|  |  |
| --- | --- |
| C:\Users\SebSan\Pictures\unach.jpg**UNIVERSIDAD NACIONAL DE CHIMBORAZO****FACULTAD DE INGENIERIA** |  |
| **GUÍA DE PRÁCTICAS** **PERIODO ACADÉMICO: 2025 1S** | **VERSIÓN:** 1 |
| **Página 1 de 2** |
| **CARRERA: TELECOMUNICACIONES** | **DOCENTE:****LEONARDO RENTERIA** | **SEMESTRE: QUINTO****PARALELO: A** |
| **NOMBRE DE LA ASIGNATURA:**SISTEMAS EMBEBIDOS. | **CÓDIGO DE LA ASIGNATURA:**TEP120355 | **LABORATORIO A UTILIZAR:**ELECTRONICA |
| **Práctica No.:****9** | **Tema:**APLICACIONES IOT  | Duración (horas)4 | No. Grupos | No. Estudiantes (por Grupo)4 |
| **Objetivos de la Práctica:**Aprender a manipular los periféricos GPIO y ADC del dispositivo a través de internet. |
| **Equipos, Materiales e Insumos:*** **Pc**
* **Internet**
* **Esp32**
* **Leds**
* **Resistencias**
* **Botones**
* **Cables**
* **Protoboard**
* **Potenciómetro**

 |
| **Procedimiento:**1. Abrir el terminal de Micropython y ejecutar los siguientes comandos y describir los resultados.

import networkwlan = network.WLAN(network.STA\_IF)wlan.active(*True*)wlan.isconnected()1. Ejecute el siguiente comando y describa los resultados

wlan.scan()1. Ejecute el siguiente comando y describa el resultado

wlan.connect('nombre de la red', 'clave si la hay')sleep(5000)wlan.isconnected()1. Ejecute el siguiente comando y describa el resultado

wlan.ifconfig() 1. Presione el botón Stop y escriba el siguiente código en un Script, ejecútelo y describa el resultado.

import networkimport timewlan = network.WLAN(network.STA\_IF)wlan.active(*True*)*print*(wlan.scan())i=1if not wlan.isconnected(): *print*('conectando a la red wifi...') wlan.connect('Nombre de la res, 'Clave') while not wlan.isconnected(): *print*("\*"\*i) time.sleep(1) i=i+1*print*("Conectado")*print*('Configuracion de la red:', wlan.ifconfig())1. Crear un cliente MQTT y suscribirse al tópico “miTopico”.
2. Crear un script con el siguiente código y ejecutarlo (debe estar conectado a la red inalámbrica primero). Describa el resultado.

import umqtt.simplecliente=umqtt.simple.MQTTClient('id unico','direccion\_del\_Broker',1883,'usuario\_si\_lo\_ hubiere','clave\_si\_la\_hubiere')cliente.connect()for i in *range*(10): cliente.publish("miTopico",*str*(i)) time.sleep(1)*print*("fin")1. Crear un script con el siguiente código y ejecutarlo (debe estar conectado a la red inalámbrica primero).

import umqtt.simpleimport timecliente=umqtt.simple.MQTTClient('id unico','direccion\_del\_Broker',1883,'usuario\_si\_lo\_ hubiere','clave\_si\_la\_hubiere')*def* mensaje(topico, mensaje): *print*(mensaje)cliente.connect()cliente.set\_callback(mensaje)cliente.subscribe('miTopico')i=0while (1): *print*(i) cliente.check\_msg() i=i+1 time.sleep(1)1. Crear un cliente MQTT y publicar un texto con el tópico “miTopico”. Describa el resultado.
2. Cambiar la línea cliente.check\_msg()por cliente.wait\_msg() .
3. Repita el paso anterior y describa el resultado.
4. Arme el circuito de la figura

1. Crear un script en microPython para encender y apagar el Led de forma remota usando un cliente MQTT.
2. Crear un script en microPython para enviar el valor del potenciómetro a un cliente MQTT cada 2s.
3. Crear una aplicación Android que permita visualizar de forma remota el valor del potenciómetro en una gráfica y controlar es estado de LED. El valor del potenciómetro se envía cada 3s y se almacena cada segundo en un archivo de texto con la hora en el formato: “hh:mm:ss;valPot”. Si el valor del potenciómetro supera determinado valor, se debe lanzar una notificación de alerta en la aplicación (aunque esta no esté abierta)
4. Implementar un sistema plug-and-play (conecta y funciona) para medir el nivel de radiación en una escala de acuerdo con la siguiente gráfica y mida los valores de temperatura y humedad ambiental.

El indicador debe permitir visualizar los niveles de radiación a una distancia mínima de 5 metros.Adicionalmente, el sistema envía una alerta a una aplicación Android cuando cambia a nivel moderado y superiores con el fin de advertir y sugerir medidas de protección. Ante este cambio, se debe emitir una notificación, aunque la **aplicación no se esté ejecutando en primer plano**.Nota: El literal 16 equivale al proyecto final correspondiente a la investigación formativa el cual debe ser presentado en el formato correspondiente.  |
| **Resultados:**Aprende a manipular los módulos GPIO y ADC del dispositivo a través de internet |
| **Anexos:**  |
| **Referencias bibliográficas:**[**https://docs.micropython.org/en/latest/esp32/tutorial/intro.html**](https://docs.micropython.org/en/latest/esp32/tutorial/intro.html)[**https://appinventor.mit.edu/**](https://appinventor.mit.edu/)[**https://ullisroboterseite.de/android-AI2-MQTT-en.html**](https://ullisroboterseite.de/android-AI2-MQTT-en.html) |

**Fecha de Revisión y Aprobación**: 01/04/2025

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Firma Director de Carrera Firma Docente**