

Use Case/Solution Approach

Goals:

- Make search and rescue autonomous and distributed
 - No human input
 - Effective coordination between various search devices
- Deliver supplies to someone in need
- Should work in variety of environments, from open space to space with obstacles

Solution Approach:

- A SWARM with small rovers that segment and search the space for a target
- This has changed since our proposal presentation
 - Drones are much more expensive, leading to smaller SWARMS under the same budget
- Open casing to closed casing
 - Added a 3D printed case to house the hardware.

Design Requirements

Use Case Requirements	Associated Design Requirements
Fully autonomous with effective distributed algorithms	Raspberry Pi for algorithms and sensor data Index-based arbitration system Space distribution leads to <20% overlap
Successfully finds target 90% of the time	Boustrophedon Search Pattern Target identification with color detection
Search time decreases with number of vehicles	Space separation continues until each rover has its own space Time to target identification in same environment scales with $1/n$
Bots converge at target	Convergence rate: $>\frac{2}{3}$ of all rovers get to target
Find target in 400sq ft room in 4 minutes with 3 bots	Quick: at least 1 foot per second Good control: Sensors for localization to verify positions

Design Requirements

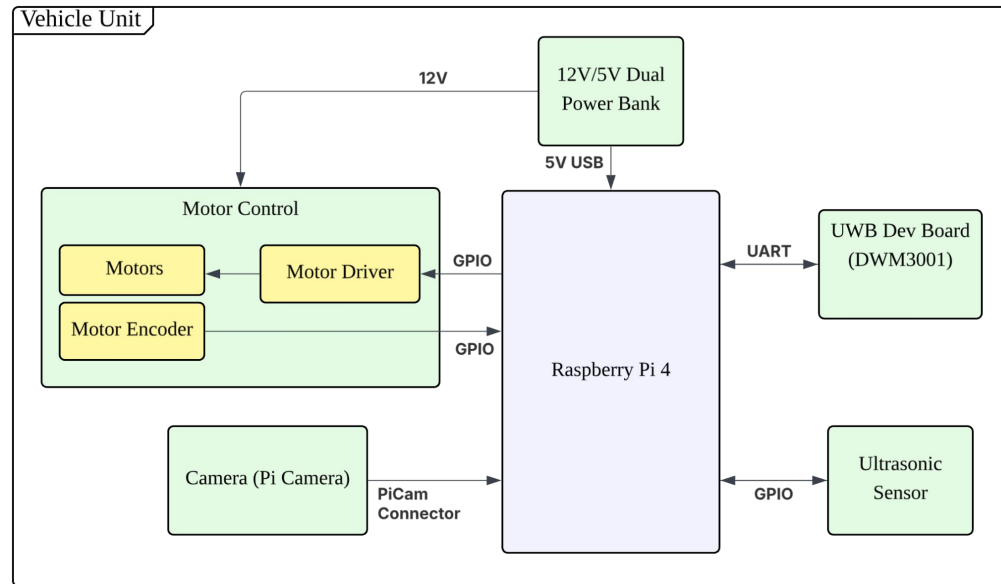
Use Case Requirements	Associated Design Requirements
Low cost per unit	Capstone budget: each rover costs less than \$100
Localization	UWB and motor encoders for tracking position
Sufficiently long battery life	At least 30 minute battery life with continuous movement
No collisions between rovers or with obstacles	Ultrasonic sensors
Communications	UWB and distributed communications algorithms

Solution Approach

- UWB: Communication and localization
- Camera: Target detection
- Motor encoders: PID
- Ultrasonic sensor: Obstacle detection

Recent changes:

- Movement functions in state machines rather than loops
- IR sensors to Ultrasonic sensors



Test, Verification and Validation

Movement Testing:

- Rotations are accurate
- Boustrophedon Path Following at 100%
- Hub and spoke movement at 100%
- Moving straight with occasional drift (<30 degrees):
 - Motor velocity calibration have been inaccurate
 - They still cover the space

Test, Verification and Validation

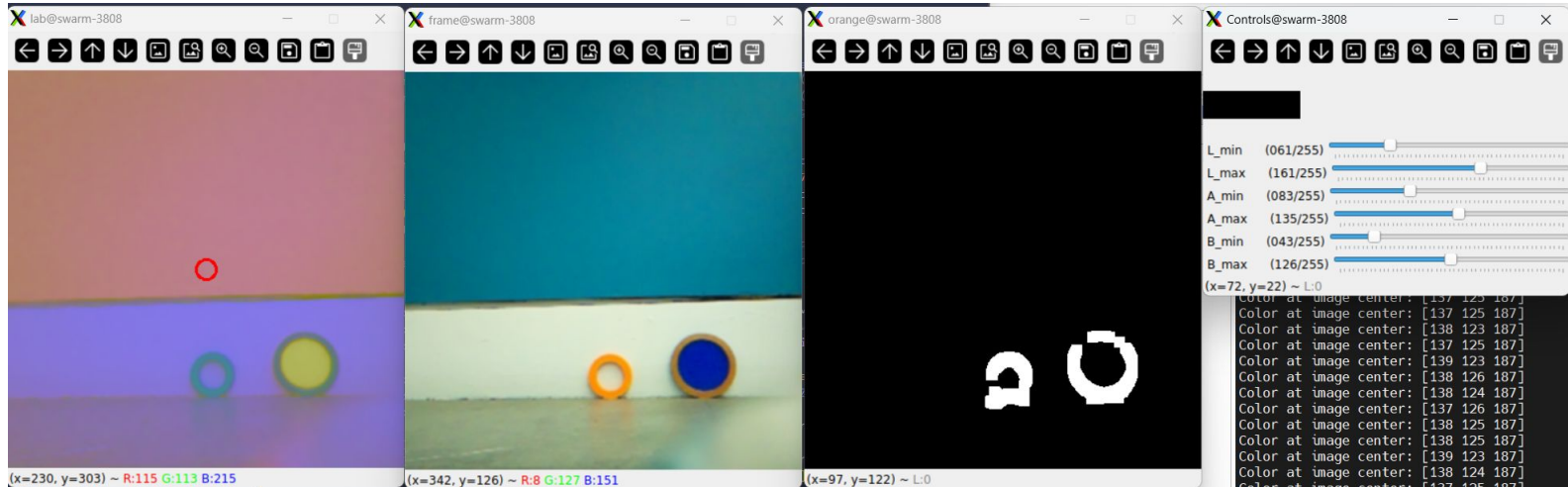
- Operating Systems Scalability
 - The operating system cloning process is now working 100% of the time. We can flash straight to an sd card and it will properly boot straight to our program with the necessary libraries.
- Hardware Integration
 - Changed from open casing to closed casing
 - Wiring problems
 - GPIO malfunction
 - Software integration on hardware
 - Rid of FSM in movement states
- Software Integration
 - Changes in movement states
 - Communication protocol $\Leftrightarrow$ PID tuning & movement algorithms
 - Reduced states machines for readability

Test, Verification and Validation

- Hardware overheating issues
 - Raspberry Pi overheating
 - Bad GPIO pin?
 - Airflow?
 - Ultrasonic Sensor overheating
 - Replacing Ultrasonic sensor
 - Get rid of duct tape
 - 2 of the 3 bots are fully functional, 1 is overheating and rebooting after ~5 minutes, but is otherwise functional

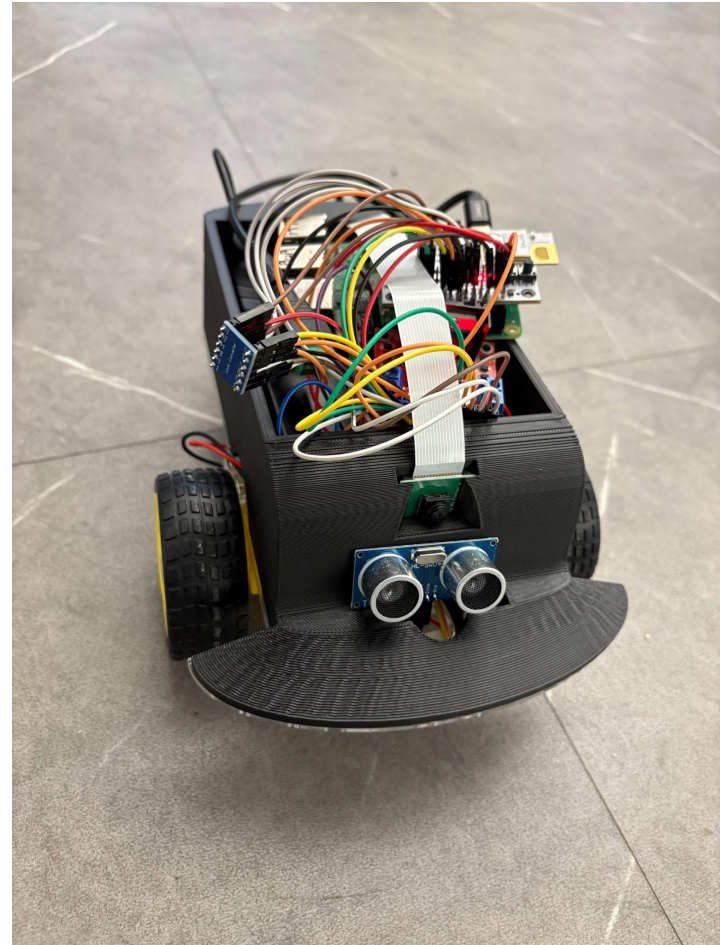
Test, Verification and Validation

Target Detection



Complete Solution

- 3 Autonomous Rovers
- 16x16 search area
- Initialization
 - Coordinated localization movement
 - Moving to each partitioned space
- Boustrophedon Pattern for searching
- Target Detection => Broadcast
- Converge
 - Hotter Colder method



Specifications and Performance

	Target Threshold	Rate Achieved
Communication	95% initial receptions, 100% after corrective retransmission	100% overall
Software Update Period	50ms	50ms
Target Detection	>90% when within 2ft >50% when within 5ft	Not fully tested, but detects in ~90% of frames within 2ft
Movement	1ft/sec	Max speed is much faster than our threshold. >2 ft/sec

