

1 Introducción

La calculadora básica creada en python y manejada con dip switch para así lograr el ingreso de un sistema binario, todo esto será controlado desde el pequeño ordenador llamado Raspberry Pi el cual será el encargado de alimentar con energía nuestros componentes ubicados en la protoboard, así se puede lograr las 4 operaciones básicas pero en un sistema binario.

La Raspberry Pi en concepto es el de un ordenador desnudo de todos los accesorios que se pueden eliminar sin que afecte al funcionamiento básico. Está formada por una placa que soporta varios componentes necesarios en un ordenador común y es capaz de comportarse como tal, 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 .

2 Índice

1. Introduccion
2. Indice
3. Planteamiento del problema
4. Objetivos
5. Estado del arte
6. Marco teórico
7. Diagramas
8. Lista de componentes
9. Mapa de variables
10. Explicación del código fuente
11. Descripción de prerequisites y configuraciones
12. Conclusiones
13. Recomendaciones
14. Anexos
15. Manual de usuario
16. Hojas técnicas

3 Planteamiento del problema

Se desconoce el control, funcionamiento, implementación de un sistema de Hardware de una calculadora que funciona con dip switch para realizar el ingreso de las operaciones y de los números los cuales son binarios para lo cual se han planteado las siguientes preguntas:

1. ¿Como funciona un DIP SWITCH conectado a los GPIO de la Raspberry Pi?
2. ¿Como se controla el ingreso de datos de los numeros binarios a la interfaz de python?
3. ¿Como cambiar los numeros binarios a reales en la interfaz de python?

4 Objetivos

4.1 Objetivo General

Determinar la relación entre la utilización de una interfaz en python que permita reconocer los DIP SWITCH conectados a los GPIO de la raspberry para la creacion de una calculadora, de manera que se pueda ingresar los numeros de forma binaria y el programa lo reconosca como numero real.

4.2 Objetivos Especificos

1. Descubrir la relación entre la raspberry, lcd, dip switch y los comandos que efectuados como respuesta en los restaurantes que las iniciaron.
2. Comparar los grados de satisfacción antes y después de los cambios realizados.
3. Definir la relación real entre encuestas y satisfacción del cliente.
4. Lograr implementar una calculadora con interfaz en python que permita reconocer los DIP SWITCH conectados a los GPIO como numeros binarios y convertirlos en numeros reales para conseguir que las operaciones que se han ingresado se ejecuten de manera correcta
5. Lograr la interaccion entre la LCD 16X2 y la Raspberry PI en la interfaz de python.
6. Comprender la utlizacion de los DIP SWITCH para el ingreso de numeros binarios.
7. Realizar la programaciòn correcta para que el programa reconozca los GPIO.

5 Estado del arte

Teckchandani, Y., Perumal, GS, Mujumdar, R., y Lokanathan, S. (2016). Sistema de visualización de avisos inalámbricos en pantalla grande. En 2015, Conferencia Internacional IEEE sobre Inteligencia Computacional e Investigación en Computación, ICCIC 2015 . Instituto de Ingenieros Eléctricos y Electrónicos Inc.

Los tableros de anuncios electrónicos inalámbricos son una alternativa más rápida a los tableros de anuncios de tipo pin-up convencionales. Una limitación importante de los métodos utilizados hasta ahora es el pequeño tamaño de las pantallas de cristal líquido (LCD) de 16 2 utilizadas para mostrar los avisos. Este documento propone un método en el que se pueden usar pantallas grandes como monitores de computadora o televisores para mostrar avisos enviados como mensajes de texto desde un teléfono móvil. El método propuesto utiliza el lenguaje de marcado de hipertexto (HTML) para presentar el resultado, ya que ofrece muchas opciones de personalización. Para obtener resultados de alta resolución, se ha utilizado la computadora Raspberry pi del tamaño de una tarjeta de crédito. Esta salida HTML puede mostrarse mediante un navegador web que se ejecute en la Raspberry pi. El aviso que se mostrará se envía como un servicio de mensajes cortos (SMS), que recibe un módem del Sistema global para comunicaciones móviles (GSM). Haciendo la recepción del mensaje inalámbrico. El módem GSM es interrogado a intervalos regulares por la Raspberry pi para mostrar los últimos mensajes.

Azni, MN, Vellasami, L., Zianal, AH, Mohammed, FA, Daud, NNM, Ve-jasegaran, R.,... Kan, PLE (2017). Sistema domótico con aplicación de android. En 2016 3ª Conferencia internacional sobre diseño electrónico, ICED 2016 (págs. 299–303). Instituto de Ingenieros Eléctricos y Electrónicos Inc.

En este sistema, el control y la supervisión de los dispositivos pueden realizarse de dos maneras. El primer método es a través de un servidor web, mientras que el segundo método es mediante un teléfono inteligente basado en la aplicación de Android. Esta interfaz web tiene un marco conocido como Restful Api y funciona para controlar la Raspberry Pi gpio mediante una solicitud http. Además de eso, las aplicaciones de Android explotarán los servicios proporcionados por Restful Api para controlar gpio de Raspberry Pi. Ambos métodos utilizan Restful Api, pero está incluido en la interfaz web, mientras que para Android necesitamos agregar nuestras funcionalidades adecuadas. Se utiliza una PC de escritorio para ejecutar el software del servidor. Raspberry Pi se utiliza como el controlador de la placa para conectar los dispositivos a través del puerto de entrada y salida. La comunicación entre el teléfono inteligente y la placa Raspberry Pi es inalámbrica.

Hoyo, Á., Guzmán, JL, Moreno, JC, y Berenguel, M. (2015). Enseñar conceptos de ingeniería de control utilizando herramientas de código abierto en una placa Raspberry Pi. IFAC-PapersOnLine , 48 (29), 99-104.

Las herramientas disponibles permiten mezclar temas relacionados con la

programación, las comunicaciones, los sistemas operativos y la teoría de control. La conocida placa Raspberry Pi se utiliza como plataforma para explotar los diferentes conceptos propuestos. Las bibliotecas SciPy, Matplotlib y NumPy, que son bibliotecas de código abierto basadas en Python para computación científica y representación gráfica, se utilizan para realizar simulaciones similares a Matlab y para implementar loops de control clásicos. Por otro lado, los procesos virtuales desarrollados en Easy Java Simulations están adaptados para ser controlados a través de la red desde un controlador implementado en la Raspberry Pi con Python. Esta opción es muy útil desde el punto de vista de la enseñanza, ya que los enfoques de control basados en el tiempo, en la red o en el evento se pueden introducir fácilmente en esta arquitectura propuesta. Además, una vez que los alumnos sepan cómo implementar bucles de control en la Raspberry Pi utilizando Python, los procesos reales externos se pueden controlar fácilmente mediante la interfaz GPIO disponible en esta placa electrónica. Luego, un proyecto basado en estas herramientas e ideas se motiva y presenta en este documento para controlar un proceso de nivel de dos tanques.

Ersin, Ç., Gürbüz, R., y Yakut, AK (2016). Aplicación de un microcontrolador Arduino basado en un sistema automático de riego de plantas que utilizan energía solar. *Fenómenos del estado sólido*, 251, 237–241.

En este sistema, el nivel óptimo de necesidad de agua de las plantas está controlado por el Microcontrolador Arduino Uno. Con la ayuda de los botones en los Escudos Lcd, se eligen las especies de plantas y, una vez que se ingresa el nivel de humedad, es controlado por el Microcontrolador Arduino Uno. Cuando el nivel de humedad de la tierra disminuye, el Microcontrolador Arduino Uno envía datos para el riego de la planta y se toma una foto con Raspberry Pi antes del riego. El correo electrónico que contiene el dato y las fotos necesarios se envía a la persona relacionada. Cuando el riego de las plantas alcanza el nivel de humedad elegido, el Microcontrolador Arduino Uno detiene el riego y Raspberry Pi envía las fotos de la planta como un correo electrónico a la persona relacionada nuevamente después del riego. La energía de este sistema es proporcionada por la carga de la batería.

6 Marco teórico

6.1 Conector GPIO

GPIO (General Purpose Input/Output, Entrada/Salida de Propósito General) es un pin genérico en un chip, cuyo comportamiento (incluyendo si es un pin de entrada o salida) se puede controlar (programar) por el usuario en tiempo de ejecución. Los pines GPIO no tienen ningún propósito especial definido, y no se utilizan de forma predeterminada. La idea es que a veces, para el diseño de un sistema completo que utiliza el chip podría ser útil contar con un puñado de líneas digitales de control adicionales, y tenerlas a disposición ahorra el tiempo de tener que organizar circuitos adicionales para proporcionarlos. Los

conectores GPIO presentes en la RaspberryPi nos permiten conectar dispositivos electrónicos y controlarlos desde la RaspberryPi. Estos conectores son conexiones digitales que se pueden usar como entradas (input) o salidas (output) y que funcionan a 3,3 V. Los periféricos GPIO varían muy ampliamente. En algunos casos son, simplemente, un grupo de pines que se puede cambiar en grupo ya sea de entrada o de salida. En otros, cada pin puede configurarse con flexibilidad para aceptar o entregar diferentes voltajes lógicos, y con la potencia de accionamiento configurables y tire ups / bajadas. Las tensiones de entrada y salida son típicamente la tensión de alimentación del dispositivo que contiene los GPIO, y pueden ser dañados por mayores tensiones. Algunos GPIO tienen entradas tolerantes de 5 V: incluso cuando el dispositivo tiene una tensión de alimentación baja (por ejemplo, 2 V), el dispositivo puede aceptar 5 V sin daño. Los puertos GPIO al igual que la RaspberryPi han evolucionado, en los primeros modelos disponíamos de 26 conectores que posteriormente pasaron a 40. Cada uno de estos conectores se identifica mediante un número, cada uno de ellos tiene asignada una función predeterminada como por ejemplo los puertos TXD y RXD que se utilizan para conectar la RaspberryPi a otro PC mediante un cable de consola y poder acceder de forma remota. Es importante saber el modelo de RaspberryPi que tenemos ya que algunos puertos tienen funciones diferentes y es muy importante no equivocarse ya que hay puertos con 3,3 V otros con 5 V a veces un error podría acabar con nuestro dispositivo.

6.2 Placa de pruebas (Protoboard)

Una placa de pruebas o placa de inserción (en inglés Protoboard o Breadboard) es un tablero con orificios que se encuentran conectados eléctricamente entre sí de manera interna, habitualmente siguiendo patrones de líneas, en el cual se pueden insertar componentes electrónicos y cables para el armado y prototipado de circuitos electrónicos y sistemas similares. Está hecho de dos materiales, un aislante, generalmente un plástico, y un conductor que conecta los diversos orificios entre sí. Uno de sus usos principales es la creación y comprobación de prototipos de circuitos electrónicos antes de llegar a la impresión mecánica del circuito en sistemas de producción comercial. Una placa de pruebas está compuesta por varios bloques de plástico perforados y numerosas láminas delgadas, de una aleación de cobre, estaño y fósforo, que unen dichas perforaciones, creando una serie de líneas de conducción paralelas. Las líneas se cortan en la parte central del bloque para garantizar que dispositivos en circuitos integrados de tipo dual in-line package (DIP) puedan ser insertados perpendicularmente y sin ser tocados por el proveedor a las líneas de conductores. En la cara opuesta se coloca un forro con pegamento, que sirve para sellar y mantener en su lugar las tiras metálicas. Debido a las características de capacitancia (de 2 a 30 pF por punto de contacto) y resistencia que suelen tener las placas de pruebas están confinados a trabajar a relativamente baja frecuencia (inferior a 10 o 20 MHz, dependiendo del tipo y calidad de los componentes electrónicos utilizados). Los demás componentes electrónicos pueden ser montados sobre perforaciones adyacentes que no compartan la tira o línea conductora e interconectados a otros

dispositivos usando cables, usualmente unifilares. Uniendo dos o más placas es posible ensamblar complejos prototipos electrónicos que cuenten con decenas o cientos de componentes.

6.3 Librería RPI-GPIO

Todas las Raspberry Pi de 40 pines (modelos B, 2, 3 y Zero) comparten la distribución de pines. Puedes conectar las cosas directamente o a través de algunos accesorios que los reproducen en una breadboard. En los dibujos a veces empleamos una Raspi Zero. Para instalar las librerías que nos permitan utilizar los puertos GPIO NNos dirigimos al terminar y mediante comandos la descargaremos.

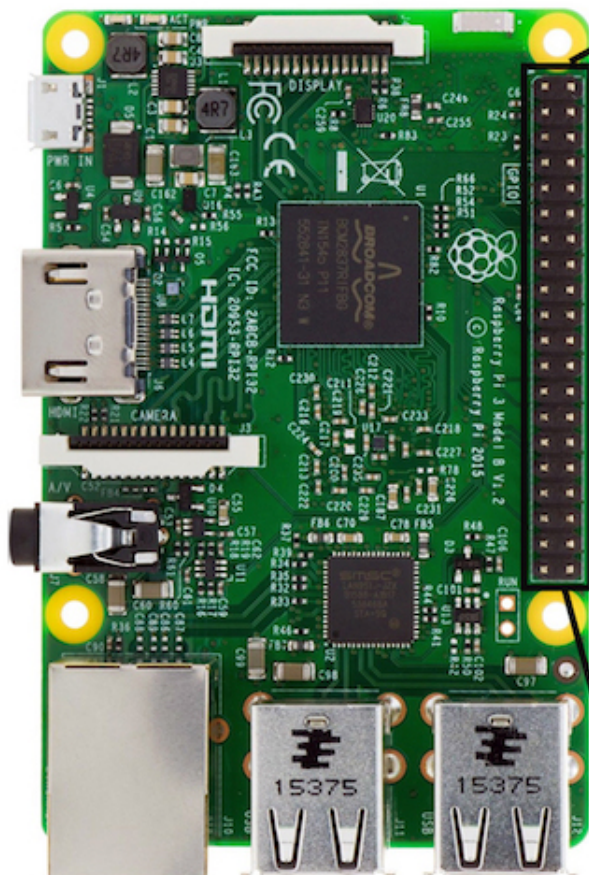
6.4 Hardware del Módulo adaptador LCD a I2C

El Módulo adaptador LCD a I2C que usaremos está basado en el controlador I2C PCF8574, el cual es un Expansor de Entradas y Salidas digitales controlado por I2C. Por el diseño del PCB este módulo se usa especialmente para controlar un LCD Alfanumérico.

La dirección I2C por defecto del módulo puede ser 0x3F o en otros casos 0x27. Es muy importante identificar correctamente la dirección I2C de nuestro módulo, pues de otra forma nuestro programa no funcionará correctamente. Para identificar la dirección específica de nuestro módulo podemos utilizar un pequeño sketch de prueba llamado: I2C Scanner, el cual nos permite identificar la dirección I2C del dispositivo conectado al Arduino. Si en caso existiera la necesidad de trabajar con más de un LCD podemos modificar la dirección I2C del módulo adaptador. Para esto es necesario soldar los puentes A0, A1 y A2 presentes en el módulo, estos tres puentes son los bits menos significativos de la dirección I2C del módulo. La dirección 0x3F en binario sería: 0—0—1—1—1—A2—A1—A0 y la dirección 0x27: 0—0—1—0—0—A2—A1—A0. Por defecto A0, A2, A1 valen 1 pero si soldamos los puentes, estos se conectan a tierra teniendo un valor 0. Por ejemplo si soldamos los tres puentes la nueva dirección sería 0—0—1—0—0—0—0—0 (0x20), para un chip que anteriormente era 0x27.

7 Diagramas

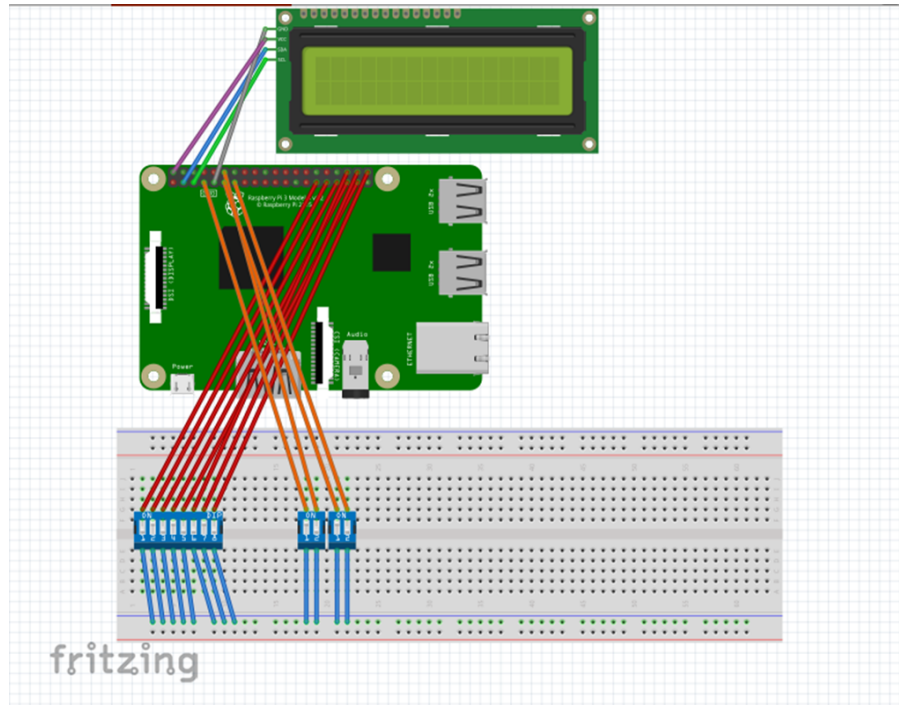
7.1 PUERTOS (PINES) DE LA RASPBERRY PI



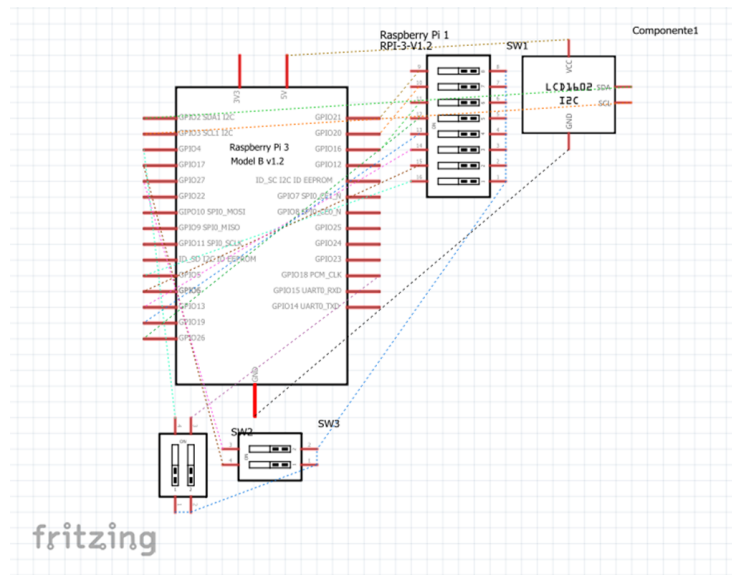
	Pin No.	
3.3V	1	2
GPIO2	3	4
GPIO3	5	6
GPIO4	7	8
GND	9	10
GPIO17	11	12
GPIO27	13	14
GPIO22	15	16
3.3V	17	18
GPIO10	19	20
GPIO9	21	22
GPIO11	23	24
GND	25	26
DNC	27	28
GPIO5	29	30
GPIO6	31	32
GPIO13	33	34
GPIO19	35	36
GPIO26	37	38
GND	39	40
5V		
5V		
GND		
GPIO14		
GPIO15		
GPIO18		
GND		
GPIO23		
GPIO24		
GND		
GPIO25		
GPIO8		
GPIO7		
DNC		
GND		
GPIO12		
GND		
GPIO16		
GPIO20		
GPIO21		

GPIO de raspberry: <https://www.bigmessowires.com/2018/05/26/raspberry-pi-gpio-programming-in-c/>

7.2 CONEXION PANTALLA LCD



7.3 CONEXIÓN DE LOS DIP-SWITCH



8 Lista de componentes

1. Raspberry pi 3.
2. lcd de 16x2 con un modulo i2c.
3. Dip switch de 8 bits.
4. Dip switch de 4 bits.
5. Cables jumpers cortos y largos dependiendo la comodidad y la conexión de la raspberry Pi.
6. Un Protoboard en el que se pueda implementar todo el circuito de un.
7. Programa con interfaz de Python el cual reconozco los diferentes GPIO que necesitaremos .
8. Libreria GPIO
9. Libreria Time
10. Libreria para la implementacion de la LCD

9 Mapa de variables

9.1 CONEXION PANTALLA LCD

CABLE COLOR GRIS	GND
CABLE COLOR VIOLETA	VCC
CABLE COLOR AZÚL	SDA
CABLE COLOR VERDE	SCL

9.2 CONEXIÓN DIP SWITCH 8 BITS

DIP-SWITCH 8 BITS

1	GPIO5	29
2	GPIO6	31
3	GPIO13	33
4	GPIO19	35
5	GPIO26	37
6	GPIO16	36
7	GPIO20	38
8	GPIO21	40

9.3 CONEXIÓN DIP SWITCH 4 BITS

DIP-SWITCH 4 BITS

1	GPIO4	7
2	GPIO18	12
3	GPIO17	11
4	GPIO27	13

10 Explicación del código fuente

1. IMPORTACION DE LIBRERIAS QUE VAMOS A UTILIZAR EN LA REALIZACION DEL CODIGO

```
#IMPORTAMOS LAS LIBRERIAS QUE VAMOS A UTILIZAR
import RPi.GPIO as GPIO
import time
print("Ingrese el primer numero:")
time.sleep(10)

GPIO.setmode(GPIO.BCM)
```

2. DEFINIR LOS PUERTOS GPIO QUE HEMOS UTILIZADO

```
#DEFINIMOS LOS PINES CON LOS GPIO DE LA RASPBERRY PI
pin1 = 5
pin2 = 6
pin3 = 13
pin4 = 19
pin5 = 26
pin6 = 16
pin7 = 20
pin8 = 21

GPIO.setup(pin1, GPIO.IN, GPIO.PUD_UP)
GPIO.setup(pin2, GPIO.IN, GPIO.PUD_UP)
GPIO.setup(pin3, GPIO.IN, GPIO.PUD_UP)
GPIO.setup(pin4, GPIO.IN, GPIO.PUD_UP)
GPIO.setup(pin5, GPIO.IN, GPIO.PUD_UP)
GPIO.setup(pin6, GPIO.IN, GPIO.PUD_UP)
GPIO.setup(pin7, GPIO.IN, GPIO.PUD_UP)
GPIO.setup(pin8, GPIO.IN, GPIO.PUD_UP)
```

3. DEFINIMOS DE LA SIGUIENTE MANERA LA ENTRADA DE LOS NUMEROS BINARIOS QUE SE REALIZARA CON EL DIP SWITCH

```
#SE DEFINE LA ENTRADA DE LOS NUMEROS BINARIOS 1 Y 0
if GPIO.input(pin1) == False:
    print ("1")
else:
    print ("0");
if GPIO.input(pin2) == False:
    print ("1")
else:
    print ("0");
if GPIO.input(pin3) == False:
    print ("1")
else:
    print ("0");|
if GPIO.input(pin4) == False:
    print ("1")
else:
    print ("0");
if GPIO.input(pin5) == False:
    print ("1")
else:
    print ("0");
if GPIO.input(pin6) == False:
```

4. DEFINIMOS LAS VARIABLES QUE CONTIENEN A LOS GPIO DE LA RASPBERRY PI

```
#VARIABLE QUE CONTIENEN A LOS GPIO QUE ESTAN CONECTADOS ENTRE LA RASPBERRY PI Y LOS DIP SWITCHES
aux1=str(GPIO.input(pin1))
aux2=str(GPIO.input(pin2))
aux3=str(GPIO.input(pin3))
aux4=str(GPIO.input(pin4))
aux5=str(GPIO.input(pin5))
aux6=str(GPIO.input(pin6))
aux7=str(GPIO.input(pin7))
aux8=str(GPIO.input(pin8))

dato=aux1+aux2+aux3+aux4+aux5+aux6+aux7+aux8
numerol= int(dato,2)
```

5. VARIABLES DEFINIDAS EN LOS PINES

```
GPIO.setmode(GPIO.BCM) #Sirve para usar números de pin de placa y no del procesador

dip1 = 5 #Variable donde ponemos el pin para el dip 1
dip2 = 6
dip3 = 13
dip4 = 19
dip5 = 26
dip6 = 16
dip7 = 20
dip8 = 21
```

6. DEFINIMOS LAS OPERACIONES QUE REALIZAREMOS CON NUMEROS BINARIOS (SUMA,RESTA,MULTIPLICACION Y DIVISION)

```
if puertofinal ==['0','1','1','1'] :           #DEFINIMOS LAS OPERACIONES CON BINARIOS
    respuesta=numerol+numero2                  #SUMA,RESTA;MULTIPLICACION Y DIVISION
elif puertofinal ==['1','0','1','1'] :
    respuesta=numerol-numero2
elif puertofinal ==['1','1','0','1']:
    respuesta=numerol*numero2
elif puertofinal ==['1','1','1','0']:
    respuesta=numerol/numero2
else:
    mal="Mal"
    print("no se puede realizar")
print("Su resultado es:")
print(respuesta)
GPIO.cleanup()
```

11 Descripción de prerequisites y configuraciones

1. Actualizar los paquetes de la raspberry mediante el comando: `sudo apt-get update`.
2. Actualizar los paquetes de la raspberry mediante el comando: `sudo apt-get upgrade`.

3. Verificar que los modulos del GPIO esten debidamente instalado y actualizados,
4. Ingresamos al terminal de Raspbian en la Raspberry Pi y nos aseguramos que tenemos instalado python
5. Instalamos los paquetes necesarios para el funcionamiento de la LCD 16X2 en la Raspberry Pi mediante linea de comandos en el terminal.
6. Recordar que siempre trabajamos en MODO SUPER USUARIO para ello ejecutamos el comando: `sudo su`
7. Tener una correcta visión y conocimiento acerca de la programacion en python para asi lograr la implementación del código requerido.

12 Aportaciones

La LCD cuenta con un modulo nos permite conectar nuestra Raspberry Pi mas facilmente ya que solo cuenta con 4 pines que son GND,VCC, SDA Y VCL los cuales son conectados directamente a los pines de la Raspberry Pi y asi ahorrarnos la conexion en protoboard con cables Puente los cuales nos llevaria mucho tiempo, asi el modulo que viene incorporado atras de la LCD tambien nos ayuda a regular el brillo y contraste de la LCD y este ya lo tiene incorporado.

12.1 Implementación de binarios a reales

Se ha logrado implementar en el codigo realizado en python la conversión de numeros binarios los cuales son ingresados con el DIP-SWITCH DE 8 BITS a numeros reales los cuales son sometidos a la realizacion de la operacion que se escoja.

13 Conclusiones

En conclusion:

La creacion de de la calculadora, nos ayuda a reforzar y adquirir conocimientos, que nos seran de gran utilidad en niveles superiores.Como la utilizacion de un protoboard, ademas la conexion de la LCD con los cables puente a la Raspberry Pi de manera correcta para que esta pueda funcionar; asi logramos que los conocimientos acerca de pequeñas conexiones realizadas con los cables puente sean de gran ayuda para el proceso de realizacion de una calculadora.

14 Recomendaciones

1. Utilizar el programa fritzing para verificar si las conexiones que se van a realizar sean las correctas antes de proceder a emplearlas en la raspberry.

2. Para mayor facilidad se puede implementar un modulo i2,c el cual permite una conexion mas sencilla de la lcd con la raspberry.
3. Verificar que los cables jumpers esten conectados en los pines en los cuales se los asigno.

15 Bibliografia

Azni, MN, Vellasami, L., Zianal, AH, Mohammed, FA, Daud, NNM, Vejasegaran, R.,... Kan, PLE (2017). Sistema domótico con aplicación de android. En 2016 3ª Conferencia internacional sobre diseño electrónico, ICED 2016 (págs. 299–303). Instituto de Ingenieros Eléctricos y Electrónicos Inc.

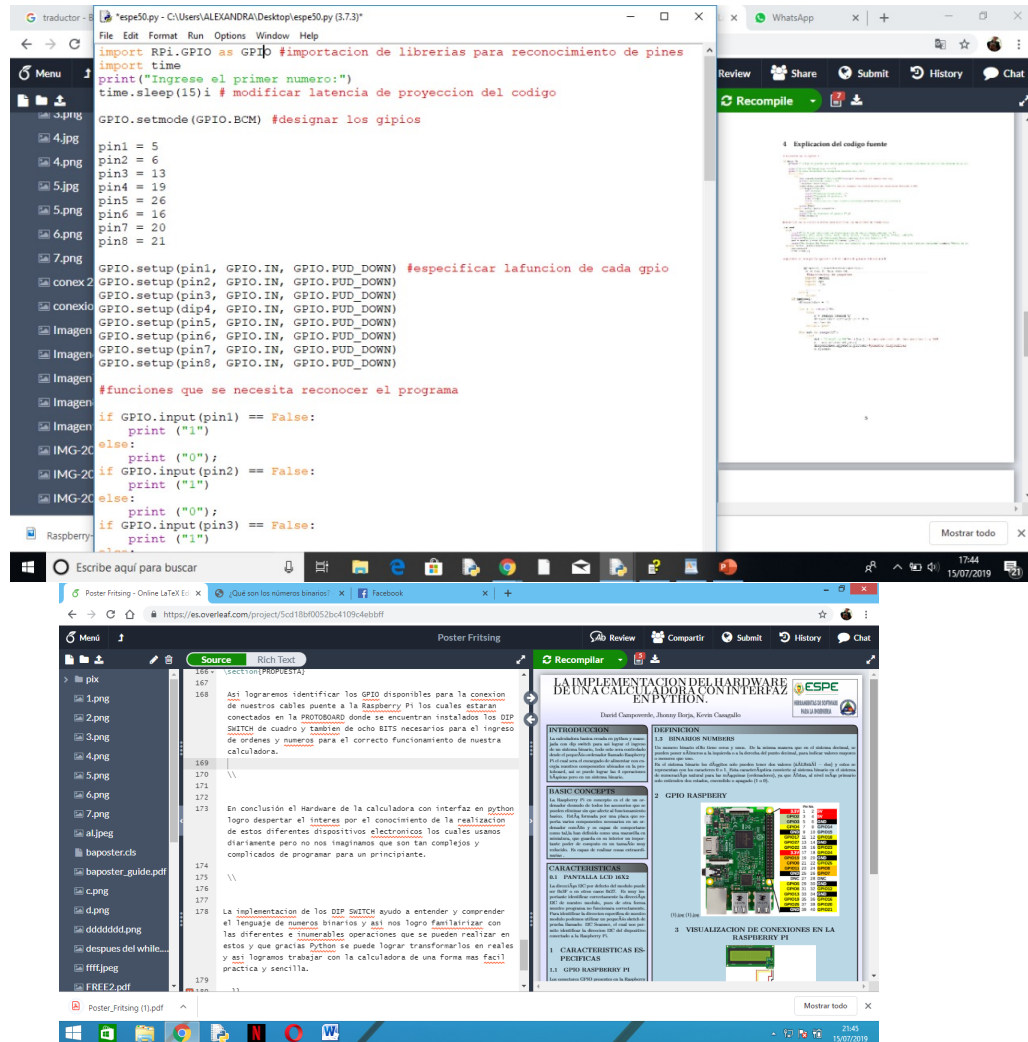
Hoyo, Á., Guzmán, JL, Moreno, JC, y Berenguel, M. (2015). Enseñar conceptos de ingeniería de control utilizando herramientas de código abierto en una placa Raspberry Pi. IFAC-PapersOnLine , 48 (29), 99-104.

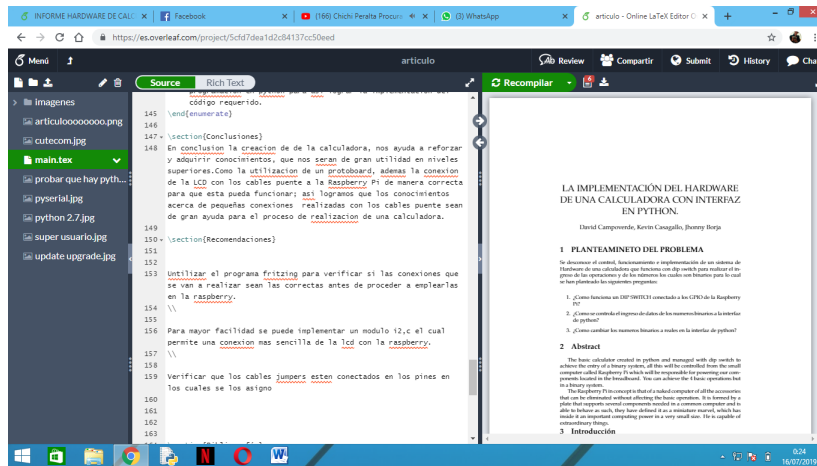
Ersin, Ç., Gürbüz, R., y Yakut, AK (2016). Aplicación de un microcontrolador Arduino basado en un sistema automático de riego de plantas que utilizan energía solar. Fenómenos del estado sólido , 251 , 237–241.

Teckchandani, Y., Perumal, GS, Mujumdar, R., y Lokanathan, S. (2016). Sistema de visualización de avisos inalámbricos en pantalla grande. En 2015, Conferencia Internacional IEEE sobre Inteligencia Computacional e Investigación en Computación, ICCIC 2015 . Instituto de Ingenieros Eléctricos y Electrónicos Inc.

16 Anexos

16.1 Repositorios





17 Manual de usuario

1. Comprobar que tenemos instalado Python y si no lo tenemos instalado lo instalamos
2. Instalar las librerías GPIO para que así nos libre reconocer los puertos de la raspberry Pi
3. Instalacion e implementacion de los DIP SWITCH en la protoboard
4. Conectar los cables puente correctamente a los puertos gpio que hemos escogido
5. Respetar siempre las conexiones GND o tierra.
6. La realizacion del codigo siempre empezar importando la libreria GPIO para que nos reconozca los DIP SWITCH.
7. Realizar el codigo para lograr cambiar de binario a numeros normales
8. Comprobar las correctas conexiones de los cables para así tener una comunicacion GPIO correcta.
- 9.

Siempre que vamos a empezar con algo en la Raspi conviene asegurarnos de que tenemos al día las librerías necesarias, y por eso es buena costumbre empezar actualizando todo.

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

Conexión de Cables macho-hembra y macho-macho De cada pin Di switch vamos a colocar a tierra en el protoboard y mandan el cable de tierra del probboard

a tierra de la Raspberry Identificamos los pines GPIO de la Raspberry que hemos seleccionado para cada pin del Dissuichw y proceden a conectar de forma correcta. DID8 GPIO PIN(RASPBERRY)

1 5 29 2 6 31 3 13 33 4 19 35 5 26 37 6 16 36 7 20 30 8 21 40

Lo mismo se aplica con el Dip switch de 4 pines DID GPIO PIN 1 4 7 2 18 12 3 17 11 4 27 13

Una vez hecha la conexión del Did Swich a la raspberry procedemos a conectar la pantalla LCD identificamos los pines LCD(gris) - GND LCD(violeta) - VCC LCD(azul) - SDA LCD(verde) - SCL

Ingresamos a Raspberry pi al terminal e introducimos el siguiente comando

Para configurar LCD en raspberry

1. ls (enter)
2. cd lcd (enter)
3. sudo sh install.sh (comenzara a instalarse y reiniciamos)
4. cd lsd (introducimos para ingresar a la lcd)
5. nano demo_lcd.py(*intrudcimos para controlar la lcd Senos abrir a la sección donde vamos a configurar las letras que*
TODA LA PROGRAMACION SE REALIZO EN PYTHON.
Se nos abre a la sección donde vamos a configurar las letras que queremos que salgan en nuestra pantalla LCD)