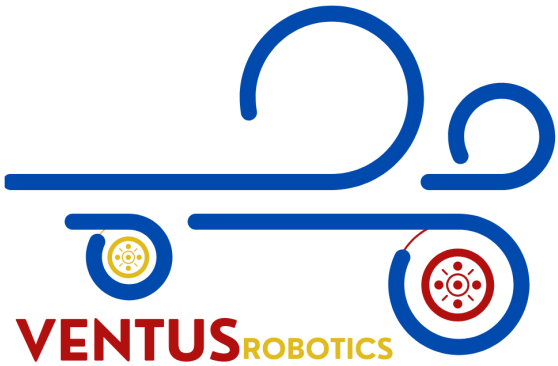
Technical Update 1

Mobile Robot for Remote Indoor Monitoring of Buildings
MREN 203



Ventus Robotics (Team 7)

Sabrina Button, CTO Embedded Software and Control - 20265627
Luke Major, CTO Software Integration and Interfacing - 20225654
Daniel Dubinko, CTO Hardware Integration - 20229482

Date: February 5th, 2023
Client: Queen's Project Management Consulting (Q-PMC)

We do hereby verify that this written report is our own individual work and contains our own original ideas, concepts, and designs. No portion of this report has been copied in whole or in part from another source, with the possible exception of properly referenced material.

Preface

Ventus Robotics is pleased to present the following technical update on the rover for remote monitoring of air quality on behalf of Queen's Project Management Corporation (Q-MPC). This update highlights the advancements made in the areas of hardware development, software development, and research, challenges encountered, and a provisional plan for future development.

1 Technical Achievements

1.1 Driving Via a Microcontroller

A top view sketch of the components on the Lynxmotion Aluminum 4Wd1 Rover is shown in Figure 1. The motors have a determined gear ratio of 30:1. The Arduino supports a maximum effective delay of 15ms.

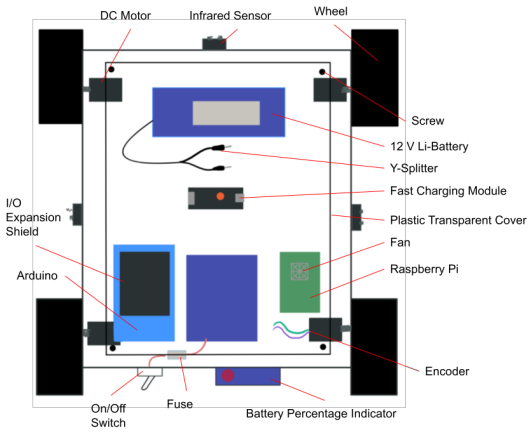


Figure 1: A diagram of relevant components in the rover.

Pulse width modulation (PWM) controls a motor's turning rate by writing a high signal and then a low signal each respectively lasting different time lengths. The percentage of time the signal is high is referred to as the duty cycle; a plot of a 0-5 V PWM signal with a 66% duty cycle is shown in Figure 2.

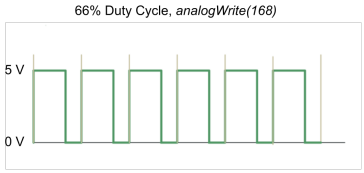


Figure 2: A diagram of a 66% duty cycle PWM signal.

PWM control can be achieved using a motor driver. The pinout connections between the motor driver and the Arduino are described Appendix 1, accompanied by a diagram of the motor driver board. The right motors are supported by port A and its corresponding pins, and left by port B.

It was noted that for the duty cycle to control the motors, continuous digital writes are necessary each digital write denotes a single "cycle". The value written to the EA and EB pins on the motor shield determines the time interval for which the cycle will be high (0 to 255). The motor driver has four pins - I1 I2 for the left motors' directions, I3 I4 for the right motors, and EA EB for controlling the left and right motor duty cycle. A table relating simple movements to motor driver inputs can be found in Table 1.

Analog Pin	FORWARD	REVERSE	RIGHT	LEFT
I1	0	1	0	0
I2	1	0	1	0
I3	0	1	0	1
I4	1	0	0	1

Table 1: A description of the appropriate pinouts to achieve various simple movements using the motor driver.

1.2 Proprioceptive Sensors

The Ventus rover is equipped with proprioceptive sensors for self-awareness and feedback control. It has a rotary optical encoder and an inertial measurement unit (IMU) with a gyroscope and accelerometer. The rotary optical encoder reads 3000 ticks for one rotation of the rover's wheels. Based on the frequency of the ticks, angular rate can be determined and then converted to linear velocity. To determine rotational direction, two light sensors 'A' and 'B' are activated at delayed times. If the 'A' light sensor activates first then the rotation is deemed as clockwise and vice versa. The accelerometer measures linear movements in units of gravity, with a zero reading for x and y and a 1g reading for z. However, a minor error of 0.01g may occur. The gyroscope estimates the angular rate for turning, with readings of 0 on x, y, and z as its initial heading. Small variations of 0.18 degrees per second for x and y and -0.24 degrees per second for z may occur, but these can be compensated in the control system. The gyroscope is more accurate than relying on encoder feedback as wheel slip or a change in terrain could result in inaccurate readings.

$$\omega = 2\pi\left(\frac{readEncoderTicks}{maxEncoderTicks * \frac{1000}{\Delta t}}\right) = \frac{\Delta\theta}{\Delta t} \quad (1)$$

$$v = \rho\omega\left(\frac{m}{s}\right) \quad (2)$$

Equation (1):
The term $2\pi * \frac{readEncoderTicks}{maxEncoderTicks}$, creates a fraction of 2π to indicate the number of radians moved. $\frac{1000}{\Delta t}$ converts milliseconds to seconds. The final units for ω are $\frac{radians}{s}$

Equation (2):
The wheels radius is ρ in meters multiplied by ω in radians per second. Since radians is a unit-less term, the final units are meters per second.

1.3 Proportional Control

The PDI control adjusts the movement of a robot toward its desired state by using mathematical calculations. The algorithm considers the robot's current and desired state to continuously make adjustments using three components: proportional control, derivative control, and integral control [1]. Proportional control sets the robot's movement based on the current position vs desired position, derivative control adjusts speed toward the desired position, and integral control adjusts based on accumulated error over time [1].

controller's inputs were the desired speed (vd), and turning rate (d), and the outputs were the left and right motor PWM command inputs uL, uR. The control development process involved computing the corresponding left and right wheel speeds given a desired speed and turning rate, followed by the design and implementation of the controller. Despite initial setbacks due to a misconfigured PMW signal, implementing the PID controller on the Arduino microcontroller was ultimately successful. A few residual bugs need to be addressed, but the program's function is satisfactory.

.

1.4 Website

A React-based landing page has been deployed to a GitHub-pages-hosted site. The site currently contains the company logo and states "Coming soon… ". Hosting may later be pivoted to the Raspberry Pi should this better accommodate

communications between ROS and the website of LiDAR and CO2 maps.

Air quality data is read from an external CO2 sensor on the rover and then transferred through a serial port from the Arduino to the Raspberry Pi. The Raspberry Pi logs the data into a file. A website, hosted by the Raspberry Pi, then opens the file and displays the data as a CO2 map of the presently selected building. The website shall be accessible by building administrators; administrators may choose to view any of the floors of the buildings they have access to. Hovering over the indicator shows a description of the room or floor's general safety. Heat maps of selected floors or areas will feature a color-coded indicator of safety (green = safe, yellow = safe for a limited time, red = unsafe). The layout of the website, including site navigation and omitting the control GUI, will adhere to the mockup shown in Figure *, created in Figma.

Figure 3: Website Mock-Up in Figma

1.5 Research

1.5.1 LiDAR Mapping

Preliminary research has been conducted regarding the methodology of using LiDAR data and Robot Operating System (ROS) to create maps of buildings. Firstly, a catkin workspace must be initialized on a machine running Ubuntu [2]. Rough data can be visualized in Rvis via scanning; Simultaneous Localization and Mapping (SLAM) is then used to translate 2D LiDAR visualizations into occupancy grids. This may be done using Hector SLAM, a mapping package for ROS which does not require odometer data [3]. Saving can be done using a package called map-server, which will save maps to YAML and PGM files that can be converted to PNG using imagemagick [3]. Alternatively, mapping can be done using the Gazebo Laser Scanner plugin and SLAM GMapping package to create an Rvis map [4]. Heat map functionality can be implemented using markers, or by overlayering a sensor data map constructed in a separate program. To optimize the accuracy of maps, the rover should be moved slowly through its surroundings to maximize data [3].

1.5.2 Remote Control

Solution 1: Bluetooth Controller [4]
A Logitech Gamepad F710 can connect to the Raspberry Pi using a USB Bluetooth dongle. Necessary libraries such as "evdev" and "pygame" will be installed to support gamepad input. Code is then written to retrieve button press data from the gamepad using the "evdev" library. Code is then tested to ensure that the gamepad is recognized by the Raspberry Pi and the button press data is being properly retrieved. The gamepad and associated code will be tested regularly.

Solution 2: Hosting a Web Server on the Arduino UNO WiFi Rev2 [5]

The WiFiNINA library may be used to create a web server that can be used to interface with the Arduino Uno WiFi Rev 2. To connect to a Wi-Fi network, the Arduino is given the network name and password. A server is created using the "WiFiServer server(portNumber)" command that listens for incoming connections on the specified port. The website interface will be lightweight to accommodate limited Arduino memory, and created using HTML, CSS, and JavaScript and is designed as a touch controller with two joysticks and an emergency stop button. One joystick will be for forward and backward motion and one for left and right motion. When the user interacts with the website controller, the "client.read" function reads the controller data and outputs it to the Arduino.

1.5.3 The Raspberry Pi

The Raspberry Pi is a versatile single-board microcontroller and a fully functional computer. It can run an operating system and various applications, making it well-suited for Ventus Robotics' rover and its automation. The Pi can run virtually any operating system; however, it is most commonly used with Ubuntu OS (A specific version of Linux) [8]. It can be programmed using several programming languages, including Python, C/C++, Java, Perl, and Ruby. Microcontroller setup requires the Pi board, a microSD card with Ubuntu OS installed, a keyboard, a mouse, and a monitor.

1.5.4 Robot Operating System (ROS)

Robot Operating System (ROS) is a collection of software frameworks that provide pre-designed rules, guidelines, common functions, tools, and libraries for building robotic applications [9]. ROS is similar to the manager of a company; it tells the different parts of the robot (employees) what nodes (jobs) to do and how to do them through packages and libraries (instructions and job descriptions). The nodes can work simultaneously on one or multiple microcontrollers allowing each node to run separately in the language of its respective microcontroller but communicate with each other using ROS. This way, the nodes can work together to achieve a larger goal. The developer can focus on writing the specific code for their robots without worrying about low-level hardware communication [9].

ROS allows the rover's motion to be controlled via the combined efforts of Raspberry Pi and the Arduino microcontrollers. The Raspberry Pi would serve as the main control center, running the ROS master node and coordinating communication between nodes to make decisions [9]. A possible sequence is shown in a systems diagram in Appendix 3. The Arduino is responsible for physically controlling the rover's motion by receiving commands from the Pi, processing them, and executing actions like controlling motor speed and direction. The two are connected via serial communication, allowing the Raspberry Pi to send commands and the Arduino to send feedback.

Ventus Robotics will use ROS 2 for speed, accuracy, security, and process communication. Ubuntu will be run on the Raspberry Pi via a virtual machine, as it provides the most support for ROS.

2 Deviations from Project Schedule

The proposed schedule for the project, Appendix 2, has undergone the following deviations.

- The original plan for external motion control was to connect a PS4 controller to the rover using an Arduino Uno USB shield and a Bluetooth dongle, however, a slight variation in the Arduino Uno Wi-Fi Rev 2 chip creates incompatibility of the USB shield library. To overcome this, alternative remote control solutions have been researched as reported.
- In addition, the team woekd on the website earlier than planned to gain a more comprehensive understanding of the user interface and allow for parallel development of the website with other features.

- Motion is more complex than anticipated, and thus has taken longer than expected; however, the mobility achieved is more responsive and robust than planned. Motion development delay also delayed LiDAR and sensor integration.

3 Provisional Plan

The following plan describes expected activity from February 6th to February 26th, at which point a Conceptual Design Report will be delivered to Q-PMC. It is expected that Ventus Robotics will continue to adhere to the timeline in presented in Appendix 2, except for discussed deviations.

The goals for the upcoming month are detailed in Table 2.

Goal	Priority	Assignee	Completion Date	Dependant Tasks
Complete PID Control	1	Luke	February 6th	Remote Control
Remote Control	1	Daniel	February 14th	UI
Raspberry Pi Setup	1	All	February 6th	ROS Setup
ROS Setup	2	Daniel	February 10th	LiDAR Sensing
Sensor Data Handling	2	Luke	February 20th	UI
LiDAR Setup and Integration	3	Sabrina	February 16th	GMapping, UI
Mappin Integration	3	Sabrina	February 19th	Heat Mapping, UI
User Interface (UI)	N/A	Luke	February 26th	
Heat Mapping	4	Sabrina	Beyond	
Underside Lighting for visibility/safety	5	Daniel	Beyond	
On-board display indicating air quality	5	Daniel	Beyond	
Stretch Goals	6	All	Beyond	

Table 2: Detailed goals for the upcoming weeks of development.

Note that the development of the UI will be iterative and dependent on the completion of other tasks; for example, maps cannot be displayed on the UI until GMapping and LiDAR is completed. Thus the priority of various aspects of the UI aligns with the priority of its dependencies. Delays in achieving the goals set out for February will result in the delay of pursuing stretch goals. The completion of all stretch goals are prioritized as follows:

1. First level autonomy: Create LiDAR map while driving rover with PS4 Controller. After the map is created, the rover may autonomously follow a predetermined path.
2. Charging dock: A charging dock is to be 3D printed and the robot can be plugged in by driving into it, accurately and automatically aligning the female and male ports.
3. Omni Wheels: Achieve a new degree of freedom by installing multi-directional wheels.
4. Stair Climbing: Install a mechanism allowing the robot to climb stairs without intervention.
5. Second level Autonomy: Create a LiDAR map and simultaneously navigate a building autonomously.

4 Evaluation

Ventus Robotics has made substantial progress on the development of both the software and hardware of the rover as well as greatly increasing the database of knowledge and sense of technical direction through research. Moving forward, the focus will be to improve and refine the design, emphasizing the implementation of various air quality sensors and LiDAR mapping.

References

[1] P. Explained. "PID Controller Explained." Pi Explained, pi-explained.com/pid-controller-explained/. Accessed 10 Feb. 2023.

[2] A. Kumar, “How to use a lidar sensor with Robot Operating System (ROS): ROS,” Maker Pro, 06-Feb-2023. [Online]. Available: https://maker.pro/ros/tutorial/how-to-use-a-lidar-sensor-with-robot-operating-system-ros. [Accessed: 05-Feb-2023].

[3] A. Addison, “How to build an indoor map using ROS and lidar-based Slam,” Automatic Addison, 22-May-2021. [Online]. Available: https://automaticaddison.com/how-to-build-an-indoor-map-using-ros-and-lidar-based-slam/. [Accessed: 05-Feb-2023].

[4] K. Palla, “Creating a map using laser scanner and Gmapping,” KiranPalla.com. [Online]. Available: https://kiranpalla.com/autonom navigation-ros-differential-drive-robot-simulation/creating-map-using-laser-scanner-and-gmapping/. [Accessed: 05-Feb-2023].

[5] “Raspberry Pi and Gamepad Programming Part 1: Reading the device,” Eric Goebelbecker, 12-Dec-2020. [Online]. Available: https://ericgoebelbecker.com/post/raspberry-pi-and-gamepad-programming-part-1-reading-the-device/. [Accessed: 05-Feb-2023].

[6] T. A. Team, “Host a web server on the Arduino Uno WIFI REV2: Arduino documentation,” Arduino Documentation | Arduino Documentation. [Online]. Available: https://docs.arduino.cc/tutorials/uno-wifi-rev2/uno-wifi-r2-hosting-a-webserver/. [Accessed: 05-Feb-2023].

[7] “L298 dual H-Bridge Motor Driver,” Seeed Studio, 11-Dec-2017. [Online]. Available: https://www.seeedstudio.com/L298-Dual-H-Bridge-Motor-Driver-p-284.html. [Accessed: 05-Feb-2023].

[8] PCMag. (2021, June 7). Beginners Guide: How to Get Started with Raspberry Pi. Retrieved February 5, 2023, from https://www.pcmag.com/how-to/beginners-guide-how-to-get-started-with-raspberry-pi.

[9] “Basics of ROS: Ros Robotics,” Home. [Online]. Available: https://www.rosroboticslearning.com/basics-of-ros. [Accessed: 05-Feb-2023].

Appendix 1

Figure 4 displays the motor driver and its pins, the connections of which are described in Table 3.

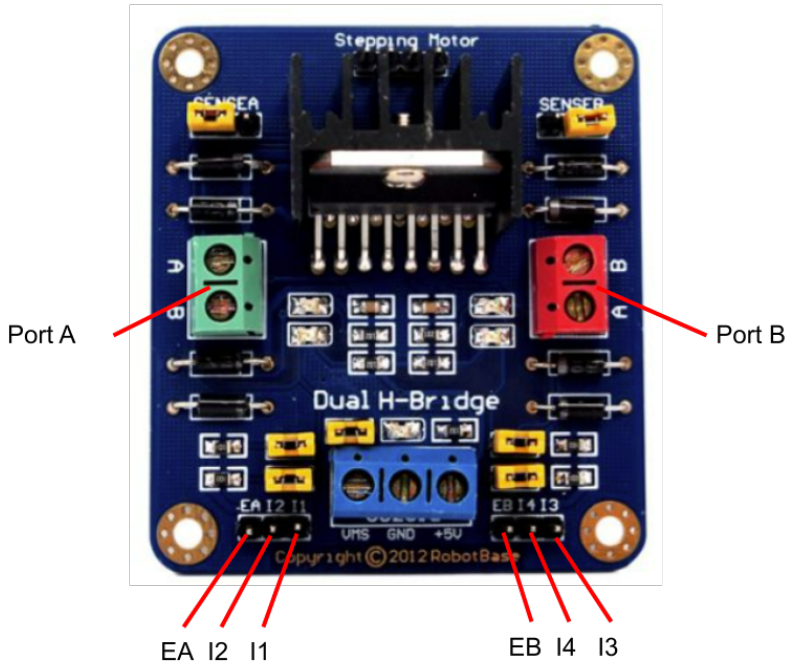


Figure 4: A diagram of the pinout for the motor driver [6].

Dual Driver Pin	**Arduino Pin**	**Colour**
EA	6	Green
EB	11	Green
I1	D3	Yellow
I2	D5	Blue
I3	D4	Yellow
I4	D2	Blue

Table 3: Pin out for connection of Arduino to the motor controller board.

Appendix 2

The Gantt Chart illustrated in Figure 5 demonstrates the expected timeline for the development for the mobile robot on behalf of Q-PMC. Deviations are with respect to this timeline.

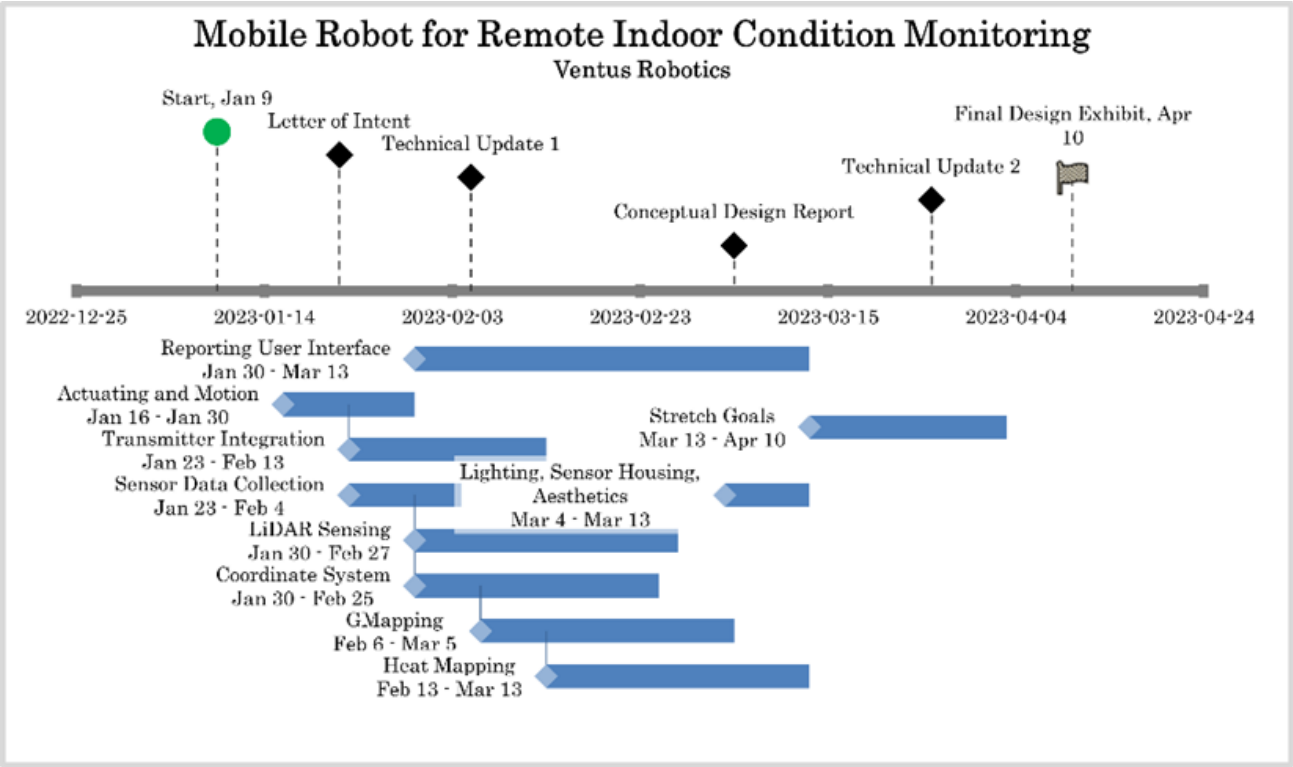


Figure 5: Expected Timeline for Mobile Robot Development - Ventus Robotics

Appendix 3

The systems diagram presented in 6 proposes a potential "workflow" for the robot, as managed by ROS.

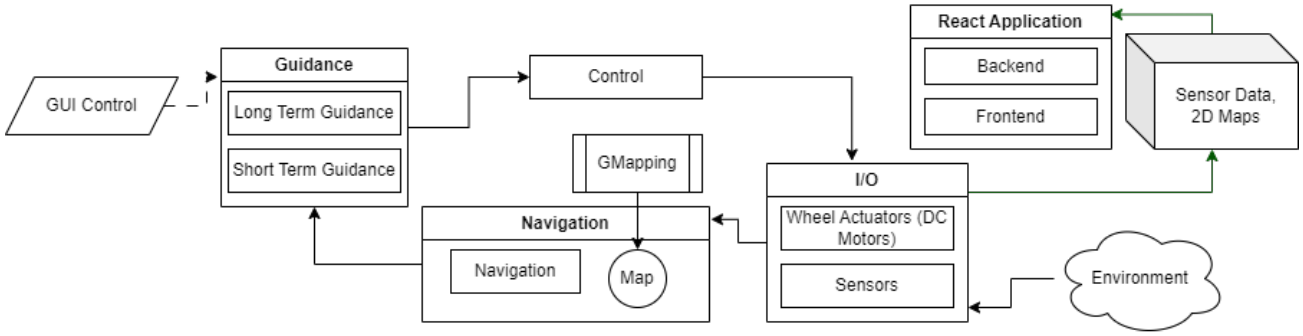


Figure 6: A systems diagram demonstrating a plan for how the robot's systems will work cohesively.