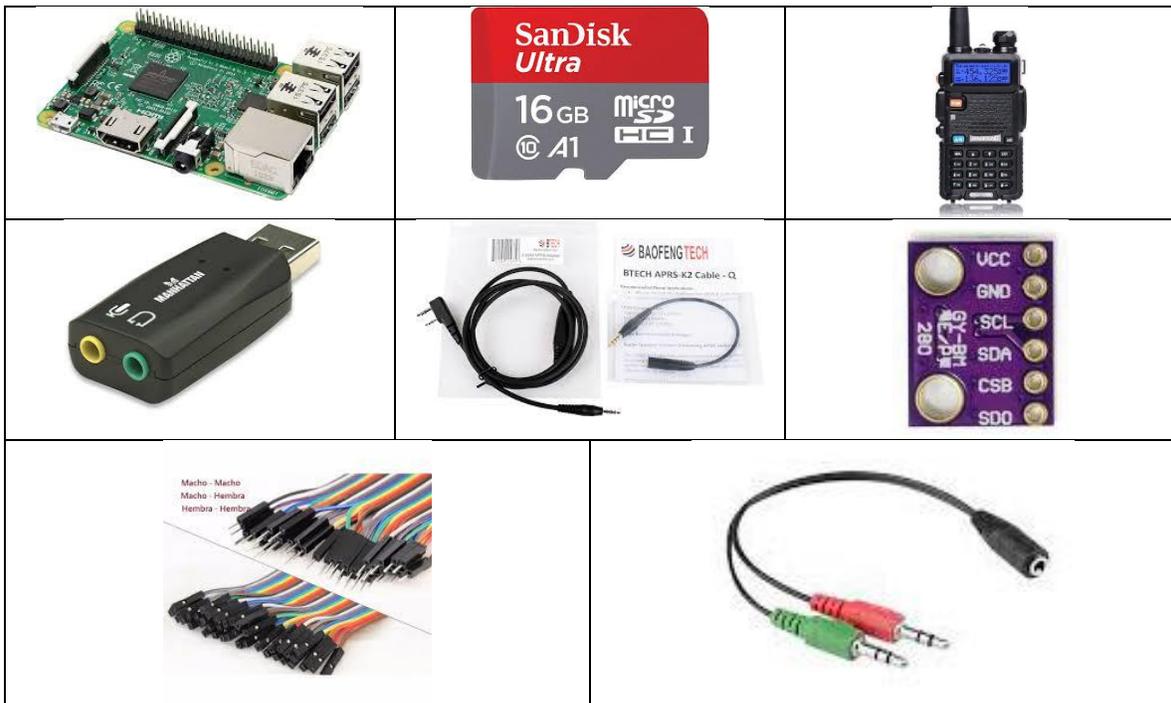


Estación Meteorológica en una Raspberry Pi, con Igate de APRS

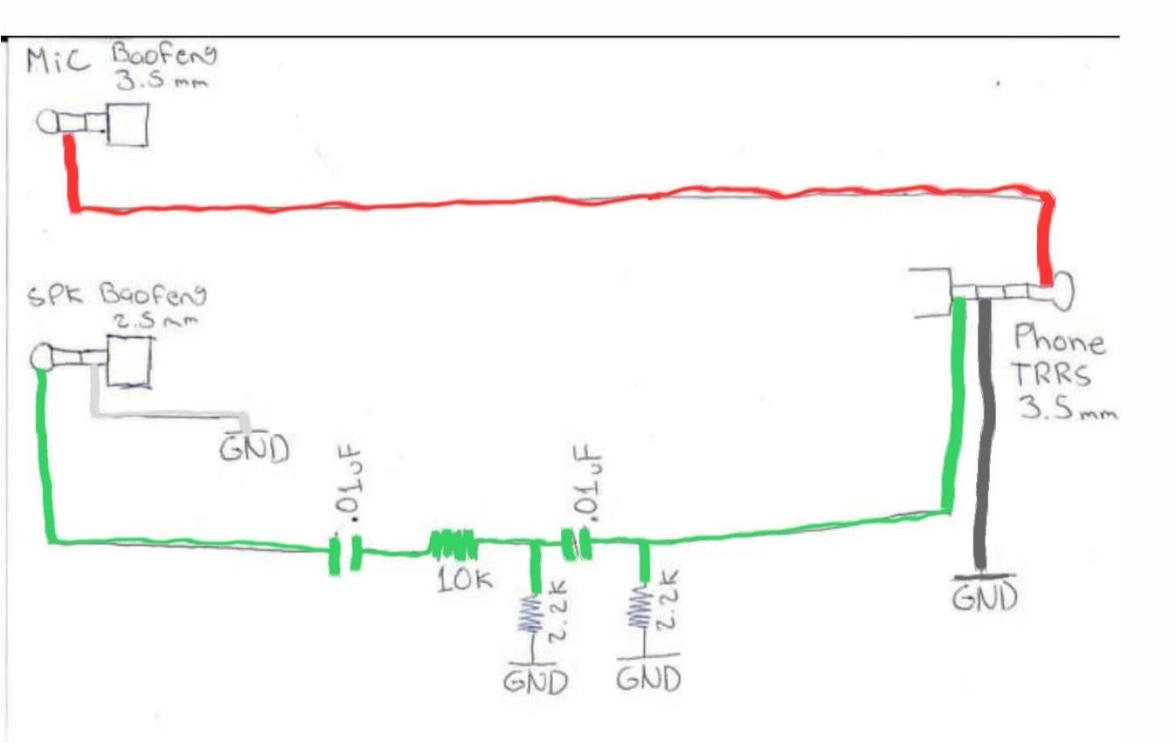
En el siguiente texto vamos a explicar la instalación, configuración de una Estación Meteorológica con Raspberry y el sensor BMP280, y en la misma Raspberry instalar Direwolf para tener un IGATE APRS conectada con un radio Baofeng.

Para empezar, vamos a requerir lo siguiente:

- Raspberry Pi3b
- Adaptador de Voltaje
- Memoria de 16gb MicroSD
- Imagen de Sistema Operativo Raspbian
- Radio Baofeng
- Tarjeta de Audio USB con entrada de micrófono y salida de audífonos
- Cable Adaptador para separar Audífono y Micrófono que venga de 1 solo PLUG a 2 Plugs Individuales
- Cable Btech-APRS-V01 (Mas Abajo pondré un Diagrama de cómo hacer el cable)
- Sensor GY-BMP280
- Cables para conectar de los que se usan en Arduino



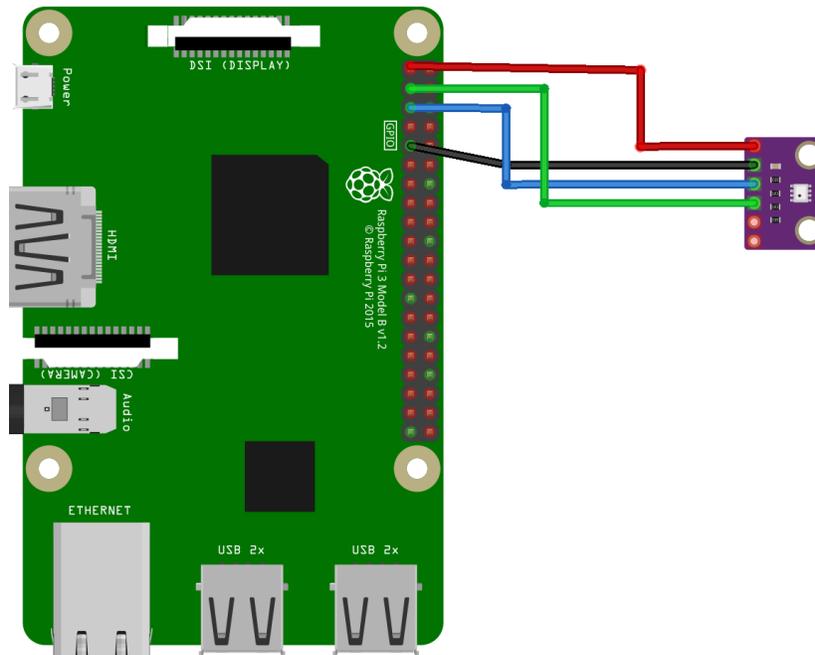
Gracias a **Ángel Abraham Mendoza Alférez XE1GLA**, compañero del **Club de Radio Amateur del Estado de Guanajuato**, que me ayudó con el diagrama de la interfaz de cable APRS del Radio Baofeng a la Raspberry; con esto se puede suplir el Cable Btech-APRS



Preparacion de Raspberry

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get dist-upgrade
sudo reboot
sudo raspi-config
    Simply navigate to no. 3 (Interfacing Options) and enable I2C.
Restart your Raspberry after doing this.
sudo reboot
sudo nano /boot/config.txt
    dtparam=i2c_arm=on
sudo reboot
sudo apt-get install git
git clone https://github.com/bastienwartz/Adafruit_Python_BMP/
cd Adafruit_Python_BMP/
sudo python setup.py install
sudo apt-get install python-smbus i2c-tools -y
```

Aquí vamos a apagar la Raspberry, para conectar el sensor como en el siguiente diagrama que se muestra a continuación, posteriormente vamos a prender la Raspberry y vamos a entrar a la siguiente ruta para crear el archivo en Python para probar nuestro Sensor BMP280 que esté funcionando.



fritzing

Raspberry	BMP280
Pin 1	VCC
Pin 3	GND
Pin 5	SCL
Pin 9	SDA

Ya que haya iniciado la Raspberry vamos a probar el siguiente código para comprobar que la Raspberry está leyendo el Sensor BMP280

```
sudo i2cdetect -y 1
```

y debemos de ver que aparezca el 76 como en la siguiente imagen

```
 0 1 2 3 4 5 6 7 8 9 a b c d e f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- 76 -- -- -- -- -- -- -- -- --
```

Si nos muestra el 76 significa que el sensor está funcionando, en caso de que no, debemos de volver a configurar tanto el cableado, como la configuración de la Raspberry habilitando el i2c.

Vamos a crear el archivo bmp280.py donde se encuentra el código para obtener la información del sensor y verla desplegada.

```
cd /home/pi/Adafruit_Python_BMP/
sudo nano bmp280.py
```

```
#!/usr/bin/python

import Adafruit_BMP.BMP280 as BMP280
sensor= BMP280.BMP280(address=0x76)

temp = int(sensor.read_temperature())
press = int(sensor.read_pressure()/100)

print (temp)
print (press)
```

Probamos el Código

```
sudo python bmp280.py
```

Vamos a recibir estos datos, la primera línea es la Temperatura y la siguiente la Presión Atmosférica

```
pi@raspberrypi:~ $ sudo python bmp280.py
34
818
pi@raspberrypi:~ $ █
```

Ahora vamos a crear el archivo en Python donde va a tomar los datos del sensor y los va a mandar a APRS, y vamos a programar que este archivo se ejecute cada 2 minutos. Antes de esto vamos a tener que crear un Indicativo (CallSign) para la Estación Meteorológica.

Paso 1:

Leer la información de esta página, debemos de solicitar un Indicativo, y posteriormente que ya estemos mandando la información, debemos de mandar un correo como lo especifica el Paso 2 dentro de esta página.

<http://www.findu.com/citizenweather/signup.html>

para hacer el registro debemos de entrar aquí y mandar la solicitud, y si todo está bien nos va a llegar un correo con nuestro Indicativo

http://www.findu.com/citizenweather/cw_form.html

Ya hecho esto, vamos a crear el Archivo en Python para que el sensor tome los datos, y los mande a APRS con nuestro nuevo indicativo, que solicitamos, podremos verlo en la página de:

<https://aprs.fi/>

[http://www.findu.com/cgi-bin/wxpage.cgi?call=\(Indicativo\)](http://www.findu.com/cgi-bin/wxpage.cgi?call=(Indicativo)) en mi caso, es esta página podrán ver el indicativo FW9275 que fue el que me dieron

<http://www.findu.com/cgi-bin/wxpage.cgi?call=FW9275>

El Archivo lo pueden descargar de
https://github.com/elihuzero/BMP280_APRS

```
cd /home/pi/Adafruit_Python_BMP/  
sudo nano sensor.py
```

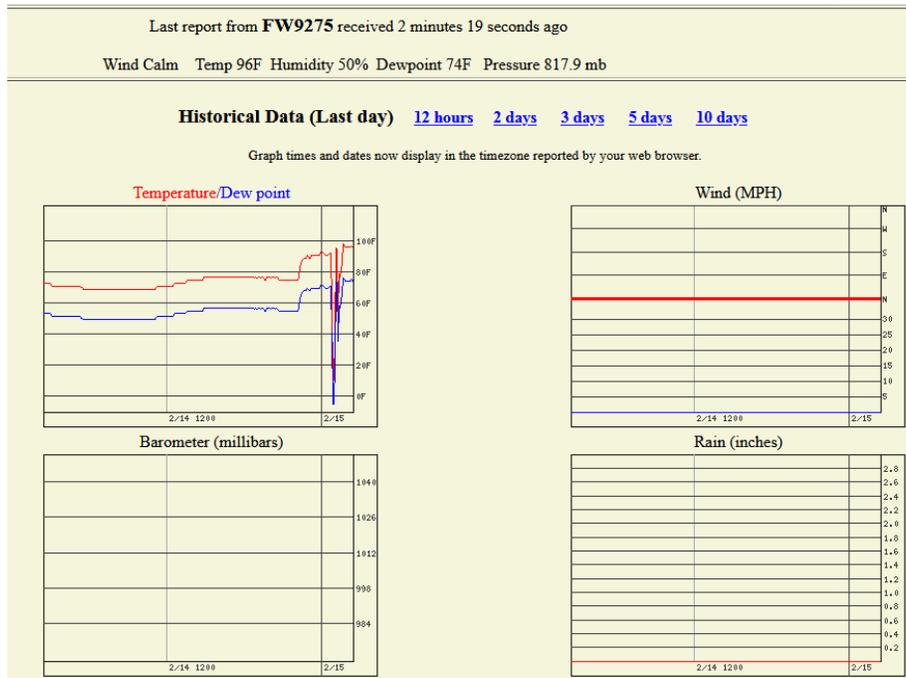
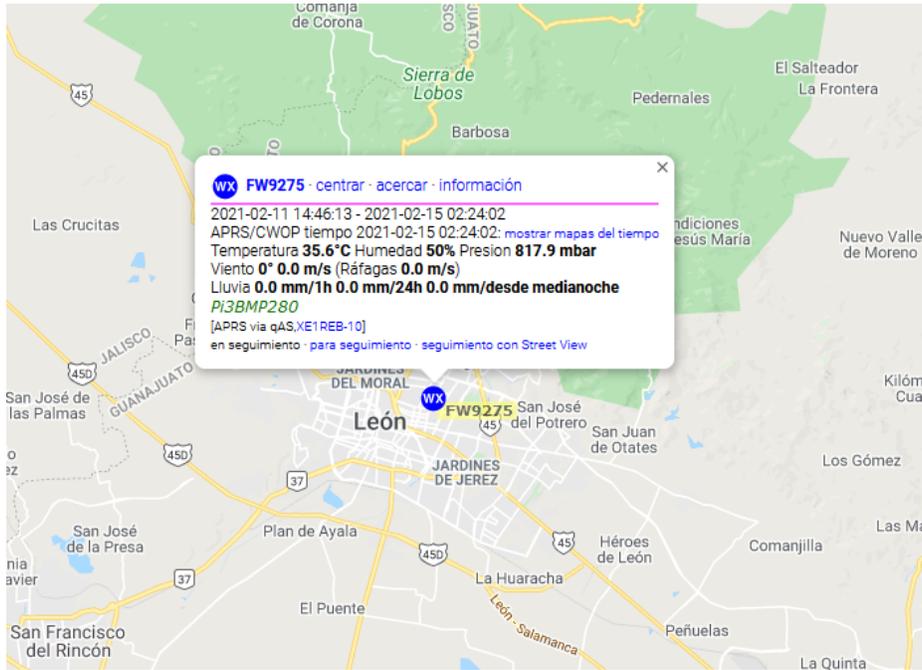
```
#!/usr/bin/python  
  
from socket import *  
import Adafruit_BMP.BMP280 as BMP280  
sensor= BMP280.BMP280(address=0x76)  
  
temp = int(sensor.read_temperature()*9/5+32)  
press = int(sensor.read_pressure()/10)  
  
# APRS-IS login info  
serverHost = 'mexico.aprs2.net'  
serverPort = 14580  
aprsUser = 'XE1REB-10'  
aprsPass = '24503'  
  
# APRS packet  
callsign = 'FW9275'  
btext1=("!2108.15N/10139.42W_000/000g000t0")  
btext3=("r000p000P000h50b0")  
btext5=("Pi3BMP280")  
btext = btext1+str(temp)+btext3+str(press)+btext5  
  
# create socket & connect to server  
sSock = socket(AF_INET, SOCK_STREAM)  
sSock.connect((serverHost, serverPort))  
# login  
sSock.send(('user %s pass %s vers Pi3BMP280\n' % (aprsUser, aprsPass)  
).encode("utf-8"))  
# send packet  
sSock.send(('>APRS:%s\n' % (callsign, btext) ).encode("utf-8"))  
# close socket  
sSock.shutdown(0)  
sSock.close()
```

NOTA: Se debe de modificar el indicativo del archivo de configuración

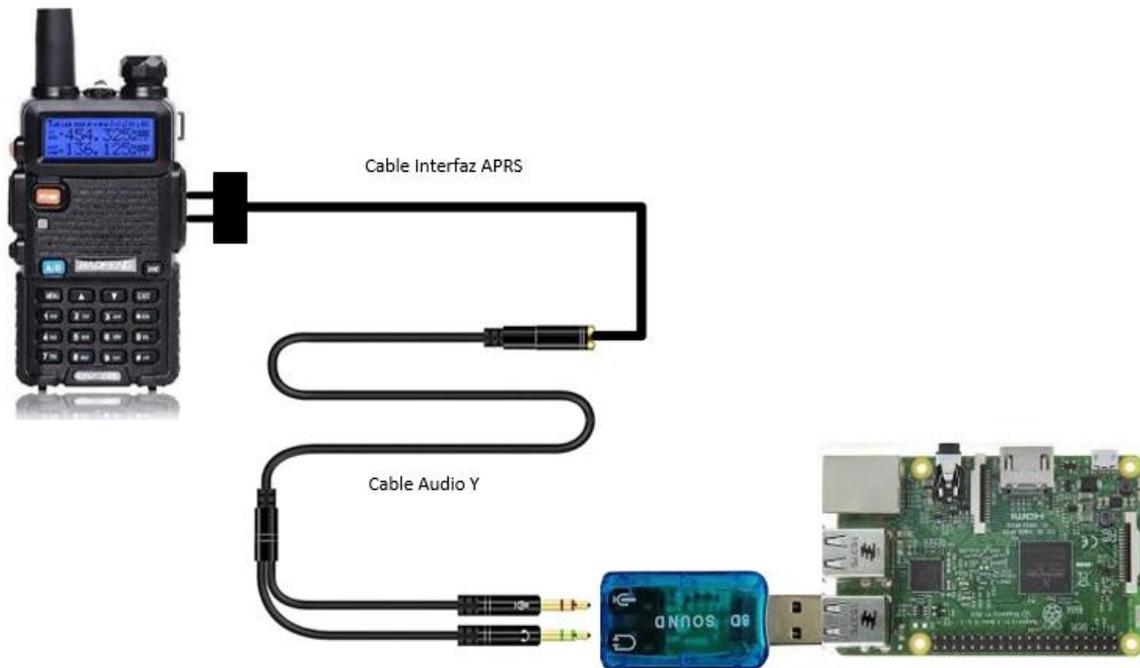
Lo siguiente es para que el archivo que acabamos de crear se ejecute desde que inicie la raspberry, y se ejecute automáticamente cada 2 minutos y este mandando la información a APRS,

```
sudo chmod a+x sensor.py  
sudo crontab -e  
*/2 * * * * /usr/bin/python /home/pi/Adafruit_Python_BMP/sensor.py >  
/tmp/sensor.log 2>&1  
sudo reboot
```

Ya ejecutando nuestro Script lo podemos ver así:



Con esto ya tenemos montada la Raspberry, la Librería Adafruit_Python_BMP, e instalado el sensor y funcionando, a continuación, vamos a hacer la instalación de DIREWOLF para hacer funcionar el iGate APRS, y posteriormente vamos a ejecutar el Script para que el Sensor mande la información hacia APRS y la veamos desplegada tanto en la página de findu.com y aprs.fi como una Estación Meteorológica. **(NOTA: debemos de tener conectada la Raspberry a Internet en todo momento)**



Instalacion de Librerias para Direwolf y deshabilitar el PulseAudio

```
sudo apt-get remove --purge pulseaudio
sudo apt-get autoremove
rm -rf /home/pi/.pulse
sudo reboot
sudo apt-get install cmake
sudo apt-get install libasound2-dev
sudo apt-get install libudev-dev
```

Instalacion de Direwolf

```
cd /home/pi/
git clone https://www.github.com/wb2osz/direwolf
cd direwolf
git checkout dev
mkdir build && cd build
cmake ..
make -j4
sudo make install
make install-conf
```

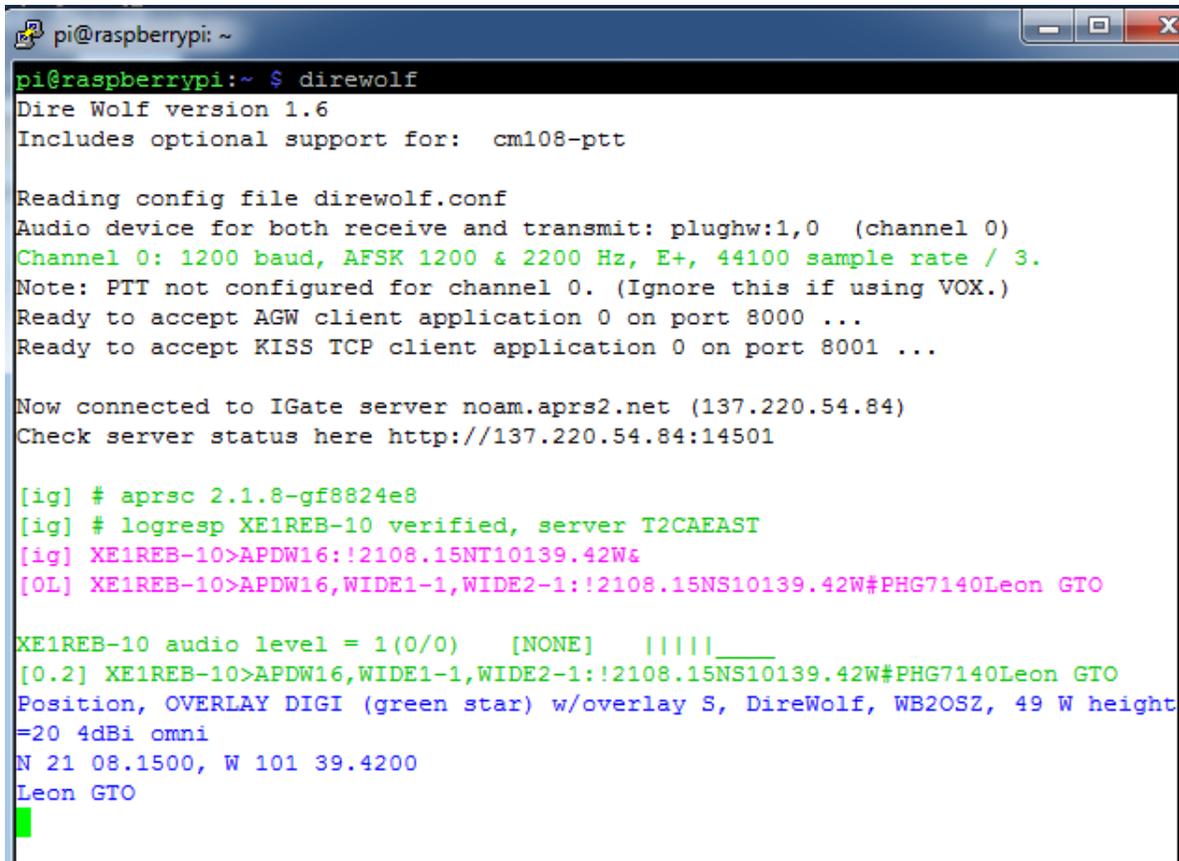
Bajar el archivo de configuracion y copiarlo dentro de la carpeta /home/pi/

https://github.com/elihuzero/Direwolf_conf.git

NOTA: Se debe de modificar el indicativo del archivo de configuración

Ya terminado esto vamos a ejecutar Direwolf, y vamos a ver la siguiente pantalla, y de igual forma lo vamos a ver reflejado en la página de <https://aprs.fi> con nuestro indicativo que le asignamos al iGate

direwolf



```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ direwolf  
Dire Wolf version 1.6  
Includes optional support for:  cm108-ptt  
  
Reading config file direwolf.conf  
Audio device for both receive and transmit: plughw:1,0 (channel 0)  
Channel 0: 1200 baud, AFSK 1200 & 2200 Hz, E+, 44100 sample rate / 3.  
Note: PTT not configured for channel 0. (Ignore this if using VOX.)  
Ready to accept AGW client application 0 on port 8000 ...  
Ready to accept KISS TCP client application 0 on port 8001 ...  
  
Now connected to IGate server noam.aprs2.net (137.220.54.84)  
Check server status here http://137.220.54.84:14501  
  
[ig] # aprsc 2.1.8-gf8824e8  
[ig] # logresp XE1REB-10 verified, server T2CAEAST  
[ig] XE1REB-10>APDW16:!2108.15NT10139.42W&  
[OL] XE1REB-10>APDW16,WIDE1-1,WIDE2-1:!2108.15NS10139.42W#PHG7140Leon GTO  
  
XE1REB-10 audio level = 1(0/0) [NONE] |||||_____  
[0.2] XE1REB-10>APDW16,WIDE1-1,WIDE2-1:!2108.15NS10139.42W#PHG7140Leon GTO  
Position, OVERLAY DIGI (green star) w/overlay S, DireWolf, WB2OSZ, 49 W height  
=20 4dBi omni  
N 21 08.1500, W 101 39.4200  
Leon GTO  
█
```

Y podemos ver que nuestra iGate ya está funcionando en la página de <https://aprs.fi> buscándolo por el indicativo que le asignamos en el archivo de configuración.

