ДОДАТОК А
ЛІСТИНГ ПРОГРАМИ

```
import network
import urequests as requests
from time import sleep
from machine import Pin, ADC
import utime
import dht

# Налаштування пінів для DHT22, PIR, вібраційного датчика, датчика диму та світлодіода
dht_pin = Pin(22)
pir_sensor = Pin(15, Pin.IN)
vibration_sensor = Pin(16, Pin.IN)
led_alarm = Pin(10, Pin.OUT)       # Світлодіод для тривоги
led_vibration = Pin(6, Pin.OUT)    # Світлодіод для вібрації
led_motion = Pin(17, Pin.OUT)      # Світлодіод для руху
mq2_AO = ADC(26)

dht_sensor = dht.DHT22(dht_pin)

# Встановлення порогів
TEMP_THRESHOLD = 27.5  # Температурний поріг у градусах Цельсія
SMOKE_THRESHOLD = 40000

# Підключення до Wi-Fi
wlan = network.WLAN(network.STA_IF)
wlan.active(True)
wlan.connect("TP-Link_200E", "********")
sleep(3)
print("Wi-Fi connected:", wlan.isconnected())
```

```
# Змінні для збереження попередніх станів датчиків
previous_pir_state = None
previous_smoke_state = None
previous_vibration_state = None
previous_temp_state = None

def blink_led(pin, times, duration=0.5):
    for _ in range(times):
        pin.on()
        sleep(duration)
        pin.off()
        sleep(duration)

while True:
    try:
        # Зчитування даних з датчика DHT22
        dht_sensor.measure()
        temperature_celsius = dht_sensor.temperature()
        humidity_percent = dht_sensor.humidity()

        # Виведення даних у консоль
        print("Temperature: {:.2f} °C".format(temperature_celsius))
        print("Humidity: {:.2f} %".format(humidity_percent))

        # Перевірка стану температури
        current_temp_state = temperature_celsius > TEMP_THRESHOLD
        if current_temp_state != previous_temp_state:
            if current_temp_state:
                print("Temperature threshold exceeded!")
                blink_led(led_alarm, 4)  # Блимання світлодіодом
```

```
            requests.post(
                "https://ntfy.sh/dyplom",
                data="Перевищено температурний поріг!",
                headers={
                    "Title": "ALERT: HIGH TEMPERATURE",
                    "Priority": "5",
                    "Tags": "thermometer",
                }
            )
            # Відправка додаткових повідомлень
            requests.post(
                "https://ntfy.sh/dyplom",
                data="Fire department called, siren activated, fire suppression and ventilation systems activated",
                headers={
                    "Title": "ALERT: FIRE RESPONSE",
                    "Priority": "5",
                    "Tags": "fire_engine",
                }
            )
        previous_temp_state = current_temp_state

    # Перевірка стану PIR сенсора
    current_pir_state = pir_sensor.value()
    if current_pir_state != previous_pir_state:
        if current_pir_state == 1:
            print("Рух є!")
            blink_led(led_motion, 4)  # Блимання світлодіодом
            requests.post(
                "https://ntfy.sh/dyplom",
```

```
            data="Movement detected by sensor",
            headers={
                "Title": "ALERT: MOVEMENT DETECTED",
                "Priority": "5",
                "Tags": "rotating_light",
            }
        )
        # Відправка додаткових повідомлень
        requests.post(
            "https://ntfy.sh/dyplom",
            data="Security service called, siren activated",
            headers={
                "Title": "ALERT: SECURITY RESPONSE",
                "Priority": "5",
                "Tags": "police_car",
            }
        )
    previous_pir_state = current_pir_state

# Перевірка стану датчика вібрації
current_vibration_state = vibration_sensor.value()
if current_vibration_state != previous_vibration_state:
    if current_vibration_state == 1:
        print("Виявлено вібрацію!")
        blink_led(led_vibration, 4)  # Блимання світлодіодом
        requests.post(
            "https://ntfy.sh/dyplom",
            data="Vibration detected by sensor",
            headers={
                "Title": "ALERT: VIBRATION DETECTED",
```

```
            "Priority": "5",
            "Tags": "vibration",
        }
    )
    # Відправка додаткових повідомлень
    requests.post(
        "https://ntfy.sh/dyplom",
        data="Security service called, siren activated",
        headers={
            "Title": "ALERT: SECURITY RESPONSE",
            "Priority": "5",
            "Tags": "police_car",
        }
    )
  previous_vibration_state = current_vibration_state

# Читання та перетворення аналогового значення з MQ-2
smoke_value = mq2_AO.read_u16()
print("AO:", smoke_value)

# Перевірка стану датчика диму
current_smoke_state = smoke_value > SMOKE_THRESHOLD
if current_smoke_state != previous_smoke_state:
  if current_smoke_state:
    print("Дим зафіксовано!")
    blink_led(led_alarm, 4)  # Блимання світлодіодом тривоги
    requests.post(
        "https://ntfy.sh/dyplom",
        data="Smoke detected by sensor",
        headers={
```

```
                "Title": "ALERT: SMOKE DETECTED",
                "Priority": "5",
                "Tags": "fire",
            }
        )
        # Відправка додаткових повідомлень
        requests.post(
            "https://ntfy.sh/dyplom",
            data="Fire department called, siren activated, fire suppression and ventilation systems activated",
            headers={
                "Title": "ALERT: FIRE RESPONSE",
                "Priority": "5",
                "Tags": "fire_engine",
            }
        )
    previous_smoke_state = current_smoke_state

  utime.sleep(0.2)  # Затримка перед наступним циклом

except Exception as e:
  print("Error reading sensors:", str(e))
```

ДОДАТОК Б
ЛІСТИНГ ПРОГРАМИ У WOKWI

```
# From: https://www.hackster.io/shilleh/connect-mpu-6050-to-raspberry-pi-pico-w-7f3345

import machine
from time import sleep
import dht
from imu import MPU6050
from machine import Pin, I2C

# Налаштування піна 15 як входу для PIR-датчика
pir_sensor = machine.Pin(15, machine.Pin.IN)

# Налаштування піна 6 як виходу для світлодіода
led_pir = machine.Pin(6, machine.Pin.OUT)

# Налаштування пінів для світлодіодів
led_temp = machine.Pin(8, machine.Pin.OUT)
led_smoke = machine.Pin(7, machine.Pin.OUT)
led_vibration = machine.Pin(1, machine.Pin.OUT)

# Налаштування піна 22 як входу для DHT22 датчика
dht_sensor = dht.DHT22(machine.Pin(22))

i2c = I2C(0, sda=Pin(4), scl=Pin(5), freq=400000)  # Pico
imu = MPU6050(i2c)

# Початковий стан для відслідковування зміни стану PIR-датчика
motion_detected = False

# Початковий стан для відслідковування стану температури і диму
```

```
previous_temperature = None
previous_smoke = 0

# Початковий стан для IMU датчика
prev_ax, prev_ay, prev_az = 0, 0, 0
prev_gx, prev_gy, prev_gz = 0, 0, 0
prev_tem = 0

while True:
    # Зчитування даних з DHT22
    try:
        dht_sensor.measure()
        temperature = dht_sensor.temperature()
        smoke = dht_sensor.humidity()  # Симуляція рівня диму через вологість
    except OSError as e:
        print("Помилка зчитування з DHT22:", e)
        temperature = None
        smoke = None

    # Виведення значень температури і диму
    if temperature is not None and smoke is not None:
        print("Температура: {:.2f} C".format(temperature))
        print("Рівень диму: {:.2f}".format(smoke))

    # Перевірка PIR-датчика і умов температури/диму
    if pir_sensor.value() == 1 and not motion_detected:
        # Рух виявлено вперше
        print("Рух виявлено!")
        led_pir.value(1)  # Запалюємо світлодіод
        motion_detected = True
```

```
    elif pir_sensor.value() == 0 and motion_detected:
        # Рух більше не виявлено
        print("Руху немає")
        led_pir.value(0)  # Гасимо світлодіод
        motion_detected = False
    elif pir_sensor.value() == 0 and not motion_detected:
        # Якщо руху немає і раніше також не було, продовжуємо виводити
"Руху немає"
        print("Руху немає")

    # Перевірка умов для температури
    if temperature is not None:
        if temperature > 27.5 and (previous_temperature is None or
previous_temperature <= 27.5):
            print("Температура вища за 27.5 C!")
            led_temp.value(1)  # Запалюємо світлодіод на 8 піні
        elif temperature <= 27.5 and (previous_temperature is None or
previous_temperature > 27.5):
            led_temp.value(0)  # Гасимо світлодіод на 8 піні
        previous_temperature = temperature

    # Перевірка умов для диму
    if smoke is not None:
        if smoke > 0 and previous_smoke == 0:
            print("Виявлено дим!")
            led_smoke.value(1)  # Запалюємо світлодіод на 7 піні
        elif smoke == 0 and previous_smoke > 0:
            led_smoke.value(0)  # Гасимо світлодіод на 7 піні
        previous_smoke = smoke
```

```
# Зчитування даних з MPU6050
ax = round(imu.accel.x, 2)
ay = round(imu.accel.y, 2)
az = round(imu.accel.z, 2)
gx = round(imu.gyro.x)
gy = round(imu.gyro.y)
gz = round(imu.gyro.z)
tem = round(imu.temperature, 2)

# Перевірка змін у даних IMU
if (ax != prev_ax or ay != prev_ay or az != prev_az or
    gx != prev_gx or gy != prev_gy or gz != prev_gz or
    tem != prev_tem):
    print("Вібрація зафіксована!")
    led_vibration.value(1)  # Запалюємо світлодіод на 1 піні
else:
    led_vibration.value(0)  # Гасимо світлодіод на 1 піні

# Оновлення попередніх значень
prev_ax, prev_ay, prev_az = ax, ay, az
prev_gx, prev_gy, prev_gz = gx, gy, gz
prev_tem = tem

sleep(0.2)  # Затримка для стабільності та оновлення даних
```

ДОДАТОК В
ДЕМОНСТРАЦІЙНИЙ МАТЕРІАЛ

Формат	Позиція	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кільк.	Примітки
			Текстові документи		
А4	1	ГЮИК. 442351.003 ПЗ	Пояснювальна записка	86	с.
			Інші документи		
	2		Демонстраційний	12	с.
			матеріал у		
			вигляді презентації		

Змін.	Арк.	Номер докум.	Підпис	Дата	ГЮИК. 442351.003 ПЗ			
Розроб.		Васильченко			Розробка системи автоматизації охоронно-пожежної сигналізації на приладобудівному підприємстві	Літера	Аркуш	Аркушів
Перевір.		Сотник				Н	1	1
						Кафедра КІТАР ХНУРЕ		
Н.контр.		Бронніков						
Затв.		Невлюдов						