

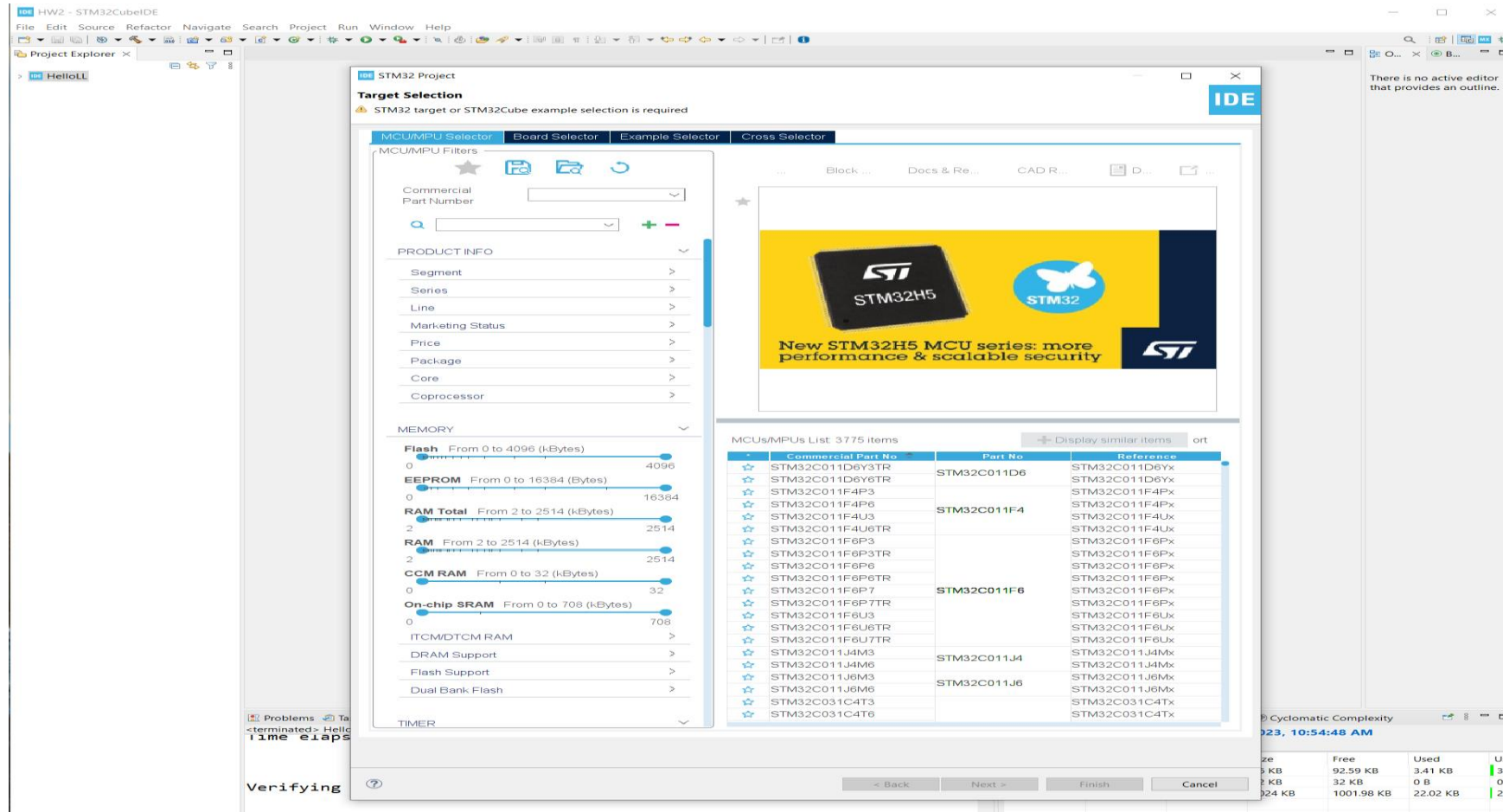
UCSD Embedded C Assignment 5

By

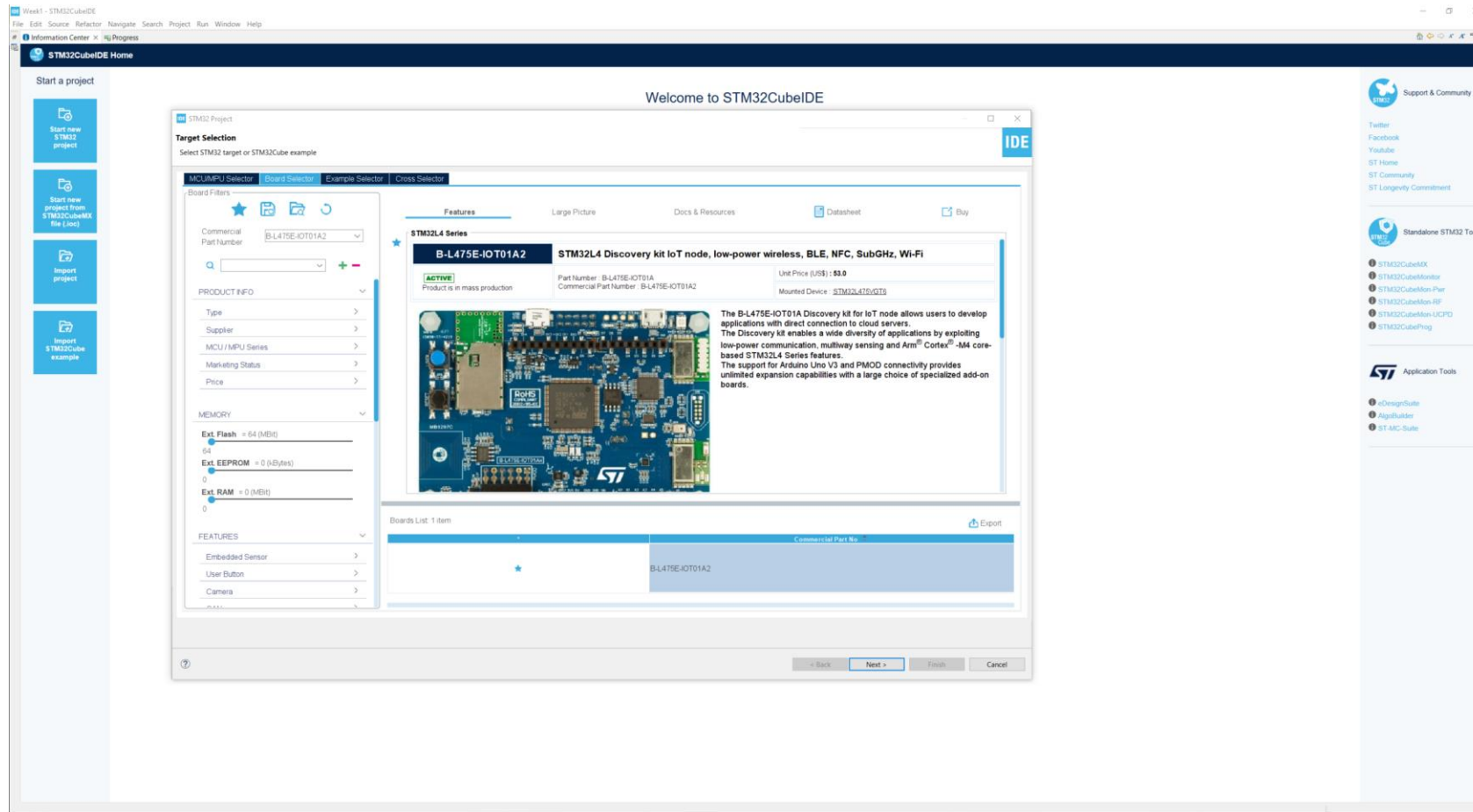
Hsuankai Chang

hsuankac@umich.edu

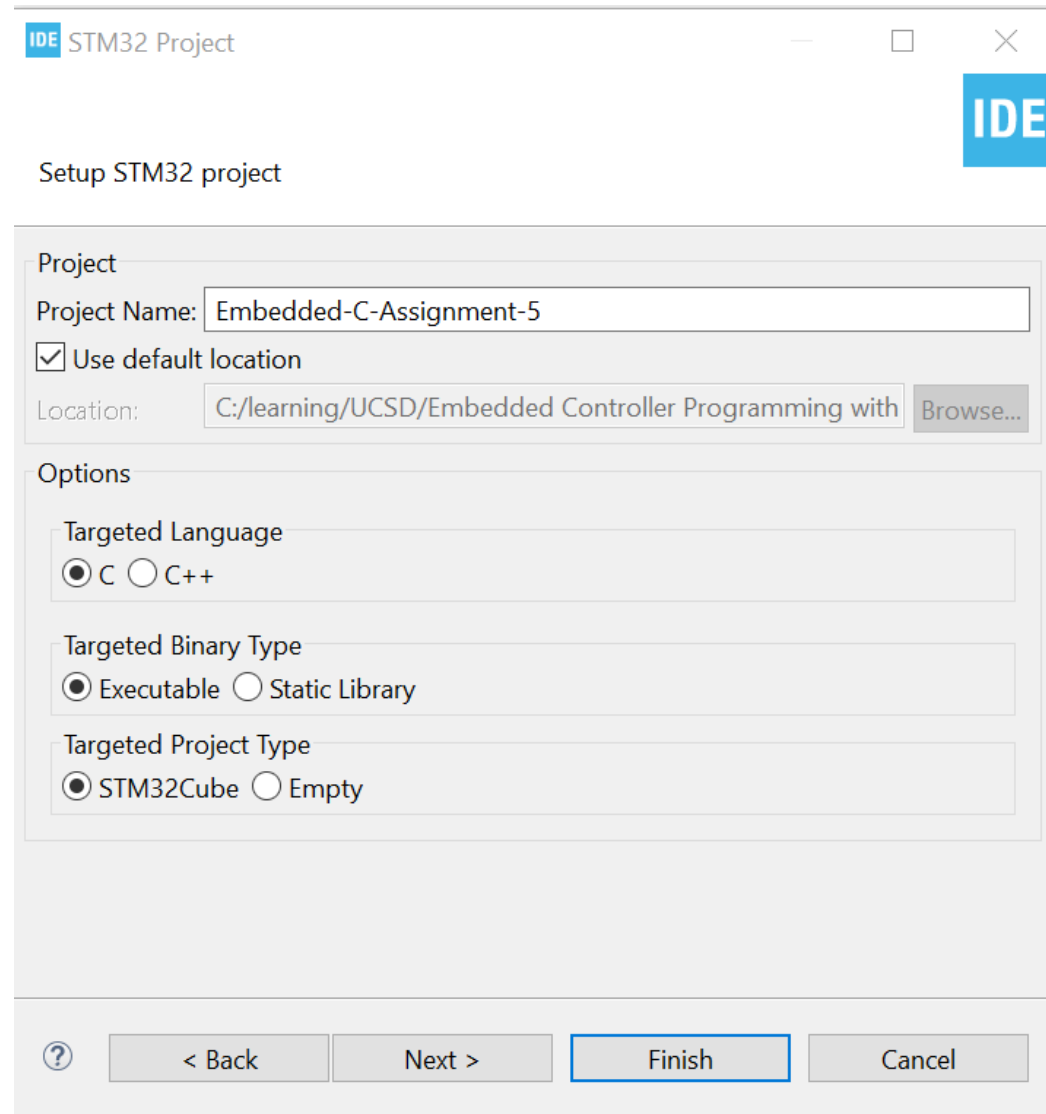
Step 1. Startup STM32CubeIDE and create new STM32 project



Step 2. Access board selector and type in the board you use, click Next



Step 3. Enter the project name then click Next



The image shows a 'Setup STM32 project' dialog box from an IDE. The window title is 'IDE STM32 Project'. The dialog is titled 'Setup STM32 project'. It contains two main sections: 'Project' and 'Options'. In the 'Project' section, the 'Project Name' field is filled with 'Embedded-C-Assignment-5'. The 'Use default location' checkbox is checked. The 'Location' field shows the path 'C:/learning/UCSD/Embedded Controller Programming with' followed by a 'Browse...' button. The 'Options' section has three groups of radio buttons: 'Targeted Language' with 'C' selected, 'Targeted Binary Type' with 'Executable' selected, and 'Targeted Project Type' with 'STM32Cube' selected. At the bottom, there are four buttons: a help button (question mark), '< Back', 'Next >', and 'Finish' (which is highlighted with a blue border), and a 'Cancel' button.

IDE STM32 Project

Setup STM32 project

Project

Project Name: Embedded-C-Assignment-5

☒ Use default location

Location: C:/learning/UCSD/Embedded Controller Programming with Browse...

Options

Targeted Language

☒ C ☐ C++

Targeted Binary Type

☒ Executable ☐ Static Library

Targeted Project Type

☒ STM32Cube ☐ Empty

? < Back Next > Finish Cancel

Step 4. See the firmware package name and version



The image shows a Windows-style dialog box titled "STM32 Project" with a subtitle "Firmware Library Package Setup". The subtitle is followed by the instruction "Setup STM32 target's firmware". The dialog is divided into three sections: "Target and Firmware Package", "Firmware and Software Package Repository", and "Code Generator Options". In the first section, "Target Reference" is set to "B-L475E-IOT01A2" and "Firmware Package Name and Version" is set to "STM32Cube FW_L4" with a dropdown menu showing "V1.17.2". The second section shows the "Location" as "C:\Users\hsuankai.chang\STM32Cube\Repository" and includes a link to the "Firmware Updater". The third section has three radio button options for code generation, with "Copy only the necessary library files" being selected. At the bottom, there are buttons for "?", "< Back", "Next >", "Finish", and "Cancel".

IDE STM32 Project

Firmware Library Package Setup

Setup STM32 target's firmware

Target and Firmware Package

Target Reference: B-L475E-IOT01A2

Firmware Package Name and Version: STM32Cube FW_L4 V1.17.2

Firmware and Software Package Repository

Location:
C:\Users\hsuankai.chang\STM32Cube\Repository

See ['Firmware Updater'](#) for settings related to package installation

Code Generator Options

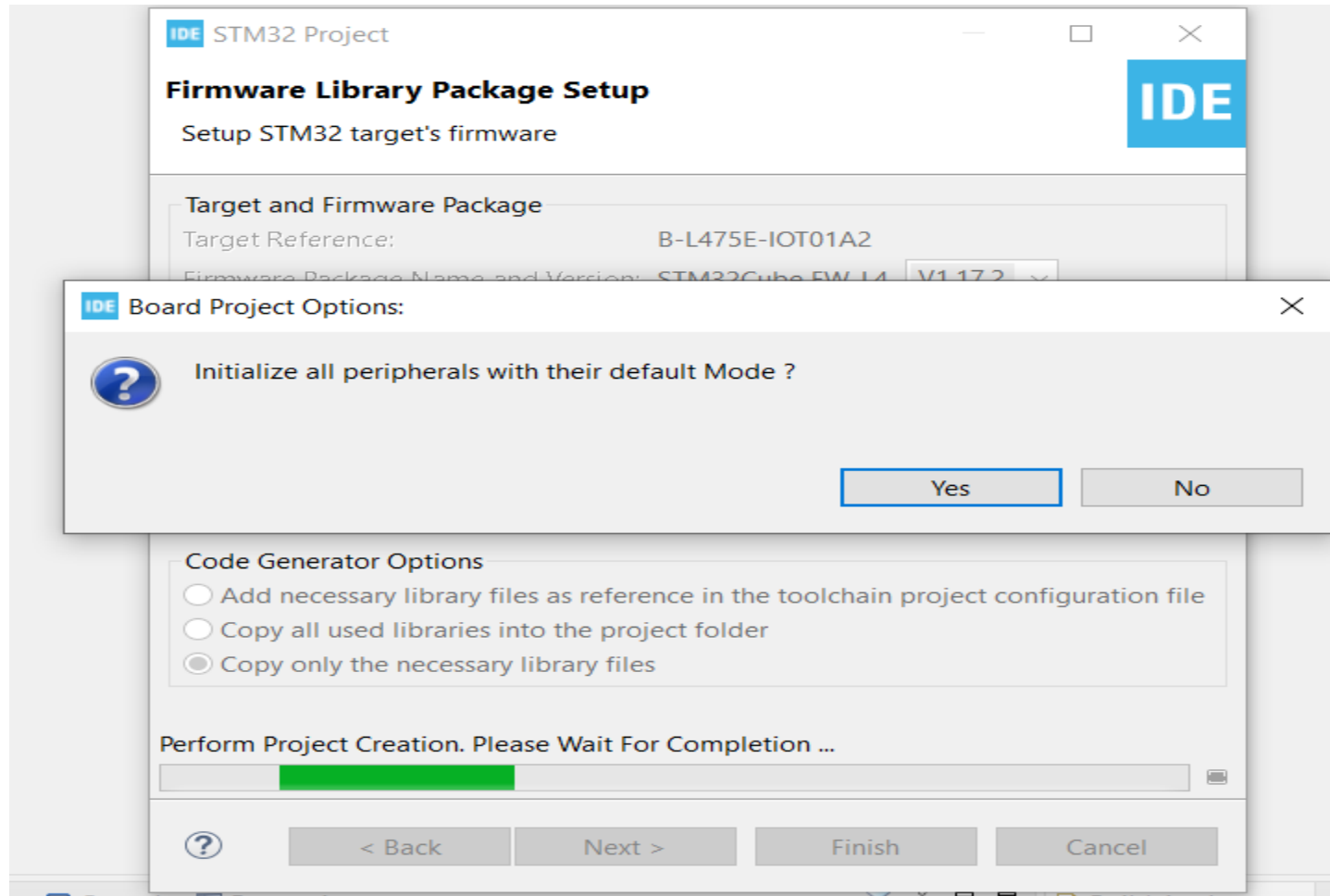
☐ Add necessary library files as reference in the toolchain project configuration file

☐ Copy all used libraries into the project folder

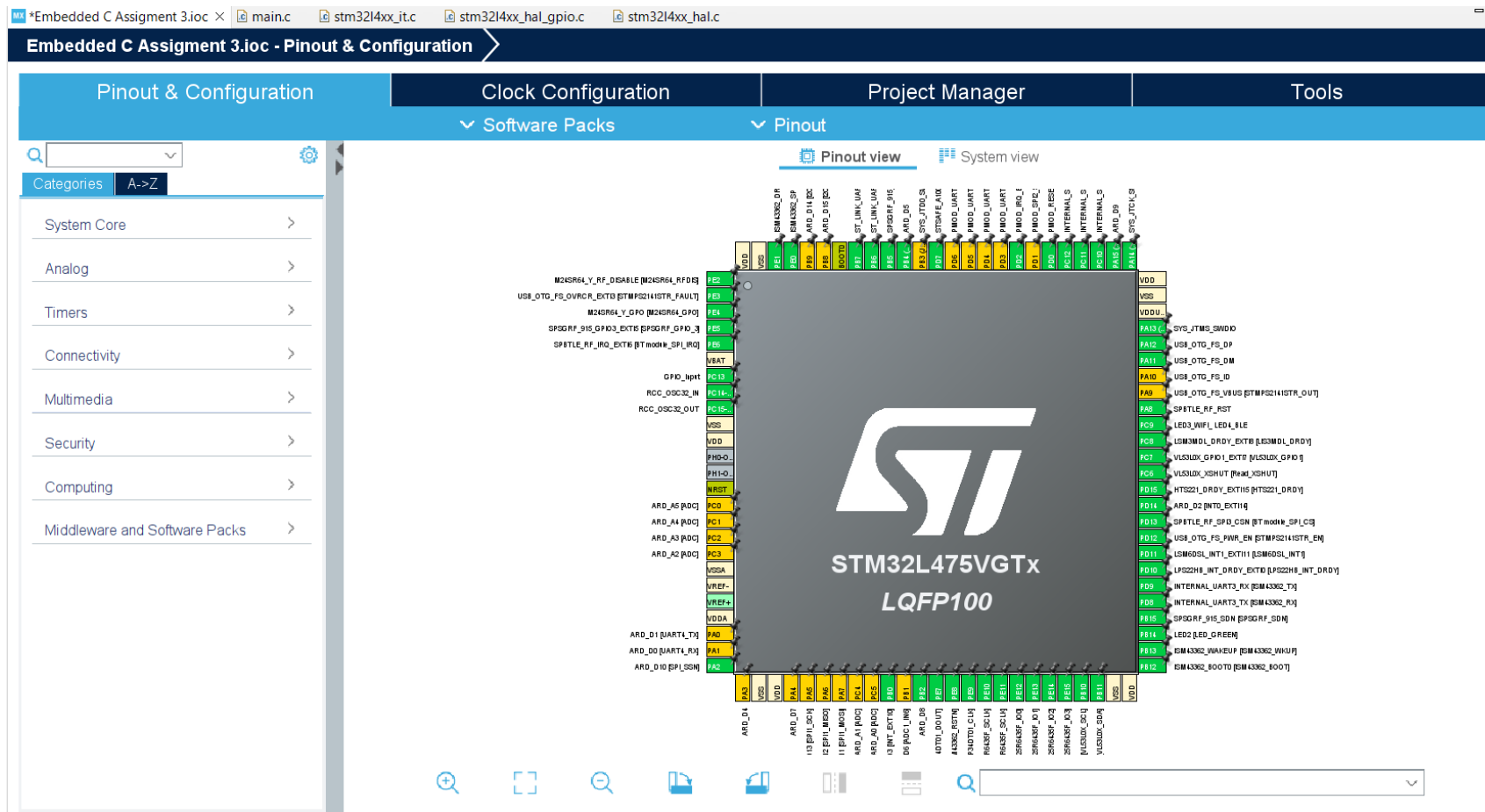
☒ Copy only the necessary library files

? < Back Next > Finish Cancel

Step 5. Click yes to initialize all peripherals to default



Step 6. When in .ioc file, click Pinout & Configurations



Step 7. Review the LED2 GPIO configuration by clicking System view -> GPIO -> PB14

The screenshot shows the 'Embedded-C-Assignment-5.ioc - Pinout & Configuration' window. The 'Pinout & Configuration' tab is active, and the 'System view' is selected. The 'GPIO Mode and Configuration' section is expanded, showing a table of GPIO pins and their configurations. The 'PB14' pin is selected, and its configuration is shown in the 'PB14 Configuration' section.

GPIO Mode and Configuration

Group By Peripherals

Configuration

Search Signals

Search (Ctrl+F)

Show only Modified Pins

Pin...	Signal...	GPIO...	GPIO...	GPIO...	Maxim...	Fast M...	User L...	Modified
PA2	n/a	Low	Output...	No pull...	Low	n/a	ARD...	✓
PA8	n/a	Low	Output...	No pull...	Low	n/a	SPBT...	✓
PA15 (...)	n/a	Low	Output...	No pull...	Low	n/a	ARD_D9	✓
PB0	n/a	n/a	Extern...	No pull...	n/a	n/a	ARD...	✓
PB2	n/a	Low	Output...	No pull...	Low	n/a	ARD_D8	✓
PB4 (...)	n/a	Low	Output...	No pull...	Low	n/a	ARD_D5	✓
PB5	n/a	High	Output...	No pull...	Low	n/a	SPSG...	✓
PB12	n/a	Low	Output...	No pull...	Low	n/a	ISM43...	✓
PB13	n/a	Low	Output...	No pull...	Low	n/a	ISM43...	✓
PB14	n/a	Low	Output...	No pull...	Low	n/a	LED2 [...]	✓
PB15	n/a	Low	Output...	No pull...	Low	n/a	SPSG...	✓
PC6	n/a	Low	Output...	No pull...	Low	n/a	VL53L...	✓
PC7	n/a	n/a	Extern...	No pull...	n/a	n/a	VL53L...	✓
PC8	n/a	n/a	Extern...	No pull...	n/a	n/a	LSM3...	✓
PC9	n/a	Low	Output...	No pull...	Low	n/a	LED3...	✓
PC13	n/a	n/a	Extern...	No pull...	n/a	n/a	BUTT...	✓
PD0	n/a	Low	Output...	No pull...	Low	n/a	PMOD...	✓
PD2	n/a	n/a	Extern...	No pull...	n/a	n/a	PMOD...	✓
PD7	n/a	Low	Output...	No pull...	Low	n/a	STSA...	✓
PD8	n/a	Low	Output...	No pull...	Low	n/a	Custo...	✓
PD10	n/a	n/a	Extern...	No pull...	n/a	n/a	LPS22...	✓
PD11	n/a	n/a	Extern...	No pull...	n/a	n/a	LSM6...	✓

PB14 Configuration :

GPIO output level: Low

GPIO mode: Output Push Pull

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

Maximum output speed: Low

User Label: LED2 [LED_GREEN]

Step 8. Review the BUTTON GPIO configuration

The screenshot displays the 'Embedded-C-Assignment-5.ioc - Pinout & Configuration' window. The 'Pinout & Configuration' tab is active, showing the 'GPIO Mode and Configuration' section. The 'Configuration' sub-tab is selected, displaying a table of GPIO pins and their configurations. The 'PC13 Configuration' dialog is open, showing the following settings:

- GPIO mode: External Interrupt Mode with Falling edge trigger detection
- GPIO Pull-up/Pull-down: No pull-up and no pull-down
- User Label: BUTTON_EXTI13 [B2]

The table of GPIO pins is as follows:

Pin Name	Signal on Pin	GPIO output	GPIO mode	GPIO Pull-up	Maximum output current	Fast Mode	User Label	Modified
PA2	n/a	Low	Output Pus...	No pull-up a...	Low	n/a	ARD_D10 [...	✓
PA8	n/a	Low	Output Pus...	No pull-up a...	Low	n/a	SPBTLE_R...	✓
PA15 (JTDI)	n/a	Low	Output Pus...	No pull-up a...	Low	n/a	ARD_D9	✓
PB0	n/a	n/a	External Int...	No pull-up a...	n/a	n/a	ARD_D3 [IN...	✓
PB2	n/a	Low	Output Pus...	No pull-up a...	Low	n/a	ARD_D8	✓
PB4 (NJTR...	n/a	Low	Output Pus...	No pull-up a...	Low	n/a	ARD_D5	✓
PB5	n/a	High	Output Pus...	No pull-up a...	Low	n/a	SPSGRF_9...	✓
PB12	n/a	Low	Output Pus...	No pull-up a...	Low	n/a	ISM43362_...	✓
PB13	n/a	Low	Output Pus...	No pull-up a...	Low	n/a	ISM43362_...	✓
PB14	n/a	Low	Output Pus...	No pull-up a...	Low	n/a	LED2 [LED...	✓
PB15	n/a	Low	Output Pus...	No pull-up a...	Low	n/a	SPSGRF_9...	✓
PC6	n/a	Low	Output Pus...	No pull-up a...	Low	n/a	VL53L0X_X...	✓
PC7	n/a	n/a	External Int...	No pull-up a...	n/a	n/a	VL53L0X_G...	✓
PC8	n/a	n/a	External Int...	No pull-up a...	n/a	n/a	LSM3MDL_...	✓
PC9	n/a	Low	Output Pus...	No pull-up a...	Low	n/a	LED3_WIFI...	✓
PC13	n/a	n/a	External Int...	No pull-up a...	n/a	n/a	BUTTON_E...	✓
PD0	n/a	Low	Output Pus...	No pull-up a...	Low	n/a	PMOD_RE...	✓
PD2	n/a	n/a	External Int...	No pull-up a...	n/a	n/a	PMOD_IRQ...	✓
PD7	n/a	Low	Output Pus...	No pull-up a...	Low	n/a	STSAFE_A...	✓
PD8	n/a	Low	Output Pus...	No pull-up a...	Low	n/a	Custom LED	✓
PD10	n/a	n/a	External Int...	No pull-up a...	n/a	n/a	LPS22HB_I...	✓
PD11	n/a	n/a	External Int...	No pull-up a...	n/a	n/a	LSM6DSL_I...	✓

Step 9. Review the PB2 Pin GPIO configuration (D8 on the board), I name it Custom LED

The screenshot shows the STM32CubeIDE Pinout & Configuration window. The left sidebar lists various system components, with GPIO selected. The main area displays a table of pin configurations. The PB2 pin is highlighted, and its configuration is shown in the bottom panel.

Pin Name	Signal on Pin	GPIO output...	GPIO mode	GPIO Pull...	Maximum o...	Fast Mode	User Label	Modified
PA2	n/a	Low	Output Pus...	No pull-up ...	Low	n/a	ARD_D10 [...]	✓
PA8	n/a	Low	Output Pus...	No pull-up ...	Low	n/a	SPBTLE_R...	✓
PA15 (JTDI)	n/a	Low	Output Pus...	No pull-up ...	Low	n/a	ARD_D9	✓
PB0	n/a	n/a	External Int...	No pull-up ...	n/a	n/a	ARD_D3 [I...	✓
PB2	n/a	Low	Output Pus...	No pull-up ...	Low	n/a	Custom LED	✓
PB4 (NJTR...	n/a	Low	Output Pus...	No pull-up ...	Low	n/a	ARD_D5	✓
PB5	n/a	High	Output Pus...	No pull-up ...	Low	n/a	SPSGRF_9...	✓
PB12	n/a	Low	Output Pus...	No pull-up ...	Low	n/a	ISM43362_...	✓
PB13	n/a	Low	Output Pus...	No pull-up ...	Low	n/a	ISM43362_...	✓
PB14	n/a	Low	Output Pus...	No pull-up ...	Low	n/a	LED2 [LED...	✓
PB15	n/a	Low	Output Pus...	No pull-up ...	Low	n/a	SPSGRF_9...	✓
PC6	n/a	Low	Output Pus...	No pull-up ...	Low	n/a	VL53L0X_X...	✓
PC7	n/a	n/a	External Int...	No pull-up ...	n/a	n/a	VL53L0X_X...	✓
PC8	n/a	n/a	External Int...	No pull-up ...	n/a	n/a	LSM3MDL_...	✓
PC9	n/a	Low	Output Pus...	No pull-up ...	Low	n/a	LED3_WIFI...	✓
PC13	n/a	n/a	External Int...	No pull-up ...	n/a	n/a	BUTTON_E...	✓
PD0	n/a	Low	Output Pus...	No pull-up ...	Low	n/a	PMOD_RE...	✓
PD2	n/a	n/a	External Int...	No pull-up ...	n/a	n/a	PMOD_IR...	✓
PD7	n/a	Low	Output Pus...	No pull-up ...	Low	n/a	STSAFE_A...	✓
PD10	n/a	n/a	External Int...	No pull-up ...	n/a	n/a	LPS22HB_I...	✓
PD11	n/a	n/a	External Int...	No pull-up ...	n/a	n/a	LSM6DSL_I...	✓
PD12	n/a	Low	Output Pus...	No pull-up ...	Low	n/a	USB_OTG...	✓

PB2 Configuration :

GPIO output level: Low

GPIO mode: Output Push Pull

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

Maximum output speed: Low

User Label: Custom LED

Step 10. Generate code, and review the code that initialize the button interrupt

```
Embedded-C-Assignment-5.ioc  main.c  stm32l4xx_hal_gpio.h
437 HAL_GPIO_WritePin(GPIOA, ARD_D10_Pin|SPBTLE_RF_RST_Pin|ARD_D9_Pin, GPIO_PIN_RESET);
438
439 /*Configure GPIO pin Output Level */
440 HAL_GPIO_WritePin(GPIOB, ARD_D8_Pin|ISM43362_BOOT0_Pin|ISM43362_WAKEUP_Pin|LED2_Pin
441 |SPSGRF_915_SDN_Pin|ARD_D5_Pin, GPIO_PIN_RESET);
442
443 /*Configure GPIO pin Output Level */
444 HAL_GPIO_WritePin(GPIOD, Custom_LED_Pin|USB_OTG_FS_PWR_EN_Pin|PMOD_RESET_Pin|STSAFE_A100_RESET_Pin, GPIO_PIN_RESET);
445
446 /*Configure GPIO pin Output Level */
447 HAL_GPIO_WritePin(SPBTLE_RF_SPI3_CSN_GPIO_Port, SPBTLE_RF_SPI3_CSN_Pin, GPIO_PIN_SET);
448
449 /*Configure GPIO pin Output Level */
450 HAL_GPIO_WritePin(GPIOC, VL53L0X_XSHUT_Pin|LED3_WIFI_LED4_BLE_Pin, GPIO_PIN_RESET);
451
452 /*Configure GPIO pin Output Level */
453 HAL_GPIO_WritePin(SPSGRF_915_SPI3_CSN_GPIO_Port, SPSGRF_915_SPI3_CSN_Pin, GPIO_PIN_SET);
454
455 /*Configure GPIO pin Output Level */
456 HAL_GPIO_WritePin(ISM43362_SPI3_CSN_GPIO_Port, ISM43362_SPI3_CSN_Pin, GPIO_PIN_SET);
457
458 /*Configure GPIO pins : M24SR64_Y_RF_DISABLE_Pin M24SR64_Y_GPO_Pin ISM43362_RST_Pin ISM43362_SPI3_CSN_Pin */
459 GPIO_InitStruct.Pin = M24SR64_Y_RF_DISABLE_Pin|M24SR64_Y_GPO_Pin|ISM43362_RST_Pin|ISM43362_SPI3_CSN_Pin;
460 GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
461 GPIO_InitStruct.Pull = GPIO_NOPULL;
462 GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
463 HAL_GPIO_Init(GPIOE, &GPIO_InitStruct);
464
465 /*Configure GPIO pins : USB_OTG_FS_OVRCR_EXTI3_Pin SPSGRF_915_GPIO3_EXTI5_Pin SPBTLE_RF_IRQ_EXTI6_Pin ISM43362_DRDY_EXTI1_Pin */
466 GPIO_InitStruct.Pin = USB_OTG_FS_OVRCR_EXTI3_Pin|SPSGRF_915_GPIO3_EXTI5_Pin|SPBTLE_RF_IRQ_EXTI6_Pin|ISM43362_DRDY_EXTI1_Pin;
467 GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
468 GPIO_InitStruct.Pull = GPIO_NOPULL;
469 HAL_GPIO_Init(GPIOE, &GPIO_InitStruct);
470
471 /*Configure GPIO pin : BUTTON_EXTI13_Pin */
472 GPIO_InitStruct.Pin = BUTTON_EXTI13_Pin;
473 GPIO_InitStruct.Mode = GPIO_MODE_IT_FALLING;
474 GPIO_InitStruct.Pull = GPIO_NOPULL;
475 HAL_GPIO_Init(BUTTON_EXTI13_GPIO_Port, &GPIO_InitStruct);
476
477 /*Configure GPIO pins : ARD_A5_Pin ARD_A4_Pin ARD_A3_Pin ARD_A2_Pin
478 ARD_A1_Pin ARD_A0_Pin */
479 GPIO_InitStruct.Pin = ARD_A5_Pin|ARD_A4_Pin|ARD_A3_Pin|ARD_A2_Pin
480 |ARD_A1_Pin|ARD_A0_Pin;
481 GPIO_InitStruct.Mode = GPIO_MODE_ANALOG_ADC_CONTROL;
482 GPIO_InitStruct.Pull = GPIO_NOPULL;
483 HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);
484
485 /*Configure GPIO pins : ARD_D1_Pin ARD_D0_Pin */
486 GPIO_InitStruct.Pin = ARD_D1_Pin|ARD_D0_Pin;
487 GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
488 GPIO_InitStruct.Pull = GPIO_NOPULL;
489 GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
490 GPIO_InitStruct.Alternate = GPIO_AF8_UART4;
491 HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
492
493 /*Configure GPIO pins : ARD_D10_Pin SPBTLE_RF_RST_Pin ARD_D9_Pin */
494 GPIO_InitStruct.Pin = ARD_D10_Pin|SPBTLE_RF_RST_Pin|ARD_D9_Pin;
495 GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
496 GPIO_InitStruct.Pull = GPIO_NOPULL;
```

Step 11. In stm32l4xx_it.c, find interrupt handler code

```
Embedded-C-Assignment-5.ioc  main.c  stm32l4xx_hal_gpio.h  stm32l4xx_it.c ×
180= /**
181  * @brief This function handles System tick timer.
182  */
183= void SysTick_Handler(void)
184 {
185     /* USER CODE BEGIN SysTick_IRQn 0 */
186
187     /* USER CODE END SysTick_IRQn 0 */
188     HAL_IncTick();
189     /* USER CODE BEGIN SysTick_IRQn 1 */
190
191     /* USER CODE END SysTick_IRQn 1 */
192 }
193
194= /**
195  * STM32L4xx Peripheral Interrupt Handlers
196  * Add here the Interrupt Handlers for the used peripherals.
197  * For the available peripheral interrupt handler names,
198  * please refer to the startup file (startup_stm32l4xx.s).
199  */
200
201= /**
202  * @brief This function handles EXTI line[9:5] interrupts.
203  */
204= void EXTI9_5_IRQHandler(void)
205 {
206     /* USER CODE BEGIN EXTI9_5_IRQn 0 */
207
208     /* USER CODE END EXTI9_5_IRQn 0 */
209     HAL_GPIO_EXTI_IRQHandler(SPSGRF_915_GPIO3_EXTI5_Pin);
210     HAL_GPIO_EXTI_IRQHandler(SPBTLF_RF_IRQ_EXTI6_Pin);
211     HAL_GPIO_EXTI_IRQHandler(VL53L0X_GPIO1_EXTI7_Pin);
212     HAL_GPIO_EXTI_IRQHandler(LSM3MDL_DRDY_EXTI8_Pin);
213     /* USER CODE BEGIN EXTI9_5_IRQn 1 */
214
215     /* USER CODE END EXTI9_5_IRQn 1 */
216 }
217
218= /**
219  * @brief This function handles EXTI line[15:10] interrupts.
220  */
221= void EXTI15_10_IRQHandler(void)
222 {
223     /* USER CODE BEGIN EXTI15_10_IRQn 0 */
224
225     /* USER CODE END EXTI15_10_IRQn 0 */
226     HAL_GPIO_EXTI_IRQHandler(LPS22HB_INT_DRDY_EXTI0_Pin);
227     HAL_GPIO_EXTI_IRQHandler(LSM6DSL_INT1_EXTI11_Pin);
228     HAL_GPIO_EXTI_IRQHandler(BUTTON_EXTI13_Pin);
229     HAL_GPIO_EXTI_IRQHandler(ARD_D2_Pin);
230     HAL_GPIO_EXTI_IRQHandler(HTS221_DRDY_EXTI15_Pin);
231     /* USER CODE BEGIN EXTI15_10_IRQn 1 */
232
233     /* USER CODE END EXTI15_10_IRQn 1 */
234 }
235
236 /* USER CODE BEGIN 1 */
237
238 /* USER CODE END 1 */
239
```

Step 12. In stm32l4xx_hal_gpio.c, find HAL_GPIO_EXTI_IRQHandler code

```
Embedded-C-Assignment-5.ioc x main.c stm32l4xx_hal_gpio.h stm32l4xx_it.c stm32l4xx_hal_gpio.c x
481
482 /* Apply lock key write sequence */
483 tmp |= GPIO_Pin;
484 /* Set LCKx bit(s): LCKK='1' + LCK[15-0] */
485 GPIOx->LCKR = tmp;
486 /* Reset LCKx bit(s): LCKK='0' + LCK[15-0] */
487 GPIOx->LCKR = GPIO_Pin;
488 /* Set LCKx bit(s): LCKK='1' + LCK[15-0] */
489 GPIOx->LCKR = tmp;
490 /* Read LCKK register. This read is mandatory to complete key lock sequence */
491 tmp = GPIOx->LCKR;
492
493 /* Read again in order to confirm lock is active */
494 if ((GPIOx->LCKR & GPIO_LCKR_LCKK) != 0x00u)
495 {
496     return HAL_OK;
497 }
498 else
499 {
500     return HAL_ERROR;
501 }
502 }
503
504 /**
505  * @brief Handle EXTI interrupt request.
506  * @param GPIO_Pin Specifies the port pin connected to corresponding EXTI line.
507  * @retval None
508  */
509 void HAL_GPIO_EXTI_IRQHandler(uint16_t GPIO_Pin)
510 {
511     /* EXTI line interrupt detected */
512     if (__HAL_GPIO_EXTI_GET_IT(GPIO_Pin) != 0x00u)
513     {
514         __HAL_GPIO_EXTI_CLEAR_IT(GPIO_Pin);
515         HAL_GPIO_EXTI_Callback(GPIO_Pin);
516     }
517 }
518
519 /**
520  * @brief EXTI line detection callback.
521  * @param GPIO_Pin Specifies the port pin connected to corresponding EXTI line.
522  * @retval None
523  */
524 weak void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
525 {
526     /* Prevent unused argument(s) compilation warning */
527     UNUSED(GPIO_Pin);
528
529     /* NOTE: This function should not be modified, when the callback is needed,
530      the HAL_GPIO_EXTI_Callback could be implemented in the user file
531     */
532 }
533
534 /**
535  * @}
536  */
537
538
539 /**
540  * @}
```

Step 13. In main.c, add interrupt code to toggle LED on interrupt

```
Embedded-C-Assignment-5.ioc  main.c x stm32l4xx_it.c  stm32l4xx_hal_gpio.c
581  /*Configure GPIO pin : PMOD_SPI2_SCK_Pin */
582  GPIO_InitStruct.Pin = PMOD_SPI2_SCK_Pin;
583  GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
584  GPIO_InitStruct.Pull = GPIO_NOPULL;
585  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
586  GPIO_InitStruct.Alternate = GPIO_AF5_SPI2;
587  HAL_GPIO_Init(PMOD_SPI2_SCK_GPIO_Port, &GPIO_InitStruct);
588
589  /*Configure GPIO pins : PMOD_UART2_CTS_Pin PMOD_UART2_RTS_Pin PMOD_UART2_TX_Pin PMOD_UART2_RX_Pi
590  GPIO_InitStruct.Pin = PMOD_UART2_CTS_Pin|PMOD_UART2_RTS_Pin|PMOD_UART2_TX_Pin|PMOD_UART2_RX_Pi
591  GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
592  GPIO_InitStruct.Pull = GPIO_NOPULL;
593  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
594  GPIO_InitStruct.Alternate = GPIO_AF7_USART2;
595  HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);
596
597  /*Configure GPIO pins : ARD_D15_Pin ARD_D14_Pin */
598  GPIO_InitStruct.Pin = ARD_D15_Pin|ARD_D14_Pin;
599  GPIO_InitStruct.Mode = GPIO_MODE_AF_OD;
600  GPIO_InitStruct.Pull = GPIO_NOPULL;
601  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
602  GPIO_InitStruct.Alternate = GPIO_AF4_I2C1;
603  HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
604
605  /* EXTI interrupt init*/
606  HAL_NVIC_SetPriority(EXTI9_5_IRQn, 0, 0);
607  HAL_NVIC_EnableIRQ(EXTI9_5_IRQn);
608
609  HAL_NVIC_SetPriority(EXTI15_10_IRQn, 0, 0);
610  HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);
611
612  /* USER CODE BEGIN MX_GPIO_Init_2 */
613  /* USER CODE END MX_GPIO_Init_2 */
614  }
615
616  /* USER CODE BEGIN 4 */
617  void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
618  {
619      UNUSED(GPIO_Pin);
620
621      HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
622  }
623  /* USER CODE END 4 */
624
625  /**
626   * @brief This function is executed in case of error occurrence.
627   * @retval None
628   */
629  void Error_Handler(void)
630  {
631      /* USER CODE BEGIN Error_Handler_Debug */
632      /* User can add his own implementation to report the HAL error return state */
633      __disable_irq();
634      while (1)
635      {
636      }
637      /* USER CODE END Error_Handler_Debug */
638  }
639
```

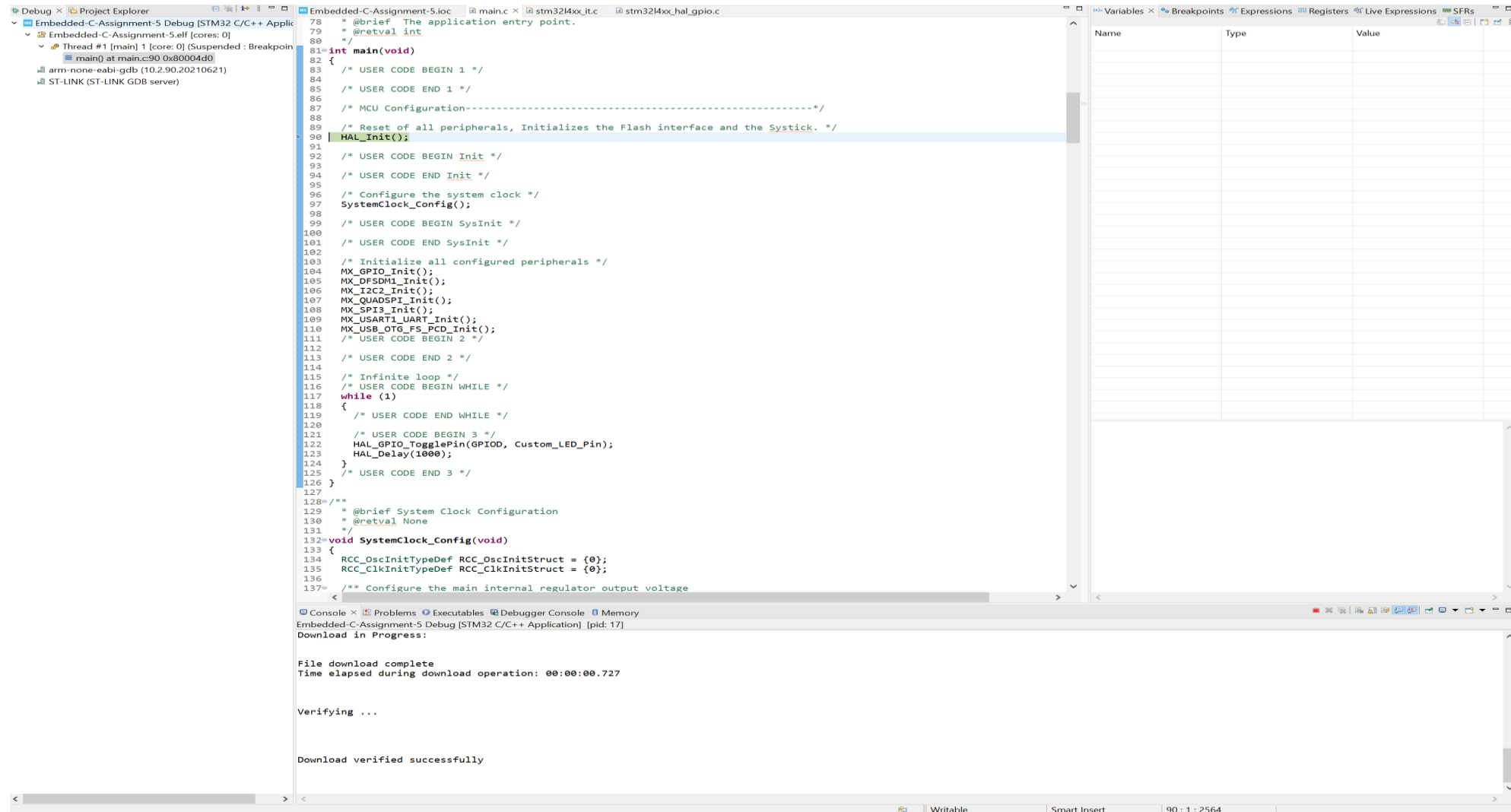
Step 14. In main.c, add code to toggle PD8 GPIO pin at 1 second rate

The screenshot displays an IDE environment with three main panels:

- Code Editor (Left):** Shows the source file `stm32l4xx_hal_gpio.c`. The code includes headers, defines pin numbers (`GPIOB`, `Custom_LED_Pin`), and implements functions like `SystemClock_Config` and `main`. It configures various peripherals including GPIO, DFSDM, I2C, SPI, UART, and USB.
- Console Window (Bottom Left):** Displays the build process output. It shows commands like `make -j8 all` and the successful compilation of `Embedded-C-Assignment-5.elf` using `arm-none-eabi-gcc`.
- Build Analyzer / Memory Regions (Bottom Right):** Provides details about the generated ELF file. A table summarizes the memory usage:

Region	Start address	End address	Size	Free	Used	Usage (%)
RAM	0x20000...	0x20017fff	96 KB	92.73 KB	3.27 KB	3.41%
RAM2	0x10000...	0x10007fff	32 KB	32 KB	0 B	0.00%
FLASH	0x08000...	0x080fffff	1024 KB	1005.72 ...	18.28 KB	1.79%

Step 15. Build and run the project in debug, when press button, you should see LED2 toggle, and every 1 second, you should also see PD8 toggle LED pin



Appendix. Attachment is the hardware connection for PD8 LED

