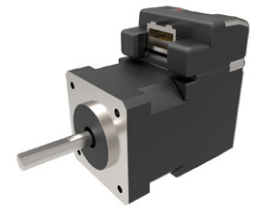
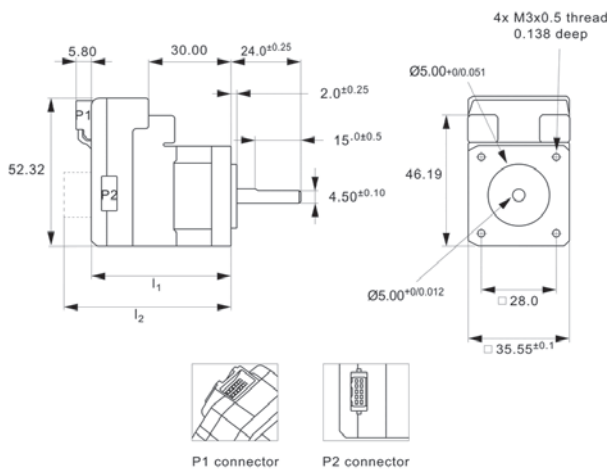




Intelligent NEMA 14 Stepper Motors with mcode software

Motorised
Linear Stages



L3530

MOTORISED LINEAR STAGES

Material

Combined 2 phase, high torque stepper motors with in-built power driver and controller.
IP20 rated (IP 67 optional).

Technical Notes

Allows easy control from a PC or PLC for single or multiple motors. Low cost alternative to motors, drivers and controllers. Easy to use free software, little cabling.

Voltage 12 to 48V DC.

20 microstepping resolutions up to 51,200 steps per rev.
Up to 8 I/O lines, one 10 bit selectable analogue input.

Tips

Can readily be programmed in Labview, VB etc. The motor can be run independently from PC if required (programmed can be downloaded to motor). Easy connection via

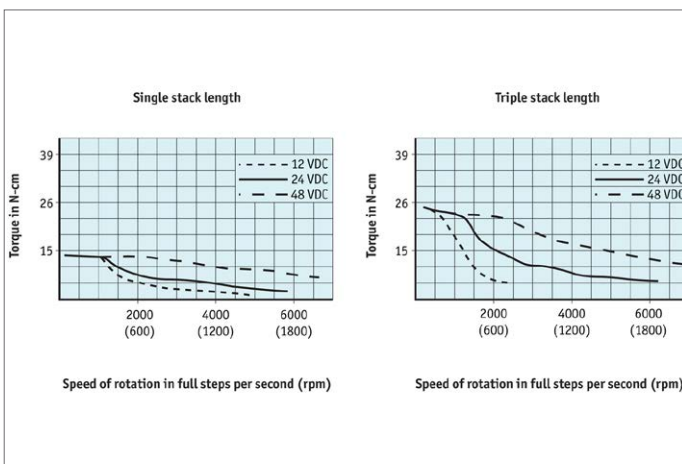
RS422/485.

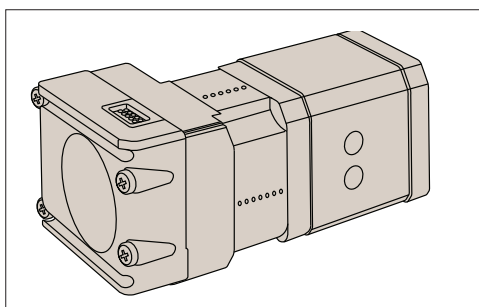
Optional encoders, gearing, motor brake etc.

Important Notes

We have a free motor selection help service - including a free motor configuration software programme and technical help to ensure the motor is to your requirement - please consult our technical department for full motor specifications.

Order No.	Holding torque Nm	Flange dimensions	l_1 max.	l_2 max.	Shaft +0 -0.013	Rotor inertia kg·cm ²	Weight kg
L3530.14-1	0.13	35x35	49	67	5.00	0.014	0.15
L3530.14-3	0.25	35x35	77	95	5.00	0.057	0.38





Intelligent stepper motor

- No need for separate motion controller.
- Inbuilt motor, driver and controller.

Options

- Standard
- With rotary encoder (512 line)

Separate motor controllers (single axis)

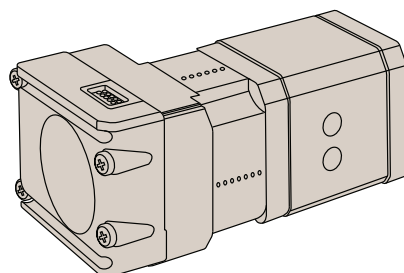
Single axis stepper controller



Single axis servo controller

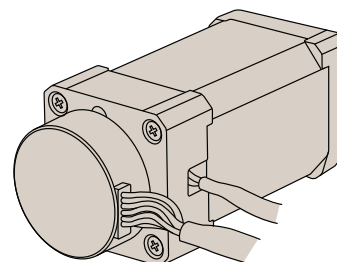


Intelligent stepper motor



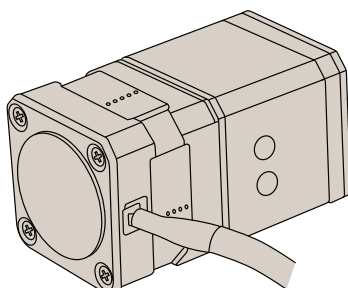
- Standard
- With rotary encoder (512 line)

Stepper motor

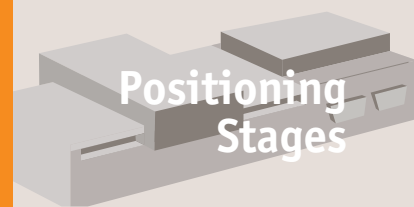


- Standard
- With rotary encoder (1000 line)

Servo motor



- Standard
- With rotary encoder (1000 line)



Controllers



L3294 Single axis stepper motor controller

- Communicate via RS-232 or Ethernet interface
- Uses virtually any programming language



L3295 Two axis stepper motor controller

- Communicate via RS-232 or Ethernet interface
- Programming via Labview, VB, C++ and OSX etc.
- Stand alone programs can be downloaded
- Max output of 1.5A



L3296 Multi axis stepper motor controller

- Communicate via RS-232 or Ethernet interface
- Can control 4 axis and perform coordinated or independent motion of each or all the axis simultaneously
- Uses virtually any programming language



L3297 Single axis servo motor controller

- Communicate via RS-232 or Ethernet interface
- Uses virtually any programming language

Accessories



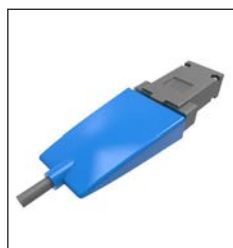
Joysticks



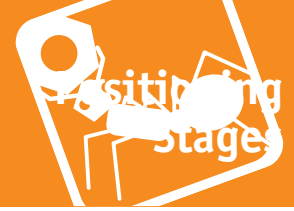
Digital readout



Connector RS232-USB



Connector RS422-USB



These have major benefits as they combine the motor (from size NEMA17 up) with an inbuilt driver and controller.

- Stepper or servo motor versions.
- Simple to install
- CE certified
- Free software programming

Plug and play

- Download free software
- Connect motor to computer (USB port)
- Connect power supply to the motor
- Start controlling/programming

- Low cost solution.
- The I/O points can be set by users to input, output or analogue input.
- NEMA17, 23, 34, 43 and larger sizes available.
- 12-48VDC.
- High torque stepper motors (1.2 to 10.5 Nm).
- Simple Windows software program provided free).
- Also Labview VB etc. programs.
- IP67, Motor brake.
- Optional Joysticks.



Setup and Programing

Software for NEMA23-43 motors

Positioning Stages

Positioning Stages from Automotion Components

MOTORISED LINEAR STAGES

Setup save/open
The complete setup can be either saved or reloaded from a file using these buttons.

System control
Use these buttons to save data permanently. Reset the motor etc.

Error Handling
Use these fields to define error limits for the position range etc.

Motor status
This field shows the actual motor load, position and speed etc.

Run status
Shows what the status of the motor is. The Bus voltage for the motor and the temperature of the driver is also shown.

Inputs
The status of the digital inputs are shown here and the analogue value.

Outputs
The status of the outputs are shown here and can be activated by the cursor.

Errors
If a fatal error occurs information will be displayed here.

Warnings
Here different warnings are shown.

Help line
Left area: If parameters entered are outside their normal values, errors are shown here.
Right area: Here it is possible to see if a motor is connected, the type, version and serial no.

Startup mode
The basic functionality of the unit is setup in this field

Profile data
All the main parameters for controlling the motor behaviour are setup in this field.

Driver parameters
These fields are used to define standby and running current.

Gear factor
The gear ration can be entered here.

Motion parameters
The distance the motor has to run is entered here.

Zero search
All the parameters regarding the position zero search can be specified.

Autocorrection
The parameters used to get the correct position if it is a motor with encoder.

Communication
The actual address of the motor can be entered

Introduction

The software is the main interface for setting up the motor for a specific application. The program offers the following features:

- Choice of the operating mode of the motor
- Changing main parameters such as speed, motor current, zero search type, etc.
- Monitoring the actual motor parameters
- Changing protection limits such as position limits.
- Saving all current parameters to disc.
- Restoring all parameters from disc.
- Saving all parameters permanently in the motor.

Command toolbox description

The toolbox used for the programming covers 14 different command types. The idea for the commands is to have an easy access to the most common functions in the motor. Some functions seem to be "missing" at first sight but the button "Set register in the quickstep motor" or "Wait for a register value before continuing" gives direct access to 50 function registers. In total this gives a very powerful programming tool since >95% of a typical program can be built using the simple command icons and the last part is obtained by accessing the basic motor registers directly if required.

Inserts a remark/comment in the program source code

Set the motor in the desired mode such as position or velocity mode.

Set a certain state at one or multiple digital outputs.

Conditional jump from one program line to another. Input dependent.

Wait for a certain state at one or multiple digital inputs.

Conditional jump from one program line to another. Input dependent.

Save the actual motor position to an intermediate register.

Initiates any motor movement relative or absolute.

Unconditional jump from one program line to another.

Inserts a delay in the program specified in milliseconds.

Write a value to almost any register in the basic unit.

Wait for a certain state at one or more of the digital inputs.

Initiates a zero search to a sensor or a torque (no sensor).

Preset the position counter to a certain value.