Mobile Steering Wheel: Final Presentation

Xiao Jin, Yuxuan Zhu, Qiaoan Shen

Use Case/Application

Bulky & Expensive

Immersive Gaming

Portable & Cheap

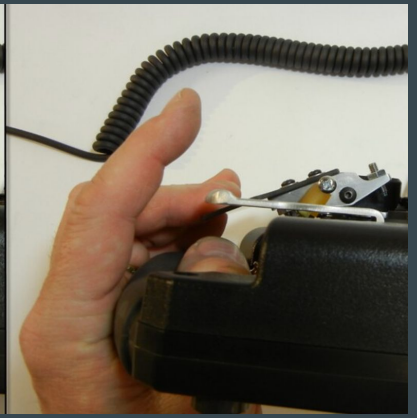
Non-ideal Controls

Best of both worlds?

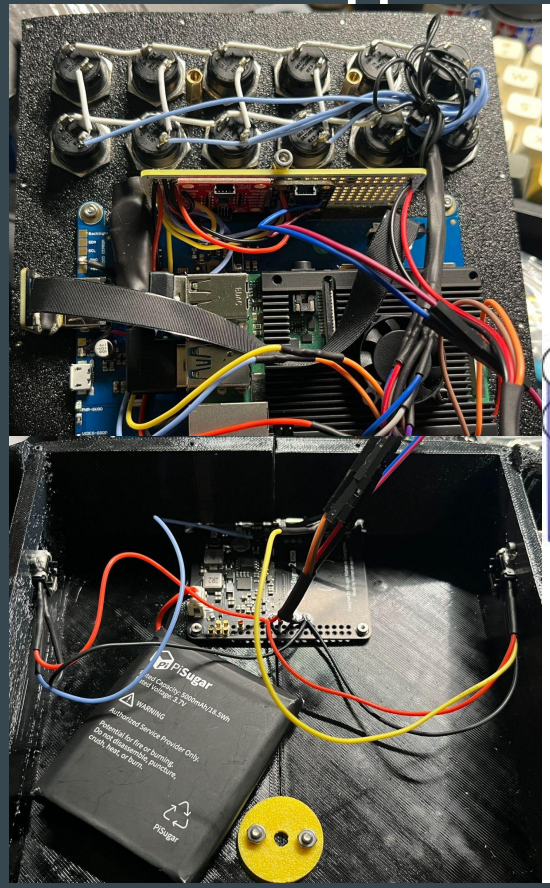
- Push Buttons
- Info Display
- Gas & Brake Control
- Steering Angle Input
- Portable
- Cheap

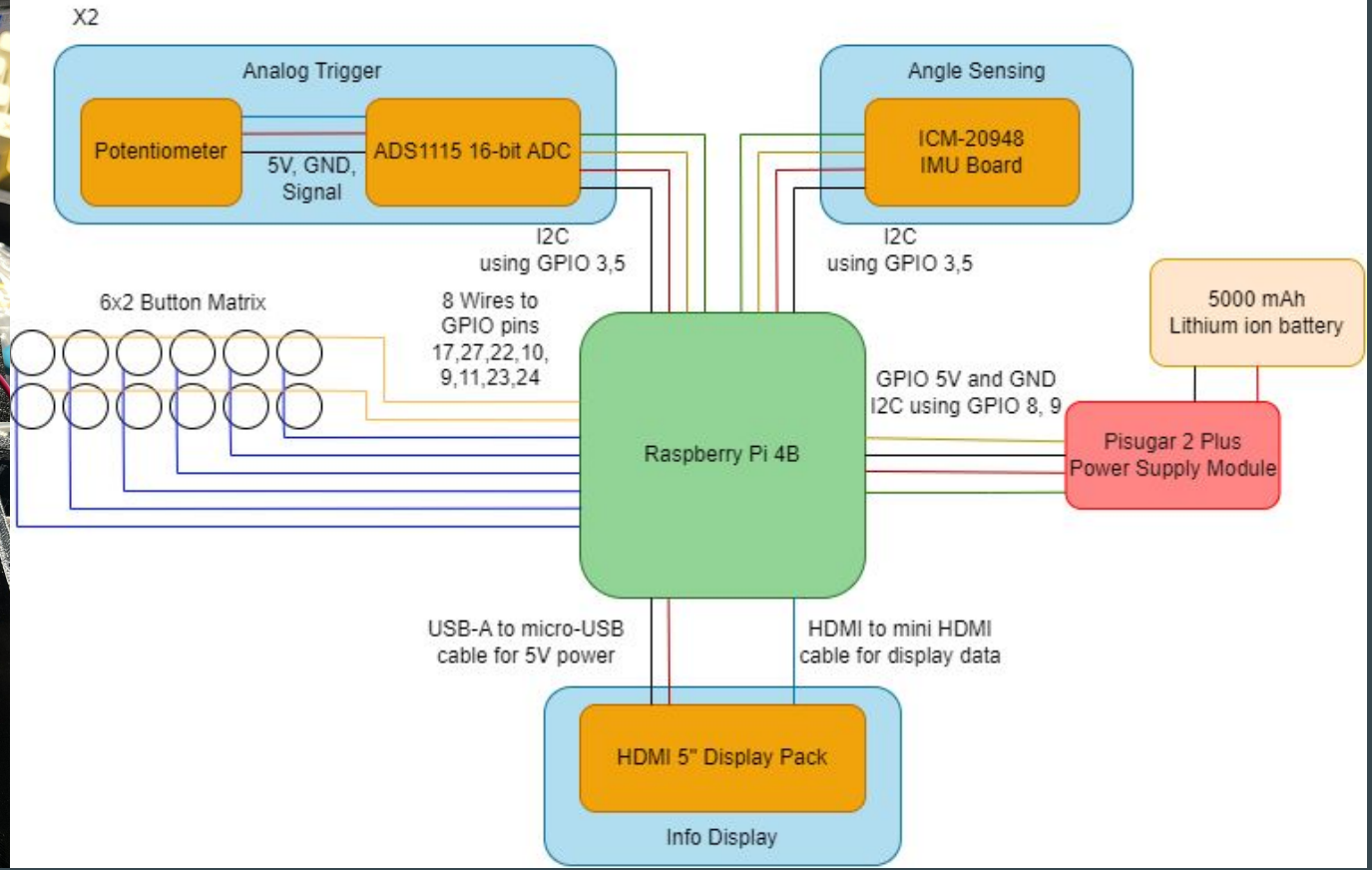
Design Requirements

- 12 Buttons
- 2 Analog Inputs
- 360-Degree Silt Sensing
- 8-hour Battery Life
- 5 inch LCD Screen
- Wireless Delay below 20ms
- Less than 400g



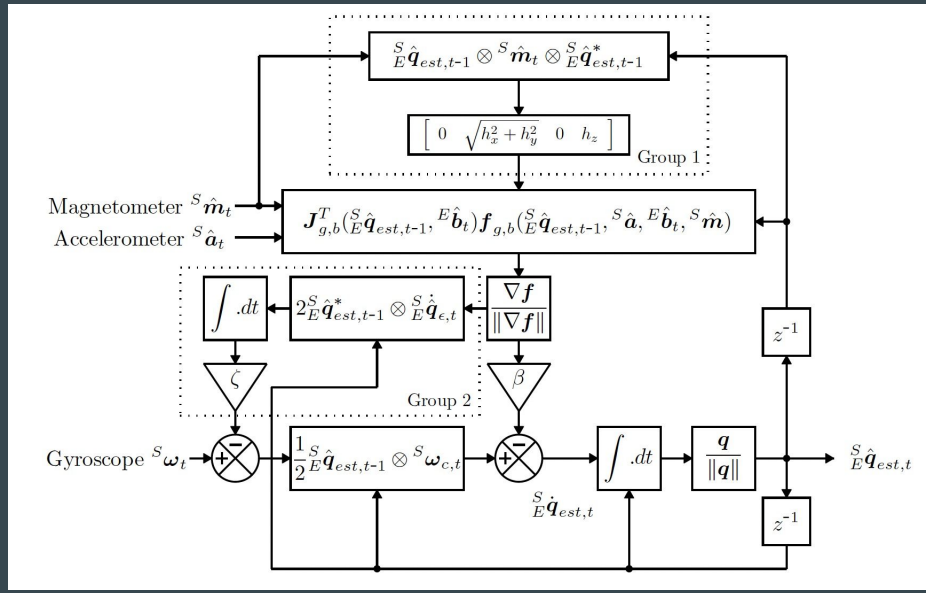
Solution Approach - Hardware



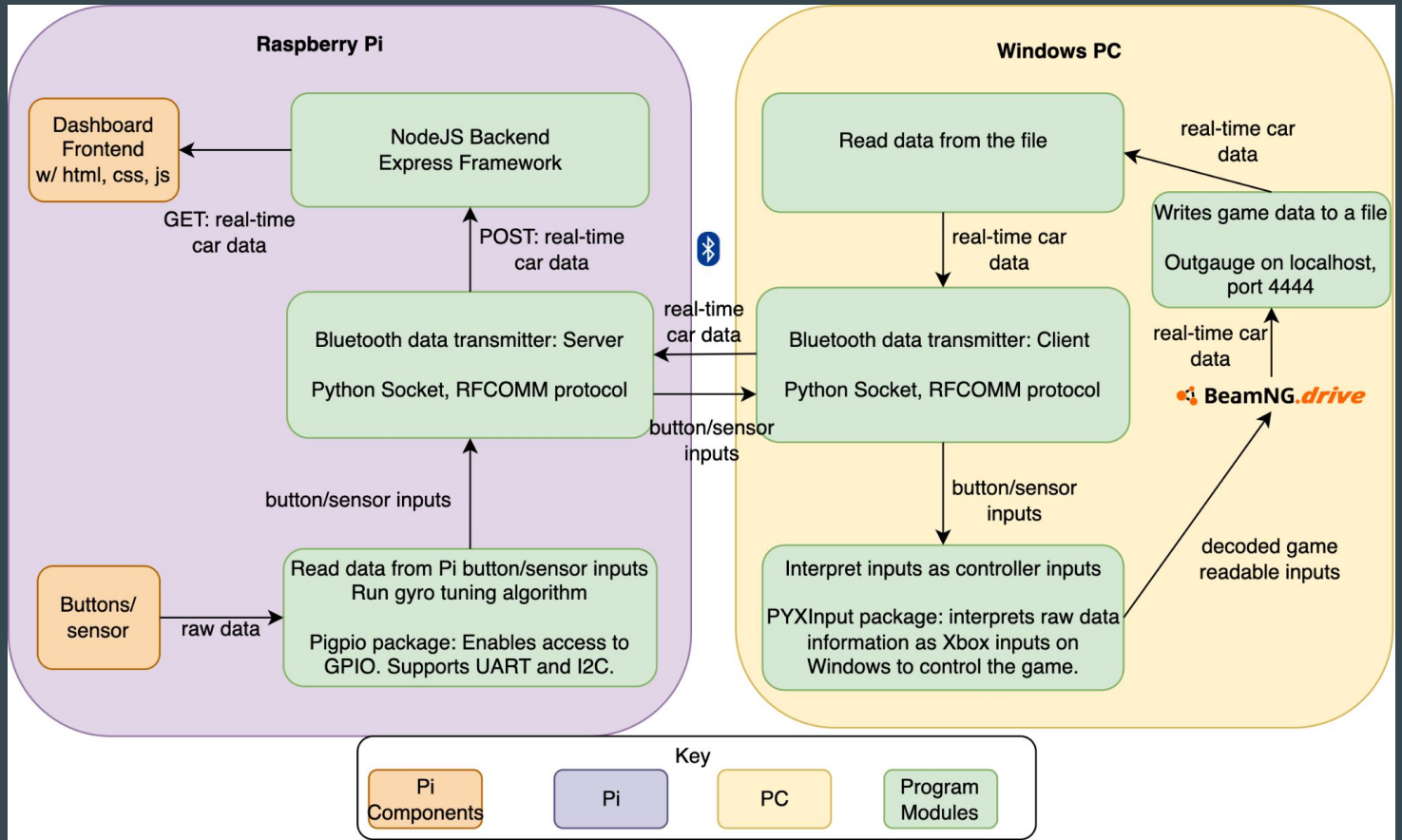


Solution Approach - Gyro Tuning

- Traditionally accelerator error is reduced through low pass filter and gyro error is reduced through high pass filter
- Madgwick filter takes in data from magnetometer, accelerator and gyroscope to estimate an optimised quaternion **q**
- Quaternion represents the relationship between Earth frame and sensor frame
- Starts with its objective function for expected orientation. Minimize the loss using gradient descent



Solution Approach - Software



System Implementation/Complete Solution

Testing, Verification and Validation

Metric	Testing Method	Goal
Weight	Using electronic scale to weigh	< 400g
Battery Life	Connect the fully charged controller to the game. Measure the time it runs out of battery as the controller continuously communicates with the game	8 hours
Input Latency	Measure the time difference between sending a request and receiving a response. The time difference is the Round Trip Time (RTT). Input Latency = RTT/2	< 20 ms
Tilt Error	Tilt the controller to a specific angle. Then compare the angle to the readings from the controller program.	< 0.5 degrees
Gyroscope Drift	Place the steering wheel on a flat surface and check the gyroscope reading, it should not change over 1 degree and we will measure the time it keeps the output in range	> 60 s

Testing, Verification and Validation

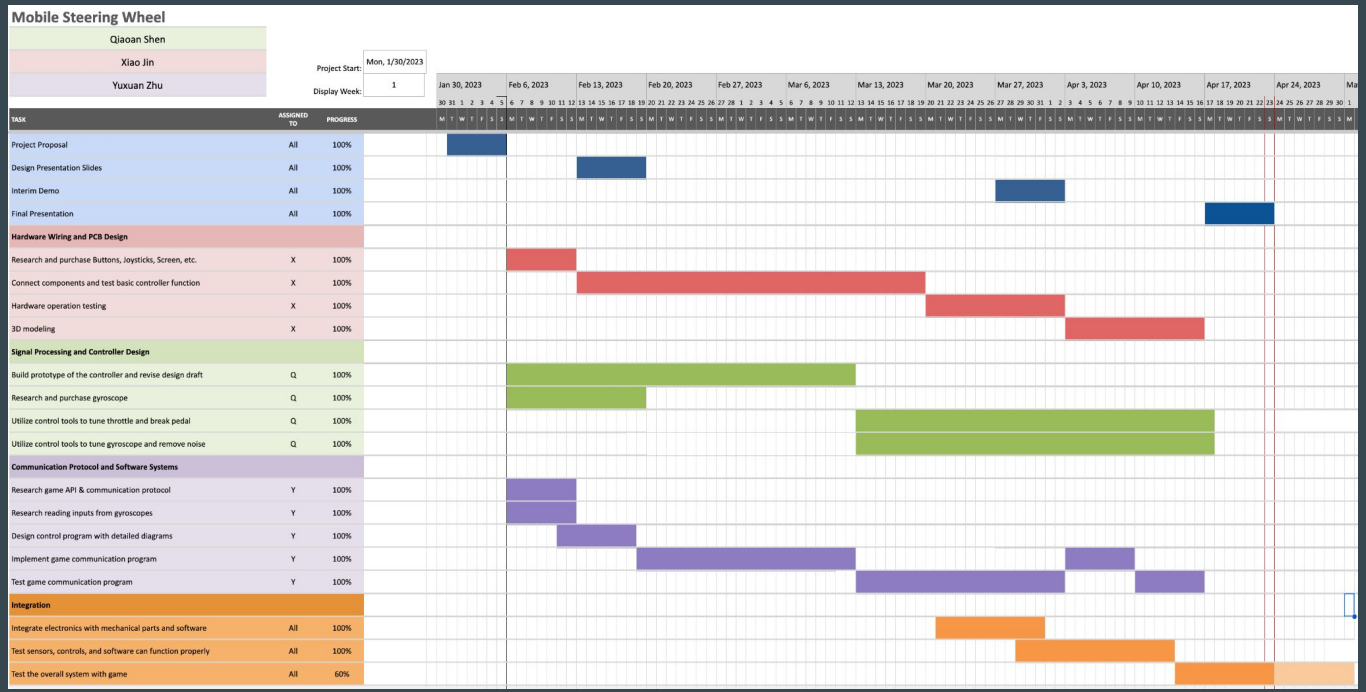
- Design tradeoffs
  - Use Python instead of C.
    - Faster implementation & less debugging over code speed
  - Implement gyroscope filter based on library Madgwick filter function instead of writing a filter of our own
    - More accuracy, better performance
    - Support with quaternion transformation and calculation
  - Prioritize weight over battery life
    - From user experience perspective, if the steering wheel goes too heavy it will be hard for users to control it
    - On the other hand, users are less likely to play the game for many hours

Testing, Verification and Validation - Results

Metric	Testing Method	Results
Weight	Using electronic scale to weigh	WIP
Battery Life	Connect the fully charged controller to the game. Measure the time it runs out of battery as the controller continuously communicates with the game	WIP
Input Latency	Measure the time difference between sending a request and receiving a response. The time difference is the Round Trip Time (RTT). Input Latency = RTT/2	WIP
Tilt Error	Tilt the controller to a specific angle. Then compare the angle to the readings from the controller program.	WIP
Gyroscope Drift	Place the steering wheel on a flat surface and check the gyroscope reading, it should not change over 1 degree and we will measure the time it keeps the output in range	WIP

Project Management

- The only job left is testing with the game



Mobile Steering Wheel

Qiaoan Shen
Xiao Jin
Yuxuan Zhu

Project Start: Mon, 1/30/2023
Display Week: 1

TASK	ASSIGNED TO	PROGRESS
Project Proposal	All	100%
Design Presentation Slides	All	100%
Interim Demo	All	100%
Final Presentation	All	100%
Hardware Wiring and PCB Design		
Research and purchase Buttons, Joysticks, Screen, etc.	X	100%
Connect components and test basic controller function	X	100%
Hardware operation testing	X	100%
3D modeling	X	100%
Signal Processing and Controller Design		
Build prototype of the controller and revise design draft	Q	100%
Research and purchase gyroscope	Q	100%
Utilize control tools to tune throttle and break pedal	Q	100%
Utilize control tools to tune gyroscope and remove noise	Q	100%
Communication Protocol and Software Systems		
Research game API & communication protocol	Y	100%
Research reading inputs from gyroscopes	Y	100%
Design control program with detailed diagrams	Y	100%
Implement game communication program	Y	100%
Test game communication program	Y	100%
Integration		
Integrate electronics with mechanical parts and software	All	100%
Test sensors, controls, and software can function properly	All	100%
Test the overall system with game	All	60%

Jan 30, 2023 | Feb 6, 2023 | Feb 13, 2023 | Feb 20, 2023 | Feb 27, 2023 | Mar 6, 2023 | Mar 13, 2023 | Mar 20, 2023 | Mar 27, 2023 | Apr 3, 2023 | Apr 10, 2023 | Apr 17, 2023 | Apr 24, 2023