

Embedded Systems

CS 397

TRIMESTER 3, AY 2021/22

Hands-On 1-1

STM32CubeIDE: Program GPIO and USART

Dr. LIAW Hwee Choo

Department of Electrical and Computer Engineering

DigiPen Institute of Technology Singapore

HweeChoo.Liaw@DigiPen.edu

STM32CubeIDE

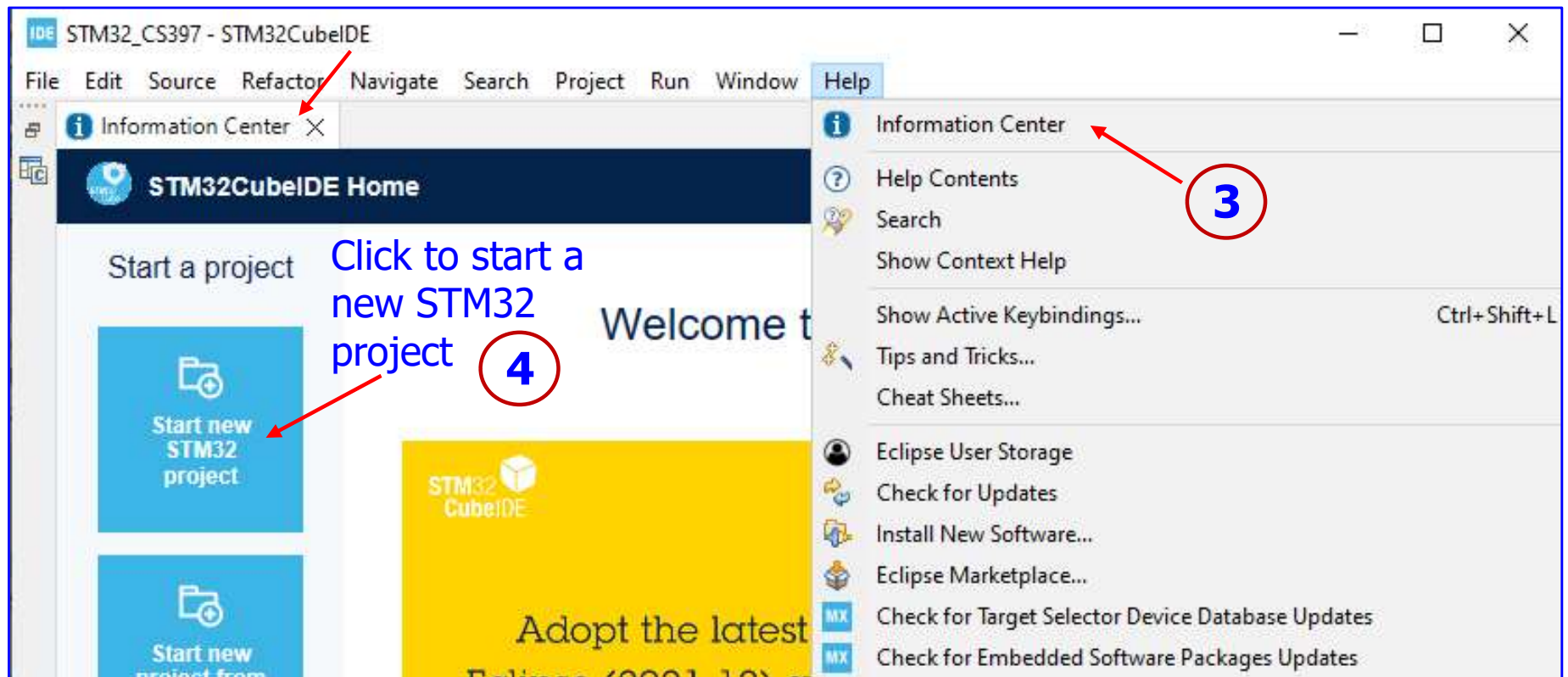
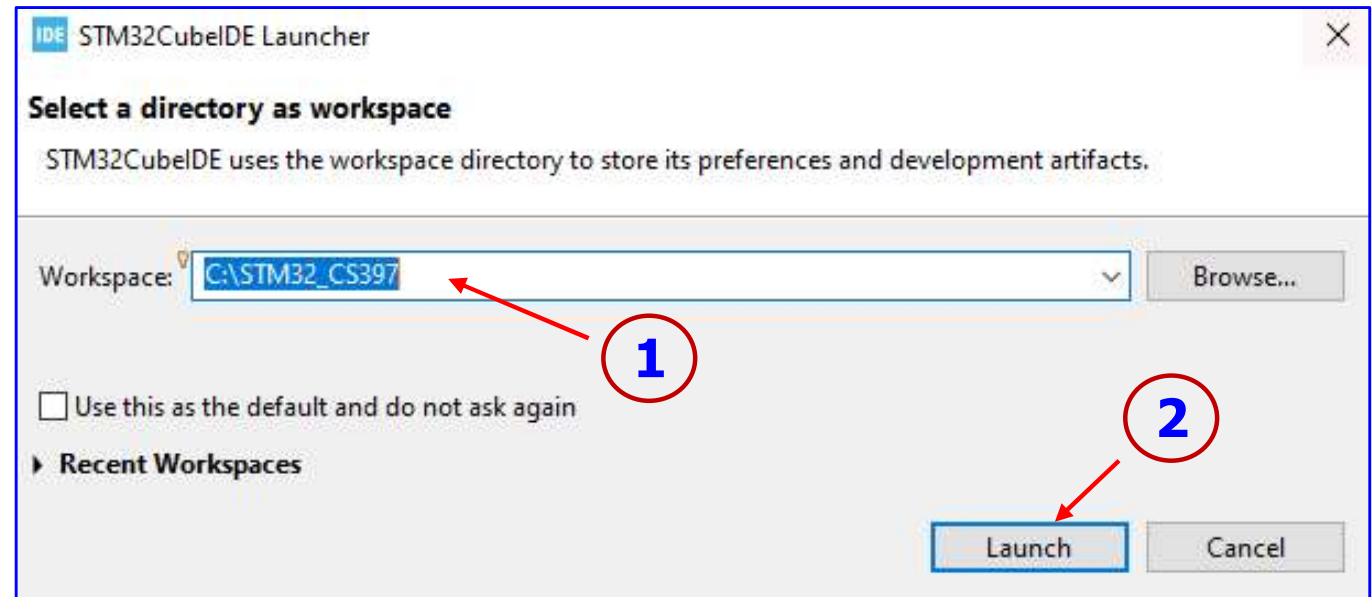
Objectives

The aims of this session are to

- create a STM32 (STM32CubeIDE) project
- understand the requirements and to set up a development system for embedded applications using the STM32F767 microcontroller
- program the GPIOs including an interrupt for a GPIO input
- program the USART channel and enable it for program debugging
- test developed programs using the “TM Terminal” or “RealTerm”, a serial terminal (COM port) software
- build-up the knowledge of embedded microcontrollers and their software development

STM32CubeIDE

Run STM32CubeIDE



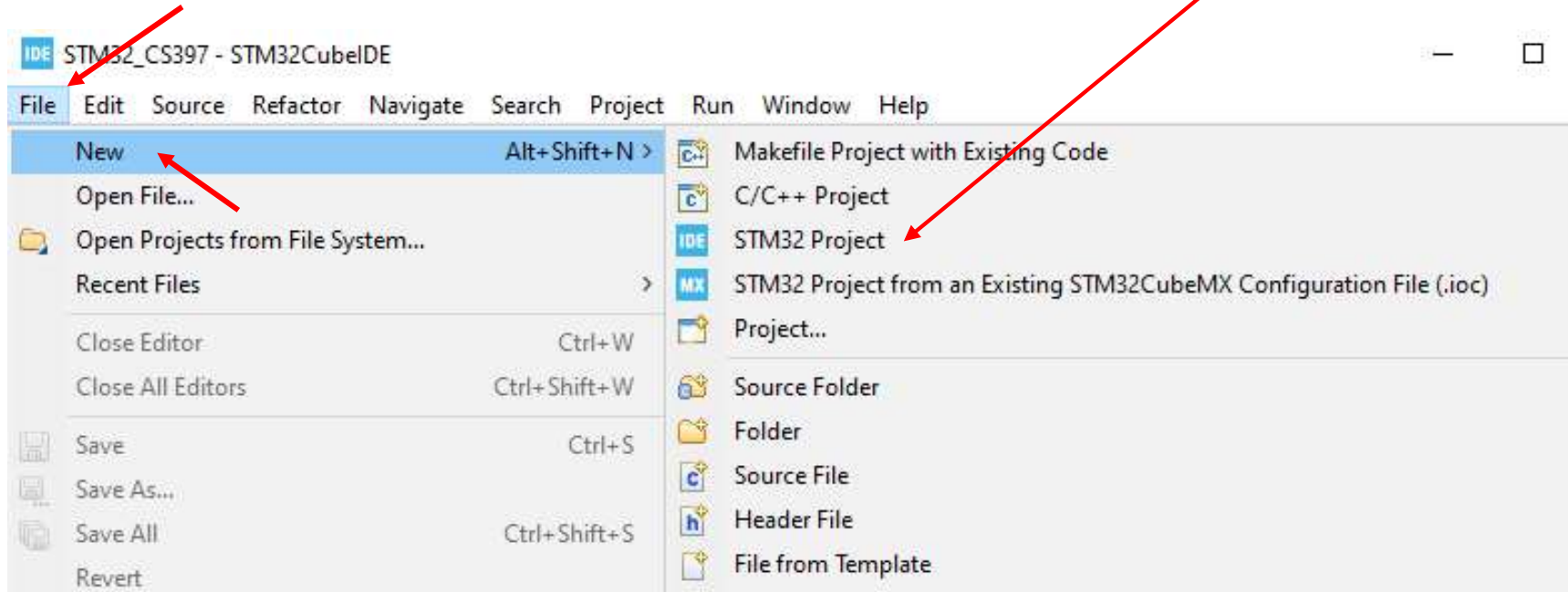
Click to start a new STM32 project

STM32CubeIDE

Or, Run [STM32CubeIDE](#)



Click to start a new STM32 project



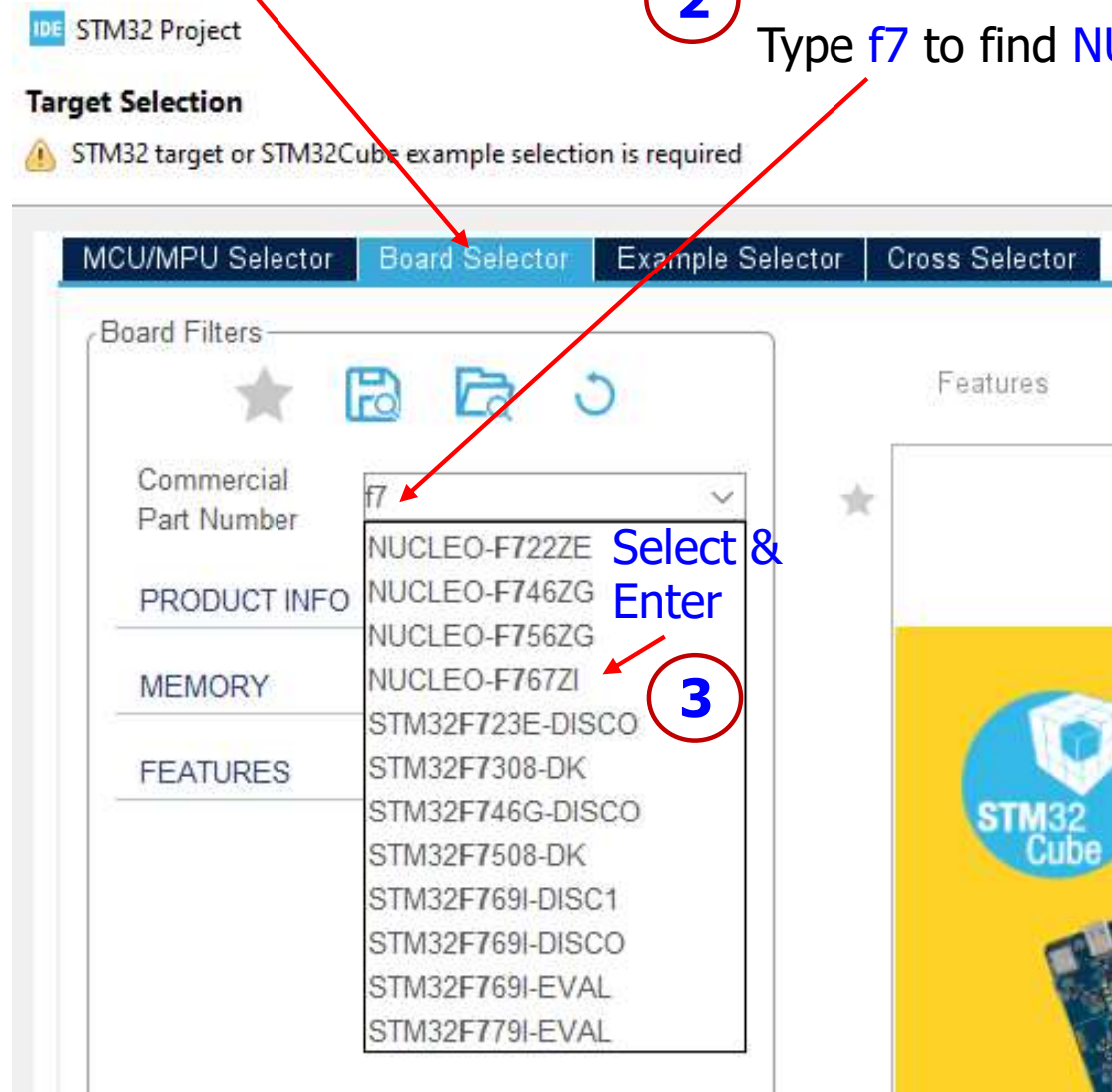
1

STM32CubeIDE

Choose Board Selector

2

Type f7 to find NUCLEO-F767ZI



Select & Enter

3

STM32CubeIDE

STM32 Project

Target Selection
Select STM32 target or STM32Cube example

MCU/MPU Selector | Board Selector | Example Selector | Cross Selector

Board Filters

Commercial Part Number: NUCLEO-F767ZI

PRODUCT INFO >
MEMORY >
FEATURES >

1

Features | Large Picture | Docs & Resources | Datasheet | Buy

STM32F7 Series

NUCLEO-F767ZI

STM32 Nucleo-144 development board with STM32F767ZI MCU, supports Arduino, ST Zio and morpho connectivity

ACTIVE
Product is in mass production

Part Number : NUCLEO-F767ZI
Commercial Part Number : NUCLEO-F767ZI

Unit Price (US\$) : 23.0
Mounted Device : [STM32F767ZIT6](#)


The STM32 Nucleo-144 board provides an affordable and flexible way for users to try out new concepts and build prototypes by choosing from the various combinations of performance and power consumption features, provided by the STM32 microcontroller. For the compatible boards, the internal or external SMPS significantly reduces power consumption in Run mode.

The ST Zio connector, which extends the ARDUINO® Uno V3 connectivity, and the ST morpho headers provide an easy means of expanding the functionality of the Nucleo open development platform with a wide choice of specialized shields.

The STM32 Nucleo-144 board does not require any separate probe as it integrates the ST-LINK debugger/programmer.

The STM32 Nucleo-144 board comes with the STM32 comprehensive free software libraries and examples available with the STM32Cube MCU Package.

Boards List: 1 item

*	Overview	Commercial Part...	Type	Marketing Status	Unit Price (US\$)	Mounted Device
☆		NUCLEO-F767ZI	Nucleo-144	Active	23.0	STM32F767ZIT6

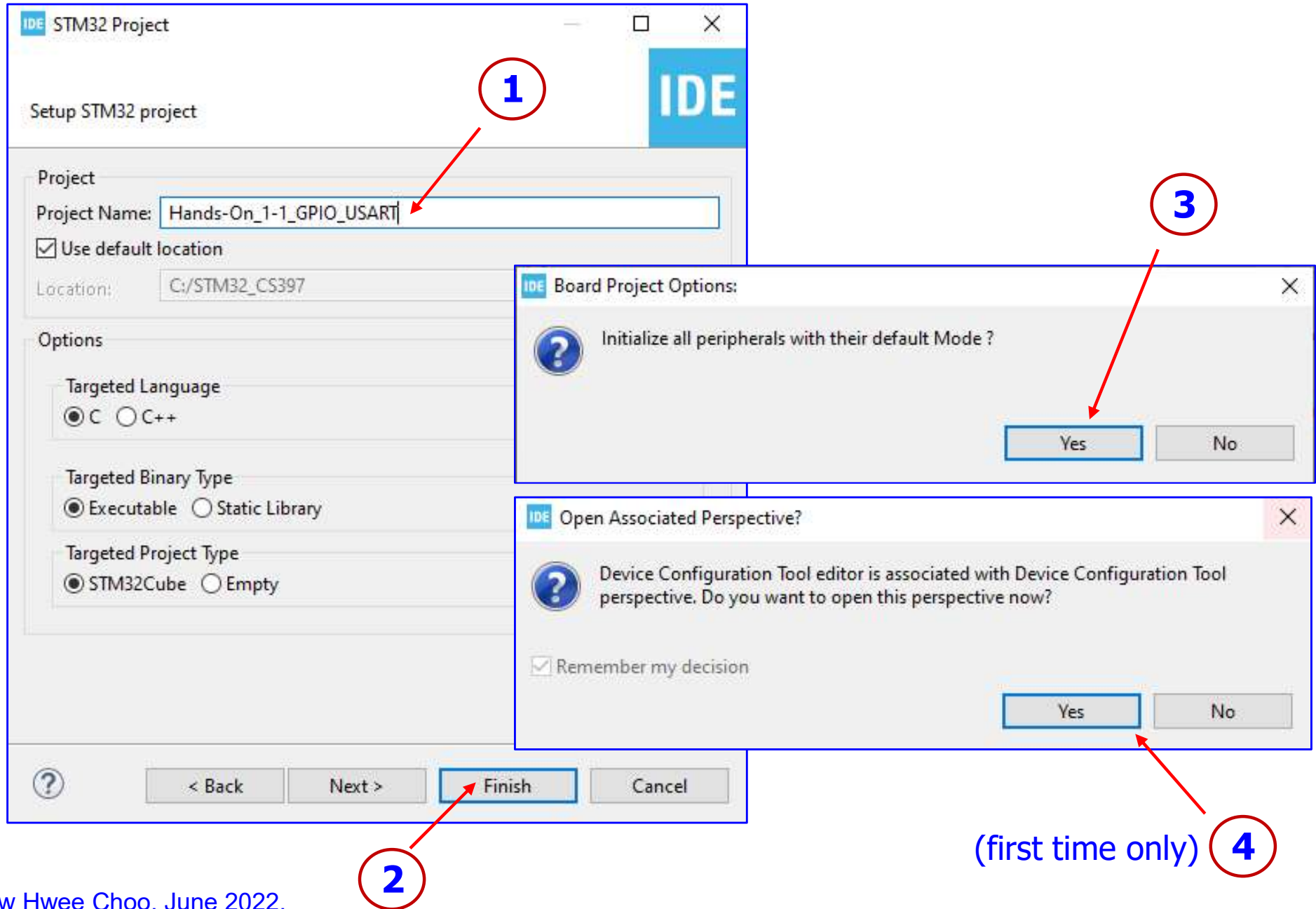
2 Select the board, and click Next

3

Next >

STM32CubeIDE

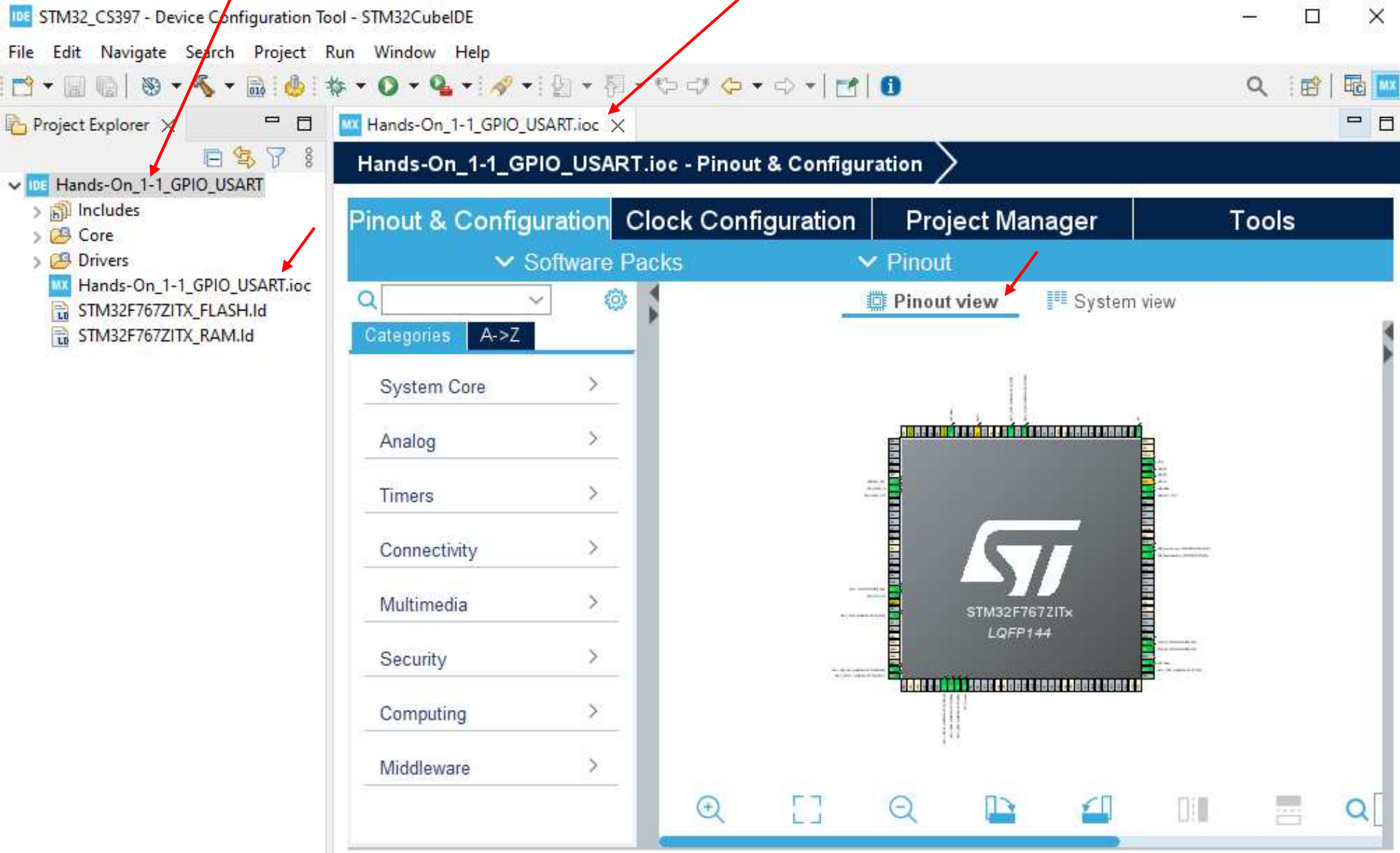
Enter Project Name: Hands-On_1-1_GPIO_USART



STM32CubeIDE

Created Project

STM32CubeMX file (.ioc)



For example, PA8 is reset by

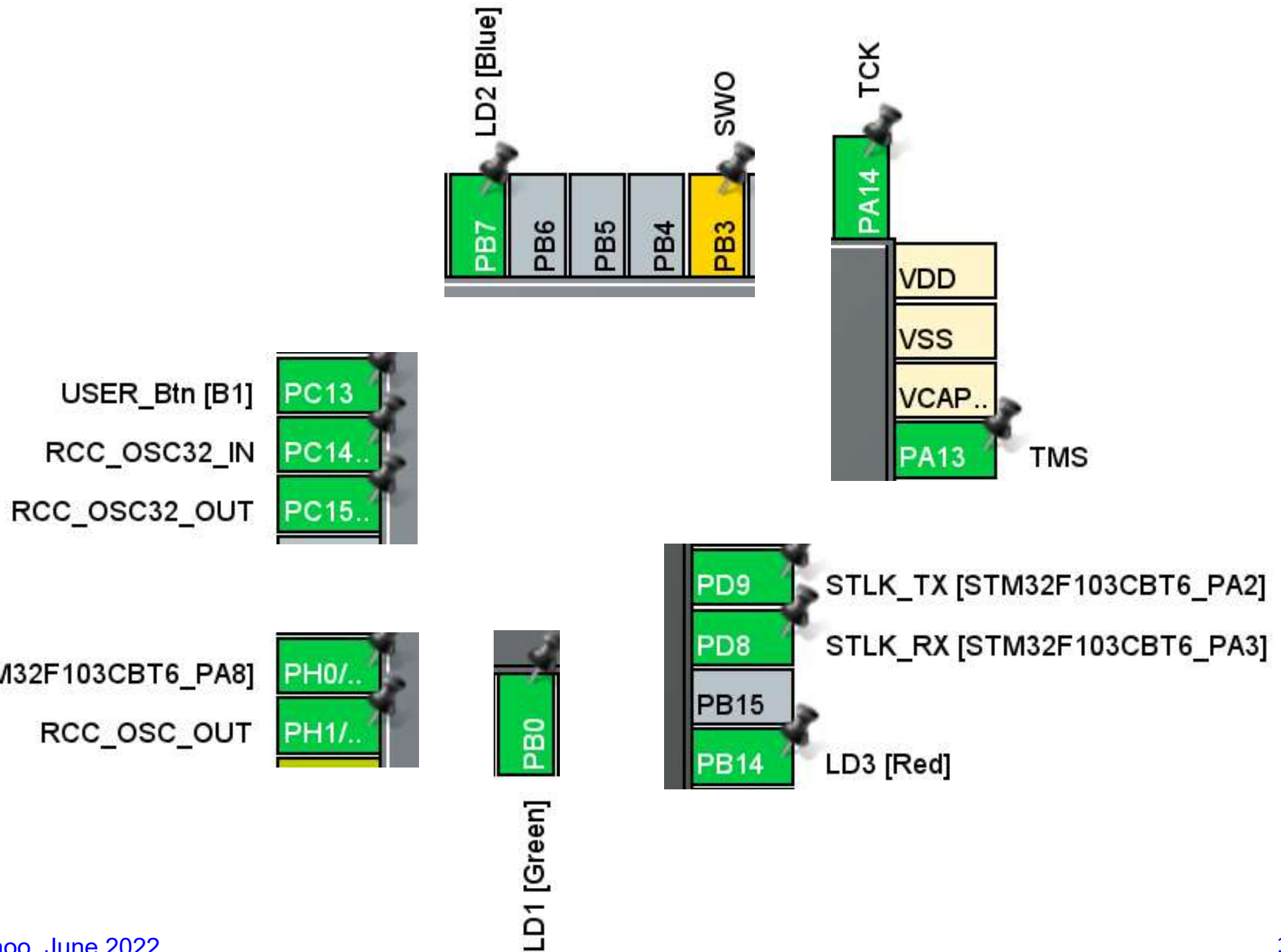
- left-click on the pin PA8, and
- select "Reset_State"



- Note that do not use the above pins for other functions/purposes.

STM32CubeIDE

Check the Configured Pins at Pinout View

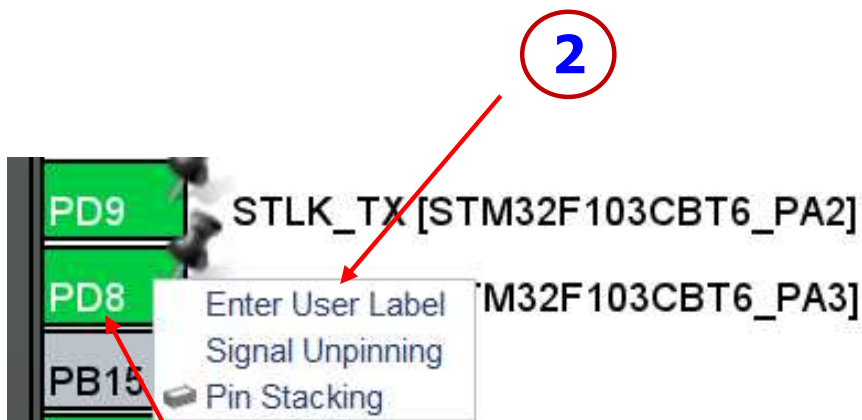


STM32CubeIDE

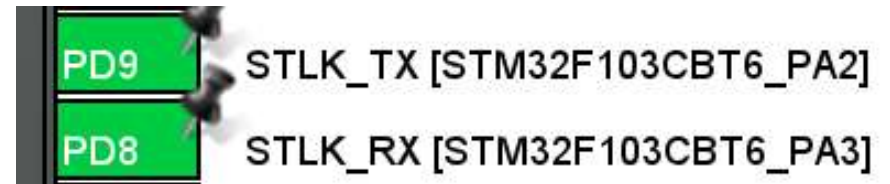
Rename PD8 and PD9 for USART3

For example, the pin PD8 can be renamed by

1. Right-click on the pin PD8, and
2. Select "Enter User Label"



PD9: DBG_USART3_RX
PD8: DBG_USART3_TX



Note: Ctrl-c and Ctrl-v can be used to cut & paste the pin names.

Hands-On_1-1_GPIO_USART.ioc - Pinout & Configuration

Pinout & Configuration | Clock Configuration | Project Manager

Software Packs | Pinout

GPIO Mode and Configuration

Mode

Configuration

Group By Peripherals

GPIO | Single Mapped Signals | RCC | SYS | USART | NVIC

Search Signals

Search (Ctrl+F)

☐ Show only Modified Pins

Pin Name	Signal on Pin	GPIO output level	GPIO mode	GPIO Pull-up/Pull-down	Maximum current	Fast Mode	User Label	Modified
PB0	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low	n/a	LD1 [Green]	<input checked="" type="checkbox"/>
PB7	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low	Disable	LD2 [Blue]	<input checked="" type="checkbox"/>
PB14	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low	n/a	LD3 [Red]	<input checked="" type="checkbox"/>
PC13	n/a	n/a	External Interrupt Mode with Rising edge trigger detection	No pull-up and no pull-down	n/a	n/a	USER_Btn [B1]	<input checked="" type="checkbox"/>

PC13 Configuration :

GPIO mode: External Interrupt Mode with Rising edge trigger detection

GPIO Pull-up/Pull-down: No pull-up and no pull-down

User Label: USER_Btn [B1]

MX *Hands-On_1-1_GPIO_USART.ioc X

Hands-On_1-1_GPIO_USART.ioc - Pinout & Configuration

Pinout & Configuration | Clock Configuration | Project Manager

Software Packs | Pinout

GPIO Mode and Configuration

Mode

Configuration

Group By Peripherals

GPIO Single Mapped Signals RCC SYS USART NVIC

Search Signals

Search (Ctrl+F)

☐ Show only Modified Pins

Pin Name	Signal on Pin	GPIO ...	GPIO mode	GPIO Pull-up/Pull-d...	Maximu...	Fast Mode	User Label	Modified
PD8	USART3_TX	n/a	Alternate Function Push Pull	No pull-up and no p...	Very High	n/a	DBG_USART3_TX	<input checked="" type="checkbox"/>
PD9	USART3_RX	n/a	Alternate Function Push Pull	No pull-up and no p...	Very High	n/a	DBG_USART3_RX	<input checked="" type="checkbox"/>

PD8 Configuration :

GPIO mode: Alternate Function Push Pull

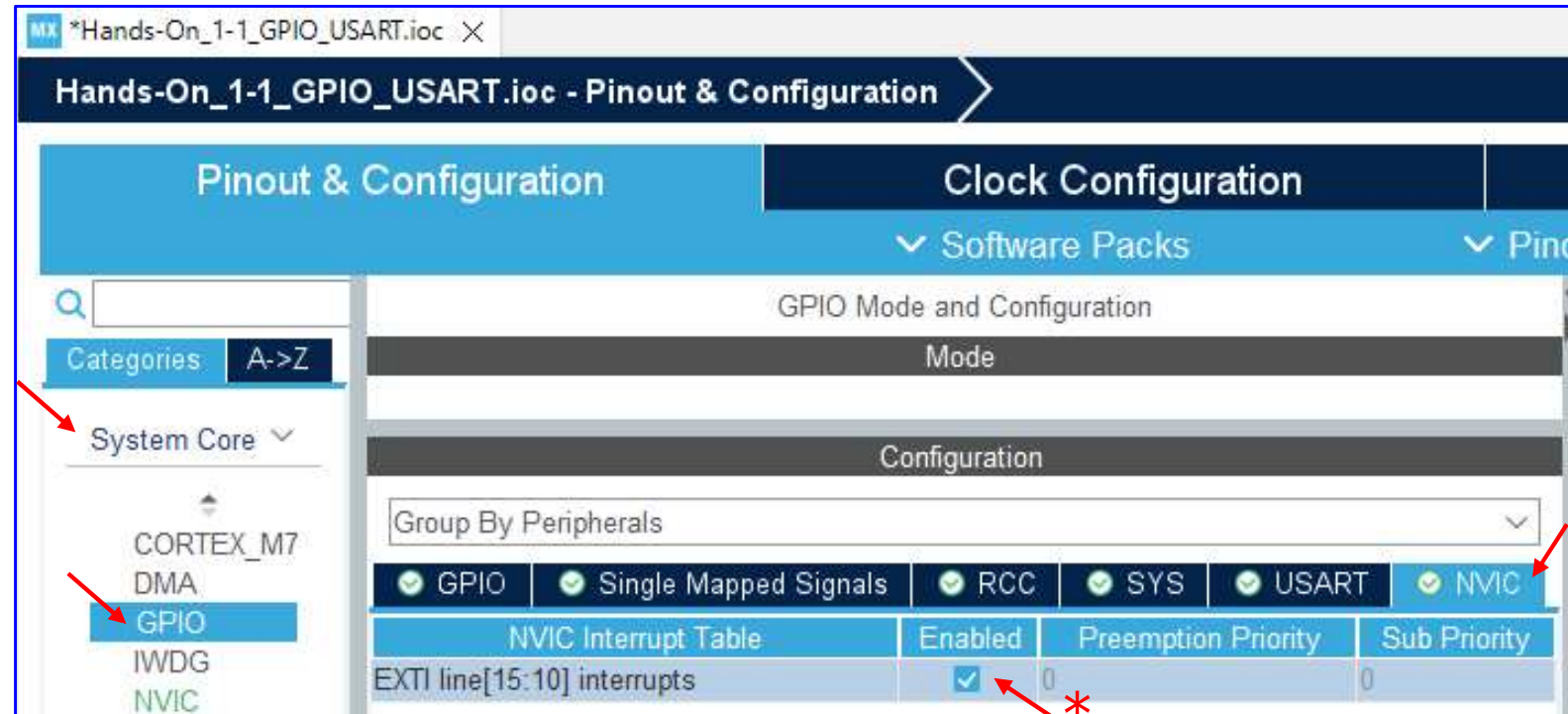
GPIO Pull-up/Pull-down: No pull-up and no pull-down

Maximum output speed: Very High

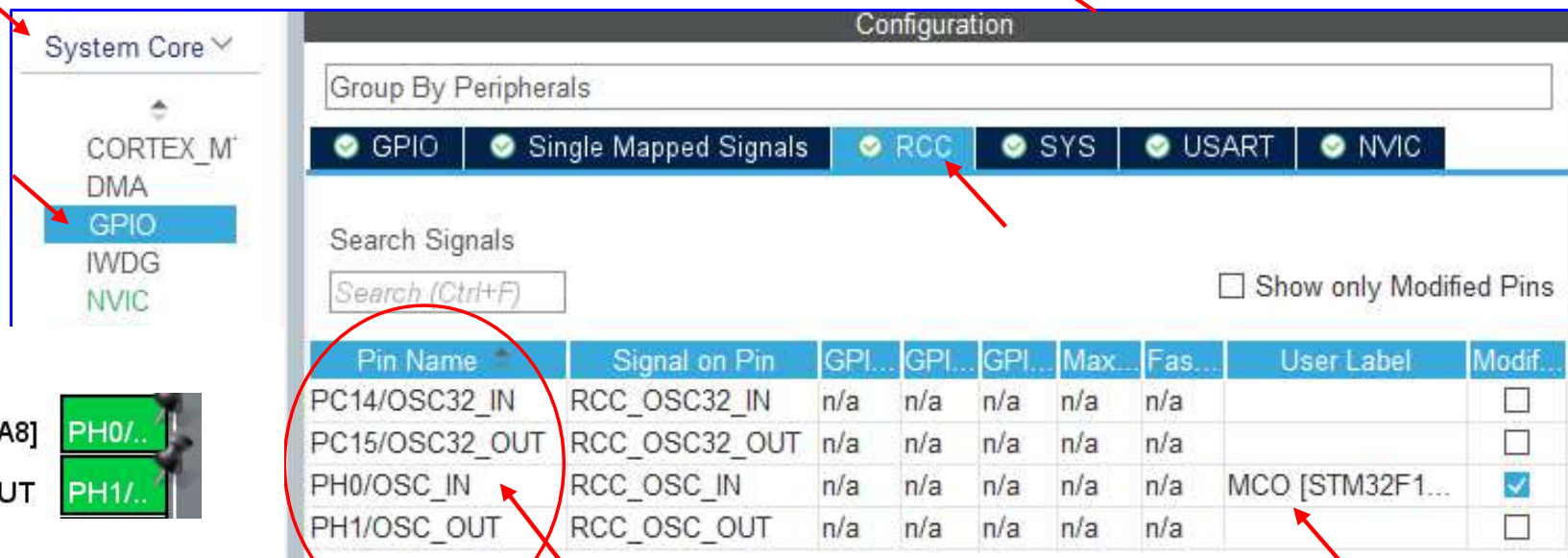
User Label: DBG_USART3_TX

STM32CubeIDE

Enable the EXTI line[15:10] interrupts for PC13, USER_Btn [B1], input



Check the RCC selection



RCC = Reset and Clock Control

STM32CubeIDE

RCC (Reset and Clock Control) Mode and Configuration

MX *Hands-On_1-1_GPIO_USART.ioc X

Hands-On_1-1_GPIO_USART.ioc - Pinout & Configuration

Pinout & Configuration | Clock Configuration

Software Packs

Categories A-Z

System Core

- CORTEX_M7
- DMA
- GPIO
- IWDG
- NVIC
- RCC**
- SYS
- WWDG

RCC Mode and Configuration

Mode

- High Speed Clock (HSE) **BYPASS Clock Source**
- Low Speed Clock (LSE) **Crystal/Ceramic Resonator**
- ☐ Master Clock Output 1
- ☐ Master Clock Output 2
- ☐ Audio Clock Input (I2S_1)

Crystal, LSE = 32.768 kHz

PC8 98 PC8
PC9 99 PC9
PC10 111 PC10
PC11 112 PC11
PC12 113 PC12
PC13 7 PC13
PC14 8 PC14
PC15 9 PC15

PC14-OSC32_IN
PC15-OSC32_OUT

USER_Btn [B1]

RCC_OSC32_IN

RCC_OSC32_OUT

MCO from where?

MCO

SB149 Closed

SB8 Open

SB9 Open

X3 NX3225GD-8.000M-EXS00A-CG04874[N/A]8C

PH0 /PF0 23
PH1 /PF1 24

PH0/PF0-OSC_IN
PH1/PF1-OSC_OUT

PC13
PC14..
PC15..

X2 NX3215SA-32.768KHZ-EXS00A-MU00525

C36 2pF
C35 2pF

R38 0
R37 0

SB173 Closed

Usr But

R59 330

B1

1 2
4 3

USER (Blue)

C52 [N/A]

R58 220K
R60 100

VDD

MCO [STM32F103CBT6_PA8]

RCC_OSC_OUT

PH0/..
PH1/..

Liaw Hwee Choo, June 2022.

SYS (System) Mode and Configuration

Hands-On_1-1_GPIO_USART.ioc - Pinout & Configuration

Pinout & Configuration

Search

Categories A-Z

System Core

- CORTEX_M7
- DMA
- GPIO
- IWDG
- NVIC
- ✓ RCC
- ⚠ SYS
- WWDG

SYS Mode and Configuration

Mode

Debug Serial Wire

- ☐ Sys Serial Wire
- ☐ Sys JTAG (4 pins)
- ☐ Sys JTAG (5 pins)
- ☐ Sys Trace Asynchronous Sw
- ☒ Sys JTAG with Trace Synchro(1 bit)
- ☐ Sys JTAG with Trace Synchro(2 bits)
- ☐ Sys JTAG with Trace Synchro(4 bits)
- ☐ Sys Trace Synchro Sw (1 bit)

Timebase

SWD (default)

SWD + SWO

SYS Mode and Configuration

Mode

Debug Trace Asynchronous Sw

☐ System Wake-Up 1

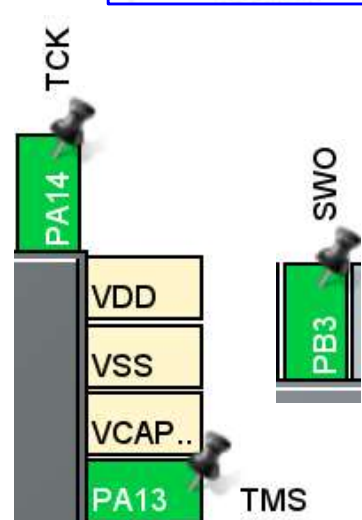
☐ System Wake-Up 2

☐ System Wake-Up 3

☒ System Wake-Up 4

Timebase Source SysTick

System Timer (SysTick)



System Core

Group By Peripherals

GPIO

GPIO

Search Signals

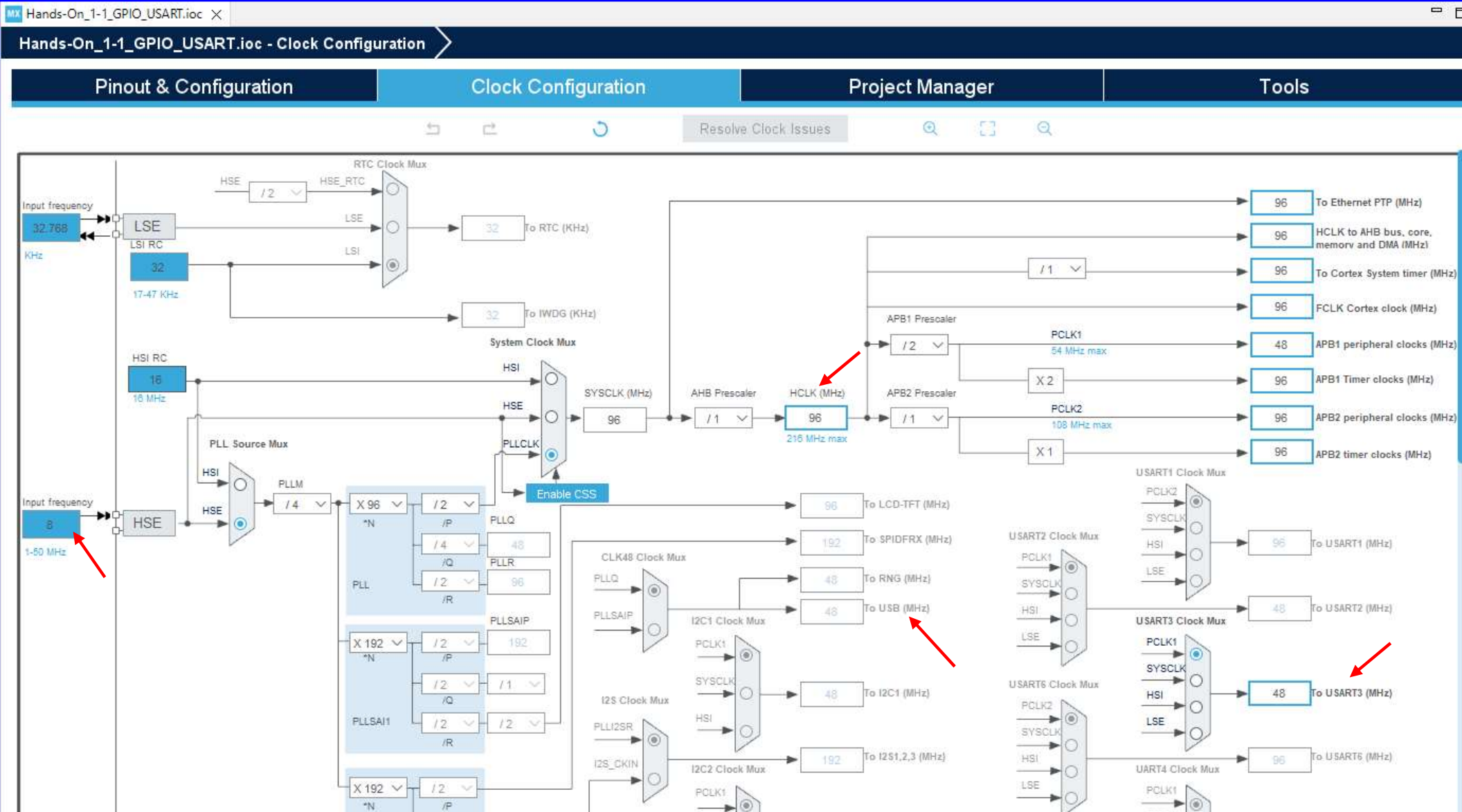
Search (Ctrl+F)

Show only Modified Pins

Pin Name	Signal on Pin	GP...	GPIO ...	GPIO ...	Maxim...	Fast M...	User L...	Modified
PA13	SYS_JTMS-SWDIO	n/a	n/a	n/a	n/a	n/a	TMS	✓
PA14	SYS_JTCK-SWCLK	n/a	n/a	n/a	n/a	n/a	TCK	✓
PB3	SYS_JTDO-SWO	n/a	n/a	n/a	n/a	n/a	SWO	✓

Note: PA13 and PA14 (SWD) are for programming.

Clock Configuration (default settings)



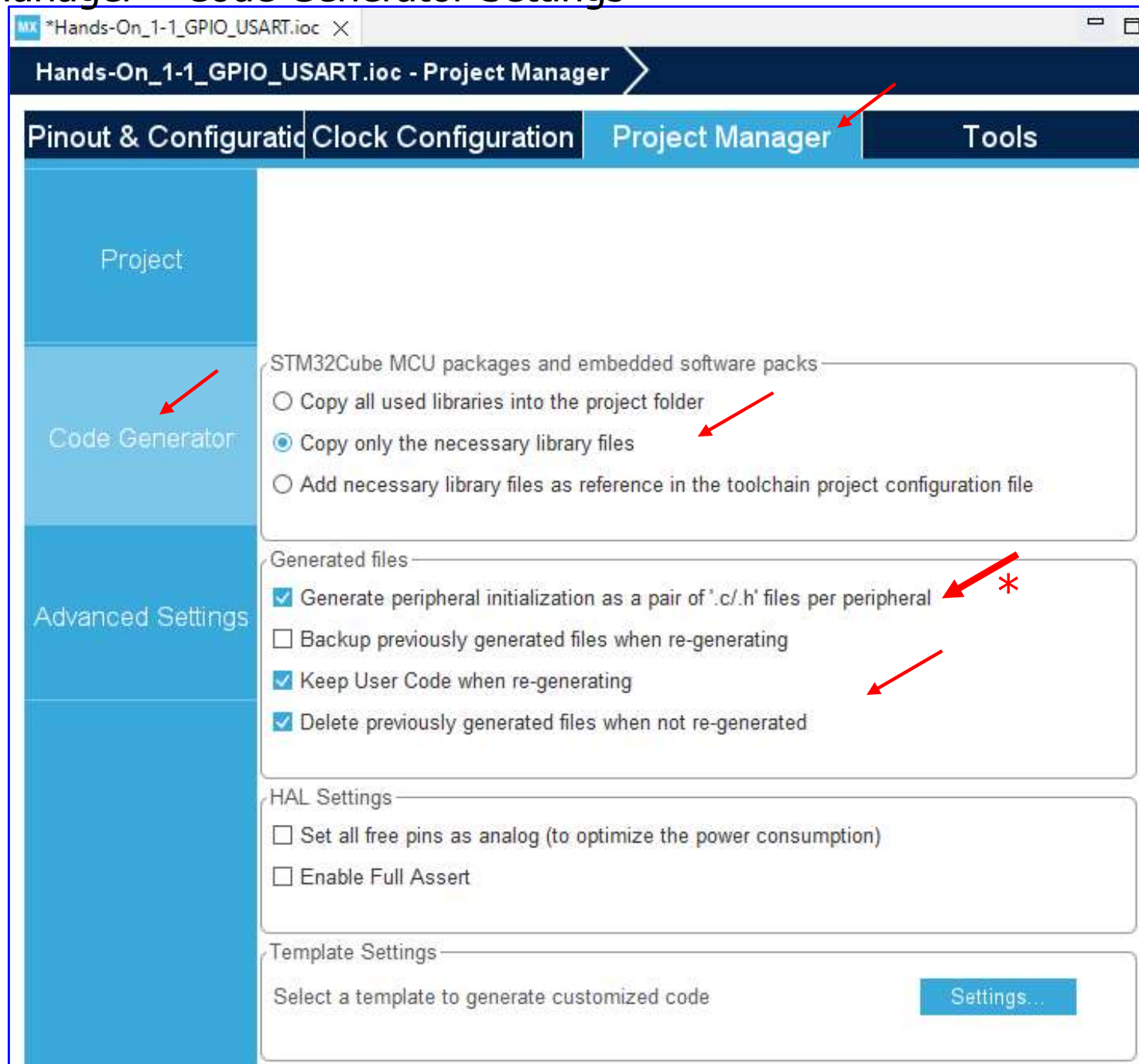
Note: Input Frequency = 8 MHz,
 SYSCCLK = HCLK (216 MHz max) = PCLK2 (108 MHz max) = 96 MHz,
 PCLK1 (54 MHz max) = USART3 Clock = 48 MHz, and
 (USB Clock = 48 MHz)

The screenshot shows the STM32CubeIDE Project Manager window for a project named "Hands-On_1-1_GPIO_USART.ioc". The window is divided into four tabs: Pinout & Configuration, Clock Configuration, Project Manager (selected), and Tools. The Project Manager tab is further divided into sections: Project, Code Generator, and Advanced Settings. The Project section contains fields for Project Name, Project Location, Application Structure, Toolchain Folder Location, and Toolchain / IDE. The Code Generator section contains fields for Linker Settings (Minimum Heap Size, Minimum Stack Size) and Thread-safe Settings (Cortex-M7NS, Enable multi-threaded support, Thread-safe Locking Strategy). The Advanced Settings section contains fields for Mcu and Firmware Package (Mcu Reference, Firmware Package Name and Version, Firmware Relative Path). Red arrows point to the following fields: Project Name, Project Location, Toolchain / IDE, Minimum Heap Size, Mcu Reference, and Firmware Package Name and Version.

Section	Field	Value
Project	Project Name	Hands-On_1-1_GPIO_USART
	Project Location	C:\STM32_CS397
	Application Structure	Advanced
	Toolchain Folder Location	C:\STM32_CS397\Hands-On_1-1_GPIO_USART\
	Toolchain / IDE	STM32CubeIDE
Code Generator	Minimum Heap Size	0x200
	Minimum Stack Size	0x400
Advanced Settings	Thread-safe Settings	Cortex-M7NS
	Enable multi-threaded support	<input type="checkbox"/>
	Thread-safe Locking Strategy	Default – Mapping suitable strategy depending on RTOS selection
	Mcu Reference	STM32F767ZITx
	Firmware Package Name and Version	STM32Cube FW_F7 V1.16.2

STM32CubeIDE

Project Manager – Code Generator Settings



STM32CubeIDE

Project Manager – Advanced Settings

MX *Hands-On_1-1_GPIO_USART.ioc X

Hands-On_1-1_GPIO_USART.ioc - Project Manager

Pinout & Configuration | Clock Configuration | **Project Manager** | Tools

Project

Code Generator

Advanced Settings

Driver Selector

Search (Ctrl+F)

RCC	HAL
GPIO	HAL
> USART	HAL
CORTEX_M7	HAL

Register CallBack

Search (...)

ADC	DISABLE
CAN	DISABLE
CEC	DISABLE
CRYP	DISABLE
DAC	DISABLE
DCMI	DISABLE
DFSDM	DISABLE
DMA2D	DISABLE
DSI	DISABLE
ETH	DISABLE
HASH	DISABLE
HCD	DISABLE
I2C	DISABLE
I2S	DISABLE
IRDA	DISABLE
JPEG	DISABLE
LPTIM	DISABLE
LTDC	DISABLE

Generated Function Calls

Generate Code	Rank	Function Name	Peripheral Insta...	Do Not Generate Function Ca	Visibility (Static)
<input checked="" type="checkbox"/>	1	SystemClock_C...	RCC	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	2	MX_GPIO_Init	GPIO	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	3	MX_USART3_U...	USART3	<input type="checkbox"/>	<input checked="" type="checkbox"/>

MX_USART3_UART_Init

STM32CubeIDE

Save Project, Generate Report, and Generate Code

The image shows the STM32CubeIDE interface with the **Project** menu open. The menu options include: Open Project, Close Project, Build All, Build Configuration, Build Project, Build Working Set, Clean..., Build Automatically, Build Targets, C/C++ Index, Generate Report, Generate Code, and Properties. Red arrows point to the **Save** and **Save All** icons in the toolbar, and to the **Generate Report** and **Generate Code** options in the Project menu.

Two confirmation dialog boxes are shown:

- Question**: Do you want generate Code? (click) **Yes** (click) "Question" after Save
- Open Associated Perspective?**: This action can be associated with C/C++ perspective. Do you want to open this perspective now? (click) **Yes** (click)

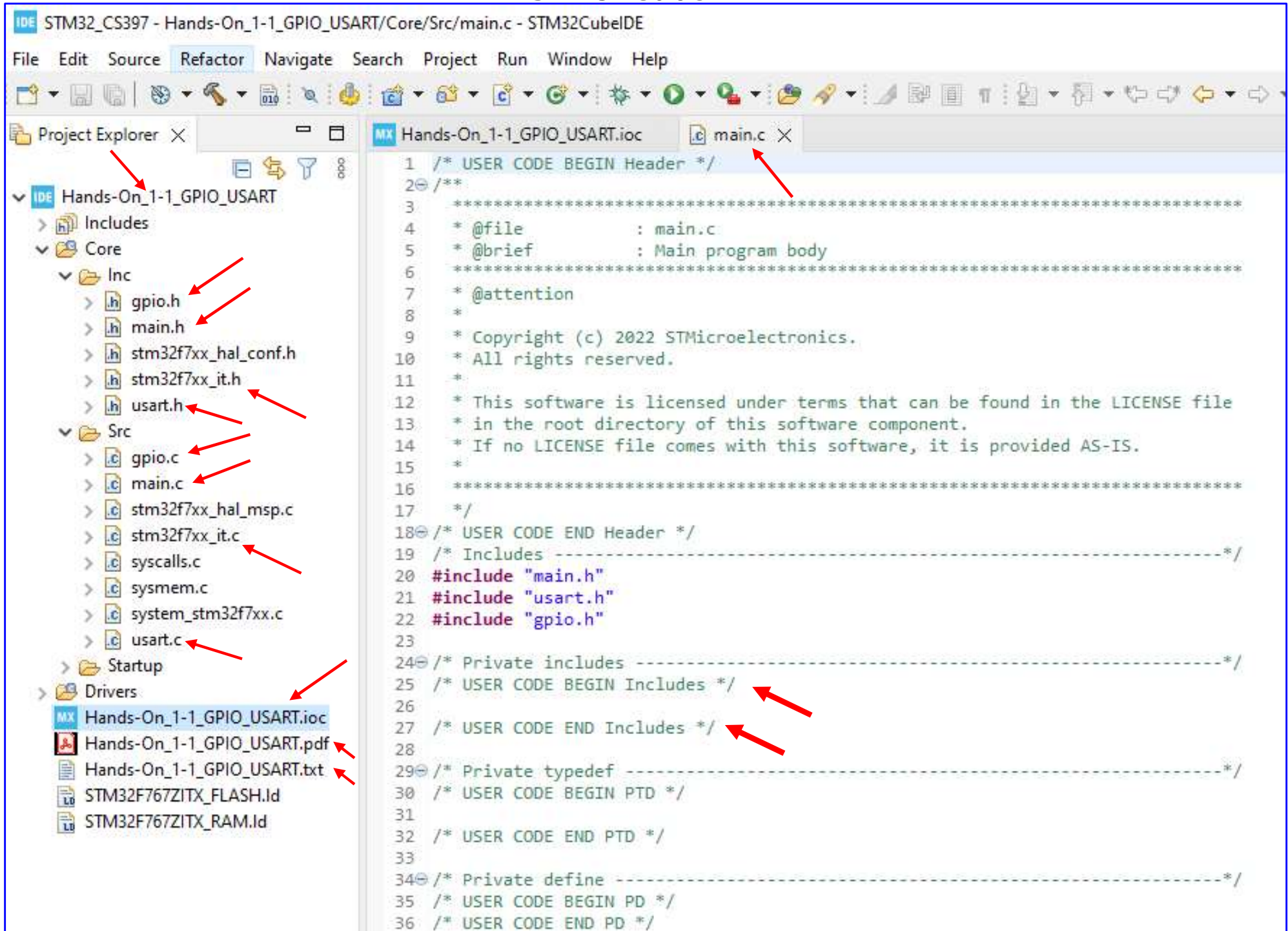
The **Generate Project Reports** window shows a message: Reports (Pdf and Text) are successfully generated under C:/STM32_CS397/Hands-On_1-1_GPIO_USART. It has **Open Folder** and **Close** buttons.

The generated files are listed below:

File Name	Time	Application	Size
Hands-On_1-1_GPIO_USART.ioc	05/09/2022 6:39 PM	STM32CubeMX	9 KB
Hands-On_1-1_GPIO_USART.pdf	05/09/2022 6:47 PM	Adobe Acrobat D...	353 KB
Hands-On_1-1_GPIO_USART.txt	05/09/2022 6:47 PM	Text Document	3 KB

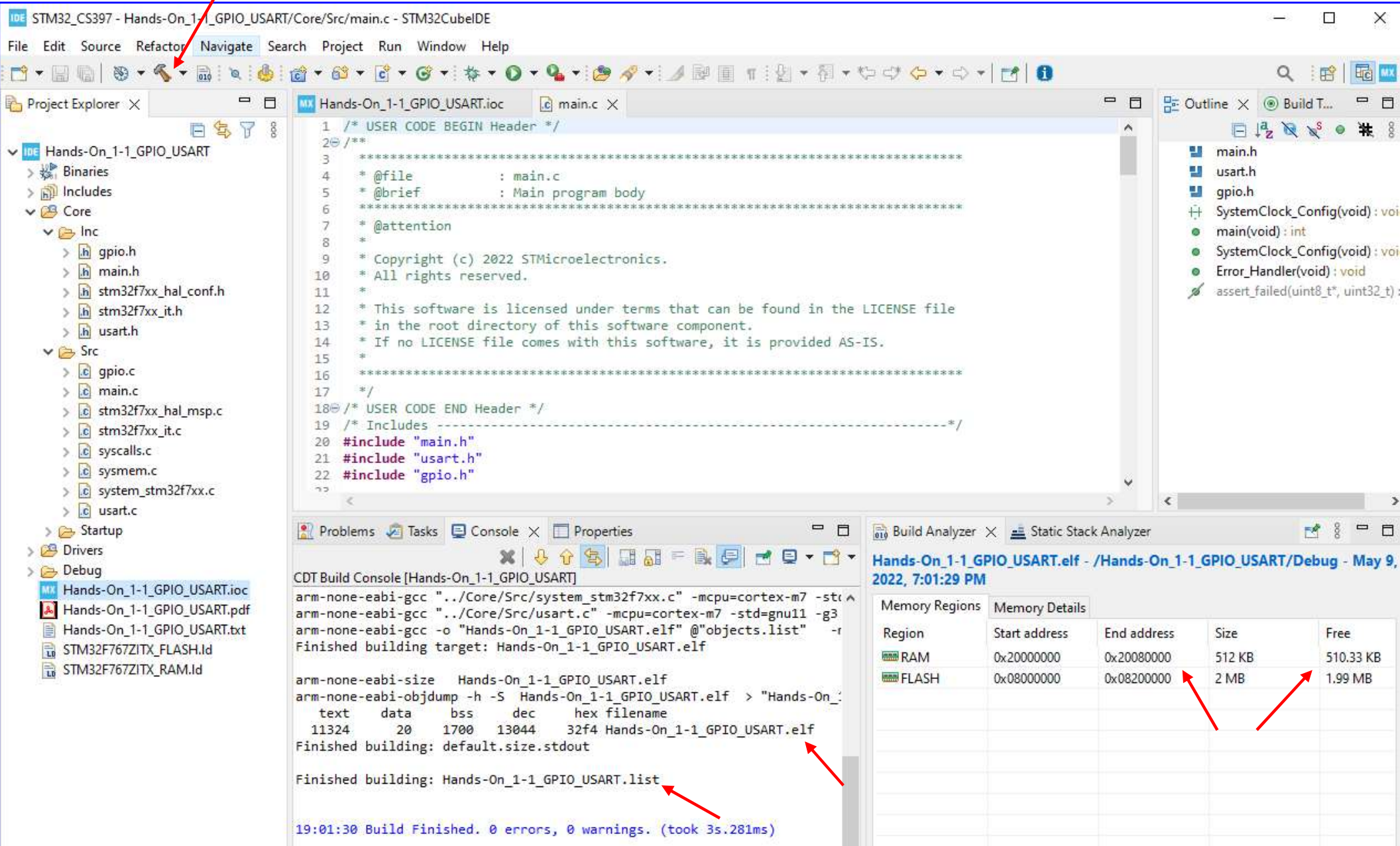
Generate Report

STM32CubeIDE



Build Project

STM32CubeIDE



STM32CubeIDE – gpio.c

```
/* gpio.c */
/* Includes */
#include "gpio.h"
/* Configure GPIO */
void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStruct = {0};
    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOC_CLK_ENABLE();
    __HAL_RCC_GPIOH_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();
    __HAL_RCC_GPIOD_CLK_ENABLE();
    __HAL_RCC_GPIOA_CLK_ENABLE();

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOB, LD1_Pin|LD3_Pin|LD2_Pin,
                      GPIO_PIN_RESET);

    /*Configure GPIO pin : PtPin */
    GPIO_InitStruct.Pin = USER_Btn_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    HAL_GPIO_Init(USER_Btn_GPIO_Port, &GPIO_InitStruct);

    /*Configure GPIO pins : PBPin PBPin PBPin */
    GPIO_InitStruct.Pin = LD1_Pin|LD3_Pin|LD2_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

    /* EXTI interrupt init*/
    HAL_NVIC_SetPriority(EXTI15_10_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);
}
```

```
/* gpio.h */
/* Define to prevent recursive
inclusion */
#ifndef __GPIO_H__
#define __GPIO_H__

#ifdef __cplusplus
    extern "C" {
#endif

#include "main.h"

void MX_GPIO_Init(void);

#ifdef __cplusplus
}
#endif

#endif /*__GPIO_H__ */
```

Hover mouse pointer on this function to find its detail.

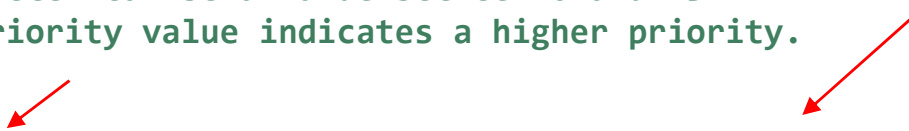
STM32CubeIDE – gpio.c & gpio.h

```
/**
 * @brief Sets the priority of an interrupt.
 * @param IRQn External interrupt number.
 *         This parameter can be an enumerator of IRQn_Type enumeration
 *         (For the complete STM32 Devices IRQ Channels list, please refer to the
 *         appropriate CMSIS device file (stm32f7xxxx.h))
 * @param PreemptPriority The preemption priority for the IRQn channel.
 *         This parameter can be a value between 0 and 15
 *         A lower priority value indicates a higher priority
 * @param SubPriority the subpriority level for the IRQ channel.
 *         This parameter can be a value between 0 and 15
 *         A lower priority value indicates a higher priority.
 * @retval None
 */
void HAL_NVIC_SetPriority(IRQn_Type IRQn, uint32_t PreemptPriority, uint32_t SubPriority)
{
    uint32_t prioritygroup = 0x00;

    /* Check the parameters */
    assert_param(IS_NVIC_SUB_PRIORITY(SubPriority));
    assert_param(IS_NVIC_PREEMPTION_PRIORITY(PreemptPriority));

    prioritygroup = NVIC_GetPriorityGrouping();

    NVIC_SetPriority(IRQn, NVIC_EncodePriority(prioritygroup, PreemptPriority, SubPriority));
}
```



STM32CubeIDE
– USART3
(default settings)

Pinout & Configuration

Clock Configuration

Project Manager

Software PacksPinout

USART3 Mode and Configuration

Mode

ModeAsynchronous

Hardware Flow Control (RS232)Disable

☐ Hardware Flow Control (RS485)

Configuration

Reset Configuration

NVIC Settings

DMA Settings

GPIO Settings

Parameter Settings

User Constants

Configure the below parameters :

Search (Ctrl+F)

Basic Parameters

Baud Rate115200 Bits/s

Word Length8 Bits (including Parity)

ParityNone

Stop Bits1

Advanced Parameters

Data DirectionReceive and Transmit

Over Sampling16 Samples

Single SampleDisable

Advanced Features

Auto BaudrateDisable

TX Pin Active Level InversionDisable

RX Pin Active Level InversionDisable

Data InversionDisable

TX and RX Pins SwappingDisable

OverrunEnable

DMA on RX ErrorEnable

MSB FirstDisable

CategoriesA->Z

Connectivity

CAN1

CAN2

CAN3

ETH

FMC

I2C1

I2C2

I2C3

I2C4

MDIOS

QUADSPI

SDMMC1

SDMMC2

PI1

PI2

PI3

PI4

SPI5

SPI6

UART4

UART5

UART7

UART8

USART1

USART2

USART3

USART6

USB_OTG_FS

USB_OTG_HS

PD9

DBG_USART3_RX

PD8

DBG_USART3_TX

STM32CubeIDE – USART3

Pinout & Configuration | Clock Configuration | Project Manager | Tools

Software Packs | Pinout

USART3 Mode and Configuration

Mode

Mode: Asynchronous

Hardware Flow Control (RS232): Disable

☐ Hardware Flow Control (RS485)

Configuration

Reset Configuration

Parameter Settings | User Constants | NVIC Settings | DMA 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 ou...	Fast Mode	User Label	Modified
PD8	USART3_TX	n/a	Alternate Func...	No pull-up a...	Very High	n/a	DBG_USART3_TX	<input checked="" type="checkbox"/>
PD9	USART3_RX	n/a	Alternate Func...	No pull-up a...	Very High	n/a	DBG_USART3_RX	<input checked="" type="checkbox"/>

PD9 DBG_USART3_RX

PD8 DBG_USART3_TX

STM32CubeIDE – usart.c

```
/* usart.c */

#include "usart.h"

UART_HandleTypeDef huart3;

/* USART3 init function */
void MX_USART3_UART_Init(void)
{
    huart3.Instance = USART3;
    huart3.Init.BaudRate = 115200;
    huart3.Init.WordLength = UART_WORDLENGTH_8B;
    huart3.Init.StopBits = UART_STOPBITS_1;
    huart3.Init.Parity = UART_PARITY_NONE;
    huart3.Init.Mode = UART_MODE_TX_RX;
    huart3.Init.HwFlowCtl = UART_HWCONTROL_NONE;
    huart3.Init.OverSampling = UART_OVERSAMPLING_16;
    huart3.Init.OneBitSampling = UART_ONE_BIT_SAMPLE_DISABLE;
    huart3.AdvancedInit.AdvFeatureInit = UART_ADVFEATURE_NO_INIT;
    if (HAL_UART_Init(&huart3) != HAL_OK)
    {
        Error_Handler();
    }
}
```

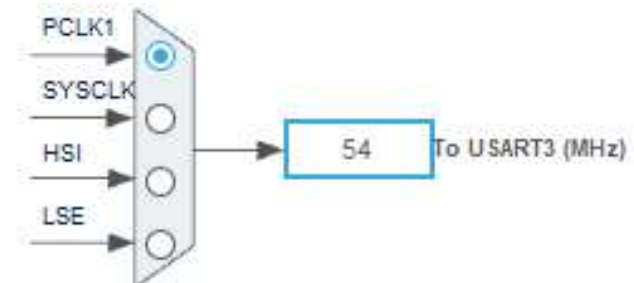

STM32CubeIDE – usart.c & usart.h

```

void HAL_UART_MspInit(UART_HandleTypeDef* uartHandle)
{
    GPIO_InitTypeDef GPIO_InitStruct = {0};
    RCC_PeriphCLKInitTypeDef PeriphClkInitStruct = {0};
    if(uartHandle->Instance==USART3)
    {
        /* Initializes the peripherals clock */
        PeriphClkInitStruct.PeriphClockSelection = RCC_PERIPHCLK_USART3;
        PeriphClkInitStruct.Usart3ClockSelection = RCC_USART3CLKSOURCE_PCLK1;
        if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInitStruct) != HAL_OK)
        {
            Error_Handler();
        }
        /* USART3 clock enable */
        __HAL_RCC_USART3_CLK_ENABLE();
        __HAL_RCC_GPIOD_CLK_ENABLE();
        GPIO_InitStruct.Pin = DBG_USART3_TX_Pin|DBG_USART3_RX_Pin;
        GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
        GPIO_InitStruct.Pull = GPIO_NOPULL;
        GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
        GPIO_InitStruct.Alternate = GPIO_AF7_USART3;
        HAL_GPIO_Init(GPIOD, &GPIO_InitStruct);
    }
}

void HAL_UART_MspDeInit(UART_HandleTypeDef* uartHandle)
{
    if(uartHandle->Instance==USART3)
    {
        /* Peripheral clock disable */
        __HAL_RCC_USART3_CLK_DISABLE();
        HAL_GPIO_DeInit(GPIOD, DBG_USART3_TX_Pin|DBG_USART3_RX_Pin);
    }
}

```



```

/* usart.h */
/* Define to prevent recursive
inclusion */
#ifndef __USART_H__
#define __USART_H__

#ifdef __cplusplus
extern "C" {
#endif

#include "main.h"

extern UART_HandleTypeDef huart3;

void MX_USART3_UART_Init(void);

#ifdef __cplusplus
}
#endif

#endif /* __USART_H__ */

```

STM32CubeIDE – main.h

```
/* main.h */
/* Define to prevent recursive inclusion */
#ifndef __MAIN_H
#define __MAIN_H

#ifdef __cplusplus
extern "C" {
#endif

/* Includes */
#include "stm32f7xx_hal.h"

/* Exported functions prototypes */
void Error_Handler(void);

/* Private defines */
#define USER_Btn_Pin GPIO_PIN_13
#define USER_Btn_GPIO_Port GPIOC
#define USER_Btn_EXTI_IRQn EXTI15_10_IRQn
#define MCO_Pin GPIO_PIN_0
#define MCO_GPIO_Port GPIOH
#define LD1_Pin GPIO_PIN_0
#define LD1_GPIO_Port GPIOB
#define LD3_Pin GPIO_PIN_14
#define LD3_GPIO_Port GPIOB
#define DBG_USART3_TX_Pin GPIO_PIN_8
#define DBG_USART3_TX_GPIO_Port GPIOD
#define DBG_USART3_RX_Pin GPIO_PIN_9
#define DBG_USART3_RX_GPIO_Port GPIOD

#define TMS_Pin GPIO_PIN_13
#define TMS_GPIO_Port GPIOA
#define TCK_Pin GPIO_PIN_14
#define TCK_GPIO_Port GPIOA
#define SWO_Pin GPIO_PIN_3
#define SWO_GPIO_Port GPIOB
#define LD2_Pin GPIO_PIN_7
#define LD2_GPIO_Port GPIOB

#ifdef __cplusplus
}
#endif

#endif /* __MAIN_H */
```

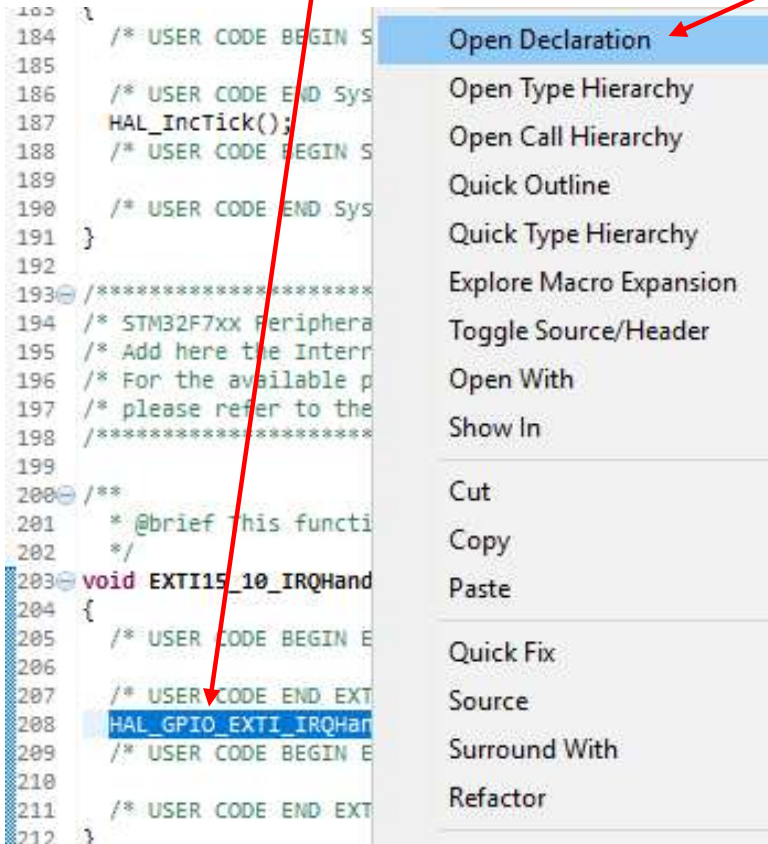
STM32CubeIDE – HAL GPIO EXTI Callback

`stm32f7xx_it.c` (Interrupt Service Routines)

```
/**
 * @brief This function handles EXTI line[15:10]
 interrupts.
 */
void EXTI15_10_IRQHandler(void)
{
    HAL_GPIO_EXTI_IRQHandler(USER_Btn_Pin);
}
```

Place the mouse pointer or cursor on this function and right click. Get the pop-up and select "Open Declaration".

`stm32f7xx_hal_gpio.c`



```
void HAL_GPIO_EXTI_IRQHandler(uint16_t GPIO_Pin)
{
    /* EXTI line interrupt detected */
    if(__HAL_GPIO_EXTI_GET_IT(GPIO_Pin) != RESET)
    {
        __HAL_GPIO_EXTI_CLEAR_IT(GPIO_Pin);
        HAL_GPIO_EXTI_Callback(GPIO_Pin);
    }
}

/* @brief EXTI line detection callbacks */
__weak void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    /* Prevent unused argument(s) compilation warning */
    UNUSED(GPIO_Pin);

    /* NOTE: This function Should not be modified, when
    the callback is needed, the HAL_GPIO_EXTI_Callback
    could be implemented in the user file */
}
```

STM32CubeIDE – main.c (1/3)

```
/* main.c */
/* Includes */
#include "main.h"
#include "usart.h"
#include "gpio.h"

/* Private includes */
/* USER CODE BEGIN Includes */
/* USER CODE END Includes */

/* Private typedef */
/* USER CODE BEGIN PTD */
/* USER CODE END PTD */

/* Private define */
/* USER CODE BEGIN PD */
/* USER CODE END PD */

/* Private macro */
/* USER CODE BEGIN PM */
/* USER CODE END PM */

/* Private variables */
/* USER CODE BEGIN PV */
/* USER CODE END PV */

/* Private function prototypes */
void SystemClock_Config(void);
/* USER CODE BEGIN PFP */
/* USER CODE END PFP */

/* Private user code */
/* USER CODE BEGIN 0 */
/* USER CODE END 0 */
```

```
int main(void)
{
    /* USER CODE BEGIN 1 */
    /* USER CODE END 1 */

    /* MCU Configuration: Reset of all peripherals,
       Initializes the Flash interface and the Systick. */
    HAL_Init();

    /* USER CODE BEGIN Init */
    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */
    /* USER CODE END SysInit */

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_USART3_UART_Init();
    /* USER CODE BEGIN 2 */
    /* USER CODE END 2 */

    /* Infinite loop */
    /* USER CODE BEGIN WHILE */
    while (1)
    {
        /* USER CODE END WHILE */

        /* USER CODE BEGIN 3 */
    }
    /* USER CODE END 3 */
}
```


STM32CubeIDE – main.c (2/3)

```
/* System Clock Configuration */
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

    /* Configure LSE Drive Capability */
    HAL_PWR_EnableBkUpAccess();

    /** Configure the main internal regulator output voltage */
    __HAL_RCC_PWR_CLK_ENABLE();
    __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE3);

    /** Initializes the RCC Oscillators according to the specified parameters
     *  in the RCC_OscInitTypeDef structure. */
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
    RCC_OscInitStruct.HSEState = RCC_HSE_BYPASS;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
    RCC_OscInitStruct.PLL.PLLM = 4;
    RCC_OscInitStruct.PLL.PLLN = 96;
    RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
    RCC_OscInitStruct.PLL.PLLQ = 4;
    RCC_OscInitStruct.PLL.PLLR = 2;
    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
    {
        Error_Handler();
    }

    /* Activate the Over-Drive mode */
    if (HAL_PWREx_EnableOverDrive() != HAL_OK)
    {
        Error_Handler();
    }
}
```

STM32CubeIDE – main.c (3/3)

```
/* Initializes the CPU, AHB and APB buses clocks */
RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK | RCC_CLOCKTYPE_SYSCLK
                             | RCC_CLOCKTYPE_PCLK1 | RCC_CLOCKTYPE_PCLK2;
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_3) != HAL_OK)
{
    Error_Handler();
}

/* USER CODE BEGIN 4 */
/* USER CODE END 4 */
/* This function is executed in case of error occurrence. */
void Error_Handler(void)
{
    /* USER CODE BEGIN Error_Handler_Debug */
    __disable_irq();
    while (1)
    {
    }
    /* USER CODE END Error_Handler_Debug */
}

#ifdef USE_FULL_ASSERT
void assert_failed(uint8_t *file, uint32_t line)
{
    /* USER CODE BEGIN 6 */
    /* User can add his own implementation to report the file name and line number,
       ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
    /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */
```

Add Code to USER CODE 4 (main.c)

```
/* USER CODE BEGIN 4 */

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if(GPIO_Pin == GPIO_PIN_13)
    {
        HAL_GPIO_TogglePin(GPIOB, LD2_Pin);
    }
}

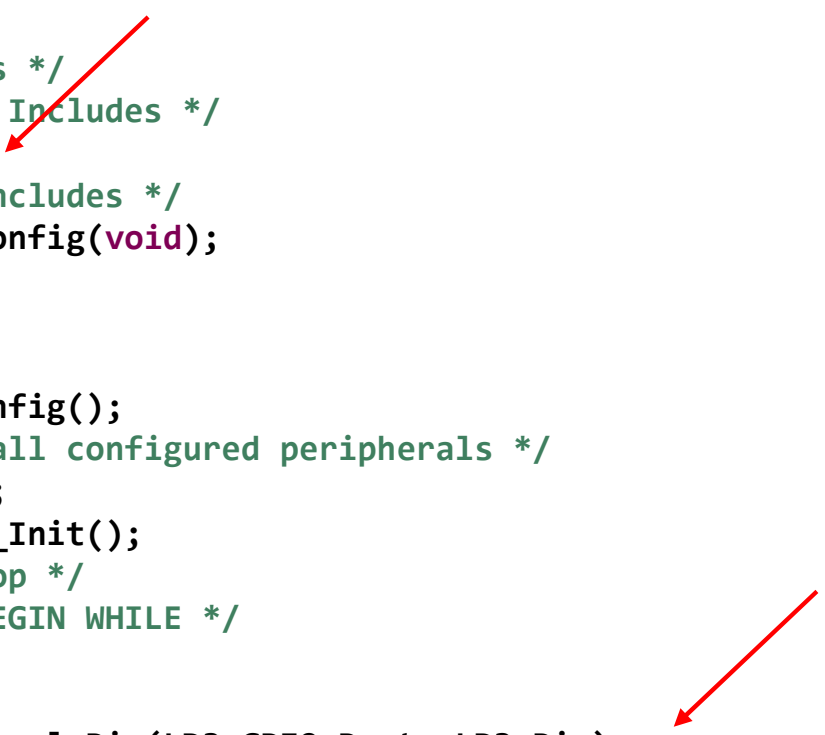
int __io_putchar(int ch)
{
    uint8_t c[1];
    c[0] = ch & 0x00FF;
    HAL_UART_Transmit(&huart3, &*c, 1, 10);
    return ch;
}
// comment the function _write() below, re-compile and run to check the program
int _write(int file, char *ptr, int len)
{
    int DataIdx;
    for(DataIdx= 0; DataIdx< len; DataIdx++)
    {
        __io_putchar(*ptr++);
    }
    return len;
}

/* USER CODE END 4 */
```

STM32CubeIDE

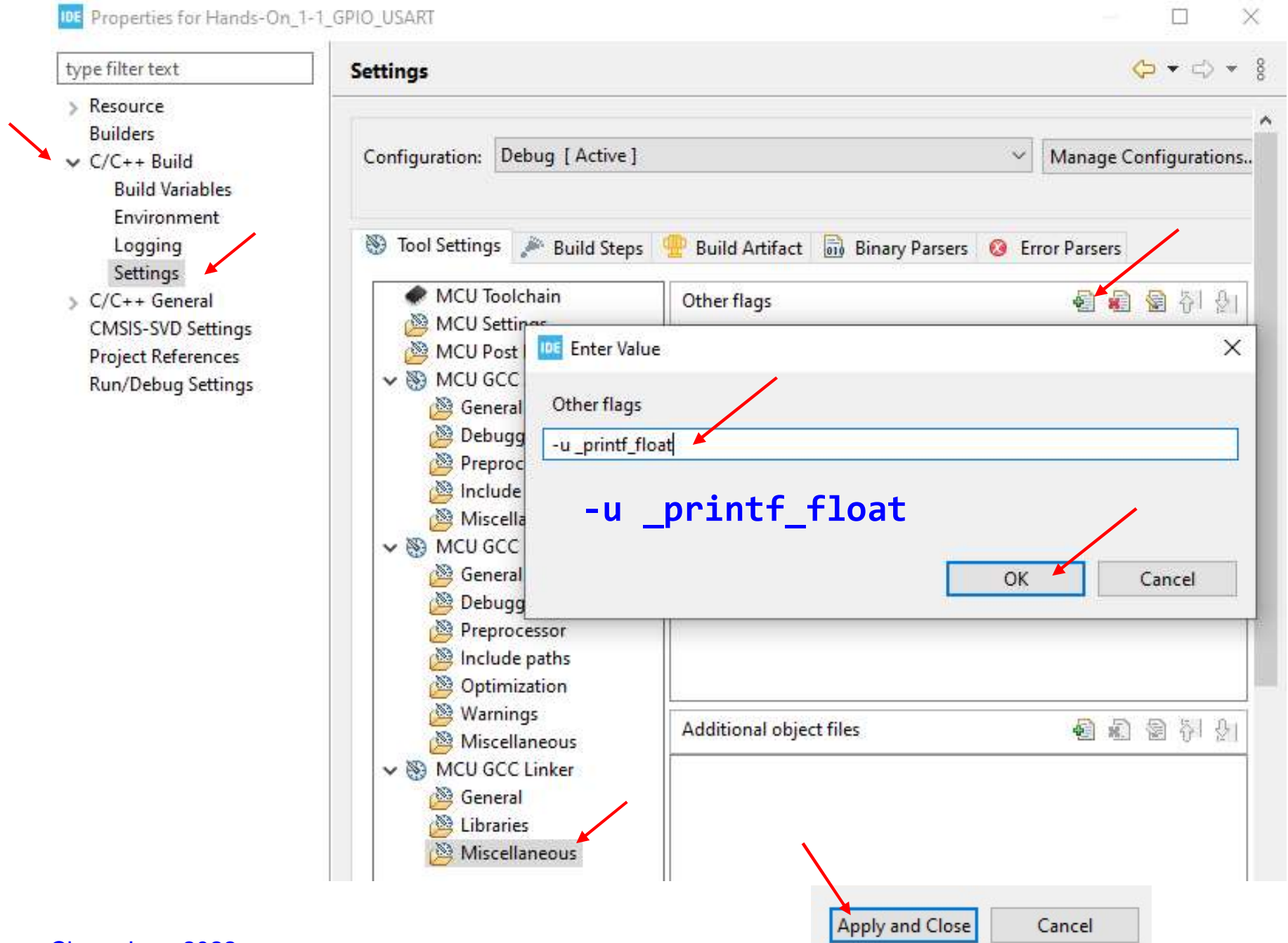
Add Code to USER CODE BEGIN Includes and WHILE (main.c)

```
/* main.c */
/* Includes */
#include "main.h"
#include "usart.h"
#include "gpio.h"
/* Private includes */
/* USER CODE BEGIN Includes */
#include <stdio.h>
/* USER CODE END Includes */
void SystemClock_Config(void);
int main(void)
{
    HAL_Init();
    SystemClock_Config();
    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_USART3_UART_Init();
    /* Infinite loop */
    /* USER CODE BEGIN WHILE */
    while (1)
    {
        HAL_GPIO_TogglePin(LD3_GPIO_Port, LD3_Pin);
        printf("Print vales = %d %ld %4.3f \r\n", 33, 345UL, 8.456f);
        HAL_Delay(500);
        // project -> properties -> C/C++ Build - Settings -> Tool Settings -> MCU GCC Linker -
        // - Miscellaneous -> Other Flags: -u _printf_float
    }
    /* USER CODE END WHILE */
    /* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */
```



STM32CubeIDE

Project Properties: Add "-u _printf_float" to MCU GCC Linker – Miscellaneous – Other flags



STM32CubeIDE

Project – Properties – Language Standard

The screenshot shows the 'Properties for Hands-On_1-1_GPIO_USART' dialog box in STM32CubeIDE. The 'Settings' tab is active, and the 'Language standard' dropdown menu is open. The menu lists various language standards, with 'GNU11 (ISO C11 + gnu extensions) (-std=gnu11)' selected and marked as the default. Red arrows point to the 'MCU Toolchain' and 'MCU GCC Compiler' settings in the left sidebar, and to the 'GNU11' option in the dropdown menu. The top right corner of the dialog indicates '(GCC 10.3)'.

Settings (GCC 10.3)

Configuration: Debug [Active]

MCU Toolchain

Select what toolchain to use

☒ Use workspace setting (GNU Tools for STM32 (10.3-2021.10))

☐ Fixed GNU Tools for STM32 (10.3-2021.10)

Language standard

- GNU11 (ISO C11 + gnu extensions) (-std=gnu11) (default)
- GCC default
- ISO C90 / ANSI C89 (-std=c90)
- ISO 9899:199409 (-std=iso9899:199409)
- ISO C99 (-std=c99)
- ISO C11 (-std=c11)
- ISO C17 (-std=c17)
- ISO C18 (-std=c18)
- GNU90 (ISO C90 + gnu extensions) (-std=gnu90)
- GNU99 (ISO C99 + gnu extensions) (-std=gnu99)
- GNU17 (ISO C17 + gnu extensions) (-std=gnu17)
- GNU18 (ISO C18 + gnu extensions) (-std=gnu18)

MCU GCC Compiler

General

Debugging

Preprocessor

Include paths

Optimization

Warnings

Miscellaneous

MCU GCC Linker

General

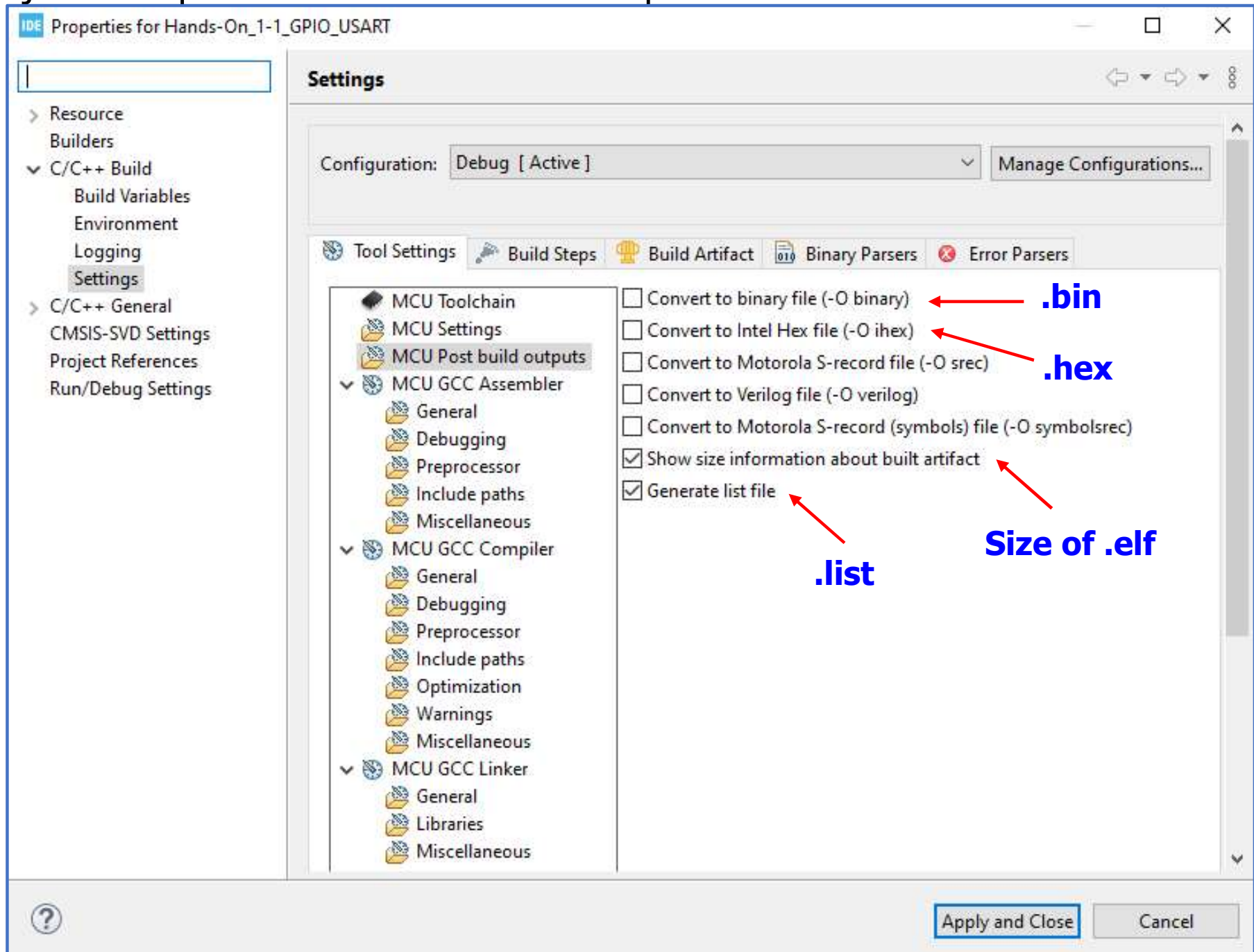
Libraries

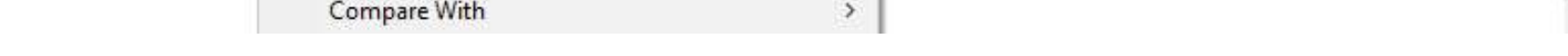
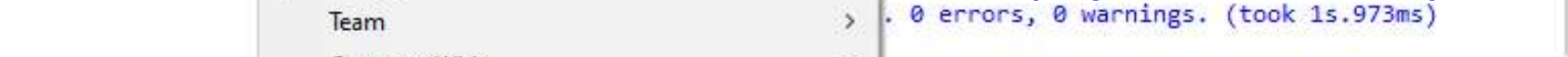
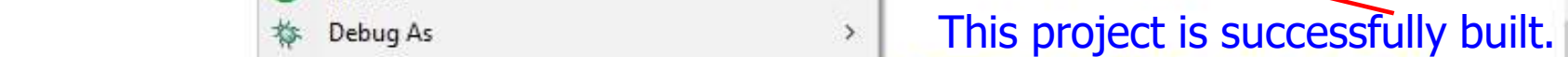
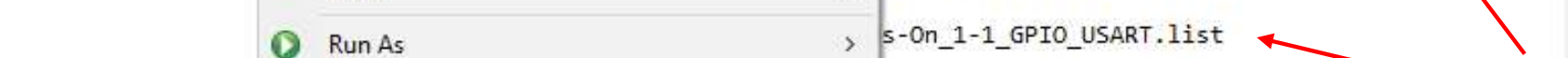
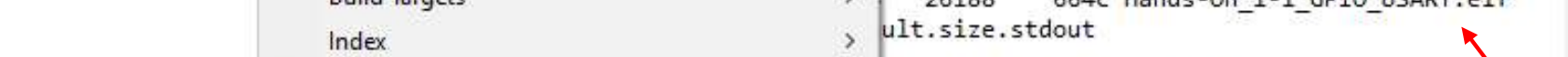
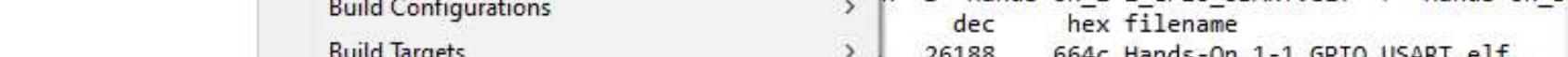
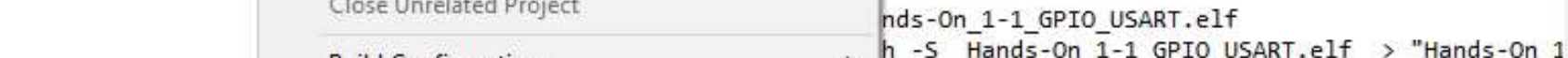
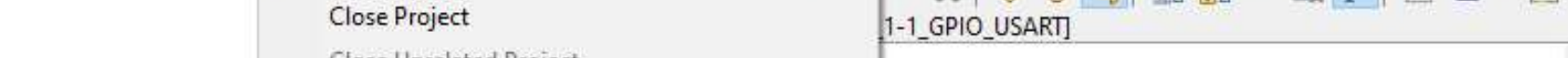
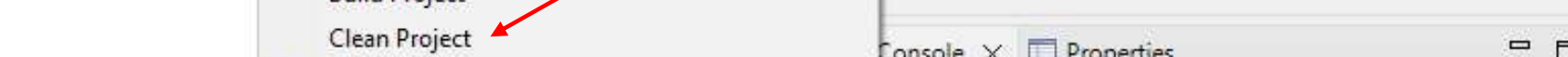
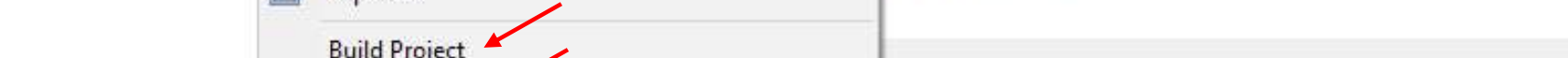
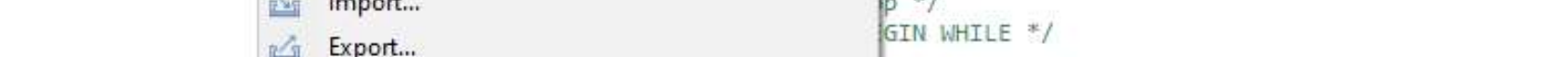
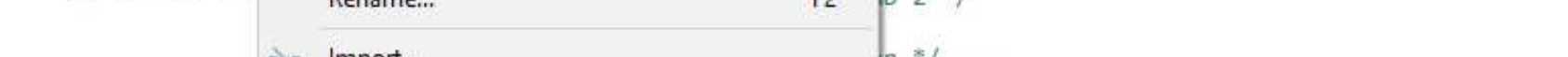
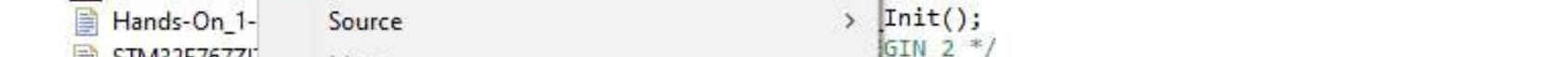
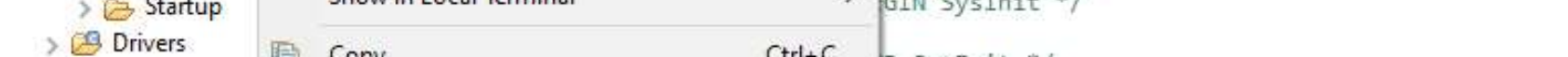
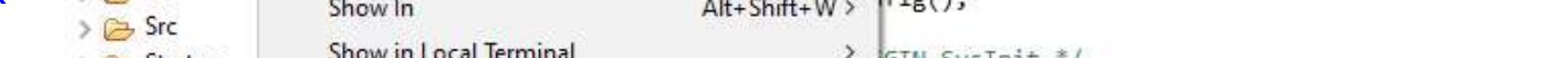
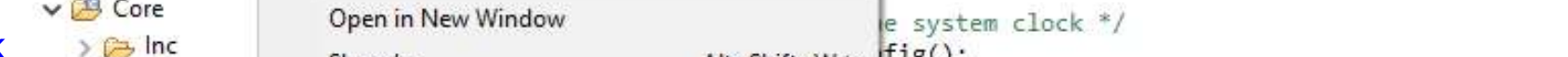
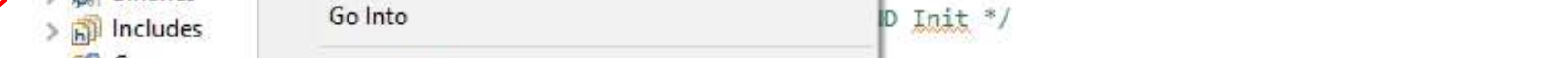
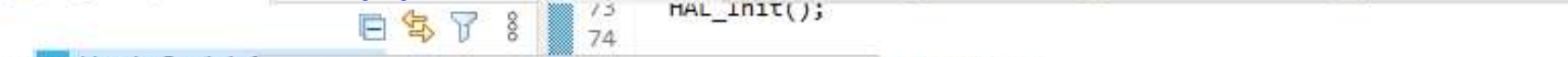
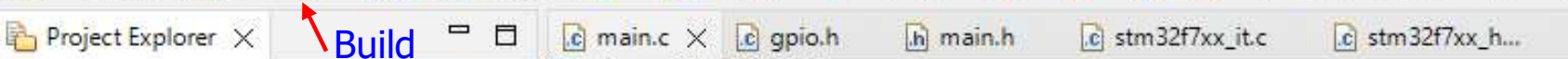
Miscellaneous

Apply and Close Cancel

STM32CubeIDE

Project – Properties – MCU Post Build Outputs





Right-click

Debug

Build

```

/*
 * INIT Init */
/*
 * ID Init */
/*
 * e system clock */
/*
 * fig();
/*
 * GIN SysInit */
/*
 * ID SysInit */
/*
 * ll configured peripherals */
/*
 * Init();
/*
 * GIN 2 */
/*
 * ID 2 */
/*
 * p */
/*
 * GIN WHILE */

```

Console X Properties



1-1_GPIO_USART]

nds-On_1-1_GPIO_USART.elf

h -S Hands-On_1-1_GPIO_USART.elf > "Hands-On_1

dec hex filename

> 26188 664c Hands-On_1-1_GPIO_USART.elf

> ult.size.stdout

> s-On_1-1_GPIO_USART.list

> . 0 errors, 0 warnings. (took 1s.973ms)

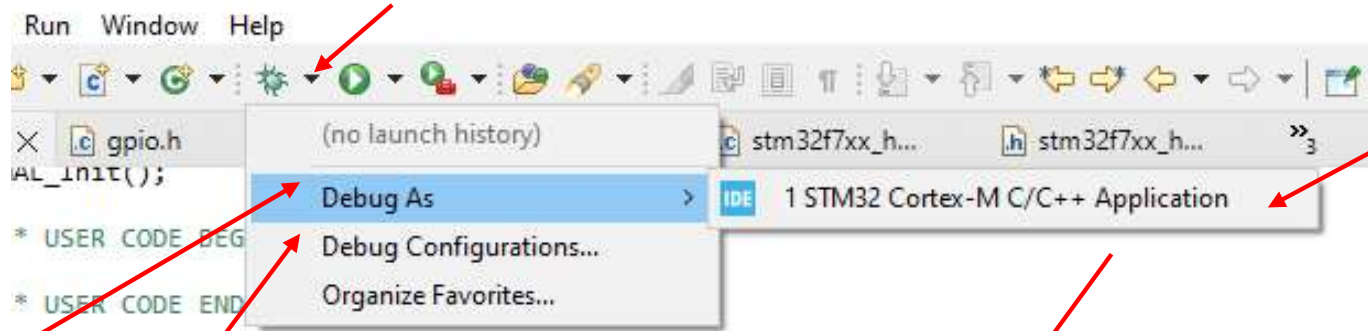
>

>

>

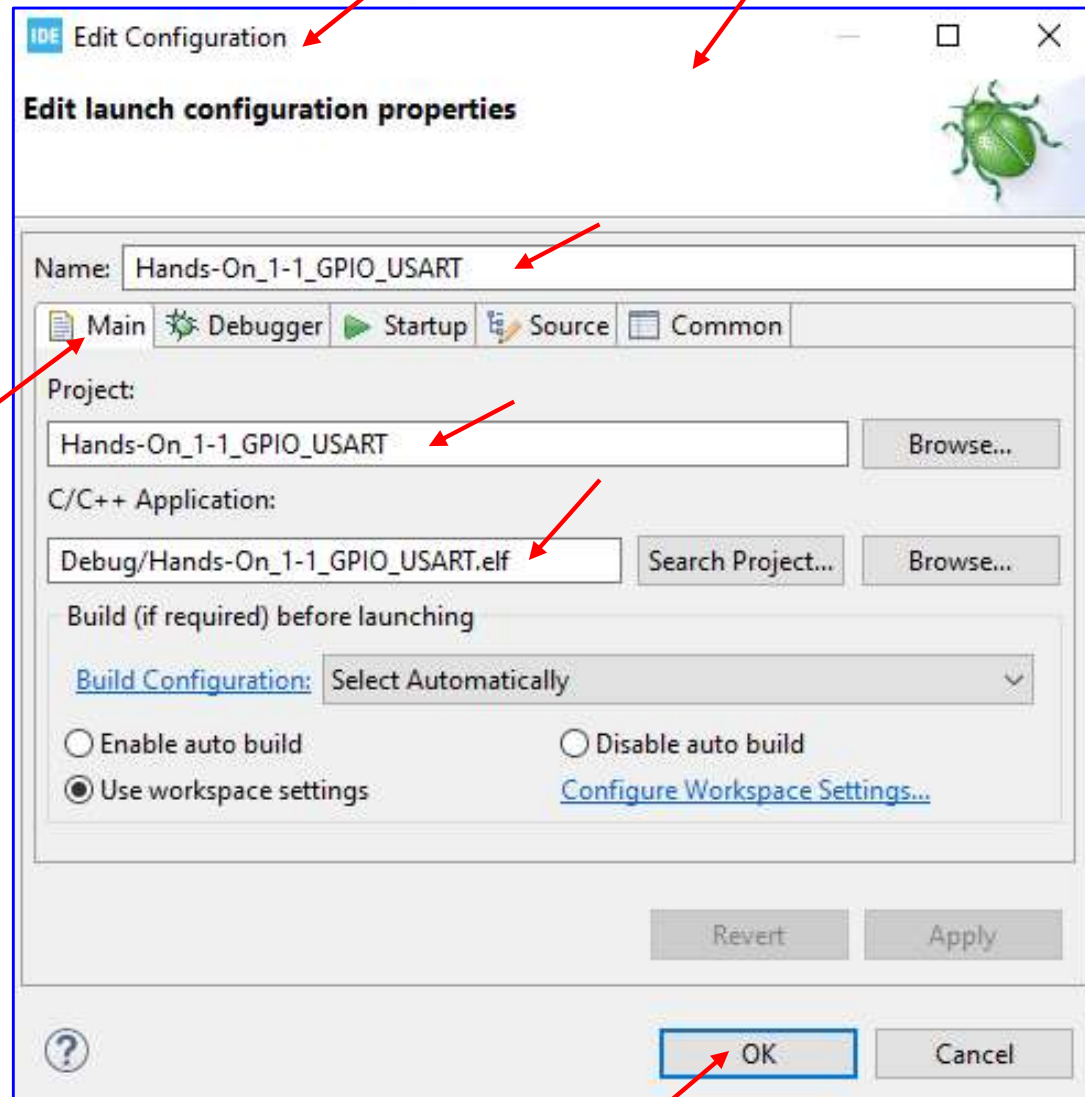
This project is successfully built.

STM32CubeIDE – Debug As (Configuration)

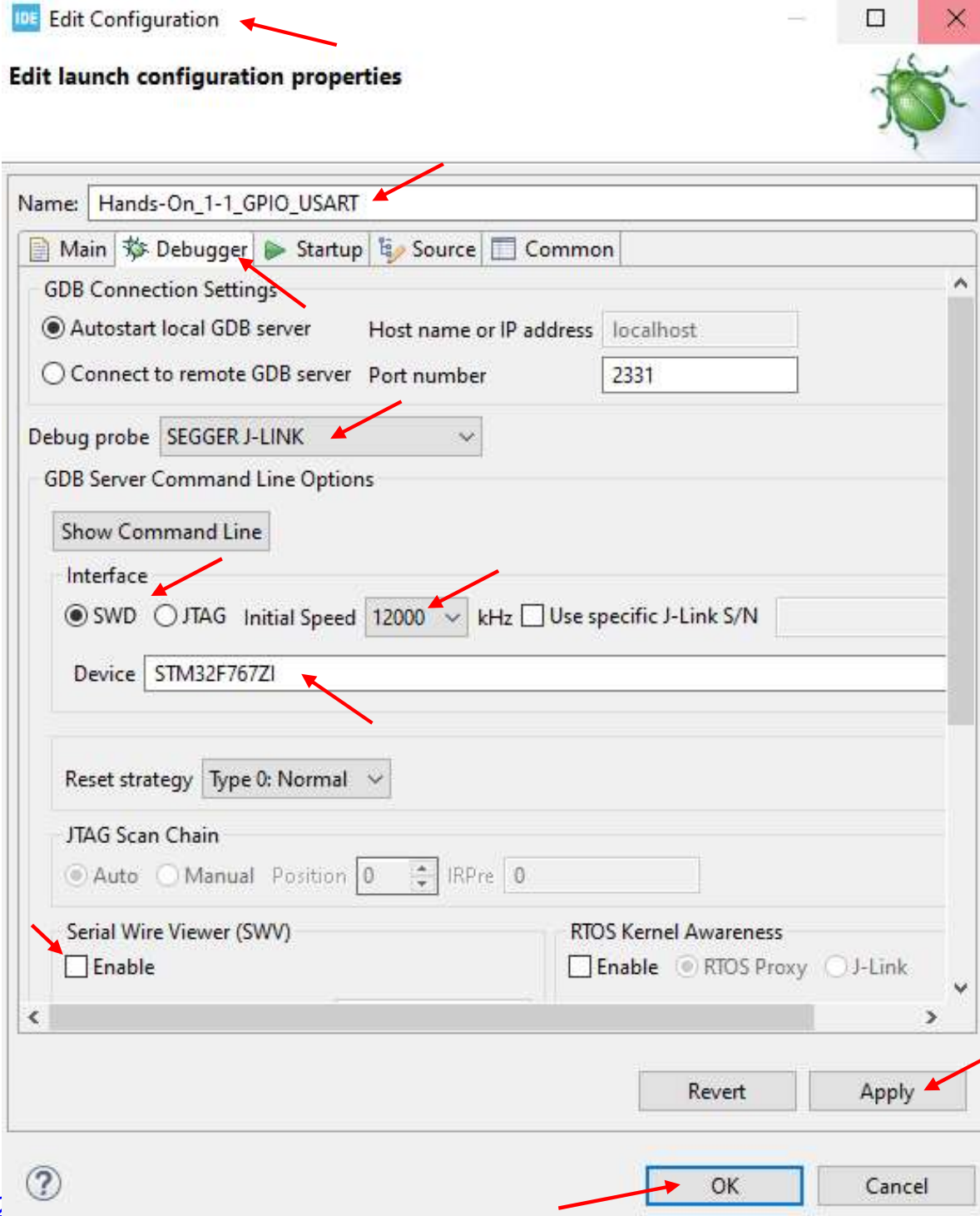


(first time)

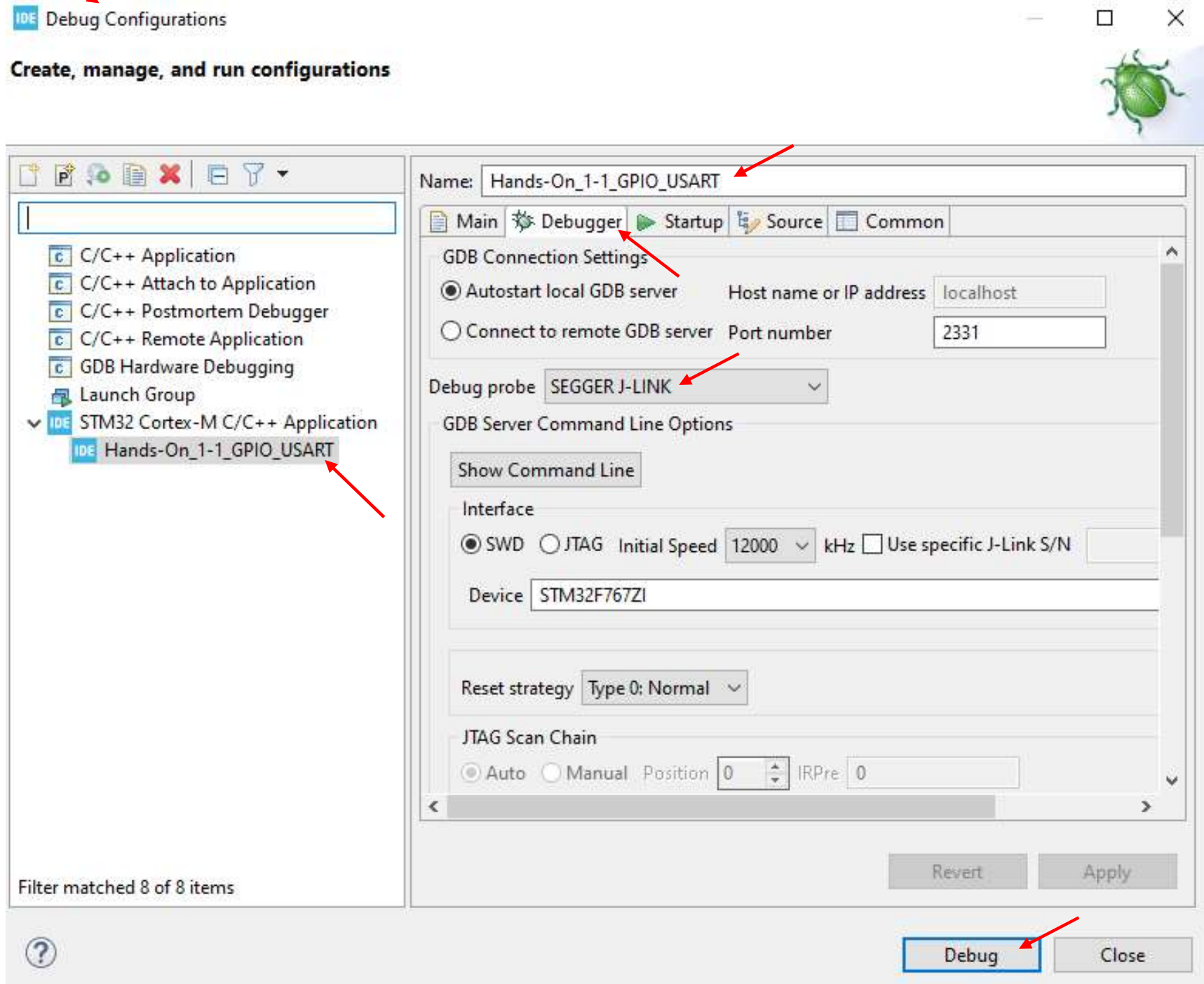
(then)



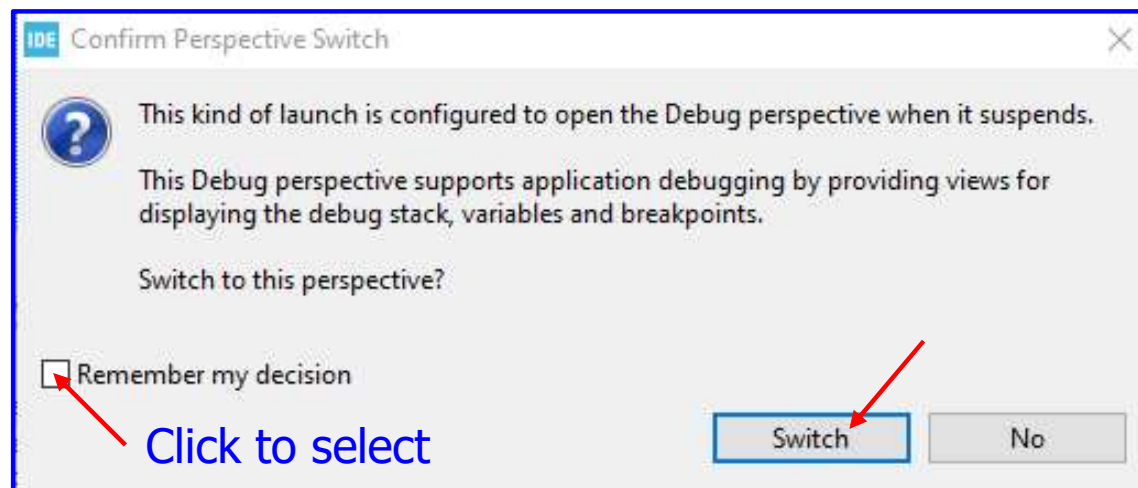
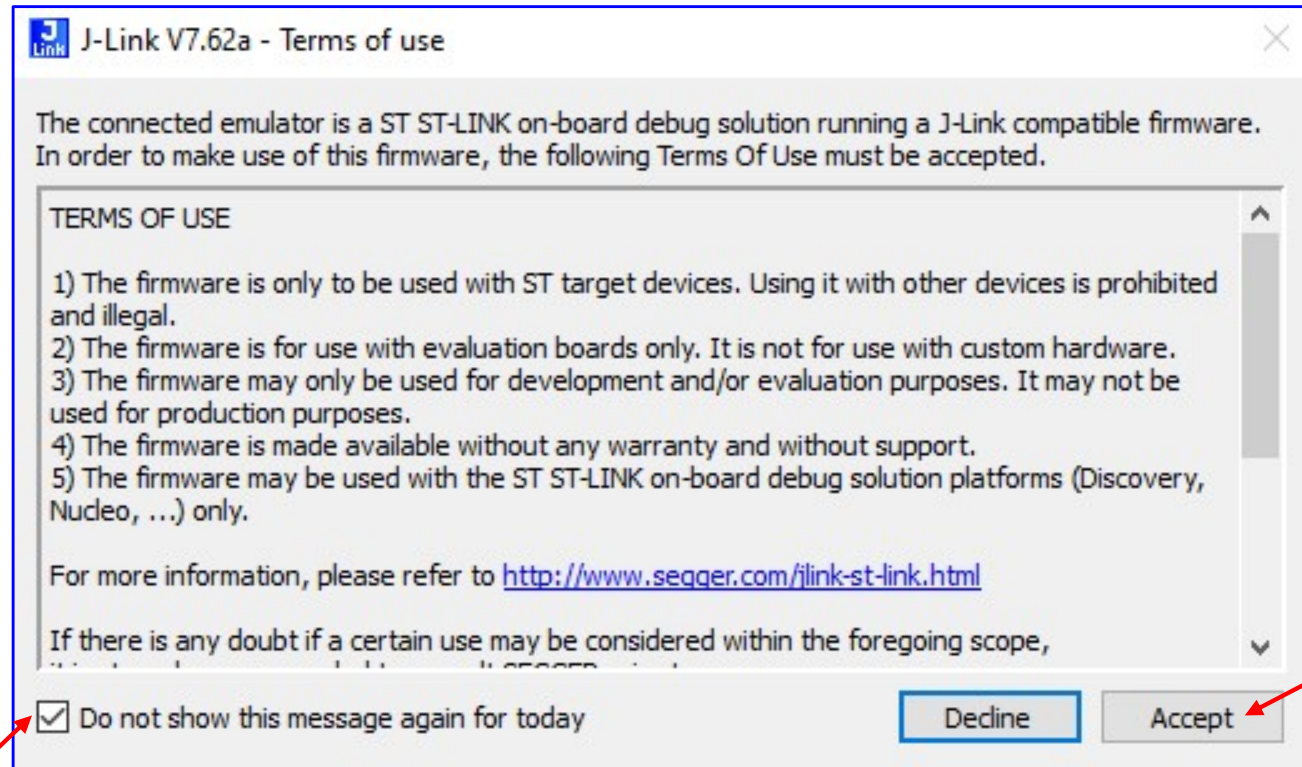
STM32CubeIDE – Debug Configuration (first time)



STM32CubeIDE – Debug Configuration (after first time)



STM32CubeIDE – Debug



STM32CubeIDE

② Resume (F8)

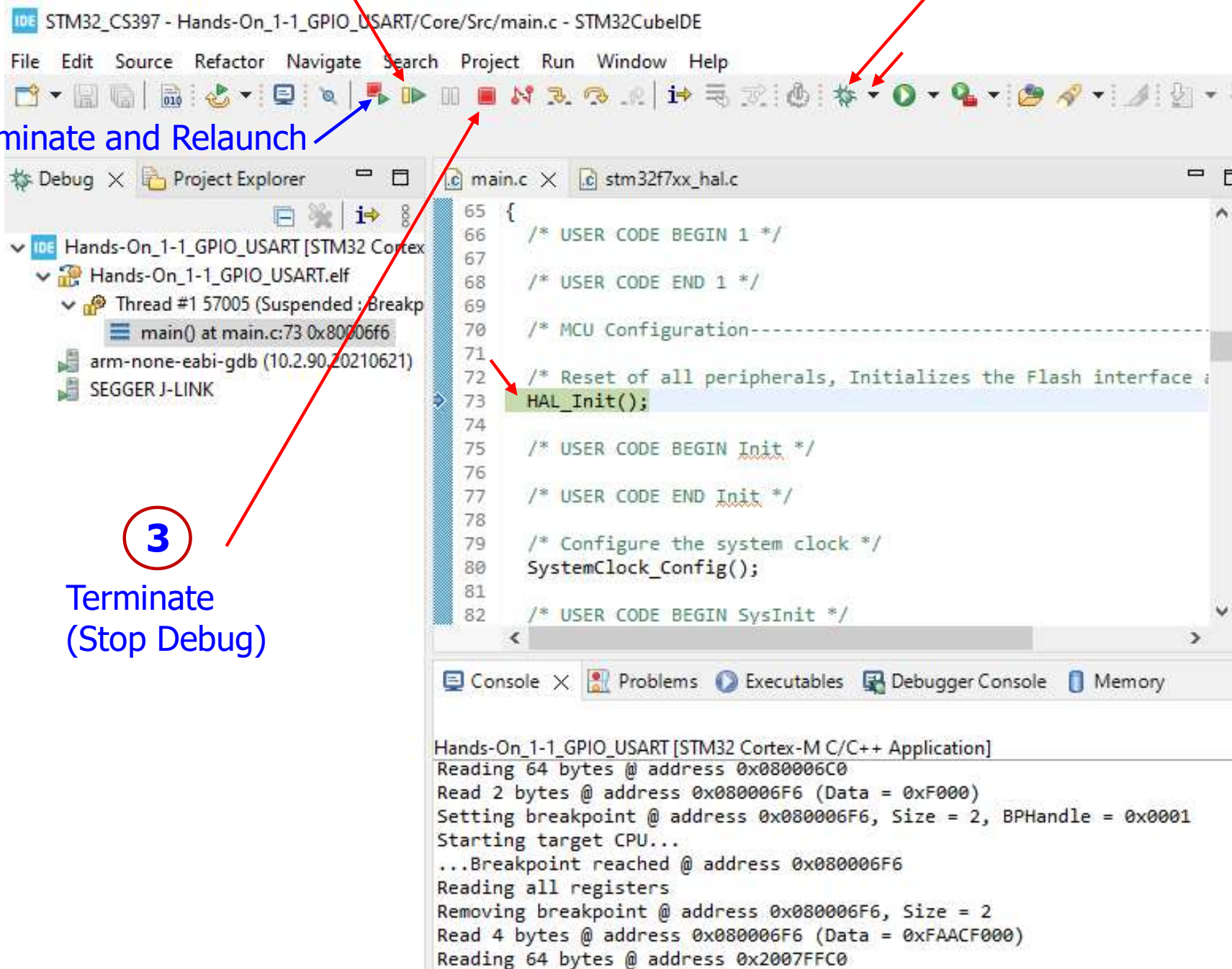
①

Debug: Download Program & Enter Debug Mode

Terminate and Relaunch

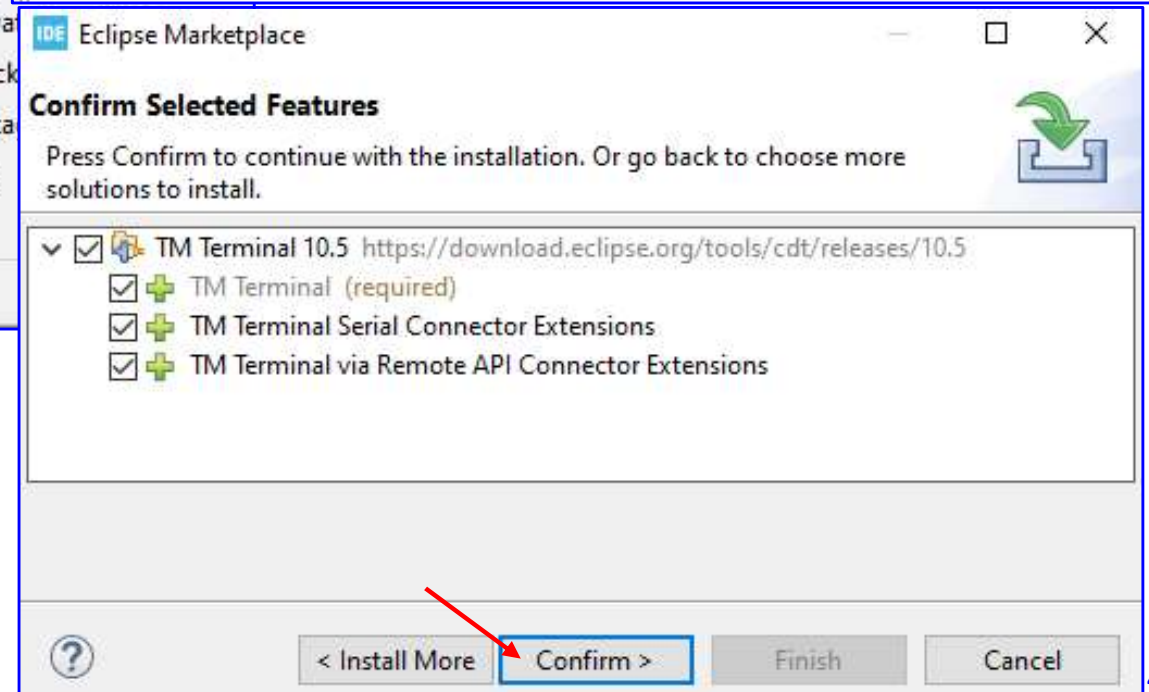
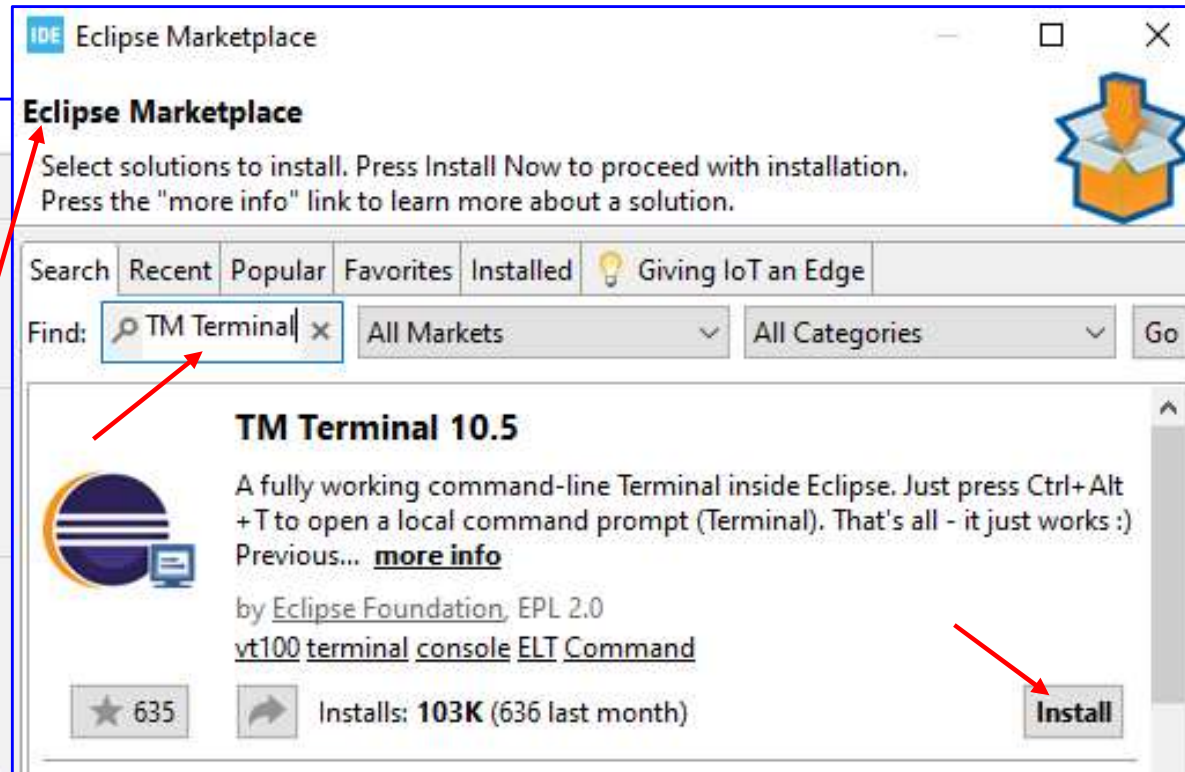
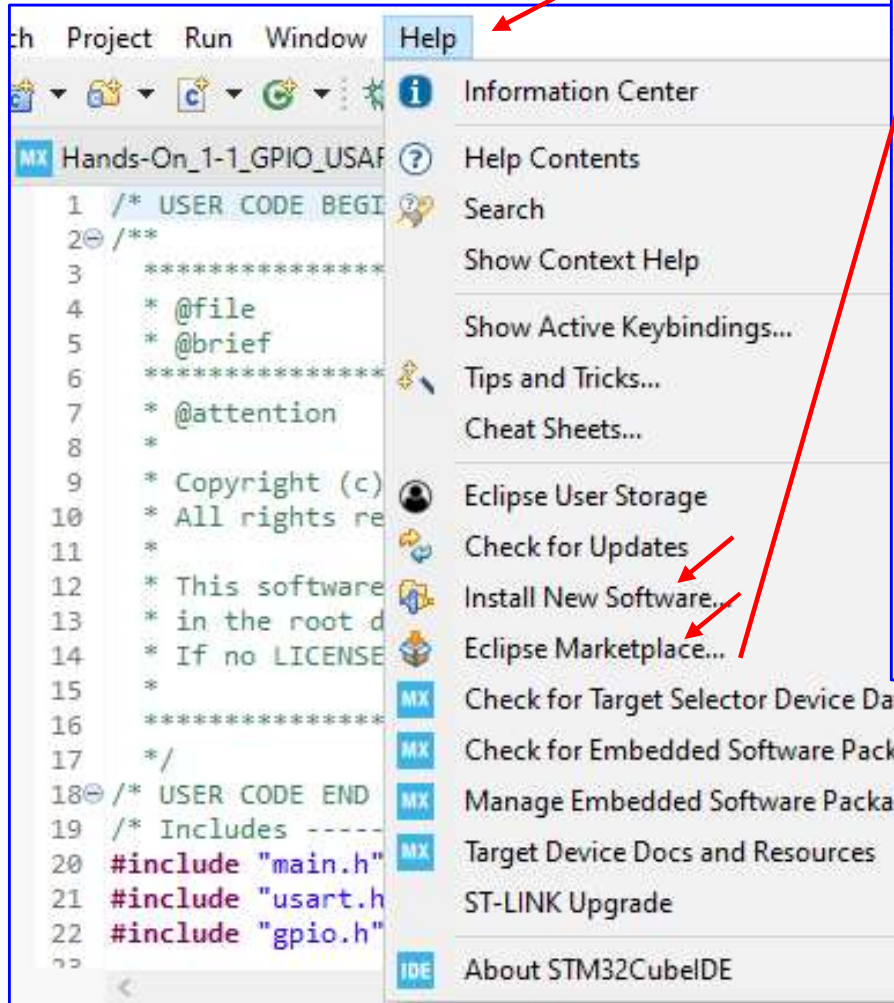
③

Terminate
(Stop Debug)

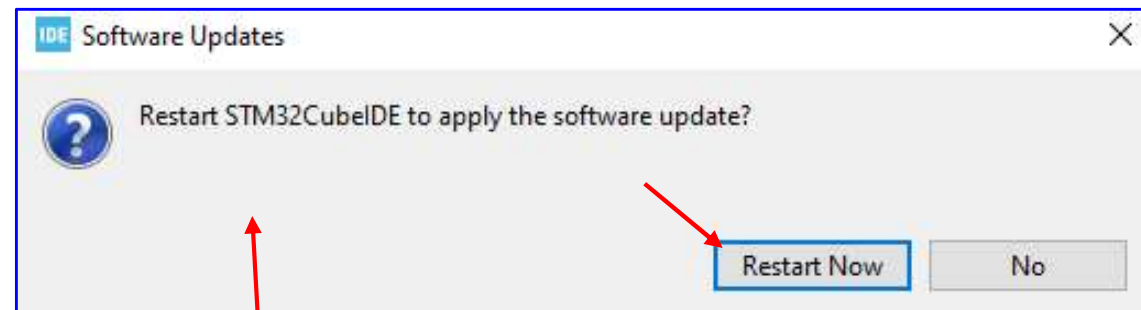
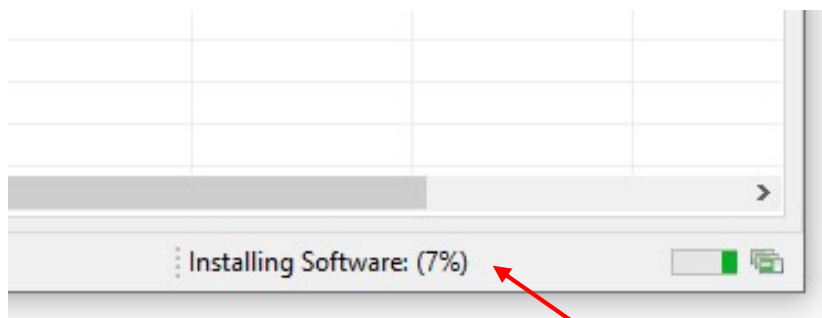
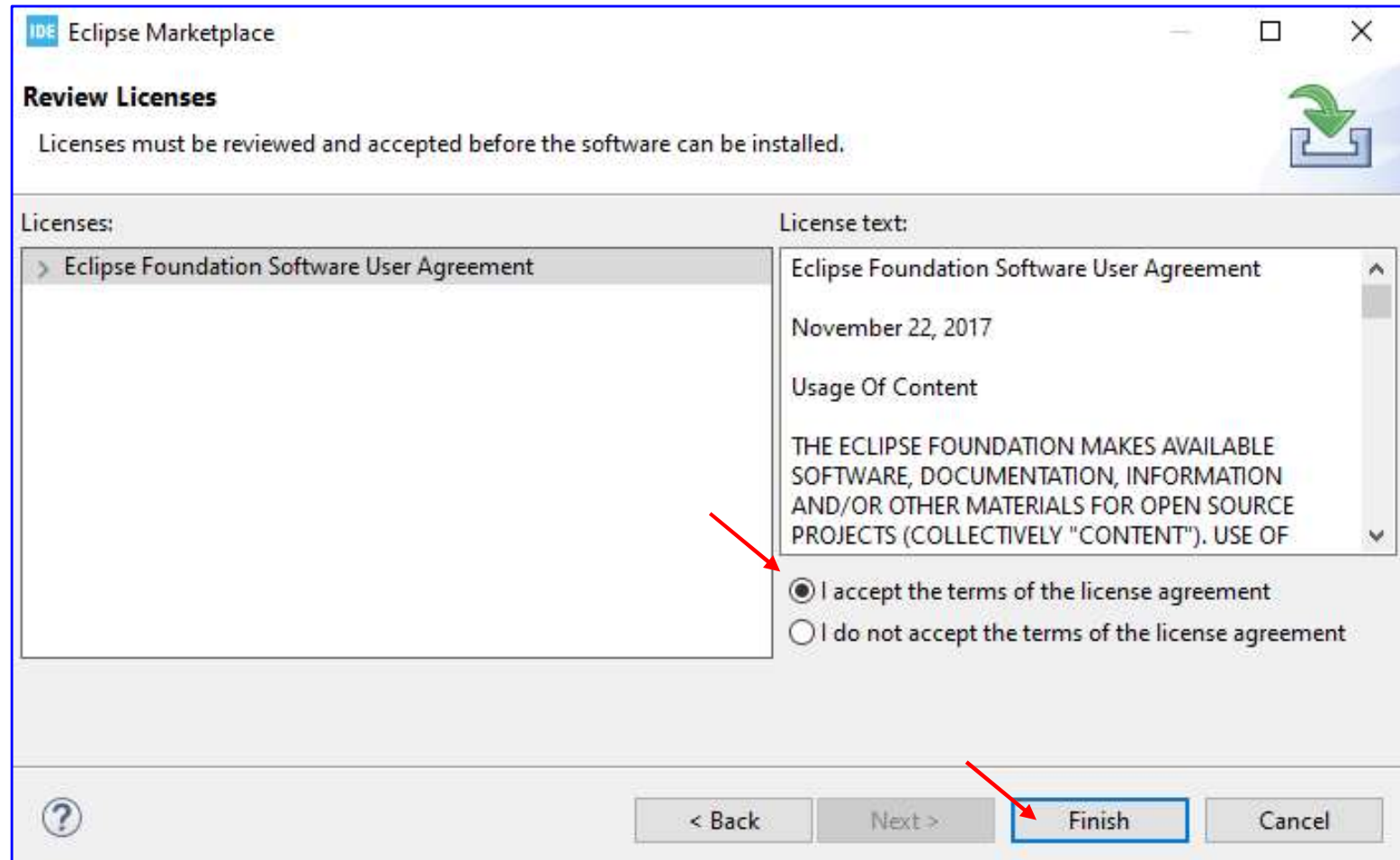


STM32CubeIDE

Install TM Terminal



STM32CubeIDE



wait ...until software installed.

STM32CubeIDE

Or, Install New Software: TM Terminal

Available Software
Check the items that you wish to install.

Work with:

Name	Version
General Purpose Tools	
<input checked="" type="checkbox"/> TM Terminal	10.5.0.202109270023
<input checked="" type="checkbox"/> TM Terminal Serial Connector Extensions	10.5.0.202109270023
<input type="checkbox"/> TM Terminal via Remote API Connector Extensions	10.5.0.202110051557
Mobile and Device Development	
<input type="checkbox"/> TCF Target Explorer	1.7.0.202002082211

2 items selected

Details

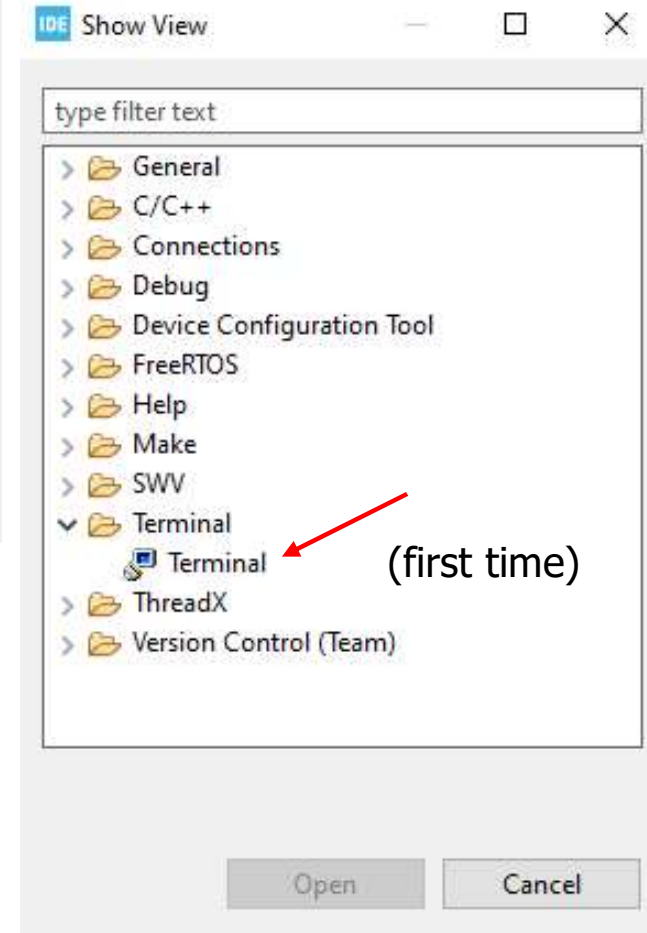
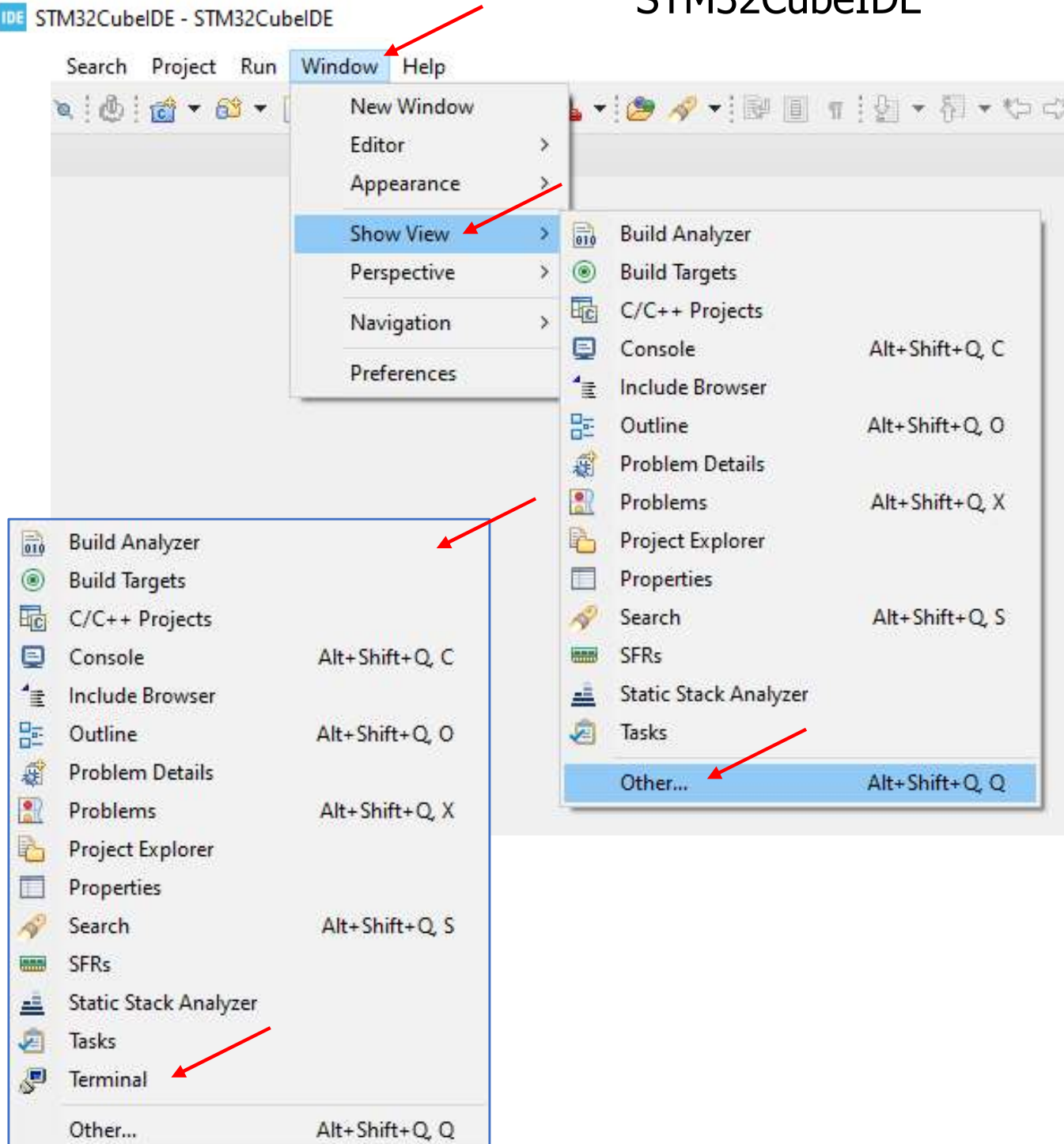
- ☒ Show only the latest versions of available software
- ☒ Group items by category
- ☒ Show only software applicable to target environment
- ☒ Contact all update sites during install to find required software

1. Go to "Help" menu -> Install New Software...
2. Select for "Work with" -> --All Available Sites--
3. In the search box (type filter text) type "terminal".
4. When search is finished, select "TM Terminal" and "TM Terminal Serial Connector Extensions" and install them (it takes times to install).
5. Restart the STM32CubeIDE after installation.

TM Terminal

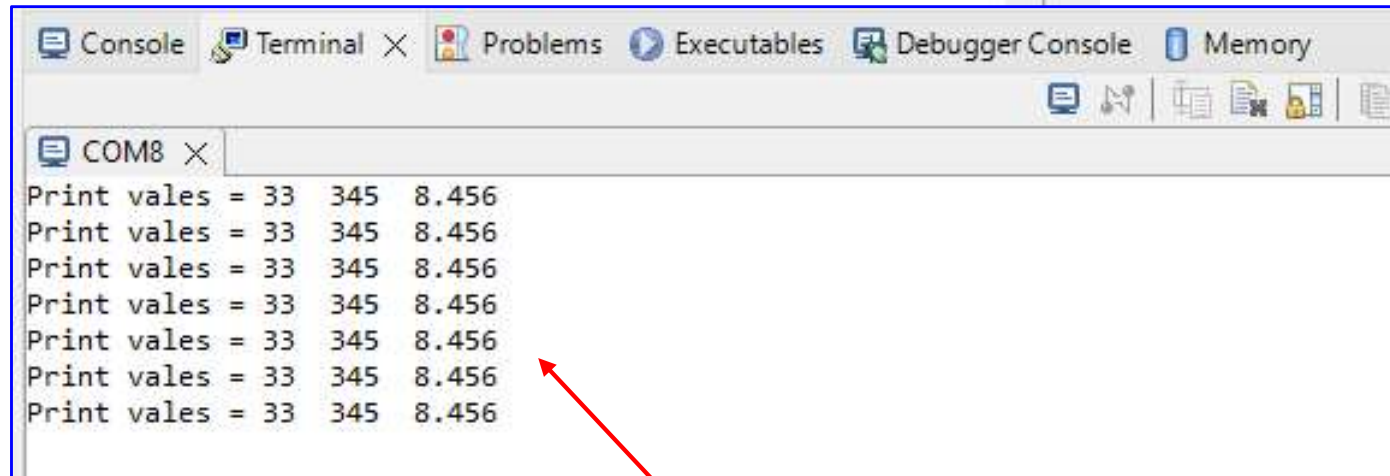
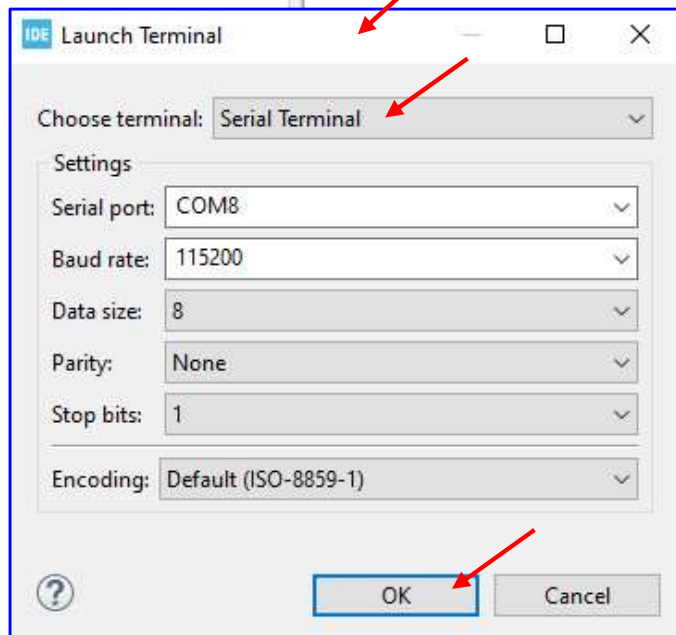
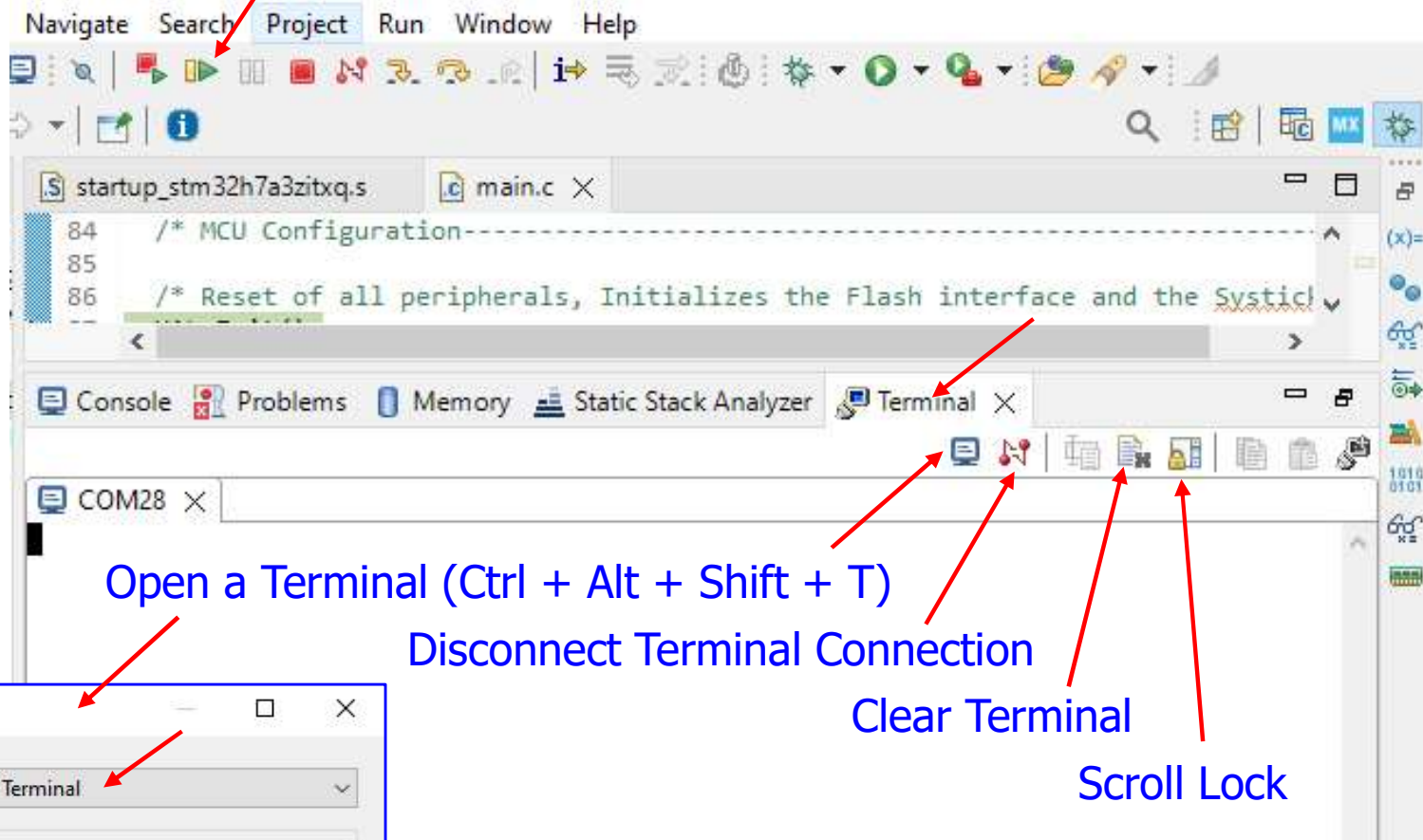
STM32CubeIDE

IDE STM32CubeIDE - STM32CubeIDE



STM32CubeIDE

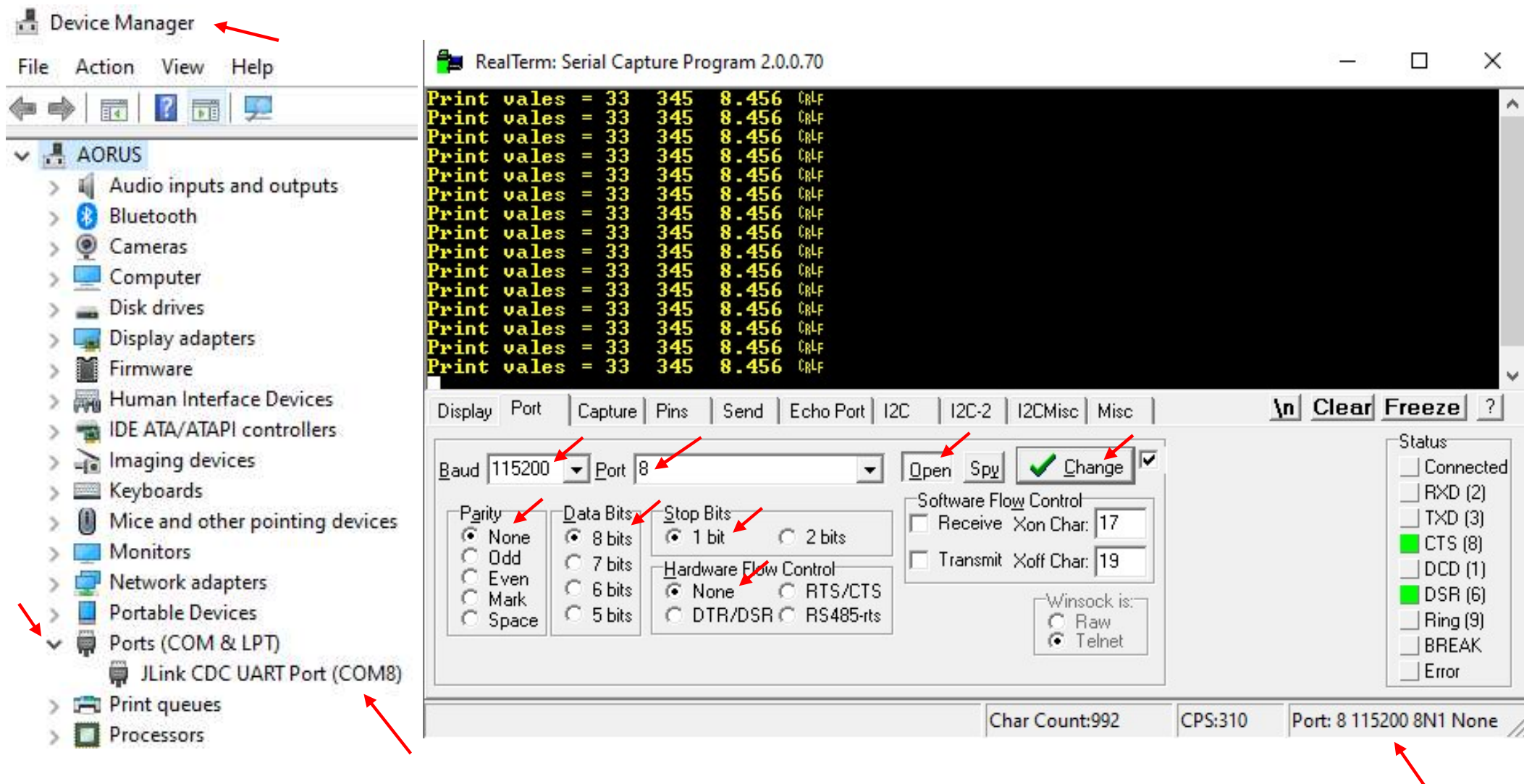
TM Terminal



STM32CubeIDE

Test Program Running in STM32F767ZI

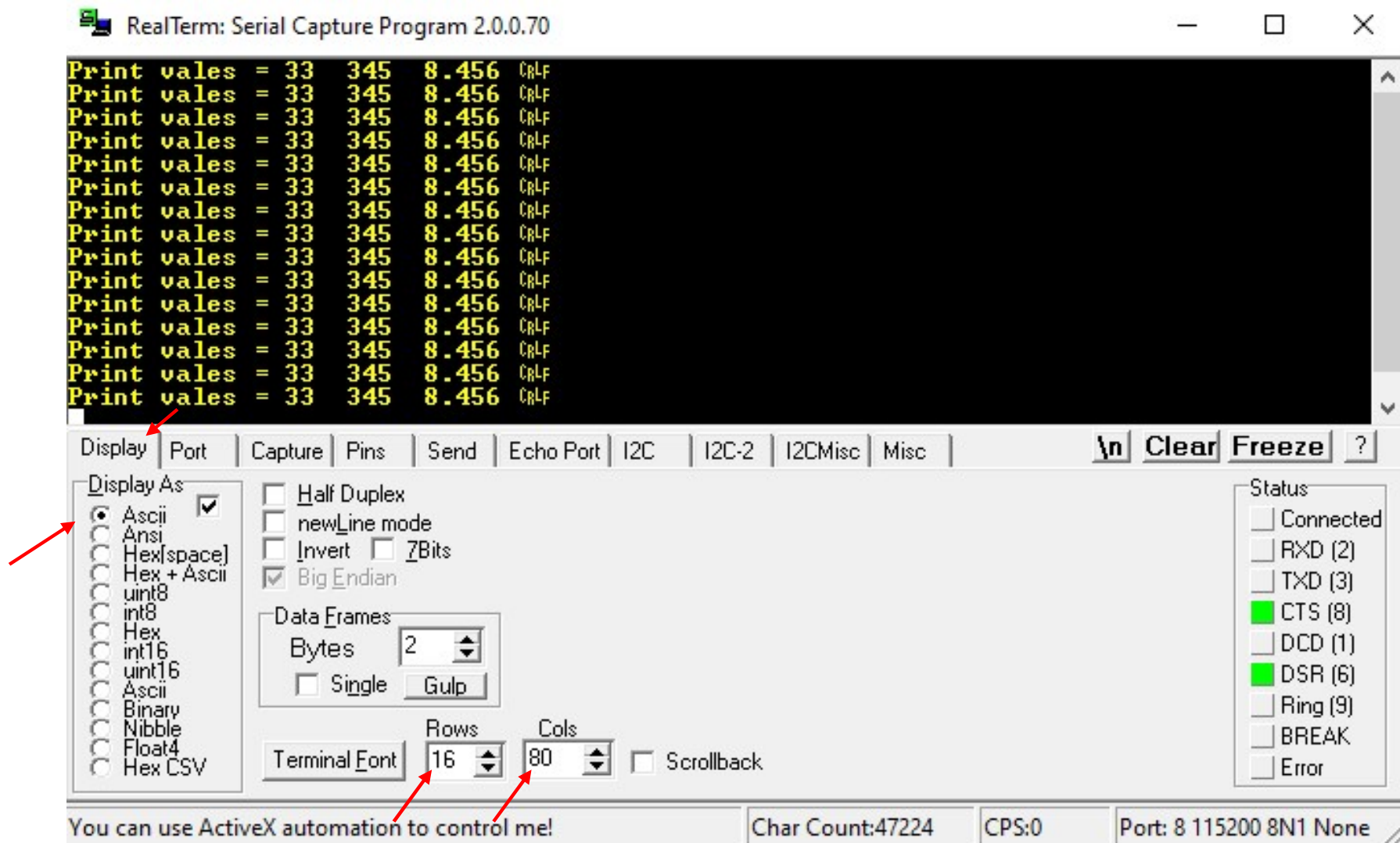
- Check "Device Manager, Ports (COM & LPT)", and find the connected JLink CDC UART Port
- Run "RealTerm", ensure proper settings, and click "Change", then "Open" to capture data



STM32CubeIDE

RealTerm, a Serial Terminal (COM Port) Software <https://realterm.sourceforge.io/>

RealTerm is an engineer's terminal software specially designed for capturing, controlling and debugging binary and other difficult data streams.



STM32CubeIDE

Test Program

- Check the values displayed on the TM Terminal or RealTerm and ensure that they are correct.
- Observe the flashing LD3 (Red LED) and measure the flashing rate.
- Press the User button (B1) a few times slowly to see the response of LD2 (Blue LED).
- Modify the program to make the values displayed on the TM Terminal or RealTerm changing accordingly to a programmed sequence.