1

Remote Soil Monitoring Robot

Chalida Pisuraj, Kyla Quilos,, Judith Rupasinghe, Carol Varkey
Tony Givargis Ph.D, Professor; Vice Chair of Undergraduate Studies, Computer Science, Advisor
Department of Electrical Engineering and Computer Science

***Abstract*— With the Remote Soil Monitoring Robot (RSMR), the future of farming will be effortlessly water efficient. We aim to create a robot to better determine hot spots for inefficient water usage on farmland and gather meaningful data for water conservation. The robot will do so by traversing the farmland and probing the soil for moisture and humidity levels and communicating this data to a mobile application through Bluetooth. With the Winter Design Review, our robot has been able to complete the proposed objectives listed above. The RSMR, however, was not able to traverse autonomously, but required manual input on the Bluetooth terminal to maneuver.**

***Index Terms*—Remote Soil Monitoring Robot (RSMR), Soil Moisture Content, Pulse Width Modulation (PWM)**

I. Introduction

AGRICULTURE, according to the Food and Agriculture Organization of the United Nations (FAO) , accounts for 70 percent of the global freshwater withdrawals and most of it is wasteful use [2]. The increase in population and food production over the last few decades has caused detrimental water usage. Good soil management and structure leads to water conservation and soil efficiency. Our robot will traverse farmland based on user inputted dimensions on our Android mobile application. The bot will then probe and gather sensor data for moisture and humidity content of soil to provide a choropleth map and suggest where water can be conserved.

1

We aim to promote sustainable environmental development by creating a faster way to collect and gather meaningful data for water conservation. By using data driven technology, tools like this will be able to assist farmers to optimize their yield, minimize input costs, and reduce environmental impact on crop growth [1].

Prior approaches to achieve a tool that can provide moisture and humidity data was to simply create a stationary probe that required manual relocation. This method proved to be time consuming and lacked efficiency and consistency because it required the farmer to traverse a farmland with sizes that can differ greatly. The current method we decided upon is to have a robot to do the traversing with accurate increments and storage of data. The probing mechanism required different approaches as well. We originally approached this mechanism with a drop-down arm mechanism to pierce the soil with the probe itself. This motion proved to be harmful to the probe because it was a rougher, inconsistent, and more sudden way to pierce soil that could be denser. The varying consistencies of soil can lead to hardware damage. To prevent breaking the probe, we decided to approach this issue by implementing a less invasive and more careful insertion method that did not rely on gravity as the driving force by placing it underneath the robot and using an actuator mechanism that rely on the torque of the wheels.

II. High-Level Hardware and Software

In the design of the RSMR our goal as a group was to integrate various hardware components in order to construct a bot that can effectively gather the soil data over our test area. In going through the engineering design process, we refined the initial details of our project to align with what could be economically and socially viable. During Fall Quarter, our focus was on learning how these hardware components work individually and mapping how they can be added to our bot. This quarter, we mainly worked on finalizing our hardware and app design in which we were determined to finish to start testing with soil.

The current code is for the motors, sensors, and Bluetooth module. The motors dictate the direction, while the sensors collect data about the soil; while the bot is in the field, these two will be used the most, as shown by the field flowchart. Once the bot is back at the base, it will connect to the user's phone and send the data collected by the sensor.

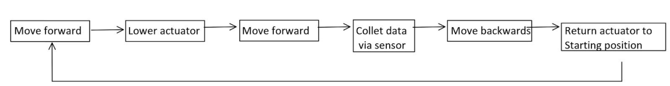

Fig. 1. Flowchart for robot in the field

A. RC Car Assembly

Throughout this quarter, we have continuously worked on refining the hardware aspect of the bot based on the purpose that it served for the project. During our first phase of development, we opted to build the bot ourselves rather than buying a prebuilt RC car that we could attach the various hardware components to; mainly for the interest of having free range in how our bot can be built with the flexibility of placement and wiring of the other parts. The CAD model below shows the design of the bot.

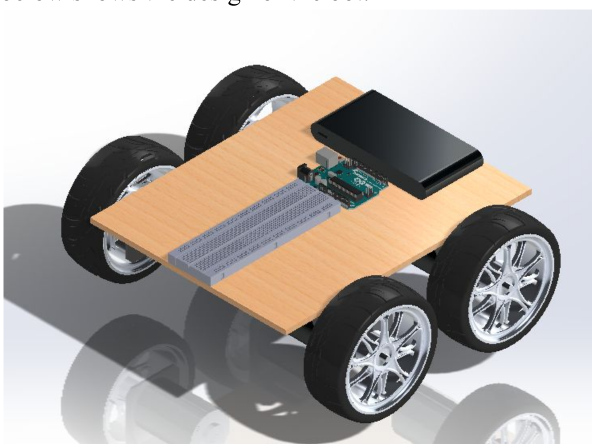

Fig. 2. Robot CAD

Materials included for assembling the bot and other parts of the bot for senor integration include: Arduino Uno, Duratrax Performance Racing Tires, 4 200 RPM DC Motors, Motor Driver Controller, Breadboard, Jumper Wires, Battery Holder Case and 12V Batteries, Wood Panels, and Brackets. We first started our assembly by using PWM (pulse width modulation) to lower/increase speed as desired for when we probe and traverse, described later in the report. From there we worked on attaching the wheels to the motor shaft, and the motor to the wood panels with additional drilling to the tires and bracket holders. As per our reasoning for creating our own RC car, the wiring appeared much more simplistic in connecting the motors, through the breadboard/Arduino Uno to power. The diagram below shows a schematic of the wiring for the motors with the Arduino Uno, 4 200 RPM DC Motors, Motor Driver Controller, Moisture sensor, and HC-05 Bluetooth Module.

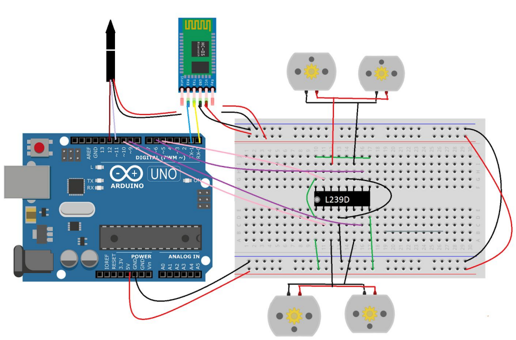

Fig. 3. Robot Schematic

B. Soil Sensors

In regard to the sensors themselves, we had initial background research on what sensors would effectively provide information on water levels of a given soil sample, on the basis that its cost effective for probe construction. We refined from using a multi-sensor approach to strictly focusing on a moisture and temperature sensor. These sensors can provide the most translative data on water levels. Below shows the main sensors for our bot including the moisture and temperature sensor with additional materials for implementation provided through the Arduino Rural Hack Kit.

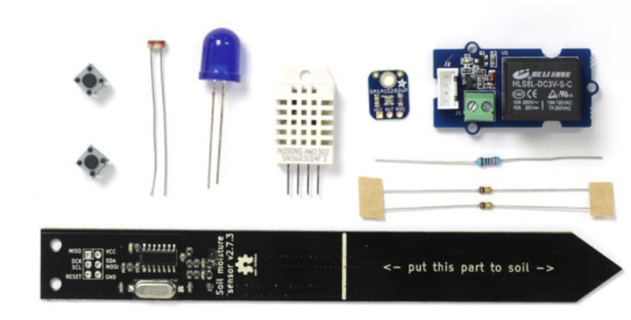

Fig. 4. Sensor Supplies

C. Bluetooth Module

The Bluetooth module was added as an addition to the RSMR as a way to facilitate communication between the data collected from the sensors and a mobile application. In adding the HC-05 Bluetooth module to the Arduino board, the wiring follows Figure 3.

III. Methods

A. Motors

The motors are connected to a motor driver, which determines what direction the bot will move-- forward, backward, left, or right. As seen in Figure 5, the motors are connected in pairs to the driver; the positive terminals of two motors are connected to one motor driver output, while their negative terminals are connected to another motor driver output. The Bluetooth module helps facilitate communication between the motor driver and Android mobile application to navigate the direction of the bot. The respective inputs are connected to the Arduino; when the input gets a high signal, then the output will turn “on”, and the motor will turn.

Our directional motion is implemented using pulse width modulation (PWM) via Arduino’s analogWrite function. PWM is a technique that changes the pulse width of a signal, resulting in a lower current; in our case, this results in a lower RPM. Using a lower speed will reduce the potential for damage while we move during probe insertion because there will be less force on the sensor as it enters the soil.

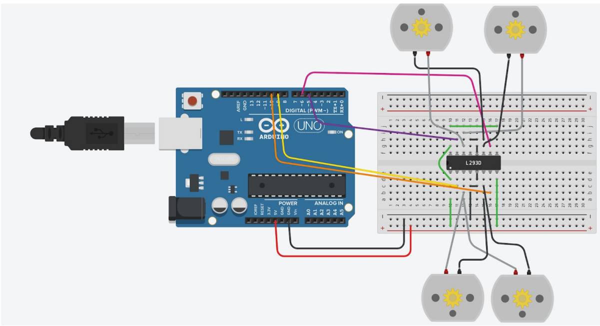

Fig. 5. Sensor Supplies

B. Bluetooth Module

We're using an HC-05 Bluetooth module to communicate our sensor data from our bot to our phones, and navigate our bot. Using a Bluetooth module uses less power, and can send and retrieve information in a short range; since we only needed data to be transferred while the bot is back base, it made more sense to choose Bluetooth over Wifi.

In our code, we first open the serial port and set the data rate to 9600, in accordance with the HC-05 module. Our code loops with the following: if there are any bytes available for reading for the serial port, then proceed to read and print to the phone. One issue we ran into was that we were struggling to upload our code while our Bluetooth module was fully wired. We later learned that our TX/RX pins (the digital pins 1 and 0) needed to be disconnected during the uploading process, because it was also sending an additional voltage to the device. In our current stage, we can send data that can turn an LED on/off from an Android , and print out a message on a Bluetooth terminal on our phone. Future modifications to the code will include receiving sensor readings in an easy to process format, such as a JSON string.

C. Sensors

The sensor used can measure soil moisture capacitance, temperature, and light levels. Our code is based on the I2CSoilMoistureSensor library provided by Miceuz, the manufacturer of the sensor being used. Light readings can be interpreted in the range of 0 to 65565, where 0 is the lightest, and 65565 is the darkest; it is also worth noting that it takes more time to get a value in lower light environments. While we didn't use this reading directly in our project, the light readings were used in testing to show that our sensor was wired correctly and working. The temperature measurement reports data in Celsius; this number may be converted in future versions of our project to line up with the imperial system. The moisture capacitance measurements will give a higher reading in soil with a higher moisture content. Our base value is expected to be around 200-300 when the sensor is not in the soil [4].

This quarter in conducting uncontrolled and controlled testing of different soil types we were able to get more accurate readings, based on improvements within the mechanism of the bot itself. This data is then collected and sent to the Bluetooth module for display on the Android mobile application through the choropleth map.

D. Actuator

In designing the probing motion of the bot, there were various factors to take into account such as how forceful the motion of the actuator probing should be based on the fragility of the sensors. In the beginning phases of our design process we relied on the sensors being probed into the soil based on a drop-down arm driven by the force of gravity. Through further testing, the sensors are at risk for possible long-term damages. Therefore, an actuator was a necessary hardware component to aid in a more calculated and less invasive insertion method. The actuator facilitates in creating "accurate, repeatable performance[s] of pushing/pulling" [3]. The prototype of our actuator is based on the motion of a dump truck. Last quarter we made a prototype of this model with the use of a mini Servo. We constructed an initial prototype to test the design that is placed underneath the robot. This quarter we finalized our bot with a reconstruction of the materials used and a more precise construction process of the actuator. Through observations made in our initial design we were able to transfer the recommended changes to accommodate the actuator to the purpose of our bot. The mechanism mimics that of the dump truck by replacing the dump body with the sensor and recreating the hoisting motion to lower it closer to the soil. This angle will allow us to be at an optimal position to insert without risk of damaging the sensor. The wheels will then move forward to allow steady force to pierce the soil.

E. Code Implementation

Our android compatible app is capable of communicating to our Arduino Uno to receive sensor data, and map the moisture concentrations of the field. It is composed of three main parts: the home page, the Bluetooth terminal, and the map page.

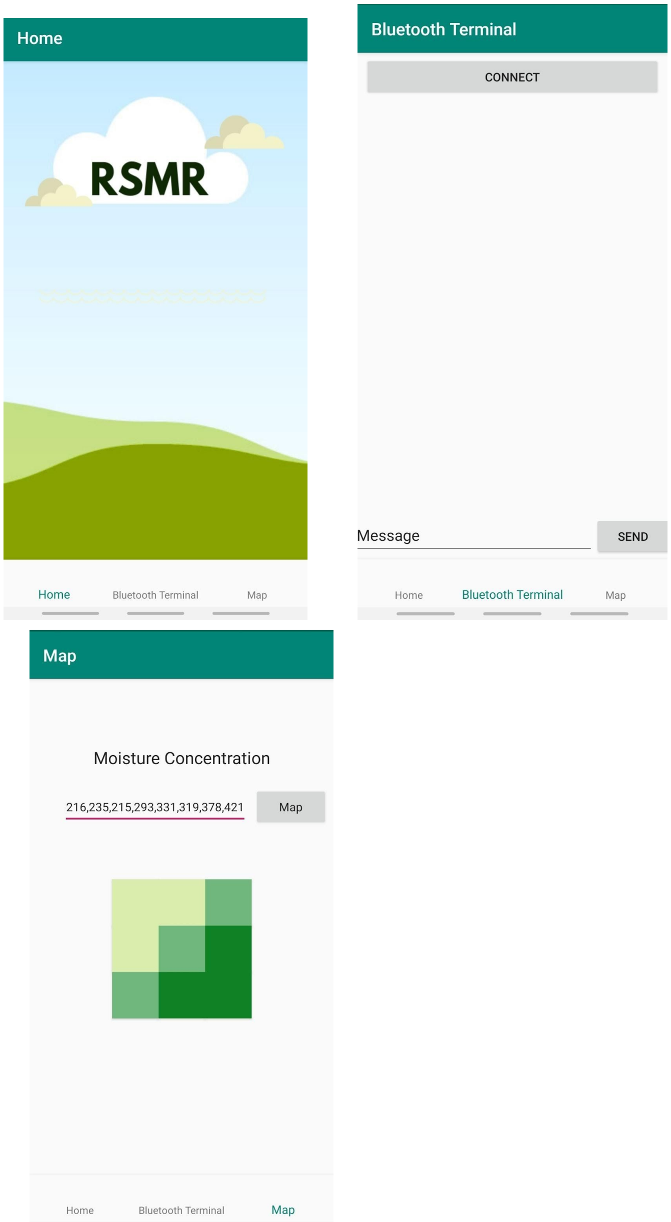

Fig. 6a. Home page; Fig. 6b. Bluetooth Terminal; Fig. 6c. Map

The home page, as seen in Figure 6a, serves as a welcome page.

As seen in Figure 6b, the Bluetooth terminal uses an Android's local Bluetooth adapter to check if Bluetooth is on, before querying for paired Bluetooth devices and scanning for other unknown devices. It then establishes RFCOMM channels and connects to the HC-05 module on the bot. Once the connection has been secure, the app can transfer data between the phone and the bot; the phone sends a "connect" message, and the bot will respond by sending the moisture concentration data. The user then copies and pastes these numbers onto the map page.

The map page takes nine or less data points and converts them to colored sections on the choropleth map. This works by assigning ranges of numbers to different shades of green, with a light green being the driest and the dark green being the most moist. An example of this can be seen in Figure 6c. In the case that less than nine values are given, the map will color those sections grey. Alternatively, if more than nine values are provided, only the first nine values will be displayed.

Future changes to the app will build on the current implementation. The home page would lead to a sign in page, where logged in users could view past choropleth maps, and average data over months. The Bluetooth terminal would be modified to allow users to input coordinates to allow for custom plots of land to be probed, along with the frequency of probing.

IV. Results and Performance

A. Motors

This quarter, we were able to get the motors to move our bot using PWM and analogWrite through communication on the Android mobile application. A major concern that was addressed this quarter in experimentation was the weight of the bot, and ability to traverse over different surface types. Once this issue was addressed through a change of motors, and reconstruction of the bot we continued experimentation through stages. First in uncontrolled testing on hardwood and how the bot was able to carry the weight, then later with more components included.

In determining the possible range of motion that the bot is able to traverse, we have the bot wired to send high power to both terminals, hence it's possible to move forward and backwards. To stop, we simply set analogWrite to 0, which results in a pulse width of 0; the short pulse width results in a low current being sent to the motor, resulting in no movement.

B. Sensor

In working with this kit the wiring follows the Figure 2 schematic above. In purchasing these sensors we had to take into account the testing conditions that the sensors had to work with, and these sensors seemed to be the most viable option with the bot we are constructing. This was especially important to take into account in positioning with the actuator, and how we can most accurately have the moisture sensor working.

Moisture capacitance has a direct relationship with the sensors resulting numbers. As a result, we used 214-216 as a baseline to represent the sensor in air. Once the sensor enters the soil, the soil capacitance number changes to represent the moisture level, seen in Figure 7. The average between the baselines are calculated and saved. This average is later sent to the application using the Bluetooth module.

```
21:53:16.500 -> Soil Moisture Capacitance: 215,
21:53:19.608 -> Soil Moisture Capacitance: 255,
21:53:22.692 -> Soil Moisture Capacitance: 261,
21:53:25.786 -> Soil Moisture Capacitance: 262,
21:53:28.861 -> Soil Moisture Capacitance: 265,
21:53:31.953 -> Soil Moisture Capacitance: 268,
21:53:35.047 -> Soil Moisture Capacitance: 261,
21:53:38.123 -> Soil Moisture Capacitance: 260,
21:53:41.226 -> Soil Moisture Capacitance: 257,
21:53:44.309 -> Soil Moisture Capacitance: 215,
```

Fig. 7. Sample sensor readings.

C. Bluetooth Module

We scaled our HC-05 Bluetooth module tests from being able to communicate information between a mobile application and an Arduino with an LED, to being able to communicate the sensor data with a mobile application. The Bluetooth module, in this case, acted as an intermediary between the app and the sensor. The application would use the phone's Bluetooth adapter to scan and pair with the Bluetooth module on the bot. Then, the application would send a message ("connect") to the Bluetooth module, which signals the Arduino to send the sensor data. Once the application receives the data, the user can then copy the data and paste it into the map section of the application..

V. Materials Used

In the beginning phase of our project, we allocated time to choosing materials that more effectively work to the purpose of our project. These materials abide by the constraints of the project itself and altered based on challenges that we faced throughout the last two quarters. Below shows the materials used as far in the construction of the bot, Bluetooth module, soil sensors, and actuator.

VI. Conclusion

The RSMR has been built to traverse firm, soft soil. Given this parameter, wheel speed and torque were adjusted accordingly by using pulse width modulation to optimize traversal time and controlled force to assist the probing mechanism. Our robot can be separated into three different modules: wheels/motors, actuator, and Bluetooth. Each of these modules were tested for Arduino integration and successful results were obtained. Our motors were able to run at the speed and increment specified. Our actuator can move the probe and perform basic mechanisms and receive sensor data successfully. The Bluetooth module provided satisfactory results as a separate implementation and has been integrated to receive and send data to the RSMR prompting movement and the transaction of data.

The system implemented uses an Arduino Uno with the 200 RPM DC motors, HC-05 Bluetooth module, and I2C Soil Moisture Sensor. Hardware designs for traversal and sensor designs for moisture and humidity have been successfully implemented. Implementation of Bluetooth for mobile application communication and data storage has been completed. Overall, a working minimal viable product of the RSMR has been created for the Winter Design Review.

Acknowledgment

We would like to thank Professor Tony Givargis for his continued guidance and support during our project. And we would like to thank Kayla San Pedro for providing us with her time and skills to modify our brackets. A final thank you to the Undergraduate Research Opportunities Program (UROP) for funding our project.

References

[1] W. Isaac, S. Varshney, and E. Khan, "An IoT based system for remote monitoring of soil characteristics - Semantic Scholar," *An IoT based system for remote monitoring of soil characteristics*, 01-Jan-1970. [Online]. Available: https://www.semanticscholar.org/paper/An-IoT-based-system-for-remote-monitoring-of-soil-Na-Isaac/be0eced902284d0e56a714fab8a64cf35b8ef1be. [Accessed: 02-Nov-2019].10.1109/INCITE.2016.7857638.

[2] Food and Agriculture Organization of the United Nations. "Water for Sustainable Food and Agriculture." *Water for Sustainable Food and Agriculture*, 2017, http://www.fao.org/3/a-i7959e.pdf. Accessed 2 Nov. 2019.

[3] Apollo77, "I2CSoilMoistureSensor," *I2CSoilMoistureSensor,* 2019, "https://github.com/Apollon77/I2CSoilMoistureSensor"

[4] "Thomson Linear Motion Optimized," *Actuators*. [Online]. Available: https://www.thomsonlinear.com/en/products/actuators. [Accessed: 11-Dec-2019].