

Embedded Controller
Programming with Embedded
C

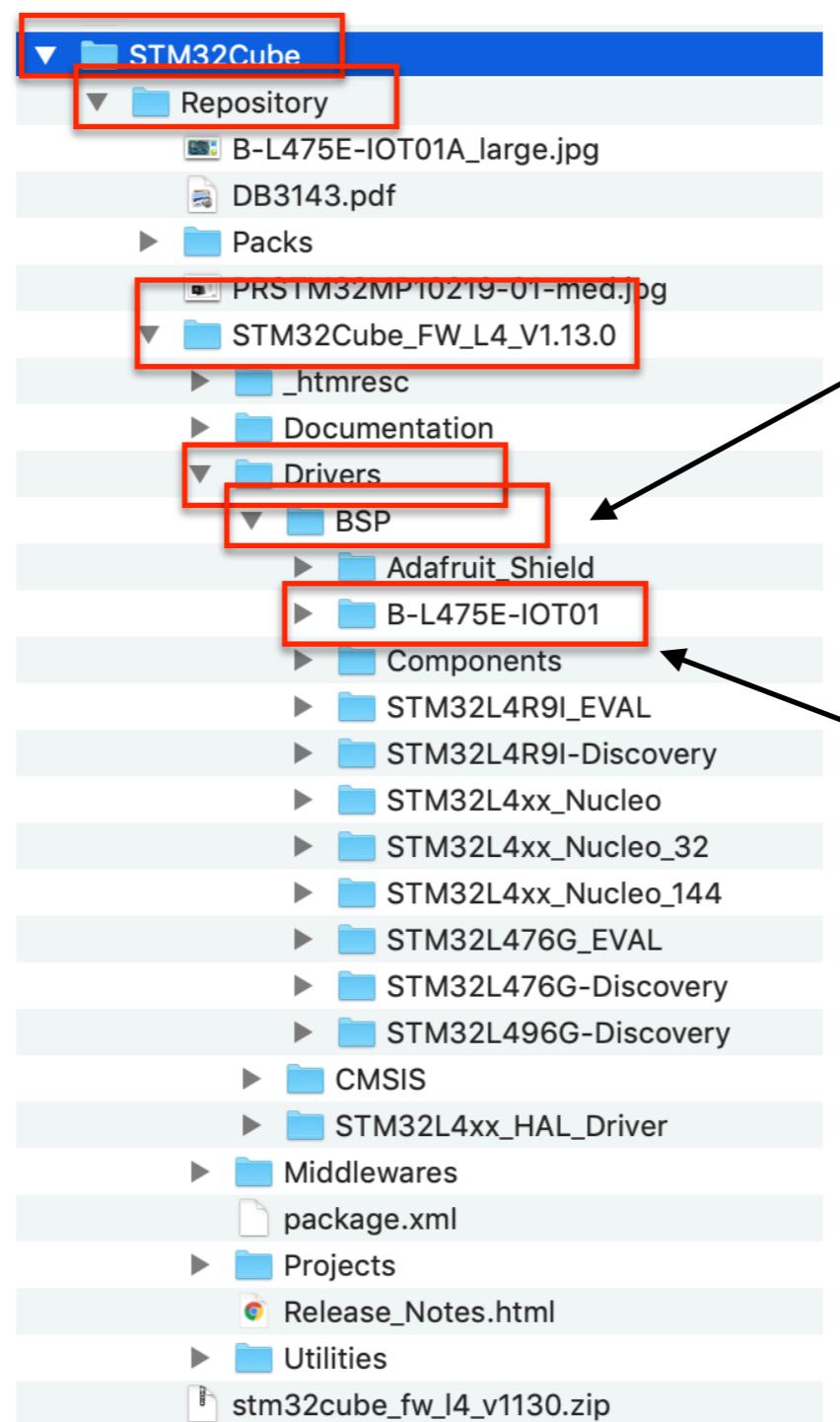
Lesson 4 - BSP and Embedded C

Norman McEntire
norman.mcentre@gmail.com

Contents

- Introduction to BSP and Embedded C
- STM32CubeMX and BSP Code Generation
- Tour of BSP
- TrueStudio and BSP
 - Blinking LED

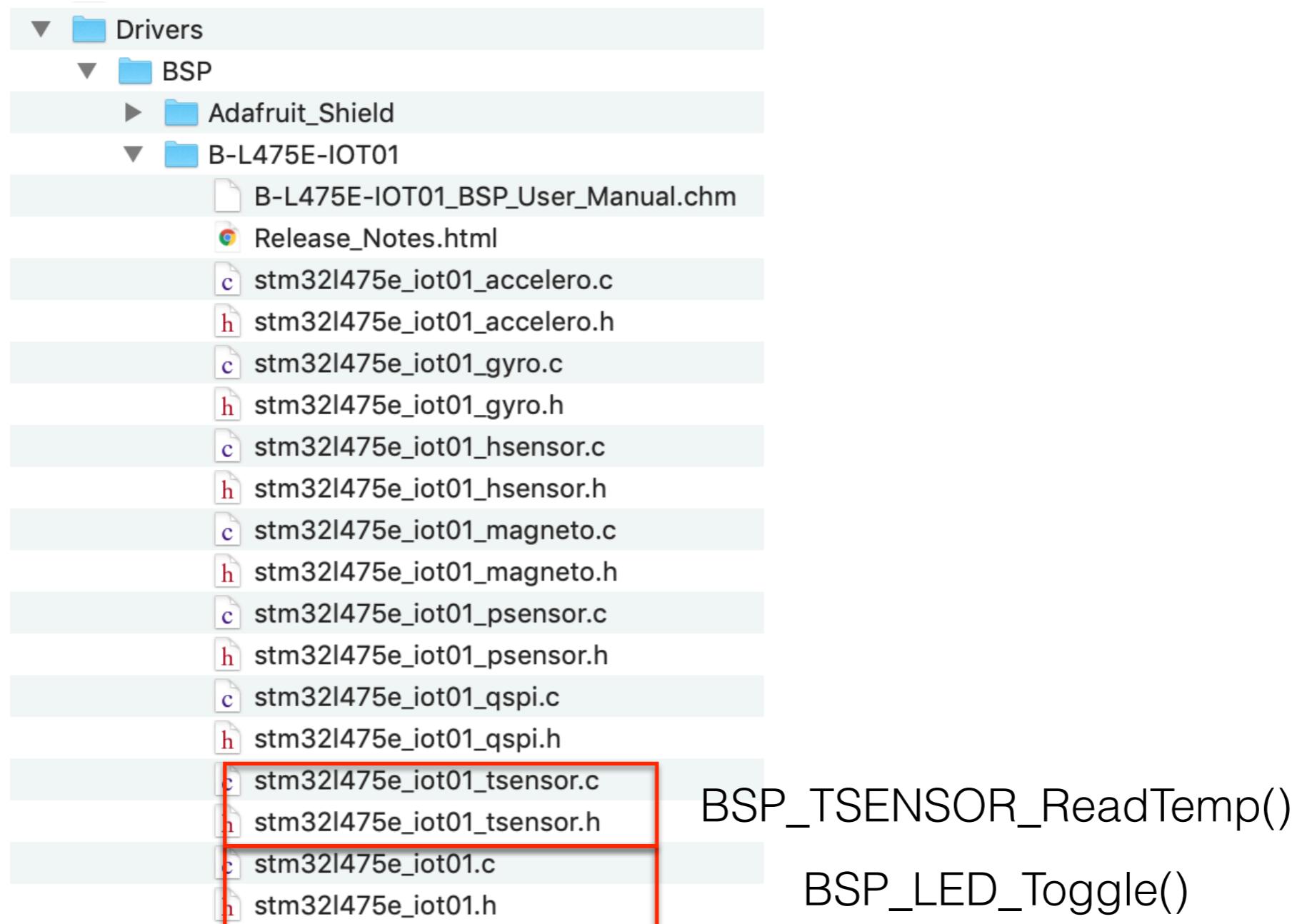
Where to find BSP - Part 1



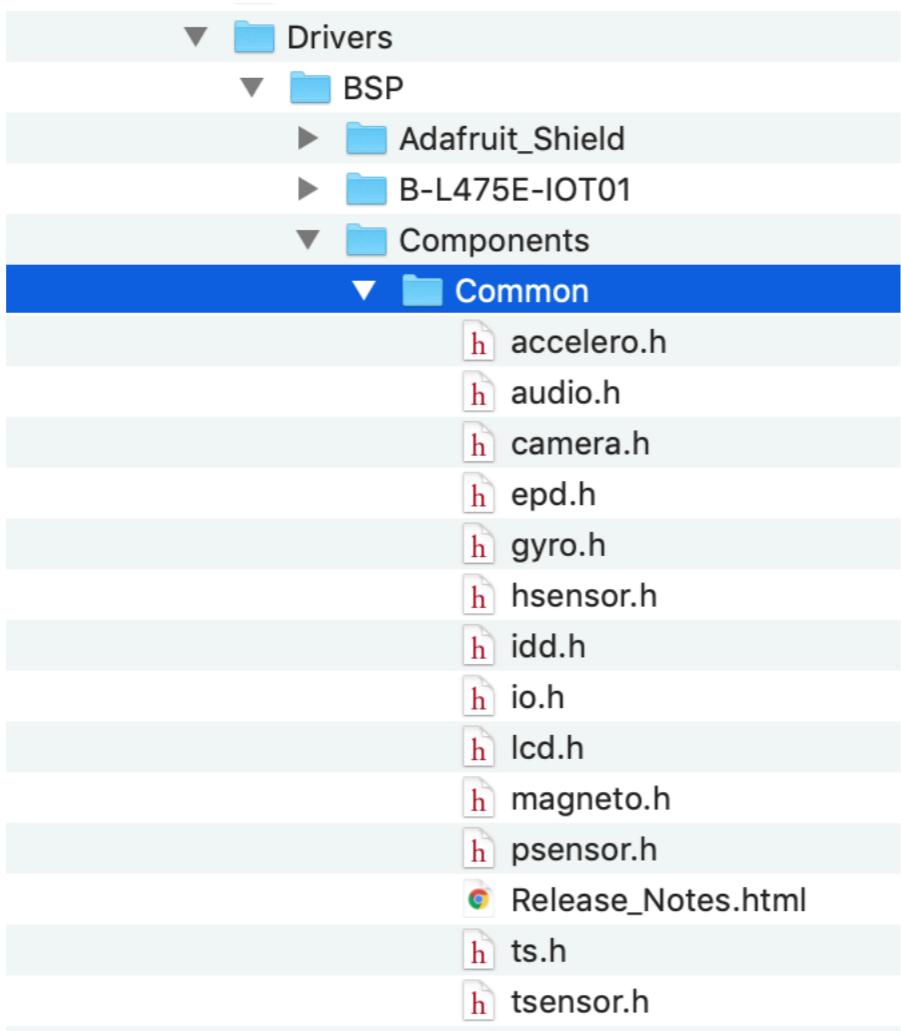
BSP =
Board
Support
Package

Board we
are using

Where to find BSP - Part 2

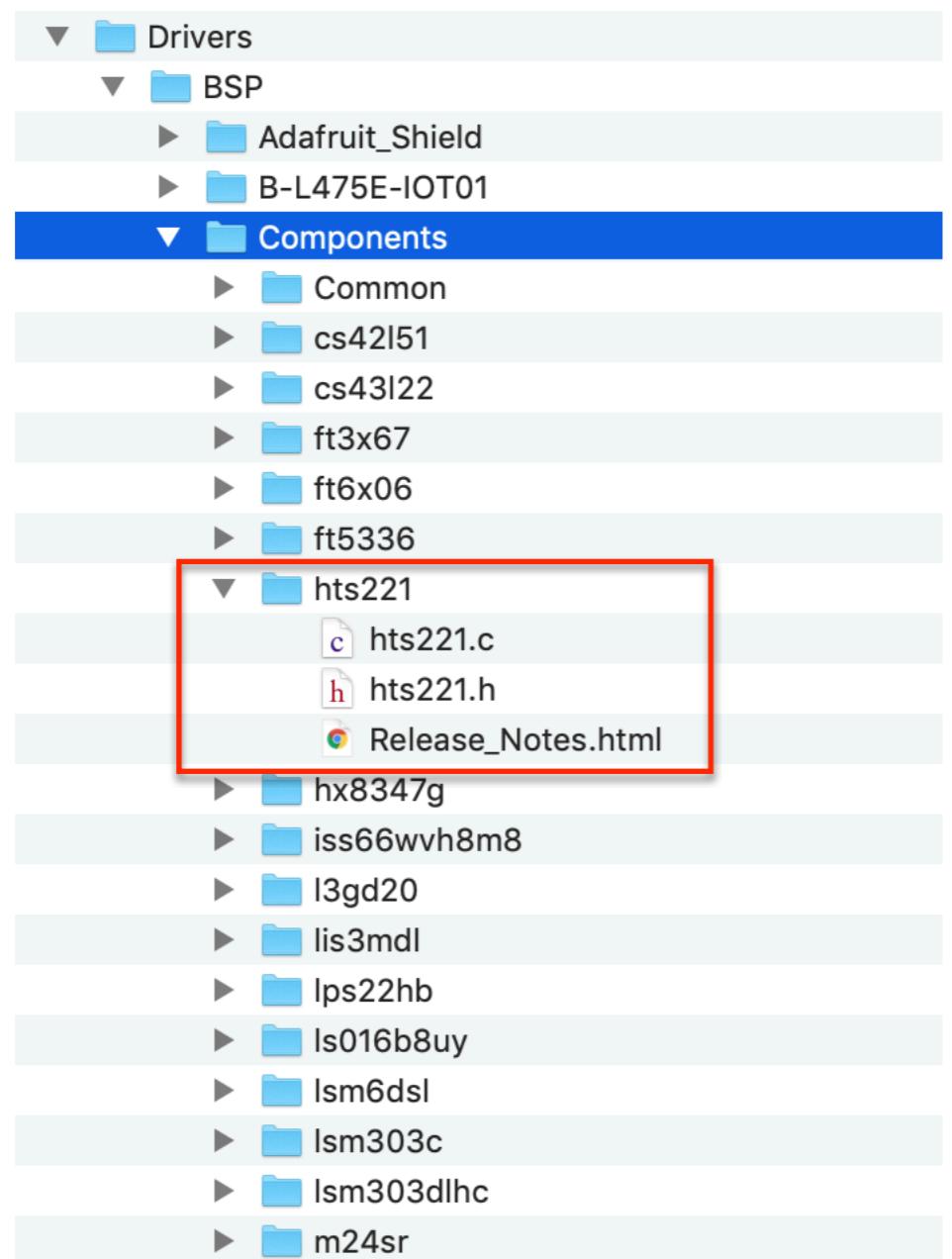


Where to find BSP - Part 3



Key Point:
Observe
Components/
Common -
these are the
common I/O
sensors that
are not
hardware specific,
e.g. tsensor.h

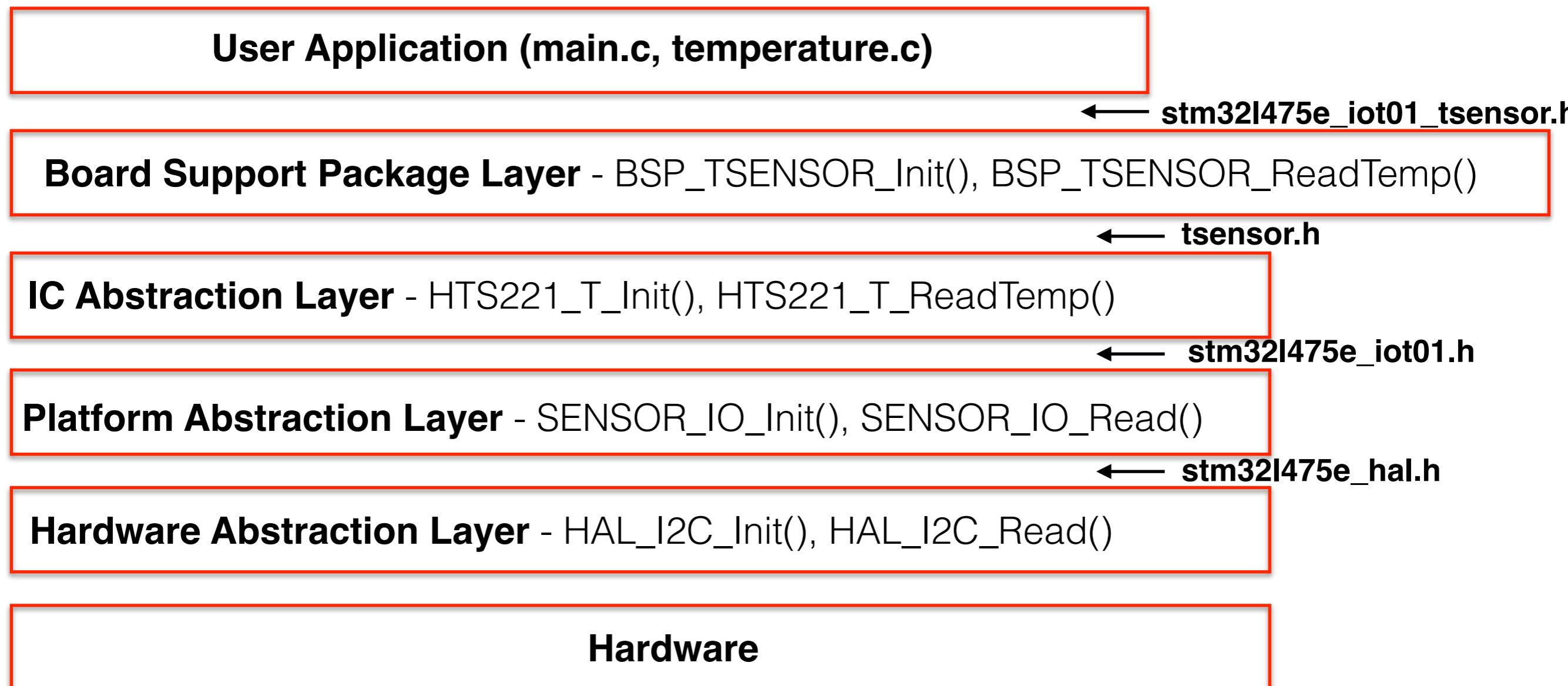
Where to find BSP - Part 4



Key Point:
Observe
Components -
these are the
actual hardware
components, e.g.
hts221

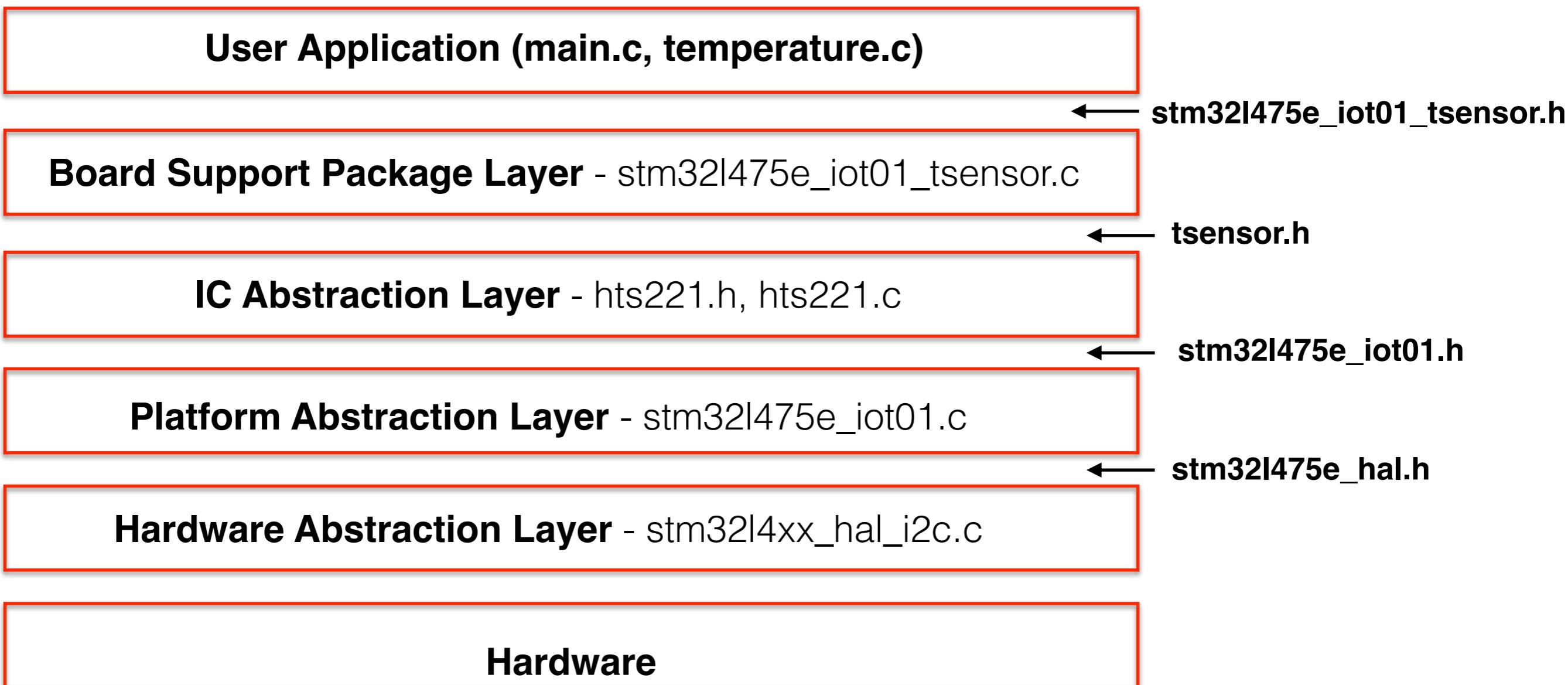
BSP Block Diagram

View 1: Function Names



BSP Block Diagram

View 2: Filenames



Trace of Code - From BSP to HAL

Trace of Code From BSP to HAL

- `BSP_TENSOR_Init()`
- `HT221_T_Init()`
- `SENSOR_IO_Read()`
- `HAL_I2C_Read()`

BSP_TSENSOR_Init()

Drivers/BSP/B-L475E-IOT01/tsensor.c

```
73  /**
74   * @brief  Initializes peripherals used by the I2C Temperature Sensor driver.
75   * @retval TSFNSOR status
76   */
77 uint32_t BSP_TSENSOR_Init(void)
78 {
79     uint8_t ret = TSENSOR_ERROR;
80
81 #ifdef USE_LPS22HB_TEMP
82     tsensor_drv = &LPS22HB_T_Drv;
83 #else /* USE_HTS221_TEMP */
84     tsensor_drv = &HTS221_T_Drv;
85 #endif
86
87     /* Low level init */
88     SENSOR_IO_Init();
89
90     /* TSENSOR Init */
91     tsensor_drv->Init(TSENSOR_I2C_ADDRESS, NULL);
92
93     ret = TSENSOR_OK;
94
95     return ret;
96 }
```

Defined in BSP/Components/hts221/
hts221.h

Notice address added

TSENSOR_DrvTypeDef

Drivers/BSP/Components/Common/ tsensor.h

```
82 |  /*
83 typedef struct
84 {
85     void        (*Init)(uint16_t, TSENSOR_InitTypeDef *);
86     uint8_t    (*IsReady)(uint16_t, uint32_t);
87     uint8_t    (*ReadStatus)(uint16_t);
88     float      (*ReadTemp)(uint16_t);
89 }TSENSOR_DrvTypeDef;
```

HTS221_T_Drv

Drivers/BSP/Components/hts221/hts221.c

```
65  /* HTS221_Temperature_Private_Variables */  
66  TSENSOR_DrvTypeDef HTS221_T_Drv =  
67  {  
68      HTS221_T_Init,  
69      0,  
70      0,  
71      HTS221_T_ReadTemp  
72  };
```

HTS221_T_Init()

Drivers/BSP/Components/hts221/hts221.c

```
169  /**
170   * @brief Set HTS221 temperature sensor Initialization.
171   * @param DeviceAddr: I2C device address
172   * @param InitStruct: pointer to a TSENSOR_InitTypeDef structure
173   *                     that contains the configuration setting for the HTS221.
174 */
175 void HTS221_T_Init(uint16_t DeviceAddr, TSENSOR_InitTypeDef *pInitStruct)
176 {
177     uint8_t tmp;
178
179     /* Read CTRL_REG1 */
180     tmp = SENSOR_I0_Read(DeviceAddr, HTS221_CTRL_REG1);
181
182     /* Enable BDU */
183     tmp &= ~HTS221_BDU_MASK;
184     tmp |= (1 << HTS221_BDU_BIT);
185 }
```

SENSOR_IO_Read()

Drivers/BSP/B-L475E-IOT01/

stm32l475e_iot01.c

```
591  /**
592   * @brief Reads a single data.
593   * @param Addr: I2C address
594   * @param Reg: Reg address
595   * @retval Data to be read
596 */
597 uint8_t SENSOR_IO_Read(uint8_t Addr, uint8_t Reg)
598 {
599     uint8_t read_value = 0;
600
601     I2Cx_ReadMultiple(&hI2cHandler, Addr, Reg, I2C_MEMADD_SIZE_8BIT, (uint8_t*)&read_value, 1);
602
603     return read_value;
604 }
```

I2Cx_ReadMultiple()

Drivers/BSP/B-L475E-IOT01/

stm32l475e_iot01.c

```
473 /**
474  * @brief Reads multiple data.
475  * @param i2c_handler : I2C handler
476  * @param Addr: I2C address
477  * @param Reg: Reg address
478  * @param MemAddress: memory address
479  * @param Buffer: Pointer to data buffer
480  * @param Length: Length of the data
481  * @retval HAL status
482 */
483 static HAL_StatusTypeDef I2Cx_ReadMultiple(I2C_HandleTypeDef *i2c_handler, uint8_t Addr,
484                                         uint16_t Reg, uint16_t MemAddress, uint8_t *Buffer, uint16_t Length)
485 {
486     HAL_StatusTypeDef status = HAL_OK;
487
488     status = HAL_I2C_Mem_Read(i2c_handler, Addr, (uint16_t)Reg, MemAddress, Buffer, Length, 1000);
489
490     /* Check the communication status */
491     if(status != HAL_OK)
492     {
493         /* I2C error occurred */
494         I2Cx_Error(i2c_handler, Addr);
495     }
496     return status;
497 }
```

Summary - Trace of Code From BSP to HAL

- `BSP_TENSOR_Init()`
- `HT221_T_Init()`
- `SENSOR_IO_Read()`
- `HAL_I2C_Read()`

Drivers/BSP/B-L475E-
IOT01/
stm32l475e_iot01.h

stm32l475e_iot01.h - 1

```
1  /**
2   ****
3   * @file    stm32l475e_iot01.h
4   * @author  MCD Application Team
5   * @brief   STM32L475E IOT01 board support package
6   ****
```

```
46  /* Define to prevent recursive inclusion -
47  #ifndef __STM32L475E_IOT01_H
48  #define __STM32L475E_IOT01_H
```

```
49
50  #ifdef __cplusplus
51  | extern "C" {
52  #endif
```

```
53
```

```
54  /* Includes -----
55  #include "stm32l4xx_hal.h"
56
```

stm32l475e_iot01.h - 2

```
71 | //  
72 |     typedef enum  
73 |     {  
74 |         LED2 = 0,  
75 |         LED_GREEN = LED2,  
76 |     }Led_TypeDef;  
77  
78  
79 |     typedef enum  
80 |     {  
81 |         BUTTON_USER = 0  
82 |     }Button_TypeDef;  
83  
84 |     typedef enum  
85 |     {  
86 |         BUTTON_MODE_GPIO = 0,  
87 |         BUTTON_MODE_EXTI = 1  
88 |     }ButtonMode_TypeDef;  
89
```

stm32l475e_iot01.h - 3

```
-->
103  /**
104   * @brief Define for STM32L475E_IOT01 board
105  */
106 #if !defined (USE_STM32L475E_IOT01)
107 #define USE_STM32L475E_IOT01
108 #endif
109
110 #define LEDn          ((uint8_t)1)
111
112 #define LED2_PIN        GPIO_PIN_14
113 #define LED2_GPIO_PORT    GPIOB
114 #define LED2_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB_CLK_ENABLE()
115 #define LED2_GPIO_CLK_DISABLE() __HAL_RCC_GPIOB_CLK_DISABLE()
116
```

stm32l475e_iot01.h - 4

```
126  /**
127   * @brief Wakeup push-button
128  */
129 #define USER_BUTTON_PIN          GPIO_PIN_13
130 #define USER_BUTTON_GPIO_PORT    GPIOC
131 #define USER_BUTTON_GPIO_CLK_ENABLE() __HAL_RCC_GPIOC_CLK_ENABLE()
132 #define USER_BUTTON_GPIO_CLK_DISABLE() __HAL_RCC_GPIOC_CLK_DISABLE()
133 #define USER_BUTTON_EXTI_IRQn    EXTI15_10_IRQHandler
134
```

stm32l475e_iot01.h - 5

```
149  * @brief Definition for COM port1, connected to USART1
150  */
151 #define DISCOVERY_COM1                                USART1
152 #define DISCOVERY_COM1_CLK_ENABLE()                   __HAL_RCC_USART1_CLK_ENABLE()
153 #define DISCOVERY_COM1_CLK_DISABLE()                 __HAL_RCC_USART1_CLK_DISABLE()
154
155 #define DISCOVERY_COM1_TX_PIN                         GPIO_PIN_6
156 #define DISCOVERY_COM1_TX_GPIO_PORT                  GPIOB
157 #define DISCOVERY_COM1_TX_GPIO_CLK_ENABLE()           __HAL_RCC_GPIOB_CLK_ENABLE()
158 #define DISCOVERY_COM1_TX_GPIO_CLK_DISABLE()          __HAL_RCC_GPIOB_CLK_DISABLE()
159 #define DISCOVERY_COM1_TX_AF                          GPIO_AF7_USART1
160
161 #define DISCOVERY_COM1_RX_PIN                         GPIO_PIN_7
162 #define DISCOVERY_COM1_RX_GPIO_PORT                  GPIOB
163 #define DISCOVERY_COM1_RX_GPIO_CLK_ENABLE()           __HAL_RCC_GPIOB_CLK_ENABLE()
164 #define DISCOVERY_COM1_RX_GPIO_CLK_DISABLE()          __HAL_RCC_GPIOB_CLK_DISABLE()
165 #define DISCOVERY_COM1_RX_AF                          GPIO_AF7_USART1
166
167 #define DISCOVERY_COM1_IRQn                        USART1_IRQn
168
```

stm32l475e_iot01.h - 6

```
215  /* I2C Sensors address */
216  /* LPS22HB (Pressure) I2C Address */
217  #define LPS22HB_I2C_ADDRESS  (uint8_t)0xBA
218  /* HTS221 (Humidity) I2C Address */
219  #define HTS221_I2C_ADDRESS   (uint8_t)0xBE
220
221  #ifdef USE_LPS22HB_TEMP
222  /* LPS22HB Sensor hardware I2C address */
223  #define TSENSOR_I2C_ADDRESS    LPS22HB_I2C_ADDRESS
224  #else /* USEHTS221TEMP */
225  /* HTS221 Sensor hardware I2C address */
226  #define TSENSOR_I2C_ADDRESS    HTS221_I2C_ADDRESS
227  #endif
```

stm32l475e_iot01.h - 7

```
450 |  */
251 uint32_t      BSP_GetVersion(void);
252 void          BSP_LED_Init(Led_TypeDef Led);
253 void          BSP_LED_DeInit(Led_TypeDef Led);
254 void          BSP_LED_On(Led_TypeDef Led);
255 void          BSP_LED_Off(Led_TypeDef Led);
256 void          BSP_LED_Toggle(Led_TypeDef Led);
257 void          BSP_PB_Init(Button_TypeDef Button, ButtonMode_TypeDef ButtonMode);
258 void          BSP_PB_DeInit(Button_TypeDef Button);
259 uint32_t      BSP_PB_GetState(Button_TypeDef Button);
260 void          BSP_COM_Init(COM_TypeDef COM, UART_HandleTypeDef *husart);
261 void          BSP_COM_DeInit(COM_TypeDef COM, UART_HandleTypeDef *huart);
262 /**
```

Drivers/BSP/B-L475E-
IOT01/
stm32l475e_iot01.c

stm32l475e_iot01.c - Part 1

```
1  /**
2  ****
3  * @file    stm32l475e_iot01.c
4  * @author  MCD Application Team
5  * @brief   STM32L475E-IOT01 board support package
6  ****
.
.
.
46  /* Includes -----
47  #include "stm32l475e_iot01.h"
48
```

stm32l475e_iot01.c - Part 2

```
65  * @brief STM32L475E IOT01 BSP Driver version number
66  */
67 #define __STM32L475E_IOT01_BSP_VERSION_MAIN      (0x01) /*!< [31:24] main version */
68 #define __STM32L475E_IOT01_BSP_VERSION_SUB1       (0x01) /*!< [23:16] sub1 version */
69 #define __STM32L475E_IOT01_BSP_VERSION_SUB2       (0x03) /*!< [15:8]  sub2 version */
70 #define __STM32L475E_IOT01_BSP_VERSION_RC        (0x00) /*!< [7:0]   release candidate */
71 #define __STM32L475E_IOT01_BSP_VERSION            (((__STM32L475E_IOT01_BSP_VERSION_MAIN << 24) \
72 | (__STM32L475E_IOT01_BSP_VERSION_SUB1 << 16) \
73 | (__STM32L475E_IOT01_BSP_VERSION_SUB2 << 8 ) \
74 | (__STM32L475E_IOT01_BSP_VERSION_RC))
```

stm32l475e_iot01.c - Part 3

```
82 |
83 const uint32_t GPIO_PIN[LEDn] = {LED2_PIN};
84
85
86 GPIO_TypeDef* GPIO_PORT[LEDn] = {LED2_GPIO_PORT};
87
```

stm32l475e_iot01.c - Part 4

```
..
89 GPIO_TypeDef* BUTTON_PORT[BUTTONNn] = {USER_BUTTON_GPIO_PORT};
90
91 const uint16_t BUTTON_PIN[BUTTONNn] = {USER_BUTTON_PIN};
92
93 const uint16_t BUTTON IRQn[BUTTONNn] = {USER_BUTTON_EXTI_IRQn};
..
```

stm32l475e_iot01.c - Part 5

```
95 USART_TypeDef* COM_USART[COMn] = {DISCOVERY_COM1};  
96  
97 GPIO_TypeDef* COM_TX_PORT[COMn] = {DISCOVERY_COM1_TX_GPIO_PORT};  
98  
99 GPIO_TypeDef* COM_RX_PORT[COMn] = {DISCOVERY_COM1_RX_GPIO_PORT};  
100  
101 const uint16_t COM_TX_PIN[COMn] = {DISCOVERY_COM1_TX_PIN};  
102  
103 const uint16_t COM_RX_PIN[COMn] = {DISCOVERY_COM1_RX_PIN};  
104  
105 const uint16_t COM_TX_AF[COMn] = {DISCOVERY_COM1_TX_AF};  
106  
107 const uint16_t COM_RX_AF[COMn] = {DISCOVERY_COM1_RX_AF};
```

stm32l475e_iot01.c - Part 6

```
100
109 I2C_HandleTypeDef hI2cHandler;
110 UART_HandleTypeDef hDiscoUart;
111
```

stm32l475e_iot01.c - Part 7

```
126
127 /* Sensors IO functions */
128 void      SENSOR_I0_Init(void);
129 void      SENSOR_I0_DeInit(void);
130 void      SENSOR_I0_Write(uint8_t Addr, uint8_t Reg, uint8_t Value);
131 uint8_t   SENSOR_I0_Read(uint8_t Addr, uint8_t Reg);
132 uint16_t  SENSOR_I0_ReadMultiple(uint8_t Addr, uint8_t Reg, uint8_t *Buffer, uint16_t Length);
133 void      SENSOR_I0_WriteMultiple(uint8_t Addr, uint8_t Reg, uint8_t *Buffer, uint16_t Length);
134 HAL_StatusTypeDef SENSOR_I0_IsDeviceReady(uint16_t DevAddress, uint32_t Trials);
135 void      SENSOR_I0_Delay(uint32_t Delay);
---
```

stm32l475e_iot01.c - Part 8

```
154 /**
155  * @brief This method returns the STM32L475E IOT01 BSP Driver revision
156  * @retval version: 0xXYZR (8bits for each decimal, R for RC)
157  */
158 uint32_t BSP_GetVersion(void)
159 {
160     return __STM32L475E_IOT01_BSP_VERSION;
161 }
```

stm32l475e_iot01.c - Part 9

```
---  
163  /**  
164   * @brief  Configures LEDs.  
165   * @param  Led: LED to be configured.  
166   *          This parameter can be one of the following values:  
167   *          @arg  LED2  
168   */  
169 void BSP_LED_Init(Led_TypeDef Led)  
170 {  
171     GPIO_InitTypeDef  gpio_init_structure;  
172  
173     LEDx_GPIO_CLK_ENABLE(Led);  
174     /* Configure the GPIO_LED pin */  
175     gpio_init_structure.Pin      = GPIO_PIN[Led];  
176     gpio_init_structure.Mode    = GPIO_MODE_OUTPUT_PP;  
177     gpio_init_structure.Pull    = GPIO_NOPULL;  
178     gpio_init_structure.Speed  = GPIO_SPEED_FREQ_HIGH;  
179  
180     HAL_GPIO_Init(GPIO_PORT[Led], &gpio_init_structure);  
181 }
```

stm32l475e_iot01.c - Part 10

```
183  /**
184   * @brief  DeInit LEDs.
185   * @param  Led: LED to be configured.
186   *         This parameter can be one of the following values:
187   *         @arg  LED2
188   */
189 void BSP_LED_DeInit(Led_TypeDef Led)
190 {
191     GPIO_InitTypeDef  gpio_init_structure;
192
193     /* DeInit the GPIO_LED pin */
194     gpio_init_structure.Pin = GPIO_PIN[Led];
195
196     /* Turn off LED */
197     HAL_GPIO_WritePin(GPIO_PORT[Led], GPIO_PIN[Led], GPIO_PIN_RESET);
198     HAL_GPIO_DeInit(GPIO_PORT[Led], gpio_init_structure.Pin);
199 }
```

stm32l475e_iot01.c - Part 11

```
201  /**
202   * @brief Turns selected LED On.
203   * @param Led: LED to be set on
204   *           This parameter can be one of the following values:
205   *           @arg LED2
206   */
207 void BSP_LED_On(Led_TypeDef Led)
208 {
209     HAL_GPIO_WritePin(GPIO_PORT[Led], GPIO_PIN[Led], GPIO_PIN_SET);
210 }
```

stm32l475e_iot01.c - Part 13

```
212  /**
213   * @brief Turns selected LED off.
214   * @param Led: LED to be set off
215   *           This parameter can be one of the following values:
216   *           @arg LED2
217   */
218 void BSP_LED_Off(Led_TypeDef Led)
219 {
220     HAL_GPIO_WritePin(GPIO_PORT[Led], GPIO_PIN[Led], GPIO_PIN_RESET);
221 }
```

~~~

# stm32l475e\_iot01.c - Part 14

```
223  /**
224   * @brief Toggles the selected LED.
225   * @param Led: LED to be toggled
226   *           This parameter can be one of the following values:
227   *           @arg LED2
228   */
229 void BSP_LED_Toggle(Led_TypeDef Led)
230 {
231     HAL_GPIO_TogglePin(GPIO_PORT[Led], GPIO_PIN[Led]);
232 }
```

---

# stm32l475e\_iot01.c - Part 15

```
234 /**
235  * @brief Configures button GPIO and EXTI Line.
236  * @param Button: Button to be configured
237  *               This parameter can be one of the following values:
238  *               @arg BUTTON_WAKEUP: Wakeup Push Button
239  * @param ButtonMode: Button mode
240  *               This parameter can be one of the following values:
241  *               @arg BUTTON_MODE_GPIO: Button will be used as simple IO
242  *               @arg BUTTON_MODE_EXTI: Button will be connected to EXTI line
243  *                           with interrupt generation capability
244 */
245 void BSP_PB_Init(Button_TypeDef Button, ButtonMode_TypeDef ButtonMode)
246 {
247     GPIO_InitTypeDef gpio_init_structure;
248
249     /* Enable the BUTTON clock */
250     USER_BUTTON_GPIO_CLK_ENABLE();
```

# stm32l475e\_iot01.c - Part 16

```
251
252     if(ButtonMode == BUTTON_MODE_GPIO)
253     {
254         /* Configure Button pin as input */
255         gpio_init_structure.Pin = BUTTON_PIN[Button];
256         gpio_init_structure.Mode = GPIO_MODE_INPUT;
257         gpio_init_structure.Pull = GPIO_PULLUP;
258         gpio_init_structure.Speed = GPIO_SPEED_FREQ_HIGH;
259         HAL_GPIO_Init(BUTTON_PORT[Button], &gpio_init_structure);
260     }
261
```

# stm32l475e\_iot01.c - Part 17

```
262     if(ButtonMode == BUTTON_MODE_EXTI)
263     {
264         /* Configure Button pin as input with External interrupt */
265         gpio_init_structure.Pin = BUTTON_PIN[Button];
266         gpio_init_structure.Pull = GPIO_PULLUP;
267         gpio_init_structure.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
268
269         gpio_init_structure.Mode = GPIO_MODE_IT_RISING;
270
271         HAL_GPIO_Init(BUTTON_PORT[Button], &gpio_init_structure);
272
273         /* Enable and set Button EXTI Interrupt to the lowest priority */
274         HAL_NVIC_SetPriority((IRQn_Type)(BUTTON IRQn[Button]), 0x0F, 0x00);
275         HAL_NVIC_EnableIRQ((IRQn_Type)(BUTTON IRQn[Button]));
276     }
277 }
```

# stm32l475e\_iot01.c - Part 18

```
279 /*
● 280     * @brief Push Button DeInit.
281     * @param Button: Button to be configured
282     *             This parameter can be one of the following values:
283     *             @arg BUTTON_WAKEUP: Wakeup Push Button
284     * @note PB DeInit does not disable the GPIO clock
285     */
286 void BSP_PB_DeInit(Button_TypeDef Button)
287 {
288     GPIO_InitTypeDef gpio_init_structure;
289
290     gpio_init_structure.Pin = BUTTON_PIN[Button];
291     HAL_NVIC_DisableIRQ((IRQn_Type)(BUTTON IRQn[Button]));
292     HAL_GPIO_DeInit(BUTTON_PORT[Button], gpio_init_structure.Pin);
293 }
```

# stm32l475e\_iot01.c - Part 19

```
290  /**
291  * @brief Returns the selected button state.
292  * @param Button: Button to be checked
293  *               This parameter can be one of the following values:
294  *               @arg BUTTON_WAKEUP: Wakeup Push Button
295  * @retval The Button GPIO pin value (GPIO_PIN_RESET = button pressed)
296  */
297 uint32_t BSP_PB_GetState(Button_TypeDef Button)
298 {
299     return HAL_GPIO_ReadPin(BUTTON_PORT[Button], BUTTON_PIN[Button]);
300 }
```

Drivers/BSP/B-L475E-  
IOT01/  
stm32l475e\_iot01\_tsensor.h

# stm32l475e\_iot01\_tsensor.h

## Part 1

```
1  /**
2   ****
3   * @file    stm32l475e_iot01_tsensor.h
4   * @author  MCD Application Team
5   * @brief   This file provides a set of functions needed to manage the temperature sensor
6   ****
7
8
9
10  /*
11   * Define to prevent recursive inclusion ---
12  #ifndef __STM32L475E_IOT01_TSSENSOR_H
13  #define __STM32L475E_IOT01_TSSENSOR_H
14
15
16  /*
17   * Includes -----
18  #include "stm32l475e_iot01.h"
19  #ifdef USE_LPS22HB_TEMP
20  #include "../Components/lps22hb/lps22hb.h"
21  #else /* USEHTS221TEMP */
22  #include "../Components/hts221/hts221.h"
23  #endif
```

# stm32l475e\_iot01\_tsensor.h

## Part 2

```
78  /**
79   * @brief TSENSOR Status
80   */
81  typedef enum
82  {
83      TSENSOR_OK = 0,
84      TSENSOR_ERROR
85 }TSENSOR_Status_TypDef;
86
87  /* ----- */
88
89  /* Sensor Configuration Functions */
90  uint32_t BSP_TSENSOR_Init(void);
91  float  BSP_TSENSOR_ReadTemp(void);
92
```

Drivers/BSP/B-L475E-  
IOT01/  
stm32l475e\_iot01\_tsensor.c

# stm32l475e\_iot01\_tsensor.c

## Part 1

```
2 ****  
3 * @file    stm32l475e_iot01_tsensor.c  
4 * @author   MCD Application Team  
5 * @brief    This file provides a set of functions needed to manage the temperature sensor  
6 ****  
  
/* Includes -----  
#include "stm32l475e_iot01_tsensor.h"  
  
63  */  
64 static TSENSOR_DrvTypeDef *tsensor_drv;  
65 /**
```

# stm32l475e\_iot01\_tsensor.c

## Part 2

```
73  /**
74   * @brief  Initializes peripherals used by the I2C Temperature Sensor driver.
75   * @retval TSENSOR status
76   */
77 uint32_t BSP_TSENSOR_Init(void)
78 {
79     uint8_t ret = TSENSOR_ERROR;
80
81 #ifdef USE_LPS22HB_TEMP
82     tsensor_drv = &LPS22HB_T_Drv;
83 #else /* USEHTS221TEMP */
84     tsensor_drv = &HTS221_T_Drv;
85 #endif
86
87     /* Low level init */
88     SENSOR_IO_Init();
89
90     /* TSENSOR Init */
91     tsensor_drv->Init(TSENSOR_I2C_ADDRESS, NULL);
92
93     ret = TSENSOR_OK;
94
95     return ret;
96 }
```

# stm32l475e\_iot01\_tsensor.c

## Part 3

```
98  /**
99   * @brief  Read Temperature register of TS751.
100  * @retval STTS751 measured temperature value.
101 */
102 float BSP_TSSENSOR_ReadTemp(void)
103 {
104     return tsensor_drv->ReadTemp(TSENSOR_I2C_ADDRESS);
105 }
```

# Drivers/BSP/ Components/Common/ tsensor.h

# Components/Common/ tsensor.h - Part 1

```
2  ****
3  * @file    tsensor.h
4  * @author   MCD Application Team
5  * @brief    This header file contains the functions prototypes for the
6  *           Temperature Sensor driver.
7  ****

37 /* Define to prevent recursive inclusion --
38 #ifndef __TSENSOR_H
39 #define __TSENSOR_H
40
41 #ifdef __cplusplus
42 | extern "C" {
43 #endif
44
45 /* Includes -----
46 #include <stdint.h>
47
```

# Components/Common/ tsensor.h - Part 2

```
```
64  /** @defgroup TSENSOR_Config_structure  Temperature Sensor Configuration structure
65  * @{
66  */
67  typedef struct
68  {
69      uint8_t AlertMode;          /* Alert Mode Temperature out of range*/
70      uint8_t ConversionMode;    /* Continuous/One Shot Mode */
71      uint8_t ConversionResolution; /* Temperature Resolution */
72      uint8_t ConversionRate;    /* Number of measure per second */
73      int8_t TemperatureLimitHigh; /* High Temperature Limit Range */
74      int8_t TemperatureLimitLow; /* Low Temperature Limit Range */
75 }TSENSOR_InitTypeDef;
```

# Components/Common/ tsensor.h - Part 3

```
80  /** @defgroup TSENSOR_Driver_structure  Temperature Sensor Driver structure
81  * @{
82  */
83  typedef struct
84  {
85      void          (*Init)(uint16_t, TSENSOR_InitTypeDef *);
86      uint8_t       (*IsReady)(uint16_t, uint32_t);
87      uint8_t       (*ReadStatus)(uint16_t);
88      float         (*ReadTemp)(uint16_t);
89 }TSENSOR_DrvTypeDef;
```

Drivers/BSP/  
Components/hts221/  
hts221.h

# Drivers/BSP/Components/ hts221.h - Part 1

```
2 ****  
3 * @file    hts221.h  
4 * @author   MCD Application Team  
5 * @version  V1.0.0  
6 * @date    14-February-2017  
7 * @brief   HTS221 header driver file  
8 ****  
38 /* Define to prevent recursive inclusion  
39 #ifndef __HTS221__H  
40 #define __HTS221__H  
41  
42 #ifdef __cplusplus  
43 extern "C" {  
44 #endif  
45  
46 /* Includes -----  
47 #include "../Common/hsensor.h"  
48 #include "../Common/tsensor.h"  
49
```

# Drivers/BSP/Components/ hts221.h - Part 2

```
66  /**
67   * @brief Bitfield positioning.
68   */
69 #define HTS221_BIT(x) ((uint8_t)x)

'2  /**
73   * @brief Device Identification register.
74   *        Read
75   *        Default value: 0xBC
76   *        7:0 This read-only register contains the device identifier for HTS221.
77   */
78 #define HTS221_WHO_AM_I_REG          (uint8_t)0x0F
79

79
80 /**
81  * @brief Device Identification value.
82  */
83 #define HTS221_WHO_AM_I_VAL         (uint8_t)0xBC
84
```

# Drivers/BSP/Components/ hts221.h - Part 3

```
86 /**
87  * @brief Humidity and temperature average mode register.
88  *        Read/write
89  *        Default value: 0x1B
90  *        7:6 Reserved.
91  *        5:3 AVGT2–AVGT1–AVGT0: Select the temperature internal average.
92  *
```

```
116 | */
117 #define HTS221_AV_CONF_REG          (uint8_t)0x10
118
119 #define HTS221_AVGT_BIT             HTS221_BIT(3)
120 #define HTS221_AVGH_BIT             HTS221_BIT(0)
121
122 #define HTS221_AVGH_MASK            (uint8_t)0x07
123 #define HTS221_AVGT_MASK            (uint8_t)0x38
124
```

# Drivers/BSP/Components/ hts221.h - Part 4

```
125 /**
126  * @brief Control register 1.
127  *        Read/write
128  *        Default value: 0x00
129  *        7 PD: power down control. 0 – power down mode; 1 – active mode.
130  *        6:3 Reserved.
131  *        2 BDU: block data update. 0 – continuous update
132  *                  1 – output registers not updated until MSB and LSB reading.
133  *        1:0 ODR1, ODR0: output data rate selection.

142 |  */
143 #define HTS221_CTRL_REG1      (uint8_t)0x20
144
145 #define HTS221_PD_BIT          HTS221_BIT(7)
146 #define HTS221_BDU_BIT         HTS221_BIT(2)
147 #define HTS221_ODR_BIT         HTS221_BIT(0)
148
149 #define HTS221_PD_MASK         (uint8_t)0x80
150 #define HTS221_BDU_MASK        (uint8_t)0x04
151 #define HTS221_ODR_MASK        (uint8_t)0x03
152
```

# Drivers/BSP/Components/ hts221.h - Part 5

```
153 /**
154  * @brief Control register 2.
155  *        Read/write
156  *        Default value: 0x00
157  *        7 BOOT: Reboot memory content. 0: normal mode
158  *                  1: reboot memory content. Self-cleared upon completion.
159  *        6:2 Reserved.
160  *        1 HEATHER: 0: heater enable; 1: heater disable.
161  *        0 ONE_SHOT: 0: waiting for start of conversion
162  *                  1: start for a new dataset. Self-cleared upon completion.
163 */
164 #define HTS221_CTRL_REG2      (uint8_t)0x21
165
166 #define HTS221_BOOT_BIT        HTS221_BIT(7)
167 #define HTS221_HEATHER_BIT     HTS221_BIT(1)
168 #define HTS221_ONESHOT_BIT    HTS221_BIT(0)
169
170 #define HTS221_BOOT_MASK       (uint8_t)0x80
171 #define HTS221_HEATHER_MASK   (uint8_t)0x02
172 #define HTS221_ONE_SHOT_MASK  (uint8_t)0x01
173
```

# Drivers/BSP/Components/ hts221.h - Part 6

```
175     * @brief Control register 3.  
176     *          Read/write  
177     *          Default value: 0x00  
178     *          7 DRDY_H_L: Interrupt edge. 0: active high, 1: active low.  
179     *          6 PP_OD: Push-Pull/OpenDrain selection on interrupt pads. 0: push-pull  
180     *                           1: open drain.  
181     *          5:3 Reserved.  
182     *          2 DRDY: interrupt config. 0: disable, 1: enable.  
183     */  
184 #define HTS221_CTRL_REG3      (uint8_t)0x22  
185  
186 #define HTS221_DRDY_H_L_BIT    HTS221_BIT(7)  
187 #define HTS221_PP_OD_BIT       HTS221_BIT(6)  
188 #define HTS221_DRDY_BIT        HTS221_BIT(2)  
189  
190 #define HTS221_DRDY_H_L_MASK   (uint8_t)0x80  
191 #define HTS221_PP_OD_MASK     (uint8_t)0x40  
192 #define HTS221_DRDY_MASK       (uint8_t)0x04
```

# Drivers/BSP/Components/ hts221.h - Part 7

```
195  * @brief Status register.  
196  *  
197  * Default value: 0x00  
198  * 7:2 Reserved.  
199  * 1 H_DA: Humidity data available. 0: new data for humidity is not yet available  
200  * 1: new data for humidity is available.  
201  * 0 T_DA: Temperature data available. 0: new data for temperature is not yet available  
202  * 1: new data for temperature is available.  
203 */  
204 #define HTS221_STATUS_REG      (uint8_t)0x27  
205  
206 #define HTS221_H_DA_BIT        HTS221_BIT(1)  
207 #define HTS221_T_DA_BIT        HTS221_BIT(0)  
208  
209 #define HTS221_HDA_MASK        (uint8_t)0x02  
210 #define HTS221_TDA_MASK        (uint8_t)0x01  
211
```

# Drivers/BSP/Components/ hts221.h - Part 8

```
287  /** @defgroup HTS221_Temperature_Exported_Functions HTS221 Temperature Exported Functions
288  * @{
289  */
290  /* TEMPERATURE functions */
291  void HTS221_T_Init(uint16_t DeviceAddr, TSENSOR_InitTypeDef *pInitStruct);
292  float HTS221_T_ReadTemp(uint16_t DeviceAddr);
293  /**
294  * @}
295  */
296
297  /** @defgroup HTS221_TempImported_Globals Temperature Imported Globals
298  * @{
299  */
300  /* Temperature driver structure */
301  extern TSENSOR_DrvTypeDef HTS221_T_Drv;
302
```

# Drivers/BSP/Components/ hts221.h - Part 9

```
307  /** @defgroup HTS221_Imported_Functions HTS221 Imported Functions
308  * @{
309  */
310 /* I/O functions */
311 extern void      SENSOR_I0_Init(void);
312 extern void      SENSOR_I0_Write(uint8_t Addr, uint8_t Reg, uint8_t Value);
313 extern uint8_t   SENSOR_I0_Read(uint8_t Addr, uint8_t Reg);
314 extern uint16_t  SENSOR_I0_ReadMultiple(uint8_t Addr, uint8_t Reg, uint8_t *Buffer, uint16_t Length);
315 extern void      SENSOR_I0_WriteMultiple(uint8_t Addr, uint8_t Reg, uint8_t *Buffer, uint16_t Length);
316 ...
```

Drivers/BSP/  
Components/hts221/  
hts221.c

# Drivers/BSP/Components/ hts221.c - Part 1

```
2 ****  
3 * @file    hts221.c  
4 * @author   MCD Application Team  
5 * @version  V1.0.0  
6 * @date    14-February-2017  
7 * @brief   This file provides a set of functions needed to manage the HTS221  
8 *          humidity and temperature devices  
9 ****  
  
50 |  
39 /* Includes -----  
40 #include "hts221.h"  
41 |
```

# Drivers/BSP/Components/ hts221.c - Part 2

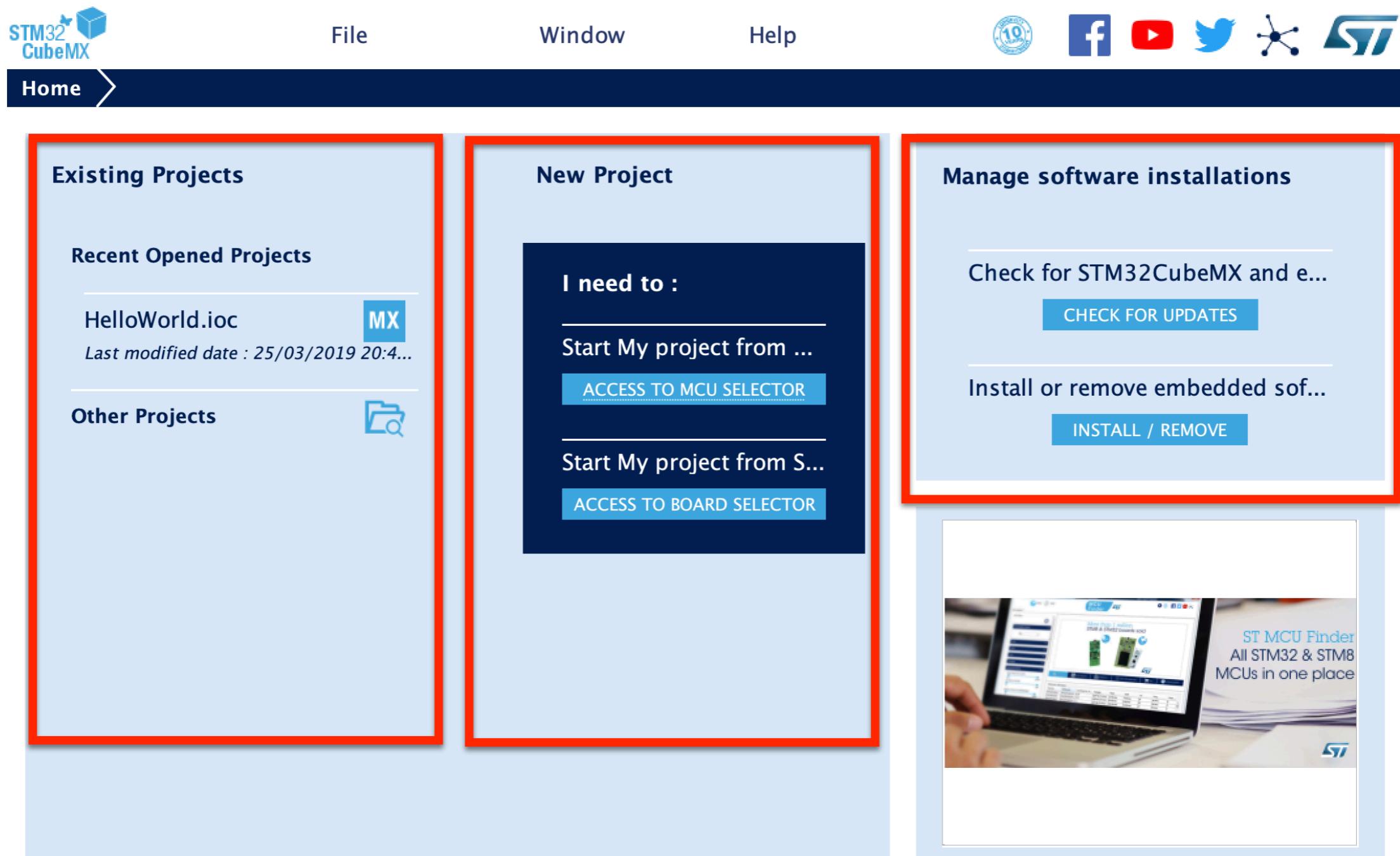
```
..| ..
57 /* HTS221 Humidity Private Variables */
58 HSENSOR_DrvTypeDef HTS221_H_Drv =
59 {
60     HTS221_H_Init,
61     HTS221_H_ReadID,
62     HTS221_H_ReadHumidity
63 };
--| --
65 /* HTS221_Temperature_Private_Variables */
66 TSENSOR_DrvTypeDef HTS221_T_Drv =
67 {
68     HTS221_T_Init,
69     0,
70     0,
71     HTS221_T_ReadTemp
72 };
```

# Drivers/BSP/Components/ hts221.c - Part 3

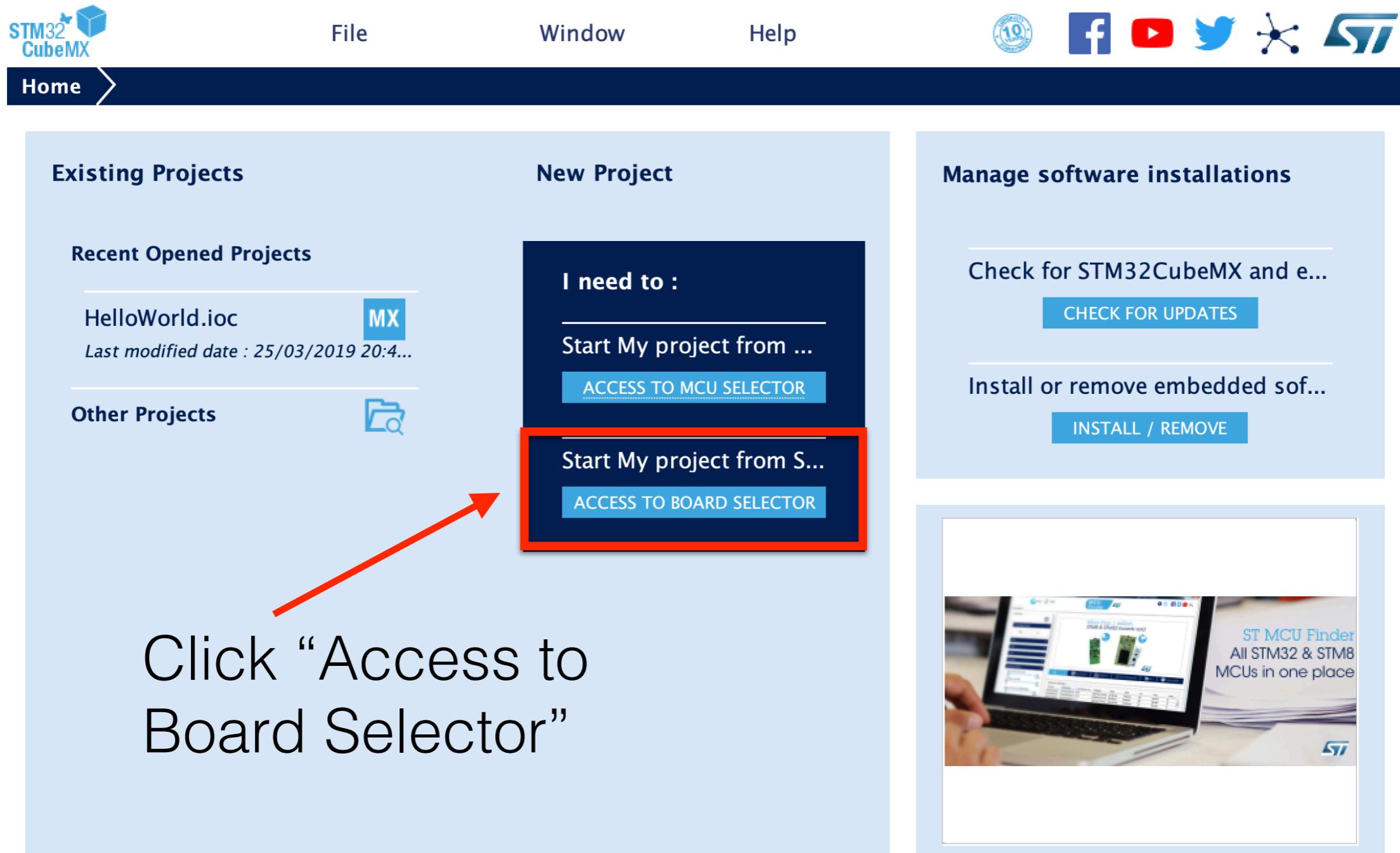
```
83 void HTS221_H_Init(uint16_t DeviceAddr)
84 {
85     uint8_t tmp;
86
87     /* Read CTRL_REG1 */
88     tmp = SENSOR_I0_Read(DeviceAddr, HTS221_CTRL_REG1);
89
90     /* Enable BDU */
91     tmp &= ~HTS221_BDU_MASK;
92     tmp |= (1 << HTS221_BDU_BIT);
93
94     /* Set default ODR */
95     tmp &= ~HTS221_ODR_MASK;
96     tmp |= (uint8_t)0x01; /* Set ODR to 1Hz */
97
98     /* Activate the device */
99     tmp |= HTS221_PD_MASK;
100
101    /* Apply settings to CTRL_REG1 */
102    SENSOR_I0_Write(DeviceAddr, HTS221_CTRL_REG1, tmp);
103 }
```

# STM32CubeMX and Generation of BSP Code

# Step: Startup STM32CubeMX



# Step: Click on “Access to Board Selector”

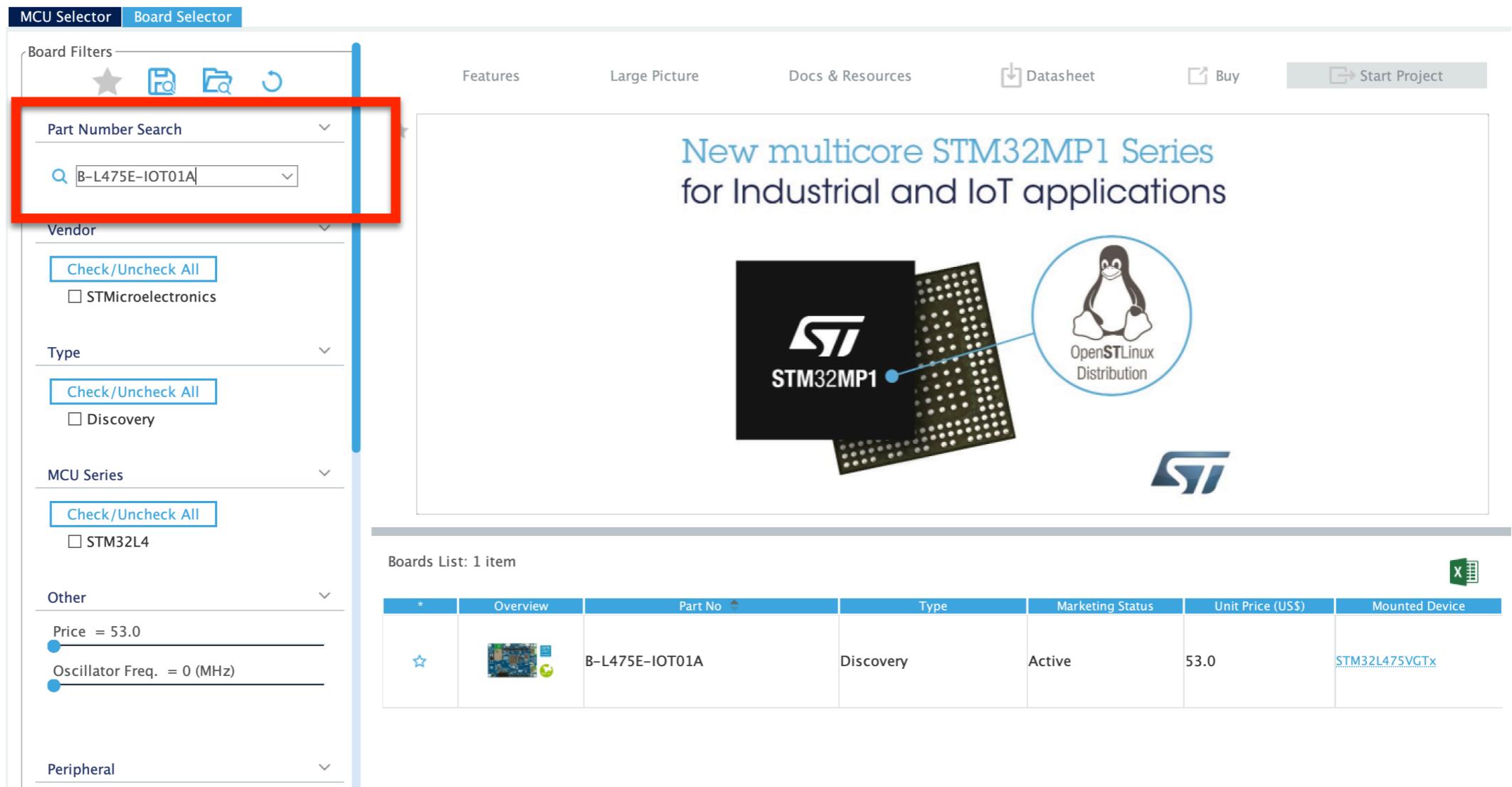


# Step: Observe “Part Number Search”

The screenshot shows the STMicroelectronics website interface. On the left, there's a sidebar with filters for 'Board Filters' (Part Number Search, Vendor, Type), 'MCU Series' (Check/Uncheck All for STM32F0, STM32F1, STM32F2, STM32F3, STM32F4), and 'MCU Selector' (Check/Uncheck All for Discovery, Evaluation Board, Nucleo USB Dongle, Nucleo144, Nucleo32, Nucleo64). A red box highlights the 'Part Number Search' input field. The main content area features a large image of an STM32MP1 chip with a Linux logo and the text 'New multicore STM32MP1 Series for Industrial and IoT applications'. Below this is a table titled 'Boards List: 107 items' with columns for Overview, Part No, Type, Marketing Status, Unit Price (US\$), and Mounted Device. Two rows are visible: one for '32F0308DISCOVERY' (Discovery, Active, \$8.9, STM32F030R8Tx) and another for '32F072BDISCOVERY' (Discovery, Active, \$10.4, STM32F072RBTx).

| * | Overview | Part No          | Type      | Marketing Status | Unit Price (US\$) | Mounted Device                |
|---|----------|------------------|-----------|------------------|-------------------|-------------------------------|
| ☆ |          | 32F0308DISCOVERY | Discovery | Active           | 8.9               | <a href="#">STM32F030R8Tx</a> |
| ☆ |          | 32F072BDISCOVERY | Discovery | Active           | 10.4              | <a href="#">STM32F072RBTx</a> |

# Step: Enter "B-L475E-IOT01A"



The screenshot shows the STMicroelectronics Board Selector interface. On the left, there are various filters: Part Number Search (with a red box around it), Vendor (STMicroelectronics selected), Type (Discovery selected), MCU Series (STM32L4 selected), Other (Price = 53.0, Oscillator Freq. = 0 MHz), and Peripheral. The main area displays a banner for the New multicore STM32MP1 Series for Industrial and IoT applications, featuring an STM32MP1 chip and a Linux penguin icon. Below the banner is a table titled 'Boards List: 1 item' with one row containing the information: Overview, Part No (B-L475E-IOT01A), Type (Discovery), Marketing Status (Active), Unit Price (US\$) (53.0), and Mounted Device (STM32L475VGTx).

| * | Overview | Part No        | Type      | Marketing Status | Unit Price (US\$) | Mounted Device |
|---|----------|----------------|-----------|------------------|-------------------|----------------|
|   |          | B-L475E-IOT01A | Discovery | Active           | 53.0              | STM32L475VGTx  |

# Step: Click on Image, and observe “Features”

MCU Selector Board Selector

Board Filters

- Part Number Search: B-L475E-IOT01A
- Vendor: STMicroelectronics
- Type: Discovery
- MCU Series: STM32L4
- Other: Price = 53.0, Oscillator Freq. = 0 (MHz)
- Peripheral

Features Large Picture Docs & Resources Datasheet Buy Start Project

B-L475E-IOT01A

STM32L4

STM32 L4

STM32L475VGTx

Unit Price (US\$) : 53.0

ACTIVE Active Product is in mass production

The B-L475E-IOT01A Discovery kit for IoT node allows users to develop applications with direct connection to cloud servers. The Discovery kit enables a wide diversity of applications by exploiting low-power communication, multiway sensing and ARM Cortex -M4 core-based STM32L4 Series features. The support for Arduino Uno V3 and PMOD connectivity provides unlimited expansion capabilities with a large choice of specialized add-on boards.

Features

- On-board ST-LINK/V2-1
- Supply through ST-Link USB
- USB OTG(Full speed) with micro AB Connector

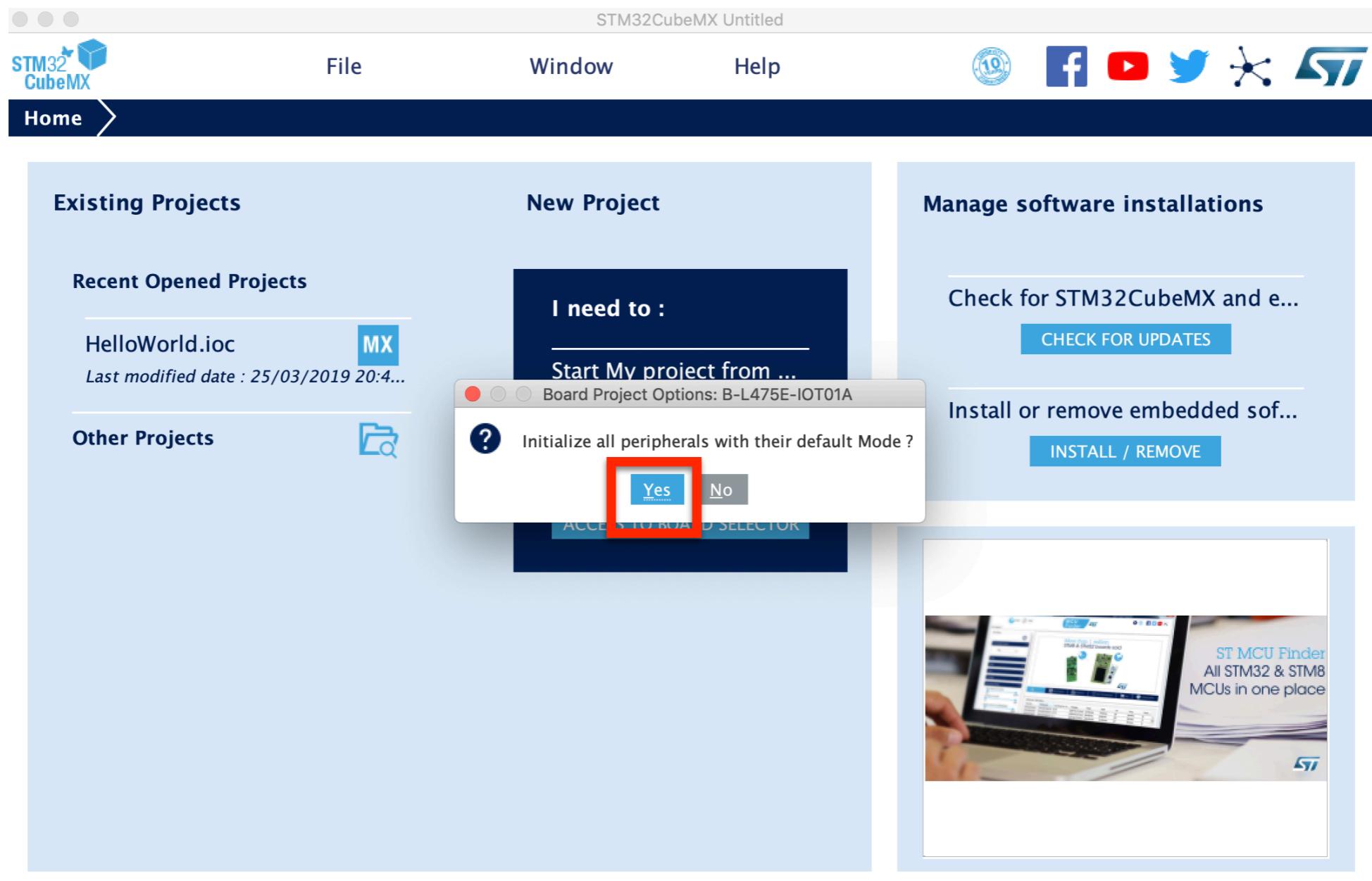
Boards List: 1 item

| * | Overview                                                                             | Part No        | Type      | Marketing Status | Unit Price (US\$) | Mounted Device |
|---|--------------------------------------------------------------------------------------|----------------|-----------|------------------|-------------------|----------------|
| ★ |  | B-L475E-IOT01A | Discovery | Active           | 53.0              | STM32L475VGTx  |

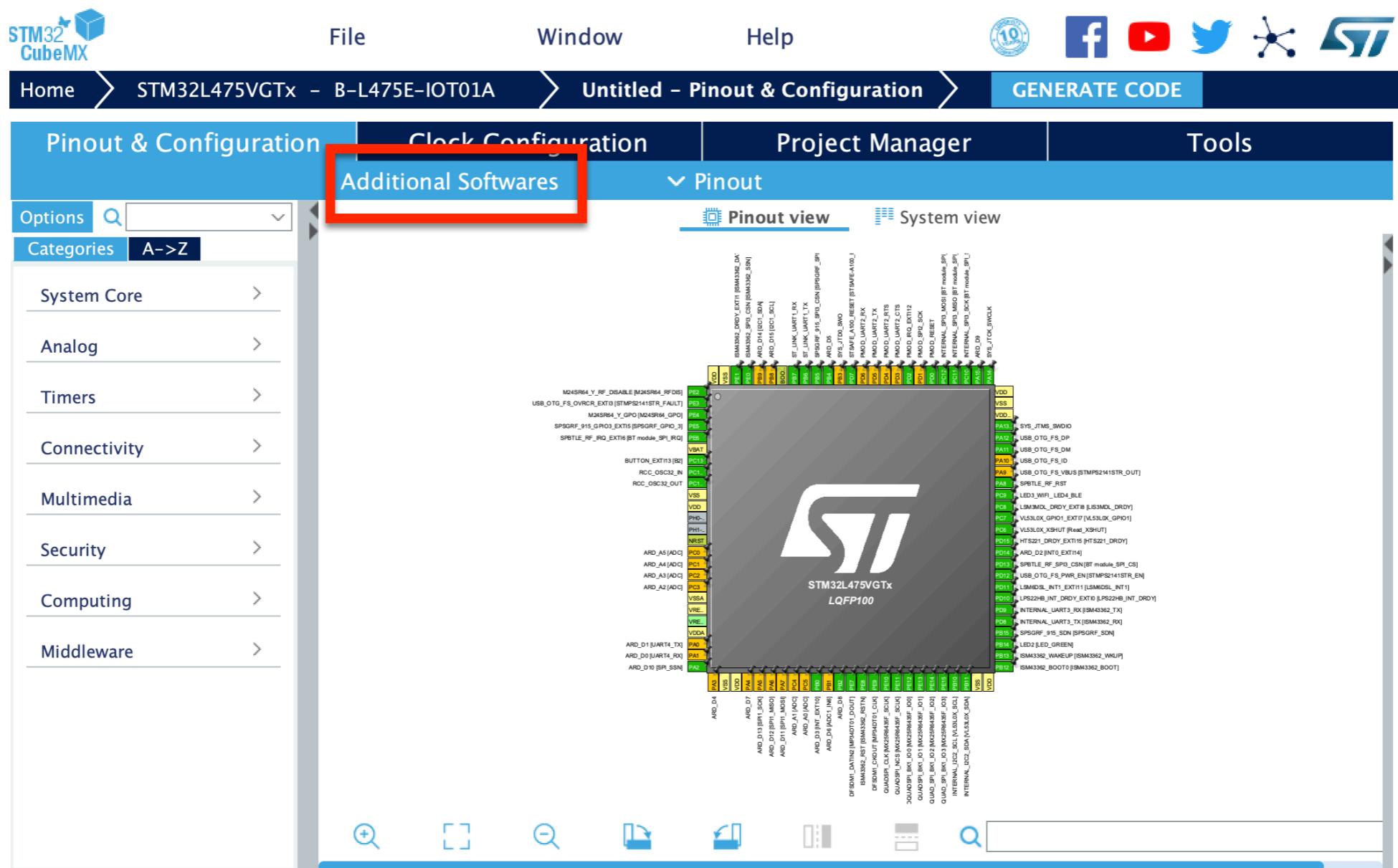
# Step: Click on “Start Project”

The screenshot shows a web-based board selector interface. On the left, there are several filter sections: 'Board Filters' (Part Number Search set to B-L475E-IOT01A), 'Vendor' (Check/Uncheck All, STMMicroelectronics selected), 'Type' (Check/Uncheck All, Discovery selected), 'MCU Series' (Check/Uncheck All, STM32L4 selected), 'Other' (Price = 53.0, Oscillator Freq. = 0 (MHz)), and 'Peripheral'. On the right, the main content area displays the B-L475E-IOT01A board details. It includes sections for 'Features', 'Large Picture', 'Docs & Resources' (which is underlined in blue), 'Datasheet', 'Buy', and a 'Start Project' button, which is highlighted with a red box. Below this, there are links for 'Data brief' (DB3143) and 'User manual' (UM2153). At the bottom, a 'Boards List' table shows one item: B-L475E-IOT01A, with columns for \* (star icon), Overview (image of the board), Part No (B-L475E-IOT01A), Type (Discovery), Marketing Status (Active), Unit Price (US\$) (53.0), and Mounted Device (STM32L475VGTx).

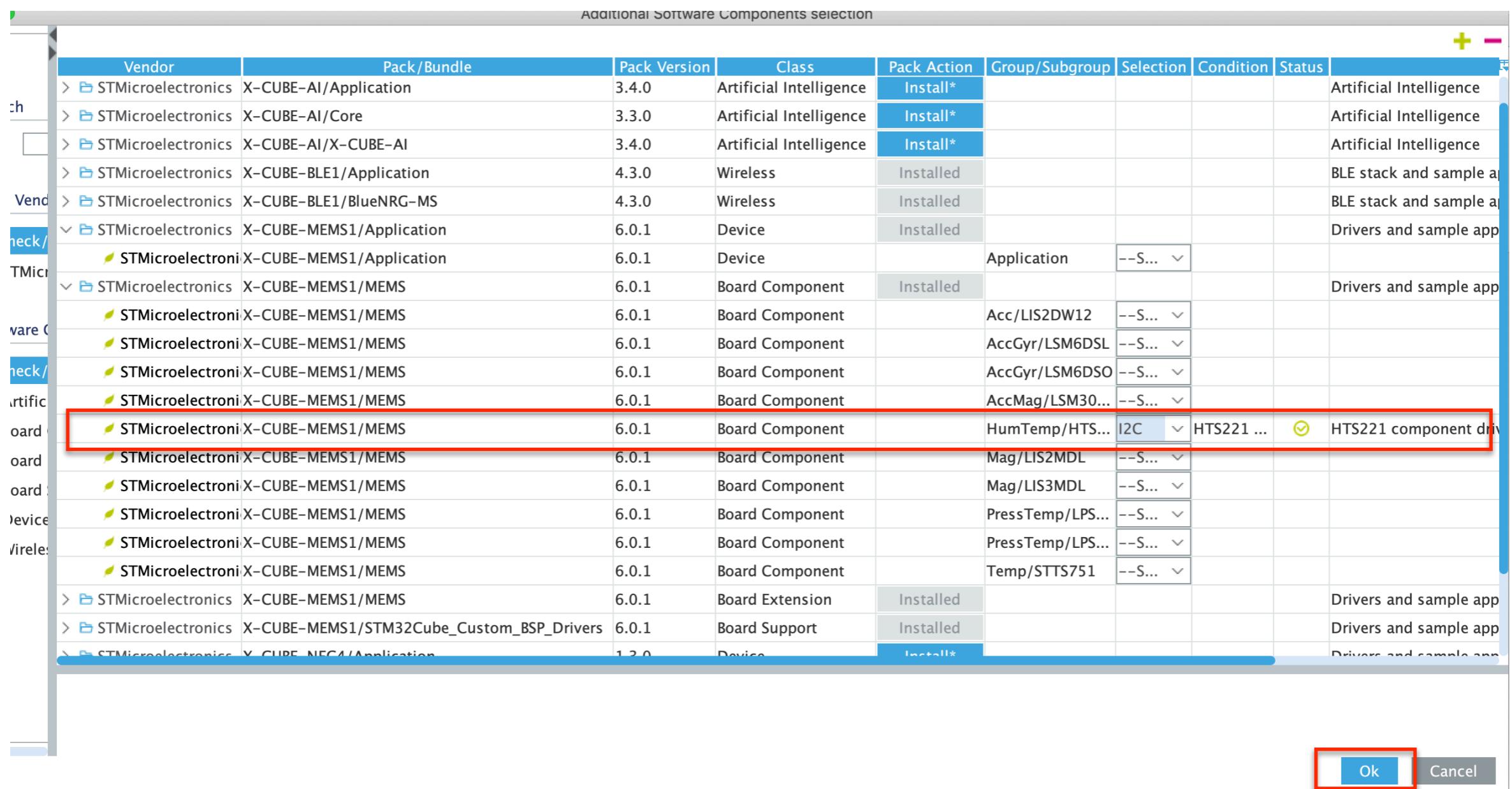
# Step: Click on “Yes” (Initialize all ...with default mode)



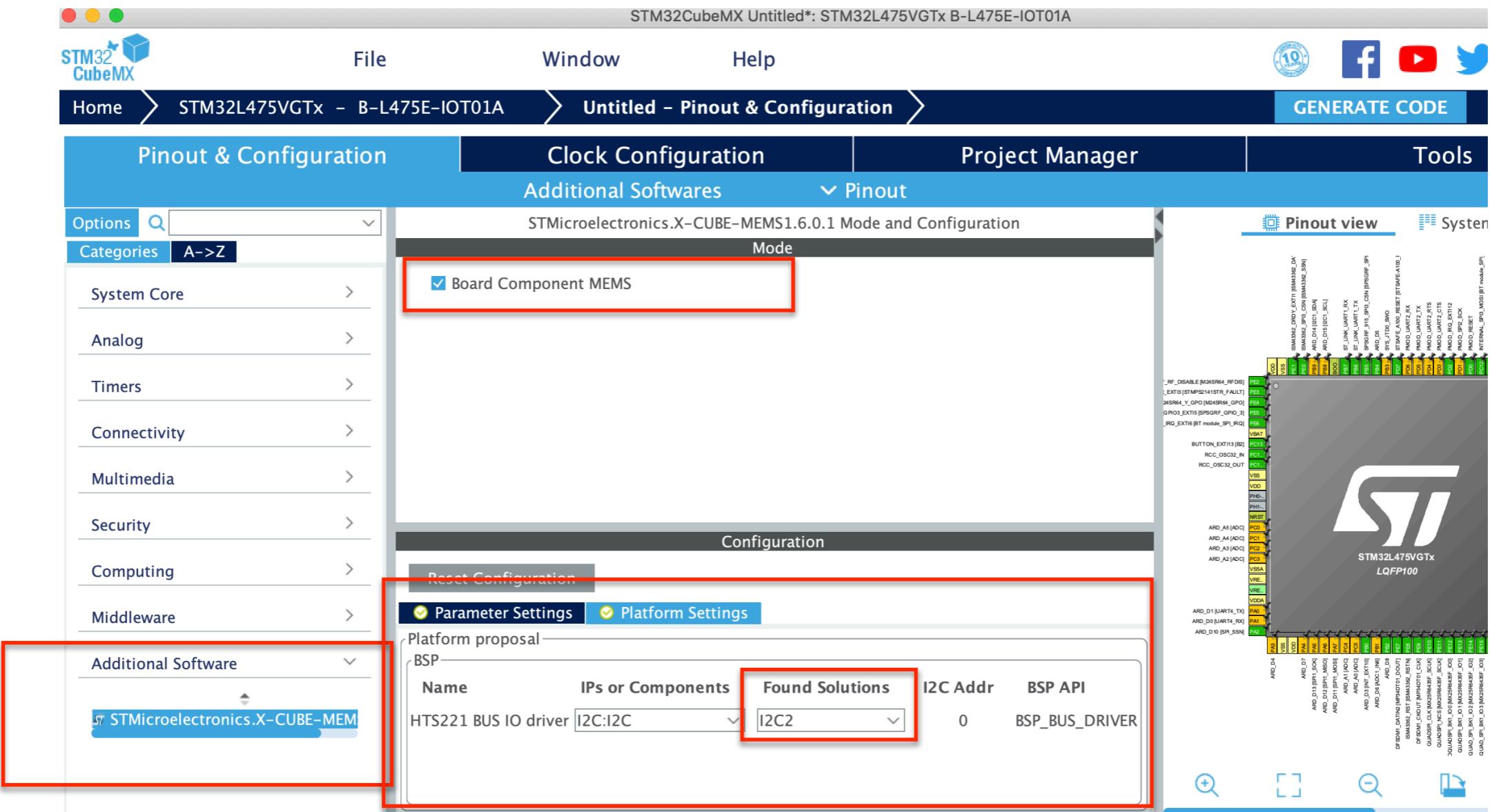
# Step: Click on “Additional Software”



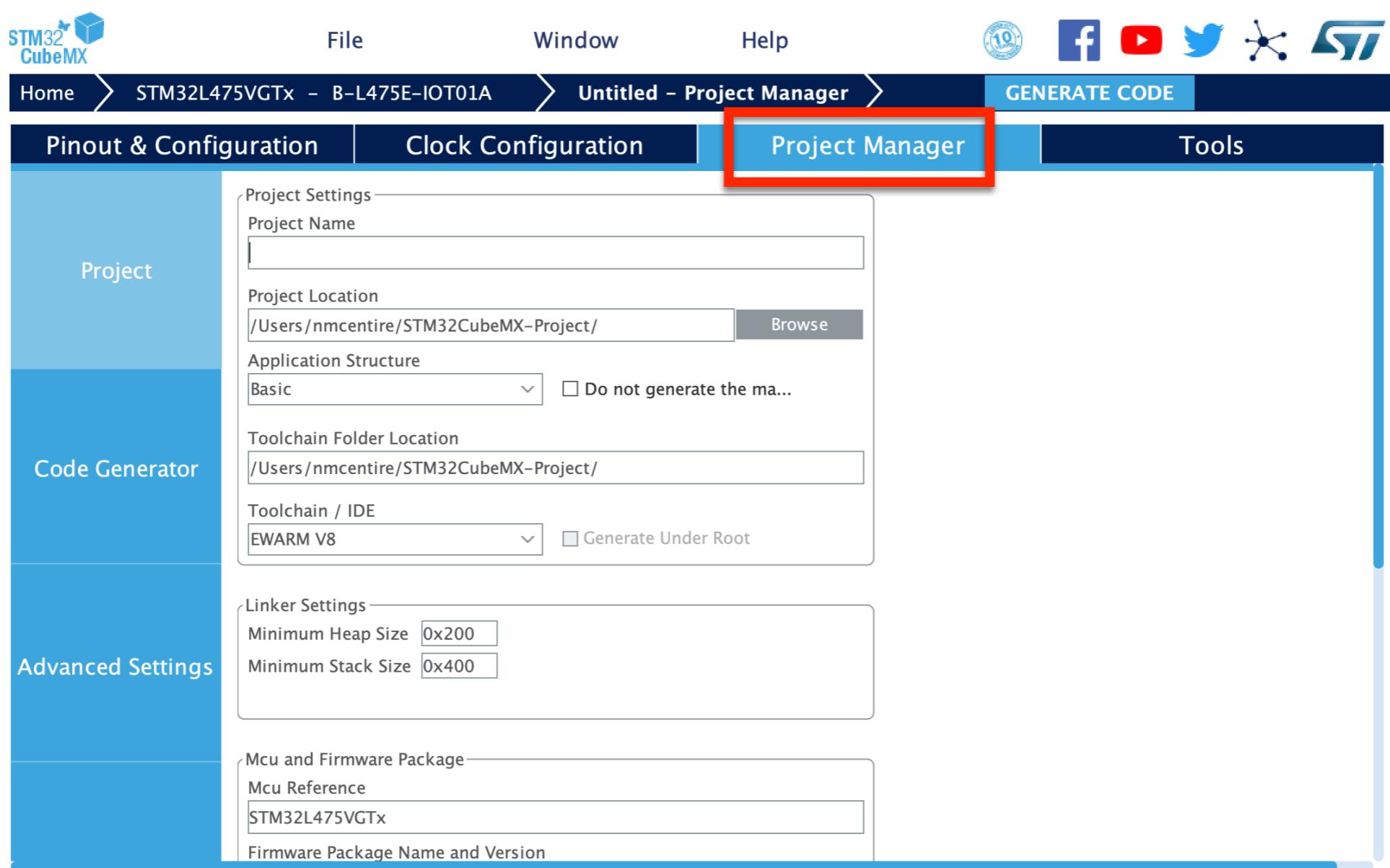
# Step: Select “I2C” for HumTemp/HTS221



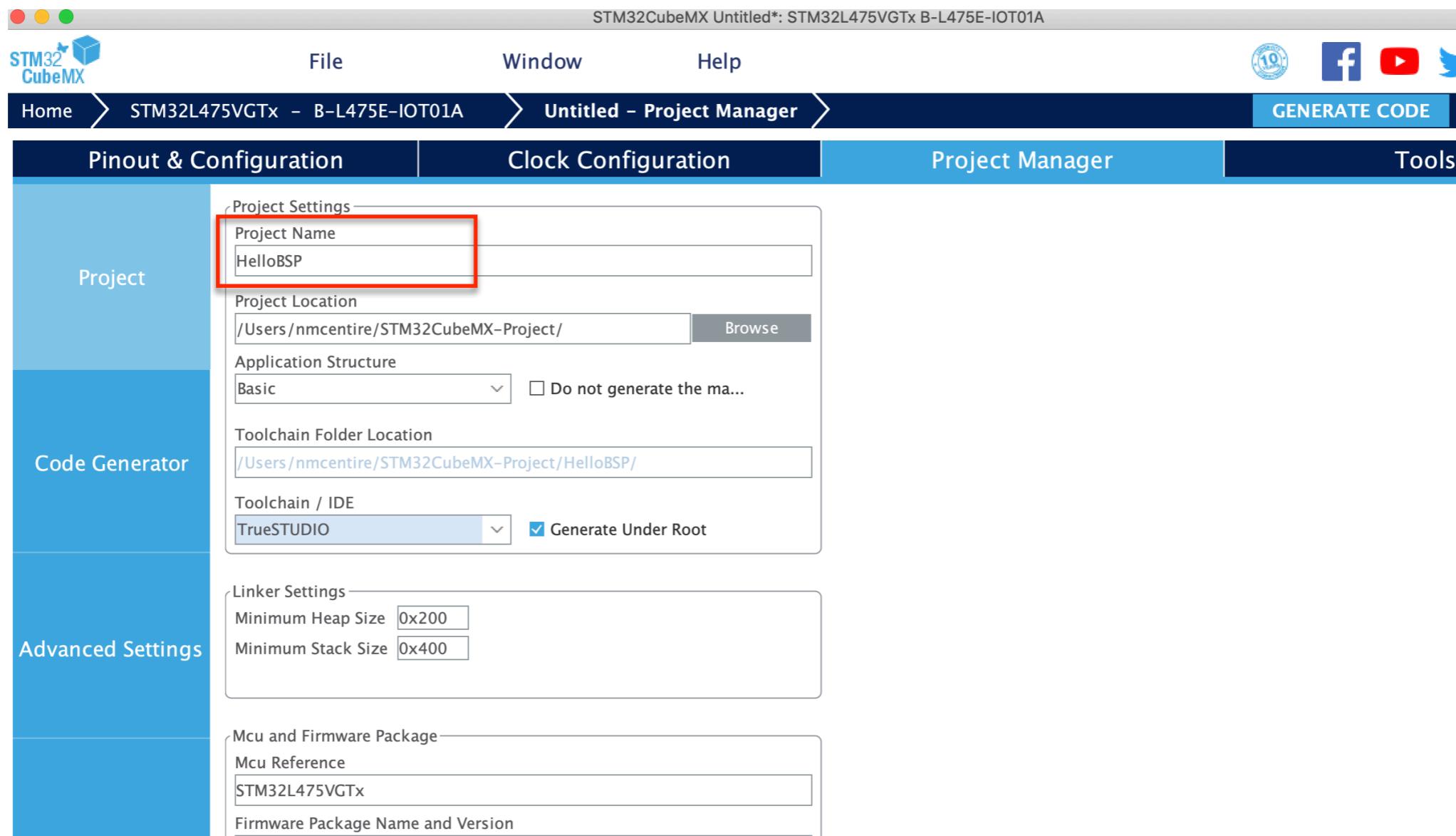
# Step: Check on “Board Component MEMS, select I2C



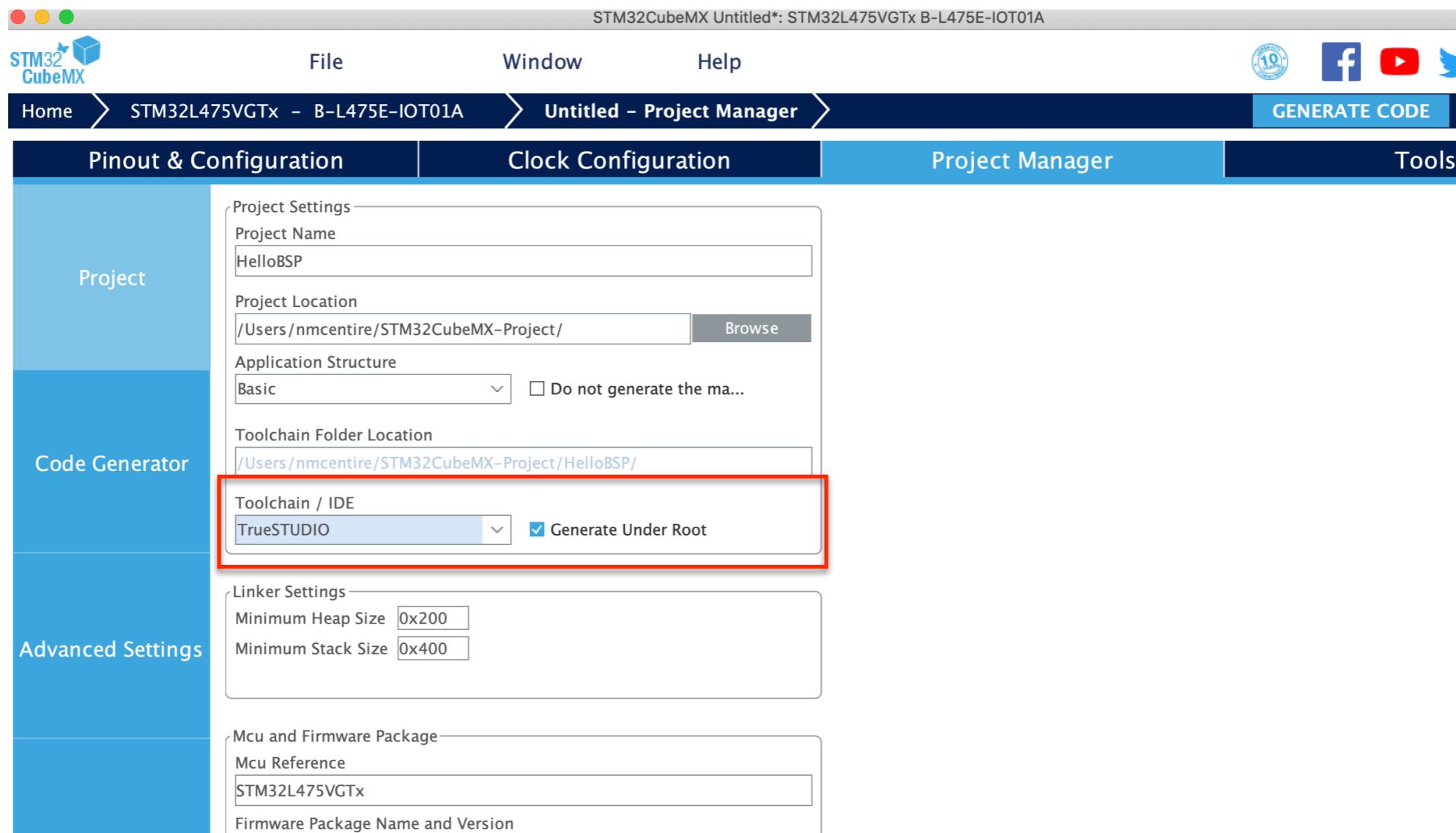
# Step: Observe “Project Manager”



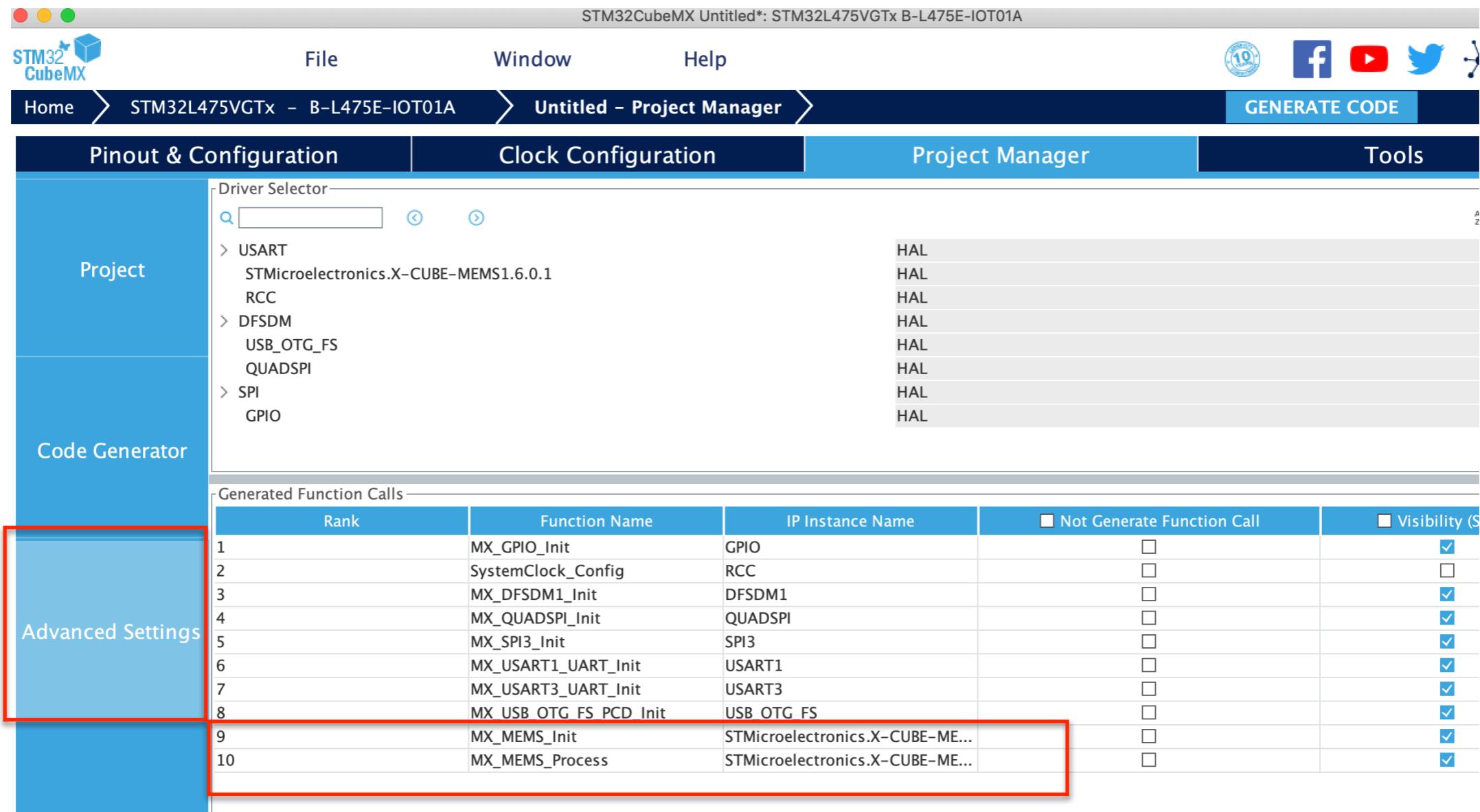
# Step: Enter “HelloBSP” for Project Name



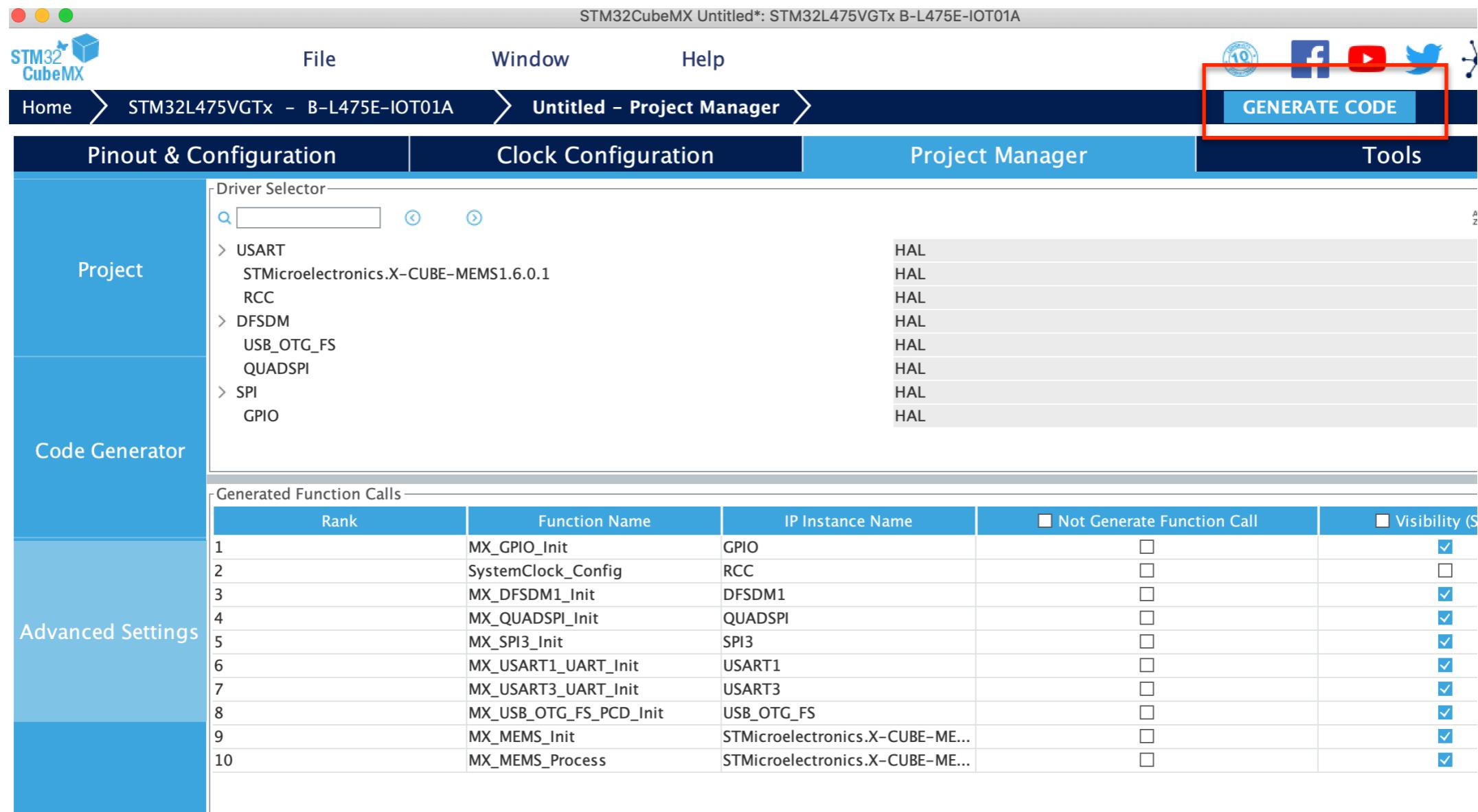
# Step: Select “TrueStudio” for Toolchain / IDE



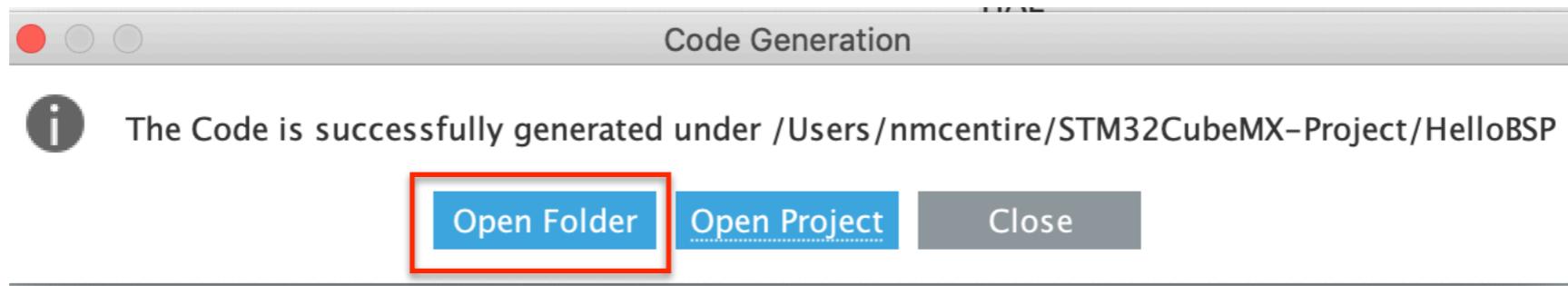
# Step: Click on “Advanced Settings”, observing MX\_MEMS\_Init



# Step: Click on “Generate Code”

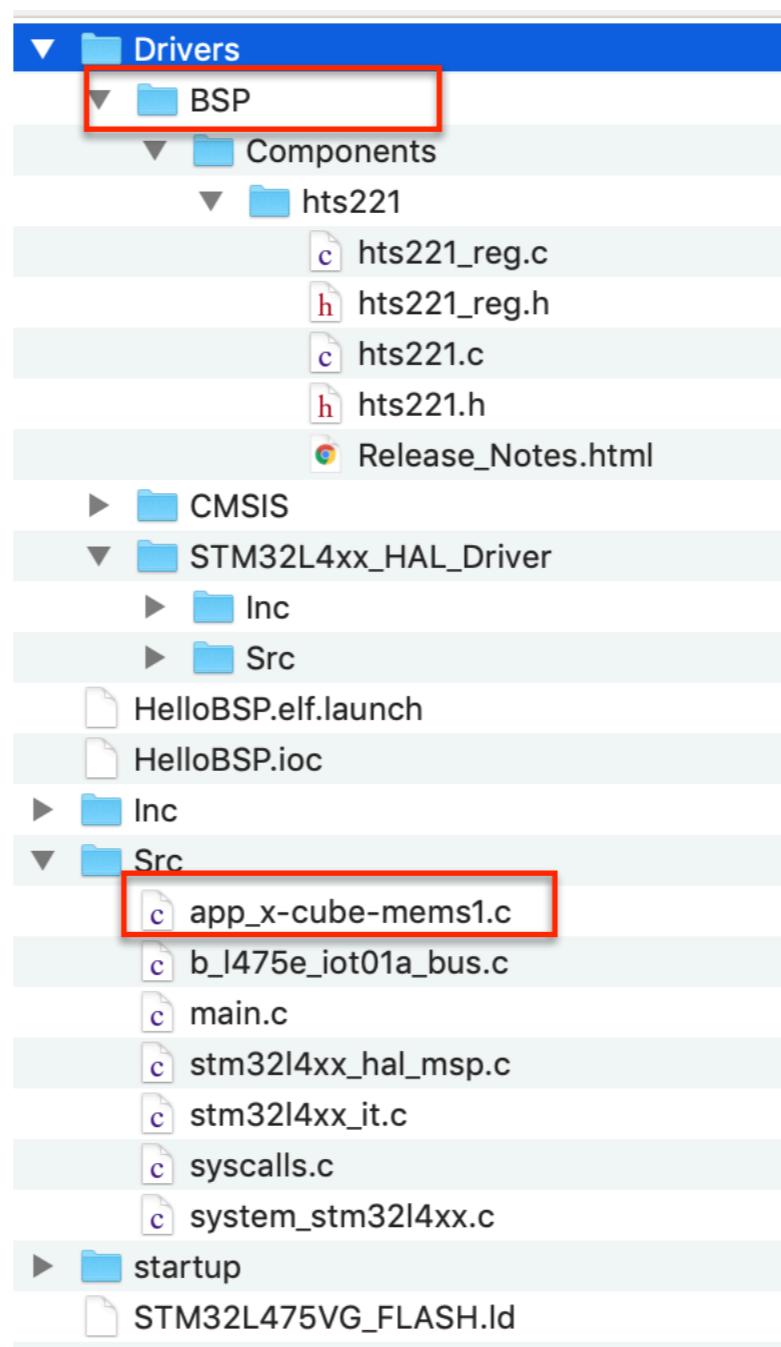


# Step: Select “Open Folder”



# Tour of Generated Project

# Step: View Open Folder



# Src/main.c - Part 1

```
21 /* Includes -----  
22 #include "main.h"  
23 #include "app_x-cube-mems1.h"  
24  
25 //-----  
79 /**  
80  * @brief  The application entry point.  
81  * @retval int  
82  */  
83 int main(void)  
84 {  
85  /* USER CODE BEGIN 1 */  
86  
87  /* USER CODE END 1 */  
88  
89  /* MCU Configuration-----  
90  
91  /* Reset of all peripherals, Initializes  
92  HAL_Init();
```

# Src/main.c - Part 2

```
/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_DFSDM1_Init();
MX_QUADSPI_Init();
MX_SPI3_Init();
MX_USART1_UART_Init();
MX_USART3_UART_Init();
MX_USB_OTG_FS_PCD_Init();
MX_MEMS_Init();
```

# Src/main.c - Part 3

```
118     /* Infinite loop */
119     /* USER CODE BEGIN WHILE */
120     while (1)
121     {
122         /* USER CODE END WHILE */
123
124         MX_MEMS_Process();
125         /* USER CODE BEGIN 3 */
126     }
127     /* USER CODE END 3 */
128 }
```

# Src/app\_x-cube-mems1.c - Part 1

```
1  /**
2   ****
3   * File Name          : stmicroelectronics_x-cube-mems1_6_0_1.c
4   * Description        : This file provides code for the configuration
5   *                      of the STMicroelectronics.X-CUBE-MEMS1.6.0.1 instances.
6   ****
7
8
23
24 #ifdef __cplusplus
25 | extern "C" {
26 |#endif
27
28 /* Includes -----
29 #include "app_x-cube-mems1.h"
30 #include "main.h"
31 #include <stdio.h>
--
```

# Src/app\_x-cube-mems1.c -

## Part 2

```
32
33 void MX_MEMS_Init(void)
34 {
35     /* USER CODE BEGIN SV */
36
37     /* USER CODE END SV */
38
39     /* USER CODE BEGIN MEMS_Library_Init_PreTreatment */
40
41     /* USER CODE END MEMS_Library_Init_PreTreatment */
42
43     /* Initialize the peripherals and the MEMS components */
44
45     /* USER CODE BEGIN SV */
46
47     /* USER CODE END SV */                                Close (⌘W)
48
49     /* USER CODE BEGIN MEMS_Library_Init_PostTreatment */
50
51     /* USER CODE END MEMS_Library_Init_PostTreatment */
52 }
```

# Src/app\_x-cube-mems1.c -

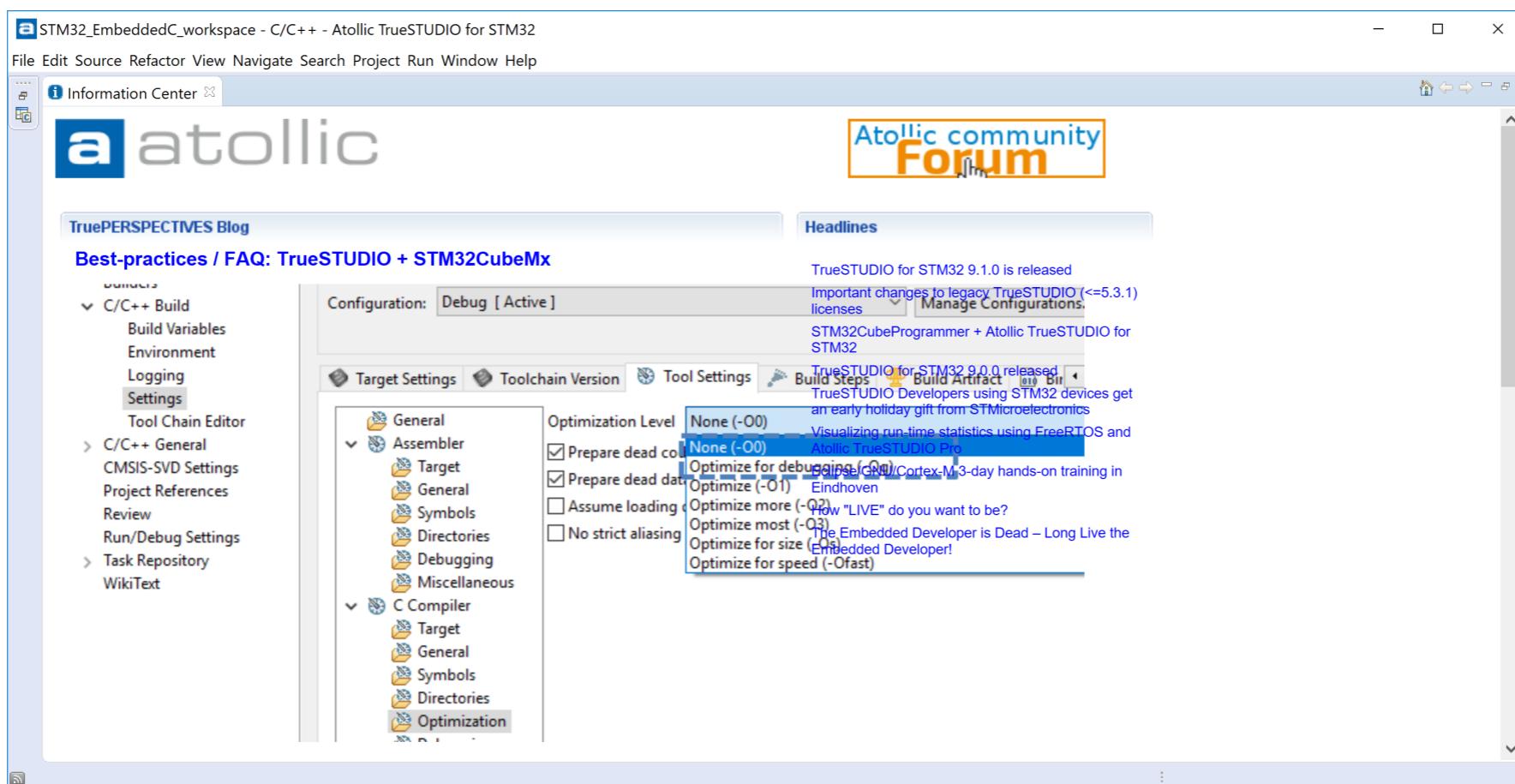
## Part 3

```
53  /*
54   * LM background task
55   */
56 void MX_MEMS_Process(void)
57 {
58     /* USER CODE BEGIN MEMS_Library_Process */
59
60     /* USER CODE END MEMS_Library_Process */
61 }
```

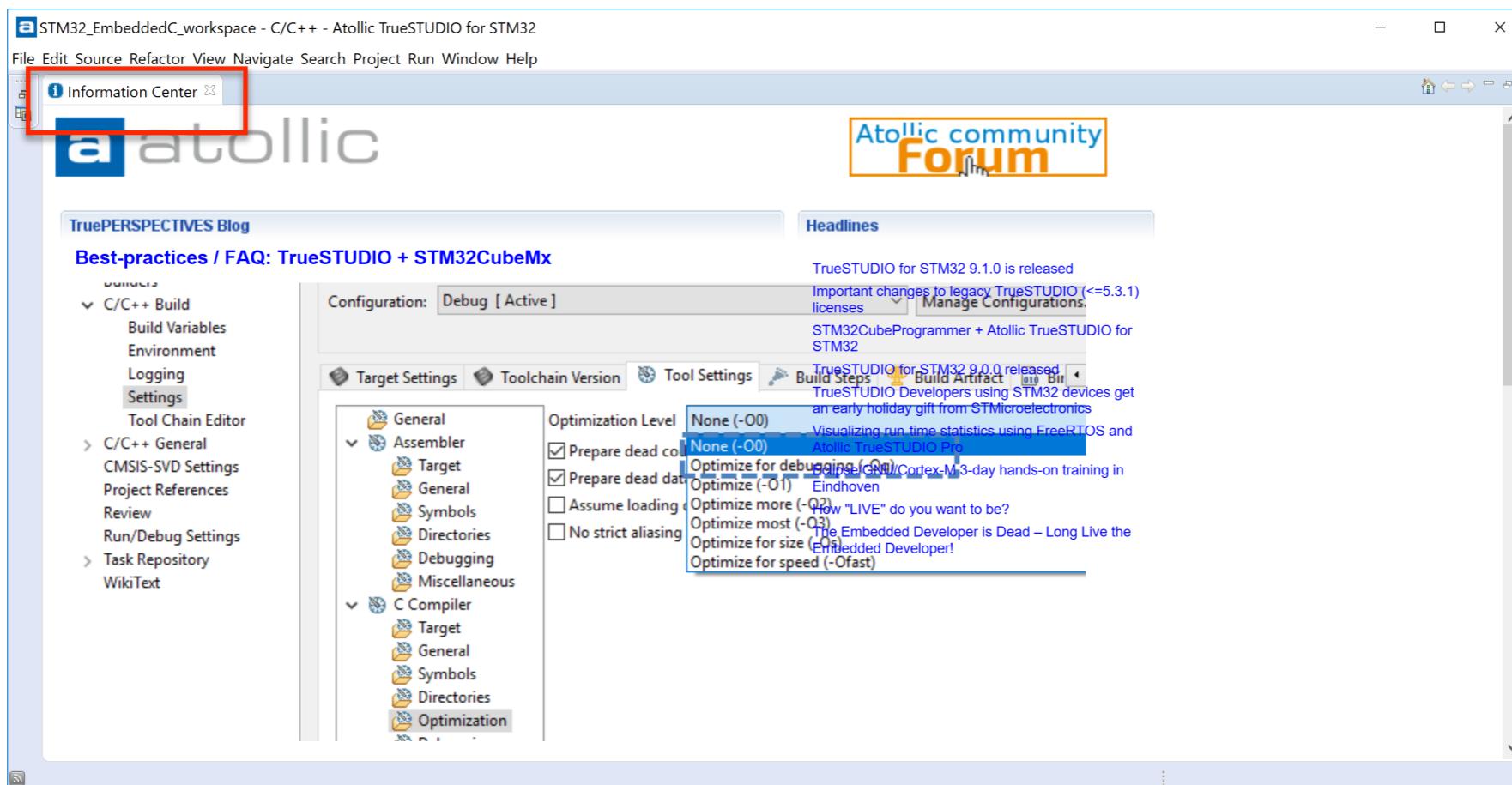
--

Use TrueStudio  
To Toggle LED2 On/Off  
Using BSP

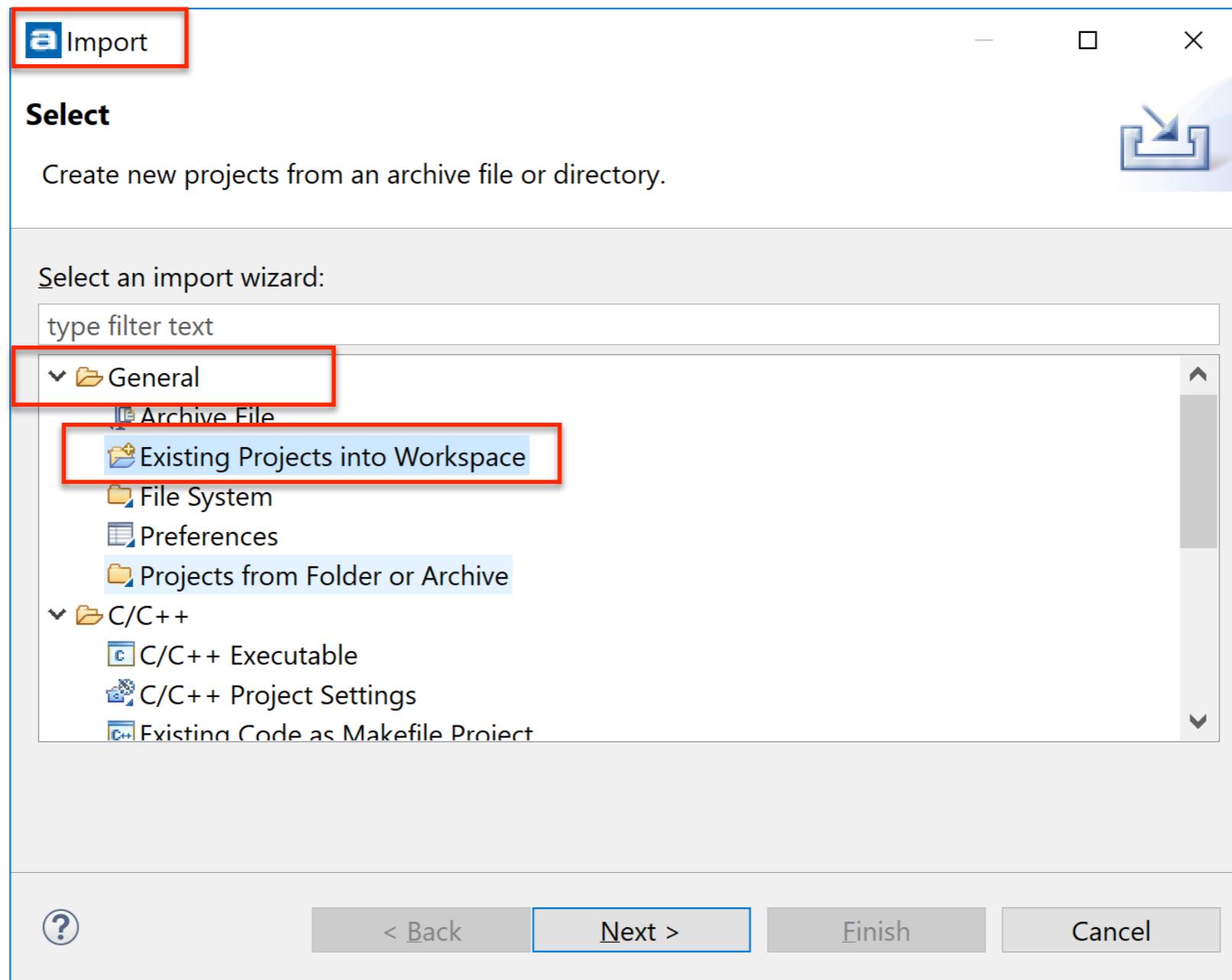
# Step: Startup TrueStudio



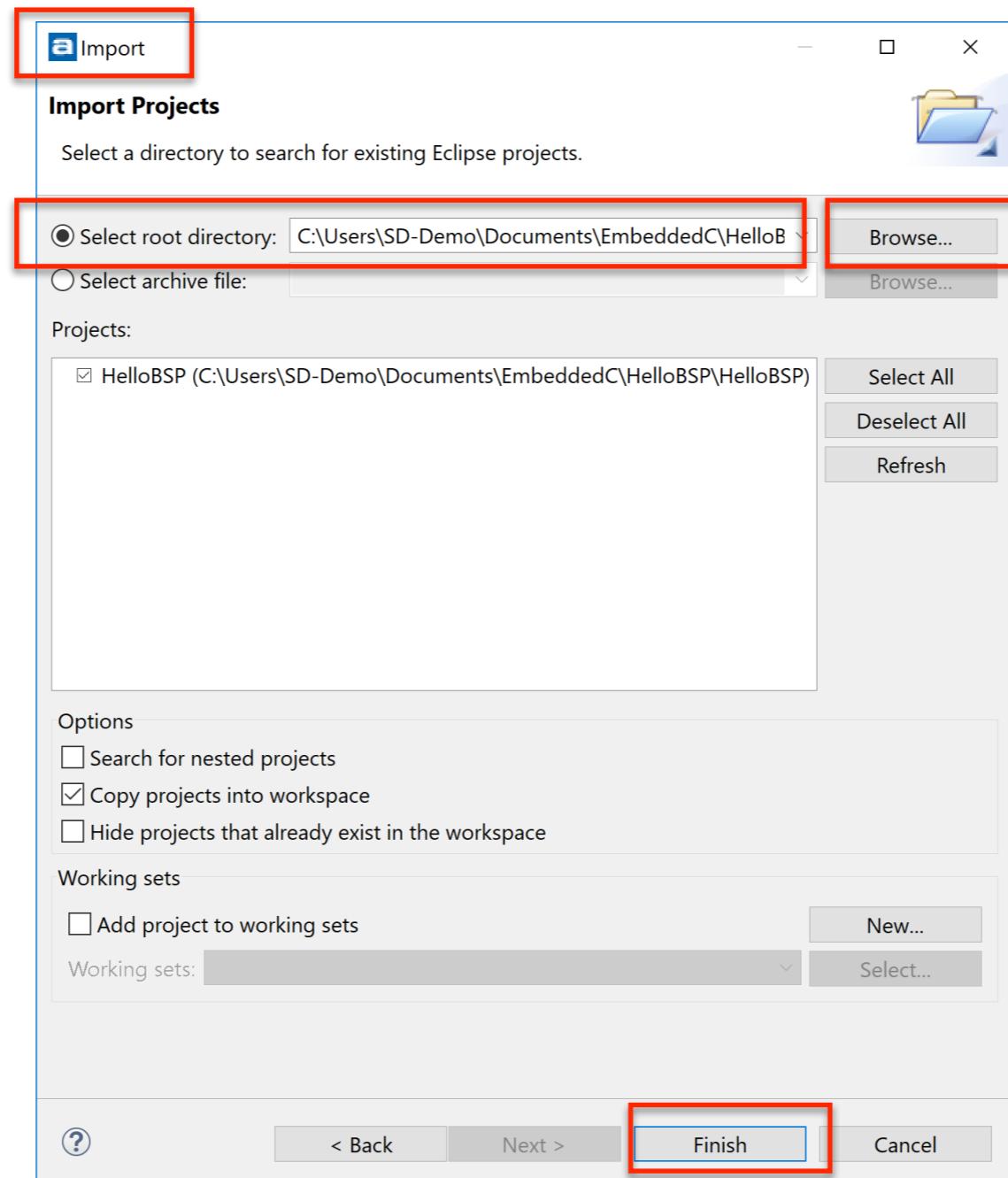
# Step: Click to Close “Information Center”



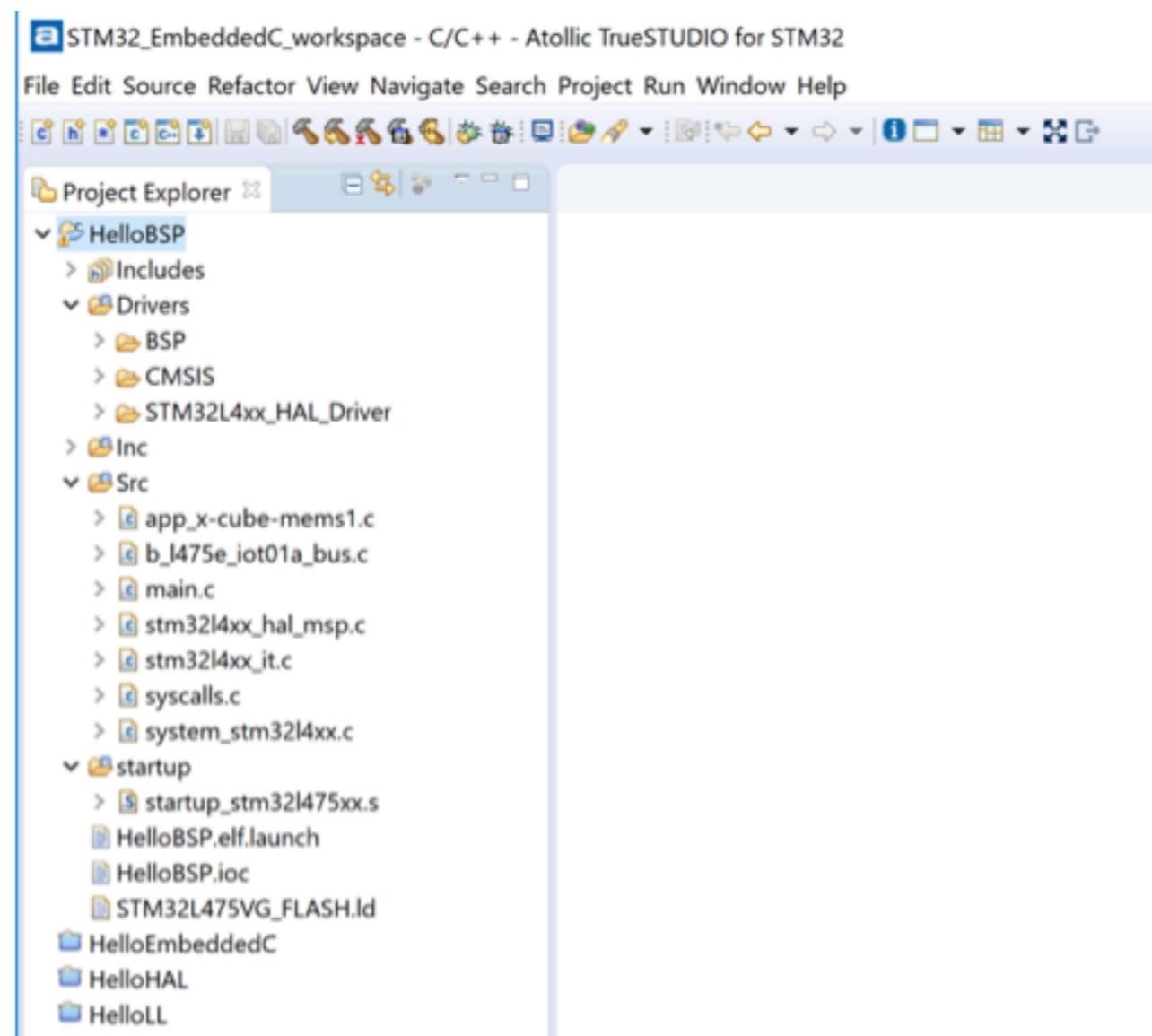
# Step: File, Import, General, Existing Projects into Workspace



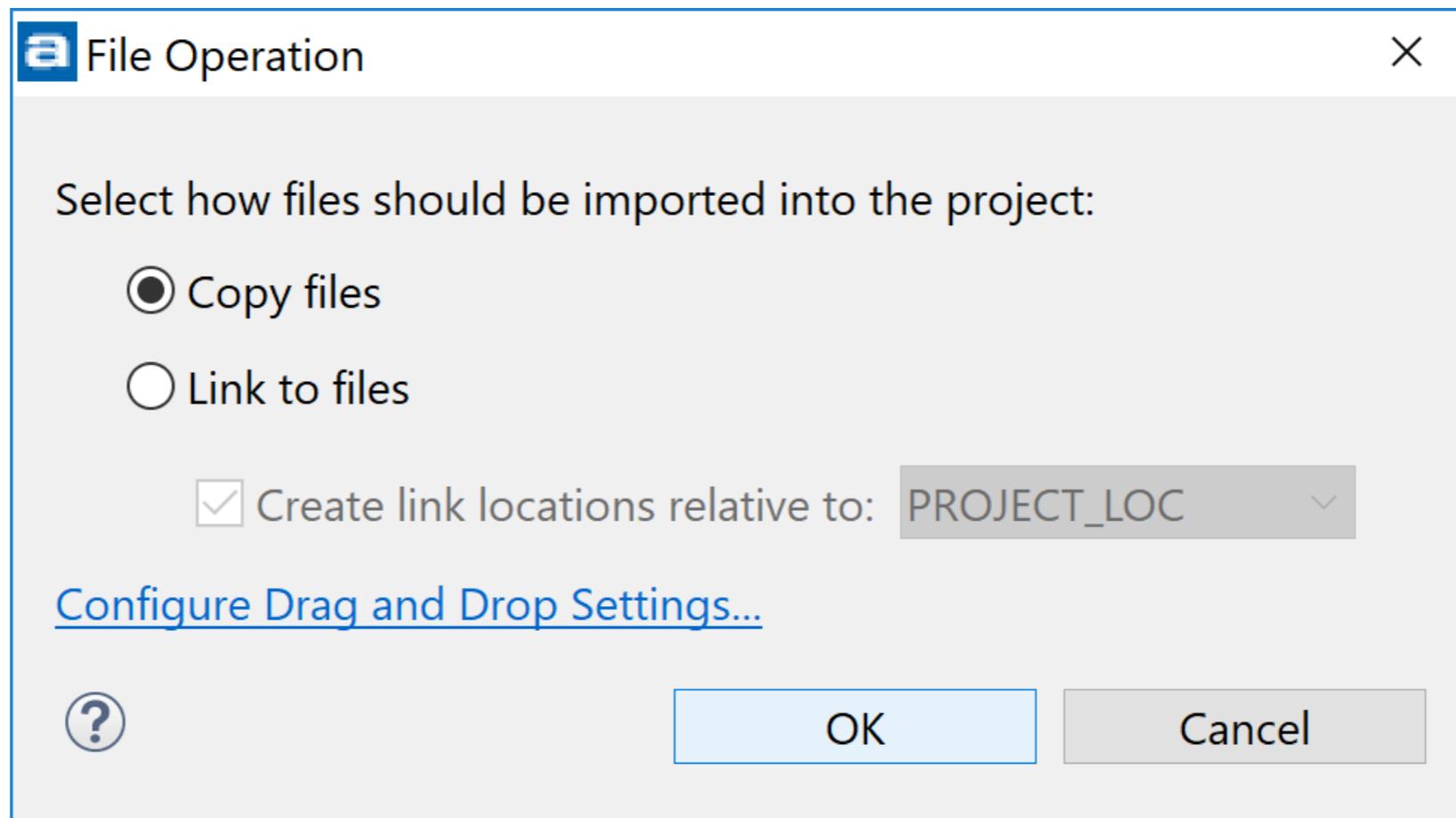
# Step: Select Root Directory (Browse to directory as needed)



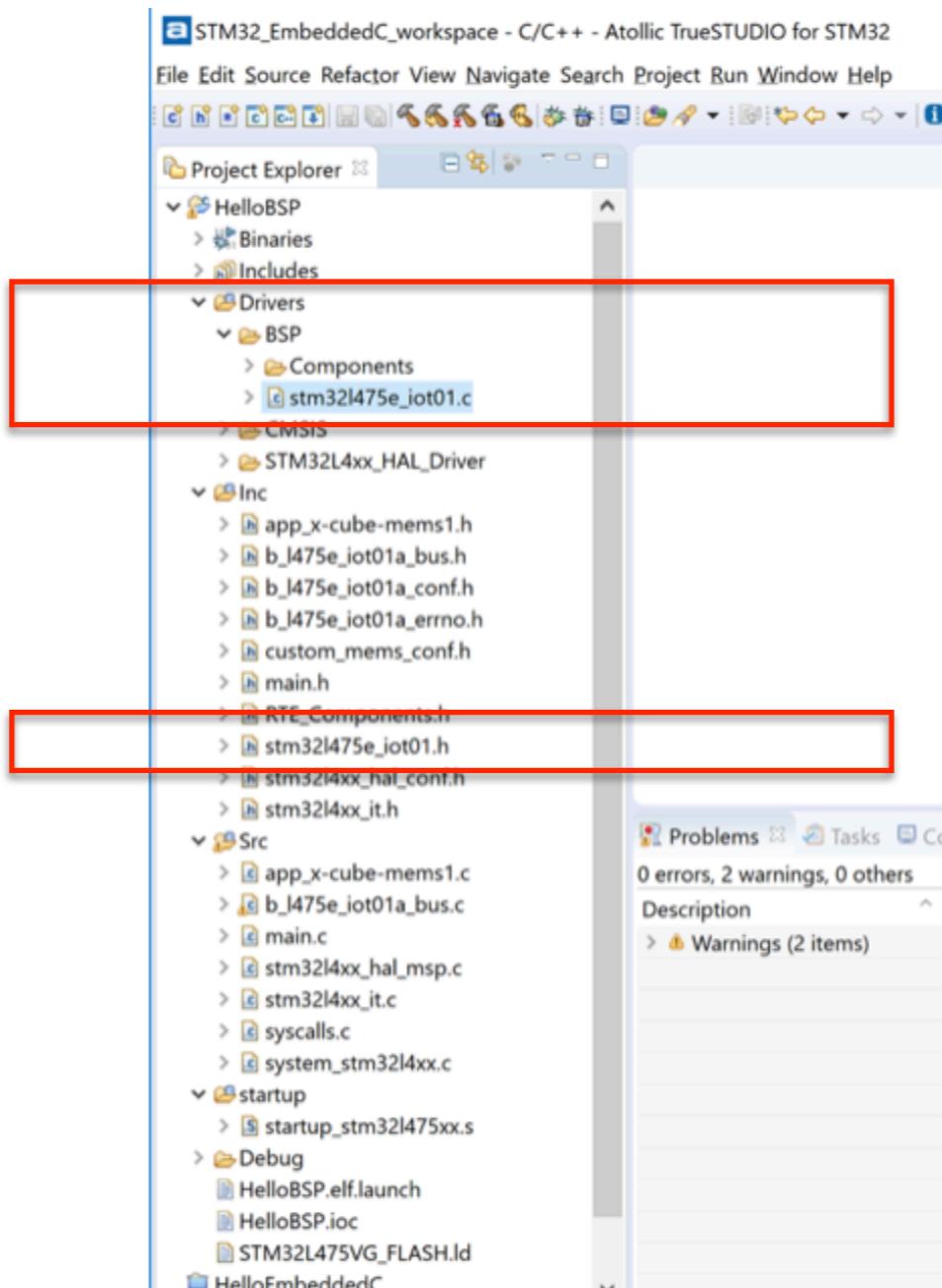
# Step: Observe Results



# Step: Drag/Drop Files into Project (See next slide)



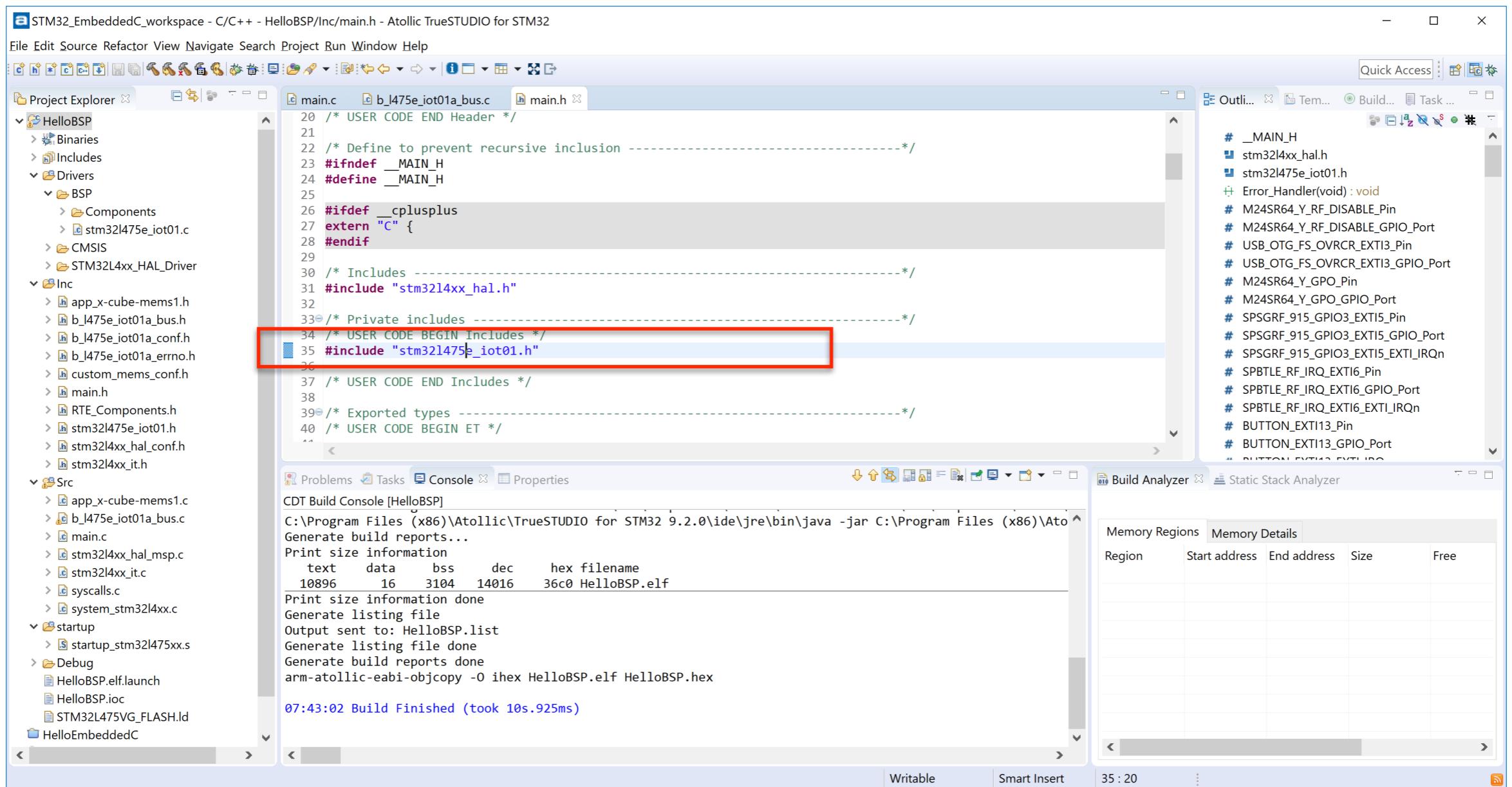
# Step: Drag these files into Project



**stm32l475e\_iot01.c**

**stm32l475e\_iot01.h**

# Step: Edit main.h



The screenshot shows the Atollic TrueSTUDIO for STM32 interface. The main window displays the file `main.h` with the following code:

```
20 /* USER CODE END Header */
21
22 /* Define to prevent recursive inclusion -----*/
23 #ifndef __MAIN_H
24 #define __MAIN_H
25
26 #ifdef __cplusplus
27 extern "C" {
28 #endif
29
30 /* Includes -----*/
31 #include "stm32l4xx_hal.h"
32
33 /* Private includes -----*/
34 /* USER CODE BEGIN Includes */
35 #include "stm32l475e_iot01.h"
36
37 /* USER CODE END Includes */
38
39 /* Exported types -----*/
40 /* USER CODE BEGIN ET */
```

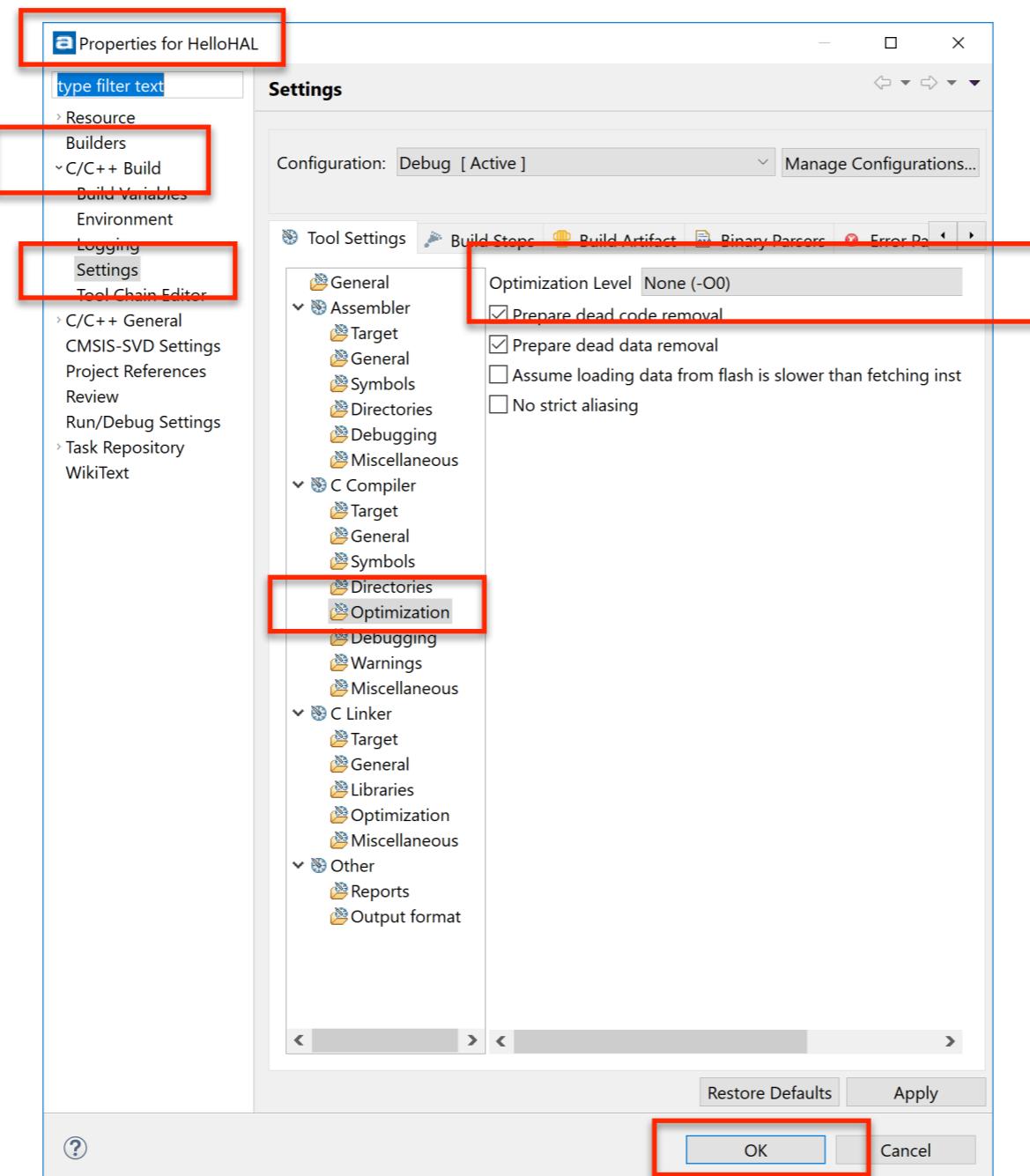
A red box highlights the line `#include "stm32l475e_iot01.h"`. The left sidebar shows the project structure under the `HelloBSP` folder, including `Binaries`, `Includes`, `Drivers/BSP` (with `Components`, `stm32l475e_iot01.c`, `CMSIS`, `STM32L4xx_HAL_Driver`), `Inc` (with various header files like `app_x-cube-mems1.h`, `b_l475e_iot01a_bus.h`, etc.), and `Src` (with source files like `main.c`, `stm32l4xx_hal_msp.c`, etc.). The right sidebar contains the `Build Analyzer` and `Static Stack Analyzer` panes. The bottom pane shows the build log:

```
CDT Build Console [HelloBSP]
C:\Program Files (x86)\Atollic\TrueSTUDIO for STM32 9.2.0\ide\jre\bin\java -jar C:\Program Files (x86)\Atollic\TrueSTUDIO for STM32 9.2.0\ide\lib\cdt\cdt.jar
Generate build reports...
Print size information
    text      data      bss      dec      hex filename
  10896        16     3104   14016  36c0 HelloBSP.elf
Print size information done
Generate listing file
Output sent to: HelloBSP.list
Generate listing file done
Generate build reports done
arm-atollic-eabi-objcopy -O ihex HelloBSP.elf HelloBSP.hex
07:43:02 Build Finished (took 10s.925ms)
```

# Step: Edit main.c

```
115  /* USER CODE END 2 */
116
117  /* Infinite loop */
118  /* USER CODE BEGIN WHILE */
119
120  BSP_LED_Init(LED2);
121
122  while (1)
123  {
124      /* USER CODE END WHILE */
125
126      MX_MEMS_Process();
127      /* USER CODE BEGIN 3 */
128
129      BSP_LED_Toggle(LED2);
130      HAL_Delay(1000);
131
132  }
133  /* USER CODE END 3 */
134 }
135 /**
136 * @brief System Clock Configuration
137 * @retval None
138 */
139
```

# Step: Project, Properties, C/C++ Build, Settings, C Compiler, Optimization, None



NOTE: This is optional step, only needed if you want to use step-by step debugging

# Step: Build, Observe Errors

The screenshot shows the Atollic TrueSTUDIO for STM32 interface. The main window displays the code editor for `main.c`. The code includes initialization of an LED and a main loop. The build console window shows several errors related to undeclared identifiers for I2C pins in the `b_1475e_iot01a_bus.c` file. The errors are:

```
..\Src\b_1475e_iot01a_bus.c: In function 'I2C2_MspInit':
..\Src\b_1475e_iot01a_bus.c:544:27: error: 'INTERNAL_I2C2_SCL_Pin' undeclared (first use in this function)
    GPIO_InitStruct.Pin = INTERNAL_I2C2_SCL_Pin|INTERNAL_I2C2_SDA_Pin;
                           ^~~~~~
..\Src\b_1475e_iot01a_bus.c:544:27: note: each undeclared identifier is reported only once for each function
..\Src\b_1475e_iot01a_bus.c:544:49: error: 'INTERNAL_I2C2_SDA_Pin' undeclared (first use in this function)
    GPIO_InitStruct.Pin = INTERNAL_I2C2_SCL_Pin|INTERNAL_I2C2_SDA_Pin;
                           ^~~~~~
..\Src\b_1475e_iot01a_bus.c: In function 'I2C2_MspDeInit':
..\Src\b_1475e_iot01a_bus.c:570:28: error: 'INTERNAL_I2C2_SDA_Pin' undeclared (first use in this function)
```

The right side of the interface shows the outline view with various system and application header files listed.

# Step: “#if 0” to remove errors. We are not using this code - Part 1

The screenshot shows the Atollic TrueSTUDIO for STM32 IDE interface. The main window displays the code editor for the file `b_l475e_iot01a_bus.c`. The code contains two sections of code enclosed in `#if 0` directives, which are highlighted with red boxes. The first section (lines 539-558) configures GPIO pins PB10 and PB11 for I2C2\_SCL and I2C2\_SDA respectively, setting them to GPIO\_MODE\_AF\_OD, GPIO\_PULLUP, and GPIO\_SPEED\_FREQ VERY HIGH. The second section (line 559) initializes the GPIOB peripheral. The code editor also shows the `I2C2_MspInit` function definition at the bottom.

```
531 static void I2C2_MspInit(I2C_HandleTypeDef* i2cHandle)
532 {
533     GPIO_InitTypeDef GPIO_InitStruct;
534     /* USER CODE BEGIN I2C2_MspInit 0 */
535
536     /* USER CODE END I2C2_MspInit 0 */
537
538 #if 0
539     HAL_RCC_ClockConfig();
540     /* I2C2 GPIO Configuration
541      PB10      -----> I2C2_SCL
542      PB11      -----> I2C2_SDA
543      */
544     GPIO_InitStruct.Pin = INTERNAL_I2C2_SCL_Pin|INTERNAL_I2C2_SDA_Pin;
545     GPIO_InitStruct.Mode = GPIO_MODE_AF_OD;
546     GPIO_InitStruct.Pull = GPIO_PULLUP;
547     GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
548     GPIO_InitStruct.Alternate = GPIO_AF4_I2C2;
549     HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
550
551     /* Peripheral clock enable */
552     HAL_RCC_I2C2_CLK_ENABLE();
553     /* USER CODE BEGIN I2C2_MspInit 1 */
554
555     /* USER CODE END I2C2_MspInit 1 */
556 #endif
557
558
559
560 }
561
562 static void I2C2_MspDeInit(I2C_HandleTypeDef* i2cHandle)
```

The build console at the bottom left shows a compilation error:

```
..\Src\b_l475e_iot01a_bus.c:544:27: error: 'INTERNAL_I2C2_SCL_Pin' undeclared (first use in this function)
    GPIO_InitStruct.Pin = INTERNAL_I2C2_SCL_Pin|INTERNAL_I2C2_SDA_Pin;
```

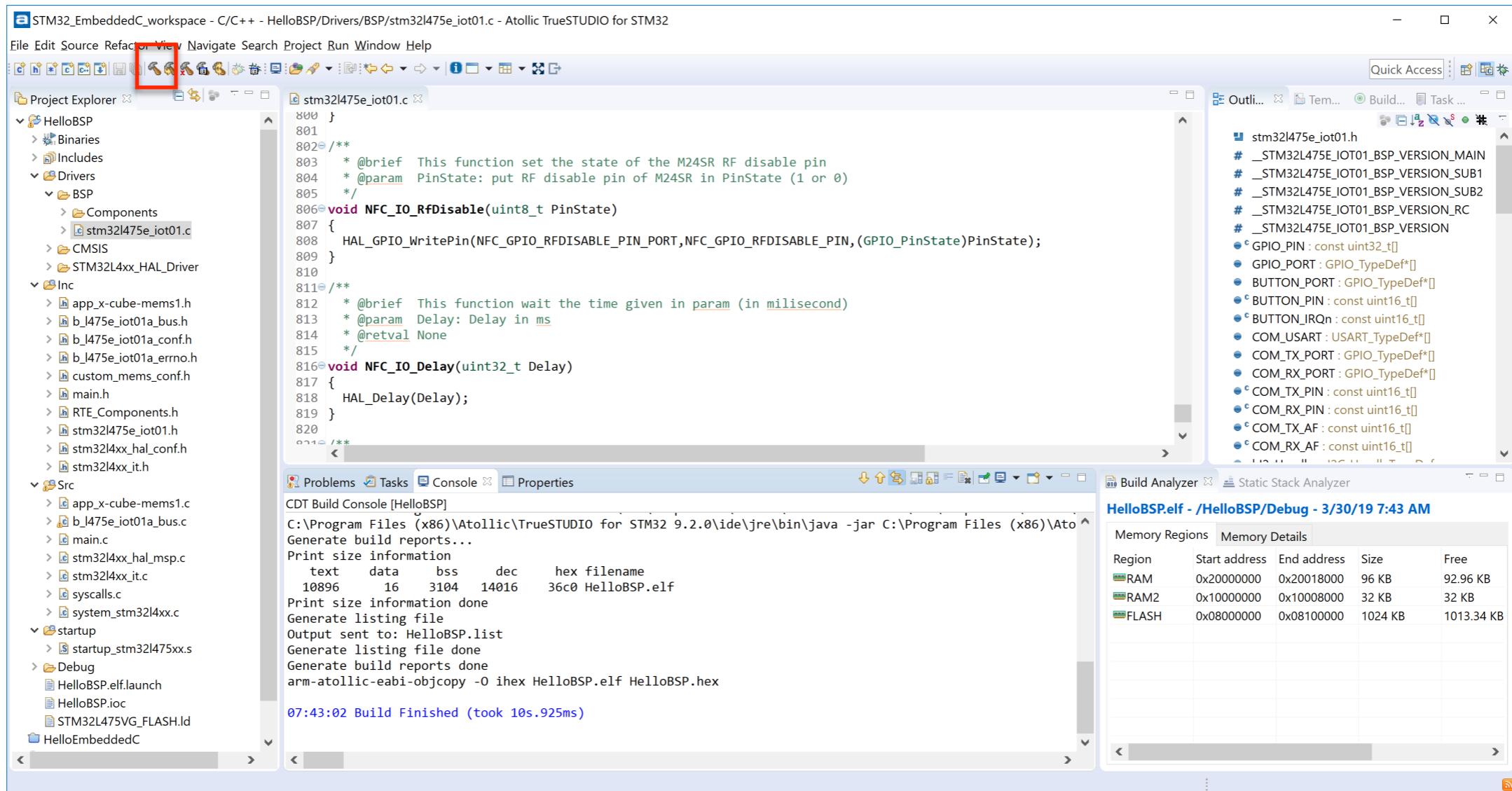
The right side of the interface includes the Project Explorer, Outliner, Build Analyzer, and Static Stack Analyzer panes.

Step: “#if 0” to remove errors. We are not using this code - Part 2

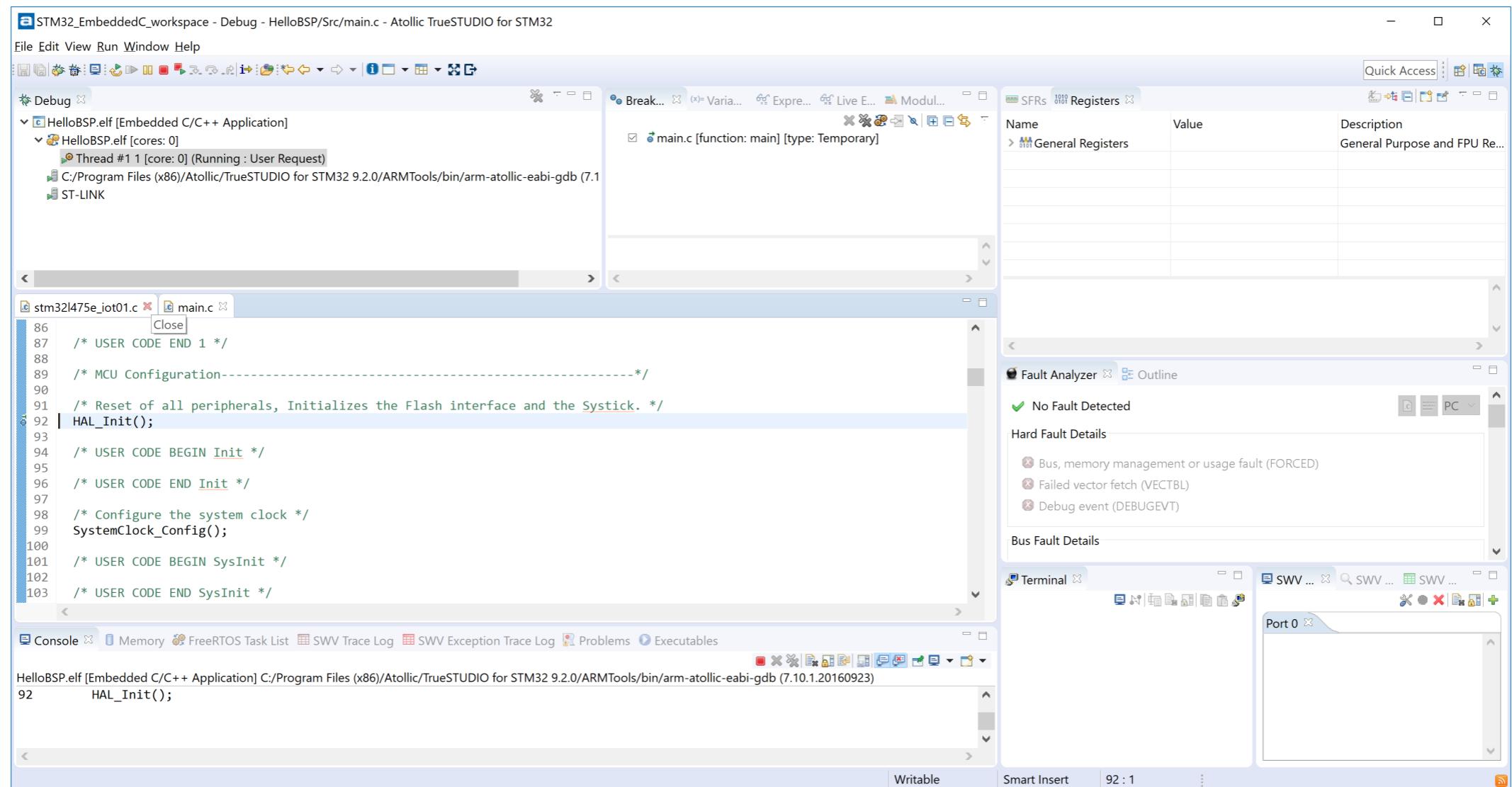
The screenshot shows the Atollic TrueSTUDIO for STM32 interface. The main window displays the code for `b_1475e_iot01a.bus.c`. Two specific sections of the code are highlighted with red boxes:

```
554     __HAL_RCC_I2C2_CLK_ENABLE();
555     /* USER CODE BEGIN I2C2_MspInit 1 */
556
557     /* USER CODE END I2C2_MspInit 1 */
558 #endif
559 }
560 }
561
562 static void I2C2_MspDeInit(I2C_HandleTypeDef* i2cHandle)
563 {
564     /* USER CODE BEGIN I2C2_MspDeInit 0 */
565
566 #if 0
567     /* USER CODE END I2C2_MspDeInit 0 */
568     /* Peripheral clock disable */
569     __HAL_RCC_I2C2_CLK_DISABLE();
570
571     /**I2C2 GPIO Configuration
572     PB10      -----> I2C2_SCL
573     PB11      -----> I2C2_SDA
574     */
575     HAL_GPIO_DeInit(GPIOB, INTERNAL_I2C2_SCL_Pin|INTERNAL_I2C2_SDA_Pin);
576
577     /* USER CODE BEGIN I2C2_MspDeInit 1 */
578
579 #endif
580     /* USER CODE END I2C2_MspDeInit 1 */
581 }
582 }
583
584 #if (USE_CUBEMX_BSP_V2 == 1)
585 /**
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
70
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
72
730
731
732
733
734
735
736
737
738
739
73
740
741
742
743
744
745
746
747
748
749
74
750
751
752
753
754
755
756
757
758
759
75
760
761
762
763
764
765
766
767
768
769
76
770
771
772
773
774
775
776
777
778
779
77
780
781
782
783
784
785
786
787
788
789
78
790
791
792
793
794
795
796
797
798
799
79
7
800
801
802
803
804
805
806
807
808
809
80
810
811
812
813
814
815
816
817
818
819
81
820
821
822
823
824
825
826
827
828
829
82
830
831
832
833
834
835
836
837
838
839
83
840
841
842
843
844
845
846
847
848
849
84
850
851
852
853
854
855
856
857
858
859
85
860
861
862
863
864
865
866
867
868
869
86
870
871
872
873
874
875
876
877
878
879
87
880
881
882
883
884
885
886
887
888
889
88
890
891
892
893
894
895
896
897
898
89
8
900
901
902
903
904
905
906
907
908
909
90
910
911
912
913
914
915
916
917
918
919
91
920
921
922
923
924
925
926
927
928
929
92
930
931
932
933
934
935
936
937
938
939
93
940
941
942
943
944
945
946
947
948
949
94
950
951
952
953
954
955
956
957
958
959
95
960
961
962
963
964
965
966
967
968
969
96
970
971
972
973
974
975
976
977
978
979
97
980
981
982
983
984
985
986
987
988
989
98
990
991
992
993
994
995
996
997
998
999
99
9
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
100
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
101
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
102
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
103
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
104
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
105
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
106
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
107
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
108
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
109
10
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
110
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
111
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
112
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
113
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
114
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
115
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
116
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
117
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
118
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
119
11
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
120
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
121
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
122
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
123
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
124
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
125
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
126
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
127
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
128
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
129
12
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
130
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
131
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
132
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
133
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
134
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
135
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
136
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
137
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
138
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
139
13
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
140
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
141
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
142
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
143
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
144
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
145
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
146
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
147
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
148
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
149
14
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
150
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
151
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
152
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
153
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
154
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
155
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
156
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
157
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
158
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
159
15
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
160
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
161
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
162
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
163
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
164
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
165
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
166
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
167
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
168
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
169
16
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
170
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
171
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
172
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
173
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
174
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
175
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
176
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
177
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
178
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
179
17
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
180
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
181
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
182
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
183
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
184
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
185
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
186
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
187
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
188
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
189
18
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
190
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
191
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
192
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
193
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
194
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
195
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
196
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
197
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
198
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
199
19
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
200
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
201
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
202
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
203
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
204
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
205
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
206
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
207
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
208
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
209
20
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
210
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
211
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
212
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
213
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
214
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
215
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
216
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
217
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
218
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
219
21
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
220
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
221
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
222
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
223
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
224
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
225
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
226
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
227
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
228
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
229
22
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
230
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
231
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
232
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
233
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
234
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
235
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
236
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
237
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
238
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
239
23
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
240
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
241
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
242
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
243
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
244
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
245
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
246
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
247
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
248
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
249
24
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
250
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
251
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
252
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
253
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
254
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
255
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
256
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
257
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
258
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
259
25
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
260
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
261
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
262
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
263
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
264
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
265
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
266
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
267
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
268
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
269
26
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
270
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
271
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
272
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
273
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
274
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
275
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
276
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
277
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
278
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
279
27
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
280
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
281
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
282
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
283
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
284
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
285
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
286
2870
2871
2872
2873
2874
2875
2876
28
```

# Step: Build - Success!

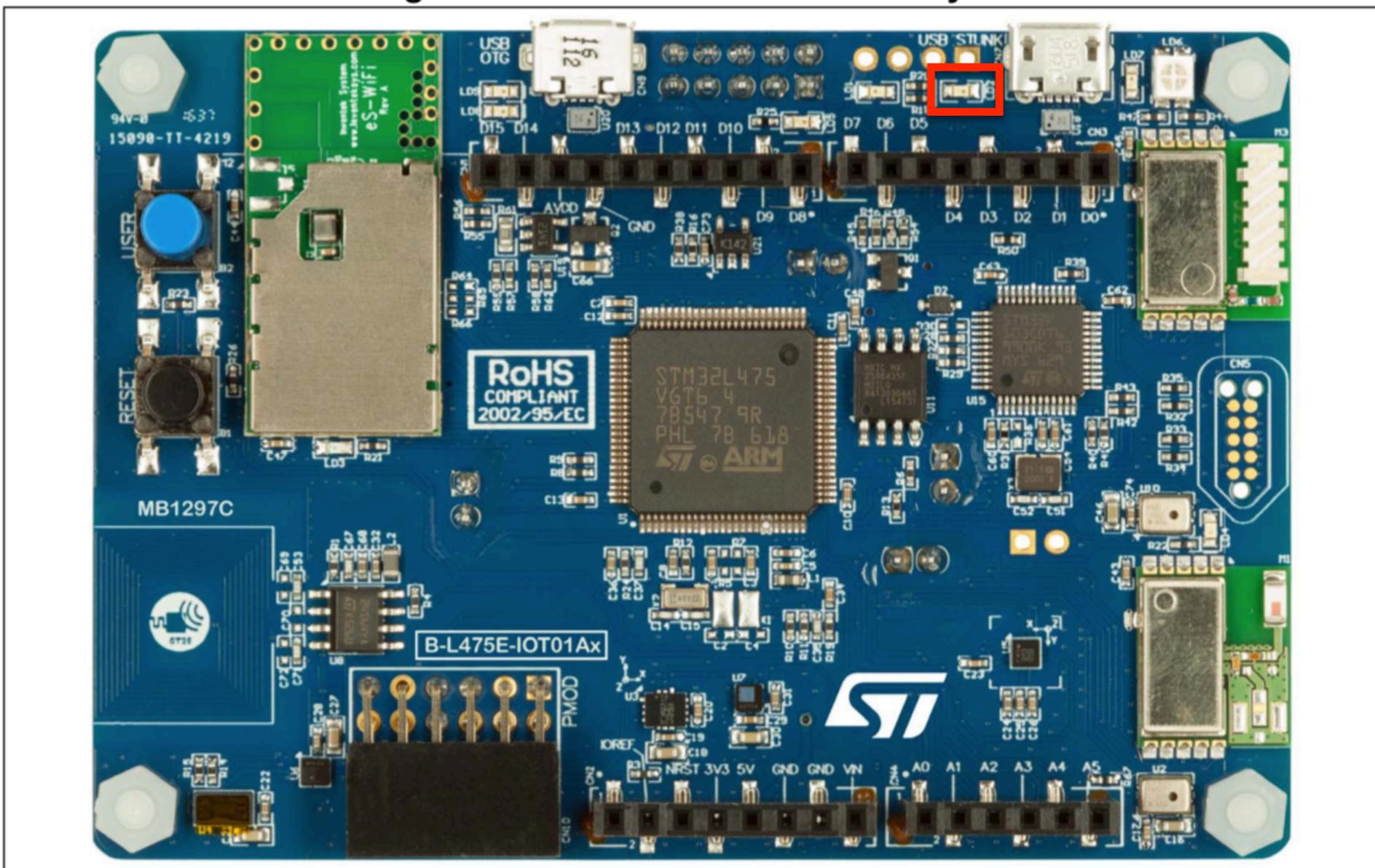


# Step: Run Project in Debug Mode



Step: Look for LED blinking on STM Discovery Board at 1 second on/off rate

Figure 1. B-L475E-IOT01A Discovery kit



# Summary

- Introduction to BSP and Embedded C
- STM32CubeMX and BSP Code Generation
- Tour of BSP
- TrueStudio and BSP
  - Blinking LED