



Setting up OurWeather MQTT

January 2020

SwitchDoc Labs

Errata	3
What is the MQTT Channel for OurWeather?	4
What is MQTT?	4
Setting up an MQTT Broker (Mosquitto) on the Raspberry Pi	5
Setting up MQTT on OurWeather	5
Testing MQTT on OurWeather	6
What is the MQTT Format for OurWeather?	6

Errata

- First Version

What is the MQTT Channel for OurWeather?

OurWeather is a connected weather station containing 7 different sensors that will teach about electronics, software and the weather. It contains all the sensors and wires to make a fully functional, WiFi weather station. No software programming is required, it works right out of the box. It is an open source software kit.

The OurWeather MQTT channel is designed to send information in a lightweight fashion to MQTT brokers both locally on your network and across the Internet.

What is MQTT?



MQTT is a publish-subscribe based "light weight" messaging protocol for use on top of the TCP/IP protocol, such as the WiFi packets that we are using in this project. It is designed for connections with remote locations where a "small code footprint" is required or the network bandwidth is limited. There is also an excellent library available for MQTT for the Arduino IDE <https://github.com/knolleary/pubsubclient>. The publish-subscribe messaging pattern requires a message broker. The broker is responsible for distributing messages to interested clients based on the topic of a message.

Publish-subscribe is a pattern where senders of messages, called publishers (in this case our OurWeather ESP32 is the publisher), don't program the messages to be sent directly to subscribers, but instead characterize message payloads into classes without the specific knowledge of which subscribers the messages are sent to. Similarly, subscribers will only receive messages that are of interest without specific knowledge of which publishers there are. Mosquitto operates as the broker in this system and routes the published data to the appropriate subscribers.

You can think of MQTT as writing stories for a newspaper where you don't know who will be subscribing to the article.

Setting up an MQTT Broker (Mosquitto) on the Raspberry Pi

A good way of testing your MQTT system is to bring up a Mosquitto Broker on a Raspberry Pi. Follow the directions in this article:

<https://www.switchdoc.com/2018/02/tutorial-installing-and-testing-mosquitto-mqtt-on-raspberry-pi/>

Once you have tested Mosquitto, then set up MQTT on OurWeather

Setting up MQTT on OurWeather

You set up MQTT on OurWeather by using the <http://www.switchdoc.com/OurWeatherAdmin.html> page. Set up your IP number (as shown in the Assembly and Operations Manual) and then fill out the OurWeather MQTT Setup on the OurWeatherAdmin page. Note: This is a stateless form so you must fill out each field every time you change something.

OurWeather MQTT Setup V4.1

<input type="text" value="Admin Password"/>	Admin Password
<input checked="" type="checkbox"/> MQTT Enable	
<input type="text" value="MQTT Server URL"/>	MQTT Server URL
<input type="text" value="1883"/>	MQTT Port
<input type="text" value="500"/>	Send MQTT Every How Many Seconds

If successful, OurWeather will reboot.

Descriptions of the fields:

- **Admin Password** - your Admin password on OurWeather - "admin" is the default
- **MQTT Enable** - Switch for Enable and Disable MQTT - Turn on for Enable

- **MQTT Server URL** - IP address or domain name of your MQTT Broker
- **MQTT Port** - Port that your MQTT Server/Broker is on. 1883 is the typical port (Raspberry Pi Mosquitto for example)
- **Send MQTT Every How Many Seconds** - Send an MQTT packet to your broker every how many seconds. Currently rate limited in the OurWeather software at no faster than every 60 seconds. Anything less than that defaults to 60 seconds.

Then hit the Update MQTT Options to write the results to OurWeather. If successful, OurWeather will reboot and come up and start sending MQTT Packets.

If you are going to test MQTT on your Raspberry Pi, put the IP address of the Raspberry Pi into the MQTT Server URL field.

Testing MQTT on OurWeather

Open up a terminal on your Raspberry Pi running Mosquitto and type the following command:

```
mosquitto_sub -d -t OurWeather
```

The topic you are subscribing to is “OurWeather”.

After a while (depending on your Seconds field above), you should see something like this on your terminal window:

```
Client mosq-aCTWecreTff3K4WarH received PINGRESP
Client mosq-aCTWecreTff3K4WarH received PUBLISH (d0, q0, r0, m0, 'OurWeather', ... (328
bytes))
{"stationname": , "softwareversion": 053, "hardware": esp32, "Controllerboard": V2,
"connected": true ,"FullDataString":
"0.00,0.00,19.78,103115.00,505.48,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0,2020-01-
19 18:45:49,,1008,4,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,V:,WXLMB
,1,,0,,0,32.84,-4101"}
```

What is the MQTT Format for OurWeather?

The MQTT Packet (around 331 bytes) is a JSON data structure consisting of some header information about the OurWeather unit that is sending the packet and then full contents of the “FullDataString”.

Here is an example:

```
{"stationname": , "softwareversion": 053, "hardware": esp32, "Controllerboard": V2,  
"connected": true , "FullDataString":  
"0.00,0.00,19.78,103178.00,500.43,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0,2020-01-  
19 17:17:34,,1008,4,4.14,232.00,12.58,2.00,5.11,1.10,0.00,0.00,0.00,0.00,0.00,0.00,V:,WXLMG  
,1,,,0,,,0,32.33,-4501"}
```

You can see all the definitions of each field in the “Advanced Usage Manual” located on the OurWeather product page: <https://shop.switchdoc.com/products/ourweather-complete-weather-kit>