

# Build your own Raspberry Pi Google Assistant



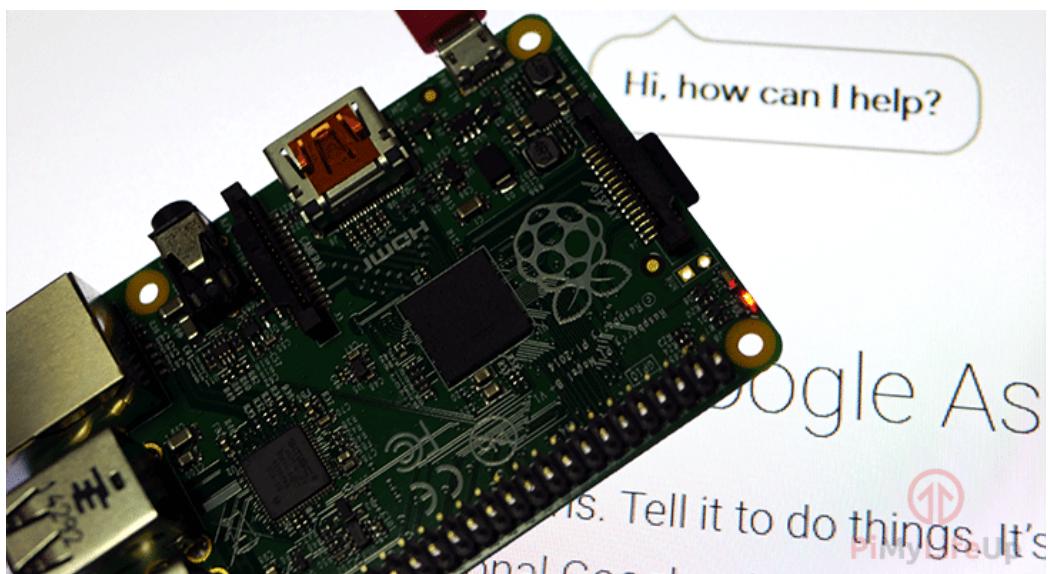
by Gus

Updated Jun 25, 2021

Published Mar 25, 2018

[Beginner, IoT](#)

This Raspberry Pi Google assistant project will walk you through on how to build and set up your very own Pi-powered Google Assistant.



This assistant will actively listen to your voice and respond to your queries, all you need to say is "Ok Google" or "Hey Google" followed by your query.

To set up your own Google Assistant, we will be walking you through how to test your Google Assistant API and also show you how to download and set up the actual Goog

[How to setup Raspberry Pi Retr...](#)

By the end of this tutorial, you should have a very capable virtual assistant operating o

---

To set up your own Google Assistant on your Raspberry Pi all you will need is a USB microphone and a set of speakers.

Of course, you will also need all the default equipment that you need for [getting your Raspberry Pi started](#).

---

## ≡ Equipment List

---

The entire list of all the pieces of equipment that we used for this Raspberry Pi Google assistant tutorial is listed below.

### Recommended

- ⌚ [Raspberry Pi](#)
- MMC [Micro SD Card](#)
- ⚡ [Power Supply](#)
- \_MICROPHONE\_ [USB Microphone](#)
- 🔊 [Speakers](#)
- ↔ [Ethernet Network Connection](#) or [Wifi dongle](#) (The Pi 3 has WiFi inbuilt)

[How to setup Raspberry Pi Retr...](#) X

### Optional



---

## G Registering for the Google API

---

1. Before we get started with setting up the Google Assistant code on the Raspberry Pi itself, we must first register and set up a project on the Google Actions Console.

With your Google account ready to go to the [Google Console Actions dashboard](#), or just go to the URL below which will take you there.

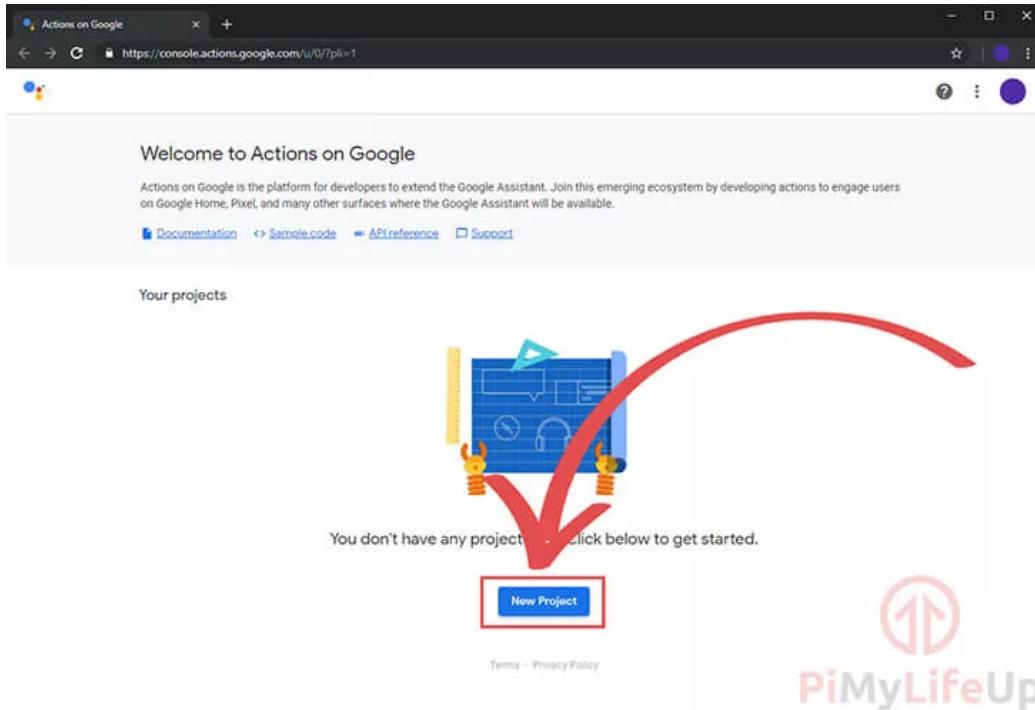
<https://console.actions.google.com>

2. Once you have logged into your account, you will be greeted with the following screen.

On here you will want to click the “**Add/Import project**” button as shown in our screenshot below.

How to setup Raspberry Pi Retr... X

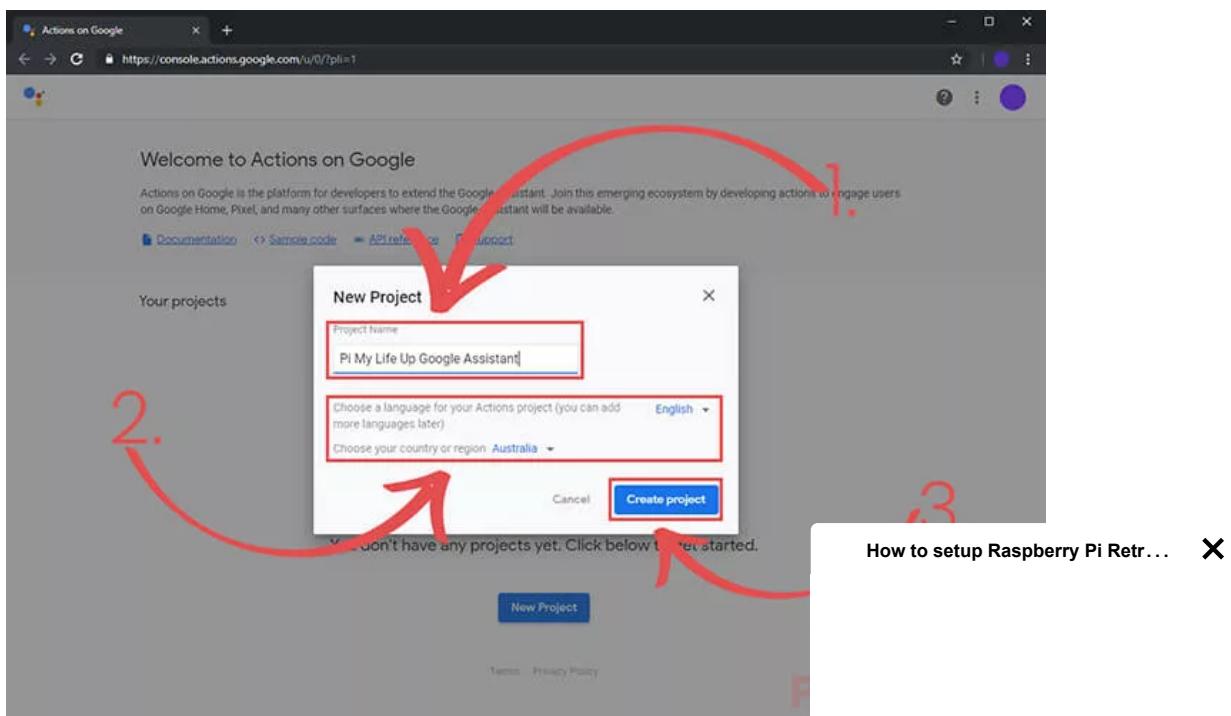




### 3. On this next screen, you will be asked to enter a “Project Name” (1.)

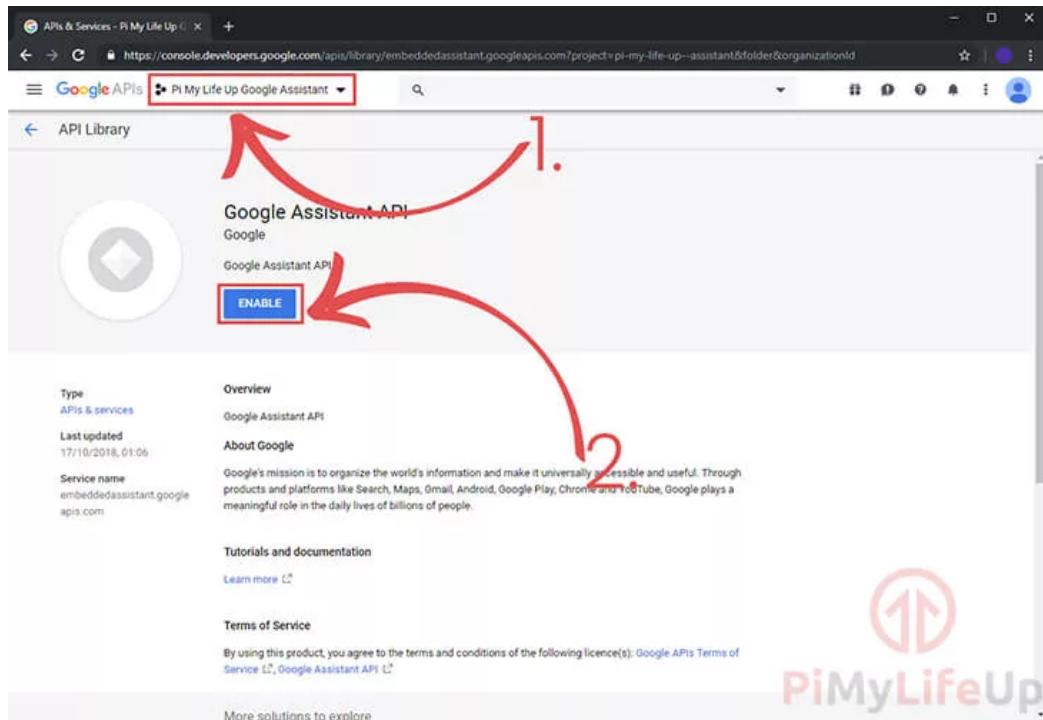
In addition to a project name you need to set both your country and your language as shown in the screenshot (2.)

Once you have set the Project Name and chosen your language and country, click the “Create Project” (3.) button.



### 4. In a new tab, go to the [Google developers console](#) and enable the Google Embedder API

Once you are sure you have your current project selected, click the “**Enable**” button (2.)

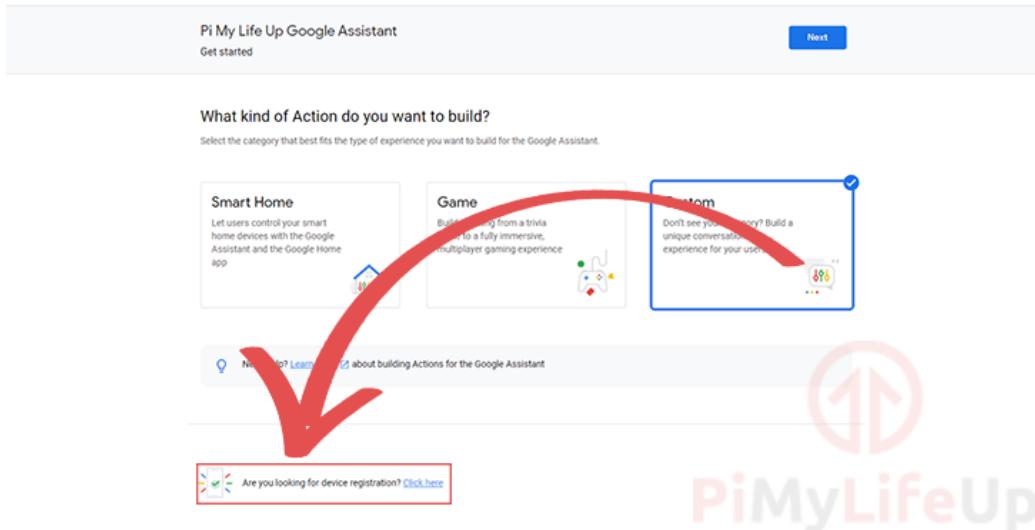


**5.** Now back in the other tab where you created the project, scroll down to the bottom of the screen.

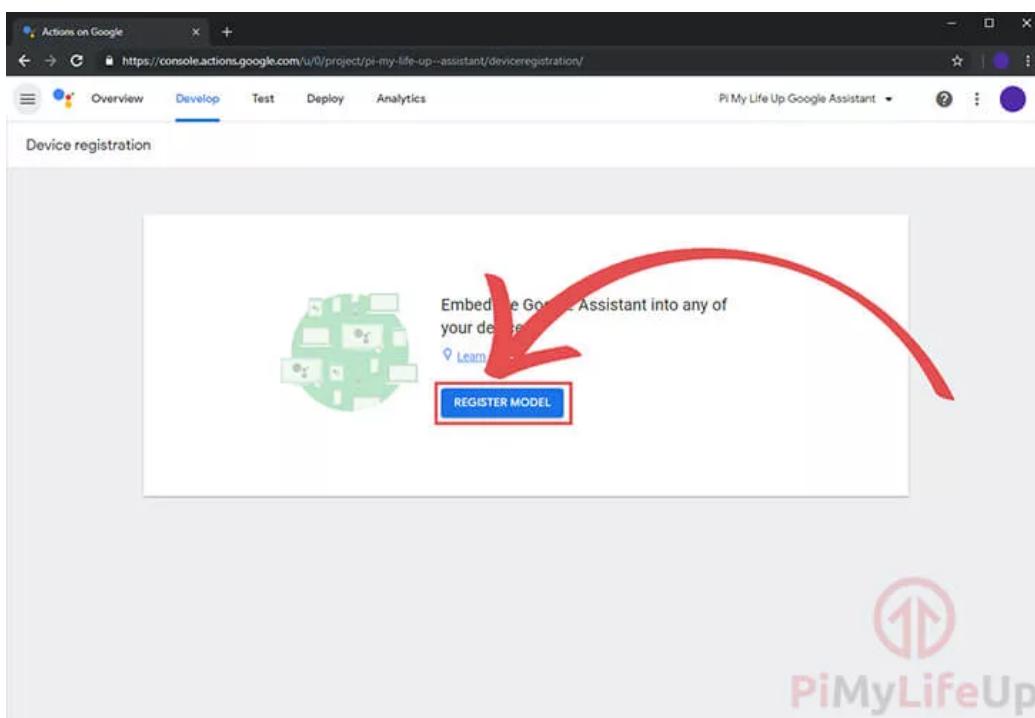
You should see some text saying, “**Are you looking for device registration? Click here**”.

All you need to do to proceed is to click the “**click here**” text.

How to setup Raspberry Pi Retr... X



6. You will now be taken to the following screen, click the "**Register Model**" button to continue.



7. On this screen, you need to set a "**Product Name**", "**Manufacturer name**" and set

How to setup Raspberry Pi Retr... X

Below you can see the data that we entered into it, it doesn't hugely matter what you : need to be set for you to be able to register your model.

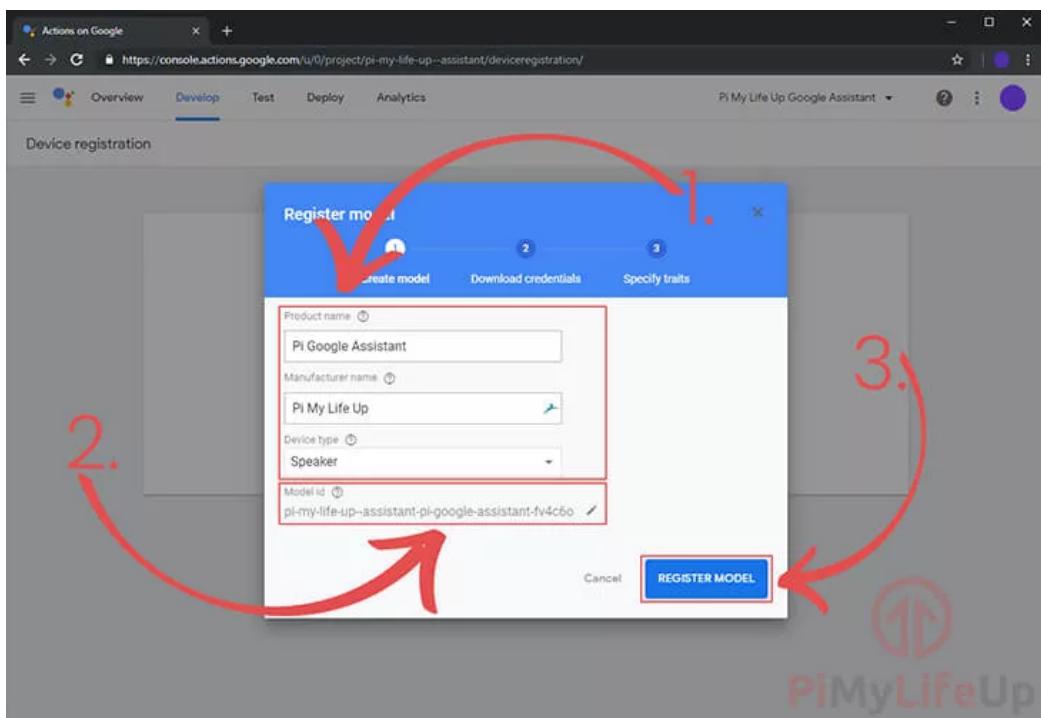
For the "**Product Name**" we just set this as a simple descriptor of what we are using th simply a "**pi3 Google Assistant**".

"**Manufacturer name**" doesn't hugely matter as we have no intention of this being a

API for on our Raspberry Pi.

Make sure you write down then “**Device Model ID**” (2.) as you will need this later in the tutorial.

Finally, once everything is set, and you have written down the “**Device Model ID**” click the “**Register Model**” (3.) button to continue.



8. Now that you have registered the model you will now be taken to the “**Download credentials**” screen.

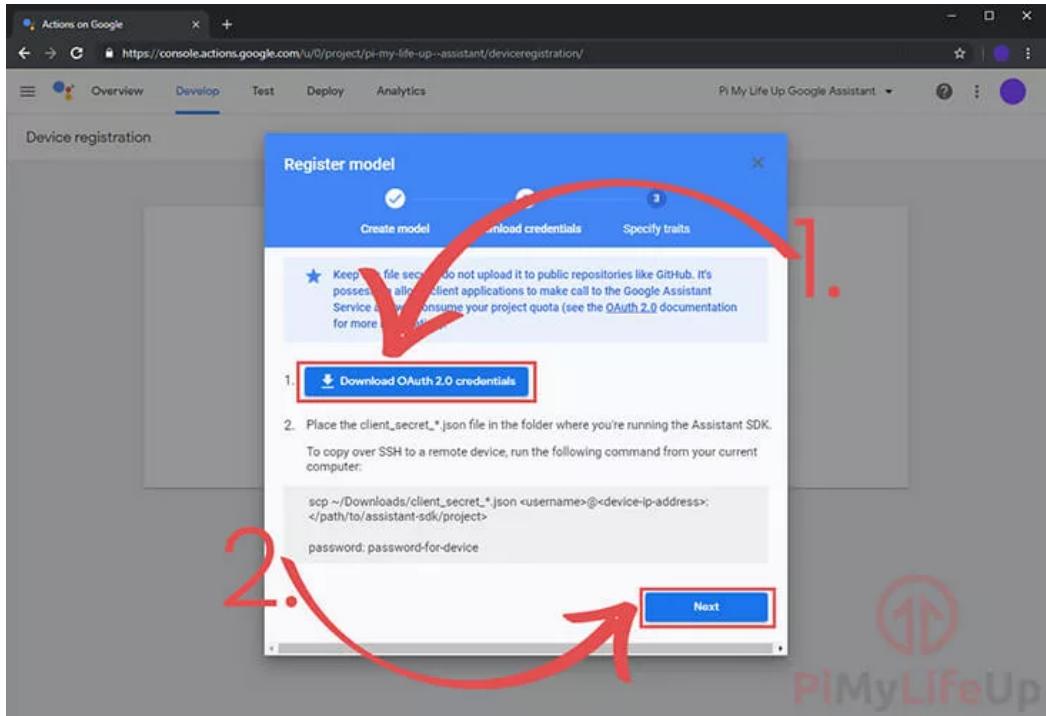
This screen is crucial as the provided credentials file is what we need for our Raspberry Pi 3 based Google Assistant to talk with the server.

[How to setup Raspberry Pi Retr...](#) X

To get this credentials file click the “**Download OAuth 2.0 credentials**” (1.) button

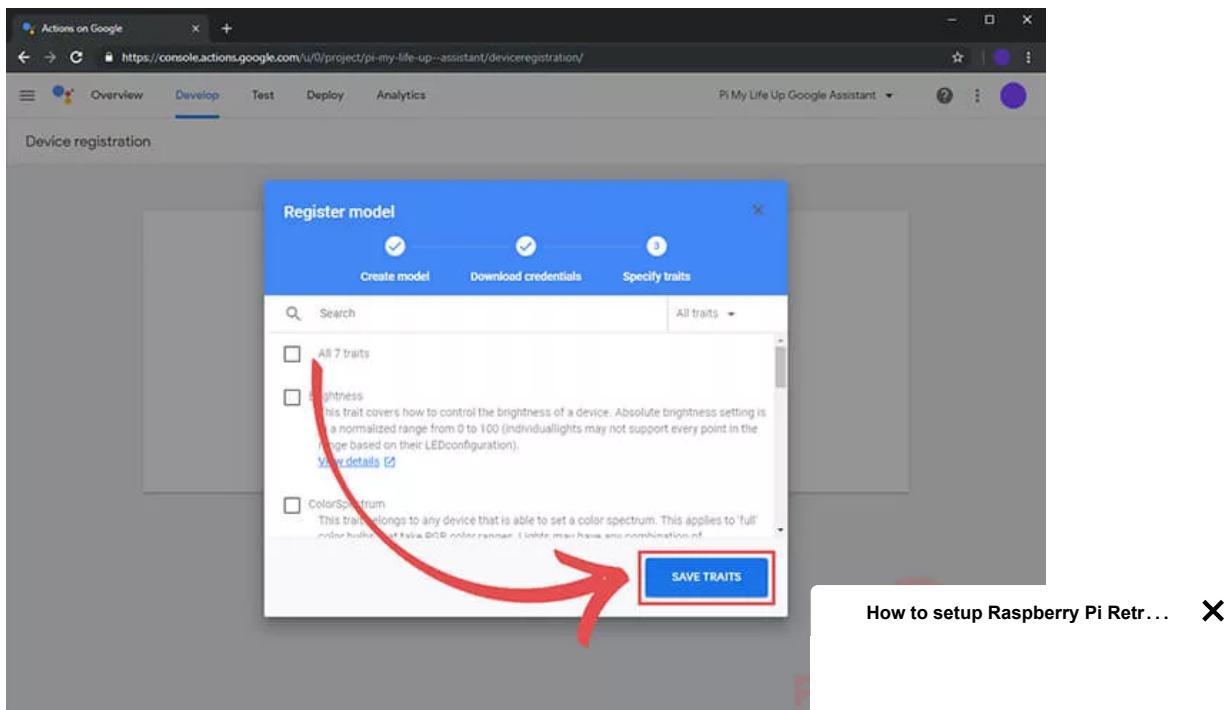
Keep this somewhere safe, as we will the text inside the file to the Raspberry Pi. (Of course directly to your Pi)

Once you have the credentials safely stored on your on your computer or Raspberry Pi (2.) button.



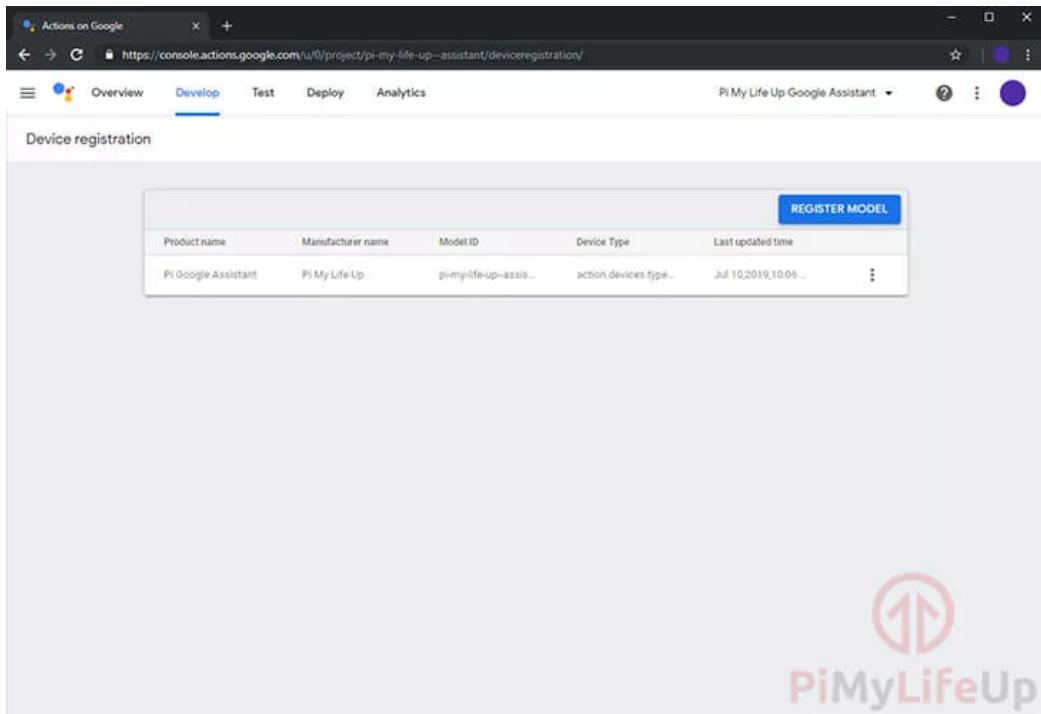
## 9. Finally, you can specify any traits that you might need.

In our case we don't need any of these so we just clicked the "**Save Traits**" button as shown below.



## 10. Once everything is done, you should be shown on this screen.

We now only have one last thing we need to do before we can set up the Google Assistant on the Raspberry Pi itself.



The screenshot shows a browser window for the Actions on Google developer console at the URL <https://console.actions.google.com/u/0/project/pi-my-life-up--assistant/deviceregistration/>. The tab bar includes Overview, Develop (which is selected), Test, Deploy, and Analytics. The title bar says "Actions on Google". The main content area is titled "Device registration" and contains a table with one row. The table has columns for Product name, Manufacturer name, Model ID, Device Type, and Last updated time. The data in the table is:

Product name	Manufacturer name	Model ID	Device Type	Last updated time
Pi Google Assistant	Pi My Life Up	pi-my-life-up-assis...	action.devices.type...	Jul 10, 2019, 10:06...

A blue "REGISTER MODEL" button is located at the top right of the table. Below the table, there is a large red circular logo with a white "Pi" symbol, followed by the text "PiMyLifeUp" in red and grey.

**11.** We also need to configure the OAuth consent screen. Without this, Google won't let us authorize our Raspberry Pi Google Assistant device later in this tutorial.

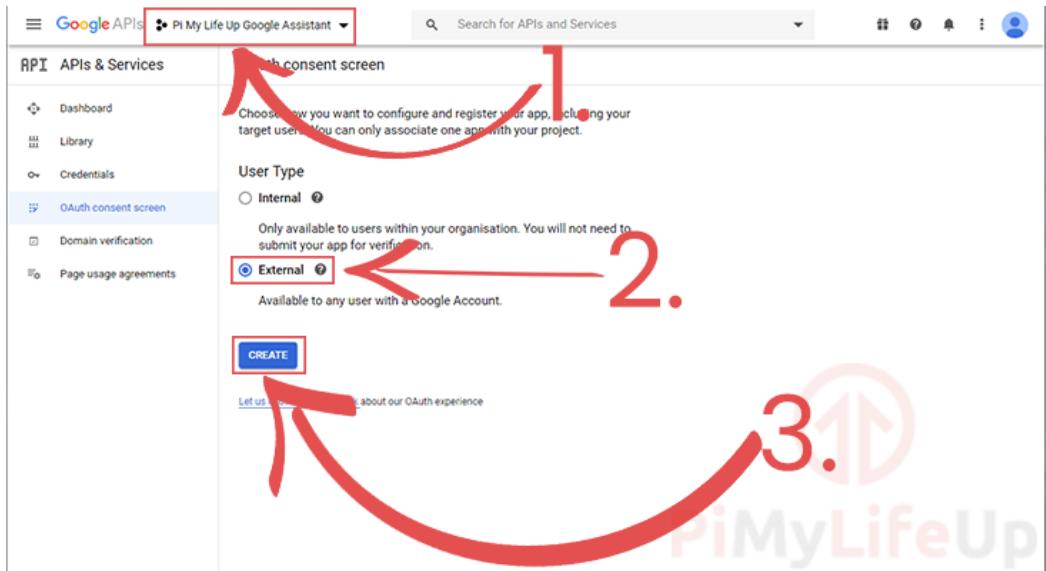
To do this, you will need to go to the [API Credentials OAuth consent screen](#) settings page.

On this first page, make sure that your project is selected in the top dropdown box (1.)

Secondly, you will need to select “**External**” as the user type (2.). This option will allow anyone with a Google account to utilize the Google Assistant software that we are setting up.

[How to setup Raspberry Pi Retr...](#) 

Once all configured, click the “**CREATE**” button to continue on to setting up the aut



**12.** This page will allow you to set up the details of the auth screen. As we don't intend this to be publicly accessible there is only one option we are required to set.

Click the email dropdown to select your accounts email address (1.), selecting this will allow us to continue with the guide.

Once everything required has been set, click the "Save" button (2.) at the bottom of the page.

How to setup Raspberry Pi Retr... X

**Support email** ⓘ  
Shown on the consent screen for user support

**Scopes for Google APIs**  
Scopes allow your application to access user data. Learn more  
If you add a sensitive scope, such as `https://www.googleapis.com/auth/drive.readonly`, it will give you full access to Gmail or Drive.  
Google will verify your consent screen before it's published.

**Scopes**  
email  
profile  
openid

**Add scope**

**Authorized domains** ⓘ  
To protect you and your users, Google only allows applications that authenticate using OAuth to use Authorized Domains. Your application's links must be hosted on Authorized Domains. Learn more

**pi-my-life-up-assistant.firebaseioapp.com**

**web app**

**example.com**  
Type in the domain and press Enter to add it

**Application Homepage link**  
Shown on the consent screen. Must be hosted on an Authorized Domain.  
 https:// or http://

**Application Privacy Policy link**  
Shown on the consent screen. Must be hosted on an Authorized Domain.  
 https:// or http://

**View Terms of Service link (optional)**  
Shown on the consent screen. Must be hosted on an Authorized Domain.  
 or http://

**Save** **Submit for verification** **Cancel**

**13.** Finally, we need to go to the [Google My Account activity controls](#).

On here you will need to activate the following activity controls to ensure that the Goc

How to setup Raspberry Pi Retr...

- Device Information
  - Voice & Audio Activity
- 

## 🔊 Setting up your Audio for Google Assistant

---

### 1. Now that we have set up an account on the Google Actions Console we must configure the audio for it.

The Google Assistant SDK that we will be using has some strict requirements for it to work correctly.

To get started with setting up the audio on the Raspberry Pi we must first obtain the card and device numbers for our various inputs and outputs.

Our steps below will show you have to get the correct numbers for these devices.

#### 1a. Locate your USB microphone by utilizing the following command.

Write down both the card number and the device number for it.

```
Terminal $
```

```
arecord -l
```

#### 1b. Now to locate your speaker, we will be utilizing the following command.

Again write down the card number and device number.

Note that the Raspberry Pi's **3.5mm-jack** is typically labeled as **Analog** or **bcm2835 ALSA**, with the **HDMI output** being identified as **bcm2835 IEC958/HDMI**.

How to setup Raspberry Pi Retr... X



```
Terminal $
```

```
aplay -l
```

- 2.** Now that we have grabbed our device and card numbers for both the microphone and our audio output we need to create a file named `.asoundrc` in the pi users home directory.

The `.asoundrc` file helps by defining which audio devices that the audio driver should be utilizing.

We can create this file by running the following command within the terminal.

```
Terminal $
```

```
nano /home/pi/.asoundrc
```

- 3.** Within this file enter the following lines.

Make sure that you replace `<card number>` and `<device number>` their respective values that you retrieved during **Step 1**.

```
Add >
```

```
pcm.!default {
    type asym
    capture.pcm "mic"
    playback.pcm "speaker"
}
pcm.mic {
    type plug
    slave {
        pcm "hw:<card number>,<device number>"
        rate 48000
    }
}
pcm.speaker {
    type plug
    slave {
        pcm "hw:<card number>,<device number>"
    }
}
```

How to setup Raspberry Pi Retr...



- 4.** With the lines in, and changes made. Save by pressing **CTRL + X** then **Y** and finally



## 🔊 Testing your Speakers and Microphone

1. Before we get into all the hard work of setting up our Raspberry Pi Google Assistant and setting up the required API, we will first test to ensure our audio is working.

At this stage, you must have your USB microphone and speakers attached to your Raspberry Pi.

Once you are sure both are connected to the Raspberry Pi, we can test to make sure that the speakers are working correctly by running the following command.

Terminal \$

```
speaker-test -t wav
```

You should hear sound from your speakers. This sound will be a person speaking.

If you do not hear anything coming from your speaker's double check they are plugged in correctly and are turned up.

2. Now, let's test our microphone by making a recording, to do this we will run the following command on your Raspberry Pi.

This command will make a short 5-second recording.

Terminal \$

[How to setup Raspberry Pi Retr...](#)



```
arecord --format=S16_LE --duration=5 --rate=16000 --file-type=raw
```

If you receive an error when running this command make sure that you have your microphone connected correctly. This command will only succeed if it can successfully listen to your microphone.



Doing this will allow you to test the playback volume and also listen to the recording volume.

Doing this is a crucial task as you don't want your Raspberry Pi picking up every little noise but you also don't want it being able to barely hear you when you say "**Ok Google**".

Terminal \$

```
aplay --format=S16_LE --rate=16000 out.raw
```

**4.** If you find the playback volume or recording volume is either too high or too low, then you can run the following command to launch the mixer.

This command will allow you to tweak the output volumes for your various output devices.

From our tests, we recommend you use a level of at least **70**, utilize the command in **Step 1** of this section to check the volume levels.

Terminal \$

```
alsamixer
```

Once you have confirmed that your microphone and speakers are working correctly, you can move onto setting up your very own Raspberry Pi Google Assistant!



## Downloading and setting up Googl

[How to setup Raspberry Pi Retr...](#)



**1.** With the Google Assistant API now configured and set up there are a few things we

Let's first update the Raspberry Pi's package list by running the following command.



```
Terminal $
```

```
sudo apt update
```

- 2.** Once the Raspberry Pi has finished updating, we can then proceed with setting up everything we need for running the Google Assistant API.

On your Raspberry Pi, we will be creating a file where we will store the credentials we downloaded earlier on our computer.

To do this run the following two commands to create a folder and begin writing our credential file.

```
Terminal $
```

```
mkdir ~/googleassistant  
nano ~/googleassistant/credentials.json
```

- 3.** Within this file, we need to copy the contents of the credentials file that we downloaded to your computer.

You can open the `.json` file in any text editor and press `CTRL + A` then `CTRL + C` to copy the contents.



最悪で最高のバディドラマ「ダブルデッカー」キャストに聞く注目ポイントとは？



Learn More

Now in your SSH window, right-click and click “**Paste**”.

- 4.** Once you have copied the contents of your credentials over to our nano session, we can then save the file by pressing `CTRL + X` then `Y` and then finally hitting `ENTER`.

[How to setup Raspberry Pi Retr...](#)

- 5.** Now with the credentials file now saved safely to our Raspberry Pi we will start installing what we rely on.

Run the following command to install Python3 and the Python 3 Virtual Environment t

```
Terminal $
```



- 6.** We can now enable python3 as our virtual environment variable by running the following command on our Raspberry Pi.

```
Terminal $  
  
python3 -m venv env
```

- 7.** With that now enabled we can go ahead and ensure that we have installed the latest versions of `pip` and the `setuptools`.

To update these Python packages, we will need to use the following command on the Raspberry Pi.

```
Terminal $  
  
env/bin/python3 -m pip install --upgrade pip setuptools --upgrade
```

- 8.** To get into this new Python environment that we have set up we should run the following command in terminal.

```
Terminal $  
  
source env/bin/activate
```

- 9.** Now that we have all the packages we need to install the Google Assistant Library.

To do this we will run the command below to utilize pip to install the latest version of the Python package.

```
Terminal $  
  
python3 -m pip install --upgrade google-assistant-library  
python3 -m pip install --upgrade google-assistant-sdk[samples]
```

How to setup Raspberry Pi Retr... X

- Now that we have set up all the prerequisites to running the Google Assistant software on our Raspberry Pi we can finally complete the last few steps to authorize the system.

To do this, we must first install the Google authorization tool to our Raspberry Pi.

This package will allow us to authenticate our device and give ourselves the rights to be able to make Google Assistant queries for your Google Account.

Run the following command on the Raspberry Pi to install the Python authorization tool.

Terminal \$

```
python3 -m pip install --upgrade google-auth-oauthlib[tool]
```

- With the Google Authentication library now installed, we need to run it. To do this, we will be running the following command on our Raspberry Pi.

This command will generate a URL you will need to go to in your web browser so be prepared.

Terminal \$

```
google-oauthlib-tool --client-secrets ~/googleassistant/credentials.json \
--scope https://www.googleapis.com/auth/assistant-sdk-prototype \
--scope https://www.googleapis.com/auth/gcm \
--save --headless
```

- You will now be presented with the text "**Please visit this URL to authorize this application:**" followed by a very long URL.

Make sure you copy this URL entirely to your web browser to open it.

How to setup Raspberry Pi Retr...



- On this screen login to your Google account, if you have multiple accounts make sure your API key with.

Afterward, you should be presented with a screen with the text "**Please copy this code, switch to your application and**



If the authentication was accepted you should see the following line appear on your command line:

```
"credentials saved: /home/pi/.config/google-oauthlib-tool/credentials.json"
```

## 5. Now with the authentication credentials now saved, Google still requires us to agree to some stuff.

With this command make sure you replace `<projectid>` with the id of your project.

If you don't know what your Project ID is you can go to [Actions Console on Google](#), click the project you created, then click the Cog in the top left-hand corner then "**Project Settings**".

Also, make sure you replace `<deviceid>` with the device ID that you obtained earlier in the tutorial.

If you have lost your device ID can find it again by going to the Console Actions website, clicking the project you created then going to "**Device registration**"

To talk with the Assistant press `ENTER` in the terminal and speak an action such as "**What is the time**".

If everything is working as it should be you will hear a verbal response from the assistant.

Terminal \$

```
googlesamples-assistant-pushtotalk --project-id <projectid> --device-model-id <deviceid>
```

## 6. If you don't hear any response from the Google Assistant but a new tab does open that contains your results, then you should go back to the "**Setting up your Audio for Google Assistant**" section and the "**Testing your Audio for Google Assistant**" section.

In the next section, we will show you how you can rerun the Google Assistant without the steps again, make sure you keep your "**Project ID**" and "**Device ID**" handy as we samples.

[How to setup Raspberry Pi Retr...](#)



Using the Google Assistant on the Raspberry Pi



- 1.** Now that we have finally fully authorized our Raspberry Pi we will now walk you through the steps on how to run the Google Assistant software without having to go through the entire tutorial.

To begin with whenever you start a new terminal session you will need to put it back into the environment that we set up the Google Assistant software in.

To do this, run the following command on your Raspberry Pi.

Terminal \$

```
source env/bin/activate
```

- 2.** You will know whether this has worked correctly by seeing (**env**) appear at the front of each line.

With (**env**) appearing it means you can once again make calls to the Google Assistant samples.

To start up the push to talk sample you will need to run the following command, lucky for us, this time we do not need to write in the project id or the device id as these were cached when we first utilized the push to talk tool.

Terminal \$

```
googlesamples-assistant-pushtotalk
```

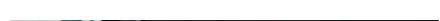
- 3.** In addition to the push to talk sample, there is also the hotword sample code. This sample code will work by listening for certain phrases to trigger it. By default, this will listen for “**Ok Google**” followed by the command that you want to issue to it.

Starting the hotword sample is very much like the push to talk sample, however you are utilizing the googlesamples-assistant-hotword utility instead.

Terminal \$

```
googlesamples-assistant-hotword --device-model-id <deviceid>
```

How to setup Raspberry Pi Retr...



**4.** Hopefully, by now you will have successfully set up your Raspberry Pi with the Google Assistant software.

If you run into audio issues, we recommend you go back to our "**Setting up your Audio for Google Assistant**" section and the "**Testing your Audio for Google Assistant**" section.

## ⚡ Getting the Google Assistant to Start on Startup

**1.** Now that you have got your Google Assistant up and running you will likely want to get it to start up on boot rather than having to go to the effort of entering the commands every time.

The easiest way to achieve this is to create a service for it. This service will allow the Google assistant to run in the background on the Raspberry Pi and easily allow us to retain control over it.

To begin, we will write our bash script that will execute the hotword sample. We will be calling this bash script from our service file.

Terminal \$

```
nano /home/pi/start_assistant.sh
```

**2.** To this file, you will want to add the following lines.

The first line helps define what should be utilized to run this file, so when the command line interprets it, it will know that it is required to make use of bash.

The second line we use **source** to read and execute commands from the activate file, this will put us into the python environment that we created earlier in the tutorial.

Finally, we run the hotwords Google Assistant sample. Make sure you replace <**deviceid**> with yours.

Add >

```
#!/bin/bash

source /home/pi/env/bin/activate
googlesamples-assistant-hotword --device-model-id <deviceid>
```

How to setup Raspberry Pi Retr...



**3.** Once you have added those lines to the file, save it by pressing **CTRL + X** then **Y** a



- 4.** Now that we have created our bash script we can move onto creating our service file for our Google Assistant by running the following command on our Raspberry Pi.

```
Terminal $  
sudo nano /lib/systemd/system/assistant.service
```

- 5.** Within this file enter the following lines.

These lines tell the operating system how to handle our service, what user to run it with and if it relies on a particular thing such as that the network is available.

```
Add >  
  
[Unit]  
Description=Google Assistant  
Wants=network-online.target  
After=network-online.target  
  
[Service]  
Type=simple  
ExecStart=/bin/bash /home/pi/start_assistant.sh  
Restart=on-abort  
User=pi  
Group=pi  
  
[Install]  
WantedBy=multi-user.target
```

[How to setup Raspberry Pi Retr...](#) 

- 6.** With those lines added to the service file, save it by pressing `CTRL + X` then `Y ↵`



To do this we just need to use systemctl, type in the following command to enable our new service.

Terminal \$

```
sudo systemctl enable assistant.service
```

**8.** With the service now enabled let's try starting it, this will allow us to make sure that everything is working as intended.

Terminal \$

```
sudo systemctl start assistant.service
```

**9.** You should now be able to use the "**Ok Google**" hotword to talk and interact with the Google Assistant.

Try it now to ensure the service is working as intended.

If for some reason it is not working you can utilize the command below to display the last output from the Google Assistant sample, keep a lookout for any errors.

Terminal \$

```
sudo systemctl status assistant.service
```

**10.** Hopefully, at this point, you will now have your Google Assistant successfully set up to start on boot up.

Love personal assistants, then you might also want to check out our tutorial on setting up [Alexa on the Raspberry Pi](#). It's pretty easy to set ip and quite a bit of fun once you get it working.

If you hear a response, you have successfully set up your very own Google Assistant on your Raspberry Pi. If you want to provide feedback or discuss this Raspberry Pi Google assistant project, then head to our forums.

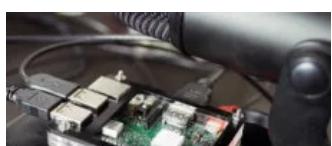
Get tutorials delivered to your inbox weekly.

[Sign up »](#)

[How to setup Raspberry Pi Retr...](#)



## Recommended



[The Basics of the chown Command](#)[Screensaver on the Raspberry Pi](#)[Raspberry Pi Email Server using Citadel](#)

## 53 Comments

---



**JagShar** on July 3, 2021 at 9:28 pm

Great Tutorial,  
Everything works fine!

But I am unable to use hotword as googlesamples-assistant-hotword command is not found..anyone came across this situation and is there any solid solution for this?

[Reply](#)



**Emmet** on July 4, 2021 at 2:00 am

 Editor

[How to setup Raspberry Pi Retr...](#) 

Hi JagShar,

Sadly Google deprecated the hotword code from their Python library and its cor originally hoping to come up with a solution using the Snowboy hotword detect has closed down all of their services.

Will hopefully eventually be able to provide a tutorial on showing how to implem

Emmet



**Marius ten Napel** on July 2, 2021 at 12:55 am

Hi,

Nice tutorial!

Do you have a solution for the OK Google command not found? I run googlesamples-assistant-hotword I get command not found.

Also on the google website... "Assistant" word seems to be part of it.

Thanks,  
– Marius

[Reply](#)



**Emmet** on July 2, 2021 at 6:35 pm

Editor

Hi Marius,

Sadly I haven't actually found a decent enough solution for this. There was originally the snowboy hotword engine, but that team sadly abandoned it and killed off the website that allowed you to create your own hotwords.

Cheers,  
Emmet



**Thorsten** on April 28, 2021 at 5:04 pm

I have successfully managed to set up Google Assistant on my Raspberry Pi 4, but the days/weeks it loses the credentials and I have to re-enter the auth code. After that even can I do to get credentials that don't expire?? d; - )

[How to setup Raspberry Pi Retr...](#)

X

[Reply](#)



**Brian Tristam Williams** on April 28, 2021 at 7:02 pm

X

I'm using the Google YouTube APIs, refresh token function. Not sure if y

ts an hour, then my code has to run a



Arsalan a

arecord : invalid option '-1'  
Try arecord --help for more information

Reply



Hi Arsalan,

You used the number 1, instead of t

Cheers,

Emmet

2

## Leave a Reply

Your email address will not be published.

Comment

[How to setup Raspberry Pi Retr...](#)



## Trending



Notify me of

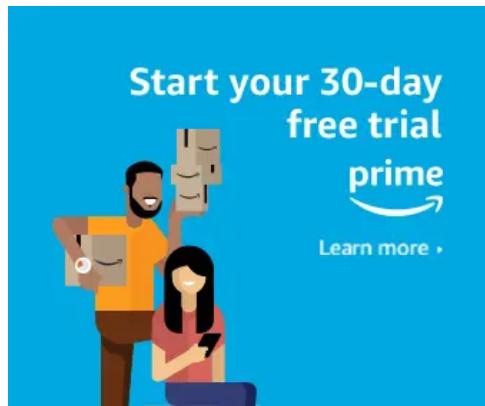


[Post Comment](#)

[How to Setup Surfshark on the Raspberry Pi](#)



[How to Run Steam on a Raspberry Pi](#)



[How to setup Raspberry Pi Retr...](#)



How to setup Raspberry Pi Retr... 



How to setup Raspberry Pi Retr... 



How to setup Raspberry Pi Retr... 



How to setup Raspberry Pi Retr... 



How to setup Raspberry Pi Retr... 



How to setup Raspberry Pi Retr... 



How to setup Raspberry Pi Retr... 



How to setup Raspberry Pi Retr... 



How to setup Raspberry Pi Retr... 



How to setup Raspberry Pi Retr... 



How to setup Raspberry Pi Retr... 



How to setup Raspberry Pi Retr... 



How to setup Raspberry Pi Retr... 



How to setup Raspberry Pi Retr... 



How to setup Raspberry Pi Retr... 



How to setup Raspberry Pi Retr... 



How to setup Raspberry Pi Retr... 



How to setup Raspberry Pi Retr... 



How to setup Raspberry Pi Retr... 



How to setup Raspberry Pi Retr... 



How to setup Raspberry Pi Retr... 



How to setup Raspberry Pi Retr... 



 [Back to Top](#)

© 2021 Pi My Life Up

[Disclaimer & Privacy Policy](#) | [About us](#) | [Contact](#)

---

AN ELITE CAFEMEDIA PUBLISHER

[How to setup Raspberry Pi Retr...](#) 

