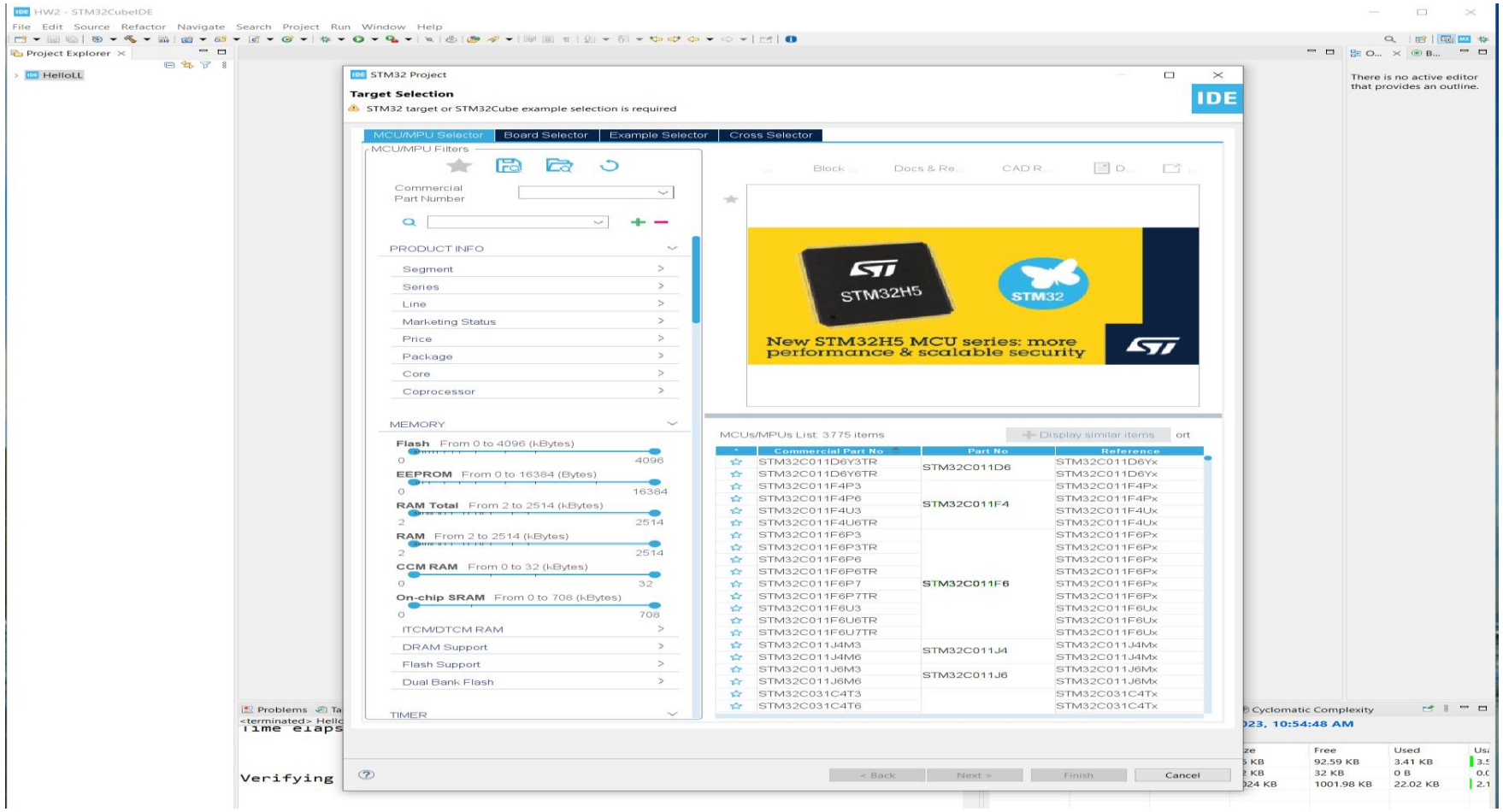
UCSD Embedded C Assignment 2

By

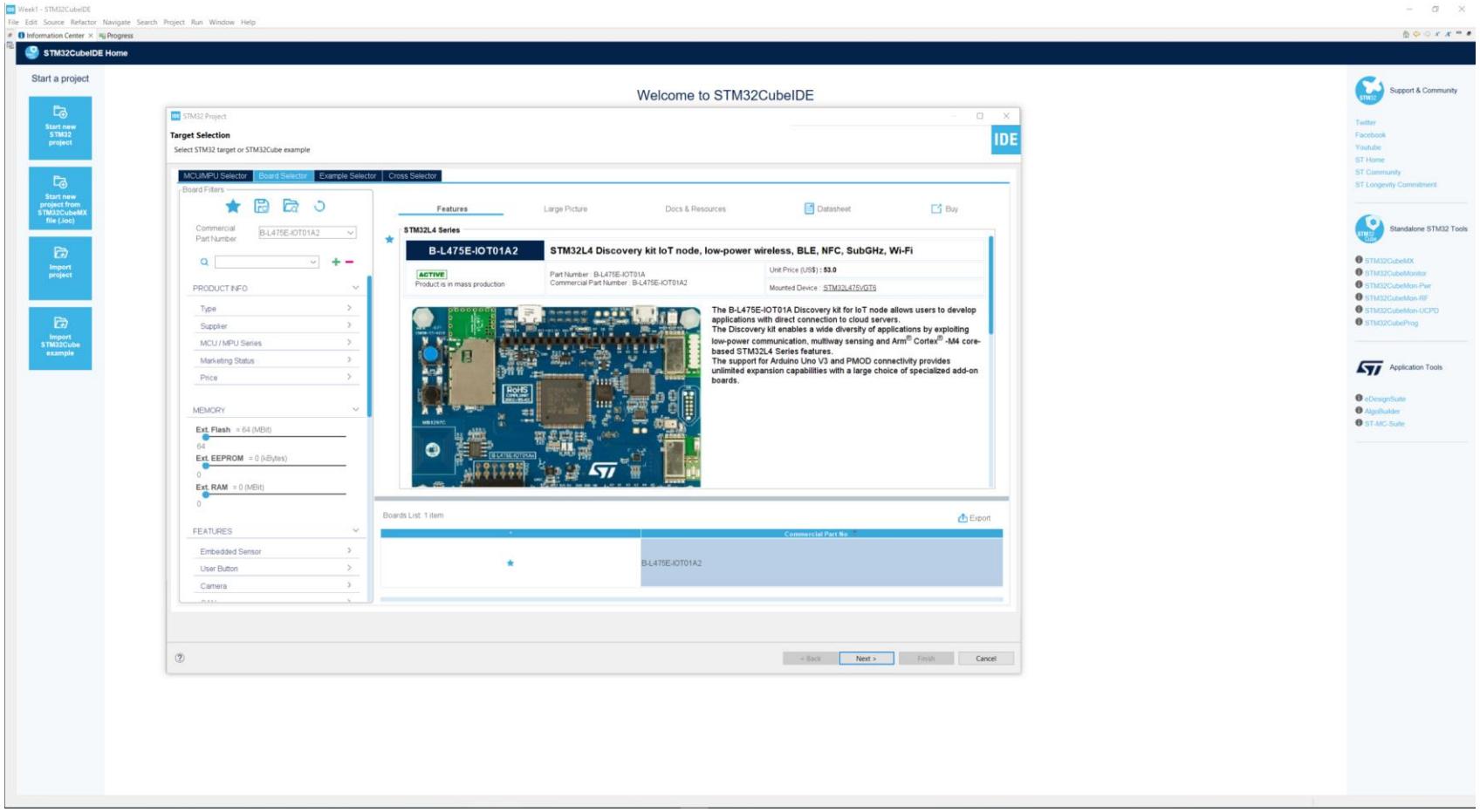
Hsuankai Chang

hsuankac@umich.edu

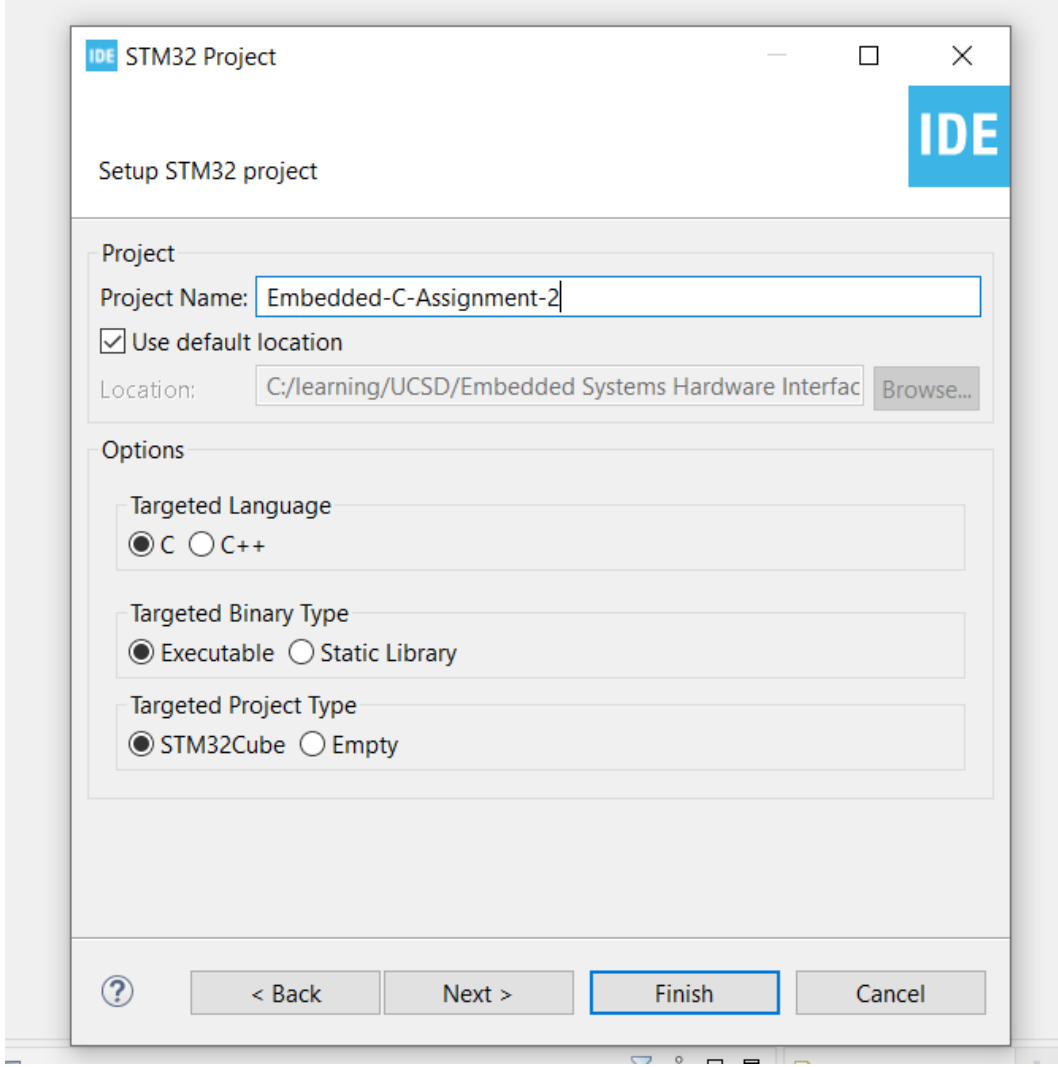
Step 1. Startup STM32CubeIDE and create new STM32 project



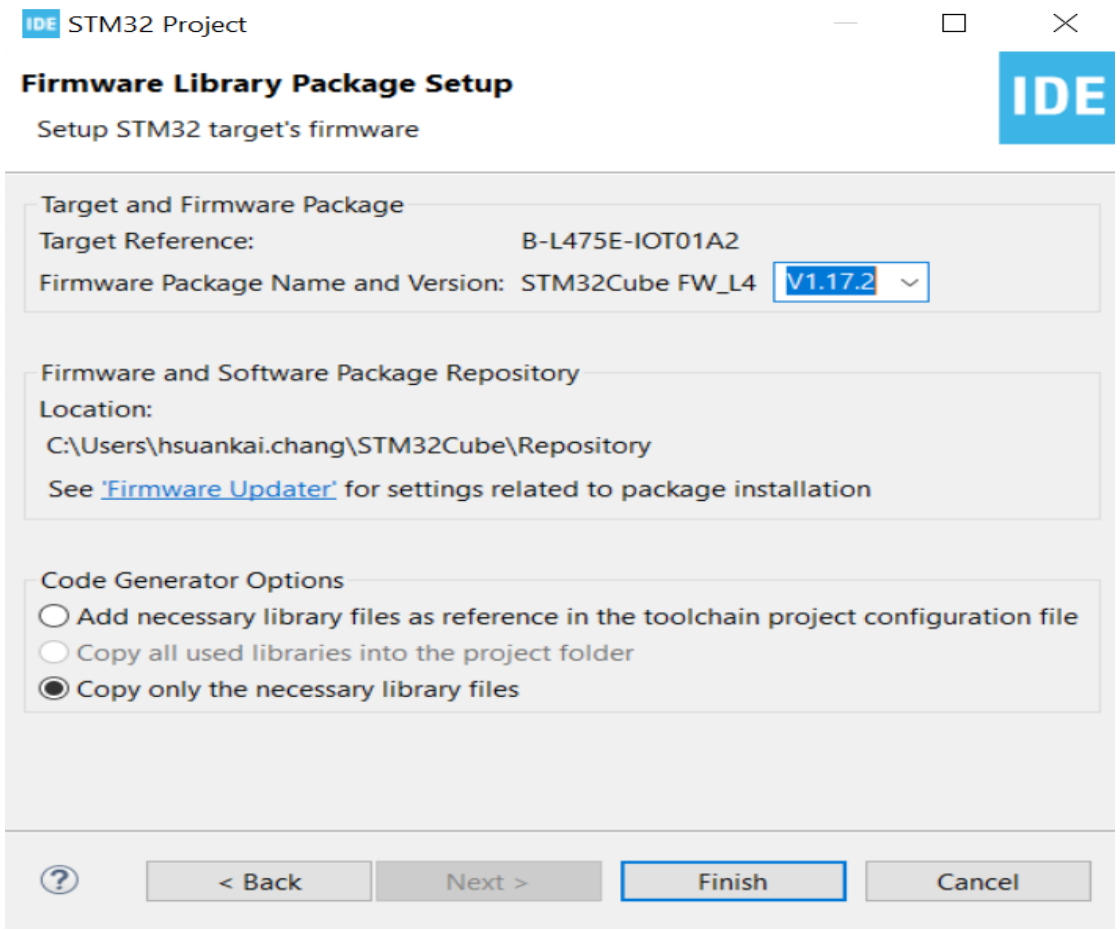
Step 2. Access board selector and type in the board you use, click Next



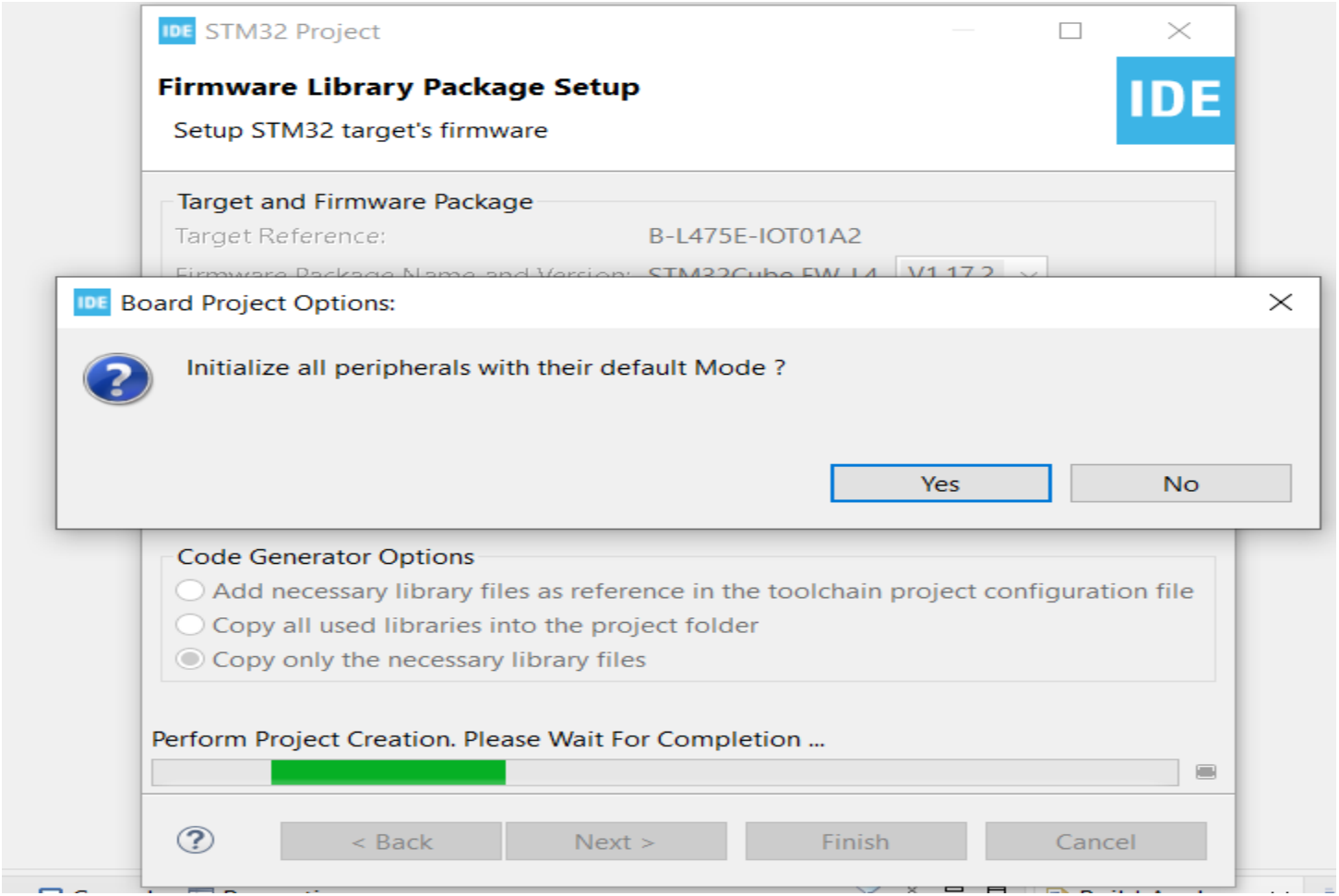
Step 3. Enter the project name then click Next



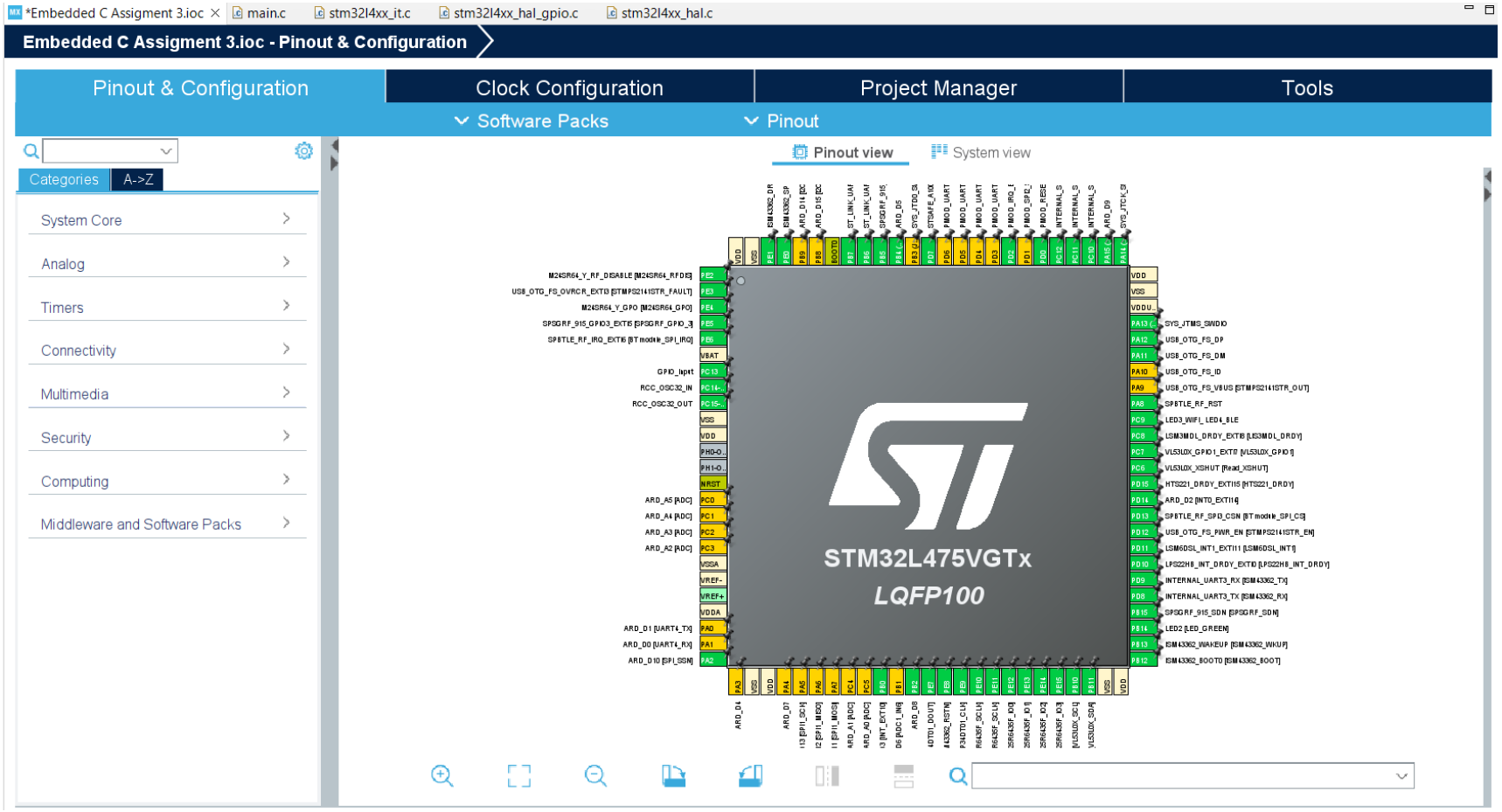
Step 4. See the firmware package name and version



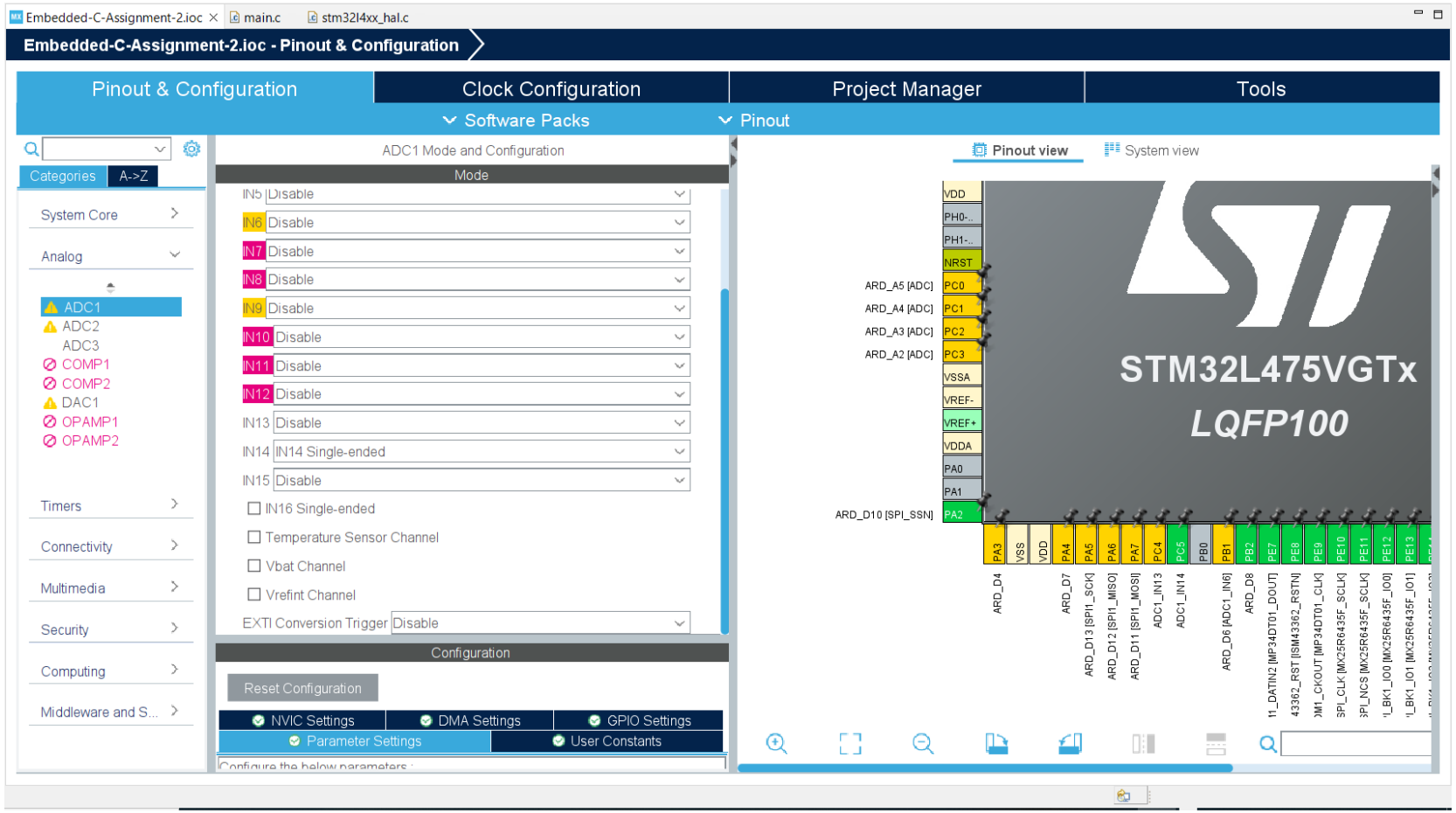
Step 5. Click yes to initialize all peripherals to default



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



Step 7. User Story 1: Connect ARD-A0 to a 1.5VDC battery and read using ADC Polling mode. Show the output to the console

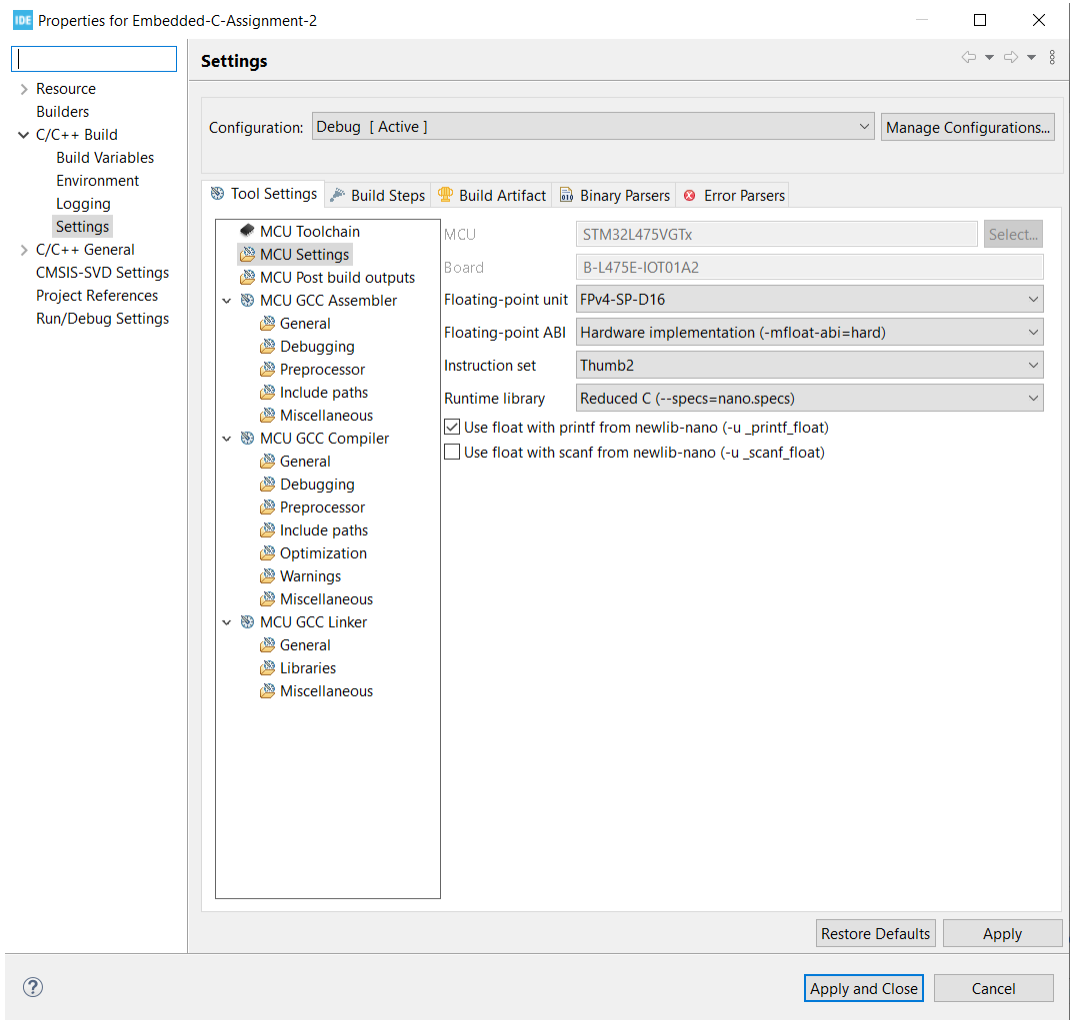


Step 8. Write the polling mode code in main.c file

Embedded-C-Assignment-2.ioc | main.c | stm32l4xx_hal.c

```c
  MX_ADC1_Init();
  /* USER CODE BEGIN 2 */
  HAL_ADCEx_Calibration_Start(&hadc1, ADC_SINGLE_ENDED);
  /* USER CODE END 2 */

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

    /* USER CODE BEGIN 3 */
    HAL_GPIO_TogglePin(LED3_WIFI__LED4_BLE_GPIO_Port, LED3_WIFI__LED4_BLE_Pin);
    HAL_Delay(1000);

    // ADC start conversion
    HAL_ADC_Start(&hadc1);

    // Poll for results, timeout is 10 us
    HAL_ADC_PollForConversion(&hadc1, 10);

    uint16_t value = HAL_ADC_GetValue(&hadc1);
    float voltage = value * (3.3 / 4096);

    // Send value to console
    char buf[100];
    snprintf(buf, sizeof(buf), "ARD-A0: raw: %u, volts: %f\r\n", value, voltage);

    HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), 1000);

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

Step 9. Since we are using floating point with printf, go to build setting and enable this flag



Step 10. Build the debug the code, open tera term and we can see the output is nearly 1.5v on the console

Embedded-C-Assignment-2.ioc | main.c | stm32l4xx_hal.c | startup_stm32l475vgtx.s

```
  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

  /* MCU Configuration--------------------------------------------------------*/

  /* Reset of all peripherals, Initializes the Flash int
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

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

/* Configure the peripherals common clocks */
  PeriphCommonClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_DFSDM1_Init();
  MX_I2C2_Init();
  MX_QUADSPI_Init();
  MX_SPI3_Init();
  MX_USART1_UART_Init();
```

Name | Type

COM4 - Tera Term VT

File Edit Setup Control Window Help

```
ARD-A0: raw: 1848, volts: 1.488867
ARD-A0: raw: 1844, volts: 1.485645
ARD-A0: raw: 1843, volts: 1.484839
ARD-A0: raw: 1844, volts: 1.485645
ARD-A0: raw: 1843, volts: 1.484839
```

Step 11. Repeat user story 1 without doing calibration for user story 4. Observe that without calibration the value deviate more from 1.5V

Embedded-C-Assignment-2.ioc | main.c | stm32l4xx_hal.c | startup_stm32l475vgtx.s

```c
  */
int main(void)
{
  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

  /* MCU Configuration--------------------------------

  /* Reset of all peripherals, Initializes the Flash in
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

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

/* Configure the peripherals common clocks */
  PeriphCommonClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_DFSDM1_Init();
  MX_I2C2_Init();
```

Va... | Br... | Ex... | Re..

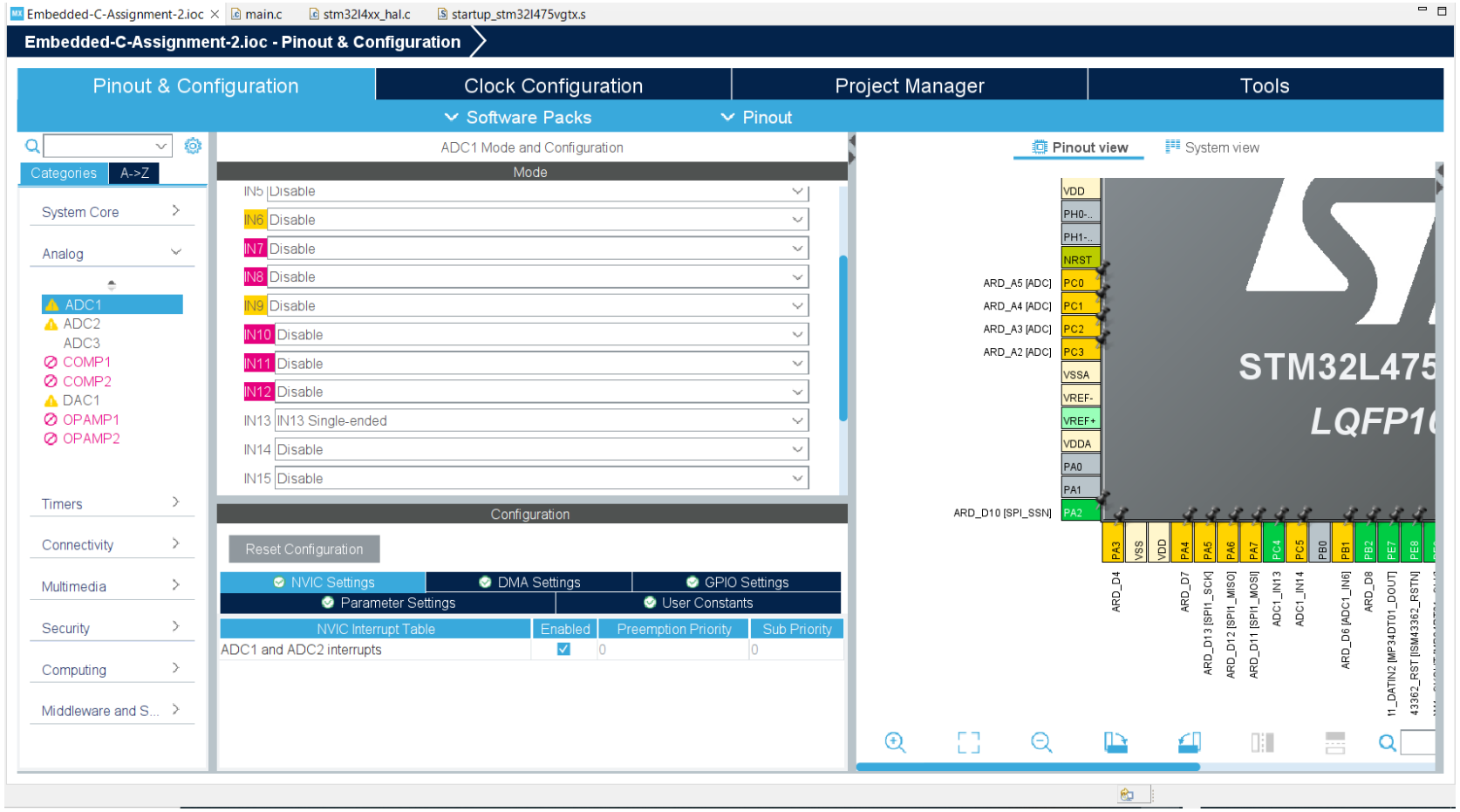
Name	Type

COM4 - Tera Term VT

File Edit Setup Control Window Help

```
ARD-A0: raw: 1794, volts: 1.445361
ARD-A0: raw: 1788, volts: 1.440527
ARD-A0: raw: 1788, volts: 1.440527
ARD-A0: raw: 1788, volts: 1.440527
ARD-A0: raw: 1787, volts: 1.439722
ARD-A0: raw: 1788, volts: 1.440527
ARD-A0: raw: 1787, volts: 1.439722
```

Step 12. User Story 2: Connect ARD-A1 to the same 1.5VDC battery and read using ADC Interrupt mode and send the output to the console



Step 13. Write the interrupt code in main.c file

Embedded-C-Assignm... | main.c | stm32l4xx_hal.c | startup_stm32l475v... | stm32l4xx_it.c | stm32l4xx_hal_adc.c | stm32l4xx_hal_adc_...

```c
  /* USER CODE END 2 */

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

    /* USER CODE BEGIN 3 */
    HAL_GPIO_TogglePin(LED3_WIFI__LED4_BLE_GPIO_Port, LED3_WIFI__LED4_BLE_Pin);
    HAL_Delay(1000);

    // ADC start conversion
//  HAL_ADC_Start(&hadc1);
    HAL_ADC_Start_IT(&hadc1);

    // Poll for results, timeout is 10 us
//  HAL_ADC_PollForConversion(&hadc1, 10);
//
//  uint16_t value = HAL_ADC_GetValue(&hadc1);
//  float voltage = value * (3.3 / 4096);
//
//  // Send value to console
//  char buf[100];
//    snprintf(buf, sizeof(buf), "ARD-A0: raw: %u, volts: %f\r\n", value, voltage);
//
//    HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), 1000);

  }
  /* USER CODE END 3 */
}

/**
```

```c
/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef *hadc)
{
    uint16_t value = HAL_ADC_GetValue(&hadc1);
    float voltage = value * (3.3 / 4096);

    // Send value to console
    char buf[100];
    snprintf(buf, sizeof(buf), "ARD-A1: raw: %u, volts: %f\r\n", value, voltage);

    HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), 1000);
}
/* USER CODE END 0 */
```

Step 14. Build and debug the code, observe the result on tera term

Embedded-C-Assignm... | main.c | stm32l4xx_hal.c | startup_stm32l475v... | stm32l4xx_it.c | stm32l4xx_hal_adc.c | stm32l4xx_hal_adc_...

```c
/* Private function prototypes -----------------------------------------------*/
void SystemClock_Config(void);
void PeriphCommonClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_DFSDM1_Init(void);
static void MX_I2C2_Init(void);
static void MX_QUADSPI_Init(void);
static void MX_SPI3_Init(void);
static void MX_USART1_UART_Init(void);
static void MX_USART3_UART_Init(void);
static void MX_USB_OTG_FS_PCD_Init(void);
static void MX_ADC1_Init(void);
/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code ---------------------------------------------------------
/* USER CODE BEGIN 0 */
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef *hadc)
{
    uint16_t value = HAL_ADC_GetValue(&hadc1);
    float voltage = value * (3.3 / 4096);

    // Send value to console
    char buf[100];
    snprintf(buf, sizeof(buf), "ARD-A1: raw: %u, volts: %f\r\n", v

    HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), 1000);
}
/* USER CODE END 0 */

/**
  * @brief  The application entry point.
```

Va... | Br... | Ex... | Re... | Li... | SF...

Name	Type	Va

COM4 - Tera Term VT

File Edit Setup Control Window Help

```
ARD-A1: raw: 1850, volts: 1.490479
ARD-A1: raw: 1849, volts: 1.489673
ARD-A1: raw: 1849, volts: 1.489673
ARD-A1: raw: 1847, volts: 1.488062
ARD-A1: raw: 1850, volts: 1.490479
ARD-A1: raw: 1848, volts: 1.488867
ARD-A1: raw: 1848, volts: 1.488867
ARD-A1: raw: 1848, volts: 1.488867
ARD-A1: raw: 1848, volts: 1.488867
ARD-A1: raw: 1847, volts: 1.488062
ARD-A1: raw: 1849, volts: 1.489673
ARD-A1: raw: 1849, volts: 1.489673
ARD-A1: raw: 1849, volts: 1.489673
ARD-A1: raw: 1848, volts: 1.488867
ARD-A1: raw: 1849, volts: 1.489673
ARD-A1: raw: 1849, volts: 1.489673
ARD-A1: raw: 1849, volts: 1.489673
ARD-A1: raw: 1849, volts: 1.489673
ARD-A1: raw: 1849, volts: 1.489673
ARD-A1: raw: 1850, volts: 1.490479
ARD-A1: raw: 1848, volts: 1.488867
ARD-A1: raw: 1849, volts: 1.489673
ARD-A1: raw: 1849, volts: 1.489673
ARD-A1: raw: 1847, volts: 1.488062
ARD-A1: raw: 1849, volts: 1.489673
ARD-A1: raw: 1849, volts: 1.489673
ARD-A1: raw: 1849, volts: 1.489673
ARD-A1: raw: 1848, volts: 1.488867
ARD-A1: raw: 1849, volts: 1.489673
ARD-A1: raw: 1851, volts: 1.491284
ARD-A1: raw: 1844, volts: 1.485645
ARD-A1: raw: 1847, volts: 1.488062
ARD-A1: raw: 1846, volts: 1.487256
ARD-A1: raw: 1845, volts: 1.486450
ARD-A1: raw: 1846, volts: 1.487256
ARD-A1: raw: 1845, volts: 1.486450
```

Step 15. Repeat user story 2, without calibration, we can see the result deviates more from 1.5V

Embedded-C-Assignm... | main.c | stm32l4xx_hal.c | startup_stm32l475v... | stm32l4xx_it.c | stm32l4xx_hal_adc.c | stm32l4xx_hal_adc_...

```c
  */
int main(void)
{
  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

  /* MCU Configuration--------------------------------------------

  /* Reset of all peripherals, Initializes the Flash interface and
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

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

/* Configure the peripherals common clocks */
  PeriphCommonClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_DFSDM1_Init();
  MX_I2C2_Init();
  MX_QUADSPI_Init();
  MX_SPI3_Init();
  MX_USART1_UART_Init();
```

COM4 - Tera Term VT

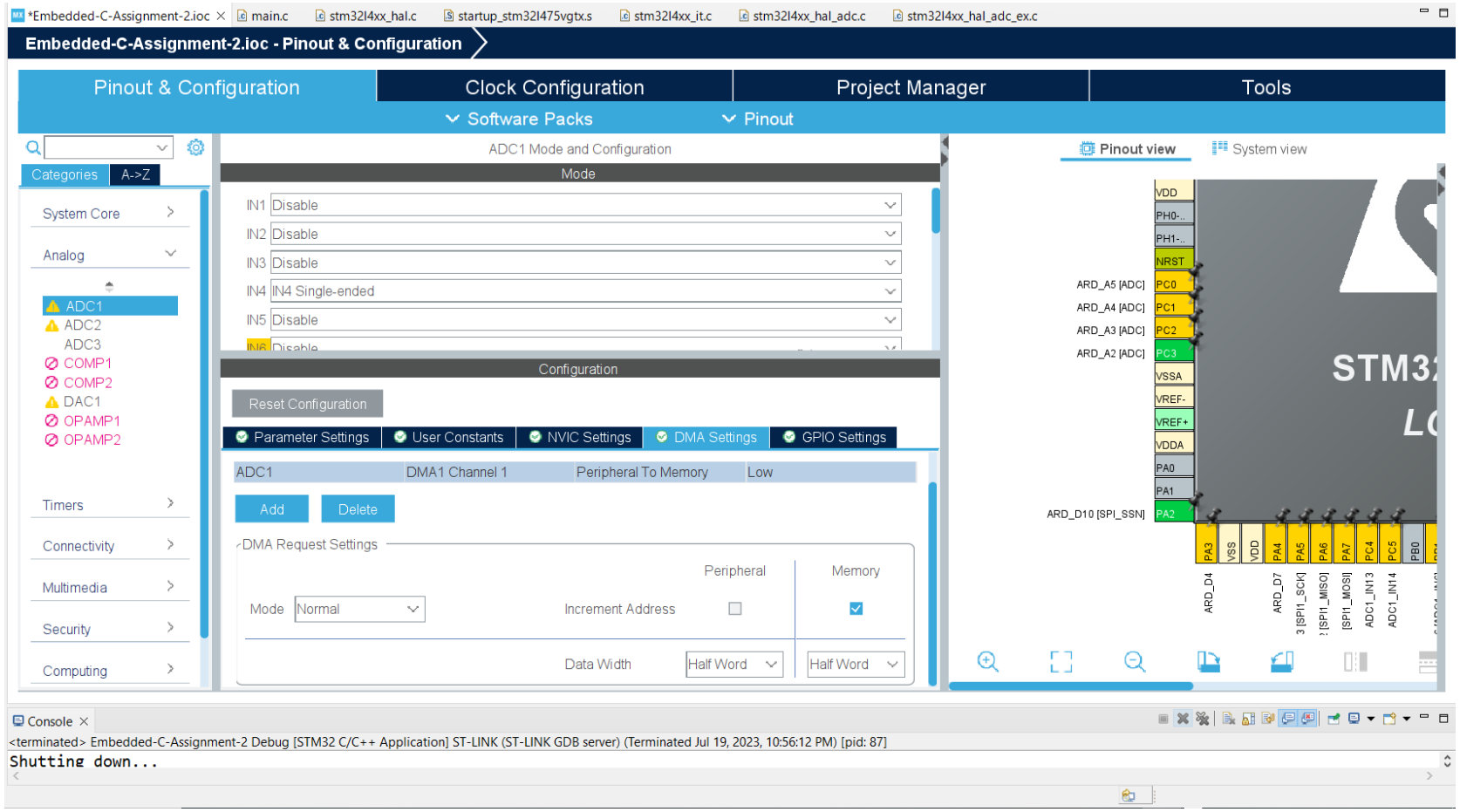
File Edit Setup Control Window Help

```
ARD-A1: raw: 1797, volts: 1.447778
ARD-A1: raw: 1793, volts: 1.444556
ARD-A1: raw: 1793, volts: 1.444556
ARD-A1: raw: 1793, volts: 1.444556
```

Name | Type | Va

Console | Problems | Executables | Debugger Console | Memory

Step 16. User Story 3: Connect ARD-A2 to the same 1.5VDC battery and read using ADC DMA Mode and send the output to the



Step 17. Write the DMA code in main.c file

Embedded-C-Assi... | main.c | stm32l4xx_hal.c | startup_stm32l... | stm32l4xx_it.c | stm32l4xx_hal_a... | stm32l4xx_hal_a... | system_stm32l4x...

```c
  MX_ADC1_Init();
  /* USER CODE BEGIN 2 */
  HAL_ADCEx_Calibration_Start(&hadc1, ADC_SINGLE_ENDED);
  /* USER CODE END 2 */

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

    /* USER CODE BEGIN 3 */
    HAL_GPIO_TogglePin(LED3_WIFI__LED4_BLE_GPIO_Port, LED3_WIFI__LED4_BLE_Pin);
    HAL_Delay(1000);

    // ADC start conversion
//  HAL_ADC_Start(&hadc1);
//  HAL_ADC_Start_IT(&hadc1);
    HAL_ADC_Start_DMA(&hadc1, (uint32_t*)&value, 1);

    // Poll for results, timeout is 10 us
//  HAL_ADC_PollForConversion(&hadc1, 10);
//
//  uint16_t value = HAL_ADC_GetValue(&hadc1);
//  float voltage = value * (3.3 / 4096);
//
//  // Send value to console
//  char buf[100];
//    snprintf(buf, sizeof(buf), "ARD-A0: raw: %u, volts: %f\r\n", value, voltage);
//
//    HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), 1000);

  }
```

```c
/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */
static uint16_t value;

void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef *hadc)
{
    float voltage = value * (3.3 / 4096);
    // Send value to console
    char buf[100];
    snprintf(buf, sizeof(buf), "ARD-A1: raw: %u, volts: %f\r\n", value, voltage);

    HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), 1000);
}
/* USER CODE END 0 */
```

Step 18, Build and debug the code, observe the result on tera term

Embedded-C-Assi... | main.c | stm32l4xx_hal.c | startup_stm32l... | stm32l4xx_it.c | stm32l4xx_hal_a... | stm32l4xx_hal_a... | system_stm32l4x...

```c
  * @retval int
  */
int main(void)
{
  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

  /* MCU Configuration--------------------------------------------

  /* Reset of all peripherals, Initializes the Flash interface and
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

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

/* Configure the peripherals common clocks */
  PeriphCommonClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_DMA_Init();
  MX_DFSDM1_Init();
  MX_I2C2_Init();
  MX_QUADSPI_Init();
```

COM4 - Tera Term VT

File Edit Setup Control Window Help

```
ARD-A1: raw: 1843, volts: 1.484839
ARD-A1: raw: 1847, volts: 1.488062
ARD-A1: raw: 1844, volts: 1.485645
ARD-A1: raw: 1844, volts: 1.485645
ARD-A1: raw: 1843, volts: 1.484839
ARD-A1: raw: 1845, volts: 1.486450
ARD-A1: raw: 1846, volts: 1.487256
ARD-A1: raw: 1839, volts: 1.481616
ARD-A1: raw: 1842, volts: 1.484033
ARD-A1: raw: 1842, volts: 1.484033
ARD-A1: raw: 1845, volts: 1.486450
ARD-A1: raw: 1844, volts: 1.485645
ARD-A1: raw: 1841, volts: 1.483227
ARD-A1: raw: 1841, volts: 1.483227
ARD-A1: raw: 1843, volts: 1.484839
ARD-A1: raw: 1844, volts: 1.485645
ARD-A1: raw: 1841, volts: 1.483227
ARD-A1: raw: 1844, volts: 1.485645
ARD-A1: raw: 1843, volts: 1.484839
ARD-A1: raw: 1845, volts: 1.486450
ARD-A1: raw: 1841, volts: 1.483227
ARD-A1: raw: 1842, volts: 1.484033
ARD-A1: raw: 1843, volts: 1.484839
ARD-A1: raw: 1840, volts: 1.482422
ARD-A1: raw: 1845, volts: 1.486450
ARD-A1: raw: 1842, volts: 1.484033
ARD-A1: raw: 1840, volts: 1.482422
ARD-A1: raw: 1841, volts: 1.483227
ARD-A1: raw: 1842, volts: 1.484033
ARD-A1: raw: 1842, volts: 1.484033
ARD-A1: raw: 1841, volts: 1.483227
ARD-A1: raw: 1849, volts: 1.489673
ARD-A1: raw: 1840, volts: 1.482422
ARD-A1: raw: 1842, volts: 1.484033
ARD-A1: raw: 1842, volts: 1.484033
ARD-A1: raw: 1842, volts: 1.484033
```

Step 19. Repeat user story 3, without calibration, we can see the result deviates more from 1.5V

Embedded-C-Assi... | main.c | stm32l4xx_hal.c | startup_stm32l... | stm32l4xx_it.c | stm32l4xx_hal_a... | stm32l4xx_hal_a... | system_stm32l4x...

```c
  * @retval int
  */
int main(void)
{
  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

  /* MCU Configuration--------------------------------------------

  /* Reset of all peripherals, Initializes the Flash interface and
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

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

/* Configure the peripherals common clocks */
  PeriphCommonClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_DMA_Init();
  MX_DFSDM1_Init();
  MX_I2C2_Init();
  MX_QUADSPI_Init();
```

COM4 - Tera Term VT

File Edit Setup Control Window Help

```
ARD-A1: raw: 1788, volts: 1.440527
ARD-A1: raw: 1783, volts: 1.436499
ARD-A1: raw: 1782, volts: 1.435693
ARD-A1: raw: 1779, volts: 1.433276
ARD-A1: raw: 1783, volts: 1.436499
ARD-A1: raw: 1780, volts: 1.434082
```

Console | Problems | Executables | Debugger Console | Memory

Embedded-C-Assignment-2 Debug [STM32 C/C++ Application] [pid: 154]

Step 20. User Store 5: Only for ADC Polling Mode, when it comes time to read, instead of reading a single time, read the ADC 12 times "back-to-back", dropping the smallest value and also the largest value

```c
  HAL_ADCEx_Calibration_Start(&hadc1, ADC_SINGLE_ENDED);
  /* USER CODE END 2 */

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

    /* USER CODE BEGIN 3 */
    HAL_GPIO_TogglePin(LED3_WIFI__LED4_BLE_GPIO_Port, LED3_WIFI__LED4_BLE_Pin);
    HAL_Delay(1000);

    uint16_t value[12];
    uint16_t value_sum = 0;

    // Poll 12 times
    for(int i = 0; i < 12; i++)
    {
        // ADC start conversion
        HAL_ADC_Start(&hadc1);
        // Poll for results, timeout is 10 us
        HAL_ADC_PollForConversion(&hadc1, 10);
        value[i] = HAL_ADC_GetValue(&hadc1);
        value_sum += value[i];
    }

    // Sum of value array minus the max and min value
    float value_avg = (float)(value_sum - find_max(value, 12) - find_min(value, 12)) / 10;

    float voltage = value_avg * (3.3 / 4096);

    // Send value to console
    char buf[100];
    snprintf(buf, sizeof(buf), "ARD-A0: raw: %f, volts: %f\r\n", value_avg, voltage);

    HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), 1000);

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

```c
/* Private user code ---------------------------------
/* USER CODE BEGIN 0 */
uint16_t find_max(uint16_t arr[], uint8_t size)
{
    int max = arr[0];
    for(int i = 0; i < size; i++){
        if(arr[i] > max){
            max = arr[i];
        }
    }
    return max;
}

uint16_t find_min(uint16_t arr[], uint8_t size)
{
    int min = arr[0];
    for(int i = 0; i < size; i++){
        if(arr[i] < min){
            min = arr[i];
        }
    }
    return min;
}

/* USER CODE END 0 */
```

Step 21. User Store 6: Compare and contrast your output from User Story 1. I found out that the averaging method has more precise value compares to User Story 1, and the value does not jump around much.

Embedded-C-A... | main.c | stm32l4xx_hal.c | startup_stm... | stm32l4xx_ha... | stm32l4xx_ha... | system_stm3... | sysmem.c

```c
MX_USB_OTG_FS_PCD_Init();
MX_ADC1_Init();
/* USER CODE BEGIN 2 */
HAL_ADCEx_Calibration_Start(&hadc1, ADC_SINGLE_ENDED);
/* USER CODE END 2 */

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

  /* USER CODE BEGIN 3 */
  HAL_GPIO_TogglePin(LED3_WIFI__LED4_BLE_GPIO_Port, LED3_WIFI__LED4_BLE_Pin);
  HAL_Delay(1000);

  uint16_t value[12];
  uint16_t value_sum = 0;

  // Poll 12 times
  for(int i = 0; i < 12; i++)
  {
      // ADC start conversion
      HAL_ADC_Start(&hadc1);
      // Poll for results, timeout is 10 us
      HAL_ADC_PollForConversion(&hadc1, 10);
      value[i] = HAL_ADC_GetValue(&hadc1);
      value_sum += value[i];
  }

  // Sum of value array minus the max and min value
  float value_avg = (float)(value_sum - find_max(value, 12) - find_min(value, 12)) /

  float voltage = value_avg * (3.3 / 4096);

  // Send value to console
  char buf[100];
  snprintf(buf, sizeof(buf), "ARD-A0: raw: %f, volts: %f\r\n", value_avg, voltage);
```

Vari... | Brea... | Expr... | Regi... | Live...

Name	Type	Value

COM4 - Tera Term VT

File Edit Setup Control Window Help

```
ARD-A0: raw: 1853.599976, volts: 1.493379
ARD-A0: raw: 1853.599976, volts: 1.493379
ARD-A0: raw: 1853.099976, volts: 1.492976
ARD-A0: raw: 1853.000000, volts: 1.492895
ARD-A0: raw: 1853.099976, volts: 1.492976
ARD-A0: raw: 1852.500000, volts: 1.492493
ARD-A0: raw: 1853.000000, volts: 1.492895
ARD-A0: raw: 1851.300049, volts: 1.491526
ARD-A0: raw: 1851.800049, volts: 1.491929
ARD-A0: raw: 1851.800049, volts: 1.491929
ARD-A0: raw: 1852.199951, volts: 1.492251
ARD-A0: raw: 1851.500000, volts: 1.491687
ARD-A0: raw: 1852.099976, volts: 1.492170
ARD-A0: raw: 1850.400024, volts: 1.490801
ARD-A0: raw: 1849.800049, volts: 1.490317
ARD-A0: raw: 1850.199951, volts: 1.490640
ARD-A0: raw: 1850.599976, volts: 1.490962
ARD-A0: raw: 1850.199951, volts: 1.490640
ARD-A0: raw: 1850.400024, volts: 1.490801
ARD-A0: raw: 1850.599976, volts: 1.490962
ARD-A0: raw: 1850.900024, volts: 1.491204
ARD-A0: raw: 1849.900024, volts: 1.490398
ARD-A0: raw: 1849.900024, volts: 1.490398
ARD-A0: raw: 1850.800049, volts: 1.491123
ARD-A0: raw: 1850.000000, volts: 1.490479
ARD-A0: raw: 1850.099976, volts: 1.490559
ARD-A0: raw: 1850.099976, volts: 1.490559
ARD-A0: raw: 1849.900024, volts: 1.490398
ARD-A0: raw: 1849.699951, volts: 1.490237
ARD-A0: raw: 1849.800049, volts: 1.490317
ARD-A0: raw: 1850.199951, volts: 1.490640
ARD-A0: raw: 1850.199951, volts: 1.490640
ARD-A0: raw: 1850.000000, volts: 1.490479
ARD-A0: raw: 1849.099976, volts: 1.489753
ARD-A0: raw: 1849.099976, volts: 1.489753
ARD-A0: raw: 1849.500000, volts: 1.490076
```

Appendix. Hardware connections