

**UNIVERSIDAD TECNOLÓGICA DE**

**SAN LUIS RIO COLORADO**

**Investigación Python**

**MTRA. IRENE GARCIA**

**ALUMNO: VICTOR MANUEL GALVAN COVARRUBIAS**

**TECNOLOGÍAS DE LA INFORMACIÓN**

**ÁREA DESARROLLO DE SOFTWARE MULTIPLATAFORMA**

San Luis Rio Colorado, Sonora Marzo, 2021

1.- ¿Qué es y cómo o cuando se utiliza GPIO?

General Purpose Input Output (GPIO) es un sistema de entrada y salida de propósito general, es decir, consta de una serie de pines o conexiones que se pueden usar como entradas o salidas para múltiples propósitos generales, se pueden adaptar a las necesidades de cada proyecto a través del lenguaje de programación que se decida usar. Estos pines están incluidos en todos los modelos de Raspberry Pi, aunque con diferencias.

Siempre que vamos a empezar con algo en la Raspberry 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

Como suele ocurrir si hace un tiempo que no has hecho esto encontrara algunas cosas que os propondrá actualizar. Hazlo sin más y podemos continuar, aunque en realidad de lo que se quiere asegurar es que se tienen al día las librerías Python para acceder a los GPIO, que puedes comprobar así:

sudo apt-get install python-dev

sudo apt-get install pyton-rpi.gpio

Normalmente en los últimos tiempos vienen instaladas y configuradas de serie y si lanzas esos comandos, lo normal es que te diga que ya están instaladas en su última versión. Y eso es todo, ya podemos empezar nuestro pequeño montaje de pruebas.

2.- ¿Qué es y cómo o cuando se utiliza setwarnings?

Los mensajes de advertencia generalmente se emiten en situaciones donde es útil alertar al usuario sobre alguna condición en un programa, donde esa condición (generalmente) no justifica generar una excepción y terminar el programa. Por ejemplo, es posible que desee emitir una advertencia cuando un programa utiliza un módulo desactualizado.

Los programadores de Python emiten advertencias llamando a la función warn () definida en este módulo. (Los programadores de C usan PyErr\_WarnEx (); consulte Manejo de excepciones para obtener más detalles).

Los mensajes de advertencia normalmente se escriben en sys.stderr, pero su disposición se puede cambiar de manera flexible, desde ignorar todas las advertencias hasta convertirlas en excepciones. La disposición de las advertencias puede variar según la categoría de advertencia, el texto del mensaje de advertencia y la ubicación de origen donde se emite. Por lo general, se suprimen las repeticiones de una advertencia particular para la misma ubicación de origen.

3.- ¿Qué es y cómo o cuando se utiliza setmode?

Para especificar en su código qué sistema numérico se está utilizando, use la función GPIO.setmode (). Si está utilizando Pi Wedge, le recomendamos que utilice la definición GPIO.BCM; esos son los números serigrafiados en la PCB. El GPIO.BOARD puede ser más fácil si está conectado directamente al encabezado.

GPIO.setmode (GPIO.BCM)

... activará los números de pin específicos del chip Broadcom.

Se requieren las líneas de código de importación y setmode, si desea utilizar Python.

4.- ¿Qué es y cómo o cuando se utiliza setup?

Como hemos utilizado Arduino, probablemente estemos familiarizados con el hecho de que debe declarar un "modo pin" antes de poder usarlo como entrada o salida. Para configurar un modo pin, use la función setup ([pin], [GPIO.IN, GPIO.OUT]. Entonces, si deseamos establecer el pin 18 como salida, por ejemplo, escribimos:

GPIO.setup (18, GPIO.OUT)

Recordemos que el número de PIN cambiará si está utilizando el sistema de numeración de la placa (en lugar de 18, sería 12).

5.- ¿Qué es y cómo o cuando se utiliza Try?

El bloque try le permite probar un bloque de código en busca de errores.

El bloque try generará una excepción, porque x no está definido:

try:

print(x)

except:

print("An exception occurred")

6.- ¿Qué es y cómo o cuando se utiliza except?

El bloque except le permite manejar el error.

try:

print(x)

except NameError:

print("Variable x is not defined")

except:

print("Something else went wrong")

7.- ¿Qué es y cómo o cuando se utiliza whileTrue?

Con el ciclo whileTrue podemos ejecutar un conjunto de declaraciones siempre que una condición sea verdadera.

i = 1

if i < 5:

True

else:

False

while True:

print(i)

i += 1

8.- ¿Qué es y cómo o cuando se utiliza time?

Para Retrasos. Si necesita ralentizar su secuencia de comandos de Python, puede agregar retrasos. Para incorporar retrasos en su secuencia de comandos, deberá incluir otro módulo: time. Esta línea, en la parte superior de su script, lo hará por usted:

Include time

Luego, durante el resto del código, puede usar time.sleep ([seconds]) para darle un descanso al código. Puede utilizar decimales para establecer con precisión su retraso. Por ejemplo, para retrasar 250 milisegundos, escriba:

time.sleep(0.25)

El módulo de time incluye todo tipo de funciones útiles, además de sleep.

9.- ¿Qué es y cómo o cuando se utiliza for?

Un bucle for se utiliza para iterar sobre una secuencia (que es una lista, una tupla, un diccionario, un conjunto o una cadena).

Esto es menos parecido a la palabra clave for en otros lenguajes de programación y funciona más como un método iterador que se encuentra en otros lenguajes de programación orientados a objetos.

Con el ciclo for podemos ejecutar un conjunto de declaraciones, una vez para cada elemento de una lista, tupla, conjunto, etc.

fruits = ["apple", "banana", "cherry"]

for x in fruits:

print(x)

10.- ¿Qué es y cómo o cuando se utiliza cleanup?

RPi.GPIO proporciona una función integrada GPIO.cleanup () para limpiar todos los puertos que ha utilizado. Pero tenga muy claro lo que hace esto. Solo afecta a los puertos que haya configurado en el programa actual. Restablece los puertos que ha utilizado en este programa de nuevo al modo de entrada. Esto evita daños por, digamos, una situación en la que tiene un puerto configurado HIGH como salida y lo conecta accidentalmente a GND (LOW), lo que provocaría un cortocircuito en el puerto y posiblemente lo freiría. Las entradas pueden manejar 0 V (BAJO) o 3,3 V (ALTO), por lo que es más seguro dejar los puertos como entradas.

import RPi.GPIO as GPIO

GPIO.cleanup()