2D Autonomous Mapping

Amukta, Tiffany, Kanupriyaa
Team B4

Design Concept

- **Need** - Autonomous vehicles are often used to explore difficult to reach places (i.e. Search and Rescue, Scientific Exploration).
- **Solution** - This project aims to create a low cost, educational demonstration of **autonomous mapping and navigation** by building a vehicle that explores a simulated environment.
- **Scope** - Eliminate mechanical design challenges by reducing a real, textured environment to a **static environment with smooth ground and flat walls.** We will build a ground vehicle capable of producing a **2D map of the maze**, traversing through it, and updating live changes to the maze
- **ECE Areas**: Software and Signals

System Specifications

1. Small wheeled vehicle capable of exploring an area with smooth surface.
2. Vehicle needs to traverse area above a minimum decided speed.
3. Must have a battery life of at least 2 hours.
4. Vehicle should keep track of map coverage (previously decided).
5. Software should be able to simultaneously do localization and mapping of the area.
6. Vehicle should travel efficiently without hitting any walls.
7. User must be able to start the vehicle from a webapp on a laptop.
8. User must be able to see the map being updated as vehicle moves.

Localization & Mapping Solution:

We will build a software for our ground vehicle that simultaneously builds a map of the area and determines the robot's (own) position within the map using the SLAM (Simultaneous Localization and Mapping) method.

The combination of these components determines what the final algorithm would be:

- **Mapping**: Grid Maps
- **Sensing**: Single Point Lidar (RPLidar A1 ~$150)
- **Sensor mode**: Raw data aka 2D point cloud
- **Kinetics**: Previous commands given + odometry data
- **Loop closure**: Using kinetics data

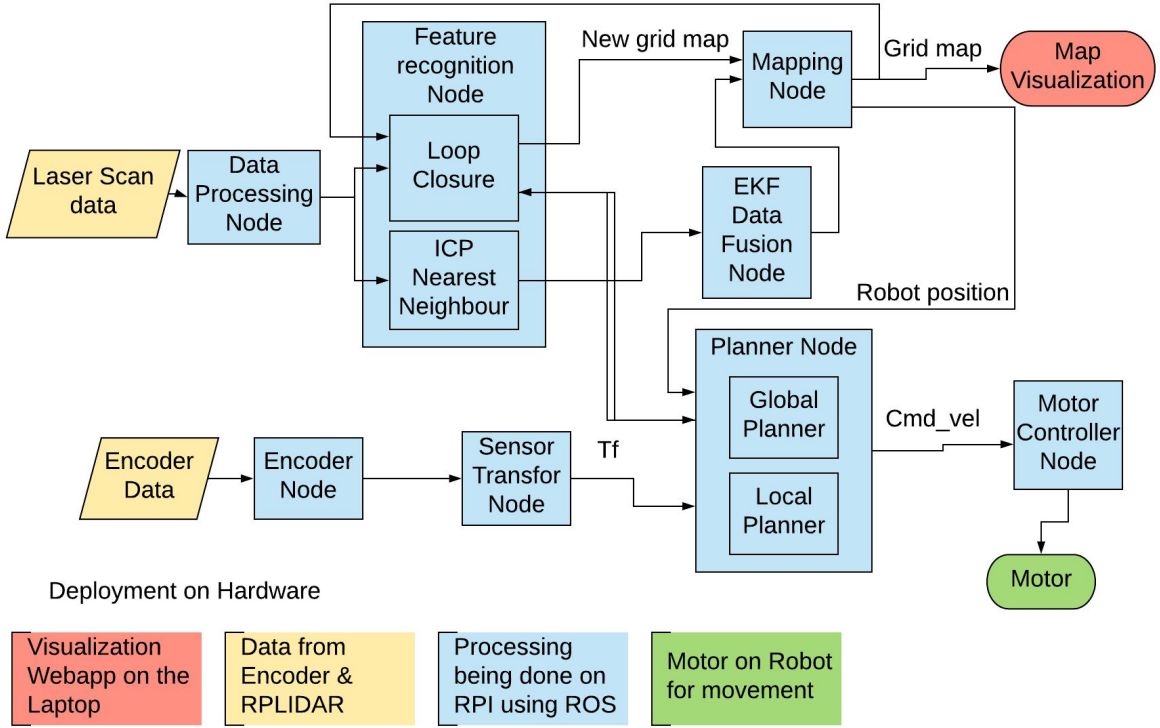
- **Not using**: cameras, depth cameras, ranged lidar, topological maps, multiple object recognition, or landmark based mapping

Autonomous Mapping Solution:

We will build a graphical representation of the map and implement a modified graph based search to verify map coverage.

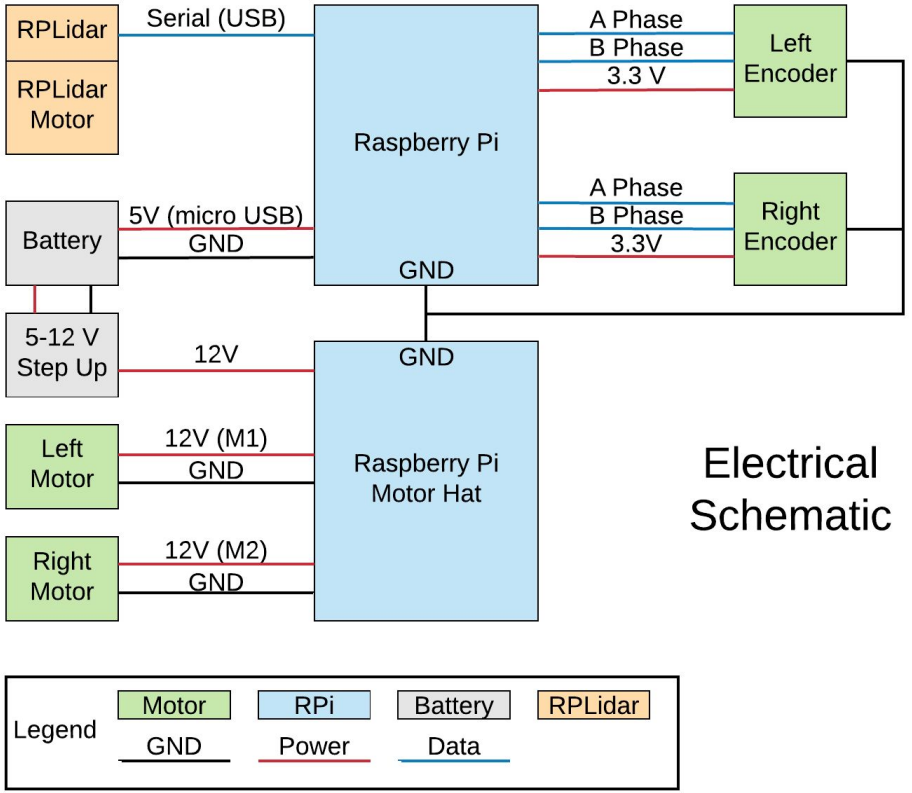
- **Store** grid map coordinates of traversed “waypoints” as graph nodes
  - Waypoints: approximate center point of each maze segment
- **Connect** graph nodes using grid map data
- **Record** unexplored paths connected to known nodes
  - Modified graph based search to determine order of exploring unknown paths
- **Check** for 36 graph nodes to verify map coverage (6x6 grid maze)
- Loop closure algorithm checks if any of the nodes have already been explored by comparing current grid map with previous scanned data and avoids those nodes.

System Architecture Diagram



Electrical Schematic

- Lidar: RPLidar A1M8 (acquired)
  - 4 bytes per point (1 degree)
  - 1980 points / sec
  - Serial port baud rate 115200
- Motor: 12V, 350 rpm DC motor + encoder
  - 229 ft / m
- Battery: Portable phone charger, 5V USB



Electrical Schematic

Risk Assessment

Risk	Likelihood	Mitigation Strategy
Robot runs out of battery during demo	low	Add an additional battery pack and make sure battery is fully charged before a new run.
Not enough processing power on RPi	low	Offload processing to laptop (IO delay, not autonomous), or to additional RPi
Lidar burns out	med	Purchasing 2nd Lidar (anticipated in budget)
Robot too slow	med	Reduce number of scans taken by LIDAR to increase speed (decreases accuracy of map).
Low accuracy localization	high	Add camera feedback, tune SLAM configurations
Robot stuck on walls / distance to walls under RPLidar range	high	Improve localization to path planning controls and tune configurations. Alternatively, add limit switch bumpers for additional sensing.

Performance Metrics

Metric	Measurement Indicator	Target
Vehicle Speed	Encoder, # map segments explored	> 20ft/min
Map coverage	Grid map completeness (pixels)	> 95% of map explored
Efficiency	# dead ends, # unnecessary paths revisited	< 0, 2 maze segments
Efficiency	Time to complete maze mapping	< 5 mins
Localization accuracy	Estimated to actual position distance	< 1 in
Battery Life	Time passed with robot operating continuously	2 hours

Validation Plan

Maze Design:

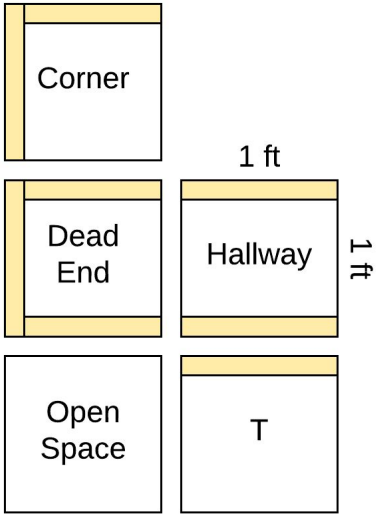
- 1 ft wide hallways
- Modular 1x1x1 ft sections, combined to form different maze configurations

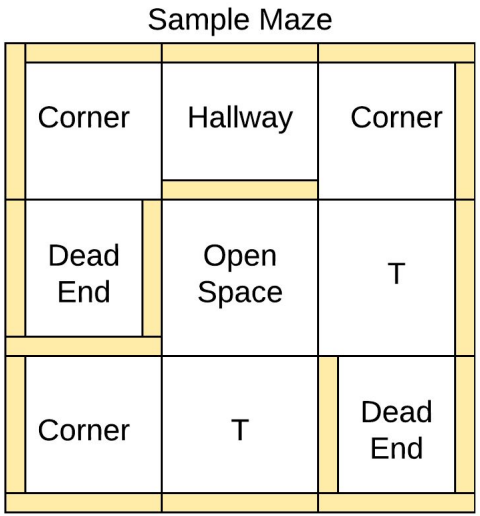
Phase 1 Testing:

- Build small maze with 4-6 sections to test basic navigation (without autonomous mapping function)

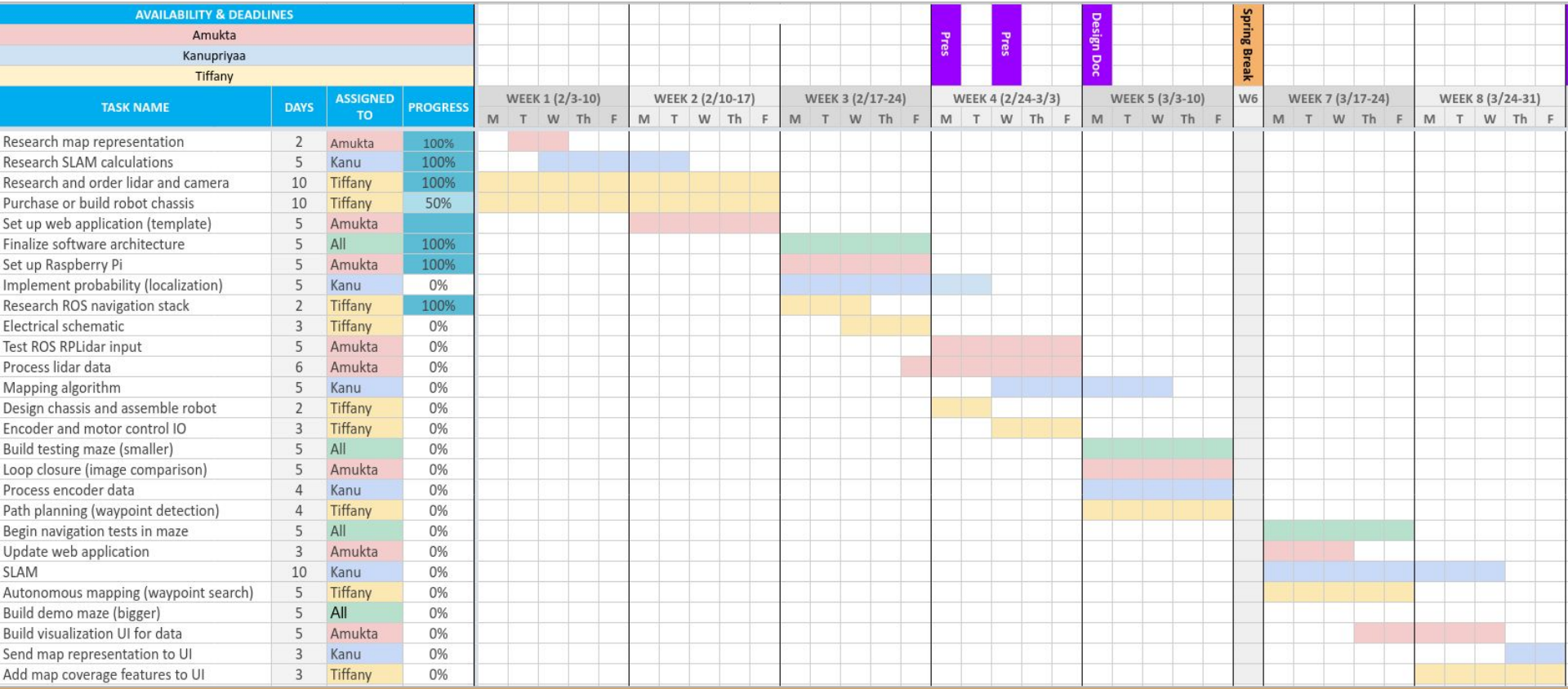
Phase 2 Testing:

- Expand maze to 6x6 grid demo size
- Test map coverage on various configurations
- Concurrently test map coverage algorithm with software simulations





Gantt Chart



AVAILABILITY & DEADLINES
Amukta
Kanupriyaa
Tiffany

Pres | Pres | Design Doc | Spring Break

TASK NAME	DAYS	ASSIGNED TO	PROGRESS
Research map representation	2	Amukta	100%
Research SLAM calculations	5	Kanu	100%
Research and order lidar and camera	10	Tiffany	100%
Purchase or build robot chassis	10	Tiffany	50%
Set up web application (template)	5	Amukta	
Finalize software architecture	5	All	100%
Set up Raspberry Pi	5	Amukta	100%
Implement probability (localization)	5	Kanu	0%
Research ROS navigation stack	2	Tiffany	100%
Electrical schematic	3	Tiffany	0%
Test ROS RPLidar input	5	Amukta	0%
Process lidar data	6	Amukta	0%
Mapping algorithm	5	Kanu	0%
Design chassis and assemble robot	2	Tiffany	0%
Encoder and motor control IO	3	Tiffany	0%
Build testing maze (smaller)	5	All	0%
Loop closure (image comparison)	5	Amukta	0%
Process encoder data	4	Kanu	0%
Path planning (waypoint detection)	4	Tiffany	0%
Begin navigation tests in maze	5	All	0%
Update web application	3	Amukta	0%
SLAM	10	Kanu	0%
Autonomous mapping (waypoint search)	5	Tiffany	0%
Build demo maze (bigger)	5	All	0%
Build visualization UI for data	5	Amukta	0%
Send map representation to UI	3	Kanu	0%
Add map coverage features to UI	3	Tiffany	0%

WEEK 1 (2/3-10)					WEEK 2 (2/10-17)					WEEK 3 (2/17-24)					WEEK 4 (2/24-3/3)					WEEK 5 (3/3-10)					W6	WEEK 7 (3/17-24)					WEEK 8 (3/24-31)				
M	T	W	Th	F	M	T	W	Th	F	M	T	W	Th	F	M	T	W	Th	F	M	T	W	Th	F		M	T	W	Th	F	M	T	W	Th	F