Alex to the Rescue

Design Report

Team: B04-3B

Table of Contents	
System Functionalities	➢ Remote Controllability ➢ Movement ➢ Mapping
Review of the State of The Art	➢ Spot ➢ PackBot
System Architecture	➢ LiDAR ➢ Motors ➢ Wheel Encoders ➢ Power ➢ Ultrasonic Sensor ➢ Color Sensor
Component Design	➢ Initialization ➢ Receiving of user commands ➢ Execution of user commands ➢ Mapping of environment ➢ Color Detection
Project Plan	➢ Project Goals and Timeline
References	

Section 1 System Functionalities

Alex is a robot built to be able to navigate a simulated "rescue environment", controlled remotely by an operator and his laptop. The robot's core is the Raspberry Pi 4, an Arduino Uno, a LIDAR, and 2 motors and wheels. The primary features that Alex needs to be able to have are, Remote controllability, map "vision" and simple 4 directional movement.

Remote Controllability

Alex needs to be able to be maneuvered by an operator without a direct view over Alex and the environment, by sending instructions wirelessly, over Wi-Fi. Our choice of method to accomplish this is through VNC on the operator's laptop, to the Pi. The Pi would then communicate these instructions to the Arduino Uno, which in turn would control the motors and LIDAR appropriately. We chose VNC as it was the simplest to set up and loved the convenience that the file transfer GUI of VNC brought, as well as the fact that there were minimal delays between the sending of commands and execution.

The LIDAR would be used to "image" Alex's surroundings, to be viewed by the operator such that they would be able to issue accurate instructions to Alex.

Movement

We want Alex to be able to move forward, backwards, as well as turn left and right. Each of the 4 movement instructions take 2 arguments. The first being distance, in the case of forward and backwards, and angle, in the case of turning left and right. The second being the speed, which is a percentage out of 100 at which to execute the movement instructions. The distance and angle are kept track of by Hall Effect sensors attached to each of the 2 motors, these sensors detect changes in magnetic flux caused by the rotation of small magnetic discs attached to the motor. We keep track of each of these changes as "ticks" and with some simple measurements of the circumference of the wheels, as well as how many ticks correspond to 1 full revolution of the wheel, we are hence able to accurately issue instructions to Alex, without direct vision over him.

Mapping

To provide information to the operator, to both issue instructions to Alex as well as to allow the operator to complete the map of the environment, we are using the LIDAR to gather information about the obstacles in the environment, which is then converted into a readable map through Hector SLAM (Simultaneous Localization and Mapping).

Section 2 Review of State of the Art

Alex is a robotic vehicle used for search and rescue purposes that can navigate dangerous locations and rescue victims (by identifying different colors) at the same time. We investigated existing robots to optimize the design and usefulness of Alex.

Spot: is a police and fire division robot that uses LiDAR to create a 3D map of the environment and to detect obstacles and navigate autonomously in complex environments. It is equipped with four LIDAR sensors that give a 360-degree view of its surroundings, allowing it to recognize objects in its path and adjust its movements accordingly. It also has a front-facing camera that provides a high-definition video feed of the robot's surroundings and can be used to identify people or objects in need of assistance and remotely assess conditions in hazardous environments before human rescue teams are dispatched. Spot has 4 legs that make it able to climb stairs, traverse rough terrain, have a stable base and maintain balance on uneven or unstable surfaces.

Spot’s strengths include versatility which means it can be customized with sensors and connections, allowing it to explore challenging situations and perform complex moves. It can be controlled remotely using a tablet or other mobile device, allowing the user to control the robot from a safe distance. Weaknesses include cost and limited battery life. Spot's battery life is currently limited to around 90 minutes of continuous use, which may be a limitation in some applications.



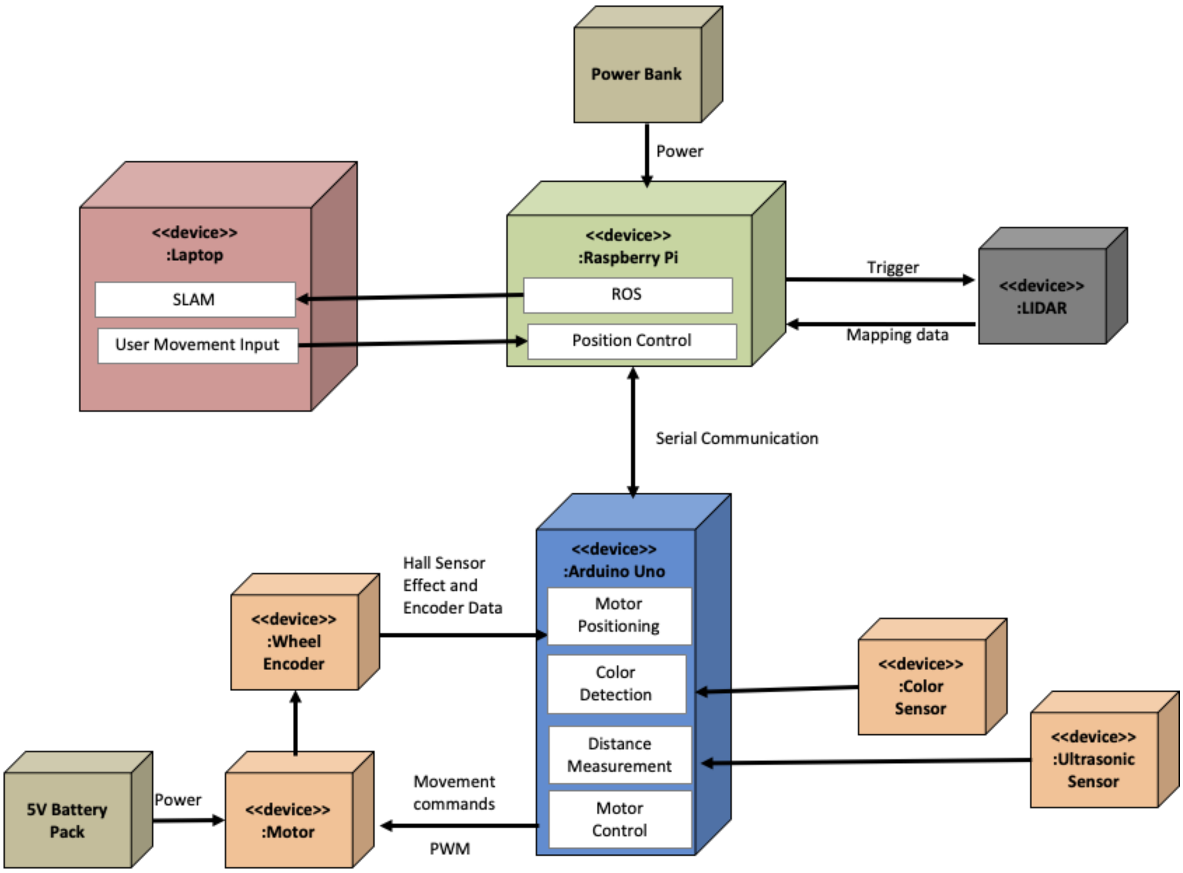
PackBot is a robot used by the US military for explosive ordnance disposal, reconnaissance, and surveillance, for bomb disposal, hostage situations, and search and rescue operations by law enforcement and for emergency responders in the aftermath of natural disasters, such as earthquakes and hurricanes, to search for survivors and assess damage by Disaster Responders. It’s equipped with several cameras, including a color zoom camera, a low-light black and white camera, and an infrared camera. It also uses LIDAR technology, to build a 3D map of its environment. Additionally, the PackBot has a manipulator arm which can be used to pick up objects or move debris out of the way. This arm is controlled remotely by a human operator, who can use the robot's cameras and LiDAR sensors to guide it to the target object.

The strengths of PackBot include, high mobility, multiple cameras and LIDAR sensors, remote control, navigation through rough terrain, climbing stairs, move through rubble and debris and presence of manipulator arm that picks up objects and move debris out of the way, making it useful for clearing paths and accessing confined spaces. However, some weaknesses include, limited payload capacity, limited battery life and vulnerability to extreme temperatures can limit its effectiveness in certain environments.

Section 3 System Architecture

Alex is built around three software components: the Arduino, the Raspberry Pi, and the operating laptop. These parts allow interaction with hardware components such as motors, hall sensors, LiDAR, colour sensors, and ultrasonic sensors. The power bank powers the RPi, which is directly connected to LiDAR and gets data points about Alex's surroundings for mapping. Shown below is the interactions between different devices.

<u>UML Development Diagram</u>



UML development diagram

<u>LIDAR</u>

The LIDAR can map out the surrounding environment of Alex by sending out pulses of light, while the detection unit attempts to capture the reflected returned light pulse with which the distance of the point is determined.

Thus, the LIDAR component is mounted on Alex and is directly connected to the Raspberry Pi. This data is then sent to Computer through a secure TLS connection. The computer then uses its 64-bit processor to map the surroundings and provides accurate live plots of data visually through Rviz using the Hector SLAM algorithm.

Motors

2 Motors of 48:1 gear ratio will be used to control Alex's movement. The motors will be connected to the Arduino Uno digital pins, and the Arduino will establish serial communication with Pi through the USB port to side-step the trouble of having to use a voltage shifter to accommodate the different voltages of the two systems.

After determining the position of Alex and the surrounding environment via the SLAM algorithm, movement stored as command packets will be received and executed by Arduino from Pi. These movement will be sent to the Pi via the laptop. As the TLS client terminal in the laptop will be remotely connected to the TLS Server on the Pi and is the main platform that controls Alex.

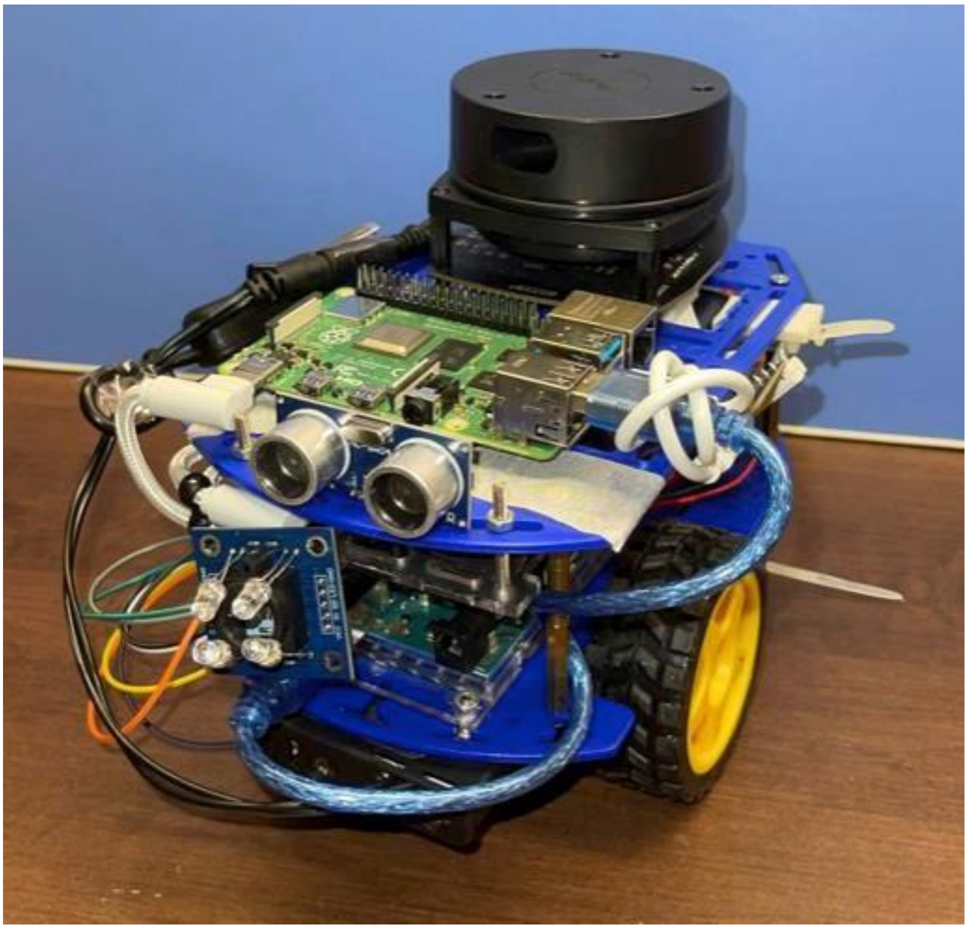
<u>Wheel Encoder</u>
2 Wheel encoders will be fastened, one to each motor. The wheel encoders can measure the revolutions of the wheels. After some calculation with the circumference of the wheel we have data on the distance travelled by Alex.

<u>Power</u>
For both motors, they will be powered by the 5V battery pack. The Pi however will be powered by the power bank instead and the Pi will power the Arduino Uno.

<u>Ultrasonic Sensor</u>
Alex will utilize 2 Ultrasonic Sensors by connecting them to the Arduino. The sensors will be mounted to the sides of Alex and provides data on the distance Alex is away from any obstacles. Should the data suggest Alex is in too proximity from an obstacle, Alex would be able to stop and avoid any possible collision. Thus, this added “safety net” allows Alex to maneuver the environment safely.

<u>Color sensor</u>
This is used in the “rescue” part of Alex’s mission. The sensor is connected to the Arduino and is in front of Alex facing away from him. It is used to sense the color of the object in front of Alex, when the LiDAR maps an unusual shape which is not a wall.

Section 4 Component Design



Physical implementation of

Alex The high-level steps to achieve the search and

rescue:

1. Initialization of system
2. Receive User Movement Command
3. Execute User Movement Command
4. Repeat Step 2 and Step 3 until the environment has been completely mapped out
5. Detection of

Color Step 1:

Initialization

For Alex to maneuver about the maze, movement commands will be sent from the PC to the RPi, which will then be sent to the Arduino and executed. A

Transport Layer Security (TLS) server is set up to enable secure communication between the PC and Alex. Data packets are sent from the RPi to Arduino via a USB Cable.

Essentially, the RPi on the Alex is accessed from our laptop via SSH, after which the TLS server is started. After an initialization handshake, the RPi and the Arduino are ready to receive and transmit data packets. On the RPi, the ROS RPLiDAR package is launched. With that, the ROS SLAM package on the remote laptop is launched to generate a map based on data from the LiDAR. The TLS client is started on our laptop and we establish a connection with the RPi. Henceforth, Alex is initialized and ready to navigate.

Step 2: Receive User Commands

After initialization, we can issue commands to control Alex. The movement will be controlled by the laptop keys ‘W’, ‘A’, ‘S’, ‘D’ and ‘T’ with W being forward, ‘A’ being left,‘S’ being backwards and ‘D’ being right, and “T” for stop. A slight change from the code provided in the studio, which uses “F” “B” “L” “R” . This is to provide a gamified experience to the operation of the Alex robot, allowing a more intuitive and natural and hence easier experience while maneuvering the robot.

The TLS server allows direct and secure communication between our laptop and RPi. We generate a Certificate Authority (CA) certificate which has a public key that clients use to verify and connect to the TLS server.

Data packets sent from the RPi to the Arduino will have the data structure called ‘TPacket’. Data packets are serialized into an array since serial devices can only read and receive bytes. Data packets are thereafter deserialized to reconstruct the information in the initial data packet.

Step 3: Execute User Commands

When a command is issued, the Arduino will execute by controlling the motors. Forward: Left motor rotate clockwise direction, right motor rotate anticlockwise direction

Backward: Left motor rotate anticlockwise direction, right motor rotate clockwise direction Left: Right motor rotate anticlockwise direction

Right: Left motor rotate clockwise direction

Step 4: Mapping of environment

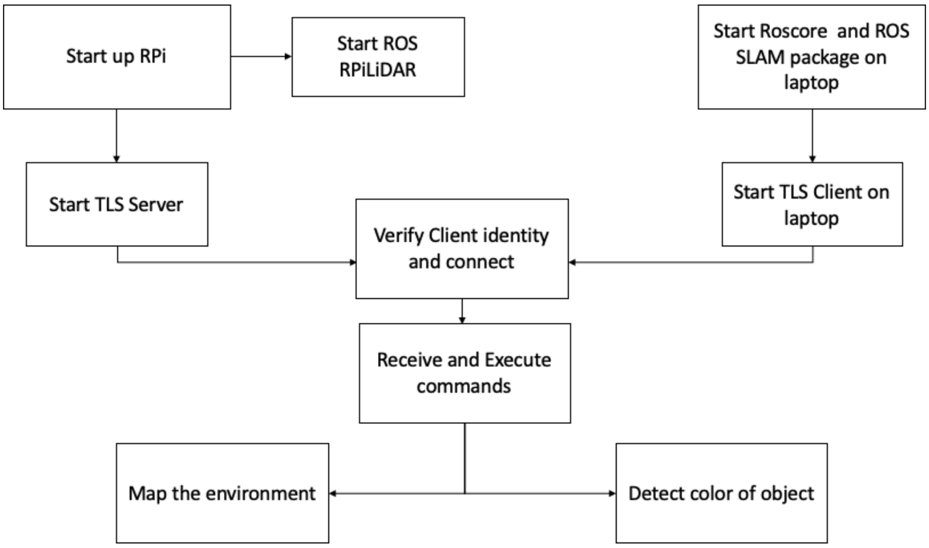
Steps 2 and 3 will be repeated until the environment has been completely mapped out.

Step 5: Detection of Colour

Upon detection of an object, the TCS3200 color sensor will run. The TCS3200 sensor contains arrays of photodiodes covered by a color filter, which allows only a certain range of wavelengths through. The sensor measures the intensity of the light that is reflected from the object. The reflected light passes through a small aperture and is then directed onto the photodiode array.

The sensor outputs these intensity values as a set of pulse-width modulation (PWM) signals. The Arduino will read these signals using digital input pins and convert them into actual intensity values using analog-to-digital converters (ADCs).

These intensity values are then transmitted to the RPi via serial communication, where the data packets are serialized, transmitted and thereafter deserialized to reconstruct the intensity values. The RPi will then determine the color of the object and relay the color back to the operator through the TLS server.



Overview of Alex functionalities

Section 5 Project Plan

Week No.	Goals
Week 8	Assemble basic version of Alex, with minimal parts, such as the Arduino, Pi, LIDAR, as well as wiring up motors and wheel encoders.
Week 9	Establish and ensure that serial communications are working between the Arduino and the Pi, and that instructions sent from the Pi are read and executed as expected on the Arduino.
Week 10	Calibrate and tune motor movement to match instructions from pi, as well as test out LIDAR.
Week 11	Calibrate and tune LIDAR and add color and ultrasonic sensors.
Week 12	Trial runs and further fine-tuning, as well as convert code for Arduino to bare metal.
Week 13	Final Trial run and final minor touch-ups and finetuning

Reference

Gossett, S. (2022). *12 examples of Rescue Robots you should know*. Built In. Retrieved March 31, 2023, from https://builtin.com/robotics/rescue-robots

Dynamics20, B. (2023). *About*. Boston Dynamics. Retrieved March 31, 2023, from https://www.bostondynamics.com/about#:~:text=Police%20departments%20use%20Spot%20to,responders%20and%20the%20public%20safe.

Dynamics, B. (2023). *Spot® - The Agile Mobile Robot*. Boston Dynamics. Retrieved March 31, 2023, from https://www.bostondynamics.com/products/spot#id_second