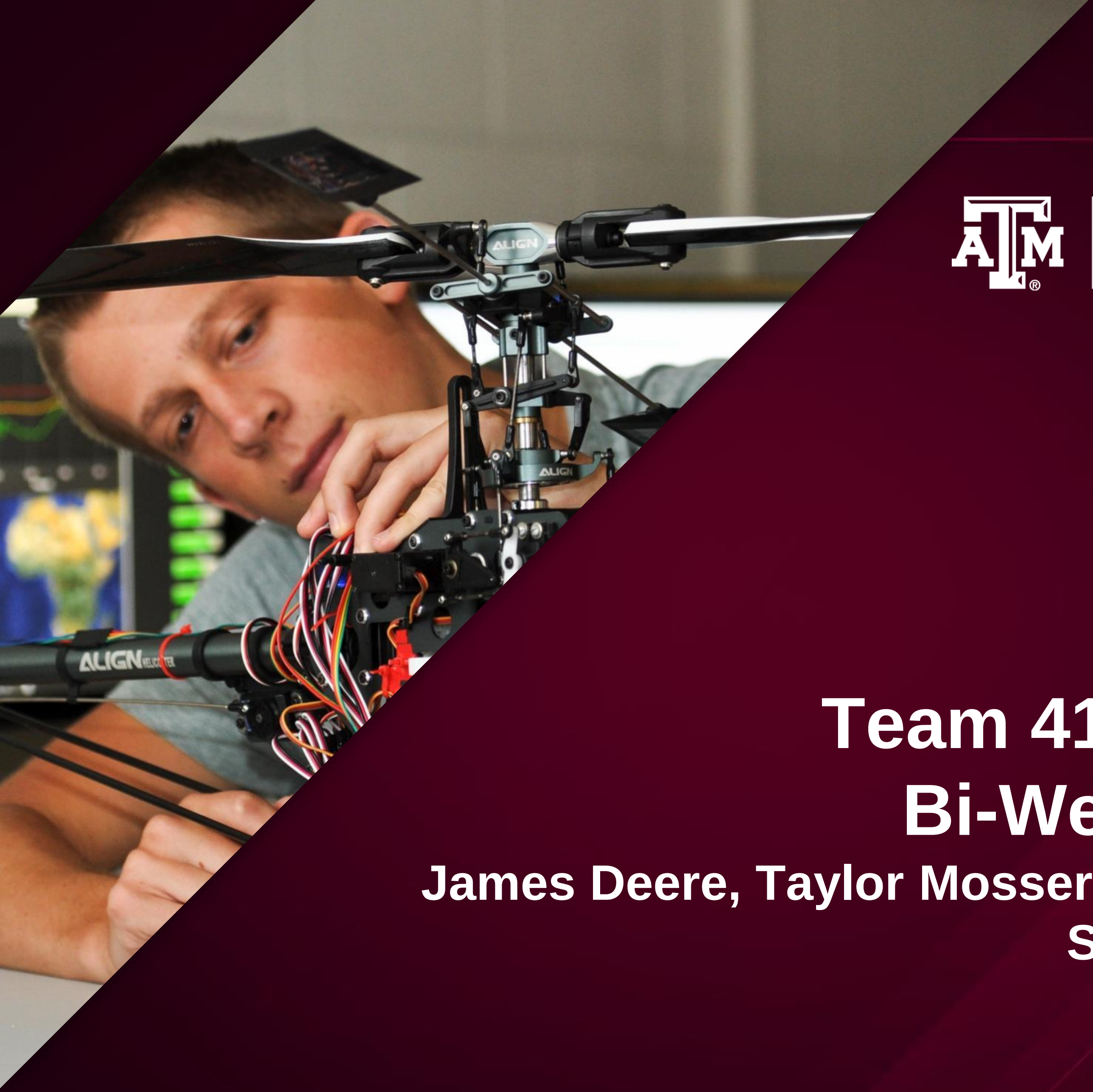


Dwight Look College of ENGINEERING TEXAS A&M UNIVERSITY

Team 41: Roombotics Bi-Weekly Update 4

James Deere, Taylor Mosser, Todd Van Klaveren

Sponsor: Eric Robles

TA: Max Lesser

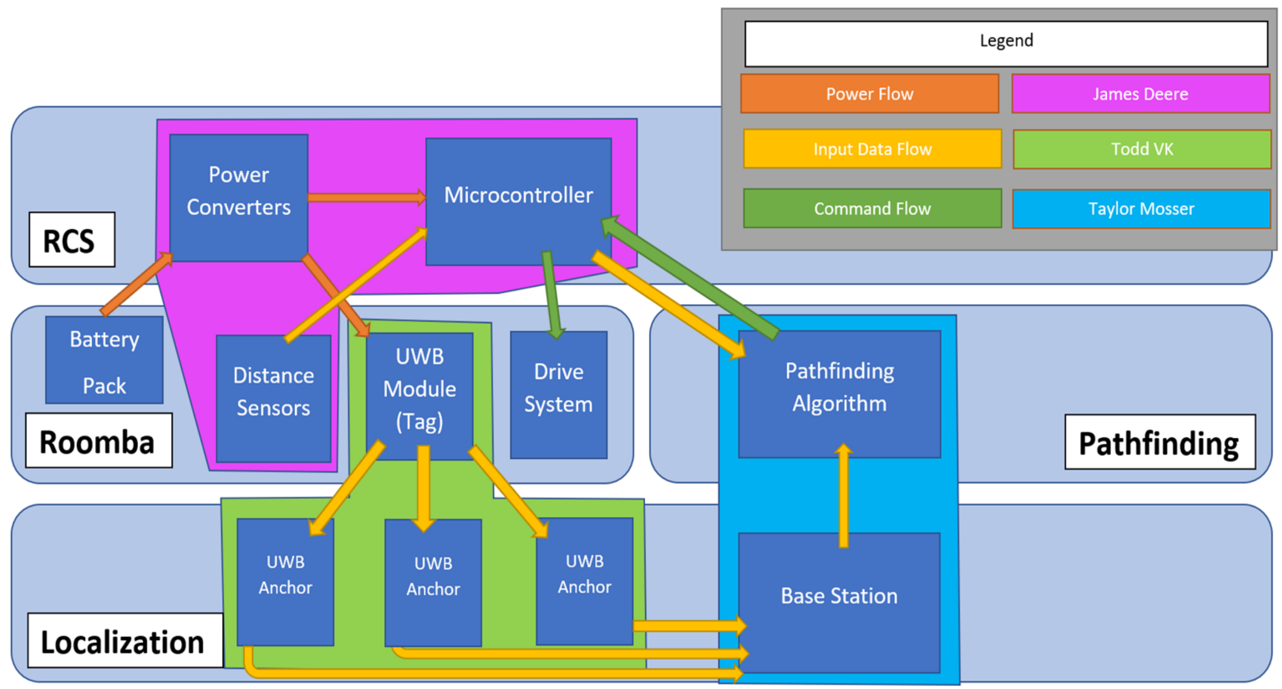
Roomba-MRS Project Summary

Problem: iRobot Roombas do not follow a specific path when cleaning a room and are prone to repeatedly cleaning the same areas or avoiding other areas altogether. While the Roombas can map the layout of a room, they do not have strategic methods of navigating the space to clean quickly and efficiently.

Additionally, one Roomba can be ineffective in cleaning larger spaces due to its small size. This system will allow for more than one Roomba to be deployed in a space in which they can work together to cover all area in the room.

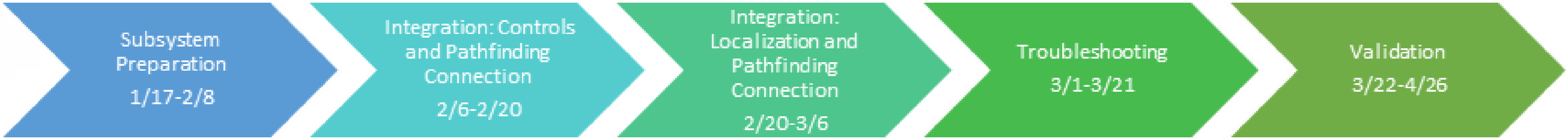
Solution: Roomba will navigate a space using the A* pathfinding algorithm to calculate lowest cost path from one point to another. Path calculation will be performed based on the location of the Roomba and any obstacles in the room. Roombas will not interfere with each other's assigned cleaning spaces and will avoid colliding with each other.

Project Overview



- Pathfinding Algorithm decides best path for each Roomba
- Base Station sends movement commands via Bluetooth
- Localization determines actual location of Roombas
- Feedback from localization updates Pathfinding algorithm.

Project Timeline



Subsystem Preparation
1/17-2/8

Integration: Controls and Pathfinding Connection
2/6-2/20

Integration: Localization and Pathfinding Connection
2/20-3/6

Troubleshooting
3/1-3/21

Validation
3/22-4/26

All subsystems have been integrated. Currently in troubleshooting and validation phases.

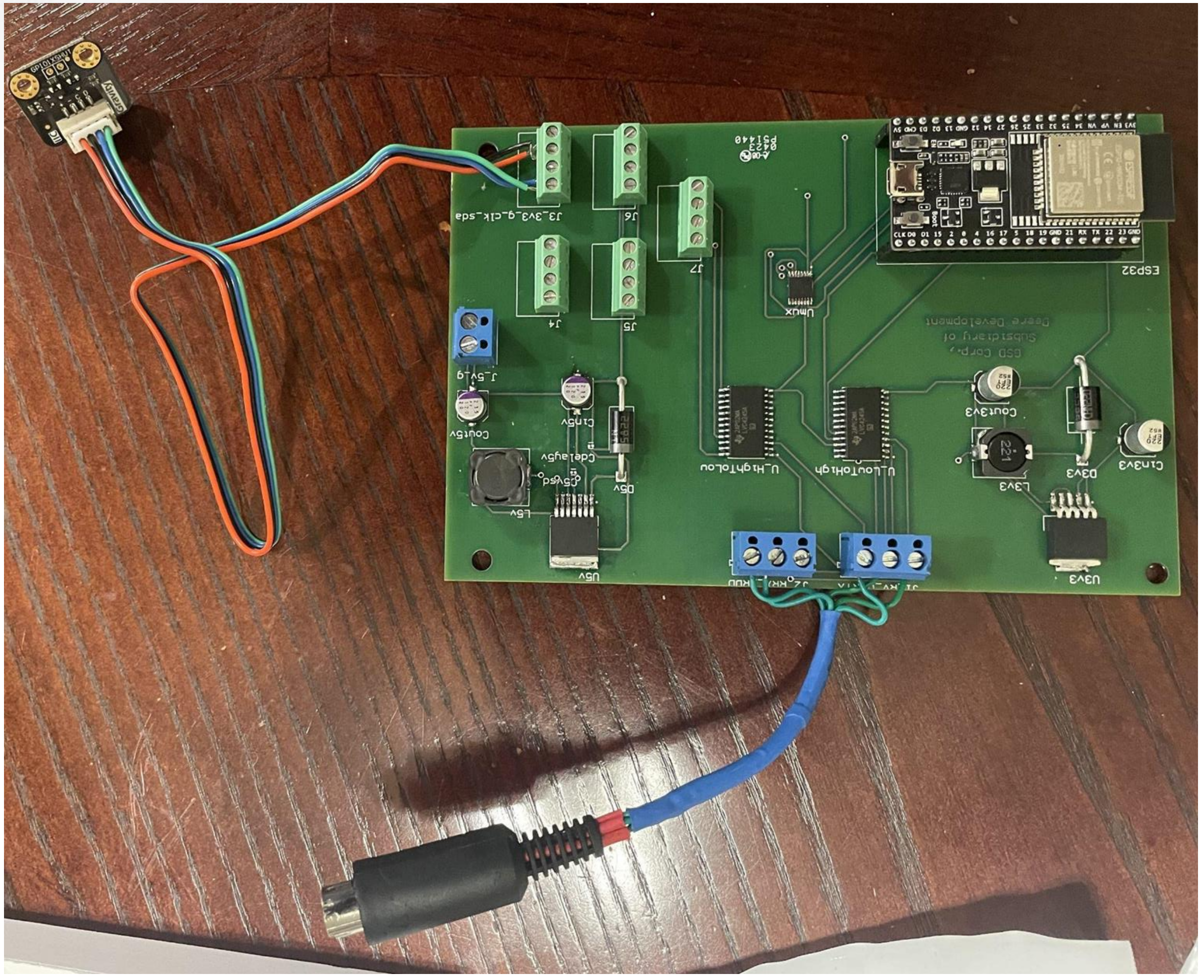
Roomba Control Subsystem

James Deere

Accomplishments since last update 25 hrs of effort	Ongoing progress/problems and plans until the next presentation
● Added command functions to Roomba control firmware to support the A* algorithm ● Soldered Mini Din7 connector onto PCB (compatible connector with Roomba) ● Obstacle detection with Roomba bumper sensor, and 5 IR distance sensors.	● Smoothing out turning in control firmware for better accuracy ● Ongoing learning of 3D CAD software to elegantly mount PCB, sensors, and localization module onto Roomba ● Update firmware to avoid obstacles with obstacle detection data from 5 distance sensor

Roomba Control Subsystem

James Deere



Pathfinding

Taylor Mosser

Accomplishments since last update 26 hours of effort	Ongoing progress/problems and plans until the next presentation
- Completed identical scene view design for running control of active Roombas alongside simulation - Integrated Unity project with controls and localization subsystems - All UI buttons and functions work in manually controlling the Roomba - Override status complete: console message, indicator light	- Make changes to Pathfinding and AIStarPath scripts for path calculation and Roomba control to be tested on real-life Roomba

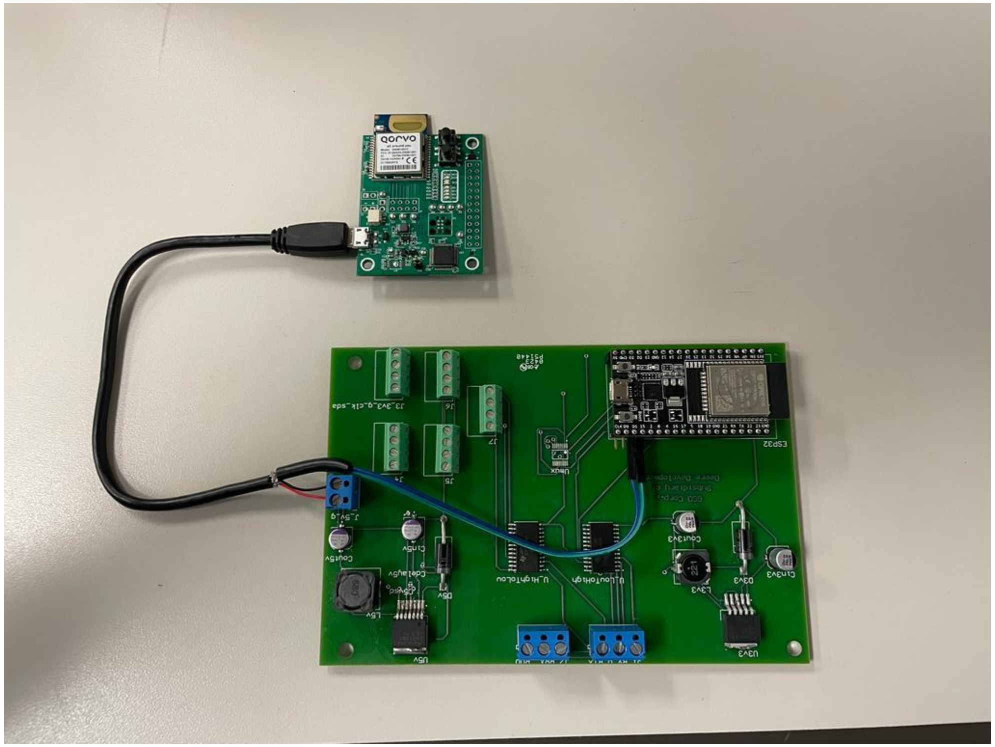
Localization/Integration Design

Todd Van Klaveren

Accomplishments since last update 27 hrs	Ongoing progress/problems and plans until the next presentation
Integrated Serial Library into Unity. Unity app polls for location from base station. (This is being changed) Built and tested second board for testing / second Roomba. Attached Anchor tag to control board with power and UART communication lines.	Expand ESP32 code to receive location data from anchor tag, send to Unity over bluetooth (partially working, need to verify working with Roomba controls traffic at same time). Use location data to display Roomba position in Unity display.

Localization/Integration Design

Todd Van Klaveren



Integrated System

All Team Members

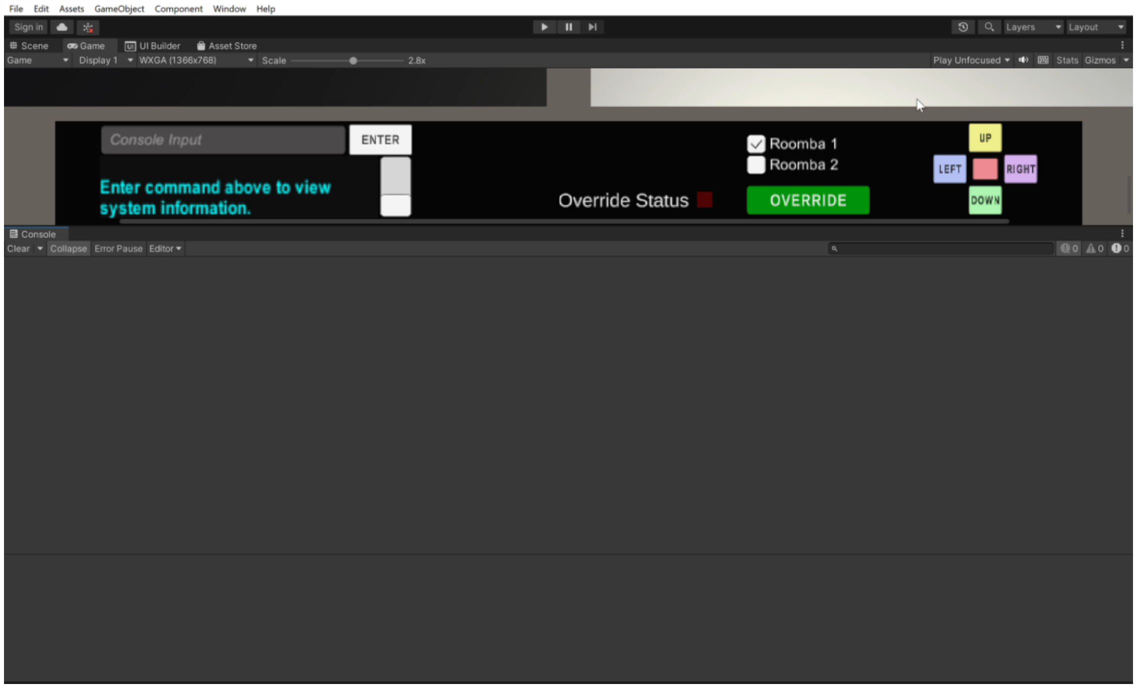
Accomplishments since last update	Ongoing progress/problems and plans until the next presentation
Location Data is transferred via serial USB connection into Unity. Unity is able to search for Roomba, send a larger set of commands, and receive messages from the Roomba. Roomba Control can detect objects in front of it. UI has movement control of Roomba, with console output for received messages.	Changing locating method to improve accuracy (The Roomba module will get position and send a message via the ESP32 bluetooth connection.) Program the AI to issue the determined commands to the Roomba based on AI path determination. Program Roomba to send a message containing info about obstacles detected.

Integrated System

All Team Members

Integrated System

All Team Members

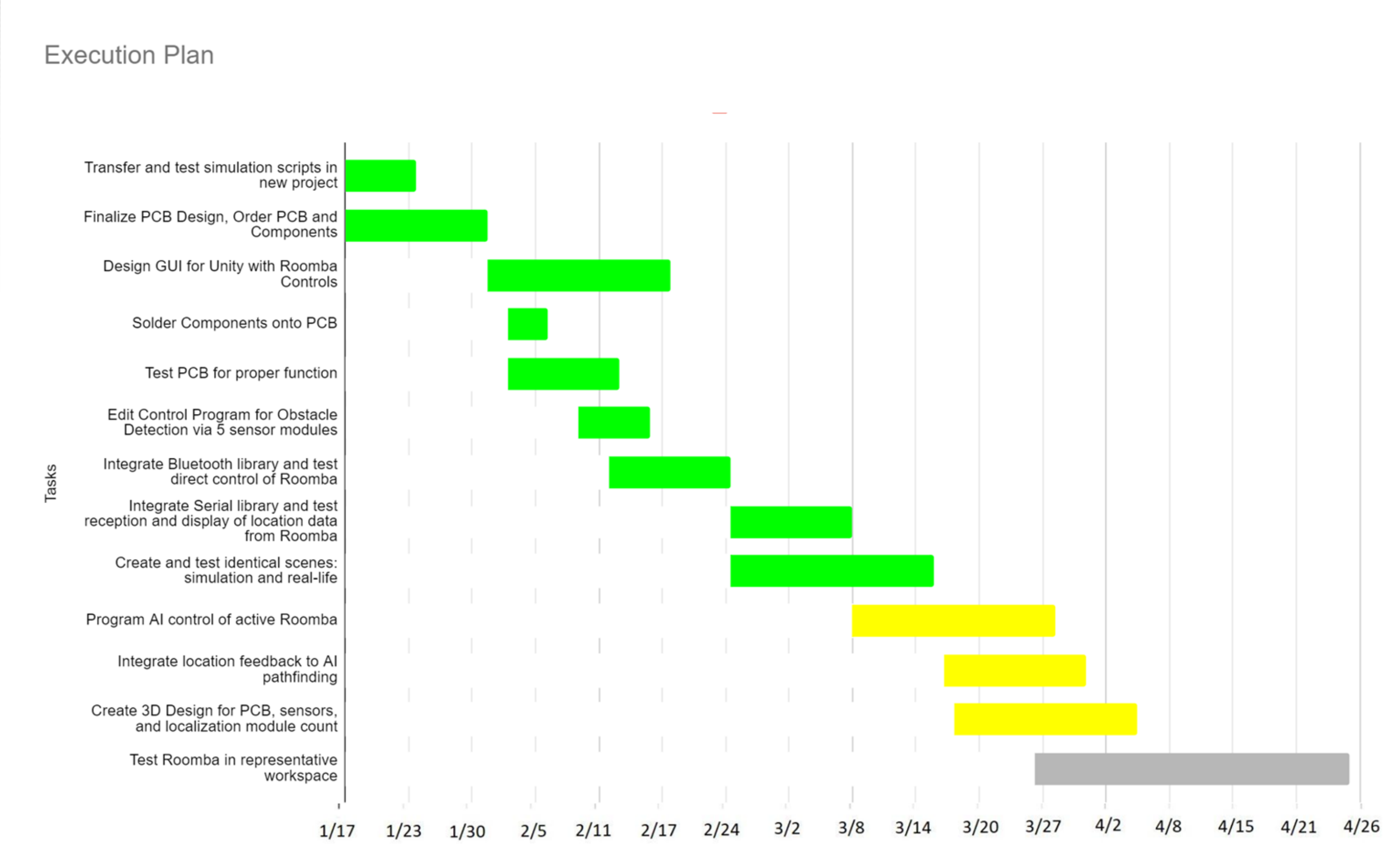


Enter command above to view system information.

☑ Roomba 1

□ Roomba 2

Execution Plan



🔴 Behind Schedule	🟢 Completed	🟡 In Progress	⚪ Not Started

Validation Plan

FSR	Test Name	Success Criteria	Methodology	Status	Responsible Engineer(s)
3.2.1.6	Object Detection	Roomba can detect an object in front of it (PCB and 5 sensor configuration)..	Obstacles will be placed in front of Roomba at various sizes and angles to determine if there are blindspots.	UNTESTED	James Deere
3.2.3.3	Voltage Converters	The two converters used on the PCB output 3.3V at 1 amp and 5V at 750 mA respectively	A multimeter and E-load is used to check the output voltage and current levels.	SUCCESS	James Deere
4.1.3	Mounting	The PCB, UWB module, and sensor modules shall be mounted onto the Roomba in a sleek and elegant way, allowing for no blindspots.	Mount adds less than 50% to the height of Roomba and allows for no blindspots.	UNTESTED	James Deere, Todd Van Klaveren
3.2.1.5	Command Response	Roomba can receive commands from the base station and respond accordingly.	A set of known commands will be flashed to the MCU from the base station via Bluetooth	SUCCESS	Team
3.2.1.6	Object Response	Roomba stops and sends an alert of the obstacle's presence to the base station.	Obstacle will be placed in Roomba's commanded path, Roomba should stop once obstacle is detected and send bluetooth signal alerting the base station that there is an obstacle.	UNTESTED	Team
3.2.1.7	Localization	Localization system reports Roomba location to Base Station	Base Station Application will output location data. Data will be compared to measurements of Roomba. Roomba must be within 13" circle of reported.	UNTESTED	Taylor Mosser, Todd Van Klaveren
3.2.1.5	Roomba Control	Base station manual control is able to issue commands to Roombas individually.	Roomba must execute commands being sent from base station application inputs made using the GUI.	SUCCESS	Team
3.2.1.1	AI Control	Pathfinding AI is able to issue commands to Roombas individually.	A* algorithm empty object and AI pathfinding script can be used in multiple scenes without interference. Commands to real-life Roombas can be given through the use of Unity GUI.	UNTESTED	Taylor Mosser
3.2.1.1, 3.2.1.2, 3.2.1.6, 3.2.1.7, 3.2.4.4	In use Testing	Roomba navigates the test environment cleanly (no bumps) and updates path according to discovered obstacles.	A successful test will have no collisions between Roombas, obstacles, or room boundaries. The roomba will report any found obstacles to the base station. The pathfinding algorithm will adjust and execute an appropriate path.	UNTESTED	Team

Questions?

Thank you!