AUTONOMOUS DRIVING CAR

Aman Kumar (B17005)[X] , Amit Chauhan (B17006)[y] , Arpit Singh Bhadauria (B17009)[z] , Aniket Sahu (B17076)[k] , Aman Raj (B17113)[l], Aman Verma (B17114) [m],

B17005@students.iitmandi.ac.in[x],B17006@students.iitmandi.ac.in[y],B17009@students.iitmandi.ac.in[z],B17076@students.iitmandi.ac.in[k], B17113@students.iitmandi.ac.in[l], B17114@students.iitmandi.ac.in[m]

Abstract:

Our aim was to design an autonomous driving car that can move from one coordinate (latitude, longitude) to other using a compass and gps module installed on it. It was also to have a self-balancing platform over it so that it can carry our stuff without dropping it. Infrared sensors were installed on it for obstacle detections such that it can travel without getting hindered from the obstacles that come in its way. We also created a web app which would provide us the waypoints between two coordinates, using Google Maps API, such that those waypoints lie on the road and our car can move on the road. We also created an android app which would connect to the Bluetooth module installed on the car and would keep sending our coordinates to the car which would follow us.

COMPONENTS REQUIRED

Table 1:

Sr. No	Components used	Quantity
1	Arduino Mega 2560	1
2	Breadboard/PCB board	1
3	Bluetooth Module (HC-05)	1
4	1500rpm DC Motors	2
5	200rpm DC Motors	2
6	Motor Driver (L298N)	2
7	12V Lithium Polymer Battery (LiPo battery)	1
8	Power Bank (5V) (To power arduino)	1
9	GPS Module NEO 6-M	1
10	Jumper Wires	Approx. 50
11	Arduino Cable	1
12	Tyres (Size according to chassis)	4
13	Wood/Ply (To make chassis)	1
14	Gyroscope MPU-6050	1
15	Servo Motor	2
	TOTAL PRICE	Rs. 4306/-

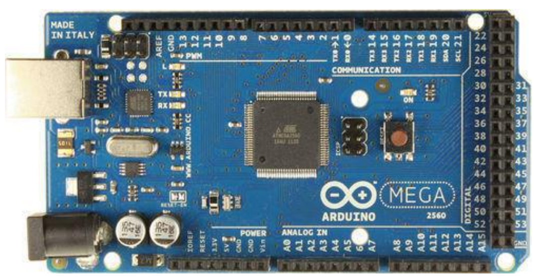

Fig 1. Arduino Mega2560

Arduino Mega2560: It is a microcontroller equipped with digital and analog output input pins that are used to control and communicate with modules attached to it. Arduino IDE is an open source software that can be used to program the microcontroller as per our need.

Fig.2 GPS Module

GPS Module: Neo 6-M module was installed on the car that provided the position of the car to the microcontroller. It works by connecting to the satellite and provides information via the NMEA sentences that contain information such as latitude, longitude, speed, course over ground, date, time, no of satellites connected, etc. The information provided was decoded using Tiny GPS++ library available for Arduino. This module has an accuracy of around 9m which is very poor, hence we needed to provide offset values so that our car can move with better accuracy. There are four pins in the module:

1). RX: Connected to TX of Arduino.
2). TX: Connected to RX of Arduino.
3). GND: Connected to GND of Arduino.
4). VCC: Connected to 5V of Arduino.

Fig 3. Compass

Compass Module: HMC5883L module was used for getting heading direction of the car so that it's heading can be checked and the car can be turned accordingly to get on the course. It is an I2C device and was used as a slave. There are five pins in the module:

1). VCC: Connected to 3.3V of Arduino.
2). GND: Connected to GND of Arduino.
3). SCL: The clock line pin connected to SCL input pin of Arduino.
4). SDA: The data line pin connected to SDA pin of Arduino.
5). RDY: This pin was not connected.

Fig 4. Gyroscope Module

Gyroscope Module: MPU6050 Triple Axis Accelerometer and Gyro module was used to make a self-balance platform. It is a triple axis device which can be used to measure angular velocity and acceleration of the module about all three axes. It is an I2C device. There are eight pins in the module:

1). VCC: Connected to 5V of Arduino.
2). GND: Connected to GND of Arduino.
3). SCL: The clock line pin connected to SCL input pin of Arduino.
4). SDA: The data line pin connected to SDA pin of Arduino.
5). XDA: Used when we use the module as a master device. It is a clock line pin.
6). XCL: Used when we use the module as a master device. It is a data line pin.
7). AD0: Used to change address of the device.
8). INT: Interrupt pin.

Fig 5. Bluetooth Module

Bluetooth Module: HC-05 Bluetooth module was used to communicate with the car to provide latitude and longitude, offset angles, and to get output on the android app. There are six pins in the module:
1). STATE: Used to check if module is working properly.
2). VCC: Connected to 5V of Arduino.
3). GND: Connected to GND of Arduino.
4). RX: Connected to TX pin of Arduino to receive data transmitted by Bluetooth.
5). TX: Connected to RX pin of Arduino to transmit data to the Bluetooth.
6). EN: Used to toggle between Data Mode and AT command mode.

Fig 6. Infrared Module

Infrared Module: IR infrared proximity obstacle sensor module was used for obstacle detection. Three modules were used for obstacle detection in forward, left and right direction. This module works by transmitting an infrared ray and receiving the reflected ray thereby sensing the presence of obstacle in front of it. There are three pins in the module:
1). VCC: Connected to 5V of Arduino.
2). OUT: Output pin of the module which sends LOW input if something is in front and HIGH otherwise.
3). GND: Connected to GND of Arduino.

Fig 7. Servo Motor

Servo Motors: Tower Pro SG90 1.8 kg-cm Micro Servo Motor was used for creating the self-balancing platform. It moves its shaft according to the output given by gyroscope to counter the declination of the platform. There are three pins in the module:
1). VCC: Connected to 5V of Arduino.
2). INPUT: Connected to PWM pin on Arduino to turn the servo by required angle.
3). GND: Connected to GND of Arduino.

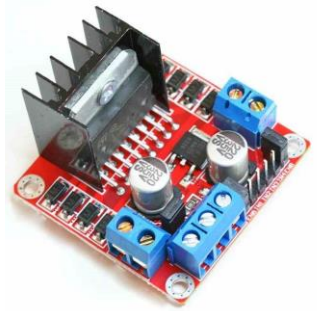

Fig 8. L298N

L298N: L298N H-bridge Dual Motor Controller that is used to control speed and direction of DC motor. It can be used to control two motors using the PWM pins of Arduino to set the duration of the pulse (values between 0-255). 12V power was supplied to the motor driver for driving the motors using LI-PO.

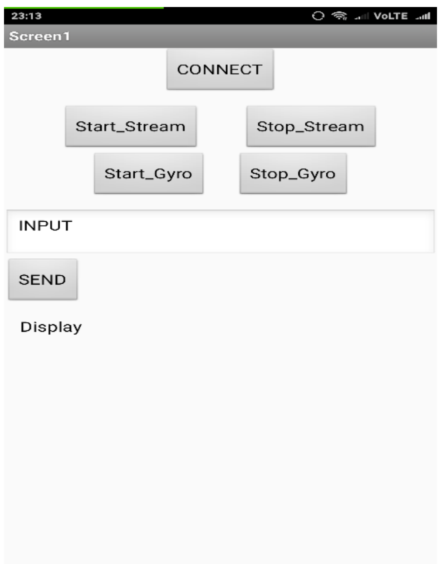

Fig 9. MIT App Inventor

MIT APP INVENTOR: It is an open source online platform hosted by MIT that is used to make android apps easily.

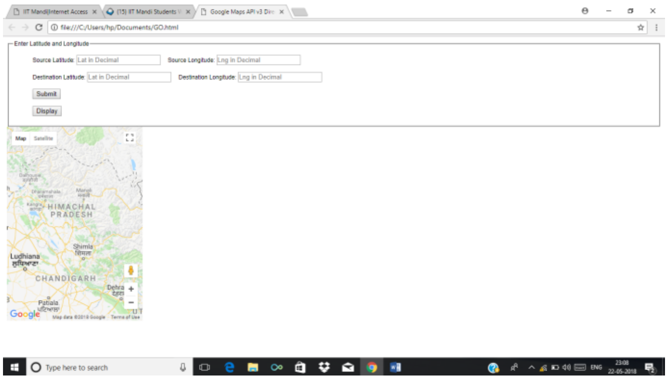

Fig 10. Google Maps API

Google Maps API: Used to create the web app that would provide us the waypoints between two latitudes and longitudes.

INTRODUCTION:

Our idea was to create a car that can move autonomously between two latitudes and longitudes using a gps and compass module which provides the current location of the car and its heading. Using the courseTo function of the TinyGPS++ library we can get the course(angle from the North direction) on which the car should move to reach the destination. Using compass we can get the direction in which the car is pointing and using the course and the heading we can get the angle by which the car should be turned to get on the course for reaching destination.

Gyroscope and servo motors were used to create a self-balancing platform placed on the car so that it can carry our stuff without dropping them.

Gyroscope module placed on the platform gives the angle and angular velocity by which the platform turns and servo motors rotate in the opposite direction to counter the movement of the platform thereby keeping it horizontal.

Three infrared Modules were installed on the car one infront, one on the left side and one on the right side for obstacle detection so that the car can move without getting hindered by any obstacle. If an obstacle is detected infront then car would check if there is an obstacle on its right side if not then it turns right and moves around the obstacle to get back on the course. If there is an obstacle on the right then left side is checked and if all three sensor detect an obstacle then the car moves back and tries to find a way around if there is not a way around then it stops.

Bluetooth module was used to communicate with the car to provide offset angles to compensate for the inaccuracy of the gps module. It is also used to give the destination's latitude and longitude to the car. Android app was created using MIT App Inventor to communicate with the car using Bluetooth module. The app could be used to send users latitude and longitude continuously at an interval of 5 seconds to car so that it can follow the user carrying his/her stuff for him/her. Web app was created using Google Maps API which would provide us the waypoints between two latitudes and longitudes which can be fed to the car so that it can move along the road between the coordinates.

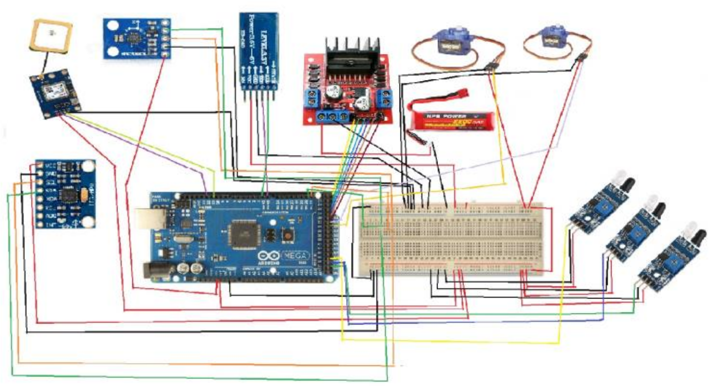

Fig.1 Circuit Diagram

Calculations:

Distance: Distance between two latitudes and longitudes was calculated using Vincenty Formula which is:

Source Latitude: lat1 Source Longitude: long1

Destination Latitude: lat2 Destination Longitude: long2

Taking radius of Earth to be 6372795m.

```
double delta = radians(long1-long2)
double sdlong = sin(delta)
double cdlong = cos(delta)
lat1 = radians(lat1)
lat2 = radians(lat2)
double slat1 = sin(lat1)
double clat1 = cos(lat1)
double slat2 = sin(lat2)
double clat2 = cos(lat2)
delta = (clat1 * slat2) - (slat1 *
clat2 * cdlong)
delta = sq(delta)
delta += sq(clat2 * sdlong)
delta = sqrt(delta)
double denom = (slat1 * slat2) + (clat1
* clat2 * cdlong)
delta = atan2(delta, denom)
distance in m = delta * 6372795
```

Course: Angle from North between two latitudes and longitudes

```
double dlon = radians(long2-long1)
lat1 = radians(lat1)
lat2 = radians(lat2)
double a1 = sin(dlon) * cos(lat2)
double a2 = sin(lat1) * cos(lat2) *
cos(dlon)
a2 = cos(lat1) * sin(lat2) - a2
a2 = atan2(a1, a2)
if (a2 < 0.0)
{
  a2 += TWO_PI
}
Convert a2 to degrees.
```

Fig.3 Expected Design of car

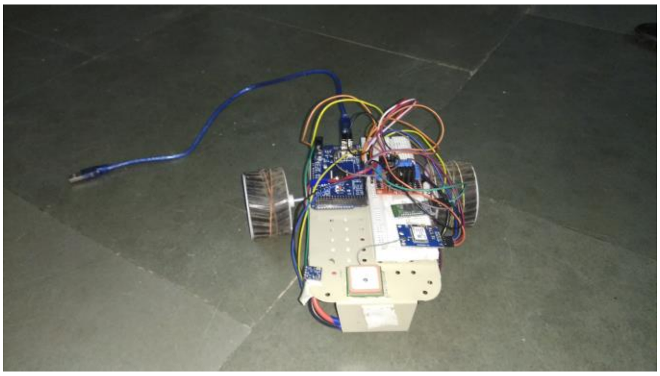

Fig.4 Actual Design of car

Result:

Our car couldn't work as expected because of the inaccuracy of the gps module. We were able to run the car between coordinates by giving required offset angles. We couldn't implement the obstacle detection and self-balancing platform due to ill-functioning of the modules. Hence we had to make our car on a smaller chasis without the platform and obstacle detection functions.

Motivation of the project:

There are many uses of our project in practical world. Our project was developed keeping in mind the uses and improvements that can be done in transportation to improve human life while at the same time trying to keep harmony with the environment.

A self-driving car Tesla Model X saved the life of a lawyer, Joshua Neally, when he experienced a severe pain abdomen and chest. He turned his car to Autopilot mode which took him to the nearest hospital.

It is estimated that 81% of car crashes are the result of human error. Computer controlled vehicles would take the danger of this equation entirely.

USES:

1). Can be used for transporting stuff from one point to other without damaging them.

2). Disabled individuals, who have to rely on public transportation or others to get around, could reap the benefits of self-driving cars.

3). Drivers can use the driving time to do other things while the computer drive the car to take them to their destination.
4). People can travel at night without fear.
5) Can be used in trolleys at airports to.
6). Can be used for data-logging.

FUTURE POSSIBILITES:

More Accurate GPS module could be used so that the car can move more accurately. More Sensors could be installed for improving the obstacle detection and ease of movement. Anti-theft systems can be installed so that the car could become more safe.

Challenges Faced:

The Gps Module used has an accuracy of 9m approx which is not very good hence we had to give offset angles to make the car move properly. Our Gyro module was broken hence we couldn't make the self-balancing platform. Due to ill-functioning of the IR modules we couldn't implement the obstacle detection functioning.

Conclusion:

At the end we could only make a car with Gps and compass module that was able to move from one coordinate to other but by giving offset angles to compensate for the inaccuracy of the gps module. The destination latitude and longitude was given in the Arduino code itself.

References:

1.Arduino Website: https://www.arduino.cc/
2.TinyGPS++ Library: http://arduiniana.org/libraries/tinygpsplus/
3.I2C Devices Library: https://github.com/jrowberg/i2cdevlib/
4. Datasheet HMC5883L: https://cdn-shop.adafruit.com/datasheets/HMC5883L_3-Axis_Digital_Compass_IC.pdf
5. Datasheet GPS Neo 6-m: https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_(GPS.G6-HW-09005).pdf
6.Google Map API: https://developers.google.com/maps/documentation/directions/intro
7. Arduino Query Site: https://arduino.stackexchange.com/questions/24893/mpu6050-code-explanation
8.Modules Tutorials: http://tronixstuff.com