Github Repository: https://github.com/KeanuChandra/Server-Client

# 1: Raspberry Pi Router Setup Instructions

**Enable IP Forwarding**
In /etc/sysctl.conf make sure you have the line:
```
net.ipv4.ip_forward=1
```

Apply the change in the terminal with:
```
sudo sysctl -p
```

**Setup wlan0 to work as a router**
Add the following lines to the end of the /etc/dhcpcd.conf file:
```
interface wlan0
static ip_address=192.168.4.1/24
static routers=192.168.50.1
static_domain_name_servers=8.8.8.8 #I don't believe this is completely
necessary for our application
```

Install hostapd:
```
sudo apt install hostapd dnsmasq
```

Configure hostapd with by creating a /etc/hostapd/hostapd.conf file:
```
country_code=US
interface=wlan0
driver=n180211
ssid=[our network name]
hw_mode=g
channel=6
wmm_enabled=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=[password]
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsa_pairwise=CCMP
```

Change your configuration file by editing /etc/default/hostapd:
```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

Github Repository: https://github.com/KeanuChandra/Server-Client

Configure dnsmasq by adding the following lines to /etc/dnsmasq.conf:

```
interface=wlan0
dhcp-range=192.168.4.2,192.168.4.100,255.255.255.0,24h
```

**Setup DHCP server**

Install DHCP server tools:

```
sudo apt install isc-dhcp-server
```

Edit DHCP server configuration in /etc/dhcp/dhcpd.conf. Add the following lines to the end of the file:

```
subnet 192.168.4.0 netmask 255.255.255.0 {
      range 192.168.4.10 192.168.4.100;
      option routers 192.168.4.1;
      option domain-name-servers 8.8.8.8, 8.8.4.4; #Also optional I
think
}
```

In the /etc/default/isc-dhcp-server file add this line to the end:

```
INTERFACES="wlan0"
```

**Restart Services**

Use the following commands to restart services

```
sudo systemctl restart hostapd
sudo systemctl restart dnsmasq
sudo systemctl restart dhcpcd
sudo systemctl restart isc-dhcp-server
```

**Run Server on Boot**

Enter the following command and choose the /bin/nano option:

```
sudo crontab -e
```

In this file type the following at the bottom (Use whatever path your server.py is in):

```
@reboot python /home/romps/Desktop/server.py &
```

**Debugging**

You may need to stop NetworkManager with the following line if it is also using wlan0.
WARNING THIS WILL DISABLE THE ABILITY FOR THE RASPBERRY PI TO CONNECT TO OTHER NETWORKS

```
sudo systemctl stop NetworkManager
```

Github Repository: https://github.com/KeanuChandra/Server-Client

Use the following command to check the server status:

```
sudo systemctl status isc-dhcp-server
sudo systemctl status hostapd
```

If isc-dhcp-server failed to start use this command to check the logs:

```
sudo journalctl -u isc-dhcp-server
sudo journalctl -u hostapd
```

To manually load drivers (In the case of our current raspberry pi the driver is rt2800usb) use:

```
sudo modprobe [driver name]
```

If you are still encountering issues you might want to find other processes that may be blocking your hostapd/dhcp-server. Here are some examples of what you want to kill before restarting the services.

```
sudo killall hostapd
sudo killall wpa_supplicant
```

To check for devices currently connected to your network, use the following command:

```
arp -a
```

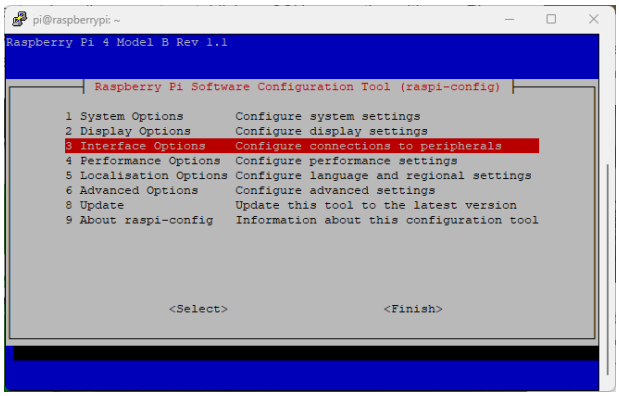
Github Repository: https://github.com/KeanuChandra/Server-Client

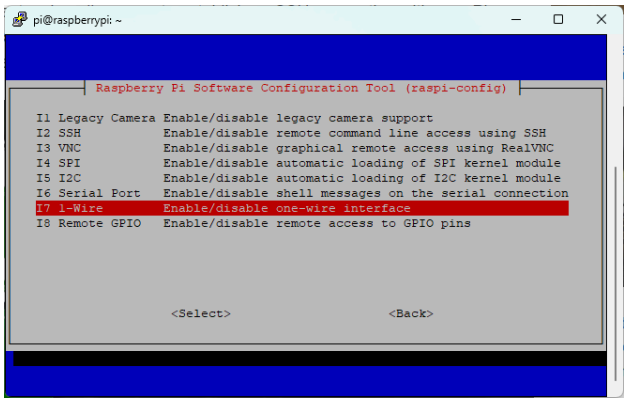
## 2: 1-Wire Setup Instructions

Instructions taken from Random Tutorials and Waveshare Wiki.

First, enable the 1-wire communication:

```
sudo raspi-config
```




Next, edit the config.txt to set up the 1-Wire Interface:

```
sudo nano /boot/config.txt
```

Then, add this to the end of the file:

```
# 1-Wire:

dtoverlay=w1-gpio-pullup,gpiopin=4
```

*You can change the GPIO pin if needed.*

Reboot:

```
sudo reboot
```

Once it is rebooted, hit the terminal with the:

```
sudo modprobe w1-gpio

sudo modprobe w1-therm
```

Github Repository: https://github.com/KeanuChandra/Server-Client