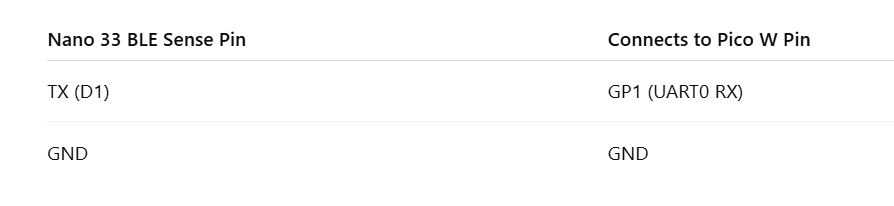
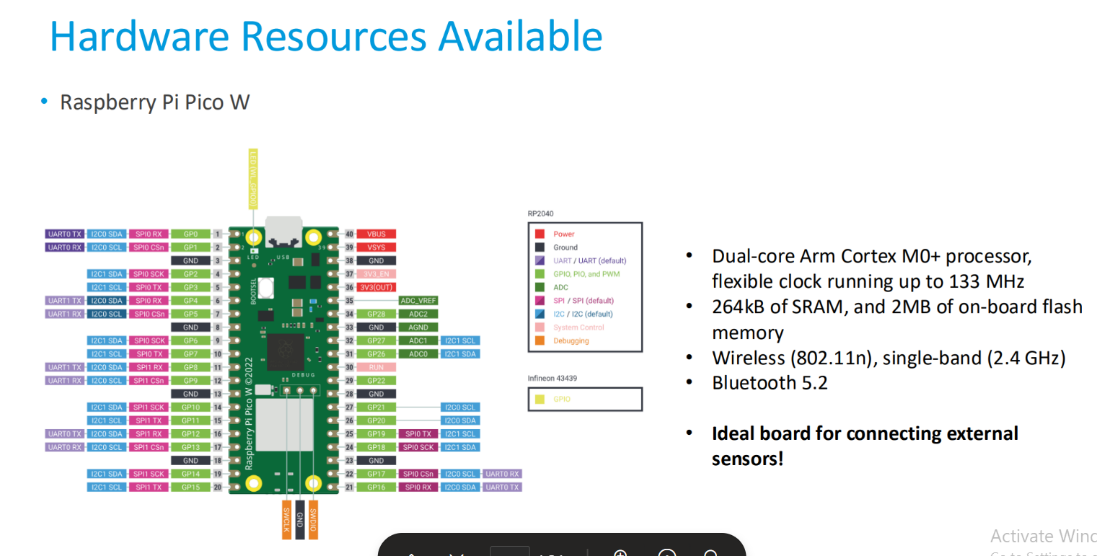
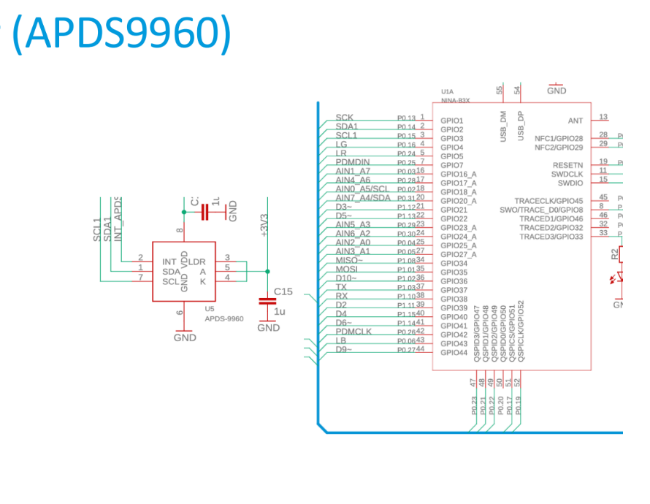
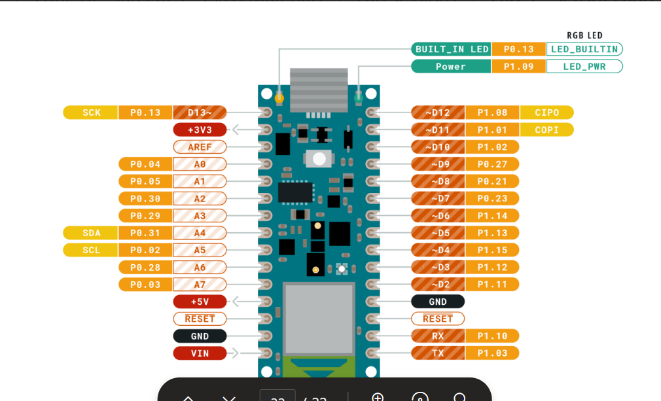
1. We will publish the imu data from nano ble to pico and publilsh it to mqtt server, At end we will subscribe from mqtt using python

S-1) Connect the nano ble & pico through usb to laptop

S-2) connect nano ble with pico through following ports:- ( also follow the connection process as mentioned in lab-5 ppt)





S-3) Run following scripts :-

S-3a) Arduino Code on Nano 33 BLE Sense

Arduino sketch that:

* Reads IMU sensor (Accelerometer, Gyro, etc.)
* Sends data over Serial (UART)

#include <Arduino\_LSM9DS1.h> // For IMU sensor

void setup() {

Serial.begin(115200);

Serial1.begin(115200);

while (!Serial);

if (!IMU.begin()) {

Serial.println("Failed to initialize IMU!");

while (1);

}

}

void loop() {

float x, y, z;

if (IMU.accelerationAvailable()) {

IMU.readAcceleration(x, y, z);

Serial.print(x);

Serial.print(',');

Serial.print(y);

Serial.print(',');

Serial.println(z);

Serial1.print(x);

Serial1.print(',');

Serial1.print(y);

Serial1.print(',');

Serial1.println(z);

}

delay(500);

}

Note:-

Println serial => to computer; serial1 => to usb further connected to pico.

Println is important as it helps to detect the new line on Part to pico (read by line = uart.readline() )

Following output comes :-

-0.02,-1.18,-0.40

-0.05,-0.56,-0.63

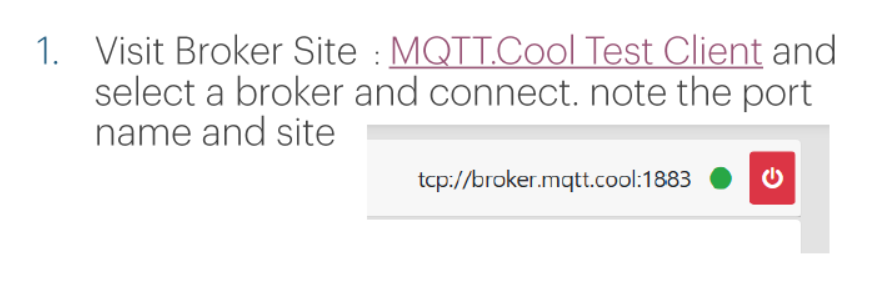
-0.02,-1.46,0.70

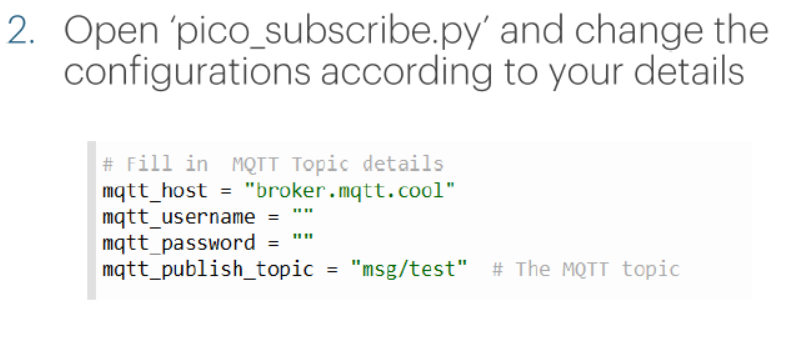
0.04,-1.00,-0.09

-0.01,-1.00,-0.12

-0.01,-1.00,-0.11

**S-3b) Visit following site** [**https://testclient-cloud.mqtt.cool/**](https://testclient-cloud.mqtt.cool/)





## S-3c) MicroPython Code on Pico W

✅ On Pico W, you will:

* Open **UART0** (machine.UART(0))
* Read incoming data line by line.
* Parse the data (if needed).
* Publish to MQTT.

from machine import UART, Pin

import network

import time

from umqtt.simple import MQTTClient

import gc

# Set up UART

uart = UART(0, baudrate=115200, tx=Pin(0), rx=Pin(1))

# Connect to Wi-Fi

wifi\_ssid = "siddharth\_hotspot"

wifi\_password = "12345789"

wlan = network.WLAN(network.STA\_IF)

wlan.active(True)

wlan.connect(wifi\_ssid, wifi\_password)

while not wlan.isconnected():

print('Waiting for connection...')

time.sleep(1)

print('Connected to WiFi')

# Connect to MQTT Broker

mqtt\_host = "broker.mqtt.cool"

mqtt\_client\_id = "pico\_uart\_mqtt"

mqtt\_client = MQTTClient(client\_id=mqtt\_client\_id, server=mqtt\_host)

try:

mqtt\_client.connect()

except Exception as e:

print("MQTT connection failed:", e)

time.sleep(2)

# Main loop

while True:

if uart.any():

line = uart.readline()

if line:

line = line.decode('utf-8').strip()

print("Received from Nano:", line)

try:

mqtt\_client.publish("patient/imu", line)

except Exception as e:

print("MQTT publish failed:", e)

# Print available memory

gc.collect()

print("Free memory:", gc.mem\_free())

time.sleep(0.5)

Following output comes:-

ree memory: 167584

Received from Nano: -0.05,-1.00,-0.12

Free memory: 167408

Received from Nano: -0.05,-1.00,-0.12

Free memory: 167232

Received from Nano: -0.05,-1.00,-0.12

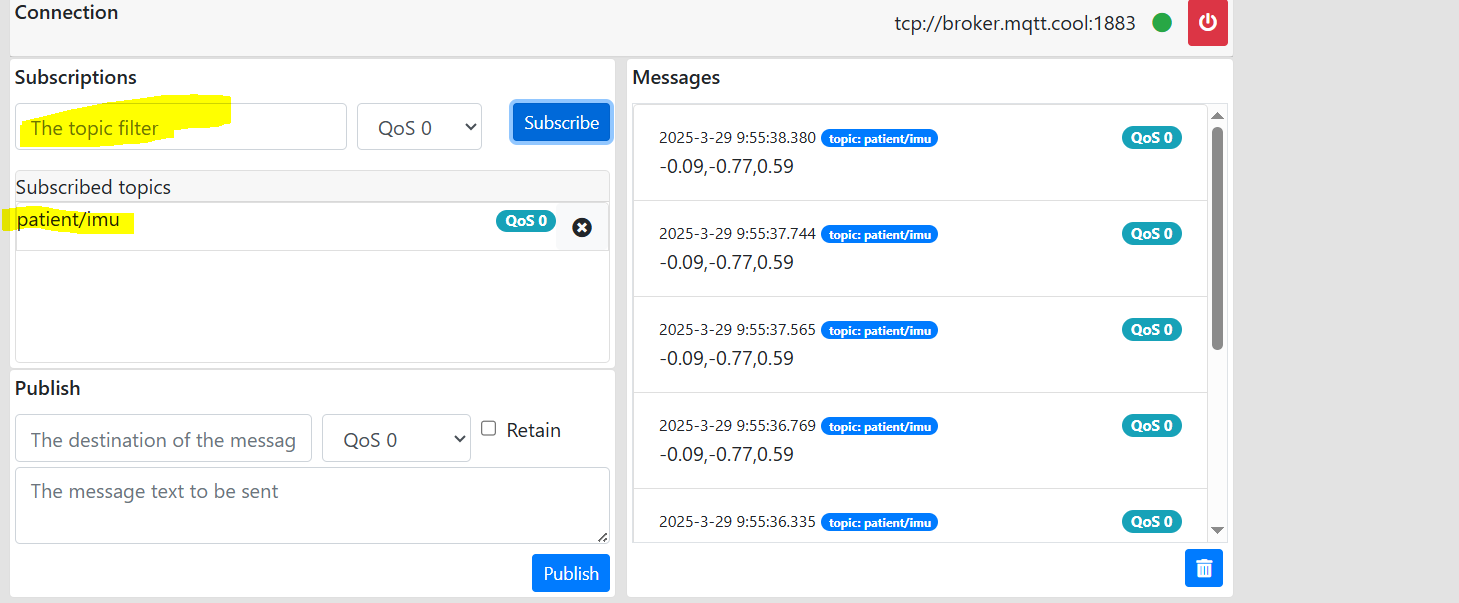
Free memory: 167056

At the end data start publishing on MQTT

**Change the mqtt\_host in pico as per configuration details**

**Now in case of broker:-**

Put patient/imu in the “topic filter” and press Subscribe, data will start coming on messages.

**Put**

S-4) Now we will subscribe from mqtt server

Open another thonny window and run following code:- ( in local Python3)

# -\*- coding: utf-8 -\*-

"""Untitled12.ipynb

Automatically generated by Colab.

Original file is located at

https://colab.research.google.com/drive/16efQJjb-BiX8gtBE6cWErWua1stDUD6i

"""

import paho.mqtt.client as mqtt

# Define MQTT parameters

MQTT\_BROKER = "broker.mqtt.cool" # or "localhost" if using Mosquitto locally

MQTT\_TOPIC = "patient/imu"

# Callback function when connection is established

def on\_connect(client, userdata, flags, rc):

print("Connected to MQTT broker with result code " + str(rc))

client.subscribe(MQTT\_TOPIC)

# Callback function when a message is received

def on\_message(client, userdata, msg):

incoming\_data = msg.payload.decode('utf-8')

print(f"Received: {incoming\_data}")

# Example processing

try:

x, y, z = map(float, incoming\_data.split(',')) # assuming message is "x,y,z"

if abs(x) > 0. or abs(y) > 1.5 or abs(z) > 1.5:

print("⚠️ Alarm: High movement detected!")

else:

print("✅ Safe movement.")

except Exception as e:

print("Error processing message:", e)

# Create MQTT Client and connect

client = mqtt.Client()

client.on\_connect = on\_connect

client.on\_message = on\_message

client.connect(MQTT\_BROKER, 1883, 60)

# Blocking loop to the broker

client.loop\_forever()

Now if you want to run the same code using the charger/adapter

**IN case of pico ,**  stop the running code GO to File> Save copy> Save to:- “Raspberry pi pico” > main.py

Now disconnect the usb from laptop and connect to charger. It automatically starts.

1. **For Nano BLE:-**

Update the code and upload on the device:-

#include <Arduino\_LSM9DS1.h> // For IMU sensor

void setup() {

  Serial.begin(115200);

  Serial1.begin(115200);

  unsigned long startTime = millis();

  while (!Serial && (millis() - startTime < 3000));  // Optional wait for Serial

  if (!IMU.begin()) {

    Serial.println("Failed to initialize IMU!");

    while (1);

  }

}

void loop() {

  float x, y, z;

  if (IMU.accelerationAvailable()) {

    IMU.readAcceleration(x, y, z);

    Serial.print(x);

    Serial.print(',');

    Serial.print(y);

    Serial.print(',');

    Serial.println(z);

    Serial1.print(x);

    Serial1.print(',');

    Serial1.print(y);

    Serial1.print(',');

    Serial1.println(z);

  }

  delay(500);

}

Note:- earlier it was busy waiting since once disconnected from the laptop due to unavailability of serial it was waiting