

Singapore Institute of Technology

BEng (Hons) Information and Communications Technology majoring in Software Engineering

INF2009 Edge Computing and Analytics

Academic Year 2024/2025 Trimester 2

Week 8 Lab – IOT Communications: MQTT

Name: Lim Chee Hean

Student ID: 2201529

4. Test MQTT Communication

```
(mqtt) cheehean@raspberrypi:~/Labs/mqtt $ python mqtt_publisher.py
/home/cheehean/Labs/mqtt/mqtt_publisher.py:4: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  client = mqtt.Client()
```

Message sent from mqtt_publisher.py

```
(mqtt) cheehean@raspberrypi:~/Labs/mqtt $ python mqtt_subscriber.py
/home/cheehean/Labs/mqtt/mqtt_subscriber.py:6: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  client = mqtt.Client()
Received message 'Hello, MQTT!' on topic 'test/topic'
Received message 'Hello, MQTT!' on topic 'test/topic'
Received message 'Hello, MQTT!' on topic 'test/topic'
Received message 'Hello, MQTT!' on topic 'test/topic'
Received message 'Hello, MQTT!' on topic 'test/topic'
Received message 'Hello, MQTT!' on topic 'test/topic'
Received message 'Hello, MQTT!' on topic 'test/topic'
Received message 'Hello, MQTT!' on topic 'test/topic'
```

Message received on mqtt_subscriber.py

5. Lab Assignment

The system works as follows:

- Aside from the MQTT broker, there are 2 components, **Camera** (mqtt_camera.py) and **Trigger** (mqtt_trigger.py).
- The **Camera is subscribed to camera/capture** and **Trigger is subscribed to camera/image**.
- First, the Trigger waits for an input from the user. When the enter key is pressed, it **publishes an event to camera/capture**.
- Next, the Camera receives the event on **camera/capture** and captures an image with its webcam. It then **publishes the image to camera/image**.

- Finally, the Trigger receives the image on **camera/image**, where it saves the image to a file locally and displays the image on the screen.

```
(.venv) C:\Users\Lim Chee Hean\Desktop\Edge Computing\mqtt>python mqtt_camera.py
Ready to capture images
Received request to capture image
Capturing image...
Image captured
Publishing image...
Image published
```

Running mqtt_camera.py

```
(.venv) C:\Users\Lim Chee Hean\Desktop\Edge Computing\mqtt>python mqtt_trigger.py
Press enter to trigger an image capture:
Requesting to capture image...
Image capture requested

Press enter to trigger an image capture:

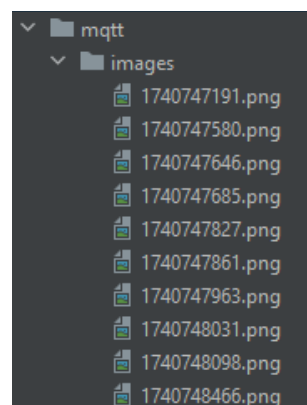
Image received, saving to file...
File saved as images/1740748466.png

Press enter to trigger an image capture:
|
```

Running mqtt_trigger.py



Display received image



Saved received images

```

1 usage
def on_message(client, userdata, message):
    filename = f"images/{int(time())}.png"

    print(f"\n\nImage received, saving to file...")
    with open(filename, "wb") as file:
        file.write(message.payload)
    print(f"File saved as {filename}")

    imshow( winname: "Received Image", imread(filename))
    waitKey(0)

    # Rewrite input prompt in console
    print("\nPress enter to trigger an image capture: ")
    while True:
        pass

# Create folder to store images
mkdirs( name: "images", exist_ok=True)

# Setup mqtt
client = mqtt.Client(CallbackAPIVersion.VERSION2)
client.on_message = on_message
client.connect( host: "192.168.1.10", port: 1883)
client.subscribe("camera/image")
client.loop_start()

# Send capture request on pressing enter
while True:
    input("Press enter to trigger an image capture: ")
    print("Requesting to capture image...")
    client.publish("camera/capture")
    print("Image capture requested\n")

```

mqtt_trigger.py

```

1 usage
def on_message(client_, userdata, message):
    print("Received request to capture image")

    print("Capturing image...")
    capture = VideoCapture(0)
    success, frame = capture.read()
    if not success:
        print("Failed to capture image")
        return
    print("Image captured")

    print("Publishing image...")
    _, buffer = imencode( ext: ".png", frame)
    data = buffer.tobytes()
    client.publish( topic: "camera/image", data)
    print("Image published")

print("Ready to capture images")
client = mqtt.Client(CallbackAPIVersion.VERSION2)
client.on_message = on_message
client.connect( host: "192.168.1.10", port: 1883)
client.subscribe("camera/capture")
client.loop_forever()

```

mqtt_camera.py

Copies of these files are submitted with this document.