Time estimates:

Reading and understanding the project requirements and instructions:

* estimate = 1 hour
* actual = 1.2 hours

Designing task diagram:

* estimate = 3 hours
* so far = 3 hours

**Task diagram,**

**showing data flow and appropriate ITC/MutEx.  (From this diagram, you should later be able to show that your design will fulfill functional requirements, and it should provide clarity about data structures between execution entities (Tasks/ISRs).)**

**A picture containing text, indoor

Description automatically generated**

**Test plan and results**

**Describe at least 2 desirable "cutting points" for testing (see lecture topic 2.5) covering main paths and/or boundary condition handling.**

1. **The input tasks (Capsense slider and Pushbutton Controls) should be in their own contained test group.**
2. **The Physics task should be developed and test on its own with known inputs that have been worked manually before being integrated with other tasks**
3. **The output tasks (Graphics, Display, Force Magnitude, Evacuation) will also be developed in their own group.**

**Statement of where your project stands:**

**Accurate summary statement of your functionality deliverables and usability so far.**

The project has functionality and usability at this stage as it is purely concerned with planning and algorithm generation. The task flow chart has been generated and will be used to begin programming the tasks and creating the required data structures during the next phase.

**Summary effort & estimate numbers.**

Reading and understanding the project requirements and instructions:

* estimate = 1 hour
* actual = 1.2 hours

Designing task diagram:

* estimate = 3 hours
* actual = 3.5 hours

**List of in-scope work items**(NOT just \_this\_ week's), **indicating complete or not-yet-complete, along with your estimates of how long you think they will take in total for each**

**Assemble task diagram - 3 hours; complete, pending review**

**Work through hand example of Physics calculating tasks – 3 hours; not started**

**Write out detailed pseudo code for each task – 3 hours; not started**

**Write code for each task – 5 hours; not started**

**Test code – 1 hour; not started**

**Debug code – 10 hours; not started**

**Update your risk register**

 Each week, re-sort your risk register and include it in your weekly report.

**A picture containing chart

Description automatically generated**