Heartbeat Triggered Arduino Safety Device

The main idea of the project is a safety device that automatically starts logging data on your phone based a trigger (typically heartbeat increase). Think of the situation where you are in a car accident, or are threatened by another individual, or you feel afraid because you have got into a dangerous situation, or even are excited to meet your daughter after a long period of time. In this situation, maybe you want your location info to be saved, or a distress signal transmitted, and the proceedings automatically recorded without it being obvious that these actions are taking place. The goal of this project is to make this kind of feature possible using Arduino and your Android phone.

Additionally, a small amount of data could be shared automatically peer-to-peer based on the trigger (see dating example below)

The data collection policy will keep data in a database for 1 month, and tag metadata such as heartbeat along with data so filtering can be performed later on this metadata. E.g. you could sort by date, excitement level, etc. using in-app UI.

Components

- Android App (always on even when phone is locked) which will take on some of the sensing responsibilities, logging and data visualization (with filtering support)

- Arduino Circuit Playground with Pulse Monitor or equivalent heart rate monitor

The app consists of a set of Triggers and Actions which are configurable.

Triggers

- heartbeat increase

- list of "safe words" that trigger automatic data collection

One can think of other triggers that could be defined that are health related, such as measuring oxygen content in blood with oxymeter, lack of mobility etc. that are more geared towards health apps. But for this project, the above two should be sufficient.

Actions

Below are some sample actions. Note that not all of these actions are safe to run always, so some discretion is required based on scenario. We should only implement a subset of these actions. A guiding principle we need to follow is that it should not be an expensive operation to run these actions, since the data collection could be noisy. We don't want the user to have to cull unnecessarily data manually.

- Location collection from phone GPS unit and save to local MySQL database and Firebase remote database. Location could be a point sample or a sequence of locations (e.g a route)

- Location transmit to list of recipients via text messaging

- Automatically start recording audio clips of several minutes' length and save to phone

- Distress message to list of contacts

- Automatically text emergency services (e.g. text 911)

- Trigger camera photo capture or video capture

- Create an audible alarm (e.g. car alarm sound, which gets attention from a lot of people)

- Trigger a remote camera.

Execution Plan

Phase 1:

Trigger a set of actions based on heartbeat increase trigger. See above for actions which can be triggered by heartbeat.

Phase 2:

Trigger set of actions based on voice recognition. While heart-rate based trigger is speculative and could have false positives, voic- based trigger is less prone to accidental triggering and can differentiate the type of actions to perform. This is similar to "safe words" people use to encode communication between people. For example, saying "forgot to call the doctor" might seem innocuous, but can be a trigger for the actions above to be performed. Similarly, some other special phrase would unequivocally indicate that you are in danger and trigger appropriate response.

Phase 3:

Peer to peer communication based on triggers: This is what I call the "dating" scenario. Imagine you come across someone you are interested in and your heartbeat increases. You have set up your app to

automatically track this. Perhaps the other person is also running this same app and advertising availability for dating. In such cases, the apps would automatically advertise availability behind the scenes via a small message exchange. Later on, you can go through your encounters on the app and find information that the other person has shared and possibly meet for a date!

While contact info dissemination is a pretty advanced scenario, there are also other scenarios you can trigger peer to peer. However, since we will only be using BLE, the proximity requirement will preclude a lot of actions.

Phase 4 (optional):

Make the trigger mechanism into an interface that can be triggered in other ways than with Arduino/Circuit Playground. You can think of using your Google Watch/Fitbit being the trigger.