SHS
Software Hardware Systems

ACP&D Limited

Manual

STEPPING MOTOR DRIVES
SERIE

STAR 2000

Mod.

APSx-x-00/D38
APSx-x-00/S47
APSx-x-00/S51

RELEASE
80FW3E – 80FW4C –80FW4E

Company Quality Assurance conforming

SINCERT

ISO 9001
BUREAU VERITAS
Certification

N° 167514

INDEX:

1. TECHNICAL DATA

1.1 POWER SUPPLY/OUTPUT CURRENT

SIZE		APS1	APS2	APS3	APS4	APS5
Vdc nom.	**[V]**	From 40 to 80	From 40 to 80	From 40 to 80	From 80 to 140	From 160 to 180
Vdc max.	**[V]**	90	90	90	155	195
Vdc min.	**[V]**	30	30	30	75	155
I max.	**[A]**	4	6	10	12	10
I min.	**[A]**	0.4	1	1	1	1
I step	**[A]**	0.2	0.5	0.5	0.5	0.5
Operating temperature	**[°C]**	0-55	0-55	0-55	0-55	0-55

PARAMETERS DESCRIPTION

VDC nom.: Rated value of voltage by which the drive can be powered.

VDC max.: Maximum voltage at which the drive can operate. Over this limit, the protection of maximum voltage inhibits the drive.

VDC min.: Minimum voltage at which the drive can operate. Under this limit, the protection of minimum voltage inhibits the drive.

I max.: Maximum value of phase current.

I min.: Minimum value of phase current.

I step: Spacing of the current values.

Operation temperature: For any current over 6 Amps, a forced ventilation is necessary.

1.2 SERIAL INTERFACE

Drives are supplied with RS485 (half duplex) / RS422 (full duplex) serial interface.
The communication protocol is on board and in the following described.

With the RS485/RS422 serial interface is possible to connect up to 32 drives, selected by address set through dip-switches.

1.3 INPUTS AND OUTPUTS

Inputs are PNP or NPN, outputs are PNP open collector optoisolated (10mA max).

INPUTS FEATURES:

INPUT VOLTAGE	IN1 HIGH SPEED INPUT	IN2, IN3, ENABLE INPUTS
LOW LEVEL	From 0 V to 8 V	From 0 V to 2.5 V
HIGH LEVEL	From 11 V to 30 V	From 4.6 V to 30 V

OUTPUTS FEATURES:

APSx-x-0x/D38	→ OUT1/OUT2 max current 10 mA
APSx-x-0x/S47	→ OUT1 max current 2 A, OUT2 max current 10mA
APSx-x-0x/S51	→ OUT1/OUT2 max current 2 A

1.4 PROTECTIONS

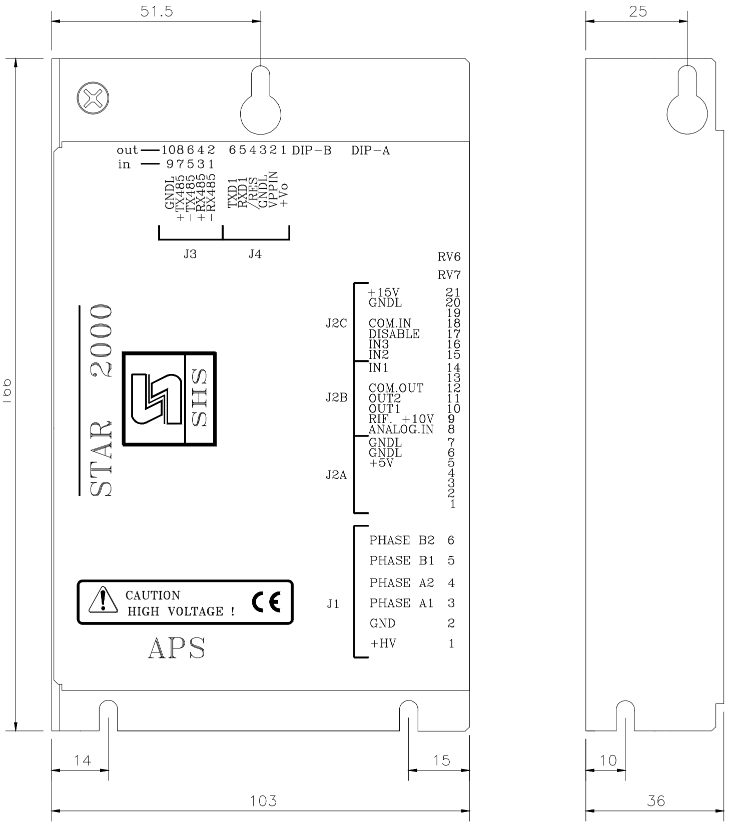
Drive is provided with protections against overtemperature, overvoltage, undervoltage, short-circuits among outputs and also among outputs and the positive power pole. If one of the mentioned conditions occurs, drive disables the power bridge and shows an error condition on the display.

'u'	- Power supply volts out of correct limits
't'	- Thermic protection event occurred
'c'	- Overcurrent protection event occurred
'd'	- Driver disable
'C'	- Paper broken (the stop sensor is not reached after relative setting quote)

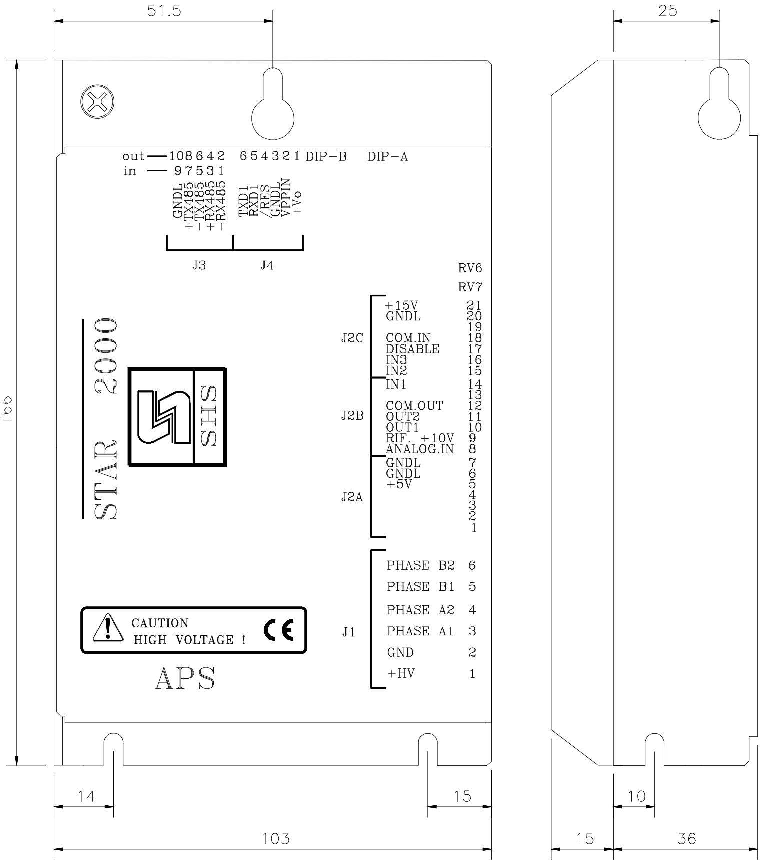
If drive is ready, display shows the letter 'r' (ready).

1.5 MECHANICAL DIMENSIONS

APS1 – APS2 – APS3:

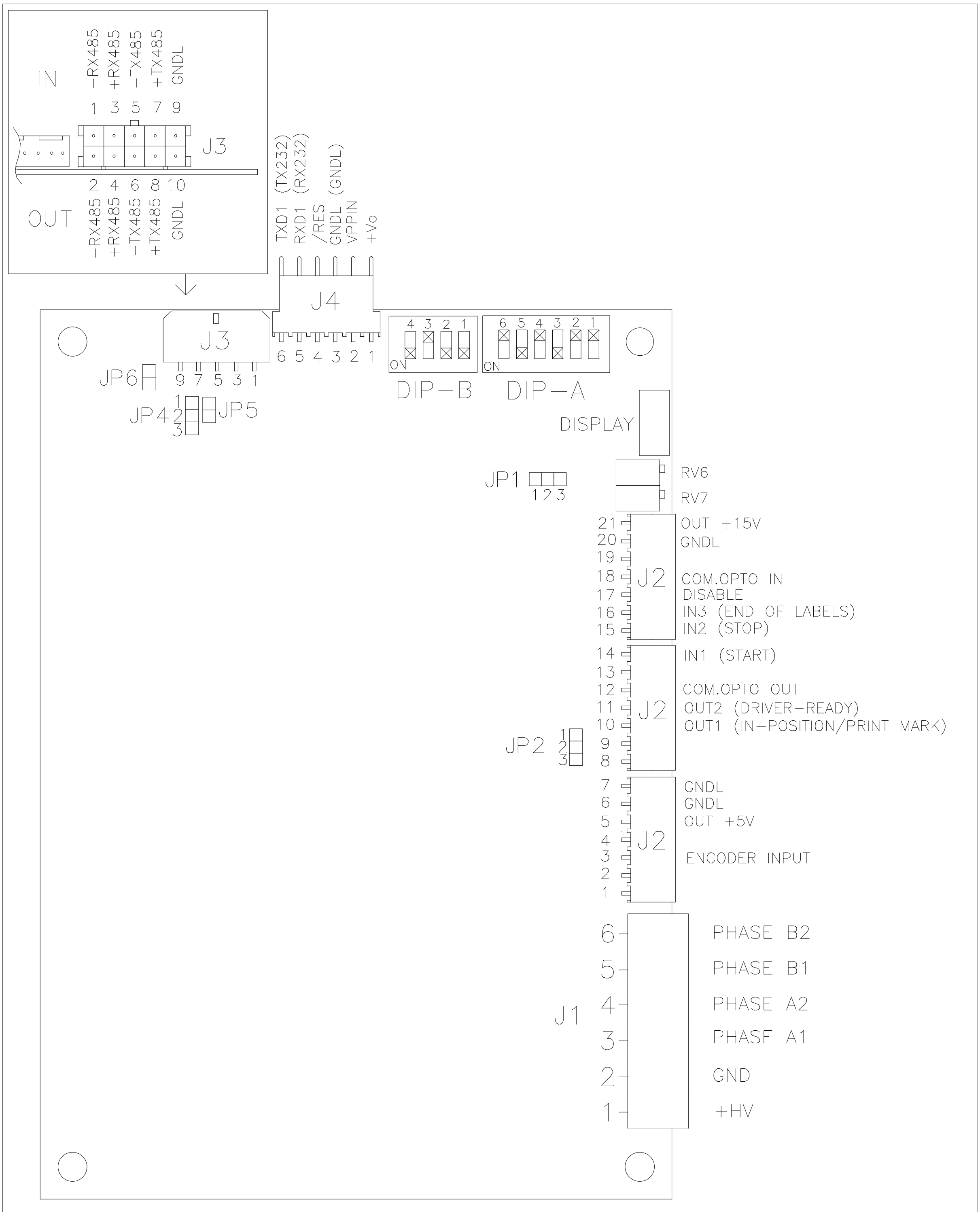


APS4 – APS5:



2. DRIVE CONNECTION

2.1 DRIVE LAYOUT



	S.H.S. S.r.l.		RESCALDINA (MI) ITALY		
TOLERANCES	TITLE	APS LAYOUT			
	DRAWN BY	VIGNATI		DATE	28.06.05
SCALE:	FILE NAME	S00-S47		REV	1.0

2.2 INSTALLATION NOTES:

WARNING

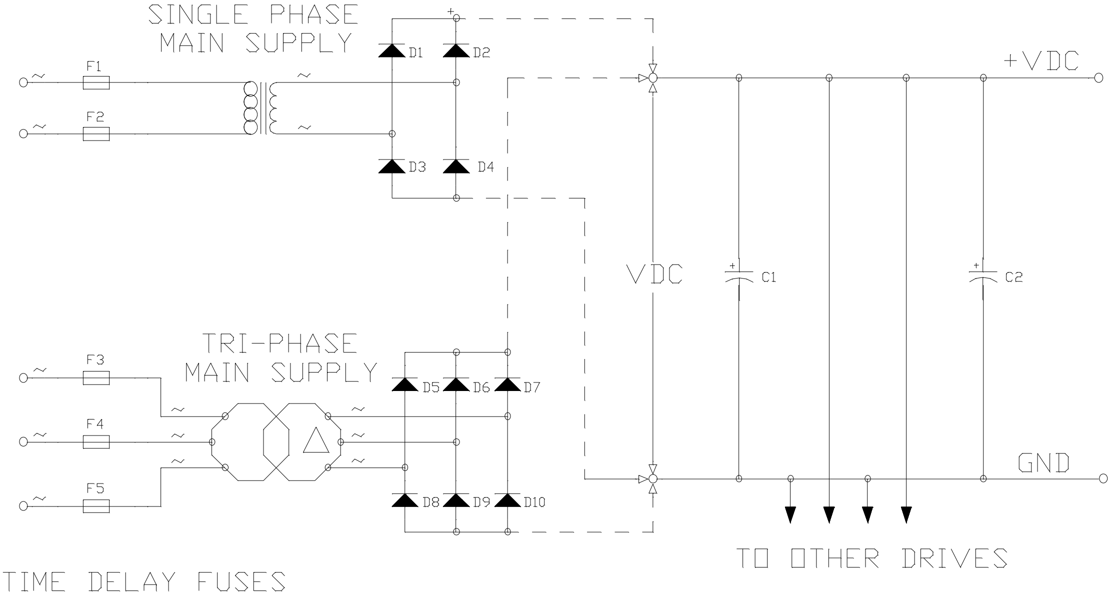
DANGER OF ELECTRICAL SHOCK . ONLY QUALIFIED INDIVIDUALS SHOULD WORK ON THIS EQUIPMENT. DISCONNECT ALL POWER BEFORE WORKING ON EQUIPMENT. DANGEROUS VOLTAGES MAY EXIST AFTER POWER IS REMOVED! CHECK DC BUS VOLTAGE OF DRIVES EACH TIME POWER IS REMOVED BEFORE WORKING ON EQUIPMENT.

2.3 POWER SUPPLY:

CONNECTION	SIGNAL	FUNCTION
J1 - Pin 1	+HV	+VDC power supply (see technical data)
J1 - Pin 2	GND	0 V power supply

The drive needs an external power supply that must be built as indicated in the FIG.1 otherwise use the TPS power supply (see TPS handbook).

FIG.1



F1..F5 TIME DELAY FUSES
(ACCORDING POWER CAPACITY OF TRASFORMER)

N.B.: IN CASE SHIELDED CABLE
IS USED, SHIELD MUST BE
CONNECTED TO GND.

To reduce EMI/RFI it is advisable:
Maximum wire reduction between power supply and drive, between drive and motor and, eventually, using shielded wires and insert the capacitor C1 (100nF 250 V, for switching applications)

The C2 capacitor value is 470µF 100V for APS1,APS2 and APS3, 160V for APS4 and 200V for APS5 for every Amps required.
i.e.: for 1 APS3/A (10A) use 4700µF 100V. For 1 APS4 (12A) use 4700 µF 160V. For 1 APS5 (10A) use 4700 µF 200V.

The transformer power is *P=Vac*(Inf(tot) + 1)*
Where *P* is VA power, *Vac* is secondary voltage in Volts and *Inf(tot)* is the sum of all nominal currents set in all the drive to be supplied.

2.4 MOTOR CONNECTION:

CONNECTION	SIGNAL	FUNCTION
J1 - Pin 3	PHASE A1	Phase A1 Step motor
J1 - Pin 4	PHASE A2	Phase A2 Step motor
J1 - Pin 5	PHASE B1	Phase B1 Step motor
J1 - Pin 6	PHASE B2	Phase B2 Step motor

2.5 INPUTS/OUTPUTS CONNECTIONS:

CONNECTION	SIGNAL	LABEL MACHINES
J2C - Pin 21	+15V	+15V output
J2C - Pin 20	GNDL	Logic signals GND
J2C - Pin 19	Unassigned	
J2C - Pin 18	COM.IN	Common inputs optocouplers
J2C - Pin 17	DISABLE	DISABLE input
J2C - Pin 16	IN3	END OF LABELS input
J2C - Pin 15	IN2	STOP input
J2B - Pin 14	IN1	START input
J2B - Pin 13	Unassigned	
J2B - Pin 12	COM.OUT	Common outputs optocouplers
J2B - Pin 11	OUT2	DRIVER READY output
J2B - Pin 10	OUT1	PRINT MARK output
J2B - Pin 9	RIF. +10V	Unassigned
J2B - Pin 8	ANALOG.IN	Unassigned
J2A - Pin 7	GNDL	Logic signals GND
J2A - Pin 6	GNDL	Logic signals GND
J2A - Pin 5	+5V	Unassigned
J2A - Pin 4	Unassigned	
J2A - Pin 3	Unassigned	ENCODER input (5-24V)
J2A - Pin 2	Unassigned	
J2A - Pin 1	Unassigned	

INPUTS:

SEGNAL	FUNCTION
ENCODER INPUT (J2-3)	ENCODER INPUT (NPN or LEVEL from 5V to 24V)
IN1 (J2-14)	START label erogation
IN2 (J2-15)	STOP
IN3 (J2-16)	END OF LABELS
DISABLE (J2-17)	It disables the power bridge.

OUTPUTS:

SEGNAL	FUNCTION
OUT1 (J2-10)	IN-POSITION output or PRINT MARK (see commands table): IN-POSITION: Motor stop : Out ON (low level) Motor run : Out OFF (high level) PRINT MARK: This output will active when motor stop for time setting with serial command
OUT2 (J2-11)	DRIVER-READY output: Drive ready : Out ON (low level) Drive fault : Out OFF (high level)

2.6 INPUTS/OUTPUTS DIAGRAM

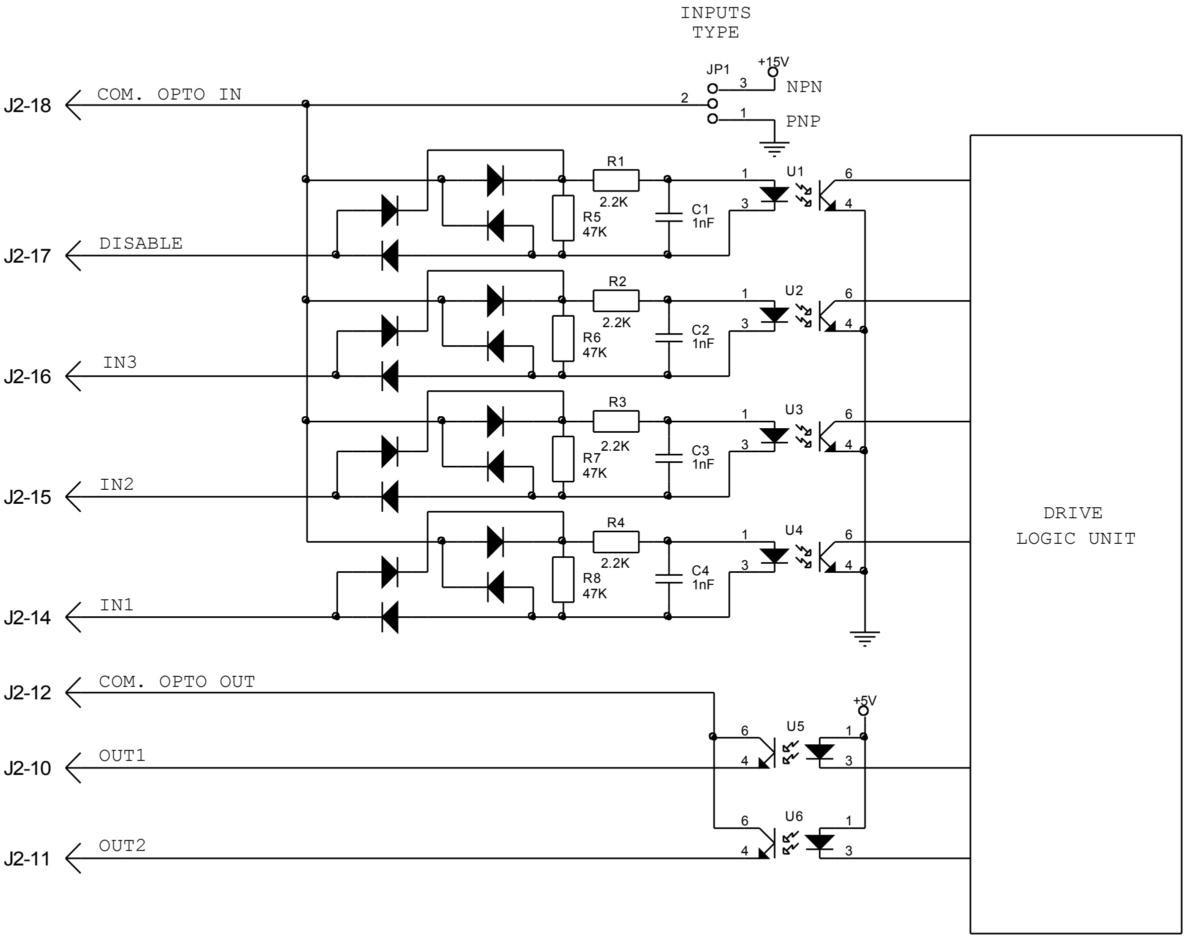


FIG.2

INPUT/OUTPUTS CONNECTION NOTES:

JP 1 - When inserted in 1-2 position, it associates the input common pole with GND of the drive (non-optoisolated PNP inputs)

JP 1 - When inserted in 2-3 position, it associates the input common pole with +15V of the drive (non-optoisolated NPN inputs)

JP 1 - When not inserted, it associates the input optoisolated. In this case, you must connect the GND of the external logic power supply to the connector J2 - pin 18 for PNP inputs, or the positive (+12VDC/ +24VDC) of external logic power supply for NPN inputs.

Outputs are always optoisolated. Therefore, an external powering (from 12VDC to 24VDC) must be connected to the common pole of the optoisolators (Pin 12, conncector J2)

2.7 RS422 SERIAL INTERFACE (RS485 full duplex):

For RS422 serial interface use J3 connector

CONNECTION	SIGNAL	FUNCTION
J3 - Pin 10	GNDL OUT	RS422 signals GND. Output for other drives
J3 - Pin 9	GNDL IN	RS422 signals GND
J3 - Pin 8	+TX485 OUT	RS422 +TX signal. Output for other drives
J3 - Pin 7	+TX485 IN	RS422 +TX signal
J3 - Pin 6	- TX485 OUT	RS422 –TX signal. Output for other drives
J3 - Pin 5	- TX485 IN	RS422 –TX signal
J3 - Pin 4	+RX485 OUT	RS422 +RX signal. Output for other drives
J3 - Pin 3	+RX485 IN	RS422 +RX signal
J3 - Pin 2	- RX485 OUT	RS422 +RX signal. Output for other drives
J3 - Pin 1	- RX485 IN	RS422 +RX signal

2.8 RS485 SERIAL INTERFACE (half duplex):

For RS485 serial interface use J3 connector

CONNECTION	SIGNAL	FUNCTION
J3 - Pin 10	GNDL OUT	RS485 signals GND. Output for other drives
J3 - Pin 9	GNDL IN	RS485 signals GND
J3 - Pin 8	+TX485 OUT	RS485 +TX/+RX signals. Output for other drives
J3 - Pin 7	+TX485 IN	RS485 +TX/+RX signals
J3 - Pin 6	- TX485 OUT	RS485 –TX/–RX signals. Output for other drives
J3 - Pin 5	- TX485 IN	RS485 –TX/–RX signals
J3 - Pin 4	+RX485 OUT	Not use
J3 - Pin 3	+RX485 IN	Not use
J3 - Pin 2	- RX485 OUT	Not use
J3 - Pin 1	- RX485 IN	Not use

2.9 RS485/RS422 SERIAL CONNECTION DIAGRAM:

Connection diagram on J3 connector

RS485 FULL DUPLEX CONNECTION

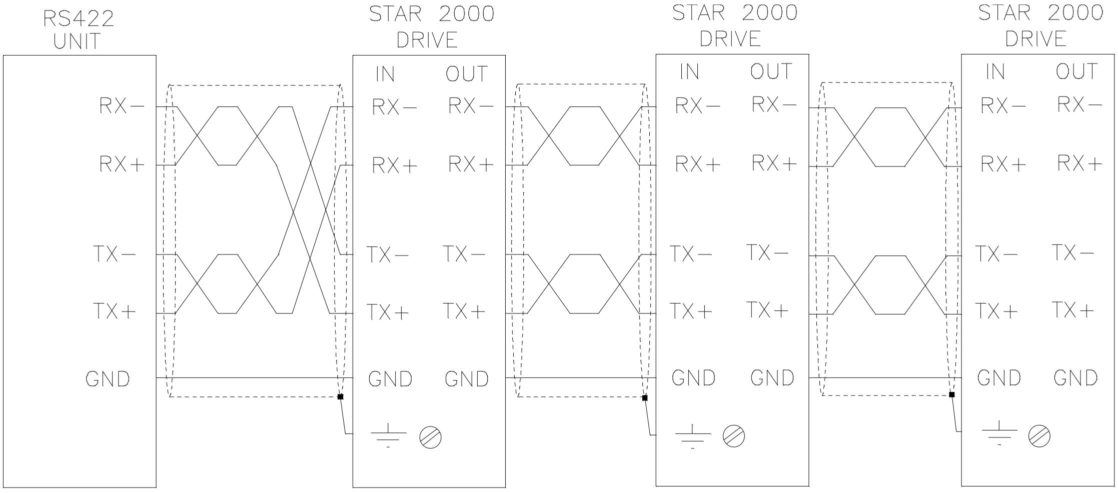


FIG.4

To use RS485 full duplex (RS422) set JP4 jumper between 1-2 position (factory default), see layout par.2.1 and FIG.4

RS485 HALF DUPLEX CONNECTION

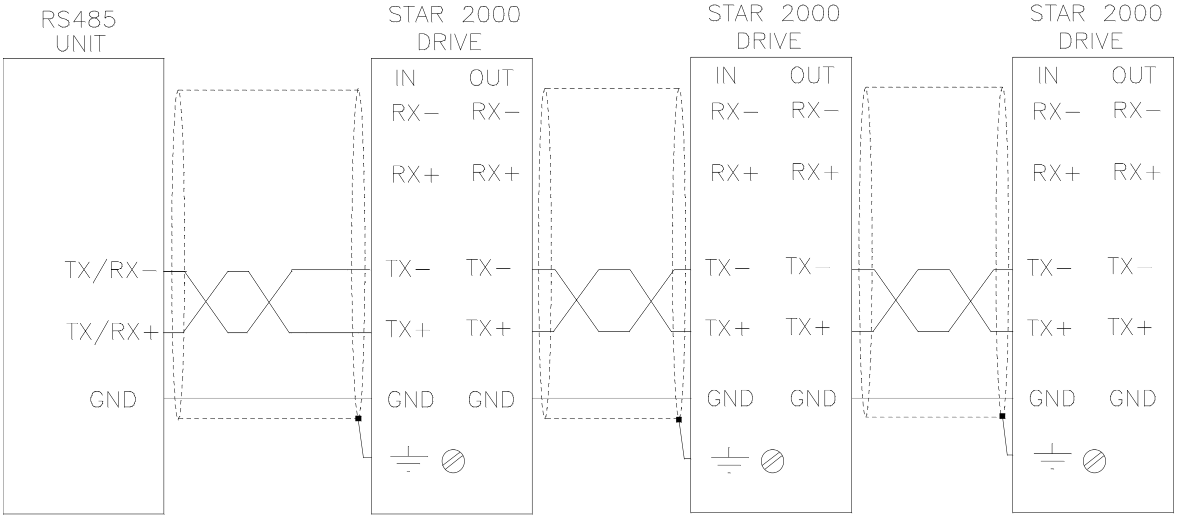


FIG.5

To use RS485 half duplex set JP4 jumper between 2-3 position, see layout par.2.1 and FIG.5

3. JUMPERS SETTINGS

JP 1 - When inserted in 1-2 position, it associates the input common pole with GND of the drive (non-optoisolated PNP inputs)

JP 1 - When inserted in 2-3 position, it associates the input common pole with +15V of the drive (non-optoisolated NPN inputs)

JP 1 - When not inserted, it associates the input optoisolated. In this case, you must connect the GND of the external logic power supply to the connector J2 - pin 18 for PNP inputs, or the positive (+12VDC/ +24VDC) of external logic power supply for NPN inputs.

Factory configuration: JP1 inserted in 1-2 position; JP2 not inserted; JP4 inserted in 1-2 position; JP5, JP6 not inserted.

4. SETTINGS AND SERIAL PROTOCOL

You can controle this drive with RS422/RS45 with your PLC or using MCU03/ET2000:

Select mode with DIPB1:

DIPB 1 → OFF : ET2000
ON : MCU03/PLC

THIS SELECTION MUST BE DONE BEFORE POWERING THE DRIVE.

4.1 CURRENT REGULATION:

For setting current proceed as follows:

- Set dip-switch B-4 to ON (current regulation mode).

- Turn RV6 trimmer until display shows the required current (CW to increase).

- Regulation field: from 1A to 10 A at steps of 0.5A for APS3/APS5 drive.
  Regulation field: from 1A to 12 A at steps of 0.5A for APS4 drive.
  Regulation field: from 1A to 6 A at steps of 0.5A for APS2 drive.
  Regulation field: from 0.4A to 4 A at steps of 0.2A for APS1 drive.

- Set dip-switch B-4 to OFF (Run mode).

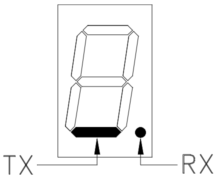
Table for setting current values and relating values shown on the display of drive APS:

DISPLAYED VALUE	SETTING CURRENT APS 1	SETTING CURRENT APS 2	SETTING CURRENT APS4	SETTING CURRENT APS 3 / APS5
1	0.4 A	1.0 A	1.0 A	1.0 A
1.	0.6 A	1.5 A	1.5 A	1.5 A
2	0.8 A	2.0 A	2.0 A	2.0 A
2.	1.0 A	2.5 A	2.5 A	2.5 A
3	1.2 A	3.0 A	3.0 A	3.0 A
3.	1.4 A	3.5 A	3.5 A	3.5 A
4	1.6 A	4.0 A	4.0 A	4.0 A
4.	1.8 A	4.5 A	4.5 A	4.5 A
5	2.0 A	5.0 A	5.0 A	5.0 A
5.	2.2 A	5.5 A	5.5 A	5.5 A
6	2.4 A	6.0 A	6.0 A	6.0 A
6.	2.6 A	-	6.5 A	6.5 A
7	2.8 A	-	7.0A	7.0A
7.	3.0 A	-	7.5 A	7.5 A
8	3.2 A	-	8.0 A	8.0 A
8.	3.4 A	-	8.5 A	8.5 A
9	3.6 A	-	9.0 A	9.0 A
9.	3.8 A	-	9.5 A	9.5 A
0	4.0 A	-	10.0 A	10.0 A
0.	-	-	10.5 A	-
a	-	-	11.0 A	-
a.	-	-	11.5 A	-
b	-	-	12.0 A	-

NOTE: ADJUST CURRENT WHEN MOTOR IS HOLDING.

4.2 SERIAL INTERFACE MONITOR:

On the display we can see the correct working of serial interface. The decimal point indicate when a command is received (RX), the nearest horizontal segment indicate when a answer is sended (TX).



4.3 SWITCHES AND COMMUNICATION INTERFACE SETTING:

DIP SWITCH A		
DIP	**ON**	**OFF**
2..6	Serial address (see table)	
1	Special function piston (only for APSx-x-0x/S51)	Default

DIP SWITCH B		
DIP	**ON**	**OFF**
4	Current regulation mode	RUN mode
3	Not used	Not used
2	Not used	Not used
1	Use PLC or MCU03	use ET2000

Communication parameters:

BAUD RATE	: 9600 (DIP A-1 ON) or 19200 (DIP A-1 OFF)
PARITY	: NO PARITY
DATA BITS	: 8
BIT STOP	: 1

DRIVE IDENTIFICATION ADDRESS SETTINGS ON 'A' DIP-SWITCH

DIPA-2 (BIT4)	**DIPA-3 (BIT3)**	**DIPA-4 (BIT2)**	**DIPA-5 (BIT1)**	**DIPA-6 (BIT0)**	**ADDRESS**
OFF	OFF	OFF	OFF	OFF	0
OFF	OFF	OFF	OFF	**ON**	1
OFF	OFF	OFF	**ON**	OFF	2
OFF	OFF	OFF	**ON**	**ON**	3
OFF	OFF	**ON**	OFF	OFF	4
OFF	OFF	**ON**	OFF	**ON**	5
OFF	OFF	**ON**	**ON**	OFF	6
OFF	OFF	**ON**	**ON**	**ON**	7
OFF	**ON**	OFF	OFF	OFF	8
OFF	**ON**	OFF	OFF	**ON**	9
OFF	**ON**	OFF	**ON**	OFF	10
OFF	**ON**	OFF	**ON**	**ON**	11
OFF	**ON**	**ON**	OFF	OFF	12
OFF	**ON**	**ON**	OFF	**ON**	13
OFF	**ON**	**ON**	**ON**	OFF	14
OFF	**ON**	**ON**	**ON**	**ON**	15
ON	OFF	OFF	OFF	OFF	16
ON	OFF	OFF	OFF	**ON**	17
ON	OFF	OFF	**ON**	OFF	18
ON	OFF	OFF	**ON**	**ON**	19
ON	OFF	**ON**	OFF	OFF	20
ON	OFF	**ON**	OFF	**ON**	21
ON	OFF	**ON**	**ON**	OFF	22
ON	OFF	**ON**	**ON**	**ON**	23
ON	**ON**	OFF	OFF	OFF	24
ON	**ON**	OFF	OFF	**ON**	25
ON	**ON**	OFF	**ON**	OFF	26
ON	**ON**	OFF	**ON**	**ON**	27
ON	**ON**	**ON**	OFF	OFF	28
ON	**ON**	**ON**	OFF	**ON**	29
ON	**ON**	**ON**	**ON**	OFF	30
ON	**ON**	**ON**	**ON**	**ON**	31

NOTE: If several drives are connected on RS485 serial line, make sure that all drives have a different address.

4.4 TRANSMISSION TIMING OF SERIAL COMMANDS:

Following instructions must be followed by sending any serial command to the drive:

COMMANDS WITH ANSWER:
- Send command to drive
- Wait for answer from drive
- Send next command
- Wait for answer from drive

COMMANDS WITHOUT ANSWER:
- Send command to drive
- Delay of at least 5 ms (for command execution)
- Send next command
- Delay of at least 5 ms (for command execution)

Sometime the answer of the drive is too fast for the control (PC or PLC) in this case it’s necessary to setting an “answer delay” command. (for example 5ms)

4.5 COMMUNICATION PROTOCOL

Systems can use a single drive or several drives, which are connected in multidrop to RS485 serial line (full duplex).
Commands can be sent either to a definite drive (by specifying its address in the string command) or to all drives.
In the first case drive will answer to the command; in the second case no reply will be given.

4.5.1 SINGLE ADDDRESS COMMAND:

DATA TO BE SENT TO THE DRIVE:
Commands succession to be sent to the drive must respect following structure:

<table>
<tr><td><i>Byte_start</i></td><td colspan="8"><i>byte_nbyte_address</i></td><td><i>byte command</i></td><td><i>[byte_par0]</i></td><td><i>[byte_par1]</i></td><td><i>byte_checksum</i></td></tr>
<tr><td>BYTE [0xFC]</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>BYTE</td><td>BYTE</td><td>BYTE</td><td>BYTE</td></tr>
<tr><td></td><td colspan="3"><i>nbyte</i></td><td colspan="5"><i>drive_address</i></td><td></td><td></td><td></td><td></td></tr>
</table>

byte_start : 0xFC. This byte means that a command will be sent to one drive only,

byte_nbyte_address : This byte contains two indications:
- *drive_address* The first 5 bits (from bit0 to bit4) contain the drive address (from 0 to 31).
- *nbyte* The following 3 bits (from bit5 to bit7) contain the bytes numbers which follow *byte_nbyte_address* before sending the *byte_checksum*.

byte_command : This byte represents the command (see commands protocol).

byte_par0, byte_par1 : The bytes, which follow the byte_command, represent the sending command parameters.

byte_checksum : This byte must be calculated by the user as complement of the less significant byte resulting from the sum of all sending bytes (including the byte_start), in order to have, as final result, one byte only. The function of this byte is to verify the correct transmission command (see example in paragraph 6.5.5).

DRIVE ANSWER:
Wrong or not foreseen command transmission, drive will answer byte-nak (0x15).
Correct command transmission, drive will answer *byte_ack* (0x06) and it will be followed by an answer bytes series as forseen in the above mentioned format.

EXAMPLES:
- To send RESET command to drive 0:
  Command string: 0xFC 0x20 0x01 0xE2 drive answer: 0x06

- To send RESET command to drive 1:
  Command string: 0xFC 0x21 0x01 0xE2 drive answer: 0x06

4.5.2 MULTI ADDRESSES COMMAND:

DATA TO BE SENT TO THE DRIVES:
Commands succession to be sent to the drive must respect following structure:

<table>
<tr><td><i>byte_start</i></td><td colspan="8"><i>byte_nbyte_address</i></td><td><i>byte_multiaddress</i></td><td><i>byte command</i></td><td><i>[byte_par0]</i></td><td><i>byte_address1</i></td></tr>
<tr><td>BYTE [0xFC]</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>BYTE [0xA5]</td><td>BYTE</td><td>BYTE</td><td>BYTE</td></tr>
<tr><td></td><td colspan="3"><i>nbyte</i></td><td colspan="5"><i>drive_address</i></td><td></td><td></td><td></td><td></td></tr>
</table>

[byte_address2]	*[byte_address3]*	*[byte_address4]*	*byte_checksum*
BYTE	BYTE	BYTE	BYTE

byte_start : 0xFC. This byte means that a command will be sent.

byte_nbyte_address : This byte contains two information:
- *drive_address* In the first 5 bytes enter no. 31 address
- *nbyte* The following 3 bytes (from 5 byte to 7 byte) show the bytes numbers which will follow *byte_nbyte_address* before sending *byte_checksum*.

byte_multiaddress : 0xA5. This byte indicates that command is addressed to several drives, whose address will be specified in the following bytes.

byte_command : This byte represents the command (see commands protocol).

byte_par0 : The byte, which follows *byte_command*, represents the sent command parameter (if necessary).

byte_address1..4 : Bytes, which follow *byte_command*, represent drives addresses to which command has been addressed. Four addresses can be sent if command foresees 1 parameter. Five addresses can be sent if command does not foresee any parameter.

byte_checksum : This byte must be calculated by the user as complement of the less significant byte resulting from the sum of all sending bytes (including *byte_start*), in order to have one byte only, as final result. Function of this byte is to verify the correctcommand transmission (see example in paragraph 6.5.5).

DRIVES ANSWER:
If command correct, it will be executed. If wrong, it will be not executed, by giving no answer.
Command being addressed to several drives, they cannot answer, otherwise an hardware conflict would be caused.

EXAMPLES:
- To send RESET command to drive 0,1,2,3:
  Command string: 0xFC 0xDF 0xA5 0x01 0x00 0x01 0x02 0x03 0x78 drive answer: none

- To send RESET command to drive 4,5:
  Command string: 0xFC 0x9F 0xA5 0x01 0x04 0x05 0xB5 drive answer: none

4.5.3 COMMAND ADDRESSED TO ALL DRIVES:

DATA TO BE SENT TO THE DRIVES:
Commands succession to be sent to the drives must respect following structure:

byte_start	*byte_switchall*	*byte_nbyte*	*byte command*	*[byte_par0]*	*[byte_par1]*	*byte_checksum*
BYTE [0xFC]	BYTE [0x00]	BYTE	BYTE	BYTE	BYTE	BYTE

byte_start : 0xFC. This byte means that a command will be sent.

byte_switchall : 0x00. This byte means that a command will be sent to all drives

byte_nbyte : This byte indicates the bytes numbers which will follow *byte-nbyte address* before sending *byte-checksum.*

byte_command : This byte represents the command (see commands protocol).

byte_par0, byte_par1 : Bytes, following *byte_command*, represent the sent command parameters.

byte_checksum : This byte must be calculated by the user as complement of the less significant byte resulting from the sum of all sending bytes (including *byte_start*), in order to have one byte only as final result.

The function of this byte is to verify the correct command transmission (see example in paragraph 6.5.5).

DRIVES ANSWER:
If command correct, it will be executed. If wrong, it will be not executed, by giving no answer.
Command being addressed to several drives, they cannot reply, otherwise an hardware conflict would be caused.

EXAMPLE:
- To send RESET command to all drives:
  Command string: 0xFC 0x00 0x01 0x01 0x01 drive answer: none

4.5.4 COMMANDS PROTOCOL

COMMAND	PARAMETERS	ANSWER	FUNCTION
0x01	None	*byte_ack*	DRIVE RESET: It stops the motor. It initializes speed and ramp to 0.
0x02	None	*byte_ack*	SOFTWARE START: By sending this command, motor is running according to the transmitted values (speed and ramp).
0x10	None	*byte_ack* + *byte_start*+ *byte_nbyte_address*+ 0x*NN*+*byte_chksum* *NN*=Software version	REQUEST FOR SOFTWARE VERSION
0x11	None	*byte_ack*	IMMEDIATE STOP: Motor slacks speed according to the preset ramp and then it stops.
0x12	None	*byte_ack+byte_start+ byte_nbyte_address +byte1+byte2+byte3+byte 4+byte_chksum*	READING PRESENT POSITION: By receiving command 0x13 drive show present motor condition with 4 bytes.
0x13	None	*byte_ack+byte_start+ byte_nbyte_address+ 0xNN+byte_chksum* NN=byte, in which the 4 less significant bits represent the inputs status (1=input activated), next 2 bits represent the outputs status (1=output activated), the last 2 bits are not utilized (always at 0)	READING OF INPUTS / OUTPUTS
0x14	None	*Byte_ack+Byte_start+ byte_nbyte_address+ 0xNN+Byte_chksum*	INQUARY FOR DRIVE TYPE: drive signals a number corresponding to the drive type
0x20	2 bytes, which indicate the minimum frequency (from 0 to 10000 Hz)	*Byte_ack*	SETTING OF MINIMUM FREQUENCY.
0x21	2 bytes, which indicate the maximum frequency (from 0 to 10000 Hz)	*byte_ack*	SETTING OF MAXIMUM FREQUENCY. NOTE: If motor is running, this parameter will be acquired to next motion command.
0x22	1 byte, which indicates the ramp inclination (from 0 to 255) expressed in ms * 10	*byte_ack*	SETTING OF RAMP INCLINATION NOTE: If motor is running, this parameter will be acquired to the next motion command.
0x23	4 bytes, which indicate the absolute motor position (expressed in 1/128 step)	*byte_ack*	SETTING OF HOME POSITION: Drive associates the entered value to home position of the motor

COMMAND	PARAMETERS	ANSWER	FUNCTION
0x26	1 byte, which indicates motor resolution. If the sent byte = 0 → full step mode = 1 → 1/2 step mode = 2 → 1/4 step mode = 3 → 1/8 step mode = 4 → 1/16 step mode	*byte_ack*	SETTING OF MOTOR RESOLUTION. NOTE: If motor is running, this parameter will be acquired to the next motion command.
0x27	1 byte, which indicates time and mode of current reduction. The first 6 bits indicate time after which current reduction (from 0 to 63) must occur on a time basis of 32 ms. The next 2 bits indicate the reduction mode: 00 – current 0 01 - no reduction 10 – reduction to 25% 11 – reduction to 50%	*byte_ack*	SETTING OF ELECTRIC CURRENT REDUCTION
0x28	1 byte, which indicates delayed answer of serial interface (from 0 to 255) expressed in µs * 512	*byte_ack*	SETTING ANSWER DELAY
0x29	1 byte: the 4 less significant bits indicate the input or the inputs, which must be enabled for STARTING (1 input enabled). The next 4 bits indicate the level of these inputs (1=active input at high level)	*byte_ack*	TRIGGER START (LOGIC AND): It defines the Input or the inputs and the respective Levels, which must be enabled for carrying out the START by an external command.
0x2A	1 byte: the 4 less significant bits indicate the input or the inputs, which must be enabled for STOPPING (1= enabled input). The next 4 bits indicate the level of these inputs (1=input active at high level)	*byte_ack*	TRIGGER STOP (LOGIC AND): It defines the input or the inputs and the respective levels, which must be enabled for carrying out the STOP by an external command.
0x2B	1 byte, indicates the level of the output 'in position': 0 >output motor is holding = 0 255 >output motor is holding= 1	*byte_ack*	'IN POSITION' OUTPUT LEVEL
0x2C	1 byte: the 4 less significant bits indicate the input or the inputs, which must be enabled for doing the HOME function (1=input enabled), the next 4 bits indicate the level of these inputs (1=input active at high level)	*byte_ack*	TRIGGER HOME (LOGIC AND): It defines the input or the inputs and the respective levels, which must be enabled for carrying out the HOME function by an external command.
0x30	4 bytes, wich indicate the absolute position to reach (expressed in 1/128 step). (values admitted: from -2147483647 to 2147483647)	*byte_ack*	ABSOLUTE POSITIONING (RELATIVE TO THE HOME POSITION)

COMMAND	PARAMETERS	ANSWER	FUNCTION
0x31	4 bytes, which indicate the carrying out positioning with respect to the present position of the motor (expressed in 1/128 step) (values admitted: from -2147483647 to 2147483647)	*byte_ack*	RELATIVE POSITIONING

0x32	1 byte, which indicates the rotation direction
	if = 0 CW
	if = 255 CCW
		byte_ack	INFINITE MOTION: When sending this command, motor is running at the speed entered in the specified rotation direction.
			NOTE: SEND THIS COMMAND ONLY WHEN THE MOTOR IS HOLDING.

0xA0	5 bytes: - the 4 less significant bits indicate the input or the inputs, which must be enabled for doing the ZERO AT FLIGHT (1=input enabled), the next 4 bits indicate the level of these inputs (1=input active at high level) - next 4 bytes: these indicate the positioning to be done in the same rotation direction since when the condition expressed in the first byte occurs (values admitted: 0 to 2147483647)	*byte_ack*	ZERO AT FLIGHT (LOGIC AND): It defines the input or the inputs and the respective levels, which must be enabled for carrying out zeroing of the value in the present motor position, when this condition occurs, and the value to be done on occasion of this condition.
0xA6	None	*byte_ack*	MOTION TO ZERO VALUE
0xA8	2 bytes, which indicate the current value (from 0 to 10000 mA)	*byte_ack*	CURRENT SETTING (ex. 10000 = 10A, 2000=2A). Entering a wrong value the answer will be byte_nack.
0xAA	4 bytes, which indicate the relative positioning with respect to the present position of the motor to be carried out at the next SOFTWARE START or TRIGGER START (expressed in 1/128 step). (values admitted: from -2147483647 to 2147483647)	*byte_ack*	SETTING OF RELATIVE VALUE (WITHOUT ANY POSITIONING)

COMMAND	PARAMETERS	ANSWER	FUNCTION
0xAB	None	*byte_ack+byte_start+ byte_nbyte_address+ Byte_status+byte_chksum* *Byte_status:* BIT0: 0 =motor is holding 1 =motor is running BIT1: 0=zero at flight not active or executed 1=zero at flight active BIT2: 0=drive ok 1=drive in protection BIT3..5: input status 1,2,3 (1=activated) BIT6-7: output status 1,2 (1=activated)	DRIVE STATUS
0xAC	None	*Byte_status:* BIT0: 0 =motor is holding 1 =motor is running BIT1: 0=zero at flight not active or executed 1=zero at flight active BIT2: 0=drive ok 1=drive in protection BIT3..5: input status 1,2,3 (1=activated) BIT6-7: output status 1,2 (1=activated)	DRIVE STATUS 1 BYTE: Only one byte Includes all information regarding the Drive status
0xAD	1 byte which shows the speed percentage (from 0 to 255)	*byte_ack*	PERCENTAGE SPEED CHANGE BY MOTOR MOTION: It allows to change speed by motor motion showing the relative percentage. Ex: 100%=no change, 50%=halve speed, 200%=double speed.
0xAE	4 bytes which show absolute position to join to present position (expressed in 1/128 step). (admitted value: from -2147483647 to 2147483647)	*byte_ack*	SETTING OF ABSOLUTE POSITION: It joins to the present position the command value.

COMMAND	PARAMETERS	ANSWER	FUNCTION
0xB5	1 byte which enable or disable STEP-OUT function on OUT1	*byte_ack*	STEP-OUT ON OUT1: 0 =OUT1→IN-POSITION 255 =OUT1→ STEP-OUT every front of this signal is a generated step.
0xB6	4 bytes which load absolute positioning without executing. (expressed in 1/128 step). (admitted value: from -2147483647 to 2147483647)	*byte_ack*	LOAD ABSOLUTE POSITIONING WITHOUT START
0xC0	1 bytes which shows trigger mode (from 0 to 255)	*byte_ack*	TRIGGER MODE SETTING: 0=Enable trigger one shot 1=Enable trigger forever FF= Enable trigger forever with infinite movement CW at first start FE= Enable trigger forever with infinite movement CCW at first start
0xC1	1 bytes which shows status byte mode (from 0 to 255)	*byte_status*	STATUS BYTE MODE: APS send on serial line one byte when inputs change. 0=Function disabled 1=Function enabled only for display ‘P’ when IN1 is active without send byte on serial line. FF=Function enabled only for display ‘P’ when IN1 is active and send byte on serial line.
0xC2	1 bytes which shows PRINT MARK activation time expressed in 1/128 step. (from 0 to 255)	*byte_ack*	PRINT MARK (OUT 1) ACTIVATION TIME: Out activation time when motor stop.
0xC3	2 bytes which shows start delay expressed in ms.	*byte_ack*	START DELAY (TIME): Time between IN1 activation and motor rotation.
0xC4	2 bytes which shows start delay expressed in motor steps (full step)	*byte_ack*	START DELAY (STEPS): Full steps between IN1 activation and motor rotation.
0xC7	2 bytes which shows encoder ratio	*byte_ack*	ENCODER STEPS/MOTOR STEPS RATIO: Set 1000 to execute one motor steps for any encoder steps, set 500 to execute one motor steps for any encoder steps. Set 0 to disable encoder.
0xC8	2 bytes which shows steps ramp	*byte_ack*	SET STEPS RAMP: set step to reach 10 KHz.
0xCB	1 bytes which shows encoder mode (from 0 to 255)	*byte_ack*	SET ENCODER MODE: 0= Reading encoder speed only when motor is stopped 1= Reading encoder speed and start/stop the motor according with encoder 2= Reading encoder speed but don’stop the motor when encoder stop

COMMAND	PARAMETERS	ANSWER	FUNCTION
0xAF	1 byte: the less 4 significant bits show the input where zero switch will be connected (1= enabled input), next 4 bits show the level of this input (1=activated input at high lavel)	*byte_ack*	AXIS ZEROING: It defines the input and relative level where zero switch will be connected and it carries out axis zeroing. The zeroing phase includes: CCW motor start; the search of zero switch with rotation at max speed; stop on zero switch; disengagement of zero switch at min. speed and zeroing of absolute position.
0xB0	1 byte: the less 4 significant bits show the input where limit switch will be connected. (1=enable input). Next 4 bits show the level of this input (1=activated input at high level). By using two switches (limit switch+ and limit switch-) they must be connected in series)	*byte_ack*	LIMIT SWITCH: It defines the input and the relative level where the limit switch will be connected. When the switch will be intercepted, it will cause the immediate motor stop and it allow only the opposite wise motion. The command activates this function until its disengagement through a new command with no specified input (Parameter=0).
0xB1	1 byte: the 4 less significant bits indicate the input or the inputs, which must be enabled for STOPPING (1= enabled input). The next 4 bits indicate the level of these inputs (1=input active at high level)	*byte_ack*	TRIGGER STOP (LOGIC OR): It defines the input or the inputs and the respective levels, which must be enabled for carrying out the STOP by an external command.
0xBD	2 bytes, which indicate the maximum frequency (from 0 to 10000 Hz)	*byte_ack*	SET MAXIMUM FREQUENCY WHEN MOTOR RUNNING
0xEE	1 byte (ammitted values: 0, 2)	*byte_ack*	LOW NOISE MODE: 2 – Enable 0 – Disable

byte_ack=0x06;   byte_start=0xFC
All values preceded by ‘0x’ are hexadecimal.

NOTES:
All values sent are expressed in 1/128 step. Therefore, if we intend to carry out a revolution to a motor of 200 steps/rev. sending value to drive would be 25600.
By changing the phase evolution mode from full step to half step, the value will remain the same.

SOME EXAMPLES OF COMMAND STRINGS ARE DESCRIBED ON THE NEXT PAGES.

4.5.5 EXAMPLES OF COMMANDS:

All examples given in the protocol refer to a drive having address 0.

COMMAND STRING	DRIVE ANSWER	FUNCTION
0xFC, 0x20, 0x01, 0xE2	0x06	Drive reset
0xFC, 0x20, 0x02, 0xE1	0x06	Software start
0xFC, 0x20, 0x10, 0xD3	0x06, 0xFC, 0x20, 0x10	Request for software version. The answer is 0x20 = version 2.0
0xFC, 0x20, 0x11, 0xD2	0x06	Immediate stop
0xFC, 0x20, 0x12, 0xD1	0x06, 0xFC, 0x80, 0x00, 0x00, 0x00, 0x00, 0x7D	Reading of current position. In this case the motor position is 0.
0xFC, 0x20, 0x13,0xD0	0x06, 0xFC, 0x40, 0x22	Input/output reading. In this case the third byte of the answer indicates that input 3 is activated.
0xFC, 0x20, 0x14, 0xCF	0x06, 0xFC, 0x20, 0x02	Request of drive type. The involved drive has the code number 0x20.
0xFC, 0x60, 0x20, 0x01, 0x5E, 0x24	0x06	Setting to 350 Hz minimum frequency.
0xFC, 0x60, 0x21, 0x07, 0xD0, 0xAB	0x06	Setting to 2000 Hz maximum frequency.
0xFC, 0x40, 0x22, 0x32, 0x6F	0x06	Setting of ramp inclination to 50 (0.5 seconds)
0xFC, 0xA0, 0x23, 0x00, 0x00, 0x00, 0x00, 0x40	0x06	Setting of absolute motor position to value 0
0xFC, 0x40, 0x26, 0x00, 0x9D	0x06	Setting of motor resolution to full steps
0xFC, 0x40, 0x27, 0x99, 0x03	0x06	Setting of current reduction to 25% of rated current after a time of 25 (25x32ms=0.8 seconds)
0xFC, 0x40, 0x28, 0x03, 0x98	0x06	Setting answer delay of the serial interface (3x512µs)
0xFC, 0x40, 0x29, 0x44, 0x56	0x06	Setting of start trigger on up-front input 3 (signal transition from low to high)
0xFC, 0x40, 0x2A, 0x22, 0x77	0x06	Setting of trigger stop on up-front input 2 (signal transition from low to high)
0xFC, 0x20, 0x2B, 0x00, 0xB8	0x06	Output level in position 0 when motor Is holding
0xFC, 0x40, 0x2C, 0x11, 0x86	0x06	Setting of home trigger of up-front input 1 (signal transition from low to high)
0xFC, 0xA0, 0x30, 0x00, 0x00, 0x64m 0x00, 0xCF	0x06	Absolute positioning equal to 1 motor rev. (value expressed in 1/128 of a step = 25600)
0xFC, 0xA0, 0x31, 0x00, 0x00, 0x64, 0x00, 0xCE	0x06	Relative positioning regarding present position equal to 1 motor rev. CW (value expressed in 1/128 step = 25600)

COMMAND STRING	DRIVE ANSWER	FUNCTION
0xFC, 0xA0, 0x31, 0xFF, 0xFF, 0x9C, 0x00, 0x98	0x06	Relative positioning regarding present position equal to 1 motor rev. CCW (value expressed in 1/128 step = -25600)
0xFC, 0x40, 0x32, 0x00, 0x91	0x06	Infinite CW motion
0xFC, 0x40, 0x32, 0xFF, 0x92	0x06	Infinite CCW motion.
0xFC, 0xC0, 0xA0, 0x11, 0x00, 0x00, 0x64, 0x00, 0x2E	0x06	Zero at flight active on input 1, transition low/high, with value to be carried out by the activation of the input equal to 1 motor rev.(expressed in 1/128 of a step = 25600)
0xFC, 0x20, 0xA6, 0x3D	0x06	Motion to value zero
0xFC, 0x60, 0xA8, 0x19, 0x64, 0x7E	0x06	Setting current to 6.5A
0xFC, 0xA0, 0xAA, 0xFF, 0xFF, 0x9C, 0x00, 0x1F	0x06	Setting of relative value to be carried out at the next START (value expressed in 1/128 of a step = -25600)

EXAMPLE OF CALCULATION Byte_Checksum (last byte of the string):

For sending the reset command to drive 0, the string will be as follows:
0xFC, 0x20, 0x01, Byte_Checksum.

For calculating the last byte, proceed as follows:

- Sum up all bytes of the command: 0xFC + 0x20 + 0x01 = 0x11D
- Consider only the less significant byte: 1D
- Complement the byte found, so to obtain the ByteChecksum: 0xFF – 0x1D = E2

The complete command to be sent will be as follows:

0xFC, 0x20, 0x01, 0xE2

4.5.6 SIMPLE MOTION PROGRAM:

This is a simple sequence of instructions to command the APS in SERIAL MODE:

The drive used is set to address 0

FUNCTION	COMMAND STRING	DRIVE ANSWER
Reset driver	0xFC 0x20 0x01 0xE2	0x06
Set serial answer delay to 5 msec	0xFC 0x40 0x28 0x0A 0x91	0x06
Set Fmin to 450 Hz	0xFC 0x60 0x20 0x01 0xC2 0xC0	0x06
Set Fmax to 5000 KHz	0xFC 0x60 0x21 0x13 0x88 0xE7	0x06
Set ramp to 100 msec	0xFC 0x40 0x22 0x0A 0x97	0x06
Set step division to ½ step	0xFC 0x40 0x26 0x01 0x9C	0x06
Start relative quote 10 revolution CW	0xFC 0xA0 0x31 0x00 0x03 0xE8 0x00 0x47	0x06

5. LABEL MACHINE SERIAL COMMANDS:

To use drive for label head see this examples:

5.1 LABEL MACHINE WITHOUT ENCODER:

FUNCTION	COMMAND	PARAMETER
- Set low loise mode	0xEE	0x02 (active)
- Set encoder mode	0xCB	0x00 (without encoder)
- Set trigger mode	0xC0	0x01 (triggers forever enabled)

- Set current	0xA8	Send current value from 1 to 12000 mA
		(Max value
		APS1=4000mA, APS2=6000mA,
		APS3=10000mA, APS4=12000mA)

- Set resolution	0x26	0=200 setps/rev, 1=400, 2=800, 3=1600, 4=3200
- Set minimum frequency (to start motor over resonance frequency)	0x20	Set value equal to step/rev. : (full step = 200, 1 / 2=400, 1 / 4=800, 1 / 8=1600, 1 / 16=3200)
- Set maximum frequency (speed)	0x21	Send value from 1 to 20000 Hz (steps / sec)
- Set acceleration ramp	0x22	Send value from 0 to 255 1/100 of second (acceleration reported from 1 to 10000Hz)
- Set start delay	0xC3	Send value from 0 to 65535 ms
- Set maximum quote	0xAA	Send maximum quote without reach fotostop [1/128 of steps]
- Set erogation quote	0xA0	Send 0x22 + quote (4 bytes) [1/128 of steps]
- Set START trigger	0x29	Send 0x11 (start on IN1 L-H edge)
- Set print mark activation time	0xC2	Send value from 0 to 255 1/10 of seconds

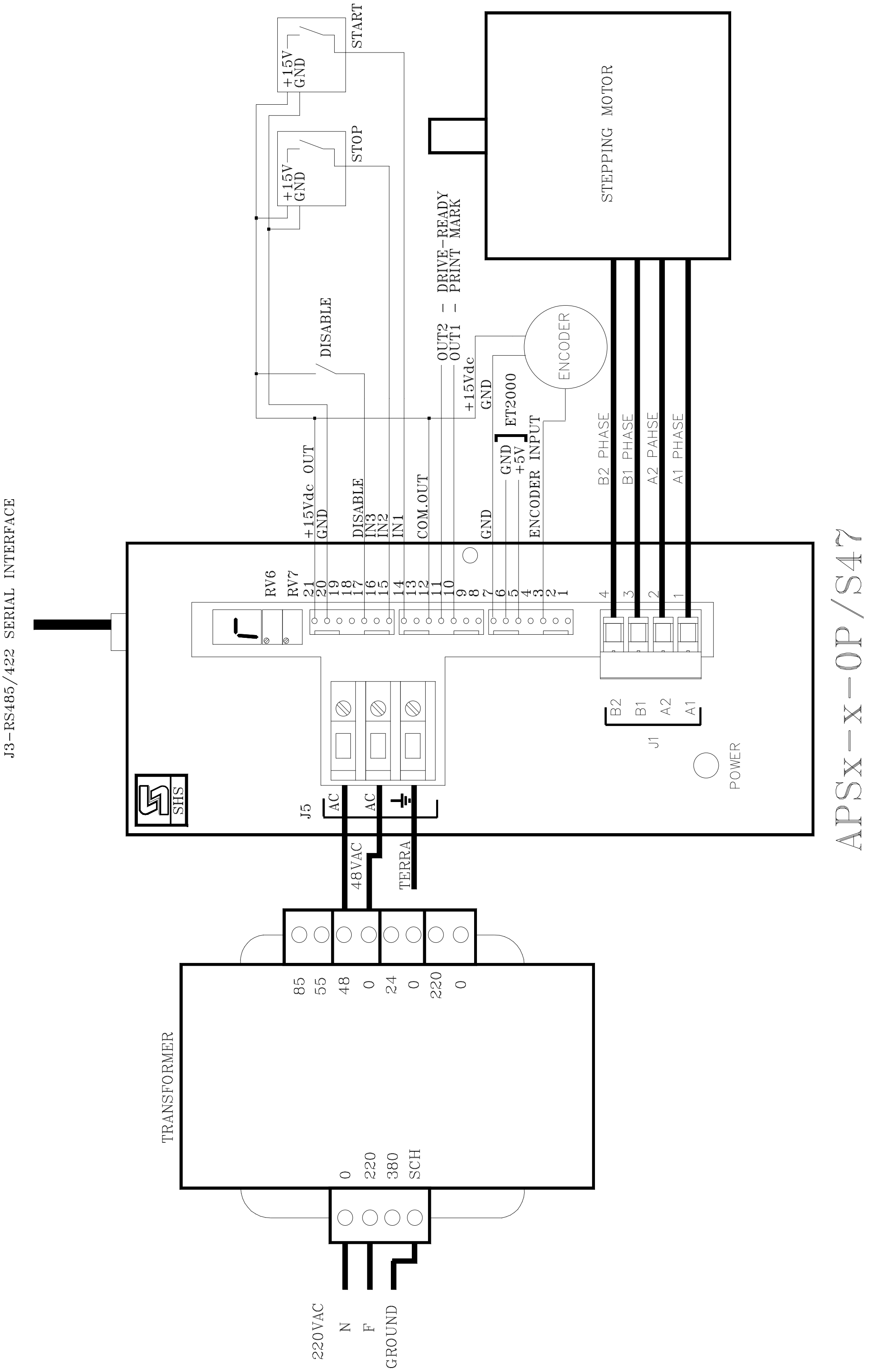
5.2 LABEL MACHINE WITH ENCODER:

FUNCTION	COMMAND	PARAMETER
- Set low noise mode	0xEE	0x02 (active)
- Set encoder mode	0xCB	0x01 (encoder enabled)
- Set trigger mode	0xC0	0x01 (triggers forever enabled)
- Set current	0xA8	Send current value from 1 to 12000 mA (Max value APS1=4000mA, APS2=6000mA, APS3=10000mA, APS4=12000mA)
- Set resolution	0x26	0=200 setps/rev, 1=400, 2=800, 3=1600, 4=3200
- Set minimum frequency (to start motor over resonance frequency)	0x20	Set value equal to step/rev. : (full step = 200, 1 / 2=400, 1 / 4=800, 1 / 8=1600, 1 / 16=3200)
- Set maximum frequency (speed)	0x21	Send value from 1 to 20000 Hz (steps / sec)
- Set acceleration ramp	0x22	Send value from 0 to 255 1/100 of second (acceleration reported from 1 to 10000Hz)
- Set start delay	0xC3	Send value from 0 to 65535 ms
- Set encoder/motor rate	0xC7	Send value from 0 to 65535. Example: Sending 1000, for each encoder pulse the motor perform one step. Sening 500, every two encoder pulse the motor perform one step.
- Set maximum quote	0xAA	Send maximum quote without reach fotostop [1/128 of steps]
- Set erogation quote	0xA0	Send 0x22 + quote (4 bytes) [1/128 of steps]
- Set START trigger	0x29	Send 0x11 (start on IN1 L-H edge)
- Set print mark activation time	0xC2	Send value from 0 to 255 1/10 of seconds

5.3 SEND A PARAMETER DURING MOTOR POSITIONING:

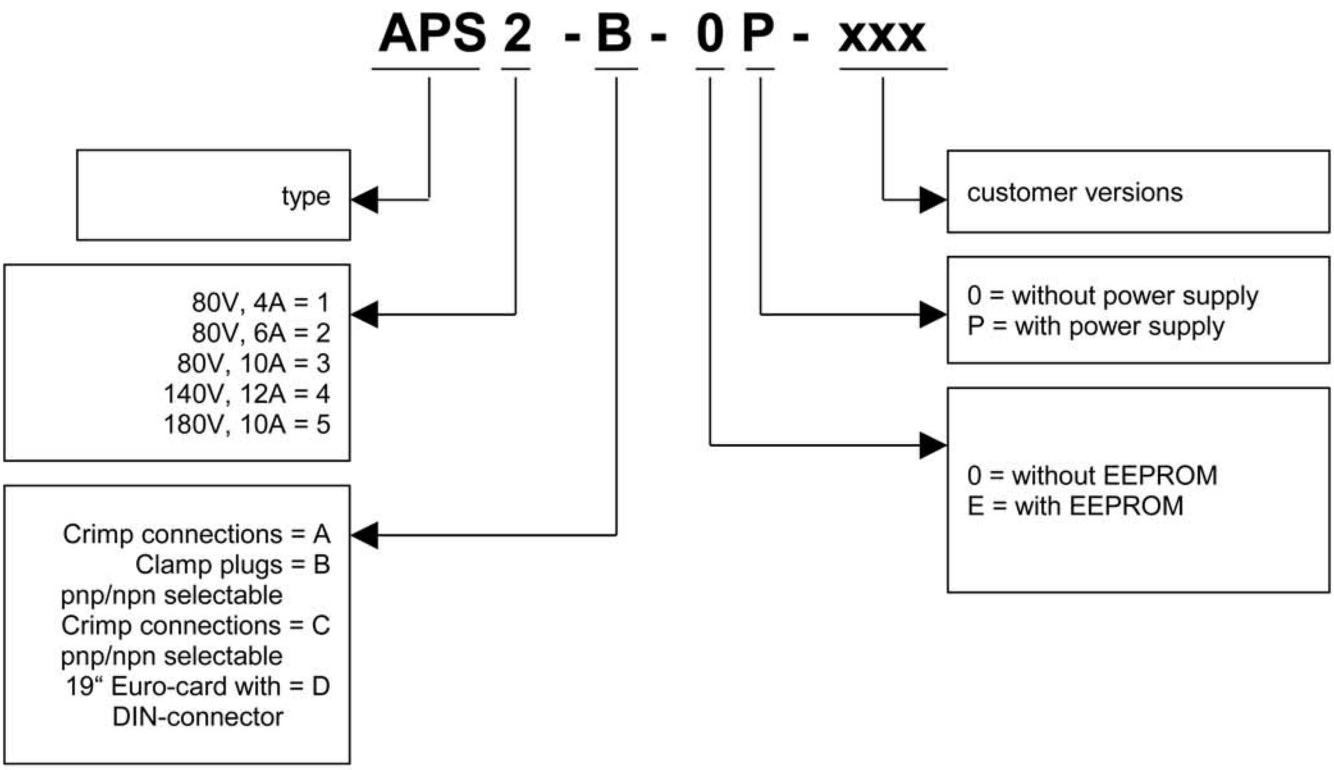
The parameter sended when motor is running will be acquired at next start. The speed setting will be active immediately with 0xAD (set % of speed) or 0xBD (set new speed) commands.

5.4 LABEL MACHINE CONNECTION DIAGRAM:



6. ORDER NUMBERS

The driver will be delivered in different versions. Please see the above typekey:



APS 2 - B - 0 P - xxx

type

80V, 4A = 1
80V, 6A = 2
80V, 10A = 3
140V, 12A = 4
180V, 10A = 5

Crimp connections = A
Clamp plugs = B
pnp/npn selectable
Crimp connections = C
pnp/npn selectable
19“ Euro-card with = D
DIN-connector

customer versions

0 = without power supply
P = with power supply

0 = without EEPROM
E = with EEPROM

ACP&D Limited
86 Rose Hill Road,
Ashton-under-Lyne,
Lancashire,
England,
OL6 8YF.

Tel: +44 (0)161 343 1884
Fax: +44 (0)161 343 7773
e-mail; sales@acpd.co.uk
Websites: www.acpd.com &
www.acpd.co.uk