Detailed Software Design
for
Video-Based Control System for Automatic Standing Desks

Version 2.0

Prepared by Marco Borg, Mohit Kapur, Justin Power, Tristan Grondin, and Constantine Valettas

University of Ottawa

April 12, 2023

Document History and Distribution

1. Revision History

Revision #	Revision Date	Description of Change	Author
1	December 8, 2022	Initial Draft	Marco Borg
2	April 10, 2023	First Prototype Submission	Tristan Grondin

2. Distribution

Recipient Name	Recipient Organization	Distribution Method
Dr. Emil Petriu	University of Ottawa	Electronic Submission

Table of Contents

1. INTRODUCTION

This document dives into detail about how the Video-Based Control System for Automatic Standing Desks will be tested to verify the goals and expectations of the system set in the System Requirements Specifications document (SRS). Specifically, it overviews how the software and hardware components will be tested. The automated and manual override features of the system will be verified through software and hardware testing. And the power electronics of the system will also be tested through software and hardware testing.

The plan to test the system is to first simulate all components to scan for any fundamental errors in design and then to test a physical working model to sort through any minor bugs after the system has been tested through software simulation programs. This document was based on the SRS document created by the group and the detailed hardware and software designs of the project.

Testing Deliverable:	**Testing Performed By:**	**Testing Date (Estimation):**
Controls Software Simulation	Marco, Mohit, Tristan	February 1, 2023
Power Electronics Simulations	Justin, Constantine	February 1, 2023
Integrated Controls Software Simulation	Marco, Mohit, Tristan	March 1, 2023
Integrated Power Electronics Simulations	Justin, Constantine	March 1, 2023
Integrated System Testing	ALL	March 15, 2023

1.1 Objectives

The objective of this report is to provide proper documentation as to how the test plan evolved with the development of the video-based control system for a height-adjusting desk. Therefore, a detailed description of the resources and schedule needed to accomplish the testing phase will be provided so that all members of the project can source and reference when need be. High-level descriptions for each task will be provided so that each team member can get a grasp of how each development test will proceed and the milestones needed to mark each testing activity a success.

1.2 Testing Strategy

The testing strategy that will be used in this project is to have different modules tested separately. This allows for more isolation in each sub-system which will make detecting errors and problems much easier to solve as each test will be regularly small and simple. This means that multiple small tests will need to be done routinely so that the project schedule can be strictly followed. Using this strategy makes the development process much easier and allows for bugs and errors in the code to be detected at an early age so that it doesn’t get lost later when the program becomes much larger; however, in using this strategy, integration between all these tests and sub-systems will become more challenging. This is because each system will be tested separately and only after every system works well the integration between them begin, which will prove to be more difficult in having each section work in unison. Even though the integration will be much harder, the benefits of doing smaller more frequent tests still outweigh the cons and will thus be used for the entire development phase of this project. The following points will indicate the details each test will provide so that an adequate description can be made for each test to be performed.

Specific test plan components include:

- *The purpose of this level of test*
- *Items to be tested*
- *Features to be tested*
- *Management and technical approach*
- *Pass / Fail criteria*
- *Individual roles and responsibilities*
- *Milestones*
- *Schedules*
- *Risk assumptions and constraints*

1.3 Scope

The project manager will distribute testing deliverables to the engineers of the project and the deadlines for each deliverable. Both scheduled and unscheduled testing will be updated via a shared document that contains forms for the test completed, if the test was successful or not, and if further testing is required. The document will automatically revise itself when there is an update and save previous versions to the project server.

1.4 Reference Material

Document Name	Author:
System Requirements Specifications	Group 8
Detailed Software Design	Group 8
Detailed Hardware Design	Group 8

2. TEST ITEMS

Items to be tested:

- *Computer Vision Software*
- *Actuator Software*
- *Button Software*
- *Level Software*
- *Power Electronics Hardware*
- *Actuator Hardware*
- *Button Hardware*
- *System Integration*
- *Requirements Specifications*

2.1 Program Modules

Test Deliverable	Criteria	To Be Tested By:
Computer Vision Software	• The user's Shoulders can be read by the software • Relative distance traveled from the user's rest position is accurate	Mohit
Actuator Software	• Can read input from buttons • Can read input from Computer Vision • Sends proper output	Marco and Tristan
Button Software	• Can read input from buttons • Sends proper output	Justin and Constantine
Level Software	• Can properly track the position of each actuator • Can follow leveling algorithm listed in Detailed Software Design	Marco and Tristan
Power Electronics Hardware	• All electronic components are powered properly	Justin
Actuator Hardware	• Both Actuators can move up and down • Both actuators have an equal rate of motion	Marco and Tristan

Button Hardware	• Both buttons are debounced • Both buttons send the proper input signal • The mode switch sends the proper	Justin and Constantine
System Integration	• All systems can communicate with each other and function	ALL
System Requirements	• All integrated components function as specified on the SRS document	ALL

3. FEATURES TO BE TESTED

The features that will be tested in this project include the necessary components to provide a functional video-based control system for a height-adjusting desk. Therefore, the list provided below marks each feature that will be tested individually before it is to be integrated and implemented into the final product.

Test Deliverable	*Passing Criteria*	*To Be Tested By:*
Automatic Mode	• Computer Vision software is integrated and functions properly with the main program as	Mohit, Justin, and Constantine
Manual Mode	• Button software is integrated and functions properly with the main program as required	Marco, Tristan, Justin, and Constantine

The following table will demonstrate the features that were tested up to this point along with its corresponding description and outcome after the testing took place. This will provide an organized review of the progress being made on the developmental prototype with a date being stamped on every testing period. The testing phases shown below are only the testing features that the product user will observe and will be noted as the most important aspect of the testing documentation as this provides the backbone of the video-based control system.

Date	User Features to be Tested	Reason for Testing	Testing Outcome
Feb 17, 2023	Automatic mode detecting and translating linear motion	The computer vision system has previously been tested through various simulation which have proven successful. This test will include the physical implementation of the automatic mode where a user will now face the camera to induce linear motion in both actuators.	The outcome for this test provided a working system which is the key focus to the first prototype. The results resembled the same as the simulations which proves a successful implementation to the real world.
Mar 23, 2023	Manual mode detecting and translating linear motion	A series of buttons are allocated to provide linear movement without the use of the automatic system. Simulations have been used before to show a proof of concept but now this stage in the development process will indicate the successful implementation to the	The outcome of running this system on the raspberry pi showed that by pressing the up or down button, the actuators will respond and show the appropriate linear movement.
Mar 30, 2023	Mode type switch implementation	With both the manual and automatic mode working correctly, they now need to be able to distinguish which mode should take precedent over the other. Thus, this test will include the use of a switch so that the user can alternate between the desired mode the system should operate in.	The results showed that by putting the switch in the automatic mode, the buttons are completely ignored even when given inputs and only relies on the camera to provide movement. Likewise, when the switch is in the manual mode the camera is ignored for all movement and only relies on the buttons which marks more control to the user.

3.1 Software Risk Issues

This section is reserved for the software issues that are to be tested with the following table marking the order and changes of the design template firstly initialized in the project charter. This stage is important because it marks the changes that occur throughout the development process while only looking at the software element of the project.

Date	Software to be Tested	Reason for Testing	Testing Outcome

Feb 17, 2023	Computer Vision System facial recognition and positioning output	The computer vision system is tested first to see the accuracy at which the system detects faces so that an early conceptualization can be done about how the main system will operate. It is important also see how the output window will be monitored showing the positioning of faces so that it can be retrieved for other sub-systems in the software.	The outcome for this stage of the project was that the computer vision system operates correctly using a desktop machine and webcam. The position of the faces is measured and displayed in the command window with up, down, and no movement, measurements being represented as a +1, -1, and 0 respectively.
Jan 18, 2023	Actuator Subroutine compiling and output	The actuator subroutine is responsible for monitoring the type of mode the control system is operating in while also translating the correct signal to the actuators to provide linear movement.	The outcome of running this section of software independently showed that it can actively distinguish and translate a test signal when given the button, computer vision, and mode type signals.
Feb 20, 2023	Self-Leveling Subroutine compiling and output	The self-leveling subroutine will need to be compiled independently to show it can operate on a raspberry pi with the intent to use 2 infrared sensors to see what the height is for both actuators and level them accordingly.	After running the program on its own, it can actively measure the actuators and adjust the heights to match each other; however, this program currently runs without th use of physical sensors and only runs through simulations.
Mar 23, 2023	Button Subroutine compiling and output	The button subroutine will need to be implemented to show that the user can change the type of mode selected and use a series of buttons to control the actuators.	The program compiles and outputs a series of signals to the command window which marks the use of buttons being operated by an external source.
Mar 31, 2023	Raspberry Pi setup and initialization	To setup the raspberry pi for use in the project, it will need to be initialized along with all pins that will be used later in the project. Thus, it is important to show that the main micro-controller functions correctly so that code can be imported onto the controller.	After initializing the raspberry pi, the pins were initialized and ready to be used for further implementations by combining the software elements with the hardware elements to complete the project.
Feb 27, 2023	Canceling the implementation for the on-board camera	After attempting to implement an on-board camera that would sit on the raspberry pi, this feature was canceled with the camera now remaining on the webcam of the computer. The reason for this was because the resolution of the on-board camera was not good enough for the computer vision system to accurately	Therefore, the results remain the same with less implementation needed now.
Apr 3, 2023	Implementation of all subroutines working together	With each sub-system developed and tested independently, they will be combined under a single file to see how they communicate with each other. This is important because the raspberry pi will function by running a single script during the demonstration.	After combining all files together, the results show that the system works accordingly and can be implemented in the raspberry pi.

Apr 5, 2023	Cancelation of the Self-Leveling subroutine	After building the hardware aspects of the project, it was noted that the 2 actuators that were given by the University of Ottawa have the same inclination rate meaning that a self-leveling system will not be needed if they are setup properly.	Therefore, the Self-Leveling system is erased from the main raspberry pi script which makes the code run a little faster as it does not need to call this function after every actuator movement.

3.2 Features Not to be Tested

Multiple systems were tested throughout the development process of the first prototype. However, it is important to note the systems or test cases that will not be tested for the first prototype deliverable. The reason for this is because the first prototype is to generate a proof of concept and is in no way seen to be and end of use product for the market. Therefore, only the basic infrastructure of the first prototype will be given by producing a manual and automatic mode for the control system that produces linear movement in 2 actuators. Therefore, a list will be constructed of more specific cases that will be implemented later but as of right now is not included into this design module.

Features Not to be Tested	Description
Computer Vision Distinguishing Faces	As of right now, the computer vision system operates by detecting a face in the camera's line of sight with linear movement being created depending on where the face is relative to the screen's line of sight. The vision system cannot function properly when there are 2 or more faces in the camera's line of sight as the system cannot provide movement to both face positions.
Physical Implementation of Self-Leveling System	With the current prototype using 2 actuators given by the University of Ottawa, their respective inclination rates are very similar making the self-leveling system redundant as the length of motion is not long enough for this system to be needed. Thus, the implementation of the self-leveling system will not be implemented into the physical circuit and design for this prototype but can be accessed if needed for a later prototype.
Actuator Strength Test	With the system properly developed and the 2 actuators being controlled by either a manual or automatic system, the actuators will need to be tested to show that they can work in unison to lift a board that will hold a certain amount of weight to it. For the first prototype, the 2 actuators used are relatively small than the ideal finished products desired travel length which means that the scale of the system is much smaller as a result. Thus, the actuators used will not be used in the final product as bigger travel actuators will be used with a higher strength threshold which is when this test will take place as it will be much more suited to real world applications.

4. ENVIRONMENTAL REQUIREMENTS

This project needs to be tested under its required environment to provide paramount safety throughout each test while also being able to conduct accurate measurements and data needed. Thus, the following section will investigate where the tests will need to be conducted to follow all these guidelines mentioned above.

4.1 Hardware

To run multiple tests, the proper hardware will be needed. The specific type of hardware will change depending on the test being conducted but throughout the testing phase of this project the required equipment and measurement tools will be needed. The required equipment will include a raspberry pi, a camera, 2 linear actuators, wires, and relays which will all need to be tested to see if each element functions before using them in the prototype project. To gather data and measurements, more hardware tools will be needed such as an oscilloscope to see current flows within a circuit along with a powerful computer to perform various simulations and analyses of the data being gathered.

4.2 Software

The software will also be needed throughout the project with most of the testing being conducted on the program itself. Thus, multiple libraries will be needed as well as compilers to compile the written code so that it can be used on the raspberry pi. Therefore, raspberry pi OS will be used to perform the tests and build the final prototype. Testing software such as Matlab, Simulink, and Multisim will be needed to test the circuit and program before the physical circuit is built so that proof of concept can be adhered to before putting resources into furthering the project.

4.3 Security

To keep security at the forefront of the development process. All tests will be done on the University of Ottawa's campus in their designated laboratory rooms so that all personnel allowed into the laboratory room can be monitored so that the intellectual property cannot be stolen before the product can search for a patent. This puts the project in a very secure location to be developed before it can go out to market with the computer vision system also only tracking those within the laboratory room who will have consent before entering. Thus, this would also give a high level of security to the general public as the development of the computer vision system will not be used against the public before it has been properly tested by those who have consented previously.

4.4 Tools

The special software tools that will be needed throughout the development of this project are the Matlab and Simulink software which will be used extensively in the early phases of testing. This is because each software allows for adequate simulations to take place without any hardware, meaning that the proof of concept can be generated using this software which would pass the first steps of testing. Another simulation software called Simulink will also be used to simulate the physical circuit itself and monitor the performance. This will be very useful as all errors in the circuit design can be detected using this software instead of being needed to physically construct the circuit and risking part malfunctions if the circuit was set up wrong. All members of this project have reasonable experience and knowledge using all three of this software which means that no additional training for the team members on this specific software will be required.

4.5 Publications

The publications that this testing document requires are the formal documents created throughout this project so that they can be properly referenced throughout the testing phase of the project. Thus, the table shown below denotes all required documentation for this project to proceed to the testing stage.

Document Title	Location (Link Path)
Project Charter	Y: \CAPSTONE_PROJECT\Final_report\Project_Charter
Hardware Design	Y: \CAPSTONE_PROJECT\Final_report\hardware_design
Post-Performance Analysis	Y: \CAPSTONE_PROJECT\Final_report\post_performance_analysis
Project Budget	Y: \CAPSTONE_PROJECT\Final_report\project_budget
Project Hazard Assessment	Y: \CAPSTONE_PROJECT\Final_report\project_hazard_assesment
Risk Management	Y: \CAPSTONE_PROJECT\Final_report\risk_management
Specific Requirement Specification	Y: \CAPSTONE_PROJECT\Final_report\specific_requirement_specification

4.6 Risks and Assumptions

Because all physical circuit tests will be conducted using the laboratory rooms provided by the University of Ottawa, the assumption will be made that these rooms will be available 3 times a week for a total of 9 hours of lab session time every week. This assumption will be held constantly from January to May so that the physical prototype can be built within these lab times. Not all the physical material has been gathered yet for the complete development of this project and with current market constraints on electrical elements, finding certain building materials will prove to be difficult. Thus, more assumptions will be made that all necessary equipment will be gathered before the scheduled testing day is reached.

With all these assumptions that have been previously made, there are also inherent risks as well because if these assumptions are not met then, they will fall as a problem in the testing stage of the product. Therefore, it is important to note that all the assumptions made throughout this report is also deemed a risk with a very low probability of failure but still needs to be addressed in the risk assessment document regardless which was done in another document.

5. CHANGE MANAGEMENT PROCEDURES

All engineers will report to the project manager should there be any revisions to the test plan. If approved as a group, the project manager will create a revision for the test plan or authorize the requesting party to do so.

6. PLAN APPROVALS

This plan is to be approved by Dr. Emil Petriu of the University of Ottawa who is our head supervisor and customer of this project. To date, the test plan has not been approved.