February 13, 2022
Dr. Michael Hegedus
School of Engineering
Simon Fraser University
8888 University Drive
British Columbia, V5A 1S6

RE: ENSC 405/440 Requirements Specification for RSC-7 A1

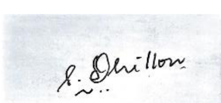
Dear Dr. Hegedus,

This requirement specifications document for RSC-7 A1 was prepared by Robo Snacks Company 7 for our Capstone courses. Our goal is to autonomously deliver snacks to conference rooms, networking events and presentations happening on university campuses and offices before an event begins. RSC-7 A1 is targeted towards catering services that deliver to such events and locations.

In this project, RSC-7 A1 would utilise LIDAR (Light Detection and Radiation) technology, ROS2 NAV2 and Raspberry Pi 4 for mapping and navigating a given area from the predefined 'Home' location to the requested end location. The requirements specification entails a project overview for the proof of concept, prototype and final prototype stages planned for RSC7-A1. The document includes rough diagrams for both the prototype and final product as well as a system block diagram. It will also cover general requirements with more detailed requirements specified in the software, system, electrical and structure sub-parts. All requirements are clearly labelled as either proof-of-concept, prototype, or final product as defined in the project overview section.

We would like to thank you in advance for taking your time to review this document. If you have any questions, please email us at skd24@sfu.ca or ikomolaf@sfu.ca.

Sincerely,



Sirpreet Kaur Dhillon & Emmanuel Komolafe

Co-CEOs

Robo Snacks Company 7 (RSC-7)

2/16/2022

Self-Driving Snack Carrying Robot: RSC-7 A1

Requirements Specification

ROBO SNACKS
SNACK DELIVERY ROBOTS

Robo Snacks Company 7 (RSC7)
COMPANY 7

The Robo Snacks Company 7 Team

Authors (Team Members)		**Roles / Affiliations**
Full Name	**Email**	
Favour Amah-Nnachi	famahnna@sfu.ca	Firmware Lead
Sirpreet Kaur Dhillon	skd24@sfu.ca	Automation Team, and Structure Team
Emmanuel Komolafe	ikomolaf@sfu.ca	Systems Lead, and Automation Team
Veronica Lund	vlund@sfu.ca	Automation Team
Robert Smyczynski	rsmyczyn@sfu.ca	Electronics, and Structure Lead
Eddie Zheng	eza7@sfu.ca	Automation Lead

Abstract

This document contains the requirement specifications for the Self Driving Snack Delivery Robot (RSC-7 A1), a semi-autonomous robot that will be delivering snacks to conference rooms, networking events and presentations happening on university campuses and offices before the event begins. By "snacks" we are referring to light ready-to-eat foods that are eaten in between meals. The robot would have features that enable it to move from a home location to a specified conference or meeting room, hold at that location for a specified period of time, allow attendees to retrieve snacks from the robot's compartment, and then return to the home location, all autonomously.

The robot will be a single unit comprised of three systems. Firstly, the structural and mechanical system, including the frame, wheels, and snack bin. Second, the electrical system including the battery, power regulating circuit, and electric motors. Lastly, the sensors, microcontrollers, and software comprising the brain of the robot.

This document will start with an overview of the main purpose, goals, problem being solved and detailed information about the functionality of our robot. The main components of the document are:

- Project Overview - provides adequate information to the reader of what makes up the robot and what the robot interacts with so that they have a clear picture of the system.
- Requirements - describes the design inputs of the robot and conditions needed to be met to fulfil its purpose as specified by our company. The requirements are divided into general, software, system, electrical and structural.

List of Tables and Figures

Contents

Glossary

Term	Definition
C	Final Product
B	Prototype
A	Proof of Concept
RSC-7	Robo Snacks Company 7
RSC-7 A1	First semi-autonomous robot design by RSC-7
LIDAR	Light Detection and Ranging

1. Introduction

In 2019, Bloomberg reported that as many as 120 million workers in the U.S. may need to be retrained due to development in artificial intelligence [1]. This may sound like a fatalistic interpretation, but there is no doubt that the emerging fields of artificial intelligence and autonomous robots will have a major impact on our lives. RSC-7 aims to explore the emerging market of using autonomous robots to assist in and perform basic tasks by introducing the Self Driving Snack Carrying Robot (RSC-7 A1). The main goal of our company is to deliver snacks and drinks to conference and meeting rooms by navigating on demand from a predetermined staging point to a specified room location where attendees of the meetings have access to these refreshments at their convenience.

RSC7's primary customers will be vending and catering services. The demand for these services is high so they are well suited to use the RSC-7 A1 to serve their existing customers more efficiently. The robot would provide services without the need to set up additional static vending machines or sending people directly to deliver the snacks. It would also not need to navigate around humans as delivery would be early in the mornings or before guests arrive for their meetings. The need to avoid humans is based on the fact that humans are unpredictable and can change motion at any given time.

2. Project Overview

In order to give a picture of what approach and plan we will be using for the design process of our system we are presenting an overview of the Self Driving Snack Delivery Robot.

2.1 Proof of Concept Stage

In this stage, we will be designing and testing a proof of concept to help realise the feasibility of building a robot, integrating robotic capabilities, and demonstrating some of the potential high-risk areas and how to overcome them.

During this stage, these are the following areas that we would explore:

- **Mobility:** Prove that the robot can move in several directions and turn when needed using ROS NAV2 stack.
- **Software:** Determine how to run ROS NAV2 stack on the Raspberry Pi.
- **Communication:**
    - Determine how to connect the robot to the Raspberry Pi via wireless communication.
    - Determine how to send and receive signals between the Raspberry Pi and the robot's Arduino.
    - Determine how the motors of the robot are turned OFF/ON. (iv) Determine how to keep the motors of all wheels at the same revolutions per minute (rpm).
    - Determine how the Raspberry Pi receives signals from LIDAR in real time
- **Payload Handling:**
    - Prove that the robot can handle a weight of 50lbs
    - Determine how much power is needed to keep the robot running for 24hrs.
- **Observations:** Determine how to put the robot on stand-by mode.

2.2 Prototype Stage

In this stage, we will be building a slimmed-down version of our final robot based on our findings from the proof of concept. The designed prototype will be tested for its functionality and usability. The end goal for this stage is to provide early feedback on the design process and have a prototype with the essential features.

A robot is a machine that performs complex tasks and decision making, in order for them to do so they require a number of sensors and actuators for interpreting the environment. Therefore, for the design process, we will be using the Light Detection and Ranging (LIDAR) sensor that makes use of a laser to measure how far apart objects are. The robot will use LIDAR for scanning its environment, creating a map of its environment and use that information to make important decisions like how to navigate and where objects are [5].

We will require a brain for the robot. A control system that provides feedback for determining the behaviour of the robot. The control system will use information from the sensors and act accordingly. We have chosen to use the Raspberry Pi microcontroller that runs a Linux operating system as the electronic brain of the robot [6].

We will require a chassis that will house motors, batteries as well as other electronic components and provide the robot with support needed to carry the snacks [6].

For the robot to move, we will require DC motors. Because motors require a high current, we will use a motor driver to convert low current from the Raspberry Pi to high current that can drive the motors. The motor driver will be connected to the motor, Raspberry Pi and power supply [6].

For the wheels of our robot, we will be using three omni wheels. Omni wheels come with special rollers that provide the robot with four degrees of freedom (left, right, forward, and backward), control and stability. The wheels will be mounted to the chassis of the robot [7].

Lastly, we will require a portable power source which includes a number of batteries and power banks that will provide enough power to the robot for 24hrs [6].

Figure 1 introduces the basic design of the prototype chassis. It includes the three omni wheels discussed earlier, space/compartment in the bottom layer to house the motors, motor drivers, microcontrollers, etc. For the snack compartment, the design includes a cylindrical container that is mounted on top of the chassis.

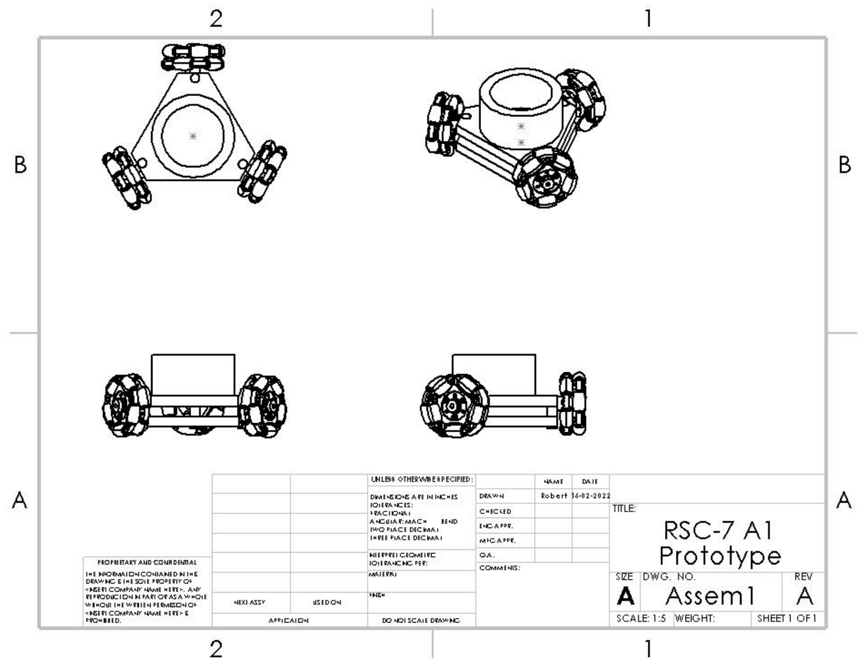


Figure 1: Schematic of initial prototype design for RSC-7 A1 system

In our design process of the prototype, we would take into consideration that the prototype will be tested for the following at the end of this stage:

- Robot can move while carrying a load of 10lbs
- Robot can make turns when necessary.
- Robot can move from location A to B
- Robot has a map of its environment.
- Robot's electrical components are properly shielded.
- Robot's body is not easily destructible.

2.3 Final Product Stage

In this stage, we will be building on and refining the prototype to create our final product. The prototype will be optimised to include all specified features.

Finally, the final robot product will undergo a series of scenario testing for functionality, performance, and usability.

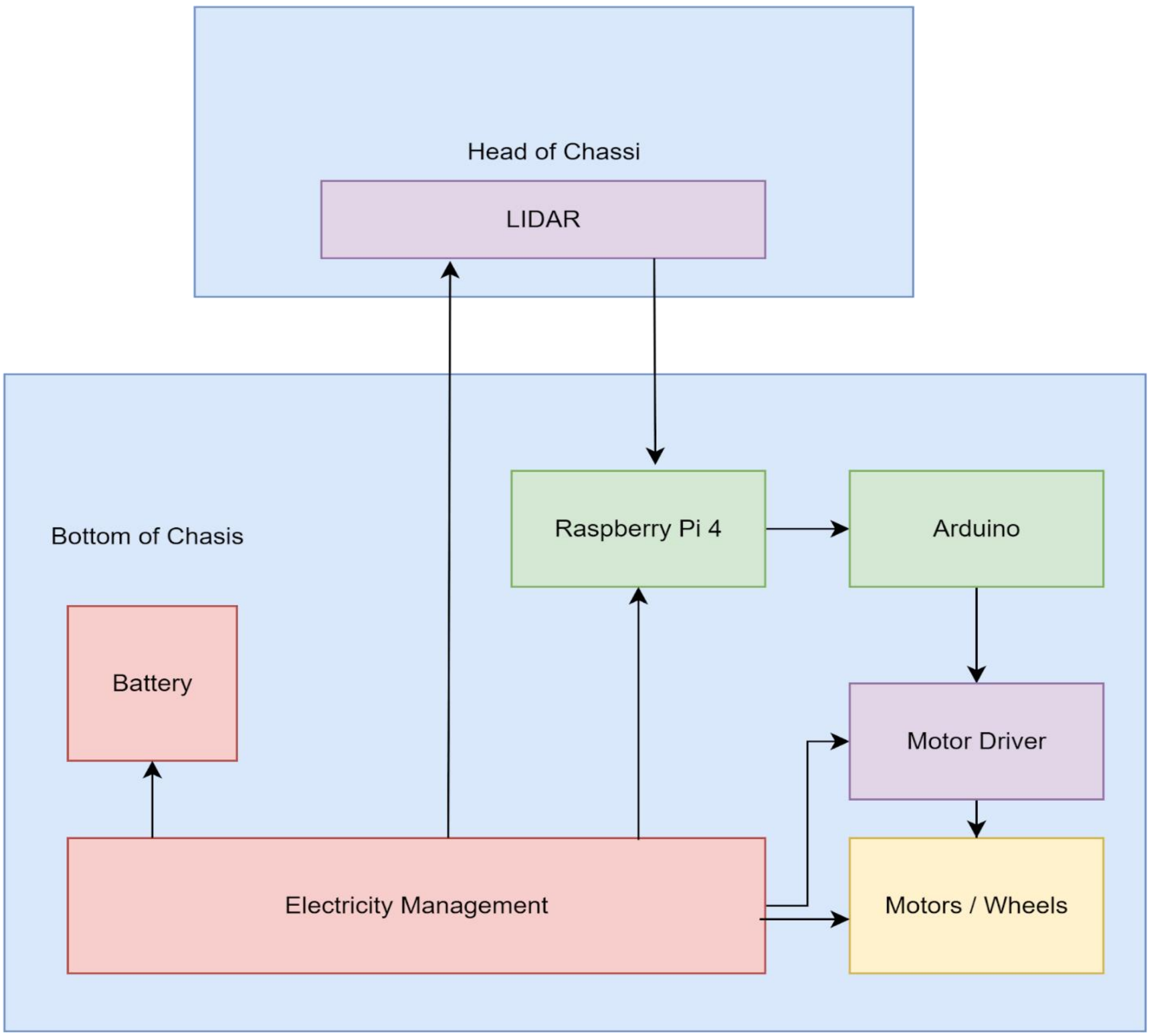


Figure 2: Block diagram of RSC-7 A1 System

3. General Requirements

This section will include all the general requirements of the snack carrying robot, RSC7 A1. These requirements outline the system as a whole and entail the things the robot should do all together.

The general requirements are a result of all the other requirements that are discussed in the subsequent sections and so for the general requirements to be met, the corresponding software, systems, electrical and structural requirements need to be met first.

The requirements are clearly identified with a Requirement ID and a product stage:

- Proof of Concept - A
- Prototype - B
- Final Product - C

Requirement ID	**Product Stage(s)**	**Requirement Description**
3.1	A, B, C	Robot should be able to move forward, backward, left, right and make turns when necessary
3.2	A, B, C	Robot should be able to accept the delivery location from the user
3.3	B, C	Robot should be able to select the best path from preloaded map
3.4	A, B, C	Robot should be able to approximate its general location at any given time
3.5	B, C	Robot should be able to start its journey and navigate to the destination address effectively (avoid getting stuck or bumping into things and avoiding obstacles like stairs, pillars, etc)
3.6	B, C	Hold food and drinks: 10 lbs
3.7	B, C	Robot should be able to move around with the maximum food capacity

3.8.1	C	After reaching the delivery location, the robot will initiate stand-by-mode
3.8.2	C	At the end of the day, the robot will exit stand-by-mode and start its journey back to the set 'Home' base/location
3.9	C	Robots will have an automatic locking and unlocking system connected to the stand-by and power management system
3.10	C	Robot will have a sanitizer unit attached to it to promote sanitization and prevent the spread of germs and disease

Figure 3: General Requirements Table

4. Software Requirements

For software, we will leverage the ROS2 NAV stack (Robot Operating system) on a Raspberry PI 3. Raspberry Pi and ROS both have a full-fledged ecosystem and are highly used in the industry and therefore we choose this combination.

To control our robot, we will be developing the controls for robots using Linux OS - Ubuntu Mate 18.04 as it is easier to use ROS2 with Linux type operating system [3][4].

For the proof-of-concept stage, we will learn how to use the ROS NAV2 stack to make a robot move in all directions.

For the prototype stage, we will have the robot navigate to the room using the ROS NAV 2 stack. The robot should avoid obstacles and reach the destination.

For the final product, we will have the robot navigate to the room and return to base once the timer is over. The robot should be able to make multiple trips and have a fallback algorithm for when dealing with unknown situations.

The following table includes the requirements for Proof of Concept, Prototyping and final product.

Requirement ID	**Product Stage(s)**	**Requirement Description**
4.1	A	The robot must be able to run ROS2 NAV stack on raspberry pi and will be connected via Wi-Fi for programming
4.2	A	The robot must be able to move in all directions on the ROS2 NAV stack tested using keyboard keys to navigate left, right, forwards and backwards.
4.3	A	Stand-by mode (a low power mode where only the clock is operational, and the lock system is operational in final product stage)
4.4	A	Using ROS2 NAV stack, we should be able to communicate to all parts wirelessly
4.5	B	The algorithm to navigate from A to B and back to A will be programmed
4.6	B	Using ROS2 NAV stack the robot should be aware of the map
4.7	C	Using ROS2 NAV stack, the robot should have a fail and fall back plan to return to base when dealing with unfamiliar situation
4.8.1	C	The robot will go into stand-by mode once it reaches the destination location
4.8.2	C	The robot will come out of the stand-by mode (see requirement ID: 4.3) “at the end of the day” (after a set number of clock cycles)

Figure 4: Software requirements table

5. System Requirements

The system requirements section includes all the assembly, design and setup requirements around the selected microcontrollers, sensors, the motor driver, and motors used in the robot.

For the system design, we have decided to split our work into computing intensive tasks (lidar, sensors, automation) performed by the raspberry pi 4 and the Arduino for controlling tasks (motors/motor drivers).

Requirement ID	Product Stage(s)	Requirement Description
5.1	A	Raspberry Pi and Arduino should be able to send and receive signals between each other
5.2	B	Arduino should be able send signals and switch the motors on and off and pick the direction of rotation
5.3	B, C	The motors connected to omni wheels should function simultaneously and as defined/designed
5.4	B, C	LIDAR signals should be received by raspberry pi and processed in real time

Figure 5: Systems requirements table

6. Electrical Requirements

Hardware is a key aspect of our design, as the robot will be fully reliant on battery power for its normal operation. Our device must reach the minimum electrical requirements to drive the motors and other control devices. The robot will need to have safety standards comparable to an indoor environment and include full shielding of circuitry to prevent any harm to the user. For a longer lifespan, the design will include rechargeable batteries that will last for a full day of operation on a single charge. To ensure complete reusability, the final circuit must include overvoltage and current limiters to prevent internal damage to other circuit elements. The basic design will include electric components such as a Raspberry Pi 4, DC motors, LIDAR sensors, and a power supply. More components may be included in the final design.

Requirement ID	Product Stage(s)	Requirement Description
6.1	A	Power supply must provide enough power for a full day of operation
6.1.1	A, B, C	Power supply must be properly shielded from external conditions
6.2	A	Raspberry Pi needs between 3.3V to 5V to operate
6.3	A	Circuitry must be separated from food storage area
6.4	B	Connector wires cannot come loose during operation
6.5	B	Power supply must be over 12V and 6 Ah to provide sufficient voltage to the motors
6.6	B	Overvoltage and current limiters must be present in the final design for safety measures
6.7	C	Electrical system will be waterproofed by encasing in a plastic box with only the required parts exposed

Figure 6: Electrical requirements table

7. Structural Requirements

For our device to generate revenue, it must hold a certain amount of snacks so various catering order sizes can be fulfilled. Mechanically, the robot will use electric DC motors for movement and will be required to be powerful enough for the device to travel to the meeting rooms at an adequate speed while holding the cargo.

Our robot will be set in an indoor environment but must meet structural requirements so the catering company will be able to transport it between customer and business. Therefore, the requirements put limits on the weight of the whole system and the supporting structure, but while also balancing functionality. Since we are planning for the robot to be reusable and have a long lifespan, the final materials used must withstand time and the environment.

Requirement ID	Product Stage	Requirement Description
7.1	A	Chassis must be able to support up to 50 lbs of weight [2]
7.2	B	Initial structure must be rigid enough to survive impact with hard objects
7.3	B, C	Must travel faster than a metre per minute
7.4	B, C	Final design must include a combination of plastic and metal for the outer structure
7.5	B, C	Device should not exceed 100 lbs in weight
7.7	B	Sensors on the external body of the robot will have proper protection in case of sudden impact
7.8	C	Locking mechanism to ensure protection of cargo
7.9	B	For waterproofing the prototype, the snacks will be put in a plastic basket which will be secured to the structure
7.10	C	For waterproofing the final product, a box type of lockable structure will be secured onto the final chassis

Figure 7: Structural requirements table

8. Conclusion

The purpose of the RSC-7 A1 is to provide university campuses, caterers, and vending companies with a fully automated refreshments delivery service that can deliver products to attendees on demand with no human efforts required outside of stocking and servicing the robot itself. The autonomous service robot market is an exciting new emerging market, and we at Robo Snacks Company 7 believe that in the near future autonomous robots will perform and assist in countless routine tasks for people of even modest economic means.

RSC-7 A1 will have an efficient, minimalist design with cost-effective components to help with cost and ease of development, giving potential customers an effective, entry-level product with which they can trial the exciting, cutting-edge service of autonomous robot refreshment delivery. RSC7 has carefully selected a proven, economical microcontroller to guide the robot's vision and navigation, and will utilise a sleek and simple structural design to simplify this navigation and make it robust.

Using this requirements specification as a roadmap, RSC7 aims to embrace the technological future and strike out into an emerging new market of endless possibility. Our design will allow the robot to safely and seamlessly integrate into and navigate its environment as it travels to and from its designated delivery point to distribute much needed refreshments to conference and meeting attendees initially, and eventually to the world!

References

[1] "More Robots Mean That 120 Million Workers Will Need To Be Trained," *Bloomberg.com*. [Online]. Available: https://www.bloomberg.com/news/articles/2019-09-06/robots-displacing-jobs-means-120-million-workers-need-retraining. [Accessed: 13-Feb-2022].

[2] "Omni Robots," 18 July 2019. [Online]. Available: https://omni-robots.com/products/203mm-8-inches-omni-wheel/. [Accessed 12 February 2022].

[3] "docs.ros.org," [Online]. Available: https://docs.ros.org/en/rolling/Installation/Ubuntu-Install-Binary.html. [Accessed 10 February 2022].

[4] "Robotic Back End," [Online]. Available: https://roboticsbackend.com/install-ubuntu-mate-18-04-on-raspberry-pi-3-b/. [Accessed 10 February 2022].

[5] "The Basics of LiDAR - Light Detection and Ranging - Remote Sensing" [Online]. Available https://www.neonscience.org/resources/learning-hub/tutorials/lidar-basics [Accessed 10 February 2022]

[6] "What You Need to Build Your Own Autonomous Robot" [Online]. Available https://www.makeuseof.com/what-you-need-to-build-your-own-autonomous-robot/amp/ [Accessed 10 February 2022]

[7] "Build a Robot - Part 4 - Selecting wheels" [Online]. Available https://dronebotworkshop.com/real-robot-004 [Accessed 10 February 2022]