ASCII – Generic ASCII Master Protocol

The ASCII driver implements communication with any device using the ASCII protocol 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.

Summary Information

Communication Driver Name: ASCII

Implementation DLL: T.ProtocolDriver.ASCII.dll

Protocol: Generic ASCII

Interface: TCP/IP and Serial

PLC types supported: Any PLC compatible with the ASCII protocol

---

Channel Configuration

Protocol Options

BlockSize: Defines the maximum amount of characters. The default value is 250

StartChar: Defines the start character of incoming messages

EndChar: Defines the end character of incoming messages

Settings

Serial and Multi-Serial channels:

Default configuration:

- **BaudRate**: 9600
- **DataBits**: 8
- **StopBits:** 1
- **Parity:** None

TCP/IP channels:

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

Node Configuration

Station Configuration

Serial channels: Nothing

TCP/IP channels:

- Station syntax: <IP address>;<Port number>
- Where:

  <IP address> = The IP address of the slave device in the ASCII network

  <Port number> = The TCP port where the slave device is listening (default is 502)

Ex: 192.168.1.101 ; 502

Point Configuration

The syntax for the ASCII communication points is: <SequenceId>:<SizeOf>

Where:

- **<SequenceID>** indicates the order of the data where the TX message will be created
- **<SizeOf>** indicates the amount of characters from the configured tag

---

Access Point

Read and Write commands will generate the same TX message

Incoming messages from the device will be treated as an Unsolicited Message

---

Example

Point 1

TagName: TagA (value equal ABCD)

Address: 0:3

Point 2

TagName: TagB (value equal 1234)

Address: 2:3

Point 3

TagName: TagC (value equal abcd)

Address: 1:4

The TX message created is: ABCabcd123

Troubleshoot

The status of the driver's execution can be observed through the diagnostic tools, which are:

- Trace window
- Property Watch
- Module Information

The above tools indicate if the operations have succeeded or have failed. A status of 0 (zero) means communication is successful. Negative values indicate internal driver errors, and positive values indicate protocol error codes.