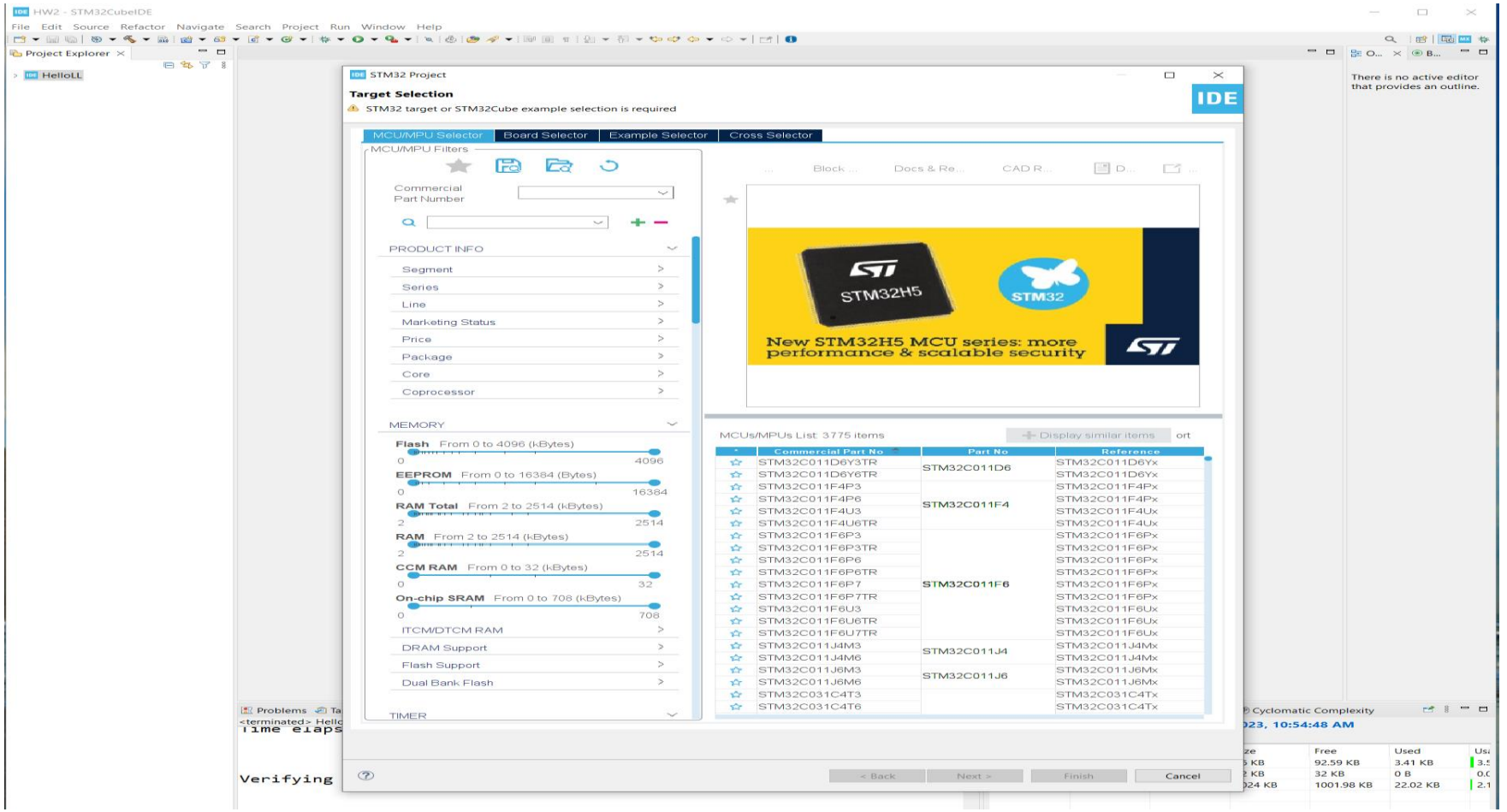
UCSD Embedded RTOS Assignment 1

By

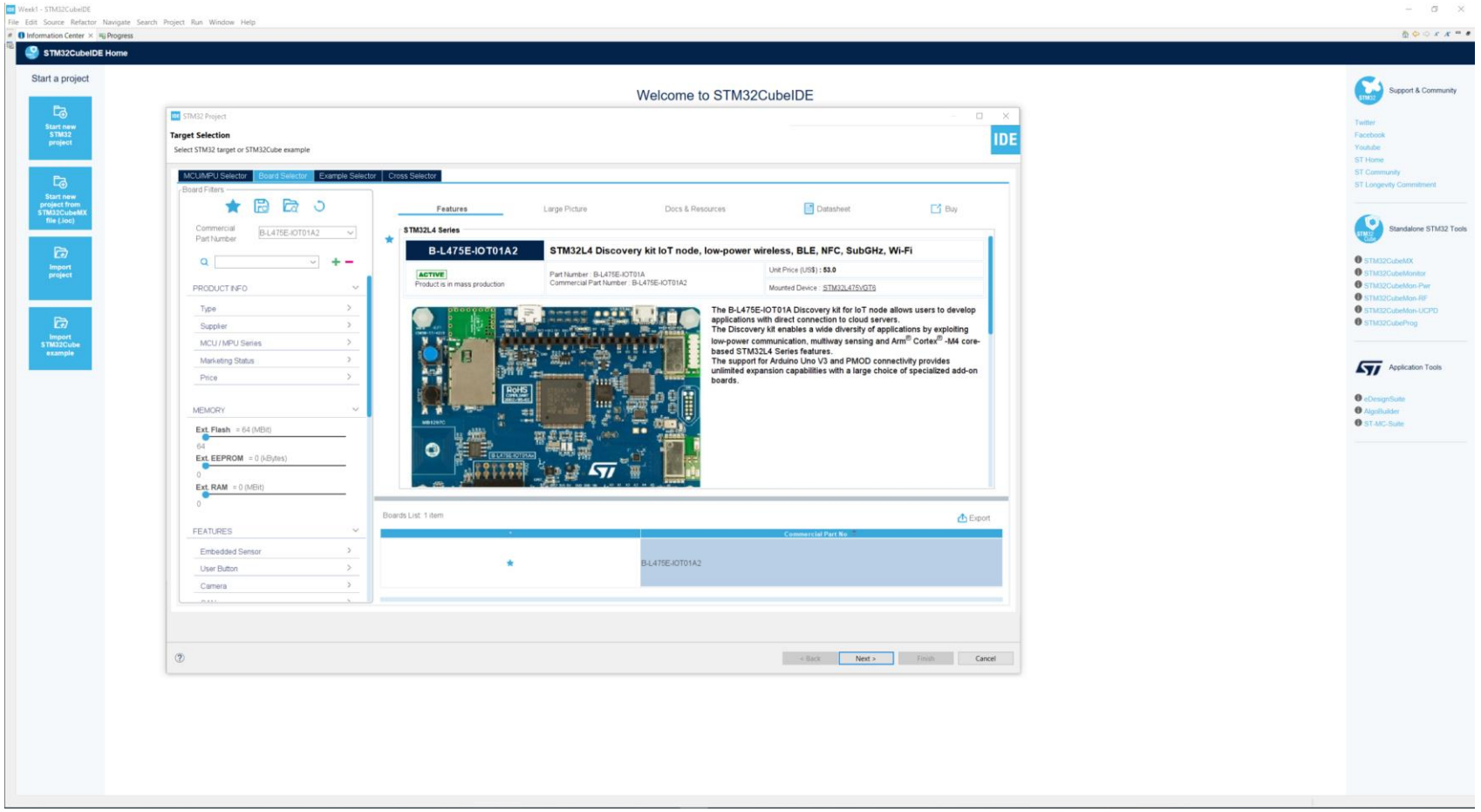
Hsuankai Chang

hsuankac@umich.edu

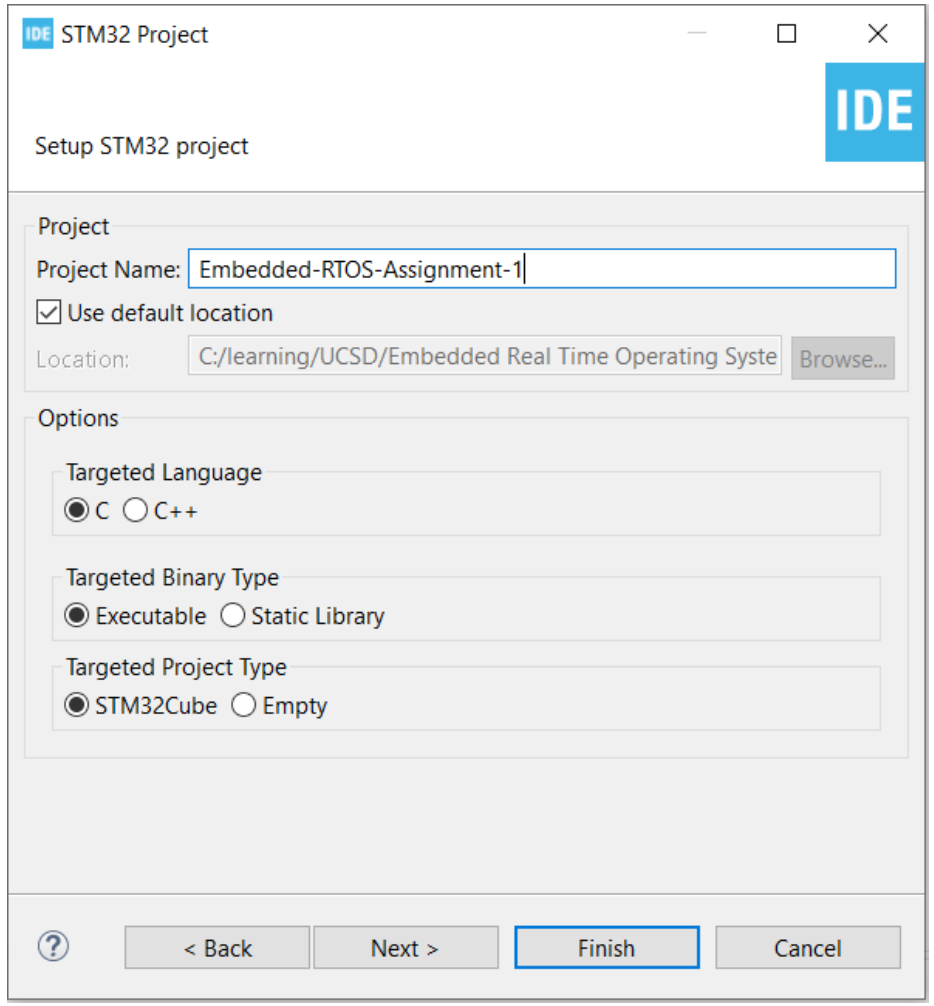
Step 1. Startup STM32CubeIDE and create new STM32 project



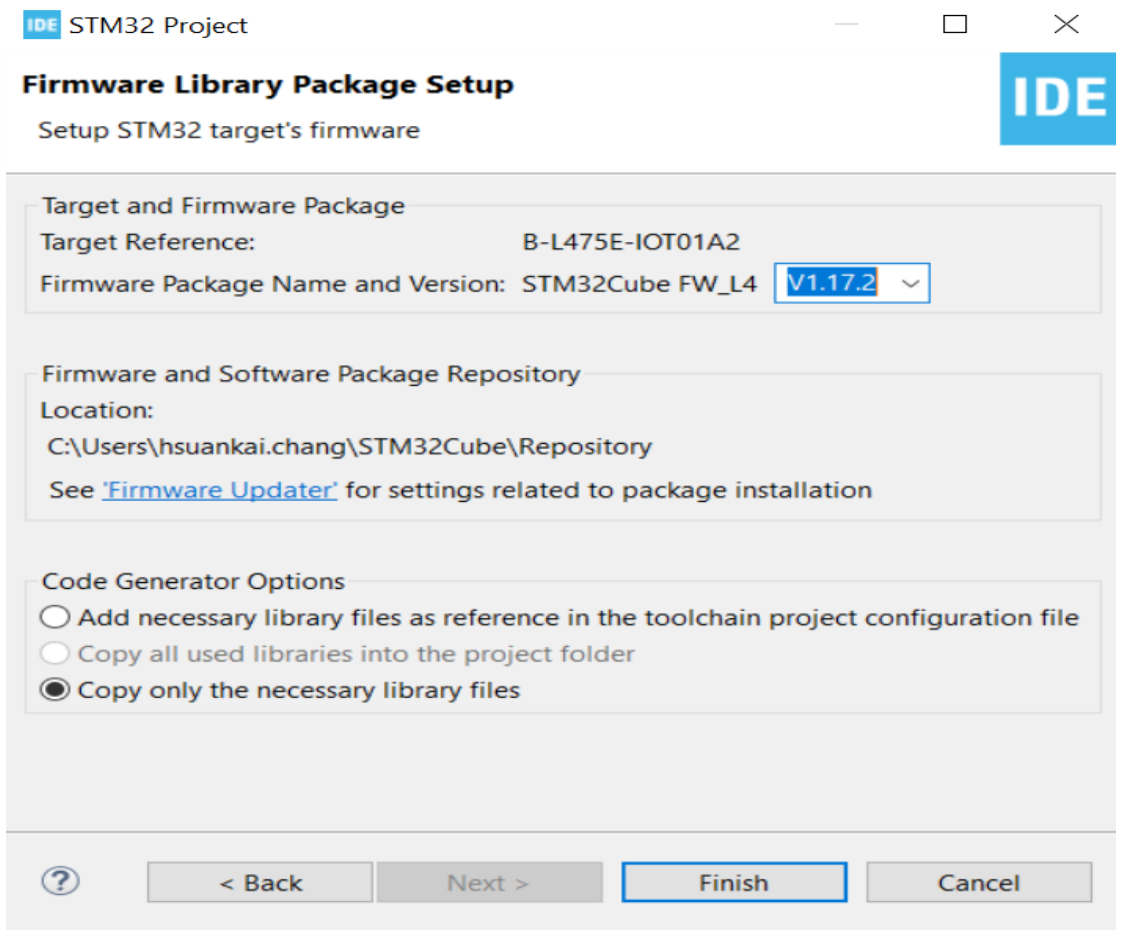
Step 2. Access board selector and type in the board you use, click Next



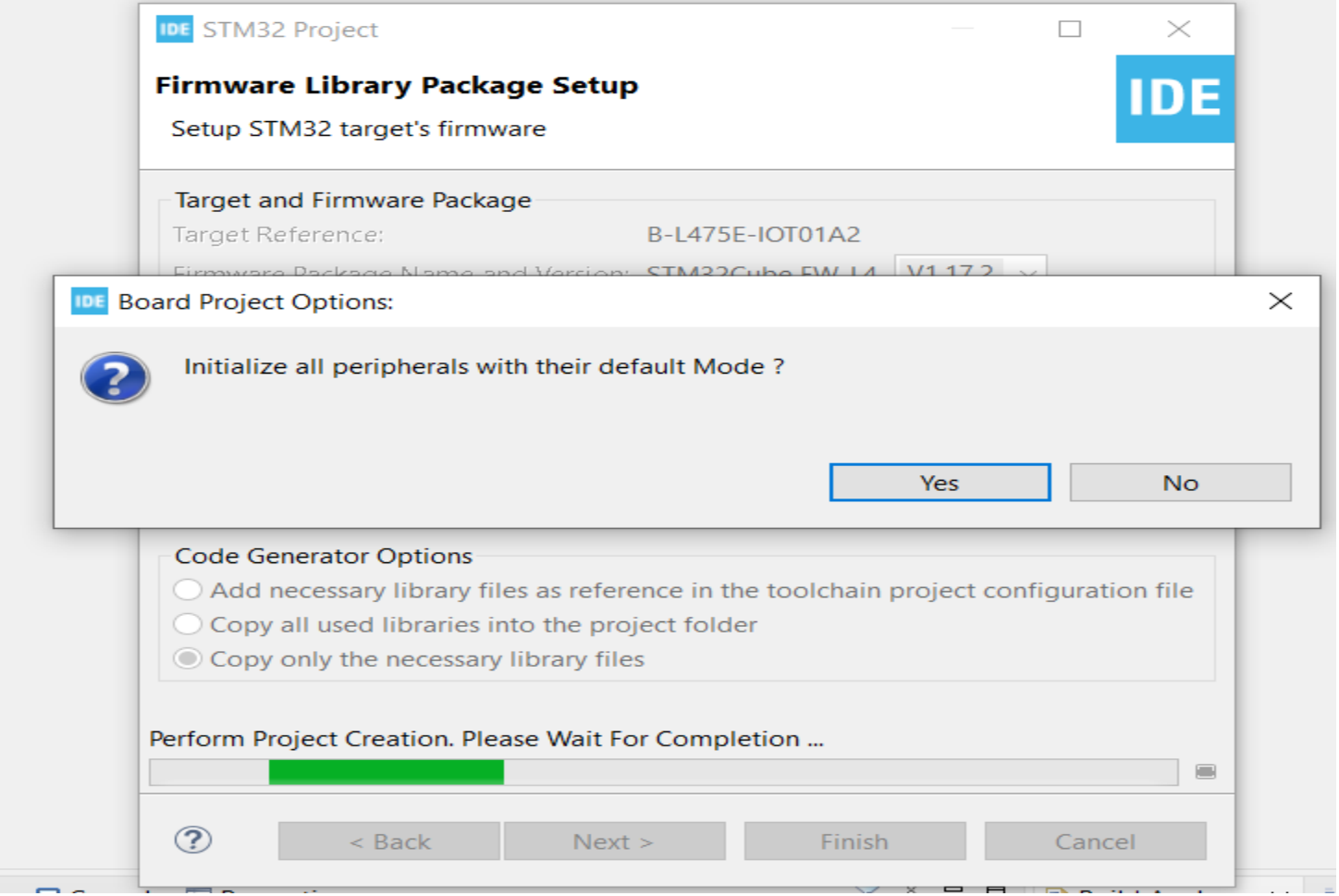
Step 3. Enter the project name then click Next



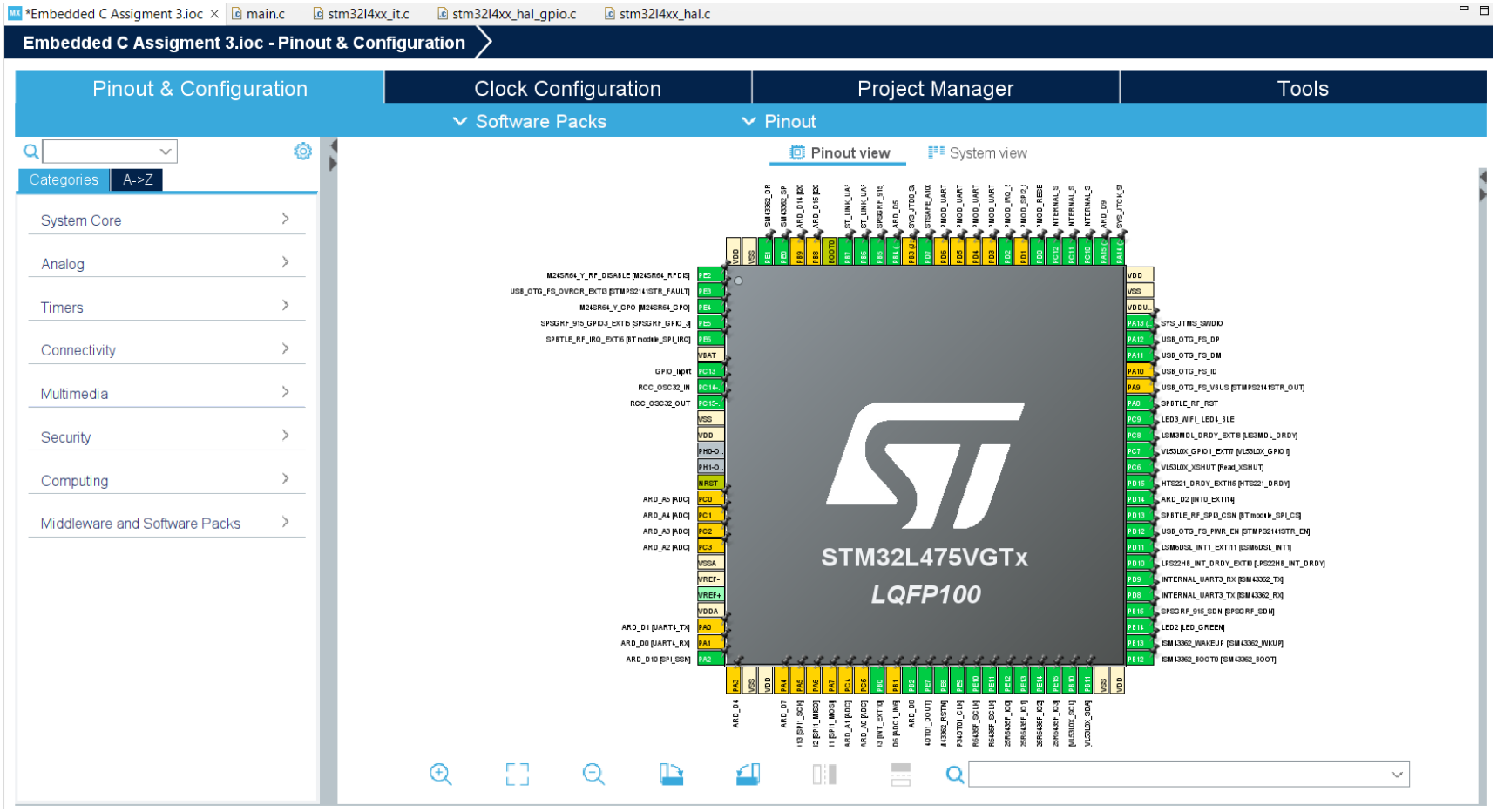
Step 4. See the firmware package name and version



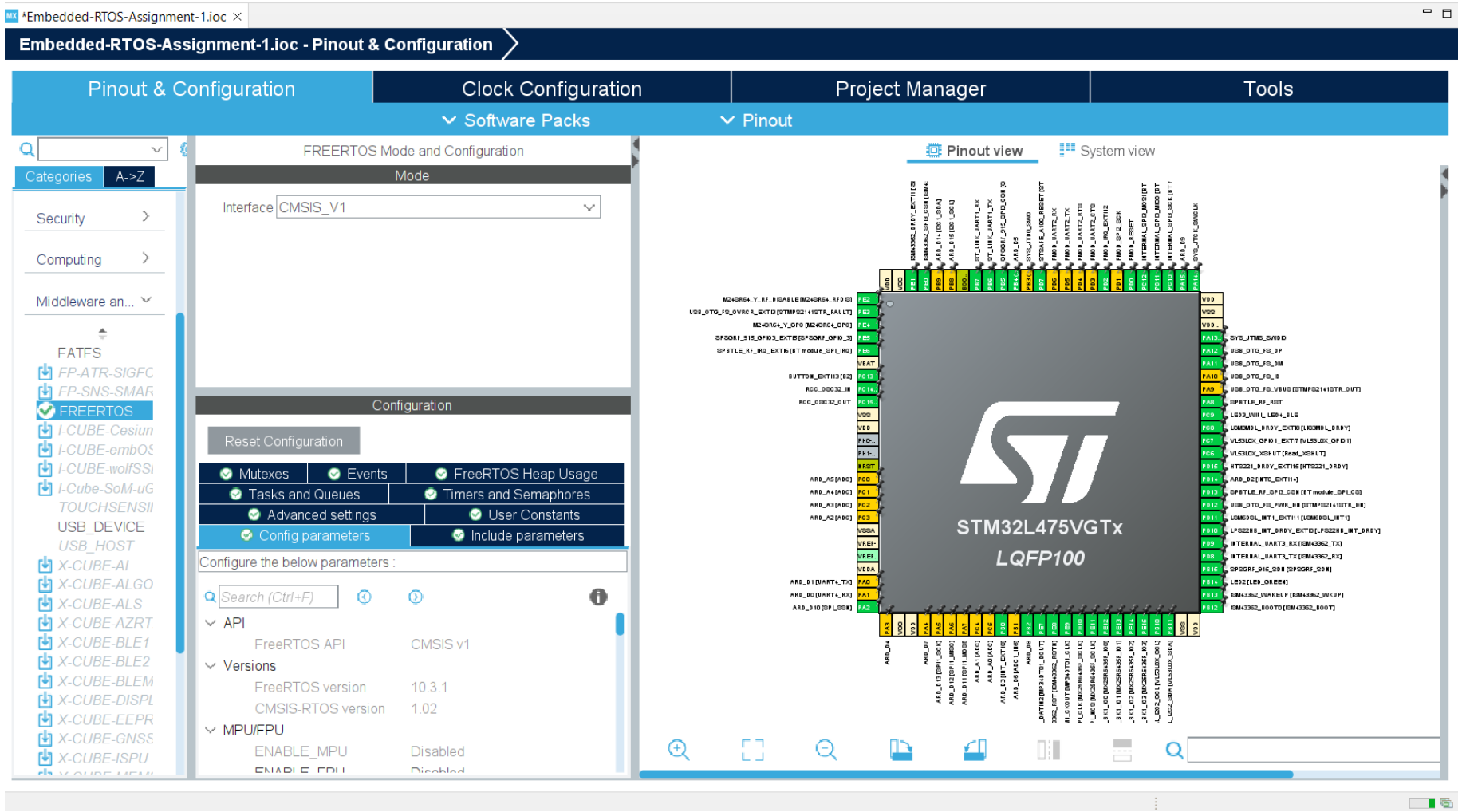
Step 5. Click yes to initialize all peripherals to default



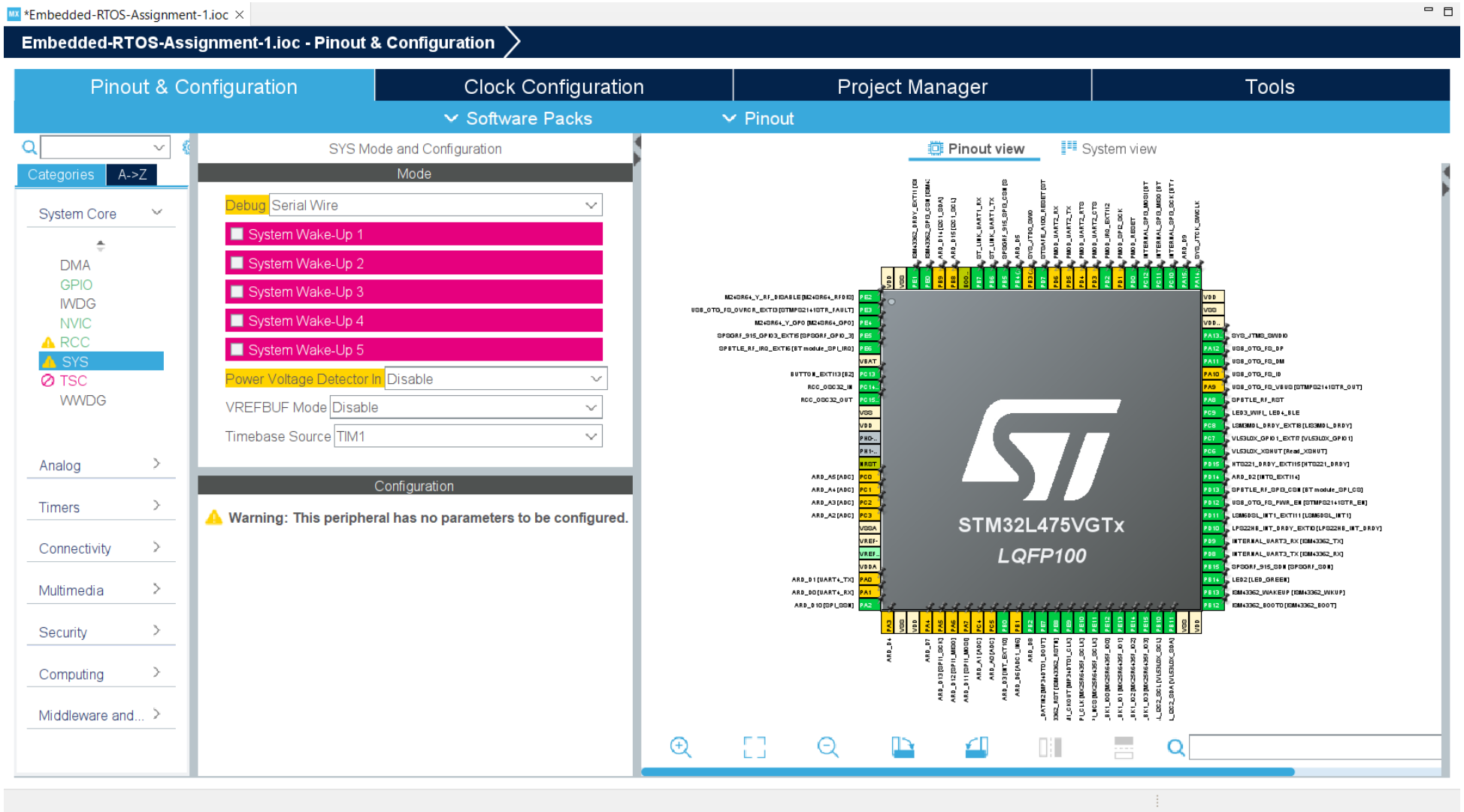
Step 6. When in .ioc file, click Pinout & Configurations



Step 7. Enable the FreeRTOS in middleware, leave all the settings as default



Step 8. Change Timebase from systick to TIM1



Step 9. Generate the code and add the LED2 toggle in default task

Embedded-RTOS-Assignment-1.ioc | main.c

```
677
678 /* USER CODE END 4 */
679
680 /* USER CODE BEGIN Header_StartDefaultTask */
681 /**
682   * @brief  Function implementing the defaultTask thread.
683   * @param  argument: Not used
684   * @retval None
685   */
686 /* USER CODE END Header_StartDefaultTask */
687 void StartDefaultTask(void const * argument)
688 {
689   /* USER CODE BEGIN 5 */
690   /* Infinite loop */
691   for(;;)
692   {
693       HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
694       osDelay(1000);
695   }
696   /* USER CODE END 5 */
697 }
698
699 /**
700   * @brief  Period elapsed callback in non blocking mode
701   * @note   This function is called  when TIM1 interrupt took place, inside
702   * HAL_TIM_IRQHandler(). It makes a direct call to HAL_IncTick() to increment
703   * a global variable "uwTick" used as application time base.
704   * @param  htim : TIM handle
705   * @retval None
706   */
707 void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
708 {
```

Step 10. Build and debug the code, you should see LED2 toggling