SWARM ROBOTICS

Members:
Abhisek Panda
Abhishek Sahoo
Hemanth Bollamreddi
Mahima Srivastava
Mayank Singh
Megha Agarwal
Prachi Agarwal

Mentors:
Shivashish Kumar Gupta   |   Pratik Bhangale   |   Ganesh Seelam
1. In this project we are trying to mimic the basic features of a bee swarm.

2. The whole search area is to be divided equally among the swarmbots. Each swarm bot will collect data on the availability of a certain goal (colour patch). At the end of surveillance, the bots will communicate among each other and decide the area with maximum targets.

3. Then all the bots will go to the final target for further surveillance.
START

MOVE FORWARD TILL IR SENSOR RETURNS OUTPUT

IS IT THE BOUNDARY?

NO

THAT MEANS OBSTACLE HAS BEEN ENCOUNTERED. FOLLOW ITS WALL TILL YOU REACH THE M-LINE.

TURN RIGHT AND CONTINUE SURVEILLANCE.

YES

TURN AROUND AND CONTINUE SURVEILLANCE.
ALGORITHM TO TRAVERSE TOTAL AREA
**IDEA FOR CIRCUMNAVIGATION**

void wallfollow(float m){
  if(X==m){
    IRM=ON;
    turnright(0);
    return;
  }
  while(IRL==ON){
    if(IRL.data<flag)
      drivemotor(100,80);
    else if(IRL.data>flag)
      drivemotor(80,100)
    else
      drivemotor(90,90)
  }
  turnleft(0);
  wallfollow(m);
}

//m is the x-coordinate of the beginning of obstacle; X is the real time x-coordinate

//IRM=main IR sensor

//To maintain a threshold distance from the wall

//Recursive calling till end

This is wall following without using PID. We will see results using this code and then implement PID after more research.
ALGORITHM TO COLLECT DATA

- Basically we will know the density of color patch by comparing the total time the color sensor receives signal.
- We will divide the entire area into a virtual 2D array. Size of each box will be equal to the size of bot (e.g., if bot size is 10cm, and area is 100x100cm then array will be A[10][10]).
- By default, every box is assigned 0 value. If the bot completes inspection of one box, we assign its value 1.
- So whenever bot enters a ‘1’ box, color sensor turns off.
- In this way we can ensure that whenever bot enters into a previously inspected area, there is no repetition in data collection.
ALGORITHM FOR REACHING FINAL GOAL

- Aim is to follow the straight line between start and goal till any obstacle is encountered (called m-line).
- When obstacle is sensed; circumnavigate around it till it intersects the m-line again.

Advantages:
1. The path of the bot is easily monitored.
2. Most efficient when the amount of area covered by obstacle is less.

Disadvantages:
1. Bot will not be easy to handle if the obstacle spacing is very tight. Eg.
COMMUNICATION USING ESP8266 (WIFI MODULE)
WORK DONE FOR NOW

1. Uploaded a firmware in ESP8266 using FTDI cable.
2. Connected ESP to computer so that it receives commands from the user.
3. Connected it to a hotspot created by mobile and transferred data between them (either way).
4. Created a web server and sent data to a browser.

SOME FEATURES OF ESP

1. AT Commands: These are commands that ESP can understand and follow.
2. ESP can work in 3 modes: (i) Station Mode (Client Mode) (ii) AP Mode (Host Mode OR Access Point Mode) (iii) Both
3. In the project we are going to implement wifi communication for (i) Comparing the density of coloured patch. (ii) Get the coordinates of point of highest density coloured patch.
# Library for creating new serial communication

SoftwareSerial wifiSerial(10,11); // Create a new serial communication port in 10 and 11 pins of arduino

void setup() {
    Serial.begin(9600); // Initial setup
    wifiSerial.begin(9600); // Begin serial communication with ESP8266 at 9600 bits/second
    Serial.println("Trying to connect to ESP8266");
}

void loop() { // Run over and over
    int x=1;
    if (wifiSerial.available()) { // If wifi is connected then write to wifi the commands inputted in computers serial port
        Serial.write(wifiSerial.read());
        if(x=1)
            Serial.write("ESP8266 connected");
        x=0;
    }
    if (Serial.available()) { // Write the output given by wifi in the computer's serial port
        wifiSerial.write(Serial.read());
    }
}
WORK TO BE DONE

1. Automate the connection process using Atmega instead of Arduino.
2. To implement circular linked list type of communication i.e each ESP is in mode 3 and every bot is connected to two other bots as shown in figure.
IR CALIBRATION: FLOWCHART

Start

Note the voltages at known distances

Plot the graph between distance and voltage in MS-Excel and to obtain the equation

Include the equation in the code to continuously obtain distances from the obstacle

End
IR CALIBRATION: INITIAL CODE

Code for obtaining the voltages at known distances.

```cpp
void setup () {
  Serial.begin(2400);
  pinMode(A1, INPUT);
}

void loop (){
  Serial.println(analogRead(A1));
}
```
IR CALIBRATION: GRAPH OF DIST. Vs VOLT

<table>
<thead>
<tr>
<th>Voltage</th>
<th>Distance (in mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>109</td>
<td>20</td>
</tr>
<tr>
<td>105</td>
<td>40</td>
</tr>
<tr>
<td>102</td>
<td>60</td>
</tr>
<tr>
<td>100</td>
<td>80</td>
</tr>
<tr>
<td>97</td>
<td>100</td>
</tr>
<tr>
<td>95</td>
<td>120</td>
</tr>
<tr>
<td>93</td>
<td>140</td>
</tr>
</tbody>
</table>

The graph shows a quadratic relationship between voltage and distance, given by the equation:

\[ y = 0.1707x^2 - 42.019x + 2570.9 \]
Using the equation obtained in the code, the code continuously calculates the distances of obstacles.

```c
void setup() {
  Serial.begin(9600);
  pinMode(A1,INPUT);
}

void loop() {
  int voltage=analogRead(A1);
  int distance=0.1707*voltage*voltage-42.019*voltage+2570.9;
  Serial.println(distance);
  delay(distance);
  delay(100);
}
```
OVERALL TASKS COMPLETED

1. IR sensor calibration.

2. Pseudocode for traversal:
   A. Conversion of mapping area into 2D array and subsequent allocations for avoiding repetitive movement is completed.
   B. Full atmega code for motor control: rectilinear and turning has been completed.

3. ESP8266 configured to receive and transmit signals when connected to an external hotspot.
TIMELINE
<table>
<thead>
<tr>
<th>Day</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IR CALIBRATED</td>
</tr>
<tr>
<td>2</td>
<td>START</td>
</tr>
<tr>
<td>3</td>
<td>BASIC BOT TRAVERSAL ALGORITHM MADE</td>
</tr>
<tr>
<td>4</td>
<td>MAJOR EQUIPMENTS BOUGHT</td>
</tr>
<tr>
<td>5</td>
<td>COMMUNICATION BETWEEN ESP AND TERMINAL OK</td>
</tr>
<tr>
<td>6</td>
<td>STEPPER MOTOR TESTING USING IC</td>
</tr>
<tr>
<td>7</td>
<td>FIRST EVALUATION</td>
</tr>
<tr>
<td>8</td>
<td>COMMUNICATION BETWEEN TWO ESP’S AND ATMEGA</td>
</tr>
<tr>
<td>9</td>
<td>FINALISING COLOR SENSING METHOD</td>
</tr>
<tr>
<td>10</td>
<td>Problem Statement Decided</td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>Notes:</td>
</tr>
<tr>
<td>Sunday</td>
<td>Monday</td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td><strong>PLACING ORDER FOR PCB</strong></td>
<td>13</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>26</td>
<td>27</td>
</tr>
</tbody>
</table>
REFERENCES

http://gritslab.gatech.edu/home/2014/11/the-gritsbot/ : We borrowed the basic design and sensors idea from here and made a few changes to suit our requirements; like omission of overhead camera and inclusion of ESP8226 module for communication.

Some other references:

AVR Programming : Make: AVR Programming: Learning to Write Software for Hardware (By Elliot Williams)
http://www.mlees-robotronics.in/home/robotics/wall-following-algorithm
Reference for ESP :
AT Commands : https://room-15.github.io/blog/2015/03/26/esp8266-at-command-reference/
Flashing Firmware : https://www.youtube.com/watch?v=Gh_pqgfeQc

And many more .....
THANK YOU
for your time