

## Raspberry Pi, control de interfaces mediante aplicaciones Web

Raspberry Pi, control interfaces using web  
applications

Medina Alfonso\*  
Quintero Veronica\*\*  
Diaz Arnoldo\*\*\*

\* Estudiante de Ingeniería de Sistemas y Computación, Area terminal especialidad de Redes en el Instituto Tecnológico de Mexicali. y Desarrollador de aplicaciones web para la empresa Ynnovateck.

[4lfonsomedina@gmail.com](mailto:4lfonsomedina@gmail.com)

\*\*Ingeniero en Computación, Maestría en Electrónica, especialidad en Telecomunicaciones, Maestría en Ciencias de la Ingeniería, especialidad de Redes y Telecomunicación, Profesor del Instituto Tecnológico de Mexicali  
[veroquintero@yandex.ru](mailto:veroquintero@yandex.ru)

\*\*\*Ingeniero en ciencias computacionales, Maestría en Ciencias en Tecnología de Redes e Informática, Doctorado en Arquitectura y Tecnología de los Sistemas Informáticos (Universidad Politécnica de Valencia España) [adiaz@itmexicali.edu.mx](mailto:adiaz@itmexicali.edu.mx)

**Correspondencia:** Veronica Quintero Rosas, Instituto Tecnológico de Mexicali, México. Departamento. de Sistemas y Computación  
Av. Tecnológico s/n Col. Elías Calles  
Mexicali, B.C, México, 21376

**Resumen** – Los sistemas de comunicación de una civilización demuestran que tan avanzados se encuentran en sus tecnologías informáticas, tales sistemas de comunicación pueden ser; el envío de un correo, una conexión con una base de datos en tiempo real o la ejecución de aplicaciones web en la “nube”, este método solo espera poder alojar información para poder utilizar nuestras aplicaciones sin tener que depender de que estas estén en nuestro computador para poderlas manipular y utilizar cuando nosotros queramos y desde cualquier parte del mundo. Teniendo todo esto a nuestra disposición pudimos realizar aplicaciones web las cuales manipulan hardware a distancia en nuestro caso con la implementación del dispositivo Raspberry pi como intermediario de control entre la aplicación y el hardware a la Raspberry con esto todas nuestras aplicaciones solo estaban cargadas en la memoria de la Raspberry y con la dirección física de esta podíamos acceder y utilizarlas entre los proyectos que realizamos está el del encendido y apagado de una lámpara con un foco de 60 watts y el controlar un brazo robótico con el acelerómetro de nuestro Smartphone estos proyectos los veremos más afondo en este documento.

**Palabras Claves:** Raspberry pi, programación, Web, servidor, parámetros gráficos, placa reducida.

## 1 Objetivo

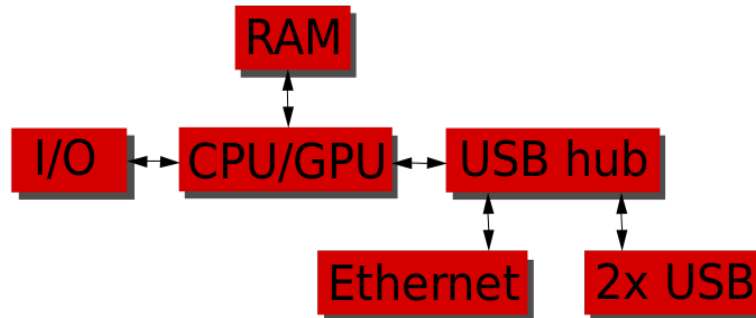
Demostrar la capacidad del dispositivo Raspberry pi en el entorno de desarrollo de aplicaciones web así como el uso, para controlar distintos dispositivos de hardware por medio de una programación avanzada y un conocimiento de electrónica básica.

## 2 Raspberry pi

Raspberry Pi un ordenador de placa reducida con un costo muy bajo, desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.

El diseño contiene un procesador central(CPU)ARM1176JZF- Sa700MHz(el firmware incluye modos “Turbo” para que el usuario pueda hacerle overclock de hasta1 GHz sin perder la garantía),un procesador gráfico(GPU)Video Core IV, y 512 MB de memoria RAM aunque originalmente al ser lanzado eran 256MB.

El diseño no incluye una unidad de memoria de estado sólido, ya que usa una tarjeta SD para el almacenamiento permanente [3]; tampoco incluye fuente de alimentación carcasa. El 29 de febrero de 2012 la fundación empezó a aceptar órdenes de compra del modelo “B”, y el 4 de febrero de 2013 del modelo “A” [12], [10].

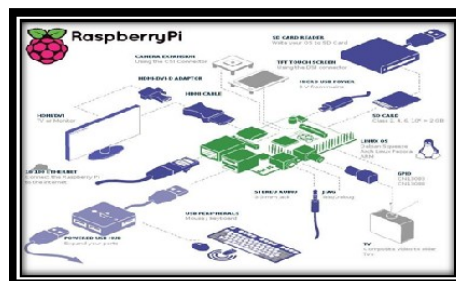


**Fig. 1:** Diagrama a bloques Raspberry

A pesar de que el Modelo A no tiene un puerto RJ45, se puede conectar a una red usando un adaptador USB-Ethernet suministrado por el usuario. Por otro lado, a ambos modelos se les puede conectar un adaptador Wi-Fi por USB, para tener acceso a redes inalámbricas o internet.

El sistema cuenta con 256MB de memoria RAM en su modelo “A”, y el modelo “B” con 512 MB de memoria RAM. Como es típico en los ordenadores modernos, se pueden usar teclados y ratones con conexión USB compatible con Raspberry Pi.

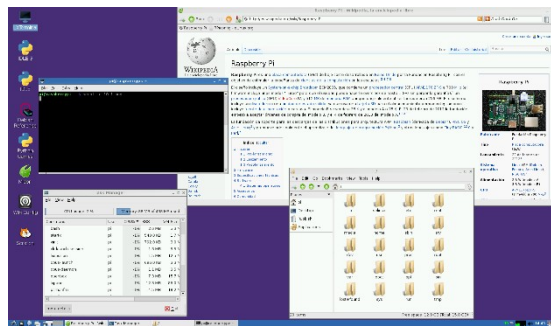
Raspberry Pi [3] no incluye reloj en tiempo real, por lo que el sistema operativo debe usar un servidor de hora en red; o pedir al usuario la hora en el momento de arrancar el ordenador. Sin embargo se podría añadir un reloj en tiempo real (como el DS1307) con una batería mediante el uso de la interface I<sup>2</sup>C.



**Fig. 3:** Tipos de dispositivos para interconectar a Raspber

### 3        **Sistemas operativos Raspberry pi**

Raspberry no utiliza un sistema operativo común ya que su microprocesador ARM11 es algo limitado en ciertos rasgos no permite la estructura de algunos sistemas operativos que nosotros conocemos como lo que es Ubuntu o Windows inclusive Mac por lo mismo tiene una gama de sistemas basados en Linux solo para este dispositivo entre ellos podemos encontrar por ejemplo Arch Linux, Raspbian, RiscOs, Pidora, RaspBmc y OpenElec todos estos sistemas son especializados en ciertas tareas que la Raspberry puede hacer pero el más utilizado y con mayor soporte de programación es Raspbian es lo muy parecido a debían y fácil de utilizar con herramientas muy buenas y fáciles de utilizar.



**Fig. 4:** Sistema operativo Raspbian el más utilizado para desarrollo.

### 4        **Entorno**

Entre los distintos lenguajes de programación que este dispositivo puede interpretar:

- a) C
- b) Python
- c) C++
- d) PHP

También se pueden manejar bases de datos y poder administrarlas para cada lenguaje que se use. La Raspberry Pi dispone de un puerto de expansión de entradas y salidas; de propósito general conocido como GPIO y que se accede a él a través de un conector de cabecera rotulado como P1 en la placa. Todos los pines, excepto los de alimentación son reconfigurables y por tanto podremos adaptarlos a nuestras necesidades; aunque por defecto una serie de pines están configurados para la conexión de las interfaces UART, I2C y SPI. La disposición y funciones de sus pines para la placa perteneciente a la Rev2 es la que se muestra en la figura 5 abajo, los pines GPIO4, GPIO17, GPIO27, GPIO22, GPIO18, GPIO23, GPIO24 y GPIO25 son los que vienen configurados por defecto para entradas y salidas digitales [9].

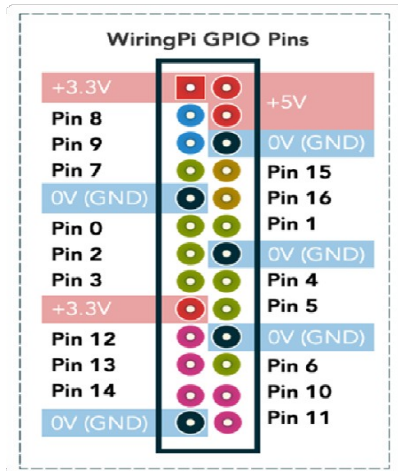


Fig. 5: Puertos GPIO.

## 5 Funcionamiento básico.

1. Antes de todo tenemos que tener todos nuestros elementos de hardware conectados: memoriaSD con nuestro sistema operativo ya montado, mouse, monitor y un teclado.
2. Conectamos a corriente el dispositivo, encendemos y empezara inmediatamente a cargar los archivos del sistema operativo.
3. Al cargar el sistema operativo; se debe introducir por default el usuario “pi” y password “raspberrypi”.
4. Al introducir el password y usuarios podrá acceder al sistema de raspberrypi, el cual es muy similar a la terminal manejada en diversos sistemas basados en Linux. Por medio de comandos podemos realizar varias tareas como la configuración de tu ip, conectarte a una red o entrar a una carpeta en específico, instalar o correr un programa etc.
5. Para poder trabajar con un entorno grafico de raspberry solo habrá que introducir el comando startx ya continuación el sistema operativo pasara de un entorno de terminal, a un entorno completamente gráfico.

```
Debian GNU/Linux wheezy/sid raspberrypi tty1
raspberrypi login: pi
Password:
Last login: Tue Aug 21 21:24:50 EDT 2012 on tty1
Linux raspberrypi 3.1.9+ #168 PREEMPT Sat Jul 14 18:56:31 BST 2012 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Type 'startx' to launch a graphical session

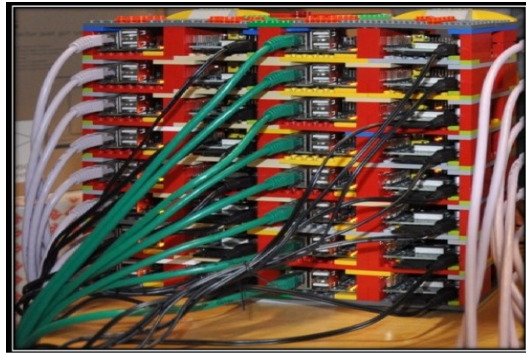
pi@raspberrypi ~ $
```

Fig. 6: Entorno de terminal antes de usar el comando start x.

## 6 ANTECEDENTES.

Existen varios proyectos ya implementados con este dispositivo, los más conocidos se basan en la idea de crear “controles remotos” para un carro de juguete por medio del wifi o Bluetooth.

Pero el más conocido mundialmente es la creación de una computadora portátil o de escritorio, incluso como una consola de videojuegos, entre los más populares se encuentra un contador Geiger, conglomerados mejor conocidos como Clúster, servidores, etc.

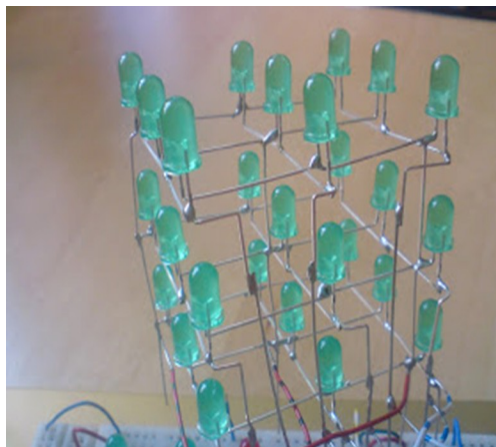


**Fig. 7:** Clúster realizado en la universidad Southampton con 64 dispositivos Raspberry.

## 7 Proyectos realizados.

### 7.1 Cubo de Leds.

En una labor de investigación y de interés por el dispositivo y su funcionalidad logramos formar varios proyectos dando comienzo por una matriz de leds de 3 dimensiones, el cual por medio de un algoritmo logramos controlar el encendido y apagado de cada uno de los leds del cubo.



**Fig.9:** Cubo de leds verdes.

En cada posición se otorgaron a su vez diversas secuencias de encendido creando efectos visuales muy llamativos, esto lo desarrollamos en lenguaje C.

```
//polarizamos las 3 capas
digitalWrite (9, 0) ;
digitalWrite (10, 0) ;
digitalWrite (11, 0) ;

//ciclo para definir la secuencia de las
columnas
int x=0;
for( x ; x < 9 ; x++ )
{
digitalWrite (x, 1) ; //encendemos el pin x
delay(1000);
digitalWrite (x, 0) ; // y apagamos
```

Después de haber terminado con este proyecto nos dimos a la tarea de realizar un seguidor de línea negra con Raspberry pi; lo más sencillo era utilizar arduino.

Pero en nuestras investigaciones nos percatamos de que existía un notable abandono hacia el uso de la Raspberri Pi para trabajar con este tipo de proyecto, armamos el sistema mecánico con múltiples sensores y varios materiales reciclados, nuevamente el software fue desarrollado completamente en C [9].

En cierto momento estos primeros proyectos nos llevaron a un área más especializada, desarrollar aplicaciones web para controlar sin tener que estar exactamente donde se encuentra el dispositivo a controlar. Utilizando el dispositivo Raspberry podemos realizar esta tarea tan interesante.

Se ha de aclarar que no es que nosotros estemos menospreciando el poder de audrino o su facilidad de uso incluso su costo económico si no es sol que estos proyectos posiblemente ya estén con audrino nosotros quisimos hacer un cambio en la movilidad de que audrino es muy usado por varios en proyectos muy buenos pero Raspberry pi ni tiene tanto auge en ese tipo de proyectos podrá ser porque es complicado si nosotros ya nos dimos cuenta de eso pero a un así salimas adelante y tratando de llevar un presupuesto barato por así decirlo lo conseguimos y para nosotros usar Raspberry pi es y será siendo nuestra mejor opción para nosotros que tenga las ventajas de un sistema open source y que maneje varios lenguajes de programación, como su interface física los gpios con su propio librería para ellos es que suficiente para trabajar muy bien con ella y mejor que audrino aparte que audrino solo es un micro controlador y las Raspberry pi es un microprocesador para ciertos usos más avanzados todo es comprar y comprar módulos cuando la Raspberry pi no se le tiene que comprar nada privatizado de ella.

## 7.2 Seguidor línea negra.

El sensor manda un 0 si se encuentra sobre el color negro y manda un 1 si esta sobre el color blanco (fuera de la línea) con esto podemos decir que:

0 = dentro de la línea

1= fuera de la línea

```
if (digitalRead (sensor1)==1)
//sensor 1 salió de la línea negra
```

Con esta condición podremos saber si el sensor esta sobre la línea o no.

```
if (digitalRead (sensor1)==1)
//sensor 1 salió de la línea negra
{
digitalWrite (direccion1, 1) ;
//rectificación de dirección
}
```



Fig.10: Seguidor de línea negra Raspberry pi.

## 7.3 Encendido de una lámpara de 60 watts a distancia.

El proyecto consiste en lograr encender una lámpara remotamente en un ambiente web usando como servidor Raspberry Pi el cual a su vez se encarga de encender una lámpara en el momento en que el cliente lo solicite a través de una página web.



Requisitos previos:

- SO Rasbian
- Librería wiring pi
- Apache
- PHP

Desarrollamos dos aplicaciones en lenguaje c una de ellas pone en alto unos de los pines de Raspberry y la otra lo pone en bajo a ese mismo pin, se podría decir que una de ellas enciende la lámpara mientras la otra aplicación la apaga.

```
//librerías usadas
#include<stdio.h>
#include<WiringPi.h>
main()
{
    if (wiringPiSetup () == -1) return 1 ;
    pinMode (0, OUTPUT) ;
    digitalWrite (0, 1) ; // pin 0 de GPIO
ALTO

    return 0 ;
}
//librerias usadas
#include<stdio.h>
#include<WiringPi.h>
main(){
    if (wiringPiSetup () == -1) return 1 ;
    pinMode (0, OUTPUT) ;
    digitalWrite (0, 0) ; // pin 0 de GPIO
BAJO

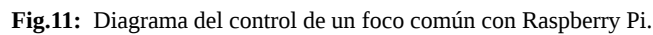
    return 0 ;
}
Encender.c      Apagar.c
```

Las dos aplicaciones en c fueron compiladas y almacenada juntas en una carpeta establecida por apache, en el caso de Raspberry la dirección de la carpeta es /var/www

La aplicación pone en alto el pin 0 pero el voltaje que este provee es insuficiente para encender una lámpara de 30w, para esto se utilizó un relay el cual en nuestro caso se encarga de bloquear un voltaje de 120v hasta que reciba una señal de 5v.

El relay o relevador es un dispositivo electromecánico que funciona como un interruptor controlado por un circuito eléctrico en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes.

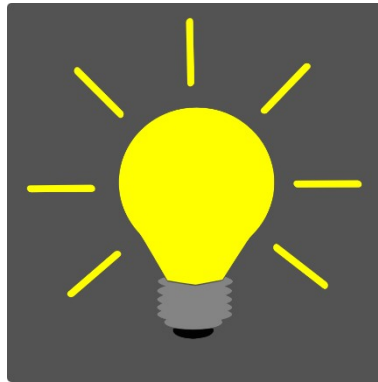
```
<?php exec("./Encender"); ?>
```



```
<html>
<body>
    <?php exec("./Encender");?>
<a href="apagar.php">
    
</a>
</body>
</html>

<html>

<body>
```



**Fig.12:** Encender.php

```
<?php exec("./Apagar");?>  
<a href="encender.php">  
      
</a>  
</body>  
</html>
```



**Fig.13:** Apagar.php

Las vistas PHP son almacenadas en la misma carpeta que las aplicaciones C ya compiladas al igual que las imágenes respectivas para las vistas.

Para la ejecución de la aplicación web es cuestión de entrar mediante el navegador de cualquier dispositivo que esté conectado a la misma red que Raspberry pi y escribir el URL de la misma junto con el nombre de vista a cargar:

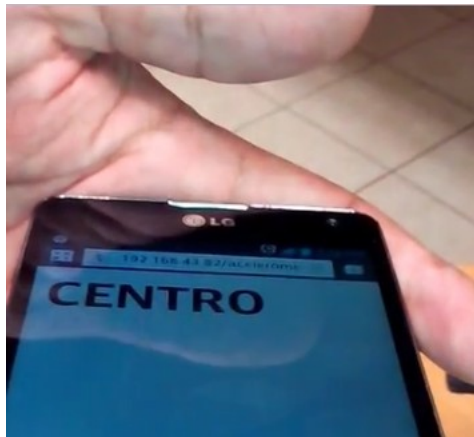
192.168.43.82/apagar.php

Al entrar se muestra la imagen de un foco apagado el cual al hacer clic te direccionara a vista con un foco encendido y por ende la lámpara es encendida al momento que se carga la página web.

#### 7.4 Control de brazo robótico con acelerómetro de celular.

Se desarrolló una aplicación en c la cual recibía un número entero como parámetro:

1 = Arriba  
2 = Abajo  
3 = Derecha  
4 = Izquierda



**Fig.14:** Smarthphone ejecutando aplicación.

Al momento de ejecutar la aplicación con un número como parámetro, la aplicación se encarga de que Raspberry Pi active uno de los cuatro motores del brazo robótico a controlar.

Ejecución mediante terminal:

```
$ sudo ./brazo.c 1 //el parámetro es el 1(unos) el cual indica Arriba
```

Código fuente en c:

```
include<stdio.h>
include<WiringPi.h> //librería de wiringPi utilizar
puerto GPIO

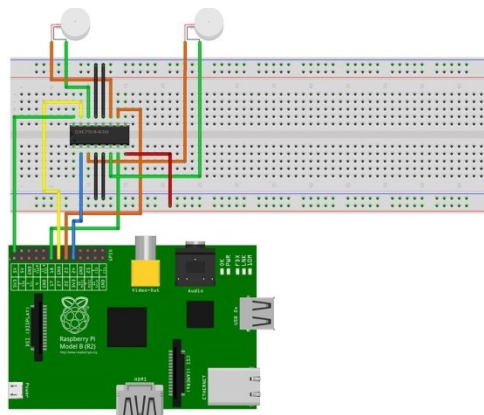
main(parametro){
if (parametro==1) //arriba
digitalwrite(pin1, 1) //pin1 en alto

if (parametro==2) //abajo
digitalwrite(pin2, 1) //pin2 en alto
if (parametro==3) //derecha
digitalwrite(pin3, 1) //pin3 en alto

if (parametro==4) //izquierda
digitalwrite(pin4, 1) //pin4 en alto

if (parametro==0) // pone en bajo todos lo pines
indicándole al brazo robótico que se detenga sea
cual sea su dirección de movimiento
digitalwrite(pin1, 0)
digitalwrite(pin2, 0)
digitalwrite(pin3, 0)
digitalwrite(pin4, 0)}
```

Cada uno de los pines esta físicamente conectado a un puente h de esta manera:



fritzing

**Fig.15:** Diagrama de la conexión del puente H a la Raspberry

Con esto lo que se tiene es, una aplicación que dependiendo el parámetro que reciba activara un motor.

Para interpretar el acelerómetro se desarrolló una aplicación web donde utilizamos Raspberry pi como servidor instalando apache y PHP. De esta forma se logró que no solo se controlara a distancia con el acelerómetro de un solo celular si no que se puede controlar con cualquier dispositivo ya sea celular o tableta que cuente con acelerómetro y sin importar el sistema operativo (android, IOS, BlackBerry) mientras este cuente con explorador web.

Las herramientas de desarrollo web utilizadas fueron PHP, JQuery y Ajax.

JQuery nos permitió obtener los datos del acelerómetro del dispositivo que accediera a la aplicación web, y por medio de un script en JavaScript logramos establecer condiciones en los valores de los ejes x, y.

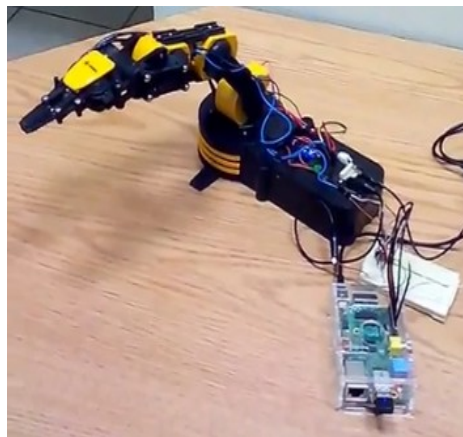
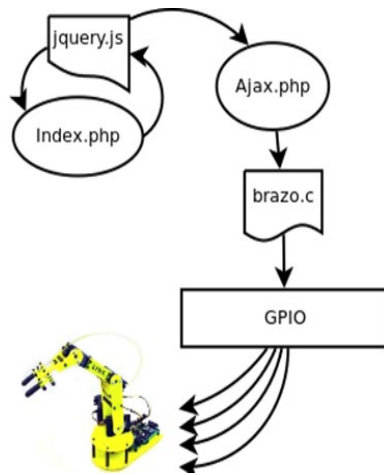
```
$(window).bind("devicemotion", function(e){
    var movitBaby = e.originalEvent,
    accelera =
    movitBaby.accelerationIncludingGravity,
    //valores de acelerometro
    x = accelera.x,
    y = accelera.y;
    direccion(x,y); //metodo que manda las
    variables a un php mediante ajax
});
```

El script que diseñamos se encarga de que cuando el dispositivo este en una posición deseada mande una variable x con la dirección de movimiento mediante Ajax se ejecuta en el servidor un documento php el cual a su vez se encarga de ejecutar y mandar los parámetros a la aplicación anteriormente desarrollada en c.



**Fig. 16:** Brazo robótico controlado desde un Smartphone.

```
<?php
parametros = $_POST['x'];
exec("sudo ./aplicacionC parametros");
?>
```



## 8 Conclusiones

La tecnología sigue evolucionando en cosas más complejas pero a las ves poderosas. Nosotros como estudiantes e investigadores pudimos darnos cuenta realizamos es solo un paso a algo poderoso estamos utilizando un dispositivo poderoso pero existen otros que son complejos y pequeños e igual de poderosos que este que a lo mejor podríamos haber usado pero nuestra meta es ser económicos en el aspecto que somos estudiantes y mantenernos en un manejo electrónico estándar ya que estamos en transcurso de terminar nuestra carrera y no podemos profundizar tanto como se debería, a un que a futuro ya con nuestro tiempo un poco más administrado tenemos pensado estudiar esto adelante así los proyectos que teníamos antes lograr subirlos de nivel solo es cuestión de tiempo un poco de a pollo económico y paciencia e amor al trabajo que se esta realizando.

## Referencias

- [1] Eben Upton and Gareth Halfacree. Raspberry Pi User Guide. Wiley John + Sons (1 de Septiembre de 2012).
- [2] Matt Richardson. Getting Started With Raspberry Pi. Maker Media, Inc.; Edition: 1 (3 de enero de 2013).
- [3] K. Dennis, Andrew. Raspberry Pi Home Automation with Arduino. Packt Publishing (5 de febrero de 2013).
- [4] Linux User and Developer 2013. Has the Raspberry pi met its match? IP imagine publishing.
- [5] Matt Richardson is a Brooklyn-based creative technologist and video producer. Programming the Raspberry Pi: Getting Started with Python. Maker Media, Inc.; Edition: 1 (3 de enero de 2013)
- [6] Brendan Horan. Practical Raspberry Pi. APRESS; Edición: 2013. (27 de marzo de 2013).
- [7] Rs components Raspberry Pi Start Guide. Rs components Vsn 1.0 (2012).
- [8] Dr. Simon Monk. Raspberry pi Cookbook. O'Reilly Media; Edición: 1 (10 de enero de 2014)
- [9] Charles Bell. Beginning Sensor Networks with Arduino and Raspberry Pi. Apress; Edition: 1 (21 de noviembre de 2013).
- [10] Andrew Hague, Karl Wright, Robert Cruse and Paul Kingett. Raspberry Pi Education Manual. Raspberry Pi Foundation.
- [11] Stefan Sjogelid. Raspberry Pi for Secret Agents. PACKT PUBLISHING (25 de abril de 2013).
- [12] Maik Schmidt. Raspberry Pi A Quick Start Guide. Pragmatic Bookshelf; Edición: 1 (6 de noviembre de 2012).

<https://www.dropbox.com/sh/dd863cwrwy0k1yz/r3-YRXFn7L/libros>