MQTT Payload Examples

Complete examples of MQTT messages for controlling and monitoring the ESP32 SHTC3 device.

Table of Contents

Home Assistant Discovery Payloads

Temperature Sensor Discovery

Topic: `homeassistant/sensor/abc123_temperature/config`
Retain: `true`

```
{
  "name": "ESP32-SHTC3-abc123 Temperature",
  "unique_id": "abc123_temperature",
  "object_id": "abc123_temperature",
  "state_topic": "home/living_room/abc123/state",
  "value_template": "{{ value_json.temperature }}",
  "unit_of_measurement": "°C",
  "icon": "mdi:thermometer",
  "device_class": "temperature",
  "state_class": "measurement",
  "device": {
    "identifiers": ["abc123"],
    "name": "ESP32-SHTC3-abc123",
    "model": "ESP32-SHTC3",
    "manufacturer": "Custom",
    "sw_version": "1.0.0"
  },
  "availability_topic": "home/living_room/abc123/status",
  "payload_available": "online",
  "payload_not_available": "offline"
}
```

Humidity Sensor Discovery

Topic: `homeassistant/sensor/abc123_humidity/config`
Retain: `true`

```
{
  "name": "ESP32-SHTC3-abc123 Humidity",
  "unique_id": "abc123_humidity",
  "object_id": "abc123_humidity",
  "state_topic": "home/living_room/abc123/state",
  "value_template": "{{ value_json.humidity }}",
  "unit_of_measurement": "%",
  "icon": "mdi:water-percent",
  "device_class": "humidity",
  "state_class": "measurement",
  "device": {
    "identifiers": ["abc123"],
    "name": "ESP32-SHTC3-abc123",
    "model": "ESP32-SHTC3",
    "manufacturer": "Custom",
    "sw_version": "1.0.0"
  },
  "availability_topic": "home/living_room/abc123/status",
  "payload_available": "online",
  "payload_not_available": "offline"
}
```


RSSI Sensor Discovery

Topic: `homeassistant/sensor/abc123_rssi/config`
Retain: `true`


```
{
  "name": "ESP32-SHTC3-abc123 RSSI",
  "unique_id": "abc123_rssi",
  "object_id": "abc123_rssi",
  "state_topic": "home/living_room/abc123/state",
  "value_template": "{{ value_json.rssi }}",
  "unit_of_measurement": "dBm",
  "icon": "mdi:wifi",
  "device_class": "signal_strength",
  "state_class": "measurement",
  "device": {
    "identifiers": ["abc123"],
    "name": "ESP32-SHTC3-abc123",
    "model": "ESP32-SHTC3",
    "manufacturer": "Custom",
    "sw_version": "1.0.0"
  },
  "availability_topic": "home/living_room/abc123/status",
  "payload_available": "online",
  "payload_not_available": "offline"
}
```


Uptime Sensor Discovery

Topic: `homeassistant/sensor/abc123_uptime/config`
Retain: `true`

```
{
  "name": "ESP32-SHTC3-abc123 Uptime",
  "unique_id": "abc123_uptime",
  "object_id": "abc123_uptime",
  "state_topic": "home/living_room/abc123/state",
  "value_template": "{{ value_json.uptime }}",
  "unit_of_measurement": "s",
  "icon": "mdi:clock-outline",
  "state_class": "total_increasing",
  "device": {
    "identifiers": ["abc123"],
    "name": "ESP32-SHTC3-abc123",
    "model": "ESP32-SHTC3",
    "manufacturer": "Custom",
    "sw_version": "1.0.0"
  },
  "availability_topic": "home/living_room/abc123/status",
  "payload_available": "online",
  "payload_not_available": "offline"
}
```


LED Light Discovery

Topic: `homeassistant/light/abc123_led/config`
Retain: `true`


```
{
  "name": "ESP32-SHTC3-abc123 LED",
  "unique_id": "abc123_led",
  "object_id": "abc123_led",
  "command_topic": "home/living_room/abc123/led/command",
  "state_topic": "home/living_room/abc123/led/state",
  "schema": "json",
  "brightness": true,
  "rgb": true,
  "effect": true,
  "effect_list": [
    "solid",
    "rainbow",
    "breathing",
    "humidity_gauge",
    "temperature_gauge"
  ],
  "device": {
    "identifiers": ["abc123"],
    "name": "ESP32-SHTC3-abc123",
    "model": "ESP32-SHTC3",
    "manufacturer": "Custom",
    "sw_version": "1.0.0"
  },
  "availability_topic": "home/living_room/abc123/status",
  "payload_available": "online",
  "payload_not_available": "offline"
}
```

LED Control Commands

Turn On (Solid White)

Topic: `home/living_room/abc123/led/command`

```
{
  "state": "ON",
  "brightness": 255,
  "color": {
    "r": 255,
    "g": 255,
    "b": 255
  },
  "effect": "solid"
}
```

mosquitto_pub command:

```
mosquitto_pub -h localhost -t "home/living_room/abc123/led/command" -m
'{"state":"ON","brightness":255,"color":{"r":255,"g":255,"b":255},"effect":"solid"}'
```

Turn On (Solid Red)

Topic: `home/living_room/abc123/led/command`

```
{
  "state": "ON",
  "brightness": 200,
  "color": {
    "r": 255,
    "g": 0,
    "b": 0
  },
  "effect": "solid"
}
```

mosquitto_pub command:

```
mosquitto_pub -h localhost -t "home/living_room/abc123/led/command" -m
'{"state":"ON","brightness":200,"color":{"r":255,"g":0,"b":0},"effect":"solid"}'
```

Rainbow Effect

Topic: `home/living_room/abc123/led/command`

```
{
  "state": "ON",
  "brightness": 128,
  "effect": "rainbow"
}
```

mosquitto_pub command:

```
mosquitto_pub -h localhost -t "home/living_room/abc123/led/command" -m
'{"state":"ON","brightness":128,"effect":"rainbow"}'
```

Breathing Effect (Blue)

Topic: `home/living_room/abc123/led/command`

```
{
  "state": "ON",
  "brightness": 150,
  "color": {
    "r": 0,
    "g": 0,
    "b": 255
  },
  "effect": "breathing"
}
```

mosquitto_pub command:

```
mosquitto_pub -h localhost -t "home/living_room/abc123/led/command" -m
'{"state":"ON","brightness":150,"color":{"r":0,"g":0,"b":255},"effect":"breathing"}'
```

Humidity Gauge Effect

Topic: `home/living_room/abc123/led/command`

```
{
  "state": "ON",
  "brightness": 180,
  "effect": "humidity_gauge"
}
```

mosquitto_pub command:

```
mosquitto_pub -h localhost -t "home/living_room/abc123/led/command" -m
'{"state":"ON","brightness":180,"effect":"humidity_gauge"}'
```

Temperature Gauge Effect

Topic: `home/living_room/abc123/led/command`

```
{
  "state": "ON",
  "brightness": 180,
  "effect": "temperature_gauge"
}
```

mosquitto_pub command:

```
mosquitto_pub -h localhost -t "home/living_room/abc123/led/command" -m
'{"state":"ON","brightness":180,"effect":"temperature_gauge"}'
```

Turn Off

Topic: `home/living_room/abc123/led/command`

```
{
  "state": "OFF"
}
```

mosquitto_pub command:

```
mosquitto_pub -h localhost -t "home/living_room/abc123/led/command" -m
'{"state":"OFF"}'
```

LED State Response

Topic: `home/living_room/abc123/led/state`

```
{
  "state": "ON",
  "brightness": 128,
  "color": {
    "r": 255,
    "g": 128,
    "b": 0
  },
  "effect": "rainbow"
}
```

Sensor State Messages

Complete State Payload

Topic: `home/living_room/abc123/state`
Published every 30 seconds or on significant change

```
{
  "temperature": 22.5,
  "humidity": 45.2,
  "rssi": -65,
  "uptime": 3600,
  "timestamp": 1707840000
}
```

Subscribe to sensor data:

```
mosquitto_sub -h localhost -t "home/living_room/abc123/state" -v
```

Availability Status

Topic: `home/living_room/abc123/status`
Retain: `true`

Online:

```
online
```

Offline (Last Will):

```
offline
```

Subscribe to availability:

```
mosquitto_sub -h localhost -t "home/living_room/abc123/status" -v
```

OTA Update Commands

OTA Update with SHA-256 Verification

Topic: `home/living_room/abc123/ota`

```
{
  "url": "http://192.168.1.100:8000/firmware/main.py",
  "sha256": "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855"
}
```

mosquitto_pub command:

```
mosquitto_pub -h localhost -t "home/living_room/abc123/ota" -m '{"url":"http://
192.168.1.100:8000/firmware/
main.py","sha256":"e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855"}'
```

OTA Update without Verification

Topic: `home/living_room/abc123/ota`

```
{
  "url": "http://192.168.1.100:8000/firmware/main.py"
}
```

mosquitto_pub command:

```
mosquitto_pub -h localhost -t "home/living_room/abc123/ota" -m '{"url":"http://
192.168.1.100:8000/firmware/main.py"}'
```

OTA Status Responses

Topic: `home/living_room/abc123/ota/status`

Downloading:

```
{
  "status": "downloading"
}
```

Success:

```
{
  "status": "success"
}
```

Failed:

```
{
  "status": "failed"
}
```

Error:

```
{
  "status": "error",
  "message": "Download failed: Connection timeout"
}
```

Preparing Firmware for OTA

```
# Calculate SHA-256 hash
sha256sum main.py

# Start HTTP server
cd firmware_directory
python3 -m http.server 8000

# Firmware will be available at:
# http://YOUR_IP:8000/main.py
```

Configuration Commands

Update MQTT Configuration

Topic: `home/living_room/abc123/config`

```
{
  "mqtt": {
    "broker": "192.168.1.100",
    "port": 1883,
    "username": "new_user",
    "password": "new_password"
  }
}
```

mosquitto_pub command:

```
mosquitto_pub -h localhost -t "home/living_room/abc123/config" -m '{"mqtt":
{"broker":"192.168.1.100","port":1883}}'
```

Update Sensor Configuration

Topic: `home/living_room/abc123/config`

```
{
  "sensor": {
    "read_interval": 5,
    "publish_interval": 20,
    "temp_offset": -0.5,
    "humidity_offset": 2.0
  }
}
```

mosquitto_pub command:

```
mosquitto_pub -h localhost -t "home/living_room/abc123/config" -m '{"sensor":
{"read_interval":5,"publish_interval":20,"temp_offset":-0.5}}'
```

Update LED Configuration

Topic: `home/living_room/abc123/config`

```
{
  "led": {
    "enabled": true,
    "brightness": 200,
    "effect": "rainbow"
  }
}
```

mosquitto_pub command:

```
mosquitto_pub -h localhost -t "home/living_room/abc123/config" -m '{"led":{"en-
abled":true,"brightness":200,"effect":"rainbow"}}'
```

Update Device Location

Topic: `home/living_room/abc123/config`

```
{
  "device": {
    "location": "bedroom"
  }
}
```

Note: Changing location will update the base MQTT topic after restart.

mosquitto_pub command:

```
mosquitto_pub -h localhost -t "home/living_room/abc123/config" -m '{"device":{"loca-
tion":"bedroom"}}'
```

General Commands

Restart Device

Topic: `home/living_room/abc123/command`

```
{
  "action": "restart"
}
```

mosquitto_pub command:

```
mosquitto_pub -h localhost -t "home/living_room/abc123/command" -m '{"ac-
tion":"restart"}'
```

Scan Wi-Fi Networks

Topic: `home/living_room/abc123/command`

```
{
  "action": "scan_wifi"
}
```

Response topic: `home/living_room/abc123/command/response`

```
{
  "networks": [
    {
      "ssid": "MyWiFi",
      "rssi": -45,
      "channel": 6,
      "authmode": 3,
      "hidden": false
    },
    {
      "ssid": "NeighborWiFi",
      "rssi": -78,
      "channel": 11,
      "authmode": 3,
      "hidden": false
    }
  ]
}
```

mosquitto_pub command:

```
mosquitto_pub -h localhost -t "home/living_room/abc123/command" -m '{"ac-
tion":"scan_wifi"}'

# Listen for response
mosquitto_sub -h localhost -t "home/living_room/abc123/command/response" -v
```

Testing MQTT Integration

Monitor All Topics

```
# Subscribe to all device topics
mosquitto_sub -h localhost -t "home/living_room/abc123/#" -v

# Subscribe to all Home Assistant discovery
mosquitto_sub -h localhost -t "homeassistant/#" -v

# Subscribe to everything (debug)
mosquitto_sub -h localhost -t "#" -v
```

Test Sequence

```
# 1. Monitor device status
mosquitto_sub -h localhost -t "home/living_room/abc123/status" -v &

# 2. Monitor sensor data
mosquitto_sub -h localhost -t "home/living_room/abc123/state" -v &

# 3. Turn LED on (red)
mosquitto_pub -h localhost -t "home/living_room/abc123/led/command" -m '{"state":"ON",
"brightness":255,"color":{"r":255,"g":0,"b":0},"effect":"solid"}'

# 4. Change to rainbow
mosquitto_pub -h localhost -t "home/living_room/abc123/led/command" -m '{"state":"ON",
"effect":"rainbow"}'

# 5. Turn LED off
mosquitto_pub -h localhost -t "home/living_room/abc123/led/command" -m '{"state":"OFF"
}'

# 6. Request device restart
mosquitto_pub -h localhost -t "home/living_room/abc123/command" -m '{"ac-
tion":"restart"}'
```

Python Examples

Publishing LED Commands

```python
import paho.mqtt.client as mqtt
import json

client = mqtt.Client()
client.connect("localhost", 1883, 60)

# Turn on with rainbow effect
payload = {
    "state": "ON",
    "brightness": 128,
    "effect": "rainbow"
}

client.publish(
    "home/living_room/abc123/led/command",
    json.dumps(payload)
)

client.disconnect()
```

Subscribing to Sensor Data

```python
import paho.mqtt.client as mqtt
import json

def on_message(client, userdata, msg):
    data = json.loads(msg.payload)
    print(f"Temperature: {data['temperature']}°C")
    print(f"Humidity: {data['humidity']}%")

client = mqtt.Client()
client.on_message = on_message

client.connect("localhost", 1883, 60)
client.subscribe("home/living_room/abc123/state")

client.loop_forever()
```

Triggering OTA Update

```python
import paho.mqtt.client as mqtt
import json
import hashlib

# Calculate SHA-256 of firmware
with open('main.py', 'rb') as f:
    sha256 = hashlib.sha256(f.read()).hexdigest()

# Publish OTA command
client = mqtt.Client()
client.connect("localhost", 1883, 60)

payload = {
    "url": "http://192.168.1.100:8000/main.py",
    "sha256": sha256
}

client.publish(
    "home/living_room/abc123/ota",
    json.dumps(payload)
)

client.disconnect()
```

Node-RED Examples

LED Control Flow

```
[
  {
    "id": "led_control",
    "type": "mqtt out",
    "topic": "home/living_room/abc123/led/command",
    "payload": "{\"state\":\"ON\",\"brightness\":255,\"color\":{\"r\":255,\"g\":
0,\"b\":0},\"effect\":\"solid\"}",
    "broker": "mqtt_broker"
  }
]
```

Sensor Data Dashboard

```
[
  {
    "id": "sensor_input",
    "type": "mqtt in",
    "topic": "home/living_room/abc123/state",
    "broker": "mqtt_broker"
  },
  {
    "id": "parse_json",
    "type": "json"
  },
  {
    "id": "temp_gauge",
    "type": "ui_gauge",
    "min": 0,
    "max": 40,
    "label": "Temperature"
  }
]
```

Home Assistant Automation Examples

Temperature Alert

```
automation:
  - alias: "High Temperature Alert"
    trigger:
      - platform: numeric_state
        entity_id: sensor.esp32_shtc3_abc123_temperature
        above: 30
    action:
      - service: notify.mobile_app
        data:
          message: "Temperature is high: {{ states('sensor.esp32_shtc3_abc123_temperat
ure') }}°C"
```

LED Color Based on Temperature

```yaml
automation:
  - alias: "LED Temperature Indicator"
    trigger:
      - platform: state
        entity_id: sensor.esp32_shtc3_abc123_temperature
    action:
      - service: light.turn_on
        target:
          entity_id: light.esp32_shtc3_abc123_led
        data:
          brightness: 200
          rgb_color: >
            {% set temp = states('sensor.esp32_shtc3_abc123_temperature') | float %}
            {% if temp < 20 %}
              [0, 0, 255]
            {% elif temp < 25 %}
              [0, 255, 0]
            {% else %}
              [255, 0, 0]
            {% endif %}
```

Turn Off LED at Night

```yaml
automation:
  - alias: "LED Night Mode"
    trigger:
      - platform: time
        at: "22:00:00"
    action:
      - service: light.turn_off
        target:
          entity_id: light.esp32_shtc3_abc123_led

  - alias: "LED Day Mode"
    trigger:
      - platform: time
        at: "07:00:00"
    action:
      - service: light.turn_on
        target:
          entity_id: light.esp32_shtc3_abc123_led
        data:
          brightness: 128
          effect: "solid"
```

Note: Replace `abc123` with your actual device ID and `living_room` with your configured location.