



# **ESPE**

**UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA**

**UNIDAD DE GESTIÓN DE  TECNOLOGÍAS**

**DEPARTAMENTO DE ELECTRÓNICA Y  
COMPUTACIÓN**

**CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN  
Y AVIÓNICA**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE TECNÓLOGO EN ELECTRÓNICA MENCIÓN  
INSTRUMENTACIÓN Y AVIÓNICA**

**TEMA: IMPLEMENTACIÓN DE UN CONTROL DE  
TEMPERATURA MEDIANTE EL MICROCOMPUTADOR  
RASPBERRY PI PARA PRÁCTICAS DE  
MICROCONTROLADORES EN LA UNIDAD DE GESTIÓN DE  
TECNOLOGÍAS.**

**AUTOR: CALAPAQUI GONZÁLEZ MARÍA JOSÉ**

**DIRECTORA: ING. EMMA CAMPAÑA RIOFRIO MSC**

**LATACUNGA**

**2017**



**DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN  
CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN &  
AVIÓNICA**

**CERTIFICACIÓN**

Certifico que el trabajo de titulación, **“IMPLEMENTACIÓN DE UN CONTROL DE TEMPERATURA MEDIANTE EL MICROCOMPUTADOR RASPBERRY PI PARA PRÁCTICAS DE MICROCONTROLADORES EN LA UNIDAD DE GESTIÓN DE TECNOLOGÍAS.”** ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar al señorita **CALAPAQUI GONZÁLEZ MARÍA JOSÉ** para que lo sustente públicamente.

**Latacunga, 31Julio del 2017**

.....  
**ING. EMMA CAMPAÑA RIOFRIO MSC**  
**DIRECTOR**



**DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN  
CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y  
AVIÓNICA**

**AUTORÍA DE RESPONSABILIDAD**

Yo, **CALAPAQUI GONZÁLEZ MARÍA JOSÉ**, con cédula de identidad N° **1724746563** declaro que este trabajo de titulación “**IMPLEMENTACIÓN DE UN CONTROL DE TEMPERATURA MEDIANTE EL MICROCOMPUTADOR RASPBERRY PI PARA PRÁCTICAS DE MICROCONTROLADORES EN LA UNIDAD DE GESTIÓN DE TECNOLOGÍAS**”, ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

**Latacunga, 31 Julio del 2017**

.....  
CALAPAQUI GONZÁLEZ MARÍA JOSÉ

C.C.1724746563



**DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN  
CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y  
AVIÓNICA**

**AUTORIZACIÓN**

Yo, **CALAPAQUI GONZÁLEZ MARÍA JOSÉ**, autorizo a la Universidad de las Fuerzas Armadas publicar en la biblioteca Virtual de la institución la presente trabajo de titulación “**IMPLEMENTACIÓN DE UN CONTROL DE TEMPERATURA MEDIANTE EL MICROCOMPUTADOR RASPBERRY PI PARA PRÁCTICAS DE MICROCONTROLADORES EN LA UNIDAD DE GESTIÓN DE TECNOLOGÍAS**”, cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

**Latacunga, 31 Julio 2017**

---

Calapaqui González María José

C.C: 1724746563

## DEDICATORIA

Dedico esta tesis a mi Dios por guiarme cada día en este camino de estudio, dándome fortaleza, perseverancia, y fuerzas para seguir adelante para poder cumplir este objetivo de mi vida.

A mi madre por el apoyo incondicional de cada día y por la enseñanza de no rendirme a medio camino de terminar lo que se empieza, por los buenos consejos que me enseñaron para mi bienestar de esta etapa de estudios y en mi vida personal. Y agradezco por estar presente en la realización de esta meta que es tan importante para mí. Por todo su apoyo, ayuda, sus palabras que me motivaron a seguir adelante, sus consejos y su dedicación hacia mí

Mis hermanos, Diego, Johanna y Miguel, por apoyarme a la distancia pero estuvieron siempre presente cuando más los necesitaba, los quiero mucho.

CALAPAQUI MARÍA JOSÉ

## **AGRADECIMIENTO**

A mi Directora del Trabajo Técnico, Ing. Emma Campaña y Director de Carrera, Ing. Pablo Pilatasig, quienes tuvieron paciencia en enseñarme sus conocimientos y guiarme para poder llegar a esta meta y formarme como una buena persona y profesional un infinito agradecimiento.

A mis compañeros y amigos, quienes vivimos día tras día aprendiendo cosas nuevas junto a ellos, por apoyarme siempre que necesitaba y por convivir todo estos tres años conmigo de experiencias, alegrías, tristezas y muchas cosas más . Mil gracias por aguantar todo lo malo o bueno de mí y aun seguir a mi lado le agradezco mucho.

A la institución por haberme permitido estudiar y así formándome primero como persona y una buena profesional para poder desempeñarme en el ámbito profesional. Y agradecer infinitamente a todos los ingenieros de la institución por sus maravillosos conocimientos.

CALAPAQUI MARIA JOSE

## ÍNDICE DE GENERAL

<b>CONTENIDO</b>	<b>Pág.</b>
Portada .....	i
CERTIFICACIÓN.....	ii
AUTORÍA DE RESPONSABILIDAD.....	iii
AUTORIZACIÓN .....	iv
DEDICATORIA.....	v
AGRADECIMIENTO.....	vi
ÍNDICE DE GENERAL .....	vii
ÍNDICE DE TABLAS .....	x
ÍNDICE DE FIGURAS.....	xi
RESUMEN.....	xiii
ABSTRACT .....	xiv
<b>CAPÍTULO I.....</b>	<b>1</b>
<b>PLANTEAMIENTO DEL PROBLEMA .....</b>	<b>1</b>
1.1    Antecedentes.....	1
1.2    Planteamiento del problema .....	2
1.3    Justificación .....	2
1.4    Objetivos.....	3
1.4.1    Objetivo General .....	3
1.4.2    Objetivos específicos .....	3
1.5    Alcance .....	3

<b>CAPÍTULO II</b> .....	4
<b>MARCO TEÓRICO</b> .....	4
2.1  Minicomputadora raspberry pi 3 modelo b.....	4
2.1.1  Características .....	4
2.2  Puerto GPIO .....	5
2.3  Sistemas operativos soportados.....	7
2.3.1  Raspbian .....	7
2.4  Real VNC.....	8
2.4.1  Características de real VNC:.....	9
2.4.2  Interfaz VNC.....	9
2.4.3  Usos del VNC.....	10
2.5  Lenguajes de programación .....	10
2.5.1  Python .....	11
2.5.2  De dónde viene python .....	11
2.5.3  Características del lenguaje python .....	12
2.5.4  Programar con python en raspberry pi.....	13
2.5.5  Programar python con IDLE .....	14
2.6  PWM para raspberry pi.....	14
2.7  Bus i2c de raspberry pi .....	16
2.7.1  ¿Dónde se encuentra el bus I2C en el raspberry?.....	17
2.7.2  Funciones SMBus (bus de administración del sistema) .....	18
2.8  Sensor PCF8591 .....	18
2.8.1  Características y beneficios. ....	18
2.8.2  Esquema de hardware .....	19
2.8.3  Descripción básica del circuito .....	20

<b>CAPÍTULO III.....</b>	<b>21</b>
<b>DESARROLLO DEL PROYECTO.....</b>	<b>21</b>
3.1 Preliminares.....	21
3.2 Herramientas y equipos utilizados.....	21
3.2.1 Instalación de sistema operativo al raspberry pi3 modelo b.....	21
3.2.2 Instalación y configuración del VNC.....	25
3.2.3 Configuración del sensor PCF8591.....	30
3.2.4 Configuración y programación del LCD.....	34
3.2.5 Programación del control de temperatura.....	36
3.3 Prácticas para microcontroladores usando el raspberry pi3.....	39
3.3.1 Lectura de temperatura de un sensor DS18B20.....	39
• Materiales.....	39
<b>CAPÍTULO IV.....</b>	<b>42</b>
<b>CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>42</b>
4.1 Conclusiones.....	42
4.2 Recomendaciones.....	43
GLOSARIO DE TÉRMINOS.....	44
REFERENCIAS BIBLIOGRÁFICAS.....	46
ANEXOS.....	48

## ÍNDICE DE TABLAS

Tabla 1 Especificaciones de los pines dePCF8591 .....	34
Tabla 2 Conexión del LCD con raspberry pi .....	35

## ÍNDICE DE FIGURAS

<b>Figura 1</b> Raspberry Pi .....	5
<b>Figura 2</b> Puertos GPIO del Raspberry Pi .....	7
<b>Figura 3</b> Sistema Operativo Raspbian .....	8
<b>Figura 4</b> Lenguaje de Programación Python.....	11
<b>Figura 5</b> Python IDLE.....	14
<b>Figura 6</b> Python IDLE.....	15
<b>Figura 7</b> Esquema de Bus.....	16
<b>Figura 8</b> Pines de PWM .....	17
<b>Figura 9</b> Sensor PFC8591.....	19
<b>Figura 10</b> Descripción del PFC9591 .....	20
<b>Figura 11</b> Ventana de elegir la dirección.....	22
<b>Figura 12</b> Ubicación del software .....	22
<b>Figura 13</b> Selección de Raspbian .....	22
<b>Figura 14</b> Confirmación.....	23
<b>Figura 15</b> Cargando la imagen.....	23
<b>Figura 16</b> Finalización de la imagen.....	23
<b>Figura 17</b> Insertar la tarjeta SD al Raspberry Pi.....	24
<b>Figura 18</b> Instalación de Debian.....	24
<b>Figura 19</b> Configuración del raspberry pi. ....	25
<b>Figura 20</b> Configuración de localización.....	25
<b>Figura 21</b> Ventana del Terminal. ....	26
<b>Figura 22</b> Dirección de wlan0.....	26
<b>Figura 23</b> Ingreso de código para interfaz.....	27
<b>Figura 24</b> Ingreso de dirección IP. ....	27
<b>Figura 25</b> Software VNCViewer. ....	28
<b>Figura 26</b> Ingreso de dirección IP del raspberry pi.....	28
<b>Figura 27</b> Ventana de comprobación. ....	29
<b>Figura 28</b> Ingreso de usuario y contraseña.....	29
<b>Figura 29</b> Comunicación de raspberry pi y PC.....	30
<b>Figura 30</b> Ingreso de código en el terminal. ....	30

<b>Figura 31</b> Comentar líneas de código. ....	31
<b>Figura 32</b> Comentar líneas de código. ....	31
<b>Figura 33</b> Activación de interfaces. ....	32
<b>Figura 34</b> Código de programación del PCF8591. ....	33
<b>Figura 35</b> Circuito del PFC8591. ....	33
<b>Figura 36</b> Código de programación del LCD. ....	35
<b>Figura 37</b> Configuración en el terminal para LCD. ....	36
<b>Figura 38</b> Programación para el control de módulo de temperatura. ....	37
<b>Figura 39</b> Comprobación del sensor con el modulo. ....	38
<b>Figura 40</b> Demostración de temperatura y set point en LCD. ....	38
<b>Figura 41</b> Resultados en Python. ....	39
<b>Figura 42</b> Esquema de ds18b20 con raspberry. ....	40
<b>Figura 43</b> Programación en Python. ....	40
<b>Figura 44</b> Resultados del sensor DS18B20. ....	41

## RESUMEN

El presente trabajo de graduación se realizó una implementación de un control de temperatura mediante el microcomputador raspberry pi. Primero se instaló el sistema operativo Debian Reference que es una versión de Linux, para tener un interfaz raspberry pi con un PC, se instaló el software VNC para facilitar el manejo. Para la programación se ocupó el software Python que viene en la versión de Linux, el cual se debe seguir la estructura de código de programación para omitir errores. Se conectó el LCD a 5V, GND y los pines GPIO correspondientes al raspberry. Se ocupó un PCF8591 este tiene 16 pines, lo cual AIN0 que es pin 1 del circuito va conectado al VIC del módulo de temperatura, AIN1 es pin 2 va dirigido hacia el centro del potenciómetro de 10K, lo que es la referencia del Set Point. SDA es pin 9 de datos va conectada al GPIO2 (pin3) y SCL pin10 de clock hacia GPIO3 (pin5), estos dos pines realizan la función de comunicación I2C maestro-esclavo. Los demás pines del PFC8591 van conectados al GND y 5V guiándose del diagrama, excepto OSC y AOUT son pines que no se utilizan y no van conectados a ningún lugar. En LCD se observa el Valor de Proceso y Set Point, lo cual SP va a estar en 70 y se enciende el módulo de temperatura va subir el nivel de temperatura hasta llegar al límite se va apagar mientras este menor se va a mantener encendido.

### **PALABRAS CLAVES:**

- **FRAMBUESA PI**
- **PCF8591**
- **LCD**
- **PYTHON**
- **GPIO**

## ABSTRACT

At the present graduation work an implementation of temperature control by the microcomputer raspberry pi was done. First the Debian Reference operating system was installed, which is a version of Linux, to have a raspberry pi interface with a PC, VNC software was installed to facilitate the handling. For programming the Python software was used, which came in the version of Linux, that should follow the code structure of programming to omit errors. The LCD was connected to 5V, GND and the GPIO pins corresponding to raspberry. A PCF8591 was occupied which has 16 pins, which AIN0 which is pin 1 of the circuit is connected to the VIC of the temperature module, AIN1 is pin 2 is directed towards the center of the 10k potentiometer which gives reference to the Set Point. SDA is data pin 9 connected to the GPIO2 (pin3) and SCL clock pin10 to GPIO3 (pin5), these two pins perform the I2C master-slave communication function. The other pins of the PFC8591 are connected to the GND and 5V guided from the diagram, except OSC and AOUT are pins that are not used and are not connected anywhere. In LCD shows the Process Value and Set Point Which SP will be at 70 and the temperature module will turn up the temperature level until it reaches the limit it will turn off while this minor is going to keep on.

### KEYWORDS:

- RASPBERRY PI
- PCF8591
- LCD
- PYTHON
- GPIO

LIC.PABLO S. CEVALLOS MSC.

**DOCENTE UGT-ESPE**

# CAPÍTULO I

## PLANTEAMIENTO DEL PROBLEMA

IMPLEMENTACIÓN DE UN CONTROL DE TEMPERATURA MEDIANTE EL MICROCOMPUTADOR RASPBERRY PI PARA PRÁCTICAS DE MICROCONTROLADORES EN LA UNIDAD DE GESTIÓN DE TECNOLOGÍAS.

### 1.1 Antecedentes

En la provincia de Cotopaxi en la ciudad de Latacunga se encuentra ubicada la Unidad de Gestión de Tecnologías de la Universidad de las Fuerzas Armadas - ESPE, una institución que brinda un alto rendimiento académico contando con talleres y laboratorios para que los estudiantes tengan un mejor conocimiento en el aspecto práctico.

En la Carrera de Electrónica Mención Instrumentación y Aviónica el desarrollo de prácticas se realiza en los respectivos laboratorios, sin embargo, el avance tecnológico se va dando día a día a nivel mundial, es por ello que se debe ir adquiriendo dispositivos tecnológicos de acuerdo al desarrollo de la industria. Se realizó una investigación en el laboratorio de Instrumentación Virtual que cuenta con módulos de temperatura y nivel determinando que carece de equipos tecnológicos para su control.

Es importante adquirir el Raspberry Pi, que es un ordenador de tamaño de tarjeta que soporta varios componentes y es capaz de realizar varias funciones que una computadora de escritorio. Tomando en cuenta el proyecto de grado del señor Díaz Córdova Pedro Gabriel con el tema: "IMPLEMENTACIÓN DE UN ROBOT EXPLORADOR TIPO VEHICULO CONTROLADO POR MEDIO DEL INTERNETE COMO MATERIAL DIDACTICO PARA LA UNIDAD DE GESTION DE TECNOLOGIAS", cuyo tema ha sido de gran utilidad para los estudiantes.

## **1.2 Planteamiento del problema**

En el Laboratorio de Instrumentación Virtual de la Unidad de Gestión de Tecnologías de la Universidad de Fuerzas Armadas-ESPE, los estudiantes realizan prácticas con módulos de temperatura y nivel para simulaciones sobre procesos industriales en lo que el laboratorio no cuenta con equipos tecnológicos necesarios para permitir el manejo y control de las mismas.

Hoy en día la tecnología va avanzando cada vez más con respecto a los microcontroladores en lo que se está actualizando a nivel mundial dejando atrás instrumentos electrónicos. El manejo y utilización de estos dispositivos que actualmente posee el laboratorio, no permitirá a los estudiantes incrementar sus conocimientos en el ámbito industrial esto puede ser de gran desventaja en el ámbito laboral ante otros profesionales.

## **1.3 Justificación**

La implementación de un control de temperatura mediante el Raspberry Pi para el laboratorio de Instrumentación Virtual ayuda a la enseñanza y aprendizaje de los estudiantes de la Carrera de Electrónica Mención Instrumentación y Aviónica de la Unidad de Gestión de Tecnologías, permitiendo obtener mayor conocimiento y destrezas necesarias para un mejor rendimiento y desempeño en el ámbito profesional.

El Raspberry Pi brinda facilidad de control y manejo de equipos, dispositivos electrónicos mediante la manipulación de sus pines, dependiendo las necesidades de los estudiantes lo cual podrán adaptarse a las nuevas tecnologías fácilmente a nivel industrial. Este proyecto es factible porque se puede encontrar en el mercado tecnológico y disfrutar de las posibilidades que ofrece, pues es un dispositivo económico y actualizado con una manipulación confiable al ser utilizado.

## **1.4 Objetivos**

### **1.4.1 Objetivo General**

Implementar un control de temperatura mediante el microcomputador Raspberry Pi para prácticas de microcontroladores en la Unidad de Gestión de Tecnologías.

### **1.4.2 Objetivos específicos**

- Indagar información teórica referente a los dispositivos que se va utilizar en el control del módulo mediante medios bibliográficos para tener conocimiento de los equipos.
- Instalar el sistema operativo mediante los requisitos técnicos para la manipulación del microcomputador Raspberry Pi.
- Realizar prueba de funcionamiento del proyecto mediante los procedimientos indicados para el control del módulo de temperatura.

## **1.5 Alcance**

Este proyecto en la Carrera de Electrónica Mención Instrumentación y Aviónica de la Unidad de Gestión de Tecnologías, brinda a los estudiantes y docentes renovar el medio de aprendizaje por un dispositivo actualizado de última tecnología donde podrán desarrollar prácticas para aplicar los conocimientos adquiridos en el aula, lo que permitirá al estudiante obtener mayor experiencia en el campo práctico para desempeñar en el ámbito profesional.

## CAPÍTULO II

### MARCO TEÓRICO

#### 2.1 Minicomputadora raspberry pi 3 modelo b

Según (Contreras, 2013) raspberry pi es:

Raspberry PI es una placa computadora de bajo coste, se podría decir que es un ordenador de tamaño reducido, del orden de una tarjeta de crédito, desarrollado en el Reino Unido por la Fundación Raspberry PI (Universidad de Cambridge) en 2011, con el objetivo de estimular la enseñanza de la informática en las escuelas, aunque no empezó su comercialización hasta el año 2012. A la Raspberry Pi la han definido como una maravilla en miniatura, que guarda en su interior un importante poder de cómputo en un tamaño muy reducido. Es capaz de realizar cosas extraordinarias. (Contreras, 2013).

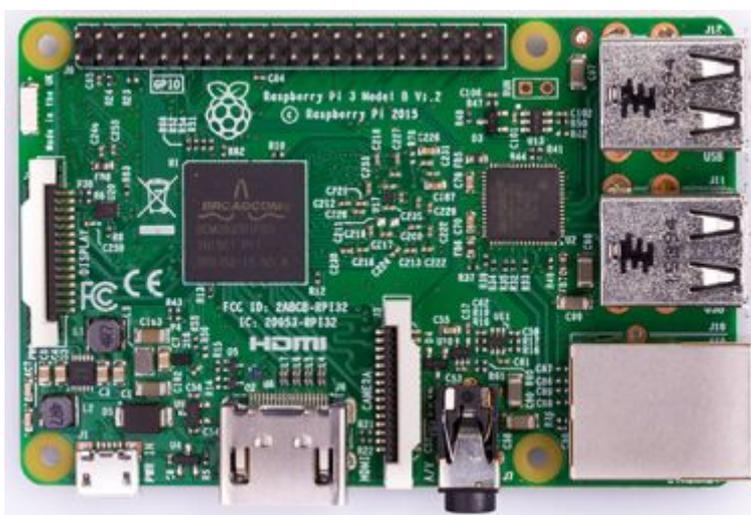
##### 2.1.1 Características

La frambuesa Pi 3 es la tercera generación de frambuesa Pi. Reemplazó a versiones modelo B en febrero de 2016. La frambuesa Pi 2 tiene:

- Una CPU ARMv8 quad-core de 64 bits de 64 bits y 1.2 GHz
- LAN inalámbrica 802.11n
- Bluetooth 4.1
- Bluetooth baja energía (BLE)
- El Pi 3, también tiene:
- 1 GB de RAM
- 4 puertos USB
- 40 clavijas GPIO
- Puerto HDMI completo
- Puerto Ethernet

- Conector de audio combinado de 3,5 mm y vídeo compuesto
- Interfaz de la cámara (CSI).
- Interfaz de pantalla (DSI)
- Ranura para tarjeta Micro SD (ahora push-pull en lugar de push-push)
- VideoCore IV núcleo de gráficos 3D (RASPBERRY PI 3 MPODELO B, 2016)

El Pi 3 de frambuesa tiene un factor de forma idéntico al anterior Pi 2 (y Pi 1 modelo B +) y tiene compatibilidad completa con Raspberry Pi 1 y 2. (RASPBERRY PI 3 MPODELO B, 2016)



**Figura 1** Raspberry Pi

**Fuente:** (RASPBERRY PI 3 MPODELO B, 2016)

## 2.2 Puerto GPIO

El estudio de (PE, 2015) encontró lo siguiente:

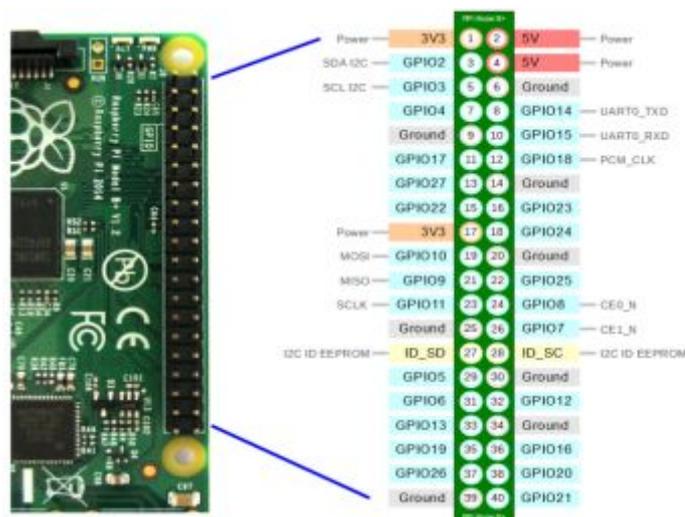
GPIO (General Purpose Input/Output) es, como su propio nombre indica, un sistema de E/S (Entrada/Salida) de propósito general, es decir, una serie de conexiones que se pueden usar como entradas o salidas para usos múltiples. Estos pines están incluidos en todos los modelos de Raspberry Pi, para que se pueda realizar proyectos interesantes como se haría con Arduino.

Los GPIO representan la interfaz entre la Raspberry Pi y el mundo exterior. Y con ellos se podrá realizar multitud de proyectos, desde hacer titilar un LED hasta otros mucho más sofisticados. Pero para eso se debe saber sus características y como se programan. Lo primero variará en función de la revisión de placa que se tenga o del modelo. (PE, 2015)

El estudio (PE, 2015) encontró lo siguiente:

Todos los pines son de tipo “unbuffered”, es decir, no disponen de buffers de protección, así que se deberá tener cuidado con las magnitudes (voltajes, intensidad) cuando se conecte componentes a ellos para no dañar la placa. También se debe tener muy en cuenta la utilización de pines, no todos los pines tienen la misma función:

- **Pines de alimentación:** se va a apreciar pines de 5V, 3V3 (limitados a 50mA) y tierra (GND o Ground), que aportan alimentación a estos voltajes para los circuitos. Pueden servir como una fuente de alimentación, aunque también se puede utilizar otras fuentes (pilas, fuentes de alimentación externas, etc). Recuerde que son unbuffered y se debe tener cuidado para no dañar la placa.
- **DNC (Do Not Connect):** son pines que por el momento no tienen función, pero en futuras implementaciones son utilizados para otros fines. Por eso solo se va a encontrar en modelos más primitivos de la Raspberry Pi. En las actuales placas han sido marcados como GND.
- **GPIO normales:** son conexiones configurables que puedes programar para tus proyectos
- **GPIO especiales:** dentro de éstos se encuentran algunos pines destinados a una interfaz UART, con conexiones TXD y RXD que sirven para comunicaciones en serie, como por ejemplo, conectar con una placa Arduino. También podemos ver otros como SDA, SCL, MOSI, MISO, SCLK, CE0, CE1, etc. (PE, 2015).



**Figura 2** Puertos GPIO del Raspberry Pi

**Fuente:** (PrototipoPi, 2016)

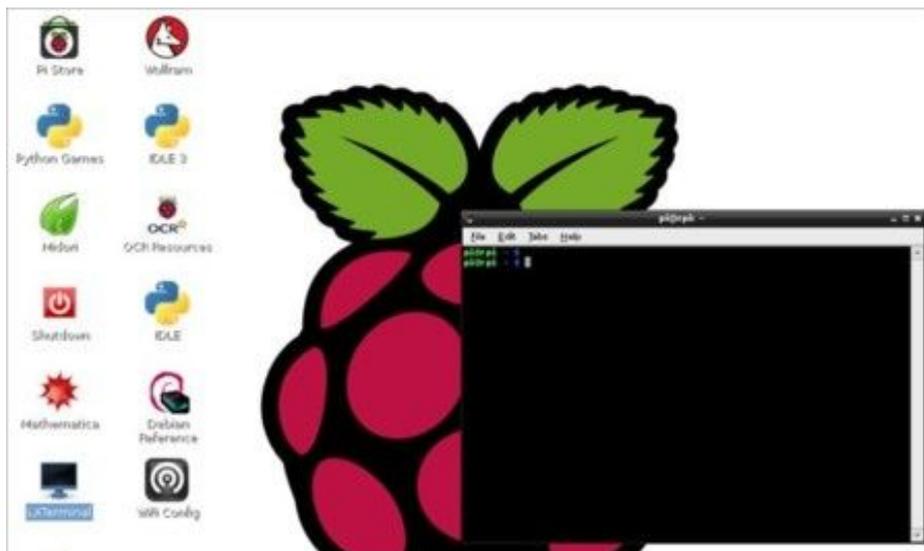
Hay 2 sistemas de numeración de los pins GPIO: **BCM** y **BOARD**. El sistema BCM usa el número de pin GPIO correspondiente. El sistema BOARD la numeración se basa en el orden de los pins de arriba a abajo de la placa (PrototipoPi, 2016).

## 2.3 Sistemas operativos soportados

### 2.3.1 Raspbian

Según (Santos, 2016) es:

Raspbian es una versión de Linux basada en Debian y especialmente desarrollada para Raspberry, y también una de las más populares. En realidad es un poco el "sistema operativo oficial" de este pequeño ordenador. Viene preinstalado con software educativo, para programación y para uso general, como un cliente de email, un navegador web o incluso la suite de ofimática LibreOffice. (Santos, 2016).



**Figura 3** Sistema Operativo Raspbian

**Fuente:** (Santos, 2016)

## 2.4 Real VNC

El estudio de (Sanz, 2016) se encontró:

VNC son las siglas en inglés de Virtual Network Computing (Computación en Red Virtual). Es un programa de software libre basado en una estructura cliente-servidor el cual permite tomar el control del ordenador servidor remotamente a través de un ordenador cliente. También llamado software de escritorio remoto. VNC permite que el sistema operativo en cada computadora sea distinto: Es posible compartir la pantalla de una máquina de “cualquier” sistema operativo conectando desde cualquier otro ordenador o dispositivo que disponga de un cliente VNC portado. (Sanz, 2016)

VNC consta de dos herramientas. Por una parte se tiene el VNC Viewer con el que controlar la Raspberry, el cual está disponible para los sistemas operativos Windows, macOS, GNU/Linux, iOS, Android, Chrome, Solaris, HP-UX o AIX. Y por otra parte el cliente o VNC Server, el cual viene preinstalado por defecto en la versión PIXEL de Raspbian.

Por lo tanto, en esencia, lo que se tiene que hacer es conectar el servidor en tu Raspberry Pi por una parte y luego acceder a él desde el cliente que se ha descargado en tu ordenador o dispositivo móvil. (Sanz, 2016)

### 2.4.1 Características de real VNC:

Según (Martz, 2010) son las siguientes:

A continuación una lista de características de Real VNC:

- Controlar equipos remotamente
- Proteger conexión con contraseña
- Ofrecer soporte remoto
- Transferir archivos
- Imprimir documentos remotamente
- Chatear con otros usuarios conectados
- Agregar direcciones de servidores a favoritos
- Realizar act
- ualizaciones gratuitas (Martz, 2010)

### 2.4.2 Interfaz VNC

El estudio de (Benjamin, 2016) se encontró:

La mayoría de los sistemas de VNC se controlan desde una interfaz gráfica de usuario simple. Una vez que el software ha sido instalado en una computadora remota, se puede acceder desde cualquier ubicación. Un servidor monitorea constantemente las entradas y salidas de una computadora, y las actualiza a la segunda ubicación.

La interfaz para la mayoría de VNCs puede operar desde cualquier sistema operativo, como Windows, Macintosh o Unix. Debido a que la información se transmite a través de Internet, el sistema elimina muchos de problemas tradicionales de plataformas cruzadas. Al igual que un sitio web se puede ver en muchos sistemas operativos diferentes, por lo que puedes operar desde VNC y controlar muchos tipos de equipos diferentes. Mientras que el "VNC" es un término genérico para este tipo de software, existen muchas marcas de software con paquetes disponibles para todos los sistemas operativos más importantes.

La conexión utilizada para datos de interfaz está protegida por una contraseña. Esto evita que los usuarios no autorizados puedan controlar un ordenador a distancia. Una vez establecida la conexión, la información comienza a ser intercambiada. Los detalles sobre la ubicación del ratón, teclado de entrada, y actualizaciones de pantalla son alimentadas a través de la conexión segura, manteniendo los ordenadores sincronizados. (Benjamin, 2016)

### **2.4.3 Usos del VNC**

Según (Benjamin, 2016) es:

El software VNC comenzó como una manera de supervisar y solucionar problemas de sistemas de forma remota, y esta característica se mantiene un uso común. Los técnicos en computación pueden proporcionar soporte técnico mediante VNC sin estar físicamente presentes. Si un usuario tiene un problema al instalar el software, por ejemplo, un profesional de soporte técnico puede conectarse a tu computadora, consultar el problema y hablarle al usuario sobre la solución.

El uso del software VNC también se ha ampliado a otras áreas. El trabajo a distancia se ha convertido en una opción cada vez más popular. VNC le permite a los trabajadores acceder y controlar el software sin estar en la oficina. Por ejemplo, usando VNC, un programador de la India puede controlar una computadora de escritorio en California, en colaboración con sus colegas y trabajar en un proyecto de forma remota. (Benjamin, 2016).

## **2.5 Lenguajes de programación**

El estudio de (Garcia, 2016) se encontró:

Raspberry Pi incorpora diversos lenguajes y/o software de programación algunos de ellos muy populares y enfocados a diferentes públicos, Scratch, Mathematica o Python son algunos de ellos. Por ejemplo Scratch es un lenguaje de programación visual a través de cajas o contenedores ideales para los jóvenes, y permite programar de forma sencilla e intuitiva.

Otro lenguaje es Mathematica muy utilizado para álgebra y programas computacionales.

Tal vez el lenguaje más popular es Python (he leído por ahí que la palabra Pi de nuestra Raspberry Pi, precisamente hace referencia a este lenguaje que se ha hecho muy popular en los últimos años y es una buena opción para las personas que tienen conocimientos de programación y quiere trastear un poco con su Raspberry Pi. En la última versión de Raspbian se ha incorporado más software de programación y siempre podemos obtener otros paquetes de software a través de apt o pi-packag (Garcia, 2016).



**Figura 4** Lenguaje de Programación Python.

**Fuente:** (Garcia, 2016)

### 2.5.1 Python

Según (Drake, 2009) es:

Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas (Drake, 2009).

### 2.5.2 De dónde viene python

El estudio (Alvarez, 2013) se encontró:

El creador del lenguaje es un europeo llamado Guido Van Rossum. Hace más de una década que diseñó Python, ayudado y motivado por su

experiencia en la creación de otro lenguaje llamado ABC. El objetivo de Guido era cubrir la necesidad de un lenguaje orientado a objetos de sencillo uso que sirviese para tratar diversas tareas dentro de la programación que habitualmente se hacía en Unix usando C. El desarrollo de Python duró varios años, durante los que trabajó en diversas compañías de Estados Unidos. En el 2000 disponía de un producto bastante completo y un equipo de desarrollo con el que se había asociado incluso en proyectos empresariales. Actualmente trabaja en Zope, una plataforma de gestión de contenidos y servidor de aplicaciones para el web, por supuesto, programada por completo en Python (Alvarez, 2013).

### 2.5.3 Características del lenguaje python

Según (Alvarez, 2013) son las siguientes:

Las características son las siguientes:

- **Propósito general:** Se pueden crear todo tipo de programas. No es un lenguaje creado específicamente para la web, aunque entre sus posibilidades sí se encuentra el desarrollo de páginas.
- **Multiplataforma:** Hay versiones disponibles de Python en muchos sistemas informáticos distintos. Originalmente se desarrolló para Unix, aunque cualquier sistema es compatible con el lenguaje siempre y cuando exista un intérprete programado para él.
- **Interpretado:** Quiere decir que no se debe compilar el código antes de su ejecución. En realidad sí que se realiza una compilación, pero esta se realiza de manera transparente para el programador. En ciertos casos, cuando se ejecuta por primera vez un código, se producen unos bytecodes que se guardan en el sistema y que sirven para acelerar la compilación implícita que realiza el intérprete cada vez que se ejecuta el mismo código.
- **Interactivo:** Python dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudarnos a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente.

- **Orientado a Objetos:** La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.
- **Funciones y librerías:** Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de strings, números, archivos, etc. Además, existen muchas librerías que podemos importar en los programas para tratar temas específicos como la programación de ventanas o sistemas en red o cosas tan interesantes como crear archivos comprimidos en .zip.
- **Sintaxis clara:** Por último, destacar que Python tiene una sintaxis muy visual, gracias a una notación indentada (con márgenes) de obligado cumplimiento. En muchos lenguajes, para separar porciones de código, se utilizan elementos como las llaves o las palabras clave begin y end. Para separar las porciones de código en Python se debe tabular hacia dentro, colocando un margen al código que iría dentro de una función o un bucle. Esto ayuda a que todos los programadores adopten unas mismas notaciones y que los programas de cualquier persona tengan un aspecto muy similar (Alvarez, 2013).

#### 2.5.4 Programar con python en raspberry pi

Según (Garcia, 2016) se encontró:

Para usar Python en Raspberry Pi se puede optar por dos versiones (Python II o Python III), en la comunidad de programación, Python II continúa siendo muy popular por la gran cantidad de librerías disponible, algo que con la última versión aún no se ha conseguido. De todas formas si se desea aprender un nuevo lenguaje y trastear con la Raspberry Pi por ahora la versión que se escoge no es muy importante, recomendable utilizar Python III (Garcia, 2016).

### 2.5.5 Programar python con IDLE

El estudio (Garcia, 2016) es:

Se utiliza Raspbian se puede programar tanto en Python II como en Python III. La forma más sencilla es encender un dispositivo y dar click en el botón Menú-Programación (Garcia, 2016)



**Figura 5** Python IDLE

**Fuente:** (Garcia, 2016)

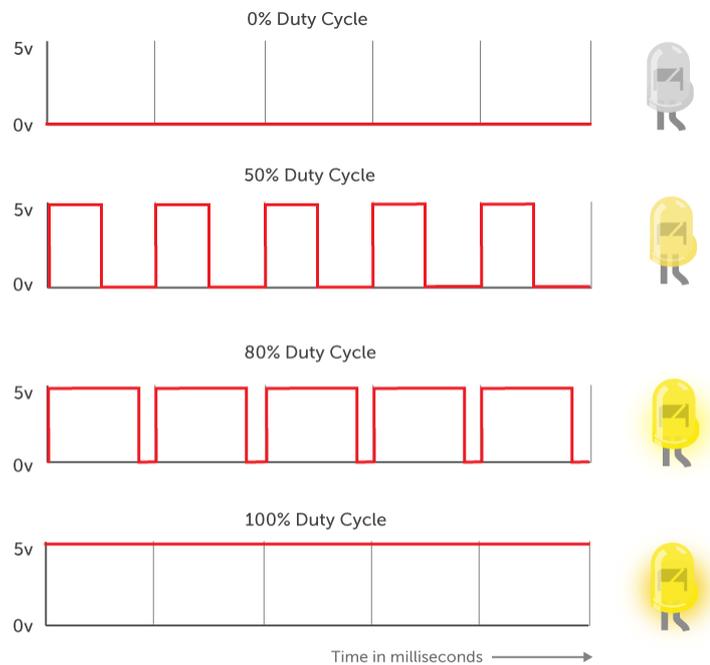
Tal y como se muestra en la imagen se dispone de un acceso directo a **Python 2(IDLE)** y otro a **Python 3(IDLE)**. Solo hacer click encima de la versión escogida (Garcia, 2016).

### 2.6 PWM para raspberry pi

Según (raspberry shop, 2013) se encontró:

La modulación por ancho de pulsos (también conocida como **PWM**, siglas en inglés de pulse-width modulation) de una señal o fuente de energía es una técnica en la que se modifica el ciclo de trabajo de una señal periódica (una senoidal o una cuadrada, por ejemplo), sea para transmitir

información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga (raspberry shop, 2013).



**Figura 6** Python IDLE

**Fuente:** (raspberry shop, 2013).

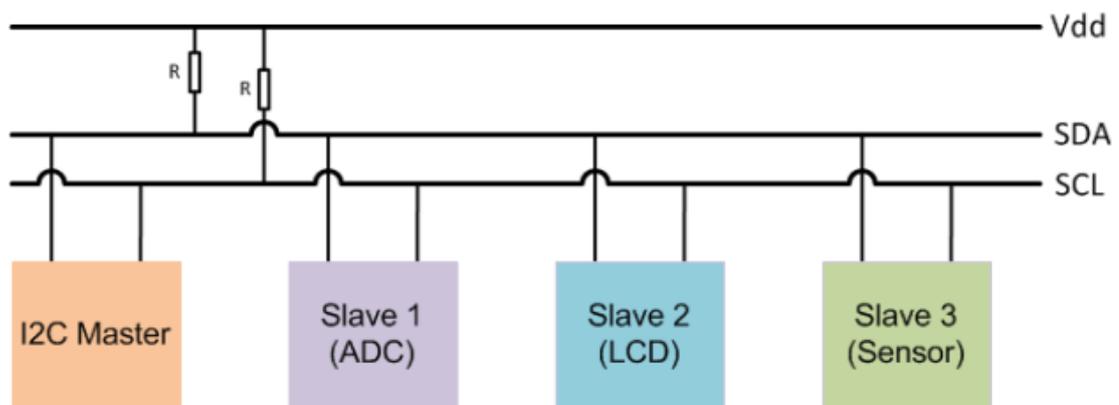
El estudio (ADM, 2017) se encontró:

Modulación de Ancho de Pulso (PWM) es una técnica para obtener resultados analógicos con medios digitales. Control digital se utiliza para crear una onda cuadrada, una señal de conmutación entre encendido y apagado. El patrón de encendido-apagado puede simular voltajes entre completo **en** (3.3 voltios) y **off** (0 voltios) cambiando la porción del tiempo cuando la señal está en frente al tiempo que la señal está apagado. La duración de “a tiempo” se llama el ancho de pulso. Para obtener variando los valores analógicos, es necesario cambiar el ancho de pulso. (ADM, 2017)

## 2.7 Bus i2c de raspberry pi

Según (Martin, 2015) se encontró:

El I2C significa “Inter Integrated Circuits”. Este bus fue desarrollado por la empresa Philips Semiconductors (ahora nxp semiconductors) a principios de los 80’s. Este bus tiene una topología maestro-esclavo. En este caso el maestro es nuestro Raspberry pi (y los esclavos una multitud de dispositivos y sensores que se puede conseguir en el mercado) (Martin, 2015)



**Figura 7** Esquema de Bus

**Fuente:** (Martin, 2015)

El estudio de (Martin, 2015) se encontró:

Las transferencias de datos se llevan a cabo mediante la línea serie de datos **SDA**. La línea **SCL** lleva la señal de reloj que sincroniza los datos de la línea SDA. El dispositivo maestro (microcontrolador, en este caso Raspberry pi) es quien siempre tiene la iniciativa de la comunicación. El maestro genera la señal de reloj y controla cuando se transmiten o reciben los datos.

Puede haber varios esclavos en la red I<sup>2</sup>C, pero el maestro solo se comunica con uno a la vez con un protocolo de pregunta/respuesta. Para poder localizar un dispositivo es necesario que cada uno tenga una dirección única dentro del bus.

En terminología I<sup>2</sup>C los datos se organizan en transferencias. Las transferencias comienzan con una señal de START y termina con otra de STOP (Martin, 2015).

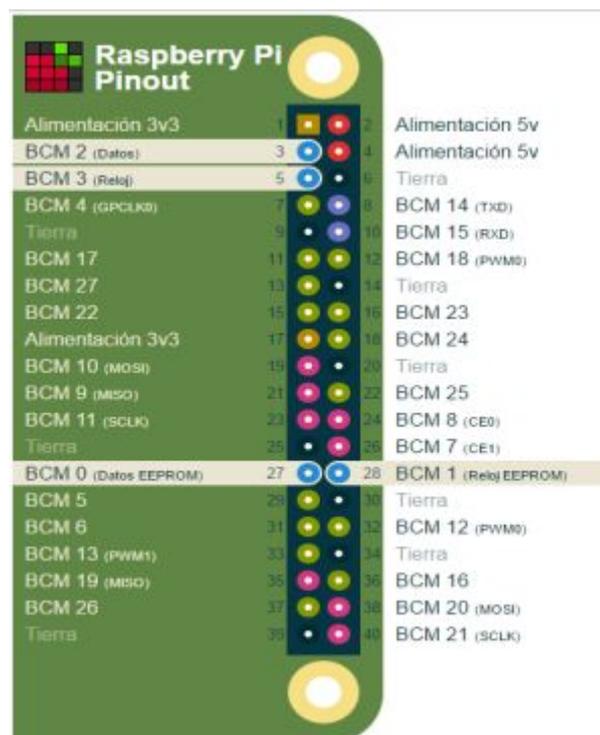
El estándar del bus I<sup>2</sup>C soporta cuatro modos de operación:

- Standard Mode, con una velocidad de hasta 100 kbit/s.
- Fast mode, con una velocidad de hasta 400 kbit/s.
- Fast mode plus, con una velocidad de hasta 1 Mbit/s.
- High-speed mode, con una velocidad de hasta 3.4 Mbit/s.

### 2.7.1 ¿Dónde se encuentra el bus I<sup>2</sup>C en el raspberry?.

Según (Raspberry Pinout, 2016) se encontró:

Se encuentra en el pin 3 (SDA) y el pin 5 (SCL).



**Figura 8** Pines de PWM

**Fuente:** (Raspberry Pinout, 2016)

### 2.7.2 Funciones SMBus (bus de administración del sistema)

Según (Martin, 2015) se encontró:

SMBus (System Management Bus) es un subconjunto del protocolo I2C al escribir un controlador para un dispositivo I2C intente utilizar los comandos SMBus si es posible (si el dispositivo utiliza sólo ese subconjunto del protocolo I2C), ya que hace posible utilizar el controlador de dispositivo en los adaptadores SMBus y los adaptadores I2C. La dirección de la nota es la dirección de 7 bits excluyendo el bit de lectura / escritura (se desplazará a la izquierda 1 bit cuando se agrega al bit de lectura / escritura) (Martin, 2015).

## 2.8 Sensor PCF8591

El estudio de (SunFounder, 2017) es:

El PCF8591 es un dispositivo de adquisición de datos CMOS de 8 bits de baja potencia de un solo chip y de suministro único con cuatro entradas analógicas, una salida analógica y una interfaz de bus serie I2C. Se utilizan tres pines de dirección A0, A1 y A2 para programar la dirección de hardware, permitiendo el uso de hasta ocho dispositivos conectados al bus I2C sin necesidad de hardware adicional. La dirección, el control y los datos hacia y desde el dispositivo se transfieren en serie a través del I2C-bus bidireccional de dos líneas.

Las funciones del dispositivo incluyen multiplexación de entrada analógica, función de pista y retención en chip, conversión de 8 bits analógico a digital y conversión de 8bits digital a analógico. La velocidad máxima de conversión viene dada por la velocidad máxima del bus I2C. (SunFounder, 2017)

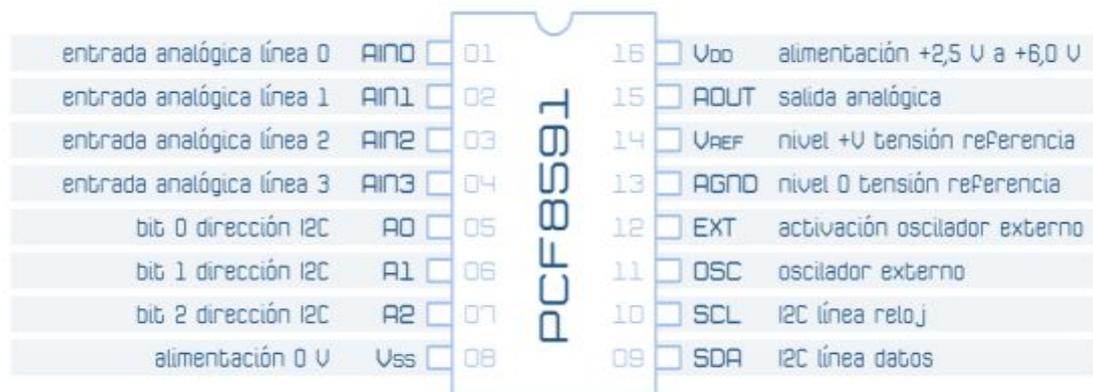
### 2.8.1 Características y benéficos.

El estudio de (SunFounder, 2017) son las siguientes:

- Fuente de alimentación única
- Tensión de alimentación de funcionamiento de 2,5 V a 6,0 V

- Corriente de reserva baja
- Entrada y salida en serie vía I<sup>2</sup>C-bus
- Selección de direcciones I<sup>2</sup>C mediante 3 pines de direcciones de hardware
- Velocidad de muestreo máxima dada por I<sup>2</sup>C Velocidad del bus C
- entradas analógicas configurables como entradas de un solo extremo o diferenciales
- Selección de canales incrementados automáticamente
- Rango de tensión analógica de V<sub>SS</sub> a V<sub>DD</sub>
- Circuito integrado de seguimiento y retención
- Conversión A / D de aproximación sucesiva de 8 bits
- Multiplicación de DAC con una salida analógica. (SunFounder, 2017).

## 2.8.2 Esquema de hardware



**Figura 9** Sensor PFC8591

**Fuente:** (Ventura, 2016).

Según (Ventura, 2016) se encontró lo siguiente:

El PCF8591 dispone de un oscilador interno con el que gestionar el ciclo de conversión analógica a digital. En lugar de utilizar este oscilador puede trabajar con uno externo que se conecta a la entrada OSC; para activarlo, la entrada EXT debe conectarse a nivel bajo. Para utilizar el oscilador

interno, la configuración más habitual, basta con dejar al aire la patilla OSC y conectar a nivel alto EXT.

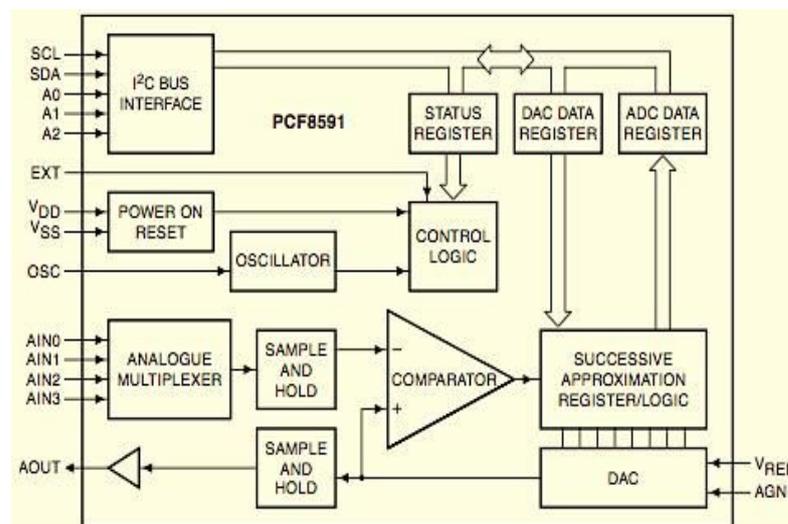
Para establecer la referencia de medida de tensión del PCF8591 utiliza las entradas VREF y AGND. En la configuración más básica se puede conectar VREF a VDD y AGND a VSS, es decir, referirlo a la propia tensión de alimentación del circuito.

En el diagrama de ejemplo de la siguiente imagen se utiliza la conexión más habitual (y más sencilla). La dirección del dispositivo en el bus I2C es 0B1001000 por establecer a cero A0, A1 y A2. La tensión de referencia es la de alimentación del sistema y se utiliza el oscilador interno al conectar EXT a VDD. (Ventura, 2016).

### 2.8.3 Descripción básica del circuito

El estudio de (Diverteka, 2015) es:

El circuito PCF8591 incorpora 4 entradas analógicas y una salida gestionables remotamente mediante bus serie I2C. Opera con un bajo consumo (8-bits CMOS) y puede ser direccionado mediante 3 patillas al efecto (en el módulo estas ya se han pre-ajustado). Internamente el chip se encarga de multiplexar la información que captura en cada entrada y almacenarla para su lectura vía bus I2C. (Diverteka, 2015).



**Figura 10** Descripción del PFC9591

**Fuente:** (Diverteka, 2015)

## CAPÍTULO III

### DESARROLLO DEL PROYECTO

#### 3.1 Preliminares

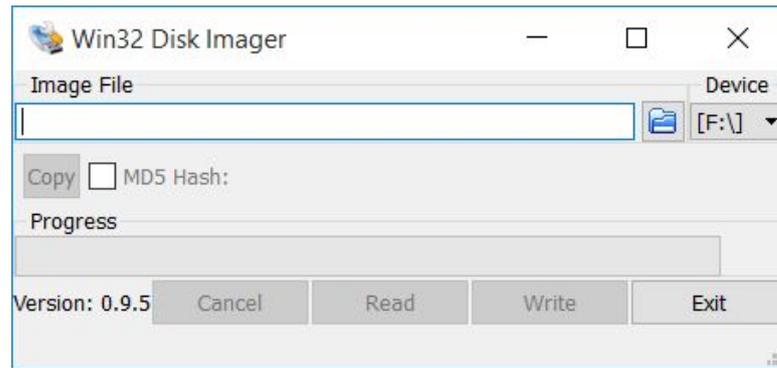
En el siguiente capítulo se explica en forma clara y concreta los pasos a realizar para el control del módulo de temperatura utilizando las herramientas y equipos adecuados.

#### 3.2 Herramientas y equipos utilizados

- Raspberry Pi3 Modelo B
- Sensor PCF8591
- Protoboard
- Cables
- Monitor
- Mouse
- Teclado
- Tarjeta SD de 16 GB
- Resistencias10K
- LCD 16x2
- Potenciómetro10K

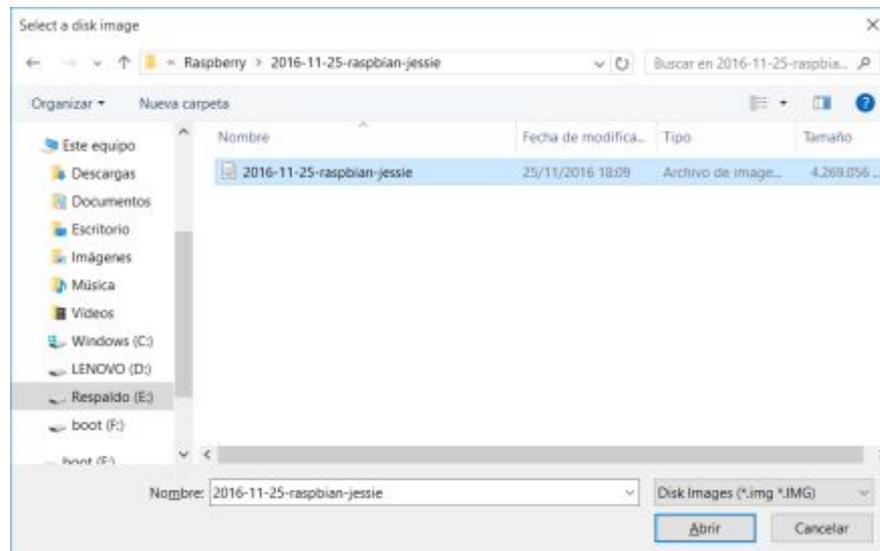
##### 3.2.1 Instalación de sistema operativo al raspberry pi3 modelo b

Se descarga el sistema operativo Debian Reference Versión 2 distribución de Linux en la computadora. Para copiar la imagen en la tarjeta SD primero dar click en F (ver figura 11)



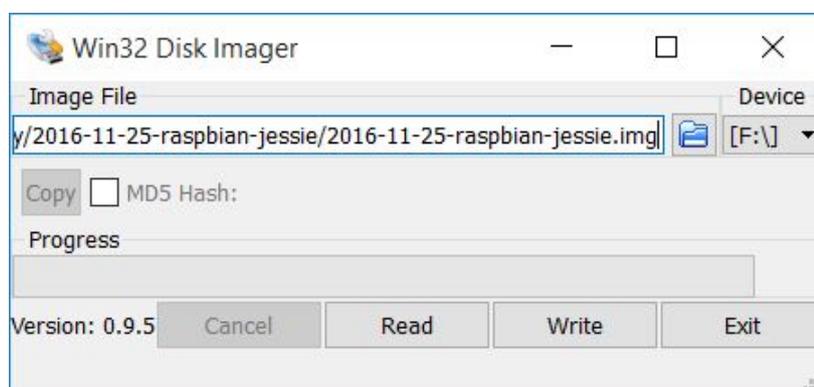
**Figura 11** Ventana de elegir la dirección

Se elige la dirección donde está ubicada el software dar doble click



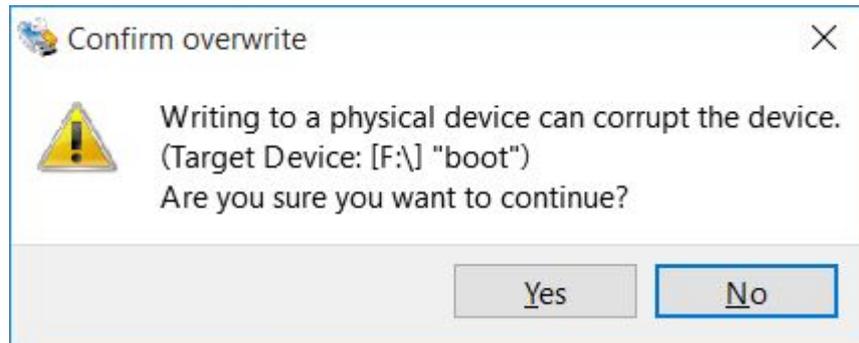
**Figura 12** Ubicación del software

Aparece la dirección de raspbian, click en Write



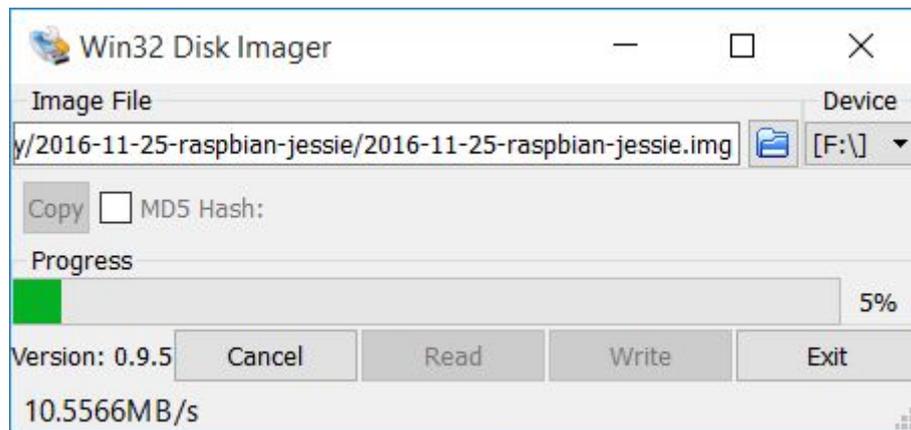
**Figura 13** Selección de Raspbian

Click Yes para confirmar overwrite



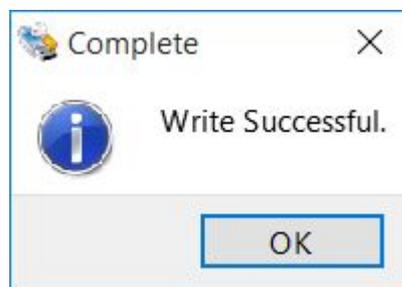
**Figura 14** Confirmación

Esperar que se cargue al 100% la imagen



**Figura 15** Cargando la imagen

Una vez completado dar click en Ok



**Figura 16** Finalización de la imagen

Insertar la tarjeta SD al raspberry pi, conectar el teclado, mouse, HDMI. Al encender el raspberry pi se instala automáticamente la imagen y segundos después se observa la interfaz en la cual se puede trabajar.



**Figura 17** Insertar la tarjeta SD al Raspberry Pi

Encender la pantalla y automáticamente la imagen del software se instala ver (figura18)



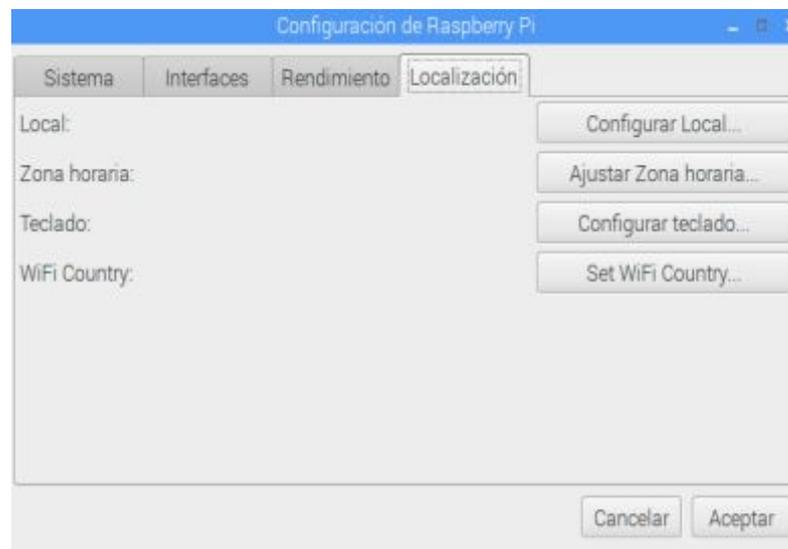
**Figura 18** Instalación de Debian.

Proceder a la configuración del raspberry pi, click en el icono elegir preferencias e ir a configuraciones



**Figura 19** Configuración del raspberry pi.

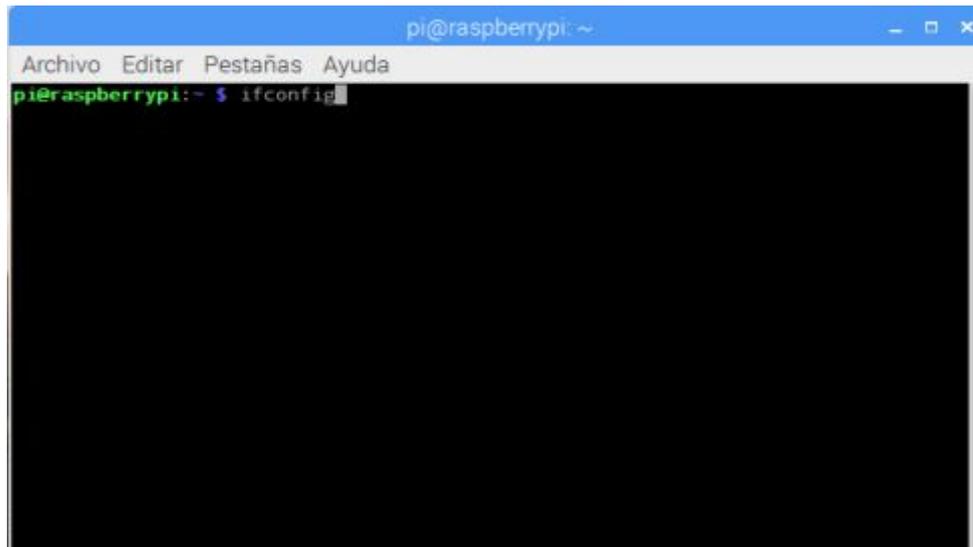
En localización configurar hora, fecha, zona horaria y aceptar los cambios que se realizan en ese momento. Y se puede utilizar el raspberry pi con normalidad.



**Figura 20** Configuración de localización.

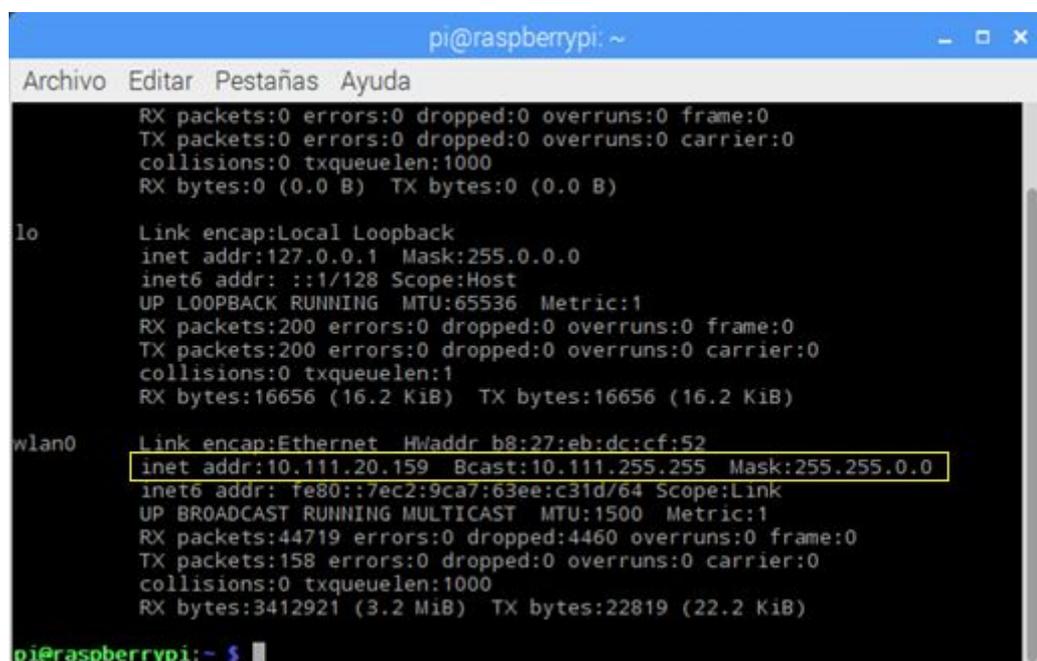
### 3.2.2 Instalación y configuración del VNC

Ingresa a la ventana del terminal escribir lo siguiente `if config` y dar enter (ver figura 21)



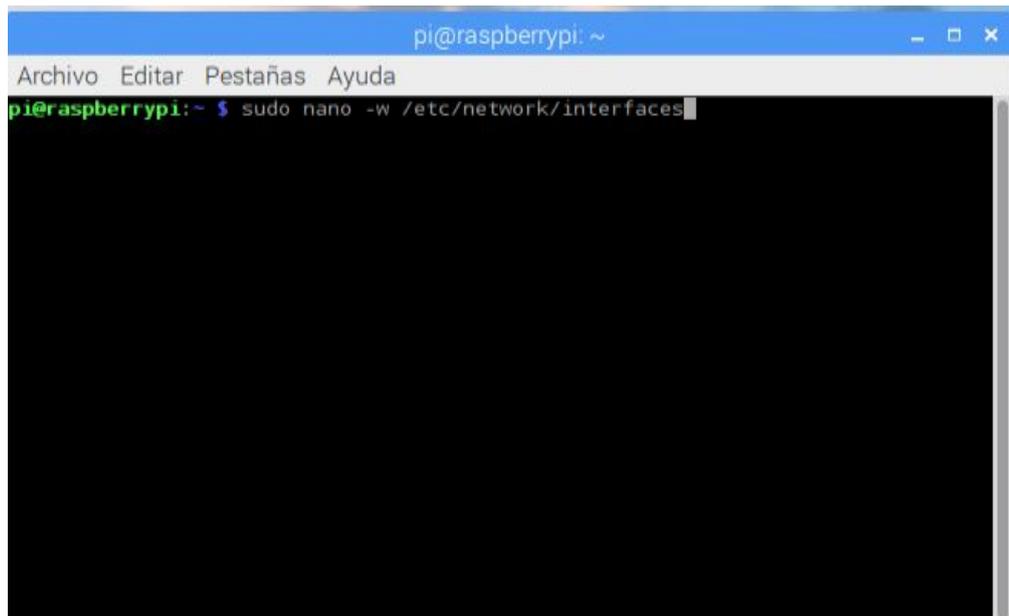
**Figura 21** Ventana del Terminal.

En la siguiente venta se despliega las direcciones IP y se observa la dirección de Wlan0 que están conectados, esta dirección es la del raspberry pi.



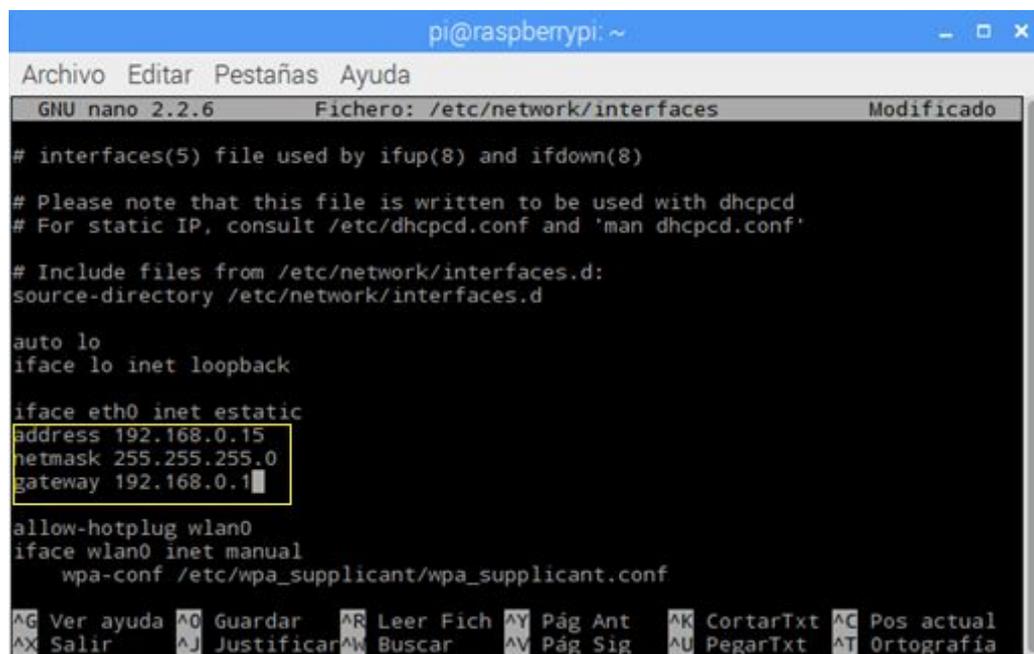
**Figura 22** Dirección de wlan0.

Abrir otra ventana de terminal e ingresar el siguiente código (ver figura 23) para ingresar al interfaz del raspberry.



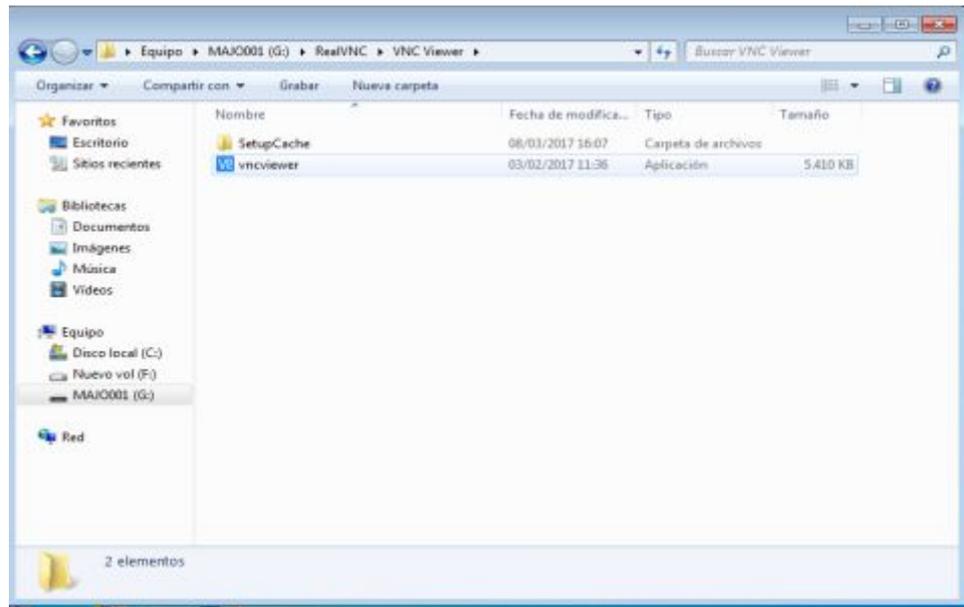
**Figura 23** Ingreso de código para interfaz.

Escribir después de la línea `iface eth0 inet static` el siguiente código (ver figura24) .Luego orpimir `ctrl +O` y `enter` para guardar.



**Figura 24** Ingreso de dirección IP.

Trasladar el raspberry pi a un PC en la que se va instalar el VNC conectar y abrir el siguiente software vncviewer lo que va ser la interfaz de la computadora con el raspberry.



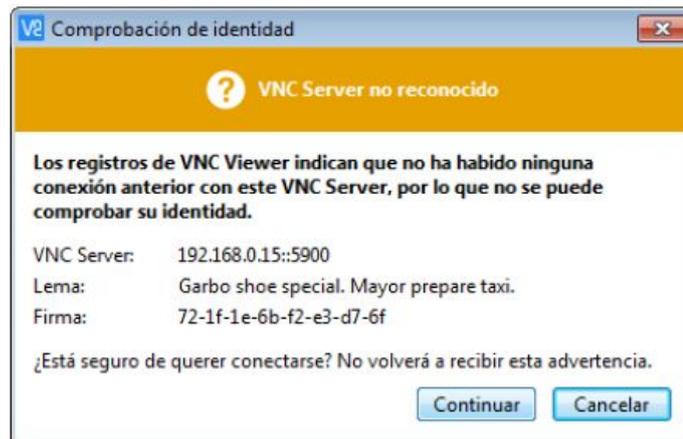
**Figura 25** Software VNCViewer.

A continuación, ingresar la dirección IP del raspberry pi 192.168.0.15 para conectarse con el raspberry



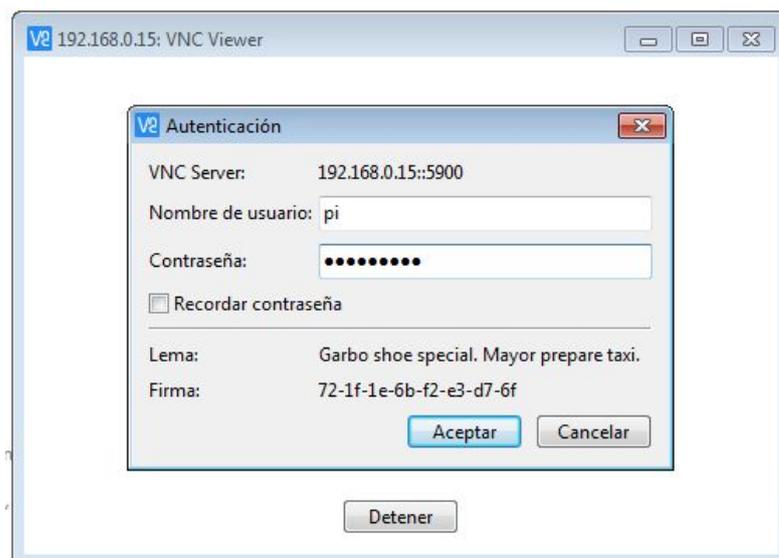
**Figura 26** Ingreso de dirección IP del raspberry pi.

Después aparece la siguiente ventana (ver figura 27) de comprobación dar click en continuar para seguir con el proceso.



**Figura 27** Ventana de comprobación.

Ingresa el usuario y contraseña para la que se va utilizar para la sesión del VNC.



**Figura 28** Ingreso de usuario y contraseña.

Una vez configurado se reinicia el raspberry y al iniciarse se tendrá el interfaz del raspberry en la PC (ver figura 29).



**Figura 29** Comunicación de raspberry pi y PC

### 3.2.3 Configuración del sensor PCF8591

Ingresar al terminal y escribir el siguiente código (ver figura 30)

```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~ $ sudo nano /etc/modprobe.d/raspi-blacklist.conf
```

**Figura 30** Ingreso de código en el terminal.

Comentar las siguientes líneas de código en la ventana del terminal para eso ingresar # seguido el código.

```

pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
GNU nano 2.2.6 Fichero: /etc/modprobe.d/raspi-blacklist.conf
blacklist spi-bcm2708
#blacklist i2c-bcm2708
[ 3 líneas leídas ]
AG Ver ayuda ^O Guardar ^R Leer Fich ^Y Pág Ant ^K CortarTxt ^C Pos actual
AX Salir ^J Justificar ^W Buscar ^V Pág Sig ^U PegarTxt ^T Ortografía

```

**Figura 31** Comentar líneas de código.

Después ingresar `sudo i2cdetect -y 1` y enter, aparecerá unos números detectando el dispositivo conectado al raspberry y se muestra la dirección (ver figura 32)

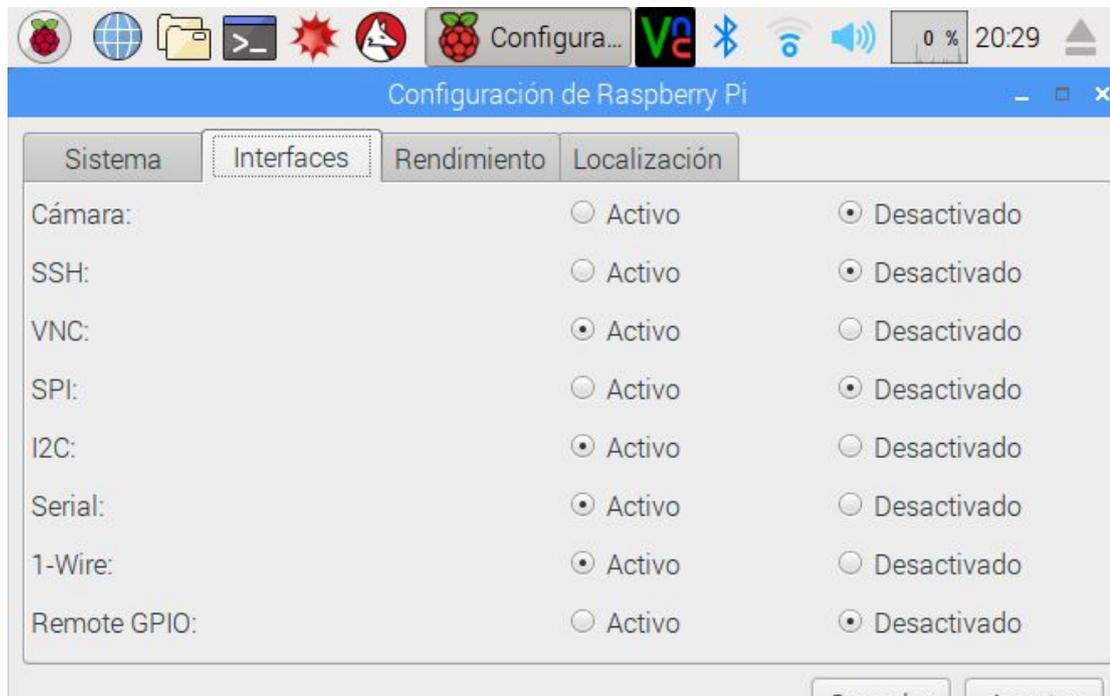
```

pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~$ sudo i2cdetect -y 1
 0 1 2 3 4 5 6 7 8 9 a b c d e f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- 48 -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
pi@raspberrypi:~$

```

**Figura 32** Comentar líneas de código.

Después ir a configuración del raspberry activar VNC,I2C, 1Wire. Y guardar los cambios.



**Figura 33** Activación de interfaces.

Una vez configurado, se programa en Python e ingresar el siguiente código. (Ver figura 34)

Import time: llama la librería de tiempo

Print: imprime lo requerido

Bus.write indica el canal que se utiliza del sensor

```

potenciometro.py - /home/pi/potenciometro.py (2.7.9)
File Edit Format Run Options Windows Help

#Read a value from analogue input 0
#in A/D in the PCF8591P @ address 0x48
import time
from smbus import SMBus

bus = SMBus(1)

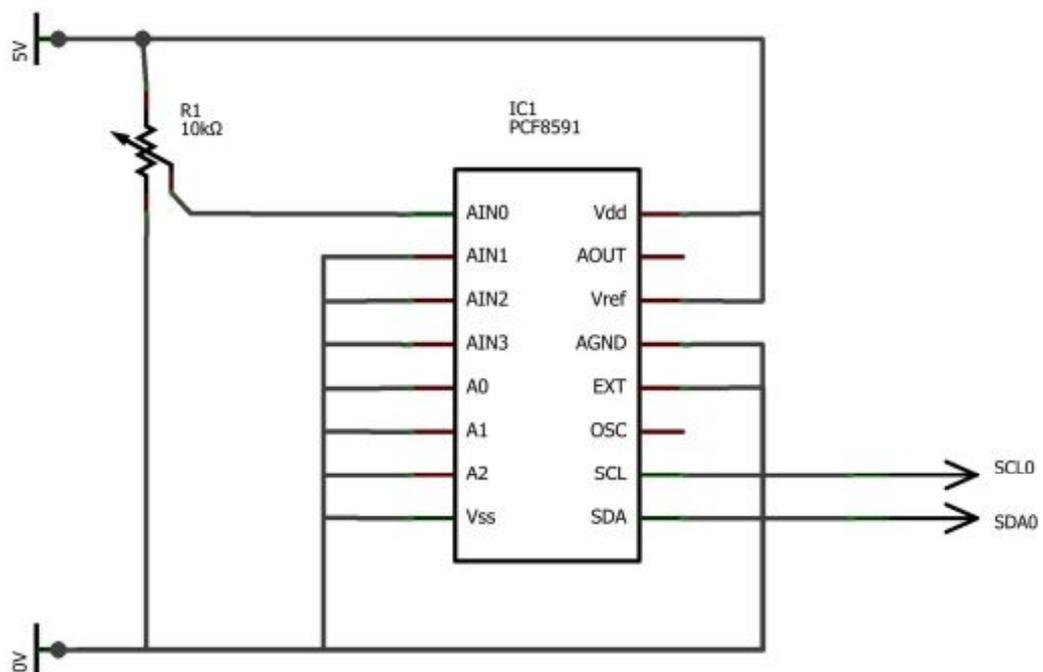
print("Read the A/D")
print("Ctrl C to stop")
bus.write_byte(0x48, 0) # set control register to read channel 0
last_reading = -1

while(0 == 0): # do forever
    reading = bus.read_byte(0x48) # read A/D
    if(abs(last_reading - reading) > 2):
        print(reading)
        last_reading = reading

```

**Figura 34** Código de programación del PCF8591.

Proceder armar el siguiente circuito para la comprobación del sensor y la programación.



**Figura 35** Circuito del PFC8591.

Conectar los pines del PCF8591 al raspberry según la siguiente tabla.

**Tabla 1** Especificaciones de los pines dePCF8591

Símbolo	Pin	Descripción
<b>AIN0</b>	1	Entrada analógica
<b>AIN1</b>	2	Entrada analógica
<b>AIN2</b>	3	Entrada analógica
<b>AIN3</b>	4	Entrada analógica
<b>A0</b>	5	Dirección de esclavo
<b>A1</b>	6	Dirección de esclavo
<b>A2</b>	7	Dirección de esclavo
<b>Vss</b>	8	Alimentación a tierra
<b>SDA</b>	9	I2c datos serial E/S
<b>SCL</b>	10	I2c reloj serial entrada
<b>OSC</b>	11	Oscilador E/S
<b>EXT</b>	12	E/S switch para OSC
<b>AGND</b>	13	Alimentación a tierra analógica
<b>Vref</b>	14	Entrada de referencia de voltaje
<b>AOUT</b>	15	Salida analógica
<b>Vdd</b>	16	Tensión de alimentación

### 3.2.4 Configuración y programación del LCD

Realizar la programación en Python lo que significa lo siguiente:

- **Import (os, glob, time):** Son para llamar las librerías a utilizar para el uso del LCD.
- **Import CharLCD:** Librería del LCD

Una vez obtenidas las librerías en la quinta línea de programación se especifica los pines que se va utilizar durante la ejecución del LCD.

- **While True:** Mientras sea verdadero se ejecutara
- **Lcd.cursor\_pos:** Especificación de fila y columna del LCD
- **Lcd.write\_string:** Lo que aparecerá en la pantalla del LCD

```

pruebaLCD.py - /home/pi/pruebaLCD.py (2.7.9)
File Edit Format Run Options Windows Help
import os
import glob
import time
from RPLCD import CharLCD

lcd = CharLCD(cols=16, rows=2, pin_rs=37, pin_e=35, pins_data=[33,31,29,23])

while True:
    lcd.cursor_pos = (0,0)
    lcd.write_string("Majo" + unichr(223) + "C")
  
```

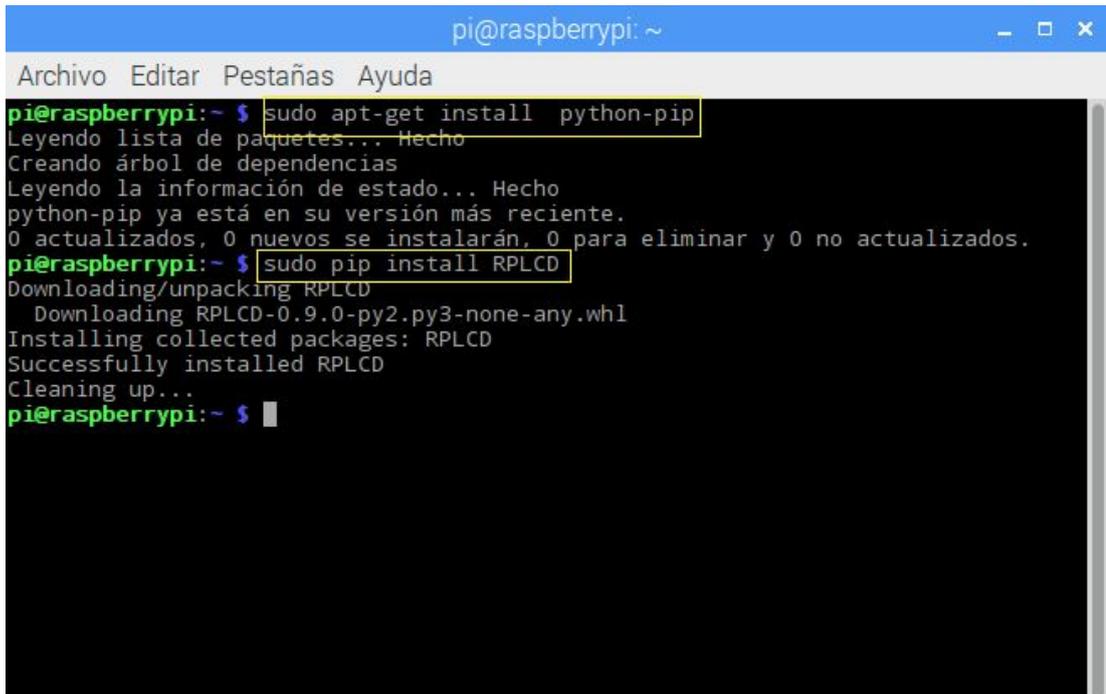
**Figura 36** Código de programación del LCD

Realizar la siguiente conexión (ver tabla)

**Tabla 2** Conexión del LCD con raspberry pi

Raspberry Pi	LCD Display
Ground	Vss
5V	Vdd
Ground	Vo
GPIO26	Rs
Ground	Rw
GPIO19	Enabled
GPIO13	D4
GPIO06	D5
GPIO05	D6
GPIO11	D7
Ground	A
5V	K

Ingresar al terminal y escribir el siguiente código (ver figura 37)



```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~$ sudo apt-get install python-pip  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
python-pip ya está en su versión más reciente.  
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.  
pi@raspberrypi:~$ sudo pip install RPLCD  
Downloading/unpacking RPLCD  
  Downloading RPLCD-0.9.0-py2.py3-none-any.whl  
Installing collected packages: RPLCD  
Successfully installed RPLCD  
Cleaning up...  
pi@raspberrypi:~$
```

**Figura 37** Configuración en el terminal para LCD

Se confirma que la configuración del LCD en el raspberry este instalada en esta versión automáticamente.

### 3.2.5 Programación del control de temperatura

Realizar el siguiente código de programación donde se une las dos programaciones anteriores del LCD y PCF8591 y adicional el control del módulo de temperatura que se va ser por PWM. Para utilizar PWM en raspberry pi conectar el pin 12 BOARD que también es GPIO18 BCM

```

prueba 2canal.py - /home/pi/prueba 2canal.py (2.7.9)
File Edit Format Run Options Windows Help
#Read a value from analogue input 0
#in A/D in the PCF8591P @ address 0x48
import RPi.GPIO as GPIO
import os
import glob
import time
from smbus import SMBus
from RPLCD import CharLCD

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)

GPIO.setup(12, GPIO.OUT)
rojo = GPIO.PWM(12, 100)
rojo.start(0)

lcd = CharLCD(cols=16, rows=2, pin_rs=37, pin_e=35, pins_data=[33,31,29,23])

bus = SMBus(1)

while True: # do forever
    bus.write_byte(0x48, 0) # set control register to read channel 0
    reading = bus.read_byte(0x48) # read A/D
    b=int((reading*50)/255+20)
    print("B= " + str(b))

    bus.write_byte(0x48, 1) # set control register to read channel 1
    reading1 = bus.read_byte(0x48) # read A/D
    a=int((reading1*50)/255+20)
    print("A= " + str(a))

    if(a<b):
        rojo.ChangeDutyCycle(100)
    else:
        rojo.ChangeDutyCycle(0)

    lcd.cursor_pos = (0, 0)
    lcd.write_string("VP: " + str(a) + unichr(223) + " C")
    lcd.cursor_pos = (1, 0)
    lcd.write_string("SP: " + str(b) + unichr(223) + " C")

    time.sleep(0.2)

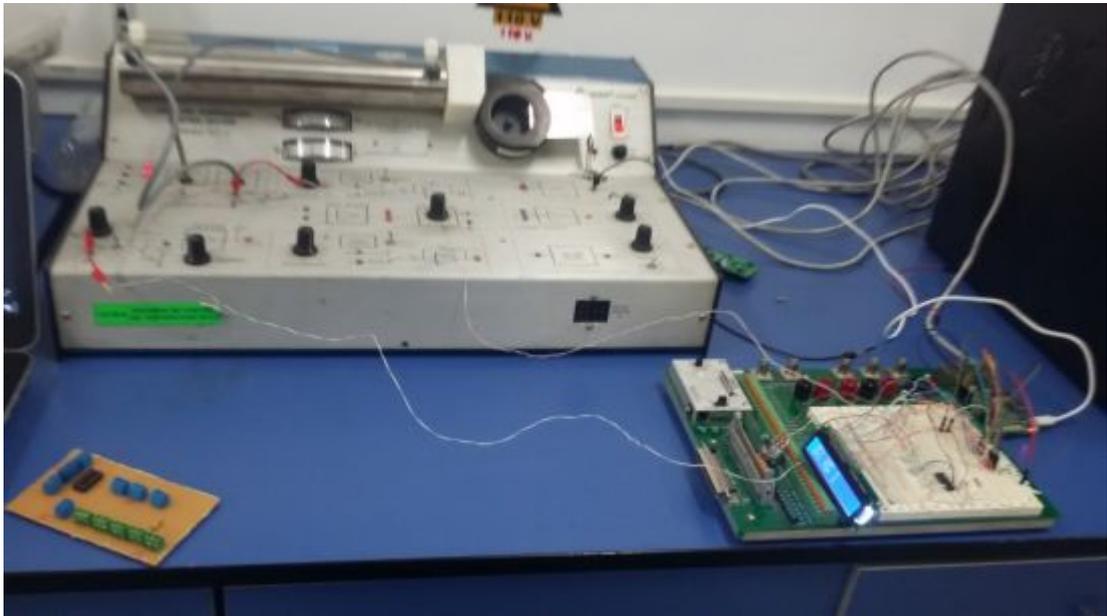
```

**Figura 38** Programación para el control de módulo de temperatura

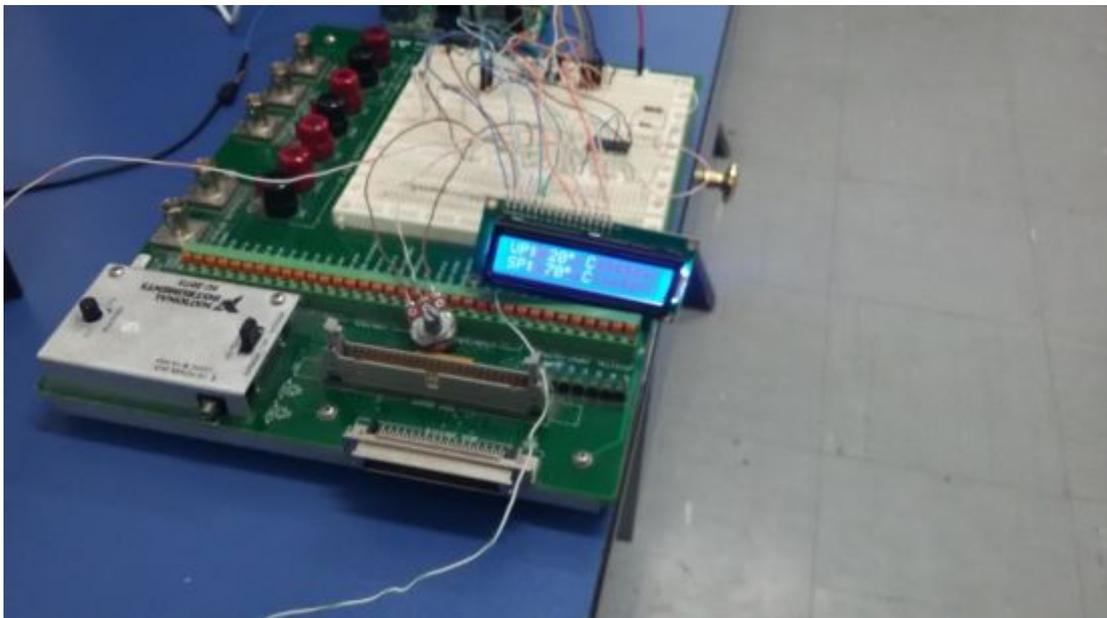
Para la utilización del PWM tener las siguientes líneas de código .Primero importar las librerías con **import** para el LCD, PCF8591 y PWM.

- **GPIO.setmode (GPIO.BOARD):** Establezca Pi para usar el número de pin al referenciar pines GPIO
- **GPIO.setup (12, GPIO.OUT):** Ponemos el pin GPIO nº12 como salida para el LED
- **GPIO.setwarnings:**

- **Rojo.start (0):** Iniciar el objeto al 0% del ciclo de trabajo (completamente encendido).



**Figura 39** Comprobación del sensor con el modulo



**Figura 40** Demostración de temperatura y set point en LCD

```

Python 2.7.9 Shell
File Edit Shell Debug Options Windows Help
A= 20
B= 70
A= 21
B= 70
A= 20
B= 70
A= 20
B= 70
A= 21
B= 70
A= 21
B= 70
A= 21
B= 70
A= 20
B= 70
A= 20
B= 70
A= 21
Ln: 6 Col: 0

```

**Figura 41** Resultados en Python

### 3.3 Prácticas para microcontroladores usando el raspberry pi3

- Lectura de temperatura de un sensor DS18B20
- Control domótica
- Controles automáticos con comunicación I2C
- Control de mando a distancia
- Control de motores ac/dc

#### 3.3.1 Lectura de temperatura de un sensor DS18B20

- **Materiales**

- Raspberry Pi3 Modelo B
- DS18B20 con carcasa de acero inoxidable
- Protoboard
- Cables de conexión
- Resistencias

- **Esquema de montaje DS18B20 y Raspberry PI**

La alimentación del sensor mediante el pin de 3,3v del conector GPIO (pin 1), al clave rojo del sensor. El pin 6 del GPIO, y conectar al cable negro, o al

pin de la derecha del sensor. La salida de datos la conectar al pin 4, al cable amarillo, o al pin del medio del sensor. Para terminar el montaje, colocar la resistencia de 4K7 Ohm, entre el pin positivo del sensor y el pin de datos.

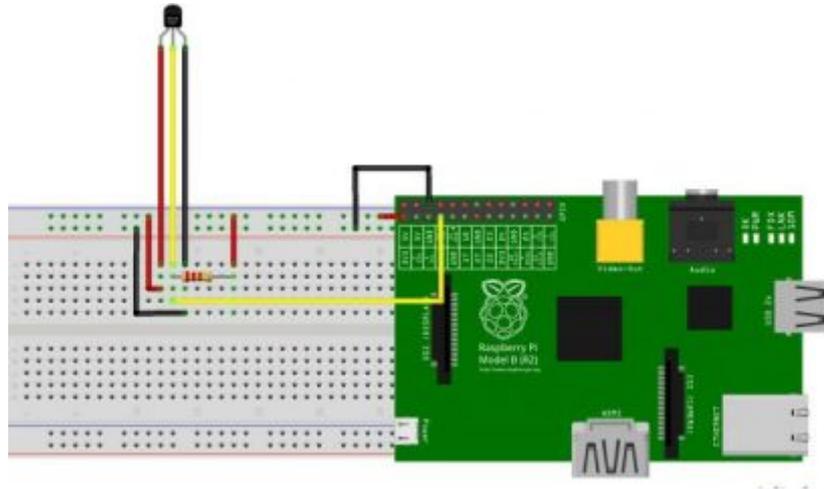


Figura 42 Esquema de ds18b20 con raspberry

- Programación en Python

```

ds18b20_prueba.py - /home/pi/ds18b20_prueba.py (2.7.9)
File Edit Format Run Options Windows Help
#Una funcion que lee datos del sensor
def read_temp_raw():

    f= open(device_file,'r') #Abre el archivo del dispositivo de temperatura
    lines= f.readlines() #Devuelve el texto
    f.close()

    return lines

#Convierte el valor del sensor en temperatura
def read_temp():
    lines= read_temp_raw() #Lee la temperatura del archivo de dispositivo

    #Mientras que la primera linea no contenga 'YES', espera 0,2 s
    #y luego lee el archivo de dispositivo de nuevo
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
        lines= read_temp_raw()

    #Busca la posicion del '=' en la segunda linea del archivo del dispositivo
    equals_pos= lines[1].find('t=')

    #Si se encuentra '=' convierte el resto de la linea despues de '='
    #en grados centigrados, luego en grados Fahrenheit

    if equals_pos != -1:
        temp_string= lines[1][equals_pos+2:]
        temp_c= float(temp_string)/1000.0
        temp_f= temp_c*9.0/5.0+32.0

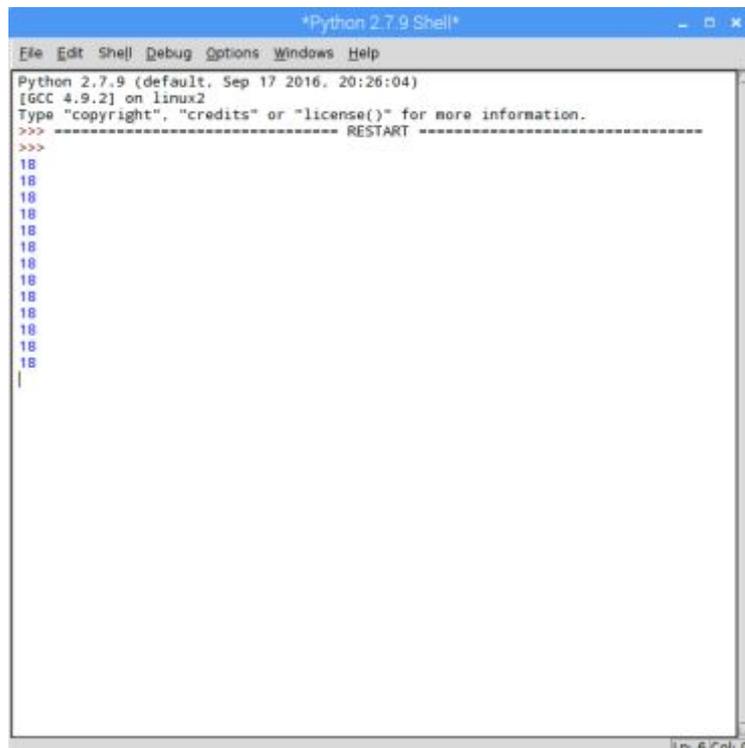
    return temp_c,temp_f

#Imprime la temperatura hasta que el programa se detenga
while True:
    print(read_temp())
    time.sleep(1)

```

Figura 43 Programación en Python

- **Resultados de la lectura del sensor DS18B20**



```
Python 2.7.9 Shell
File Edit Shell Debug Options Windows Help
Python 2.7.9 (default, Sep 17 2016, 20:26:04)
[GCC 4.9.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> ----- RESTART -----
>>>
18
18
18
18
18
18
18
18
18
18
18
18
18
18
18
18
|
```

**Figura 44** Resultados del sensor DS18B20

## CAPÍTULO IV

### CONCLUSIONES Y RECOMENDACIONES

#### 4.1 Conclusiones

- Se investigó la parte teórica en sitios web, pues este dispositivo es actual y la información obtenida es muy poca pero concreta y precisa para tener mayor conocimiento al utilizar el raspberry pi. Facilita la conexión y programación adecuada para ser manipulado.
- El sistema operativo Raspbian una versión de LINUX, es un software libre, de fácil uso y manejo, solo basta con descargar la imagen a la PC se instala automáticamente. Es compatible con el raspberry pi brindando todas las herramientas adecuadas para la manipulación del usuario.
- El control del módulo de temperatura se realizó mediante PWM del raspberry pi, cual es configurado para que realice la función correctamente con referencia a un set point cual es tomada del PCF8591. Está programado en un lapso que el valor de proceso mientras este menor al set point seguirá encendido caso contrario se apagará.
- Se realizó una implementación de un control de temperatura mediante el microcomputador raspberry pi para que los estudiantes puedan realizar prácticas de microcontroladores y obtener más conocimientos de dispositivos tecnológicos.

## 4.2 Recomendaciones

- Para manipular el raspberry pi con el módulo de temperatura analizar la programación para observar los pines que se están utilizando y conectar máximo de 5V para evitar que se quemé el PCF8591.
- Tener en cuenta las fuentes bibliográficas para consultar sobre los dispositivos ocupados para tener conocimiento y poder utilizar, para no provocar errores en el funcionamiento.
- Para la descargar del sistema operativo del raspberry pi hacerlo en la página web principal para que no haya errores al momento de instalar. También tener en cuenta que versión del raspberry pi para descargar una versión actual que sea compatible.
- La conexión del PWM realizarlo en el pin correspondiente del raspberry pi, para no haya una mala manipulación del mismo. Es recomendable utilizar el pin 12 del raspberry pi para el PWM que es el más usado y factible. Tener cuidado con las utilizaciones de pines pues cada pin tiene diferente funcionamiento.

## GLOSARIO DE TÉRMINOS

**CMOS:** El semiconductor complementario de óxido metálico o complementary metal-oxide-semiconductor (CMOS) es una de las familias lógicas empleadas en la fabricación de circuitos integrados. Su principal característica consiste en la utilización conjunta de transistores de tipo pMOS y tipo nMOS.

**GND:** Ground es la palabra inglesa para hablar de Tierra. tierra es el punto de referencia en un circuito eléctrico a partir del cual se miden las tensiones, una vía de retorno común para la corriente eléctrica o una conexión física directa a la Tierra .

**GPIO:** Entrada / salida de uso general; en este caso específico los pines de la frambuesa Pi y lo que puede hacer con ellos. Así se llama porque puedes usarlos para todo tipo de propósitos; la mayoría se puede utilizar como entradas o salidas, dependiendo de su programa.

**IDLE:** Es el Integrated DeveLopment Environment para Python, es decir, el entorno de desarrollo integrado para Python, que trae esta herramienta de programación.

**I2C:** Un circuito interintegrado (I<sup>2</sup>C, del inglés Inter-Integrated Circuit) es un bus serie de datos desarrollado en 1982 por Philips Semiconductors. Se utiliza principalmente internamente para la comunicación entre diferentes partes de un circuito.

**LCD:** Liquid Crystal Display o pantalla de cristal líquido es un dispositivo empleado para la visualización de contenidos o información de una forma gráfica, mediante caracteres, símbolos o pequeños dibujos dependiendo del modelo.

**PWM:** La modulación de ancho de pulso (PWM, por sus siglas en inglés) de una señal es una técnica que logra producir el efecto de una señal analógica sobre una carga, a partir de la variación de la frecuencia y ciclo de trabajo de una señal digital.

**SCL:** System Clock por la cual transitan los pulsos de reloj que sincronizan el sistema.

**SDA:** System Data por la cual viajan los datos entre los dispositivos.

**SMBUS:** Bus de Administración del Sistema, es un subconjunto del protocolo I<sup>2</sup>C. Usado para la comunicación de dispositivos de bajo ancho de banda en la placa madre, especialmente en chips relacionados a la energía como el subsistema de baterías.

**VDD:** Tensión positiva de alimentación.

**VNC:** Son las siglas en inglés de Virtual Network Computing (Computación Virtual en Red). Es un programa de software libre basado en una estructura cliente-servidor que permite tomar el control del ordenador servidor remotamente a través de un ordenador cliente.

## REFERENCIAS BIBLIOGRÁFICAS

- ADM. (marzo de 2017). *ADMfactory*. Obtenido de <http://www.admfactory.com/breathing-light-led-on-raspberry-pi-using-python/>
- Alvarez, M. A. (19 de noviembre de 2013). *Desarrolloweb.com*. Obtenido de <https://desarrolloweb.com/articulos/1325.php>
- Benjamin, A. (2016). *eHow*. Obtenido de [http://www.ehowenespanol.com/funciona-vnc-como\\_155984/](http://www.ehowenespanol.com/funciona-vnc-como_155984/)
- Contreras, L. (18 de diciembre de 2013). *Blog Historia de la informatica*. Obtenido de <http://histinf.blogs.upv.es/2013/12/18/raspberry-pi/>
- Diverteka*. (febrero de 2015). Obtenido de <http://www.diverteka.com/?p=1814>
- DIY Makers*. (2016). Obtenido de <http://diymakers.es/usando-el-puerto-gpio/>
- Drake, F. (septiembre de 2009). *Tutorial de Python*. Obtenido de <http://docs.python.org.ar/tutorial/pdfs/TutorialPython2.pdf>
- electrónico, c. (s.f.). *monografias*. Obtenido de <http://www.monografias.com/trabajos82/circuito-electrico/image003.jpg>
- Garcia, A. (06 de enero de 2016). *frambuesa pi*. Obtenido de <http://frambuesa-pi.blogspot.com/2016/01/programar-con-raspberrypi-python.html>
- Martin, C. (Agosto de 2015). *TIECS*. Obtenido de <http://www.cmdearcos.es/bus-i2c-de-raspberry-pi-1/>
- Martz, R. (10 de Julio de 2010). *Tecnologias de Informacion*. Obtenido de <http://tiwindows-renan.blogspot.com/2010/07/real-vnc.html>
- PE, I. (26 de mayo de 2015). *comohacer.eu*. Obtenido de <http://comohacer.eu/gpio-raspberry-pi/>

*PrototipoPi*. (1 de mayo de 2016). Obtenido de

<https://prototipopi.wordpress.com/2016/05/02/puertos-gpio/>

*RASPBERRY PI 3 MPODELO B*. (2016). Obtenido de

<https://translate.google.com.ec/translate?hl=es&sl=en&u=https://www.raspberrypi.org/products/raspberry-pi-3-model-b/&prev=search>

*Raspberry Pinout*. (2016). Obtenido de <https://es.pinout.xyz/pinout/i2c>

*raspberry shop*. (26 de septiembre de 2013). Obtenido de

<https://www.raspberrishop.es/wp/software-pwm-en-una-raspberry-pi/>

Santos, E. (26 de julio de 2016). *GENBETA*. Obtenido de

<https://www.genbeta.com/herramientas/9-sistemas-operativos-y-gestores-de-contenido-que-puedes-instalar-en-una-raspberry-pi>

Sanz, J. (2016). *SoftZone*. Obtenido de <https://www.softzone.es/manuales-software-2/ultravnc-controlar-de-forma-remota-un-ordenador-con-ultravnc/>

*SunFounder*. (14 de Abril de 2017). Obtenido de

<https://translate.google.com.ec/translate?hl=es&sl=en&u=https://www.sunfounder.com/learn/sensor-kit-v2-0-for-raspberry-pi-b-plus/lesson-13-pcf8591-sensor-kit-v2-0-for-b-plus.html&prev=search>

Ventura, V. (14 de Abril de 2016). *Polaridad.es*. Obtenido de Polaridad.es:

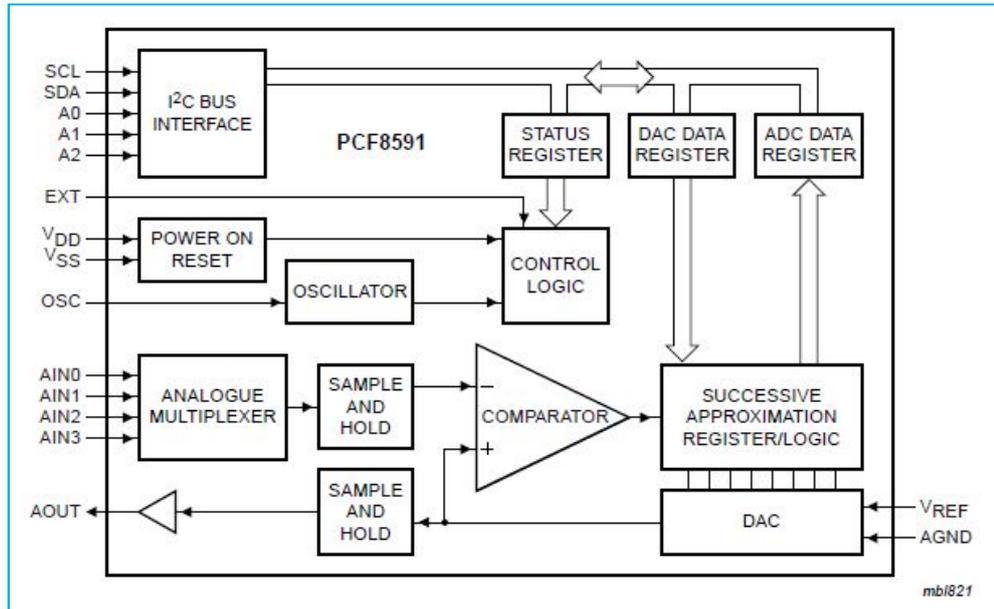
<https://polaridad.es/pcf8591-conversor-ad-da-analogico-digital-8-bits-bus-i2c/>

# ANEXOS

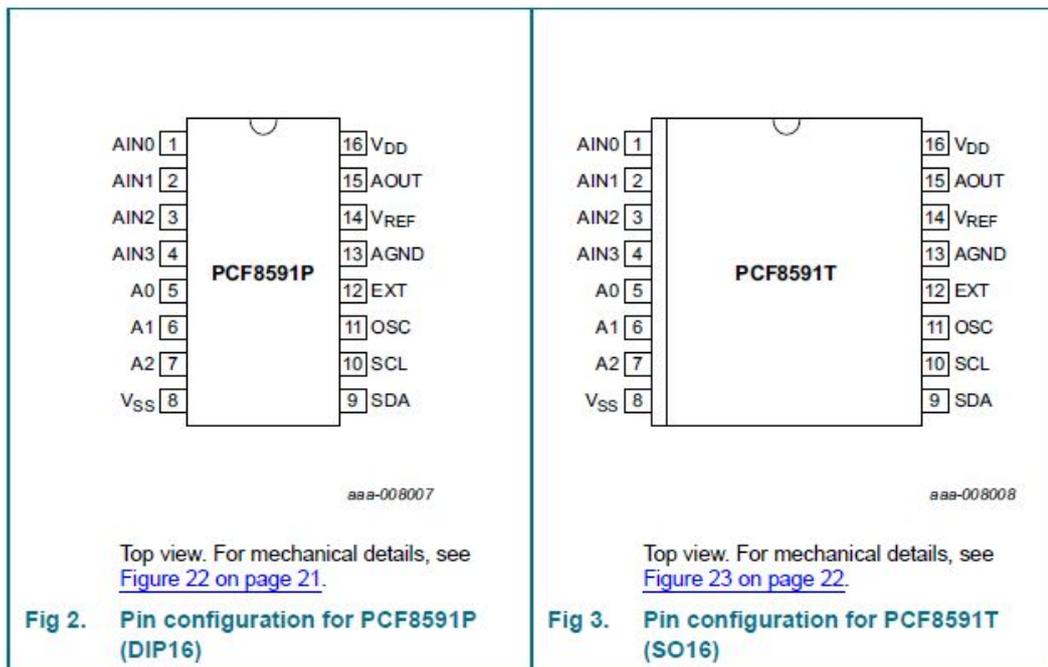
## ÍNDICE DE ANEXOS

**ANEXO A** ESPECIFICACIONES DEL PCF 8591

## DIAGRAMA DE BLOQUES



## INFORMACION DE PINES



## DESCRIPCION DE PINES

Table 4. Pin description

Symbol	Pin	Description
AIN0	1	analog inputs (A/D converter)
AIN1	2	
AIN2	3	
AIN3	4	
A0	5	hardware slave address
A1	6	
A2	7	
V <sub>SS</sub>	8	ground supply voltage
SDA	9	I <sup>2</sup> C-bus serial data input and output
SCL	10	I <sup>2</sup> C-bus serial clock input
OSC	11	oscillator input/output
EXT	12	external/internal switch for oscillator input
AGND	13	analog ground supply
V <sub>REF</sub>	14	voltage reference input
AOUT	15	analog output (D/A converter)
V <sub>DD</sub>	16	supply voltage

## CONTROL BYTE

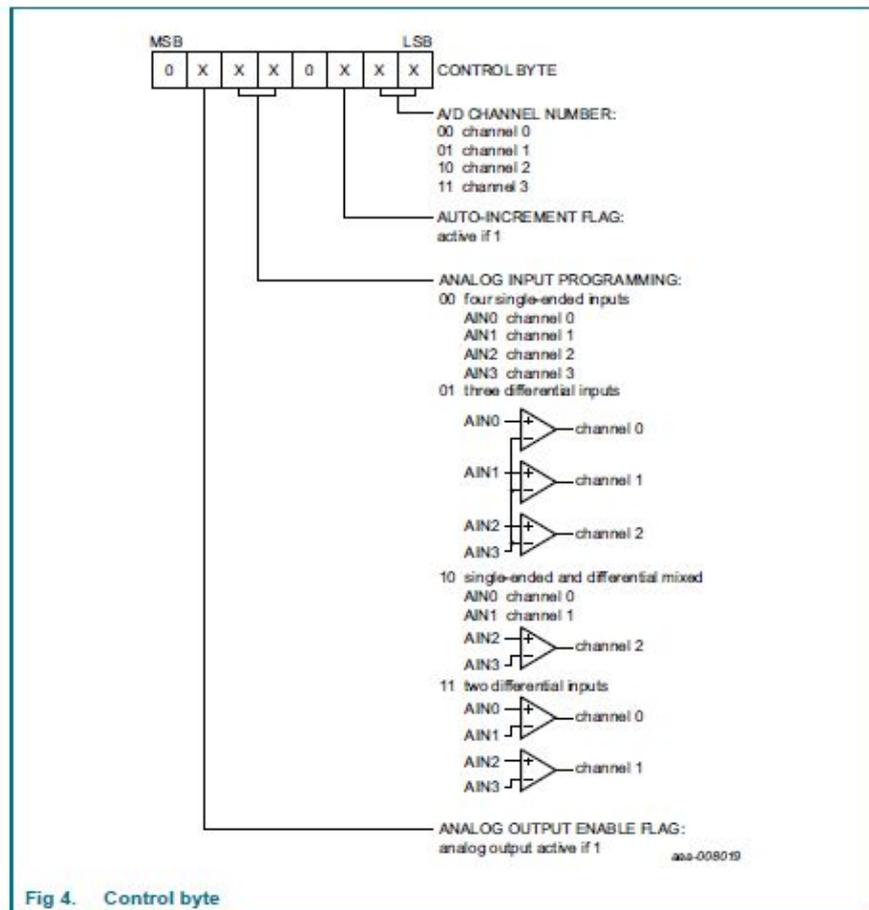


Fig 4. Control byte

## HOJA DE VIDA



### DATOS PERSONALES:

---

**Nombres y apellidos:** José Eduardo Taipe Cando

**Documento de identidad:** 0504012105

**Fecha de nacimiento:** 18 de Agosto de 1993

**Lugar de nacimiento:** Cotopaxi – Latacunga

**Estado civil:** Soltero

**Dirección:** Barrio Sur - Av rossbell

**Teléfono:** 0999040618

**E-mail:** eduarstaipe123@hotmail.es

**FORMACIÓN ACADÉMICA:**

---

<b>Estudios Primarios:</b>	Escuela "Marquesa de Solanda"
<b>Estudios Secundarios:</b>	Academia Militar "Gral." Miguel Iturralde
<b>Universitarios:</b>	Unidad de Gestión de Tecnologías-ESPE Carrera de Electrónica Mención Instrumentación y Aviónica

**CURSOS:**

---

- Curso de mantenimiento de computadoras "SECAP" Año 2011
- Curso de informática "FUDELA" Año 2012

**EXPERIENCIA PROFESIONAL O PRÁCTICAS PREPROFESIONALES:**

---

- Técnico- Pasante. Centro de Mantenimiento de la Aviación del Ejército "CEMAE" de la brigada de Aviación #15  
Año: 2015 160 horas
- Pasante- Ayudante en Mantenimiento. Grupo de Aviación del Ejército "GAE" #15  
Año: 2016 150 horas

---

**CALAPAQUI GONZÁLEZ MARÍA JOSÉ**

**C.C.: 1724746563**

## **ACEPTACIÓN DEL USUARIO**

Latacunga, Julio de 2017

Yo, ING PABLO PILATÁSIG en calidad de encargado del Laboratorio de Instrumentación Virtual de la Unidad de Gestión de Tecnologías, me permito informar lo siguiente:

El proyecto de graduación elaborado por la Srta. CALAPAQUI GONZÁLEZ MARÍA JOSÉ, con el tema: "IMPLEMENTACIÓN DE UN CONTROL DE TEMPERATURA MEDIANTE EL MICROCOMPUTADOR RASPBERRY PI PARA PRÁCTICAS DE MICROCONTROLADORES EN LA UNIDAD DE GESTIÓN DE TECNOLOGÍAS", ha sido efectuado de forma satisfactoria en las dependencias de mi cargo y que la misma cuenta con todas las garantías de funcionamiento, por lo cual extiendo este aval que respalda el trabajo realizado por el mencionado estudiante.

Por tanto me hago cargo de todas las instalaciones realizadas por la Sr. Estudiante.

Atentamente,

**ING. PABLO PILATÁSIG**

**ENCARGADO DEL LABORATORIO DE INSTRUMENTACION VIRTUAL.**

## HOJA DE LEGALIZACIÓN DE FIRMAS

DEL CONTENIDO DE LA PRESENTE INVESTIGACIÓN SE  
RESPONSABILIZA EL AUTOR

---

CALAPAQUI GONZÁLEZ MARÍA JOSÉ

C.C: 17247476563

DIRECTOR DE LA CARRERA DE ELECTRÓNICA MENCIÓN  
INSTRUMENTACIÓN Y AVIÓNICA

---

ING. PABLO PILATASIG

Latacunga, Julio de 2017