# MQTT Instructions

## The Protocol

The modules I have designed for model train automation use a protocol called [MQTT](). In MQTT, devices can subscribe to certain topics and can publish messages under an unlimited number of topics. The topics used by JMRI begin with "/trains/track/".

All inputs are considered sensors and all output are considered turnouts.

Topics used to change a turnout are in the format of "/trains/track/turnout/<number>".

Topics used to report a change in a sensor are in the format of "/trains/track/sensor/ <number>".

Messages contain a topic and a payload. Subscriptions only contain a topic.

A module that has outputs, subscribes to messages for its outputs using topics corresponding to the output (turnout) numbers, then listens for messages with those topics. Those incoming messages contain a payload of either "THROWN" or "CLOSED".

A module that has inputs (sensors), publishes messages when the inputs change using the sensor numbers assigned to it in the sketch. These messages have a payload of either "ACTIVE" or "INACTIVE".

## The Broker

To allow the MQTT modules to communicate with JMRI, you need an MQTT broker to distribute the published messages to all the modules and to JMRI. The most common MQTT broker is called mosquitto. I prefer to run Mosquitto on a Linux system because of the reliability and stability of the Linux operating system.

A fairly nice Linux system can be built on a mini PC, such as the MinisForum Z83-F. I bought [this one](). It has USB jacks, HDMI, ethernet, and other stuff.



The mini PC comes with Windows already installed. To install Linux, follow [these instructions]().

Once Linux installs for the first time, it may tell you it needs to be updated.  Go ahead and complete the updates, then when it's no longer bugging you about updates, install mosquitto.

Run the following commands in a terminal window:

```
sudo apt-get update
sudo apt-get install mosquitto
sudo apt-get install mosquitto-clients
```

Mosquitto is installed as a service and should start automatically after installation.  You can check if mosquitto is running by running the following command.

```
ps -ef | grep mosquitto
```

You should see two lines, one with grep and one that has something like "`/usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf`"

Some instructions tell you to set up your router/firewall to allow port 1883 access.  If you want to keep your model railroad system secure, I would advise against doing this.  All the MQTT messages need to stay within your own network.

Once everything is working, you can monitor MQTT activity from a command window with the following command.

```
mosquitto_sub -v -t "#"
```

It seems to list everything, past and present.  If you want to clear out old messages, try this:

```
sudo service mosquitto stop
```

```
sudo rm /var/lib/mosquitto/mosquitto.db
```

```
sudo service mosquitto start
```

If you do this with JMRI running, you will probably need to stop and restart it.

If you're having a problem reading the text in the xterm window from across the room try starting a new xterm with a big font.

```
xterm -font 12x24
```

If you don't like that font, you can try to find something you like with `xlsfonts | less.`

# Interfacing With JMRI

Linux is a good platform for JMRI also so this would be a good place to install it.  You can find plenty of instructions on the JMRI website.  Remember to install the correct version of Java as per their instructions.

After JMRI starts, you can add the MQTT protocol.  On the main **PanelPro** window, go to **Edit→Preferences**.  Under **System manufacturer**, select **MQTT**, then under **System connection**, select **MQTT Connection**.

Under **Settings**, for **IP Address/Host Name**, enter the IP address of the Linux system you just built.  To get the IP address, type `hostname -I`.

For **Connection Prefix**, enter **M**, and for **Connection Name**, enter **MQTT**.

Click **Save** (lower left corner).

## Testing a Turnout Message

On the main **PanelPro** window, go to **Tools**→**Tables**→**Turnouts**. Click the **Add…** button at the bottom.

In the **Add New Turnout** dialog box, make sure **MQTT** is selected as the **System Connection**. Enter a **Hardware Address**, such as **101**, and a **User Name**, such as **MT101**. This is just for testing; you can delete it later. Click **Create**.

If you want to save this turnout, go to **File**→**Store**→**Store Configurations And Panels To File…**, choose and file name and click **Store**.

Open a terminal window and run the command to monitor MQTT as mentioned in the previous section. Position the terminal window so you can access the **Turnout** table at the same time.

Click the button under **State**. You should see a message in the terminal window corresponding to the state shown in the **Turnouts** window.

## Testing a Sensor Message

On the main **PanelPro** window, go to **Tools**→**Tables**→**Turnouts**. Click the **Add…** button at the bottom.

In the **Add New Sensor** dialog box, make sure **MQTT** is selected as the **System Connection**. Enter a **Hardware Address**, such as 2**01**, and a **User Name**, such as **MS201**. This is just for testing; you can delete it later. Click **Create**.

If you want to save this sensor, go to **File**→**Store**→**Store Configurations And Panels To File…**, choose and file name and click **Store**.

Open a terminal window if not already open and run the following command. Position the terminal window so you can access the **Sensor** table at the same time.

```
mosquitto_pub -d -t "/trains/track/sensor/201" -m "ACTIVE"
```

Including the -d option allows you to see the various steps in the process.

Check the status under **State**. The state should have changed to **Active**. You can send "INACTIVE" and change the state to Inactive.