Raspberry Pi

Notes

- Setup notes
- List of config.txt options
- Google AIY voice kit v1
- WireGuard

Setup notes

Writing an OS image to the sdcard (or usb drive)

```
sudo dd bs=4M if=/path/to/os/image of=/dev/mmcblk0 status=progress
```

Headless wifi setup

Mount the boot partition and create a file called *wpa_supplicant.conf* with the following contents:

```
country=US
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="SSID"
    scan_ssid=1
    psk="PASSWORD"
    key_mgmt=WPA-PSK
}
```

Enable SSH prior to first boot

Mount the boot partiton and create an empty file called *ssh:*

```
touch /path/to/boot ssh
```

Initial setup

Use the *raspi-config* utility to set the follwing options as desired:

- Network Options > Hostname > HOSTNAME
- Localization Options > Change Locale > *en_US.UTF-8*
- Localization Options > Change Time Zone > *America/New_York*
- Localization Options > Change WLAN Country > *US*
- Interfacing Options > SSH > *Yes*
- Interfacing Options > SPI > *Yes*
- Interfacing Options > I2C > *Yes*
- Advanced Options > Overscan > No
- Advanced Options > Memory Split > 32
- Advanced Options > Screen Blanking > No

Changing the default username (pi)

> This must be done while logged in as the root user. It cannot be done (with sudo or su) while the pi user is looged in.

> To enable root login, edit */etc/passwd,* removing the 'x' after the first colon on the line that begins with `root:x:...`. Also, if attempting this over SSH, the */etc/sshd.conf* file will need to be modified, changing the line `PermitRootLogin=prohibit-password` to `PermitRootLogin=yes`. Don't forget to revert these changes once the pi user has been renamed.

```
usermod -l USERNAME pi
usermod -d /home/USERNAME -m USERNAME
groupmod -n USERNAME pi
rm /etc/sudoers.d/010_pi-nopasswd
```

Disable WiFi Power Managment

Create the file `/etc/network/interfaces.d/wlan0` with the following contents:

```
allow-hotplug wlan0
iface wlan0 inet manual
wireless-power off
```

List of config.txt options

Miscellaneous config.txt options:

Display

```
disable_splash=1  # Disable the rainbow splash screen on bootup
lcd_rotate=2  # Rotate lcd displays like the official touchscreen 180 degrees;
```

LEDs

```
[pi0]
dtparam=act_led_trigger=none  # Green LED, available triggers are listed in
/sys/class/leds/led0/trigger, default is mmc0
dtparam=act_led_activelow=off  # inverts the activity state

[all]
dtparam=act_led_trigger=none  # Green LED, available triggers are listed in
/sys/class/leds/led0/trigger, default is mmc0
dtparam=act_led_activelow=off

dtparam=pwr_led_trigger=none  # Red LED, available triggers are listed in
/sys/class/leds/led1/trigger, default is default_on
dtparam=pwr_led_activelow=off
```

Wireless

```
dtoverlay=disable-bt  # Disable buit-in bluetooth
dtoverlay=disable-wifi  # Disable built-in wifi
```

Google AIY voice kit v1

Enable the driver:

Edit/Add to */boot/config.txt:*

```
dtoverlay=i2s=on
...
#dtparam=audio=on
...
# Google AIY voice kit v1
dtoverlay=i2s-mmap
dtoverlay=googlevoicehat-soundcard
```

ALSA configuration:

Create */etc/asound.conf:*

```
options snd_rpi_googlevoicehat_soundcard index=0

pcm.softvol {
    type softvol
    slave.pcm dmix
    control {
        name Master
        card 0
    }
}

pcm.micboost {
    type route
    slave.pcm dsnoop
    ttable {
        0.0 30.0
        1.1 30.0
```

```
    }
}

pcm.!default {
    type asym
    playback.pcm "plug:softvol"
    capture.pcm "plug:micboost"
}

ctl.!default {
    type hw
    card 0
}
```

Reboot after editing *asound.conf.*

Verify:

Verify that the sound card's output and input is recognized with

```
aplay -l
arecord -l
```

Test:

Record a 5 second *test.wav* with:

```
arecord -c 2 -f cd -d 5 test.wav
```

Play *test.wav* with:

```
aplay test.wav
```

Pulseaudio

Install the `pulseaudio` package, and optionally `pulsemixer` for volume control. The microphone volume is low under the default ALSA driver, pulseaudio can 'overdrive' the source volume. 200-

250% should be sufficent.

After installed, start the pulseaudio server with `pulseaudio -D`. List sources (inputs) with `pactl list sources`, note the index # of the soundcard, make sure it's not the null-monitor or loopback from the sink (output). Use that index number to increase the source volume, for example if the source index is #1: `pactl set-source-volume 1 250%`

WireGuard

WireGuard® is an extremely simple yet fast and modern VPN.

Installation (Raspberry Pi 2/3/4)

> At the time of this writing, the WireGuard module has yet to be included in the Raspberry Pi kernel. The Debian Testing repository must be added to the Raspberry Pi in order to install the necessary tools and the kernel module.

Add the Debian Testing Repository

The following Commands will add the Debian testing repository and set it's priority lower than the Raspberry Pi OS stable repository. This way, all packages will be updated against the stable repos, unless they are not available in which case apt will fall back to check the Debian testing repo.

```
$ echo "deb http://archive.raspbian.org/raspbian testing main" | sudo tee --append
/etc/apt/sources.list.d/testing.list
$ printf 'Package: *\nPin: release a=testing\nPin-Priority: 50\n' | sudo tee --append
/etc/apt/preferences.d/limit-testing
$ sudo apt update
```

Install the WireGuard Package

```
$ sudo apt install wireguard -y
```

Allow Remote Access to the Local Network

- Configure a static IP address or DHCP reservation on the Raspberry Pi
- Configure a static WAN IP address or DDNS service on the router
- **Port forward UDP port 51820** from the router to the Raspberry Pi

Enable IP Forwarding

On the Raspberry Pi, edit `/etc/sysctl.conf` uncommenting the following line:

```
...
net.ipv4.ip_forward=1
...
```

This will require a reboot to take effect.

Create the the server/client keys

Start by creating a directory in a secure location for the keys. inside this directory run the following command for the "server" keys:

```
$ mkdir wg-configs
$ cd wg-configs
$ wg genkey | tee wg0_privkey | wg pubkey > wg0_pubkey
```

This will create two files, `wg0_privkey` and `wg0_pubkey` that each contain a hashed key.

For each "client" peer, run the same command as above, changing the output filenames to reflect which peer key-pair is being generated:

```
$ wg genkey | tee wg0client1_privkey | wg pubkey > wg0client1_pubkey
$ wg genkey | tee wg0client2_privkey | wg pubkey > wg0client2_pubkey
...
```

Server Configuration

Create the file `/etc/wireguard/wg0.conf` containing the following, replace items indicated by `<...>` with the key hashes found in the files created above.

Additional peers may be added by appending more `[Peer]` blocks.

```
[Interface]
Address = 192.168.2.1/24
PrivateKey = <wg0_privkey>
ListenPort = 51820  # udp
```

```
# allow access to local network from wireguard interface, change eth0 to wlan0 if using wifi
PostUp = iptables -A FORWARD -i %i -j ACCEPT; iptables -A FORWARD -o %i -j ACCEPT; iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
PostDown = iptables -D FORWARD -i %i -j ACCEPT; iptables -D FORWARD -o %i -j ACCEPT; iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE

[Peer]
PublicKey = <wg0client1_pubkey>
AllowedIPs = 192.168.2.2/32

[Peer]
PublicKey = <wg0client2_pubkey>
AllowedIPs = 192.168.2.3/32
```

Activate the wg0 interface with systemd

Enable and start the WireGuard Interface with:

```
$ sudo systemctl enable wg-quick@wg0
$ sudo systemctl start wg-quick@wg0
```

Client Configuration

> Note: the filename of the config file determines the WireGuard interface name, for example `wg0client1.conf` creates an interface called `wg0client1`. Interfaces can be named whatever you want, it may be helpful to name them after the network you're connecting to and/or the peer name

Very similar to the "server" configuration above, each client config should contain the following, again replacing the items indicated by `<...>` with the key hashes:

```
[Interface]
Address = 192.168.2.2/24
DNS = 192.168.1.1
PrivateKey = <wg0client1_privkey>

[Peer]
```

```
PublicKey = <wg0_pubkey>
AllowedIPs = 0.0.0.0/0  # routes all traffic
Endpoint = <DNS-resolvable-name>:51820
```

> The endpoint can be either the router's WAN Ip address or the (D)DNS name, but must also contain the port number of the server, i.e. www.franklin57.com:51820.

Create a config file or enter the information above for each "client" device, updating the `Address` and `PrivateKey` to match what is in the "server's" config and the specific client's public key file.