Artículo **Revista de Ingeniería Innovativa**
Junio 2020 Vol.4 No.14 1-13

Raspberry Pi, conectividad y programación mediante puertos GPIO

Raspberry Pi, connectivity and programming through GPIO ports

CABALLERO-JULIÁN, Franco Gabriel †*, MORALES-HERNÁNDEZ, Maricela, SILVA-CRUZ, Eric Mario y CABALLERO-CANTARELL, Diego Gabriel

Instituto Tecnológico de Oaxaca, México.

ID 1er Autor: *Franco Gabriel, Caballero-Julián* / **ORC ID:**0000-0002-5924-7759, **CVU CONACYT ID:** 88993

ID 1er Coautor: *Maricela, Morales-Hernández* / **ORC ID:** 0000-0002-3521-2041, **CVU CONACYT ID:** 731036

ID 2do Coautor: *Eric Mario, Silva-Cruz* / **ORC ID:** 0000-0002-0496-9682, **CVU CONACYT ID:** 206891

ID 3er Coautor: *Diego Gabriel, Caballero-Cantarell* / **ORC ID:** 0000-0001-5364-6321

DOI: 10.35429/JOIE.2020.14.4.1.13 Recibido Abril 10, 2020; Aceptado Junio 30, 2020

Resumen

Hoy en día, existe una cantidad innumerable de dispositivos electrónicos y sistemas informáticos que están orientados como plataformas para la gestión de la información a través de la red; manipulando datos y al mismo tiempo, realizando tareas mediante señales electrónicas de baja tensión. Pueden activar, desactivar o controlar un dispositivo o un sistema mediante sensores. En este contexto, la plataforma *Raspberry Pi* tiene una interfaz compuesta por un conjunto de pines (GPIO) a través del cual se puede acceder a las señales de voltaje, y puede interactuar con otros sistemas eléctricos y electrónicos para la adquisición de datos y control. Este artículo tiene como objetivo facilitar el acceso a la plataforma *Raspberry Pi* desde su instalación, configuración, hasta su programación de los puertos a través de ejemplos e instrucciones, considerando la plataforma como un sistema informático. El interés de los autores es proporcionar al lector las herramientas, para que la programación de aplicaciones específicas sea el nuevo desafío.

Raspberry, GPIO, WebIOPi

Abstract

Today, there is an innumerable amount of electronic devices and computer systems that are oriented as platforms for the management of information through the network; manipulating data and at the same time, performing tasks using low voltage electronic signals. They can activate, deactivate or control a device or a system using sensors. In this context, the *Raspberry Pi* platform has an interface composed of a set of pins (GPIO) through which voltage signals can be accessed, and can interact with other electrical and electronic systems for data acquisition and control. This article aims to facilitate access to the *Raspberry Pi* platform from its installation, configuration, to its programming of the ports through examples and instructions, considering the platform as a computer system. The interest of the authors is to provide the reader with the tools, so that the programming of specific applications is the new challenge.

Raspberry, GPIO, WebIOPi

Citación: CABALLERO-JULIÁN, Franco Gabriel, MORALES-HERNÁNDEZ, Maricela, SILVA-CRUZ, Eric Mario y CABALLERO-CANTARELL, Diego Gabriel. *Raspberry Pi*, conectividad y programación mediante puertos GPIO. Revista. Revista de Ingeniería Innovativa. 2020. 4-14:1-13.

* Correspondencia del Autor (franco.caballero@itoaxaca.edu.mx)
† Investigador contribuyendo como primer autor.

Introducción

En sistemas electrónicos digitales con innumerables aplicaciones para la red, de control y automatización, de ahorro y aprovechamiento de la energía solar fotovoltaica (Sesma Martínez, 2015), en tecnología bioelectrónica, en domótica e inmótica, entre otras; las soluciones son diversas. Así disponemos de una amplia gama de fabricantes de microcontroladores, muchos otros de microprocesadores, de *FPGAs*, de sistemas de desarrollo y aplicaciones (Asadi, Baguery & Imam, 2016), y cada uno con un extenso catálogo de productos con características específicas, dispositivos que permiten la interconectividad con el mundo real. En esos dispositivos se dispone de módulos de comunicación paralelo, serial, *USB, I2C, SPI, Ethernet, wifi, bluetooth*, incluso (Amaya,2020) con soluciones tecnológicas *LP-WAN* como *Narrow Band Internet of Things (NB-IoT*), *Sig Fox* y *Long Range (Lo-Ra)*.

En ese contexto existe un sistema de bajo costo para posicionarse en el mercado, la plataforma *Raspberry Pi (*ver figura 1), con el interés primario de hacer que las nuevas tecnologías y los sistemas digitales lleguen a todos los niveles en el ámbito educativo. La *Raspberry Pi Foundation* desarrolló la *Raspberry Pi* como una computadora de placa única u ordenador de placa simple (*SBC*) de bajo costo, basada en un microprocesador embebido. El diseño original es un entorno de desarrollo práctico, que permite un sinfín de posibilidades para aplicaciones novedosas.

Figura 1 La Tarjeta Rasberry Pi
Fuente: Descargada de: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

La plataforma *Raspberry Pi* es una computadora que utiliza un sistema operativo basado en *Linux*, es una solución total equivalente a una computadora convencional con el popular sistema operativo *Windows*. *Raspberry Pi* dispone de sus propias herramientas de uso muy fácil e intuitivo, donde para cada aplicación deseada se instala el sistema operativo correspondiente y sus controladores necesarios. Así disponemos de aplicaciones (Suehle y Callaway, 2014) en servidores, multimedia, con vídeo *Full HD*, videojuegos, domótica e inmótica (Falcone et al.,2015) y monitoreo, telefonía, Internet de las Cosas, procesamiento de imágenes, se le puede instalar *Android*, etc. Y entre tantas virtudes, el área de oportunidad en el que se centra el interés de esta investigación, es el monitoreo de variables y control de dispositivos (Arostegui Gallardo et al., 2019) por medio de sus puertos de entrada/salida de propósito general *(GPIO*), ver figura 2.

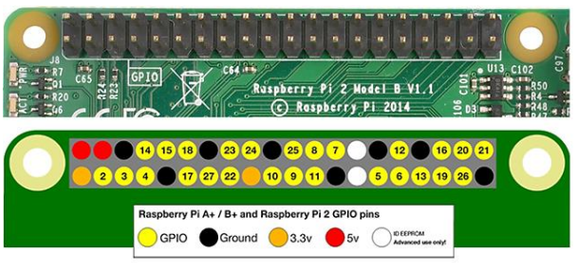

Figura 2 La GPIO de la Tarjeta Rasberry Pi
Fuente:https://www.programoergosum.com/cursos-online/raspberry-pi/238-control-de-gpio-con-python-en-raspberry-pi/que-es-gpio

La arquitectura de *Raspberry Pi* (Ray, 2017), está centrada en un procesador multinúcleo *Broadcom (BCM 2835, BCM2836 y BCM 2837)*, memoria compartida entre la *CPU y* la unidad de procesamiento de gráficos *(GPU)*, puertos *USB, HDMI, RJ45,* 40 pines *GPIO* multiplexados, *miniUSB,* y un conector para cámara, etc. El software *Raspbian* es su sistema operativo oficial, de código abierto y basado en *Linux*, aunque permite usar otros sistemas operativos, incluido *Fedora, ArchLinux Arm, Debian, Ubuntu* y una versión de *Windows 10*.

La fundación promueve principalmente el aprendizaje del Lenguaje de programación *Python*, aunque también están soportados otros lenguajes de programación como son: *Tiny BASIC, C, PERL* y *Ruby*.

ISSN 2523-6873

La fundación da soporte para otras descargas de las distribuciones para arquitectura *ARM*, *Raspbian*, derivada de *Debian; RISC OS 5; Arch Linux ARM*, derivado de *Arch Linux* y *Pidora*, derivado de *Fedora*.

De la diversidad de modelos de *Raspberry Pi* existentes en el mercado, la *Raspberry Pi 3* modelo A+ data del año 2018, cuyo modelo A+ están limitado en sus prestaciones y con un menor costo. Cuenta con 512 MB de memoria *RAM*, compartidos con la *GPU VideoCore IV*, un solo puerto *USB* y sin puerto de conexión de red por cable *RJ-45*.

La *Raspberry Pi 3* B+ también data del año 2018 y entre sus mejoras cuenta con un nuevo procesador y mejor conectividad, opera a 1.4Ghz y ahora incorpora doble banda a 2,4GHz y 5GHz, y su nuevo puerto *Ethernet* pasa a 300 *Mbits/s* en el nuevo modelo, también cuenta con *Bluetooth 4.2 (Low Energy)*.

La *Raspberry Pi 4* modelo B estuvo disponible desde junio del año 2019. Sus puertos *HDMI* de tamaño completo ahora son dos puertos micro *HDMI*. Tiene un procesador *Broadcom* nuevo hasta tres veces por encima que el anterior. Cuenta con la capacidad de manejar dos pantallas 4K a 60 Hz. Incluye por primera vez *USB 3.0*, y el puerto *Ethernet* ya no está limitado a 300 *Mbps*.

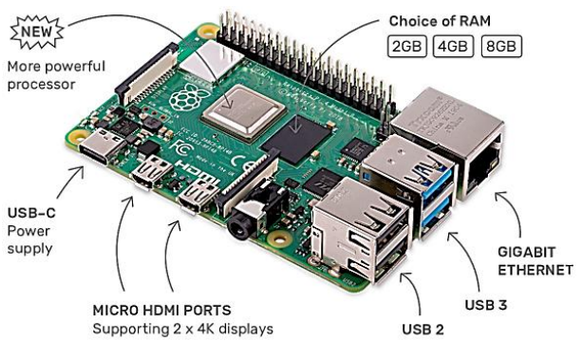

Figura 3 La Tarjeta Rasberry Pi 4 modelo B+
Fuente: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/

En la figura 4, se muestra la localización de los componentes de una placa de *Raspberry Pi* modelo B.

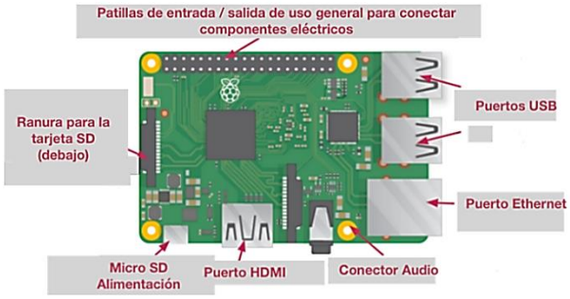

Figura 4 Localización de componentes
Fuente:https://projects.raspberrypi.org/esES/projects/raspberry-pi-getting-started/2

En relación con sistema operativo, *Raspbian* es una distribución del sistema operativo *GNU/Linux* basado en *Debian*, libre para la *SBC Raspberry Pi*, fue desarrollado por el equipo que creó *Raspberry Pi* (Grey,2018). Desde 2015, la *Raspberry Pi Foundation* lo ha proporcionado de forma oficial como el sistema operativo primario para la familia de placas SBC de *Raspberry Pi*.

Hay varias versiones (Ray, 2017) de *Raspbian*, *Raspbian Wheezy* es una imagen basada en la versión *Debian 7.x*, *Raspbian Jessie* basada en la versión *Debian 8.x*, siendo la actual *Raspbian Buster* con la que se hicieron los ejemplos de este artículo.

En cuanto a los recursos para hacer programación, aunque cualquier lenguaje de programación para *Linux* está disponible para *Raspberry Pi*, se puede escribir código desde la línea de comandos (*Shell)*, también en *C y C++*, *Java*, *Ruby*, *Visual Basic*, etc., son más populares los lenguajes interpretados directamente como *Python*, *Bash*, *Perl* y *Lua* (Ray, 2017); el sistema operativo *Raspbian Buster* tiene *Thonny Python IDE* como una herramienta de programación, y dada su versatilidad y preferencia en el gusto de los programadores, elegimos *Python* que es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código.

Es un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.

ISSN 2523-6873

Administrado por la *Python Software Foundation* (Guzdial y Ericson,2013), posee una licencia de código abierto, denominada *Python Software Foundation License*, que es compatible con la Licencia pública general de *GNU* a partir de la versión 2.1.1. *Python* usa tipado dinámico y conteo de referencias para la administración de memoria.

Una característica importante de *Python* es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos).

Planteamiento del problema

El problema al cual nos enfrentamos al utilizar *Raspberri Pi* hasta nuestros días, es la complejidad que representa iniciar con una nueva microcomputadora y un sistema basado en *Linux*, haciendo parecer que sólo los más atrevidos exploran los diversos recursos de *Raspberry Pi*, los que tienen acceso a cursos presenciales bien estructurados, y los que están dispuestos a aprender por su cuenta y a ensayar nuevos experimentos (Magpi,2019). *Raspberry Pi* es fácil, muy entretenido y sobre todo con un sinfín de oportunidades de desarrollo.

El interés se centra en que el lector arme y configure paso a paso una computadora *Raspberry Pi* y se concentre en el manejo de los puertos de propósito general (*GPIO*). Puertos con los que sea posible desarrollar aplicaciones en las que desde esa computadora *Raspberry Pi*, la misma plataforma o algún dispositivo inteligente; conectado en su red *LAN*, tenga acceso para prender, apagar, programar, leer o controlar otros actuadores y sistemas de comunicación (Magpi, s.f).

Propuesta de solución

Para desarrollar aplicaciones *GPIO* con la *Raspberry Pi*, se le presenta al lector un procedimiento en seis pasos que lo guían hasta probar varios ejercicios resueltos por medio de *leds* y *switches* que solo tienen que remplazarse por dispositivos específicos para crear aplicaciones de utilidad.

En el paso 1 se enlista el material necesario para dar comienzo a la instalación, en el paso 2 se describe la preparación de la memoria requerida para la plataforma, en el paso 3 se hacen las conexiones de hardware en donde encontramos una diversidad de elementos, en el paso 4 se describe la configuración inicial de una computadora *Raspberry Pi 3*, para continuar en el paso 5 se le guía en la configuración del *Software* y manipulación de archivos y carpetas, y en el paso 6 se le muestran los ejemplos de programación desde la consola, programación con *Arduino*, desde *Python* y en un ambiente *Web* por medio de *WebIOPi*.

Metodología

De acuerdo con la clasificación expuesta por Bernal (2010), el método apropiado para la realización de este trabajo es el método hipotético-deductivo porque partimos de aseveraciones en calidad de hipótesis que en el desarrollo se van comprobando.

En el presente trabajo se utiliza la metodología en “V” o diagrama en “V” como se le llama por distintos autores. En esta técnica, el ciclo de vida para el desarrollo de un sistema la progresión en el tiempo es iterativa y se presenta de izquierda a derecha, y que este ciclo es iterativo, se realiza tantas veces como sea necesario (debe ser el mínimo) hasta obtener un producto adecuado y acorde con los requisitos del usuario (Oshana y Kraeling, 2013).

Desarrollo

Material necesario

Lo que se requiere para incursionar en el mundo de *Raspberry Pi* y empezar a desarrollar aplicaciones con los puertos es lo siguiente:

- Una computadora *Raspberry Pi* 2, 3 ó su versión más reciente.
- Una tarjeta *micro SD* mínimo de 8G clase 4, considerando de preferencia 16 o 32G de almacenamiento.
- *Mouse.*
- Teclado.

ISSN 2523-6873

- Un monitor.
- Cable *hdmi* y un convertidor *hdmi-VGA* dependiendo el monitor.
- Una fuente de alimentación dedicada con salida micro USB de 5V a 1.5 Amp.
- Un cable de red RJ-45.
- *Internet* para actualizar el sistema operativo, descargar los controladores y recursos, un *protoboard*, resistencias y diodos *led*.

Preparación de la memoria

En su computadora de escritorio o en su *laptop* con acceso a *internet*, vamos al navegador y escribir la siguiente dirección de *Raspberri Pi*: *https://www.raspberrypi.org*, ir a la sección de descargas(*downloads*). Ahí podemos ver *NOOBS* y *Raspbian*, *Raspbian* es el Sistema Operativo oficial para todos los modelos de *Raspberry Pi*. Se puede utilizar *Raspbian Pi imager* para una forma fácil de instalar *Raspbian* y otro sistema operativo en su tarjeta *SD* listo para usarse en su *Raspberry Pi*, viene en formato *exe*.

Es muy sencillo utilizar *Raspbian*, por ser el más popular. La versión *LITE* solo contiene el Sistema Operativo (SO) para conectarse remotamente a la placa. La versión *desktop* nos da el acceso a periféricos, esto debido a que la otra versión solo tiene software recomendado. Aquí se recomienda descargar *Raspbian buster full with desktop and recommended software* en formato *zip* versión septiembre 2019 de 2530 MB.

Desempacar el archive *zip* para tener una nueva carpeta que ahora contiene el archivo imagen*: raspbian-buster-full.img.*, ese es el archivo que vamos a *flashear* en la *SD card*. También podemos considerar elegir *NOOBS*, la cual viene en archivos separados y el proceso a realizar es desempacar la carpeta y tal cual su contenido se copia en la *SD card*, así está lista para insertarse en su *Raspbery Pi* e iniciar el proceso de instalación en la plataforma para realizar el reconocimiento de periféricos y dispositivos conectados a la tarjeta *Raspberry Pi*.

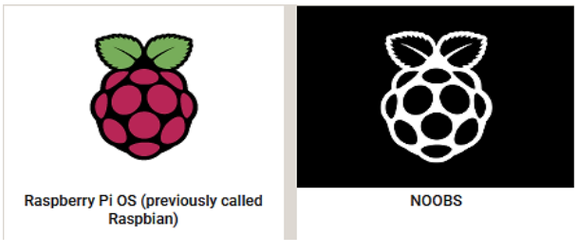

Figura 5 Raspbian y NOOBS, el sistema operativo
Fuente: https://www.raspberrypi.org/downloads/

Para formatear la *SD card* utilizamos la herramienta *SD Card formatter* y para cargar el SO vamos a utilizar *balena etcher* en su versión para *Windows*, este es un programa que nos permite *flashear* memorias.

Funciona en tres pasos:

- En seleccionar imagen, seleccionamos el archivo *raspbian-buster-full.img*
- Ahora permite hacer el segundo paso, deja seleccionar en donde se va a hacer la instalación, la opción *change* da la lista de dispositivos donde se puede instalar, seleccionamos la memoria *SD*.
- Seleccionar la opción *flash* para iniciar el proceso de carga de los archivos.

Otra opcion es *Win32DiskImager*, como se muestra en la figura 6.

Figura 6 SD Formatter, Win32DiskImager y balenaEtcher, las herramientas
Fuente: Elaboración propia

Conexiones de hardware

Con su *micro SD* en la mano, insertarla en la ranura de la plataforma *Raspberry Pi*, y previamente se deber conectar el teclado, *mouse*, el monitor, el cable RJ-45 a la red de internet (si dispone de la *Raspberry Pi 3 Model B* ya tiene integrado el módulo *wifi*) y finalmente conectar la fuente de alimentación para inicializar la plataforma. Otro medio de conexión a una red local es mediante una computadora por medio de un puente con un cable *RJ-45* con una conexión *Ethernet.*

ISSN 2523-6873

Al instante su computador *Raspberry Pi* va a iniciar, y podemos monitorear este proceso mediante dos leds de actividad, el rojo indica energizado y el verde parpadeando indica el arranque de la computadora. Una vez inicializada la carga de los controladores, la tarjeta tendra acceso al video y se visualizará la carga de controladores.

Figura 7 Conectando los componentes asociados a la computadora Raspberry Pi
Fuente: Elaboración propia

Configuración inicial

Al inicio el sistema le da la bienvenida con un *Welcome to the Raspberry Pi Desktop* con la imagen de la frambuesa, y le indica que hay algunas cosas que configurar antes de poder utilizarla, ahí mismo le proporciona una dirección *ip* que toma de su red, por ejemplo *192.168.1.66*, y espera que le diga *Next*, enseguida elegir su país (*México*), lenguaje (*Mexican Spanish*), la zona horaria(*México City*) y se recomienda dejar desmarcados el lenguaje y el teclado *US*; damos *Next* y la computadora pide esperar para hacer esa configuración, enseguida pide proporcionar y confirmar un *password*, (podemos utilizar los valores por default), muestra las redes *wifi* que tiene a su alcance solicitando la clave de acceso a la Red Local.

Enseguida requiere actualizar el *software*, (este proceso se puede omitir, es mejor hacerlo de una vez); después de varios minutos, el sistema avisa que ha terminado y que para que las actualizaciones tengan efecto hay que restablecer en ese instante (*Restart*), lo cual es lo recomendable.

Al reiniciar la tarjeta y restablecer el sistema, se muestra la pantalla con todos los recursos de inicio que son: *Programación, Educación, Oficina, Internet, Sonido y Vídeo, Gráficos, Juegos, Accesorios, Help, Preferencias, Run y Shutdown*. También en la barra de tareas, en la parte superior izquierda podemos encontrar los íconos de *Internet*, carpetas y la consola (*Shell*).

El entorno de manejo de archivos es parecido al basado en Ventanas (Windows), con *click* en el ícono de *Internet* tenemos acceso a todos los recursos disponibles en la red, podemos crear carpetas y archivos, por medio del ratón podemos consultar o eliminar carpetas y por medio de la consola podemos ejecutar una serie de comandos, en donde la instrucción **s**udo, nos permite realizar operaciones como administrador.

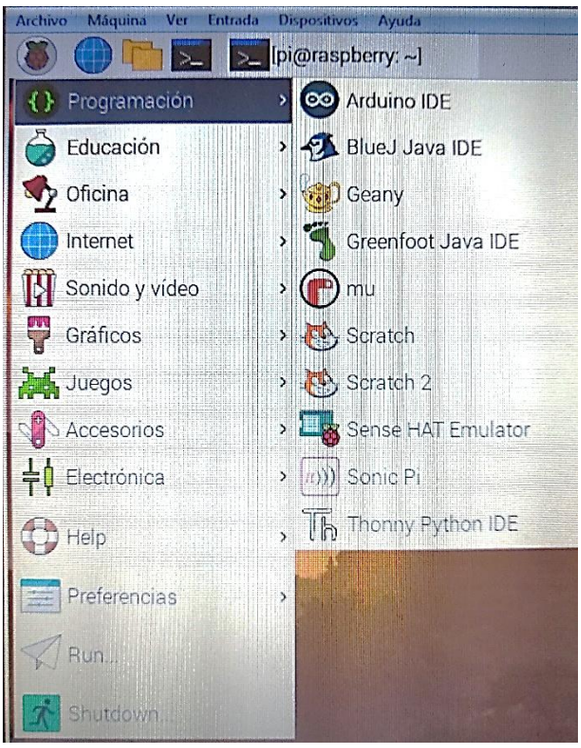

Figura 8 La computadora Raspberry Pi, al arranque
Fuente: Elaboración propia

Configuración del Software y manipulación de archivos y carpetas

Aunque ya en el paso anterior el sistema se actualizó, por medio de la consola podemos en cualquier momento actualizar y ejecutar instrucciones para manipular archivos y carpetas.

ISSN 2523-6873

Para actualizar la fecha:

$***sudo date*** *04291630*

donde: 04 es el mes, 29 es el día, 16 es la hora y 30 son los minutos.

Para actualizar el sistema:

$***sudo apt-get update && sudo apt-get upgrade***

Un proceso de interés particular en esta investigación es el desarrollo de aplicaciones mediante la instalación de *Arduino*, para lo cual se realiza la instalación del *IDE* de *Arduino* en la *Raspberry Pi*, por medio del comando:

$***sudo apt-get install arduino***

Para mostrar el directorio:

$***pwd***

Para listar el contenido de una carpeta, escribir:

$***ls***

Para moverme a la carpeta raíz, escribir:

$ **cd ..**

Para crear una carpeta escribir:

$***mkdir nombredecarpeta***

Para crear un archico escribir:

$***touch nombredearchivo***

Para remover escribimos:

$***rm nombredearchivo***

Para utilizar los comandos utilizados recientemente escribimos:

$***history***

Programación

Instrucciones desde la terminal

Desde la consola, es posible controlar las terminales por medio de instrucciones que declaran la *GPIO* de entrada o salida (Ray,2017); en el caso de salida el valor digital que asume es de 5V o 3.3V dependiendo la tecnología.

En las siguientes líneas se declara la *GPIO26* de uso, configurada como salida y se le envía un valor uno y después un cero lógico.

$ ***echo 26 > /sys/ class/gpio/export***

$ ***echo out > /sys/class/ gpio/gpio26/direction***

$***echo 1 > /sys/class/gpio/gpio26/value***

$***echo 0 >/sys/class/gpio/gpio26/value***

Programación en *Python*

En *Thonny Python IDE*, un recurso precargado en la *Raspberry Pi*, se escriben los programas de aplicación (Mocq, 2020). Aquí se exponen tres ejemplos probados con una entrada y cuatro salidas. Estos ejemplos son muy básicos (Programo Ergo Sum, 2020), a partir de aquí que el lector pueda utilizar los recursos más potentes de programación para hacer programas más complejos que se puedan utilizar en sistemas de aplicación. Se eligen pines de entrada o salida.

Entradas:

INP0: GPIO23, PIN12

Salidas:

D0:GPIO22, PIN 15

D1:GPIO27, PIN 13

D2:GPIO17, PIN 11

D3:GPIO4, PIN 7

El cable de tierra es por medio del PIN 6 de GND.

ISSN 2523-6873

La entrada se implementa con un *dip-switch* para generar un 1 o un 0, este circuito se alimenta de los 5V de VCC de la *Raspberry Pi* a través de una resistencia de 15K; cuando el *switch* está en ON, la muestra que se deriva de la resistencia está a un potencial de referencia y manda un 0, cuando se abre el *switch* la resistencia queda deshabilitada reflejando un 1.

Las salidas se presentan como D0, D1, D2, y D3; se implementan por medio de leds con limitadores de corriente de 1K. El circuito puede verse en la figura 9.

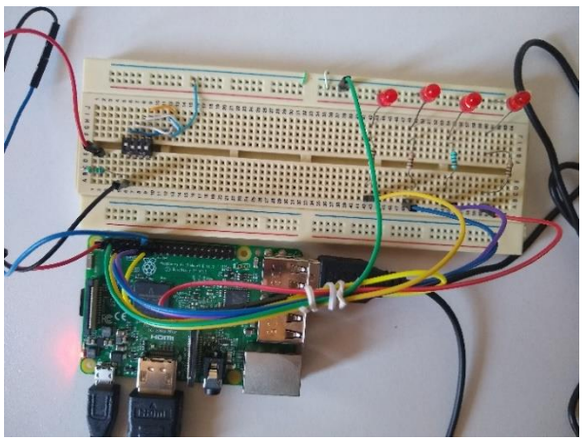

Figura 9 El circuito de prueba de los ejemplos
Fuente: Elaboración propia.

El ejemplo 1 es un ejercicio muy simple en el que se implementa una secuencia de prender y apagar un grupo de 4 leds por medio de la librería *gpiozero*.

El ejemplo 2 por medio de la estructura *while* implementa una secuencia de 5 repeticiones para prender y apagar un grupo de 4 leds por medio de la librería *GPIO*.

El ejemplo 3 utiliza las estructuras de control *if* y *while* para implementar una rutina de prender y apagar dos leds por medio de la librería GPIO. Sí GPIO23 es 1 entonces la rutina aplica en D0, D1 y sí GPIO23 es 0 la rutina aplica en D3, D4.

Ejemplo1.py

```
#####################################

from gpiozero import LED

from time import sleep

led0=LED(22)

led1=LED(27)

led2=LED(17)

led3=LED(4)

while True:

led0.on()

sleep(1)

led0.off()

sleep(1)

led1.on()

sleep(1)

led1.off()

sleep(1)

led2.on()

sleep(1)

led2.off()

sleep(1)

led3.on()

 sleep(1)

led3.off()

sleep(1)
```

```
#####################################
```

Ejemplo2.py

```
#####################################

import RPi.GPIO as GPIO

import time

LEDPin0= 22

LEDPin1= 27

LEDPin2= 17
```

ISSN 2523-6873

```
LEDPin3= 4
Inicio=0
Fin=5
GPIO.setmode (GPIO.BCM)
GPIO.setup(LEDPin0, GPIO.OUT)
GPIO.setup(LEDPin1, GPIO.OUT)
GPIO.setup(LEDPin2, GPIO.OUT)
GPIO.setup(LEDPin3, GPIO.OUT)
while (Inicio<= Fin):
GPIO.output(LEDPin0, GPIO.HIGH)
time.sleep(0.5)
GPIO.output(LEDPin0, GPIO.LOW)
time.sleep(0.5)
GPIO.output(LEDPin1,GPIO.HIGH)
time.sleep(0.5)
GPIO.output(LEDPin1,GPIO.LOW)
time.sleep(0.5)
GPIO.output(LEDPin2,GPIO.HIGH)
time.sleep(0.5)
GPIO.output(LEDPin2,GPIO.LOW)
time.sleep(0.5)
GPIO.output(LEDPin3,GPIO.HIGH)
time.sleep(0.5)
GPIO.output(LEDPin3,GPIO.LOW)
time.sleep(0.5)
time.sleep(0.5)
print("Numero de rutina:", Inicio)
Inicio = Inicio + 1
GPIO.cleanup()
```

```
####################################
```

Ejemplo3.py

```
########################################
import RPi.GPIO as GPIO
import time
LEDPin0= 22
LEDPin1= 27
LEDPin2= 17
LEDPin3= 4
Inicio=0
Fin=5
GPIO.setmode (GPIO.BCM)
GPIO.setup(LEDPin0, GPIO.OUT)
GPIO.setup(LEDPin1, GPIO.OUT)
GPIO.setup(LEDPin2, GPIO.OUT)
GPIO.setup(LEDPin3, GPIO.OUT)
GPIO.setup(23,GPIO.IN)
while (Inicio<= Fin):
input_state=GPIO.input(23)
if input_state == True:
GPIO.output(LEDPin0,GPIO.HIGH)
time.sleep(0.5)
GPIO.output(LEDPin0,GPIO.LOW)
time.sleep(0.5)
GPIO.output(LEDPin1,GPIO.HIGH)
time.sleep(0.5)
GPIO.output(LEDPin1,GPIO.LOW)
```

ISSN 2523-6873

time.sleep(0.5)

print("Numero de rutina:", Inicio)

Inicio = Inicio + 1

else:

GPIO.output(LEDPin2,GPIO.HIGH)

time.sleep(0.5)

GPIO.output(LEDPin2,GPIO.LOW)

time.sleep(0.5)

GPIO.output(LEDPin3,GPIO.HIGH)

time.sleep(0.5)

GPIO.output(LEDPin3,GPIO.LOW)

time.sleep(0.5)

time.sleep(0.5)

print("Numero de rutina:", Inicio)

Inicio = Inicio + 1

GPIO.cleanup()

##

Programación de Arduino

Arduino es muy versátil, y presenta una variedad de modelos, con distintos recursos, así ha ganado el interés de muchos desarrolladores en todo el mundo; también ha logrado que se posicione en el mercado por ser de bajo costo y muy fácil de usar (Rumberg, 2015). Entre muchas de sus posibilidades, una placa *Arduino* se le puede dejar en modo esclavo para transmitir o recibir señales de voltaje mediante comunicación serial y realizar interacción con entornos de programación como son: *Matlab*, *LabVIEW*, *Processing*, también mediante el puerto *USB* Ethernet de la computadora *Raspberry Pi*.

Instalación de *Arduino*

$***sudo apt-get install arduino***

La conectividad de *Arduino* con los puertos disponibles se puede visualizar mediante la siguiente instrucción.

$***ls/ dev/ tty****

De este modo al conectar una tarjeta *Arduino* UNO en el puerto USB de la *Raspberry Pi*, es posible realizar la programación de cualquier programa al cargarlo y ejecutarlo en la tarjeta *Arduino*.

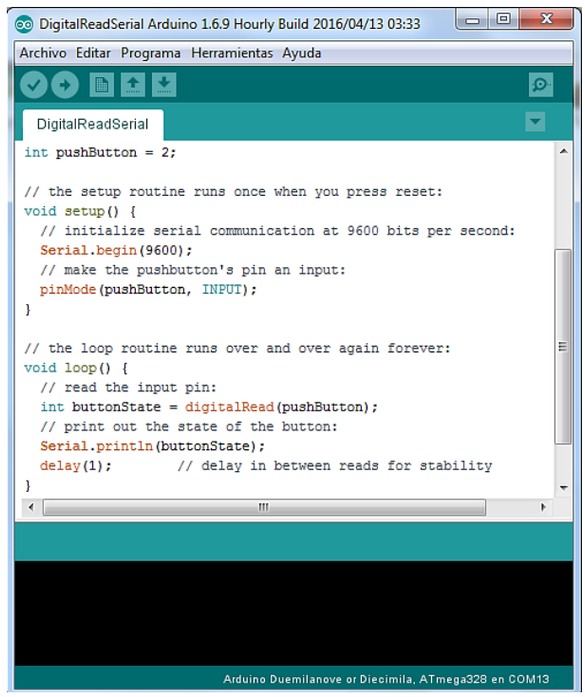

Figura 10 El entorno de desarrollo Arduino
Fuente: Elaboración propia

Por medio de una página Web

Entre los recursos desarrollados por terceros está *WebIOPi* que es de fácil instalación y muy intuitivo. De antemano, nuestra computadora *Raspberry Pi* ya tiene acceso a *Internet*, entonces abrir el navegador y en *The Raspberry Internet of Things Toolkit* (2020), vamos al apartado de descargas y hacer la descarga que por defecto se aloja en la carpeta *Downloads*. O desde la terminal de comandos hacemos también la descarga de la última versión de la aplicación.

$wget

http://webiopi.googlecode.com/files/WebIOPi-0.7.0.tar.gz

$tar xvzf WebIOPi-0.7.1.tar.gz
$cd WebIOPi-0.7.1

Descargamos el parche:

$**wget** https://raw.githubusercontent.com/doublebind/raspi/master/webiopi-pi2bplus.patch

Y lo aplicamos

$**patch -p1 –i webiopi-pi2bplus.parch**

Instalamos

$**sudo ./setup.sh**

Configurar

$**sudo webiopi –d –c/etc/webiopi/config**

Y arrancar la aplicación:

$**sudo /etc/init.d/webiopi start**

Para detenerla

$**sudo /etc/init.d/webiopi stop**

Una vez inicializada la aplicación, ahora simplemente en el navegador de la *Raspberry pi*, escribimos la siguiente dirección: http://raspberrypi:8000/ ó escribiendo en lugar de *raspberry* la dirección *ip* de su computadora que puede obtenerse por medio de la red usando *nmap*. De igual modo puede acceder a su página web desde cualquier otro dispositivo vinculado a su red local. El nombre de usuario por defecto es *webiopi* y el *password* es *raspberry*. En el ejercicio que aquí se presenta por medio de la red local desde un teléfono inteligente, o desde cualquier computadora en la misma red se puede prender el led amarillo de la figura 12, por medio de la *GPIO4*. La *GPIO* en color negro indica apagado, al darle *click* cambia a naranja y hace que se prenda el led. El mismo ejercicio se probó con el circuito de la figura 9 con los leds colocados en las *GPIO 22,27,17 y 4*. En la figura 11 se aprecia el menú de *WebIOPi* al cargar, y en la figura 12 se muestra la distribución de la *GPIO Header* para manipulación.

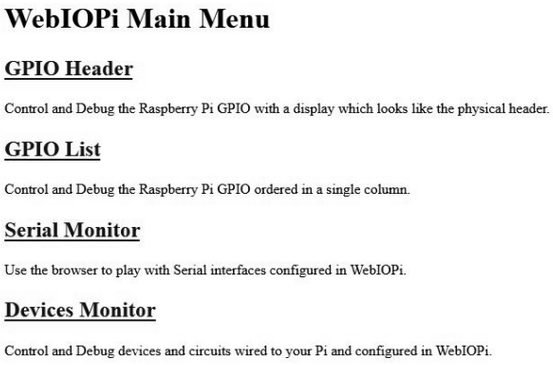

Figura 11 El menú principal de WebIOPi
Fuente: Elaboración propia

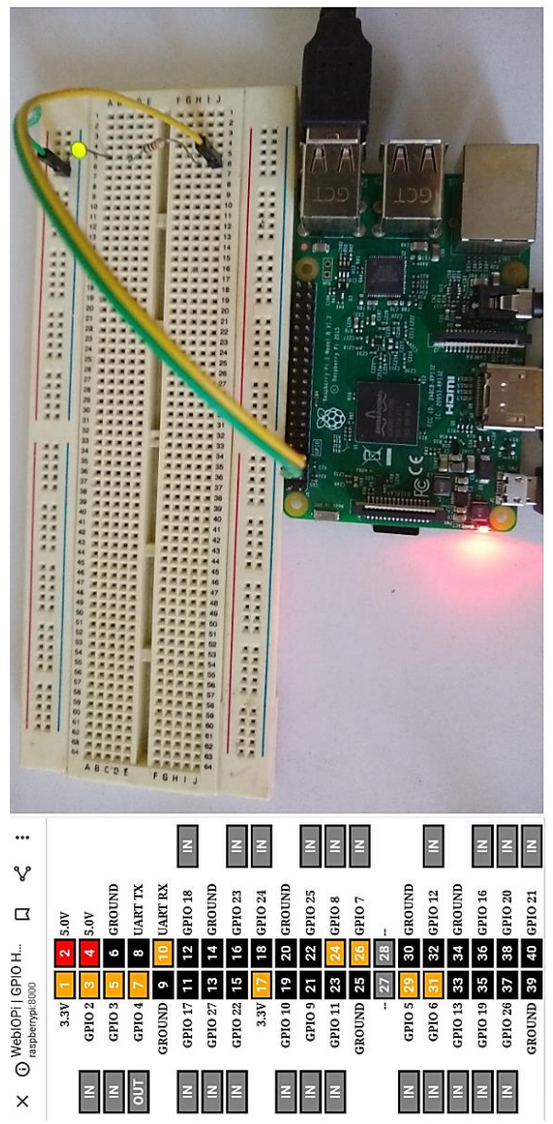

Figura 12 Activar o desactivar una carga en GPIO4 desde WebIOPi
Fuente: Elaboración propia

Resultados

Finalmente, en la presente investigación sobre el proceso de interacción con los puertos *GPIO* de la plataforma *Raspberry Pi*, presentamos los elementos necesarios que pueden servir como una guía que nos lleva de la mano paso a paso en nuestro objetivo:

- Elegir los elementos necesarios para montar una computadora *Raspberry Pi*, *flashear* su micro SD con el sistema operativo *Raspbian*, configurarla de inicio, actualizarla y cargarle los programas de interés.

- Desde ahí, ya disponemos de una computadora muy completa con acceso a internet, con opciones de programación, con *office*, juegos y muchos otros recursos ya incluidos.

- De manera muy directa, se muestran cuatro formas de acceder al accionamiento de los *GPIO*, podemos elegir cualquiera y mediante lenguaje *Python* se puedan crear aplicaciones de interés, o también por medio de la plataforma *Arduino*, y así mismo esta opción que considera la obtención de información y con otra aplicación sea posible mostrarla en pantalla y darle el tratamiento adecuado, como un trabajo a futuro.

- Por medio de *WebIOPi*, disponemos de una herramienta *Web* muy práctica para el acceso de lectura o escritura de los puertos *GPIO*.

- Para construir interfaces que nos permitan crear soluciones reales de proyectos más complejos, incluso en el modo *headless* se puedan manipular vía remota usando *SSH* y *VNC Viewer*.

Agradecimiento

Este trabajo es posible gracias a las facilidades otorgadas por el Tecnológico Nacional de México/Instituto Tecnológico de Oaxaca que a través de distintos medios buscan mejorar el desempeño de sus profesores, también se agradece a PRODEP que por medio de sus convocatorias también encuentra la forma de financiar y otorgar estímulos a los docentes para mejorar su desempeño académico-científico.

Gracias a los profesores y estudiantes que involucrados en tareas de investigación permiten estos logros. Gracias a los colaboradores en este artículo.

Conclusiones

En este artículo muy concreto y simple tenemos una guía de instalación, configuración y programación de una computadora *Raspberry Pi*, mediante la cual los autores pretenden contribuir con su experiencia a los interesados en desarrollador y crear sistemas y aplicaciones mediante este sistema de cómputo.

Los ejemplos presentados tienen un grado de complejidad simple, lo cual deja oportunidad de explorar los innumerables recursos de los que dispone *Raspberry Pi*, para así realizar interfases de lectura y escritura de señales digitales y analógicas, todo depende del entusiasmo y magnitud del problema que se presente.

El reto para el futuro desarrollador consiste en encontrar el problema y proponer una solución con la computadora *Raspberry Pi* mediante el manejo de los puertos GPIO.

Referencias

Amaya, A. (2020). Desarrollo de un Sistema de Comunicación LP-WAN para transmitir señales de geolocalización de personas dentro de entornos naturales. Tesis Profesional. Instituto Tecnológico de Oaxaca-CICESE.

Arostegui Gallardo, C.I., Mata Cruz. F.A.C., & Romero Rodríguez, R.O. (2019). Accesos controlados al cuarto de racks transaccionales financieros aplicando tecnología de corto alcance. Reporte Técnico. ESIME, IPN.

Asadi, A., Bagheri, S., Imam, A. (2016). A data acquisition system based on Raspberry Pi. Design, construction and evaluation. IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON).

Bernal, C., (2010). Metodología de la Investigación. Tercera edición. Pearson-Prentice Hall.

ISSN 2523-6873

Falcone, F., Matías, I., Miltino, J., Gabilondo, A., (2015). Domótica e Inmótica. Instalaciones de telecomunicaciones para edificios. Alfaomega-marcombo.

Grey, G. (2018). RASPBERRY PI, Guía paso a paso para principiantes de Raspberry Pi. Gabriel Grey.

Guzdial, M., Ericson, B., (2013). Introducción a la computación y programación con Python. PEARSON.

Magpi (s.f). Beginner's book. The official Raspberry Pi magazine.

Magpi (2019). The 50 best tips. Issue 80. April 2019. The official Raspberry Pi magazine.

Mocq, F., (2020). Raspberry Pi 3 o Pi Zero. Explote todo el potencial de su nano-ordenador. https://www.ediciones-eni.com/open/mediabook.aspx?idR=0d1aa94cd5698d4846e4519ae4030923

Oshana, R. y Kraeling, M. (2013). Software Engineering for Embedded Systems: Methods, Practical Techniques, and Applications. USA: Elsevier, Inc.

Programo Ergo Sum(2020). Control de Raspberry pi con Python en Raspberry Pi. https://www.programoergosum.com/cursos online/raspberry-pi/238-control-de-gpio-con python-en-raspberry-pi/ que-es-gpioRay, R. (2017). Raspberry PI, Guía paso a paso para dominar Hardware y Software Raspberry PI 3. Ranny Ray.

Runberg, D.(2015). The Sparkfun guide to processing. No Starch Press, Inc.

Sesma Martínez, J. (2015). Monitorización y detección de fallos en una instalación solar fotovoltaica mediante sistema remoto. Trabajo fin de Máster. Escuela Politécnica Superior de Orihuela.

Suehle, R., Callaway, T. (2014). Raspberry Pi Hacks. Tips & Tools for Making Things with the Inexpensive Linux Computer. O'REILLY. USA.

The Raspberry Internet of Things Toolkit (2020). The Raspberry Internet of Things Toolkit- Now in two flavors: Cayenne-the Spicy one, WebIOPi- the Original one. Recuperado de *http://webiopi.trouch.com/*.

ISSN 2523-6873