

Using SimplexMotion integrated motors for CNC machines

This document describes how to use the SimplexMotion integrated motor units in CNC applications. In the described setup the motors utilize a simple interface using [direction] and [step] signals. This interface is the standard for interfacing step motors that are common in low cost CNC applications. We also add signals [enable] and [stop] to handle error conditions. Replacing step motors with this kind of closed loop servo motors increase performance significantly and offers some important added functionality, such as stopping the machine if one motor is overloaded.

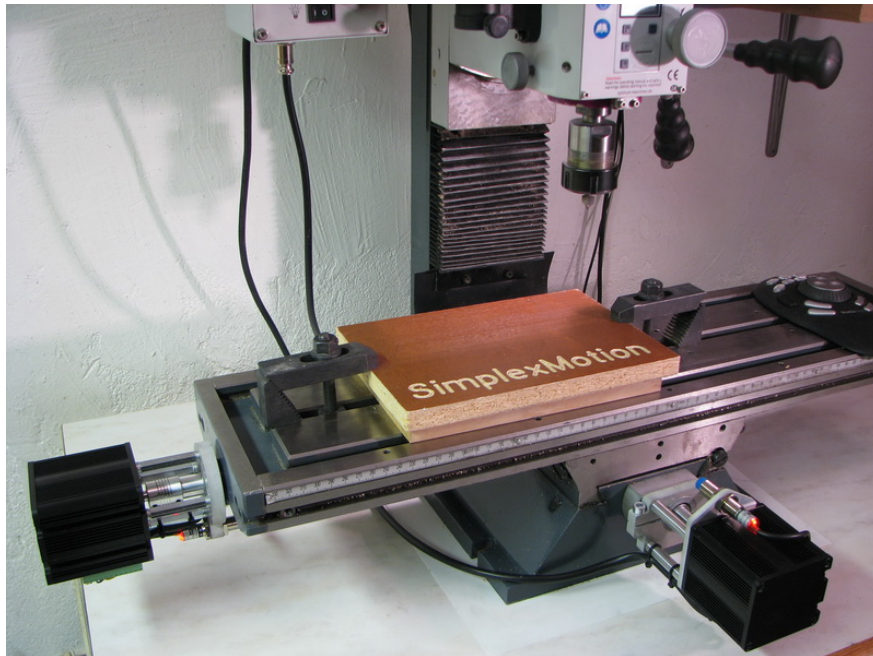


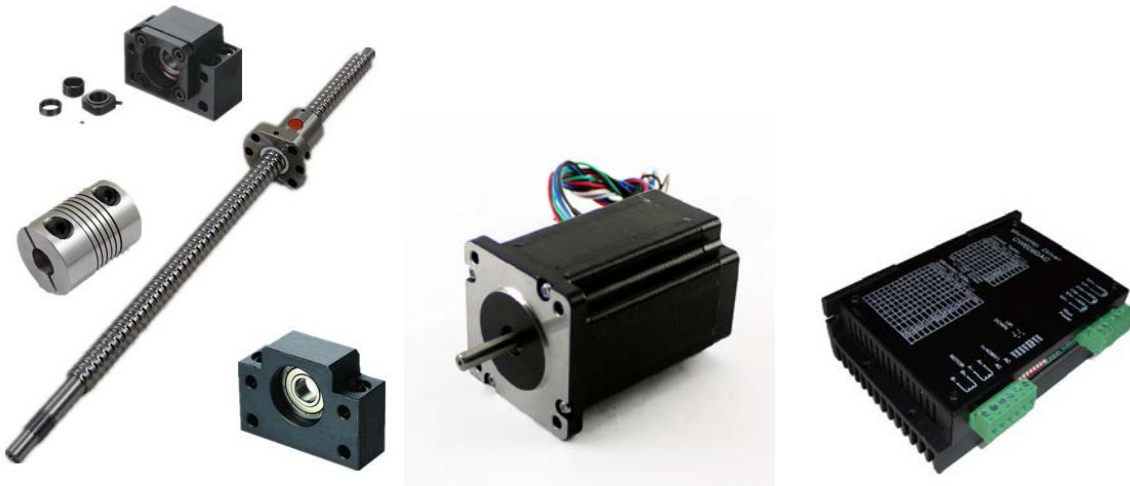
Table of contents:

| | | |
|-------|---|----|
| 1 | Low cost CNC machines with PC computer control..... | 2 |
| 2 | Servo motors versus step motors..... | 3 |
| 3 | The SimplexMotion integrated motor..... | 5 |
| 3.1 | Software tool for motor configuration..... | 5 |
| 3.2 | Standard configuration for CNC use..... | 6 |
| 3.2.1 | Electrical connections..... | 6 |
| 3.2.2 | Motor operation..... | 6 |
| 3.2.3 | The available parameter files..... | 7 |
| 3.3 | Detailed configuration of the SimplexMotion unit..... | 7 |
| 3.3.1 | [step] and [direction] signals..... | 8 |
| 3.3.2 | [enable] and [stop] signals..... | 8 |
| 4 | Check motor operation..... | 10 |
| 5 | Future work – new possibilities..... | 11 |
| 6 | Further information..... | 11 |

1 Low cost CNC machines with PC computer control

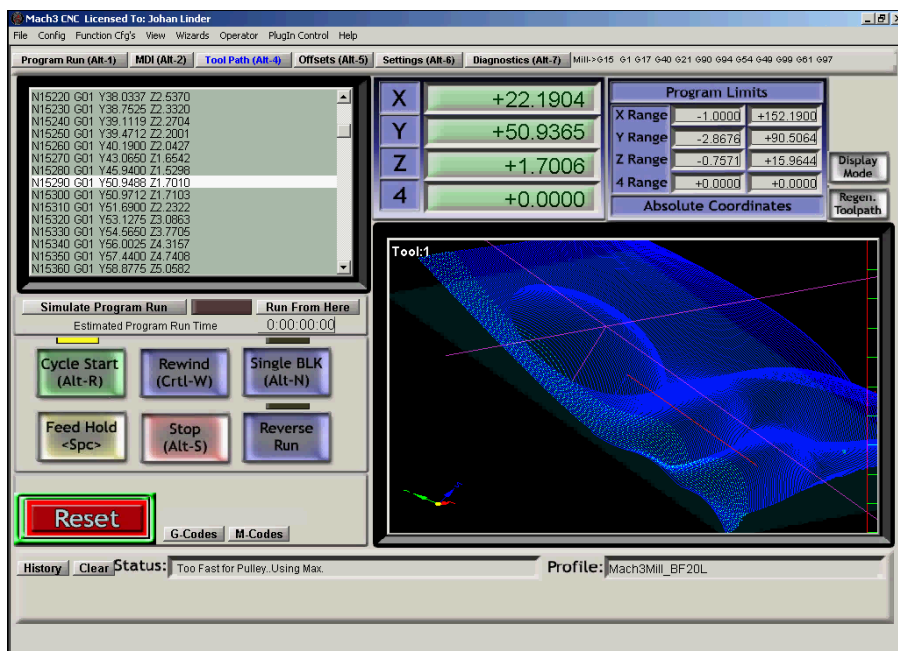
Mechanical production machines such as mills and lathes are generally controlled from computers for fast and efficient processing. These machines are called CNC (Computer Numerical Control) machines. The technology needed for automatic computer control has quickly become affordable making it appropriate to fit prototyping machines and other not so often used equipment with CNC abilities.

Today there is a large range of components for CNC machine building and conversion available on the market. Many of these products are marketed on the internet and sold at very low prices. This includes both linear transmission components such as ball screws, linear bearings, step motors, driver electronics and so on.



Another important part of this trend is the development of PC computer software to support the use of CNC equipment. There is a selection of low cost software available that implements motor control from manual input or programmed machine control. The g-code standard set of instructions for programmed machining are used, generated from standard CAD/CAM software packages.

One of the most used software for CNC machine control is Mach3 available from <http://www.machsupport.com/>. There is also free software for the Linux platform available at <http://www.linuxcnc.org/>



This picture shows the screen of the Mach3 software running a 3D milling sequence.

These systems provides direct motor control by use of [step] and [direction] signals straight from the PC computer parallel port. These signals are then used by the motor drivers to control the motor currents. This is a simple interface that is easy to use. However it requires real time capabilities from the computer, something that is not always the case with operating systems like Microsoft Windows. This limits the precision and speed of the drives, at least for slower computers. This fact, in combination with the outdated parallel port interface that has been removed from the standard PC peripherals since long, has led to the development of external hardware to create the [step] and [direction] signals. There are a number of such products available on the market, such as the [Smoothstepper](#) and the [UC100 motion controller](#).

But using a fairly modern PC computer and supplementing it with a plug-in parallel port interface card still provides a good cost/performance ratio and will provide adequate performance for many applications.

To connect the PC parallel port to the motors and sensors a breakout board is commonly used. It usually has screw terminals for easy wiring, and in some cases optocouplers for isolation to minimize the risk of damaging the PC computer if something goes wrong.

But please note that the low cost breakout boards with isolation often use low speed optocouplers that may limit the rate of the step pulses, and hence the motor speed.

For applications with SimplexMotion servo motors the opto isolation is not really necessary, and direct connection from the parallel port to the step/direction motor inputs can be used. A series resistor of typically 100Ohm will allow some protection.

Here follows an example of the PC parallel port (DSUB25 connector) pinout and usage, although most softwares allow custom configuration of pin assignments.

| Pin nr | Signal direction | Use |
|--------|------------------|--|
| 1 | Out | X axis step pulse |
| 2 | Out | X axis direction |
| 3 | Out | Y axis step pulse |
| 4 | Out | Y axis direction |
| 5 | Out | Z axis step pulse |
| 6 | Out | Z axis direction |
| 7 | Out | A axis step pulse |
| 8 | Out | A axis direction |
| 9 | Out | Enable (enables all motors) |
| 10 | In | X axis limit/home switch |
| 11 | In | Y axis limit/home switch |
| 12 | In | Z axis limit/home switch |
| 13 | In | A axis limit/home switch, or spare input |
| 14 | Out | Extra output 1 |
| 15 | In | Quickstop (to stop/disable the motion) |
| 16 | Out | Extra output 2 |
| 17 | Out | Extra output 3 |
| 18-25 | - | GND |

This application note will describe how the SimplexMotion integrated motor fits in this use, and we will not discuss the selection of software or hardware to generate the [step] and [direction] signals further.

2 Servo motors versus step motors

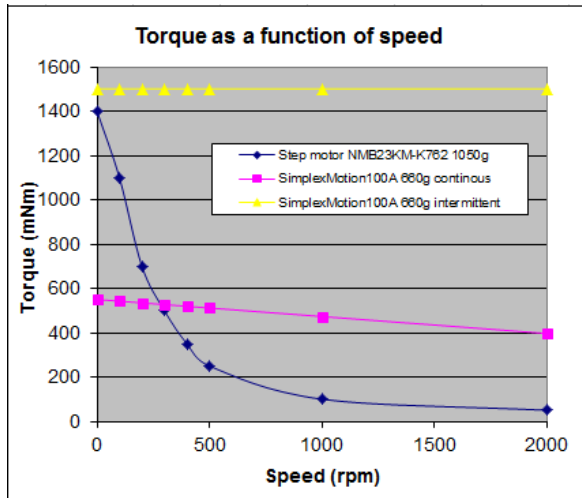
Most low cost mill conversions utilize step motors, since they are very affordable and easy to use.

However these motors have a few drawbacks:

- Step motors are used in an open loop fashion, there is no position feedback. This means that if the motor loses some steps it will have a position error for the remainder of the working session.
- They have a large holding torque, but the torque falls off quickly as the rotational speed is increased. This makes it difficult to attain high speeds.
- The step nature of the motor creates vibrations and noise at certain speeds which can be annoying and in some cases affect the finish of the machined parts.
- They consume lots of electrical power and get quite hot, since they need continuous current to maintain torque.
- The step motors are fairly heavy and large compared to servo motors.

In high end applications servo motors are used to overcome these issues. The servo motor system uses a motor and a position encoder to measure the true position. The driver controller compares the actual position with the target position and controls the motor torque to minimize this difference, thus closing the control loop. This allows the motor to consume power only when necessary and error conditions such as tool/work piece collisions can be detected and cause the machine to stop.

Servo motors can also maintain a high torque at high speeds, at least for brief moments. They also have a higher power-to-weight ratio and usually run much quieter.



This diagram shows a comparison of motor torque between a typical step motor and the SimplexMotion100A servo motor.

One reason the servo motors have been so much more expensive is the added electronics needed for their control. But due to the ever increasing cost efficiency of electronics this is beginning to change. With integrated motor control units, where the electronics is integrated with the motor, the added cost for servomotors is quickly decreasing.

One example of this is the SimplexMotion concept. It is a small and cost optimized servo motor system with integrated electronics. It is based on a high torque outer rotor brushless motor and a novel patented technology for motor position sensing. It has a USB port for control and configuration, but also offers other means of control. One of these is a 'step/direction' interface that is compatible to most step motor drives. This makes it easy to use for CNC applications. The SimplexMotion100A model is capable of 100W continuous output power and up to 1.5Nm torque, which is suitable for direct drive of ballscrews on CNC milling machines. Since torque can be maintained at speed much better than step motors, and the use of position feedback, this motor can replace step motors rated at holding torques of 4-6Nm.

More information on these motors is available at www.simplexmotion.com.



SimplexMotion motor units.

3 The SimplexMotion integrated motor

The SimplexMotion units allow several ways of controlling mechanical motion. For CNC applications there is a Step/Direction interface making the units directly compatible with typical step motor drives. The basic motor resolution is 4096 pulses per revolution, allowing high precision. But by configuration we can change the ratio of input step pulses to motor positions.

Servo motors also allow more functionality than step motors, such as monitoring the motor torque and stopping the machine if it goes above a certain threshold. This requires some more signals and configuration to be done.

Since the available flexibility allows many configuration options, it can be regarded as a complicated process of setting up the motors for proper operation. But most applications can use a default standard setup, and there are therefore standard parameter sets available to simply download to the motor.

3.1 Software tool for motor configuration

The SimplexMotion motor units are interfaced using USB or RS485 Modbus connection. A support software called 'SimplexMotionTool' is available at www.simplexmotion.com to allow access to a set of registers in the motor unit that defines its behavior.



The 'register' tab of the software is continuously updated with the register contents once the communication has been established using the 'start' tab. Please consult the software manual for further information on how to use the software. When the registers are updated by editing the numeric values they are immediately updated in the working registers in the motor unit. But make sure to select 'Write to memory' when finished as the register contents will be lost when the power is turned off otherwise. There are also buttons to read the register contents from a file, or to write all parameters to a file for backup.

The 'run' tab in the software is useful for general motor testing, but please note that this will change the registers and the motor settings, which will require reconfiguration for CNC use with step/direction input etc. The same goes for the 'position' tab.

Using the standard parameter files requires the following workflow with the software:

1. Get the parameter file to use from www.simplexmotion.com.
2. Start the SimplexMotionTool software.
3. Connect the motor unit to the PC with a USB cable.
4. On the 'Start' tab, select 'Connect USB'. The text in the bottom status field should show the connection in green text.
5. On the 'Register' tab, select 'Read from File'. Select the parameter file to download to the motor unit.
6. Select 'Write to memory' to permanently store the new parameters.

3.2 Standard configuration for CNC use

To avoid the detailed configuration issues there are standard setups available as parameter files. These can be downloaded from www.simplexmotion.com.

We will describe this standard setup and the required connections here.

3.2.1 Electrical connections

The following table lists the connections available on the SimplexMotion100A product, and the required signals for CNC use according to the standard settings.

| Motor connection | Signal | Description |
|------------------|-------------|---|
| IN1/OUT1 | Enable | Active low enable input signal. Pull low to activate motor. Input has pullup resistor. |
| IN2/OUT2 | Stop | Active low stop output signal. Pulsed low at error. Uses internal pullup resistor + open drain output which makes it possible to directly connect this signal from several motors for one combined stop signal. |
| IN3/OUT3 | | |
| IN4/OUT4 | | |
| IN5/ENCA | Step | Step pulse input. Changes motor position one step for each positive going transition. |
| IN6/ENCB | Direction | Selects motor direction. Low level is positive direction, defined as clockwise rotation looking at the motor front. High level is negative direction. |
| IN7/RS485A | | |
| IN8/RS485B | | |
| GND | Signal GND | |
| +5V (out) | | |
| GND | Supply GND | |
| +24V (in) | Supply +24V | |

The control signals are TTL compatible, and the voltage should be less than 0.7V for a low level, and higher than 2.4V for a high level. The [step] and [direction] inputs can withstand voltages up to 8V, while the [enable] and [stop] signals can handle voltages up to 30V.

Please note that some low cost opto isolated breakout boards does not provide TTL output levels with adequate noise margin.

In systems where the SimplexMotion units are mixed with stepper motors and/or other noisy motor control systems, extra care need to be taken to limit the electrical noise that couples into the low level signals to the motor units.

Screened cables should be used, and a noise limiting filter may be needed at the SimplexMotion motor inputs. This filter can be as simple as a small capacitor of 1-100nF connected to GND at the screw terminal.

There is also a software implemented filter for the step/direction inputs that can be configured with the <EncoderControl> register.

The recommended power supply is 24V with a current rating of 5A. For light loads on small machines it could be a lower rating, and for large machines that require fast acceleration and speed the rating should be increased.

Please note that for fast decelerations on heavy machine there will be a lot of mechanical energy transferred to electrical energy when the motors are breaking. This energy can cause the power supply voltage to rise to dangerous levels, since the absolute maximum supply voltage is 30V. In such cases there will have to be an overvoltage protection installed.

3.2.2 Motor operation

The indicator light on the motor shows its current state according to the following table:

| State | Indicator |
|---|--|
| Idle (deactivated or standstill) | Constant green |
| Running (motor moving or applying torque) | Flashing green (faster at higher speeds) |
| Error | Red light with yellow flashes. The number of flashes indicates the error source. |

When the [enable] signal is not activated, the motor will not be energized and the motor shaft can be rotated freely. The Step/direction interface is disabled and [step] signal pulses will not change the target motor position. The target position is memorized until the next time the motor is activated.

When the [enable] signal is activated, the motor is energized and set in motion if the actual position is not equal to the target position. This motion will be limited in acceleration and speed. When the actual position is equal to the target position the motor will stop and the motor will hold this position. The Step/direction interface is now enabled and any [step] signal pulses will change the target position and the motor will immediately move to the new position.

If a motor error occurs the motor will automatically stop and signal the error on the motor indicator light. The [stop] signal will then be active for a short time period to signal the faulty condition. In normal case the overall control system then deactivates the [enable] signal to all motors to cause them to stop. Any motors with faults will continue to show the error source on the indicator while disabled. When the [enable] signal is activated again, all motors will be activated, even those that previously caused an error.

The following table shows the possible error sources and the indicator light for these errors:

| Error | Indicator (nr of flashes) | Description |
|---------------------|---------------------------|-------------------------------------|
| Internal error | 1 | Internal fault in the unit |
| Communication error | 2 | USB or Modbus communication error |
| Current error | 3 | Motor current is too high |
| Voltage error | 4 | Supply voltage too low or too high |
| Temperature | 5 | Motor temperature is too high |
| Torque error | 6 | Motor torque is above set threshold |

3.2.3 The available parameter files

There are several parameter files available with some alterations. One common thing that will differ between machines is the required steps per motor revolution. Using the largest possible step count of 4096 per one revolution will provide the maximum resolution. But in some cases the maximum step pulse rate is limited, and that may require a lower step count per revolution to reach the target speed of the machine.

Available parameter files:

| File | Description |
|--------------------------------|--|
| Param_SM100A_Cnc_EnSt_4096.csv | With highest possible resolution, 4096 steps/revolution. |
| Param_SM100A_Cnc_EnSt_2048.csv | With 2048 steps/revolution |
| Param_SM100A_Cnc_EnSt_1024.csv | With 1024 steps/revolution |
| Param_SM100A_Cnc_EnSt_800.csv | With 800 steps/revolution |

3.3 Detailed configuration of the SimplexMotion unit

Below we discuss each of these features separately for the users that want to customize these settings. But since there is quite some flexibility in motor configuration it will also require careful reading of specifications and some understanding of the inner workings. For users with more standard requirements we recommend using complete and tested setup parameter files that are available at www.simplexmotion.com. To fully understand the text below, also read the datasheet for the motor unit.



3.3.1 [step] and [direction] signals

The step/direction interface utilizes the encoder inputs on the motor unit, and requires the encoder interface to be configured for [step] and [direction] signals. The encoder interface can also handle quadrature encoder inputs. For noisy signals it is also possible to select filtering of the inputs.

The register <EncoderControl> is used to configure the encoder interface. See the datasheet for full explanation.

| Register | Value | Comment |
|------------------|-------|---|
| <EncoderControl> | 2 | Sets Step/dir mode and no filtering. For noisy environments use a value of 66 (enables filtering, max step rate 625kHz). |

Now the target motor position is available at the register <Encoder>. But we also have to select this value for use by the motor position control. The target value to use is selected by the register <TargetSelect>.

The target value can also be further manipulated, which is useful to change the resolution and speed of the motor. In many cases the maximum step pulse frequency is limited, and to achieve high motor speeds we can let one step pulse be equivalent to several motor positions.

The Encoder value is first multiplied with the <TargetMul> register and then divided by <TargetDiv> register to form the motor target position. By setting a large <TargetMul> value the target motor position will be large with a smaller number of input steps, and hence the usable motor resolution has decreased. But the motor speed at the same step rate has been increased.

| Register | Value | Comment |
|--------------------|-------------|---|
| <TargetSelect> | 5 | Select the encoder interface as the source of target motor position |
| <TargetMul> | 1 | Setup for maximum resolution; 4096 steps per revolution |
| <TargetDiv> | 1 | Setup for maximum resolution; 4096 steps per revolution |
| <TargetOffset> | 0 | Not used |
| <TargetMin> | -2147483647 | The lowest possible value |
| <TargetMax> | 2147483647 | The highest possible value |
| <TargetHysteresis> | 0 | Not used |
| <TargetFilter> | 0 | Not used |

The following table shows some settings for useful motor position resolutions:

| TargetMul | TargetDiv | Motor resolution (step pulses per revolution) |
|-----------|-----------|---|
| 1 | 1 | 4096 |
| 2 | 1 | 2048 |
| 256 | 125 | 2000 |
| 4 | 1 | 1024 |
| 128 | 25 | 800 |
| 256 | 25 | 400 |

3.3.2 [enable] and [stop] signals

To make sure that the motor does not cause motion when not intended, we use an [enable] signal that allows the motor to be energized. When the [enable] signal is passive we want the motor to rotate freely to allow service to the machine. The motor unit mode of operation is controlled by the <Mode> register. Thus we want this register to change depending on the [enable] input signal. We do this by connecting the [enable] signal to a general purpose digital input of the motor unit and define this behavior using a feature called events. These events have a trigger condition, and an operation that is completed when the trigger condition is true. This feature allows a flexible setup where most user requested behavior can be programmed.

The <Mode> register can also change automatically in case of an error cause within the motor unit. When an error defined by the <MaskShutdown> register is detected, the unit mode of operation is changed to 'shutdown'. In this mode the error cause is shown on the indicator as a blinking pattern. See the motor unit datasheet for further explanation and a list of error causes.

But we also want to make the PC software aware of such an error so that the other motors and possibly a machining program in use can be stopped. This is accomplished by the [stop] output signal. We implement a [stop] output signal pulse by using the pulsing feature of the general digital outputs on the motor unit.

To complicate things further we cannot turn motor position regulation mode on directly when the enable signal is activated. If the user has turned the motor shaft manually while the motor was disabled we will get a very high speed movement back to the previous target value stored in the motor. To make this movement at a controlled speed and acceleration we first set the motor mode to 'position with ramp control' that will perform the movement at set

acceleration and maximum speed. When the target position has been reached we switch mode to direct position control, which will improve precision. All this is done using events.

Finally we want to disable the encoder interface to ignore [step] signal pulses when the [enable] signal is not active. This is also handled by events.

| Register | Value | Comment |
|------------------|-------|---|
| <InputPolarity> | 15 | Sets IN1-4 to be active low inputs. Makes the [enable] signal active low. |
| <OutputControl1> | 4 | Enables pull up resistor on combined IN1/OUT1 |
| <OutputControl2> | 36 | Configures OUT2 to be a pulsed active low output with pull up resistor enabled. This output is used as the [stop] signal. |
| <RampSpeedMax> | 10000 | The maximum speed used at first when the unit is enabled. Unit is [positions/s * 1/16]. |
| <RampAccMax> | 500 | Acceleration used at first when the unit is enabled. Unit is [position/s^2 * 1/256]. |
| <RampDecMax> | 500 | Deceleration used at first when the unit is enabled. Unit is [position/s^2 * 1/256]. |
| <MaskShutdown> | 48 | Enables motor shutdown for over torque and high motor temperature. |

The events are also configured by registers according to:

| Nr | Event description | Register | Value | Setting description |
|----|--|------------------|--------|---|
| 1 | When the [enable] signal goes active, write 21 to <Mode> register (setting position control with ramping). | <EventControl1> | 0xb137 | Trigger: register AND literal, edge triggered, filtered |
| | | <EventTrgReg1> | 145 | <Input> register |
| | | <EventTrgData1> | 1 | |
| | | <EventSrcData1> | 21 | 21 = position control with ramping |
| 2 | Always write the value 21 to register X. | <EventControl2> | 0xb000 | Trigger = always |
| | | <EventTrgReg2> | 0 | |
| | | <EventTrgData2> | 0 | |
| | | <EventSrcData2> | 21 | 21 = position control with ramping |
| 3 | When the <RegError> register value is small (the actual position is close to the target position), write value 20 to register X. | <EventControl3> | 0xb003 | Trigger: register < literal |
| | | <EventTrgReg3> | 308 | <RegError> register |
| | | <EventTrgData3> | 32 | Value 32 |
| | | <EventSrcData3> | 20 | 20 = position control mode |
| 4 | When the <Mode> register = 21, write X register to <Mode>. This will change mode when the target and actual positions are the same. | <EventControl4> | 0xb0b1 | Trigger: register = literal, filtered |
| | | <EventTrgReg4> | 400 | <Mode> register |
| | | <EventTrgData4> | 21 | 21 = position control with ramping |
| | | <EventSrcData4> | X | Value is set by the previous two events |
| 5 | When <Mode> = 4, set register Y to 4. | <EventControl5> | 0xb001 | Trigger: Register = literal |
| | | <EventTrgReg5> | 400 | <Mode> register |
| | | <EventTrgData5> | 4 | 4 = Shutdown mode |
| | | <EventSrcData5> | 4 | |
| 6 | When <Mode> != 4, set register Y to 0. | <EventControl6> | 0xb002 | Trigger: register != literal |
| | | <EventTrgReg6> | 400 | <Mode> register |
| | | <EventTrgData6> | 4 | 4 = Shutdown mode |
| | | <EventSrcData6> | 0 | |
| 7 | When [enable] signal is deactivated, write register Y to the <Mode> register. Will turn off the motor or keep the shutdown mode. | <EventControl7> | 0xb136 | Trigger: register NOR literal, edge triggered, filtered |
| | | <EventTrgReg7> | 145 | <Input> register |
| | | <EventTrgData7> | 0xfffe | |
| | | <EventSrcData7> | Y | Value is set by the previous two events |
| 8 | When the <Mode> register becomes equal to 4 (shutdown mode) a value is written to the <Output2> register to cause an active low pulse. | <EventControl8> | 0xb131 | Trigger: register = literal, edge triggered, filtered |
| | | <EventTrgReg8> | 400 | <Mode> register |
| | | <EventTrgData8> | 4 | 4 = Shutdown mode |
| | | <EventSrcData8> | 4 | Specifies the length of the pulse in milliseconds. |
| 9 | When the [enable] input goes active, write 2 to the <EncoderControl> register. This will enable the step/direction interface. | <EventControl9> | 0xb137 | Trigger: register AND literal, edge triggered, filtered |
| | | <EventTrgReg9> | 145 | <Input> register |
| | | <EventTrgData9> | 1 | |
| | | <EventSrcData9> | 2 | Enable the encoder interface |
| 10 | When the [enable] input goes inactive, write 0 to the <EncoderControl> register. This will disable the step/direction interface. | <EventControl10> | 0xb136 | Trigger: register NOR literal, edge triggered, filtered |
| | | <EventTrgReg10> | 145 | <Input> register |
| | | <EventTrgData10> | 0xfffe | |
| | | <EventSrcData10> | 0 | Disable the encoder interface |
| | | <EventDstReg10> | 180 | Write to <EncoderControl> register |

4 Check motor operation

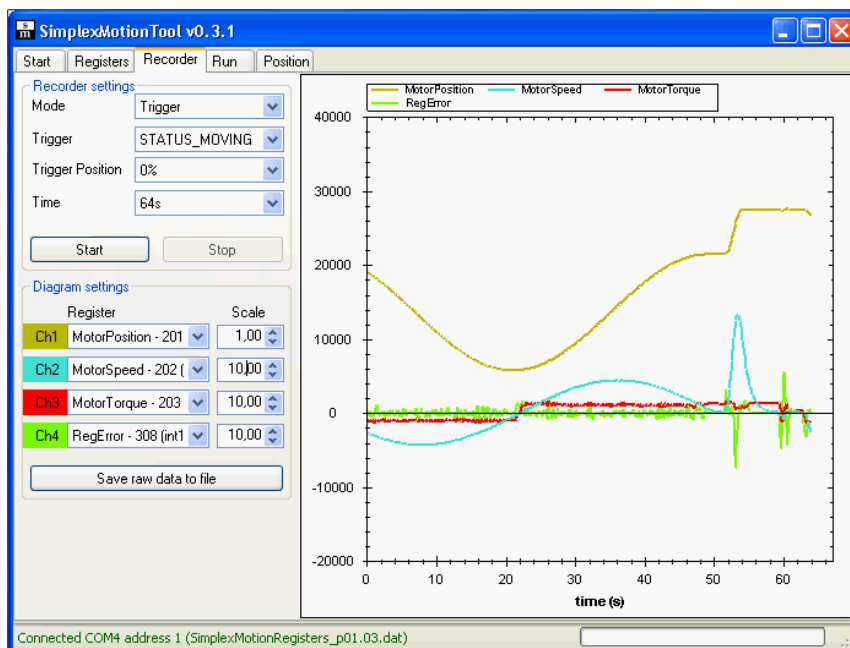
Using the SimplexMotionTool software it is possible to monitor motor data and take measurements of important parameters. It is also a valuable tool to find and correct problems.

The software connects to the motor using USB or RS485 Modbus communication, and this will work just as fine while the motor is running in the CNC application.

By utilizing the 'recorder' tab of the software it is possible to record 4 channels of data during actual motion. The speed of the recorder and triggering conditions can be changed to suit the requirements. Please read the SimplexMotionTool manual for further information.

Useful things to measure could be:

- The actual motor torque needed at different speeds and accelerations. The register <MotorTorque> provides the motor torque value in units of $[Nm * 1/100]$.
- The deviation between target and actual position. A servo motor system relies on motor position feedback and a regulator that continuously minimizes the deviation between the actual and desired position. But there is always a small difference between the two and checking the amount of this difference can be done by inspecting the <RegError> register. It shows the deviation in units $[positions * 1/16]$. One mechanical degree = $1/360$ of one revolution of the motor shaft thus corresponds to a value of 182.
- Verify that the supply voltage does not change significantly when operating the motor. This could be due to a weak power supply (supply voltage dropping) or due to excessive power feedback when braking the motors (voltage increasing at times). The voltage can be found in register <Supply>, and the unit is $[V * 1/100]$.
- Motor temperature will increase when the motor is heavily loaded for long times. Check the internal temperature by reading register <TempMotor>.



This picture shows a recording of motor position, speed, torque and regulator error while a circular move is performed. It can be seen that the required torque is less than 0.2Nm.

5 Future work – new possibilities

The step/direction interface is a widespread standard for interfacing the main CNC controller with the motor drivers. However there is a requirement to create the step pulses at a high rate with good precision to realize even motion. PC computers running operating systems like Microsoft Windows lacks the real time properties to do this reliably at high speeds. For limited step pulse rates it works fairly well, but for high speed CNC motion it does not. Therefore there are a number of external hardware options available that interface by USB or Ethernet to the PC computer, and handle the generation of the step pulses at higher speed and precision.

Since the SimplexMotion unit has a higher level of communication interface already, it would be possible to send motion data on a higher level and let the motor unit create the step pulses internally. This would allow full precision and full speed motion without the limits posed by step pulse generation from the PC computer.

Using the RS485 interface it is possible to network a large number of motors on the same connection. Work will be done later to implement plugin software to Mach3 and similar CNC control software to allow direct connection of motor units to the PC computer in a network fashion. This will strengthen the position as a low cost servo motor control systems for CNC applications further.

6 Further information

For a deeper understanding of the SimplexMotion100A product please consult the datasheet available at: <http://simplexmotion.com/support/documentation/>

There is also a document describing a mill conversion using the BF20L manual milling machine (a small and robust benchtop machine) using the SimplexMotion100A motors. The Mach3 software is used and the setup also includes spindle on/off and speed control.

A video also shows the conversion and the milling operations for this setup. All material is available at: <http://simplexmotion.com/support/application-notes/>

There is loads of information on CNC mill conversions on the internet. Here are some pointers:

- The Mach3 setup manual is a good text to understand the concepts of PC machine control. It can be found at <http://www.machsupport.com/documentation.php>
- The largest and most active forum for CNC machining is www.cnczone.com