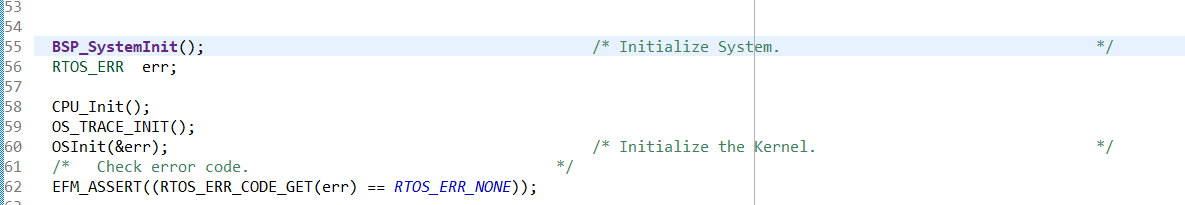
Final Project Week5 Summary

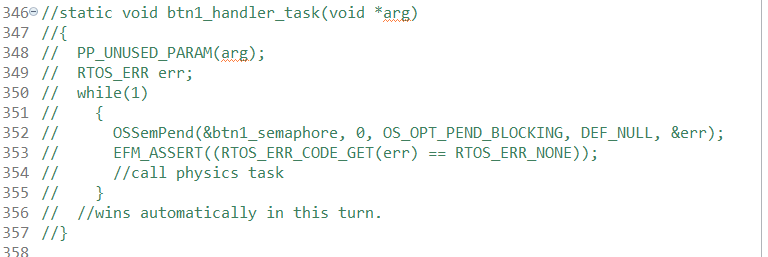
This week I have another 20% of my work done. During this week, I have completed the LCD Display Task so that it can show the updated location of the ball, the platform movement, and the canyon wall onto the LCD screen, although we have encountered some challenges.

At the beginning of the week, I only completed the x direction movement of the ball and was ready to test that on the screen. It shows me some errors after I debug the project. Most of them are just syntax errors but there were two of them which are very tricky. Once I first ran the program, it was stuck at a default function inside the BSP\_SystemInit() function, which is showed in the image below:



At first, me and the TAs didn’t imagine it would be stuck here because that means some hardwares like interrupts were not setting up correctly, which should not be the case because the final project lab is based on lab 7 and my lab 7 works perfectly fine. After a few rounds of checks, We finally found the reason that causes this error. Since we are not using the Systick Handler function in this project. So I commented out the Systick\_Handler function in my blick.c file. But I have not commented out the Systick Config function in the main.c file which sets up the systick handler. This means in the BSP\_SystemInit function, they are supposed to check for systick creation and a handler function that deals with what will happen after the systick interrupt takes place. But now since I have commented out the interrupt handler function, when there is an interrupt happening, there will be no functions to handle that, so an error will occur.

After this error has been solved, another error appeared: in my btn1 irq handler function, I posted a semaphore every time I received an interrupt from button 1. In my btn1 handler task function, I pend the semaphore but do nothing. Normally it would go to the semaphore post from the irq handler function and go directly to the semaphore pend function and go to other tasks based on the scheduler. But what it actually does is it goes to the semaphore post first, then goes to the semaphore pend and then it goes to the same semaphore pend another time and returns an error. At first, we thought it might be because of the wrong semaphore pointer or we were posting the same semaphore somewhere we are not supposed to. But later on we found out that none of these were the reason why this error occurred. So in order to avoid this error, we just commented out the btn1 handler task function after we have eliminated all of the possible reasonings.



After these two errors have been fixed, all other errors and warnings were very easy to fix. So far the LCD task, Physics task and Platform task were finished. The LCD screen can show the movement of the platform and the projectile motion of the ball and the canyon wall. Right now the only two tasks left are the button task and the LED task.