Generic User Protocol

Supported device types and versions
Communication line configuration
Communication station configuration
I/O tag configuration
Literature
Changes and modifications
Document revisions

Supported device types and versions

Generic User Protocol is intended to support an application-level implementation of simple and lightweight communication protocols directly in the D2000 Event process via an ESL script or Java code. It supports several types of lines including line-redundant and system-redundant ones. Data read from the communication are published into input I/O tags (IN) without waiting for any confirmation from the ESL script's side, therefore it is recommended to implement data handling via Server Event using the action ON CHANGE, eventually by a trigger event with a request queue configured, eventually by enabling multiple executions of the script (action ENABLE) so that all the published data is handled even under heavier load.

Communication line configuration

Category of communication line:

- Serial, Serial Line Redundant, Serial System&Line Redundant
- SerialOverUDP Device Redundant, SerialOverUDP Line Redundant, SerialOverUDP System&Line Redundant
- MOXA IP Serial Library
- RFC2217 Client
- TCP/IP-TCP, TCP/IP-TCP Redundant
  Note: if a TCP/IP-TCP or TCP/IP-TCP Redundant line has all stations set to StOFF, TCP connection will be closed. Thus it is possible to control TCP communication from the ESL script using a STSTAT tell command.

Communication line protocol parameters

Configuration line dialog box - tab **Protocol parameters**.
They influence some of the optional protocol parameters. The following line parameters can be set:

Table 1

Keyword	Full name	Description	Unit	Default value
RT	Read Wait Timeout	Waiting between individual data readouts from the communication. If no data was received during this period and previously some data was read, it will be published as a value of input I/O tag IN. If some data was received during this period, it will be appended to a buffer, until a timeout occurs or the buffer becomes full (see a parameter Read Size). If a parameter Log Each Read is set to True, data will be published immediately regardless of the value of parameter *Read Wait Timeout*.	sec. mss	1.000
RS	Read Size	Maximum size (in bytes) of input data. Longer data will be published in several chunks.	-	1024
LE	Log Each Read	If this parameter is set to True, data will be published immediately regardless of the value of parameter Read Wait Timeout.	-	False
LF	Log Format	Format of data traces in log files: "0 - Hexa log" or "1 - Text log". Setting the correct format helps to make a communication log file more readable depending on whether the specific protocol is text-oriented or binary by dumping the data in text or hexadecimal format.	-	0 - Hexa log

SL	Single Log	Setting this parameter to True will make all logs to go to a single file on redundant lines. Setting this parameter to False means that two log files will be created for the primary/secondary line (Serial Line Redundant, SerialOverUDP Line Redundant, TCP/IP-TCP Redundant), or four log files will be created for the primary /secondary line of A/B system (Serial System&Line Redundant, SerialOverUDP System&Line Redundant).	-	False

Communication station configuration

- Communication protocol Generic User.
- The station address is not configured; it is recommended to configure a single station per line, but multiple stations are supported too. In this case, when data is received from the communication, it will be published via input I/O tags IN on all configured stations. Likewise, it will be possible to use output I/O tags OUT on any configured station.

I/O tag configuration

Possible value types of I/O tag: **TxtI, TxtO**.

Input I/O tags:

- The input I/O tag has address **IN**.
  Note: on redundant lines an I/O tag with address **IN** receives values read from any line (primary/secondary), eventually from any system (A/B).
- on line-redundant lines (Serial Line Redundant, SerialOverUDP Line Redundant, TCP/IP-TCP Redundant) it is possible to configure I/O tags with addresses **IN_A** and **IN_B** to distinguish inputs from primary/secondary line
- on system-redundant lines (Serial System&Line Redundant, SerialOverUDP System&Line Redundant) it is possible to configure I/O tags with addresses **IN_A**, **IN_B**, **IN_C**, **IN_D** to distinguish inputs from primary/secondary line of system A/B

Output I/O tags:

- The output I/O tag has address **OUT**.
  Note: on redundant lines values written to an I/O tag with address **OUT** are written to both lines (primary/secondary), eventually to both systems (A /B).
- on line-redundant lines (Serial Line Redundant, SerialOverUDP Line Redundant, TCP/IP-TCP Redundant) it is possible to configure I/O tags with addresses **OUT_A** and **OUT_B** to write data to primary/secondary line only
- on system-redundant lines (Serial System&Line Redundant, SerialOverUDP System&Line Redundant) it is possible to configure I/O tags with addresses **OUT_A**, **OUT_B**, **OUT_C**, **OUT_D** to write data to primary/secondary line of system A/B only

Note: it is possible to configure and use an I/O tag with the address **IN** and I/O tags **IN_A**, **IN_B**, **IN_C**, **IN_D** on a single station at the same time.
In a similar way, it is possible to configure and use an I/O tag with the address **OUT** and I/O tags **OUT_A**, **OUT_B**, **OUT_C**, **OUT_D** on a single station at the same time.

Literature

Blog

You can read a blog about Generic User Protocol.

Changes and modifications

-

Document revisions

- Ver. 1.0 - October 15, 2015 - creating document

Related pages:

Communication protocols