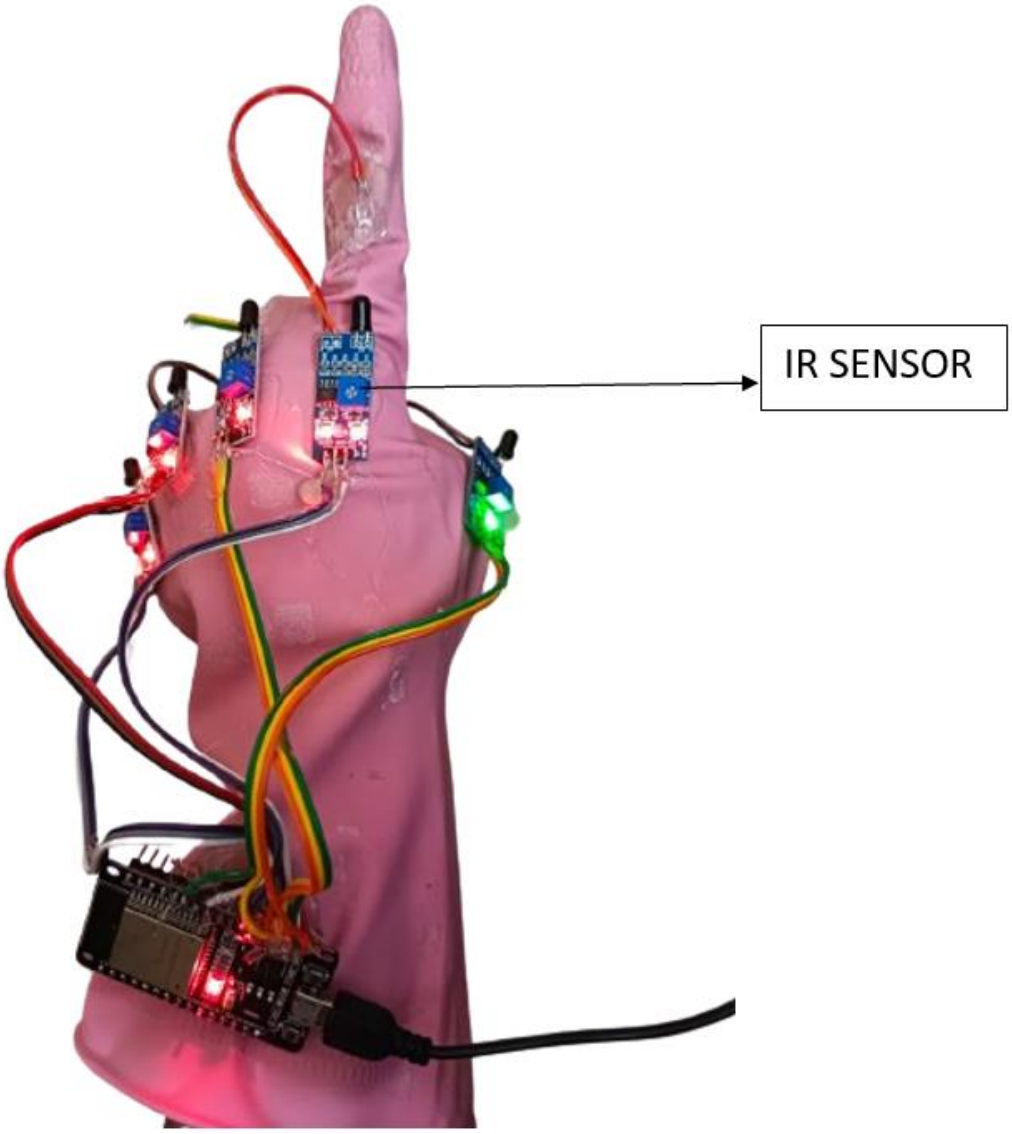
MODULES

MODULE 1 : DESIGNING A GLOVE CAPABLE OF DETECTING SIGN LANGUAGE.

IR sensors are thin, flexible devices that change their resistance when bent. The first step is to integrate the sensor on the rubber glove. When integrated onto gloves, these sensors serve as a means of capturing and quantifying the bending or flexing movements of the wearer's fingers. By strategically placing IR sensors along the length of each finger, including the thumb, index, middle, ring, and pinky fingers, the glove can accurately track the individual bending angles of each digit.



IR SENSOR INTEGRATED WITH GLOVE

MODULE 2: INTERFACING THE SENSOR WITH ESP-32

1. First we connected one leg of the IR sensor to 5V and the other to an analog pin (e.g., A0) on the ESP-32.

2. Then we added a LED from the IR sensor .

3. As the IR sensor bends, it comes in contact with the photodiode on the sensor, altering the voltage at the analog pin.

4. And we incorporate the sensor readings into our project later experimented with different configurations and code adjustments to optimize sensor performance.

MODULE 3: CONNECTING FIREBASE TO MOBILE APPLICATION FOR SPEECH

1. **Set Up Firebase Project**:

- Go to the Firebase Console (console.firebase.google.com) and create a new project.

- Follow the prompts to set up your project, including choosing a project name and enabling Firebase services like Realtime Database.

2. **Add Firebase to Your Android/iOS App**:

- In the Firebase Console, add your Android/iOS app to the project by providing the package name/bundle identifier.

- Download the google-services.json (for Android) or GoogleService-Info.plist (for iOS) configuration files and add them to your app's project directory.

3. **Integrate Firebase SDK**:

- Follow the Firebase documentation to integrate the Firebase SDK into your Android/iOS app. This usually involves adding the Firebase dependencies to your project's build.gradle (for Android) or Podfile (for iOS) and initializing Firebase in your app's code.

4. **Implement Firebase Realtime Database**:

- Set up the Firebase Realtime Database rules in the Firebase Console to control access to your database.

- In your app, implement the necessary code to read and write data to the Realtime Database. This includes handling user sign language input and storing it in the database.

5. **Ensure Both Devices Are on the Same WiFi Network**:

- Connect both the device running your MIT Companion app and the device running your sign-to-speech application to the same WiFi network. This ensures optimal communication between the devices.

6. **Transmit Sign Language Input to Firebase**:

- Implement the logic in your MIT Companion app to capture sign language input from the user.

- Use Firebase SDK to transmit this input data to the Firebase Realtime Database, where it can be accessed by your sign-to-speech application.

7. **Retrieve Data from Firebase in Sign-to-Speech App**:

- Implement the logic in your sign-to-speech application to listen for changes in the Firebase Realtime Database.

- When new sign language input data is detected, process it as needed (e.g., converting it to speech) and provide the output to the user.