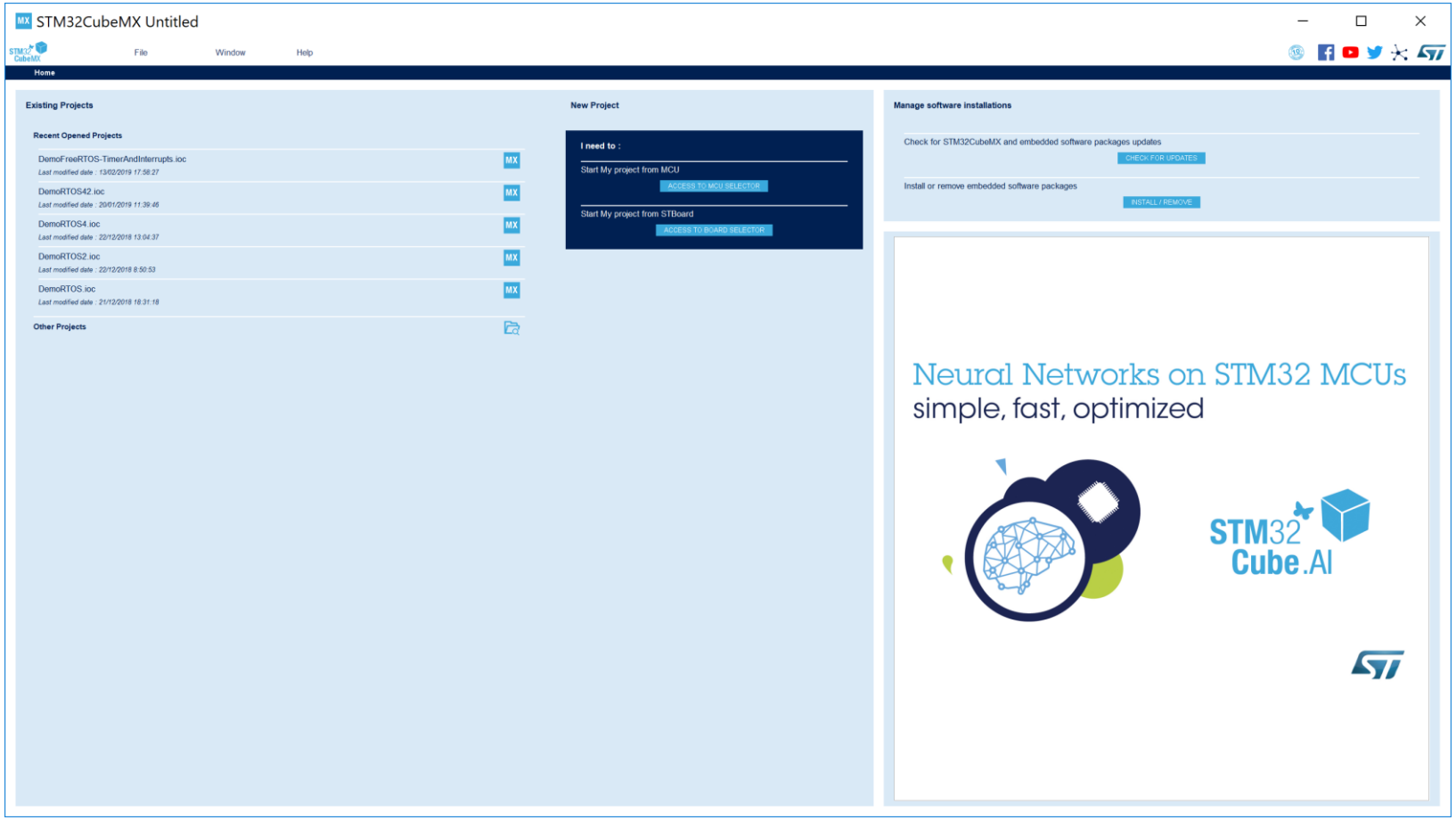
Embedded RTOS Assignment 7

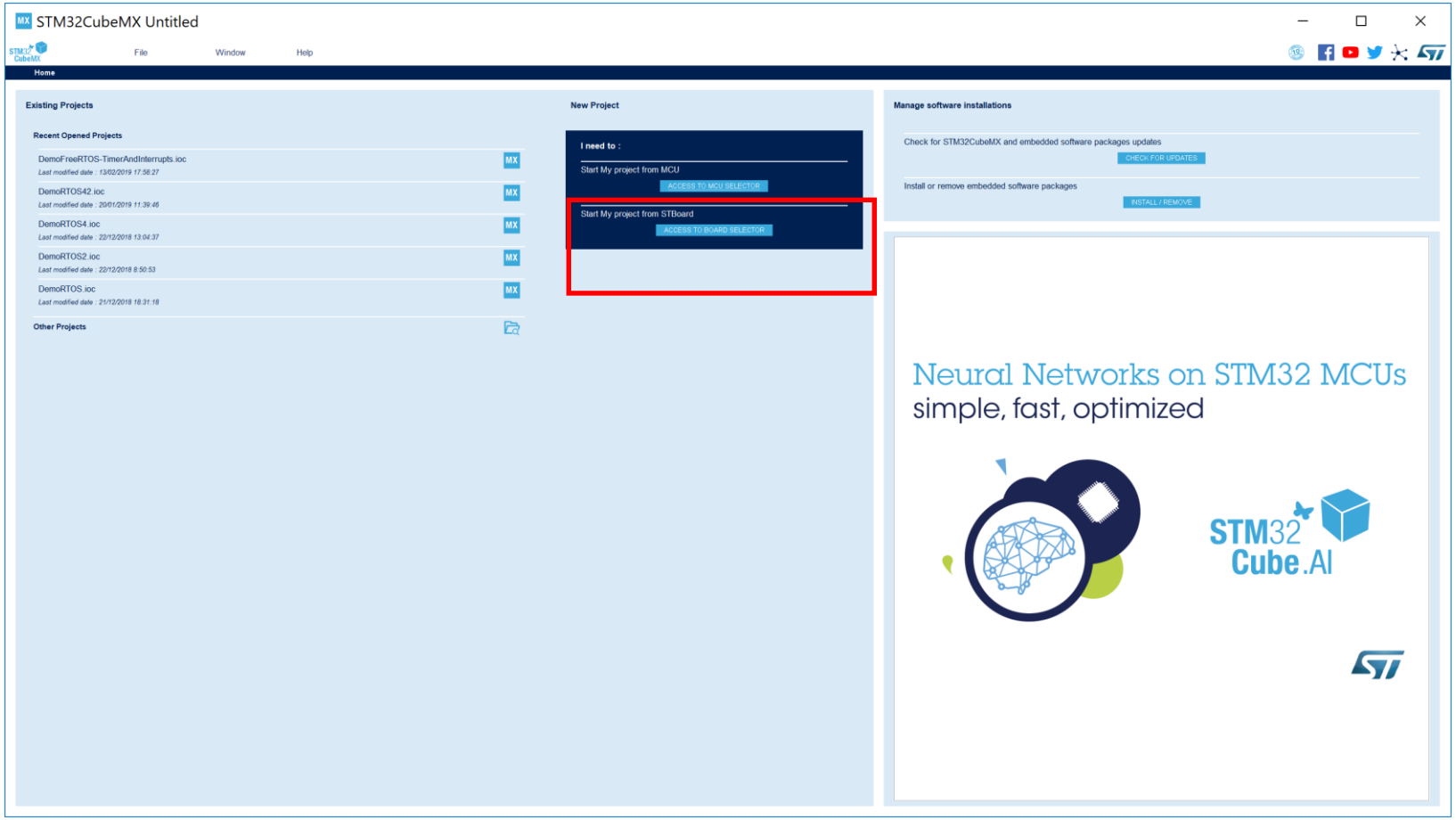
By
Norman McEntire

Norman.mcentire@gmail.com

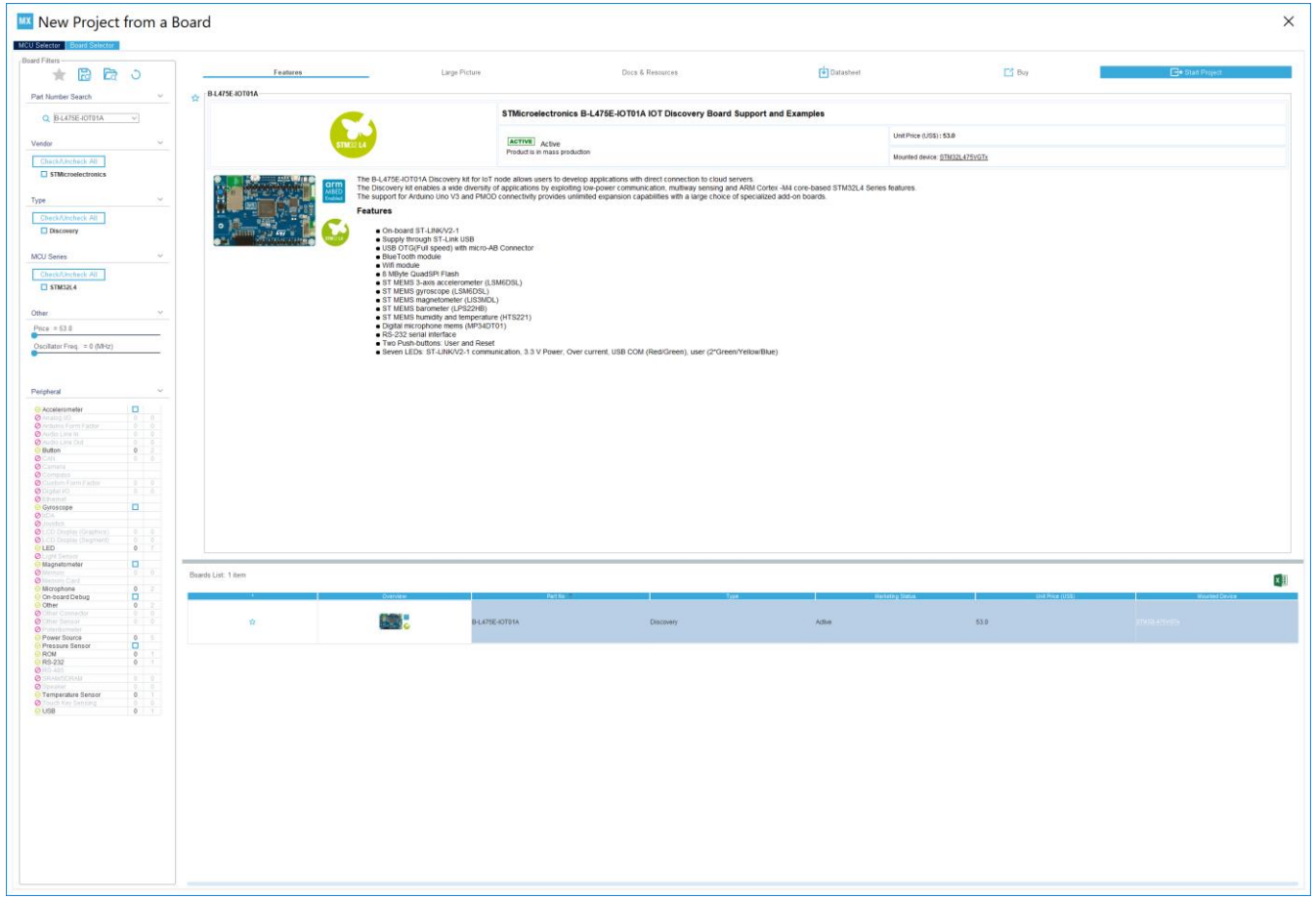
Step 1. Startup STM32CubeMX



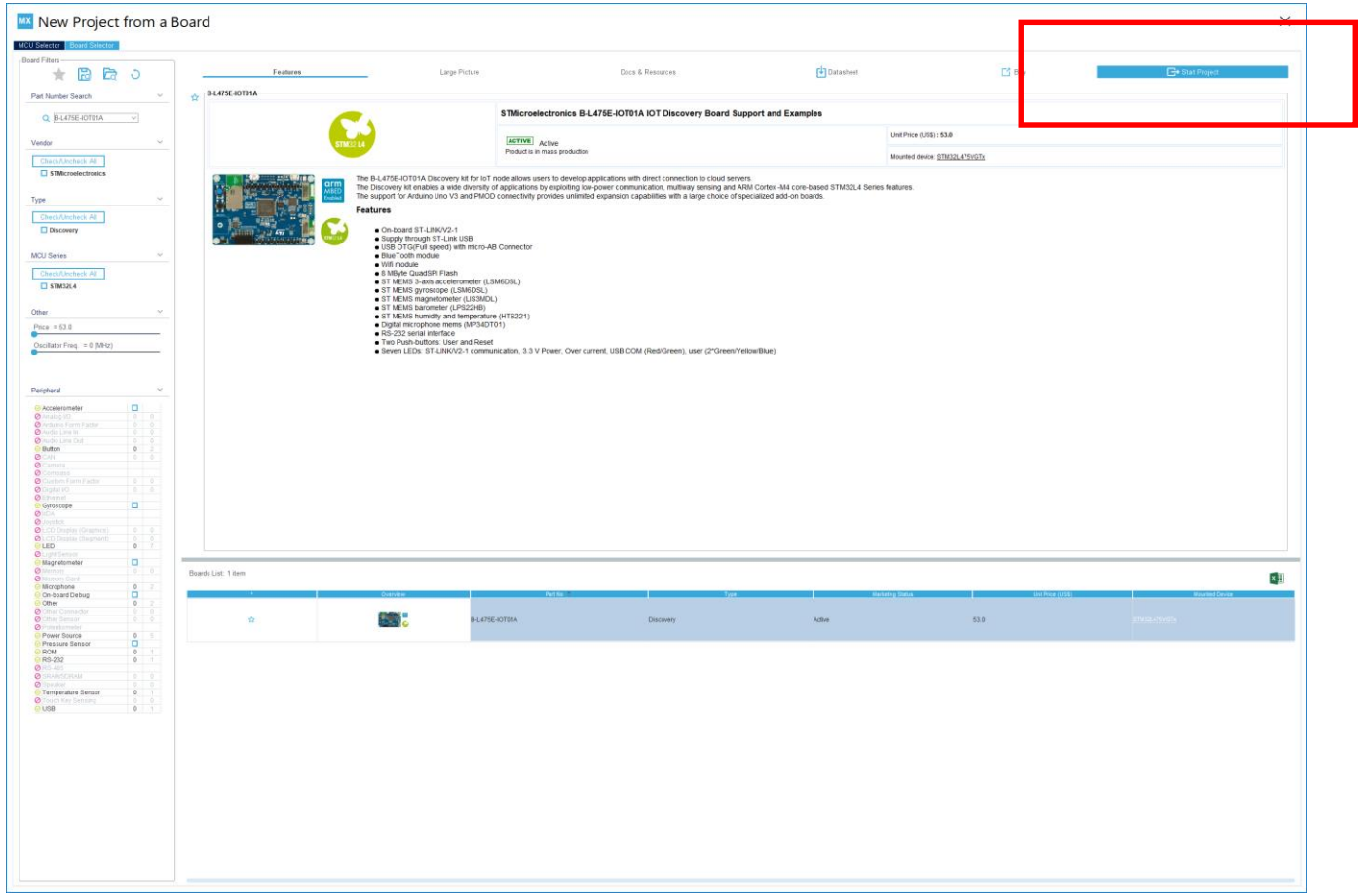
Step 2. Access Board Selector



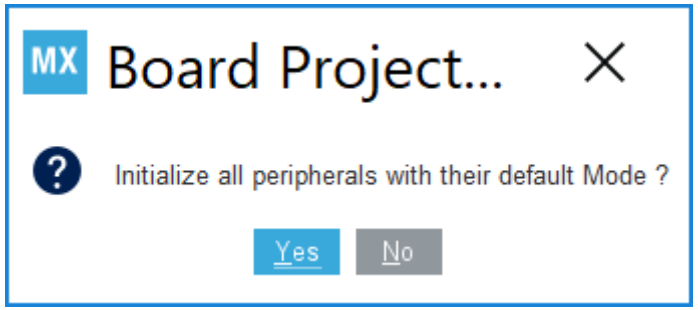
Step 3. Select “B-L475E-IOT01A” Board



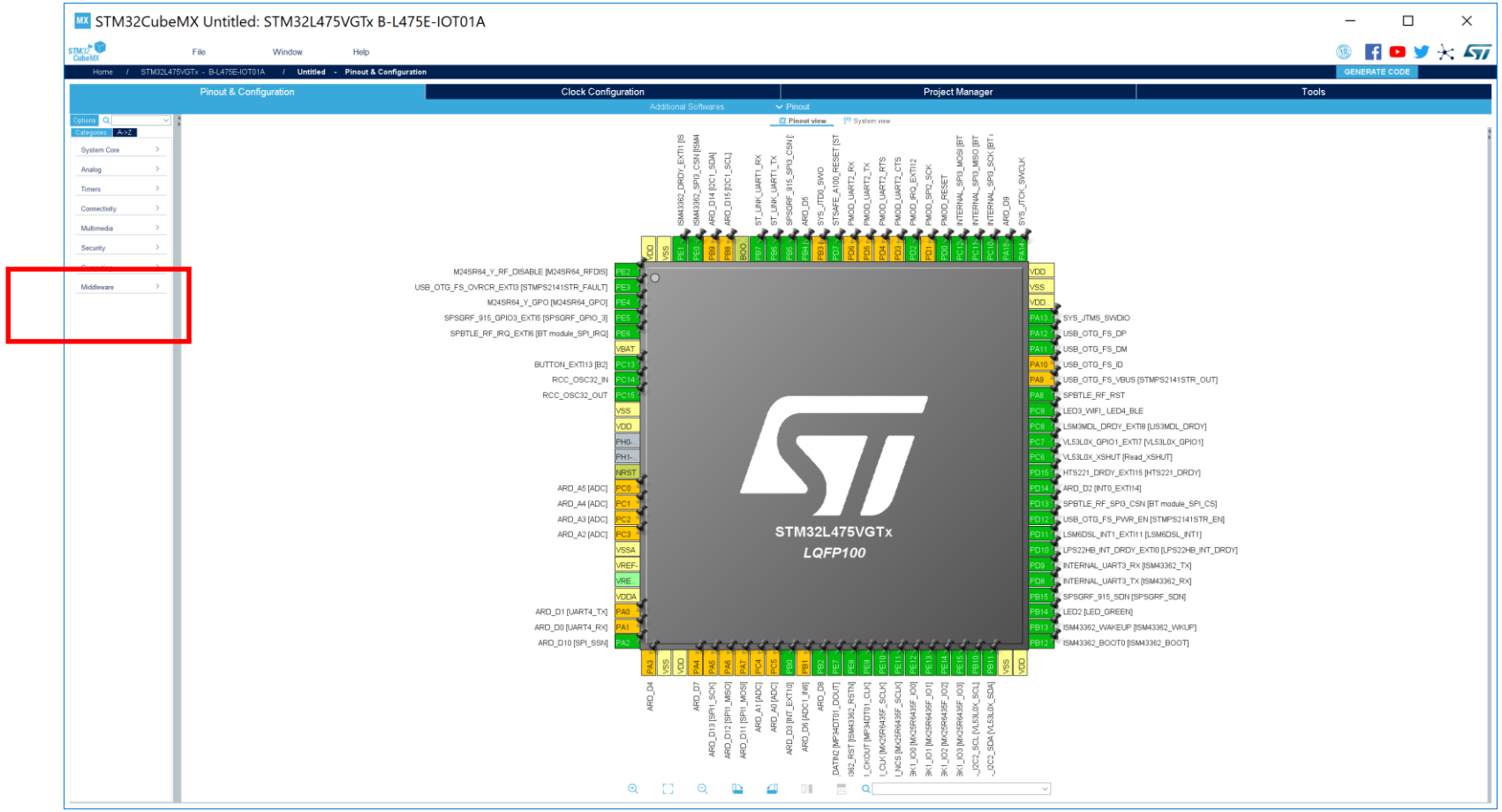
Step 4. Select “Start Project”



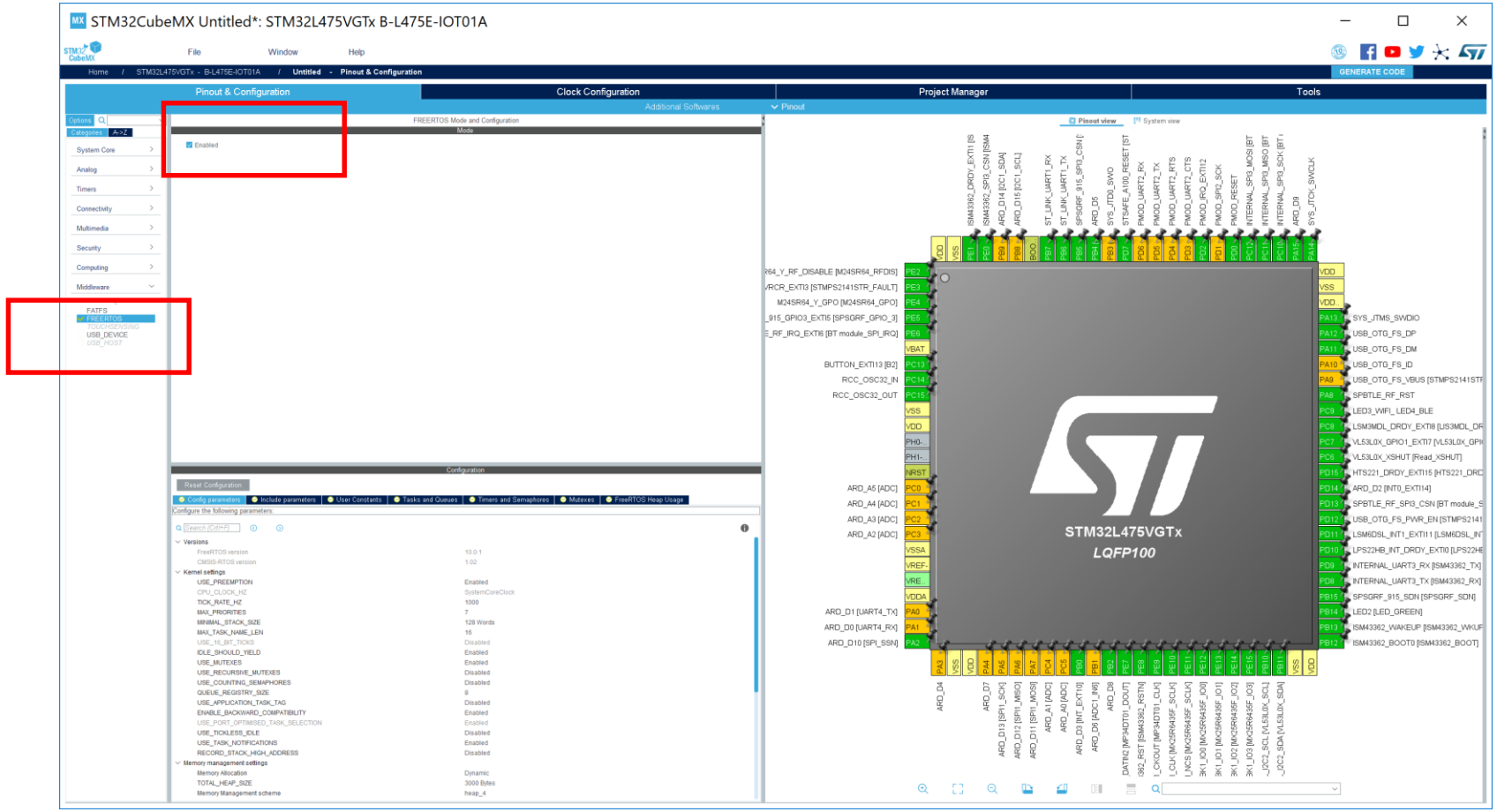
Step 5. Select YES: “Initialize all peripherals with their default Mode”



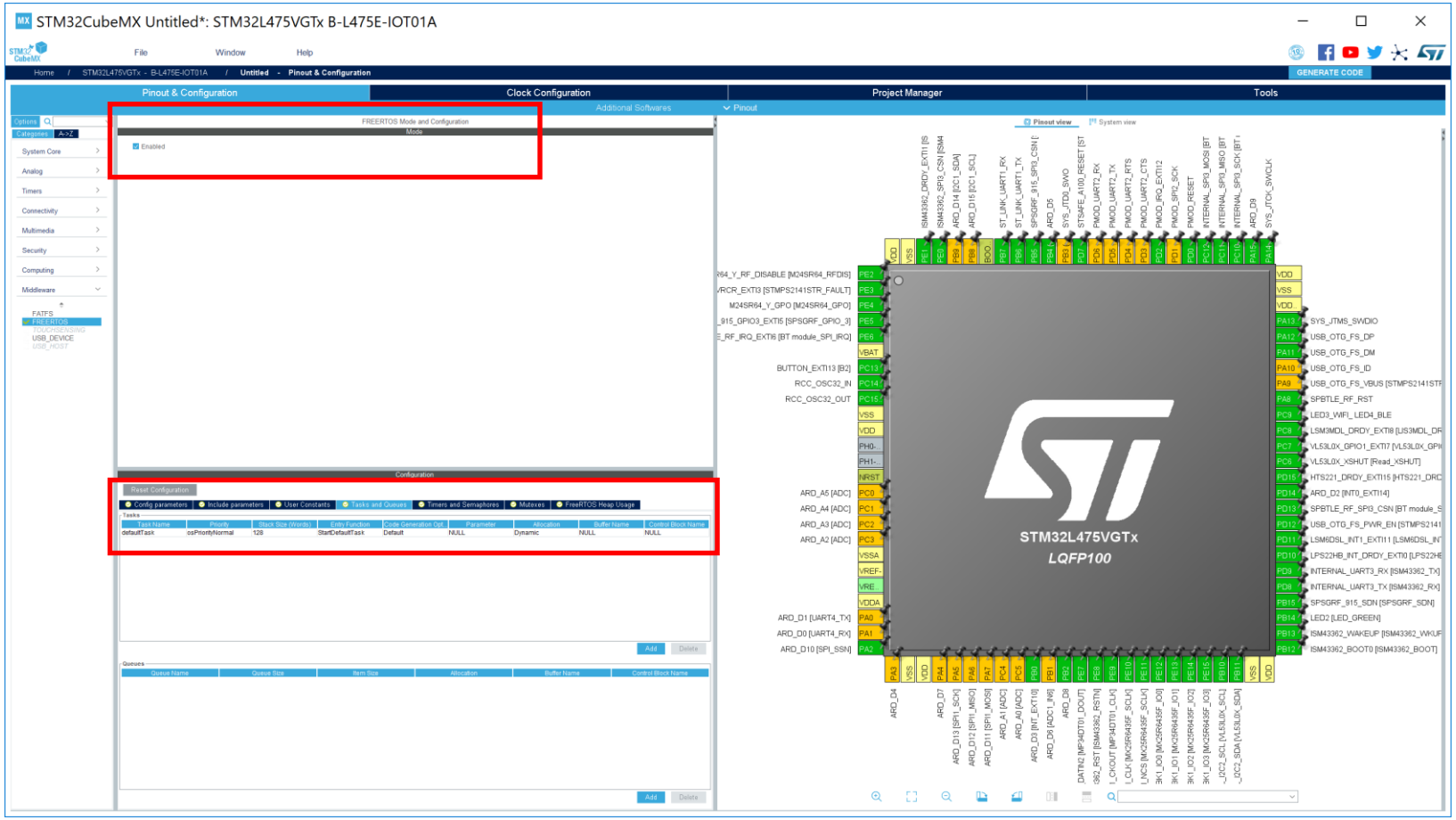
Step 6. Select “Middleware”



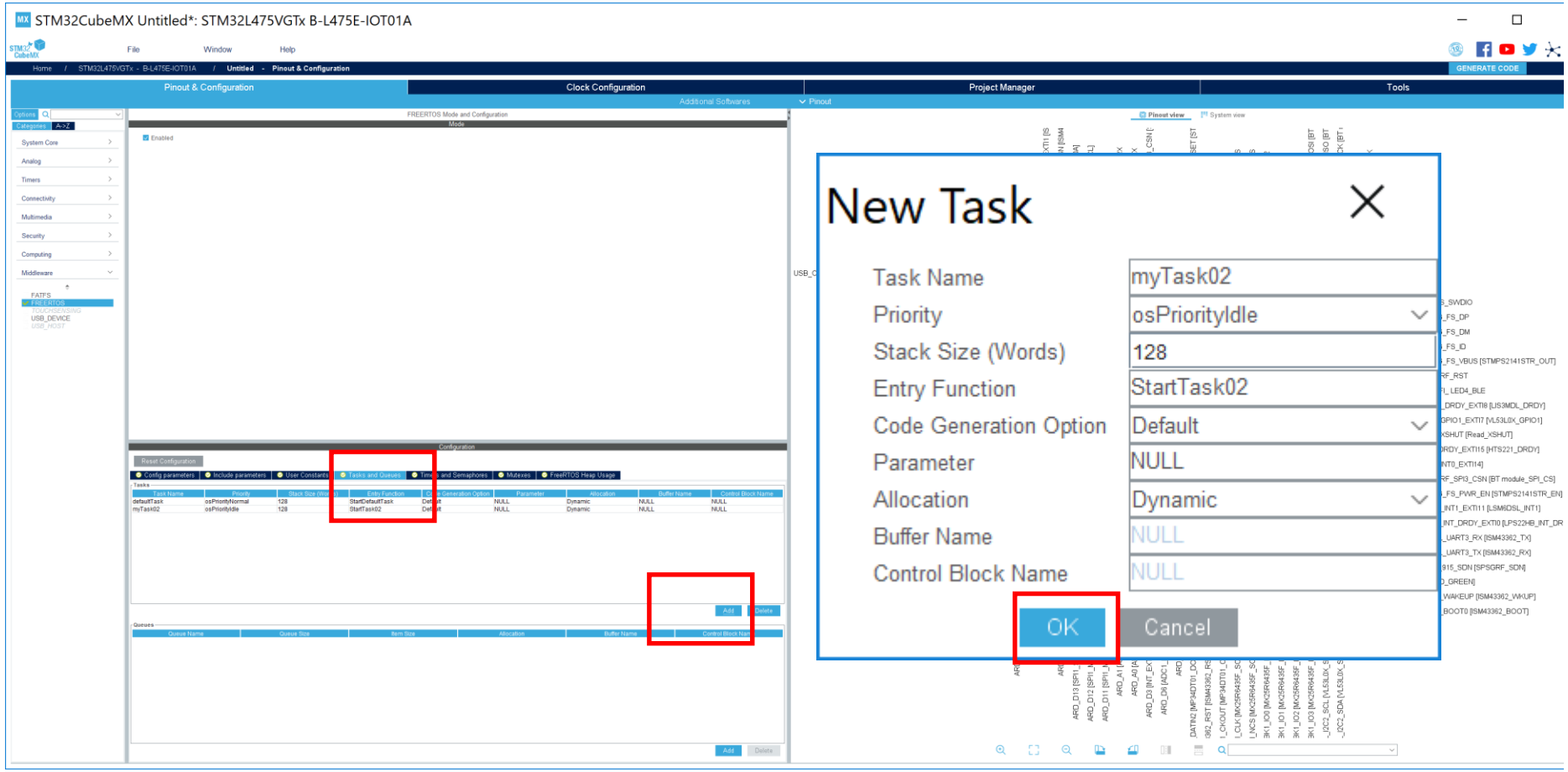
Step 7. Select “FreeRTOS”, then select “Enable”



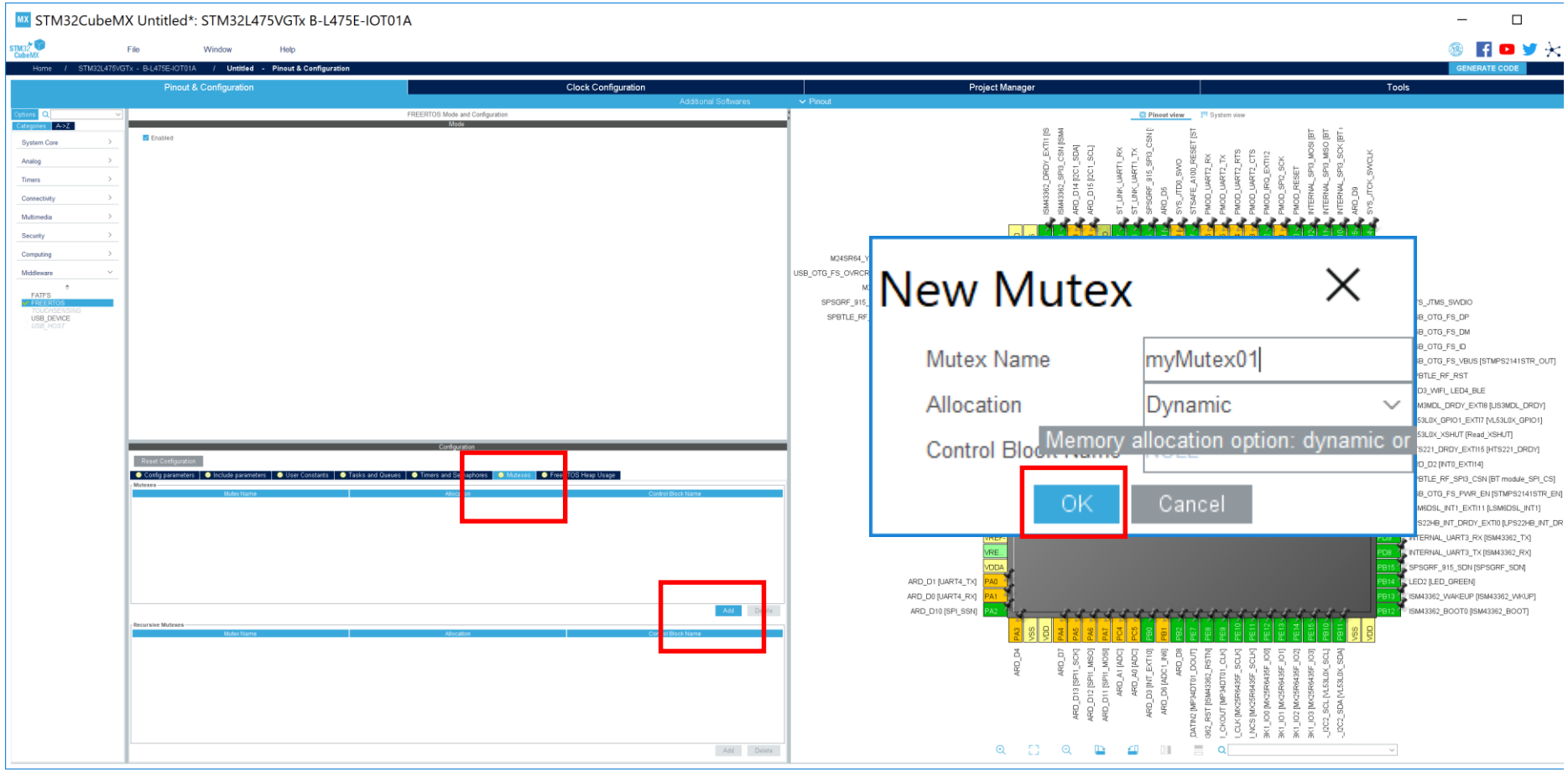
Step 8. Select “Task and Queues”, and observe 1 default task created



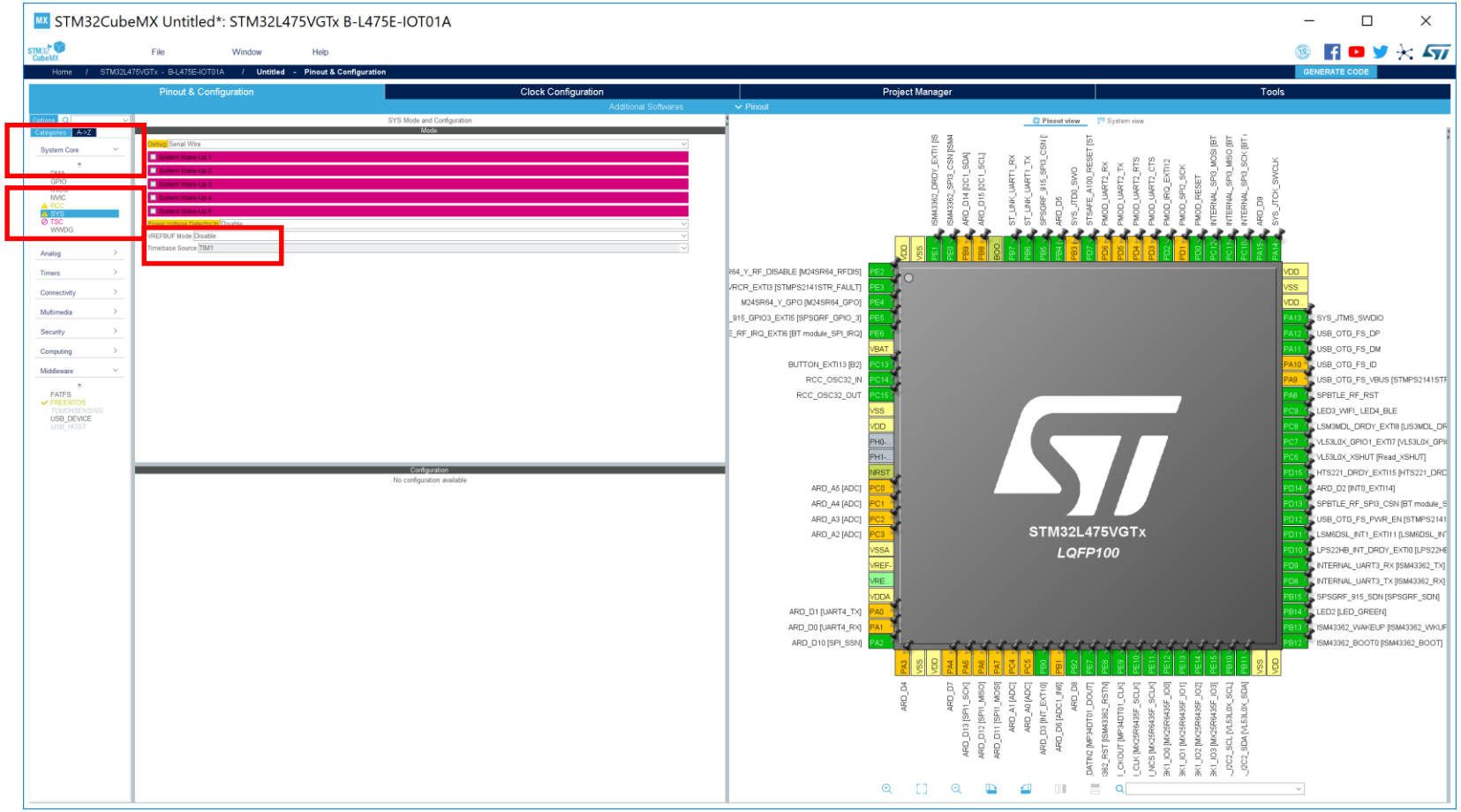
Step 9. Click Add to add 2nd Task



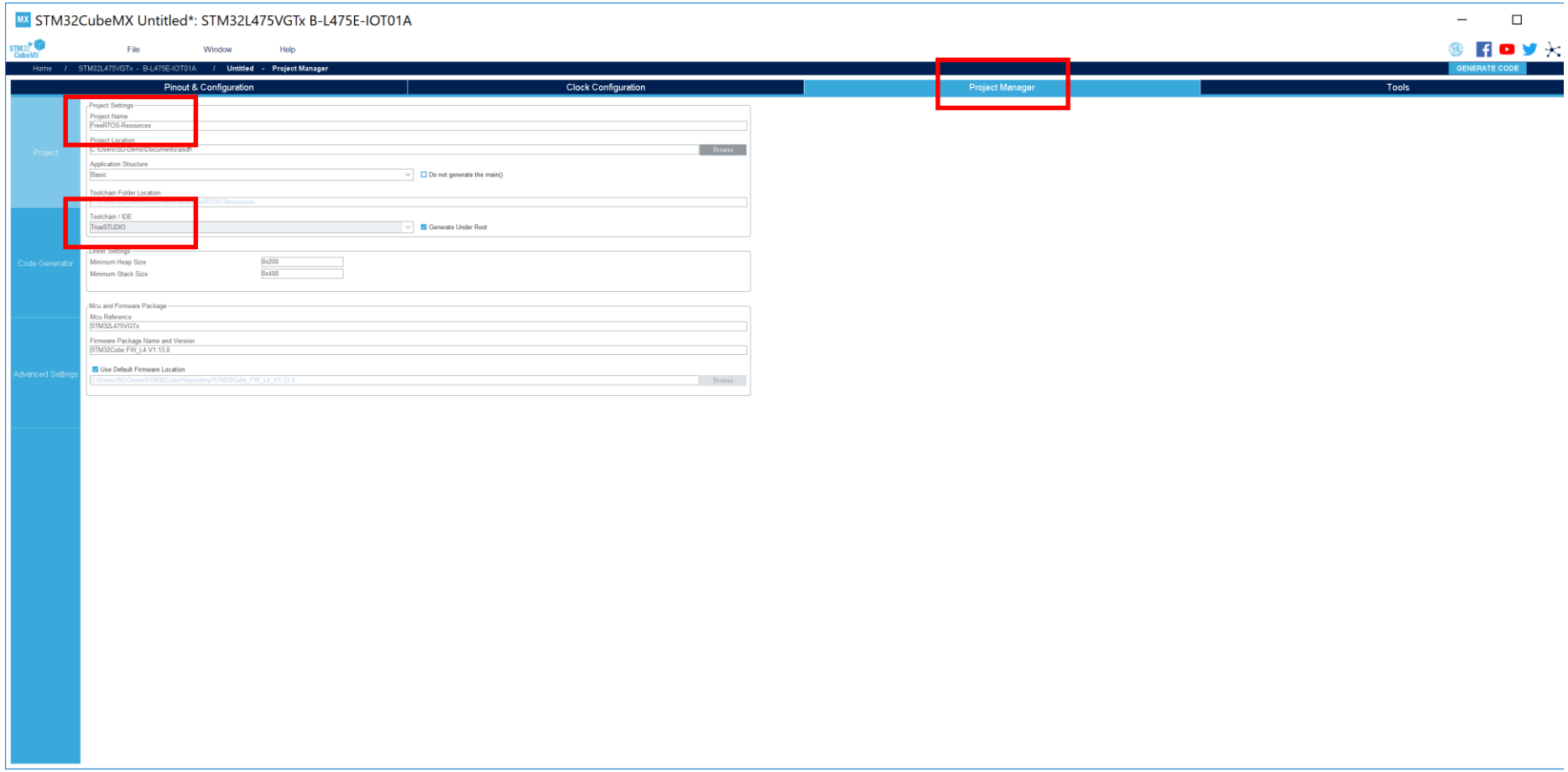
Step 10. Select Mutexes, then Add, then OK



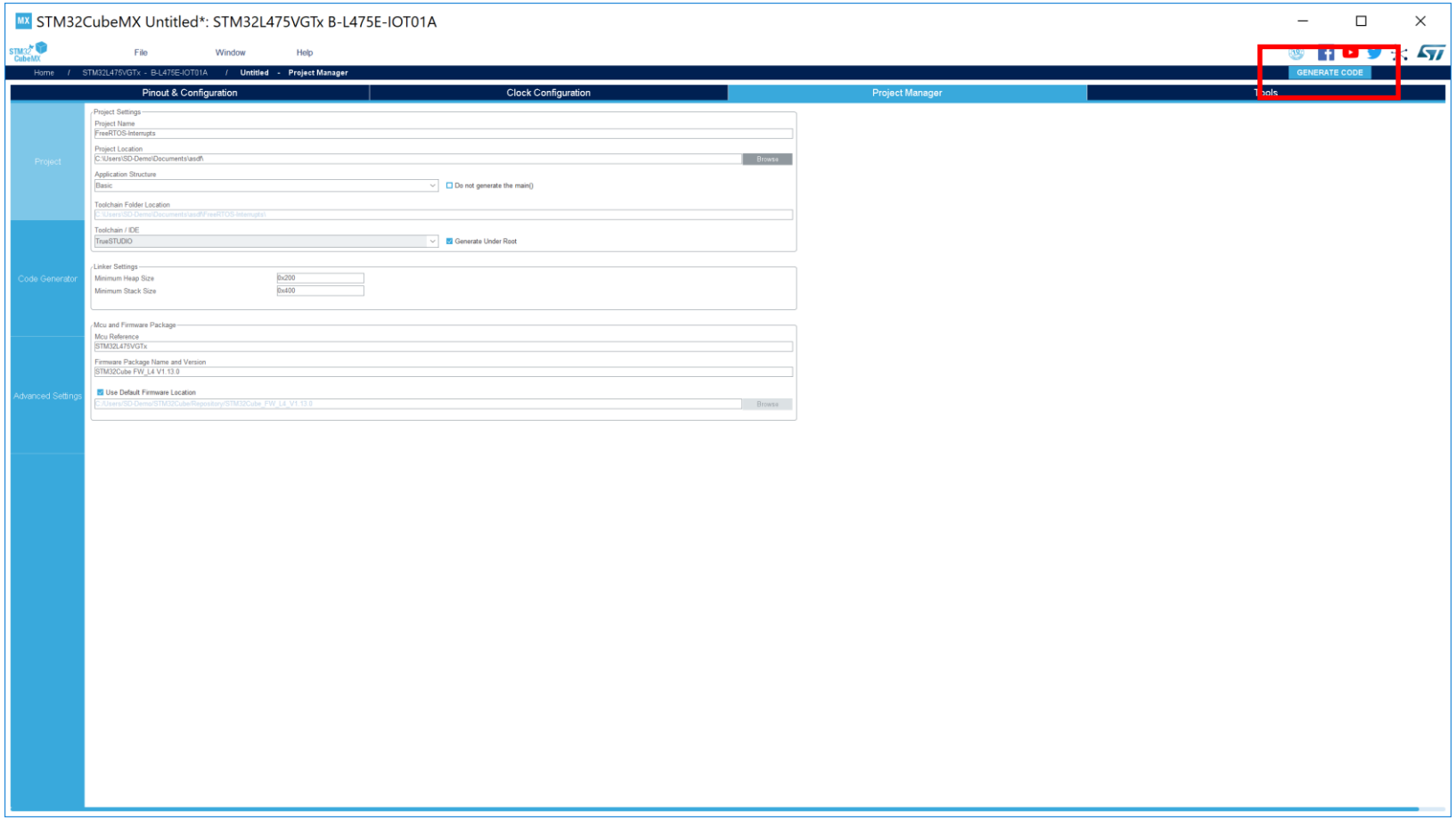
Step 11. Select System Core, Sys, Timebase Source, TIM1



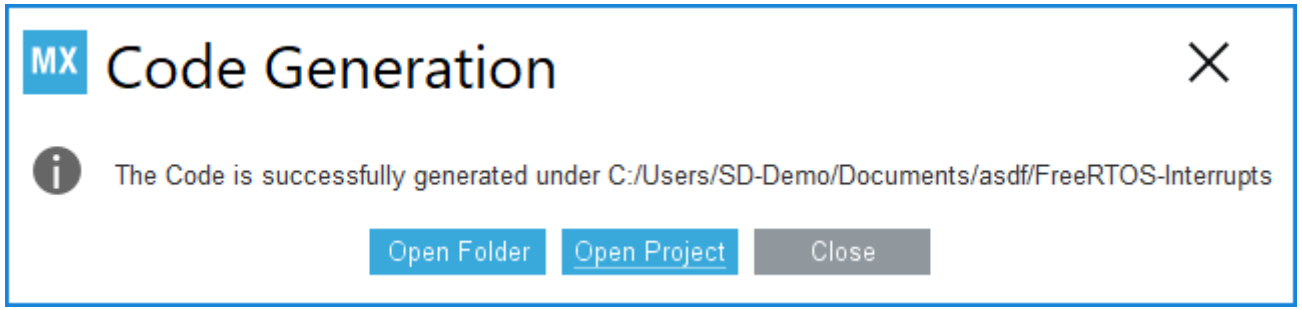
Step 12. Enter Project Name: “FreeRTOS-Resources” and Toolchain/IDE: TrueStudio



Step 13. Select “Generate Code”



Step 14 Select “Open Project”



Step 15. Add interrupt code to increment counter

```
211
212 int count;
213
214 void EXTI15_10_IRQHandler(void)
215 {
216   /* USER CODE BEGIN EXTI15_10_IRQn 0 */
217   int ret = taskENTER_CRITICAL_FROM_ISR();
218   count++;
219   taskEXIT_CRITICAL_FROM_ISR(ret);
220
221   /* USER CODE END EXTI15_10_IRQn 0 */
222   HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_10);
223   HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_11);
224   HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_13);
225   HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_14);
226   HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_15);
227   /* USER CODE BEGIN EXTI15_10_IRQn 1 */
228
229   /* USER CODE END EXTI15_10_IRQn 1 */
230 }
```

Step 16. Add Task 1 Code to read count and flash LED that many times after grabbing mutex

stm32l4xx_it.c | *main.c | portmacro.h | task.h | stm32l4xx_hal_

```c
extern int count;

void StartDefaultTask(void const * argument)
{
  /* USER CODE BEGIN 5 */
  for(;;)
  {
    osDelay(2000); // Sleep 2 seconds
    // Read count value
    int flashCount = 0;
    taskENTER_CRITICAL();
    flashCount = count;
    // If count is 0 then nothing to do
    if (count == 0) {
        taskEXIT_CRITICAL();
        continue;
    }
    // Else clear the count
    count = 0;
    taskEXIT_CRITICAL();

    // Grab the mutex
    osMutexWait (myMutex01Handle, osWaitForever);

    // Flash the LED flashCount times
    while (flashCount != 0) {
        flashCount--;
        osDelay(500);
        HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin );
    }
    //Always end with LED off
    HAL_GPIO_WritePin(LED2_GPIO_Port, LED2_Pin, 0);

    // Release the mutex
    osMutexRelease (myMutex01Handle);
  }
```

Step 17. Add code for 2nd Task

stm32l4xx_it.c | *main.c | portmacro.h | task.h | stm32l4xx_hal_gp

```c
/* USER CODE END Header_StartTask02 */
void StartTask02(void const * argument)
{
  for(;;)
  {
        osDelay(5000); // Sleep 2 seconds
        // Read count value
        int flashCount = 0;
        taskENTER_CRITICAL();
        flashCount = count;
        // If count is 0 then nothing to do
        if (count == 0) {
            taskEXIT_CRITICAL();
            continue;
        }
        // Else clear the count
        count = 0;
        taskEXIT_CRITICAL();

        // Grab the mutex
        osMutexWait (myMutex01Handle, osWaitForever);

        //Always end with LED ON
        HAL_GPIO_WritePin(LED2_GPIO_Port, LED2_Pin, 1);

        // Flash the LED flashCount times
        while (flashCount != 0) {
            flashCount--;
            osDelay(1000);
            HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin );
        }
        //Always end with LED off
        HAL_GPIO_WritePin(LED2_GPIO_Port, LED2_Pin, 0);

        // Release the mutex
        osMutexRelease (myMutex01Handle);
```

Screenshot of LED2 #defines

main.c | main.h | cmsis_os.h | freertos.c | stm32l4xx_it.c | main.c

```
#define QUAD_SPI_BK1_IO3_Pin GPIO_PIN_15
#define QUAD_SPI_BK1_IO3_GPIO_Port GPIOE
#define INTERNAL_I2C2_SCL_Pin GPIO_PIN_10
#define INTERNAL_I2C2_SCL_GPIO_Port GPIOB
#define INTERNAL_I2C2_SDA_Pin GPIO_PIN_11
#define INTERNAL_I2C2_SDA_GPIO_Port GPIOB
#define ISM43362_BOOT0_Pin GPIO_PIN_12
#define ISM43362_BOOT0_GPIO_Port GPIOB
#define ISM43362_WAKEUP_Pin GPIO_PIN_13
#define ISM43362_WAKEUP_GPIO_Port GPIOB
#define LED2_Pin GPIO_PIN_14
#define LED2_GPIO_Port GPIOB
#define SPSGRF_915_SDN_Pin GPIO_PIN_15
#define SPSGRF_915_SDN_GPIO_Port GPIOB
#define INTERNAL_UART3_TX_Pin GPIO_PIN_8
```

Screenshot of BLUE Button Interrupt Defines

main.c | main.h | cmsis_os.h | freertos.c | stm32l4xx_it.c | main.c

```
#define M24SR64_Y_GPO_GPIO_Port GPIOE
#define SPSGRF_915_GPIO3_EXTI5_Pin GPIO_PIN_5
#define SPSGRF_915_GPIO3_EXTI5_GPIO_Port GPIOE
#define SPSGRF_915_GPIO3_EXTI5_EXTI_IRQn EXTI9_5_IRQn
#define SPBTLE_RF_IRQ_EXTI6_Pin GPIO_PIN_6
#define SPBTLE_RF_IRQ_EXTI6_GPIO_Port GPIOE
#define SPBTLE_RF_IRQ_EXTI6_EXTI_IRQn EXTI9_5_IRQn
#define BUTTON_EXTI13_Pin GPIO_PIN_13
#define BUTTON_EXTI13_GPIO_Port GPIOC
#define BUTTON_EXTI13_EXTI_IRQn EXTI15_10_IRQn
#define ARD_A5_Pin GPIO_PIN_0
#define ARD_A5_GPIO_Port GPIOC
#define ARD_A4_Pin GPIO_PIN_1
#define ARD_A4_GPIO_Port GPIOC
```