



# MoxieWorld API Guide

MoxieWorld offers three ways to consume your data:

- **MoxieWorld iOS App Interface.** Interactive 3D and map-based visualization, create live and historic channels for plotting as heatmaps or as plotted channels on the dashboard.
- **Live Data: MQTT Broker.** Subscribe to the live data stream of your MoxieTags to create your own real-time visualization, or to store in your preferred database
- **Historic Data: InfluxDB Database.** Access historic data for your MoxieTags. Query, filter, and parse to see trends over time or to replay past scenarios

This guide provides the details needed to access your Live and Historic data streams. The app operation is documented in the application notes and our online tutorials.

## Live Data using MQTT

Live data from MoxieTag is provided by MQTT Broker. The broker is hosted on Amazon Web Services by 3<sup>rd</sup> party vendor [CloudMQTT](#). Your MoxieWorld subscription account provides a secure partition on this broker. The broker can be accessed with these credentials, and will have Read Access to the topic pattern listed:

<b>Server</b>	smart-accountant.cloudmqtt.com
<b>Port</b>	1883
<b>Websocket Port (TLS)</b>	443
<b>Client ID</b>	[any unique string]
<b>Pattern</b>	[CUSTOMER SPECIFIC]
<b>Username</b>	[CUSTOMER SPECIFIC]
<b>Password</b>	[CUSTOMER SPECIFIC]
<b>Topic Filter</b>	MW/[CUSTOMER SPECIFIC]/#
<b>Protocol</b>	Mosquitto 1.5.8

Please ensure each device connecting to the broker has a unique Client ID, otherwise the Broker will be forced to disconnect each of the duplicate names. In the case you need live support from Moxie IoT, it is best if your chosen username is descriptive, e.g. “myCompanyUser001”. Because Moxie has provided a filter on your username/password, it is acceptable for your client to simply subscribe to “#”.

If you are new to MQTT, we suggest you become familiar with the topic/payload system using a UI tool like [MQTT Explorer](#). When integrating into your own project, there are almost certainly MQTT client libraries for your preferred programming language: we use [paho-mqtt](#) in Python for internal development.

## MQTT Topics

Each tag publishes data to a unique topic based on its serial number (UUID) and the account (CO) that owns it. To allow easy filtering, the packet type is also specified in the topic.

Packet type	Topic Pattern	Example
Status – Last Will Topic	MW/[CO]/Tag/[UUID]/Status	MW/Moxie/Tag/ABCD1234/Status



<b>UWB</b> (indoor) position	MW/[CO]/Tag/[UUID]/UWB	MW/Moxie/Tag/ABCD1234/UWB
<b>GPS</b> (outdoor) position	MW/[CO]/Tag/[UUID]/GPS	MW/Moxie/Tag/ABCD1234/GPS
<b>MDU</b> Machine Doctor	MW/[CO]/Tag/[UUID]/MDU	MW/Moxie/Tag/ABCD1234/MDU
<b>MOT</b> 3-Phase Motor	MW/[CO]/Tag/[UUID]/MOT	MW/Moxie/Tag/ABCD1234/MOT

This is a complete listing of standard packets; their contents are listed in the following sections. Based on customer 'Plus One' installation there may be additional data fields in the packets: See 'Plus One Data' for a listing of these fields.



## ***Status Payload***

The Status Payload is a JSON Packet that serves as the MQTT LW&T message. It notifies subscribers whether the Tag is connected to the broker or not (due to loss of internet or power). This message is available on all MoxieWorld devices. Occasionally a Tag will repeat the Status message with an additional RSSI field containing the WIFI or LTE signal strength. This can be used to diagnose connectivity issues around your facility. This message is sent with Retain = True.

Example:

```
{"ID":168,"status":1,"rssi":-78}
```

<b>Field</b>	<b>Type</b>	<b>Units</b>	<b>Description</b>
"ID"	Int	N/A	Tag identifier (unique per facility)
"status"	Int	N/A	1 when online, 0 when offline (sent as LW&T)
"rssi"	Int	dBm	WIFI RSSI (31 = unknown) ( <i>this key is optional</i> )



## UWB Payload

The UWB Payload is a JSON Packet that provides positioning information for MoxieTags with indoor positioning capability. This message is sent with Retain = False.

Example:

```
{ "ID":168, "x":1.7854, "y":321.9117, "z":10.7230, "vx":0, "vy":0, "vz":0, "v":0, "h":0, "mxx":2.7763, "myy":0.7389, "mxy":1.2580, "ok":2, "anchors":"181917" }
```

Field	Type	Units	Description
"ID"	Int	N/A	Tag identifier (unique per facility)
"moving"	Bool	N/A	1 when tag is moving, 0 when stationary (vibration)
"x"	Float	m	X position
"y"	Float	m	Y position
"z"	Float	m	Z position
"vx"	Float	m/s	X speed
"vy"	Float	m/s	Y speed
"vz"	Float	m/s	Z speed
"v"	Float	m/s	Horizontal Speed
"h"	Float	rad	Heading (CCW from X axis)
"mxx"	Float	m <sup>2</sup>	X position variance estimate
"myy"	Float	m <sup>2</sup>	Y position variance estimate
"mxy"	Float	m <sup>2</sup>	XY position covariance estimate
"uwb"	Int	[UWB Status]	Tag position solution status (0-6)
"anchors"	String	N/A	[for internal use]

### UWB Status Enumeration

0	Solution diverged
1	Solve OK
2	Solve OK (Stationary)
3	Out-of-Bounds
4	Only 2 Anchors
5	Only 1 Anchor
6	No Anchors



## GPS Payload

The GPS Payload is a JSON Packet that provides positioning information for MoxieTags with outdoor positioning capability. This message is sent with Retain = False.

Example:

```
{"ID":168, "fix":0, "lat":1.7854, "lon":321.9117, "alt":10.7230,  
"hAcc":3.45, "v":0.01, "h":0.05, "numSV":4}
```

Field	Type	Units	Description
"ID"	Int	N/A	Tag identifier (unique per facility)
"fix"	Int	[GNSS Fix]	Fix type (0-6)
"lat"	Double	deg	Latitude
"lon"	Double	deg	Longitude
"alt"	Double	m	Height above ellipsoid
"hAcc"	Float	M	Horizontal accuracy estimate
"v"	Float	m/s	Horizontal speed
"h"	Float	rad	Heading (CW from north)
"numSV"	Int	N/A	Number of satellites in view

### GNSS Fix Enumeration

0	No fix
1	Dead reckoning only
2	2D-fix
3	3D-fix
4	GNSS+Dead reckoning
5	Time-only fix



## MDU Payload

The MDU Payload is a JSON Packet that provides positioning information for MoxieTags with machine monitoring capability. The MDU unit and each of its probes will send its own payload, which one is denoted by the "src" field. This MQTT message is sent with Retain = False.

Example:

```
{"ID":168, "src":0, "ax":0.124, "ay":0.345, "az":2.897, "a":2.906, "fft":"[DATA]", "VOC":4.6, "pres":101.453, "temp":29.34, "hum":60.45}
```

Field	Type	Units	Description
"ID"	Int	N/A	Tag identifier (unique per facility)
"src"	Int	[Source]	Data source
"ax"	Float	m/s <sup>2</sup>	X acceleration (RMS, 1 second)
"ay"	Float	m/s <sup>2</sup>	Y acceleration (RMS, 1 second)
"az"	Float	m/s <sup>2</sup>	Z acceleration (RMS, 1 second)
"a"	Float	m/s <sup>2</sup>	total acceleration (RMS, 1 second)
"fft"	String	N/A	Acceleration FFT encoded as a string [See AN-004]
"VOC"	Float	ppm	Volatile organic compound
"pres"	Float	kPa	Ambient pressure
"temp"	Float	°C	Ambient temperature
"hum"	Float	%RH	Ambient humidity

### Source Enumeration

0	MDU internal
1	Probe 1
2	Probe 2
3	Wireless 1
4	Wireless 2



### 3-Phase Motor Payload

MoxieWorld 3-Phase motor sensors provide information about the load, speed, and direction of a 3-phase (A/C Synchronous) motor. The RMS and frequency are determined from Clamp 1. Direction is determined by noting if the waveform of Clamp 1 leads or lags Clamp 2. Data is valid in a range of approximately 10 to 400 Hz, 0.1 to 100 Amps. If included as a plus-one option for UWB or GPS positioning Tag, this data will be directly appended in the UWB or GPS packet. This message is sent with Retain = False.

Example:

```
{"ID":168, "rms":4.072,"freq":60.24,"dir":1,"active":1, "load":2}
```

Field	Type	Units	Description
<b>3-Phase Motor</b>			
"ID"	Int	N/A	Tag identifier (unique per facility)
"rms"	Float	A	Measured current (RMS) from clamp 1
"freq"	Float	Hz	Current frequency from clamp 1(based on zero-crossing)
"dir"	Int	[Direction]	3-phase current sequence
"active"	Int	N/A	Logical OR of vibration motion and ("load" > 0)
"load"	Int	[Load]	Enumeration of load, based on RMS current

#### Direction Enumeration

-1	Counterclockwise (L2 leads L1)
0	Stationary/unknown
1	Clockwise (L1 leads L2)

#### Load Enumeration

0	Unknown (below noise floor)
1	Low Load (unloaded spindle/hoist)
2	Moderate Load (cutting/lifting)
3	High Load (Overload)



**Plus-One Fields**

MoxieWorld “Plus-One” sensors provide extended sensing capability to any MoxieTag. To ensure this data is synchronized with positioning information, it is appended to positioning packets. This means that UWB and GPS packets may or may not contain these keys – the end-user application must handle packets with and without this data.

Field	Type	Units	Description
<b>3-Phase Motor</b>			
“rms”	Float	A	Measured current (RMS) from clamp 1
“freq”	Float	Hz	Current frequency from clamp 1 (based on zero-crossing)
“dir”	Int	[Direction]	3-phase current sequence
“active”	Int	N/A	Logical OR of vibration motor and (“load” > 0)
“load”	Int	[Load]	Enumeration of load, based on RMS current
<b>Machine Cycle Counter</b>			
“count”	Int	N/A	Cycle count. Reset when JOB is started or paused
“duration”	Float	s	Cycle duration in seconds
“duty”	Float	N/A	Fraction of cycle time during which the switch is closed

**Direction Enumeration**

-1	Counterclockwise (L2 leads L1)
0	Stationary/unknown
1	Clockwise (L1 leads L2)

**Load Enumeration**

0	Unknown (below noise floor)
1	Low Load (unloaded spindle/hoist)
2	Moderate Load (cutting/lifting)
3	High Load (Overload)



## Historic Data using InfluxDB 2.0

Historic data from MoxieTag is stored in our [InfluxDB 2.0](#) time-series database. The database is hosted on Amazon Web Services by 3<sup>rd</sup> party vendor InfluxDB Cloud. Your MoxieWorld subscription account provides a secure partition on this system. Your subscription terms determines how long data is retained before being automatically deleted as well as the sample rate: most subscriptions log data every 10 seconds (if available on MQTT) and retains for 1 year. The database can be accessed with these credentials, with Read Access to the bucket listed:

<b>URL</b>	https://us-west-2-1.aws.cloud2.influxdata.com
<b>Org</b>	acd6274a33eb9b84
<b>Bucket</b>	CUSTOMER SPECIFIC
<b>Token</b>	CUSTOMER SPECIFIC

All data is stored in a manner that best preserves the information in the live data packets. Nomenclature for the tables in InfluxDB 2.0 is different than the MQTT topics.

MQTT Concept	InfluxDB 2.0 Concept	Description
Tag UUID	_measurement	Unique device
JSON key	_field	Each json key is stored to a _field of the same name
JSON value	_value	Each json value is stored to a _value for its _field
Live Data	_time	Each packet is timestamped in UTC time

The database can be accessed in many ways. There are client libraries in these languages:

C#	JavaScript/Node.js	PHP
GO	Kotlin	Python
Java	PHP	Ruby

When used for queries, all of these tools essentially expose one of two primary access techniques: Flux and InfluxQL. The MoxieWorld app performs Flux queries (input as strings) using the Flux language HTTP Post method as documented [here](#). Most users can decide whether [Flux](#) or [InfluxQL](#) is best for them based on which of the following query examples seems most familiar/comfortable. We recommended using Flux language queries because the query can be transferred as a string across platforms, including importing into the MoxieWorld App for interactive visualization.

InfluxQL Example	<pre>SELECT "x", "y" AND "ABCD1234" FROM "MoxieBucket" WHERE time &gt;= '2020-01-01T00:00:00Z'</pre>
Flux Example	<pre>data = from(bucket: "MoxieBucket")    &gt; range(start: 2020-01-01T00:00:00Z)    &gt; filter(fn: (r) =&gt;     r._measurement == "ABCD1234" and     r._field == "x" or r._field == "y"   )</pre>



## Appendix A – System Enumerations

### WIFI Status Enumeration

0	Connected
1	SSID not found
2	Bad password
3	Connection lost
4	Disconnected
5	Idle
6	Scan complete
255	Unknown

### MQTT Client Connection Status Enumeration

-4	Connection timeout
-3	Connection lost
-2	Connection failed
-1	Disconnected
0	Connected
1	Bad protocol
2	Bad Client ID
3	Unavailable
4	Bad Credentials
5	Unauthorized

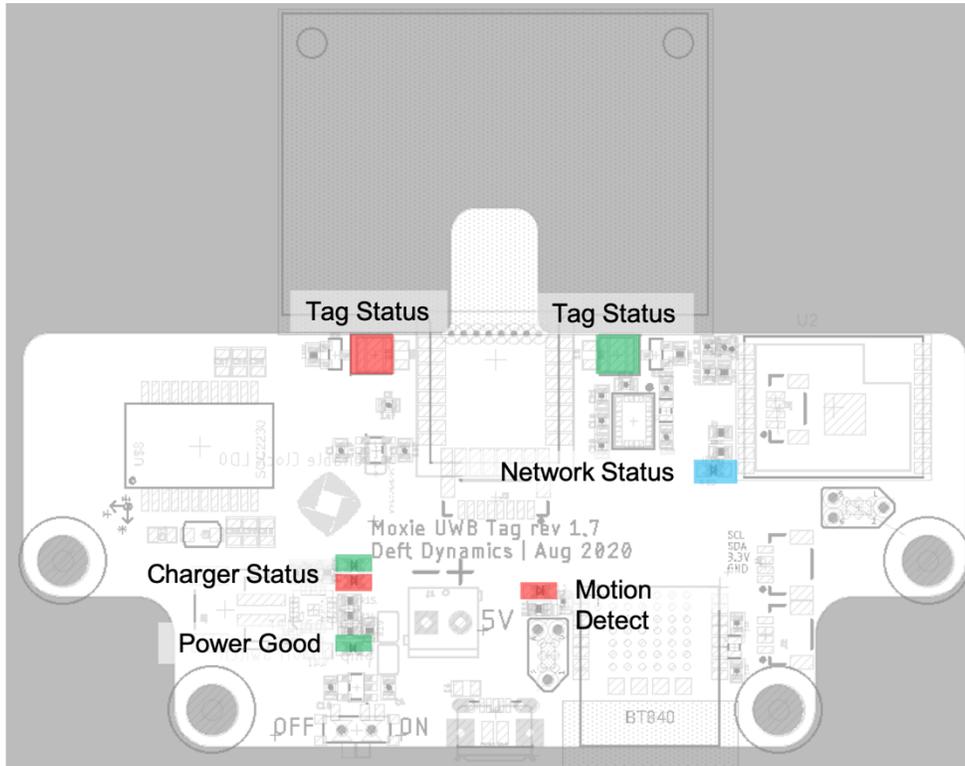
## Appendix B – MoxieAnchor LED Interpretation

### UWB Status (large green/red LED, case glow color)

Blink green	Anchor was used by a tag in the last 5 seconds
Blink red	Anchor was not used by a tag in the last 5 seconds
Pulse green	Device is connected to BLE



## Appendix C – MoxieWorld Hardware LED Interpretation



### Tag Status (large green/red LED, case glow color)

	UWB/GPS Tag	Motor/Machine Doctor Tag
Blink green	Position solution is OK	Motor load Safe
Blink red	Could not solve position	Motor load High
Pulse green	Device is connected to BLE	Device is connected to BLE

### Motion Detect Status (small red LED, bottom right)

Off	In motion
On	Stationary (no motion detected)

### Internet Status (small blue LED, top right)

Off	WIFI and MQTT are both connected
One blink (each second)	WIFI is connected, MQTT is not
Two blinks (each second)	Neither WIFI nor MQTT is connected



API v1.0  
Document rev 2.42  
Last Edited 9-29-20

**Battery Charger & Power Good**  
(set of three small red/green LEDs, bottom left)

<b>Single Green</b>	<b>Pair of Red/Green</b>	<b>Meaning</b>
Off	Both Off	Battery is powering the system <i>or</i> system is OFF
On	Both Off	Power good, no battery connected
On	Green On, Red Off	Power good, battery fully charged
On	Green Off, Red On	Power good, battery charging
On	Both On	Power good, no battery connected