|  |  |
| --- | --- |
| Madde mosquito broker run automatically and checked the status as seen here |  |
| Disable and stop broker |  |
| Simple sub and pub |  |
| Lab assignment | // File 1: /mqtt\_publisher.py  import paho.mqtt.client as mqtt  import time  # MQTT Config  BROKER = "waafiadam-pi.local" # Replace with your broker IP if needed  PORT = 1883  TOPIC = "iot/capture"  # MQTT Setup  client = mqtt.Client(client\_id="Publisher")  client.connect(BROKER, PORT)  while True:  input("Press Enter to capture an image...") # User triggers capture  client.publish(TOPIC, "capture")  print("Capture command sent!")  time.sleep(2) # Avoid spamming  // File 2: /mqtt\_subscriber.py  import paho.mqtt.client as mqtt  import cv2  import base64  import numpy as np  # MQTT Config  BROKER = "waafiadam-pi.local" # Replace with your broker IP if needed  PORT = 1883  SUBSCRIBE\_TOPIC = "iot/capture"  def on\_message(client, userdata, message):  print("Message received! Capturing image...")  # Open webcam  cap = cv2.VideoCapture(0)  ret, frame = cap.read()  cap.release()  if ret:  # Encode image as Base64  \_, buffer = cv2.imencode('.jpg', frame)  img\_base64 = base64.b64encode(buffer).decode()  # Publish image  client.publish("iot/image", img\_base64)  print("Image sent via MQTT!")  else:  print("Failed to capture image.")  # MQTT Setup  client = mqtt.Client(client\_id="Subscriber")  client.on\_message = on\_message  client.connect(BROKER, PORT)  # Subscribe to capture topic  client.subscribe(SUBSCRIBE\_TOPIC)  client.loop\_forever()  pip  // File 3: /mqtt\_image\_receiver.py  import paho.mqtt.client as mqtt  import base64  import cv2  import numpy as np  # MQTT Config  BROKER = "waafiadam-pi.local"  PORT = 1883  TOPIC = "iot/image"  def on\_message(client, userdata, message):  print("Image received via MQTT!")    # Decode Base64 image  img\_data = base64.b64decode(message.payload)  np\_arr = np.frombuffer(img\_data, np.uint8)  img = cv2.imdecode(np\_arr, cv2.IMREAD\_COLOR)    # Show image  cv2.imshow("Received Image", img)  cv2.waitKey(5000) # Display for 5 sec  cv2.destroyAllWindows()  # MQTT Setup  client = mqtt.Client(client\_id="ImageReceiver")  client.on\_message = on\_message  client.connect(BROKER, PORT)  client.subscribe(TOPIC)  client.loop\_forever()  Output: |
|  |  |
|  |  |