Raspberry Pi Talking Barcode Reader

03/2017 – version 1.0 R.Grokett

Source: DX.com (see Parts below)

Overview

My mom is unable to read labels on grocery items anymore, so I went looking for solutions. After seeing that bar code readers for the blind run many hundreds of dollars, I wanted to see what could be done using a Raspberry Pi and a USB Barcode reader.

I tried a camera based version of bar code reader, but lighting and camera position caused reliability issues for someone unable to see clearly. So I switched to a USB Laser bar code reader, especially since they are now available for under $30 and the Raspbian Jessie supports them right out of the box.

The idea was to scan an item and have the Raspberry query an internal database of UPC codes and then speak the product description using text-to-speech.

In order to make the unit as small as possible, I wanted to use a Raspberry Pi Zero. I also didn't want it to be dependent on Internet connectivity, so the database and the text-to-speech would be local. I also wanted to run the unit off battery power as well.

To meet these requirements some tradeoffs were necessary. Since the Zero doesn't have an external audio jack I reused the audio design from my "Earthquake Pi" project as it was low cost (about $6) and very small. For the Laser Bar Code scanner, I found one from DX.com. It is powered off the Raspberry USB and costs about $25.

UPDATE

With the arrival of the Raspberry Pi Zero W, I am going to switch to it. Everything here is the same, except that you need to set up the WiFi connection to the Pi. This means that you won't need a USB hub for the Zero for setup and use.

You can set up the WiFi using:

https://thepihut.com/blogs/raspberry-pi-tutorials/83502916-how-to-setup-wifi-on-raspbian-jessie-lite

---

Did you know?

If you install Raspbian Jessie on an SD card using a Windows PC, you can create two files on the card to configure WiFi and SSH access before you boot it on a Raspberry?

For this, assume your SD card is currently mounted as K: on your PC:
1) Download the latest Raspbian Jessie image. For this project the Jessie Lite should work.
   https://www.raspberrypi.org/downloads/raspbian/

2) With notepad, create a file called just "**ssh**" and use Save As "All files" to **K:/ssh**
   The file can contain anything. It's the **filename** that is important. Must **NOT** be "ssh.txt"!!!

3) With notepad, create a file called "**wpa_supplicant.conf**" with following:

```
ctrl_interface=DIR=/var/run/wpa_supplicant
GROUP=netdev
update_config=1

network={
    ssid="mySSID"
    psk="mypassword"
    key_mgmt=WPA-PSK
}
```

Use Save As "All files" to **K:/wpa_supplicant.conf**
Again, do not let Notepad change it to "wpa_supplicant.conf.**txt**"!!

When you boot the Raspberry the first time, Jessie will look for these and connect to your Wifi. You will have to look on your Router for the IP address, though, since its auto assigned.

Parts
- Raspberry Pi Zero (or any Raspberry Pi, if size isn't an issue)
- USB Laser bar code scanner
  http://www.dxsoul.com/product/ls-4209-laser-bar-code-scanner-grey-black-901221397
- For Raspi Zero, add Adafruit I2S 3W Class D Amplifier
  https://www.adafruit.com/products/3006 and speaker
  Other Pi's, use external amp & speaker.

- USB Micro to USB OTG adapter
  https://www.adafruit.com/products/2910
- 5V 1Amp USB battery pack such as
  https://www.adafruit.com/products/1959
- USB to micro USB cable for battery pack.
- USB & Ethernet hub (at least for initial setup work)
  https://www.adafruit.com/products/2992
- Software from GITHUB:
  https://github.com/rgrokett/TalkingBarcodeReader

USB Laser Bar Code Reader

Laser bar code readers are more forgiving in their ability to scan a UPC label than cameras. They also work well even in poorly lit or poorly placed labels. You will need the USB to USB micro adapter to plug it into the Zero. The hand grip and trigger of the scanner makes it easy to use. It also gives feedback when it successfully scans a label.

UPC Database

This is the most poorly covered part of the project, as there are literally millions of UPC codes and very few open-source databases available for download. Unfortunately, most UPC databases are rather pricy and require internet connectivity. I was able to put a couple open-source databases together (sources in Appendix below), but even with nearly 700,000 items in it, a vast number are missing. To this end, I have done two things; one is to focus on grocery items specifically and the other is to add a web server to the Raspberry to allow adding new UPC codes manually, though, this does require at least local network connectivity.

Text-to-Speech

I used the text to speech program called "Flite", as it had the best compromise between voice quality and conversion speed on the Zero. Others sounded a little better, but were too slow to convert.

(You can also check out Adafruit's tutorial)
https://learn.adafruit.com/speech-synthesis-on-the-raspberry-pi/programs

Note that it will take a bit of "ear training" to get used to the speech inflection, but it's not too bad.

Battery Power

The system can be run from a 5V USB battery, such as used to recharge cell phones. The catch is that when you unplug or the battery dies, the Pi will crash with a chance that it will corrupt the SD drive. To solve this, a button that runs a shutdown command is needed. I will probably use one like this:

https://www.element14.com/community/docs/DOC-78055/l/adding-a-shutdown-button-to-the-raspberry-pi-b
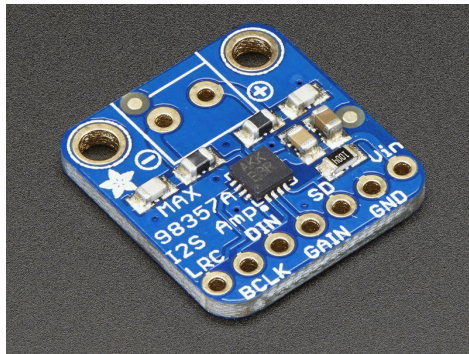
Part I - Adding Audio to Pi Zero
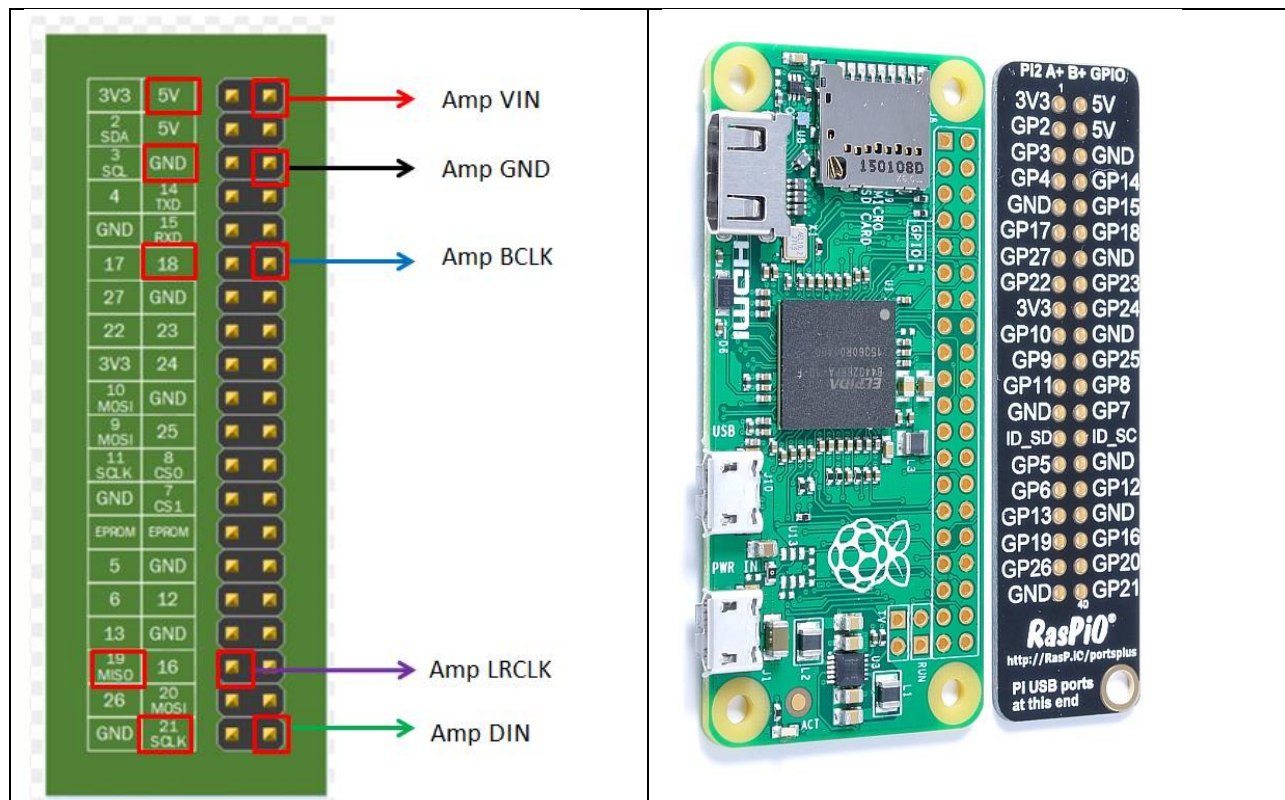
(Skip this part if not using a Zero!)

On a Pi Zero, you will need to add external audio, since it's not built-in. I used the low cost "Adafruit I2S 3W Amplifier breakout MAX98357A" https://www.adafruit.com/products/3006

This requires setup using Adafruit's excellent tutorial at:
https://learn.adafruit.com/adafruit-max98357-i2s-class-d-mono-amp/overview

Adafruit I2S Amplifier

As shown in the wiring tutorial, connect:

- **Amp Vin**   to Raspi Zero Pi **5V**
- **Amp GND**   to Raspi Zero Pi **GND**
- **Amp DIN**   to Raspi Zero Pi **#21**
- **Amp BCLK** to Raspi Zero Pi **#18**
- **Amp LRCLK** to Raspi Zero Pi **#19**

Connect a small speaker to the audio leads on the board.

Pi Zero Adafruit Audio Software Setup

Also shown in the Adafruit software setup tutorial, execute the CURL command below:

```
$ curl -sS https://raw.githubusercontent.com/adafruit/Raspberry-Pi-
Installer-Scripts/master/i2samp.sh | bash
```

After the reboot, test the audio using speaker-test:

```
$ speaker-test -c2 --test=wav -w /usr/share/sounds/alsa/Front_Center.wav
```

If you have problems, go back thru the tutorial starting back at the overview. You may need to go thru the "Detailed installation" steps in the tutorial to verify all the files updated.

Important Required Steps for Zero & Max98357 Audio:

By default, the ALSA audio only works in Stereo while Flite text-to-speech is mono. Also, the audio has "pops" in it every time it starts playing.
(See: https://learn.adafruit.com/adafruit-max98357-i2s-class-d-mono-amp/raspberry-pi-test#reducing-popping)

You must do the following:

1) Append a line to /boot/config.txt
   ```
   $ sudo nano /boot/config.txt
   ```

   ```
   dtoverlay=i2s-mmap
   ```

2) Replace /etc/asound.conf
   ```
   $ sudo nano /etc/asound.conf
   ```

   ```
   pcm.hifiberry {
   type hw card 0
   }

   pcm.!default {
   type plug
   slave.pcm "dmixer"
   }
   ```

```
pcm.dmixer {
type dmix
ipc_key 1024
slave {
pcm "hifiberry"
channels 2
}
}

ctl.dmixer {
type hw
card 0
}
```

3) Reboot your Pi.

Try the Speaker-test again. It should sound better now.

Once you complete the setup tutorial including the audio tests, then sound should be ready.

Note that you may want to add an earphone jack for audio, if you take this to a grocery store or noisy environment.

Part II – Software Installation

This assumes you already have Raspbian Jessie installed and can access the Linux command-line. You do not need the GUI version of Raspbian, but is ok to use.

You DO NOT need the USB scanner plugged in at the moment, in case you need the USB port for Ethernet or keyboard/mouse.

Make sure your OS is up to date:

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

From command-line, go to the pi home directory and download the GITHUB repository:

```
$ cd /home/pi
```

```
$ sudo apt install git
```

```
$ git clone https://github.com/rgrokett/TalkingBarcodeReader.git
```

```
$ cd TalkingBarcodeReader
```

```
$ ./install.sh
```

This will take some time, particularly if this is the first time and the web server and PHP need to be installed or updated. If necessary, you can run this more than once.

Among other things, this installs the barcode files and UPC database to: `/var/www/html/upc`

Next, edit `/etc/rc.local` and append following line before the "`exit 0`"

```
$ sudo nano /etc/rc.local
```

```
fi

sudo /var/www/html/upc/bin/run.sh

exit 0
```

Once installation is completed, shut down your Pi and plug in the USB Barcode reader and restart.

```
$ sudo shutdown now
```

Power it back up and you should hear a short startup speech.

If so, try scanning a bar code from a grocery item. If all went well, you should hear it speak the item (or speak "Not Found") . Otherwise, check on the troubleshooting guide below.

Troubleshooting

If you don't hear any sounds, you will need to try the following commands:

1) Verify the text-to-speech audio is working:
   ```
   sudo flite -voice awb -t "testing"
   ```
   If you get an error, be sure flite was installed:
   ```
   sudo apt install flite
   ```

   If you get the following error, be sure you updated /boot/config.txt and /etc/asound.conf shown in the Audio Setup section above on Raspi Zero's:
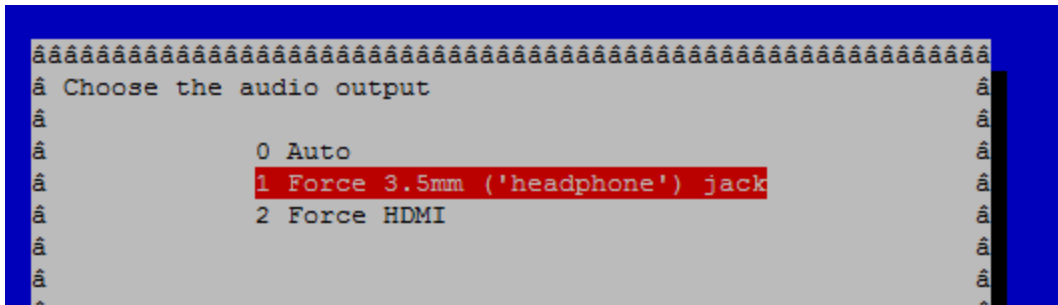   **audio_open_alsa: failed to set number of channels to 1. Invalid argument.**

   If you get no error and no sound, be sure your Raspberry audio is functioning.
   ```
   sudo aplay /usr/share/sounds/alsa/Front_Center.wav
   ```

   You may need to force the audio out the audio jack:

   $ sudo raspi-config -> 7 Advanced Options -> A4 Audio

```
ââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââ
â Choose the audio output                                      â
â                                                              â
â         0 Auto                                               â
â         1 Force 3.5mm ('headphone') jack                     â
â         2 Force HDMI                                         â
â                                                              â
â                                                              â
```

2) Once you have audio and text-to-speech verified good, check the barcode reader:
   ```
   cd /var/www/html/upc/bin
   sudo python ./test.py
   ```
   You should see a numeric barcode number displayed when you scan a barcode. (The test program does NOT include audio output or database connections.)

3) If all of this is good, then check the "nohup.out" file for error messages:
   ```
   cd /var/www/html/upc/bin
   sudo cat nohup.out
   ```
   Likely issues are missing packages or bad permissions (or BUGs in the code!). Be sure your OS is up to date and rerun the install.sh again.

Part III – Web Server

Since there are MILLIONS of UPC codes and most publicly (i.e. free) available databases are only a small subset of codes, this project focused on UPC codes for groceries. It includes several hundred thousand items, but many, many will not be found.

If you need to add or update the UPC Database, there are two programs available; a command-line one and a web browser one.

```
$ cd /var/www/html/upc/bin
$ sudo ./addDB.sh 01234567890 "My new item"
```

This will add a record to the database. Otherwise, it is suggested to use the web browser DB function.

For the Web Server GUI, you need the IP address of your raspberry.
To find the IP address, you can use:

```
$ hostname –I
```

```
192.168.1.108
```

From a PC browser, go to: http://192.168.1.108/upc/

You should see a screen similar to this:

Talking Barcode Reader

Click the buttons on datagrid toolbar to update.

Navigate to the Last Page (>|) for newest records.

UPC		
⊕ New UPC　✎ Edit UPC　▭ Remove UPC　UPC: [　　　　　]		Search　Clear
UPC	Description	
upc14	name	▲
1	Piacci Lightly Smoked Aged Provolone	
1075	Buffalo Strip Steak	
1163	Ground Beef Burgers - 4 Ct	
1164	Ground Beef Burgers - 6 Ct	
1165	Ground Beef Burgers - 8 Ct	
1205	Usps Usa First-class Self Adhesive Forever Postage Stamps - 20 Ct	
1313	Lean Ground Pork	
1473	Flat Iron Beef Steak	
2503	Tallgrass Beef Stew Frozen	
3113	Peaches Saturn Donut	
3278	Plums Dinosaur Egg	
3556	Tallgrass Ground Beef	
3656	Tallgrass Beef Roast Chuck Frozen - 1 Ct	
3743	Chef's Choice Usda Choice Beef Steak New York Strip Center Cut Frozen - 1 Ct	▼
20 ▼　|◄　◄　Page 1　of 38208　▶　▶|　↻		Displaying 1 to 20 of 764151 items

If so, you can add/update and delete records.

Navigate to the Last Page using the ('>|') "last record" button to see your newest records as they are appended to the end of the database.

SECURITY
NOTE that this web service is not secured, (no authentication, no SSL), so don't expose to the public Internet. (But since it's not set up as a WiFi Access Point, it is safe to carry to the store.)

You can also limit access to specific IP addresses so no others can get in. Just substitute the IP(s) here for the ones you wish to allow. All others will be blocked from accessing.

$ sudo vi /etc/lighttpd/lighttpd.conf

```
$HTTP["remoteip"] =~ "127.0.0.1|192.168.1.108" {
      url.access-deny = ( "" )
  }

$ sudo /etc/init.d/lighttpd restart
```

Also, CHANGE the default PI password!
Everybody knows the default is pi/raspberry
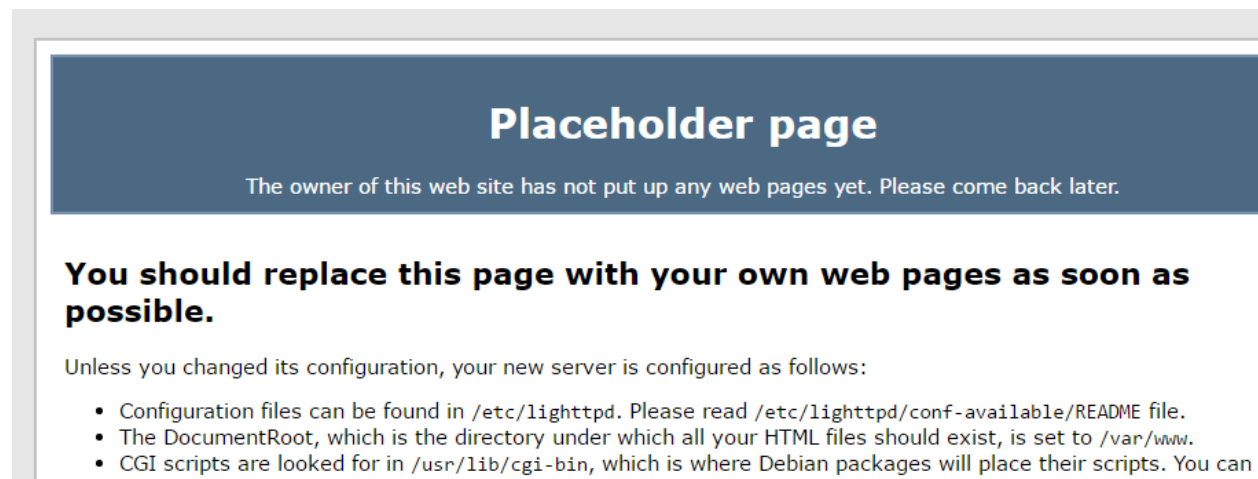
```
$ passwd
```

Troubleshooting the Web Server

Using your browser go to your Raspberry's IP address, like this:

http://192.168.1.108/

You should see the default screen from `/var/www/html/index.html`

Also be sure that the webserver is running:

```
$ ps -ef|grep lighttpd
```


To verify PHP is good, create a php test file:

```
$ sudo nano /var/www/html/phpinfo.php
```

```
<?PHP
phpinfo()
?>
```

http://192.168.1.108/phpinfo.php

If you get errors or blank screens, check for errors in

```
$ sudo cat /var/log/lighttpd/error.log
```

Again, rerunning the install.sh may fix missing dependencies.

Appendix

A BIG THANKS goes to the following for making available a large UPC database.

Open Grocery UPC DB

http://www.grocery.com/open-grocery-database-project/

Open Data DB

http://www.product-open-data.com/download/

http://www.upc-search.org/perl/upc-search.pl?q=849344023999

Thanks to **EasyUI** for the simple web CRUD GUI

https://www.jeasyui.com/index.php