



life.augmented

Quest 5 Material

STMicroelectronics Korea

2021

Prerequisite and setup

Software you need to install

- CubeIDE 1.6.1
- CubeMX 6.3.0
- TouchGFX 4.18
- CubeProgrammer 2.8.0
 - In graphic applications, it is common to use external memories to store images, texts, etc. therefore, you cannot download the binary without a dedicated bootloader. In other words, downloading with IAR will not work since IAR will only flash the internal memory. So, you need to use CubeProgrammer. To know how, see next slide.

Download binary using CubeProgrammer (1/3)

Available external loaders:

| Select | Name | Board | Start Address | Memory Size | Page Size | Type |
|-------------------------------------|--|------------------------|---------------|-------------|-----------|-----------|
| <input type="checkbox"/> | M29W128GL_STM32H7B3I-EVAL | STM32H7B3I-EVAL | 0x60000000 | 64M | 0x10 | NOR_FLASH |
| <input type="checkbox"/> | MX25LM51245G_STM32H7B0x-EVAL | STM32H7B0x-EVAL | 0x90000000 | 64M | 0x1000 | NOR_FLASH |
| <input type="checkbox"/> | MX25LM51245G_STM32H7B3I-DISCO-SFix | STM32H7B3I-DISCO-SFix | 0x90000000 | 64M | 0x1000 | NOR_FLASH |
| <input checked="" type="checkbox"/> | MX25LM51245G_STM32H7B3I_DISCO | STM32H7B3I | 0x90000000 | 64M | 0x1000 | NOR_FLASH |
| <input type="checkbox"/> | MX25LM51245G_STM32H7B3I-EVAL-REVA-SFix | STM32H7B3I-EVAL-REV... | 0x90000000 | 64M | 0x1000 | NOR_FLASH |
| <input type="checkbox"/> | MX25LM51245G_STM32H7B3I-EVAL-REVA | STM32H7B3I-EVAL-REVA | 0x90000000 | 64M | 0x1000 | NOR_FLASH |
| <input type="checkbox"/> | MX25LM51245G_STM32H7B3I-EVAL-REVB | STM32H7B3I-EVAL-REVB | 0x90000000 | 64M | 0x1000 | NOR_FLASH |
| <input type="checkbox"/> | MX25LM51245G_STM32H7B3I-EVAL-REVC | STM32H7B3I-EVAL-REVC | 0x90000000 | 64M | 0x1000 | NOR_FLASH |

Log

13:43:47 : STM32CubeProgrammer API v2.7.1

Verbosity level: 1 2 3

ST-LINK configuration

Serial number: No ST-LIN...
Port: SWD
Frequency (kHz):
Mode: Normal
Access port: 0
Reset mode: Software reset
Shared: Disabled
External loader: MX25LM51245G_STM32H7B...
Target voltage:
Firmware version:
Firmware upgrade

Target information

Board: -
Device: -
Type: -
Device ID: -
Revision ID: -
Flash size: -
CPU: -

Download binary using CubeProgrammer (2/3)

Erasing & Programming

Download

File path: C:\TouchGFXProjects\H7B3-DK_Quest_5_UART\STM32CubeIDE\Debug\STM32H7B3I_DISCO.hex **Browse**

Start addr...

Skip flash erase before programming

Verify programming

Run after programming

Start Program...

Automatic Mode

Full chip erase

Download file

Option bytes commands: -ob **Start automatic mode**

Erase flash memory | Erase external memory

Erase selected sectors | Full chip erase

| Select | Index | Start Address | Size |
|--------------------------|-------|---------------|------|
| <input type="checkbox"/> | 0 | 0x08000000 | 8K |
| <input type="checkbox"/> | 1 | 0x08002000 | 8K |
| <input type="checkbox"/> | 2 | 0x08004000 | 8K |
| <input type="checkbox"/> | 3 | 0x08006000 | 8K |
| <input type="checkbox"/> | 4 | 0x08008000 | 8K |
| <input type="checkbox"/> | 5 | 0x0800A000 | 8K |
| <input type="checkbox"/> | 6 | 0x0800C000 | 8K |
| <input type="checkbox"/> | 7 | 0x0800E000 | 8K |
| <input type="checkbox"/> | 8 | 0x08010000 | 8K |
| <input type="checkbox"/> | 9 | 0x08012000 | 8K |
| <input type="checkbox"/> | 10 | 0x08014000 | 8K |

Log

Verbosity level: 1

```
14:55:46: ST-LINK error (DEV_CONNECT_ERR)
14:55:46: ST-LINK error (DEV_CONNECT_ERR)
14:55:56: Warning: Device changed, refreshing...
14:55:56: ST-LINK error (DEV_CONNECT_ERR)
14:55:56: ST-LINK SN : 0043002A3038511234333935
14:55:56: ST-LINK FW : V3J7M3
14:55:56: Board : STM32H7B3I-DK
14:55:56: Voltage : 3.28V
14:55:56: SWD freq : 24000 KHz
14:55:56: Connect mode: Normal
14:55:56: Reset mode : Software reset
14:55:56: Device ID : 0x480
14:55:56: Revision ID : Rev Z
14:55:56: UPLOADING OPTION BYTES DATA ...
14:55:56: Bank : 0x00
14:55:56: Address : 0x5200201c
14:55:56: Size : 308 Bytes
14:55:56: UPLOADING ...
14:55:56: Size : 1024 Bytes
14:55:56: Address : 0x8000000
14:55:56: Read progress:
14:55:56: Data read successfully
14:55:56: Time elapsed during the read operation is: 00:00:00.002
```

ST-LINK configuration

Serial number: 0043002A3...
Port: SWD
Frequency (kHz): 24000
Mode: Normal
Access port: 0
Reset mode: Software reset
Shared: Disabled

External loader: MX25LM51245G_STM32H7B3...
Target voltage: 3.28 V
Firmware version: V3J7M3

Firmware upgrade

Target information

Board: STM32H7B3I-DK
Device: STM32H7A/B
Type: MCU
Device ID: 0x480
Revision ID: Rev Z
Flash size: 2 MB
CPU: Cortex-M7

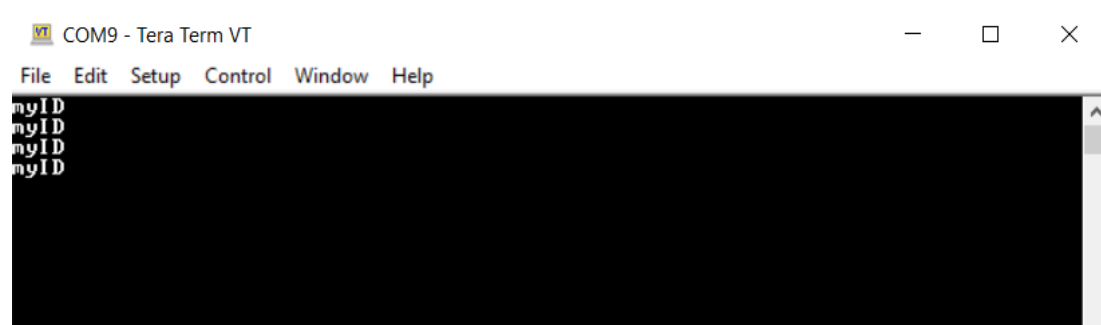
100%

life.augmented

1. Click on the browse button and select your .hex file
2. Click on Start Programming. Make sure you have "Run after programming" enabled. "Verify programming" is not necessary.

Download binary using CubeProgrammer (3/3)

- Here is what you should see on your board.

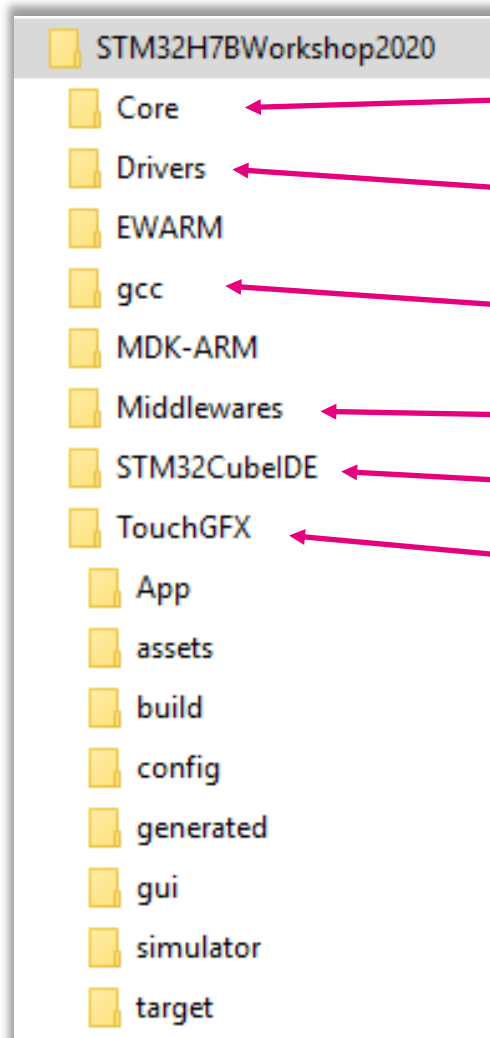


Folder architecture of a TouchGFX project



Explore generated code

In directory : <your project folder>\



Application common config
(main.c, HAL and FreeRTOS conf)

ST MCU Drivers (BSP, HAL)

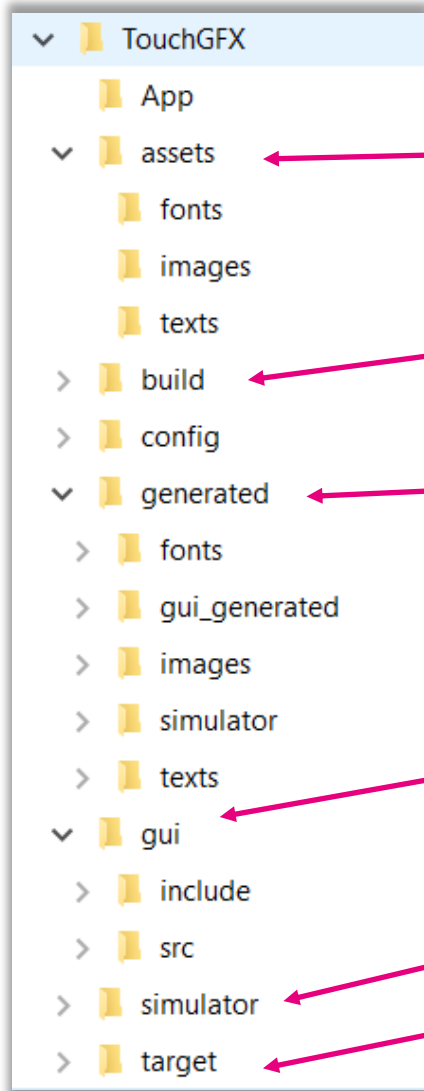
gcc toolchain files

FreeRTOS, TouchGFX framework

STM32CubeIDE configuration files

TouchGFX Designer common files

Explore generated TouchGFX directory



GUI binary inputs (images, texts, fonts)

Working directory for simulator and target compilation

All the cpp files generated from the assets directory + GUI files that should not be modified (in gui_generated)

GUI files that can be modified to implement interactions and customize

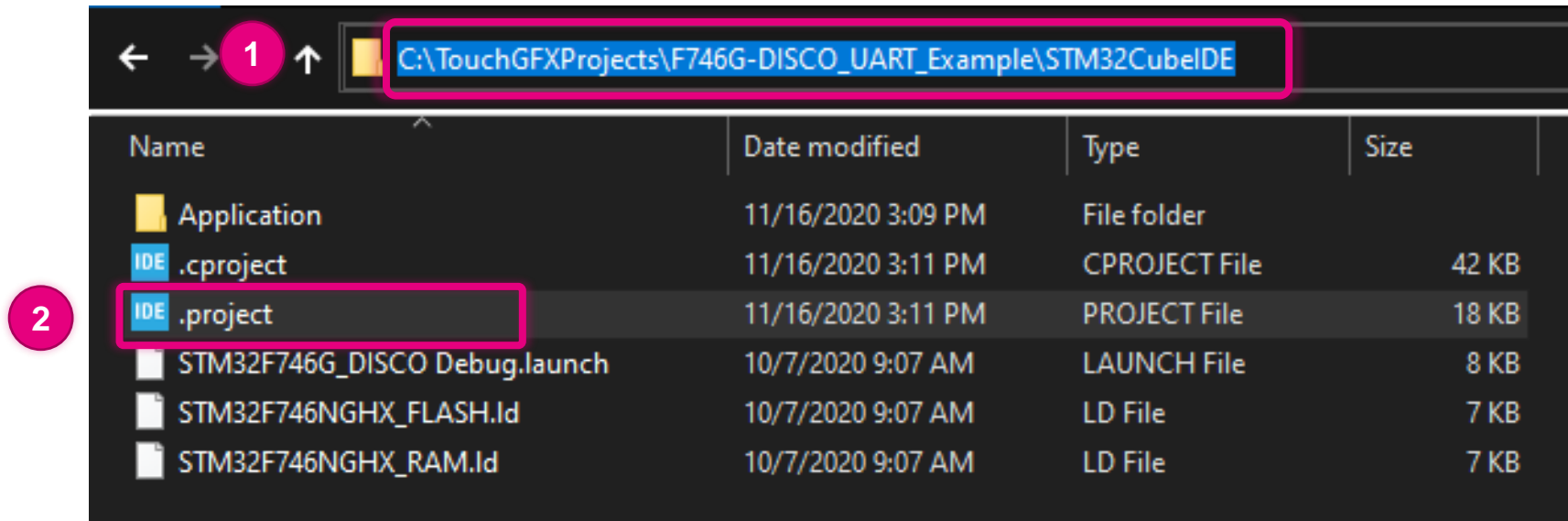
Simulator compilation scripts

Display HW initialisation (LTDC, DMA2D, ...)



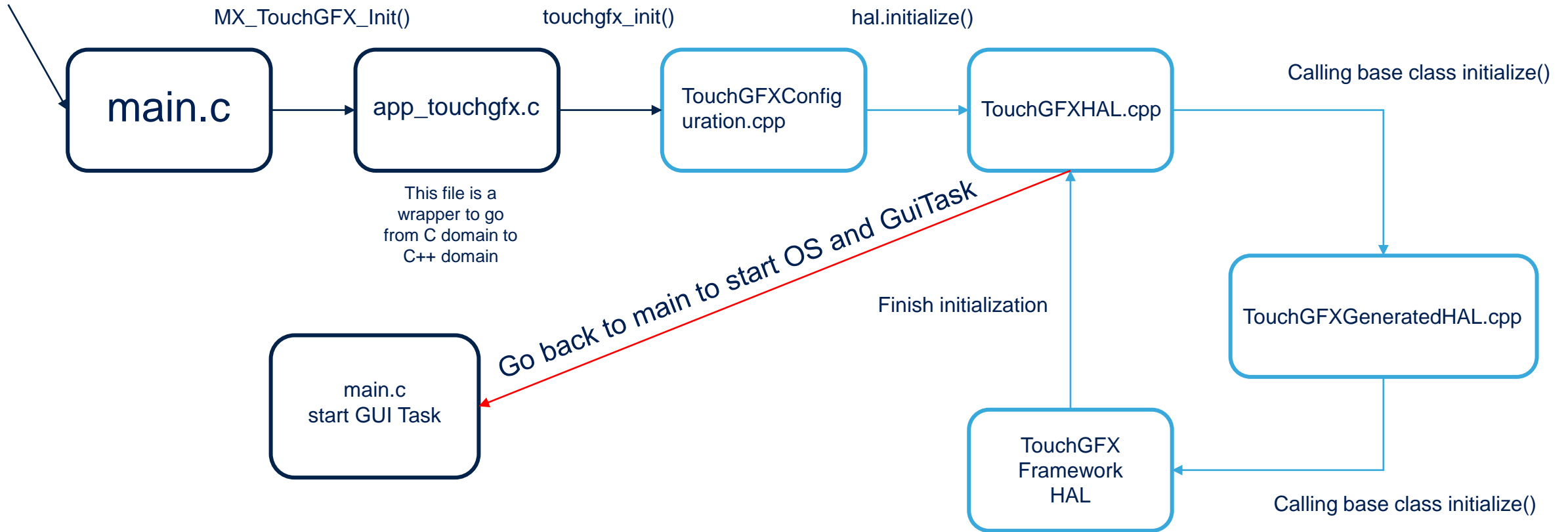
Start STM32CubeIDE

- 1 Go to <your project folder>\STM32CubeIDE\
2 Double click on **.project**, which will bring up STM32CubeIDE



TouchGFX Framework Startup Workflow

TouchGFX Framework Startup Workflow



 C++ domain

 C domain



TouchGFX Framework Startup Workflow

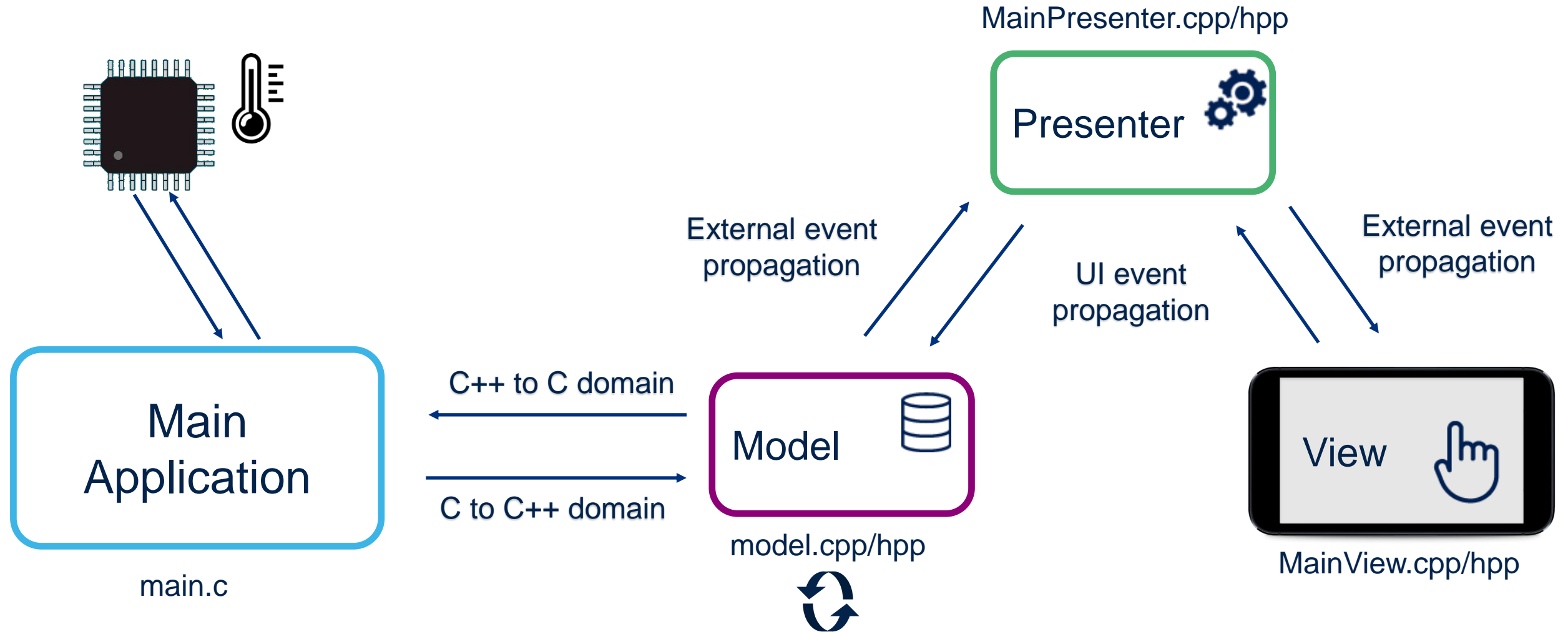
```
/* In main.c */  
void StartGuiTask(void *argument)  
{  
    /* USER CODE BEGIN 5 */  
    MX_TouchGFX_Process();  
    /* Infinite loop */  
    for(;;)  
    {  
        osDelay(1);  
    }  
    /* USER CODE END 5 */  
}
```

```
/**  
 * TouchGFX application entry function (in app_touchgfx.c)  
 */  
void MX_TouchGFX_Process(void)  
{  
    // Calling forward to touchgfx_init in C++ domain  
    touchgfx_taskEntry();  
}
```

```
/* In TouchGFXConfiguration.cpp */  
void touchgfx_taskEntry()  
{  
    /*  
     * Main event loop. Will wait for VSYNC signal, and then process next frame. Call  
     * this function from your GUI task.  
     */  
    /* Note This function never returns  
     */  
    hal.taskEntry();  
}
```

MVP system

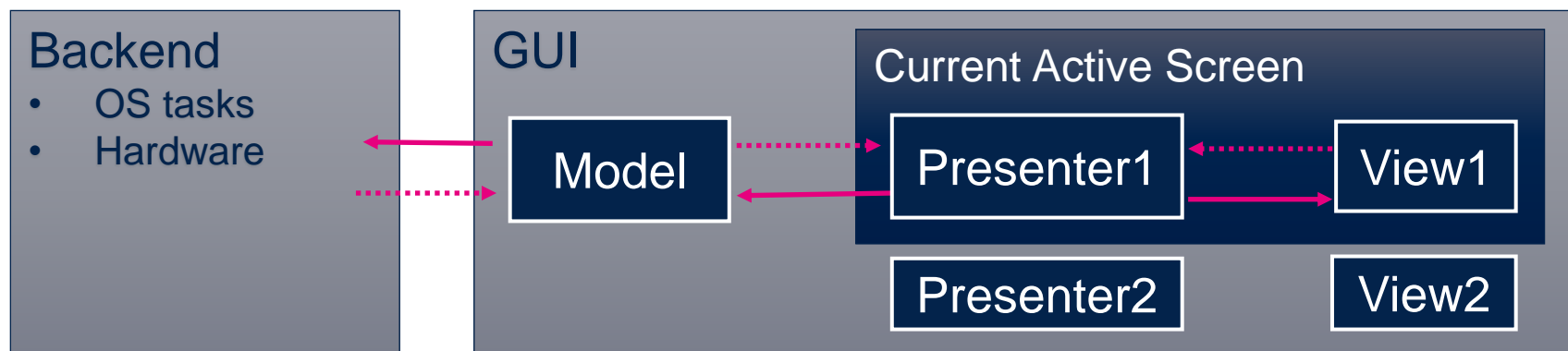
Model-View-Presenter



Model::tick() is called on each frame rendering (TGFX framework)

TouchGFX Application (1/2)

- Composed of one or several SCREENS
 - Group of widgets (VIEW)
 - Their logic regarding user/system/widgets events (PRESENTER)
 - Only one SCREEN is active at a time (for **lower memory usage**)
- In the MVP architecture:
 - Events flow from Model and View to the presenter
 - Presenter processes events and update accordingly Model and View
 - Model is the only interface with the hardware and backend application



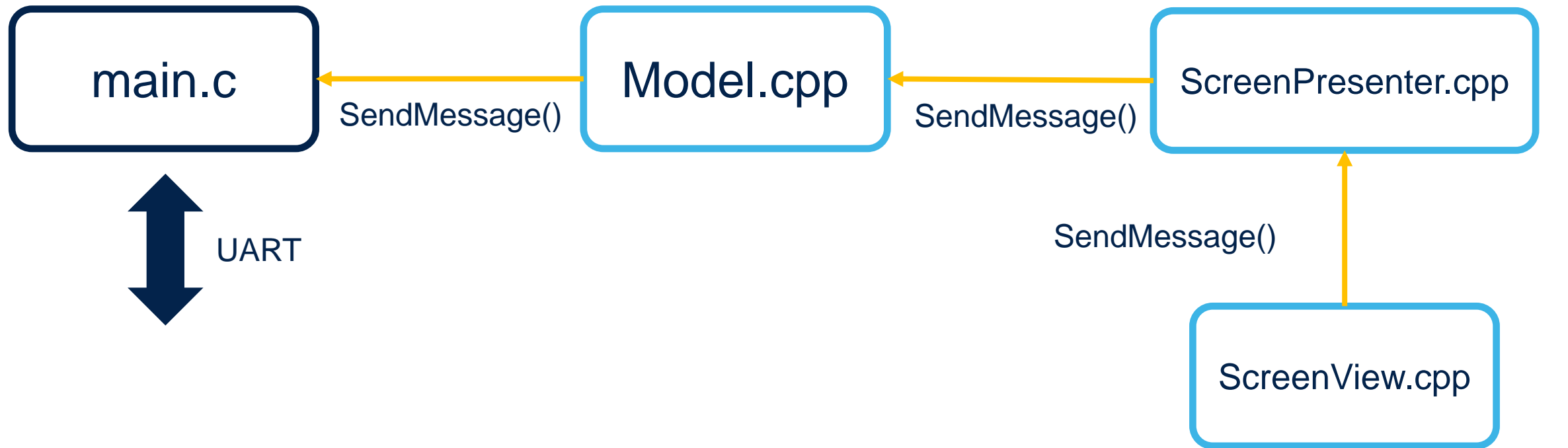
TouchGFX Application (2/2)



- Model
 - Only one in an application
 - Stores state of non-active screens
 - Interface with the hardware, relaying events to and from it
 - Has a pointer to the active presenter
- View
 - Holds and configure the widgets of one screen
 - Has a pointer to it associated presenter to communicate events
- Presenter
 - Receives events from both associated View and Model
 - Decides which action to take

UI to Backend – UART example (no OS)



UART example – Overview - UI to Backend



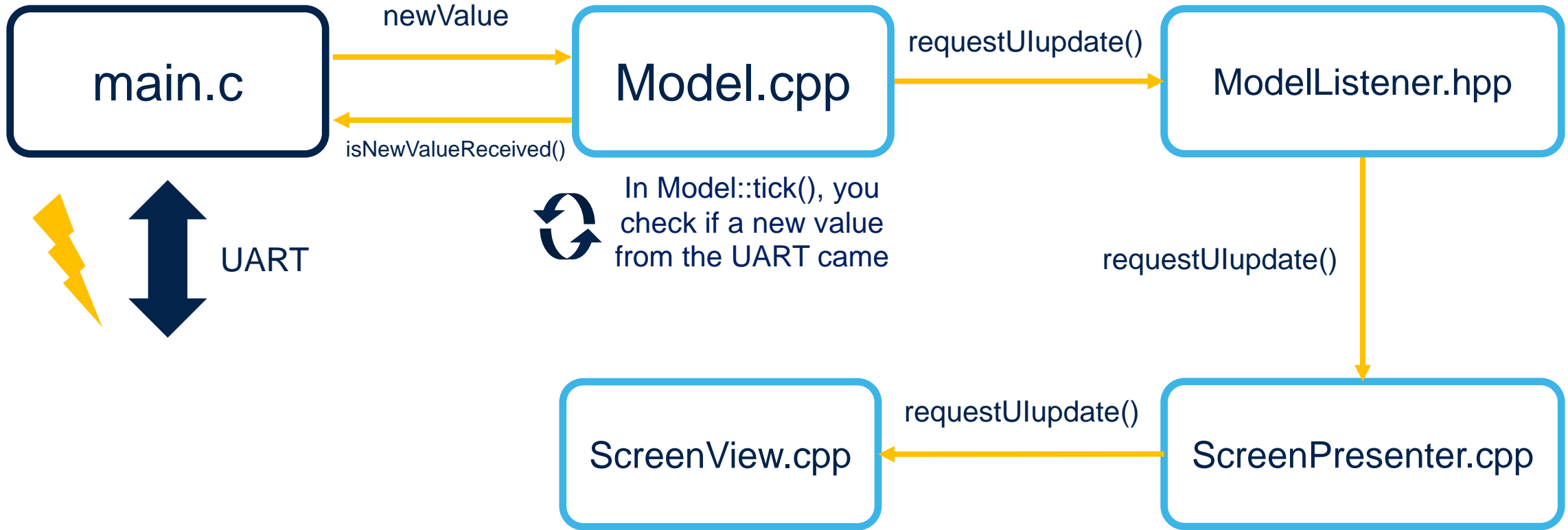
-  C++ domain
-  C domain

Event triggered by UI
e.g buttonClicked

Backend to UI (no OS)



UART example – Overview – Backend to UI

The system receives data from UART ISR



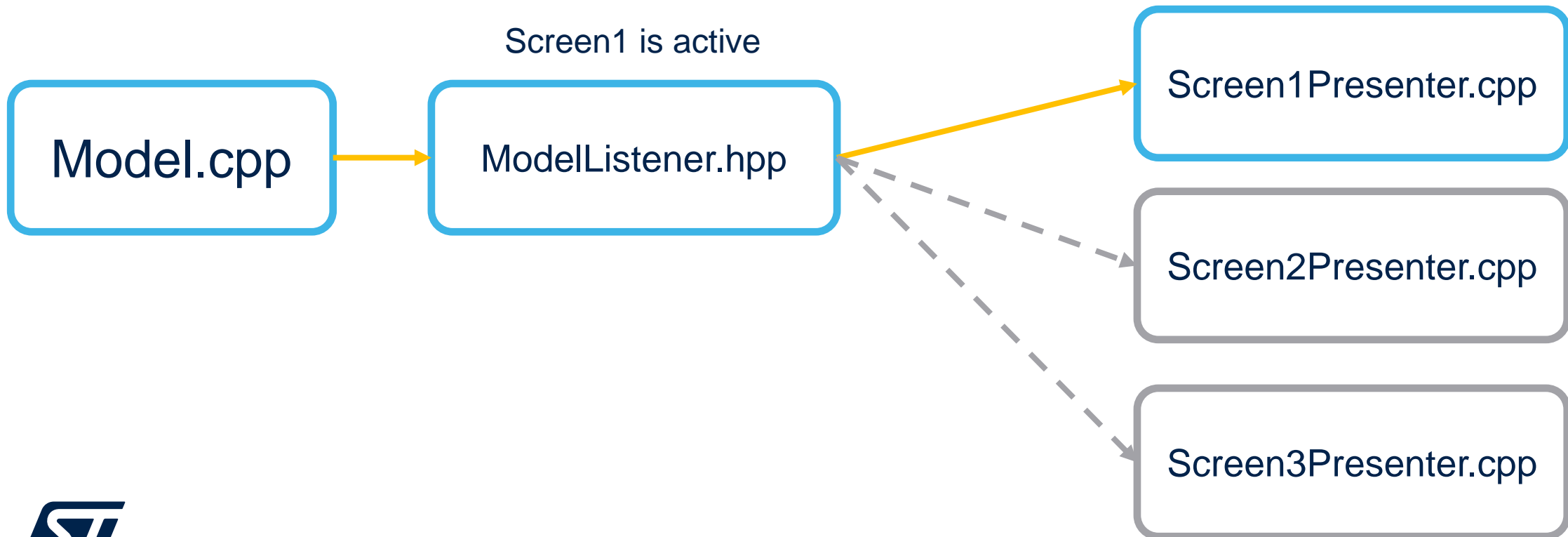
In `Model::tick()`, you check if a new value from the UART came

Update UI e.g. change TextArea according to what was received

-  C++ domain
-  C domain

What's the ModelListener ?

- The Model has a pointer to the currently active Presenter. The type of this pointer is an interface (ModelListener) which you can modify to reflect the application-specific events that are appropriate.



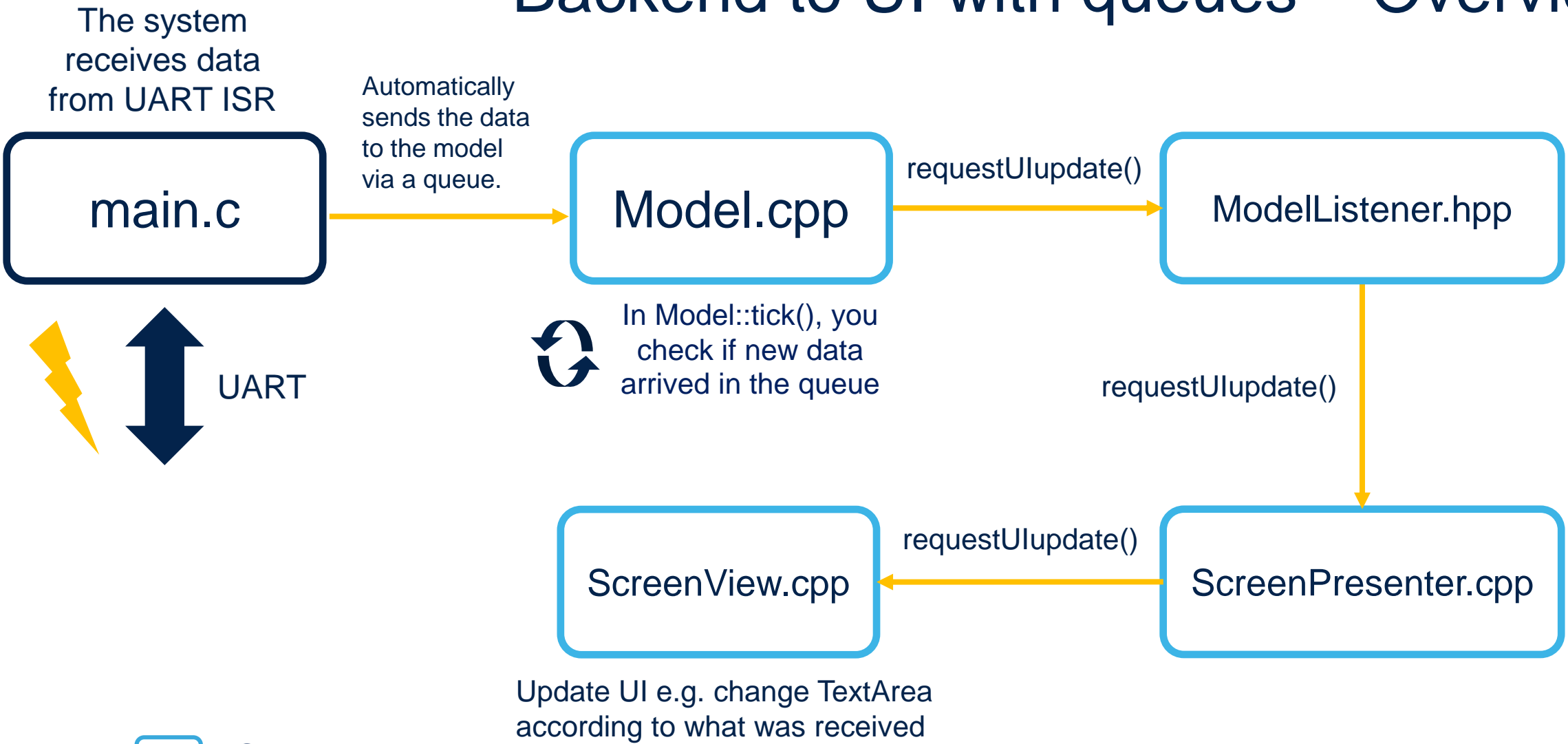
Backend to UI (with OS)





When using FreeRTOS - Queues

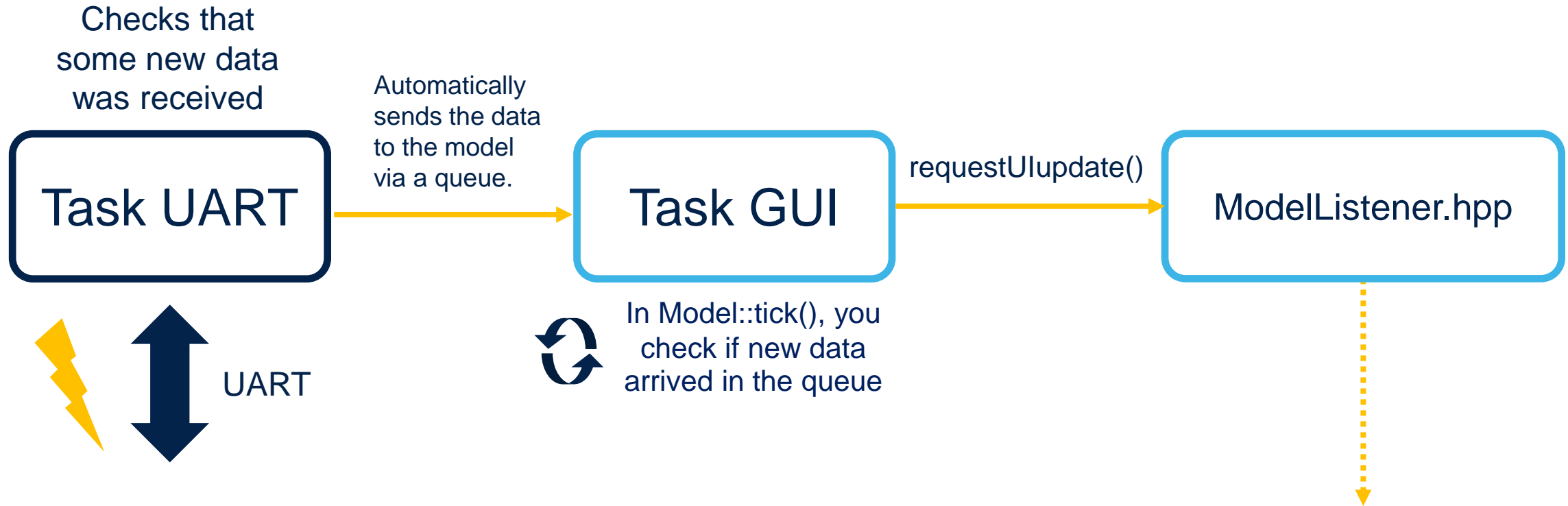
- Prerequisite : None. No need to learn extensively how FreeRTOS works.
- When using FreeRTOS, or any Embedded OS, you most likely use different tasks.
- To send information from one task to the other, you need something called a **queue**.
- Queues have 2 main benefits :
 - Provide a way to communicate between tasks.
 - A non-blocking communication system.



Backend to UI with queues – Overview



-  C++ domain
-  C domain

Queues for multi-tasks communication



-  C++ domain
-  C domain

FreeRTOS Queue API

- For using queues with FreeRTOS you only need to know the following elements.
 - A queue is declared like this :
 - `xQueueHandle myQueue;`
 - A queue is created as follows :
 - `myQueue = xQueueCreate(nbElements, sizeof(element));`
 - To add an element in a queue :
 - `xQueueSendFromISR(myQueue, &element, 0);` // When call inside an interrupt handler
 - `xQueueSendToBack(myQueue, &element, 0);` //When called from a task
 - To check if an element is in the queue :
 - `if (uxQueueMessagesWaiting(myQueue) > 0) { /* Retrieve new data */ }`
 - To take the element from the queue :
 - `xQueueReceive(myQueue, &newValue, 0);` // newValue is the new value received from the queue

Important side notes

Important side notes

- Everything done in the Designer, can also be done in User Code
 - But the Designer can help you with a lot of things
- Avoid going back and forth between User Code and Generated Code
- Remember to utilize the MVP Pattern
- Inspect Generated code
- Reuse code from examples
- The TouchGFX API: Button
- Suggestion :
 - Use the code editor Visual Studio Code.
 - Use paint.NET for image editing

Common questions answered

How to go from C to C++ domain and vice-versa ?

```
/**  
 * Declaration of a C function in a C++ file  
 */  
extern "C"  
{  
    void myFunctionInCDomain();  
}
```

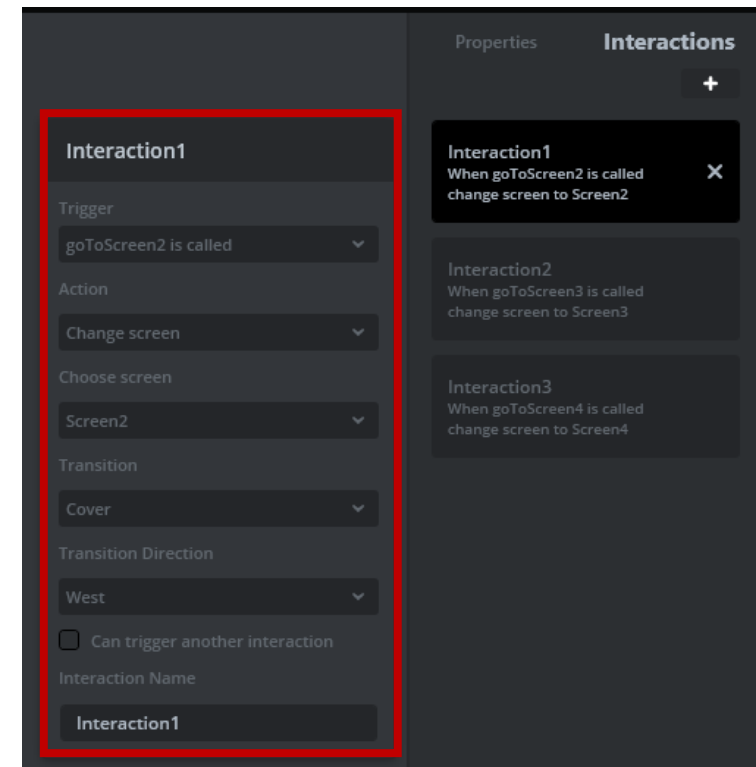
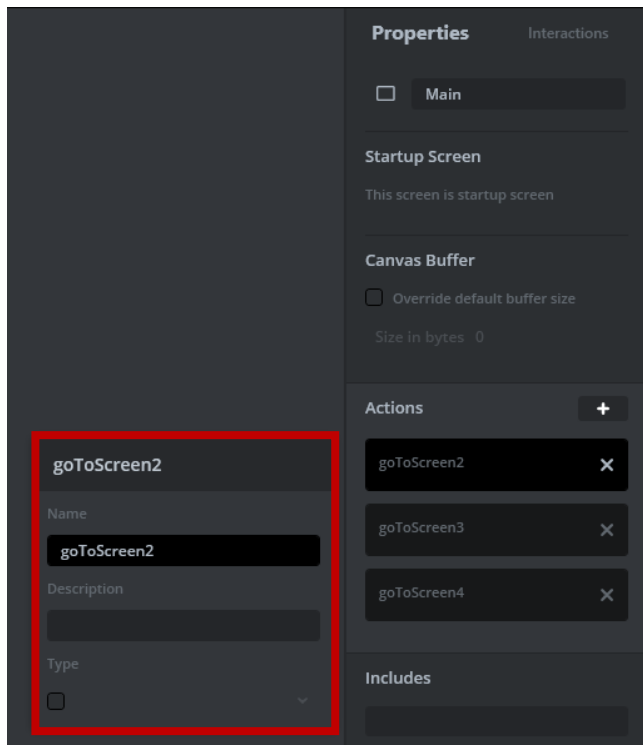
```
/* The function previously declared  
 * can be called anywhere in the file  
 */
```

```
void Model::func1()  
{  
    myFunctionInCDomain();  
}
```

```
/**  
 * Implement the function in your C file  
 */  
void myFunctionInCDomain()  
{  
    // Code executed  
}
```

How to add a screen change upon external event ?

- Steps :
 - You create a change screen action (you will call it from user code later on)
 - Then you create your interaction that performs a screen transition where the trigger is the call of your previously created action



Thank you

© STMicroelectronics - All rights reserved.

ST logo is a trademark or a registered trademark of STMicroelectronics International NV or its affiliates in the EU and/or other countries.

For additional information about ST trademarks, please refer to www.st.com/trademarks.

All other product or service names are the property of their respective owners.



life.augmented