

Team Members

Quality Assurance / Solution Demonstrator: Alex Carter
Other stuff: Blaine Carter

Problem

My son Alex attends a school where the students have some ‘bonus features’, or as the school puts it: “Educating Exceptional People”. There are some students at Alex’s school who tend to wander away.

There are commercial systems available that can notify the administration and/or lock doors when the beacon they are wearing is detected in a hazardous zone, such as leaving the school. The problem is, those systems can be expensive.

Solution

Video

<https://www.youtube.com/watch?v=kio7fDQdQgA>

Code

<https://github.com/bcarter/beaconScan>

Hardware/Software (my setup)

- Hardware
 - 2 Raspberry Pi 3 – built in wifi and bluetooth.
 - 1 Raspberry Pi Zero W – built in wifi and bluetooth.
 - Z-Wave enabled Schlage Connect deadbolt.
 - Z-Stick Z-Wave usb hub.
- Software
 - Raspbian Linux
 - Linux modules
 - bluetooth
 - bluez
 - libbluetooth-dev
 - libudev-dev
 - NodeJS
 - NodeJS Modules
 - bleacon
 - request
 - IFTTT.com
 - Home-Assistant.io
 - Optional Software
 - Oracle Database
 - Oracle Rest Data Services (ORDS)

Installation

1. Install Raspbian Linux on a Raspberry Pi with bluetooth.
2. `sudo apt-get update`
3. `sudo apt-get install bluetooth bluez libbluetooth-dev libudev-dev`
4. `reboot`
5. `cd ~/`
6. `git clone`
7. `cd beaconScan`
8. `sudo npm install`
9. Connect to your DB as sys
10. `@beacon_master_install.sql`
11. create environment variables
 1. SCANNER_ID – The ID for this scanner.
 2. IFTTT_KEY – your account key for IFTTT
 3. ORDS_URI – Example: `http://192.168.0.188:8080/ords/beacon/beacon/alert/`
 4. BEACON_UUID – UUID of the beacon you're tracking.
 5. DEADBOLT_URI – URI for the REST API to unlock the deadbolt.
12. Run the application using `sudo` for access to bluetooth:
`sudo -E node scanBleacon.js`

This will set everything up for the way I have my system designed.

I'm temporarily (see below) using Home-Assistant.io to control the deadbolt through the Z-Stick. In the video you can see the Z-Stick (white usb stick) plugged into the Raspberry Pi on the table next to the computer. I followed the Home-Assistant guides for pairing my devices. Once everything is paired, I use the Home-Assistant.io REST API to lock/ unlock the deadbolt and to send text through the tts (Text To Speech) interface to the Sonos speaker.

To test the system I configured each Raspberry Pi to preform a different action:

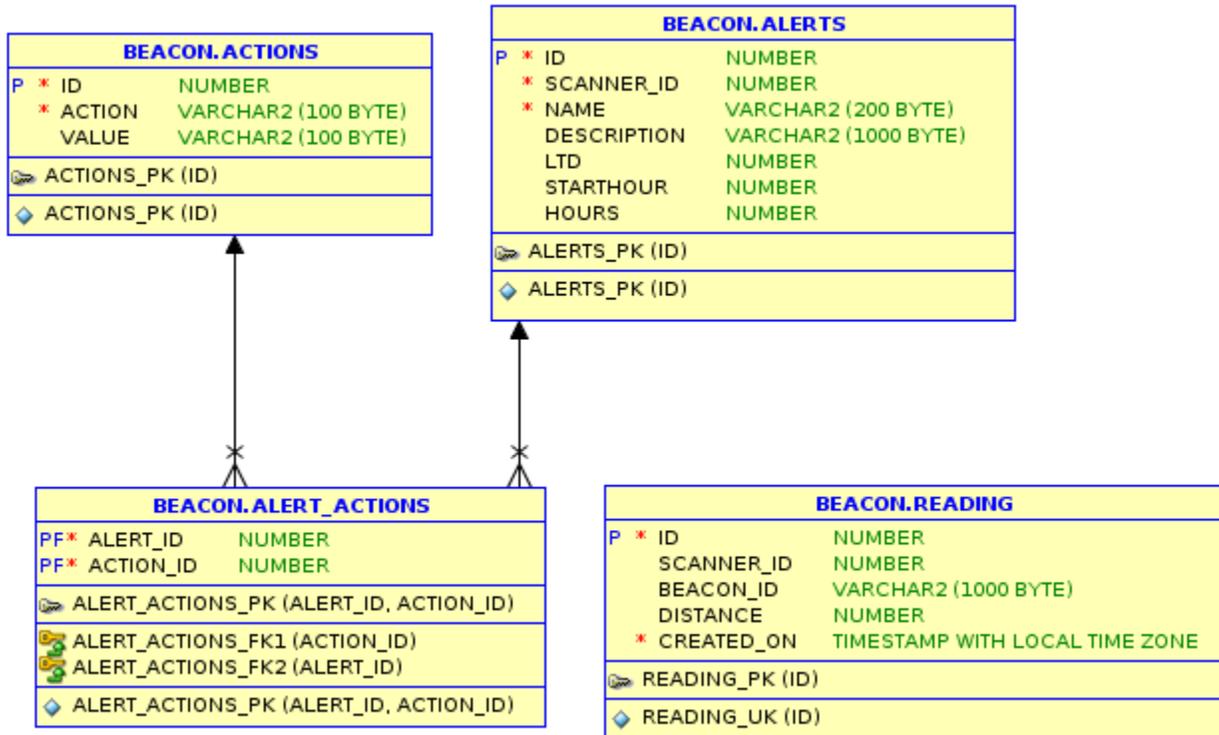
- I placed a Raspberry Pi 3 in the hallway outside of the bedroom and created an alert that is triggered when the beacon is approximately 2 meters away. This alert is set to send a notification through IFTTT*.
- I placed a Raspberry Pi 3 near the front door and created an alert that is triggered when the beacon is approximately 1 meter away. This alert is set to send the 'lock' command to the deadbolt, send 'Locking the front door' to the text to speech function and send a notification through IFTTT.
- I placed a Raspberry Pi Zero W outside near the front door and created an alert that is triggered when the beacon is approximately 1 meter away. This alert is set to send the 'unlock' command to the deadbolt, send 'Unlocking the front door' to the text to speech function and send a notification through IFTTT. (If he makes it outside, I want the door unlocked so he can come back in.)

*IFTTT can also send a text message but it only allows a limited number of texts to be sent each month. I chose to use notifications through their android app since they are unlimited and I would have burned through the text limit the first time I forgot to limit how often I send a notification. In a live situation it could send out multiple texts.

I designed the system to use a database in order to simplify running as many Raspberry Pi as may be needed and to track each time a beacon is detected by a scanner. The alerts and actions for each Raspberry Pi are stored in the database so it can be re-purposed when needed without having to modify the on-board software.

Each Raspberry Pi can trigger multiple alerts each with one or more actions. For example, I may want one alert to always trigger but a second alert only triggers during a certain time of the day.

Database Tables



Column comments:

- **ALERTS:** Trigger alert when the following conditions are met.
 - LTD: 'Less Than Distance in Meters'
 - STARTHOUR: 'Alert is active after this hour (UTC)'
 - HOURS: 'If STARTHOUR is populated, alert will stay active for this many hours'
- **ACTIONS**
 - ACTION: 'Function to run'
 - VALUE: 'Value passed to function'

The application can be altered to run without a database by:

- Removing the request.get near the beginning of the application and manually defining the alerts array in each scanner. For example:


```

        [{"name":"Outside",
        "description":null,
        "ltd":1,
        "startHour":null,
      
```

```

"hours":null,
"actions":[{"action":"ifttt",
            "value":null},
          {"action":"deadBolt",
            "value":"unlock"},
          {"action":"sonosTts",
            "value":"Unlocking the front door"}]
}

```

- Removing the request.post call after the comment “//post beacon distance to the database”.

Future Improvements (if the school wants to use the system)

- Add an Oracle JET front end.
- Display the approximate physical location of one or more beacons. This will require placing multiple Raspberry Pi in known locations and using Trilateration to determine the beacons location from the data. The application is already collecting this data.
- I’m not real happy with the amount of “repeated code” in the node app. I would like to DRY the code a bit and move more of the environment variable values to the database. For example: I intend to convert the deadbolt and Sonos rest calls into a generic function that accepts an action (POST/GET), URL and Payload (from the database) .
- Incorporate the beacon into a small wearable with a rechargeable battery. This would need to be small enough that the students could wear it and not try to remove it. I haven’t done this yet, since the design would need to be something that the school and parents would approve of.

Challenges

- I intended to use OpenHab to control the deadbolt but there is currently a bug in the Z-Wave module for high security devices. I used Home-Assistant.io as a temporary solution installed on the Raspberry Pi near the front door. (Although, Home-Assistant is really growing on me.)
- The Schlage deadbolt I’m using has some ‘quirks’. When I needed to start over, I would have to un-pair the deadbolt from the system *and* do a factory reset. Just doing a factory reset does not un-pair the devices.
- The beacon distance data is not as accurate as I had hoped. I implemented a rolling weighted average in the calculateDistance function to make it a bit more accurate.
- I’m not comfortable enough with the existing security features of either OpenHab or Home-Assistant to actually deploy the ‘unlock’ action. This is mainly because I haven’t taken time to evaluate the security yet, not because I know of anything wrong. Even if those applications are rock solid, it’s too easy to clone and spoof a beacon uuid so I would probably just use notification and text to speech actions.

Thanks

- My son Alex inspired the project.

- My wife put up with project parts all over the house.
- I learned a lot about beacon tracking from this blog post - <https://blog.truthlabs.com/beacon-tracking-with-node-js-and-raspberry-pi-794afa880318>
- Jeff Smith helped me with a couple issues I had getting my JSON just the way I wanted from the database.