DS30 – DS5x

(Hardware rev. 1.10   Firmware rev. 0.07)

Programmable stepper motor drives with integrated Modbus-RTU field bus

User’s Manual

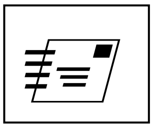
Contacts:

- *General information:* info@lamtechnologies.com
- *Technical support:* support@lamtechnologies.com
- *Sales information:* sales@lamtechnologies.com

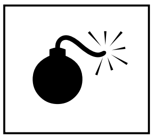
1 Notes, Terms and Warnings

In this manual some symbols, whose meaning is listed below, are used to underline particular arguments.

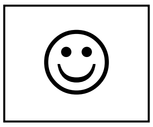
There is a dangerous condition that must be measured and avoided. The not-respect of indications marked with this symbol can cause serious damages to people, animals and things.



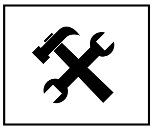
The argument is very wide and could require a deeper examination with the technical support.



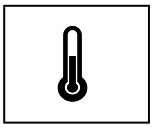
The not-observation of what described could damage the products.



Features and functionalities which cannot be easily found in other products. A shortcut to reach a target is shown.



A change or repairing intervention which can be made directly by the user.



An aspect which is connected to the temperature or longevity of the product.

The terms listed below are also used:

Product

The microstepping driver described herein.

User

Who selects and/or installs and /or uses the product.

Application

The machine, the equipment, the device, etc. on which the product is applied.

2 Risks and Precautions

A T T E N T I O N

Following are listed the most important warnings to be fully comprehended and applied by the user who, in case of non-complete comprehension or impossibility to apply them correctly, must not use the product at all.

The DS30 – DS5x drivers are components. It is user's responsibility as the installer to be sure the product complies with the rules and regulations in force. The user must also be trained in the installation of the electronic equipment to fully comprehend the features, the calibration parameters and indications contents of this guide.
The user must provide for the application of all the local safety laws and regulations in force in the Country and/or application in which the product is used.

The user must provide that the product is inaccessible while powered on. The user must also consider that, because of the capacitors inside the driver, it is necessary to wait at least for 30 seconds after the power off before acceding to the driver. According to the external capacitors eventually mounted on the power supply circuit, it is possible the user shall have to wait more time.

While working the product produces heat which can raise the temperature of some parts (as heat sink, for example) up to values that can cause burns. Such a condition remains for a long time even after the product has been powered off. The user must provide for the appropriate protections and signals, must train the operator, the staff of the technical support and risk maintenance, and then must state it in the service manual of the finished product.

The high performance driver is able to produce rapid accelerations with high motor torque. Never touch any mechanical part while the driver is powered on. The user must prearrange the application in order this condition is always granted.

The power supply of the product must be isolated from the electrical net. The user must always place a protective fuse in series to the power supply circuit.

The control signals are usually isolated from the power supply while the product is in operation; anyway, while designing the application, the eventuality that in case of breakdown these lines could reach the same potential as the power supply must be considered to meet safety requirements.

The product could be permanently damaged by corrosive substances (such as gas, salts, etc), liquids or conductive dusts. Even a long and constant exposure to strong vibrations could cause the damage.

On some damage conditions the driver could emit sparks and fire. The cabinet and the nearby components must be chosen to tolerate this eventuality and to avoid propagation of flames to the application.

The products cannot be used in life support applications or where the failure of the products could cause death or injury to people, animals or things, or economic losses. The user not able to assure this condition must not use the products herein described.

!	Do not dismantle the product, do not try and repair it and do not modify it without being expressly authorized by LAM Technologies.

!	If the product is used in any manner that does not comply with the instructions given in this manual, then the product could be permanently damaged. For example, the product could be permanently damaged if power supplied with voltage superior to the allowed one, if supply polarity is inverted, if the motor is connected or disconnected while the driver is operating, and so on.

3 Introduction

3.1 Product Description

The DS30 – DS5x series drivers have a built-in flexible programmable motion controller able to perform accurate motor control in speed and position.

The programming is quick and simple and it is made putting in sequence the various functional blocks available in *UDP Commander* development software tool, such as assignment blocks, conditional jump blocks, etc. Particularly powerful are the mathematical blocks able to execute additions, subtractions, multiplications and divisions and which allow to realize even complex applications.

The connection with the external devices is through 4 inputs and 2 digital outputs each one optocoupled, independently PNP or NPN or line driver usable. Two +/-10V analog inputs and one 0-10Vdc analog output complete the available interface signals.

Suitable for the bi-polar mode driving of two phase stepper motors, the DS30 – DS5x series drivers feature an accurate current control which assures a correct and smoothness movement of the motor. The use of last generation power components together with the development of an innovative current control technology it has been possible to reach high level of efficiency, to be compact sized and to reduce driver heating.

The high technology in the product has allowed to overcome the previous full step or fractioned limited step solutions, offering at the same price a product able to drive in microstepping mode with an high step frequency.

The DS30 – DS5x drivers series is equipped with a special port called DUP, designed for the programming, the debug and diagnostics of the driver. Through this special device it is possible to download into the driver the application program, to visualize the content of registers, I/O signals, etc.

The bundled driver software simply and quickly guides the user in all the programming phases and assists him during the diagnostics and application debug.

The use of last generation components and technologies, together with the computerized test made on each single unit, gives to the product itself high reliability and economic competitiveness.

3.1.1 Available models

The DS30 – DS5x family drivers are available in a complete range of models diversified according to the power supply voltage and the phase current delivered by the motor.
All the models share the same functional features:

Model	Power supply voltage (Vdc)		Effective phase current (Arms)		Peak phase current (Apk)	
	Min	Max	Min	Max	Min	Max
DSxx44	20	50	1	4	1.4	5.6
DSxx48	20	50	3	8	4.2	11.3
DSxx73	24	90	0.8	3	1.1	4.2
DSxx76	24	90	2	6	2.8	8.5
DSxx78	24	90	4	10	5.6	14.1
DSxx84	45	160	2	4	2.8	5.6
DSxx87	45	160	4	8.5	5.6	12.0
DSxx98	45	240	4	10	5.6	14.1

3.1.2 Main Features

- ✓ Simple programming at blocks
- ✓ Mathematical functions at 32bit
- ✓ Speed or position control
- ✓ Independent acceleration and deceleration ramps
- ✓ Absolute or relative positioning
- ✓ Resolution up to 2500rpm at 1/128 step/rev
- ✓ Quote from -2,147,483,638 to +2,147,483,647
- ✓ Four digital inputs and two +/-10V analog inputs
- ✓ Two digital outputs and one 0-10V analog output
- ✓ Optocoupled and differential I/O, independently NPN or PNP usable
- ✓ Analog inputs at 11bit
- ✓ Digital inputs from 3Vdc up to 30Vdc with current regulator
- ✓ Line driving supported
- ✓ Protective diode for inductive loads on each digital output
- ✓ Wide range of power supply
- ✓ Resonance damping
- ✓ Automatic current reduction settable from 0% up to 100% with increments of 10%
- ✓ Single power supply voltage for the whole driver
- ✓ Current setting with increments of 0,1Arms
- ✓ Accurate current control
- ✓ High efficiency, low losses and contained heat
- ✓ Colored and numbered removable terminal blocks for easy and fast wiring
- ✓ Complete diagnostics
- ✓ Over/under voltage protections
- ✓ Phase to phase short circuit protection both direct and crossed
- ✓ Phase to ground short circuit protection
- ✓ Positive supply short circuit protection
- ✓ Temperature protection
- ✓ Interrupted phase alarm independent for each phase
- ✓ Univocal indication for each anomaly
- ✓ Malfunctioning status LED indicator
- ✓ Motor qualification LED indicator
- ✓ Power supply LED indicator
- ✓ Digital realization
- ✓ PC assisted simple setting
- ✓ PC assisted diagnostics
- ✓ Compact size
- ✓ Simple and fast DIN rail mounting
- ✓ Low cost

3.2 Accessories

Code	Description
UDP30	Setting and diagnostics interface
LSP1004	DIN rail kit consisting of hook and spring
LSP4002	Terminal blocks kit consisting of: 1pc. numbered removable terminal block, red color, 2 poles 1pc. numbered removable terminal block, grey color, 5 poles 1pc. numbered removable terminal block, grey color, 10 poles
LSP4004	5pcs. numbered removable terminal block, red color, 2 poles
LSP4006	5pcs. numbered removable terminal block, grey color, 5 poles
LSP4007	5pcs. numbered removable terminal block, grey color, 6 poles
LSP4008	5pcs. numbered removable terminal block, grey color, 10 poles

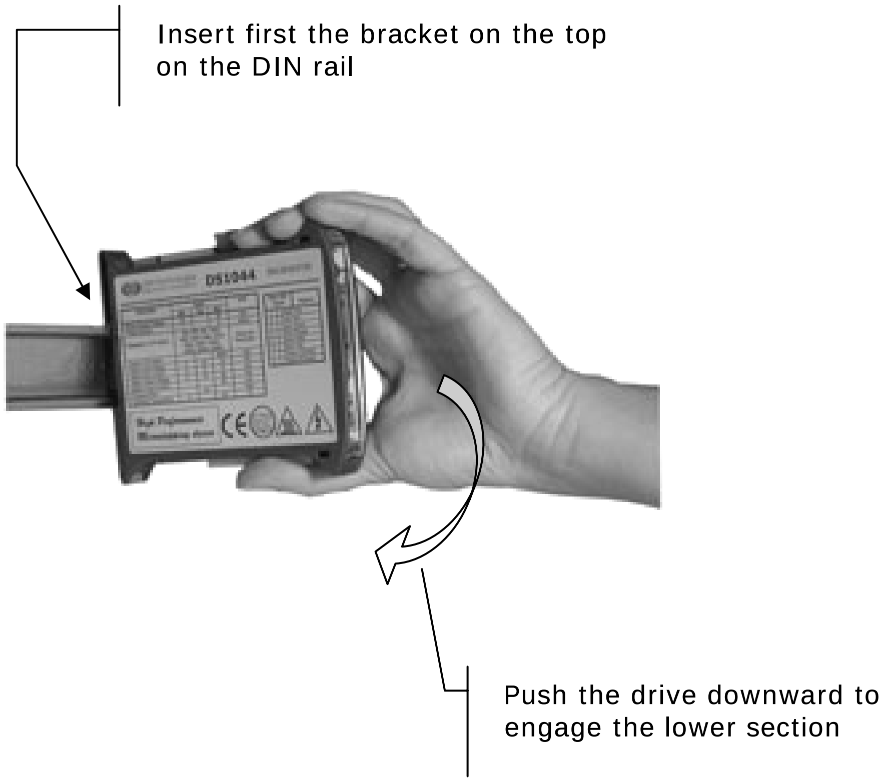
4 Installation

4.1 Inspection

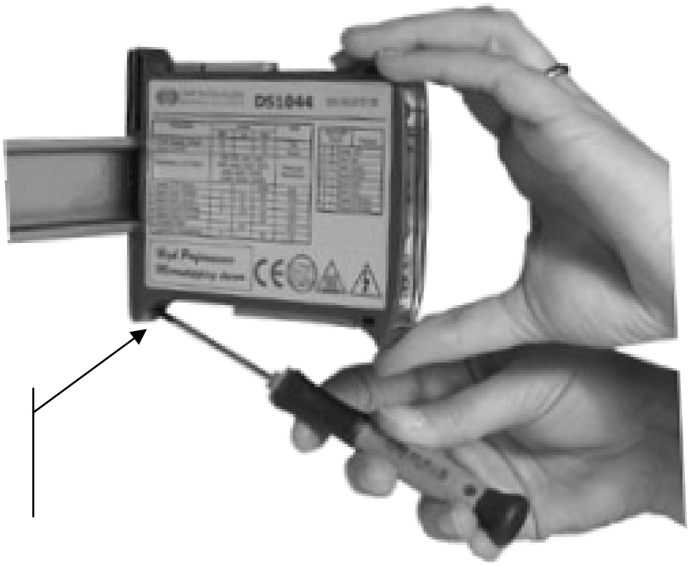
Verify the driver is not damaged, the package is intact and all accessories of the ordered product are included. Furthermore, control the driver code corresponds to the ordered one, eventual special and customized version included. In case of problems please address to the product's vendor.

4.2 Mechanical Installation

The driver is designed to be mounted vertically on a 35mm DIN rail.
To block the driver on the DIN rail, insert first the bracket on the top, on the back of the driver, over the top of the DIN rail, keeping the driver slightly inclined as shown in the figure, then push the driver downward to engage the lower section of the rail. To verify the correct engagement of the driver try and pull it slightly upward to control it is still in position.



To remove the driver from the DIN rail insert a small flat bladed screwdriver into the red colored hook on the bottom, on the back of the driver. Push the hook downward and pull the driver upward slightly rotating it, releasing it from the DIN rail as shown in the figure.

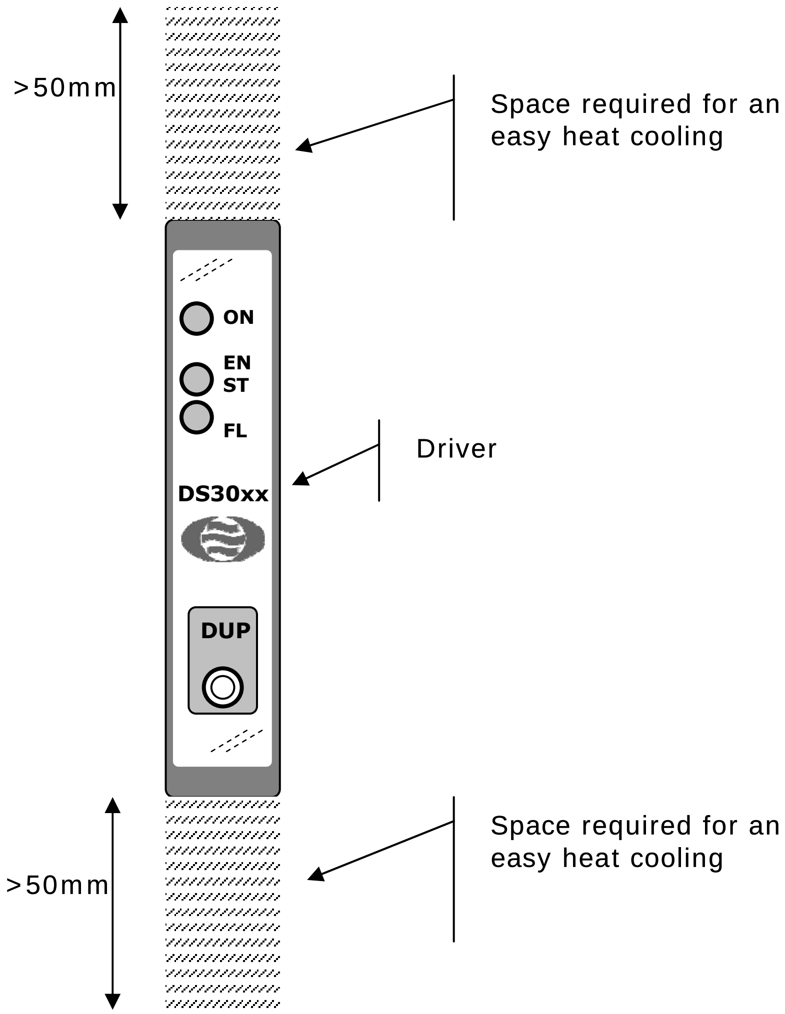


Insert a small flat bladed screwdriver into the red colored hook

Push the screwdriver slightly downward and pull the driver upward, releasing it from the DIN rail

The heat generated by the driver while operating must be dissipated toward the surrounding air. To assist cooling, the driver must be installed vertically in an area with a sufficient air gap of about 50mm above and below the driver itself, with no obstructions (wiring cables are anyway allowed). No space needs to be left on the sides and more drivers can be packaged side by side taking a very compact space.

According to the driver calibration and to the running cycle, the space can be also substantially reduced without compromising the correct working of the driver.

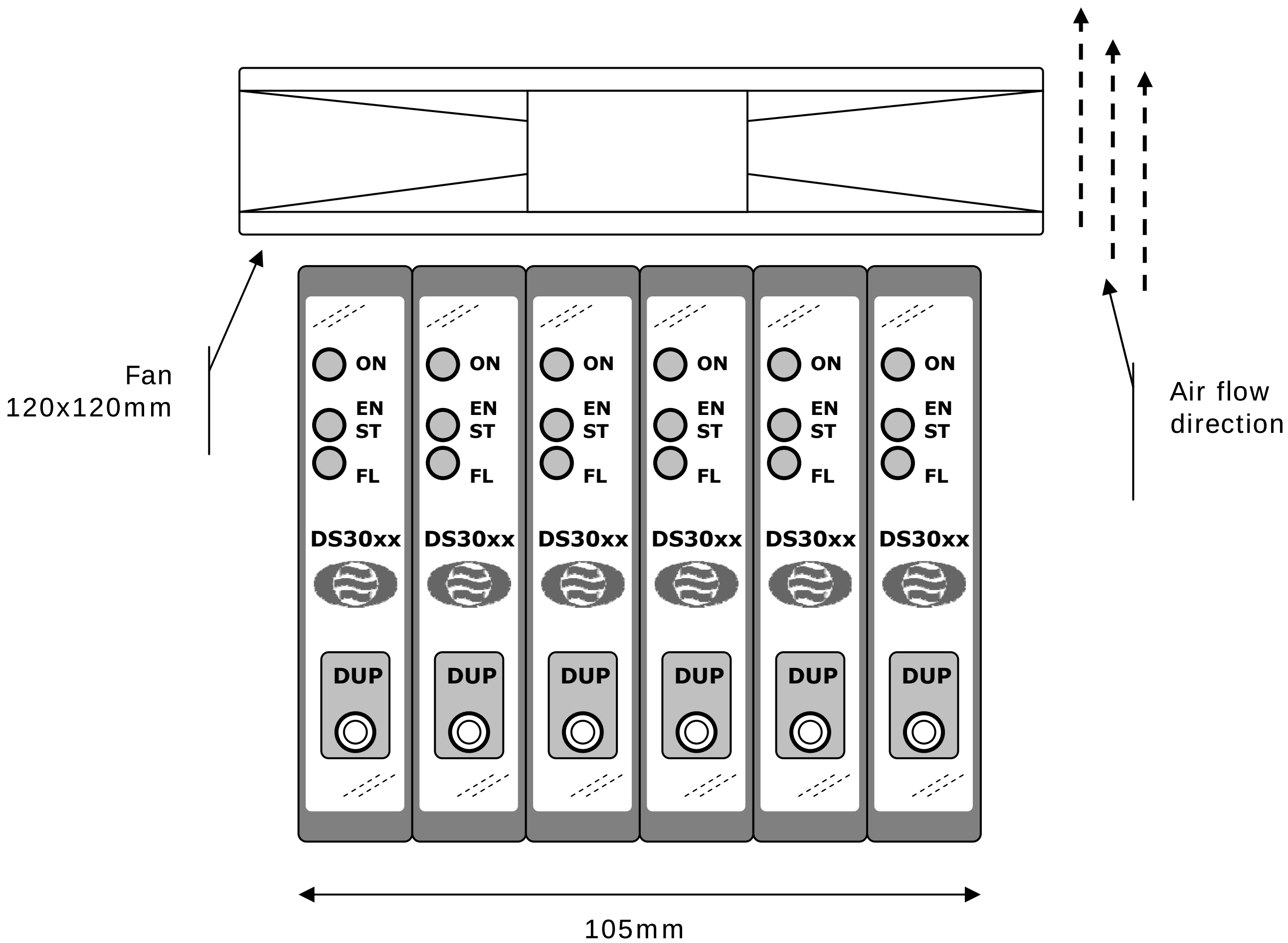


4.3 Air Cooling

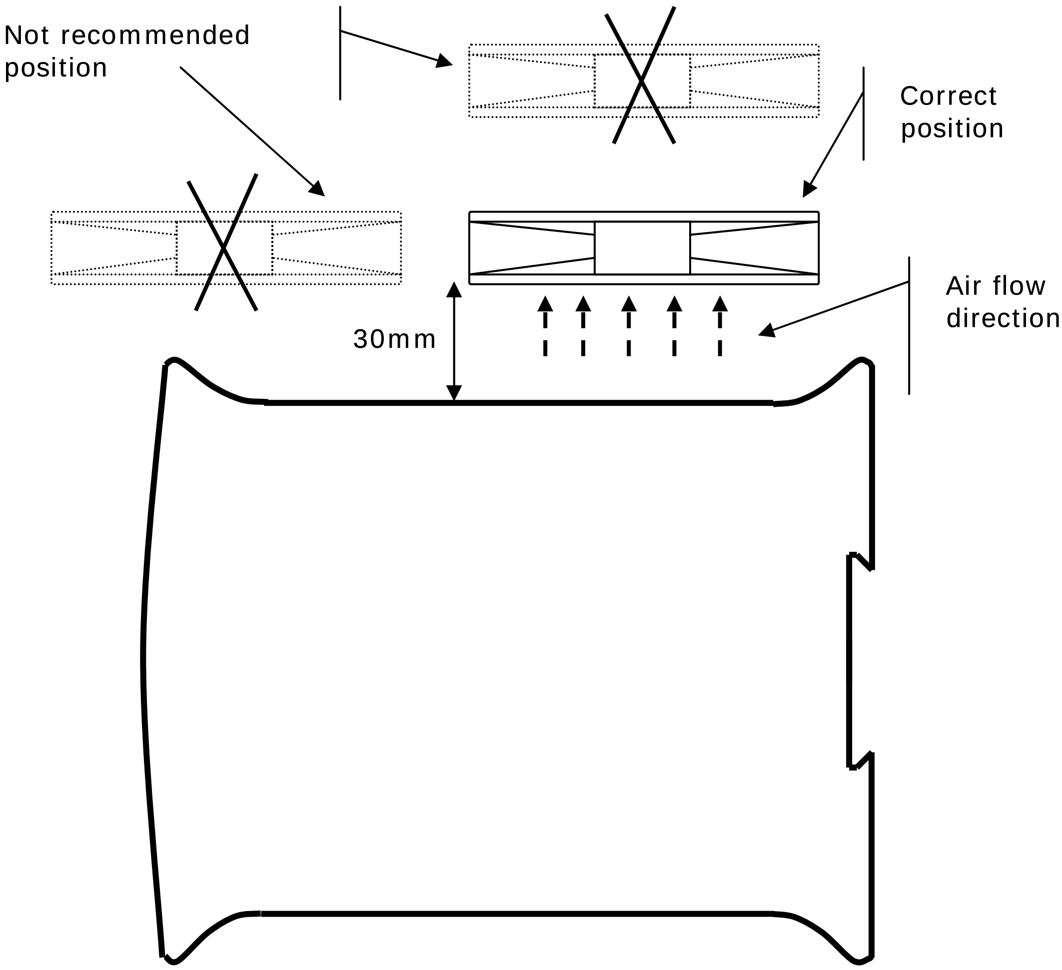
Its exceptional efficiency allows the product to be used even if absence of forced ventilation.

Whereas the driver is used with high calibration current or at high power supply voltage, whereas the working cycle is very heavy or the ambient temperature is high, it is possible to recur to forced ventilation to maintain the driver temperature within the functioning range values.

The fan can be positioned over or under the driver. A lateral position is not recommended. Thanks to the compact size of the driver, an only and economic fan of 120mm x 120mm can supply air circulation sufficient for 6 drivers simultaneously (models of 18mm wide).



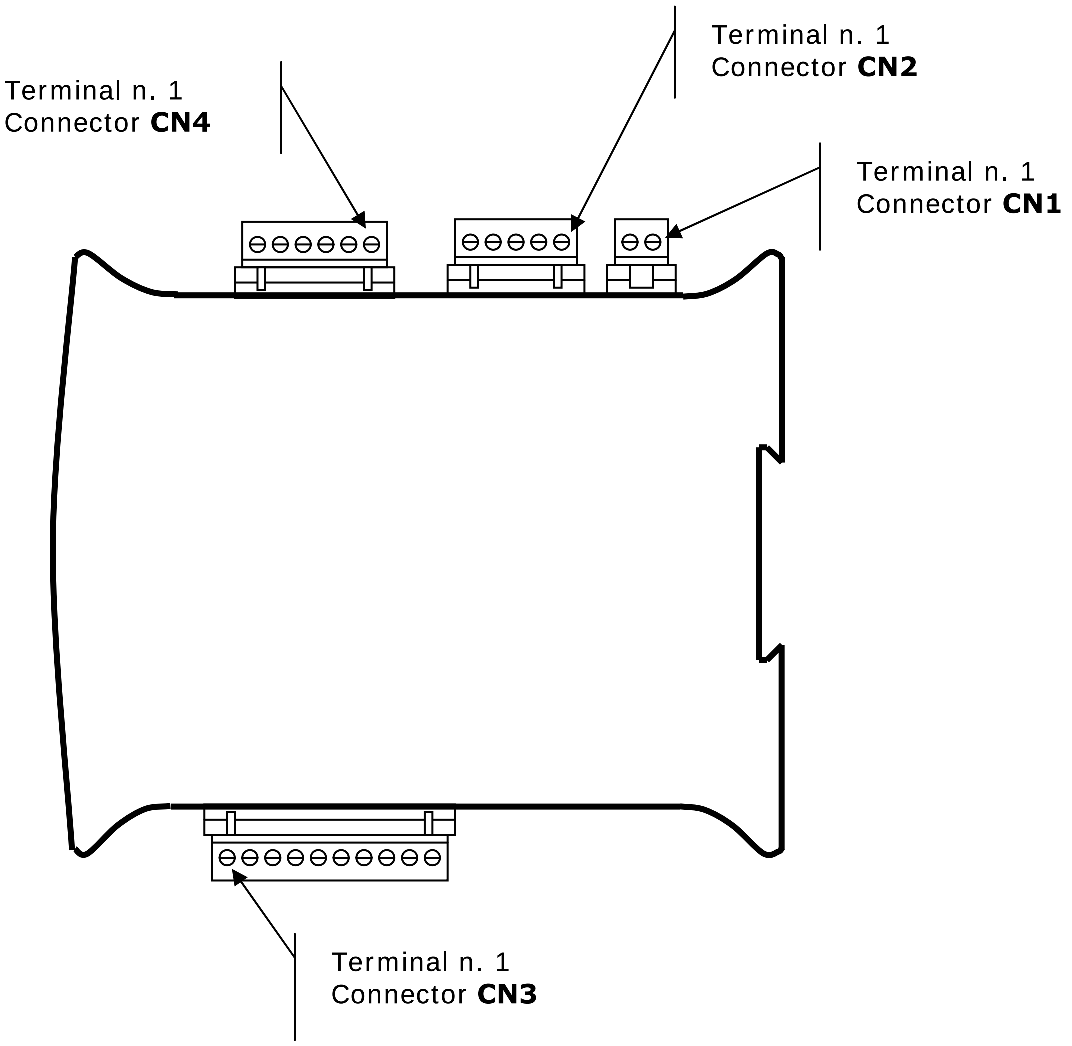
In case of a reduced size fan, it must be set in a rear position and at about 30mm high from the driver. A nearer or more distant position could reduce its effectiveness.



It is important to avoid the driver is covered with powder, dust or other. The buildup of such substances inside the driver could cause the malfunctioning or the breaking of the driver. Filters and necessary solutions must be got ready to avoid them to occur.

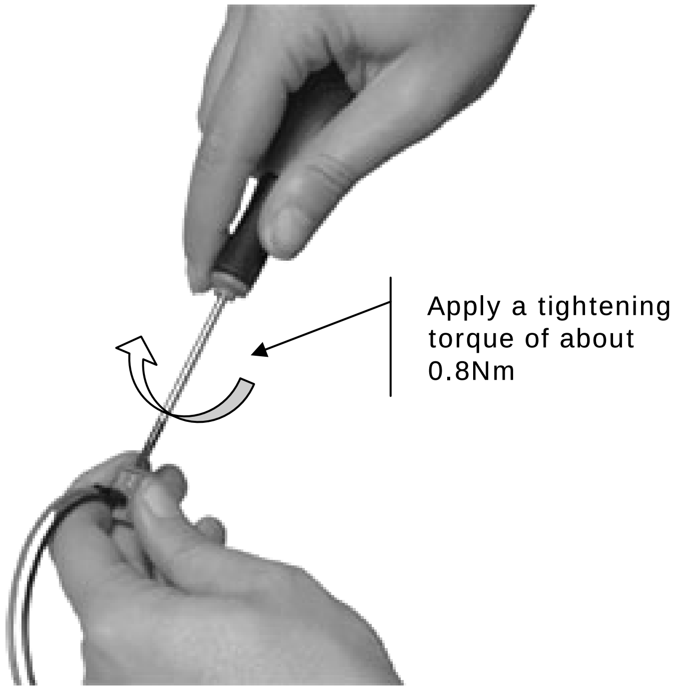
4.4 Wiring

☺ The use of numbered and colored movable terminal blocks makes easier the wiring of the driver.



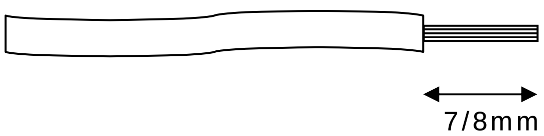
Each terminal block does not contain iron and is supplied with mobile truck. The clamping screw is slotted head sized for screwdriver of 3 x 0.6mm.

We recommend to apply a tightening torque of about 0.8Nm.



All terminals can tighten correctly cables with section between 0.1mm^2 and 2.5mm^2 (24…14 AWG).

We recommend to skin off the cable for 7/8mm as shown in the figure below.

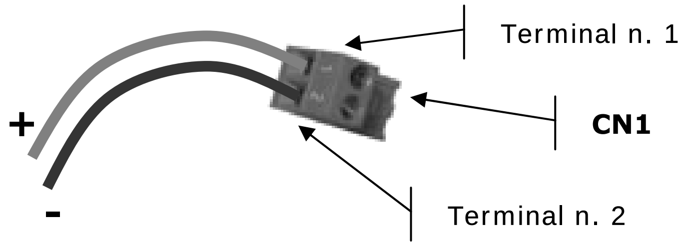


4.4.1 Power Supply

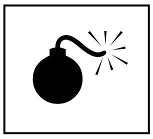
The driver needs a DC supply *Vp* voltage as specified in the table below.
In the highest absorption conditions the power supply must grant a ripple voltage inferior to *Vprp*.

Symbol	Description		Value			Unit
			Min	Typ	Max	
Vp	Nominal DC supply voltage	**DSxx44, DSxx48**	**20**		**50**	**Vdc**
		DSxx73, DSxx76, DSxx78	**24**		**90**	**Vdc**
		DSxx84 DSxx87	**45**		**160**	**Vdc**
		DSxx98	**45**		**240**	**Vdc**
Vprp	Allowed ripple voltage	**DSxx44, DSxx48**			**8**	**Vpp**
		DSxx73, DSxx76, DSxx78			**15**	**Vpp**
		DSxx84 DSxx87			**25**	**Vpp**
		DSxx98			**30**	**Vpp**
Vpbrk	Voltage causing permanent damage	**DSxx44, DSxx48**	**-0.5**		**58**	**Vdc**
		DSxx73, DSxx76, DSxx78	**-0.5**		**105**	**Vdc**
		DSxx84 DSxx87	**-0.5**		**210**	**Vdc**
		DSxx98	**-0.5**		**265**	**Vdc**
Vph	Over voltage protection intervention threshold	**DSxx44, DSxx48**	**54**		**56**	**Vdc**
		DSxx73, DSxx76, DSxx78	**96**		**102**	**Vdc**
		DSxx84 DSxx87	**170**		**180**	**Vdc**
		DSxx98	**242**		**255**	**Vdc**
Vpl	Under voltage protection intervention threshold	**DSxx44, DSxx48**	**18**		**19.5**	**Vdc**
		DSxx73, DSxx76, DSxx78	**21**		**23**	**Vdc**
		DSxx84 DSxx87 DSxx98	**41**		**44**	**Vdc**

The power supply is connected by the red colored 2 poles connector. The positive terminal must be connected to terminal 1, while the negative terminal to the terminal 2.



CN1 - Signals set-up	
Contact n.	**Description**
1	+Vp, positive DC supply voltage
2	-Vp (GND), negative DC supply voltage



Reverse polarity connection damages irreversibly the driver, just as the overcoming of the *Vpbrk* voltage limit.
Do not install the driver before the wiring is complete.

If the distance between the driver and the power supply is more than 1 meter, it is necessary to place near the driver (less than 10cm) an electrolytic capacitor whose features are listed below:

Model	C2 Voltage (Vdc)	C2 Capacity (μF)
DSxx44	63	100
DSxx48	63	220
DSxx73	100	100
DSxx76	100	100
DSxx78	100	220
DSxx84	200	100
DSxx87	200	220
DSxx98	250	220

To connect together the power supply, the driver and the eventual local capacitor, it is necessary to use a conductor with section adequate to the driver's current calibration (for safety's it is better to use the max current supplied by the driver).

The power supply cable can be installed together with the ones connecting the driver to the motor. We recommend not to place the power supply cable nearby the signal ones.

There are two types of power supplies commonly used, regulated and unregulated. The regulated power supply maintains a stable output voltage which allows to supply the driver even with voltage values near to the allowed maximum ones, as this power supply compensate the net and load fluctuations avoiding the above voltage protection to intervene. The disadvantage of regulated power supplies is their cost.

An unregulated power supply is cheaper, but it requires the consideration of safety’s tolerance during its sizing so that, in presence of net and load fluctuations, voltage remains however within the acceptable working values.

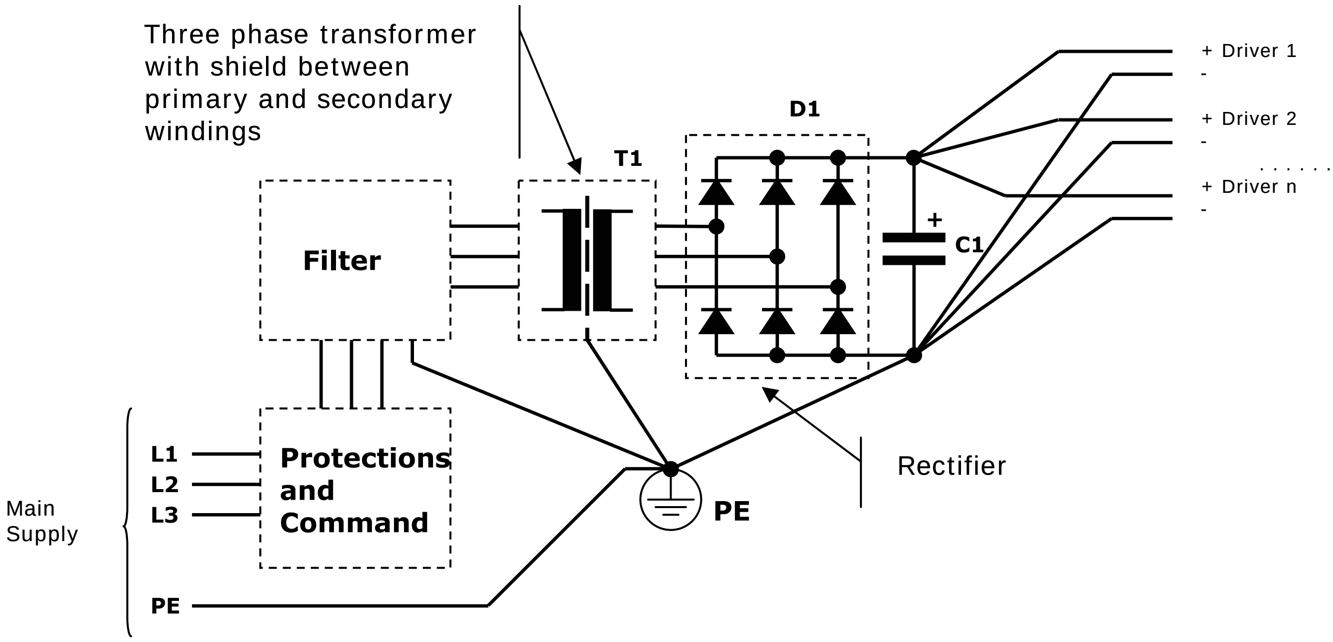
A detailed description about the sizing of the power supply is outside of this manual. The user who decides to construct its own power supply must be technically qualified to size it, to assure its correct working and to fulfill each safety requirements. To determine the power supply output voltage it must be considered the maximum net fluctuation expected on worse operating conditions, the maximum voltage at empty load and minimum voltage at full load, and to assure that the values resulting from these components combination are within the range of max and min. voltage values specified for the chosen driver model.

The potency the power supply must deliver is given by the one absorbed by the load (depending from the torque required to the motor as well as the rotating speed) and by the motor and driver efficiencies. The potency lost on the driver for each model is as shown in the following table.

Model	*Min*	*Typ*	*Max*	*Unit*
DSxx44			10	**W**
DSxx48			20	**W**
DSxx73			10	**W**
DSxx76			20	**W**
DSxx78			30	**W**
DSxx84			30	**W**
DSxx87			35	**W**
DSxx98			40	**W**

Because of the many variables in play it is clear the impossibility to prior indicate an only one kind of dimensioning valid for each application.
Just for example, not to be considered exhaustive and right for the application, see the following electric layout of an unregulated power supply with a brief indication of the components value.

You can also ask to your own vendor to evaluate if the following values are usable for your application too.



As shown in the layout, the earth wiring must be star like, where the earth connection of various components ends in one only point electrically connected with the metal chassis and the plant earth.
Also the wiring towards the drivers must be star like, with the center of the star on C1 capacitor poles.

It is necessary, as per diagram, to place a filter in series to the transformer primary winding to stop emissions coming from the driver and/or present on the main supply. Furthermore, the filter must be able to support the maximum power required by the driver plus the transformer losses.

It is obligatory to provide on every phase a fuse able to intervene in case of a short circuit at the supply output or a malfunctioning of its parts.

The reduction level the filter must guarantee can vary a lot according to the laws applied to the field to which the application and/or installation belong.
The producers of filters SHAFFNER and CORCOM can be a good reference point to find the right filter.

In the following table the characteristic values of the main power supply components are shown. In the calculation an oscillation of the main supply voltage contained within +/-20% has been considered.

Model	Number of drivers	Secondary T1 (Vac)	Power T1 (VA)	Current D1 (Arms)	Voltage C1 (Vdc)	Capacity C1 (μF)
DSxx44	1	30	125	25A	63	3300
	2		250	25A		4700
	3		375	25A		5600
	4..5		600	25A		8200
	6..8		900	35A		10000
DSxx48	1	30	250	25A	63	4700
	2		500	25A		6800
	3		750	25A		8200
	4..5		1100	35A		10000
	6..8		1800	50A		15000
Dxx073	1	50	150	25A	100	1800
	2		300	25A		2200
	3		450	25A		3300
	4..5		700	25A		4700
	6..8		1000	35A		5600
DSxx76	1	50	300	25A	100	2200
	2		600	25A		3300
	3		900	25A		3900
	4..5		1400	35A		4700
	6..8		2100	50A		6800
DSxx78	1	50	400	25A	100	3300
	2		800	25A		4700
	3		1200	25A		5600
	4..5		1800	35A		8200
	6..8		2800	50A		10000
DSxx84	1	90	350	25A	200	1000
	2		700	25A		1500
	3		1050	25A		1800
	4..5		1600	25A		2200
	6..8		2500	35A		3300
DSxx87	1	90	700	25A	200	1500
	2		1400	25A		2200
	3		2000	25A		3300
	4..5		2500	35A		4700
	6..8		5000	50A		6800
DSxx98	1	130	1000	25A	250	1800
	2		2000	25A		2200
	3		3000	35A		3300
	4..5		4500	50A		4700
	6..8		7000	50A		6800

The capacitor C1 values can also be obtained placing more capacitors in parallel amongst them. Eventual approximations must be made in excess.
The user can add in parallel to the C1 capacitor a resistor, opportunely dimensioned, to discharge the capacitor more quickly.

The working voltage of the T1 transformer primary winding must be chosen according to the main supply voltage available during the installation of the application. The transformer must have a shield between primary and secondary windings which must be connected to earth with a short and not inductive connection. The secondary winding voltage is meant without the load with the primary winding supplied at the nominal voltage.

The rectifier, besides supporting the maximum current required by the driver, must be able to tolerate the current supplied during the C1 capacitor charge. Such current, as being essentially limited only by the internal resistor of the transformer secondary winding, usually very low, and by the wiring, can also be of elevated entity, even if of short length (it is exhausted when the capacitor is charged).
Furthermore, the rectifier needs an heat sink able to maintain the temperature within the range defined by the manufacturer. The working voltage of the D1 rectifier must be then chosen according to the T1 transformer secondary winding voltage multiplied at least by 2.

In a configuration with more than a driver, where the drivers have not been all calibrated for the max current and/or where the working cycle is not simultaneous, the transformer power can be considerably reduced. Sometimes this can also be made when the motors speed is limited.

The diagram and the components values refer to a three phase power supply. Differently dimensioning the components it is possible to realize a mono phase power supply, which use is not recommended when the required power is greater than 500W.

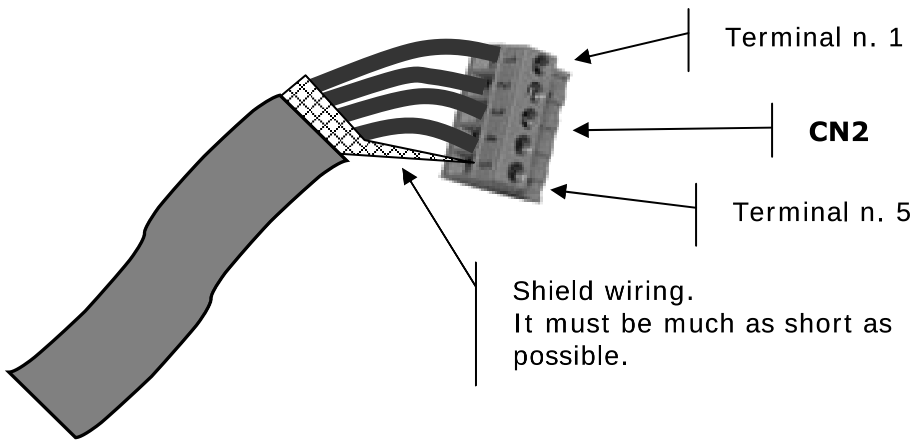
The whole set composed by the filter, the transformer and the power supply must be used only to supply voltage to the drivers. It is not advised to draw other supplies from any of these parts. On the contrary, we suggest to get auxiliary supplies using directly the main supply up the filter.

4.4.2 Motor

The connection between the driver and the motor must be made using shielded cable and connecting the screen to the SHIELD terminal on the driver. The cable shield on the motor side must be connected to the motor body only if this one is electrically insulated from the structure where it is fastened. In case the motor is electrically earth connected (for example through a mechanical fastener) the motor side shield must not be connected.

Only if problems connected to electromagnetic emissions arise it is possible to try and connect the shield from the motor side. Because of possible ground loops which could start, we advise to make this wiring only when strictly necessary.

The driver regulates the current in the motor phases through the supply voltage modulation by chopper technique. The use of a good quality shielded cable and of a correct wiring are essential to better reduce the electromagnetic emissions.



CN2 - Signals set-up	
Contact n.	**Description**
1	FA-, negative output phase A
2	FA+, positive output phase A
3	FB+, positive output phase B
4	FB-, negative output phase B
5	SHIELD (internally connected with a GND)

The cable section can be dimensioned according to the driver current calibration, anyway we suggest to choose a cable suitable for maximum current deliverable from the model of driver itself.

The recommended cable to connect the motor to the driver must have a length inferior to 10mt. For greater cable length, the cable size must be increased to counterbalance the voltage drop.

The cable connecting the driver to the motor can be installed together with the power supply one, but it must be kept separate from the signal cables.

If difficulties are met to pass compatibility tests, it is possible to place in series to each phase an inductor with a value included between 10uH and 100uH and with current adequate to the set phase current. The inductor must be placed directly on the driver output.

4.4.3 Control Signals

The cable used for the control signals wiring must be a shielded type.

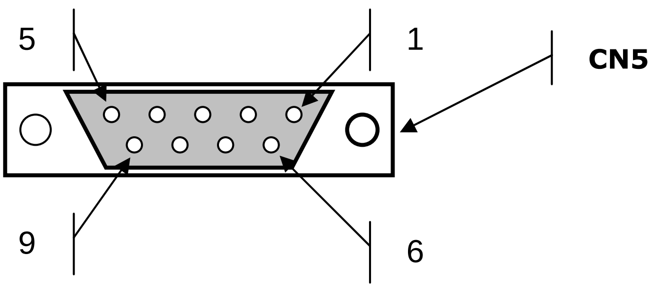
4.4.3.1 Fieldbus (DS5x series)

The information contained in the following chapters apply only to the drives provided with fieldbus (DS50, DS52 and DS54 series).

4.4.3.1.1 DS50 series (RS485 - ModBus RTU)

The DS50 series is equipped with a RS485 type hardware interface, optoisolated from the power stage, and with a Modbus-RTU type communication protocol.

The connection to the bus is through a 9 pin female D-sub connector as required by the Modbus specifications. In the following representation is shown the signals arrangement on the 9 pin connector.



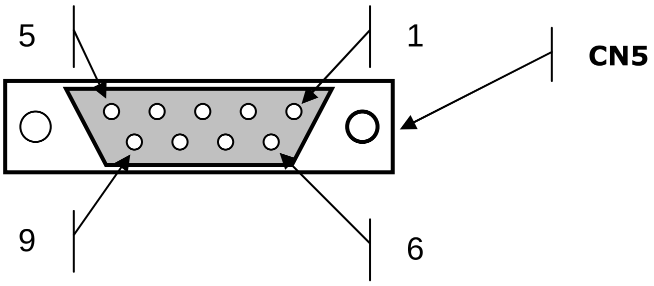
CN5 - Signals disposition	
Contact n.	**Description**
1	signals common, it is the reference of the D0 and D1 signals
2	+5V, voltage referred to the 0V with max current of 10mA
3	not connected
4	not connected
5	D1, data line also called B/B'
6	not connected
7	not connected
8	not connected
9	D0, data line also called A/A'

The connections 1 and 2 can be used to connect the RS485 polarization resistors if needed. Keep in mind that the max source current from the pin 2 is 10mA.

4.4.3.1.2 DS52 series (RS232 - Modbus RTU)

The DS52 series is equipped with a RS232 hardware interface, optoisolated from the power stage, and with a Modbus-RTU type communication protocol.

The connection to the bus is through a 9 pin female D-sub connector as required by the Modbus specifications. In the following representation is shown the signals disposition on the 9 pin connector.



CN5 - Signals disposition	
Contact n.	**Description**
1	**not connected**
2	**TXD, data transmitted from the driver**
3	**RXD, data transmitted from the driver**
4	**not connected**
5	**signals common, it is the reference (0V) of the TXD and RXD signals**
6	**not connected**
7	**not connected**
8	**not connected**
9	**not connected**

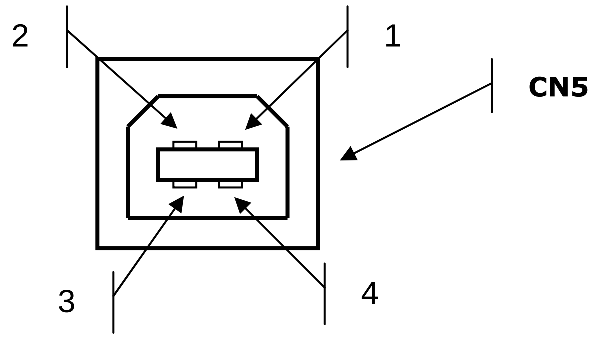
For the connection to the master we suggest to use a shielded cable with a length not superior to 20m.

4.4.3.1.3 DS54 series (USB - Modbus RTU)

The DS54 series is equipped with an USB Full-Speed type hardware interface, optoisolated from the power stage, and with a Modbus-RTU type communication protocol.

The USB interface is particularly suitable to control the driver through a PC. The bundled driver software allows to emulate a standard system serial port. This solution facilitates the writing of the communication program with the driver which will be able to make use of all the provided standard functions to communicate through a system serial port. The drivers currently supplied are for operating systems such as Windows 98, 98SE, Me, 2K, XP and Vista.

The connection is through a standard female USB-B connector. In the following representation is shown the signals disposition on the USB-B connector.

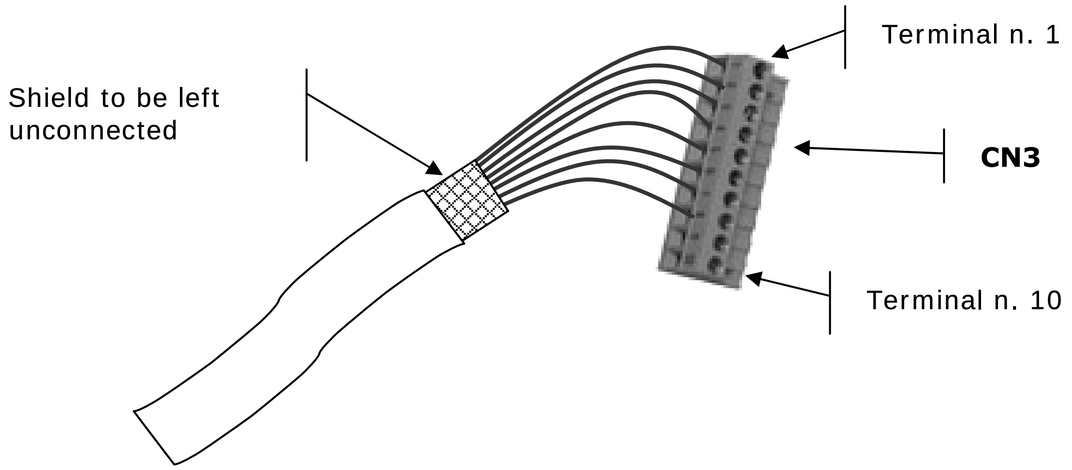


CN5 - Signals disposition	
Contact n.	**Description**
1	**Vbus, positive power supply coming from the PC**
2	**D-, negate data line**
3	**D+, data line**
4	**ground, 0V reference for the signals and for Vbus**

For the connection to the PC we suggest to use a good quality shielded cable with a length not superior to 5m.

4.4.3.2 Digital inputs

The driver is provided with a total of 4 digital inputs referring to the CN3 10 poles connector (two poles are reserved to a digital output). The shielding of the connection cable must be connected to the numerical control device side (PC, PLC or other), while on the driver side it must remain unconnected as shown in the figure below.



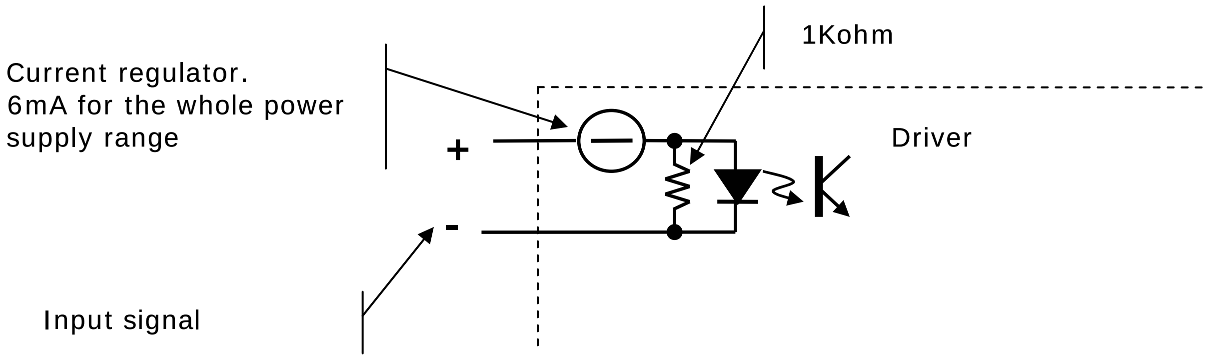
CN3 - Signals	
Contact N.	**Description**
1	**DI0+ DigitalInput(0), positive 0 digital input**
2	**DI0- DigitalInput(0), negative 0 digital input**
3	**DI1+ DigitalInput(1), positive 1 digital input**
4	**DI1- DigitalInput(1), negative 1 digital input**
5	**DI2+ DigitalInput(2), positive 2 digital input**
6	**DI2- DigitalInput(2), negative 2 digital input**
7	**DI3+ DigitalInput(3), positive 3 digital input**
8	**DI3- DigitalInput(3), negative 3 digital input**
9	DO0+ DigitalOutput(0), positive 0 digital output
10	DO0- DigitalOutput(0), negative 0 digital output

The section of the cable does not have much importance as the circulating currents are meager (in the order of few mA), therefore choose the section according to wiring functionality.

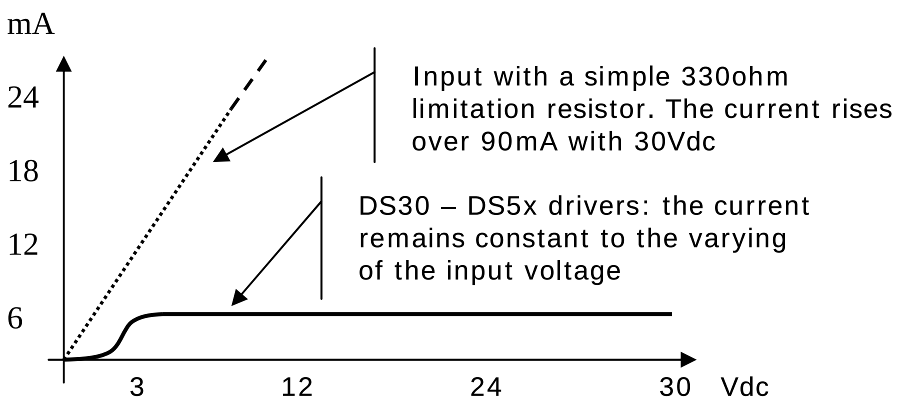
The control signal cable must not be placed together with the power supply or the motor cables. If it happens there is the possibility that the high slew rate voltage and high rate current could be coupled with the logic level signal and corrupt this.

The DS30 – DS5x drivers digital inputs can be independently connected in NPN or PNP logic. Each input exposes both the connections making possible mixed settings.

All inputs and outputs are optocoupled amongst them, towards the motor and towards the power supply.



The DS30 – DS5x drivers have on each input a particular current limitation circuit which grants a constant absorption to the varying of the applied voltage. This device allows a wide input power supply range (from 3Vdc up to 30Vdc) without requiring excessive current to the control equipment.
The most part of drivers now on the market, of the same category, uses to place in series to the input a simple current limitation resistor which reduces the input working voltage setting and causes, moreover, high current values when the input voltage is high.. The DS30 – DS5x drivers, on the contrary, maintain an almost constant absorption on the whole working range, as shown in the diagram below.



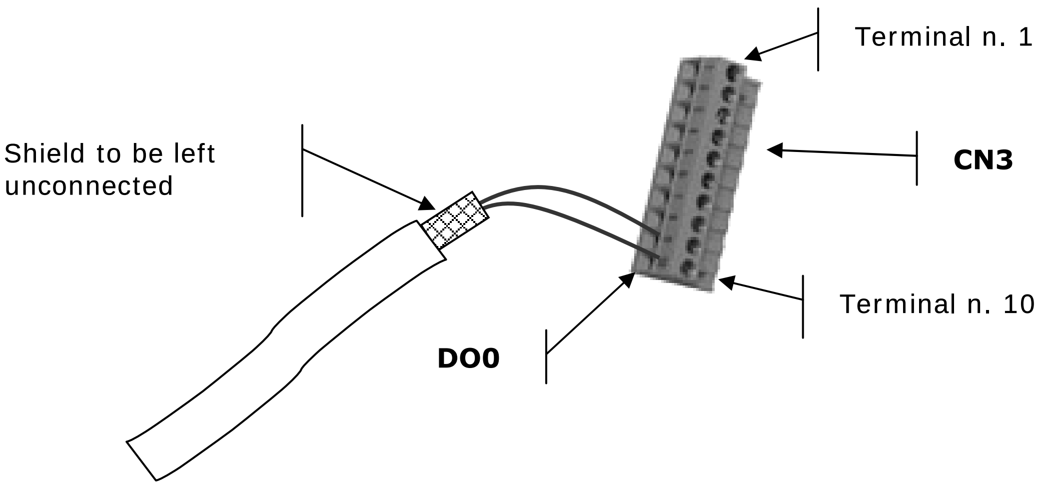
☺ Each single input can be used on line driving mode independently from the others.

The following table resumes the electrical characteristics of the digital inputs.

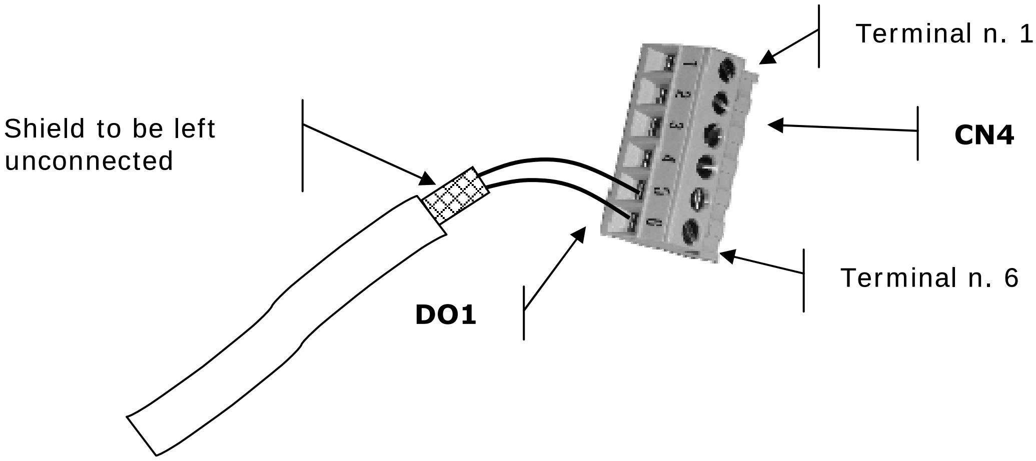
Symbol	Description	Value			Unit
		Min	Typ	Max	
Vdi	Active digital inputs voltage	3		30	**Vdc**
Vdioff	Inactive digital inputs voltage	-30		1	**Vdc**
Vdibrk	Digital inputs breakdown voltage	-35		+35	**Vdc**
Idi	Current absorbed by the digital inputs	4	6	8	**mA**

4.4.3.3 Digital outputs

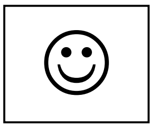
The driver is provided with a total of two digital outputs. The DO0 outputs is available on the CN3 connector while the DO1 one on the CN4 connector as shown in the figures below. The shielding of the connection cable must be connected to the numerical control device side (PC, PLC or other) while on the driver side it must remain unconnected.



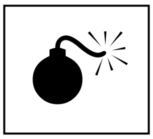
CN3 - Signals	
Contact N.	**Description**
1	DI0+ DigitalInput(0), positive 0 digital input
2	DI0- DigitalInput(0), negative 0 digital input
3	DI1+ DigitalInput(1), positive 1 digital input
4	DI1- DigitalInput(1), negative 1 digital input
5	DI2+ DigitalInput(2), positive 2 digital input
6	DI2- DigitalInput(2), negative 2 digital input
7	DI3+ DigitalInput(3), positive 3 digital input
8	DI3- DigitalInput(3), negative 3 digital input
9	**DO0+ DigitalOutput(0), positive 0 digital ouput**
10	**DO0- DigitalOutput(0), negative 0 digital ouput**



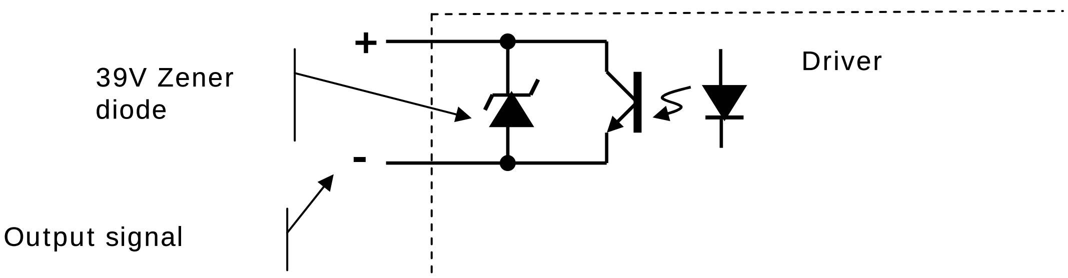
CN4 - Signals	
Contact N.	**Description**
1	GND, analog signals ground (internally connected with power round)
2	AI0, 0 analog input
3	AI0, 1 analog input
4	AO0, 0 analog output
5	**DO1+ DigitalOutput(1), positive 1 digital output**
6	**DO1- DigitalOutput(1), negative 1 digital output**



The outputs have a zener diode placed in parallel which allows to connect the output with medium entity loads (for example signal relays) without having to use an external recirculating diode.



The outputs are protected from a short length short circuit, a longer one can damage them permanently.



The following table resumes the electrical characteristics of the digital outputs

Symbol	***Description***	***Value***			***Unit***
		Min	***Typ***	***Max***	
Vdo	Digital outputs operating voltage	1	24	30	**Vdc**
Vdobrk	Digital outputs breakdown voltage	-0.5		42	**Vdc**
Vdoz	Zener diode voltage placed in parallel to each output	37	39	42	**Vdc**
Ido	Digital outputs available current			80	**mA**
Idobrk	Digital outputs breakdown current	120			**mA**
Pwdo	Digital outputs dissipable power			400	**mW**

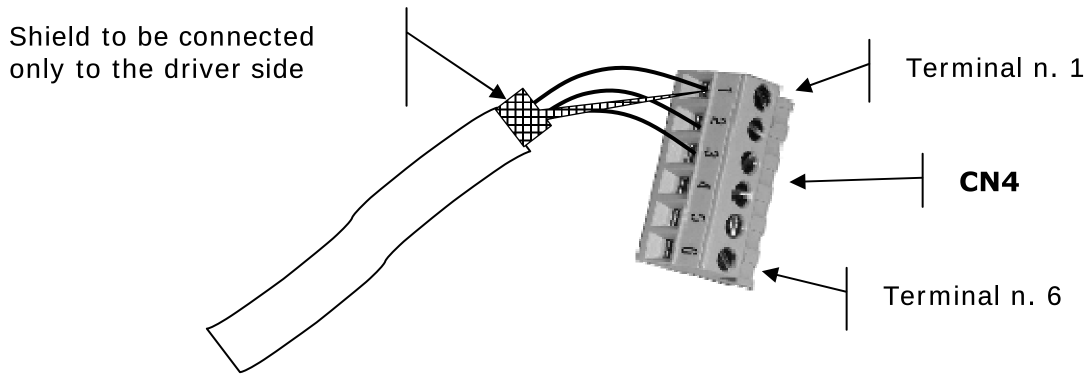
4.4.3.4 Analog inputs

The DS30 – DS5x series drivers are provided with two analog inputs able to measure voltages inside the range of +/- 10V.

☺

The breakdown voltage superior to 30Vdc allows to use the analog inputs as digital inputs as well.

The connection to the analog inputs must be through a shielded cable, having care to connect the shield only to the driver side as shown in the following figure.



CN4 – Signals	
N. contatto	**Descrizione**
1	**GND, analog signals ground (internally connected with power ground)**
2	**AI0, 0 analog input**
3	**AI0, 1 analog input**
4	AO0, 0 analog output
5	DO1+ DigitalOutput(1), positive 1 digital output
6	DO1- DigitalOutput(1), negative 1 digital output

The following table resumes the electrical characteristics of the analog inputs and the A/D converter inside the driver.

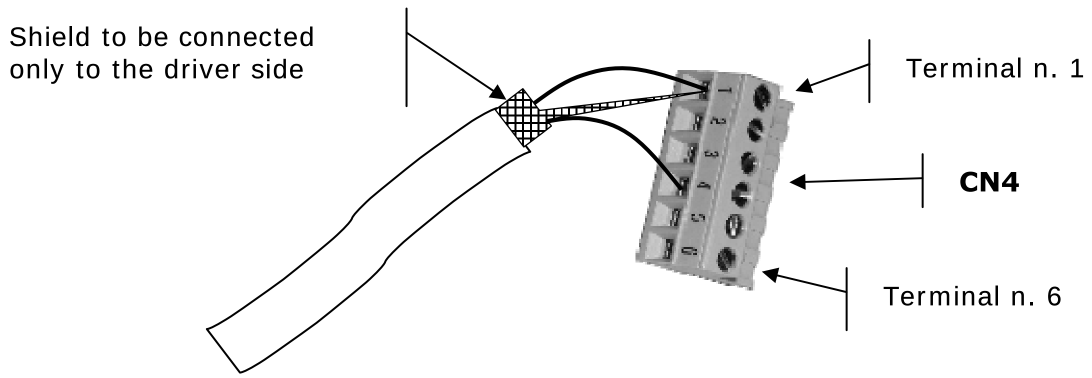
Symbol	*Description*	*Value*			*Unit*
		Min	*Typ*	*Max*	
Vai	Analog inputs operating voltage	-10		+10	**Vdc**
Vaibrk	Analog input breakdown voltage	-45		+45	**Vdc**
Rai	Analog inputs impedance		47		**KΩ**
ADrai	A/D converter resolution		11		**bit**
ADst	A/D converter converting time		10		**msec**
ADsoff	A/D converter start offset		1	2	**%fs**
ADdoff	A/D converter offset drift		0.3	0.5	**%fs**
ADline	A/D converter linearity error		1	2	**%fs**

When the driver is supplied only through the UDP30 setting interface the analog inputs are not correctly converted.

4.4.3.5 Analog outputs

As a completion of the I/O signals set, the DS30 – DS5x series drivers are also provided with an analog output able to supply a voltage between 0 e 10V.

The connection to the analog output must be through a shielded cable, having care to connect the shield only to the driver side as shown in the following figure.



CN4 - Signals	
Contatto N.	**Description**
1	**GND, analog signals ground (internally connected with power round)**
2	AI0, 0 analog input
3	AI0, 1 analog input
4	**AO0, 0 analog output**
5	DO1+ DigitalOutput(1), positive 1 digital output
6	DO1- DigitalOutput(1), negative 1 digital output

The following table resumes the electrical characteristics of the analog output and the D/A converter inside the driver.

Syimbol	Description	Value			Unit
		Min	Typ	Max	
Vao	Analog outputs operating voltage	0		+10	**Vdc**
Iao	Analog outputs operating current			10	**mA**
Iasc	Analog outputs short circuit current			20	**mA**
Rao	Analog outputs output impedance			1	**Ω**
DArai	D/A converter resolution	11			**bit**
DAud	D/A converter update		10		**msec**
DAsoff	D/A converter start offset		0.3	1	**%fs**
DAdoff	D/A coverter offset drift		0.1	0.3	**%fs**
DAline	D/A coverter linearity error		0.3	1.5	**%fs**

When the driver is supplied only through the UDP30 setting interface the analog output voltage is undetermined.

4.4.4 Chassis setting

In order to contain the electromagnetic emissions and to better shield the driver, it is necessary to give particular attention to the setting of the chassis.
The driver must be placed inside a metal case, preferably iron made, capable to successfully shield the electromagnetic emissions. The case must be electrically placed to ground as better described onwards.

Filter

The filter must be correctly placed in series to the main supply. The ideal position is on the chassis edge in order to have a short wiring coming out from the main supply. If, on the contrary, the main supply cables run inside the chassis, they can be invested by electromagnetic interferences making ineffective or much reducing the filter efficiency.
The filter output earth must be connected to the metallic body of the chassis. It is important the connection is of short length and made with a large section and low inductance conductor. The point of connection between ground coming from the filter and the chassis constitutes the star center to which all other components earth must be connected. Moreover, the filter metallic body must be electrically placed in contact with the case.

Transformer

The transformer must be placed close to the filter and must have a shield between the primary and the secondary windings. The shield must be ground connected in the star center obtained inside the case. Furthermore, the transformer metallic body must be electrically connected to the case.

Power Supply

It is advisable to install the power supply near the transformer. The power supply earth, usually the negate pole of the filter capacitor, must be connected to ground in the star center obtained inside the case.

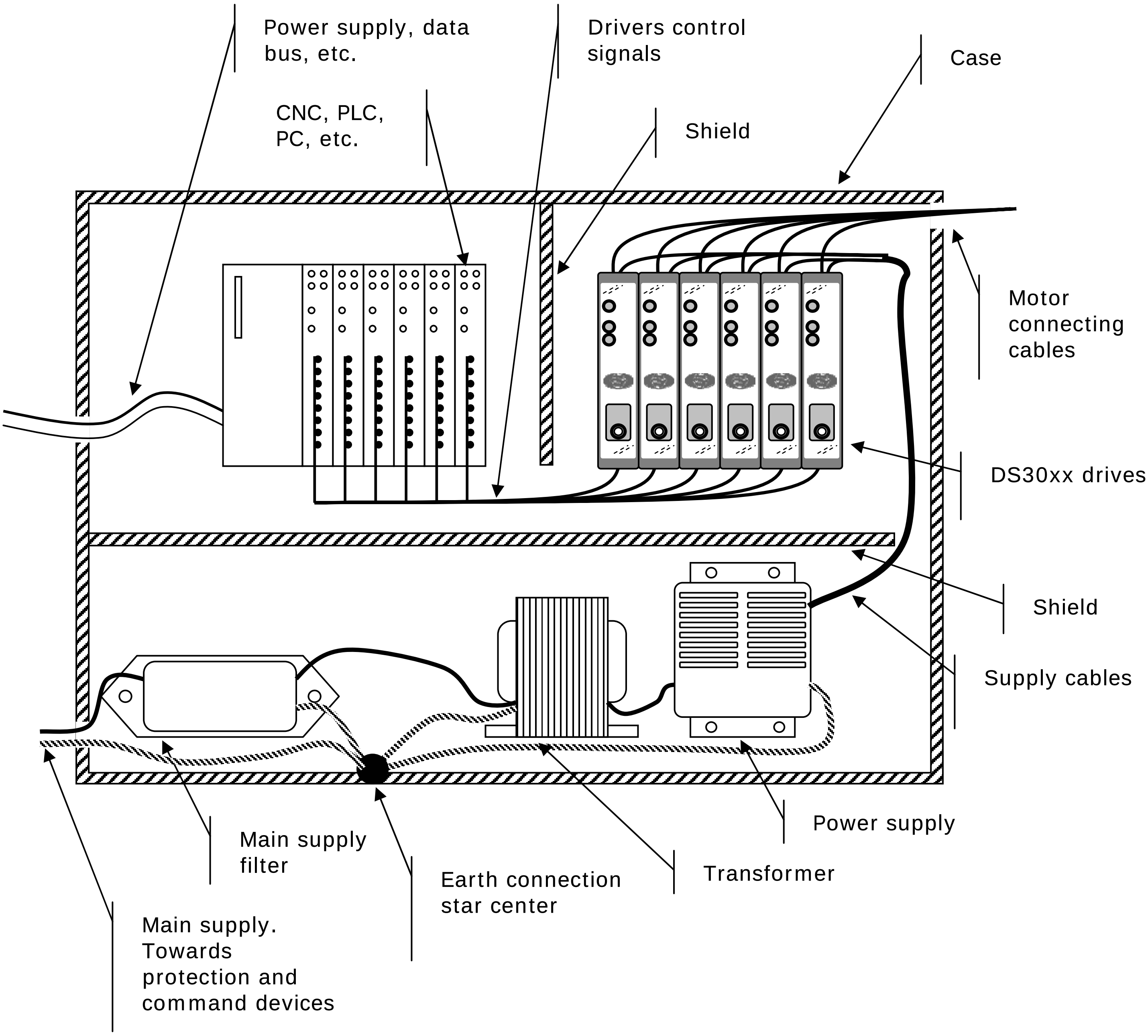
Driver

The driver position must be chosen in order the motor cables can immediately come out from the case without running long distances inside the case itself.

Numerical Control

The numerical control device, PLC or other, which generates the driving signals of the driver must be as far as possible from the drivers and from the power supply group. Moreover, the signals wiring must be remain distant from the power supply and motor cables. When the distance from the numerical control device and the driver and/or the power supply is short, there must be one or more shielding walls, electrically connected to the case.

The following figure shows a possible setting of the chassis.



Safety must never be compromised. Safety is always of first priority.

5 Functionality and configuration

5.1 Application notes

5.1.1 Phase current

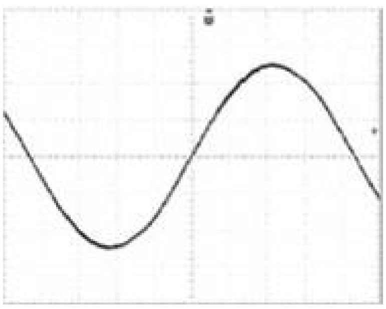
It is necessary to pay more attention to the phase current setting as a wrong value can cause the break of the motor.

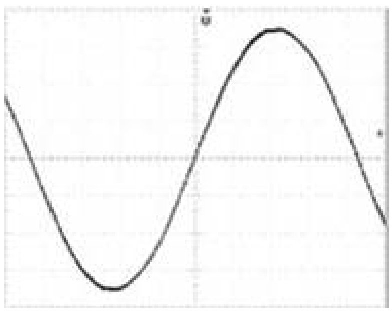
It is important to note that in the DS30 – DS5x series the current values are expressed as effective current and not as peak current. Some producers indicate the peak current. To obtain the effective current value it is sufficient to divide the peak current by 1.41 (this calculation is not applicable in case of full step or not sinusoid currents given to the motor).

☺

From what above explained it is clear that a 4Arms driver (4A effective current) is equivalent to a driver with peak current equal to 5.6Ap (4 x 1.41). Do consider this while comparing the features of the DS30 – DS5x drivers with other producers devices.

The following figures show the phase current supplied by the DS3048 driver calibrated to 5Arms (right figure) and the current of a competitor's driver calibrated to 5Apk (left figure).





☺

It appears evident how the DS30 – DS5x drivers (which characterizes the current to the effective Arms value) supply greater current than those drivers which characterize the phase current as peak value.

The current calibration must be mainly executed according to the motor features. In particular cases it is possible to use an higher current value than the nominal one declared by the motor manufacturer. This could be useful to obtain more torque from the motor, but this method leads to an higher motor heating. If motor temperature exceeds the maximum allowed values the motor could be permanently damaged. Usually, the motor is over supplied only if the working cycle is soft and therefore allows a medium working temperature within the maximum allowed values declared by the manufacturer.

Because of the saturation phenomena of the magnetic circuit inside the motor (which vary from model to model) there is not a direct correspondence between torque and phase current when the nominal value is exceeded. In other words, even doubling the current, a double torque is never obtained from a motor. For this reason it is usually superfluous to over supply the motor of more than 30% of the nominal current.

If the motor works in a worm place or without a mechanic device capable to discharge the heat, it is possible it reaches high temperatures even if supplied by the nominal current. In this case it is necessary to introduce a forced ventilation on the motor or, if the torque margins allow it, to reduce the current calibration on the driver.
Remember that in this case the benefit obtained in terms of temperature will go at the expense of the torque supplied by the motor. Moreover, consider that when the driver calibration current does not correspond to the motor nominal current, the microstepping movement can lose regularity. In the same way, sometimes it is possible to improve the microstepping movement by slightly modifying the current calibration.

Typically a two phase stepping motor disposes of four wires, two per each phase. In this case the driver current calibration must correspond to the motor current plate. If for example the motor quotes 3A/phase also the driver shall be to be calibrated to 3Arms.

Some motors, supplied with more windings, allow to choose between series and parallel connections.

The series configuration requires a lower phase current but it also exposes a superior inductance towards the driver which penalizes the motor torque at high speeds. Therefore, this kind of connection is used in applications where motor speed is limited or supply voltage sufficiently high.
When a series configuration is used it is necessary to calibrate the driver to the motor phase current multiplied by the coefficient 0.71. If for example we are using a motor with plate current 3A, with the windings connected in series we should calibrate the driver to a current of 2.1Arms (3 * 0.71).
Furthermore take note that the inductance exposed by the motor to the driver is 2 times the single phase one.

On the contrary, the parallel configuration requires an higher phase current but it has the advantage to maintain the torque supplied by the motor more constant to the increasing of the speed. It happens because the inductance exposed by the parallel connection is inferior. This connection is preferable when the supply voltage is low or the motor working speed is high.
When a parallel configuration is used it is necessary to calibrate the driver to the motor phase current multiplied by the coefficient 1.41. If for example we are using a motor with plate current 3A, with the windings connected in parallel we should calibrate the driver to a current of 4.2Arms (3 * 1.41).
Furthermore take note that the inductance exposed by the motor to the driver is in this case the same as the single phase one.

5.1.2 Automatic current reduction

Stepper motor drivers works at impressed current, i.e. they always supply the motor at the set nominal current independently from the motor giving torque to the load or not.
In other types of motor, as for example in DC or Brushless motors, the driver supplies the motor with the current strictly necessary to maintain the required position or rotation speed. In these cases the driver is able to obtain the information on the position or speed errors through transducers such as tachogenerator, encoder, resolver, etc. Therefore, such a system disposes of a feedback and it is a reactioned or closed loop kind.

On the contrary, the step motors can be used at open-loop without the help of any transducer, also granting constant speeds and precise and repeatable positioning.

What makes this operating method possible is the particular motor and driver conformation. As being without feedback, it always delivers nominal phase current to the motor which therefore always produces the maximum torque.

When the application is dimensioned in such a way that the requested load torque remains inferior to the one supplied by the motor (on the whole required speed range), the speed and the position are preserved, even without feedback.

When the stepper motor is firm it is often necessary to grant a maintaining torque capable to keep the load in stable position (for example a suspended load), for this reason the driver supplies current to the motor even when it is motionless. However, in many cases the current sufficient to grant this condition is inferior to the required operating value; this happens mainly for two reasons: when the motor is firm there are no inertial loads which develop during speed changes, furthermore the torque / speed curve of the motor reaches its utmost just in proximity of zero speed (motionless motor).

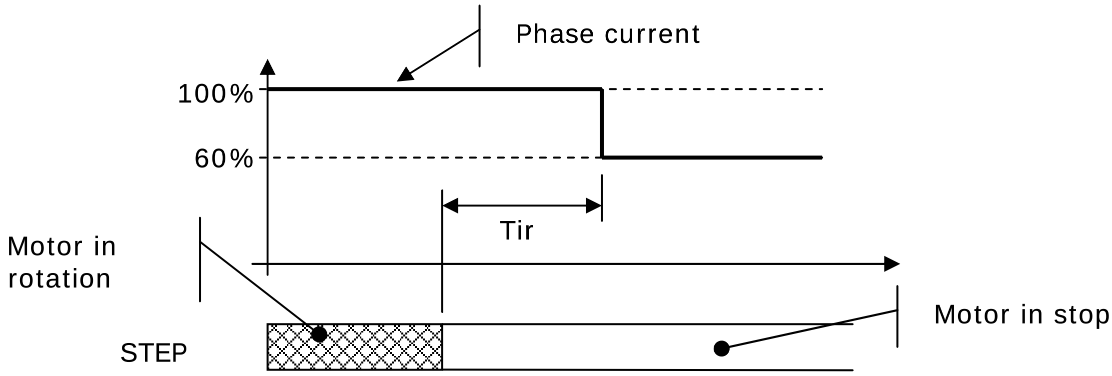
In order to limit the motor and driver heating there is a function able to automatically reduce the current supplied to the motor while in still position.

The DS30 – DS5x drivers allow to define accurately both the current reduction percentage and the timing from the motor stop to the reduction intervention. The regulation dynamics is extraordinarily wide and allows to change the current between 0% and 100% of the nominal current and the timing from few milliseconds up to 10 seconds. Such a flexibility can hardly be found in other drivers of the same class.
Setting a value equal to 0% the current is completely set off when motor is firm, while setting 100% the current is always maintained at the nominal value.

The current reduction percentage must be set considering the real torque required by the application when the motor is firm, while the second parameter (the timing) must be set according to the time that the load takes to stable after the stop of the motor or to the application working cycle.

Observe that removing the current reduction with motor at rest (that is to say setting the 100% value) the motor and the driver heat more. It is therefore suggested to make this calibration only if the application requires an high static torque to the motor.

The following diagram shows the relation between the motor cycle and the automatic current reduction.



After the stop of the motor the *Tir* time starts to pass, after which the automatic current reduction intervenes to reduce the phase current, in the example, to the 60% of the nominal value.

5.2 Alarms and Protections conditioning

In order to protect the driver and to make easier the location of the most common functioning or setting anomalies, the DS30 – DS5x drivers are equipped with many alarms and a complete diagnostics.

Furthermore, the extraordinary flexibility offered by the product allows to singularly configure the operative mode of each alarm choosing among *Automatic, Permanent, Enable* or *Disable.*
The configuration is made without the necessity to intervene on the driver’s hardware, without jumper, dip-switch, etc.

Automatic

The driver constantly examines the alarm condition and when this disappears it provides autonomously to remove the signal and to turn back to the operative status.

Permanent

Each alarm condition remains in the driver memory. To remove the signal the driver must be switched off and then switched on again.

Enable

The breakdown condition signal remains till the driver is not disabled through the *bEnable* boolean register.

Disable

The alarms are disabled and ignored by the driver. For security reasons some protections cannot be put in this status.

In the following table are resumed the various possible settings for each single alarm.

Alarm type / protection	Possible setting			
	Automatic	Permanent	Enable	Disable
Under voltage	Yes	Yes	Yes	No
Over voltage	Yes	Yes	Yes	No
Over temperature	Yes	Yes	Yes	No
Phase-to-phase short circuit	Yes	Yes	Yes	No
Phase-to-ground short circuit (GND)	Yes	Yes	Yes	No
Phase-Vp short circuit	Yes	Yes	Yes	No
Interrupted Phase A	Yes	Yes	Yes	Yes
Interrupted Phase B	Yes	Yes	Yes	Yes

For a complete description of protections and alarms see chapter 5.4 Protections.

5.3 Signal LEDs

The driver has three signal LEDs: the green one called *On*, the yellow LED called *Enable* / *Step* and the red one called *Fault*.

On

The LED called *On* lights up when the driver is supplied with a voltage sufficient to allow a correct functioning of the control electronics.
When the LED is on, the driver is able to elaborate the information but it is not necessarily in the working status (for examples, because the supply voltage is beyond the working limits or the temperature is excessive).

Enable / Run

This LED supplies at the same time various information on the status of the driver.
When the motor is enabled the LED is fixed lighted up, when the motor is disabled the LED is off.

When the motor is rotating the LED status is for a little while inverted. In other words, with the motor is enabled and rotating this LED is mainly lighted up with short power-offs.

Relation between the yellow LED and the status of the motor		
Rotation	**Enable**	**Yellow LED status**
No	No	On Off
No	Yes	On Off
Yes	No	On Off 
Yes	Yes	On Off 

Fault

This LED lights up each time there is at least an alarm signaling.
When the LED is lighted up, the driver is not operative and the motor is without supply (observe that also in this condition the yellow LED continues in its own activity as described in the previous chapter).

☺ In presence of an alarm the red LED lights up and starts a sequence of flashing related to the kind of problem found.
The following table shows the association between the number of flashes and the error found.

Codification of errors signaled by red LED	
Number of flashes	**Problem description**
1	Under voltage, the supply voltage is inferior to *Vpl* value
2	Over voltage, the supply voltage is superior to *Vph* value
3	Over temperature, the heat sink voltage is superior to *Tchh*
4	Phase-to-phase short circuit, one or two phases are in short circuit
5	Phase-to-ground short circuit (GND), phase in short circuit with ground
6	Phase-Vp short circuit, one phase is in short circuit with power supply
7	Interrupted phase A, connection between driver and motor is missing
8	Interrupted phase B, connection between driver and motor is missing

If more than a problem occurs at the same time the lighting flashes sequences associated to each one of them are cyclically executed. If for example the over temperature condition occurs together with the over voltage one, the LED will flash twice and then three times alternatively till when the malfunction signaling will be removed.

The driver features grant at least a visualization cycle for each activated alarm. For this reason it is not possible to remove an alarm before the first visualization cycle is finished.

Note that the error signals can be removed in different ways according to how the driver's setting has been executed (see chapter 5.2 Alarms and Protections conditioning).

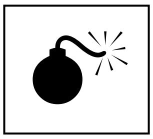
For a detailed description on the various alarms and protections see chapter 5.4 Protections.

When in the driver the *Loader* (an always resident small software necessary for the firmware update) is active, the *Fault* LED remains always on.

5.4 Protections

5.4.1 General description

The driver is equipped with efficient protections which protect its integrity whenever one of the most common problems occurs. Furthermore, some alarms have been implemented; they can detect positioning errors even before the motor starts to move.



In spite of the care and attention used in the development and manufacturing of the driver, an installation or an use not in compliance with the indications present in this manual, or out of stated maximum limits, can damage the driver permanently.

Through the configuration it is possible to define the driver's behaviour at the occurrence of the various alarm conditions.
For example, it is possible to choose to make an alarm permanent or to make the driver re-enable automatically at the end of the alarm condition. For a detailed description about the various setting options see chapter 5.2 Alarms and Protections conditioning.

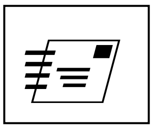
When there is at least an alarm signal the driver is not active and the motor is not supplied, consequently the torque is null.

5.4.2 Under voltage

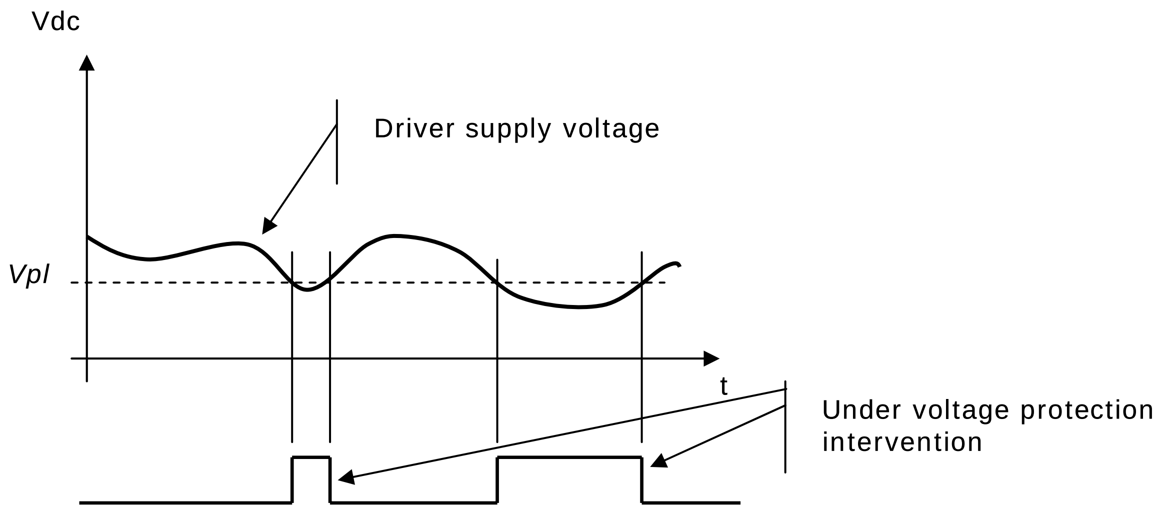
The under voltage alarm intervenes when the driver supply voltage is inferior to the *Vpl* value. Such value varies according to the driver model as per the following table.

Model	**Vpl Value**			***Unit***
	Min	***Typ***	***Max***	
DSxx44		20		**Vdc**
DSxx48		20		**Vdc**
DSxx73		24		**Vdc**
DSxx76		24		**Vdc**
DSxx78		24		**Vdc**
DSxx84		45		**Vdc**
DSxx87		45		**Vdc**
DSxx98		45		**Vdc**

The driver constantly verifies the supply voltage value; it is sufficient this goes beyond the *Vpl* threshold for few instants to generate the under voltage alarm. It does not have to be astonishing the fact that the protection intervenes despite the measured voltage is within the functioning limits, as it is possible that, because of the main supply fluctuations or cables length, the voltage that effectively reaches the driver becomes, in particular moments (for example during the motor acceleration phase), inferior to *Vpl* value.



In order to make a correct measurement it is necessary to act directly on the driver supply terminal blocks using a band-pass instrument of at least 10KHz (as for example an oscilloscope) able to memorize the voltage transients minimum values.



If the supply voltage is very near to the driver functioning limit and sporadically the under voltage alarm intervenes, it is possible, in some cases, to solve the problem reducing the distance between the power supply and the driver, increasing the cables section or placing an electrolytic capacitor near the driver itself.

5.4.3 Over voltage

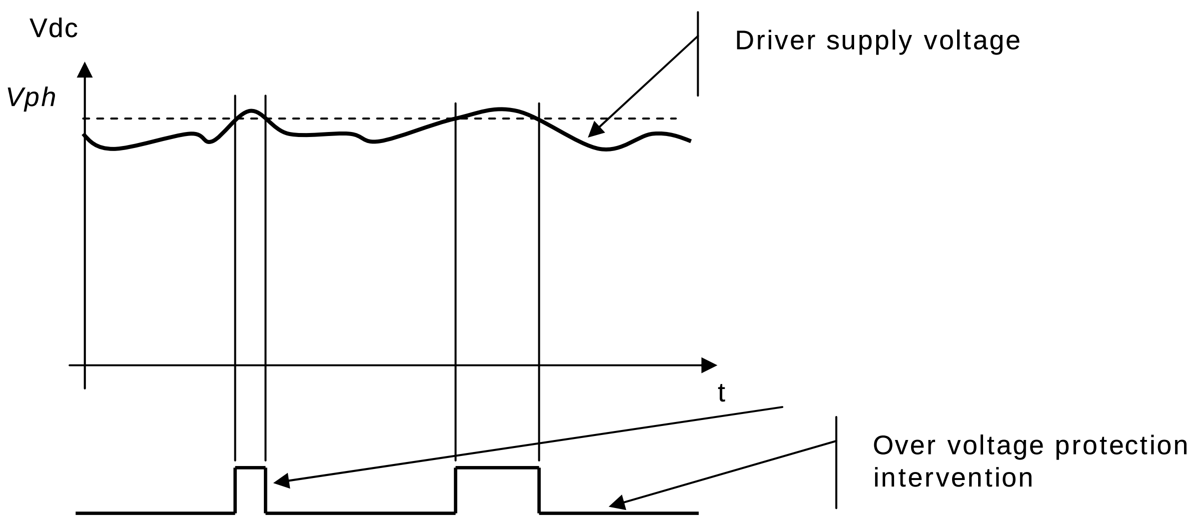
The over voltage protection intervenes when the supply voltage is superior to the *Vph* value. In this condition the driver protects the power stage turning it off. Such value varies according to the driver model, as per the following table.

Model	**Vph value**			***Unit***
	Min	***Typ***	***Max***	
DSxx44		55		**Vdc**
DSxx48		55		**Vdc**
DSxx73		98		**Vdc**
DSxx76		98		**Vdc**
DSxx78		98		**Vdc**
DSxx84		175		**Vdc**
DSxx87		175		**Vdc**
DSxx98		248		**Vdc**

The driver constantly controls the supply voltage and when it goes beyond the *Vph* value, even also for few instants, the over voltage protection intervenes.

It does not have to be astonishing the fact that the protection intervenes despite the measured voltage is within driver functioning limits. In fact it is possible that, in some occasions (sudden motor decelerations, sudden voltage changes, etc.) the voltage which supplies the driver goes beyond the *Vph* value causing the protection intervention.

In order to make a correct measurement of the effective driver supply voltage it is necessary to act directly on the supply terminal blocks of the driver itself, using a band-pass instrument of at least 10KHz and able to capture the voltage transients peaks.



If the supply voltage is very near to the driver functioning limit and the over voltage alarm sporadically intervenes, it is possible in some cases to solve the problem reducing the distance between the power supply and the driver, increasing the cables section or placing an electrolytic capacitor near the driver.

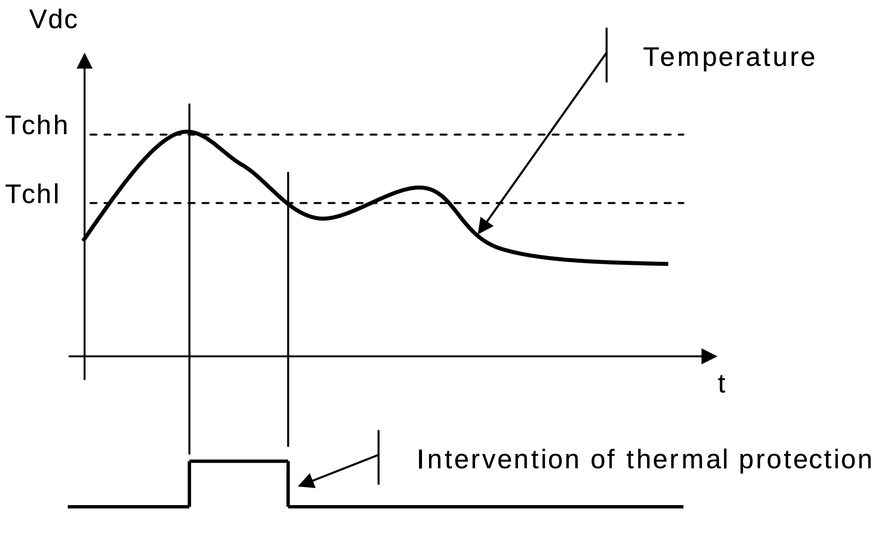
ATTENTION, if the supply voltage increases a lot beyond the maximum functioning value, and in particular it goes beyond the *Vpbrk* breakdown voltage, the driver will be irreversibly damaged.

When the over voltage protection intervenes because of the energy returned from the motor during the deceleration phase, which causes a bus voltage rise beyond the *Vph* value, it is possible to avoid such condition using a braking resistor calibrated to intervene before the bus voltage reaches the *Vph* value.
The power supplies of the DP1xx2 series integrate this functionality and represent a valid solution to solve this problem.

5.4.4 Over temperature

The over temperature protection intervenes when the driver power stage temperature goes beyond the *Tchh* value. In this condition the driver stops working.
When the temperature falls below *Tchl* the alarm signal is removed. According to the driver's setting, when this occurs (see chapter 5.2 Alarms and Protections conditioning) the alarm signal is removed or not.

Symbol	Description	Value			Unit
		Min	Typ	Max	
Tchh	Intervention threshold of thermal protection	84	86	89	**°C**
Tchl	Restoration threshold of thermal protection	64	66	69	**°C**



5.4.5 Phase-to-phase short circuit

In case of a motor wirings short circuit, on the same phase or on different ones (cross phase short circuit), the driver stops working and activates the phase-to-phase short circuit protection.

This kind of protection requires a careful survey to find the cause of the short circuit. Consider that the short circuit can be, apart in the wiring, inside the motor.
According to the driver’s setting (see chapter 5.2 Alarms and Protections conditioning) the protection can be removed through an off-on cycle or disabling temporaneously the driver through the ENABLE signal.

5.4.6 Phase-to-ground short circuit

The protection starts whenever one of the connections towards the motor is in short circuit with the driver ground connection (GND). In these conditions the driver protects itself switching off the supply to the motor and turning in an inactive status.

This protection requires a careful survey to find the cause of the short circuit. Consider that the short circuit can be, apart in the wiring, inside the motor.
According to the driver’s setting (see chapter 5.2 Alarms and Protections conditioning) the protection can be removed through an off-on cycle or disabling temporaneously the driver through the ENABLE signal.

5.4.7 Phase-to-Vp short circuit

Whenever one of the phase connections cause a short circuit with the positive supply voltage (+Vp) the phase-to-Vp short circuit protection starts. When this condition occurs the driver protects itself switching off the supply to the motor and turning in an inactive status.
According to the driver’s setting (see chapter 5.2 Alarms and Protections conditioning) the protection can be removed through an off-on cycle or disabling temporaneously the driver through the ENABLE signal

5.4.8 Interrupted phase A, interrupted phase B

If the connection between the driver and the motor is interrupted, also of a single phase, or an inner motor winding is damaged (interrupting itself) the driver activates the interrupted phase alarm. The signal is distinguished for the phase A and phase B for a more accurate diagnostics.

Take present that the driver makes the interrupted phase control only with the driver enabled and with the motor at rest or with rotation speed inferior to 15 rpm.

6 Programming

6.1 Overview

The DS30 – DS5x series drives allow to execute a program developed by the user driving the motor in both position and/or speed mode.

The programming is made putting in sequence more instruction blocks. An instruction block performs a specific function, as for example the variable assignment, a conditional jump, etc. Each instruction block may have a reference label to be used as jump destination.

At the beginning the drive executes in sequence the functional blocks, starting from the first one (block 001). The jump instruction blocks change the natural sequential program execution order.

Each drive's parameter and function is controlled by the value present in a series of variables called system registers. The instruction blocks can read all the registers while only some can be written. For example the *TargetPos* register, which represents the motor final position, can be both read and written; while the *bInPosition* status boolean register, which indicates when the motor is in position, can be only read.

Besides the system registers, the user is able to create proper program variables defining their name and their size. Variables can be created 1 byte, 2 byte, 3 byte or 4byte long.

The last type of data are the constants or numbers. The constants can get any value included between -2,147,483,648 and 2,147,483,647.

The firmware inside the drive is multitasking. This means that the application program and the motion control algorithm are executed in two different tasks, therefore if the application program stops this does not strictly mean motor stop and vice versa.

The application program does not wait the ending of a movement, if the application has to wait for an event to continue it is necessary to insert a conditional jump that checks the required condition.

The whole application development takes place inside the *UDP Commander* software which assists the programmer from the application program writing, to the compiling, to the download of the compiled program into the drive until the debug session.

The driver connection is made through the DUP port. This connection allows to download the application program into the driver, to control its execution, to read the registers, the user variables, the I/O status, etc. Through the same port it is then possible to update the firmware inside the driver. This last characteristic deserves particular attention as it allows to maintain the product always updated.

The connection between the DUP port of the driver and the PC is made through the UDP30 interface, which is connected to the PC by the USB port.

☺

To grant high noise immunity and to safeguard the equipment integrity, the connection between the PC and the driver must be of insulated type.
The UDP30 interface makes use of the power supply on the USB port to feed the driver digital section, making possible the programming and the reading of information therein contained even without the power supply.

For a more detailed description on the features and installation of the UDP30 interface please see the dedicate user's manual.

6.2 Control hardware signals

Through the control hardware signals we obtain the interaction of the application program with the external equipment.

The application program reads and/or changes the hardware signals through some specialized system registers. The I/O is updated in real time therefore, writing 1 in the register associated to a digital output, the output itself is immediately activated.

In the following tables are detailed the main features of each signal and the associated system register.

Digital inputs			
Number	**Voltage range**	**System register**	**Possible values**
0	3..30Vdc	DigitalInput(0)	0..1
1	3..30Vdc	DigitalInput(1)	0..1
2	3..30Vdc	DigitalInput(2)	0..1
3	3..30Vdc	DigitalInput(3)	0..1

A digital input is defined active when it is supplied. When the input is in the active status, the corresponding system register assumes value 1. When the input is not active the register's value is 0.

Digital outputs			
Number	**Voltage range**	**System register**	**Allowed values**
0	3..30Vdc	DigitalOutput(0)	0..1
1	3..30Vdc	DigitalOutput(1)	0..1

A digital output is active when it allows the current flow. To activate an output the corresponding registers must be set to value 1. Setting a value equal to 0 the output is placed inactive.

Analog inputs				
Number	**Voltage range**	**Input impedance**	**System register**	**Possible values**
0	-10Vdc..+10Vdc	47K	AnalogInput(0)	-1024..+1023
1	-10Vdc..+10Vdc	47K	AnalogInput(1)	-1024..+1023

Analog outputs				
Number	**Voltage range**	**Output current**	**System register**	**Possible values**
0	0..10Vdc	10mA	AnalogOutput(0)	0..1023

6.3 System registers

The drive registers allow to control many parameters and functions. Some may be only read (R), others may be both read or written (RW).
If we try to assign to a register a value out of the allowed range, the register is automatically clamped to the nearest maximum or minimum value. If for example we try and assign the value -35 to a register with range 0..100, then the register will assume the value 0; similarly trying to assign to the same register the value 122 the register will be set at a maximum value of 100.

The following table resumes the available registers.

Name	Description	Input / size	Default	Possible / Allowed values	Unit
TargetPos	Destination position	RW, 4byte	0	-2147483648… +2147483647	1/128 step
RefVel	Speed reference	RW, 2byte	0	-32768..+32767	0.25rpm
Position	Real time position of the motor	RW, 4byte	0	-2147483648… +2147483647	1/128 step
MaxVel	Maximum rotation speed	RW, 2byte	0	0..12000	0.25rpm
Acceleration	Allowed maximum acceleration	RW, 2byte	0	1..30000	0.25rpm /s
Deceleration	Allowed maximum deceleration	RW, 2byte	0	1..30000	0.25rpm /s
ControlMode	Movement control mode	RW, 1byte	1	0..1	
PhaseCurrent	Motor phase current	RW, 1byte	I min	Imin .. Imax	0.1Arms
StByCurrent_Time	Delay before current reduction	RW, 1byte	5	1..100	100ms
StByCurrent_Percentage	Current reduction percentage	RW, 1byte	50	0..100	%
TimerA	Timer	RW, 2byte	0	0..32767	1ms
Status	Drive status	R, 1byte		-128..+127	
ControlFlags	It contains various control bits	RW, 1byte	0	-128..+127	
Fault	Drive's faults	R, 1byte		-128..+127	
Error	Drive's errors	R, 1byte		-128..+127	
bEnable	Drive enabling	RW, bool	0	0..1	
bInPosition	In position motor flag	R, bool	0	0..1	
bEnabled	Drive enabling status	R, bool	0	0..1	
bInStop	In stop motor flag	R, bool	0	0..1	
DigitalInput(0)	0 digital input image	R, bool		0..1	
DigitalInput(1)	1 digital input image	R, bool		0..1	
DigitalInput(2)	2 digital input image	R, bool		0..1	
DigitalInput(3)	3 digital input image	R, bool		0..1	
DigitalOutput(0)	0 digital output image	RW, bool	0	0..1	
DigitalOutput(1)	1 digital output image	RW, bool	0	0..1	
AnalogInput(0)	0 analog input image	R, 2byte		-1024..+1023	9.77mV
AnalogInput(1)	1 analog input image	R, 2byte		-1024..+1023	9.77mV
AnalogOutput(0)	0 analog output image	RW, 2byte	0	0..1023	9.77mV

The DS30 – DS5x series contains an evolved motor control algorithm able to recalculate in real time the motor movement. This allows to change in any time the value of the movement control registers, also without waiting for the motor to stop.
.

6.3.1 Description of each system register

The following is a detailed description of the various registers.

6.3.1.1 TargetPos

It is the destination position required to the motor in 1/128 of full step. This register acts when the *position control* mode is selected (see *ControlMode* register description). When the drive works in *speed control* mode, the value of the *TargetPos* register is ignored.

To command a positioning you have to simply write the destination quote value in the register. The motor will reach the target quote according to the values of acceleration, deceleration and maximum speed present respectively in the *Acceleration, Deceleration* and *MaxVel* registers.

The *TargetPos* registers is accessible in both reading and writing mode and can assume any value between -2,147,483,648 and +2,147,483,647.

6.3.1.2 RefVel

It is the speed reference used in the *speed control* mode (see *ControlMode* register description). When the drive works in *position control* mode, the content of the *RefVel* register is ignored.

With the register set at 0 the motor stops, while at values different from 0 the motor rotates at a speed proportional to the absolute value of the register itself. The sign, positive or negative, of the value written in the *RefVel* register determines the rotation direction. In other words setting the *RefVel* register at the value -800 or +800, we obtain an identical rotation speed but in two different directions.

The motor movement variations always occur according to the acceleration, deceleration and maximum speed values present respectively in the *Acceleration, Deceleration* e *MaxVel* registers.

The *RefVel* register is accessible in both reading and writing mode and can assume values between -12000 and +12000.

6.3.1.3 Position

Through this register it is possible to know the real time motor position expressed in 1/128 of step. The value is constantly updated both the drive is working in position and in speed control mode.

The *Position* register, besides in reading, it is also accessible in writing and this allows to force the motor quote at any value. A typical case is the zero procedure where we need to put to 0 the motor quote, usually when the home sensor is activate.

The *Position* register is accessible in both reading and writing mode and can be set or assume values between -2,147,483,648 and +2,147,483,647.

6.3.1.4 MaxVel

The *MaxVel* register is useful to set the maximum speed limit at which the motor can operate. The maximum speed value is an absolute type value and it is applied for the movements executed by the motor in both directions. The speed constrainment occurs when the drive works both in position and speed control mode.
If the *MaxVel* register is modified dynamically with the motor in movement, the motor speed adjustment always occurs according to the acceleration and deceleration values present respectively in the *Acceleration*, *Deceleration* registers.
The *MaxVel* register is accessible in both reading and writing mode and it can assume values between 0 and +12000. Each unit corresponds to 0,25rpm, therefore setting for example the value 2000 in the *MaxVel* register the maximum speed is set at 500rpm.

6.3.1.5 Acceleration

The maximum motor acceleration is limited by the *Acceleration* register. It is possible to assign to the register values included between 1 and 30000 and it is possible to change the value also with the motor in movement or during acceleration. The *Acceleration* register is accessible in both reading and in writing mode. Each unit corresponds to 0,25rpm/s, therefore assigning for example the value 10000 to the *Acceleration* register an acceleration equal to 2500 rpm/s is set.

6.3.1.6 Deceleration

The motor deceleration is limited by the *Deceleration* register. It is possible to assign to the register values included between 1 and 30000 and it is possible to change the value also with the motor in movement or during deceleration. The *Deceleration* register is accessible in both reading and writing mode. Each unit corresponds to 0,25rpm/s, therefore assigning for example the value 4000 to the *Deceleration* register a deceleration equal to 1000 rpm/s is set.

6.3.1.7 ControlMode

The *ControlMode* register is useful to set the motor control mode. When the register value is 1 the drive works in speed control mode, while when the value is 0 the drive operates a position control. The register is accessible both in reading and writing mode and it can be dynamically changed during the movement. The following table resumes the possible values.

Value	Description
0	Position control
1	Speed control

6.3.1.8 PhaseCurrent

The value of this register defines the effective current (rms) supplied to the motor during rotation. Each unit corresponds to 100mA, therefore setting for example 35 in the *PhaseCurrent* register the current supplied to the motor will be equal to 3.5Arms.
The value of this register can be dynamically changed by the application program. This makes possible to adapt the motor torque to the various working needs reducing for example the current when the motor nominal torque is not necessary, or boost the motor current if a greater torque is required for short time.
The *PhaseCurrent* register is accessible in both reading and writing mode. The minimum and maximum limit of the values writeable in the register is not fixed but it varies according to the drive model. In the following table are the limits according to the drive model:

Model	Minimum value	Maximum value
DSxx44	10	40
DSxx48	30	80
DSxx73	8	30
DSxx76	20	60
DSxx78	40	100
DSxx84	20	40
DSxx87	40	85
DSxx98	40	100

6.3.1.9 StByCurrent_Time

The *StByCurrent_Time* register allows to change the time expected after the motor stop before reducing the phase current. Each unit produces a delay of 100ms, therefore for example setting the register at the value 15, the motor phase current will be reduced after 1.5 seconds from stop.
The *StByCurrent_Time* register is accessible in both reading and writing mode and it can assume the values included between 1 and 100.

6.3.1.10 StByCurrent_Percentage

It allows to define the phase current reduction percentage set through the *PhaseCurrent* register, elapsed the time from the motor stop defined in the *StByCurrent_Time* register.
The *StByCurrent_Percentage* register is accessible in both reading and writing mode and it can assume values included between 0 and 100.

6.3.1.11 TimerA

The *TimerA* register has a particular characteristic: the value it contains is automatically decreased by the drive of one unit each msec until the 0 value is reached; such a value is maintained until the register is not written again.

The register is normally used to realize delays or to measure the time elapsing between two events.
The *TimerA* register is accessible in both reading and writing mode and it can assume values included between 0 and 32767.

6.3.1.12 Status

It is a read only register whose bits give various information on the drive status. The following table shows the correspondence between the bits and the given information.

Bit	Name	Description
bit7	bInPosition	It becomes active (value= 1) when the motor reaches the set position, i.e. when the *Position* register is equal to the *TargetPos* register
bit6	bInStop	It becomes active (value = 1) when the motor is in stop
Bit5	bEnabled	It becomes active (value = 1) when the motor is enabled
Bit4...bit0	Not used	

Some bits are accessible also as boolean type independent registers.

6.3.1.13 ControlFlags

It is a register that controls some driver's functions. The following table shows the correspondence between the various bits and the function performed.

Bit	Name	Description
bit7..bit1	not used	
bit0	bEnable	It allows to enable (value = 1) or disable (value = 0) the motor.

Some bits are accessible also as boolean type independent registers.

6.3.1.14 Fault

It is a read only register which assumes a value different from 0 each time it detects a fault condition. The fault cause or causes can be easily determined analyzing the status of the various bits composing the register. The correspondence is resumed in the following table:

Bit	Name	Description
bit7		It becomes active (value = 1) if the drive detects a break phase condition on A phase.
bit6		It becomes active (value = 1) if the drive detects a break phase condition on B phase
bit5		It becomes active (value = 1) when the drive detects a short circuit between a phase and the Vp supply
bit4		It becomes active (value = 1) when the drive detects a short circuit between a phase and the supply negative

		(GND)
bit3		It becomes active (value = 1) when the drive detects a short circuit among the motor phases
bit2		It becomes active (value = 1) when the heat sink temperature exceeds the maximum allowed value
bit1		It becomes active (value = 1) when the power supply voltage exceeds the maximum allowed value
bit0		It becomes active (value = 1) when the power supply voltage is inferior to the minimum allowed value

6.3.1.15 Error

It is a read only register which assumes values different from 0 each time an error occurs inside the drive.
In the following table are the possible values assumed by the register and their meaning:

Value	Description
0	No error
1	Wrong application program checksum

6.3.1.16 bEnable

It is a boolean type register useful to enable or disable the motor. When the value described in the register is 1 the motor is enabled, when the value is 0 the motor is disabled.
The *bEnable* register allows accesses in both reading and writing mode.
In the following table are resumed the effects produced by the various values assignable to the register:

Value	Description
0	Disable the motor
1	Enable the motor

6.3.1.17 bInPosition

It is a boolean type read only register which becomes active (value equal to 1) when the motor is in position. In other words the *bInPosition* register is activated when the value of the *Position* register is equal to the one present in the *TargetPos* register.
This register is often used together with a conditional jump to suspend the application program execution until the motor has reached the position required.
The *bInPosition* register is updated only when the position control mode is active.
In the following table are resumed the meanings expressed by the various values:

Value	Description
0	Motor not in position
1	Motor in position

6.3.1.18 bInStop

It is a boolean type read only register which becomes active (value equal to 1) when the motor is in stop (speed equal to 0).
In the following table are resumed the meanings expressed in the various values:

Value	Description
0	Motor in rotation
1	Motore in stop

6.3.1.19 bEnabled

It is a boolean type read only register useful to know is the motor is enabled or not.
The enable status in fact does not depend only on the *bEnable* register, but also on the drive status. If for example there are fault conditions, the drive disables the motor independently from the value of the *bEnable* register.
The purpose of the *bEnabled* register is to give information about the real status of the motor.

Value	Description
0	Disabled motor
1	Enabled motor

6.3.1.20 DigitalInput(0)..(3)

It is a boolean type read only register which reflects the status of the corresponding digital input. When the register assumes value 1 it means that the input is supplied (active); when the register's value is 0 it means that the input is disabled (inactive).

Value	Description
0	Not supplied input (inactive)
1	Supplied input (active)

6.3.1.21 DigitalOutput(0)..(1)

It is a boolean type register accessible in both reading and writing mode. Setting the register at 1, the corresponding output is activated and it allows the current flow; on the contrary, setting the register at 0 the output is deactivated preventing the current flow.

Value	Description
0	Inactive output
1	Active output

6.3.1.22 AnalogInput(0)..(1)

It is a read only register which reflects the status of the corresponding analog input. It can assume values included between -1024 and +1023 in correspondence with input voltage from -10V to +10V.

6.3.1.23 AnalogOutput(0)

It is the register to be used to set the analog output voltage. It can be assigned to any value included between 0 and 1023. The *AnalogOutput(0)* is accessible in both reading and writing mode.

6.4 Variables

Besides the registers, there are available program variables freely configurable by the user both as name and size. The program variables are an essential resource to ensure the maximum flexibility and to allow to realize complex applications. The use of the variables allows to realize counters, *for..next* cycles, mathematic computations, etc.
The variables are always signed integer type and can represent both positive and negative integer numbers. The maximum value storable in the variable depends on the size of the variable which can be defined as of 1 byte, 2 bytes 3 bytes or 4 bytes.

In the following table are resumed the maximum and minimum values which can be stored in each variable size:

Size	Access	Allowed values
1 byte	RW	-128..+127
2 byte	RW	-32768..+32767
3 byte	RW	-8388608..+8388607
4 byte	RW	-2147483648..+2147483647

If we try to assign a value greater than the maximum allowed one, the variable is automatically limited to the maximum or minimum allowed value.

6.5 Constants

The constants are represented by integer numbers with sign. The constants can assume any value included between -2,147,483,648 and +2,147,483,647.

6.6 Instruction blocks

The instruction block is the basic unit of the application program. An application program is built putting in sequence more instruction blocks. The *UDP Commander* software is a convenient graphic interface which assists the user during the application program writing.
The following is a detailed description of the available instruction blocks.

6.6.1 Assignment

The assignment block allows to write a value to a register or to a variable. The left operand is the destination while the right one is the source. They can be both registers and program variables while only the right operand can be also a constant (a number) or a read only register. The following are some assignment examples:

Instruction block	Description
Assign: TargetPos = 1500	Assign to the *TargetPos* register the value 1500
Assign: DigitalOutput(0) = DigitalInput(1)	Put the 0 digital output in the same status of the 1 digital input.
Assign: myVar = Position	Assign the value of the *Position* register to the *myVar* variable

6.6.2 Jump

The jump instructions block allow to change the sequential execution normal flow of the application program. The jumps can be unconditional type, which are always executed, or conditioned to the meeting of condition.
The jump destination is identified by a label, inside the program there must be an instruction block, and one only, with the same label.
When the jump is executed, the application program flow changes instantaneously from the jump block to the block with the corresponding destination label.

An unconditional jump is simply coded by declaring the jump destination label.
The following is an example of unconditional jump:

Instruction block	Description
Jump to *jumpHere* **always:**	Jump always to the instruction block identified by the *jumpHere* label.
Jump to *thisPlace* **always:**	Jump always to the instruction block identified by the *thisPlace* label.

The conditional jumps are characterized by two different operands and a comparison one. When the equation result is *true* the jump is executed, otherwise the drive executes the next instruction block in the application program.
Both the right operand and the left one can be a register, a variable or a constant. Instead, the comparison operand can be chosen among the ones in the following table:

Operator	**Description**
=	The comparison is *true* when the left operand value is equal to the right one
<>	The comparison is *true* when the left operand value is different from the right one
>	The comparison is *true* when the left operand value is greater than the right one
>=	The comparison is *true* when the left operand value is greater or equal to the right one
<	The comparison is *true* when the left operand value is smaller than the right one
<=	The comparison is *true* when the left operand value is smaller or equal to the right one

Following are some examples of conditional jump:

Instruction block	Description
Jump to *jumpHere* **if:** Position > 12500	Jump to the instruction block identified by the *jumpHere* label if the value of the *Position* register is greater than 12500
Jump to *startMovement* **if:** DigitalInput(0) = 1	Jump to the instruction block identified by the *startMovement* label if the value of the 0 digital input is 1 (If the input is active)
Jump to *noElapse* **if:** TimerA <> 0	Jump to the instruction block identified by the *noElapse* label if the value of the *TimerA* register is different from 0
Jump to *doSomething* **if:** myVar1 <= myVar2	Jump to the instruction block identified by the *doSomething* label if the value of the *myVar1* variable is smaller or equal to the value contained in the *myVar2* variable

6.6.3 Mathematics

☺

The mathematic instruction block is a powerful instrument which is rarely present in drives belonging to this price range. The block resolves the following equation:

$$result = \frac{(m1 \times m2)}{bd} + c$$

The left operand, the *result*, can be any variable or register with the writing access. The *m1*, *m2* and *c* operands can be indifferently variables, registers or constants. The *bd* division coefficient can be chosen among the 2 powers included between 2^0 and 2^30. Thanks to the denominator it is possible to simulate computations with decimal coefficients. If, for example, it was necessary to multiply a value by 1.5, it is possible to obtain an identical result multiplying first by 3 and then dividing by 2 (2^1). Choosing the right multiplier/divisor combination it is possible to approximate any decimal multiplication/division.

Following are some examples of mathematic instruction blocks:

Instruction block	Description
Resolve: TargetPos = 10 * AnalogInput(0)	The *TargetPos* register is set at 10 times the value assumed by the 0 analog input
Resolve: counter = counter + 1	The *counter* variable is increased by one unit
Resolve: MaxVel = (DigitalInput(2) * 1000) + 500	The *MaxVel* register assumes value 500 when the 2 digital input is inactive (value = 0) and 1500 when the 2 input is active (value = 1)
Resolve: myVar = (3 * myVar) / 8	The value contained in the *myVar* variable is multiplied by a 0.375 (3/8) factor

The mathematic block has an execution time greater than the other functional blocks, therefore when a quick reaction is required (less than 1ms) it is better to execute before the computations, storing the result in a temporary variable.

6.7 Common programming construct

During the application program writing we can often meet some recurrent instruction blocks combinations. Following are the examples of the most common ones:

6.7.1 Loop

This construct is very frequently used. Each application program has at least an infinite loop cycle: the main loop. The program main loop is useful to transfer the control from the last instruction block to the first one again for the next computation. Without a main loop, the application program ends after that the last instruction block is executed.

Instruction block	Description
***startLoop*:**	
...: ...	First loop block. The *startLoop* label is used as block reference
...: ...	Instruction blocks inside the loop
Jump to *startLoop* **always:**	Unconditional block that brings the execution of the program back to the loop start

6.7.2 Delay

In many cases it is necessary to insert a delay between an operation and the subsequent one. The *TimerA* register is automatically decreased of one unit per msec and it can also be used to realize precise delays.

Instruction block	Description
Assign: TimerA = 80	The required delay time (80ms) is assigned to the *TimerA* register
waitHere:	
Jump to *waitHere* **if:** TimerA < > 0	Jump to the *waitHere* label (on itself) until the value of the *TimerA* register (automatically decreased) does not reach the value 0

6.7.3 For .. Next cycle

The For..Next cycles are useful to execute one or more instruction blocks for a well defined number of times.

It is assumed that the *forNextCnt* variable is already declared with a length equal to 2 bytes. The following example is valid for cycles repeated at least one time.

Instruction block	Description
Assign: forNextCnt = 1450	The number of cycles to be execute (1450) is assigned to the variable used as counter
startFor:	
...: ...	First for..next cycle block. The *startFor* label is used as block reference
...: ...	Instruction blocks inside the for..next cycle
Resolve: forNextCnt = forNextCnt - 1	The *forNextCnt* variable is decreased by one unit
Jump to *startFor* **if:** forNextCnt < > 0	Jump to the instruction block identified by the *startFor* label (for..next cycle start) if the value of the *forNextCnt* variable is different from 0
...: ...	First block executed after the for..next cycle is completed (*forNextCnt* variable = 0)

6.7.4 If …Then…Else

Whenever it is necessary to make an operation conditioned to the occurring of determined conditions, we recur to the If..Then construct; if it is then necessary to execute a second operation, in the case the required condition is not satisfied, we recur to the Else structure.

Instruction block	Description
Jump to *hereIfTrue* **if:** Position > 18000	Jump to the instruction block identified by the *hereIfTrue* label if the value of the *Position* register is greater than 18000
…: …	First executed block if the condition is false. This represents the first block of the Else case
…: …	Instruction blocks of the Else case
Jump to *nextOperation* **always:**	Unconditional jump which terminates the blocks of Else case and transfers the program execution to the first following block to the If…Then…Else construct.
hereIfTrue:	
…: …	First block executed if the condition is true. This represents the first block of the Then case
…: …	Instruction blocks of the Then case
nextOperation:	
…: …	First block subsequent to the If…Then…Else construct executed independently from the test result

6.7.5 Hysteresis

Whenever we use analog values it is often necessary to condition the signal with an hysteresis to avoid that continuous little signal fluctuations (often due to electric noise) produce on the application undesired effects.
To better comprehend this problem, let's think for example about the use of an analog input to command the motor position. Because of the electrical noise the A/D converted value oscillates around the medium value of some units; if the converted number is used directly as motor destination position, the motor will continuously make little movements around the medium position. This may cause mechanical stress and noises and the not intervention of the current reduction function (which occurs only if the motor is in stop). To avoid this behavior it is sufficient to insert a simple algorithm which realizes an hysteresis on the analog input signal.
The following example realizes and hysteresis of +/-3 units in respect to the value read by the analog input.
It is assumed that the *myVar*, *hystHigh* and *hystLow* variables are already declared with a length equal to 2 bytes.

Instruction block	Description
readAgain:	
Assign: myVar = AnalogInput(0)	The real time value of the 0 analog input is assigned to the *myVar* program variable
Jump to *outOfHyst* **if:** myVar > hystHigh	If the *myVar* variable is greater than the hysteresis higher limit (*hystHigh* variable) we proceed updating the hysteresis limits and carrying out the required operations
Jump to *readAgaint* **if:** myVar >= hystLow	This block is reached only if the *myVar* variable is smaller or equal to the hysteresis positive limit. Now we test if the *myVar* variable is smaller than the hysteresis lower limit (*hystLow* variable), in this case we proceed updating the hysteresis limits and carrying out the required operations. Note that the test is negate to avoid the unconditional jump block which would be necessary to go back to the *readAgain* label
outOfHyst:	
Resolve: hystLow = myVar - 3	We update the hysteresis negative limit with a value 5 units lower than the *myVar* variable (which corresponds to the analog input reading)
Resolve: hystHigh = myVar + 3	We update the hysteresis positive limit with a value 5 units higher than the *myVar* variable (which corresponds to the analog input reading)
...: ...	Other operations to execute...

6.7.6 Constrainment of a value

The shown algorithm is useful to contain a value between a minimum and a maximum one. In the example we presuppose to associate the drive analog output to an analog input with the constrainment that the analog output reproduces input variations included only between 1Vdc and 5Vdc. Note that these voltages are equivalent to a numeric value converted from the A/D of 102 and 512 respectively.
It is assumed that the *myVar* variable is already declared with a length equal to 2 bytes.

Instruction block	Description
readAgain:	
Assign: myVar = AnalogInput(0)	The real time value of the 0 analog input is assigned to the *myVar* program variable
Jump to *checkHigh* **if:** myVar >= 102	If the *myVar* variable is higher or equal to the allowed minimum one, jump to the block for the maximum limit control
Assign: myVar = 102	This block is reached if the value of the *myVar* variable is lower than the allowed minimum one. In this case the minimum value is assigned.
Jump to *upDateAO* **always:**	Unconditional jump to the analog output update block
checkHigh:	
Jump to *upDateAO* **if:** myVar <= 512	This block is reached only if the *myVar* variable is higher or equal to the allowed minimum one. Now we test if the *myVar* variable is lower than the maximum limit, in this case jump to the analog output update block.
Assign: myVar = 512	We reach this block if the value of the *myVar* variable is higher than the allowed maximum one. In this case the maximum value is assigned
upDateAO:	
Assign: AnalogOutput(0) = myVar	We update the analog output with the value clipped between 102 and 512
Jump to *readAgain* **always:**	Unconditional jump which brings the program execution back to the *readAgain* label

7 Fieldbus

The information contained in the following chapters apply only to the drives provided with fieldbus belonging to the DS5x series. The DS30 products are not supplied with fieldbus.

7.1 General description

Through the fieldbus it is possible to access to the diver's registers and to the program variables to monitor or control the status of the motor.

The faculty of the driver to execute an own internal program and to communicate contemporaneously with an external device (PC, PLC, CNC, etc.) offers a great flexibility and infinite applicative possibilities.

The machine cycle can, for example, be implemented inside the driver to obtain execution rapidity and modified from time to time in the control master parameters (such as times for instance, repetition numbers, quotes or motor speed, etc.). Another advantage offered by this kind of approach is to extremely limit the number of messages passing on the bus. Inside a structure with decentralized intelligence it is in fact sufficient to send only once the configuration parameters, to command the start and eventually to interrogate once in a while the driver to know its status.

The various I/O of the driver (both analog and digital) allow the local control of sensors such as end-run, home, load cells, resistors, etc., as well as actuators like electromagnets, proportional valves, etc., furtherly relieving the control master of the working load and improving the real-time reply of the system.

7.2 Modbus-RTU Communication protocol

The communication protocol implemented in the drives belonging to the DS5x series is the Modbus-RTU.

The Modbus protocol, born in 1979, soon imposed itself as a standard in the industrial field. Appreciated for its simplicity and efficiency, it is a free protocol which does not require the payment of royalties.

Despite the inferior communication speed in comparison to other protocols (such as Profibus and CAN), it vaunts a great number of installation, constantly increasing, as it represents a simple and economic solution to realize nets and undernets on the machine.

Do not confuse the communication protocol with the physical mean (called also layer) used for data conveyance. The protocol (in our case the Modbus-RTU) is the "language" used by the devices to "speak" to each other while the communication mean can be different (RS485, optical fibre, current loop, etc.).

For example, it is possible to communicate in Spanish by phone, fax, e-mail, etc. The phone, the fax, etc. are the physical mean (the layer) while the Spanish grammar is the protocol.

☺ The DS5x series drives are available in 3 different physical layers as detailed in the following table:

Model	Physical Layer	Notes
DS50xx	**RS485**	It allows to connect more devices to a same communication bus. Optimum noise immunity. Maximum distance 1000m.
DS52xx	**RS232**	Suitable for point-to-point connections. Maximum distance between the devices 20m.
DS54xx	**USB**	Through the Hub it is possible to connect more drives to a same PC. Maximum distance between the PC (or the Hub) and the driver 5m.

The Modbus-RTU protocol strictly provides for a *master-slave* type relation. In the net there is always one only *master* and one or more *slave* devices.
Each communication is always generated by the *master*. The *slave* can send data only further to a command or a question sent by the *master*.
The *master* device (PC, PLC, etc) identifies the *slave* to communicate with through an address which must be univocal inside the net. In other words each *slave* must have an address different from all the others.
The possible addresses varies from 1 to 247. The special address 0 has the function of *broadcast address*, i.e. it allows to send a command which will be elaborated by all the devices in the net. The use of the address 0 does not provide any reply from the *slaves*.
The set of data which compose a Modbus-RTU command includes, besides the *slave* device address, also a function code, a group of data depending on the function invoked and finally a checksum useful to evaluate the integrity of the received data.

In the following description the number in hexadecimal notation are anticipated by the suffix “0x”. The value 0x22 indicates the hexadecimal number 22 which corresponds to the value 34 in the decimal notation.

A detailed description of the Modbus-RTU communication goes beyond the purpose of the this manual. Following are detailed only some basic notions to help and better comprehend the functioning of the product.
For an exhaustive description of the Modbus protocol see the official documentation on the site www.modubus.org.

7.2.1 Data structure (frame)

The set of data (called *frame*) which pass from the *master* control device (PC, PLC or other) towards the *slave* (the driver for example) and vice versa is always organized in the same way and satisfies the following structure:

Address	*Function Code*	*Data*	*CRC*

Address, it is the address of the *slave* receiver of the command when the data flow is from the *master* towards the *slave* or the address of the *slave* which transmits when the data flow is from the *slave* to the *master* (reply).

Function code, it identifies the function the *master* wants the *slave* to execute when the data flow is from the *master* towards the *slave* or the function elaborated by the *slave* when the data flow is from the *slave* to the *master* (reply).

Data, specific information about the evoked function, transmitted to the *slave* or received by the same.

CRC, it is the checksum calculated by the sender using all the data of the *frame*. The receiver then calculates the CRC of the received data and compares it with the CRC received.

7.2.2 Data processing (frame)

The device *slave* processes the data which follow the *address* only if the address itself corresponds to its own or to the address 0 (*broadcast address*). In all the other cases the data which follow the address are ignored.

Once recognized the address the *slave* memorizes all the remaining data until it receives the CRC and then calculates the CRC using the received data and compares it with the received CRC. If the two values are different, the *slave* ignores the *frame* and keeps listening again without sending any reply. The *master* device detects this condition as transmission *time-out* (lack of reply from the *slave*).
If the CRC is correct the *slave* tries to execute the command given by the *master* and if the operation successfully ends the *slave* sends the appropriate reply to the *master* (according to the *Function Code* or no reply if the address was 0, i.e. *broadcast address*).

In case the *slave* cannot execute the command given by the *master*, it sends an error reply called *Exception Frame* composed as follows:

Address	*Function Code* OR 0x80	*Exception Code*	*CRC*

Address, it is the address of the *slave* which sends the reply.

Function Code OR 0x80, it is the code of the function which has generated the error with the bit7 (the most significant bit or MSB) set to 1 (it is equivalent to make the OR of the function code with the value 0x80).

Exception Code, it is a code which gives an indication about the kind of problem detected by the *slave*.

CRC, it is the checksum calculated by the *slave*.

The possible transmitted *Exception Codes* are the following:

Exception Code	Description
0x01	It indicates that it is not possible to execute the required function. Possible causes are the missed implementation of the same function, an attempt to write onto an only writing location, etc.
0x02	It indicates that the location to which we try to access does not exist.
0x03	It indicates that one or more values are beyond the allowed limits.

7.2.3 Checksum computation (CRC)

The checksum computation (CRC) examines the whole *frame* (with exception of the CRC which is instead calculated). The CRC has the size of a word (2byte) and is calculated applying the following algorithm to any byte of the *frame*.

1. The CRC is initialized to the value 0xFFFF.
2. It is executed the XOR (OR exclusive) of the byte with the low byte of the CRC memorizing the CRC result.
3. It is executed the shift (rotation) of the CRC of a position (one bit) towards right and it is set to 0 the bit15 (MSB) of the CRC.
4. If the bit0 (LSB) before the shift was 0 it is back to the point 3 while if the bit0 (LSB) before the shift was at 1 it is executed the XOR of the CRC with the value 0xA001.
5. The points 3 and 4 must be repeated 8 times for each byte (8 shifts).
6. It is repeated from point 2 till when all the *frame* bytes are processed.
7. The CRC value is the checksum.

Take present that the checksum is transmitted in reverse order in respect to the other words. The first byte transmitted is the less significant one and the second byte transmitted the most significant one. Inside the *frame* the CRC is therefore disposed as follows:

	CRC Lo	*CRC Hi*

7.2.4 Data organization

Inside the driver there are 5 different types of data with different dimensions; *bit*, *byte*, *word* (2 bytes), *3byte*, and *dword* (double word i.e. 4 bytes).
Any register or program variable read or written through the Modubus-RTU protocol must been formatted inside one or more words.
Following are detailed the rules with which the data in the various dimensions are or must be formatted and interpreted:

7.2.4.1 bit

The binary type data corresponds to the bit0 of the word. The other bits, from bit1 to bit15, remain unused and have value 0.

Ex. 1
bit=0

Word 0															
bit15															**bit0**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**

Value of the bit

Ex. 2
bit=1

Word 0															
bit15															**bit0**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**

Value of the bit

7.2.4.2 byte

The byte type data are present in the lowest 8 bits of the word (from bit7 to bit0), the higher 8 bits (from bit15 to bit8) are instead set to 1 or 0 following the rule of the extension of the sign; if the byte value is >= 0 the bits are set to 0, if instead it is < 0 the are all set to 1.

Es. 1
byte = 58 (0x3A)

Word 0															
bit15															**bit0**
0	0	0	0	0	0	0	0	**0**	**0**	**1**	**1**	**1**	**0**	**1**	**0**

Sign extension Byte value (58)

Es. 2
byte = -61 (0xC3)

Word 0															
bit15															**bit0**
1	1	1	1	1	1	1	1	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**

Sign extension — Byte value (-61)

7.2.4.3 word

The word type data are represented in native mode, without any particular format.

Es. 1
word = 12384 (0x3060)

Word 0															
bit15															**bit0**
0	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**0**

Word value (0x3060)

Es. 2
byte = -3172 (0xF39C)

Word 0															
bit15															**bit0**
1	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**1**	**0**	**0**

Word value (0xF39C)

7.2.4.4 3byte

The 3 bytes data are represented using two words. The first word, the one with minor address, contains in the lowest 8bit (from bit7 to bit0) the most significant byte of the data while the superior 8 bits (from bit15 to bit8) are set to 1 or 0 following the rule of the extension of the sign; if the byte value is >= 0 the bits are set to 0, if instead it is < 0 they are all set to 1. The second word, with address + 1, contains the two remaining bytes of the data.

Es. 1
3byte = 1526317 (0x174A2D)

Word 0															
bit15															bit0
0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1

Sign extension — Most significant byte (0x17)

Word 1															
bit15															bit0
0	1	0	0	1	0	1	0	0	0	1	0	1	1	0	1

Remaining 2 bytes (0x4A2D)

Es. 2
3byte = -1270301 (0xEC9DE3)

Word 0															
bit15															bit0
1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0

Sign extension — Most significant byte (0xEC)

Word 1															
bit15															bit0
1	0	0	1	1	1	0	1	1	1	1	0	0	0	1	1

Remaining 2 bytes (0x9DE3)

7.2.4.5 dword

The dword type data are represented using two words. The first word, the one with the minor ID (address), contains the two most significant bytes of the dword while the second word with address + 1, contains the two remaining bytes of the word.

Es. 1
word = 898271058 (0x358A8752)

Word 0															
bit15															**bit0**
0	**0**	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**

Value of the two most significant bytes (0x358A)

Word 1															
bit15															**bit0**
1	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**1**	**0**	**1**	**0**	**0**	**1**	**0**

Value of the two most significant bytes (0x8752)

Es. 2
word = -1026521534 (0xC2D08642)

Word 0															
bit15															**bit0**
1	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**1**	**0**	**1**	**0**	**0**	**0**	**0**

Value of the two most significant bytes (0xC2D0)

Word 1															
bit15															**bit0**
1	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**1**	**0**

Value of the two most significant bytes (0x8642)

7.2.5 Supported functions

The manipulation of each register or program variable is executed using only three standard functions of the Modbus protocol. The function with code 0x03 is useful to read the registers or the variables and allows to contemporaneously read up to 2 words at the time. The function 0x10 instead is useful to write both registers and variables and allows to write up to 2 words at the time. Finally, the function with code 0x16 allows to change only some bits of a register or a variable leaving the remaining bits unchanged.

7.2.5.1 0x03, Reading of registers and variables (Read Holding Registers)

The command is useful to read the registers or the program variables.

Through the parameter *Word Number* it is possible to chose if reading one only word or two words contemporaneously. This last mode is useful to read registers or variables with a dimension of 3 or 4 bytes.

The following table shows the composition of the *frame* transmitted to the driver and the relative reply in case of command successfully concluded.

Function	**0x03, Read Holding Registers**					
Request (Master → Slave)						
Transmitted *Frame* structure	*Address*	*Function Code*	*Word ID*		*Word Number*	*CRC*
Dimension (byte)	1	1	2		2	2
Possible values	1..247	0x03	0..65535[1]		1 or 2	0..65535
Reply (Slave → Master)						
Received *Frame* structure	*Address*	*Function Code*	*Byte Counter*	*Word0 Value*	*[Word1 Value]*	*CRC*
Dimension (byte)	1	1	1	2	2	2
Possible values	1..247	0x03	2 or 4	0..65535	0..65535	0..65535

Address, it is the address of the *slave* receiver of the command when the data flow is from the *master* towards the *slave* or the address of the *slave* which transmits when the data flow is from the *slave* to the *master* (reply).

Function code, it is the identification of the function for the reading of registers and program variables (0x03).

Word ID, it is the ID (also called address) of the first word to which we want to access.

Word number, it is the quantity of words on which to operate. At most it is possible to operate on 2 words at the time.

Byte counter, it contains the number of bytes which will follow excepted the CRC. Its value is the double of the parameter *Word Number*.

Word0 value, it is the value of the first read word corresponding to the *Word ID*.

Word1 Value, it is the value of the second read word relative to the *Word ID* + 1. The parameter is indicated between square brackets because it is present only when the command is sent with *Word Number* = 2. Differently the *Word1 Value* is not received.

[1]Only some addresses correspond to valid registers and variables (see further on). The access to non-existent registers or variable produces a reply message of error from the driver.

CRC, it is the checksum calculated by the sender using all the *frame* data. The receiver calculates the CRC of the received data and compares it with the CRC received.

Ex. 1, reading of the register 41226 (*Acceleration)* from the driver with address 1

Function	**0x03, Read Holding Registers**					
Request (Master → Slave)						
Transmitted *frame* structure	*Address*	*Function Code*	*Word ID*		*Word Number*	*CRC*
Transmitted values	0x01	0x03	0xA109		0x0001	0x77F4
Reply (Slave → Master)						
Received *frame* structure	*Address*	*Function Code*	*Byte Counter*	*Word0 Value*	*[Word1 Value]*	*CRC*
Received values	0x01	0x03	0x02	0x03E8[2]		0xB8FA

7.2.5.2 0x10, Writing of registers and variables (Write Multiple Registers)

The command is useful to write the records or the program variables.

Through the parameter *Word Number* it is possible to choose to write one only word or two words contemporaneously. This last mode is useful to write registers or variables with a size of 3 or 4 bytes.

The following table shows the composition of the *frame* transmitted to the driver and of the relative reply in case of command successfully concluded.

Function	**0x10, Write Multiple Registers**							
Request (Master → Slave)								
Transmitted *frame* structure	*Address*	*Function Code*	*Word ID*	*Word Number*	*Byte Counter*	*Word0 Value*	*[Word1 Value]*	*CRC*
Dimension (byte)	1	1	2	2	1	2	2	2
Possible values	1..247	0x10	0..65535	1 or 2	2 or 4	0..65535	0..65535	0..65535
Reply (Slave → Master)								
Received *frame* structure	*Address*	*Function Code*	*Word ID*	*Word Number*				*CRC*
Dimension (byte)	1	1	2	2				2
Possible values	1..247	0x10	0..65535	1 or 2				0..65535

Address, it is the address of the *slave* receiver of the command when the data flow is from the *master* to the *slave* or the address of the *slave* which transmits when the data flow is from the *slave* to the *master* (reply).

Function code, it is the identification of the function for the writing of registers and program variables (0x10).

Word ID, it is the ID (also called address) of the first word to which we want to access.

[2] The values change according to the real contain of the register

Word number, it is the quantity of words on which to operate. At most it is possible to operate on 2 words at the time.

Byte Counter, it contains the number of bytes which will follow excepted the CRC. Its value is the double of the parameter *Word Number*.

Word0 Value, it is the value of the first written word corresponding to the *Word ID*.

Word1 Value, it is the value of the second written word relative to the *Word ID* + 1. The parameter is indicated between square brackets because it is present only when the command is sent with *Word Number* = 2. Differently the *Word1 Value* is not received.

CRC, it is the checksum calculated by the transmitter using all the *frame* data. The receiver calculates the CRC of the received data and compares it with the CRC received.

Ex.1, writing of a dword value in the register 41730 (*TargetPos)* of the driver with address 1

Function	**0x10, Write Multiple Registers**							
Request (Master → Slave)								
Transmitted *frame* structure	*Address*	*Function Code*	*Word ID*	*Word Number*	*Byte Counter*	*Word0 Value*	*Word1 Value*	*CRC*
Possible values	1	0x10	0xA301	0x0002	0x04	0x0003	0xE800	0X6094
Reply (Slave → Master)								
Received *frame* structure	*Address*	*Function Code*	*Word ID*	*Word Number*				*CRC*
Possible values	1	0x10	0Xa301	0x0002				0x324C

7.2.5.3 0x16, Changing of the bytes of a register (Mask Write Registers)

The command is useful to change only some bits of a register or of a program variable, leaving the status of the remaining bits unchanged.

The following table shows the composition of the *frame* transmitted to the driver and the relative reply in case of command successfully concluded.

Function	**0x16, Mask Write Registers**					
Request (Master → Slave)						
Transmitted *frame* structure	*Address*	*Function Code*	*Word ID*	*AND Mask*	*OR Mask*	*CRC*
Dimension (byte)	1	1	2	2	2	2
Possible values	1..247	0x16	0..65535	0..65535	0..65535	0..65535
Reply (Slave → Master)						
Received *frame* structure	*Address*	*Function Code*	*Word ID*	*AND Mask*	*OR Mask*	*CRC*
Dimension (byte)	1	1	2	2	2	2
Possible values	1..247	0x10	0..65535	0..65535	0..65535	0..65535

Address, it is the address of the *slave* receiver of the command when the data flow is from the *master* to the *slave* or the address of the *slave* which transmits when the data flow is from the *slave* to the *master* (reply).

Function code, it is the identification of the function for the changing of registers and program variables (0x16).

Word ID, it is the ID (also called address) of the word on which we want to operate.

AND Mask, it is the mask of bit put in AND with the register or the program variable indicated by *Word ID*.

OR Mask, it is the mask of bit put in OR with the register or the program variable indicated by *Word ID*.

Ex. 1, setting to 0 of the bit0 and setting to 1 of the bit 1 of the register 41473 (*DigitalOutputsA)* of the drive with address 1

Function	**0x16, Mask Write Registers**					
Request (Master → Slave)						
Transmitted *frame* structure	*Address*	*Function Code*	*Word ID*	*AND Mask*	*OR Mask*	*CRC*
Possible values	1	0x16	0xA201	0xFFFE	0x0002	0x02A1
Reply (Slave → Master)						
Received *frame* structure	*Address*	*Function Code*	*Word ID*	*AND Mask*	*OR Mask*	*CRC*
Possible values	1	0x16	0xA201	0xFFFE	0x0002	0x02A1

7.3 Map of the registers and user's variables

The following table shows the position of the registers inside the addressing space. For the program variables is indicated only the address of start and end of the whole space available as the ID of each single variable depends on how the same variable is declared and positioned. The ID of each single variable is shown in the *Id.* column in the chart *Variables* of the *UDP Commander* software.

Map of the registers corresponding to **RegTableVer=1**				
ID [3] (address)		Access (byte)	Register	Description
Dec	Hex			
40193	0x9D01	R (2)	RegTableVer	Version of the registers table
40194	0x9D02	R (2)	ProductCode	Code of the product
40195	0x9D03	R (2)	FirmwareVer	Revision of the Firmware
40196	0x9D04	R (2)	HardwareRev	Revision of the hardware
40197	0x9D05	R (2)	SpecialVersion	Special version
40198	0x9D06	R (4)	SerialNumber	Serial number of the product
40199	0x9D07			
40961	0xA001	RW(..)	...	Program variables
...				
40993	0xA021	RW(..)	...	Program variables
41217	0xA101	R (1)	Fault	Faults present in the driver
41218	0xA102	R (1)	Error	Errors present in the driver
41219	0xA103	R (1)	Status	Status of the driver
41220	0xA104	RW (1)	PhaseCurrent	Phase current of the motor
41221	0xA105	RW (1)	ControlMode	Mode of the movement control
41222	0xA106	RW (1)	StByCurrent_Time	Time before the current reduction
41223	0xA107	RW (1)	StByCurrent_Percentage	Percentage of the current reduction
41224	0xA108	RW (2)	MaxVel	Maximum rotation speed
41226	0xA10A	RW (2)	Acceleration	Maximum acceleration allowed
41227	0xA10B	RW (2)	Deceleration	Maximum deceleration allowed
41228	0xA10C	RW (4)	Position	Instant position assumed by the motor
41229	0xA10D			
41230	0xA10E	RW (2)	TimerA	Timer with base 1ms
41231	0xA10F	RW (1)	ControlFlags	It contains various control bits
41473	0xA201	R (1)	DigitalInputsA	Image of the digital inputs
41474	0xA202	RW (1)	DigitalOutputsA	Image of the digital outputs
41475	0xA203	R (2)	AnalogInput(0)	Image of 0 analog input
41476	0xA204	R (2)	AnalogInput(1)	Image of 1 analog input
41477	0xA205	RW (2)	AnalogOutput(0)	Image of 0 analog output
41729	0xA301	RW (2)	RefVel	Speed reference
41730	0xA302	RW (4)	TargetPos	Position of destination
41731	0xA303			

Many registers detailed in the table have been already described in chapter 7.3 Map of the registers and user's variables, therefore following are reported only the details of the new registers.

[3] Take present that according to the Modbus specification the registers are allocated starting from 1 while the address effectively transmitted on the bus has base 0. Consequently, if for example we want to access to the register *DigitalOutputsA it is necessary to compose the frame* using the address 0xA201.

7.3.1 RegTableVer

It helps to know the registers map version. Future evolutions of the product could present a different organization of the registers and it is important that the master device is able to ascertain that the version of the registers table in use is compatible with its own implementation.

7.3.2 ProductCode

It allows to know the kind of product connected with.

This information is useful to validate the net or to adapt automatically to a specific configuration of the machine.

In the following table is reported the correspondence between the code and the product:

Code		Product
Dec	Hex	
1280	0x0500	DS5x44
1282	0x0502	DS5x48
1284	0x0504	DS5x73
1286	0x0506	DS5x76
1288	0x0508	DS5x78
1290	0x050A	DS5x84
1292	0x050C	DS5x87
1294	0x050E	DS5x98

7.3.3 FirmwareVer

It is a 2 byte registers useful to trace the origin of the firmware version present in the drive.
The version is represented by two number pairs divided by a decimal dot. The number pair before the decimal dot is given by the most significant byte of the register represented in hexadecimal notation, while the pair after the dot is the less significant byte, always in hexadecimal notation. If for example the value of the register *FirmwareVer* is 533 (in hexadecimal 0x0215) it means that the software version present in the driver is 2.15.

7.3.4 HardwareRev

It is a 2 byte register useful to know the hardware version of the driver.
The revision number is represented by two number pairs divided by a decimal dot.
The number pair before the decimal dot is given by the most significant byte, represented in hexadecimal notation, of the register while the number pair after the decimal dot is given by the less significant byte, always represented in hexadecimal notation. If for example the value of the register is 259 (in hexadecimal 0x0103) the hardware revision of the product is 1.03.

7.3.5 SpecialVersion

It is a 2 byte register which assumes a value different from 0 when the hardware of the product has been personalized and consequently differs from the standard one.
The special version code is composed by a 4 figures number. The first two figures are given by the most significant byte of the resister represented in hexadecimal notation, while the second two figures are the less significant byte of the register always represented in hexadecimal notation. If for example the value of the register is 802 (in hexadecimal 0x0322) this means that the special version code of the product is 322.

7.3.6 SerialNumber

It helps to know the serial number of the product. It is a 4 byte register whose decimal value directly represents the SN of the driver.
If for example the value of the register is 7136335 (in hexadecimal notation 0x006CE44F) it means that the serial number of the product is 07136335 (the serial number is always composed by 8 figures).

8 Technical data

The following are the electric, physical and mechanical details of each single driver.

DS3044/DS5x44					
Symbol	**Description**	**Value**			**Unit**
		Min	**Typ**	**Max**	
Vp	Power supply voltage	20		50	**Vdc**
If	Phase current (**effective current**)	1		4	**Arms**
Vprp	Allowed ripple (100Hz)			8	**Vpp**
Vpbrk	Permanent breakdown voltage	-0.5		60	**Vdc**
Vph	Over voltage protection intervention	56.0		57.5	**Vdc**
Vpl	Under voltage protection intervention	18.5		19.7	**Vdc**
Tchh	Thermal protection intervention threshold	84	86	94	**°C**
Tchl	Thermal protection restoration threshold	64	66	69	**°C**
Plss	Power lost on the driver			10	**W**
Ml	Motor inductance seen by the driver	0.8		50	**mH**
Vdi	Digital input voltage range	3		30	**Vdc**
Vdibrk	Digital input breakdown voltage	-35		+35	**Vdc**
Idi	Digital input supply current	4	6	8	**mA**
Vdo	Digital output voltage range	1		30	**Vdc**
Vdobrk	Digital output breakdown voltage	-0.5		37	**Vdc**
Vdoz	Output zener diode voltage	37	39	42	**Vdc**
Ido	Digital output current range			50	**mA**
Idobrk	Digital output breakdown current	120			**mA**
Pwdo	Digital output dissipable power			300	**mW**
Vai	Analog input voltage range	-10		+10	**Vdc**
Vaibrk	Analog input breakdown voltage	-45		+45	**Vdc**
Rai	Analog input impedance		47		**KΩ**
ADrai	A/D converter resolution		11		**bit**
ADst	A/D converter converting time		10		**msec**
ADsoff	A/D converter start offset		1	2	**%fs**
ADdoff	A/D converter offset drift		0.3	0.5	**%fs**
ADline	A/D converter linearity error		1	2	**%fs**
Vao	Analog output voltage range	0		+10	**Vdc**
Iao	Analog output current range			10	**mA**
Iasc	Analog output short circuit current			20	**mA**
Rao	Analog output impedance			1	**Ω**
DArai	D/A converter resolution		11		**bit**
DAud	D/A converter update		10		**msec**
DAsoff	D/A converter start offset		0.3	1	**%fs**
DAdoff	D/A converter offset drift		0.1	0.3	**%fs**
DAline	D/A converter linearity error		0.3	1.5	**%fs**
Prt	Protections and alarms	Over/Under voltage, Short circuit Overheating, Break phase			
	Mechanical specifications				
FDh	Height		100.4		**mm**
FDl	Depth		119.0		**mm**
FDw	Width		17.5 (22.5 for DS5x)		**mm**
FDnw	Weight		185		**gr**
	Rated range of use				
FCa	Altitude			2000	**m**
Idr	*If* current degrading every 1000m beyond the *FCa* altitude value		5		**%**
FCt	Temperature	0		50	**ºC**
FCh	Humidity (no condensing)	5		90	**%**
	Conditions of storage and transport				
SCa	Altitude			4000	**m**
SCt	Temperature	-20		70	**ºC**
SCh	Humidity (no condensing)	5		95	**%**

DS3048/DS5x48					
Symbol	**Description**	**Value**			**Unit**
		Min	**Typ**	**Max**	
Vp	Power supply voltage	20		50	**Vdc**
If	Phase current (**effective current**)	3		8	**Arms**
Vprp	Allowed ripple (100Hz)			8	**Vpp**
Vpbrk	Permanent breakdown voltage	-0.5		60	**Vdc**
Vph	Over voltage protection intervention	56.0		57.5	**Vdc**
Vpl	Under voltage protection intervention	18.5		19.7	**Vdc**
Tchh	Thermal protection intervention threshold	84	86	94	**°C**
Tchl	Thermal protection restoration threshold	64	66	69	**°C**
Plss	Power lost on the driver			20	**W**
Ml	Motor inductance seen by the driver	0.5		30	**mH**
Vdi	Digital input voltage range	3		30	**Vdc**
Vdibrk	Digital input breakdown voltage	-35		+35	**Vdc**
Idi	Digital input supply current	4	6	8	**mA**
Vdo	Digital output voltage range	1		30	**Vdc**
Vdobrk	Digital output breakdown voltage	-0.5		37	**Vdc**
Vdoz	Output zener diode voltage	37	39	42	**Vdc**
Ido	Digital output current range			50	**mA**
Idobrk	Digital output breakdown current	120			**mA**
Pwdo	Digital output dissipable power			300	**mW**
Vai	Analog input voltage range	-10		+10	**Vdc**
Vaibrk	Analog input breakdown voltage	-45		+45	**Vdc**
Rai	Analog input impedance		47		**KΩ**
ADrai	A/D converter resolution		11		**bit**
ADst	A/D converter converting time		10		**msec**
ADsoff	A/D converter start offset		1	2	**%fs**
ADdoff	A/D converter offset drift		0.3	0.5	**%fs**
ADline	A/D converter linearity error		1	2	**%fs**
Vao	Analog output voltage range	0		+10	**Vdc**
Iao	Analog output current range			10	**mA**
Iasc	Analog output short circuit current			20	**mA**
Rao	Analog output impedance			1	**Ω**
DArai	D/A converter resolution		11		**bit**
DAud	D/A converter update		10		**msec**
DAsoff	D/A converter start offset		0.3	1	**%fs**
DAdoff	D/A converter offset drift		0.1	0,3	**%fs**
DAline	D/A converter linearity error		0.3	1.5	**%fs**
Prt	Protections and alarms	Over/Under voltage, Short circuit Overheating, Break phase			
Mechanical specifications					
FDh	Height		100.4		**mm**
FDl	Depth		119.0		**mm**
FDw	Width		35		**mm**
FDnw	Weight		295		**gr**
Rated range of use					
FCa	Altitude			2000	**m**
Idr	*If* current degrading every 1000m beyond the *FCa* altitude value		5		**%**
FCt	Temperature	0		50	**°C**
FCh	Humidity (no condensing)	5		90	**%**
Conditions of storage and transport					
SCa	Altitude			4000	**m**
SCt	Temperature	-20		70	**°C**
SCh	Humidity (no condensing)	5		95	**%**

DS3073/DS5x73					
Symbol	**Description**	**Value**			**Unit**
		Min	**Typ**	**Max**	
Vp	Power supply voltage	24		90	**Vdc**
If	Phase current (**effective current**)	0.8		3	**Arms**
Vprp	Allowed ripple (100Hz)			15	**Vpp**
Vpbrk	Permanent breakdown voltage	-0.5		105	**Vdc**
Vph	Over voltage protection intervention	95		98	**Vdc**
Vpl	Under voltage protection intervention	22.5		23.5	**Vdc**
Tchh	Thermal protection intervention threshold	84	86	94	**°C**
Tchl	Thermal protection restoration threshold	64	66	69	**°C**
Plss	Power lost on the driver			10	**W**
Ml	Motor inductance seen by the driver	0.8		50	**mH**
Vdi	Digital input voltage range	3		30	**Vdc**
Vdibrk	Digital input breakdown voltage	-35		+35	**Vdc**
Idi	Digital input supply current	4	6	8	**mA**
Vdo	Digital output voltage range	1		30	**Vdc**
Vdobrk	Digital output breakdown voltage	-0.5		37	**Vdc**
Vdoz	Output zener diode voltage	37	39	42	**Vdc**
Ido	Digital output current range			50	**mA**
Idobrk	Digital output breakdown current	120			**mA**
Pwdo	Digital output dissipable power			300	**mW**
Vai	Analog input voltage range	-10		+10	**Vdc**
Vaibrk	Analog input breakdown voltage	-45		+45	**Vdc**
Rai	Analog input impedance		47		**KΩ**
ADrai	A/D converter resolution		11		**bit**
ADst	A/D converter converting time		10		**msec**
ADsoff	A/D converter start offset		1	2	**%fs**
ADdoff	A/D converter offset drift		0.3	0.5	**%fs**
ADline	A/D converter linearity error		1	2	**%fs**
Vao	Analog output voltage range	0		+10	**Vdc**
Iao	Analog output current range			10	**mA**
Iasc	Analog output short circuit current			20	**mA**
Rao	Analog output impedance			1	**Ω**
DArai	D/A converter resolution		11		**bit**
DAud	D/A converter update		10		**msec**
DAsoff	D/A converter start offset		0.3	1	**%fs**
DAdoff	D/A converter offset drift		0.1	0.3	**%fs**
DAline	D/A converter linearity error		0.3	1.5	**%fs**
Prt	Protections and alarms	Over/Under voltage, Short circuit Overheating, Break phase			
Mechanical specifications					
FDh	Height		100.4		**mm**
FDl	Depth		119.0		**mm**
FDw	Width		17.5 (22.5 for DS5x)		**mm**
FDnw	Weight		185		**gr**
Rated range of use					
FCa	Altitude			2000	**m**
Idr	*If* current degrading every 1000m beyond the *FCa* altitude value		5		**%**
FCt	Temperature	0		50	**ºC**
FCh	Humidity (no condensing)	5		90	**%**
Conditions of storage and transport					
SCa	Altitude			4000	**m**
SCt	Temperature	-20		70	**ºC**
SCh	Humidity (no condensing)	5		95	**%**

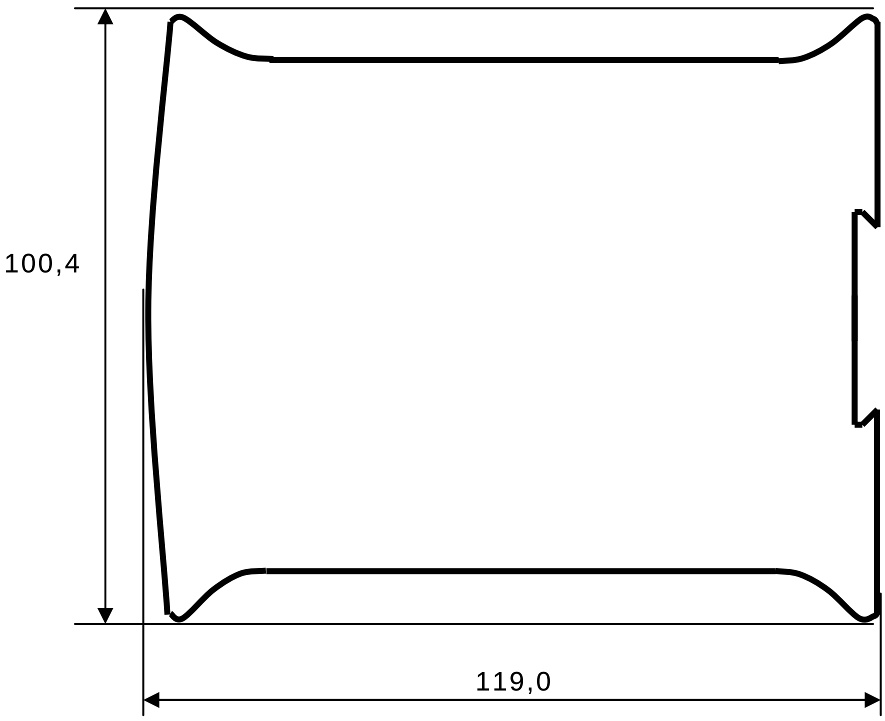
DS3076/DS5x76					
Symbol	*Description*	*Value*			*Unit*
		Min	*Typ*	*Max*	
Vp	Power supply voltage	24		90	**Vdc**
If	Phase current (**effective current**)	2		6	**Arms**
Vprp	Allowed ripple (100Hz)			15	**Vpp**
Vpbrk	Permanent breakdown voltage	-0.5		105	**Vdc**
Vph	Over voltage protection intervention	95		98	**Vdc**
Vpl	Under voltage protection intervention	22.5		23.5	**Vdc**
Tchh	Thermal protection intervention threshold	84	86	94	**°C**
Tchl	Thermal protection restoration threshold	64	66	69	**°C**
Plss	Power lost on the driver			20	**W**
Ml	Motor inductance seen by the driver	0.6		40	**mH**
Vdi	Digital input voltage range	3		30	**Vdc**
Vdibrk	Digital input breakdown voltage	-35		+35	**Vdc**
Idi	Digital input supply current	4	6	8	**mA**
Vdo	Digital output voltage range	1		30	**Vdc**
Vdobrk	Digital output breakdown voltage	-0.5		37	**Vdc**
Vdoz	Output zener diode voltage	37	39	42	**Vdc**
Ido	Digital output current range			50	**mA**
Idobrk	Digital output breakdown current	120			**mA**
Pwdo	Digital output dissipable power			300	**mW**
Vai	Analog input voltage range	-10		+10	**Vdc**
Vaibrk	Analog input breakdown voltage	-45		+45	**Vdc**
Rai	Analog input impedance		47		**KΩ**
ADrai	A/D converter resolution		11		**bit**
ADst	A/D converter converting time		10		**msec**
ADsoff	A/D converter start offset		1	2	**%fs**
ADdoff	A/D converter offset drift		0.3	0.5	**%fs**
ADline	A/D converter linearity error		1	2	**%fs**
Vao	Analog output voltage range	0		+10	**Vdc**
Iao	Analog output current range			10	**mA**
Iasc	Analog output short circuit current			20	**mA**
Rao	Analog output impedance			1	**Ω**
DArai	D/A converter resolution		11		**bit**
DAud	D/A converter update		10		**msec**
DAsoff	D/A converter start offset		0.3	1	**%fs**
DAdoff	D/A converter offset drift		0.1	0.3	**%fs**
DAline	D/A converter linearity error		0.3	1.5	**%fs**
Prt	Protections and alarms	Over/Under voltage, Short circuit Overheating, Break phase			
Mechanical specifications					
FDh	Height		100.4		**mm**
FDl	Depth		119.0		**mm**
FDw	Width		35		**mm**
FDnw	Weight		295		**gr**
Rated range of use					
FCa	Altitude			2000	**m**
Idr	*If* current degrading every 1000m beyond the *FCa* altitude value		5		**%**
FCt	Temperature	0		50	**ºC**
FCh	Humidity (no condensing)	5		90	**%**
Conditions of storage and transport					
SCa	Altitude			4000	**m**
SCt	Temperature	-20		70	**ºC**
SCh	Humidity (no condensing)	5		95	**%**

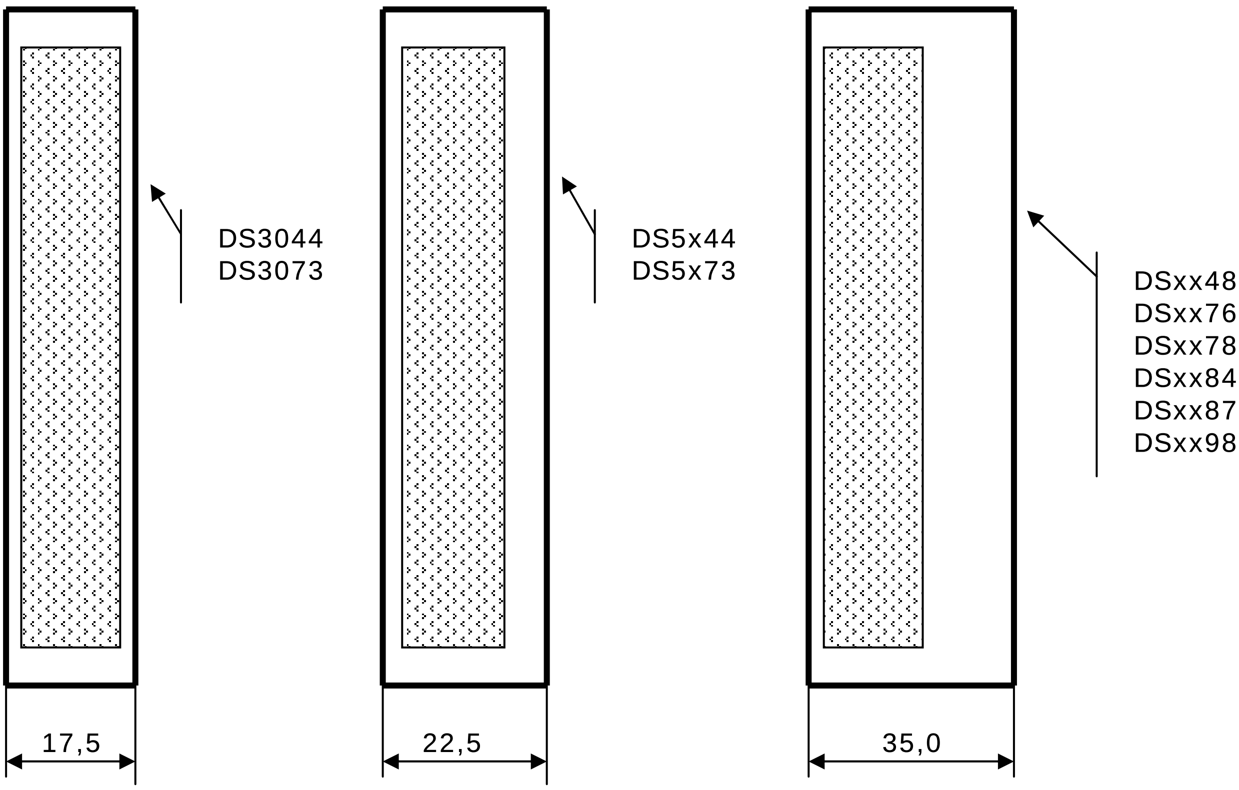
DS3078/DS5x78					
Symbol	***Description***	***Value***			***Unit***
		Min	***Typ***	***Max***	
Vp	Power supply voltage	24		90	**Vdc**
If	Phase current (**effective current**)	4		10	**Arms**
Vprp	Allowed ripple (100Hz)			15	**Vpp**
Vpbrk	Permanent breakdown voltage	-0.5		105	**Vdc**
Vph	Over voltage protection intervention	95		98	**Vdc**
Vpl	Under voltage protection intervention	22.5		23.5	**Vdc**
Tchh	Thermal protection intervention threshold	84	86	94	**°C**
Tchl	Thermal protection restoration threshold	64	66	69	**°C**
Plss	Power lost on the driver			30	**W**
Ml	Motor inductance seen by the driver	0.5		30	**mH**
Vdi	Digital input voltage range	3		30	**Vdc**
Vdibrk	Digital input breakdown voltage	-35		+35	**Vdc**
Idi	Digital input supply current	4	6	8	**mA**
Vdo	Digital output voltage range	1		30	**Vdc**
Vdobrk	Digital output breakdown voltage	-0.5		37	**Vdc**
Vdoz	Output zener diode voltage	37	39	42	**Vdc**
Ido	Digital output current range			50	**mA**
Idobrk	Digital output breakdown current	120			**mA**
Pwdo	Digital output dissipable power			300	**mW**
Vai	Analog input voltage range	-10		+10	**Vdc**
Vaibrk	Analog input breakdown voltage	-45		+45	**Vdc**
Rai	Analog input impedance		47		**KΩ**
ADrai	A/D converter resolution		11		**bit**
ADst	A/D converter converting time		10		**msec**
ADsoff	A/D converter start offset		1	2	**%fs**
ADdoff	A/D converter offset drift		0.3	0.5	**%fs**
ADline	A/D converter linearity error		1	2	**%fs**
Vao	Analog output voltage range	0		+10	**Vdc**
Iao	Analog output current range			10	**mA**
Iasc	Analog output short circuit current			20	**mA**
Rao	Analog output impedance			1	**Ω**
DArai	D/A converter resolution		11		**bit**
DAud	D/A converter update		10		**msec**
DAsoff	D/A converter start offset		0.3	1	**%fs**
DAdoff	D/A converter offset drift		0.1	0.3	**%fs**
DAline	D/A converter linearity error		0.3	1.5	**%fs**
Prt	Protections and alarms	Over/Under voltage, Short circuit Overheating, Break phase			
Mechanical specifications					
FDh	Height		100.4		**mm**
FDl	Depth		119.0		**mm**
FDw	Width		35		**mm**
FDnw	Weight		295		**gr**
Rated range of use					
FCa	Altitude			2000	**m**
Idr	*If* current degrading every 1000m beyond the *FCa* altitude value		5		**%**
FCt	Temperature	0		50	**ºC**
FCh	Humidity (no condensing)	5		90	**%**
Conditions of storage and transport					
SCa	Altitude			4000	**m**
SCt	Temperature	-20		70	**ºC**
SCh	Humidity (no condensing)	5		95	**%**

DS3084/DS5x84					
Symbol	**Description**	**Value**			**Unit**
		Min	**Typ**	**Max**	
Vp	Power supply voltage	45		160	**Vdc**
If	Phase current (**effective current**)	2		4	**Arms**
Vprp	Allowed ripple (100Hz)			25	**Vpp**
Vpbrk	Permanent breakdown voltage	-0.5		210	**Vdc**
Vph	Over voltage protection intervention	177		181	**Vdc**
Vpl	Under voltage protection intervention	26		27	**Vdc**
Tchh	Thermal protection intervention threshold	84	86	94	**°C**
Tchl	Thermal protection restoration threshold	64	66	69	**°C**
Plss	Power lost on the driver			30	**W**
Ml	Motor inductance seen by the driver	1		50	**mH**
Vdi	Digital input voltage range	3		30	**Vdc**
Vdibrk	Digital input breakdown voltage	-35		+35	**Vdc**
Idi	Digital input supply current	4	6	8	**mA**
Vdo	Digital output voltage range	1		30	**Vdc**
Vdobrk	Digital output breakdown voltage	-0.5		37	**Vdc**
Vdoz	Output zener diode voltage	37	39	42	**Vdc**
Ido	Digital output current range			50	**mA**
Idobrk	Digital output breakdown current	120			**mA**
Pwdo	Digital output dissipable power			300	**mW**
Vai	Analog input voltage range	-10		+10	**Vdc**
Vaibrk	Analog input breakdown voltage	-45		+45	**Vdc**
Rai	Analog input impedance		47		**KΩ**
ADrai	A/D converter resolution		11		**bit**
ADst	A/D converter converting time		10		**msec**
ADsoff	A/D converter start offset		1	2	**%fs**
ADdoff	A/D converter offset drift		0.3	0.5	**%fs**
ADline	A/D converter linearity error		1	2	**%fs**
Vao	Analog output voltage range	0		+10	**Vdc**
Iao	Analog output current range			10	**mA**
Iasc	Analog output short circuit current			20	**mA**
Rao	Analog output impedance			1	**Ω**
DArai	D/A converter resolution		11		**bit**
DAud	D/A converter update		10		**msec**
DAsoff	D/A converter start offset		0.3	1	**%fs**
DAdoff	D/A converter offset drift		0.1	0.3	**%fs**
DAline	D/A converter linearity error		0.3	1.5	**%fs**
Prt	Protections and alarms	Over/Under voltage, Short circuit Overheating, Break phase			
Mechanical specifications					
FDh	Height		100.4		**mm**
FDl	Depth		119.0		**mm**
FDw	Width		35		**mm**
FDnw	Weight		295		**gr**
Rated range of use					
FCa	Altitude			2000	**m**
Idr	*If* current degrading every 1000m beyond the *FCa* altitude value		5		**%**
FCt	Temperature	0		50	**ºC**
FCh	Humidity (no condensing)	5		90	**%**
Conditions of storage and transport					
SCa	Altitude			4000	**m**
SCt	Temperature	-20		70	**ºC**
SCh	Humidity (no condensing)	5		95	**%**

DS3087/DS5x87					
Symbol	***Description***	***Value***			***Unit***
		Min	***Typ***	***Max***	
Vp	Power supply voltage	45		160	**Vdc**
If	Phase current (**effective current**)	4		8.5	**Arms**
Vprp	Allowed ripple (100Hz)			25	**Vpp**
Vpbrk	Permanent breakdown voltage	-0.5		210	**Vdc**
Vph	Over voltage protection intervention	177		181	**Vdc**
Vpl	Under voltage protection intervention	26		27	**Vdc**
Tchh	Thermal protection intervention threshold	84	86	94	**°C**
Tchl	Thermal protection restoration threshold	64	66	69	**°C**
Plss	Power lost on the driver			35	**W**
Ml	Motor inductance seen by the driver	0.6		35	**mH**
Vdi	Digital input voltage range	3		30	**Vdc**
Vdibrk	Digital input breakdown voltage	-35		+35	**Vdc**
Idi	Digital input supply current	4	6	8	**mA**
Vdo	Digital output voltage range	1		30	**Vdc**
Vdobrk	Digital output breakdown voltage	-0.5		37	**Vdc**
Vdoz	Output zener diode voltage	37	39	42	**Vdc**
Ido	Digital output current range			50	**mA**
Idobrk	Digital output breakdown current	120			**mA**
Pwdo	Digital output dissipable power			300	**mW**
Vai	Analog input voltage range	-10		+10	**Vdc**
Vaibrk	Analog input breakdown voltage	-45		+45	**Vdc**
Rai	Analog input impedance		47		**KΩ**
ADrai	A/D converter resolution		11		**bit**
ADst	A/D converter converting time		10		**msec**
ADsoff	A/D converter start offset		1	2	**%fs**
ADdoff	A/D converter offset drift		0.3	0,5	**%fs**
ADline	A/D converter linearity error		1	2	**%fs**
Vao	Analog output voltage range	0		+10	**Vdc**
Iao	Analog output current range			10	**mA**
Iasc	Analog output short circuit current			20	**mA**
Rao	Analog output impedance			1	**Ω**
DArai	D/A converter resolution		11		**bit**
DAud	D/A converter update		10		**msec**
DAsoff	D/A converter start offset		0.3	1	**%fs**
DAdoff	D/A converter offset drift		0.1	0.3	**%fs**
DAline	D/A converter linearity error		0.3	1.5	**%fs**
Prt	Protections and alarms	Over/Under voltage, Short circuit Overheating, Break phase			
Mechanical specifications					
FDh	Height		100.4		**mm**
FDl	Depth		119.0		**mm**
FDw	Width		35		**mm**
FDnw	Weight		295		**gr**
Rated range of use					
FCa	Altitude			2000	**m**
Idr	*If* current degrading every 1000m beyond the *FCa* altitude value		5		**%**
FCt	Temperature	0		50	**ºC**
FCh	Humidity (no condensing)	5		90	**%**
Conditions of storage and transport					
SCa	Altitude			4000	**m**
SCt	Temperature	-20		70	**ºC**
SCh	Humidity (no condensing)	5		95	**%**

DS3098/DS5x98					
Symbol	***Description***	***Value***			***Unit***
		Min	***Typ***	***Max***	
Vp	Power supply voltage	45		240	**Vdc**
If	Phase current (**effective current**)	4		10	**Arms**
Vprp	Allowed ripple (100Hz)			30	**Vpp**
Vpbrk	Permanent breakdown voltage	-0.5		265	**Vdc**
Vph	Over voltage protection intervention	242		255	**Vdc**
Vpl	Under voltage protection intervention	35		37	**Vdc**
Tchh	Thermal protection intervention threshold	84	86	94	**°C**
Tchl	Thermal protection restoration threshold	64	66	69	**°C**
Plss	Power lost on the driver			40	**W**
Ml	Motor inductance seen by the driver	0.6		30	**mH**
Vdi	Digital input voltage range	3		30	**Vdc**
Vdibrk	Digital input breakdown voltage	-35		+35	**Vdc**
Idi	Digital input supply current	4	6	8	**mA**
Vdo	Digital output voltage range	1		30	**Vdc**
Vdobrk	Digital output breakdown voltage	-0.5		37	**Vdc**
Vdoz	Output zener diode voltage	37	39	42	**Vdc**
Ido	Digital output current range			50	**mA**
Idobrk	Digital output breakdown current	120			**mA**
Pwdo	Digital output dissipable power			300	**mW**
Vai	Analog input voltage range	-10		+10	**Vdc**
Vaibrk	Analog input breakdown voltage	-45		+45	**Vdc**
Rai	Analog input impedance		47		**KΩ**
ADrai	A/D converter resolution		11		**bit**
ADst	A/D converter converting time		10		**msec**
ADsoff	A/D converter start offset		1	2	**%fs**
ADdoff	A/D converter offset drift		0.3	0.5	**%fs**
ADline	A/D converter linearity error		1	2	**%fs**
Vao	Analog output voltage range	0		+10	**Vdc**
Iao	Analog output current range			10	**mA**
Iasc	Analog output short circuit current			20	**mA**
Rao	Analog output impedance			1	**Ω**
DArai	D/A converter resolution		11		**bit**
DAud	D/A converter update		10		**msec**
DAsoff	D/A converter start offset		0.3	1	**%fs**
DAdoff	D/A converter offset drift		0.1	0.3	**%fs**
DAline	D/A converter linearity error		0.3	1.5	**%fs**
Prt	Protections and alarms	Over/Under voltage, Short circuit Overheating, Break phase			
Mechanical specifications					
FDh	Height		100.4		**mm**
FDl	Depth		119.0		**mm**
FDw	Width		35		**mm**
FDnw	Weight		295		**gr**
Rated range of use					
FCa	Altitude			2000	**m**
Idr	*If* current degrading every 1000m beyond the *FCa* altitude value		5		**%**
FCt	Temperature	0		50	**ºC**
FCh	Humidity (no condensing)	5		90	**%**
Conditions of storage and transport					
SCa	Altitude			4000	**m**
SCt	Temperature	-20		70	**ºC**
SCh	Humidity (no condensing)	5		95	**%**





Dimensions expressed in millimeters. Drawing not to scale.

LAM Technologies
Viale Ludovico Ariosto, 492/D
50019 Sesto Fiorentino – FIRENZE - ITALY
Tel. 055-4207746 – Fax 055-4207651

General information: info@lamtechnologies.com
Technical support: support@lamtechnologies.com
Sales information: sales@lamtechnologies.com

rev. 0.03 del 26/12/2007