Mobile Steering Wheel: Design Review Presentation

Xiao Jin, Yuxuan Zhu, Qiaoan Shen

Use Case/Application

Bulky & Expensive

Immersive Gaming

Portable & Cheap

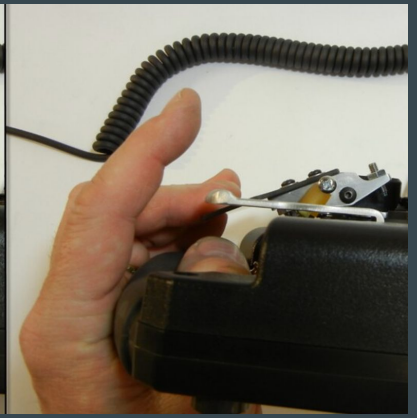
Non-ideal Controls

Best of both worlds?

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

Design Requirements

- 14 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

User Input	Buttons & Sensors
Computation & Communication	Single Board Computer /w Bluetooth or WIFI
Info Display	Small Form Factor LCD Screen
Power Management	Rechargeable Battery & Power Management Circuit
Internal Wiring	Custom PCB
Controller Enclosure	3D-Printed Plastic

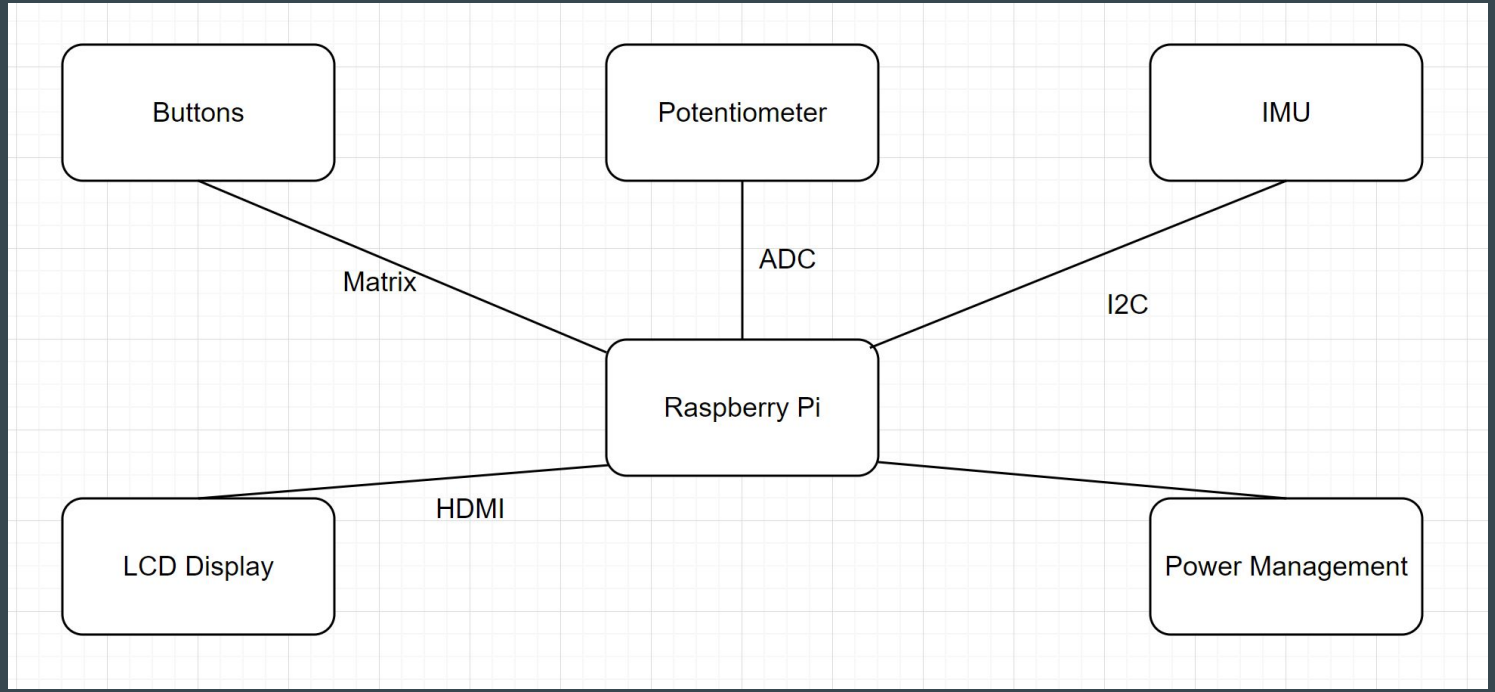
System Specification

Category	Components Chosen	Reasoning
Computing & Communication	Raspberry Pi 4B	40 GPIO pins, Bluetooth 5.0, micro-HDMI port, 5V DC power
Push Button	16mm Momentary Push Button	Cheap, Easy wiring, Large surface area
Tilt Sensing	Adafruit ICM-20948 9-DoF IMU	16-bit ADC, 3-Axis gyro, up to 2000 dps
Info Display	Adafruit HDMI 5" Display	HDMI interface, USB power
Gas & Brake Input	Potentiometer - 10K ohm	Analog sensing
Rechargeable Battery	PiJuice HAT Power Board	Lipo battery, Onboard charging, Full API with RPi OS
Lightweight	3D-Printed Housing	Carbon Fiber Reinforced, Lightweight Structure, Fast Prototyping

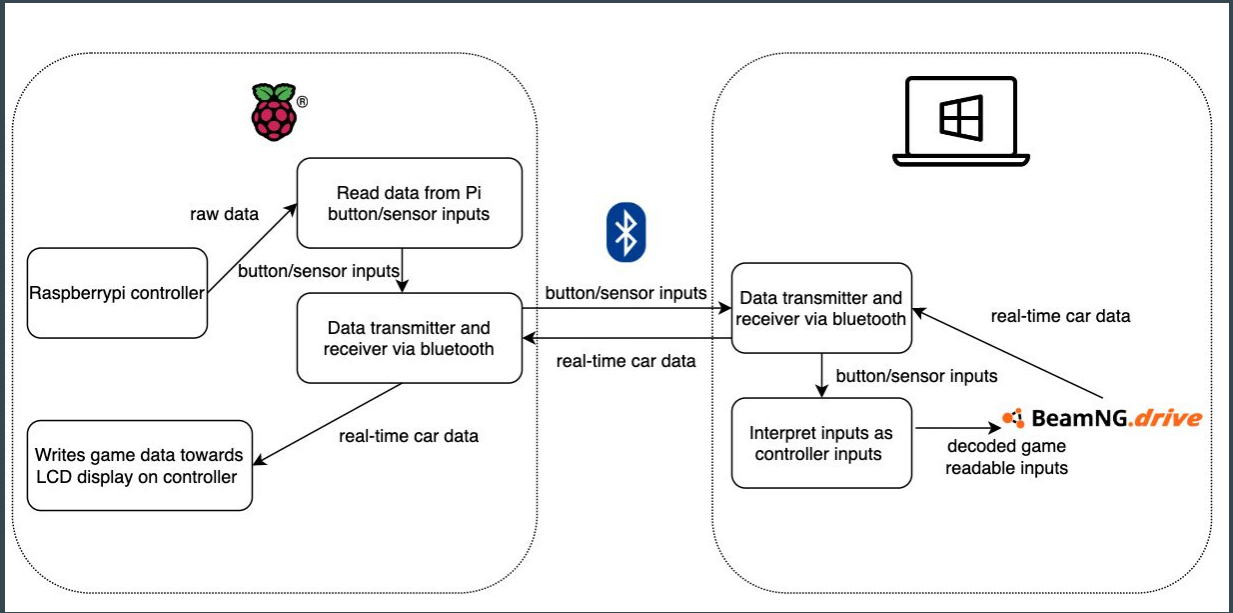
Implementation Plan

- Physical Design
  - We will create 3D print shell for our steering wheel
  - We will purchase and install Adafruit HDMI 5" Display as screen
- Sensors
  - We will purchase and install gyroscope (Adafruit ICM-20948 9-DoF IMU)
  - We will purchase and install 10k ohms for breaks and pedals
  - We will write software programs to process the input from sensors
- Integration
  - We will purchase and program on Raspberry Pi 4B as the controller of our project
  - We will wire the components by ourselves
  - We will write program to communicate with the game as its controller

Hardware Block Diagram



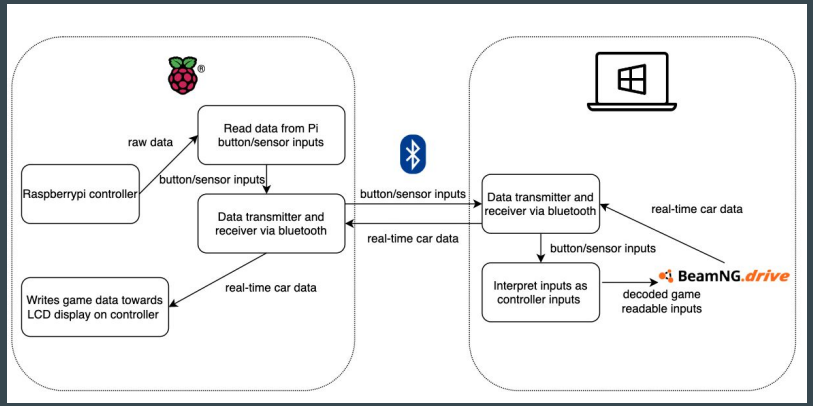
Software Block Diagram



Software Implementation

PC Side

- Reading real time game data
  - Outgauge protocol
- Interpret game controller inputs as Xbox inputs on PC
  - PYXInput python package
  - Xbox controller emulator



Pi side

- Reading data from Pi
  - Pigpio python package
  - Supports reading from GPIO pins, UART, I2C, SPI

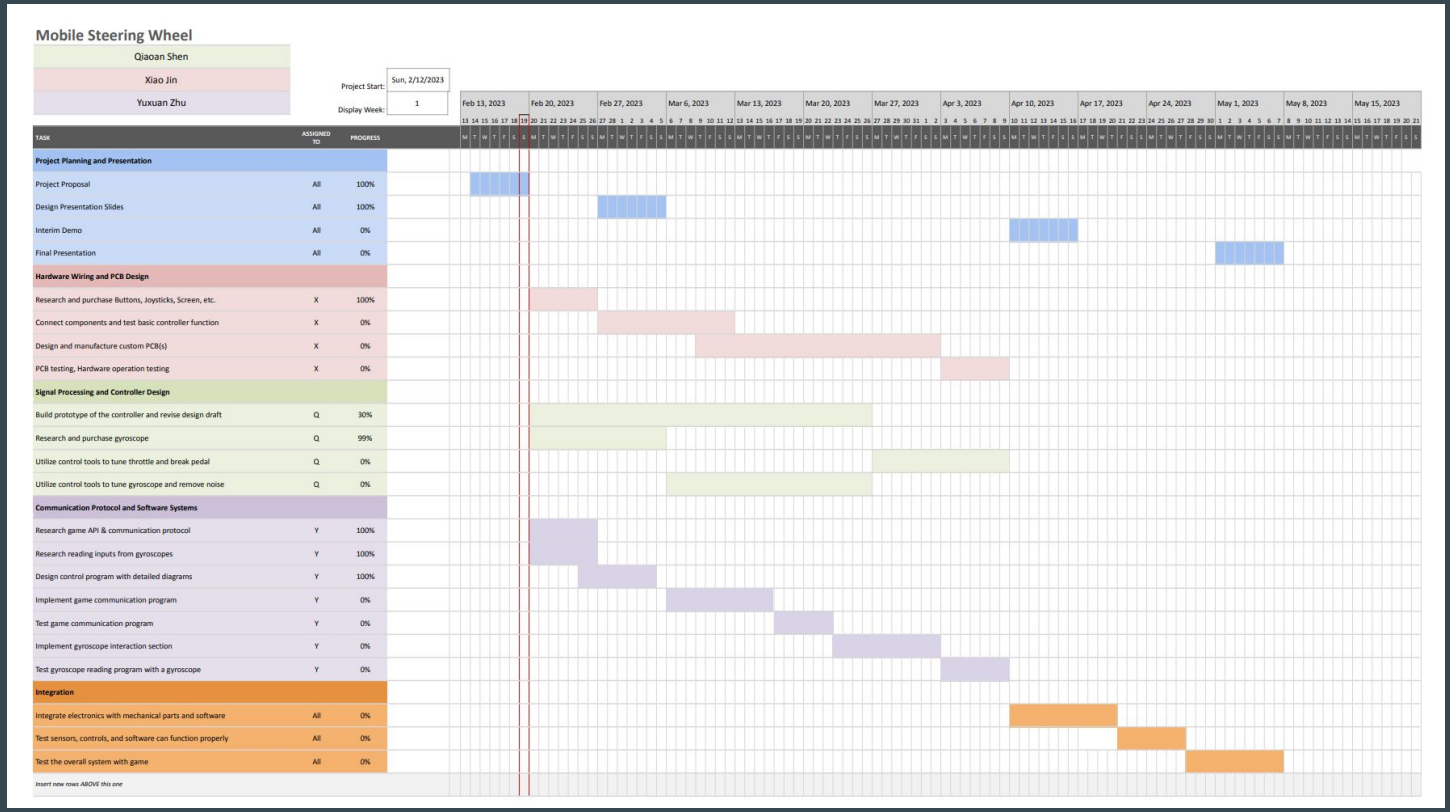
Both

- Bluetooth communication between two devices
  - Pybluez: supports bluetooth communication between devices

Test, Verification and Validation

Metric	Testing Method	Goal
Weight	Using electronic scale	< 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	Use a slow motion camera. Count number of frames it takes for the game to react after user presses a button. Latency = # of frames * 1000ms/fps	< 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
Data Rate	Write a program to calculate the total size of data transmitted by controller and received from game per second.	< 125 kB/s

Project Management



Mobile Steering Wheel

Qiaoan Shen
Xiao Jin
Yuxuan Zhu

Project Start: Sun, 2/12/2023
Display Week: 1

TASK	ASSIGNED TO	PROGRESS
Project Planning and Presentation		
Project Proposal	All	100%
Design Presentation Slides	All	100%
Interim Demo	All	0%
Final Presentation	All	0%
Hardware Wiring and PCB Design		
Research and purchase Buttons, Joysticks, Screen, etc.	X	100%
Connect components and test basic controller function	X	0%
Design and manufacture custom PCB(s)	X	0%
PCB testing, Hardware operation testing	X	0%
Signal Processing and Controller Design		
Build prototype of the controller and revise design draft	Q	30%
Research and purchase gyroscope	Q	99%
Utilize control tools to tune throttle and break pedal	Q	0%
Utilize control tools to tune gyroscope and remove noise	Q	0%
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	0%
Test game communication program	Y	0%
Implement gyroscope interaction section	Y	0%
Test gyroscope reading program with a gyroscope	Y	0%
Integration		
Integrate electronics with mechanical parts and software	All	0%
Test sensors, controls, and software can function properly	All	0%
Test the overall system with game	All	0%

Insert new rows ABOVE this one