Boston University
Electrical & Computer Engineering
EC464 Capstone Senior Design Project

Final Prototype Test Report

Augmented Reality Climbing Wall

by

Team 14
Augmented Reality Climbing Wall

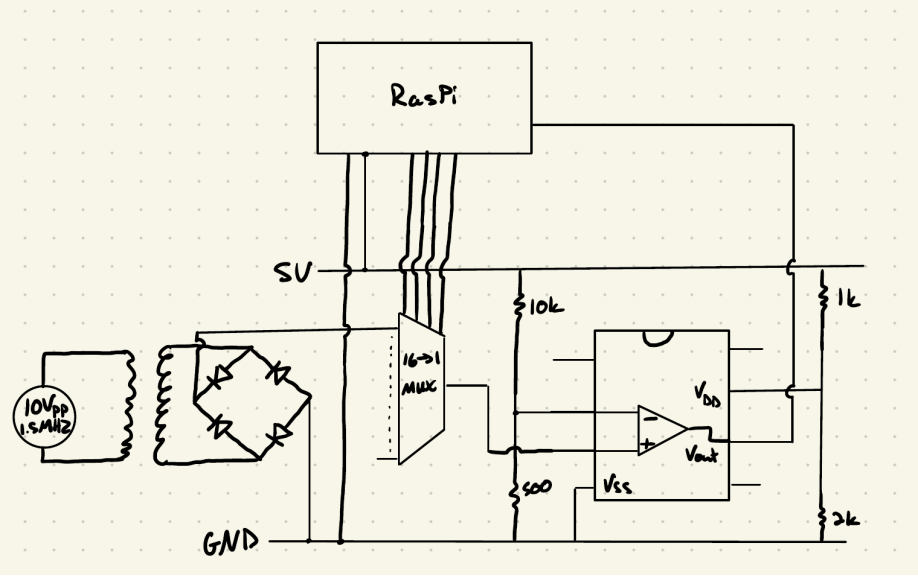
Team Members

Tom Panenko tompan@bu.edu
Taylor Hartman hartmant@bu.edu
Michael Igwe migwe@bu.edu
Ryan Smith rsmith66@bu.edu

Required Materials

- Circuitry
  - Full Bridge Rectifier
  - Diode
  - Comparator
    - MCP6241 Op Amp
    - Resistors
  - Multiplexor
  - Coils
  - Push Buttons
  - Function Generator
  - Power Supply
  - Jumper Wires
- Hardware
  - Enclosure
  - Button
  - Push Button Holder
  - Coil Holder
  - Raspberry Pi 4 Model B
  - Plywood Sheet
- Software
  - Laptop
  - Unity game engine
  - Simon Game .cs scripts

Setup



For this testing, we plan on running the game locally on a laptop and having it communicate with the raspberry pi to read the file that is being updated with the active sensor data matrix. On the laptop, the game is run on a local web browser and, to access the file information, uses a web server that shows the contents of the data.txt file and an additional proxy

server that attaches CORS headers to every request that is made. This is necessary as the web server does not allow GET requests from other unconfirmed browsers, so these headers need to be added to get access to the data. The raspberry pi continually sends the data.txt file with the multiplexing data to the laptop through ssh and overwrites the file that is being read from in the web server.

The holds use AC power to transmit wirelessly through the wall. This power is then converted into DC through a full bridge rectifier and brought up to 0V or 3.3V with a comparator. The current hold is selected with a system of multiplexers that is designed to support up to 64 holds in total. This selection and reading post comparison is all controlled by the raspberry pi. Much like last prototype testing, we set up the multiplexing capability by connecting multiple buttons and holds to the multiplexer and grounding the unused inputs. We connected everything to the multiplexer and then connected the output to a comparator that translated the signal to a constant 3.3V if the input was greater than 0.5V and a constant 0V if it was below. This then reads the data matrix of which holds were being pressed to the data.txt file which, as we discussed earlier, is sent to the laptop to overwrite the file being read from for the game.

Measurements Taken

Here is the completed score sheet that was used to track results based off the test procedure in our Final Prototype Test Plan:

Action	Correct? (1/0)
Hold data console displays all 0s when no holds pressed	1
Hold data console correctly displays 1 in the row/column that the depressed hold is assigned to.	1
Hold data console correctly displays 1s for simultaneously depressed holds	1
Hold data console correctly displays 1s for simultaneously depressed holds attached to the same multiplexer index but different GPIO ports.	1
Game plays without error and within expectations for the gameplay	1
Game successfully calibrates hold position and activates the depressed holds	1
Game reads change in depressed holds and reacts successfully	1

Total	7/7

We believe that, for this final prototype test, we were able to accurately meet all the Measurement Criteria we set in the Test Plan. This criteria specifically had to do with the project operating as a complete system, with the holds reading in the specific input of which one was pressed and not and sending that data to the game engine which plays the game accordingly. We were very happy with how the system performed as the game was able to read the input and play the game of Simon to all of our expectations. We believe that all the testers were happy as well and look forward to showcasing our completed system on ECE Day.

Conclusions

For this final prototype testing, our project behaved as we had expected and hoped, and we successfully scored a 7/7 on our measured objectives. This indicates that the complete system that we have developed is correctly connected and the hardware can read the inputs given by the user, send them to the game engine, and the game engine can update accordingly. We are extremely happy with the results and believe that the professors were happy as well as we produced a product that we believe solves the initial problems that our client approached us with. Moving forward, we were told to begin brainstorming about how we will present our project on ECE day and may even change it slightly to demonstrate the wireless power as well as the completed system including the data transfer. One thing that we will have to think about is how we will create the makeshift wall that we will attach the holds onto as, for this prototype, we used a small piece of plywood, but we will have to brainstorm on if we want to do a larger piece of wood. We are very much looking forward to ECE Day and showing off the work that we have put into creating this project.