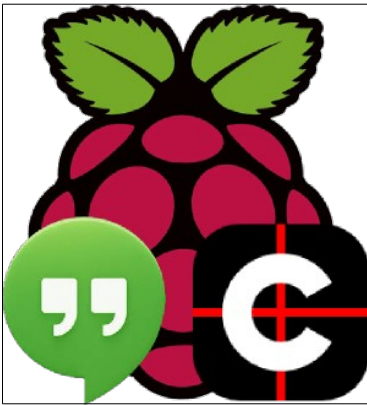


Uso de Hangouts (GTalk) para el envío de notificaciones



... y para mucho más

Hace un tiempo publiqué un post ([este](#)) sobre el uso de GTalk desde la consola, para el envío de notificaciones. Era un poco tosco, pero funcionaba. Al poco, ratoncio me comentaba que se podía hacer eso mismo pero usando un par de funciones del centerim. Desde entonces lo estoy usando de esa forma, pero no había podido poner unas instrucciones decentes, aquí están.

Para no perder el norte, nuestro objetivo es poder enviar notificaciones vía Hangouts y usar este además para enviar ordenes de todo tipo a nuestro RaspberryPi.

Empezamos instalando:

```
apt-get update
apt-get install centerim
```

y ya no hay que instalar más. Tenemos que registrar una cuenta de Gmail en el centerim, así que usaremos una que no tengamos en uso o incluso (it's free!!) crearemos una nueva ¡¡NO USES TU CUENTA PRINCIPAL DE GMAIL!! más que nada porque puedes volverte loco enviándote mensajes a ti mismo ... y además es un poco triste. Para registrar en centerim tu cuenta de Gmail, editamos el archivo `/root/.centerim/config` y añadimos al final:

```
jab_nick CUENTA.PARA:RASPBERRY@gmail.com
jab_pass clavesuperseguradelamuerte
jab_server talk.google.com:5223
jab_osinfo 1
jab_prio 4
jab_ssl 1
```

Si lo vas a ejecutar con un usuario que no sea root, el archivo config está en el home de dicho usuario. Ahora ejecutamos una vez centerim para que ver todo funciona, he incluso comprobaremos que podemos mandar y recibir mensajes.

Ahora vamos a configurar las respuestas automáticas, para ello editamos el archivo `/root/.centerim/external` :

```
%action contestador automagico
event msg
proto all
status all
options stdin stdout
%exec
msg=`cat`
/bin/bash /usr/local/bin/contestador.sh $msg
```

Eso configura centerim para que al recibir cualquier mensaje, ejecute el script contestador.sh pasándole como parámetro el contenido del mensaje. El script /usr/local/bin/contestador.sh lo creamos según nuestras necesidades, un ejemplo puede ser:

```
#!/bin/bash
#programado por Juanmol para http://rsppi.blogspot.com
case $1 in
    espacio)
        echo "El sistema tiene libre `df -h | grep rootfs | awk '{ print $4,$5 }'"
        ;;
    domo)
        echo "Ejecutando $@"
        /bin/bash /usr/local/bin/x10.sh $2 $3
        ;;
    laser)
        if [ "$2" = "on" ];
        then
            echo "Encendiendo el \"LASER\"";
            /bin/bash /usr/local/bin/laserdedestruccionmasiva.sh on ;
        else
            echo "Desconectando el \"LASER\"";
            /bin/bash /usr/local/bin/laserdedestruccionmasiva.sh off ;
        fi
        ;;
    *)
        echo "$1 no lo entiendo, los comandos que entiendo son: espacio, domo, laser"
        ;;
esac
```

Que yo creo que no necesita explicación. Entre otras cosas lo vamos a usar para **mandarle órdenes a nuestro sistema domótico**, por Hangouts (GTalk). Si buscas un poco, solo hay una empresa instaladora de domótica que ofrece esto mismo, con su propio sistema cerrado y cobrándote una considerable suma de dinero.

Añadimos esta línea al archivo /etc/rc.local y la ejecutamos tal cual, para que arranque centerim y lo deje en segundo plano:

```
screen -dmS centerim /usr/bin/centerim
```

Con esto hemos conseguido que nuestro Raspberry nos hable cuando le preguntamos, si queremos que nos diga cosas sin tener que preguntarle primero, cuando se genera un evento en Asterisk, cuando termina de descargar un torrent, cuando ... lo que sea, simplemente tenemos que ejecutar:

```
echo "Evento a las: `date`: $contenido" | centerim -s msg -p jab -t CUENTA.DE.DESTINO@gmail.com
```

Es mucho más fácil y útil. Ya me diréis para que lo vais a usar ;)