

Mako & PassPort

RF Integration



Mako & PassPort

RF Integration

Contents Page

INTRODUCTION

SECTION	PAGE
Mako V2 – Programmable Logic Controller	3
LoRa RF Telemetry	4
About This Guide	5

V-NET SET-UP

LoRa RF Configuration	6
Modbus over LoRa RF Configuration	7-8
Modbus Tables	9-11
Modbus Configuration Finalised	12
Multiple Device Topology	13

RF COMMUNICATION SET-UP

Simulated RF Communications Setup	14
Basic Modbus & Decoder Nodes	15
Modbus Getter Node	16-17
Modbus Queue Node	18
Modbus Decoder Node	19
Inject & Debug	20

PREPARING THE PAYLOAD

Merge Your Payloads	21
Formatter Node	22

SENDING DATA TO SMITHTEK.CLOUD

SECTION	PAGE
Smithtek.Cloud Node	23
Store & Forward	24
CONCLUSION	25

Mako & PassPort

RF Integration

Mako V2 Programmable Logic Controller

- Plenty of analog and digital IO.
- GPS for asset tracking.
- Wi-Fi protocols, Mqtt, TCP, UDP, HTTP.
- PID, sequencers, counters, delays.
- Multiple RS485 interfaces.
- SD card for data logging.
- Secure RF for telemetry over long distances 15KM LOS.
- Free drag and drop programming software.
- Low cost Australian built.
- Reliable local support.



Mako & PassPort

RF Integration



Mako & PassPort

RF Integration

This guide will help you connect the **Mako PLC** to the **PassPort** via **LoRa RF** using the **Modbus** protocol. It covers building basic Modbus tables that align between the Mako and the PassPort, understanding the timing required to ensure proper communication while adhering to air data rules. The settings described are optimised for long-range communication up to 15KM+.

This guide also addresses handling scenarios where Mako devices go offline, ensuring continued reading of remaining devices, and managing PassPort's store and forward functionality when internet connectivity is lost. Additionally, it explains how to combine data from multiple Mako devices into a single payload and format it for transmission to **Smithtek.cloud**.

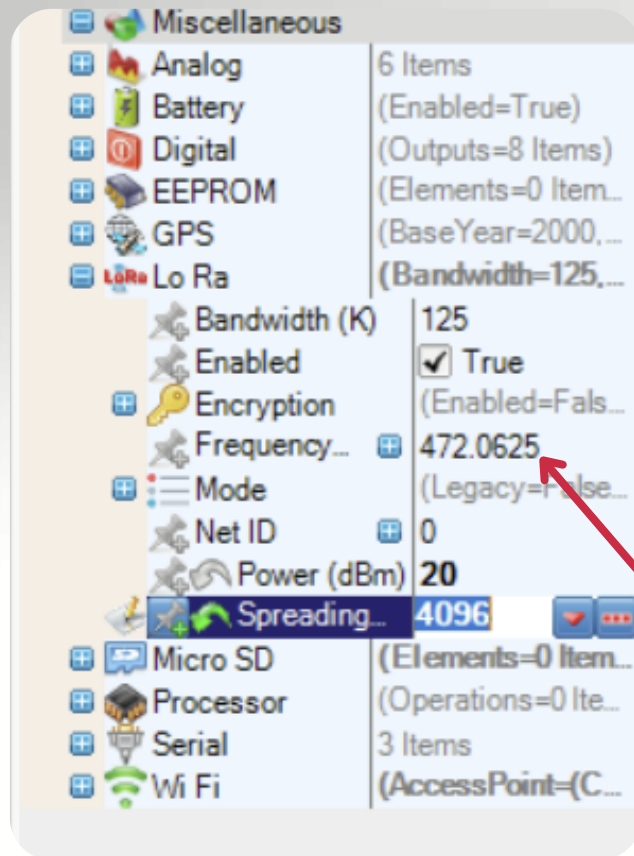
- Description text will be in **black**
- Instructions will be in **green**
- Important notes will be in **red**

This guide assumes you already know how to access and upload program changes to both the Mako PLC and PassPort Gateway.

Mako & PassPort

RF Integration

V-NET Setup - Config



In V-NET, we will configure the LoRa RF antenna settings for optimal long-range performance.

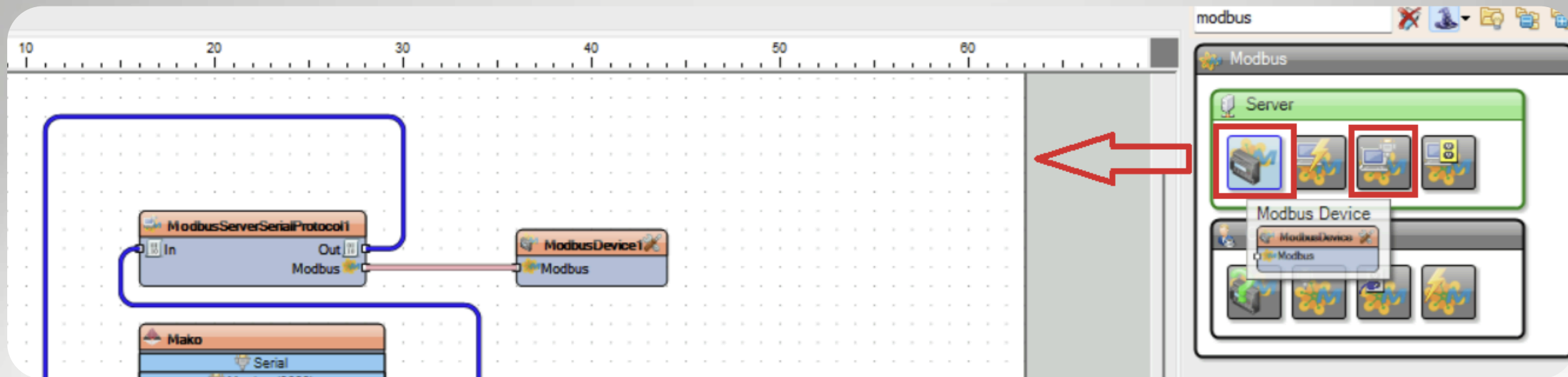
1. Open V-NET.
2. Click the Mako board and look for LoRa in the properties
3. Configure your LoRa RF settings the same as the image.

The Frequency 472.0625 is for the Australian region only!

Mako & PassPort

RF Integration

V-NET Setup - Modbus over LoRa RF Config



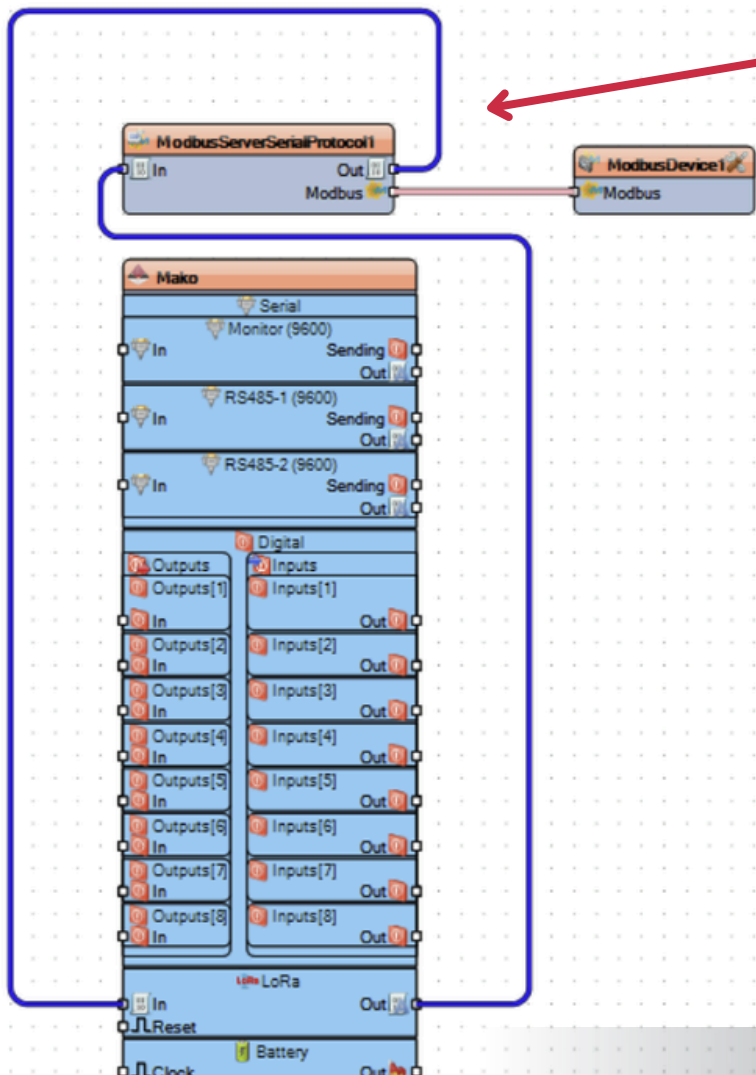
Now we will setup the Mako to be a Modbus slave. We will configure it to be a slave over the LoRa RF, you would treat this exactly like a hardware slave device. Drag and drop the nodes in the red boxes to your work space.

- Add 1 * Modbus device
- Add 1 * Modbus Server serial slave protocol (RTU ASCII)

Mako & PassPort

RF Integration

V-NET Setup - Modbus over LoRa RF Config



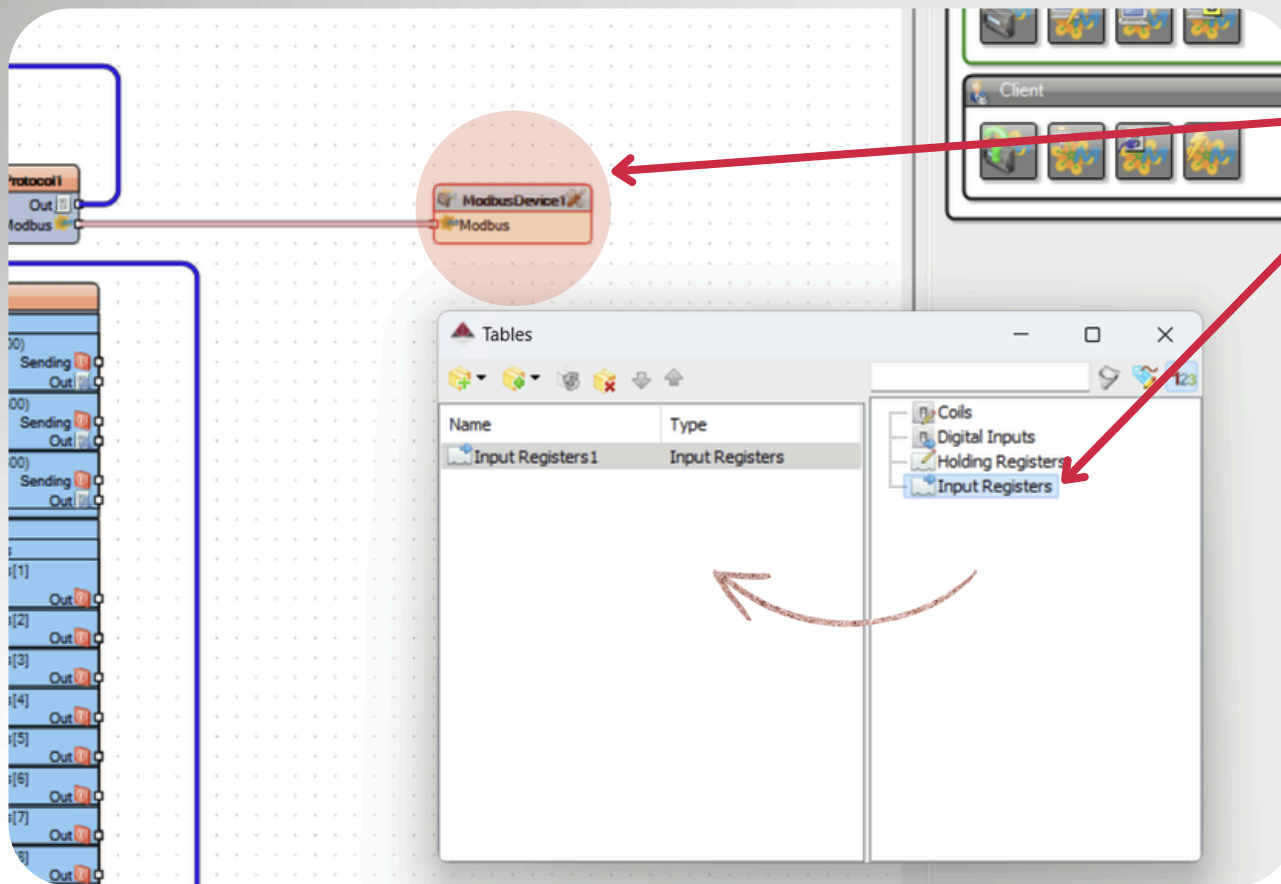
Now wire the nodes as per the picture. The Modbus protocol node will wire to the LoRa in/out, this is for data in and data out.

- Wire between each node exactly the same as the Image.
- Left mouse click and drag the wire from pin to pin.

Mako & PassPort

RF Integration

V-NET Setup - Modbus over LoRa RF Config - Modbus Tables



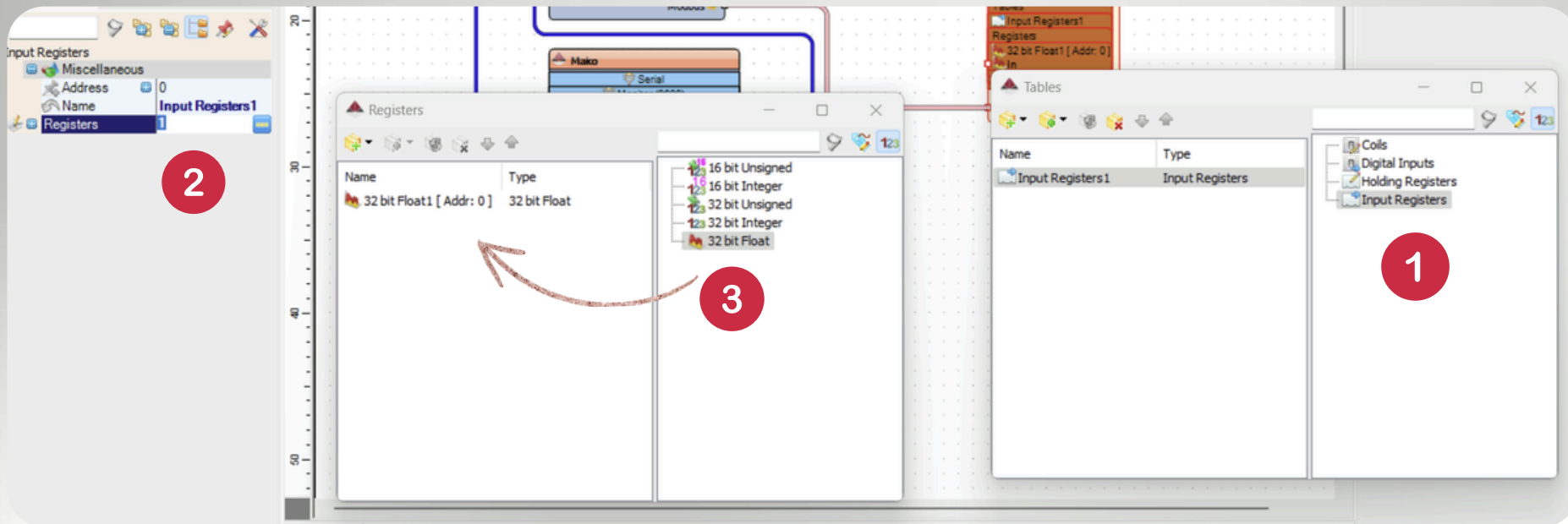
1. Double click the Modbus device component
2. In the tables window double click the registers you want, they will jump from the right-side window to the left.

Now we will add the **Modbus** tables. **Input registers**, **holding registers**, **coils** and **input status**.

Mako & PassPort

RF Integration

V-NET Setup - Modbus over LoRa RF Config - Modbus Tables



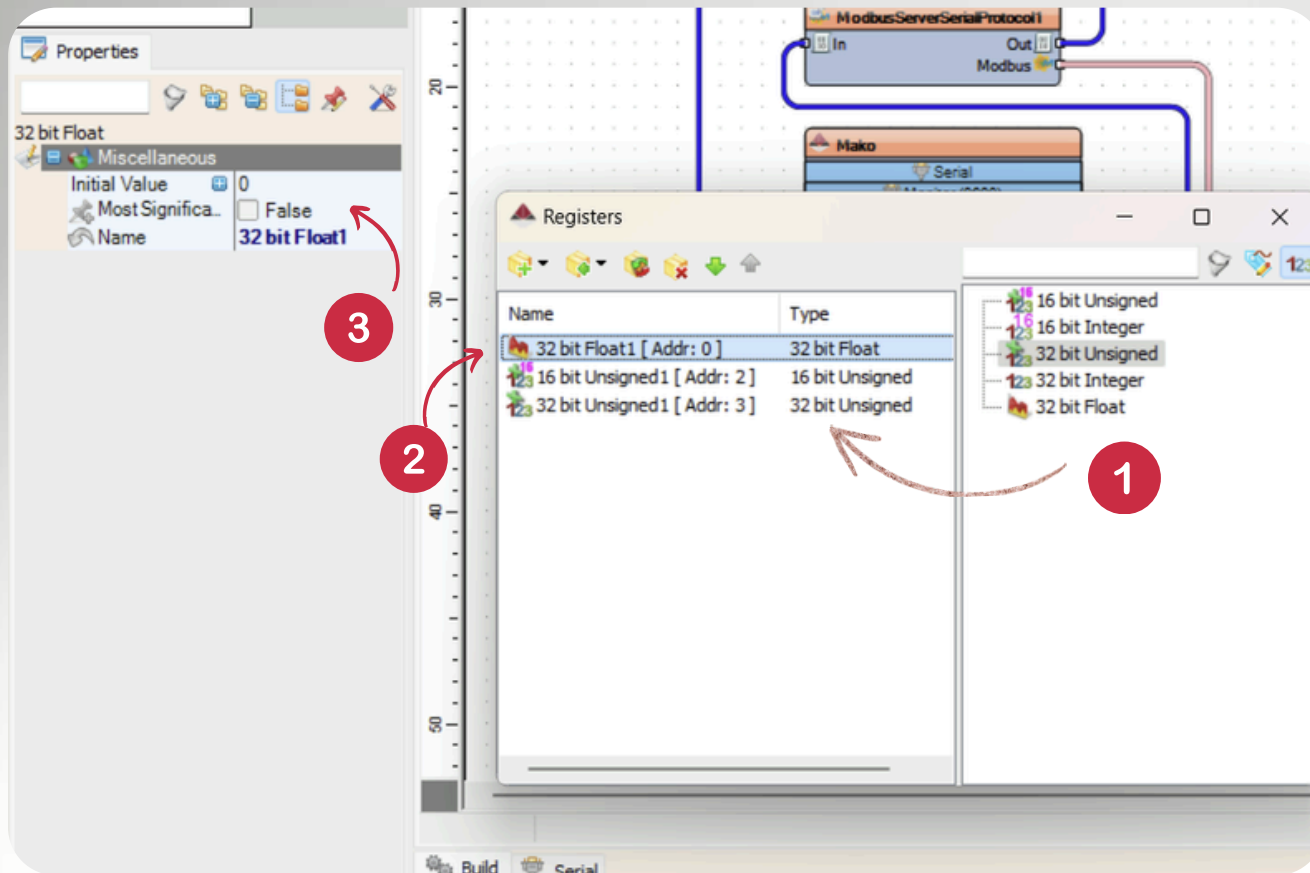
Add the registers per table type. Registers are data like 16bit, signed * unsigned, 32 bit and floats etc.

1. Click on your new Modbus table
2. The very left side of the V-NET editor a properties window will appear, click the little blue box with the 3 dots.
3. A new register window will open. Add your data types, double click and they will jump from the left side of the window to the right.
4. Add as many data types as you need for your program, in this guide we just added 1*32bit float.
5. Repeat this for every different table you add, so if you add coils or holding registers, repeat the process to add the data registers.

Mako & PassPort

RF Integration

V-NET Setup - Modbus over LoRa RF Config - Modbus Tables



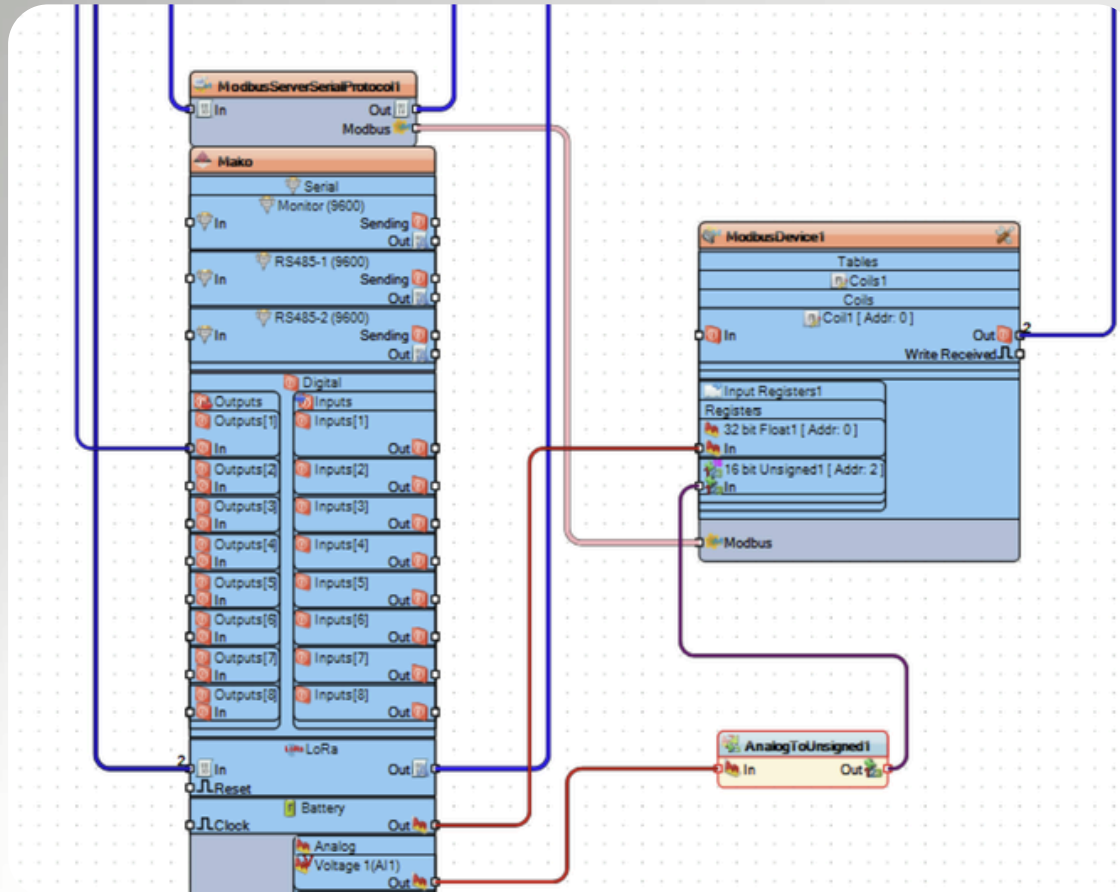
1. You can add as many data types to your registers table. If you add any of the 32bit data types you need to set the MSB ('Most Significant Byte') byte setting to true in the property
2. Click on the 32bit register
3. Set "Most Significant Byte" to 'True'

It is important to ensure 'Most Significant Byte' is set to true when communicating with the PassPort using Modbus. This implies to hardware wired serial, RF or Modbus over TCP. The PassPort will not be able to parse (decode) the data properly if not set. The 16bit registers do not require any other property settings.

Mako & PassPort

RF Integration

V-NET Setup - Modbus over LoRa RF Config - Modbus Finalised



The completed example uses two **Modbus** data tables, **Input Registers** and **Coils**.

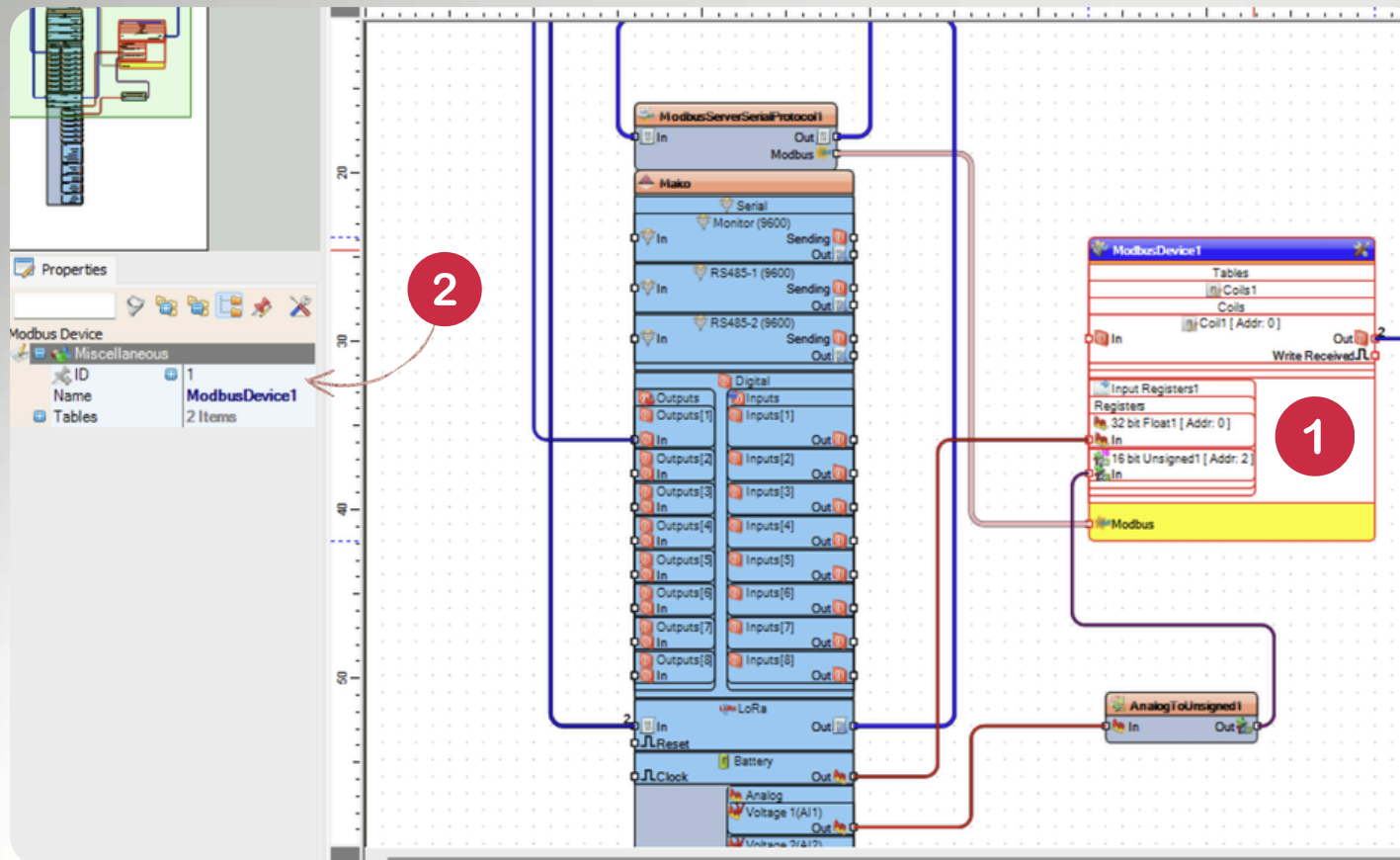
The Input Register table contains two registers: one float and one 16-bit unsigned integer. We have also added one coil so we can write back and control a digital output (on/off).

Next, we will connect to the PassPort and open **Node-RED**. The goal is to match the tables in Node-RED to those in V-NET, in the exact same order and by data type. This ensures the data is polled and decoded in a consistent, structured way.

Mako & PassPort

RF Integration

V-NET Setup - Multiple Device Topology



1. Click the Modbus device component in the middle.
2. A new properties window will open on the left side.
3. Change the Modbus ID
4. Be sure to do this for every Mako on the same network.
5. No extra property changes need to be done to the LoRa settings.

Before we access Node-RED through the PassPort, we have 1 final setting to change and that is your Mako Modbus slave ID. Just like a standard hardwired RS485 bus, the Mako still needs its own Modbus ID, even over LoRa RF. If you have multiple Makos on the LoRa RF Network, be sure to increase the ID for each new Mako added to the system. Moving on we are going to pretend we have 3 Makos communicating back to the PassPort.

Mako & PassPort

RF Integration

A Simulated RF Comms Setup



Our simulation will emulate three Mako units onsite.

Each Mako has been programmed with different Modbus ID.

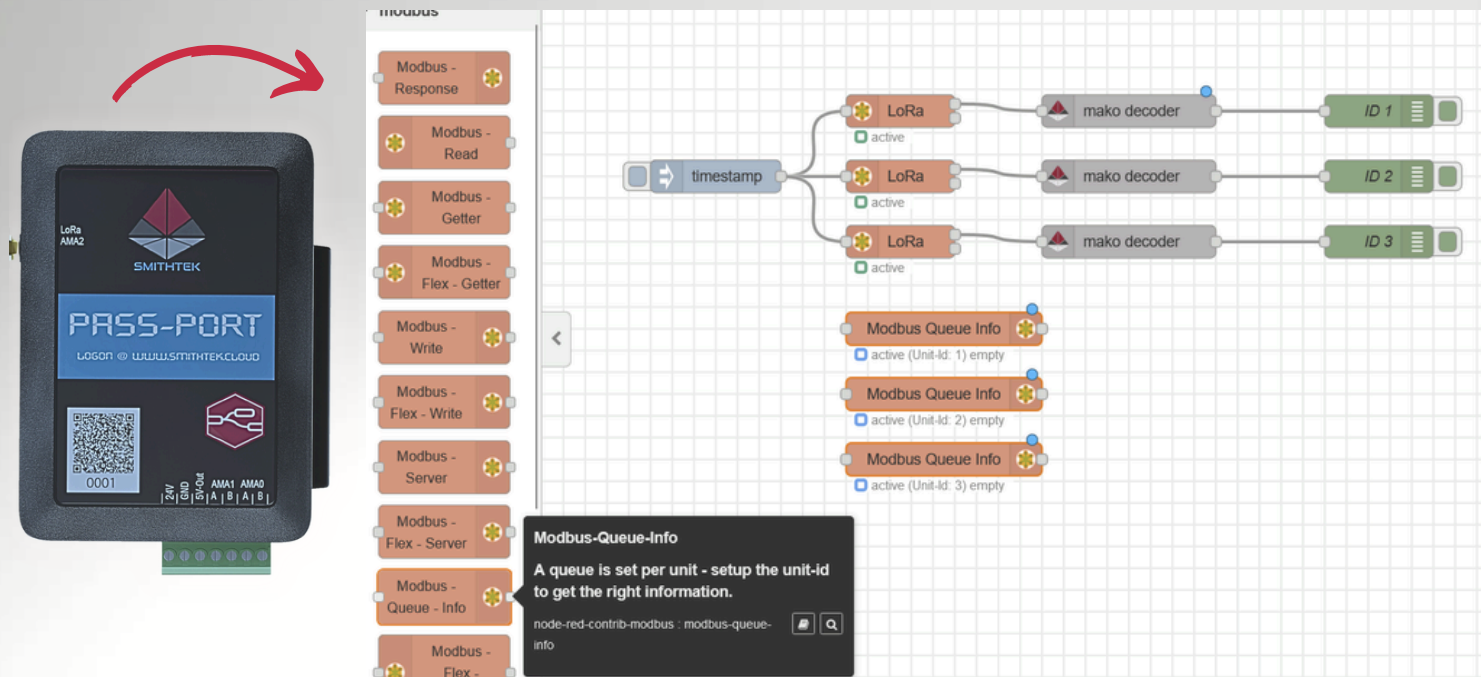
The PassPort will communicate with each Mako via these ID's individually. The concept is the same as a hardwired Modbus circuit.



Mako & PassPort

RF Integration

Simulation Topology: Basic Modbus and Decoder Nodes



Inside the PassPort Node-RED, import the same nodes in the image:

- 3* Modbus Getter (Orange)
- 3* Mako Decoder (Grey)
- 3* Modbus Queue (Orange)
- 1* Inject (Blue)
- 3* Debug Nodes (Green)

Connect to the PassPort and log into **Node-RED**. We will import a set of nodes to interact with the LoRa RF antenna. The flow will communicate with three simulated Makos over a simulated long range. We will add a **'Timed Inject'** node to trigger polling every minute and a **'Debug'** node to read the values. Then, we will use a **'Join node'** to combine all Mako data into a single payload. Next, we will add a **'Format'** node to include a timestamp and convert booleans to 1/0. Finally, we will add the **'Smithtek Out'** node to send data to the cloud and a **'Store and Forward'** node to log data locally if the internet connection is lost.

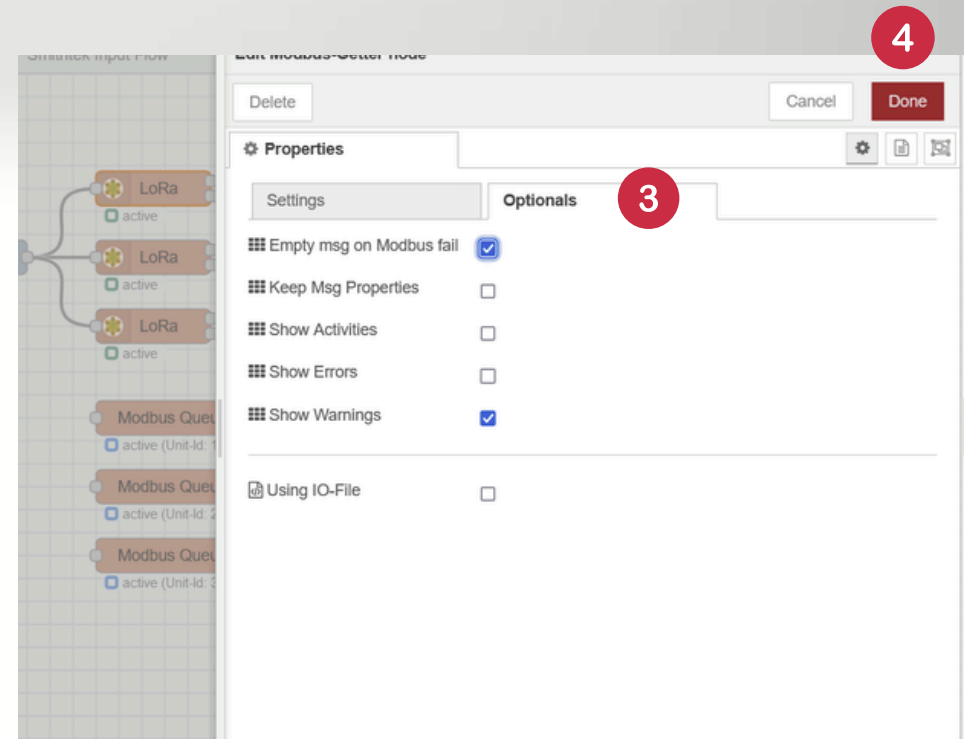
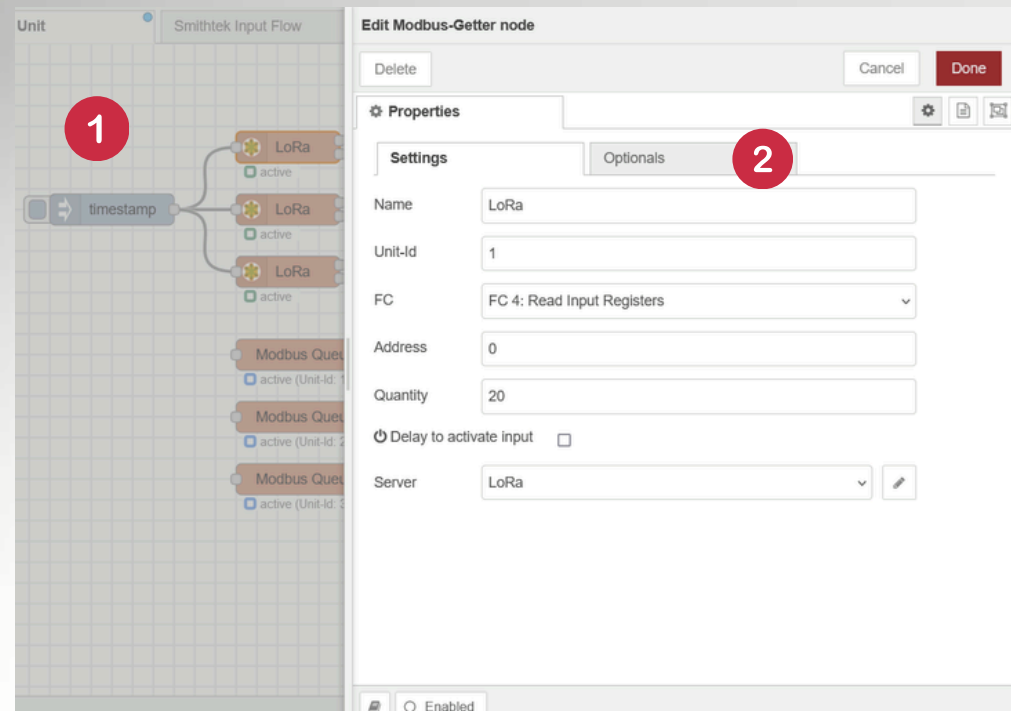
Mako & PassPort

RF Integration

Simulation Topology:

Modbus Getter Node (Repeat Config for Each Node)

We will click and open the setting for one of the Modbus nodes and setup some basic properties that set how the node will behave if it has a fault. Activating more fault settings is your preference.



1. Double click the 'Modbus Getter node'

2. Click 'Optional'

3. In 'Optional', copy these settings

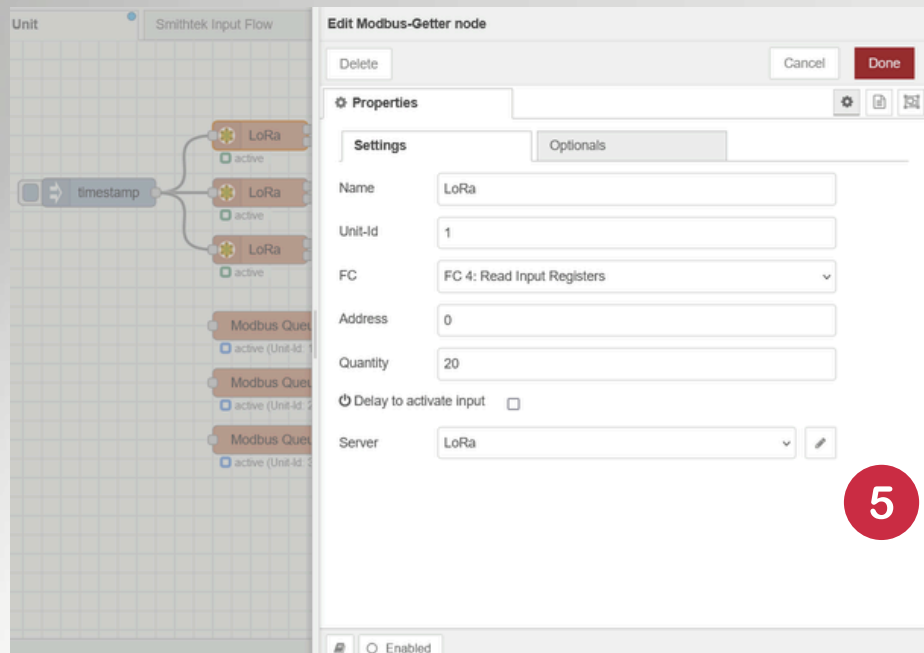
4. Click 'Done'

Mako & PassPort

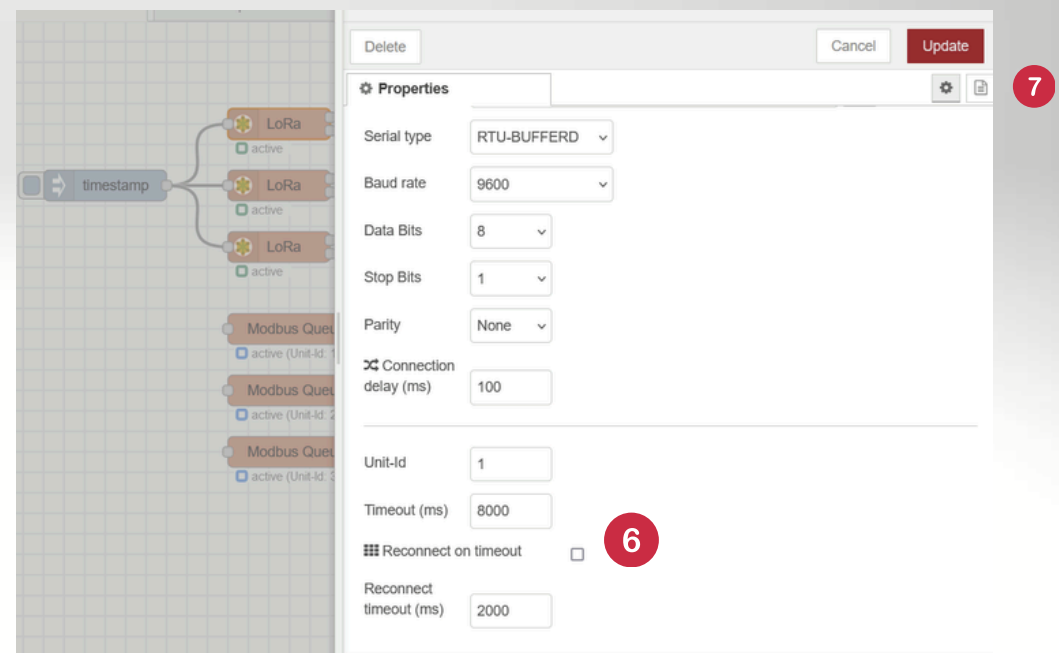
RF Integration

Simulation Topology:

Modbus Getter Node (Repeat Config for Each Node)



5. Now click the 'edit icon' on the right next to the LoRa Server



6. Scroll down and deselect 'reconnect on timeout'

7. Click 'Update'. Then click 'Done'

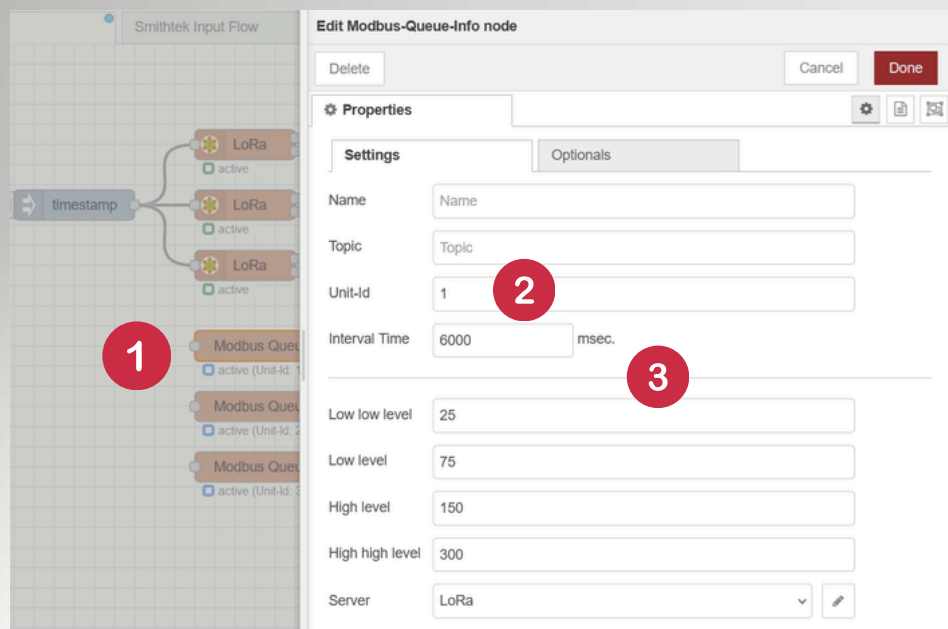
Make a basic change to the LoRa Antenna server configurations. The 'reconnect on timeout' is set to 'false' by default. But check every time to be sure!

Mako & PassPort

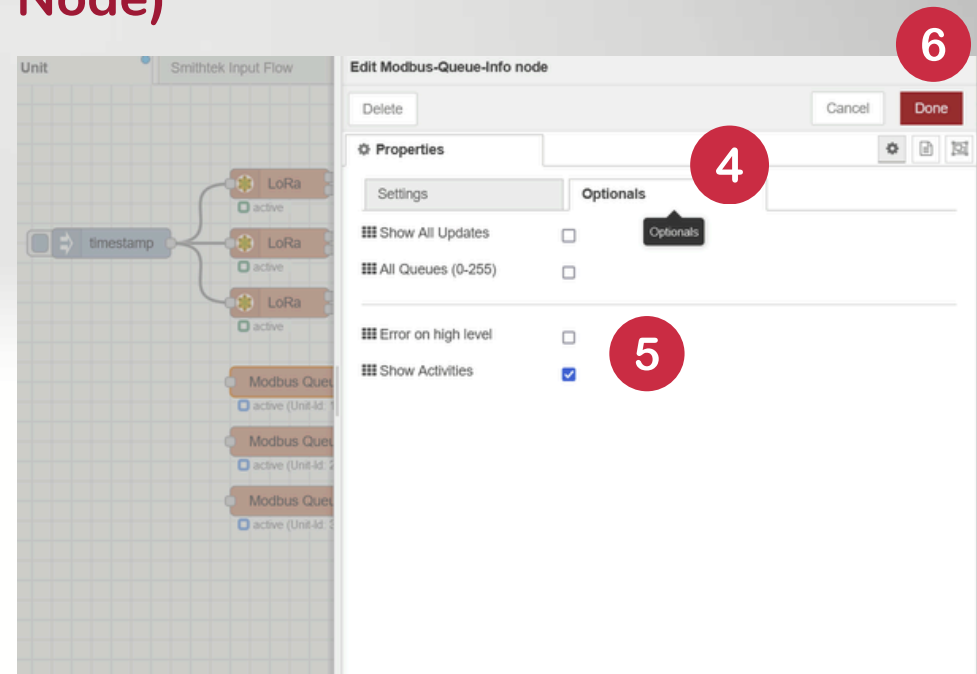
RF Integration

Simulation Topology:

Modbus Queue Node (Repeat Config for Each Node)



1. Click one of the 'Queue' nodes.
2. Change the Unit ID one for each Mako ID. These will automatically link to a 'Modbus Getter' node with the same ID.
3. Change the 'interval time' in ms.



4. Click 'Optional's'.
5. Match the settings in the image.
6. Click 'Done'.

The Queue nodes are optional for any serial or RF Modbus system but recommended when using LoRa.

They manage RF transmissions in the PassPort to prevent air data clashes or dropped packets caused by congestion. Acting as a timing mechanism, each poll waits for a response before moving to the next device. The interval (in milliseconds) defines how long each poll remains in the queue. Using Queue nodes, you can poll multiple Makos simultaneously, with responses returned one at a time within the set timeout, eliminating the need for separate delay nodes between Modbus devices.

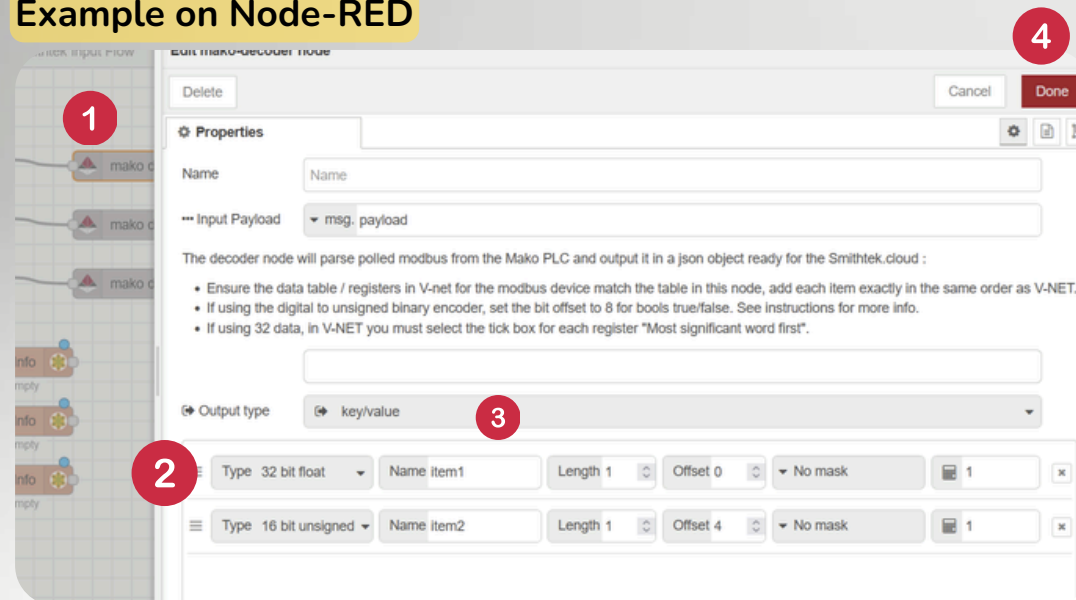
Mako & PassPort

RF Integration

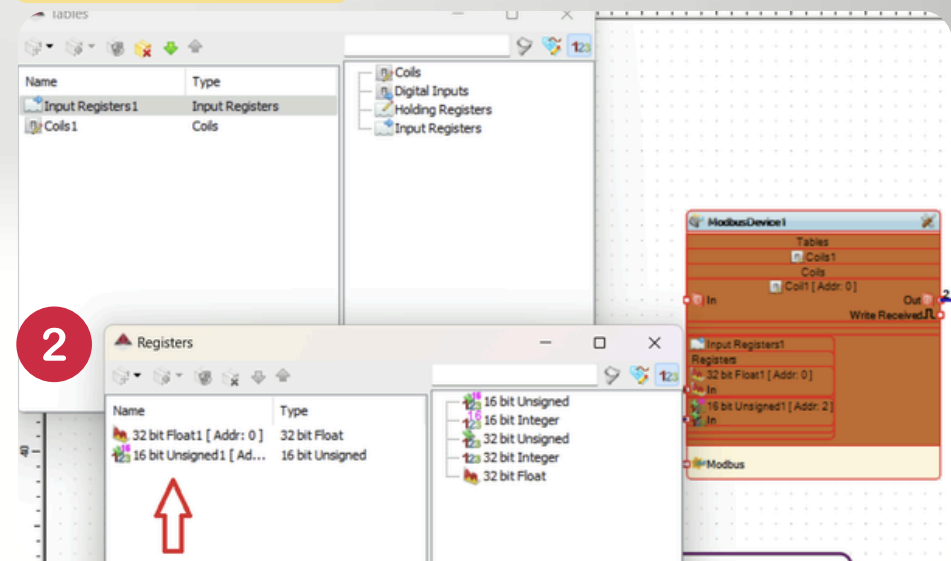
Simulation Topology:

Modbus Decoder Node (Repeat Config for Each Node)

Example on Node-RED



Example on V-NET



1. Click one of the Mako decoder nodes
2. Add items to the registers table. They have to be in the same order as your **V-NET** table. The decoder list has the same data types as V-NET
3. Rename your data types to something human readable like **'Tank Level'** flow rate Alarm signal. This will be added to the output payload
4. Click **'Done'**

The Mako decoder node processes the polled Modbus data, typically in decimal format, adds a unique name to each data point and outputs the data in JSON format. It is especially useful for decoding 32-bit values or floats, as it automatically handles the byte order, eliminating the need for custom buffer parsing scripts. ****Ensure that the register items in the Mako decoder node are listed in the same order as the registers in your V-NET tables for accurate data interpretation****

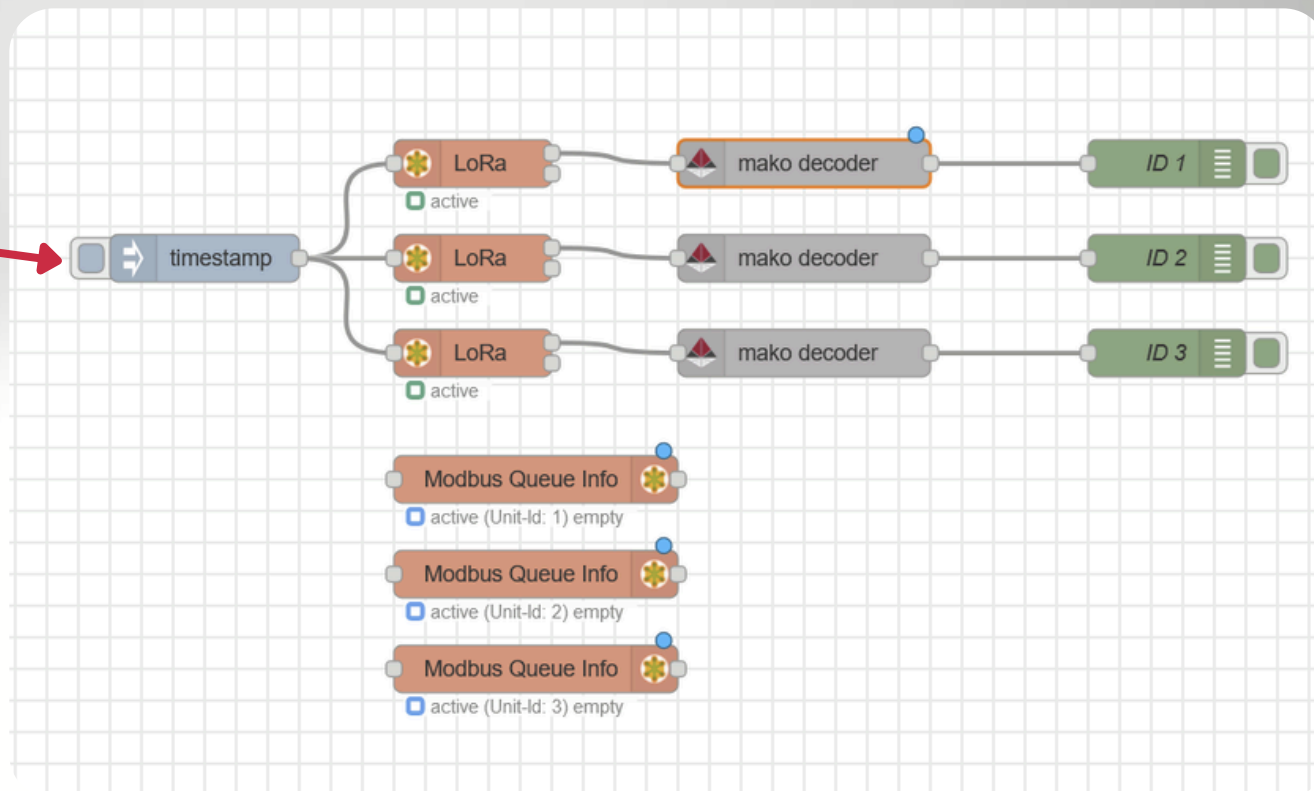
Mako & PassPort

RF Integration

Simulation Topology: Inject & Debug

Add an **'Inject node'** to trigger and read the polled values from each Mako.

Be mindful of air data rate limitations. For example, if the Inject node is set to trigger readings from three Makos, and each Mako takes around 5 seconds to respond due to air data speed, the Inject interval should be no shorter than the total polling time plus a 10% buffer.



Example:

Interval time = $(5 + 5 + 5) + 10\% = 16.5$ seconds

The response time for each Mako will vary depending on its distance from the PassPort. Fine-tuning this interval helps achieve the fastest and most reliable polling performance.

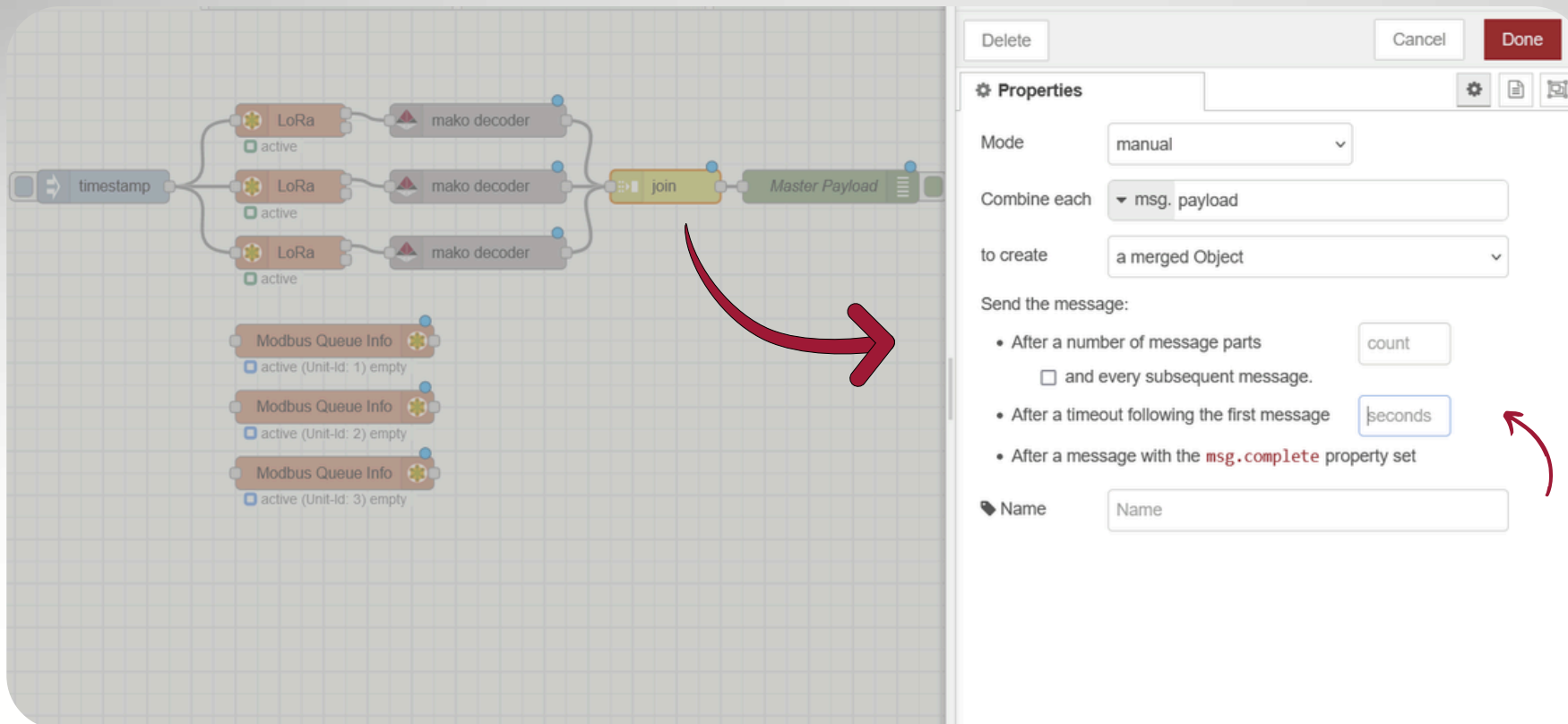
Mako & PassPort

RF Integration

Wrapping up You Payload

Merge Your Payloads

To merge all your Mako data into a single payload, use the 'Join' node and follow the settings shown in the image below. It is important to ensure that the 'Item' names in the Mako 'Decoder' node are unique. If you have three sites with identical Mako payloads, add a prefix to differentiate them, such as 'M1 tank level' and 'M2 tank level'. If you do not follow this rule, the Join node will drop duplicates. The timeout following the first message must be greater than your expected polling cycle. For example, if it takes 45 seconds to poll and retrieve the data from each Mako, set the timeout to more than 46 seconds.



Mako & PassPort

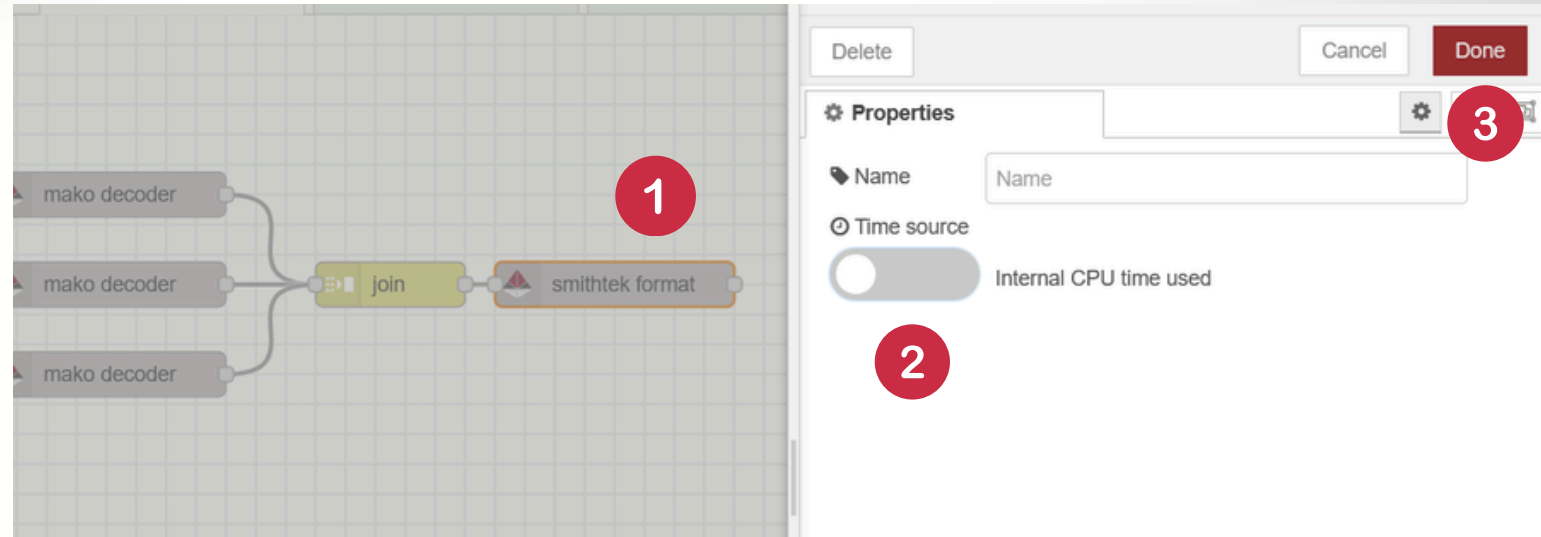
RF Integration

Wrapping up Your Payload Formatter Node

The Smithtek format node does two things:

1. It will convert scan your Modbus json data and convert any bool true/false to 1 and 0's. Smithtek.cloud only accepts 1 and 0's (ones & zeros).
2. It will add a timestamp to the data that can be used for local logging or the **'Store and Forward'** node. The time stamp source can be internal, the CPU or an external source like the **'Smithtek GPS'** node. If the GPS node is used, the Formatter node will detect its presence in the flow and use its GPS UTC Time.

1. Click the **'Smithtek Format node'**
2. Select your time source.
3. Click **'Done'**

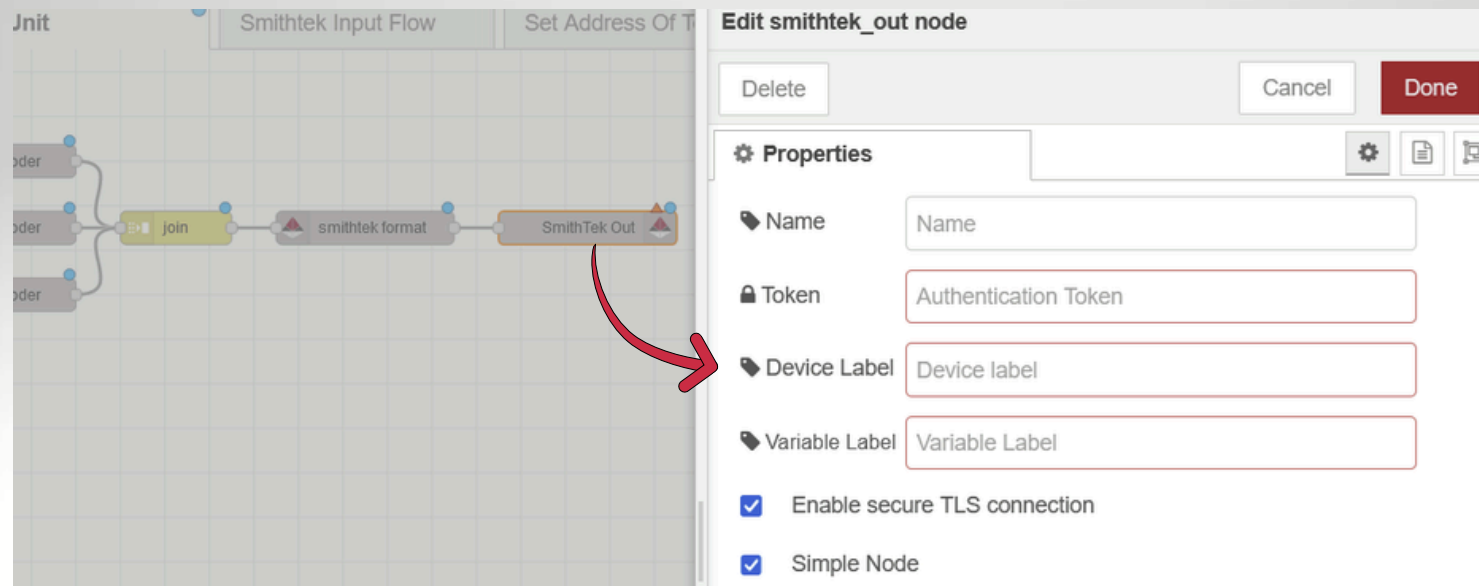


Mako & PassPort

RF Integration

Sending to Smithtek.Cloud

Smithtek.Cloud Node



If you want to send your data to Smithtek.Cloud you need to add two more nodes after the Join node.

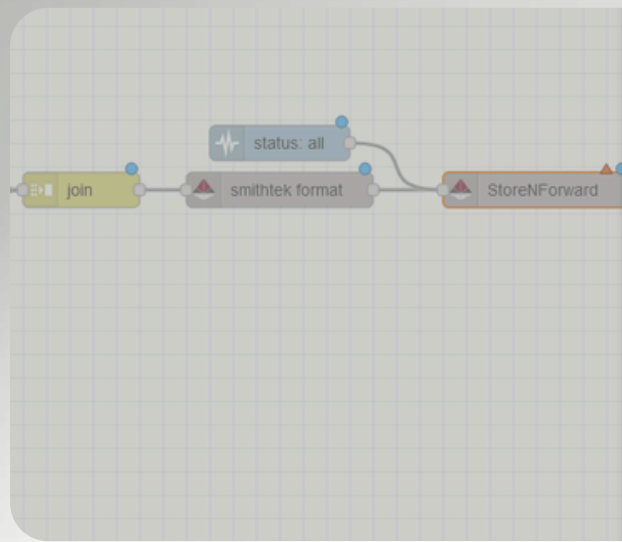
1. **Smithtek Format Node:** This node will take the Payload and change any bool to 1 or 0 and it lets you choose where the time source is from, by default it is the CPU, if set to external the time can come from the Smithtek GPS node or another source. Check the help link here for more details. <https://flows.nodered.org/node/smithtek-node-red-format>
2. **Smithtek Out Node:** This requires a active Smithtek account, a security token for your device and the device name, the device name is the serial number of your PassPort. This will wire to the very end of your flow. It will send the newly polled data at a time interval of your choice. 1 minute update is standard.

Mako & PassPort

RF Integration

Sending to Smithtek.Cloud

Smithtek.Cloud Node



This is the status name that will instruct the Passport, the coms is OK (forward the data to the cloud)

This is the status name that will instruct the passport the coms is Bad (Store the data to the PassPort)

To make your data resilient to communication outages, Smithtek includes one final tool, the **'Store & Forward'** node.

Place this node at the end of your flow. It relies on the **'Status'** node to detect connection changes. In the Status node, set the **'Smithtek Out'** node as the source. When internet connectivity is good, data is sent directly to the cloud. If communication is lost, the node automatically stores data locally with timestamps and packet order. Once the connection is restored, it uploads the stored packets to the cloud at the defined speed.

This node can be used not only with Smithtek Cloud but also with any IoT service.

Configuration fields:

- **Connected Status:** Type 'Connected' (if using the Smithtek Out node)
- **Disconnected Status:** Type 'Disconnected'
- **Path to SQLITE DB File:** Name your local file, e.g. 'my_device_file'
- **Speed (ms):** For Smithtek, set no lower than 300 ms. This defines how quickly stored data is uploaded once communication resumes.

Example: 1 hour of data sampled every minute with speed set to 1000ms, will take about 60 seconds to offload.

Mako & PassPort

RF Integration

Conclusion

By following the best practices in this guide, you will ensure efficient and reliable communication between the Mako PLC and the PassPort Gateway using Modbus over LoRa RF.

This setup delivers long-range connectivity, robust data transmission, and seamless IoT integration with Smithtek.Cloud. Regularly review and fine-tune your settings to maintain optimal performance and adapt to any network changes. Managing timing, handling offline conditions, and utilising the Store & Forward feature will further enhance system reliability.

Each Node-RED node includes a hyperlink to its web repository. To access it, open the Palette Manager, search for the node, and click the link icon to visit its webpage.

Thank you for choosing Smithtek solutions. If you have any questions or need assistance, our support team is here to help.
Happy fishing!

info@smithtek.com.au
www.smithtek.com.au
T: (08) 6118 9176