Carrera: INGENIERIA EN INFORMATICA

Asignatura: 3631-Fundamentos de sistemas embebidos.

Tema: Máquinas de Estado Micropython

Unidad: 5.1

Objetivo: Implementar una FSM en Micropython

Competencia/s a desarrollar:

- Concepción, diseño y desarrollo de proyectos de ingeniería en sistemas de información / informática.
- Gestión, planificación, ejecución y control de proyectos de ingeniería en sistemas de información / informática.
- Utilización de técnicas y herramientas de aplicación en la ingeniería en sistemas de información / informática.
- Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.
- Desarrollo de una actitud profesional emprendedora.
- Aprendizaje continuo.
- Actuación profesional ética y responsable.
- Comunicación efectiva.
- Desempeño en equipos de trabajo.
- Identificación, formulación y resolución de problemas de ingeniería en sistemas de información/informática.

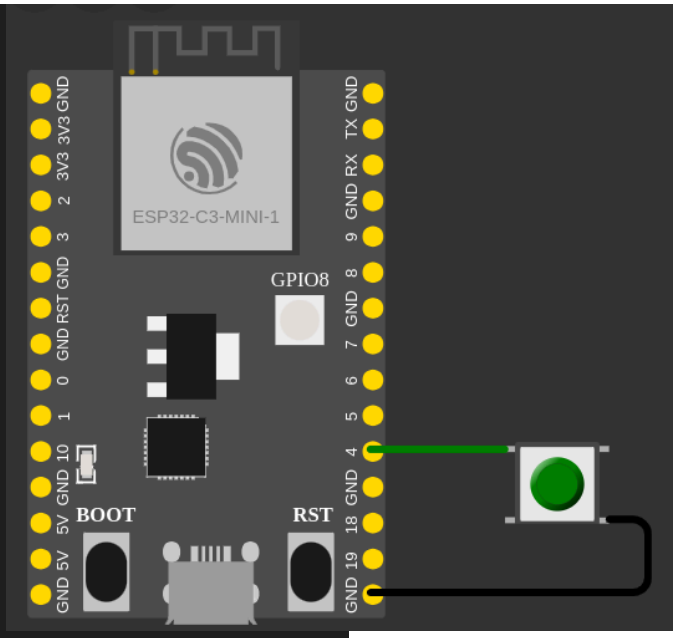
Descripción de la actividad:

1-Tiempo estimado de resolución: una clase
2-Metodología: Práctica de laboratorio
3-Forma de entrega: No obligatoria
4:Metodología de corrección y feedback al alumno: Presencial

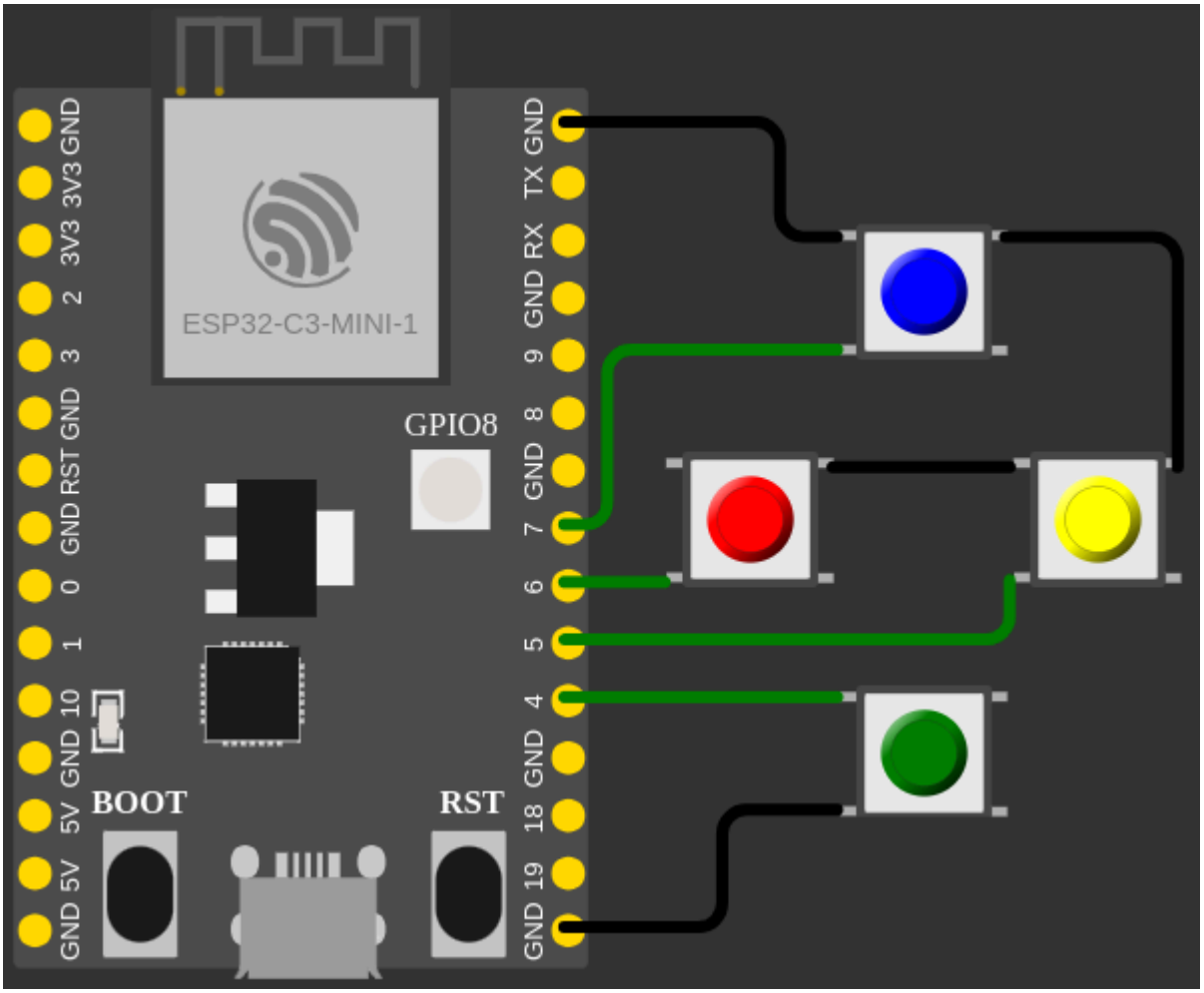
A- Bouncing

A.01 *Implemente el siguiente circuito con un pulsador con BOUNCING activo y compruebe que detecta correctamente un botón pulsado. Dibuje el diagrama de transición de estados. Los estados son NADA, EVENTO y ESPERA. La entrada es V o F.*

```python
import machine
from machine import Pin, Timer
import time

#Definimos tipos
NADA = 0
EVENTO = 1
ESPERA = 2

Verde= Pin(4, Pin.IN, Pin.PULL_UP)

estadoVerde = NADA #Estado del pulsador

def procesaEntrada(estadoActual,valorPin):
    if estadoActual==NADA and valorPin==False:
        return EVENTO
    if estadoActual==EVENTO:
        return ESPERA
    if estadoActual==ESPERA and valorPin==False:
        return ESPERA
    return NADA

while True:
    time.sleep_ms(5) #Dejamos pasar tiempo para eliminiar bouncing
    #Procesamos la entrada a ver si hay evento
    estadoVerde = procesaEntrada( estadoVerde, Verde.value())
    if (estadoVerde == EVENTO):
        print("Verde!")

```



A.02 *Expanda el diseño anterior para que soporte 4 botones (Verde-Abajo Pin4 , Rojo-Izquierda Pin6, Azul-Arriba Pin 7 , Amarillo-Derecha Pin5). Cada botón presionado debe generar un print indicando el color.*



A.03 *Ejecute el siguiente ejemplo (Copy/Paste) que detecta la secuencia: Arriba, Derecha, Abajo e Izquierda.*

```
import machine
from machine import Pin, Timer
import time
#Definimos tipos
NADA = 0
EVENTO = 1
ESPERA = 2
```

```
ARRIBA =1
ABAJO = 2
DERECHA = 3
IZQUIERDA = 4

Verde= Pin(4, Pin.IN, Pin.PULL_UP)
Rojo=  Pin(6, Pin.IN, Pin.PULL_UP)
Azul=  Pin(7, Pin.IN, Pin.PULL_UP)
Amarillo= Pin(5, Pin.IN, Pin.PULL_UP)

estadoVerde = NADA #Estado del pulsador
estadoRojo = NADA #Estado del pulsador
estadoAzul = NADA #Estado del pulsador
estadoAmarillo = NADA #Estado del pulsador

estadoFSM=0

def FSM(estadoActual,entrada):
   #Esta FSM recibe el estado actual y una entrada
   #que puede ser 1 (arriba), 2(abajo), 3 (izq), 4 (der)
   #y reconoce la secuencia Arrriba, Der, Abajo, Izq
   if estadoActual==0 and entrada==ARRIBA:
       return 1
   if estadoActual==1 and entrada==ARRIBA:
       return 1 #caso de dos arriba seguidos
   if estadoActual==1 and entrada==DERECHA:
       return 2
   if estadoActual==2 and entrada==ABAJO:
       return 3
   if estadoActual==3 and entrada==IZQUIERDA:
       return 4
   return 0
```

```
def procesaEntrada(estadoActual,valorPin):
    if estadoActual==NADA and valorPin==False:
        return EVENTO
    if estadoActual==EVENTO:
        return ESPERA
    if estadoActual==ESPERA and valorPin==False:
        return ESPERA
    return NADA

while True:
    time.sleep_ms(5) #Dejamos pasar tiempo para eliminiar bouncing
    #Procesamos la entrada a ver si hay evento
    estadoVerde = procesaEntrada( estadoVerde, Verde.value())
    if (estadoVerde == EVENTO):
        estadoFSM=FSM(estadoFSM,ABAJO)

    estadoAzul = procesaEntrada( estadoAzul, Azul.value())
    if (estadoAzul == EVENTO):
        estadoFSM=FSM(estadoFSM,ARRIBA)

    estadoRojo = procesaEntrada( estadoRojo, Rojo.value())
    if (estadoRojo == EVENTO):
        estadoFSM=FSM(estadoFSM,IZQUIERDA)

    estadoAmarillo = procesaEntrada( estadoAmarillo, Amarillo.value())
    if (estadoAmarillo == EVENTO):
        estadoFSM=FSM(estadoFSM,DERECHA)

    if (estadoFSM==4):
        print("Konami")
        estadoFSM=0
```

A.04 *Modifique el ejemplo para que detecte la secuencia correcta (Arriba Arriba, Abajo Abajo, Izquierda Derecha Izquierda Derecha). Escriba el diagrama de transición de estados donde las entradas son ARRIBA, ABAJO, DERECHA e IZQUIERDA y los estados van de 0 a N.*