Lab 2: Bluetooth radio beacons

Overview

Radio beacons based on Bluetooth technology is coming strongly, both within retail, logistics and asset tracking, and with Google's Physical Web project - linking the digital and physical world together. There are several dialects for beacons, both basic beacons only reporting a simple hardware address over and over, to more sophisticated ones like iBeacons (by Apple) and Eddystone (from Google). Smartphones speak bluetooth natively in apps, and in some cases via the Chrome browser. We'll use the open source tool suite **Evothings Studio** in this lab exercise, at is takes some of the heavy lifting out of the equation.

Using the Viewer app adds a few functions, extending what you can do with javascript in a browser, such as:

// Start tracking beacons now!

app.startScanForBeacons();

And that's what you need to do, and just wait for the callback, which is a list of what it's found.

The whole list of plug-ins and APIs is here: https://evothings.com/doc/api/api-overview.html

You will also start with a template project, and adapt it to suit your needs in the group. This could be an app for a small museum, a city guide tour or similar, you can decide for yourselves what the application should be. And since you're doing your final project very soon, this last lab is more of an introduction to new technology than actually being very tricky and hard to do. We will use iBeacons for this lab, not Eddystone.

Protip: You don't need to use the editor that comes with the Evothings Workbench software if you don't want to; Instead of clicking "edit" just press “more” > “files” to reveal where the files are located. Use any editor you like, and every time you save locally, the latest version is sent to your phone via the cloud.

Background

Here is some useful information:

Set up Evothings on your computer and phone

http://evothings.com/doc/starter-guides/evothings-studio-starter-guide.html

If downloaded on a KTH school computer where you don't have root acceess, and you lack installation rights in the Applications folder, just run it directly from your download folder instead.

Mobile development app

- For **iOS**, a popular Viewer is the CGTek viewer: https://itunes.apple.com/us/app/cgtek-viewer/id1135118284?mt=8
- For Android, look for “Evothings Viewer” on Google Play,

Evothings iBeacon starter guide

https://evothings.com/doc/starter-guides/ibeacon-starter-guide.html

Google’s physical web project

http://google.github.io/physical-web/

https://developers.google.com/beacons/

iBeacons by Apple

https://developer.apple.com/ibeacon/

Beacon exercise (8p)

Task

Adapt the template code (found on the bottom of this page) to suit your needs, making a proximity-centric app. Find a theme that you think is relevant. It could be maintenence, tourist info, information for cleaners, leaving bookmarks in the area for other students et cetera. One student group last year made a scary story, a crime mystery where details were added as you traverse the rooms with images and sound!

Details

Locate the iBeacons in the lab halls, right now there are only two in Konsthallen. Please don't mess with them physically or move them around, be good friends and citizens.

If you want to do the lab but don't want to go to the lab halls to make it happen, feel free to emulate an ibeacon with an old phone:

https://play.google.com/store/apps/details?id=net.alea.beaconsimulator&hl=sv

There are in general three kinds of beacons, and the code is written for the ibeacon format (Apple).

You also need to customize the app somewhat, as the app was made with other beacons in mind. Check that the UUID is the same, while the major and minor settings are probably different. (did you read about this in the background, otherwise please do as it is central to the exercise).

The beacons we use have this UUID (from Polish company Estimote).

B9407F30-F5F8-466E-AFF9-25556B57FE6D

The major and minor keys can both be any unsigned number from 0 and 65535, anything you like. WIth real beacons, they're often set by the manufacturer. Often beacons from the same manufacturer all share the same UUID, while the Major and Minor parameters are different, and presumably unique.

There are several apps available, like the LightBlue app for iOS, that can scan for UUIDs, and the Major and Minor of devices. In Evothings Studio, the iBeacon Scan or Estimote Beacon app is the right way to go [http://evothings.com/doc/examples/estimote-beacons.html].

Also, as the beacons need work not only when very close to you. There are three general tiers of distance quotas for a beacon; (1) immediately close (inches away), (2) near (feet) and (3) far (meaning yards, maybe up to 30-55 yds).

Therefore, please change the code (in app.js) from (means beacons have to be super close)

beacon.proximity == 'ProximityImmediate'

... to at least this…. (beacons close OR medium range, a few meters)

beacon.proximity == 'ProximityImmediate' ||

beacon.proximity == 'ProximityNear'

...so you don't have to be so near to get results, and be trampled by other students attempting to solve the same issue.

And yes, find the right UUID, major and minor for the three beacons in the room(s) using the "iBeacon Scan" example in Evothings Studio, or any other Bluetooth Low Energy scanning app. It's important that all these values are correct, otherwise the application won't find the iBeacons and nothing will happen. For iOS, there is an app called **Core Beacons** on the appstore, which is also nice.

The template

This app was initially made for yoga and mindfulness practitioners, the structure is basically in index.html and the action in app.js

Drag-drop the index.html to Evothings Studio after connecting the client. This populates the list in "My Apps" and then press Run to push it to the Viewer.

http://evothings.com/demo/relaxation-beacons/relaxation-beacons.zip