Toyopuc PC10G Ethernet Connector

JTEKT Toyopuc PC10G controllers over Ethernet

- Name: **Toyopuc**
- Version: 1.0.0.1
- Protocol: Computer Link (binary, Ethernet)
- Interface: TCP/IP
- Runtime: Multiplatform
- Configuration:
  - Devices / Protocols

Overview
Controller Configuration
Channel Configuration
Node Configuration
Points Configuration
Troubleshooting
Driver Revision History

Overview

The Toyopuc driver reads and writes controller registers on JTEKT Toyopuc PC10G devices using the Computer Link protocol over the controller built-in Ethernet port. Communication blocks are dynamically created according to the polling cycle defined on the AccessType for each Device Point.

The driver addresses data registers, link registers, present-value and special registers, the relay areas at word granularity, and the expansion data register area. Individual programs on a multi-program controller can be addressed explicitly.

This connector uses the binary command set of the Ethernet port. It is not the ASCII frame used by the RS-422 serial Computer Link, which is a separate protocol on the same controller family.

Controller Configuration

Before connecting, the controller built-in Ethernet port must be configured in PCWin with an IP address and a listening port:

- **Own Node IP Address**: the IP address the driver will connect to.
- **Own Node Port No.**: the TCP port the controller listens on.
- **Open Protocol**: the connection must be set to a TCP passive open so the controller accepts the incoming connection.

A connection configured as a non-specified passive open accepts one client at a time. If another node is already connected on that port, the existing connection is reset.

Channel Configuration

Protocol Options

- **BlockSize**: maximum bytes per request, two bytes per word. Default is **240**, which is 120 words. The limit is 1018 bytes on the PC10G built-in Ethernet port. An FL/ET-T-V2 module on a PC3J or PC2J CPU caps at 512 bytes, so lower this value if the controller reports error 0x41.
- **TraceFrames**: when enabled, every transmitted and received frame is written to the Trace Window as hexadecimal. Intended for diagnostics. Default is **false**, and it should be left disabled in production.

Node Configuration

Station Configuration

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

Where:

- ***<IP address>***: IP address of the controller built-in Ethernet port, matching Own Node IP Address in PCWin.
- ***<Port number>***: TCP port the controller is listening on, matching Own Node Port No. in PCWin.

E.g.: ***192.168.1.10; 2000***

This protocol carries no station number, so the node requires only the address and port.

Points Configuration

The syntax for the Toyopuc communication points is: ***<Operand><Address>***, written as a single token with no separator.

Where:

- ***<Operand>***: the memory area, optionally prefixed with a program number.
- ***<Address>***: the word offset within the area, written in **hexadecimal**, exactly as displayed in PCWin.

Program Prefix

Prefix	Description
none	The program selected by the controller link parameters. The most efficient option, and the default.
P1-	Program 1, addressed explicitly.
P2-	Program 2, addressed explicitly.
P3-	Program 3, addressed explicitly.

On a controller running a single program, an unprefixed address and the matching program prefix return the same data. The **U** area is shared rather than per-program and therefore takes no prefix.

Memory Areas

Area	Description	Address range (hex)
D	Data register	0 to 2FFF
R	Link register	0 to 7FF
N	Present value register	0 to 1FF
S	Special register	0 to 3FF
M	Internal relay, word access	0 to 7F
XY	Input and output relay, word access	0 to 7F
L	Link relay, word access	0 to 7F
T	Timer and counter, word access	0 to 1F
V	Special relay, word access	0 to F
K	Keep relay, word access	0 to 2F
U	Expansion data register	0 to 7FFF

The relay areas are accessed one word at a time, matching the word-address notation used by PCWin. A relay word address shown as **M20W** in the programming software is written as **M20** here.

Referencing the Node

In the Devices Points table the **Node** column takes the fully qualified node reference, ***Node.<NodeName>***, not the bare node name. A node called **Toy opucNode1** is referenced as **Node.ToyopucNode1**. This matters when points are imported from a file rather than picked in the editor.

Examples

Address	Reads
D0	Data register D0000 in the link-parameter program
D100	Data register D0100, hexadecimal offset
P1-D160	Data register D0160 in program 1
P2-R0	Link register R0000 in program 2
M20	Internal relay word M20W
U106	Expansion data register U0106

Bit Access

Individual bits are addressed through the **Modifiers** column of the Devices Points table, not through the address itself. Set **Bit** to a value from 0 to 15 and leave the address as the enclosing word.

	Where it works	How it works
Bit read	Every memory area	The connector reads the enclosing word as part of its normal block poll and the bit is taken from it. No extra request is sent, so a bit read costs nothing beyond the word read that was already happening.
Bit write	**M, XY, L, T, V and K only**, and only on an operand with **no program prefix**	A dedicated single-bit command sets or clears one bit in the controller. The other fifteen bits of the word are never touched, so there is no read-modify-write cycle that could overwrite a bit the ladder program changed in the meantime.

Two restrictions follow from the protocol itself:

- The registers **D**, **R**, **N**, **S** and the expansion register **U** have no bit addresses, so they cannot be bit-written. Use a word-level tag on those areas.
- There is no program-scoped single-bit command, so **P1-**, **P2-** and **P3-** prefixed operands cannot be bit-written. Use the unprefixed form, which addresses the program selected by the controller link parameters. Confirm that this is the program you intend before substituting one form for the other.

A bit write carries exactly one bit per request, so a bit-modified point must be the only item in its write block. If another writable point in the same area falls inside the **BlockSize** window, the write is refused. Keep bit-modified points clear of other writable points in the same area, or write them in separate scans.

Every refused bit write names its reason in the Trace Window and sends nothing to the controller.

Troubleshooting

When a request is rejected, the controller returns an error code that is written to the Trace Window together with its description. The most frequently seen codes are:

Code	Meaning	Usual cause
0x40	Address, or address plus data count, out of range	The point address exceeds the area, or a block runs past the end of the area. Check the address range table above.
0x41	Word or byte count out of range	BlockSize is larger than the controller accepts. Reduce it.
0x34	Access prohibited by configuration	The controller is configured to deny access to that area.
0x31	Write prohibited during sequence operation	A write was attempted against a region the running program protects.

If no reply is received at all, confirm the controller port is open and that no other client is holding the connection.

Enable **TraceFrames** on the channel to capture the exact bytes exchanged when reporting a communication problem.

Driver Revision History

Toyopuc Revision History	
Version	**Notes**
1.0.0.1	Communication rewritten to the binary command set of the PC10G built-in Ethernet port, verified on a live controller (the previous version could not communicate with it); addresses are entered in hexadecimal as in PCWin (for example D100), with optional program prefixes P1-/P2-/P3- and the U expansion register; single bits on relay areas can be written without changing the rest of the word; controller error codes are reported correctly; the station number is no longer configured, the default port is 2000, block size is set in bytes and a frame-trace diagnostic option was added; the Designer help icon now opens the connector page.
1.0.0.0	Initial release. Word read and write over the Ethernet Computer Link binary command set, with per-program scope and expansion data register support.

In this section...