Midway Design Review

Team 16
December 6, 2018

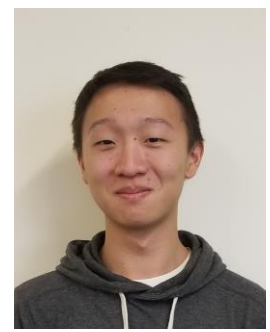
Advisor: Professor Ganz

Mapper

Kelvin Nguyen
ME

Marcus Le
EE

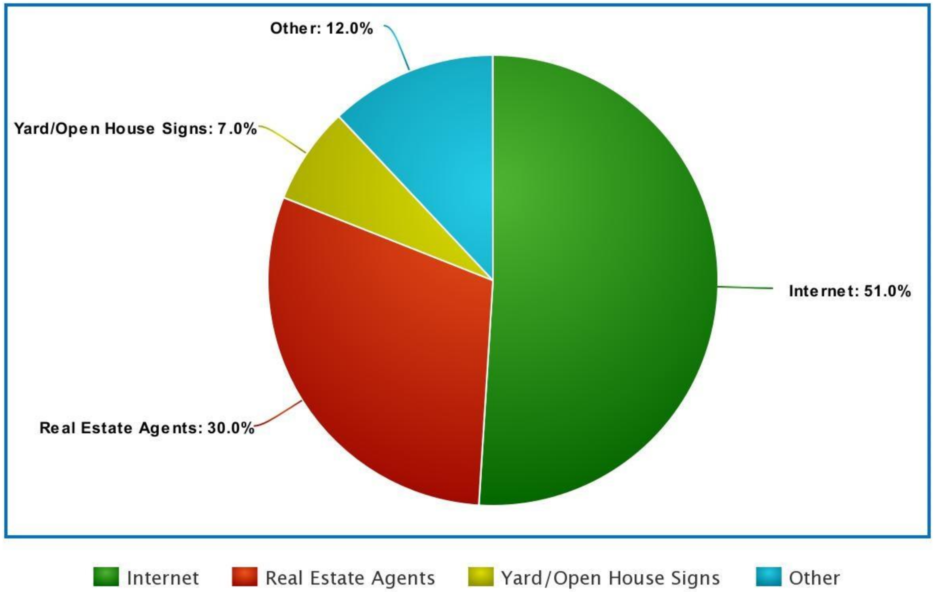
Bryan Martel
CSE



Derek Sun
CSE

Background and Motivation

- In 2017, homeowners found their new houses through:
  - Internet - 51%
  - Real Estate Agents - 30%
  - Yard/Open House signs - 7%
  - Other - 12%



https://www.nar.realtor/research-and-statistics/quick-real-estate-statistics

Background and Motivation

- Over 1/3 of all homebuyers are below the age of 36
- This age group tends to be more technologically savvy
- This leads to the idea that less and less people are going out in active search of houses, instead resorting to the internet to complete their search

https://www.nar.realtor/sites/default/files/reports/2017/2017-real-estate-in-a-digital-age-03-10-2017.pdf

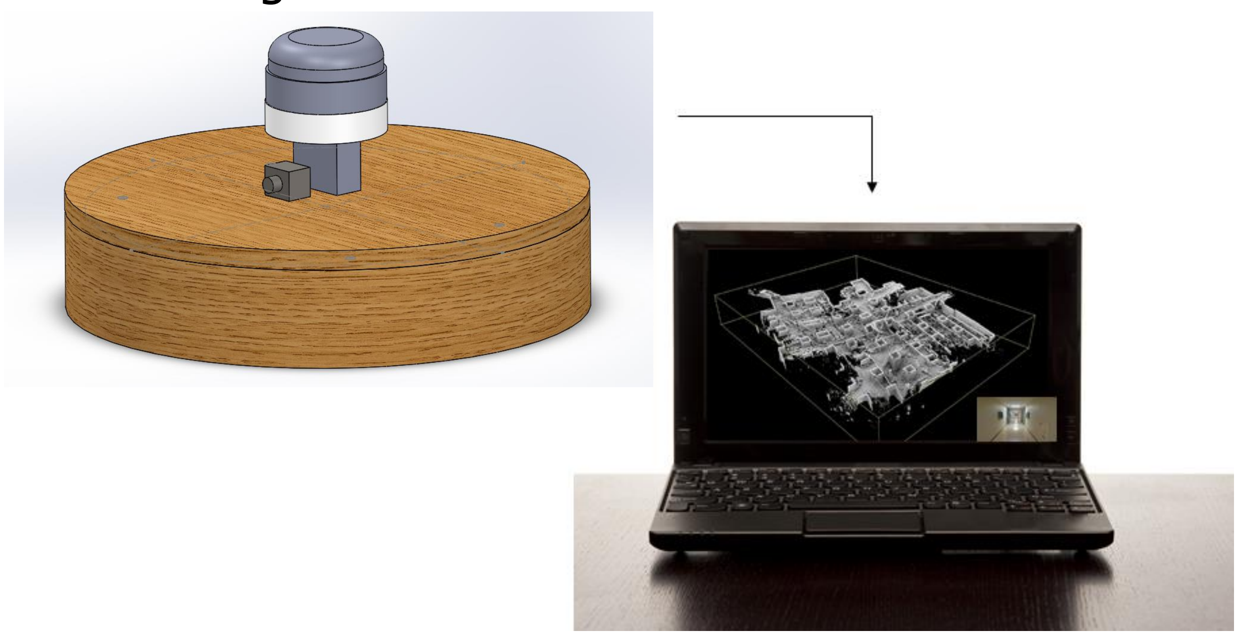
Goal

- Provide homeowners or real estate agents with the ability to post an updated model of the interior of their house

- Potential integration with virtual reality tours
  - Similar to an open house
  - Cater toward the younger, more technologically adept generation that will inevitably dominate the future real estate market

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
- A camera mounted on the robot will allow for live video feed to assist in user navigation



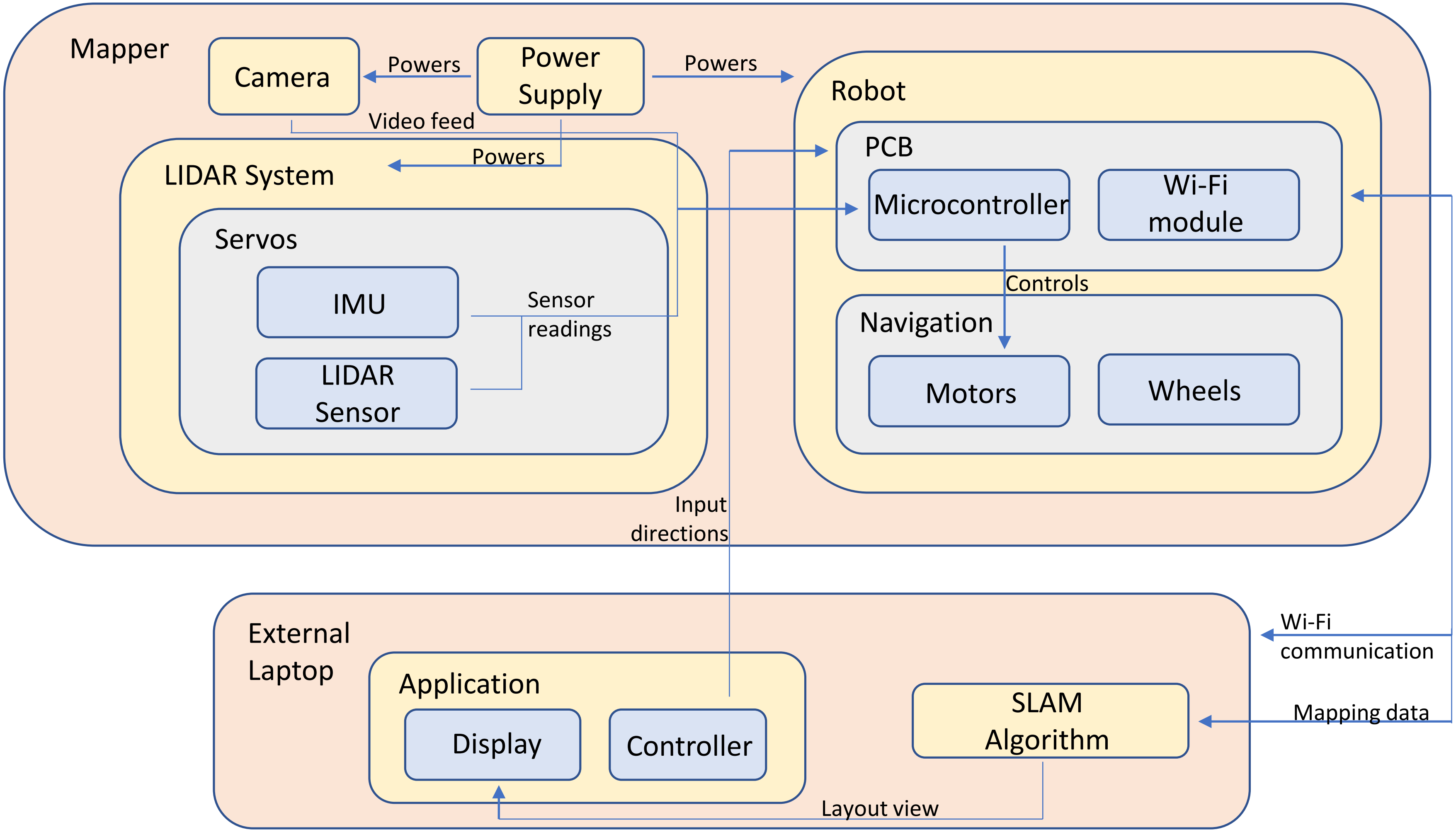
Requirements Analysis: Specifications

- Speed of up to 3mph
- Effective detection range of 15ft
- Approximately 8 pounds
- Approximately 2 hours of battery life
- Durable enough to withstand minor collisions

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



Proposed MDR Deliverables and Responsibilities

- Functioning LIDAR sensor and IMU
- 2D SLAM with manually moved sensor
  - Simulated data input for wheel movement
  - Live data from LIDAR and IMU
- Robot
  - Build housing for LIDAR sensor and prepare for it to be mounted
  - Arduino/PCB able to send navigation instructions to Roomba motors
- Basic lift
  - Scissor-lift structure for elevation of sensor

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

Design Changes

Roomba → Custom robot

- Build our own robot with a custom chassis, instead of using Roomba as our robot
- Roomba chassis does not have enough room to implement the LIDAR sensor and IMU setup we had envisioned

Basic lift → Pan and tilt

- Pan and tilt, instead of using a basic lift for elevation
- Decided that pan and tilt would be the better alternative for implementing 3D SLAM
  - Eliminates stability problems that occurred in lift
  - Simplifies the wiring of the components

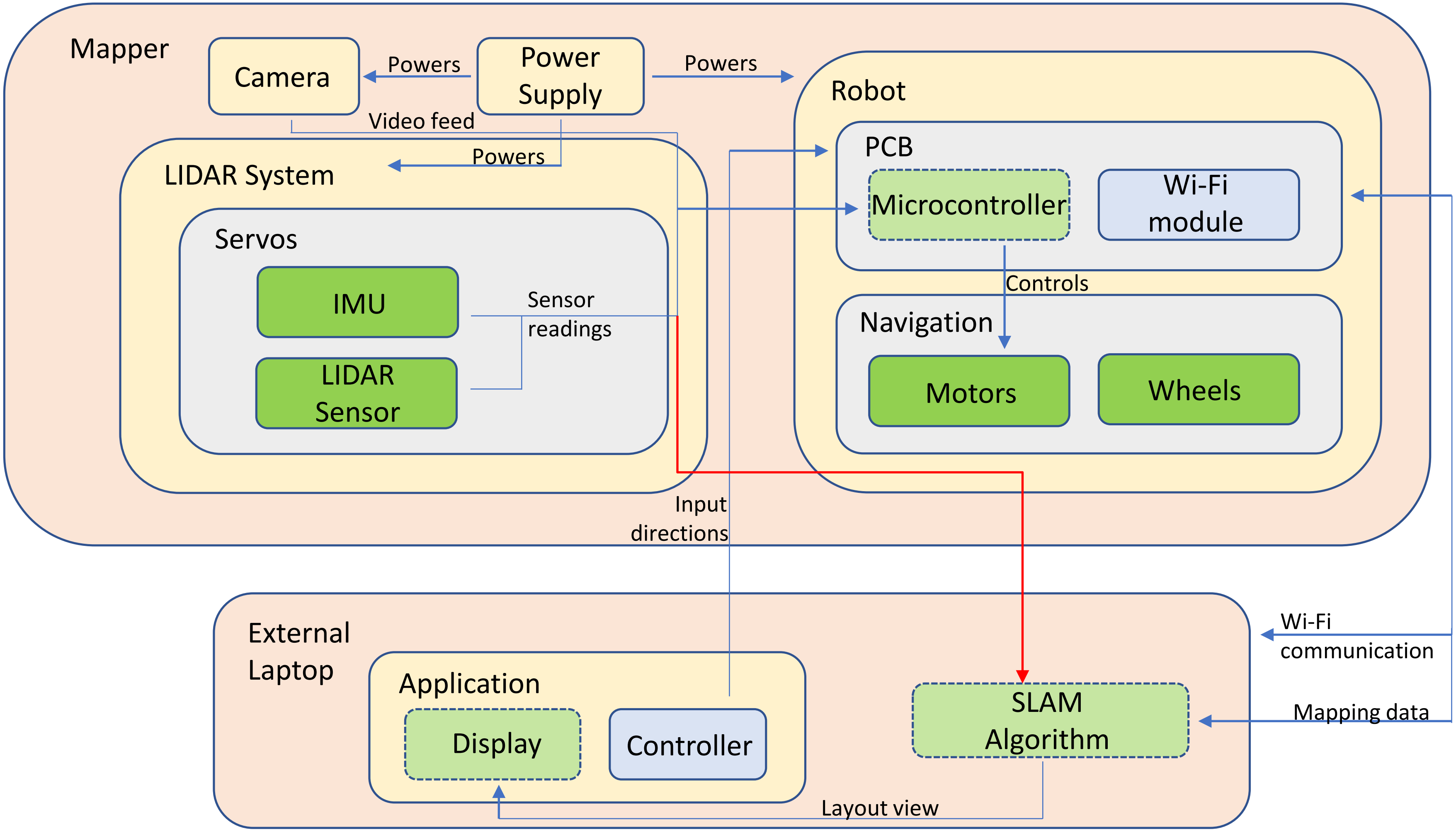
Actual MDR Deliverables and Responsibilities

- ✓ Functioning LIDAR sensor and IMU
- ✓ 2D SLAM with manually moved sensor
  - Simulated data input for wheel movement
  - Live data from LIDAR and IMU
- ✓ Robot
  - Build housing for LIDAR sensor and prepare for it to be mounted
  - Arduino/PCB able to send navigation instructions to Roomba motors
- ~~Basic lift~~
  - ~~Scissor-lift structure for elevation of sensor~~

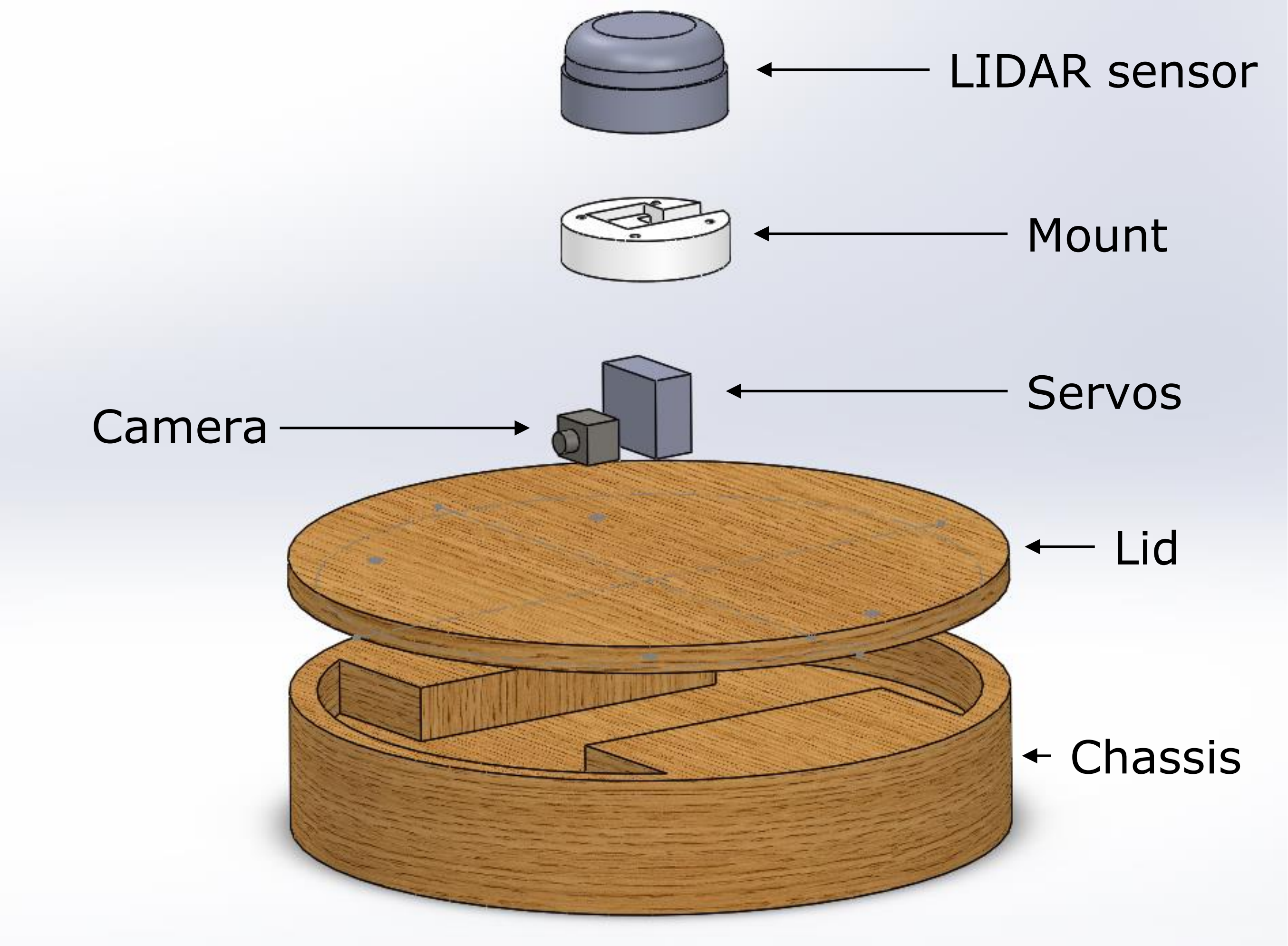
Responsibilities

- Kelvin (ME)
  - Robot chassis and LIDAR mount design/production
- Marcus (EE)
  - Roomba motor integration with Arduino controller
- Derek (CSE) & Bryan (CSE)
  - SLAM programming to create a map from LIDAR point cloud

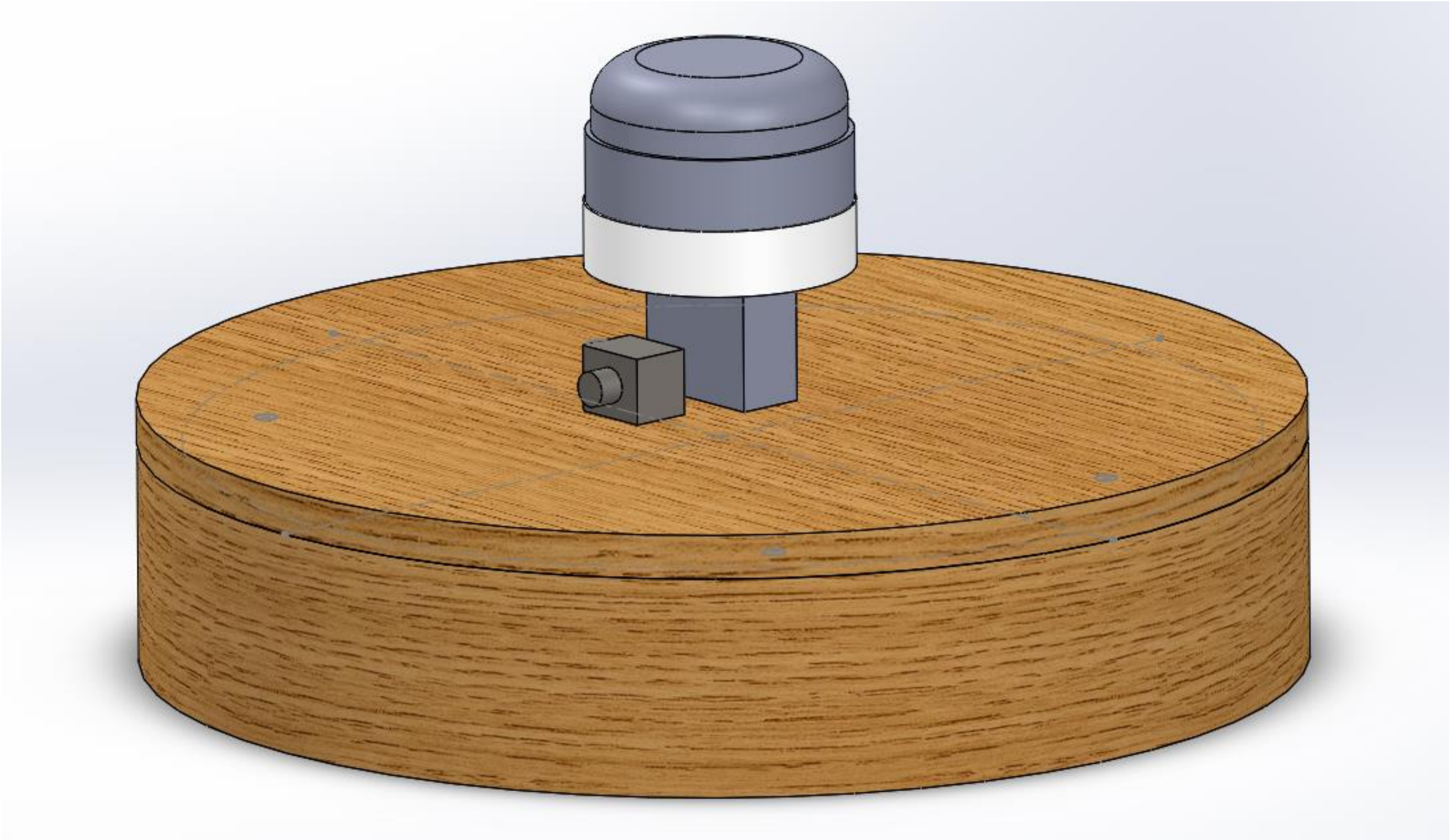
Block Diagram



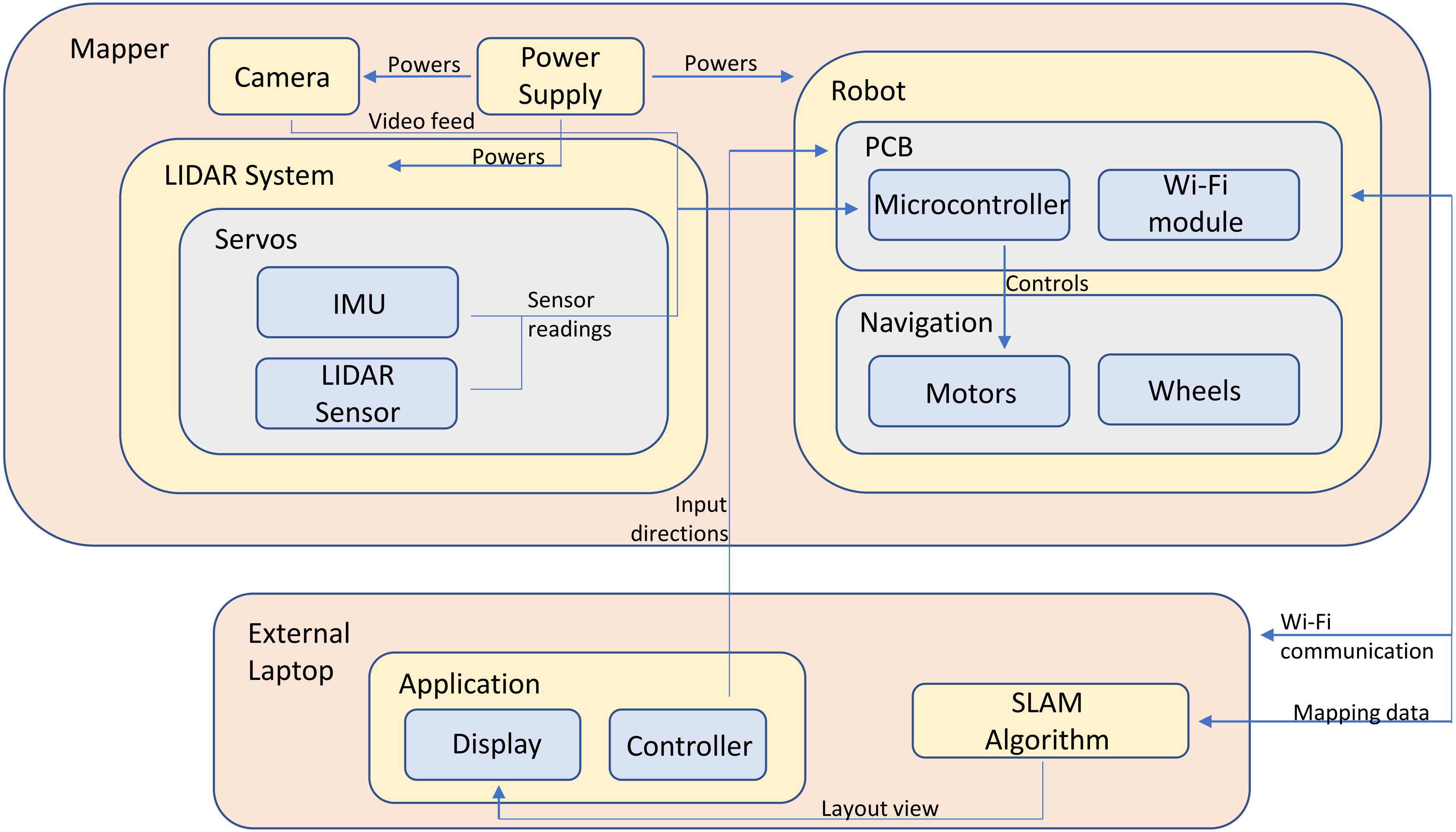
Our Product



Our Product



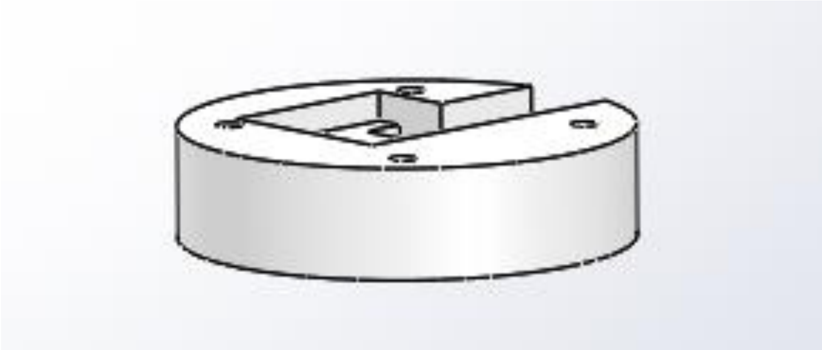
Block Diagram



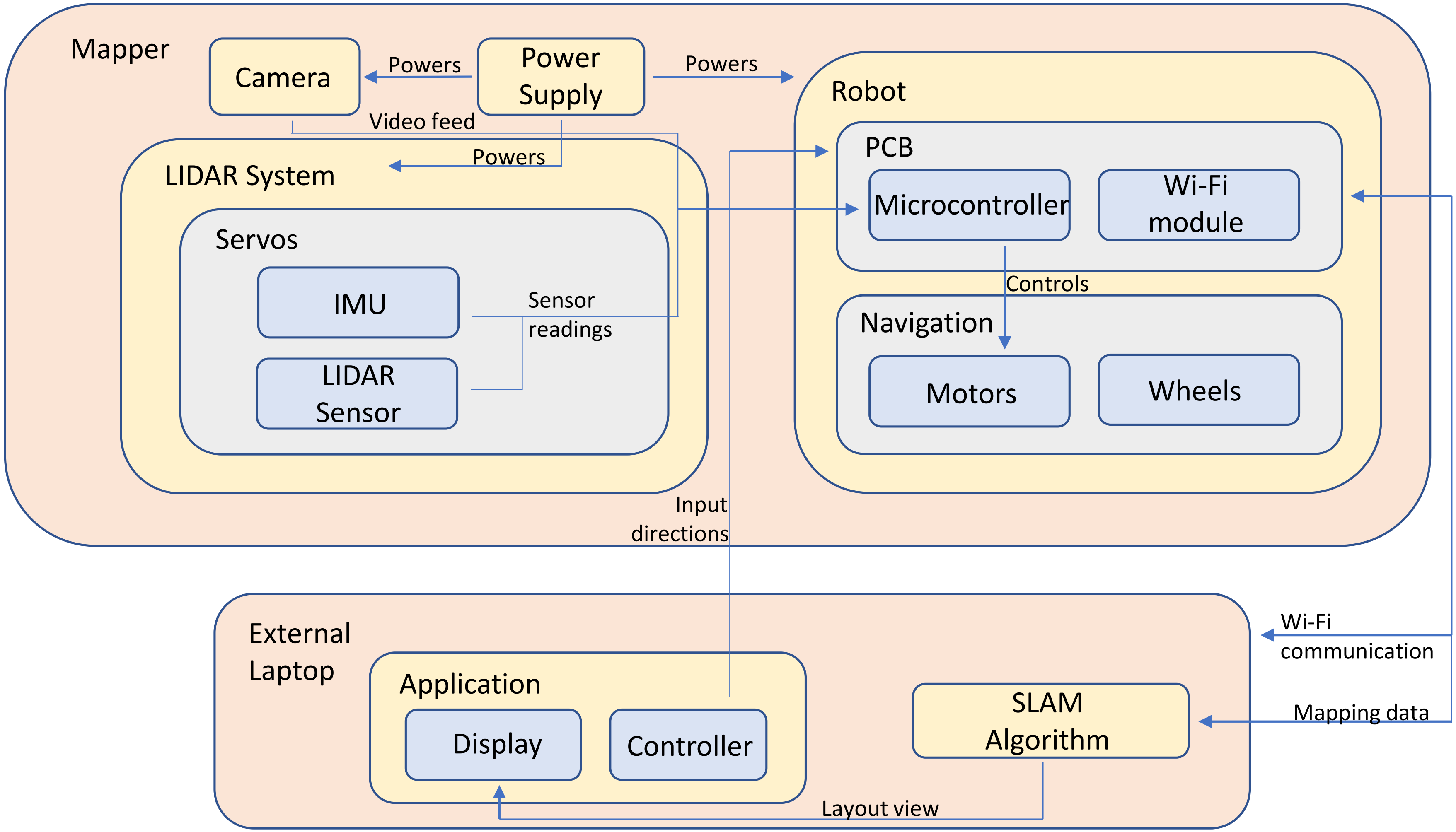
MDR Deliverable (Robot)

Two main components

- Chassis
  - Will be cut out from wood, since wood is lightweight and easy to shape.
- LIDAR mount
  - Will be 3D printed, since the dimensions are fairly small and high precision is required.

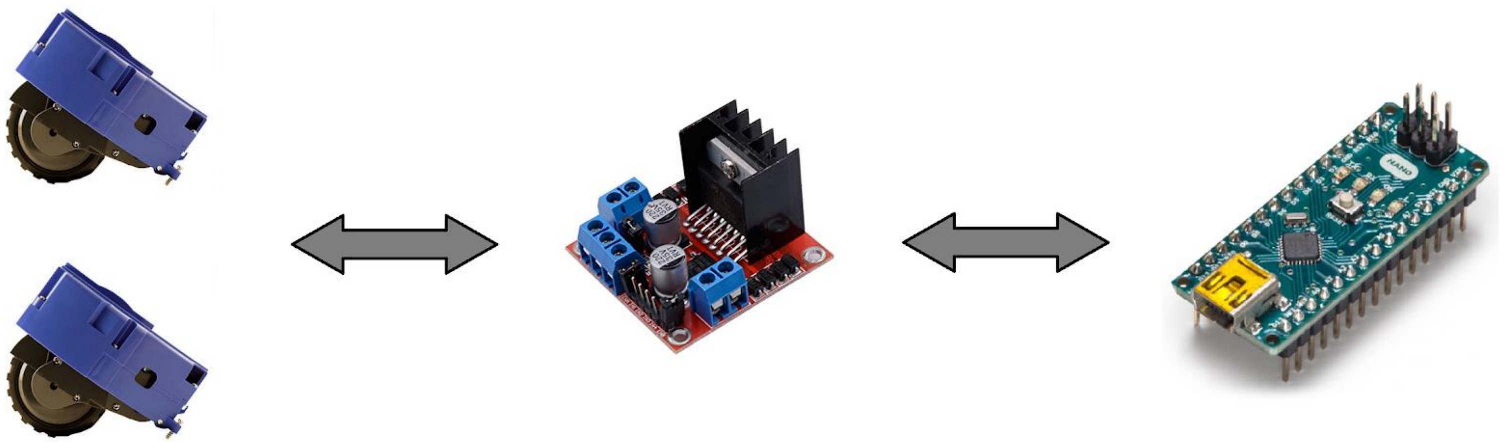


Block Diagram



MDR Deliverable (Motor)

- We decided to use the Roomba’s motors in our custom robot
- There are two motors that control each wheel
- Used an Arduino to code the wheel controls
- Motors wired to H bridge that can control both wheels independently



MDR Deliverable (Motor)

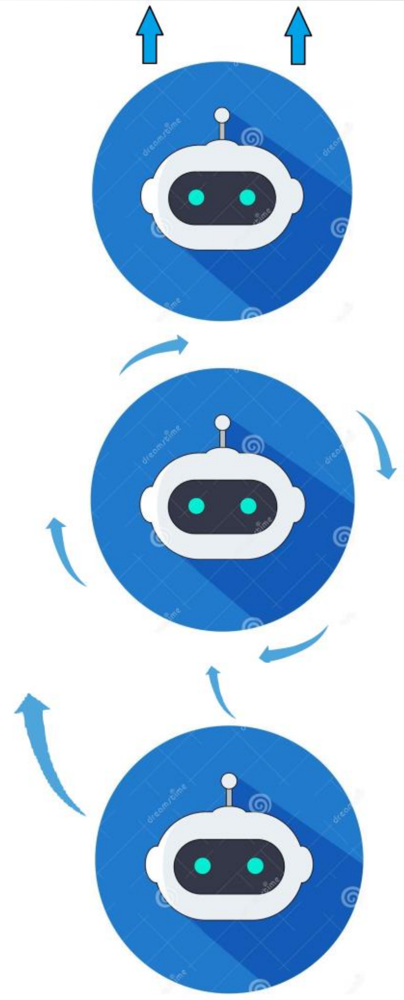
- How the wheel works:
  - Two terminals across the motor,
  - If terminal A goes to high, (9-18V), and terminal B goes low, (0V), the wheel moves forward,
  - If A goes low, and B goes high, the wheel moves backwards

- We can control motor speed via:
  - High voltage across the two terminals
  - A PWM signal that enables A and B to go high or low (H bridge)

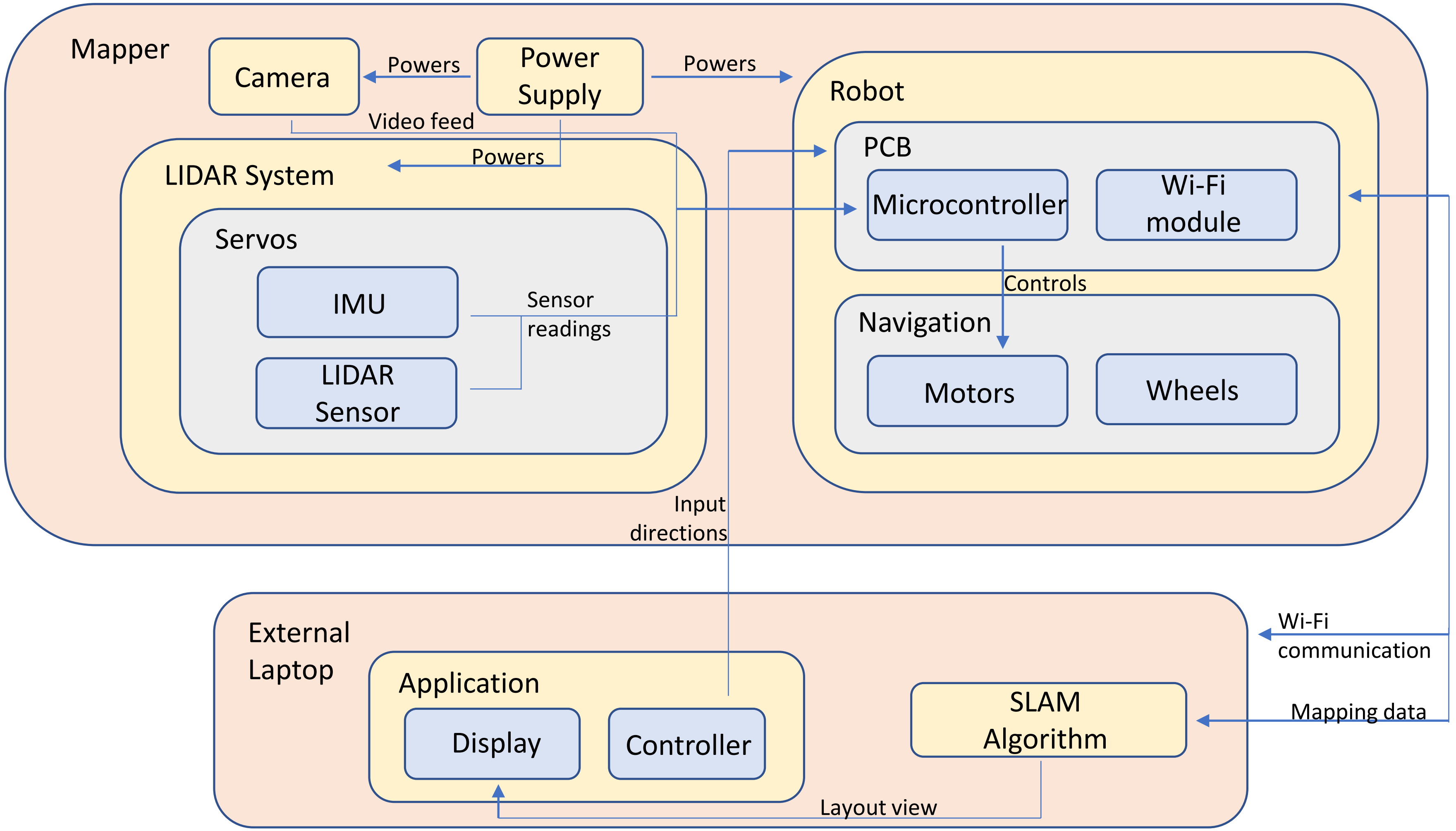
MDR Deliverable (Motor)

We'll focus on 3 major robot motor movements:

- Forward/Backwards:
  - Both wheels receive the same PWM signal
  - Terminals of both wheels are the same
- Rotate in place:
  - Both wheels receive the same PWM signal
  - Terminals of both wheels are opposite each other
    - One wheel moves forward, the other backwards
- Curve left or right:
  - Terminals of both wheels are the same
  - Each motor receives a different PWM signal

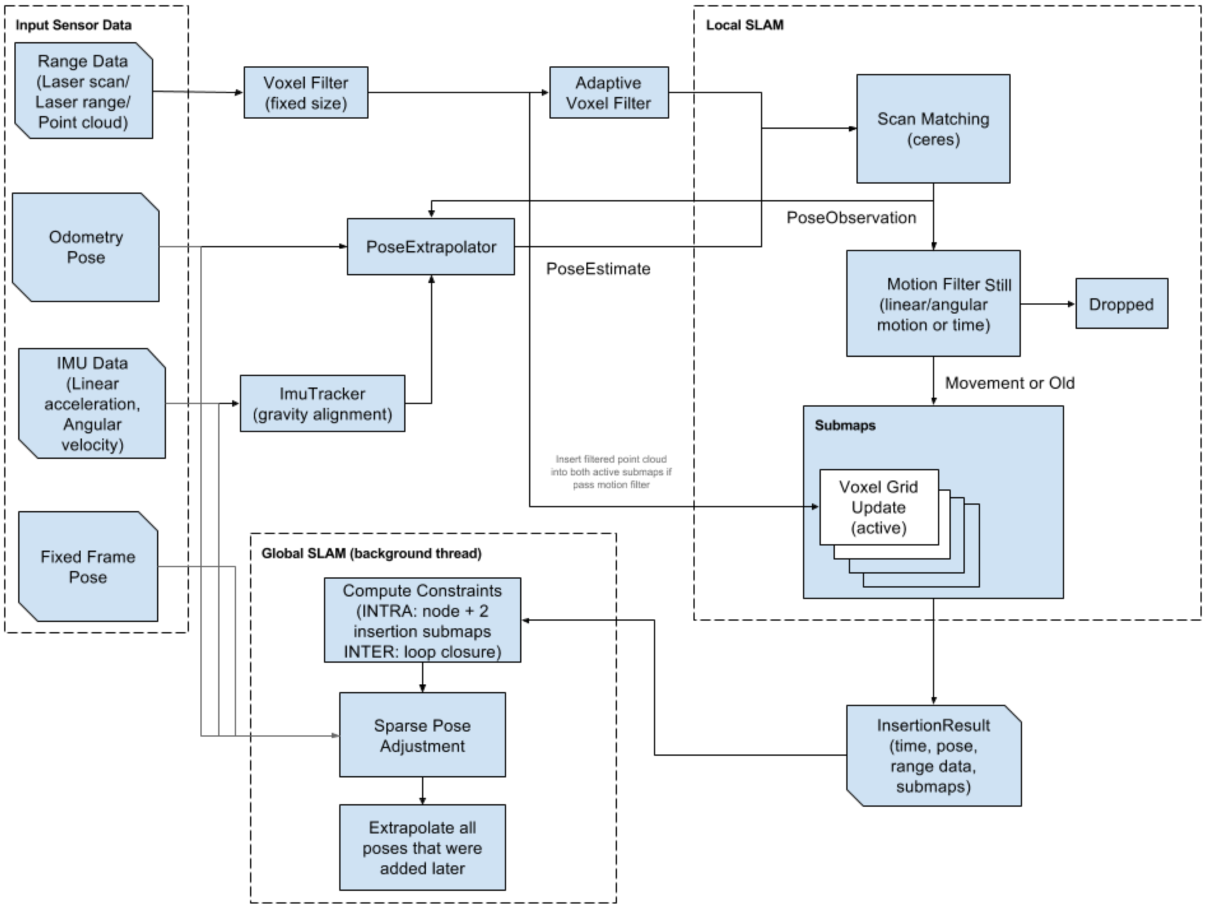


Block Diagram



MDR Deliverable (2D SLAM)

Google Cartographer Algorithm Overview



MDR Deliverable (2D SLAM)

RPLIDAR A2 → rplidar ROS package

- read RPLIDAR raw scan result using RPLIDAR's SDK
- convert to ROS /scan messages
- record /scan messages to a rosbag

ROS /scan messages → Google Cartographer node

- node processes data with Cartographer SLAM algorithm
- scans combined to generate local submaps
- local submaps merged to global map

MDR Deliverable (2D SLAM)

SparkFun 9DoF Razor IMU → razor_imu_9dof ROS package

- read IMU sensor data
- convert to ROS /imu messages
- record /imu messages to a rosbag

Output (displayed in ROS rviz)

- Pointcloud
- Local and global map
- IMU orientation
  - not yet integrated into Cartographer SLAM algorithm

MDR Deliverable (2D SLAM)

Open Source

- Cartographer libraries
- ROS packages/nodes
- RPLIDAR A2 SDK

Our contributions

- LIDAR ROS integration
- IMU calibration and ROS integration
- LIDAR and IMU compatibility with ROS Cartographer package
- ROS configurations

Demo

What we plan to bring to CDR

CDR Deliverables

- 3D SLAM
  - Simultaneous LIDAR and IMU data input
  - Pan & tilt calibration and integration
- Functional robot
  - Mount motors to our custom robot
  - Integrate custom PCB
- Wi-Fi connectivity between robot and PC

Responsibilities

- Kelvin (ME)
  - Ensure functional robot and mount SLAM components
- Marcus (EE)
  - Create PCB, ensure functional robot, and Wi-Fi data transfer
- Derek (CSE) & Bryan (CSE)
  - Programming 2D SLAM → 3D SLAM and Wi-Fi data transfer

Challenges

- 2D SLAM → 3D SLAM
  - 2D SLAM requires just the LIDAR sensor
  - 3D SLAM requires LIDAR, IMU, and pan & tilt servos
  - Integration of all components to SLAM algorithm will greatly increase overall complexity
- Wi-Fi connectivity between robot and PC
  - Transfer of camera data, SLAM sensor data, and user navigation controls
- Integration of all components

What we plan to bring to FPR and Demo Day

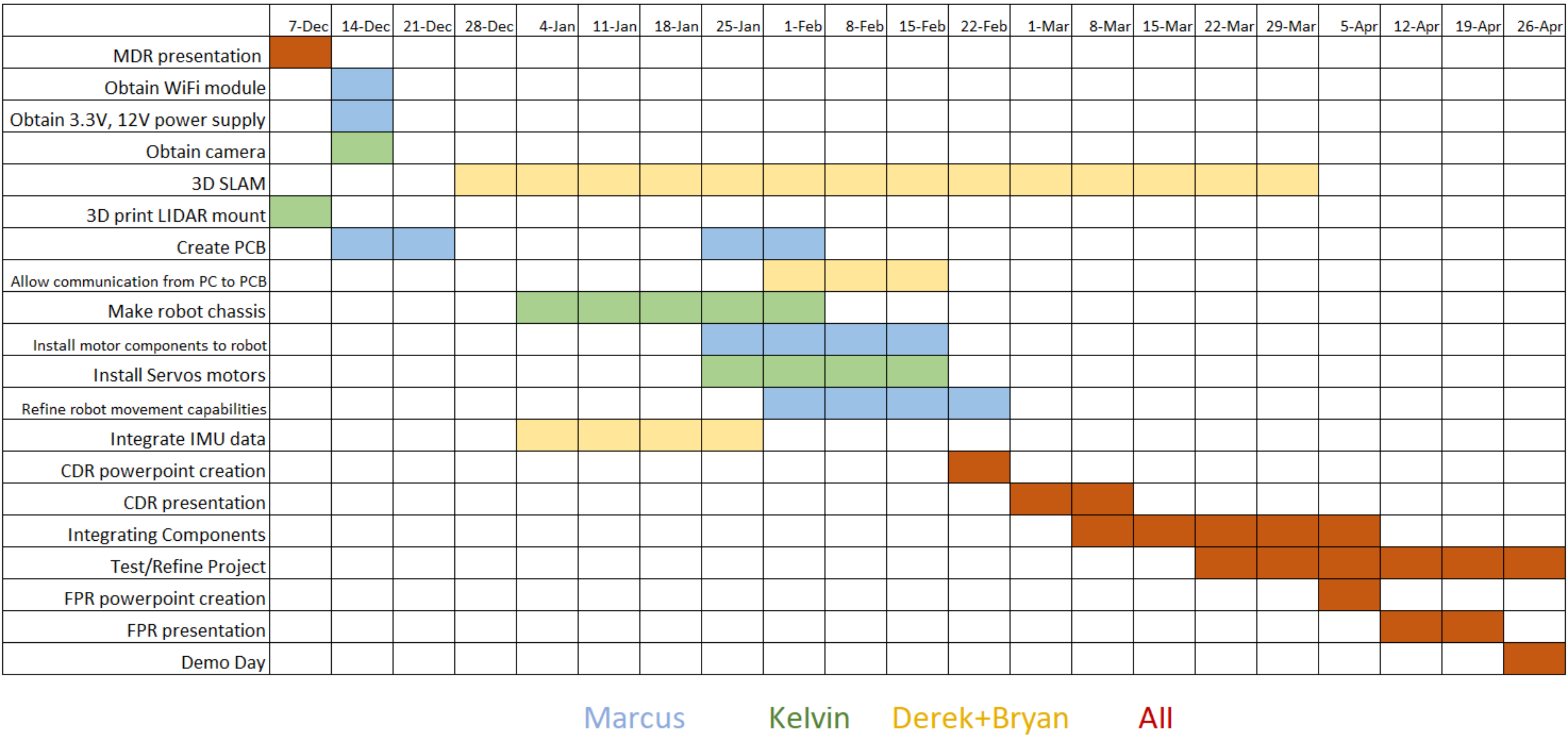
FPR

- 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

Schedule


7-Dec
14-Dec
21-Dec
28-Dec
4-Jan
11-Jan
18-Jan
25-Jan
1-Feb
8-Feb
15-Feb
22-Feb
1-Mar
8-Mar
15-Mar
22-Mar
29-Mar
5-Apr
12-Apr
19-Apr
26-Apr
MDR presentation
Obtain WiFi module
Obtain 3.3V, 12V power supply
Obtain camera
3D SLAM
3D print LIDAR mount
Create PCB
Allow communication from PC to PCB
Make robot chassis
Install motor components to robot
Install Servos motors
Refine robot movement capabilities
Integrate IMU data
CDR powerpoint creation
CDR presentation
Integrating Components
Test/Refine Project
FPR powerpoint creation
FPR presentation
Demo Day
Marcus
Kelvin
Derek+Bryan
All

Questions?