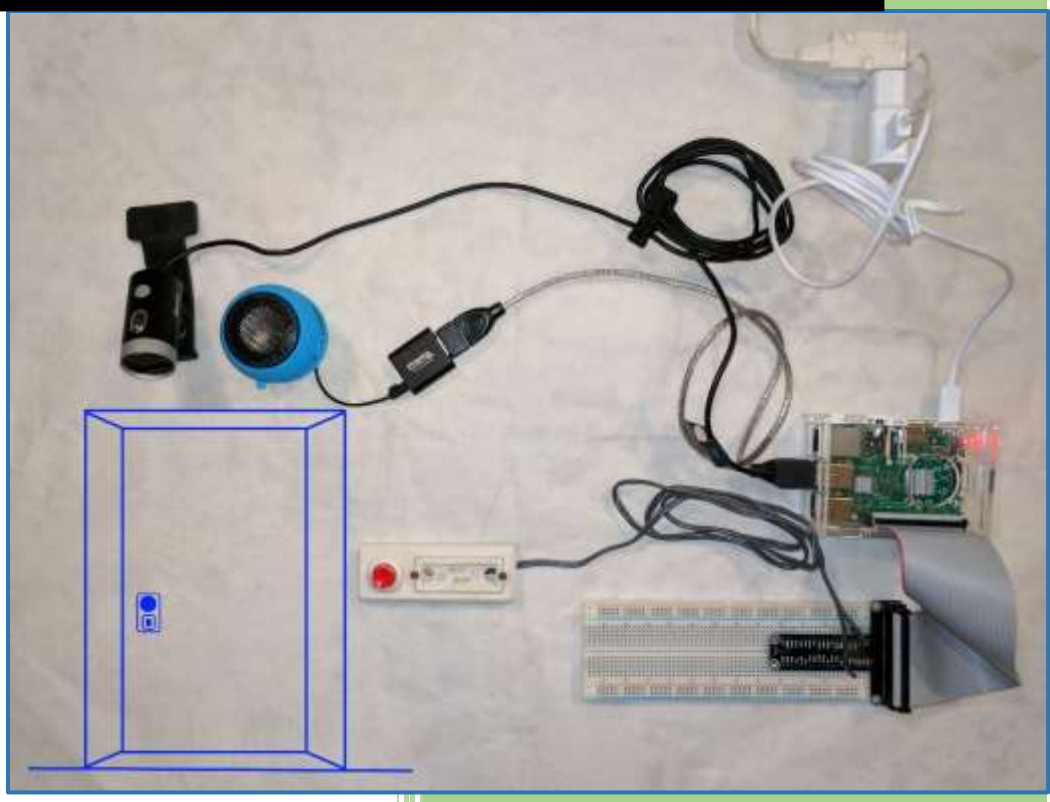
How to make a DIY smart doorbell



Dan Khen

Bellevue College

12/17/2016

Copyright ©

Dan Khen

danny@khen.net

Disclaimer

This document, the *How to make a DIY smart doorbell* guide (hereinafter *this guide* or *the guide*), is provided as a service to hobbyists. The author, Dan Khen (hereinafter *the author*), makes no warranties, express or implied, for the correctness, reliability, sufficiency, or completeness of the material presented in the guide for any purpose. Any reliance you place on information contained in the guide is therefore strictly at your own risk.

In no event will the author be liable for any loss or damage, including without limitation, indirect or consequential loss or damage, or any loss or damage whatsoever arising out of or in connection with the use of this guide.

Through this guide, you can link to other websites, which are not under the control of the author. The author has no control over the nature, content, and availability of those sites. The inclusion of any links in the guide does not imply a recommendation of, or endorse the views expressed within, the linked websites. The author does not endorse, and is not endorsed by, any website, product, brand, or company mentioned in this guide.

Contents

How to make a DIY smart doorbell

1 Introduction: smart doorbells, and why you might want to make one

A *smart doorbell* replaces your conventional front-door bell. It provides you, the homeowner or tenant, with additional "smart" services. For example, a smart doorbell allows you to view people at the door and talk to them. You do that on your phone or other mobile device. It works even if you are at a far corner of your home, or not at home at all.

Good commercial smart doorbells have lately appeared on the market. Ring™ is a well-known one. If you are interested in having a simple, reliable smart doorbell, you may consider getting one of these products.

If, however, you'd like to save some money, and are comfortable with a little bit of Do-It-Yourself (DIY) tinkering, read on! This guide will teach you how to build a smart doorbell yourself. You'll spend much less than the price of buying one. And you'll work on a fun technology project that is simple and cool.

You may also find this guide useful if you are, like the author, a technology geek. You'll be creating your own Internet-of-Things device. You'll experience the fun of programming a tiny powerful computer and extending its interfaces. And finally, the sky is the limit with a general-purpose operating system… You'll be able to extend the smart doorbell's functionality to your heart's content, and make it do additional things for you. It's a geek's dream!

2 Overview: custom doorbell and security camera on a Raspberry Pi

To create your smart doorbell, you will use a *Raspberry Pi* — an inexpensive, small, yet powerful computer. I will call it *the Pi* for short. It is great for digital making — building things with computers. You will also use a free video conferencing service called *Linphone*. It works on all major computers and mobile devices, and is easily programmable.

A few other components complement the Pi for this project. Section 4, Prepare: gather what you'll need, covers them. You will spend about $100 or slightly more (fall 2016 prices).

When you are done, you will have:

1. *A smart doorbell* you can use for a two-way chat with a visitor ringing the doorbell (you can also see them).
2. *A security camera* you can activate at any time to see and hear what's going on in front of your home.

Figure 1: credit-card-sized Raspberry Pi, next to a 5" smartphone

Out of scope for this project: physical installation at the door. The guide ends when you have a working smart doorbell on your desk.

In addition to acquiring the necessary components, you'll also need:

- *Basic technical skills.* Anyone capable of setting up a desktop computer or connect audio/video components should be fine.
- *Basic computer usage skills.* Understanding how a computer works or how to program are not necessary.
- *A mobile device.* This can be a smartphone or a tablet (any major flavor, check at http://www.linphone.org/). You'll use it to remotely connect to your smart doorbell. A laptop or desktop computer will work too when you are at home or at work. This guide uses the term *phone* from here on, for brevity.

3 How to use this guide

The rest of the guide provides steps and information in the order that you'll need them. Reading the sections in order is therefore the best approach.

Most sections have two complementary parts:

1. *Author's choice* or *Instructions*. Make sure to read this part. It is the best approach to getting the things you need, or the essential steps to a part of the project.
2. *More information*. You don't have to read this part. It provides more options (when the author's choice isn't the best for you) or more context (when you need clarification to some of the steps).

Reading the guide start-to-end before starting the project isn't necessary. However, carefully read section 4*,* Prepare: gather what you'll need, before making any purchase decisions. Section 4 will help you make up your mind about starting the project and choosing the options that are right for you.

4 Prepare: gather what you'll need

The cost of the necessary hardware starts at around $100 (fall 2016 prices). The cost might go slightly up depending on options that I cover below. It might also go down if you already have some of the items.

The subsections below explain what you should get, and provide some more information about available options.

4.1 Raspberry Pi

You need a Raspberry Pi 3 Model B and several other items. The Pi retails on its own for around $35. You can buy the other items separately (this is especially attractive if you already have some of them). Or you can shop for a Raspberry Pi kit that contains all the required items.

Figure 2: Raspberry Pi 3 Model B (from http://www.vesalia.de)

Author's choice

Go online and buy a kit that has these items:

1. Raspberry Pi 3 Model B
2. Protective case
3. Micro USB power adapter / charger (2.5 Amp)
4. MicroSD card (min. 8 GB) with the NOOBS operating system (OS) pre-installed
5. Raspberry Pi GPIO parts: ribbon cable, breadboard, GPIO-to-breadboard interface board, jumper wires, push button

More information

- *Kit examples:* Some good examples are: an $80 kit; a $90 kit (fall 2016 prices).
- *Power supply:* Provides power to the Pi. I recommend a 2.5 Amp micro USB charger (min. 2 Amp). The Pi's OS often crashes when running any software if the power supply is too weak.
- *MicroSD card:* Serves as the "hard drive" of your Pi. An 8 GB card is sufficient for this project. It helps if the NOOBS OS is pre-installed on it. Don't worry if it isn't, or if you want to use an existing MicroSD card that you already have. Downloading NOOBS is free and simple. Section 5.1 explains how.
- *GPIO:* The Pi's *input/output* (I/O) extension subsystem, to which you will connect a push button. For easy prototyping, you'll need a flat cable, a GPIO-to-breadboard interface board, a breadboard, a push button that can be inserted into a breadboard, and a couple of jumper wires. A typical Raspberry Pi GPIO kit (often included with the Pi itself) has them. It might include other electronic components; they are nice to have in case you decide to keep trying more projects with your Pi.
- *Case:* Holds and protects your Pi. While not technically required, I highly recommend it. The Pi is a sensitive piece of electronics. You should protect it from dirt, physical damage, and electrical shorts due to accidentally touching the board while it's powered on.

- *Unnecessary items:* No reason to pay any extra for a kit that includes these:
  - *Raspberry Pi camera module:* Better webcam options exist. See 4.3, *Webcam*.
  - *CPU cooling fan:* It is typically unnecessary. And it's hard to wire — it draws power from two of the Pi's GPIO connector pins, the same ones to which you need to connect a flat cable.

4.2 Monitor, keyboard, mouse, network cable

You need these four items for the initial setup of your system.

Author's choice

For initial system setup, use (but don't buy):

1. An HDMI-compatible computer monitor or TV
2. USB keyboard and mouse
3. A network cable

More information

Setting up the system and testing it are easiest with a computer-desktop-like configuration, using the items described above. It only takes a short time, and you don't need these items later. Therefore, you don't have to get dedicated items. You can borrow them from a friend, or even from yourself, for example, from your desktop computer.

If you can't get a hold of these items even temporarily, you can still build the project. You should be able to create a pre-installed MicroSD card, edit it to set up a Wi-Fi connection, then connect to the Pi remotely, without ever connecting a monitor, a keyboard, a mouse. Or a wired network. This is slightly more advanced. You can find information about it online.

After initial system setup, you can disconnect the peripherals and network cable, and use a remote shell or a remote desktop into your Pi. Alternatively, keep everything hooked up while you are building your doorbell, then disconnect. The doorbell does not require these items to operate.

4.3 Webcam

The smart doorbell lets you view the visitor at the door, and can be used as a security camera. You need a *webcam*, a digital camera connected to the Raspberry Pi.

Author's choice

Go online and buy a USB webcam. Make sure it includes a built-in microphone.

More information

- If you have a webcam to play with, you can use it. You may have to add a microphone if your webcam does not have one (see 4.4, *Audio*).
- If your Raspberry Pi kit includes the Pi's camera module, you can try using it. Bear in mind that the camera module has some issues:
    - It has no microphone. You will have to add one (see 4.4, *Audio*).
    - It is very sensitive to static, and breaks down easily.
    - It costs $20, so you won't be saving money compared to a USB webcam.
- Your best option is to buy a USB webcam that has a built-in microphone. You will have a reliable camera, and will not have to add a microphone. Inexpensive webcams sell for under $10. Good ones should cost $20-$30. If you intend to install the smart doorbell at your front door, consider a good webcam, and opt for wide angle capture.

4.4 Audio

The smart doorbell allows you to hear what visitors at the door say, and speak back to them. You need a *microphone* and a *speaker*.

Author's choice

1. *Audio input:* get a USB webcam that has a mic (see 4.3, *Webcam*).
2. *Audio output:* use the Raspberry Pi's on-board analog audio output jack.
3. *Speaker:* Go online and buy an inexpensive small speaker.

More information

1. The author's choice above is the least expensive and easiest to configure.
2. You may have to get a *USB audio adapter* (such as Plugable). The main reason would be when your webcam does not have a built-in microphone, and you are using a separate microphone. The Pi's built-in audio jack does not support audio input, while the adapters typically have both input and output jacks.
3. The speaker does not have to be particularly good. You can also do without stereo speakers. A simple speaker should cost under $10. For example, this Portable Mini Hamburger Speaker.

5 Get started: set up your Raspberry Pi

5.1 Install the operating system

The Raspberry Pi's OS that you will use is called *Raspbian*, a derivative of *Linux*. The Raspberry Pi Foundation also has a starter OS called *NOOBS* (for *New Out Of the Box Software*). NOOBS often comes pre-installed on MicroSD cards. It contains Raspbian, or (in some cases) allows to easily download and install Raspbian.

CAUTION:

- Always unplug the power from the Raspberry Pi when connecting or disconnecting parts.
- When power is on, touching any exposed circuitry might damage some of the electronics beyond repair. Be careful to hold the Pi's circuit board by its edges only.

Instructions

1. Use a MicroSD card that has the NOOBS OS installed on it.
2. Insert the MicroSD card into the Pi's card slot. The slot is located on the bottom side.

Figure 3: Inserting the MicroSD card (from beebom.com)

3. Connect the Pi to the monitor, keyboard, mouse, and network cable.
4. Power on the Pi: connect the USB power adapter, and then plug the adapter in to a power outlet.
5. The first time you power on, wait for NOOBS to ask you to select an operating system, and then select *Raspbian*.

More information

1. If your MicroSD card does not come pre-loaded with NOOBS, you have two options:
   1.1. Download NOOBS and copy it to the card, and then proceed as explained above. Details: https://www.raspberrypi.org/downloads/noobs/ (Download NOOBS, 2017)
   1.2. Download a Raspbian image and use an image writing tool to install it on an SD card. Details: https://www.raspberrypi.org/downloads/raspbian/ (Download Raspbian, 2017)

2. At step 5 above, NOOBS downloads Raspbian from the Internet (if it isn't already included on the MicroSD card). Then it starts Raspbian and launches its graphic desktop.

5.2 Configure the operating system

There are a few simple configuration steps you should make to your newly installed Raspberry Pi.

5.2.1 Change password

The default user name on the Raspberry Pi is **pi**, with the default password **raspberry**. For security reasons, you should change this default password.

Instructions

1. Open a terminal window by clicking the **Terminal** button on the task bar.
2. Type **passwd**, then press **Enter**.
3. Follow the prompts and type the current password, **raspberry**, then your new password, and finally your new password again.

```
pi@raspberrypi:~$ passwd
Changing password for pi.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
pi@raspberrypi:~$
```

Figure 4: Changing a password

More information

1. Here is the **Terminal** button on the task bar:

Figure 5: Opening a terminal window

2. On the terminal window, press the **Enter** key at the end of each command or input line.
3. While running the **passwd** command, the characters you type are invisible and nothing happens on the screen. The system silently reads them as you type, and the program receives them when you press **Enter**.

5.2.2 Set up Wi-Fi

Now that your OS is up and running, you can set up a *Wi-Fi* (wireless) network connection to your network router. After setting up Wi-Fi, you may disconnect the wired network cable.

Instructions

Raspbian has a graphic user interface (GUI) for setting up Wi-Fi.

Follow the instructions on Raspberry Pi's Wi-Fi page (WiFi, 2017).

5.2.3 Choose audio device

If you intend to use a USB audio adapter (see 4.4, *Audio*), you should set it as the default audio device.

Instructions

1. Connect your USB audio adapter to a USB port on the Raspberry Pi.
2. On the task bar, right-click the **Volume** (speaker) icon.
3. Select **USB Audio Device**.

5.3 Remotely connect to your Raspberry Pi

Now that your Pi is set up, you have the option to remotely connect to your Pi from another computer. Working remotely is often more convenient. In addition, it frees up the monitor, keyboard, and mouse connected to your Pi. If you work remotely, you can disconnect them.

Remote connection is entirely optional. You can keep using your monitor, keyboard, and mouse for the rest of the project.

Instructions

These two pages will show you how to work remotely:

1. Remote shell: use a remote terminal window. (SSH - Secure Shell, 2017)
2. Remote desktop: use an entire remote graphic desktop. (VNC - Virtual Network Computing), 2017)

6 Make it: create your smart doorbell

6.1 Install Linphone

6.1.1 Linphone app

This subsection shows you how to install the *Linphone* video conferencing app on two devices: your Raspberry Pi and your phone.

Instructions

1. Open a terminal window, and type:

```
sudo apt-get install Linphone
```

2. Answer **Y** to the question about getting extra archives.
3. Click the **Applications** menu (raspberry icon), then select **Internet**, then select **Linphone**.
4. Use the wizard that opens to sign up for a Linphone account for your Pi.
5. On the email that you receive, click the link to confirm the account.
6. Open the **Options** menu, and then click **Preferences** to open the **Settings** dialog.
7. Select the **Multimedia settings** tab, and set your audio **Playback device** and **Capture device**.
8. Install the free Linphone app on your phone, using the instructions here: http://www.linphone.org (Linphone, 2017).
9. Run the app and follow its instructions to set up a *second* account for your phone.
10. Try out Linphone! Call your phone from your Pi. Call the Pi from the phone.

More information

- To communicate between the Pi and your phone, you need separate Linphone accounts, one for each side.
- Write down the usernames and passwords of all accounts you create.
- You don't have to do the email verification step on the Pi. You can read the email anywhere and click the link.
- To install Linphone on your phone, you may be redirected to the phone's app store.
- Exit the Linphone app on the Pi when you are done trying it. You will not use this app for the project. Instead, you will use a program that "drives" it through custom code.

6.1.2 Linphone python for Raspberry Pi

This is a collection of software libraries that the project code uses to run and control Linphone.

Instructions

To install it, open a terminal window on the Raspberry Pi, and type these two commands:

```
wget https://linphone.org/releases/linphone-python-
   raspberry/linphone4raspberry-3.9.0-cp27-none-any.whl
sudo pip install linphone4raspberry-3.9.0-cp27-none-any.whl
```

6.2 Set up the doorbell software

This is the only part of the guide that requires you to do some software investigation and editing. The instructions below aim to be detailed enough for readers without experience with such tasks.

Some steps require editing text files. See *More information* for information about text editing.

Instructions

1. To download the smart doorbell software, open a terminal window, and type:

```
cd ~
sudo apt-get install Linphone
git clone https://github.com/dannykhen/SmartDoorbell
cd SmartDoorbell
chmod +x *.py
rm -rf .git*
```

2. Let the **smartDoorbell.py** program show you the names of video and audio devices:
   2.1. Edit the file **smartDoorbell.py** in the directory **SmartDoorbell**.
   2.2. Scroll to the bottom and locate the statement that creates the **`SmartDoorbell`** object. It has a long list of parameters (see *Figure 6*).
   2.3. Locate the second parameter: **`module_log_level=logging.ERROR`**
   2.4. Modify the word **`ERROR`** to **`INFO`**, and then save the file.
   2.5. In the terminal window, type:

```
smartDoorbell.py |& egrep 'Webcam|Card'
```

   2.6. After the program prints some information and stops, type **Ctrl+C** to terminate it.
   2.7. Go back to the editor, and change back the word **`INFO`** to **`ERROR`**.

Figure 6: code creating the SmartDoorbell object

3. Configure the program to use your devices and accounts: on the **`SmartDoorbell`** object creation at the bottom of the file, set the parameters shown in the table below. To set a parameter, change the value after the = sign (leave the comma at the end if there is one). You'll find the values to the **`camera`**, **`sound_capture`** and **`sound_playback`** parameters in the output of the program you ran in the previous step.

Parameter name	Set it to…
`username`	Name of Pi's Linphone account (see 6.1.1, *Linphone app*).
`password`	Password of Pi's Linphone account.
`trusted`	List of one or more Linphone account addresses that can call into the Pi to use the security camera. The program calls the first one when a visitor rings the doorbell. Typically, just one address.
`camera`	A camera device name. It looks like `Webcam V4L2: /dev/`…
`sound_capture`	An audio device name for the microphone. In the example above it is the audio device of the webcam, a *Microsoft LifeCam*, since I am using the webcam's microphone.
`sound_playback`	An audio device name for the speaker. In the example above, it is the internal analog audio device. I also included a commented-out value for the USB audio device. To use it, move the comment sign (a hash - #) from this value to the one above it.

4. Save the file and exit the editor.
5. Make the program run automatically when the Raspberry Pi powers on:
   5.1. In a terminal window, edit the file **/etc/rc.local** by typing:

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

   See *More information* for editor tips.
   5.2. Navigate to the bottom of the file, and add this line above the line with `exit 0`:

```
sudo -u pi /home/pi/SmartDoorbell/smartDoorbell.py >&2 2>
   /home/pi/SmartDoorbell/smartDoorbell.log &
```

   5.3. Save the file and exit the editor.

More information

About file location:

The guide assumes that you are logged in as the default Raspbian user, *pi*, and that you are installing the project code in pi's home directory, `/home/pi`. If this isn't the case, modify occurrences of `/home/pi` in the code and in the guide's steps with the correct location.

About text editing:

The file **smartDoorbell.py** is a Python program. The best editor to use is the Python developer environment. To use it:

1. Open the **File Manager** (the icon that looks like file folders).
2. Double-click the folder **SmartDoorbell**.
3. Right-click the file **smartDoorbell.py**, and then select **Python 2 (IDLE)**.
4. Use your mouse or keyboard to navigate and select text, the keyboard to type text, **File > Save** to save, and **File > Exit** to quit.

When editing **/etc/rc.local**, you need system permissions. This requires using the terminal window, and editing using a command-line editor. The instructions above use *nano*, which is a very simple command-line editor. Some basic facts about it:

1. Use your keyboard to navigate: arrow keys, PageUp, PageDown.
2. Use the keyboard commands that are listed at the bottom to perform various operations. For example, use **Ctrl+X** for **Exit**. You will be prompted for saving your changes.

6.3 Set up the doorbell I/O extension

This project has one I/O extension: the input signal coming from the push button that your visitors will use to ring the doorbell.

CAUTION:

- Always unplug the power from the Raspberry Pi when connecting or disconnecting parts.
- When power is on, touching any exposed circuitry might damage some of the electronics beyond repair. Be careful to hold the Pi's circuit board by its edges only.

Instructions

1. In a terminal window, type:

```
sudo shutdown now
```

2. Wait for the Pi to shut down, and unplug the power adapter.
3. Connect the push button hardware:
   3.1. Insert the GPIO-to-breadboard interface board (a T-shaped board with pins sticking out at the bottom) into the breadboard. See *Figure 7* below.
   3.2. Connect the flat ribbon cable to the GPIO connector on the interface board.
   3.3. Connect the other end of the flat cable to the GPIO connector on the Raspberry Pi.
   3.4. Connect the push button:
      3.4.1. Insert the button's pins into the breadboard (per your kit's instructions).
      3.4.2. Insert two jumper wires next to the button's pins. Insert the other ends of the wires next to the interface board pins marked **GND** and **GPIO17**.

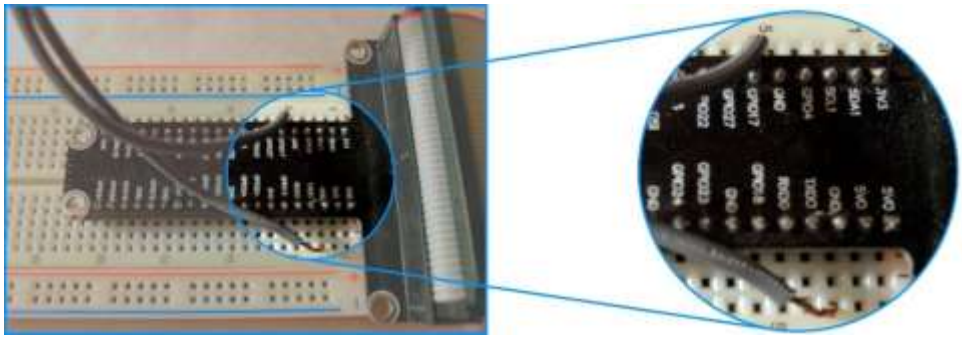


Figure 7: breadboard with GPIO interface and push button wires

7 Try it: use your new security camera and smart doorbell

It is time to try your new doorbell. Restart the Raspberry Pi. Once running again, it is ready to use: the necessary hardware is connected, and the software was automatically loaded during restart and is now running.

7.1 Use the security camera

Launch the Linphone app on your phone, and call your Raspberry Pi account. The program on the Pi automatically answers your call, and you should be able to see what's in front of the Pi's webcam and hear what the microphone is picking up.

When you are done, hang up the call on your phone. The program on the Pi disconnects the call.

7.2 Use the smart doorbell

Push the doorbell button. The Pi's speaker plays a doorbell sound (*Ding Dong!*). Then, Linphone rings on your phone. Answer the call! You are now connected to the Pi. You can see the "visitor at the door" and speak back.

When you are done, hang up the call on your phone. The program on the Pi disconnects the call.

8 Conclusion

You now have your very own smart doorbell and security camera. You used a Raspberry Pi computer, the Linphone free video conferencing app, and some other hardware and software components, and you went through the steps of putting them all together.

The picture in *Figure 8* below shows the completed project. I laid out the outside components on a schematic front door. The remaining components can be inside, with wires passing through the wall.

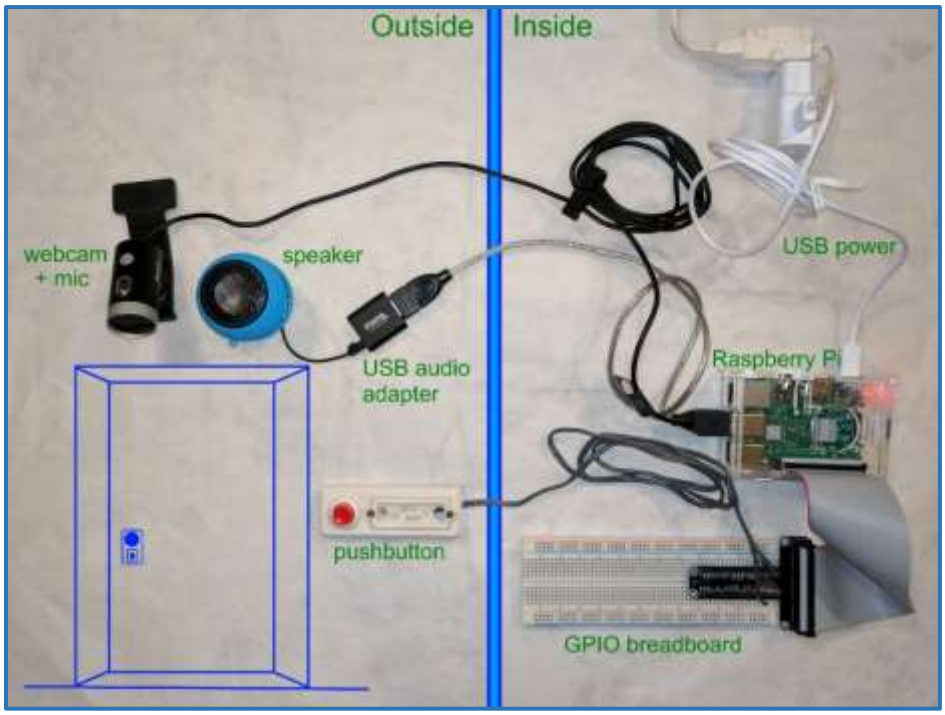


Figure 8: the finished smart doorbell

The picture shows the speaker connected to a USB audio adapter. As explained above, it is optional. You can use the Pi's audio output. Next to the speaker you can see a webcam that includes a microphone. The GPIO board features a real doorbell push button instead of a hardware kit breadboard button. It's there for a more realistic effect (and because the author is a geek who has such items lying around the house).

Notice also that *Figure 8* does not show computer peripherals — monitor, keyboard, and mouse. They are not necessary after the software is all set up.

As a next step, you can think about physically installing your new doorbell at your front door. Beyond that, set your imagination free! After all, your doorbell is a fully-programmable computer with a general-purpose OS and a hardware interface. You can extend it in countless creative ways.

I hope you had fun building this project and experiencing the joy of digital making. Happy tinkering!

References

Download NOOBS. (2017, 1). Retrieved from Raspberry Pi: https://www.raspberrypi.org/downloads/noobs/

Download Raspbian. (2017, 1). Retrieved from Raspberry Pi: https://www.raspberrypi.org/downloads/raspbian/

Linphone. (2017, 1). Retrieved from Linphone: http://www.linphone.org/

SSH - Secure Shell. (2017, 1). Retrieved from Raspberry Pi: https://www.raspberrypi.org/documentation/remote-access/ssh/

VNC - Virtual Network Computing). (2017, 1). Retrieved from Raspberry Pi: https://www.raspberrypi.org/documentation/remote-access/vnc/

WiFi. (2017, 1). Retrieved from Raspberry Pi: https://www.raspberrypi.org/documentation/configuration/wireless/

Resources

Setting up Linphone on the Raspberry Pi, as well as the starting point for the project's Python code, are based on the Linphone wiki: https://wiki.linphone.org/wiki/index.php/Raspberrypi:start

Code to play a doorbell sound uses *Pygame*, an open source python multimedia library. The author relied on the Pygame wiki: http://www.pygame.org/wiki/index

Appendix A: style sheet

Scope and sources

- Name: Dan Khen
- Date: 1/6/2017
- Document scope: this document only
- Style manual: *Microsoft® Manual of Style, 4th Edition (2012)* (MMS)
- Dictionary: https://www.merriam-webster.com/ (1/6/2017)
- Other sources:
    - Raspberry Pi Foundation
    - Wikipedia

Style decisions

Category	Rule	Source
Separating a phrase / clause	Use an em dash.	MMS, Chapter 9, p. 198
Em dash spacing	Surround it by ¼ en spaces (aka hair spaces, per Wikipedia).	MMS, Chapter 9, p. 198 https://en.wikipedia.org/wiki/Dash
Hyphen	Hyphenate two or more words that precede and modify a noun as a unit if confusion might otherwise result.	MMS, Chapter 9, p. 195
pre*, sub*	Do not hyphenate words beginning with these prefixes, unless it is necessary to avoid confusion, as in *pre-engineered*, or if *pre* is followed by a proper noun, as in *pre-C++* or *sub-Saharan*.	MMS, Usage Dictionary, pp. 361,391
Title / headings	Use sentence caps Use Word's *Heading 1,2,3* styles for headings. Use custom-designed headings for *Author's choice*, *Instructions*, *More information*, and *Caution* sections.	MMS, Chapter 7, p. 134
Contractions	Use common contractions, such as it's, you're, and don't, but avoid more ambiguous ones such as there'd, it'll, and they'll.	MMS, Chapter 1, p. 11
Acronyms	Spell them out on first mention, followed by the acronym in parentheses. Later, just use the acronym.	MMS, Chapter 11, p. 215
Numbers	Spell out zero to nine (except when preceding a unit of measure).	MMS, Chapter 7, p. 152
Cross references	Format as: see <section #>, *<section heading>* Example: see 4.4, *Audio* Use sentence casing.	MMS, Chapter 7, pp. 147-148

	If needed, use *earlier* or *later*, not *above* or *below*.	
URLs	It is ok to quote URLs (not all hyperlinks must have meaningful display text).	MMS, Chapter 7, pp. 163
Hyperlinks	Use the title or a description of the new webpage as the link, rather than an empty phrase such as *click here*. Use the preposition *at* with the location of an address.	MMS, Chapter 7, pp. 164

Usage dictionary

Use this	Not this	Source / Notes
around		https://www.merriam-webster.com/dictionary/around Can be used to mean *approximately*.
doorbell	door bell	https://www.merriam-webster.com/dictionary/doorbell
Internet	internet	MMS, Usage Dictionary, p. 317
pi		Use for the default user name on Raspbian. (No source; that's the verbatim user name.)
Raspberry Pi or **Pi**	pi, rpi, RPi	Raspberry Pi Foundation, various reference pages. Use for the computer's name. Use the full name in headings, on first mention each section, and as part of compound names like *Raspberry Pi GPIO kit*. Otherwise, can use *Pi* for short.
push button (n.)	pushbutton, push-button	https://www.merriam-webster.com/dictionary/push-button The compound *pushbutton* isn't in the dictionary. The hyphenated *push-button* is an adjective, e.g. *a push-button phone*.
setup (n.)	set up, set-up	https://www.merriam-webster.com/dictionary/setup
subsection	sub-section, sub section	https://www.merriam-webster.com/dictionary/subsection
webcam		https://www.merriam-webster.com/dictionary/webcam
Wi-Fi	WiFi, Wifi, wifi, wi-fi	MMS, Usage Dictionary, p. 415