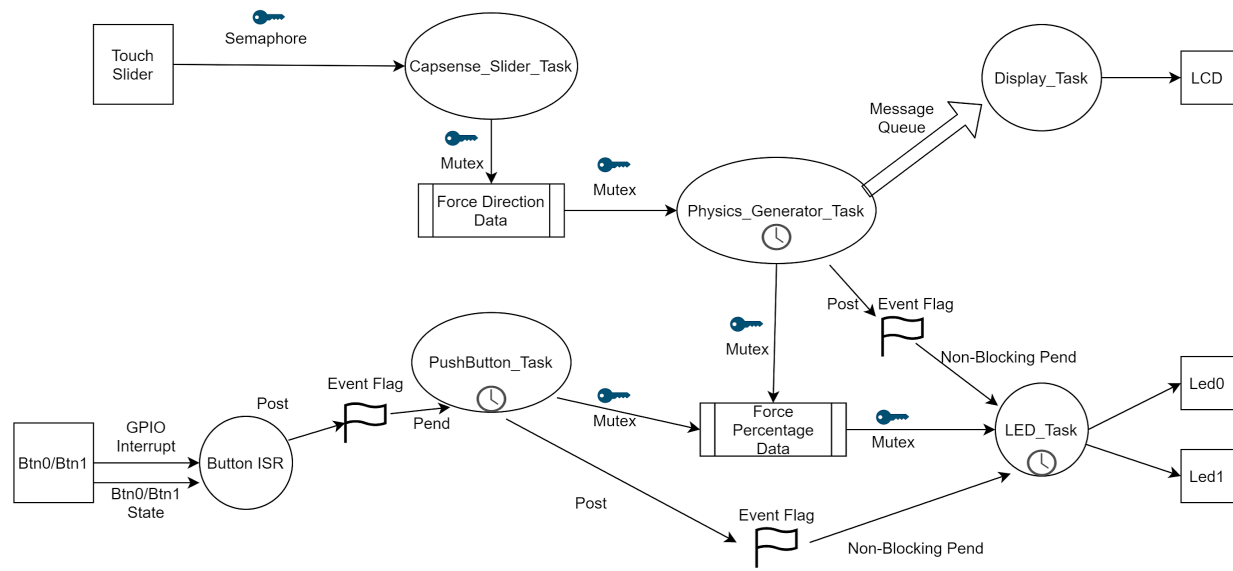


Week 1 Report

Task Diagram



Unit Tests and Functional tests:

My current plan for unit tests is to have these tests be implemented as calling and testing the inner functions of a task. This allows me to simulate desired testing data as if I received the data from ITC without having to actually load the data into the ITC. The unit tests will be as follows:

1. Button 0 reads to ensure a button0 pressed event and then a button0 released event is sent into the pushbutton task the button recording function of PushButton_task will return that pushbutton 0 is pressed. This is cut by the Button ISR and before the writing of force percentage to the shared Force Percentage Data.
2. Button 1 reads to ensure a button1 pressed event and then a button1 released event is sent into the pushbutton task the button recording function of PushButton_task will return that pushbutton 1 is pressed. This is cut by the Button ISR and before the writing of force percentage to the shared Force Percentage Data.
3. Button error read to insure a button1 pressed event and then a button0 pressed event and then a button1 released event are sent into the pushbutton task the button recording function of PushButton_task will return that no push buttons are pressed. This is cut by the Button ISR and before the writing of force percentage to the shared Force Percentage Data.

4. Button error read to ensure a button0 pressed event and then a button1 pressed event and then a button0 released event are sent into the pushbutton task the button recording function of PushButton_task will return that no push buttons are pressed. This is cut by the Button ISR and before the writing of force percentage to the shared Force Percentage Data.
5. Slider Reading pressed test, ensures that given values of the normalized slider positions function with all below the pressed threshold, it will return the lowest slider position. This unit test is cut by the Touch slider semaphore and the input to the force direction data.
6. Slider Reading not pressed, ensures that given values of the normalized slider positions function with all values above the pressed threshold, it will return that no slider is pressed. This unit test is cut by the Touch slider semaphore and the input to the force direction data.
7. Slider Reading out of bounds error, ensures that given values of the normalized slider positions are all 0 or all above the given threshold, it will return an error code that can be caught to prevent buggy performance. This unit test is cut by the Touch slider semaphore and the input to the force direction data.
8. LED1 Control, this unit test looks at the handling logic and makes sure if a stub provides the Event Flag for LED1 to be turned on, the handling logic will turn on LED1. This Unit test is cut Around the LED Task by the Event Flag data structure and the BSP LED functions.
9. LED0 Control, this unit test looks at the handling logic and makes sure if a stub provides the Force percentage at 100, the logic will return that LED0 should be high. This Unit test is cut Around the LED Task by the force percentage data structure and the BSP LED functions.
10. LED0 Control, this unit test looks at the handling logic and makes sure if a stub provides the Force percentage at 0, the logic will return that LED0 should be low. This Unit test is cut Around the LED Task by the force percentage data structure and the BSP LED functions.
11. Physics_generator task, this unit test tests the physics engine with given global inputs as well as the force direction and force percentage and makes sure that given the current state, the change in current state is correct.

Testing Status Summary

All unit tests 1-11 are currently at status "NotRun". This is because none of the tests are currently implemented.

Project Summary Statement

This week I picked to Implement option 1 of the project, from there I implemented my task diagram, as well as unit tests. Finally I scoped out my project.

I have completed 15% (2.5/17hrs) of the project scope in this first setup week. This took me 21% (3.5/17hrs) of the time. This may mean I will have to adjust my allotted time, however it may

also be because I am slow at writing and planning. I may have to adjust next week depending on how long my programming takes.

Project Scope

- Task Diagram
 - Complete
 - Takes 1 hour
 - Spent 1 hour

This was a good idea to have this task in the project be completed first, it gave me an understanding and preparedness of what I should be expecting and what I need to complete in the upcoming weeks.

- Unit Test generating
 - Complete
 - Takes 0.5 hours
 - Spent 1 hour

This was a good next step in nailing down how this project works. However it took me longer than expected to come up with good tests. I found I really needed to know my implementation to get this right.

- Project Scoping
 - Complete
 - Takes 1 hours
 - Spent 0.5 hours

This was a different project planning too that I have not done before, but I feel that it will be helpful in knowing how long it will take to meet my deadlines. It took slightly longer because I kept finding tasks I would need to complete in the project.

- Risks
 - Complete
 - Takes 0.5 hours
 - Spent 0.5 hours

Coming up with risks seemed difficult at first, but by the end of planning on risks, I could start seeing them appear, just by letting my mind start to think about them over the course of a half hour.

- Setting up the Project In simplicity with skeleton tasks and stub functions
 - Yet to complete
 - Takes 1 hour
- Data Structure Definitions
 - Yet to complete
 - Takes 1 hour
- Unit testing implementation of all unit tests
 - Yet to complete
 - Takes 2 hours
- Pushbutton Task
 - Yet to complete

- Takes 1.5 hours
- Slider Task implementation
 - Yet to complete
 - Takes 1 hour
- Physics Engine Task
 - Yet to complete
 - Takes 2.5 hours
- Display Task
 - Yet to complete
 - Takes 2 hours
- LED Task
 - Yet to complete
 - Takes 1 hours
- Functional Tests
 - Yet to complete
 - Takes 1 hour
- Integrating project subsystems together for complete functionality
 - Yet to complete
 - Takes 1 hour
- Final quality inspection
 - Yet to complete
 - Takes 0.5 hours

Total Complete/ Total Required = 2.5/17 hours 15%

Risks

See the attached spreadsheet for my filled out Risks register.