For further Questions, please call:

ACP&D Limited.

Unit 9A,
Charlestown Industrial Estate,
Robinson Street,
Ashton-under-Lyne,
Lancashire, OL6 8NS.
Tel: +44 (0)161 343 1884
Fax: +44 (0)161 339 0650
e-mail: sales@acpd.co.uk

ACP & D Limited

Operating Manual

STEPPING MOTORS DRIVES
SERIES
STAR 2000
Mod.
APD
ver. 01/2000 Standard

RELEASE: 20FW9B

Edition: 19.02.2002

APD99English - 020618.doc

TECHNICAL DATA

SIZE	APD 1
Vac nom. [V]	From 14 to 26
Vac max. [V]	28
Vac min. [V]	13
I max. [A]	2.0
I min. [A]	0.2
I step [A]	0.2
Operating temperature [°C]	0-55

PARAMETERS DESCRIPTION

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

Vac max.: Maximum voltage at which drive can operate.

Vac min.: Minimum voltage at which drive can operate.

I max.: Maximum value of phase current.

I min.: Minimum value of phase current.

I step: Spacing of the eight current values.

Operation temperature: For any current over 1.5 Amps, a forced ventilation is necessary.

PROTECTIONS

Drive is provided with protections against over-temperature. If the mentioned condition occurs, drive disables the power bridge and shows an error condition on the display.

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

INPUTS AND OUTPUTS

Inputs are PNP (from 12Vdc to 30Vdc) and outputs are PNP (max. 10 mA).

SERIAL INTERFACE

Drives are supplied with RS 232 or RS 485 serial interface according to the models:

MODEL	SERIAL INTERFACE:
APD1/A	RS485 interface, half duplex or full duplex selectable by JP4 jumper.
APD1/B	RS232.

Communication protocol is on board and described afterwards.

DRIVE CONNECTION

POWER SUPPLY:

LEADS:

J1-1	power supply: AC or DC input
J1-2	power supply: AC or DC input

POWER SUPPLY INPUT STAGE:

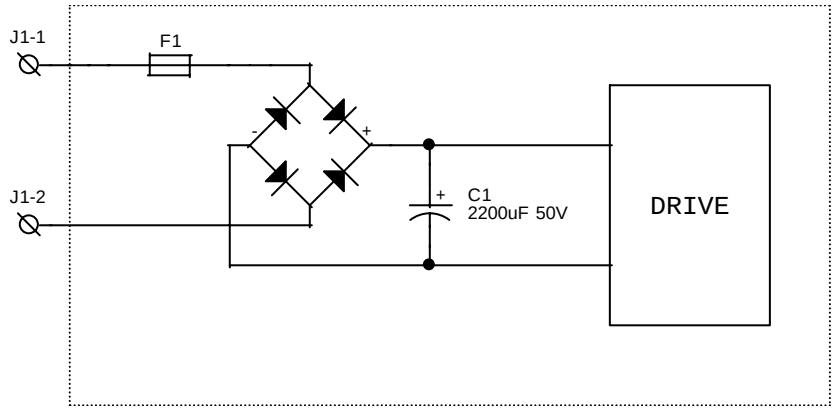


FIG.1

MOTOR CONNECTION:

LEADS:	**MOTOR PHASE:**
1-PHASE A1	Phase A1 step motor 1
1 PHASE A2	Phase A2 step motor 1
1-PHASE B1	Phase B1 step motor 1
1-PHASE B2	Phase B2 step motor 1

LEADS:	**MOTOR PHASE:**
2-PHASE A1	Phase A1 step motor 2
2-PHASE A2	Phase A2 step motor 2
2-PHASE B1	Phase B1 step motor 2
2-PHASE B2	Phase B2 step motor 2

INPUT / OUTPUT CONNECTIONS:

For inputs/outputs connections see diagram Fig. 2.

INPUTS/OUTPUTS DIAGRAM

Two drives are mounted on APD card, each one has a “disabled” signal: 3 inputs – 2 outputs.
Input/output status of each drive is shown in the following diagram:

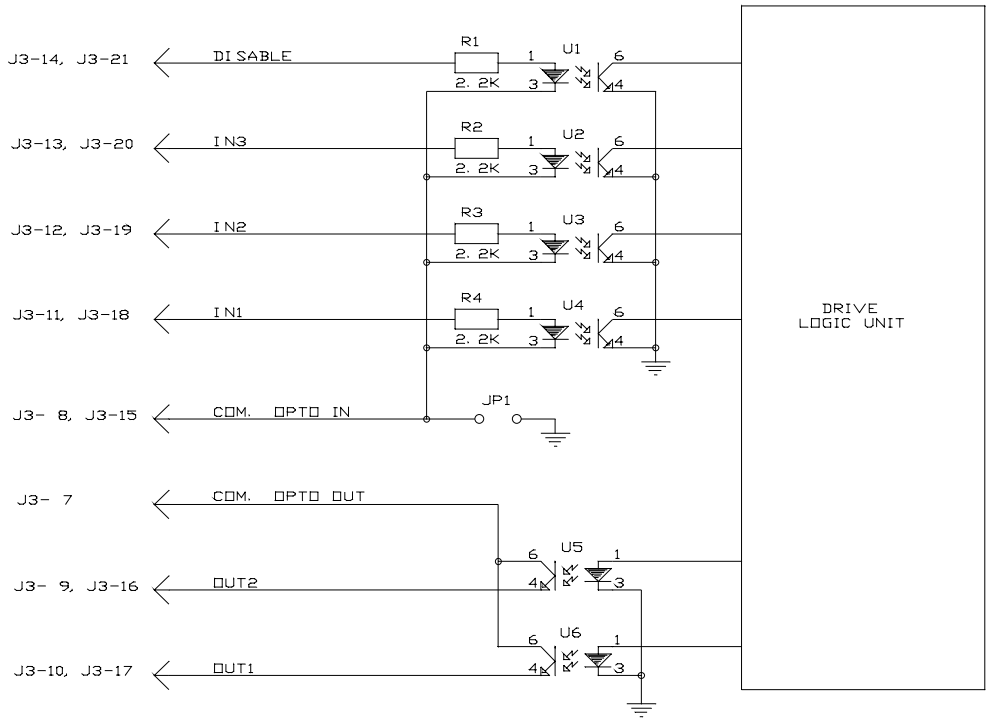


FIG.2

Inputs and outputs are PNP.

The drive assumes as high logic status any voltage between 12V and 24V

Non-optoisolated inputs	: Jumper JP1	- Inserted
Optoisolated inputs	: Jumper JP1	- Not Inserted
	Pin 18 of J3 connector	- inserted to GND of external power supply (common input optoisolators)

Outputs are always optoisolated. Therefore, an external powering (+12V/+24V) must be connected to the common pole of the optoisolators (Pin 7 of conncector J3). Max. current for each output 10 mA.

JUMPERS SETTINGS

JP1 - When inserted, it associates the inputs common pole with drive GND (non-optoisolated inputs)

JP2 -When inserted into 1-2 position, aux. Output +12V
When inserted into 2-3 position, aux. Output +5V

JP4 - When inserted into 1-2 position RS485 serial interface is on full duplex mode (only APD1/A model)
When inserted into 2-3 position RS485 serial interface is on half duplex mode (only APD1/A model)

JP5,JP6 - When inserted, they add termination resistors (120 Ohm) needed for the last drive of the chain, between the signals TX+,TX- and RX+,RX- of the RS485 serial interface.

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

OPERATING MODE

Drive can be used in standard mode (with step and direction signals), or in serial interface mode.

DIPB 1	OFF:	Standard mode (steps/direction)
	ON :	Serial mode (RS232 or RS485 according to the models)

Select must be done before powering the drive (through Dip-Switch B 1)

STANDARD MODE (STEP/DIRECTION)

In each of both drives following signals are present :

INPUTS:

SIGNAL	FUNCTION
DISABLE	It disables the power bridge.
CURRENT REDUCTION	It reduces the motor current. The percentage of current reduction can be set from 25% to 50% of the regulated current through dip-switch A-1 (for motor 1) and dip-switch A-4 (for motor 2).
DIRECTION	Select the motor wise Signals must be stable for at least 50 microsecunds before and 50 microsecunds after the low/high transition of the STEP-IN signal.
STEP-IN	Motor execute the step on the LOW/HIGH transition of this signal. Use a square wave with duty-cycle of 50%. Signal absence for 0.5 seconds determines automatic current reduction (stand-by condition). Percentage of reduction in stand-by can be set to 25% or to 50% of the regulated current through dip-switch A-1 (for motor 1) and dip-switch A-4 (for motor 2).

OUTPUTS:

SIGNAL	FUNCTION
OUT1	Unassigned
OUT2	Output DRIVE-READY . Open collector (10 mA max) Drive in protection: Low level Drive ready : High level

CURRENT REGULATION:

For current regulation of both drives proceed as follows:

1 - Turn dip-switch B-4 to ON (current regulation mode). A number will appear on display showing selected current See table stated here below.
2 - Turn dip-switch B3 to OFF in order to display and to regulate current on DRIVE 1.
Turn dip-switch B3 to ON in order to display and to regulate current on DRIVE 2.
3 - Turn dip-switch B4 to OFF in order to exhit from current regulation mode.

Following table shows the relation between the displayed numbers and the effective entered current:

DISPLAYED VALUE	SETTING CURRENT (A)
1	0.2
2	0.4
3	0.6
4	0.8
5	1.0
6	1.2
7	1.4
8	1.6
9	1.8
A	2.0
B	2.2
C	2.4

NOTE: ADJUST CURRENT WHEN MOTOR IS HOLDING

DIP-SWITCHES SETTINGS IN STANDARD MODE:

DIP SWITCH A		
DIP	**ON**	**OFF**
6	Half step (400 steps/rev.) (MOTOR 1)	Full step (200 steps/rev.) (MOTOR 1)
5	Unassigned	Unassigned
4	50% current reduction in stand-by condition in respect to the entered current. (MOTOR 1)	75% current reduction in stand by condition in respect to the entered current. (MOTOR 1)
3	Half step (400 steps/rev.) (MOTOR 2)	Full step (200 steps/rev.) (MOTOR 2)
2	Unassigned	Unassigned
1	50% Current reduction in stand-by condition in respect to the entered current. (MOTOR 2)	75%Current reduction in stand-by condition in respect to the entered current. (MOTOR 2)

DIP SWITCH B		
DIP	**ON**	**OFF**
4	Current regulation mode	RUN mode
3	MOTOR 1 current regulation (dip-switch A-4 ON)	MOTOR 2 current regulation (dip-switch A-4 ON)
2	Unassigned	Unassigned
1	Serial commands operation (select before switching on the drive)	Steps/Direction operation (select before switching on the drive)

SERIAL MODE

Commands enter through serial interface.

INPUTS:

SIGNAL	FUNCTION
DISABLE	It disables power bridge
IN1	Programmable inputs as per instructions stated in the following.
IN2	Idem
IN3	Idem

OUTPUTS:

SIGNAL	FUNCTION
OUT 1	IN-POSITION output: Motor holding : Low level Motor running : High level The indicated levels are the default ones. They can be inverted through a serial command. (see 0x2B, page 14)
OUT 2	DRIVE-READY output: Drive in protection: Low level Drive ready : High level

CURRENT REGULATION:

Motor current can be set by on board trimmers.pot., as described in STANDARD mode: besides this value can be changed through a serial command.

The trimmers.pot setting value is acquired by the processor at 'power on' or at current regulation setting.

Current setting through serial command will remain available only until drive will be un-powered or until current regulation will be set through the trimmers.pot..

Therefore trimmer can be used for setting a default current at 'power-on', subsequently current can be set to a different value through serial command.

SWITCHES AND COMMUNICATION INTERFACE SETTINGS

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 DRIVE 1	ADDRESS DRIVE 2
OFF	OFF	OFF	OFF	OFF	0	1
OFF	OFF	OFF	OFF	**ON**	1	2
OFF	OFF	OFF	**ON**	OFF	2	3
OFF	OFF	OFF	**ON**	**ON**	3	4
OFF	OFF	**ON**	OFF	OFF	4	5
OFF	OFF	**ON**	OFF	**ON**	5	6
OFF	OFF	**ON**	**ON**	OFF	6	7
OFF	OFF	**ON**	**ON**	**ON**	7	8
OFF	**ON**	OFF	OFF	OFF	8	9
OFF	**ON**	OFF	OFF	**ON**	9	10
OFF	**ON**	OFF	**ON**	OFF	10	11
OFF	**ON**	OFF	**ON**	**ON**	11	12
OFF	**ON**	**ON**	OFF	OFF	12	13
OFF	**ON**	**ON**	OFF	**ON**	13	14
OFF	**ON**	**ON**	**ON**	OFF	14	15
OFF	**ON**	**ON**	**ON**	**ON**	15	16
ON	OFF	OFF	OFF	OFF	16	17
ON	OFF	OFF	OFF	**ON**	17	18
ON	OFF	OFF	**ON**	OFF	18	19
ON	OFF	OFF	**ON**	**ON**	19	20
ON	OFF	**ON**	OFF	OFF	20	21
ON	OFF	**ON**	OFF	**ON**	21	22
ON	OFF	**ON**	**ON**	OFF	22	23
ON	OFF	**ON**	**ON**	**ON**	23	24
ON	**ON**	OFF	OFF	OFF	24	25
ON	**ON**	OFF	OFF	**ON**	25	26
ON	**ON**	OFF	**ON**	OFF	26	27
ON	**ON**	OFF	**ON**	**ON**	27	28
ON	**ON**	**ON**	OFF	OFF	28	29
ON	**ON**	**ON**	OFF	**ON**	29	30
ON	**ON**	**ON**	**ON**	OFF	30	31

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

DIP SWITCH B – SERIAL MODE		
DIP	**ON**	**OFF**
4	Current regulation mode	RUN mode
3	MOTOR 1 current regulation (with DIPA-4 ON)	MOTOR2 current regulation (with DIP A-4 ON)
2	Damping deactivated for MOTORS 1 and 2	Damping activated for MOTORS 1 and 2 (for damping the mechanical motor's resonances)
1	Serial mode (set before powering up the drive)	Standard mode (set before powering up the drive)

TRANSMISSION TIMING OF SERIAL COMMANDS:

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

Enter drive delay answer commands (command 0x28); this value allowes PC or PLC to have enough time for receiving drive answer (ex. 5ms)

Enter following commands:

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)

RS485/422 SERIAL CONNECTION DIAGRAM:

RS485 FULL DUPLEX CONNECTION

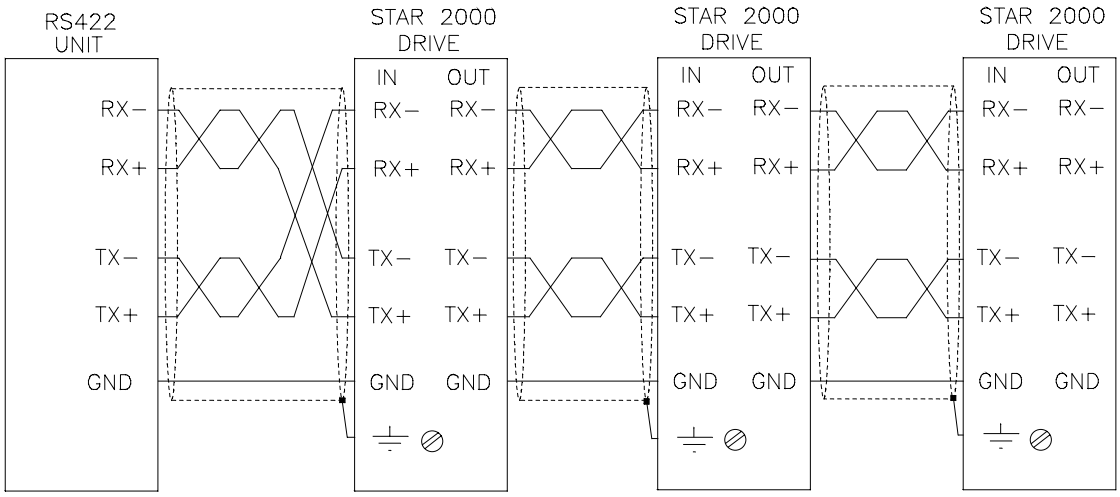


FIG.3

To use RS485 full duplex (RS422) set JP4 jumper between 1-2 position (factory default) and see FIG.3.

RS485 HALF DUPLEX CONNECTION

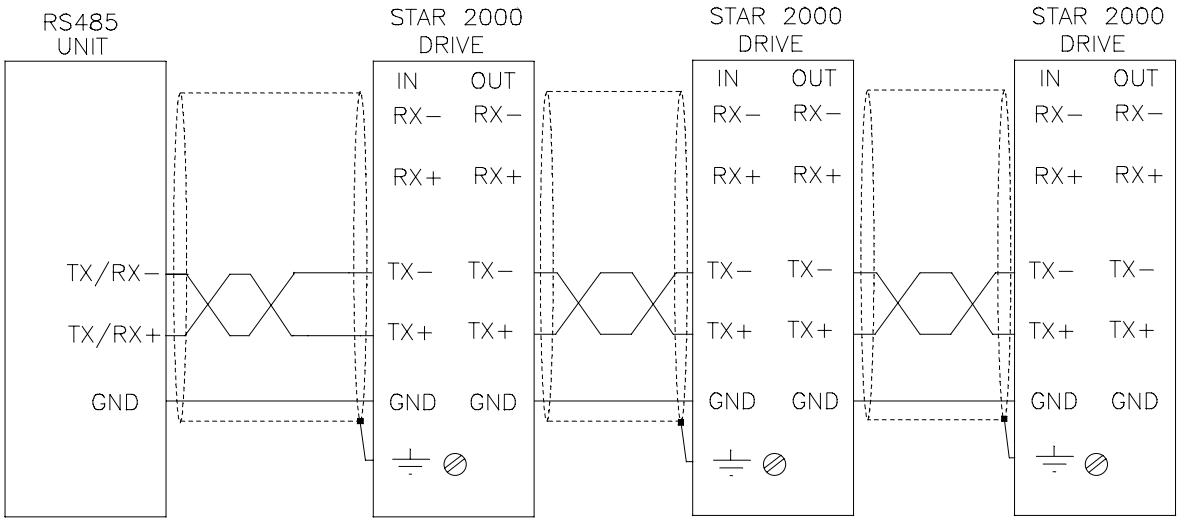


FIG.4

To use RS485 half duplex set JP4 jumper between 2-3 position and see FIG.4.

COMMUNICATION PROTOCOL

Systems can use a single drive or several drives, which are connected in multi-drop 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.

SINGLE ADDDRESS COMMAND:

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

byte_start, byte_nbyte_address, byte command, [byte_par0] , [byte_par1] , byte_checksum

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

byte_nbyte_address : This byte contains two indications:
- The first 5 bits (from bit0 to bit4) contain the drive address (from 0 to 31).
- 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 entered 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 command transmission (see example in Appendix A).

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 foreseen in the above mentioned format.

MULTI ADDRESSES COMMAND:

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

byte_start, byte_nbyte_address, byte_multiaddress, byte_command, [byte_par0], byte_address1, [byte_address2], [byte_address3], [byte_address4], byte_checksum

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

byte_nbyte_address : This byte contains two information:
- In the first 5 bytes enter no. 31 address
- 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 entered 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 Appendix A).

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

COMMAND ADDRESSED TO ALL DRIVES:

DATA TO BE SENT TO THE DRIVES:
Commands format 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_start : 0xFC. This byte means that a command will be sent to only one drive

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

byte_nbyte : This byte indicates the bytes number 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, which follow *byte_command*, represent the entered 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 Appendix A).

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.

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 decelerates according to the preset ramp and then it stops.
0x12	None	*byte_ack+byte_start+ byte_nbyte_address+byte1 +byte2+byte3+byte4+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 by 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 by the next motion command.
0x23	4 bytes, which indicate the absolute motor position (expressed in 1/128 step)	*byte_ack*	SETTING OF THE ABSOLUTE POSITION: Drive associates the entered value as present position of the motor

COMMAND	PARAMETERS	ANSWER	FUNCTION
0x26	1 byte, which indicates motor resolution. If the entered byte = 0 full step mode = 1 half step mode	*byte_ack*	SETTING OF MOTOR RESOLUTION NOTE: If motor is running, this parameter will be acquired by 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. 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 first 4 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. 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. 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 as to the HOME position, which the motor must reach (expressed in 1/128 step). (values admitted: from -2147483647 to 2147483647)	*byte_ack*	ABSOLUTE POSITIONING (REGARDING HOME POINT)

COMMAND	PARAMETERS	ANSWER	FUNCTION
0x31	4 bytes, which indicate the to be executed positioning with respect to the present position of the motor (expressed in 1/128 step) (value 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: It defines the input or the inputs and the respective levels, which must be enabled for executing zeroing of the value in the present motor position, when this condition comes and the executing value on occasion of this condition.
0xA6	None	*byte_ack*	MOTION TO ZERO VALUE
0xA8	2 bytes, which indicate the current value (from 0 to 2000 mA)	*byte_ack*	CURRENT SETTING (ex. 1000 = 1A, 2000=2A). Entering a wrong value the answer will be Byte_nack.

COMMAND	PARAMETERS	ANSWER	FUNCTION
0xAC	None	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: Only one byte includes all information regarding the drive status

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

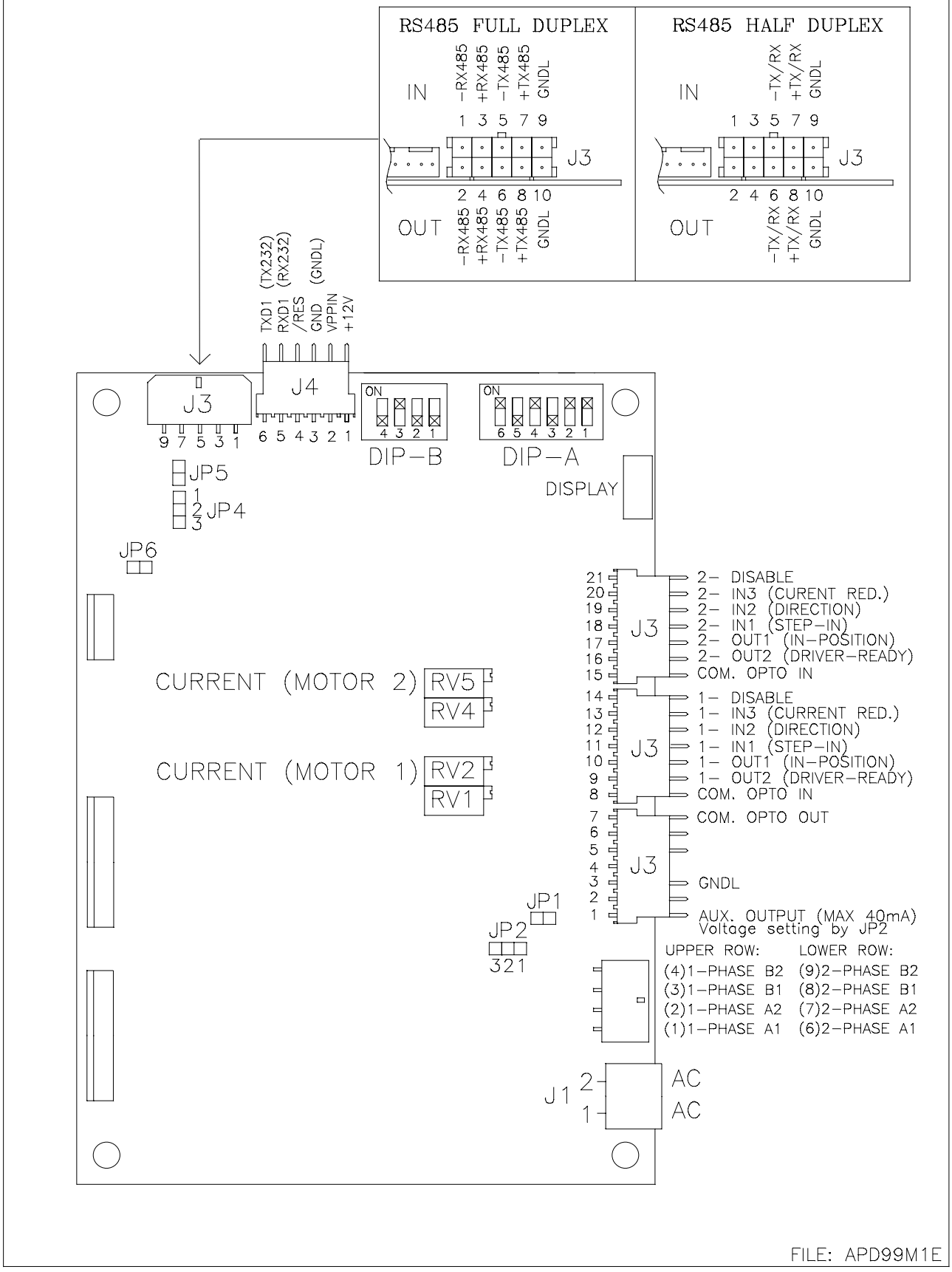
NOTES:
All values sent are expressed in 1/128 step. Therefore, if we intend to carry out a revolution to a motor of 200 steps/rev. sending value to drive would be 25600.

By changing the phase evolution mode from full step to half step, the value will remain the same.

Input trigger commands are enabled as soon as the command is sended and still active till executed. To repeat send command again

SOME EXAMPLES OF COMMAND STRINGS ARE DESCRIBED IN APPENDIX A.

COMPONENTS LAYOUT APD DRIVE



APPENDIX A:

EXAMPLES OF COMMANDS:

All examples given here below 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 present 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 answer byte 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 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)

0xFC, 0xA0, 0x31, 0xFF, 0xFF, 0x64, 0x00, 0xCE	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 executed 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

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