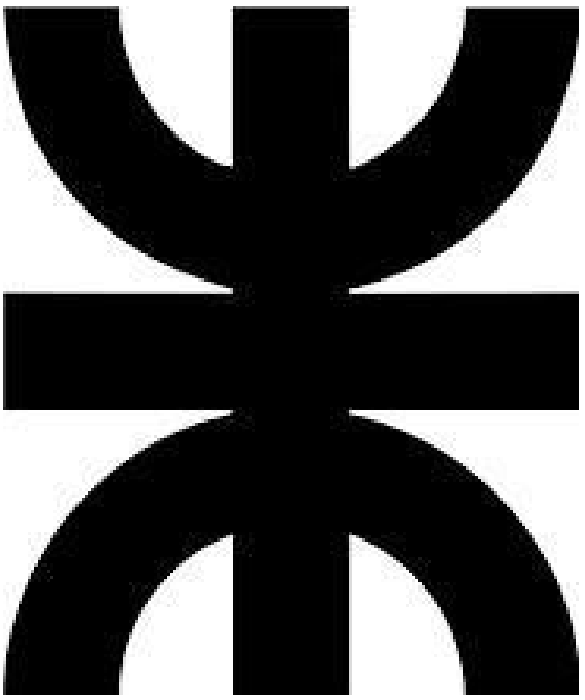
UNIVERSIDAD TECNOLÓGICA NACIONAL



LABORATORIO N°1: MICROCONTROLADORES

Tecnologías para la Automatización

DOCENTES:

- **Ing. Arturo Castaño.**
- **Ing. Dominga C. Aquino.**
- **Aux. Carlos G. Pérez.**

INTEGRANTES:

- **Gomez, Marcelo.**
- **Gomez, Christian.**
- **Medina, Mariano.**
- **Tourn, Miguel.**

GRUPO “Los Ogata”

AÑO 2024

Escenario

Suponga que usted tiene un cine en casa y desea automatizar algunas funciones del mismo, para eso, se plantea la necesidad de un **controlador** tanto de volumen como de luces de la sala.
Al estar siendo reproducida la película, las voces de los espectadores deben estar calladas, por lo que no debería haber ruido de voces y las luces deberían estar **apagadas**, y el volumen se mantiene en un nivel **normal**, que es el volumen con el cual inicia una función de cine.
Los decibeles de las voces de los espectadores van fluctuando en ciertos rangos de decibeles, nunca van a quedar en un mismo rango, ya que el volumen de voces va subiendo y bajando respecto al tiempo que pasa.
El controlador realiza mediciones de los decibeles de las voces de manera constante y tomando decisiones respecto del volumen de la película. Cabe destacar que la película originalmente inicia en un volumen de nivel normal.
Si el volumen está en un **nivel normal,** las decisiones a tomar son las siguientes.

- Si las voces permanecen en un rango por debajo del volumen normal de la película, se **mantiene** el volumen en el filme.
- Si las voces permanecen durante **siete segundos seguidos** en un rango por encima del volumen normal de la película, se **sube** el volumen del filme a un volumen **alto**.
- Si las voces permanecen **cuatro segundos seguidos** en un rango por encima de el volumen alto de la película, se **sube** el volumen a un volumen **superior**.

Si el volumen se encuentra en un **nivel alto**, las decisiones a tomar serán las siguientes.

- Si las voces se mantienen en un rango por debajo del volumen actual durante **un segundo,** se **disminuye** el volumen del filme a un rango anterior.
- Si las voces permanecen en un mismo rango que el del volumen actual, el volumen se **mantiene** en el nivel actual.
- Si las voces permanecen **siete segundos seguidos** un rango más alto que el volumen actual de la película, se **sube** el volumen a un volumen **superior**.

Si el volumen se encuentra en un **nivel superior**, las decisiones a tomar serán las siguientes.

- Si las voces se mantienen en un rango por debajo del volumen actual, y permanecen **un segundo** en este rango, se **disminuye** el volumen del filme a un rango anterior.
- Si las voces permanecen la mayor parte del tiempo en un rango igual al del volumen actual de la película, se **mantiene** el volumen del filme.
- Si las voces permanecen durante **siete segundos seguidos** en un rango más alto que el volumen actual de la película, esto significa que a los espectadores no les importa para nada la película, por lo que se **pausa la película y se encienden las luces de la sala.**

Tabla de valores para el volumen.

Rango	Normal	Alto	Superior
Mínimo (dB)	70	81	91
Máximo (dB)	80	90	98

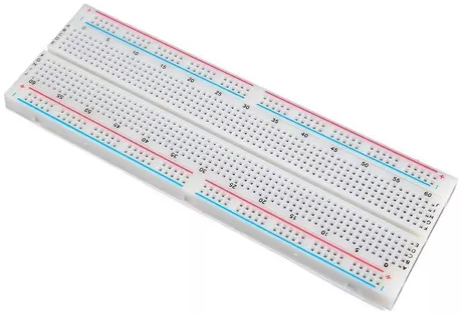
La otra función del controlador es la de manejar las luces de la sala, esto lo hace según las entradas de las personas a la sala de cine. Si se detecta que **ingresa alguien**, se debe **pausar** la película, **encender** las luces y darle **diez segundos** al espectador para que se acomode, para así luego reiniciar la película y apagar las luces. Cabe destacar que la puerta de entrada la sala, **tiene un pequeño foco, cuando una persona entra, este foco se enciende,** de esta manera el sensor detecta el encendido de esta luz.
Siempre que la película se pausa, las mediciones de volumen se cancelan, y una vez que se vuelve a iniciar el filme, las mediciones vuelven a ser tomadas.

Referencias:

Nivel	Valor en display
Normal	3
Alto	6
Superior	8

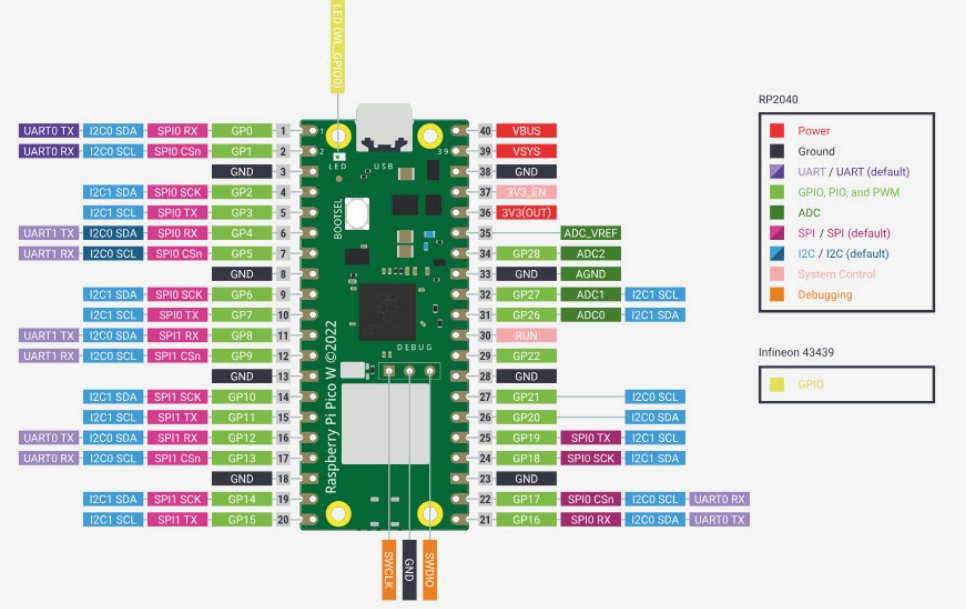
Elementos a utilizar

- **Placa Protoboard**
  La Protoboard, llamada en inglés breadboard, es una placa de pruebas en los que se pueden insertar elementos electrónicos y cables con los que se arman circuitos sin la necesidad de soldar ninguno de los componentes. Las Protoboards tienen orificios conectados entre sí por medio de pequeñas láminas metálicas.

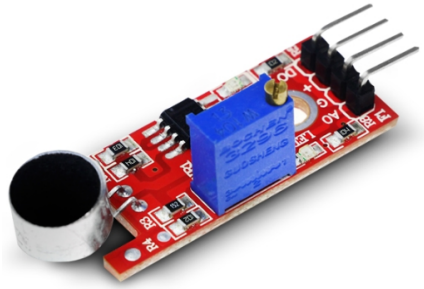


- **Microcontrolador Raspberry Pi Pico W**
  Es una placa de desarrollo que cuenta con un microcontrolador diseñado por Raspberry Pi el **RP2040**. Utiliza un puerto micro usb para alimentación y datos. Además de alimentarlo por el
  puerto micro usb 5v o por fuente o batería externa entre 1,8 y 5,5 voltios, hay que

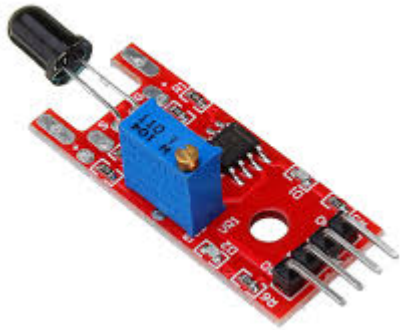
tener en cuenta que los puertos GPIO y el funcionamiento general es a 3,3v es decir cualquier sensor y actuador que conectemos ha de ser de este voltaje , aunque se alimente por el usb o por una alimentación externa existe un regulador que baja la tensión a 3,3v.
- Cuenta con un encapsulado de 40 pines. De esos 30 son multi-función que podremos usar como GPIO (Entrada/salida de propósito general)a 3,3v.
- Existe la posibilidad de conectarle una SD card a través de SPI.
- Es importante destacar que puede emular interfaces tales como SD Card y VGA.



- **Sensor de sonido KY-037**
  Este módulo está compuesto por: un Micrófono sensible KY-037, dos salidas: una llamada "AO", que es una salida analógica, señal de tensión de salida en tiempo real de micrófono y otra llamada "Do", que es una señal digital que se activa cuando la intensidad del sonido a alcanzado un cierto umbral previamente configurado. El grupo utiliza este sensor con el fin de captar el ruido en la sala de cine y en base a esto tomar decisiones.

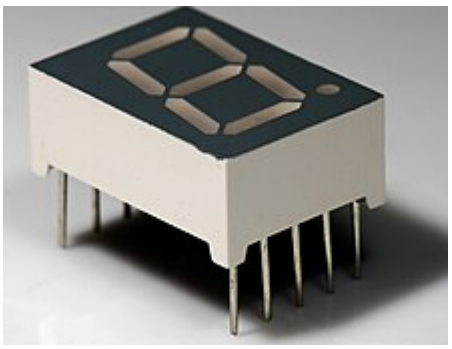


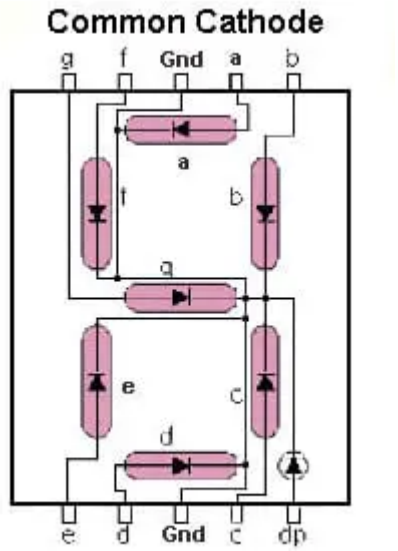
- **Sensor infrarrojo KY-026**
  Este sensor tiene la capacidad de detectar llamas o directamente fuego, posee un ángulo de detección de 60° y es capaz de detectar llamas ya que posee una estructura LED detectora de fuego. El grupo lo utiliza con el fin de detectar la presencia de una luz, lo que significa que la puerta de la sala de cine ha sido abierta.



- **LED**
  El grupo lo utiliza con el fin de simular las luces de la sala de cine.

- **Display de 7 segmentos**
  Es una forma de representar caracteres en equipos electrónicos. Está compuesto de siete segmentos que se pueden encender o apagar individualmente. Cada segmento tiene la forma de una pequeña línea. Se podría comparar a escribir números con cerillas o fósforos de madera. El grupo lo utiliza para simular el volumen de la pantalla en donde se proyecta la película.





- **Cables**

- **Resistencias de 330 OHM**

Aclaraciones

- **Conversión de valores analógicos a decibeles**
  Como el sensor KY-037 mide solamente diferencias de voltajes, y esta diferencia es traducida a un valor analógico por el microcontrolador, por lo que primero se deben traducir estos valores analógicos a voltaje, esto se lo hace de la siguiente manera:

$$V = \frac{valorAnalogico * 3.3V}{65535}$$

El número 65535, es el máximo valor analógico que puede leer el microcontrolador, que se puede formar con 16 bits, ya que es la capacidad de lectura del microcontrolador.

Una vez que tenemos el voltaje, es momento de pasarlo a decibeles (dB). Esta conversión se logró mediante la aplicación de una función lineal, esta misma fue obtenida gracias a que el grupo realizó experimentaciones y en base a estas, fue aplicada la regresión lineal que nos permitió obtener la función lineal para realizar la conversión.

IMPORTANTE: LAS EXPERIMENTACIONES FUERON REALIZADAS EN UNA HABITACIÓN EN COMPLETO SILENCIO, SIN UN MÍNIMO BULLICIO.

Link a experimentación

- **Calibración**
  Para el correcto funcionamiento del código, ambos sensores deben estar calibrados, esta calibración se logra, girando en sentido horario para una mayor sensibilidad o sentido antihorario para una menor sensibilidad. Por lo que al ejecutar el código, no va a funcionar por completo hasta que estén calibrados ambos sensores, por lo que existe una función la cual se encarga de tomar valores con ambos sensores y ver si estos están calibrados.

  El sensor KY-037, estará calibrado cuando los valores que reciba sean de entre 33200 y 34500 (valores analógicos), convirtiendo a voltaje son 1.68-1.75 V, convirtiendo a decibeles son 68-75 dB, lo que significa que el volumen de la pantalla de cine está funcionando pero el público está en silencio. Cabe destacar que estos **rangos de calibración fueron medidos en una espacio físico en completo silencio**, al probar la ejecución de la simulación en otro espacio físico donde exista un mínimo bullicio, la calibración del sensor cambia, por lo que se recomienda ejecutar la simulación en una habitación en completo silencio.

  El sensor KY-026 estará calibrado cuando el voltaje sea mayor a 3,00 V, lo que nos indica que no está detectando ninguna luz.

Código

```
import uasyncio as asyncio  # Reemplazar por `asyncio` si utiliza CircuitPython
# Reemplazar por `circuit_monitoring` si utiliza CircuitPython
import micro_monitoring

import machine as m
import time as t
import math
import network  #pt2



async def operations():

    global ant
    global prom
    global segcorridos, segcorridos2
    global actual
    global convertir_dB
    global num_3, num_6, num_8, num_0, num_nada
    global normal, alto, superior, normal_1, alto_1, superior_1, mediciones
    global activado
    # Definición de pines
    luz_sala = m.Pin(20, m.Pin.OUT)  # Luz LED que representa la luz de la sala.
    foco_puerta = m.ADC(m.Pin(27))  # Sensor de luz para captar la luz del foco al abrir la puerta.
    sonido = m.ADC(m.Pin(28))      # Sensor de sonido.

    # Pines para los segmentos del display (a, b, c, d, e, f, g)
    display_pins = (19, 18, 13, 15, 14, 16, 17)
    segments = [m.Pin(pin, m.Pin.OUT) for pin in display_pins]

    # Configuraciones para los números
```

```
    #b a c e d f g <- orden del arreglo
    num_3 = [1, 1, 1, 0, 1, 0, 1]
    num_6 = [0, 1, 1, 1, 1, 1, 1]
    num_8 = [1, 1, 1, 1, 1, 1, 1]
    num_0 = [1, 1, 1, 1, 1, 1, 0]
    num_nada = [0, 0, 0, 0, 0, 0, 0]  # Nada encendido.
        ## funciones y procedimientos ##
    def mostrar_vol(number):
        for i, segment in enumerate(segments):
            segment.value(number[i])

    async def pausar_todo():
        global ant, actual, activado, prom, convertir_dB
        await asyncio.sleep (0.7)
        print("Entro alguien a la sala")
        ant = actual [0]
        activado = True
        convertir_dB (prom)
        luz_sala.on()
        total_segundos = 10
        await asyncio.sleep (0.7)
        print("Pelicula pausada, se reanudará en 10 segundos.")
        await asyncio.sleep (0.7)
        for segundos_restantes in range(total_segundos, 0, -1):
            print(f"Reanudando en {segundos_restantes} segundos...", end="\r")
            t.sleep(1)

        luz_sala.off()
        print("\n¡La película se está reanudando!")
        await asyncio.sleep (2)
```

```
async def ir_avol(volumen, d,promedio,vol_act, segundos):
    vol = volumen
    global prom
    global segcorridos, segcorridos2
    global ant
    global normal, alto, superior, normal_1, alto_1, superior_1
    segcorridos = 0
    segcorridos2 = 0
    t.sleep (3)
    if actual [0] == 3 :
        actual1 = normal_2

    elif actual [0] == 6:
        actual1 = alto_2

    else:
        actual1 = superior_2
    print (f"Se detectó un constante ruido durante {segundos} segundos seguidos fuera del rango de [{actual1[1]} dB - {actual1[2]} dB] ")
    t.sleep (3)
    print (f"Este ruido tiene un valor de: {convertir_dB (promedio)} dB \nSe toma la siguiente decision:")
    t.sleep (2)
    activado = False
    if d == "baja":
        print(f"Bajar el volumen a: {vol}")
    else:
        print(f"Subir el volumen a: {vol}")

    if volumen == 3:
        mostrar_vol(num_3)
```

```
            ant = 3
            await asyncio.sleep (1.5)
            return normal
        elif volumen == 6:
            mostrar_vol(num_6)
            ant = 6
            await asyncio.sleep (1.5)
            return alto
        else:
            mostrar_vol(num_8)
            ant = 8
            await asyncio.sleep (1.5)
            return superior



    def fin_pelicula():
        t.sleep (1.5)
        print("A los espectadores no les importa la película parece...")
        t.sleep(1)
        print("Prendemos las luces")
        t.sleep(2)
        luz_sala.on()
        t.sleep(2)
        print("Apagamos la tele")
        t.sleep(2)
        mostrar_vol(num_nada)
        return True



    def calibrar (info,sensor):
        calibracion = []
```

```
    for i in range (0,20) :  #para observar la calibracion
        if info == 1:
            sal = convertir_dB (sonido.read_u16())
            #print (sonido.read_u16())
            print (f"Lectura numero {i+1}= {sal} dB")



        if sensor == 1:
            calibracion.append(sonido.read_u16())
        else:
            calibracion.append (foco_puerta.read_u16())

    return (sum (calibracion) // len (calibracion))

def convertir_dB (volt):
    ent = convertir_volt (volt)
    result = (120.24 * (ent) - 132.86)  ## experimentacion con regresion lineal
    return (round (result , 1))

def convertir_volt (analogico) :
    return ((analogico *3.3) / 65535)

def fin_normal ():
    print ("Prendemos las luces")
    t.sleep(2)
    luz_sala.on()
    t.sleep(2)
    print ("Apagamos la television")
    t.sleep(2)
    mostrar_vol (num_nada)
```

```
##############################fin funciones y procedimientos ##################################

#volumen actual, valor bajo del rango, valor alto del rango
normal = [3,33400,34500]
alto = [6,34501,36800]
superior = [8,36801,38400]

#rta = int (input ("Seleccione opcion \n1-Habitacion en silencio\n2-Biblioteca de la facu con un minimo bullicio\n>"))



normal = [3,33400,34500]
alto = [6,34501,36800]
superior = [8,36801,38400]
"""if rta == 2 :
    normal = [3,36200,37000]
    alto = [6,37001,38800]
    superior = [8,38801,4000]"""
normal_2 = [3,70,80]
alto_2 = [6,81,90]
superior_2 = [8,91,98]

mediciones = []

# Inicio de la simulación
luz_sala.on()
print("Bienvenidos al CineOgata")

t.sleep(1)
mostrar_vol(num_0)
```

```
    print ("Primero veremos si el sensor de infrarrojo esta calibrado...")
    luz= calibrar (0,2)

    while luz < 55000:
        t.sleep (1)
        print ("Debe calibrar de nuevo el sensor KY-026")
        t.sleep (0.5)
        if luz < 55000:
            print ("Pruebe girando el tornillo en sentido antihorario")
        luz = calibrar (0,2)
    t.sleep (1)

    ## calibracion
    try:
        print ("Calibracion sensor KY-026 correcta")
        rta = int (input ("Desea arrancar la pelicula? \n1- SI \n2- NO \n>"))
        if rta == 1:
            print("Prendemos la televisión")
            t.sleep (1)
            pelicula = True
            print("Apagamos las luces...", end="\r")
            t.sleep(1)
            luz_sala.off()
            t.sleep(1)

            mostrar_vol(num_3)  # Volumen inicial.
            prom = 0
            print ("Mientras pasan los anuncions, veremos si el sensor de sonido esta
calibrado, se le pide silencio al publico")
            rta = int (input ("Ver valores sensor de sonido? \n1- SI \n2- NO \n>" ))
```

```
        promcal= calibrar (rta,1)
        actual = normal
        #print (normal)
        while promcal >= actual [2] or promcal <= actual [1]: #bucle se ejecuta hasta que el sensor este calibrado
            print ("Debe calibrar de nuevo el sensor KY-037")
            t.sleep (1)
            if promcal <= actual [1]:
                print ("Pruebe girando el tornillo en sentido antihorario")
            else:
                print ("Pruebe girando el tornillo en sentido horario")
            t.sleep (1)
            rta = int (input ("Ver valores sensor de sonido? \n1- SI \n2- NO \n>" ))
            promcal= calibrar (rta,1)

        dB = convertir_dB (promcal)



        print(f"Nivel de dB calibrado: {dB:.2f} dB, es correcto, indica que el publico esta en silencio, y el volumen en 3")
        t.sleep (2)
        print ("Pelicula -El Padrino- reproduciendose, disfrutela!")
        t.sleep (0.5)
        print ("Si en cualquier momento desea cortar con la pelicula, presione cualquier tecla del teclado")
        t.sleep (2)



        seg = 0
        segcorridos = 0
        segcorridos2 = 0
```

```
        actual = normal

        prom = 36000
        ##IMPORTANTE = CALIBRACION SENSOR EN 33500 -34500 publico en silencio
        while pelicula and seg <= 360: ## 3 minutos de pelicula -> 3 horas de pelicula.
            mediciones = []
            band = False
            ## volumen
            if prom > actual [2]:
                band = True
            if actual [0] == 3: # mi volumen es normal
                if segcorridos2 == 4:  #pasaron 4 segundos seguidos de que mi medicion supero al rango actual
                    if prom > alto [2]: # si este rango supera al rango del alto lo mando a 8
                        actual = await ir_avol(8, "subir",prom, actual, segcorridos2)

                if segcorridos==  7: #pasaron 7 segundos seguidos de que mi medicion supero al rango actual
                    if normal[2] <= prom <= superior [1] : #ste rango supera al valor normal pero es menor al superior
                        actual = await ir_avol(6, "subir",prom, actual, segcorridos)

            if actual [0] == 6:
                if segcorridos == 7:
                    actual= await ir_avol(8, "subir",prom, actual, segcorridos)

            if actual [0] == 8:
                if segcorridos == 7:
                    fin_pelicula ()
```

```
                    pelicula = False
                    break

            if prom < actual [1] :
                band = True
                if actual [0] == 3:
                    pass
                elif actual [0]==6:
                    if segcorridos == 1:
                        actual= await ir_avol(3, "baja",prom, actual, segcorridos)
                else:
                    if segcorridos == 1:
                        actual = await ir_avol(6, "baja",prom,actual, segcorridos)

            if not band:
                segcorridos = 0
                if actual [0] == 3:
                    segcorridos2 = 0



            for i in range (0,10):  # 10 muestras en  en un segundo
                valor_adc = sonido.read_u16 ()
                if valor_adc <= 33000:
                    valor_adc = 34000

                if valor_adc >= 36800:
                    valor_adc = valor_adc + 20000

                if valor_adc >= 34500 and valor_adc <= 36800:
                    valor_adc = valor_adc + 11000

                mediciones.append (valor_adc)
```

```
            """
            normal = [3,33400,34500]
            alto = [6,34501,36800]
            superior = [8,36801,38400]
            """

            segcorridos += 1
            segcorridos2 += 1
            prom = (prom + (sum (mediciones)) // (len(mediciones))) // 2
#promedio de ruido



            # Sensor de movimiento
            sensor_foco = foco_puerta.read_u16()
            if sensor_foco < 10000:
                await pausar_todo()
                print ("Pelicula reanudada", end= "\r")

            t.sleep(1)  # paso un segundo
            seg += 1
            #print (f"Promedio {prom} ---- actual [{actual [1]} - {actual[2]}]")
            #print (f"Segundos transcurridos {seg}")

        if seg > 360:
            t.sleep (1)
            print ("Fin de la  pelicula")
            fin_normal ()
            t.sleep (2)
            print ("La Cosa Nostra te agredece por ver su pelicula")
```

```
        else:
            print ("Que aburrido... ")
        t.sleep (0.5)
    except KeyboardInterrupt:
        print("\nInterrupción de teclado detectada. La pelicula se termina.")
        fin_normal()
        t.sleep (2)
        await asyncio.sleep(2)
    pass



def get_app_data():
    global actual, foco_puerta, prom, ant

    return {
        "volumen_actual": ant,  # El volumen actual (normal, alto o superior).
        "promedio_ruido": convertir_dB (prom),  # Promedio de las mediciones de ruido.
        "entro_alguien": activado
    }




async def main():
    # Funcionamiento del equipo y monitoreo con el maestro se ejecutan concurrentemente.
    await asyncio.gather(
        operations(),
        micro_monitoring.monitoring(get_app_data)  # Monitoreo del maestro
    )

asyncio.run(main())
```

[Link a repositorio en GITHUB]
[Link a repositorio en GITHUB-maestro]

Diagrama del Circuito