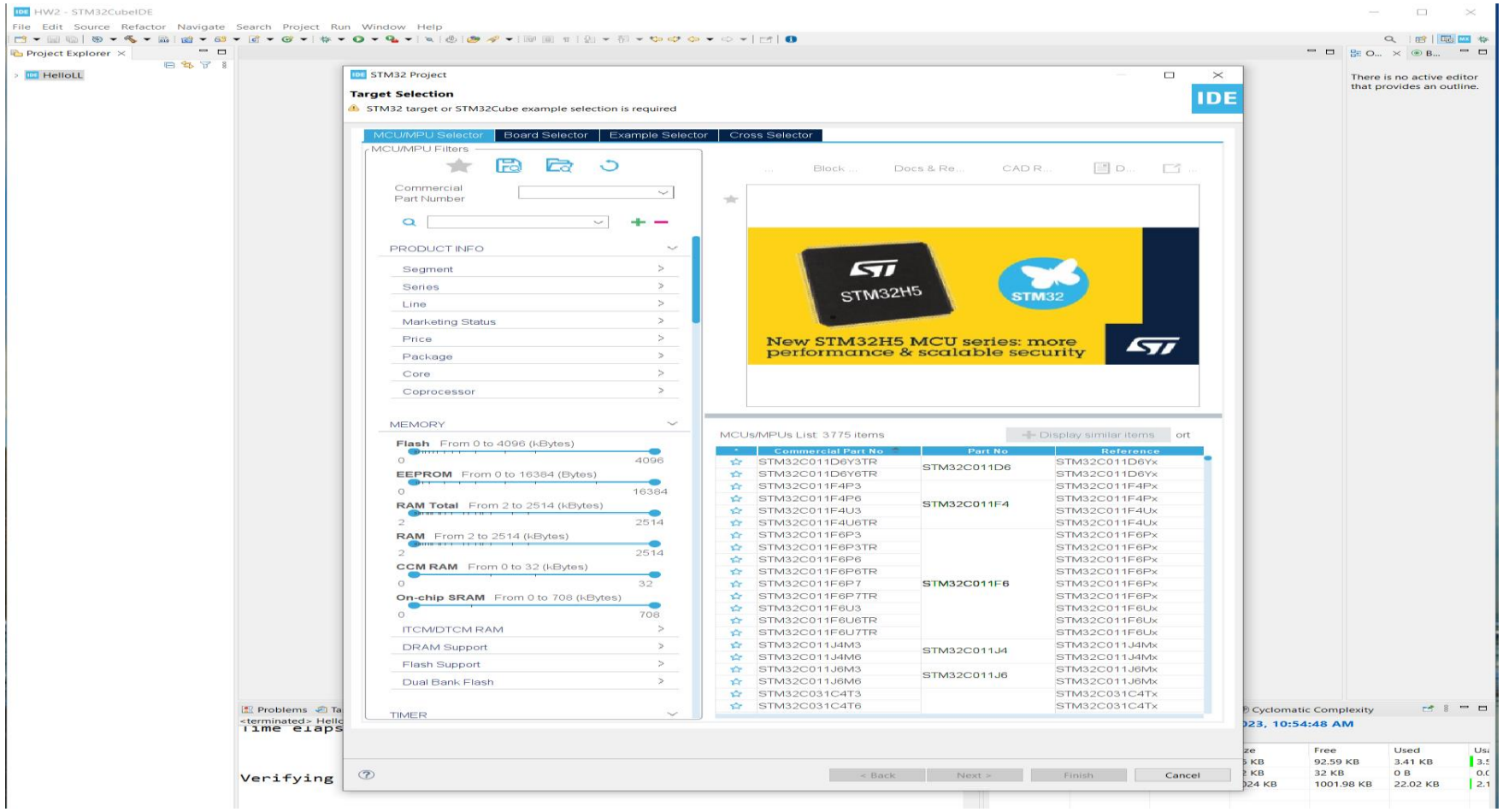
UCSD Embedded C Assignment 6

By

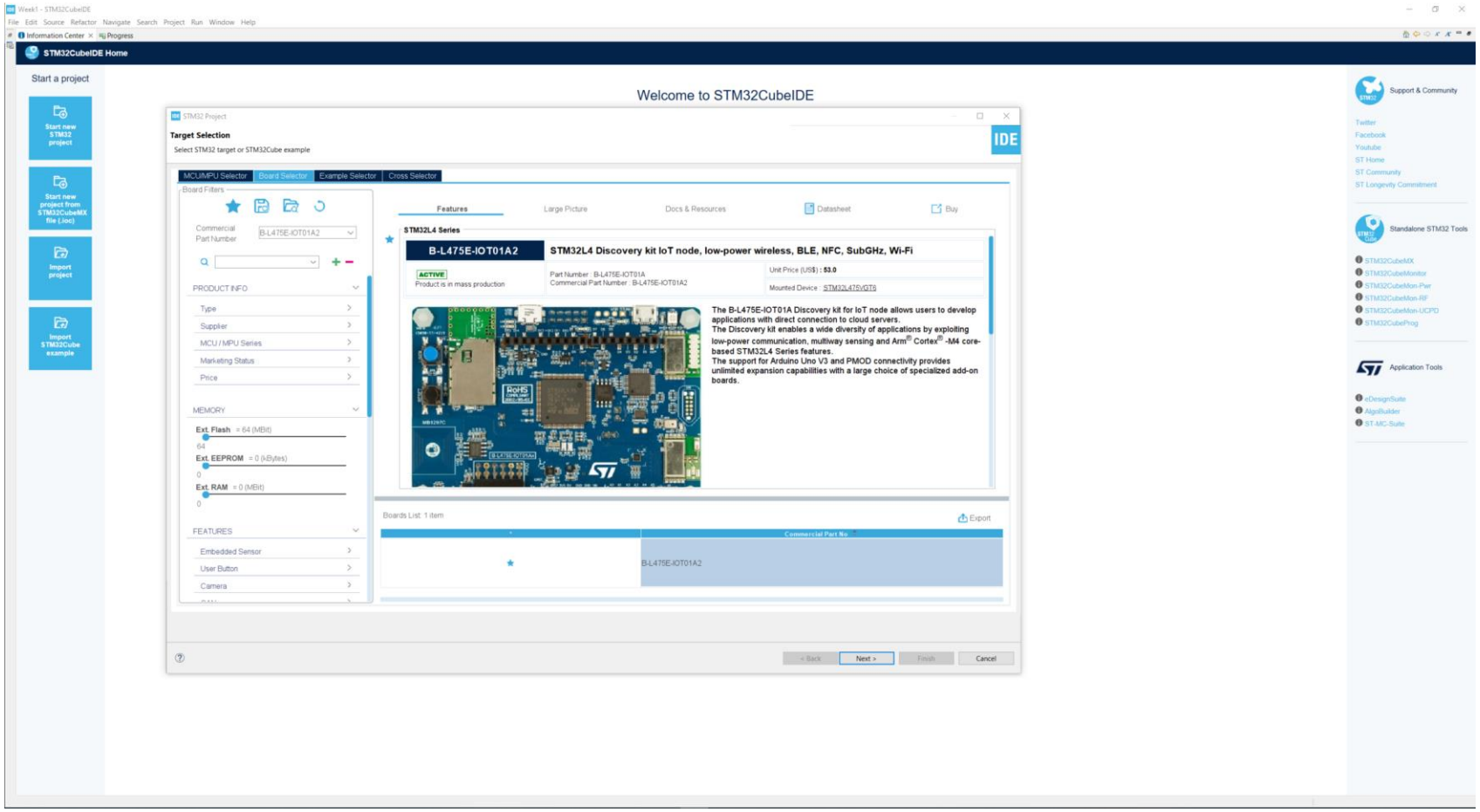
Hsuankai Chang

hsuankac@umich.edu

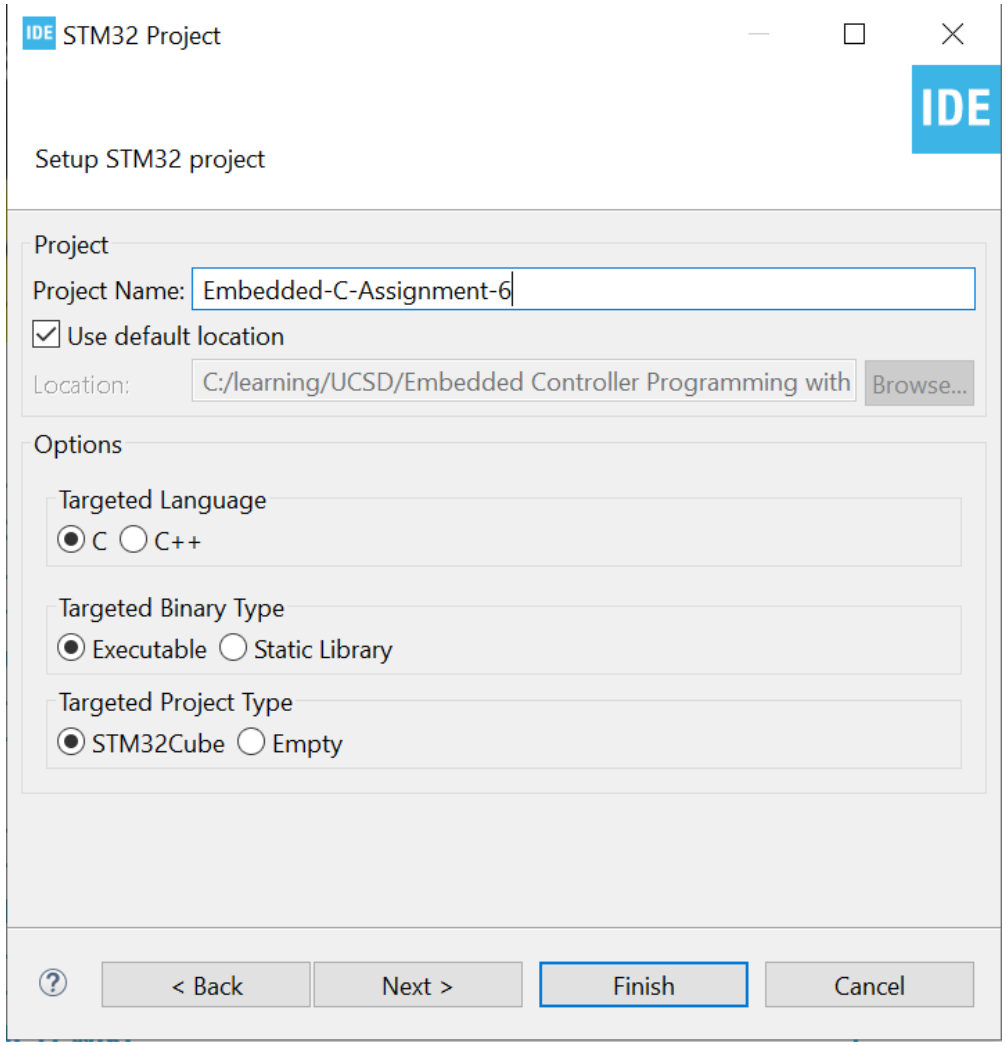
Step 1. Startup STM32CubeIDE and create new STM32 project



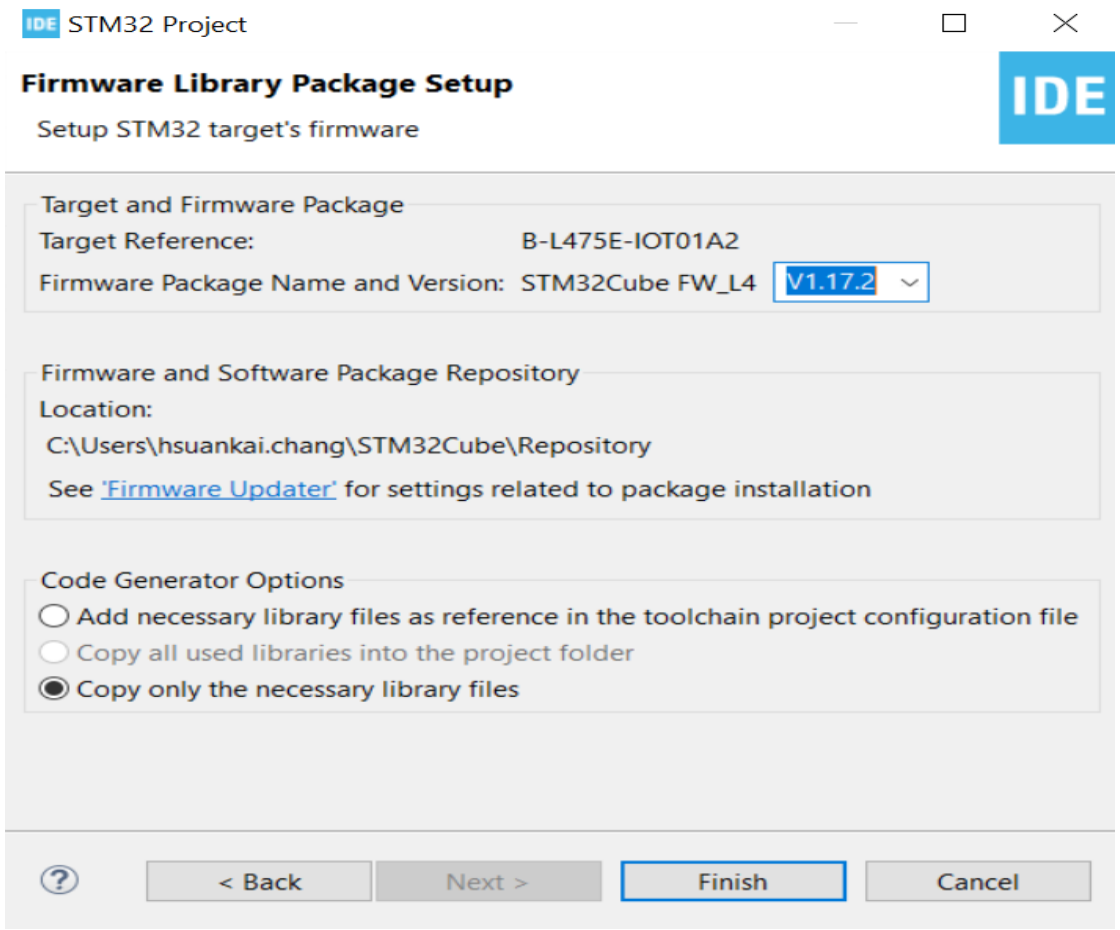
Step 2. Access board selector and type in the board you use, click Next



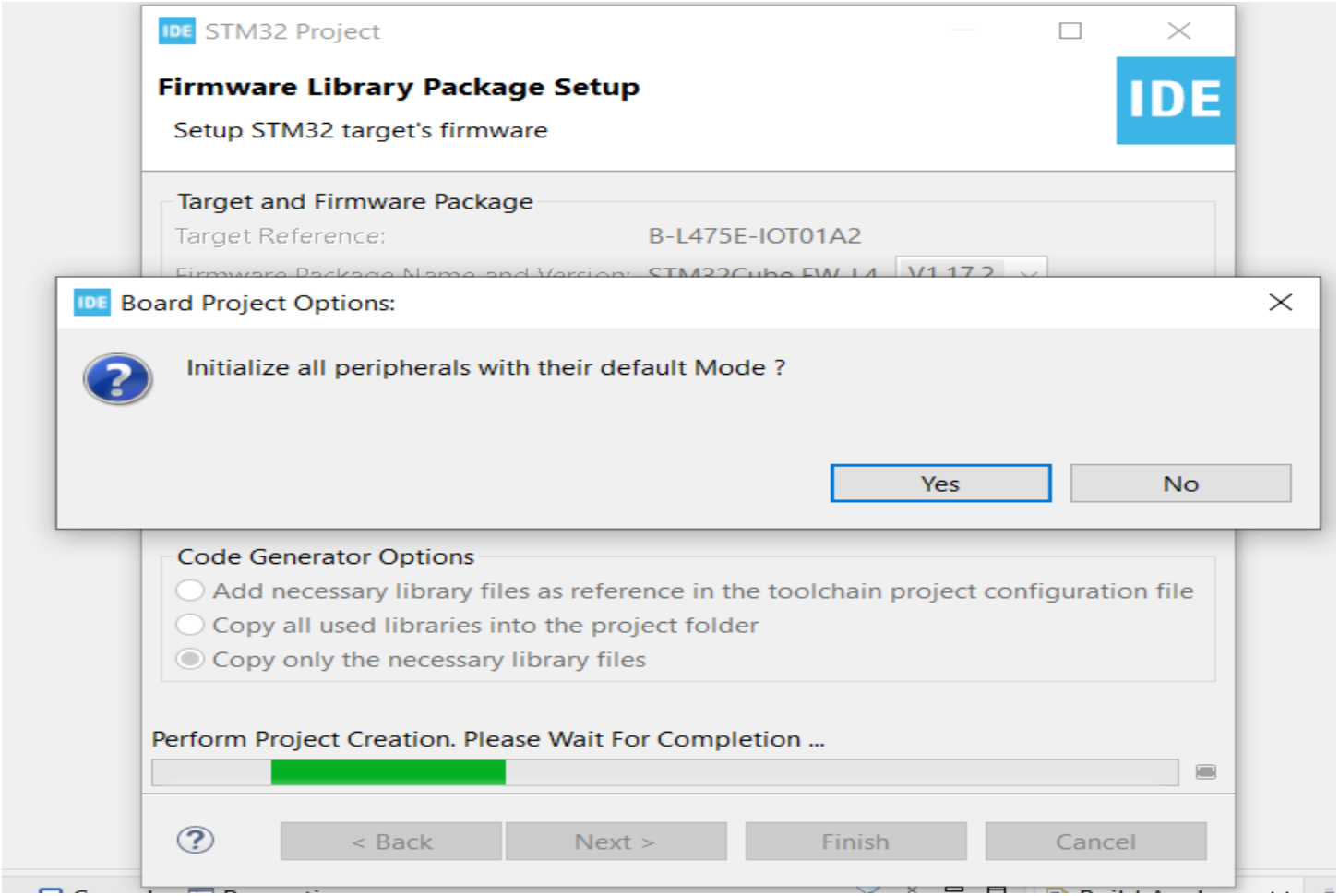
Step 3. Enter the project name then click Next



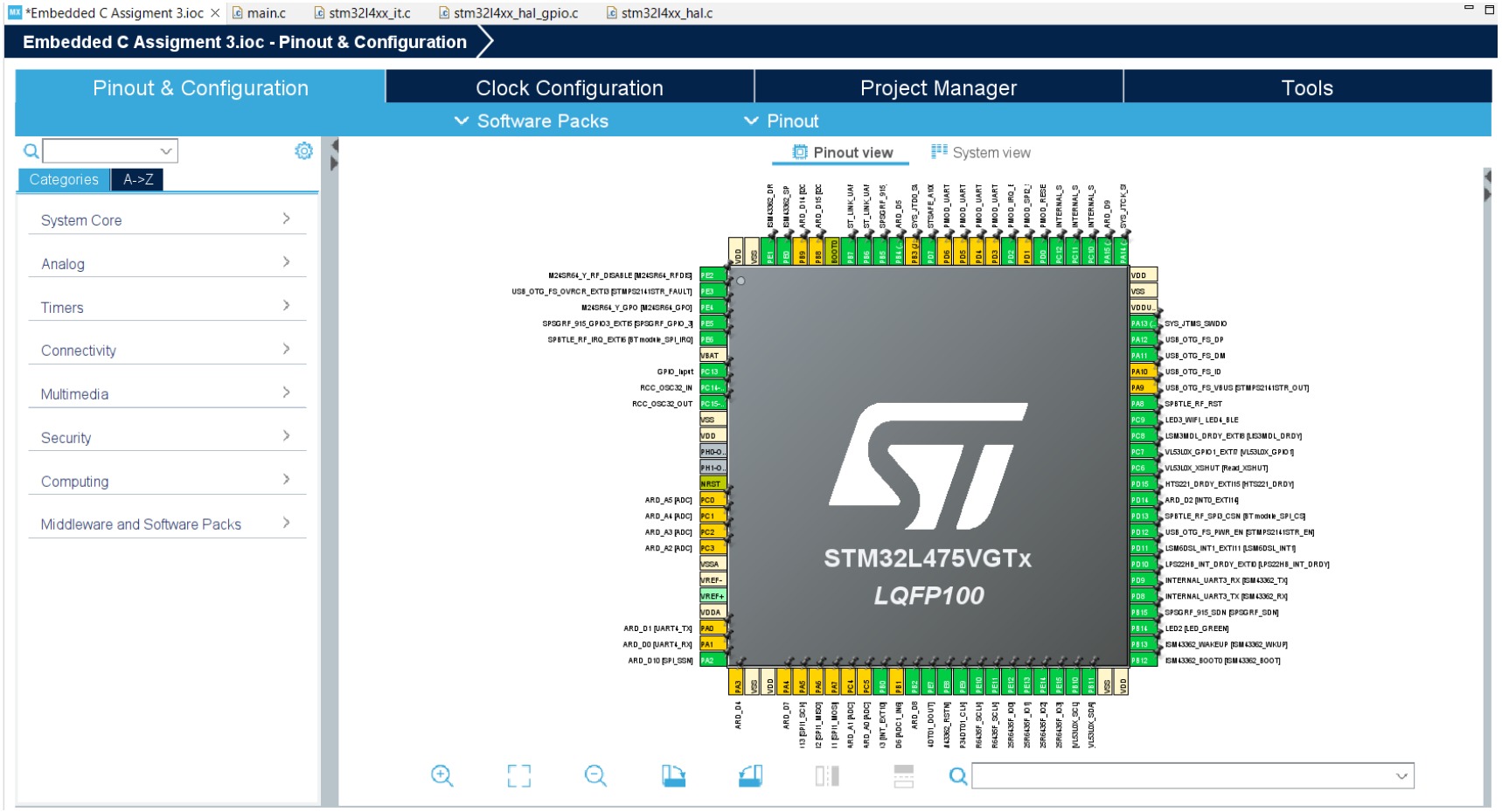
Step 4. See the firmware package name and version



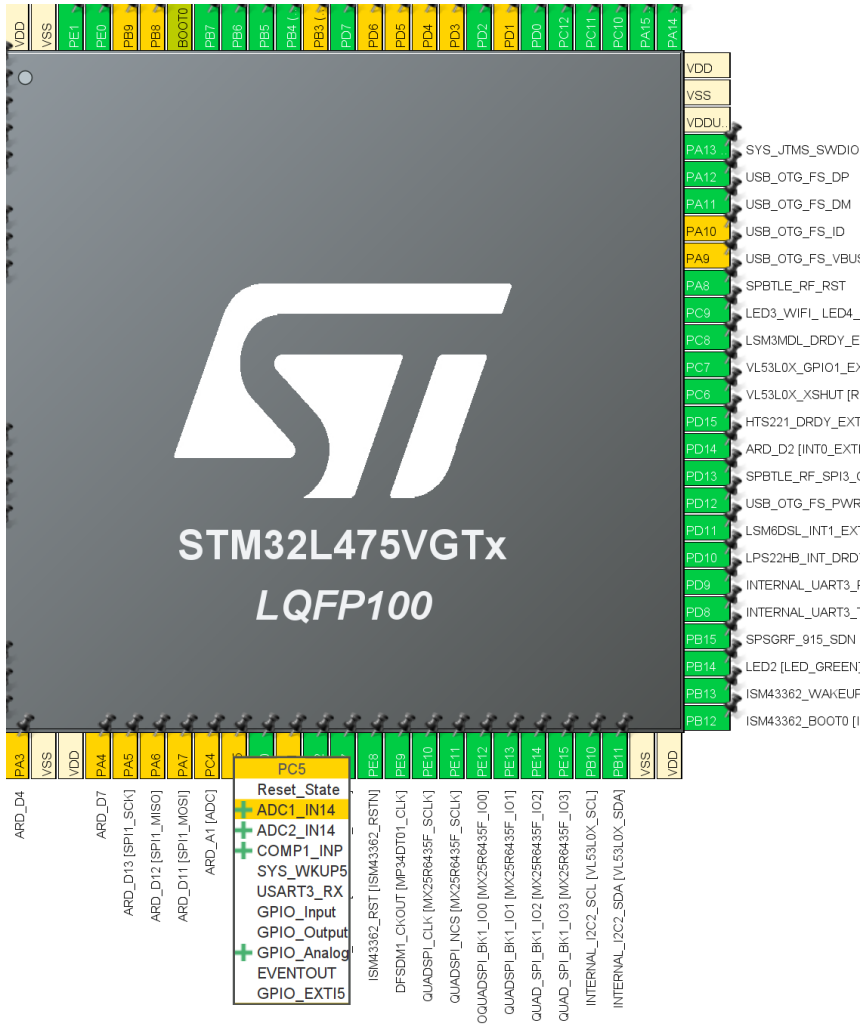
Step 5. Click yes to initialize all peripherals to default



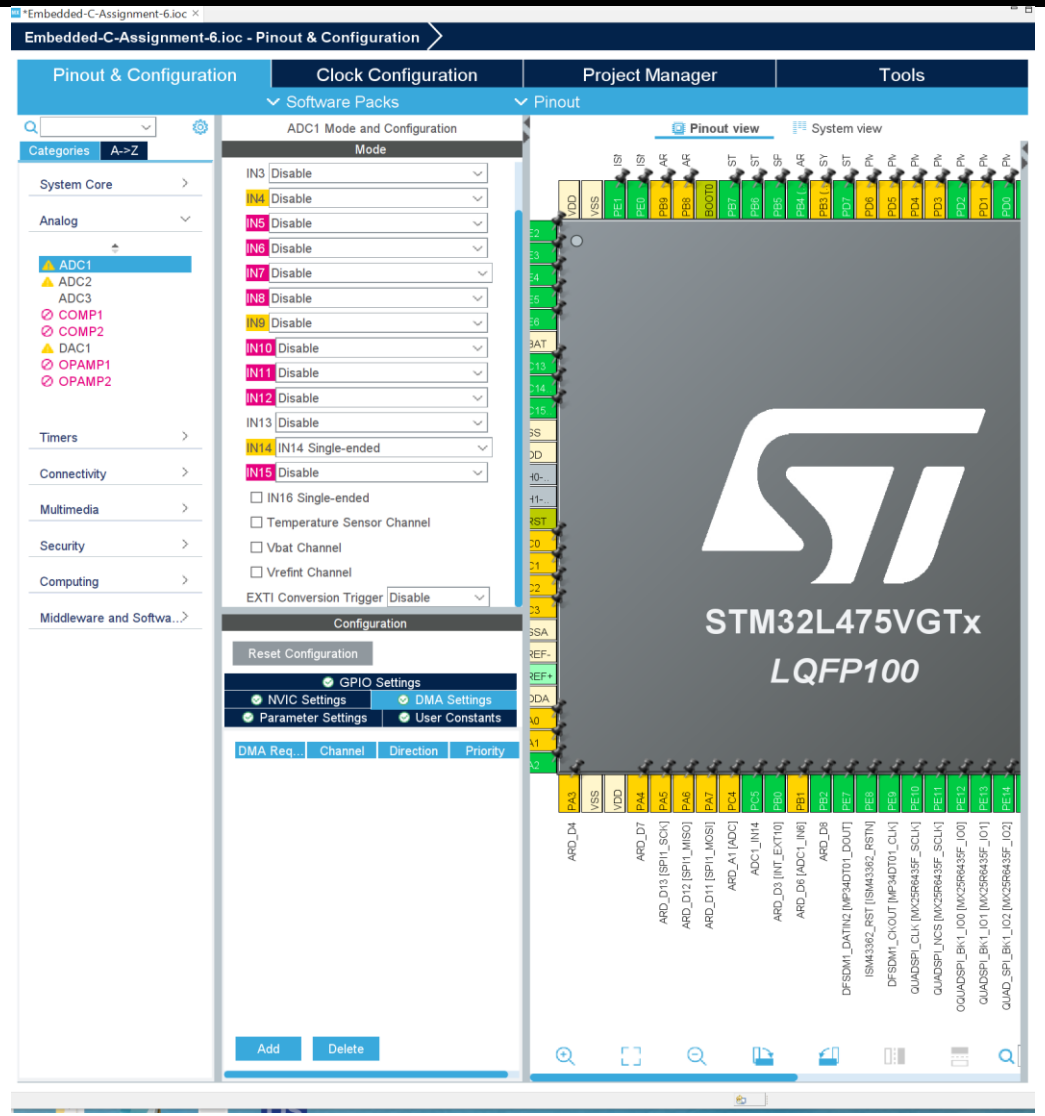
Step 6. When in .ioc file, click Pinout & Configurations



Step 7. Observe PC5 -> ADC1_IN14 -> ADR_A0



Step 8. Set ADC1, IN14 to Single-ended



Step 9. Generate code

Step 10. In main.c, find code that initializes the ADC1

Embedded-C-Assignment-6.ioc | main.c

```c
228 static void MX_ADC1_Init(void)
229 {
230
231   /* USER CODE BEGIN ADC1_Init 0 */
232
233   /* USER CODE END ADC1_Init 0 */
234
235   ADC_MultiModeTypeDef multimode = {0};
236   ADC_ChannelConfTypeDef sConfig = {0};
237
238   /* USER CODE BEGIN ADC1_Init 1 */
239
240   /* USER CODE END ADC1_Init 1 */
241
242   /** Common config
243   */
244   hadc1.Instance = ADC1;
245   hadc1.Init.ClockPrescaler = ADC_CLOCK_ASYNC_DIV1;
246   hadc1.Init.Resolution = ADC_RESOLUTION_12B;
247   hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;
248   hadc1.Init.ScanConvMode = ADC_SCAN_DISABLE;
249   hadc1.Init.EOCSelection = ADC_EOC_SINGLE_CONV;
250   hadc1.Init.LowPowerAutoWait = DISABLE;
251   hadc1.Init.ContinuousConvMode = DISABLE;
252   hadc1.Init.NbrOfConversion = 1;
253   hadc1.Init.DiscontinuousConvMode = DISABLE;
254   hadc1.Init.ExternalTrigConv = ADC_SOFTWARE_START;
255   hadc1.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;
256   hadc1.Init.DMAContinuousRequests = DISABLE;
257   hadc1.Init.Overrun = ADC_OVR_DATA_PRESERVED;
258   hadc1.Init.OversamplingMode = DISABLE;
259   if (HAL_ADC_Init(&hadc1) != HAL_OK)
260   {
261     Error_Handler();
262   }
263
264   /** Configure the ADC multi-mode
265   */
266   multimode.Mode = ADC_MODE_INDEPENDENT;
267   if (HAL_ADCEx_MultiModeConfigChannel(&hadc1, &multimode) != HAL_OK)
268   {
269     Error_Handler();
270   }
271
272   /** Configure Regular Channel
```

Outline | Build Targets

- main.h
- hadc1 : ADC_HandleTypeDef
- hdfsdm1_channel1 : DFSDM_Ch
- hi2c2 : I2C_HandleTypeDef
- hqspi : QSPI_HandleTypeDef
- hspi3 : SPI_HandleTypeDef
- huart1 : UART_HandleTypeDef
- huart3 : UART_HandleTypeDef
- hpcd_USB_OTG_FS : PCD_Handle
- SystemClock_Config(void) : void
- PeriphCommonClock_Config(vo
- MX_GPIO_Init(void) : void
- MX_DFSDM1_Init(void) : void
- MX_I2C2_Init(void) : void
- MX_QUADSPI_Init(void) : void
- MX_SPI3_Init(void) : void
- MX_USART1_UART_Init(void) : vo
- MX_USART3_UART_Init(void) : vo
- MX_USB_OTG_FS_PCD_Init(void)
- MX_ADC1_Init(void) : void
- main(void) : int
- SystemClock_Config(void) : void
- PeriphCommonClock_Config(vo
- MX_ADC1_Init(void) : void
- MX_DFSDM1_Init(void) : void
- MX_I2C2_Init(void) : void
- MX_QUADSPI_Init(void) : void
- MX_SPI3_Init(void) : void
- MX_USART1_UART_Init(void) : vo
- MX_USART3_UART_Init(void) : vo
- MX_USB_OTG_FS_PCD_Init(void)
- MX_GPIO_Init(void) : void
- Error_Handler(void) : void
- assert_failed(uint8_t*, uint32_t) :

Problems | Tasks | Console | Properties

0 items

Description	Resource	Path

Build Analyzer | Static Stack Analyzer | Cyclomatic Complexity

Memory Regions | Memory Details

Region	Start add...	End addr...	Size	Free	Used	Usage (%)

Writable | Smart Insert | 241 : 1 : 6667

Step 11. In main.c, find code that initialize the channel IN14

Embedded-C-Assignment-6.ioc | main.c

```c
  hadc1.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;
  hadc1.Init.DMAContinuousRequests = DISABLE;
  hadc1.Init.Overrun = ADC_OVR_DATA_PRESERVED;
  hadc1.Init.OversamplingMode = DISABLE;
  if (HAL_ADC_Init(&hadc1) != HAL_OK)
  {
    Error_Handler();
  }

  /** Configure the ADC multi-mode
  */
  multimode.Mode = ADC_MODE_INDEPENDENT;
  if (HAL_ADCEx_MultiModeConfigChannel(&hadc1, &multimode) != HAL_OK)
  {
    Error_Handler();
  }

  /** Configure Regular Channel
  */
  sConfig.Channel = ADC_CHANNEL_14;
  sConfig.Rank = ADC_REGULAR_RANK_1;
  sConfig.SamplingTime = ADC_SAMPLETIME_2CYCLES_5;
  sConfig.SingleDiff = ADC_SINGLE_ENDED;
  sConfig.OffsetNumber = ADC_OFFSET_NONE;
  sConfig.Offset = 0;
  if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK)
  {
    Error_Handler();
  }
  /* USER CODE BEGIN ADC1_Init 2 */

  /* USER CODE END ADC1_Init 2 */

}

/**
  * @brief DFSDM1 Initialization Function
  * @param None
  * @retval None
  */
static void MX_DFSDM1_Init(void)
{

  /* USER CODE BEGIN DFSDM1_Init 0 */

```

Step 12. Add code to main.c to read from ARD_A0

```c
  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_DFSDM1_Init();
  MX_I2C2_Init();
  MX_QUADSPI_Init();
  MX_SPI3_Init();
  MX_USART1_UART_Init();
  MX_USART3_UART_Init();
  MX_USB_OTG_FS_PCD_Init();
  MX_ADC1_Init();
  /* USER CODE BEGIN 2 */
  uint32_t adcResult;
  /* USER CODE END 2 */

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

    /* USER CODE BEGIN 3 */
    HAL_ADC_Start(&hadc1);
    HAL_ADC_PollForConversion(&hadc1, 100);
    adcResult = HAL_ADC_GetValue(&hadc1);

    printf("adcResult: %lu - 0x%lx\n", adcResult, adcResult);

    HAL_ADC_Stop(&hadc1);

    HAL_Delay(1000);
  }
  /* USER CODE END 3 */
}
```

Step 13. Build project and run in debug mode

Embedded-C-Assignment-6.ioc | main.c | stm32l4xx_hal_adc.c

```c
/* USER CODE END 0 */

/**
  * @brief  The application entry point.
  * @retval int
  */
int main(void)
{
  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

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

  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
  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();
  MX_USART3_UART_Init();
  MX_USB_OTG_FS_PCD_Init();
  MX_ADC1_Init();
  /* USER CODE BEGIN 2 */
  uint32_t adcResult;
  /* USER CODE END 2 */

```

Name	Type
adcResult	uint32_t

Console | Problems | Executables | Debugger Console | Memory

Embedded-C-Assignment-6 Debug [STM32 C/C++ Application] [pid: 27]

```
Time elapsed during download operation: 00:00:01.088


Verifying ...


Download verified successfully
```

Writable | Smart Insert | 96 : 1 : 2749

Step 14, Set the break point, then read the adcResult when nothing connected to ARD_A0. In my case adcResult = 916

Embedded-C-Assignment-6.ioc × main.c × stm32l4xx_hal_adc.c

```c
  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_DFSDM1_Init();
  MX_I2C2_Init();
  MX_QUADSPI_Init();
  MX_SPI3_Init();
  MX_USART1_UART_Init();
  MX_USART3_UART_Init();
  MX_USB_OTG_FS_PCD_Init();
  MX_ADC1_Init();
  /* USER CODE BEGIN 2 */
  uint32_t adcResult;
  /* USER CODE END 2 */

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

    /* USER CODE BEGIN 3 */
    HAL_ADC_Start(&hadc1);
    HAL_ADC_PollForConversion(&hadc1, 100);
    adcResult = HAL_ADC_GetValue(&hadc1);

    printf("adcResult: %lu - 0x%lx\n", adcResult, adcResult);

    HAL_ADC_Stop(&hadc1);

    HAL_Delay(1000);
  }
  /* USER CODE END 3 */
}

/**
  * @brief System Clock Configuration
  * @retval None
  */
void SystemClock_Config(void)
{
  RCC_OscInitTypeDef RCC_OscInitStruct = {0};
  RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

  /** Configure the main internal regulator output voltage
  */
```

Variables × Breakpoints Expressions Registers Live Expre... SFRs

Name	Type	Value
adcResult	uint32_t	1421

Console × Problems Executables Debugger Console Memory

Embedded-C-Assignment-6 Debug [STM32 C/C++ Application] [pid: 27]

Step 15. Connect ADR_A0 to GND, then adcResult = 0

Embedded-C-Assignment-6.ioc | main.c | stm32l4xx_hal_adc.c

```c
  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_DFSDM1_Init();
  MX_I2C2_Init();
  MX_QUADSPI_Init();
  MX_SPI3_Init();
  MX_USART1_UART_Init();
  MX_USART3_UART_Init();
  MX_USB_OTG_FS_PCD_Init();
  MX_ADC1_Init();
  /* USER CODE BEGIN 2 */
  uint32_t adcResult;
  /* USER CODE END 2 */

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

    /* USER CODE BEGIN 3 */
    HAL_ADC_Start(&hadc1);
    HAL_ADC_PollForConversion(&hadc1, 100);
    adcResult = HAL_ADC_GetValue(&hadc1);

    printf("adcResult: %lu - 0x%lx\n", adcResult, adcResult);

    HAL_ADC_Stop(&hadc1);

    HAL_Delay(1000);
  }
  /* USER CODE END 3 */
}

/**
  * @brief System Clock Configuration
  * @retval None
  */
void SystemClock_Config(void)
{
  RCC_OscInitTypeDef RCC_OscInitStruct = {0};
  RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

  /** Configure the main internal regulator output voltage
  */
```

Variables | Breakpoints | Expressions | Registers | Live Expre... | SFRs

Name	Type	Value
adcResult	uint32_t	0

Console | Problems | Executables | Debugger Console | Memory

Embedded-C-Assignment-6 Debug [STM32 C/C++ Application] [pid: 27]

Step 16. Connect ADR_A0 to 1.5VDC. Then adcResult = 1856

Embedded-C-Assignment-6.ioc | main.c | stm32l4xx_hal_adc.c | startup_stm32l475vgtx.s

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

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

    /* USER CODE BEGIN 3 */
    HAL_ADC_Start(&hadc1);
    HAL_ADC_PollForConversion(&hadc1, 100);
    adcResult = HAL_ADC_GetValue(&hadc1);

    printf("adcResult: %lu - 0x%lx\n", adcResult, adcResult);

    HAL_ADC_Stop(&hadc1);

    HAL_Delay(1000);
  }
  /* USER CODE END 3 */
}

/**
  * @brief System Clock Configuration
  * @retval None
  */
void SystemClock_Config(void)
{
  RCC_OscInitTypeDef RCC_OscInitStruct = {0};
  RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

  /** Configure the main internal regulator output voltage
  */
  if (HAL_PWREx_ControlVoltageScaling(PWR_REGULATOR_VOLTAGE_SCAI
  {
    Error_Handler();
  }

  /** Configure LSE Drive Capability
  */
  HAL_PWR_EnableBkUpAccess();
  __HAL_RCC_LSEDRIVE_CONFIG(RCC_LSEDRIVE_LOW);

  /** Initializes the RCC Oscillators according to the specifie
  * in the RCC_OscInitTypeDef structure.
```

Variables | Breakpoints | Expressions | Registers | Live Expre... | SFRs

Name	Type	Value
adcResult	uint32_t	1856

```
Name : adcResult
    Details:1845
    Default:1845
    Decimal:1845
    Hex:0x735
    Binary:11100110101
    Octal:03465
```

Console | Problems | Executables | Debugger Console | Memory

Embedded-C-Assignment-6 Debug [STM32 C/C++ Application] [pid: 32]

```
Time elapsed during download operation: 00:00:01.097


Verifying ...


Download verified successfully
```

Writable | Smart Insert | 136 : 1 : 3604