

Beacon Audio Triggers

SLQ Wiki Fabrication Lab 2026/01/11 23:29

Beacon Audio Triggers

Using iBeacon modules ([Core-Electronics](#)) we can trigger specific audio files to play when in range.

During the Grumpus the sculptures in the gallery will each have a beacon placed in or near them. These will allow visitors with one of the “listening devices” to listen to sculpture specific stories or sounds as they move around the space.

Visitors will be able to borrow one of our devices from the greeting desk.

The device

- Raspberry Pi Model 3 A+
- Comes with rechargeable and detachable battery
- Headphones
- Custom enclosure

The enclosure will be designed so that battery packs can easily be removed and replaced for charging. Enclosure design should adhere to the Visual ID.

How it works

A python script on each device will use the raspberry pi's Bluetooth module to scan for recognized beacons. The beacons have unique Mac addresses to identify them and are used to trigger specific audio files which are located on each raspberry pi. There are currently no plans to add any buttons (pause, unpause, skip, volume, etc.) to the device, the expectation is that it all works autonomously.

The script scans for devices and determines which is the closest (based on its signal strength) and plays the audio file of that particular beacon. To add beacons and audio files you need to add them to the corresponding lists in the script. If you want to add or change the audio files you must also do so in the audio folder. Please note that the two lists should be in the same order, meaning if beacon1 is first in the beacon list, it's audio file should be first in the audio list.

When the device is running it will spend a couple of seconds scanning for devices and immediately play the corresponding audio track. It will currently not scan until the audio track has finished.

Setting up a new device

Although the idea is to clone the existing SD card once the audio files have been added you may need to setup a new raspberry pi from scratch.

These steps presume you already have a fresh install of Raspberry Pi OS on your Raspberry Pi. If you need support for setting up the device, follow the guides on the Raspberry Pi site [here](#).

During development we used [Visual Studio Code](#) (ARM, .deb). This is not necessary but the built-in terminal, auto activation of virtual environments is convenient. It is recommended you download and install the official Python extension within VSC.

Once you have your development environment setup you will need to install dependencies, mainly [bluepy](#).

Here is the script and audio files we are working with

[pibeaconaudiotrigger.zip](#)

```
$ sudo apt-get install libglib2.0-dev
$ sudo pip3 install bluepy
```

Automatically run script on boot

```
sudo nano /etc/profile
```

Scroll to the bottom and add the following line :

(change myscript to the beacon one below)

```
sudo python /home/pi/myscript.py
```

```
sudo python /home/pi/beacon_audio_trigger.py
```

Once this is done we need the pi to boot into terminal as our devices won't have any screens and in order to get the above script to run when booting. To do so, in the terminal enter:

```
sudo raspi-config
```

Select **1. System Options** then **S5 Boot/ Auto Login** then **B2 Console Autologin**. Selecting this will bring you back to the main menu, go down to Finish and hit enter to save changes.

Bill of Materials

- Raspberry Pi 3 Model A+
- Power pack (2000mAH+) - needs some testing
- Enclosure

Tips

- Use **alsamixer** to adjust volume via the terminal
- When HDMI is connected the Raspberry Pi will favour using it as the audio output over the 3.5mm output (headphones). Disconnecting the HDMI swaps it back to the headphone output.

TODO

- Determine whether to pause a track that is playing when out of range and continue playing it if you get back in range (or to simply play the audio file to the end and then start a new scan)
- Have a “cooldown” for each audio track/beacon so that visitors don't accidentally trigger the same audio file twice
- Design and build case based on visual id or inspired by some of the work in the project
- Purchase more beacons, battery packs and raspberry pis
- Extensive testing
- ~~Run script automatically when raspberry pi boots~~
- ~~Change audio file directory to absolute path (or use working directory)~~