Requirements

- A SmartZone controller running software 5.2 or later, configured with one access point and one WLAN;
- A windows workstation with Python 3.7.4 and pip installed;
- Internet access.

Configuration Steps

1. Log in to SmartZone a create a Northbound Data Streaming Profile where:

   Name: is a name to identify the profile

   Server Host: is the ip address where the external MQTT broker resides

   Server Port: is 8883

   User: is the identity that will be used to connect to the external MQTT broker

   Password: is the password associated with the user identity

   System ID: is a name to identify the external application

   Data Type: allows the selection of which statistical data will be sent by SmartZone
2. Toggle the control **Enable Nortbound Data Streaming** to "ON" and click on **OK** to submit the form.
3. Navigate to https://mosquitto.org/files/binary/win32/ to download and install the mosquito 1.3.5 broker for Windows. Do not install the software as service. Under the directory where the software was installed, edit the **mosquitto.conf** configuration file. The configuration file needs to contain the following lines only:

   ```
   port 1883
   listener 8883
   psk_hint alpha
   psk_file alpha.txt
   tls_version tlsv1.2
   ```

   Now, under the same directory, create a text file named **alpha.txt** and add the following line:

   ```
   <user>:<password>
   ```

   where:

   `<user>` is the **User:** configured in the SmartZone Northbound Data Streaming profile and

   `<password>` is the **Password:** configured in the SmartZone Northbound Data Streaming profile, converted to hexadecimal.

4. Open a command line, go to the directory where mosquitto was installed and type in the following command to start the broker:

```
mosquitto -c mosquitto.conf -v
```

Return to your SmartZone instance and check the status of your profile. It should show as **Connected**.

5. Install the command line tools to decode the GPB messages using a .proto file templates. Navigate to https://github.com/protocolbuffers/protobuf/releases/tag/v3.10.0, then download and install the package protoc-3.10.0-win64.zip.
6. Download the SmartZone GPB .proto files from the Ruckus support site at https://support.ruckuswireless.com/software/2392-smartzone-5-2-0-0-699-ga-gpb-proto-google-protobuf-image-for-gpb-mqtt, unzip the files and copy them over to the same directory were **protoc.exe** was installed. Make sure to copy the google directory too.
7. Use the **protoc.exe** utility from the installed package to compile python classes from the .proto files. Go to the directory where all .proto files have been copied, and type in the following commands:

```
protoc -I=. --python_out=. .\ap_avc_all.proto
protoc -I=. --python_out=. .\ap_avc.proto
protoc -I=. --python_out=. .\ap_client.proto
protoc -I=. --python_out=. .\ap_hccd_report.proto
protoc -I=. --python_out=. .\ap_mesh.proto
protoc -I=. --python_out=. .\ap_peerlist.proto
protoc -I=. --python_out=. .\ap_report.proto
protoc -I=. --python_out=. .\ap_rogue.proto
protoc -I=. --python_out=. .\ap_status.proto
protoc -I=. --python_out=. .\ap_wired_client.proto
protoc -I=. --python_out=. .\commons.proto
protoc -I=. --python_out=. .\nanopb.proto
protoc -I=. --python_out=. .\ScgSessMgrPubIpc.proto
protoc -I=. --python_out=. .\sci-alarm.proto
protoc -I=. --python_out=. .\sci-configuration.proto
protoc -I=. --python_out=. .\sci-event.proto
protoc -I=. --python_out=. .\sci-message.proto
protoc -I=. --python_out=. .\sci-pci.proto
protoc -I=. --python_out=. .\sci-rogue.proto
protoc -I=. --python_out=. .\session_manager.proto
protoc -I=. --python_out=. .\simple-storage.proto
protoc -I=. --python_out=. .\switch_all.proto
protoc -I=. --python_out=. .\switches.proto
```

Every command will create a different python class ending in **_pb2.py.** You can disregard the messages

```
"No syntax specified for the proto file: xxxxx.proto. Please use 'syntax =
"proto2":' or 'syntax = "proto3":' to specify a syntax version. (Defaulted to
proto2 syntax.)
```

8. Copy all the **_pb2.py** files to the directory where your python scripts are located.
9. Open a command line and use the following command to install the **paho mqtt** library:

```
pip install paho-mqtt
```

10. Create a python script named **main.py** in the same directory as other scripts, including the following lines:

```
import paho.mqtt.client as mqtt #import the mqtt library
import sci_message_pb2 #import all SZ .proto classes compiled by the protoc tool

#read incoming messages
def on_message(client, userdata, message):
scim = sci_message_pb2.SciMessage().FromString(message.payload)
print(scim)
#    print(scim.apClient)
#    print(scim.apClient.ap)
#    print(scim.apClient.clients[0].clientMac)
#    print(scim.apReport)
#    print(scim.apWiredClient)
#    print(scim.apStatus)
#    print(scim.switchDetailMessage)
#    print(scim.switchConfigurationMessage)
#    print(scim.apRogue)

#log function
def on_log(client, userdata, level, buf):
      print("log: ",buf)

broker_address="localhost"

client = mqtt.Client("P1") #create new instance
client.on_message=on_message #attach function to message callback
#client.on_log=on_log

print("connecting to mosquitto broker")
client.connect(broker_address) #connect to mosquitto broker


print("Subscribing to sci-topic")
client.subscribe("sci-topic")

client.loop_forever() #starts a thread to the message buffers continuously
```

11. Use the following command to run the script:

```
python main.py
```

Ruckus solutions are part of CommScope's comprehensive portfolio for Enterprise environments (indoor and outdoor).

We encourage you to visit **commscope.com** to learn more about:

- Ruckus Wi-Fi Access Points
- Ruckus ICX switches
- SYSTIMAX and NETCONNECT: Structured cabling solutions (copper and fiber)
- imVision: Automated Infrastructure Management
- Era and OneCell in-building cellular solutions
- Our extensive experience about supporting PoE and IoT

COMMSCOPE®

RUCKUS®

commscope.com

Visit our website or contact your local CommScope representative for more information.