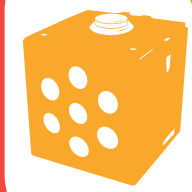
The MagPi ESSENTIALS

FREE KIT
NOT INCLUDED
with **digital edition**

AIY PROJECTS



CREATE A VOICE KIT WITH YOUR *Raspberry Pi*

Written by **Lucy Hattersley**

The MagPi ESSENTIALS

LEARN | CODE | MAKE

AVAILABLE NOW:

- > CONQUER THE COMMAND LINE
- > EXPERIMENT WITH SENSE HAT
- > MAKE GAMES WITH PYTHON
- > CODE MUSIC WITH SONIC PI
- > LEARN TO CODE WITH SCRATCH
- > HACK & MAKE IN MINECRAFT
- > ELECTRONICS WITH GPIO ZERO
- > LEARN TO CODE WITH C
- > THE CAMERA MODULE GUIDE

The MagPi ESSENTIALS

From the makers of the official Raspberry Pi magazine

The MagPi ESSENTIALS

WELCOME TO AIY PROJECTS

Google AIY Projects brings do-it-yourself artificial intelligence (AI) to the maker community.

AIY Projects is a series of open-source designs that demonstrate how easy it is to add AI to your projects.

We are thrilled to present the very first project, a kit that lets you explore voice recognition and natural language understanding.

You will build a cardboard device that uses the Google Assistant to answer questions, like "how far away is the moon?" or "what is 18 percent of 92?".

Then you will learn how to add voice commands to your own projects. For example, you can register commands, such as "turn the lights on" or "robot, turn right and move forwards". In the kit, Google has included a microphone, a speaker, and an accessory board called Voice HAT, that is loaded with breakout pins to wire up a variety of sensors and components.

Google can't wait for makers to build intelligent devices that solve real-world problems and share them back to the community using the #AIYProjects hashtag on social media.

Your kit is a fantastic way to add voice control to your projects, and start exploring what's possible with AI.

Lucy Hattersley

FIND US ONLINE raspberrypi.org/magpi **GET IN TOUCH** magpi@raspberrypi.org

The MagPi

Available on the App Store

Google play

EDITORIAL
Publishing Director: **Russell Barnes**
Editor: **Lucy Hattersley**
lucy@raspberrypi.org
Sub Editors: **Rachel Churcher and Phil King**

DESIGN
Critical Media: **criticalmedia.co.uk**
Head of Design: **Dougal Matthews**
Designers: **Lee Allen, Mike Kay**

DISTRIBUTION
Seymour Distribution Ltd
2 East Poultry Ave, London
EC1A 9PT | **+44 (0)207 429 4000**

THE MAGPI SUBSCRIPTIONS
Select Publisher Services Ltd
PO Box 6337, Bournemouth
BH1 9EH | **+44 (0)1202 586 848**
magpi.cc/Subs1

recycle

In print, this product is made using paper sourced from sustainable forests and the printer operates an environmental management system which has been assessed as conforming to ISO 14001.

ISSN: 2051-9982. ISBN 978-1-912047-71-0.

The MagPi ESSENTIALS

CONTENTS

[LUCY HATTERSLEY]

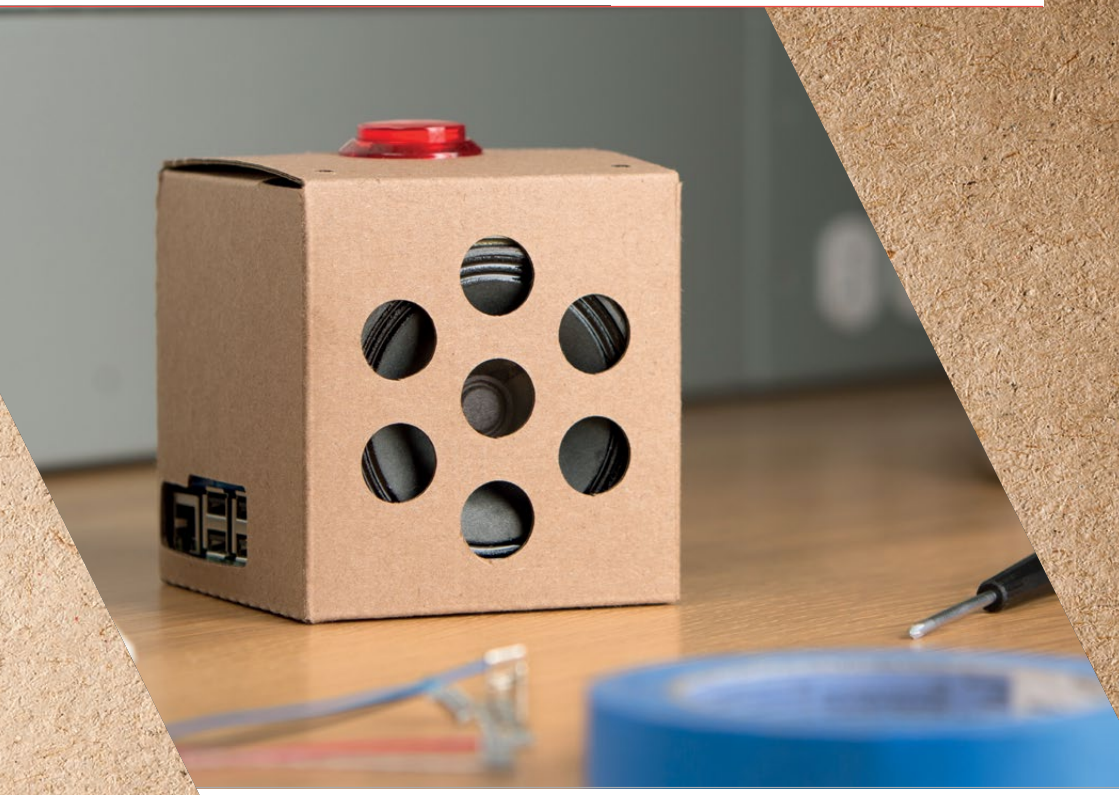
Lucy is the Editor of The MagPi – The Official Raspberry Pi magazine. Her first computer was a Sinclair Spectrum, but Commodore was her true love. First with the VIC-20, then the C64, and finally the adorable Amiga. Lucy learnt to code at school using Acorn computers; then learnt it all again with MITx. Lucy has been making computer magazines since she left school.

The MagPi ESSENTIALS

[CHAPTER ONE]

MADE BY YOU WITH Google

The AIY Projects team chats to us about the making of this incredible Raspberry Pi kit

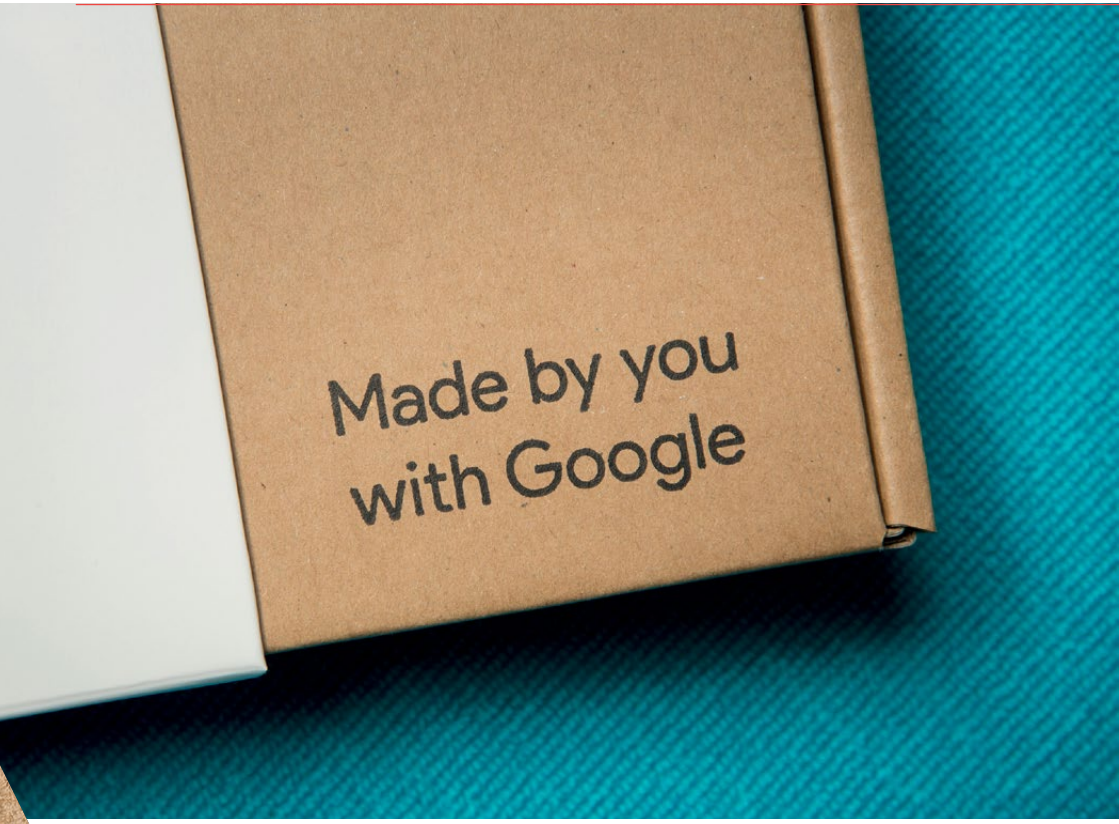


"Natural Human Interaction is this idea of being able to communicate with an electronic device the same way you and I are talking right now," says Billy Rutledge, Director of AIY Projects at Google. We've caught up at Raspberry Pi Towers to discuss the AIY Projects kit, and the future of artificial intelligence with the maker community.

"We're all familiar with graphical user interfaces (GUI)", notes Billy. "Well, building a VUI is now the big thing." Voice has become "very popular" in the last year, says Billy. "Not just with consumer products, but also as a set of tools for device makers."

Google wants to help makers familiarise themselves with voice interfaces, but it's also really keen to tap into the creative prowess of the maker community.

"We're excited to put the kits out into the world and see what people make with them," says Kristine Kohlhepp, a User Experience



Researcher working on AIY Projects at Google. "We've done a lot of research to make sure people can assemble the kit and figure out how to make it work."

"The initial project is just an opener," reveals Blaise Agüera y Arcas, Principal Scientist at Google. "It's fun to be able to make a cardboard kit that uses the Google Assistant, but this is about a lot more than just making a lower-cost DIY version of Home."

The future belongs to intelligent devices. Billy says: "At some point soon, we'll see a new generation of devices that you can just walk up to and ask 'what are you and what do you do?' Then you'll have a conversation with it, to use its services in a very easy-to-understand way."

Natural Human Interaction is the term used for this kind of interaction between humans and devices. "A generation or two ago, all of our devices

had analogue dials and knobs," explains Billy. "Then there was a shift to digital buttons and displays. Now we are moving to a human interface where you simply have a conversation with the device."

It's also important to ease people into AI as part of their natural interaction with electronic devices, alongside touching buttons and screens. "We can easily become distracted by personification of these kinds of systems," says Blaise.

It's important for makers to realise that VUIs are something they can create, and use, in their projects. "I think letting the makers see how easy it is to put AI, specifically Natural Human Interaction capabilities, into their projects will be a great thing," says Kristine.

"We want to show you how easy it is to use AI, and then share back with us to inspire new project ideas and keep the whole cycle going," Kristine continues.

> "We want to show you how easy it is to use AI"

"My top-secret plan is to build more engineers," discloses James McLurkin, Senior Hardware Engineer of AIY Projects at Google. "Getting kits like this out into the world with Raspberry Pi allows us to build the things that then create more engineers." AIY Projects enables young makers to explore the possibilities with AI. "So this is very exciting for us," says James.

"What's interesting about the maker environment is what happens when we shut up, and listen, and see what people try," says Blaise. Historically, there have been many 'Hello World' types of starter projects for various programming languages and platforms, and in recent years we've seen exciting new hardware like the Raspberry Pi emerge. Now there is AI, another technology for makers and developers to add to their projects. AIY Projects brings these three things together, which will be "super interesting," reckons Blaise.

"I don't know what will come out from the mixture of those, but I'm very keen to see."

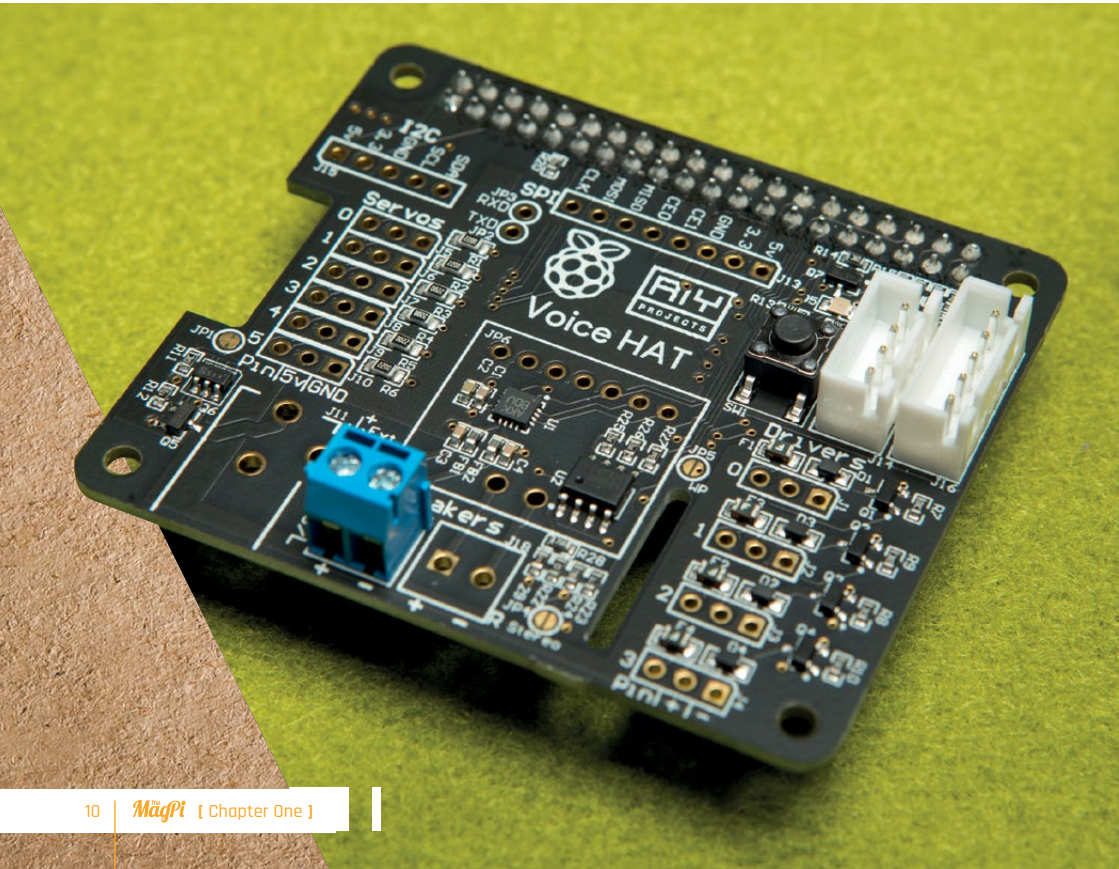
"I view this as an essential component in a maker's toolkit," Billy tells us. "We want makers to see that using AI is not hard or complex."

The Australian artist Stelarc has said that technology constructs our human nature. "We would not be who we are if we hadn't invented fire and woven clothes and built Raspberry Pis," says Blaise. "That is what being human is all about, and that's what distinguishes us from the other animals. So I don't like this idea that talks about AI as a competitive landscape of human exceptionalism, and ways that it is being eroded. That really misses the point of what all this is about."

"This first kit showcasing voice is just the start of our effort to bring Google AI to the maker community," reveals Billy. "Our projects will largely focus on Natural Human Interaction." Following voice, we intend to feature projects with vision, motion, and learning."

Google wants makers to add AI to their own projects, and share their results with others. "We want to learn what this community needs," says Billy, "and then work with them to build the tools they want."

Below The Voice HAT hardware



The MagPi ESSENTIALS

[CHAPTER TWO]

YOUR AIY PROJECTS VOICE KIT

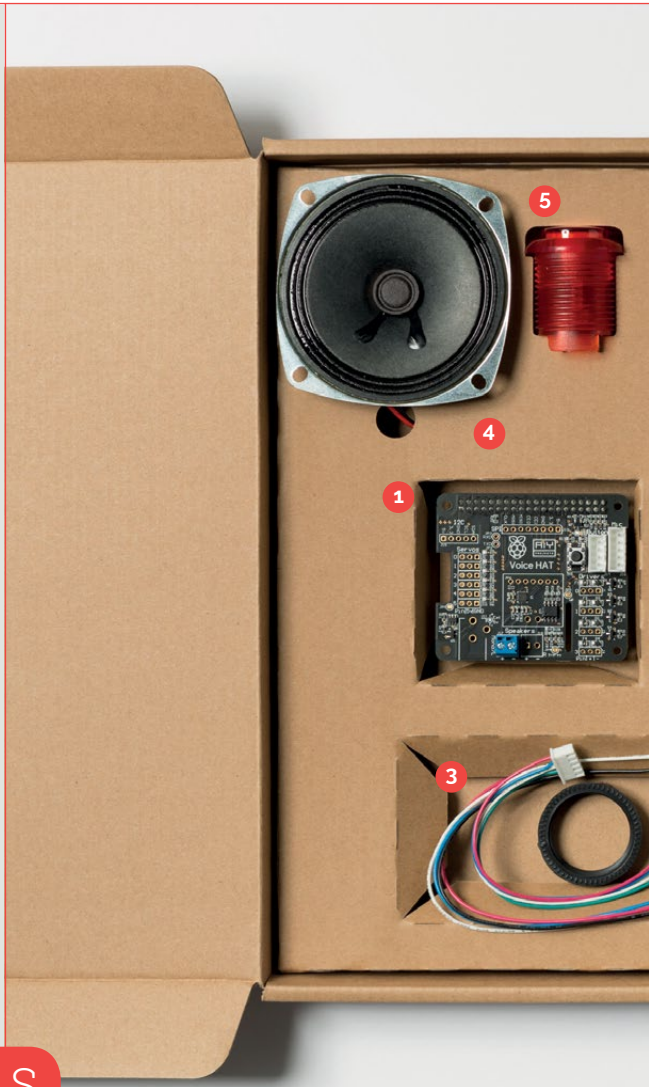
Construct your AIY Projects voice kit and explore natural language recognition

Inside the kit will be the components you need to build a voice-capable device with Raspberry Pi.

Open the box and you'll find two pieces of cardboard, an arcade-style button, a speaker, and some cables, along with a HAT (Hardware Attached on Top) board and another narrower board. One is to connect all the accessories together; the other is a stereo microphone.

All of these components fit together to build the AIY Projects kit: a small cardboard device with a colourful button on the lid. You press the button, or clap your hands (or create a custom trigger), and speak out loud to ask the device a question. The speaker, at the front, then announces the answer.

Use the Bill Of Materials list below to check you have all the components.



BILL OF MATERIALS

1. Voice HAT accessory board
2. Voice HAT microphone board
3. 2× plastic standoffs
4. 3-inch speaker (wires attached)
5. Arcade-style push button
6. 4-wire button cable

7 5-wire daughter board cable
8 External cardboard box
9 Internal cardboard frame

The MagPi ESSENTIALS

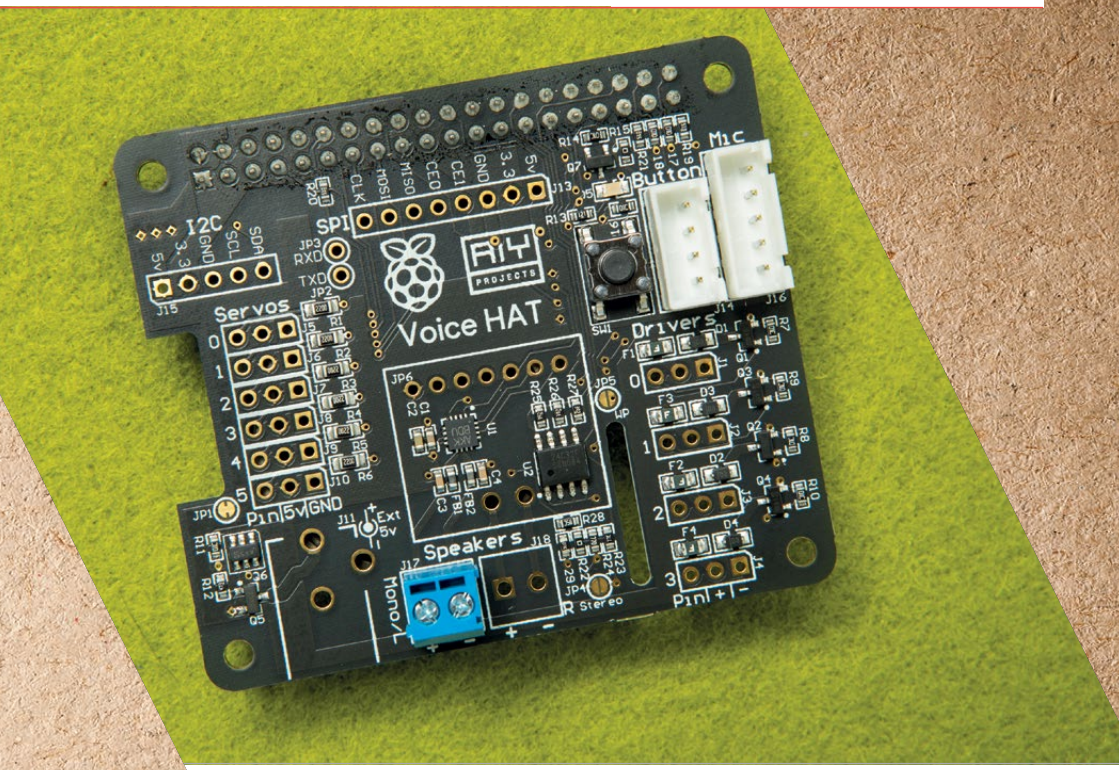
[CHAPTER THREE]

ASSEMBLE THE KIT

Put the parts together to build a voice-enabled device

You'll Need

- Raspberry Pi 3
- Small, needle-nose pliers
- Phillips 00 screwdriver
- Two-sided tape



With all your parts ready, it's time to build the AIY Projects voice kit. The aim is to assemble all the included parts (and a Raspberry Pi board) and create a small cardboard device with a button on top.

This project is a relatively easy build, and you won't need to solder any of the components. Be careful to line up the wires correctly, especially the wires for the button. It's also a good idea to take a close look at the Voice HAT accessory board (the larger board). The Voice HAT is the heart of the AIY Projects kit, and everything connects to it. It also provides breakout GPIO pins, organised into two blocks: Servos and Drivers.

You'll connect the Voice HAT accessory board to your Raspberry Pi via the GPIO pins. The Raspberry Pi is the brains of the outfit: it connects to Google's cloud services through a local Python application. The Python source code is provided with the software image, as well as on GitHub.

Also take a close look at the smaller microphone board, which enables the device to hear you speak.

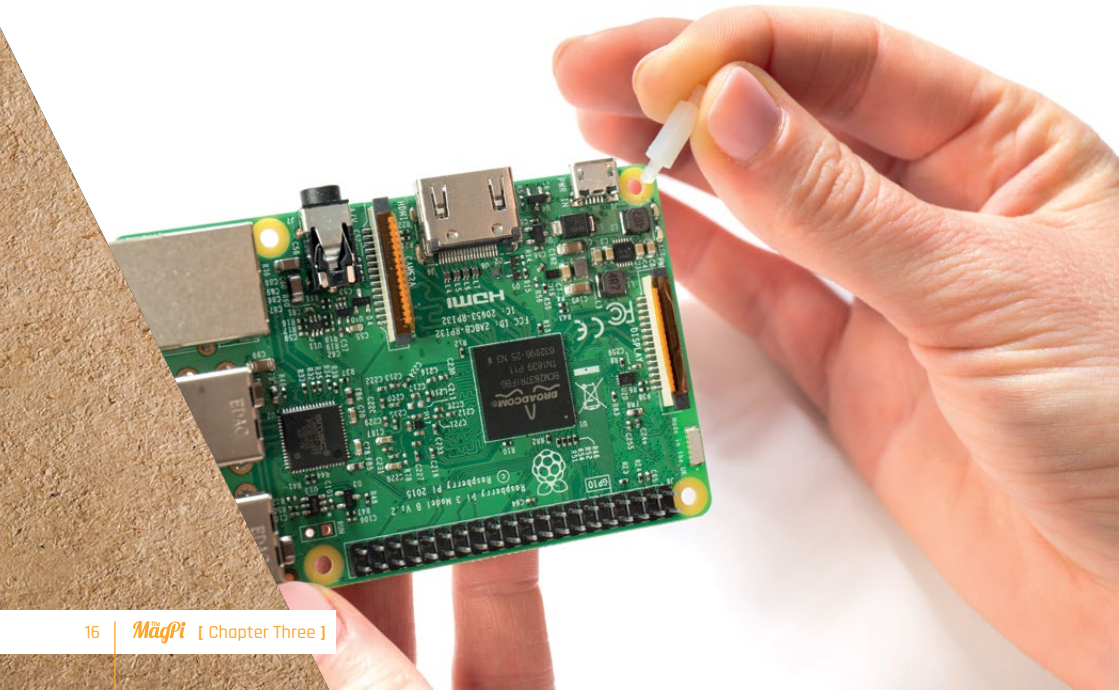
But first, we need to get it all assembled. The first step is to mount the Voice HAT accessory board to your Raspberry Pi, and then connect the speaker and microphone. Then you'll move on to folding the cardboard case and placing the components inside. Finally, you'll assemble the arcade-style button and secure it (and the microphone) to the case.

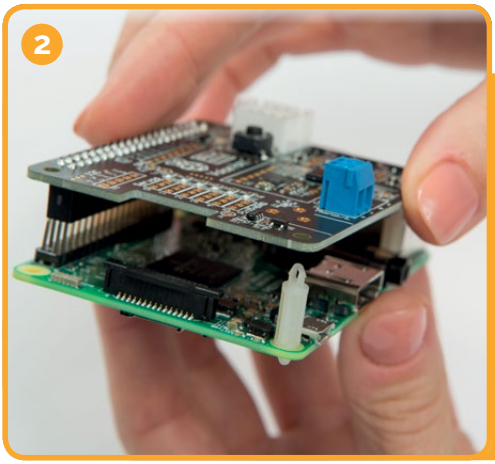
Ready? Let's start building your kit.

SET UP THE VOICE HAT

1 INSERT THE STANDOFFS

Start with the two standoffs. These are the small plastic cylinders, and they fit into the yellow mount holes on the Raspberry Pi board. Insert the standoffs into the two yellow holes on the opposite side from the 40-pin GPIO header (on the same side as the HDMI connection). Push them firmly, and they will hold in place.



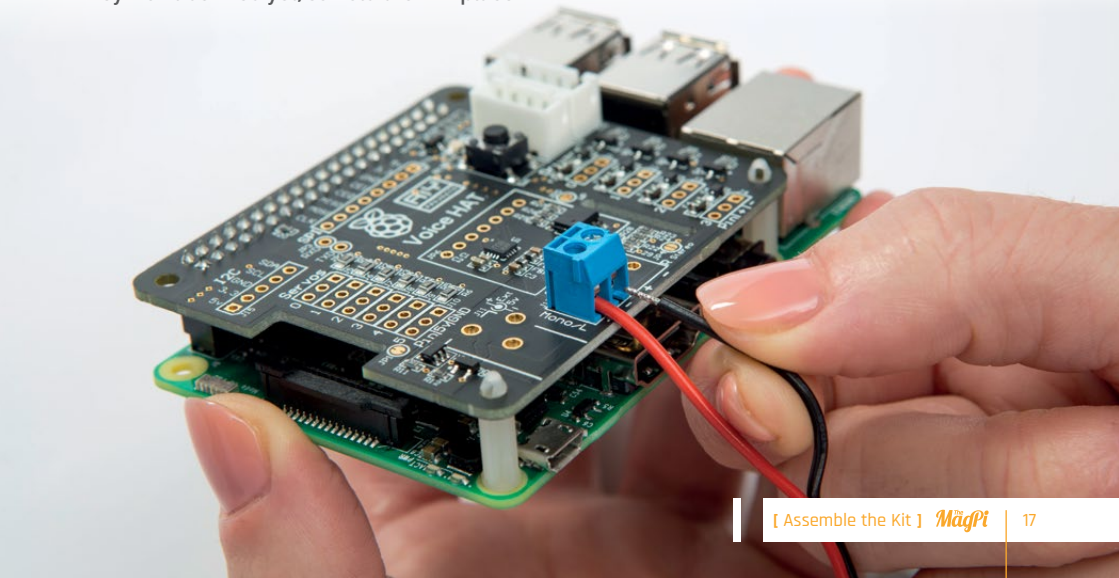


SECURE THE HAT

Now get the Voice HAT accessory board and attach it to the GPIO pins on the Raspberry Pi board. Carefully line up the GPIO connector on the Voice HAT accessory board with the pins of the GPIO header on the Raspberry Pi. Gently press down to make sure the Voice HAT accessory board is secure. Press down on the spacers on the other side of the board to snap the boards together.

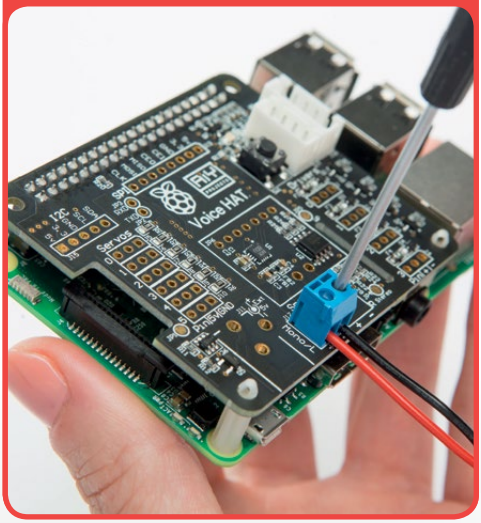
3 ATTACH THE SPEAKER WIRES

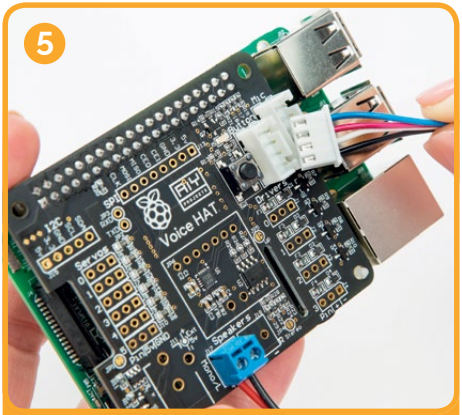
Take a close look at the Voice HAT accessory board and find the blue terminal with two small screws. This terminal is the speaker connection (it has 'Speakers' printed above it on the board). Each of the two connections has a small '+' and '-' symbol printed below. Find the speaker with the red and black wires attached. Insert the red wire into the positive '+' terminal on the Voice HAT accessory board. Now add the black wire into the negative '-' terminal. They won't be fixed yet, so hold them in place.



4 SCREW IN THE WIRES

At this point, the two wires will be sitting in the sockets unsecured. Hold the wires in place, and gently turn each screw in the socket using a Phillips 00 screwdriver. Gently tug on the wires to make sure they're secure. Now place the speaker to one side of the board so you can access the other components.



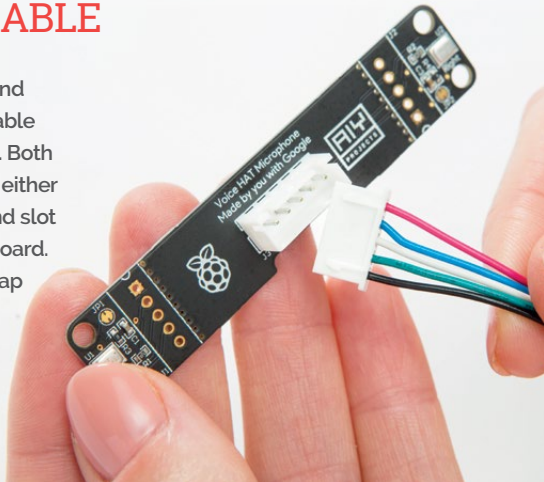


THE BUTTON CABLE

Find the 4-wire button cable: it has a white male connector on one end and four separate wires with metal contacts on the other. Insert the white plug into the matching white socket marked 'Button' on the Voice HAT accessory board (it is the one nearest to the red button). The cable will only go in one way around, so don't force it. Check that the colours of the cable match the image. Don't worry about the four separate wires with metal contacts; we'll come back to these later.

6 THE MICROPHONE CABLE

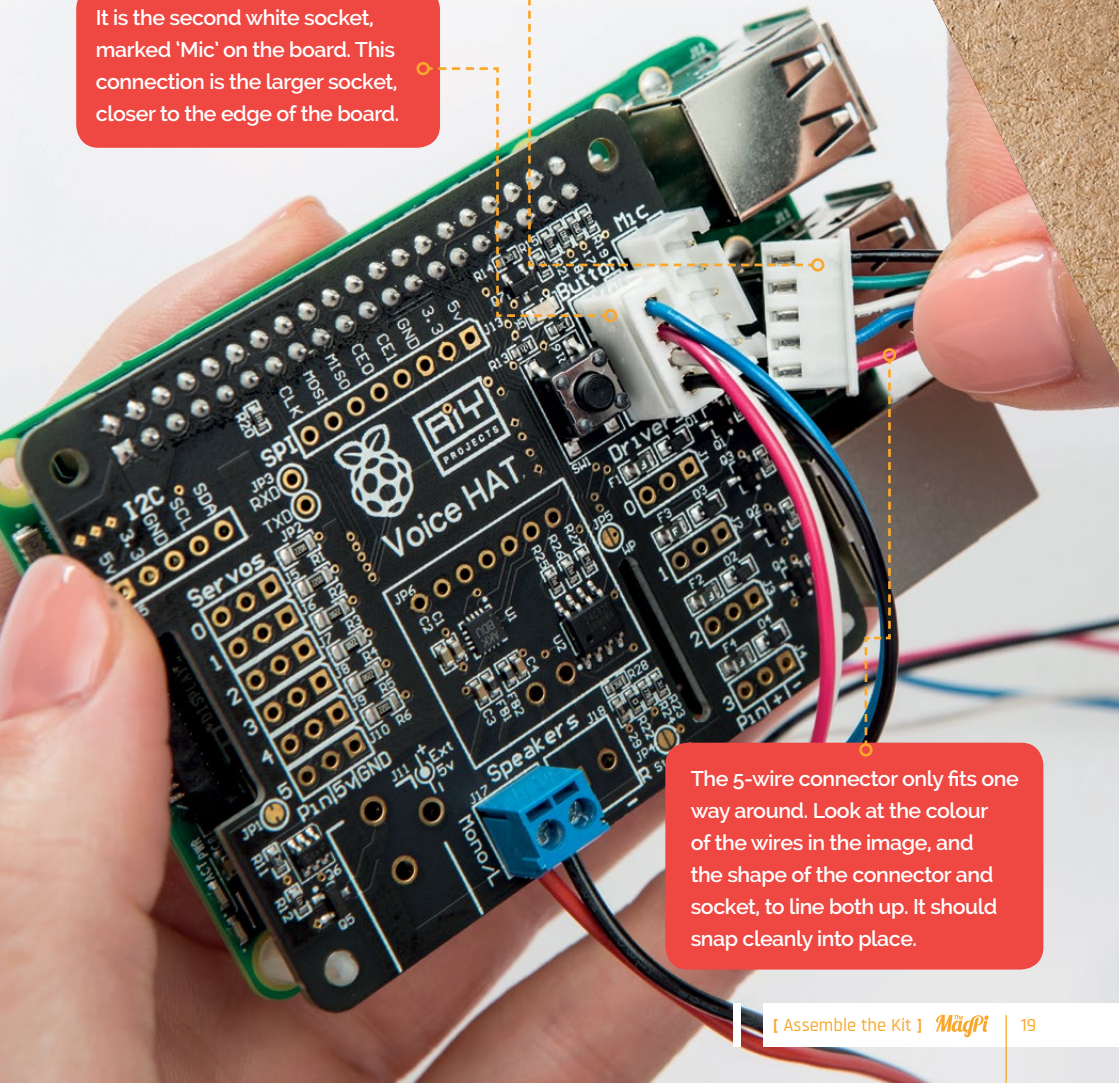
Find the Voice HAT microphone board and the 5-wire daughter board cable. The cable has matching white plugs on either end. Both ends of the cable are identical, so take either end of the 5-wire connector cable and slot it into the Voice HAT microphone board. It will only fit one way around. Snap the cable in, but don't force it.



7 CONNECT THE MICROPHONE

Take the other end of the 5-wire daughter board cable and connect it to the Voice HAT accessory board.

It is the second white socket, marked 'Mic' on the board. This connection is the larger socket, closer to the edge of the board.



The 5-wire connector only fits one way around. Look at the colour of the wires in the image, and the shape of the connector and socket, to line both up. It should snap cleanly into place.

ASSEMBLE THE BOX

8

FOLD THE CARDBOARD

Now let's move on to the box. Find the larger cardboard piece with a bunch of holes on one side (as shown in the image). Fold along the creases, then find the side with four flaps and fold the one marked FOLD 1.

9 SECURE THE BOX

Do the same for the other folds, tucking FOLD 4 underneath to secure it in place. Now set it aside.

10 FOLD THE FLAPS

Find the other cardboard piece that came with your kit (as shown in the picture). This piece will build the inner frame to hold the hardware. Fold the flaps labelled 1 and 2 along the creases.

11 PUSH IT OUT

The flap above the 1 and 2 folds has a U-shaped cutout. Push it out.

12 FOLD OUT THE FLAP

Then fold the rest of the flap outward. Fold the section labelled FOLD UP so that it is flush with the surface you're working with. There's a little notch that wraps behind the U-shaped flap to keep it in place.

13 CHECK THE FLUSH

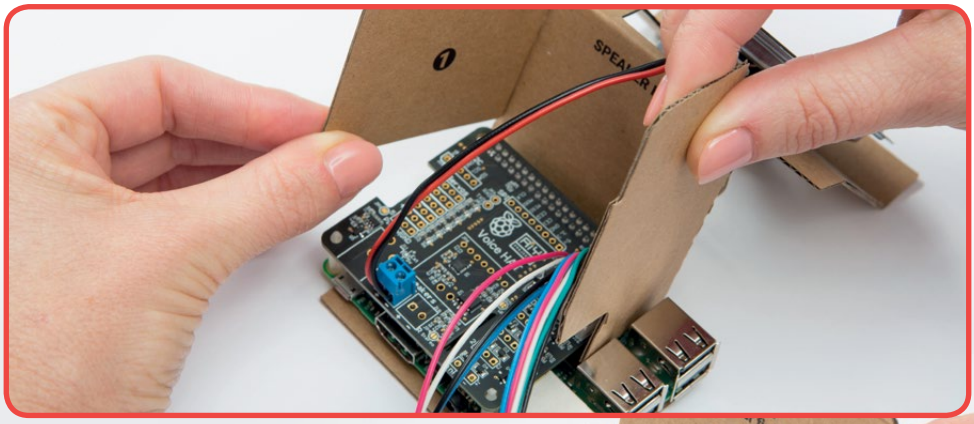
The U-shaped flap should lie flush with the box side. At this point, the cardboard might not hold its shape. Don't worry: it'll come together once it's in the box.

14 ADD THE SPEAKER

Find your speaker (which is now attached to your Raspberry Pi 3). Slide the speaker into the U-shaped pocket on the cardboard frame.

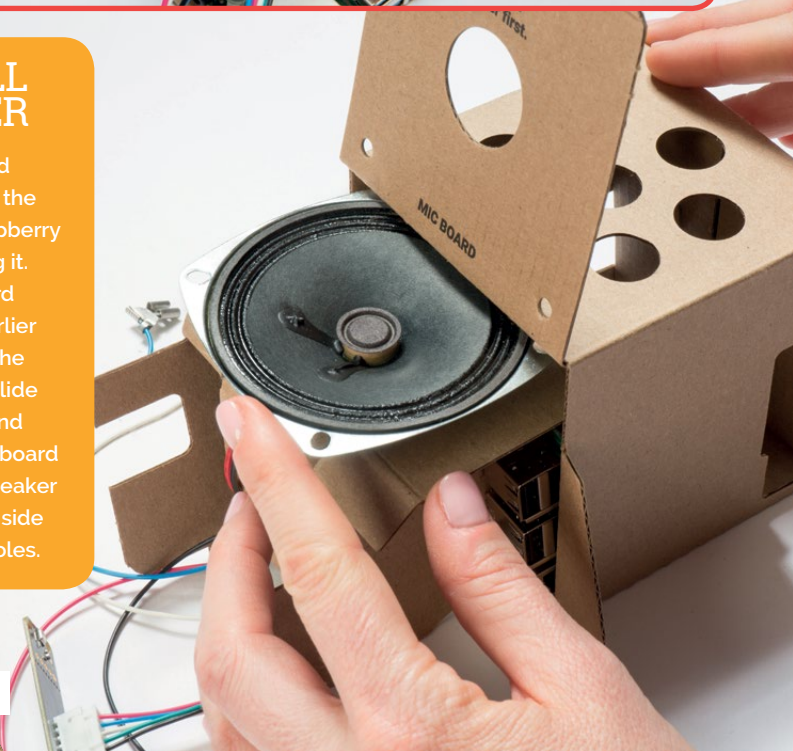
15 SLIDE INTO THE RASPBERRY PI

Turn the cardboard frame around. Take the Pi + Voice HAT hardware and push it into the bottom of the frame below flaps 1 and 2 (pictured). The cardboard frame should expose the USB ports of the Raspberry Pi.



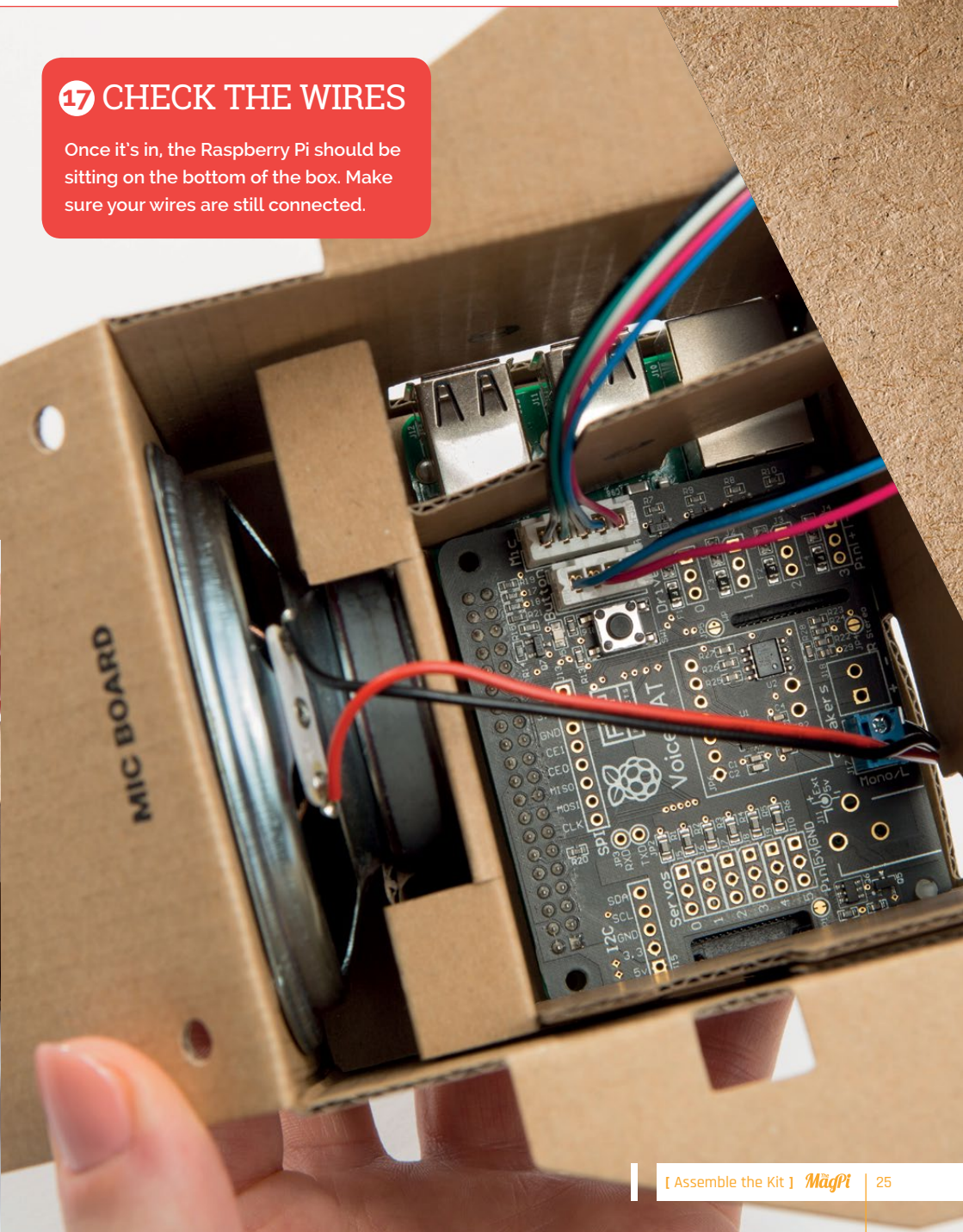
16 PUT IT ALL TOGETHER

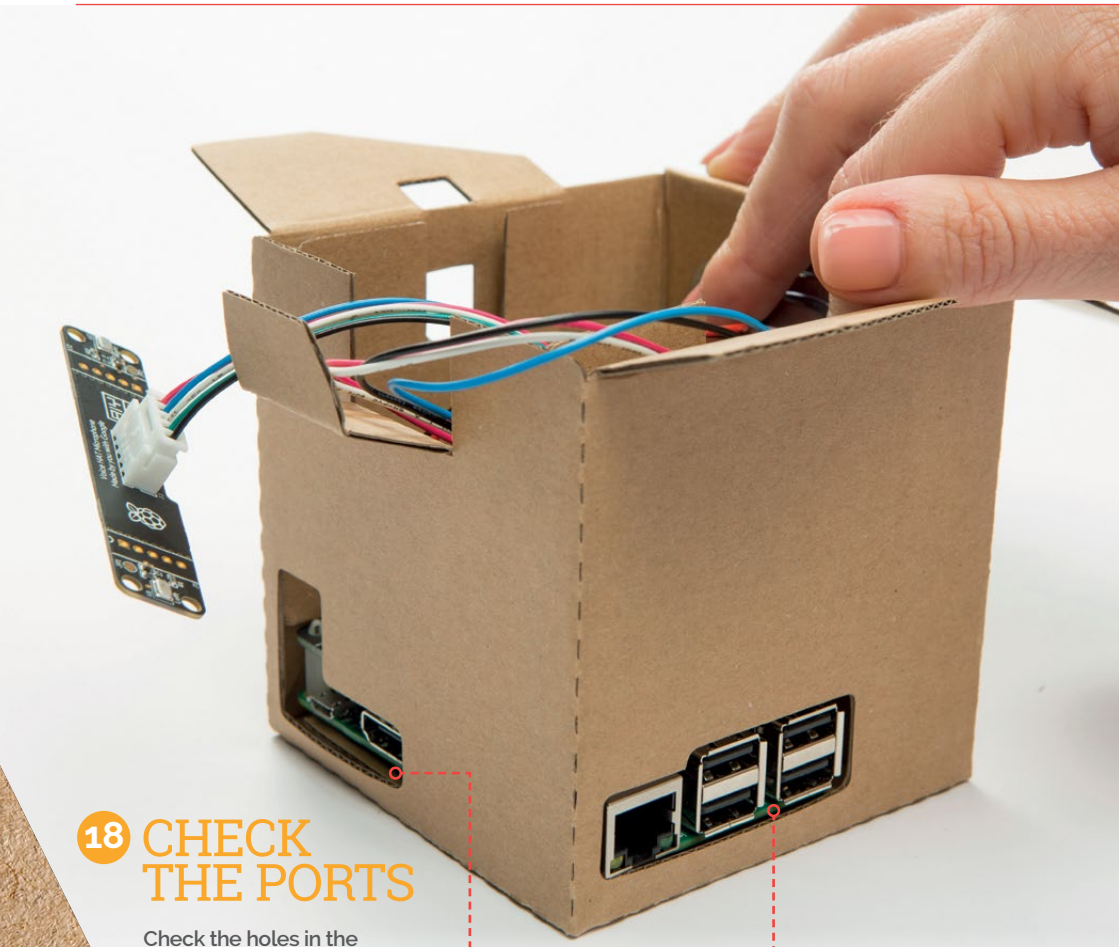
It's time to put the build together. First, remove the SD card from your Raspberry Pi to prevent damaging it. Now take the cardboard box you assembled earlier and find the side with the seven speaker holes. Slide the cardboard frame and hardware into the cardboard box. Ensure that the speaker is aligned with the box side that has the speaker holes.



17 CHECK THE WIRES

Once it's in, the Raspberry Pi should be sitting on the bottom of the box. Make sure your wires are still connected.





18 CHECK THE PORTS

Check the holes in the cardboard box. The Raspberry Pi ports should be clearly visible.

The AIY Projects voice kit is designed to work without a display, but you can access the HDMI socket for troubleshooting. This hole also provides access to the power socket.

One hole provides access to the USB ports. These ports enable you to hook up a keyboard and mouse to the AIY Projects kit, although it is designed to be controlled hands-free with your voice.

19 ADD THE BUTTON

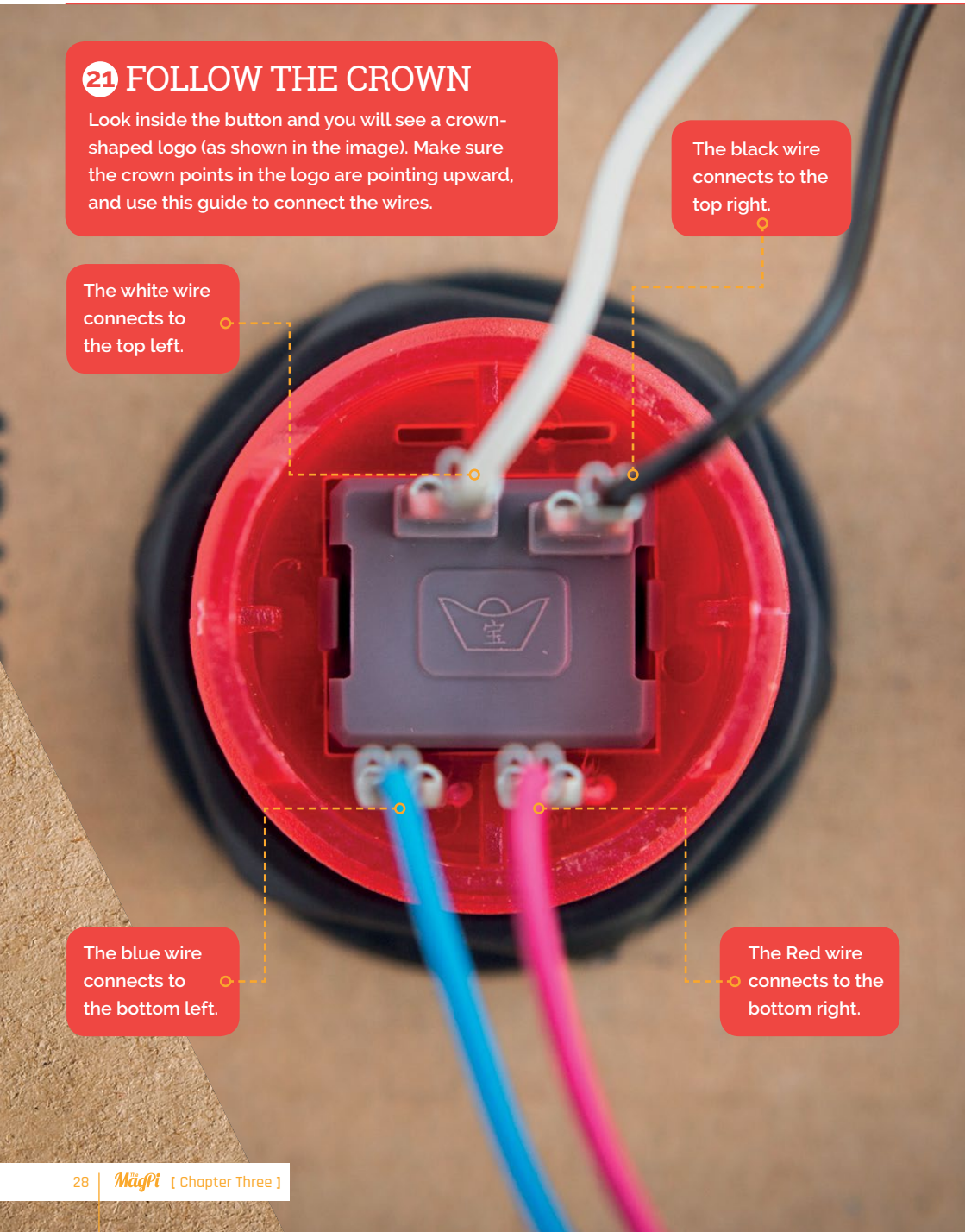
Insert the plastic button into the top flap of the cardboard box from the outside in. The pushable button side should face outward, with the larger screw on the inside; i.e. the side marked 'BUTTON.'

20 SECURE THE BUTTON

Now, screw in the washer nut to secure the button to the cardboard lid. Carefully screw the plastic nut around the thread of the button to firmly hold it in place.

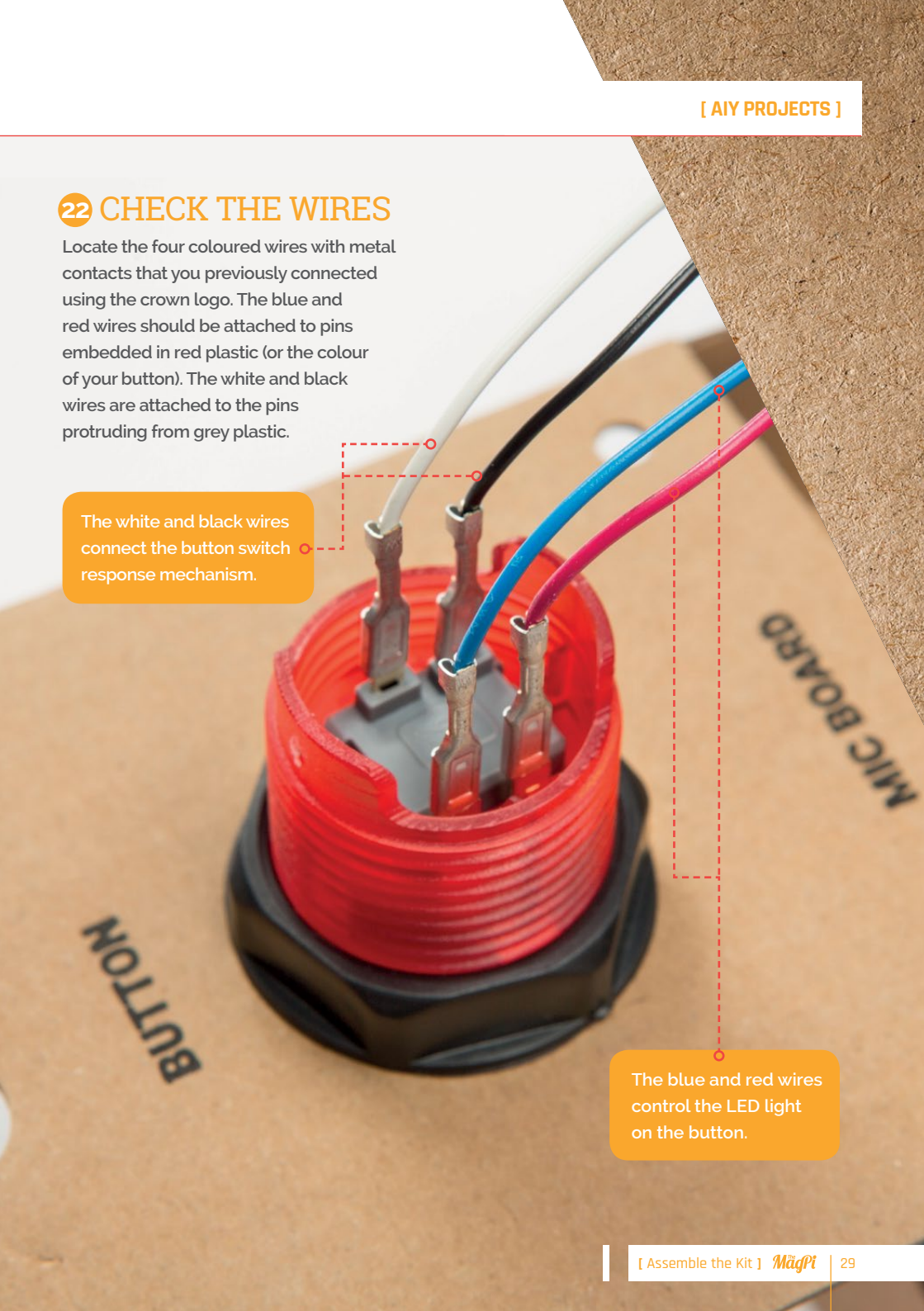
21 FOLLOW THE CROWN

Look inside the button and you will see a crown-shaped logo (as shown in the image). Make sure the crown points in the logo are pointing upward, and use this guide to connect the wires.



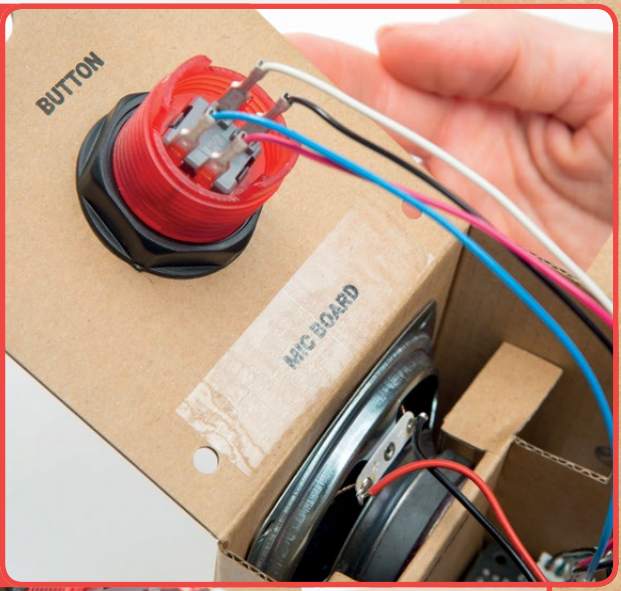
22 CHECK THE WIRES

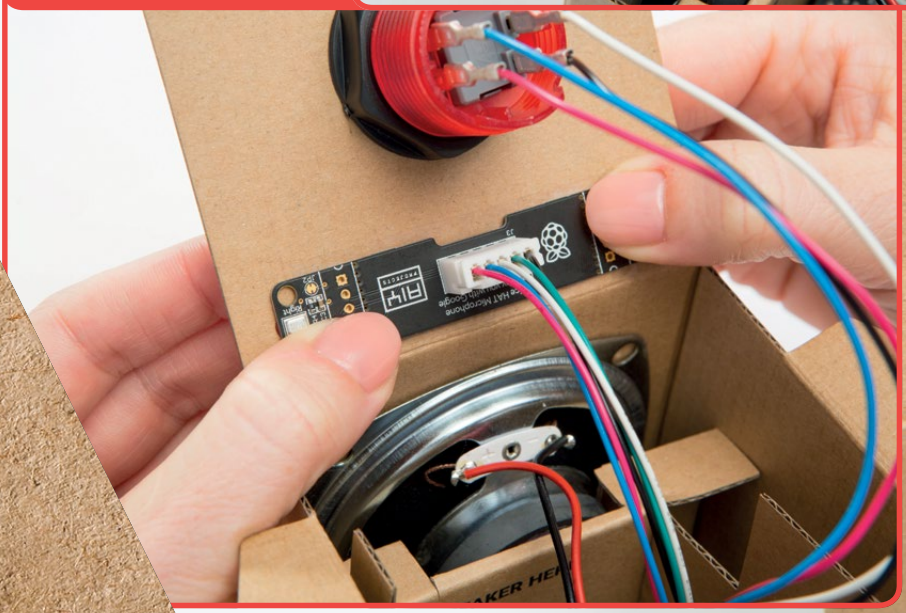
Locate the four coloured wires with metal contacts that you previously connected using the crown logo. The blue and red wires should be attached to pins embedded in red plastic (or the colour of your button). The white and black wires are attached to the pins protruding from grey plastic.



23 TAPE THE MICROPHONE

Next, we use two-sided tape to secure the Voice HAT Microphone board to the top flap. You can also use a spot of hot glue if you don't have two-sided tape. The board sits below the button on the top flap, with the two microphones aligned with the two holes. Check that the holes, on the other side, are aligned with the two microphones before fixing down the board.





24 CHECK THE MICROPHONE

Turn the flap around, and double-check that the microphones match the cardboard holes. Correct alignment ensures that the Microphone board can clearly hear you when you start issuing voice commands.

25 THE FINISHED BUILD

That's it. Your voice kit is assembled, and you can now start installing the software and using the Google Assistant to answer your questions. Fold the top flap down to close the box up and admire your handiwork.

The MagPi ESSENTIALS

Google Assistant SDK

[CHAPTER FOUR]

SET UP THE SOFTWARE

Download and set up the AIY Projects software and connect your device to the internet

You'll Need

- Assembled AIY Projects Voice Kit
- USB keyboard
- Mouse
- HDMI monitor
- HDMI cable
- MicroSD card
- AIY Projects image file

Info & Updates:

aiyprojects.withgoogle.com/voice

You now have a fully assembled cardboard device that is almost ready to respond to your questions. Now that your box is assembled, we will begin the process of turning it into a Voice Assistant, and an intelligent voice-powered interface for your own projects.

To do this, you'll set up a Google Developer project and activate the brand-new Google Assistant SDK.

But first, you need the base to work with. And that's a custom operating system designed especially for the AIY Projects kit.

>STEP 01
Download image
First, you need to download the AIY Projects image from **magpi.cc/2x7JQfS**. Please check the website for any updates to this process. AIY Projects software is routinely updated. Click the Get the Voice Kit SD Image link. The image file is saved to your Downloads folder.

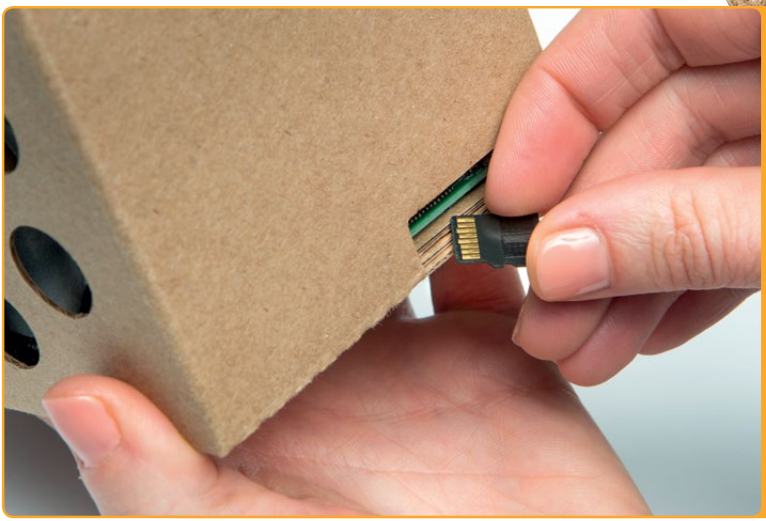
>STEP 02
Copy image
Burn the image to a microSD card using a program like Etcher (**etcher.io**) on a Mac, Windows, or Linux computer.

Etcher software copies the image to the SD card (see 'Burn SD cards with Etcher', **magpi.cc/2fZkyJD**, if you're unfamiliar with the process).

>STEP 03
Plug in peripherals
Now that your box is assembled, plug your peripherals in:

1 USB keyboard
2 USB mouse
3 HDMI monitor

>STEP 04
Insert SD card
Insert your SD card (the one with the Voice Kit SD image) into the slot on the bottom side of the Raspberry Pi board. The SD card slot should be accessible through a cutout provided in the external cardboard form.



The SD card can be tricky to remove after it's been inserted. We recommend using either small, needle-nose pliers to remove it, or attaching tape to the SD card before inserting so you can remove it by pulling the tape.

>STEP 05

Power up

With the microSD card inserted into the Raspberry Pi, and the peripherals (monitor, keyboard, and mouse) connected, plug in the power supply. The Raspberry Pi will begin booting up, and you should see the AIY Projects desktop.

>STEP 06

Check LED lights

Once booted, the small LED in the centre of the Voice HAT and the LED inside the arcade button should both indicate the device is running.If you have any problems booting, check the troubleshooting guide in the appendix. If you don't see anything on your monitor, or you see 'Openbox Syntax Error', check the troubleshooting guide at the end.

>STEP 07

Connect to network

Click the network icon in the upper right corner of the Raspberry Pi desktop. Choose your preferred wireless access point. Enter the wireless LAN password in the Pre Shared Key box and click OK.

>STEP 08

Check network

Double-click the Check WiFi icon on your desktop. This script verifies that your WiFi is configured and working properly on the Raspberry Pi board.

If everything is working correctly, you'll see a confirmation message. Press **ENTER** to close.

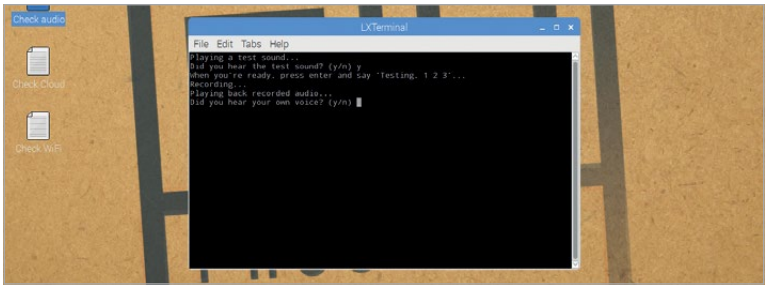
>STEP 09

Check speaker

Double-click the Check Audio icon on your desktop. This script verifies the audio input and output components on the Voice HAT accessory board are working correctly.

When you click the script, you should hear "Front, Centre" announced from the speaker. An LXTerminal window opens with 'Did you hear the test sound? (y/n)'.

Enter **y** if you heard the sound. Now press **ENTER** to test the microphone.



>STEP 10

Check microphone

Say "Testing, 1 2 3" out loud. It will play back your voice with the message 'Did you hear your own voice (y/n)'. Again, enter **y** and press **ENTER**. Press **ENTER** again to end the test.

If you see an error message, follow the message details to resolve the issue and try again.

TROUBLESHOOTING TIPS

- A red LED on the Raspberry Pi near the power connector should light. If it doesn't, unplug the power, unplug the connector to the microphone, and power-up again. If it lights after powering-up without the microphone, then the microphone board may be defective.
- If the lamp in the button doesn't light up, it might be the wrong way around. Take the lamp out of the button (see Chapter 3), turn it 180°, and put it all back together. If it still doesn't light, check that the wire colours are the same as the picture in Chapter 3 step 12.
- If you don't see anything on your monitor, make sure the HDMI and power cables are fully inserted into the Raspberry Pi.
- If you see 'Openbox Syntax Error', you'll need to rewrite the image to the SD card and try booting the device again.

The MagPi ESSENTIALS

Google Assistant SDK

[CHAPTER FIVE]

BUILD A VOICE RECOGNIZER

Use the Google Assistant SDK to create a device that answers your questions and helps you get things done

Congratulations on assembling your voice recognizer device – now, let's bring it to life! We're going to build a voice recognizer that uses Google Assistant, much like Google Home.

The voice recognizer uses the Google Assistant SDK to recognise speech, along with a local Python application that evaluates local commands.

Your voice recognizer will let you talk to the Google Assistant, and it will respond with smart answers to your questions.

>STEP 01

Google Cloud Platform

To try the Google Assistant API, you need to first sign into Google Cloud Platform (GCP) and then enable the API.

Quick Tip

Use your Google account to sign in to the Google Cloud Platform. If you don't have one, you'll need to create one. Trying the Google Assistant API is free to use for personal use.

>STEP 02

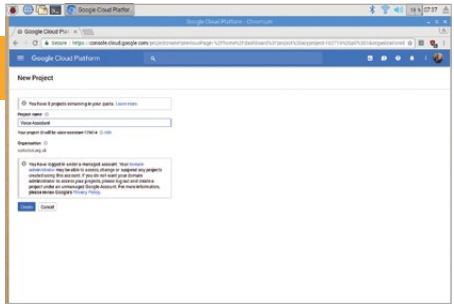
Log into GCP

Using AIY Projects on your voice recognizer device, open up the Chromium web browser (click on the blue globe icon in the top bar of the desktop). Go to the Cloud Console (**console.cloud.google.com**). Enter your Google account ID and password.

>STEP 03

Create a project

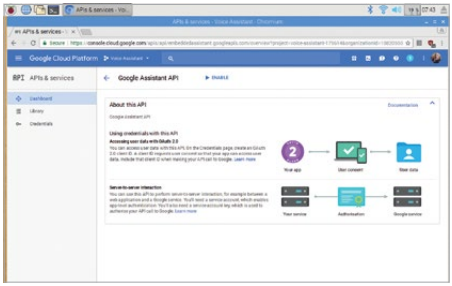
GCP uses projects to organise things, so you'll need to create a new project for your AIY Voice Kit. In Cloud Console, click the drop-down button to the right of the Google Cloud Platform logo (in the top-left of the screen). Now choose Create Project.



>STEP 04

Name the project

Enter a project name, such as 'Voice Assistant' and click Create. After your project is created, make sure the drop-down has your new project name displayed (if not, click on it and choose it from the list of projects).

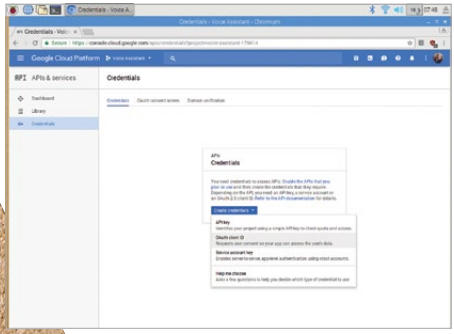


>STEP 05

Turn on the API

Click Product & Services (the triple line icon) in the top-left of the GCP. Choose APIs & services and Dashboard and click Enable APIs.

Enter 'Google Assistant API' into the Search box and click it from the list below. Now click Enable.



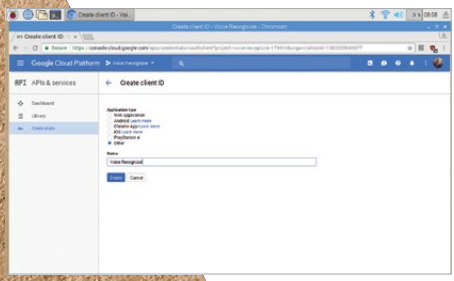
>STEP 06

Create credentials

In the Cloud Console, create an OAuth 2.0 client by going to APIs & Services > Credentials.

Click on 'Create credentials' and choose OAuth client ID.

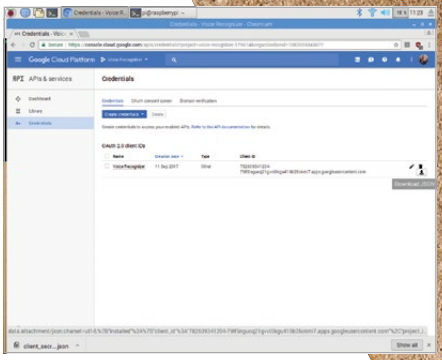
If this is your first time creating a client ID, you'll need to configure your consent screen. Click 'Configure consent screen'. Enter a Product Name, such as Voice Assistant. Click Save.



>STEP 07

Name credentials

Select Other. It will have the default name 'Other client 1'. Change the name to 'Voice Recognizer' to help you remember the credentials. Click Create.



>STEP 08
Client ID and secret

A window will pop up, named 'OAuth client', with 'Here is your client ID' and 'Here is your client secret'. Don't worry about memorising the long numbers, just click OK.

>STEP 09
Download JSON

In the Credentials list, find your new credentials and click the Download JSON icon. If you don't see the download icon, try expanding the width of your browser window or zooming out (**CTRL+–**). A JSON file starting with 'client_secrets' is saved to your Downloads folder.

>STEP 10
Find credentials

Open a Terminal window (click Terminal in the taskbar) and enter:

```
cd Downloads
ls
```

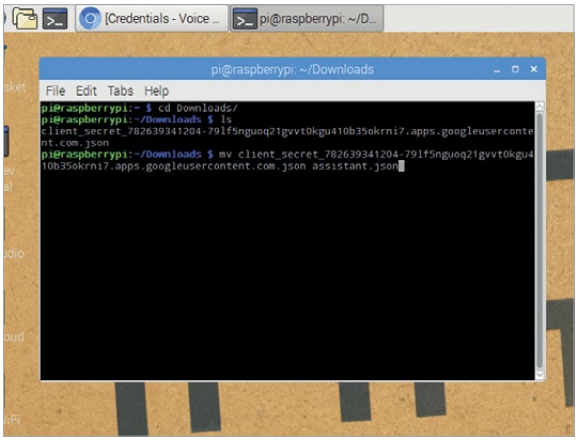
...to view the client secret file. It will have a lot of numbers and end with .json.

>STEP 11
Rename the credentials

You need to rename the file to **assistant.json** and move it to your home directory. Enter:

```
mv client_secret
```

...and press the **TAB** key. This will fill out the rest of the letters in the file. Now add **/home/pi/assistant.json** to the end of the file and press **ENTER**.



androidthings

Android Things is a new OS for connected devices that is fully compatible with the AIY Projects Voice Kit.

Developed by Google, it is a ready-to-use solution for building connected devices.

Developers can use existing Android development tools, security updates, APIs, resources, and a thriving developer community. It also includes new Android framework APIs that provide low-level I/O and libraries for common components like temperature sensors and display controllers.

In addition, a wide range of Google APIs and services – such as Google Play services, TensorFlow, and Google Cloud Platform – are available on Android Things. Developers can push Google-provided OS updates and their own app updates, using the same OTA infrastructure used on Google's own products.

To get started on building your kit with Android Things, visit the AIY Projects website (**aiyprojects.withgoogle.com/voice**). More information about Android Things is available on the developer website (**developer.android.com/things**).

This is how the full command looks on our AIY Projects voice recognizer (your client secret file will be different):

```
mv client_secret_782639341204-791f
5nguoq21gvvt0kgu410b35okrni7.apps.
googleusercontent.com.json /home/pi/
assistant.json
```

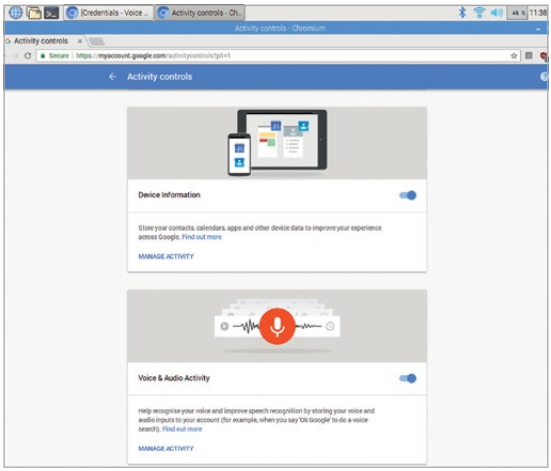
>STEP 12

Activity controls

Return to Chromium and visit your Google Activity Controls at **myaccount.google.com/activitycontrols**. Make sure to log in with the same Google account as before.

Turn on the following by ticking the slider to the right, so they appear blue:

- **Web and app activity. Make sure the 'Include Chrome browsing history and activity from websites and apps that use Google services' checkbox is ticked).**
- **Device information**
- **Voice and audio activity**



GOOGLE DEMO APPS

Demo App	Description	Raspberry Pi supported
assistant_library_demo.py	Showcases the Google Assistant Library and hotword detection ("Okay, Google")	2B, 3B
assistant_grpc_demo.py	Showcases the Google gRPC APIs and button trigger	2B, 3B, Zero W
cloud_speech_demo.py	Showcases the Google Cloud Speech APIs, button trigger, and custom voice commands	2B, 3B, Zero W

>STEP 13

Start the voice unit

Double-click the 'Start dev terminal' icon and enter:

```
src/assistant_library_demo.py
```

A 'Request for Permission' window appears. Click Allow and close the web browser window. Return to the terminal window and you will see 'Say "OK, Google" then speak, or press Ctrl+C to quit…'

>STEP 14

The demo apps

In the Assistant Library demo, you can use the AIY Projects kit as a voice assistant. Say "OK Google" and ask a question out loud, such as "what is the weather in Cambridge?". Press **CTRL+C** when you're done.

Google provides three demo apps that showcase voice recognition and Google Assistant with different capabilities. They may be used as templates to create your own apps.

POWER OFF CAREFULLY

Take care to always turn off the Raspberry Pi using **Menu > Shutdown > Shutdown** or **sudo shutdown -h now** in Terminal. If you want to use your AIY Projects kit without a screen connected, you should add the **shutdown_demo.py** code from Chapter 7 and issue the "shut down" voice command before disconnecting the power.

The MagPi ESSENTIALS

Google Cloud Platform

[CHAPTER SIX]

CREATE A VOICE USER INTERFACE FOR YOUR PROJECTS

Swap out traditional interfaces with a custom voice control using your AIY Projects Voice Kit

You'll Need

- Google Cloud Speech API

By now, you have built a device that embeds the Google Assistant. That's cool, but it's just the beginning. With Google Cloud Speech API, you can create an interactive, custom voice-user interface (VUI) for your project.

This enables you to explore a new generation of devices that you can have a conversation with, without the need for remote-control devices (such as joysticks or smartphone apps). Let's reconfigure the kit to use the Google Cloud Speech API.

> STEP 01

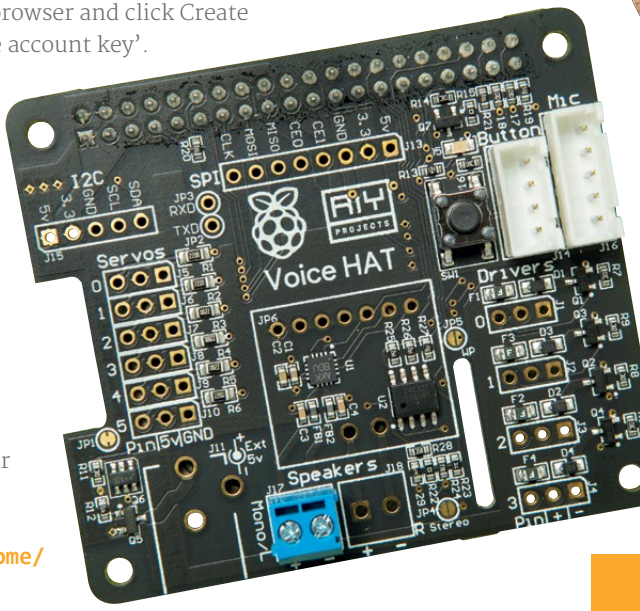
View the source

The source code for the voice recognizer app is part of the image that you've just installed. You can view the Python source code in the **/home/pi/AIY-voice-kit-python/src** directory. Alternately, the project source is available on GitHub: **github.com/google/aiyprojects-raspbian/tree/voicekit**. It is released under the 'voicekit' branch.

>STEP 02

Create service account

Head to Google Cloud Console in the browser and click Create Credentials. This time choose 'Service account key'. Click the 'Service account' menu and choose 'New service account'. Give it a name, such as 'AIY Projects', and change the Role to Project > viewer. Make sure the Key type is JSON and click Create. The key is downloaded to your computer.



>STEP 03

Create credentials

Find the file you've downloaded. You need to rename the file to **cloud_speech.json** and place it in your home folder.

```
mv My Project-[123etc].json /home/
pi/cloud_speech.json
```

TensorFlow

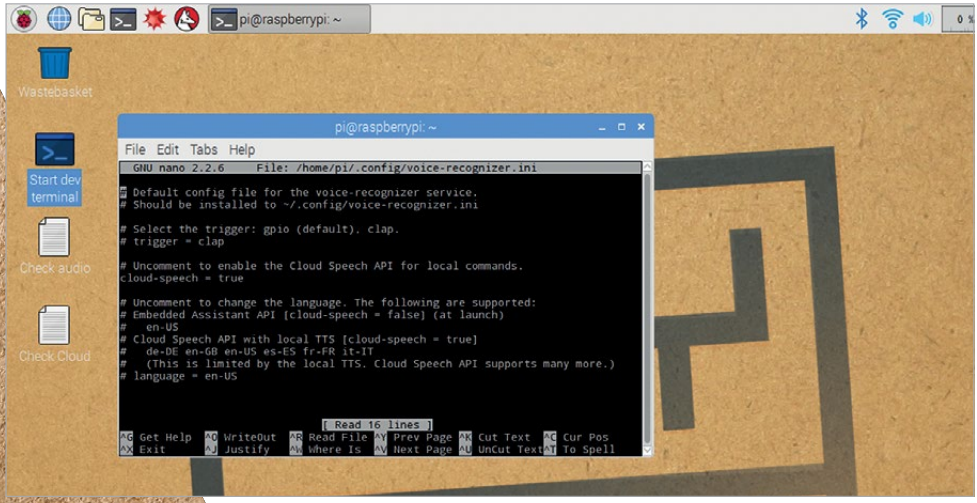
TensorFlow is an open-source software library for machine learning. It was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organisation. You can learn more about TensorFlow, and how it can be used to add intelligence to your own projects, at **tensorflow.org**. To see how you can use TensorFlow to enable on-device audio detection, visit the AIY Projects website at **aiyprojects.withgoogle.com/voice**.

> STEP 04
Check Cloud Speech API

You need to have the Cloud Speech API enabled to use the service. In Cloud Console, go to API & Services > Library. Click on Speech API and click Enable (then follow the prompts to set up a billing account, or follow the instructions in step 6, then click Enable again).

> STEP 05
Check billing

You need to have billing set up with Google to use the Cloud Speech API. Open Cloud Console, click 'New billing account' and go through the setup. Check your project is selected in the Projects menu at the top. Click Products & Services > Billing. To connect or change the billing account, click the three-dot button, then select 'Change billing account'.



NOTE FOR EU USERS

At this time, the Cloud Speech API is not available to use with the AIY Projects Voice Kit. In the European Union, Google Cloud Platform services can be used for business purposes only, including the Cloud Speech API. Learn more here: **cloud.google.com/free/docs/frequently-asked-questions**.

>STEP 06

Check Cloud

On your desktop, double-click the Check Cloud icon. Follow along with the script. If everything is working correctly, you'll see this message: 'The cloud connection seems to be working.'

If you see an error message, try restarting your Raspberry Pi with `sudo reboot`. Then follow the instructions above, or take a look at the instructions on the AIY Projects page (**magpi.cc/2q5SSF7**).

> STEP 07

Start it up

Open 'Start dev terminal' and enter:

```
src/cloudspeech_demo.py
```

You can now issue a limited number of commands:

- **Turn on the light (turns on the LED on the Voice HAT).**
- **Turn off the light (turns off the LED).**
- **Blink (the LED blinks).**
- **Goodbye (the program exits).**

Press **CTRL+C** to quit the interaction.

GETTING HELP

With so many options to explore with this first AIY Project from Google, you should make good use of the vibrant Raspberry Pi community. The Raspberry Pi community is on hand to help you with any issues, and make suggestions for your projects. Head to the Raspberry Pi forums and find the new AIY Projects page: **magpi.cc/1NlH5rQ**

>STEP 08

Create a new activation trigger

An activation trigger is a general term describing the condition on which we activate voice recognition or start a conversation with the Google Assistant. Previously you have seen two different types of activation triggers:

- **Voice activation trigger: This is the "Okay, Google" hotword detection in the assistant library demo.**
- **Button trigger: This is when you press the arcade button.**

You may design and implement your own triggers. For example, you may have a motion detection sensor driver that can call a function when motion is detected:

motion.py

```
import aiy.audio
import aiy.cloudspeech
import aiy.voice


def main():
    '''Start voice recognition when motion is detected.'''
    my_motion_detector = MotionDetector()
    recognizer = aiy.cloudspeech.get_recognizer()
    aiy.audio.get_recorder().start()
    while True:
        my_motion_detector.WaitForMotion()
        text = recognizer.recognize()
        aiy.audio.say('You said ', text)

if __name__ == '__main__':
    main()
```

PYTHON API REFERENCE

Module	APIs Provided	Description & Uses in Demo Apps
aiy.voicehat	get_button() get_led() get_status_ui()	For controlling the Arcade button and the LED. See uses in any demo app.
aiy.audio	get_player() get_recorder() record_to_wave() play_wave() play_audio() say()	For controlling the microphone and speaker. It is capable of speaking some text or playing a wave file. See uses in assistant_grpc_demo.py and cloudspeech_demo.py.
aiy.cloudspeech	get_recognizer()	For accessing the Google CloudSpeech APIs. See uses in cloudspeech_demo.py.
aiy.i18n	set_locale_dir() set_language_code() get_language_code()	For customizing the language and locale. Not used directly by demo apps. Some APIs depend on this module. For example, aiy.audio.say() uses this module for speech synthesis.
aiy.assistant.grpc	get_assistant()	For accessing the Google Assistant APIs via gRPC. See uses in assistant_grpc_demo.py.
google.assistant.library		The official Google Assistant Library for Python. See the online documentation at **developers.google.com/assistant/sdk/reference/library/python/**

The MagPi ESSENTIALS

Google Cloud Platform

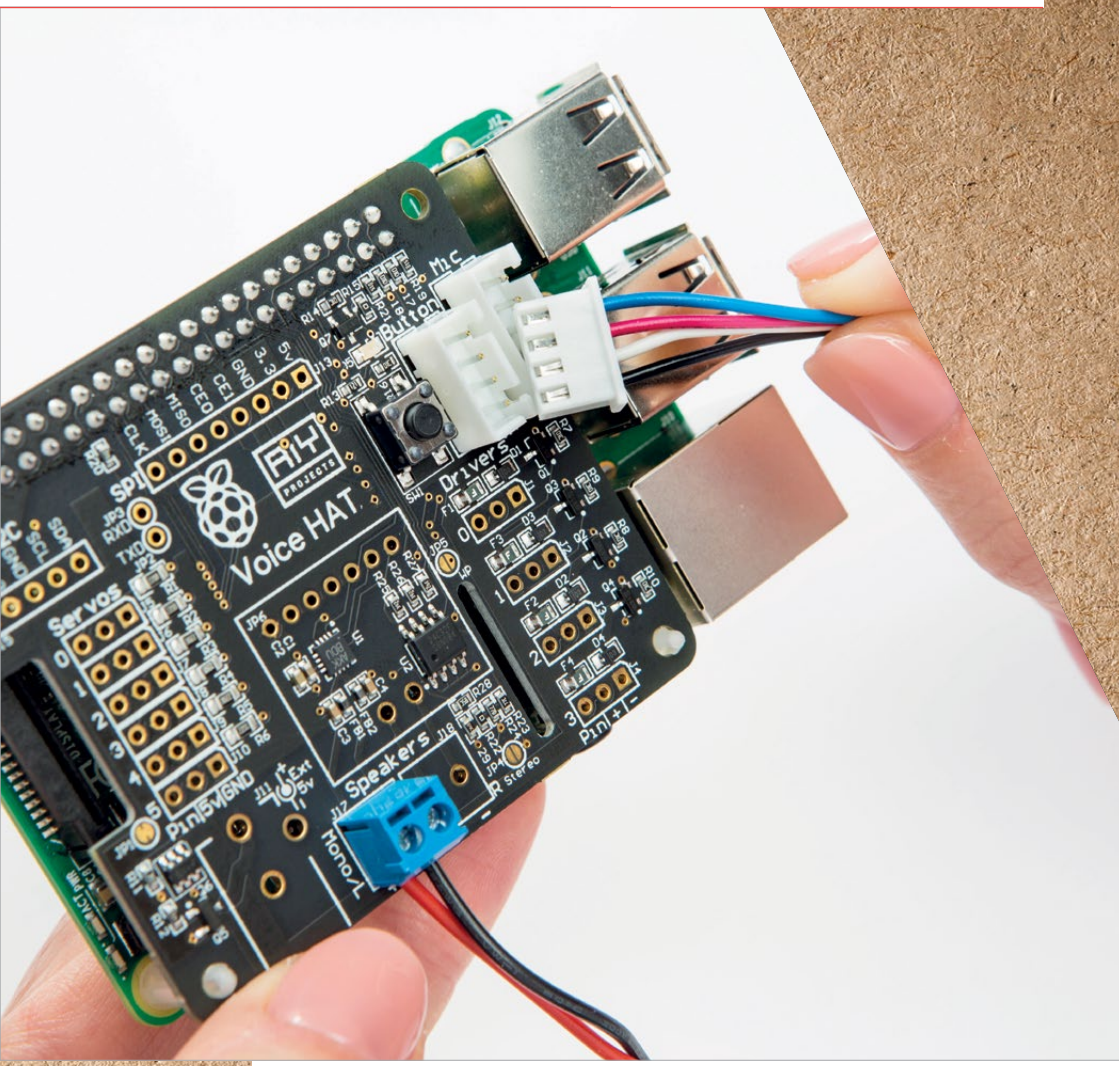
[CHAPTER SEVEN]

CONTROL AN LED

Create custom voice commands for AIY Projects

You'll Need

- AIY Projects voice kit
- Cloud Speech API
- Breadboard
- LED, resistor, and cables



Now that you've switched from the Assistant SDK to the Cloud Speech API, you'll want to know what you can do with it. You add custom commands to your own Python files.

There is a selection of example voice commands located in cloudspeech_demo.py. We're going to modify this file to see how they work, then create our own Python programs to control circuit components attached to the Voice HAT.

>STEP 01
Backup first

You can create new actions and link them to new voice commands by modifying src/cloudspeech_demo.py directly. First, backup the file:

```
cp src/cloudspeech_demo.py src/cloudspeech_demo_backup.py
```

Open the cloud speech demo using

```
nano src/cloudspeech_demo.py
```

>STEP 02
Expect phrase

To add a custom voice command, you first have to make it explicit what command is expected to the recognizer. This improves the recognition rate.

We're going to add a new recognizer.expect_phrase method to the cloudspeech_demo.py code:

```
recognizer.expect_phrase('repeat after me')
```

The program now expects to hear "repeat after me" along with the other commands.

>STEP 03
Handle phrase

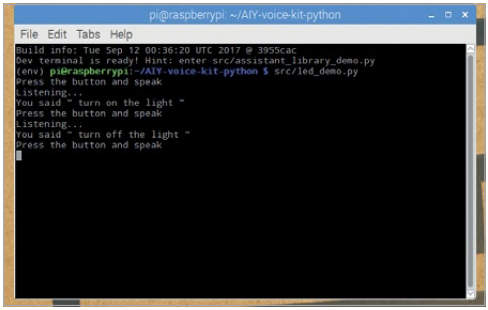
Next we add the code to handle the command. We will use aiy.audio.say to repeat the recognized transcript.

```
elif 'repeat after me' in text:
    to_repeat = text.replace('repeat after me', '', 1)
    aiy.audio.say(to_repeat)
```

You'll find the full modified code in the **cloudspeech_demo.py** code listing. Use **src/cloudspeech_demo.py** to run the modified program. Now press the button and say 'repeat after me 1, 2, 3' the AIY Projects voice kit should say '1, 2, 3'.

cloudspeech_demo.py

```
"""A demo of the Google CloudSpeech recognizer."""

import os

import aiy.audio
import aiy.cloudspeech
import aiy.voicehat


def main():
    recognizer = aiy.cloudspeech.get_recognizer()
    recognizer.expect_phrase('turn off the light')
    recognizer.expect_phrase('turn on the light')
    recognizer.expect_phrase('blink')
    recognizer.expect_phrase('repeat after me')

    button = aiy.voicehat.get_button()
    led = aiy.voicehat.get_led()
    aiy.audio.get_recorder().start()

    while True:
        print('Press the button and speak')
        button.wait_for_press()
        print('Listening...')
        text = recognizer.recognize()
        if text is None:
            print('Sorry, I did not hear you.')
        else:
            print('You said "', text, '"')
            if 'turn on the light' in text:
                led.set_state(aiy.voicehat.LED.ON)
            elif 'turn off the light' in text:
                led.set_state(aiy.voicehat.LED.OFF)
            elif 'blink' in text:
                led.set_state(aiy.voicehat.LED.BLINK)
            elif 'repeat after me' in text:
                to_repeat = text.replace('repeat after me', '', 1)
                aiy.audio.say(to_repeat)
            elif 'goodbye' in text:
                os._exit(0)


if __name__ == '__main__':
    main()
```



Use code to control GPIO pins via your assistant. This short program turns an LED light on or off

>STEP 04
Control an LED

Now that we can add custom commands, we're going to use the AIY Projects kit to control some hardware. Set up an LED circuit using a breadboard – follow the diagram shown on page 58. We are connecting the LED via the pins on Servo 0. Connect the live wire to Pin (on the left). This is GPIO 26 using the BCM numbering system. Connect the ground wire to GND (on the right). The middle pin provides a constant 5V of power. You can see the reference for each pin underneath the Servo 5 rail (check the diagram in 'Voice HAT hardware extensions' at the back of this book).

We have found that it will work by connecting wires directly to the through-holes on the board. For a more reliable circuit, carefully solder the pins supplied with your Voice HAT.

>STEP 05
Enter LED code

Create a new file using **nano src/led_demo.py** and enter the code from the **led_demo.py** listing. Notice the first line: **#!/usr/bin/env python3**. This enables you to run this code from the command line.

>STEP 06
Run the code

We need to make the file executable to run it from the command line.

```
chmod +x /src/led_demo.py
```

Now run the code using:

```
src/led_demo.py
```

Press the button and say "turn on the light". The LED on your breadboard lights up; say "turn off the light" to switch it off.

VOICE RECORDER

If you get a message that says: "Server error: Audio data is being streamed too slowly or too quickly. Please stream audio data approximately at real time." Then you've forgotten to turn on the voice recorder: `aiy.audio.get_recorder().start()`

led_demo.py

```
#!/usr/bin/env python3

import aiy.audio
import aiy.cloudspeech
import aiy.voicehat
import RPi.GPIO as GPIO

def main():
    recognizer = aiy.cloudspeech.get_recognizer()
    recognizer.expect_phrase('turn on the light')
    recognizer.expect_phrase('turn off the light')

    button = aiy.voicehat.get_button()
    aiy.audio.get_recorder().start()

    GPIO.setmode(GPIO.BCM)
    GPIO.setwarnings(False)
    GPIO.setup(26,GPIO.OUT)

    while True:
        print('Press the button and speak')
        button.wait_for_press()
        print('Listening...')
        text = recognizer.recognize()
        if text is None:
            print('Sorry, I did not hear you.')
        else:
            print('You said "', text, '"')
            if 'turn on the light' in text:
                GPIO.output(26,GPIO.HIGH)
            elif 'turn off the light' in text:
                GPIO.output(26,GPIO.LOW)

if __name__ == '__main__':
    main()
```

SET UP AN LED CIRCUIT

CIRCUIT

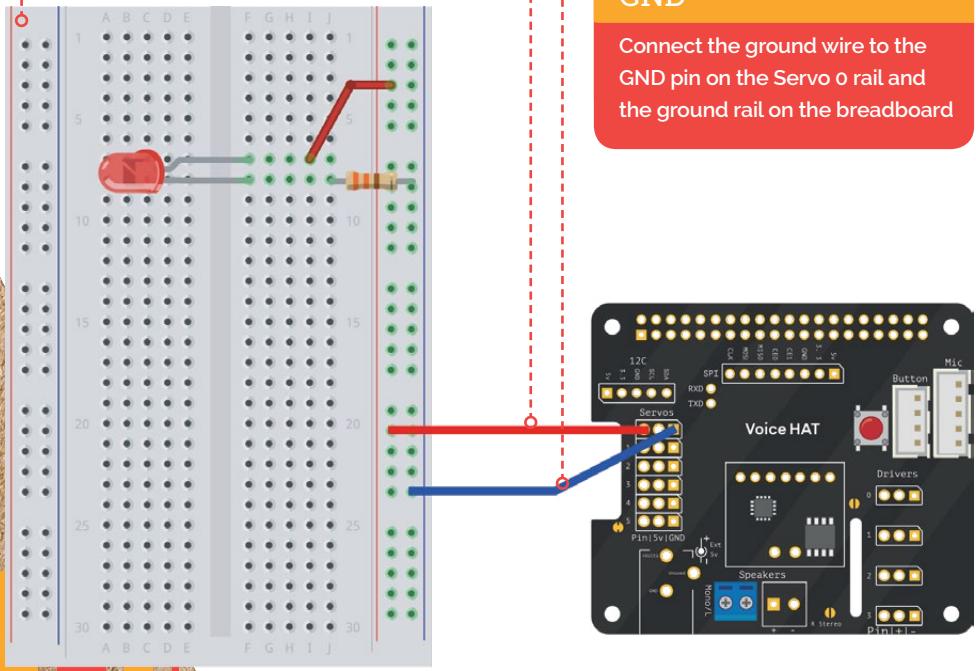
Connect an LED to the breadboard and create a circuit (with the longer leg connected to live and the shorter leg connected to ground). Don't forget to use a resistor (around 330 ohms) to protect the LED

GPIO 26

Connect the live wire to GPIO 26, the leftmost pin on Servo 0, and the live rail on the breadboard. See the GPIO layout guide from the previous page for guidance

GND

Connect the ground wire to the GND pin on the Servo 0 rail and the ground rail on the breadboard



SAFE SHUTDOWN

One script that's well worth knowing is shutdown. This will safely turn off your AIY Projects kit. The **shutdown_demo.py** code uses the subprocess modtule to run a shutdown Unix command. Simply say "shut down" to turn off your AIY Projects kit.

shutdown_demo.py

```
#!/usr/bin/env python3

import aiy.audio
import aiy.cloudspeech
import aiy.voicehat
import subprocess

def main():
    recognizer = aiy.cloudspeech.get_recognizer()
    recognizer.expect_phrase('shutdown')

    button = aiy.voicehat.get_button()
    aiy.audio.get_recorder().start()

    while True:
        print('Press the button and speak')
        button.wait_for_press()
        print('Listening...')
        text = recognizer.recognize()
        if text is None:
            print('Sorry, I did not hear you.')
        else:
            print('You said "', text, '"')
            if 'shutdown' in text:
                subprocess.call(["sudo", "shutdown", "-h", "now"])

if __name__ == '__main__':
    main()
```

The MagPi ESSENTIALS

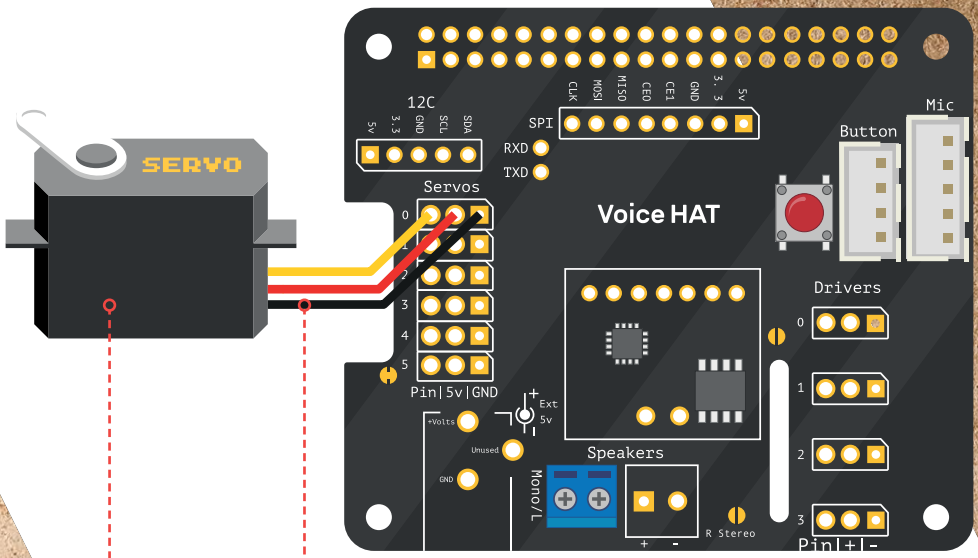
[CHAPTER EIGHT]

ATTACH A SERVO

Servo motors are used to perform fine motor functions, and with AIY Projects you can program them for voice activation

You'll Need

- 9 g micro servo
- AIY Projects Voice HAT



MOTOR CONTROL

Servo motors are controlled using pulses generated by a GPIO pin on the Raspberry Pi. The arm is moved between a high and low position.

SERVO WIRES

The three wires from the servo are connected to row 0 on the Servos rail. Make sure you connect the Pin, 5v, and GND wires in the correct order.

One of the big hopes for AIY Projects is that Raspberry Pi owners will integrate the kit into their own projects. In our last tutorial for AIY Projects, we looked at hooking up the hardware to control an LED light (a typical first hardware project). Here, we're going to take things up a notch and hook up a servo to the AIY Projects board.

On the Voice HAT hardware you will see two columns of pins. The one on the left (marked Servos) is for servos, and has a 25 mA drive limit. The one on the right, marked Drivers, is typically used for motors and has a 500 mA limit. You can connect wires directly to the Voice HAT hardware, but it's easier to prototype your circuits by soldering the pins (supplied with the kit) to the board.

GET YOUR AIY PROJECTS KIT

If you didn't manage to get hold of an AIY Projects kit, don't worry – Google hopes to have more available soon. Sign up for our newsletter, and we'll let you know when more AIY Projects Voice Kits are available: **magpi.cc/Email-me**

>STEP 01

Servo motors

Servo motors move in a circular motion to a set position. They are often used to control robotic arms and legs, grippers, and the position of surfaces (like elevators and rudders on an RC plane).

It is relatively easy to hook up a servo motor to a Raspberry Pi, but the AIY Projects Voice HAT board makes it even easier, with a dedicated column of pins designed to control servo motors. Connecting your servos using the Voice HAT allows them to be controlled using voice commands and the Cloud Speech API.

>STEP 02

Servo control

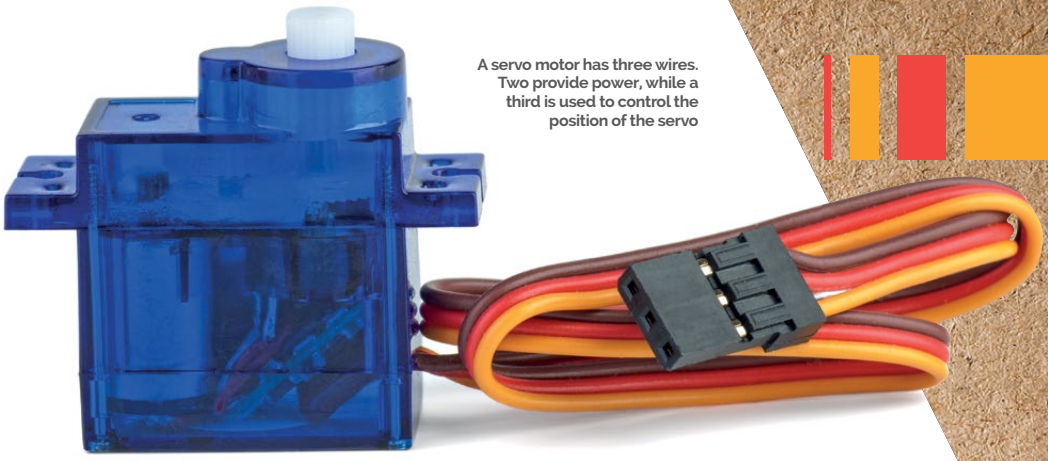
Servo motors are controlled using pulses generated by a GPIO pin on the Raspberry Pi (we're using GPIO 26 on the AIY Projects Voice HAT board). The servo motor expects a pulse (the GPIO pin to be turned on, or high, and then off again) every 20 milliseconds. The length of the pulse determines the position of the servo arm. If it's 1 ms then the servo arm is rotated towards the left; 1.5 ms puts it at the mid-point; 2 ms and it's all the way to the right. The code for detecting these pulses and moving the servo around is provided with the GPIO Zero library. Install using:

```
sudo apt install python3-gpiozero
```

>STEP 03

Connect the servo

We're using a standard 9 g micro servo in this tutorial. These are tiny 5 V servos with an operating voltage of 4.8 V. Each servo comes with three wires: usually these are red, brown, and orange. Red and brown provide power to the servo, and are live and neutral respectively, while the third wire detects the pulse. Make sure your Raspberry Pi is powered down, and connect the servo wires directly to the Servos 0



A servo motor has three wires. Two provide power, while a third is used to control the position of the servo

row on your AIY Projects Voice HAT. Many servos have all three wires bundled into a JR connector; this can be plugged directly into the Voice HAT board. Be sure to fit it the correct way around, with the orange/yellow cable in the GPIO pin on the left.

>STEP 04

Power the servo

The 5 V running through the GPIO pins on the Raspberry Pi is enough to power two to three very small servo motors. We're only using one here in our tutorial, so we aren't going to add additional power.

>STEP 05

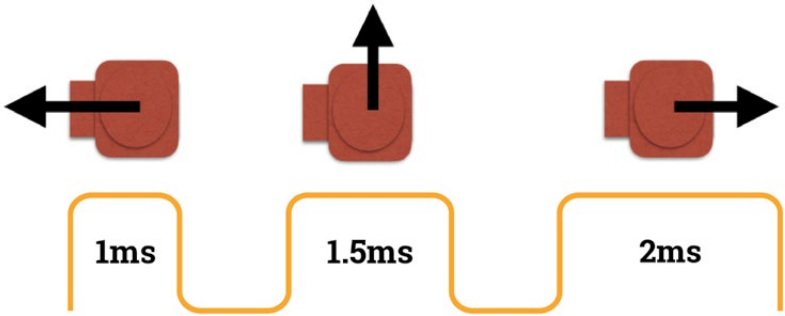
Test it out

Before coding the servo to respond to your voice commands, you can test the circuit using GPIO Zero in Python. Open IDLE 3 and enter the code from **servo_test.py**. Save the code and press **F5** to run it. The servo will move from its minimum position to the mid-point, and then to the maximum position with a pause between each step. Press **CTRL+C** to quit the program and stop the movement. If the servo doesn't work, double-check your connections.

>STEP 06

Integrate with voice

Now that the servo is working, it's time to integrate it with the AIY Projects code using the **servo_demo.py** script. Open Start dev



Right: A servo motor measures the length between pulses (a GPIO pin being turned rapidly on and off). When the pin is on for 1 ms, the servo moves to the low position. When it's on for 2 ms, it moves to the high position. Other pulse lengths are used to set it between low and high

terminal and use **nano src/servo_demo.py** to create the empty text file. Don't forget to use **chmod +x src/servo_demo.py** afterwards to make it executable.

Run **src/servo_demo.py** and press the button on your AIY Projects Voice HAT board. Now say "change to minimum" or "change to maximum" to move the arm up and down. Saying "change" and any other command moves the arm back to the middle.

Now everything is working, you can attach the servo to the side of the kit's cardboard box. On the side of the kit, you'll see an arc-shaped hole. Most small servos will fit in this space. Twist the servo to lock it in place. Try adding a wooden or 3D-printed arm to the servo so you can clearly see it moving.

servo_test.py

```
from gpiozero import Servo
from time import sleep

servo = Servo(26)
while True:
    servo.min()
    sleep(1)
    servo.mid()
    sleep(1)
    servo.max()
    sleep(1)
```

server_demo.py

```
#!/usr/bin/env python3

import aiy.audio
import aiy.cloudspeech
import aiy.voicehat
from gpiozero import Servo

def main():
    recognizer = aiy.cloudspeech.get_recognizer()
    recognizer.expect_phrase('maximum')
    recognizer.expect_phrase('minimum')
    recognizer.expect_phrase('middle')

    button = aiy.voicehat.get_button()
    aiy.audio.get_recorder().start()

    servo = Servo(26)

    while True:
        print('Press the button and speak')
        button.wait_for_press()
        print('Listening...')
        text = recognizer.recognize()
        if text is None:
            print('Sorry, I did not hear you.')
        else:
            print('You said "', text, '"')
            if 'maximum' in text:
                print('Moving servo to maximum')
                servo.max()
            elif 'minimum' in text:
                print('Moving servo to minimum')
                servo.min()
            elif 'middle' in text:
                print('Moving servo to middle')
                servo.mid()

if __name__ == '__main__':
    main()
```

The MagPi ESSENTIALS

[CHAPTER NINE]

CONTROL A DC MOTOR

Connect a motor to your AIY Projects Voice HAT board

You'll Need

- DC motor
- 4×AA battery pack
- Breadboard and jumper wires
- Utility / Stanley knife

DC motors are used to control wheels, arm joints, and moving components. They are often found in robotic projects

In our previous AIY Projects tutorials, we've looked at how to move beyond using the Voice Assistant, and towards using your Voice HAT with basic electronics.

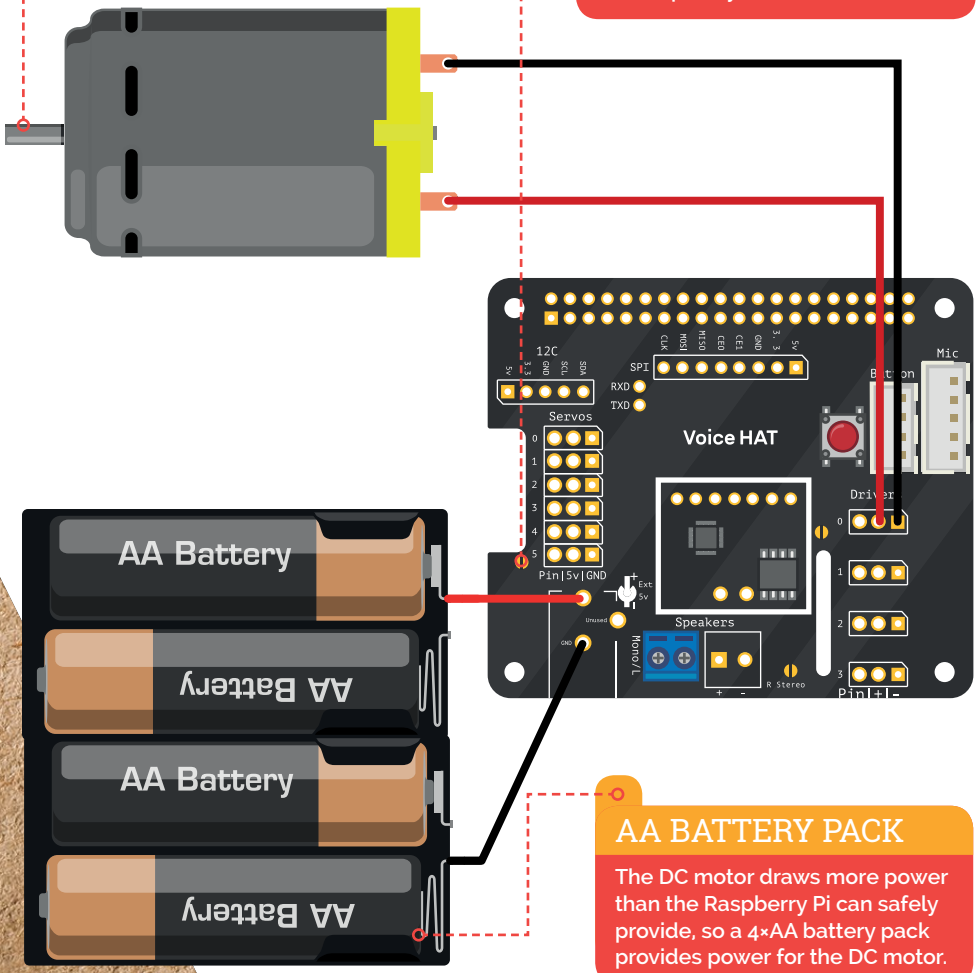
If you've been following our tutorials, you will have discovered how to connect the Voice HAT hardware to simple circuits. So far we've looked at how to control LED lights and servo motors, but in this tutorial we'll look at something a little more complex: using the AIY Projects Voice HAT to control a motor.

DC MOTOR

The DC motor connects to the positive and negative voltages via two wires. The direction in which the motor moves depends on which way around the two wires are placed.

EXTERNAL POWER SOLDER JUMPER

You need to cut the external power solder jumper, located just to the left of Servos 5 on the board. This isolates the Voice HAT power from the Raspberry Pi.



AA BATTERY PACK

The DC motor draws more power than the Raspberry Pi can safely provide, so a 4×AA battery pack provides power for the DC motor.

>STEP 01
Cut the power
The first thing you need to do is isolate the Raspberry Pi's power supply from the power on the Voice HAT board. This will prevent the DC motor from draining too much power and shorting out your Raspberry Pi. Locate the external power solder jumper marked JP1 (just to the left of Servos 5 on the Voice HAT board). Use a utility knife to cut the connection in the jumper (you can always re-solder this joint if you wish to share the power between the board and the motor again).

>STEP 02
Power off
Make sure your Raspberry Pi and Voice HAT board are powered off. Now connect the positive leg of the DC motor to the middle pin on Drivers 0. Notice that at the bottom of the Driver pins is a '+' symbol.

>STEP 03
Wire for power
Next, connect the negative wire of the motor to the '–' pin on Drivers 0 (the pin on the right). You may have noticed that we're not connected to

motor_test.py

```
from gpiozero import PWMOutputDevice
from time import sleep

pwm = PWMOutputDevice(4)
while True:
  pwm.on()
  sleep(1)
  pwm.off()
  sleep(1)
  pwm.value = 0.5
  sleep(1)
  pwm.value = 0.0
  sleep(1)
```

the GPIO Pin on the left (which is GPIO4); this doesn't matter as it also controls the negative '–' pin that we have just connected to. This allows us to turn the motor on and off.

>STEP 04

Power up

Finally, connect the 4×AA battery pack to the +Volts and GND pins at the lower left-hand corner of the Voice HAT. This pack will ensure that the motor has enough power when you are using the Voice HAT, which will prevent your Raspberry Pi from crashing. Connect the power and turn on the battery pack.

>STEP 05

Turn on the Pi

Now turn on the Raspberry Pi and boot into the AIY Projects software. Enter the code from **motor_test.py** to test the circuit. We are using `PWMOutputDevice` from GPIO Zero to control the motor. This enables us to manage the speed of the motor (**magpi.cc/2tnAGrz**). We can use the `.on()` and `.off()` methods to start and stop our motor. Alternatively, we can set the value instance variable to a value between 0.0 and 1.0 to control the speed. These techniques are shown in the **motor_demo.py** code. You can also use `pwm.pulse()` to pulse the motor on and off.

>STEP 06

Use voice control

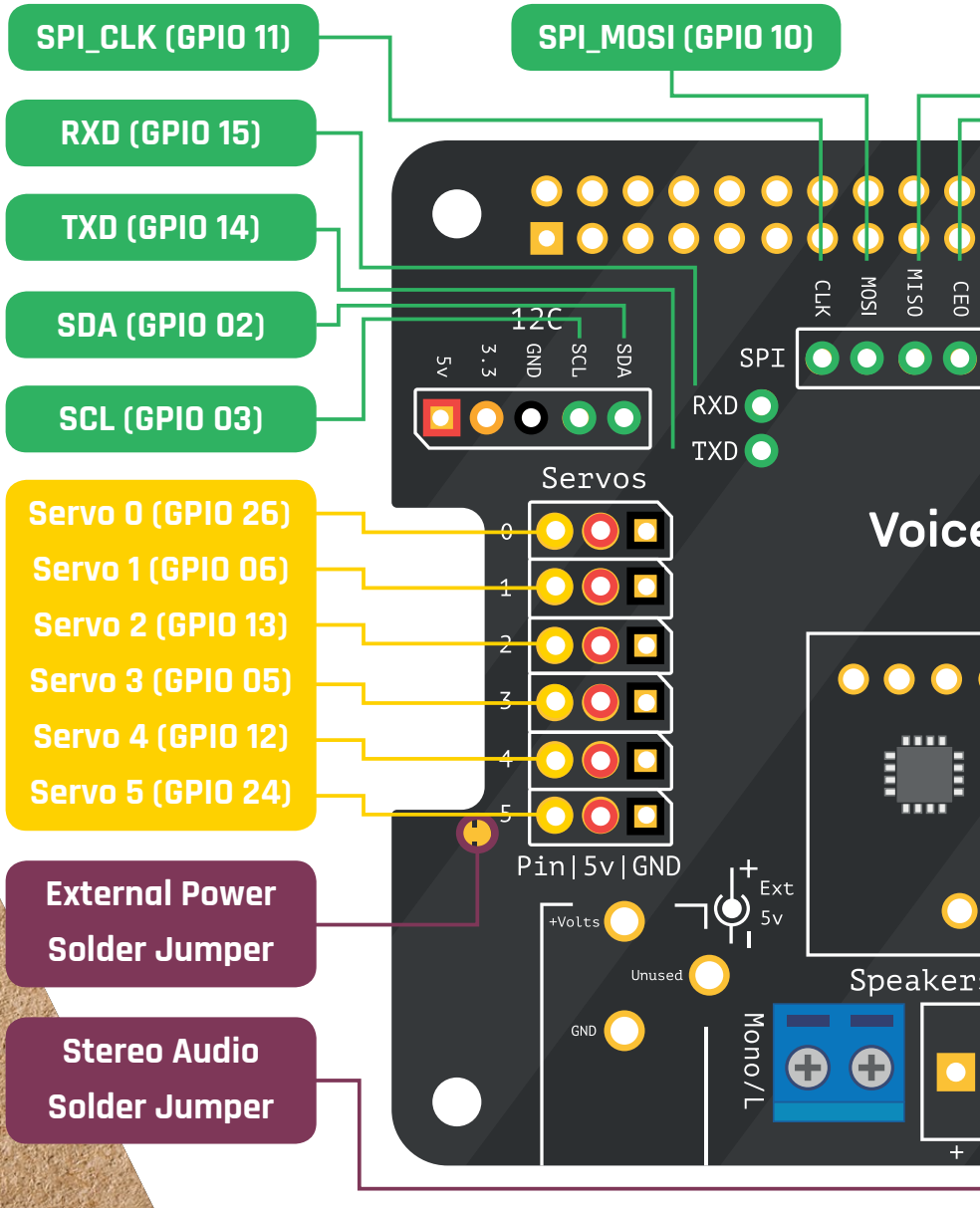
Now that we've seen how to control the motor using GPIO Zero, it is time to integrate it with the Cloud Speech API. Push the button on your Voice HAT board and say "motor on" to start the motor running; push the button again and say "motor off" to stop it

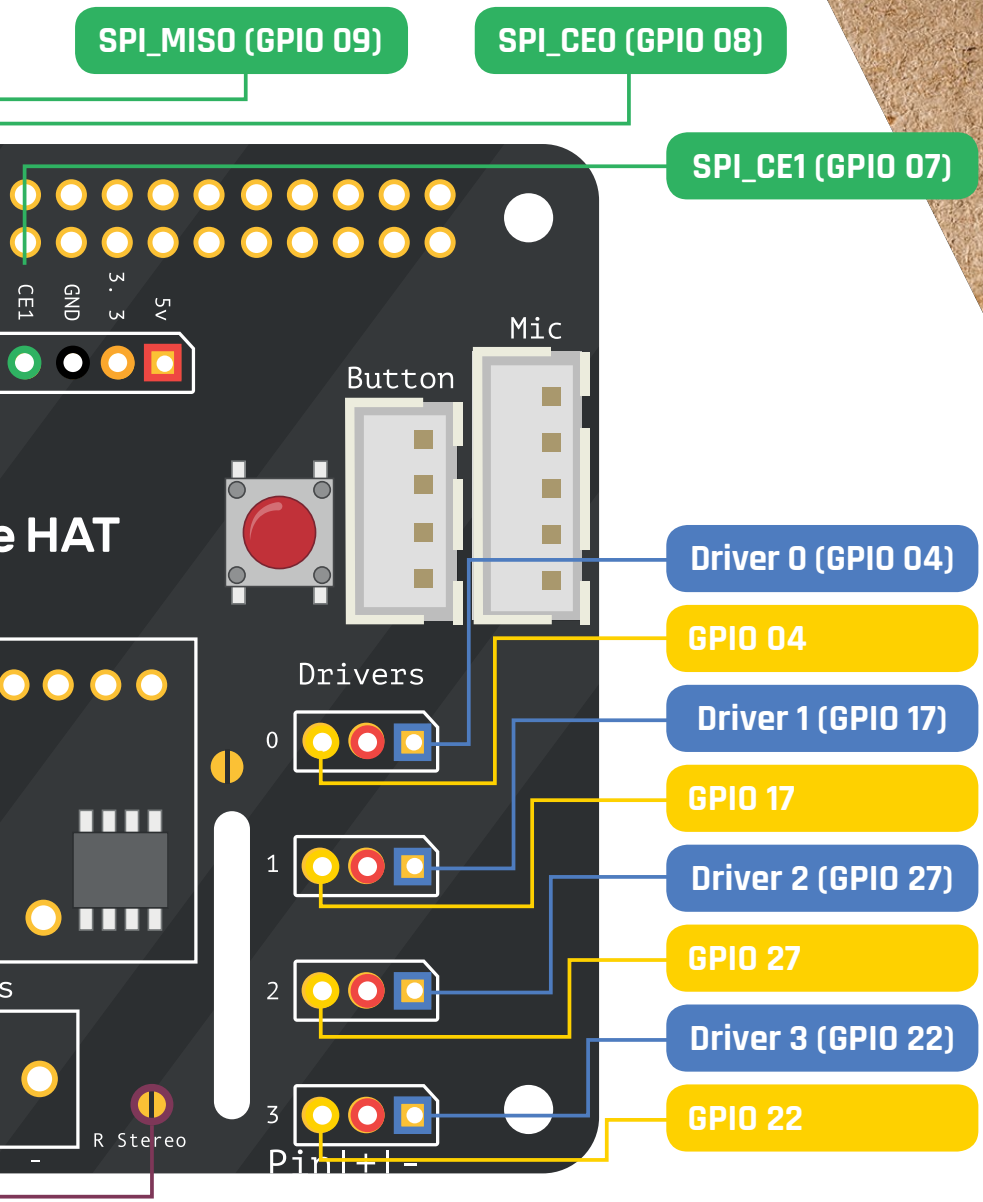
You can add more motors to your AIY Projects kit using the four rows of Drivers on the Voice HAT board. These can be used to build robots and other motion projects. Discover more project ideas for your Voice Kit at the AIY Projects forum (**magpi.cc/2wuTMMW**). We hope you've enjoyed this guide and build many great things with your AIY Projects Voice Kit.

motor_demo.py

```
#!/usr/bin/env python3

import aiy.audio
import aiy.cloudspeech
import aiy.voicehat
from gpiozero import PWMOutputDevice

def main():
    recognizer = aiy.cloudspeech.get_recognizer()
    recognizer.expect_phrase('on')
    recognizer.expect_phrase('off')

    button = aiy.voicehat.get_button()
    aiy.audio.get_recorder().start()

    pwm = PWMOutputDevice(4)

    while True:
        print('Press the button and speak')
        button.wait_for_press()
        print('Listening...')
        text = recognizer.recognize()
        if text is None:
            print('Sorry, I did not hear you.')
        else:
            print('You said "', text, '"')
            if 'on' in text:
                print('Turning motor on')
                pwm.on()
            elif 'off' in text:
                print('Turning motor off')
                pwm.off()

if __name__ == '__main__':
    main()
```

VOICE HAT HARDWARE EXTENSIONS




SPI_MISO (GPIO 09)
SPI_CE0 (GPIO 08)
SPI_CE1 (GPIO 07)
CE1
GND
3. 3
5v
Mic
Button
e HAT
Drivers
0
1
2
3
R Stereo
Driver 0 (GPIO 04)
GPIO 04
Driver 1 (GPIO 17)
GPIO 17
Driver 2 (GPIO 27)
GPIO 27
Driver 3 (GPIO 22)
GPIO 22

PLUS

AN OFFICIAL **PI ZERO CASE** WITH 3 COVERS

AND FREE CAMERA MODULE CONNECTOR AND USB / HDMI CONVERTER CABLES

Pricing:

Get six issues:

£30 (UK)

£45 (EU)

$69 (USA)

£50 (Rest of World)

Subscribe for a year:

£55 (UK)

£80 (EU)

$129 (USA)

£90 (Rest of World)

FREE PI ZERO W!

Get three issues:

£12.99 (UK) **(Direct Debit)** | **$37.50** (US) **(quarterly)**

How to subscribe:

- **magpi.cc/Subs-2** (UK / ROW)
- **imsnews.com/magpi** (USA)
- Call +44(0)1202 586848 (UK/ROW)
- Call 800 428 3003 (USA)

Search 'The MagPi' on your app store:

GET IT ON Google Play

Available on the App Store

The MagPi ESSENTIALS

raspberrypi.org/**magpi**