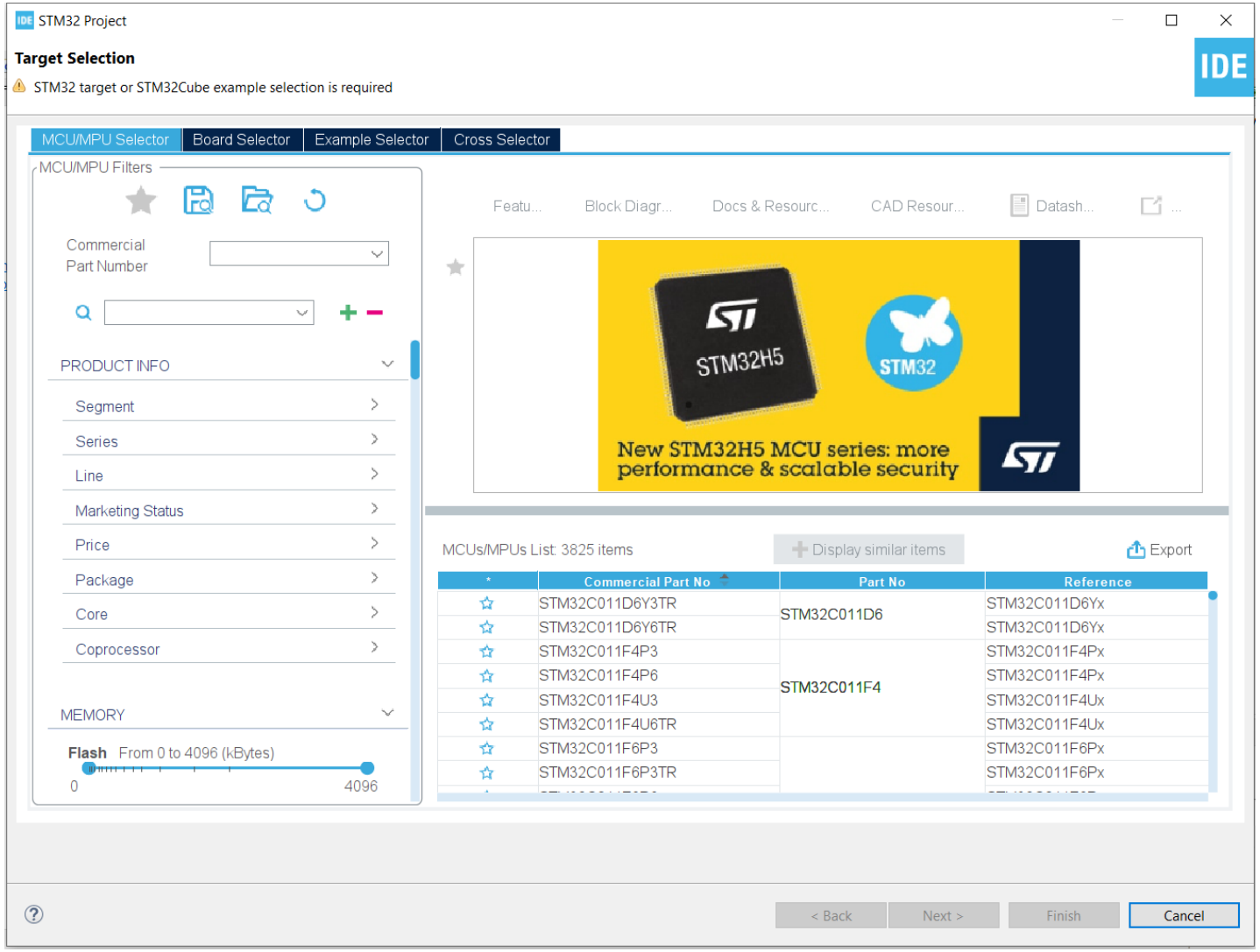
UCSD Embedded C Assignment 8

By

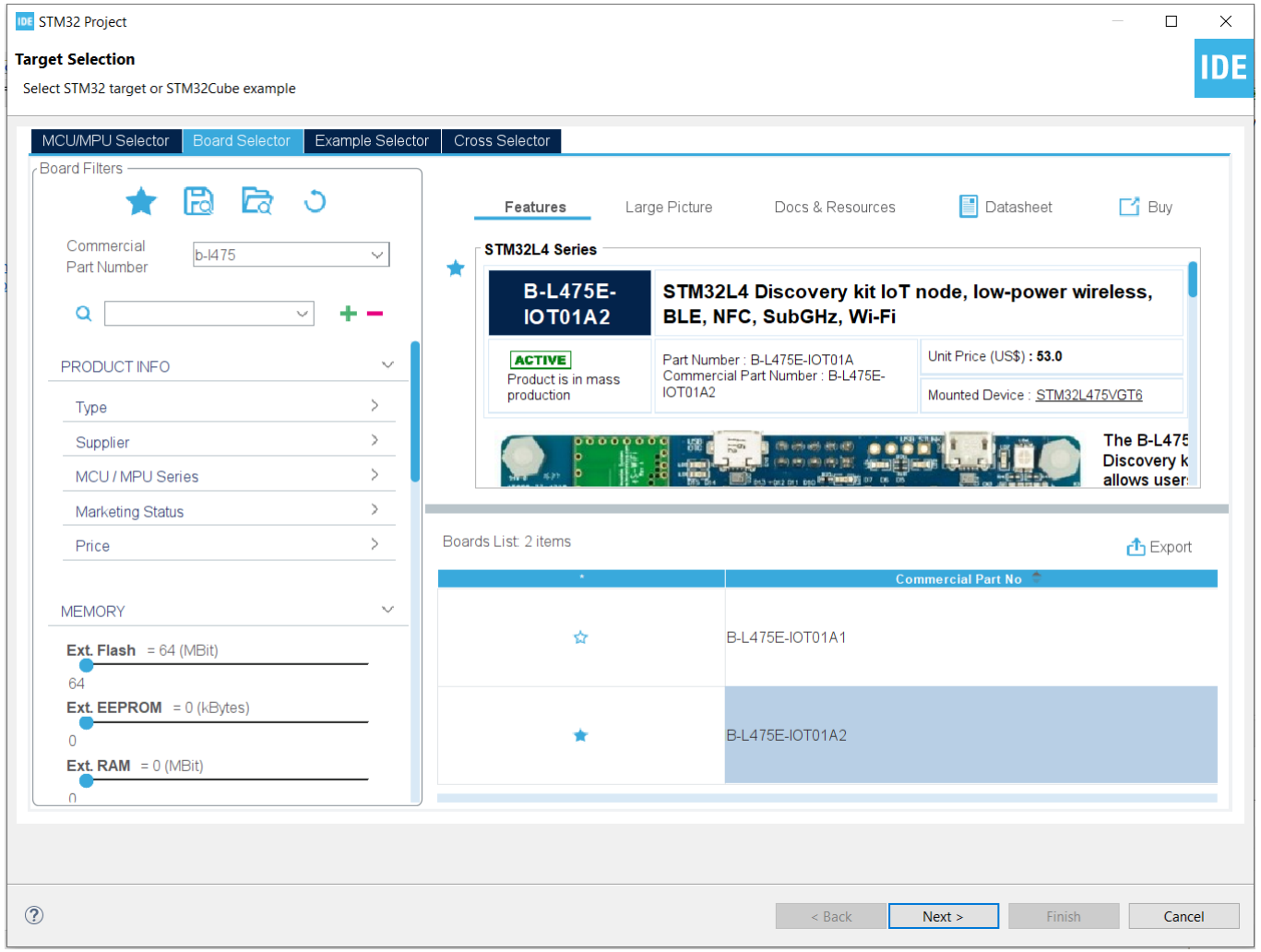
Hsuankai Chang

hsuankac@umich.edu

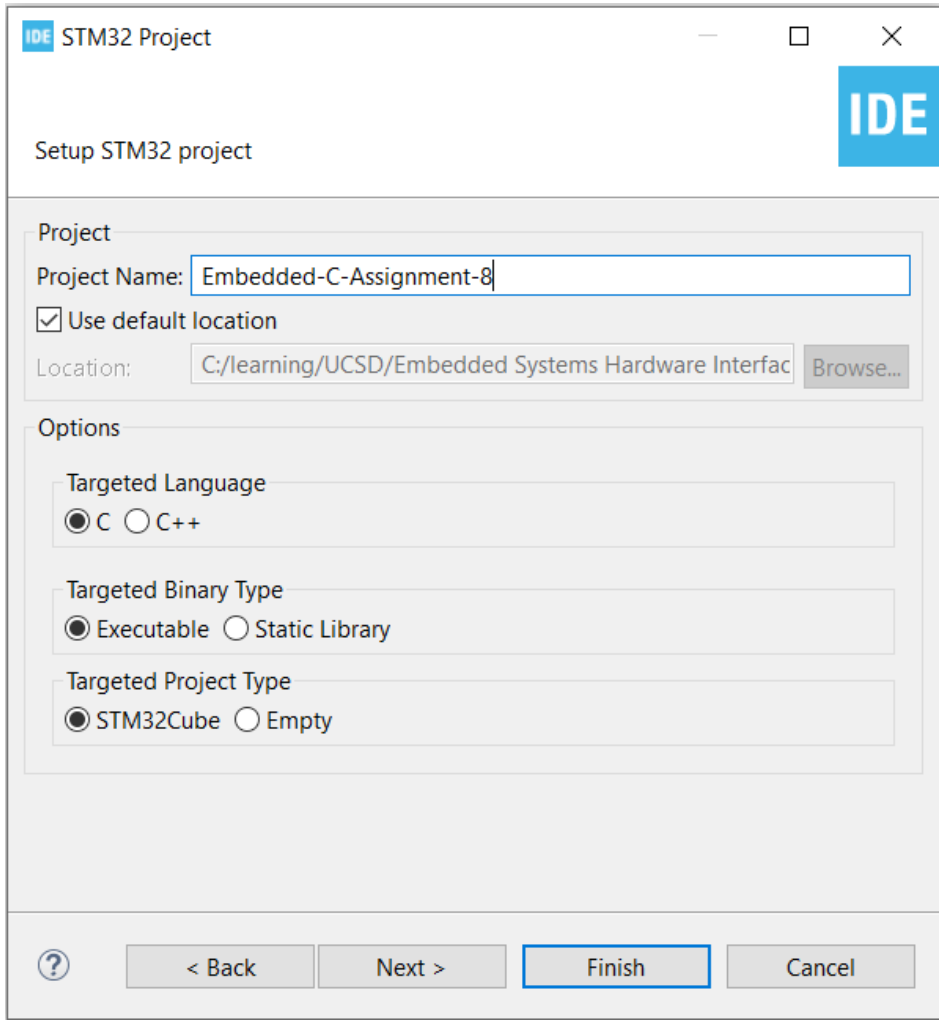
Step 1. Startup STM32CubeIDE and create new STM32 project



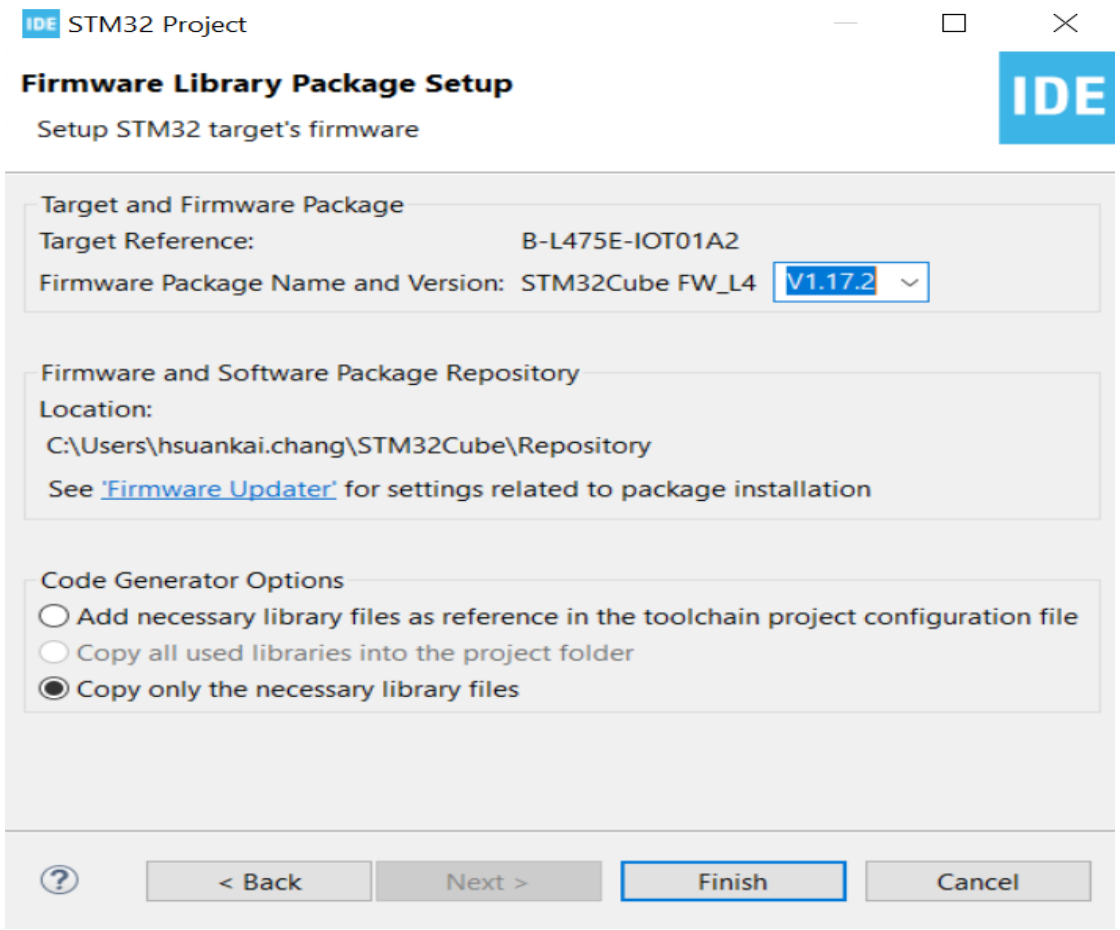
Step 2. Access board selector and type in the board you use, click Next



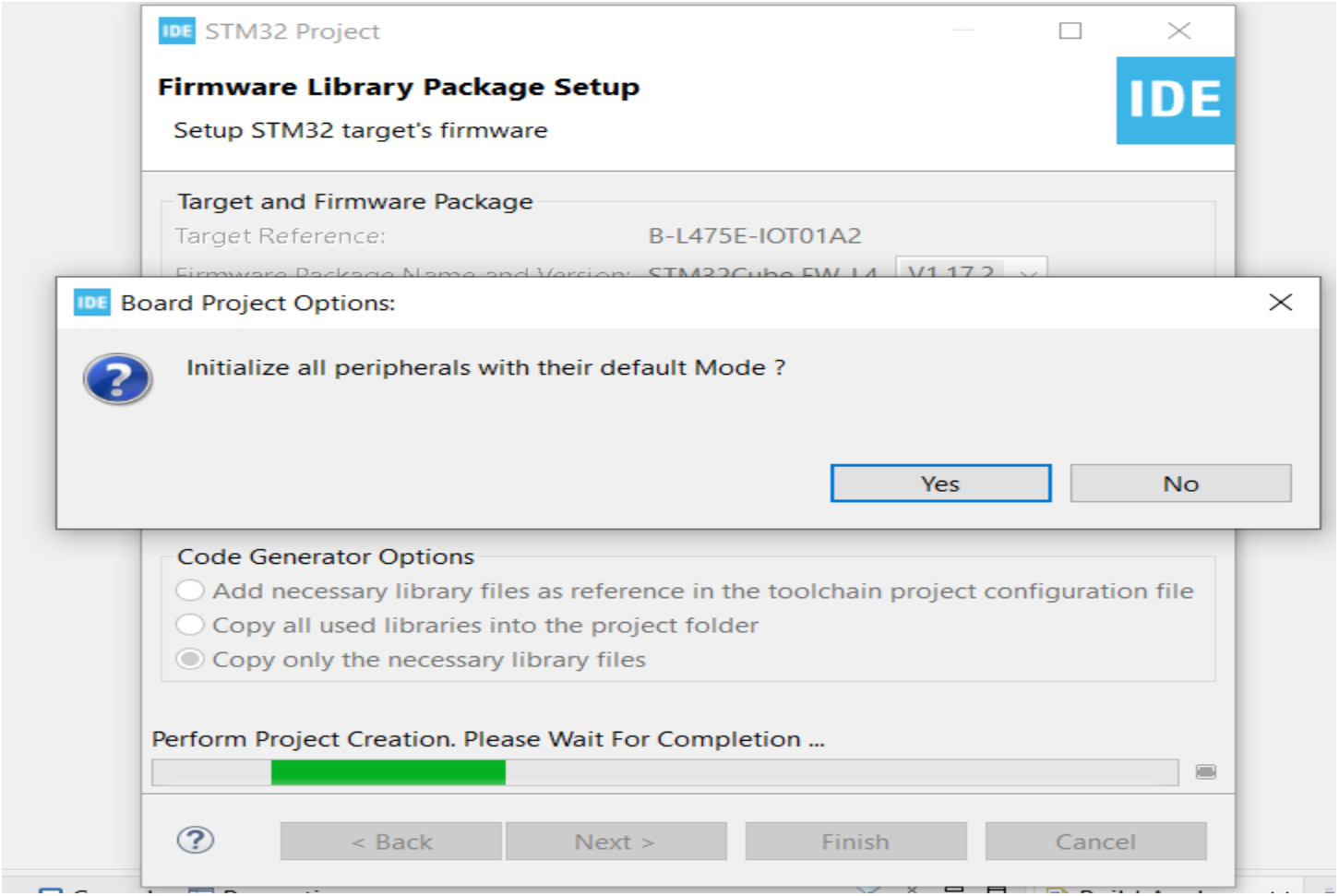
Step 3. Enter the project name then click Next



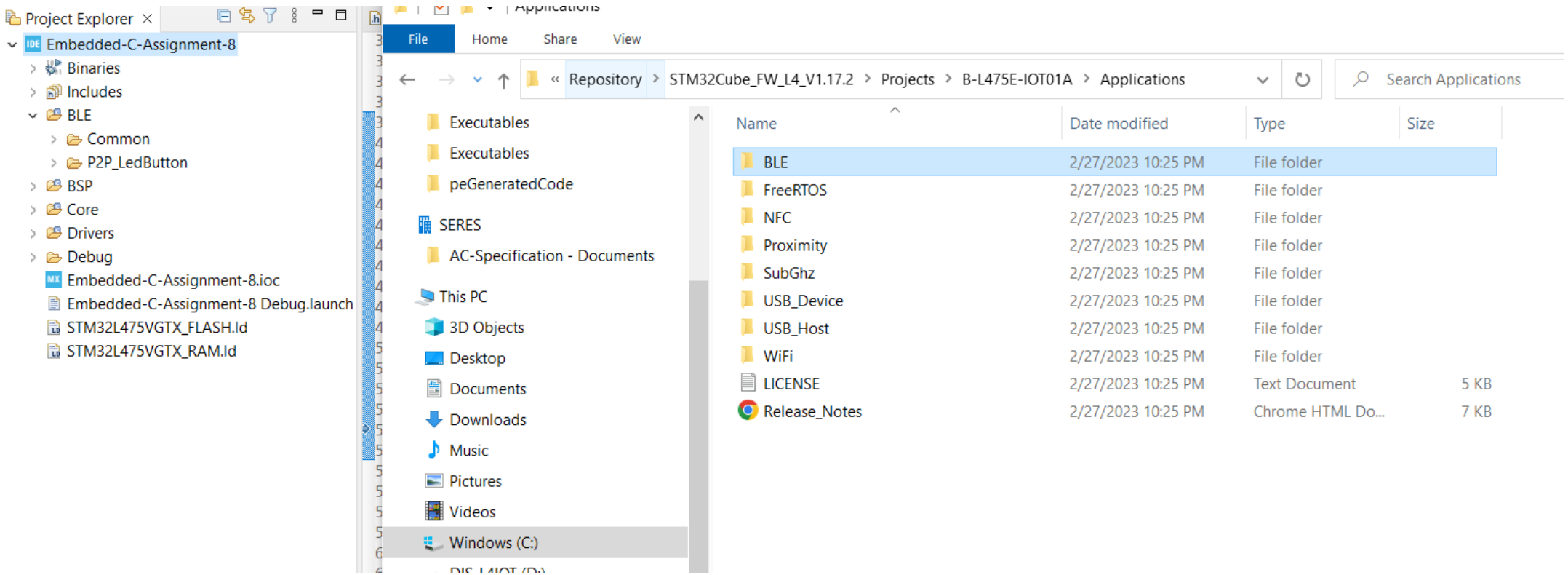
Step 4. See the firmware package name, version and location



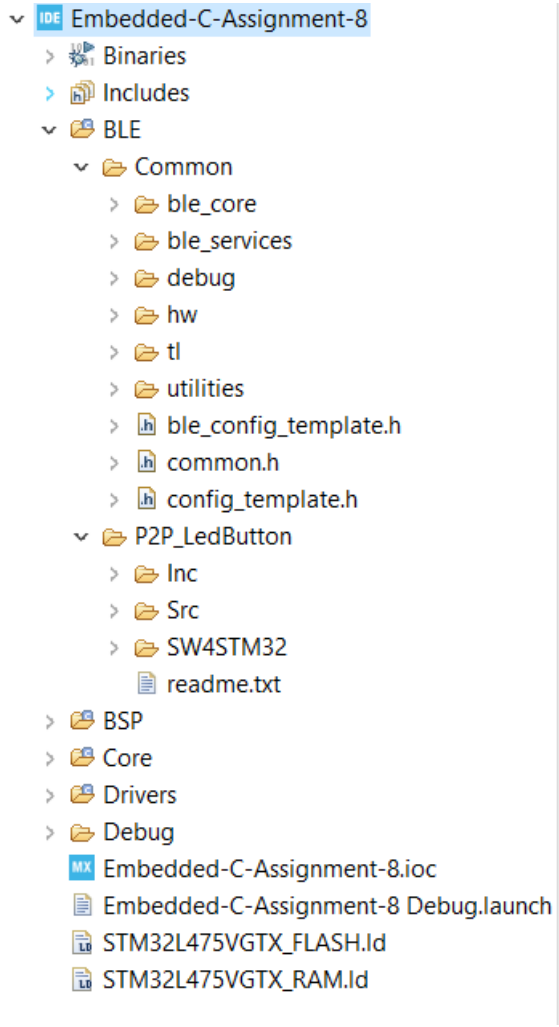
Step 5. Click yes to initialize all peripherals to default



Step 6. Copy the BLE folder into the project, and delete the HeartRate folder



Step 7. Delete EWARM and MDK-ARM folder



Step 8. Add the BLE source path and include path

Includes | # Symbols | Libraries | Library Paths | Source Location | References

Languages
- GNU C
- Assembly

Include directories

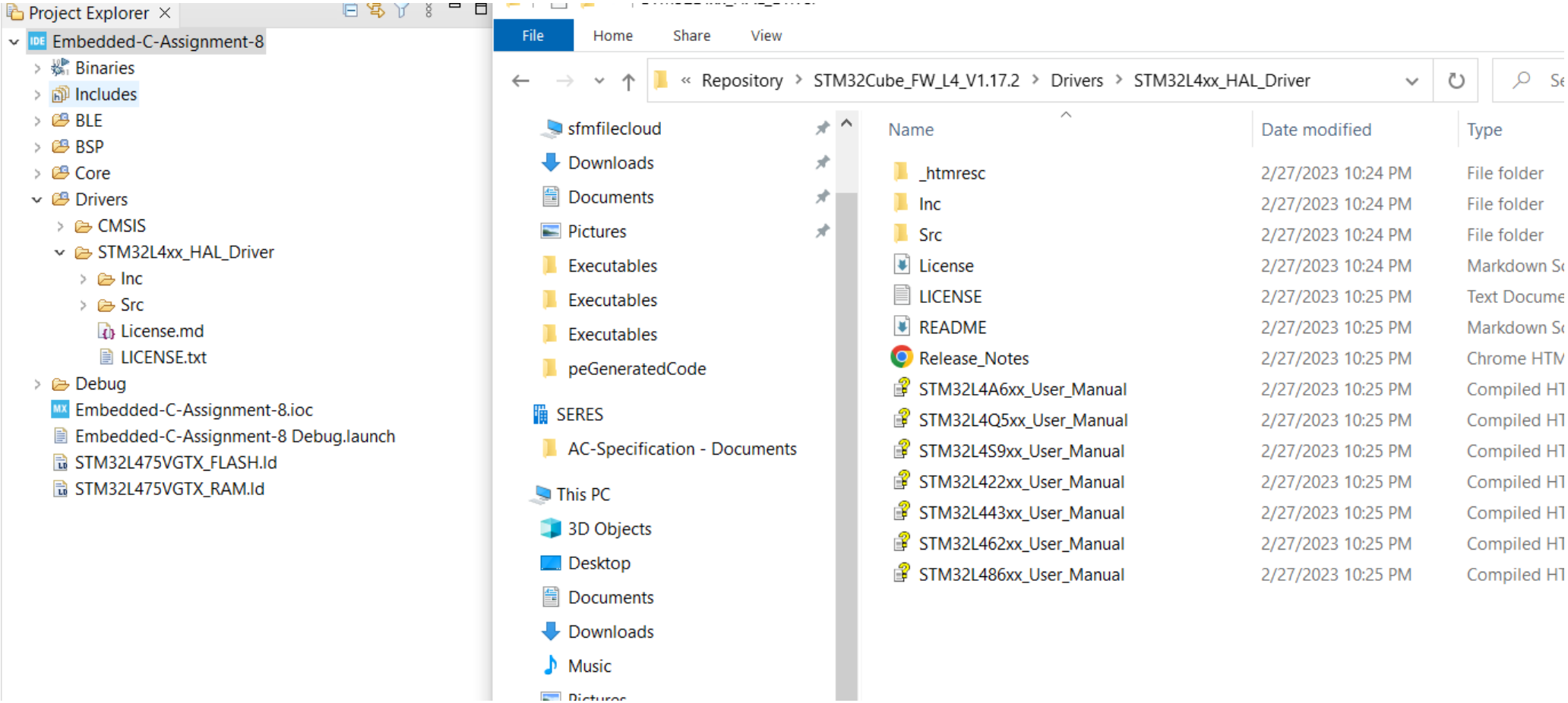
- Drivers/STM32L4xx_HAL_Driver/Inc
- Drivers/STM32L4xx_HAL_Driver/Inc/Legacy
- Drivers/CMSIS/Device/ST/STM32L4xx/Include
- Drivers/CMSIS/Include
- /Embedded-C-Assignment-8/BLE/Common/ble_core
- /Embedded-C-Assignment-8/BLE/Common/ble_services
- /Embedded-C-Assignment-8/BLE/Common/debug
- /Embedded-C-Assignment-8/BLE/Common/hw
- /Embedded-C-Assignment-8/BLE/Common/tl
- /Embedded-C-Assignment-8/BLE/Common/utilities
- /Embedded-C-Assignment-8/BLE/P2P_LedButton/Inc
- /Embedded-C-Assignment-8/BLE/Common

Includes | # Symbols | Libraries | Library Paths | Source Location | References

Source folders on build path:

> /Embedded-C-Assignment-8/BLE

Step 9. Copy the missing LL and HAL header and source file into the project, which includes stm32l4xx_ll_pwr.h, stm32l4xx_ll_rtc.h, stm32l4xx_ll_cortex.h, stm32l4xx_ll_spi.h, stm32l4xx_ll_dma.h, stm32l4xx_hal_rtc.h, stm32l4xx_hal_rtc_ex.h



Step 10. Add stm32l4xx_hal_rtc.h in hw.h file

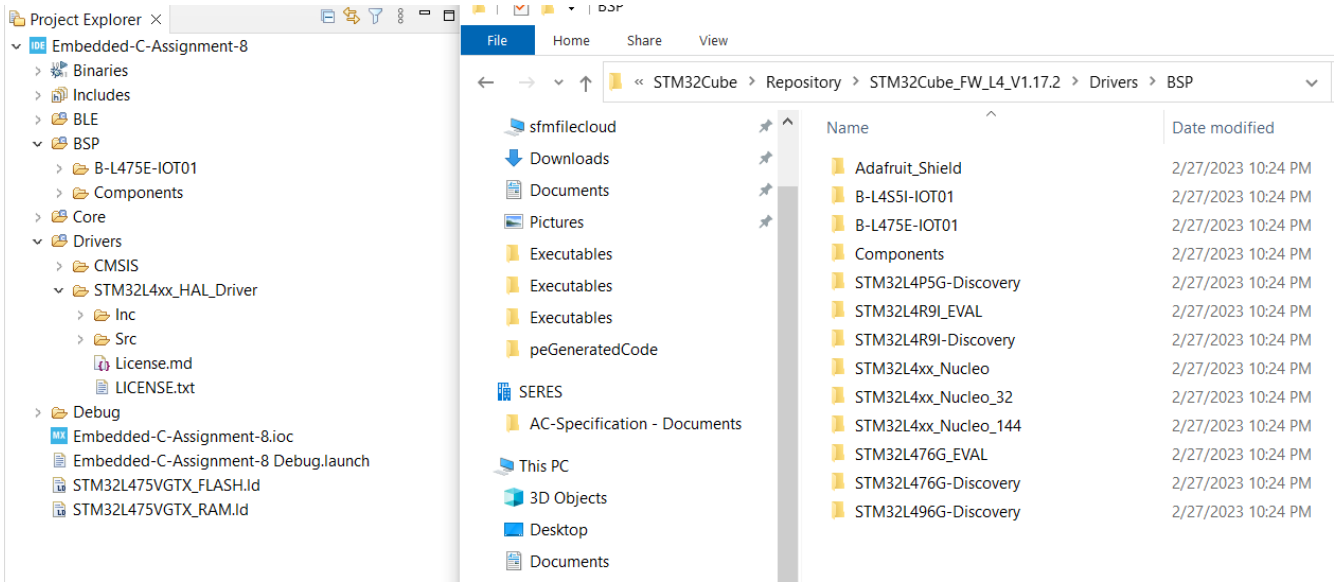
Project Explorer

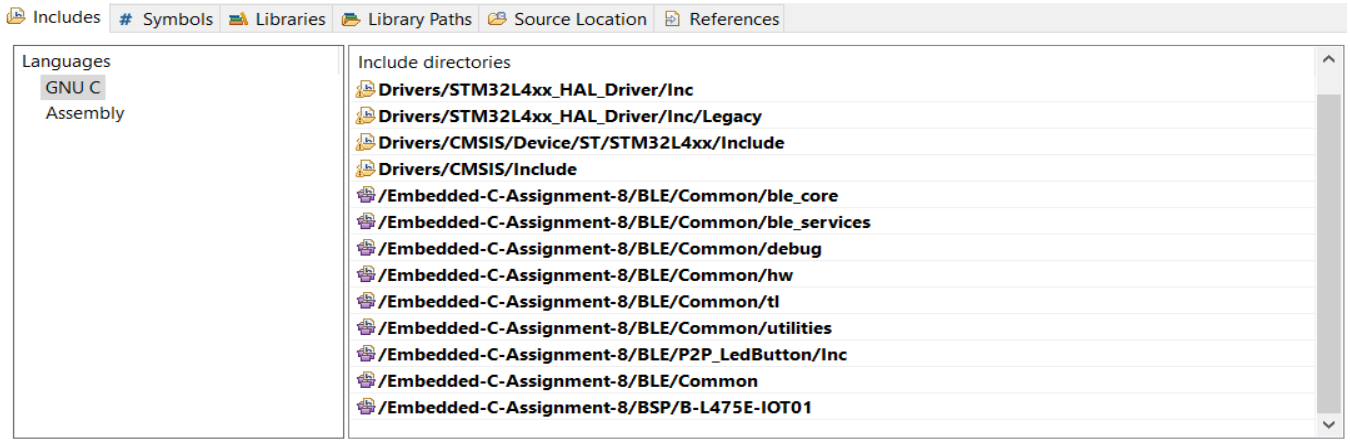
- Embedded-C-Assignment-8
  - Binaries
  - Includes
  - BLE
    - Common
      - ble_core
      - ble_services
      - debug
      - hw
        - hw_lpm.c
        - hw_spi.c
        - hw_timerserver.c
        - hw.h
      - tl
      - utilities
      - ble_config_template.h
      - common.h
      - config_template.h
    - P2P_LedButton
      - Inc

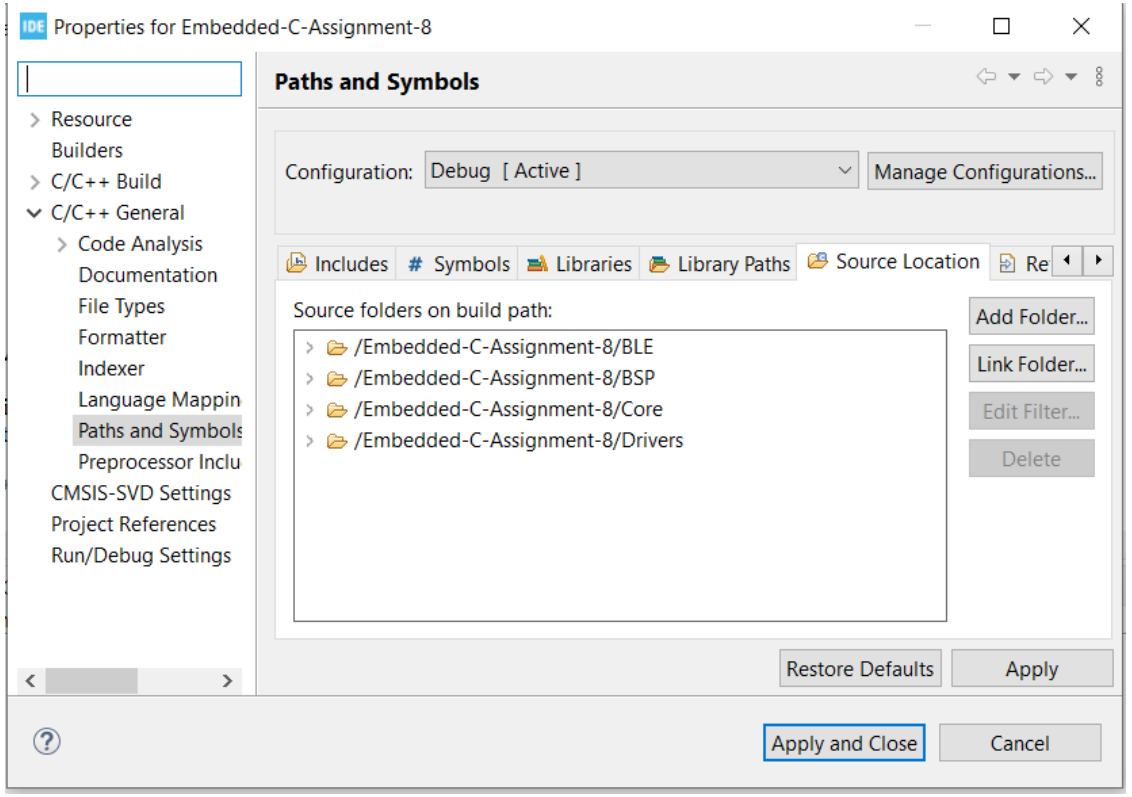
hw.h | tl_ble_hci.c | stm32l4xx_ha... | main.c | hw_timerser...

```c
/* Define to prevent recursive inclusion ---------------
#ifndef __HW_H
#define __HW_H

#ifdef __cplusplus
extern "C" {
#endif

  /* Includes ----------------------------------------
#include "stm32l4xx.h"
#include "stm32l4xx_ll_pwr.h"
#include "stm32l4xx_ll_cortex.h"
#include "stm32l4xx_ll_rtc.h"
#include "stm32l4xx_ll_spi.h"
#include "stm32l4xx_ll_dma.h"

#include "stm32l4xx_hal_rtc.h"



  /*****************************************************
   * HW Low Power Management (lpm)
   *****************************************************
```

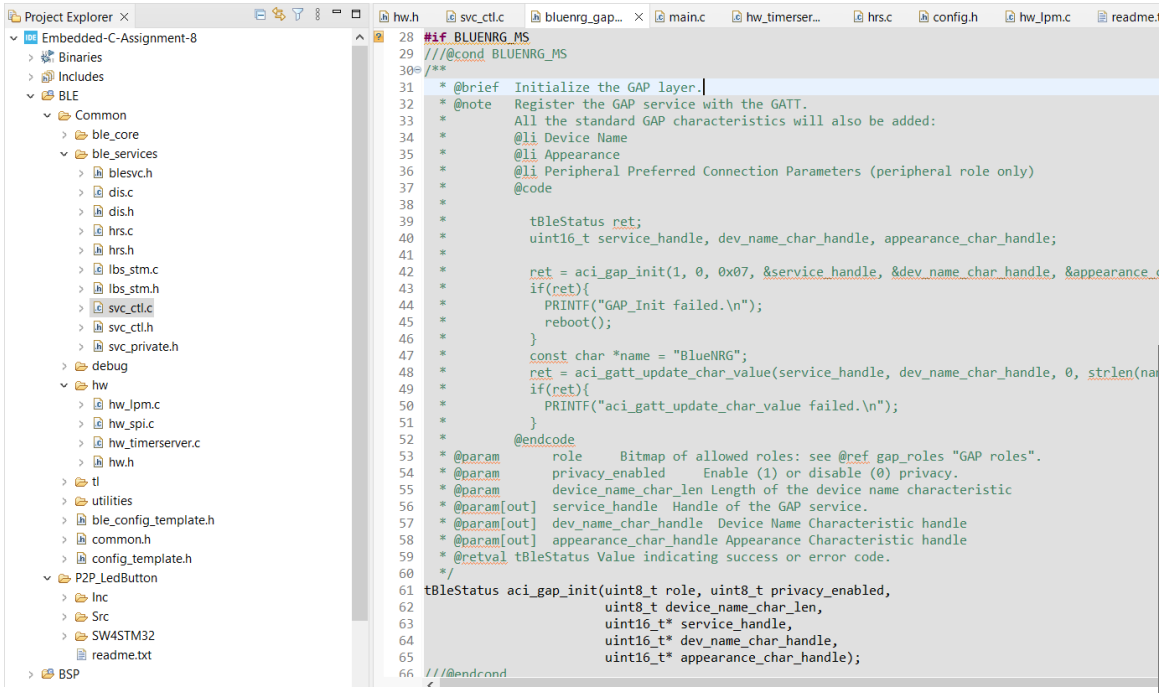
Step 11. Add BSP folders into the project, and also add the include path and source path

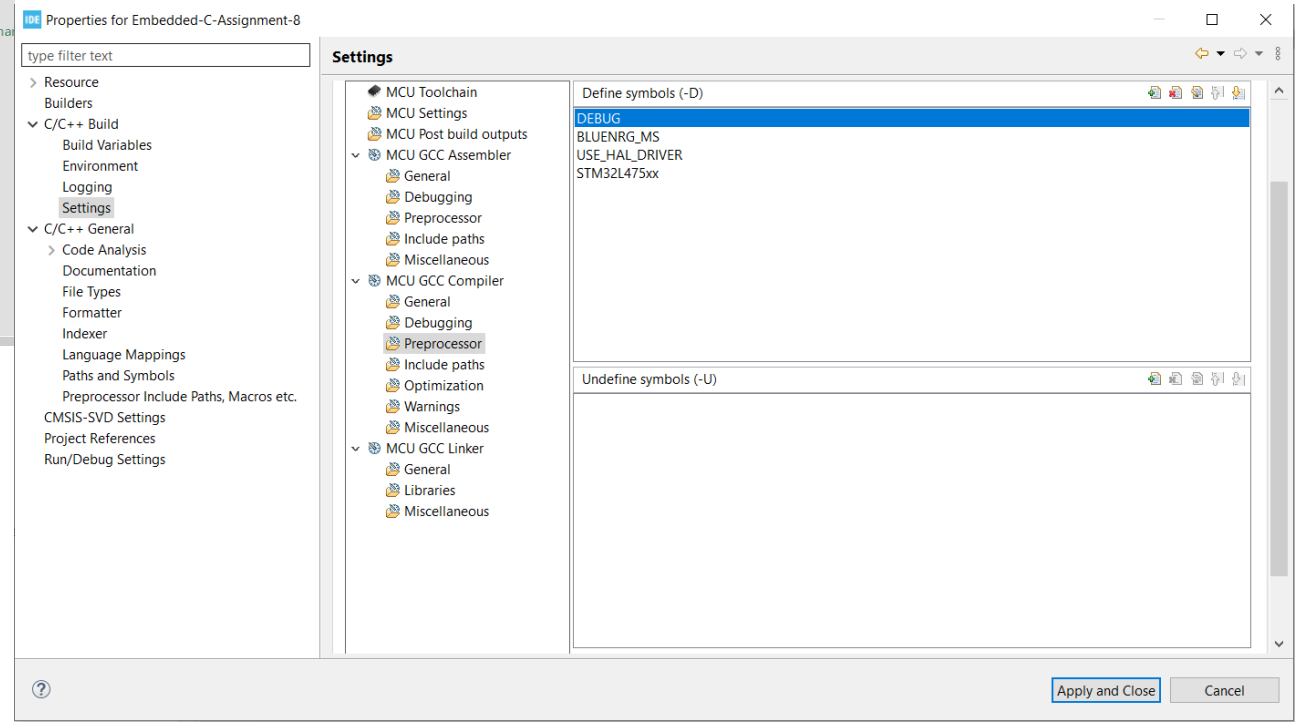






Step 12. Add BLUENRG_MS into preprocessor symbols, since our aci_gap_init function use 5 parameters





Step 13. Change the hal_types.h and hal.h include header files to bluenrg_private_hal_types.h and stm32l4xx_hal.h file in hci_ie.c and bluenrg_l2cap_aci.c file

Project Explorer

- Embedded-C-Assignment-8
  - Binaries
  - Includes
  - BLE
    - Common
      - ble_core
        - ble_lib.h
        - ble_status.h
        - bluenrg_aci_const.h
        - bluenrg_gap_aci.c
        - bluenrg_gap_aci.h
        - bluenrg_gap.h
        - bluenrg_gatt_aci.c
        - bluenrg_gatt_aci.h
        - bluenrg_gatt_server.h
        - bluenrg_hal_aci.c
        - bluenrg_hal_aci.h
        - bluenrg_l2cap_aci.c
        - bluenrg_l2cap_aci.h
        - bluenrg_private_hal_types.h
        - compiler.h
        - hci_const.h
        - hci_le.c
        - hci_le.h
        - hci_tl_io.h
        - link_layer.h
        - osal.c
        - osal.h
        - uuid.h
      - ble_services
      - debug

hw.h | svc_ctl.c | bluenrg_gap... | hci_le.c | main.c | hr

```c
 * All rights reserved.
 *
 * This software is licensed under terms that can be fou
 * in the root directory of this software component.
 * If no LICENSE file comes with this software, it is pr
 *
 ******************************************************
 */

//#include "hal_types.h"
#include "bluenrg_private_hal_types.h"
#include "osal.h"
#include "ble_status.h"
//#include "hal.h"
#include "stm32l4xx_hal.h"
#include "hci_const.h"
#if (STM == 1)
#include "gp_timer.h"
#endif
#include "hci_tl_io.h"

#define MIN(a,b)            ((a) < (b) )? (a) : (b)
#define MAX(a,b)            ((a) > (b) )? (a) : (b)

int hci_reset()
{
  struct hci_request rq;
  uint8_t status;

  Osal_MemSet(&rq, 0, sizeof(rq));
  rq.ogf = OGF_HOST_CTL;
  rq.ocf = OCF_RESET;
  rq.rparam = &status;
  rq.rlen = 1;

  if (hci_send_req(&rq, FALSE) < 0)
```

Project Explorer

- Embedded-C-Assignment-8
  - Binaries
  - Includes
  - BLE
    - Common
      - ble_core
        - ble_lib.h
        - ble_status.h
        - bluenrg_aci_const.h
        - bluenrg_gap_aci.c
        - bluenrg_gap_aci.h
        - bluenrg_gap.h
        - bluenrg_gatt_aci.c
        - bluenrg_gatt_aci.h
        - bluenrg_gatt_server.h
        - bluenrg_hal_aci.c
        - bluenrg_hal_aci.h
        - bluenrg_l2cap_aci.c
        - bluenrg_l2cap_aci.h
        - bluenrg_private_hal_types.h
        - compiler.h
        - hci_const.h
        - hci_le.c
        - hci_le.h
        - hci_tl_io.h
        - link_layer.h
        - osal.c
        - osal.h
        - uuid.h

hw.h | svc_ctl.c | bluenrg_gap... | hci_le.c | bluenrg_l2ca... | main.c | hw_lpm.c | startup_stm... | bluenrg_

```c
/******************* (C) COPYRIGHT 2013 STMicroelectronics *******************
* File Name          : bluenrg_hci.c
* Author             : AMS - HEA&RF BU
* Version            : V1.0.0
* Date               : 4-Oct-2013
* Description        : File with HCI commands for BlueNRG FW6.0 and above.
*******************************************************************************
* This software is licensed under terms that can be found in the LICENSE file
* in the root directory of this software component.
* If no LICENSE file comes with this software, it is provided AS-IS.
*******************************************************************************/

//#include "hal_types.h"
#include "bluenrg_private_hal_types.h"
#include "osal.h"
#include "ble_status.h"
//#include "hal.h"
#include "stm32l4xx_hal.h"
#include "osal.h"
#include "hci_const.h"
#include "bluenrg_aci_const.h"
#include "bluenrg_hal_aci.h"
#include "bluenrg_gap.h"

#define MIN(a,b)            ((a) < (b) )? (a) : (b)
#define MAX(a,b)            ((a) > (b) )? (a) : (b)

tBleStatus aci_l2cap_connection_parameter_update_request(uint16_t conn_handle, uint16_t interval_min,
                                                         uint16_t interval_max, uint16_t slave_latency,
                                                         uint16_t timeout_multiplier)
{
  struct hci_request rq;
  uint8_t status;
```

Step 14. Add hci_tl_io.h include file into hci_le.c file

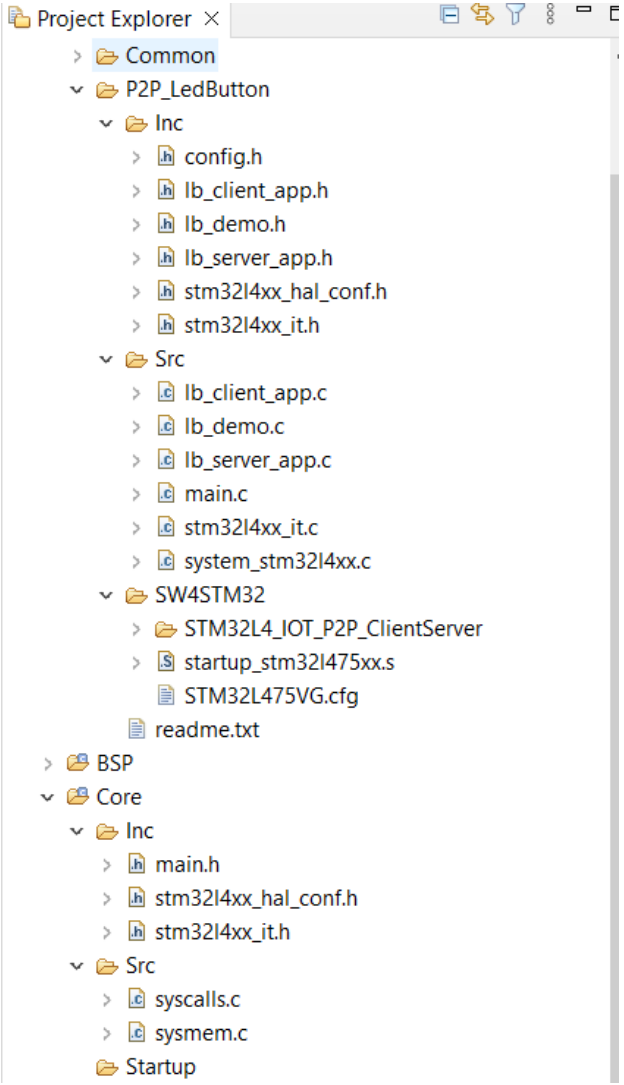
Project Explorer

- Embedded-C-Assignment-8
  - Binaries
  - Includes
  - BLE
    - Common
      - ble_core
        - ble_lib.h
        - ble_status.h
        - bluenrg_aci_const.h
        - bluenrg_gap_aci.c
        - bluenrg_gap_aci.h
        - bluenrg_gap.h
        - bluenrg_gatt_aci.c
        - bluenrg_gatt_aci.h
        - bluenrg_gatt_server.h
        - bluenrg_hal_aci.c
        - bluenrg_hal_aci.h
        - bluenrg_l2cap_aci.c
        - bluenrg_l2cap_aci.h
        - bluenrg_private_hal_types.h
        - compiler.h
        - hci_const.h
        - hci_le.c
        - hci_le.h
        - hci_tl_io.h
        - link_layer.h
        - osal.c
        - osal.h
        - uuid.h
      - ble_services
      - debug

hw.h | svc_ctl.c | bluenrg_gap... | hci_le.c | bluenrg_l2ca... | main.c

```c
//#include "hal_types.h"
#include "bluenrg_private_hal_types.h"
#include "osal.h"
#include "ble_status.h"
//#include "hal.h"
#include "stm32l4xx_hal.h"
#include "hci_const.h"
#if (STM == 1)
#include "gp_timer.h"
#endif
#include "hci_tl_io.h"

#define MIN(a,b)            ((a) < (b) )? (a) : (b)
#define MAX(a,b)            ((a) > (b) )? (a) : (b)

int hci_reset()
{
  struct hci_request rq;
  uint8_t status;

  Osal_MemSet(&rq, 0, sizeof(rq));
  rq.ogf = OGF_HOST_CTL;
  rq.ocf = OCF_RESET;
  rq.rparam = &status;
  rq.rlen = 1;

  if (hci_send_req(&rq, FALSE) < 0)
    return BLE_STATUS_TIMEOUT;

  return status;
}

int hci_disconnect(uint16_t handle, uint8_t reason)
{
  struct hci_request rq;
```

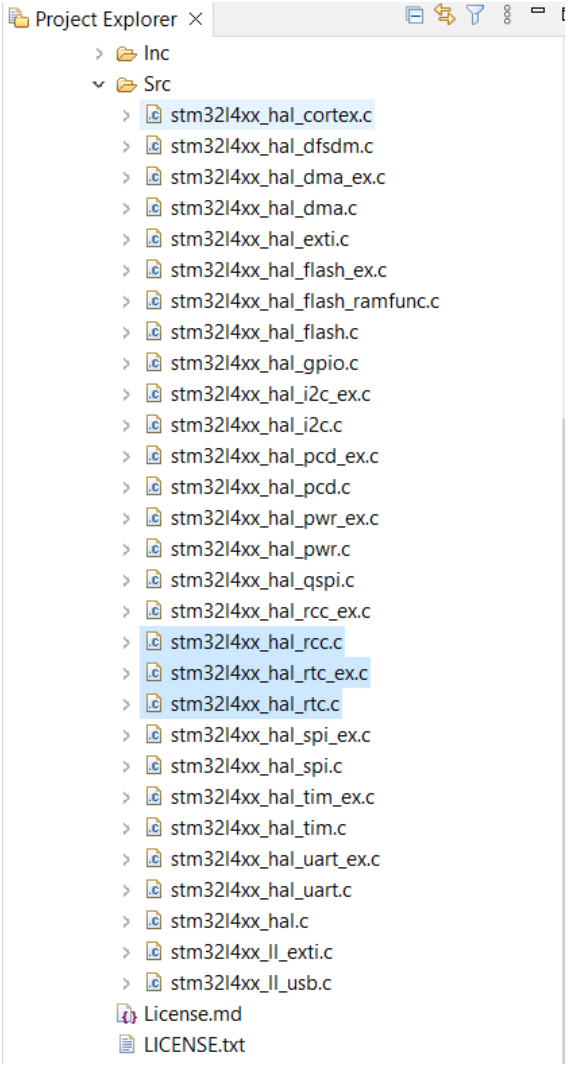
Step 15. Delete main.c, stm32l4xx_it.c, system_stm32l4xx.c, stm32l4xx_hal_msp.cand also startup file in Core folder

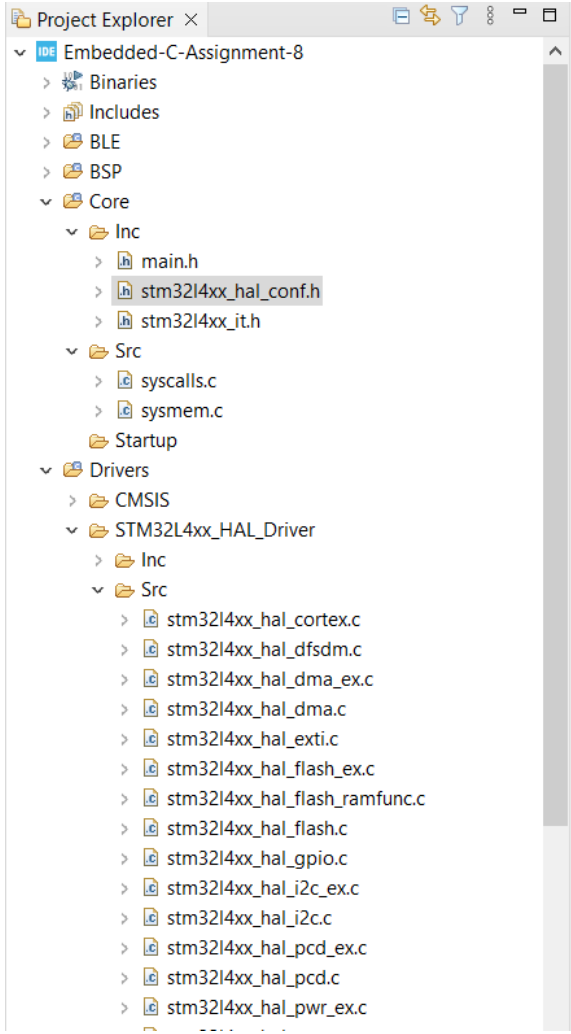


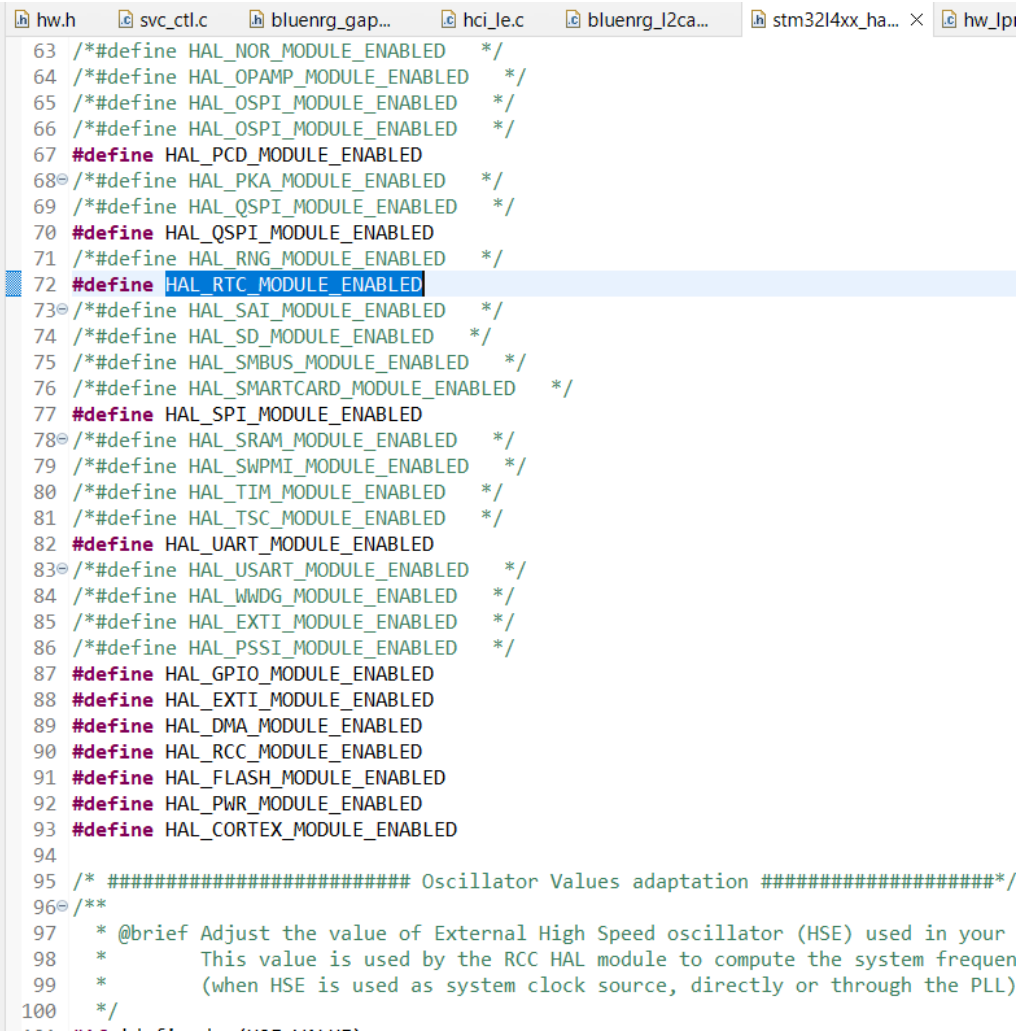
Step 16. Delete ble_hci_ti_io_template.c file to solve multiple definition of hci_send_rq function

ble_core	2/27/2023 10:25 PM
ble_services	2/27/2023 10:25 PM
debug	2/27/2023 10:25 PM
hw	2/27/2023 10:25 PM
tl	2/27/2023 10:25 PM
utilities	2/27/2023 10:25 PM
ble_config_template	2/27/2023 10:25 PM
ble_hci_tl_io_template	2/27/2023 10:25 PM
common	2/27/2023 10:25 PM
config_template	2/27/2023 10:25 PM

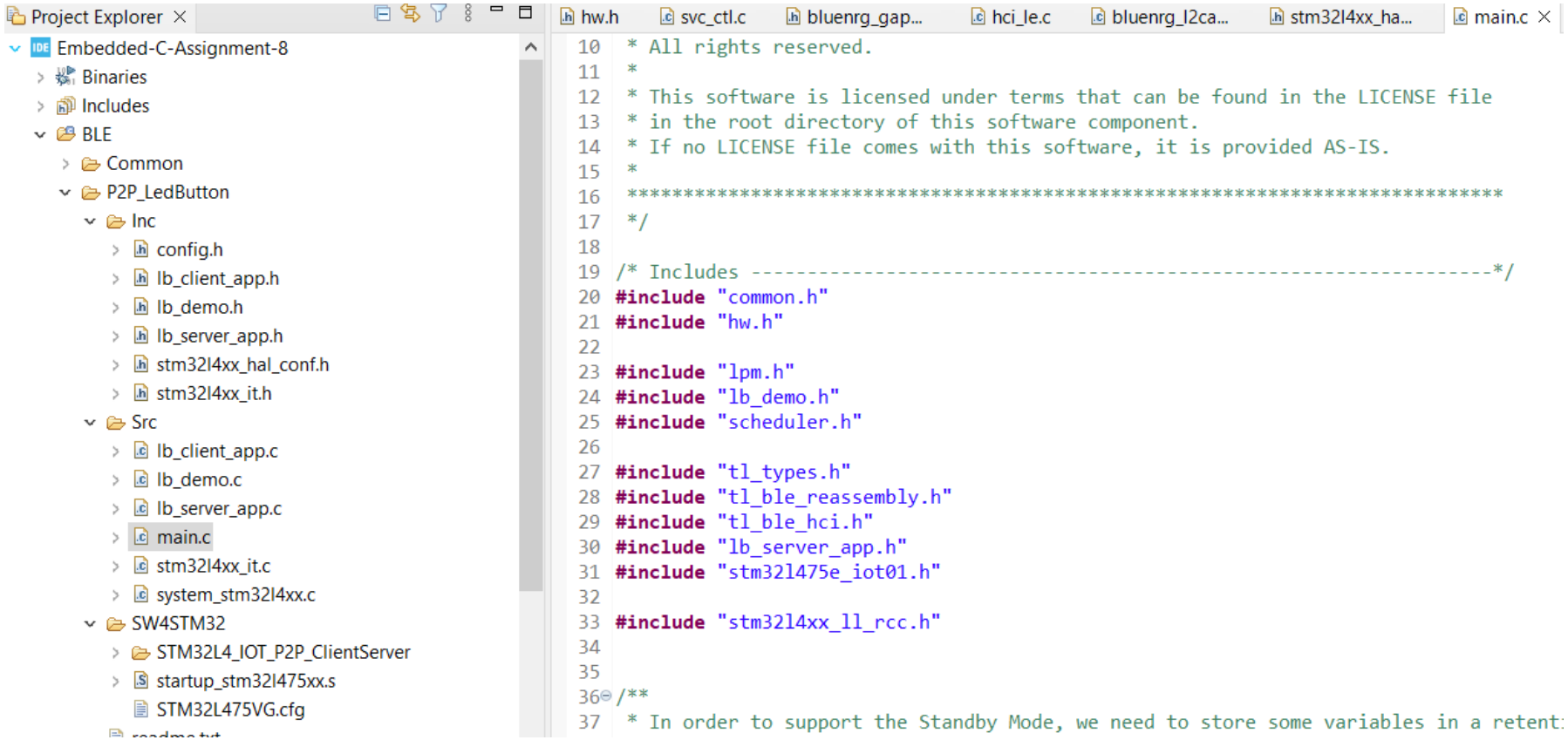
Step 17. Add stm32l4xx_hal_rcc, rtc and rtc_ex source files in driver folder, and uncomment HAL_RTC_MODULE_ENABLED to use the feature



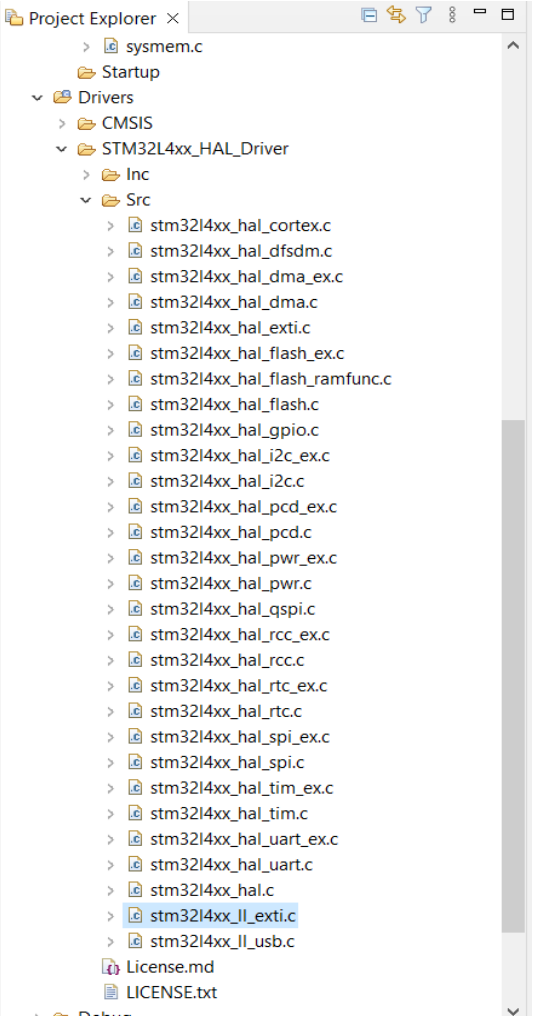


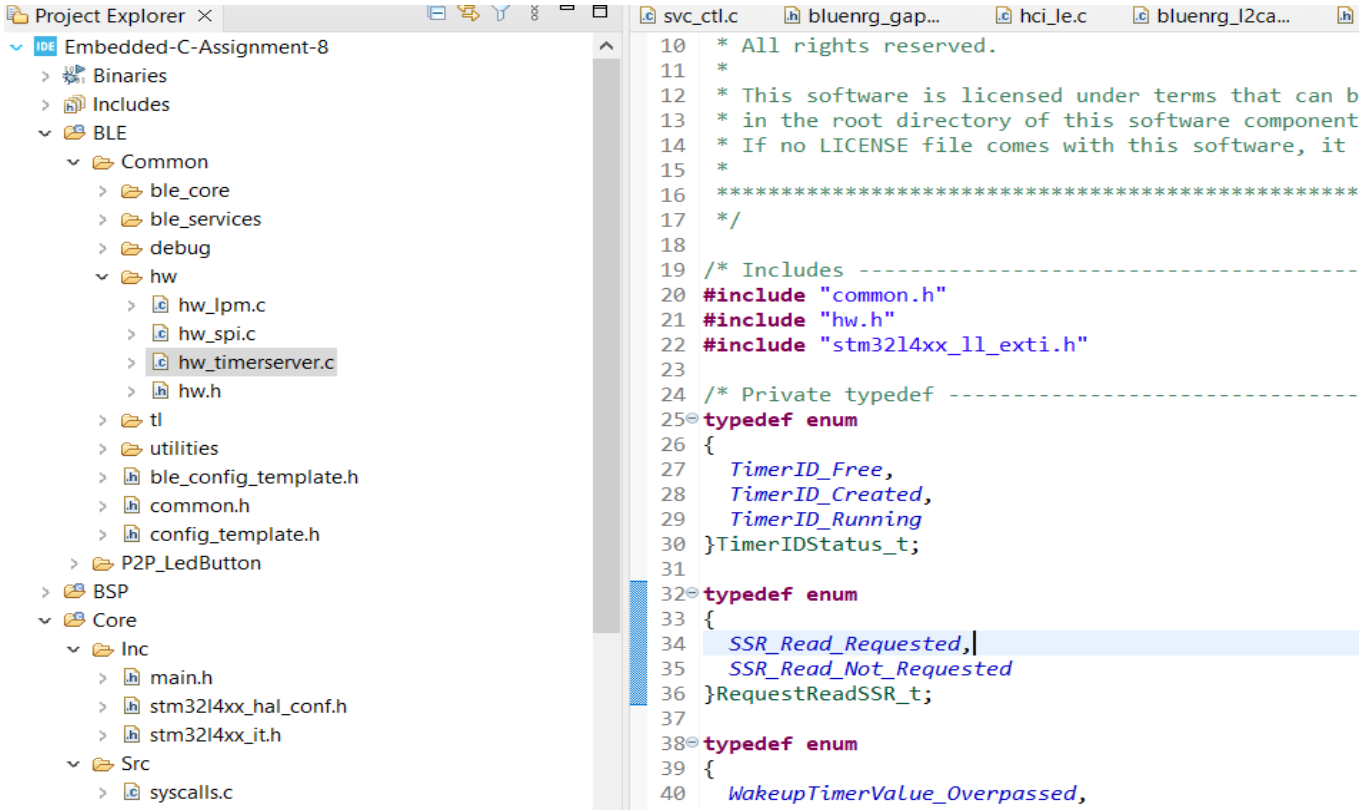


Step 18. Add stm32l4xx_ll_rcc.h file in main.c file under BLE folder to resolve LL_RCC errors

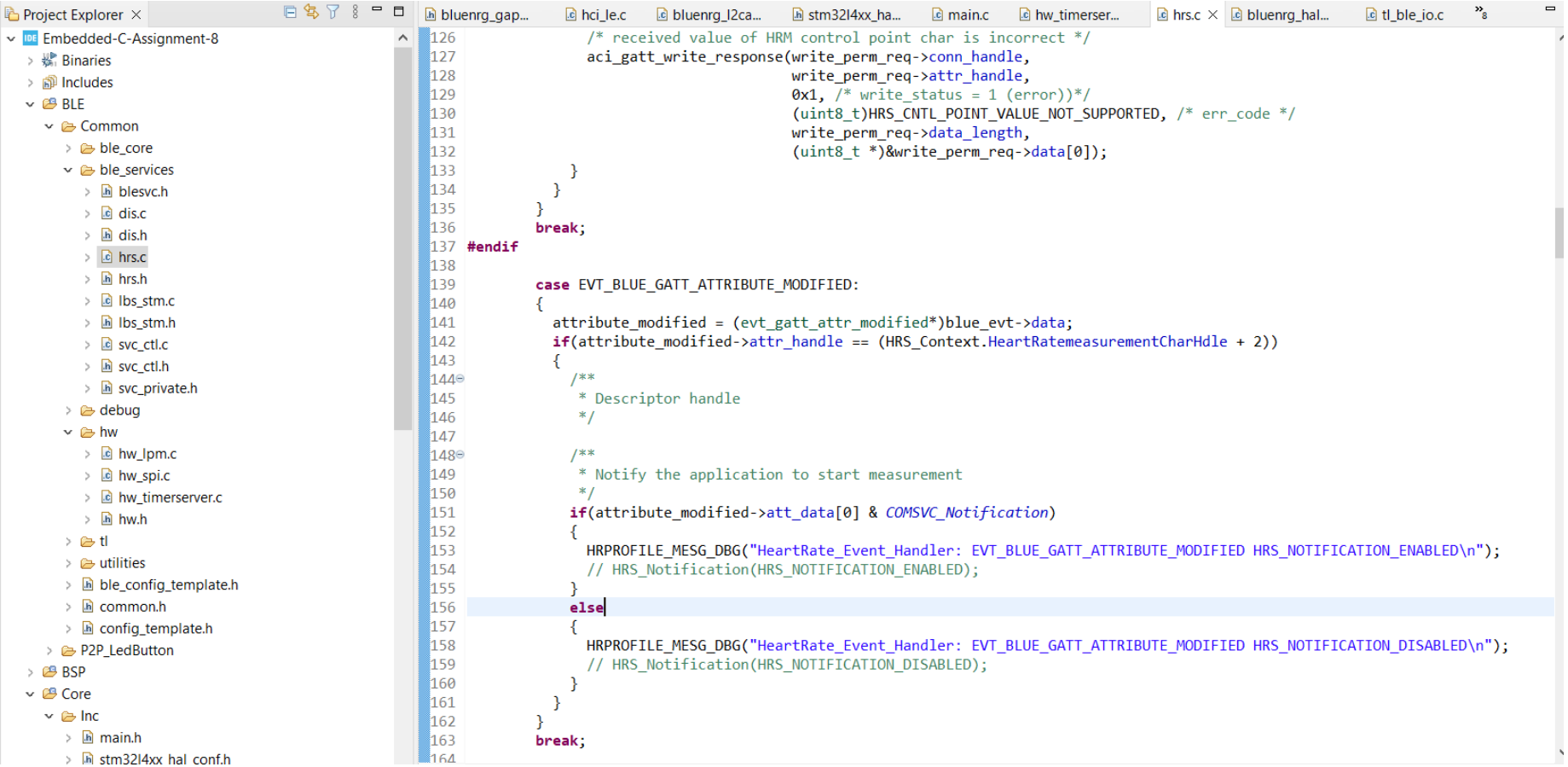


Step 19. Add stm32l4xx_ll_exit.h and .c file into driver inc and src folder, then add the header file into hw_timerserver.c file





Step 20. Comment out all the HRS_Notification function in hrs.c file



Step 21. Finally, we can build the project successfully, and when run in debug mode, we can search the Bluetooth module of our stm board

Project Explorer

- Embedded-C-Assignment-8
  - Binaries
  - Includes
  - BLE
  - BSP
  - Core
  - Drivers
  - Debug
  - Embedded-C-Assignment-8.ioc
  - Embedded-C-Assignment-8 Debug.launch
  - STM32L475VGTX_FLASH.ld
  - STM32L475VGTX_RAM.ld

bluenrg_gap... | hci_le.c | bluenrg_l2ca... | stm32l4xx_ha... | main.c | hw_timerser... | hrs.c | bluenrg_hal... | tl_ble_io.c

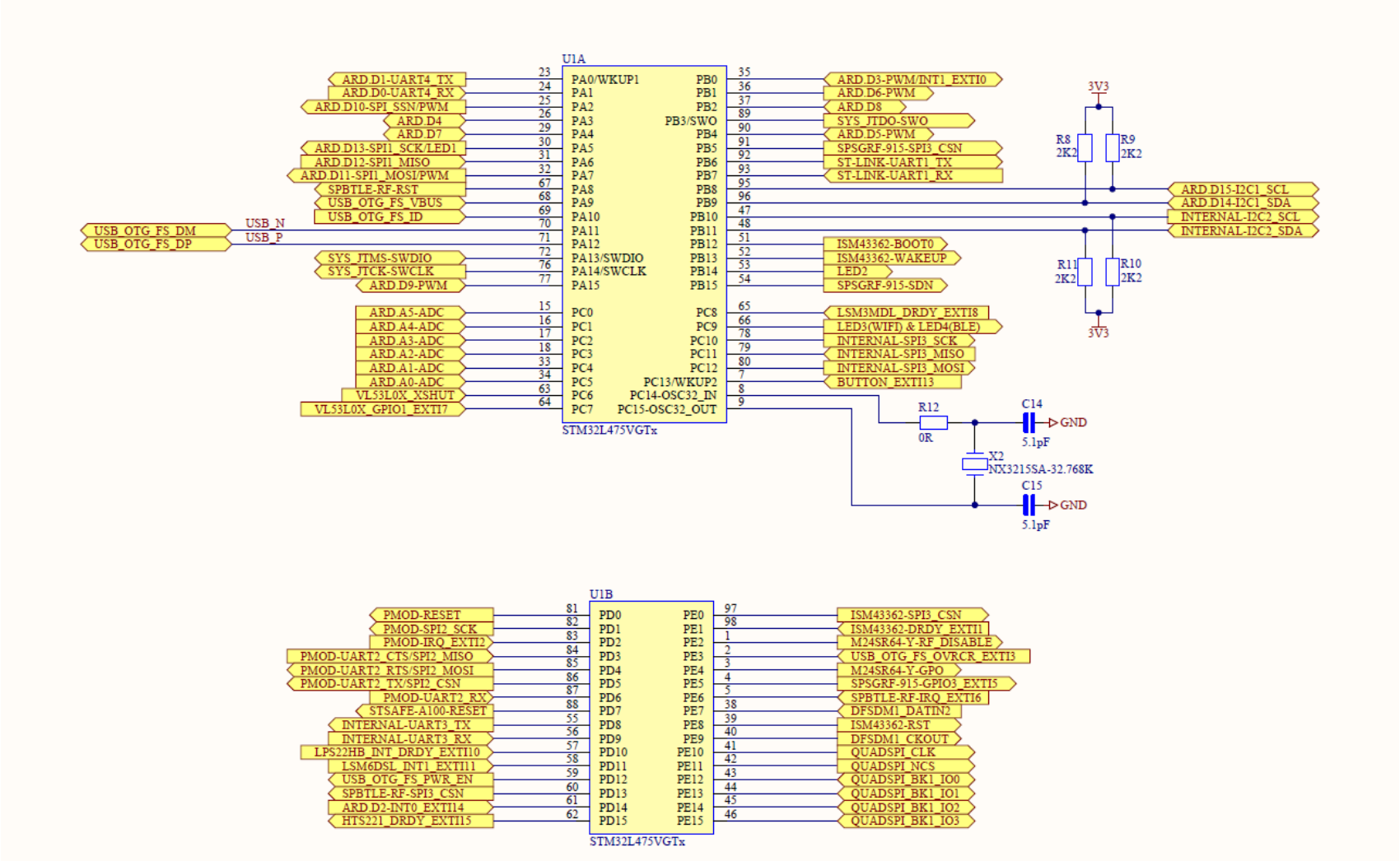
```c
            /* received value of HRM control point char is incorrect */
            aci_gatt_write_response(write_perm_req->conn_handle,
                                    write_perm_req->attr_handle,
                                    0x1, /* write_status = 1 (error))*/
                                    (uint8_t)HRS_CNTL_POINT_VALUE_NOT_SUPPORTED, /* err_code */
                                    write_perm_req->data_length,
                                    (uint8_t *)&write_perm_req->data[0]);
          }
        }
      }
      break;
#endif

      case EVT_BLUE_GATT_ATTRIBUTE_MODIFIED:
      {
        attribute_modified = (evt_gatt_attr_modified*)blue_evt->data;
        if(attribute_modified->attr_handle == (HRS_Context.HeartRatemeasurementCharHdle + 2))
        {
          /**
           * Descriptor handle
```

Problems | Tasks | Console | Properties

CDT Build Console [Embedded-C-Assignment-8]

```
21:58:11 **** Incremental Build of configuration Debug for project Embedded-C-Assignment-8 ****
make -j8 all
arm-none-eabi-size   Embedded-C-Assignment-8.elf
   text    data     bss     dec     hex filename
  25616      32    4304   29952    7500 Embedded-C-Assignment-8.elf
Finished building: default.size.stdout


21:58:12 Build Finished. 0 errors, 0 warnings. (took 1s.355ms)
```

Appendix, schematic for BLE module, processor side connection



Appendix, schematic for BLE module