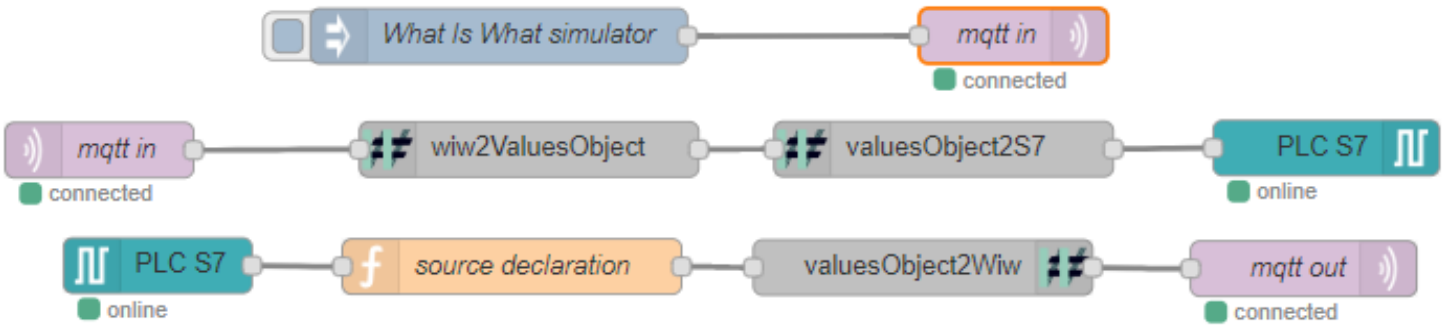
# Introduction

This document explains how to use *3* different nodes to communicate with a Siemens S7 PLC :

- *wiw2ValuesObject* used to cast a *What Is What* message into an object,
- *valuesObject2Wiw* used to cast an object into a *What Is What* message,
- *valuesObject2S7* used to cast an object into a message compatible with a n*ode-red-contrib-s7*'s *s7 out* node.

# The flow




## Third party dependencies

A *Mqtt broke*r must be running. See *mqttBroker* document for more information.

To read/write to *Siemens S7 PLCs*, dedicated nodes are needed. Here we are using: node-red-contrib-s7.

To test this example, we recommend to use a S7 PLC, but a simulator may be good enough. You can find one in Snap7 library, available here https://freefr.dl.sourceforge.net/project/snap7. Snap7 v1.4.1 has been successfully tested.

## Explanation

This flow demonstrates how to use the three presented nodes to send and receive values to and from *What Is What.*

### *First line*

The first line simulates a message from *What Is What.* It is needed only in this example, and is not relevant in production. The inject node creates a JSON object that could have been sent by *What Is What*.

## Second line

The second line shows the reception of this message from a *Mqtt broker*. The message is sent to a *Wiw2ValuesObject* node which will cast it into a new message (see *wiw2ValuesObject* documentation). Then *valuesObject2S7* casts it into a message compatible with S7 node, which writes data onto *PLC*.

## Third line

The third line reads values from *PLC*. A function node is used to set the *wiwSource* property to "machinedemo" (the name previously used in the entity declaration, see *entities2Wiw* example) to indicate to *What Is What* which entity has to be updated. Then, *valuesObject2Wiw* node creates a *What Is What* message to declare the values. This message is sent by *Mqtt* node.

### `Note`

Please make sure to use the right *Mqtt* broker settings in the *Mqtt* client setup.