

Activar servicios Audi Connect en MMI 3G+ con Raspberry Pi - v1.0

El presente manual tiene como objetivo configurar una Raspberry Pi para que el navegador MMI 3G+ pueda salir a internet y acceder a los servicios de Audi Connect, aparte de poder navegar con los mapas de Google Earth, StreetView e incluso representar el tráfico on-line de Google. Además, como suelo comentar siempre, éste manual es una referencia y puede no coincidir con lo que tenéis en vuestros coches y por supuesto no me hago responsable de nada de lo que le hagáis a los mismos al seguir estas instrucciones.

Agradecimientos

Quiero agradecer personalmente a Hispano73 de la a4-klub de Polonia por su colaboración y a su amigo Yoghurt por averiguar el script definitivo para dejar permanentes los DNS y el gateway por defecto en el MMI.

Materiales necesarios

- Raspberry Pi Model B+ (Vale cualquiera teniendo un mínimo de dos puertos USB)
- Tarjeta MicroSD 4GB o 8GB para el Sistema Operativo de la Raspberry
- Cargador de coche para la Raspberry Pi con 2000 mA de potencia

- Carcasa para la Raspberry Pi (a gusto del consumidor)
- Tarjeta SD 2 GB o 4 GB para el script que hay que pasarle al MMI.
- 2 Adaptadores WiFi nano USB. Os recomiendo el Edimax EW-7811UN. Puede valer cualquiera pero éste que os recomiendo lo reconoce el S.O. sin tener que hacer nada adicional.

Concepto

La idea es aprovechar las capacidades de la Raspberry Pi para que haga de router entre los dos interfaces de red inalámbricos (wlan0 conectado al hotspot del MMI y wlan1 conectado al del móvil) y así el MMI pueda salir a internet para buscar mapas, datos del tráfico, etc. Para ello habrá que instalar un sistema operativo y configurarlo. Yo he elegido Raspbian, pero podéis elegir cualquiera aunque lo mismo los comandos cambian.

Procedimiento 1. Raspberry

Tenéis que instalar el sistema operativo Raspbian en la tarjeta MicroSD, siguiendo las siguientes instrucciones:

<https://www.raspberrypi.org/documentation/installation/installing-images/README.md>

Una vez instalado, montáis la tarjeta en el cacharrito, le ponéis los adaptadores WiFi nano y lo conectáis por HDMI a la TV, aparte de ponerle teclado y/o ratón para las configuraciones. Arrancáis la Raspberry y hacéis la configuración inicial que os sale, metiendo una contraseña para el usuario "pi" y diciéndole que coja todo el disco completo. Reiniciáis y procedéis con la configuración. Doy por hecho que el sistema ha detectado perfectamente ambos dongles USB, lo cual podéis ver al ejecutar el comando (Sin el \$):

```
$ ip addr
```

Os debe de salir la configuración de IP de los distintos adaptadores, entre ellos el de wlan0 y wlan1.

Entonces empezamos:

1. Activar el forwarding ejecutando el comando:

```
$ sudo echo 1 > /proc/sys/net/ipv4/ip_forward
```

2. Editar el archivo /etc/sysctl.conf y descomentar la línea que pone:

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

3.- Reiniciar la Raspberry Pi con:

```
$ sudo init 6
```

4.- Meter las reglas del cortafuegos necesarias para el funcionamiento del router (Raspberry Pi), ejecutando (esto os sirve para probar pero se pierde al reiniciar):

```
$ iptables -t nat -A POSTROUTING -o wlan1 -j MASQUERADE
```

5.- Esta regla es volátil, por lo que hay que hacerla permanente para que no se pierda en cada arranque. Para ello hay que crear un archivo script en /etc/init.d/ con el nombre que queráis, por ejemplo script-cortafuegos.sh:

```
$ sudo vi /etc/init.d/script-cortafuegos.sh
```

Y le dáis permisos con:

```
$ sudo chmod 755 /etc/init.d/script-cortafuegos.sh
```

Como ejemplo del contenido del script para el cortafuegos, os dejo el mío:

```
#!/bin/sh
```

```
# Script para hacer permanentes las reglas del cortafuegos en cada  
arranque
```

```
iptables -t nat -A POSTROUTING -o wlan1 -j MASQUERADE
```

Posteriormente ejecutáis éstos dos comandos, el segundo es para aplicar la regla y no hace falta si lo habéis ejecutado anteriormente:

```
$ sudo update-rc.d script-cortafuegos.sh defaults
```

```
$ sudo sh /etc/init.d/script-cortafuegos.sh
```

6.- Reiniciar la Raspberry Pi.

En éste punto la raspberry si se conecta al Wi-Fi del MMI por la wlan0 y al del móvil ya debe de ser capaz de rutar los paquetes entre los interfaces wlan0 y wlan1. Siempre y cuando hayamos configurado ambos interfaces. Si no es así, os pongo mi archivo de configuración de la red para que tengáis una idea. Si os fijáis en la configuración del interface wlan0 he puesto una IP estática para luego configurarla como gateway en el MMI (192.168.1.254), tiene que ser la misma... ;)

Contenido de /etc/network/interfaces:

www.audisport-iberica.com by Mad-i, Hispano73 y Yoghurt


```
auto lo
```

```
iface lo inet loopback
```

```
#iface default inet dhcp
```

```
# ----eth0----
```

```
auto eth0
```

```
# ----wlan0----
```

```
# Interface para conectar al MMI
```

```
#allow-hotplug wlan0
```

```
#iface wlan0 inet manual
```

```
#wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
```

```
auto wlan0
```

```
iface wlan0 inet static
```

```
address 192.168.1.254
```

```
netmask 255.255.255.0
```

network 192.168.1.0

broadcast 192.168.1.255

wpa-ssid Audi99999

wpa-psk XXXXXXXXXXXXXXXX

----wlan1----

Interface para conectar al telefono movil

*# La IP de abajo, 172.20.10.1 es la del gateway, es decir el telefono. Podeis
comprobarla conectando el portatil a la wifi del movil*

auto wlan1

iface wlan1 inet dhcp

wpa-ssid Nombre_Telefono

wpa-psk XXXXXXXXXX

*# Estas ordenes son para cambiar el gateway por defecto y poner el del
telefono*

post-up route add default gw 172.20.10.1

post-up route del default gw 192.168.1.1

pre-down route del default gw 172.20.10.1

```
pre-down route add default gw 192.168.1.1
```

A parte, está la configuración de wpa_supplicant, que a priori no es necesario pero por si acaso.

A continuación el contenido del archivo
/etc/wpa_supplicant/wpa_supplicant.conf

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
```

```
update_config=1
```

```
network={
```

```
    ssid="Nombre_Red_Wifi_Telefono"
```

```
    psk="XXXXXXXXXXXX"
```

```
    proto=RSN
```

```
    key_mgmt=WPA-PSK
```

```
    pairwise=CCMP
```

```
    auth_alg=OPEN
```

```
}
```

```
network={
```

```
    ssid="Audi99999"
```



```
psk="XXXXXXXXXX"  
  
proto=RSN  
  
key_mgmt=WPA-PSK  
  
pairwise=CCMP  
  
auth_alg=OPEN  
  
}
```

Donde pone XXXXXXXXX, hay que poner las correspondientes claves WiFi... ;)

En teoría tocando el archivo /etc/network/interfaces y añadiendo aquí redes WiFi, el cacharro es capaz de conectarse a distintas redes WiFi, por ejemplo si tenemos varios teléfonos...

Procedimiento 2. MMI 3G+

Una vez realizados todos los pasos, ahora hay que configurar el MMI para que pueda funcionar el invento. Para ello tenéis que tener activado el menú verde de ingeniería del MMI, lo cual podéis hacer de la siguiente manera con el VagCom o VCDS:

Módulo 5F

Adaptation

Canal 6 - Poner a 1

También activamos el tráfico on-line en el mismo módulo:

Módulo 5F

Adaptation

Canal 69

Incrementar el valor actual en 32768.

Ejemplo: El valor actual es 15. $15 + 32768 = 32783$

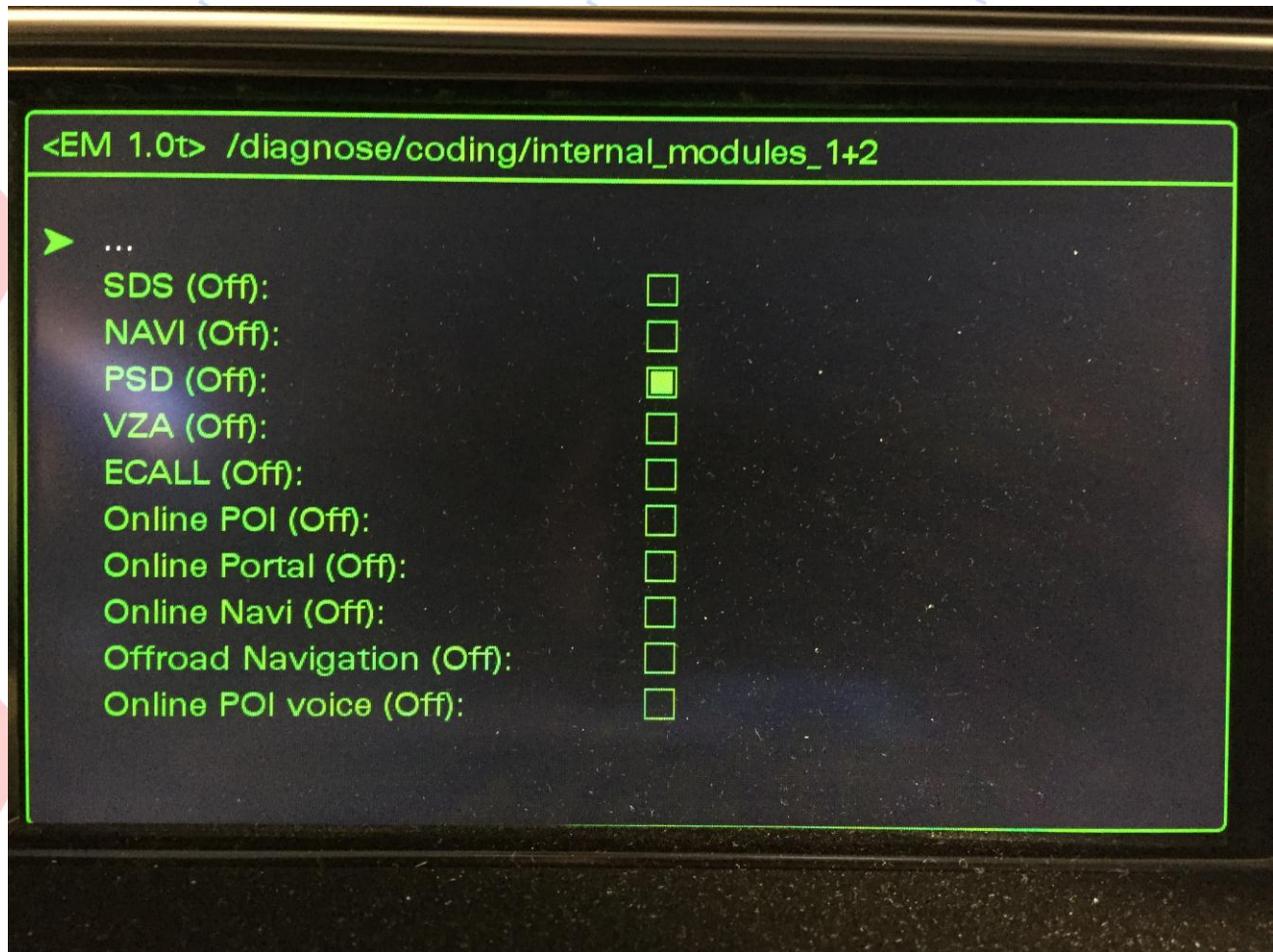
Nuevo valor para el canal 69: 32783

Reiniciar MMI

Una vez reiniciado podéis entrar pulsando CAR+MENU varios segundos.

Entonces procedemos:

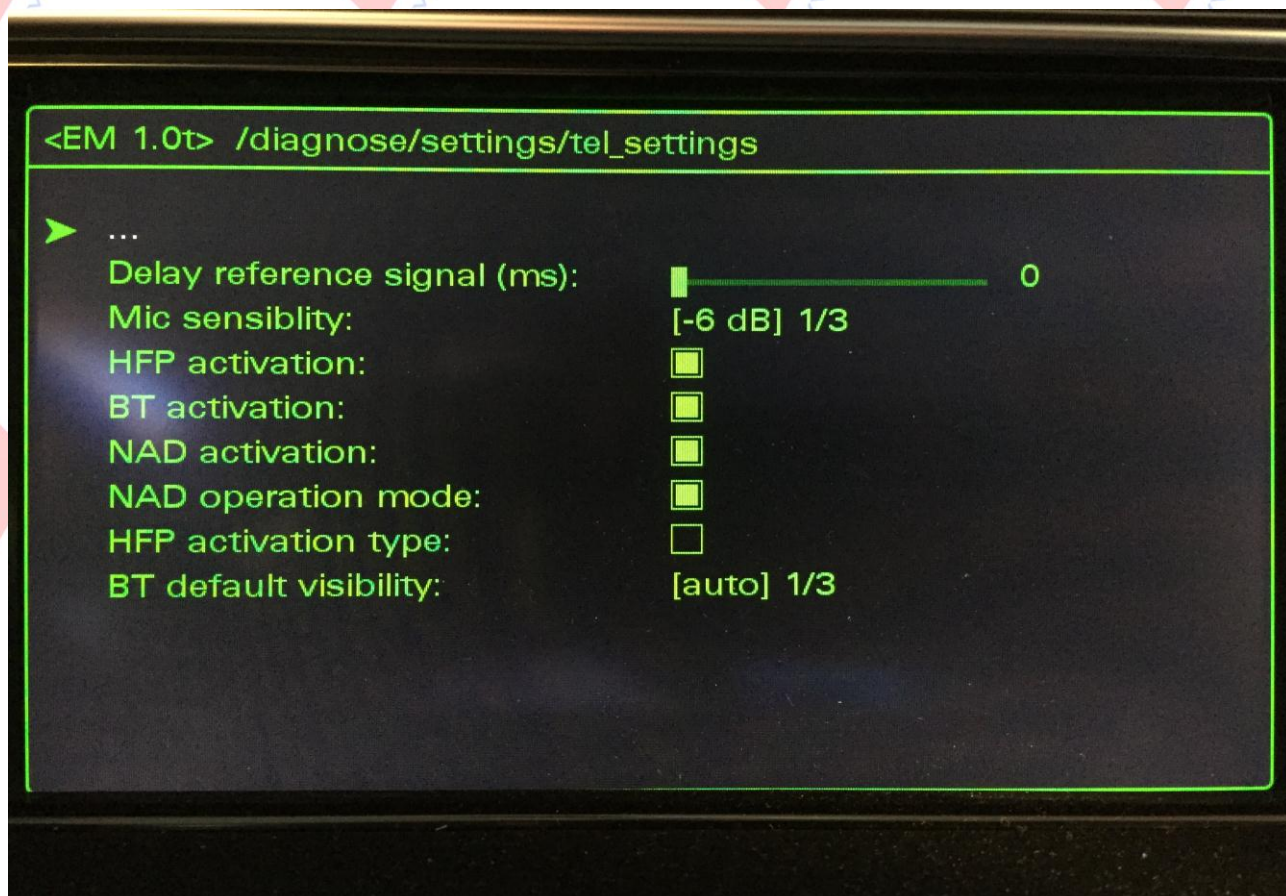
1. /diagnose/coding/internal_modules_1+2: Aquí hay que deseleccionar todo lo que pone "Online...(off)" sin tocar otros valores. En la siguiente imagen podéis ver como queda en mi caso.



2. En /diagnosis/settings/tel_settings hay que activar:

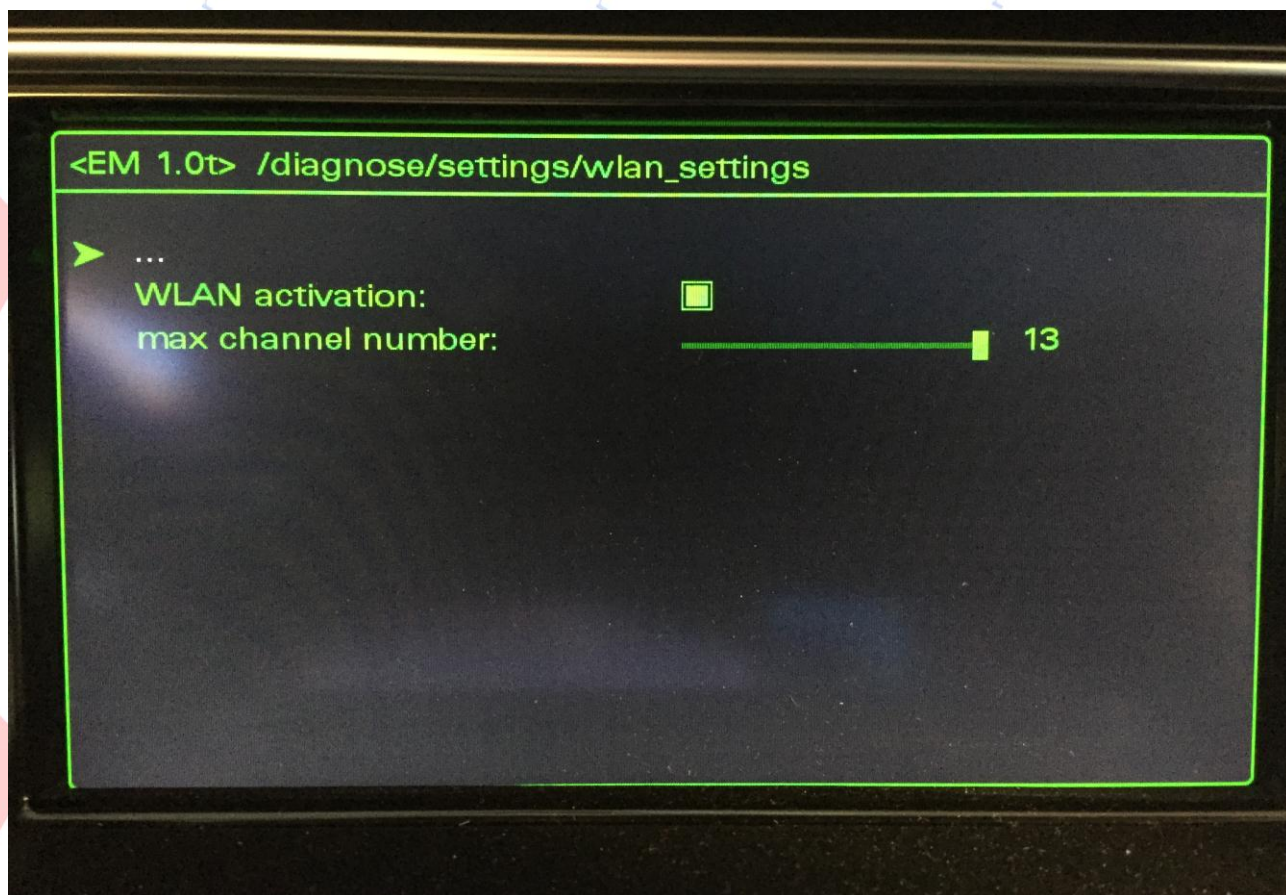
NAD activation ---> Activate

NAD operation mode ---> Activate (En este caso yo lo he puesto pero parece que no es necesario)

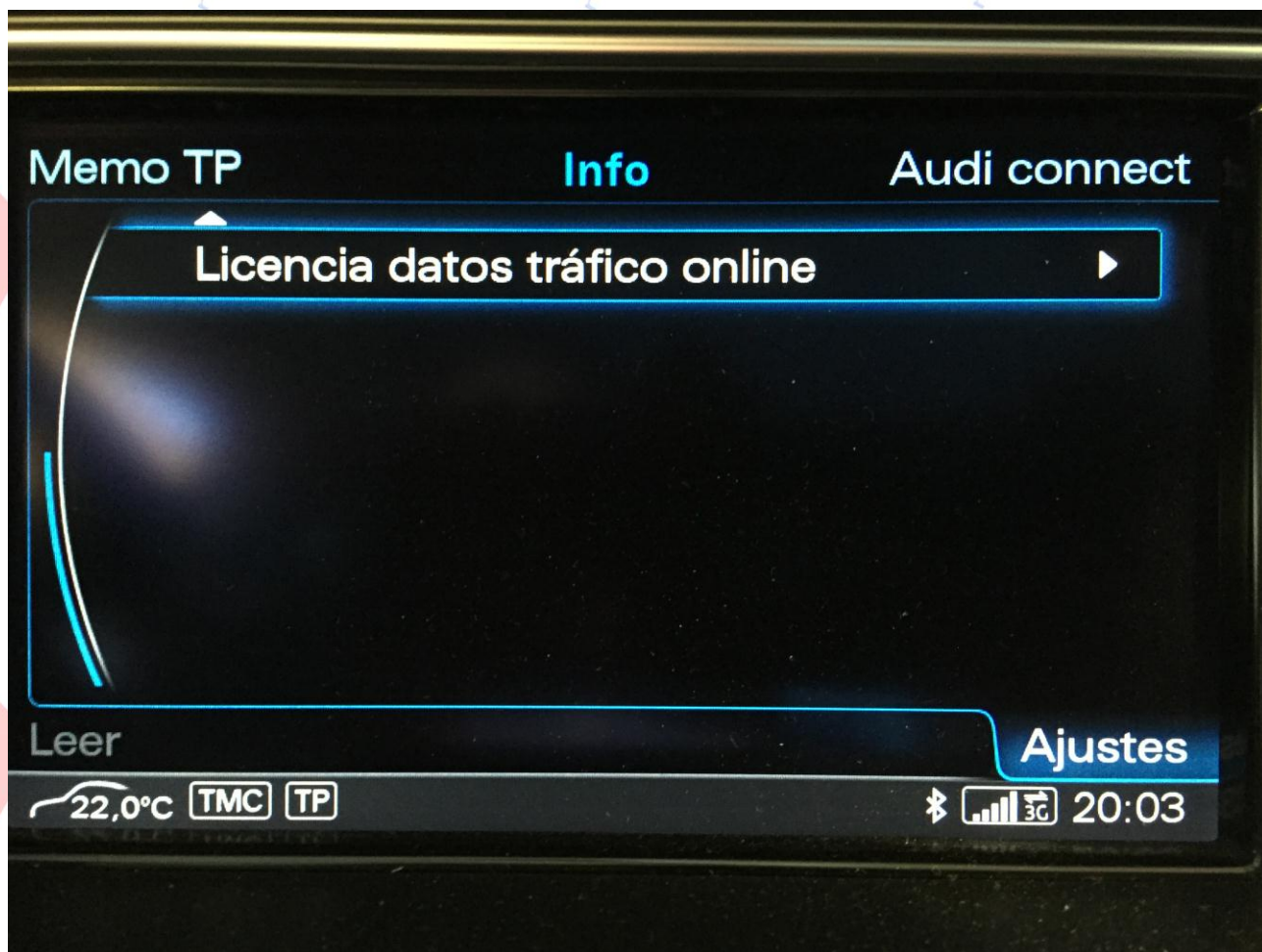


En la siguiente imagen podéis ver como queda en mi caso:

3. Al parecer en algunos casos el hotspot WiFi no viene activado y hay que hacerlo:



Finalmente, pone que hay que ir a info -> settings, habilitar tráfico online y activar la licencia, todo ello en el menú de ajustes del apartado INFO del MMI.



Procedimiento 3. Script configuración

Por último hay que ejecutar el script para configurar el MMI. Para ello os lo descargáis y grabáis el contenido en una tarjeta SD formateada en formato FAT32. Hay que tener en cuenta que el script viene preparado para usar la dirección IP 192.168.1.254 que es la misma que la del interfaz wlan0 de la Raspberry, por lo que si en vuestro caso es distinta o queréis otra hay que cambiarla en el archivo run.sh del paquete del script.

www.audisport-iberica.com by Mad-i, Hispano73 y Yoghurt

Para ejecutar el script basta con esperar a que el MM se haya iniciado completamente (unos 3-4 minutos) e introducir la tarjeta SD en la ranura 1, esperando a que salgan las pantallas de inicio y final del script.

Tras haberlo realizado, ya lo tenéis todo, ahora a disfrutar de vuestro nuevo extra... ;)

Pendiente. Roadmap

Queda pendiente lo siguiente, que iré mirando en lo sucesivo:

1. Probar con dongle usb 3G
2. Controlar el apagado del cacharro

3. Optimizar el arranque para que lo haga más rápido

4. Configurar el roaming de WiFi para poder usar varios teléfonos con el cacharro