System Requirements Specification

Version 1.0

**Table Of Contents**

1. **Introduction**
   1. Project Overview

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.

* 1. Purpose and Scope

In Scope:  
All development, testing, and documentation for:

* The web server/client interface to communicate with the software controlling the hardware
* The lower level software for the pi that controls the hardware
* The driver to simulate the hardware
* The networking code to handle communication between the server and client

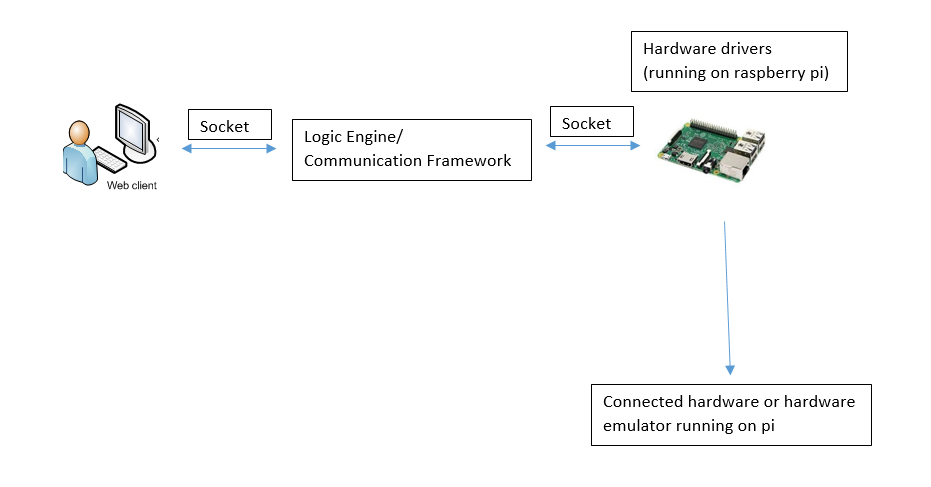
Out of Scope:

Any development or work on the hardware itself

1. **Overall Description**
   1. Product Functions/Context

The interface should load on any computer, and send commands to a server running on a raspberry pi, which in turn controls the hardware via lower level software commands. The hardware may also be emulated when access to it is restricted.

Generalized layout:



* 1. User Characteristics

The user will be the POC, who has knowledge of the hardware, knows the correct commands to make the hardware components accomplish tasks, and has enough technical knowledge to determine if technical requirements are being met correctly.

* 1. Assumptions

It is assumed that:

* The hardware will include, at a minimum, 4 stepper motors and an accelerometer
* The hardware will only occasionally be available for testing
* The user knows the limitations of the hardware
* Raspbian will be the OS on the pi
* All driver, server, client, and communication functionality can be implemented via Python code and raspberry pi libraries
  1. Constraints

Design options are constrained by the following:

* All communication between the client and server should be done via sockets, with no common read/write memory
* There is limited access to the hardware
* The software must utilize libraries already installed on the raspberry pi, which are currently unknown
* All communication between the client and server must be logged
* All QA responsibilities will be shared amongst the dev team
  1. Dependencies/Environments

The web client and interface should run on any modern laptop or desktop PC with Python 3.0 installed. The server and drivers should run on a raspberry pi running Raspbian with Python 3.0 installed. The low level drivers controlling the hardware must be developed before the client, server, and communication framework can be completed, because handling commands and alerts to and from the driver is the core of their functionality.

1. **External Interfaces**
   1. User Interfaces

A UI will be developed in Python to run in a terminal and accept input in the form of angle offsets, specific angle locations, and xyz offsets

* 1. Network Interfaces

Sockets will be used to communicate between the web client and server.

* 1. API

A ReST API must be created to allow for automated input to be sent over sockets.

* 1. Protocols

I2C will be the protocol for direct communication with the hardware.