I.G.O.R. PDR

(Intelligent General Order-fulfillment Robot)

Team 20 - LH 27

Adam, Alex, Johnathon, Josh, Victor

Problem Statement

- Delivering objects indoors takes time and resources
- Package sizes and shapes can vary, making them challenging for a robot to pick up
- Dynamic environments with moving humans can make operation of autonomous robots dangerous for both parties

Design Alternative 1 - Starship Technologies

- Utilizes GPS, Camera and IMU; Outdoors only
- Human loaded and unloaded; designed for short distance food delivery (DoorDash)
- Cannot own (per delivery business model)

Design Alternative 2 - Savioke Relay

- Utilizes LiDAR and other sensors for indoor usage
- Human loaded and unloaded; designed for hotel room service
- Large frame necessitates a clear open vertical environment to traverse across

Proposal

- Use LiDAR for map generation of delivery area using SLAM (Simultaneous Localization and mapping) techniques
- Request/Schedule/Queue deliveries through phone application
- Create a system to autonomously load and unload packages, allowing deliveries to be picked up at convenience as well as enable deliveries to be queued

Package Specifications

- Package fits in our custom-built box (9″ x 11.5″)
- Package weighs no more than three pounds

Lifting Specifications

- IGOR is able to:
  - Pick up the package successfully 90% of attempts
  - Lift the loaded custom box at least 1 inch off the ground

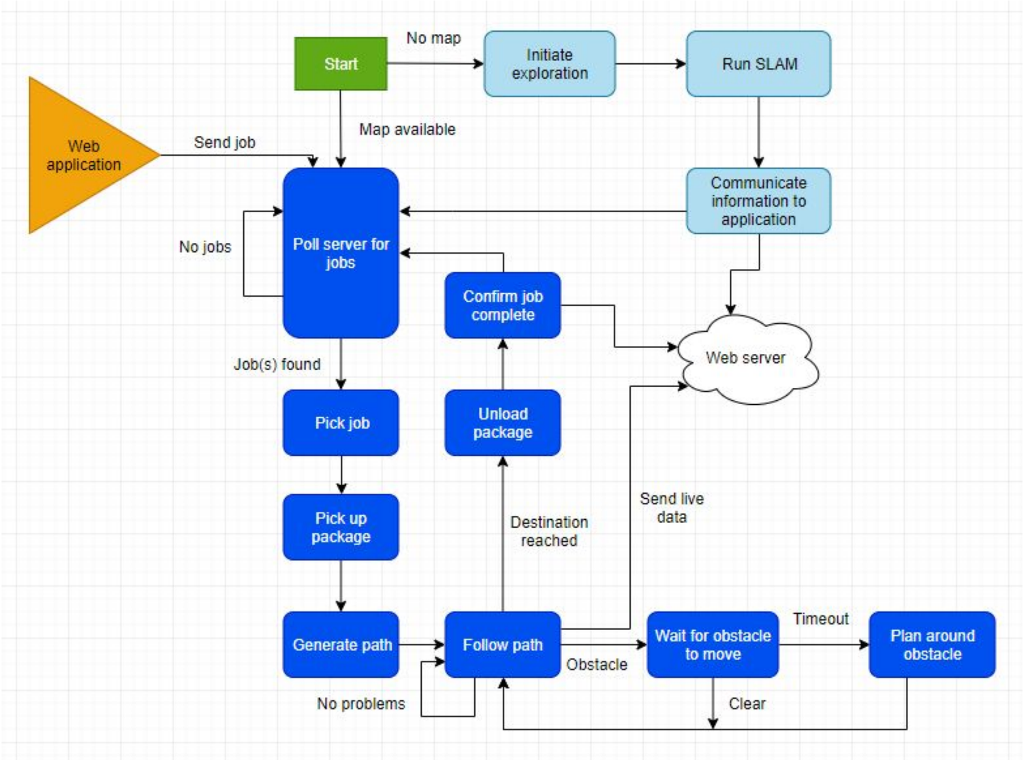
Moving Specifications

- IGOR:
  - Can move between 1 to 3 mph while on smooth floors (e.g. tile) or rough floors (e.g. carpet), with inclines/declines that do not exceed a slope ratio of 1:12
  - Is able plan a path between source and destination points
    - Able to deliver the package within 1 meter of the marked destination within 90% of all attempts, given successful pickup
  - Battery capacity will allow for at least 30 minutes of operation time without charging

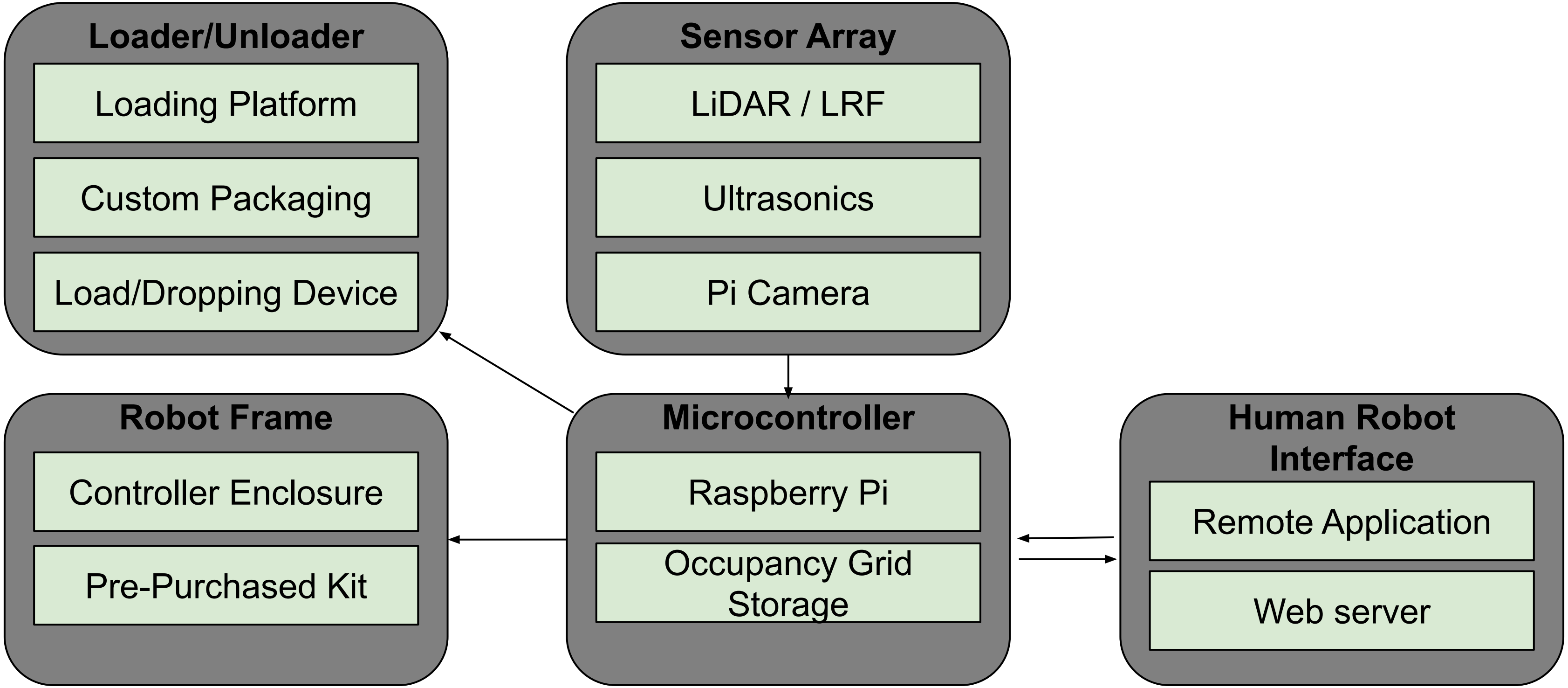
Moving Specifications (cont.)

- IGOR:
  - Can estimate its position within 0.5 meters 90% of the time, given its starting position
  - Can avoid collisions with non-adversarial obstacles 90% of the time
    - These avoidable adversaries do not include human factors, such as intentionally or accidentally being kicked, run into, or fallen on

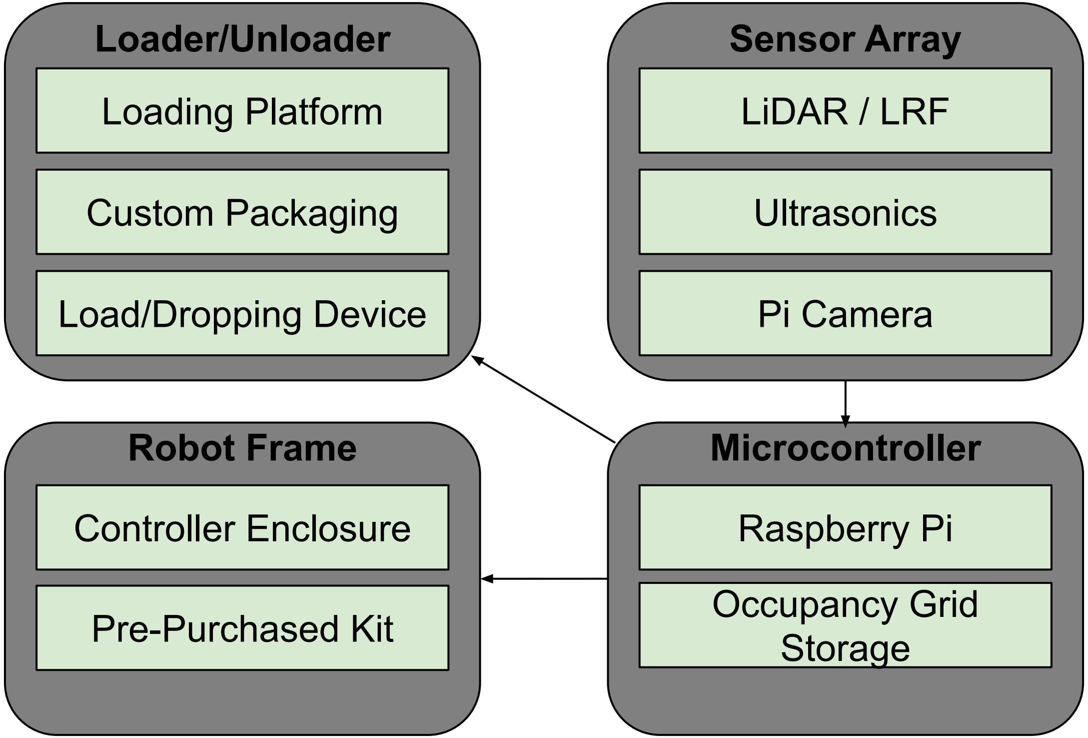
State Machine



Block Diagram



Block Diagram



Subsystem: Robot Frame

- Chassis/Frame, Motors and Wheels
- Mounts for External Components
- Microcontroller and PCB Enclosure
- Allows for traversal in a single floor

Subsystem: Loader/Unloader

- Chain based loader and unloader storage system to carry payloads
- Will be controlled by processed input from camera and April Tags

Chain for package loading

Subsystem: Sensor Array

- Camera for precise loading and unloading
- Laser ranging sensors for mapping
- Proximity sensors for collision avoidance

Raspberry Pi Camera

LiDAR

Ultrasonic Distance Sensor

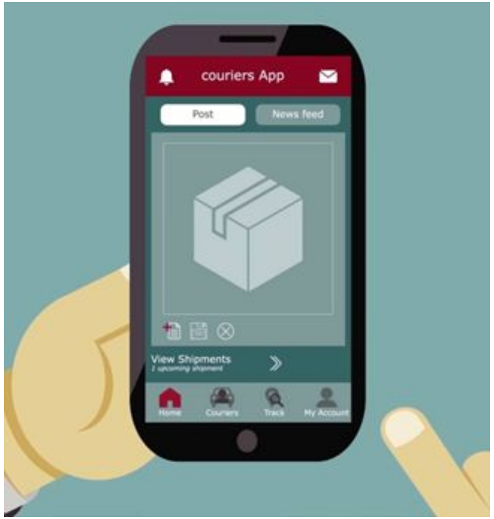
Subsystem: Microcontroller

- Microcontroller to be in charge of creating SLAM map and communicating with user interfaces

- Will employ occupancy grid for map storage

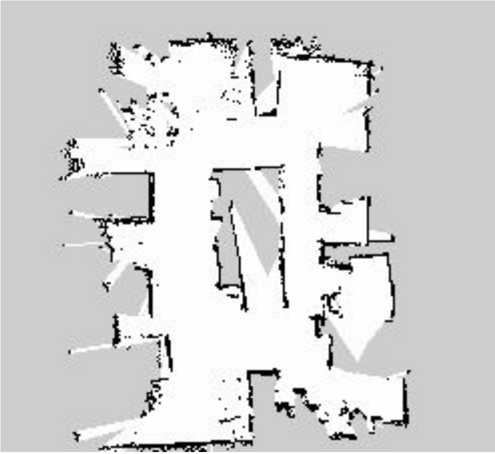
Subsystem: Human Robot Interface

- Human robot interface to send and receive delivery requests in addition to delivery confirmation via an app on computer or phone.
- Signal to be sent upon delivery to ensure drop and retrieval has been done successfully



Software Integration

- Use LiDAR data to construct a mapping of any floor using SLAM within a Raspberry Pi Microcontroller
- Close proximity collision detection will be performed by code executed on a custom PCB to ensure safe operation



Hardware Component

PCB to interpret ultrasonic sensor data to detect if collision will occur.

If potential collision is found, PCB will send an interrupt to the rest of the system

Budget

Robot Chassis/Motors	$170
LiDAR/Range Finder	$150
PCB	$60
Battery	$50
Reserve	$70
Total	$500

Risks

- Mechanical:
  - Picking up objects is *hard*
  - Debugging complicated physical systems is a slow process
  - A well-made physical system is easier to program for - a poorly-made physical system makes the rest of the system harder to work with

Risks

- Software:
  - 2D SLAM doesn't inform the robot about anything not at the sensor's height
  - Lots of processing needs to happen:
    - SLAM
    - AprilTag recognition
    - Sensor readings
    - Communication with base station

Presentation for MDR

- At MDR, our robot should be able to:
  - Autonomously load and unload packages when pre-aligned with the package
  - Plan a path with a given coordinate, floor plan, initial position ground truth
  - Receive directives to pick up and deliver packages at the correct locations
  - Close-range collision detection via a breadboard and Raspberry Pi GPIOs

Presentation for FDR

- Our robot will be able to:
  - Autonomously load and unload packages
  - Plan a path and navigate from source to destination
  - Have a functional interface via an application, such that the user can select where on a map the package should be delivered to

FDR Stretch Goals:

- If time permits:
  - Use SLAM to generate and map a floor as an alternative to being given an existing floor plan of a building
  - Implement collision detection to prevent the robot from running into moving people

Presentation for Demo Day

- We will construct a maze
- The maze will be a model floor plan
  - The robot will be given a map of the maze
  - The robot will receive jobs and deliver packages

Breakdown of Activities

Adam	Alex	Johnathan	Josh	Victor
• Collection Device • Chassis construction and design • Programming movement controls	• Sensor testing and interface • Getting SLAM to create an accurate map	• Interfacing April Tags and Camera • Possibly web server/application • Research on utilizing SLAM	• PCB Design • Creating collision detection • Programming the movement controls for the robot	• Creating interface for issued commands • Programming robot to drive based on map