Vergil Mendizabel

May 3, 2023

EECE 4712

Robinson

Final Project Write-up

Hardware:

Calendar

Description automatically generated with medium confidence Graphical user interface

Description automatically generated

Schematic PCB Design

Graphical user interface

Description automatically generated

Printed PCB w/ Components

Designing the hardware for this project was straightforward. There were no complications in finding a custom footprint for the RP2040 microcontroller and the Kalih Switches. Soldering also proved to have no complications pertaining to the buttons, but the RP2040 had a complication due to it coming with header pins that couldn’t be easily desoldered. Otherwise using solder paste and a heat gun made it an easy process to solder onto the PCB.

Software:

Text

Description automatically generated Text

Description automatically generated

HID Descriptor Base Code Initialization of Pinouts and Device Descriptor

Text

Description automatically generated A screenshot of a computer screen

Description automatically generated with medium confidenceInitialization of Axis for D-Pad and L/R Stick SOCD Cleaner and D-Pad Movement

Text

Description automatically generated Text

Description automatically generated

Button Input HID Report Initialization of Pins and main() function

For the project I had to write an HID descriptor to be able to report to the host computer, so the computer can recognize it is a controller that can be seamlessly used by Steam. The HID Descriptor basically sends binary data that sends reports that specifies to the host computer that allows the host to recognize the device type and the inputs that are being registered to it. Writing an HID descriptor from scratch is a complicated process that requires a deeper understanding to be able to do. To subvert this, I used an open-source base descriptor template from a HORI Fighting Stick v3 that gives me the basis to work on the inputs I needed for my device. Most of the code above is initializing the button pinouts and its functions, but the only logical coding I made was to implement the SOCD cleaning and the recognition that the button was pressed. The most complicated part of this was using C-Make to be able to convert this into a uf2 file to upload to the RP2040. Overall, this was a great introduction to writing HID descriptors for future USB based projects.

Conclusion:

Overall, the project went very well. I was able to delve into the world of embedded systems. Writing an HID descriptor in C to make firmware for the project was a good learning experience for myself and it allows me to have knowledge for the next USB-based project I work on later down the line. Though my PCB circuit was not complicated at all, I still feel that it was a good learning experience also. It allowed me to show my knowledge of the basics of KiCad and it can be applied to later PCB projects also. This project was really enjoyable to work with and I look forward to being able to do more things like it in the future.