

MODBUS CLIENT

Version 1.0.0
December 2, 2021

Modbus Client [X]

Setup Inputs Outputs

Inputs (Read Only)

Bits	Address	Description	Value
1	E 0 0000	Input 1	<input type="text"/>
2	E 1 0001	Input 2	<input type="text"/>
3	E 2 0002	Input 3	<input type="text"/>
4	E 3 0003	Input 4	<input type="text"/>
5	E 4 0004	Input 5	<input type="text"/>
6	E 5 0005	Input 6	<input type="text"/>
7	E 6 0006	Input 7	<input type="text"/>
8	E 7 0007	Input 8	<input type="text"/>

Registers

Address	Description	Value
1	E 10 000A	Analog 1 34120 8548
2	E 11 000B	Analog 2 50462 C51E
3	E 0 0000	0 0000
4	E 0 0000	0 0000
5	E 0 0000	0 0000
6	E 0 0000	0 0000
7	E 0 0000	0 0000
8	E 0 0000	0 0000

Modbus Client [X]

Setup Inputs Outputs

Outputs (Read Write) Clear Forced Outputs

Bits	Address	Description	Value	Force	Out
1	E 0 0000	Output 1	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
2	E 0 0000	Output 2	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
3	E 0 0000	Output 3	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
4	E 0 0000	Output 4	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
5	E 0 0000	Output 5	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
6	E 0 0000	Output 6	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
7	E 0 0000	Output 7	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
8	E 0 0000	Output 8	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>

Registers

Address	Description	Value	Force	Output
1	E 0 0000	Analog Output 1 8520 2148	0 0000 F	8520 2148
2	E 0 0000	Analog Output 2 32768 8000	0 0000 F	0 0000
3	E 0 0000	0	0 0000 F	0
4	E 0 0000	0	0 0000 F	0
5	E 0 0000	0	0 0000 F	0
6	E 0 0000	0	0 0000 F	0
7	E 0 0000	0	0 0000 F	0
8	E 0 0000	0	0 0000 F	0



MODBUS CLIENT

allows Q-SYS to interface with external Modbus Server devices over standard TCP/IP communication. This allows Q-SYS to receive inputs and control outputs of automation equipment that supports the Modbus over TCP protocol. This can greatly expand the IO capabilities of a Q-SYS system by allowing both digital and analog signals to be easily integrated into a design.

The plugin offers up to 999 each of Input Bits, Input Registers, Output Bits, and Output Registers offering a staggering number of control IO points – hundreds more than a typical control system. Inputs are polled from the Modbus Server Device up to 20 times per second, configurable in 50ms increments. Outputs are updated up to 30 times per second. All inputs and outputs have control pins that can be connected to other components in the design allowing for complex control capabilities.

Both Input and Output Bits are represented as toggle buttons for easy control pin wiring. Registers are represented as integer fields with possible values from 0 to 65,535, and as hexadecimal string fields with values from 0x0000 to 0xFFFF.

Each Input and Output in the plugin may be configured for any IO address and do not have to be sequential. Unused IO can be disabled to reduce clutter and confusion. IO may also be temporarily disabled while testing or commissioning is ongoing. This provides maximum flexibility for working with larger, more complex systems.

Outputs offer a “Force” feature which allows manual override of Server Device’s outputs and ignores data from the control pin. When a Force is active, the provided value will be applied to the output allowing for easier system testing, commissioning, and temporary bypass.

Refer to the Plugin Guide Notes for information regarding protocol compatibility.

Warning

The Modbus Client plugin is not intended to be used in manufacturing, motion, or safety applications, or any other applications where a software failure could result in damage to equipment or property, injury to persons or animals, or loss of life. The plugin is only intended to expand the IO capabilities of a Q-SYS system for control usage within an entertainment-related permanent installation in which General Purpose IO is not sufficient.

While Programation LLC attempts to mitigate software errors, we do not guarantee that our software is free from errors as outlined in the Software License Agreement. By using this plugin, you assume all risk associated with any consequences that may arise if the plugin should malfunction, including but not limited to, a malfunction of the Q-SYS system, the scripting environment, or with the intellectual property of Programation LLC.

Important Disclaimer

Due to data processing limitations within the Q-SYS control engine, information may only be processed up to 30 times per second, or about every 33ms. Further, the Modbus Client plugin can only poll and refresh input data from a Server Device up to 20 times per second or every 50ms. These refresh rates are far too slow for use in any motion or safety applications. These refresh rates may also be considered too slow for certain control applications. Careful consideration should be given when designing and implementing a system that relies on this plugin or third-party devices. These performance rates are approximately equal to the refresh rates of native Q-SYS GPIO which does suffice for most general-purpose control applications.

Properties

Input Bits

an integer between 0 and 999 that defines the number of input bits available for configuration

Input Registers

an integer between 0 and 999 that defines the number of input registers available for configuration

Output Bits

an integer between 0 and 999 that defines the number of output bits available for configuration

Output Registers

an integer between 0 and 999 that defines the number of output registers available for configuration

Is Required

a boolean that sets whether the target device is required for system functionality

Registered Owner

a string that lists the license holder name of the plugin registration for this installation

Registered Project

a string that lists the licensed project name of the plugin registration for this installation

Registered Key

a string that lists the license key code of the plugin registration for this installation

Status

displays various status messages to aid in monitoring the connection with the Server Device. Communication errors will show as a fault. When a force is present on an output, the status will show as Compromised as a reminder that the system has been overridden.

Server Port

sets the TCP communication port number to use when connecting to the Server Device

Server Address

sets the IP address to use when connecting to the Server Device

Bus Address

sets the Modbus Bus Address (or Device ID) of the Server Device. This is useful if more than one Server exists at the same IP address. Refer to the Notes section for more details.

Polling Steps

sets the polling frequency of inputs from the Server Device in increments, or steps, of 50 milliseconds

Polling Interval

displays the time of each input poll

Input Enable

a toggle button that sets whether this Input is used and should be polled

Input Dec Address

an integer field that ranges from 0 to 65,536 that sets the memory address of the input in decimal

Input Hex Address

a string field that ranges from 0x0000 to 0xFFFF that sets the memory address of the input in hexadecimal

Input Description

a text field that provides human-readable labeling of each input

Input Bit Value

a toggle button that represents the state of the input on the Server Device

Input Register Dec Value

an integer field that ranges from 0 to 65,535 that represents the value of the register on the Server Device in decimal

Input Register Hex Value

a string field that represents the register value in hexadecimal, ranging from 0x000 to 0xFFFF

Output Enable

a toggle button that sets whether this Output is used and should send updates to the Server Device

Output Dec Address

an integer field that ranges from 0 to 65,536 that sets the memory address of the output of the Server Device in decimal

Output Hex Address

a string field that ranges from 0x0000 to 0xFFFF that sets the memory address of the output in hexadecimal

Output Description

a text field that provides human-readable labeling of each output

Output Bit Value

a toggle button that sets the desired state of the output of the Server Device

Output Bit Force Value

a toggle button that sets the value of the output in the forced state

Output Bit Force Enable

a toggle button that overrides the output bit's value with the value specified by the Bit Force Value toggle button. When active, the output will only follow the forced value.

Output Bit Output

a toggle button that represents the actual output state, after considering the forced state

Output Register Dec Value

an integer field ranging from 0 to 65,535 that sets the desired decimal value of the output of the Server Device

Output Register Hex Value

a string field ranging from 0x0000 to 0xFFFF that indicates the hexadecimal value of the decimal value

Output Register Dec Force Value

an integer field ranging from 0 to 65,535 that sets the decimal value of the output in a forced state

Output Register Hex Force Value

a string field ranging from 0x0000 to 0xFFFF that sets the hexadecimal value of the output in a forced state

Output Register Force Enable

a toggle button that overrides the output register's value with the value specified by the Register Force Dec/Hex Value fields. When active, the output will only follow the forced values.

Output Register Dec Output

an integer field ranging from 0 to 65,535 that represents the actual output value in decimal, after considering the force state

Output Register Hex Output

a string field ranging from 0x0000 to 0xFFFF that represents the actual output value in hexadecimal, after considering the force state

IMPLEMENTED MODBUS FUNCTIONS

The Modbus Client plugin is compliant with the official Modbus TCP protocol and uses the methods:

- 0x02 (Read Discrete Inputs)
- 0x04 (Read Input Registers)
- 0x15 (Write Multiple Coils)
- 0x16 (Write Multiple Registers)

Polling uses only methods 2 and 4 and writing only uses methods 15 and 16. If your Server Device does not support this, please contact us for assistance.

I/O ADDRESSING – SUPPORTING BOTH MODBUS TCP AND JBUS

Modbus TCP is a somewhat modified version of the original Modbus Serial communication protocol. Modbus TCP has certain limitations to the number of addresses available and the quantity of coils and registers that can be communicated in a single request. This is to keep a TCP message under the default MTU size. JBus, on the other hand, is more relaxed with these requirements. The most noticeable difference between the two protocols for system programmers is that Modbus TCP is 0-based in its memory addressing while JBus is 1-based.

The Modbus Client plugin is designed to not be concerned with whether the requests are 0-based or 1-based. It is up to the Q-SYS programmer to define the memory addresses for their use-case appropriately. The plugin does not apply any offset to the address numbers listed. If no I/O are addressed as 0, and a request is issued for memory address 0, then the request will fail.

Consequently, the plugin allows I/O to be addressed as high as 65,536 or 0x10000. This is to maintain compliancy with JBus, however it is an illegal data address for the Modbus protocol. Since all hexadecimal values are limited to 4 digits (2 bytes), the plugin will show a decimal address of 65,536 but a hexadecimal address of 0xFFFF.

Hexadecimal readouts of address and value fields are just for the programmer's convenience and do not have any practical effect on the data transmission. The plugin only considers decimal values for requests and transmissions. When a hexadecimal value is entered or is changed, however, the corresponding decimal field will also be updated to the correct value.

While the plugin will attempt to receive either Modbus TCP or JBus style messages without additional user configuration, it is recommended that messages stay under the default MTU size of 1500 bytes, otherwise the message may be split into multiple packets and the request may fail to be processed correctly.

NON-SEQUENTIAL AND NON-CONTIGUOUS ADDRESSES

As the Modbus Client plugin allows any Input or Output to have any address, addressing may be non-sequential and non-contiguous. This allows flexibility in only creating I/O that is needed at any address. Several installations typically prefer to use logical numbering conventions for various groups of I/O which often skips over blocks of addresses. The plugin supports this without having to have a large number of unused controls.

When a poll request is made for inputs, all inputs are arranged first by bits, then by registers, in ascending order based on their Modbus Address, regardless of line number in the plugin. Contiguous regions of address numbers are then requested in a single message. Each non-contiguous break will result in a separate message being transmitted.

A large number of requests may result if many non-contiguous input addresses are requested. Performance may degrade if the requests cannot be fulfilled faster than the specified polling time. For this reason, the polling time should be set to a slower interval when using a large number of I/O, depending on the performance of the Server Device.

I/O PRIORITY

All enabled inputs are polled on every polling interval. Outputs are only transmitted when a control is changed or when the plugin connects to the Server Device. As Q-SYS only updates controls at a rate of 30 hertz, this becomes the frequency limit to updating or controlling outputs. Transmitting output write messages and input read requests are generated asynchronously within the plugin. To prevent overwhelming the server device with requests, only one message may be in flight at a time. Messages are queued within the plugin and are only transmitted when a response is received. The queue honors output write requests first before sending poll requests from the queue. This allows outputs to be more time accurate at the minor expense of delaying input triggers.

BUS ADDRESS

Traditional Modbus is a serial line protocol that supported multiple devices to be connected to the same line. Therefore, the protocol includes a “Bus Address” field which allows requests to target a specific device on the communication bus. Modbus TCP offers the same functionality for cases where multiple servers share the same IP address. The default for most devices is a Bus Address of zero.

Notes

Multiple instances of the Modbus Client plugin can run in the same Q-SYS design simultaneously provided that they are connecting to different Server Devices. It is not recommended for two Client plugins to attempt to connect to the same Server Device.

Change Log

V1.0.0 – DECEMBER 2, 2021

Original release