**Weekly Tasks Time Estimates:**

Make input tasks diagram

* Estimate = 1 hour
* Actual = 1 hour

Writing buttons input task

* Estimate = 1.5 hours
* Actual = 1 hour

Writing capsense slider input task

* Estimate = 1 hours
* Actual = 0.5 hours

Debugging input tasks

* Estimate = 3 hours
* Actual = 4 hours

**Project Status:**

The task diagram which I wrote last week served me very well this week. The addition of a Mutex protected data structure between the PushButton controls task and the Physics task was required, but it was otherwise unchanged. The completed portion of the task diagram is shown in fig. 1.

Diagram

Description automatically generated

Figure 1: task diagram completion status

All of the input processing that will be required for the physics task has been completed. Of the cutting points I previously identified, this constitutes approximately 25% of the work for the project as the Physics task is estimated to require 50% and the Display task the last 25%.

In order to verify that the input process was functioning correctly, I wrote a test LCD display function which output the inputs after they were processed and as the Physics task will see them. Each data structure used by more than one task has been protected with a Mutex. The following parameters were verified.

Buttons:

* If button 0 is pressed it triggers an interrupt, which then increments the charge level of the capacitors every 1 second while it is held.
* If button 0 is released an interrupt is triggered. This then sets the capacitor energy counter to zero and indicates that a fire operation is required.
* If button 1 is pressed while the capacitor energy level is zero, nothing happens.
* If button 1 is pressed while the capacitor energy level is greater than zero, the capacitor energy counter is set to zero and it indicates that a shield operation is required.
* When button 1 is released, no additional interrupt is generated and it is ignored as input.

Slider:

* The slider’s input is periodically sampled.
* If the slider is touched the variable tracking what level and direction of force is required is set.
* If the slider is not being touched, the tracking variable is reset to a no applied force state.

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Estimate** | **Actual Time Used** | **Status** |
| Task Diagram | 3 hours | 5 hours | Complete |
| Physics hand example | 3 hours | ---------- | Not Started |
| Write Input Tasks | 5.5 hours | 7.5 hours | Complete |
| Write Physics Tasks | 8 hours | ---------- | Not Started |
| Write Display Tasks | 8 hours | ---------- | Not Started |
| Integrate Tasks | 3 hours | ---------- | Not Started |
| Debug Final Code | 5 hours | ---------- | Not Started |

Table 1: Work Items Status and Time

As can be seen in table 1, my estimates of time required to complete tasks has been within expected accuracy for a good estimate with development projects. They have thus far avoided the one-third of actual error that was quoted to us by the instructor. I am glad to have started the project with the input tasks, as I will be able to integrate each cutting points group of code with the one prior to it, and hopefully have a much smoother debugging stage than if I had developed all three separately and integrated at the end of the project. My updated Risk Register is shown in fig. 2.

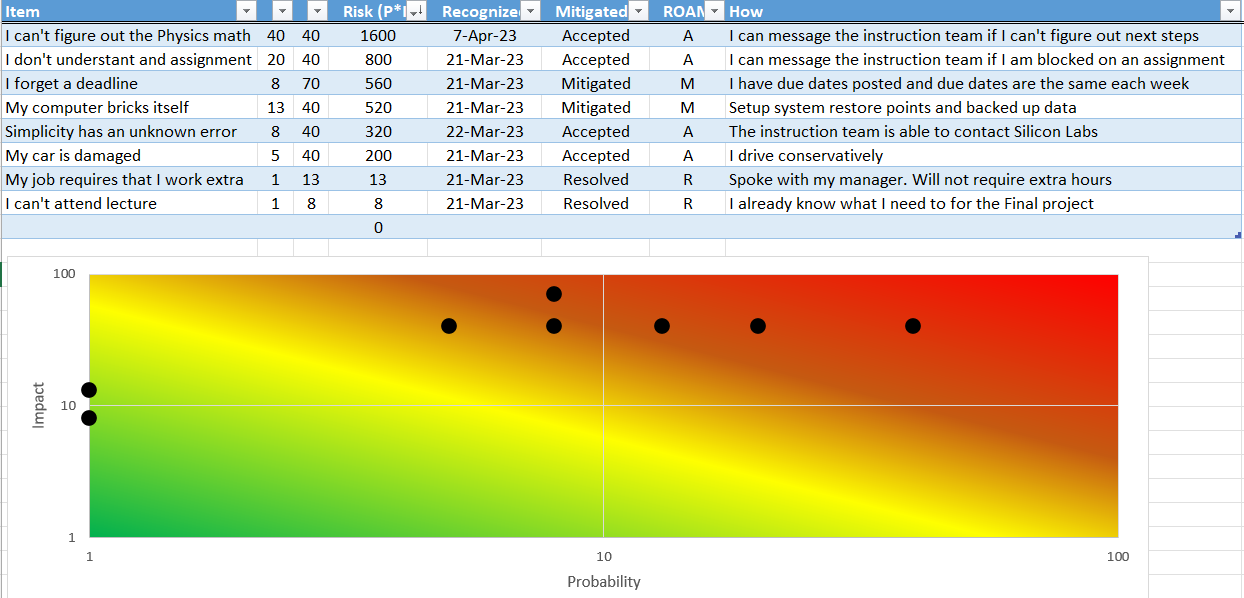


Figure 2: Updated Risk Register