

Bluetooth Low Energy Warning System for Android

About

The Bluetooth Low Energy scanner is a proximity warning system designed to be a component for the Aggie oCT COVID-19 contact tracing app developed by NMSU. It monitors for four types of contact risks: immediate danger which is contact with another person that is within proximity of less than six feet, crowd danger which is contact with too many people nearby, continuous exposure which is prolonged contact with same individual for more than five minutes, and repeated exposure which is accumulated close contact with same individual in a 24-hour period.

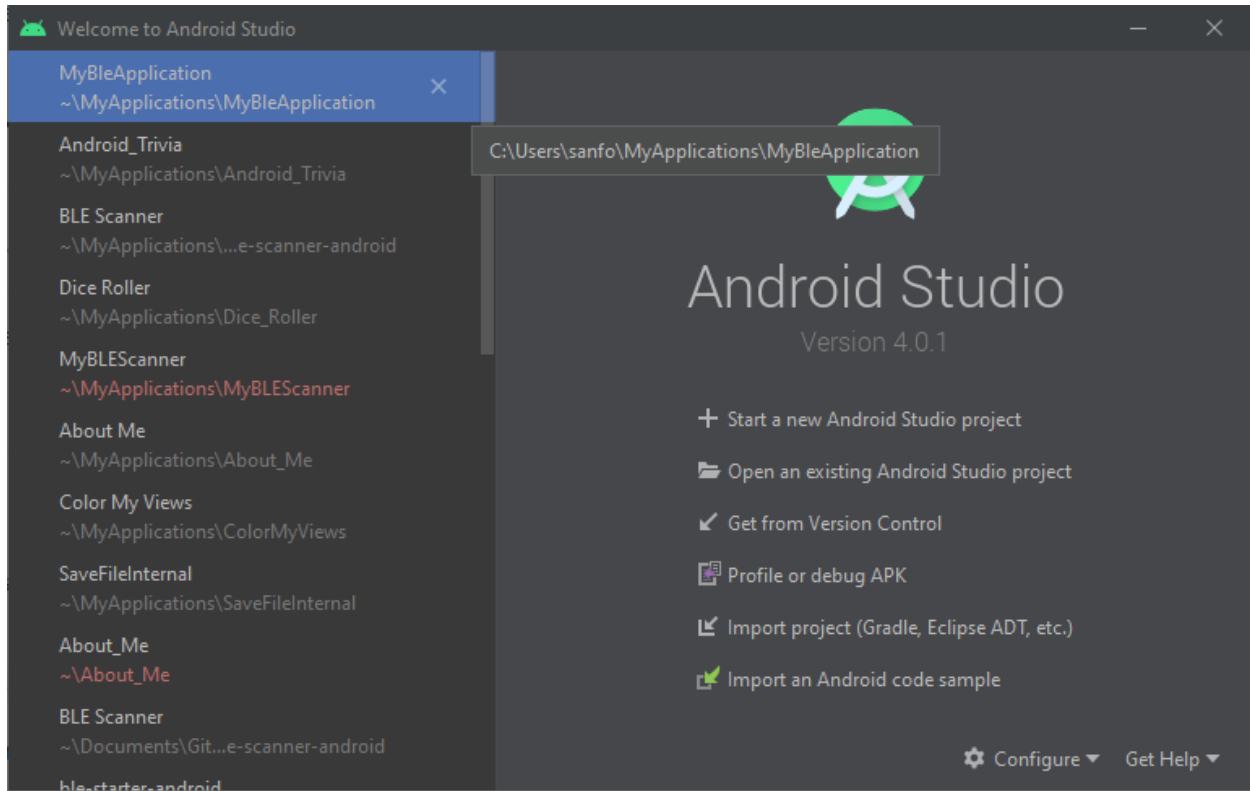
The system scans for nearby devices, measures their proximity from a combination of signal strength and measured power, displays the scan results, and monitors all devices for the four warning conditions. Power consumption is minimal due to the low energy nature of BLE.

Installation

Dependencies

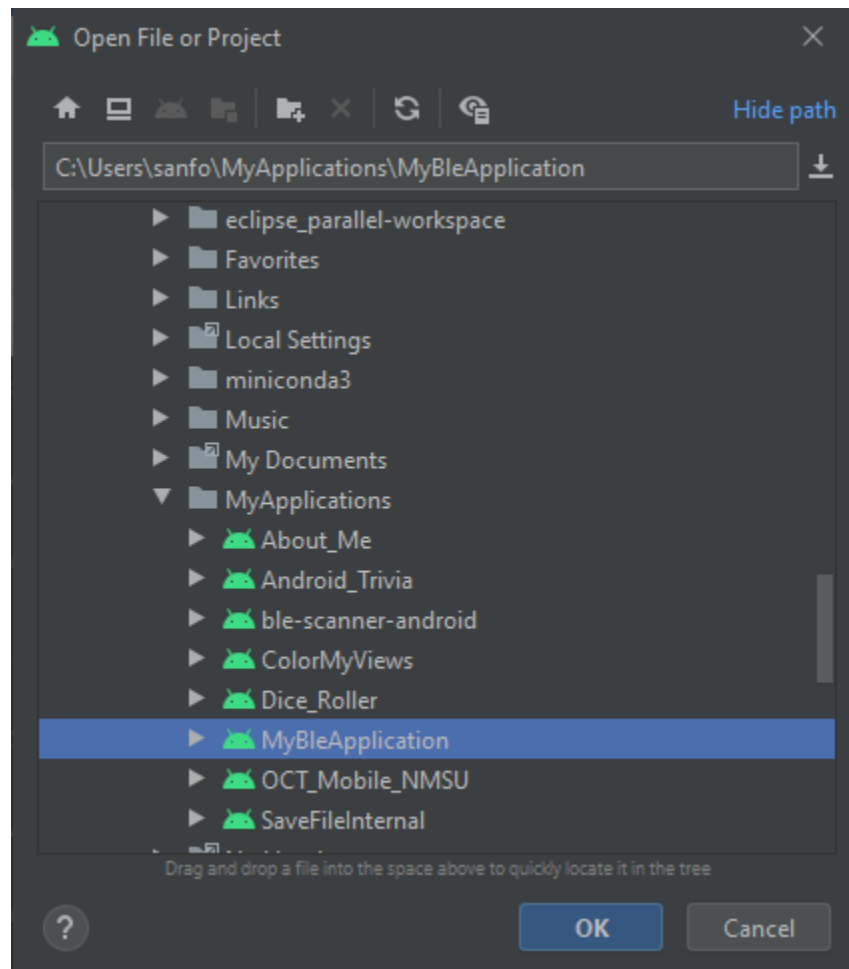
- Download and install [Android Studio](#). Follow the instructions from the developers site.
- Save the MyBleApplication folder to a desired location on your local computer. On the Android Studio Welcome screen, select Open an existing Android Studio project.

Bluetooth Low Energy Warning System for Android



- In the Open File or Project dialog, navigate to the project folder

Bluetooth Low Energy Warning System for Android

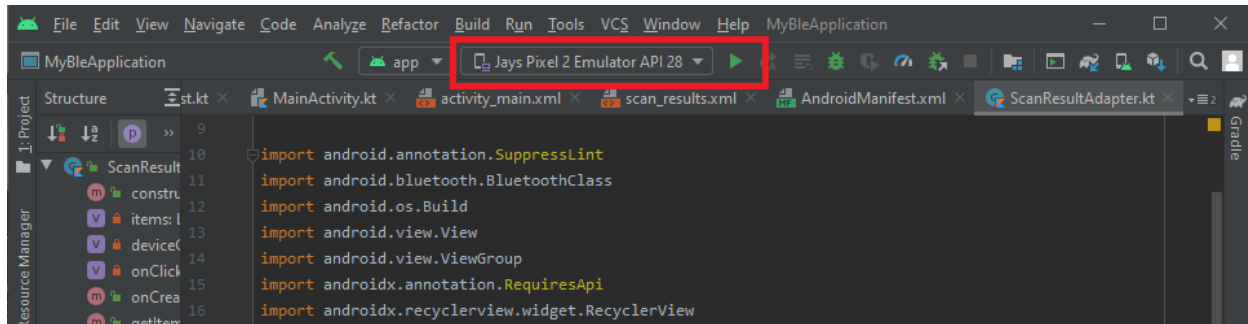


- Select OK
- The file will load. From here, you can browse the project and its files.
- You may need to install Java or other packages to your computer for Android Studio to compile and run the program. If so, Android Studio will detect any dependencies and prompt you to download and install the packages automatically.

Testing

- This application cannot be tested on an Android Virtual Device. You will need to connect an Android device to your computer to test it. Once you have an Android device connected, you will see the option to run the app on the device in a window at the top of the Android Studio frame.

Bluetooth Low Energy Warning System for Android



There are no licenses needed for this app.

Design

The app employs a custom data structure where we created an array of the Linked Node objects that tracked the devices by unique ID as well as their length of contact with the user in real time without requiring access to an outside database. Memory use is kept to a minimum. The data structure is set up so that it can be easily recorded onto the device locally or to a cloud database.

The proximity and time thresholds of the warning system are coded as constants so that the warning system can be employed on any public health authority contact tracing app and can be customized to the local health authority minimum space and time mandates. Theoretically, if a new pandemic were to outbreak that had different spacing and time constraints, this system could be easily employed onto a new contact tracing app tailored to the new public viral safe distance and contact requirements. Or, if the current space and time constraints changes for the local public health authority, then the constants can be updated, and a software update can be pushed to all users keeping them up to date on any changes to social distancing requirements.

If a user is too close to another person, the user will get a warning, "Alert! Someone is too close..." If the user is surrounded by too many people in close proximity, they will get another warning, "Alert! You are in a crowd..." If the user has been close to the same person for more than five consecutive minutes, they will get a warning, "Alert! Continuous contact with same person. Maintain distance." And last, if the user has encountered the same person in close proximity off and on throughout a 24-hour window for an accumulated 15 or more minutes, they will be warned, "Alert! Repeated contact with same person. Stay safe."

The BLE proximity warning system is wholly contained in a single layout. The UI for the Android version consists of a single activity. The main activity is a constraint layout within a coordinator layout. This outer architecture is necessary for implementing the Snackbar

Bluetooth Low Energy Warning System for Android

notifications. Within the constraint layout is the UI. The UI consists of a start/stop button at the top with three recycler views and three image views alternating each other. The three recycler views serve as the three proximity risk zones, threat, caution and safe. The images that split the proximity risk zones are lines to intuitively partition the proximity risks. The image at the bottom serves as an icon to represent the user.

To match the software architecture between the two systems, a device's unique ID is saved as a string. This way, the iOS can save a device as UUID, Android can save a device as MAC. When the code is implemented into the Aggie oCT COVID-19 app, either UUID or MAC can be used, and the code will still work.

Author: Sanford "Jay" Johnston

Contact: Sanford.johnston@gmail.com