

Vergleich Modbus TCP zu KNX IoT stack

Criteria	Modbus TCP	KNX netIP with properties (M_Prop Services)	KNX Point API with properties (https://gitlab.knx.org/public-projects/knx-iot-point-api-schema/-/blob/work_in_progress/README.md?ref_type=heads#enrollment-step-1)
Connection	IP, today IP v4	IP, today IP v4	IP v6
Communication type	Client/Server (P2P) via TCP	Client/Server (P2P) via TCP	Client/Server (P2P) via TCP
IP Discovery	None	KNXnet/IP Search Request Via UDP Multicast	- Device discovery based on DNS-SD and mDNS (RFC 6762) -Multicast resource discovery in CoAP (CoRE) is accomplished by using the "/.well-known/core" Semantic discovery based on KNX Volume 7 and EN50090-6-2 (Functional Blocks and Points)
Port No.	502	3671	CoAP port configuration [RFC7252]
Security	TLS optional, Barely used Requires certificates	KNX IP Security Key exchange via Diffie-Hellman, password per connection and authentication code. VDE approved	<input type="checkbox"/> OSCORE (RFC 8613) Application Layer Security for multicast and low latency communication <input type="checkbox"/> SPAKE2+ (RFC 9383) for password-based authentication (OSCORE key configuration) <input type="checkbox"/> TLS Transport Layer Security (RFC 5246) for configuration (e.g. OSCORE keys) <input type="checkbox"/> Software update capability
Data types	Modbus registers holding 1 bit or 2 byte each.	KNX Properties can have a data length of 1..20 bytes. Defined data point type for all typical values, e.g. float.	KNX Properties can have a data length of 1..20 bytes. Defined data point type for all typical values, e.g. float.

	<p>Data types with more than 2 bytes have to be mapped to several registers.</p> <p>No specification for data types.</p>		
Data arrays	None	Each property can represent an array with up to 4094 elements.	Each property can represent an array of elements.
Data point organization	<p>Separated register ranges depending on technical differences:</p> <p>1 bit vs. 2 bytes</p> <p>Read-only vs. RW</p>	<p>Properties are organized in so-called KNX interface objects.</p> <p>Each object has an ID and can hold up to 255 properties, for applications up to 150 IDs can be defined plus 50 IDs for manufacturer specific IDs.</p> <p>An interfaces type can have multiple instances.</p>	<p>CBOR and JSON Parameter Property</p> <p>Properties are organized in so-called KNX interface objects.</p> <p>Each object has an ID and can hold up to 255 properties, for applications up to 150 IDs can be defined plus 50 IDs for manufacturer specific IDs.</p>

Data point addressing	Based on register number	Based on interface object ID and object instance number together with property ID (PID) and start index for array elements.	URIs
Multi access	Different Multi read and write services for the different register classes.	Multi read and write not on property level but for array elements.	Pub/sub
Error handling	Error codes	Error codes to detect non-existing properties, access violation etc.	Response codes [RFC7252]
Data type discovery	None	Possible with M_PropDesc Read service.	Metadata Query Parameter

Further details may be found via following links:

[Public Projects / KNX IoT Point API Scheme · GitLab](#)

Other useful links (Point API, 3rd party API...) can be found here:

[Downloads – KNX Association](#)