

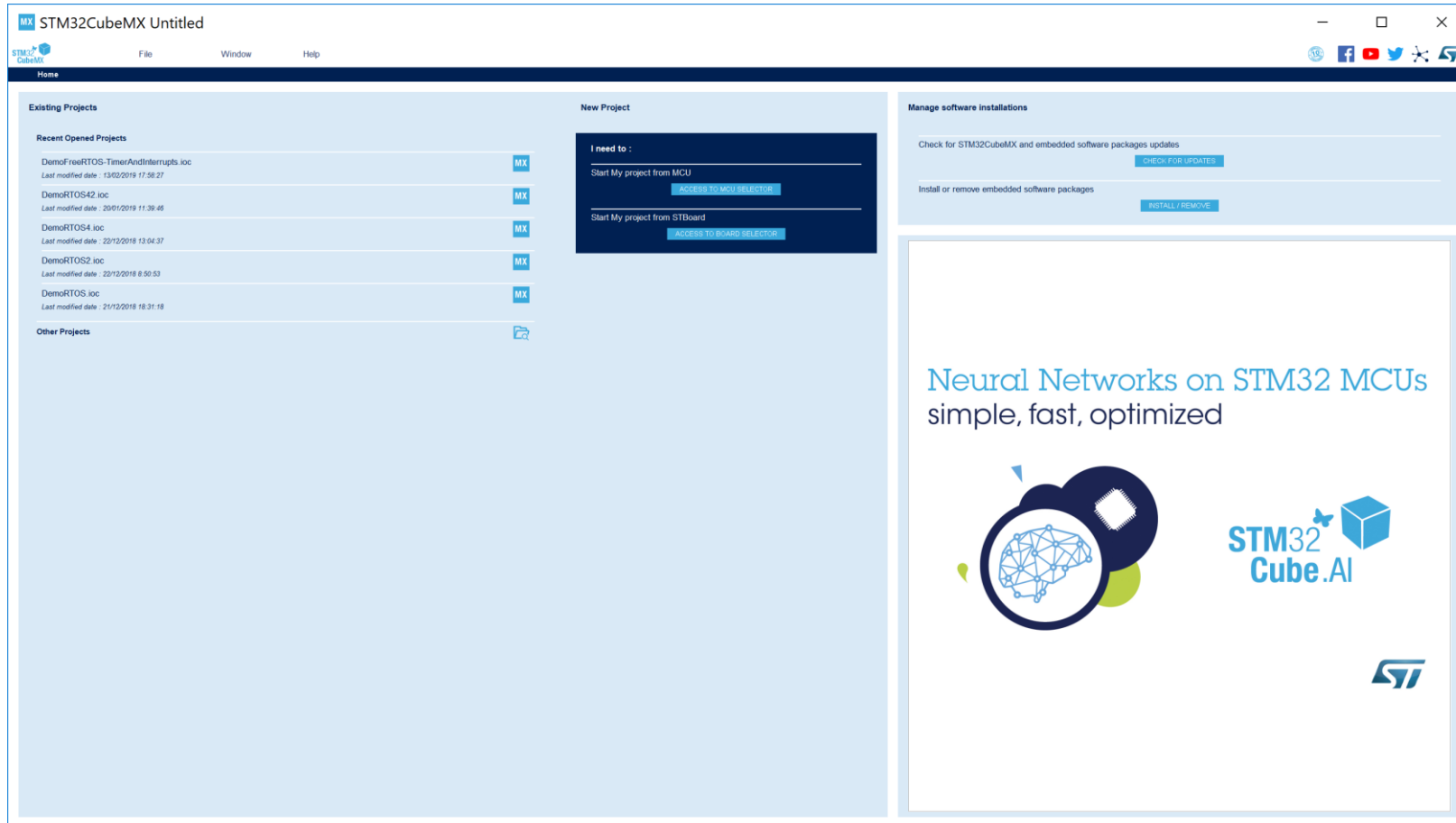
USCD Embedded C Assignment 6

By

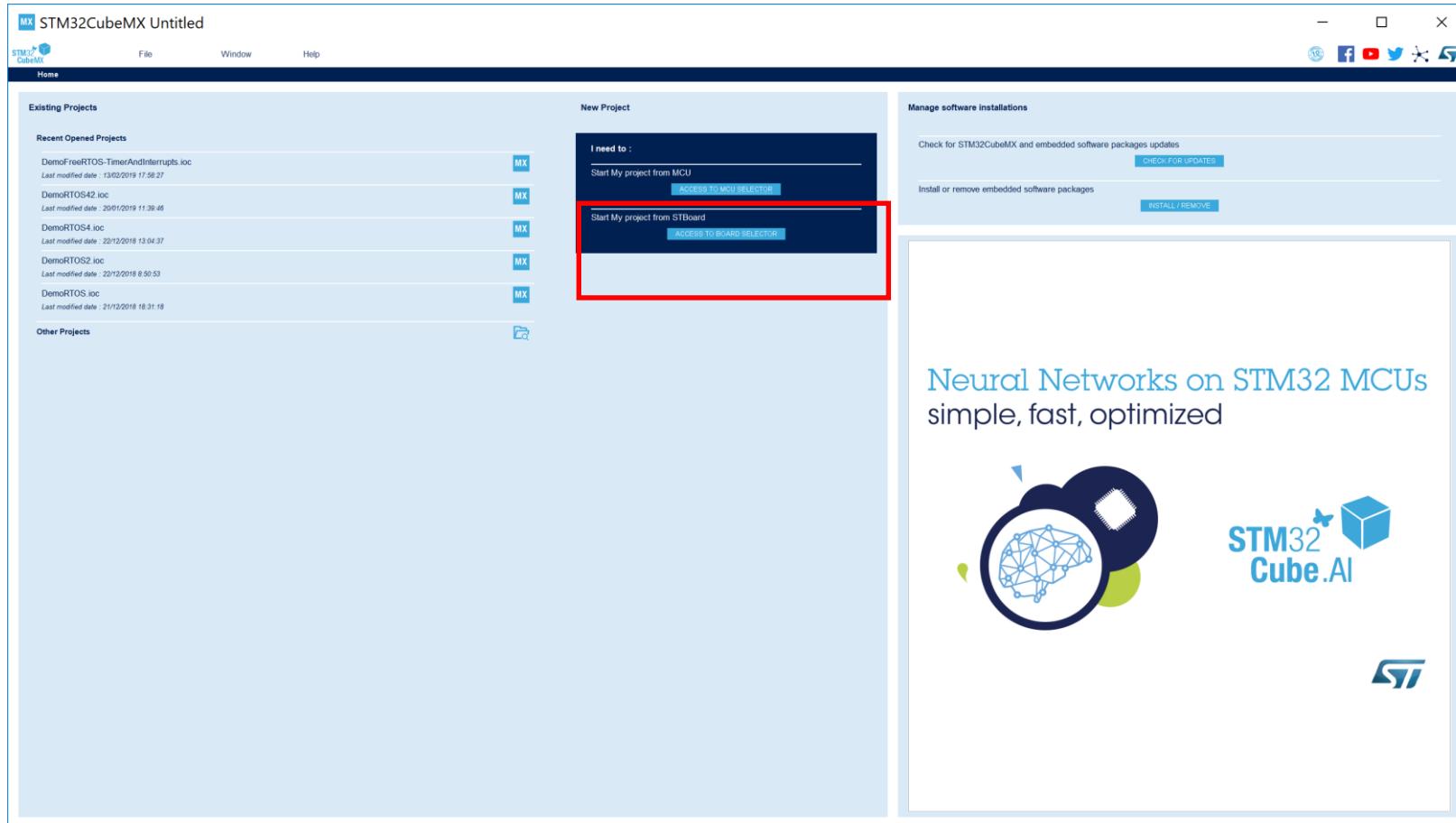
Norman McEntire

Norman.mcentire@gmail.com

Step 1. Startup STM32CubeMX



Step 2. Access Board Selector



Step 3. Enter “B-L475E-IOT01A” Board

New Project from a Board

Board Filter

Product Search:

Vendor:

Type:

MCU Series:

Other:

Peripheral:

- ☒ Accelerometer
- ☐ Analog I/O
- ☐ Analog Front End
- ☐ Audio Line In
- ☐ Audio Line Out
- ☐ Button
- ☐ Camera
- ☐ Display
- ☐ External Flash
- ☐ External I2C
- ☐ External SPI
- ☐ External UART
- ☐ External USB
- ☐ GPS
- ☐ Gyroscope
- ☐ I2C
- ☐ I2C Device (Slave)
- ☐ I2C Device (Master)
- ☐ LED
- ☐ Magnetometer
- ☐ Memory Card
- ☐ Microphone
- ☐ On-board Debug
- ☐ Other
- ☐ Power Source
- ☐ Pressure Sensor
- ☐ ROM
- ☐ RS-232
- ☐ RS-485
- ☐ RS-485/RS-422
- ☐ RS-485/RS-422
- ☐ Temperature Sensor
- ☐ Touch Screen
- ☐ USB

B-L475E-IOT01A

STMicroelectronics B-L475E-IOT01A IOT Discovery Board Support and Examples

ACTIVE Active
Product is in mass production

Unit Price (USD): \$3.8
Mounted device: [STM32L475G6Z](#)

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 the power communication, multi-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-LINKV2-1
- Supply through ST-Link USB
- USB OTG (Full speed) with micro-AB Connector
- Blue Tooth module
- WiFi module
- 8 MByte QuadSPI Flash
- ST MEMS 3-axis accelerometer (LSM2DS1)
- ST MEMS gyroscope (LSM2DS1)
- ST MEMS magnetometer (LSM2DS1)
- ST MEMS barometer (LPS2DS1)
- ST MEMS humidity and temperature (HTS221)
- Digital microphone (MP34DT01)
- RS-232 serial interface
- Two Push-buttons: User and Reset
- Seven LEDs: ST-LINKV2-1 communication, 3.3 V Power, Over current, USB COM (Red/Green), User (2*Green/Yellow/Blue)


Boards List: 1 item

Image	Chipset	Type	Status	Unit Price (USD)	Mounted device
	B-L475E-IOT01A	Discovery	Active	\$3.8	STM32L475G6Z

Step 4. Select Board Photo

The screenshot shows the 'New Project from a Board' dialog with the following components:

- Board Filters:**
 - Part Number Search: B-L475E-IOT01A
 - Vendor: STMMicroelectronics
 - Type: Discovery
 - MCU Series: STM32L4
 - Other: Price >= \$3.8, Oscillator Freq. >= 0 (MHz)
 - Peripheral: Accelerometer, Audio, Button, Gyroscope, LED, Magnetometer, Microphone, On-board Debug, Other, Power Source, Pressure Sensor, ROM, RS-232, Serial, Temperature Sensor, USB
- Board Details (B-L475E-IOT01A):**
 - Features: STM32L475E-IOT01A IOT Discovery Board Support and Examples
 - Unit Price (USD): \$3.8
 - Mounted device: STM32L475E-IOT01A
 - Features list:
 - On-board ST-LINK/V2-1
 - Supply through ST-Link USB
 - USB OTG (Full speed) with micro-AB Connector
 - Blue Tooth module
 - IoT module
 - 8 MByte QuadSPI Flash
 - ST MEMS 3-axis accelerometer (LSM6DSL)
 - ST MEMS gyroscope (LSM6DSL)
 - ST MEMS magnetometer (LSM6DSL)
 - ST MEMS barometer (LPS22DH)
 - ST MEMS humidity and temperature (HTS221)
 - Digital microphone (MP34DT01)
 - RS-232 serial interface
 - Two Push-buttons: User and Reset
 - Seven LEDs: ST-LINK/V2-1 communication, 3.3 V Power, Over current, USB COM (Red/Green), User (2*Green/Yellow/Blue)
- Boards List:**

Icon	Part Number	Type	Status	Unit Price (USD)	Mounted device
	B-L475E-IOT01A	Discovery	Active	\$3.8	STM32L475E-IOT01A

Step 5. Select “Start Project”

The screenshot shows the 'New Project from a Board' dialog box. On the left, there are filters for Board Number Search, Vendor (STMicroelectronics), Type (Discovery), MCU Series (STM32L4), and Other (Price, Oscillator Frequency). A list of peripherals is shown on the far left. The main area displays details for the 'B-L475E-IOT01A' board, including its features and a 'Start Project' button, which is highlighted with a red rectangle. Below the main area is a 'Boards List' table.

Board Details:

- Part Number Search: B-L475E-IOT01A
- Vendor: STMicroelectronics
- Type: Discovery
- MCU Series: STM32L4
- Other: Price = \$3.8, Oscillator Freq = 0 (MHz)

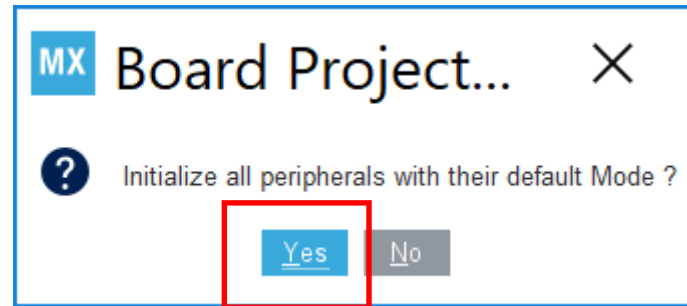
Features:

- On-board ST-LINKV2-1
- Supply through ST-Link USB
- USB OTG (Full speed) with micro-AB Connector
- Blue Tooth module
- WiFi module
- 8 MByte QuadSPI Flash
- ST MEMS 3-axis accelerometer (LSM2DS1)
- ST MEMS gyroscope (LSM2DS1)
- ST MEMS magnetometer (LSM2DS1)
- ST MEMS barometer (PS220B)
- ST MEMS humidity and temperature (HTS221)
- Digital microphone (MP34DT01)
- RS-232 serial interface
- Two Push-buttons: User and Reset
- Seven LEDs: ST-LINKV2-1 communication, 3.3 V Power, Over current, USB COM (Red/Green), User (2*Green/Yellow/Blue)

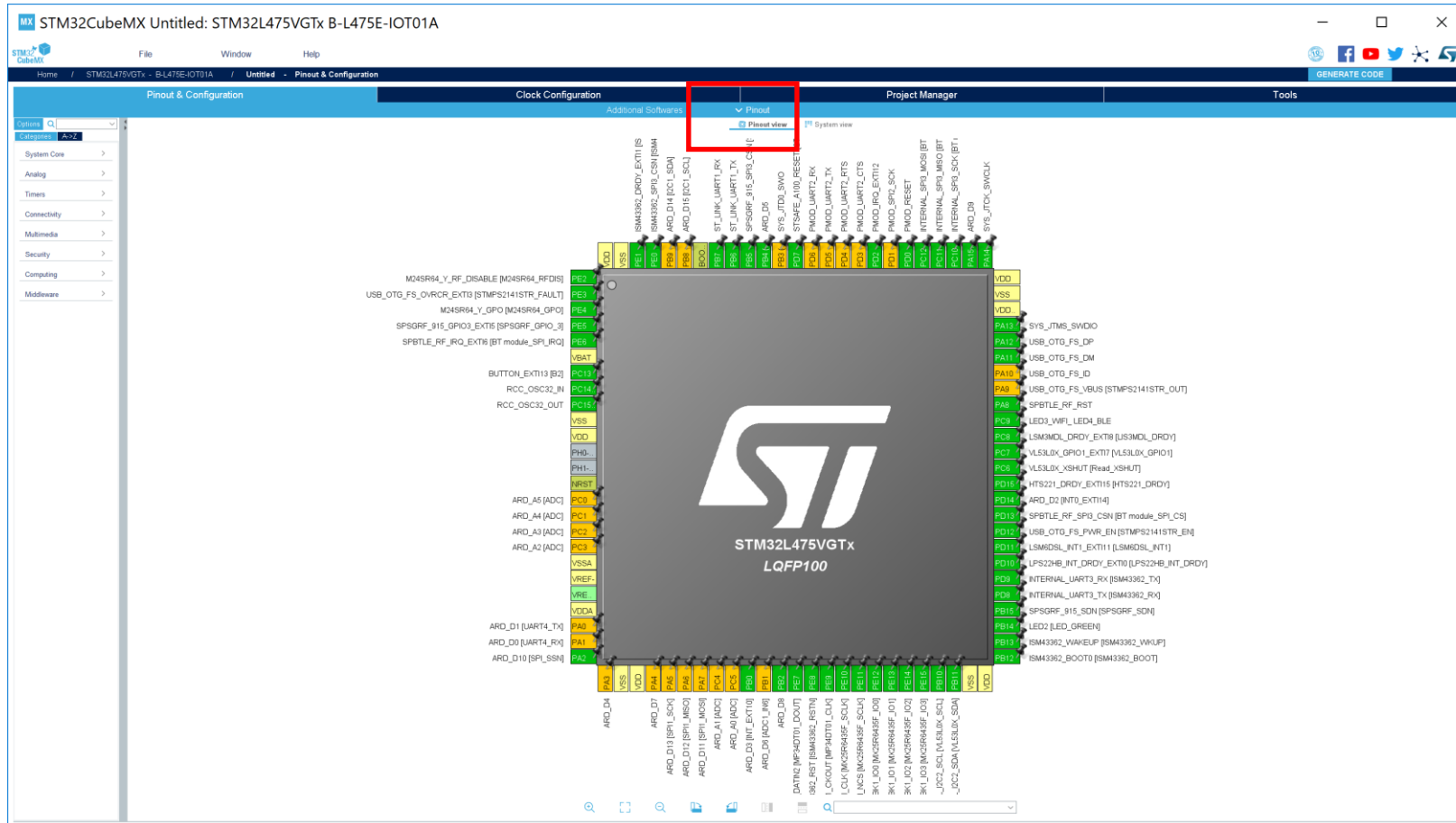
Boards List:

Part Number	Image	Part Number	Type	Status	Price (USD)	Mounted device
B-L475E-IOT01A		Discovery	Active	\$3.8	STMicroelectronics	

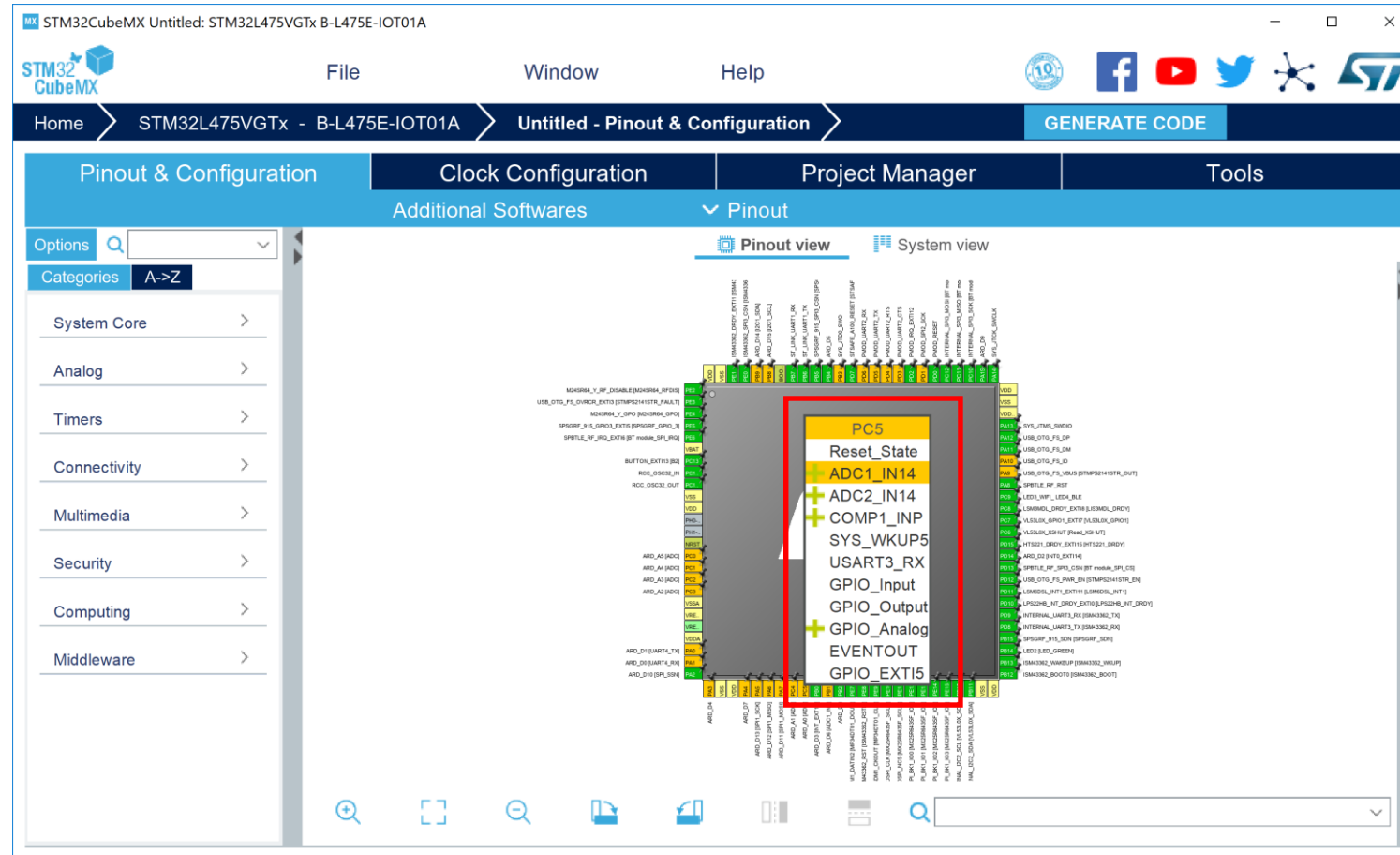
Step 6. Select **YES** (initialize all peripherals with the default mode)



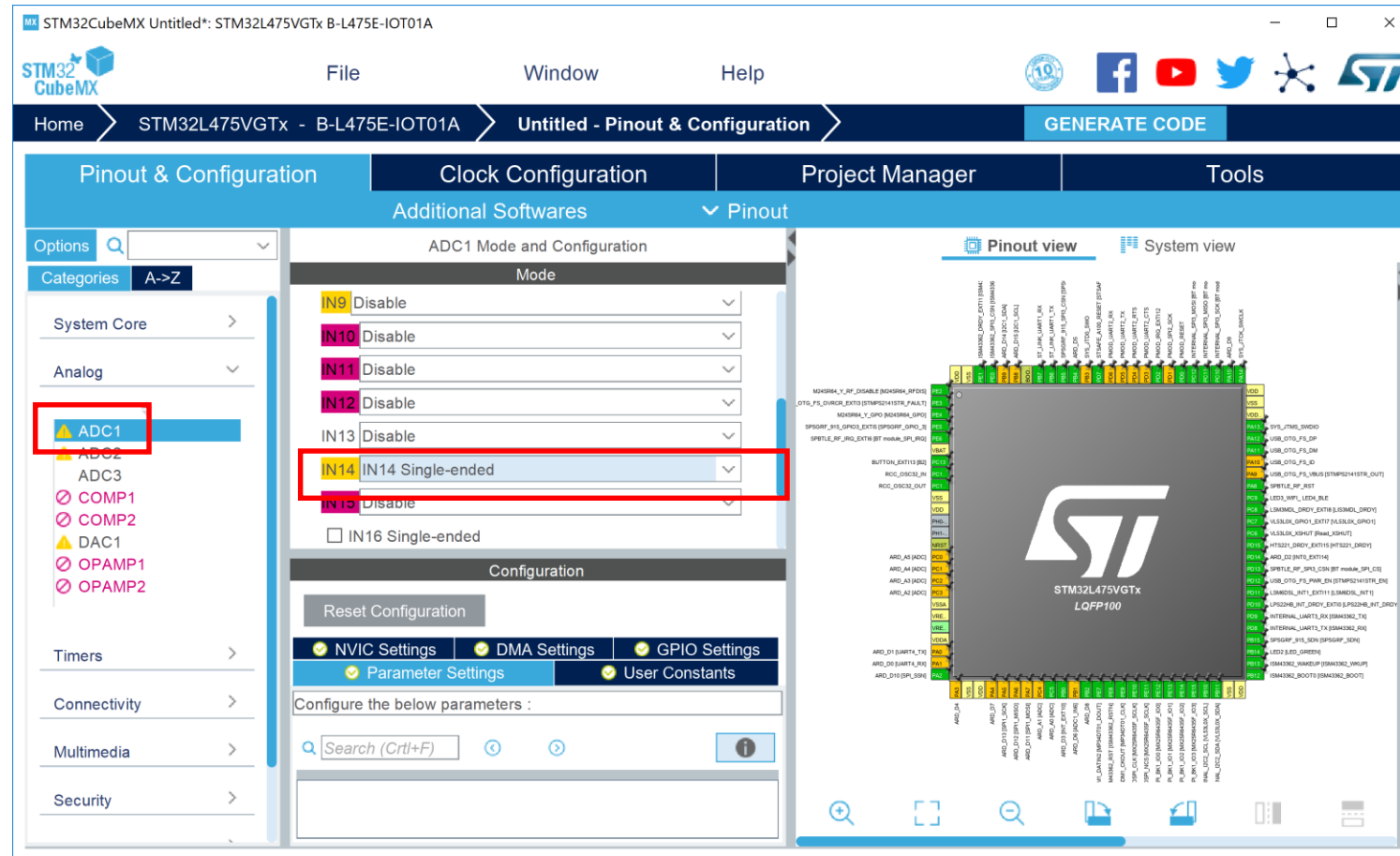
Step 7. Observe Results (Pinout View)



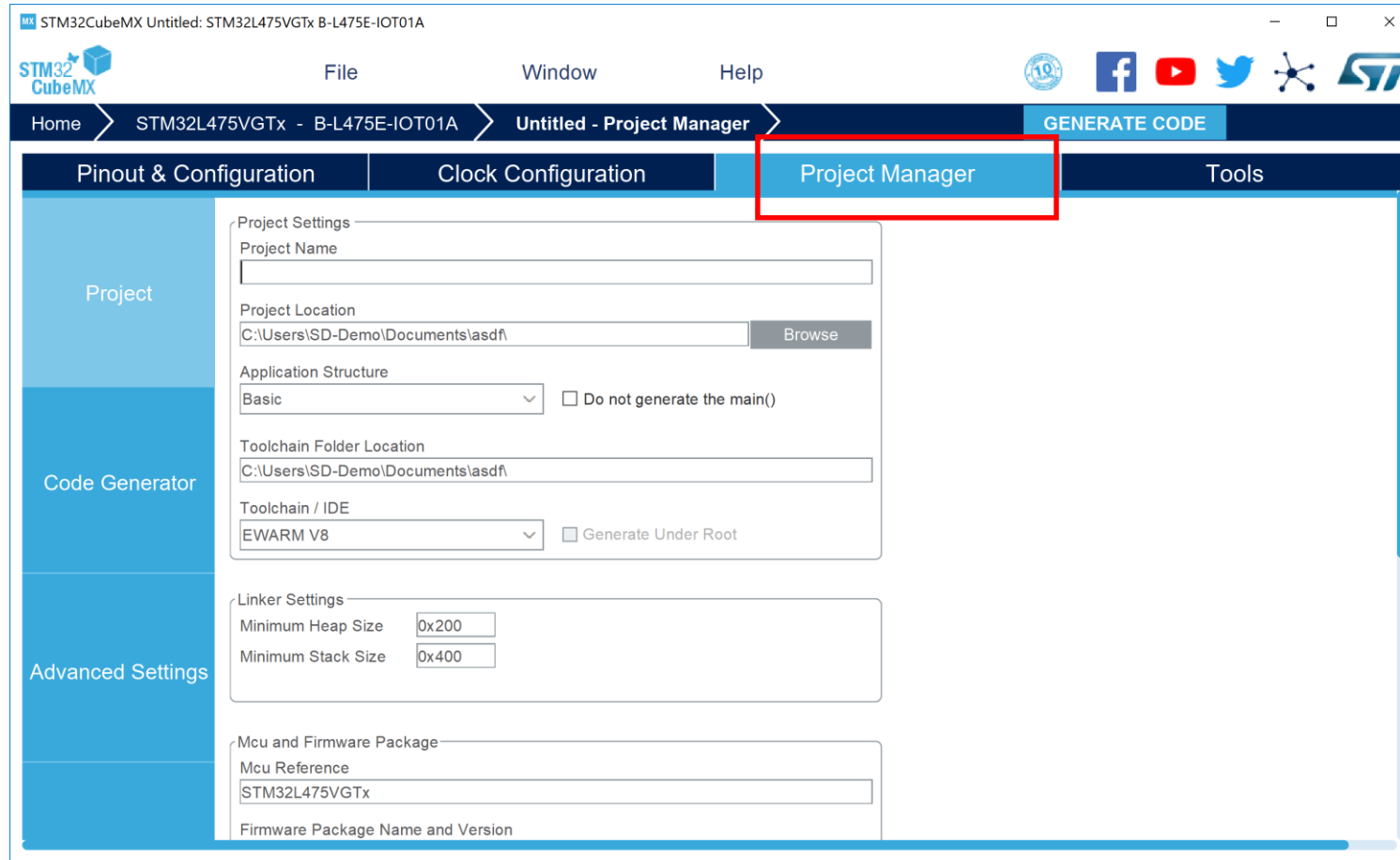
Step 8. Observe PC5 – ADC1_IN14 – ARD_A0



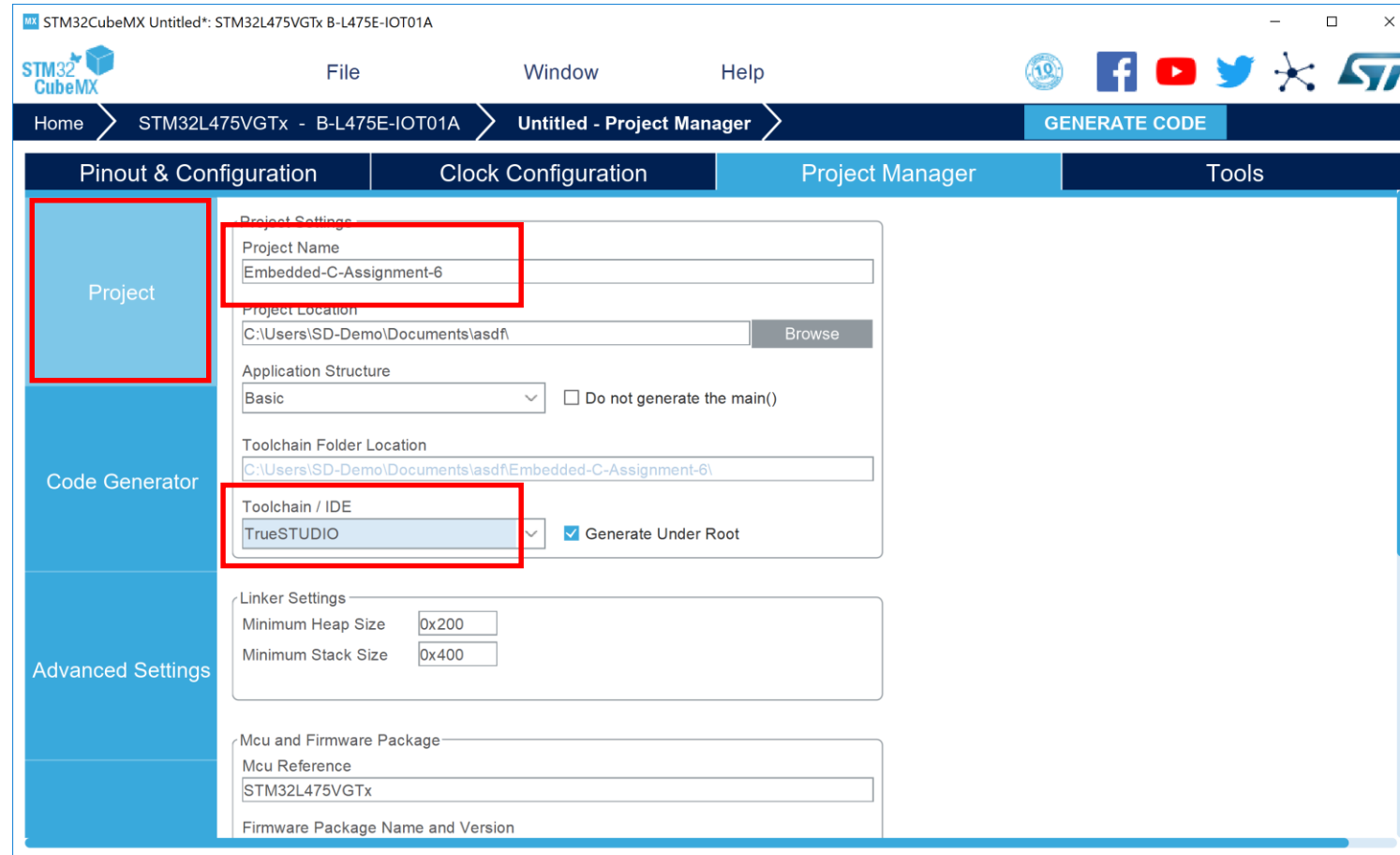
Step 9. Set ADC1, IN14 to Single-ended



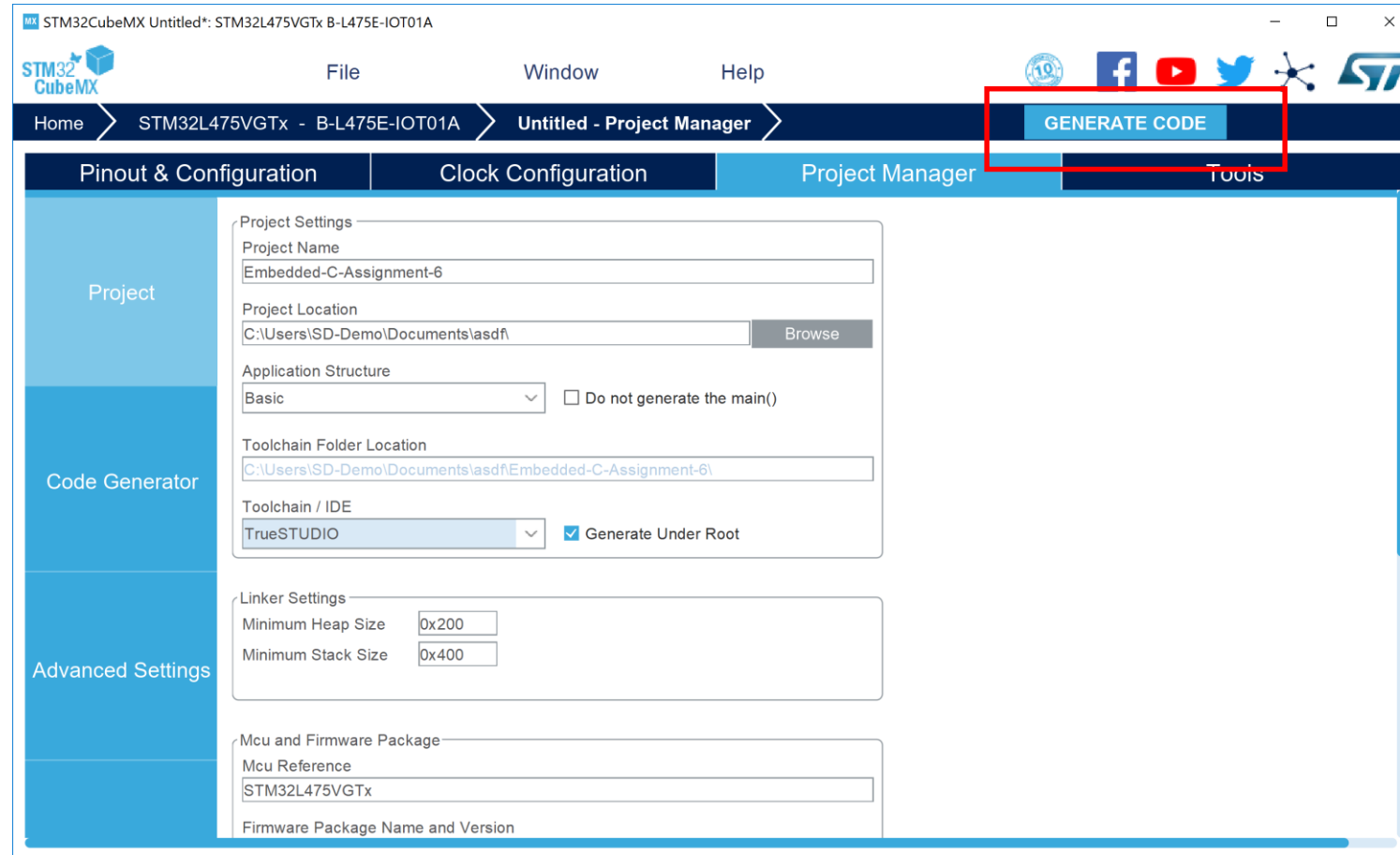
Step 10. Select Project Manager Tab



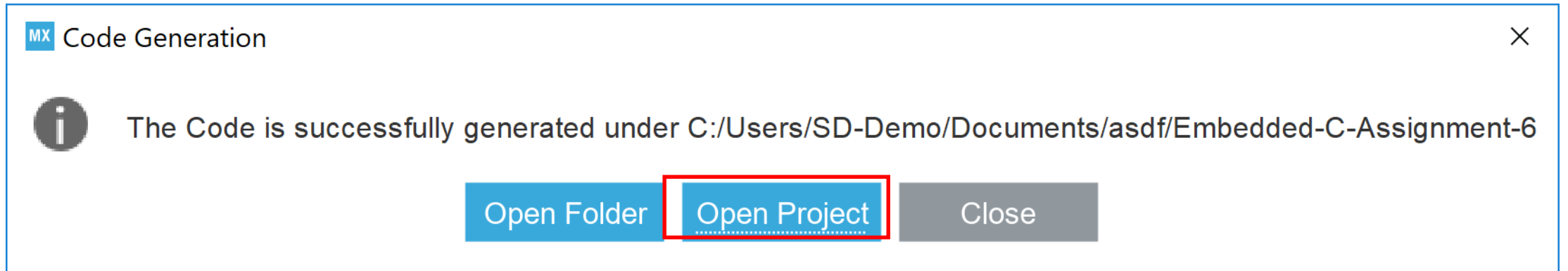
Step 11. Enter “Embedded-C-Assignment-6” and select TrueStudio as IDE



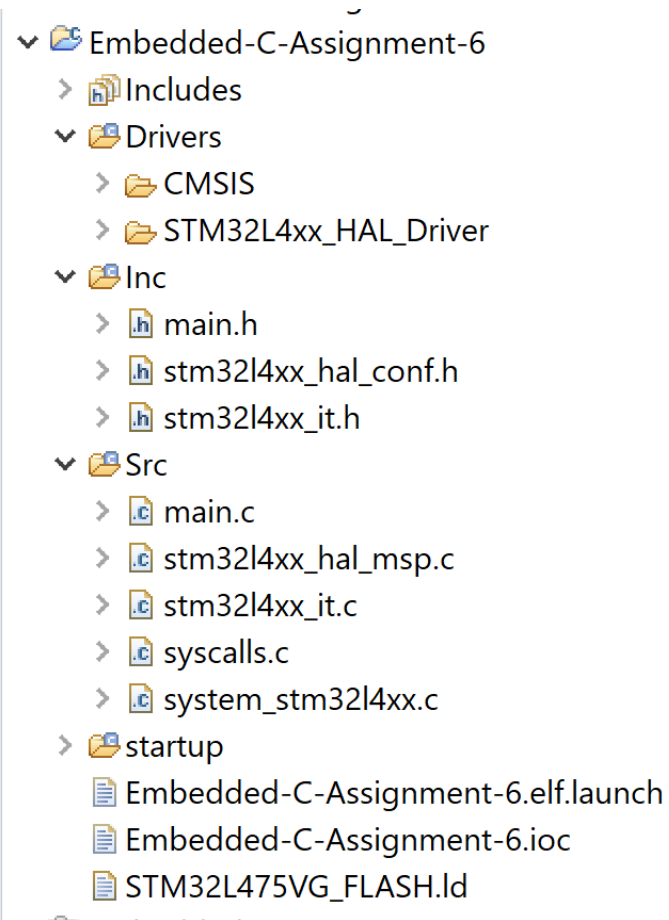
Step 12. Select “Generate Code”



Step 13. Select “Open Project”



Step 14. Resulting Project



Step 15. In main.c, find code that initializes the ADC1

```
216 static void MX_ADC1_Init(void)
217 {
218
219     /* USER CODE BEGIN ADC1_Init 0 */
220
221     /* USER CODE END ADC1_Init 0 */
222
223     ADC_MultiModeTypeDef multimode = {0};
224     ADC_ChannelConfTypeDef sConfig = {0};
225
226     /* USER CODE BEGIN ADC1_Init 1 */
227
228     /* USER CODE END ADC1_Init 1 */
229     /** Common config
230     */
231     hadc1.Instance = ADC1;
232     hadc1.Init.ClockPrescaler = ADC_CLOCK_ASYNC_DIV1;
233     hadc1.Init.Resolution = ADC_RESOLUTION_12B;
234     hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;
235     hadc1.Init.ScanConvMode = ADC_SCAN_DISABLE;
236     hadc1.Init.EOCSelection = ADC_EOC_SINGLE_CONV;
237     hadc1.Init.LowPowerAutoWait = DISABLE;
238     hadc1.Init.ContinuousConvMode = DISABLE;
239     hadc1.Init.NbrOfConversion = 1;
240     hadc1.Init.DiscontinuousConvMode = DISABLE;
241     hadc1.Init.ExternalTrigConv = ADC_SOFTWARE_START;
242     hadc1.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;
243     hadc1.Init.DMAContinuousRequests = DISABLE;
244     hadc1.Init.Overflow = ADC_OVR_DATA_PRESERVED;
245     hadc1.Init.OversamplingMode = DISABLE;
246     if (HAL_ADC_Init(&hadc1) != HAL_OK)
247     {
248         Error_Handler();
249     }
250     /** Configure the ADC multi mode
```

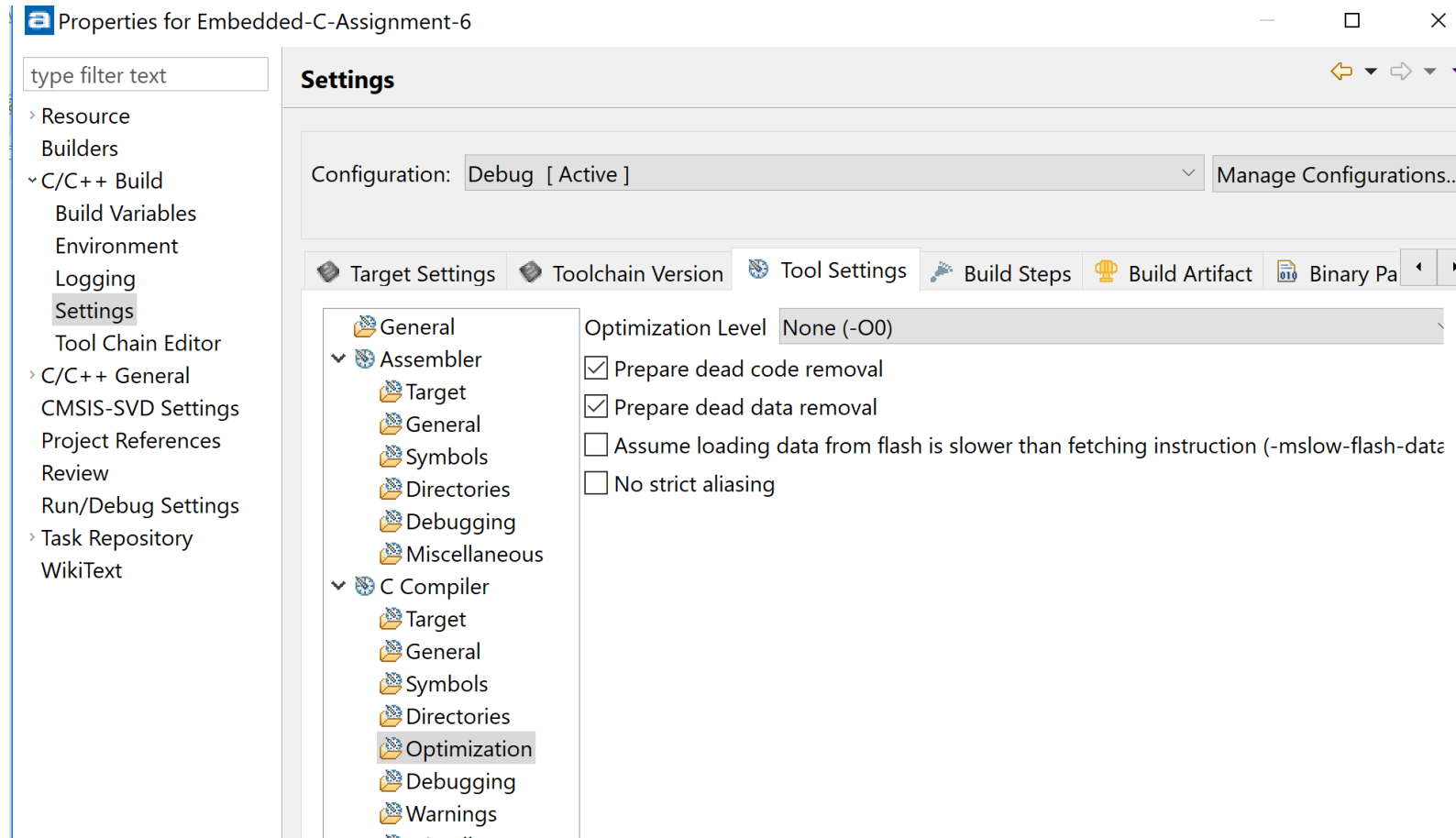

Step 16. In main.c, find code that initializes the Channel IN14

```
257  /** Configure Regular Channel
258  */
259  sConfig.Channel = ADC_CHANNEL_14;
260  sConfig.Rank = ADC_REGULAR_RANK_1;
261  sConfig.SamplingTime = ADC_SAMPLETIME_2CYCLES_5;
262  sConfig.SingleDiff = ADC_SINGLE_ENDED;
263  sConfig.OffsetNumber = ADC_OFFSET_NONE;
264  sConfig.Offset = 0;
265  if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK)
266  {
267      Error_Handler();
268  }
269  /* USER CODE BEGIN ADC1_Init 2 */
270
271  /* USER CODE END ADC1_Init 2 */
272
273 }
```

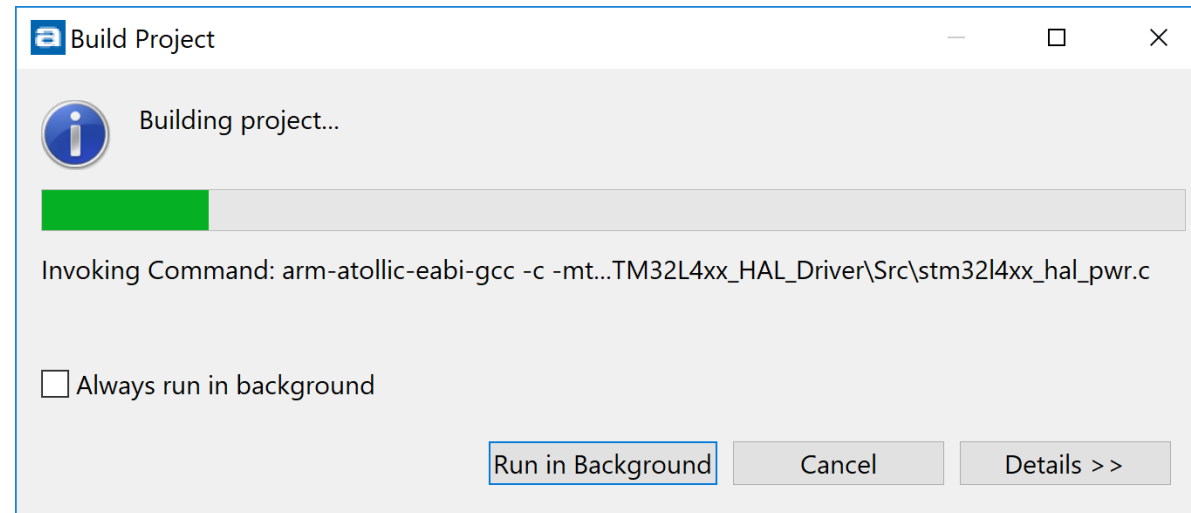
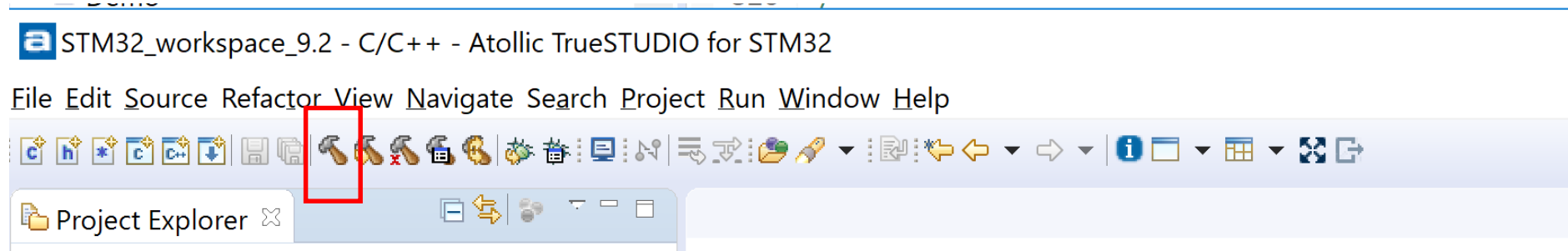
Step 17. Add code to main.c to read from ARD_A0

```
main.c
127  /* USER CODE BEGIN WHILE */
128  uint32_t adcResult;
129
130  while (1)
131  {
132      HAL_ADC_Start(&hadc1);
133
134      HAL_ADC_PollForConversion(&hadc1, 100);
135      adcResult = HAL_ADC_GetValue(&hadc1);
136      printf("adcResult: %lu - 0x%lx\n", adcResult, adcResult);
137
138      HAL_ADC_Stop(&hadc1);
139
140      HAL_Delay(1000);
141  }
142  /* USER CODE END WHILE */
143
144
145  /* USER CODE BEGIN 3 */
146  }
147  /* USER CODE END 3 */
148 }
```

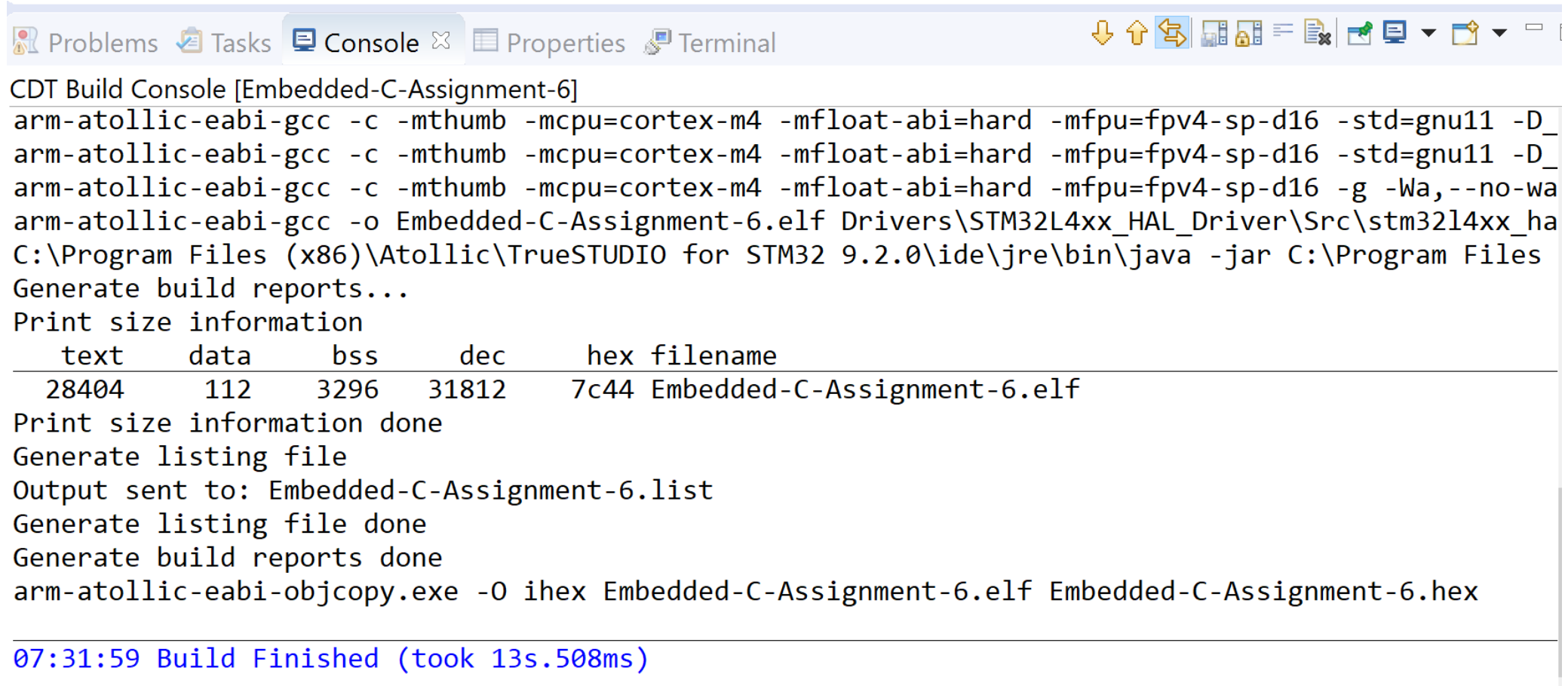
Step 18. Set optimization to none



Step 19. Build Project



Step 20. Results of Build – Part 1



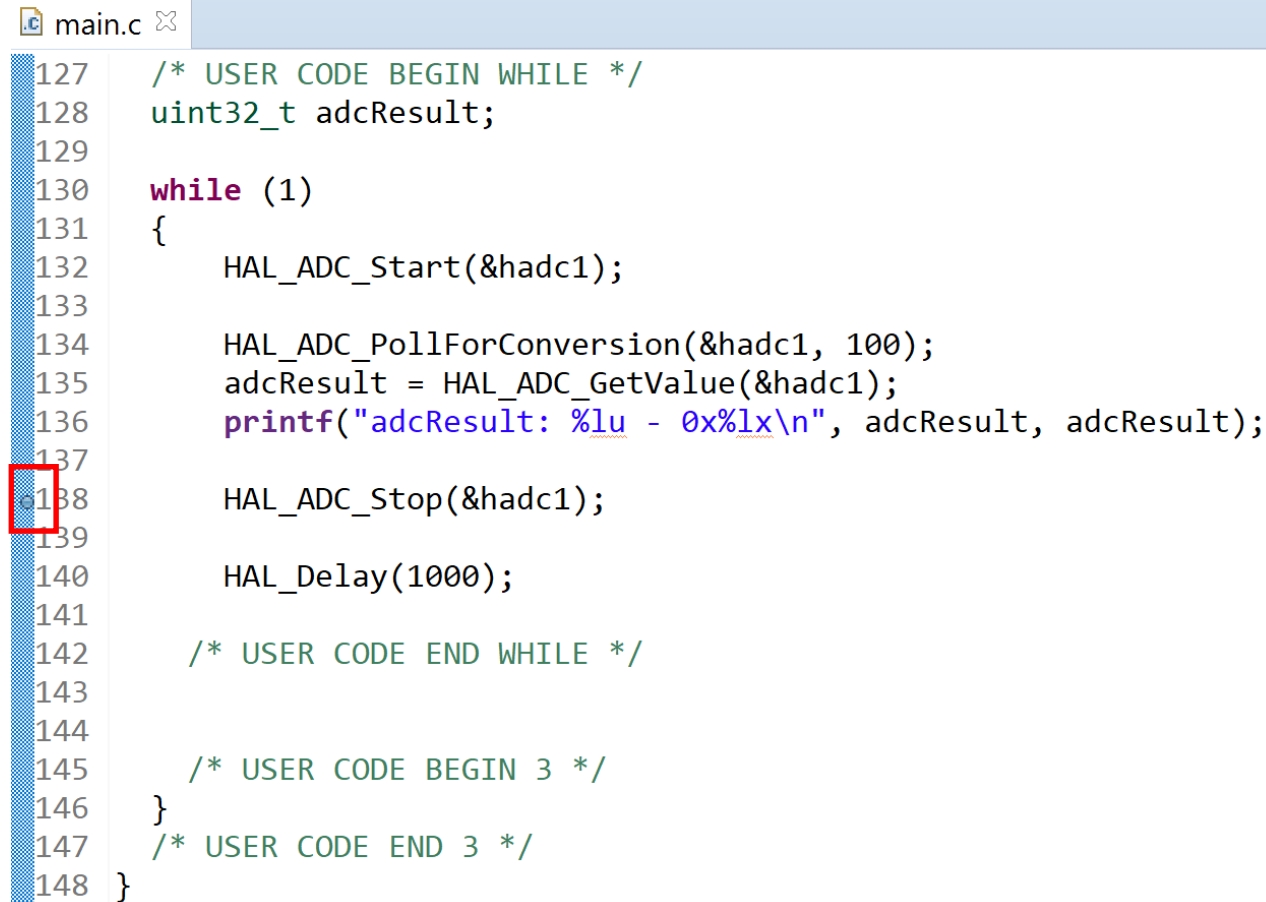
The screenshot shows the CDT Build Console window for a project named "Embedded-C-Assignment-6". The console displays the output of the build process, including the compilation of source files, the generation of build reports, and the final build completion message.

```
CDT Build Console [Embedded-C-Assignment-6]
arm-atollic-eabi-gcc -c -mthumb -mcpu=cortex-m4 -mfloat-abi=hard -mfpv4-sp-d16 -std=gnu11 -D_
arm-atollic-eabi-gcc -c -mthumb -mcpu=cortex-m4 -mfloat-abi=hard -mfpv4-sp-d16 -std=gnu11 -D_
arm-atollic-eabi-gcc -c -mthumb -mcpu=cortex-m4 -mfloat-abi=hard -mfpv4-sp-d16 -g -Wa,--no-wa
arm-atollic-eabi-gcc -o Embedded-C-Assignment-6.elf Drivers\STM32L4xx_HAL_Driver\Src\stm32l4xx_ha
C:\Program Files (x86)\Atollic\TrueSTUDIO for STM32 9.2.0\ide\jre\bin\java -jar C:\Program Files
Generate build reports...
Print size information
  text    data    bss    dec    hex filename
 28404    112    3296   31812   7c44 Embedded-C-Assignment-6.elf
Print size information done
Generate listing file
Output sent to: Embedded-C-Assignment-6.list
Generate listing file done
Generate build reports done
arm-atollic-eabi-objcopy.exe -O ihex Embedded-C-Assignment-6.elf Embedded-C-Assignment-6.hex

07:31:59 Build Finished (took 13s.508ms)
```

Step 21. Set Breakpoint

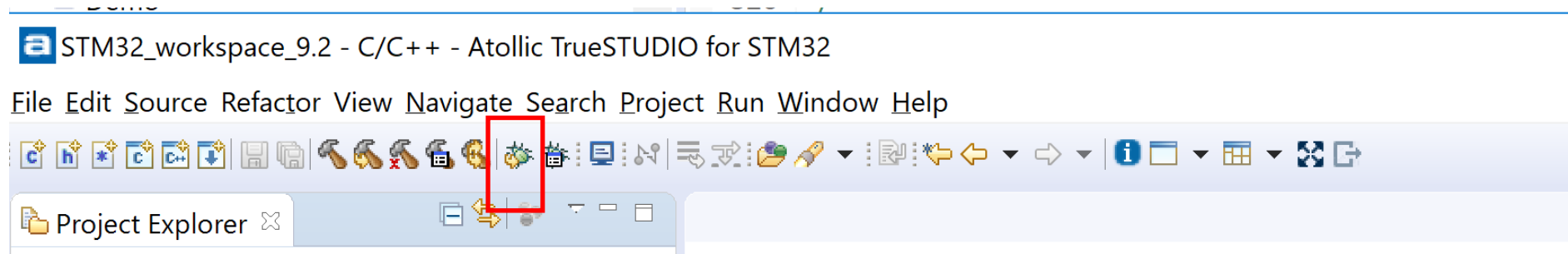
Double-click
Here to
Add “green”
breakpoint



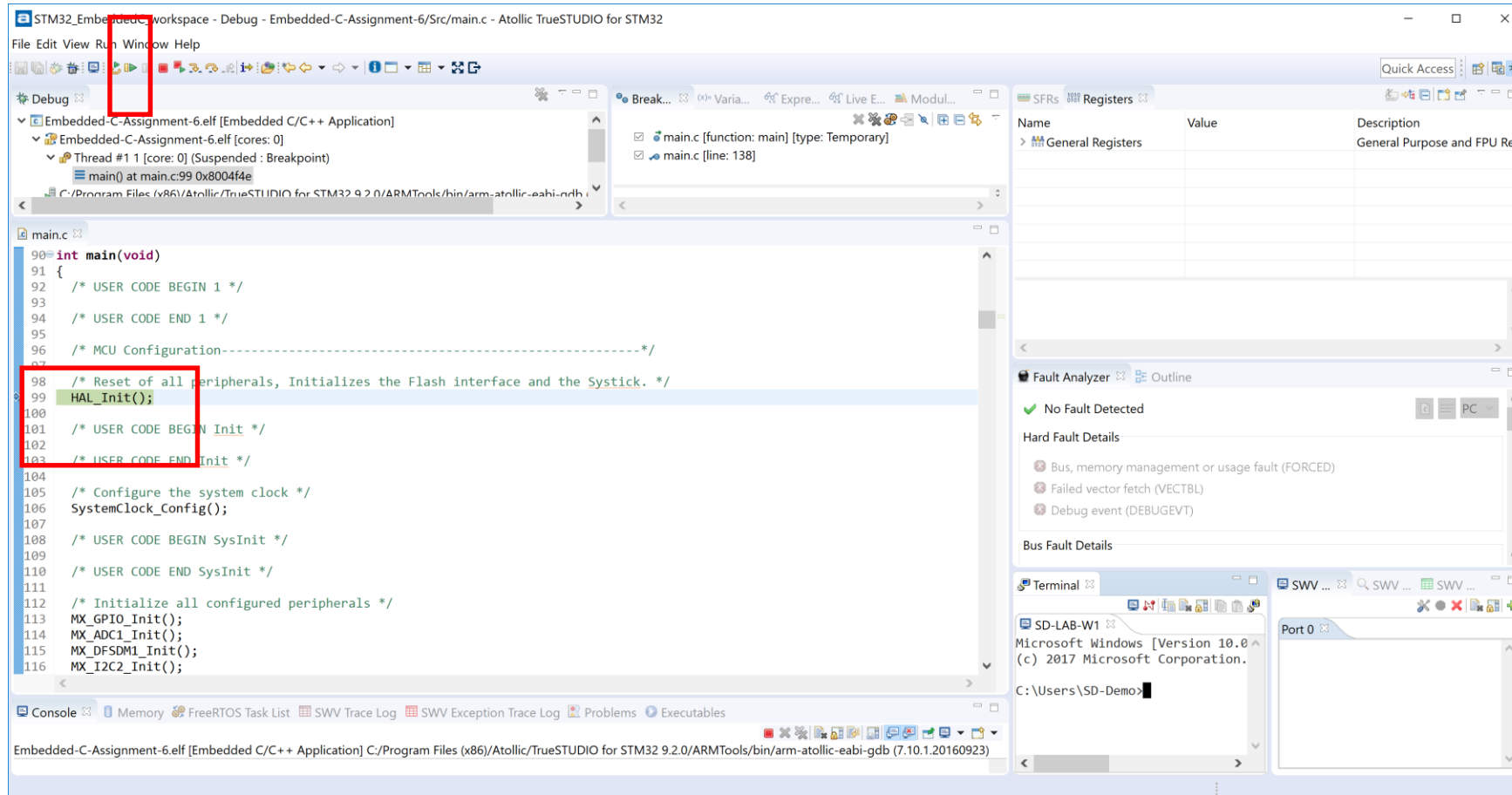
The screenshot shows a code editor window titled 'main.c'. The code is a C program with a while loop. A red square highlights the line number '138' in the left margin, indicating where a breakpoint has been set. The code is as follows:

```
127  /* USER CODE BEGIN WHILE */
128  uint32_t adcResult;
129
130  while (1)
131  {
132      HAL_ADC_Start(&hadc1);
133
134      HAL_ADC_PollForConversion(&hadc1, 100);
135      adcResult = HAL_ADC_GetValue(&hadc1);
136      printf("adcResult: %lu - 0x%lx\n", adcResult, adcResult);
137
138      HAL_ADC_Stop(&hadc1);
139
140      HAL_Delay(1000);
141
142      /* USER CODE END WHILE */
143
144
145      /* USER CODE BEGIN 3 */
146  }
147  /* USER CODE END 3 */
148 }
```

Step 22. Run in Debug



Step 23. Hit Breakpoint, then click Resume



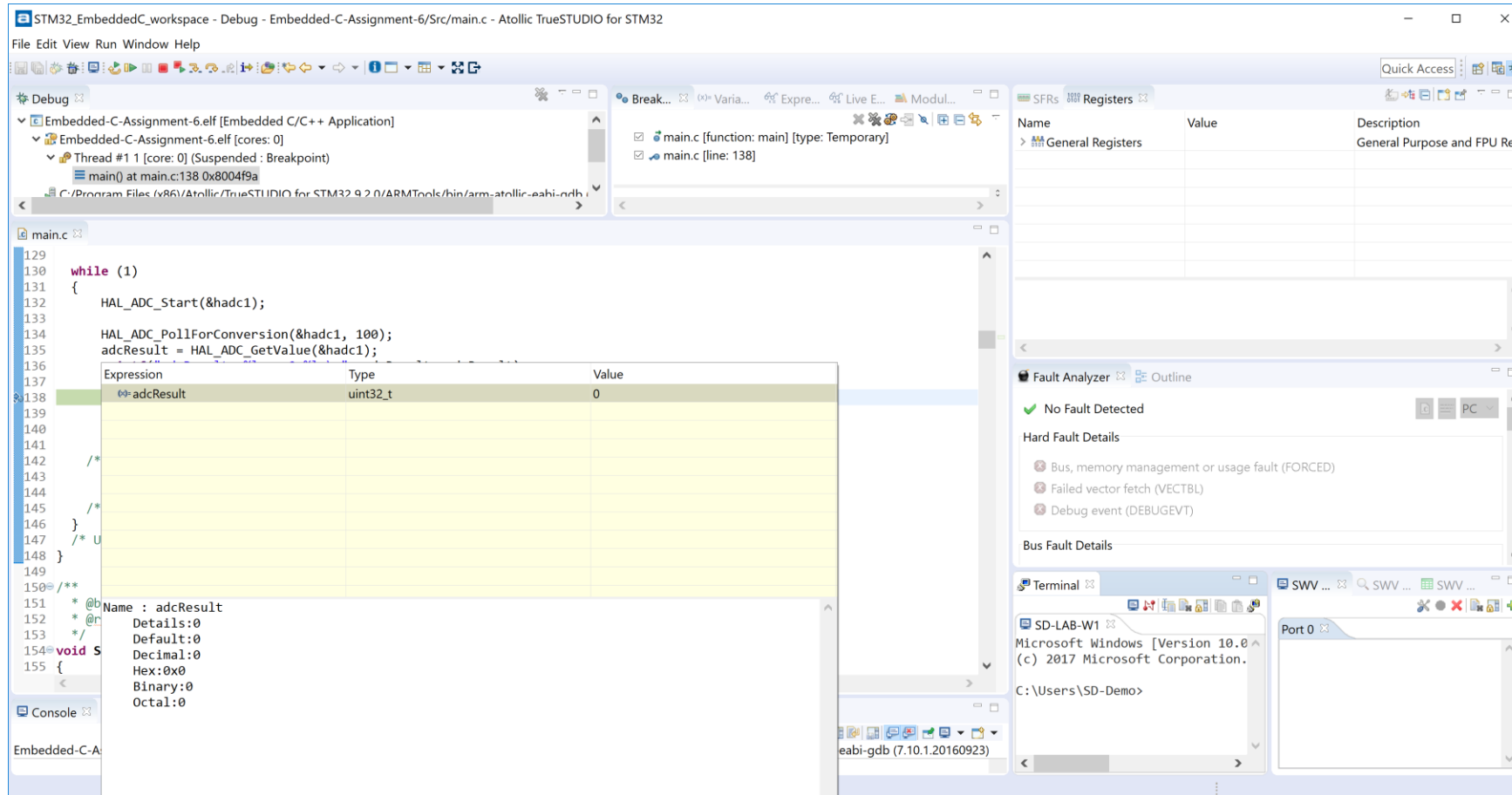
Step 24. Read value when nothing connected to ARD_A0 – In my case adcResult = 916

The screenshot displays the Atollic TrueSTUDIO IDE interface during a debug session. The main.c file is open, showing a while loop that reads the ADC value. The Variable Watcher window is open, showing the variable `adcResult` with a value of 916, which is highlighted with a red box. The console window shows the variable's details:

```
Name : adcResult
Details: 916
Default: 916
Decimal: 916
Hex: 0x394
Binary: 1110010100
Octal: 01624
```

The right-hand side of the IDE shows the SFRs and Registers window, the Fault Analyzer (No Fault Detected), and the Terminal window (C:\Users\SD-Demo>).

Step 24. Connect ADR_A0 to GND (two pins over).
Then read value of ARD_A0 – In my case adcResult
= 0



Step 25. Connect ADR_A0 to 1.5VDC. Then read value of ARD_A0 – In my case adcResult = 1845

The screenshot shows the Atollic TrueSTUDIO for STM32 interface during a debug session. The main editor displays the following C code in `main.c`:

```
129
130 while (1)
131 {
132     HAL_ADC_Start(&hadc1);
133
134     HAL_ADC_PollForConversion(&hadc1, 100);
135     adcResult = HAL_ADC_GetValue(&hadc1);
136
137     Expression      Type      Value
138     -----
139     adcResult      uint32_t    1845
140
141
142     /*
143
144     /*
145     */
146     */
147     */
148 }
149
150 /**
151 * @b Name : adcResult
152 * @b Details:1845
153 * @b Default:1845
154 * @b Decimal:1845
155 * @b Hex:0x735
156 * @b Binary:11100110101
157 * @b Octal:03465
```

The variable watch window shows the variable `adcResult` of type `uint32_t` with a value of `1845`.

The console window displays the variable details:

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

The right sidebar contains the following panels:

- SFRs and Registers:** A table with columns for Name, Value, and Description. The 'General Registers' section is visible.
- Fault Analyzer:** Shows 'No Fault Detected' and lists 'Hard Fault Details' (Bus, memory management or usage fault (FORCED), Failed vector fetch (VECTBL), Debug event (DEBUGEVT)) and 'Bus Fault Details'.
- Terminal:** Shows the command prompt for 'SD-LAB-W1' with the text 'Microsoft Windows [Version 10.0 (c) 2017 Microsoft Corporation. C:\Users\SD-Demo>'. The 'Port 0' window is also visible.