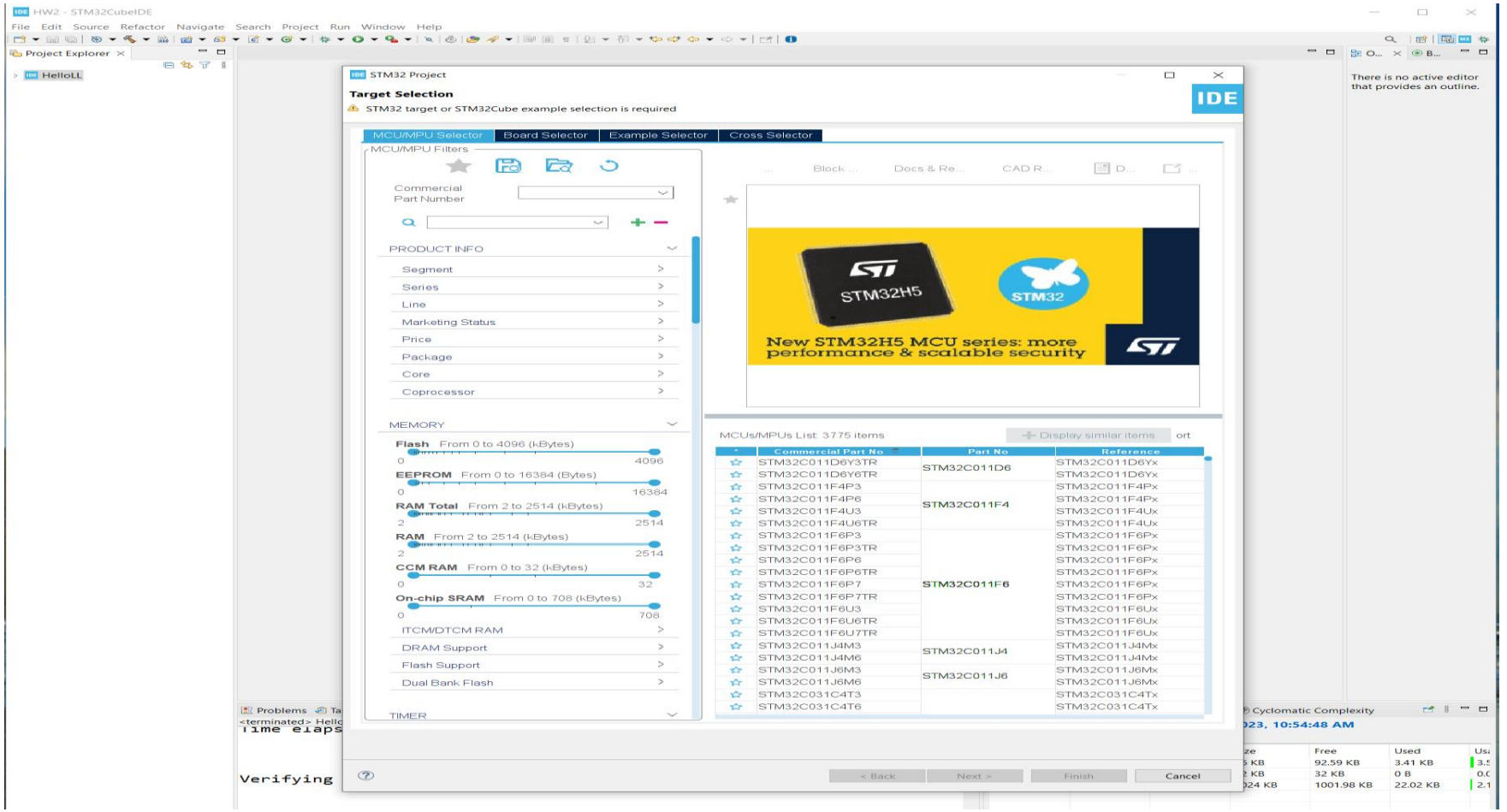
UCSD Embedded C Assignment 5

By

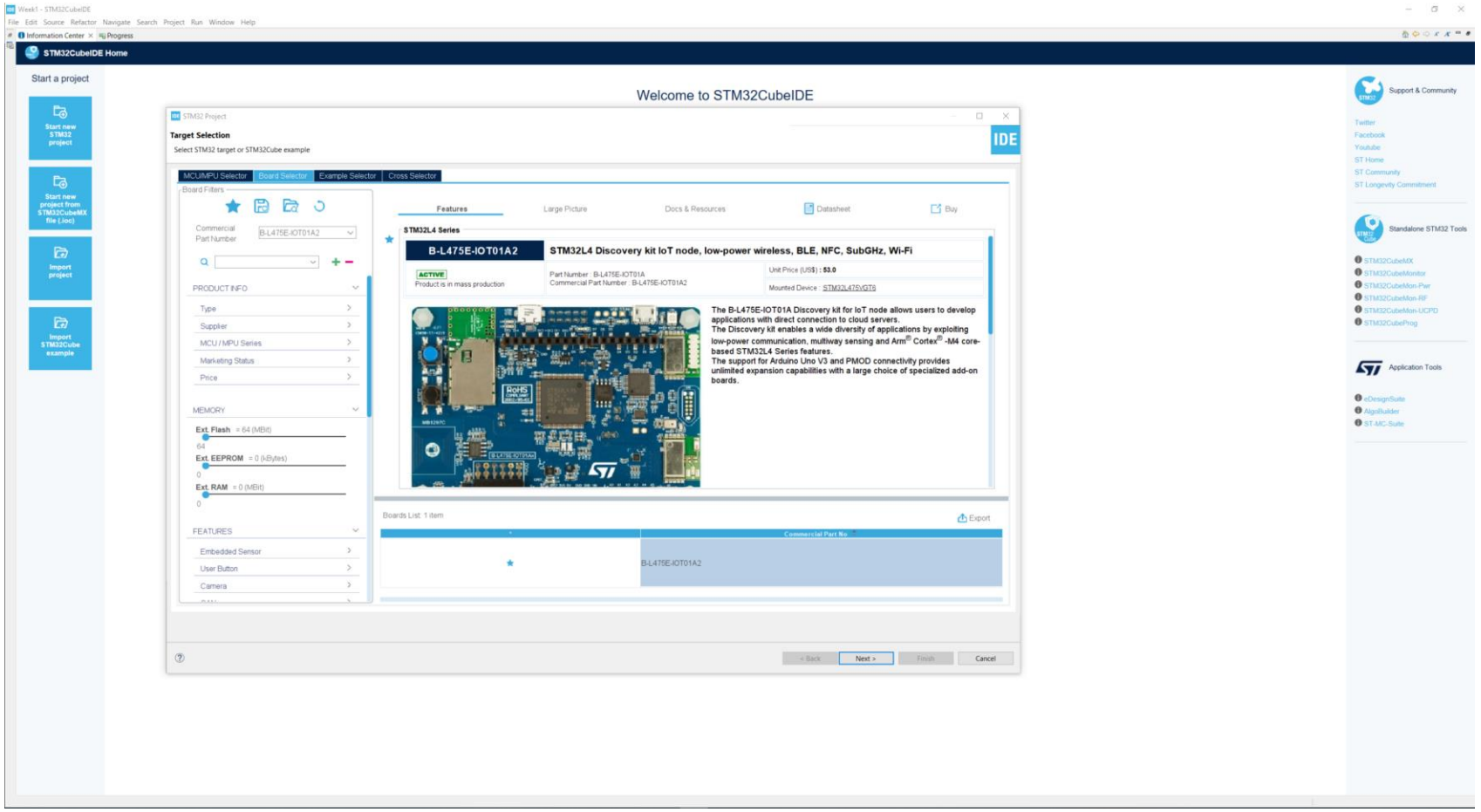
Hsuankai Chang

hsuankac@umich.edu

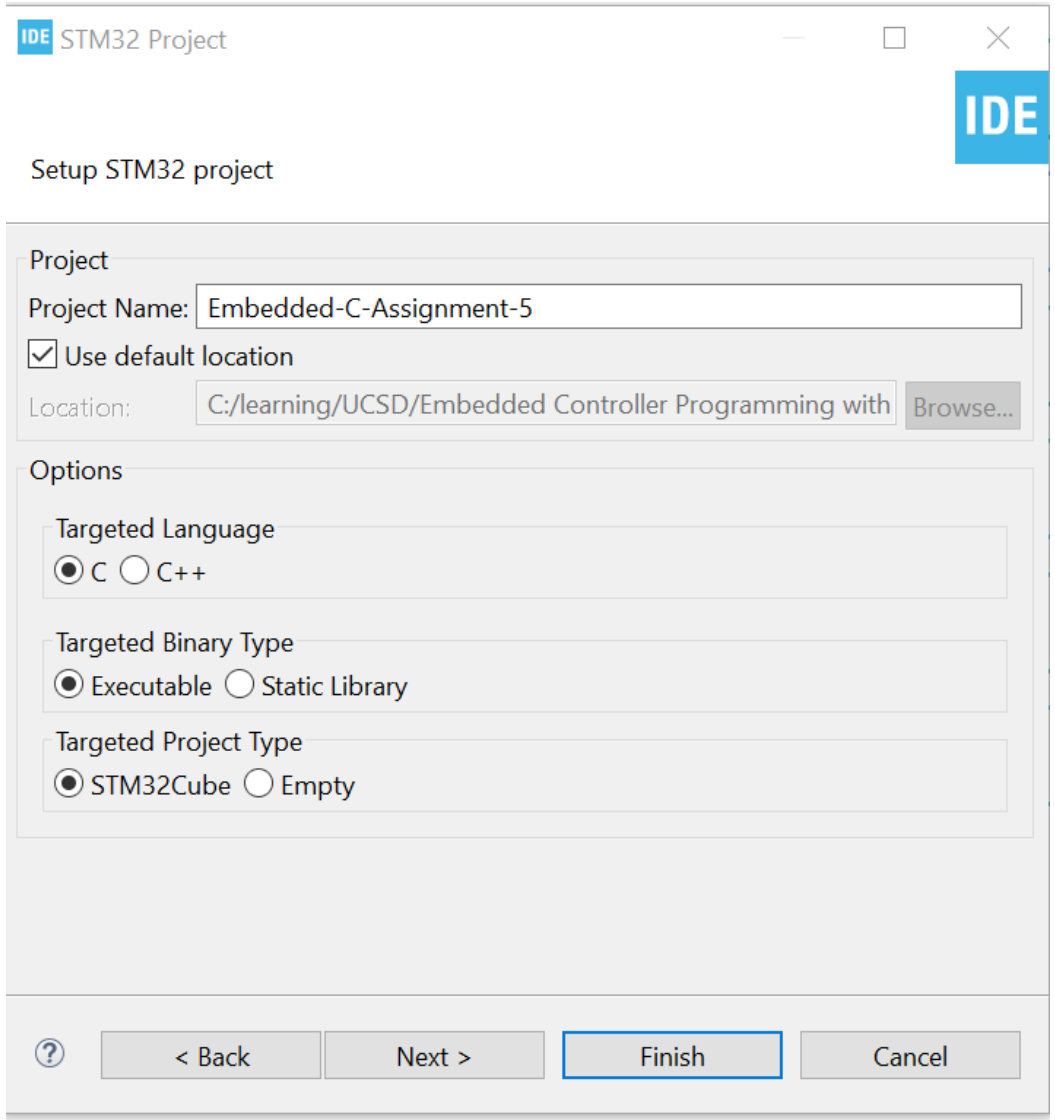
Step 1. Startup STM32CubeIDE and create new STM32 project



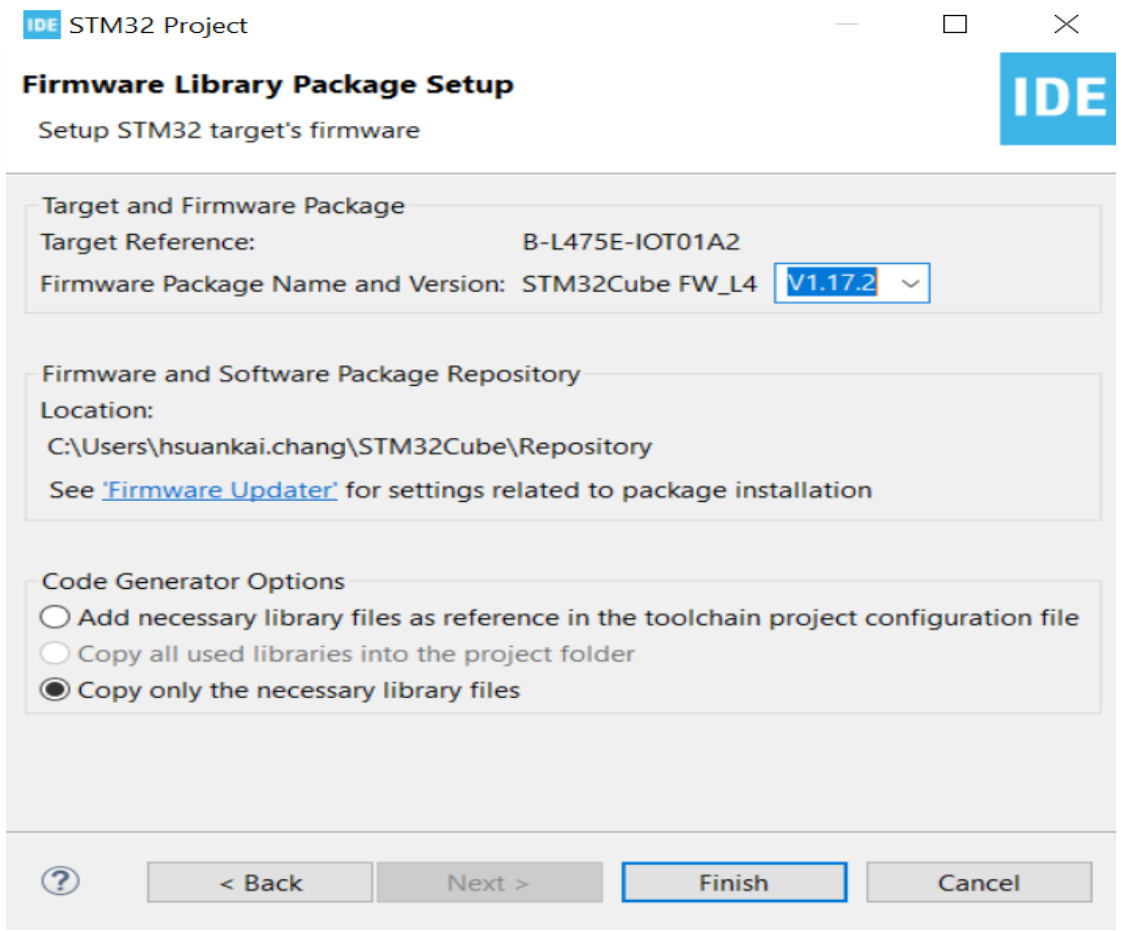
Step 2. Access board selector and type in the board you use, click Next



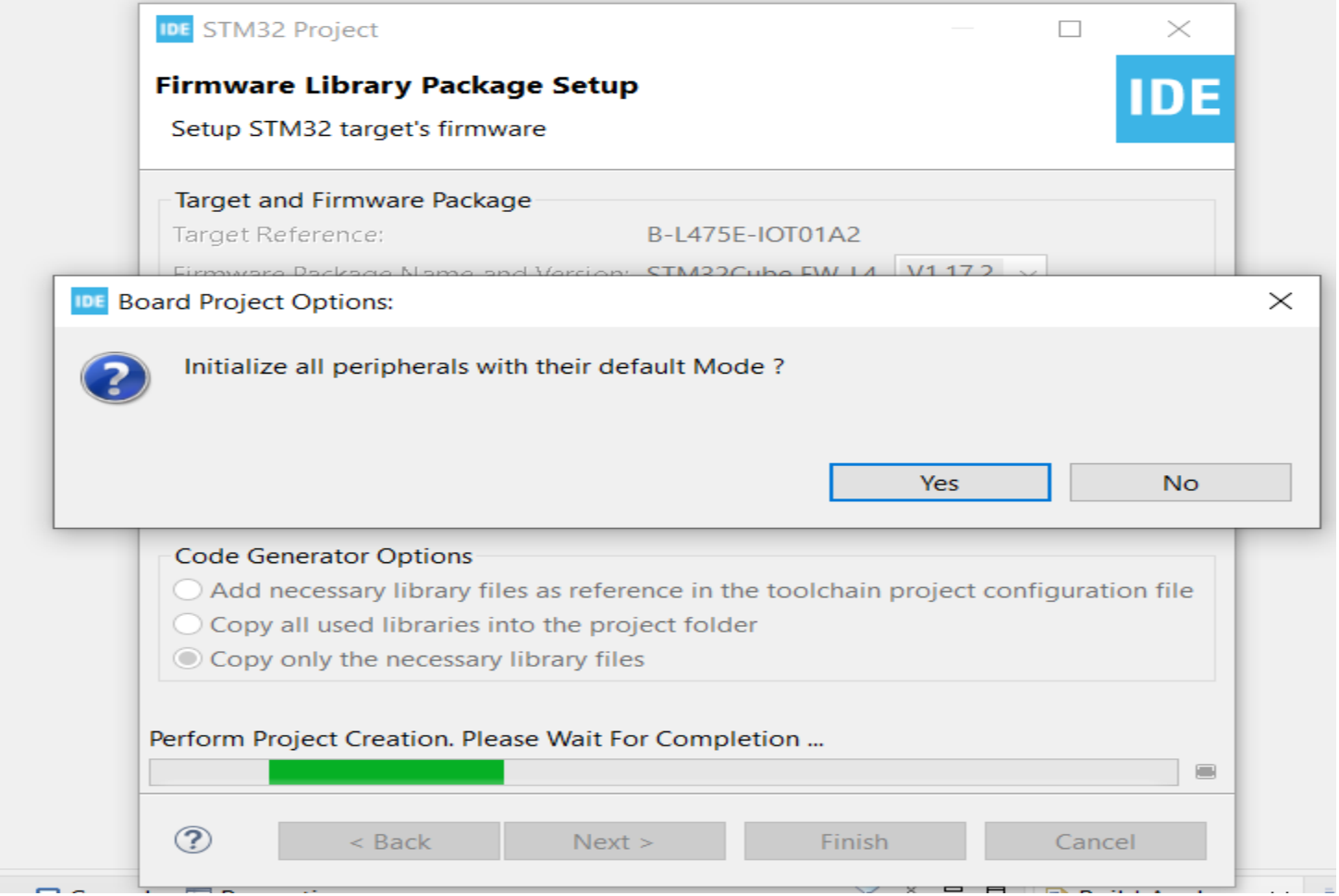
Step 3. Enter the project name then click Next



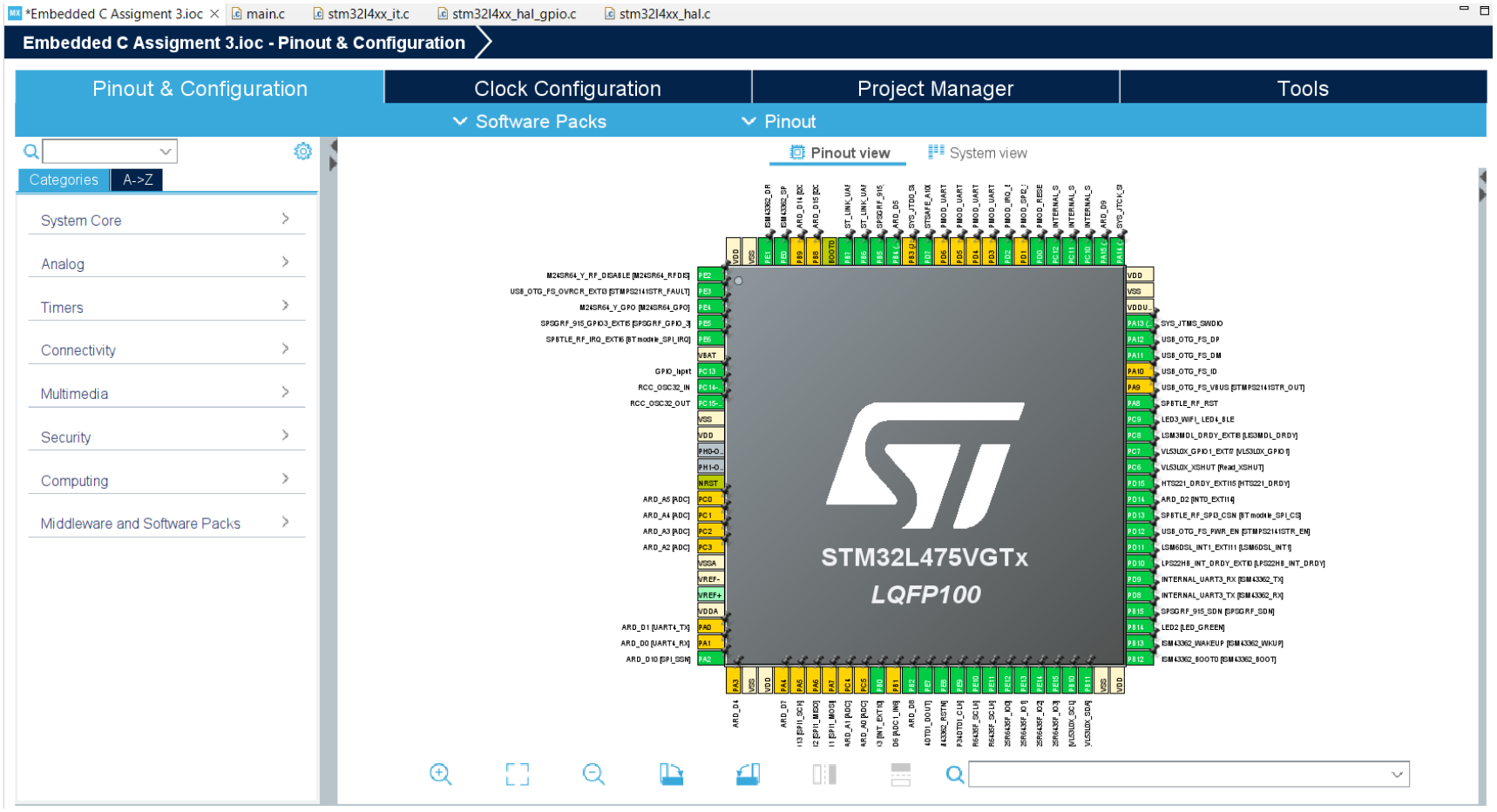
Step 4. See the firmware package name and version



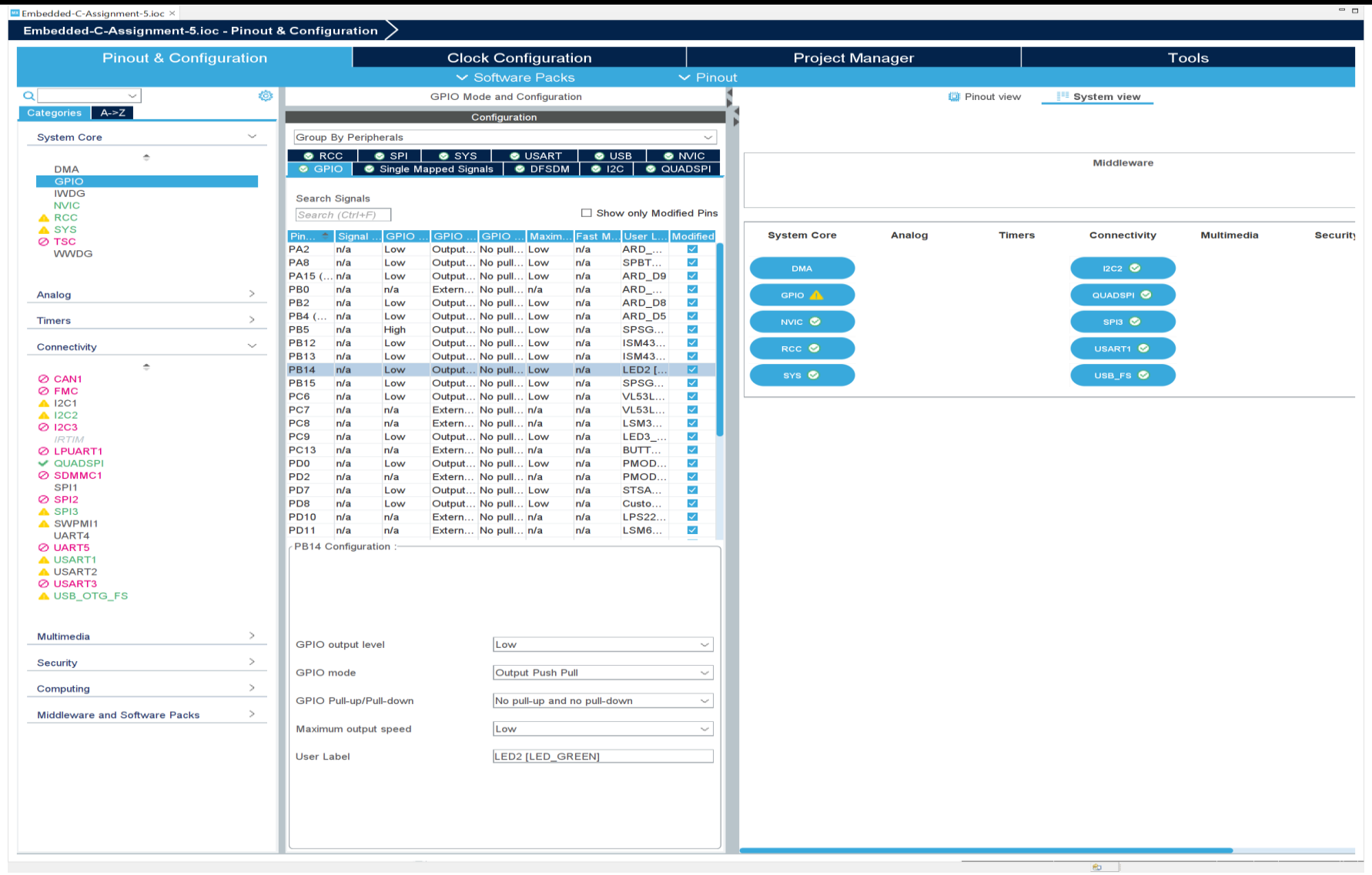
Step 5. Click yes to initialize all peripherals to default



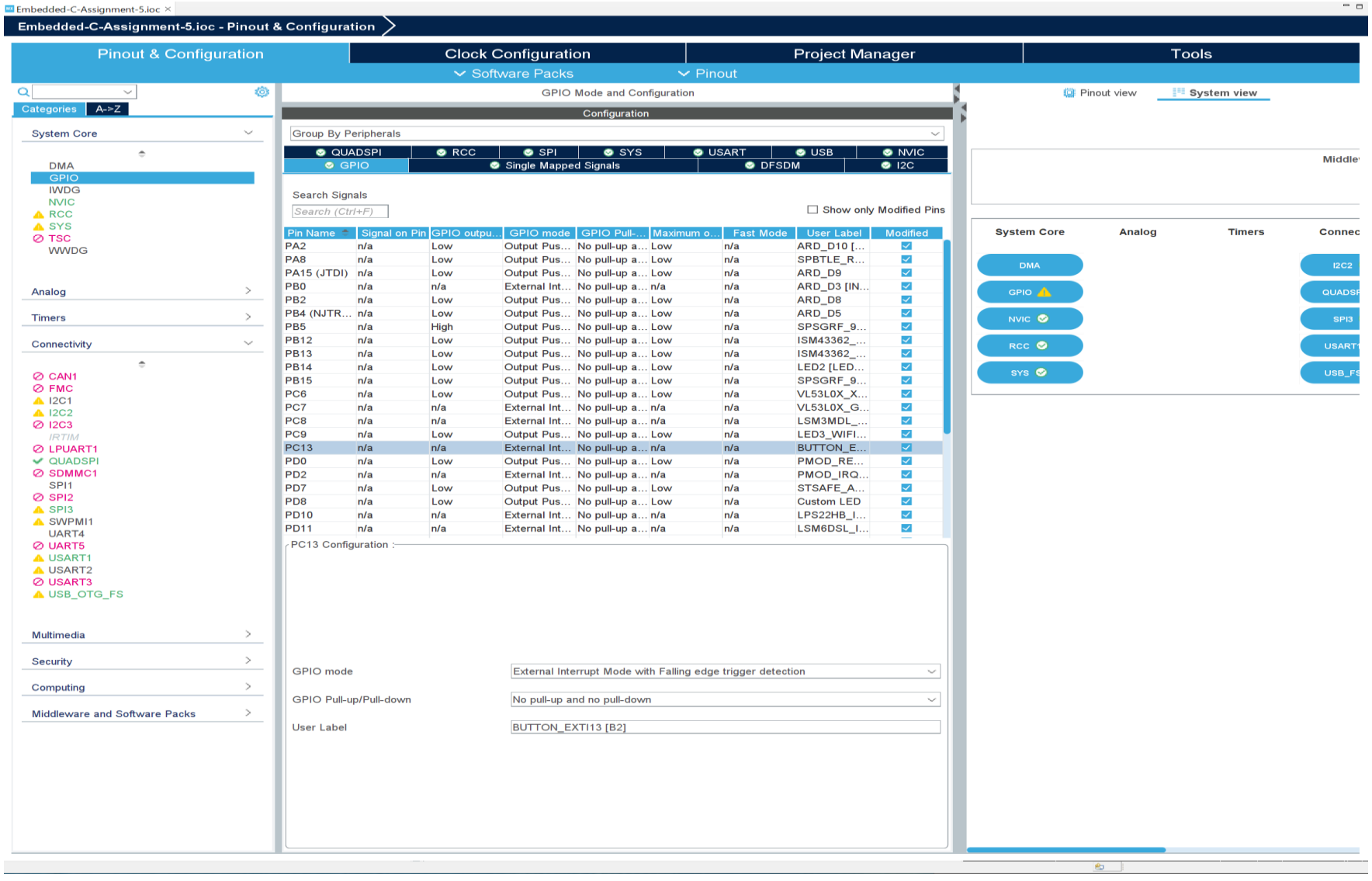
Step 6. When in .ioc file, click Pinout & Configurations



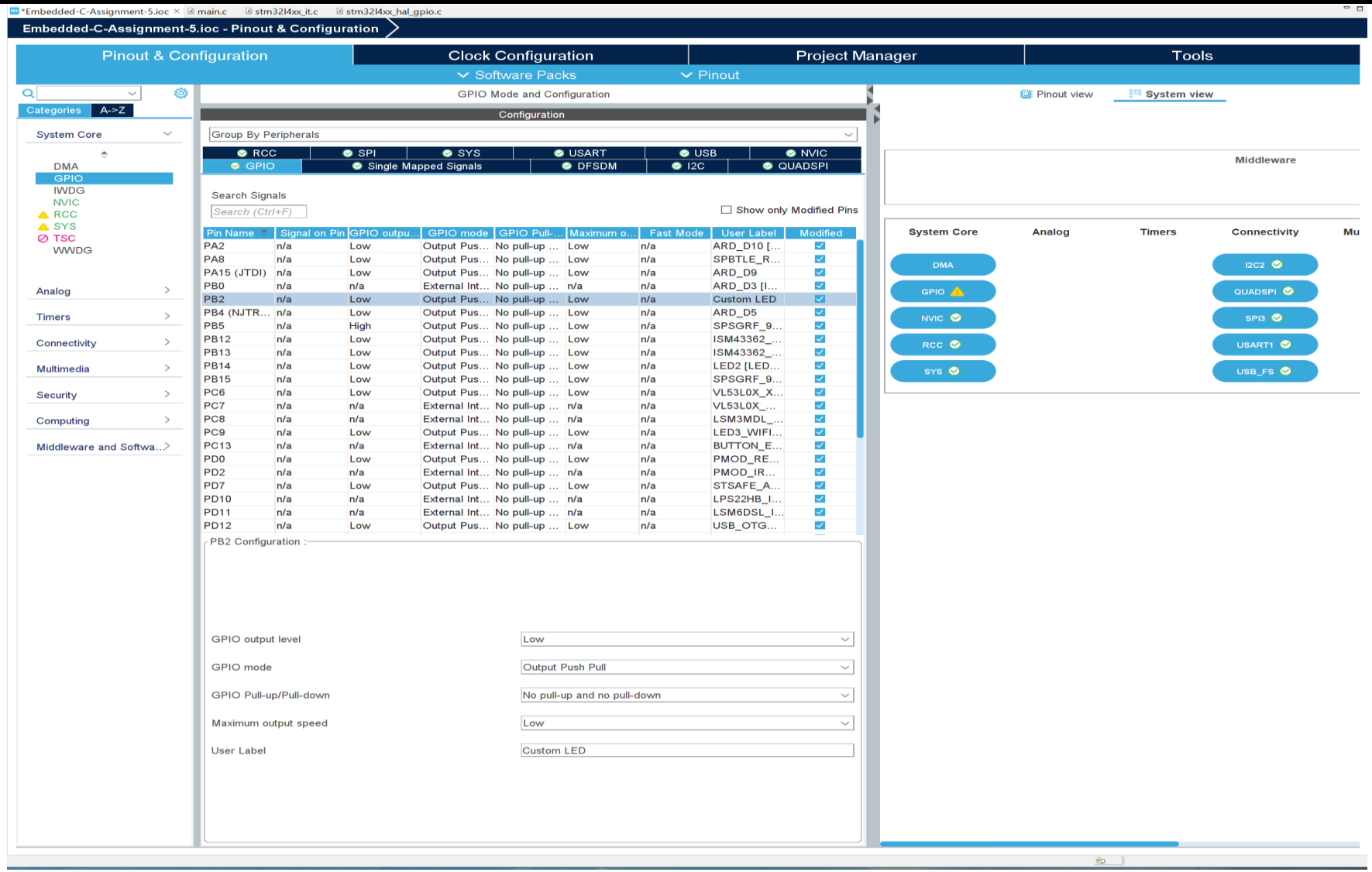
Step 7. Review the LED2 GPIO configuration by clicking System view -> GPIO -> PB14



Step 8. Review the BUTTON GPIO configuration



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

```
/**
  * @brief This function handles System tick timer.
  */
void SysTick_Handler(void)
{
  /* USER CODE BEGIN SysTick_IRQn 0 */

  /* USER CODE END SysTick_IRQn 0 */
  HAL_IncTick();
  /* USER CODE BEGIN SysTick_IRQn 1 */

  /* USER CODE END SysTick_IRQn 1 */
}

/******************************************************************************/
/* STM32L4xx Peripheral Interrupt Handlers                                    */
/* Add here the Interrupt Handlers for the used peripherals.                  */
/* For the available peripheral interrupt handler names,                      */
/* please refer to the startup file (startup_stm32l4xx.s).                    */
/******************************************************************************/

/**
  * @brief This function handles EXTI line[9:5] interrupts.
  */
void EXTI9_5_IRQHandler(void)
{
  /* USER CODE BEGIN EXTI9_5_IRQn 0 */

  /* USER CODE END EXTI9_5_IRQn 0 */
  HAL_GPIO_EXTI_IRQHandler(SPSGRF_915_GPIO3_EXTI5_Pin);
  HAL_GPIO_EXTI_IRQHandler(SPBTLE_RF_IRQ_EXTI6_Pin);
  HAL_GPIO_EXTI_IRQHandler(VL53L0X_GPIO1_EXTI7_Pin);
  HAL_GPIO_EXTI_IRQHandler(LSM3MDL_DRDY_EXTI8_Pin);
  /* USER CODE BEGIN EXTI9_5_IRQn 1 */

  /* USER CODE END EXTI9_5_IRQn 1 */
}

/**
  * @brief This function handles EXTI line[15:10] interrupts.
  */
void EXTI15_10_IRQHandler(void)
{
  /* USER CODE BEGIN EXTI15_10_IRQn 0 */

  /* USER CODE END EXTI15_10_IRQn 0 */
  HAL_GPIO_EXTI_IRQHandler(LPS22HB_INT_DRDY_EXTI0_Pin);
  HAL_GPIO_EXTI_IRQHandler(LSM6DSL_INT1_EXTI11_Pin);
  HAL_GPIO_EXTI_IRQHandler(BUTTON_EXTI13_Pin);
  HAL_GPIO_EXTI_IRQHandler(ARD_D2_Pin);
  HAL_GPIO_EXTI_IRQHandler(HTS221_DRDY_EXTI15_Pin);
  /* USER CODE BEGIN EXTI15_10_IRQn 1 */

  /* USER CODE END EXTI15_10_IRQn 1 */
}

/* USER CODE BEGIN 1 */

/* USER CODE END 1 */

```

Step 12. In stm32l4xx_hal_gpio.c, find HAL_GPIO_EXTI_IRQHandler code

Embedded-C-Assignment-5.ioc | main.c | stm32l4xx_hal_gpio.h | stm32l4xx_it.c | stm32l4xx_hal_gpio.c

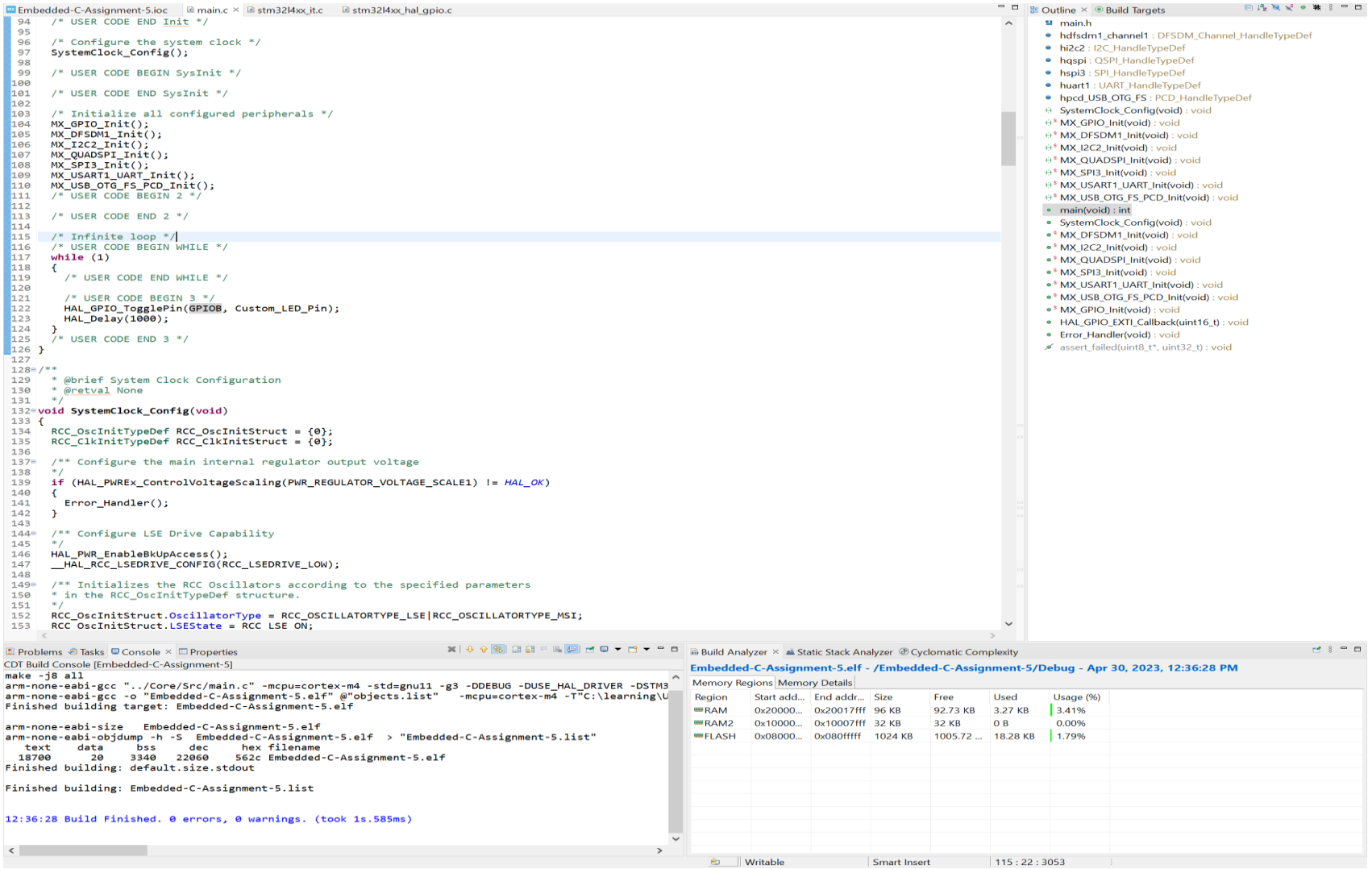
```
481
482   /* Apply lock key write sequence */
483   tmp |= GPIO_Pin;
484   /* Set LCKx bit(s): LCKK='1' + LCK[15-0] */
485   GPIOx->LCKR = tmp;
486   /* Reset LCKx bit(s): LCKK='0' + LCK[15-0] */
487   GPIOx->LCKR = GPIO_Pin;
488   /* Set LCKx bit(s): LCKK='1' + LCK[15-0] */
489   GPIOx->LCKR = tmp;
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

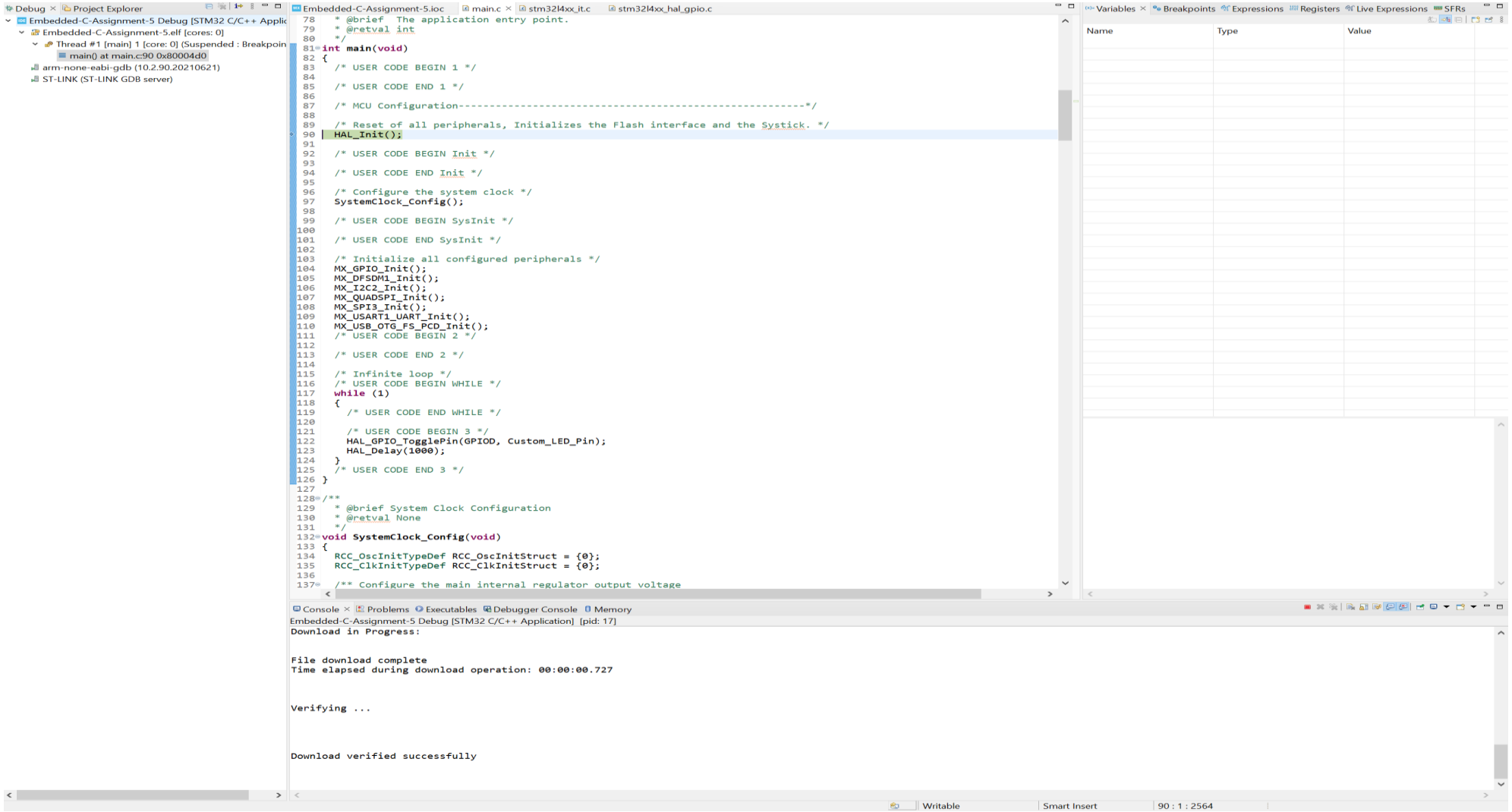
Embedded-C-Assignment-5.ioc | main.c | stm32l4xx_it.c | stm32l4xx_hal_gpio.c

```
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_
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(GPIOD, &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 }
```

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



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