Bryan Yao: ID: 922709642 | bryao
John Carter: ID: 923104855 | nuttycream
Jingxi Hu ID: 921182779 | Heison0818
Github Link

CSC680 Application Development for Mobile Devices

Project Proposal - Pipin

This project involves the development of a remote mobile Device using SwiftUI, designed to control GPIO (General Purpose Input/Output) pins on a Raspberry Pi/embedded system. The app provides a user-friendly interface for remotely managing and interacting with GPIO pins, enabling functions such as turning pins on and off, monitoring pin status, and configuring pin modes. The application utilizes a reliable communication protocol to connect to the hardware, offering seamless control and real-time updates. This project aims to simplify the process of remotely interfacing with hardware, making it ideal for applications in automation, IoT, and embedded system development.

Project Scope

Features

Essential Features	Additional Features(Nice to have)
• Toggling Pins OFF/ON • Queued list of commands • WebSockets • HTTP requests for POST/GET/DELETE	• Robot Controls • Pullup/PullDown Individual GPIO pins • Multiple Connections Handling • Authentication

Toggling pins

Toggle individual GPIO Pins from (0-27) using SwiftUI as a WebSocket Client and the Rust Web Server as the receiver.

Queued list of Commands

Queue up a list of actions the GPIO pins will do. Such as:

- Set High/Low: sets GPIO pin to high or low
- Wait For High/Low: waits for input signal
- Delay in MS: sets a delay for individual actions

- Set Pulldown/Pullup Clocks: sets the pull type for the GPIO pin (useful for managing High Low signals)

Milestones and Timeline

A list of key milestones with expected completion dates.

Milestones 1 (3/1-3/31)Getting set up
3/1 We decide to work on the FinalProject as a group
3/1-3/7 we are thinking about what we should build for our Final Project, we have a group meeting through discord. We brainstorm some ideas about what our Final Project should look like, what kind of feature should be in our Final project. Based on our class lecture, we did cover the simple calculator. We are going to see more demos from the class lecture then decide what we should build for our Final Project.
3/10-3/21 We have another group meeting, and we make up our mindt and decide to build our Final Project device using the Raspberry Pi. While we all took CSC-415 before, we all thought building the Final Project connected with the Raspberry Pi could be cool and nice.
3/24-3/31 Spring break, we didn't start any meeting but we did communicate through discord and we discuss about some must-have features in our Final Project

Milestones 2 (4/2-4/30)Prototype
4/2 After our group discussion, we decide our must-have feature including: Toggling Pins OFF/ON, Queued list of commands, WebSockets, HTTP requests for POST/GET/DELETE
4/11-4/30 We start to work on the must-have feature in our Final Project, also, we keep communicating in discord. Also, we post and discuss our coding problem on discord, so that our group can solve the problem together and faster. Each member worked on the part they wanted.

Milestone 3(5/1-5/12): Submission
5/1-5/12: We are starting to work on our documentation for the Final Project, meanwhile, we did go over and review our code and do the final testing. We made sure everything worked as we expected.

Bryan Yao: ID: 922709642 | bryao
John Carter: ID: 923104855 | nuttycream
Jingxi Hu ID: 921182779 | Heison0818

Github Link

CSC680 Application Development for Mobile Devices

Target Audience

- Tinkerers
- Raspberry Pi Enthusiasts
- Embedded System Programmers

Future Enhancements

1. We can add an eye tracker or a camera with Raspberry Pi to track the user's eye motion and translate it into GPIO input.
2. Use speech recognition to control GPIO (like turn on/off the light, turn up/down the volume)

Hardware requirements

- Raspberry Pi 2+
- iPhone 14+
- Working internet connection

Software requirements

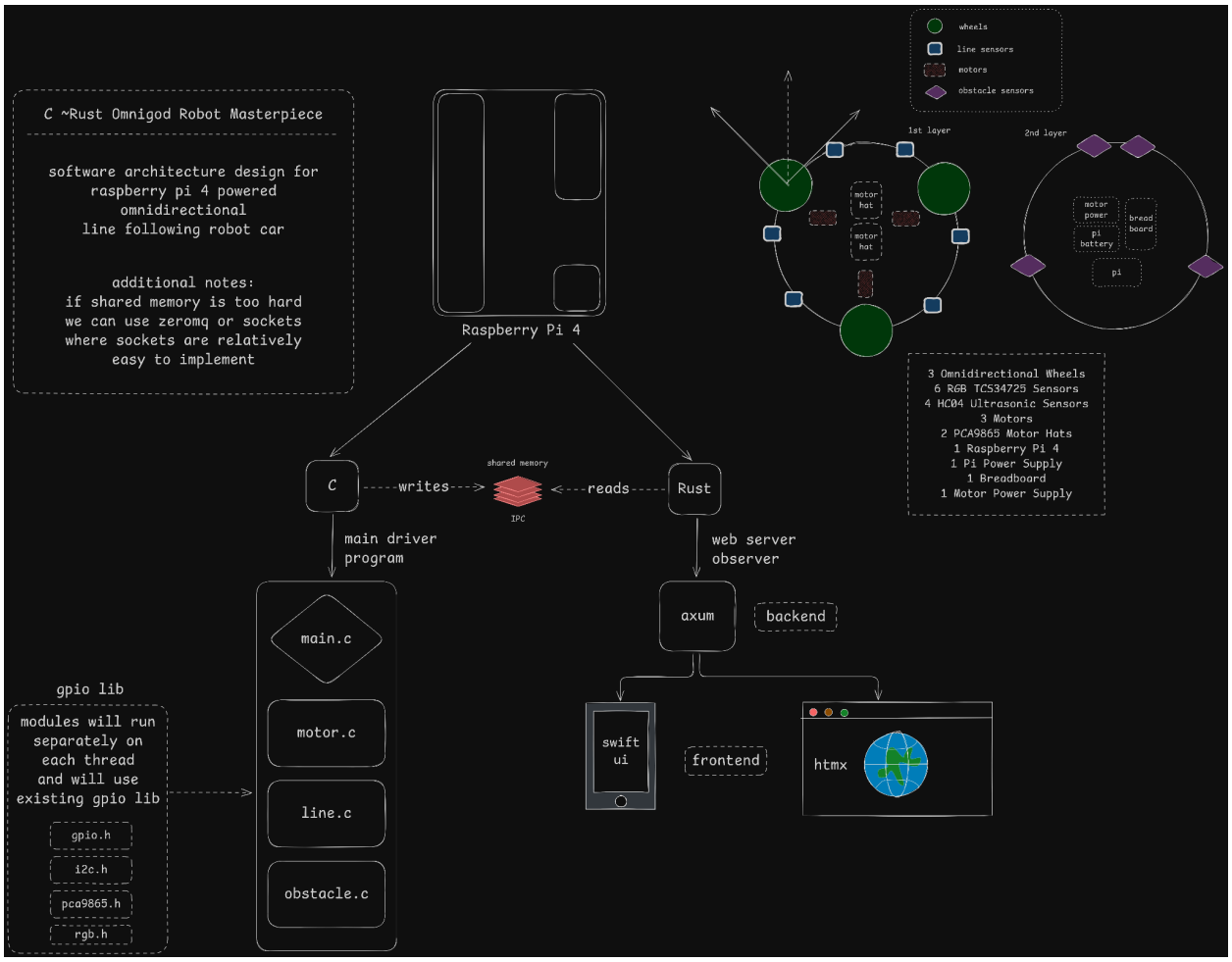
- pipin running on a Raspberry Pi
- iOS 10+

Bryan Yao: ID: 922709642 | bryao
John Carter: ID: 923104855 | nuttycream
Jingxi Hu ID: 921182779 | Heison0818

Github Link

CSC680 Application Development for Mobile Devices

High level Diagram

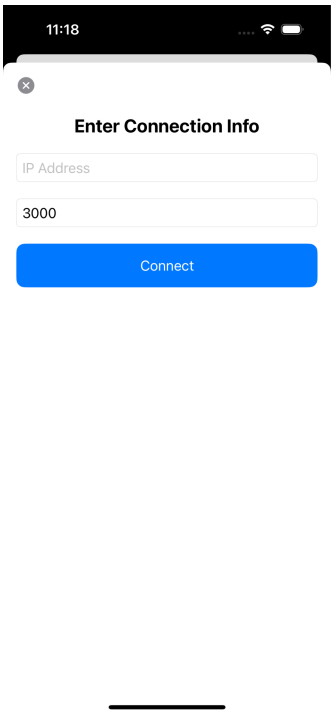


Bryan Yao: ID: 922709642 | bryao
John Carter: ID: 923104855 | nuttycream
Jingxi Hu ID: 921182779 | Heison0818

Github Link

CSC680 Application Development for Mobile Devices

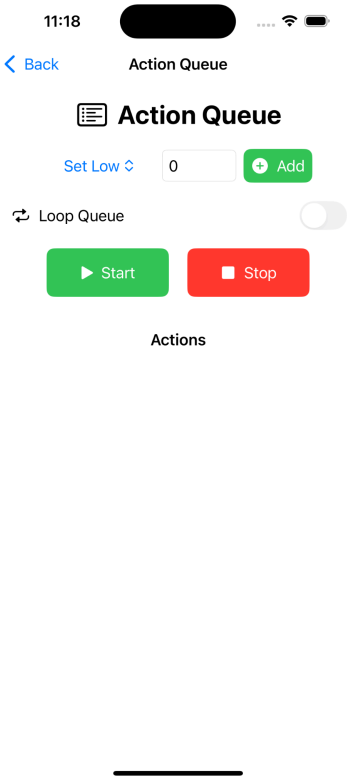
ScreenShots

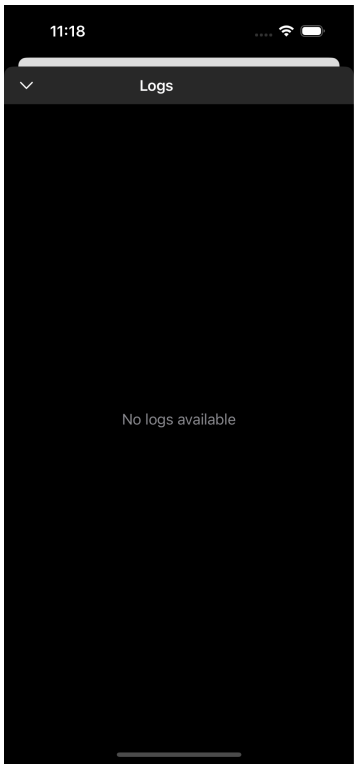


Bryan Yao: ID: 922709642 | bryao
John Carter: ID: 923104855 | nuttycream
Jingxi Hu ID: 921182779 | Heison0818

Github Link

CSC680 Application Development for Mobile Devices





Usage

- Build and Run the XCode project
- Ensure pipin is running on the Raspberry Pi
- Connect to the IP address and Port
- Initialize the GPIO pins by pressing Initialize
- Use as you please

Bryan Yao: ID: 922709642 | bryao
John Carter: ID: 923104855 | nuttycream
Jingxi Hu ID: 921182779 | Heison0818

Github Link

CSC680 Application Development for Mobile Devices

Acknowledgements/Resources

pipin - the original self-contained web app that also doubles as a web server. Made originally for testing GPIO pins. It can also be used to manage and control them using a relatively simple GUI. Built with Rust by the way.

OpenAI ChatGPT - UI MockUps to clone existing Web App look. While we did use the original Web App as heavy inspiration, the UI mockup ChatGPT gave us was lacking in terms of consistency and ergonomics. Significant changes were made to look both pleasant and easier to use on a mobile phone.