

INF2009 - MQTT Lab

[Mosquitto.conf]

```

pi@pi: ~
File Edit Tabs Help

Processing triggers for libc-bin (2.28-10+rpt2+rpi1+deb10u2) ...
pi@pi:~ $ sudo nano /etc/mosquitto/mosquitto.conf
pi@pi:~ $ sudo mosquitto -c /etc/mosquitto/mosquitto.conf
pi@pi:~ $ systemctl status mosquitto
● mosquitto.service - Mosquitto MQTT v3.1/v3.1.1 Broker
   Loaded: loaded (/lib/systemd/system/mosquitto.service; enabled; vendor preset
   Active: active (running) since Thu 2025-03-06 04:07:01 GMT; 2min 53s ago
     Docs: man:mosquitto.conf(5)
           man:mosquitto(8)
   Main PID: 1215 (mosquitto)
      Tasks: 1 (limit: 2059)
    CGroup: /system.slice/mosquitto.service
            └─1215 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf

Mar 06 04:07:01 pi systemd[1]: Starting Mosquitto MQTT v3.1/v3.1.1 Broker...
Mar 06 04:07:01 pi systemd[1]: Started Mosquitto MQTT v3.1/v3.1.1 Broker.

pi@pi:~ $ sudo systemctl disable mosquitto
Synchronizing state of mosquitto.service with SysV service script with /lib/syst
emd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable mosquitto
Removed /etc/systemd/system/multi-user.target.wants/mosquitto.service.
pi@pi:~ $ sudo systemctl stop mosquitto
pi@pi:~ $

```

[MQTT Subscriber]

[illegible]

[MQTT Publisher]

The screenshot shows a Raspberry Pi desktop environment accessed via RealVNC Viewer. The desktop background is a blue Raspberry Pi logo. A Geany IDE window is open, displaying a Python script named `mqtt_publisher.py`. The script imports `paho.mqtt.client` and `time`, creates an MQTT client named `client`, connects it to `192.168.248.184` on port `1883`, and enters a `while True` loop where it publishes `"Hello, MQTT!"` to the `test/topic` every 5 seconds.

Below the Geany window, a terminal window is open, showing the execution of the script. The terminal output indicates a connection failure with the error `OSError: [Errno 113] No route to host`. The traceback shows the error originates from the `socket.create_connection` call in the `paho.mqtt.client` library. The user has entered the command `python3 mqtt_publisher.py` in the terminal.

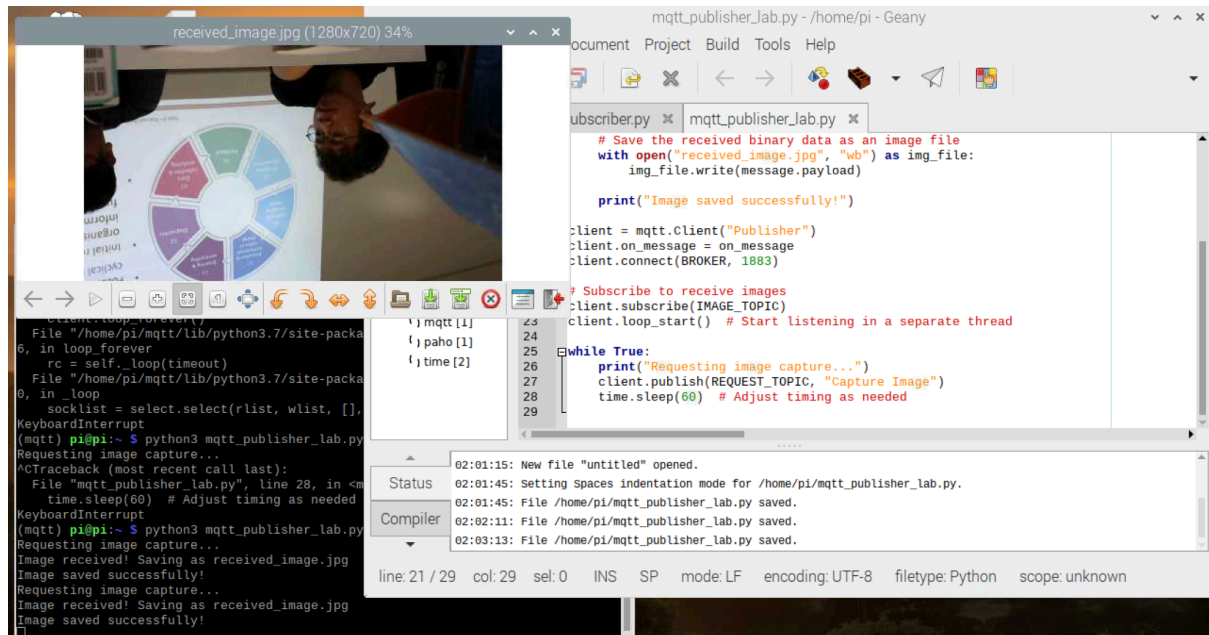
```
mqtt_publisher.py - /home/INF2009T2 - Geany
File Edit Search View Document Project Build Tools Help
Symbols mqtt_publisher.py x
Variables
  client [4]
Imports
  mqtt [1]
  mqtt [1]
  paho [1]
  time [2]
1 import paho.mqtt.client as mqtt
2 import time
3
4 client = mqtt.Client("Publisher")
5 client.connect("192.168.248.184", 1883)
6
7 while True:
8     client.publish("test/topic", "Hello, MQTT!")
9     time.sleep(5)
10
```

```
INF2009T2@INF2009T2: ~
File Edit Tabs Help
09:36:28:
09:36:28:
09:39:28:
09:40:50:
09:42:12:
09:44:15:
Status
line: 7 / 10 col: 10
File "/home/INF2009T2/myenv/lib/python3.11/site-packages/paho/mqtt/client.py",
line 1044, in reconnect
    sock = self._create_socket_connection()
File "/home/INF2009T2/myenv/lib/python3.11/site-packages/paho/mqtt/client.py",
line 3685, in _create_socket_connection
    return socket.create_connection(addr, timeout=self._connect_timeout, source_
address=source)
File "/usr/lib/python3.11/socket.py", line 851, in create_connection
    raise exceptions[0]
File "/usr/lib/python3.11/socket.py", line 836, in create_connection
    sock.connect(sa)
OSError: [Errno 113] No route to host
(myenv) INF2009T2@INF2009T2:~ S python3 mqtt_publisher.py
^CTraceback (most recent call last):
  File "/home/INF2009T2/mqtt_publisher.py", line 9, in <module>
    time.sleep(5)
KeyboardInterrupt
(myenv) INF2009T2@INF2009T2:~ S python3 mqtt_publisher.py
```

[Lab Assignment]

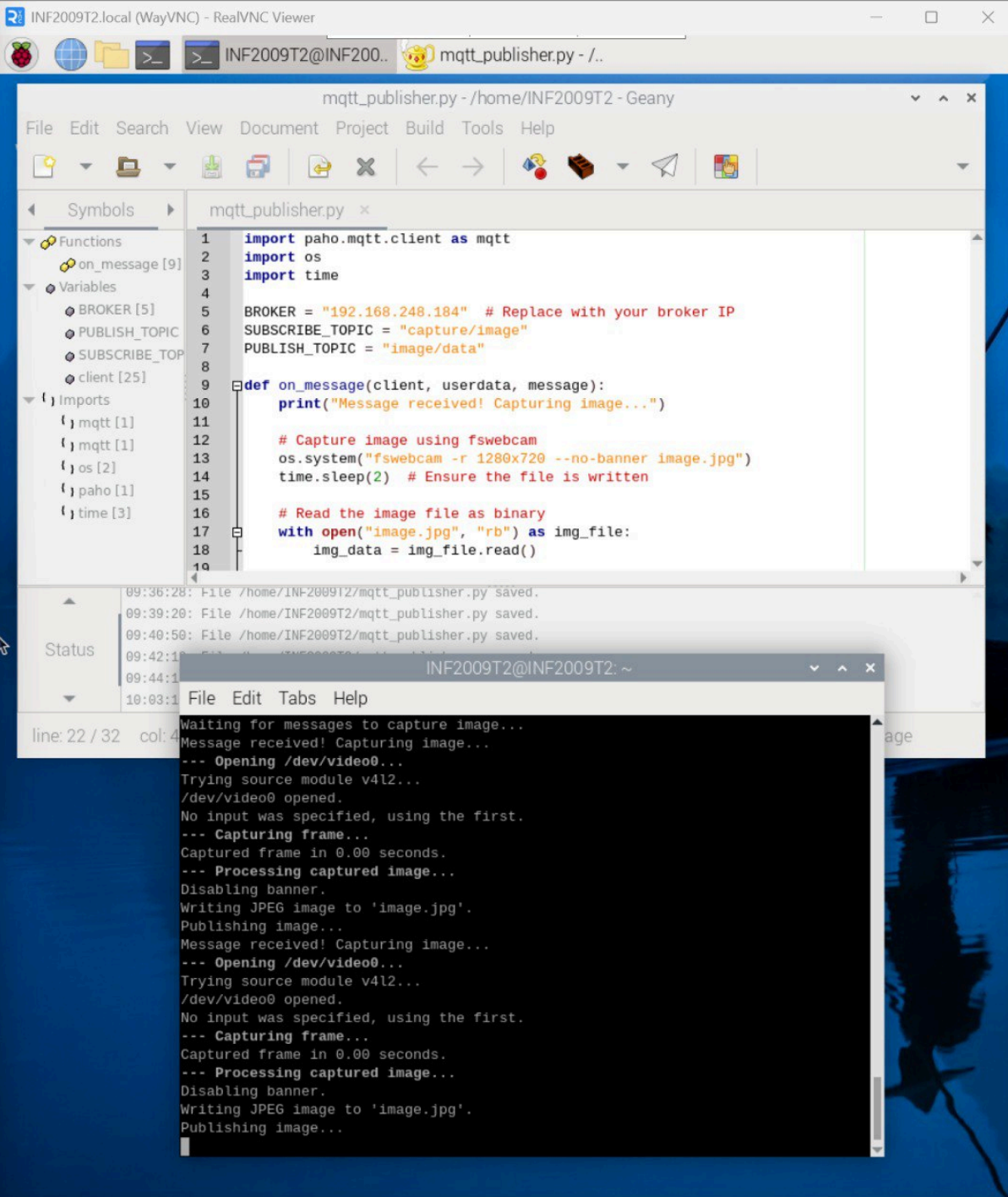
[MQTT Publisher]

Sends request for image capture to subscriber, receives image from subscriber's web-cam.



[MQTT Subscriber]

Listens for publisher request, captures image and sends it to publisher



The screenshot displays a Raspberry Pi desktop environment. At the top, a RealVNC Viewer window shows the desktop. The main application is Geany, an IDE for editing the file `mqtt_publisher.py` located at `/home/INF2009T2`. The script is a Python program that listens for MQTT messages, captures images using `fswebcam`, and publishes them to a broker.

```
1 import paho.mqtt.client as mqtt
2 import os
3 import time
4
5 BROKER = "192.168.248.184" # Replace with your broker IP
6 SUBSCRIBE_TOPIC = "capture/image"
7 PUBLISH_TOPIC = "image/data"
8
9 def on_message(client, userdata, message):
10     print("Message received! Capturing image...")
11
12     # Capture image using fswebcam
13     os.system("fswebcam -r 1280x720 --no-banner image.jpg")
14     time.sleep(2) # Ensure the file is written
15
16     # Read the image file as binary
17     with open("image.jpg", "rb") as img_file:
18         img_data = img_file.read()
```

The Geany interface includes a 'Symbols' pane on the left showing the script's structure, a 'Status' pane at the bottom left, and a terminal window at the bottom right. The terminal window, titled `INF2009T2@INF2009T2: ~`, shows the execution of the script. It displays the output of the `on_message` function, including the capture and publication of two images.

```
09:36:28: File /home/INF2009T2/mqtt_publisher.py saved.
09:39:20: File /home/INF2009T2/mqtt_publisher.py saved.
09:40:50: File /home/INF2009T2/mqtt_publisher.py saved.
09:42:10: File /home/INF2009T2/mqtt_publisher.py saved.
09:44:10: File /home/INF2009T2/mqtt_publisher.py saved.
10:03:11: File /home/INF2009T2/mqtt_publisher.py saved.

Waiting for messages to capture image...
Message received! Capturing image...
--- Opening /dev/video0...
Trying source module v4l2...
/dev/video0 opened.
No input was specified, using the first.
--- Capturing frame...
Captured frame in 0.00 seconds.
--- Processing captured image...
Disabling banner.
Writing JPEG image to 'image.jpg'.
Publishing image...
Message received! Capturing image...
--- Opening /dev/video0...
Trying source module v4l2...
/dev/video0 opened.
No input was specified, using the first.
--- Capturing frame...
Captured frame in 0.00 seconds.
--- Processing captured image...
Disabling banner.
Writing JPEG image to 'image.jpg'.
Publishing image...
```