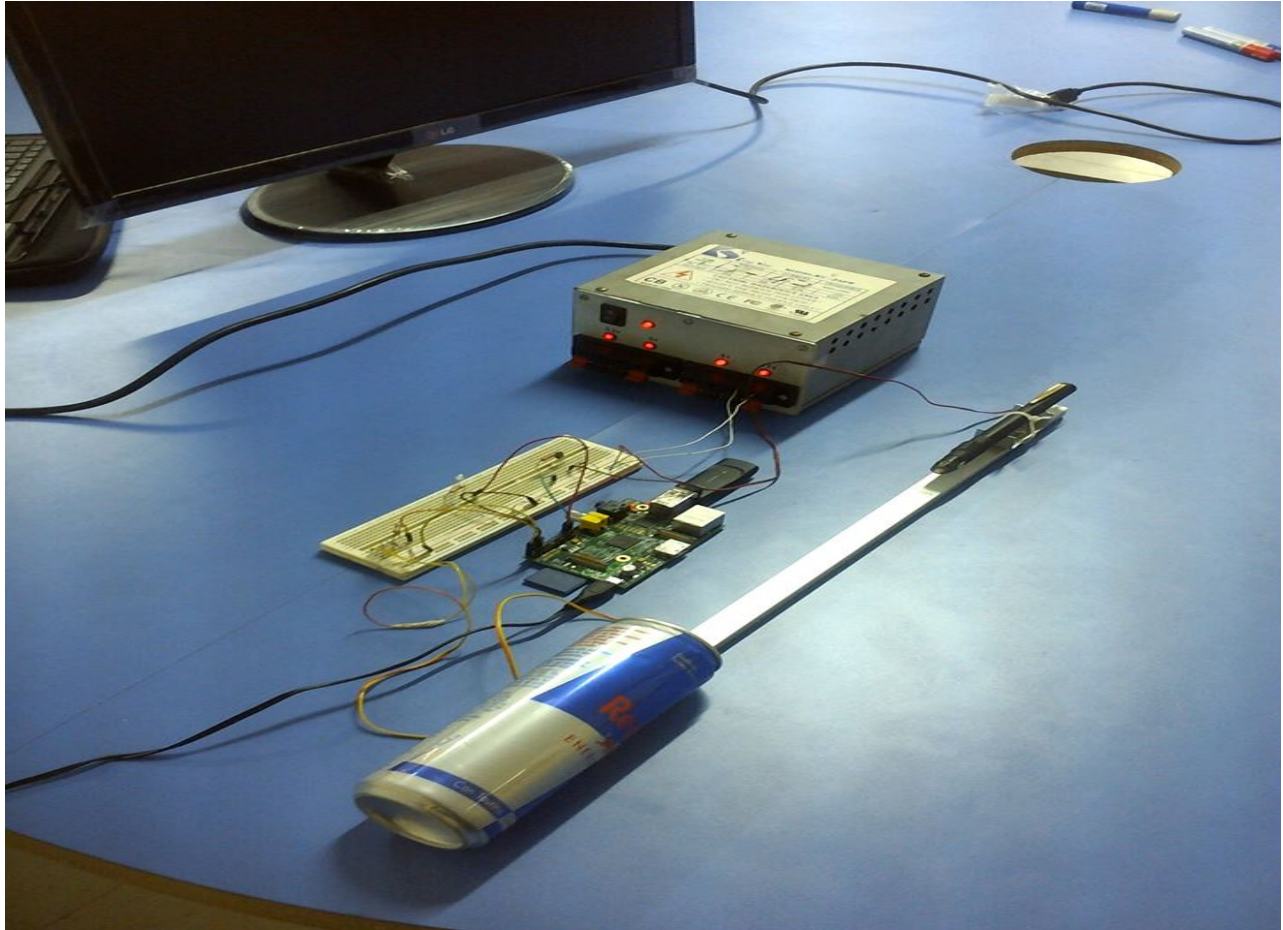


Informe Introducción a la Ingeniería

Sistema para la Detección de Movimiento



Claudio Campuzano, 201330049-9

Sebastián Torrico, 201330061-8

José Turra, 201330028-6

Alejandro Santtiz, 201330008-1

Bryan Meza, 201330032-4

Fecha Entrega:

11-09-2013

Profesor:

Werner Creixell

Índice

Introducción	3
Materiales	4
Raspberry Pi	5
Esquema y Electrónica	7
Preparando la Raspberry Pi	10
Configurando las Funcionalidades	13
Breve explicación del Programa	16
Fotos	17
Conclusión	21
Bibliografía	22

Introducción

Hoy en día la seguridad es muy vulnerable, muchas veces cuando salimos de viaje, llevamos todas nuestras cosas, ya sean, teléfonos, computadores, cosas de uso personal, etc, pero también nos llevamos la inseguridad de dejar la casa sola y/o dejar algún objeto de valor. Tenemos sistemas de alarmas para las casas, el cual su uso es bastante limitado y además hay que instalarlas y activarlas manualmente. A veces un sistema de alarma común y corriente es limitado y no muy efectivo, por ejemplo: no es algo poco cotidiano ver las noticias que muestran acerca del robo de una casa y que los ladrones hayan escapado sin saber nunca más de ellos.

Debido a estas necesidades, hemos implementado un sistema eficaz en instalación, de uso simple, económico y a nivel de cualquier usuario. Es por esto que a base de un sistema de software en conjunto de hardware y de nuestra práctica Raspberry Pi (cual detallaremos más adelante) logramos cubrir estas inseguridades. Un sistema que puede ser llevado a pequeñas y grandes aplicaciones, trabajando con un láser y un receptor que al ser interrumpido se activa una señal. Cualquier persona nos podría decir: “¿Qué es lo novedoso de esto? Existen láseres infrarrojos que detectan movimientos y dan alarma”. Lo cual nosotros responderemos: “¿Lo novedoso? Un sistema que dejas y manejas donde quieras. Un sistema que puede ser activado vía correo electrónico y cada vez que el láser sea interrumpido, te mandamos una notificación al correo e incluso podríamos grabar tal instante”.

A este sistema le podemos dar usos prácticos, ya sea, si tenemos un objeto de valor en casa y queremos respaldarlo ante cualquier persona que pueda tener acceso al lugar. Instalamos el láser en el lugar, como por ejemplo, guardar el objeto en una caja fuerte (dinero por ejemplo), poner el laser en la parte interior de la puerta y poner su receptor al frente de ella, entonces cada vez que se abra la puerta, el láser no llegará al receptor y por ende se nos enviaría una señal al correo electrónico sobre el sistema ha sido interrumpido. Además, podemos activar y desactivar el láser vía correo electrónico con sólo un mensaje en caso de que no haya necesidad de respaldar el objeto. Otra aplicación que podemos darle, es dejar el láser en la entrada de la casa y registrar en caso de que alguien traspase la luz del láser.

Asimismo, no solamente podemos aplicarlo al tema de la seguridad a algún objeto, podemos usarlo también para registrar entradas/salidas, por ejemplo, notificar si alguien pasa por un lugar que no queremos, ya sea un bebé o una mascota.

Desde ahora, comenzaremos a exponer todos los detalles acerca de los materiales y acerca del sistema completo, usaremos imágenes y otras funciones. Explicaremos desde lo más simple a lo más complejo, considerando al lector como si no supiera nada acerca de software y hardware.

Materiales

Materiales para que el sistema funcione:

- 1 Raspberry Pi
- 5 Cables pin
- 1 Cable Ethernet o USB WIFI
- 1 Alimentador 12 Volts
- 1 Laser y Fotorresistencia
- 1 Transformador 9Volts
- 5 Resistencias: 3 de 1k Ohm, 1 de 10k Ohm, 1 de 3.3 Ohm
- 1 Condensador
- 1 Led
- 1 PC con internet

Materiales usados para la programación del sistema:

- 1 Monitor HD
- 1 Cable HDMI
- 1 Teclado
- 1 Tester

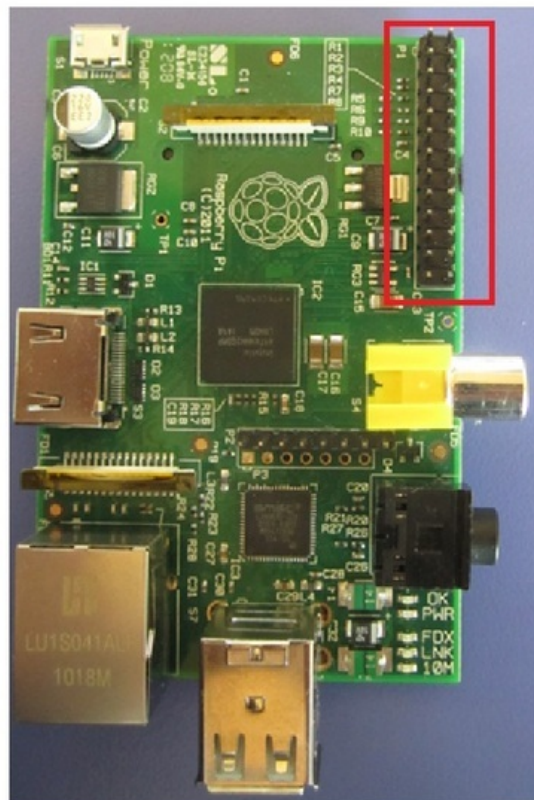
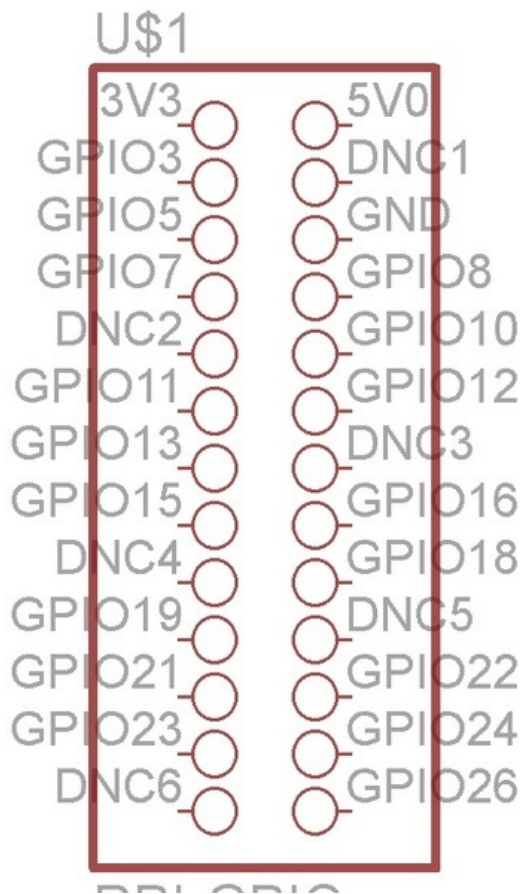
Raspberry Pi

La Raspberry Pi es un computador de placa reducida y de bajo costo creado por Fundación Raspberry Pi para estimular la enseñanza de ciencias de la computación en las escuelas, ya que hoy en día, el usuario tiene más facilidades al momento de trabajar con el computador. Actualmente, los computadores traen todo por defecto y usarlos es un trabajo simple. El objetivo de este aparato es animar a los niños a aprender informática, según su creador. Existen 2 modelos, A y B. El modelo usado por nosotros es el B, es el mas completo.

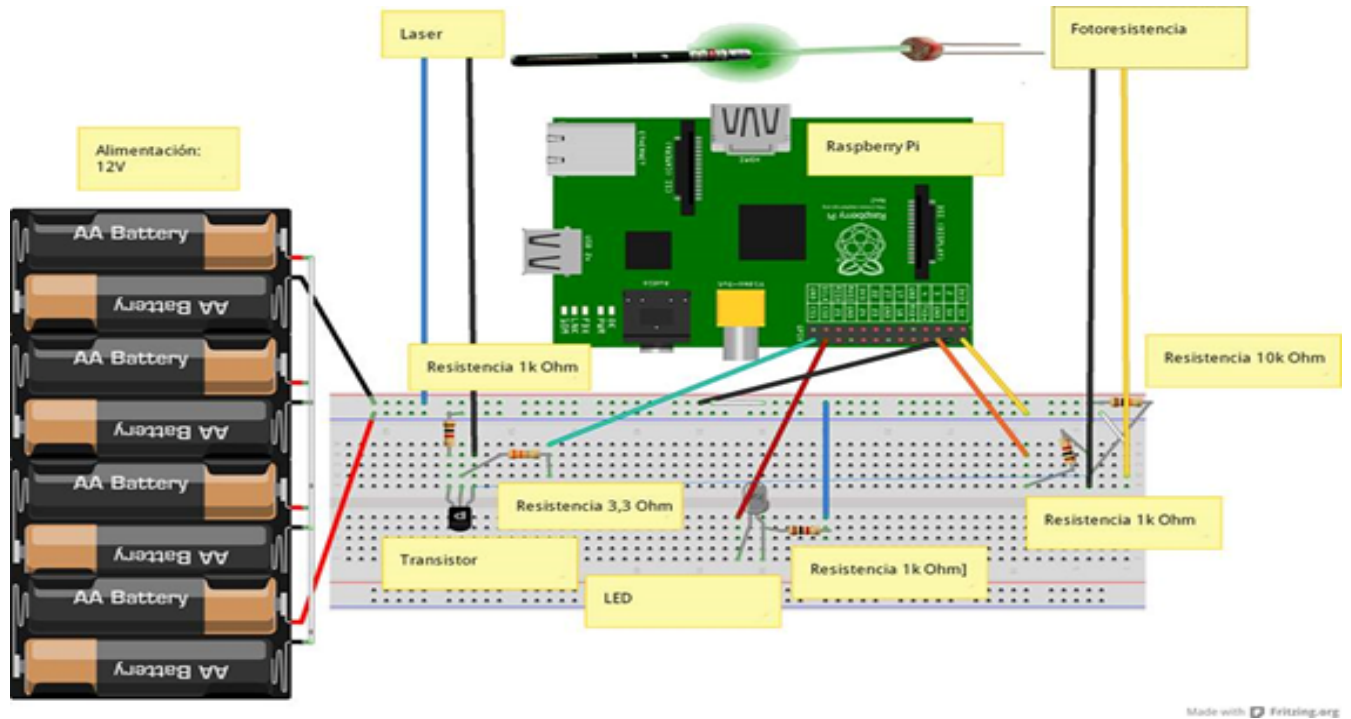
Raspberry Pi - Model B

- Procesador (con gráfica integrada):
 - Broadcom BCM2835. Contiene ARM1176JZFS, con unidad de coma flotante, funciona a 700Mhz y un Videocore 4 GPU.
- Memoria:
 - 256MB
- Características Técnicas de la GPU:
 - La GPU es capaz de mover contenidos con calidad Bluray, usando H.264 hasta 40MBits/s. Dispone un core 3D con soporte para las librerías OpenGL ES2.0 y OpenVG. Es capaz de decodificar 1080p30 H.264 high-profile.
- Dispositivo de Arranque:
 - Memoria SD card. Tras el arranque inicial de la SD se puede terminar el arranque desde un dispositivo USB.
- Conectores:
 - 2x Conectores USB 2.0
 - Conector Ethernet RJ-45 10/100
 - Salida Video Digital HDMI (Audio y Video)
 - Salida De Video Analógico (S-Video)
 - Audio Analógico (Conector 3,5mm)
 - Conector GPIO
 - Conector de alimentación Micro USB
 - Lector de memorias SD (Utilizado para arrancar el dispositivo)
- Alimentación:
 - Vía Micro USB 5 Voltios. Casi cualquier dispositivo con alimentación USB puede servir como fuente de alimentación

- Sistemas Operativos Soportados:
 - Raspbian “wheezy” (Debian)
 - ArchLinux
 - Fedora
 - QtonPi (QT SDK)
 - OpenElec
 - Raspbcm
 - Android (en desarrollo por usuarios)
- Rangos de Temperatura
 - LAN9512 de 0°C a 70°C
 - AP de -40°C a 85°C
- Dimensiones:
 - 85.60mm x 53.98mm x 17mm
- Pack compuesto por:
 - 1x Dispositivo Raspberry PI
 - 1x Hoja indicando la normativa del dispositivo



Esquema y Electrónica



Entremos en Materia: ¿Qué es esto?

Nota: Para complementar la información que leerá a continuación en este documento, puede visitar los siguientes enlaces que lo direccionan a contenido de la enciclopedia libre “Wikipedia”.

- <http://es.wikipedia.org/wiki/Transistor> (Transistores)
- http://es.wikipedia.org/wiki/Unión_PN (Unión PN)
- <http://es.wikipedia.org/wiki/Electrónica> (Electrónica)
- http://es.wikipedia.org/wiki/Resistencia_eléctrica (Resistencias)
- http://es.wikipedia.org/wiki/Placa_de_pruebas (Placa de pruebas o “Proto board” en inglés)
- http://es.wikipedia.org/wiki/Circuitos_de_ayuda_a_la_conmutación_de_transistores (Circuitos de ayuda a la conmutación de transistores)

- Podemos observar en el circuito que tenemos una red básica de LRT, (Led, Resistencia, Transistor). Este circuito representado por el esquema tiene como función explicar de manera gráfica el funcionamiento electrónico del “Sistema de detección de movimiento”.
- El cuadrante central representa el Raspberry-Pi model B.
- Este esquema puede ser utilizado para crear un circuito autosuficiente o dependiente de una CVA, (Caja de Voltaje Auxiliar).

- Los materiales empleados, (resistencias, diodos, transistores, etc), pueden variar según los cálculos realizados por los desarrolladores del proyecto.

¿No entiendes? Vamos paso a paso:

1. Este circuito comienza con la extracción de energía de uno de los pines de la Raspberry-pi el cual aporta 3,3V al circuito para operar los “0” y “1” que se generarán.
Nota: los “0” y “1” son estados del circuito que pueden ser usados para controlar y asignar tareas en un programa.
2. Ahora, luego de extraer la energía del pin 3,3V, paralelamente, al sistema se le aporta una cantidad de energía equivalente a 12V que puede ser suministrada con pilas o con una CVA.
3. Esta conexión de suministro de energía paralela la llamaremos CET, (Concentración de Energía para el Transistor); ¿Su función?...
4. Los transistores pueden funcionar como conductor o resistencia dependiendo de la cantidad de voltaje, (energía), que circule a través de ellos... ¿Cómo es eso posible?...
5. Existen transistores que tienen uniones denominadas “PN”, que se pronuncia pe-ene.

¿Qué hace una unión PN?

Cuando un transistor de tipo PN logra una circulación específica de energía puede lograr 3 cosas. Muy bien adivinaste, a cierto nivel de energía se quemará, pero en los otros niveles restantes sucederán cosas muy importantes. En una de las situaciones el transistor alcanzará el “Umbral Mínimo de Energía” (UME), el cual permite que la energía pase desde la base (B) hasta el colector (C) y corte el paso de energía por el emisor.

-No sé lo que es base, emisor o colector.

-Te lo explicaremos, pone atención.

Un transistor generalmente tiene 3 entradas, salidas, “patitas”, o conexiones, cada una de ellas recibe un nombre

Transistor PNP



De izquierda a derecha y con “la cara plana” del transistor “mirádonos”, el nombre que reciben las conexiones de un transistor de 3 pines es:

PIN 1 a Emisor - PIN 2 a BASE - PIN 3 a Colector

¿Todo “bien”, hasta ahora? Sigamos o repetimos el proceso, hasta que nos quede claro.

Cuando la energía del transistor rompe el UME la energía automáticamente alimenta la conexión de un láser, (seguir el dibujo por la izquierda), el cual genera un haz de luz que es recibido en un fotorreceptor que absorbe la energía aportada por el láser y la utiliza para complementar el voltaje del circuito pasando por el led.

¿Qué LED?

Entonces no analizaste el esquema, detente, léelo aunque no sepas nada, pero así tendrás una visión para que, a lo menos, sepas a qué nos referimos, ¿cierto?

Nota: para seguir con el plan de este “diálogo” con uno mismo es necesario visitar los links asociados a esta parte del informe.

Entonces ahora que el LED está recibiendo energía, luego de que esta fuera “amortiguada” por las resistencias dibujadas en el esquema, tenemos “0” y “1” para trabajar.

¿Cómo lee la Raspberry Pi estos estados?

Un estado “0” o “1” está representado por un voltaje que es recibido en uno de los PINes de la raspberry-pi este voltaje está entre 0 y 2,3 para el “0” y entre 2,4 y 3,2 para el “1”. Estos estados los recibe el PIN número 17 de la Raspberry-Pi.

Ver pagina 17 para fotos del hardware armado

Preparando la Raspberry Pi

En este proyecto trabajaremos con un S.O particular por lo que, primero que haremos, será descargarnos el S.O Raspbian desde el siguiente enlace:

<http://www.raspberrypi.org/downloads>

E instalarlo en nuestra tarjeta SD siguiendo detalladamente las instrucciones que salen en dicha página. Una vez terminado, procederemos a volver a colocar la SD en la Raspberry y conectamos todos los cables necesarios para hacer funcionar nuestra Raspberry Pi. Una vez el SO termine de cargarse nos pedirá un Usuario y contraseña, lo que por defecto son “pi” y “raspberrypi” respectivamente. Ahora como trabajamos con una versión de Debían, necesitaremos hacer unas pequeñas labores de mantenimiento antes por lo que en consola ejecutamos los siguientes comandos:

```
sudo apt-get update  
sudo apt-get upgrade
```

Ahora Instalaremos una herramienta que se encarga de mantener nuestra Raspberry Pi actualizada llamada Hexxeh's RPI, para instalarla digitamos los siguientes códigos en consola:

```
sudo apt-get install ca-certificates  
sudo apt-get install git-core  
sudo wget http://goo.gl/1BOfJ -O /usr/bin/rpi-update && sudo chmod +x  
/usr/bin/rpi-update  
sudo rpi-update  
sudo shutdown -r now
```

Instalado de nuestro cliente de mensajería instantánea

En este trabajo necesitaremos de un cliente de mensajería para poder comunicarnos con la raspberry y que esta también se comuniquen con nosotros, por lo que trabajaremos con uno llamado CenteRim, para esto digitamos los siguientes códigos en consola:

```
sudo apt-get install Centerim
```

Antes de seguir debemos ejecutar al menos una vez el CenteRim para generar la estructura de archivos necesarios:

```
sudo centerim
```

Para el siguiente paso necesitamos contar con una cuenta de correo gmail, por lo que procedemos a crearnos una, o si ya tienes una, puedes usar esta. Una vez creada procederemos a registrar dicha cuenta en nuestro cliente de mensajería CenteRim, por lo que en consola digitamos:

```
sudo nano /root/.centerim/config
```

Y al final de este archivo añadimos lo siguiente:

```
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
```

Con esto hecho, si ejecutamos nuevamente nuestro CenteRim podremos mandar mail y recibir y todo se podrá hacer desde consola, ahora procedemos a ahora a configurar nuestra contestador automático que será el pilar central de este proyecto. Por lo que ejecutamos en consola lo siguiente:

```
sudo nano /root/.centerim/external
```

Y editamos el archivo dejándolo de con esta información:

```
%action contestador automatico
event msg
proto all
status all
options stdin stdout
%exec
msg=`cat`
/bin/bash /home/pi/proyecto/verificador.sh $msg
```

Ahora en términos simples, con lo anterior, hicimos que cada vez que la que mandemos un mensaje por GTalk a la cuenta que registramos con la raspberry, el mensaje se guardará en la variable “msg”, la cual después será exportada a nuestro programa que a continuación creamos que se llamara verificador.sh, que dependiendo de que le programemos nos devolverá un mensaje de vuelta automático con lo que le hayamos indicado en verificado.sh.

Hecho esto, ahora nosotros deseamos que este programa funcione como un Demonio, que quiero decir que siempre esté corriendo en el sistema, para que si detecta algún mensaje ejecute la acción y respondan como corresponde por lo que para esto, tenemos que hacer que centerim se ejecute desde el principio y nunca se cierre, para esto añadimos debemos añadir al final del archivo rc.local que es la encargada de ver que programas se ejecutan al inicio para esto ejecutamos lo siguiente: **Sudo nano** /etc/rc.local

Y al final del archivo añadimos esto

screen -dmS centerim /usr/bin/centerim

Al usar screen este programa nos permite dejar al CenteRim ejecutándose en por así decirlo una terminal auxiliar virtual, por lo que este programa resulta bastante útil para nuestros propósitos.

Ahora que ya tenemos nuestra recepción y respuesta de mensajes casi listo, ahora mostrare un pequeño ejemplo de lo que podría contener nuestro archivo verificador.sh:

```
#!/bin/bash

if [ $1 = "luces" ]; then
  if [ $2 = "on" ]; then
    /home/pi/proyecto/electronica/lucesOn
  elif [ $2 = "off" ]; then
    /home/pi/proyecto/electronica/lucesOff
  else
    echo "clave incorrecta, recuerde (on/off)"
  fi
fi

elif [ $1 = "seguridad" ]; then
  if [ $2 = "on" ]; then
    /home/pi/proyecto/electronica/laserOn
  elif [ $2 = "off" ]; then
    /home/pi/proyecto/electronica/laserOff
  else
    echo "clave incorrecta, recuerde (on/off)"
  fi
fi

elif [ $1 = "ayuda" ]; then
  if [ $2 = "seguridad" ]; then
    echo "Activacion o desactivacion del sistema de seguridad de laser"
  elif [ $2 = "luces" ]; then
    echo "Activacion o desactivacion del sistema de luces"
  elif [ $2 = "estado" ]; then
    echo "Muestra el estado del las funciones (si estan apagadas o no)"
  else
    echo "opciones disponibles:"
    echo "-> 'luces on' // 'luces off'"
    echo "-> 'seguridad on' // 'seguridad off'"
    echo "-> 'estado'"
    echo "-> 'Si requiere mas info acerca de las funciones teclee por ej 'ayuda
seguridad'"
  fi
fi

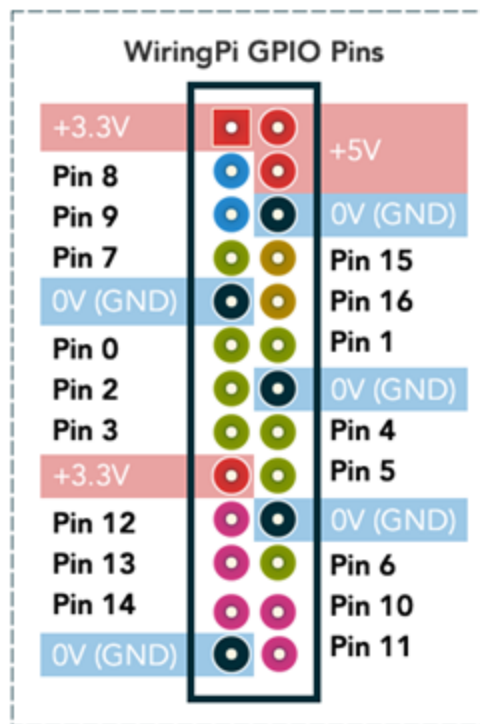
else
  echo "Opcion ingresada no valida, para mayor informacion teclee 'ayuda'"
fi
```

Configurando las funcionalidades

Antes de seguir para poder programar en c lo que deseamos hacer con nuestra raspberry necesitamos una librería, que no trae por defecto el sistema por lo cual necesitamos instalarla se llama WiringPi por lo que procedemos a ejecutar las siguiente línea de código:

```
sudo git clone git://git.drogon.net/wiringPi  
cd wiringPi  
sudo git pull origin  
cd wiringPi  
sudo ./build
```

Ahora que ya tenemos lista nuestra librería, podemos empezar a programar los pines de nuestra raspberry, antes de seguir cabe destacar que esta librería los pines los enumera de la siguiente manera:



Ahora programaremos nuestro encendido de láser como ejemplo, ya que las demás funcionalidades en general tienen la misma estructura:

```
#include <stdio.h>
#include <wiringPi.h>

#define LED 10

int main (void)
{
    printf("Sistema de Luces ENCENDIDO");
    wiringPiSetup () ;
    pinMode (LED, OUTPUT) ;
    digitalWrite (LED, HIGH);
    return 0;
}
```

Una breve explicación del programa:

1. Llamamos a las dos bibliotecas que necesitaremos para trabajar
2. A la variable LED la definimos con el número 10, esto quiere decir que trabajaremos con el pin numero 10
3. Aplicamos un printf el cual devolverá un mensaje el cual, se auto enviara gracias a CenteRim (esto debido a que el programa se ejecuta dentro del programa principal llamado verificador.sh que es nuestro súper auto contestador)
4. Ahora ejecutamos wirinPiSetup() que llama a la setup de wiring pi.
5. Luego ejecutamos un pinMode(LED,OUTPUT) que le indica al programa que al pin (en este caso el 10) lo ejecute en modo de salida
6. Ahora con el digitalWrite(LED,HIGHT) le decimos, dale energía a ese pin, lo que ocasiona el encendido del láser

Y eso, seria, para apagar el láser, lo único que difiere en el programa es que en digitalWrite, se cambia el HIGHT por LOW, que le indica deje de dar corriente.

Ahora necesitamos compilar nuestro programa, para que esté listo para el consumo por lo cual ejecutamos (debido a que usamos la biblioteca wiringPi, la forma del código de compilación es algo distinta):

```
gcc -Wall -o laserOn laserOn.c -lwiringPi
```

Y con esto ya casi tenemos listo nuestro proyecto, ahora necesitamos idearse la forma de que cuando detecte movimiento, mande el mensaje de alerta, para esto creamos bash con la siguiente estructura:

```

#!/bin/bash
while true
do
    sudo /home/pi/proyecto/electronica/detector
    sleep 5s
done

```

Lo que hará este bash, es que se iniciará al principio del sistema y correrá siempre, ejecutando el programa detector (el cual veremos acentuación) por lo que, al igual que lo hicimos con el CenteRim, debemos añadir la siguiente línea de código a rc.local, para que se ejecute al inicio:

```
screen -dmS DEMONIO /home/pi/demon.sh
```

Ahora debemos programar el programa detector en c, el que detectará cuando el láser es interrumpido.

```

#include <stdio.h>
#include <wiringPi.h>

#define LED 15

int main (void)
{
    int val;
    int control=1;
    wiringPiSetup () ;
    pinMode (LED, INPUT) ;
    while (control==1)
    {
        val = digitalRead(LED);
        if (!val)
        {
            control=0;
        }
    };
    system("echo ALERTA ALERTA, Movimiento DETECTADO | centerim -s msg -p jab -t
klavdio@gmail.com");
    return 0;
}

```

Una breve explicación del programa:

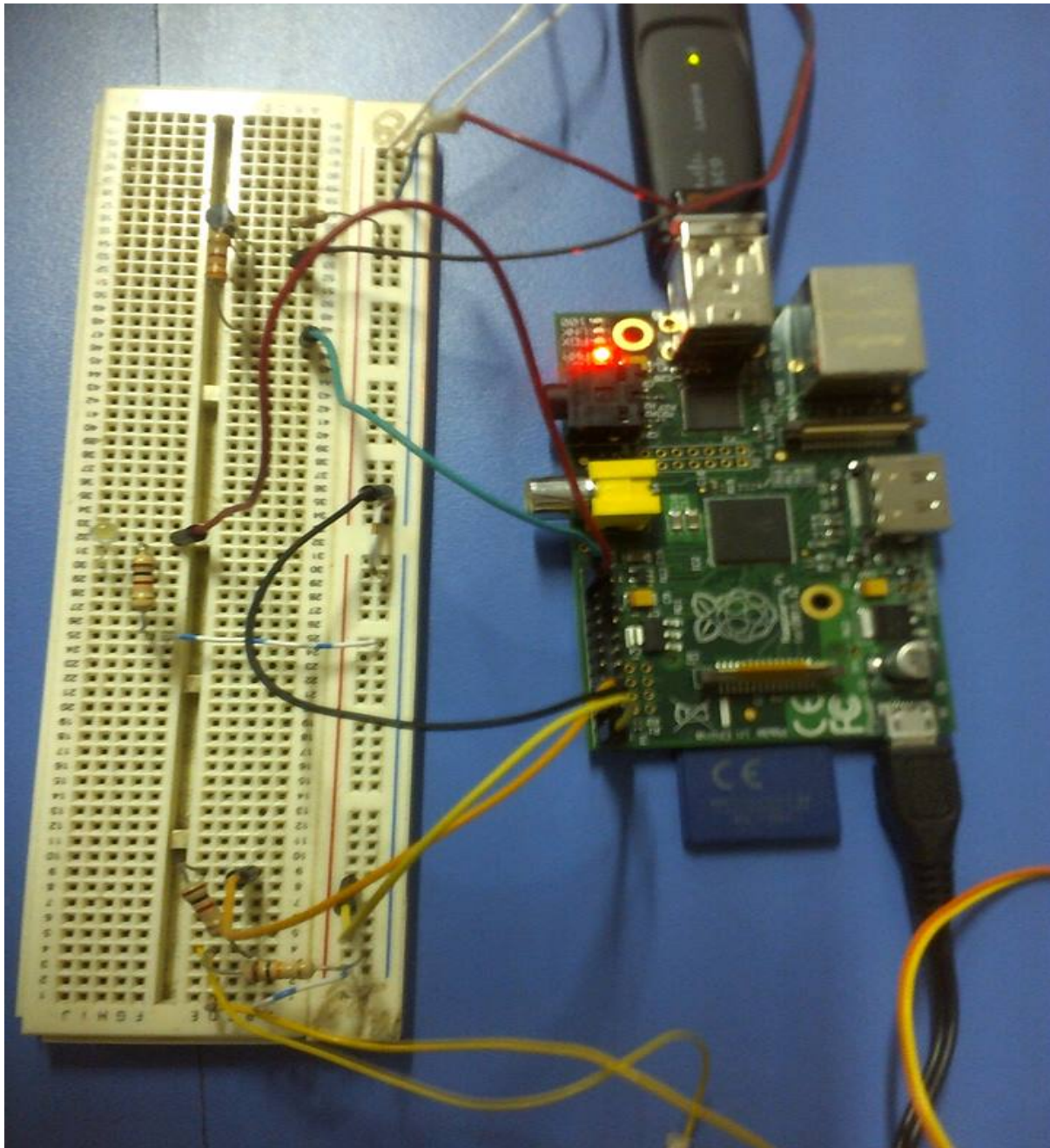
1. Llamamos a las bibliotecas que ocuparemos
2. Definimos el pin que ocuparemos
3. Declaramos las variables que ocuparemos, una es val y control, ambas de int, sin embargo, la primera es un tipo int pero la ocuparemos como un boolean.
4. Inicializamos el wiringSetup y declaramos al pin para que funciones esta vez de entrada, por así decirle que ahora lea
5. Ahora nos metemos en un ciclo repetitivo que solamente se saldrá cuando la variable control sea distinta de 1.
6. En la variable val guardamos el el valor booleano que devuelva la función digitalRead(led), esta función lee el pin "led"
7. Ahora ponemos un if y solamente cuando ese if sea falso, ósea cuando el láser está siendo interrumpido, la variable control tornara a 1, lo que conlleva a que se salga del ciclo.
8. Y como solamente se sale del ciclo cuando hubo interrupción del láser, a continuación mandamos el mensaje de alerta.
9. Finalmente y como anteriormente lo hicimos compilamos dicho programa: `gcc -Wall -o detector detector.c -lwiringPi`

Y con esto damos por finalizado lo que es la programación de la raspberry en si, el procedimiento en sí queda resumido de la siguiente forma:

1. -Preparación de la raspberry
 - Instalación del SO
 - Actualizamos el Firmware de la raspberry Pi
2. -Instalación del CenteRim
 - Configuración de la cuenta que usaremos
 - Configuración del archivo auto contestador
 - Configuración para iniciar CenteRim al inicio en modo oculto
3. -Programado de las funcionalidades
 - Programado de nuestro archivo verificador.sh, el encargado de ver que es lo que se hace
 - Programado de las diferentes funciones de la raspberry encendido de láser u otros
 - Creación del bash demonio, que correrá oculto de la misma forma que el CenteRim esperando que haya alguna señal de detección

Quedan cosas por mejorar y pulir, pero en un comienzo es un gran logro.

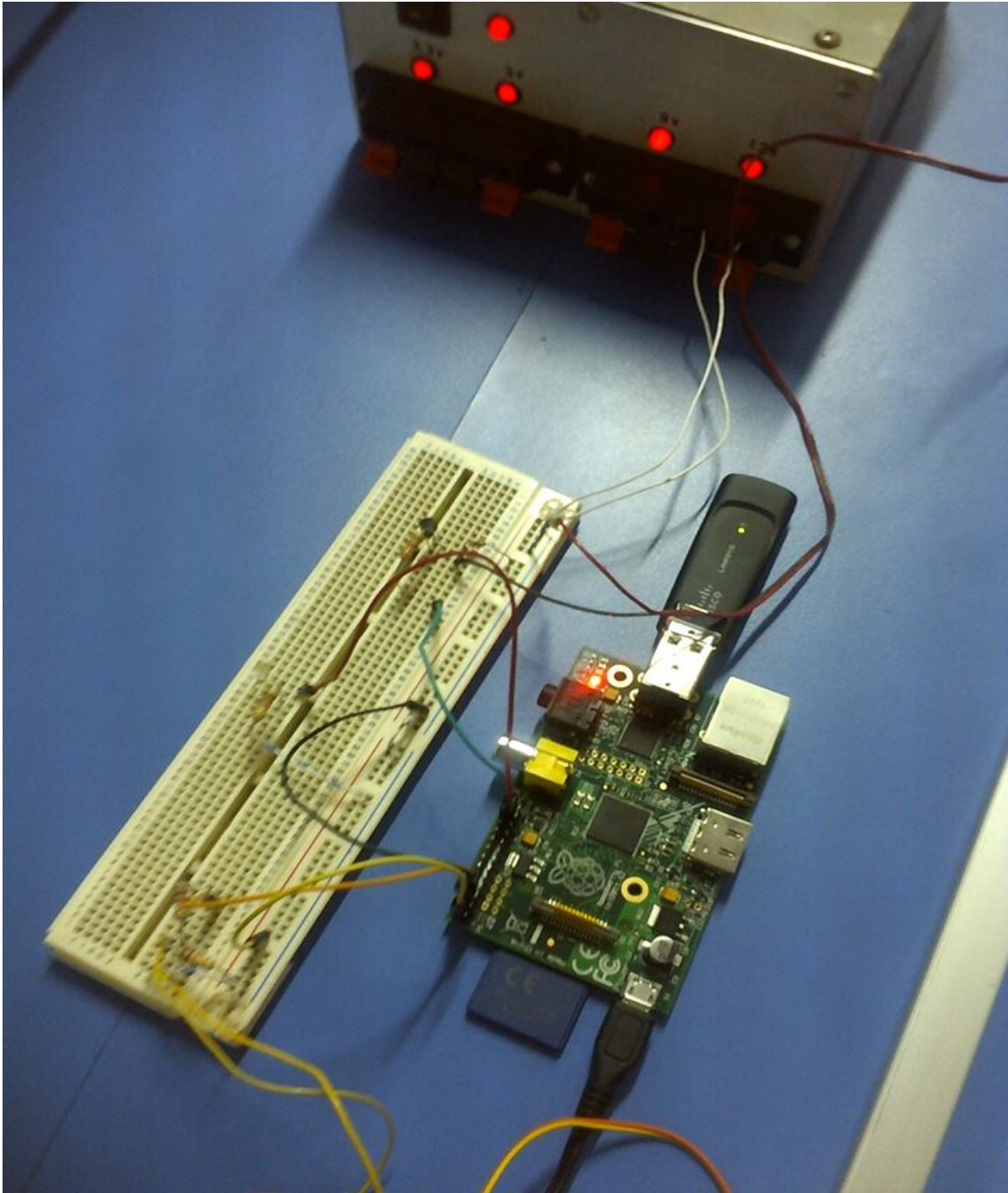
Fotos



Acá se muestra como debe quedar la parte del hardware. Usando todos los materiales.



Aquí, el sistema conectado y mostrando en pantalla el código que se realizó para que éste funcionara.



Otro ángulo. Se puede notar que la fuente de alimentación está dando 12V.

Conclusión

De esta manera, logramos darle vida a un sistema tan sólo con la Raspberry Pi, quien funciona como cerebro de todo. Como vimos anteriormente, los usos que le podemos dar a este sistema son muchos, realmente, es irrelevante dar una cifra ya que puede ser usado para cualquier tipo de seguridad.

Lo práctico de todo esto es que todo se puede manejar a distancia y no es necesario tener a alguien, ya sea, una empresa de alarma de quien debamos depender.

Es fantástico pensar en cómo se pueden hacer cosas tan prácticas con un mini computador. Esperamos que con este informe se hayan aclarado todos los detalles técnicos acerca del hardware y software del sistema, incluyendo de su función.

El sistema puede ser modificado para hacer mas cosas, solo es cosa de ingenio para ver que se puede agregar.

Al final de cuentas, con todo lo visto anteriormente, quizás no sea del todo simple ya que hay que poseer dominio del tema (esa es la razón de este informe) pero una cosa es cierta: es considerablemente más económico que cualquier sistema de seguridad. De todos modos depende de la persona a que recurrir, nosotros sólo otorgamos la facilidad de ahorrar y dejar todo a manos del usuario.

Bibliografía

Software:

<http://rsppi.blogspot.com.es/2013/07/uso-de-hangouts-gtalk-para-el-envio-de.html>

http://www.centerim.org/index.php/CenterIM4_Documentation#External_actions_and_auto-responses

<http://wiringpi.com/reference/setup/>

<http://wiringpi.com/reference/core-functions/>

<http://www.gnu.org/software/bash/manual/bashref.html>

Hardware:

<http://es.wikipedia.org/wiki/Transistor> (Transistores)

http://es.wikipedia.org/wiki/Uni%C3%B3n_PN (Unión PN)

<http://es.wikipedia.org/wiki/Electr%C3%B3nica> (Electrónica)

http://es.wikipedia.org/wiki/Resistencia_el%C3%A9ctrica (Resistencias)

http://es.wikipedia.org/wiki/Placa_de_pruebas (Placa de pruebas o “Proto board” en inglés)

http://es.wikipedia.org/wiki/Circuitos_de_ayuda_a_la_conmutaci%C3%B3n_de_transistores (Circuitos de ayuda a la conmutación de transistores)