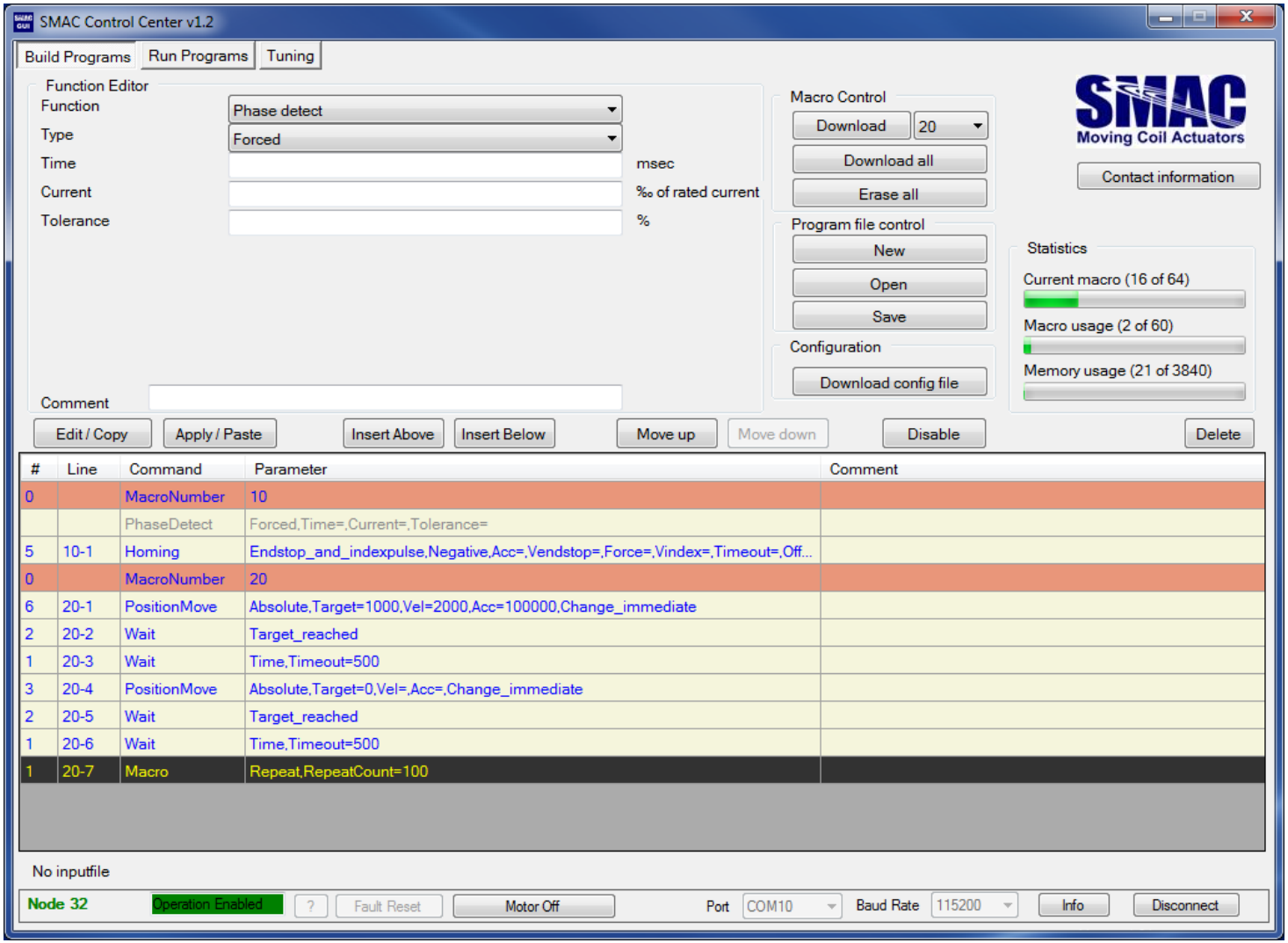
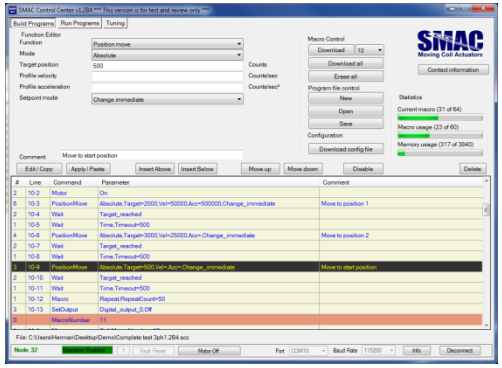
SMAC Control Center Training v1.2

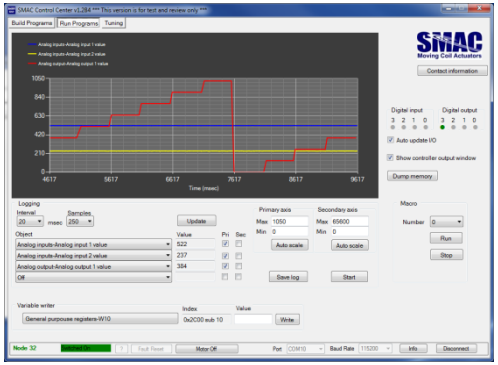


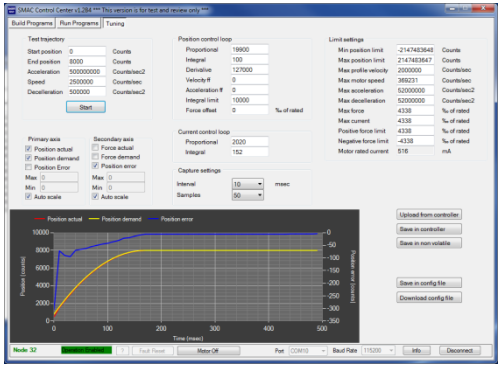
The Ability to do Work & Verify it's Accuracy at the Same Time

Main functions

- Easy creation of macros
- Download programs in controller
- Run macros
- Download existing configuration in controller
- Logging and charting
- Tuning of position and current control loops







The Ability to do Work & Verify it's Accuracy at the Same Time

PC Requirements

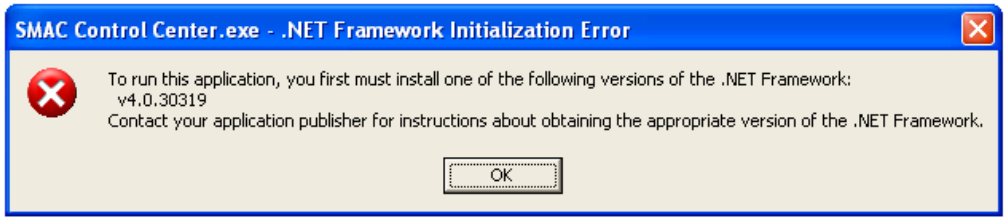
- 512MB Ram, 1GHz processor, 600 MB disk space (1.5 GB for 64 bit version)
- Windows XP, Vista or Win7, latest service packs must be installed.
- Serial port to communicate with the LCC (CAN not yet supported). For a USB to Serial adapter use one from the list at www.smac-mca.nl

LCC Requirements

- Firmware version 1.0 or newer.
- Baudrate must be 115200.
- Polarity for velocity and position must be set to positive (object 0x607E must be 0)
- Daisy chain mode must be set to daisy chain (object 0x2000 sub 3 must be 1)
- A warning message is shown after connection if one of these requirements is not ok.

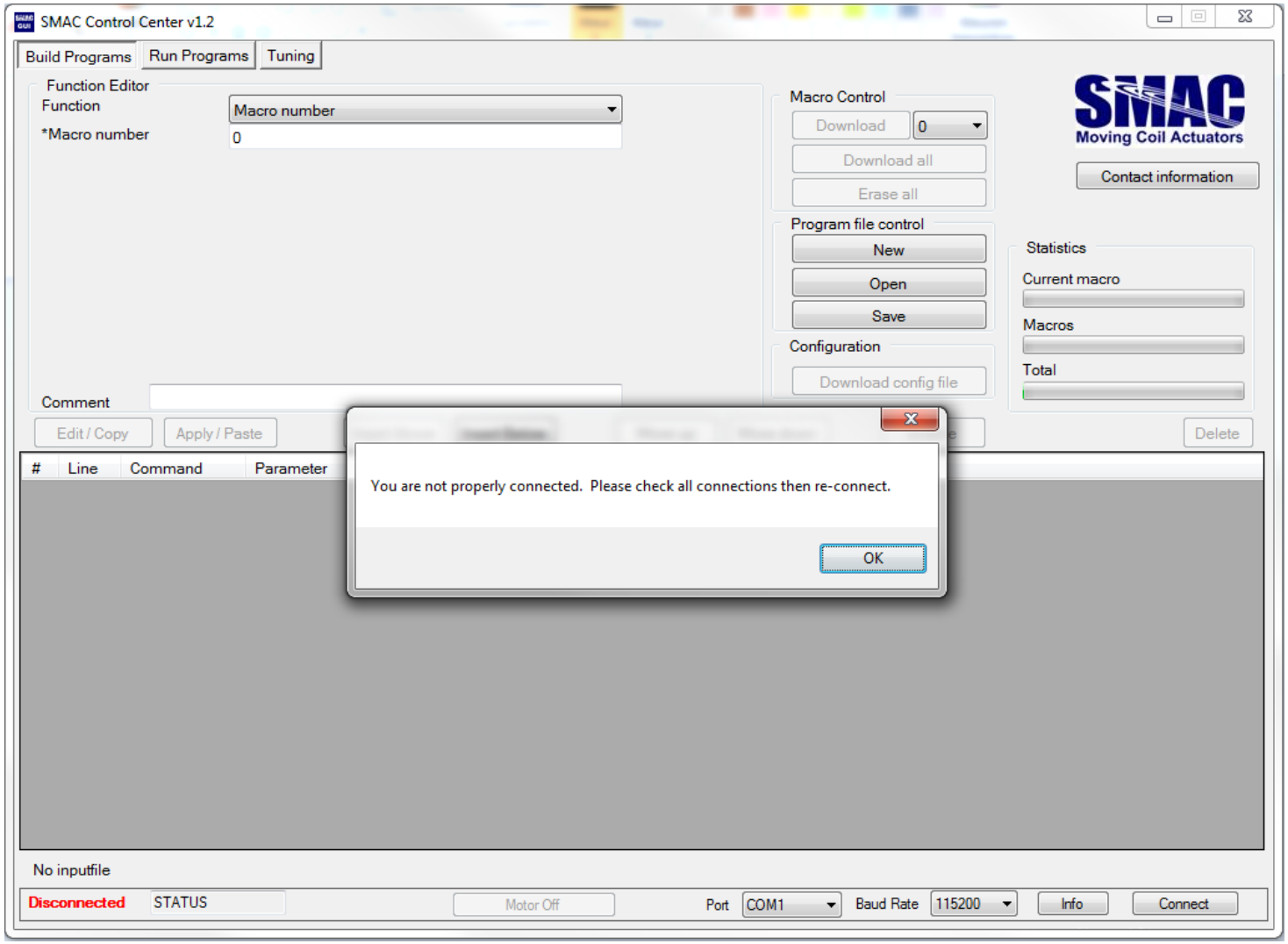
Installation

- Zipped installer with installation instructions and getting started document can be downloaded from www.smac-mca.nl
- If the following message appears when running SMAC Control Center:



then install the .NET framework from http://www.microsoft.com/en-us/download/details.aspx?id=24872

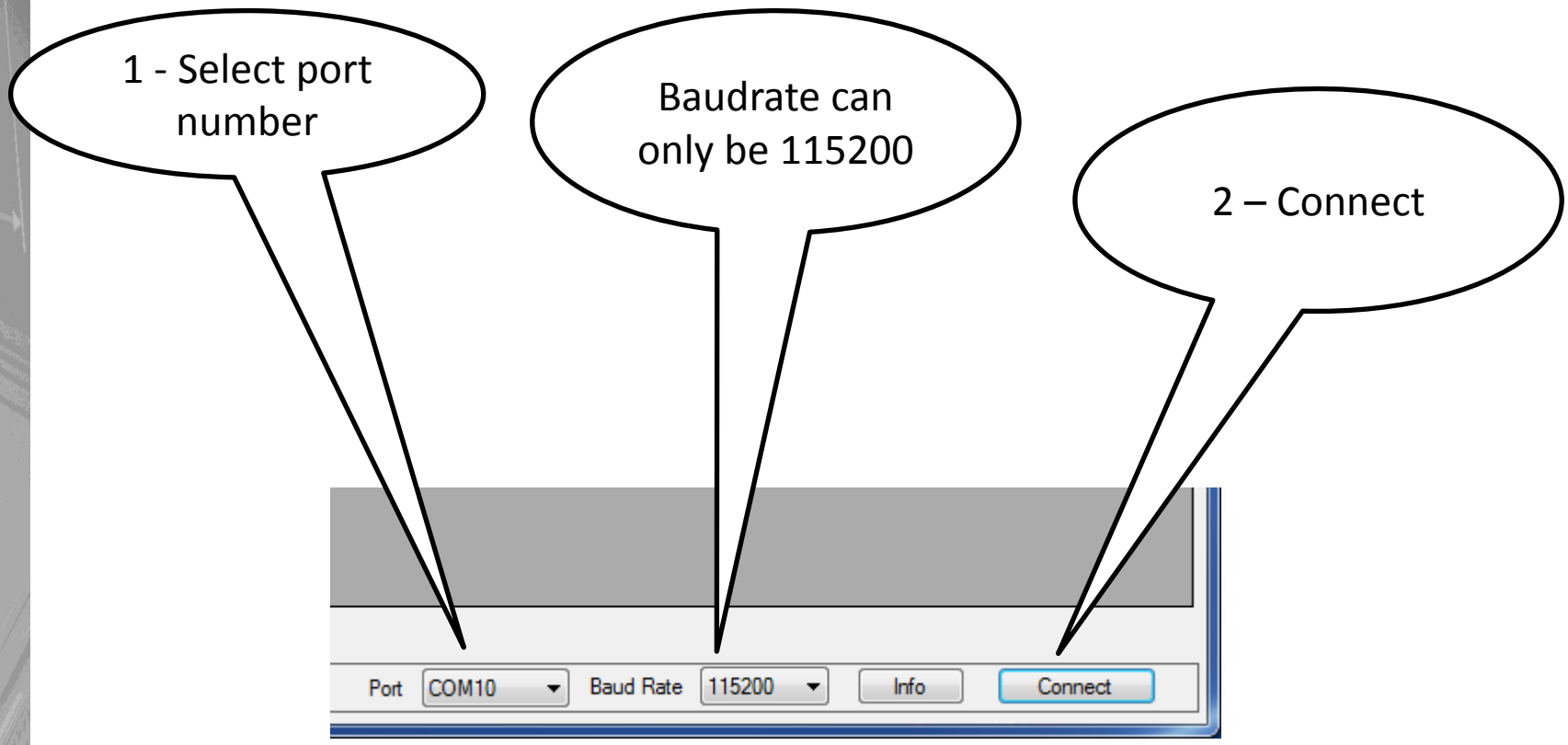
First start of the program



The Ability to do Work & Verify it's Accuracy at the Same Time

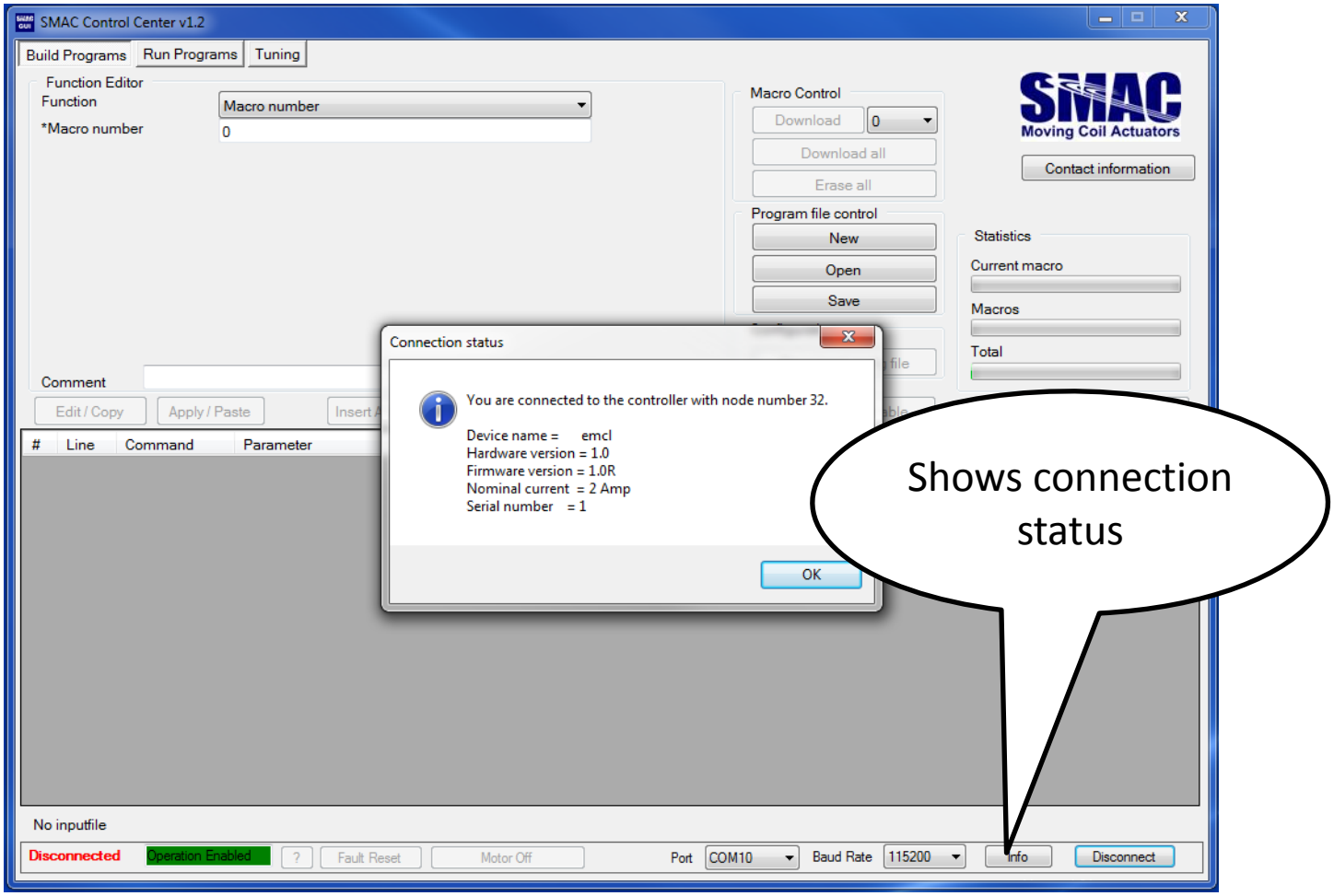
Connect with the controller

SMAC
Moving Coil Actuators



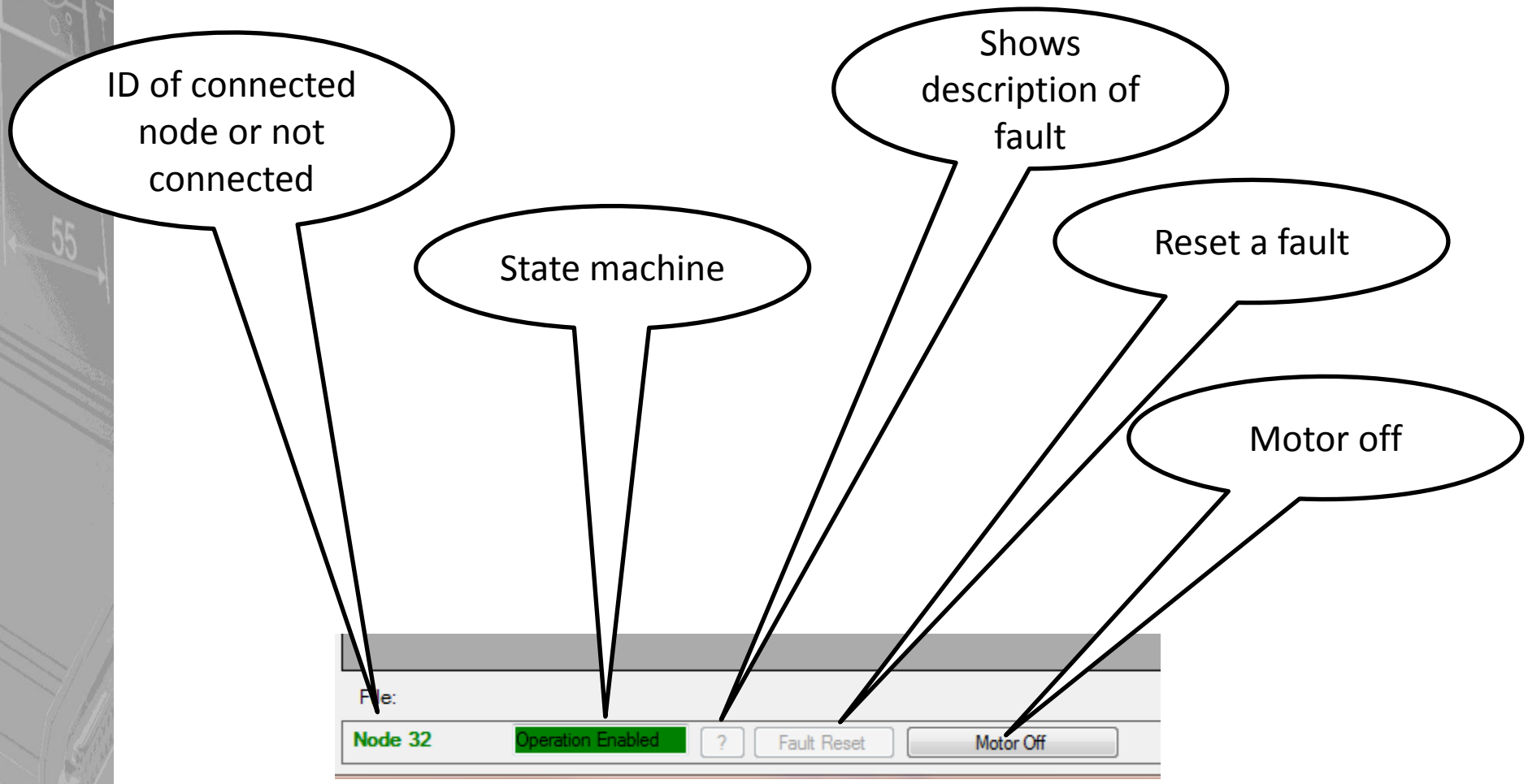
The Ability to do Work & Verify it's Accuracy at the Same Time

Connection status



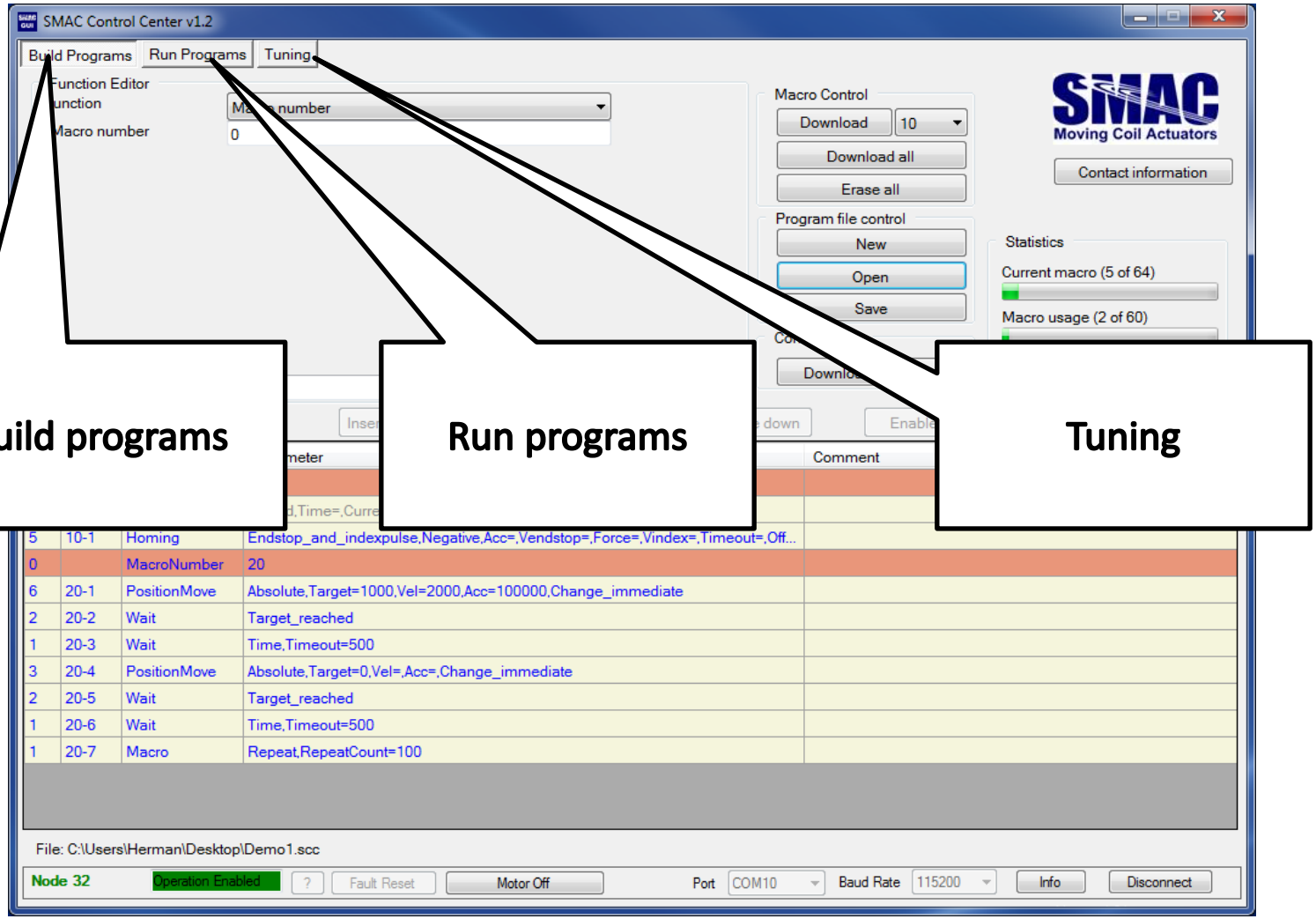
The Ability to do Work & Verify it's Accuracy at the Same Time

Status area



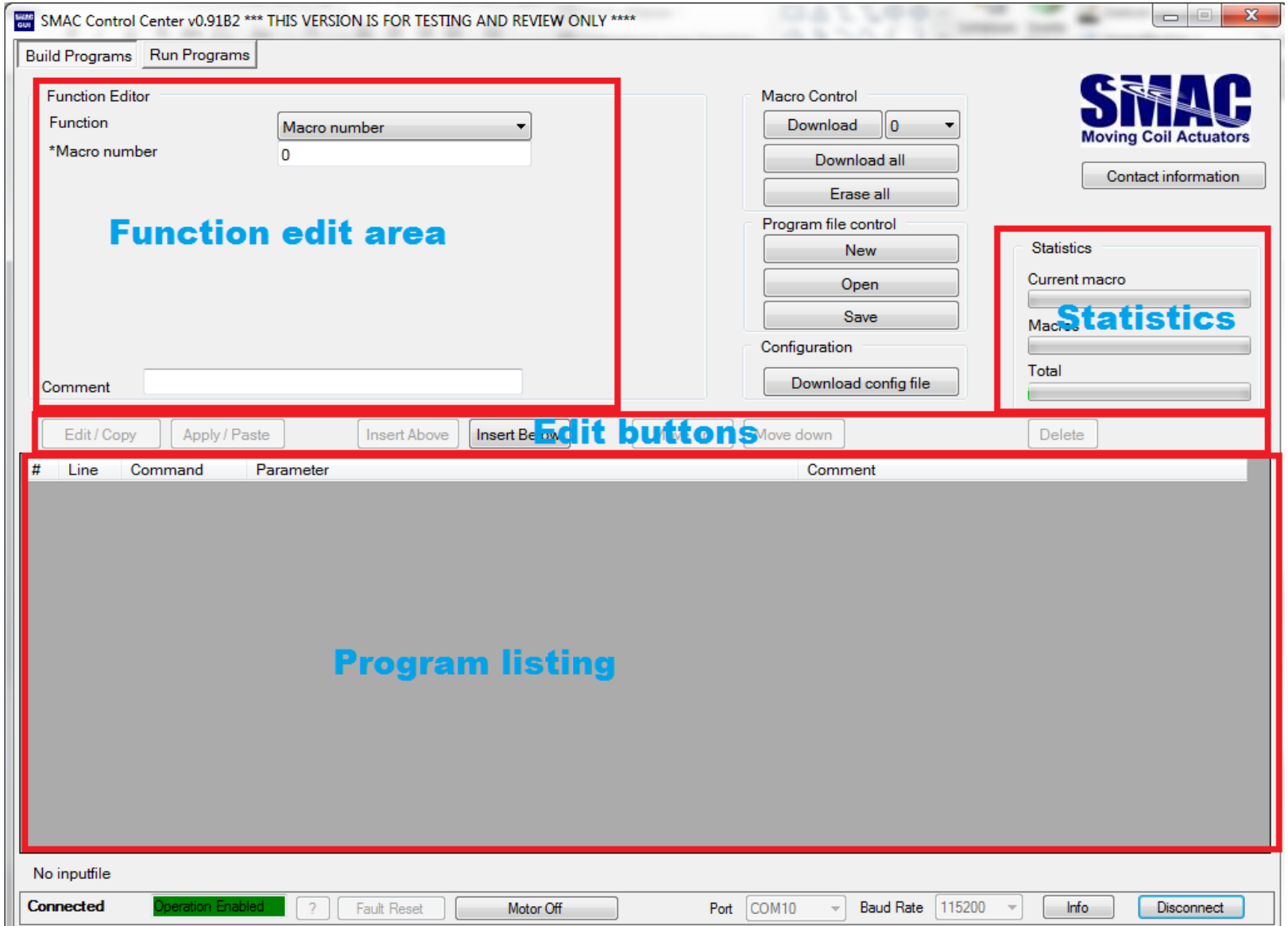
The Ability to do Work & Verify it's Accuracy at the Same Time

Main tabs

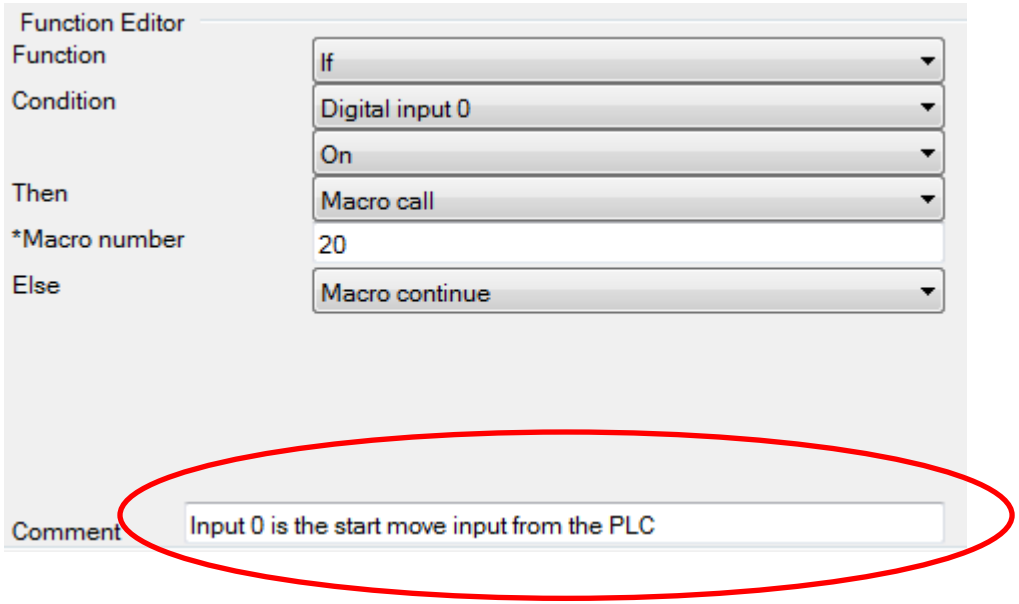


The Ability to do Work & Verify it's Accuracy at the Same Time

Function editor

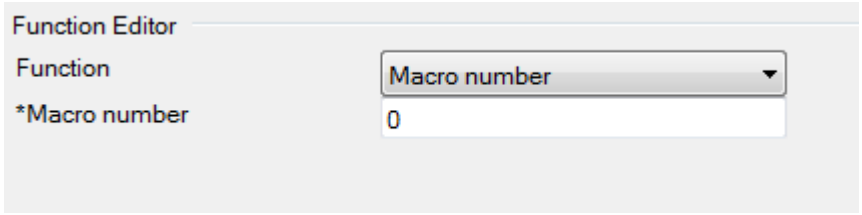


Function editor



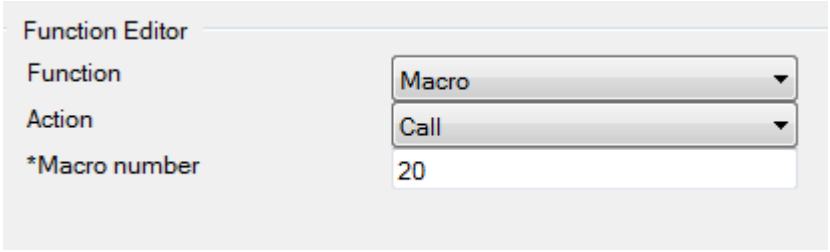
- Use the comment !!!
- Mandatory parameters are marked with *
- Optional parameters retain last used value

Function: Macro number



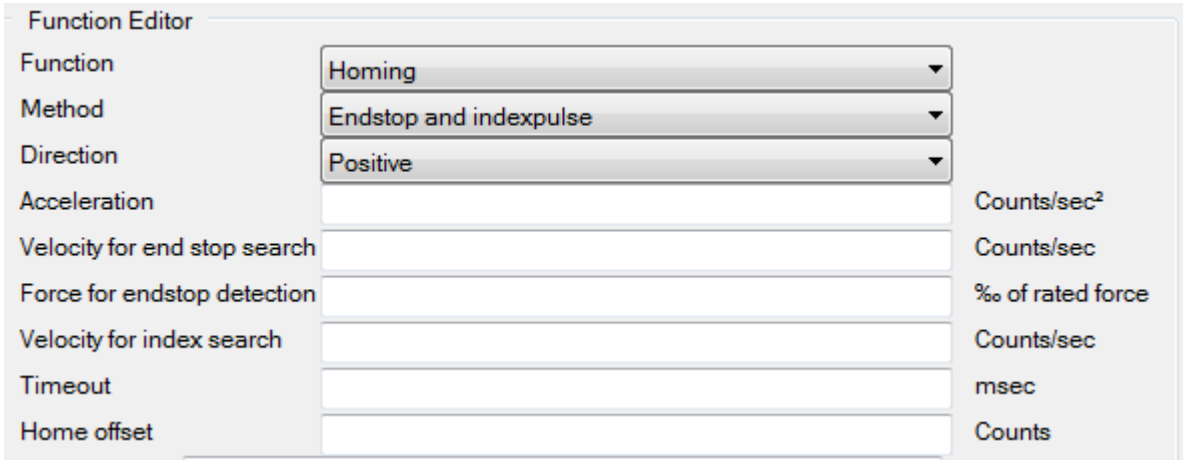
- Start of a macro definition
- Valid range for macro numbers is 0 .. 59
- Macro numbers 60 .. 63 are reserved for system macros

Function: Macro



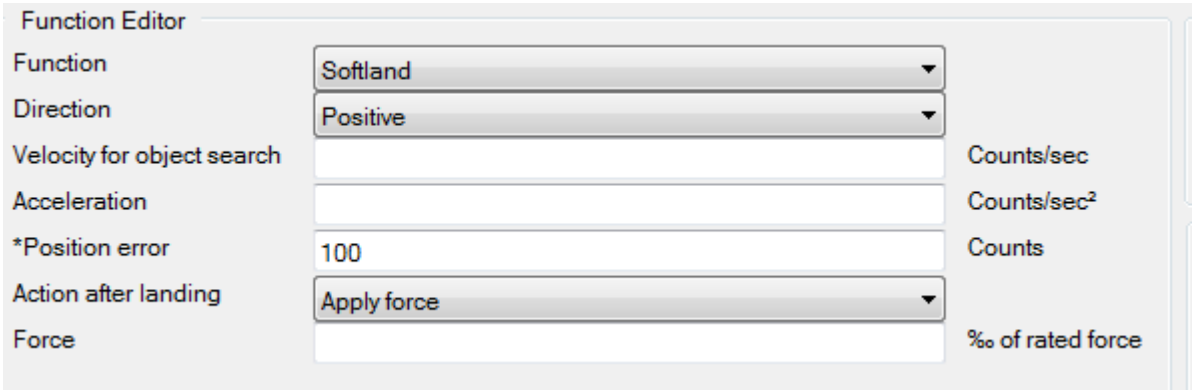
- Actions can be:
  Call: Call another macro. The called macro must end with a "Macro return" function.
  Jump: Jump to a macro
  Return: Return to the caller of this macro
  Repeat: Repeats the current macro from the start the number of times given. 0 means repeat forever (endless loop)

Function: Homing



- Methods: Endstop, Endstop and Indexpulse, Indexpulse, Current position
- Direction: positive or negative
- Rest of parameters depend on method

Function: Softland

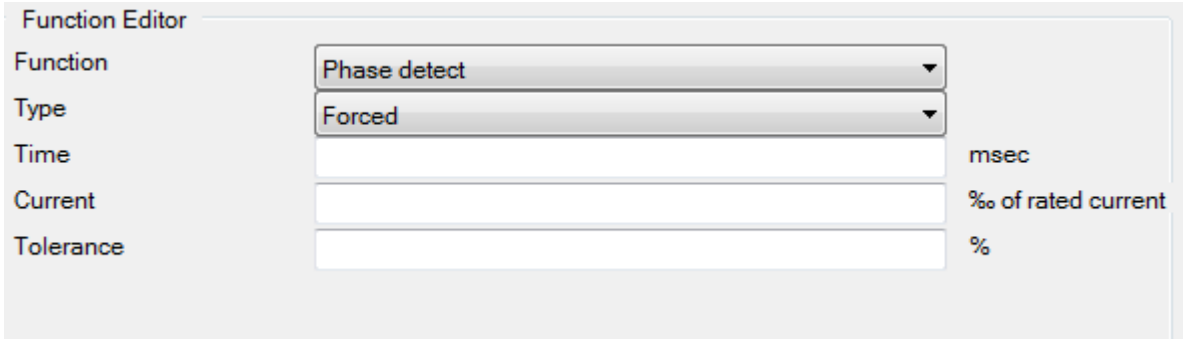


- Direction: positive or negative
- Action after landing: Apply force, Hold force, Hold position, Hold motion status, Motor off
- Starts profile position move with target set to +∞ or –∞. Detects landing if following error exceeds Position error parameter. Then performs Action after landing.

Softland – After landing actions

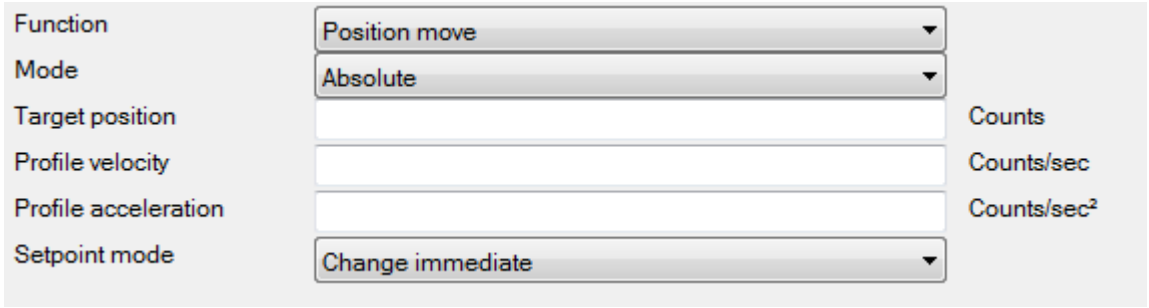
- Apply force: Switches to force mode and applies the force that is given as parameter
- Hold force: Switches to force mode and applies the force that was measured at the moment of landing
- Hold position: Keeps the actuator in position mode with the actual position at the moment of landing as target position
- Hold motion status: No action after landing. The programmer must define the actions to perform next in the following program lines. **If no actions are defined then the force will ramp up.**
- Motor off

Function: Phase detect



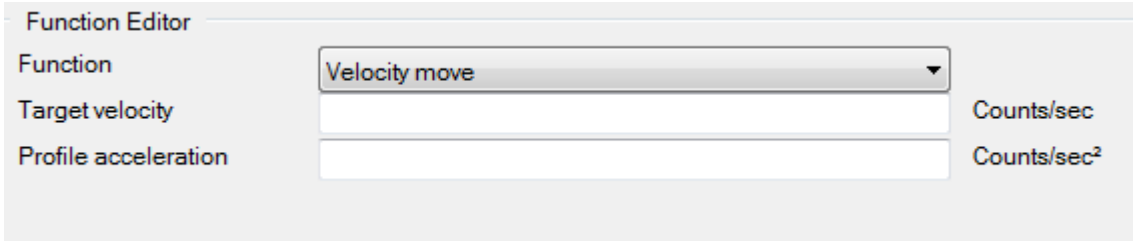
- For multipole actuators only
- Type: Forced or initial position always known
- Turns motor off
- Rewrites the motor type to force a new phase detect cycle.
- Turns motor on. This will activate the phase detect
- Waits until phasing is reached.

Function: Position move



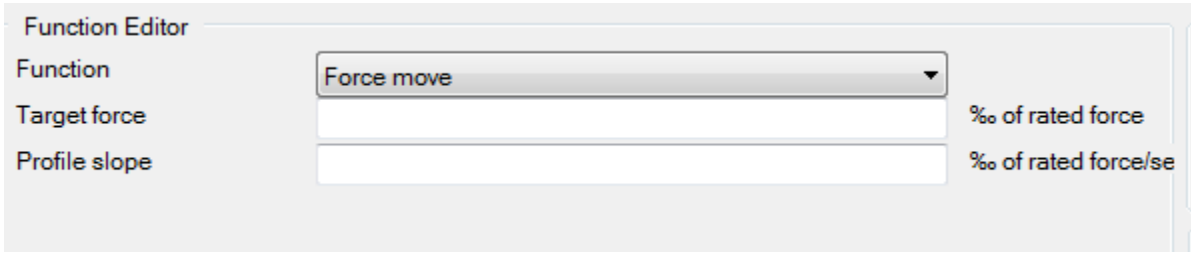
- Starts a move in profile position mode
- Mode: Absolute or relative (to target position).
- Setpoint mode: Change immediate or Add to queue (the queue has 16 entries)
- Does not wait for target reached. Use “Wait for target reached” or “Wait for trajectory generator ready” function to do that if necessary

Function: Velocity move



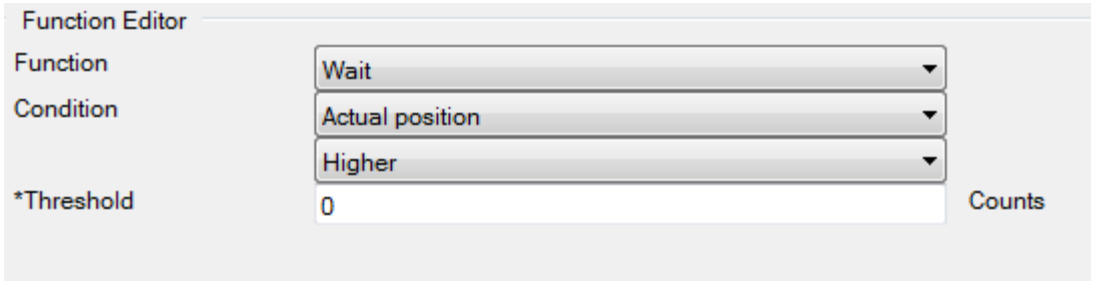
- Starts a move in profile velocity mode

Function: Force move



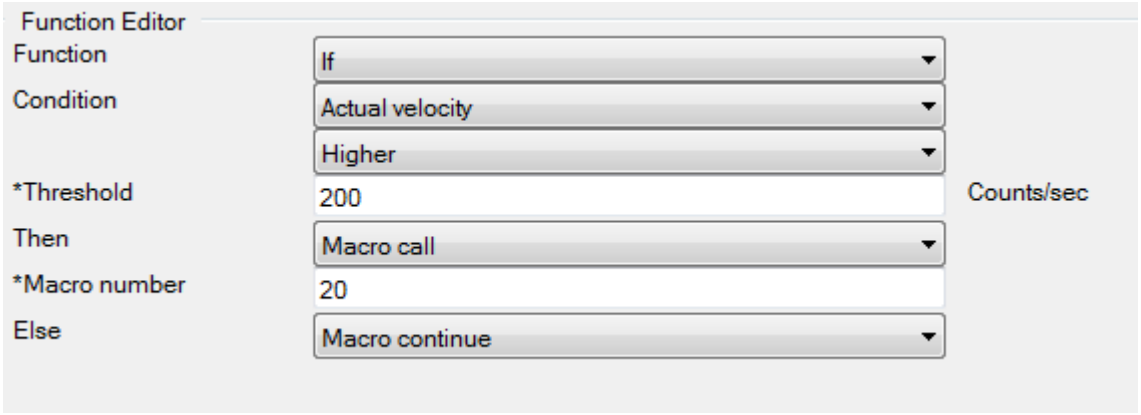
- Starts a move in profile force mode

Function: Wait



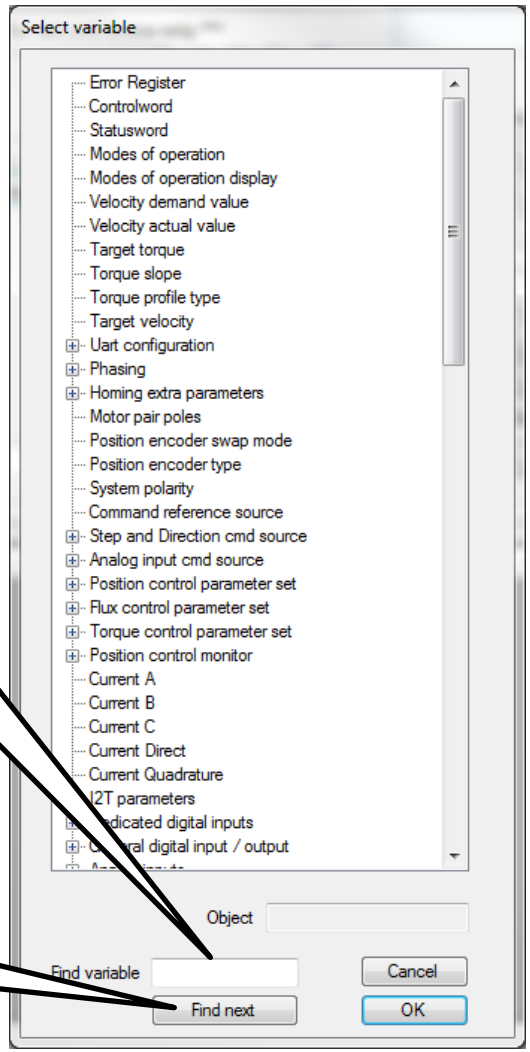
- Waits until a condition is reached
- Condition can be: Time (msec), Target reached, Trajectory generator ready, Index pulse, Actual position, velocity force or analog input > or < Threshold, Digital input on or off.

Function: If



- Condition can be: Target reached, Trajectory generator ready, Actual position, velocity force, variable or analog input > or < Threshold, Digital input on or off.
- Variable can be selected with “Smart object selector”
- Then and Else can be Macro jump, Macro call, Macro return or Macro continue

The smart object selector



Type (partial) name of object here

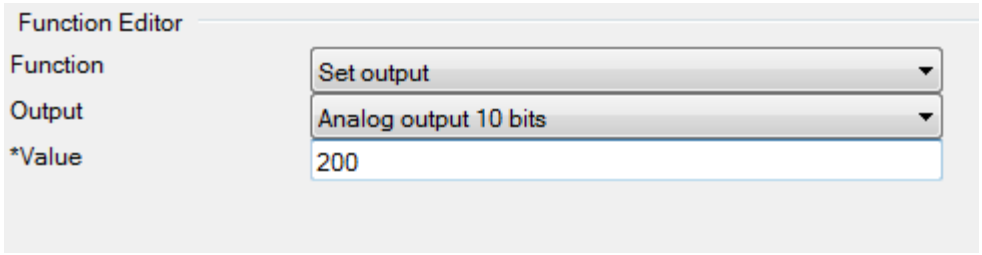
And hit [Find next] or [Enter]

Tips:
Also object addresses are allowed with or without leading 0x

Type only a few letters of the object. If you search for “position actual value” then “actual” is enough to find what you want.

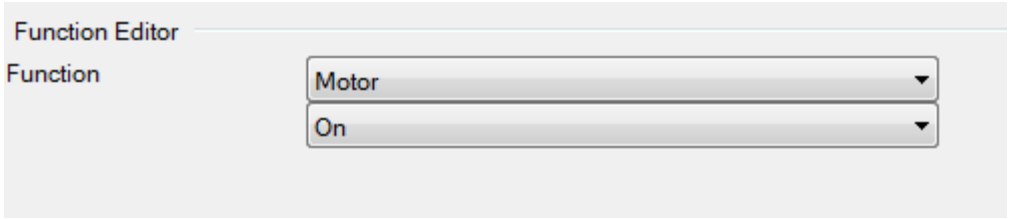
Hit Enter key to find next match.

Function: Set output



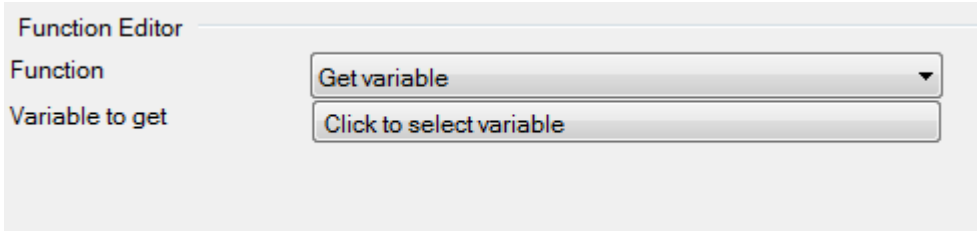
- Output can be: Analog output 10 bits, Analog output 16 bits, Digital output 0 .. 3.
- Analog outputs can be set to a value
- Digital outputs can be set On or Off

Function: Motor



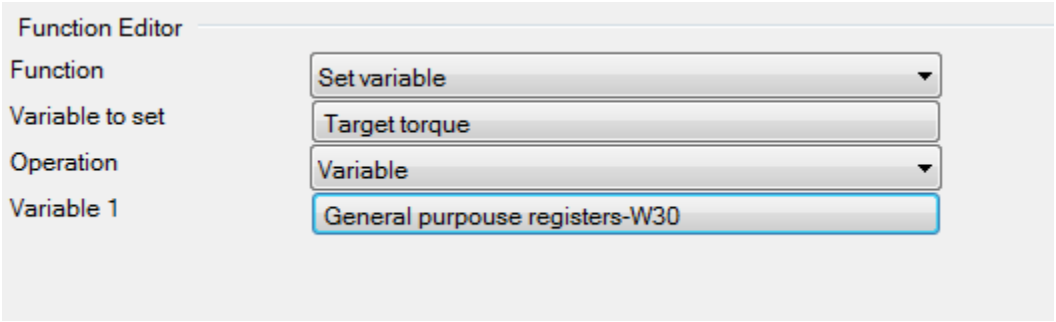
- Switches the motor On or Off

Function: Get variable



- Gets any readable variable in the controller and reports this variable on the serial port.
- Example: The controller will respond with:
  0x20 W 0x6064 1234
  on a get of the actual position.
- Variables can be selected with the “Smart object selector”

Function: Set Variable



- Sets any writeable variable in the controller to the result of an expression.
- The expression can be:
  - Constant
  - Variable
  - Add, subtract, multiply, divide, and, or, xor, shift left, shift right of two variables or a variable and a constant
  - Absolute or complement of variable
- Variables can be selected with “smart object selector”

Program listing area

Moving Coil Actuators

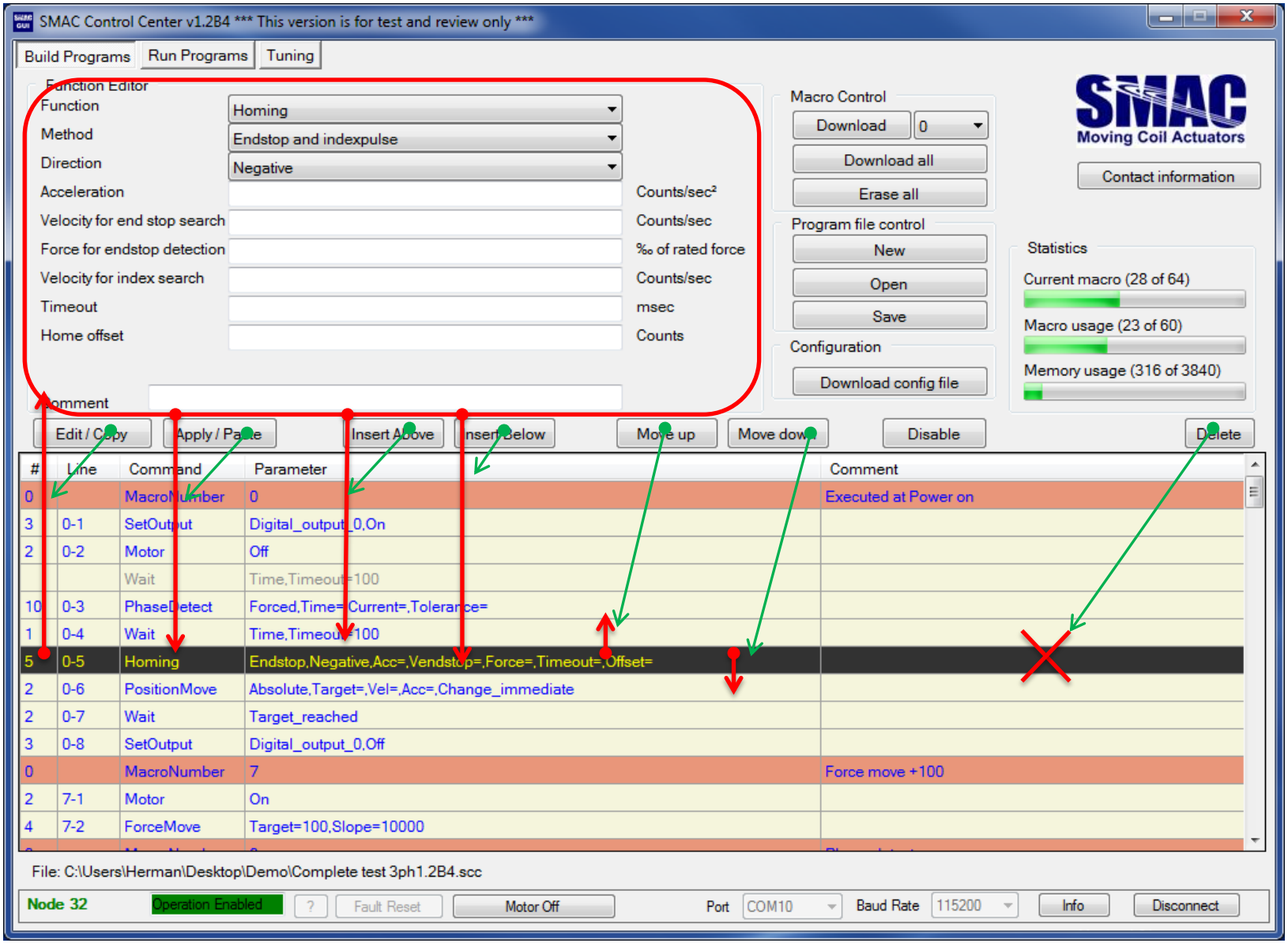
Edit / Copy | Apply / Paste | Insert Above | Insert Below | Move up | Move down | Disable | Delete

#	Line	Command	Parameter	Comment
0		MacroNumber	0	Executed at Power on
3	0-1	SetOutput	Digital_output_0,On	
2	0-2	Motor	Off	
		Wait	Time,Timeout=100	
10	0-3	PhaseDetect	Forced,Time=,Current=,Tolerance=	
1	0-4	Wait	Time,Timeout=100	
5	0-5	Homing	Endstop,Negative,Acc=,Vendstop=,Force=,Timeout=,Offset=	
2	0-6	PositionMove	Absolute,Target=,Vel=,Acc=,Change_immediate	
2	0-7	Wait	Target_reached	
3	0-8	SetOutput	Digital_output_0,Off	
0		MacroNumber	7	Force move +100
2	7-1	Motor	On	
4	7-2	ForceMove	Target=100,Slope=10000	

File: C:\Users\Herman\Desktop\Demo\Complete test 3ph1.2B4.scc

- Current line is marked with black background
- MacroNumber lines are marked orange
- Disabled lines: Grey on yellow background
- Enabled lines: Blue on yellow background
- First column: number of instructions (cost of this programline)
- Second column: Macronumber-Linenumber

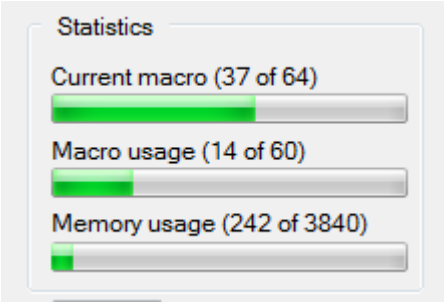
Edit buttons



Edit buttons - continued

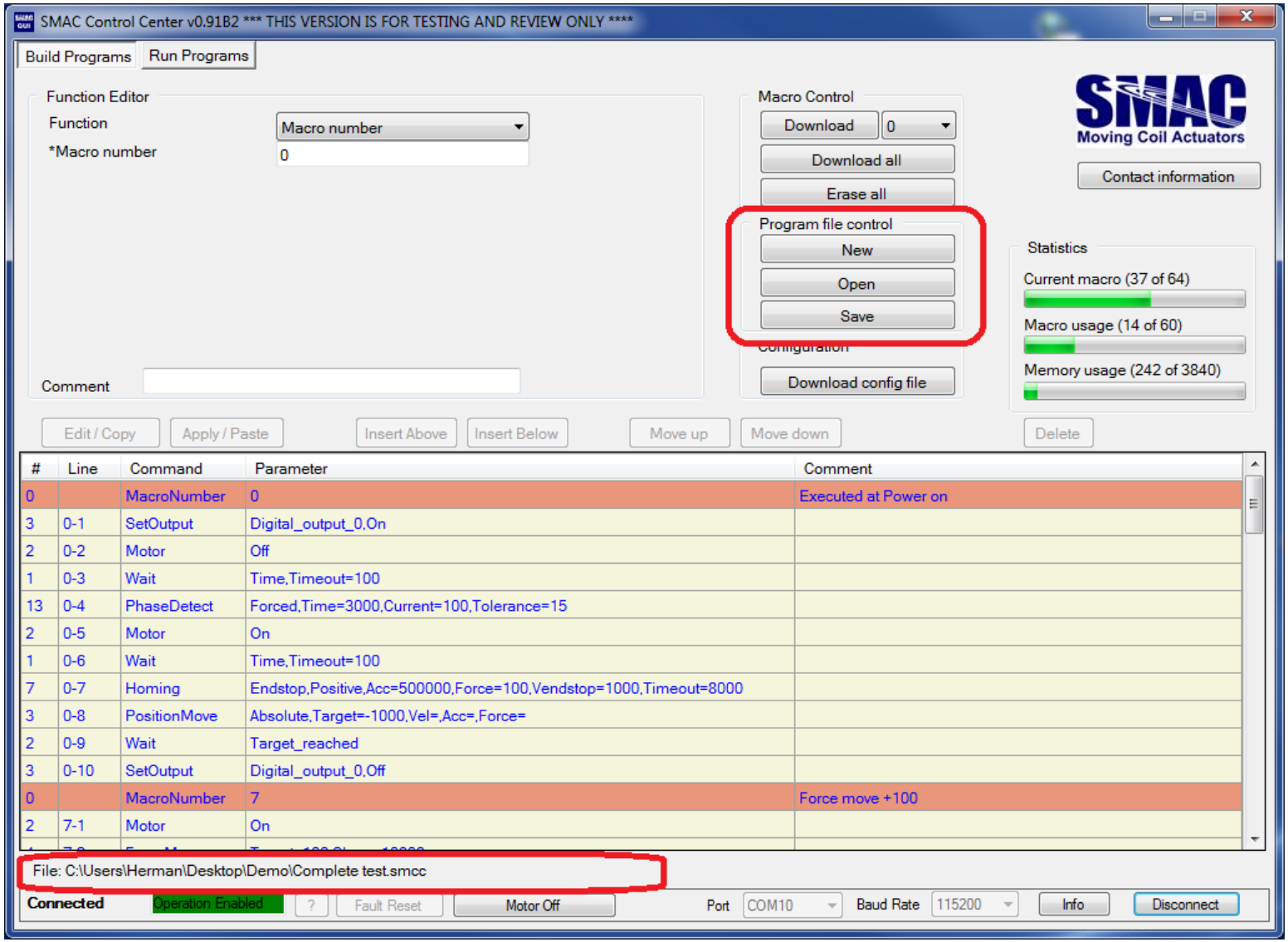
- Double click a row in the listing will copy this row to the Function Editor.
- Hold [CTRL] key and click on a selected line to deselect that line
- To delete a line from the program you also can hit the [Delete] key on the keyboard.
- Don’t forget to hit apply after changing a function
- There is no undo functionality so be careful when deleting lines.
- The disable button will keep the line in the program but will not download the line in the controller.

Statistics

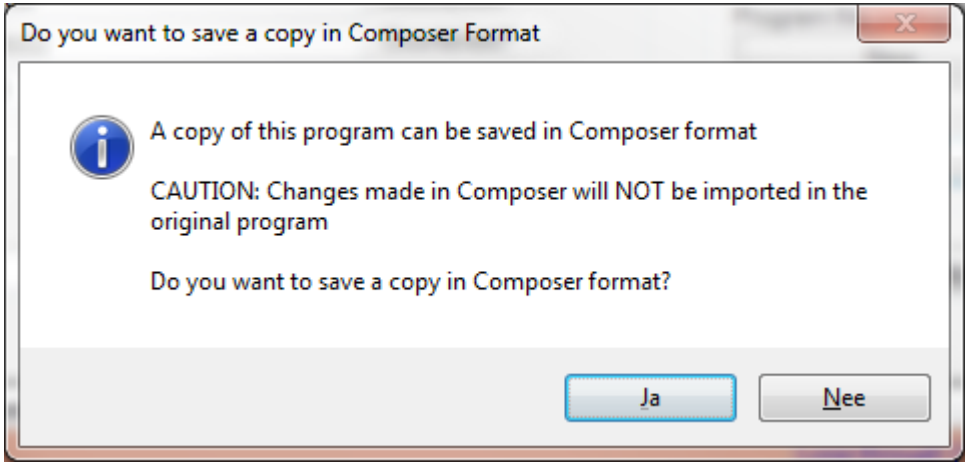


- Number of instructions in current macro (64 max)
- Total number of macros used (60 max, top 4 macros are used for system macros)
- Total memory usage

File New/Open/Save



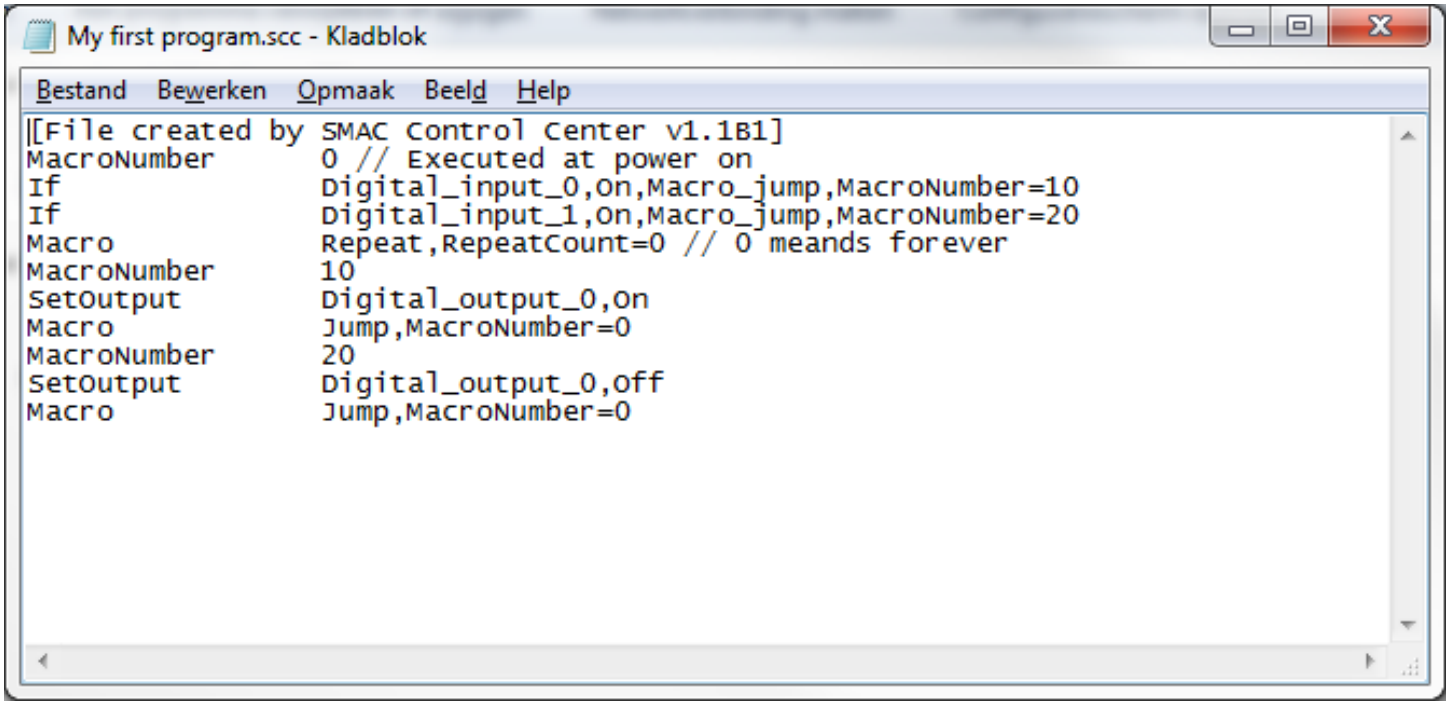
File Save



The Ability to do Work & Verify it's Accuracy at the Same Time

File formats

- Program file suffix is .scc (Smac Control Center)
- Plain text file



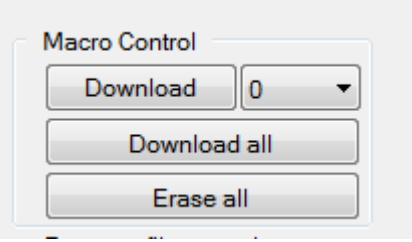
File formats - continued

- Option to save file in Composer format (.mlm)
- Well commented
- Original program line in comment field in []
- Original comment in comment field in {}

My first program.mlm

	#	Instruction	Value	Value (hex)	Comment
					File created by SMAC Control Center v1.1B1
					Manual edits will NOT be imported in the original SMAC Control Center file.
		Macro number	**0**	**0x00**	[MacroNumber 0]{Executed at power on}
	0	Execute if input is On	4	0x04	[If Digital_input_0,On,Macro_jump,MacroNumber=10]
	1	Macro jump	10	0x0A	
	2	Execute if input is On	5	0x05	[If Digital_input_1,On,Macro_jump,MacroNumber=20]
	3	Macro jump	20	0x14	
	4	Repeat	0	0x00	Sequence Macro Repeat,RepeatCount=0 0 meands forever
		Macro number	**10**	**0x0A**	[MacroNumber 10]
	0	Write register16 to accumulator	141826	0x22A02	[SetOutput Digital_output_0,On]Get value of digital output in ACCUM
	1	Or constant to accumulator	1	0x01	Set output bit to 1
	2	Write accumulator to register	141826	0x22A02	Write ACCUM back to digital outputs
	3	Macro jump	0	0x00	Sequence Macro Jump,MacroNumber=0
		Macro number	**20**	**0x14**	[MacroNumber 20]
	0	Write register16 to accumulator	141826	0x22A02	[SetOutput Digital_output_0,Off]Get value of digital output in ACCUM
	1	And constant to accumulator	65534	0xFFFE	Set output bit to 0
	2	Write accumulator to register	141826	0x22A02	Write ACCUM back to digital outputs
	3	Macro jump	0	0x00	Sequence Macro Jump,MacroNumber=0

Macro Control



- Download a selected or all macros in the controller (macros are always saved in NVM)
- A download will also download the system macros (60 .. 63) when necessary.
- Erase all macros in the controller.

System macros

SMAC
Moving Coil Actuators

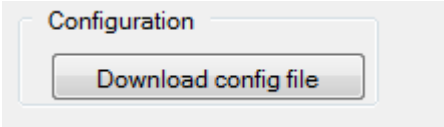
- Macro #60 .. #63 are reserved for system macros. (62 and 63 are used in version 1.0)
- A system macro call takes two steps:
  1. Load ACCUM with the function number
  2. Call macro 63

Defined system macros

SMAC
Moving Coil Actuators

0 - Absolute position move
1 - Absolute position move and change set immediately
2 - Relative position move
3 - Relative position move and change set immediately
4 - Wait for target reached
5 - Motor on
6 - Motor off
7 - Homing
8 - Profile velocity move
9 - Profile force move
10- Homing on current position

Download configuration file



- Downloads a configuration file (.mlc) in the controller.
- Configuration can be created in MotionLab. Standard configuration files can be downloaded from www.smac-mca.nl

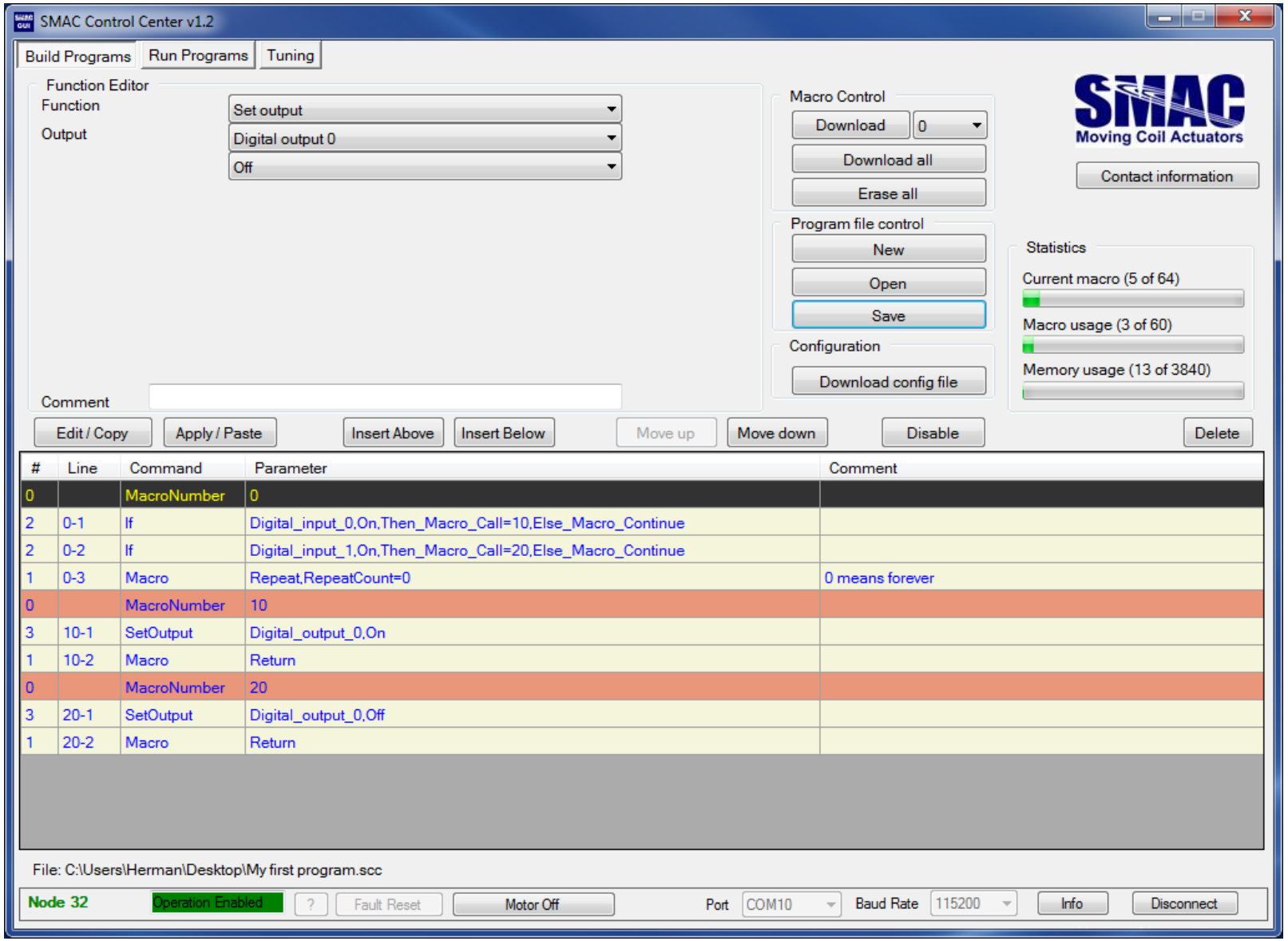
Contact information

- Hyperlinks to help sources , mail and internet

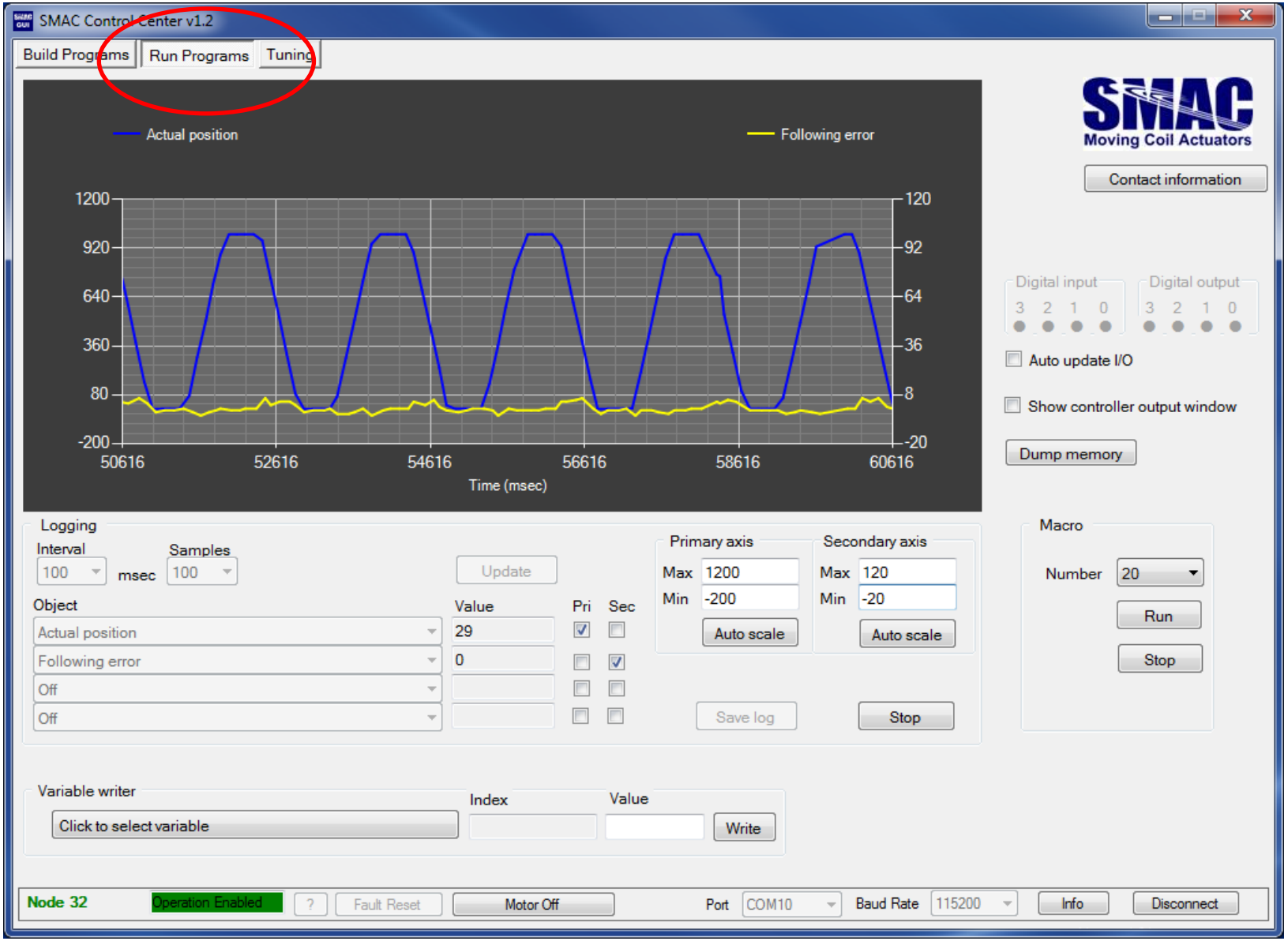
Exercise 1: Create first program

- In an endless loop: Switch digital output 0 on if digital input 0 is on and switch digital output 0 off if digital input 1 is on
- Macro 0 is the endless loop that checks the digital inputs
- Use macro 10 to switch digital output 0 on
- Use macro 20 to switch digital output 0 off
- Save the file as “My first program.scc”
- Download the created macros in the controller
- Switch power of the controller off and on again, this will start macro 0
- Check if the program works as expected with the LCC demo board

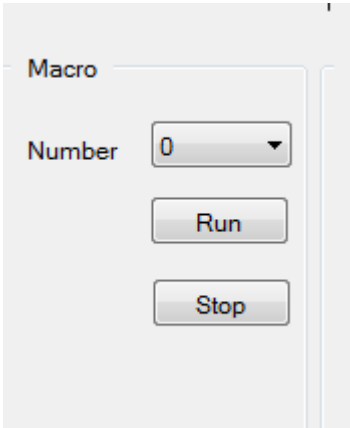
First program



Running programs



Run a macro

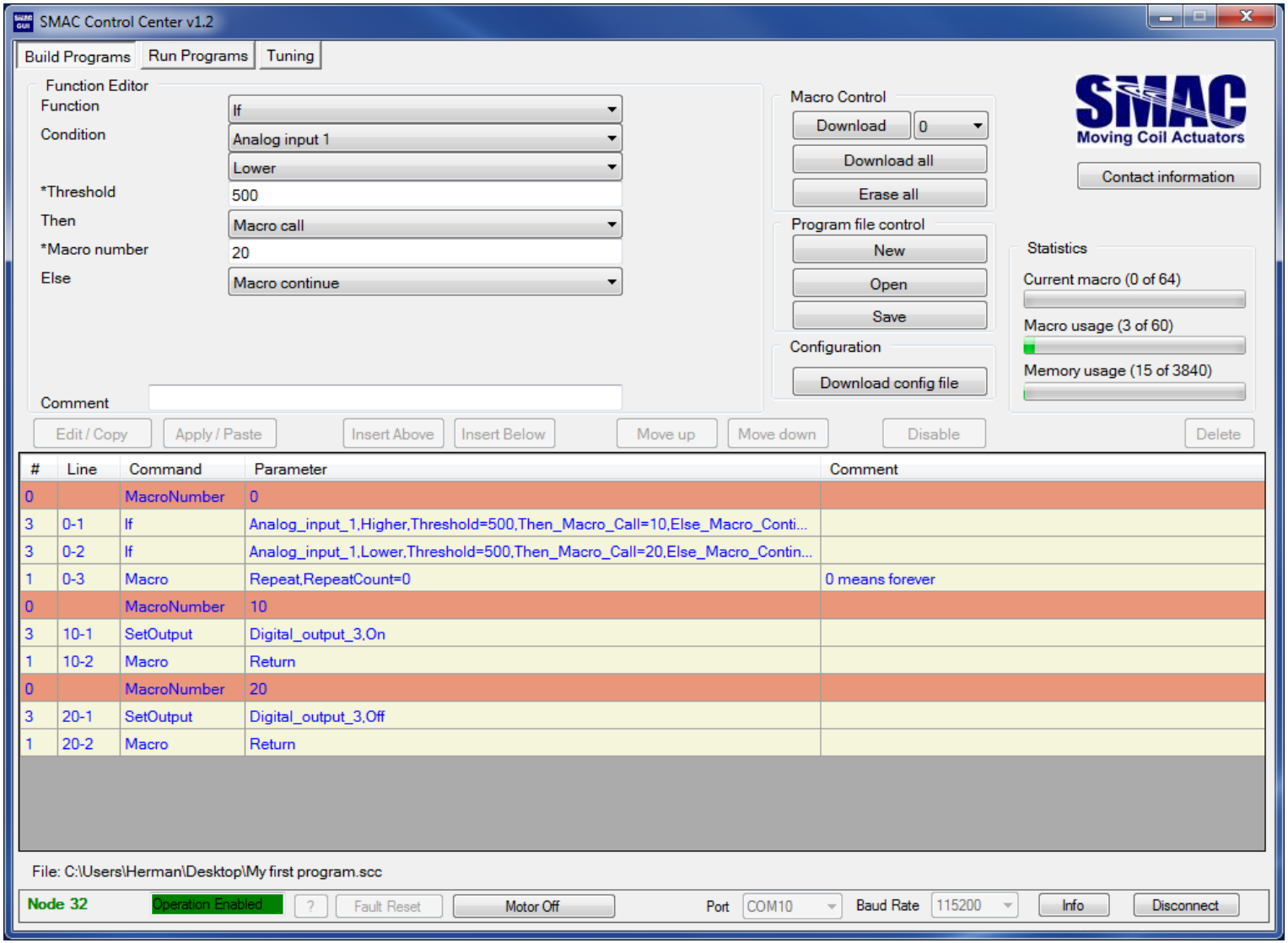


- Select macro number
- Hit Run
- Stop will stop macro execution

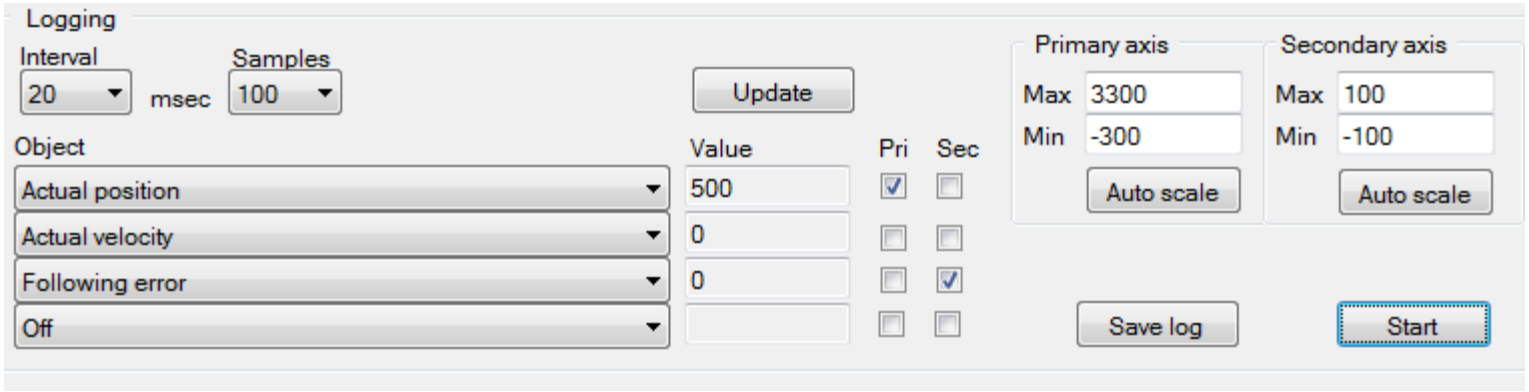
Exercise 2

- Modify the program to set output 3 if the value of analog input 1 is above 500 and to reset output 3 if the value of analog input 1 is below 500.
- Save the new program as “My second program”
- Download the program in the controller
- Check operation with the potentiometer that is on the LCC demo board

Second program



Logging

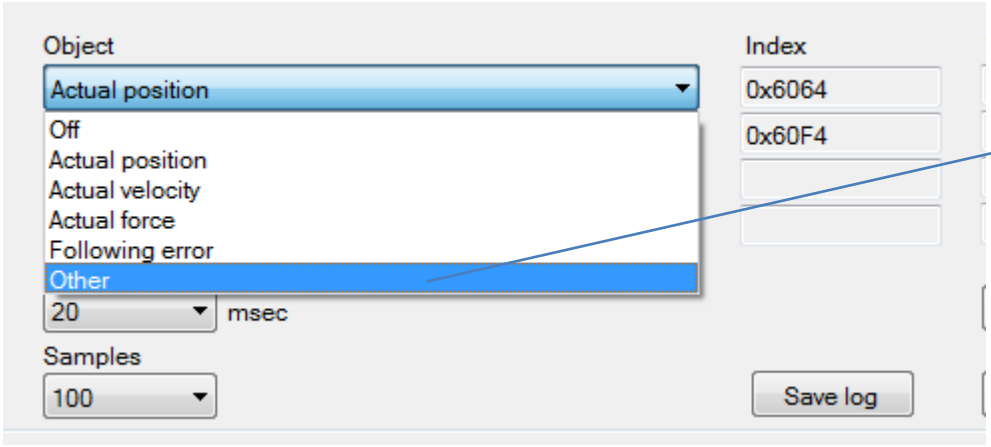


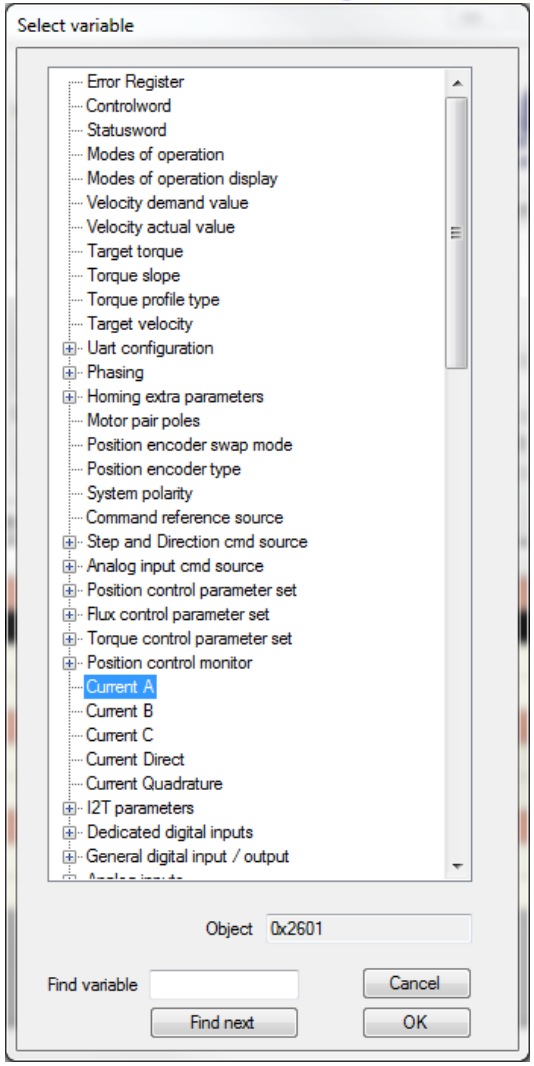
- 4 objects can be selected for logging.
- All objects can be shown in the live chart, Axis can be selected and axis can be scaled manually or automatic.
- Interval can be 20, 50, 100, 200, 500 or 1000 msec
- Number of samples in the chart can be 50, 100 or 250
- Update will only refresh the values once
- Start will start live logging

Logging - continued

- Logging always saves log output in a file named LiveLogging.csv in the users “My documents” folder
- File is readable in Excel, the separator is a semicolon (;)
- File is overwritten at the start of each log session.
- The “Save log” button will copy LiveLogging.csv to a new file with a user definable name in a folder of your choice
- Warning: Logging timing not very accurate (depends on windows timing)

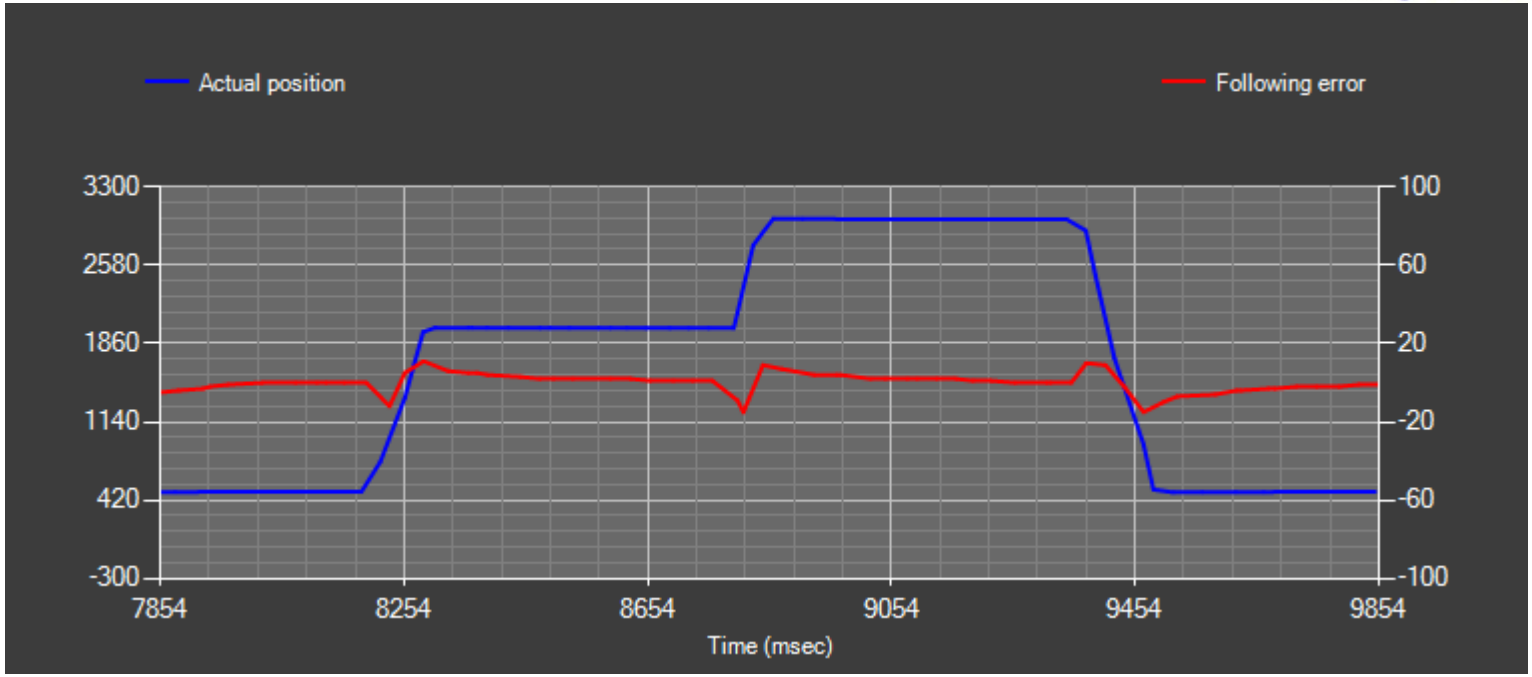
Logging – object selection





- All readable objects can be selected by “Other” or the last item in dropdown list

Chart area



- X- axis is msec since start of logging.
- 2 Y axes: Primary (left) and Secondary (right)
- Object names are shown above the primary or secondary axis

Autosave settings

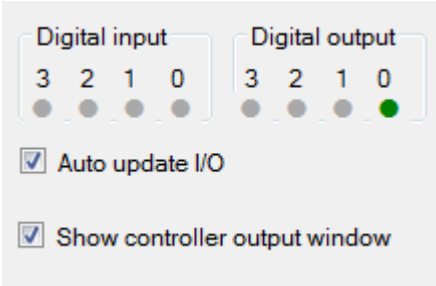
Between sessions the following settings are saved:

- Serial Port number (auto connect at startup)
- Filename (auto open at startup)
- Chart axes
- Log Interval and samples
- Logging objects

Exercise 3

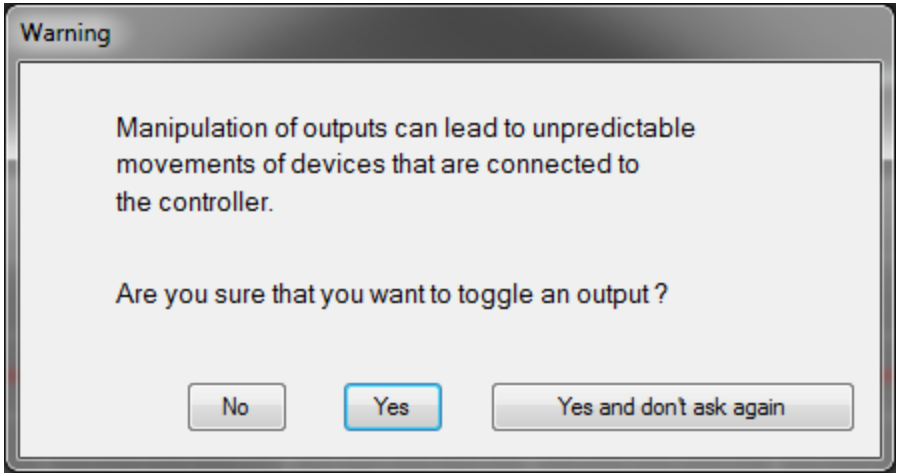
- Show the value of analog input 1 in the live chart

Digital IO monitor



- If Auto update I/O is checked then inputs and outputs are polled every 100 msec
- Outputs are clickable to toggle

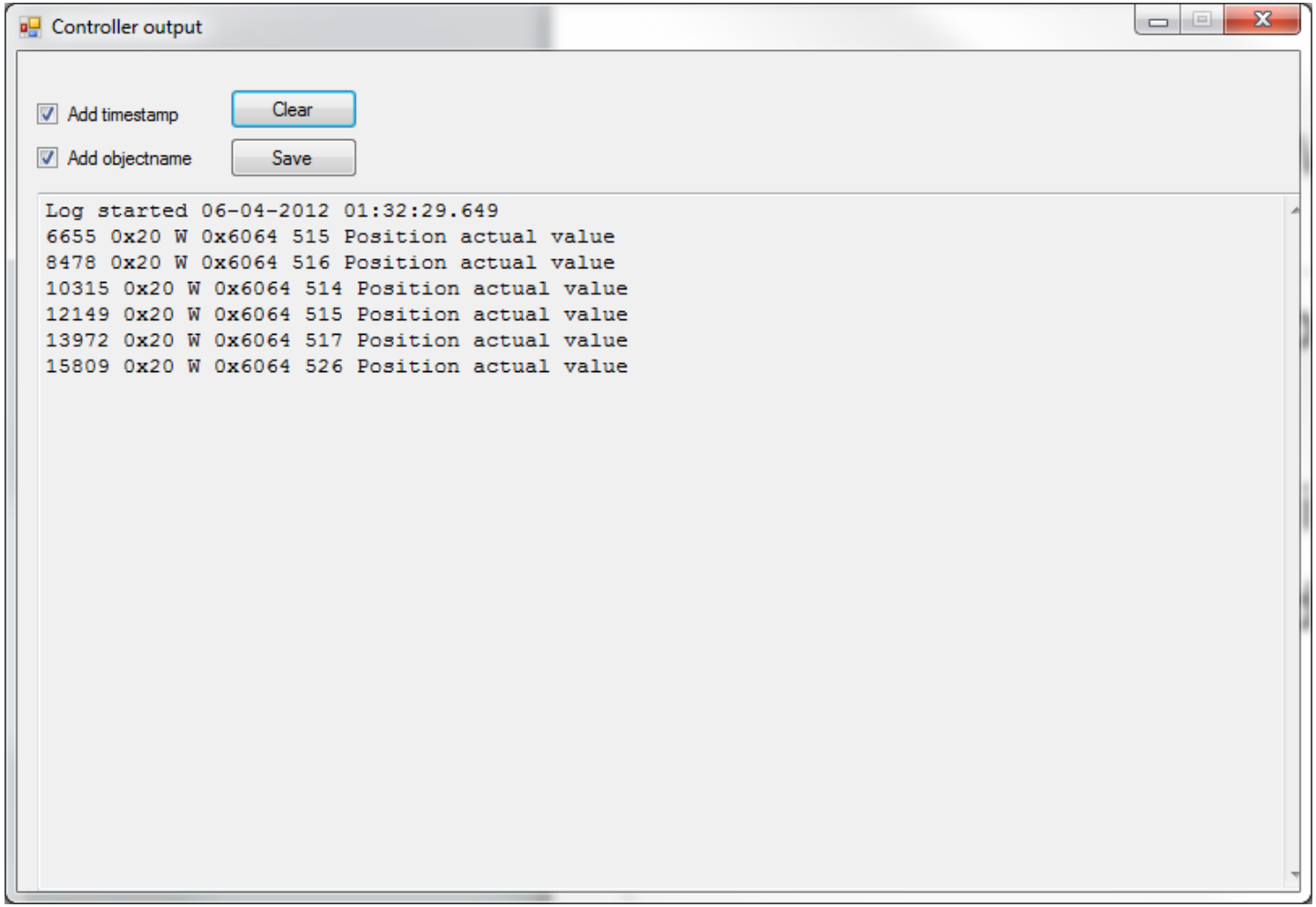
Digital IO monitor (continued)



- Messagebox after click output
- The “Yes and don’t ask again” option is valid for this session only

Controller output window

SMAC
Moving Coil Actuators

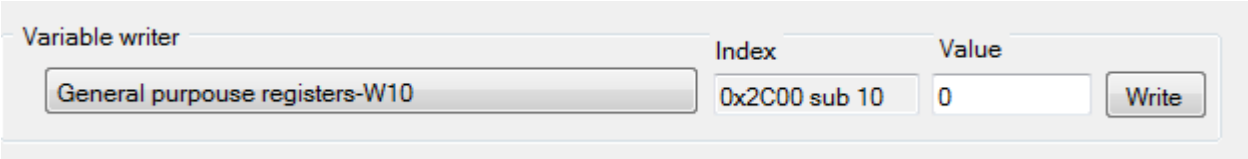


The Ability to do Work & Verify it's Accuracy at the Same Time

Controller output window (continued)

- Shows the response of the controller of get commands
- Output can be saved in a file
- Time stamp can be added (msec since start of SMAC Control Center)
- Only window width can be dragged
- Controller output window is always active from the start of SMAC Control Center. The “Show controller output window” checkbox only switches visibility.
- Warning: Variables that are in the live logging will not show up here

Variable writer



- Can be used to change any writeable variable in the controller immediate.
- Select the variable with the smart object selector and then type a value in the Value box.
- Hit Enter key or click Write button to write the value in the controller.

Exercise 4

- Add a wait 100 msec in macro 0
- Increment General purpose register W10 at each loop iteration
- Add a “Get variable” function for W10 in Macro 0
- Run the program and observe the result in the controller output window
- Warning: Too many get commands can cause overflow of the serial buffers (always use a wait)

Exercise 5: Actuator initialization

- Find the correct configuration file for your actuator at www.smac-mca.nl and download that configuration file in the controller.
- Create a macro 0 that does the following:
  - Phase detect (only for multipole)
  - Homing
  - A position move
  - A jump to macro 10 (macro 10 still empty)
- Test this macro

Exercise 6: Wait for input and cycle motion

- In an endless loop in macro 10:
  - Wait for input 0 to become active
  - Do a motion cycle (out and in)

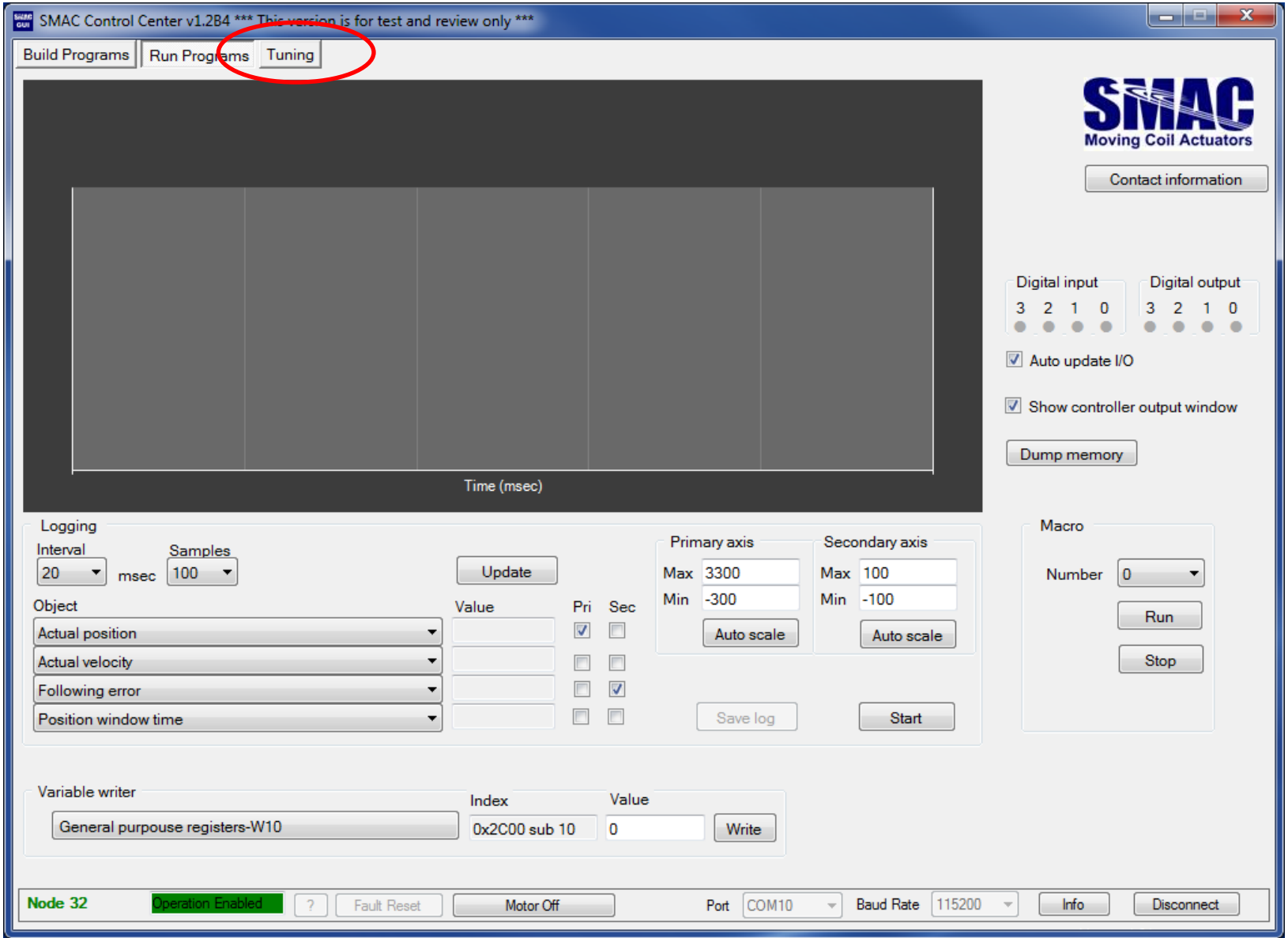
Exercise 7: Test softland

- Write a macro to do a softland and test the various softland modes.
- Use the live logging chart to show the variables that you are interested in (following error, force, position, ….)

Exercise 8: Analog input controls position

- Create a macro that reads the analog input 1 and uses that value to set the target position.

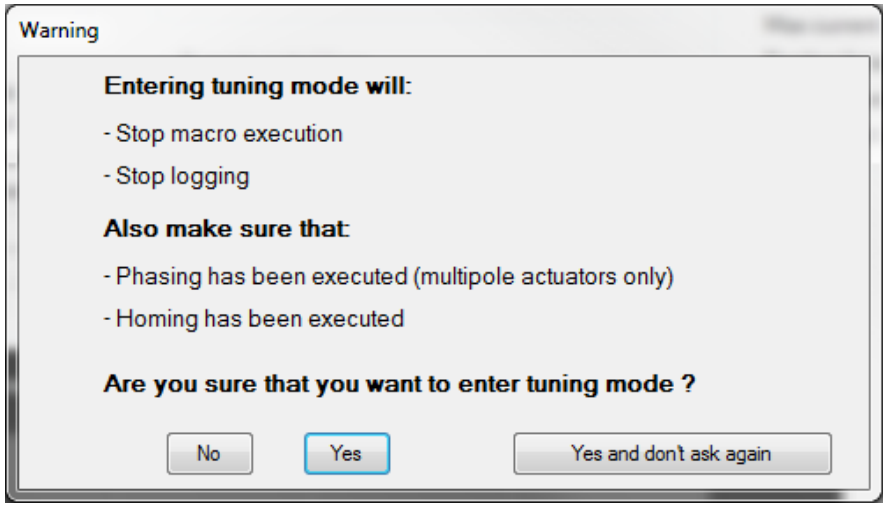
Tuning


SMAC Control Center v1.2B4 *** This version is for test and review only ***
Build Programs
Run Programs
Tuning
Time (msec)
Contact information
Digital input
3 2 1 0
Digital output
3 2 1 0
Auto update I/O
Show controller output window
Dump memory
Logging
Interval
20
msec
Samples
100
Update
Object
Value
Pri
Sec
Actual position
Actual velocity
Following error
Position window time
Primary axis
Max 3300
Min -300
Auto scale
Secondary axis
Max 100
Min -100
Auto scale
Save log
Start
Macro
Number 0
Run
Stop
Variable writer
General purpouse registers-W10
Index
0x2C00 sub 10
Value
0
Write
Node 32
Operation Enabled
?
Fault Reset
Motor Off
Port
COM10
Baud Rate
115200
Info
Disconnect

Tuning functions

- View actuator response plots.
- Adjust the position control loop parameters to match the actual application (Compensate for mass, friction, spring, mounting orientation…)
- View and change limit settings
- Create (partial) configuration files.
- Advise: always start with a configuration file from www.smac.nl

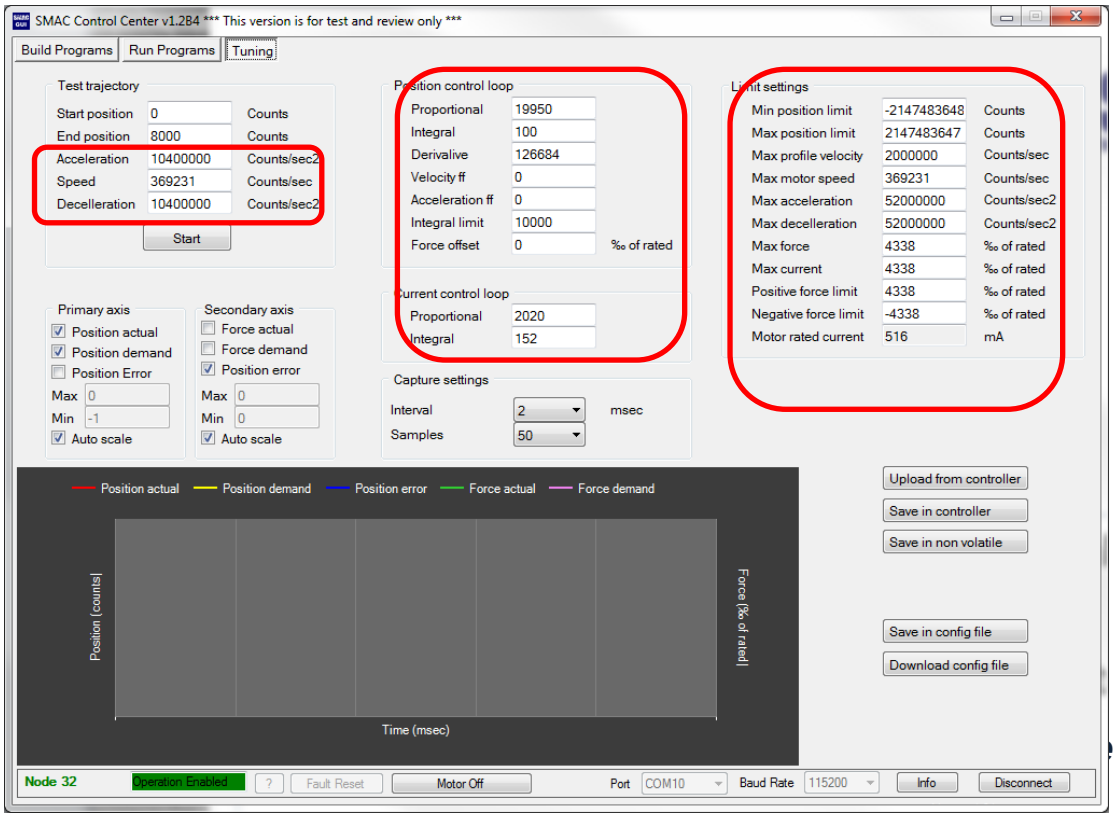
Tuning startup



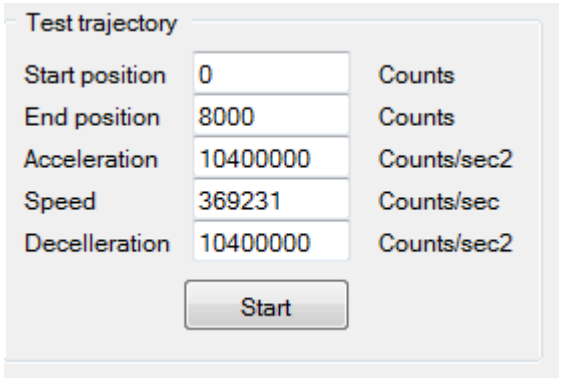
- The “Yes and don’t ask again” option is valid for this session only
- The actual settings are read from the controller and shown in the tuning tab.

Auto upload from controller

- Actual parameters are read from the controller on entering tuning tab (or click Upload from controller)

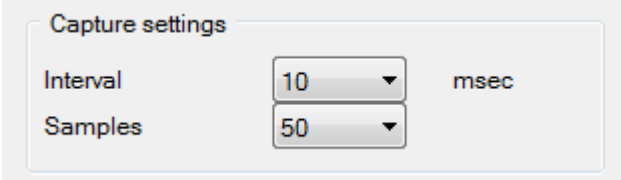


Tuning: Test trajectory



- Define Start position, End position, Acceleration, Speed and Deceleration for the test trajectory.
- Click Start. The actuator will first move to the start position at 50% of Acceleration, Speed and Deceleration, wait 0.5 second and then run the test trajectory at 100%.
- During the test trajectory the controller captures Position actual, Position demand, Force actual and force demand.

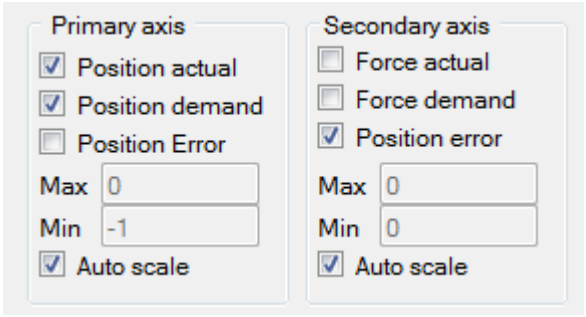
Capture settings

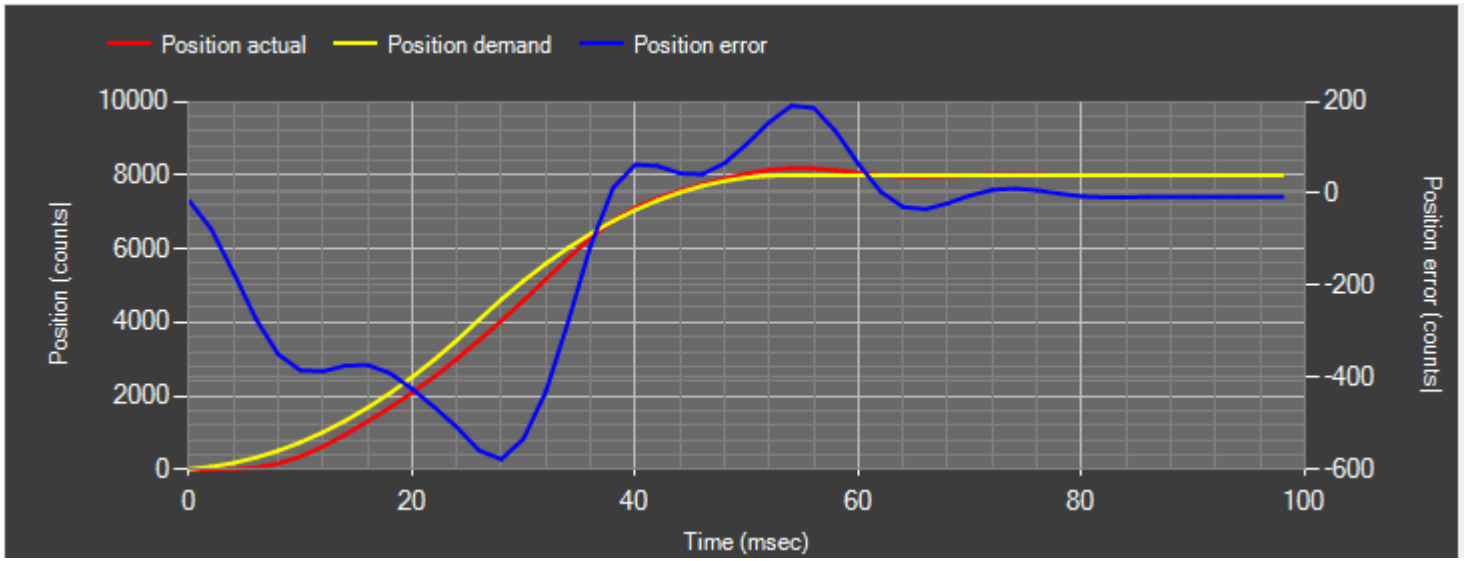


- Select interval and number of samples to match your specific application.
- Interval can be 1, 2, 5, 10, 20, 50, 100, 200 or 500 msec.
- Number of samples can be 50, 100 or 250

Tuning plot

Moving Coil Actuators



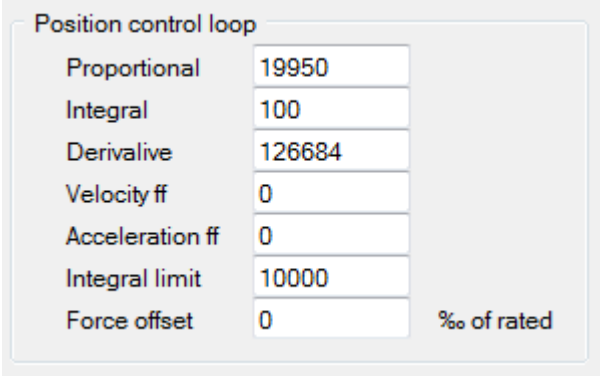


The Ability to do Work & Verify it's Accuracy at the Same Time

Tuning plot (continued)

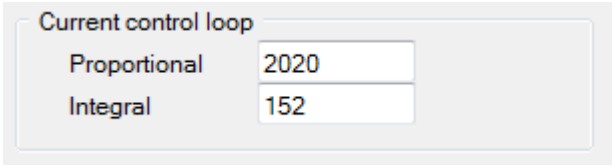
- Shows the result of the test traject logging.
- Plot on primary and secondary axes selectable
- Primary axis can show positions (actual, demand or error)
- Secondary axis can show force (actual and demand) or following error
- Both axes can be scaled automatic or manual.

Position control loop tuning



- Change any value to tune actuator response.
- All values are written in the controller at a start of the test trajectory.
- Use the plot to examine the effect of the change.

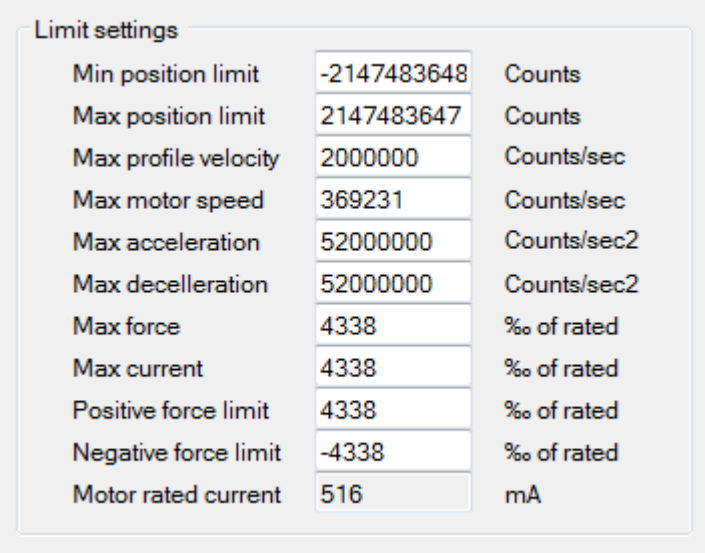
Current control loop tuning



- Current control loop parameters can be changed here but normally this is not necessary.
- The default configuration files have proper values for the current control loop already and these values are not sensitive to application conditions.
- Values are written in the controller at a start of the test trajectory.

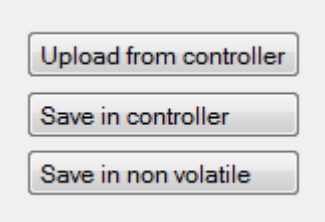
Limit settings

Moving Coil Actuators



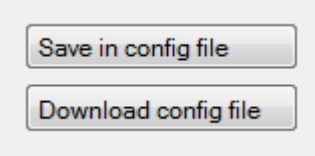
- All objects that can limit motion are listed here and can be changed also.
- All values are written in the controller at the start of a test trajectory.
- Motor rated current is listed here to make it possible for the user to calculate the currents in mA from the ‰ value.

Controller buttons

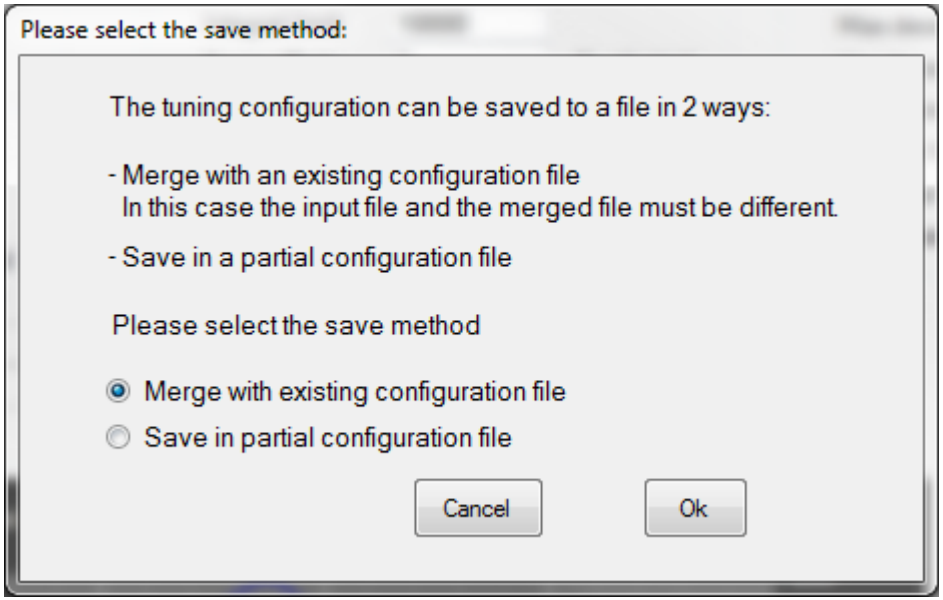


- Upload from controller:
  Read all values on the tuning tab from the controller.
- Save in controller:
  Write all values on the tuning tab to the controller (in volatile memory)
- Save in non volatile:
  Write all values on the tuning tab to the controller (in non volatile memory)

File buttons



- Save in config file:



File buttons (continued)

- Download config file:
  Writes an existing configuration file in the controllers non volatile memory. (same as on the Build programs tab)

Exercise 9: Position loop tuning

- Define a test trajectory for your actuator and look at the response using the default configuration file
- Then change proportional, integral and derivative constants (small changes and only one change at a time) and examine the effect of the change in the plot.

Future developments

SMAC
Moving Coil Actuators

- Expand functions
- Add new tab: PID tuning : Done !!
- More objects in live chart : Done !!
- CAN support
- And improvements that are requested by our users: We welcome suggestions very much to make this tool better !!

Questions ?

SMAC
Moving Coil Actuators

The Ability to do Work & Verify it's Accuracy at the Same Time