Internship 2022

Progress report

Name: Allan Were

Tasks completed last week

- [#25] obstacle avoidance using ultrasonic sensor

The robot was assembled and the program uploaded. The robot car was able to avoid obstacles as per the programs instruction.

The robot was able to move while avoiding objects using the ultrasonic technology

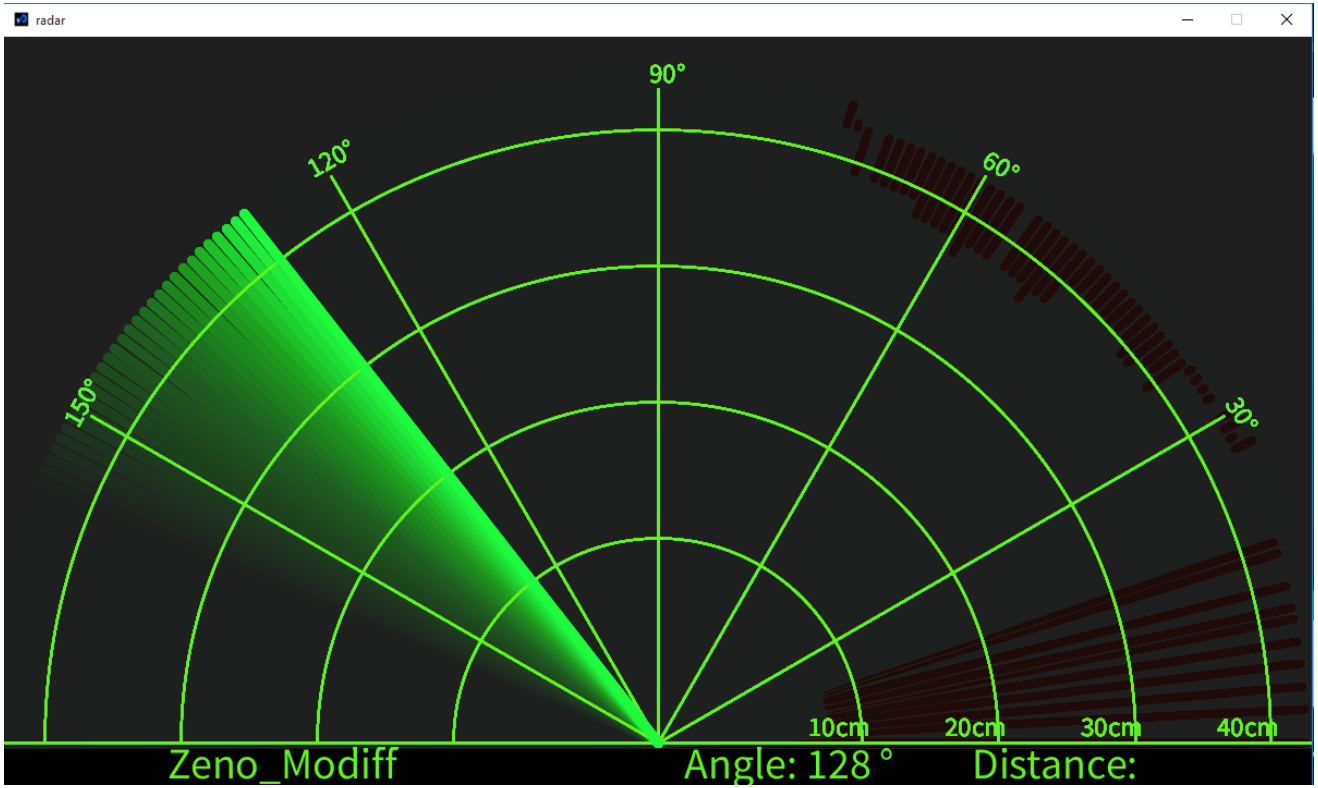

Figure 1 A radar output from processing IDE

- [#35] GPS testing

The GPS module NEO-6M GPS was tested and results were obtained.

Figure 2 GPS module

```
// Create a TinyGPS++ object
TinyGPSPlus gps;

// Create a software serial port called
SoftwareSerial gpsSerial(RXPin, TXPin);

void setup()
{
  // Start the Arduino hardware serial p
  Serial.begin(9600);

  // Start the software serial port at t
  gpsSerial.begin(GPSBaud);
}

void loop()
{
  // This sketch displays information ev
  while (gpsSerial.available() > 0)
    if (gps.encode(gpsSerial.read()))
      displayInfo();

  // If 5000 milliseconds pass and there
  // over the software serial port, show a "No GPS detected" error
  if (millis() > 5000 && gps.charsProcessed() < 10)
  {
```

COM3

```
Latitude: -1.095618
Longitude: 37.012531
Altitude: 0.00
Date: 2/3/2022
Time: 09:05:11.00

Latitude: -1.095618
Longitude: 37.012531
Altitude: 0.00
Date: 2/3/2022
Time: 09:05:11.00

Latitude: -1.095618
```

Autoscroll Show timestamp Newline 9600 baud Clear ou

Figure 3 GPS input

- [#37] Haversine formula for distance calculation between two points

The **Haversine** formula calculates the shortest distance between two points on a sphere using their latitudes and longitudes measured along the surface. It is important for use in navigation.

The haversine can be expressed in trigonometric function as:

$$haversine(\theta) = sin^2\frac{\theta}{2}$$

The haversine of the central angle (which is d/r) is calculated by the following formula:

$$\left(\frac{d}{r}\right) = haversine(\Phi_2 - \Phi_1) + cos(\Phi_1)cos(\Phi_2)haversine(\lambda_2 - \lambda_1)$$

where r is the radius of the earth(6371 km), d is the distance between two

points, ϕ_1, ϕ_2 is the latitude of the two points, and λ_1, λ_2 is the longitude of the two points respectively.
Solving d by applying the inverse haversine or by using the inverse sine function, we get:

$$d = rhav^{-1}(h) = 2rsin^{-1}(\sqrt{h})$$

or

$$d = 2rsin^{-1}\left(\sqrt{sin^2\left(\frac{\Phi_2-\Phi_1}{2}\right) + cos(\Phi_1)cos(\Phi_2)sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}\right)$$

- [#38] Compass(HMC8883L) for the orientation and waypoint

The compass module was researched on and it will be used to give the robot the orientation.

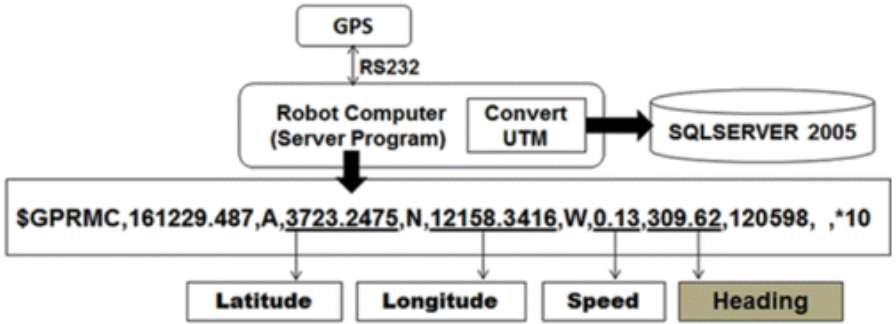

Figure 4 GPS Expectations

The first stage of the autonomous guidance algorithm determined the angle difference by comparing the current heading to the target azimuth. This step allowed the robot to determine what direction (left or right) it should turn. The next step determined the distance between the robot location and the target location. The robot program continuously calculates the distance and the minimum turn required. When the heading angle is equal to the azimuth angle, the robot is steered towards the destination point. When the distance is equal to zero, the robot arrives at the destination point

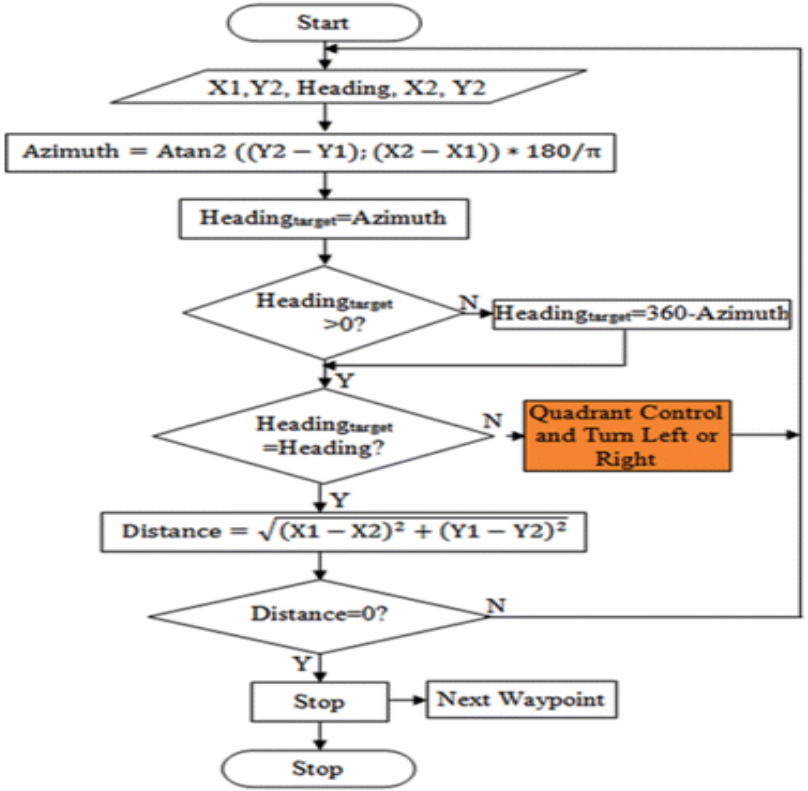

Figure 5 Autonomous robot guidance algorithm

A waypoint is the term used to describe a reference point for robot navigation. It includes destination latitude and longitude data. In this study, waypoints stored previously in the database were used for point-to-point steering in autonomous robot guidance. Two important angles called heading and azimuth angles were used to steer the robot autonomously. The heading angle was an angle between North (true or magnetic) and the current direction of the longitudinal axis of a robot in the horizontal plane. It was continuously received from the GPS receiver by the robot program. The azimuth was an angle between North and the destination point. It was continuously calculated by the robot program with Harvesine equation.

Tasks in this week

- [#16] Definition of the path for the robot car
- [#28] path generation and GPS

- Timeline

Month	Intern week	Tasks
Jan		
	Week 1	Identification of parts and drawing of the chassis diagram.
	Week 2	Circuit diagram and acquisition of parts.
	Week 3	Definition of the path to be followed by the robot car. Laser cutting of the parts.
Feb	Week 4	• Assembly of the robot • Ultrasonic program implementation

Week 5	• GPS and compass navigation • Path definition
Week 6	• Object identification using computer vision. (Raspberry pi & camera)
Week 7	• Transmission of live feed and data from the robot (transmitter and receiver)
Week 8	• Implementation of tillage program on the robot (gathering and casting).