

COMO AÑADIR CLIENTES EN HBLINK.CFG Y REGLAS EN CONFBRIDGE RULES.PY

Nos bajamos los siguientes lanzadores para que nos sea más cómodo editar los ficheros necesarios, en este caso tendremos que editar el fichero hblink.cfg y hb_confbridge_rules.py

Abrimos una terminal y copiamos la siguiente línea:

```
git clone https://github.com/ea4gax/Lanzadores\_Hblink
```

Por defecto nos lo bajará en /home/pi en una carpeta llamada Lanzadores_Hblink

Por defecto si hemos instalado el HBLINK en /home/pi no tendremos que editar los dos lanzadores que vamos a utilizar, si fuera necesario los editaríamos y la línea a cambiar sería: Exec=sh -c 'cd y poner la ruta correcta, entonces abriríamos una terminal en la carpeta Lanzadores_Hblink y editaríamos el fichero con el comando sudo nano nombre_del_lanzador.

Es opcional, podremos editarlos de la manera tradicional desde terminal:

```
cd Hblink
sudo nano hblink.cfg
sudo nano hb_confbridge_rules.py
```

El primer fichero a editar sería el hblink.cfg, damos un doble click en el icono llamado "Editar hblink", se nos abrirá el fichero hblink.cfg y nos vamos abajo del todo.

Pongamos que queremos añadir el 4374 de DMR+, añadimos las siguientes líneas:

```
[DMR+4374]
MODE: PEER
ENABLED: True
LOOSE: False
EXPORT_AMBE: False
IP:
PORT: 54002
MASTER_IP: eamaster04.xrefletor.es
MASTER_PORT: 55555
PASSPHRASE: PASSWORD
CALLSIGN: TU QRZ
RADIO_ID: TU ID
RX_FREQ: 430000000
TX_FREQ: 430000000
TX_POWER: 25
COLORCODE: 1
SLOTS: 2
LATITUDE: 38.0000
LONGITUDE: -095.0000
HEIGHT: 75
LOCATION: Madrid, SPAIN
DESCRIPTION: This is a cool repeater
URL: www.wlabc.org
SOFTWARE_ID: 20170620
PACKAGE_ID: MMDVM_Hblink
GROUP_HANGTIME: 5
OPTIONS: StartRef=4374;Relinktime=10
USE_ACL: True
REG_ACL: PERMIT:ALL
SUB_ACL: DENY:1
TGID_TS1_ACL: PERMIT:ALL
TGID_TS2_ACL: PERMIT:ALL
```

Nota aclaratoria: Es importante el StartRef=4374;Relinktime=10 ya que si no ponemos esa línea se nos conectará al servidor pero no al reflector.

Pondremos nuestro indicativo, ID y luego si queremos podremos editar la frecuencia, potencia, coordenadas, url...

AÑADIR CLIENTE EN BRANDMEISTER:

Se añade igual, solo que hay que poner las credenciales del mismo, pongamos el ejemplo de añadir el Multiprotocolo 21461

```
[BM21461]
MODE: PEER
ENABLED: True
LOOSE: False
EXPORT_AMBE: False
IP:
PORT: 54003
MASTER_IP: 84.232.5.113
MASTER_PORT: 62031
PASSPHRASE: passw0rd
CALLSIGN: TU QRZ
RADIO_ID: TU ID
RX_FREQ: 430000000
TX_FREQ: 430000000
TX_POWER: 25
COLORCODE: 1
SLOTS: 2
LATITUDE: 38.0000
LONGITUDE: -095.0000
HEIGHT: 75
LOCATION: Madrid, SPAIN
DESCRIPTION: This is a cool repeater
URL: www.wlabc.org
SOFTWARE_ID: 20170620
PACKAGE_ID: MMDVM_HBlink
GROUP_HANGTIME: 5
OPTIONS:
USE_ACL: True
REG_ACL: PERMIT:ALL
SUB_ACL: DENY:1
TGID_TS1_ACL: PERMIT:ALL
TGID_TS2_ACL: PERMIT:ALL
```

Nota aclaratoria: He cambiado la IP y el puerto que usa Bradmeister y he añadido un puerto más en la línea "PORT: 54003", (Siempre que añadimos más clientes hay que poner un puerto más en la línea "PORT:") Pondremos nuestro indicativo, ID y luego si queremos podremos editar la frecuencia, potencia, coordenadas, url...

Ahora debemos figar el TG, para ello nos vamos a la web <https://brandmeister.network/>, nos logeamos y nos vamos al apartado "My hostspot" ahí debemos fijar el TG en la id que hemos elegido.

Ahora toca editar el fichero hb_confbridge_rules.py, damos un doble click en el icono llamado Editar hb_confbridge_rules, se nos abrirá el mismo, si tenemos añadido el loro tendrá este aspecto:

'''

THIS EXAMPLE WILL NOT WORK AS IT IS - YOU MUST SPECIFY YOUR OWN VALUES!!!

This file is organized around the "Conference Bridges" that you wish to use. If you're a c-Bridge person, think of these as "bridge groups". You might also liken them to a "reflector". If a particular system is "ACTIVE" on a particular conference bridge, any traffic from that system will be sent to any other system that is active on the bridge as well. This is not an "end to end" method, because each system must independently be activated on the bridge.

The first level (e.g. "WORLDWIDE" or "STATEWIDE" in the examples) is the name of the conference bridge. This is any arbitrary ASCII text string you want to use. Under each conference bridge definition are the following items -- one line for each HBSysystem as defined in the main Hblink configuration file.

- * SYSTEM - The name of the system as listed in the main hblink configuration file (e.g. hblink.cfg)
 - This MUST be the exact same name as in the main config file!!!
- * TS - Timeslot used for matching traffic to this conference bridge
- * TGID - Talkgroup ID used for matching traffic to this conference bridge
- * ON and OFF are LISTS of Talkgroup IDs used to trigger this system off and on. Even if you only want one (as shown in the ON example), it has to be in list format. None can be handled with an empty list, such as " 'ON': [] ".
- * TO_TYPE is timeout type. If you want to use timers, ON means when it's turned on, it will turn off after the timeout period and OFF means it will turn back on after the timeout period. If you don't want to use timers, set it to anything else, but 'NONE' might be a good value for documentation!
- * TIMEOUT is a value in minutes for the timeout timer. No, I won't make it 'seconds', so don't ask. Timers are performance "expense".
- * RESET is a list of Talkgroup IDs that, in addition to the ON and OFF lists will cause a running timer to be reset. This is useful if you are using different TGIDs for voice traffic than triggering. If you are not, there is NO NEED to use this feature.

'''

```
BRIDGES = {
    'LORITO': [
        {'SYSTEM': 'MASTER-1', 'TS': 2, 'TGID': 9990, 'ACTIVE': True,
        'TIMEOUT': 2, 'TO_TYPE': 'NONE', 'ON': [], 'OFF': [], 'RESET': []},
        {'SYSTEM': 'LORO', 'TS': 2, 'TGID': 9990, 'ACTIVE': True,
        'TIMEOUT': 2, 'TO_TYPE': 'NONE', 'ON': [], 'OFF': [], 'RESET': []},
    ]
}

if __name__ == '__main__':
    from pprint import pprint
    pprint(BRIDGES)
```

Si no tenemos el loro se añadirán las reglas debajo de las últimas 3 '''

Añadiremos las dos reglas necesarias para el DMR+4374 y BM21461.
Debajo de esto añadiremos las dos reglas necesarias, quedaría así:

```

BRIDGES = {
    'LORITO': [
        {'SYSTEM': 'MASTER-1', 'TS': 2, 'TGID': 9990, 'ACTIVE': True,
        'TIMEOUT': 2, 'TO_TYPE': 'NONE', 'ON': [], 'OFF': [], 'RESET': []},
        {'SYSTEM': 'LORO', 'TS': 2, 'TGID': 9990, 'ACTIVE': True,
        'TIMEOUT': 2, 'TO_TYPE': 'NONE', 'ON': [], 'OFF': [], 'RESET': []},
    ],

    'DMR+4374': [
        {'SYSTEM': 'MASTER-1', 'TS': 2, 'TGID': 4374, 'ACTIVE': True,
        'TIMEOUT': 10, 'TO_TYPE': 'ON', 'ON': [4374], 'OFF': [4], 'RESET': []},
        {'SYSTEM': 'DMR+4374', 'TS': 2, 'TGID': 9, 'ACTIVE': True,
        'TIMEOUT': 2, 'TO_TYPE': 'NONE', 'ON': [], 'OFF': [], 'RESET': []},
    ],

    'BM21461': [
        {'SYSTEM': 'MASTER-1', 'TS': 1, 'TGID': 21461, 'ACTIVE': True,
        'TIMEOUT': 10, 'TO_TYPE': 'NONE', 'ON': [21461], 'OFF': [4], 'RESET': []},
        {'SYSTEM': 'BM21461', 'TS': 1, 'TGID': 21461, 'ACTIVE': True,
        'TIMEOUT': 2, 'TO_TYPE': 'NONE', 'ON': [], 'OFF': [], 'RESET': []},
    ]
}

```

Nota aclaratoria: He puesto un tiempo de desconexión de 10 minutos, podemos desactivarlo (en el caso del loro esta fijo, no se desconecta), si queremos que no se desconecte el TG hacerlo igual que en su caso. Siempre que añadamos una regla hay que poner una , después del último] de la misma, en la última regla no hace falta ya que cerramos con un }.

Si nos fijamos el nombre del cliente (en el hbblink.cfg) y de la regla (en el hb_confbridge_rules.py) es igual, eso siempre tiene que coincidir junto con el nombre del master (en este caso MASTER-1).

Una vez hecho todo esto solo tenemos que arrancar el servidor y el monitor con los iconos "Ejecutar HBLINK" y "Monitor HBLINK" Respectivamente, si hemos instalado el loro el icono para arrancarlo es "Ejecutar LORO" luego para parar todo podemos ejecutar el icono "Parar HBLINK".

EA4GAX - Sergio