ModBot
Project Design Document

Document Version: **1.0**

AUTHORS

This document was prepared by:

Christian Garcia
Project Lead / Developer
christianmgarcia@siu.edu

Courtney Kinnard
Project Manager / Developer
ckinnard@siu.edu

Vincent Davis
Lead Developer
vdavis@siu.edu

J. Kole Cralley
Hardware SME / Developer
jkolecr@siu.edu

VERSION HISTORY

Date	Document Version	Document Revision History	Document Author/Reviser
Feb 16, 2018	1.0	Initial draft	Courtney Kinnard
Feb 26, 2018	2.0	Revisions per Houssam Abbas' request	Courtney Kinnard

APPROVALS

Date	Document Version	Approver Name and Title	Approver Signature
Mar 5, 2018	3.0	James Doe, Project Mentor	
Mar 5, 2018	3.0	John Doe, Project Manager	

TABLE OF CONTENTS

1. INTRODUCTION

The Product Design Specification document documents and tracks the necessary information required to effectively define architecture and system design in order to give the development team guidance on architecture of the system to be developed. The Product Design Specification document is created during the Planning Phase of the project. Its intended audience is the project manager, project team, and development team. Some portions of this document such as the user interface (UI) may on occasion be shared with the client/user, and other stakeholder whose input/approval into the UI is needed.

The Product Design Specification document for this project will also document architecture and design for the hardware components of the robot, such as, but not limited to: robot structure and design, plans for components to be 3-D printed, and a description of the networking between devices. This document will also include the software components previously mentioned.

2. GENERAL OVERVIEW AND DESIGN GUIDELINES/APPROACH

2.1 Assumptions / Constraints / Standards

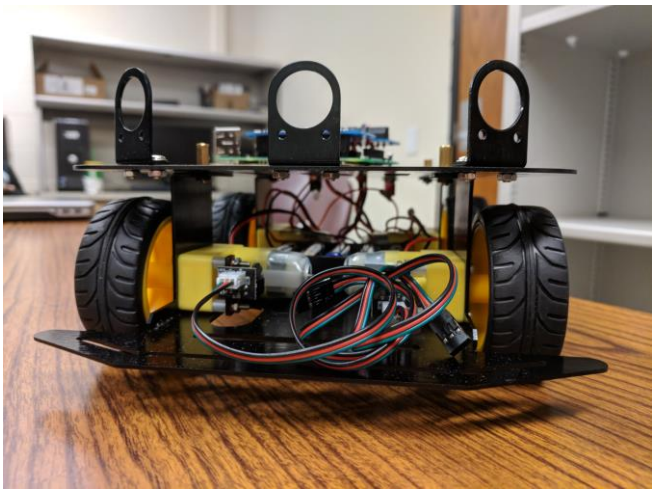
- **Minimum viable product available by mid-April**
- **Completed product available by May**
- **Robot showcases a degree of modularity**
- **There will be an accessible wireless network wherever this robot is being used**
- **Camera performs as described by manufacturer**
- **Robot will have full source of power**
- **Socket programming can be completed successfully**

3. ARCHITECTURE DESIGN

This section outlines the system and hardware architecture design of the system that is being built.

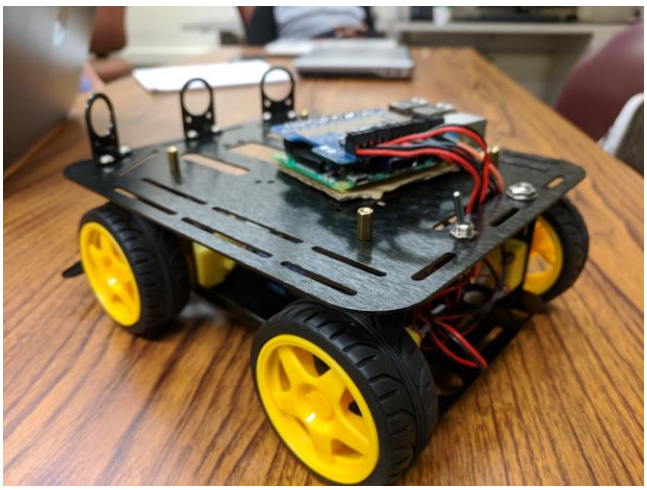
3.1 Hardware Architecture

The robot is composed of a DFRobot Mobile Platform chassis, Raspberry Pi with a motor Pi hat, an infrared PiCamera, and four motors (included with robot kit). It also includes five ElecRight sonar sensors with a range of approximately sixteen feet, a PWM expansion board for the Raspberry Pi, Adafruit Ultimate GPS, 10 kilogram straight bar load cell, and AA battery bank. Photos follow this description for the sake of clarity.

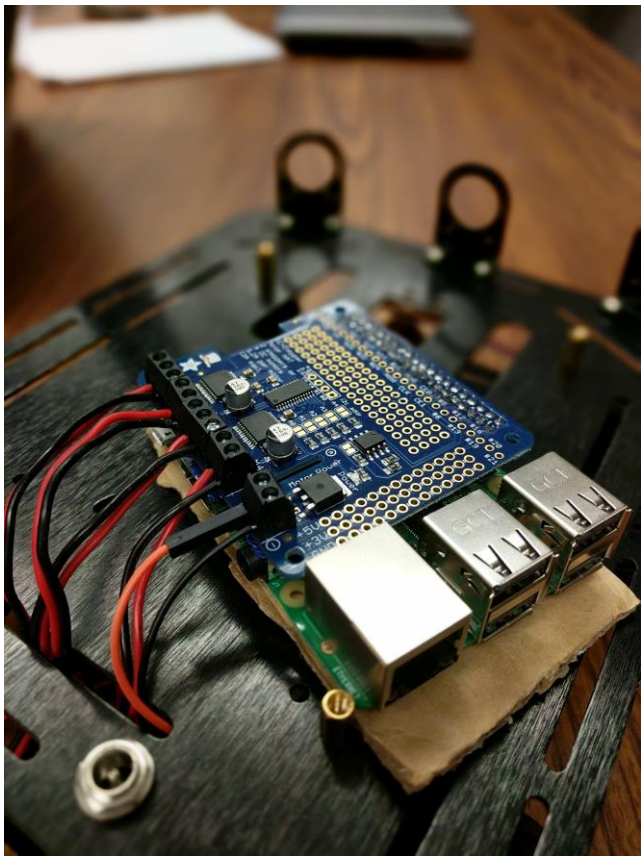


Front-left of robot with mounts for sonar sensors

Front of robot head-on



Aerial view of robot with Raspberry Pi



Rear view

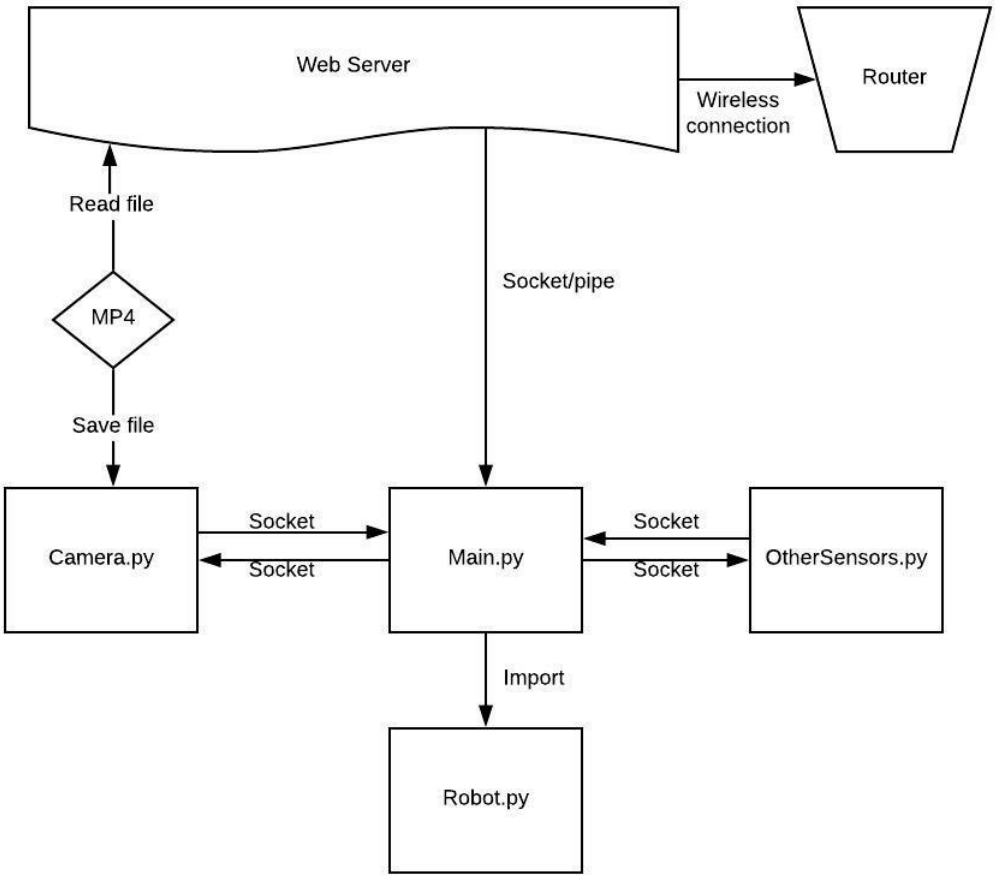
Raspberry Pi with motor hat

3.2 Software Architecture

- **Robot.py**: Firmware piece; contains all functions necessary to control the motion of the robot; to be imported into Main.py to be used as a library
- **Camera.py**: Contains all software needed for receiving/processing input from the camera and analysing the captured images
- **OtherSensors.py**: Contains all software needed for receiving/processing input from all sensors other than the camera (i.e. sonar, GPS, load cell); all sensors are controlled through a single process for the sake of optimization of the Pi's four cores.
- **WebClient.py:** Receives input from user to control robot

- **Main.py**: Forks Camera.py and OtherSensors.py; interprets input from those and WebClient.py and translates into robot locomotion using Robot.py

Software architecture is described by the following architecture diagram.



The import from Robot.py into Main.py was chosen because it will be utilized as a library to provide locomotive functions to the robot. The rest of the communication between Python processes will be completed via network sockets because it provides the most flexibility in data transfer.

All of the programming described above will be completed by the ModBot development team in the Python programming language. Camera.py will utilize the OpenCV libraries and will be responsible for communicating to Main.py whether or not the camera is attached to the modular robot.

3.3 Security Architecture

There are no relevant security architecture requirements for the ModBot project at this time.

3.4 Communication Architecture

Communication with the robot is completed over Wi-Fi, as the Raspberry Pi has a built-in Wi-Fi router. All of the processes will be connected to one another via the use of sockets, as previously mentioned.

3.5 Performance

- Sonars must be able to detect an obstacle within a maximum range of six inches and successfully avoid collision with that object.
- Camera shall be able to process images and provide a measure of distance/direction to aid in navigating to a target.
- Parallel processes can work properly in tandem

4. SYSTEM DESIGN

4.1 Use-Cases

- No camera attached: robot receives directional instructions via web application and moves around environment, avoiding obstacles through use of sonar sensors
- Camera attached: identify and follow visual symbol “signposts” though use of OpenCV libraries to autonomously navigate through a pre-specified path to reach and detect a target object

4.2 Database Design

ModBot does not require a database.

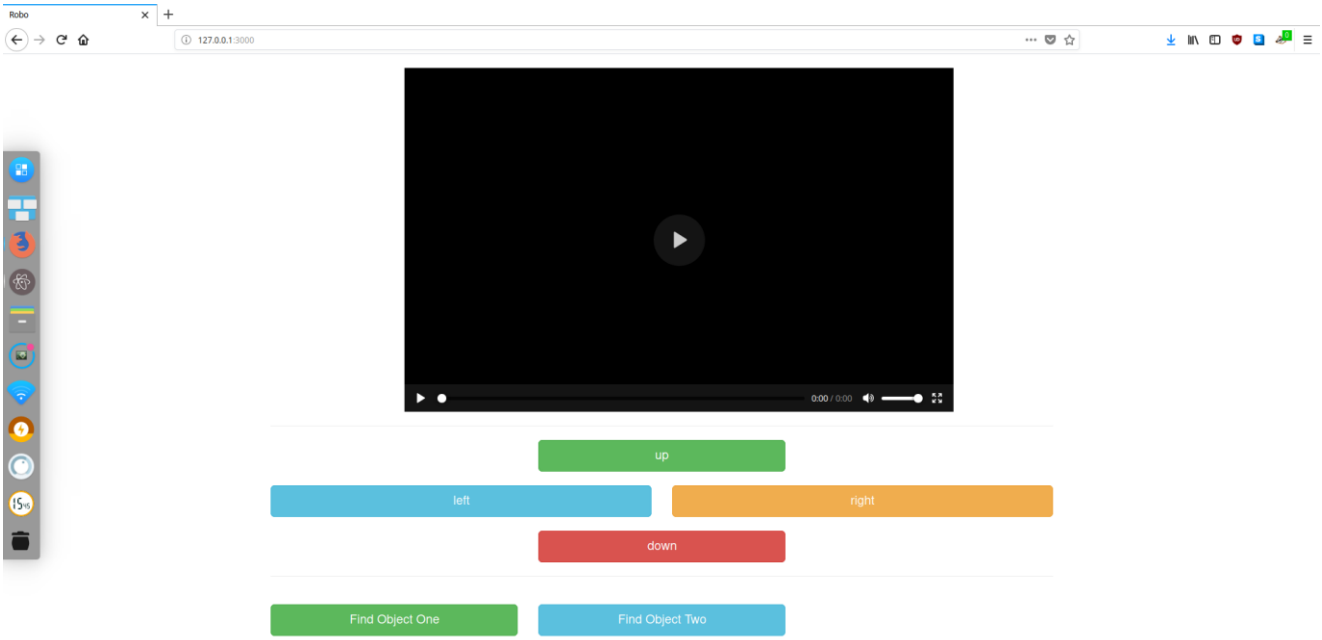
4.3 Data Conversions

ModBot does not require any data conversions.

4.4 Application Program Interfaces

OpenCV is the only API used in ModBot.

4.5 User Interface Design



Date: 2/26/2018

Approved by: _____

Approver Signature: _____

Mentor Name: _____

Mentor Signature: _____