Schneider Electric Control Expert Connector

PLC's Quantum, Momentum M340 and M580, UnityPro

- Name: **ControlExpert**
- Version: 1.7.0.0
- Protocol: MODBUS RTU, ASCII and TCP
- Interface: TCP/IP and Serial
- Runtime: Multiplatform
- Configuration:
  - Devices / Protocols

Overview
Supported Operands
Channel Configuration
Node Configuration
Points Configuration
Troubleshoot
Driver Revision History

Overview

Communication with PLC and IO devices compatible with Modbus Open Standard protocol. It operates as a Master on TCP/IP or serial networks. The communications blocks are dynamically created according to the pooling cycle defined on the AccessType for each Device Point.

Supported Operands

Operand	Read	Write	Data Type	Address size
0 – %Q or %M - Coils	**Yes**	**Yes**	Bit	1 bit
1 – %I - Input Status	**Yes**	-	Bit	1 bit
3 – %IW - Input Registers	**Yes**	-	Word	2 bytes
4 – %MW - Holding Registers	**Yes**	**Yes**	Word	2 bytes

Channel Configuration

Protocol Options

The driver uses 4 ProtocolOptions fields (all positional via DrvParams):

- `DrvParams[0]` = **MsgFormat** (string): Encoding/protocol format — `"RTU TCP"`, `"RTU"`, or `"ASC"` (ASCII). Default: `"RTU TCP"`.
- `DrvParams[1]` = **SingleWrite** (string): Write mode — `"Use Single Write"` for individual writes, anything else for block writes. Default: block write.
- `DrvParams[2]` = **WriteFFonCoils** (string): If `"FF"`, writes 0xFF instead of 0x01 for coil ON state (required by some PLCs). Default: 0x01.
- `DrvParams[3]` = **OffsetAddress** (boolean): If true, adds +1 offset to Modbus addresses (for PLCs that use 1-based addressing). Default:

Encoding: Determines how the information will be packed into the message fields and decoded. The options are:

- **RTU**: Remote Terminal Unit mode, where each 8–bit byte in a message contains two 4–bit hexadecimal characters.
- **ASCII**: The message is encoded in ASCII mode, where each 8–bit byte in a message is sent as two ASCII characters.
- **RTU TCP**: The default transmission mode when the message is carried on a MODBUS TCP/IP network. It contains information to allow the recipient to recognize message boundaries even if the message has been split into multiple packets
- **SingleWrite:** Indicates the driver behavior for the writings with only one item:
  - **Use block write** : The driver uses the 0x0F command for Coils, or the 0x10 command for Holding Registers
  - **Use single write:** The driver uses the 0x05 command for Coils, or the 0x06 command for Holding Registers

Settings

Serial and MultiSerial channels:

- Default configuration for ASCII mode:
  - **DataBits**: 7
  - **StopBits**: 1 if parity is used; 2 if no parity is used
- Default configuration for RTU mode:
  - **DataBits**: 8
  - **StopBits**: 1 if parity is used; 2 if no parity is used

Set the other fields according to your Serial or MultiSerial port configuration

TCP/IP channels:

- **NodeConnections**: Defines the maximum number of parallel requests that will be sent to each node (asynchronous communication)

Node Configuration

Station Configuration

- **SlaveId**: Set this field with the address of the slave device in the network. They can be addressed from 1 to 247 for serial nodes, or 0 to 255 for TCP/IP nodes. The address 0 is used for the broadcast.

Serial channels:

- Station syntax: ***<SlaveId>***

E.g.: ***1***

MultiSerial channels:

- Station syntax: ***<Com Port> ; <SlaveId>***

Where:

- ***<Com Port>***: the serial port number

E.g.: ***com1 ; 1***

TCP/IP channels:

- Station syntax: ***<IP address> ; <Port number> ; <SlaveId>***

Where :

- ***<IP address>***: IP address of the slave device in the modbus network
- ***<Port number>***: TCP port where the slave device is listening (default is **502**)

E.g.: ***192.168.1.101 ; 502 ; 1***

Points Configuration

The syntax for the UnityPro communication points is: ***<Operand><Address>***

Where:

- ***<Operand>***: indicates the memory area, the valid values are:

Operand
0 – %Q or %M - Coils
1 – %I - Input Status
3 – %IW - Input Registers
4 – %MW - Holding Registers

For more information about the valid operands, see the **Supported Operands** table aforementioned.

- ***<Address>***: indicates the data address in the memory area, from **1** to **65535**

E.g.: ***400001*** (Operand = ***4*** (%MW), Address = ***1***)

Troubleshoot

Modbus protocol error codes		
Error	**Name**	**Description**
1	ILLEGAL FUNCTION	The function code received in the query is not allowable.
2	ILLEGAL DATA ADDRESS	The data address received in the query is not allowable.
3	ILLEGAL DATA VALUE	A value contained in the query data field is not allowable.
4	SLAVE DEVICE FAILURE	Error while attempting to perform the requested action.
5	ACKNOWLEDGE	Request accepted, but a long duration of time will be required.
6	SLAVE DEVICE BUSY	The slave is engaged in a long–duration program command.
7	NEGATIVE ACKNOWLEDGE	Cannot perform the program function received in the query.
8	MEMORY PARITY ERROR	Parity error in the extended memory.

Driver Revision History

UnityPro Revision History	
Version	**Notes**
1.7.0.0	Initial release on new documentation standards.

In this section...