Week 5 Statement on Project Standing

This week I finished the PWM work for the LEDs and the blackout handling. Instead of having another task for the LED handling I decided to call the functions from the physics task. The software timers run in a kernel level timer task so there was no need for a LED task that would only start timers and then exit. The task diagram has been updated accordingly. The blackout handling uses a software timer to toggle a flag, when blackout occurs it sets a flag that stays on until the timer call back resets it, the flag determines if the user can control the acceleration.

I have completed 88% of the scoped work, (30/34hr) in 82% of the initial time estimate (28/34). I am ahead of schedule at the moment. Note I am also counting debug work as complete, the only work item left is to implement configurability which may bring in its own debugging but should fit in the scheduled 4 hours.

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	Complete
Throttle Task	Develop code for throttle task to pend on semaphore from timer and write to	0.5 hrs	Complete

	throttle setpoint.		
LED/PWM Math	Develop code to drive LED's based on PWM calculations. PWM code itself is created in a separate work item.	0.5 hrs	Complete
Rocket Design	Practice with the micrium graphics library to design how the rocket will look and what are the meaningful points.	1 hr	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	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	Complete
PWM config	Create the routines necessary to program configurable PWMs using software timers. Needed by LED task.	2.5 hrs	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.	4 hrs	Not Yet Complete
Blackout Calculations	Figure out max acceleration and how to respond to blackout.	1 hr	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.	3 hrs	Complete

Completed this Week

PWM config:
To make the LEDs toggle the physics initialization starts a periodic timer whose callback turns on the selected LED. That callback also starts a one-shot timer whose callback turns off the LED. It took some learning to figure out how to pass inputs into the timer callback functions so that the first callback can set a variable time for the second timer that controls the duty cycle. This functionality is done through functions like pwm_set and pwm_start so it was decided there was no need for a separate LED task.

LED/PWM Math:
To make the LEDs blink faster as the percent of their ratios increased was a little challenging. For example, when there is no finger on the slider LED 1 blinks slowly because there is no fuel usage, as the finger moves onto the slider the LED blinks faster and when at the maximum fuel usage the LED blinks fastest. To make this happen the period of the PWM had to be dynamically calculated. I decided to make it a linear relationship, finger position to blink period.

Debug:
Marked debug this week because the game is playable and complete. Any major bugs have been overcome. The configurability is the last work item, of course this could introduce more bugs but the time and work to correct them is being included in the estimate that configurability implementation will take 4 hours.