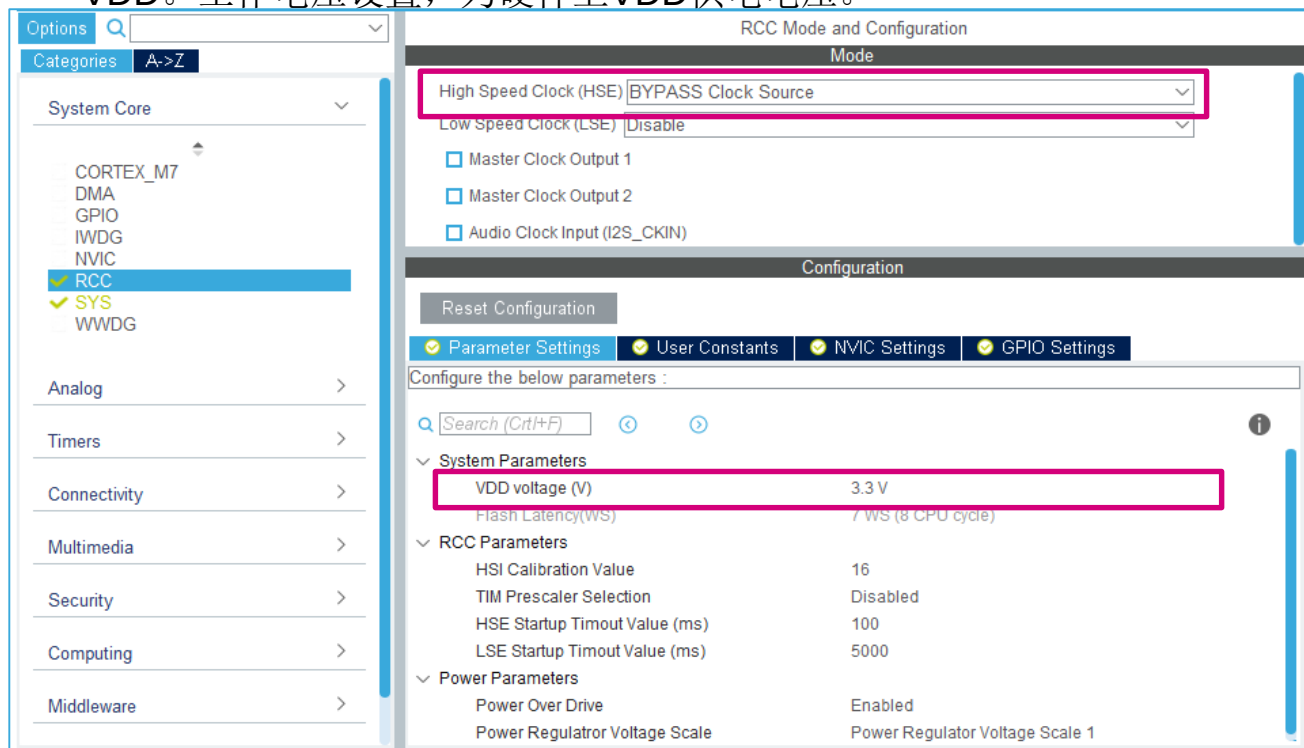




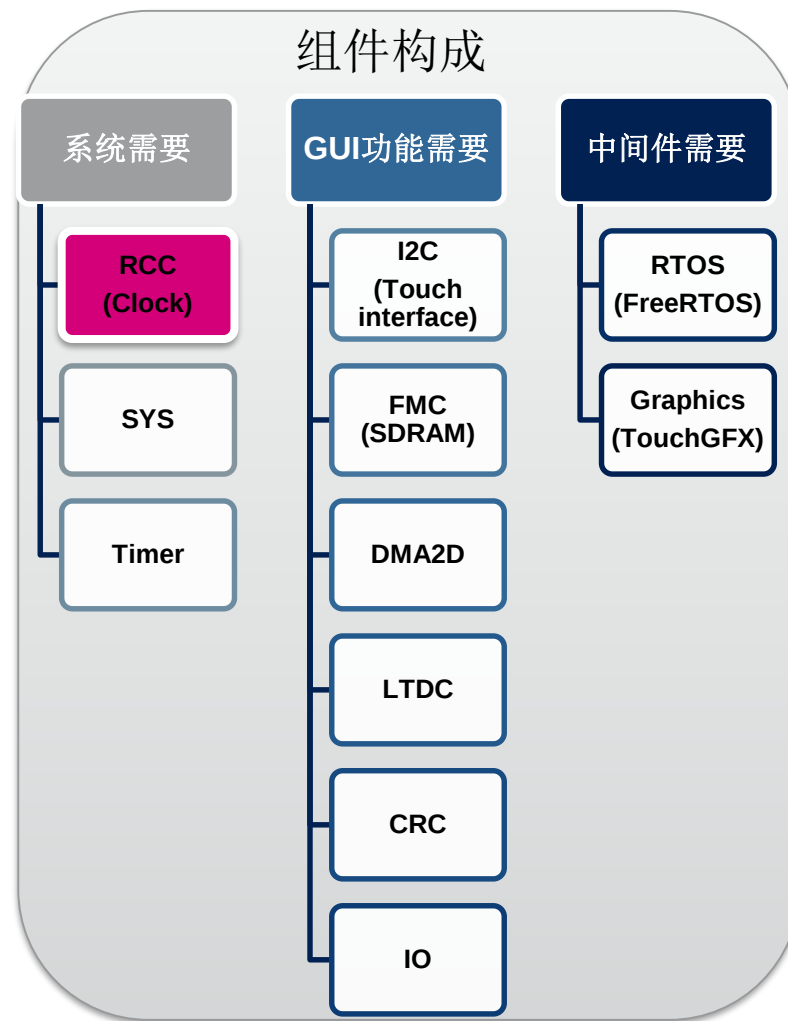
• 组件（系统需要）

• RCC

- 时钟源。根据应用选择。
 - 采用内部时钟，不用选择；
 - 采用外部高速有源时钟，选择BYPASS Clock Source；
 - 采用外部高速无源时钟，选择Crystal/Ceramic Resonator
- VDD。工作电压设置，为硬件上VDD供电电压。



组件构成





实现示例（续）

9



• 组件（系统需要）

• SYS

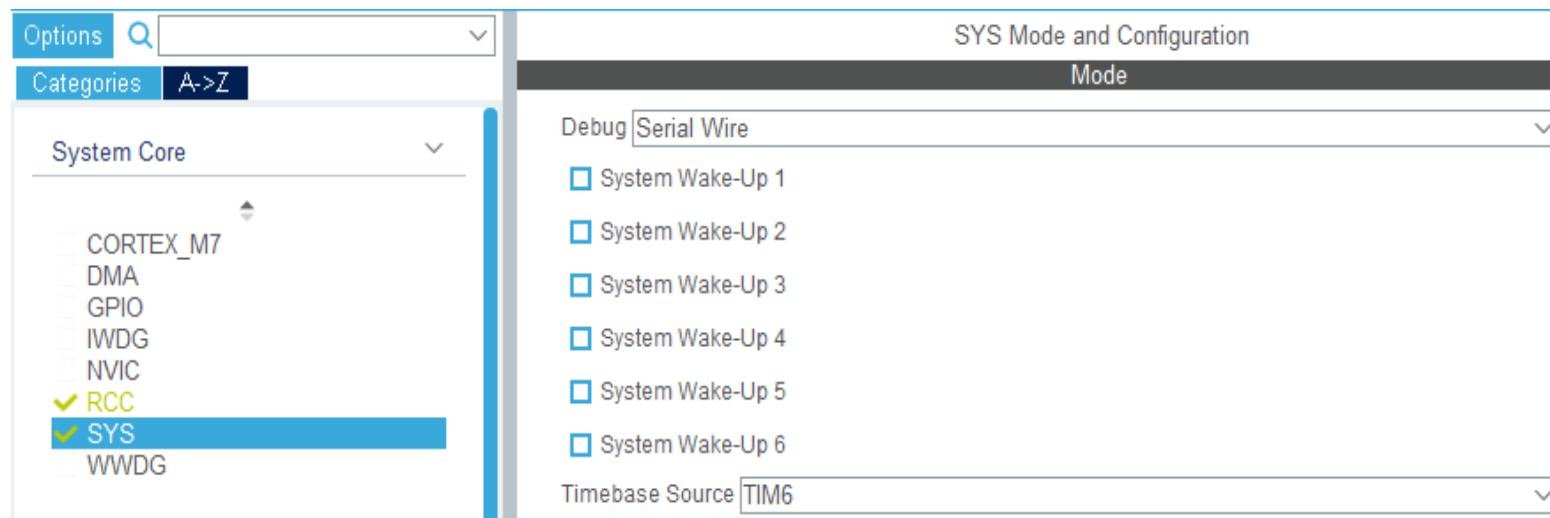
• Debug

- 调试接口。

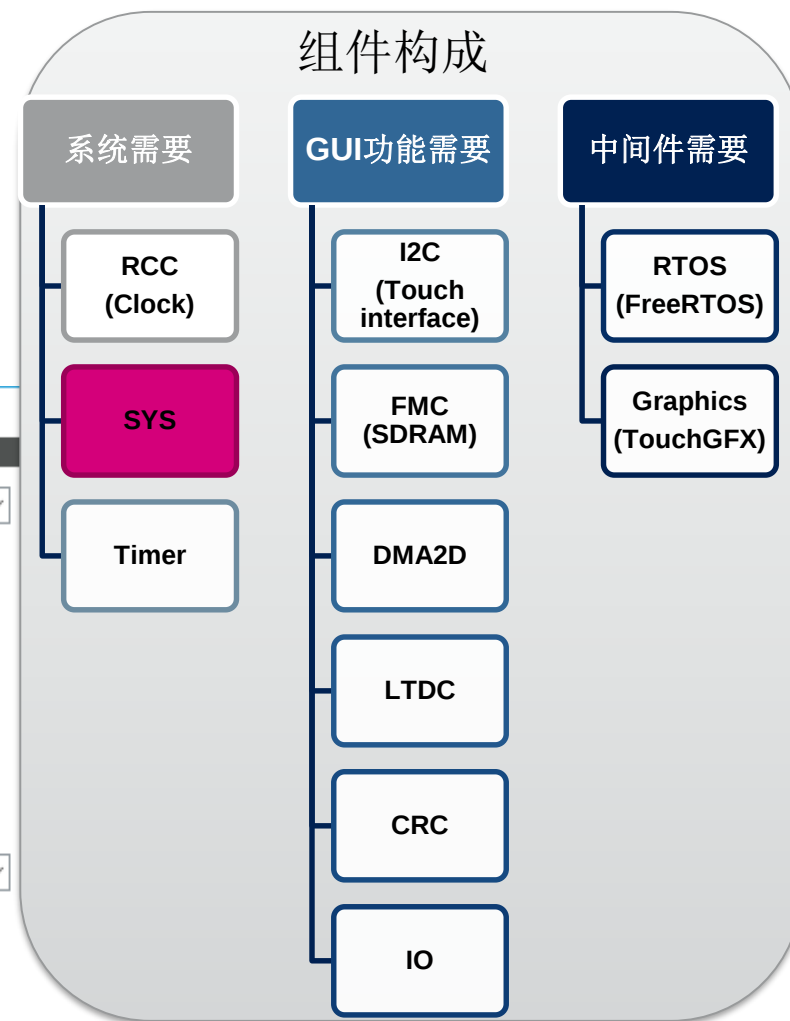
• Timebase source

- HAL库延时函数HAL_Delay关联的定时器

注：Timebase source不建议选择Systick。应用中涉及到FreeRTOS



组件构成



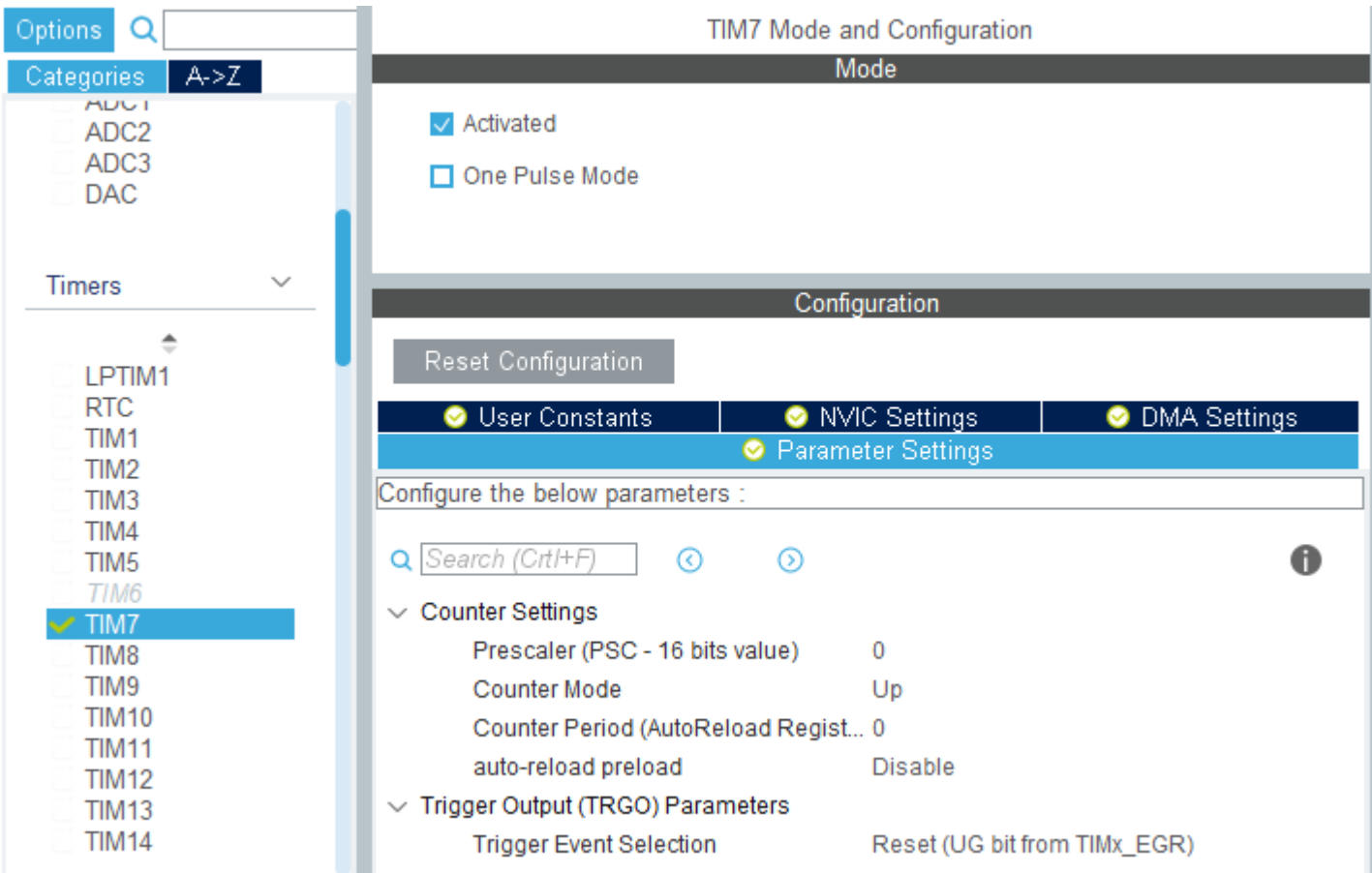


实现示例（续）

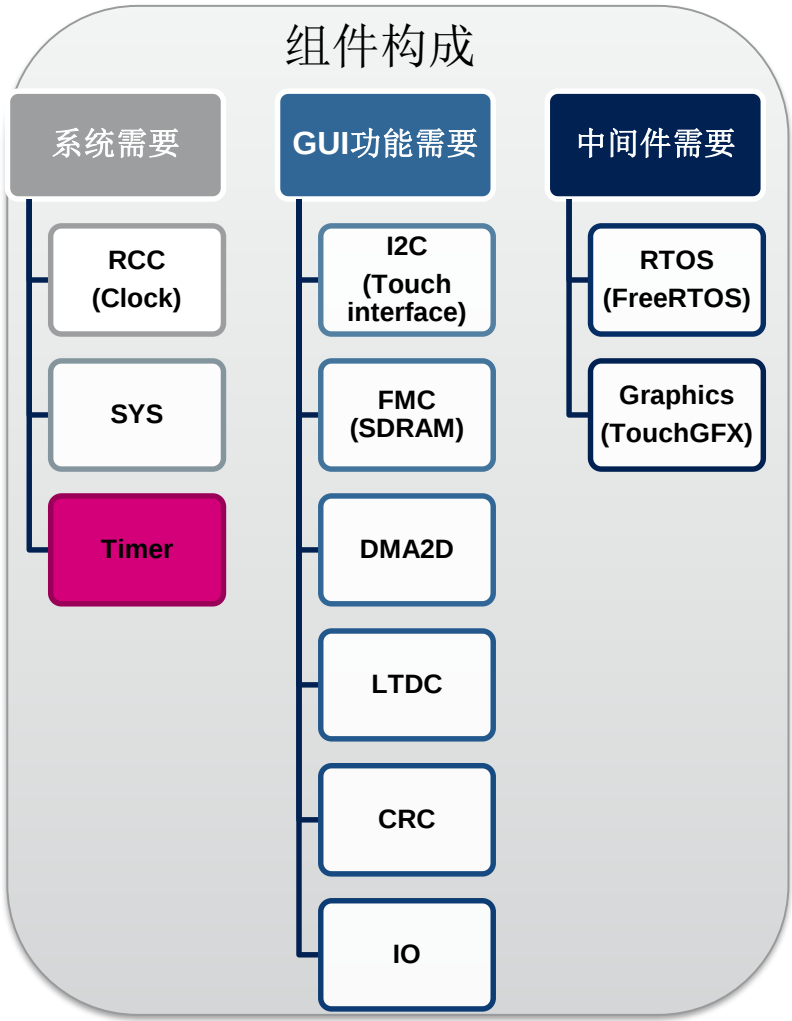


• 组件（系统需要）

• Timer



组件构成

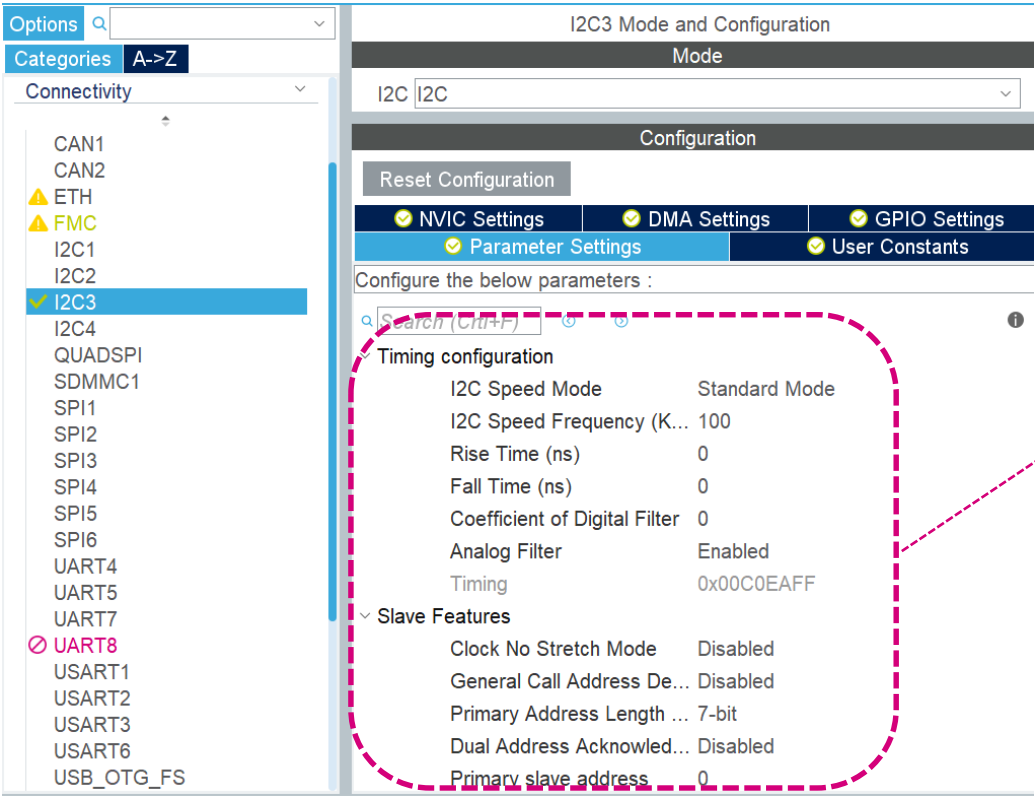




• 组件（GUI功能需要）

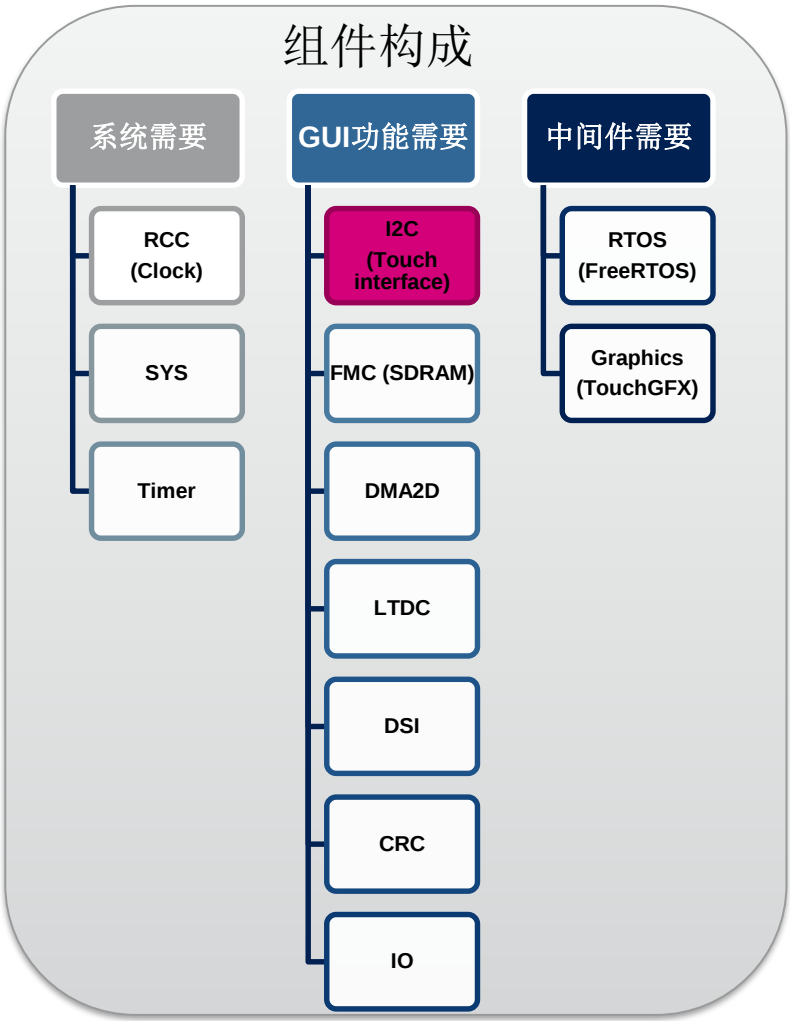
• I2C

- 触摸芯片访问接口（在STM32F746G-DISCO上，与I2C3连接）
- 选择&配置I2C，如下图



根据触摸板控制器I2C接口参数要求，完成参数配置。

- 左图参数对应为演示例中采用的触摸控制器FT5336





• 组件（GUI功能需要）

• I2C

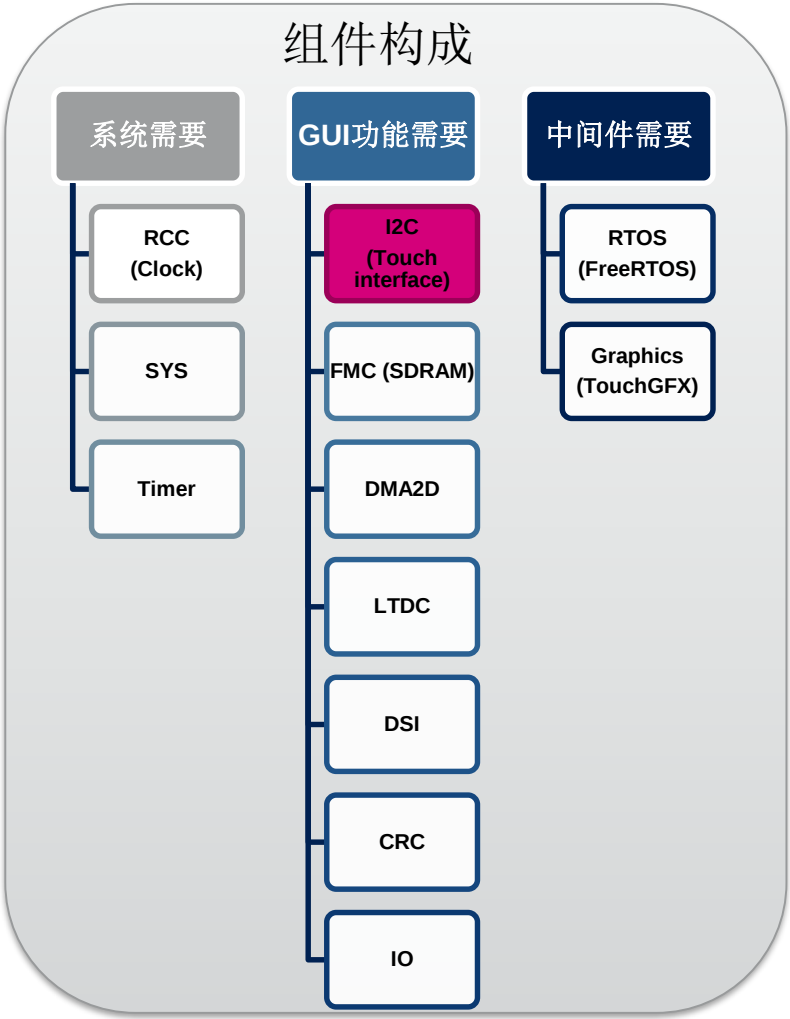
- 由于STM32 外设为了灵活性，提供了不同引脚实现同一个外设功能。在应用时，根据硬件连接，选择具体关联引脚
- 默认关联的I2C接口引脚与实际硬件采用可能不同，检查&调整

表. 触摸面板接口 连接情况

功能	名称	引脚				描述
		引脚号	模式	上下拉	速度	
触摸屏 (FT 5366)	LCD_INT	PI13	IT_FALLING	PULL_UP	FAST	触摸触发中断信号
	LCD_SDA	PH8	AF4_I2C3	NOPULL	FAST	I2C数据线
	LCD_SCL	PH7	AF4_I2C3	NOPULL	FAST	I2C时钟线

注：演示例中，采用轮询的方式采集触摸坐标信息，没有配置和使用 LCD_INT功能。

组件构成





实现示例（续）

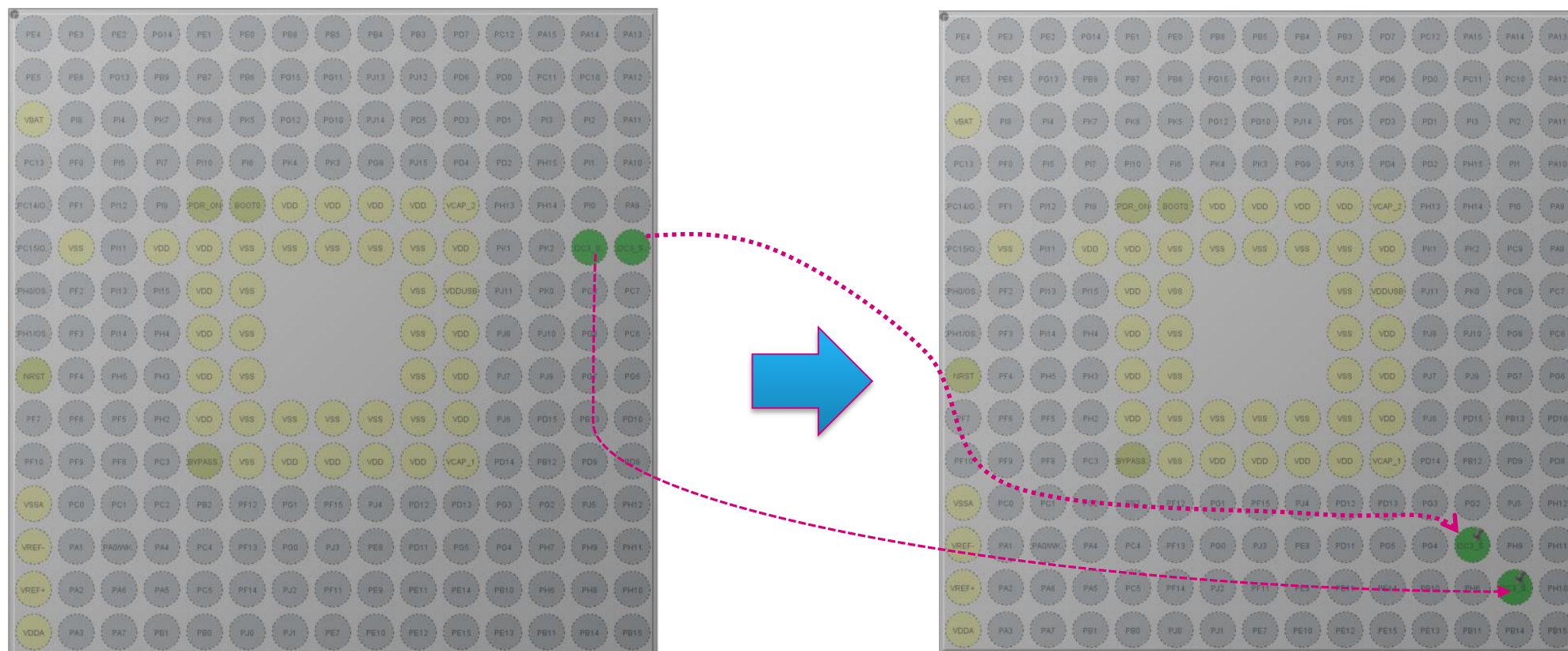
13



• 组件（GUI功能需要）

• I2C

- 默认I2C3关联引脚与实际硬件连接不一致，调整与硬件连接保持一致。

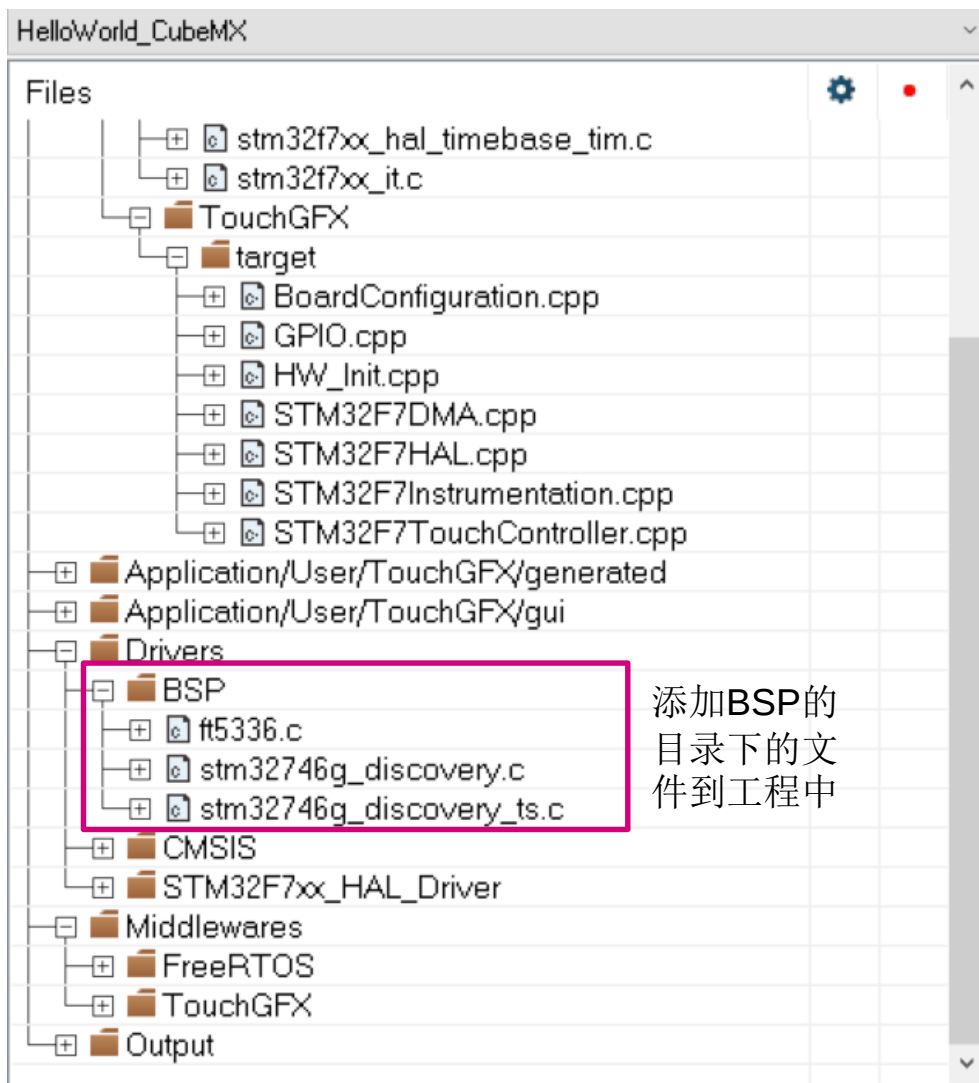




实现示例（续）

14

需要移植ft5336的驱动到生成的工程中
代码在如下的压缩包

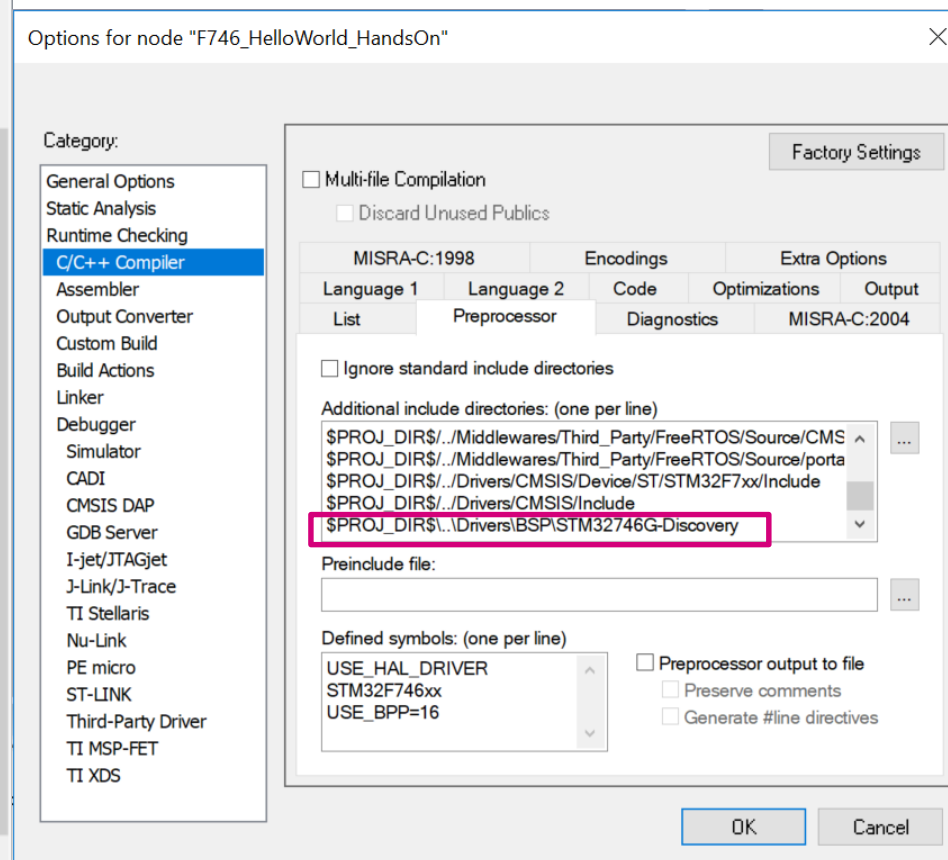


添加BSP的
目录下的文件到工程中



FT5336_driver_porting.zip

1. 拷贝BSP到driver下。
2. 用比较工具比较
\\TouchGFX\\target\\STM32F7TouchController.cpp,然后覆盖整个文件

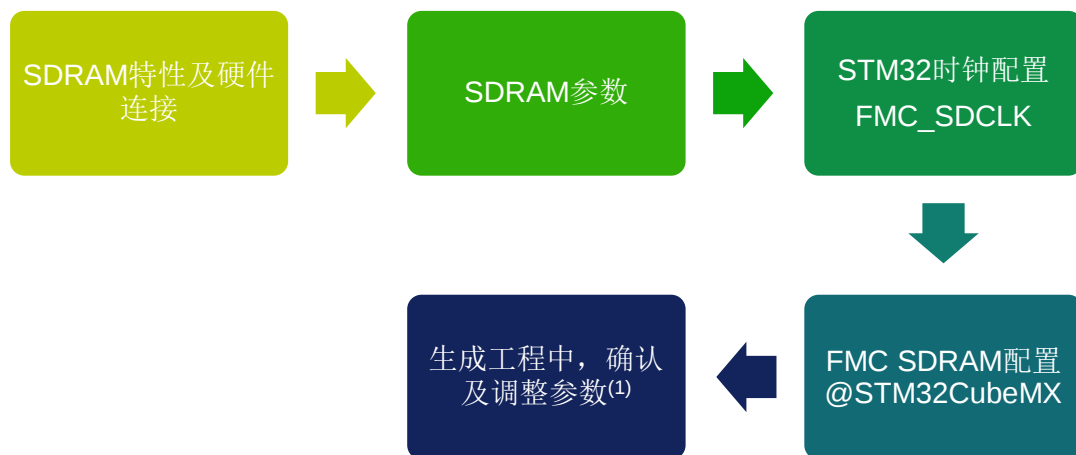




• 组件（GUI功能需要）

• FMC（外部存储器）

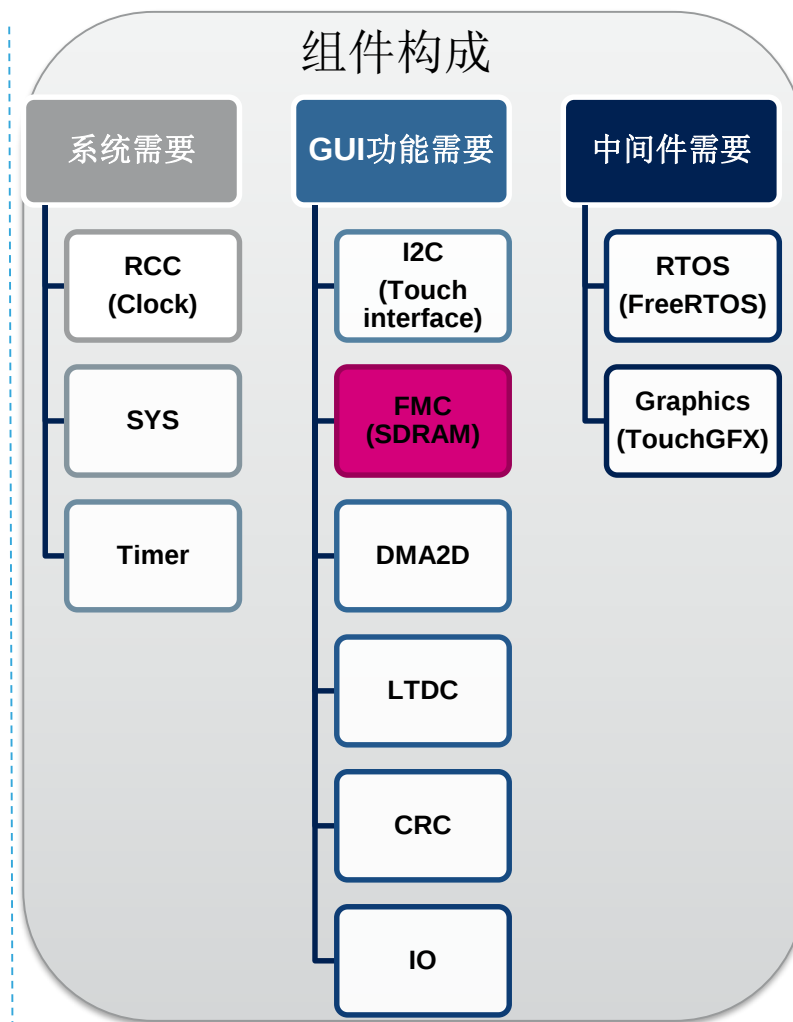
- 不同的存储器、硬件连接和时钟配置，对应FMC配置参数不同。
- 首先，呈现在STM32CubeMX上FMC配置。可以了解FMC主要的一些配置项。
- 然后，以STM32F746-Disco板上的SDRAM (型号 **MT48LC4M32B2B5-6A**)为例，介绍如何根据实际情况，获取配置参数。



FMC配置流程图（以演示例中采用的SDRAM为例）

实现示例（续）

15





• 组件（GUI功能需要）

• FMC（外部存储器）

• FMC配置（STM32CubeMX中）

Options

Categories A-Z

- CAN1
- CAN2
- ETH
- FMC
- I2C1
- I2C2
- I2C3
- I2C4
- QUADSPI
- SDMMC1
- SPI1
- SPI2
- SPI3
- SPI4
- SPI5
- SPI6
- UART4
- UART5
- UART7
- UART8
- USART1
- USART2
- USART3
- USART6
- USB_OTG_FS
- USB_OTG_HS

Multimedia

- DCMI
- DMA2D
- HDMI_CEC
- I2S1
- I2S2
- I2S3
- LTDC
- SAI1
- SAI2
- SPDIFRX

FMC Mode and Configuration

Mode

- > NOR Flash/PSRAM/SRAM/ROM/LCD 1
- > NOR Flash/PSRAM/SRAM/ROM/LCD 2
- > NOR Flash/PSRAM/SRAM/ROM/LCD 3
- > NOR Flash/PSRAM/SRAM/ROM/LCD 4
- > NAND Flash 1
- > SDRAM 1
 - Clock and chip enable SDCKE0+SDNE0
 - Internal bank number 4 banks
 - Address 12 bits Max: 13 bits
 - Data 16 bits
 - Byte enable 16-bit byte enable

Configuration

Reset Configuration

SDRAM 1 User Constants NVIC Settings GPIO Settings

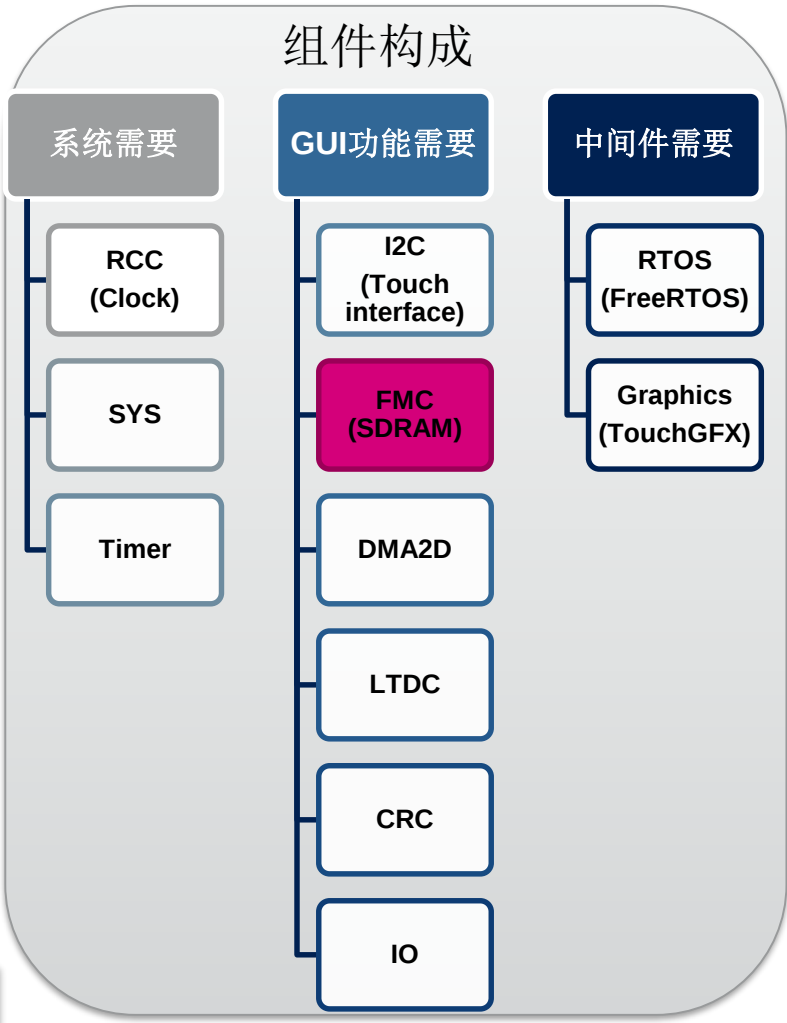
Configure the below parameters :

SDRAM control	
Bank	SDRAM bank 1
Number of column address bits	8 bits
Number of row address bits	12 bits
CAS latency	3 memory clock cycles
Write protection	Disabled
SDRAM common clock	2 HCLK clock cycles
SDRAM common burst read	Enabled
SDRAM common read pipe delay	0 HCLK clock cycle

SDRAM timing in memory clock cycles	
Load mode register to active delay	2
Exit self-refresh delay	7
Self-refresh time	4
SDRAM common row cycle delay	7
Write recovery time	5
SDRAM common row precharge delay	2
Row to column delay	2

NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
FMC global interrupt	☒	0	0

组件构成





• 组件（GUI功能需要）

- FMC（外部存储器）
- 配置参数获取

在进行FMC的配置前，需要先了解与FMC关联的SDRAM硬件连接情况。

- 1. 根据SDRAM硬件支持和使用情况，配置SDRAM1
 - MT48LC4M32B2B5-6A支持32位数据线，硬件上采用了16位数据线
 - MT48LC4M32B2B5-6A内部含有4个Bank，硬件上采用了4个Bank
 - ...

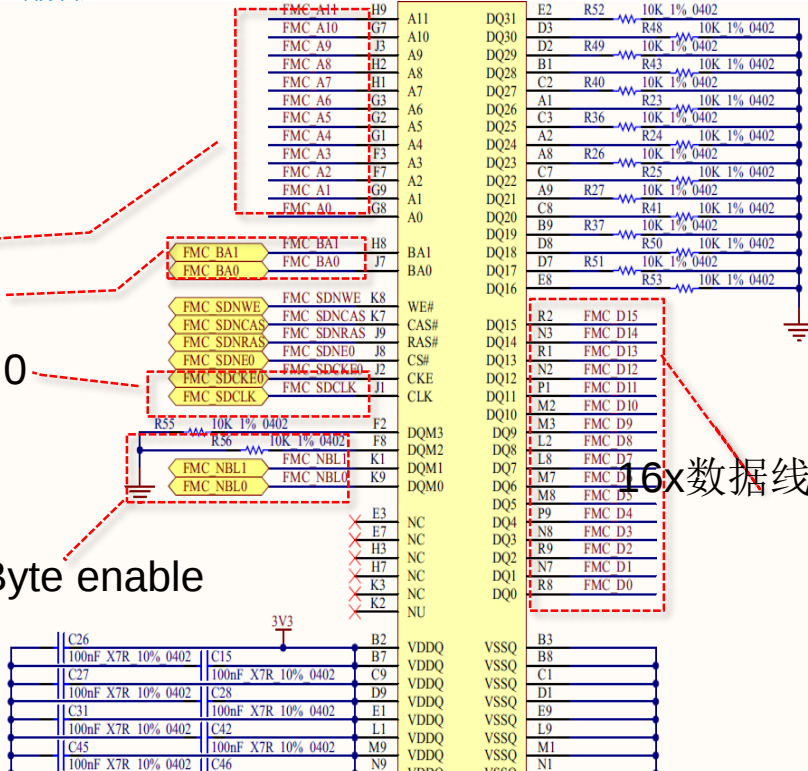
Parameter	4 Meg x 32
Configuration	1 Meg x 32 x 4 banks
Refresh count	4K
Row addressing	4K A[11:0]
Bank addressing	4 BA[1:0]
Column addressing	256 A[7:0]

图 摘自 MT48LC4M32B2数据手册

12x地址线
4x内部Banks

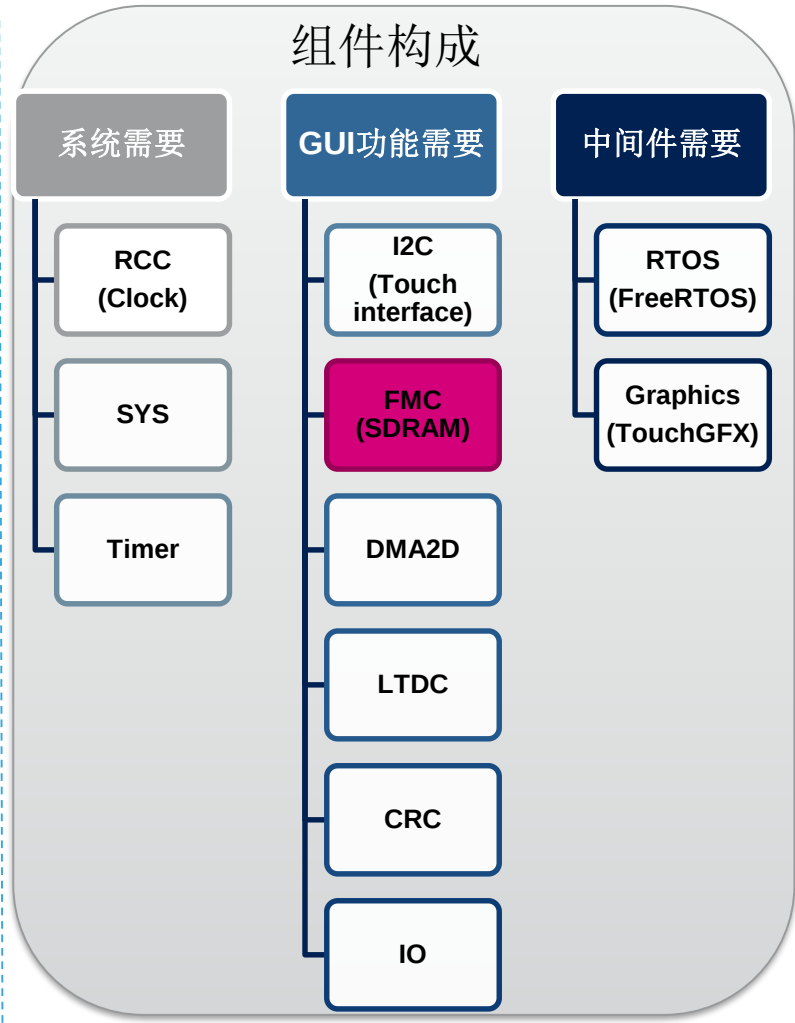
SDRAM Bank 0

4xByte enable



类别	MT48LC4M32B2B5-6A支持	硬件采用
SDRAM	-	SDCKE0 SDNE0
Bank	4	4
行地址 (Bit)	12	12
数据线	32	32
字节使能	32bit (4字节)	32bit (4字节)

组件构成





• 组件（GUI功能需要）

• FMC（外部存储器）

• 配置参数获取

在进行FMC的配置前，需要先了解与FMC关联的SDRAM硬件连接情况。

- 1. 根据SDRAM硬件支持和使用情况，配置SDRAM1
 - MT48LC4M32B2B5-6A支持32位数据线，硬件上采用了16位数据线
 - MT48LC4M32B2B5-6A内部含有4个Bank，硬件上采用了4个Bank
 - ...

类别	MT48LC4M32B2B5-6A支持	硬件采用
SDRAM		SDCKE0+SDNE0
Bank	4	4
行地址 (Bit)	12	12
数据线	32	16
字节使能	32bit (4字节)	16bit (2字节)

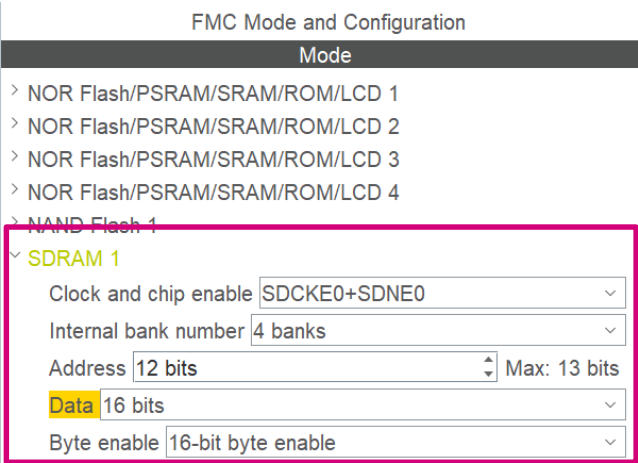
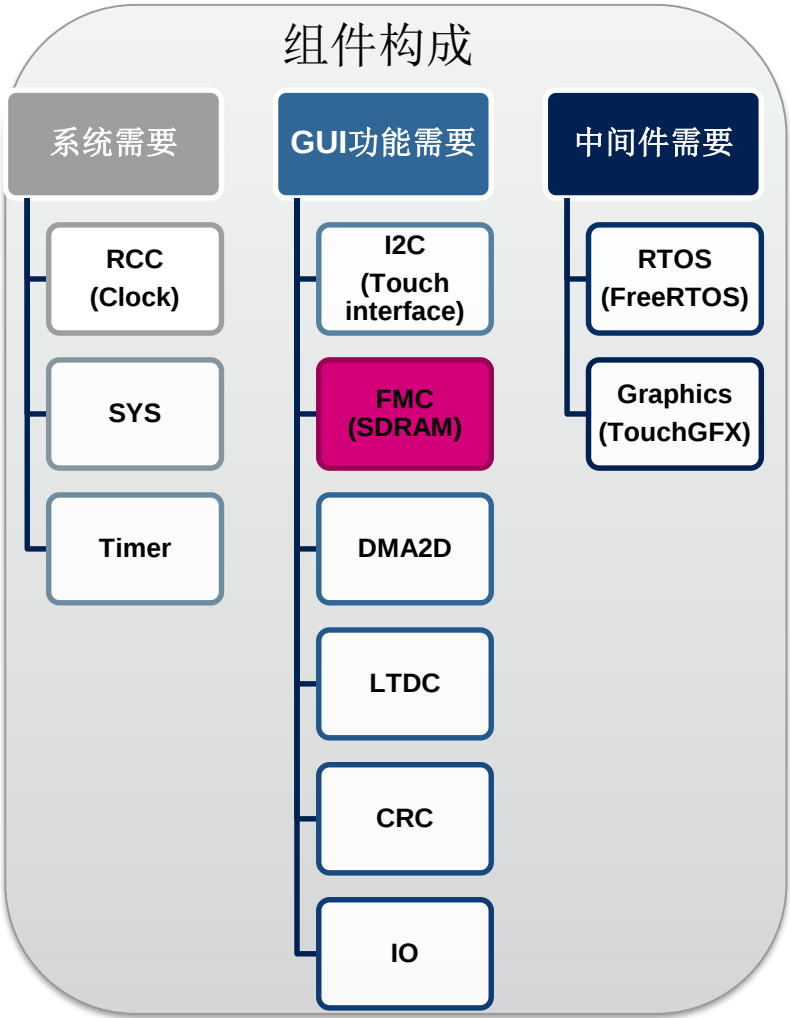


图. FMC SDRAM配置 (@STM32CubeMX)





• 组件（GUI功能需要）

• FMC（外部存储器）

• 配置参数获取（接上页）

- 2. 根据硬件情况，确认/调整接口的I/O引脚到硬件对应的复用引脚。

例如，STM32F746NGH6 有三个I/O能复用为 FMC_SDCKE0，分别是PH2, PC3, PC5。硬件电路上，PC3与存储器的CKE连接。

确认FMC_SDCKE0是否被分配到PC3引脚，如果不是，可按照如下步骤更改：

1. 查询SDCKE0
2. 找到FMC_SDCKE0对应的管脚
3. Ctrl + 左键按下，出现可复用管脚
4. 拖动到符合硬件连接的PC3

根据硬件连接，确认/调整复用引脚

注：实测自动分配的引脚与STM32F746-DISCO板采用引脚一致，可以直接跳过这个步骤

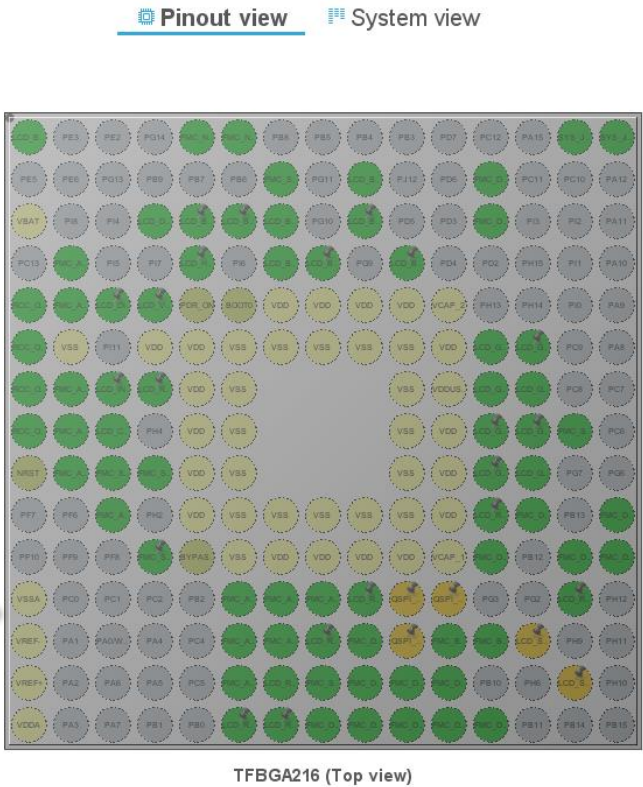
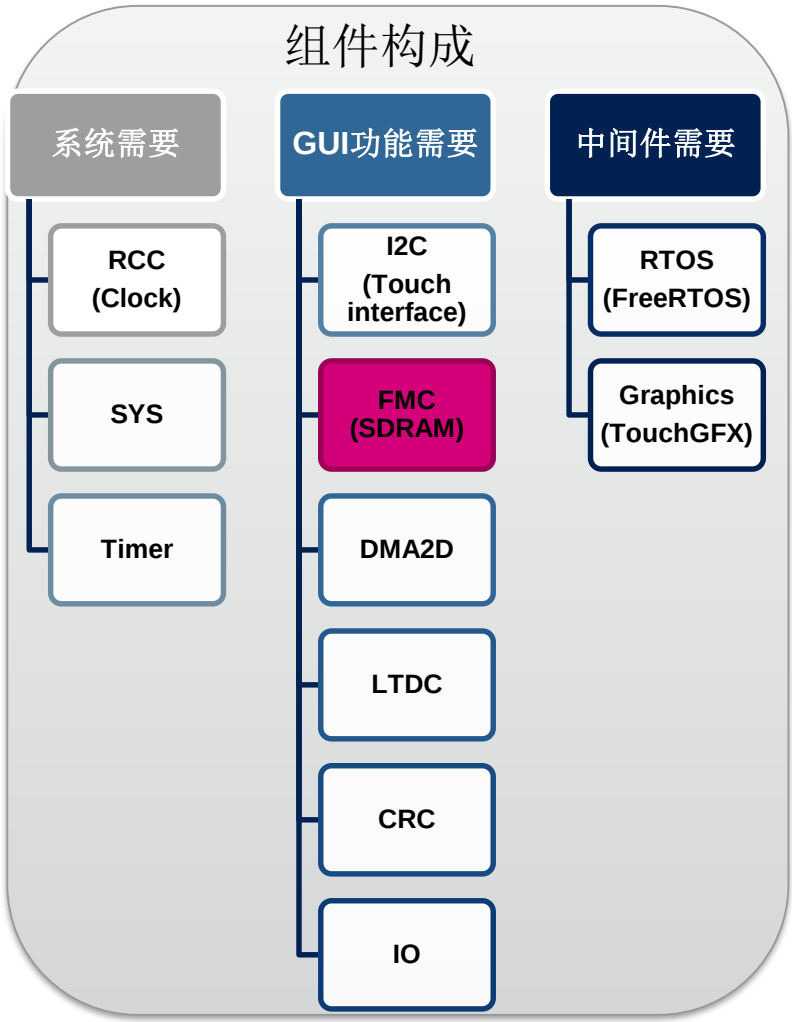


图. 引脚分布
（@STM32CubeMX）





• 组件（GUI功能需要）

• FMC（外部存储器）

• 配置参数获取（接上页）

- 3. 进入SDRAM1标签页
- 4. 根据**SDRAM数据手册**中参数，配置SDRAM控制参数及时序参数

根据SDRAM数据手册中参数及要求

根据应用需要，确定FMC时钟源频率

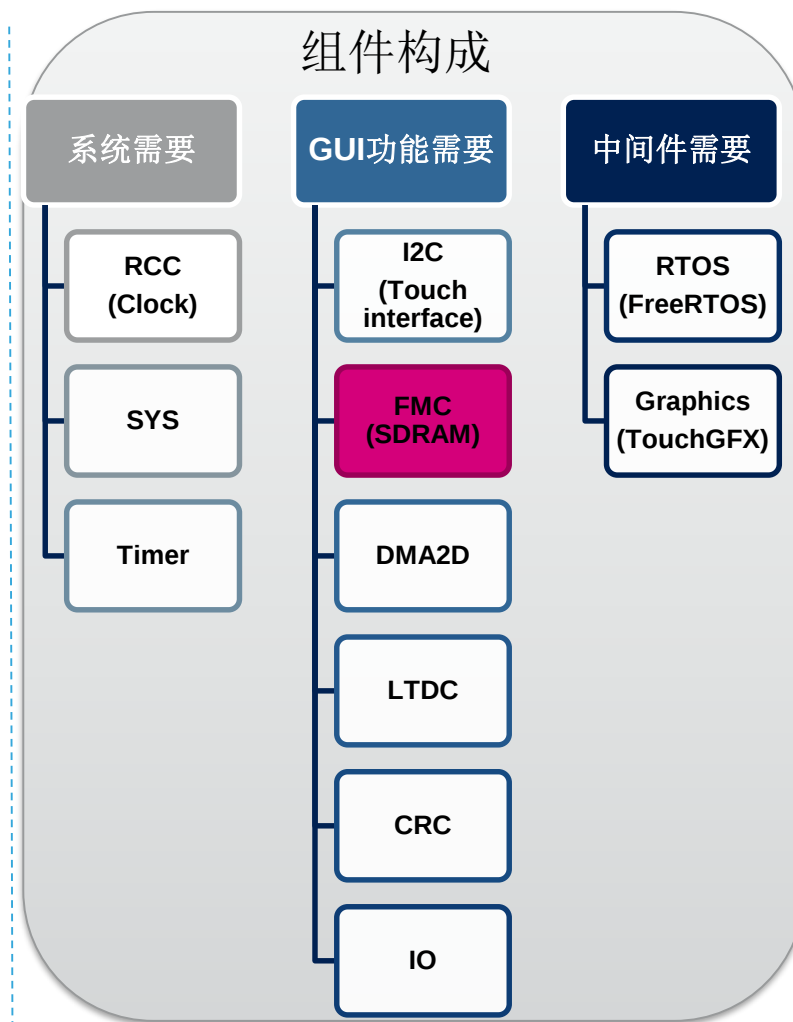


FMC配置参数

在STM32CubeMX中的FMC参数

实现示例（续）

20





• 组件（GUI功能需要）

• FMC（外部存储器）

• 配置参数获取（接上页）

- 3. 进入SDRAM1标签页
- 4. 根据SDRAM数据手册中参数，配置SDRAM控制参数及时序参数

首先，根据存储器数据手册，获取相关的电气参数。如下所示。

（SDRAM型号： MT48LC4M32B2B5-6A ）

Parameter	4 Meg x 32
Configuration	1 Meg x 32 x 4 banks
Refresh count	4K
Row addressing	4K A[11:0]
Bank addressing	4 BA[1:0]
Column addressing	256 A[7:0]

• Number of row address bits = 12

• Number of column address bits = 8

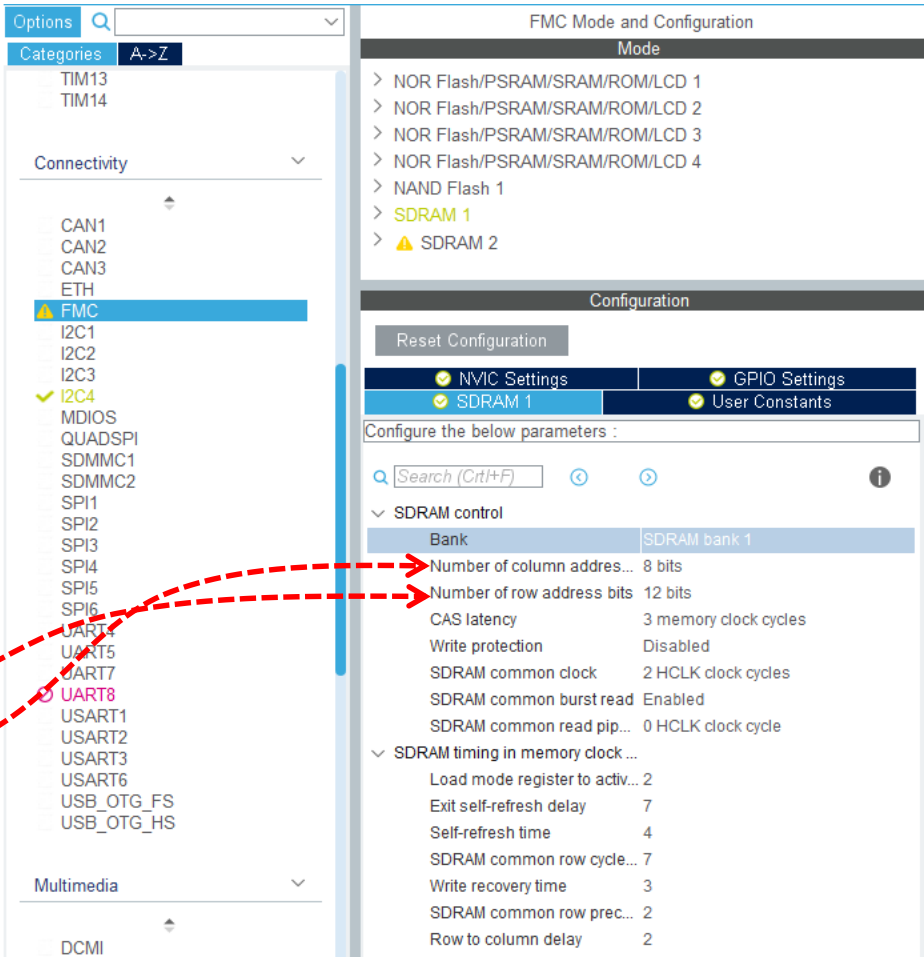


图. FMC SDRAM配置
（@STM32CubeMX）



实现示例（续）

22



• 组件（GUI功能需要）

• FMC（外部存储器）

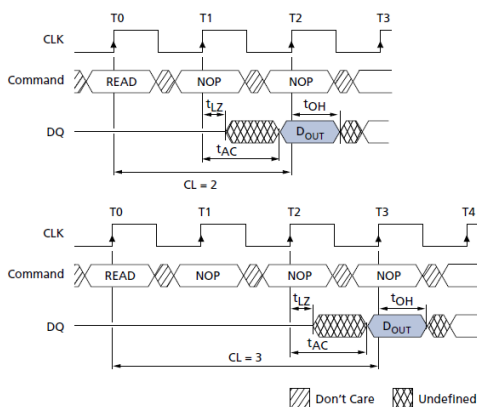
• 配置参数获取（接上页）

• 3. 进入SDRAM1标签页

• 4. 根据SDRAM数据手册中参数，配置SDRAM控制参数及时序参数

首先，根据存储器数据手册，获取相关的电气参数。如下所示。

（SDRAM型号： **MT48LC4M32B2B5-6A** ）



• 根据存储器描述，在CAS latency后，直接可以访问数据。所以SDRAM common read pipe delay(CAS latency后至访问数据的延时周期)为0。

• **MT48LC4M32B2B5-6A**支持Burst read

• **MT48LC4M32B2B5-6A**不支持写保护

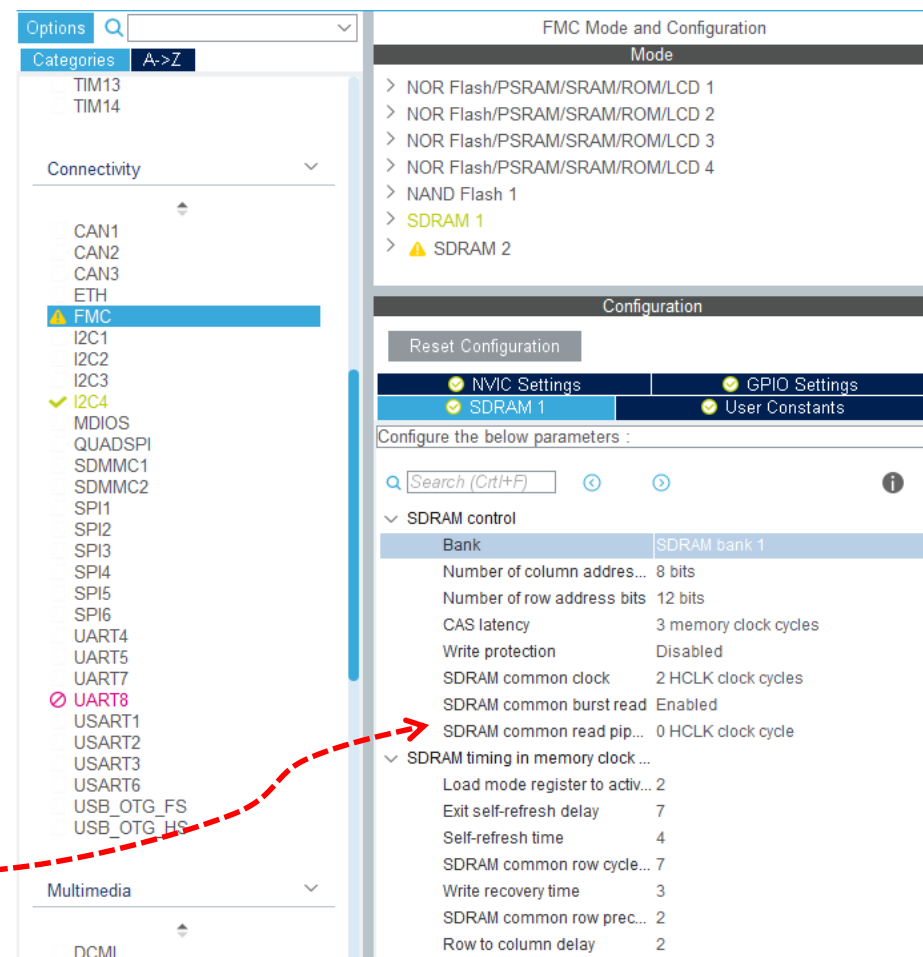


图. FMC SDRAM配置
（@STM32CubeMX）



• 组件（GUI功能需要）

• FMC（外部存储器）

• 配置参数获取（接上页）

- 3. 进入SDRAM1标签页
- 4. 根据SDRAM数据手册中参数，配置SDRAM控制参数及时序参数

首先，根据存储器数据手册，获取相关的电气参数。如下所示。

（SDRAM型号： MT48LC4M32B2B5-6A ）

CL = CAS (READ) latency

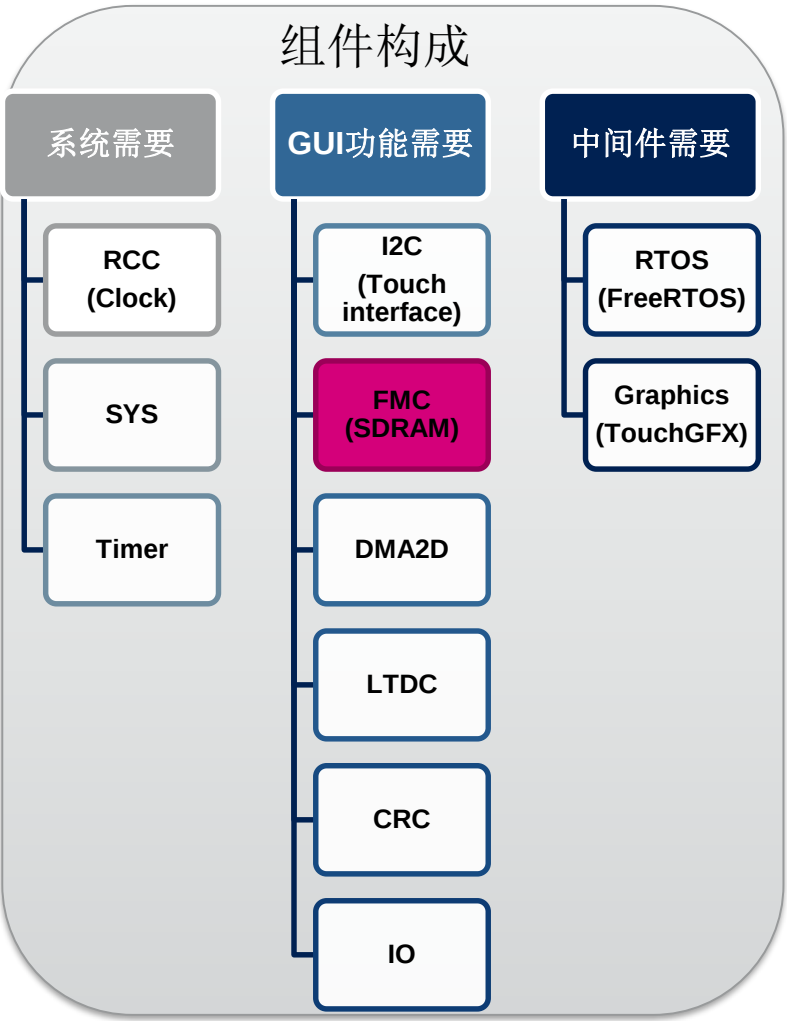
Speed Grade	Clock Frequency (MHz)	Target 'RCD-'RP-CL	'RCD (ns)	'RP (ns)	CL (ns)
-6A	167	3-3-3	18	18	18
-6	167	3-3-3	18	18	18
-7	143	3-3-3	20	20	21

• CAS (READ) latency = 18 ns

Parameter	Symbol	-6A ¹		-6		-7		Unit	Notes
		Min	Max	Min	Max	Min	Max		
Clock cycle time	CL = 3	¹ CK(3)	6	–	6	–	7	ns	8
	CL = 2	¹ CK(2)	10	–	10 ²	–	10	ns	8
	CL = 1	¹ CK(1)	20	–	20 ²	–	20	ns	8
ACTIVE-to-PRECHARGE command	¹ RAS	42	120,000	42	120,000	42	120,000	ns	
ACTIVE-to-ACTIVE command period	¹ RC	60	–	60	–	70	–	ns	10
AUTO REFRESH period	¹ RFC	60	–	60	–	70	–	ns	
ACTIVE-to-READ or WRITE delay	¹ RCD	18	–	18	–	20	–	ns	
Refresh period (4096 rows)	¹ REF	–	64	–	64	–	64	ms	
Refresh period – automotive (4096 rows)	¹ REF _{AT}	–	16	–	16	–	16	ms	
PRECHARGE command period	¹ RP	18	–	18	–	20	–	ns	
ACTIVE bank a to ACTIVE bank b command	¹ RBD	12	–	12	–	15	–	ns	11
Transition time	¹ T	0.3	1.2	0.3	1.2	0.3	1.2	ns	3
WRITE recovery time	¹ WR	1	–	1	–	1	–	¹ CK	12
		CLK + 7ns	–	CLK + 6ns	–	CLK + 7ns	–		
		12	–	12	–	14	–	ns	13
Exit SELF REFRESH-to-ACTIVE command	¹ XSR	67	–	70	–	70	–	ns	14

- FMC SDCLK
- Self-refresh ≥ 42ns
- SDRAM common row cycle delay ≥ 60ns
- Auto refresh ≥ 60ns
- Row to column delay ≥ 18ns
- SDRAM common row precharge delay ≥ 18ns
- Write recovery time ≥ 1CLK + 7ns
- Exit self-refresh delay ≥ 67ns

组件构成





• 组件（GUI功能需要）

• FMC（外部存储器）

• 配置参数获取（接上页）

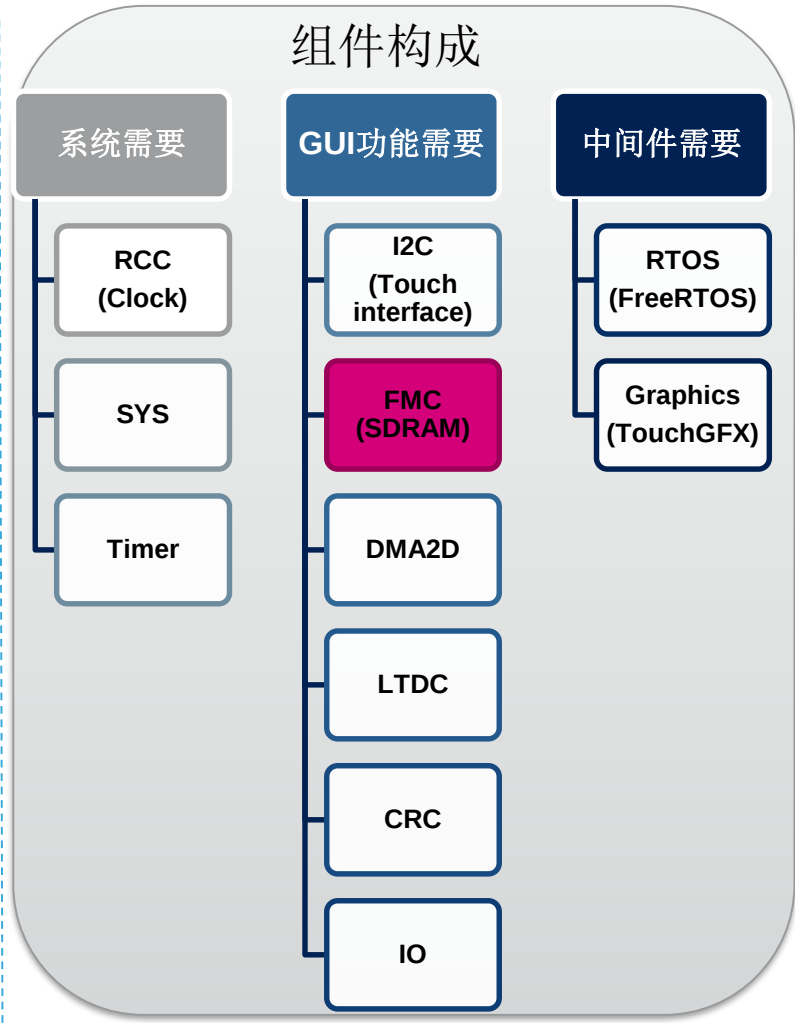
- 3. 进入SDRAM1标签页
- 4. 根据SDRAM数据手册中参数，配置SDRAM控制参数及时序参数

首先，根据存储器数据手册，获取相关的电气参数。如下所示。

（SDRAM型号：MT48LC4M32B2B5-6A）

Parameter		Symbol	-6	-6A	-7	Unit	Notes
READ/WRITE command to READ/WRITE command		t _{CCD}	1	1	1	t _{CK}	
CKE to clock disable or power-down entry mode		t _{CKED}	1	1	1	t _{CK}	15
CKE to clock enable or power-down exit setup mode		t _{PED}	1	1	1	t _{CK}	
DQM to input data delay		t _{DQD}	0	0	0	t _{CK}	
DQM to data mask during WRITES		t _{DQM}	0	0	0	t _{CK}	
DQM to data High-Z during READs		t _{DQZ}	2	2	2	t _{CK}	
WRITE command to input data delay		t _{DWD}	0	0	0	t _{CK}	
Data-in to ACTIVE command	CL = 3	t _{DAL} (3)	5	4	5	t _{CK}	16
	CL = 2	t _{DAL} (2)	4	4	4	t _{CK}	16
	CL = 1	t _{DAL} (1)	3	3	3	t _{CK}	16
Data-in to PRECHARGE command		t _{DPL}	3	3	3	t _{CK}	17
Last data-in to burst STOP command		t _{BDL}	1	1	1	t _{CK}	
Last data-in to new READ/WRITE command		t _{CDL}	1	1	1	t _{CK}	
Last data-in to burst PRECHARGE command		t _{RDL}	2	2	2	t _{CK}	17
LOAD MODE REGISTER command to ACTIVE or REFRESH command		t _{MRD}	2	2	2	t _{CK}	
Data-out to High-Z from PRECHARGE command	CL = 3	t _{ROH} (3)	3	3	3	t _{CK}	
	CL = 2	t _{ROH} (2)	2	2	2	t _{CK}	
	CL = 1	t _{ROH} (1)	1	1	1	t _{CK}	

• Load mode register to active delay = 2 tCK





• 组件（GUI功能需要）

• FMC（外部存储器）

• 配置参数获取（接上页）

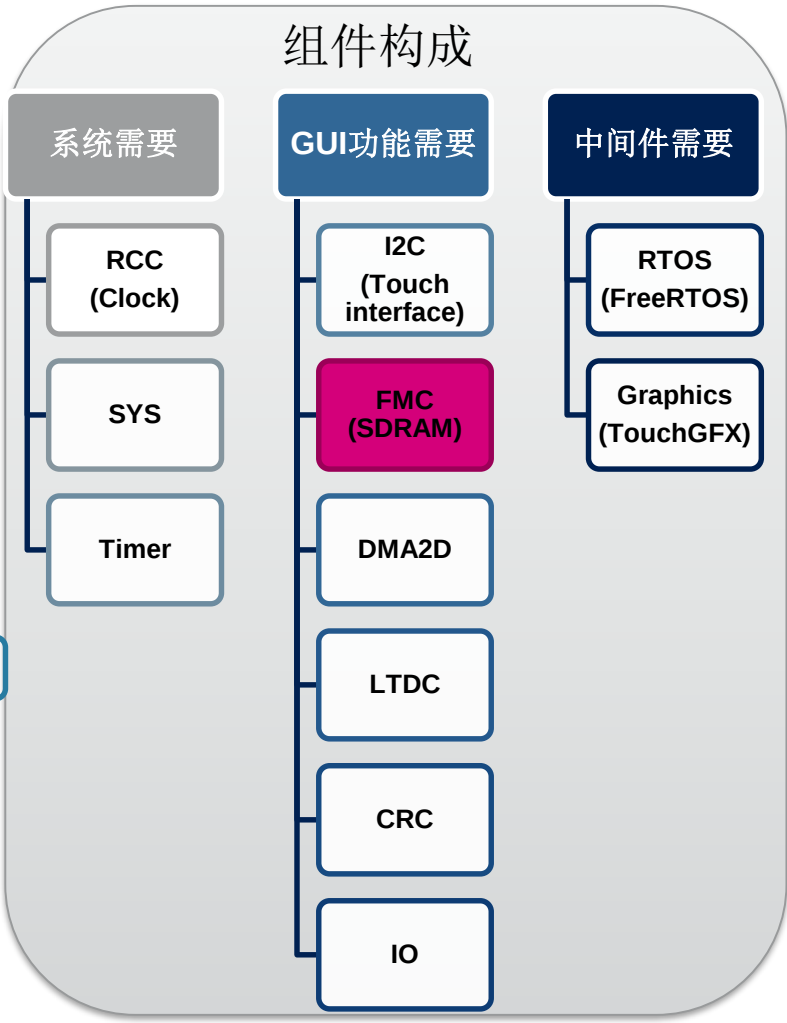
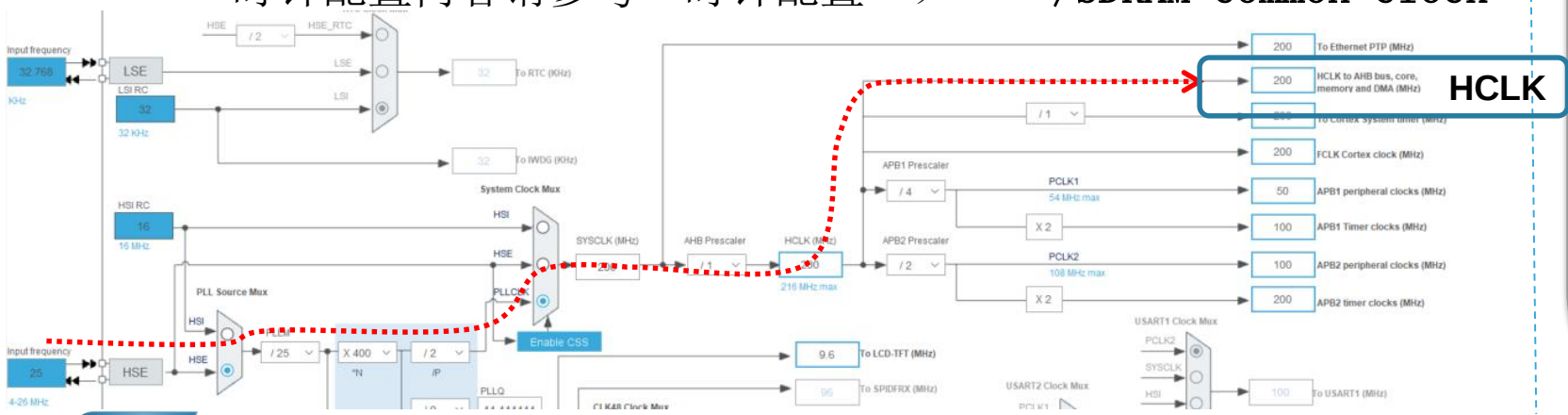
- 3. 进入SDRAM1标签页
- 4. 根据SDRAM数据手册，获取SDRAM的时间参数，配置SDRAM控制参数及时序参数

获取/设置主频。演示例中，采用最高主频。**FMC**时钟源频率路径如下。（更多时钟配置内容请参考“时钟配置”）

$$FMC_SDCLK = HCLK / SDRAM\ common\ clock$$



/SDRAM common clock



注：SDRAM common clock为STM32CubeMX中，SDRAM（FMC）配置项。具有SDRAM分频作用。



实现示例（续）

26



• 组件（GUI功能需要）

• FMC（外部存储器）

• 配置参数获取（接上页）

- 3. 进入SDRAM1标签页
- 4. 根据SDRAM数据手册中参数，配置SDRAM控制参数及时序参数
- MT48LC4M32B2B5-6A支持CAS latency (CL) : 1, 2, 3
- CAS (READ) latency = 18 ns



$$HCLK = 200MHz$$
$$FMC_SDCLK = \frac{HCLK}{SDRAM \text{ common clock}}$$

$$\frac{CL}{FMC_SDCLK} \geq 18ns$$

SDRAM common clock 范围： 2, 3

- 为了实现更高的SDRAM访问频率，在能满足参数的前提下，SDRAM common clock选择2



- $CL = 2$ 时, $\frac{CL}{FMC_SDCLK} \approx 20ns$; $CL = 3$ 时, $\frac{CL}{FMC_SDCLK} \approx 30ns$

- 考虑到冗余，选择CL为3

注：SDRAM common clock为STM32CubeMX中，SDRAM（FMC）配置项。具有SDRAM分频作用。

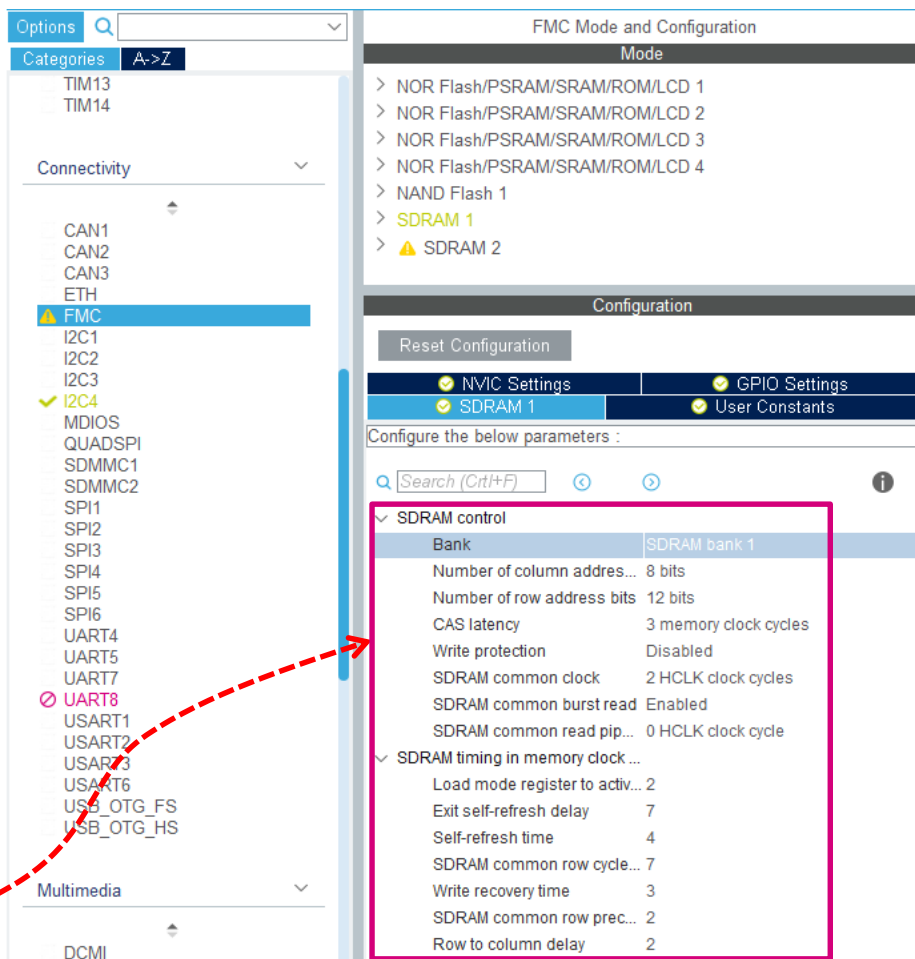


图. FMC SDRAM配置
（@STM32CubeMX）



• 组件（GUI功能需要）

• FMC（外部存储器）

• 配置参数获取（接上页）

- 3. 进入SDRAM1标签页
- 4. 根据SDRAM数据手册中参数，配置SDRAM控制参数及时序参数

参数	最小值
Load mode register to active delay	2 CLK
Exit self-refresh delay	67ns
Self-refresh	42ns
SDRAM common row cycle delay	60ns
Write recovery time	1 CLK + 7ns
SDRAM common row precharge delay	18ns
Row to column delay	18ns



参数	最小值 (FMC SDCLK)
Load mode register to active delay	2
Exit self-refresh delay	7
Self-refresh	5
SDRAM common row cycle delay	7
Write recovery time	3
SDRAM common row precharge delay	2
Row to column delay	2

考虑到如下要求（STM32CubeMX中提示）：

1: WriteRecoveryTime >= SelfRefreshTime - RowToColumnDelay;

2: WriteRecoveryTime >= RowCycleDelay - RowToColumnDelay - RowPrechargeDelay.

考虑上面的限制，将设置其为3

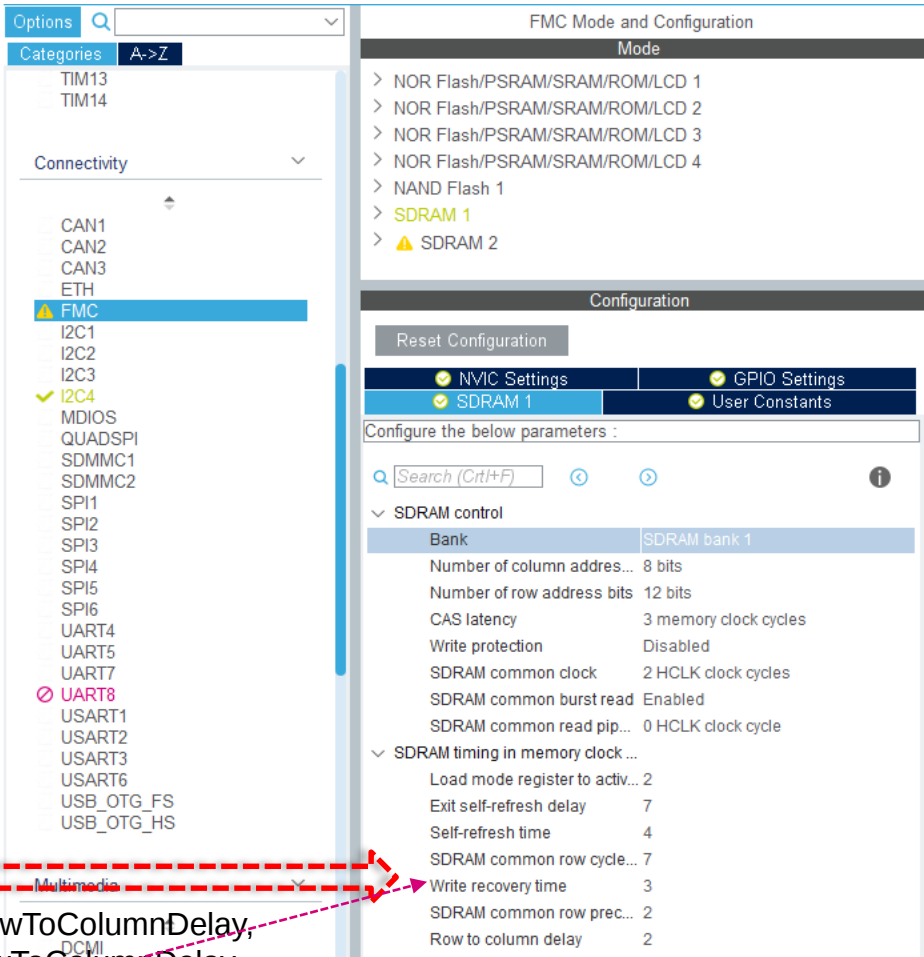


图. FMC SDRAM配置 (@STM32CubeMX)

• 在之前，确定了FMC_SDCLK = 200/2 MHz = 100MHz



注：STM32CubeMX生成初始化中，未对Self-refresh进行设置，演示例中Self-refresh参数无实际意义。



实现示例（续）

28



- 组件（GUI功能需要）
 - FMC（GPIO Settings，连接关系按照原理图配置）

SDRAM 1 User Constants NVIC Settings GPIO Settings							
Search Signals							
Search (Ctrl+F)							
☐ Show only Modified Pins							
Pin Name	Signal on Pin	GPIO output...	GPIO mode	GPIO Pull-u...	Maximum o...	User Label	Modified
PC3	FMC_SDCK...	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_SDCK...	✓
PD0	FMC_D2	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_D2 [M...	✓
PD1	FMC_D3	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_D3 [M...	✓
PD8	FMC_D13	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_D13 [...]	✓
PD9	FMC_D14	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_D14 [...]	✓
PD10	FMC_D15	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_D15 [...]	✓
PD14	FMC_D0	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_D0 [M...	✓
PD15	FMC_D1	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_D1 [M...	✓
PE0	FMC_NBL0	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_NBL0 ...	✓
PE1	FMC_NBL1	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_NBL1 ...	✓
PE7	FMC_D4	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_D4 [M...	✓
PE8	FMC_D5	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_D5 [M...	✓
PE9	FMC_D6	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_D6 [M...	✓
PE10	FMC_D7	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_D7 [M...	✓
PE11	FMC_D8	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_D8 [M...	✓
PE12	FMC_D9	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_D9 [M...	✓
PE13	FMC_D10	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_D10 [...]	✓
PE14	FMC_D11	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_D11 [...]	✓
PE15	FMC_D12	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_D12 [...]	✓
PF0	FMC_A0	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_A0 [M...	✓

SDRAM 1 User Constants NVIC Settings GPIO Settings							
Search Signals							
Search (Ctrl+F)							
☐ Show only Modified Pins							
Pin Name	Signal on Pin	GPIO output...	GPIO mode	GPIO Pull-u...	Maximum o...	User Label	Modified
PF0	FMC_A0	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_A0 [M...	✓
PF1	FMC_A1	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_A1 [M...	✓
PF2	FMC_A2	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_A2 [M...	✓
PF3	FMC_A3	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_A3 [M...	✓
PF4	FMC_A4	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_A4 [M...	✓
PF5	FMC_A5	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_A5 [M...	✓
PF11	FMC_SDNR...	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_SDNR...	✓
PF12	FMC_A6	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_A6 [M...	✓
PF13	FMC_A7	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_A7 [M...	✓
PF14	FMC_A8	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_A8 [M...	✓
PF15	FMC_A9	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_A9 [M...	✓
PG0	FMC_A10	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_A10 [...]	✓
PG1	FMC_A11	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_A11 [...]	✓
PG4	FMC_BA0	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_BA0 [...]	✓
PG5	FMC_BA1	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_BA1 [...]	✓
PG8	FMC_SDCLK	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_SDCL...	✓
PG15	FMC_SDNC...	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_SDNC...	✓
PH3	FMC_SDNE0	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_SDNE...	✓
PH5	FMC_SDNWE	n/a	Alternate Fu...	No pull-up a...	Very High	FMC_SDN...	✓



实现示例（续）

29



• 组件（GUI功能需要）

• DMA2D

- 2D图形加速器

• DMA2D在STM32系统架构中位置及作用

- 系统总线主从设备间数据传递/填充

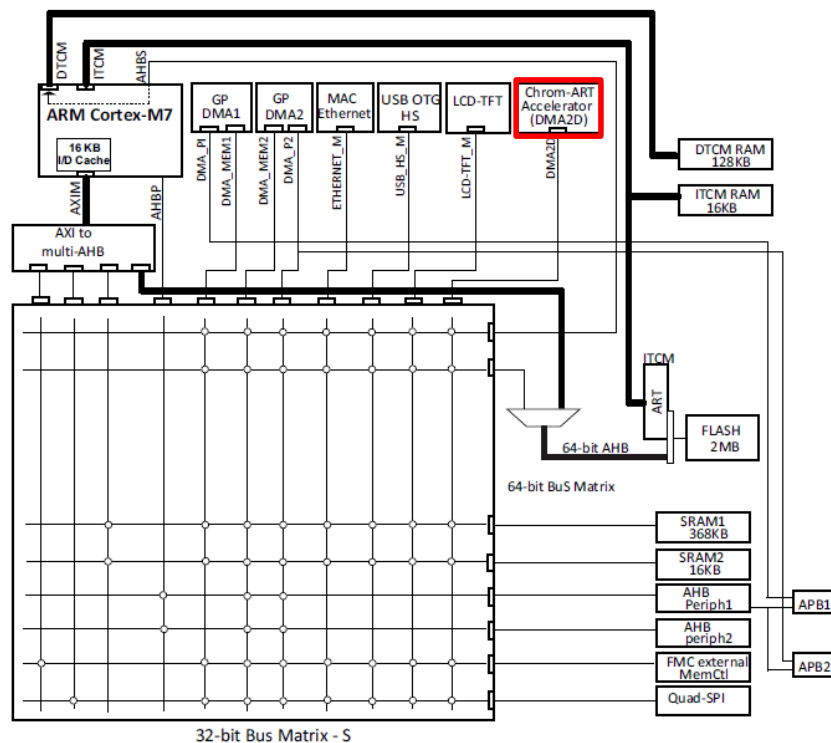
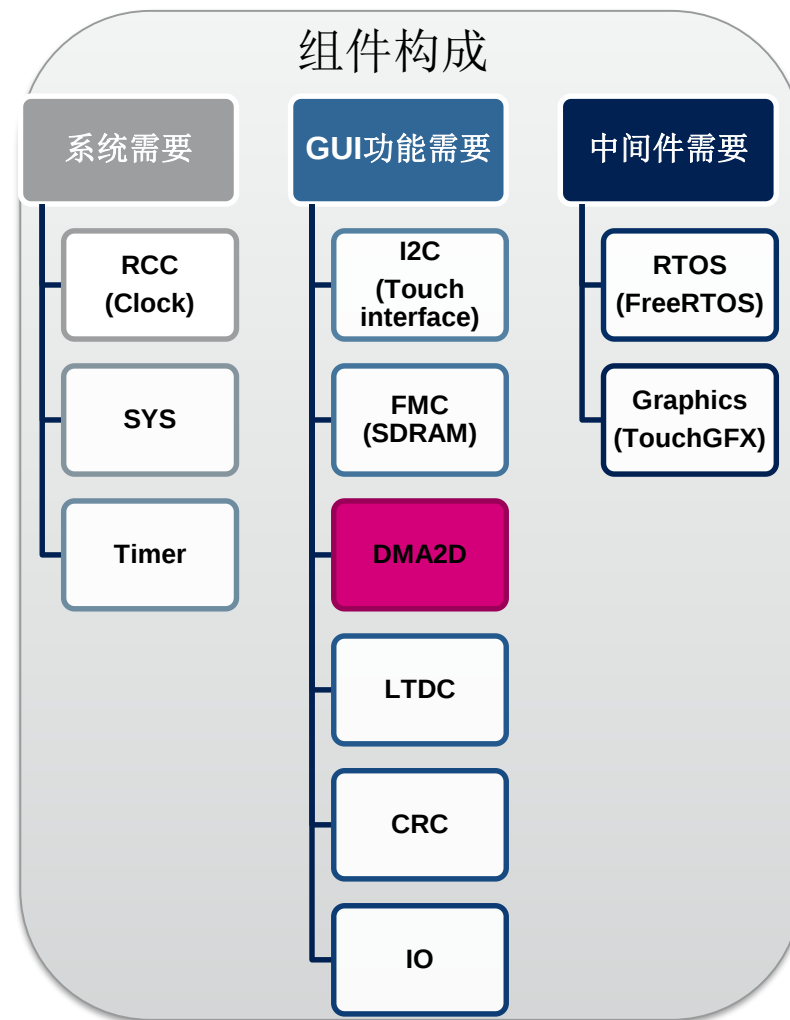


图. 系统结构图摘自参考手册RM0410

组件构成

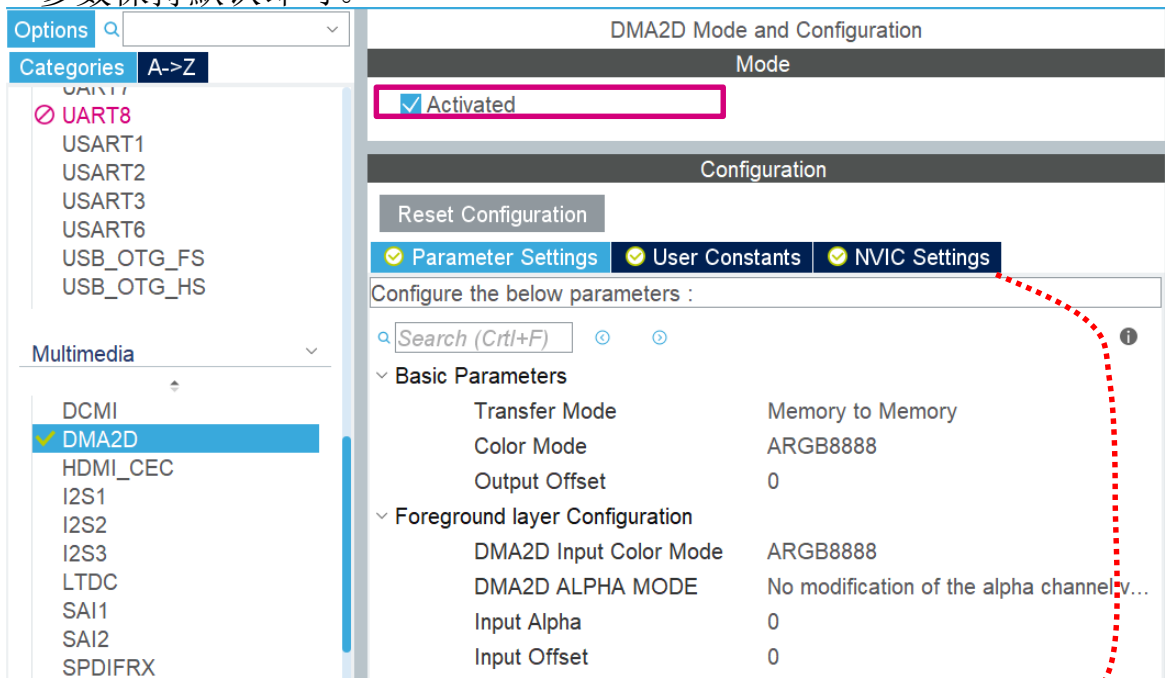




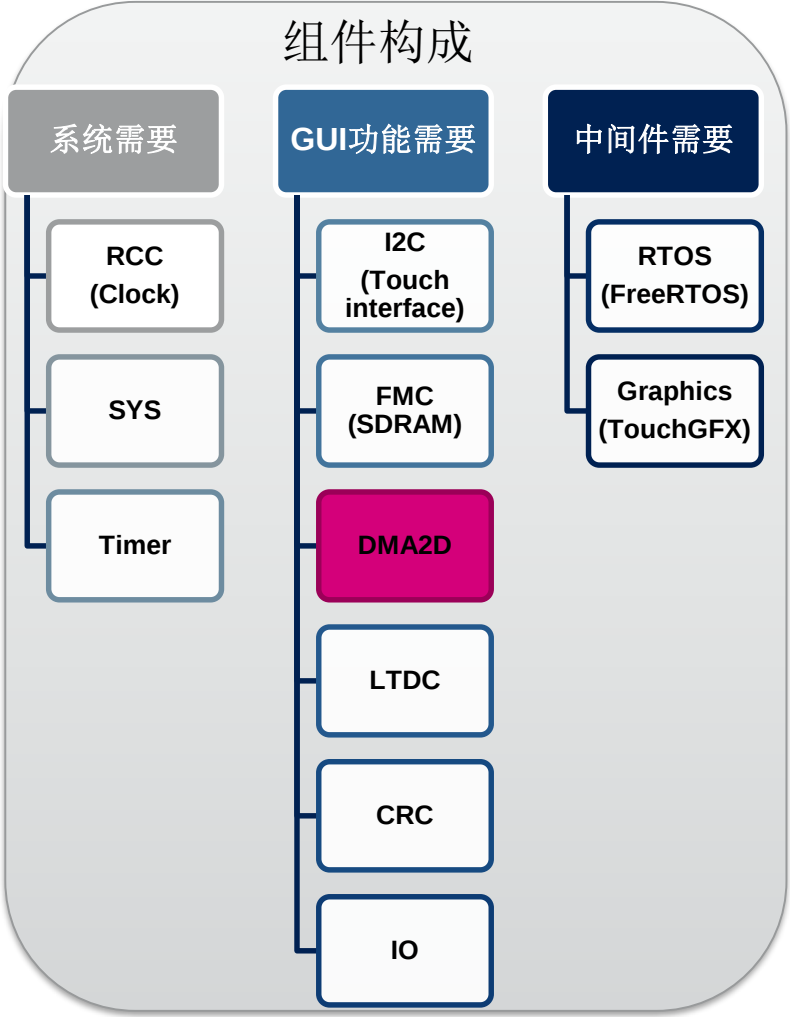
• 组件（GUI功能需要）

• DMA2D

- 2D图形加速器
- 参数保持默认即可。



Parameter Settings	User Constants	NVIC Settings
NVIC Interrupt Table		
	Enabled	Preemption Priority
DMA2D global interrupt	☒	0





实现示例（续）

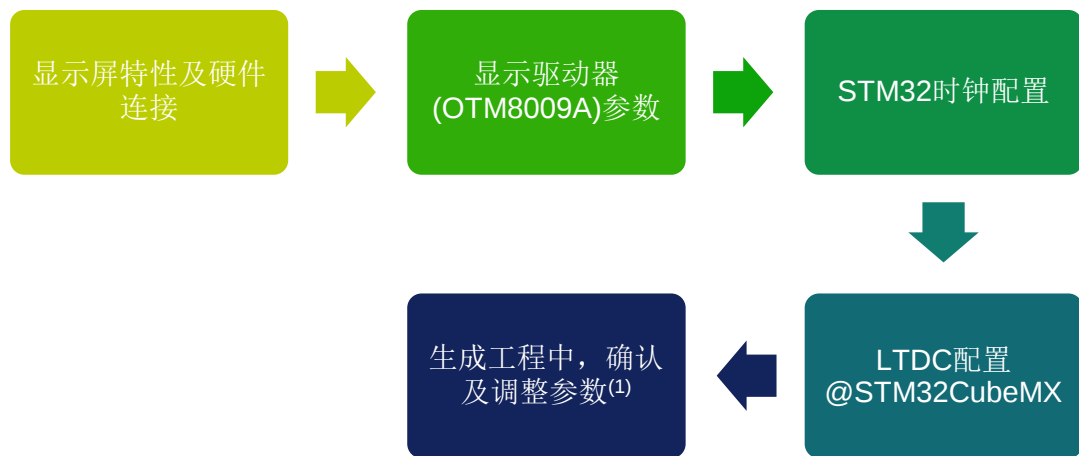
31



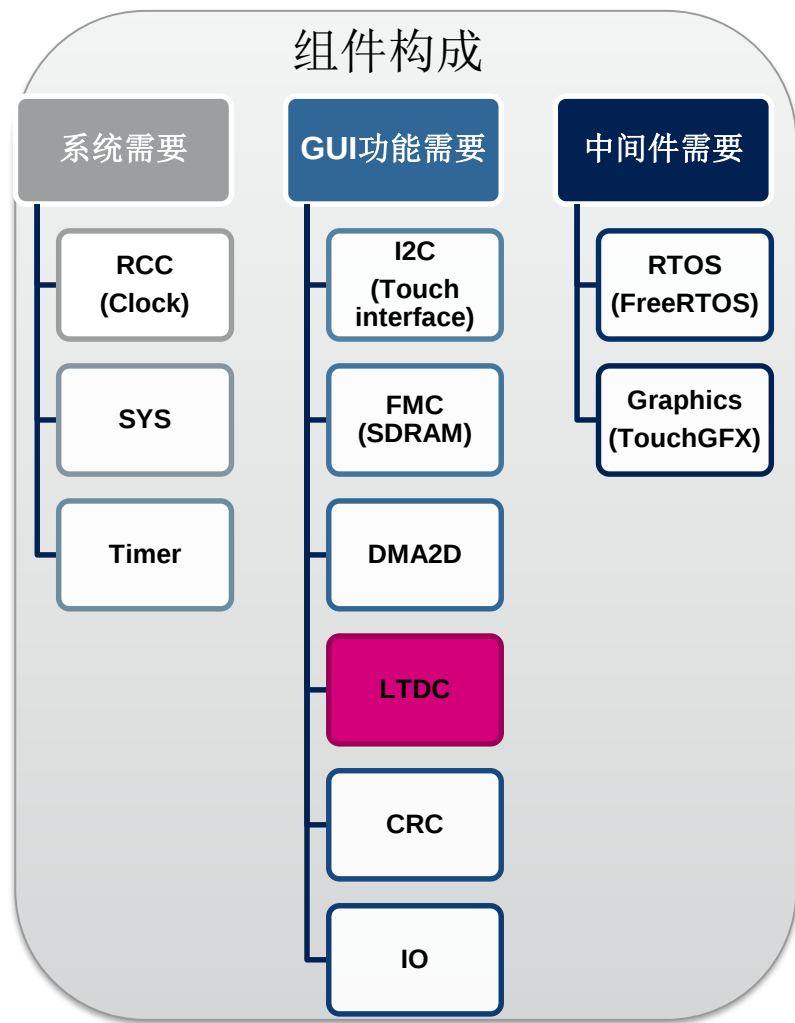
• 组件（GUI功能需要）

• LTDC（LCD-TFT Display Controller）

- 不同的显示驱动器，对应的LTDC配置参数不同。
- 首先，针对当前演示例，呈现在STM32CubeMX上LTDC配置。
- 然后，以STM32F746G-Disco板上的显示屏RK043FN48H-CT672B为例，介绍如何根据实际情况，获取配置参数。



LTDC配置流程图（以演示例中采用的RK043FN48H-CT672B为例）





- 组件（GUI功能需要）
 - LTDC配置（STM32CubeMX中）
 - Parameter settings设置

LTDC Mode and Configuration

Mode

Display Type RGB888 (24 bits)

Configuration

Reset Configuration

Parameter Settings Layer Settings User Constants NVIC Settings GPIO Settings

Configure the below parameters :

Search (Ctrl+F)

Synchronization for Width

Horizontal Synchronization Width RK043FN48H_HSYNC pixels

Horizontal Back Porch RK043FN48H_HBP pixels

Active Width 480 pixels

Horizontal Front Porch RK043FN48H_HFP pixels

HSync Width 40

Accumulated Horizontal Back Porch Width 53

Accumulated Active Width 533

Total Width 565

Synchronization for Height

Vertical Synchronization Height RK043FN48H_VSYNC lines

Vertical Back Porch RK043FN48H_VBP lines

Active Height 272 lines

Vertical Front Porch RK043FN48H_VFP lines

VSyn Height 9

Accumulated Vertical Back Porch Height 11

Accumulated Active Height 283

Total Height 285

Signal Polarity

Horizontal Synchronization Polarity Active Low

Vertical Synchronization Polarity Active Low

Not Data Enable Polarity Active Low

Pixel Clock Polarity Normal Input

BackGround Color

Red 0

Green 0

Blue 0

LTDC Mode and Configuration

Mode

Display Type RGB888 (24 bits)

Configuration

Reset Configuration

Parameter Settings Layer Settings User Constants NVIC Settings GPIO Settings

Search Constants

Search (Ctrl+F)

Constant Name	
RK043FN48H_HSYNC	41
RK043FN48H_VSYNC	10
RK043FN48H_HBP	13
RK043FN48H_VBP	2
RK043FN48H_WIDTH	480
RK043FN48H_HEIGHT	272
RK043FN48H_HFP	32
RK043FN48H_VFP	2

图. LTDC配置（@STM32CubeMX）



• 组件（GUI功能需要）

• LTDC配置（STM32CubeMX中）

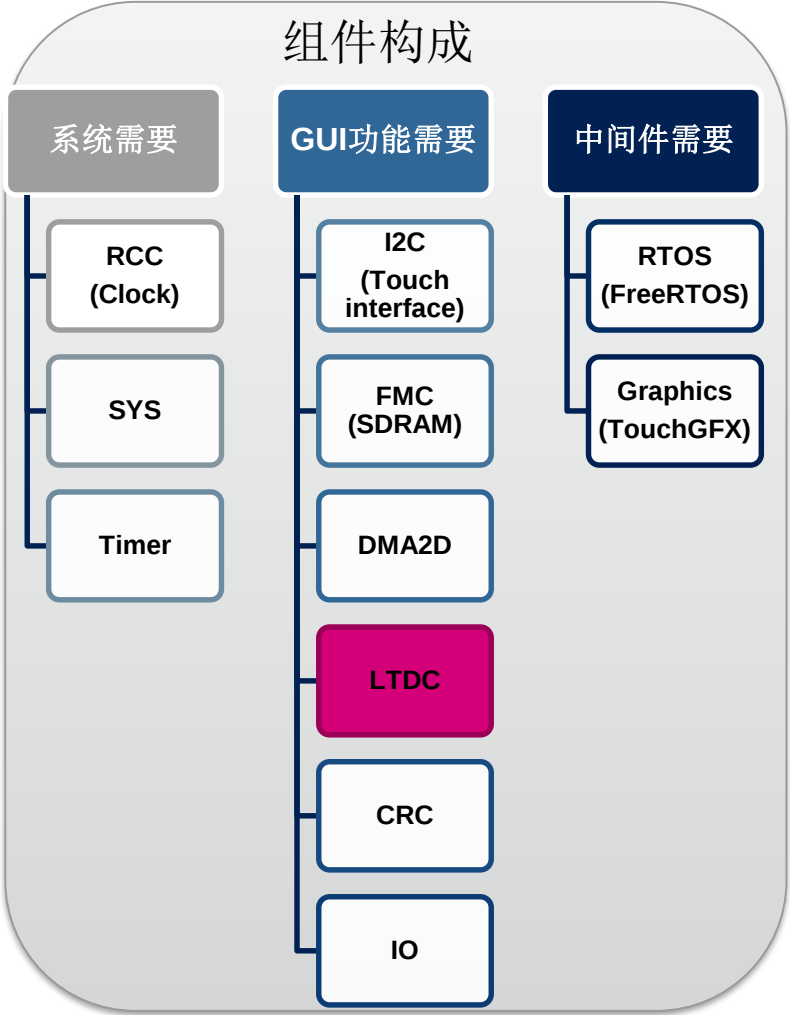
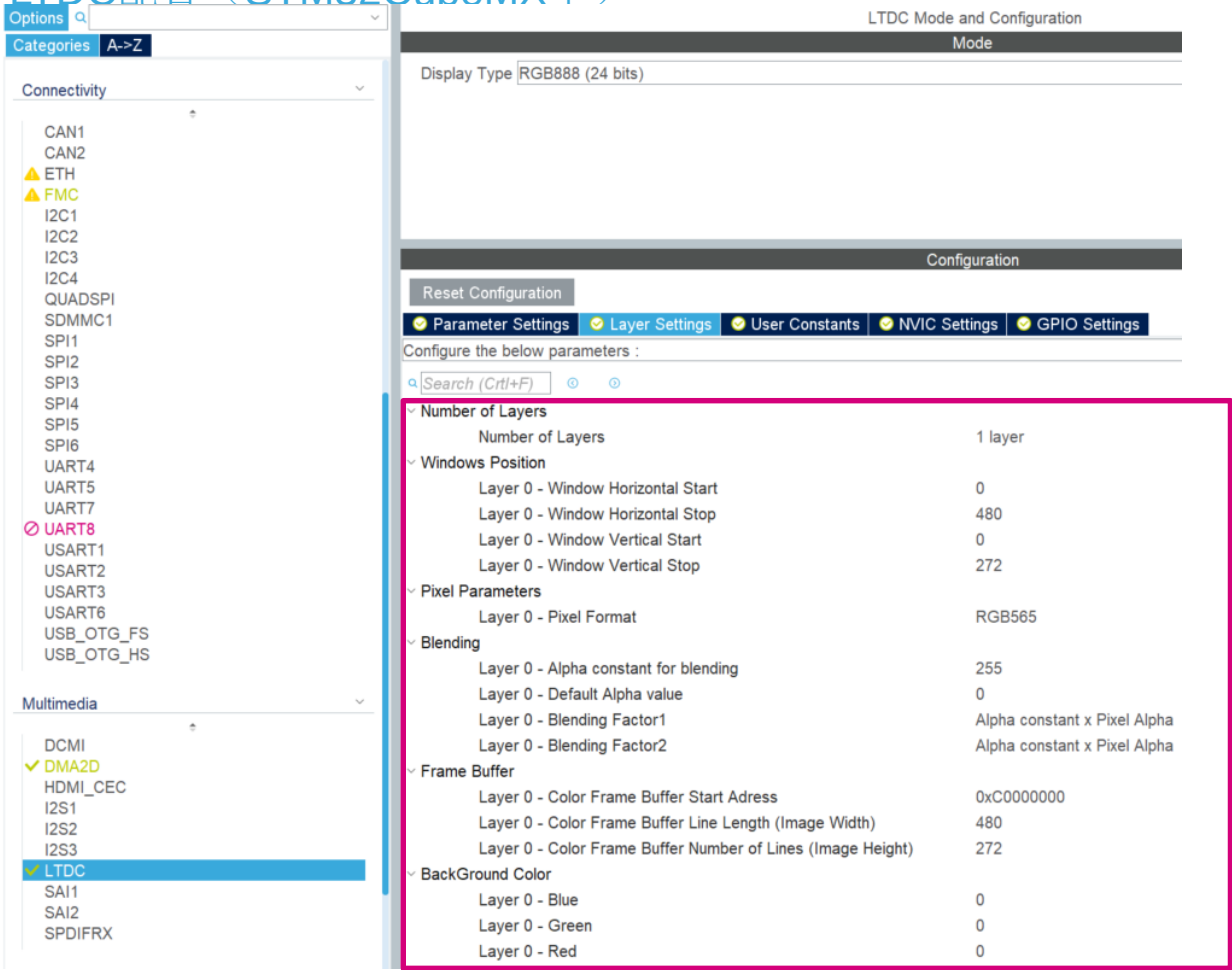


图. LTDC配置（@STM32CubeMX）



• 组件（GUI功能需要）

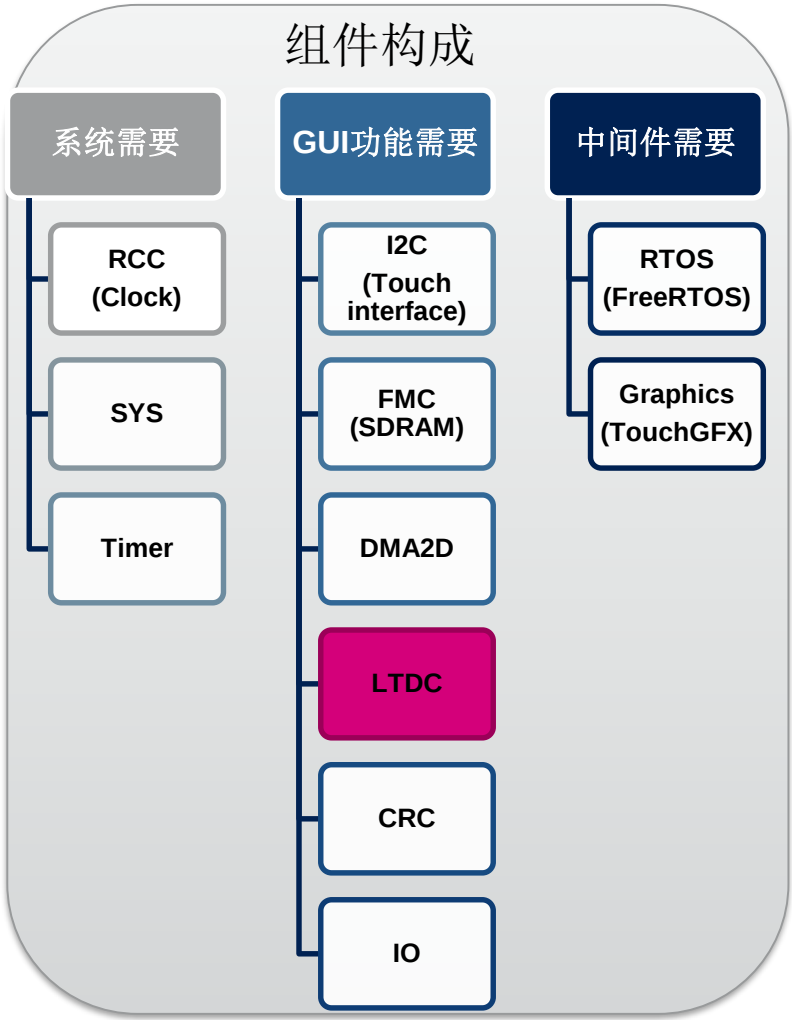
• LTDC（LCD-TFT Display Controller）

- Parameter settings设置
- 配置参数获取
 - 2. 根据LTDC（DSI模式）和显示屏模组规范书，完成Parameter Settings配置

• LTDC中同步信号时间参数和信号极性考虑因素，如下表。

屏类型	显示接口	模式	显示模组	
			同步信号时间参数	控制信号极性
RGB屏	LTDC		Y	Y
MIPI屏	DSI（内部关联LTDC）	Video	Y	N
		Command/Adapted Command	N	N

注： Y 表示需要根据显示模组中参数/特性进行匹配、配置。
N 不需要与显示模组中参数/特性匹配。但LTDC需保持与DSI中信号极性一致。

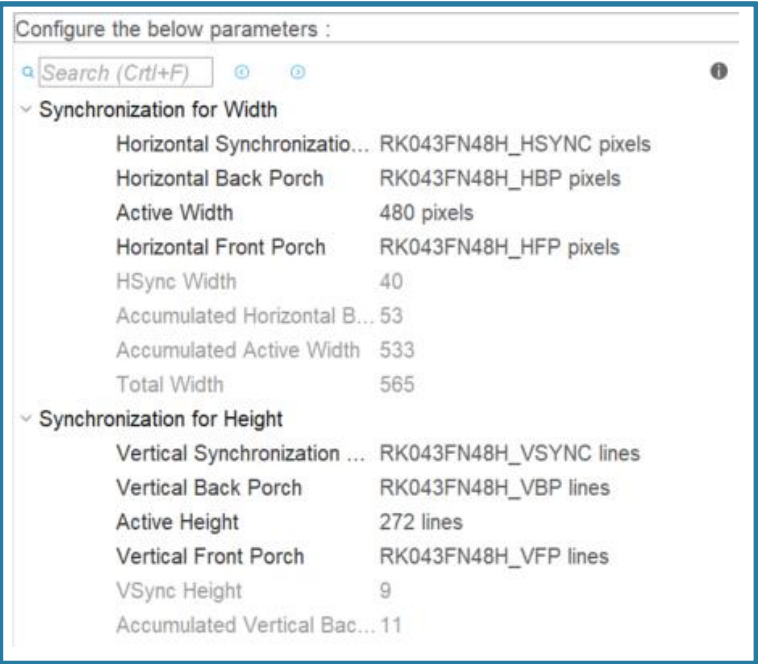




• 组件（GUI功能需要）

• LTDC（LCD-TFT Display Controller）

- Parameter settings设置
- 配置参数获取
 - 2. 根据LTDC和显示屏模组规范书，完成Parameter Settings配置
 - 演示例中采用272x480分辨率
 - 注：使用LTDC，需要根据显示屏模组规范书中电气参数，进行配置，如右表所示。



Item		Symbol	Min.	Typ.	Max.	Unit
DCLK Frequency		Fclk	5	9	12	MHz
DCLK Period		Tclk	83	110	200	ns
Hsync	Period Time	Th	490	531	605	DCLK
	Display Period	Thdisp		480		DCLK
	Back Porch	Thbp	8	43		DCLK
	Front Porch	Thfp	2	8		DCLK
	Pulse Width	Thw	1			DCLK
Vsync	Period Time	Tv	275	288	335	H
	Display Period	Tvdisp		272		H
	Back Porch	Tvbp	2	12		H
	Front Porch	Tvfp	1	4		H
	Pulse Width	Tvw	1	10		H

RK043FN48H 272x480显示时间参数表



实现示例（续）

36



• 组件（GUI功能需要）

• LTDC（LCD-TFT Display Controller）

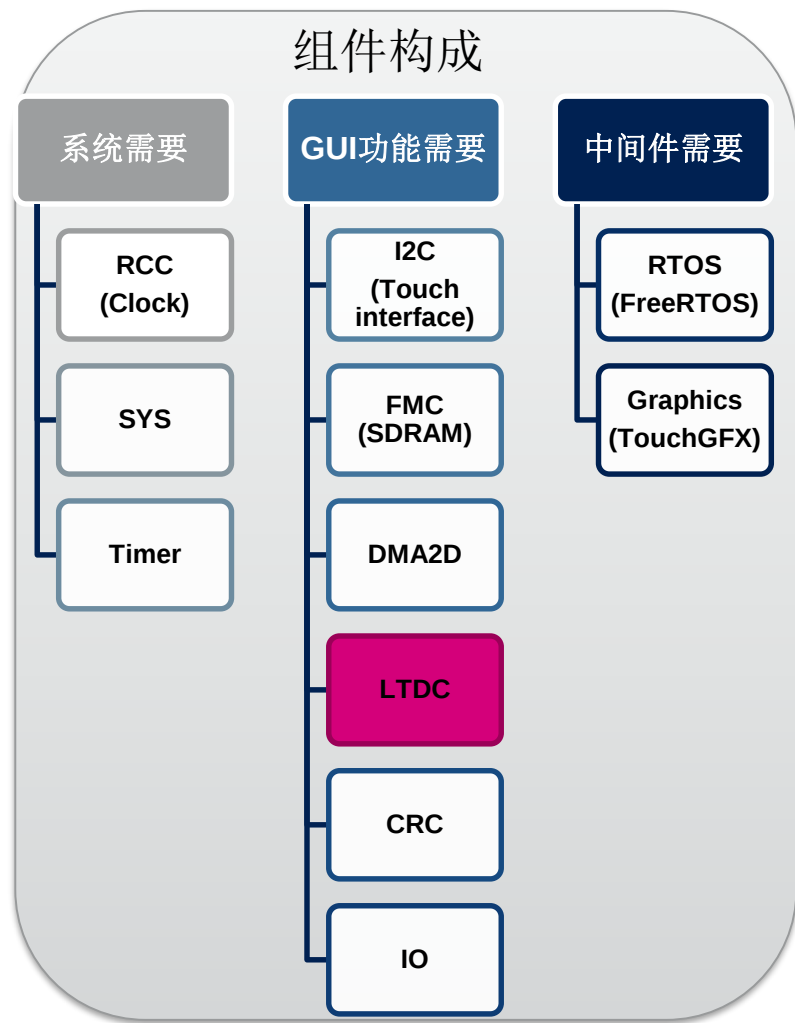
- Parameter settings设置

• 配置参数获取

- 2. 显示屏模组规范书，完成Parameter Settings配置



背景图层颜色（用于图层混合），详情请参考AN4861或者参考手册。
根据显示效果进行配置，演示例中保持默认。





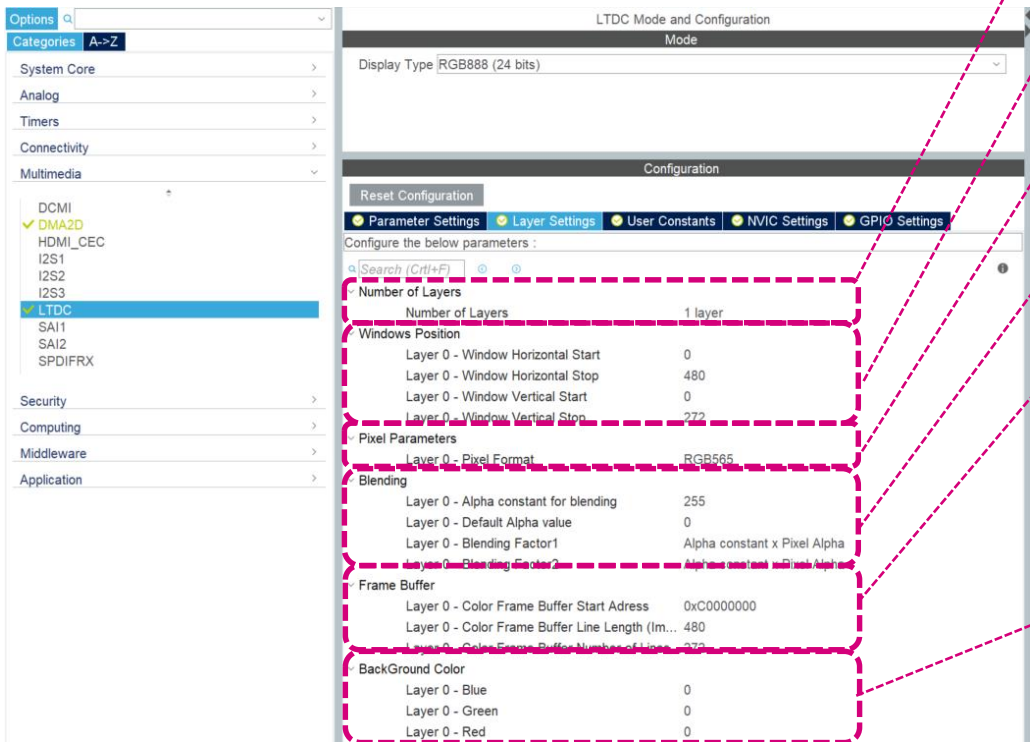
• 组件（GUI功能需要）

• LTDC配置（STM32CubeMX中）

• Layer settings 设置

• 图层配置

• 3. 根据显示要求和处理，完成Layer settings 配置



图层数（LTDC物理图层数：1~2）

图层中显示窗口起始和终止位置

图层颜色格式（对应FrameBuffer中像素数据存放格式。

图层混合方式（请参考《GUI方案中ALPHA通道处理介绍》）

图层FrameBuffer

- 图层FrameBuffer首地址（演示例，FrameBuffer保存在外扩SDRAM。外扩SDRAM BANK1首地址 = 0xC0000000）
- FrameBuffer对应的行列像素个数，STM32CubeMX自动将其与Parameter settings中Active Width/Height关联。

图层默认颜色

- 定义默认图层颜色，详情请参考AN4861或者参考手册。

图. LTDC配置（@STM32CubeMX）



实现示例（续）

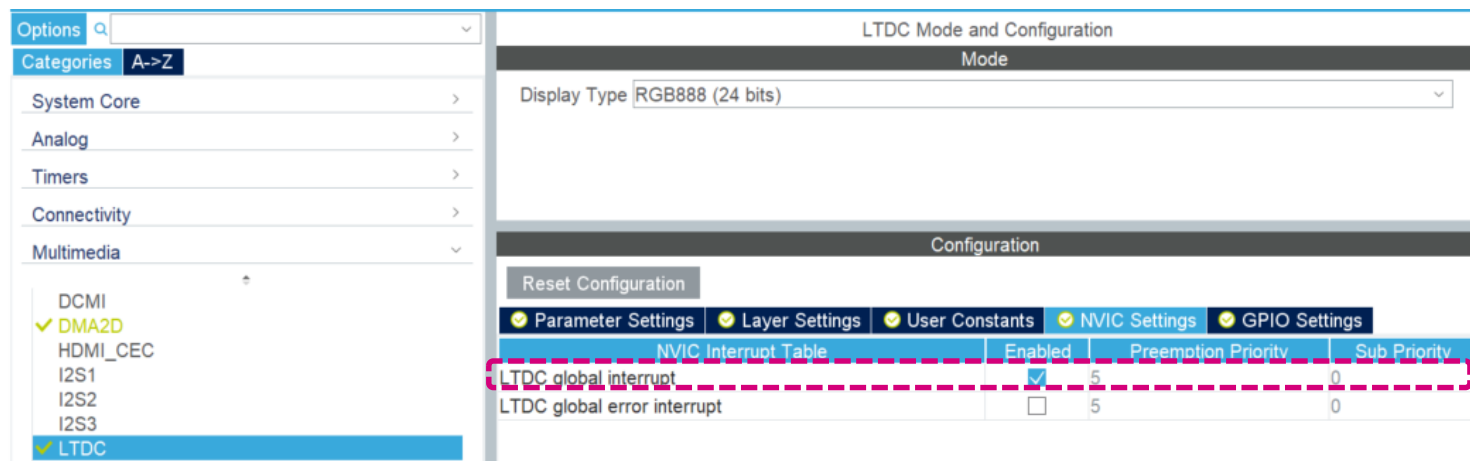
38



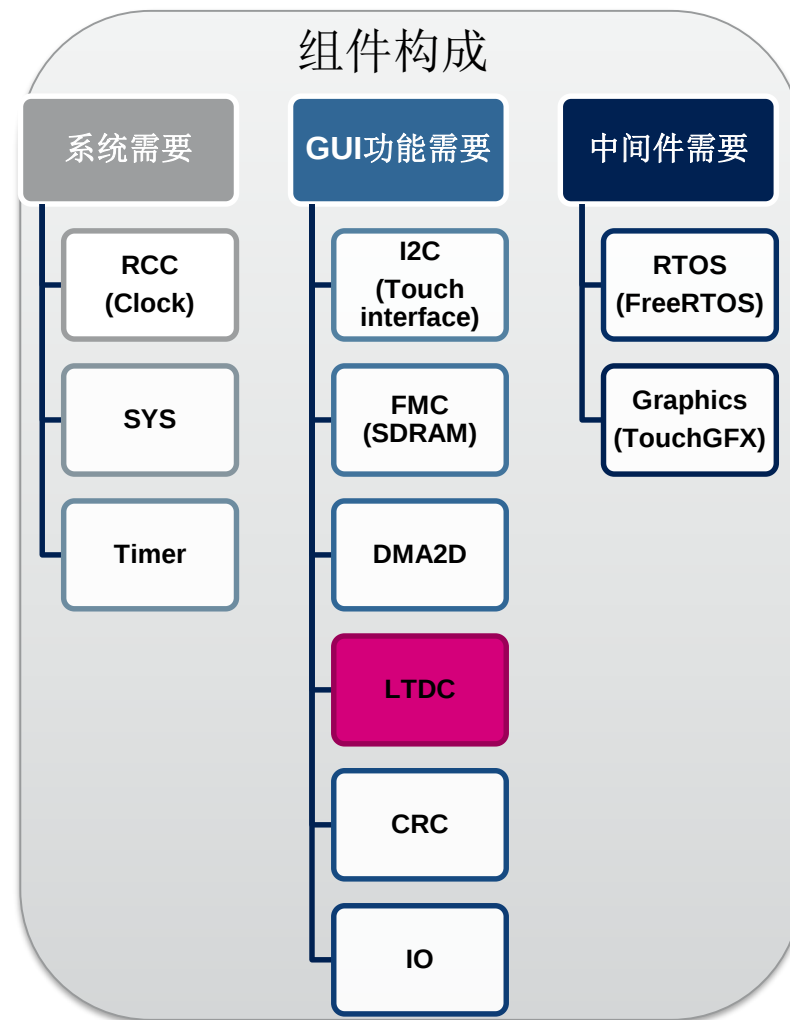
• 组件（GUI功能需要）

• LTDC配置（STM32CubeMX中）

• NVIC settings 设置



组件构成



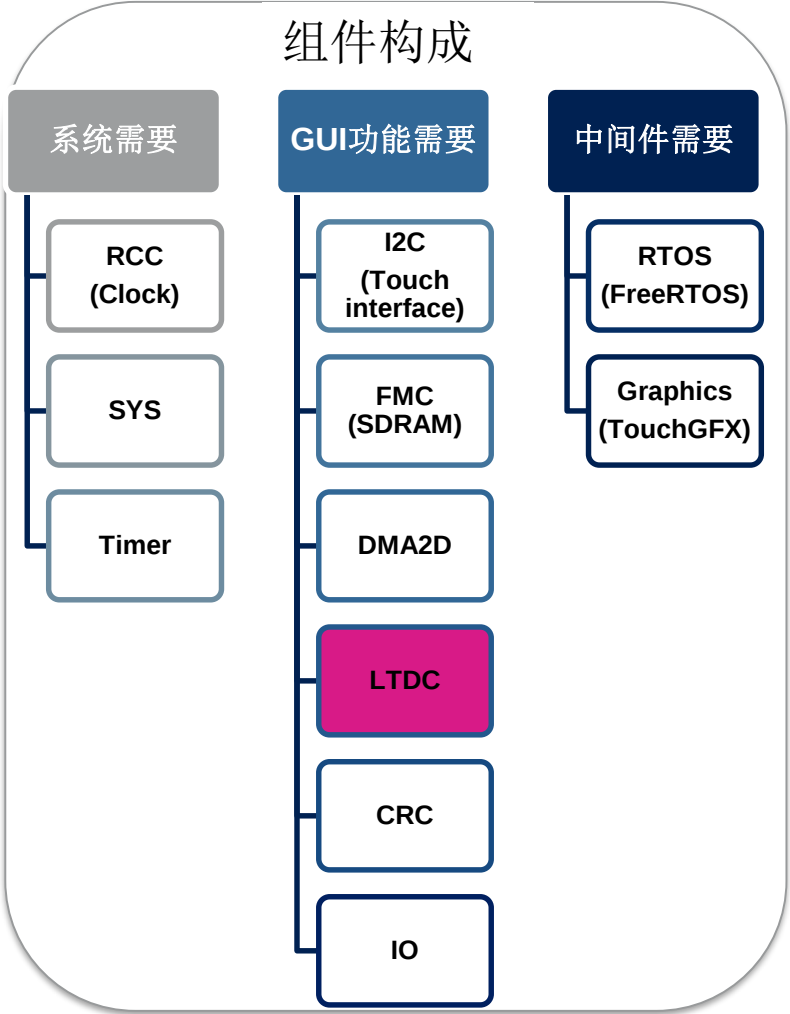


实现示例（续）



- 组件（GUI功能需要）
 - 根据硬件原理图，匹配GPIO Settings

Parameter Settings Layer Settings User Constants NVIC Settings GPIO Settings							
Search Signals							
Search (Ctrl+F)							
☐ Show only Modified Pins							
Pin Name	Signal on Pin	GPIO outp...	GPIO mode	GPIO Pull-...	Maximum o...	User Label	Modified
PE4	LTDC_B0	n/a	Alternate F...	No pull-up ...	Low	LCD_B0 [R...	✓
PG12	LTDC_B4	n/a	Alternate F...	No pull-up ...	Low	LCD_B4 [R...	✓
PI9	LTDC_VSYNC	n/a	Alternate F...	No pull-up ...	Low	LCD_VSY...	✓
PI10	LTDC_HSYNC	n/a	Alternate F...	No pull-up ...	Low	LCD_HSY...	✓
PI14	LTDC_CLK	n/a	Alternate F...	No pull-up ...	Low	LCD_CLK [...	✓
PI15	LTDC_R0	n/a	Alternate F...	No pull-up ...	Low	LCD_R0 [R...	✓
PJ0	LTDC_R1	n/a	Alternate F...	No pull-up ...	Low	LCD_R1 [R...	✓
PJ1	LTDC_R2	n/a	Alternate F...	No pull-up ...	Low	LCD_R2 [R...	✓
PJ2	LTDC_R3	n/a	Alternate F...	No pull-up ...	Low	LCD_R3 [R...	✓
PJ3	LTDC_R4	n/a	Alternate F...	No pull-up ...	Low	LCD_R4 [R...	✓
PJ4	LTDC_R5	n/a	Alternate F...	No pull-up ...	Low	LCD_R5 [R...	✓
PJ5	LTDC_R6	n/a	Alternate F...	No pull-up ...	Low	LCD_R6 [R...	✓
PJ6	LTDC_R7	n/a	Alternate F...	No pull-up ...	Low	LCD_R7 [R...	✓
PJ7	LTDC_G0	n/a	Alternate F...	No pull-up ...	Low	LCD_G0 [...	✓
PJ8	LTDC_G1	n/a	Alternate F...	No pull-up ...	Low	LCD_G1 [...	✓
PJ9	LTDC_G2	n/a	Alternate F...	No pull-up ...	Low	LCD_G2 [...	✓
PJ10	LTDC_G3	n/a	Alternate F...	No pull-up ...	Low	LCD_G3 [...	✓
PJ11	LTDC_G4	n/a	Alternate F...	No pull-up ...	Low	LCD_G4 [...	✓
PJ13	LTDC_B1	n/a	Alternate F...	No pull-up ...	Low	LCD_B1 [R...	✓
PJ14	LTDC_B2	n/a	Alternate F...	No pull-up ...	Low	LCD_B2 [R...	✓
PJ15	LTDC_B3	n/a	Alternate F...	No pull-up ...	Low	LCD_B3 [R...	✓
PK0	LTDC_G5	n/a	Alternate F...	No pull-up ...	Low	LCD_G5 [...	✓
PK1	LTDC_G6	n/a	Alternate F...	No pull-up ...	Low	LCD_G6 [...	✓
PK2	LTDC_G7	n/a	Alternate F...	No pull-up ...	Low	LCD_G7 [...	✓
PK4	LTDC_B5	n/a	Alternate F...	No pull-up ...	Low	LCD_B5 [R...	✓
PK5	LTDC_B6	n/a	Alternate F...	No pull-up ...	Low	LCD_B6 [R...	✓
PK6	LTDC_B7	n/a	Alternate F...	No pull-up ...	Low	LCD_B7 [R...	✓
PK7	LTDC_DE	n/a	Alternate F...	No pull-up ...	Low	LCD_DE [...	✓





- 组件（GUI功能需要）
 - 另外LCD的需要配置如下2个GPIO，分别是控制背光，显示使能

组件构成

LCD_DISP:
0: LCD进入Sleep模式 1: Normal模式（上电默认拉高）

PI12 Configuration :

GPIO output level	High
GPIO mode	Output Push Pull
GPIO Pull-up/Pull-down	No pull-up and no pull-down
Maximum output speed	Low
User Label	LCD_DISP [RK043FN48H-CT672B_DISP]

LCD_BL_CTRL:
0: 关闭背光 1: 打开背光（上电默认拉高）

PK3 Configuration :

GPIO output level	High
GPIO mode	Output Push Pull
GPIO Pull-up/Pull-down	No pull-up and no pull-down
Maximum output speed	Low
User Label	LCD_BL_CTRL [STLD40DPUR_EN]



实现示例（续）

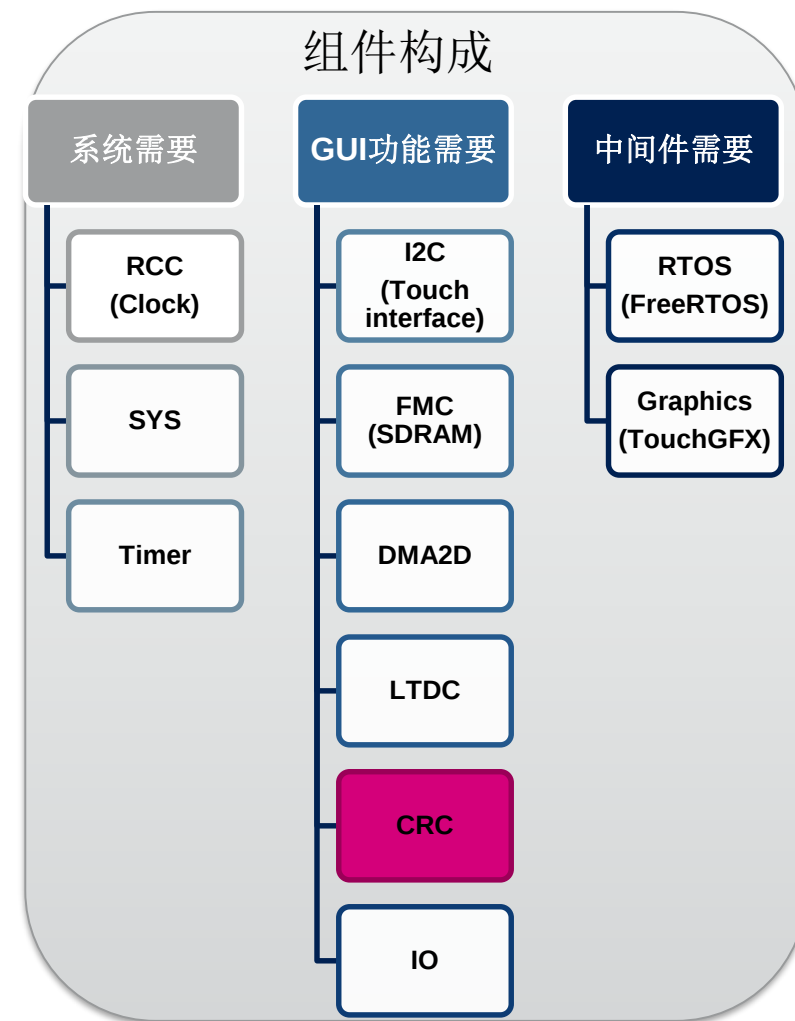
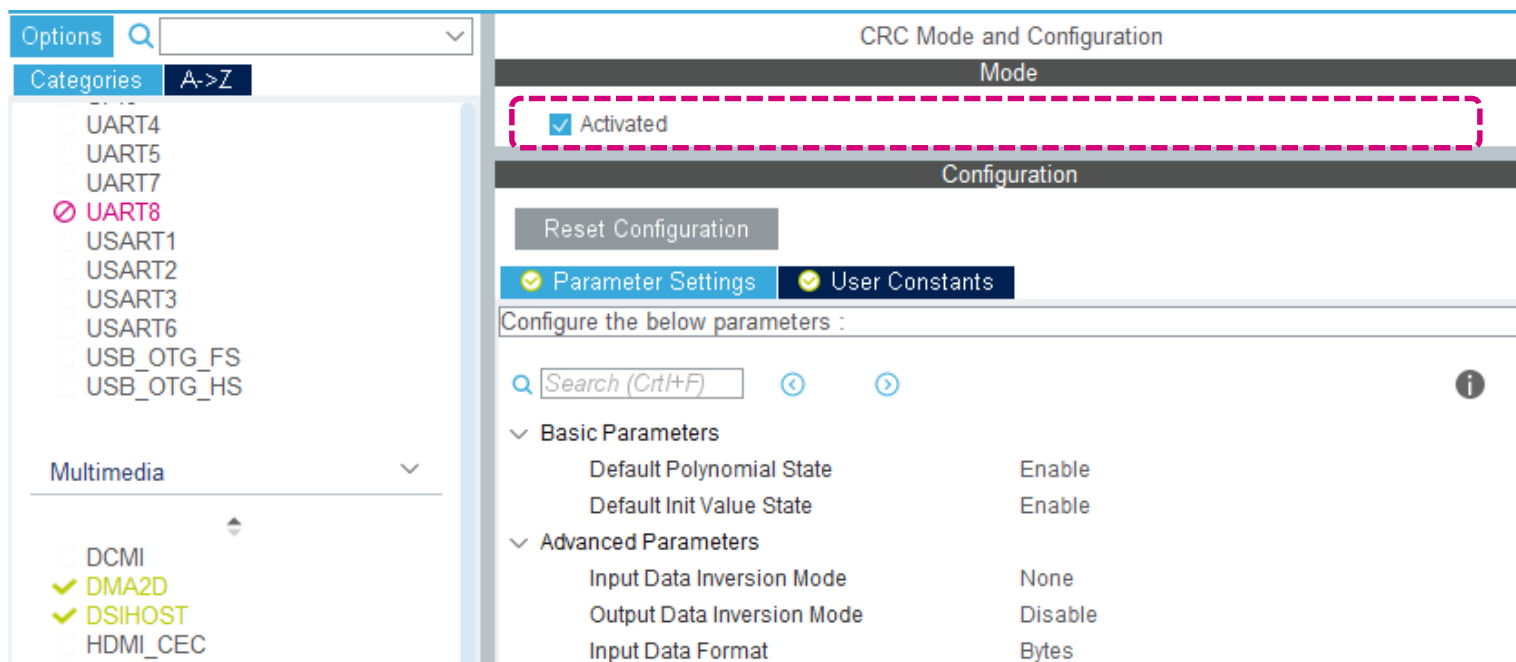
41



• IP组件（GUI功能需要）

• CRC

- TouchGFX中间件需要。保持默认值。





实现示例（续）

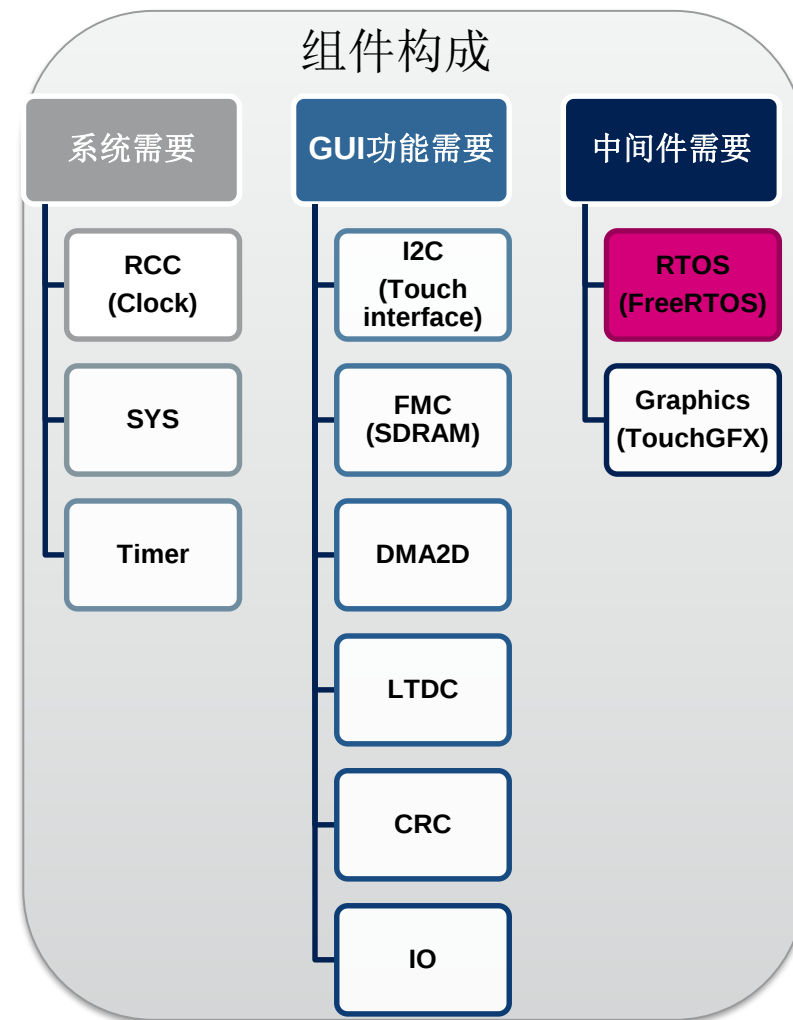
42



• 组件（中间件需要）

• FreeRTOS

- STM32CubeMX对TouchGFX开发支持，需要使用RTOS。





实现示例（续）

43

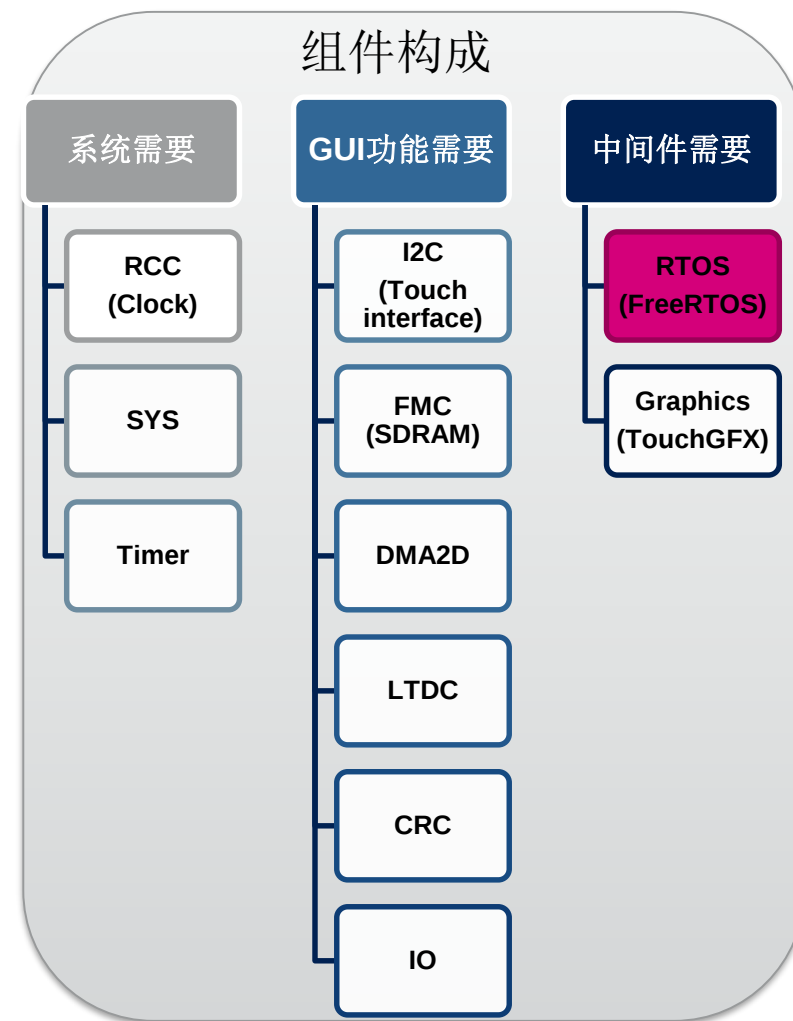
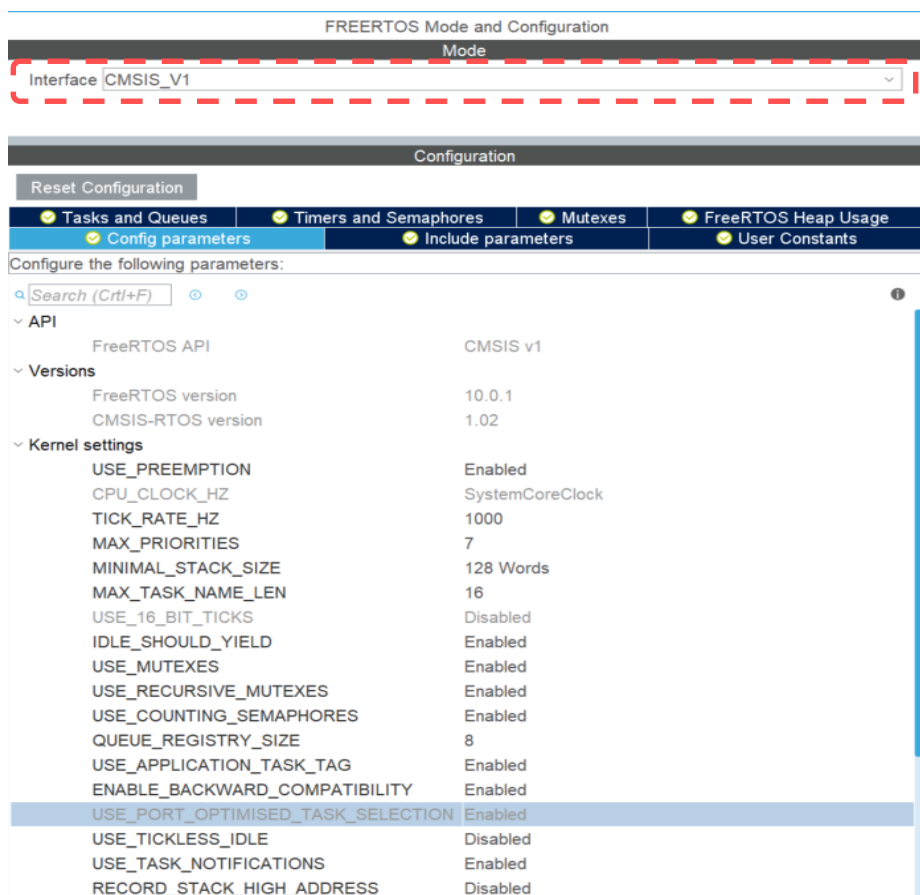


• 组件（中间件需要）

• FreeRTOS

• 1. 使能FreeRTOS。

- STM32CubeMX对TouchGFX开发支持，需要使用RTOS。





实现示例（续）

44



• 组件（中间件需要）

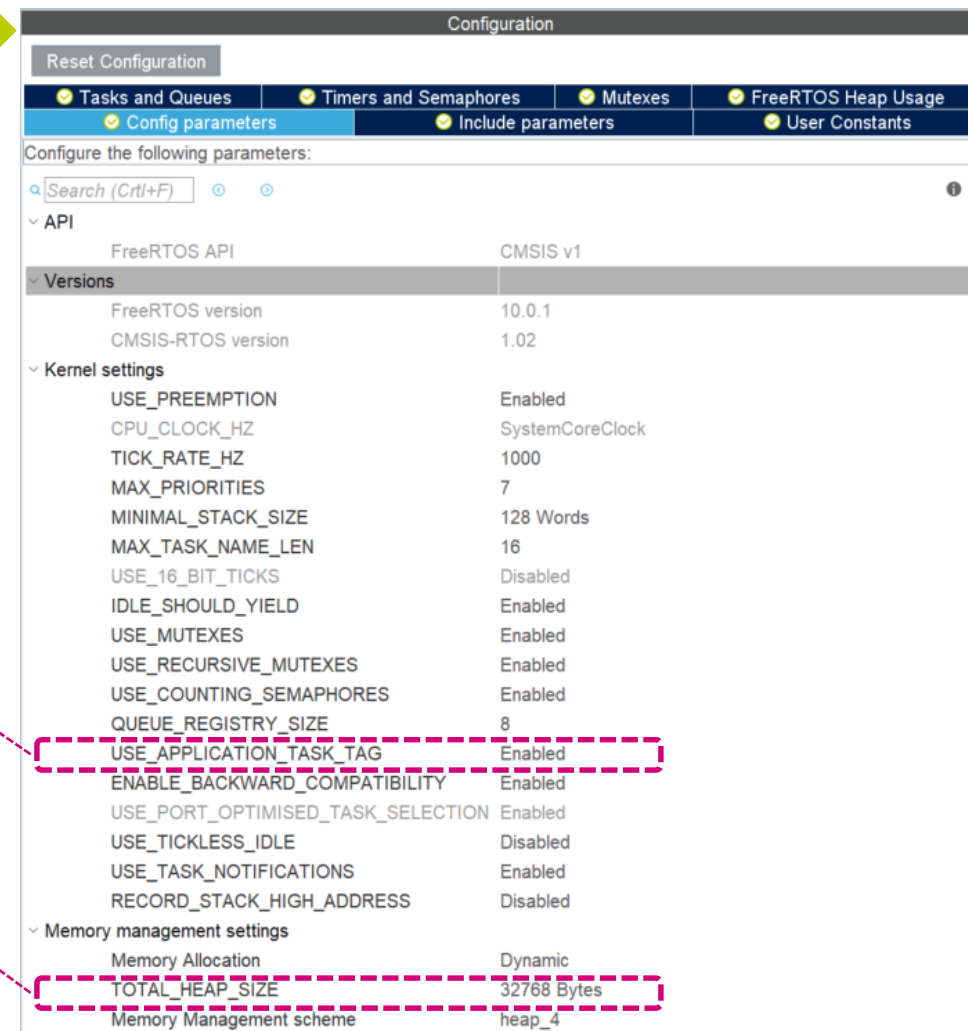
• FreeRTOS

• 2. 配置参数。

- 在使能FreeRTOS时，已经默认了一些常用配置。在这些配置基础上，针对应用情况，进行调整。
- 这里对调整部分进行介绍，其他部分请参考FreeRTOS帮助文档

- 需要使能USE_APPLICATION_TASK_TAG，测试MCU负载会涉及到这个功能的使用。

- 根据应用需要，调整TOTAL_HEAP_SIZE。调整分配给FreeRTOS的RAM空间大小。





实现示例（续）

45



• 组件（中间件需要）

• FreeRTOS

• 3. 增加任务堆栈空间

• 调整Stack Size空间大小

Task N...	Priority	Stack ...	Entry F...	Code ...	Param...	Allocati...	Buffer ...	Contro...
default...	osPrio...	4096	StartD...	Default	NULL	Dyna...	NULL	NULL

Buttons: Add, Delete

Queues section is also visible with columns: Queue Name, Queue Size, Item Size, Allocation, Buffer Name, Control Blo...



Task Name	defaultTask
Priority	osPriorityNormal
Stack Size (Words)	4096
Entry Function	StartDefaultTask
Code Generation Option	Default
Parameter	NULL
Allocation	Dynamic
Buffer Name	NULL
Control Block Name	NULL

Buttons: OK, Cancel



实现示例（续）

46

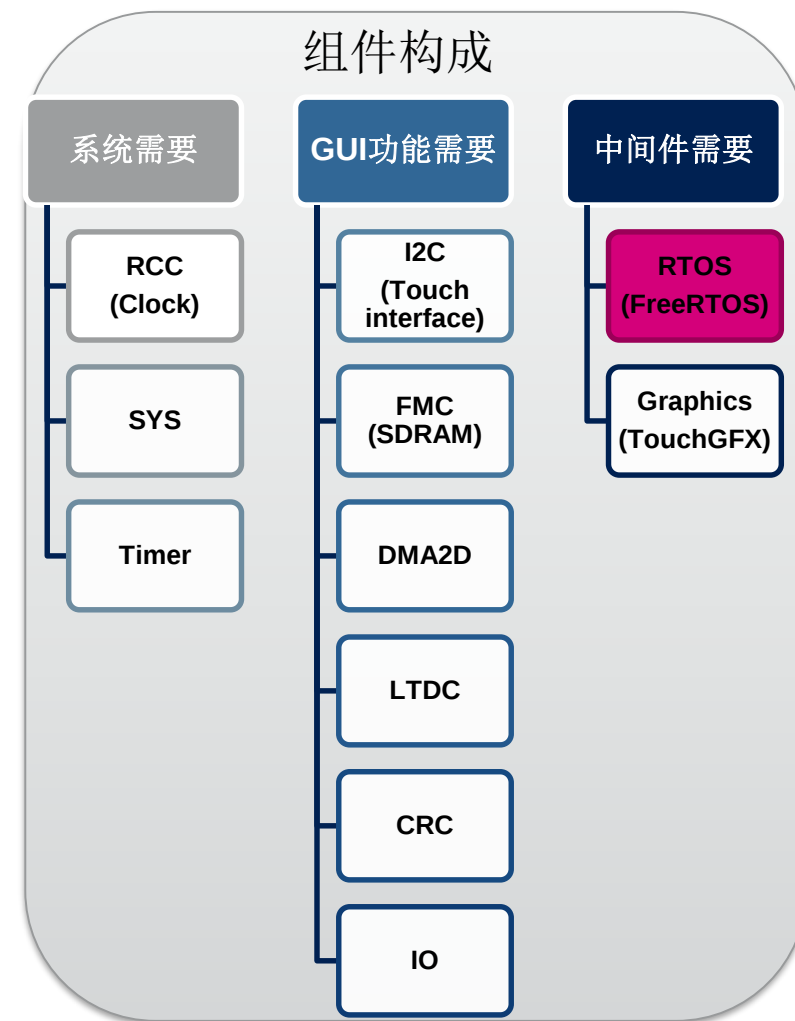


• 组件（中间件需要）

• FreeRTOS

- 除上述介绍配置，FreeRTOS还有其他可供配置项。保持默认（涉及到其他应用功能时，根据需要进行调整）。
- 更多详情可参考
 - [UM1718 STM32CubeMX for STM32 configuration and initialization C code generation](#)
 - [STM32 FreeRTOS培训：FreeRTOS基本原理介绍](#)
 - [STM32 RTOS培训:STM32嵌入式操作系统介绍](#)
 - [FreeRTOS ST April Training](#)
 - [FreeRTOS官网](#)

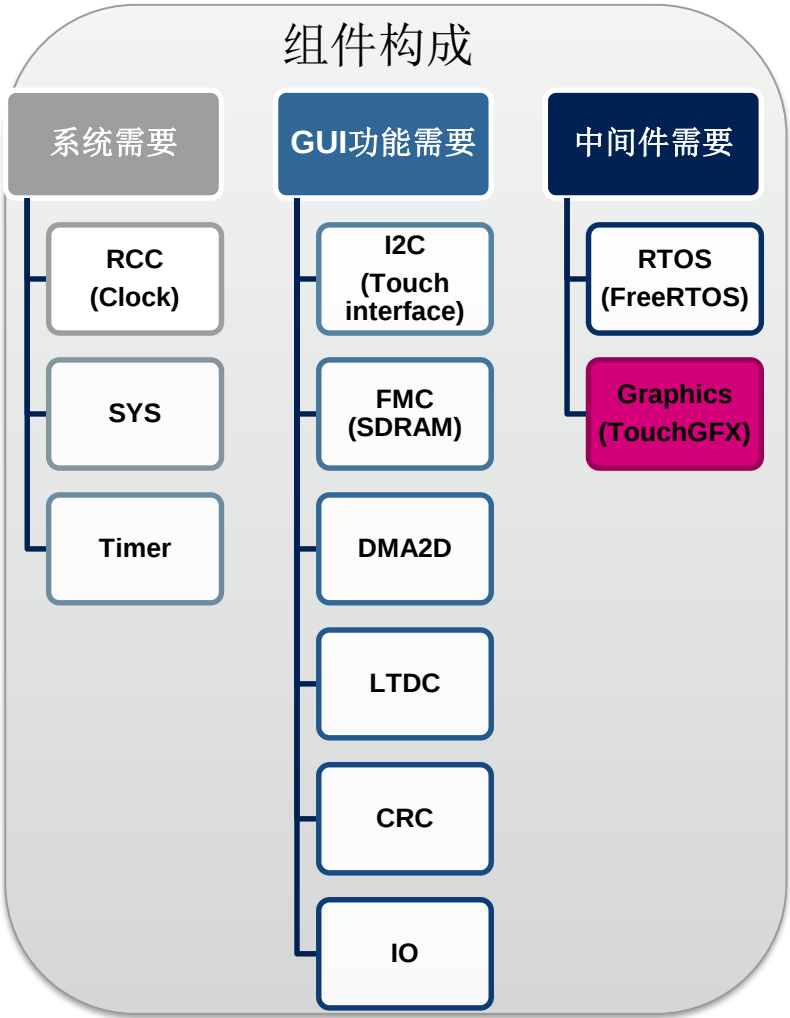
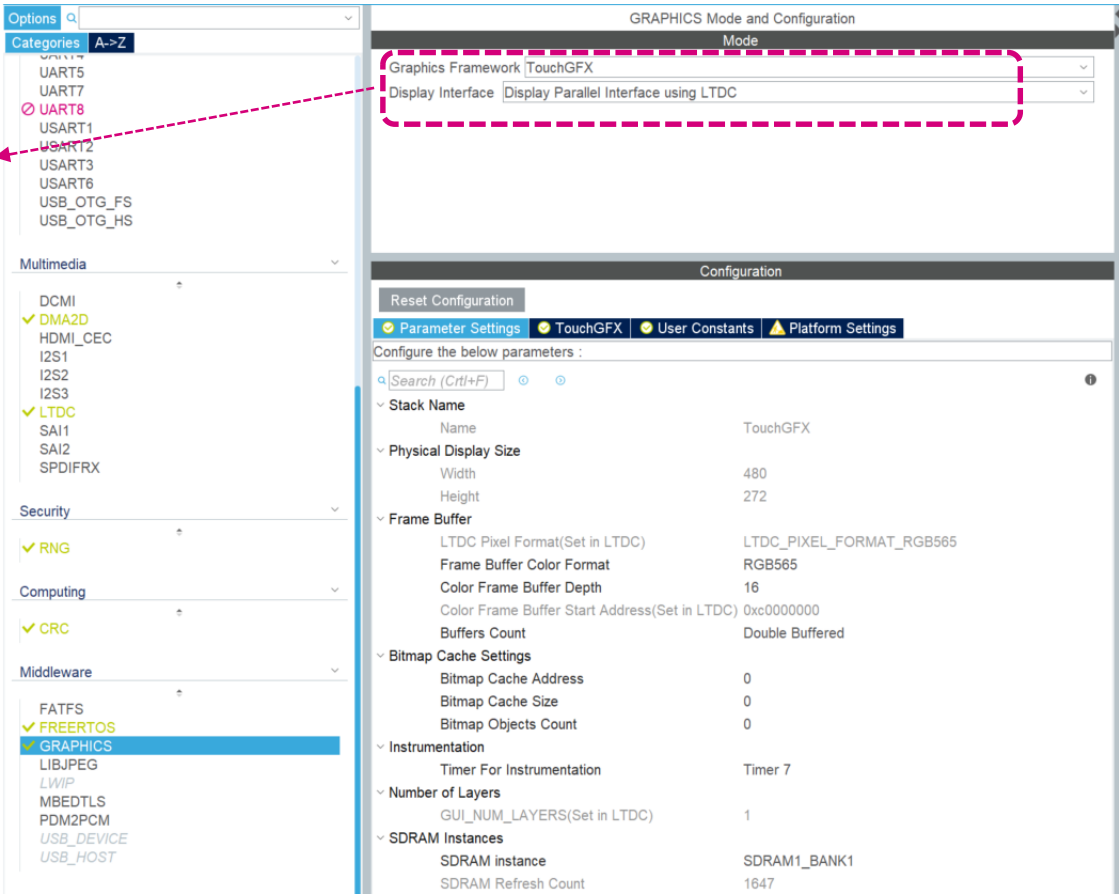
STM32Cube软件包中，提供了多个FreeRTOS示例可供参考。
[STM32CubeF7获取链接](#)





- 组件（GUI功能需要）
 - TouchGFX（GUI中间件）
 - Parameter Settings (参数设置)

- 选择TouchGFX中间件
- 选择显示器接口
 - RGB接口屏选择LTDC
 - DSI接口屏选择LTDC-DSIHOST
 - 8080/6800接口屏选择FMC
 - 注：演示例中，STM32F746G-DISCO连接LTDC接口屏，选择LTDC





实现示例（续）

48



• 组件（GUI功能需要）

• TouchGFX（GUI中间件）

• Parameter Settings (参数设置)

• 对应为显示屏分辨率

• 设置中间件帧缓存

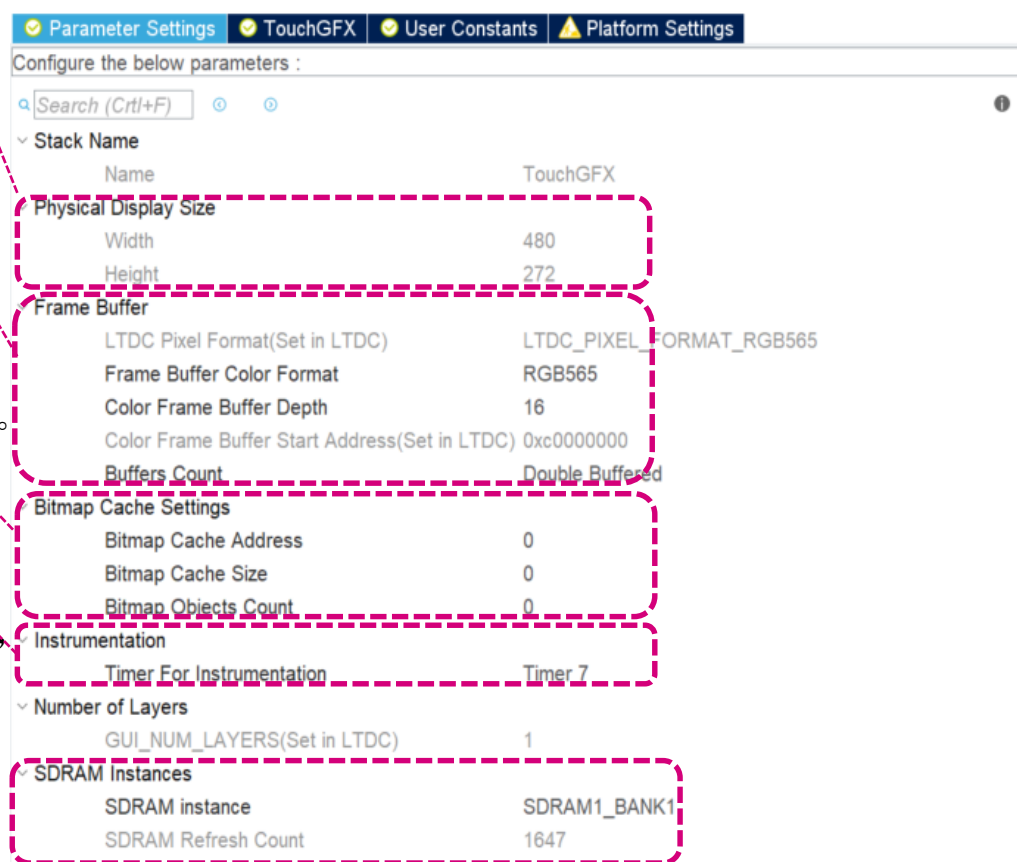
- 帧缓存中颜色格式自定义关联至DSI接口颜色格式

• 设置位图缓存

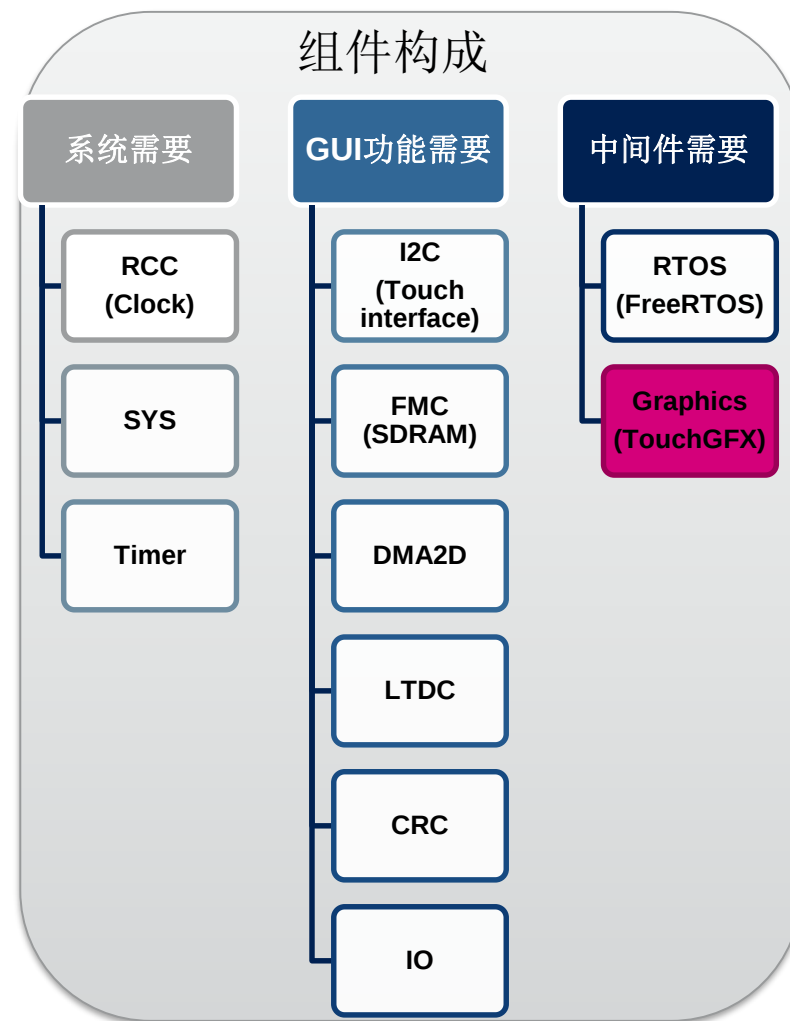
- 适用于位图存储于非存储器映射的存储空间时。
- 演示例中没有用到，保持默认。

• 需要提供一路定时器

- 为RTOS获取MCU负载，提供计数单元。用于在获取空闲任务外MCU主时钟消耗情况，进而获得MCU负载信息。



组件构成





实现示例（续）

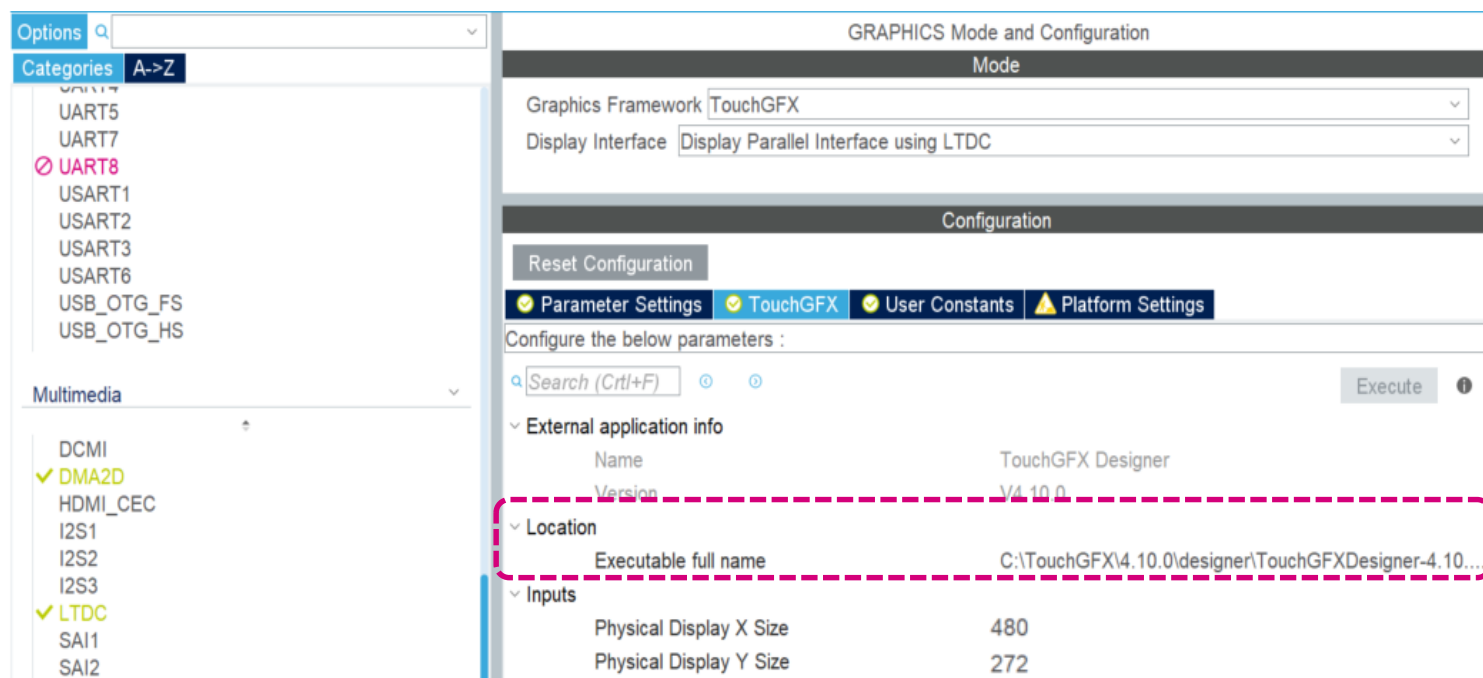
49



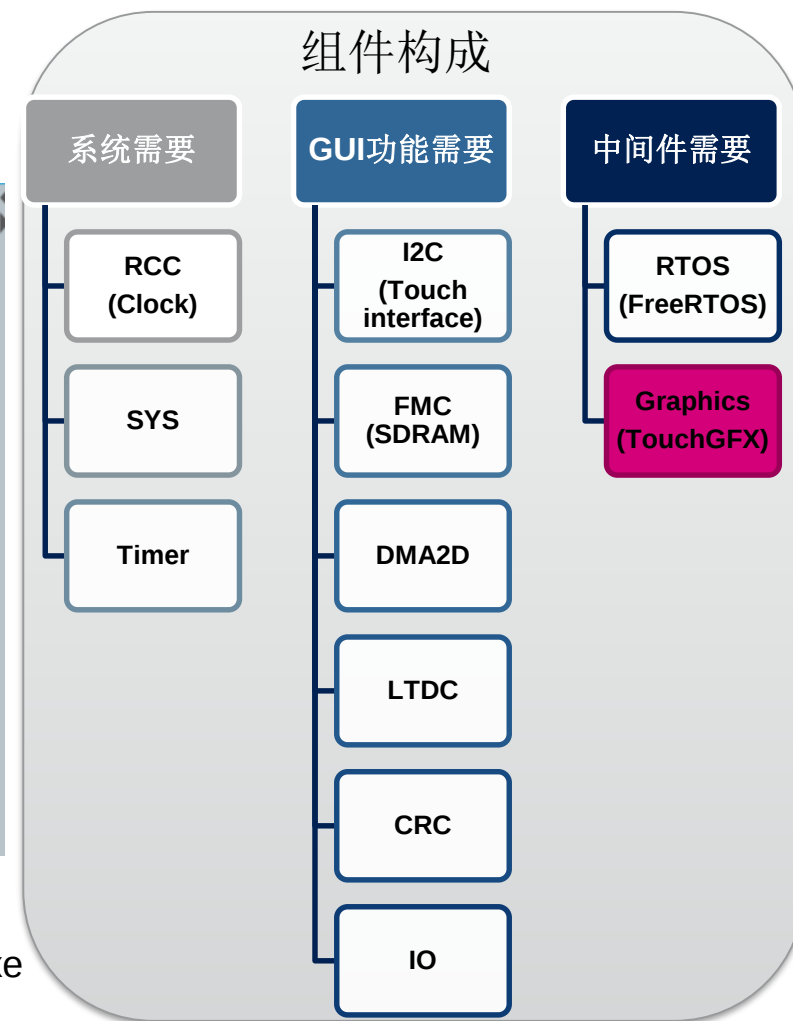
• 组件（GUI功能需要）

• TouchGFX（GUI中间件）

- 指定TouchGFX Designer执行文件 (PC辅助设计工具)



TouchGFXDesigner默认安装路径: C:\TouchGFX\4.10.0\designer\TouchGFXDesigner-4.10.0.exe





实现示例（续）

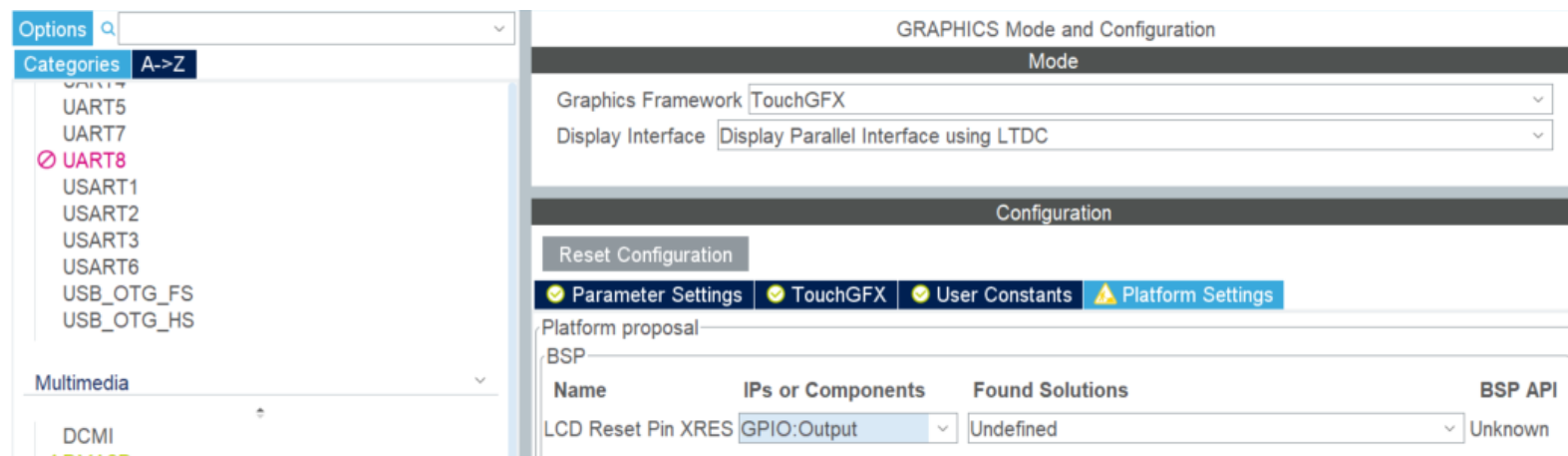
50



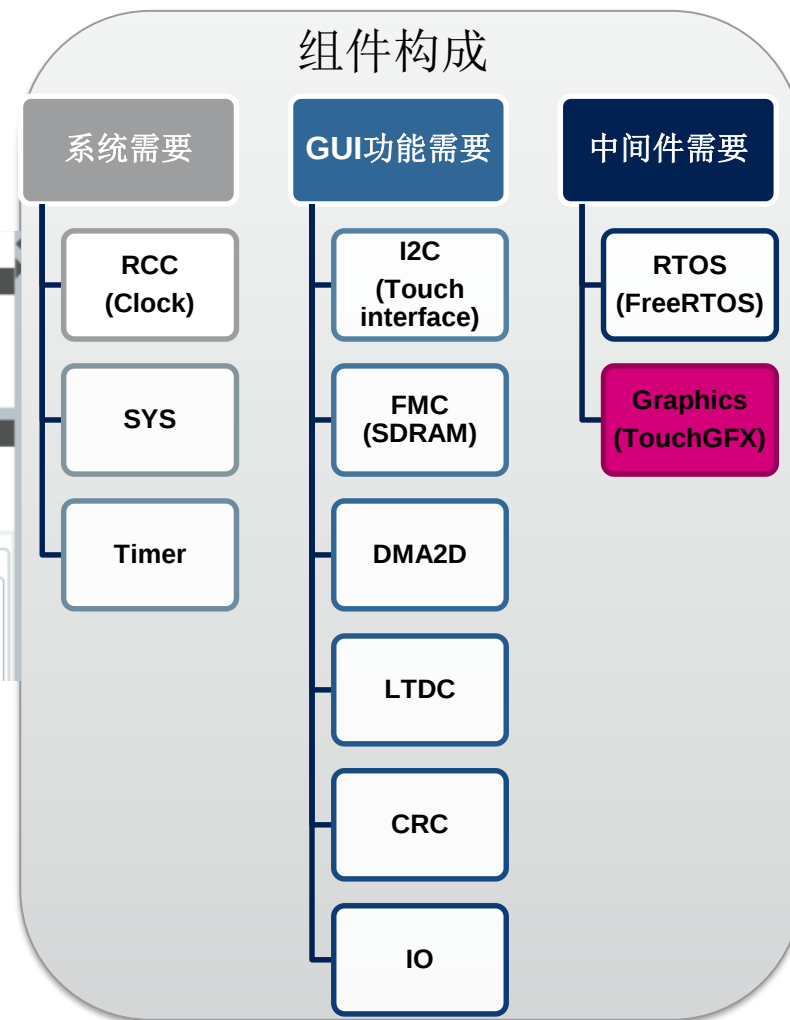
• 组件（GUI功能需要）

• TouchGFX（GUI中间件）

- 指定显示屏复位引脚（STM32F746G-DISCO的LCD reset引脚是上电复位，所以LCD RESET Pin可以不配置）



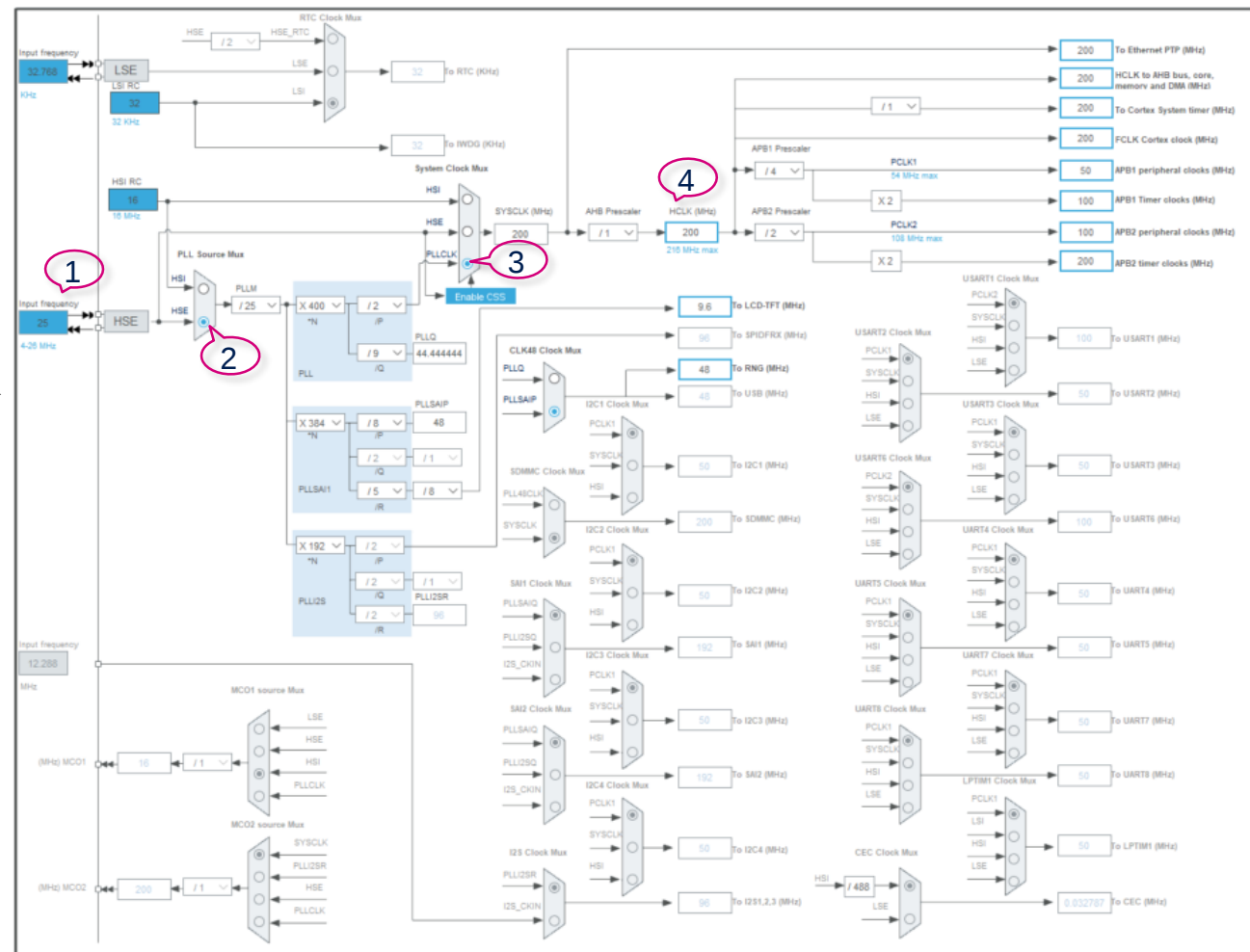
组件构成





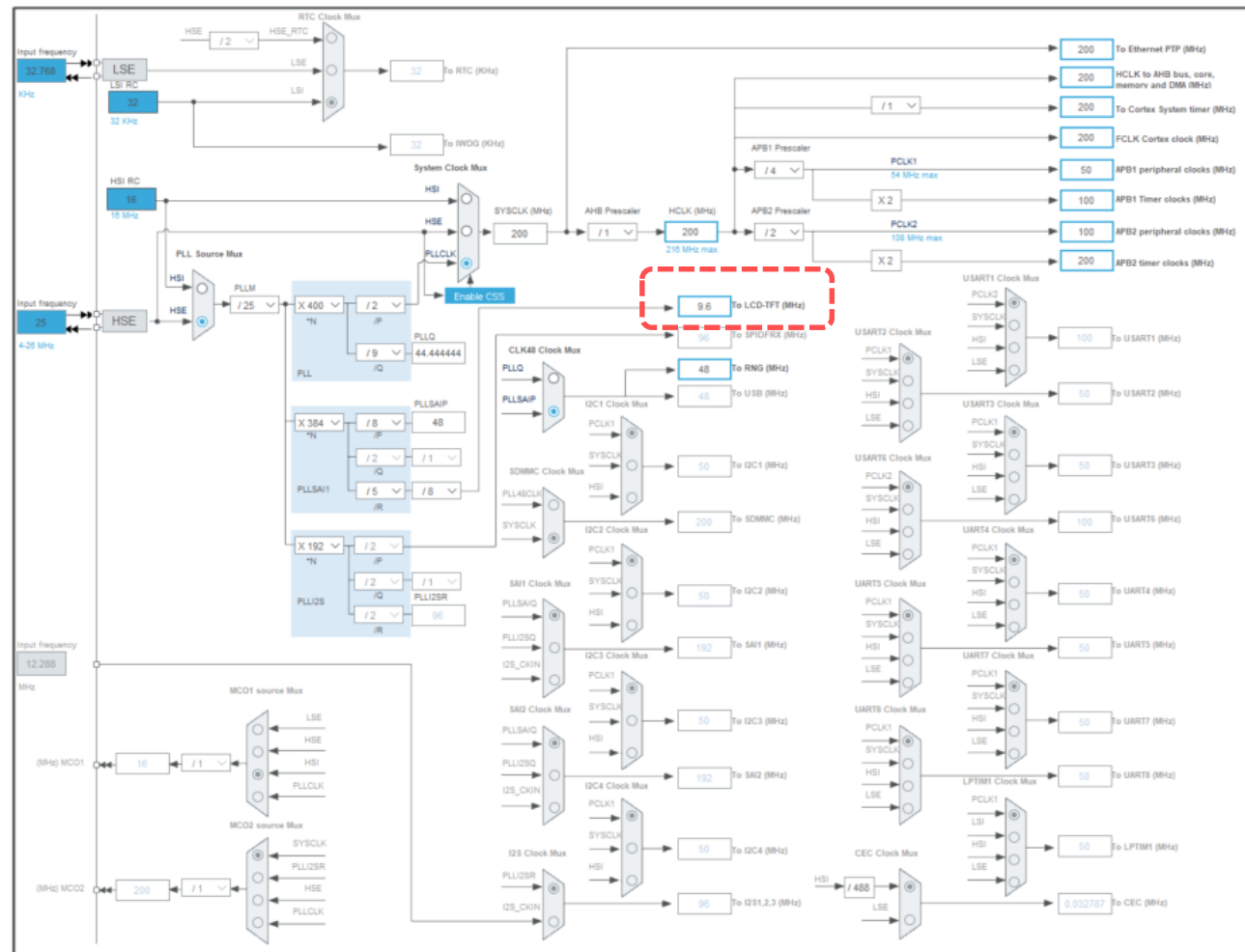
- 主频

- 如果选用外部时钟，根据外部时钟频率，设置HSE。
 - 演示例原理图如下，外部时钟频率为25MHz。
- 根据应用需要，选择时钟源、是否倍频
- 根据应用需要，设置主频和其他时钟频率。
- 演示例中，采用外部时钟和内部PLL倍频，设置最高主频为216MHz，SYSCLK设置成200MHz。





- LCD-TFT (9.6MHz)





实现示例（续）

53



• 设置工程

- 设置工程名、保存路径等
- STM32CubeMX支持生成IAR(EWARM)、MDK、TrueSTUDIO、SW4STM32等工程
 - 演示例中，采用IAR。
- 根据应用情况，调整堆栈大小
 - 演示例中，系统堆栈开销小，保持默认。
- 选择固件包版本，需要先添加固件包至STM32CubeMX
- 更多详细介绍请参考菜单栏Help/Help文档。

Project Settings

Project Name
HelloWorld_CubeMX

Project Location
C:\workshop\STM32F746_GUI_handson\STM32F746_GUI_Hands_On\

Application Structure
Advanced ☐ Do not generate the main()

Toolchain Folder Location
C:\workshop\STM32F746_GUI_handson\STM32F746_GUI_Hands_On\

Toolchain / IDE
EWARM V8 ☐ Generate under Root

Linker Settings
Minimum Heap Size 0x200
Minimum Stack Size 0x400

Mcu and Firmware Package
Mcu Reference
STM32F746NGHx

Firmware Package Name and Version
STM32Cube FW_F7 V1.15.0

☒ Use Default Firmware Location
Users/hans gu/STM32Cube/Repository/STM32Cube_FW_F7_V1.15.0 Browse

Project

Code Generator

Advanced Settings

EWARM V8
EWARM V8
EWARM V7
MDK-ARM V4
MDK-ARM V5
TrueSTUDIO
SW4STM32
Makefile
Other Toolchains (GPDSC)



实现示例（续）

54

选择目标芯片

选择和配置组件
(系统、外设、中间件)

配置时钟树

设置工程

生成工程

执行TouchGFX
Designer

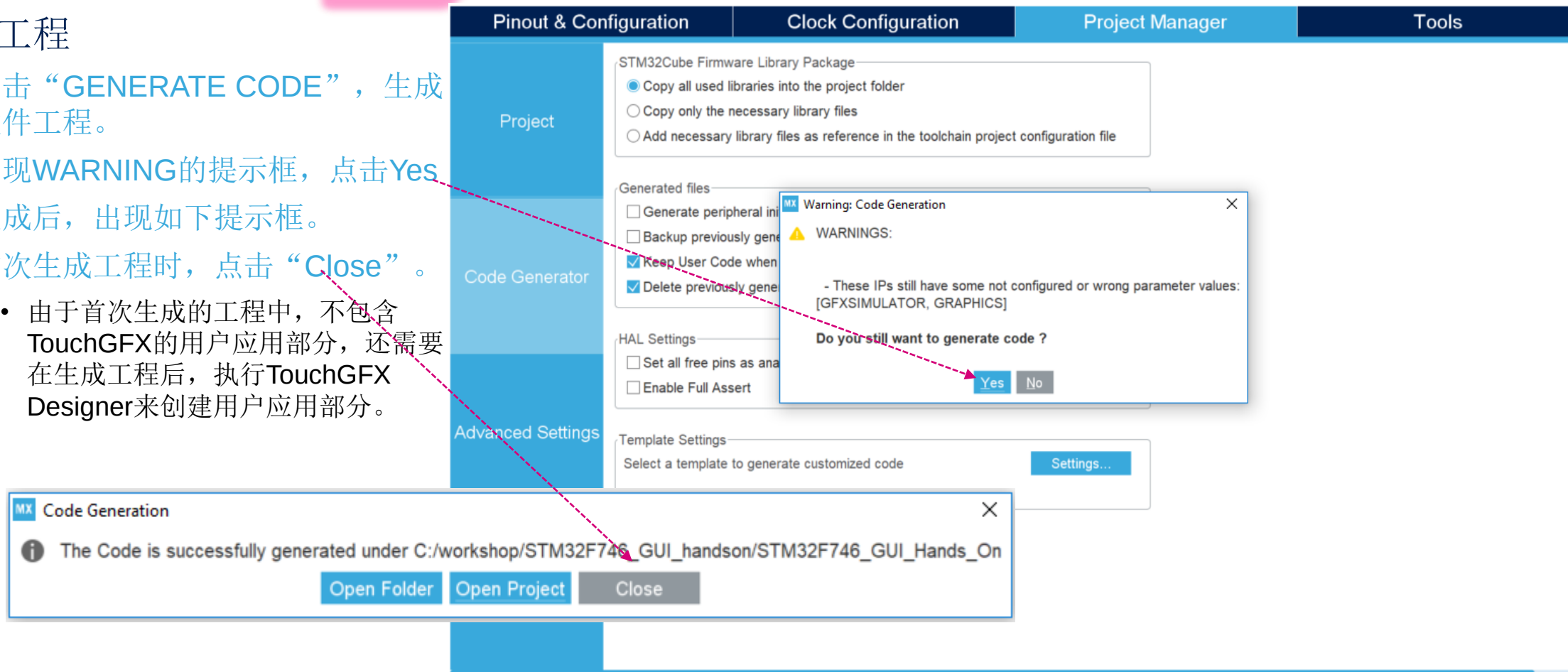
完善工程

编译工程、获取
GUI应用固件

烧录、运行固件

• 生成工程

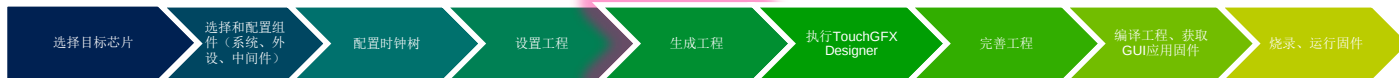
- 点击“GENERATE CODE”，生成软件工程。
- 出现WARNING的提示框，点击Yes
- 生成后，出现如下提示框。
- 首次生成工程时，点击“Close”。
- 由于首次生成的工程中，不包含TouchGFX的用户应用部分，还需要在生成工程后，执行TouchGFX Designer来创建用户应用部分。





实现示例（续）

55



• 生成工程

通过右图步骤，回到TouchGFX中间件的配置界面，点击“Execute”，运行TouchGFX Designer

Options

Categories **A->Z**

- SPDIFRX
- Security >
- Computing v
- CRC
- Middleware v
 - FATFS
 - ✓ FREERTOS
 - ✓ GRAPHICS
 - LIBJPEG
 - LWIP
 - MBEDTLS
 - PDM2PCM
 - USB_DEVICE
 - USB_HOST
- Application >

GRAPHICS Mode and Configuration

Mode

Graphics Framework TouchGFX v

Display Interface Display Parallel Interface using LTDC v

Configuration

Reset Configuration

✓ TouchGFX ✓ User Constants ⚠ Platform Settings

✓ Parameter Settings

Configure the below parameters :

Execute ⓘ

External application info

Name	TouchGFX Designer
Version	V4.10.0

Location

Executable full name	C:\TouchGFX\4.10.0\designer\Touch...
----------------------	--------------------------------------

Inputs

Physical Display X Size	480
Physical Display Y Size	272



实现示例（续）

56



- 执行TouchGFX Designer
 - 利用PC辅助设计工具，设计GUI应用

Options

Categories A-Z

- SPDIFRX
- Security
- Computing
- CRC
- Middleware
 - FATFS
 - FREERTOS
 - GRAPHICS
 - LIBJPEG
 - LWIP
 - MBEDTLS
 - PDM2PCM
 - USB_DEVICE
 - USB_HOST
- Application

GRAPHICS Mode and Configuration

Mode

Graphics Framework TouchGFX

Display Interface Display Parallel Interface using LTDC

Configuration

Reset Configuration

TouchGFX User Constants Platform Settings

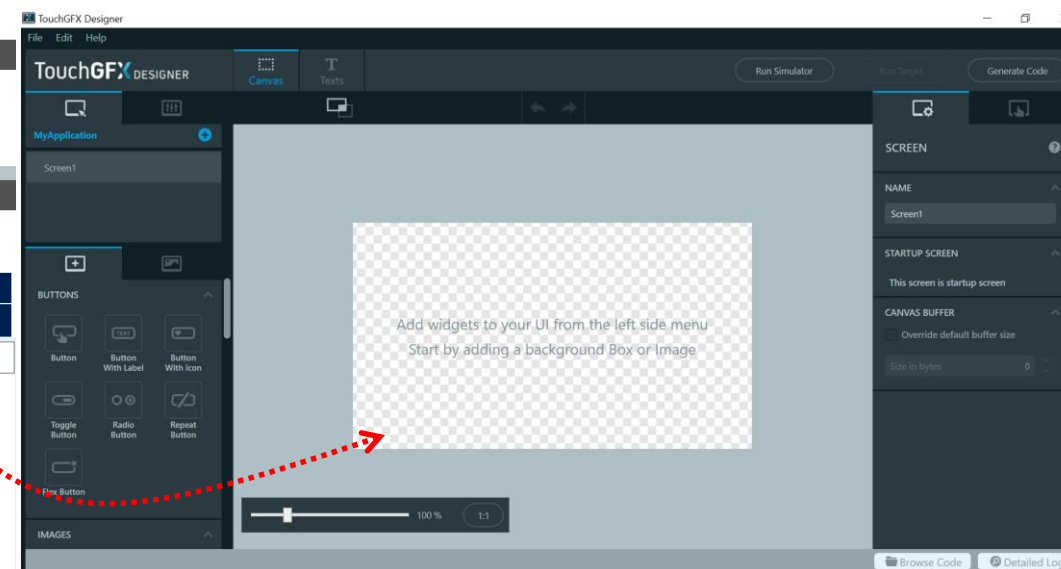
Parameter Settings

Configure the below parameters :

Search (Ctrl+F)

Execute

External application info	
Name	TouchGFX Designer
Version	V4.10.0
Location	
Executable full name	C:\TouchGFX\4.10.0\designer\Touch...
Inputs	
Physical Display X Size	480
Physical Display Y Size	272





实现示例（续）



• 执行TouchGFX Designer

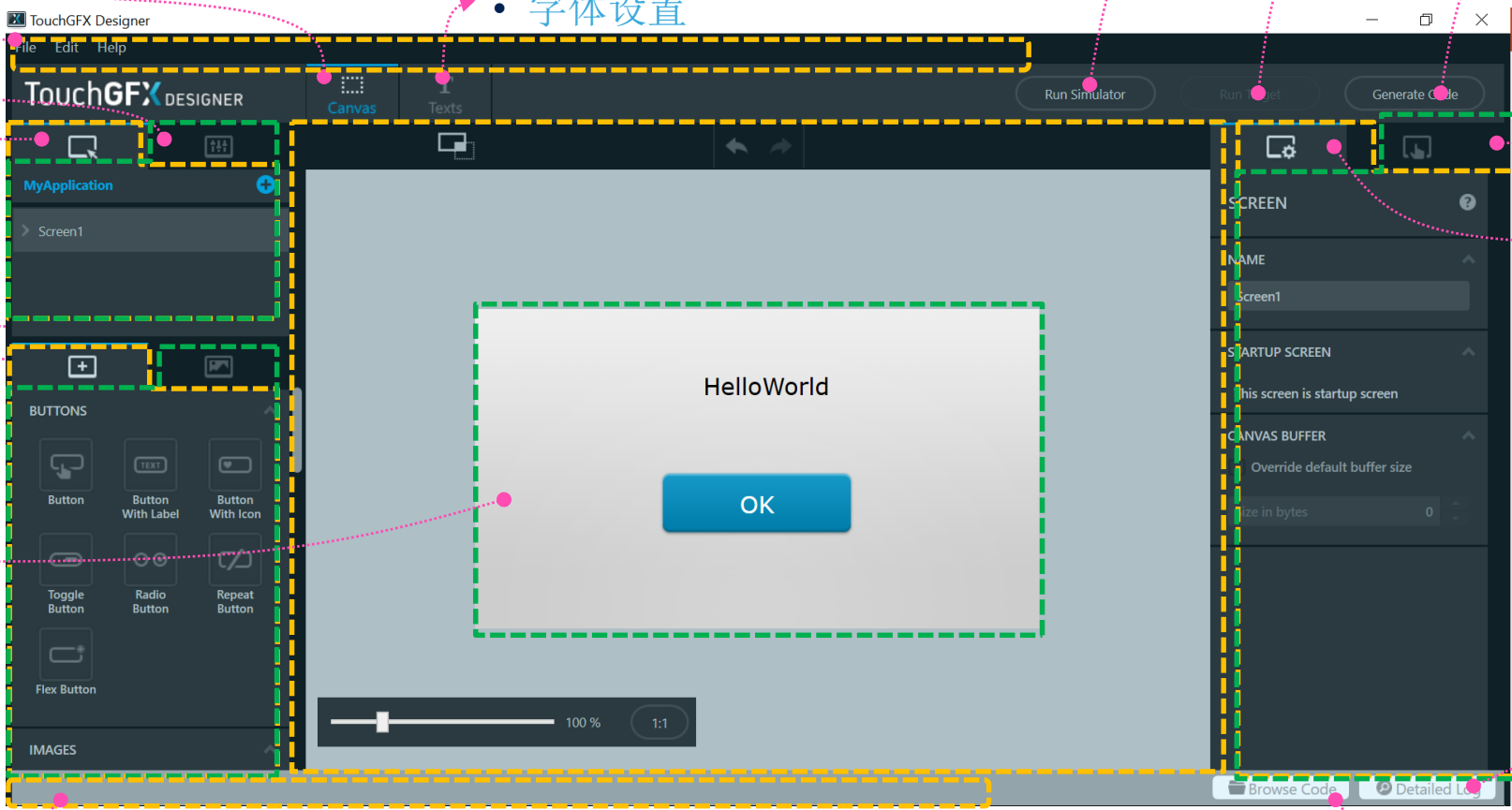
• TouchGFX Designer界面

• 在添加GUI应用前，先了解下TouchGFX Designer界面

- 画布
- 菜单栏
- 自定义控件
- 界面

- 图片素材标签页
- 控件标签页

- 设计窗口



• 状态栏

- 仿真
- 运行目标
- 生成代码

- 交互
- 属性设置

- Log信息

• 打开代码所在目录





实现示例（续）

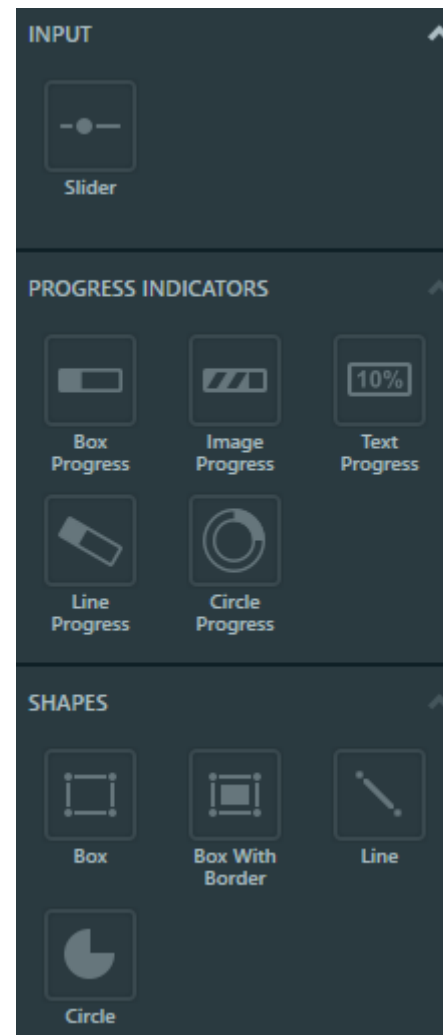
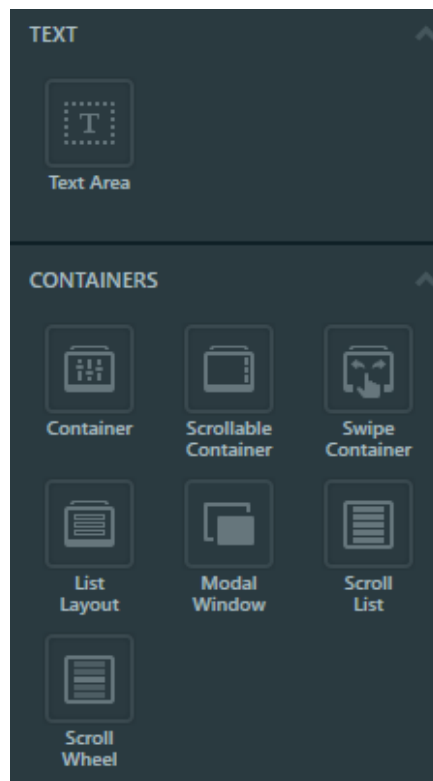
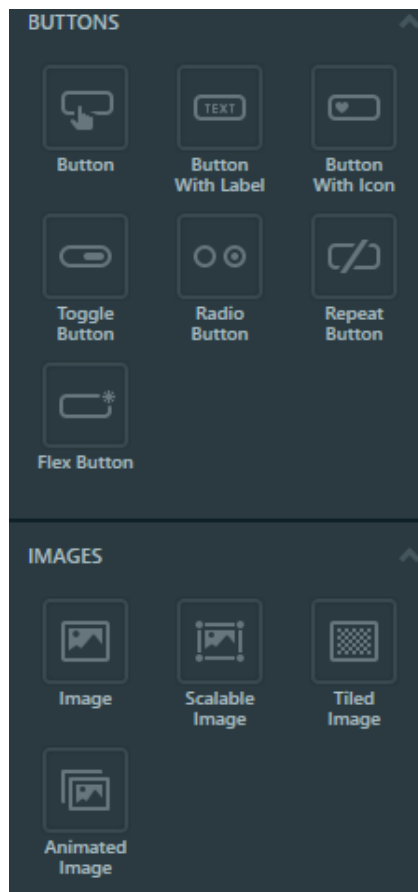
58



• 执行TouchGFX Designer

• TouchGFX Designer界面

- 其中，控件标签页中，包含如下控件。方便GUI开发者快速实现GUI应用设计。





实现示例（续）

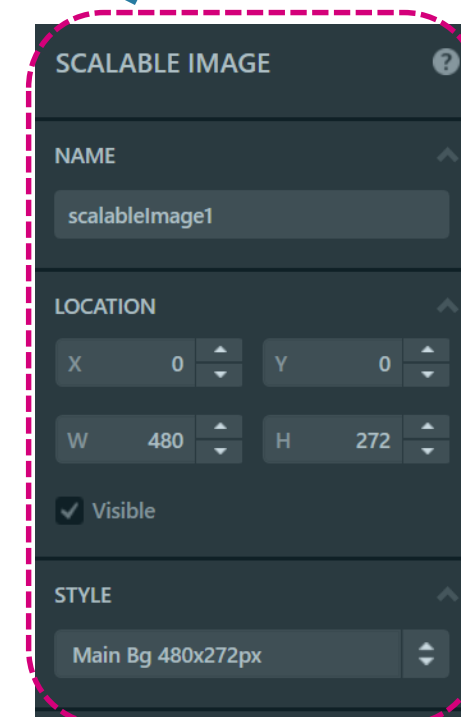
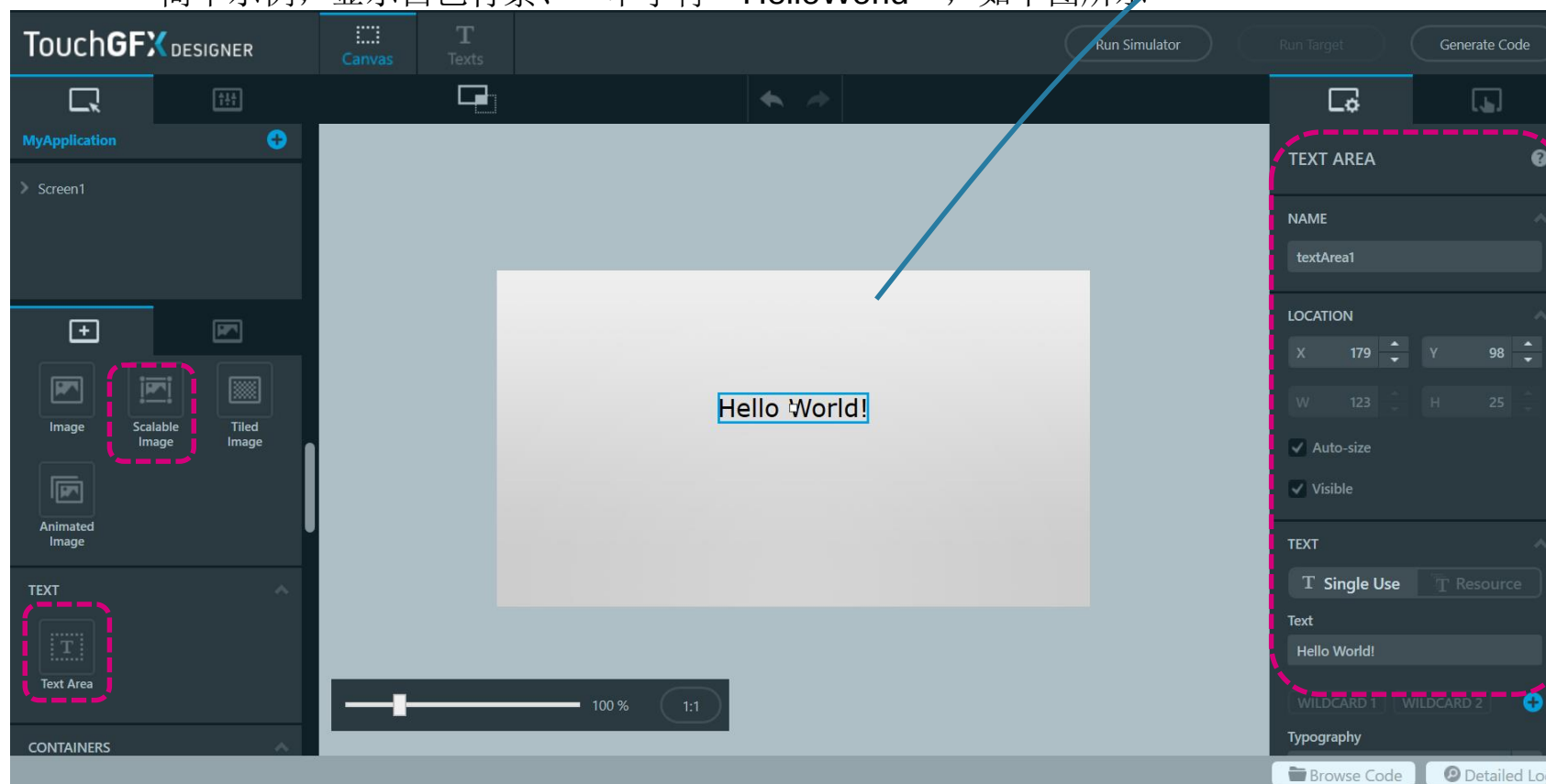
59



• 执行TouchGFX Designer

• 添加GUI应用。

• 简单示例，显示白色背景、一串字符“HelloWorld”，如下图所示





实现示例（续）

60



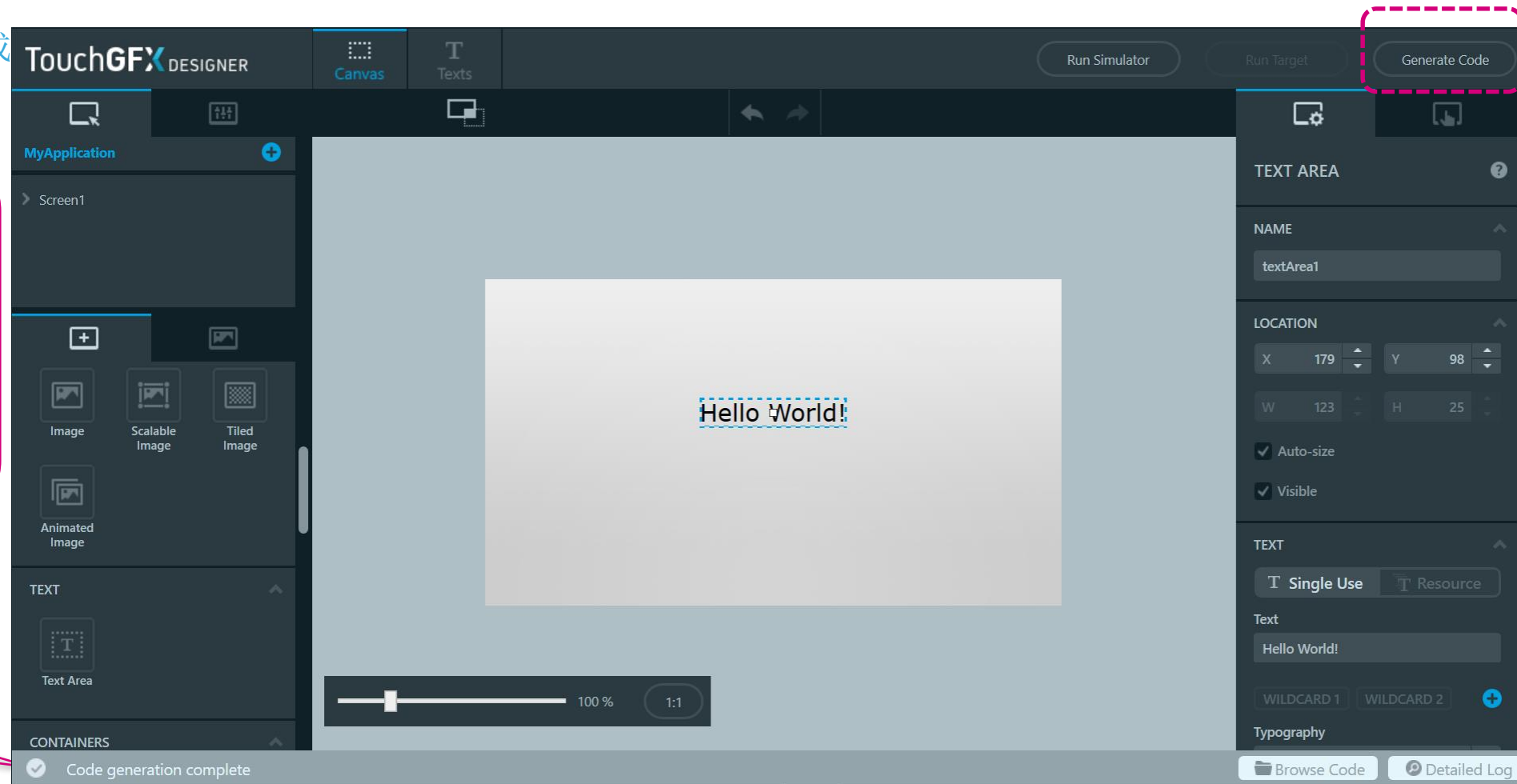
• 执行TouchGFX Designer

- 然后，点击右上角“Generate Code”，生成TouchGFX用户的应用代码。左下角会显示处理状态。
- 等待出现生成完成

注：详细TouchGFX Designer使用及GUI应用开发介绍，请参考：

[TouchGFX官网](#)

TouchGFX Designer
自带示例（通过
File\New选择）



代码生成
完成提示



实现示例（续）

61



• 执行TouchGFX Designer

- 至此，STM32CubeMX中配置工作完成。
- 再次点击STM32CubeMX中的“GENERATE CODE”，生成IDE工程。并且点击弹出菜单栏的“Open Project”。
- 工程中文件结构如图所示。

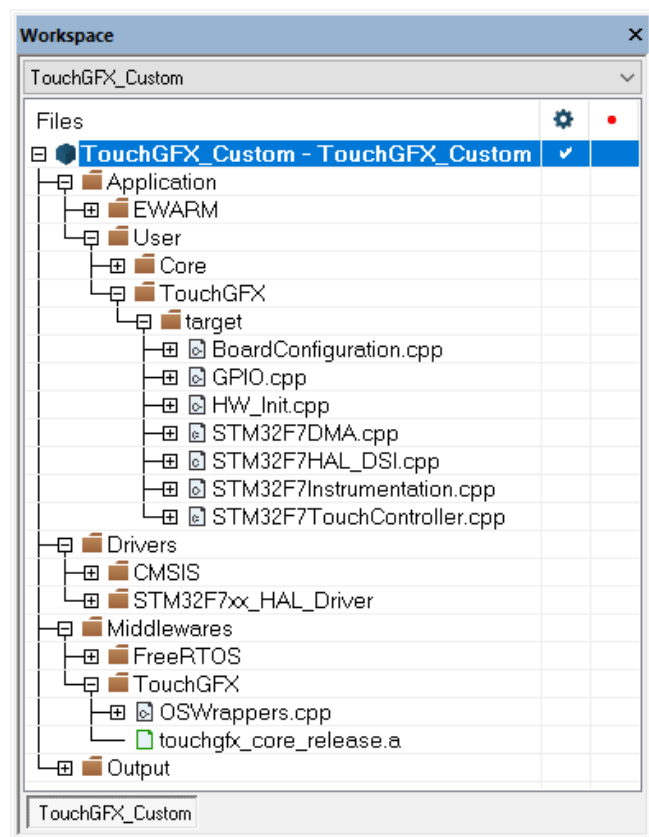


图1. IDE工程文件结构
(运行TouchGFX Designer前)

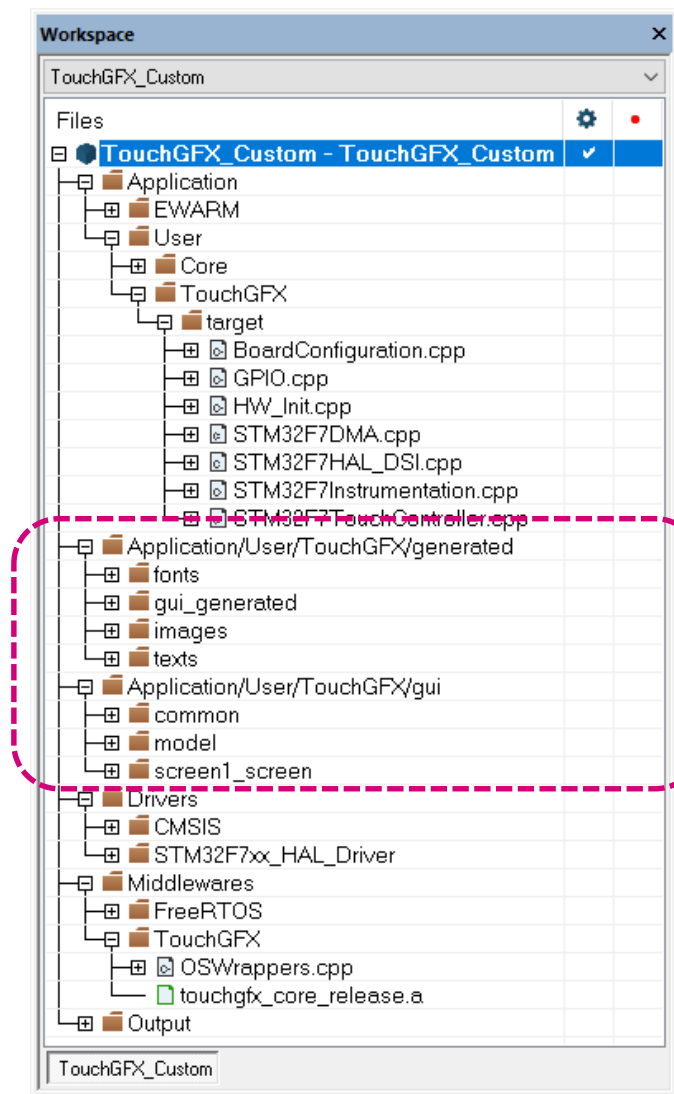


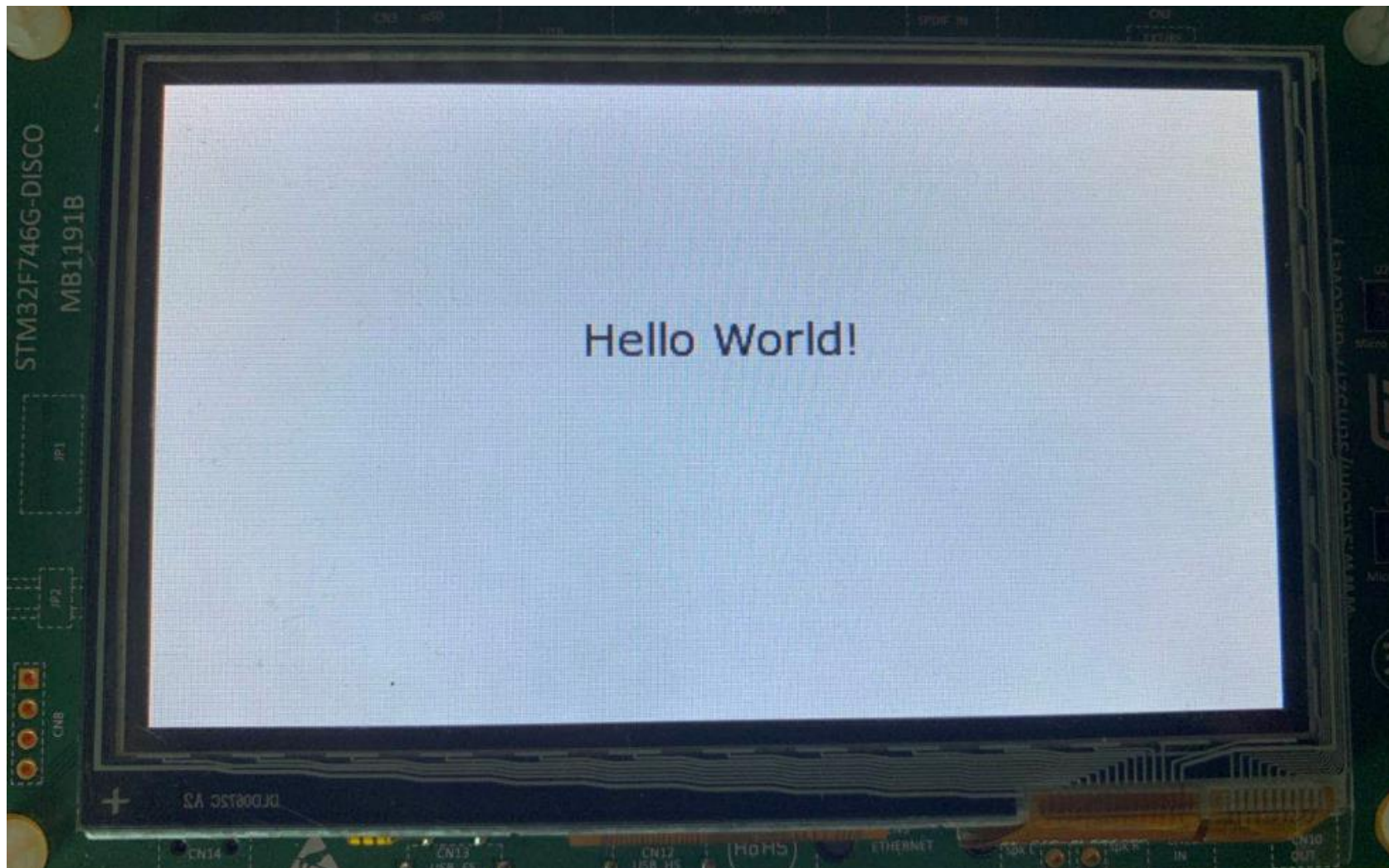
图2. IDE工程文件结构
(运行TouchGFX Designer后)



实现示例（续）

62

实验运行结果如下所示：





谢谢