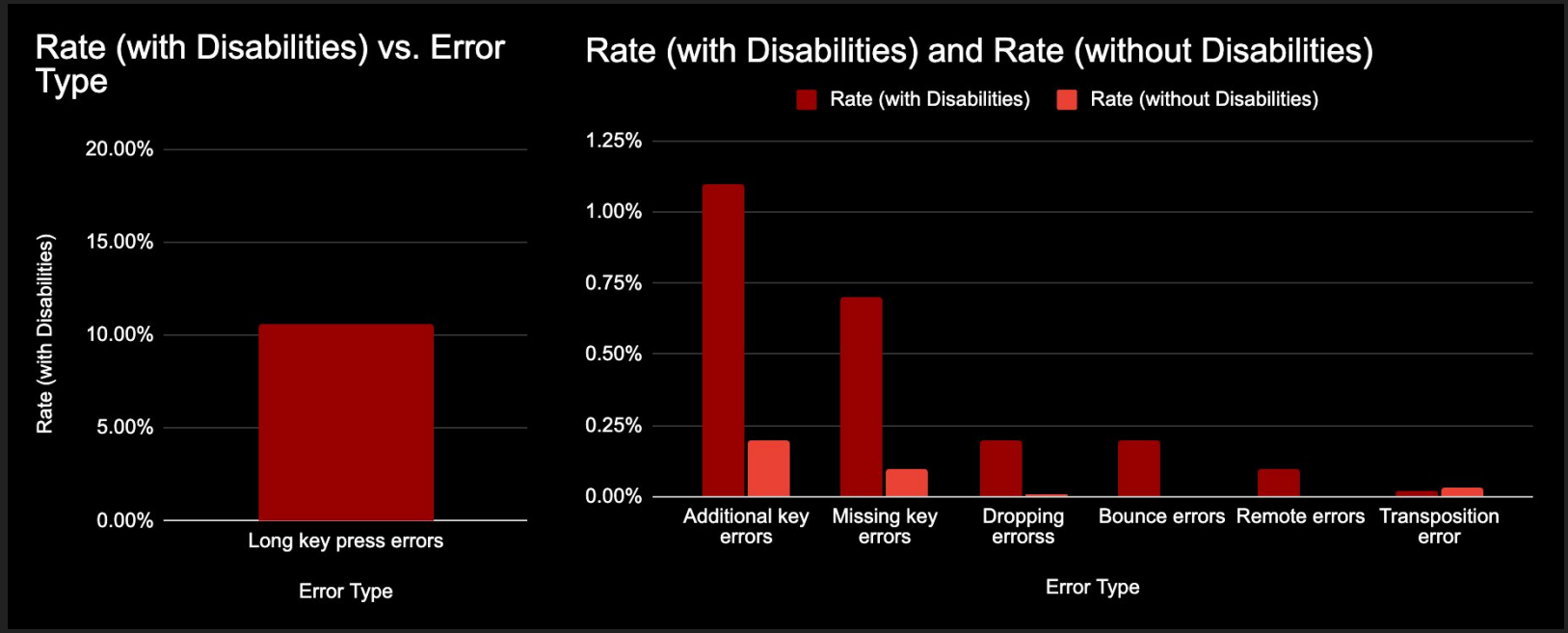
# Accessibility Pi/O

Carlos Armendariz, Jorge Tamayo, Ji Chang

# Problem Statement & Use Case

- Keyboard & mouse accessibility
- Disabilities, such as cerebral palsy
  - Most common form is spastic hemiplegia
  - Can use one arm and one leg
- Solutions are expensive and not open source
  - Or cheaper and very generic
- Limited keyboard/mouse combination options
- Designing solution covers software and circuitry

# Common Typing Errors




LoPresti, Edmund F. et al. “Measuring Keyboard Performance for People with Disabilities.” (2006).

# Correcting for most common errors

- Long key press errors
  - Most common kind of error for people with disabilities
  - Solvable using software as well as a toggle switch
  - Allow user to toggle long key press functionality on their keyboard
- Additional Key errors
  - Key guard and key size are key factors in minimizing this error
  - Optional keyguard allows the option to space out keys
- Missing key errors
  - Fixable through good key design and switch choice
  - using buttons that require minimal pressure
- Drop errors
  - Pressing two keys simultaneously treated as one
  - Shift key has both toggle and hold

# Requirements

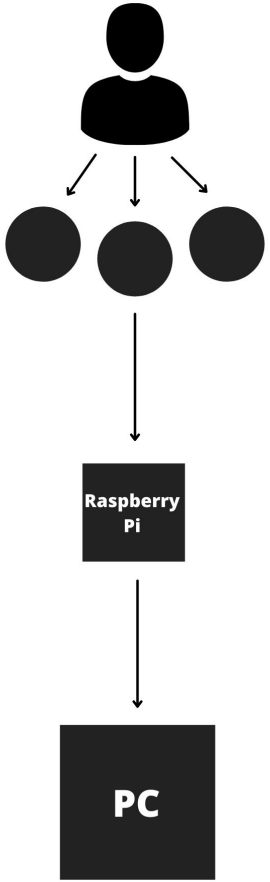
- 74 characters supported
  - 26 letters (upper and lowercase), 10 digits, space, enter, punctuation at minimum
  - desired punctuation: ():.,!?”-’
- Can interface with text-to-speech programs
  - Generally usable for standard text entry as well
- Can be used efficiently
  - Someone with one-handed use can type about 30 WPM
- Can use mouse and keyboard simultaneously
- Low Latency
  - key-press to USB latency of about 30ms
- Durable
  - Will not break if dropped from the height of a table
- Costs less than what is currently on the market
  - $200 at max
- Open source and reproducible
  - Good documentation and easy to obtain components

# Technical Challenges

- Physical Layout
  - Effectively encoding 74 characters into a smaller number of keys
  - Simultaneous mouse and keyboard input that don’t interfere with each other
  - Maintain accessibility and typing speed while doing so
- Low Latency
  - Sensors for key presses must be fast enough
  - software must  be efficient
- Encase the raspberry pi for durability
  - Creating a structure that is durable and easy to use
- Reproducibility
  - Ideally all parts used are easy to obtain and not prohibitively expensive

# Solution Approach

- Raspberry pi with Python scripts
- USB to connect to Raspberry pi to PC
- Allow us hardware and software flexibility to solve different problems
- Mechanical keyboard “switches”, because capacitor keyboard can be finicky
  - Buy switches, make custom mount
- Foot operable switches
  - Shift key
  - Mouse



# Testing Metrics

- For verification, we test with one hand and one foot
  - Could get a control group
  - How long does it take to get tired? How tired after five minutes?
- 30 WPM is normal for child unaccustomed to keyboard
  - WPM can be tested using an online tool
- Latency can be tested using software

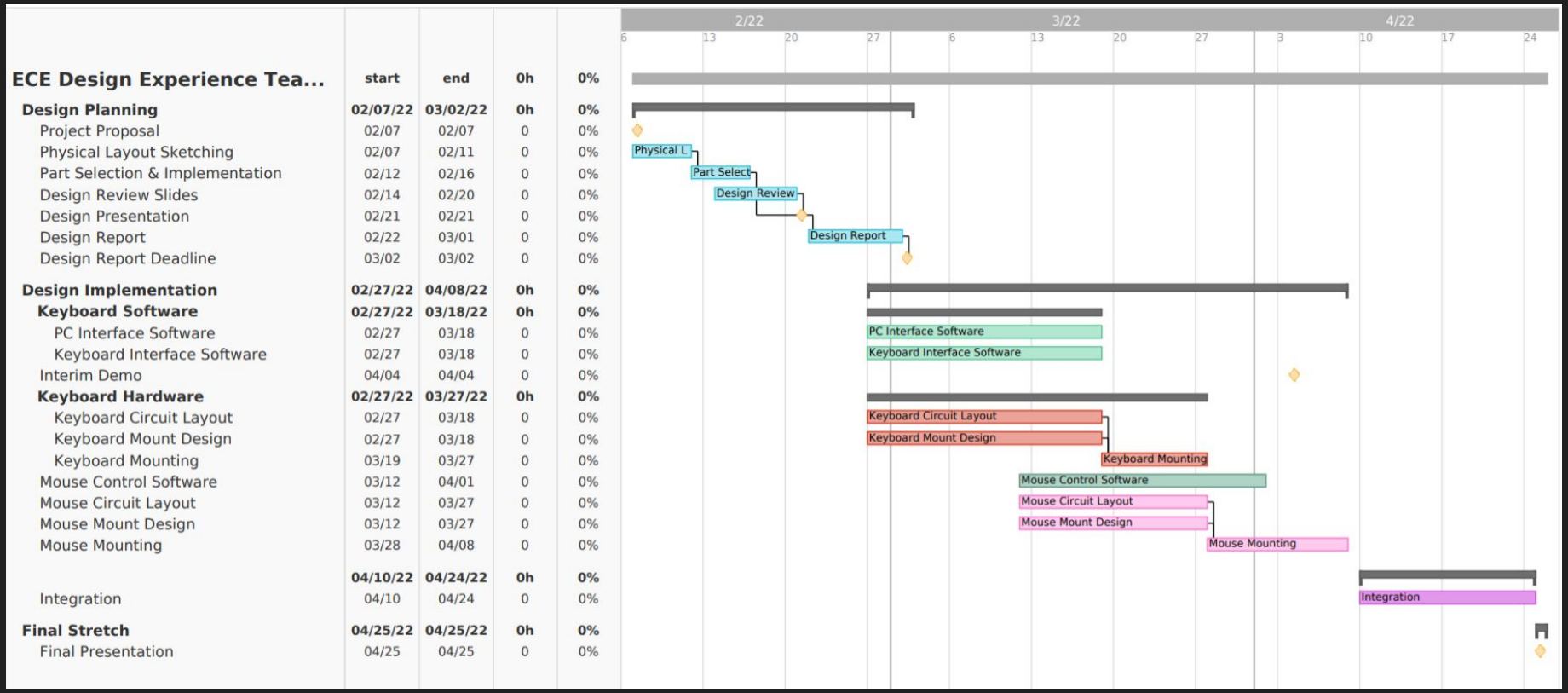
# Validation of Use Case

- Find someone with disability to test design
- Backup plan to test with CMU students
- Survey after use to measure Metrics
  - Comfort & ergonomics
  - WPM measurements
  - Common error avoidance

# Tasks & Division of Labor

- Software  - JT, CA
  - Receiving key signals from I/O and sending to external computer via USB
- Designing hardware - CA
  - Keyboard circuit design - JC, JT
  - Mount for Keyboard - CA
  - Mouse circuit design -JC, JT
  - Mount for Mouse - CA
- Build and Integration

# Schedule



| ECE Design Experience Tea... | start | end | 0h | 0% |
|---|---|---|---|---|
| **Design Planning** | **02/07/22** | **03/02/22** | **0h** | **0%** |
| Project Proposal | 02/07 | 02/07 | 0 | 0% |
| Physical Layout Sketching | 02/07 | 02/11 | 0 | 0% |
| Part Selection & Implementation | 02/12 | 02/16 | 0 | 0% |
| Design Review Slides | 02/14 | 02/20 | 0 | 0% |
| Design Presentation | 02/21 | 02/21 | 0 | 0% |
| Design Report | 02/22 | 03/01 | 0 | 0% |
| Design Report Deadline | 03/02 | 03/02 | 0 | 0% |
| **Design Implementation** | **02/27/22** | **04/08/22** | **0h** | **0%** |
| **Keyboard Software** | **02/27/22** | **03/18/22** | **0h** | **0%** |
| PC Interface Software | 02/27 | 03/18 | 0 | 0% |
| Keyboard Interface Software | 02/27 | 03/18 | 0 | 0% |
| Interim Demo | 04/04 | 04/04 | 0 | 0% |
| **Keyboard Hardware** | **02/27/22** | **03/27/22** | **0h** | **0%** |
| Keyboard Circuit Layout | 02/27 | 03/18 | 0 | 0% |
| Keyboard Mount Design | 02/27 | 03/18 | 0 | 0% |
| Keyboard Mounting | 03/19 | 03/27 | 0 | 0% |
| Mouse Control Software | 03/12 | 04/01 | 0 | 0% |
| Mouse Circuit Layout | 03/12 | 03/27 | 0 | 0% |
| Mouse Mount Design | 03/12 | 03/27 | 0 | 0% |
| Mouse Mounting | 03/28 | 04/08 | 0 | 0% |
| | **04/10/22** | **04/24/22** | **0h** | **0%** |
| Integration | 04/10 | 04/24 | 0 | 0% |
| **Final Stretch** | **04/25/22** | **04/25/22** | **0h** | **0%** |
| Final Presentation | 04/25 | 04/25 | 0 | 0% |

# Final Thoughts

- There is a large number of solutions in the accessibility space
- Most designed to be applicable to the largest number of people
- Our design seeks to focus on a smaller area
  - People with spastic hemiplegia
  - Is the most common form that Cerebral Palsy takes
- Goal is to make a better input device for this group instead of a broadly applicable one
- Ideally it will be open source
  - Can be recreated cheaply or potentially modified for more specific use cases