C3 BRIDGE INTERFACE

Application Protocol Specification

Version 1, Revision 2
30 May, 2021

Dmitry Lavygin
S.P. Kapitsa Research Institute of Technology
Ulyanovsk State University

Ulyanovsk • Russian Federation

COPYRIGHT

DOCUMENT HISTORY

Revision	Notes	Date
1	Initial release	2020-08-04
2	**Fixed** • Typo in the table in the section 1.1 (the port number should be 6999) • Typo in the description of the section 2 • Bit counting in the section 3.8.2 • Interpreter type in the section 3.6.1 must be INT16 **Added** • 1.5. SOFTWARE VERSIONING • 2.7. PROXY LISTENING ADDRESS REQUEST • 2.8. PROXY LISTENING PORT REQUEST • 2.9. PROXY ENABLED REQUEST • 3.8.5. MESSAGE #17. PERFORM PROXY BENCHMARK • 3.9. MESSAGES FOR FILE OPERATIONS **Changed** • 1.4. DATA TYPES • 2.3. PROXY VERSION REQUEST • 3.4. MESSAGE TYPES • 3.5.1. MESSAGE #0. READ VARIABLE (ASCII) • 3.5.5. MESSAGE #4. READ VARIABLE • 3.5.7. MESSAGE #6. READ MULTIPLE VARIABLES • 3.5.8. MESSAGE #7. WRITE MULTIPLE VARIABLES • 3.8.1. MESSAGE #13. GET PROXY INFORMATION	2021-05-30

TABLE OF CONTENTS

1. OVERVIEW 9

1.1. TYPES AND PURPOSE OF PROTOCOLS 9

1.2. BYTE ORDER 10

1.3. HEXADECIMAL NUMBERS 10

1.4. DATA TYPES 10

1.5. SOFTWARE VERSIONING 11

2. DISCOVERY PROTOCOL 12

2.1. PRESENCE REQUEST 13

2.2. PROXY TYPE REQUEST 14

2.3. PROXY VERSION REQUEST 15

2.4. PROXY FEATURES REQUEST 16

2.5. COMPUTER NAME REQUEST 19

2.6. DATE AND TIME REQUEST 20

2.7. PROXY LISTENING ADDRESS REQUEST 21

2.8. PROXY LISTENING PORT REQUEST 22

2.9. PROXY ENABLED REQUEST 23

3. PRIMARY PROTOCOL 24

3.1. MESSAGE HEADER 25

3.2. RESPONSE FOOTER 26

3.3. ERROR CODES 27

3.4. MESSAGE TYPES 29

3.5. MESSAGES FOR VARIABLE HANDLING 32

3.5.1. MESSAGE #0. READ VARIABLE (ASCII) 32

3.5.2. MESSAGE #1. WRITE VARIABLE (ASCII) 35

3.5.3. MESSAGE #2. READ ARRAY (ASCII) 38

3.5.4. MESSAGE #3. WRITE ARRAY (ASCII) 38

3.5.5. MESSAGE #4. READ VARIABLE 38
3.5.6. MESSAGE #5. WRITE VARIABLE........ 42
3.5.7. MESSAGE #6. READ MULTIPLE VARIABLES........ 45
3.5.8. MESSAGE #7. WRITE MULTIPLE VARIABLES 49
3.6. MESSAGES FOR KRL PROGRAM HANDLING........ 54
3.6.1. MESSAGE #10. PROGRAM CONTROL (SUBTYPE I) 54
3.6.2. MESSAGE #10. PROGRAM CONTROL (SUBTYPE II) 56
3.7. MESSAGES FOR MANUAL ROBOT CONTROL 59
3.7.1. MESSAGE #11. MOTION CONTROL 59
3.7.2. MESSAGE #12. KCP KEY EMULATION 62
3.8. SERVICE MESSAGES........ 64
3.8.1. MESSAGE #13. GET PROXY INFORMATION 64
3.8.2. MESSAGE #14. GET PROXY FEATURES........ 67
3.8.3. MESSAGE #15. GET PROXY INFORMATION (EXTENDED) 71
3.8.4. MESSAGE #16. GET CROSS3 INFORMATION........ 71
3.8.5. MESSAGE #17. PERFORM PROXY BENCHMARK 72
3.9. MESSAGES FOR FILE OPERATIONS 75
3.9.1. CONSTANT VALUES........ 75
3.9.2. MESSAGE #20. SET FILE ATTRIBUTES 79
3.9.3. MESSAGE #21. LIST DIRECTORY CONTENTS 81
3.9.4. MESSAGE #22. CREATE NEW FILE 83
3.9.5. MESSAGE #23. DELETE FILE........ 85
3.9.6. MESSAGE #24. COPY FILE........ 87
3.9.7. MESSAGE #25. MOVE FILE 89
3.9.8. MESSAGE #26. GET FILE PROPERTIES........ 91
3.9.9. MESSAGE #27. GET FILE FULL PATH 94
3.9.10. MESSAGE #28. GET KRC PATH........ 95
3.9.11. MESSAGE #29. WRITE FILE CONTENT (BEGINNING)........ 97
3.9.12. MESSAGE #29. WRITE FILE CONTENT (DATA CHUNK) 99

3.9.13. MESSAGE #29. WRITE FILE CONTENT (CHECKSUM) 101
3.9.14. MESSAGE #29. WRITE FILE CONTENT (FINAL) 101
3.9.15. MESSAGE #30. READ FILE CONTENT (BEGINNING).......................... 103
3.9.16. MESSAGE #30. READ FILE CONTENT (DATA CHUNK)....................... 105
3.9.17. MESSAGE #30. READ FILE CONTENT (CHECKSUM) 107
3.9.18. MESSAGE #30. READ FILE CONTENT (FINAL) 107
3.10. MESSAGES FOR CROSSCOMMEXE COMPATIBILITY 109
3.10.1. MESSAGE #64. CONFIRM ALL.. 109

TERMS USED

Term	Description
IP	Internet Protocol The Internet Protocol (IP) is the principal communications protocol in the Internet protocol suite. It is responsible for addressing host interfaces, encapsulating data into datagrams (including fragmentation and reassembly) and routing datagrams from a source host interface to a destination host interface across one or more IP networks.
TCP/IP	Transmission Control Protocol The Transmission Control Protocol provides a communication service at an intermediate level between an application program and the Internet Protocol. It provides host-to-host connectivity at the transport layer of the Internet model.
UDP/IP	User Datagram Protocol UDP is a simple message-oriented transport layer protocol that is documented in RFC 768. Although UDP provides integrity verification (via checksum) of the header and payload, it provides no guarantees to the upper layer protocol for message delivery and the UDP layer retains no state of UDP messages once sent.
KRC	KUKA Robot Controller.
KRL	KUKA Robot Language KUKA Robot programming language.

KUKA Cross 3	Internal mechanism of interprocess communication in the KUKA robot control system.
KukavarProxy	KukavarProxy is a TCP/IP server that allows KRL variables to be read and written over a network connection.

1. OVERVIEW

This document describes the protocols used by the C3 Bridge Interface Server. The C3 Bridge Interface Server is a lightweight network application that allows remote clients to execute requests to KUKA Cross 3 subsystem and return responses. The application provides advanced functionality and high performance.

1.1. TYPES AND PURPOSE OF PROTOCOLS

The C3 Bridge Interface Server can use two network protocols.
The first protocol, called the Discovery Protocol, is based on UDP and can be used to detect a remote server and find out its capabilities. The Discovery Protocol can operate in legacy or standard mode, or both. The legacy mode is implemented for compatibility with KukavarProxy features.
The second protocol is the primary one. It is based on TCP and is designed for remote interaction with the KUKA robot control system.
The table below shows a summary of the protocols and network ports on which they operate by default.

Protocol	Based on	Listening port	Port to answer	Support in KukavarProxy
Discovery (legacy)	UDP	6999	7000	Yes
Discovery (standard)	UDP	7000	source port of peer	
Primary	TCP	7000	---	Yes, limited

1.2. BYTE ORDER

All multibyte fields in protocol messages are composed using the network byte order (or big-endian, most significant byte is transmitted first). Although this is in contradiction with the Intel IA-32 platform byte order (little-endian), the network byte order was chosen to provide compatibility with the KukavarProxy protocol. **The exceptions to this order are characters and strings in UTF16 format. The system byte order (little-endian) is used for them.**

1.3. HEXADECIMAL NUMBERS

Base 16 (hexadecimal) numbers are represented by a string of hexadecimal digits followed by the character "h" (for example, 0D0Ah). A hexadecimal digit is a character from the following set: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F.

1.4. DATA TYPES

The following table gives information about the types of data used:

Name	Description	Size (bytes)	Range
UINT8	Unsigned Integer, 8–bit	1	0..255
INT8	Signed Integer, 8–bit	1	-128..127
UINT16	Unsigned Integer, 16–bit	2	0..65535
INT16	Signed Integer, 16–bit	2	-32768..32767
UINT32	Unsigned Integer, 32–bit	4	0..4294967295
INT32	Signed Integer, 32–bit	4	-2147483648.. 2147483647
BOOL	Boolean Type	1	0..1*

CHAR	ASCII / ISO/IEC 8859−1 Character	1	
STRING	ASCII / ISO/IEC 8859−1 String	*variable*	
WCHAR	Unicode Character (encoded in UTF−16LE)	2	
WSTRING	Unicode String (encoded in UTF−16LE)	*variable (even)*	
BINARY	Binary Data (array of UINT8)	*variable*	

NOTES

- The Boolean value is encoded with one byte. This means that the actual value of this field may be between **0** and **255**. The recipient must treat all non-zero values as TRUE and **0** as FALSE. The sender must encode the TRUE value with **1** and the FALSE value with **0**.

1.5. SOFTWARE VERSIONING

All products in the C3 Bridge family use versions consisting of three or, in some cases, four numbers. The first two numbers represent the major and minor components of the version. The third number denotes the type of build shown in the table below. The fourth number can only be used for internal builds.

Version Number	**Build Type**
0	Open Source
1	Proprietary
2	Freeware
3	Internal Build

2. DISCOVERY PROTOCOL

The Discovery Protocol uses the UDP datagrams that contain text messages of a predetermined length. In the legacy mode, the sender makes requests to the server on port 6999 and the server responds to port 7000 of the sender. In standard mode, the sender makes requests to the server on port 7000, and the server responds to the sender's address and port, allowing the sender to use any port to receive responses.

2.1. PRESENCE REQUEST

Minimum supported version: 1.0.0 (Open Source).
Support in KukavarProxy: Yes.

PURPOSE

Determination of the control system address and readiness of the control system to process requests of the primary protocol (indirectly).

REQUEST

Offset (bytes)	Size (bytes)	Type	Value
0	12	STRING	`WHEREAREYOU?`

RESPONSE

Offset (bytes)	Size (bytes)	Type	Value
0	*variable*	STRING	KUKA|*<model name>*|*<serial #>*

<model name> is the value of **$MODEL_NAME[]** KRL variable.
<serial #> is the value of **$KR_SERIALNO** KRL variable.

In case of an error when accessing the KUKA Cross 3 subsystem, the fields *<model name>* and *<serial #>* may be empty. In this case, the response has the following form:

Offset (bytes)	Size (bytes)	Type	Value
0	6	STRING	KUKA||

2.2. PROXY TYPE REQUEST

Minimum supported version: 1.0.0 (Open Source).
Support in KukavarProxy: No.

PURPOSE

Determining the type of proxy server. The C3 Bridge Interface responds to this request and KukavarProxy does not.

REQUEST

Offset (bytes)	Size (bytes)	Type	Value
0	11	STRING	`@PROXY_TYPE`

RESPONSE

Offset (bytes)	Size (bytes)	Type	Value
0	19	STRING	`C3 BRIDGE INTERFACE`

2.3. PROXY VERSION REQUEST

Minimum supported version: 1.0.0 (Open Source).
Support in KukavarProxy: No.

PURPOSE

Obtaining information about the application version and license type.

REQUEST

Offset (bytes)	Size (bytes)	Type	Value
0	14	STRING	`@PROXY_VERSION`

RESPONSE

Offset (bytes)	Size (bytes)	Type	Value
0	*variable*	STRING	`<major>.<minor> <type>`

<*major*> is the major number of the software version.
<*minor*> is the minor number of the software version.
<*type*> is the type of the software edition (look at section 1.5. SOFTWARE VERSIONING), it can be `(OPEN SOURCE)`, `(PROPRIETARY)`, `(FREEWARE)`, or `(INTERNAL BUILD)`.

SAMPLE RESPONSE

Offset (bytes)	Size (bytes)	Type	Value
0	17	STRING	`1.0 (OPEN SOURCE)`

2.4. PROXY FEATURES REQUEST

Minimum supported version: 1.0.0 (Open Source).
Support in KukavarProxy: No.

PURPOSE

Determining the list of supported messages for the primary protocol.

REQUEST

Offset (bytes)	Size (bytes)	Type	Value
0	15	STRING	`@PROXY_FEATURES`

The response table is located on the next page.

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Value	Meaning
REQUIRED	-1	1	CHAR	1	Message #0 is available
				0	Message #0 is NOT available
	-2	1	CHAR	1	Message #1 is available
				0	Message #1 is NOT available
	...				
	-8	1	CHAR	1	Message #7 is available
				0	Message #7 is NOT available
OPTIONAL	-9	1	CHAR	1	Message #8 is available
				0	Message #8 is NOT available
	...				
	-256	1	CHAR	1	Message #255 is available
				0	Message #255 is NOT available

Negative offset means bytes counted from the end of the received datagram. For example, -1 means the last byte, -2 means the penultimate byte, etc.

SAMPLE RESPONSE

Offset (bytes)	Size (bytes)	Type	Value	Meaning
-1	1	CHAR	1	Message #0 is available
-2	1	CHAR	1	Message #1 is available
-3	1	CHAR	0	Message #2 is NOT available
-4	1	CHAR	0	Message #3 is NOT available
-5	1	CHAR	1	Message #4 is available
-6	1	CHAR	1	Message #5 is available
-7	1	CHAR	1	Message #6 is available
-8	1	CHAR	1	Message #7 is available
-9	1	CHAR	0	Message #8 is NOT available
-10	1	CHAR	0	Message #9 is NOT available
-11	1	CHAR	1	Message #10 is available
-12	1	CHAR	1	Message #11 is available
-13	1	CHAR	1	Message #12 is available
-14	1	CHAR	1	Message #13 is available
-15	1	CHAR	1	Message #14 is available
-16	1	CHAR	0	Message #15 is NOT available

In this example, the string representation of the received data is as follows: `0111110011110011`.

2.5. COMPUTER NAME REQUEST

Minimum supported version: 1.0.0 (Open Source).
Support in KukavarProxy: No.

PURPOSE

Obtaining the computer name on which the robot control system is located.

REQUEST

Offset (bytes)	Size (bytes)	Type	Value
0	15	STRING	`@PROXY_HOSTNAME`

RESPONSE

Offset (bytes)	Size (bytes)	Type	Value
0	*variable*	STRING	`<KRC hostname>`

SAMPLE RESPONSE

Offset (bytes)	Size (bytes)	Type	Value
0	9	STRING	`C010-07VM`

2.6. DATE AND TIME REQUEST

Minimum supported version: 1.0.0 (Open Source).
Support in KukavarProxy: No.

PURPOSE

Obtaining the date and time set on the robot control system in ISO 8601 format.

REQUEST

Offset (bytes)	Size (bytes)	Type	Value
0	11	STRING	`@PROXY_TIME`

RESPONSE

Offset (bytes)	Size (bytes)	Type	Value
0	20	STRING	`YYYY-MM-DDThh:mm:ssZ`

[YYYY] indicates a four-digit year, 1601 through 9999.
[MM] indicates a two-digit month of the year, 01 through 12.
[DD] indicates a two-digit day of that month, 01 through 31.
[T] is just ANSI character `T`, which is used to separate the date and time.
[hh] refers to a zero-padded hour between 00 and 23.
[mm] refers to a zero-padded minute between 00 and 59.
[ss] refers to a zero-padded second between 00 and 59.
[Z] means that the Coordinated Universal Time (UTC) is used.

SAMPLE RESPONSE

Offset (bytes)	Size (bytes)	Type	Value
0	20	STRING	`2020-08-04T06:46:10Z`

2.7. PROXY LISTENING ADDRESS REQUEST

Minimum supported version: 1.2.0 (Open Source).
Support in KukavarProxy: No.

PURPOSE

Obtaining the TCP/IP address where the primary C3 Bridge Interface protocol is listening. The address 0.0.0.0 means that listening is performed on all available network interfaces.

REQUEST

Offset (bytes)	Size (bytes)	Type	Value
0	15	STRING	`@PROXY_ADDRESS`

RESPONSE

Offset (bytes)	Size (bytes)	Type	Value
0	*variable*	STRING	`<TCP/IP address>`

SAMPLE RESPONSE

Offset (bytes)	Size (bytes)	Type	Value
0	7	STRING	`0.0.0.0`

2.8. PROXY LISTENING PORT REQUEST

Minimum supported version: 1.2.0 (Open Source).
Support in KukavarProxy: No.

PURPOSE

Obtaining the TCP/IP port where the primary C3 Bridge Interface protocol is listening.

REQUEST

Offset (bytes)	Size (bytes)	Type	Value
0	11	STRING	`@PROXY_PORT`

RESPONSE

Offset (bytes)	Size (bytes)	Type	Value
0	*variable*	STRING	`<TCP/IP port>`

SAMPLE RESPONSE

Offset (bytes)	Size (bytes)	Type	Value
0	4	STRING	`7000`

2.9. PROXY ENABLED REQUEST

Minimum supported version: 1.2.0 (Open Source).
Support in KukavarProxy: No.

PURPOSE

Getting information about the status of the TCP/IP server of the primary protocol.

REQUEST

Offset (bytes)	Size (bytes)	Type	Value
0	14	STRING	`@PROXY_ENABLED`

RESPONSE

Offset (bytes)	Size (bytes)	Type	Value
0	*variable*	STRING	`<Boolean Value>`

SAMPLE RESPONSE

Offset (bytes)	Size (bytes)	Type	Value
0	4	STRING	`TRUE`

3. PRIMARY PROTOCOL

The primary protocol uses long-term TCP/IP sessions. Data exchange takes place using binary messages of variable length. Transmitted messages are divided into requests and responses. The requests contain only the header and payload. The responses contain the header, payload, and error code at the end of the message. The server has the right not to reply to unknown or incorrect requests.

3.1. MESSAGE HEADER

Each message begins with a header, the structure of which is shown in the table below.

Offset (bytes)	Size (bytes)	Type	Meaning
0	2	UINT16	**Tag ID** This field specifies the message identifier. The response from the server will contain the same identifier as the request. This identifier does not define the type of request and can accept any values in the range from 0 to 65 535.
2	2	UINT16	**Message Length** The full length of the message, excluding the Tag ID and Message Length fields.
4	1	UINT8	**Message Type** An important field that defines the type of message. The message type indicates the number of the function that will be or has been executed by the C3 Bridge Interface.

3.2. RESPONSE FOOTER

Each response message ends with a footer, the structure of which is shown in the table below.

Offset (bytes)	Size (bytes)	Type	Meaning
0	2	UINT16	**Error Code** The error codes are listed in the next section.
2	1	BOOL	**Success Flag** TRUE in case of a successful response, FALSE in case of error.

3.3. ERROR CODES

The full list of error codes can be found in the file *include/c3bi.h*.

Code	Name	Description
0	ErrorGeneral	Unspecified error. In some cases it may be the result of an **E_FAIL** error from the Windows COM subsystem.
1	ErrorSuccess	Not an error. The operation was successful.
2	ErrorAccess	General access denied error. COM equivalent: **E_ACCESSDENIED**.
3	ErrorArgument	One or more arguments are not valid. COM equivalent: **E_INVALIDARG**.
4	ErrorMemory	Failed to allocate necessary memory. COM equivalent: **E_OUTOFMEMORY**.
5	ErrorPointer	**NULL** was passed incorrectly for a pointer value. COM equivalent: **E_POINTER**.
6	ErrorUnexpected	Unexpected failure. COM equivalent: **E_UNEXPECTED**.
7	ErrorNotImplemented	The requested function has not been implemented. In some cases it may be the result of an **E_NOTIMPL** error from the Windows COM subsystem.

8	ErrorNoInterface	No such interface supported. COM equivalent: **E_NOINTERFACE**.
9	ErrorProtocol	Error in message content, incorrect number of fields, or their values.
10	ErrorLongAnswer	The response message is too big. The data cannot fit into a single message.

3.4. MESSAGE TYPES

The full list of message types can be found in the file *include/c3bi.h*.

Type	Name
0	CommandReadVariableAscii
1	CommandWriteVariableAscii
2	CommandReadArrayAscii
3	CommandWriteArrayAscii
4	CommandReadVariable
5	CommandWriteVariable
6	CommandReadMultiple
7	CommandWriteMultiple
8	Reserved
9	
10	CommandProgramControl
11	CommandMotion
12	CommandKcpAction
13	CommandProxyInfo
14	CommandProxyFeatures
15	CommandProxyInfoEx
16	CommandProxyCrossInfo
17	CommandProxyBenchmark
18	Reserved
19	
20	CommandFileSetAttribute

21	CommandFileNameList
22	CommandFileCreate
23	CommandFileDelete
24	CommandFileCopy
25	CommandFileMove
26	CommandFileGetProperties
27	CommandFileGetFullName
28	CommandFileGetKrcName
29	CommandFileWriteContent
30	CommandFileReadContent
31	Reserved
…	
49	
50	CommandCrossSetInfoOn
51	CommandCrossSetInfoOff
52	CommandCrossGetRobotDirectory
53	CommandCrossDownloadDiskToRobot
54	CommandCrossDownloadMemToRobot
55	CommandCrossUploadFromRobotToDisk
56	CommandCrossUploadFromRobotToMem
57	CommandCrossDeleteRobotProgram
58	CommandCrossRobotLevelStop
59	CommandCrossControlLevelStop
60	CommandCrossRunControlLevel

61	CommandCrossSelectModul
62	CommandCrossCancelModul
63	CommandCrossConfirmAll
64	CommandCrossKrcOk
65	CommandCrossIoRestart
66	CommandCrossReserved
67	
68	
69	
70	Reserved
…	
128	
129	Free Range
…	
254	
255	CommandExtended

3.5. MESSAGES FOR VARIABLE HANDLING

3.5.1. MESSAGE #0. READ VARIABLE (ASCII)

Minimum supported version: 1.0.0 (Open Source).
Support in KukavarProxy: Yes.

PURPOSE

Retrieving the value of KRL variable or internal variable (ASCII version).

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 0
PAYLOAD	5	2	UINT16	**LVN** Length of Variable Name
	7	LVN	STRING	Variable Name

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 6 + LVV
	4	1	UINT8	Message Type Value: 0

PAYLOAD	5	2	UINT16	**LVV** Length of Variable Value
	7	LVV	STRING	Variable Value
FOOTER	7 + LVV	2	UINT16	Error Code
	9 + LVV	1	BOOL	Success Flag

SAMPLE REQUEST

	Offset (bytes)	Size (bytes)	Type	Value
HEADER	0	2	UINT16	256
	2	2	UINT16	14
	4	1	UINT8	0
PAYLOAD	5	2	UINT16	11
	7	11	STRING	$ACCU_STATE

SAMPLE RESPONSE

	Offset (bytes)	Size (bytes)	Type	Value
HEADER	0	2	UINT16	256
	2	2	UINT16	16
	4	1	UINT8	0

PAYLOAD	5	2	UINT16	10
	7	10	STRING	#CHARGE_OK
FOOTER	17	2	UINT16	1 (ErrorSuccess)
	19	1	BOOL	TRUE

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess

INTERNAL VARIABLES

C3 Bridge Interface contains several internal variables whose values can be obtained with the Read Variable message. Access to the internal variable is possible provided that there is no variable with the same name in the KRL system.

Variable Name	Variable Value
PING	PONG
@PROXY_TYPE	C3 BRIDGE INTERFACE
@PROXY_VERSION	Look at section 2.3. PROXY VERSION REQUEST
@PROXY_FEATURES	Look at section 2.4. PROXY FEATURES REQUEST
@PROXY_HOSTNAME	Look at section 2.5. COMPUTER NAME REQUEST
@PROXY_TIME	Look at section 2.6. DATE AND TIME REQUEST

@PROXY_ADDRESS	Look at section 2.7. PROXY LISTENING ADDRESS REQUEST
@PROXY_PORT	Look at section 2.8. PROXY LISTENING PORT REQUEST
@PROXY_ENABLED	Look at section 2.9. PROXY ENABLED REQUEST

NOTES

* The PING variable is also supported by KukavarProxy.

3.5.2. MESSAGE #1. WRITE VARIABLE (ASCII)

Minimum supported version: 1.0.0 (Open Source).
Support in KukavarProxy: Yes.

PURPOSE

Writing the new value of the KRL variable (ASCII version).

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 5 + LVN + LVV
	4	1	UINT8	Message Type Value: 1
PAYLOAD	5	2	UINT16	**LVN** Length of Variable Name
	7	LVN	STRING	Variable Name

	7 + LVN	2	UINT16	**LVV** Length of Variable Value
	9 + LVN	LVV	STRING	Variable Value

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 6 + LVV
	4	1	UINT8	Message Type Value: 1
PAYLOAD	5	2	UINT16	**LVV** Length of Variable Value
	7	LVV	STRING	Variable Value
FOOTER	7 + LVV	2	UINT16	Error Code
	9 + LVV	1	BOOL	Success Flag

SAMPLE REQUEST

	Offset (bytes)	Size (bytes)	Type	Value
HEADER	0	2	UINT16	256
	2	2	UINT16	17
	4	1	UINT8	1

PAYLOAD	5	2	UINT16	8
	7	8	STRING	$VEL_ACT
	15	2	UINT16	4
	17	4	STRING	10.2

SAMPLE RESPONSE

	Offset (bytes)	Size (bytes)	Type	Value
HEADER	0	2	UINT16	256
	2	2	UINT16	10
	4	1	UINT8	1
PAYLOAD	5	2	UINT16	4
	7	4	STRING	10.2
FOOTER	11	2	UINT16	1 (ErrorSuccess)
	13	1	BOOL	TRUE

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess

3.5.3. MESSAGE #2. READ ARRAY (ASCII)

Minimum supported version: None.
Support in KukavarProxy: Yes.

PURPOSE

(*Translated from KukavarProxy source code*) Reading and formatting an array variable for the PLC.

3.5.4. MESSAGE #3. WRITE ARRAY (ASCII)

Minimum supported version: None.
Support in KukavarProxy: Yes.

PURPOSE

(*Translated from KukavarProxy source code*) Writing an array variable to the PLC.

3.5.5. MESSAGE #4. READ VARIABLE

Minimum supported version: 1.0.0 (Open Source).
Support in KukavarProxy: No.

PURPOSE

Retrieving the value of KRL variable or internal variable.

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 3 + LVN * 2

	4	1	UINT8	Message Type Value: 4
PAYLOAD	5	2	UINT16	**LVN** Length of Variable Name (in characters)
	7	LVN * 2	WSTRING	Variable Name

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 6 + LVV * 2
	4	1	UINT8	Message Type Value: 4
PAYLOAD	5	2	UINT16	**LVV** Length of Variable Value (in characters)
	7	LVV * 2	WSTRING	Variable Value
FOOTER	7 + LVV * 2	2	UINT16	Error Code
	9 + LVV * 2	1	BOOL	Success Flag

SAMPLE REQUEST

	Offset (bytes)	Size (bytes)	Type	Value
HEADER	0	2	UINT16	512
	2	2	UINT16	21
	4	1	UINT8	4
PAYLOAD	5	2	UINT16	9
	7	18	WSTRING	$ACT_BASE

SAMPLE RESPONSE

	Offset (bytes)	Size (bytes)	Type	Value
HEADER	0	2	UINT16	512
	2	2	UINT16	8
	4	1	UINT8	4
PAYLOAD	5	2	UINT16	1
	7	2	WSTRING	1
FOOTER	9	2	UINT16	1 (ErrorSuccess)
	11	1	BOOL	TRUE

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess
9	ErrorProtocol

INTERNAL VARIABLES

Look at section 3.5.1. MESSAGE #0. READ VARIABLE (ASCII).

3.5.6. MESSAGE #5. WRITE VARIABLE

Minimum supported version: 1.0.0 (Open Source).
Support in KukavarProxy: No.

PURPOSE

Writing the new value of the KRL variable.

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 5 + LVN * 2 + LVV * 2
	4	1	UINT8	Message Type Value: 5
PAYLOAD	5	2	UINT16	**LVN** Length of Variable Name (in characters)
	7	LVN * 2	WSTRING	Variable Name
	7 + LVN * 2	2	UINT16	**LVV** Length of Variable Value (in characters)
	9 + LVN * 2	LVV * 2	WSTRING	Variable Value

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 6 + LVV * 2
	4	1	UINT8	Message Type Value: 5
PAYLOAD	5	2	UINT16	**LVV** Length of Variable Value (in characters)
	7	LVV * 2	WSTRING	Variable Value
FOOTER	7 + LVV * 2	2	UINT16	Error Code
	9 + LVV * 2	1	BOOL	Success Flag

SAMPLE REQUEST

	Offset (bytes)	Size (bytes)	Type	Value
HEADER	0	2	UINT16	256
	2	2	UINT16	23
	4	1	UINT8	5
PAYLOAD	5	2	UINT16	8
	7	16	WSTRING	$VEL_ACT
	15	2	UINT16	1

	17	2	WSTRING	5

SAMPLE RESPONSE

	Offset (bytes)	Size (bytes)	Type	Value
HEADER	0	2	UINT16	256
	2	2	UINT16	8
	4	1	UINT8	5
PAYLOAD	5	2	UINT16	1
	7	2	WSTRING	5
FOOTER	9	2	UINT16	1 (ErrorSuccess)
	11	1	BOOL	TRUE

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess
9	ErrorProtocol

3.5.7. MESSAGE #6. READ MULTIPLE VARIABLES

Minimum supported version: 1.0.0 (Open Source).
Support in KukavarProxy: No.

WARNING

Do not use this function with the C3 Bridge Interface server versions lower than 1.2.0 (Open Source).
Earlier implementations did not work correctly.

PURPOSE

Retrieving the values of several KRL variables or internal variables.

REQUEST

		Offset (bytes)	Size (bytes)	Type	Meaning
HEADER		0	2	UINT16	Tag ID
		2	2	UINT16	Message Length
		4	1	UINT8	Message Type Value: 6
PAYLOAD		5	1	UINT8	Number of Variables Value: 0-255
	OPTIONAL	6	2	UINT16	**LVN1** Length of Variable 1 (in characters)
		8	LVN1 * 2	WSTRING	Variable 1

	OPTIONAL	*variable*	2	UINT16	**LVN2** Length of Variable 2 (in characters)
		variable	LVN2 * 2	WSTRING	Variable 2
		…			
		variable	2	UINT16	**LVNL** Length of Last Variable (in characters)
		variable	LVNL * 2	WSTRING	Last Variable

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 6
PAYLOAD	5	1	UINT8	Number of Variables
	6	1	UINT8	**CODE1** Error code of Variable 1 request (look at section 3.3. ERROR CODES)
	7	2	UINT16	**LVV1** Length of Variable 1 Value (in characters)

	9	LVV1 * 2	WSTRING	Variable 1 Value
	…			
	variable	1	UINT8	**CODEL** Error code of Last Variable request (look at section 3.3. ERROR CODES)
	variable	2	UINT16	**LVVL** Length of Last Variable Value (in characters)
	variable	LVVL * 2	WSTRING	Last Variable Value
FOOTER	*variable*	2	UINT16	Error Code
	variable	1	BOOL	Success Flag

SAMPLE REQUEST

	Offset (bytes)	**Size (bytes)**	**Type**	**Value**
HEADER	0	2	UINT16	`1024`
	2	2	UINT16	`36`
	4	1	UINT8	`6`
PAYLOAD	5	1	UINT8	`2`
	6	2	UINT16	`4`
	8	8	WSTRING	`PING`
	16	2	UINT16	`11`
	18	22	WSTRING	`@PROXY_PORT`

SAMPLE RESPONSE

	Offset (bytes)	Size (bytes)	Type	Value
HEADER	0	2	UINT16	1024
	2	2	UINT16	27
	4	1	UINT8	6
PAYLOAD	5	1	UINT8	2
	6	1	UINT8	1 (ErrorSuccess)
	7	2	UINT16	4
	9	8	WSTRING	PONG
	17	1	UINT8	1 (ErrorSuccess)
	18	2	UINT16	4
	20	8	WSTRING	7000
FOOTER	28	2	UINT16	1 (ErrorSuccess)
	30	1	BOOL	TRUE

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess
9	ErrorProtocol
10	ErrorLongAnswer

INTERNAL VARIABLES

Look at section 3.5.1. MESSAGE #0. READ VARIABLE (ASCII).

3.5.8. MESSAGE #7. WRITE MULTIPLE VARIABLES

Minimum supported version: 1.0.0 (Open Source).
Support in KukavarProxy: No.

WARNING

Do not use this function with the C3 Bridge Interface server versions lower than 1.2 (Open Source).
Earlier implementations did not work correctly.

PURPOSE

Writing new values of several KRL variables.

REQUEST

		Offset (bytes)	Size (bytes)	Type	Meaning
HEADER		0	2	UINT16	Tag ID
		2	2	UINT16	Message Length
		4	1	UINT8	Message Type Value: 7
PAYLOAD		5	1	UINT8	Number of Variables Value: 0-255
	OPTIONAL	6	2	UINT16	**LVN1** Length of Variable 1 (in characters)
		8	LVN1 * 2	WSTRING	Variable 1
		8 + LVN1 * 2	2	UINT16	**LVV1** Length of Variable 1 Value (in characters)

		10 + LVV1 * 2	LVV1 * 2	WSTRING	Variable 1 Value
		…			
		variable	2	UINT16	**LVNL** Length of Last Variable (in characters)
		variable	LVNL * 2	WSTRING	Last Variable
		variable	2	UINT16	**LVVL** Length of Last Variable Value (in characters)
		variable	LVVL * 2	WSTRING	Last Variable Value

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 7
PAYLOAD	5	1	UINT8	Number of Variables
	6	1	UINT8	**CODE1** Error code of Variable 1 request (look at section 3.3. ERROR CODES)
	7	2	UINT16	**LVV1** Length of Variable 1 Value (in characters)
	9	LVV1 * 2	WSTRING	Variable 1 Value
	…			
	variable	1	UINT8	**CODEL** Error code of Last Variable request (look at section 3.3. ERROR CODES)
	variable	2	UINT16	**LVVL** Length of Last Variable Value (in characters)
	variable	LVVL * 2	WSTRING	Last Variable Value

FOOTER	*variable*	2	UINT16	Error Code
	variable	1	BOOL	Success Flag

SAMPLE REQUEST

	Offset (bytes)	Size (bytes)	Type	Value
HEADER	0	2	UINT16	1024
	2	2	UINT16	24
	4	1	UINT8	7
PAYLOAD	5	1	UINT8	1
	6	2	UINT16	8
	8	16	WSTRING	$VEL_ACT
	24	2	UINT16	1
	26	2	WSTRING	5

SAMPLE RESPONSE

	Offset (bytes)	Size (bytes)	Type	Value
HEADER	0	2	UINT16	1024
	2	2	UINT16	10
	4	1	UINT8	7
PAYLOAD	5	1	UINT8	1
	6	1	UINT8	1 (ErrorSuccess)
	7	2	UINT16	1

	9	2	WSTRING	5
FOOTER	11	2	UINT16	1 (ErrorSuccess)
	13	1	BOOL	TRUE

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess
9	ErrorProtocol
10	ErrorLongAnswer

3.6. MESSAGES FOR KRL PROGRAM HANDLING

3.6.1. MESSAGE #10. PROGRAM CONTROL (SUBTYPE I)

Minimum supported version: 1.0.0 (Open Source).
Support in KukavarProxy: No.

PURPOSE

Reset, start, stop or cancel the KRL program.

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 4
	4	1	UINT8	Message Type Value: 10
PAYLOAD	5	1	UINT8	Command code: 1 – Reset 2 – Start 3 – Stop 4 – Cancel
	6	2	INT16	Interpreter Type: 0 – Sumbit Interpreter 1 – Robot Interpreter

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 5
	4	1	UINT8	Message Type Value: 10
PAYLOAD	5	1	UINT8	Command code
FOOTER	6	2	UINT16	Error Code
	8	1	BOOL	Success Flag

SAMPLE REQUEST

	Offset (bytes)	Size (bytes)	Type	Value
HEADER	0	2	UINT16	652
	2	2	UINT16	4
	4	1	UINT8	10
PAYLOAD	5	1	UINT8	1 (Reset)
	6	2	UINT16	0 (Submit Interpreter)

SAMPLE RESPONSE

	Offset (bytes)	Size (bytes)	Type	Value
HEADER	0	2	UINT16	652
	2	2	UINT16	5
	4	1	UINT8	10
PAYLOAD	5	1	UINT8	1 (Reset)
FOOTER	6	2	UINT16	1 (ErrorSuccess)
	8	1	BOOL	TRUE

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess
7	ErrorNotImplemented
9	ErrorProtocol

3.6.2. MESSAGE #10. PROGRAM CONTROL (SUBTYPE II)

Minimum supported version: 1.0.0 (Open Source).
Support in KukavarProxy: No.

PURPOSE

Select or run the KRL program.

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 9 + (LN + LP) * 2
	4	1	UINT8	Message Type Value: 10
PAYLOAD	5	1	UINT8	Command code: 5 – Select 6 – Run
	6	2	INT16	Interpreter Type (NOT USED)
	8	2	UINT16	**LN** Length of Name (in characters)
	10	LN * 2	WSTRING	Name
	10 + LN * 2	2	UINT16	**LP** Length of Parameters (in characters)
	12 + LN * 2	LP * 2	WSTRING	Parameters
	12 + LN * 2 + LP * 2	1	BOOL	Force Select/Run

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 5
	4	1	UINT8	Message Type Value: 10
PAYLOAD	5	1	UINT8	Command code
FOOTER	6	2	UINT16	Error Code
	8	1	BOOL	Success Flag

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess
7	ErrorNotImplemented
9	ErrorProtocol

3.7. MESSAGES FOR MANUAL ROBOT CONTROL

3.7.1. MESSAGE #11. MOTION CONTROL

Minimum supported version: 1.0.0 (Open Source).
Support in KukavarProxy: No.

PURPOSE

Initiate a movement of type PTP, PTP_REL, LIN or LIN_REL.

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 4 + LP * 2
	4	1	UINT8	Message Type Value: 11
PAYLOAD	5	1	UINT8	Motion Type: 1 – PTP 2 – LIN 3 – PTP_REL 4 – LIN_REL
	6	2	UINT16	**LP** Length of Position String (in characters)
	8	LP * 2	WSTRING	Position String

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 5
	4	1	UINT8	Message Type Value: 11
PAYLOAD	5	1	UINT8	Motion Type
FOOTER	6	2	UINT16	Error Code
	8	1	BOOL	Success Flag

SAMPLE REQUEST

	Offset (bytes)	Size (bytes)	Type	Value
HEADER	0	2	UINT16	128
	2	2	UINT16	74
	4	1	UINT8	11
PAYLOAD	5	1	UINT8	1 (PTP)
	6	2	UINT16	35
	8	70	WSTRING	{POS: X 0, Y 0, Z 0, A 0, B 0, C 0}

SAMPLE RESPONSE

	Offset (bytes)	Size (bytes)	Type	Value
HEADER	0	2	UINT16	128
	2	2	UINT16	5
	4	1	UINT8	11
PAYLOAD	5	1	UINT8	1 (PTP)
FOOTER	6	2	UINT16	1 (ErrorSuccess)
	8	1	BOOL	TRUE

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess
7	ErrorNotImplemented
9	ErrorProtocol

3.7.2. MESSAGE #12. KCP KEY EMULATION

Minimum supported version: 1.0.0 (Open Source).
Support in KukavarProxy: No.

PURPOSE

Emulation of button pushing on the KCP device.

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 12
	4	1	UINT8	Message Type Value: 12
PAYLOAD	5	1	UINT8	Key Type: 1 – Start Key 2 – Stop Key 3 – Jog Key 4 – 6D Space Mouse
	6	4	INT32	Interpreter Type: 0 – Sumbit Interpreter 1 – Robot Interpreter *or* Axis Number
	10	4	INT32	Key Code
	14	1	BOOL	Direction

	15	1	BOOL	Key Status TRUE – Released FALSE - Pressed

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 5
	4	1	UINT8	Message Type Value: 12
PAYLOAD	5	1	UINT8	Key Type
FOOTER	6	2	UINT16	Error Code
	8	1	BOOL	Success Flag

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess
7	ErrorNotImplemented
9	ErrorProtocol

3.8. SERVICE MESSAGES

3.8.1. MESSAGE #13. GET PROXY INFORMATION

Minimum supported version: 1.0.0 (Open Source).
Support in KukavarProxy: No.

PURPOSE

Request information about the C3 Bridge Interface Server.

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 1
	4	1	UINT8	Message Type Value: 13
PAYLOAD	NO PAYLOAD			

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 25 + LCN * 2
	4	1	UINT8	Message Type Value: 13

PAYLOAD	5	1	UINT8	Version Major Number
	6	1	UINT8	Version Minor Number
	7	1	UINT8	Version Type (look at section 1.5. SOFTWARE VERSIONING)
	8	2	UINT16	Current Year
	10	2	UINT16	Current Month
	12	2	UINT16	Current Day of Week
	14	2	UINT16	Current Day
	16	2	UINT16	Current Hour (UTC)
	18	2	UINT16	Current Minute (UTC)
	20	2	UINT16	Current Second (UTC)
	22	2	UINT16	Current Millisecond
	24	2	UINT16	**LCN** Length of Computer Name (in characters)
	26	LCN * 2	WSTRING	Computer Name
FOOTER	26 + LCN * 2	2	UINT16	Error Code
	28 + LCN * 2	1	BOOL	Success Flag

SAMPLE REQUEST

	Offset (bytes)	Size (bytes)	Type	Value
HEADER	0	2	UINT16	0
	2	2	UINT16	1
	4	1	UINT8	13

SAMPLE RESPONSE

	Offset (bytes)	Size (bytes)	Type	Value
HEADER	0	2	UINT16	0
	2	2	UINT16	47
	4	1	UINT8	13
PAYLOAD	5	1	UINT8	1
	6	1	UINT8	0
	7	1	UINT8	0 (Open Source)
	8	2	UINT16	2020
	10	2	UINT16	8
	12	2	UINT16	2
	14	2	UINT16	4
	16	2	UINT16	8
	18	2	UINT16	56
	20	2	UINT16	6
	22	2	UINT16	889
	24	2	UINT16	11

	26	22	WSTRING	VDMHOSTTEST
FOOTER	48	2	UINT16	1 (ErrorSuccess)
	50	1	BOOL	TRUE

POSSIBLE ERROR CODES

Code	Name
1	ErrorSuccess

3.8.2. MESSAGE #14. GET PROXY FEATURES

Minimum supported version: 1.0.0 (Open Source).
Support in KukavarProxy: No.

PURPOSE

Request the list of supported messages for the primary protocol of the C3 Bridge Interface Server.

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 1
	4	1	UINT8	Message Type Value: 14

PAYLOAD	**NO PAYLOAD**

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 36
	4	1	UINT8	Message Type Value: 14
PAYLOAD	5	1	UINT8	Bit field of available messages: from 255 to 248.
	6	1	UINT8	Bit field of available messages: from 247 to 240.
	…			
	35	1	UINT8	Bit field of available messages: from 15 to 8.
	36	1	UINT8	Bit field of available messages: from 7 to 0.
FOOTER	37	2	UINT16	Error Code
	39	1	BOOL	Success Flag

SAMPLE REQUEST

	Offset (bytes)	Size (bytes)	Type	Value
HEADER	0	2	UINT16	0
	2	2	UINT16	1
	4	1	UINT8	14

SAMPLE RESPONSE

	Offset (bytes)	Size (bytes)	Type	Value
HEADER	0	2	UINT16	0
	2	2	UINT16	36
	4	1	UINT8	14
PAYLOAD	5	1	UINT8	0
	6	1	UINT8	0
	7	1	UINT8	0
	8	1	UINT8	0
	9	1	UINT8	0
	10	1	UINT8	0
	11	1	UINT8	0
	12	1	UINT8	0
	13	1	UINT8	0
	14	1	UINT8	0
	15	1	UINT8	0
	16	1	UINT8	0

	17	1	UINT8	0
	18	1	UINT8	0
	19	1	UINT8	0
	20	1	UINT8	0
	21	1	UINT8	0
	22	1	UINT8	0
	23	1	UINT8	0
	24	1	UINT8	0
	25	1	UINT8	0
	26	1	UINT8	0
	27	1	UINT8	0
	28	1	UINT8	0
	29	1	UINT8	80h (Message #63)
	30	1	UINT8	0
	31	1	UINT8	0
	32	1	UINT8	0
	33		UINT8	0
	34		UINT8	0
	35		UINT8	7Ch (Messages ##10-14)
	36	1	UINT8	F3h (Messages ##0,1,4-7)

FOOTER	37	2	UINT16	1 (ErrorSuccess)
	39	1	BOOL	TRUE

POSSIBLE ERROR CODES

Code	Name
1	ErrorSuccess

3.8.3. MESSAGE #15. GET PROXY INFORMATION (EXTENDED)

Minimum supported version: None.
Support in KukavarProxy: No.

This feature has not yet been implemented; the section is reserved for future use.

3.8.4. MESSAGE #16. GET CROSS3 INFORMATION

Minimum supported version: None.
Support in KukavarProxy: No.

This feature has not yet been implemented; the section is reserved for future use.

3.8.5. MESSAGE #17. PERFORM PROXY BENCHMARK

Minimum supported version: 1.2.0 (Open Source).
Support in KukavarProxy: No.

PURPOSE

Calculation of execution time for multiple read or write operations of KRL variables.

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 17
PAYLOAD	5	1	BOOL	**WRITE** Type of Operation: FALSE – Read TRUE – Write
	6	1	UINT8	Count of Variables Value: 0-255
	7	4	UINT32	Number of Iterations Value: 0-4294967294
	11	2	UINT16	**LVN1** Length of First Variable (in characters)
	13	LVN1 * 2	WSTRING	First Variable

	13 + LVN1 * 2	2	UINT16	**ONLY WHEN** **WRITE == TRUE** **LVV1** Length of First Variable Value (in characters)
	15 + LVN1 * 2	LVV1 * 2	WSTRING	**ONLY WHEN** **WRITE == TRUE** First Variable Value
	…			
	variable	2	UINT16	**LVNL** Length of Last Variable (in characters)
	variable	LVNL * 2	WSTRING	Last Variable
	variable	2	UINT16	**ONLY WHEN** **WRITE == TRUE** **LVVL** Length of Last Variable Value (in characters)
	variable	LVVL * 2	WSTRING	**ONLY WHEN** **WRITE == TRUE** Last Variable Value

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 22
	4	1	UINT8	Message Type Value: 17
PAYLOAD	5	1	BOOL	Type of Operation: FALSE – Read TRUE – Write
	6	1	UINT8	Count of Variables
	7	4	UINT32	Number of Iterations
	11	4	UINT32	Start Time (ms)
	15	4	UINT32	Stop Time (ms)
	19	4	UINT32	Time Difference (ms)
FOOTER	23	2	UINT16	Error Code
	25	1	BOOL	Success Flag

POSSIBLE ERROR CODES

Code	Name
1	ErrorSuccess
9	ErrorProtocol

3.9. MESSAGES FOR FILE OPERATIONS

3.9.1. CONSTANT VALUES

This section lists the basic constants used in file system operations.

3.9.1.1 Item Attributes

Value	Name
0	None
1	Read Only
2	Hidden
4	System
16	Directory
32	Archive
16 384	Encrypted
268 435 456	Signed

3.9.1.2 Item Types

Value	Name
0	Unknown
1	Directory
2	Virtual Directory
4	Archive
8	Binary File
16	Text File
32	Module
64	Raw

128	Motion File
256	Protected File Container

3.9.1.3 Item List Flags

Value	Name
0	None
1	Recursive
2	Expand
4	Long
8	Old Long
16	No PFC
32	PFC as File
64	ZIP as File

3.9.1.4 Module Parts

Value	Name
0	Unknown
1	SUB
2	SRC
4	DAT
5	SUBDAT
6	SRCDAT
8	Template
16	Motion

3.9.1.5 Copy Flags

Value	Name
0	None
1	Archive
3	Modify
4	Continue
8	Recursive
16	Refresh
48	Update
64	Overwrite Exists
128	No Directory Entries
256	Junk Directory
512	Force Binary
1024	Force Text
2048	No Version Check
4096	Overwrite Read-Only
8192	No KRL Analysis

3.9.1.6 Item Property Flags

Value	Name
0	None
1	Type
2	Name
4	Size
8	Attributes

16	Creation Time
32	Access Time
64	Modified Time
128	Edit Mode
256	All

3.9.1.7 Edit Modes

Value	Name
-1	Unknown
0	Full Edit
1	DatKor
2	ProKor
3	Read Only

3.9.1.8 File IO Operations

Value	Name
0	None
1	Begin Operation
2	Data Transfer
3	Get Data Size
4	End Operation
5	Get Data Checksum

3.9.2. MESSAGE #20. SET FILE ATTRIBUTES

Minimum supported version: 1.1.2 (Freeware).
Support in KukavarProxy: No.

PURPOSE

Changes file attributes.

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 20
PAYLOAD	5	4	INT32	Attributes 3.9.1.1 Item Attributes
	9	4	INT32	Mask 3.9.1.1 Item Attributes
	13	2	UINT16	**LN** Length of Name (in characters)
	15	LN * 2	WSTRING	Name

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 4
	4	1	UINT8	Message Type Value: 20
PAYLOAD	**NO PAYLOAD**			
FOOTER	5	2	UINT16	Error Code
	7	1	BOOL	Success Flag

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess
7	ErrorNotImplemented
9	ErrorProtocol

3.9.3. MESSAGE #21. LIST DIRECTORY CONTENTS

Minimum supported version: 1.1.2 (Freeware).
Support in KukavarProxy: No.

PURPOSE

Get a list of files and subdirectories in a directory.

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 21
PAYLOAD	5	4	INT32	Item Type 3.9.1.2 Item Types
	9	4	INT32	Flags 3.9.1.3 Item List Flags
	13	2	UINT16	**LP** Length of Path (in characters)
	15	LN * 2	WSTRING	Path

RESPONSE

	Offset (bytes)	**Size (bytes)**	**Type**	**Meaning**
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 21
PAYLOAD	5	2	UINT16	Count of Items
	7	2	UINT16	**IL1** Length of First Item
	9	IL1 * 2	WSTRING	First Item
	…			
	variable	2	UINT16	**ILL** Length of Last Item
	variable	ILL * 2	WSTRING	Last Item
	variable	2	UINT16	Count of Item Infos
	variable	2	UINT16	**IIL1** Length of First Item Info
	variable	IIL1 * 2	WSTRING	First Item Info
	…			
	variable	2	UINT16	**IILL** Length of Last Item Info
	variable	IILL * 2	WSTRING	Last Item Info
FOOTER	*variable*	2	UINT16	Error Code
	variable	1	BOOL	Success Flag

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess
7	ErrorNotImplemented
9	ErrorProtocol
10	ErrorLongAnswer

3.9.4. MESSAGE #22. CREATE NEW FILE

Minimum supported version: 1.1.2 (Freeware).
Support in KukavarProxy: No.

PURPOSE

Create a new text file, binary file or KRL module.

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 22
PAYLOAD	5	2	UINT16	Item Type 3.9.1.2 Item Types
	7	1	UINT8	Module Part 3.9.1.4 Module Parts
	8	1	BOOL	Always Create

	9	2	UINT16	**LN** Length of Name (in characters)
	11	LN * 2	WSTRING	Name
	variable	2	UINT16	**LTP** Length of Template (in characters)
	variable	LTP * 2	WSTRING	Template

RESPONSE

	Offset (bytes)	**Size (bytes)**	**Type**	**Meaning**
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 4
	4	1	UINT8	Message Type Value: 22
PAYLOAD	**NO PAYLOAD**			
FOOTER	5	2	UINT16	Error Code
	7	1	BOOL	Success Flag

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess
7	ErrorNotImplemented
9	ErrorProtocol

3.9.5. MESSAGE #23. DELETE FILE

Minimum supported version: 1.1.2 (Freeware).
Support in KukavarProxy: No.

PURPOSE

Delete a file.

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 23
PAYLOAD	5	1	BOOL	Always Delete
	6	2	UINT16	**LN** Length of Name (in characters)
	8	LN * 2	WSTRING	Name

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 4
	4	1	UINT8	Message Type Value: 23
PAYLOAD	NO PAYLOAD			
FOOTER	5	2	UINT16	Error Code
	7	1	BOOL	Success Flag

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess
7	ErrorNotImplemented
9	ErrorProtocol

3.9.6. MESSAGE #24. COPY FILE

Minimum supported version: 1.1.2 (Freeware).
Support in KukavarProxy: No.

PURPOSE

Copy a file to another location.

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 24
PAYLOAD	5	4	INT32	Flags 3.9.1.5 Copy Flags
	9	2	UINT16	**LS** Length of Source (in characters)
	11	LS * 2	WSTRING	Source
	variable	2	UINT16	**LD** Length of Destination (in characters)
	variable	LD * 2	WSTRING	Destination

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 4
	4	1	UINT8	Message Type Value: 24
PAYLOAD	NO PAYLOAD			
FOOTER	5	2	UINT16	Error Code
	7	1	BOOL	Success Flag

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess
7	ErrorNotImplemented
9	ErrorProtocol

3.9.7. MESSAGE #25. MOVE FILE

Minimum supported version: 1.1.2 (Freeware).
Support in KukavarProxy: No.

PURPOSE

Move a file to another location.

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 25
PAYLOAD	5	4	INT32	Flags 3.9.1.5 Copy Flags
	9	2	UINT16	**LS** Length of Source (in characters)
	11	LS * 2	WSTRING	Source
	variable	2	UINT16	**LD** Length of Destination (in characters)
	variable	LD * 2	WSTRING	Destination

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 4
	4	1	UINT8	Message Type Value: 25
PAYLOAD	**NO PAYLOAD**			
FOOTER	5	2	UINT16	Error Code
	7	1	BOOL	Success Flag

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess
7	ErrorNotImplemented
9	ErrorProtocol

3.9.8. MESSAGE #26. GET FILE PROPERTIES

Minimum supported version: 1.1.2 (Freeware).
Support in KukavarProxy: No.

PURPOSE

Getting a file's type, attributes, size and time.

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 26
PAYLOAD	5	4	INT32	Flags 3.9.1.6 Item Property Flags
	9	2	UINT16	**LN** Length of File Name (in characters)
	11	LN * 2	WSTRING	File Name

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 26
PAYLOAD	5	4	INT32	Item Type 3.9.1.2 Item Types
	9	4	INT32	Item Size (Low Part)
	13	4	INT32	Item Size (High Part)
	17	4	INT32	Attributes 3.9.1.1 Item Attributes
	21	4	INT32	Creation Time(Low Part) A Windows file creation time expressed in ticks
	25	4	INT32	Creation Time (High Part)
	29	4	INT32	Creation Time Bias (minutes)
	33	4	INT32	Access Time (Low Part) A Windows file last access time expressed in ticks
	37	4	INT32	Access Time (High Part)
	41	4	INT32	Access Time Bias (minutes)

	45	4	INT32	Modified Time (Low Part) A Windows file last modification time expressed in ticks
	49	4	INT32	Modified Time (High Part)
	53	4	INT32	Modified Time Bias (minutes)
	57	4	INT32	Edit Mode 3.9.1.7 Edit Modes
	61	2	UINT16	**LN** Length of Item Name
	63	LN * 2	WSTRING	Item Name
FOOTER	63 + LN * 2	2	UINT16	Error Code
	65 + LN * 2	1	BOOL	Success Flag

POSSIBLE ERROR CODES

Code	**Name**
0	ErrorGeneral
1	ErrorSuccess
9	ErrorProtocol

3.9.9. MESSAGE #27. GET FILE FULL PATH

Minimum supported version: 1.1.2 (Freeware).
Support in KukavarProxy: No.

PURPOSE

Getting the full path to a file.
For example, the path "*/R1/TEST.SRC*" will be converted to "*KRC:\R1\PROGRAM\TEST.SRC*".

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 27
PAYLOAD	5	2	UINT16	**LP** Length of File Path (in characters)
	7	LP * 2	WSTRING	File Path

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 27

PAYLOAD	5	2	UINT16	**LP** Length of Output Path
	7	LP * 2	WSTRING	Output Path
FOOTER	7 + LP * 2	2	UINT16	Error Code
	9 + LP * 2	1	BOOL	Success Flag

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess
9	ErrorProtocol

3.9.10. MESSAGE #28. GET KRC PATH

Minimum supported version: 1.1.2 (Freeware).
Support in KukavarProxy: No.

PURPOSE

Getting the KRC path to a file.
For example, the path "*KRC:\R1\PROGRAM\TEST.SRC*" will be converted to "*/R1/TEST.SRC*".

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 28

PAYLOAD	5	2	UINT16	**LP** Length of File Path (in characters)
	7	LP * 2	WSTRING	File Path

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 28
PAYLOAD	5	2	UINT16	**LP** Length of Output Path
	7	LP * 2	WSTRING	Output Path
FOOTER	7 + LP * 2	2	UINT16	Error Code
	9 + LP * 2	1	BOOL	Success Flag

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess
9	ErrorProtocol

3.9.11. MESSAGE #29. WRITE FILE CONTENT (BEGINNING)

Minimum supported version: 1.1.2 (Freeware).
Support in KukavarProxy: No.

PURPOSE

FileIoBegin: Creates a new buffer of a given size for further data writing to it. Any previously created buffer is destroyed along with the data.
FileIoGetSize: Specifies the size of a previously allocated buffer, or 0 if no buffer exists.

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 29
PAYLOAD	5	1	UINT8	**OPERATION** 1 – FileIoBegin 3 – FileIoGetSize
	6	4	UINT32	**ONLY WHEN OPERATION == 1** Total Size (in bytes)

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 9
	4	1	UINT8	Message Type Value: 29
PAYLOAD	5	1	UINT8	Operation
	6	4	UINT32	Buffer Size
FOOTER	10	2	UINT16	Error Code
	12	1	BOOL	Success Flag

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess
7	ErrorNotImplemented
9	ErrorProtocol

3.9.12. MESSAGE #29. WRITE FILE CONTENT (DATA CHUNK)

Minimum supported version: 1.1.2 (Freeware).
Support in KukavarProxy: No.

PURPOSE

FileIoData: Writes a chunk of data of a certain size to the buffer at the specified offset.

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 29
PAYLOAD	5	1	UINT8	Operation 2 – FileIoData
	6	4	UINT32	Offset (in bytes)
	10	4	UINT32	**SIZE** Chunk Size (in bytes)
	14	SIZE	BINARY	Chunk Data

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 13
	4	1	UINT8	Message Type Value: 29
PAYLOAD	5	1	UINT8	Operation Value: 2
	6	4	UINT32	Offset (in bytes)
	10	4	UINT32	Size of Bytes Written
FOOTER	14	2	UINT16	Error Code
	16	1	BOOL	Success Flag

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess
7	ErrorNotImplemented
9	ErrorProtocol

3.9.13. MESSAGE #29. WRITE FILE CONTENT (CHECKSUM)

Minimum supported version: None.
Support in KukavarProxy: No.

This feature has not yet been implemented; the section is reserved for future use.

3.9.14. MESSAGE #29. WRITE FILE CONTENT (FINAL)

Minimum supported version: 1.1.2 (Freeware).
Support in KukavarProxy: No.

PURPOSE

FileIoEnd: Writes the contents of the buffer to a file on disk.

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 29
PAYLOAD	5	1	UINT8	Operation 4 – FileIoEnd
	6	4	INT32	Flags 3.9.1.5 Copy Flags
	10	2	UINT16	**LN** Length of File Name
	12	LN * 2	WSTRING	File Name

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 5
	4	1	UINT8	Message Type Value: 29
PAYLOAD	5	1	UINT8	Operation Value: 4
FOOTER	6	2	UINT16	Error Code
	8	1	BOOL	Success Flag

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess
7	ErrorNotImplemented
9	ErrorProtocol

3.9.15. MESSAGE #30. READ FILE CONTENT (BEGINNING)

Minimum supported version: 1.1.2 (Freeware).
Support in KukavarProxy: No.

PURPOSE

FileIoBegin: Transfers the contents of a file on disk to the read buffer and returns its size.
FileIoGetSize: Specifies the size of the read buffer, or 0 if no buffer exists.

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 30
PAYLOAD	5	1	UINT8	**OPERATION** 1 – FileIoBegin 3 – FileIoGetSize
	6	4	INT32	**ONLY WHEN OPERATION == 1** Flags 3.9.1.5 Copy Flags
	10	2	UINT16	**ONLY WHEN OPERATION == 1** **LN** Length of File Name

	12	LN * 2	WSTRING	**ONLY WHEN OPERATION == 1** File Name

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 9
	4	1	UINT8	Message Type Value: 30
PAYLOAD	5	1	UINT8	Operation
	6	4	UINT32	Buffer Size
FOOTER	10	2	UINT16	Error Code
	12	1	BOOL	Success Flag

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess
7	ErrorNotImplemented
9	ErrorProtocol

3.9.16. MESSAGE #30. READ FILE CONTENT (DATA CHUNK)

Minimum supported version: 1.1.2 (Freeware).
Support in KukavarProxy: No.

PURPOSE

FileIoData: Reads a data fragment of the specified size into the buffer at the specified offset.

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 30
PAYLOAD	5	1	UINT8	Operation 2 – FileIoData
	6	4	UINT32	Offset (in bytes)
	10	4	UINT32	Maximum Chunk Size (in bytes) This value can be FFFFFFFFh to read the largest amount of data at the time.

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 30
PAYLOAD	5	1	UINT8	Operation Value: 2
	6	4	UINT32	Offset (in bytes)
	10	4	UINT32	**SIZE** Size of Bytes Read
	14	SIZE	BINARY	Chunk Data
FOOTER	*variable*	2	UINT16	Error Code
	variable	1	BOOL	Success Flag

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess
7	ErrorNotImplemented
9	ErrorProtocol

3.9.17. MESSAGE #30. READ FILE CONTENT (CHECKSUM)

Minimum supported version: None.
Support in KukavarProxy: No.

This feature has not yet been implemented; the section is reserved for future use.

3.9.18. MESSAGE #30. READ FILE CONTENT (FINAL)

Minimum supported version: 1.1.2 (Freeware).
Support in KukavarProxy: No.

PURPOSE

FileIoEnd: Clearing the read buffer and freeing the memory.

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length
	4	1	UINT8	Message Type Value: 30
PAYLOAD	5	1	UINT8	Operation 4 – FileIoEnd

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 5
	4	1	UINT8	Message Type Value: 30
PAYLOAD	5	1	UINT8	Operation Value: 4
FOOTER	6	2	UINT16	Error Code
	8	1	BOOL	Success Flag

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess
7	ErrorNotImplemented
9	ErrorProtocol

3.10. MESSAGES FOR CROSSCOMMEXE COMPATIBILITY

3.10.1. MESSAGE #64. CONFIRM ALL

Minimum supported version: 1.0.0 (Open Source).
Support in KukavarProxy: No.

PURPOSE

Reset all errors on the KRC (emulation of pressing the Confirm All button).

REQUEST

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 1
	4	1	UINT8	Message Type Value: 63
PAYLOAD	NO PAYLOAD			

RESPONSE

	Offset (bytes)	Size (bytes)	Type	Meaning
HEADER	0	2	UINT16	Tag ID
	2	2	UINT16	Message Length Value: 4

	4	1	UINT8	Message Type Value: 63
PAYLOAD	NO PAYLOAD			
FOOTER	5	2	UINT16	Error Code
	7	1	BOOL	Success Flag

POSSIBLE ERROR CODES

Code	Name
0	ErrorGeneral
1	ErrorSuccess