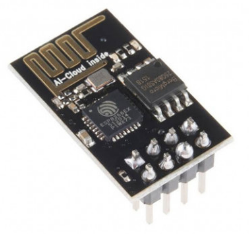
Getting started with mqTrains IO Board

mqTrains OVERVIEW

ESP-01S microprocessors are an inexpensive option for controlling model railroad accessories such as turnouts, sensors and signals. They connect quite easily to JMRI using the ESP's WiFi interface and the MQTT messaging protocol. An ESP-01S can be purchased from online stores for about a couple of bucks each.

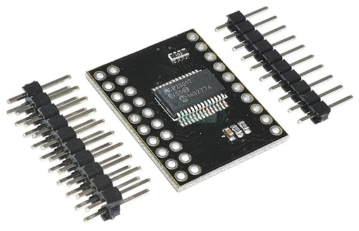


mqTrains sketches have been developed or are in development to handle some common model train requirements such as turnout control, signalling and occupancy detection. Visit mqTrains.com for more information.

ESP devices only use the 2.4 GHz WiFi band. Some common domestic WiFi routers are limited to 16 or 32 concurrent sessions. Check your router specifications if you intend to have a high number of these devices.

mqTrains IO Board

The mqTrainsIO sketch is designed to use MCP23017 Input/Output (IO) modules for IO additional pins. The sketch supports up to six IO modules, each providing 16 I/O pins for peripherals so that's a total of 96 IO pins from one ESP-01S. Each pin can be configured for use as an input for a sensor or as an output to control compatible turnouts, lights or other devices.



LET'S GET STARTED

These are the main steps to get started with mqTrains IO Board. More complex features are covered in the mqTrains User Reference.

1. Install the MQTT Server Software

Linux and Pi users:

We recommend using a Raspberry Pi for the MQTT Server Software. The Pi Zero W is quite adequate and inexpensive. Follow the vendor instructions to set up the Raspberry Pi. To install Mosquitto Server, use the commands:

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

Mosquitto version 2 (since December 2020) uses tighter security rules. Create this config file:

```
sudo nano
/etc/mosquitto/conf.d/mosquitto.conf
```

Insert these two lines and then press ctrl-o <enter> and ctrl-x

```
listener 1883
allow_anonymous true
```

and restart mosquitto with

```
sudo service mosquitto restart
```

Windows users:

Mosquitto can also be installed on most Windows computers. Refer to the Mosquitto website for details. If you install on Windows, you will most likely need to grant Firewall permission. Instructions for granting permission with the Windows Defender Firewall are shown at the end of this document. If you have a Firewall with non-Microsoft security software, refer to the vendor instructions.

Mac users:

Refer to the Mosquitto website and/or the Mac App Store for installation details.

✶ At the time of printing, standard Linux installs of mosquitto are mostly version 1.5 or 1.6 and the Windows download is version 2.0.9. These are all adequate.

2. Install mqTrains on your ESP-01S

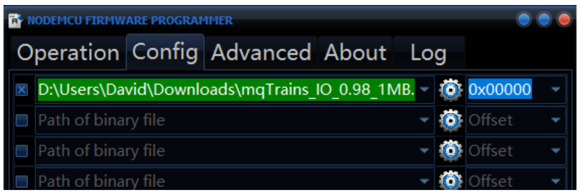
Download the mqTrains binary file from here:
mqTrains.com/downloads/

Windows users: Download the nodemcu-flasher program from the appropriate link here: x64 version (most common) or x32 version (for old 32-bit computers).

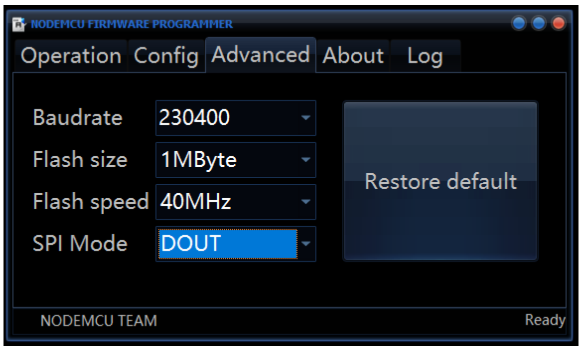
The ESP-01S requires an adapter to connect to a computer. We recommend the one shown here. Point the ESP-01S towards the USB connector.

Run the installer (double click the exe file downloaded), select the Config tab, click on the wheel in the first row, then navigate to and select the binary file you downloaded.

Select the Advanced tab and select DOUT for SPI Mode:

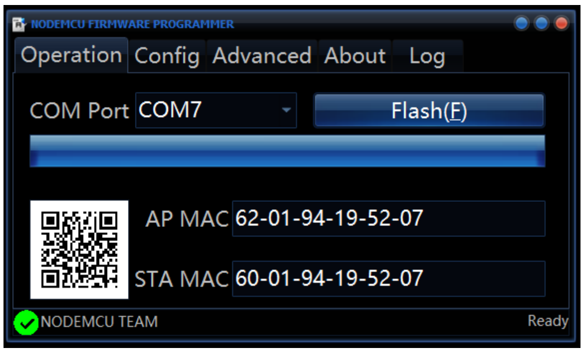


D:\Users\David\Downloads\mqTrains_IO_0.98_1MB. 0x00000
Path of binary file Offset
Path of binary file Offset
Path of binary file Offset



Baudrate 230400
Flash size 1MByte
Flash speed 40MHz
SPI Mode DOUT

NODEMCU TEAM Ready



COM Port COM7
AP MAC 62-01-94-19-52-07
STA MAC 60-01-94-19-52-07
NODEMCU TEAM Ready

On the Operation tab, the correct COM Port will most likely be selected for you, change if need be, then click on Flash to start the upload. The blue bar will show the upload progress which takes about 3 minutes. Wait for the green tick and "Ready" message at the bottom of the screen.

Linux and Pi users:

Run these commands:

```
sudo apt-get update
sudo apt-get install python python-pip
sudo pip install esptool
```

Then, while in the folder with the .bin file, run (all in one line and with the correct serial port, ttyUSB0 shown here) :

```
esptool.py --chip esp8266 --port
/dev/ttyUSB0 --baud 460800 --before
default_reset --after hard_reset
write_flash 0x0 mqTrains_IO_x.xx_1MB.bin
```

3. Wiring your ESP-01S

Connect a 3.3 Vdc power supply, as shown below to power the ESP-01S and other modules.

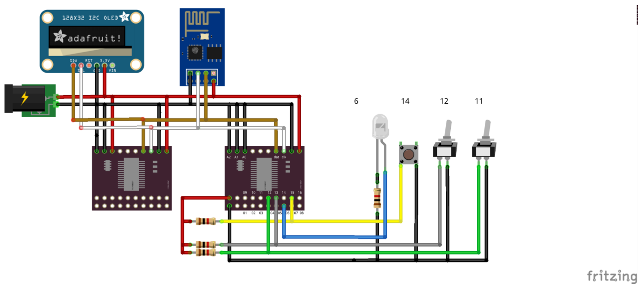


Fig 1. Two MCP23017 modules with ESP-01S and an optional OLED display.

Connect as many IO modules as you require, up to a maximum of 6. Take care to connect to the correct pins. The above diagram shows the IO module's pin connections as viewed from the side that has the integrated circuit (chip).

+3.3 Vdc is shown in red.
Ground is shown in black.
I^2C CLK is shown in white (GPIO2 on ESP).
I^2C DAT is shown in ochre (GPIO0).

Each IO module has 3 address pins at the top left of the boards in the picture above. These are labelled, left to right, A2, A1 and A0.

Connect these to either Vcc or Gnd to give a binary 1 (Vcc) or 0 (Gnd). Each module must have a unique address. Start with all 3 connected to black (Gnd) to give address 0 as shown with the module on the right in the diagram above. The module on the left in the diagram above has red (Vcc) connected to the A0 pin so will have address 1.

Connect whatever accessories you require. Switches, push buttons or occupancy detectors are examples of sensor inputs. We recommend using a 1 kOhm pullup resistor on each input as shown above.

LEDs can be used as outputs. Each LED should have an appropriate resistor.

An OLED display can be added to see the IP address, the serial number, topic and the last MQTT message received. This is optional. If you choose to include it, connect the +3.3 Vdc, Ground, CLK and DAT connections as you have done with the IO modules.

4. Configuring your ESP-01S

Using a smartphone or computer, look in your available WiFi networks list for the mqTrains Access Point (AP) network service started by the ESP01. The default name is "**mqTrains-1001**" with the password "**mqtrains**".

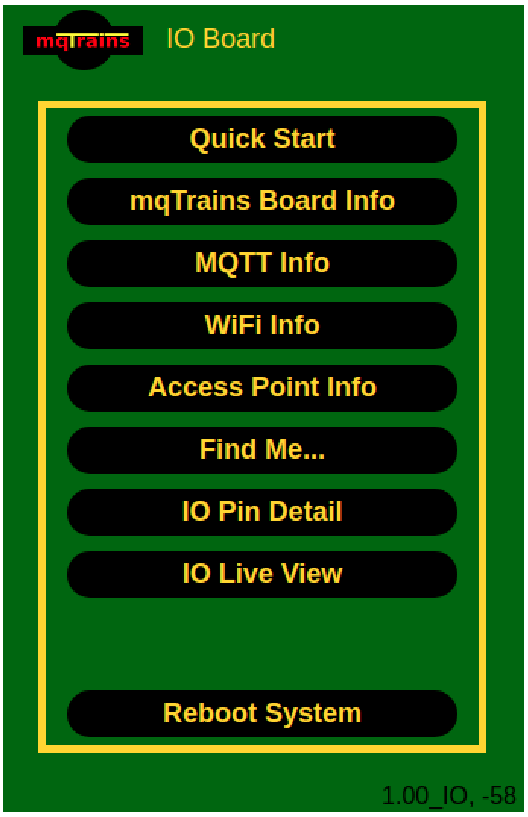
Once connected to the ESP's AP, use a web browser to connect to the mqTrains configuration page using http://2.2.2.1.

You can use these QR codes to connect to the default Access Point and web page.

(The 2nd QR code for the web page one won't work if your app needs the internet to decode, like Google Lens)

WiFi mqTrains-1001: Web Page 2.2.2.1:

Select the Quick Start option from the main menu and change each of these values to suit your needs.



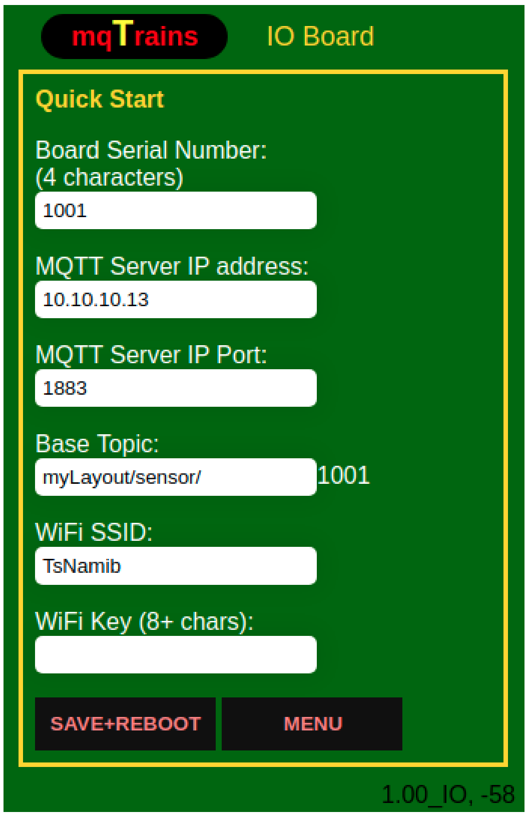
Set a 4 digit Serial Number for the board. If you have more than one mqTrains board, each should have a unique serial number.

This will be used as part of the MQTT messaging. We recommend not leaving the default value, since the next new one will also be 1001. Some use 1000s for sensor boards and 2000s for turnout boards and others use two numbers for a location on the layout and two more for a serial number at that place, so why not change it to 1002?

Set the IP address of the computer on which your MQTT Server (Broker) is installed and the port number it uses. 1883 is the default port.

Set the base portion of the MQTT topic. The board number and individual I/O pin numbers will be automatically appended.

For example, using the default base topic, the full MQTT topic for I/O pin number /03 on board number 1002 will be myLayout/sensor/1002/03



Quick Start

Board Serial Number: (4 characters): 1001

MQTT Server IP address: 10.10.10.13

MQTT Server IP Port: 1883

Base Topic: myLayout/sensor/ 1001

WiFi SSID: TsNamib

WiFi Key (8+ chars):

1.00_IO, -58

If you have a mix of inputs and outputs, you can put the wildcard '+' symbol instead of 'sensor'. 'sensor' will be used in the topic for inputs but you can use 'turnout' or 'light' in topics for outputs.

Set the WiFi SSID and Key to connect to your desired WiFi service. Keep in mind that the WiFi service will need to provide you access to the MQTT Server who's IP address you entered earlier.

Press SAVE+REBOOT to save all these values to the ESP01's internal storage and restart it using those values.

Using a utility such as MQTT Explorer or an MQTT subscribe line command, look for messages for topics beginning with the Base Topic. Expect to see a message like "10.10.10.101 is here!" This is the IP Address of the ESP01 on your WiFi network. Use this IP address in your browser to connect to the configuration page to complete the set up or to make changes. The AP will no longer be needed and will shut down once the ESP01 is able to post a message to the MQTT Server.

5. Configure the I/O pins

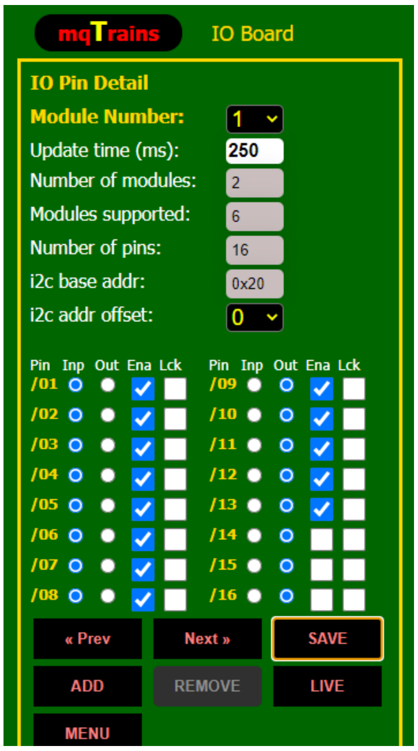
For each MCP23017 module, set the I^{2}C address offset you gave when wiring the address pins.

The pin labels on the module are A0 to A7 and B0 to B7. The numbers mqTrains assigned to these pins are 01 to 08 for the 'A' pins and 09 to 16 for the 'B' pins on the first module.

Subsequent modules will take the next block of 16 numbers, e.g. the second module will have pins numbered 17 to 32. If you have the maximum of 6 supported modules, pin B7 on the last module will have number 96.

For each pin, set whether it is to be used as an input (sensor) or an output (e.g. a turnout or light). The *Enabled* checkbox (ENA) must be checked for pins being used and should be unchecked for unused pins. No messages are processed for disabled pins.

Once I/O pins are configured, press "**SAVE**".

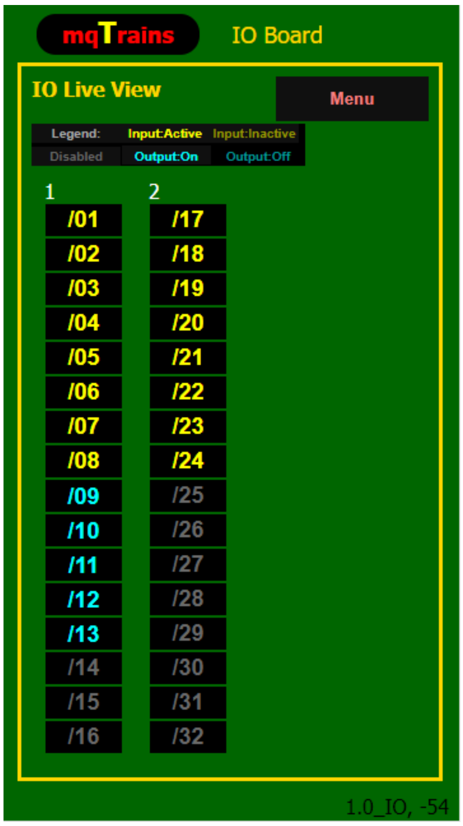


IO Pin Detail

Module Number:	1
Update time (ms):	250
Number of modules:	2
Modules supported:	6
Number of pins:	16
i2c base addr:	0x20
i2c addr offset:	0

Pin	Inp	Out	Ena	Lck	Pin	Inp	Out	Ena	Lck
/01	◉	○	✓	☐	/09	○	◉	✓	☐
/02	◉	○	✓	☐	/10	○	◉	✓	☐
/03	◉	○	✓	☐	/11	○	◉	✓	☐
/04	◉	○	✓	☐	/12	○	◉	✓	☐
/05	◉	○	✓	☐	/13	○	◉	✓	☐
/06	◉	○	✓	☐	/14	○	◉	☐	☐
/07	◉	○	✓	☐	/15	○	◉	☐	☐
/08	◉	○	✓	☐	/16	○	◉	☐	☐

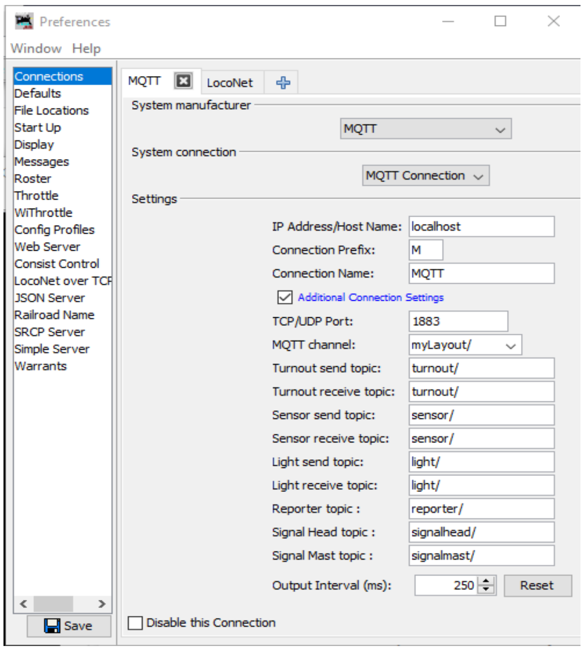
Pin states are monitored using the Live View page. You can also toggle outputs by tapping/clicking on the pin number. This is useful for testing or local use but the change is not published to MQTT.


mqTrains
IO Board
IO Live View
Menu
Legend: Input:Active Input:Inactive
Disabled Output:On Output:Off
1.0_IO, -54

1	2
/01	/17
/02	/18
/03	/19
/04	/20
/05	/21
/06	/22
/07	/23
/08	/24
/09	/25
/10	/26
/11	/27
/12	/28
/13	/29
/14	/30
/15	/31
/16	/32

Set JMRI to talk with your ESP01

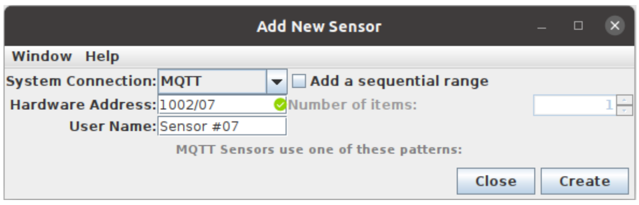
In JMRI, using Edit -> Preferences -> Connections, create an MQTT connection with the settings as shown, and especially make sure you specify the same host on which you have installed the MQTT Server as you provided to the ESP. You can specify the host name, or the IP address as you did on the ESP01. If the MQTT server is running on the same computer as JMRI, you can also use `localhost` as shown.



System manufacturer	MQTT
System connection	MQTT Connection
IP Address/Host Name:	localhost
Connection Prefix:	M
Connection Name:	MQTT
☑ Additional Connection Settings	
TCP/UDP Port:	1883
MQTT channel:	myLayout/
Turnout send topic:	turnout/
Turnout receive topic:	turnout/
Sensor send topic:	sensor/
Sensor receive topic:	sensor/
Light send topic:	light/
Light receive topic:	light/
Reporter topic :	reporter/
Signal Head topic :	signalhead/
Signal Mast topic :	signalmast/
Output Interval (ms):	250
☐ Disable this Connection	

*JMRI **Connections** (version 4.21.2)*

Create MQTT devices using the board serial number and the 2 digit pin number for the hardware address, for example, the 7th pin on the first IO module for board number **`1002`** would have a hardware address of **`1002/07`** as shown below. (With a maximum of 6 modules and 16 pins per module, mqTrainsIO uses two digits for the possible range of pins from 01 to 96).



System Connection:	MQTT
☐ Add a sequential range	
Hardware Address:	1002/07
Number of items:	1
User Name:	Sensor #07

MQTT Sensors use one of these patterns:

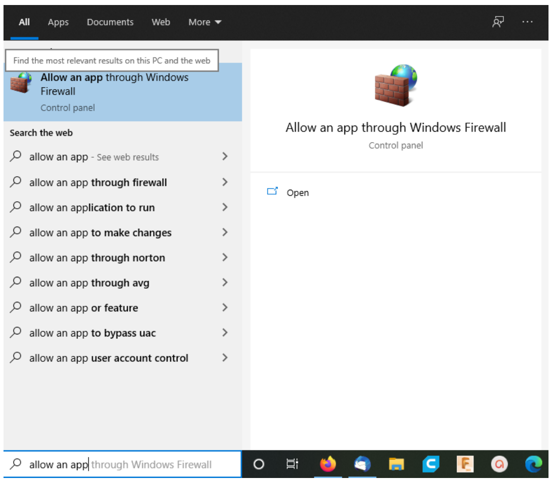
***Add New Sensor** in JMRI*

With the sensor created, and defined as an input on the `1002` mqTrains board, when the MCP23017 pin changes from High to Low, the sensor in JMRI will go **ACTIVE**. So any push button between the pin and ground will activate the JMRI sensor when pressed, and make the sensor INACTIVE when the push button is released.

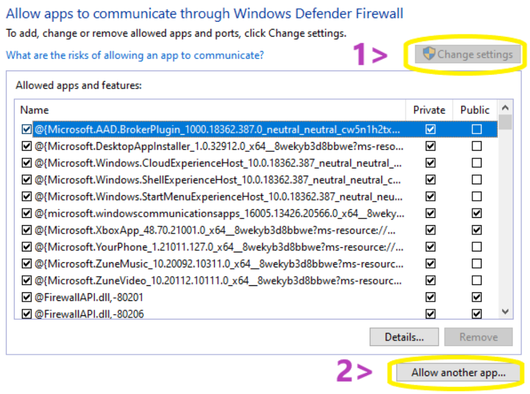
Burning through the Windows Firewall

If you install MQTT Server Software (mosquitto) on a Windows computer, such as your JMRI machine, you may find that the ESP nodes cannot connect to the MQTT Server due to the Windows Defender Firewall blocking WiFi traffic coming into the computer. This section describes how to alter the Firewall to allow MQTT messages to pass through.

Type "allow an app" into the Windows search bar. The "Allow an app through the Windows Firewall" will appear.


All
Apps
Documents
Web
More
Find the most relevant results on this PC and the web
Allow an app through Windows Firewall
Control panel
Search the web
allow an app - See web results
allow an app through firewall
allow an app to make changes
allow an application to run
allow an app through norton
allow an app through avg
allow an app or feature
allow an app to bypass uac
allow an app user account control
Allow an app through Windows Firewall
Control panel
Open
allow an app through Windows Firewall

Click on this and then, on "Change Settings" and "Allow another app".



Allow apps to communicate through Windows Defender Firewall

To add, change or remove allowed apps and ports, click Change settings.

What are the risks of allowing an app to communicate?

1> Change settings

Allowed apps and features:

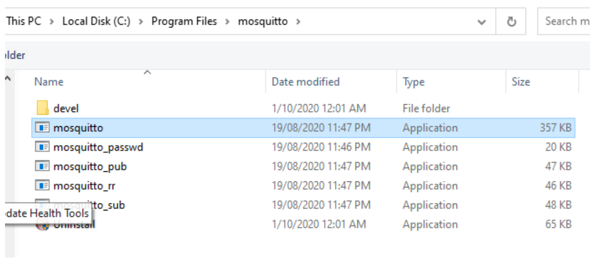
Name	Private	Public
☑ @{Microsoft.AAD.BrokerPlugin_1000.18362.387.0_neutral_neutral_cw5n1h2tx...	☑	☐
☑ @{Microsoft.DesktopAppInstaller_1.0.32912.0_x64__8wekyb3d8bbwe?ms-reso...	☑	☐
☑ @{Microsoft.Windows.CloudExperienceHost_10.0.18362.387_neutral_neutral_...	☑	☐
☑ @{Microsoft.Windows.ShellExperienceHost_10.0.18362.387_neutral_neutral_c...	☑	☐
☑ @{Microsoft.Windows.StartMenuExperienceHost_10.0.18362.387_neutral_neu...	☑	☐
☑ @{microsoft.windowscommunicationsapps_16005.13426.20566.0_x64__8weky...	☑	☑
☑ @{Microsoft.XboxApp_48.70.21001.0_x64__8wekyb3d8bbwe?ms-resource://...	☑	☑
☑ @{Microsoft.YourPhone_1.21011.127.0_x64__8wekyb3d8bbwe?ms-resource://...	☑	☐
☑ @{Microsoft.ZuneMusic_10.20092.10311.0_x64__8wekyb3d8bbwe?ms-resourc...	☑	☐
☑ @{Microsoft.ZuneVideo_10.20112.10111.0_x64__8wekyb3d8bbwe?ms-resourc...	☑	☐
☑ @FirewallAPI.dll,-80201	☑	☑
☑ @FirewallAPI.dll,-80206	☑	☑

Details... Remove

2> Allow another app...

With Browse, select mosquitto.exe from C:\Program Files\mosquitto

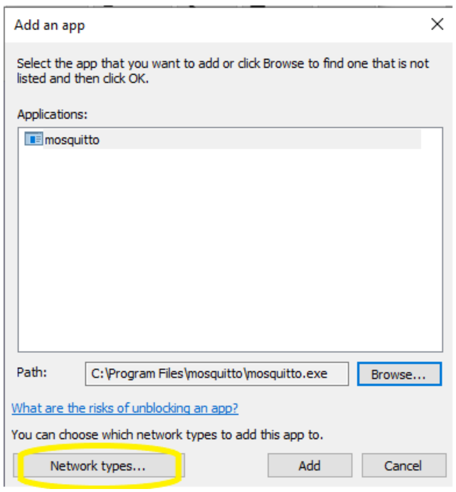
(32-bit machines would have X86 in that folder name)



This PC › Local Disk (C:) › Program Files › mosquitto ›

Name	Date modified	Type	Size
devel	1/10/2020 12:01 AM	File folder	
mosquitto	19/08/2020 11:47 PM	Application	357 KB
mosquitto_passwd	19/08/2020 11:46 PM	Application	20 KB
mosquitto_pub	19/08/2020 11:47 PM	Application	47 KB
mosquitto_rr	19/08/2020 11:47 PM	Application	46 KB
mosquitto_sub	19/08/2020 11:47 PM	Application	48 KB
uninstall	1/10/2020 12:01 AM	Application	65 KB

And click on "Open", then click on "Network types".



Add an app

Select the app that you want to add or click Browse to find one that is not listed and then click OK.

Applications:

mosquitto

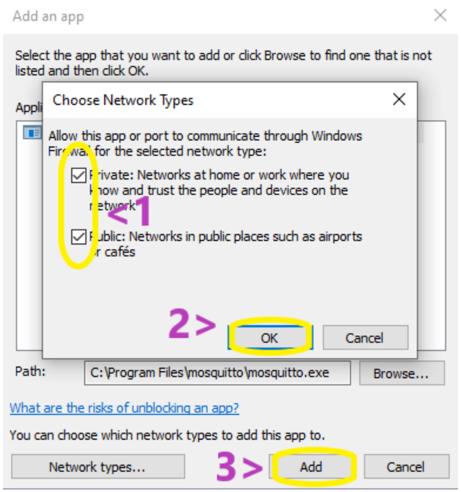
Path: C:\Program Files\mosquitto\mosquitto.exe Browse...

What are the risks of unblocking an app?

You can choose which network types to add this app to.

Network types... Add Cancel

Depending on how your WiFi network is configured, you may require both Private and Public options so enable both anyway. Then click on "OK" and "Add".



Choose Network Types

Allow this app or port to communicate through Windows Firewall for the selected network type:

☑ Private: Networks at home or work where you know and trust the people and devices on the network <1

☑ Public: Networks in public places such as airports or cafés

2> OK Cancel

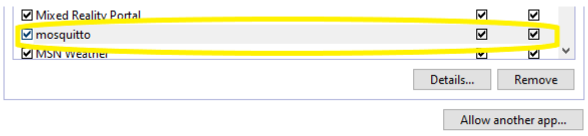
Path: C:\Program Files\mosquitto\mosquitto.exe Browse...

What are the risks of unblocking an app?

You can choose which network types to add this app to.

Network types... 3> Add Cancel

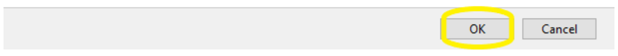
You will then see mosquitto added to the list. Now press OK to finish.



☑ Mixed Reality Portal	☑	☑
☑ mosquitto	☑	☑
☑ MSN Weather	☑	☑

Details... Remove

Allow another app...



Try mqTrains from the command line

The mqTrains web pages allow you to see I/O pin states and change output states. On a command line, with the mosquitto tools installed, you can see the messages being published for those changes while mqTrains is connected to the MQTT Server:

```
        prompt> mosquitto_sub -v -h localhost -t "myLayout/#"
or in Windows
       "c:\Program Files\mosquitto\mosquitto_sub.exe" -v -h localhost -t "myLayout/#"

        -v means verbose - show the full message (topic and payload).
        -t means topic
        -h means the MQTT Server hostname or IP address
        -m means message, aka payload.
```

You can also publish messages (THROWN or CLOSED, ON or OFF, ACTIVE or INACTIVE) to the MQTT Server and see the state on the web page or in JMRI (using the topic you have chosen):

```
        prompt> mosquitto_pub -h localhost -r -t "myLayout/light/1002/02" -m "ON"
          -r means retain the message in the Server.
```

Credits

mqTrains uses, without modifying any of their respective code as available on github:

ArduinoJSON by Bernoit Blanchon
Arduino Queue by Einar Arnason
Adafruit MCP23017 Arduino Library
Adafruit PWM Servo Driver Library
Adafruit NeoPixel
Adafruit SSD1306
PubSubClient by Nick O'Leary
LittleFS file system from earlephilhowe
ESPAsyncTCP and ESPAsyncWebServer from me-no-dev