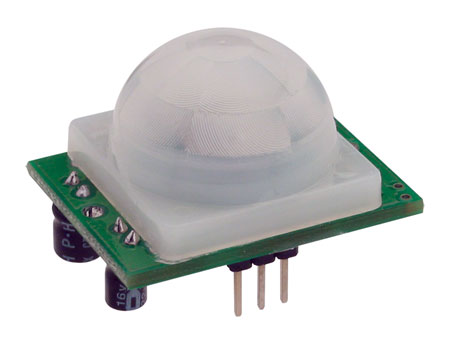
**9.7 ALARMA UTILIZANDO SENSOR DE MOVIMIENTO**

Incluir un sistema de alarma Low Cost en nuestra casa es muy fácil, utilizando un sensor de movimiento y un reproductor de música, en mi caso el omxplayer.

**Los materiales a utilizar serán:**

1 x Sensor de movimiento PIR 

1 x Resistencia de 8 k Ohm

**Conexión**

Las conexiones de este sensor son muy simples, la patilla GND irá conectado a la tierra de la arduino, mientras que la patilla VCC que dará corriente al circuito del sensor, irá a la patilla de 5 V de la arduino, la patilla central out, emitirá una señal ttl lógica (3.3V) cuando detecte movimiento. Esta patilla irá conectada al GPIO4 de la raspberry, que será la encargada de detectar si hay movimiento y en consecuencia reproducir el sonido de alarma.

**Software**

Para empezar **instalaremos el omxplayer**, que es un reproductor de música vía consola de comandos:

*Sudo apt-get install omxplayer*

Una vez instalado **crearemos una carpeta en el home** de pi que contendrá el script de alarma y el archivo de sonido a reproducir.

*Mkdir /home/pi/alarma*

**Creamos el fichero fifo que gestionará las llamadas al comando omxplayer:**

*Mkfifo /tmp/cmd*

*Chown pi /tmp/cmd*

**Procedemos a crear el script principal con** nano /home/pi/alarma/sensor\_ir.sh**:**

#!/bin/bash

while true

do

#analizamos el estado del sensor

sensor=`cat /sys/class/gpio/gpio4/value`

# si el sensor esta detectando activamos la alarma sonara durante 3 minutos

if [ $sensor -eq 1 ]

then

omxplayer /home/pi/alarma/alarma.mp3 </tmp/cmd | 2>/dev/null &

echo -n . >/tmp/cmd &

sleep 2

echo -n + >/tmp/cmd &

sleep 1

echo -n + >/tmp/cmd &

sleep 2

echo -n + >/tmp/cmd &

sleep 182

fi

done

**Ahora crearemos el script de apagado de la alarma con** nano /home/pi/alarma/stop\_alarm.sh :

#!/bin/sh

pid=`ps aux | grep "sh /home/pi/alarma/sensor\_ir.sh" | tr -s " " | cut -d" " -f8,2 | grep -w "S$" | cut -d" " -f 1`

case "$1" in

start)

if [ -e $pid ]

then

echo "Encendiendo alarma"

sh /home/pi/alarma/sensor\_ir.sh &

else

echo ya está encendida

fi

;;

restart)

echo "reiniciando alarma"

sudo kill $pid 2>/dev/null

sh /home/pi/alarma/sensor\_ir.sh &

;;

stop)

echo "Apagando alarma"

sudo kill $pid 2>/dev/null

;;

\*)

echo "se usa de la siguiente forma alarma {start|stop|restart}"

exit 3

;;

esac luego le damos permisos con: sudo chmod a+x /etc/init.d/alarma

Y lo incluimos como servicio con: sudo update-rc.d alarma defaults

F**uncionamiento:**

**Para hacerlo funcionar simplemente debemos ejecutar los siguientes comandos:**

Sudo service alarma start

Sudo service alarma stop

Sudo service alarma restart

Fuentes: <http://liberatucodigo.wordpress.com/2012/05/19/crear-demonio-en-linux-servicios/>, <http://skuarch.blogspot.com.es/2010/02/crear-un-servicio-en-ubuntu.html>