STM32CubeMX Hands-on

life.augmented

STM32L4+

High integration with high memory size in small packages

Parallel Interface

FSMC 8-/16-bit
(TFT-LCD, SRAM, NOR, NAND)

Display

DSI MIPI
LCD TFT Controller

Timers

17 timers including:
2 x 16-bit advanced motor control timers
2 x ULP timers
7 x 16-bit-timers
2 x 32-bit timers

I/Os

Up to 136 I/Os
Touch-sensing controller

Cortex-M4
120 MHz
FPU
MPU
ETM

DMA

ART Accelerator™

Chrom-ART™

Up to
2-Mbyte Flash
with ECC
Dual Bank

640-Kbyte RAM

Connectivity

USB OTG Crystal less,
1x SD/SDIO/MMC, 3 x SPI,
4 x I²C, 1x CAN, 2 x Octo SPI,
5 x USART + 1 x ULP UART

Digital

AES (256-bit), SHA (256-bit),
TRNG, 2 x SAI, DFSDM (8 channels),
Camera I/F, Chrom-GRC™

Analog

1 x 16-bit ADC, 2 x DAC,
2 x comparators, 2 x op amps
1 x temperature sensor

Package size down to 5.24 x 5.24 mm

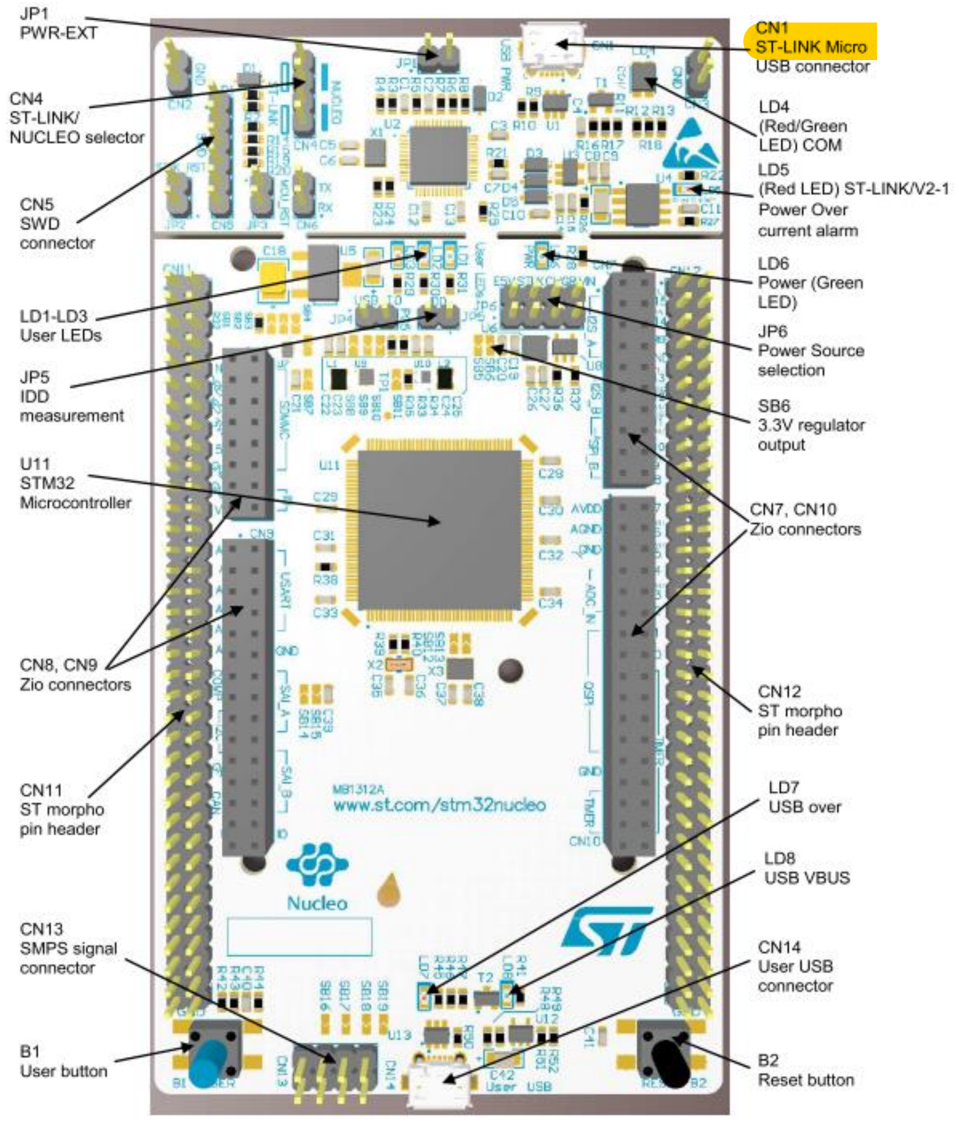
For Compensate osillator

+ More Integration

life.augmented

STM32L4R5 Nucleo-144

- Features
  - Flexible board power supply
    - USB or external source
  - Integrated ST-Link/V2-1 debugger
    - Drag & drop device flash programming
    - Virtual COM port
  - For user application
    - 3 LED
    - Push button (blue)
  - STM32L4R5 microcontroller
  - USB OTG
  - Connectors
    - Arduino Uno
    - ST Zio
    - ST Morpho Extension - direct access to all MCU I/Os



JP1 PWR-EXT
CN4 ST-LINK/ NUCLEO selector
CN5 SWD connector
LD1-LD3 User LEDs
JP5 IDD measurement
U11 STM32 Microcontroller
CN8, CN9 Zio connectors
CN11 ST morpho pin header
CN13 SMPS signal connector
B1 User button
CN1 ST-LINK Micro USB connector
LD4 (Red/Green LED) COM
LD5 (Red LED) ST-LINK/V2-1 Power Over current alarm
LD6 Power (Green LED)
JP6 Power Source selection
SB6 3.3V regulator output
CN7, CN10 Zio connectors
CN12 ST morpho pin header
LD7 USB over
LD8 USB VBUS
CN14 User USB connector
B2 Reset button
www.st.com/stm32nucleo
Nucleo

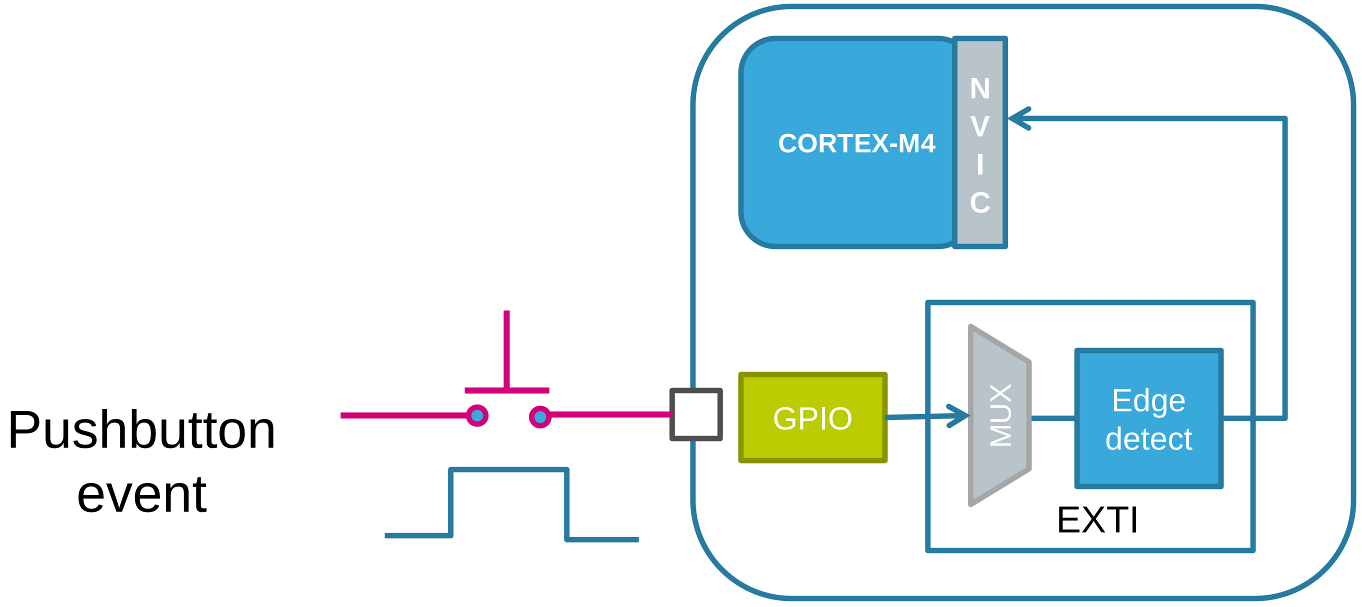
Patch for Atollic TrueSTUDIO v9.0.0

- Atollic TrueSTUDIO v9.0.0 is unable to erase the flash when STM32L4R5 is configured in dual-bank mode (default from factory)

- Workaround
  - Configure STM32L4R5 in single bank mode **or**
  - Replace C:\Program Files (x86)\Atollic\TrueSTUDIO for STM32 9.0.0\Servers\ST-LINK_gdbserver\ST-LINK_gdbserver.exe with a fixed version
    - Fixed version can be found in the thumb drive content
      - D:\Atollic TrueSTUDIO\TrueSTUDIO_Patch.zip
    - Unzip ST-LINK_gdbserver.exe into the specified TrueSTUDIO folder

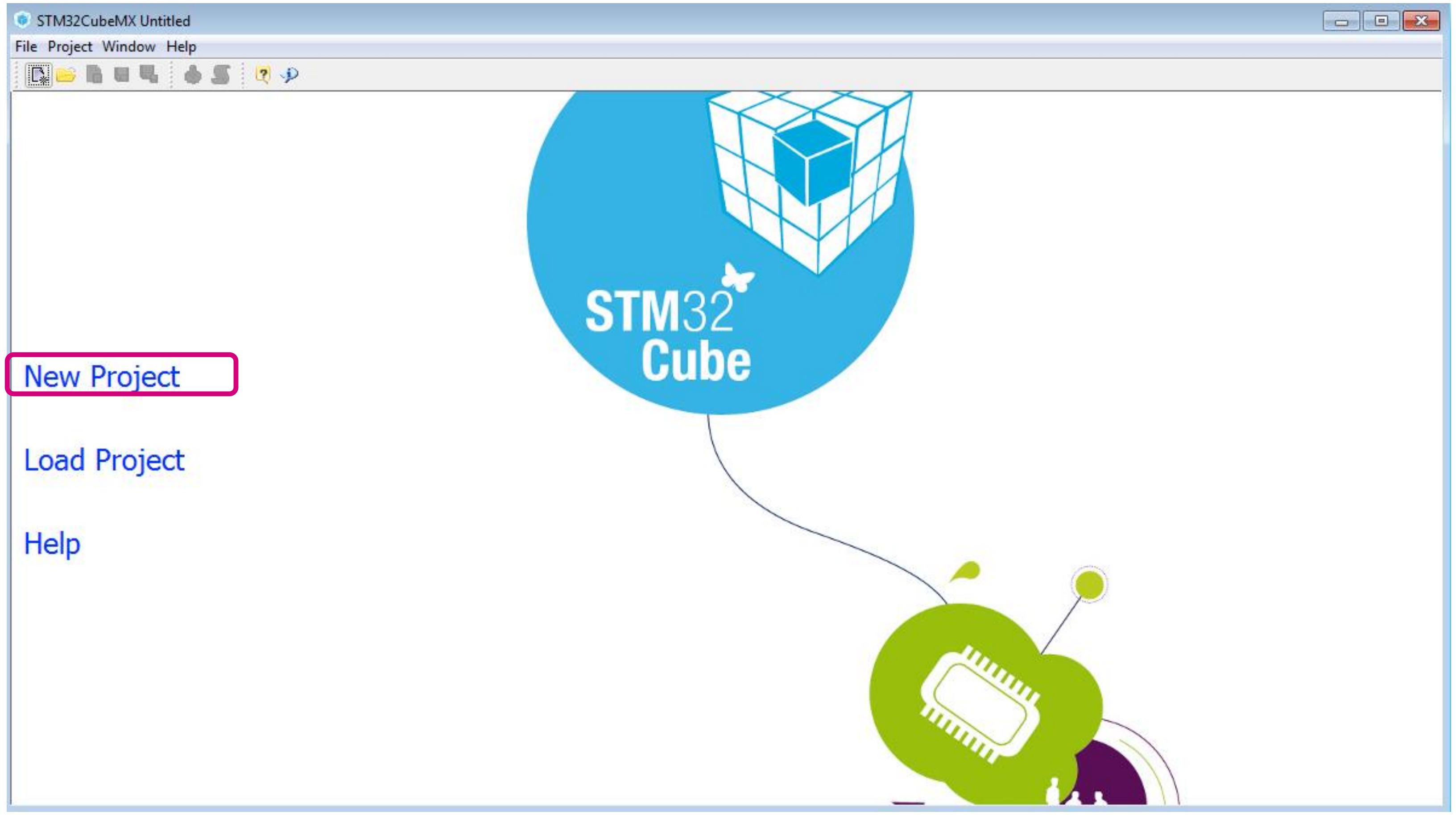
GPIO and EXTI Hands-on

- This hands-on describes how to use the GPIO HAL APIs. The User push button, configured as input with interrupt, will be used to change the states of the LEDs.
- STM32CubeMX will be used to generate the initialization codes for the EXTI, GPIO and System clock.
- This process will speed up the development as the initialization codes are generated by the STM32CubeMX tool. The user then will only need to add the user codes as per application.



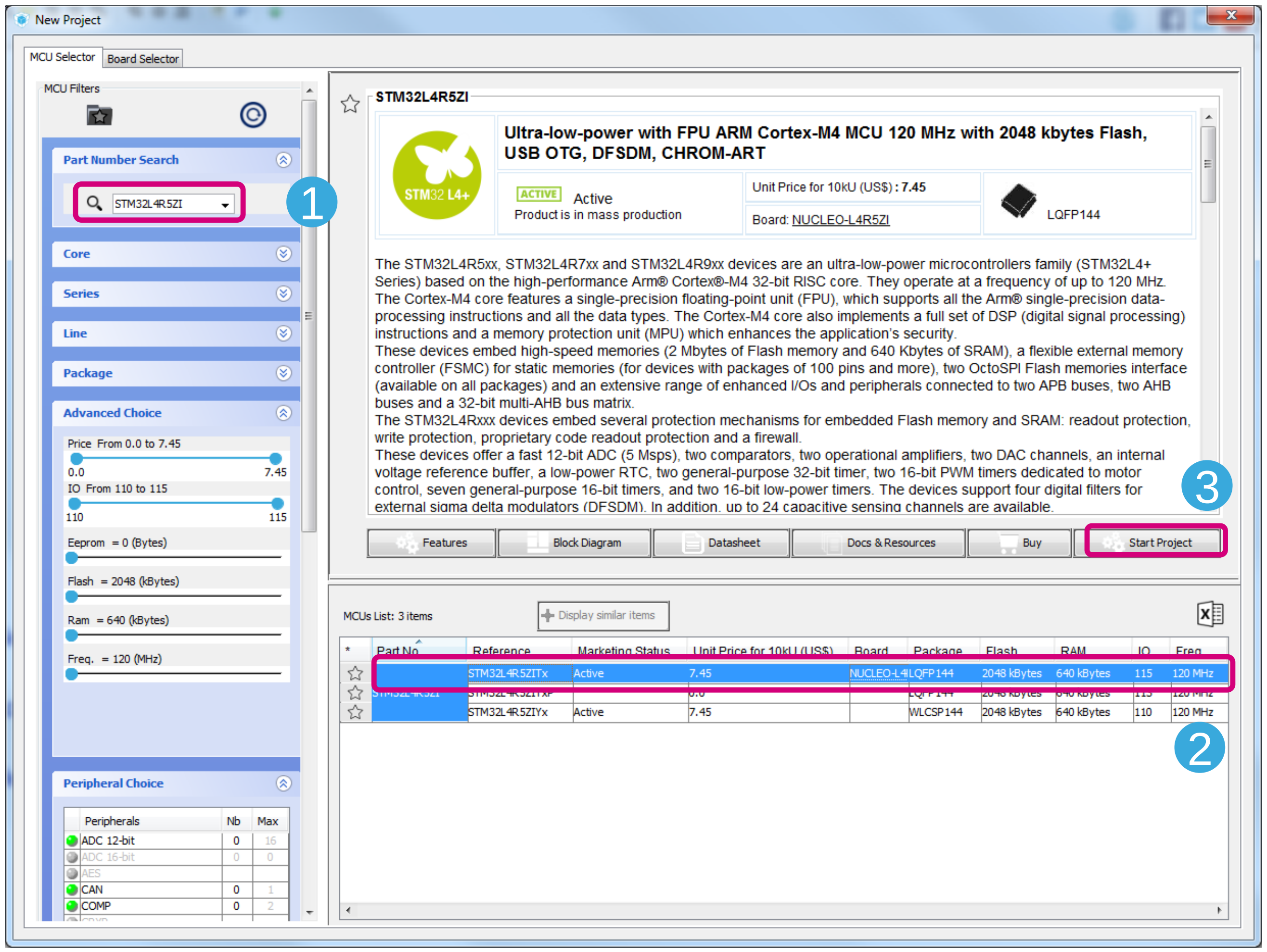
Step 1: Open New Project

- Click on **New Project**



Step 2: Select MCU

- Use [MCU Selector] to select STM32L4R5ZI device
- MCU Filter
  - Type “STM32L4R5ZI” in [Part Number Search]
- MCU List
  - Select [Part No.-> STM32L4R5ZI (LQFP144)]
  - Click [Start Project] or double click [STM32L4R5ZI] to continue



Peripherals	Nb	Max
ADC 12-bit	0	16
ADC 16-bit	0	0
AES		
CAN	0	1
COMP	0	2

Ultra-low-power with FPU ARM Cortex-M4 MCU 120 MHz with 2048 kbytes Flash, USB OTG, DFSDM, CHROM-ART

Active
Product is in mass production

Unit Price for 10kU (US$) : **7.45**

Board: NUCLEO-L4R5ZI

LQFP144

The STM32L4R5xx, STM32L4R7xx and STM32L4R9xx devices are an ultra-low-power microcontrollers family (STM32L4+ Series) based on the high-performance Arm® Cortex®-M4 32-bit RISC core. They operate at a frequency of up to 120 MHz.
The Cortex-M4 core features a single-precision floating-point unit (FPU), which supports all the Arm® single-precision data-processing instructions and all the data types. The Cortex-M4 core also implements a full set of DSP (digital signal processing) instructions and a memory protection unit (MPU) which enhances the application's security.
These devices embed high-speed memories (2 Mbytes of Flash memory and 640 Kbytes of SRAM), a flexible external memory controller (FSMC) for static memories (for devices with packages of 100 pins and more), two OctoSPI Flash memories interface (available on all packages) and an extensive range of enhanced I/Os and peripherals connected to two APB buses, two AHB buses and a 32-bit multi-AHB bus matrix.
The STM32L4Rxxx devices embed several protection mechanisms for embedded Flash memory and SRAM: readout protection, write protection, proprietary code readout protection and a firewall.
These devices offer a fast 12-bit ADC (5 Msps), two comparators, two operational amplifiers, two DAC channels, an internal voltage reference buffer, a low-power RTC, two general-purpose 32-bit timer, two 16-bit PWM timers dedicated to motor control, seven general-purpose 16-bit timers, and two 16-bit low-power timers. The devices support four digital filters for external sigma delta modulators (DFSDM). In addition, up to 24 capacitive sensing channels are available.

MCUs List: 3 items

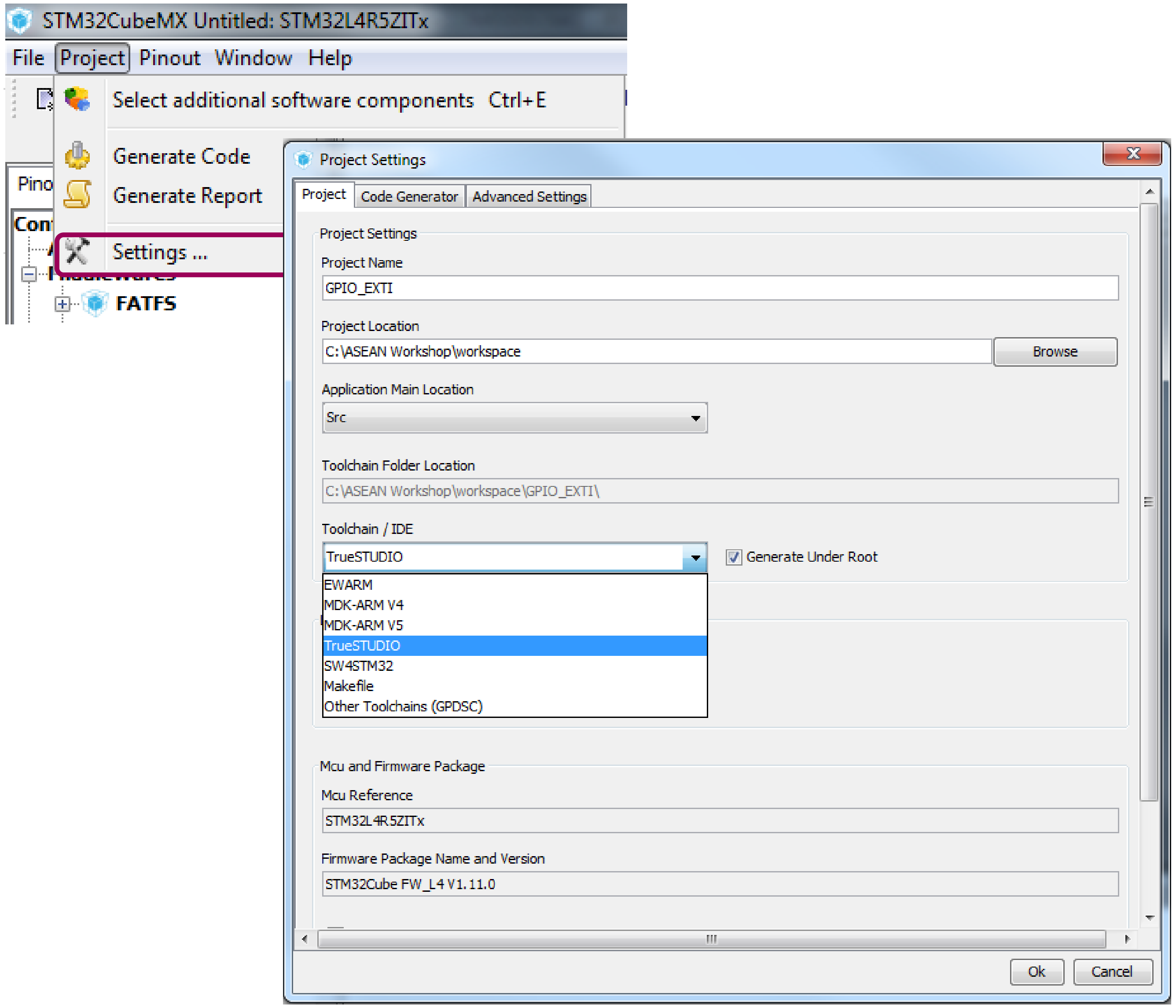
*	Part No	Reference	Marketing Status	Unit Price for 10kU (US$)	Board	Package	Flash	RAM	IO	Freq
☆	STM32L4R5ZI	STM32L4R5ZITx	Active	7.45	NUCLEO-L4	LQFP144	2048 kBytes	640 kBytes	115	120 MHz
☆		STM32L4R5ZITxP		0.0		LQFP144	2048 kBytes	640 kBytes	115	120 MHz
☆		STM32L4R5ZIYx	Active	7.45		WLCSP144	2048 kBytes	640 kBytes	110	120 MHz

life.augmented

Step 3a: Project Settings

Configure project settings

- Select [Project->Settings…]
- [Project] tab
    - [Project Name] : Any name. For example GPIO_EXTI
    - [Project Location] : Location to store project folders. In the case of Atollic TrueSTUDIO, the workspace folder location. For example C:\ASEAN Workshop\workspace
    - [Application Main Location] : Src
    - [Toolchain Folder Location] : Will automatically be generated
    - [Toolchain / IDE] : TrueSTUDIO
    - [Generate Under Root] : Checked
- [Code Generator] tab
    - Keep default configuration
- Click [OK] to finish



Project Settings

Project Name: GPIO_EXTI

Project Location: C:\ASEAN Workshop\workspace

Application Main Location: Src

Toolchain Folder Location: C:\ASEAN Workshop\workspace\GPIO_EXTI\

Toolchain / IDE: TrueSTUDIO

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

[✓] Generate Under Root

Mcu and Firmware Package

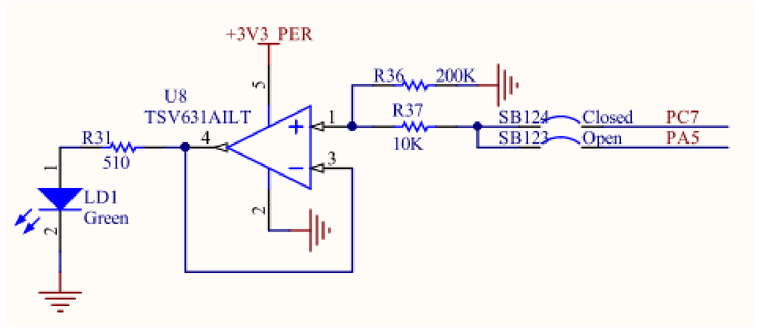
Mcu Reference: STM32L4R5ZITx

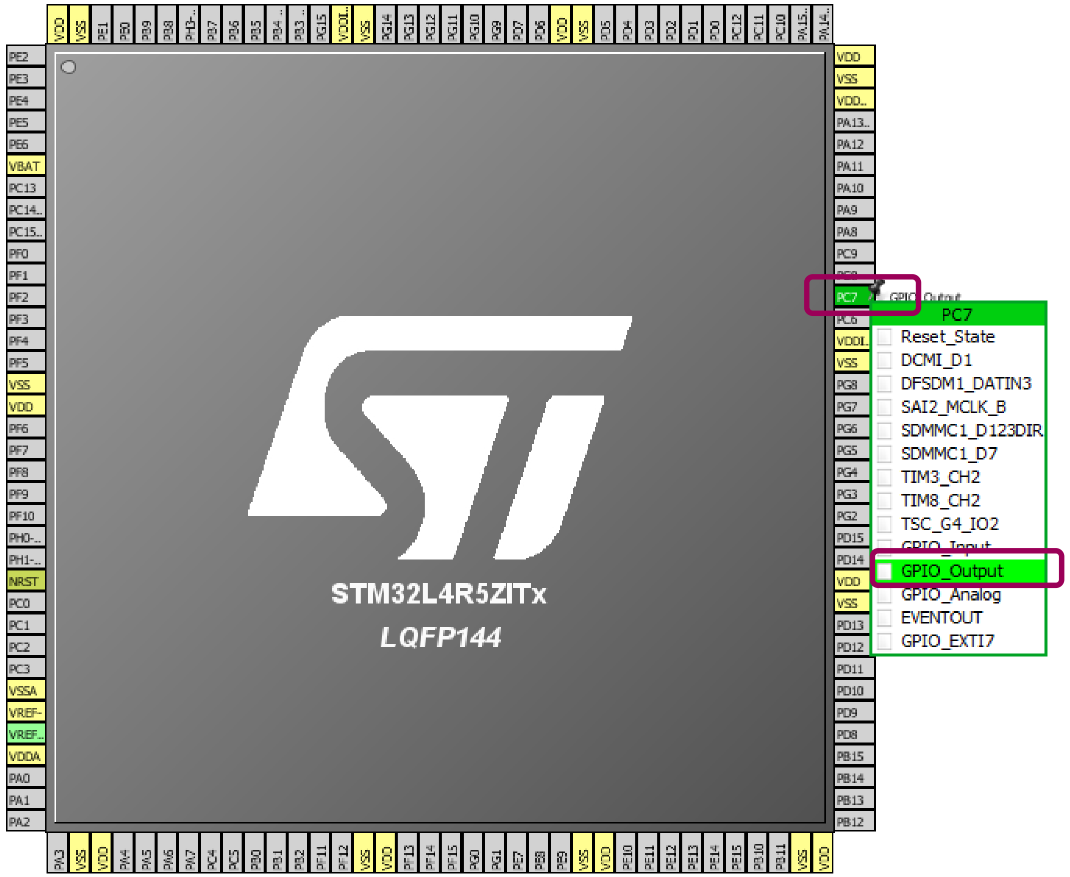
Firmware Package Name and Version: STM32Cube FW_L4 V1.11.0

Step 4: Configure GPIO

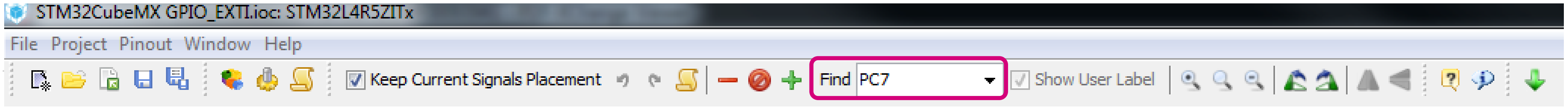
[Pinout] tab

- Left-click pin PC7 and set to [GPIO_Output] mode
- Note : Drive LED
  - Turn OFF – GPIO is LOW
  - Turn ON – GPIO is HIGH





Hint – Pin PC7 can also be found by using [Find] feature in STM32CubeMx

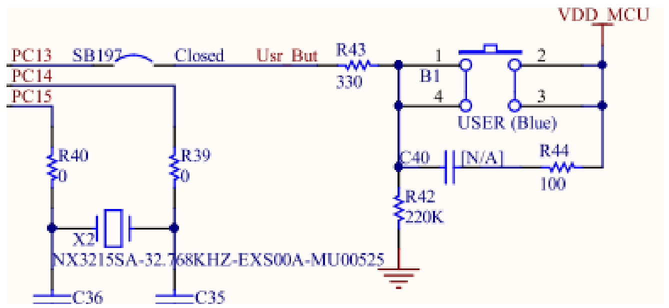


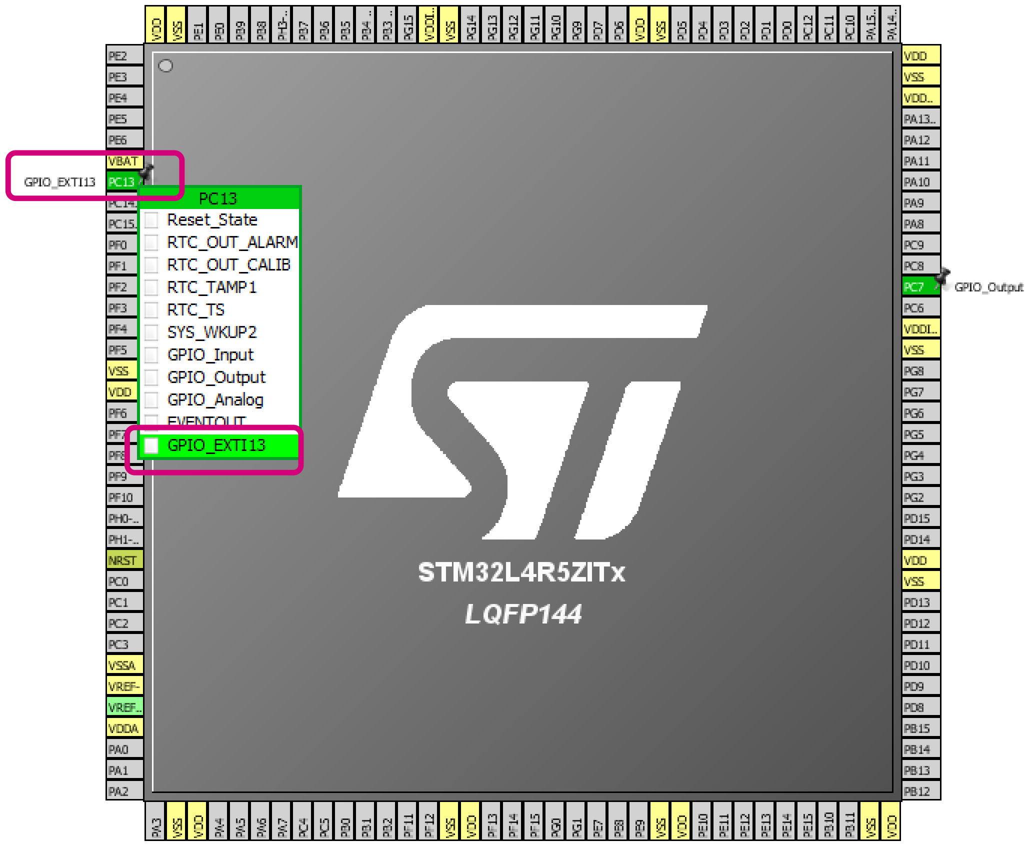
STM32CubeMX GPIO_EXTI.ioc: STM32L4R5ZITx
File Project Pinout Window Help
Keep Current Signals Placement
Find PC7
Show User Label

Step 4: Configure GPIO

[Pinout] tab

- Left-click pin PC13 and set to [GPIO_EXTI13] mode
- Note : USER button (Blue)
  - Button not press – GPIO is LOW
  - Button press – GPIO is HIGH



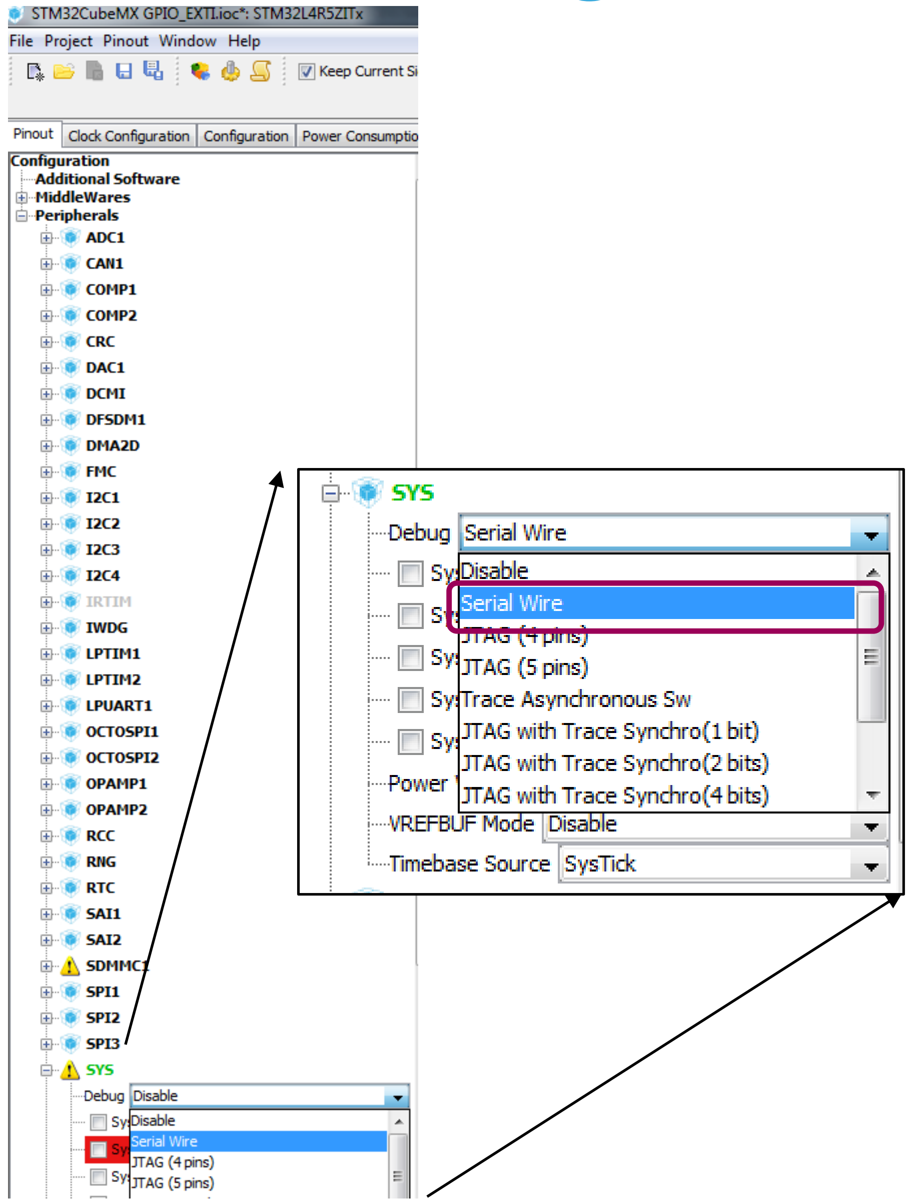


Step 5: Enable Debug Pins

[Pinout] tab

- Select [Configuration >Peripherals] tree, expand the [SYS] sub-tree
- Set the [Debug] to “Serial Wire”

Although the SWD debug pins are active after reset, it is a good practice to make sure the debug pins are reserved for debug purposes while assigning pins for your application. This avoids assigning it for other alternate function by mistake during firmware development stage



STM32CubeMX GPIO_EXTI.ioc*: STM32L4R5ZITx

File Project Pinout Window Help

Keep Current Si

Pinout | Clock Configuration | Configuration | Power Consumptio

Configuration
- Additional Software
- MiddleWares
- Peripherals
  - ADC1
  - CAN1
  - COMP1
  - COMP2
  - CRC
  - DAC1
  - DCMI
  - DFSDM1
  - DMA2D
  - FMC
  - I2C1
  - I2C2
  - I2C3
  - I2C4
  - IRTIM
  - IWDG
  - LPTIM1
  - LPTIM2
  - LPUART1
  - OCTOSPI1
  - OCTOSPI2
  - OPAMP1
  - OPAMP2
  - RCC
  - RNG
  - RTC
  - SAI1
  - SAI2
  - SDMMC1
  - SPI1
  - SPI2
  - SPI3
  - SYS
    - Debug: Disable
      - Disable
      - Serial Wire
      - JTAG (4 pins)
      - JTAG (5 pins)

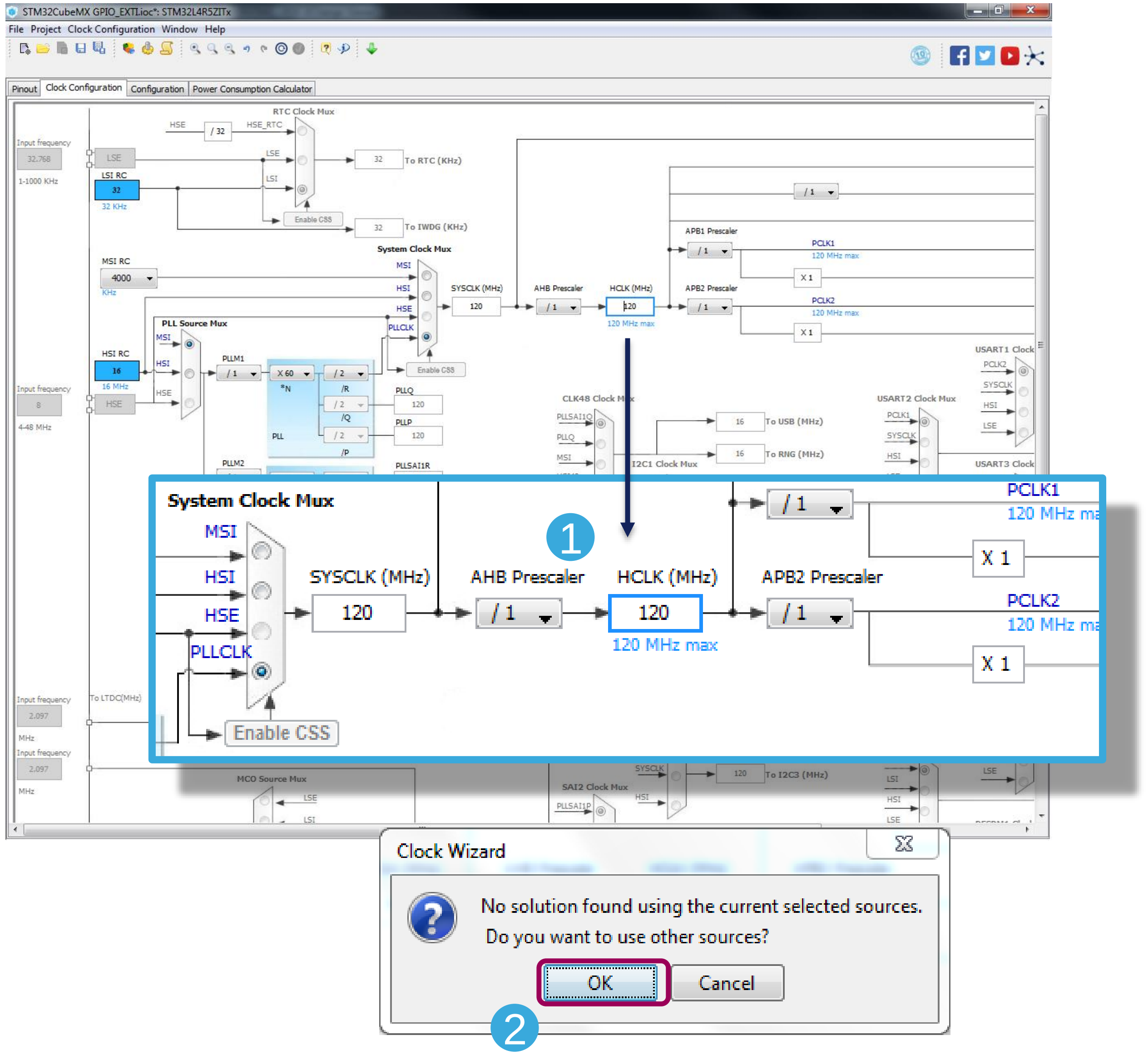
SYS
- Debug: Serial Wire
  - Disable
  - Serial Wire
  - JTAG (4 pins)
  - JTAG (5 pins)
  - Trace Asynchronous Sw
  - JTAG with Trace Synchro(1 bit)
  - JTAG with Trace Synchro(2 bits)
  - JTAG with Trace Synchro(4 bits)
- Power
- VREFBUF Mode: Disable
- Timebase Source: SysTick

life.augmented

Step 6: Clock Configuration

[Clock Configuration] tab

- Set [HCLK (MHz)] to 120
- Click OK when [Clock Wizard] message pop out to automatically find the correct clock sources
- The appropriate clock source and PLL values will be set automatically



Clock Wizard

No solution found using the current selected sources.
Do you want to use other sources?

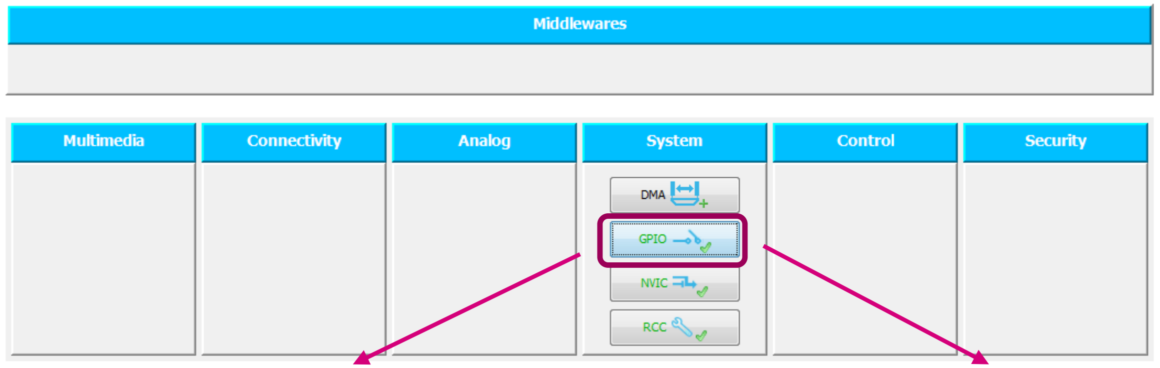
Step 7: Peripheral Configuration

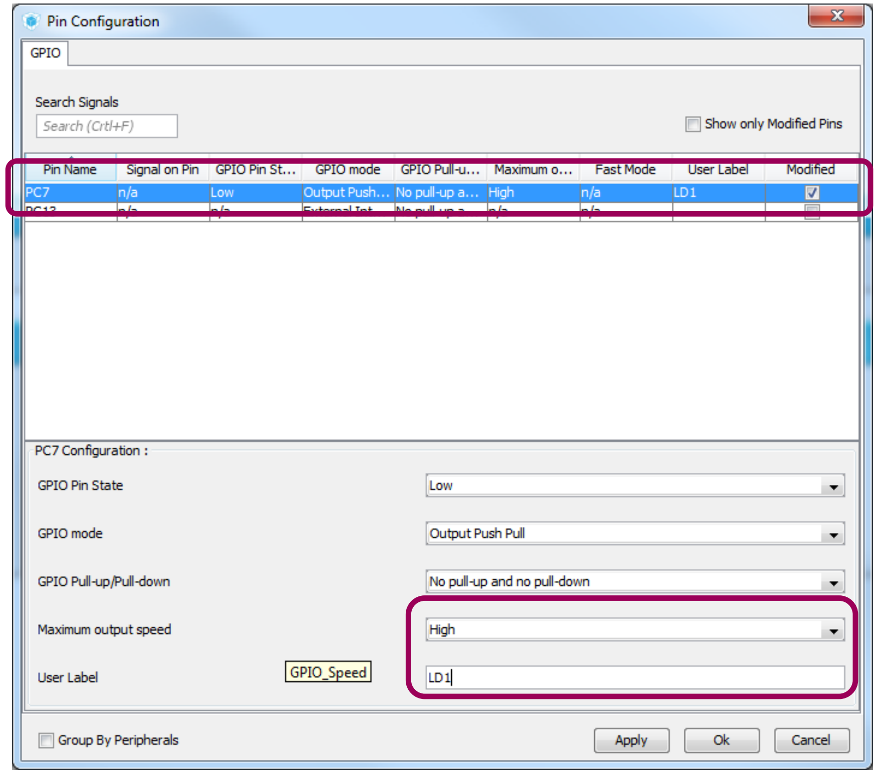
[Configuration] tab

- Select [GPIO]
  - Configure PC7
    - [GPIO Mode] : Output Push Pull
    - [Maximum output speed] : High
    - [User label] : Any name (optional)
    - Other settings use default
  - Configure PC13
    - [GPIO Mode] : External Interrupt Mode with Rising edge trigger detection
    - [User label] : Any name (optional)
    - Other settings use default
  - Click [Apply] and [OK]

Note: Refer to next slide for picture of configuration

Step 7: Peripheral Configuration





Pin Configuration

GPIO

Search Signals

Search (Crtl+F)

☐ Show only Modified Pins

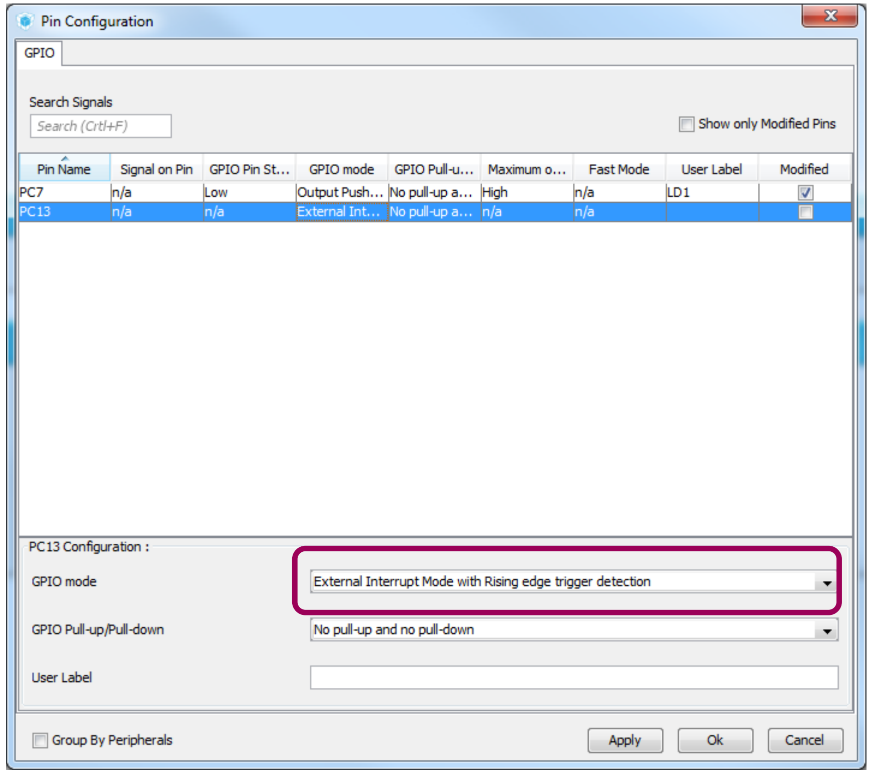
Pin Name	Signal on Pin	GPIO Pin St…	GPIO mode	GPIO Pull-u…	Maximum o…	Fast Mode	User Label	Modified
PC7	n/a	Low	Output Push…	No pull-up a…	High	n/a	LD1	✓
PC13	n/a	n/a	External Int…	No pull-up a…	n/a	n/a		☐

PC7 Configuration :

GPIO Pin State	Low
GPIO mode	Output Push Pull
GPIO Pull-up/Pull-down	No pull-up and no pull-down
Maximum output speed	High
User Label	LD1

GPIO_Speed

☐ Group By Peripherals



Pin Configuration

GPIO

Search Signals

Search (Crtl+F)

☐ Show only Modified Pins

Pin Name	Signal on Pin	GPIO Pin St…	GPIO mode	GPIO Pull-u…	Maximum o…	Fast Mode	User Label	Modified
PC7	n/a	Low	Output Push…	No pull-up a…	High	n/a	LD1	✓
PC13	n/a	n/a	External Int…	No pull-up a…	n/a	n/a		☐

PC13 Configuration :

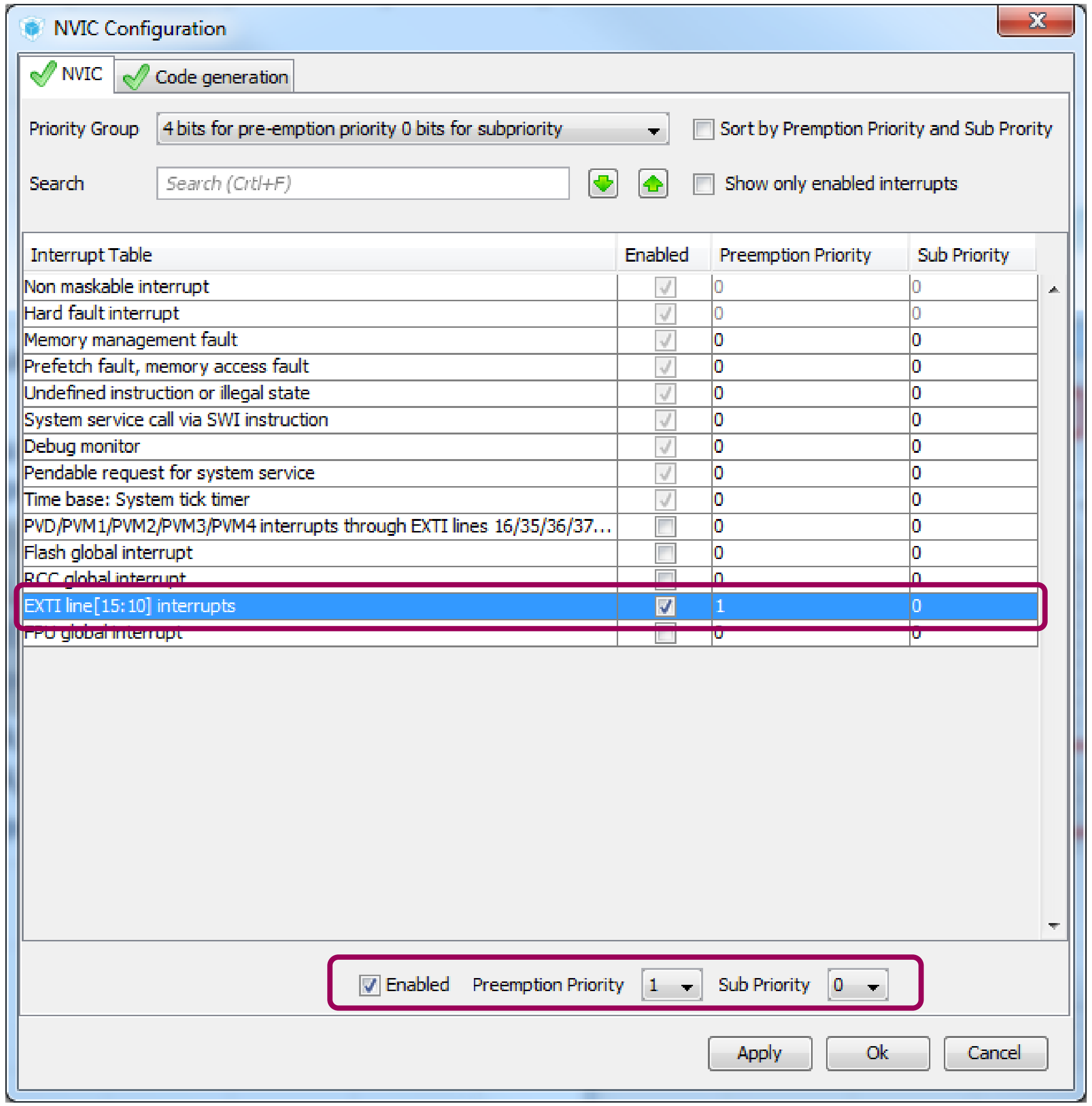
GPIO mode	External Interrupt Mode with Rising edge trigger detection
GPIO Pull-up/Pull-down	No pull-up and no pull-down
User Label	

☐ Group By Peripherals

life.augmented

Step 7: Peripheral Configuration

- NVIC
  - EXTI line [15:10] interrupt (PC13 B1 USER)
    - Enable
    - Preemption Priority: 1
- Caution!!!
  - HAL_Delay() function provides accurate delay (in milliseconds) based on variable incremented in System Tick Timer(SysTick) ISR.
  - If HAL_Delay() is called from a peripheral ISR process, then the SysTick interrupt must have higher priority (numerically lower) than the peripheral interrupt. Otherwise the caller ISR process will be blocked.



Priority Group: 4 bits for pre-emption priority 0 bits for subpriority

☐ Sort by Premption Priority and Sub Prority

Search: Search (Crtl+F)

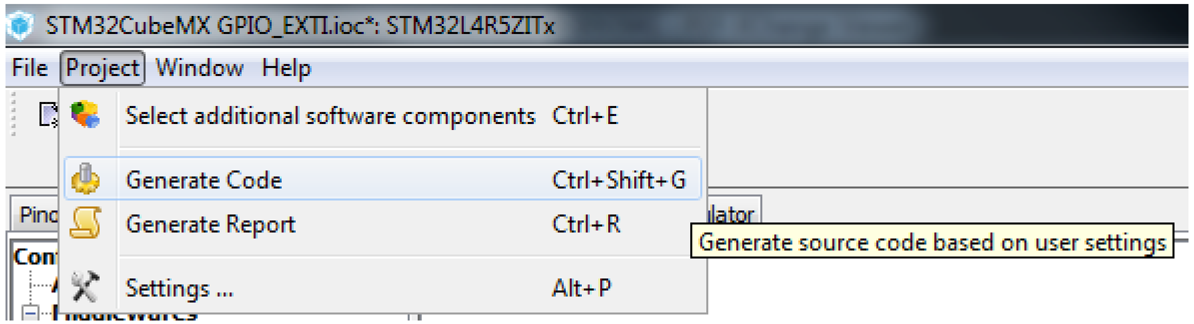
☐ Show only enabled interrupts

Interrupt Table	Enabled	Preemption Priority	Sub Priority
Non maskable interrupt	✓	0	0
Hard fault interrupt	✓	0	0
Memory management fault	✓	0	0
Prefetch fault, memory access fault	✓	0	0
Undefined instruction or illegal state	✓	0	0
System service call via SWI instruction	✓	0	0
Debug monitor	✓	0	0
Pendable request for system service	✓	0	0
Time base: System tick timer	✓	0	0
PVD/PVM1/PVM2/PVM3/PVM4 interrupts through EXTI lines 16/35/36/37...	☐	0	0
Flash global interrupt	☐	0	0
RCC global interrupt	☐	0	0
EXTI line[15:10] interrupts	✓	1	0
FPU global interrupt	☐	0	0

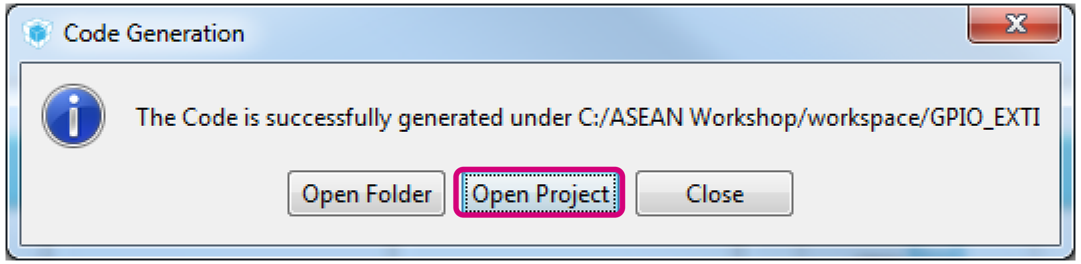
☑ Enabled Preemption Priority: 1 Sub Priority: 0

Step 8: Generate Code

- Select [Project->Generate Code]


STM32CubeMX GPIO_EXTI.ioc*: STM32L4R5ZITx
File
Project
Window
Help
Select additional software components Ctrl+E
Generate Code Ctrl+Shift+G
Generate Report Ctrl+R
Settings ... Alt+P
Generate source code based on user settings

- Select [Open Project] to launch TrueSTUDIO and import project into TrueSTUDIO automatically



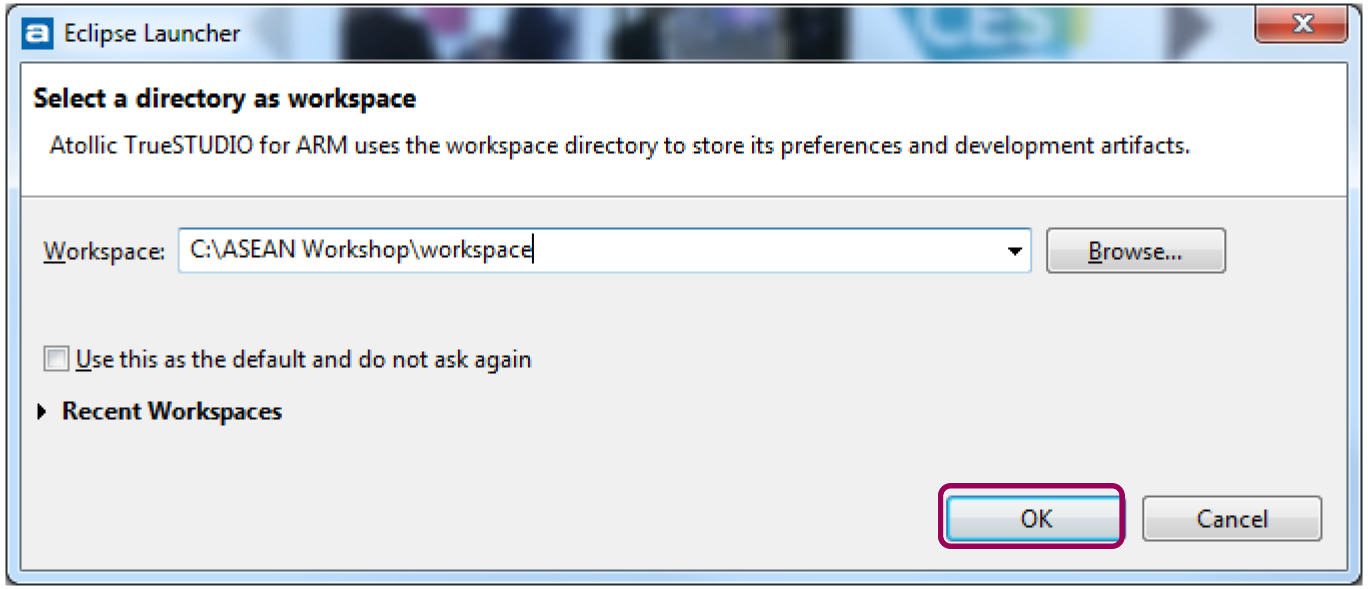
The Code is successfully generated under C:/ASEAN Workshop/workspace/GPIO_EXTI

- Selecting [Open Folder] will open folder containing STM32CubeMx generated code.
  - You will need to start TrueSTUDIO and import project manually

Starting TrueSTUDIO

- Starting TrueSTUDIO will start the [Eclipse Launcher]
- Select the workspace path:
  - For example C:\ASEAN Workshop\workspace

  as specified in the STM32CubeMX Project Setting i.e. [Project Location]
- Click [OK] to proceed



Select a directory as workspace

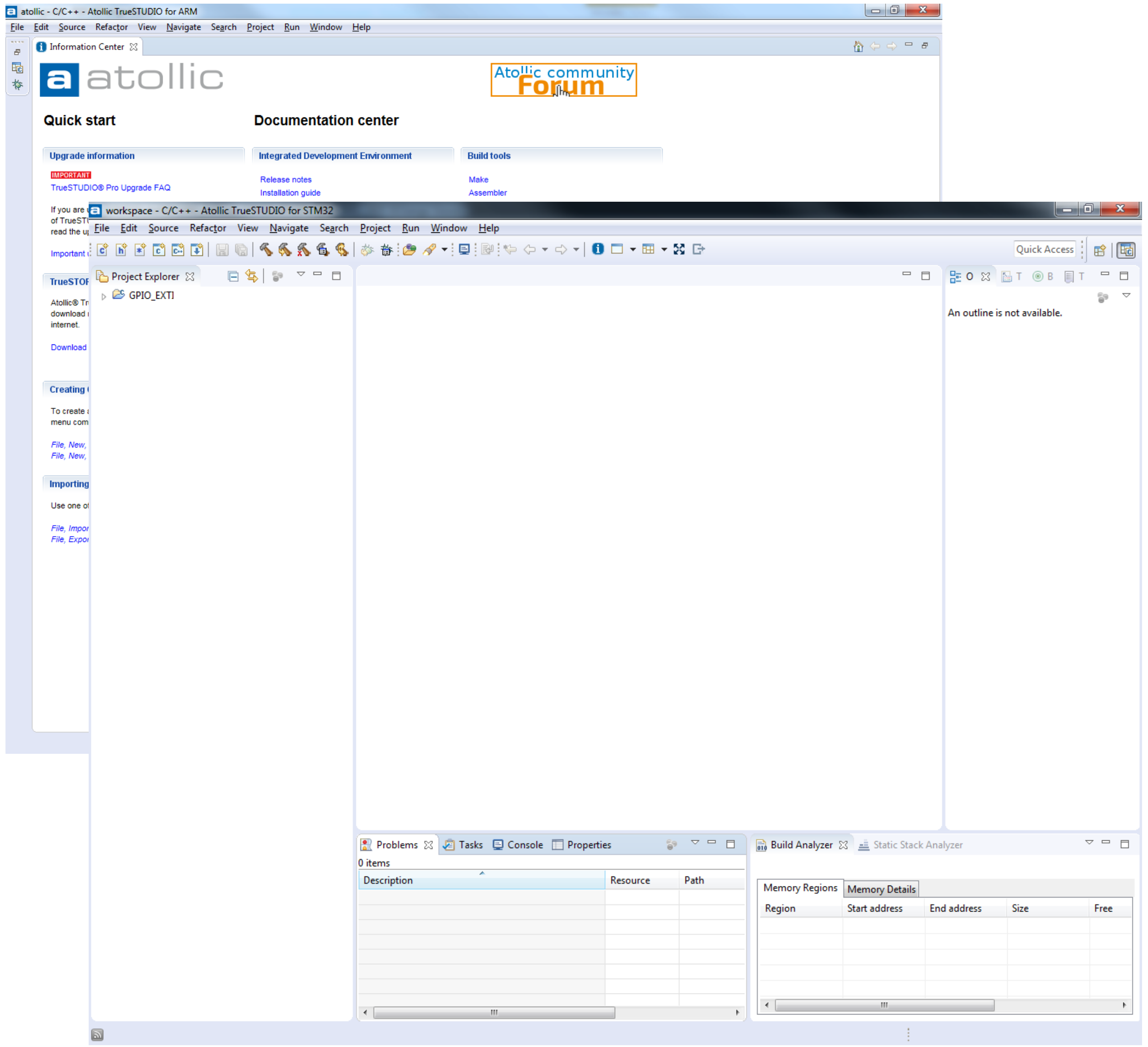
Atollic TrueSTUDIO for ARM uses the workspace directory to store its preferences and development artifacts.

Workspace: C:\ASEAN Workshop\workspace

First TrueSTUDIO start

- [Information Center] panel contains links to various information related to TrueSTUDIO.
  - Close the [Information Center] panel
  - C/C++ perspective will appear

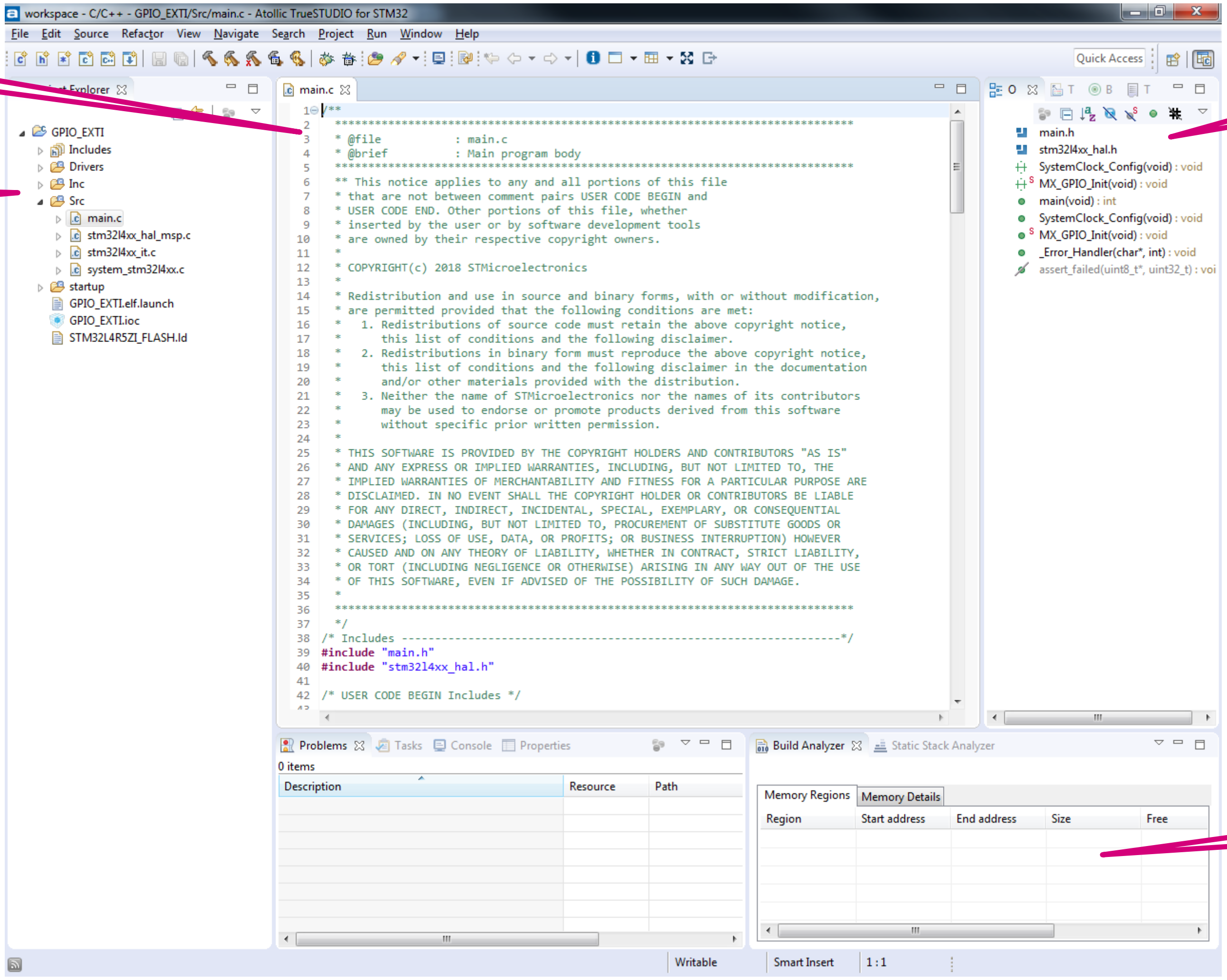
Please note content of [Information Center] panel might take some time to appear. Be patient.



An outline is not available.

0 items

C/C++ Perspective



```
/**
  ******************************************************************************
  * @file           : main.c
  * @brief          : Main program body
  ******************************************************************************
  ** This notice applies to any and all portions of this file
  * that are not between comment pairs USER CODE BEGIN and
  * USER CODE END. Other portions of this file, whether
  * inserted by the user or by software development tools
  * are owned by their respective copyright owners.
  *
  * COPYRIGHT(c) 2018 STMicroelectronics
  *
  * Redistribution and use in source and binary forms, with or without modification,
  * are permitted provided that the following conditions are met:
  *   1. Redistributions of source code must retain the above copyright notice,
  *      this list of conditions and the following disclaimer.
  *   2. Redistributions in binary form must reproduce the above copyright notice,
  *      this list of conditions and the following disclaimer in the documentation
  *      and/or other materials provided with the distribution.
  *   3. Neither the name of STMicroelectronics nor the names of its contributors
  *      may be used to endorse or promote products derived from this software
  *      without specific prior written permission.
  *
  * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
  * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
  * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
  * DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
  * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
  * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
  * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
  * CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
  * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
  * OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
  *
  ******************************************************************************
  */
/* Includes ------------------------------------------------------------------*/
#include "main.h"
#include "stm32l4xx_hal.h"

/* USER CODE BEGIN Includes */
```

life.augmented

Eclipse definition

- **Workspace**
  - Container that include project folders and information about project
  - **Project** is a directory containing files that may be organized in sub-directories
  - Can contain multiple projects and be located anywhere in the storage media
- **Perspective**
  - Set of windows/views dedicated to a purpose
  - Typically used perspective - **C/C++** and **Debug**
- **View**
  - Dedicated windows for specific purpose
  - By default not all views are available in a perspective

Modifying generated code

- Add the following code to main.c

```
/* USER CODE BEGIN 0 */
uint8_t MODE_SELECTION;
/* USER CODE END 0 */
```

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

  /* USER CODE BEGIN 3 */
      if (MODE_SELECTION == 0) {
             /* Toggle LEDs - Use the HAL functions from stm32l4xx_hal_gpio.c file */
             HAL_GPIO_TogglePin(GPIOC, GPIO_PIN_7);    //LD1 (green) – PC7
             HAL_Delay(100); //100ms
      } else if (MODE_SELECTION == 1) {
             /* Turn OFF the LEDs - Use the HAL functions from stm32l4xx_hal_gpio.c file
             */
             HAL_GPIO_WritePin(GPIOC, GPIO_PIN_7, GPIO_PIN_RESET); //Turn off LD1 (green)
             HAL_Delay(100); //100ms
      } else if (MODE_SELECTION == 2) {
             /* Turn ON the LED - Use the HAL functions from stm32l4xx_hal_gpio.c file */
             HAL_GPIO_WritePin(GPIOC, GPIO_PIN_7, GPIO_PIN_SET);//LD1  (green) – PC7
             HAL_Delay(1000);//1secs
      }
}
  /* USER CODE END 3 */
```

life.augmented

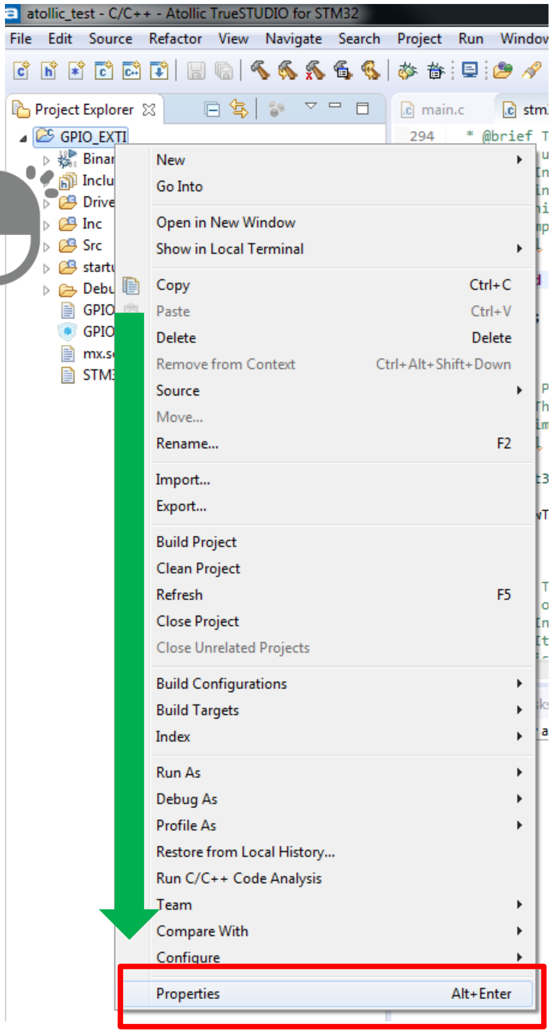
Modifying generated code

```
/* USER CODE BEGIN 4 */
/**
  * @brief EXTI line detection callback. The function will be call by EXTI15_10_IRQHandler in “stm32l4xx_it.c” .
  * @param GPIO_Pin: Specifies the pins connected EXTI line
  * @retval None
  */
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
  if(GPIO_Pin == GPIO_PIN_13)
  {
    MODE_SELECTION++;
    if(MODE_SELECTION > 2) MODE_SELECTION=0;
    /* Debounce - wait until the button is released . Read the GPIO to get the state. Refer to the schematics. */
    /* - Use the HAL functions from stm32l4xx_hal_gpio.c file */
    while(HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) != GPIO_PIN_RESET);//Blue pushbutton – PC13
  }

}
/* USER CODE END 4 */
```

life.augmented

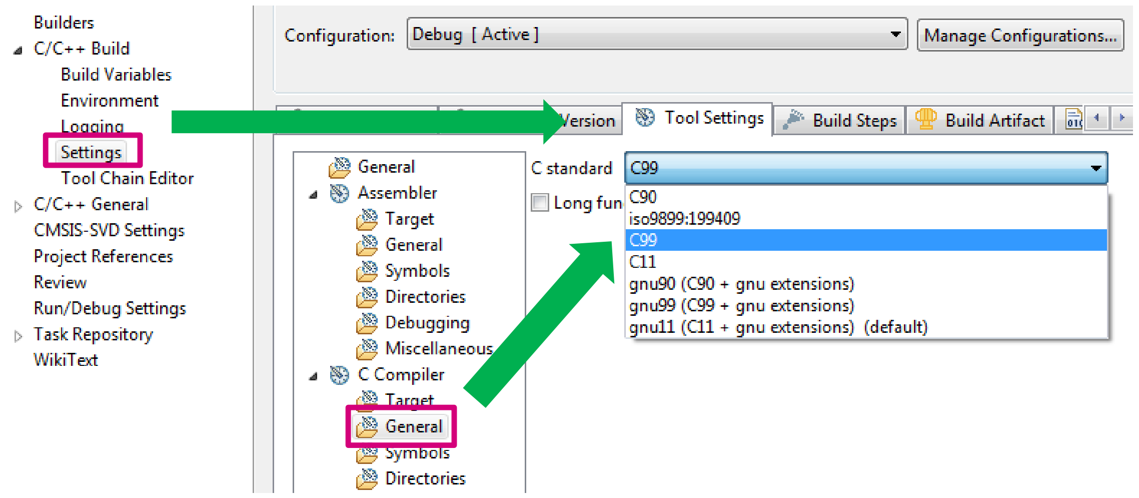
Tips: Useful project settings in TrueSTUDIO

C dialect and parallel build



C/C++ Build->Settings->Tools Settings->C Compiler->General->C standard

1. Configure C standard to **C99** to avoid possible compilation errors

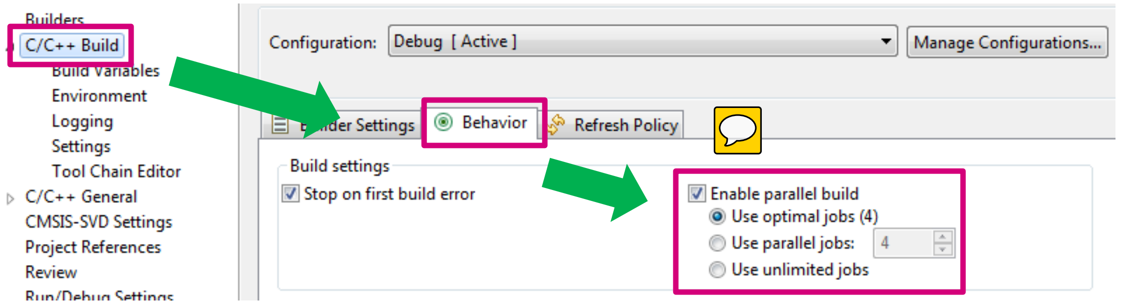


Configuration: Debug [Active]

C standard: C99

C/C++ Build->Behavior tab

2. **Enable parallel build** to make use of your machine potential and to shorten compilation time



Configuration: Debug [Active]

[✓] Stop on first build error

[✓] Enable parallel build

[✓ Use optimal jobs (4)]

Use parallel jobs: 4

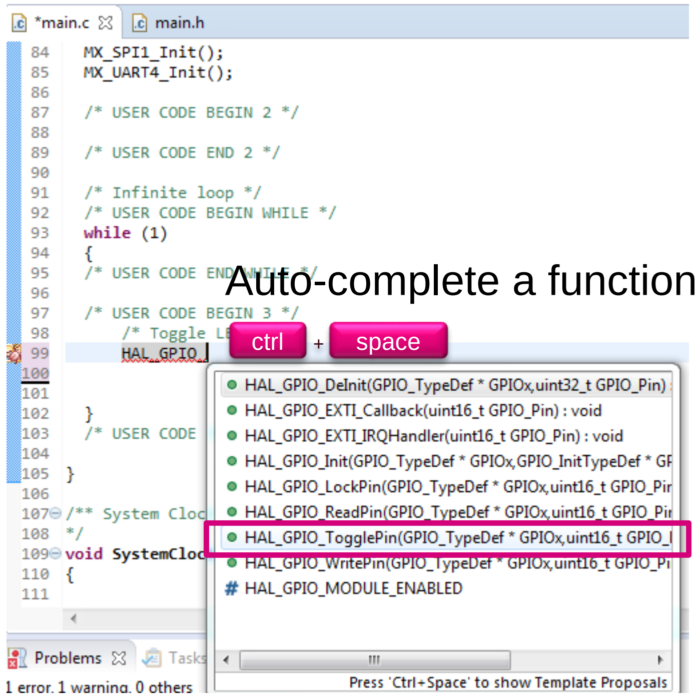
Use unlimited jobs

= It is a mouse !

life.augmented

Tips: Using Code completion

- You can complete a function or parameter by using **CTRL + SPACE** keys after a typing a few characters of the function or parameter.



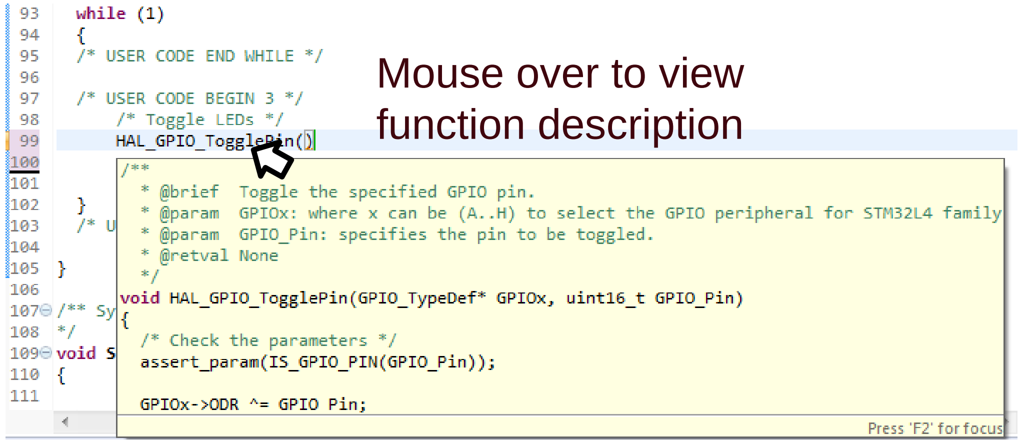
Auto-complete a function

```
  MX_SPI1_Init();
  MX_UART4_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 */
      /* Toggle LE
      HAL_GPIO_


  }
  /* USER CODE

}

/** System Cloc
*/
void SystemCloc
{
```

```
HAL_GPIO_DeInit(GPIO_TypeDef * GPIOx,uint32_t GPIO_Pin)
HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin) : void
HAL_GPIO_EXTI_IRQHandler(uint16_t GPIO_Pin) : void
HAL_GPIO_Init(GPIO_TypeDef * GPIOx,GPIO_InitTypeDef * GF
HAL_GPIO_LockPin(GPIO_TypeDef * GPIOx,uint16_t GPIO_Pir
HAL_GPIO_ReadPin(GPIO_TypeDef * GPIOx,uint16_t GPIO_Pir
HAL_GPIO_TogglePin(GPIO_TypeDef * GPIOx,uint16_t GPIO_l
HAL_GPIO_WritePin(GPIO_TypeDef * GPIOx,uint16_t GPIO_Pi
HAL_GPIO_MODULE_ENABLED
```

Press 'Ctrl+Space' to show Template Proposals

1 error, 1 warning, 0 others

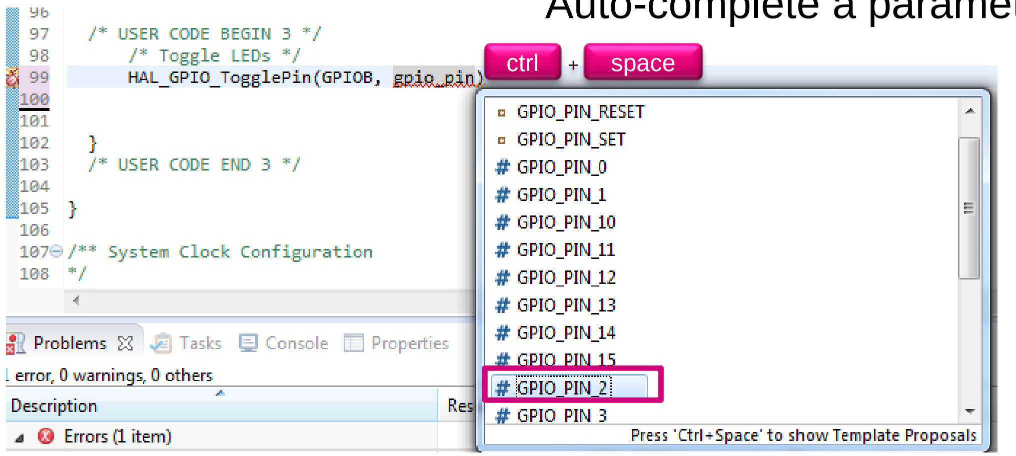


Mouse over to view function description

```
  while (1)
  {
  /* USER CODE END WHILE */

  /* USER CODE BEGIN 3 */
      /* Toggle LEDs */
      HAL_GPIO_TogglePin()
```

```
/**
  * @brief  Toggle the specified GPIO pin.
  * @param  GPIOx: where x can be (A..H) to select the GPIO peripheral for STM32L4 family
  * @param  GPIO_Pin: specifies the pin to be toggled.
  * @retval None
  */
void HAL_GPIO_TogglePin(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin)
{
  /* Check the parameters */
  assert_param(IS_GPIO_PIN(GPIO_Pin));

  GPIOx->ODR ^= GPIO_Pin;
```

Press 'F2' for focus



Auto-complete a parameter

```
  /* USER CODE BEGIN 3 */
      /* Toggle LEDs */
      HAL_GPIO_TogglePin(GPIOB, gpio_pin)


  }
  /* USER CODE END 3 */

}

/** System Clock Configuration
*/
```

```
GPIO_PIN_RESET
GPIO_PIN_SET
GPIO_PIN_0
GPIO_PIN_1
GPIO_PIN_10
GPIO_PIN_11
GPIO_PIN_12
GPIO_PIN_13
GPIO_PIN_14
GPIO_PIN_15
GPIO_PIN_2
GPIO_PIN_3
```

Press 'Ctrl+Space' to show Template Proposals

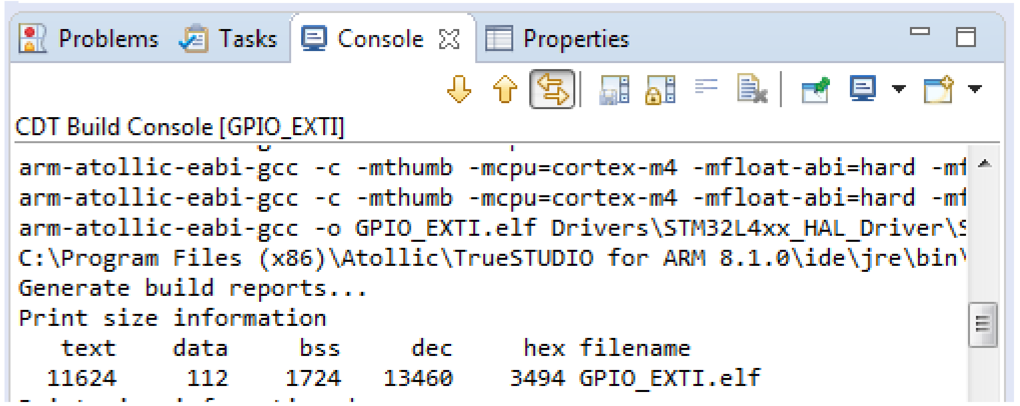
1 error, 0 warnings, 0 others

Errors (1 item)

life.augmented

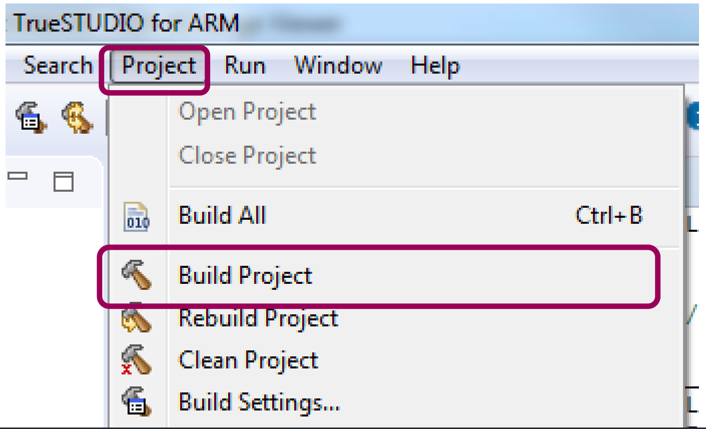
Build project

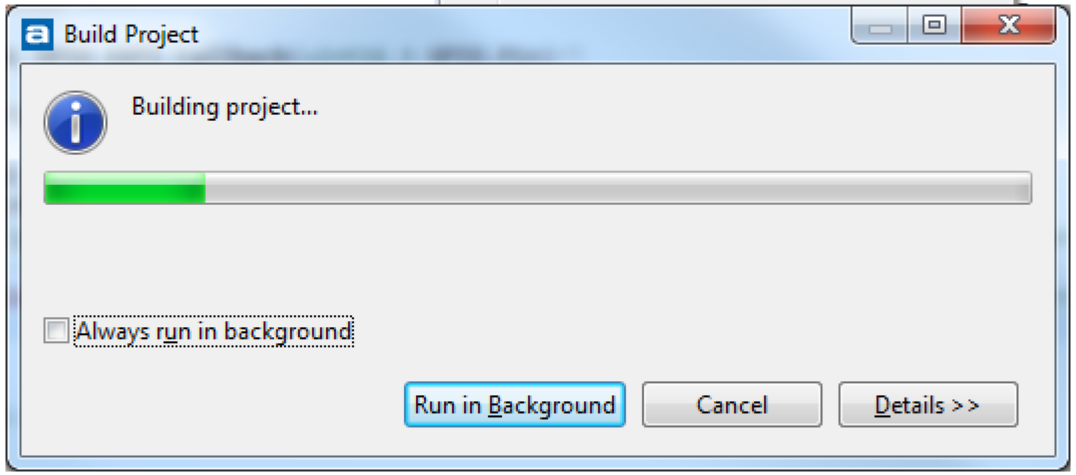
- Select [Project > Build Project] or [Project > Build All]
- Build result is displayed in the [Console] window



CDT Build Console [GPIO_EXTI]

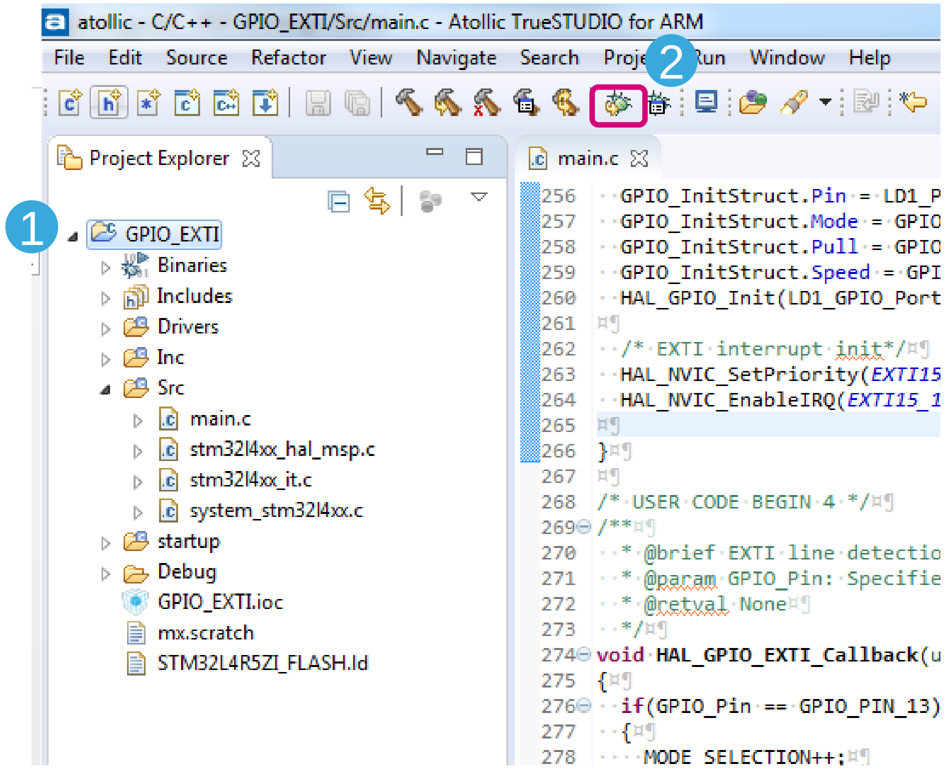
```
arm-atollic-eabi-gcc -c -mthumb -mcpu=cortex-m4 -mfloat-abi=hard -mf
arm-atollic-eabi-gcc -c -mthumb -mcpu=cortex-m4 -mfloat-abi=hard -mf
arm-atollic-eabi-gcc -o GPIO_EXTI.elf Drivers\STM32L4xx_HAL_Driver\S
C:\Program Files (x86)\Atollic\TrueSTUDIO for ARM 8.1.0\ide\jre\bin\
Generate build reports...
Print size information
   text    data     bss     dec     hex filename
  11624     112    1724   13460    3494 GPIO_EXTI.elf
```





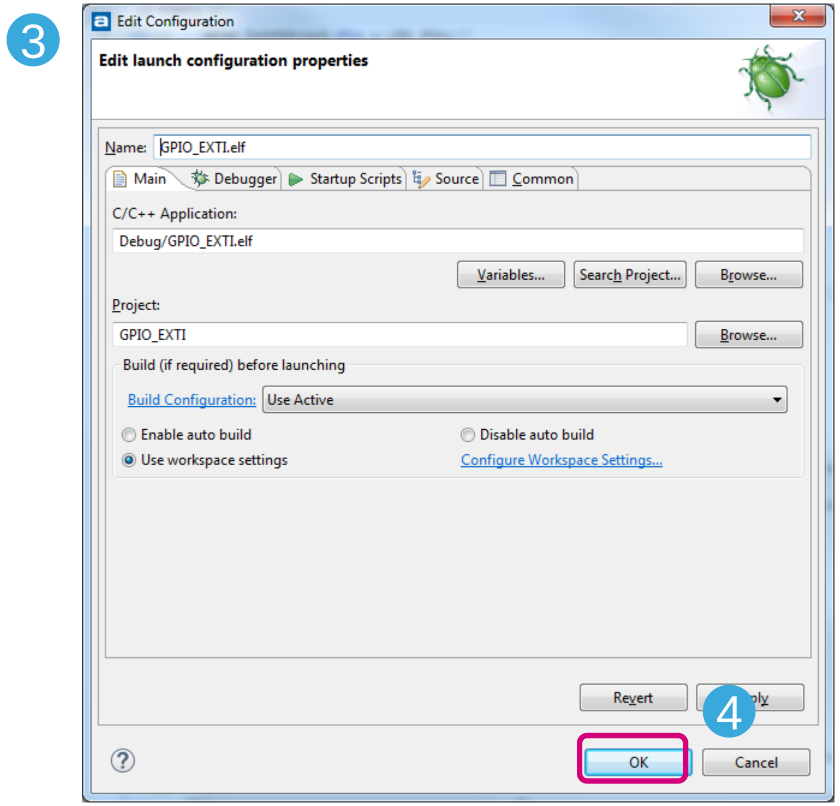
Starting the debugger

1. Select project in [Project Explorer] view
2. Click on [Debug] button ( ) or press **F11** to start debug session
3. [Debug Configuration] dialog box will appear when debugging project the first time
4. Click [OK] to accept default configuration


1
2
atollic - C/C++ - GPIO_EXTI/Src/main.c - Atollic TrueSTUDIO for ARM
File Edit Source Refactor View Navigate Search Project Run Window Help
Project Explorer
main.c

- GPIO_EXTI
  - Binaries
  - Includes
  - Drivers
  - Inc
  - Src
    - main.c
    - stm32l4xx_hal_msp.c
    - stm32l4xx_it.c
    - system_stm32l4xx.c
  - startup
  - Debug
  - GPIO_EXTI.ioc
  - mx.scratch
  - STM32L4R5ZI_FLASH.ld

```
  GPIO_InitStruct.Pin = LD1_P
  GPIO_InitStruct.Mode = GPIO
  GPIO_InitStruct.Pull = GPIO
  GPIO_InitStruct.Speed = GPI
  HAL_GPIO_Init(LD1_GPIO_Port

  /* EXTI interrupt init*/
  HAL_NVIC_SetPriority(EXTI15
  HAL_NVIC_EnableIRQ(EXTI15_1

}

/* USER CODE BEGIN 4 */
/**
  * @brief EXTI line detectio
  * @param GPIO_Pin: Specifie
  * @retval None
  */
void HAL_GPIO_EXTI_Callback(u
{
  if(GPIO_Pin == GPIO_PIN_13)
  {
    MODE_SELECTION++;
```



Edit launch configuration properties

Name: GPIO_EXTI.elf

C/C++ Application: Debug/GPIO_EXTI.elf

Project: GPIO_EXTI

Build (if required) before launching

Build Configuration: Use Active

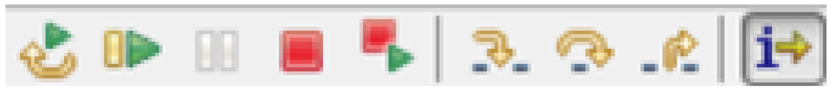
Enable auto build ○ Disable auto build

[✓ Use workspace settings] Configure Workspace Settings...

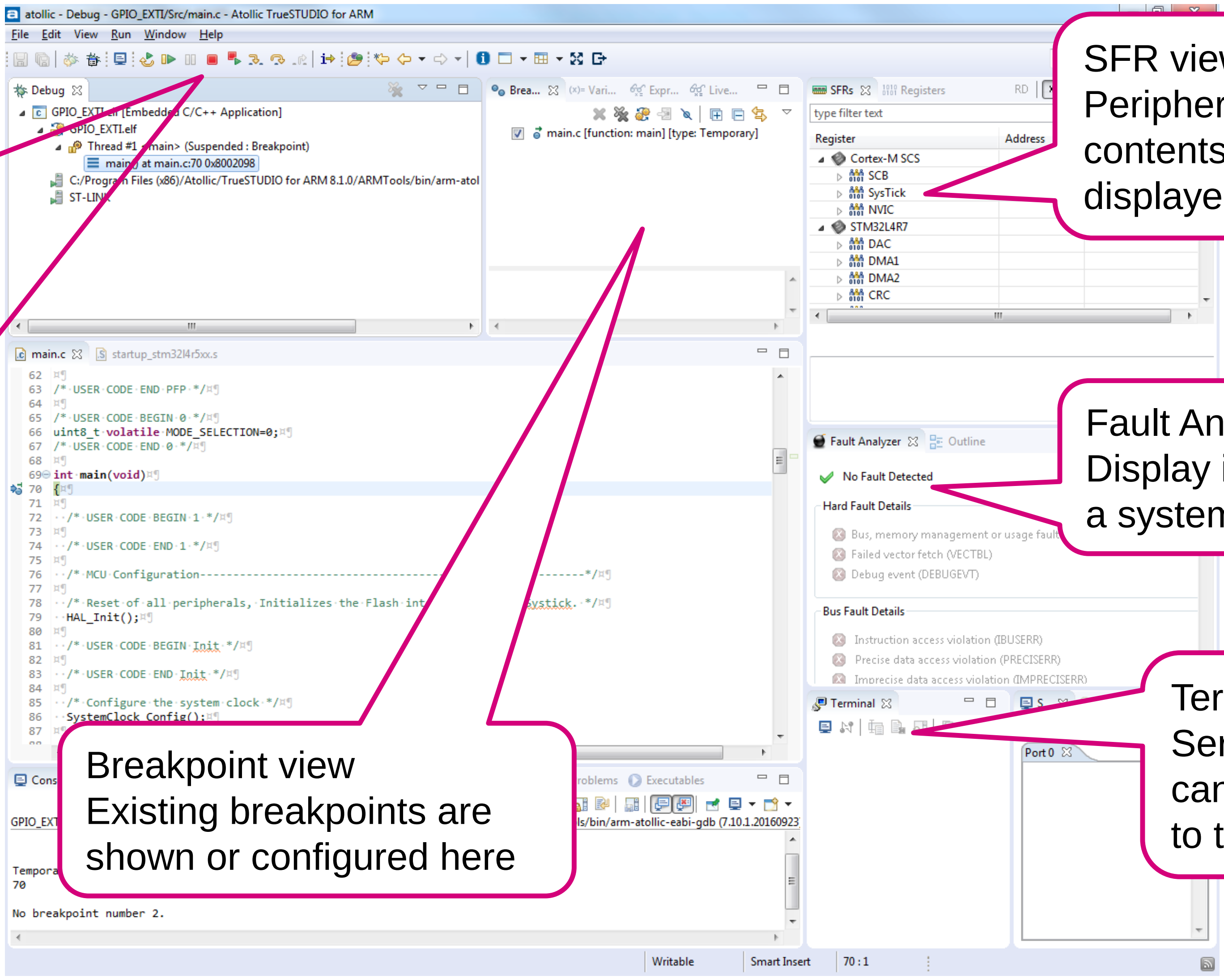
life.augmented

Debug perspective

Run control toolbar



1. Restart
2. Resume
3. Suspend
4. Terminate
5. Terminate & Relaunch
6. Step Into
7. Step Over
8. Step Return
9. Instruction Stepping mode



atollic - Debug - GPIO_EXTI/Src/main.c - Atollic TrueSTUDIO for ARM

```
GPIO_EXTI.elf [Embedded C/C++ Application]
  GPIO_EXTI.elf
    Thread #1 <main> (Suspended : Breakpoint)
      main() at main.c:70 0x8002098
  C:/Program Files (x86)/Atollic/TrueSTUDIO for ARM 8.1.0/ARMTools/bin/arm-atol
  ST-LINK
```

```
main.c [function: main] [type: Temporary]
```

Register	Address
Cortex-M SCS	
SCB	
SysTick	
NVIC	
STM32L4R7	
DAC	
DMA1	
DMA2	
CRC	

type filter text

```

/* USER CODE END PFP */

/* USER CODE BEGIN 0 */
uint8_t volatile MODE_SELECTION=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 int                 systick. */
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

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

```

Fault Analyzer — Outline

✓ No Fault Detected

Hard Fault Details
- Bus, memory management or usage fault
- Failed vector fetch (VECTBL)
- Debug event (DEBUGEVT)

Bus Fault Details
- Instruction access violation (IBUSERR)
- Precise data access violation (PRECISERR)
- Imprecise data access violation (IMPRECISERR)

```
GPIO_EXT                                  ls/bin/arm-atollic-eabi-gdb (7.10.1.20160923


Tempora
70

No breakpoint number 2.
```

SFR view
Peripheral registers contents are displayed here

Fault Analyzer view
Display information if a system fault occurs

Breakpoint view
Existing breakpoints are shown or configured here

Terminal view
Serial port output can be directed to this view

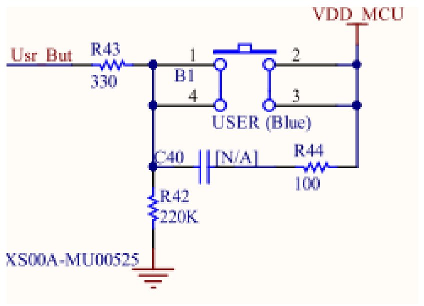
life.augmented

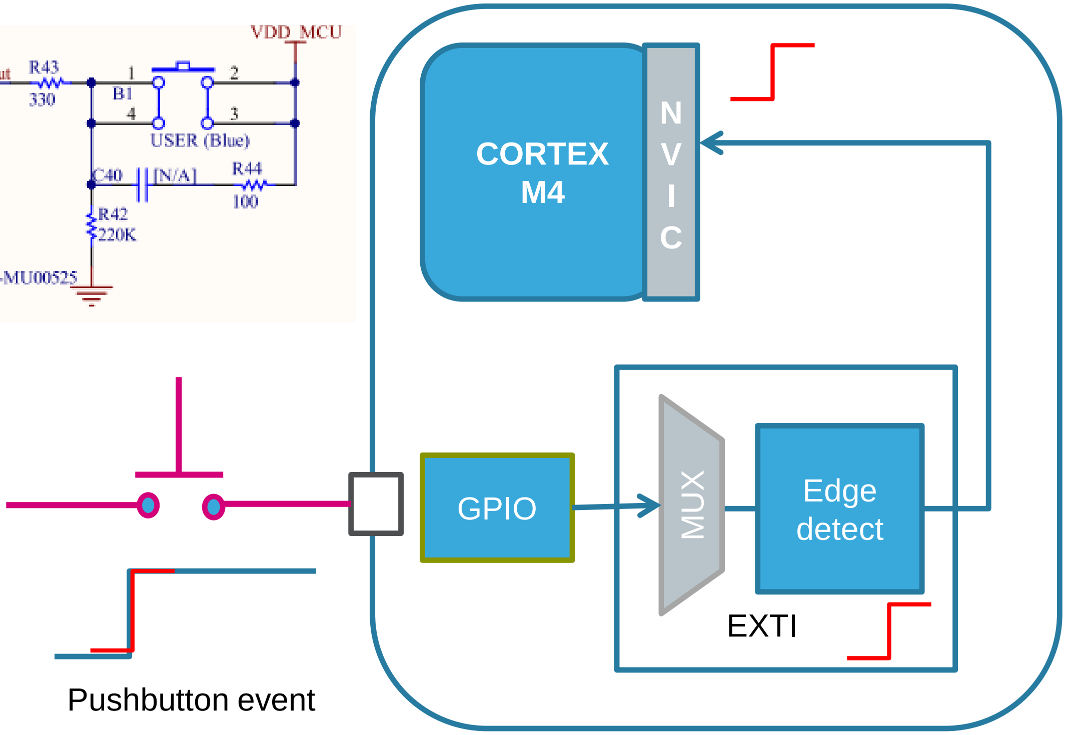
Verification

- Expected behavior:
  - When blue button (B1 USER) is pressed an interrupt is triggered and will call the EXTI IRQ handler in stm32l4xx_it.c file. The IRQ handler will then call the HAL_GPIO_EXTI_Callback() function in main.c file where the global variable (MODE_SELECTION) will be incremented.
  - MODE_SELECTION == 0 (Default), Green LED will toggle
  - MODE_SELECTION == 1, Green LED will turn off.
  - MODE_SELECTION == 2, Green LED will turn on.

Discussion (Interrupts)

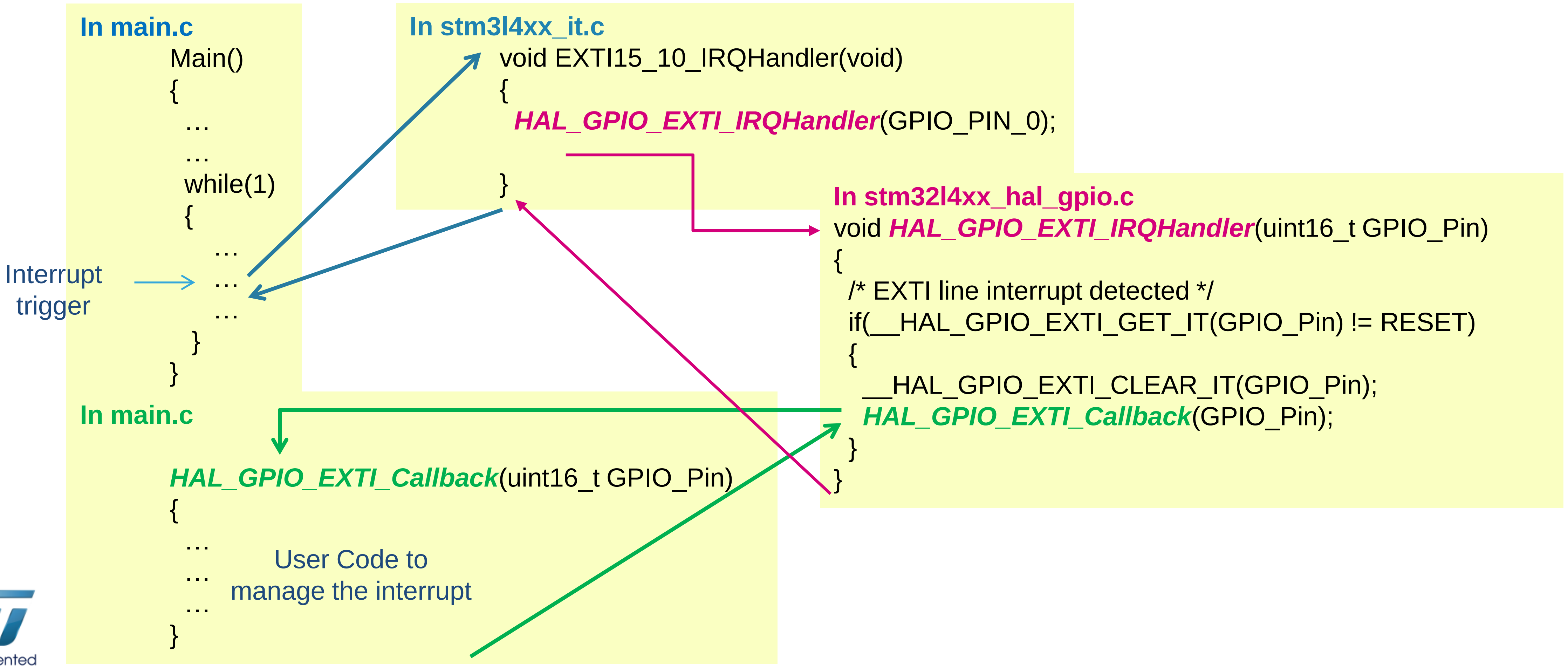
- Flow of interrupt
  - Pushbutton event occurs
  - EXTI detects valid edge
  - EXTI generates interrupt request
  - If the interrupt channel is enabled, the NVIC will acknowledge the interrupt request and checks the priority
  - When priority is higher, NVIC fetches EXTI Line interrupt vector. (Otherwise the interrupt will be set as pending until its priority becomes the highest compared to other pending interrupts)
  - Core executes EXTI IRQ Handler. Note that the handler will eventually call a callback function where the user will have to add and write the corresponding service routine.





Discussion (Callback)

- This flow of xxx_IRQhandler calls and xxx_Callback calls is similarly implemented for the other peripherals when interrupt request is enabled



In main.c

```
Main()
{
  …
  …
  while(1)
  {
     …
     …
     …
  }
}
```

In stm3l4xx_it.c

```
void EXTI15_10_IRQHandler(void)
{
  HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_0);

}
```

In stm32l4xx_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);
  }
}
```

In main.c

```
HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
  …
  …
  …
}
```

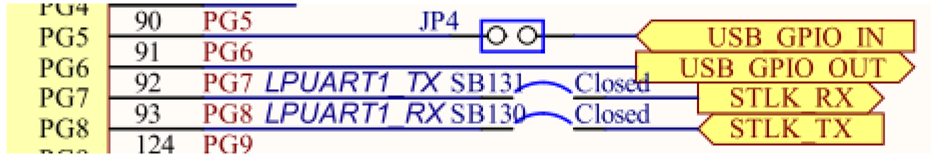
Using printf over UART

Using printf over UART

- Printf is often used to print debug messages to PC monitor when debugging a program. For embedded systems this is often not possible.
- Typically the serial port or UART is used instead to print debug message to PC
- This lab will show how to add printf capability to your STM32 code

UART printf

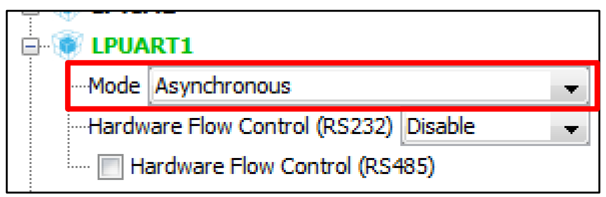
- For this hands-on, the STM32CubeMX will be used to generate the initialization codes for the LPUART1. For STM32L4R5 NUCLEO-144 board, ST-LINK Virtual COM port feature is only supported via LPUART pins
- Open the previous project STM32CubeMX project (GPIO_EXTI.ioc)
  - For example : C:\ASEAN Workshop\workspace\GPIO_EXTI.ioc
- Use STM32CubeMX to configure the LPUART1
  - Pinout tab
    - LPUART1 – Asynchronous mode on PG7 & PG8
  - Clock configuration tab
    - No change
  - Configuration tab
    - LPUART1
      - Baud Rate : 115200 bit/s
      - Word length : 8 bits (including Parity)
      - Parity: None
      - Stop bits: 1



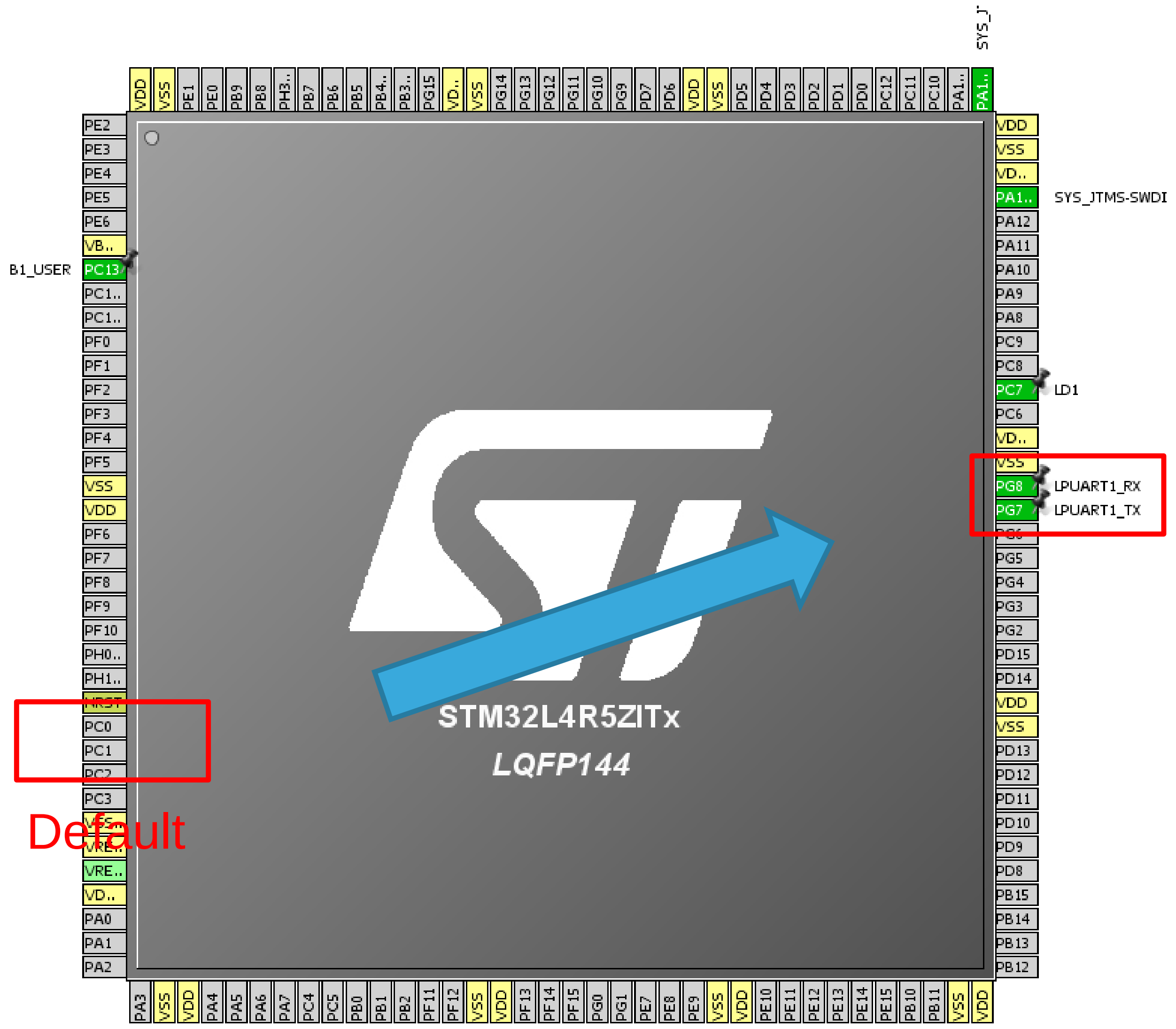
life.augmented

Pinout Configuration

- On STM32L4R5ZI, multiple pins support LPUART1 functionality
- Please ensure that PG8 and PG7 is selected as LPUART1 pins
  - By default PC0 and PC1 will be selected



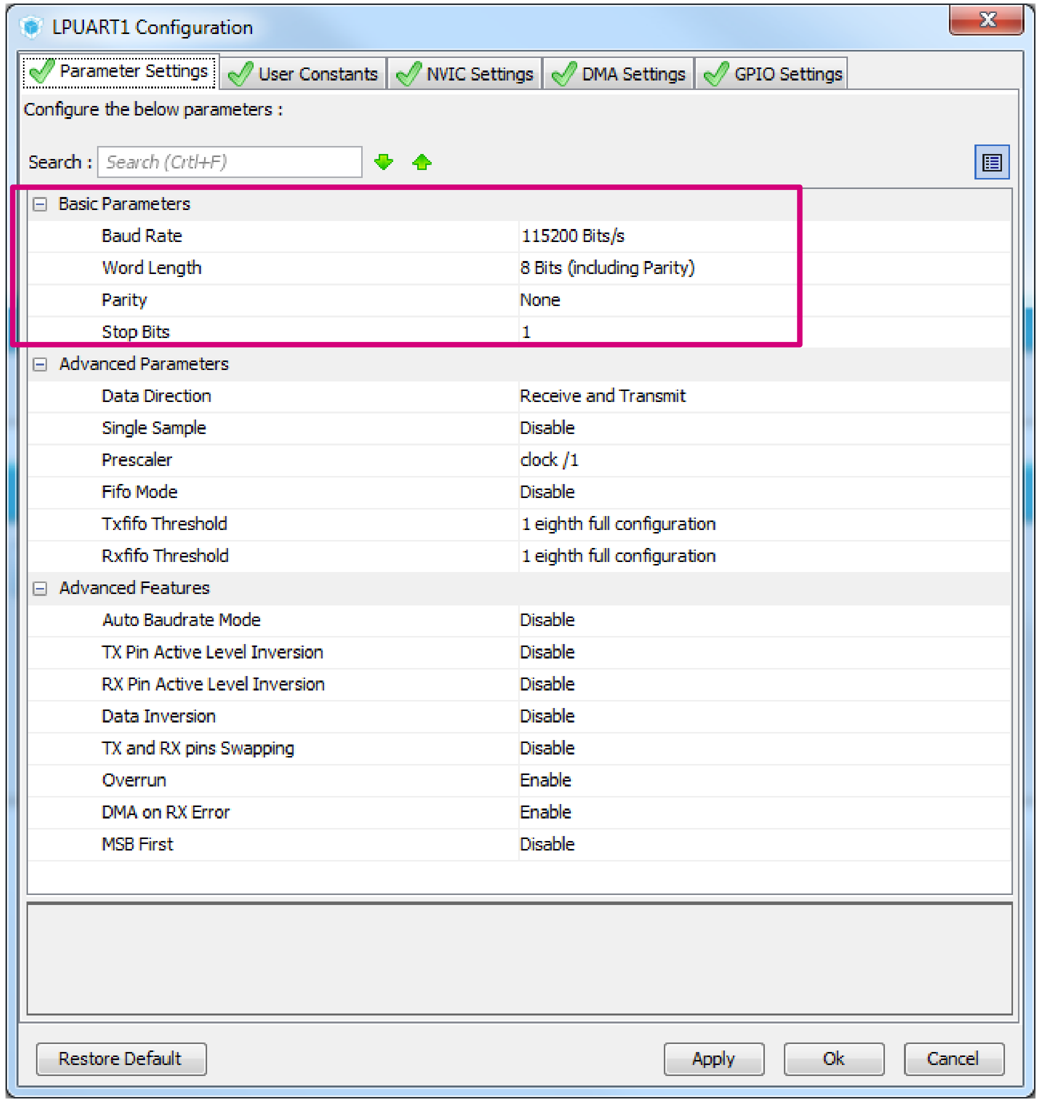
LPUART1
Mode: Asynchronous
Hardware Flow Control (RS232): Disable
Hardware Flow Control (RS485): ☐



LPUART Configuration

[Configuration] tab

- Select [LPUART1]
  - [Baud Rate] : 115200 bit/s
  - [Word length] : 8 bits (including Parity)
  - [Parity] : None
  - [Stop bits] : 1
  - Other settings use default



Configure the below parameters :

Basic Parameters	
Baud Rate	115200 Bits/s
Word Length	8 Bits (including Parity)
Parity	None
Stop Bits	1
Advanced Parameters	
Data Direction	Receive and Transmit
Single Sample	Disable
Prescaler	clock /1
Fifo Mode	Disable
Txfifo Threshold	1 eighth full configuration
Rxfifo Threshold	1 eighth full configuration
Advanced Features	
Auto Baudrate Mode	Disable
TX Pin Active Level Inversion	Disable
RX Pin Active Level Inversion	Disable
Data Inversion	Disable
TX and RX pins Swapping	Disable
Overrun	Enable
DMA on RX Error	Enable
MSB First	Disable

life.augmented

UART printf

- Save the project once all configuration are done.
- To complete, perform the following:
  - Generate Code
    - This will generate a project based on the Toolchain/IDE selected and all the necessary user and library files.
  - Generate Report (optional)
    - This will create a .pdf, .txt, and .jpg file
- Open Atollic TrueSTUDIO
  - When Code Generation is done, just click [Open Project]

Modifying the code

- In main.c source file, add code to print “Hello World” messages

```
/* USER CODE BEGIN 2 */
int count = 0;
/* USER CODE END 2 */

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

/* USER CODE BEGIN 3 */
    //Send message to UART port
    printf("\n\rHello World %d", count++);

    if (MODE_SELECTION == 0) {
             ..
             ..

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

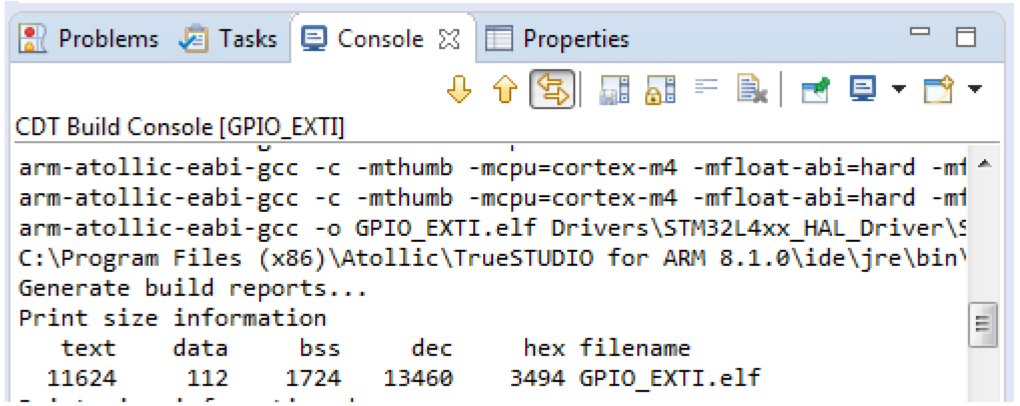
Modifying the code

- Override _write() function used to send data over UART using HAL function

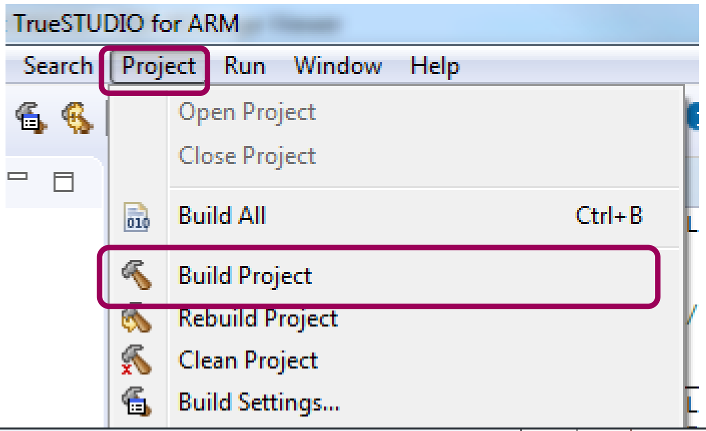
```
/* USER CODE BEGIN 4 */

int _write(int file, char *ptr, int len)
{
 HAL_UART_Transmit(&hlpuart1,(uint8_t *)ptr,len,HAL_MAX_DELAY);
 return len;
}
..
..
..
```

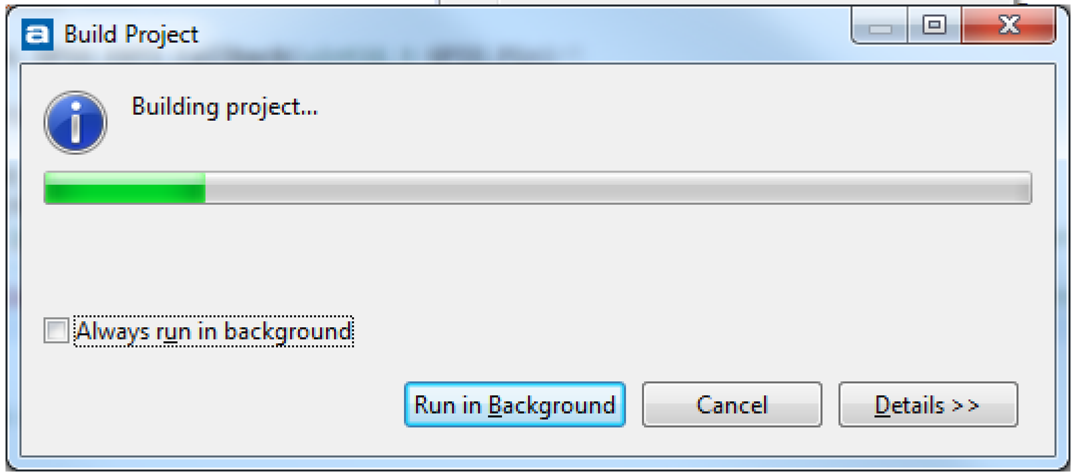
Build project

- Select [Project > Build Project] or [Project > Build All]
- Build result is displayed in the [Console] window



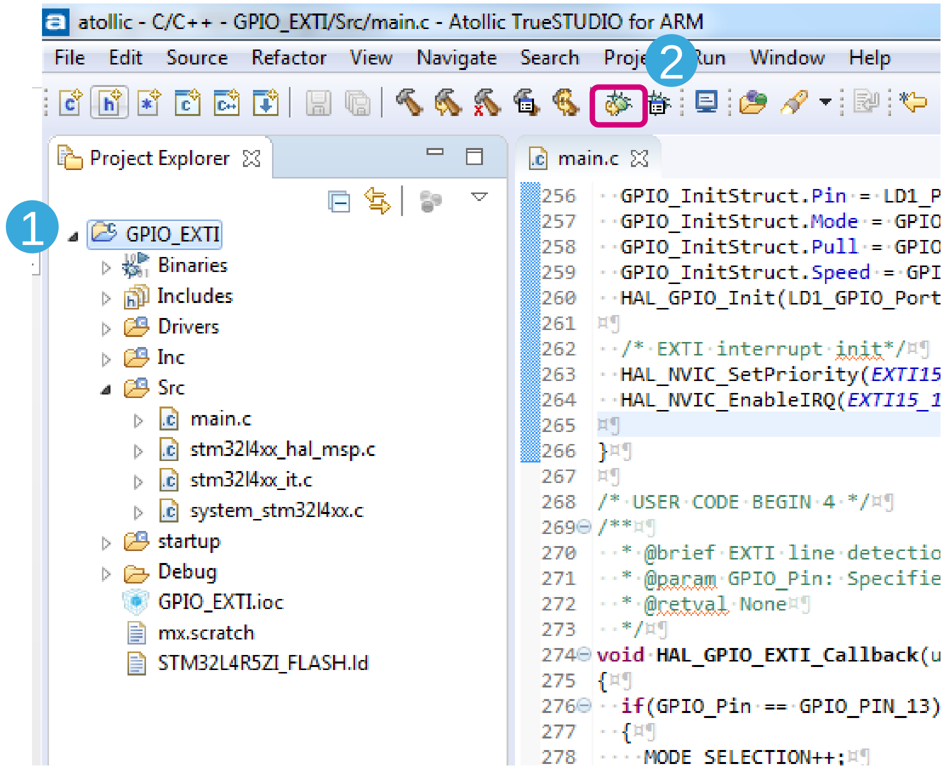
```
arm-atollic-eabi-gcc -c -mthumb -mcpu=cortex-m4 -mfloat-abi=hard -mf
arm-atollic-eabi-gcc -c -mthumb -mcpu=cortex-m4 -mfloat-abi=hard -mf
arm-atollic-eabi-gcc -o GPIO_EXTI.elf Drivers\STM32L4xx_HAL_Driver\S
C:\Program Files (x86)\Atollic\TrueSTUDIO for ARM 8.1.0\ide\jre\bin\
Generate build reports...
Print size information
   text    data     bss     dec     hex filename
  11624     112    1724   13460    3494 GPIO_EXTI.elf
```



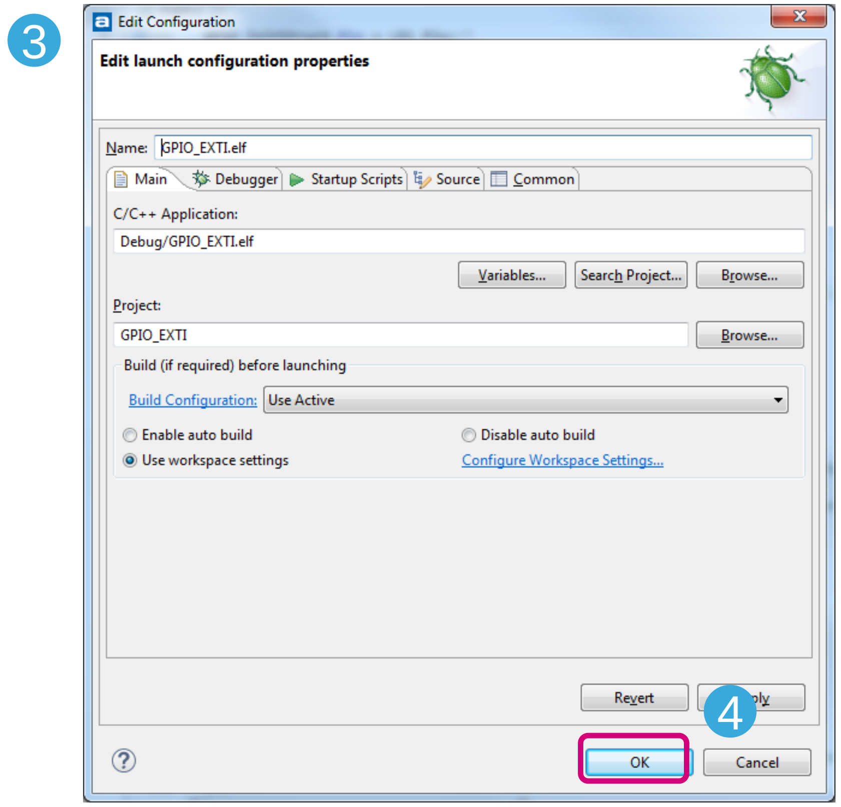


Start the debugger

1. Select project in Project Explorer view
2. Click on Debug button ( ) or press **F11** to start debug session
3. [Debug Configuration] dialog box will appear when debugging project the first time
4. Click [OK] to accept default configuration



```
  GPIO_InitStruct.Pin = LD1_P
  GPIO_InitStruct.Mode = GPIO
  GPIO_InitStruct.Pull = GPIO
  GPIO_InitStruct.Speed = GPI
  HAL_GPIO_Init(LD1_GPIO_Port

  /* EXTI interrupt init*/
  HAL_NVIC_SetPriority(EXTI15
  HAL_NVIC_EnableIRQ(EXTI15_1

}

/* USER CODE BEGIN 4 */
/**
  * @brief EXTI line detectio
  * @param GPIO_Pin: Specifie
  * @retval None
  */
void HAL_GPIO_EXTI_Callback(u
{
  if(GPIO_Pin == GPIO_PIN_13)
  {
    MODE_SELECTION++;
```



Name: GPIO_EXTI.elf

C/C++ Application: Debug/GPIO_EXTI.elf

Project: GPIO_EXTI

Build (if required) before launching

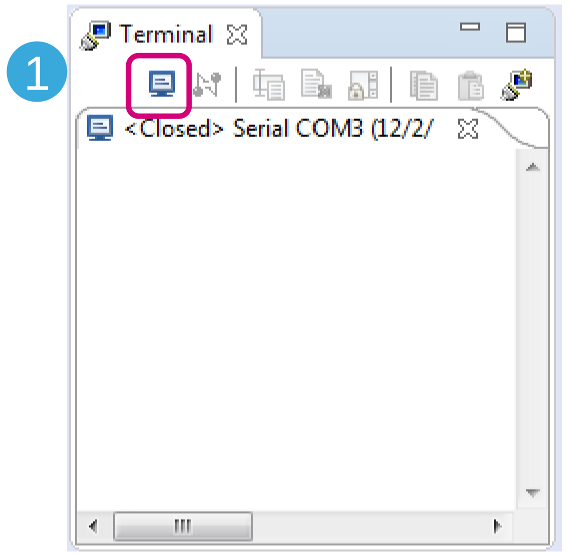
Build Configuration: Use Active

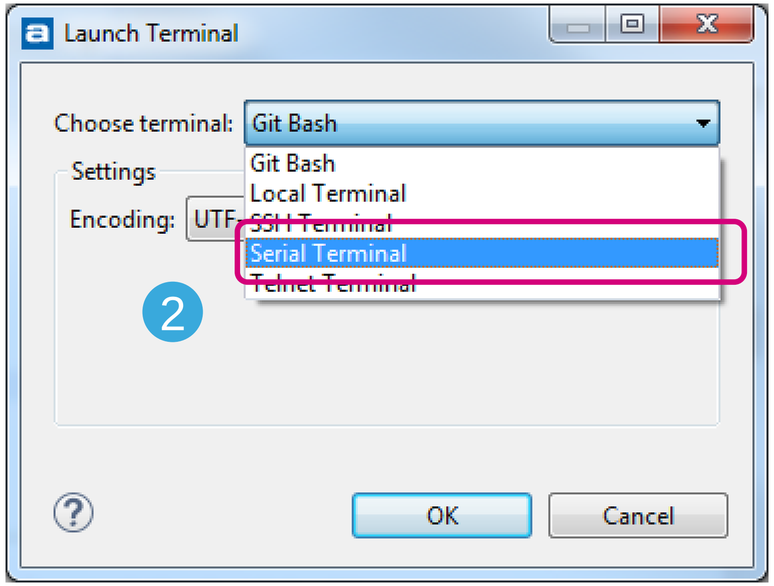
( ) Enable auto build ( ) Disable auto build

(●) Use workspace settings Configure Workspace Settings...

Terminal view

- Allows I/O communication with target using Serial communication
- Steps to configure
  1. Open a Terminal
  2. Select “Serial Terminal”
  3. Select [Port] – refer to next slide to determine your port name
  4. Change [Baud Rate] to the one configured on MCU (115200)
  5. Click [OK]

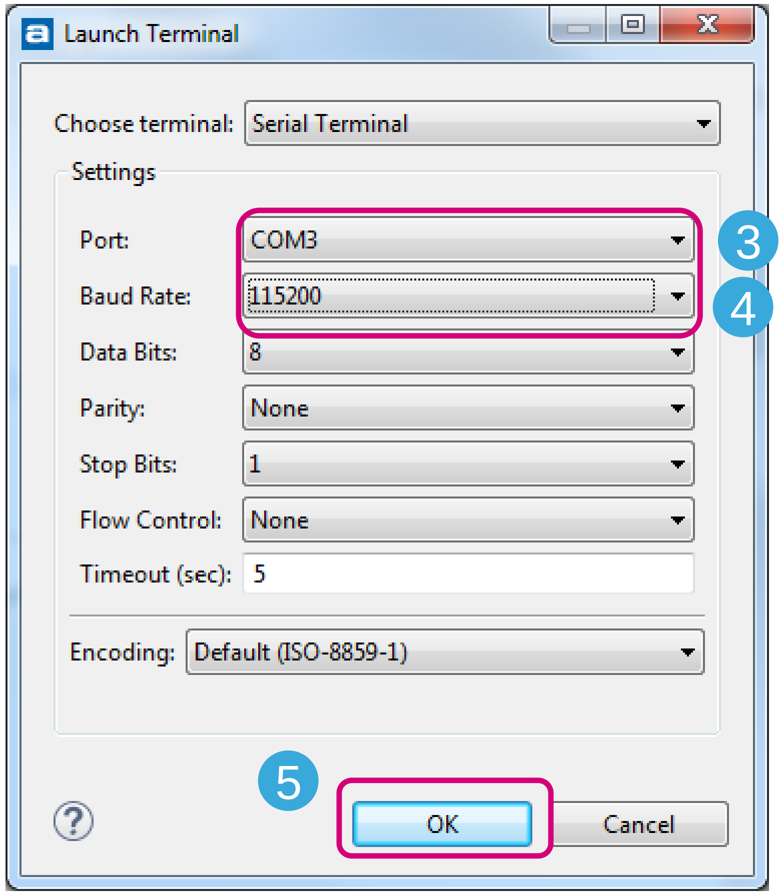




Choose terminal: Git Bash

Settings

Encoding: UTF-



Choose terminal: Serial Terminal

Settings

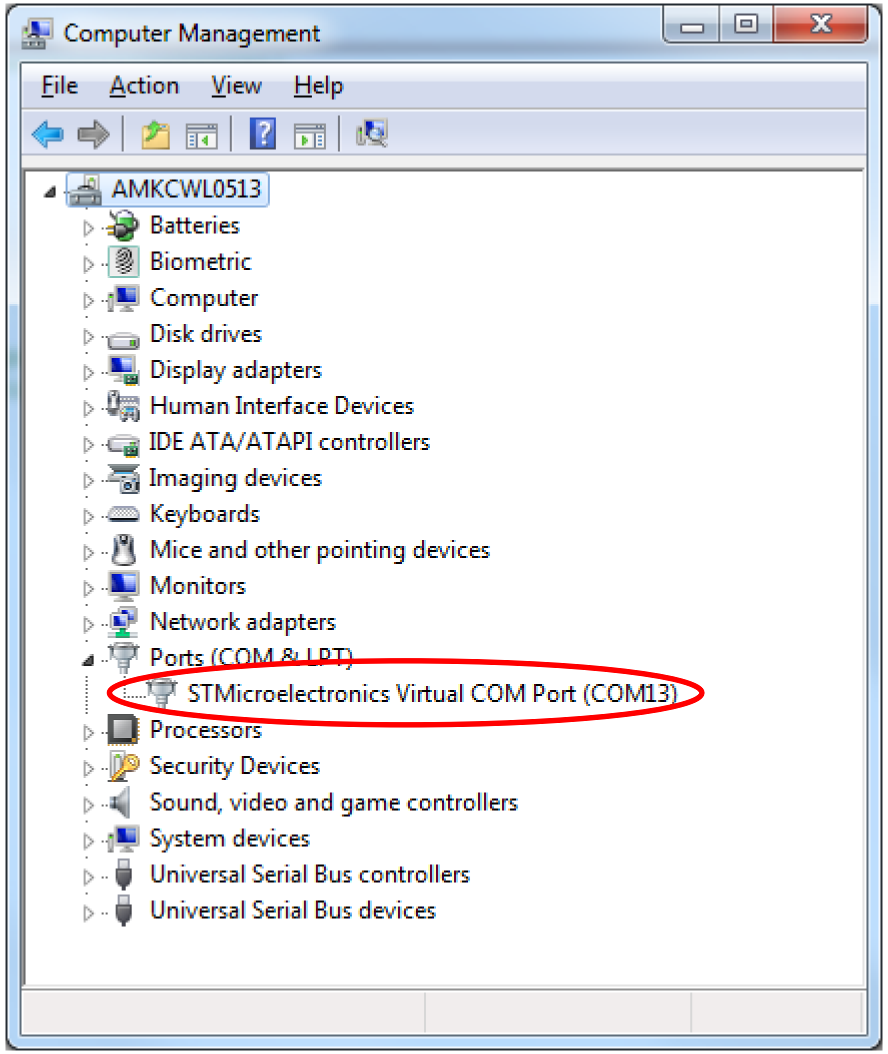
Port:	COM3
Baud Rate:	115200
Data Bits:	8
Parity:	None
Stop Bits:	1
Flow Control:	None
Timeout (sec):	5

Encoding: Default (ISO-8859-1)

Virtual COM Port name (COM#)

- If you have successfully installed the driver, you should be able to find the COM Port number from Windows Device Manager

- If not please install refer to the installation instructions again

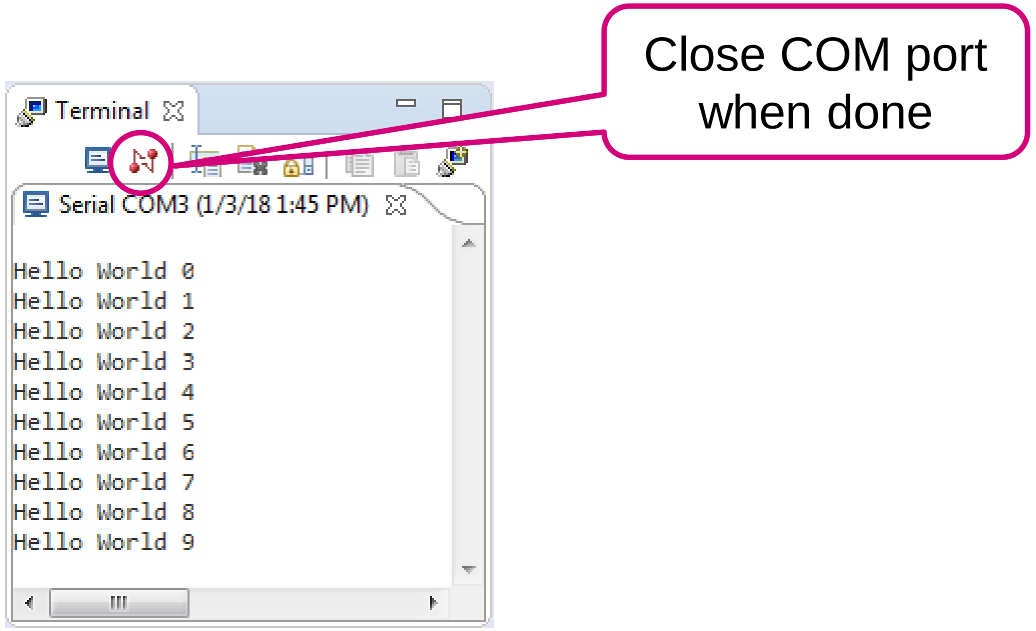


AMKCWL0513
- Batteries
- Biometric
- Computer
- Disk drives
- Display adapters
- Human Interface Devices
- IDE ATA/ATAPI controllers
- Imaging devices
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Ports (COM & LPT)
  - STMicroelectronics Virtual COM Port (COM13)
- Processors
- Security Devices
- Sound, video and game controllers
- System devices
- Universal Serial Bus controllers
- Universal Serial Bus devices

life.augmented

Verification

- Expected behaviour
  - The message “Hello World” and incrementing count value will appear in the Terminal View
  - If you suspend/pause the program execution, the message will stop printing.



```
Hello World 0
Hello World 1
Hello World 2
Hello World 3
Hello World 4
Hello World 5
Hello World 6
Hello World 7
Hello World 8
Hello World 9
```

life.augmented