raveno
REGISTRO • ALERTA • VIGILANCIA •
ENFERMEDADES DE NOTIFICACION OBLIGATORIA

Manual Técnico

Dispositivo Digital Programable

Proyecto Plataforma de integración tecnológica para el registro, vigilancia y alerta de enfermedades de notificación obligatoria (IT 13I10059).

Descripción del dispositivo

Hardware

- Placa base Raspberry Pi modelo B, con procesador ARMv6
- Adaptador Wifi USB TP-LINK TL-WN725N versión 2, con chipset RTL8188EU
- Pantalla LCD 20x4 compatible con Hitachi HD47780
- Teclado de matrix de 4x4
- Tarjeta SD de 4GB
- Adaptador DC de 5v, 1ª

Componentes de software

Debe asegurarse de que el Raspberry tenga precargado los componentes de software descritos en la siguiente tabla.

Componente de software	Comentario
Sistema operativo Raspbian con kernel Linux 3.12.24+	Se recomienda utilizar exactamente el mismo kernel para asegurar la compatibilidad con el driver del Wifi
Driver para adaptador Wifi 8188eu	El driver es distinto al incluido con el kernel, y da soporte para redes ad-hoc.
Módulo de kernel Batman-adv habilitado	Permite la conectividad entre nodos en una red adhoc
OpenSSH, servidor y cliente	Creación de túneles que encriptan los datos entre las tecleras y evita que todos los nodos tengan acceso a internet.
AutoSSHd	Levantamiento automático de túneles.
Monit	Vigilancia del software de RAVENO en caso de caídas.
Avahi Daemon y Autoipd	Autoconfiguración de cada nodo dentro de la red.
Python 2.7 con librería Rpi	Plataforma que permite la ejecución del software de RAVENO.
Software de notificación RAVENO	El software desarrollado

Archivos de RAVENO

El software incluye los siguientes archivos

Nombre del archivo	Comentario
conf.py	
datos.py	
input.py	
lcd.py	
led.py	
menus.py	
raveno.conf	
raveno.py	Debe tener permiso de ejecución (755)
ravenothread.py	
wifi.py	
scripts/raveno	Script para init.d
scripts/autosshd	Script para init.d
ssh_keys/known_hosts	Claves públicas y privadas para cada dispositivo
ssh_keys/authorized_keys	Claves públicas y privadas para cada dispositivo
kos/	Drivers del adaptador Wifi para distintas versiones del kernel
kernel/	Copia del kernel de linux 3.12.24+ y los drivers

Configuración

Instalación del Kernel 3.12.24+

Para instalar el kernel, ejecutar lo siguiente dentro del directorio kernel/

```
# cp 3.12.24+.tar.gz /lib/modules/
# cp kernel.img /boot
# cd /lib/modules
# tar –xvf 3.12.24+.tar.gz
# reboot
```

Considerar que Raspberry bootea automáticamente el kernel que se llame `kernel.img`. Hacer copias de seguridad de cualquier otro kernel en el directorio `/boot`.

Instalación de los componentes de software

Todos los componentes son instalados usando `apt-get` y forman parte del repositorio de Raspbian.

```
# apt-get install autossh avahi-autoipd avahi-daemon avahi-utils firmware-linux-free monit
```

Luego, copiar el archivo `scripts/autosshd` a `/etc/init.d` y ejecutar

```
# update-rc.d autosshd defaults
```

Editar el archivo /etc/monit/monitrc y en la sección “Services” agregar:

```
check process raveno with pidfile /var/run/raveno.pid
        start program = "/etc/init.d/raveno start"
        stop program = "/etc/init.d/raveno stop"

check process autosshd with pidfile /var/run/autossh/autossh.0.pid
        start program = "/etc/init.d/autosshd start"
        stop program = "/etc/init.d/autosshd stop"
```

Archivo de configuración

Cada dispositivo debe configurarse editando (o creando) el archivo `/etc/raveno.conf`. El archivo es del tipo JSON y contiene los siguientes campos:

Campo	Descripción	Tipo	Valor por defecto
`“cestab”`	El código del establecimiento. Por ejemplo, Hospital Carlos van Buren es 106100. Este valor DEBE ser distinto a cero para que funcione el software. Obligatorio.	Numérico	`0`
`“tipo”`	El rol del dispositivo dentre de la red. Puede ser “cliente” o “gateway”. Dentro de la red debe haber un gateway. Obligatorio	Cadena	`“cliente”`

Campo	Descripción	Tipo	Valor por defecto
“area”	El nombre del área donde está el dispositivo. Los dispositivos con la misma área pertenecerán a la misma red, cuyo nombre será “red_raveno_area”. Obligatorio	Cadena	“sin_area”
“ip”	La dirección IP del servidor remoto. Opcional.	Cadena	“200.14.68.107”
“puerto”	El puerto de conexión	Numérico	443
“protocolo”	El protocolo. “http” o “https”. Opcional	Cadena	“https”
“prefijo”	Ubicación del web service en el servidor remoto. Opcional.	Cadena	“raveno/ravenoMobile”

Si el archivo de configuración no existe, el software de RAVENO creará uno al iniciar, el cual incluye sólo los campos obligatorios con sus valores por defecto, pero fallará su ejecución debido a que el campo “cestab” tendrá como valor cero. Esto es así para asegurar que el administrador del sistema edite el archivo y escriba los valores correspondientes. Los campos opcionales pueden omitirse si se usarán los valores por defecto.

Autoejecución del software

Para que el software se autoejecute se debe editar el archivo `raveno` en el directorio `script`, modificando la ubicación del ejecutable de RAVENO:

```
DIR=/usr/local/raveno           #directorio donde se encuentra raveno.py
DAEMON=$DIR/raveno.py           #debe tener permiso de ejecución
DAEMON_NAME=raveno
```

Una vez hechos los cambios, copiar el archivo al directorio `/etc/init.d`, y ejecutar

```
# update-rc.d raveno defaults
```

Una vez hecho esto, se puede iniciar el software de RAVENO ejecutando

```
# service raveno start
```

y detener el software ejecutando

```
# service raveno stop
```

Instalación del driver Wifi

El driver que debe ejecutar para el funcionamiento del adaptador Wifi es el `8188eu`, el cual no es incluído con el kernel de Linux. En caso de que se esté ejecutando otro driver (`rtl8188eu` o `r8188eu`, o similar), debe verificarse de que exista el archivo `/etc/modprobe.d/nombre_modulo.conf` y que contenga como única entrada:

```
blacklist r8188eu  #si el módulo es r8188eu, dentro del archivo r8188eu.conf
```

También se debe verificar de que el módulo 8188eu se encuentre instalado

```
# ls /lib/modules/3.12.24+/kernel/drivers/net/wireless/
```

En el resultado debe estar presente el archivo 8188eu.ko. De no ser así se debe copiar el correspondiente al kernel instalado desde el directorio `kos/` hacia el directorio mencionado y ejecutar

```
# chown root:root /lib/modules/3.12.24+/kernel/drivers/net/wireless/8188eu.ko
# chmod 644 /lib/modules/3.12.24+/kernel/drivers/net/wireless/8188eu.ko
# depmod -a
```

Configuración de /etc/network/interfaces

Es importante dejar como estática la configuración de la interfaz ethernet para no complicar futuras conexiones a ésta. Las interfaces `wlan0` y `bat0` se dejan manual porque su configuración las maneja el software RAVENO.

```
auto lo
iface lo inet loopback
allow-hotplug wlan0
iface wlan0 inet manual
iface bat0 inet manual
iface eth0 inet static
    address 192.168.x.y
    netmask 255.255.255.0
    gateway 192.168.x.z
```

Configuración de SSH

```
# cd ssh_keys
# cp known_hosts/known_hosts ~/.ssh/
# cp authorized_keys/authorized_keys ~/.ssh/
# cp known_hosts/X.ECDSA /etc/ssh/ssh_host_ecdsa_key
# cp known_hosts/X.ECDSA.pub /etc/ssh_host_ecdsa_key.pub
# cp authorized_keys/XXX ~/.ssh/id_rsa
# cp authorized_keys/XXX.pub ~/.ssh/id_rsa.pub
```

Ejecución del software

Inicialización

El software se ejecuta automáticamente, pero en su inicialización, se hacen cambios en la configuración del sistema operativo:

- Se levanta una red inalámbrica con SSID “red_raveno_area”, donde area es según `/etc/raveno.conf`
- Cambia el hostname a “area-id”, donde area es el nombre del área en según `/etc/raveno.conf` e “id” es
    - “gateway” para el nodo gateway
    - últimos seis digitos del mac address de la interfaz ethernet para los nodos clientes
- Se activa `avahi-daemon` y `avahi-autoipd`, dándole una IP automática al nodo y un nombre dentro de un dominio ad-hoc, llamado “local”. Entonces, los nodos clientes pueden encontrar al nodo gateway con el nombre “area-gateway.local”
- Se crea un túnel SSH desde el nodo cliente hacia el nodo gateway.

Diagnosticar problemas

Si el software deja constantemente de funcionar, puede diagnosticarse el problema deteniendo el servicio y ejecutándolo directamente en la línea de comandos. De esta forma, el software mostrará los mensajes de error en la salida estándar.

```
# service monit stop
# service raveno stop
# /usr/local/raveno/raveno.py
```

También se puede revisar los parámetros de `autossh` y `ssh`.




```
# ps ax| grep ssh
4776 ?        Ss     0:00 /usr/lib/autossh/autossh -M 0 -q -N    -o
ServerAliveCountMax=3 -o ServerAliveInterval=60 -L
127.0.0.1:7575:200.14.68.107:443 pi@testing-gateway.local
 4777 ?        S      0:35 /usr/bin/ssh -q -N -o ServerAliveCountMax=3 -o
ServerAliveInterval=60 -L 127.0.0.1:7575:200.14.68.107:443 pi@testing-
gateway.local
```