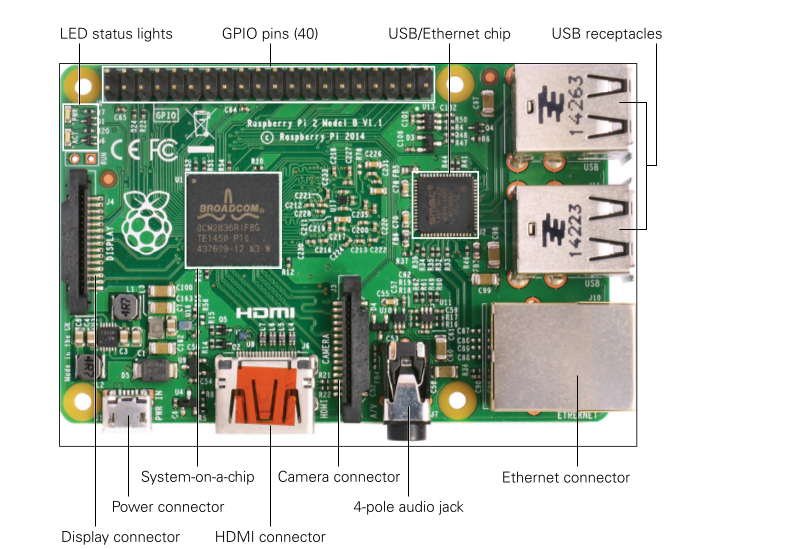
|  |  |
| --- | --- |
| PROGRAMA ACADÉMICO: **Ingeniería Informática** | |
| Módulo: **Laboratorio de Arquitectura de Hardware** | |
| Área : **Infraestructura** | Docentes: **Carlos Andrés Madrigal** |

1. **Laboratorio N. 1. Introducción a la arquitectura e instalación de una Single Board Computer Raspberry pi 3.**
2. **Objetivo de Aprendizaje**

* Conocer la arquitectura general de la single board computer Raspberry pi 3.
* Instalar el sistema operativo de la Raspberry pi 3.

1. **Fundamentos Teóricos**

La Raspberry pi es una microcomputadora que tiene como corazón un SoC(system-on-a-chip). Este SoC en un circuito integrado (IC) que tiene la gran mayoría de componentes de una computadora: (CPU) unidad de procesamiento central, (GPU) unidad de procesamiento gráfico, puertos análogos y digitales, módulos específicos de comunicación. La versión 3 de la Raspberry pi tiene como SoC un chip Broadcom BCM2837 con una CPU con 4 cores.



Los componentes más significativos del SoC Broadcom son:

* **CPU**: con 4 Cores corriendo a 1.2GHz de 64 bits, desarrolla el procesamiento de los datos bajo el control del sistema operativo.
* **GPU**: La unidad de procesamiento gráfico Broadcom [VideoCore](https://es.wikipedia.org/w/index.php?title=VideoCore&action=edit&redlink=1) IV, permite acelerar la visualización de información.
* **Memoria**: 1GB de memoria SDRAM compartida con la GPU.
* **Timers**: Permiten generar bases de tiempo periódicas para la sincronización de tareas.
* **Interrupciones**: Permite a la CPU controlar todas las fuentes de interrupción internas y externas.
* **Entradas** **y Salidas de Propósito General (GPIO):** Permite a la Raspberry controlar circuitos externos e ingresar señales digitales a través de los 40 pines externos. (GPIO Overview and the Broadcom SoC)
* **USB**: Con 4 puertos USB permite la conexión de periféricos de entrada y salida.
* **PCM/I²S**: Modulación de código de pulso, permite convertir sonido digital a análogo.
* **SPI**: Interface de comunicación serial a través de los pines del GPIO.
* **UART**: Comunicación serial asíncrona entre 2 dispositivos a través de 2 puertos UART0 y UART1.
* **PWM**: Modulación de ancho de pulso. Usado por ejemplo para el control de servomotores.

**Características Principales**

La Raspberry Pi 3 es la tercera generación y reemplaza la Raspberry Pi. Comparada con la Raspberry Pi 2, esta tiene:

* A 1.2GHz 64-bit quad-core ARMv8 CPU
* 802.11n Wireless LAN
* Bluetooth 4.1
* Bluetooth Low Energy (BLE)

Al igual que la PI 2, la Raspberry pi 3 tiene:

* 4 USB ports
* 40 GPIO pins
* Full HDMI port
* Ethernet port
* Combined 3.5mm audio jack and composite video
* Camera interface (CSI)
* Display interface (DSI)
* Micro SD card slot (now push-pull rather than push-push)
* VideoCore IV 3D graphics core

**Aplicaciones que se pueden realizar con la Raspberry Pi**

* Automatización del Hogar
* Seguridad en el hogar
* Estaciones Metereológicas
* Computador Portátil
* Controlador para Robots
* Controlador de Drones
* Servidor Web
* Servidor Email
* Rastreador GPS
* Controlador de Cámara Web
* Máquina de Café
* Controlador de Motores Eléctricos
* Administrador de Fotografías
* Controlador de Video Juegos
* Controlador A bordo de Autos
* Y Mucho más …

1. **Recursos**

Computador con acceso a internet.

Single Board Computer Raspberry pi 3

Cable MiniUSB de alimentación de 3A.

Tarjeta SD de 16GHz con el sistema operativo raspbian instalado.

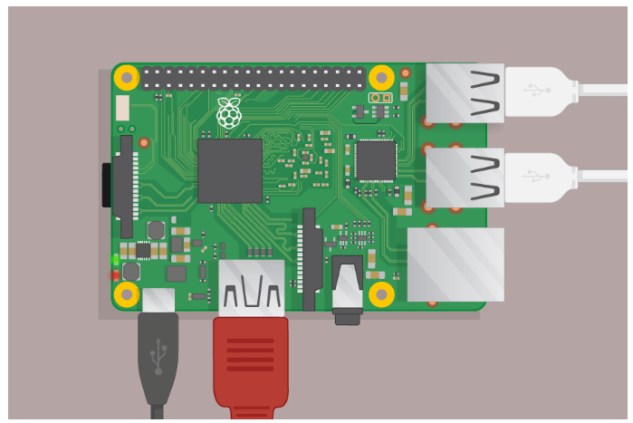
Cable conversor HDMI-VGA

1. **Guía de desarrollo**

**PROCEDIMIENTO**

**1. Conexión de la tarjeta.**

Se debe verificar que se cuenta con todo el hardware necesario para la conexión adecuada de la Raspberry. ([https://www.Raspberrypi.org/learning/hardware-guide/quickstart/](https://www.raspberrypi.org/learning/hardware-guide/quickstart/)).



**2. Instalación del sistema operativo.**

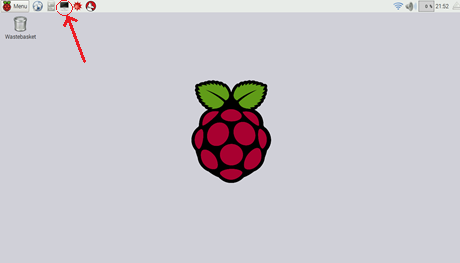
Si su SD no cuenta con el sistema operativo siga los pasos de la guía para instalar Raspbian el cual es una versión GNU/Linux diseñada específicamente para trabajar con la Raspberry Pi. [https://www.Raspberrypi.org/learning/software-guide/quickstart/](https://www.raspberrypi.org/learning/software-guide/quickstart/)

**3. Conexión a Internet.**

La conexión a internet de la Raspberry puede hacerse a través una red WiFi o a través de una conexión LAN directa. Este paso debe ser automático sin embargo si presenta problemas puede guiarse a través de [https://www.Raspberrypi.org/learning/software-guide/wifi/](https://www.raspberrypi.org/learning/software-guide/wifi/)

**4. Actualizar Sistema Operativo.**

Para actualizar las aplicaciones y software del sistema operativo se debe ingresar a la terminal y digitar



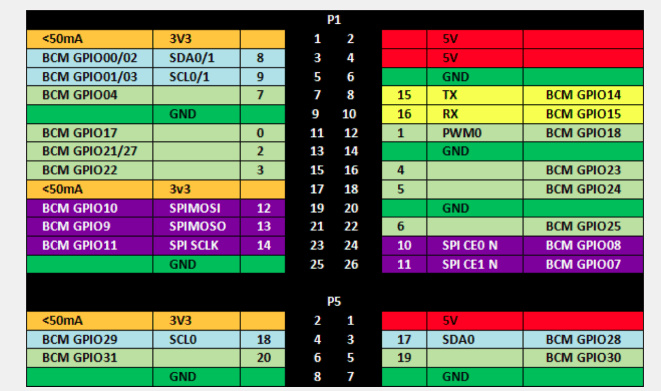
(Esta actualización debe hacerse regularmente para mantener su sistema operativo seguro y actualizado con las mejoras de cada software)

$ sudo apt-get update

$ sudo apt-get upgrade

5. Instalación de la librería para el manejo de GPIO desde Python

$ sudo apt-get install rpi.gpio



**5. Programa para encender un Led con la Raspberry Pi desde Python.**

Debemos abrir un editor de texto como nano o el shell de python.

#Este programa controla el encendido o apagado de un led por el pin 12 de acuerdo al nivel lógico de la entrada 11.

import RPi.GPIO as GPIO

# to use Raspberry Pi board pin numbers

GPIO.setmode(GPIO.BOARD)

GPIO.setwarnings(False)

# set up the GPIO channels - one input and one output

GPIO.setup(11, GPIO.IN)

GPIO.setup(12, GPIO.OUT)

# input from pin 11

input\_value = GPIO.input(11)

# output to pin 12

GPIO.output(12,input\_value )

**6. Modificación para ejecutar el programa anterior de manera infinita.**

import RPi.GPIO as GPIO

# to use Raspberry Pi board pin numbers

GPIO.setmode(GPIO.BOARD)

GPIO.setwarnings(False)

# set up the GPIO channels - one input and one output

GPIO.setup(11, GPIO.IN)

GPIO.setup(12, GPIO.OUT)

while True:

# input from pin 11

input\_value = GPIO.input(11)

# output to pin 12

GPIO.output(12,input\_value )

**7. Programa usando la numeración BCM de los pines**

import RPi.GPIO as GPIO

# to use Raspberry Pi board pin numbers

GPIO.setmode(GPIO.BCM)

GPIO.setwarnings(False)

# set up the GPIO channels - one input and one output

GPIO.setup(17, GPIO.IN)

GPIO.setup(18, GPIO.OUT)

while True:

# input from pin 17

input\_value = GPIO.input(11)

# output to pin 128

GPIO.output(18,input\_value)

**8. Programa para encender y apagar un led 10 veces**

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BCM)

GPIO.setwarnings(False)

led = 18

GPIO.setup(led, GPIO.OUT)

print ("Blinking")

for x in range(0, 9):

GPIO.output(led, 1)

time.sleep(1)

GPIO.output(led, 0)

time.sleep(1)

GPIO.cleanup()

**9. Realizar un programa que permita visualizar 4 secuencias de luces con 6 leds. Se deben controlar a partir de 2 swichs.**

1. **Entregables**

* Se deben adjuntar los programas .py en Python.

1. **Criterios de Evaluación**

|  |  |
| --- | --- |
| **Aspecto a evaluar** | **Desarrollo del laboratorio y preguntas en el desarrollo** |
| **Escala de Valoración** | **Criterios** |
| 5 | Demuestra total comprensión del problema. Todos los requerimientos de la tarea están incluidos en la desarrollo de la práctica. |
| 4 | Demuestra considerable comprensión del problema. Todos los requerimientos de la tarea están incluidos en la desarrollo de la práctica. |
| 3 | Demuestra comprensión parcial del problema. La mayor cantidad de requerimientos de la tarea están incluidos en la desarrollo de la práctica. |
| 2 | Demuestra poca comprensión del problema. Muchos de los requerimientos de la tarea faltan en el desarrollo de la práctica.. |
| 1 | No comprende el problema |
| 0 | No responde. No intentó hacer la tarea. |

|  |
| --- |
| **Observaciones:** |
|  |
|  |
|  |
|  |
|  |
|  |