Dwight Look College of
ENGINEERING
TEXAS A&M UNIVERSITY

Team 41: Roombotics
Bi-Weekly Update 5

James Deere, Taylor Mosser, Todd Van Klaveren
Sponsor: Eric Robles
TA: Max Lesser

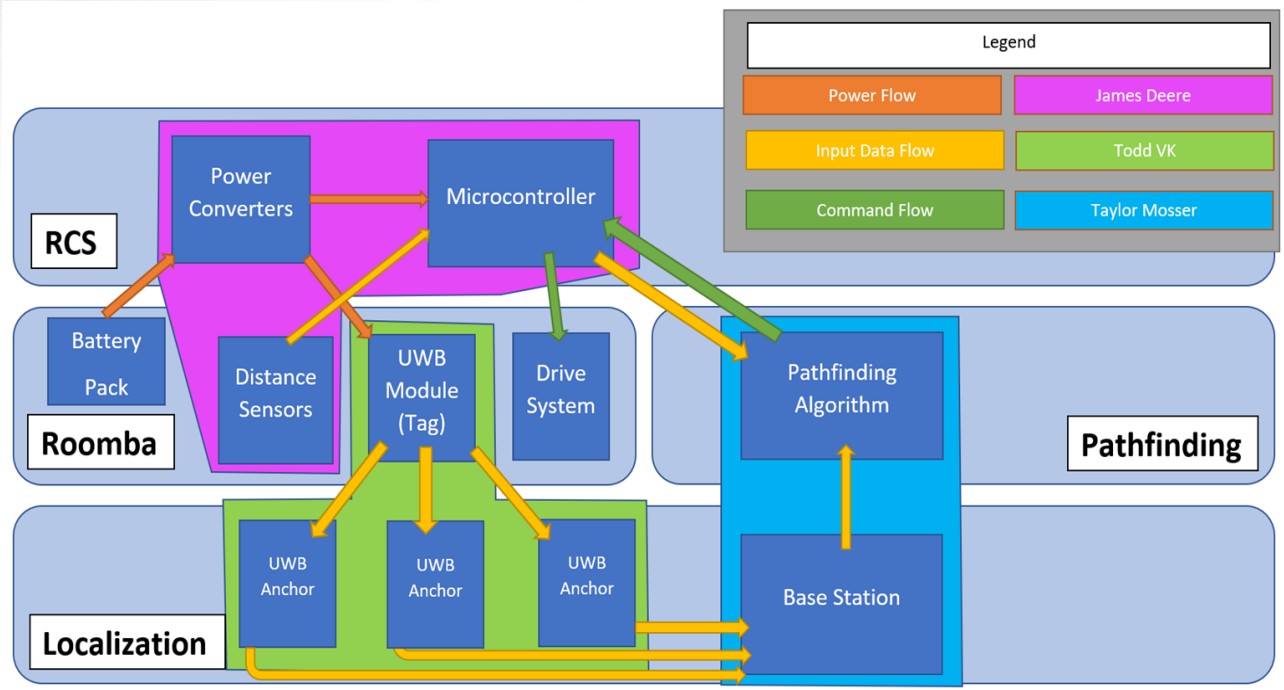
Roomba-MRS Project Summary

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

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

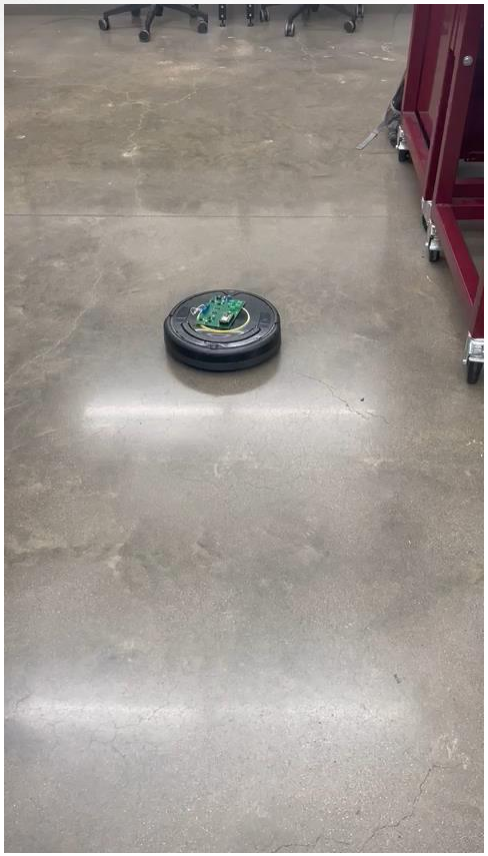
Integrated System

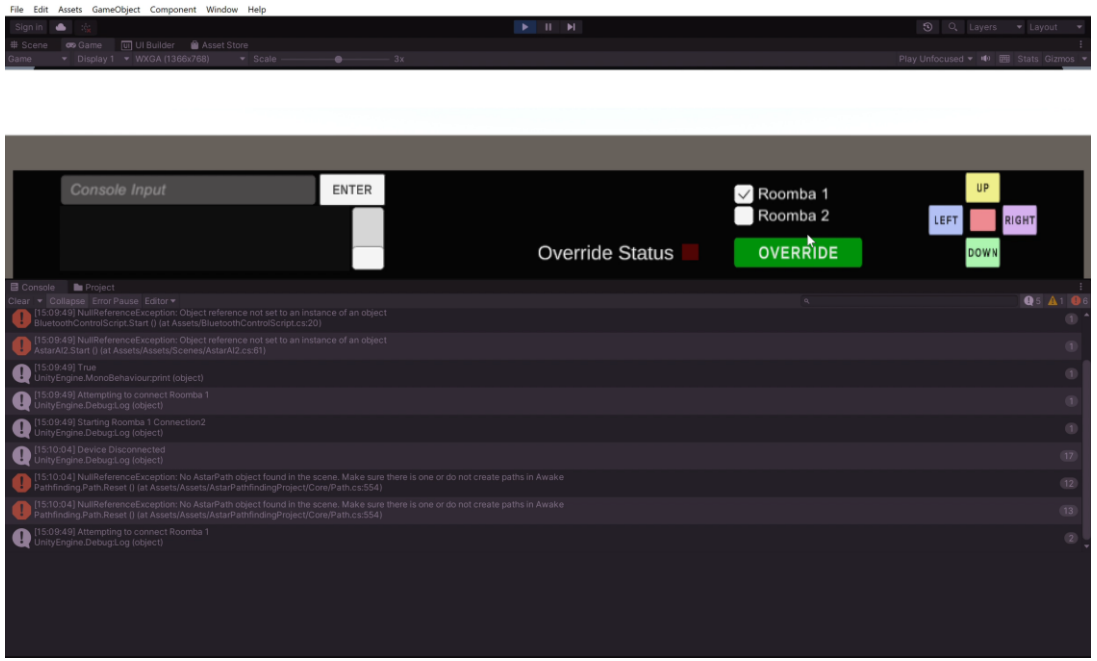
All Team Members

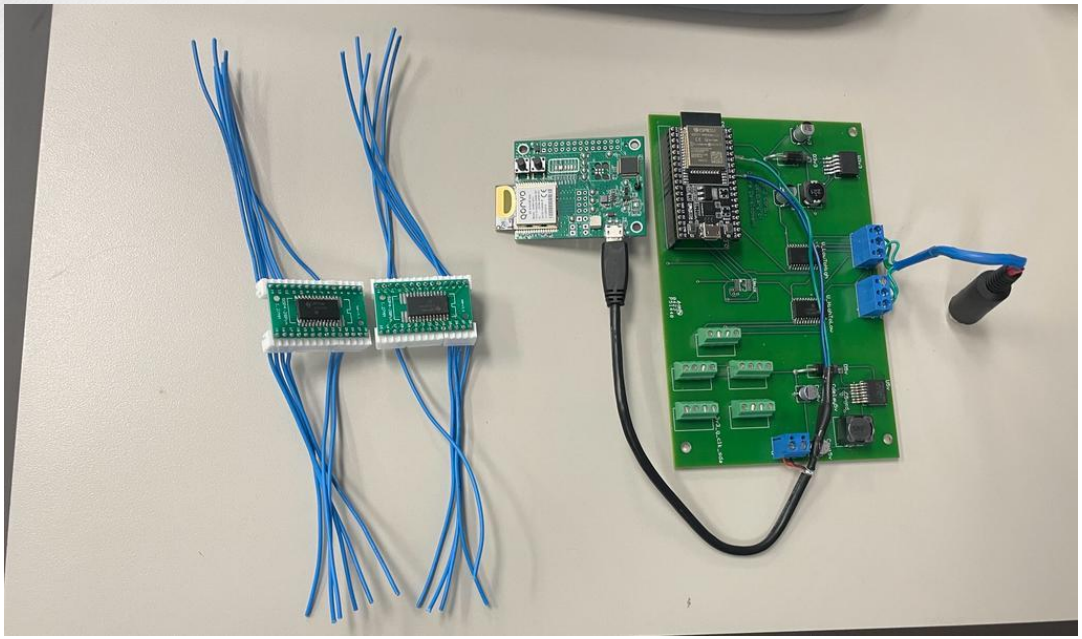
Accomplishments since last update 30 hours each	Ongoing progress/problems and plans until the next presentation
- Roomba follows pathing script commands without need for manual control - Roomba can exit pathfinding control and enter override mode without issue - Cleaning of an empty space can be completed - Implemented bumper sensor obstacle avoidance with computer connection - Smoothed out turn and distance control commands - Added localization code to ESP32 - Added level shifter for UART communication - Designed new faceplate for Roomba with mounting provisions for control board - Added sensor mount locations in coordination with James - Designed enclosure boxes for localization units	- Test pathing calculation with pathfinding script on Roomba - Multiplexer on PCB is having issues, need to make changes to allow for distance sensor obstacle detection - Get faceplate and UWB module enclosures printed. - Test new localization communication method for speed.

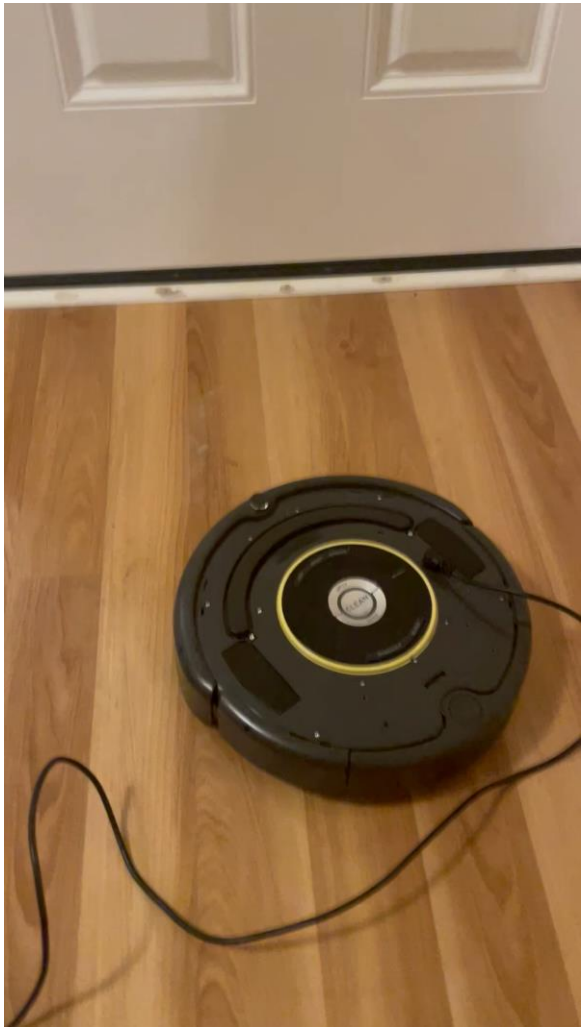
Integrated System

All Team Members

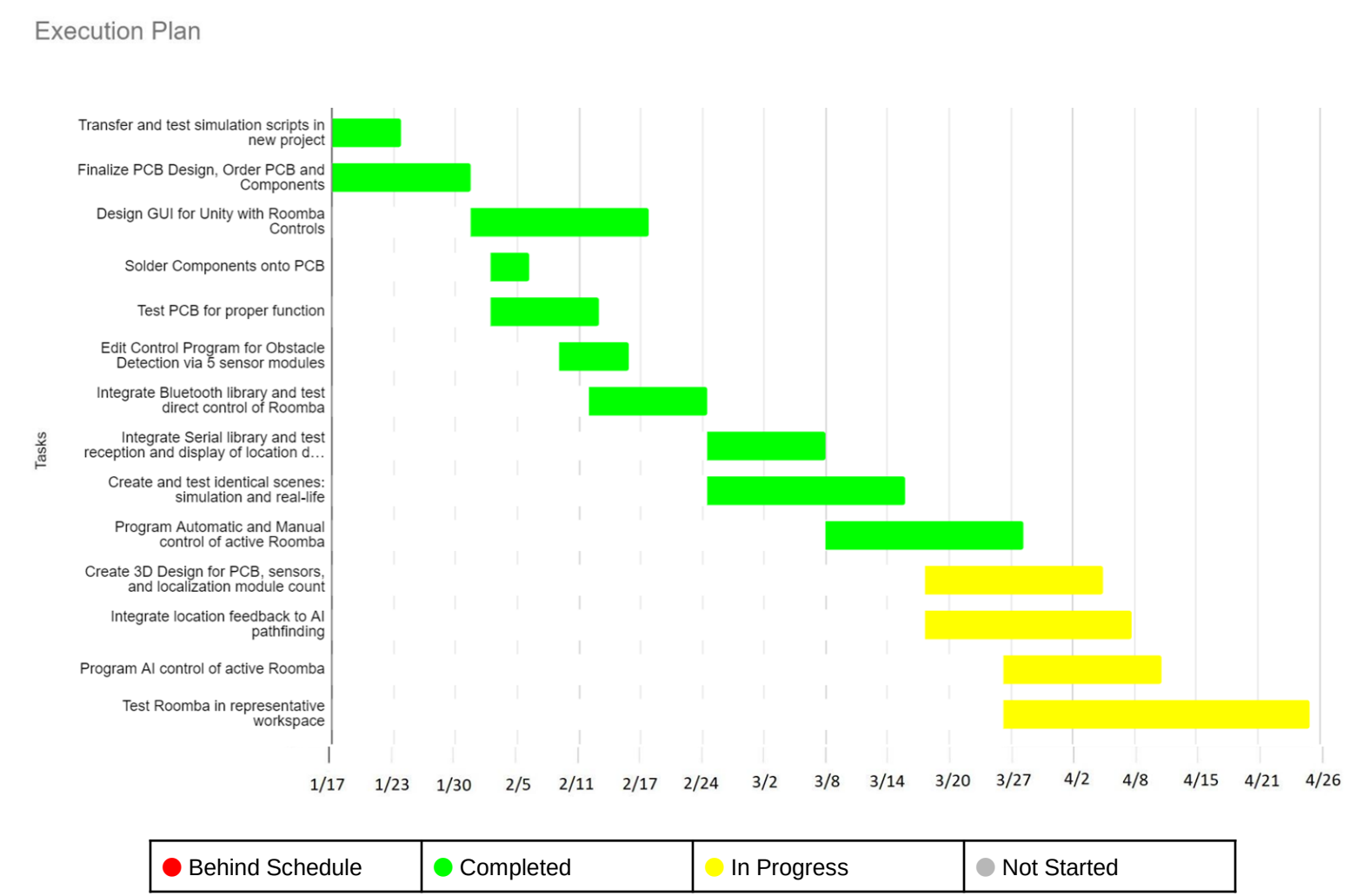








Execution Plan



Dwight Look College of ENGINEERING
TEXAS A&M UNIVERSITY

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 blind spots.	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 blind spots.	Mount adds less than 50% to the height of Roomba and allows for no blind spots.	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!