CS1980 – Hardware / Software Interface

Deliverable 1 – Proposal

Rob Colleran, Eric Hochendoner, Andrew Panzl

POC: Don Bullock

The task for this project is to develop software that can interface with a hardware device consisting of a raspberry pi that controls various components. These components could include a gyroscope, temperature sensor, motors, or other peripherals. This hardware will not always be available for testing and its exact specifications may be changed by the POC over time, therefore hardware will need to be simulated during development. The software to be developed should include a web server/client interface to communicate with the software controlling the hardware, lower level software for the pi that controls the hardware, and a driver to simulate the hardware. As part of the server/client interface, networking code will be needed to handle communication over the bus between the server and client and handle problems such as communication errors and race conditions. In addition to the software, documentation will need to be done. This will include a system design document, system requirement specification, and a working agreement. It will also be necessary to do quality assurance (QA) work and some form of mock budgeting according to the number of man hours each task will take.

Our team plans to use Slack and (if necessary) Google Hangouts to communicate about development and with our POC. We will also have weekly in-person meetings with our POC to learn more about the project and assess progress. We plan to use Scrum as our development methodology, with sprints every two weeks followed by meetings and code reviews. Python will be the primary development language, and Raspbian will most likely be our target operating system for the pi, although this might change. Unit tests and manual tests will be done on the code as part of QA. We plan to share QA responsibilities for our code, although this might change as development progresses. Some potential issues may arise due to the fact that this project will require learning over time and that this is the first iteration of the project developed by our POC.

**User stories:**

Highest Priority:

* As a user, I want the hardware drivers to send an alert and halt operation if the motors reach the rotation limit of 90 degrees per axis, so that the software doesn’t break my hardware
* As a user, I want the web interface to allow the input of angle change commands or 3-dimensional locations to aim the gyroscope, so that I can easily control the hardware
* As a user, I want the web interface to detect and disable invalid parameters, so that there is less risk of my hardware being damaged
* As a user, I want the driver to drive the motors from 90 degrees per second (minimum) to 130 degrees per second (maximum) so that the motors can run fast enough without risk of breaking
* As a developer, I want an up-to-date hardware emulator so that I can continue writing drivers when I don’t have access to the hardware
* As a user, I want the driver to reference changes and send commands at a rate of 50hz (minimum) to 100hz (maximum), so that the software can execute fast and successive commands
* As a user, I want the web interface to display current angle offsets of the device at a rate of 50hz (minimum) to 100hz (maximum), so that I can be informed for the device’s position quickly
* As a developer, I want the communication framework to poll both the web interface and driver for status, so that there is a central place to handle and track alerts and updates

Medium Priority:

* As a developer, I want detailed documentation about the code base so that I can easily debug or contribute to other developers’ work.
* As a user, I want a user manual about the system so that I can easily figure out how to use it
* As a user, I want the gyroscope to be able to be leveled using inputs from the accelerometer, so that the hardware has a balancing function
* As a developer, I want all communication between the web client and hardware to be logged, so that I have a central place to look for logs when debugging
* As a developer, I want API endpoints for all functions so that I can stream instructions to the driver

Lower Priority:

* As a user, I want the software to be able to accept batches of commands, so that I can enter commands faster than I could by manually entering each one
* As a developer, I want the web client interface to have an admin panel with debugging commands and info so that I can debug the software
* As a developer, I want all logs to be downloadable, so that I don’t have to connect to the client or hardware each time I want to debug

**Required Documentation:**

System Requirements Spec (SRS): Higher level listing of requirements, more customer focused

System Design Doc (SDD): Lower level listing of requirements with implementation details, maps to SRS

Statement of Work: Agreement with customer about what work will be done, when it will be done, how much it will cost, how budgeting will be done, etc

Work Agreement: Includes details about team organization, division of work, communication and development strategies, etc

**Generalized Layout of the Software Components:**

Hardware drivers (running on raspberry pi)

****

Socket

Socket

Logic Engine/ Communication Framework

Connected hardware or hardware emulator running on pi