mqTrains: Inexpensive Layout Control with WiFi using MQTT

by Speed Muller

(with help from David and Joel)

mqTrains - Inexpensive Layout Control with WiFi and MQTT

What are you going to see today?

- **mqTrains** - what is it, why and how
- Some inexpensive components
- A quick overview of MQTT
- Using mqTrains with JMRI
- Using mqTrains without JMRI

…by the 3 amiqToes!

(In no particular order)

Speed Muller, Dallas Texas (representing the Helix and below)

Joel Davidson, Austin Texas (representing Texas and beyond)

David McMorran (representing all of Down Undah)

*…and sneaking **Brad Anderson** in here, doing testing in the background…*

What is mqTrains ?

What is mqTrains ?

Started as Firmware for the Espressif ESP8266 microprocessors!

Primarily for ESP-01 or ESP-01S devices (*where footprint is fixed*).

Also works on other ESP8266 devices, but not ESP32 yet.

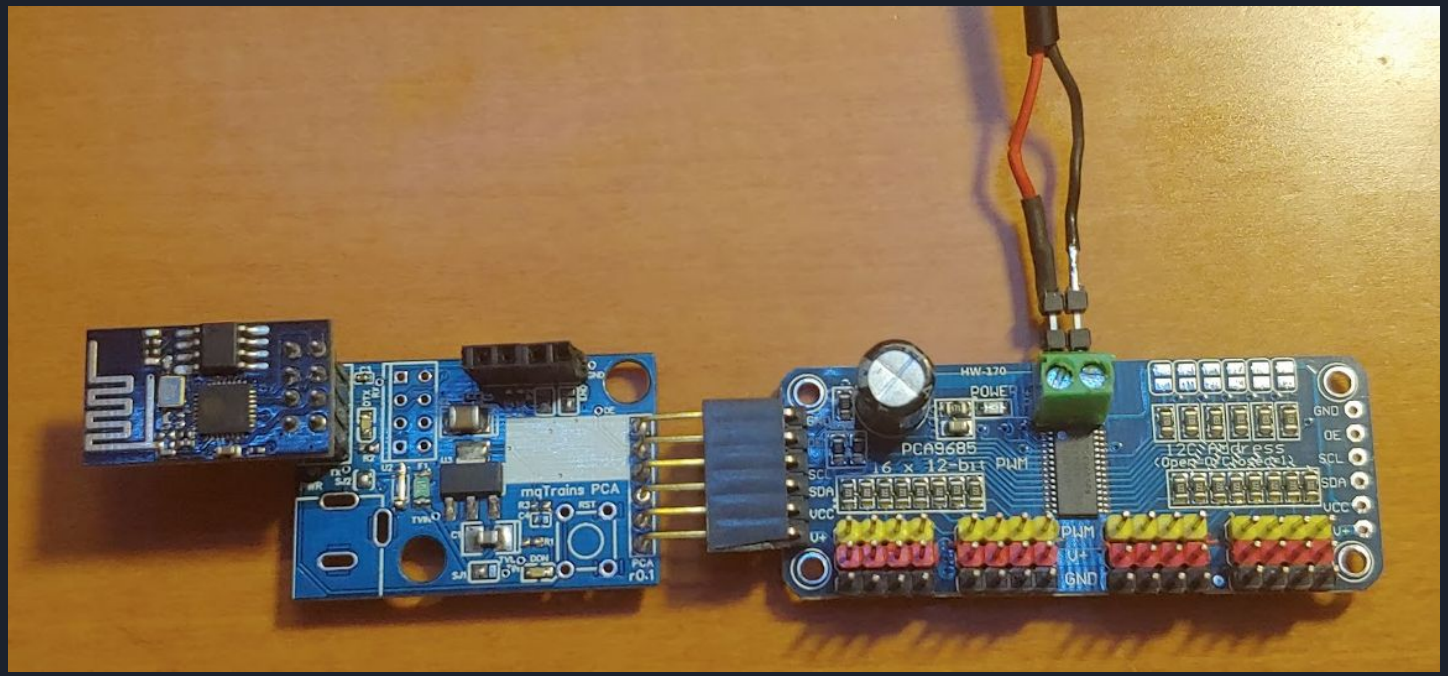
Hardware available everywhere: Amazon, AliExpress, Ebay and interface board designs now shared for you to build your own!

It can drive servo motors, can drive solenoid turnout motors,

control outputs like lights and signals **and** can monitor sensors.

Free to download, the code will become Open Source too

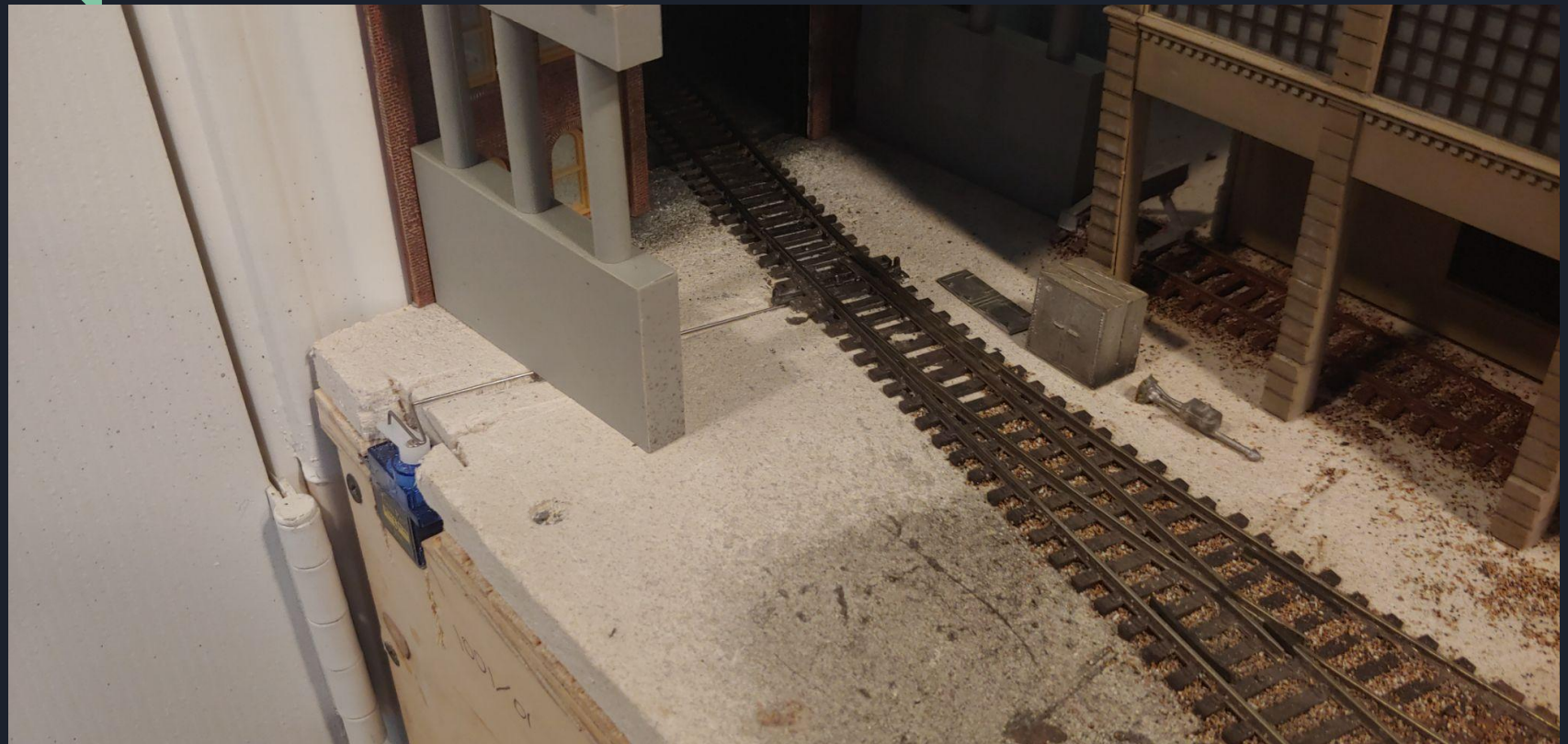
What is mqTrains ?



What is mqTrains ?

What is mqTrains ?

What is mqTrains ?



…but, why?

There are many inexpensive electronic components readily available…

Finding out **which** components to get is **simple** in most cases

Working out how to put them **together** is a bit **harder**

But, **writing code** for a device is **not for everyone**

The objective behind mqTrains is to make this technology **available to everyone** by having the programming done for you and ONLY showing you how to put things together

The motto: Less wires, less time, less cost!

…but, why?

There are many inexpensive electronic components readily available…

Finding out **which** components to get is **simple** in most cases

Working out how to put them **together** is a bit **harder**

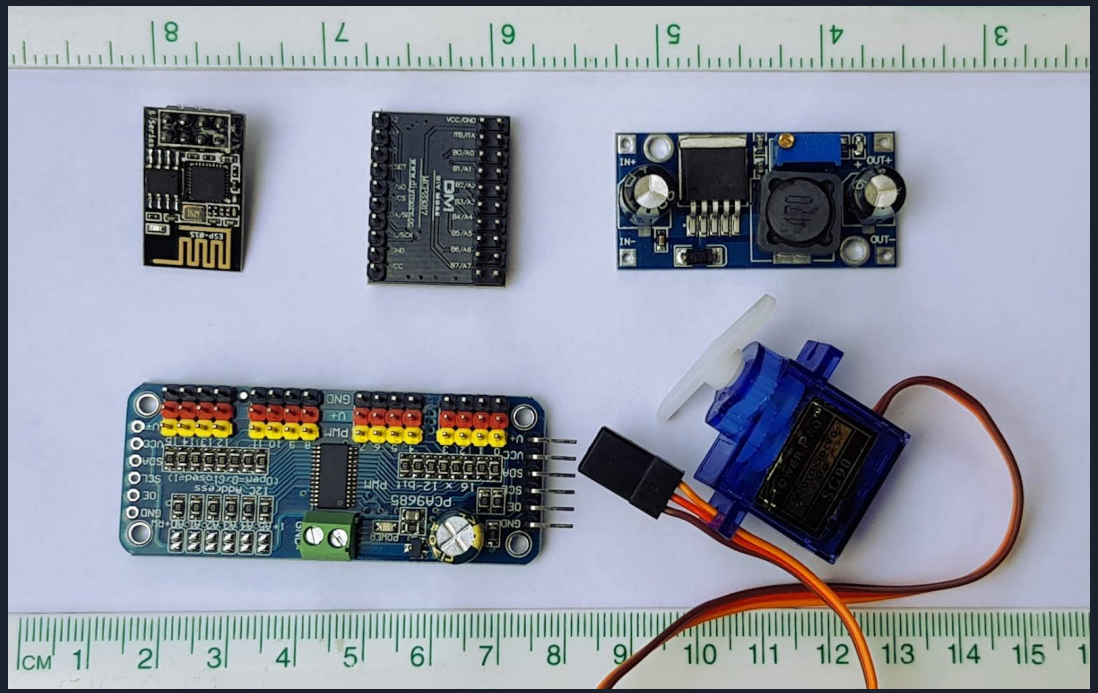
But, **writing code** for a device is **not for everyone**

The objective behind mqTrains is to make this technology **available to everyone** by having the programming done for you and ONLY showing you how to put things together

The motto: Less wires, less time, less cost!

(Speed also has a 3-foot rule: the “fix” to the problem should be within arms reach)

Inexpensive parts?



Approximate costs

ESP-01S	- $1.00
MCP23017	- $2.00
PCA9685	- $3.00
Buck Converter	- $0.50
9G Servo Motor	- $1.20

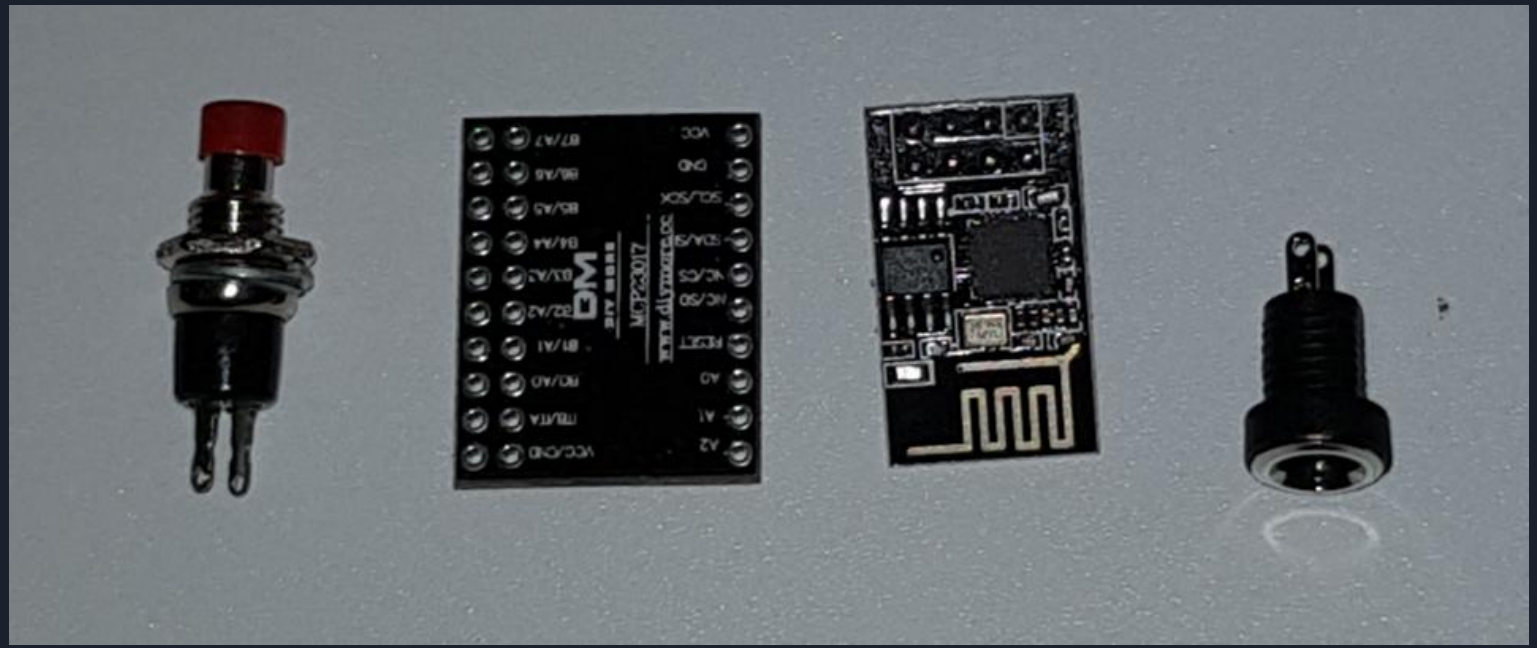
(and this can control 16 servos)

*1 + 3 + 0.50 + (16 * 1.20) = $23.70*

mqTrains

IOs...

Push-button, MCP23017, ESP-01 and an LED (16 I/Os)



mqTrains

WiFi? All you need is 802.11n

- Not Internet, just WiFi!
- The team tested 22 nodes (ESPs) with both a
  - WRT54G (802.11b/g circa 2002), and
  - DIR-655 (802.11b/g/n circa 2007)

  Sending 20,000 MQTT messages in 24 hours
- Cost of a WiFi router that can do 802.11n?

  Linksys N600 @ $29.99?

(The TxNamib currently runs 42 nodes on a Ubiquiti UniFi AC Pro, planning to use 30 more!)

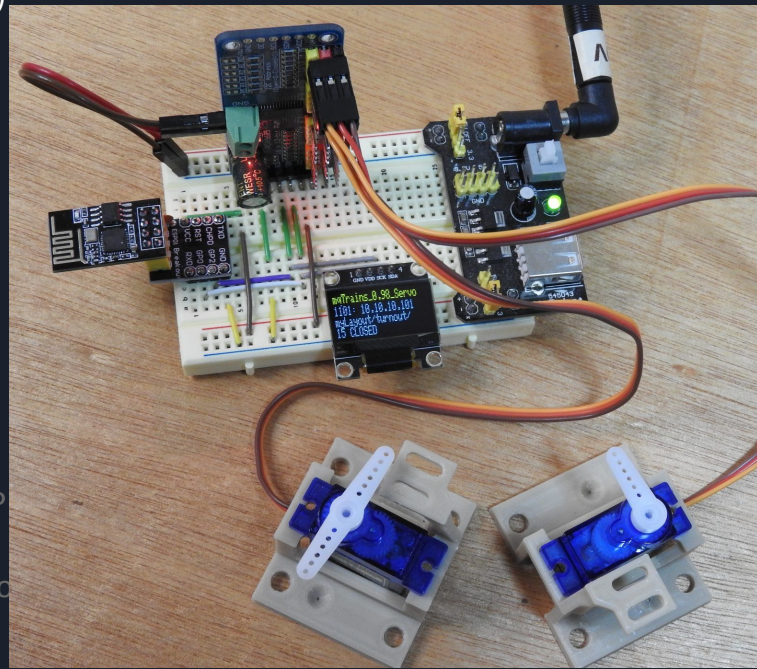
How to set it up! Very simple!!!

- Connect your Windows, Linux, Mac or Raspberry Pi computing device to a WiFi access point / router
  - WiFi or Wired
- Install mosquitto or any other MQTT Server (broker)
  - Linux or Pi: sudo apt-get install mosquitto
  - Mac: brew install mosquitto
  - Windows: https://mosquitto.org/download/
    - Configure Windows firewall to allow mosquitto
- Find your mosquitto computer's IP address
  - Linux, Mac or Pi: ifconfig
  - Windows: ipconfig

How to set it up! Very simple!!!

Nodes:

- Connect all the parts (*also need to program the ESP-01, instructions provided*)
- Power up!
- Connect with wireless device (phone, tablet, laptop or WiFi enabled PC)
    - SSID: mqtrains-0001
    - Key: mqtrains
- Use browser to go to 2.2.2.1
- Select Quick Start
- Change Serial Number (####) to something unique, like 0002
- Set MQTT Server IP address (port is by default 1883)
- Enter your WiFi SSID and Key (the latter will never be shown again)
- Save and Reboot
- If you have an OLED LCD or Serial Port connected, or you are monitoring your
  you will see the IP address of the module on your network. Else use the DHCP
  router's config page.
- Go to the new IP address with browser and configure the servos, I/O, signals c



How to set it up! Very simple!!!

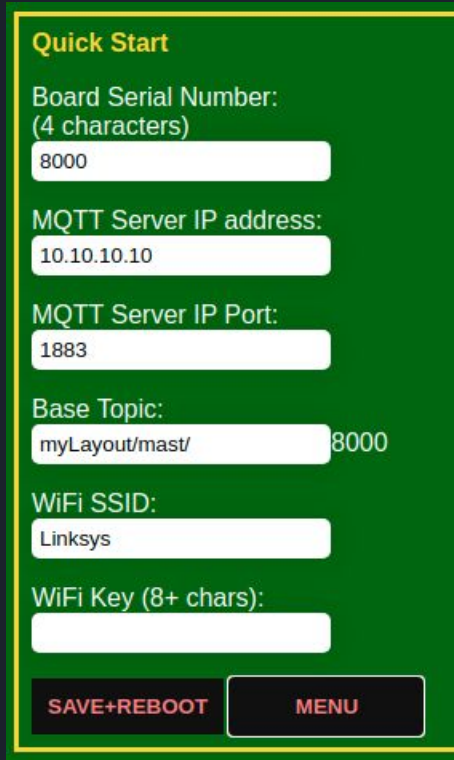
Nodes:

- Connect all the parts (also need to program the ESP-01, instructions provided)
- Power up!
- Connect with wireless device (phone, tablet, laptop or WiFi enabled PC)
    - SSID: mqtrains-0001
    - Key: mqtrains
- Use browser to go to 2.2.2.1
- Select Quick Start
- Change Serial Number (####) to something unique, like 0002
- Set MQTT Server IP address (port is by default 1883)
- Enter your WiFi SSID and Key (the latter will never be shown again)
- Save and Reboot
- If you have an OLED LCD or Serial Port connected, or you are monitoring your MQTT Server, you will see the IP address of the module on your network. Else use the DHCP table in the router's config page.
- Go to the new IP address with browser and configure the servos, I/O, signals or Logiqs

How to set it up! Very simple!!!

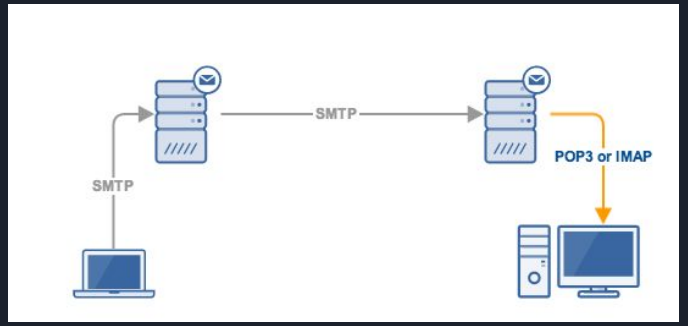
Nodes:

- Connect all the parts
- Power up!
- Connect with wireless device (phone, tablet, laptop or WiFi enabled desktop PC)
    - SSID: mqtrains-0001
    - Key: mqtrains
- Use browser to go to 2.2.2.1
- Select Quick Start
- Change Serial Number (####) to something unique, like 0002
- Set MQTT Server IP address (port is by default 1883)
- Enter your WiFi SSID and Key (the latter will never be shown again)
- Save and Reboot
- If you have an OLED LCD or Serial Port connected, or you are monitoring your MQTT Server, you will see the IP address of the module on your network. Else use the DHCP table in the router's config page.
- Go to the new IP address with browser and configure the servos, I/O, signal masts or Logiqs



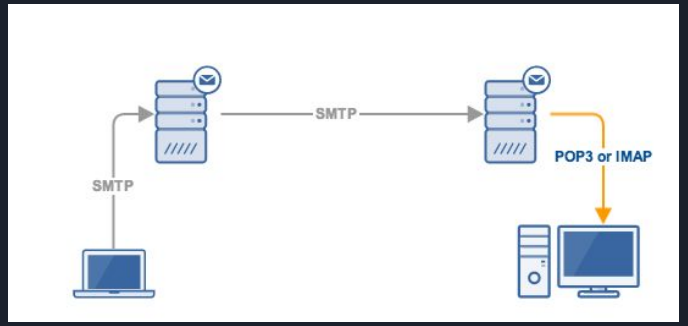
What is MQTT ?

You know these:

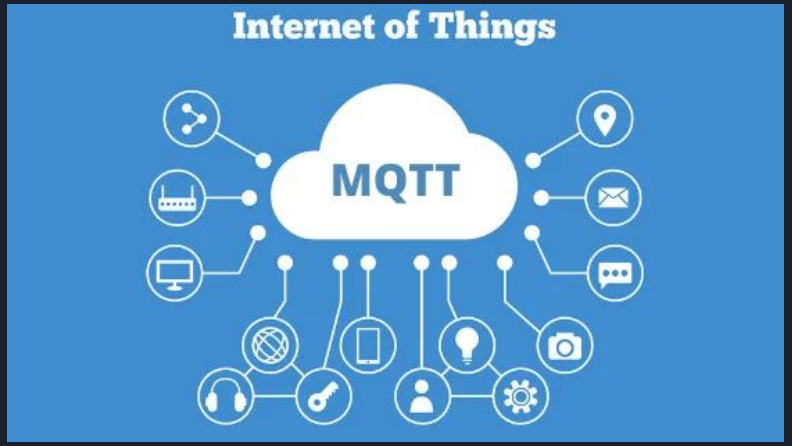


What is MQTT ?

You know these:



Just add another one:



MQTT - the messenger

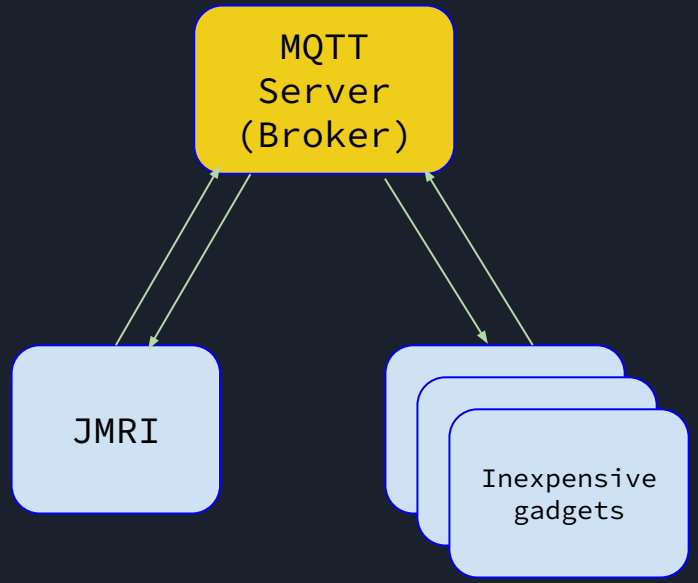
MQTT is a messaging protocol (often used in home automation systems among other uses, IOT sounds familiar)

It uses a Broker (now called a Server, a piece of software) through which all messages are sent

Clients connect to the Server to send (Publish) and receive messages (as Subscribed to)

Publishing is posting a message to the server

Subscribing is saying, give all messages I care about, to me.



Messages travel using **TCP/IP** (WiFi or wired)

MQTT - the messenger

MQTT is a messaging protocol (often used in home automation systems among other uses, IOT sounds familiar)

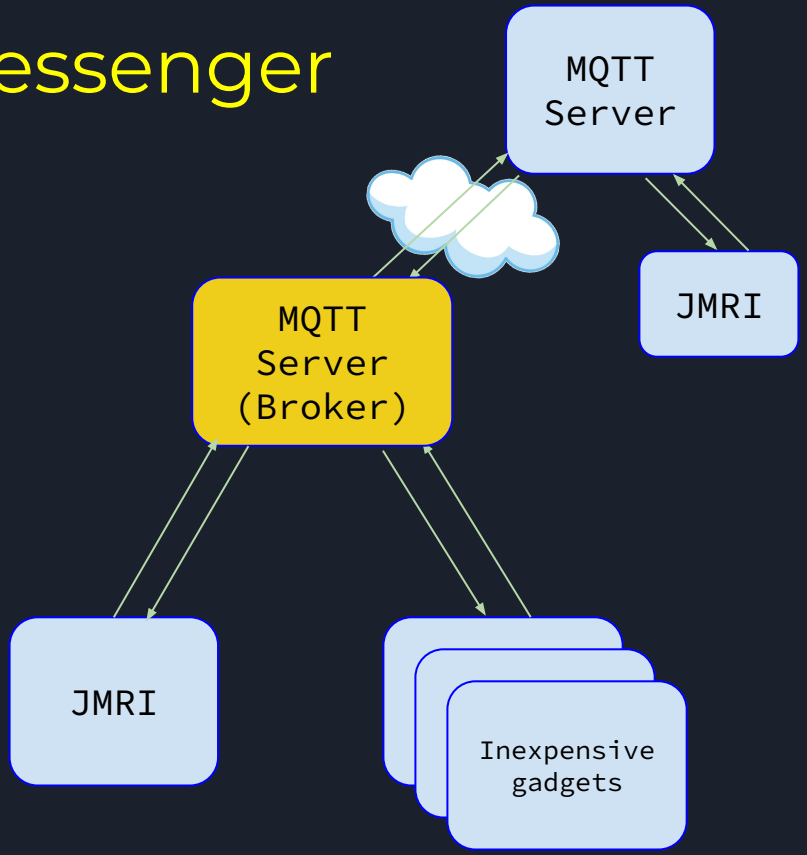
It uses a Broker (now called a Server, a piece of software) through which all messages are sent

Clients connect to the Server to send (Publish) and receive messages (as Subscribed to)

Publishing is posting a message to the server

Subscribing is saying, give all messages I care about, to me.

The MQTT Server can be "bridged" to another: Now TxNamib's remote dispatcher gets the same messages



Messages travel using **TCP/IP** (WiFi or wired)

MQTT - the messages

Messages look like this for a Turnout using MQTT in JMRI:

Topic: myLayout/turnout/0201/01 JMRI system name: MT0201/01

Payload: **Thrown** The 'State' being set in the JMRI Turnouts Table

Signal Mast messages:

Topic: myLayout/mast/8000/01 JMRI system name: IF$mqm:basic:...($8000/01)

Payload: **Approach; Lit; Unheld** The 'State', Lit and Held being set in the JMRI Signal Masts Table

I/O, or Sensors and Lights, same thing.

mqTrains with JMRI

Two, maybe three, awesome NMRAx clinics with more detailed steps by David McMorran, 2020/09/26, 2021/04/24 and 2021/06/11

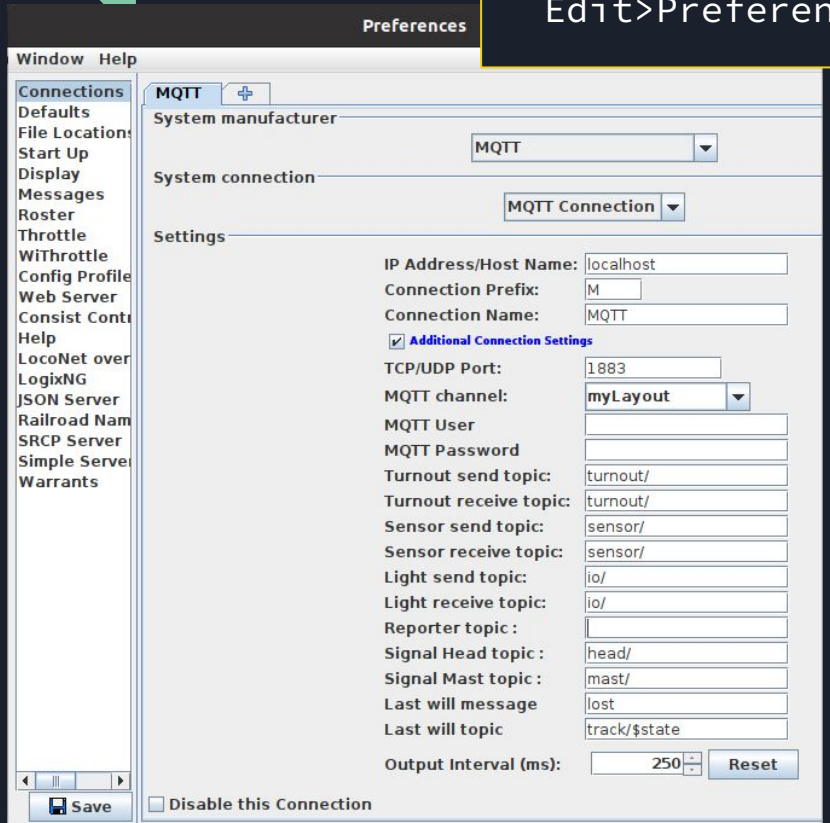
Subscribe to NMRA.org on YouTube and search for NMRAx on the dates shown

Or go to…

https://www.TxNamib.com/nmrax-links

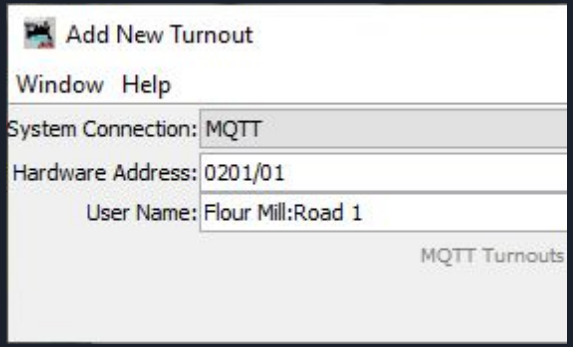
Setting up MQTT in JMRI

Edit>Preferences>Connections

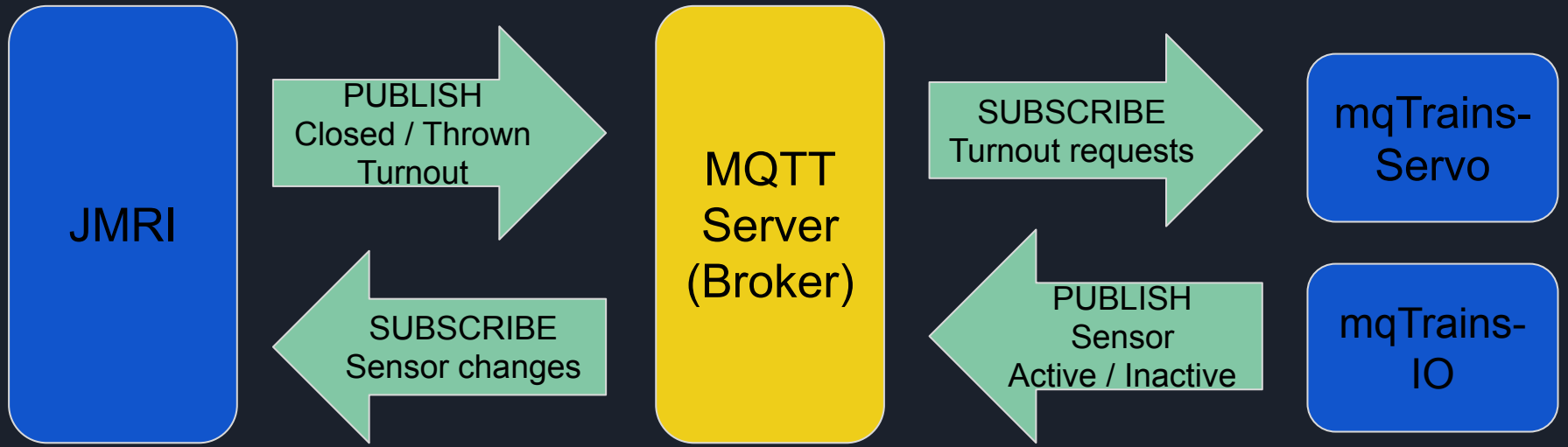


Topic: myLayout/turnout/0201/01

Tools>Tables>Turnouts>Add

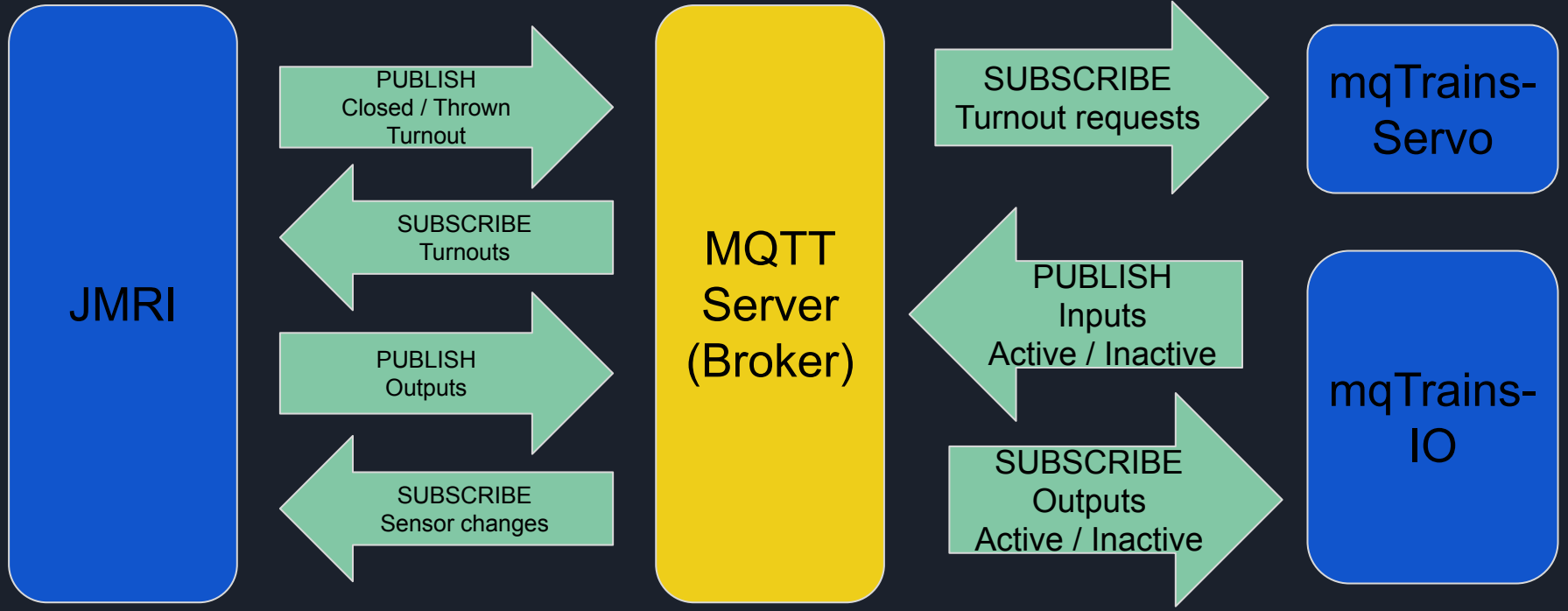


MQTT messages with JMRI



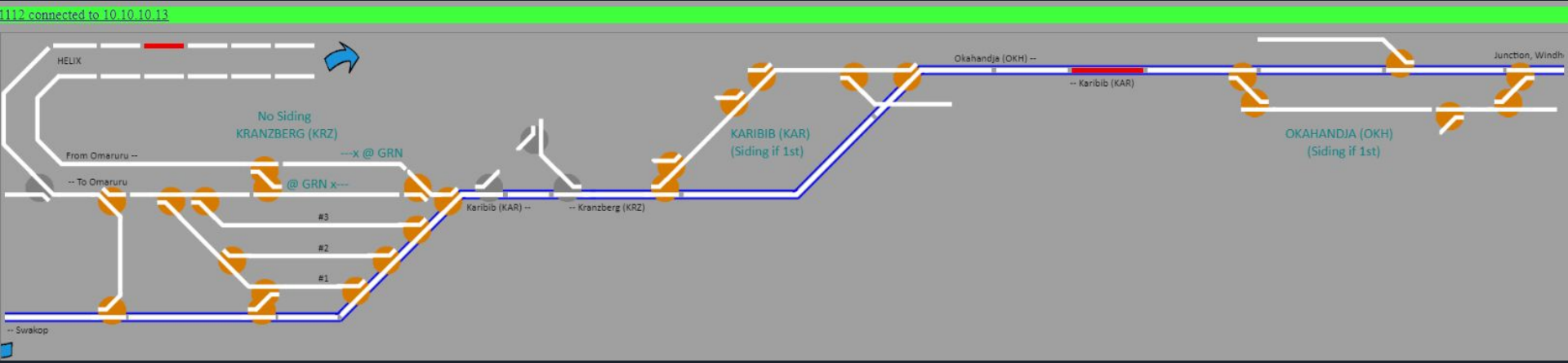
Messages transit over TCP/IP (WiFi or wired)

MQTT messages with JMRI



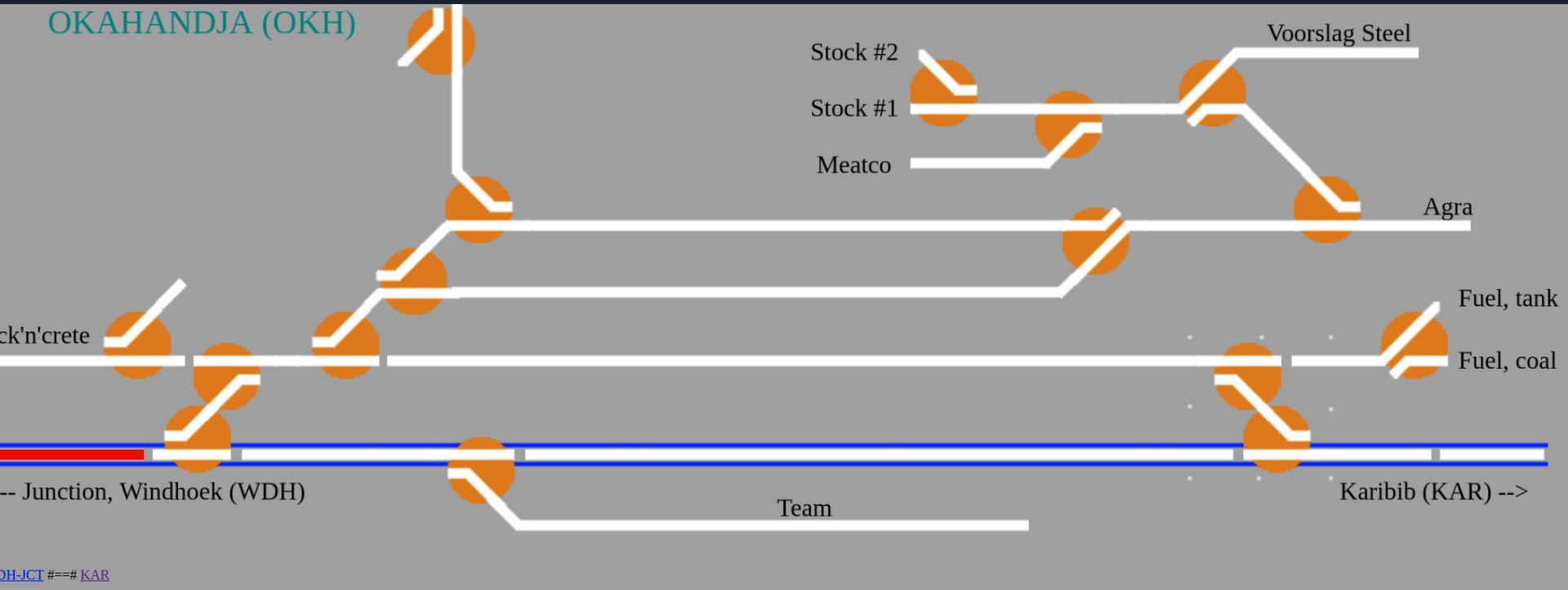
mqTrains w/Javascript (like in a web page)

mqTrains with Javascript (dispatcher view)



mqTrains with Javascript (local control)

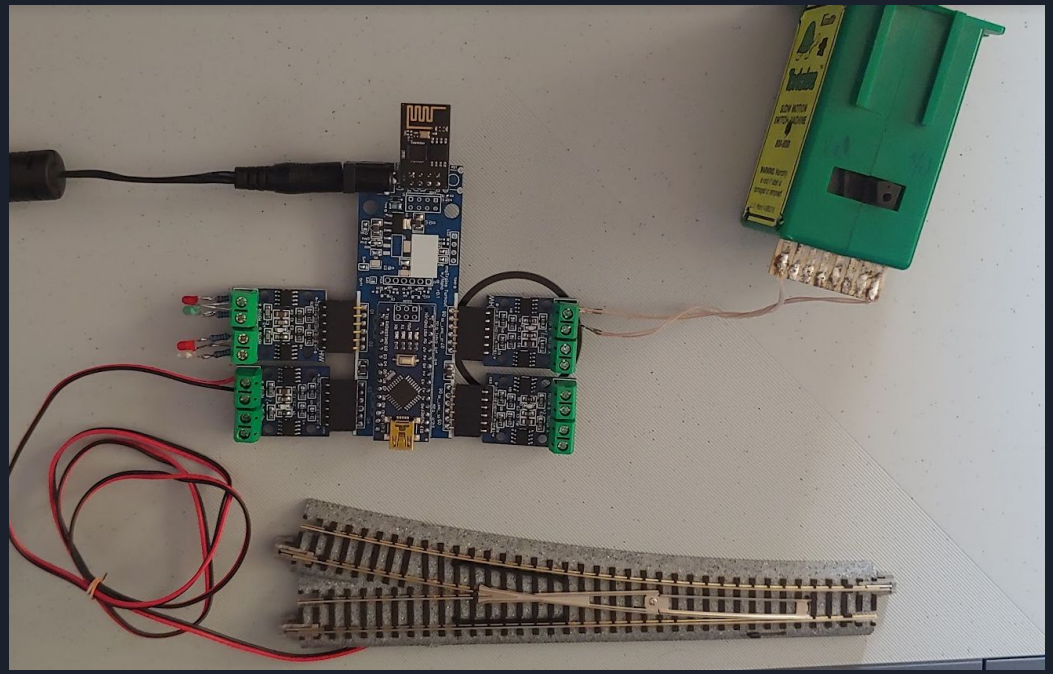
mqTrains with Javascript (any good browser)



OH-JCT #==# KAR

A few more examples...

mqTrains Turnout:
Solenoids and Stall motors (4 x L9110S boards, 8 turnouts)

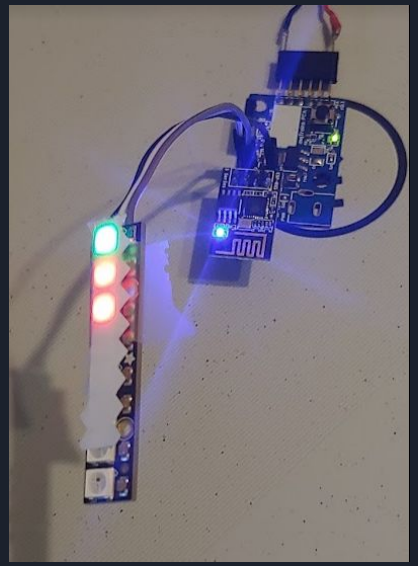


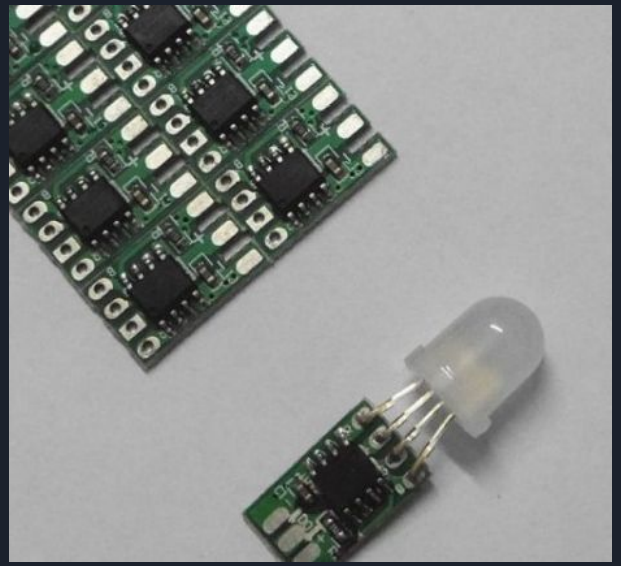
A few more examples...

mqTrains Signal Masts:

Just an ESP-01(s) and Pixels:

- NeoPixels,
- WS2811/12/12b or
- SK6812





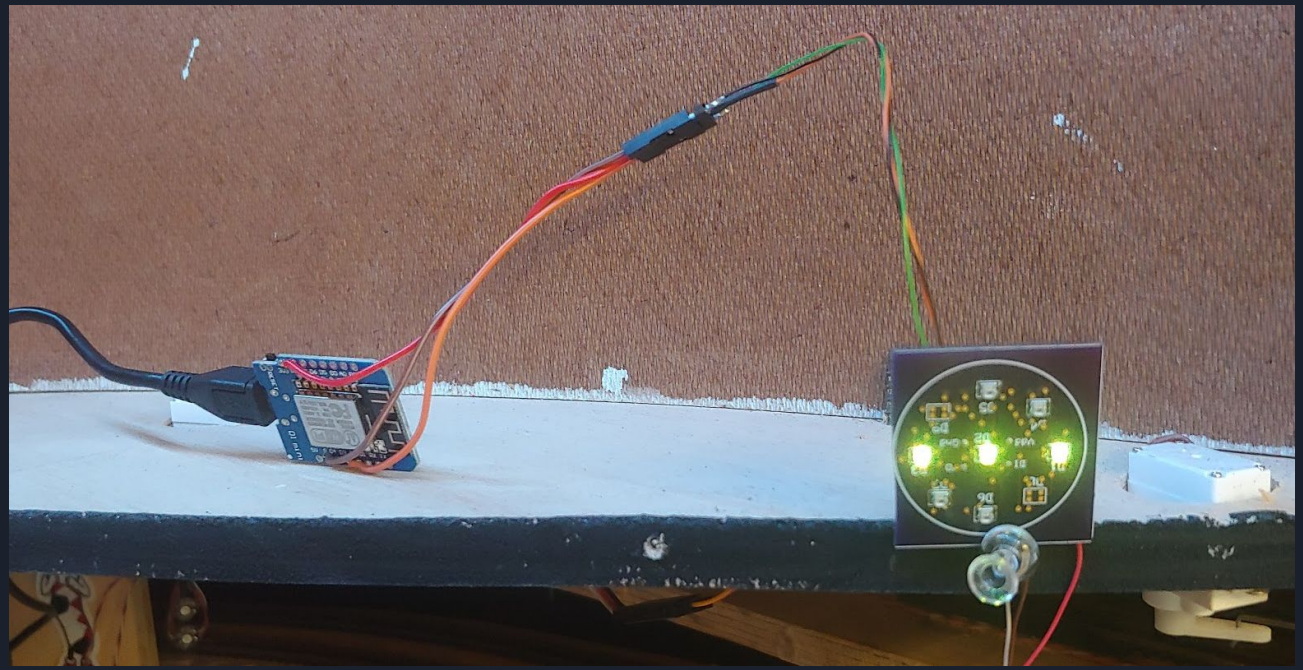
Where do I start?

Buy an ESP8266 Starter Kit and then mqTrains.com!

mqTrains

Any questions ?

The only bad question is the one you did not ask!

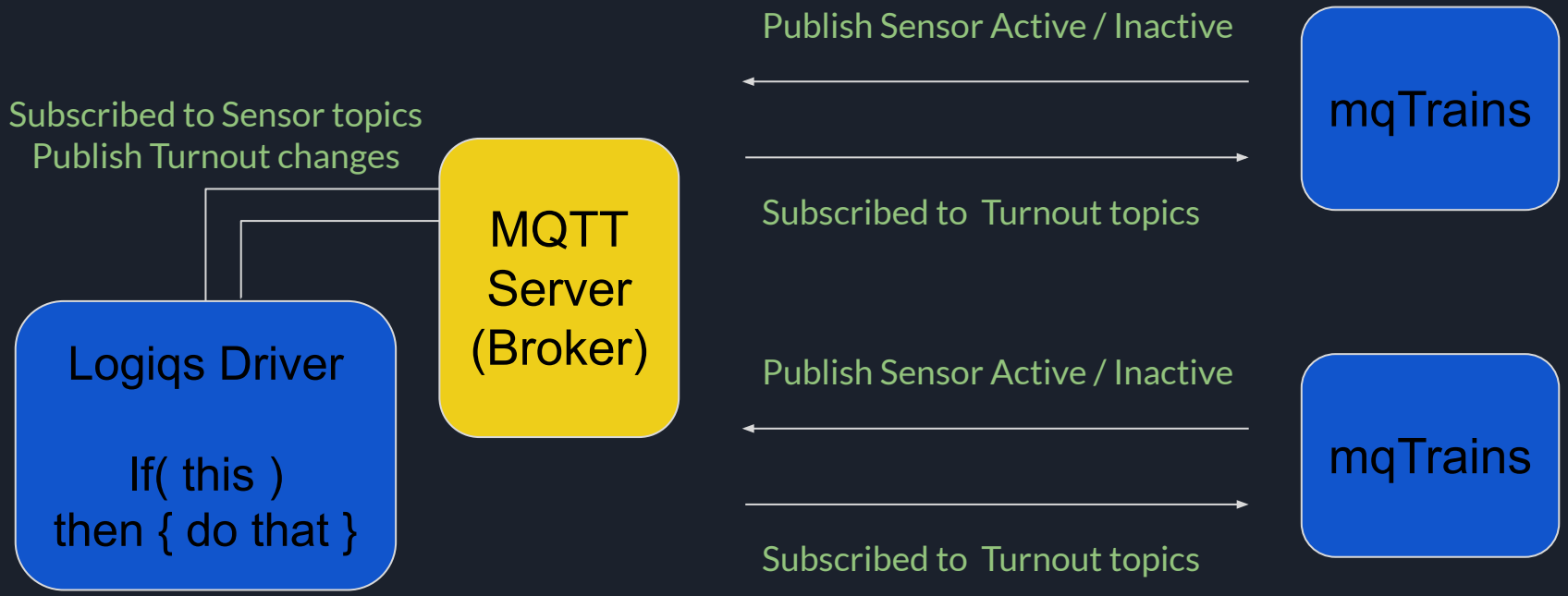


mqTrains blank slide

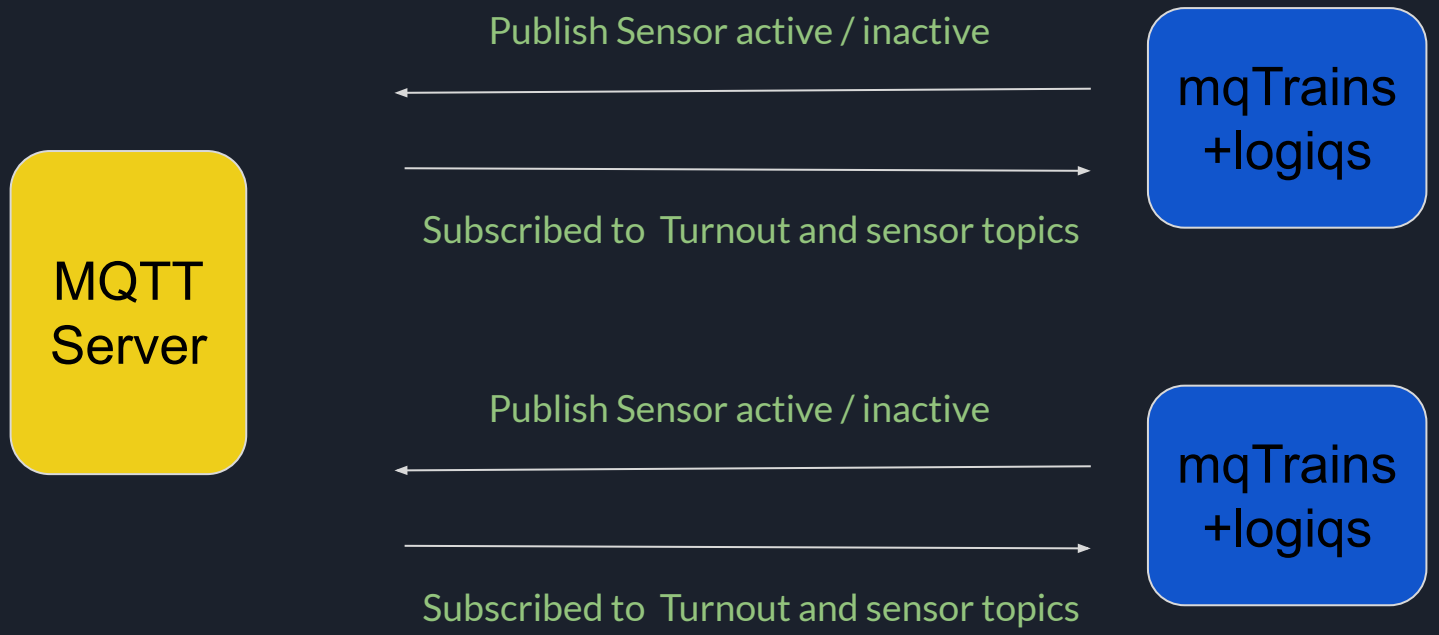
With text box...for future use…

Copy slide and move to required slot!

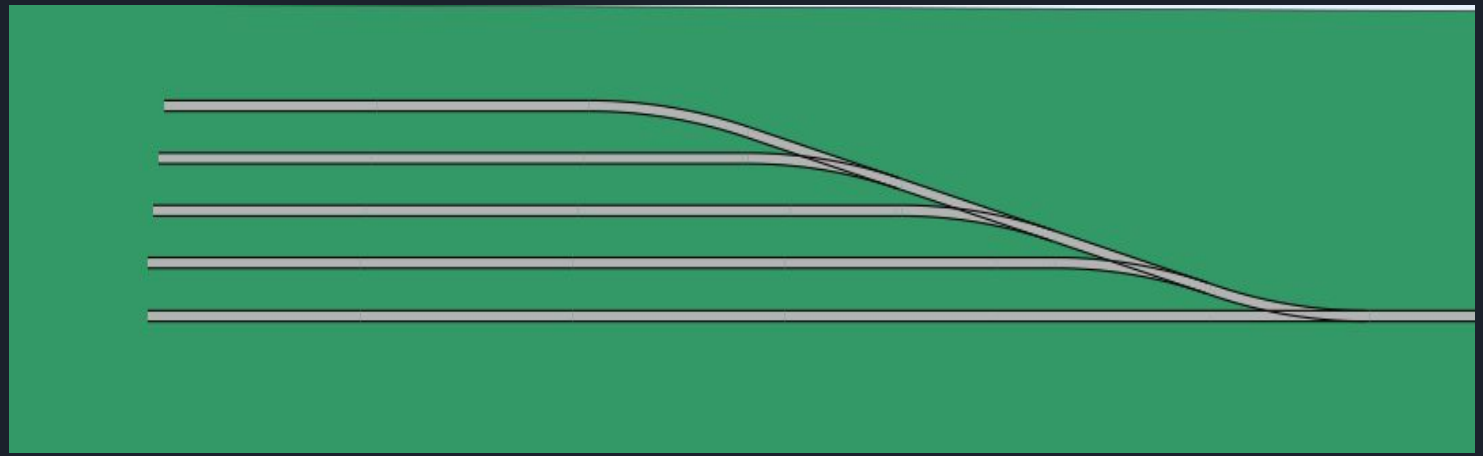
MQTT messages between devices



MQTT messages between devices



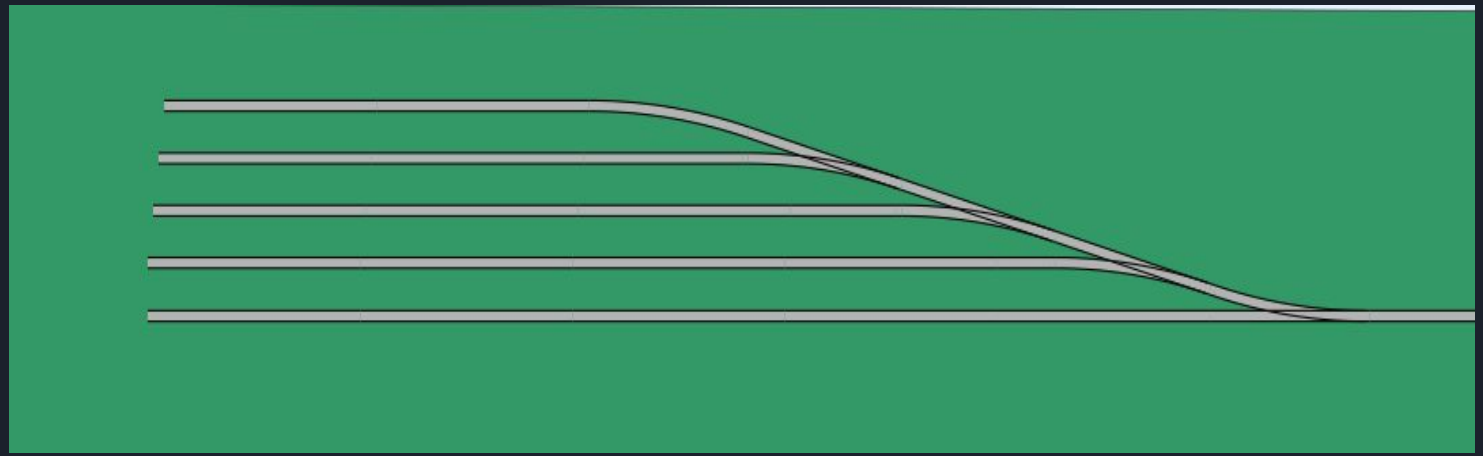
mqTrains with a staging yard



Staging yard

1 2 3 4 5

Push button route selection



Staging yard

Javascript, why?

- If the website is public, then no holes are needed in a firewall for remote ops!
- JMRI only had MQTT /track/turnouts early on and the web server was not easy to control from a tablet (might work better now)
- So, why not just...
  - Install lighttpd with `sudo apt install lighttpd` (or apache)
  - Download mqttws31.js
  - Download konva.min.js
  - Draw your own images
  - Create a few functions to subscribe to MQTT topics and publish when the user touch or click on something
- When a user changes an element with a topic, it changes the state and we “publish”
- When a subscribed to message comes in, it finds a match for the topic and updates the element

Javascript, why?

- If the website is public, then no holes are needed in a firewall for remote ops!
- JMRI only had MQTT /track/turnouts early on and the web server was not easy to control from a tablet (might work better now)
- So, why not just...
  - Install lighttpd with `sudo apt install lighttpd` (or apache)
  - Download mqttws31.js
  - Download konva.min.js
  - Draw your own images
  - Create a few functions to subscribe to MQTT topics and publish when the user touch or click on something
- When a user clicks an element with a topic, it changes the state and “publish”
- When a subscribed to message comes in, it finds a match for the topic and updates the element

EASY!!!

Javascript, how? okahandja.html

```
<html><head>
  <script src="mqttws31.js" type="text/javascript"></script>
  <script src="konva.min.js"></script>      <script src="txn.js"></script>
  <title>Okahandja (OKH)</title>
</head><body>
  <script type="text/javascript">
    SUBSCRIBED1 = 'TxNamib/servo/0021';     SUBSCRIBED2 = 'TxNamib/servo/0023';
    SUBSCRIBED3 = 'TxNamib/sensor/2003';

    MYNAME = 'Okahandja';     myWidth = 800;     myHeight = 280;

    // DRAGGING = true;
    // mainline (blue, imageIdx: 5 )
    arr.push( { key:  3, imageIdx: 5, locked: true,                 x:  10, y: 215, width: 760, rotation:   0 } );
    arr.push( { key:  5, imageIdx: 2, locked: true, state:false, x:  10, y: 215, width:   89, rotation:   0, topic: "TxNamib\/sensor/2003\/06", inv:true } );
    arr.push( { key:  7, imageIdx: 0, locked:false, state: true, x:  98, y: 203, width :  40, rotation:   0, topic: "TxNamib\/servo/0021\/01" } );
```

Javascript, how?

```
var stage = new Konva.Stage( {
  container: 'container',
  width: myWidth,
  height: myHeight
} );
stage.on( 'click', function( evt ) {
  doIt( evt );
} );  // on
stage.on( 'tap', function( evt ) {
  doIt( evt );
} );  // on

loadImages( arr, imageLayer );   // load and draw all at same time
stage.add( imageLayer );
setupClient( MYNAME );

// connect the client
client.connect( options );
```

And in txn.js:

```
function setupClient( myName ) {
  myIP = Math.floor( Math.random() * 100000 + 1 );
  client = new Paho.MQTT.Client( MQTTIP, PORT, myName+"."+myIP );
  client.onConnectionLost = onConnectionLost;
  client.onMessageArrived = onMessageArrived;
} // setupClient()

// Client name could be Okahandja.14531

// Reloading the page, or loading it from somewhere else will create
//  a unique (almost always) client name
```

Javascript, how? txn.js

```
function doIt( evt ) {
  var tg = evt.target;
  if( connected ) {
    if( tg.locked ) {   /* do nothing */   } else {
      if( tg.state ) {
        tg.setImage( images[ tg.imageIdx ].closedImage );
        tg.state = false;        imageLayer.draw( );
      } else {
        tg.setImage( images[ tg.imageIdx ].thrownImage );
        tg.state = true;        imageLayer.draw( );
      }  // if

      var payload = tg.state ? 'THROWN' : 'CLOSED';
      publish( tg.topic, payload );
    } // if locked
  } else {
    document.getElementById( '_logLine' ).innerHTML = "Sorry, not connected, refresh the page please.";
  } // if connected
} // function doIt( evt )
```

Javascript, how? txn.js

```
function onMessageArrived( message ) {       // called when a message arrives
  var mask = 1;
  var topic = message.destinationName;
  var payload = message.payloadString;
  if( topic[ topic.length-3] == '/' ) {    // need to have something/something/something/xx
    for( var sorc in kImages ) {           // loop through all the elements, crude but works!
      if( topic == kImages[ sorc ].topic ) {
        if( ( payload == 'THROWN' ) || ( payload == 'ACTIVE' ) ) {
          kImages[ sorc ].state = true;
          if( kImages[ sorc ].inv )          kImages[ sorc ].setImage( images[ kImages[ sorc ].imageIdx].closedImage );
            else                             kImages[ sorc ].setImage( images[ kImages[ sorc ].imageIdx].thrownImage );
        } // if topic matches
        if( ( payload == 'CLOSED' ) || ( payload == 'INACTIVE' ) ) {  /* do the reverse of above */  }
        imageLayer.draw( );
      } // if
    } // for
  } // if '/'
} // onMessageArrived( )
```

Javascript, Ernie? Who else?

- Take care of the disconnections and timeouts.
- Next on the Todo list
  - Need to handle the turnout “locking” and “request permission” to use
  - Show your train and which cars need to go where!

(Wait that is off-topic…)

REMOVE THIS - Suggested structure

Speed opens. Talk about your journey to here - the evolution of these sketches from way back whenever. Why is it what it has become? What are the objectives/motivations?

Do ~~we~~ you chat a bit about the technologies being used? (let's see how much time we have)

Walk through one of the mqTrains sketches - (your servo style?) (this will eat a lot of time)

Perhaps mention ideas for future development.

Hand over to David - getting back to basics. A recap of MQTT. Using mqTrains with JMRI. Using mqTrains without JMRI, Driving a staging yard. mqTrains with T-TRAK,

Hand back to Speed. mqTrains with JavaScript on the TxNamib

REMOVE THIS - It' just reminders

They want to know how your layout operations work without JMRI.

Running a 'Train-spotting' layout at an exhibition.

I can talk about Martin Watts' strategy of changing signals as a train passes - it's not operations but looks impressive to people who don't know better.

Reducing wires on modular layouts

Reducing wires on tiny modular layouts - T-TRAK

The title has mqTrains first so we can focus a bit on that

We can go through mqTrains in more detail

We can talk a bit more personal - the development journey - the technologies we're using, ideas/technologies we've tried but didn't work, time zone challenges, visions for the future

mqTrains the Journey, how small things create other things!

- 2016/06 LSR Arlington convention: Speed met Joel at a Miles Hale 3D printing clinic and promised Joel a SpaceNavigator 3D mouse *(Joel had to wait 12 months!)*
- 2017/03 One-wire Arduino signals from JMRI
- 2017/04 ESPs instead of Arduinos? Let go of all the cables!
- 2017/05 JMRI MQTT Signal nodes with buddy Chris
- 2017/11 MQTTSignalMastChanger.py in JMRI to send MQTT messages for Virtual Masts *(published on blog.RRRduino.com)*
- 2018/11 David asked a question about 3D printed signals that can hold NeoPixels in N-scale
- 2019/04 TxNamib decides to host operating sessions in that November DFW Interchange: need to control 220 turnouts (MQTT and JavaScript to the rescue)
- 2019/08 jmri_4.17.3ish_With_MQTT_Sensors.jar *(Please download 4.22 or later now!!!)*

mqTrains the Journey, how small things create other things!

- 2020/02 LSR Houston convention: Speed shows the one-wire Arduino Signals during hotel breakfast and Joel needed the ***WS2812b*** part number
- 2020/04 ESP with 1-wire Pennsy signals over MQTT
- 2020/04 Private github with Joel created
- 2020/05 v0.1 and David also joins github (and now we have 3 guys in 2 time zones) *(Can't believe it is only 1 year ago!!!)*

And the journey started to change all the code from a single .ino using C to multi-file C++, with reusable parts across all, create a web server and html pages to configure everything

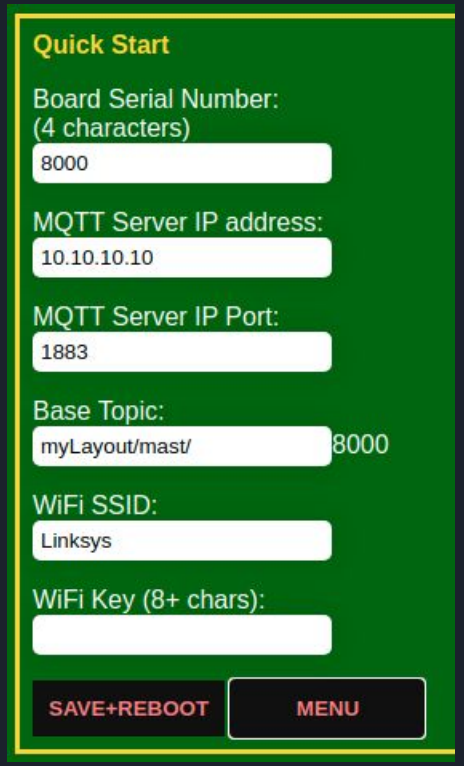
the Journey, tech talk about the upgrades!

From:

```
#define VERSION         "ESP8266-01, 2017.05.12, 0.06"
#define SERIALNUM       "8000"
#define MQTT_SERVER     "10.10.10.10"
#define HOSTNAME        "myLayout/mast/"
#define ssid            "Linksys"
#define password        "pass_word"
```

(then re-compile, upload and power cycle!)

To:



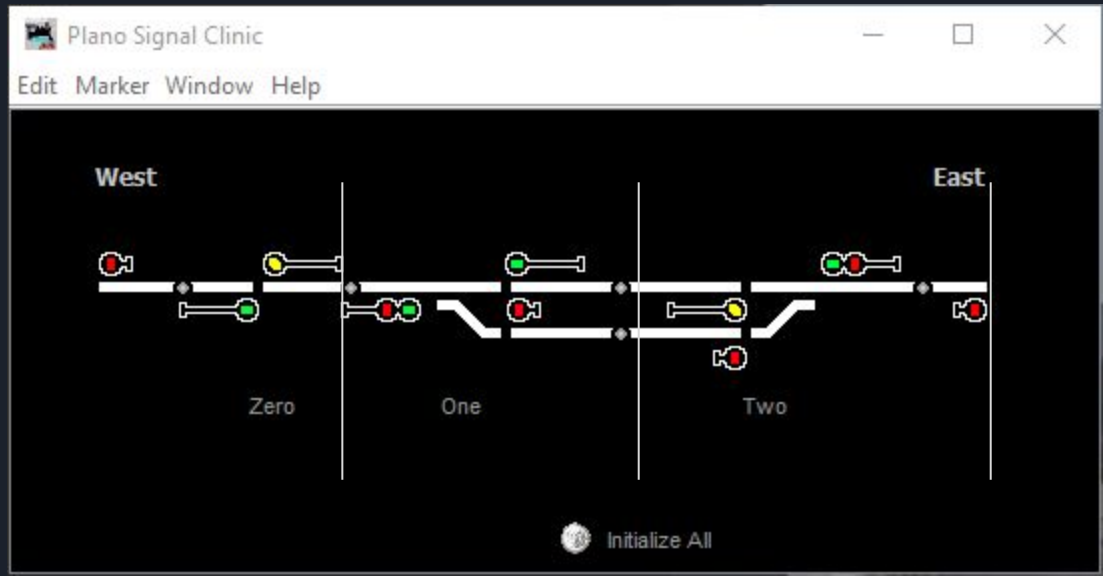
mqTrains

the Journey, tech talk about what we did, what did not work!

- const values changed to be configured by html
- EEPROM to SPIFFs to LittleFS
- Configurations saved with JSON
- Things present not documented yet:
  - OTA (Over the Air update)
  - Serial interface
- Things that did not work:
  - mDNS, DNS, iPhone vs Android *(decided to let you type in 2.2.2.1)*
  - Hardware: H-bridge drivers to move solenoid motors

mqTrains the Journey, tech talk about the future!

- Configuring Signals *(the rest is already done)*
- A **mqTrains** “plant”, see One and Two in the JMRI panel
- **mqTrains** Logiqs
- **mqTrains** Touch sensors
- PCBs for stall motors
- ESP-now?
- Move to Platform.io
  - get main.cpp back
  - And better debugging
- Oh, and please clean up the code to go Open Source!



Who is Speed?

- NMRAx at the bottom of the Helix
- Namibia, south western Africa
- Murphy, Texas
- Modeling the TransNamib in N-scale!
- Weird nickname? Well, how fast can you read?