

Developing an Accessible Calculator for the Visually Impaired

Abhinav Bapanapalli, Amith Chintalapati, Anush Shah, Henry Liu

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.

Final Prototype: The final prototype consists of a modified keyboard containing braille keycaps connected to a Raspberry Pi, which would be used to perform calculations and deliver the output to the client. The braille modification allows for visually impaired clients to utilize the keyboard functionality to a complete degree and input information into the Raspberry Pi. The input data is then transferred to the Wolframscript software, which performs the calculations and delivers the output to the main Python script, which converts the Wolframscript output to readable data. The terminal finally connects this data with the eSpeak software, which uses text-to-speech to deliver an audible output through the Raspberry Pi speaker.

Required Components

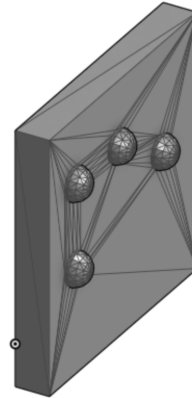
To construct the final prototype, the following hardware is required:

- Wired keyboard ([COLIKES USB 2.0 Wired Keyboard with Number Pad](#))
- Raspberry Pi ([Raspberry Pi Model 4 B - 1 GB RAM](#))
- 3D printed braille keycaps for keyboard (see CAD models below)

In addition, the following software and packages are required:

- Python packages
 - OS (used to connect Python script to terminal)
 - Pynput and Keyboard (used to extract and analyze keyboard input)
- Software
 - OpenSCAD (3D CAD software used to create braille keycaps)
 - OnShape (software used to edit and print braille keycaps)
 - Visual Studio Code (software used to create and test Python and Bash scripts)
 - eSpeak (software used to convert text to speech)
 - Wolframscript (software used to perform calculations)

Images



```
1 import os
2 from gtts import gTTS
3 from pynput import keyboard as keyboard
4 import keyboard as kb
5 import pyautogui
6
7 # Variables
8 global inputText
9 inputText = ""
10
11 while True:
12     inputLine = input()
13     os.system("wolframscript -code '"+inputLine+"' > /home/amithc/Desktop/output.txt")
14
15     f = open('output.txt', 'r')
16     inputText = f.read()
17
18     inputText = inputText.replace("-", " equals ")
19     inputText = inputText.replace("-", " minus ")
20     inputText = inputText.replace("*", " times ")
21     inputText = inputText.replace("Sqrt", " square root of ")
22     inputText = inputText.replace("^", " to the ")
23
24     inputText = inputLine
25
26     os.system("espeak '"+inputText+"' 2>/dev/null")
27     print(inputText)
```

Top Left: Final prototype hardware; **Top Right:** Example CAD model of a braille keycap;
Bottom: Python script used to connect hardware and software components

Instructions for Use

1. Immediately following Raspberry Pi power supply, the device may be powered on by pressing the switch attached to the power cable.
2. Following bootup, wait for the script to load. After the system is loaded and ready to use, the speaker will confirm and deliver the message "Calculator Ready."
3. After the ready message has been received, the user may begin to input any calculations or functions they would like to perform. After the input is complete, the user presses the enter key and waits for the the speaker to return the output.
4. The user may repeat step 3 as many times as necessary, waiting for the device to output the previous result before each new calculation.
5. After the user is done performing calculations, either the script can be turned off by typing "end" into the device or the device itself can be shut down using the power switch. When the user would like to use the device again, they should either power on or restart the device through the power switch.

Safety and Maintenance: The battery of the device should remain charged or at a higher battery level. At minimum, the battery used for the device should be charged at least monthly. In addition, the wires used in the device, primarily those attached to the Pi not including the keyboard wire, should remain closed and inaccessible as to prevent accidental disconnections. Any damaged or malfunctioning wires should immediately be reported or fixed as to prevent any safety hazards.