ALIS Protocol Spesification

ALIS (Automation Layer Intercommunication System) is a toy protocol created to demonstrate various network artifacts for the purpose of knowledge transfer. It is a request-response protocol. This protocol operates in litte-endian order.

The controller side holds ten 64-bit registers and a boolean value representing the running state, which can be queried. There is a 2-byte session identifier used throughout the lifetime of a session to keep track of sessions. This variable randomly selected by the client (workstation).

Message Types

Message Code	Name	Description
0x01	NEW-CONNECTION	Initiates new connection.
0x02	READ-REGISTERS	Reads registers on the memory.
0x03	WRITE-REGISTERS	Writes registers to the memory.
0x04	GET-STATUS	Queries controller status.
0xFA , 0x05	STOP-START	Stop – Start (Toggle Switch)

NEW-CONNECTION (Code 0x01)

When workstation wants to communicate with the controller, the message it sends is a *NEW-CONNECTION-REQUEST* message. The controller will respond with a *NEW-CONNECTION-RESPONSE*.

NEW-CONNECTION-REQUEST

Byte / Byte Range	Field Name	Description
[0x00-0x01]	Session Identifier	Random identifier generated by client
0x02	Message Code	NEW-CONNECTION (0x01)
[0x03-0x04]	Checksum	CRC-16 of the previous bytes in the frame
[0x05-0x08]	Timestamp	32 Bit UNIX timestamp (seconds)

NEW-CONNECTION-RESPONSE (Success)

Byte / Byte Range	Field Name	Description
[0x00-0x01]	Session Identifier	Random identifier generated by client (resent)
0x02	Message Code	NEW-CONNECTION (0x01)

READ-REGISTERS (Code 0x02)

This is used when the workstation wants to read registers. All registers will be returned in the response.

READ-REGISTERS-REQUEST

Byte / Byte Range	Field Name	Description
[0x00-0x01]	Session Identifier	Random identifier generated by client (resent)
0x02	Message Code	READ-REGISTERS (0x02)
[0x03-0x04]	Checksum	CRC-16 of the previous bytes in the frame
[0x05-0x08]	Timestamp	32 Bit UNIX timestamp (seconds)

READ-REGISTERS-RESPONSE (Success)

Byte / Byte Range	Field Name	Description
[0x00-0x01]	Session Identifier	Random identifier generated by client (resent)
[0x02-END]	List of Registers	80 byte data containing ten 8 byte registers

WRITE-REGISTERS (Code 0x03)

This is used when the workstation wants to write to registers. It takes register indexes and the variables to be written as fields. Register indexes that are not intended to be changed should be filled with 0xFF.

IC: Index Count

WRITE-REGISTERS-REQUEST

Byte / Byte Range	Field Name	Description
[0x00-0x01]	Session Identifier	Random identifier generated by client
0x02	Message Code	WRITE-REGISTERS (0x03)
[0x03-0x0C]	Registers Indexes	Indexes of the registers intented for writing
[0x0C-0x0C+0x08**IC*]	Register Values	Values intented for writing
[(END-0x05)-(END-0x04)]	Checksum	CRC-16 of the previous bytes in the frame
[(END-0x03)-END]	Timestamp	32 Bit UNIX timestamp (seconds)

WRITE-REGISTERS-RESPONSE (Success)

Byte / Byte Range	Field Name	Description
[0x00-0x01]	Session Identifier	Random identifier generated by client (resent)
0x02	Message Code	WRITE-REGISTERS (0x03)

GET-STATUS (Code 0x04)

This is used when the workstation wants to query the software version on the controller and it's current running status.

GET-STATUS-REQUEST

Byte / Byte Range	Field Name	Description
[0x00-0x01]	Session Identifier	Random identifier generated by client (resent)
0x02	Message Code	GET-STATUS (0x04)
[0x03-0x04]	Checksum	CRC-16 of the previous bytes in the frame
[0x05-0x08]	Timestamp	32 Bit UNIX timestamp (seconds)

GET-STATUS-RESPONSE (Success)

Byte / Byte Range	Field Name	Description
[0x00-0x01]	Session Identifier	Random identifier generated by client (resent)
[0x02-END]	Status String	UTF-8 status string with an undefined length

STOP-START (Code 0xFA, 0x05)

This is used when the workstation wants to toggle the run status of the controller.

STOP-START-REQUEST

Byte / Byte Range	Field Name	Description
[0x00-0x01]	Session Identifier	Random identifier generated by client (resent)
0x02	Message Code	NEW-CONNECTION (0x05)
[0x03-0x04]	Checksum	CRC-16 of the previous bytes in the frame
[0x05-0x08]	Timestamp	32 Bit UNIX timestamp (seconds)

STOP-START-RESPONSE (Success)

Byte / Byte Range	Field Name	Description
[0x00-0x01]	Session Identifier	Random identifier generated by client (resent)
0x02	Message Code	0x05 for running, 0xFA (invert) for stopped