UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIAFACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO



MANUAL DE PRÁCTICAS

UNIDAD DE APRENDIZAJE: SISTEMAS EMPOTRADOS

PROFESOR:

DR. EVERARDO INZUNZA GONZÁLEZ

FORMATO DEL REPORTE DE PRÁCTICAS DEL LABORATORIO

PORTADA

Nombre de la universidad

Nombre de la Facultad

Carrera

Materia

Alumno

Nombre y número de Práctica

Nombre del maestro

Lugar y fecha

CONTENIDO

Objetivo

Lista de material y equipo

Antecedentes relacionados al tema de la práctica

Desarrollo experimental

Diagrama a bloques

Descripción detallada del procedimiento

Fotografías de los prototipos y experimentos realizados

Análisis y discusión de resultados

Código fuente de los programas realizados

Screen o pantallazos de los resultados visualizados en el monitor y del trabajo realizado en la práctica (de los comandos o software utilizado)

Conclusiones individuales

Bibliografía

PRÁCTICA #1

Nombre: Instalación de Sistema Operativo LINUX/RASPBIAN.

Objetivo: Formatear una memoria MicroSD para que sea compatible con el sistema operativo LINUX/RASPBIAN, además instalar el sistema operativo LINUX/RASPBIAN en la MicroSD.

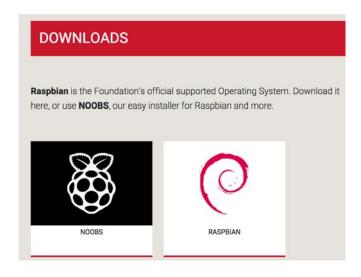
Material:

- 1 Computadora Raspberry Pi 2
- 1 Monitor HDMI
- 1 Teclado y mouse
- 1 Dongle WiFi
- 1 Fuente de poder para Raspberry Pi (5V DC @ 2A)
- 1 Conexión a red ethernet (Nodo y cable de red con conectores RJ-45)
- 1 Memoria MicroSD clase 10 o similar con capacidad mínima de 8 GB

A) PROCEDIMIENTO

 Usando una computadora personal descargar el sistema operativo LINUX/RASPBIAN desde el sitio oficial de Raspberry Pi

https://www.raspberrypi.org/
https://www.raspberrypi.org/downloads/

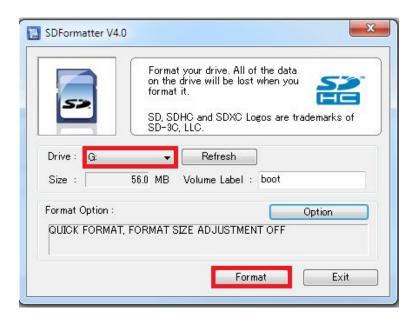


Elaboró: Dr. Everardo Inzunza González

2. Seleccionar la versión **NOOBS** (New Out Of the Box Software)



- 3. Hacer click en **Download ZIP** debajo de NOOBS (offline and network install) y seleccione el folder donde desea guarder el archive .zip
- 4. Descomprima el archivo .zip
- 5. **Inserte la memoria MicroSD** en el lector de su computadora personal, observe que letra de disco duro se le asignó (Ej. D:/, E:/, etc)
- 6. **Formatee su memoria MicroSD**, elija formatoo FAT32. (Puede utilizar el software SD Fromatter 4.0 o superior) (Elija bien el drive de su microSD).



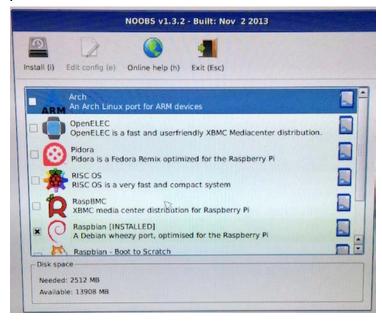
Elaboró: Dr. Everardo Inzunza González

COPIANDO LOS ARCCHIVOS DE NOOBS A LA MICRO SD

- 7. Una vez que la memoria MicroSD ha sido formateada, **copie los archivos extraíd**os en la carpeta NOOBS **hacia la memoria MicroSD**. (Los archivos del sistema operativo deben ser trasnferidos a la memoria MicroSD).
- 8. Una vez finalizado la transferencia de archivos en la MicroSD, **retire cuidadosamente la MicroSD** del lector.
- 9. **Inserte la memoria MicroSD en el lector de la Raspberry Pi** (Asegúrese que esté bien conectada y no quede haciendo falso contacto).

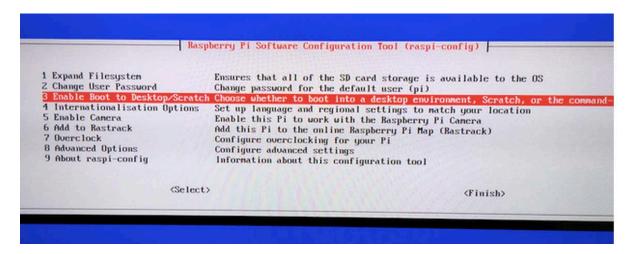
Primer arrangue (First Boot)

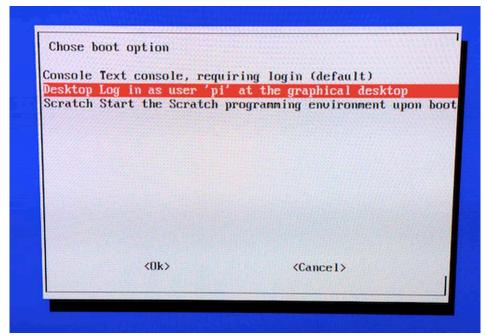
- 10. Conecte el teclado a la Raspberry
- 11. Conecte el monitor HDMI a la Raspberry.
- 12. Conecte el cable de red Ethernet
- 13. Conecte la fuente de alimentación en el puerto Micro USB de la Raspberry
- 14. Conecte la fuente de alimentación al tomacorriente.
- 15. La Raspberry deberá arrancar y aparecerá una ventana con la lista de diferentes sistemas opersativos que se pueden instalar.
- 16. Se recomienda que utilice RASPBIAN.



Elaboró: Dr. Everardo Inzunza González

- 17. Posteriormente, seleccione el idioma y hacer click en el icono Install.
- 18. El proceso de instalación demora aproximadamente 20 minutos.
- 19. **Cuanto esté completa la instalación, hacer Click** en el botón OK y esperar a que re-inicie el sistema.
- 20. Una vez re-iniciado el sistema, **escoja el ambiente de arranque** (Consola, Escritorio o Scratch).





- 21. **Explore el Escritorio de Raspberry**, pruebe el navegador de internet y comience a instalar el software de su interés.
- 22. Disfrute la Raspberry Pi.

PRÁCTICA #2

Nombre: Conociendo la Raspberry, actualizando el sistema operativo e instalando paquetes.

Objetivo: Familiarizarse con el entorno de trabajo de la Raspberry Pi, conocer el sistema operativo Raspbian, actualizar el sistema e instalar diferentes paquetes.

Material:

- 1 Computadora Raspberry Pi 2 o superior
- 1 Monitor HDMI
- 1 Teclado y mouse
- 1 Dongle WiFi
- 1 Fuente de poder para Raspberry Pi (5V DC @ 2A)
- 1 Conexión a red ethernet (Nodo y cable de red con conectores RJ-45)

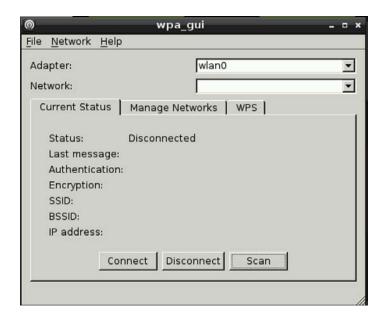
I) PROCEDIMIENTO

A) CONECTANDO LA RASPBERRY

- 1) Conectar teclado y mouse
- 2) Conectar monitor
- 3) Conectar Dongle Wi Fi o cable Ethernet RJ-45
- 4) Encender monitor
- 5) Conectar fuente de poder a Raspberry
- 6) Esperar a que arranque la Raspberry (si es la primera vez que arranca, configurar zona horaria, idioma de teclado, expandir partición de disco, etc.)

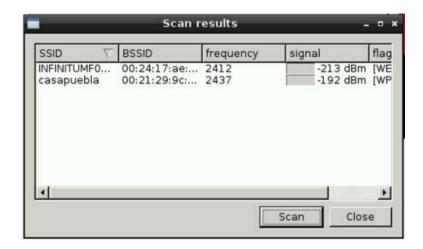
B) Configurando la conexión de red (Si tiene Raspi 3, pase al inciso C)

D) En la consola o terminal ejecutar el comando wpa_gui



2.) Hacer click en el Botón SCAN

3) Se abre una segunda ventana con el listado de las redes detectadas. Damos doble click en la red correspondiente, escribimos la **contraseña** en el campo **PSK** y elegimos **Add.**



Elaboró: Dr. Everardo Inzunza González

4) En la ventana principal de configuración damos click en Connect y listo, ahora tenemos conexión a internet, incluso podemos ver la dirección IP en la parte inferior de la ventana.



C) ACTUALIZANDO EL SISTEMA

D) Primero se tiene que actualizar (descargar) la lista de los paquetes con el siguiente ommando:

\$ sudo apt-get update

2) Luego se **instalan** todas las actualizaciones

\$ sudo apt-get upgrade

D) INSTALANDO SOFTWARE

Nota: Siempre que se va instalar un nuevo software o paquete en Raspi, se recomienda primero hacer un actualización del sistema (update y upgrade).

Nombre del paquete/software	Comando Raspbian
Programa para hacer ScreenShots con Raspi	sudo apt-get install scrot
Libre Office (Similar/compatible con Microsoft Office)	sudo apt-get install libreoffice
Okular (Para abrir archivos PDF)	sudo apt-get install okular
Actualizando navegador de internet de Raspberry	sudo apt-get update
	sudo apt-get dist-upgrade
	sudo apt-get install epiphany-browser
Instalando navegador Mozilla FireFox (IceWeasel)	sudo apt-get install iceweasel
Editor de texto para programar en Lenguaje C o Python	sudo apt-get install geany
Reproductor de videos	sudo apt-get install omxplayer
Editor de imágenes	sudo apt-get install gimp
Para comprimir archivos y crear .zip	sudo apt-get install zip
Para descomprimir archivos zip	sudo apt-get install unzip

Para Des-instalar un programa o paquete:

sudo apt-get remove <nombre-del-paquete/software>

Nota: Para ejecutar el Sistema operativo LINUX en ambiente gráfico: startx

E) NAVEGANDO EN LA RED

- 1. Probar los navegadores de internet utilizando el buscador GOOGLE.
- 2. Probar la vizualización de videos en youtube.
- 3. Revisar el correo de UABC o GMAIL.
- 4. Iniciar sesión en facebook o alguna otra red social.

F) Revisar/Consultar el Hardware de la Raspberry utilizando los comandos de Raspbian

- Para conocer la información hardware general: cat /proc/cpuinfo
- Para saber el estado de la memoria: at /proc/meminfo
- Para ver las particiones de la tarjeta de memoria o el disco duro: cat /proc/partitions
- Si queremos conocer la versión de nuestra Rasp: cat /proc/version
- Temperatura de la *CPU*: vcgencmd measure temp
- Visualizar todos los dispositivos USB conectados: 1susb

Todos estos comandos nos permitirán **conocer con detalle todo lo que tiene nuestro dispositivo** y así saber qué podemos y qué no podemos hacer con él.

PRÁCTICA #3

Nombre: Instalación y configuración de escritorio remoto en Raspberry Pi 2.

Objetivo: Conectar la Raspberry Pi a través de una red LAN con una PC/Laptop para ejecutar el escritorio de la Raspberrey Pi desde el sistema operativo de la PC/Laptop.

Material:

- 1 Computadora Raspberry Pi 2
- 1 Dongle WiFi
- 1 Fuente de poder para Raspberry Pi (5V DC @ 2.5A)
- 1 Cable de conexión a red Ethernet
- 1 Puerto de red Ethernet (Router o Acces point con puerto Ethernet)
- 1 Memoria MicroSD clase 10 o similar con capacidad mínima de 8 GB
- 1 PC o Laptop

1. PROCEDIMIENTO

- 1. Conectar dongle wifi, mouse, teclado, monitor por medio del cable HDMI y conectar la fuente de poder a la raspberry.
- 2. Para encender conectar la fuente a la línea de AC
- En la raspberry actualizar los paquetes y sistema operativo sudo apt-get update sudo apt-get upgrade
- 4. Instalar los siguientes programas, empleando los siguientes comandos en el Shell de la Raspberry Py:

sudo apt-get install vnc4server sudo apt-get install xrdp

Hay que configurar el programa **VNCSERVER**, pedirá una clave de 8 dígitos y que la verifique dos veces, ejecute el siguiente comando.

pi@raspberrypi ~ \$ vncserver :1

1. Conectar la Raspberry a la PC o laptop con el cable Ethernet



(Verifique que la luz Ethernet esté encendida)

2. Después haremos que la IP del adaptador Ethernet de la raspeberry sea estático para evitar problemas en conexiones futuras. Escribimos en la terminal el siguiente comando **ifconfig** y aparecerá lo siguiente:

```
🜉 pi@raspberrypi: ~
 File Edit Tabs Help
pi@raspberrypi:~ $ ifconfig
           Link encap:Ethernet Hwaddr b8:27:eb:7c:9d:18
inet addr:169.254.57.126 Bcast:169.254.255.255 Mask:255.255.0.0
inet6 addr: fe80::4b70:ec1a:b334:b029/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
           Link e
           RX packets:10317 errors:0 dropped:0 overruns:0 frame:0
            TX packets:13854 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000 '
RX bytes:776536 (758.3 KiB) TX bytes:8288926 (7.9 MiB)
           Link encap:Local Loopback
            inet addr:127.0.0.1 Mask:255.0.0.0
            inet6 addr: ::1/128 Scope:Host
           UP LOOPBACK RUNNING MTU:65536 Metric:1
           RX packets:119244 errors:0 dropped:0 overruns:0 frame:0
           TX packets:119244 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:0
RX bytes:253817191 (242.0 MiB) TX bytes:253817191 (242.0 MiB)
wlan0
           Link encap:Ethernet Hwaddr 00:0f:60:05:4a:de
           inet addr:192.168.1.158 Bcast:192.168.1.255 Mask:255.255.25
            inet6 addr: fe80::3104:1ddb:9c23:63e2/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
           RX packets:8907 errors:0 dropped:2153 overruns:0 frame:0
           TX packets:312 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000
            RX bytes:970317 (947.5 KiB) TX bytes:38108 (37.2 KiB)
pi@raspberrypi:~ $
```

3. Anotamos **addr** y **mask** de eth0, ya que se necesitará posteriormente, en nuestro caso son.

addr: 169.254.57.126 mask: 255.255.0.0

4. Ejecutar:

sudo nano /etc/dhcpcd.conf

5. Al final del archivo abierto, agregar lo siguiente:

Custom static IP address for eth0.

interface eth0 static ip_address=192.168.0.200 static routers=192.168.0.1 static domain_name_servers=192.168.0.1

(Recuerde que lo tienen que hacer con sus números IPs, este es solo un ejemplo).

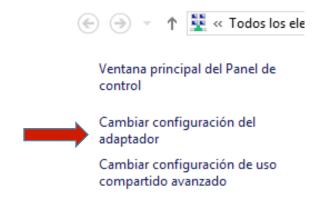
10. Guardar los cambios al archivo y salir.

Salvar la configuración (Ctr+X)+Y y re-iniciar la Raspberry Pi, con el comando **sudo shutdown -r now**.

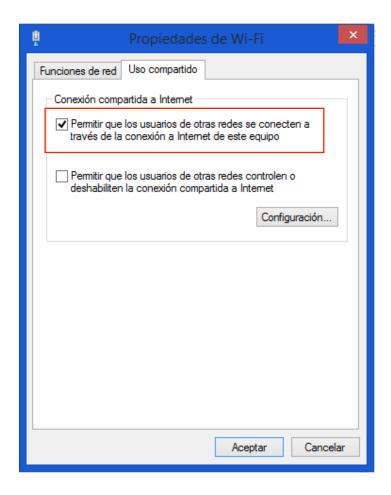
- 11.. Reiniciar la Raspberry Pi.
- 12. Ahora en nuestra computadora personal abrir acceso a la red LAN de forma manual dando clic derecho en el icono mostrado en el recuadro rojo, seleccionar **Centro de Redes y Recursos compartidos.**



13. Seleccionar Cambiar configuración del Adaptador

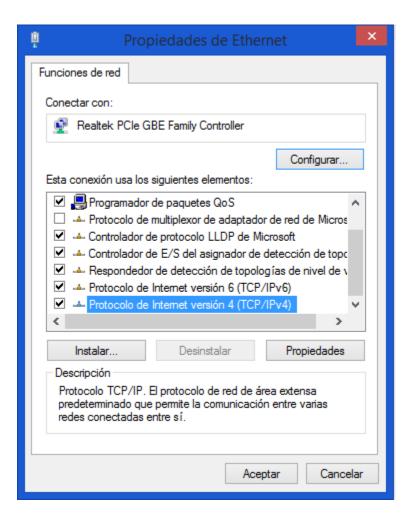


14. Seleccionar las propiedades del adaptador Wi-Fi y en la pestaña de **uso compartido** seleccionamos la siguiente opción como se muestra en la figura y damos aceptar.

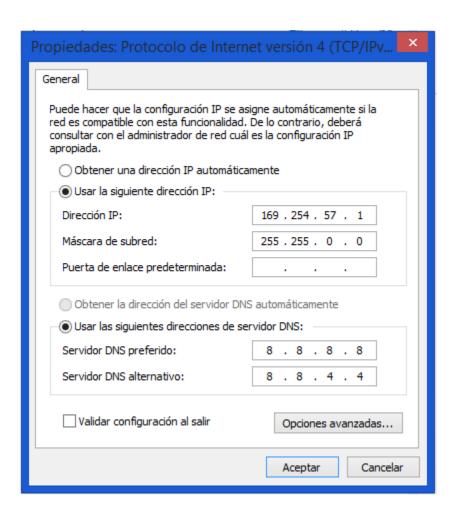


15. Seleccionar las propiedades del puerto Ethernet dando clic derecho sobre él.

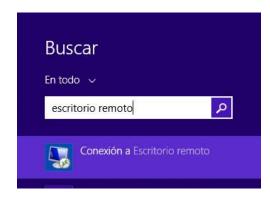
16. Seleccionar las propiedades del TCP/IPv4.



17. Ahora llenamos los campos como en la siguiente figura. En **Dirección IP** pondrás lo mismo que pusimos en **gateway** en el paso 8, al igual que **mascara de subred** será **netmask**, damos aceptar para guardar los cambios realizados.

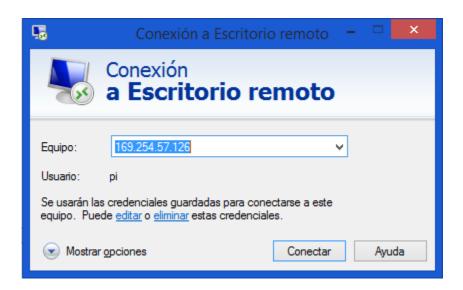


18. Buscamos escritorio remoto.

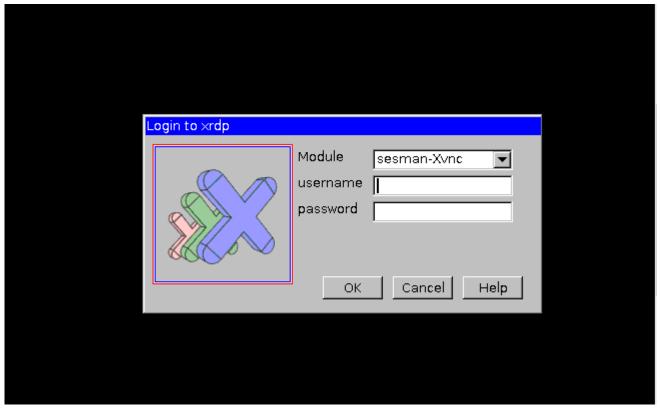


Elaboró: Dr. Everardo Inzunza González

19. En la ventana de escritorio remoto tecleamos la IP de nuestra Raspberry (**addr** del paso 8) y damos click en **Conectar.**



20. Después aparecerá lo siguiente

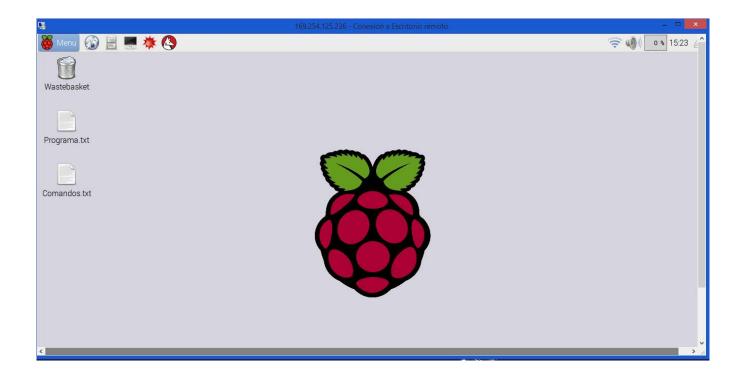


Para acceder tecleamos:

Elaboró: Dr. Everardo Inzunza González

Username: pi

Password: raspberry



Y con esto logramos la conexion a nuestra raspberry.

PRÁCTICA #4

Nombre: Control secuencial de luces empleando Raspberry Pi

Objetivo: Desarrollar un programa en C para que controle el encendido/apagado en forma secuencial con un retardo entre secuencias de medio segundo segundo.

Requerimientos:

Se requiere que el sistema encienda secuencialmente 8 luces con retardo intersecuencias de 500 mS. Posteriormente que las vaya apagando con el mismo tiempo de retardo.

Lista Material y equipo

- 1 Computadora Raspberry Pi 2
- 1 Monitor HDMI
- 1 Teclado y mouse
- 1 Dongle WiFi
- 1 Fuente de poder para Raspberry Pi (5V DC @ 2A)
- 1 Conexión a red ethernet (Nodo y cable de red con conectores RJ-45)
- 1 Memoria MicroSD clase 10 o similar con capacidad mínima de 8 GB
- 8 LED's

A) PROCEDIMIENTO

- 1. Escribir el programa en lenguaje C para Raspberry Pi.
- 2. Compilar el programa.
- 3. Conectar los leds a los pines GPIO con su respectiva resistencia limitadora de corriente.
- 4. Probar el circuito incluyendo la Raspberry Pi
- 5. Realizar los ajustes correspondientes al programa/circuito.
- 6. Compilar programa y revisar circuito nuevamente
- 7. Probar el circuito.

Código ejemplo

```
secuencia8.c ×
 1
       * secuencia8.c:
 2
 3
       * Enciende una secuencia de LED utilizando los primeros
      * 8 WiringPi pines (GPIO) con un retardo inter secuencias
 4
     * de 0.5 Segundos
*/
 5
 6
 7
 8
      #include <stdio.h>
 9
      #include <wiringPi.h>
10
      int main (void)
11
12
    ₽{
        int i, led;
13
        clr();
printf ("\n");
14
15
        printf ("Raspberry Pi - Secuencia de 8 LEDS\n");
16
        printf ("===
17
18
        printf ("\n");
        printf ("Conecte 8 LEDS en los primeros 8 wiringPi pines.\n");
19
20
        wiringPiSetup ();
21
22
        for (i = 0; i < 8; ++i)
23
          pinMode (i, OUTPUT);
24
25
26
        for (;;)
27
28
          for (led = 0 ; led < 8 ; ++led)</pre>
29
30
            digitalWrite (led, 1);
31
            delay (500) ;
32
33
          for (led = 0 ; led < 8 ; ++led)</pre>
34
35
             digitalWrite (led, 0);
36
            delay (500);
37
38
39
        }
40
41
```

PRÁCTICA #5

Nombre: Control secuencial de motores con Raspberry Pi

Objetivo: Desarrollar un programa en C para que encienda 5 motores de CD en forma

secuencial con un retardo entre secuencias de dos segundos.

Requerimientos:

El sistema deberá tener un botón de arranque para que inicie el encendido secuencial y

un botón de paro secuencial en orden inverso a la secuencia de arrangue. Además

considere el uso de un botón de paro de Emergencia para apagar inmediatamente (al

mismo tiempo) todos los motores. Los motores deberán conectarse a los pines GPIO por

medio de una etapa de potencia (Por ejemplo usar transistor de potencia TIP 41C). El

tiempo inter-secuencia es de 2000 mS.

Lista Material y equipo

1 Computadora Raspberry Pi 2 o superior

1 Monitor HDMI

1 Teclado y mouse

1 Dongle WiFi

1 Fuente de poder para Raspberry Pi (5V DC @ 2A)

1 Conexión a red ethernet (Nodo y cable de red con conectores RJ-45)

1 Memoria MicroSD clase 10 o similar con capacidad mínima de 8 GB

5 motores de DC

5switches o push buttons

Elaboró: Dr. Everardo Inzunza González

A) PROCEDIMIENTO

- 1. Escribir el programa en lenguaje C para Raspberry Pi.
- 2. Compilar el programa.
- 3. Conectar los swtiches, etapa de potencia y motores a los pines GPIO programados utilizando las resistencias limitadoras de corriente (330 Ohms)
- 4. Probar el circuito incluyendo la Raspberry Pi
- 5. Realizar los ajustes correspondientes al programa/circuito.
- 6. Compilar programa y revisar circuito nuevamente
- 7. Probar el circuito.
- 8. Elaborar el reporte de práctica.

Elaboró: Dr. Everardo Inzunza González

PRÁCTICA #6

Nombre: Alarma digital basada en Raspberry Pi

Objetivo: Diseñar y construír una alarma digital utilizando la Raspberry Pi y lenguaje de programación C.

- 1 Computadora Raspberry Pi 2
- 1 Monitor HDMI
- 1 Teclado y mouse
- 1 Dongle WiFi
- 1 Fuente de poder para Raspberry Pi (5V DC @ 2A)
- 1 Conexión a red ethernet (Nodo y cable de red con conectores RJ-45)
- 1 Memoria MicroSD clase 10 o similar con capacidad mínima de 8 GB
- 1 LED ultra-brillante
- 4 sensores on/off para alarma (Movimiento, Reed switch, ópticos, etc.).
- 1 dip switch (para Activar o desactivar la alarma)

Requerimientos generales del sistema

El sistema tendrá como entrada 4 sensores del tipo ON/OFF y una entrada para ACTIVAR/DESACTIVAR la alarma. Estas 5 entradas serán a través de los puertos GPIO de la Raspberry Pi, pueden ser activadas en ALTO/BAJO, esto depende del tipo de sensor. Cuando la alarma esté habilitada y cualquiera de los sensores se active, la alarma deberá encender una luz de emergencia, un BUZZER de 5V y un LED ultra-brillante. Cuando la alarma esté des-habilitada, deberá ignorar todas las señales provenientes de los sensores.

ALARMA DIGITAL

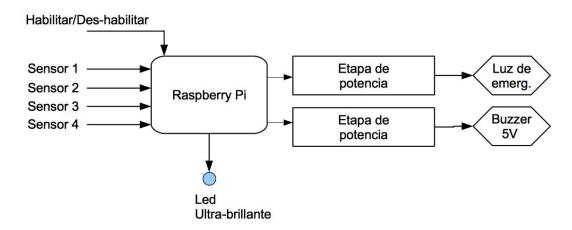


Figura 1. Diagrama a bloques de la alarma basada en Raspberry Pi.

Nota 1: Recuerde que todas las entradas GPIO de la Raspi deben ser de 3.3V.

A) PROCEDIMIENTO

- 1. Escribir el programa en lenguaje C para Raspberry Pi.
- 2. Compilar el programa.
- 3. Conectar los sensores, Buzzer y LED's a los pines GPIO programados utilizando las resistencias limitadoras de corriente (330 Ohms)
- 4. Probar el circuito.
- 5. Realizar los ajustes correspondientes al programa/circuito.
- 6. Compilar programa y revisar circuito nuevamente
- 7. Probar el circuito.
- 8. Escribir el reporte de la práctica

Nota 2: Es importante interconectar diversos sensores a la Raspi para que la práctica sea válida.

PRÁCTICA #7

Nombre: Configuración y uso de puertos seriales RS-232 en una Raspberry Pi 2 utilizando Python.

Objetivo: Configurar y habilitar el puerto serial RS-232 de la Raspberry pi 2 y hacer un programa en Python que utilice dicho puerto y un puerto genérico USB para transmisión y recepción de datos en forma serial.

Material:

- 1 Convertidor TTL a RS-232 (Chip integrado MAX 232)
- 1 Convertidor RS-232 a USB
- 1 Conector DB-9 Hembra
- 1 Conector DB-9 Macho
- 1 Computadora Raspberry Pi 2
- 1 Monitor HDMI (opcional)
- 1 Teclado y mouse (opcional)
- 1 Dongle WiFi
- 1 Fuente de poder para Raspberry Pi (5V DC @ 2A)
- 1 Conexión a red ethernet (Nodo y cable de red con conectores RJ-45)
- 1 Software Python 2.7.4

A) PROCEDIMIENTO

- 1. Primero es necesario actualizar la consola sudo apt-get update y sudo apt-get upgrade.
- 2. Una vez actualizada la consola, escriba en la terminal dmesg | grep tty, este comando busca en las características de la consola el texto tty, el cual se relaciona a la configuración del puerto serial, buscamos deshabilitar una configuración de fábrica, ya que ciertas características de control fuera del tema de estudio se hacen por medio de la comunicación serial.

Se observarán algunos puertos tty en color rojo.

Despues de ejecutar el comando dmesg | grep tty, se debe visualizar al parecido a la siguiente información:

```
pi@raspberrypi ~ $ dmesg | grep tty

[ 0.000000] Kernel command line: dma.dmachans=0x7f35 bcm2708_fb.fbwidt
h=656 bcm2708_fb.fbheight=416 bcm2709.boardrev=0xa01041 bcm2709.serial=0x93
f9c7f9 smsc95xx.macaddr=B8:27:EB:F9:C7:F9 bcm2708_fb.fbswap=1 bcm2709.disk_
led_gpio=47 bcm2709.disk_led_active_low=0 sdhci-bcm2708.emmc_clock_freq=250
000000 vc_mem.mem_base=0x3dc00000 vc_mem.mem_size=0x3f000000 dwc_otg.lpm_e
nable=0 console=tty1 console=ttyAMA0,115200 root=/dev/mmcblk0p2 rootfstype=
ext4 elevator=deadline rootwait

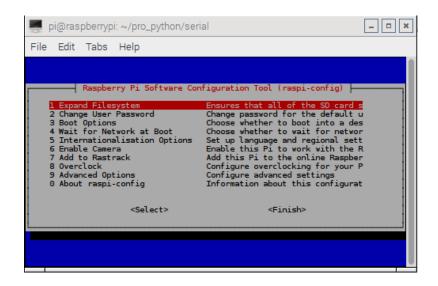
[ 0.001774] console [tty1] enabled
[ 0.749509] dev:f1: ttyAMA0 at MMIO 0x3f201000 (irq = 83, base_baud =
0) is a PL011 rev3
[ 1.268971] console [ttyAMA0] enabled
pi@raspberrypi ~ $
```

3. Para visualizar los todos puertos seriales disponibles en Raspberry Pi, teclee el comando:

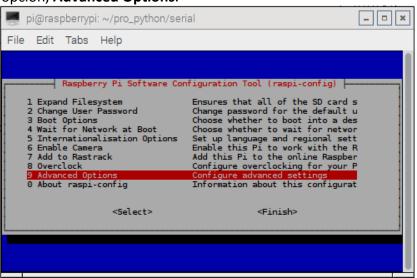
GPIO serial port is /dev/ttyAMA0

En linux los puertos seriales típicamente se llaman de la siguiente manera:

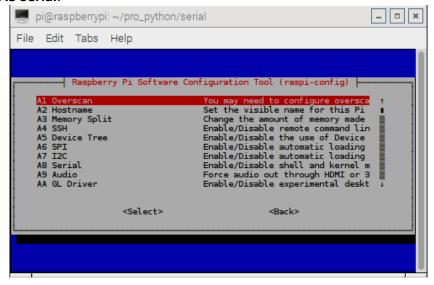
- Built-in (standard) Serial Port: the Linux standard is /dev/tty0, /dev/tty1, and so on.
- USB Serial Port Adapter: /dev/ttyUSB0, /dev/ttyUSB1, and so on.
 - o Some types of USB serial adapter may appear as /dev/ttvACM0 ...
- 4. Se debe configurar el puerto serial, en la terminal escribe **sudo raspi-config**, se abrirá un menú en entorno gráfico.



5. Seleccionaras la opción, Advanced Options.

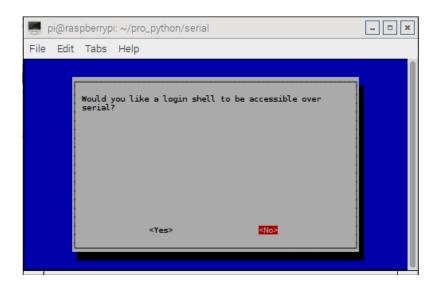


6. Seleccionaras A8 Serial.



Elaboró: Dr. Everardo Inzunza González

7. Posteriormente, **Seleccionas NO**, con esto aparece deshabilitado el puerto serial, de esta forma podemos utilizar de forma genérica el puerto serial.



- 8. Seleccionar FINISH
- 9. RE-INICIAR LA RASPBERRY PARA QUE SE CONFIGUREN LOS CAMBIOS.

Nota: Es muy importante hacer esta re-inicialización para que se actualicen los cambios.

10. Escriba de nuevo el comando: dmesg | grep tty (Para verificar que ya está des-activada la cnonsola ttyAMAO), se debe visualizar algo similar a la siguiente figura:

```
pi@raspberrypi ~ $ dmesg | grep tty

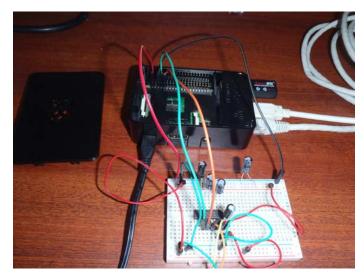
[ 0.000000] Kernel command line: dma.dmachans=0x7f35 bcm2708_fb.fbwidt
h=656 bcm2708_fb.fbheight=416 bcm2709.boardrev=0xa01041 bcm2709.serial=0x93
f9c7f9 smsc95xx.macaddr=B8:27:EB:F9:C7:F9 bcm2708_fb.fbswap=1 bcm2709.disk_
led_gpio=47 bcm2709.disk_led_active_low=0 sdhci-bcm2708.emmc_clock_freq=250
000000 vc_mem.mem_base=0x3dc00000 vc_mem.mem_size=0x3f000000 dwc_otg.lpm_e
nable=0 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline
rootwait

[ 0.001769] console [tty1] enabled
[ 0.749438] dev:f1: ttyAMA0 at MMIO 0x3f201000 (irq = 83, base_baud =
0) is a PL011 rev3
pi@raspberrypi ~ $
```

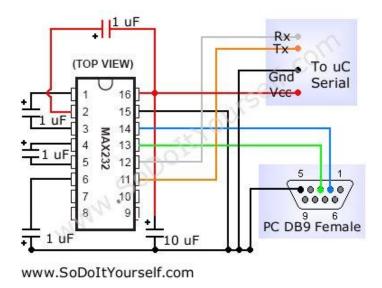
Observe que en el primer párrafo debe aparecer: console=tty1

11. Antes de continuar debe conectar de forma apropiada el convertidor TTL a RS-232, el pin el 8 y 10 son los pines de comunicación UART, RX y TX, respectivamente.





12. Si tiene el convertidor TTL a RS-232 pase al número 9, de lo contrario, arme el siguiente circuito:



Nota: Vcc es de 5 Volts

Elaboró: Dr. Everardo Inzunza González

13. Escribir el siguiente código ejemplo, con nombre de archivo serial_wryte.py (TRANSMISOR DE INFORMACIÓN)

```
→ ⊗ lee_ser.py ⊗ ser_write.py*
 1#!/usr/bin/python
 3 import time # importa librerias para manejo de retardos
 4 import serial # importa libreria de comunicacion serial
 6# Configura Puerto serial RS-232
 7 ser = serial.Serial(
 8 port='/dev/ttyAMA0',
 9 \text{ baudrate} = 9600,
10 parity=serial.PARITY_NONE,
11 stopbits=serial.STOPBITS_ONE,
12 bytesize=serial.EIGHTBITS,
13 timeout=1)
15 counter=0 # Inicializa en cero cotandos
17 print ("Transmitiendo contador") # imprime mensaje
18 while 1:
      ser.write('Transmitiendo contador: %d \n'%(counter)) # envia cadena de caracteres
19
20
      time.sleep(1) #retardo de 1 segundo
      counter += 1 # incrementa contados
21
22
      print counter
```

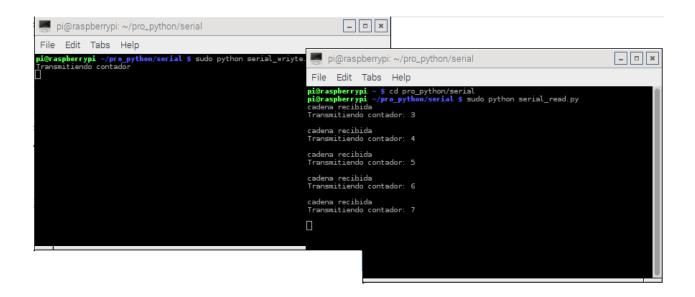
14. Escribir el siguiente código, serial_read.py (RECEPTOR DE INFORMACIÓN)

```
lee_ser.py Ser_write.py*
 1#!/usr/bin/env python
 3 import time # importa librerias para manejo de retardos
 4 import serial # importa libreria de comunicacion serial
 6# Configura Puerto serial RS-232
 7 ser = serial.Serial(
      port='/dev/ttyAMA0',
 9
      baudrate = 9600,
10
      parity=serial.PARITY NONE,
11
      stopbits=serial.STOPBITS_ONE,
12
      bytesize=serial.EIGHTBITS,
13
      timeout=1)
14
15 counter=0
16
17 while 1:
      x=ser.readline() # Lee informacion recibida por Puerto serial
18
19
      print 'cadena recibida'
20
      print x
21
```

15. Ejecutar los dos códigos en dos terminales Linux por separado (Asegúrese que esté conectado el circuito convertidor TTL a RS-232 y el convertidor RS-232 a USB), o bien **conecte un puente entre los pines físicos 8 y 10 (Tx y Rx).**

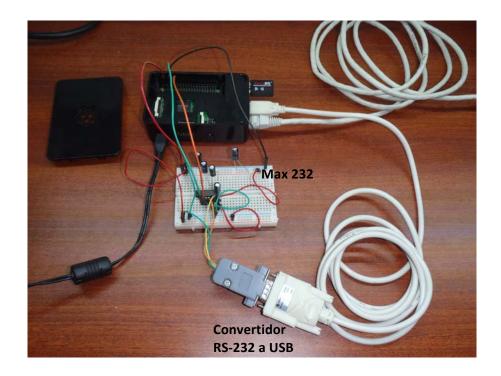
```
serial_wryte.py
python serial read.py
```

- 16. Para fines prácticos se armará un lazo cerrado (loop) con los puertos seriales de la misma Raspi, SOLO ASEGURESE CAMBIAR EL NOMBRE DEL PUERTO SERIAL EN EL RECEPTOR, sin embargo también es válido utilizar una PC o laptop con puerto serial.
 - a) Ejecute primero el código transmisor de caracteres , posteriormente el receptor y anote sus observaciones.
 - b) Ahora ejecute primero el código receptor de información, posteriormente el transmisor y anote sus observaciones.



ANEXOS

ARREGLO EXPERIMENTAL



Elaboró: Dr. Everardo Inzunza González



Elaboró: Dr. Everardo Inzunza González

También puede probar los siguientes códigos

Códiogo Python del transmisor de datos seriales

serial_write.py

```
#!/usr/bin/env python
       import time
       import serial
       ser = serial.Serial(
           port='/dev/ttyAMA0',
           baudrate = 9600,
           parity=serial.PARITY_NONE,
           stopbits=serial.STOPBITS_ONE,
           bytesize=serial.EIGHTBITS,
           timeout=1
       )
       counter=0
       while 1:
           ser.write('Write counter: %d \n'%(counter))
           time.sleep(1)
           counter += 1
```

Códiogo Python del receptor de datos seriales

serial_read.py

```
#!/usr/bin/env python

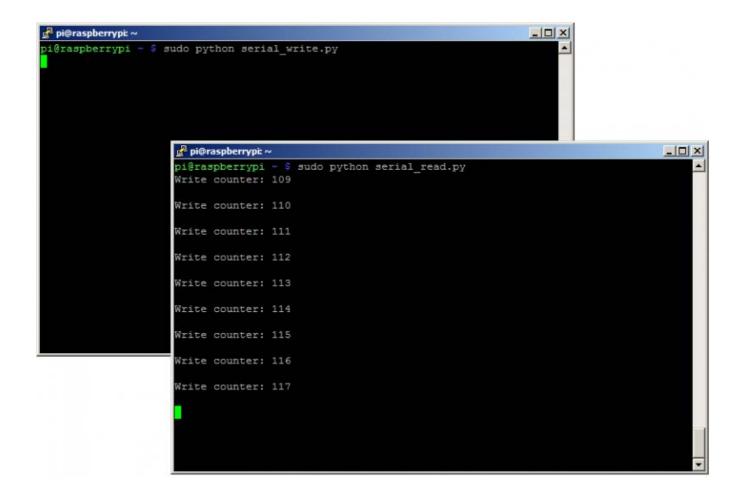
import time
import serial

ser = serial.Serial(

    port='/dev/ttyUSB0',
    baudate = 9600,
    parity=serial.PARITY_NONE,
    stopbits=serial.STOPBITS_ONE,
    bytesize=serial.EIGHTBITS,
    timeout=1
)
counter=0

while 1:
    x=ser.readline()
    print x
```

Si hacemos el loop (es decir conectamos el Tx con Rx de Raspi y ejecutamos los dos programas (transmisor y receptor), se observaría algo como la siguiente figura:



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO SISTEMAS EMPOTRADOS Práctica #8

Nombre: Comunicación Serial entre Arduido y Raspberry Pi

Objetivo: Desarrollar un programa para comunicar una Raspberry Pi con Arduino mediante puerto serial y realizar algunas consultas de componentes conectados con Arduino.

Instalación de Software Arduino IDE para la Rasberry Pi:

\$ sudo apt-get update

\$ sudo apt-get upgrade

\$ sudo apt-get dist-upgrade (Para asegurar que se tiene instalada la versión mas reciente de RPi.GPIO)

\$ sudo apt-get install arduino (Click Y en caso que pregunte sobre algunas dependencias)

Para visualizar los puertos seriales disponibles en Raspberry Pi, teclee el comando:

ls -l /dev/tty*

GPIO serial port is /dev/ttyAMA0

En linux los puertos seriales típicamente se llaman de la siguiente manera:

- Built-in (standard) Serial Port: the Linux standard is /dev/tty0, /dev/tty1, and so on.
- USB Serial Port Adapter: /dev/ttyUSB0, /dev/ttyUSB1, and so on.
 - o Some types of USB serial adapter may appear as /dev/ttyACM0 ...

También se pueden visualizar los puertos seriales asignados a sus periféricos conectados a Raspi

dmesg | grep tty

Procedimiento de la práctica:

Desde la Raspberry Pi, y mediante un código en Python, se enviarán comandos al Arduino para que nos devuelva los datos que esté leyendo.

Si por ejemplo mandamos una 'P', nos devolverá el valor del potenciómetro. Si mandamos una 'B', Arduino nos informará de si el botón está pulsado o no.

Finalmente, si mandamos una 'L', podremos leer el valor del LDR.

Paso #1

1. Armar el circuito eléctrico de la figura #1.

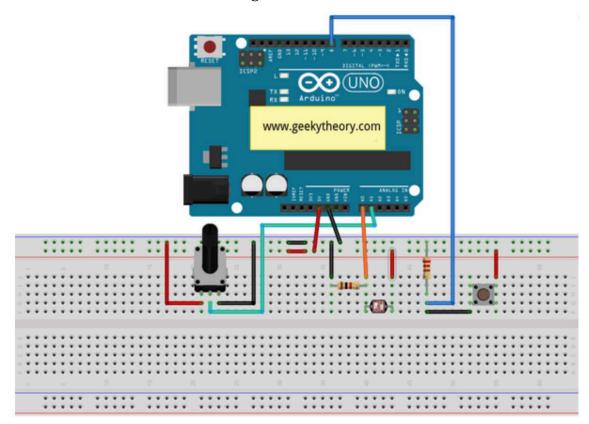


Figura 1. Diagrama esquemático de conexiones de Arduino con potenciómetro, LDR y push button.

Paso #2.

En caso de **no** tener los componente electrónicos conectados a Arduino, se tiene que cargar el siguiente código en Arduino, de lo contrario pase al paso #3.

```
void setup () {
        Serial.begin(9600); //Inicializo el puerto serial a 9600 baudios
3
  }
4
  void loop () {
        if (Serial.available()) { //Si está disponible
             char c = Serial.read(); //Guardamos la lectura en una variable char
             if (c == 'H') { //Si es una 'H', enciendo el LED
            8
             } else {
                  Serial.println("Caracter no reconocido");
14
       }
15 }
```

Figura 2. Código para Arduino sin LDR, potenciómetro, push button.

Paso #3

En caso tener los componentes electrónicos conectados a Arduino, cargar el siguiente código en Arduino.

```
int LDRPin = 0;
   int valor;
4
   int min = 0;
5
  int max = 1023;
  int boton = 8;
8
9 void setup () {
         Serial.begin(9600);
         pinMode( boton, INPUT );
12 }
14 void loop () {
         if(Serial.available()) {
               char c = Serial.read();
               if (c=='B') {
                     valor = digitalRead(boton);
                     if (valor == 0) {
20
                           Serial.println("Boton no pulsado");
                           Serial.println("Boton pulsado");
24
               } else if (c=='L') {
                     valor = analogRead(LDRPin);
                     valor = map(valor, min, max, 0, 100);
                     Serial.println(valor);
               } else if (c=='P') {
                     Serial.println(analogRead(1));
               }
         }
32 }
```

Figura 3. Código de Arduino para la comunicación serial.

En la Raspberry Pi, se tiene que instalar la biblioteca (librería) para comunicación serial:

```
1 sudo apt-get install python-serial
```

```
pi@raspberrypi: ~ - + ×

File Edit View Search Terminal Help

pi@raspberrypi ~ $ sudo apt-get install python-serial
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-serial is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

pi@raspberrypi ~ $
```

A continuación escribir el siguiente código para Python

```
import serial
2 import time
4 arduino=serial.Serial('/dev/ttyACM0',baudrate=9600, timeout = 3.0)
5
  arduino.open()
6 txt=''
8 while True:
         var = raw_input("Introducir un Comando: ")
10
        arduino.write(var)
11
         time.sleep(0.1)
         while arduino.inWaiting() > 0:
12
13
               txt += arduino.read(1)
14
               print txt
15
               txt =
16 arduino.close()
```

En este momento la Raspi espera la respuesta de Arduino, la cual dependerá del comando o caracter enviado.

Comandos válidos para realizar las pruebas de comunicación serial entre Arduino y Rasberry.

B es el comando para visualizar el estado del push button conectado a Arduino.

L es el comando para visualizar el valor del LDR en porcentaje conectado al canal A0 del ADC del Arduino.

P es el comando para visualizar el valor que entrega el potenciómetro conectado al canal A1 del ADC de Arduino.

Para ejecutar el progama de python de la Raspi

```
1 python Arduino_ReadData.py
```

Se deberán mostrar algunos resultados en el monitor de la Raspi, tal como se muestra en la siguiente figura:

```
pi@raspberrypi: ~ - + ×

File Edit View Search Terminal Help

pi@raspberrypi ~ $ python Arduino_ReadData.py
Introducir un Comando: L

Introducir un Comando: B

Boton no pulsado

Introducir un Comando: P

613

Introducir un Comando: LBP

26

Boton no pulsado

613

Introducir un Comando:
```

REFERENCIAS:

https://geekytheory.com/arduino-raspberry-pi-lectura-de-datos/

http://www.instructables.com/id/Read-and-write-from-serial-port-with-Raspberry-Pi/

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO SISTEMAS EMPOTRADOS

PRÁCTICA #9

Nombre: Procesamiento de imágenes digitales empleando Python y OpenCV.

Objetivo: Realizar operaciones básicas y de filtrado con imágenes digitales mediante Python y OpenCV.

Material:

- 1 Computadora Raspberry Pi 2 o equivalente
- 1 Monitor HDMI
- 1 Teclado y mouse
- 1 Dongle WiFi
- 1 Fuente de poder para Raspberry Pi (5V DC @ 2A)
- 1 Conexión a red ethernet (Nodo y cable de red con conectores RJ-45)
- 1 Memoria MicroSD clase 10 o similar con capacidad mínima de 8 GB

A) PROCEDIMIENTO

1. Actualizar la Raspberry.

Con el fin de poder encontrar librerías en los repositorios, es necesario tener actualizada la Raspberry por lo cual desde la terminal se ingresan los siguientes comandos:

\$sudo apt-get update

\$sudo apt-get upgrade

2. Instalar OpenCV.

Para poder trabajar con imágenes en Python, es necesario utilizar adicionalmente la herramienta OpenCV, el primer paso para poder utilizarlo es **instalar la librería** numpy:

\$sudo apt-get install python-numpy

Ahora para instalar opencv2 es necesario realizar las siguientes instrucciones:

\$sudo apt-get install python-opencv \$sudo apt-get install python-scipy \$sudo apt-get install ipython \$sudo apt-get install libgl1-mesa-dri

Finalmente se debe reiniciar la Raspberry:

Ssudo reboot

Una vez que se haya reiniciado, desde la terminal se ejecuta python: Sidle

Y aparecerá el Shell de python:

Para realizar una prueba es necesario crear un nuevo programa, seleccione File y de clic en New window, posteriormente escriba el siguiente código y guárdelo con algún nombre con extension ".py", en este caso se le dió el nombre de ejemplo.py:

```
*ejemplo.py - /home/pi/ejemplo.py*
File Edit Format Run Options Windows Help
import numpy as np #importar librerias numpy
import cv2
                  #importar libreria de opencv
img = cv2.imread('/home/pi/image.jpg') #leer imagen y guardarla en img
cv2.imshow('imagen_original',img) #mostrar imagen en ventana
print "Presione tecla e para salir o tecla s para quardar imagen"
k = cv2.waitKey(0) #funcion que espera Oms a que se presione una tecla
if k == ord('e'):
                           #si presiona tecla e se sale del programa
   cv2.destroyAllWindows() #cierra todas las ventanas abiertas
                           #entra si se presiona tecla s
elif k == ord('s'):
   cv2.imwrite('/home/pi/imagen2.jpg',img) #guardar una imagen
   cv2.destroyAllWindows() #cierra todas las ventanas abiertas
```

Elaboró: Dr. Everardo Inzunza González

Hay que poner especial atención en la línea del código:

img = cv2.imread('/home/pi/image.jpg')

en ella se indica entre apóstrofes la ruta de archivo donde se encuentra la foto original que se abrirá, la imagen usada en este ejemplo es la siguiente:



se presiona F5 para correr el programa y como resultado aparecerá la imagen principal, si se presiona la tecla "s" se guardara la foto original con el nombre de "imagen2.jpg"



image.jpg

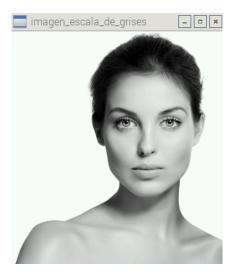


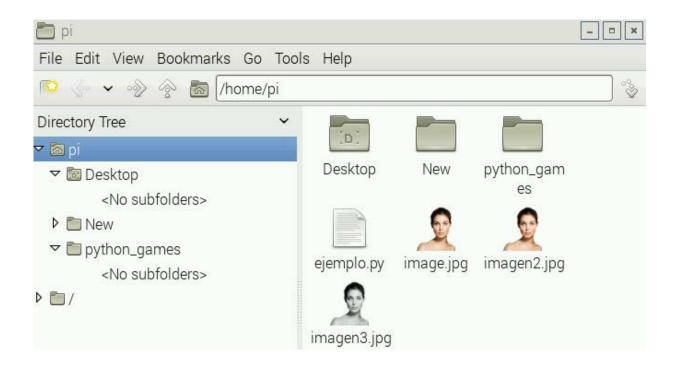
imagen2.jpg

Ahora se modificará el código para que la imagen de salida sea en escala de grises, el código es el siguiente:

```
ejemplo.py - /home/pi/ejemplo.py
<u>F</u>ile <u>E</u>dit F<u>o</u>rmat <u>R</u>un <u>O</u>ptions <u>W</u>indows <u>H</u>elp
import numpy as np #importar librerias numpy
                    #importar libreria de opencv
import cv2
img = cv2.imread('/home/pi/image.jpg',0) #leer imagen y guardarla en img
cv2.imshow('imagen_escala_de_grises',img) #mostrar imagen en ventana
print "Presione tecla e para salir o tecla s para guardar imagen"
k = cv2.waitKey(0) #funcion que espera Oms a que se presione una tecla
if k == ord('e'):
                             #si presiona tecla e se sale del programa
   cv2.destroyAllWindows() #cierra todas las ventanas abiertas
elif k == ord('s'):
                             #entra si se presiona tecla s
    cv2.imwrite('/home/pi/imagen3.jpg',img) #guardar una imagen
    cv2.destroyAllWindows() #cierra todas las ventanas abiertas
```

Y el resultado es el siguiente:





3. Obtener el histograma de una imagen mediante python y OpenCV

Para obtener el histograma de una imagen en escala de grises es necesario instalar la siguiente librería:

\$ sudo apt-get install python-matplotlib

La imagen con la que se trabajará en este segundo programa será la de cameraman.tif que es la siguiente:



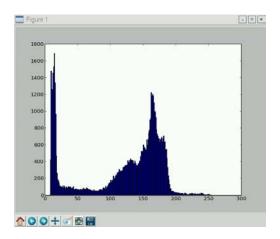
posteriormente se escribe el código en donde se obtiene el histograma de la imagen:

```
histograma.py - /home/pi/histograma.py

File Edit Format Run Options Windows Help

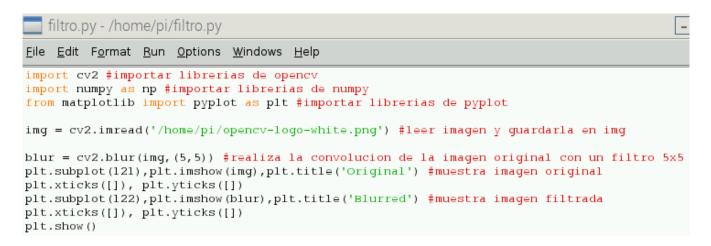
import cv2 #importar librerias de opencv
import numpy as np #importar librerias de numpy
from matplotlib import pyplot as plt #importar librerias de pyplot

img = cv2.imread('/home/pi/cameraman.tif',0) # leer imagen y guardarla en img
plt.hist(img.ravel(),256,[0,256]); #obtener histograma
plt.show() #mostrar histograma
```

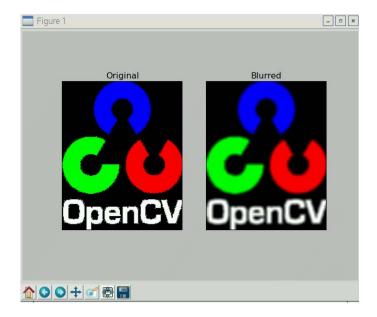


4. Realización de un proceso de filtrado

OpenCV también tiene la capacidad de realizar filtros, por ejemplo el filtro promedio, el cual es logrado mediante la convolución de la imagen con un filtro de caja normalizada. El siguiente código realiza dicho filtro:



Y el resultado es el siguiente:



5. Tarea: Investigar 5 filtros adicionales de procesamiento de imágenes y realizar un ejemplo en python de cada uno de estos filtros.

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO SISTEMAS EMPOTRADOS

PRÁCTICA #10

Nombre: Uso, manejo y configuración de la cámara CSI con Python.

Objetivo: Habilitar la cámara de la Raspberry y usarla mediante Python.

Material:

- 1 Computadora Raspberry Pi 2
- 1 Monitor HDMI
- 1 Teclado y mouse
- 1 Dongle WiFi
- 1 Fuente de poder para Raspberry Pi (5V DC @ 2A)
- 1 Conexión a red ethernet (Nodo y cable de red con conectores RJ-45)
- 1 Memoria MicroSD clase 10 o similar con capacidad mínima de 8 GB
- 1 Cámara para Raspberry Pi 5MP (con puerto CSI)
- 1 Web Cam (USB)

A) PROCEDIMIENTO

1. Actualizar la Raspberry.

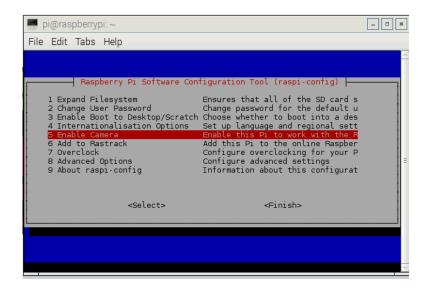
Con el fin de poder encontrar librerías en los repositorios, es necesario tener actualizada la Raspberry Pi, por lo cual desde la terminal se ingresan los siguientes comandos:

\$sudo apt-get update \$sudo apt-get upgrade

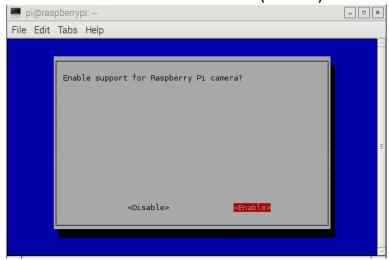
2. Habilitar la cámara:

Desde terminal ingrese el siguiente comando: \$sudo raspi-config

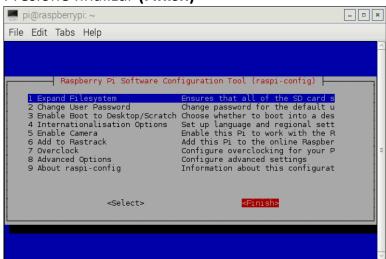
Posteriormente seleccione la opcion "Enable camera" para habilitar la cámara:



Nuevamente seleccione habilitar (Enable)



Presione finalizar (Finish)



Elaboró: Dr. Everardo Inzunza González

Y finalmente reinicie el sistema (Reboot now)

\$ sudo reboot now



3. Instalación de librerias:

Para que la cámara funcione correctamente es necesario desde Terminal instalar las librerias:

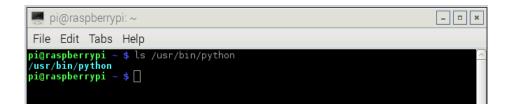
\$sudo apt-get install python-picamera

4. Comprobación de Python:

Ahora es necesario comprobar que se cuenta con Python intalado por lo se debe teclear lo siguiente:

\$ls /usr/bin/python

y debe aparecer lo siguiente lo cual indica que si se encuentra instalado:



En caso contrario se tendría que instalar mediante la siguiente instrucción: \$sudo apt-get install python

5. Instalación de editor de textos:

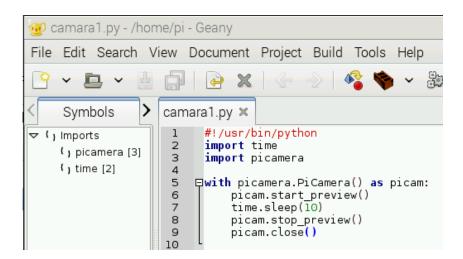
Ahora es necesario contar con un editor de textos, uno de los editores más versátiles y recomendados es geany, por lo que desde terminal se instalará tecleando lo siguiente:

\$sudo apt-get install geany

```
| Symbol | S
```

6. Primer código del uso de la cámara en Python:

A continuación un ejemplo desde python sobre el uso de la cámara. Ingrese el siguiente código desde geany:



Se guarda con extensión .py (en este caso camara1.py) para indicar que se usará con python, posteriormente se ejecuta mediante Terminal. Para ejecutar este archivo se debe introducir la siguiente instrucción:

\$ python ruta_archivo/nombre_archivo.py

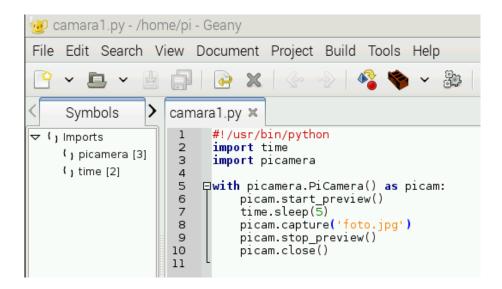
en este caso sería:

\$python camara1.py

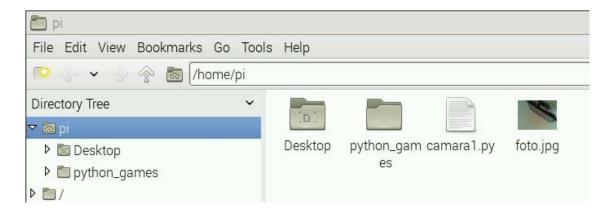
y listo, la imagen de la cámara será mostrada durante 10 segundos.

7. Tomar una foto.

Ahora se modifica un poco el programa para poder grabar la imagen tomada por la cámara.



El archivo de la foto se encontrará en la misma carpeta que el programa que lo ejecuta:



Ejercicios

8. Grabar fotos en otros formatos.

En este ejemplo se guardó una captura con el formato JPG, a continuación capture una foto con cada una de las extensiones que soporta la cámara, es decir con: **png, gif, bmp, yuv, rgb y raw**, con lo cual se comprobará que resulta bastante cómodo y útil contar con una variedad de formatos.

9. Instalación de web cam (cámara USB en Raspberry Pi)

Investigar el proceso de instalación y configuración de web cam para Raspberry, en el siguiene link se presenta una guía.

https://www.raspberrypi.org/documentation/usage/webcams/

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO SISTEMAS EMPOTRADOS

PRÁCTICA #11

Nombre: Procesamiento de señales de audio empleando Octave.

Objetivo: Procesar de señales de audio en el dominio del tiempo y frecuencia.

Material:

- 1 Computadora Raspberry Pi 2 o equivalente
- **2** Archivos de audio en formato .wav (uno con mensaje de audio en español y el otro con mensaje de audio en inglés.
- 1 Bocinas de PC
- 1 Un cable RCA para audio y video (3.5mm triple a RCA, donde RCA amarillo es para la señal de video, RCA blanco y rojo son para la señal de audio estereofónico).
- 1 Plug para convertir de RCA a 3.5 mm hembra
- 1 Monitor HDMI
- 1 Teclado y mouse
- 1 Dongle WiFi
- 1 Fuente de poder para Raspberry Pi (5V DC @ 2A)
- 1 Conexión a red ethernet (Nodo y cable de red con conectores RJ-45)
- 1 Memoria MicroSD clase 10 o similar con capacidad mínima de 8 GB

A) PROCEDIMIENTO

1. Actualizar la Raspberry.

Con el fin de poder encontrar librerías en los repositorios, es necesario tener actualizada la Raspberry por lo cual desde la terminal se ingresan los siguientes comandos:

\$sudo apt-get update \$sudo apt-get upgrade

Sauto apt-get upgrade

2. Instalar Octave

\$ sudo apt-get install octave

- 3. Verificar los toolboxes disponibles para Octave
- \$ sudo apt-cache search octave-
- 4. Instalar los toolboxes para graficado de información
- \$ sudo apt-get install octave-plplot octave-plot
- 5. Instalar los toolboxes para procesamiento de audio y de señales
 - \$ sudo apt-get install octave-audio \$ sudo apt-get install octave-signal
- 6. Finalmente se debe reiniciar la Raspberry:

Ssudo reboot

7. Una vez que se haya reiniciado, desde la terminal se **ejecuta Octave**: **\$ octave**

Y aparecerá el Shell de Octave:

```
pi@cesco ~ $ octave
GNU Octave, version 3.6.2
Copyright (C) 2012 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. For details, type `warranty'.

Octave was configured for "arm-unknown-linux-gnueabihf".

Additional information about Octave is available at http://www.octave.org.

Please contribute if you find this software useful.
For more information, visit http://www.octave.org/help-wanted.html

Read http://www.octave.org/bugs.html to learn how to submit bug reports.

For information about changes from previous versions, type `news'.

P>■
```

Figura 1. Prompt (shell) de Octave

8. Realizar el siguiente programa en el editor Geany o nano, y guardar el archivo como proc audio.m

```
clear all
close all
clc
[y,Fs] = wavread('pruebal.wav'); % Lee audio 1 en espanol
nf=1024; %number of point in DTFT
        %y = datos, Fs = tasa de muestreo
        %sound(y,Fs); para reproducir el audio
acomodo= y'; % Calcula la transpuesta
subplot(2,1,1)
plot(acomodo)
title('Senal audio en espanol');
xlabel('Tiempo');
ylabel('Amplitud de senal de audio en espanol');
subplot(2,1,2)
Y = fft(y,nf); % Calcula FFt de la senal de audio
f = Fs/2*linspace(0,1,nf/2+1); % calc eje de frecuencia
plot(f,abs(Y(1:nf/2+1))); % Grafica el espectro de frec
xlabel('Frecuencia (Hz)')
ylabel('|Y(f)| de senal')
[y,Fs] = wavread('prueba2.wav'); % Lee audio 2 ingles
nf=1024; %number of point in DTFT
        %y = datos, Fs = tasa de muestreo
        %sound(y,Fs); para reproducir el audio
acomodo= y';
figure(2)
subplot(2,1,1)
plot(acomodo)
title('Senal en ingles');
xlabel('Tiempo');
ylabel('Amplitud de senal en ingles');
subplot(2,1,2)
Y = fft(y,nf);
f = Fs/2*linspace(0,1,nf/2+1);
plot(f,abs(Y(1:nf/2+1)));
xlabel('Frecuencia (Hz)')
ylabel('|Y(f)| de senal')
```

Resultados esperados

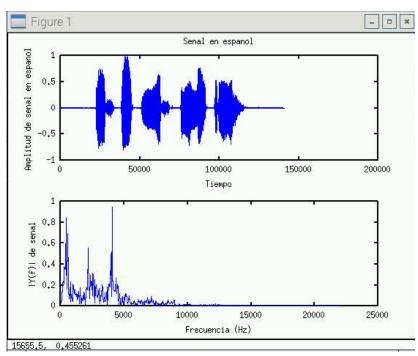


Figura 2. Señal de audio en español en el dominio del tiempo y frecuencia.

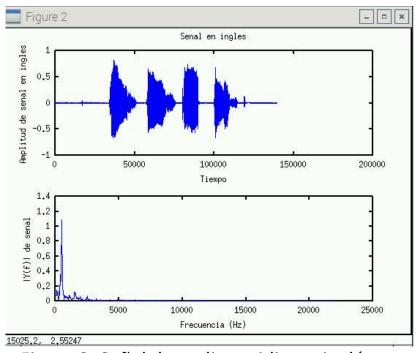


Figura 3. Señal de audio en idioma inglés mostrada en el dominio del tiempo y frecuencia.

9. Investigar e implementar un filtro pasa altas con frecuencia de corte de 2.5 KHz (Puede ser Chebyshev o Butterworth) de orden 3 o 4.

Mostrar una gráfica/figura comparativa de la señal original versus la señal filtrada.

En una figura debe estar la señal en español vs la señal español filtrada.

En la siguiente figura señal en inglés versus señal en inglés filtrada.

Describir detalladamente las observaciones de las señales filtradas.

Elaboró: Dr. Everardo Inzunza González