Product: FarmFriend
Team: The FarmFriends

Abstract

FarmFriend empowers farmers by digitizing farms, giving them precise and actionable soil health data in real-time for the entirety of the field. In this demonstration, we worked towards the sensor functionality which will be used to return all the data needed to digitize farms. We successfully implemented a mechanism to lower and raise the sensors, in order to allow them to take soil readings to be shown on an LCD display. In addition, we began implementing a navigation system by programming simple movement on the prowler chassis. Furthermore, we have managed to combine both of these elements by building a LEGO structure mounted onto the prowler chassis, giving us a robot prototype that is able to be expanded easily for further demonstrations.

1. Project management update

Originally, we intended to use ROS to manage the control of the motors. However, the Raspberry Pi given to us was not sufficient, as it did not have ROS installed and it was blocked off from the Internet. After consulting with the technicians, they agreed to take apart one of the Turtlebots, and give us the Raspberry Pi from it (those Pi's have ROS preinstalled). This delayed our progress by 2 days. On Thursday of Week 3 (the 3rd of February), we held a meeting with Thomas Corbères, an expert on robotics, where we were given advice on using ROS. He suggested that, should we use ROS, we would use one of the Navigation packages that are provided. However, this comes with the downside that using ROS may not be the most efficient option, given our relatively small number of motors that we would have controlled. Therefore, we decided not to use ROS for this demonstration, although this may change as circumstances arise in the future.

We were split into teams, each responsible for a specific subsystem. For this demo, the two main teams were Sensors and Navigation. The sensor team began the week with designing and prototyping the sensor lowering functionality, starting with experiments using the MeArm, but settling with a gear-racked system of mostly our design. To ensure task progress, we utilized ClickUp to assign tasks. This allowed us in a team to work towards achievable goals in an organized fashion. Furthermore, the in-person nature of our work sped development as we could constantly trial ideas, discuss functionalities and show progress to each other. Consequently, this also helped us find ways to interface the work between teams more easily than if we had done it asynchronously.

Specifically, even though this demonstration implies the work of two separate teams, the nature of the goal (designing the robot prototype) naturally entailed close cooperation between the two. This means that we were flexible with assignment of tasks to allow members to work where they could provide the most benefit at the time:
Project Manager:

- Sury created the control script that SSH's into the Raspberry Pi, and allows the Pi to be controlled remotely through the DICE machine. He also modified the Arduino code to allow for communication with the Raspberry Pi via a serial interface, helped create the presentation and performed market research for the value proposition section of the presentation. Sury, alongside Xingchen, gave the presentation for the first demonstration. Finally, Sury helped collate the data required for the quantitative analysis section of the report, as well as performed the analysis on the data, which helped gather key information for the navigation team.

Moreover, from the Navigation team:

- Alin helped with setting up ROS, looking into which packages are used by the Turtlebot (as those packages include ones that interface with motors and sensors, which we undoubtedly would have needed), and helped the Sensor team with building the robot prototype.
- Niamh helped with connecting the Raspberry Pi to the prowler's motors, and wrote the initial scripts for moving the motors together to move the robot; this was after getting a suitable Pi from an existing Turtlebot, in order to make use of installed packages.
- Niveda helped set up the Raspberry Pi for navigating the robot and researched what packages of ROS our robot would require for the necessary autonomous navigation tasks. Also wrote code for the fake GPS system using OpenCV that would help with the navigation of the robot, although this is not content that is presented in the first demonstration.
- Gabriel created a mock design for the web application.

Finally, from the Sensors team:

- Andrew began with prototyping the Arduino script for the motor to move the gear and helped make a basic sequence of the lowering, pausing and lifting of the rack. Helped to build the LEGO structure and mount it on the prowler.
- Xingchen helped create the presentation, designed the mechanical sub-components of the robotic system, drew the stl graph for 3D printing a new rack with Tinkerpad, drew the dxf file for the box with Makercase, designed and built the LEGO structure for the

gear-rack system, designed the structure for moisture sensor and LCD display, helped with the motor for the gear, designed the Arduino code, helped with the structure of the robot body.

- Charles helped with the idea of the gear-rack design of the robot, and helped to build this system by using LEGO. Furthermore, helped to write Arduino code for lowering and lifting of the gear-rack system, the setup of moisture sensor and the Arduino code for displaying the moisture level on screen.

Figure 4 shows the distribution of the workload undertaken be each team member.

2. Quantitative analysis and testing

Currently, we have developed code that is responsible for controlling the moisture sensor and moving the prowler motors. We have created a script that remotely tells the Raspberry Pi to start the process of moving from a certain point for a specified amount of time, then lowering the sensors and showing the moisture reading on the LCD display. This is our main script that we used for testing these components.

We identified that one of the greatest risks that our project faces is the navigation system, which, due to its complex autonomous nature, poses key challenges in terms of reliability. Furthermore, we have replaced the chassis of the Turtlebot, which provided reliable and accurate movement due to its pre-built movement modules, which modelled its motion very precisely. Since we used the Prowler base system for our project; although we can control the amount of power we supply to each of the motors of the Prowler wheels, we were concerned that we have no concrete measurement of the distance travelled by the robot.

A simple test that we derived was to measure the distance output (in cm) of the robot with varying levels of power. The aim of this data collection is to understand whether we can reliably estimate the location of our system based on the output that we provide the wheels with.

MEASURED DISTANCE TRAVELLED (CM) WITH DIFFERENT MOTOR SPEEDS

	MOTOR SPEED				
ATTEMPT	30	35	40	45	50
1	0.2	81	111	135	154
2	0.3	80	116	131	155
3	0.1	79	113	133	157
4	0.6	83	115	137	152
5	0.2	83	112	132	159

Table 1. The measured distance travelled (cm) by the robot at different motor speeds, with 5 repetitions for each motor speed.

Analysis of the data from the table above has been performed in Figure 1 and Figure 2.

Figure 1 shows how the distance travelled varies with different motor speeds, as well as showing the error bars showing the uncertainty between the 5 measurements taken for each motor speed. The graph confirms our suspicion of the correlation between the motor speed and the distance travelled deviating from being linear. It is clear from the graph that the distance travelled increases at a decreasing rate as we increase the motor speed. This makes it very difficult for us to model and predict the distance travelled if the robot is going at different motor speeds at points in its navigation path.

It should be noted here that there are many assumptions being made with the data that we have collected also. One of the most significant assumptions we are making is of the resistive forces in action being constant. For example, the data that we have collected is for one surface type only, and a different surface, such as soil, will not give the same values, due to more frictional forces, and uneven terrain, causing more gravitational forces to overcome. In an ideal scenario with flat, even terrain, like our test data, we would be able to drive the motor at a constant speed throughout for different durations. However, in practice, this is not feasible, as to cover the same distance, sometimes we would need to drive the motor at higher speeds to overcome a slope for example.

From this data, we were also able to figure out that a motor speed of 30 was the minimum required speed for our robot to be able to move.

Figure 2 shows how the standard deviation of the distances measured change with varying levels of the motor speeds. From the chart, it is clear that there is a strong positive correlation between the uncertainty of the distance travelled, and the speed of the motor. In practise, this would mean that the higher motor speeds means that we would be less certain of the exact distance travelled (and thus, the location) of our robot. This suggests that, to get more accurate measurements of the distance travelled (and thus, location) of our robot, we would use lower motor speeds for longer durations.

In retrospect, we would have also tested how the accuracy of our distance measurements changes with longer durations of the motor running at constant speeds. However, we believe that the data we gathered is enough information for us to decide on having to use additional sensors for estimating the position of the robot more reliably.

Given the analysis above, we believe that it is very important for us to have telemetry on the trajectory and motion of our robot. For the next demo, the navigation team are working towards finding the best sensors to use for this; currently, we are considering between an inertial measurement unit and wheel encoders. This would be alongside our plan of using the camera feed from the demo room to mimic the use of a GPS navigation system.

Finally, We also exhaustively tested the full demonstration sequence to ensure its repeatability in the presentation. The tests involved placing the demo soil a set distance in front of the robot; the robot would then move so that the sensor is in position above the soil. The sensor would lower, allowing a reading to be taken, and then retract back up. This test shows the full functionality for the first demonstration, which represents an important milestone as we have completed the previously set goals of our project plan.

3. Budget

The prices of the items we used that were included in our kit, and that we borrowed, are listed in Table 2:

Item	**Quantity**	**Price (£)**
Arduino Uno	1	19
Prowler Robot	1	≈ 170
Zippy 30C Series 5000	1	≈ 30
Arduino Base Shield	1	3.50
NXT Lego Motor	1	32.39
Raspberry Pi 3	1	30
PowerBank	1	15
Raspberry Pi Camera	1	20
Power Board (Arduino)	1	10
Motor Board	2	20
AA Rechargeable Batteries	8	9.60
Grove LCD RGB Backlight	1	13.10
Grove Capacitive Moisture Sensor	1	6.50
LEGO	-	5
TB3 Waffle Plate	2	25.76
Miscellaneous	-	5
Total		**414.85**

Table 2. The prices of items currently used.

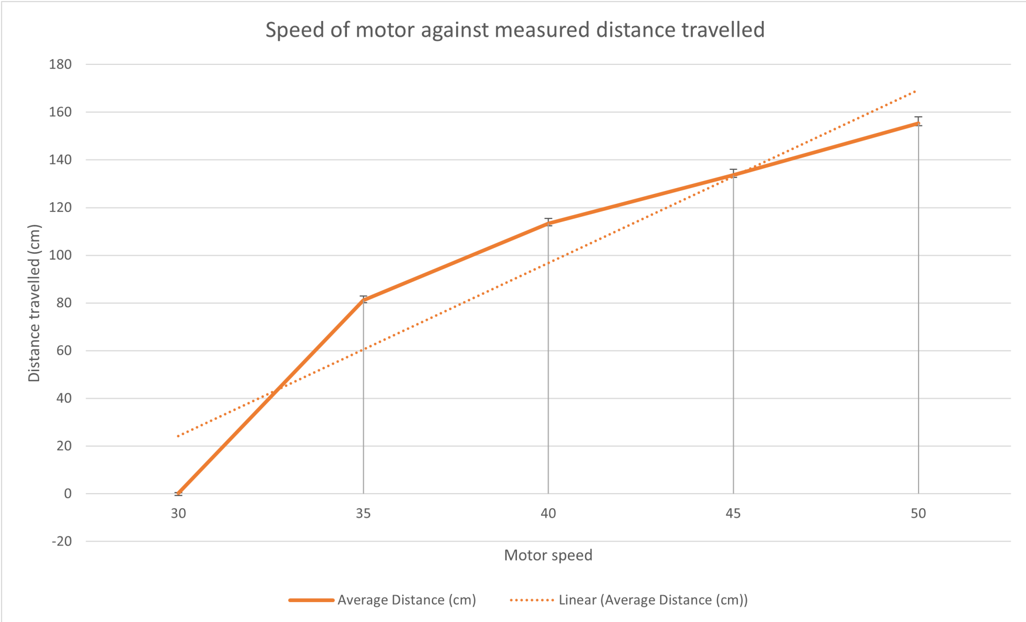

Figure 1. The distance travelled (cm) by the motor with varying levels of motor speeds for 3 seconds.

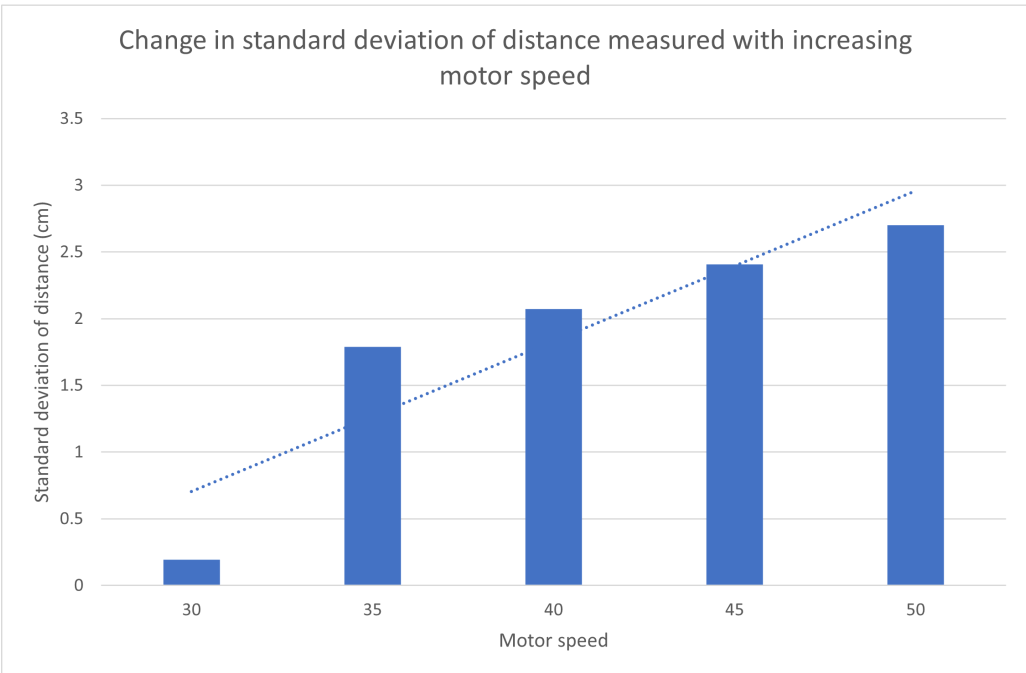

Figure 2. The standard deviation of the distance measured when varying levels of the motor speeds for 3 seconds.

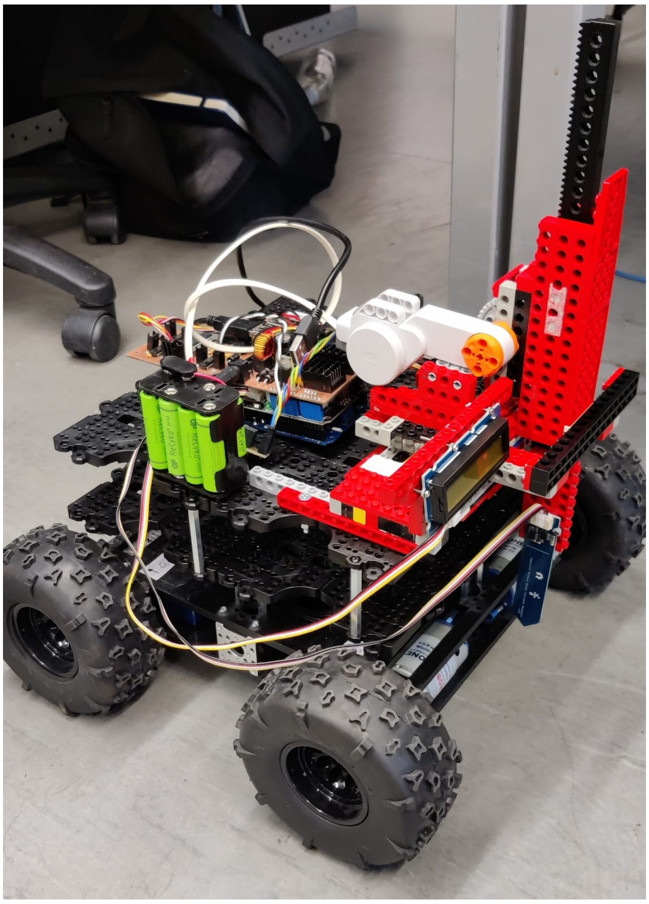

Figure 3. The robot prototype.

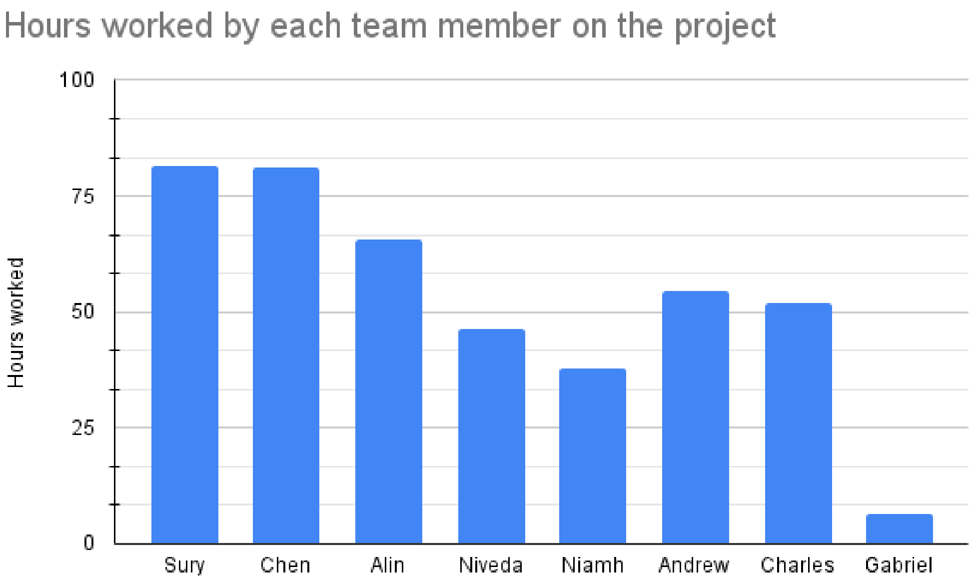

Figure 4. The hours worked on the project (including project meetings attended) by person.