



“Responsabilidad con pensamiento positivo”

UNIVERSIDAD TECNOLÓGICA ISRAEL

TRABAJO DE TITULACIÓN

CARRERA: ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES

TEMA: Estudio, diseño e implementación de un prototipo de Entrenador de módulo Raspberry Pi.

AUTOR: Herrera Acosta Eduardo Xavier

TUTOR: Ing. José Robles Salazar Mg.

AÑO 2014

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Graduación certifico:

Que el proyecto integrador de carrera **“Estudio, diseño e implementación de un prototipo de Entrenador de módulo Raspberry Pi.”**, presentado por el Sr. Eduardo Xavier Herrera Acosta, estudiante de la carrera de Electrónica y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D.M. Agosto del 2014

TUTOR

.....

ING. JOSÉ ROBLES SALAZAR MG.

UNIVERSIDAD TECNOLÓGICA ISRAEL

AUTORÍA DE TRABAJO DE TITULACIÓN

El abajo firmante, en calidad de estudiante de la Carrera de Electrónica y Telecomunicaciones, declaro que los contenidos de este Trabajo de Titulación, requisito previo a la obtención del Grado de Ingeniería en Electrónica y Telecomunicaciones, son absolutamente originales, auténticos y de exclusiva responsabilidad legal y académica del autor.

Quito D.M. Agosto del 2014

.....

Eduardo Xavier Herrera Acosta

CC: 1716792591

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL TRIBUNAL DE GRADO

Los miembros del Tribunal de grado, aprueban el Proyecto Integrador de Carrera para la graduación de acuerdo con las disposiciones reglamentarias emitidas por la Universidad Tecnológica Israel para Títulos de Pregrado.

Quito D.M. Agosto del 2014

Para constancia firma:

TRIBUNAL DE GRADO

.....

PRESIDENTE

.....

MIEMBRO 1

.....

MIEMBRO 2

DEDICATORIA

El proyecto está dedicado en primer lugar a Dios, que es el pilar fundamental en mi vida y que la voluntad de Él se ve reflejada día a día, a mis padres que fueron quienes me apoyaron y en especial a mi hija María Paz que es la fuerza para conseguir mis logros y deseos.

AGRADECIMIENTO

Agradezco primeramente a Laservision Cía. Ltda. quienes me permitieron realizar el presente proyecto, a mis profesores que han sabido educarnos de la mejor manera.

ÍNDICE

APROBACIÓN DEL TUTOR	I
AUTORÍA DE TRABAJO DE TITULACIÓN	II
APROBACIÓN DEL TRIBUNAL DE GRADO	III
DEDICATORIA	IV
AGRADECIMIENTO	V
ÍNDICE	VI
ÍNDICE DE TABLAS	XII
INTRODUCCIÓN	1
OBJETIVOS	3
Objetivo general.....	3
Objetivos específicos.....	3
CAPÍTULO I.....	4
FUNDAMENTACIÓN TEÓRICA.....	4
1.1 Introducción.....	4
1.2 Marco teórico.....	4
1.2.1 Raspberry Pi	4
1.2.2 Leds	5
1.2.3 Displays de 7 segmentos	5
1.2.4 Buzzer o zumbador	5
1.2.5 Decodificador de 7 segmentos	5
1.2.6 Suiche pulsadores.....	5
1.2.7 Teclado matricial.....	5
1.2.8 Triac	5
1.2.9 Relé de contacto seco.....	6
1.2.10 Reloj RTC.....	6
1.2.11 Módulos RF	6
1.2.12 Pantallas LCD 2x16.....	7
1.2.13 Transistor de potencia	7

1.2.14 Arduino.....	7
1.2.15 Shields Arduino	7
1.2.16 Motores	8
1.2.17 Servomotor.....	8
1.2.18 Dipsuiches.....	8
1.2.19 Focelda	8
1.2.20 Sensor de temperatura LM35.....	8
1.2.21 Lenguaje de programación Phyton.....	8
1.2.22 Raspbian Wheezy	9
1.2.23 Win32DiskManager.....	9
1.2.24 Advance IP Scanner	9
1.2.25 X-Window.....	9
1.2.26 Tarjeta SD	10
1.2.27 Salida de video HDMI.....	10
1.2.28 Librerías Phyton	11
1.2.29 Puerto USB.....	11
1.3 Marco conceptual.....	12
1.3.1 Comparación de microcontroladores y Raspberry.....	12
1.3.2 Comparación entre diferentes placas de desarrollo.....	12
CAPÍTULO II.....	14
BREVE DESCRIPCIÓN DEL PROCESO INVESTIGATIVO REALIZADO.....	14
CAPÍTULO III.....	18
PRESENTACIÓN DE RESULTADOS	18
3.1 Introducción	18
3.2 Propuesta de solución al problema planteado.....	18
3.2.1 Diseño del equipo.....	18
3.2.1 Zona de leds	19
3.2.2 Zona de displays	20
3.2.3 Zona Motores	21

3.2.4 Zona de Servomotor.....	22
3.2.5 Zona de Buzzer.....	23
3.2.6 Zona de suiches pulsadores.....	23
3.2.7 Zona de dipsuiches	24
3.2.8 Zona de RTC (Real Time Clock).....	24
3.2.9 Zona de relé.....	25
3.2.10 Zona de triac	26
3.2.11 Zona de transistor de potencia	26
3.2.12 Zona de pantalla LCD	27
3.2.13 Zona Arduino.....	28
3.2.14 Zona de teclado matricial	29
3.2.15 Zona fotocelda	30
3.2.16 Zona temperatura.....	30
3.2.17 Zona RF	31
3.2.18 Circuito impreso del equipo	31
3.2.19 Funcionamiento del equipo	33
3.2.20 Lista de materiales	34
3.3 Implementación del equipo	34
3.4 Pruebas de funcionamiento.....	41
3.4.1 Software para probar las zonas del entrenador para Raspberry.....	42
3.4.1.1 Encendido intermitente.....	42
3.4.1.2 Generación de PWM.....	43
3.4.1.3 Suiche y led.....	44
3.5 Análisis de resultados	44
3.5.1 Análisis de resultados de compilación / carga vs tiempo	44
CONCLUSIONES Y RECOMENDACIONES	46
Conclusiones	46
Recomendaciones.....	46
Bibliografía.....	47

ANEXOS.....49

ÍNDICE DE FIGURAS

Figura 1.1 Raspberry Pi.....	4
Figura 1.2 Módulos RTC.....	6
Figura 1.3 Módulos RF	6
Figura 1.4 Pantalla lcd 2x16.....	7
Figura 1.5 Interfaz Phyton para linux	9
Figura 1.6 Polarización de SD para PIC.....	10
Figura 1.7 Pin out Para Micro SD.....	11
Figura 1.8 Conectores USB	11
Figura 3.1 Diagrama de Bloques	18
Figura 3.2 Zona de leds.....	20
Figura 3.3 Zona de displays.....	21
Figura 3.4 Zona de Motores.....	22
Figura 3.5 Zona de Servomotor	22
Figura 3.6 Zona de Buzzer	23
Figura 3.7 Zona de suiches pulsadores	23
Figura 3.8 Zona de dipsuiches.....	24
Figura 3.9 Zona de RTC	25
Figura 3.10 Zona de relé.....	26
Figura 3.11 Zona de triac	26
Figura 3.12 Zona de transistor de Potencia	27
Figura 3.13 Zona de pantalla lcd.....	28
Figura 3.14 Zona Arduino	29
Figura 3.15 Zona de Teclado Matricial	29
Figura 3.16 Zona fotocelda	30
Figura 3.17 Zona Temperatura.....	30
Figura 3.18 Zona RF.....	31
Figura 3.19 Ruteado y screen del pcb	32

Figura 3.20 Vista en 3D de pcb.....	32
Figura 3.21 Vista cobre de pcb	33
Figura 3.22 Transferencia térmica.....	35
Figura 3.23 Taladrado del PCB	35
Figura 3.24 Taladrando el PCB	36
Figura 3.25 Taladrando el PCB	37
Figura 3.26 WIn32 Disk 1	38
Figura 3.27 WIn32 Disk 2	38
Figura 3.28 Iniciando sesión con XLaunch.....	39
Figura 3.29 Pantalla del escritorio de Raspbian Wheezy	39
Figura 3.30 Pantalla configuración Raspberry Pi.	40
Figura 3.31 Pantalla de Actualización de Software “update”	41
Figura 3.32 Pines GPIO para Raspberry Revisión 1 y 2	41
Figura 3.33 Pin GPIO.17 configurado como salida	42

ÍNDICE DE TABLAS

Tabla 1.1 Comparación entre Raspberry y Microcontroladores.....	12
Tabla 1.2 Comparación entre diferentes placas para sistema de desarrollo.....	13
Tabla 3.3 Resultado de pruebas en zona de leds.....	43
Tabla 3.4 Resultado de pruebas generación PWM.....	43
Tabla 3.5 Resultado de pruebas suiche y led.....	44
Tabla 3.6 Resultados de compilación/carga vs tiempo.....	45

INTRODUCCIÓN

La empresa Laservision Cía. Ltda. ubicada en la ciudad de Quito, en las calles Inglaterra 512 y Vancouver, se dedica a la venta de material electrónico y desarrollo de hardware y software desde hace 20 años a nivel del Ecuador. En los inicios la compañía Laservisión Cia. Ltda se dedicaba a la importación, venta y distribución de la línea Cekit. Como actividad paralela a la distribución de esta línea, la compañía Laservisión se dedica a la venta al por menor de material electrónico, semiconductores entre otros. Este campo que cubre la empresa sirve para dar apoyo a la línea principal que era la distribución de kits electrónicos importados, ya que en muchas ocasiones se veía la necesidad de dar un servicio técnico al cliente, y en otras se requería de material electrónico ya sea para la venta o reparación de algún circuito electrónico.

Estas actividades transcurren sin problemas por 10 años seguidos, pero debido al fallecimiento del Gerente principal de la Editorial Cekit la empresa fué dividida, pero sin tener éxito. Las publicaciones de cursos, revistas, fascículos no fueron actualizadas, no hubo actualización de la misma calidad de antes, no surgieron kits electrónicos nuevos.

Debido a estos acontecimientos surge la necesidad de realizar importaciones de material electrónico y distribuirlo al por mayor a nivel nacional. La compañía se ve en la necesidad de montar un departamento de investigación. Aquí se prueba, analiza, y concluye cómo es la forma correcta de operar los nuevos elementos. En este departamento nace la idea de replicar la forma de trabajo que tenía la Editorial Cekit; comenzando a elaborar kits electrónicos (elementos, placas e información) en un sólo paquete.

Llegó un momento en que el hardware libre comienza a sentirse a nivel mundial, debido a la globalización no tardó en llegar a Ecuador. La primera vez que se veía un producto de hardware y software libre a nivel de aprendizaje y diseño electrónico se dio con las placas Arduino de procedencia Italiana. Esta placa incorpora en su interior un programador de microcontroladores AVR de conexión USB, capaz de programar el microcontrolador que lleva incorporado en ella. El lenguaje de programación utilizado es C. El departamento de investigación fue el encargado de desarrollar aplicaciones, ver factibilidad de trabajo, arrojando como resultado que la integración electrónica y las aplicaciones que se podían hacer con estas placas, superaba al trabajo de usar microcontroladores PIC o AVR de la forma tradicional.

La tecnología a nivel mundial fue avanzando y salió al mercado la tarjeta Raspberry PI, según el departamento de investigación de Laservisión Cia. Ltda. estas placas van

a dar solución a muchos problemas existentes en el desarrollo de software y hardware que tiene al momento, y que todavía no se han podido solucionar o van a dar un mejor funcionamiento en ciertas aplicaciones.

Actualmente en el mercado del campo electrónico no existe un módulo entrenador para placas Raspberry Pi con características de poder albergar microcontroladores e incluso módulos de la línea Arduino; tampoco existe un módulo que contenga periféricos de entrada/salida para poder implementar las prácticas realizadas con dicha placa.

OBJETIVOS

Objetivo general

- Estudiar, diseñar e implementar un prototipo de entrenador electrónico para manejar la placa Raspberry Pi basados en hardware y software libre, para facilitar el uso de diferentes periféricos de entrada y salida.

Objetivos específicos

- Estudiar los requisitos específicos que son necesarios para usar el módulo Raspberry Pi.
- Diseñar un prototipo de módulo entrenador para Raspberry Pi.
- Implementar un prototipo de módulo entrenador para Raspberry Pi
- Validar el prototipo módulo entrenador para Raspberry Pi.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

Para la fabricación del Entrenador de módulos Raspberry Pi es necesario tener en cuenta algunos conceptos básicos que servirán para el estudio y diseño del Software y Hardware que utilizará el presente trabajo. Los elementos electrónicos empleados servirán para configurarlos tanto como periféricos de entrada o salida para la placa Raspberry Pi y poder realizar diferentes tipos de pruebas. El entrenador está dividido en 17 zonas para diferentes usos.

1.2 Marco teórico

1.2.1 Raspberry Pi

Raspberry Pi es una plataforma de desarrollo para productos o proyectos electrónicos; tiene incorporado en su estructura de hardware puertos USB, conector HDMI, conector de salida de video, puerto para comunicaciones TCP/IP, salida de audio de 3.5mm, slot para tarjeta SD, procesador de 700Mhz entre otros.

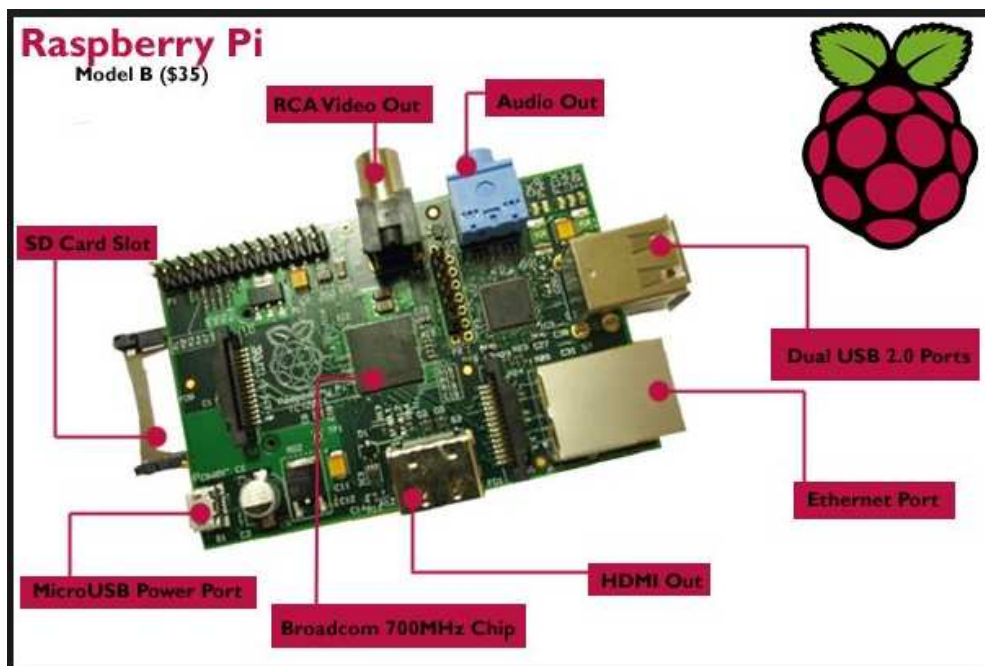


Figura 1.1 Raspberry Pi.

Fuente: (Raspberry, 2104)

1.2.2 Leds

Led es un semiconductor que emite luz cuando se le hace circular una corriente entre ánodo y cátodo. Esta luz se genera por la ionización de un gas en su interior, el color del led está en función del gas. La corriente que circula va desde los 10 mA hasta los 30mA. dependiendo si es de bajo o alto brillo (Reyes, 2008, pág. 48)

1.2.3 Displays de 7 segmentos

El display de 7 segmentos es un conjunto ordenado de leds capaz de conformar un número entre el cero y nueve al momento de encenderse ciertos segmentos del display. Existen 2 clases, dependiendo la conexión interna de los leds se dividen en cátodo común y ánodo común. (Corrales, 2006)

1.2.4 Buzzer o zumbador

El zumbador o buzzer es un dispositivo de características piezoeléctricas o de bobina que al momento de ser polarizado emite un sonido que sirve de alerta sonora para algún evento. Existen de diferentes tipos de voltaje, van desde los 3.3VDC hasta los 24 VDC, tienen un consumo de corriente relativamente bajo va en el orden de los 20 – 30 mA.. (Corrales, 2006, pág. 89)

1.2.5 Decodificador de 7 segmentos

El decodificador BCD es un dispositivo electrónico capaz de convertir una palabra de 4 bits en la entrada y entregar un código decimal para conectar un display de 7 segmentos. (Corrales, 2006, pág. 56)

1.2.6 Suiche pulsadores

Los suiches pulsadores son láminas aceradas que tienen la capacidad de deformarse cuando existe una pequeña presión sobre la superficie. El contacto óhmico cuando está presionado si son metálicos es de 0 Ω , por el contrario si son suiches de película de carbono el contacto sería de unas décimas de ohmio. (Fuente Investigador)

1.2.7 Teclado matricial

El teclado matricial es una matriz de pulsadores colocados conformando filas y columnas con el objeto de disminuir el número de pines necesarios para la conexión con microcontroladores o sistemas digitales. Existen teclados de suiche de lámina metálica, de tinta conductora o de carbono. (Penagos, 2010, pág. 88)

1.2.8 Triac

El triac es un semiconductor que pertenece a la familia de elementos de control (tiristores), prácticamente un triac es la unión de dos tiristores conectados en paralelo pero en sentido opuesto, compartiendo la misma compuerta. (Unicrom, 2012)

1.2.9 Relé de contacto seco

Un relé es un dispositivo electromecánico que posee una bobina en su interior que al ser polarizada genera un campo electromagnético, el cual sirve para hacerlo trabajar como imán y atraer unas platinas que hacen la función de suiche. Este suiche puede ser normalmente abierto o normalmente cerrado. (Corrales, 2006)

1.2.10 Reloj RTC

Un reloj RTC viene de las siglas Real Time Clock o traducido al español Reloj de Tiempo Real, es un circuito integrado que tiene la capacidad de generar una señal de reloj para obtener una frecuencia capaz de ser igual al tiempo calendario y proporcionarnos la fecha y hora real a través del protocolo de comunicaciones I2C. (Monk, 2013, pág. 292)

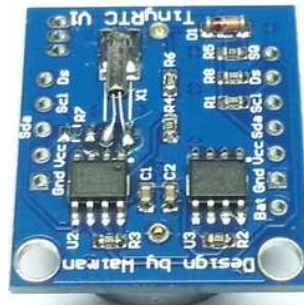


Figura 1.2 Módulos RTC

Fuente: (Arduino, Arduino, 2014)

1.2.11 Módulos RF

Son dispositivos que están sintonizados a cierta frecuencia portadora en el orden de los 300-433Mhz y enviar datos seriales a través de forma inalámbrica. Funcionan a 5 VDC y soportan comunicaciones desde los 2400-9600 bpps (Corrales, 2006, pág. 136)



Figura 1.3 Módulos RF

Fuente: Investigador

1.2.12 Pantallas LCD 2x16

La pantalla lcd 2x16, es un periférico de salida, sirve para desplegar caracteres alfanuméricos mediante una comunicación paralela de 4 u 8 bits. Posee un pin RS, un pin E (enable), luz posterior o backlight a base de un led de alto brillo en su interior. Los compiladores actuales para su funcionamiento trabajan a base de librerías. (Penagos, 2010, pág. 40)



Figura 1.4 Pantalla lcd 2x16

Fuente: Investigador

1.2.13 Transistor de potencia

Los transistores son semiconductores o componentes de estado sólido de tres pines, emisor, base, colector. Existen de 2 clases NPN o PNP, su forma de polarización difiere. Manejan diferentes tipos de corrientes dependiendo la numeración, cuando se requiere manejar corrientes superiores a 600 mA se denominan transistores de potencia, cambian el encapsulado y pueden manejar corrientes de hasta 15 A. (Unicrom, Unicrom Electronica, 2012)

1.2.14 Arduino

Las placas Arduino son módulos de software y hardware libre, posee librerías propias que abrevian muchísimo el trabajo para el programador. Existen diferentes modelos que varía según la capacidad de memoria, número de pines de entrada y salida, entre otros. (Banzi, 2011, pág. 17)

1.2.15 Shields Arduino

La zona de shields para Arduino, es un espacio en el que se pueden insertar la gran mayoría de placas Arduino, la intención es que el Entrenador sea universal. (Arduino, 2014). Los Shields de Arduino sirven para trabajar sobre módulos ya construidos que tienen una aplicación específica, por ejemplo el Shield GSM tiene la función exclusiva de comunicarse con un chip de telefonía celular. El Shied LCD + Botones, tiene la función exclusiva de comunicarse con un Arduino y desplegar caracteres alfanuméricos y a la vez tiene suiches pulsadores de propósito general.

1.2.16 Motores

Motor es una máquina eléctrica o transductor que transforma energía eléctrica en energía mecánica mediante las interacciones electromagnéticas. Existen motores eléctricos que son reversibles, pueden transformar energía mecánica en energía eléctrica funcionan como generadores.

1.2.17 Servomotor

Es un conjunto de elementos electrónicos y electromecánicos, traen consigo una red de engranajes conformando una caja de desmultiplicación que se controla a través de tensión y corrientes muy bajas por medio de pulsos que se pueden generar con un microcontrolador. Dependiendo la fuerza que se necesite varían desde los 1.5 kg hasta los 30Kg torque. (Corrales, 2006, pág. 169)

1.2.18 Dipsuiches

Es un conjunto de suiches deslizantes que vienen encapsulados en una sola unidad. Pueden ser de 2, 3, 4, 8,12 según la aplicación que se requiera. La corriente que manejan estos suiches es muy limitada, su funcionamiento es mas para el uso de control electrónico más no para potencia. (Penagos, 2010, pág. 26)

1.2.19 Fococelda

La fotocelda es una resistencia que varía su valor óhmico en función de la luz que incide sobre la superficie. La polarización se la hace generalmente como un divisor de tensión con ayuda de una resistencia fija. (Monk, 2013, pág. 301)

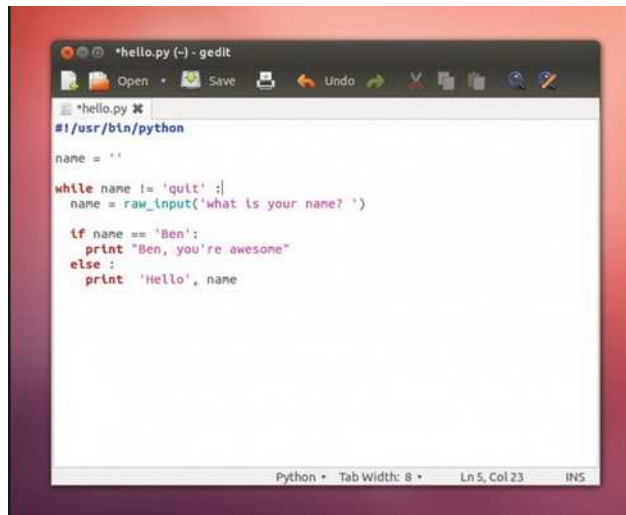
1.2.20 Sensor de temperatura LM35

El LM35 es un circuito integrado que tiene la capacidad de variar su voltaje de salida en función de la temperatura. Su polarización es con 5 VDC y en el pin de salida existe una variación de 10mV por cada grado centígrado (Reyes, 2008, pág. 171)

1.2.21 Lenguaje de programación Phyton

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma. (Admin, 50 ebooks gratis en Phyton, 2014)

Python es un lenguaje de programación muy popular, hoy en día entre los programadores, sobre todo para aquellos que se dedican a la programación de Raspberry Pi porque es software y hardware libre aprobada por OSI.. (Admin, 50 ebooks gratis en Phyton, 2014)



```
*hello.py (-) - gedit
#!/usr/bin/python
name = ''
while name != 'quit' :|
    name = raw_input('what is your name? ')
    if name == 'Ben':
        print "Ben, you're awesome"
    else :
        print 'Hello', name
Python - Tab Width: 8 - Ln 5, Col 23 INS
```

Figura 1.5 Interfaz Phyton para linux

Fuente: (Beauford Court, 2014)

1.2.22 Raspbian Wheezy

Es un software de libre descarga, es una versión de Linux usada para el trabajo en una tarjeta Raspberry Pi (Raspberry Pi Foundation, 2014)

1.2.23 Win32DiskManager

Es un programa de libre descarga del internet que sirve para grabar la imagen binaria de un sistema operativo y almacenarla en un disco duro, tarjeta SD o memoria flash que se alojará posteriormente en un CPU o en una tarjeta Raspberry Pi. (Canonical LTDA, 2014)

1.2.24 Advance IP Scanner

Es un software de libre descarga que utiliza un protocolo de comunicación llamado SSH, permite mantener o enlazar comunicación tanto de ida como de vuelta mediante una red local donde se puede conectar computadores o una tarjeta Raspberry Pi. (Famatech, 2014)

1.2.25 X-Window

X- Window es un software que se lo consigue en la red internet, sirve para acceder desde la plataforma Windows a otros sistema operativos conectados en una red de área local por medio del protocolo de comunicaciones TCP/IP, funciona conjuntamente con el programa Putty (Putty.ORG, 2014) y el software cliente llamado X-Window. (Dice, 2014) De esta manera se puede acceder al escritorio de otro sistema operativo.

1.2.26 Tarjeta SD

Son dispositivos de almacenamiento masivo que se ha popularizado su uso en celulares, cámaras o cualquier otro artefacto electrónico. Según el fabricante y la necesidad existe desde 1G hasta los 32G de capacidad de memoria. Una variación de esta tecnología son las memorias micro SD, compatibles pin a pin a las tarjetas SD pero se debe utilizar un adaptador para poder insertar en la ranura de alojamiento de la memoria. La polarización electrónica para la tarjeta SD es similar para cualquier dispositivo que lea o escriba datos. (Ladelec, 2014)

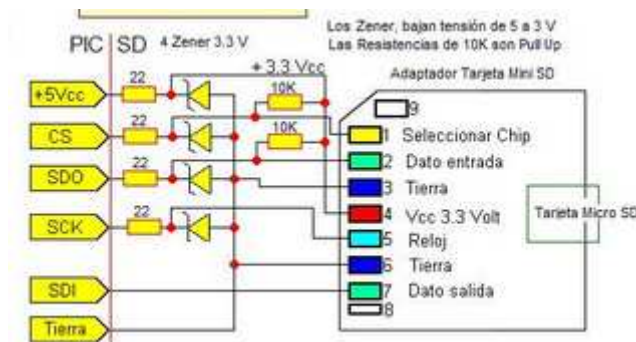


Figura 1.6 Polarización de SD para PIC

Fuente: (Ladelec, 2014)

1.2.27 Salida de video HDMI

High-Definition Multimedia Interface o HDMI (interfaz multimedia de alta definición) es la norma de audio y vídeo digital cifrado sin compresión construida por la industria para que sea sustituya al conector RCA. HDMI dispone una interfaz entre cualquier fuente de audio y vídeo digital como podría ser un DVD, un TDT, Blu-ray, un Tablet PC, un ordenador, un Raspberry (Microsoft Windows, Linux, Apple Mac OS X, etc.) o un receptor A/V, y monitor de audio/vídeo digital compatible, como un televisor digital (DTV).

HDMI permite usar el vídeo computarizado, mejorado o en HD, así como audio digital multicanal en un sólo cable. Es independiente de los varios estándares DTV como ATSC, DVB (-T,-S,-C), que son encapsulaciones de datos del formato MPEG. Cuando se envía a un decodificador, se obtienen los datos de vídeo sin comprimir, pueden ser en HD. Los datos se codifican en formato TMDS para transmitirlos digitalmente por medio de conector HDMI. Dicho formato incluye 8 canales de audio digital sin compresión. Después de la versión 1.2, HDMI puede utilizar hasta 8 canales de audio de un bit. El audio de 309 bit se usa en los Super audio CD.

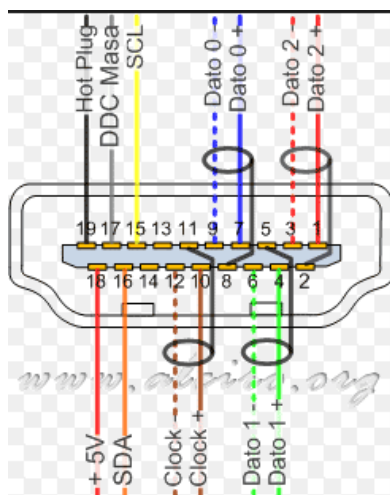


Figura 1.7 Pin out Para Micro SD
Fuente: (Alciro, 2014)

1.2.28 Librerías Phyton

Adafruit es un fabricante y proveedor de componentes electrónicos así como de complementos y accesorios. Ofrece gratuitamente librerías creadas en lenguaje Python para poder interactuar con todos sus dispositivos, así como multitud de ejemplos para que se pueda realizar pruebas y sistemas de desarrollo. Todo el código fuente distribuido es software libre aunque en el mayor de los casos con limitaciones. La instalación de esta librería es por medio del internet y comandos tal como si se está trabajando en el sistema operativo Linux. (Admin, Adafruit actualiza sus librerías en Python para la Raspberry Pi, 2014)

1.2.29 Puerto USB

Las siglas significa ("Universal Serial Bus") ó su traducción en español es línea serial universal de transporte de datos. Es un conector metálico del tipo rectangular de 4 pines que permite la transmisión de datos entre una gran gama de dispositivos periféricos con la computadora; debido a esto es considerado puerto, se considera también como una evolución del puerto serial tradicional (Db-09) ya que funciona a velocidades sumamente más elevadas. (Informatica Moderna, 2014)

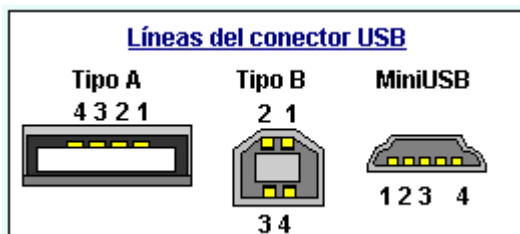


Figura 1.8 Conectores USB
Fuente: (Informatica Moderna, 2014)

1.3 Marco conceptual

1.3.1 Comparación de microcontroladores y Raspberry

A continuación la tabla 1.1 muestra la comparación entre Raspberry Pi y otros microcontroladores

TECNOLOGÍA	PRECIO	CONEXIÓN TCP/IP	PUERTOS USB	SALIDA DE VIDEO	SERVIDOR WEB
AVR	9.00 (Promedio)	NO	NO	NO	NO
PIC	9.00 (Promedio)	NO	NO	NO	NO
RASPBERRY PI	70.00	SI	SI	SI	SI

Tabla 1.1 Comparación entre Raspberry Pi y Microcontroladores

Fuente: Investigador

La tabla 1.1 muestra que la Placa Raspberry Pi trae incorporado en el hardware cualidades que si se desea implementar con cualquier otro microcontrolador se requerirá fabricar un hardware adicional para llegar a tener los mismos recursos.

Características Raspberry Pi

- 2 entradas USB
- Un jack tipo Rj-45 para conexión de Red (internet)
- Conector HDMI
- Salida de audio
- Salida de video compuesto
- Ranura para tarjetas SD, que hace la función de disco duro.
- Procesador central (CPU)ARM1176JZF-S a 700 MHz

1.3.2 Comparación entre diferentes placas de desarrollo

En la tabla 1.2 se compara tres tipos de módulos para desarrollo de proyectos electrónicos.

Placa Electrónica	CPU	Precio en Exterior	Video	Sistema operativo
Raspberry Pi	700Mhz	35	HDMI/Video RCA	Linux
PcDuino	1 Ghz	45	HDMI	Linux, Android, Cloud9 IDE on Node.js w/ BoneScript
BeagleBone Black	1 Ghz	59	HDMI	Linux Android

Tabla 1.2 Comparación entre diferentes placas para sistema de desarrollo

Fuente: Investigador

La tabla 1.2 arroja como resultado que la placa Raspberry Pi es la mejor opción debido al precio y por tener 2 opciones de salida de video, ya que en algunos circuitos es necesario instalar permanentemente un televisor con conector RCA.

CAPÍTULO II

BREVE DESCRIPCIÓN DEL PROCESO INVESTIGATIVO REALIZADO

Del problema Investigado como primera etapa del proceso investigativo para este proyecto, se tiene como problema principal que la empresa Laservision Cía Ltda. no posee un módulo entrenador para placas Raspberry Pi con características de poder albergar microcontroladores e incluso módulos de la línea Arduino, tampoco existe un módulo que contenga periféricos de entrada/salida para poder implementar las prácticas realizadas con dicha placa.

Para el desarrollo del Entrenador se propuso un objetivo general plateado en la **INTRODUCCIÓN** del presente informe con el fin de implementar un módulo que permita elaborar circuitos o proyectos electrónicos a través de un Raspberry Pi.

Los objetivos específicos fueron propuestos con el fin de desarrollar el proyecto e investigar todos los componentes necesarios para su implementación.

La hipótesis o idea a defender del presente proyecto de titulación se plantea de la siguiente manera: si se implementa un Entrenador para módulos Raspberry Pi, se reducirá notablemente el tiempo de ensamble y pruebas al momento de realizar aplicaciones. Teniendo como variable independiente: la implementación del entrenador para módulos Raspberry Pi y como variable dependiente: la reducción del tiempo de ensamble y pruebas al momento de realizar aplicaciones.

En el proyecto de titulación se utiliza la tecnología a base de Raspberry Pi para la construcción del Entrenador, esta tecnología permitirá desarrollar proyectos electrónicos de mayor complejidad.

El proyecto fue realizado con métodos de investigación de acuerdo a las cuatro etapas de desarrollo como sigue:

Se utilizó el método de análisis y síntesis para recopilar toda la información necesaria y útil para el desarrollo del Entrenador.

El método experimental se aplicó en la Implementación y realización de pruebas para validar el funcionamiento del Entrenador para módulos Raspberry Pi.

En este proceso investigativo se utilizó la entrevista como técnica investigativa para la recopilación de cierta información necesaria para el desarrollo del proyecto. A

continuación se presenta el formato con el cual se efectuó las entrevistas, en el anexo 1 se puede observar el resultado de las entrevistas.

ENTREVISTA AL PERSONAL DE LASERVISIÓN CIA. LTDA

Nombre:

Cargo:

Fecha:

1. ¿Qué opina usted sobre la implementación del nuevo entrenador para placas Raspberry Pi?

Bueno

Malo

Porque

.....
.....
.....

2.- ¿Cree Ud. que el Entrenador agilizará el desarrollo de prototipos electrónicos?

SI

NO

Porque

.....
.....
.....

3.- ¿Cree ud. que el Entrenador será un producto que a futuro se lo pueda comercializar en la empresa?

SI

NO

Porque

.....

.....
.....
4.- ¿Sabe cómo poner en funcionamiento el entrenador?

SI

NO

Porque

.....
.....
.....

5.- ¿Cree Ud. que el entrenador servirá para la fabricación de nuevos proyectos electrónicos?

SI

NO

Porque

.....
.....
.....

A continuación se detallan los resultados de la entrevista realizada.

Pregunta 1

El resultado de la entrevista indica que el 100% de las personas opinan que es buena la implementación del nuevo entrenador para placas Raspberry Pi.

Pregunta 2

El 100% de la personas entrevistadas opinan que si se agilizará el tiempo de implementación de prototipos y proyectos electrónicos.

Pregunta 3

El 100% de las personas entrevistadas opinan que el entrenador será un producto que a futuro se lo pueda comercializar en la empresa.

Pregunta 4

No todas las personas saben cómo poner en funcionamiento el entrenador

Pregunta 5

El 100% de las personas opinan que el entrenador servirá para la fabricación de nuevos proyectos electrónicos

Se espera que una vez que se haya implementado el entrenador, la empresa Laservisión Cía. Ltda. logre investigar y generar proyectos electrónicos más complejos para dar soluciones en mas campos de aplicaciones que requieran el uso de tecnología con hardware y software.

CAPÍTULO III PRESENTACIÓN DE RESULTADOS

3.1 Introducción

La elaboración del prototipo de Entrenador de módulo Raspberry Pi, demanda el uso de elementos electrónicos configurados como entrada o salida dependiendo la aplicación o proyecto electrónico que se vaya a implementar, conjuntamente el software realizado sobre la plataforma Phyton debe concordar con el hardware ensamblado.

3.2 Propuesta de solución al problema planteado

3.2.1 Diseño del equipo

Las zonas que se pueden implementar en el entrenador pueden ser ilimitadas, pero el presupuesto y precio final del entrenador debe ser accesible al usuario, por ello se ha limitado a 17 zonas de acuerdo a lo establecido en el plan del proyecto integrador de carrera. A continuación se presenta el diagrama de bloques de las zonas del entrenador, figura 3.1

ZONAS DEL ENTRENADOR RASPBERRY PI

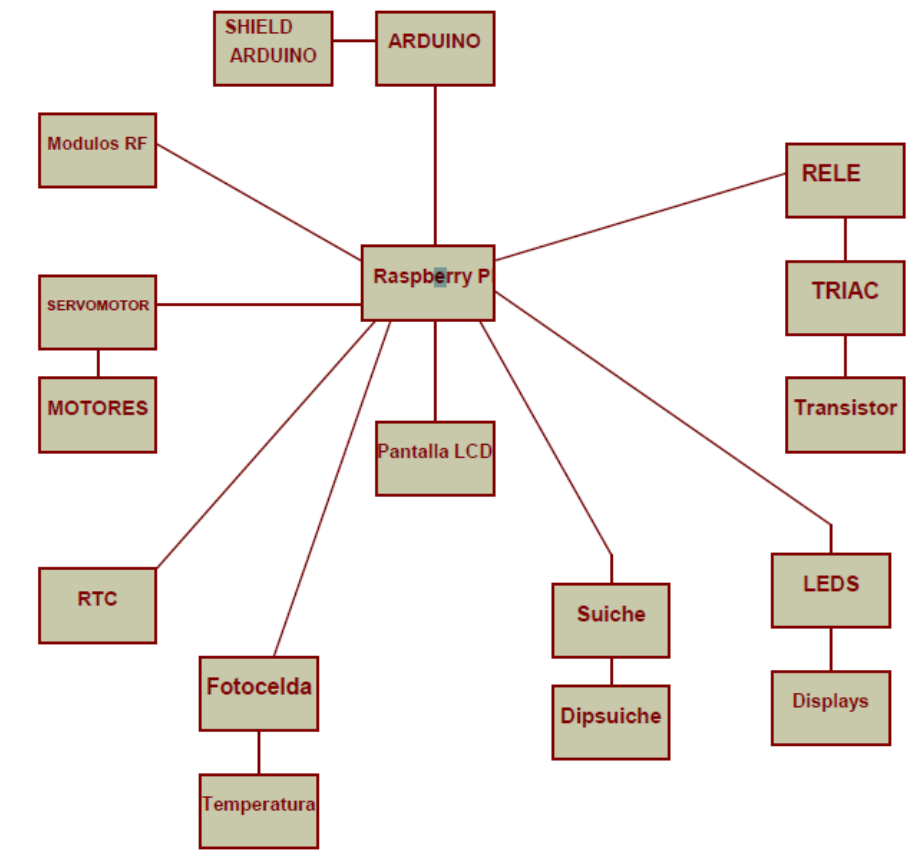


Figura 3.1 Diagrama de Bloques

Fuente: Proteus – Investigador

El diagrama en bloques de la figura 3.1 representa las diferentes zonas que tiene el entrenador, se tiene las siguientes:

- Zonas de leds
- Zona de displays
- Zona de motores
- Zona de servomotor
- Zona buzzer
- Zona suiches pulsadores
- Zona dipsuiche
- Zona RTC
- Zona relé
- Zona triac
- Zona transistores
- Zona LCD
- Zona Arduino
- Zona teclado
- Zona fotocelda
- Zona temperatura
- Zona RF

3.2.1 Zona de leds

Esta zona está constituida por 8 diferentes leds, todos los cátodos están unidos entre sí. Los ánodos poseen una resistencia de 330Ω como limitadora de corriente (figura 3.2).

Los 8 pines GPIO (General Propouse In Out) de la tarjeta Raspberry se configuran como salida, por lo tanto se tiene dos niveles de voltaje, un alto o un bajo, 5 Vdc o Gnd, dependiendo como esté estructurado el programa.

Con esta zona del entrenador se puede probar un secuenciador de luces, sirve también como leds monitores en el caso de requerir alguna señal informativa de un determinado proceso. Puede servir también como un periférico de salida al momento de mostrar el resultado de una operación en código binario.

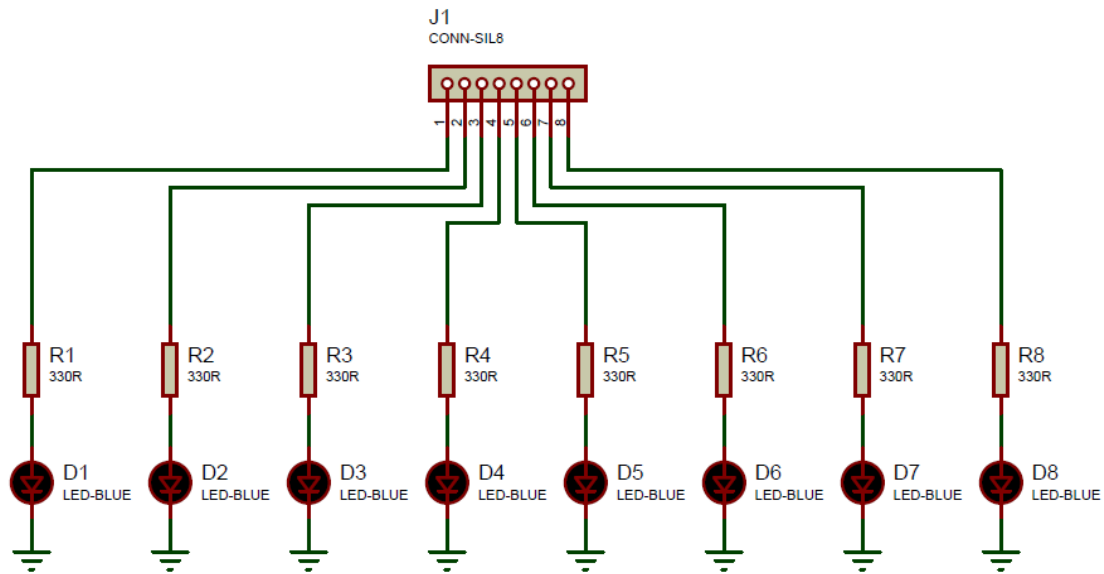


Figura 3.2 Zona de leds

Fuente: Proteus – Investigador

3.2.2 Zona de displays

Esta zona está usando 4 displays de ánodo común, esto quiere decir que el punto común debe estar polarizado a V+. Para ello se utiliza la polarización de 4 diferentes transistores para cada display. La base del transistor posee una resistencia de 4.7K Ω de polarización, el emisor directamente a tierra y la carga del colector es el punto común del display (V+). El transistor a usar es un PNP 2N3906 debido a que suministra una corriente de 300mA, suficiente para manejar la corriente total por displays. Por lo tanto para habilitar el funcionamiento de cada display se debe poner un nivel bajo en la resistencia de polarización de base (figura 3.3).

Cada segmento del display A, B, C, D, E, F y G llevan una resistencia de 330 Ω como limitadora de corriente. Para encender cada segmento se debe poner un nivel bajo en el extremo de la resistencia.

Se puede utilizar 8 pines GPIO de la tarjeta Raspberry para visualizar o desplegar un número en el display.

Existe otra opción para usar los displays, para ello se utiliza un conversor BCD el 74LS47, las 7 salidas de este circuito integrado se las conecta directamente a los diferentes segmentos del display desde la A hasta la G. La entrada del conversor BCD se maneja con una palabra de 4 bits, proveniente de los pines GPIO. Esta opción se la utiliza cuando se desee realizar un barrido y usar los 4 displays a la vez.

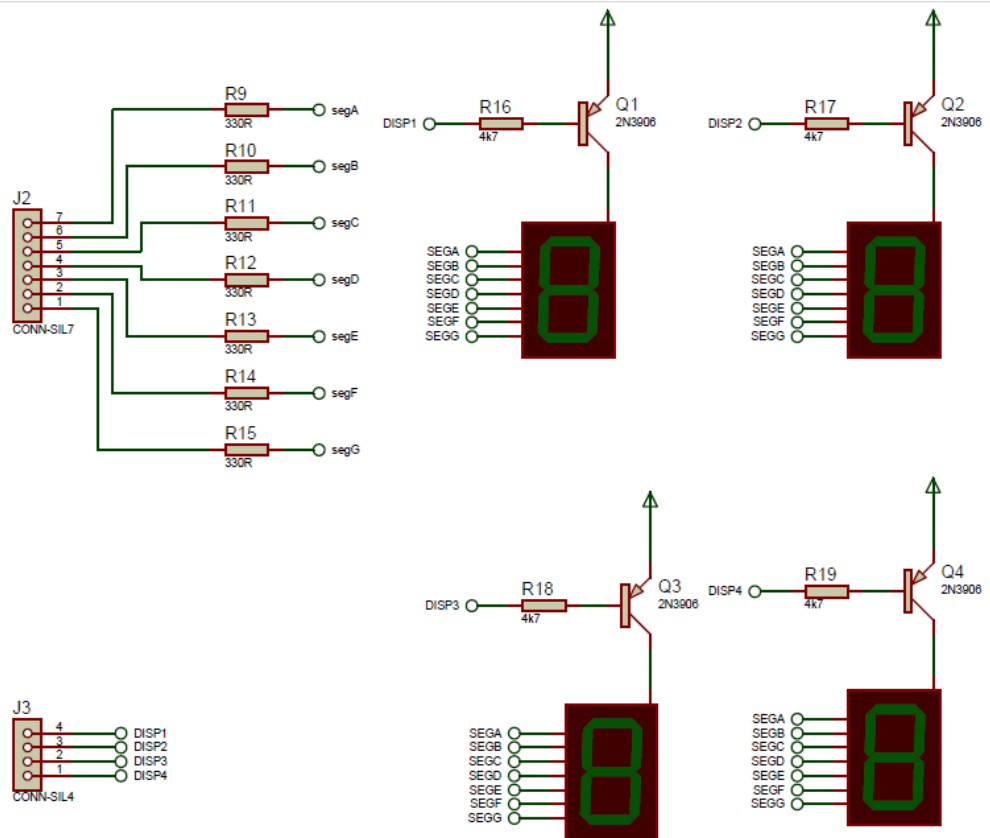


Figura 3.3 Zona de displays

Fuente: Proteus – Investigador

3.2.3 Zona Motores

Esta zona está determinada por un puente H, se utiliza un circuito integrado el L293D. Este elemento tiene diodos de protección incluidos en su interior. La fuente de alimentación está dividida en dos, la de control y la de potencia. Los habilitadores están unidos directamente a V+. Tiene 4 salidas y 4 entradas. Las salidas sirven para conectar directamente el motor PAP, puede ser un bipolar, unipolar o incluso para realizar la prueba con motores DC. Las 4 entradas las maneja la tarjeta Raspberry a través de sus pines GPIO (figura 3.4).

Se puede realizar pruebas para el desplazamiento y giro del motor para aplicaciones de brazos robóticos, puertas corredizas o circuitos inversores de giro para motores dc.

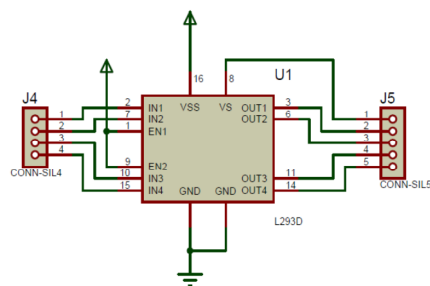


Figura 3.4 Zona de Motores

Fuente: Proteus – Investigador

3.2.4 Zona de Servomotor

Esta zona determina el funcionamiento de cualquier servomotor con polarización de 5Vdc, Gnd y un pin de control o datos. Debido a que las características de hardware de los servomotores que se consiguen en el mercado son similares, se puede unificarlos para realizar pruebas incluso si son de diferentes marcas. La capacidad de torque que posee cada servomotor está determinada por el motor y principalmente por la caja de engranajes que lleva internamente. Existen servomotores de 3.5kg, 7kg, 30kg o más dependiendo la aplicación o su función. (figura 3.5).

La ventaja es que las conexiones electrónicas son las mismas, sólo varía el programa, específicamente el tren de pulsos aplicado en el pin de control.

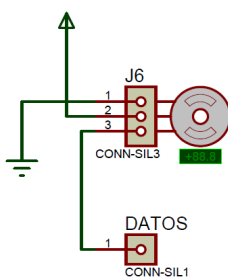


Figura 3.5 Zona de Servomotor

Fuente: Proteus – Investigador

3.2.5 Zona de Buzzer

Esta zona está constituida por un zumbador o chicharra de 5 VDC, la función principal es proporcionar a sistemas una alerta sonora de baja potencia. Sirve de alerta auditiva cuando por ejemplo se trabaja con teclados matriciales o suiches pulsadores para dar un aviso a través de un “beep” que indica que la tecla o botón fué presionado (Figura 3.6).

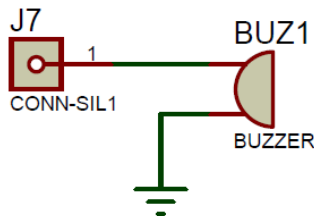


Figura 3.6 Zona de Buzzer

Fuente: Proteus – Investigador

3.2.6 Zona de suiches pulsadores

La zona de suiches es un periférico de entrada, para ello se utiliza resistencias de polarización de 4.7KΩ conectadas directamente desde V+ hasta una entrada o GPIO de la tarjeta Raspberry, en este mismo punto está colocado un microsuiche con un lado a tierra, para conseguir dos estados, un alto o bajo dependiendo si se presiona el pulsador (figura 3.7).

Esta zona sirve para realizar pruebas de simulación de sensores, ingreso de datos, botones de aumento o disminución de alguna variable, o cualquier otra aplicación en la que se requiera ingresar algún dato hacia la tarjeta Raspberry Pi.

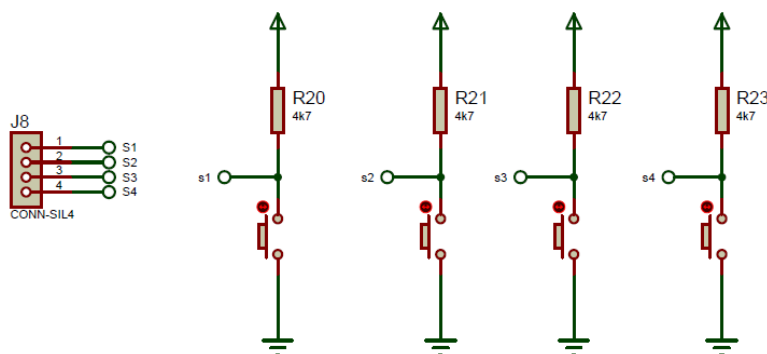


Figura 3.7 Zona de suiches pulsadores

Fuente: Proteus – Investigador

3.2.7 Zona de dipsuiches

Esta zona es muy similar a la anterior, se pueden realizar las mismas pruebas incluso la forma de polarizar con la resistencia de $4.7K\Omega$ a cada suiche y conseguir un nivel lógico alto o bajo, se vuelve a repetir en esta zona, la diferencia radica en que ahora son suiches deslizantes. Su uso está establecido por ejemplo cuando se requiere poner una palabra de 8 bits y dejarla estática (figura 3.8).

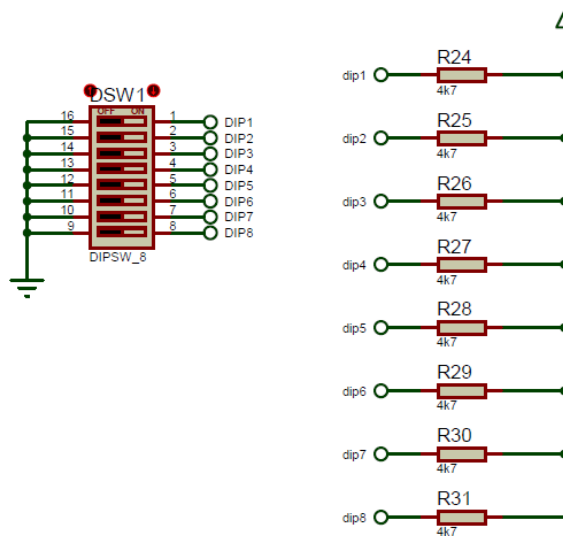


Figura 3.8 Zona de dipsuiches

Fuente: Proteus – Investigador

3.2.8 Zona de RTC (Real Time Clock)

Los relojes de tiempo real o RTC son los encargados de proporcionarnos la fecha y hora, específicamente se trabaja con el DS1307, es un circuito integrado que se comunica a través del protocolo de comunicaciones I2C. Los pines SCL y SDA son los pines de reloj y datos respectivamente, están polarizados con una resistencia de $4.7K\Omega$ a $V+$. Posee una salida de onda cuadrada, medio segundo en nivel alto y medio segundo en nivel bajo, este pin se lo conoce con el nombre de Vout. Para respaldar la hora posee un pila modelo CR2032 de 3VDC (figura 3.9), esta entra a funcionar cuando el entrenador no está conectado a la red eléctrica o por el puerto USB.

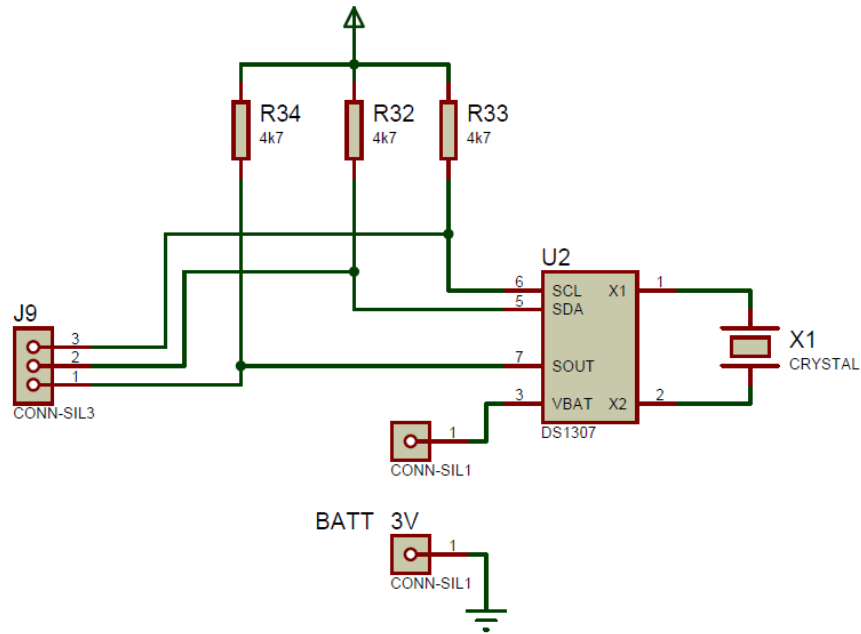


Figura 3.9 Zona de RTC

Fuente: Proteus – Investigador

3.2.9 Zona de relé

Esta zona está enfocada a manejar una carga de potencia, conmutada a través del suiche interno de un relé. Con la salida del relé se puede manejar voltajes y corrientes elevadas en relación a la tarjeta Raspberry. Se utiliza un relé de 12VDC para polarizar la bobina interna. Los contactos del suiche soportan hasta 220VAC/VDC, 10 amperios. Este dispositivo puede manejar el encendido de un foco de 115VAC. La señal de activación proviene de los pines GPIO del Raspberry que es aplicada a una resistencia de 4.7KΩ que polariza la base de un transistor NPN 2N3904 (figura 3.10), el emisor directamente a tierra y como carga de colector la bobina del relé, adicional un diodo 1N4007 que sirve para protección contra la fuerza contra-electromotriz almacenada en la bobina del relé al momento de la desconexión.

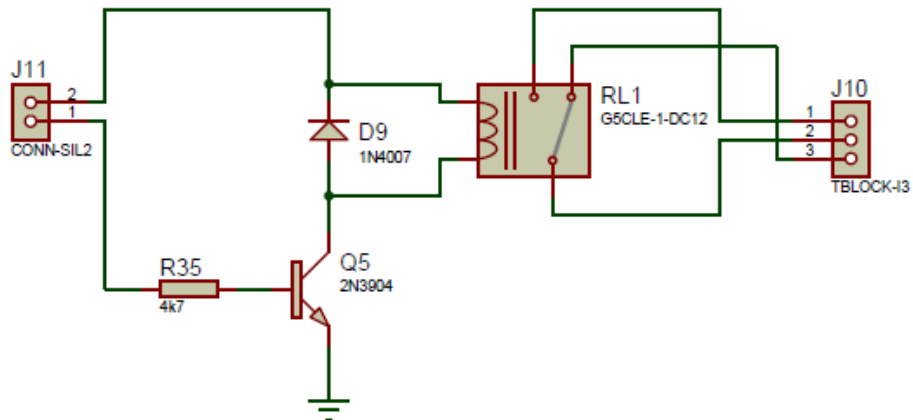


Figura 3.10 Zona de relé

Fuente: Proteus – Investigador

3.2.10 Zona de triac

Siguiendo con la conmutación de cargas de potencia, se tiene un relé de estado sólido a base de un triac BT138. Este dispositivo permite conmutar cargas, por ejemplo de 115VAC sin el característico chispazo de los contactos secos (suiches del relé) evitando así el desgaste.

Este componente electrónico para su polarización necesita de un optoacoplador de numeración MOC3031, es un opto diac va a conducir por avalancha. La señal se obtiene de un pin GPIO de la tarjeta Raspberry y se aplica al led interno del optoacoplador a través de una resistencia de 560 Ω (figura 3.11).

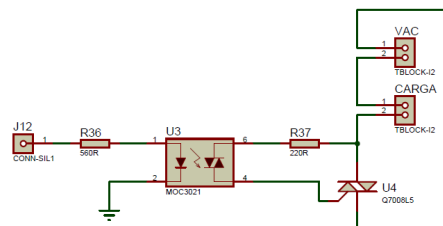


Figura 3.11 Zona de triac

Fuente: Proteus – Investigador

3.2.11 Zona de transistor de potencia

Otra forma de conmutar cargas de potencia pero en esta ocasión para manejo de corriente directa puede ser un transistor, la numeración con la que se va a trabajar es el TIP122, entre sus características principales está el poder

controlar hasta 8 amperios en colector. Una aplicación puede ser el uso de PWM para controlar la luminosidad de focos o incluso la velocidad de un motor DC. La señal se la toma de algún pin GPIO y se la inyecta a la base a través de una resistencia de $1K\Omega$, la carga va en paralelo al diodo 1N4007. Puede ser carga inductiva o resistiva (figura 3.12).

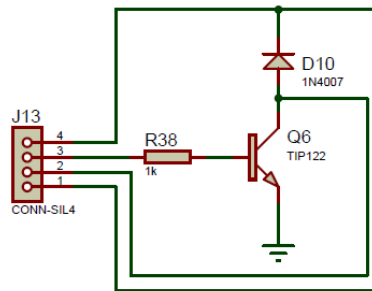


Figura 3.12 Zona de transistor de Potencia

Fuente: Proteus – Investigador

3.2.12 Zona de pantalla LCD

Esta zona es un periférico de salida, aquí se puede visualizar mensajes, leyendas, estados de algún evento. La pantalla que se usa es una del tipo LCD 2x16, de características de 2 filas por 16 columnas, alimentación de 5 VDC, backlight o luz posterior a través de un led blanco de alto brillo, que se debe conectar a V+ a través de una resistencia de $10\ \Omega$ como limitadora de corriente (figura 3.13).

Los pines de comunicación de la pantalla hacia la tarjeta Raspberry son el RS, E y la palabra de 4 bits D0, D1, D2 y D3.

Las librerías que posee el compilador tienen la capacidad de desplegar un mensaje en la pantalla lcd con una sola instrucción que se ingresa en el código fuente. La forma y manejo de estas pantallas son muy similares a las de dimensiones 4x20.

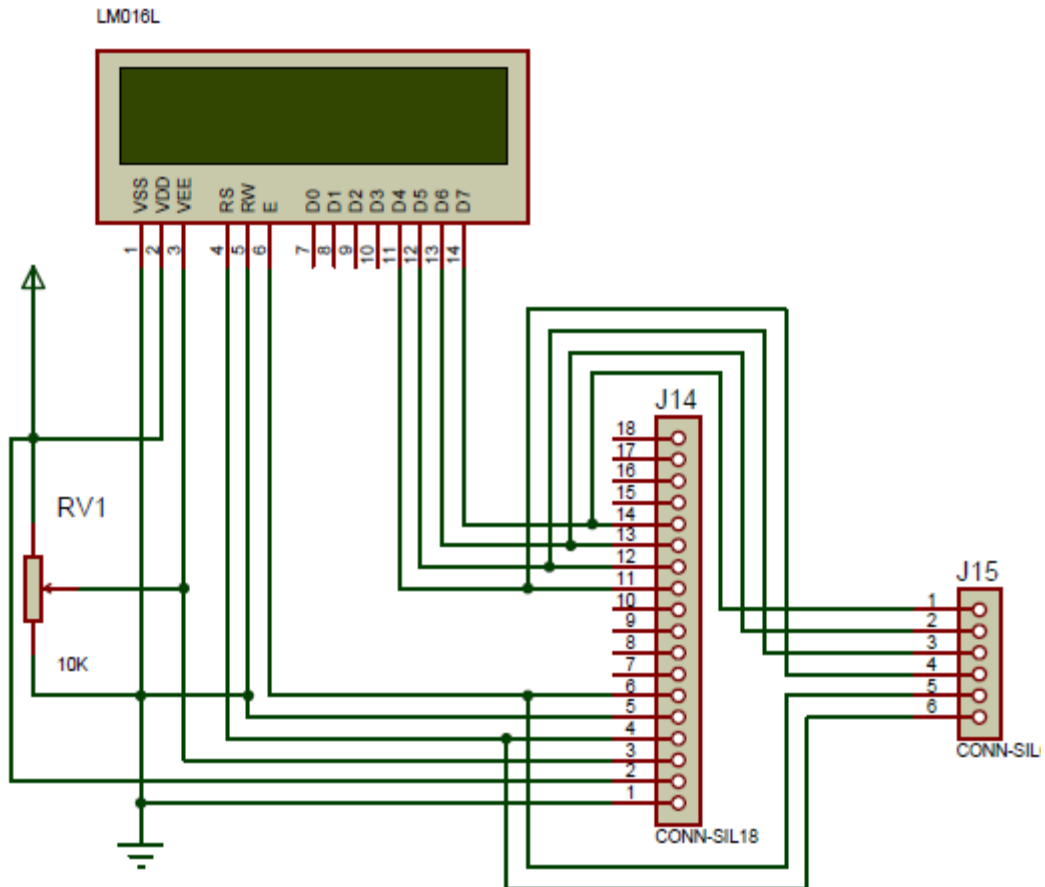


Figura 3.13 Zona de pantalla lcd
Fuente: Proteus – Investigador

3.2.13 Zona Arduino

Arduino es una plataforma para desarrollo de hardware y software, es una tarjeta electrónica que incorpora en su interior un programador de microcontroladores AVR y un ATMEGA328 (microcontrolador a programar por medio del lenguaje C). Es un producto de libre distribución ya que es de hardware y software libre (figura 3.14).

En el Entrenador de Raspberry se denomina zona Arduino debido a que posee un zócalo compatible para Shield de Arduino Uno. Se tiene la necesidad de incluir el trabajo con los Shields ya que en la actualidad existe un gama amplia de modelos, por ejemplo shield para comunicaciones con Wifi, Ethernet, Modem GSM, Modem GPS, pantallas GLCD (Gráficas), GLCD touch, sensores de varios propósitos entre otros.

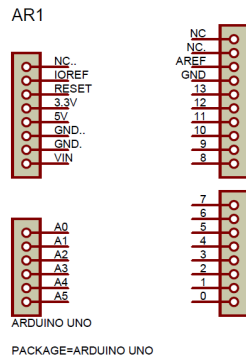


Figura 3.14 Zona Arduino
Fuente: Proteus – Investigador

3.2.14 Zona de teclado matricial

Esta zona es enfocada al uso de un teclado matricial, el funcionamiento de este teclado se lo hace por medio de un barrido de filas y columnas, todo esto de forma sincronizada y por medio del software se reconoce cual tecla es presionada. En el mercado existen teclados 3x4 que quiere decir 3 columnas por 4 filas, también hay los 4x4, cuatro filas por cuatro columnas. La estructura o material también los diferencian, existen los que tienen suiches pulsadores de contacto seco, los que poseen suiches con contacto de película o caucho conductor, los del tipo de membrana como los que se usan en los microondas. La forma de polarizarlos es conectar las columnas a V+ a través de una resistencia de $4.7K\Omega$ y las filas directamente al microcontrolador (figura 3.15).

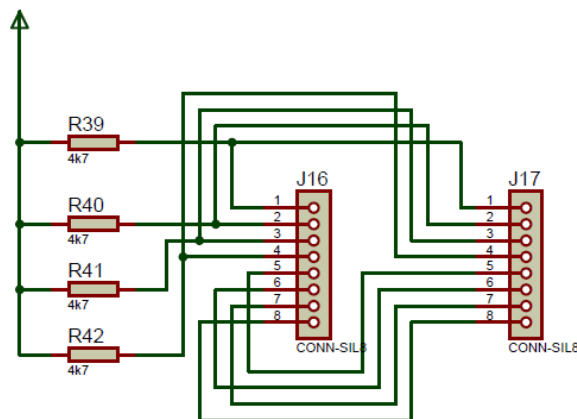


Figura 3.15 Zona de Teclado Matricial
Fuente: Proteus – Investigador

3.2.15 Zona fotocelda

Esta zona está conformada básicamente por un divisor de tensión , una resistencia de 1KΩ hacia V+, y el otro extremo hacia un lado de la fotocelda, el otro extremo de la fotocelda a tierra. Con este circuito se consigue variar el voltaje en el punto de unión de la resistencia y la fotocelda dependiendo la cantidad de luz que incida sobre la superficie de la fotocelda (figura 3.16).

Este circuito sirve por ejemplo como medidor de la cantidad de luz en un determinado ambiente. Sirve también para realizar pruebas con el conversor análogo digital de la tarjeta Raspberry Pi.

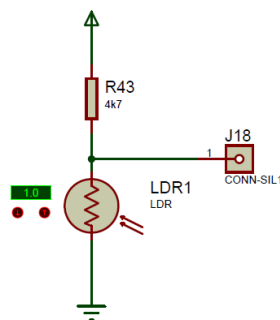


Figura 3.16 Zona fotocelda

Fuente: Proteus – Investigador

3.2.16 Zona temperatura

La zona temperatura posee un sensor de denominación LM35, es un circuito integrado incorporado en el encapsulado de un transistor, posee tres pines V+, GND y datos. El pin de datos entrega un voltaje proporcional a la temperatura, 10mV por cada grado centígrado. Es directamente proporcional a más temperatura más voltaje. Los rangos de medición van desde -50 a +150°C (figura 3.17).

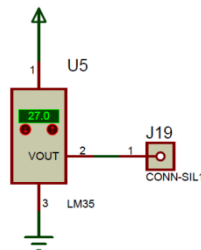


Figura 3.17 Zona Temperatura

Fuente: Proteus – Investigador

3.2.17 Zona RF

Esta zona es la encargada la comunicación inalámbrica serial, con ella se puede transmitir datos seriales inalámbricamente hacia el receptor, de esta manera se evita cables. Funcionan con 5 voltios, es comunicación serial asincrónica, se puede optar por velocidades de 2400, 4600, 9600, 115000 entre otras. La antena según el fabricante de los módulos debe ser un alambre de 16cm de largo (figura 3.18).

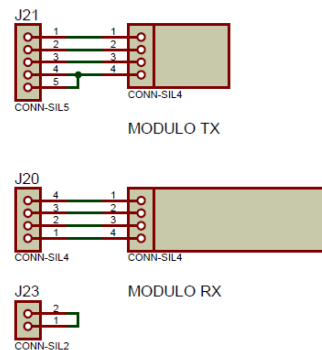


Figura 3.18 Zona RF

Fuente: Proteus – Investigador

3.2.18 Circuito impreso del equipo

El pcb se realizó en el software Proteus, se utilizó fibra de vidrio debido a la alta resistencia de temperatura ya que la placa se fabrica por proceso de transferencia térmica y con ayuda de una plancha de gran consumo de corriente.

La figura 3.19 muestra el ruteado y screen del pcb.

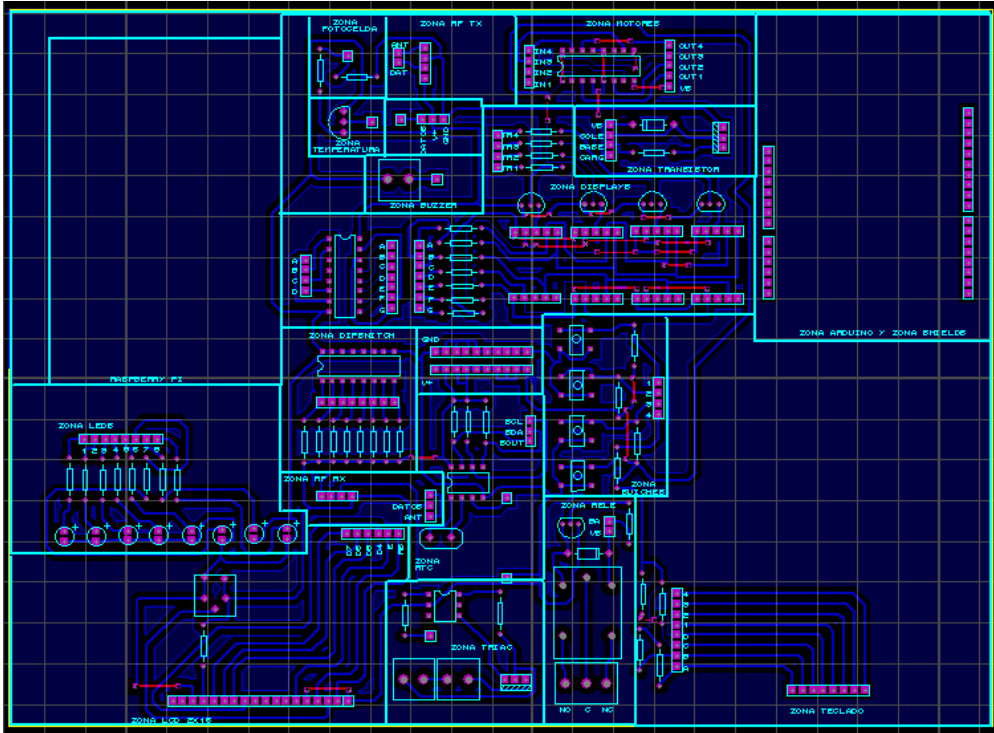


Figura 3.19 Ruteado y screen del pcb

Fuente: Proteus – Investigador

La figura 3.20 muestra el pcb en formato de 3D generado con el software Proteus, esto nos da una idea de las dimensiones de altura que posee cada elemento utilizado, además sirve para tener una visión de cómo va a quedar el pcb una vez ensamblado.

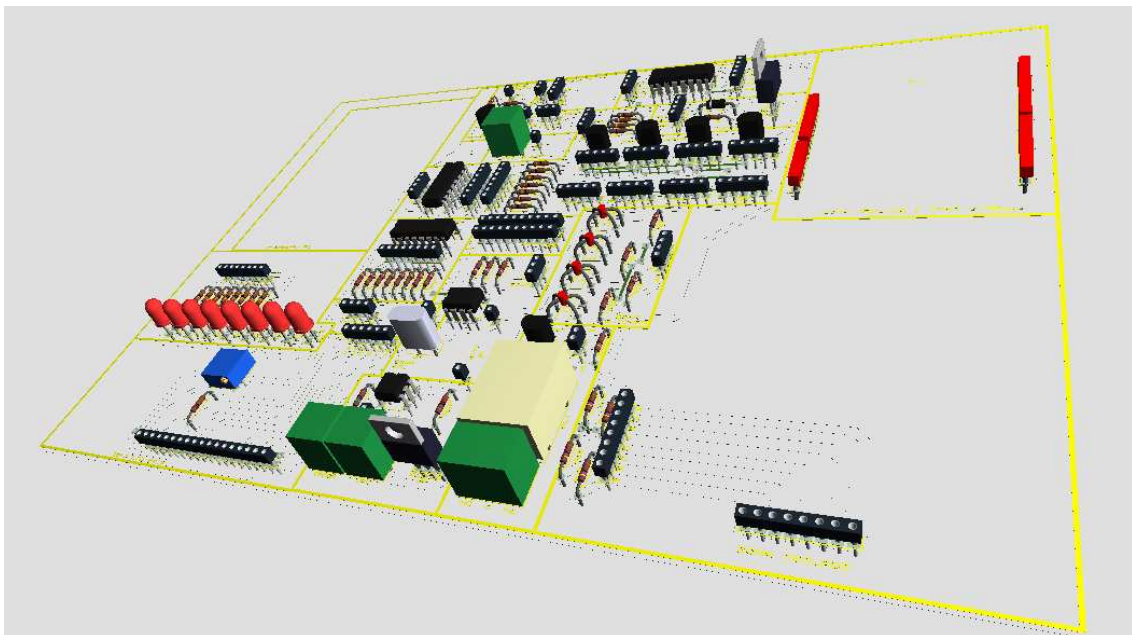


Figura 3.20 Vista en 3D de pcb

Fuente: Proteus – Investigador

La figura 3.21 permite visualizar las pistas desde la parte de abajo del entrenador, se utilizó un relleno de espacios libres para reducir el uso de cloruro férrico al momento de quemar la placa.



Figura 3.21 Vista cobre de pcb

Fuente: Proteus – Investigador

3.2.1 9 Funcionamiento del equipo

Para poner en funcionamiento el entrenador es necesario seguir los siguientes pasos:

- Desconectar la fuente de alimentación
- Implementar el diagrama esquemático sobre el entrenador.
- Utilizar cable utp o de protoboard para unir las entradas/salidas de las diferentes zonas del entrenador hacia la tarjeta Raspberry.
- Si va a utilizar un televisor conecte la salida video hacia la tarjeta Raspberry Pi (conector RCA), en el caso de usar un PC conecte la tarjeta Raspberry Pi a un router de la red de área local.

- Conectar la fuente de alimentación e ingresar al escritorio de la tarjeta Raspberry Pi.
- Realizar el programa en la plataforma Phyton
- Compilar y cargar el programa
- Si el programa está correctamente elaborado se podrá observar el resultado sobre el hardware del entrenador.

3.2.20 Lista de materiales

- Placa de circuito impreso (fibra de vidrio)
- Conectores hembra maquinados
- Resistencias (4.7K Ω , 10k Ω , 330 Ω , 220 Ω)
- Condensadores cerámicos (0.01uf, 0.1uf)
- Condensadores electrolíticos (10uf, 100uf, 470uf)
- Transistores (2N3904, 2N3906, TIP122)
- Zócalos para circuitos integrados (16,14, 18, 8 pines)
- Leds de varios colores de 5mm
- Tarjeta Raspberry Pi Rev. B
- Tarjeta Arduino Rev. 3
- Circuitos integrados (74LS47, DS1307, L293D)
- Cables de conexión
- Fuente de poder 5VDC 2A.

3.3 Implementación del equipo

Para la elaboración de la placa se utilizó el método de transferencia térmica, por lo tanto se imprimió el diseño de pistas en una hoja de transferencia.

A continuación la figura 3.25 muestra la transferencia térmica con una plancha, la placa de fibra de vidrio está recubierta sobre una tela gruesa.



Figura 3.22 Transferencia térmica

Fuente: Investigador

La figura 3.26 muestra un taladro de banco que permite alojar brocas finas exclusivas para el uso de perforaciones para componentes electrónicos, todavía no se retira el tóner transferido ya que esto ayuda para saber donde se tiene que realizar los respectivos huecos.

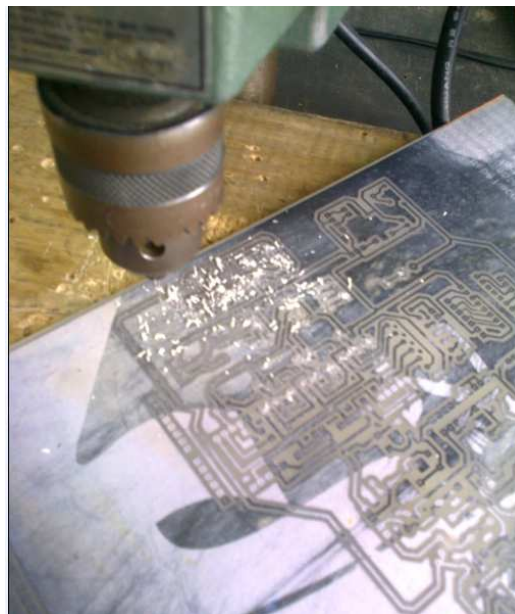


Figura 3.23 Taladrado del PCB

Fuente: Investigador

La figura 3.27 muestra desde el lado del screen como va quedando la placa teniendo ya un 60% de perforaciones realizadas.

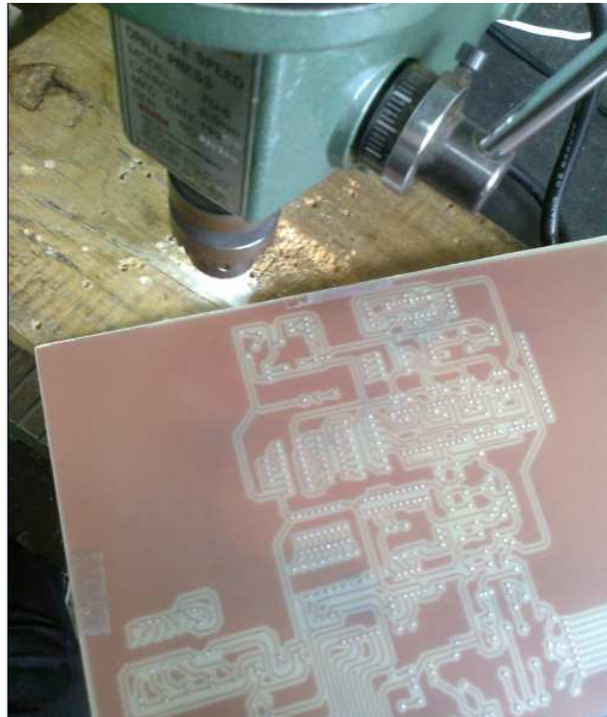


Figura 3.24 Taladrando el PCB

Fuente: Investigador

Una vez concluido las perforaciones se realiza la impresión del screen ya que es más fácil centrar tomando como referencia todos los huecos realizados.

La figura 3.28 muestra el screen ya transferido sobre la placa de fibra de vidrio.

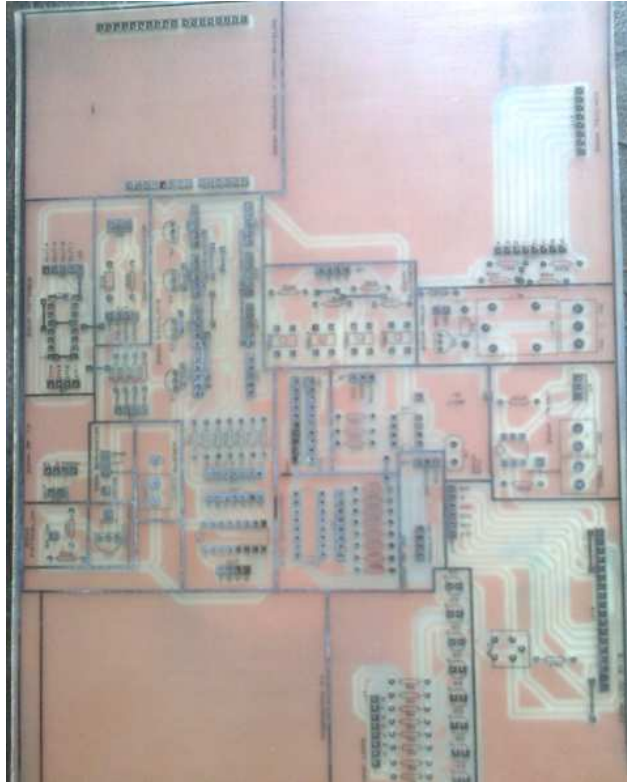


Figura 3.25 Taladrando el PCB

Fuente: Investigador

A continuación se explicará cómo poner en funcionamiento el software, para ello se necesita descargar del internet los siguientes programas:

- WIn32DiskManager
- XLaunch Configuration
- Raspbian Wheezi

Una vez descargado todos los programas hay que instalarlos, para ello se procede de la siguiente manera:

Se inserta la memoria SD en el PC, se abre la aplicación WIn32DiskManager, saldrá la siguiente imagen:

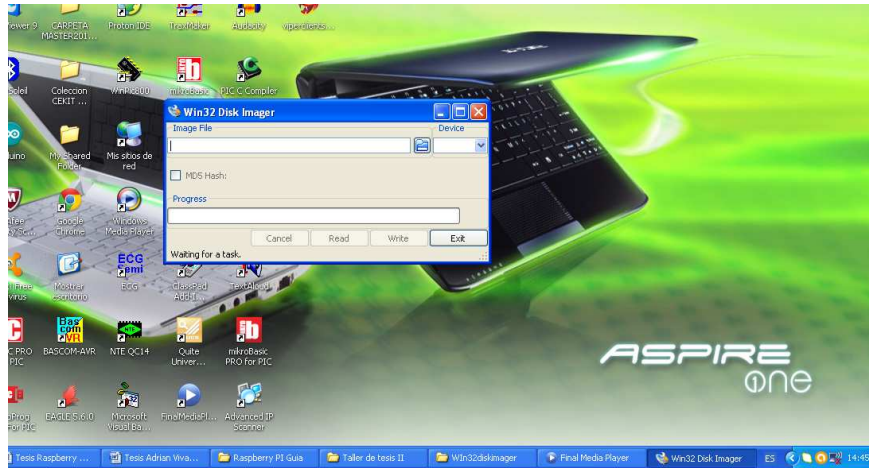


Figura 3.26 WIn32 Disk 1

Fuente: Investigador

En esta pantalla se debe seleccionar la unidad de memoria SD, sirve para escribir sectores en una unidad de disco. Se presiona el botón de búsqueda de archivos, saldrá la siguiente pantalla.

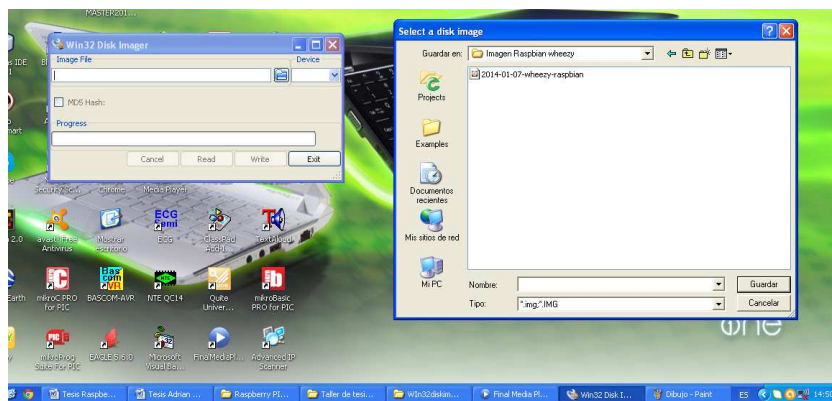


Figura 3.27 WIn32 Disk 2

Fuente: Investigador

Aquí se busca la imagen binaria del sistema operativo Raspbian Wheezy que se descargó del internet. Seleccionado el archivo se presiona el botón “write”. Una vez concluida la instalación se retira la memoria SD y se inserta en la ranura de la tarjeta Raspberry PI.

Una vez instalado todo el software, se conecta el Raspberry a un puerto del router de la red de área local, como un computador más. Se da inicio a la sesión de Raspberry dando doble click en el Icono previamente configurado con XLaunch, y se introduce la contraseña que por defecto viene cargada como “raspberrypi”.

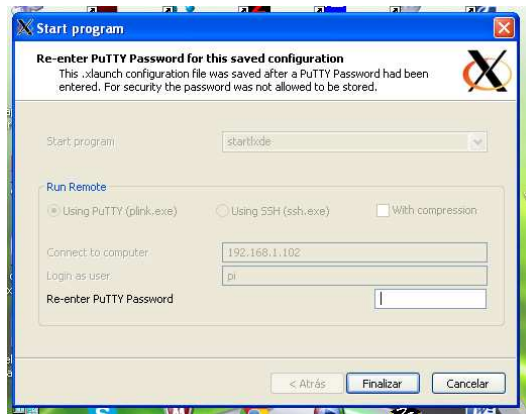


Figura 3.28 Iniciando sesión con XLaunch

Fuente: Investigador

Si la clave ingresada fue correcta arranca en la ventana principal o escritorio del sistema operativo cargado en la tarjeta SD de la placa Raspberry. Como se puede observar en la figura siguiente, es un ambiente amigable al usuario, se conserva el uso de íconos tal como trabajar en Windows o Linux.

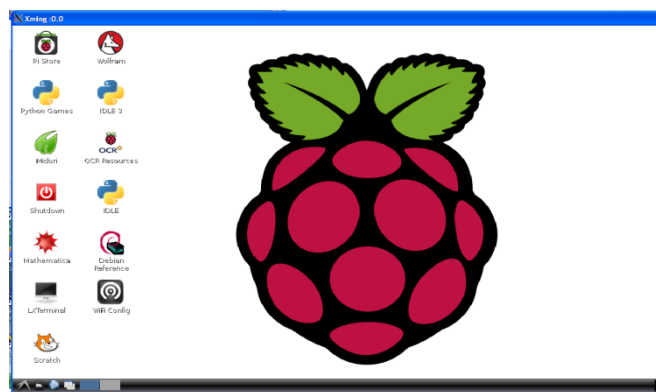


Figura 3.29 Pantalla del escritorio de Raspbian Wheezy

Fuente: Investigador

La placa Raspberry PI posee un icono de nombre LXTerminal, si se dá doble click se ingresa a la pantalla principal, ahí se teclea el comando: “sudo raspi-config” para ingresar a la configuración básica del sistema operativo, aquí se fijan parámetros como clave de ingreso, overclock, lenguaje entre otros.

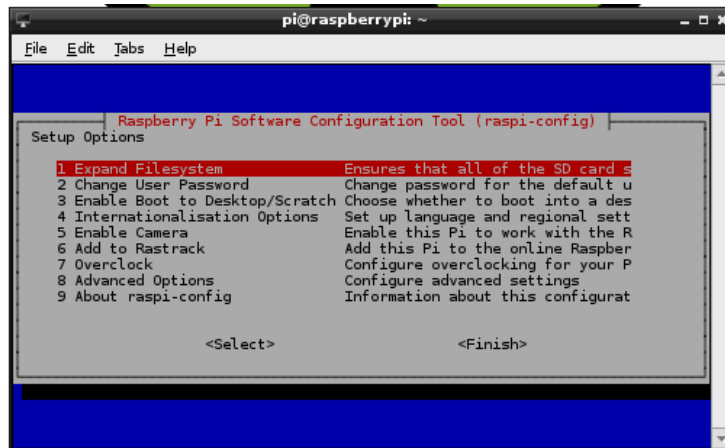


Figura 3.30 Pantalla configuración Raspberry Pi.

Fuente: Investigador

Existen comandos tanto para actualizar, descargar actualizaciones, importar librerías, llamar a programas ya creados, entre ellos se tiene:

- Sudo raspi-config
- Sudo apt-get update
- Sudo apt-get upgrade
- sudo apt-get install tightvncserver
- sudo vncserver :2 -geometry 1920x1080 -depth 24
- cat /proc/cpuinfo ----> para saber la versión y revisión de la placa.
- sudo apt-get update ----> Para actualizar los repositorios
- sudo apt-get upgrade ----> Actualiza paquetes del S.O
- sudo apt-get install tightvncserver ---> Instala VNC server en la Raspberry Pi
- sudo vncserver <parámetros> ---> configuración del servidor VNC
- sudo shutdown -h now ---> para apagar el sistema
- sudo reboot -f ---> para reiniciar el sistema
- sudo raspi-config ---> para acceder al menú de configuración en Raspbian
- ps -a ---> para saber todos los procesos que se están ejecutando.
- Kill -9 ID_proceso --> para matar un proceso determinado.
- passwd <Enter> --> para cambiar la contraseña
- df -f --> espacio libre en GB o MB de la tarjeta SD

- sudo -i --> para acceder como root
- ifconfig --> para saber la IP de la tarjeta
- exit --> para salir de la sesión

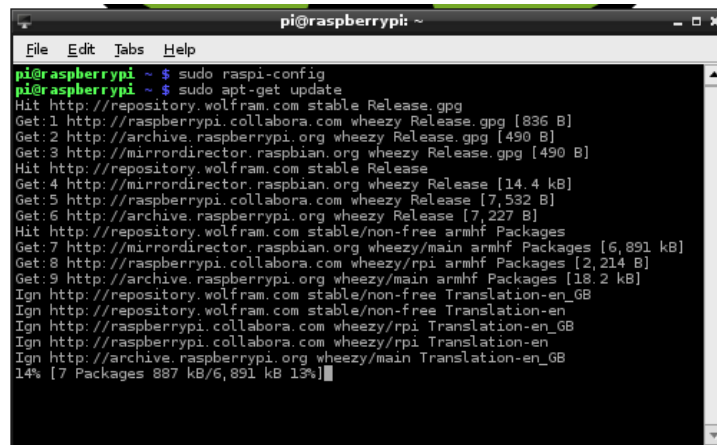


Figura 3.31 Pantalla de Actualización de Software “update”

Fuente: Investigador

3.4 Pruebas de funcionamiento

Los pines GPIO son los puertos que posee la placa Raspberry Pi para comunicarse con el exterior, al igual que los microcontroladores PIC o AVR se debe configurar los pines para que se comporten como entrada o como salida, este proceso se lo realiza mediante software. La figura siguiente muestra un diagrama del conector que se observa físicamente desde la parte superior de la placa Raspberry, tanto para las dos versiones que se consigue en el mercado.

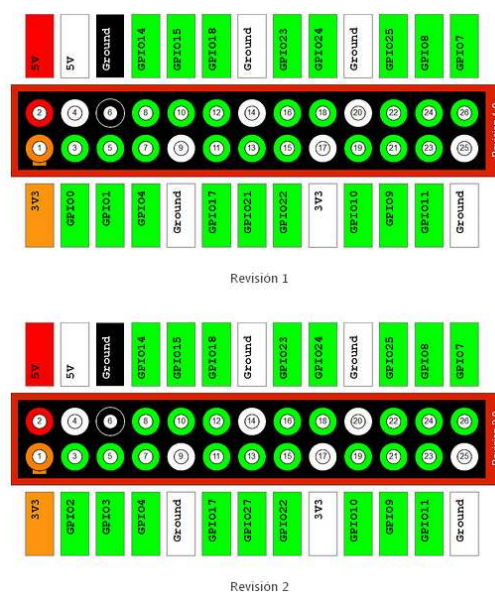


Figura 3.32 Pines GPIO para Raspberry Revisión 1 y 2

Fuente: (CyberHades, 2013)

3.4 .1 Software para probar las zonas del entrenador para Raspberry

El software se escribe en lenguaje de programación Python (Monk, 2013, pág. 204), una vez compilado se debe armar la parte del hardware para probar el correcto funcionamiento.

Los diferentes programas para probar son:

- Encendido intermitente
- Generación de PWM
- Suiche y led

3.4.1.1 Encendido intermitente

Un ejemplo práctico del uso del módulo Raspberry Pi es realizar un programa que mantenga una intermitencia permanente en un diodo led, para ello se realiza una conexión externa en un protoboard de un led con su resistencia limitadora de corriente hacia el pin GPIO.17 del módulo Raspberry.

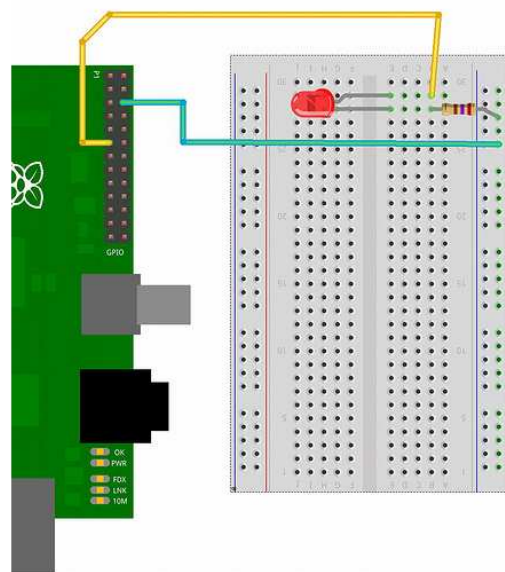


Figura 3.33 Pin GPIO.17 configurado como salida

Fuente: (CyberHades, 2013)

Las zonas de salidas se puede probar con un mismo programa, se puede realizar un programa configurando como salida un pin GPIO que se comporte de manera intermitente. Este pin puede ser conectado a cualquier zona para que funcione como periférico de salida, se puede usar la zona led, zona relé, zona transistor entre otras, el resultado es ver el efecto de intermitencia. (Monk, 2013, pág. 204)

En el anexo 2 encontrará el programa demostrativo.

La tabla 3.3 muestra los resultados cambiando el tiempo de encendido / apagado y verificando el resultado obtenido en la intermitencia del led.

Prueba realizada	Periodo de encendido (ms)	Periodo de apagado (ms)	Resultado obtenido
1	100	100	Correcto
2	200	200	Correcto
3	500	500	Correcto
4	1000	1000	Correcto
5	1500	1500	Correcto
6	50	50	Correcto
7	20	20	Correcto

Tabla 3.3 Resultado de pruebas en zona de leds

Fuente: Investigador

3.4.1.2 Generación de PWM

Se realizó pruebas de funcionamiento generando una señal PWM para controlar la intensidad de un diodo led, esto es aplicable también para controlar la velocidad de un motor DC. (Monk, 2013, pág. 351)

En el anexo 3 encontrará el programa demostrativo.

En la tabla 3.4 se muestra los resultados obtenidos en esta segunda prueba.

Prueba realizada	PWM ingresado	Resultado obtenido
1	100	Correcto
2	70	Correcto
3	50	Correcto
4	34	Correcto
5	20	Correcto
6	10	Correcto
7	0	Correcto

Tabla 3.4 Resultado de pruebas generación PWM

Fuente: Investigador

3.4.1.3 Suiche y led

Es un programa que sirve para configurar los pines del Raspberry como entrada y como salida. (Monk, 2013, pág. 229)

En el anexo 4 se encuentra el programa demostrativo.

En la tabla 3.5 se muestra los resultados obtenidos en esta tercera prueba, se puede ver como se ha configurado por software diferentes pines GPIO para probar que cualquier pin elegido puede servir como entrada o salida.

Prueba realizada	GPIO como Salida	GPIO como entrada	Resultado obtenido
1	GPIO 2	GPIO 10	Correcto
2	GPIO 3	GPIO 22	Correcto
3	GPIO 4	GPIO 27	Correcto
4	GPIO 17	GPIO 4	Correcto
5	GPIO 27	GPIO 17	Correcto
6	GPIO 22	GPIO 3	Correcto
7	GPIO 10	GPIO 2	Correcto

Tabla 3.5 Resultado de pruebas suiche y led

Fuente: Investigador

3.5 Análisis de resultados

3.5.1 Análisis de resultados de compilación / carga vs tiempo

En la tabla 3.6 se ve que la compilación y la carga del software sobre la tarjeta Raspberry Pi es más rápida debido a que no se debe retirar el microcontrolador (Pic/Avr) para pasarlo al programador cada vez que se tenga que programar o reprogramar el dispositivo, incluso comparándolo con la línea Arduino es sumamente más rápida.

Otro resultado obtenido es que la estática presente en el cuerpo al momento de extraer un microcontrolador (Pic/Avr) para reprogramarlo es perjudicial y al momento de volver a poner el microcontrolador ya programado puede existir error en la ubicación dando como resultado que se averíe y no pueda ser utilizado.

Tecnología Utilizada	Tiempo de compilación (seg.)	Tiempo de carga de software (seg.)
PIC	5	8
AVR	6	10
Arduino Uno	8	4
Raspberry Pi	2	2

Tabla 3.6 Resultados de compilación/carga vs tiempo

Fuente: Investigador

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- Se estudió el manejo de la placa Raspberry Pi, tanto para cargar el sistema operativo en la memoria SD, y se determinó que el lenguaje Python es el mejor recurso como plataforma de programación.
- Si hay la presencia de algún mal funcionamiento de un circuito implementado en el entrenador para Raspberry revisar primero la parte de conexiones electrónicas con la fuente de alimentación apagada, y luego revisar el software.
- Se comprobó con la implementación del entrenador que al realizar diseños electrónicos, se redujo el tiempo de ensamble, debido a que las conexiones de polarización las trae consigo el entrenador y se tendrá sólo que conectar las entradas o salidas de las respectivas zonas al Raspberry.
- Se diseñó una placa de circuito impreso con el software Proteus que genera los archivos Gerber que sirven para las máquinas CNC que fabrican circuitos impresos al por mayor, esto es útil ya que la idea de la empresa es fabricar placas del entrenador al por mayor.
- El departamento de investigación de la empresa Laservisión Cía. Ltda. validó el correcto funcionamiento del entrenador mediante la prueba de cada una de las zonas y determinó que es el equipo adecuado para el desarrollo de proyectos con Raspberry Pi.

Recomendaciones

- Para la fabricación de la placa de circuito impreso se sugiere el uso de fibra de vidrio, debido a la resistencia a altas temperaturas al momento de realizar la transferencia térmica.
- Usar cable utp para realizar las conexiones entre las diferentes zonas y el módulo Raspberry.
- Si no va a utilizar un pc para trabajar con Raspberry y desea trabajar con un televisor, utilice una con conexión HDMI debido a la calidad superior de la salida de video compuesto.

Bibliografía

- Admin. (2014). *50 ebooks gratis en Python*. Retrieved from <http://www.raspberrypi.com/news/50-ebooks-gratis-en-python-para-utilizar-con-la-raspberry-pi/>
- Admin. (2014). *Adafruit actualiza sus librerías en Python para la Raspberry Pi*. Retrieved from <http://www.raspberrypi.com/news/adafruit-actualiza-sus-librerias-en-python-para-la-raspberry-pi/>
- Alciro. (2014). *Correspondencia en las patillas del conector HDMI*. Retrieved from http://www.alciro.org/alciro/conectores_26/patillas-cable-HDMI-a-DVI-D_274.htm
- Arduino. (2014). *Arduino*. Retrieved MAYO 2014, from <http://www.arduino.cc/>
- Arduino. (2014). *Arduino*. Retrieved Junio 3, 2014, from <http://arduino.cc/es/Reference/serial>
- Arduino. (2014). *Arduino*. Retrieved Mayo 30, 2014, from <http://arduino.cc/en/Tutorial/LiquidCrystal>
- Arduino. (2014). *Arduino Playground*. Retrieved from <http://playground.arduino.cc/Main/SimilarBoards#goShie>
- Arduino, T. (2013, Diciembre 28). *Taller Arduino*. Retrieved 2014, from <http://tallerarduino.com/tag/teclado/>
- Banzi, M. (2011). *Getting Started with Arduino*. In M. Banzi, *Getting Started with Arduino*. USA.
- Beauford Court. (2014). *Learn to program your Raspberry*. Retrieved from <http://www.techradar.com/news/software/learn-to-program-your-raspberry-pi-1148194/2>
- Canonical LTDA. (2014). *Imagen Writer*. Retrieved from <https://launchpad.net/win32-image-writer/+download>
- Corrales, S. (2006). *Electrónica Práctica con microcontroladores PIC*. In S. Corrales, *Electrónica Práctica con microcontroladores PIC*. Quito.
- CyberHades. (2013). Retrieved from Raspberry Pi, interactuando con GPIO desde la línea de comandos: <http://www.cyberhades.com/2013/01/18/raspberry-pi-interactuando-con-gpio-desde-la-linea-de-comandos/>
- Dice. (2014). *Xming X server for Windows*. Retrieved from <http://sourceforge.net/projects/xming/files/test/download>
- Famatech. (2014). *Advanced IP Scanner*. Retrieved from <http://www.advanced-ip-scanner.com/es/>
- Garagem, L. d. (2012, marzo 29). *Laboratório de Garagem*. Retrieved mayo 22, 2014, from <http://labdegaragem.com/profiles/blogs/tutorial-ultra-rapido-de-arduino>

- Google Maps. (2014). *Google Maps*. Retrieved from <https://www.google.com.ec/maps/@-0.1808383,-78.4801591,794m/data=!3m1!1e3>
- Honeywell / Scanning & Mobility. (2014). *Honeywell / Scanning & Mobility*. Retrieved mayo 20, 2012, from <https://www.honeywellaidc.com/en-LA/Pages/Product.aspx?category=hand-held-barcode-scanner&cat=HSM&pid=9520/40>
- Informatica Moderna. (2014). Retrieved from http://www.informaticamoderna.com/EI_puerto_USB.htm
- Ladelec. (2014). *Pic 18F4550 con tarjeta SD*. Retrieved from <http://www.ladelec.com/practicas/circuitos-por-arossini/754-pic-18f4550-con-tarjeta-sd-y-archivos-de-texto>
- Margolis, M. (2011). *Arduino Cookbook*. United States of America: O'Reilly Media, Inc.,.
- Monk, S. (2013). *Raspberry Pi Cookbook*. O'Reilly Media.
- Penagos, R. (2010). Cómo programar en lenguaje C los microcontroladores. In R. Penagos, *Cómo programar en lenguaje C los microcontroladores*. Quito.
- Putty.ORG. (2014). *Download Putty*. Retrieved from <http://www.putty.org/>
- Raspberry Pi Foundation. (2014). *Downloads Raspberry Pi*. Retrieved from <http://www.raspberrypi.org/downloads/>
- Raspberry, P. (2104). Retrieved from <http://www.raspberrypi.org/>
- Reyes, C. (2008). Microcontroladores Pic Programación en Basic. In C. Reyes, *Microcontroladores Pic Programación en Basic*. Quito.
- Sparkfun. (2014). *Sparkfun*. Retrieved Marzo 22, 2014, from <https://www.sparkfun.com/products/10438>
- Unicrom. (2012). *Unicrom Electronica*. Retrieved from http://www.unicrom.com/Tut_transistor_bipolar.asp
- Unicrom. (2012). *Unicrom Electrónica*. Retrieved from http://www.unicrom.com/Tut_triac.asp
- Valencia, R. (2010). Aplicaciones electrónicas con microcontroladores AVR. In R. Valencia, *Aplicaciones electrónicas con microcontroladores AVR*. Ibarra.
- Weblogs S.L. (2014). *Conoce a la placa que quiere revolucionar tu mundo digital*. Retrieved from <http://www.xataka.com/componentes-de-pc/conoce-a-la-placa-que-quiere-revolucionar-tu-mundo-digital-raspberry-pi-a-fondo>

ANEXOS

ANEXO 1
ENTREVISTAS REALIZADAS

ANEXO 2
PROGRAMAS PYTHON
ENCENDIDO INTERMITENTE


```
def blink():
print "Ejecucion iniciada..."
    iteracion = 0
    while iteracion < 30: ## Segundos que durara la funcion
GPIO.output(17, True) ## Enciendo el 17
GPIO.output(27, False) ## Apago el 27
time.sleep(1) ## Esperamos 1 segundo
GPIO.output(17, False) ## Apago el 17
GPIO.output(27, True) ## Enciendo el 27
time.sleep(1) ## Esperamos 1 segundo
    iteracion = iteracion + 2 ## Sumo 2 porque he hecho dos parpadeos
print "Ejecucion finalizada"
GPIO.cleanup() ## Hago una limpieza de los GPIO
```

ANEXO 3
PROGRAMAS PYTHON
GENERACIÓN PWM

Software para Raspberry

```
board = pyfirmata.Arduino('/dev/ttyACM0')
led_pin = board.get_pin('d:10:p')
while True:
    duty_s = raw_input("Enter Brightness (0 to 100):")
    duty = int(duty_s)
    led_pin.write(duty / 100.0)
```

ANEXO 4
PROGRAMAS PYTHON
SUICHE Y LED

Suiche y led:

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BOARD)
GPIO.setup(11, GPIO.OUT)
GPIO.setup(12, GPIO.IN)
state = False

def wait():
    time.sleep(0.1)
while True:
    wait() #don't burn the CPU
    if not GPIO.input(12):
        print state
        state = not state
        GPIO.output(11, state)
        while not GPIO.input(12):
            wait()
        continue
```

ANEXO 5
COSTOS DE IMPLEMENTACIÓN

Elementos	Costo Unitario	Costo Total
Resistencias	0.03	1.20
Transistores	0.09	0.45
Modulos RF tx/rx	8.87	8.87
Pantalla 2x16	5.47	5.47
Displays	0.46	1.84
Raspberry	80.00	80.00
Arduino Uno	30.00	30.00
Teclado	7.5	7.50
Extras	40.00	40.00
Zumbador	1.18	1.18
Caja	40.00	40.00
SUBTOTAL		216.26 + IVA

Tabla: Costos del Equipo

Fuente: Investigador