

## Week 2 Statement on Project Standing

In week two I wrote the unit tests, set up the initial data structures, set up the tasks, and created the structures for sharing resources. The unit tests were much harder than expected because it relied on planning what many of my functions would look like, in order to call them to test them. The shared resource set up was fun, following my task diagram it was simple and doing it made me aware of my learning from this class: it is the part of this project most unique to RTOS.

I have completed 23.5% of the scoped work, (8/34hr) in 26.5% of the initial time estimate (9/34). Besides the unit tasks causing much more thought than anticipated I still feel good about my progress and am anxious to spend more time developing the game itself.

### Scoped Work

Work Item	Description	Estimated Time	Status
Project Planning	Create an initial task diagram and documents for week1. Have an idea of project design.	2 hrs	Complete
Data Structure Setup	Setup initial data structures for rocket, message queue, angle setpoint, and throttle setpoint.	2 hrs	Complete
Writing Unit Tests	Write initial unit tests. Should fail until further development. Challenge in porting in data needed between cutpoints.	3 hrs	Complete
ITC and Shared Resource Setup	Establish the structures needed between tasks, such as semaphores, timers, mutexes, and the tasks themselves.	1 hr	Complete
Angle Task	Develop code for angle task to pend on semaphore from button ISR and write to angle setpoint.	0.5 hrs	Not Yet Complete
Throttle Task	Develop code for throttle task to pend	0.5 hrs	Not Yet

	on semaphore from timer and write to throttle setpoint.		Complete
LED/PWM Task	Develop code to drive LED's based on PWM. PWM code itself is created in a separate work item.	0.5 hrs	Not Yet Complete
Rocket Design	Practice with the micrium graphics library to design how the rocket will look and what are the meaningful points.	1 hr	Not Yet Complete
Display Task	Develop code for display task to take data from rocket data structure and display the rocket. Challenge is to display the rotation of the rocket graphic, here the math is done to move the vertices of the rocket.	5 hrs	Not Yet Complete
Physics Task	Develop code for physics task. This is the bulk of the project. Includes programming kinematic equations to computing the acceleration, thrust, fuel, position, and rotation of logic. Will also be responsible for knowing if a win or loss has occurred.	8 hrs	Not Yet Complete
PWM config	Create the routines necessary to program configurable PWMs using software timers. Needed by LED task.	2.5 hrs	Not Yet Complete
Configurability Implementation	Program a home screen on the game that takes in input via the buttons. Also takes config either through config file or changing settings in project code.	3 hrs	Not Yet Complete
Debug	Built in time to debug. After all previous work items complete it is expected that the project is not fully functional without substantial debug work.	5 hrs	Not Yet Complete

## **Completed this Week**

### **ITC and Shared Resource Setup:**

This was simple thanks to the task diagram. Created the tasks and all of the necessary components for the tasks (stacks, tcbs, etc). Also set up the OS timers, semaphores, and message queue that will be used for inter task communication. Finally created the mutexes needed to protect the fuel setpoint, angle setpoint, and rocket structure.

### **Data Structure Setup:**

Defined the structs that are needed such as the rocket structure, configuration structure, and polygon struct. The polygon struct will keep track of the vertex coordinates for plotting the rocket. Also created structs for the angle and fuel setpoints, right now these are just variables so they do not need to be structs. However, I anticipate that in the future I may need to keep track of more than just the current value, so making them structs will make them easier to modify.

### **Writing Unit Tests:**

This was much harder than anticipated. Basically this task forced me to think ahead much more than I was prepared for. I want to test certain critical pieces, like the PWM, so to come up with how to test them I had to design them, or at least decide how they will be accessed. This was tough but I like the tests that I have.