Dwight Look College of ENGINEERING
TEXAS A&M UNIVERSITY

Team 36: Self-Navigating, Obstacle Avoiding Robot
Bi-Weekly Update 3

Teammates:

Arkadi Zhanov

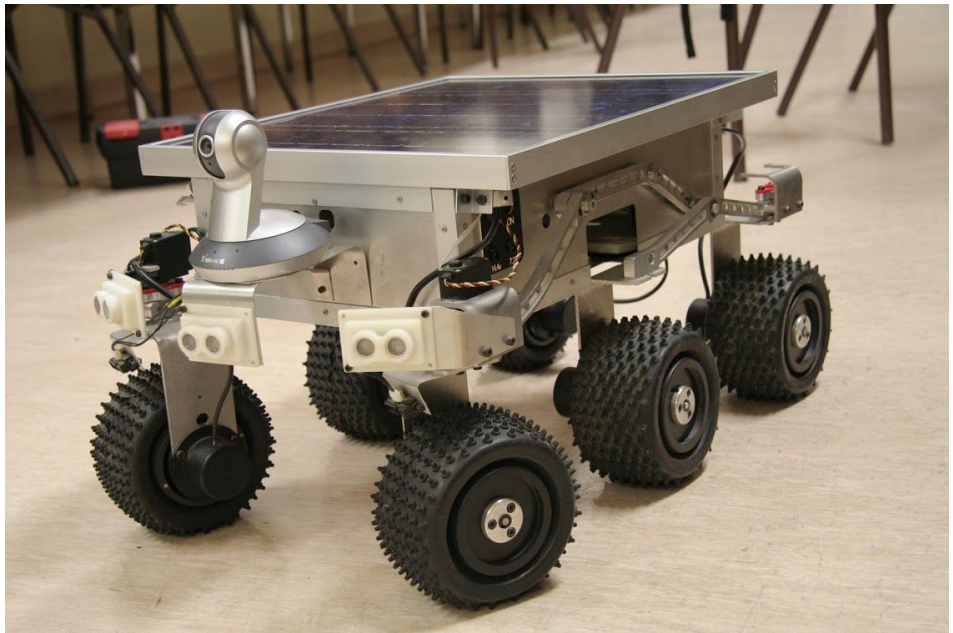
Nathan Sommer

Nikolai Paderin

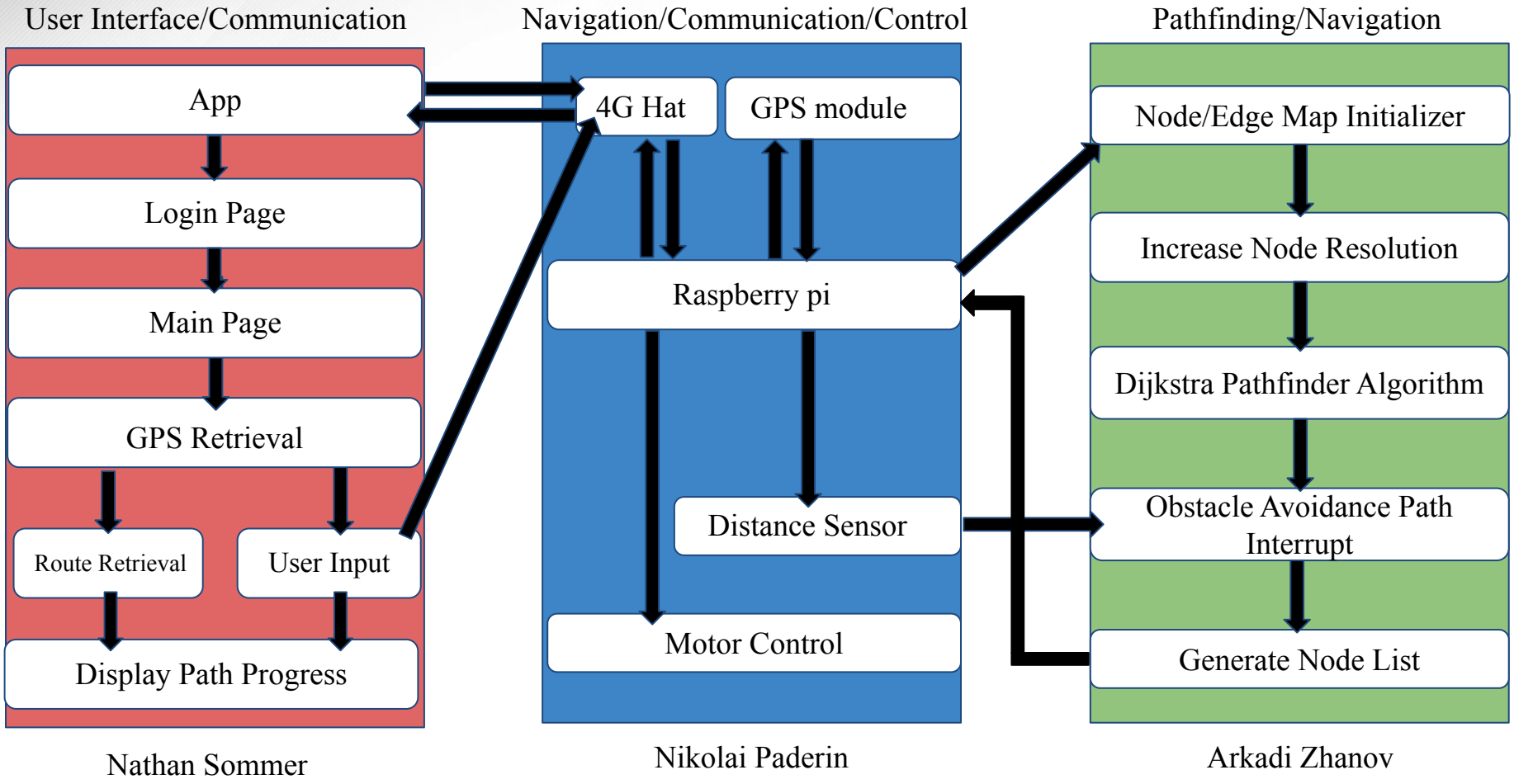
Sponsor: Stavros Kalafatis

Project Summary

- Create a system that takes in a point on a map as the rover's destination, creates a route for the rover, and gets the rover to its destination and back while avoiding obstacles.
- The main motivation for this project is for application in military settings such as aid delivery to wounded soldiers in the battlefield or local, residential delivery of food or mail.



Project/Subsystem Overview



Nathan Sommer

Nikolai Paderin

Arkadi Zhanov

Project Timeline

(Green done, yellow underway, red in trouble, white not started)

Subsystem Designs and Testing (completed 9/11)	Integrate Pathfinder and Movement/Controls and test (to complete by 9/20	Integration with Android App and Raspberry Pi using cellular data (to complete by 10/5)	Final Integration (to complete by 10/15)	Systems Test (to complete by 11/2)	Validation (to complete by 11/26)	Demo and Report (to complete by 12/5)

Interface and Communication

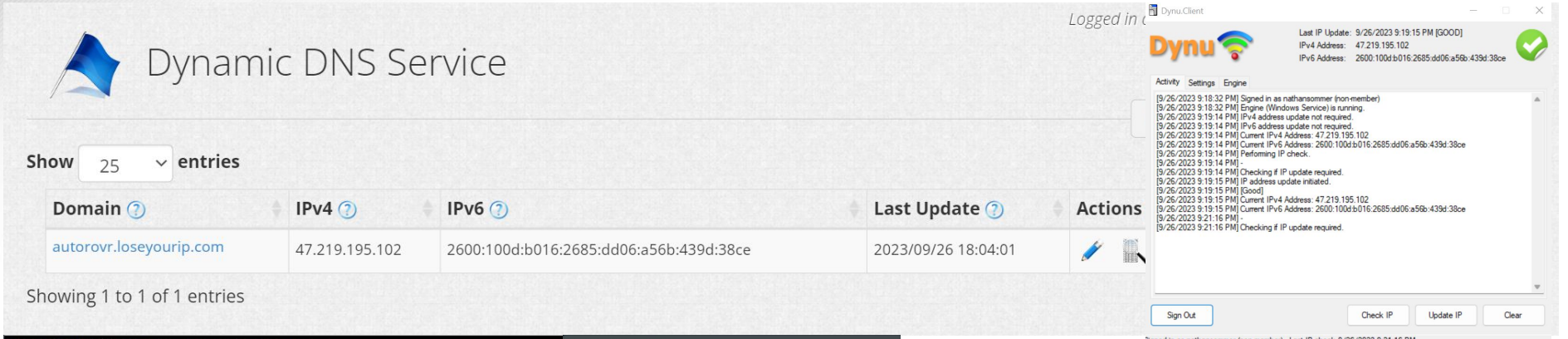
By Nathan Sommer

Accomplishments since last update 30 hrs of effort	Ongoing progress/problems and plans until the next presentation
- Created a DDNS hostname to track the IP address of the client using a client application using Dnyu - Created a VPN Server using WireGuard - Created a VPC using OpenVPN	**Problems:** - DDNS hostname only works when on same network (Can't access hostname when on different network) - Wireguard only works when on same network - Currently trying to run the OpenVPN profile on raspberry pi (Working with OpenVPN helpdesk) **Progress:** - Started Integrating the client code into app

Dwight Look College of ENGINEERING
TEXAS A&M UNIVERSITY

Interface and Communication

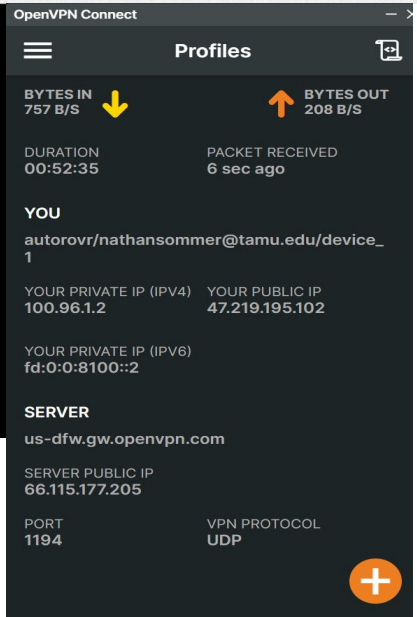
Nathan Sommer




```
root@raspberrypi:/etc/wireguard# sudo wg
interface: wg0
  public key: 0QvbQC/9hWpsuTx5GOue2tGmBamYRcI7iJlmW94WeUo=
  private key: (hidden)
  listening port: 25642

peer: 1dnna7bZOlfRCio57on7jrLkOoPtPjudn8LrCm6+V18=
  endpoint: 192.168.1.151:11110
  allowed ips: 10.9.0.2/32
  latest handshake: 26 seconds ago
  transfer: 5.74 MiB received, 24.05 MiB sent

peer: hQIXJ7hKP9UM9Sic8D/Y7C3KlL5ekvtJi1SEMXNlDGc=
  endpoint: 192.168.1.43:54345
  allowed ips: 10.9.0.3/32
  latest handshake: 2 minutes, 7 seconds ago
  transfer: 124.96 KiB received, 330.93 KiB sent
```




```
class ClientThread implements Runnable {

    2 usages
    private Socket socket;
    no usages
    private BufferedReader input;
    1 usage
    public PrintWriter out;
    2 usages
    public BufferedReader inn;
    @Override
    public void run() {

        try (Socket socket1 = new Socket( host: "100.96.1.4",  port: 12345)){
            System.out.println("Connected");
            out = new PrintWriter(socket.getOutputStream(),  autoFlush: true);
            inn = new BufferedReader(new InputStreamReader(socket.getInputStream()));
            String input;
            MapsActivity.handler.postDelayed(new Runnable() {
                @Override
                public void run() {
                    MapsActivity.dismissPopup();
                }
            },  delayMillis: 2000);
            while(true){
                input = inn.readLine();
                System.out.print("Destination");
            }
        } catch (UnknownHostException e) {
            e.printStackTrace();
            System.out.println("ERROR");
        } catch (IOException e) {
            e.printStackTrace();
        }

    }
}
```

Pathfinding and Navigation

Arkadi Zhanov

Accomplishments since last update 25 hrs of effort	Ongoing progress/problems and plans until the next presentation
- Pathfinder integrated with the Movement/Control subsystem - Developed algorithm and framework obstacle avoidance code which is integrated with the Movement/Control subsystem - Created a common case obstacle avoidance testbench to validate functionality	**Problems:** - Identify best weights and weight vector orders **Ongoing Progress:** - Continue testing obstacle avoidance by adding more test cases to testbench - Finish adjusting obstacle avoidance code as needed - Final validation of obstacle avoidance code - Complete final integration into Movement/Control subsystem

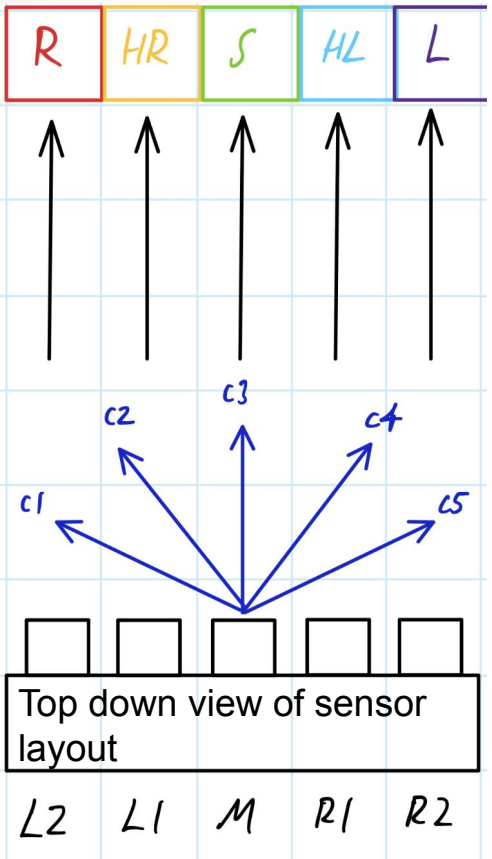
Dwight Look College of ENGINEERING
TEXAS A&M UNIVERSITY

Pathfinding and Navigation

Arkadi Zhanov

Obstacle Avoidance Overview

Potential Obstacles and Desired Direction Based on Position



Top down view of sensor layout

Vector Determination and Case Classifier Table

Cases	Type	Sensor Order (Highest to least effect
c1	Hard Left	R1, R2, M, L1, L2
c2	Left	R2, R1, M, L1, L2
c3	Straight	M, (L1/R1), (L2/R2)
c4	Right	L2, L1, M, R1, R2
c5	Hard Right	L1, L2, M R1, R2

This table is used to determine weight and set up weight vectors for the cases

Sample Output From Classifier for Obstacle Avoidance

```
Hard Left Case
Scenario: [10, 10, 10, 1, 10]
The cases values are: [820, 1330, 1120, 1420, 1711]
the case determined is: c1
the case type is: hard left

Left Case
Scenario: [10, 10, 10, 10, 1]
The cases values are: [1270, 610, 760, 1501, 1711]
the case determined is: c2
the case type is: left

Straight Case
Scenario: [1, 10, 10, 10, 1]
The cases values are: [1261, 601, 310, 601, 1261]
the case determined is: c3
the case type is: straight

Right Case
Scenario: [1, 10, 10, 10, 10]
The cases values are: [1711, 1501, 760, 610, 1270]
the case determined is: c4
the case type is: right

Hard Right Case
Scenario: [10, 1, 10, 10, 10]
The cases values are: [1711, 1420, 1120, 1330, 820]
the case determined is: c5
the case type is: hard right
```

Navigation/Movement/Control

Nikolai Paderin

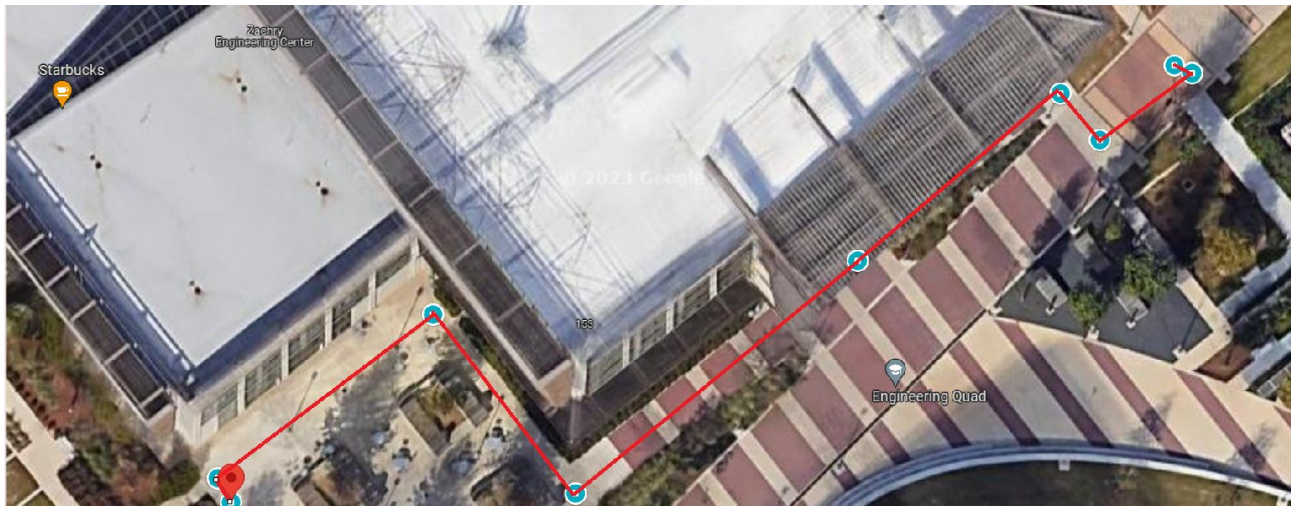
Accomplishments since last update 12 hrs of effort	Ongoing progress/problems and plans until the next presentation
Node Navigation completed -Able to seek nodes autonomously -Segmented movements resolution in terms of location and direction(0.1 meter segmentations) Integration with other subsystem -Able to orient itself towards other nodes and navigate towards nodes - Fixed issues with Rover -⅓ chance of misfire command -movement refined to reduce over/undercompensation -re-allocated sensors to allow for greater object resolution detection	Object Avoidance integration -Adding corner cases to account for niche cases -account for road angles 4G Hat -Test out GPS accuracy + Precision -Test Internet connectivity -Send and receive instructions from application

Navigation/Movement/Control

Nikolai Paderin

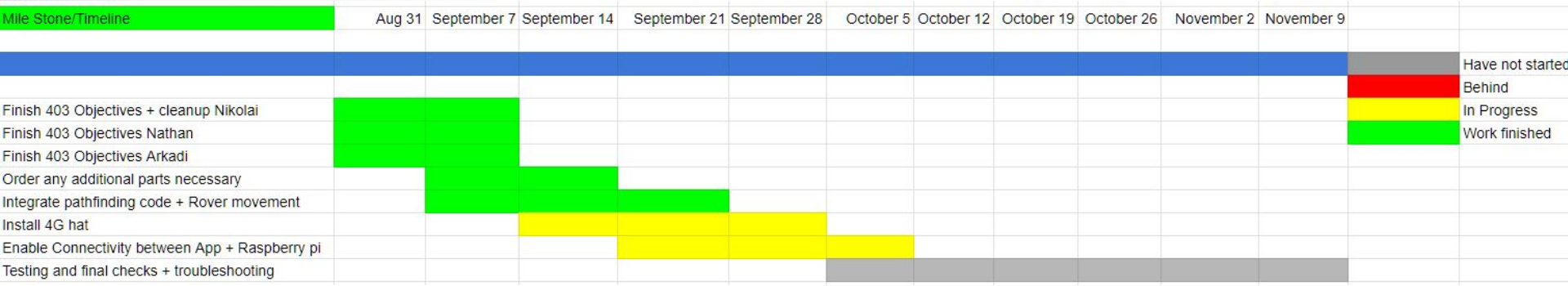
Latitude	Longitude
30.62106	-96.33951
30.6210661	-96.3395267
30.6210051	-96.3395974
30.6210433	-96.3396346
30.620907	-96.3398266
30.6207174	-96.3400937
30.620863	-96.340229
30.6207297	-96.3404325
30.62071	-96.34042

```
Step 1: Walk 1.74 m and turn to face 293.00 degrees.
Step 2: Walk 9.58 m and turn to face 224.93 degrees.
Step 3: Walk 5.54 m and turn to face 320.04 degrees.
Step 4: Walk 23.82 m and turn to face 230.48 degrees.
Step 5: Walk 33.13 m and turn to face 230.48 degrees.
Step 6: Walk 20.73 m and turn to face 321.35 degrees.
Step 7: Walk 24.47 m and turn to face 232.72 degrees.
Step 8: Walk 2.50 m and turn to face 151.36 degrees.
```



right	68.07
right	95.11
left	89.56
no turn	0
right	90.87
left	88.63
left	81.36
left	151.36

Execution & Plan



Thank you for your patience!
Any questions?