Project Title: **Vision++**

Team: *Enrique Flores Medina, Tianyi Liu*

EECS 149/249A Project Charter, Fall, 2022

Architecture drawing:

The following architecture drawing shows the complete scheme and reach of our project. The connections made with solid lines are wired connections, while the dotted lines are wireless network connections. In bold letters are shown the data types for all the transmitted information.

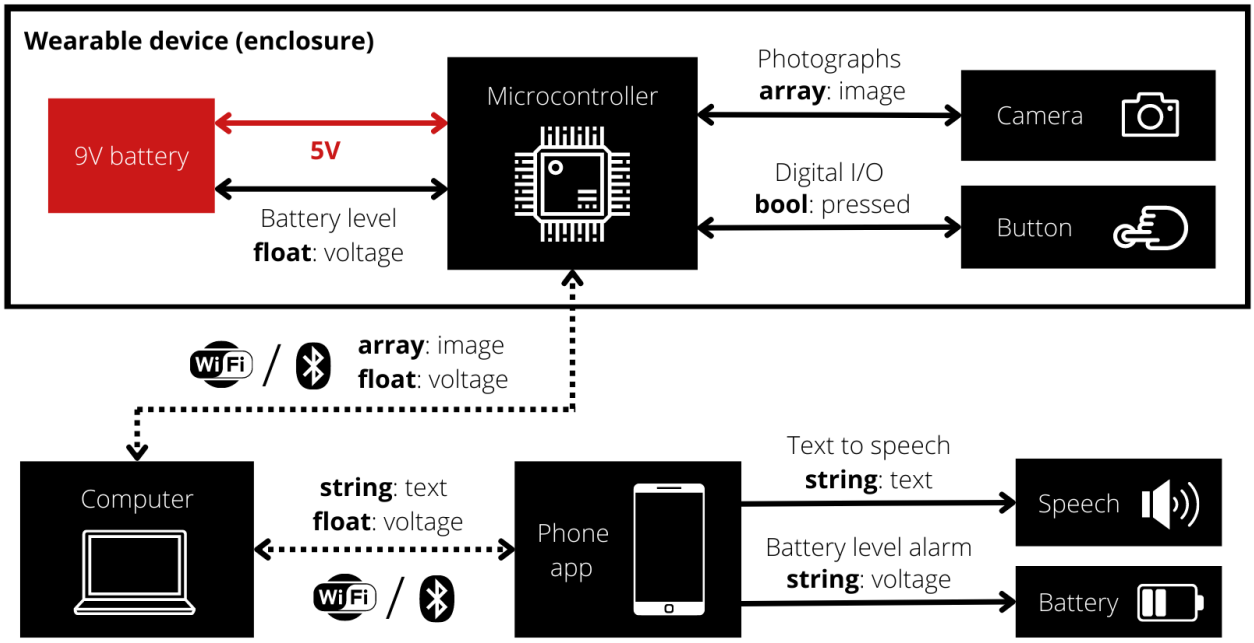

Figure 1. Architecture drawing of the project

Progress so far:

Made an initial sketch of the wearable device. This helped us obtain a general overview on how to make the design ergonomic and the dimension limits we will be working with. The sketch is shown in figure 2.

A first draft for the PCB design that will be contained in the enclosure of the wearable device was also made. This includes all the parts that are inside the wearable device rectangle in the architecture drawing of figure 1. The CAD design, as well as the schematic for the circuit, are shown in figure 3.

Also drew the state machine that models the behavior of our system. It can be found in the GitHub repository.

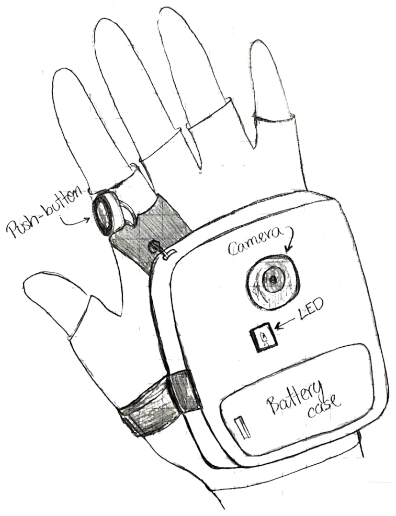

Figure 2. Initial sketch of wearable device.

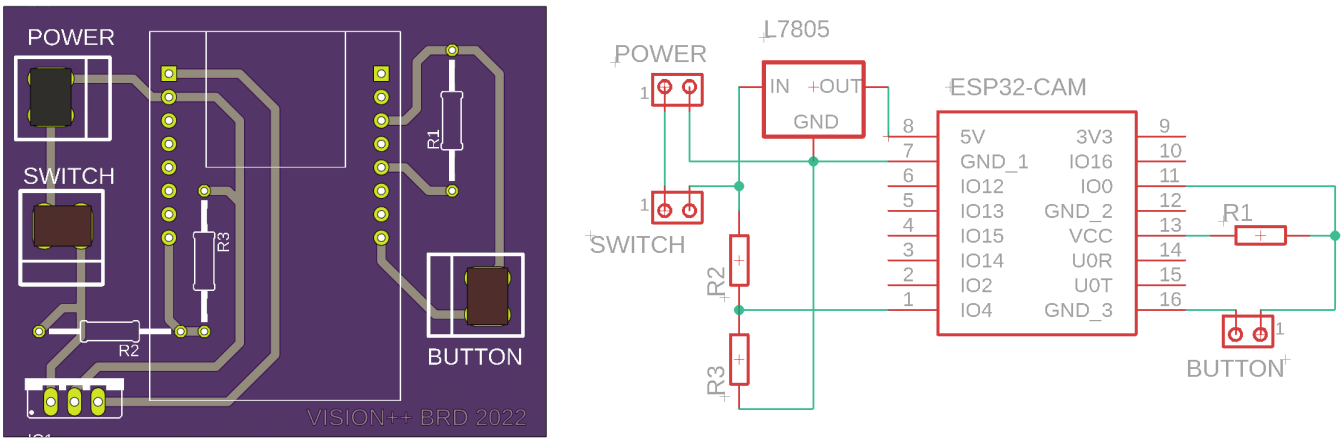

Figure 3. CAD and PCB schematic for the wearable device.

We started testing with MATLAB to read the text using the OCR function. More work is needed to process the text and turn it into a string that is easy to read, but we now have the initial code to read text from images, and to also turn it into speech using a third party connected to MATLAB. To add to this, we started developing an app that will turn the text from the images into speech. (All the figures and files for this section are on GitHub).

Finally, we purchased and requested all the necessary components for the project. The full list is shown in resources.

Goals:

Our goal is still to turn any text into speech, however, our deterministic state machine now only receives one sensor input, which is the button signal. This button will instruct the camera to take a picture, and send it to a computer. Then, the computer will read the text in the received image, and, instead of turning it into speech right there, it will send it to a smartphone app. Finally, the app will convert the aforementioned text into speech that can be heard with either headphones or a speaker.

Resources:

We will use the following libraries for our ESP32 development board, using an arduino IDE:

1. To build a web server and upload the images:
    a. https://github.com/me-no-dev/ESPAsyncWebServer
    b. https://github.com/me-no-dev/AsyncTCP
2. To take, process, and send the images through wifi
    a. WiFi.h, Esp_camera.h, Esp_timer.h, and Img_converters.h

The pinout reference for the ESP32 chip, which the ESP32-CAM development board uses: https://randomnerdtutorials.com/esp32-pinout-reference-gpios/

The MATLAB OCR function is part of MATLAB's pre installed libraries, and our smartphone app will be developed using App Inventor:

https://appinventor.mit.edu/

The parts and mechanical assembly tools we need:

1. 9V battery, and clip
2. L7805CV Voltage regulator 5V
3. ESP32-CAM development board with FTDI programmer
4. Momentary push button
5. Male and female pin headers
6. Three 1kΩ resistors
7. Three 5mm 2 pin screw terminals
8. Blank PCB board (at least 60mm x 50 mm)
9. Polymer glue (to attach the PCB to the enclosure)
10. Velcro tape
11. 3D printer
12. PCB printer
13. Gym or exercise glove

Updated schedule:

- November 6: Milestone 1 - Present architecture drawing, conceptual sketch, and PCB design for the wearable device.
- November 14: Reliable MATLAB script to read text from images taken with the embedded camera. Code to send images from the ESP32 to a web server. Print the PCB and finalize the enclosure CAD.
- November 21: Midterm 2, 3D print enclosure.
- November 22: Code to run MATLAB script when an image is uploaded to the web server, and retrieve said image. Finalize design of smartphone app.
- November 23: Milestone 2 - Showcase progress until this point.
- November 28 - December 2: Milestone 2 Meetings
- December 7: Smartphone app to read the text files that the MATLAB script outputs, and turn it into speech. Finalize details on the rest of the project.
- December 15: Project Poster/Demo Expo
- December 16: Peer Evaluations, Project report

Risk and Feasibility

There are many unknowns. First, by sending the image from the microcontroller to the computer, some packets could be lost. Therefore, the image could lose quality which would significantly reduce our ability to detect the text in it. Another potential issue could be the latency from the trigger of the camera to the speech, which could make the functionality of the product inefficient. If the system consumes a lot of power, and the hardware components are not ergonomic for the user, the experience would also be worsened. The quality of the ESP32-CAM camera is a big factor in determining if we can reliably read the text or not. Finally, the time that the hardware takes to arrive is also important, since it determines how much work can be done.

Link to GitHub:

https://github.com/tianyi-liu-fr/project_ee149_ucBERKELEY