GippsTech 2017

Remote Radio Control Server

By

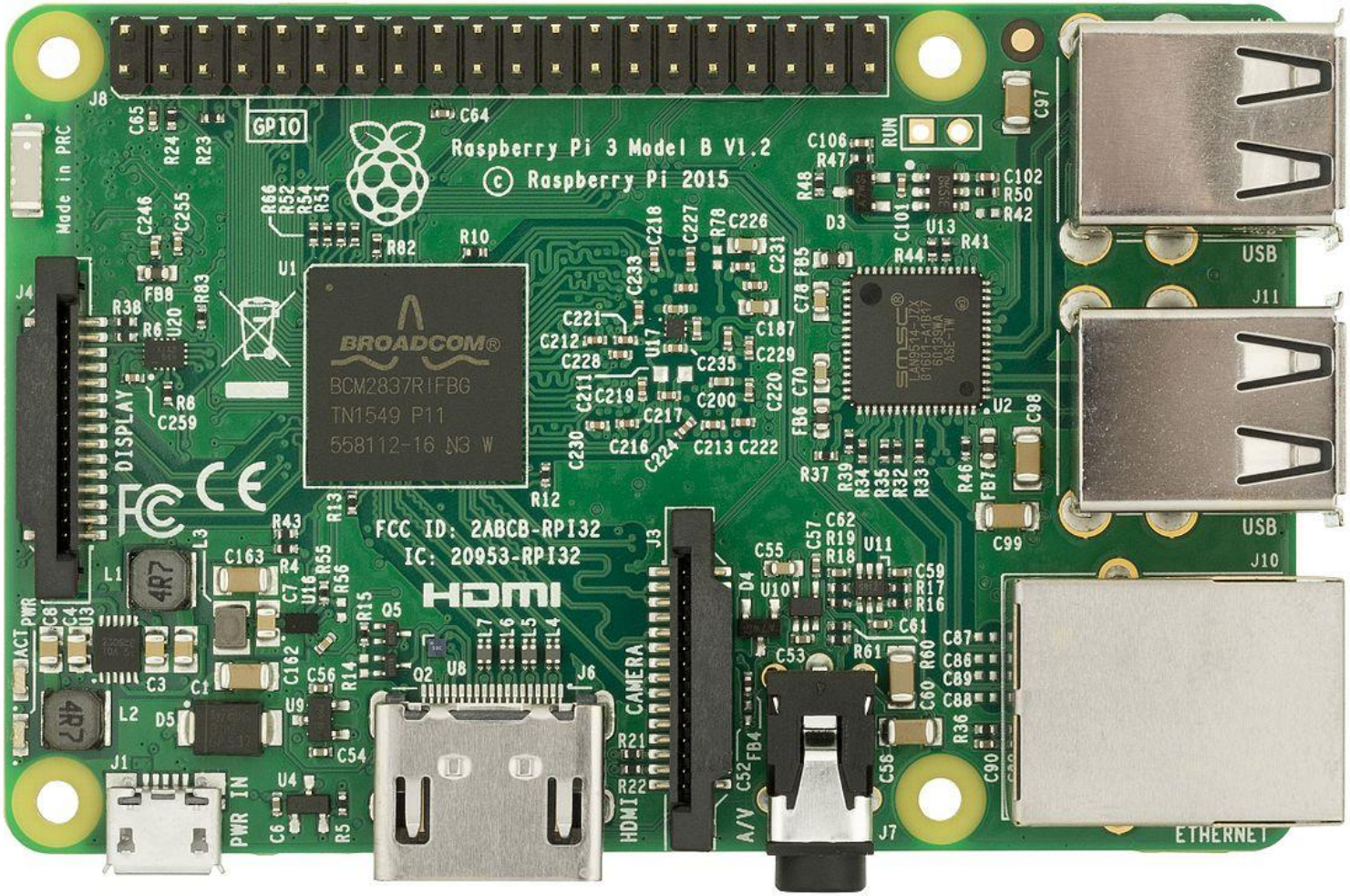
Julie VK3FOWL and Joe VK3YSP

Radio Remote Control Server

Hardware

- Raspberry RPi2 or later
- 8GB SD card
- Modified USB Sound Card
- USB WiFi Adapter (not required for RPi3)
- Cables, Connectors and Ferrites
- 5V USB power supply
- Wemo Switch
- Home WiFi Router

Raspberry Pi



Linux USB Sound Cards

CMEDIA CM-108 7.1CH

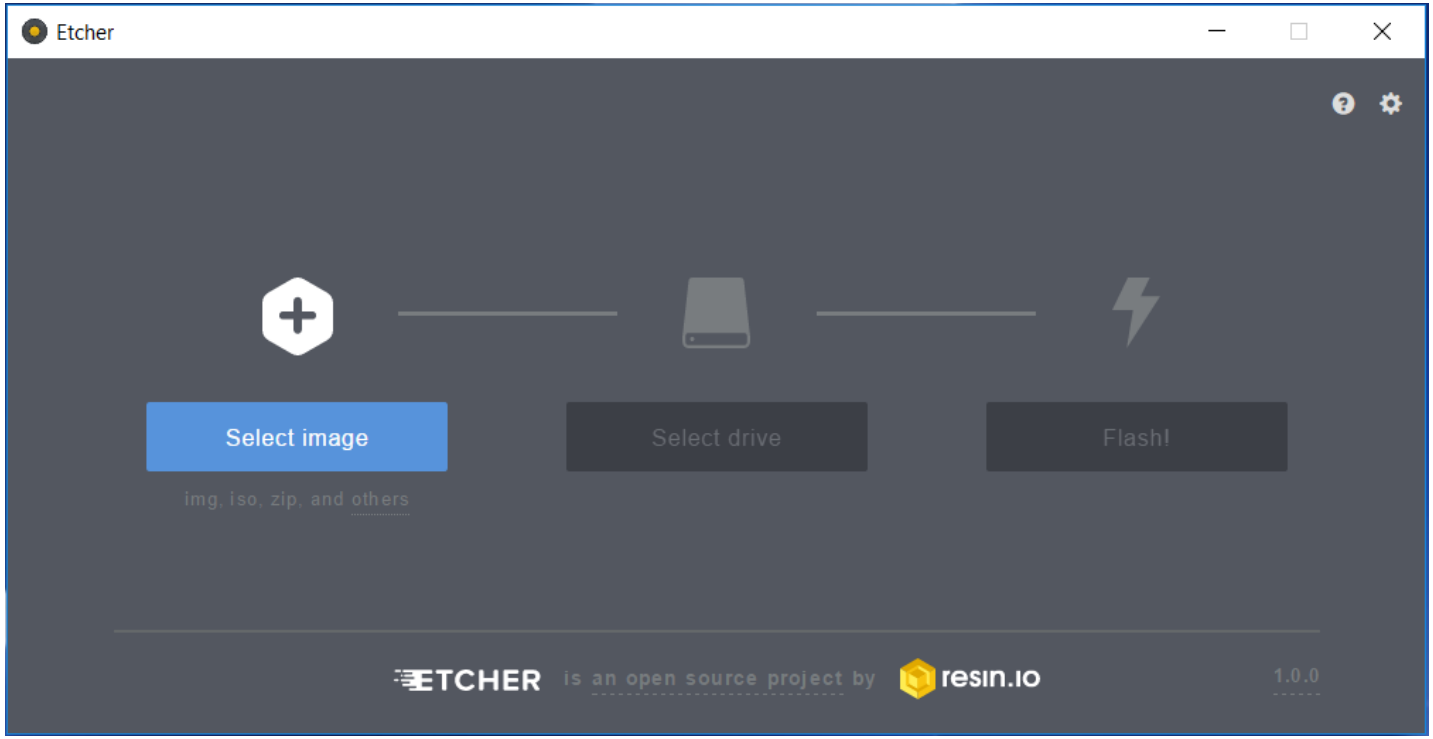
USB Sound Card Modifications

Wemo Switch

Software

- Could use IRLP and Echolink! But this is DIY:
- RPi Raspbian “Jessie” distro
- Echer SD card imaging tool
- Python dev library
- Python Radio Control Gateway “cgateway”
- Mumble VoIP Conference Server
- Mumble Client App
- Wemo App

Echer SD Card Imaging Tool



Requires an 8GB Micro SD card

Installing RPi Software

```
sudo raspi-config
sudo apt-get update
sudo apt-get -y upgrade
sudo apt-get -y install mumble
sudo apt-get -y install mumble-server
sudo dpkg-reconfigure mumble-server
set superuser password MySuperUserPassword
sudo nano /etc/mumble-server.ini
        serverpassword=MyServerPassword
        logfile=
sudo apt-get install python-dev
mkdir /home/pi/cgateway
put Joe's cgateway.py in /home/pi/cgateway

nano /home/pi/autostart.sh
        #! /bin/sh
        sleep 5
        sudo python3 /home/pi/cgateway/cgateway.py &
        sleep 2
        /usr/bin/mumble mumble://IC-7200: MyServerPassword@192.168.1.4 &

chmod 755 autostart.sh

nano /home/pi/.config/lxsession/LXDE-pi/autostart
        @lxpanel --profile LXDE-pi
        @pcmanfm --desktop --profile LXDE-pi
        @/home/pi/cgateway/autostart.sh
        @xscreensaver -no-splash
```

WiFi Router Setup

- RPi Must have a Fixed IP Address
- Open Port 80 for Python Web Server
- Open Port 64738 for Mumble VoIP Server

Enable DHCP Server

Start IP Address:	192.168.1.5
End IP Address:	192.168.1.254
Leased Time (hour):	24

Static IP Lease List: (A maximum 32 entries can be configured)

Edit DHCP Option | Edit DHCP Option 60 | DHCP Advance setup

MAC Address	IP Address	Remove
00:60:64:96:02:8B	192.168.1.1	☐
4c:34:88:d3:69:63	192.168.1.2	☐
84:1B:5E:96:D3:8E	192.168.1.3	☐
84:1B:5E:96:81:86	192.168.1.4	☐

Add Entries | Remove Entries

Server Name	External Port Start	External Port End	Protocol	Internal Port Start	Internal Port End	Server IP Address	WAN Interface	Remove
Web Server (HTTP)	80	80	TCP	80	80	192.168.1.4	ppp0	☐
Mumble	64738	64738	TCP/UDP	64738	64738	192.168.1.4	ppp0	☐

Dynamic DNS Setup

- Don’t have a real IP Address? Subscribe to a free DDNS.
- RPi sends its IP Address to DDNS on start-up
- Python script:

```
def ddnsUpdate():
   ipAddressServer='http://myip.dnsdynamic.org’
   userID='joe@gonzales.com.au:MyPassword’
   myip=subprocess.check_output(['curl',ipAddressServer]).decode()

ddnsServer='https://www.dnsdynamic.org/api/?hostname=joe.x64.me&myip='+myip
response=subprocess.check_output(['curl','u',userID,ddnsServer]).decode()
```

Python Web Server

```
Import http.server

class httpServer(http.server.BaseHTTPRequestHandler):

    def do_HEAD(self):
      self.send_response(200)
      self.send_header("Content-type", "text/html")
      self.end_headers()

    def do_GET(self):
     do_HEAD(self)
     #SEND AN HTML FORM IN RESPONSE TO A GET
     write(self,'<!DOCTYPE html>')
     write(self,'<html lang="en">')
     write(self,'<head>')
     write(self,'<meta content="text/html; charset=utf-8" http-equiv="content-type"/>')
     #ETC

    def do_POST(self):
      self.query_string = self.rfile.read(int(self.headers['Content-Length']))
      self.args = dict(cgi.parse_qsl(self.query_string))
      do_HEAD(self)
      usr={}
      for i in self.args:
          usr.update({i.decode('utf-8'):self.args[i].decode('utf-8')})
      #PROCESS THE FORM DATA THAT WAS POSTED BACK
```

```
#MAIN PROGRAM STARTS HERE

#UPDATE THE DDNS WITH YOUR IP ADDRESS
try:
    ddnsUpdate()
except:
    pass

#Start your web server
try:
    server = http.server.HTTPServer((lanAddr,80), httpServer)
    server.serve_forever()
except:
    printException()
finally:
    server.socket.close()
```

Transceiver Control Protocols

- Native, Proprietary Protocols:
  - YAESU CAT
  - Kenwood CAT
  - ICOM CI-V
- Open Source, Generic, Protocols:
  - HAMLIB

Another Approach: HAMLIB rigctl

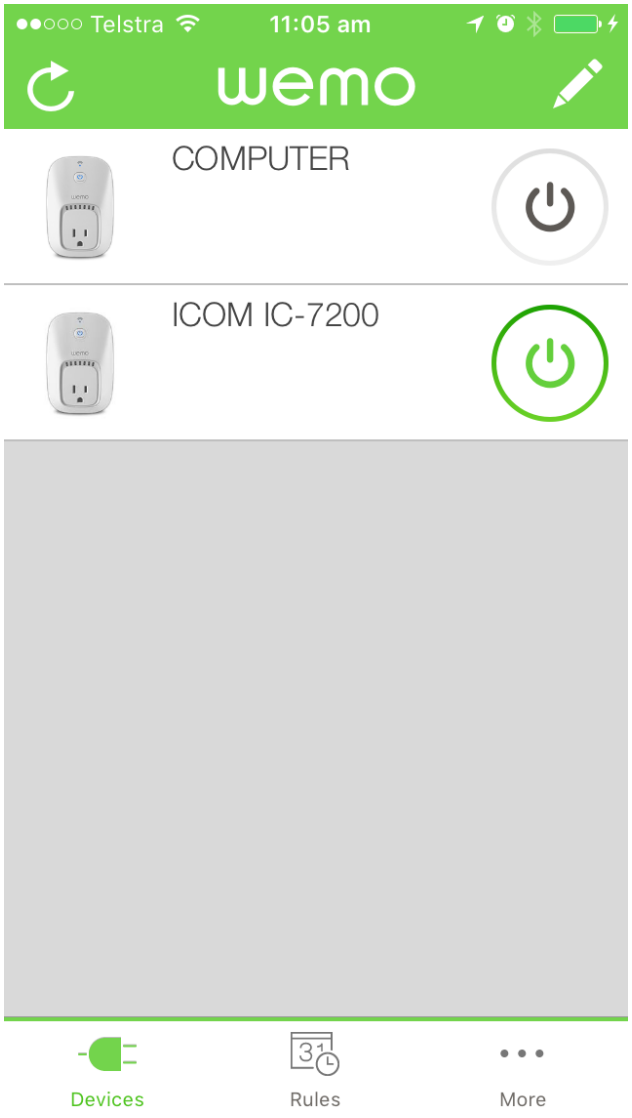
- **List rigctl rigs:**
  - "C:\Program Files (x86)\hamlib-w32-3.0\bin\rigctl.exe" –l
- **Start rigctl:**
  - "C:\Program Files (x86)\hamlib-w32-3.0\bin\rigctl.exe" -m361 -rCOM3 -s19200
- **List rigctl capabilities:**
  - Rig command: 1

    Caps dump for model: 361
    Model name: IC-7200
    Mfg name: Icom
    …
    Mode list: AM CW USB LSB RTTY CWR RTTYR PKTLSB PKTUSB
    VFO list: MEM VFOA VFOB
    VFO Ops: CPY XCHG FROM_VFO TO_VFO MCL TUNE
    Scan Ops: MEM SLCT PRIO PROG
    Filters:
    2.4 kHz: USB LSB
    1.8 kHz: USB LSB
    3 kHz: USB LSB
    500 Hz: CW RTTY CWR RTTYR
    250 Hz: CW RTTY CWR RTTYR
    1.2 kHz: CW CWR
    2.4 kHz: RTTY RTTYR
    6 kHz: AM
    3 kHz: AM
    9 kHz: AM
    …

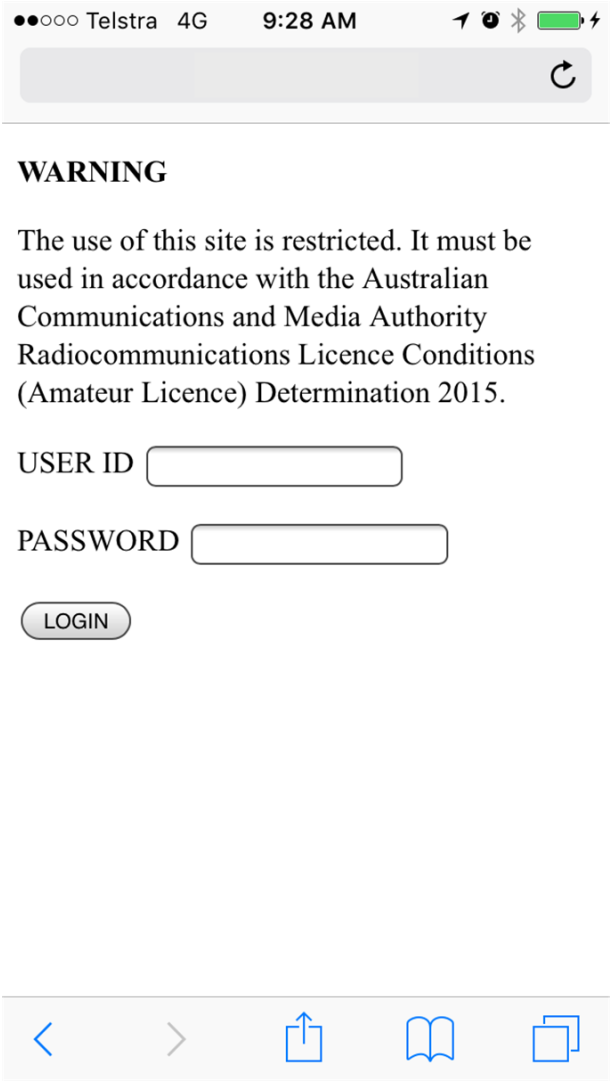
HAMLIB rigctl Session

C:\Users\joe>"C:\Program Files (x86)\hamlib-w32-3.0\bin\rigctl.exe" -m361 -rCOM3 -s19200 -v
Opened rig model 361, 'IC-7200’
Rig command: f
Frequency: 14076000
Rig command: F7100000
Frequency:
Rig command: f
Frequency: 7100000
Rig command: m
Mode: USB
Passband: 3600
Rig command: MLSB 2400
Mode: Passband:
Rig command: m
Mode: LSB
Passband: 2400

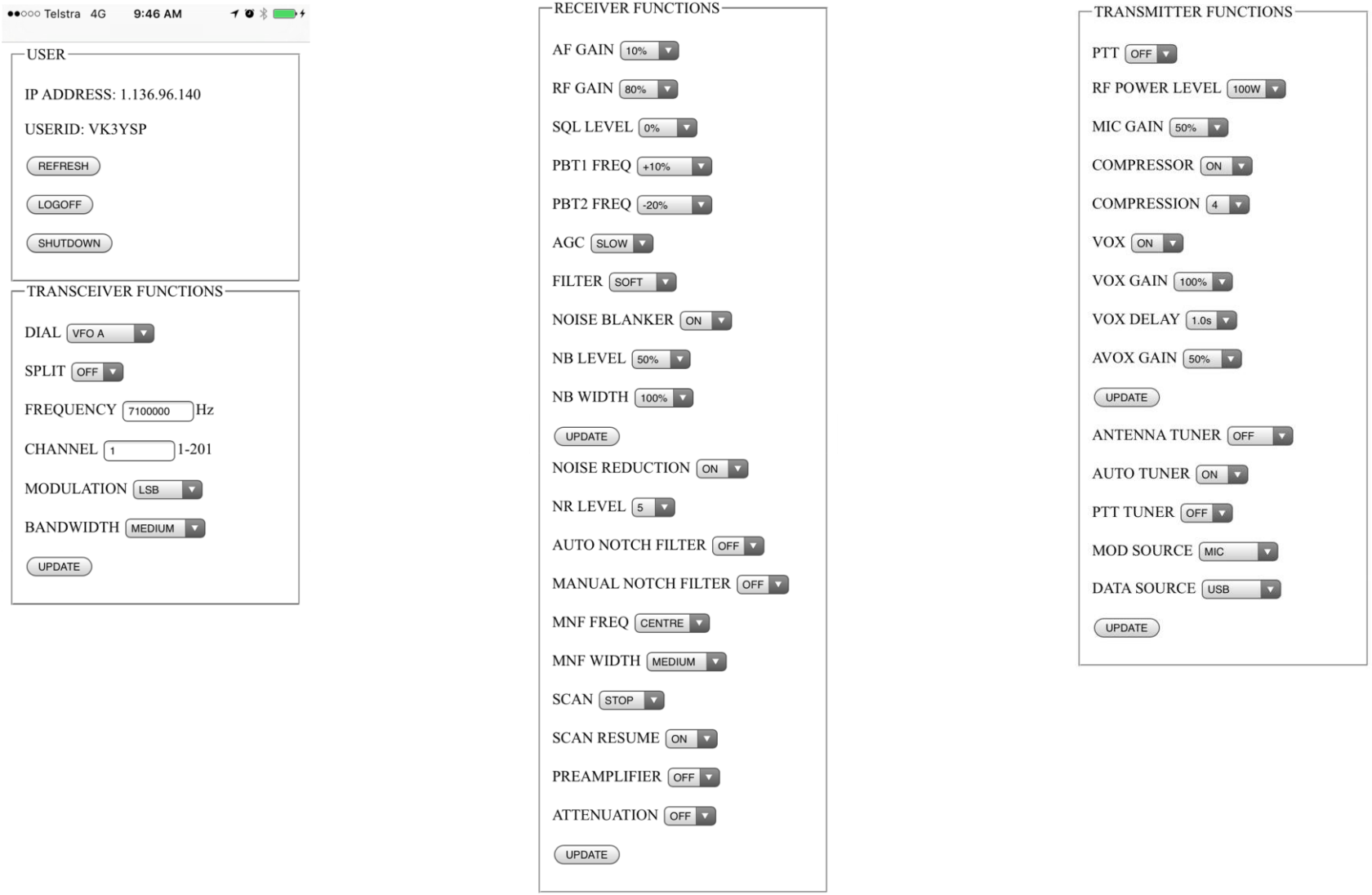
Wemo App


Telstra
11:05 am
wemo
COMPUTER
ICOM IC-7200
Devices
Rules
More

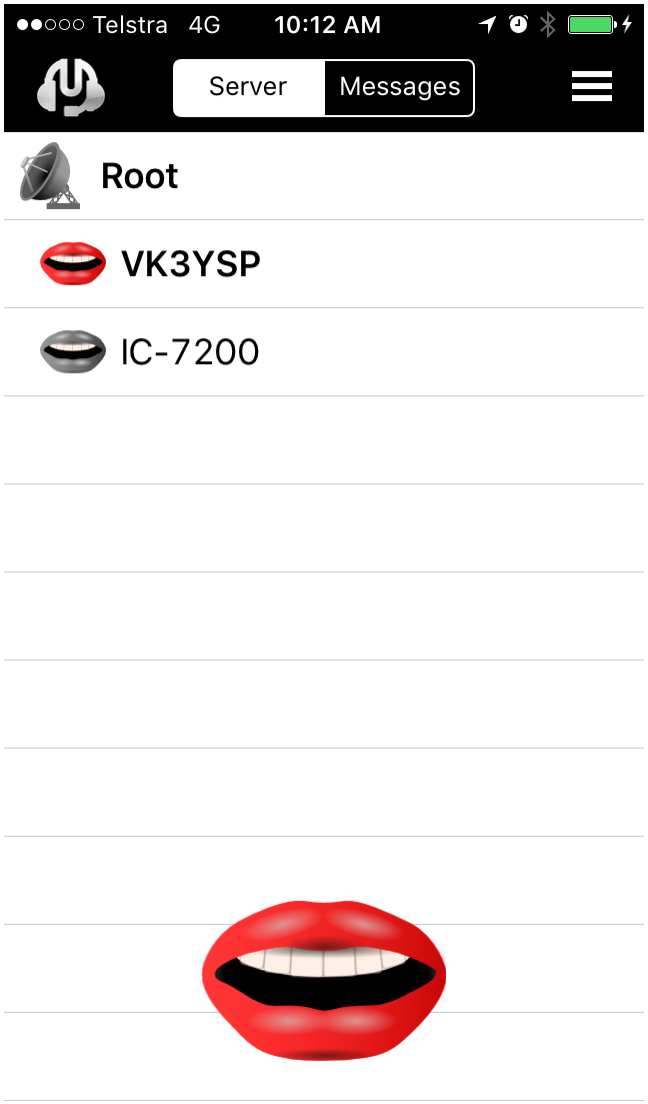
Login Page


Telstra 4G
9:28 AM
WARNING
The use of this site is restricted. It must be used in accordance with the Australian Communications and Media Authority Radiocommunications Licence Conditions (Amateur Licence) Determination 2015.
USER ID
PASSWORD
LOGIN

Remote Control Page



Mumble Client App



Links

- http://www.sarcnet.org/projects.html
- http://www.belkin.com/au/Products/home-automation/c/wemo-home-automation/
- https://wiki.mumble.info/wiki/Main_Page
- https://www.python.org/
- https://www.raspberrypi.org/downloads/raspbian/