WECEGlasses

Smart glasses by UIUC's Women in ECE

We envisioned creating a pair of glasses that would implement easy access to the most important information that you would need from your phone, without the need to actually have to use your phone to get this information. This implementation began by focusing on 3 aspects of our device: the Hardware that would be involved in the implementation of the circuit, that integrates our various components together, the physical design that would be the physical prototype that is displayed in front of you, and the app, which required the extraction of code from different sources cooperating together to actually execute the functions.

In Developing the Hardware:

A very early decision in the design process had to do with choosing the microcontroller that would be the brain of our project. The microcontroller we used would need to be compact in order to fit within the glasses' form factor, would need to support bluetooth in order to connect to our app, and provide an avenue to mobile powerin, all while ideally still being straightforward to prototype on. The two main options we considered were the Arduino Nano 33 IoT and the Sparkfun Thing Plus - ESP32 Wroom–the Arduino for its extensive hobbyist community support and the Thing Plus for its onboard charging circuit, among other things. While certainly not without its challenges, in the end, the ESP32 afforded us more compute power, a simpler, safer, more compact circuit, and an all-in-all similar development experience at a lower cost than an equivalent Arduino setup.

Button press

We used the Arduino IDE to create the code for detecting button presses. The program is able to differentiate between short, long, and double presses. Distinguishing between different button presses allows for multiple functions to be performed on the glasses. We tested the program through a microcontroller development breadboard - consisting of a microcontroller, an OLED display, and a push-button. Several aspects had to be taken into account such as the time between button presses, how long the button was pressed, how long the button was released, and so on.

Bluetooth:

One of the main reasons we picked ESP32 over Arduino IDE was its built in bluetooth functionality. Initially when two devices know nothing about each other, one device sends out the inquiry request, and any device listening for such a request will respond with its address and other information. Once these devices are paired, they can start sharing data. At the software end, our code consists primarily of call back functions. These are functions that respond whenever they are called upon. When an on-write signal is sent to the microcontroller, it parses through the data identifying which screen the instruction corresponds to. Once this is done, the data sent from the phone is displayed on the screen of the microcontroller. The connection between the two devices allows the device to keep checking if a button is being pressed or any specific instructions are being given.

Screen Handling:

The screen is an external component that we have connected to our circuit. The bluetooth provides a mechanism for the software to interact with the hardware. The phone sends a signal to the microcontroller which parses through the data and identifies what to display on the screen. When turned on, our screen is initialized with a start up animation.

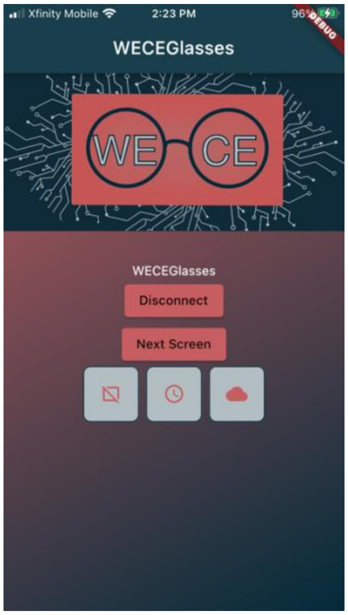
As For The Physical Design:

In building our glasses, we prioritized 3 factors: simplicity, low-cost and wearability. Our goal was to have a sleek design with few externally visible components. Working off a pair of sunglasses, the display is based on a reflection of a small OLED display which is placed at angle for optimal visibility onto the lens.We utilized encapsulation when designing the 3D printed chassis for safety purposes and to hide the components. Encapsulation also allows us to expand our design to create the component as a removable component which allows users to attach the component to their own glasses.

App Development:

Lastly, we have the App. The app serves as a platform for user interface. The app was developed using flutter. We decided to use flutter over other platforms due to flutter’s capability of developing on both IOS and android systems. The software development for the app was split into two main components. First we have the main app functionality which allows for users to select what features they want displayed on the glasses, and to connect and disconnect from bluetooth. Secondly, as a part of app development we developed the features for the glasses. The system for managing and developing features was done using abstraction which for a smooth and easy method for addition of features.

The glasses have three features: time and date, weather, and a game. The time screen displays the current date and time in month day and year order. The weather displays the temperature at your current location. The weather was developed using APIs in order to retrieve what the current temperature is at your location. Lastly the game is inspired by flappy bird and the google dinosaur game. It is played by pressing the button on the side till the player hits an obstacle.


Xfinity Mobile
2:23 PM
96%
DEBUG
WECEGlasses
WE CE
WECEGlasses
Disconnect
Next Screen