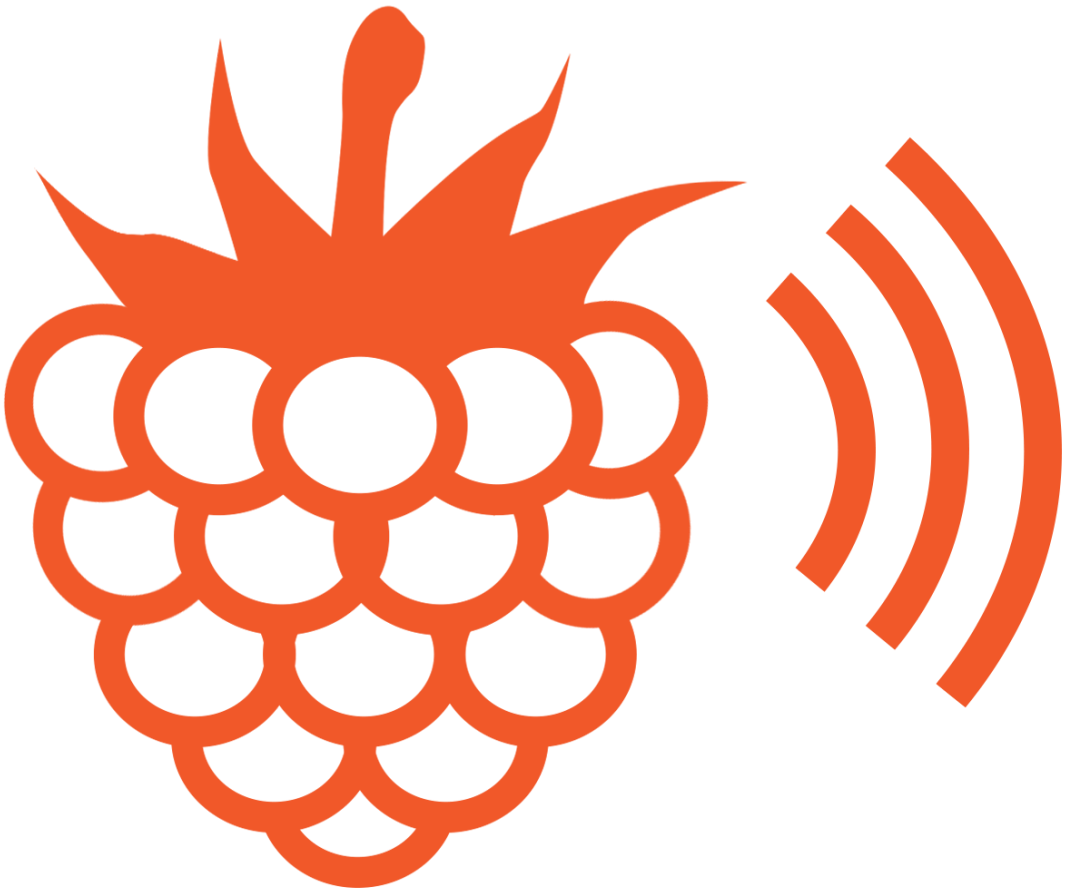

piDENT

Building `pident`

The most recent version of `pident`, *1.12.x*, runs from a linux server. While it should be possible to run `pident` on any distribution of linux, this guide for building assumes you are running **Ubuntu 20.04 LTS**.

`pident` is build with ReactJS and uses Node for various interface options. There are different GitHub repos for different things. There is the main application, each site has it's specific audio (although this will change in the future), and there is a repo of various files you'll need along the way.

Let's start.

Pre-requisites

`pident` isn't overly resource heavy, and as it is built as a SPA it is platform agnostic. However, to utilise all the available features of `pident`, you need to be running linux or, at a push, macOS.

Assumptions

- You've installed a clean copy of Ubuntu 20.04 LTS
- You have a working and tested network connection
- You have root (or `sudo`) access
- You're comfortable with basic terminal commands (although this guide will tell you exactly what you'll need to use, it's good to have an idea of why we are doing what we are doing, and to be able to understand the output if there are any errors)
- You have a GitHub account and have been given access to the repo

Required packages

We will be installing several packages to be able to use all the features of `pident`, including:

- NodeJS (which includes npm, the package manager)
- Python (while you can install any version you need for other projects, this build specifically requires 2.7 and won't work with 3.x yet)
- Git (you'll need this to get the actual application, as well as download your site's audio)
- Streamdeck UI
- Various packages to enable the USB interface to work

Beginning installation

First thing's first, let's make sure we are up to date:

```
$ sudo apt-get update && sudo apt-get upgrade -y
```

This executes two commands one after the other, the first checks the list of packages we have against a list of the most recent versions, the second (split by `&&`) then downloads them. The `-y` at the end just means we don't need to agree to download anything half way though and the command will execute completely without other interactions by us.

Now we can install some packages.

Node and NPM

First we will check if `curl` is installed:

```
$ curl
```

If you see the output as something like `curl: try 'curl --help' or 'curl --manual' for more information` then `curl` is installed and you can skip to the next step. If not, run:

```
$ sudo apt install curl
```

Once we have `curl` installed, we can go ahead and grab the NodeJS package:

```
$ curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -
```

When that command has been executed, we can go ahead and install node using `apt`:

```
$ sudo apt install nodejs -y
```

This will install both node and npm. To check that the installation has worked correctly, we can check for versions:

```
$ nodejs -v
```

which should output something like

```
$ v12.18.3
```

and similarly for npm run:

```
$ npm -v
```

which should output

```
$ v6.14.6
```

That's it for node and npm being installed. We will come back to npm later to install some dependencies for `pident` but for now, we will move on to the next thing.

Python 2.7

In order for the Streamdeck to function correctly, we need to install Python 2.7, and NOT 3.x as this won't work.

First let's make sure we have the required packages installed. You can type these out one at a time if you like, but it's probably easier to copy and paste them to make sure there are no mistakes:

```
$ sudo apt-get install build-essential checkinstall
```

and then run (all as one line):

```
$ sudo apt-get install libreadline-gplv2-dev libncursesw5-dev libssl-dev
libsqlite3-dev tk-dev libgdbm-dev libc6-dev libbz2-dev
```

This will install everything we need, and next we can get and install python.

Change directories into the users src folder:

```
$ cd /usr/src
```

and then download the compressed package:

```
$ wget https://www.python.org/ftp/python/2.7.18/Python-2.7.18.tgz
```

We then need to extract this, and you'll need to do this as root (or `sudo`):

```
$ sudo tar xzf Python-2.7.18.tgz
```

And then we are ready to complie the source code. The final command uses `make altinstall` so that if you have another version of python installed we don't overwrite it. Run these commands one after the other:

```
$ cd Python-2.7.18
$ sudo ./configure --enable-optimizations
$ sudo make altinstall
```

Finally we want to check that everything is working as it should by running:

```
$ python2.7 -V
```

which should output:

```
$ Python 2.7.18
```

Congrats, we now have python installed!

Git

This one is fairly straight forward. First let's check if you already have `git` installed:

```
$ git --version
```

If you do, you'll get a version number back, and if not, you'll get a command not found error. If you have it installed, skip ahead, if not, run:

```
$ sudo apt install git
```

Then rerun the test above to verify that everything worked to plan.

You can also complile git from source, but that adds a layer of complexity that we simply don't need for this project.

Git is now installed. Moving on.

streamdeck_ui

The Elgato Streamdeck has been a popular choice for physical controllers for a number of years due to it's simplicity and durability. However, Elgato only support Windows or macOS. Fear not, there is a version for linux. It requires a bit of tinkering but once up and running it should just work. Unfortunately there is some manual configuration needed, but that does mean you can set the layout as you please.

Installing streamdeck_ui

First off, let's get the packages we require:

```
$ sudo apt-get install libhidapi-hidraw0 libudev-dev libusb-1.0-0-dev
```

Then we need to add our user to the *plugdev* group so we can access the Streamdeck:

```
$ sudo usermod -a -G plugdev USERNAME
```

Change `USERNAME` to whatever your username is.

We need to add some *udev* rules, we'll use `nano` but you can use whichever editor you like:

```
$ sudo nano /etc/udev/rules.d/99-streamdeck.rules
```

Add these lines (again it is easier to copy and paste to avoid mistakes):

```
SUBSYSTEM=="usb", ATTRS{idVendor}=="0fd9", ATTRS{idProduct}=="0060",
MODE:="666", GROUP="plugdev"
SUBSYSTEM=="usb", ATTRS{idVendor}=="0fd9", ATTRS{idProduct}=="0063",
MODE:="666", GROUP="plugdev"
SUBSYSTEM=="usb", ATTRS{idVendor}=="0fd9", ATTRS{idProduct}=="006c",
MODE:="666", GROUP="plugdev"
SUBSYSTEM=="usb", ATTRS{idVendor}=="0fd9", ATTRS{idProduct}=="006d",
MODE:="666", GROUP="plugdev"
```

Then reload the rules with:

```
$ sudo udevadm control --reload-rules
```

Now if you already have your Streamdeck plugged in, you need to un-plug and re-plug it now. If you haven't, you'll need to plug it in now.

Now let's install the streakdeck_ui package using the python package manager:

```
$ pip3 install --user streakdeck_ui
```

And then we can launch streamdeck with the command:

```
$ streamdeck
```

This should load the GUI of streamdeck, but you may find that you get an error message saying "`streamdeck: command not found`". This isn't a problem, all you need to do is to change the command to have the entire path:

```
$ /home/USERNAME/.local/bin/streamdeck
```

Again, change `USERNAME` to your username.

The last thing to do is to make sure that streamdeck starts as the system starts. Ubuntu makes this really easy. Open up the Startup Applications app from the software launcher, then click Add to add a new application, and type the command above in the Command box to run streamdeck on startup. Save all of that and then you can test with a system reboot.

Configuring streamdeck_ui

Now we have the streamdeck GUI installed, we need to make sure that it can work with `pident`. For simplicity, `pident` listens for keyboard input and runs various functions based on the key that is pressed.

API

Key	Command
t	Toggles the new/old state
d	Runs the bed
b	Plays the loaded bong
s	Plays the STC stab
i	Plays the loaded ident

You can add the key to whichever button you like. Included in the misc repo which we will download later are some icons for use with the streamdeck, but you can add your own if you like.

Cloning repos

There are three repos that we will need to clone. They are listed below with a brief description of what they contain.

```
https://github.com/quartersight/pident-spa.git
```

- This repo contains the source code for the `pident` app which we will build later

  https://github.com/quartersight/SITENAME-audio.git

- This repo will contain the "starter" audio for your site
- Changing `SITENAME` to your site in lowercase will get the right audio
- This repo has the audio furniture in the root and all idents in a subfolder
- The idents are named in lowercase as *PROGTYPE-HOUR-NETWORKINITIALS-LOCALINITIALS.wav*

  http://github.com/quartersight/pident-misc.git

- This repo contains various other files we need to set up a `pident` server, such as icons and logos

Now we understand a bit more about what's in each repo, let's get set up to clone them.

First, let's make sure we are in our home directory:

```
$ cd ~
```

And we will save our GitHub credentials the first time we use them:

```
$ git config --global credential.helper store
```

Then we can start by cloning the `pident`

```
$ git clone https://github.com/quartersight/pident-spa.git
```

Input your username and password for GitHub.

Once this action is completed you will have a folder in your home directory called `pident-spa`

Next we will move into the new directory and clone the audio:

```
$ cd pident-spa
$ git clone https://github.com/quartersight/SITENAME-audio.git
```

Change `SITENAME` to your site name.

This may take some time depending on your network speeds, but once complete you will have a folder with all the audio in.

The last thing to do is to clone the misc items. Stay in the pident-spa folder and run:

```
$ git clone https://github.com/quartersight/pident-misc.git
```

That's it! You now have everything you need to build and deploy `pident`!

Node packages

We need some packages to work with node for some features of `pident` to work. This is where we will talk through installing them.

The two we need are:

- serve
- robotjs

We will install them globally, so need to use `sudo`, but will install them in our pident-spa folder:

```
$ cd ~/pident-spa
$ sudo npm install -g serve robotjs
```

Now we can make use of these when we sort out the autoboot.

Building `pident`

`Pident` is fairly simple to build, just run the following commands:

```
$ cd pident-spa
$ npm run build
```

And then serve it (ie run it) with

```
$ serve -s build
```

`Pident` will now be running on `http://localhost:5000`

Moving the audio

We need to move the audio we cloned into the right place.

```
$ cd pident-spa
$ cp -r SITENAME-audio build/audio
```

Replace `SITENAME` with your site name. This copies everything into the build folder.

Setting up the autoboot

As well as auto-booting the streamdeck interface which we set up earlier, there are a few other things we need to do at boot. First, let's start the server and start `pident`, and also start the GPI listener. Both done with `crontab`:

```
$ crontab -e
```

If this is the first time that you've edited your crontab, you'll have to pick an editor. We used `nano`. Add these two lines to the end of the file:

```
@reboot path/to/nodejs path/to/gpi.js
@reboot cd path/to/pident-spa && serve -s build
@reboot google-chrome --start-fullscreen --app=http://localhost:5000
```

Then exit by pressing `ctrl+x` to exit, `y` to save, and then `enter` to confirm the name.

That's it. Pident should now be up and running. To get things going, just reboot:

```
$ sudo reboot
```

And we're done.