



Piezo Ring Tones with Raspberry Pi

Created by Roberto Marquez



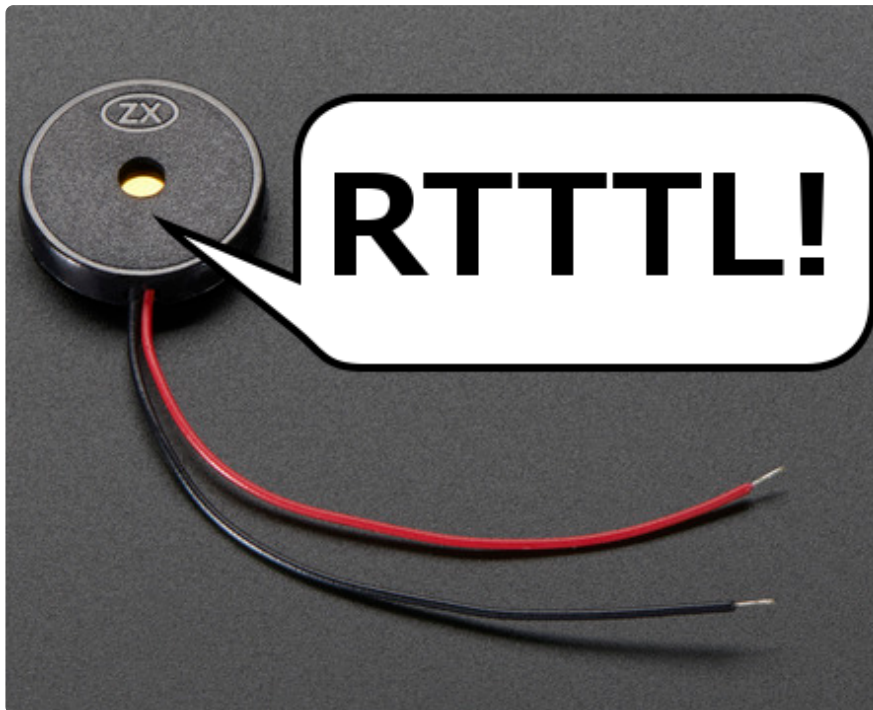
<https://learn.adafruit.com/piezo-ring-tones-with-raspberry-pi>

Last updated on 2023-08-29 02:45:50 PM EDT

Table of Contents

Overview	3
Assembly	3
Software	5
• System Requirements	
• Application	
Usage	6
• Built-In RTTTLs	
• Playing other RTTTLs	
Going Further	8
• Customize the Built-In RTTTLs	
Resources	9

Overview



The ringtone text transfer language (RTTTL) is a way to represent music data for play on older mobile phones. The format was created by Nokia for their phones, but it also works with GPIO pins and a piezo buzzer.

This project shows how to use the Raspberry Pi's GPIO pins to play RTTTL on a piezo buzzer. This project was inspired by existing RTTTL implementations, but differs from those in that its user interface is Web enabled.

The application, written in Java, starts an HTTP server. The HTTP server presents the user interface via HTML. The user interface provides a list of built-in RTTTL ditties available for play. The application's interface also accepts user-provided RTTTL sequences, via its on-the-fly input controls.

Assembly

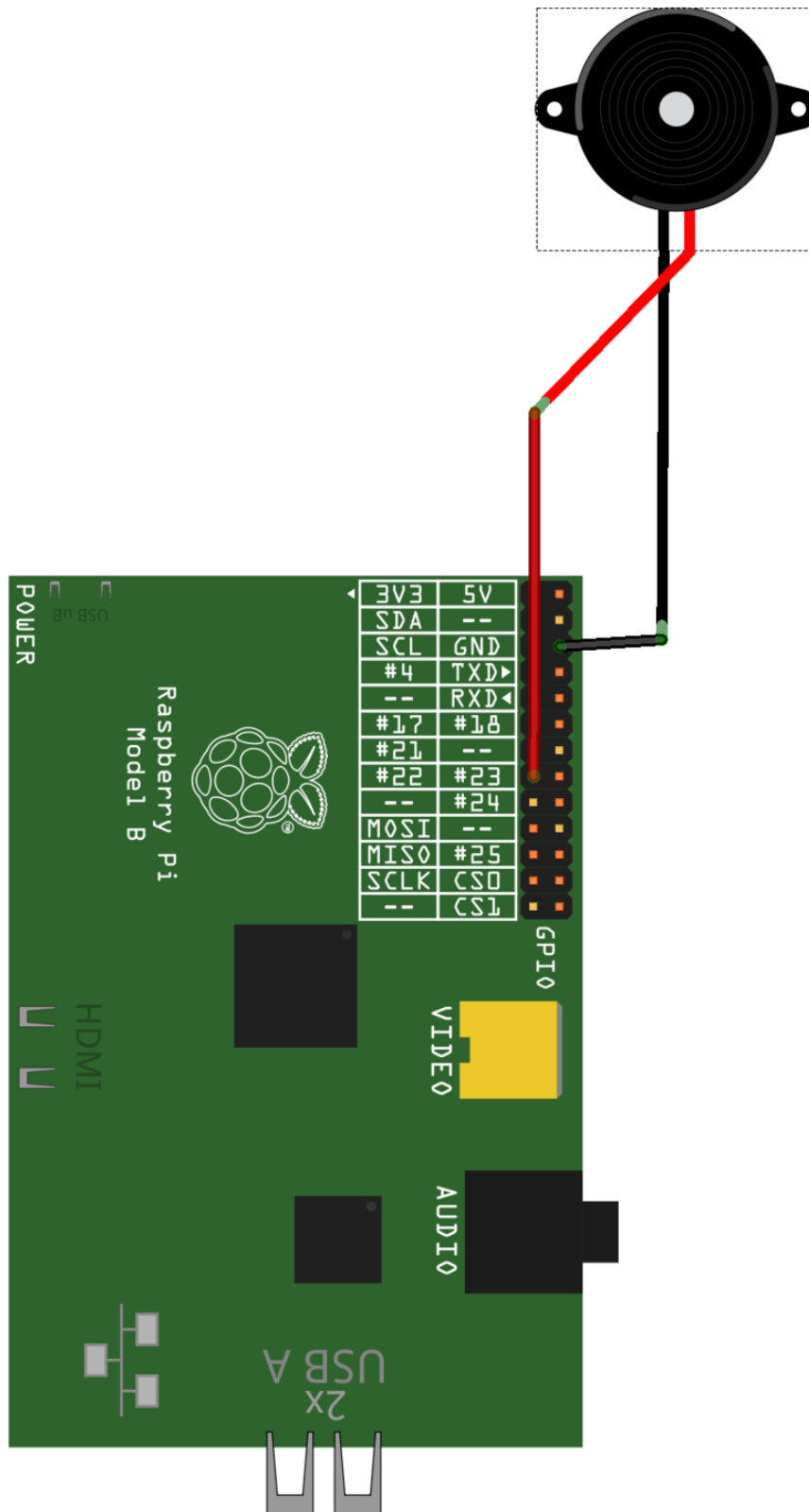
This guide assumes a Raspberry Pi, with network access, is available.

Need help setting up a network-enabled Raspberry Pi? If so, check out these [awesome Raspberry Pi tutorials](#) ().

Assembling this project involves connecting a piezo buzzer's leads to Raspberry Pi GPIO pins.

Connecting a piezo buzzer to the Raspberry Pi is a blast, but don't be alarmed. It is also easy to do.

Here is a wiring diagram for this project:



Made with  Fritzing.org

Connect one of piezo leads to ground on the Raspberry Pi. The other piezo lead goes to GPIO pin 22. It is the 8th pin down, in the left column, from the end closest to the SD card slot.

Software

System Requirements

Java 8 is a requirement for this project. The latest version of Raspbian, the recommended OS for this project, comes pre-bundled with Java 8.

Don't have Java 8 on the Raspberry Pi? Then use these commands to install or update:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install oracle-java8-jdk
```

Application

The application software for this project is distributed as an executable JAR file.

Download the JAR and copy it to the Raspberry Pi.

pisoundo-0.0.1-SNAPSHOT-jar-with-
dependencies.jar

If you are not familiar with copying files over the network to a Raspberry Pi, then see these links for details:

- [easiest way](#) ()
- [remote copy](#) () (also easy)

With the JAR file copied over, issue this at the Raspberry Pi's command line interface:

```
sudo java -jar pisoundo-0.0.1-SNAPSHOT-jar-with-dependencies.jar
```

Replace pisoundo-0.0.1-SNAPSHOT-jar-with-dependencies.jar with whatever you named the JAR file when copying to the Raspberry Pi.

Verify the server application is running and accessible. Browse to this URL, from a device on the same network where the application is running.

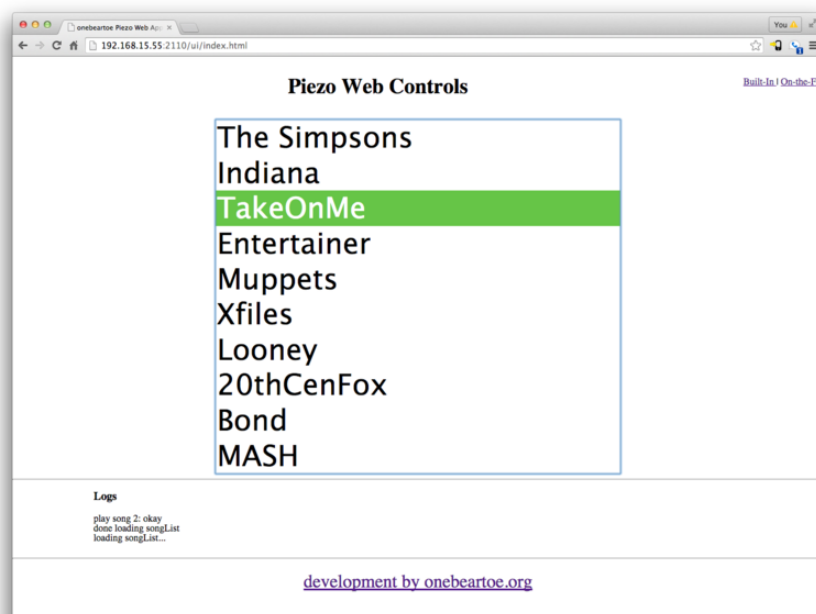
<http://raspberry-pi:2110/ui/index.html>

Notes:

- The 'raspberry-pi' part of the URL should be replaced with the actual IP or hostname of the Raspberry Pi.
- Keep the port number in the URL.

Usage

If everything went well in the Software section, then you should see the following user interface in the browser:

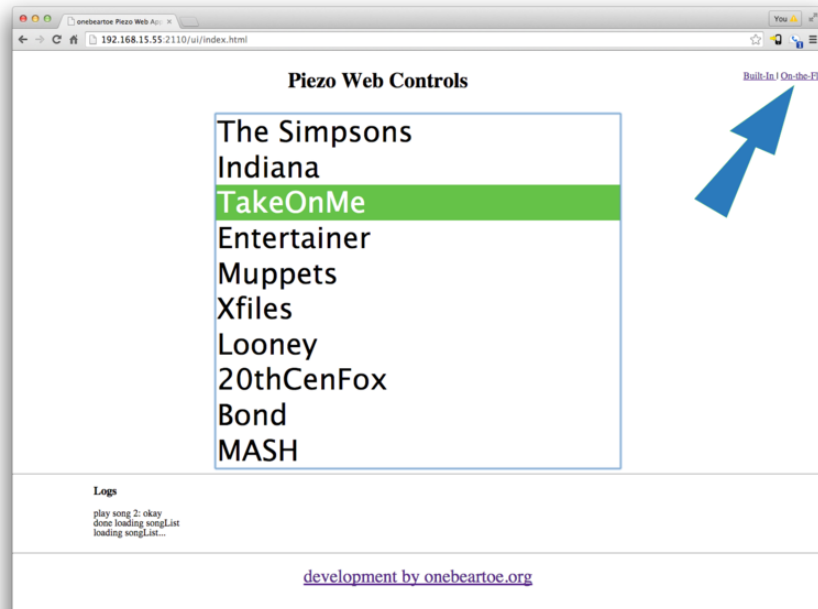


Built-In RTTTLs

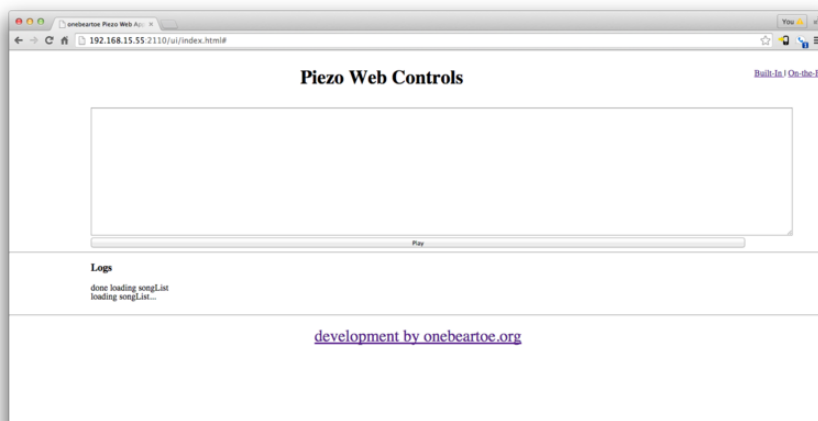
The screenshot above shows the list of built-in RTTTL sequences. Play a ditty by clicking on its name in the list.

Playing other RTTTLS

Play other RTTTLS on the-fly! Click the 'On-the-Fly' link at the top right of the UI.



This brings up a input text box and 'Play' button.



Copy and paste an RTTTL sequence in the textbox, then click the 'Play' button.

An RTTTL sequence is something like:

- PEANUTS:d=4,o=5,b=160:f, 8g, a, 8a, 8g, f, 2g, f, p, f, 8g, a, 1a, 2p, f, 8g, a, 8a, 8g, f, 2g, 2f, 2f, 8g, 1g

- Austin Powers Theme:d=4,o=5,b=140:a#, 8c#6, 8a#, 8d#6, 8c#6, 8a#, a#, 8a#, 8a#, 8f, g#, 8a#, 8f, g#, 8a#, 8f, g#, 8a#, g#, 8a#, 8a#, 8g#, 8a#, 8c#6, 8d#6, 8c#6, f6, f6, 8g#6, f6, 8g#6, 8a#6, 8a#6, 8a#6, 8g#6, 8f6, 8d#6, 8c#6, 8g#, 8a#, 8f, 8g#, 8f, g#, 8a, a#

The 'Resources' section has a link to more RTTTL selections.

Going Further

Customize the Built-In RTTTLs

Bring down the source code for the project.

```
git clone https://github.com/onebeartoe/media-players.git
```

The built-in selections are updated/modified by editing the BuiltInSongs.java file.

The path to that file, under the cloned github project, is:

```
path-to-github-clone/media-players/pi-ezo/src/main/java/org/onebeartoe/media/piezo/ports/rtttl/BuiltInSongs.java
```

Open and edit this file with your favorite text editor.

The actual RTTTL data is stored in the elements of the String array variable named `songData`.

To add RTTTL sequences, place a comma character after the last one, followed by the RTTTL data in double quotes.

So if the RTTTL array data looked like this before adding a sequence:

```
private final String[] songData =
{
  "The Simpsons:d=4,o=5,b=160:c.6,e6,f#6,8a6,g.6,e6,c6,8a,8f#,8f#,8f#,2g,8p,8p,8f#,8f#,8f#,8g,a#.,8c6,8c6,8c6,c6",
  "Indiana:d=4,o=5,b=250:e,8p,8f,8g,8p,1c6,8p.,d,8p,8e,1f,p.,g,8p,8a,8b,8p,1f6,p,a,8p,8b,2c6,2d6,2e6,e,8p,8f,8g,8p,1c6,p,d6,8p,8e6,1f.6,g,8p,8g,e.6,8p,d6,8p,8g,e.6,8p,d6,8p,8g,f.6,8p,e6,8p,8d6,2c6"
};
```

Then the RTTTL array data would look like this after adding a sequence:


```
private final String[] songData =
{
"The Simpsons:d=4,o=5,b=160:c.6,e6,f#6,8a6,g.6,e6,c6,8a,8f#,8f#,8f#,2g,8p,8p,8f#,
8f#,8f#,8g,a#. ,8c6,8c6,8c6,c6",
"Indiana:d=4,o=5,b=250:e,8p,8f,8g,8p,1c6,8p.,d,8p,8e,1f,p.,g,8p,8a,8b,8p,1f6,p,a,8p,
8b,2c6,2d6,2e6,e,8p,8f,8g,8p,1c6,p,d6,8p,8e6,1f.6,g,8p,8g,e.6,8p,d6,8p,8g,e.
6,8p,d6,8p,8g,f.6,8p,e6,8p,8d6,2c6",
"MissionImp:d=16,o=6,b=95:32d,32d#,32d,32d#,32d,32d#,32d,32d#,32d,32d,32d#,32e,32f,
32f#,32g,g,8p,g,8p,a#,p,c7,p,g,8p,g,8p,f,p,f#,p,g,8p,g,8p,a#,p,c7,p,g,8p,g,
8p,f,p,f#,p,a#,g,2d,32p,a#,g,2c#,32p,a#,g,2c,a#5,8c,2p,32p,a#5,g5,2f#,32p,a#5,g5,2f,
32p,a#5,g5,2e,d#,8d"
};
```

Save the changes.

Next, repackage the application for distribution to the Raspberry Pi. At a command line window, change directories to the directory where the github project was cloned. Then issue the following command:

```
mvn package
```

This will create an executable JAR file, at this location:

```
path-to-github-clone/media-players/pi-ezo/target/pisoundo-0.0.1-SNAPSHOT-jar-with-dependencies.jar
```

Copy this updated JAR to the Raspberry Pi, and run as before to see the updated list of built-in RTTTLs.

Resources

- [Piezo on Raspberry Pi using C \(\)](#) (helped with wiring and the pin number in the software)
- [RTTTL on Arduino \(\)](#) (source of the Java port)
- [More RTTTL Tones](#)
- [\(\)](#)
- [RTTTL on Wikipedia](#)
- [\(\)](#)
- [Source Code \(\)](#)
- [Development Builds \(\)](#)