Instructions to set up Raspberry Pi and run Seajei Demo with iOS app

Version 3.2

Hardware to use

Raspberry Pi

The Seajei demo has been tested on the following models:

- Raspberry Pi 0
- Raspberry Pi 3 Model B and B+
- Raspberry Pi 4

WARNING: When using the latest Raspberry Pi OS based on (Debian 10 (Buster), the audio player does not work properly (at this time the PortAudio library used inside the player is unable to play audio in Debian 10).

Unfortunately Pi 4 requires Debian 10. So unless you don't currently require playing audio on the Raspberry Pi, we recommend using a Raspberry Pi 3 or 0 instead, running Debian 9.

Camera Module

Two different camera modules were tested:

- Raspberry Pi 3 2 model B B+ A+ Mini Camera Video Module 5MP 1080p OV5647 Sensor with 15 Pin FPC Cable
- Arducam 5 Megapixels 1080p Sensor OV5647 Mini Camera Video Module for Raspberry Pi Model A/B/B+, Pi 2 and Raspberry Pi 3,3B+

Microphone

Any USB microphone can be used to capture audio on the Raspberry Pi.

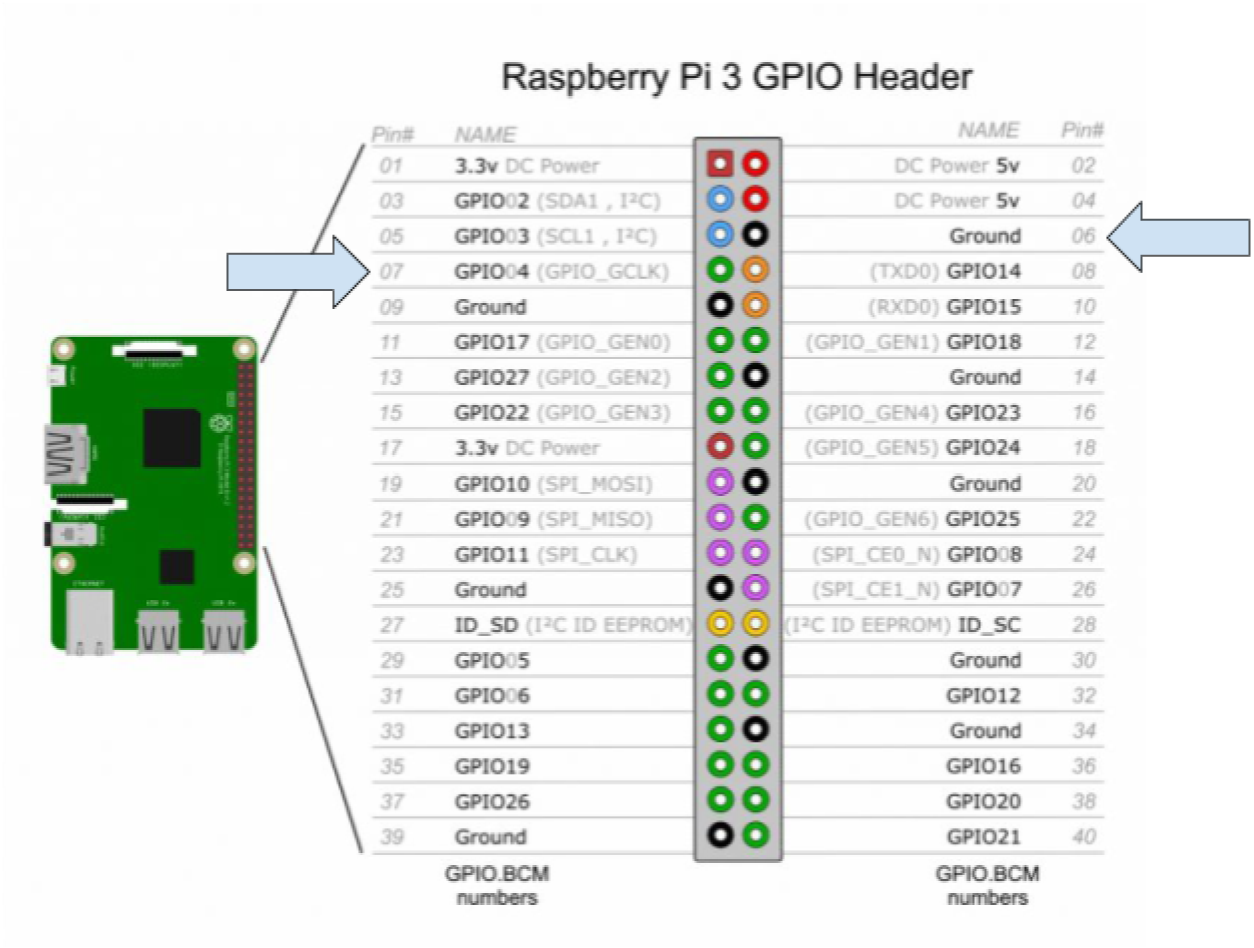
It is recommended to use a higher quality microphone for the best possible experience.

WARNING: the microphone needs to be plugged in before the executable starts.

Push button

Any simple push button can be used for push notifications. It simply needs to close its circuit when pushed.

For the button to work with the sample doorbell program in the SDK, connect one wire to pin #7 (4th from the top on right column), and the other to a ground pin (for example pin #6, 3rd from the top on right column).



Raspberry Pi 3 GPIO Header

Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I²C)	DC Power 5v	04
05	GPIO03 (SCL1 , I²C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I²C ID EEPROM)	(I²C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40
	GPIO.BCM numbers	GPIO.BCM numbers	

Raspberry Pi basic installation and setup

Download and install operating system

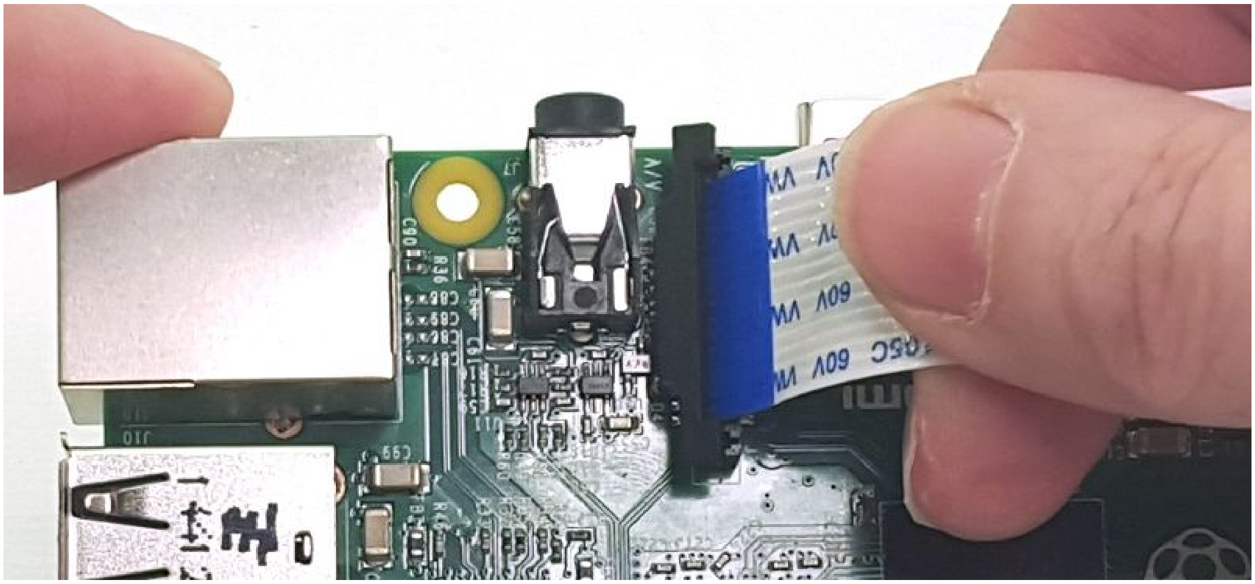
Download and install a new Raspberry Pi OS by using the Raspberry Pi Imager.

It is recommended to install Debian 9 (Stretch) (CjAudioPlayer will not work in Debian 10). To do so, download Debian 9. Then when using the Raspberry Pi Imager and selecting an OS, select "Use Custom" and point to the downloaded image.

Attach and enable camera hardware

Follow the instructions on the following page to connect and enable the camera:

https://www.raspberrypi.org/documentation/configuration/camera.md



If you are having trouble to enable the camera module, after running:

```
> sudo raspi-config
```

Go to "5 Interfacing Options" -> "P1 Camera" to enable.

Prepare audio input and output

Install portaudio

```
> sudo apt install portaudio19-dev
```

Set volumes

The following steps are for Raspberry PI OS based on Debian 9 (Stretch). If using the latest Debian 10 (Buster), the steps are slightly different but similar, but remember as mentioned earlier that the audio player does not currently work in Debian 10.

Output (speaker)

To set audio output and volume, click on the speaker icon at the top right of the screen: 🔊

And set the volume to near maximum.

To set audio output to analog (headphone jack), right click on the speaker icon, and select *Analog*. Re-check the volume setting.

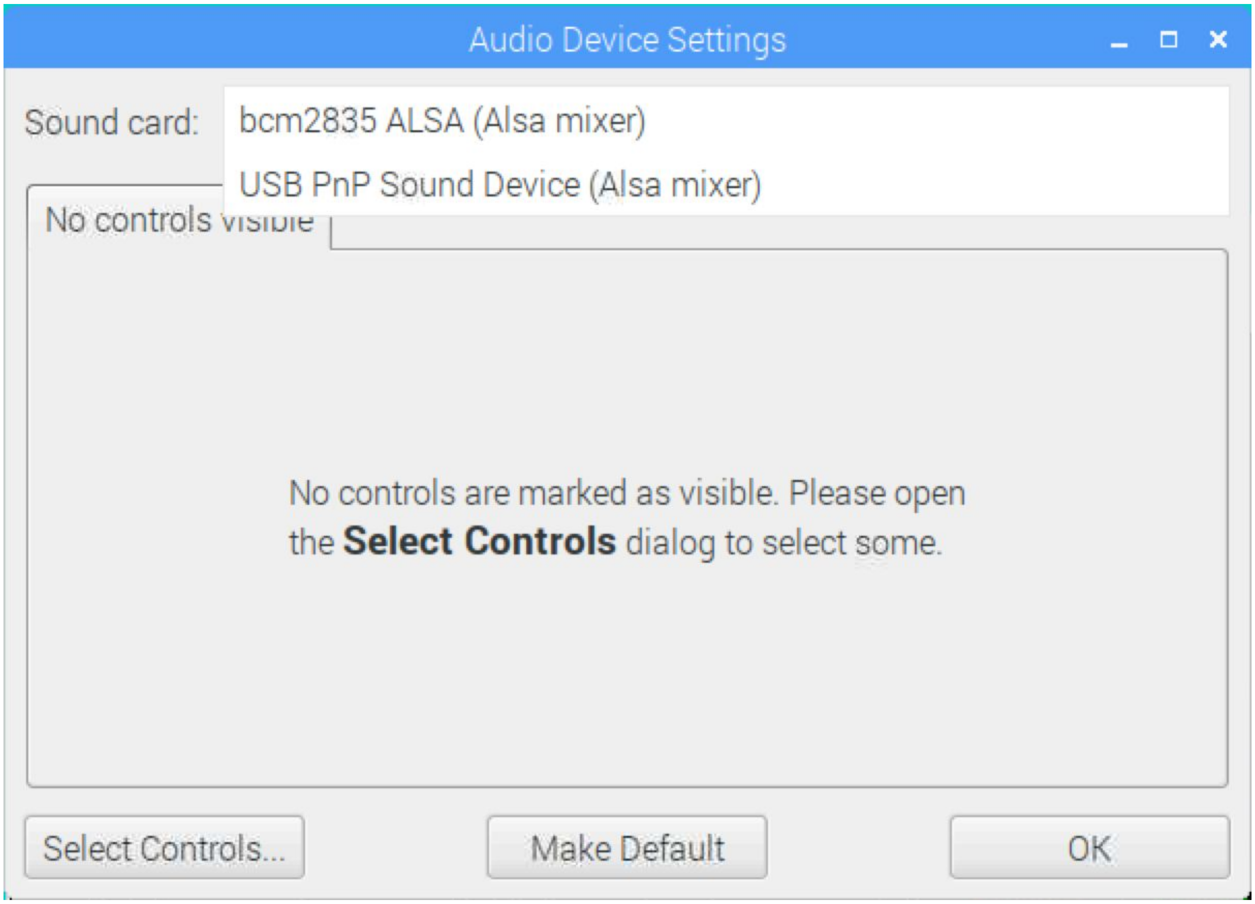
We have seen issues when audio is sent out through HDMI so we suggest always sending it via headphone jack (Analog).

Input (microphone)

The microphone gain may be very low by default.

To set microphone input gain, right click on the speaker icon at the top left 🔊, and select *USB Device Settings*.

A dialog will open. Click next to *Sound card*, and select *USB PnP Sound Device*. If that option doesn't show, be sure to plug in a USB microphone and try again.



Audio Device Settings

Sound card: bcm2835 ALSA (Alsa mixer)
USB PnP Sound Device (Alsa mixer)

No controls visible

No controls are marked as visible. Please open the **Select Controls** dialog to select some.

Then click on *Select Controls…*, and click the checkbox next to *Microphone*.

Now you get a slider that by default is set to minimum value. Set it to somewhere near maximum:

Force audio output

If after going through the steps above, the sound output reverts to HDMI, you can try the following extra steps:

Edit the config.txt file:

```
> sudo vi /boot/config.txt
```

Append the following at the end:

```
hdmi_ignore_edid_audio=1
```

Alternatively, for a temporary solution while the Raspberry Pi is running (but may not persist through reboots), run the configuration:

```
> sudo raspi-config
```

Then go to `Advanced Options`, select `Audio`, and select `Force 3.5mm ('headphone') jack`.

Running the Raspberry Pi executable as a service

Whether you received a doorbell executable from Seajei or built it yourself with the SDK, you can set up your Raspberry Pi to run it as a service, so it runs automatically when you power on the device.

Assuming your executable is named `doorbell` and placed on the Desktop, create a file named raspidoorbell.service, and put the following inside:

```
[Unit]
Description=Raspi doorbell service
After=network.target

[Service]
Type=idle
ExecStart=/home/pi/Desktop/doorbell
Restart=on-failure
```

```
RestartSec=1s

[Install]
WantedBy=multi-user.target
```

Then copy file to proper location:

```
> sudo cp raspidoorbell.service /lib/systemd/system
```

Start service and set it to start on bootup:

```
> sudo systemctl start raspidoorbell.service
> sudo systemctl enable raspidoorbell.service
```

If the file gets edited, run the following:

```
> sudo systemctl daemon-reload
```

Reboot the device and the service should start the `doorbell` executable automatically (camera LED will turn on).

Because it is now a service, any time you will want to run the executable manually, you will first need to stop the service with the following command:

```
> sudo systemctl stop raspidoorbell.service
```

Check the LED light has turned off to confirm the service has stopped. Now you can run the `doorbell` executable via command line if you wish.

How to run custom demo received from Seajei

If you received the prebuilt Raspberry Pi directly from Seajei, as well as an invitation to the SeajeiDemoApp via TestFlight so that you can run your own company-centric demo, read on.

If on the other hand you compiled your own Raspberri Pi and iOS samples with the SDK, then this chapter does not apply to you.

Running Seajei Demo on Raspberry Pi

If you have been provided with a simple drop-in executable from Seajei, do the following.

Drop the executable provided by Seajei onto the Desktop or anywhere else.

The file should be named "doorbell_<your company name>_<random numbers>". For our example let's assume it's named `doorbell_abc_8325`.

Next open up a Terminal, and navigate to the directory where the file is. Assuming it is on the desktop navigate there by typing the following:

```
> cd ~/Desktop
```

Then make sure the file is executable by running the following:

```
> chmod +x doorbell_abc_8325
```

Now simply execute the program:

```
> ./doorbell_abc_8325
```

You should see some output indicating the library has initialized, as well as the unique ID to use in order to connect from the iOS app.

You will also see connection and disconnection events.

If you see an error it is likely because the camera module has not been enabled in the Raspbian settings, or because another program using the camera is already running.

Using the app to run the demo on iOS

If you received an executable to run on your Raspberry Pi, then you should also have received an invitation in your email to install the SeajeiDemoApp via TestFlight.

Run the app and tap on the text field to specify the unique ID of the doorbell Raspberry Pi sample that you are going to connect to.

The unique ID to use may be sent to you separately. It will also show in the debug output on Raspberry Pi when running the provided executable.

Typically it will be similar to the executable file you received. In our example the file received was `doorbell_abc_8325`. The corresponding unique ID to use would be: `abc_8325`.

Now tap on CONNECT, and you should see the live stream!