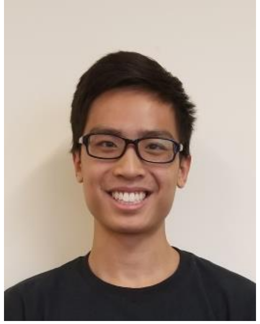
Preliminary Design Review

Team 16
October 22, 2018

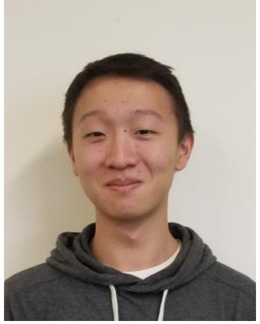
Mapper

Kelvin Nguyen
ME



Marcus Le
EE

Bryan Martel
CSE



Derek Sun
CSE

Background and Motivation

- 44 significant building collapses since 2010
  - 1975 casualties
  - 3459 injuries

- 40-50 cave related incidents per year
  - ~10% of incidents are fatal

https://en.wikipedia.org/wiki/List_of_structural_failures_and_collapses
https://www.outsideonline.com/1903801/exploring-caving-accidents-deaths-and-rescues-united-states

Background and Motivation

- Safety and knowledge are very important aspects of rescue missions
- Not fully understanding the environment and situation can lead to unnecessary risks and dangers

Examples

- Cave rescue
  - Trapped or lost explorers
- Urban search and rescue
  - Collapsed buildings

Goal

- Reduce possible risks or dangers that are associated with traversing through unknown environments
- Enable easier navigation through the field and aid in figuring out the best method of approach
- Improve efficiency of rescue teams in unknown environments

Method of Resolution

- A robot that utilizes LIDAR sensors to remotely navigate around the surrounding environment and produce a 3D layout of an indoor area

Requirements Analysis: Specifications

- Speed of up to 3mph
- Effective detection range of 15ft
- Approximately 12 pounds
- Approximately 1 hour of battery life
- Elevation 1-6 feet
- Durable

Design Alternatives

Google Cartographer

- Backpack mounted
- LIDAR mapping
- Inertial measurement unit

UCSD 3D RGB Mapping Robot

- Infrared and RGB camera
- Two-wheeled
- Able to climb stairs

Design Alternatives

Why we chose LIDAR instead of RGB camera for measurements

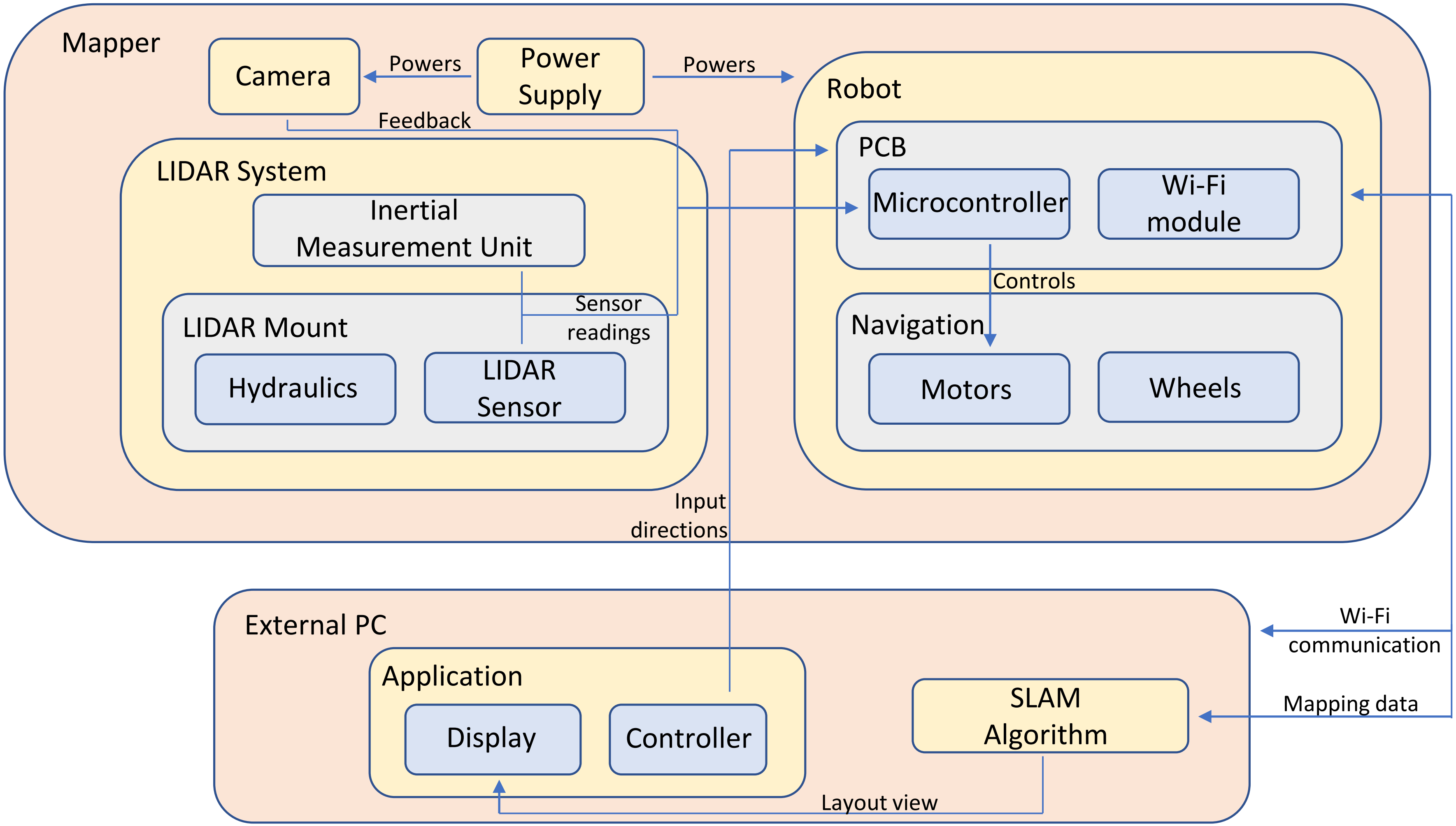
- Higher precision
- Higher range
- Does not require light
- Less computationally intensive
- RGB camera requires movement to estimate distance, increases computational complexity for localization

Our Design

Requirements Analysis: Inputs and Outputs

- Input
  - LIDAR sensor data
  - Inertial measurement unit data
  - Camera data
  - User navigation control
- Output
  - Live video feed
  - Map data

Block Diagram



UMassAmherst

LIDAR Sensors

- Rapid pulses of laser light sent out
- Measure time each pulse takes to bounce back
- Commonly used for police speed guns and mapping
- Data is output as a coordinate of distance and heading
- Will be used in our project to generate point cloud of area

Data Type	Unit	Description
Distance	mm	Current measured distance value between the rotating core of the RPLIDAR and the sampling point
Heading	degree	Current heading angle of the measurement
Start Flag	(Bool)	Flag of a new scan
Checksum		The Checksum of RPLIDAR return data

Figure 1-4 The RPLIDAR Sample Point Data Information

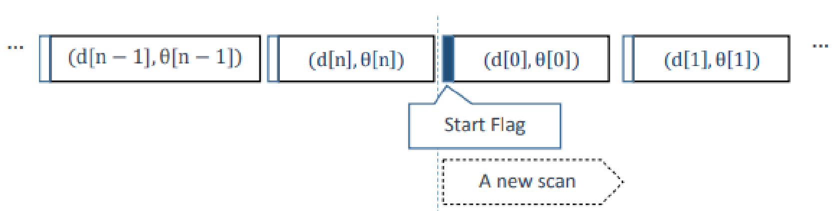


Figure 1-5 The RPLIDAR Sample Point Data Frames

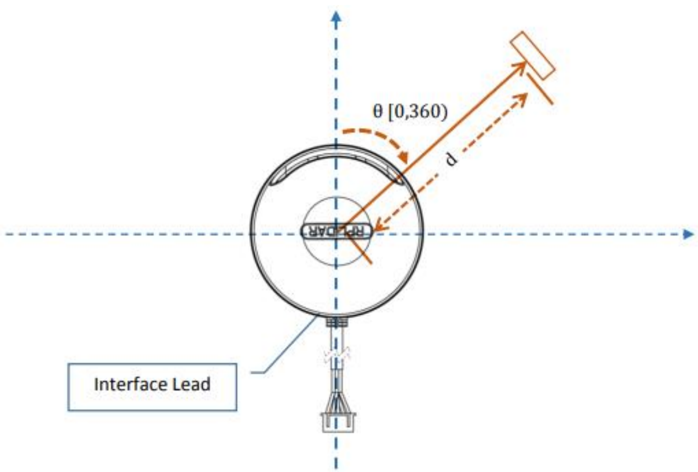
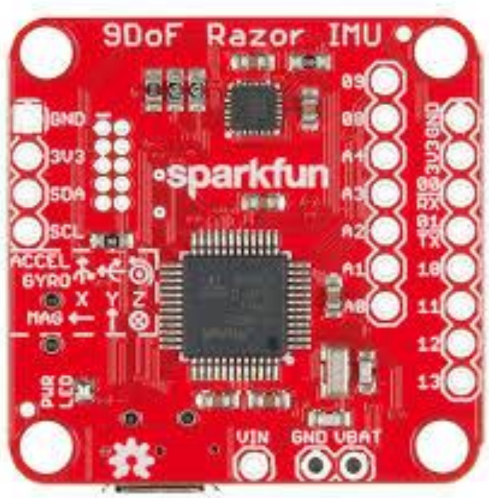


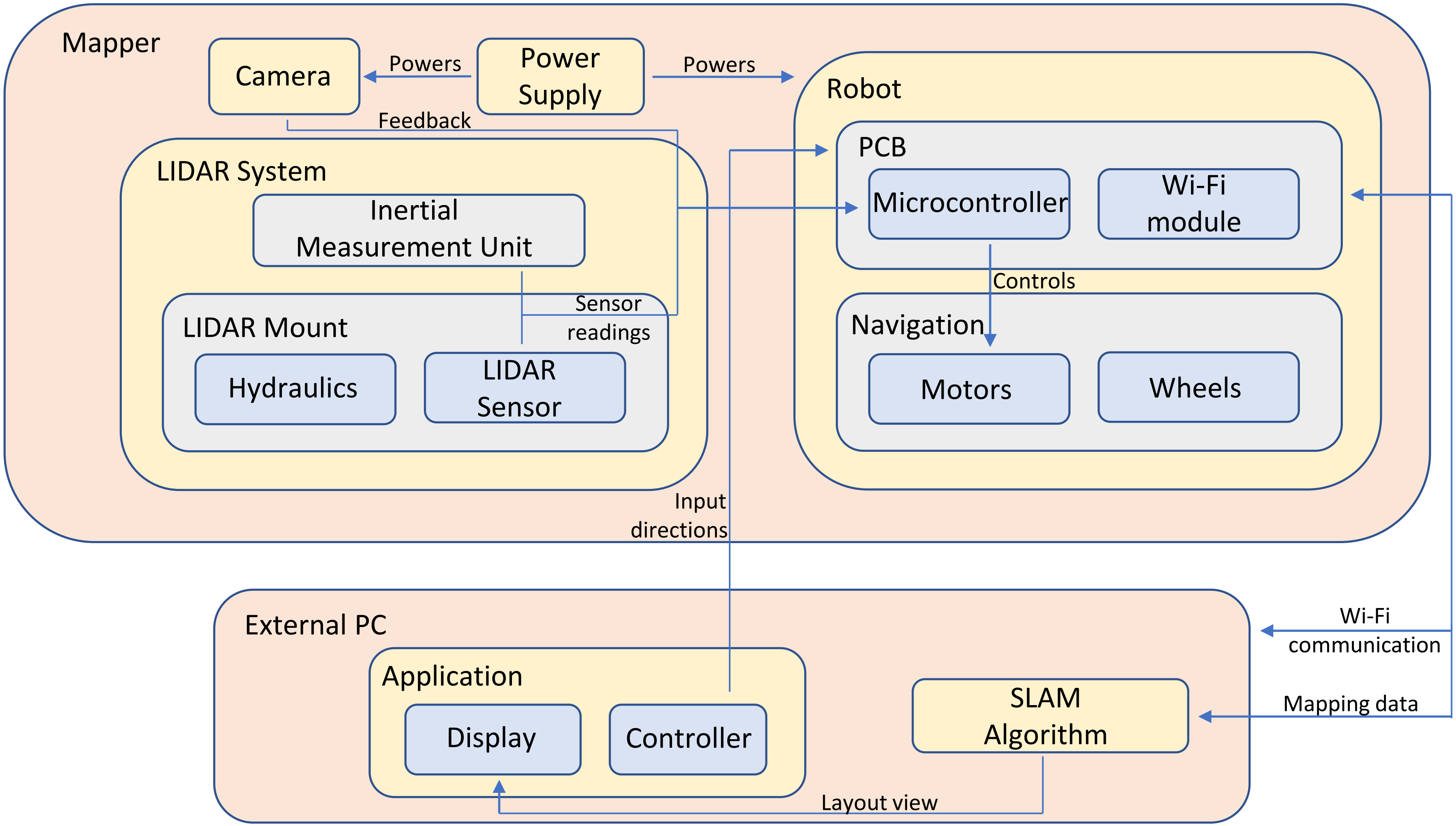
Figure 2-5 RPLIDAR Scanning Data Coordinate System Definition

Inertial Measurement Unit

- Utilizes three 3-axis sensors
  - Accelerometer
  - Gyroscope
  - Magnetometer
- Why use an IMU with a LIDAR sensor?
  - Must understand orientation in order to understand position of data



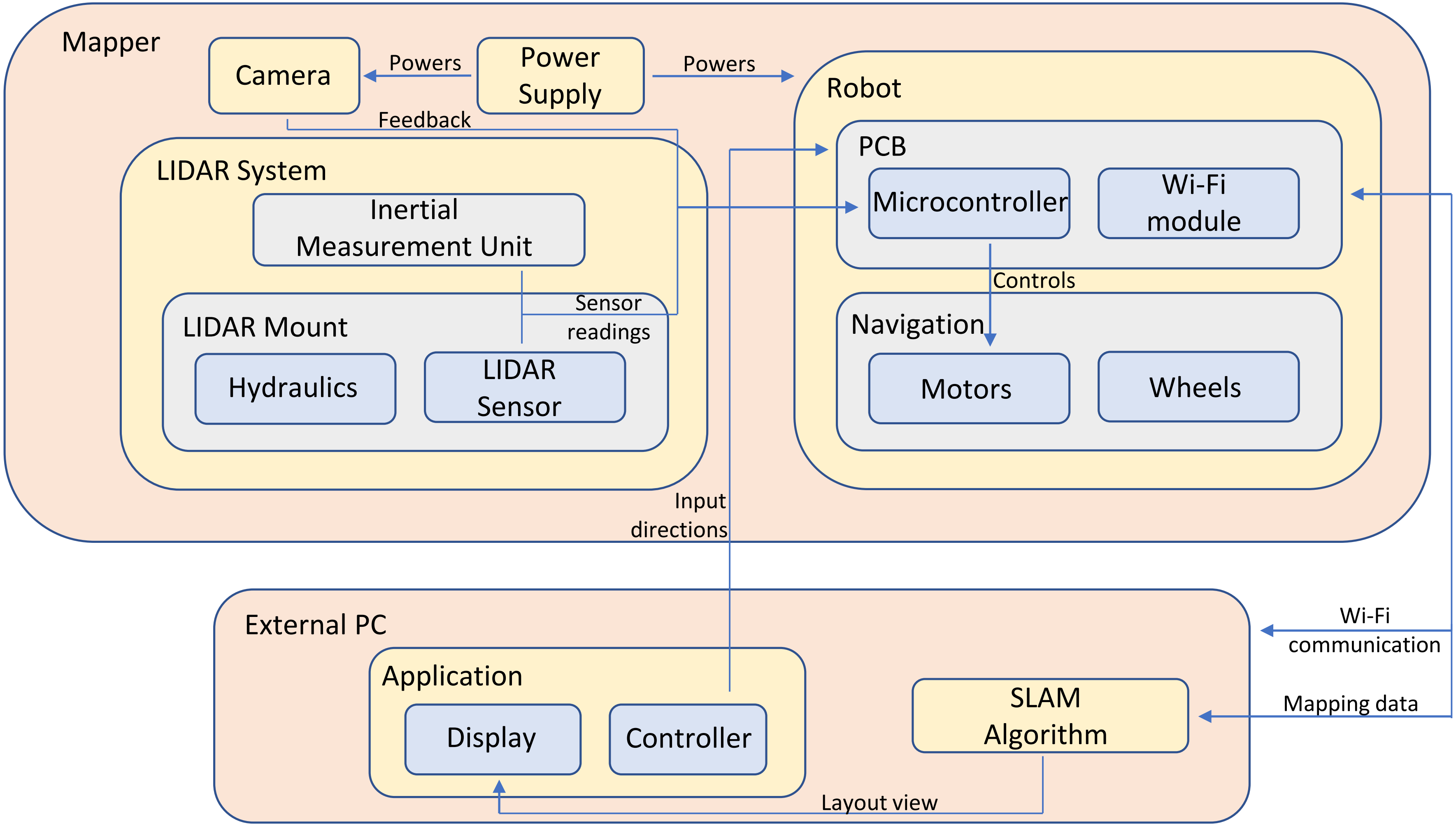
Block Diagram



Robot

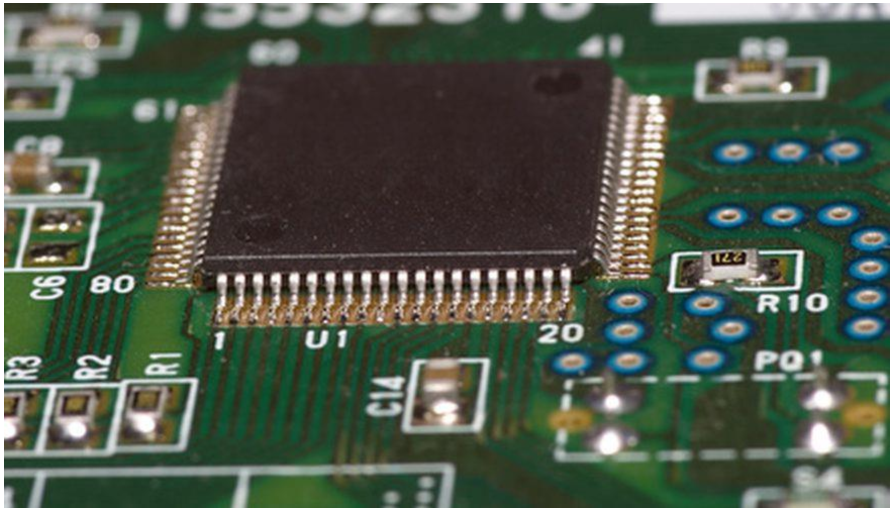
- Requirements
  - House camera and sensor
  - Maneuver LIDAR sensor
  - Integrate our PCB with the components of the Roomba
  - Modifiable
  - Portable
  - Robust
  - Stable

Block Diagram

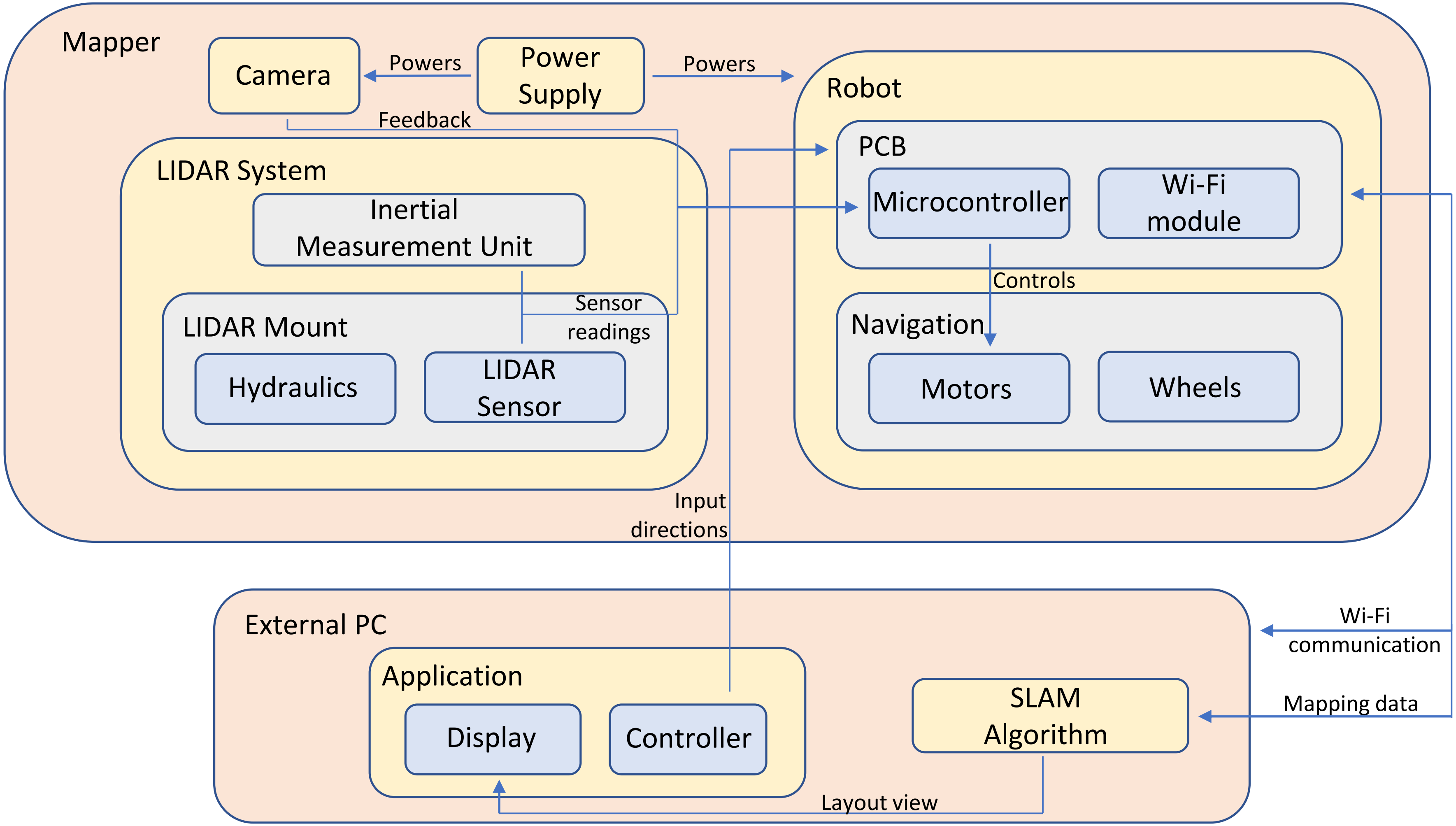


Printed Circuit Board

- Replaces the default board inside the Roomba
- Provides power to the motors
- Houses Wi-Fi module used for communication with external PC
- Microprocessor receives sensor outputs and relays the data to external PC
- Receives inputs from the controller and navigates the Roomba accordingly



Block Diagram



External PC

- Requirements
  - Communicate with robot through Wi-Fi
  - Transmit navigation instructions to robot
  - Process data collected by the robot and run SLAM algorithm
  - Display live video feed
  - Display map model generated from LIDAR point cloud

Simultaneous Localization and Mapping (SLAM)

Essential techniques

- Landmarking
  - Identifies distinct points to relate the same object from different locations
- Re-localization
  - Corrects the robot's positioning using updated map points
- Loop closure
  - Detects previously visited locations

- How this applies to Mapper

- Plan on adapting open-source Google Cartographer library and tailoring it for our needs

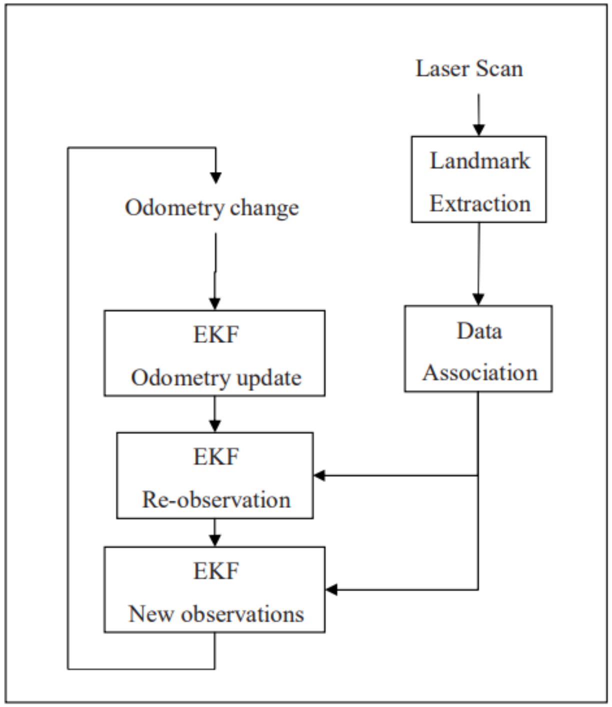


Figure 1 Overview of the SLAM process

Budget

- 360° LIDAR sensor $300
- Pistons/Housing $120
- IMU $30
- Camera $20
- Wi-Fi module $20
- Power supply $10
- Roomba (from M5) $0
- External PC (owned) $0

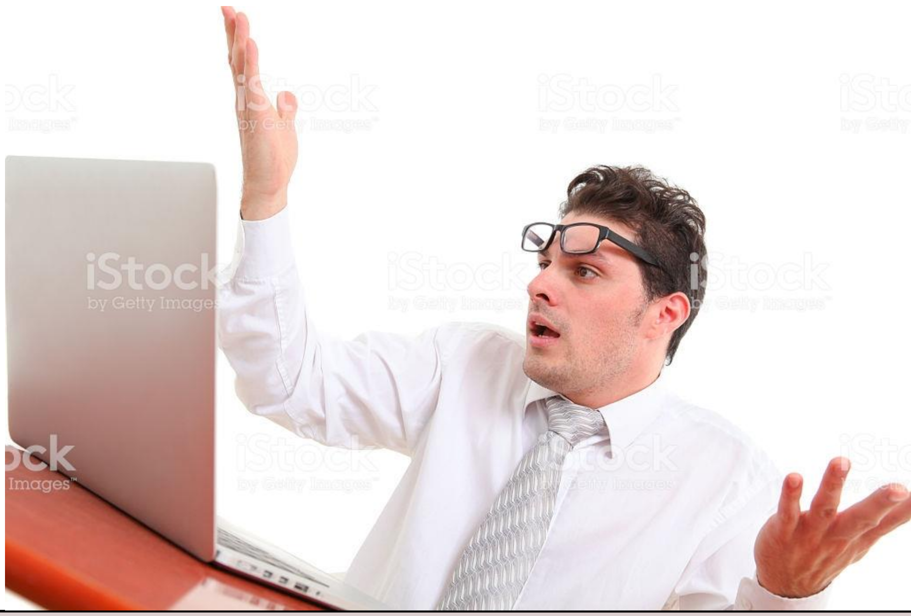
Total: $500

Responsibilities

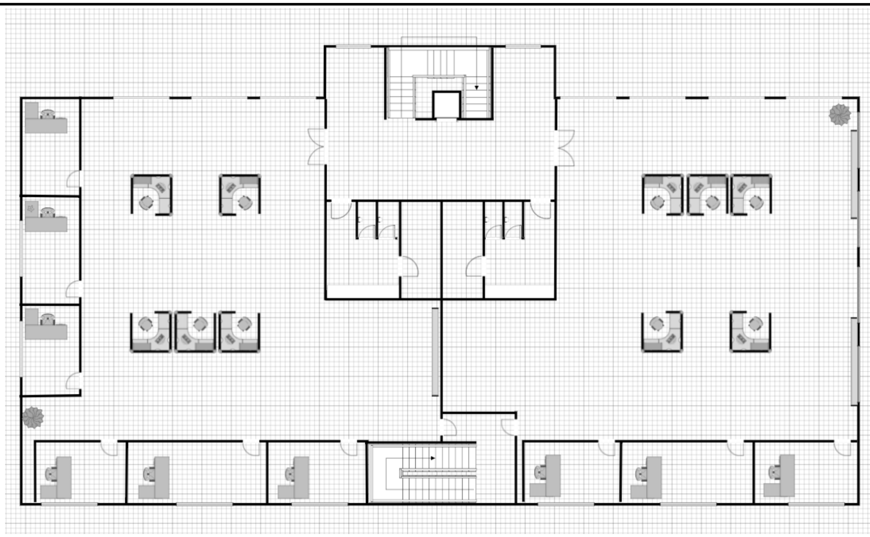
- Kelvin (ME)
  - Robot modification
  - LIDAR sensor elevation
  - Connectivity between motors and PCB
- Marcus (EE)
  - Programming of robot
  - Powering the system
  - Connectivity between PCB and external PC
- Derek (CSE) & Bryan (CSE)
  - LIDAR SLAM implementation
  - Application development

Roadblocks/Challenges

1. SLAM algorithm
2. Robot localization estimates
3. System Connectivity
4. Application development
5. Modification of Roomba
6. Stabilization of LIDAR sensor



What we plan to bring to MDR



Functioning LIDAR

- 2D mapping
- Stationary or manually moved sensor to map the layout of a floor

Responsibilities

- Kelvin (ME)
  - Remodeling the Roomba and supply power to LIDAR sensors
- Marcus (EE)
  - Program microcontroller and early stage application development
- Derek (CSE) & Bryan (CSE)
  - SLAM programming to create a map from LIDAR point cloud

What we plan to bring to FPR and Demo Day

FPR

- Live demonstration of Mapper capabilities

Demo Day

- Mapper on display
- Video that shows Mapper fabricating 3D model of a room
  - Perspective of robot
  - Current map that is being created
  - Tracker that shows where the robot is relative to the room

Questions?