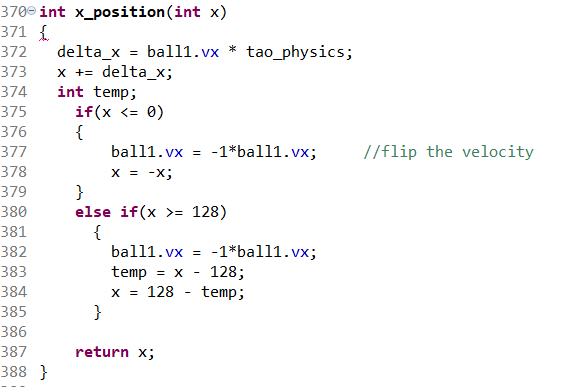
Final Project RTOS Week 4

This week, I completed another 20% of my Final Project by fixing some of the logic errors that I never imagined would happen in my physics task. At first I change the x position of the ball simply by adding or subtracting a delta x from its original position. This leads to a very complex situation because coordinates don’t have directions. But then I found that there is an equation we can use that computes the change in position in terms of velocity. S = V \* t + ½ \*a \* t^2. Since we don’t have acceleration in x direction, the equation then becomes s = v \* t. Since velocity contains both speed and direction. So now every time the ball hits the wall, I can just flip the sign of the velocity to make the delta x change its direction.

After thinking of several situations, there are still some errors because I check its original position, then add the delta\_x to its original position and assign to it. This only works when the displacement is 1. If it is larger than 1, then it will not work. For instance, if the right most boundary is at 100 and the current ball position is at 90 and the displacement is 11, by default the ball should end up at x = 99 but according to my logic, it will end up at x = 101.

Once again I made some changes and my current logic is shown in the image below. Inside the function, I’m calculating where its new position will be first, then check if the new position will exceed the limit. If it’s less than 0, which means hit the leftmost wall, then simply change the sign of the position and the velocity. If x is greater than 128 (hits the rightmost wall), flip the sign of the velocity first, then create a temporary variable that stores the difference between new position and the rightmost limit. Then subtract the temporary variable from the limit and assign it to x to get the new position.



In order to make the project work, functional tests are necessary. For this final project, there are a number of functional tests that we can make.

1. When the ball hits the right wall, the ball should bounce back with the flipped sign velocity in x direction
2. When the ball hits the left wall, the ball should bounce back with the flipped sign velocity in x direction
3. When the ball hits the non-boostered platform, it will bounce back up with a slower velocity.
4. When the ball hits the boostered platform, it will bounce back up with a higher velocity. And it should bounce higher
5. When the ball did not hit the platform, game over.
6. The ball should fall and bounce in a projectile motion.
7. By default mode, the platform should just move as the x position of the ball does.
8. If I press the left side of the slider, the platform should slow down its velocity generally if it’s moving to the right then start moving to the left.
9. If I press the right side of the slider, the platform should slow down its velocity generally if it’s moving to the left then start moving to the left.
10. If I press the left side of the slider, the platform should speed up its velocity generally if it’s moving to the left.
11. If I press the right side of the slider, the platform should speed up its velocity generally if it’s moving to the right.
12. If the ball hits the top, then win.

So far I finished 90 % of the physics task and 90% of the platform task. For the next few weeks, I need to complete the physics task and platform task and work on the LCD and LED task.