



# Developing an Accessible Calculator for the Visually Impaired

Abhinav Bapanapalli, Amith Chintalapati, Anush Shah, Henry Liu  
 Advisor: Dr. Kevin Crowthers, Ph.D.



## Problem

Most calculators are too reliant on visual displays and not designed to accommodate those who are visually impaired. Current calculators that have been designed to be accessible to those who are visually impaired do not have the functions present on many scientific calculators that are required by many users, especially students.

Requirement Number	Type	Level	Description
1	Functional	1	The device inputs shall be labeled using braille.
2	Client: Functional	1	The device shall have the computational requirements of a graphing calculator such as a TI-84.
3	Functional	1	The device shall contain either a braille display or text-to-speech capabilities to communicate the answers.
4	Functional	1	The device shall contain alphabetical keys in addition to numerical inputs.
5	User	1	The device will not require the user's vision.
6	Physical	1	The device shall weigh no more than 2.5 kilograms.

## Methods:

### Keyboard Caps Through OpenSCAD and OnShape:

- Used an open-source editable script to generate keycaps of a specific size for each letter, number, and operation key
- Exported the keys as .stl files to be rendered in OnShape for dimensional analysis and print preparation

### Calculator Script Using Wolfram and gTTS:

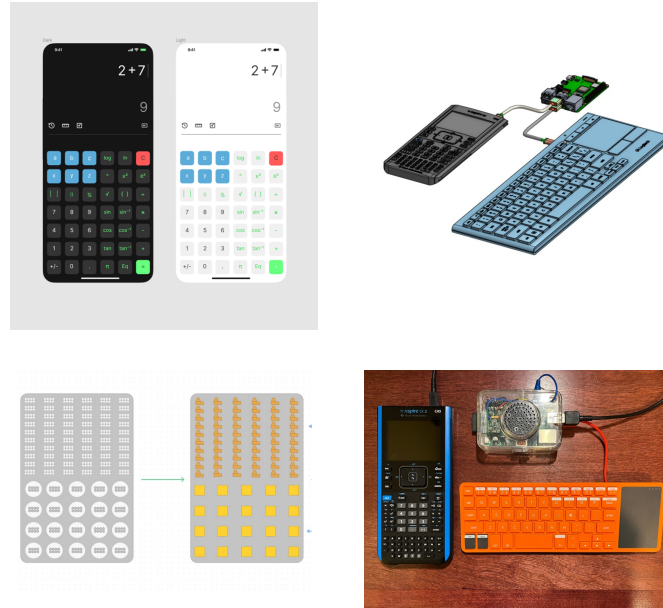
- Raspberry Pi Model 4 B
- Wolfram used as computational software for calculations
- gTTS used for text-to-speech functionality



## Required Materials

- Wired Keyboard (COLIKES USB 2.0 wired keyboard with number pad)
- Raspberry Pi Model 4 B - GB RAM
- 3D printed braille keycaps for keyboard

## Designs



## Final Design



## Design Studies

**Design Study #1: Testing the New 3D-printed Braille Keys**  
**Hypothesis:** Regardless of the input braille key pressed, the returned result to the Raspberry Pi from the keyboard will respectively remain the same and the braille will be clear.  
**Results:** Not all keys have conventional braille symbols. For visually impaired persons, having convenience in finding and recognizing the keys is of the utmost importance due to the reliance on these braille bumpers.

**Design Study #2: Testing the Value of the Output**  
**Hypothesis:** Regardless of the mathematical statement inputted, the returned result to the Raspberry Pi from the keyboard will respectively remain the same and reflect the accurate and precise result of the calculation  
**Results:** The results of the testing for the final prototype show that the algorithm for calculating the output with any given input functions as intended

**Design Study #3: Testing the Text-to-Speech of the Output**  
**Hypothesis:** Regardless of the mathematical statement inputted, the returned result to the Raspberry Pi from the keyboard will be spoken to precision and match the intended calculated result.  
**Results:** The results of the testing for the final prototype show that the text-to-speech feature, with any given input functions, works as intended

## Conclusions

**What Works for the Final Prototype:**

- Assuming the input information is correct, the software can return the correct answer to the calculations or functions
- The text-to-speech software can successfully deliver most general answer cases
- The braille on the keycaps are easily recognizable and the keyboard successfully functions as an input device to the calculator
- The device can function completely on startup and operates without the need for a visual display

**What Does Not Work for the Final Prototype:**

- Not all text-to-speech cases have been tested, and there may be additional outputs from the Wolframscript software, which may result in output errors
- The keyboard redirecting shortcut functionalities have not yet been implemented completely, meaning the functionalities must be inputted manually
- Any syntax errors in input are not automatically resolved, instead requiring the user to re-input the information

**Future Work**

- Revise prototype with a special Raspberry Pi-integrated keyboard
- Improve the efficiency of algorithm and complete bug testing
- Implement Bluetooth compatibility
  - Airpods compatible
  - iPhone/Mac compatible