

The background of the slide features a decorative graphic consisting of three concentric blue circles of varying sizes. Two thin blue lines intersect at a point, forming an 'X' shape that divides the slide. The circles are positioned in the upper right and lower right areas, while the lines extend from the top left towards the center.

Emergency Response System

CS 6235 Project Report

Emergency Response System (ERS) is a framework that provides fast and efficient real-time processing and dispatch of distress signals to relevant authorities via SMS. An Arduino-based wearable panic button and an Android service running in the background on a phone are used to demonstrate the use of ERS framework.

Nataraj Mocherla & Sandeep Manchem

12/2/2013

Contents

Abstract	3
Motivation.....	4
Objective	5
Features of ERS	5
Architecture	7
Implementation	8
User Interface	9
SMS Format.....	9
Android Service	9
Location & History Map.....	9
Results	9
Lessons Learned	9
Future Work	9
Conclusion	9
References	9

Abstract

Emergency Response System (ERS) is a framework that provides quick processing and dispatch of distress signals via SMS. We use Node.js, an event-driven, non-blocking I/O model perfectly suited for large-scale data intensive real-time applications that run across distributed devices, to implement the backend server that receives an emergency message via SMS, processes it and forwards the augmented message to relevant authorities who can then help the person who sent the distress signal. We have implemented one use-case of this framework wherein an emergency signal is sent from a wearable Arduino-based panic button which is connected to the user's phone via Bluetooth, the lightweight Android service running on the phone in the background then quickly adds location information to the signal and sends an SMS to the backend server. The server processes the SMS and augments it with user information along with human readable location of the user and sends it the relevant Police Authorities. Subsequently, the Android service sends tracking information to the server which plots the location of the phone in real-time over a map which the Police can access and act accordingly.

Motivation

Safety is everybody's concern. It is often the case that when you need help, you may not have the time or it may not be feasible to pick up the phone and dial for help. In case of an emergency such as an impending robbery, a heart attack or a tumble down the stairs it is crucial to contact authorities as soon as possible. This line of thought led us to the notion of a panic button that when pressed, quickly transmits all the relevant information including the user's location to appropriate authorities and/or family and friends, so that help can be dispatched immediately.



Figure 1 : Georgia Tech Emergency Posts

On Georgia Tech campus we find Emergency Alarms installed on specific locations throughout the campus as seen in the image above [Fig.1] that allow people to send an emergency message to Georgia Tech/Atlanta Police, with just a push of a button. However, these posts are not really useful when the student is not in the vicinity of one of them. To solve this problem we propose the idea of a **wearable panic button** to be worn all the time that can be pressed in the blink of an eye and send a distress signal. The idea of a wearable panic button is particularly appealing to those who want to conceal the panic button and send a distress signal without anybody realizing it, for example during an ongoing robbery. The panic button could be part of a bracelet, belt buckle, necktie, or simply kept in a pocket.

Objective

We intend to create a fast and efficient Emergency Response System (ERS) to send distress signals to appropriate authorities with as much ease as possible. We develop a **wearable panic button** that when pressed will send an emergency signal to the user's phone via Bluetooth, which in turn forwards it to a backend server along with the phone's location information in the form of an SMS. The backend server will process the SMS and augment it with user information and send the distress signal with all relevant information to appropriate authorities (Police, Friends/Family etc).

We must also ensure that accidental triggering of the panic button can be corrected i.e., we should be able to cancel an Emergency Signal that was sent unintentionally. To achieve this we need to not only have a mechanism to send a Cancel Signal, but also inform the user that an Emergency Signal was sent so as to initiate a cancel procedure.

The distress signal (SMS) sent to the authorities should be easy to read and immediately actionable i.e., the SMS must contain all the required information to reach the location of the incident and also enable easy identification of the user (victim).

It is desirable to have constant contact between the user and the helper (Police/Friends/Family) once a distress signal has been sent so as to ensure that any updates in the situation can be transmitted quickly and course correction could take place if required.

From the point of view of the helper (Police), an appropriate interface must be provided to cross-check user information and also handle multiple emergencies occurring at the same time.

Features of ERS

The Emergency Response System is primarily a framework for fast and efficient framework for receiving, processing and dispatching distress signals via SMS. We use the framework to implement an end-to-end solution using a Wearable Panic Button and an Android Service running on a phone. Our solution has the following components:

1. **Wearable Personal Security Bracelet (PSB)** [Fig.2] built using an Arduino Lilypad Microcontroller [7] with Bluetooth module [8].
2. **Android Mobile Service** which listens to signals from the PSB via Bluetooth.
3. **Backend servers** to process the incoming panic messages and routing them to the police (GATech or Atlanta police depending on location of crime scene).
4. **Web based visualization** of data for Police.



Figure 2 : Wearable Personal Security Bracelet

We have achieved each of our objectives by having the following features:

1. The wearable Personal Security Bracelet includes a **button** and an **LED light** that will begin blinking upon sending the emergency signal. The LED light acts as a feedback mechanism to the user to indicate that an emergency signal has been sent and can be cancelled if required.
2. A “**double press**” of the panic button will send an *Emergency Signal* and a “**prolonged press**” (press and hold) will send a *Cancel Signal*.
3. A window of 30 seconds is provided for the user to cancel an emergency signal, after which the backend servers will process the signal and forward the message to appropriate authorities.
4. Button press translates into a **Bluetooth Signal** that is transmitted to the user’s mobile which can be within **10 meters** from the panic button thereby removing the constraint of having the phone on person all the time.
5. The Android mobile service is responsible for **determining the location of the device**. This information is sent via SMS to the server in the form of latitude and longitude, which is then translated into an actual address that is easy to read and reach.
6. The Android Service continues to send **Tracking SMS** to the backend servers with location information to help the authorities track the mobile phone and approach it sooner.
7. The web based user interface for the Police enables them to easily visualize the location of incident and subsequent tracking of victim/assailant.

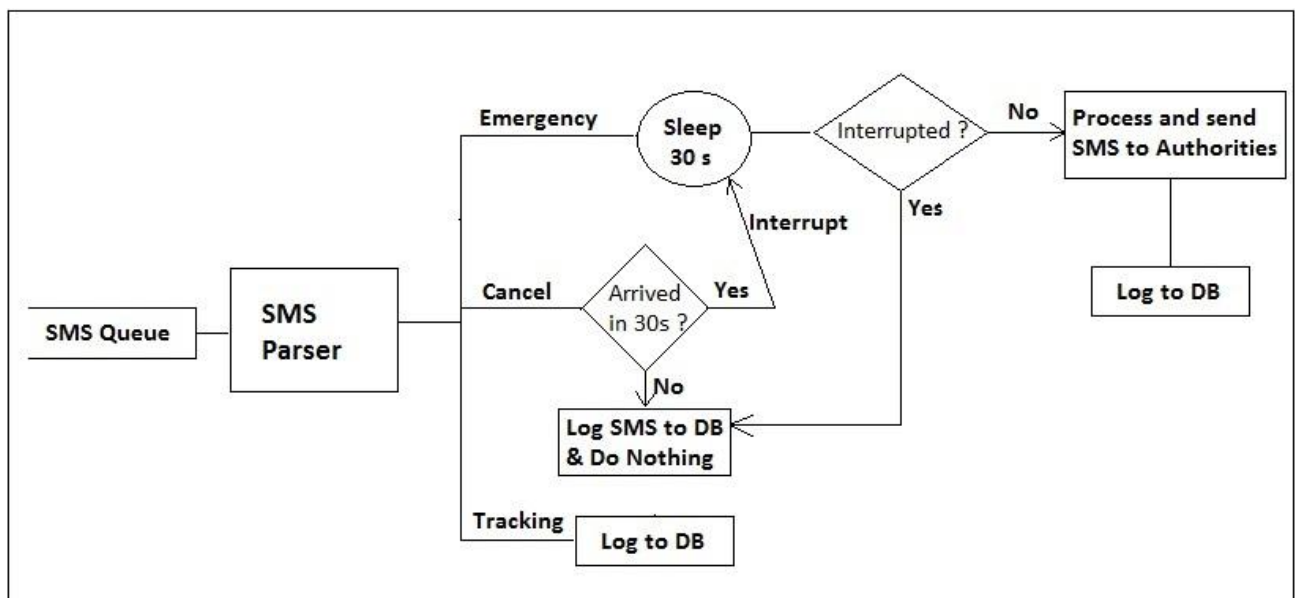
Architecture

Our solution has the below Basic System Architecture [Fig.3].



Figure 3 : Basic System Architecture

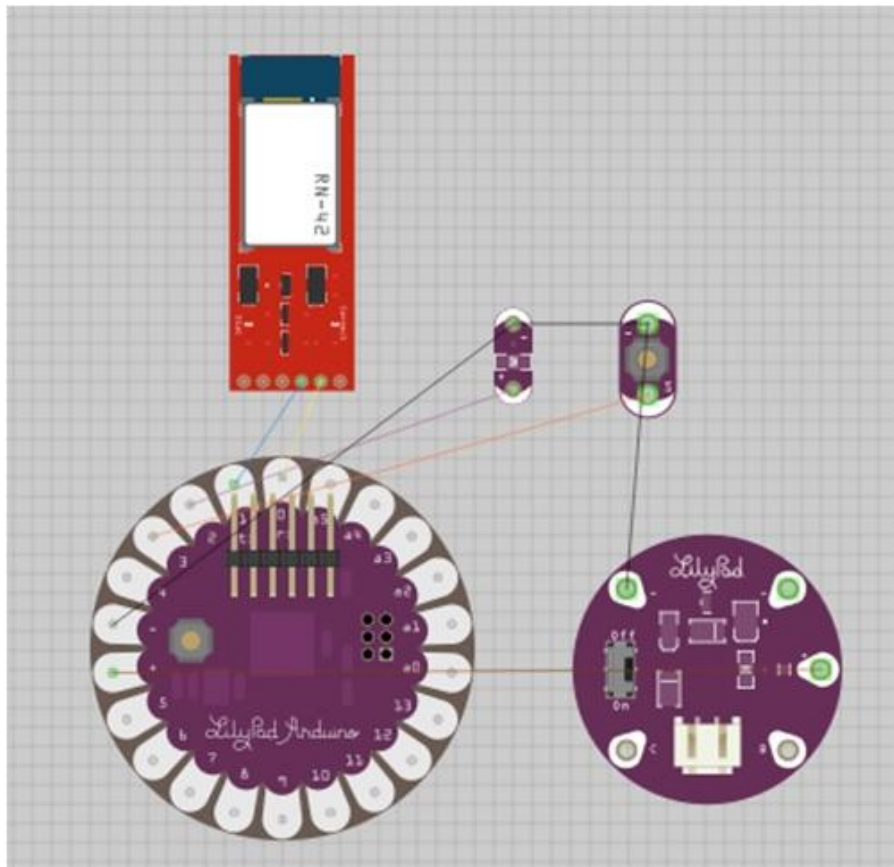
Architecture of the Backend Servers is shown in [Fig.4]



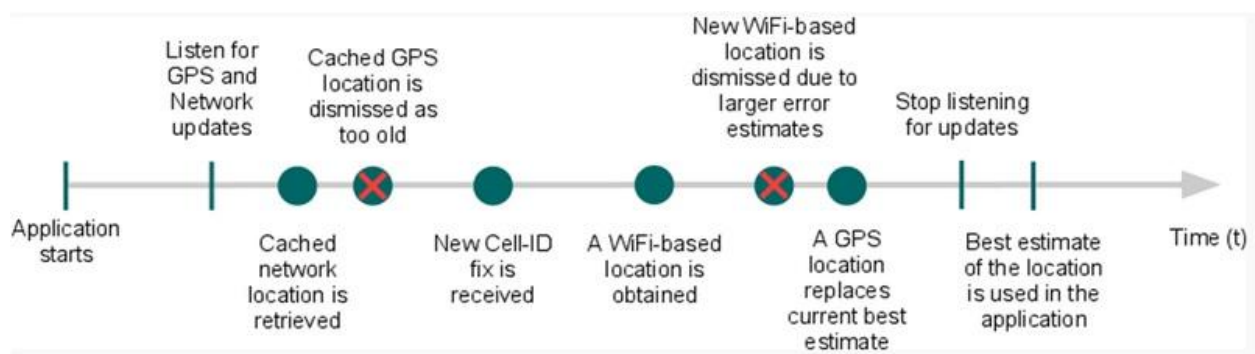
Implementation

[Describe in detail the implementation of each the components, the technology used and the difficulties overcome.]

Arduino Lilypad + Bluetooth module



Android Service



Node.js server – include “contacting correct authority”

MongoDB Database. – different tables and ease and speed of MongoDB.

Node.js webserver.

User Interface

SMS Format

SMS format sent to server and sent to police.

Android Service

Screen shots

Location & History Map

Screen shots

Results

End to end time

Server processing time

Server processing time when multiple signals sent simultaneously.

Lessons Learned

Future Work

Conclusion

References

[7] Arduino. Retrieved 19 Sep 2013 from <http://www.arduino.cc/>

[8] Arduino Bluetooth Module. Retrieved 19 Sep 2013 from
<http://arduino.cc/en/Guide/ArduinoBT>