HE Valve – CAN Communication with Main Board

This document provides Technical Specifications for communication between the lift's control panel and the HEVOS SCH001 board of the pump unit, using CAN bus.

1 MANAGEMENT OF OPERATING SIGNALS BETWEEN CONTROL PANEL AND SCH001 BOARD (SINGLE AND MULTI-VALVE SYSTEM)

1) Each device connected to the CAN bus constitutes a node with its own address.

2) In communication, the **electronic board** SCH001 of the single **valve** (or one of the boards, in the case of the multi-valve system) will act as the master node, while the control panel will be a slave node.

3) The single SCH001 control board (or each board, in the case of the multi-valve system) will provide "physically" to the control panel **through a dedicated electrical cable connection**, independent of CAN:

- the status of each **AVV** relay, for starting and maintaining the power supply of each motor of the system, or commanding the closing of at least one contactor of each individual motor and its opening when the state of the AVV relay changes (also during the manoeuvre);
- the status of the **ERR** relay or the cumulative ERR signal, in the case of the multi-valve system (the cumulative *ERR* signal is obtained from the series of all NC contacts of the *ERR* relays of the various boards or from the parallel of all NO contacts of the ERR relays of the various boards).

⚠NOTE: the SCH001 board does not provide the status of the motor thermal protection and therefore the specific input of the control panel must be connected directly to the wires of the motor thermistors (if there are several motors, as a series of all the protections).

4) The master board will receive from the panel, via CAN line, the digital signals in **INPUT to the board**:

- *UP* (upward signal);
- *DW* (downward signal);
- *HSP* (upward high speed signal, during Downward also when Par. 453 = 0);
- *MSP* (maintenance and speed signal V2);
- *SFY* (pump motor started signal);
- *SP1* (speed V3 (combined with the high speed signal));
- *SP2* (speed V4 (combined with the high speed signal));
- *SP3* (downward high speed signal, when Par. 453 = 1);

which, in the case of multi-valve systems, will then be transmitted to the slave boards via CAN bus by the master. In the multi-valve system, the control panel must ensure that SFY = 1 only when all the motors have started, or when the contactors of each motor are all closed due to direct start-up or when the motor start-up phases are completed in case of soft starter or star delta starting, while the SFY signal must go to zero when the motor (s) is stopped.

5) The board will transmit the status to the control panel, via CAN line, as **OUTPUT from the board**:

- of the following relays:

- *T1* (cumulative state, in the case of multi-valve system): oil temperature range exit;
- *P1* (cumulative state, in the case of multi-valve system): min e max pressure range exit;
- *P2* (cumulative state, in the case of multi-valve system): overload pressure overcoming;
- *RDY* (cumulative state, in the case of multi-valve system);
- *AVV* (it was valid only in the case of a single system, no multi-valve);
- *ERR* (cumulative state, in the case of multi-valve system, valid only if CAN communication is working).

- of the following voltage outputs:

- *PNP1* (cumulative state, in the case of multi-valve system);
- *PNP2* (it was valid only in the case of a single system, no multi-valve);

6) To **avoid conflicts** between the various devices interconnected by CAN bus, it is required that:

is **reserved for the** SCH001 **control board** (or boards, in the case of the multi-valve system) a range of **96 consecutive addresses** (for example 0x550-0x5AF). These addresses cannot in any way be used by devices normally connected by CAN bus to the control panel because they are exclusively for use with the SCH001 control board (or boards, in the case of the multi-valve system);

- the second address of the range (for example 0x551) is the transmission ID of the "master" node;
- The fiftieth address of the range (for example 0x581) is the identifier in reception of the "control panel node".

2 SETTING OF CAN PARAMETERS

It is necessary to configure CAN bus in the following way:

1) **P113** (=P479) basic value for CAN addresses (offset).

- With default value = 0, a basic address of 0x550 (1360) is assumed automatically, corresponding to the decimal value (1360).
- Specify a different decimal value, if necessary, to guarantee 96 consecutive free addresses, in case the default ones are already occupied by the control panel board for other communications.

2) **P114** (=P480) relative address of the control panel node.

- With default value = 0, the SCH001 control board does **not recognize** operating signals transmitted via the CAN line.
- With the used value = 49, the SCH001 control board recognizes the operating signals transmitted via CAN line.

3) **P476** (address of the SCH001 board **-** master node) to receive messages from the control panel, always referred to the offset.

- With default value = 1 for the master board is the first of the network addresses. If there is only one board, no change is required, otherwise you will need to assign a unique number from 2 to 8 when adding slaves.

4) The **Jumper JP5** should only be placed on the last board of the series in multi-valve systems or on the only board in single-valve systems, in actual fact only on the board with only one CAN cable.

3 COMMUNICATION PROTOCOL BETWEEN LIFT CONTROL PANEL AND PUMP UNIT

CAN bus communication will be established with a speed of 125 kbit/s, with standard 11-bit identifier.

3.1 COMMUNICATION ELECTRONIC BOARD → CONTROL PANEL

Every 100ms, the SCH001 board (master node) will transmit to the lift's control panel as an 8-byte "packet" (8bit), following information:

- **1st byte** equal to *0x61*, to communicate the beginning of the transmission,
- **2nd byte** with the status of the SCH001 board, or the status of the *AVV, T1, P1, P2, RDY* relays and voltage output *PNP1, PNP2* according to the following format represented by a sequence of 8 values 0/1:

Bit 7-MSB	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0- LSB
ERR	PNP2	AVV	T1	P1	P2	RDY	PNP1

according to convention:

  - 0 = relay de-energized, PNP1, PNP2 = low logic level;
  - 1 = relay energized, PNP1, PNP2 = high logic level.

⚠NOTE: the status of the AVV relay and of voltage output PNP2 are valid only in the case of single system, not multi-valve. In any case for the AVV relay, the physical connection is valid (see chapter1)

To consent **diagnostic functions**, additional **six bytes** has been available as option by setting P496=1. These bytes consent an additional function of data exchange with the control panel, allowing it to interrogate the electronic board to acquire the parameters values and even errors codes, as specified in the table below:

- **3rd byte** bit 0-7 Bit 0-7 of the parameter code
- **4th byte** bit 0-7 Bit 8-15 of the parameter code
- **5th byte** bit 0-7 Bit 0-7 of the parameter value
- **6th byte** bit 0-7 Bit 8-15 of the parameter value
- **7th byte** bit 0-7 Bit 16-23 of the parameter value
- **8th byte** bit 0-7 Bit 24-31 of the parameter value

3.2 COMMUNICATION CONTROL PANEL → ELECTRONIC BOARD

Upon receipt of the above mentioned package, the lift control panel will reply to the SCH001 board with a four byte "package" as follows:

- **1st byte** equal to *0x68*, to communicate the beginning of the transmission to the electronic board;
- **2nd byte** with the status of the digital signals *UP*, *DW*, *HSP*, *MSP*, *SFY*, *SP1*, *SP2*, *SP3* according to the following format represented by a sequence of 8 values 0/1:

Bit 7-MSB	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0- LSB
SP3	SP2	SP1	SFY	MSP	HSP	DW	UP

According to the convention:

  - 0 = command absent;
  - 1 = command present;

- **3rd byte** represents the number of the floor on which the cabin is located and which is updated each time the deceleration magnet is encountered (0 = lowest floor);
- **4th byte** represents the number of the floor to which the cabin will have to come at the end of the manoeuvre (0 = lowest floor).

Additional bytes are available in case of **diagnostic functions** activated only (P496=1):

- **5th byte** bit 0-7 Bit 0-7 of the parameter code request
- **6th byte** bit 0-7 Bit 8-15 of the parameter code request
- **7th byte** bit 0-7 Bit 16-23 of the parameter value request
- **8th byte** bit 0-7 Bit 24-31 of the parameter value request

⚠ The lift control panel will be able to supply the **movement commands** *(UP, DW, HSP, MSP, SFY, SP1, SP2, SP3 = 1)* only when the SCH001 board is not in error (ERR relay).

⚠ The control panel must constantly **monitor the RDY signal** and behave towards the SCH001 control board of the pump unit as follows:

- provide movement commands, when RDY = 1;
- remove the movement commands, setting them to zero, when RDY = 0.

⚠ The lift control panel must go into **Error** state, due to communication timeout, if it does not receive any information from the SCH001 board within 10 s. The aforementioned timeout must be activated or deactivated using the specific commands specified below:

- *0x52* enables timeout. When the command is executed, the control panel will reply with *0x65*;
- *0x5A* disables the timeout. When the command is executed, the control panel will reply with *0x6B.*

It should be noted that at the control panel switching on the aforementioned timeout must be enabled and the situation will remain in that state up to the contrary command.

An example of the exchange of board panel communication messages is shown in chapter.8.

Parameters P521-P528 monitor the status of the individual digital inputs transmitted by the control panel:

- P521 UP signal;

- P522 DW signal;
- P523 HSP signal;
- P524 MSP signal;
- P525 SP3 signal;
- P526 SFY signal;
- P527 SP1 signal;
- P528 SP2 signal;

4 DIAGNOSTIC FUNCTIONS

4.1 COMMUNICATION LEVEL ELECTRONIC BOARD <-> CONTROL PANEL

The extended communication protocol is a specific protocol that can be adopted to allow the management and communication of a wider set of information between the Electronic Board and the Control Panel and to allow the implementation of **diagnostic functions**. It is available starting from firmware rev.393

The type of communication protocol chosen depends on the setting of some parameters:

a. P496 Communication level

- P496=0 (default) the communication between the Hevos board and the control panel is set to 2 Byte and between the control panel and Hevos board to 4 Byte,
- P496=1 the extended communication is enabled at 8 Byte for both the communication messages.

b. P498 Input signals

- P498=0 (default). The digital input signals on the connectors CN6-CN7 are disabled. The electronic board acquires the only CAN commands coming from the control panel

⚠ **The requests sent in the 5th and 6th byte of the message from the control panel to the SCH001 board are ignored if they remain the same for more than 1 second.**

This protocol can also be used by a control system external to the control panel.

The control system can delegate to the control panel the function of providing the digital manoeuvring signals to the SCH001 board using the parameter P498=1.

4.2 READING PARAMETERS AND CURVES

4.2.1 PARAMETERS AND CURVES VALUES

The enablement of the 8-Byte extended communication level, provides the possibility of transmitting data included between index 0 and index 9405 at the request of the Control Panel, introducing the code value of the requested parameter in the 5th and 6th byte of the message and 0xFFFF in the 7th and 8th byte of the message Control Panel→Board

The Hevos board responds with the same address in the 3rd and 4th byte and the value of parameter in the 5th, 6th, 7th and 8th byte of the message.

⚠ The Hevos board responds the "0xFFFFFFFF" value in the 5th, 6th, 7th and 8th byte of the message if the request from the control panel remains the same for at least 1 second.

Only with the extended communication level activated, the Control Panel may require:

- data with index from 0 to 999 that correspond to the parameters of the Hevos board.

⚠ If there is no match between the message length as per P496, the message is ignored.

- additional advanced data, with indexes between 1000 and 9405 to receive the operating curves recorded by the Hevos board, respecting the following specific procedure:
  a. request the address **20000** (on the 5th and 6th Byte of the message Control panel → Board) to enable the communication of the indexes between 1000 and 9405
  b. wait that the response to 20000 from the Hevos board changes from 0xFFFFFFFF to 1 in the 5th, 6th, 7th and 8th bytes of the message. The answer 20000 = 1 indicates that, in this moment,

the system is stopped and the curves memorization is disabled during the subsequent manoeuvres.

c. request the indexes from 1000 to 9405 progressively (on the 5th and 6th Byte of the message Control panel → Board)

⚠ During the data transferring the elevator can move, but there won't be acquired any other curve (new data acquisition and saving disabled)

d. request the index **21000** at the end of the data download up to index 9405; in that way the control panel restores the recordings on the Hevos board,

⚠ If the saving is not newly enabled, the function remains permanently disabled and no new data and curves are recorded!

e. The SCH001 Board will returns 21000=1 at the end of the current manoeuvre, to indicate that it is possible to request updated curves with the subsequent manoeuvres. Once the data up to index 9405 has been downloaded, the control panel should set up the Hevos board to start again recording the operating curves in its memory.

4.3 READING CURRENT ERRORS

For reading the current errors it is necessary that the extended communication level is activated with the parameter P496=1.

It is possible to send a request for reading the current errors values, by requesting the addresses from **13001** and **13020** in the 5th and 6th byte of the message Control Panel → Board.

It is therefore possible to request up to 20 current error codes.

The Hevos board responds with the same address in the 3rd and 4th byte and the requested value in the 5th, 6th, 7th and 8th byte of the message if the request is accepted.

The conditions for accepting the request are:

- There are errors
- The system is not operating
- There are no attempts of auto reset in progress (the error has to be definitely acquired)

⚠ If there are not errors currently, the board returns the value "0xFFFFFFFF" on the address 13001; if there is a list of current errors and the request returns the value 0xFF it means that the end of the list has been reached and the last current error is the previous one (0xFF is not an error code!).

4.4 READING THE ERROR HISTORY LOG

For reading the errors log it is necessary that the extended communication level is activated with the parameter P496=1. Under this condition it is possible to send a request to read and decode the errors list recorded in the parameters from P600 to P679.

By requesting the address from **10600** to **10679** in the 5th and 6th byte of the message, only the error code request is transmitted.

By requesting the address from **11600** to **11679** in the 5th and 6th byte of the message, only the request of the difference between the machine time and the error storage time is transmitted. This request corresponds to "how long ago the fault happened" in terms of time during which the board has been powered and active (ON)

By requesting the address from **12600** to **12679** in the 5th and 6th byte of the message, only the request of the machine time when the error was stored is transmitted.

The Hevos board responds with the same address in the 3rd and 4th byte and the requested value in the 5th, 6th, 7th and 8th byte of the message if the request is accepted.

⚠ ATTENTION: Performing a request of the indexes from 600 to 679, the response are values which contain the error code joined to the machine time when they were stored, that are difficult to interpretate.

4.5 ERROR STATE RESET

For enable a remote errors-reset functionality, it is necessary that the extended communication level is activated with the parameter P496=1, so that it is possible to send a request to reset the current errors.

By requesting the address **30000** in the 5th and 6th byte of the message Control Panel→Board, a request to enable the errors-reset function for the current errors is transmitted.

The Hevos board responds with the same address in the 3rd and 4th byte and the value 1 in the 5th, 6th, 7th and 8th byte of the message if the request is accepted, or the value 0xFFFFFFFF.

The conditions for accepting the request are:

- There are errors
- The system is not operating
- There are not auto-reset attempts in progress

Requesting the address **31000** in the 5th and 6th byte of the message Control Panel→Board, a request to reset the current errors is transmitted.

The Hevos board responds with the same address in the 3rd and 4th byte and the value 1 in the 5th, 6th, 7th and 8th byte of the message if the request is accepted, or the value 0xFFFFFFFF.

The conditions for accepting the request are:

- The function reset was previously enabled with the address 30000
- The system is not operating
- There are not auto-reset attempts in progress

⚠ The procedure also activates error reset on the Slave boards if present

4.6 MANAGEMENT OF 100, 200 AND 300 SERIES PARAMETERS

4.6.1 PARAMETERS MODIFICATION

For enable a remote parameters modification functionality, it is necessary that the extended communication level is activated with the parameter P496=1.

It is possible to modify only the following parameters:

- 100 series – General Parameters
- 200 series – UP
- 300 series - DOWN

By requesting the address **40000** in the 5th and 6th byte of the message Control Panel→Board, the request to enable the parameters modification function is transmitted.

The Hevos board responds with the same address in the 3rd and 4th byte and the acceptance value in the 5th, 6th, 7th and 8th byte of the message. This value can be:

- 1 if the activation is accepted
- 0xFFFFFFFF if the activation is rejected since the system is in motion.

If the function is activated, the control panel can send to the Hevos board the parameter code to be modified in the 5th and 6th byte and the new value in the 7th and 8th byte.

The Hevos Board responds with the same parameter code and the same new value, confirming the modification or de value 0xFFFFFFFF if then new value is not between the minimum and maximum limit values of the parameter.

⚠ The modification of one parameter resets the function; consequently, if you want to modify another parameter, it is necessary to previously enable the parameters modification function

4.6.2 PARAMETERS LIMITS

The extended communication level, activated with the parameter P496=1, enables even to read the limits stored for each parameter on the Hevos board.

- Minimum permitted value for a parameter can be read by requesting the address **26xxx** in the 5th and 6th byte of the message.
- Maximum permitted value for a parameter can be read by requesting the address from **27xxx** in the 5th and 6th byte of the message

In both cases the **xxx** code to be used for the requests is composed by "26" or "27" respectively, followed by the 3-digits ID of the parameter number (e.g. 26110 requests the minimum value of P110 – Low temperature limit, instead of 27110 requests the maximum value of the same parameter)

The Hevos board responds with the same address in the 3rd and 4th byte and the value request in the 5th, 6th, 7th and 8th byte of the message or the value 0xFFFFFFFF if the requested values are not stored.

4.6.3 DEFAULT VALUES

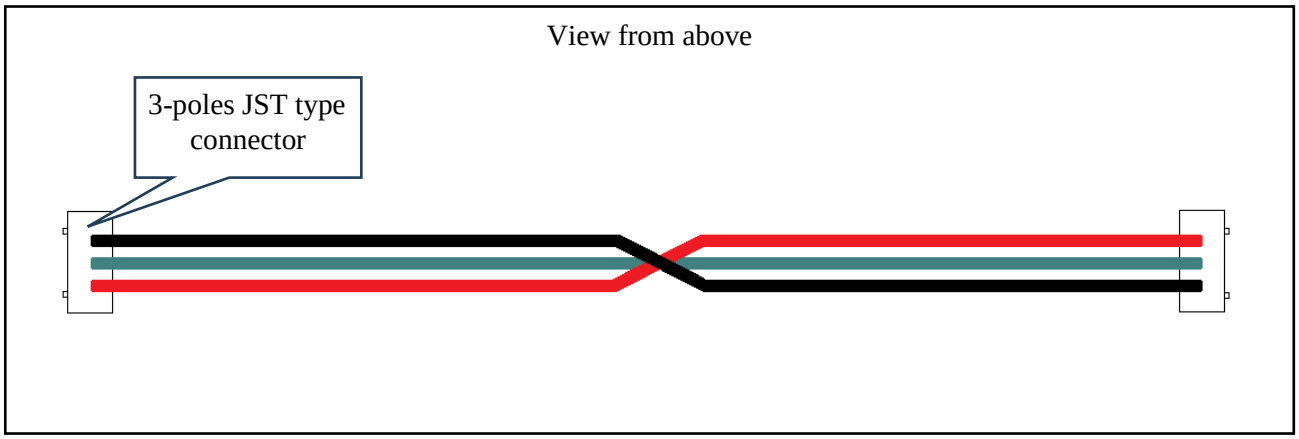
The extended communication level, activated with the parameter P496=1, enables even to read the factory's values for the parameters stored on the Hevos board.

- Default values can be read by requesting the addresses **25xxx** in the 5th and 6th byte of the message. The code to be used for the request is composed by "25" followed by the 3-digits ID of the parameter number (e.g. 25101 requests the default value of P101 - Piston diameter)

The Hevos board responds with the same address in the 3rd and 4th byte and the value requested in the 5th, 6th, 7th and 8th byte of the message, or the value 0xFFFFFFFF if the requested values are not stored

5 CAN CABLE

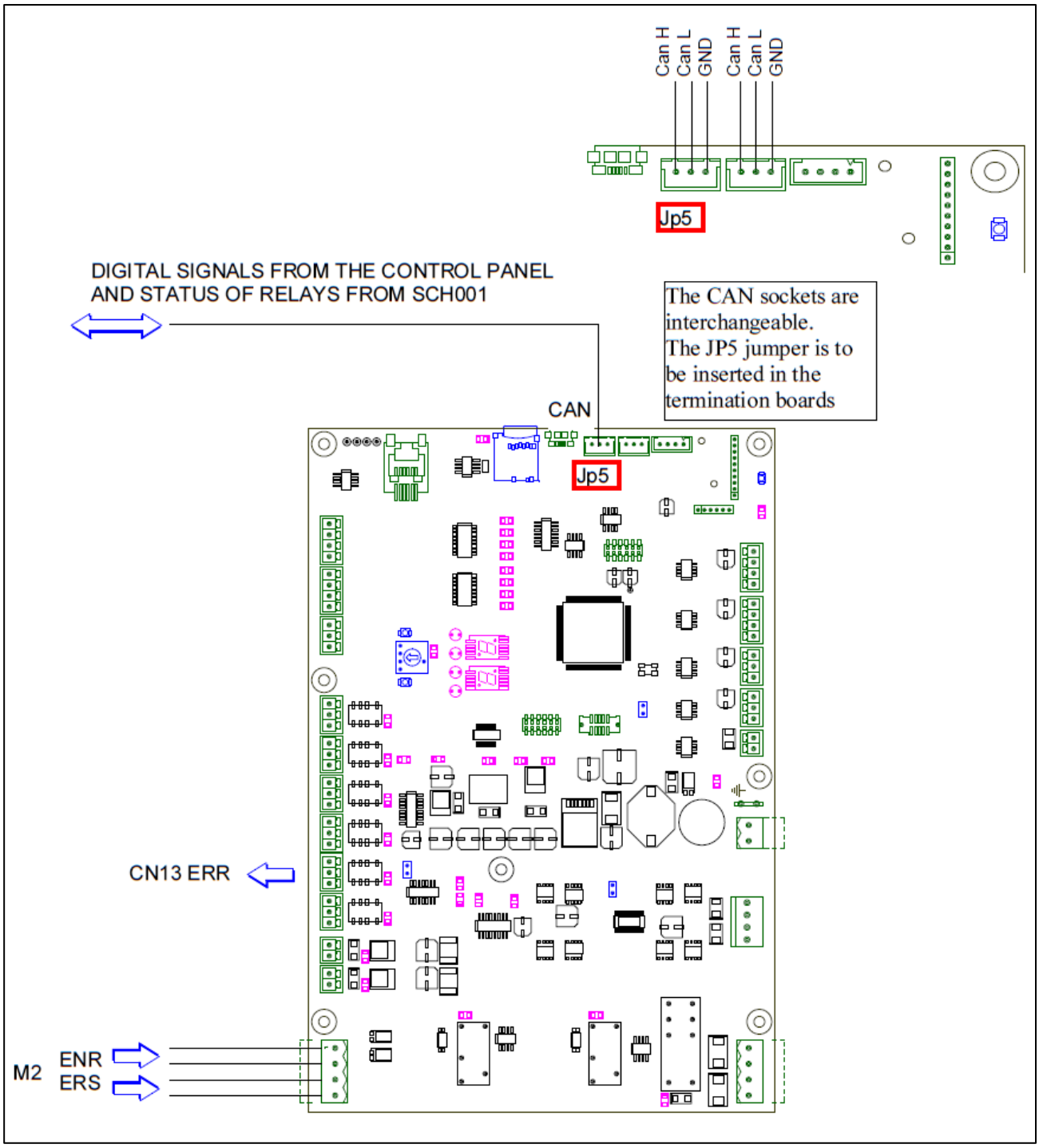
Here below are detailed the CAN cable specifications:



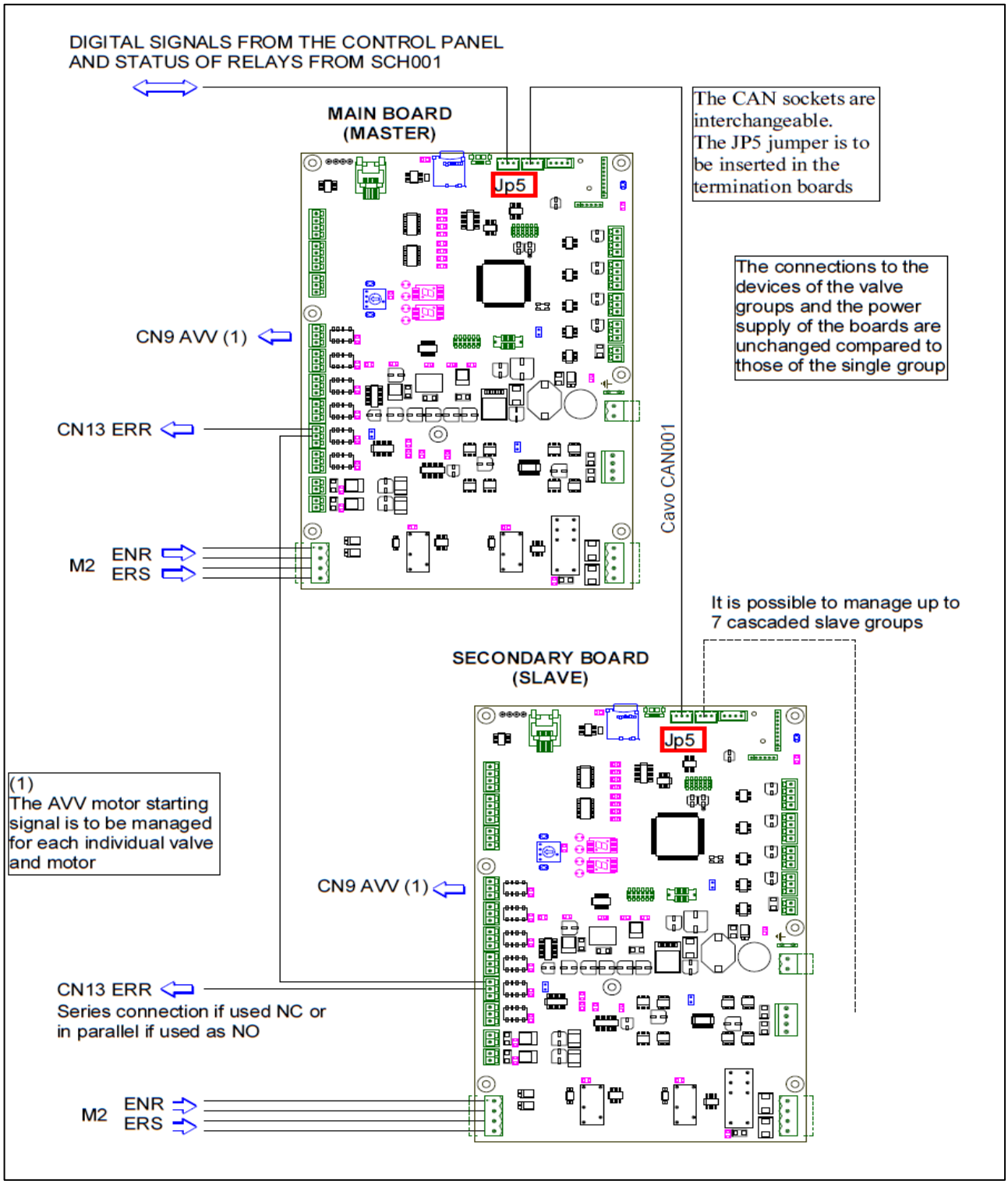
Cable outer sheath color	= grey
Cable type	= unshielded multipolar - 3 conductors with 0.22 mm^2 section
Type crimp contact	= JST series XH – type SXH-001T-P0.6 (cod. RS: 123-5947)
Connector type	= JST series XH – type XHP-3 (cod. RS: 820-1614)
Pinout	= see colors and data shown in the figure

HEVOS makes available upon request a 3meter length cable.

6 SCH001 BOARD CONNECTION



7 SCH001 BOARD CONNECTION – MULTI-VALVE SYSTEM



8 EXAMPLES of CAN COMMUNICATION MESSAGES BOARD – CONTROL PANEL

A single-valve system is considered and for simplicity CAN messages of the control panel with other components are not reported. The encoding of the messages is **hexadecimal**, according to the criteria set out in par. 3.

In the example:

0x100 = $(256)_{10}$ - P112=256 beginning of CAN communications band (Offset)

0x101 = $(256+1=257)_{10}$ - valve board node address (MASTER)

0x131 = $(256+49=305)_{10}$ - slave node address

CAN BUS Analyzer

BASE COMMUNICATION PROTOCOL

TIME (s) | |Address|Bit| Message

1064,8252;RX;0x131;2;0x61;0x02;;;;;;;
1064,8262;RX;0x101;4;0x68;0x00;0x06;0x81;;;;;
1064,9252;RX;0x131;2;0x61;0x02;;;;;;;
1064,9262;RX;0x101;4;0x68;0x00;0x06;0x81;;;;;
1065,0252;RX;0x131;2;0x61;0x02;;;;;;;
1065,0262;RX;0x101;4;0x68;0x00;0x06;0x81;;;;;
1065,1252;RX;0x131;2;0x61;0x02;;;;;;;
1065,1262;RX;0x101;4;0x68;0x00;0x06;0x81;;;;;
1065,2252;RX;0x131;2;0x61;0x02;;;;;;;
1065,2262;RX;0x101;4;0x68;0x00;0x06;0x81;;;;;
1065,3252;RX;0x131;2;0x61;0x02;;;;;;;
1065,3262;RX;0x101;4;0x68;0x00;0x06;0x81;;;;;
1065,4252;RX;0x131;2;0x61;0x02;;;;;;;
…..
1074,4193;RX;0x131;2;0x61;0x02;;;;;;;
1074,4202;RX;0x101;4;0x68;0x04;0x06;0x81;;;;;
1074,5192;RX;0x131;2;0x61;0x02;;;;;;;
1074,5202;RX;0x101;4;0x68;0x04;0x06;0x81;;;;;
1074,6192;RX;0x131;2;0x61;0x02;;;;;;;
1074,6202;RX;0x101;4;0x68;0x04;0x06;0x81;;;;;
1074,7192;RX;0x131;2;0x61;0x02;;;;;;;
1074,7202;RX;0x101;4;0x68;0x06;0x06;0x05;;;;;
1074,8212;RX;0x131;2;0x61;0x02;;;;;;;
1074,8212;RX;0x101;4;0x68;0x06;0x06;0x05;;;;;
1074,9212;RX;0x131;2;0x61;0x02;;;;;;;
1074,9212;RX;0x101;4;0x68;0x06;0x06;0x05;;;;;
….
1076,3202;RX;0x131;2;0x61;0x02;;;;;;;
1076,3212;RX;0x101;4;0x68;0x06;0x06;0x05;;;;;
1076,4202;RX;0x131;2;0x61;0x42;;;;;;;
1076,4203;RX;0x101;4;0x68;0x06;0x06;0x05;;;;;
1076,5202;RX;0x131;2;0x61;0x42;;;;;;;
….

EXTENDED COMMUNICATION PROTOCOL (8bit)

2000,4193;RX;0x131;8;0x61;0x02;0;0;0;0;0;0;
2000,4202;RX;0x101;8;0x68;0x06;0x05;0x81;0x65;0;0;0;
2000,5192;RX;0x131;8;0x61;0x02;0x65;0x5A;0;0;0;0;
2000,5202;RX;0x101;8;0x68;0x06;0x05;0x81;0x65;0;0;0;
2000,6192;RX;0x131;8;0x61;0x02;0x65;0x5A;0;0;0;0;
2000,6202;RX;0x101;8;0x68;0x06;0x05;0x81;0x65;0;0;0;
2000,7192;RX;0x131;8;0x61;0x02;0x65;0x5A;0;0;0;0;
2000,8212;RX;0x101;8;0x68;0x06;0x05;0x81;0x65;0;0;0;
2000,8212;RX;0x131;8;0x61;0x02;0x65;0x5A;0;0;0;0;
2000,9212;RX;0x101;8;0x68;0x06;0x05;0x81;0x65;0;0;0;

Board → Panel Message (ID 0x131)

Panel → Board Message (ID 0x101)

Transmission of messages from the valve board (MASTER) to the panel every 0.1s.

Number of components the message (Bytes)

0x61 Byte of start transmission from Board. Message:
0x02=$(0000\ 0010)_2$ →RDY

0x68 Byte of start transmission from Panel. Message:
0x04=$(0000\ 0100)_2$ → HSP High Speed
0x06=$(0000\ 0110)_2$ =$(6)_{10}$ → current floor 6
0x81=$(1000\ 0001)_2$ =$(129)_{10}$ → destination floor [the lack of a destination floor is coded in the example control panel with the value $(129)_{10}$]

0x68 Byte of start transmission from Panel. Message:
0x06=$(0000\ 0110)_2$ → DWN+HSP command
0x06=$(0000\ 0110)_2$ =$(6)_{10}$ → Current floor 6
0x05=$(0000\ 0101)_2$ =$(5)_{10}$ → Destination floor 5

0x61 Byte of start transmission from Board. Message:
0x42=$(0100\ 0010)_2$ =$(66)_{10}$ →PNP2+RDY
(PNP2= flow start)

0x68 Byte of start transmission from Panel. Message:
0x65=$(0110\ 0101)_2$ =$(101)_{10}$ →
Request to read parameter P101 (Piston diameter)

0x61 Byte of start transmission from Board. Message:
0x02=$(0000\ 0010)_2$ →RDY
0x65=$(0110\ 0101)_2$ =$(101)_{10}$ → Parameter requested
0x5A=$(0101\ 1010)_2$ =$(90)_{10}$ → Piston diameter value

Reaction to Continuous request to read the same parameter: request aborted (0xFF;0xFF;0xFF;0xFF) due to time limit exceeded (>1sec)

575,4429;RX;0x101;8;0x68;0x00;0x00;0x81;0x65;0x00;0x00;0x00;
575,5419;RX;0x131;8;0x61;0x02;0x65;0x00;0xFF;0xFF;0xFF;0xFF;
575,5429;RX;0x101;8;0x68;0x00;0x00;0x81;0x65;0x00;0x00;0x00;
575,6419;RX;0x131;8;0x61;0x02;0x65;0x00;0xFF;0xFF;0xFF;0xFF;
575,6429;RX;0x101;8;0x68;0x00;0x00;0x81;0x65;0x00;0x00;0x00;
575,7419;RX;0x131;8;0x61;0x02;0x65;0x00;0xFF;0xFF;0xFF;0xFF;
575,7429;RX;0x101;8;0x68;0x00;0x00;0x81;0x66;0x00;0x00;0x00;
575,8419;RX;0x131;8;0x61;0x02;0x65;0x00;0xFF;0xFF;0xFF;0xFF;
575,8429;RX;0x101;8;0x68;0x00;0x00;0x81;0x66;0x00;0x00;0x00;
575,9419;RX;0x131;8;0x61;0x02;0x66;0x00;0x78;0x00;0x00;0x00;
575,9429;RX;0x101;8;0x68;0x00;0x00;0x81;0x66;0x00;0x00;0x00;
576,0410;RX;0x131;8;0x61;0x02;0x66;0x00;0x78;0x00;0x00;0x00;
576,0429;RX;0x101;8;0x68;0x00;0x00;0x81;0x66;0x00;0x00;0x00;
576,1410;RX;0x131;8;0x61;0x02;0x66;0x00;0x78;0x00;0x00;0x00;
576,1420;RX;0x101;8;0x68;0x00;0x00;0x81;0x66;0x00;0x00;0x00;
576,2409;RX;0x131;8;0x61;0x02;0x66;0x00;0x78;0x00;0x00;0x00;
576,2419;RX;0x101;8;0x68;0x00;0x00;0x81;0x66;0x00;0x00;0x00;
576,3409;RX;0x131;8;0x61;0x02;0x66;0x00;0x78;0x00;0x00;0x00;
576,3419;RX;0x101;8;0x68;0x00;0x00;0x81;0x66;0x00;0x00;0x00;
576,4409;RX;0x131;8;0x61;0x02;0x66;0x00;0x78;0x00;0x00;0x00;
576,4419;RX;0x101;8;0x68;0x00;0x00;0x81;0x66;0x00;0x00;0x00;
576,5409;RX;0x131;8;0x61;0x02;0x66;0x00;0x78;0x00;0x00;0x00;
576,5419;RX;0x101;8;0x68;0x00;0x00;0x81;0x66;0x00;0x00;0x00;
576,6409;RX;0x131;8;0x61;0x02;0x66;0x00;0x78;0x00;0x00;0x00;
576,6419;RX;0x101;8;0x68;0x00;0x00;0x81;0x66;0x00;0x00;0x00;
576,7409;RX;0x131;8;0x61;0x02;0x66;0x00;0x78;0x00;0x00;0x00;
576,7419;RX;0x101;8;0x68;0x00;0x00;0x81;0x66;0x00;0x00;0x00;
576,8409;RX;0x131;8;0x61;0x02;0x66;0x00;0xFF;0xFF;0xFF;0xFF;
576,8419;RX;0x101;8;0x68;0x00;0x00;0x81;0x66;0x00;0x00;0x00;
576,9409;RX;0x131;8;0x61;0x02;0x66;0x00;0xFF;0xFF;0xFF;0xFF;
576,9419;RX;0x101;8;0x68;0x00;0x00;0x81;0x66;0x00;0x00;0x00;
577,0409;RX;0x131;8;0x61;0x02;0x66;0x00;0xFF;0xFF;0xFF;0xFF;
577,0419;RX;0x101;8;0x68;0x00;0x00;0x81;0x66;0x00;0x00;0x00;
577,1409;RX;0x131;8;0x61;0x02;0x66;0x00;0xFF;0xFF;0xFF;0xFF;
577,1419;RX;0x101;8;0x68;0x00;0x00;0x81;0x66;0x00;0x00;0x00;
577,2409;RX;0x131;8;0x61;0x02;0x66;0x00;0xFF;0xFF;0xFF;0xFF;
577,2419;RX;0x101;8;0x68;0x00;0x00;0x81;0x66;0x00;0x00;0x00;
577,3409;RX;0x131;8;0x61;0x02;0x66;0x00;0xFF;0xFF;0xFF;0xFF;
577,3419;RX;0x101;8;0x68;0x00;0x00;0x81;0x66;0x00;0x00;0x00;
577,4409;RX;0x131;8;0x61;0x02;0x66;0x00;0xFF;0xFF;0xFF;0xFF;

>1sec

Request from the Panel. Message:
0x65=$(0110\ 0101)_2$ =$(101)_{10}$→ piston diam.

Answer from the Board. Message:
0x65=$(0110\ 0101)_2$ =$(101)_{10}$→ piston diam.
0xFFFFFFFF→time limit exceeded

New Request from the Panel. Message:
0x66=$(0110\ 0110)_2$ =$(101)_{10}$→ Pump flow

Last Answer from the Board to previous request. Message:
0x65=$(0110\ 0101)_2$ =$(101)_{10}$→ piston diam.
0xFFFFFFFF→time limit exceeded

Answer from the Board. Message:
(First answer to 0x66)
0x66=$(0110\ 0110)_2$ =$(101)_{10}$→ Pump flow
0x78=$(0111\ 1000)_2$ =$(120)_{10}$→ flow value

Answer from the Board. Message:
0x66=$(0110\ 0110)_2$ =$(102)_{10}$→ Pump flow
0xFFFFFFFF→time limit exceeded (1sec from the first request)