

CANOpen Graphical Interface (GUI) User Guide 2.1.12

1.0 Prerequisites for running the GUI

- mPLC application has to be installed
- Download location: <https://www.haydonkerkpittman.com/products/motors/brushless-dc-motors-with-idea-drive>
- install mPLC (v2.91), when finished restart your PC
- plug in the miCAN-stick into an USB port

This GUI provides easy access to most (but not all) of the functionality in our CANOpen IDEA drives. It is a demonstration/test tool, and not a commercial product.

It runs Python in the background, extended by wxPython graphical class. This combination provides a completely open and freeware approach, and it still remains somewhat compatible with plain, non-graphical Python scripts. Source code of this GUI can be used as a resource/example for writing Python scripts.

1.1 Procedure before running the GUI

- miCAN-stick has to be plugged into USB port
- drive and miCAN-stick connected via cable provided

1.2 Starting the GUI

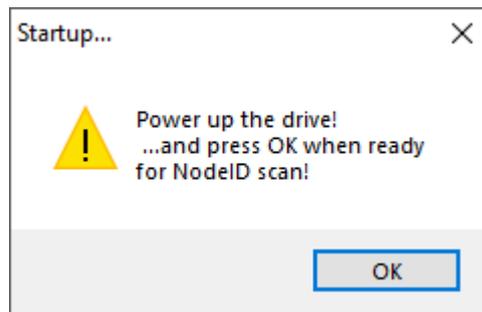
The GUI is run double clicking the „IDEA Drive GUI.pyx“ file from the „HKP_CANOpen_Tool_v2.1.12“ folder.



IDEA Drive
GUI.pyx

1.3 Detecting NodeID by the GUI

On the startup the GUI will pop up a window remainder to power up the drives. If powered, click „OK“.

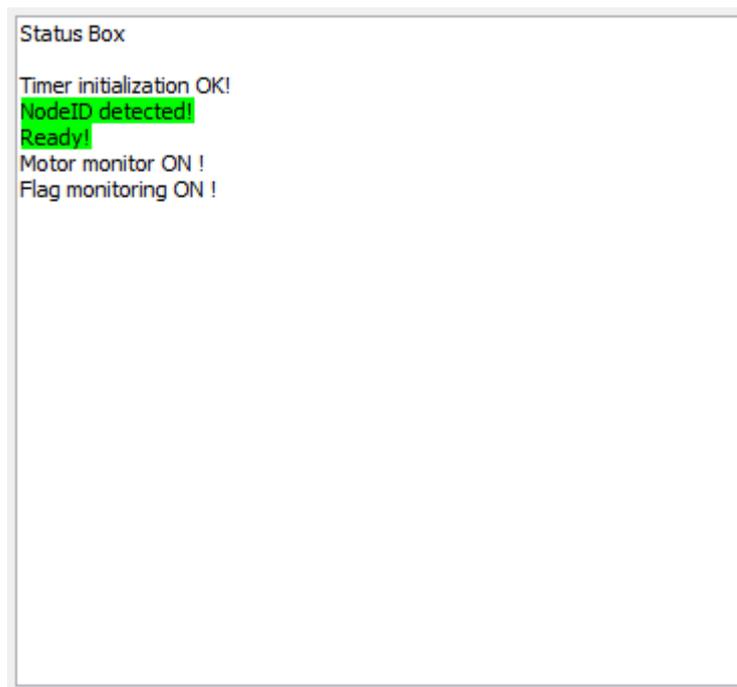


After proceeding the GUI will search for the NodeID of the connected drive. The GUI first looks for the default NodeID , if not found, it will begin to search for NodeID from 1 to 127. Baudrate is also scanned at startup in a predefined order (125>250>500>1M).

If multiple drives are connected to the network, the first NodeID detected gets the priority. It is safe to use the GUI with multiple drives, but users have to be aware of the detecting sequence.

If no drive detected, the GUI will offer to do the scanning again.

If NodeID detected, you get a **feedback message** in the Status Box, see on the picture bellow.



The current NodeID and Baud rate can be seen on the top of the window:

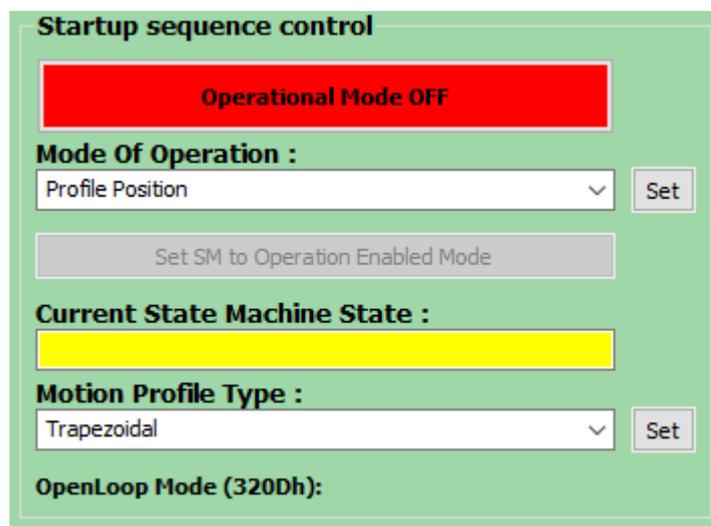
 HKP CANOpen Tool For IDEA Drive v2.0.10 (NodeID = 127 @500kbps), Firmware version : 5.70.00.17

2.0 Using the GUI – First steps

2.1 Internal State Machine (SM) and mode of operation set

According to DS402 CANOpen standard, and some internal (manufacturer specific) requirements, the following sequence of steps are required to prepare the motor for operation (executing moves). These steps are to be found under the group „Startup sequence control“ in the GUI.

1. Switch the CANOpen communication state-machine into „Operational“ mode by clicking the „Operation Mode“ button
2. Choose mode of operation, and click the „Set“ button to set the desired mode of operation
3. Click the „Set SM to Operation Enabled Mode“ button.
4. You can check the current SM state under „Current State Machine State“
5. Optionally, you can change the motion profile type, default is „Trapezoidal“
6. Optionally, check the „Monitoring ON“ checkbox to monitor position and velocity (note: this is located on the right side of the GUI, not under „Startup sequence control“)



After every step, it is recommended to check drive's response in the Message Window. These steps must complete without errors in order to achieve the correct state.

2.2 Setting the default object parameters

Multiple objects (both manufacturer-specific and DS402 specific) has to be set correctly before the user can execute a move. These objects are listed in the grey area at the left side of the GUI.

By clicking the „Load Defaults“ button, the GUI will fill up the object values with suggested values.

To use these values, click the „Set All“ button. This will write all the objects into the drive. Users can change any of these values by entering a different value before sending.

Default values can cause the drive to respond with error message during the process, because the GUI is not aware of the drive's type, so some of the values may fall outside of the drive's capabilities.

This is not considered as an error, because the drive has its internal limiters to prevent invalid setup. For example, if the GUI tries to transfer a „Run Current“ value higher than that a drive can handle, the drive will limit the value to the maximum possible run current. This will cause an error message in the Message Window, because the sent value does not match the readback value. The object however will still be rewritten by the internally limited maximum possible value. This behaviour is true for most of the objects, so the „value mismatch“ type messages are most of the time not considered as a real error.

Basic parameters which are specific for the drive are already pre-set in the firmware (like encoder CPR, electrical and winding parameters...etc).

The grey area is divided to upper and lower part, to split object to DS402-related (upper part), and manufacturer -specific (lower part) objects.

The „Read All“ button will read back and display all the listed objects directly from the drive.

2.2.1 Steppers and default parameters

The default object values are ideal to drive BLDC motors, however, if stepper motors are used, some parameters MUST be changed. The steppers commutation can not handle high velocity and accelerations. Acceleration and deceleration is recommended to be at least 1000 times lower, under 20 and velocity under 200 RPM.

2.3 Monitoring Parameters

If the „Monitoring ON“ is switched on, it will constantly poll the drive for basic parameters and display it in realtime. The refresh rate of this window is about 300msec, so during quick movements, the displayed position and/or velocity may not be correct all the time.

Notice : Motor Monitor can be used to verify correct setup by turning the motor's shaft manually and observe the resulting change in position/velocity. Velocity is displayed in RPM. Position is displayed in encoder-counts.

Turning the shaft clockwise/anticlockwise should increase/decrease the displayed position. This is a very good indication of a correct setup, and encoder functionality.

3.0 Inputs/Outputs – the I/O ports section

I/O port section can control and display input and output port states. I/O monitoring can be switched on/off by checking/unchecking the „Monitoring ON“ checkbox under the group „I/O ports“.

I/O monitor is designed to match the original I/O box functionality, so a switched on output means a switched-on LED that corresponds to that output. In case of inputs, the switched on LED means the input is ON.

Other I/O designs by users may have an inverse logic, so the GUI may not display the I/O states correctly in those applications.

IDEA drive has 4 physical inputs and 4 physical outputs. However, DS402 standard enables further ports by defining virtual inputs/outputs that can be tied (mapped) to physical ports. This is very useful in creating universal applications, where the physical change in port assignment can be handled in setup, eliminating changes in code.

Also, virtual I/O is required for Homing functionality, by defining limit and home switches.

Acronyms in I/O port section:

Inputs:

NL – negative limit switch

PL – positive limit switch

HO – home switch

CLE – error clear switch (not used in IDEA drive)

ST – stop switch (not used in IDEA drive)

Outputs:**BRAKE – output for external brake (not used in IDEA drive)****ERROR – error-indicating output (not used in IDEA drive)**

The virtual outputs which are not implemented in IDEA drive are still present because these are defined in DS402, and they can be implemented in the future if needed.

I/O functionality is not tied to a specific mode of operation as other motion-related portions of the GUI. I/O ports should function regardless of the chosen mode, and even without selecting a mode.

Inputs can't be enabled or disabled. The yellow input indicators will change to ON/OFF (green/red) according to real inputs when the I/O monitor is on. If an input is mapped to a virtual input, the virtual input's indicator has to change too.

Outputs can be enabled/disabled by checking/unchecking the boxes above each output. In order to control an output, it has to be enabled. Mapped outputs will change in sync with a real output, but if there is a conflict between real/mapped state, the mapped state will always have the higher priority.

Output state can be set by the user by clicking on the corresponding indicator (colored square). The I/O box (if installed) has to display the state immediately on the corresponding LED indicator.

The I/O monitor's refresh rate is limited, so it can not display I/O changes happening too fast. The drive still can detect those changes, as it has a much faster polling rate.

3.1 Mapping inputs/outputs

State of mapping can be read/set. It is possible to map multiple inputs/outputs to the same virtual I/O in GUI, but the firmware will only allow mapping the first occurrence, so it is not recommended.

Each mapping has a „high active“ bit setting. This bit can be used to set the „default open“ / „default closed“ state of the mapped switch. This setting has to match the physical implementation of the used switch. The switch can be closed by the default, but connected I/O electronics can change this state before it can be read by the firmware. If the I/O electronics details are unknown, only tests can conclude which is the correct setting.

4.0 Velocity control mode

Velocity control can be switched on by choosing „Profile Velocity“ mode in the „Mode of operation“ box, and clicking the „Set“ button.

Upon setting the operation mode to „Profile Velocity“, the field in the middle of the GUI should jump to tab „Velocity“, or the user can click on it manually, to access all the controls and parameters used during „Profile Velocity“ mode.

Target velocity can be set by entering a value manually into the „RPM“ box, or by sliding the yellow slider above. When using the slider, the velocity value is immediately sent to the motor during slide, so very fast control is possible.

Target velocity is limited by the „Max Motor Speed“ (6080h) object, and by the „Max Profile Velocity“ (607Fh) object. Max Profile Velocity object is limited by the Max Motor Speed object.

Max Profile Velocity will limit the velocity-slider.

To change Max Profile Velocity, enter the value in the object's edit box, and click the „Set All“ button.

Objects in the grey area can not be changed individually. By clicking the „Set All“ button, all object values will be sent to the drive.

Setting the velocity will not start the motor, it will just transfer the value to the drive.

To start the motor , click the green „START“ button in special controls section.

The motor should immediately start rotating, and should reach the target velocity and maintain it.

Once the START command is sent, now the velocity can be changed on-the-fly with the slider. The motor will react and should reach the new commanded velocity.

The movement can be stopped with the big „STOP“ button, and can be restarted with the latest velocity settings immediately by clicking the „START“ button.

There are two more settings associated with velocity mode. These are defined by the DS402 standard, and provide some feedback to user in form of flags. The monitoring of these flags are turned on by default in the GUI. It can be turned off by checking the „Monitoring ON“ checkbox in the yellow field.

4.1 Velocity window settings

This setting is associated with the „Target Reached“ flag provided by the system.

The target reached flag will turn green (ON) if the motor's velocity (RPM) is within the defined window, relative to the target RPM, for the time defined.

For example :

Target RPM = 2000RPM

Velocity window = 10RPM

Velocity window time = 300msec

The target-reached flag will turn ON, if motor RPM is higher than 1990RPM but it's lower than 2010RPM for a measurement period of 300msec. Target-reached flag is updated real-time, so it can happen that a too strict limit will cause the target-reached flag to flicker, being in and out of the limit depending on motor load and other factors.

4.2 Velocity threshold settings

This setting is associated with the „Velocity Threshold“ flag (bit12 in statusword).

It will become active (ON), if the measured velocity (RPM) passes the set threshold for longer than the Threshold Time. It can be used in simple systems, when constant velocity monitoring is not needed, but it is still important to know if the motor has reached a velocity threshold.

5.0 Profile Torque Mode – controlling torque

Torque control can be switched on by choosing „Profile Torque“ mode in the „Mode of operation“ box, and clicking the „Set“ button.

Upon setting the operation mode to „Profile Torque“, the field in the middle of the GUI should jump to tab „Torque“ , or the user can click on it manually, to access all the controls and parameters used during „Profile Torque“ mode.

Warning ! When using Profile Torque mode, and the motor's shaft is not loaded, the motor will spin up very fast, and reach max. possible RPM even for minimal torque values as it tries to reach the given torque. This is a normal behaviour, but should be avoided if possible.

Notice : In this mode, the position display in Motor Monitor section will automatically switch to torque-mode, and instead of position, it will display torque measured by the firmware.

Torque mode has a different principle of objects than other modes : The most important objects are defined as „per thousand of rated torque“. This will help application engineers writing a more universal code that can work across different motors.

For eg : Max Torque object (6072h) Indicates the maximum permissible torque in the motor. All other objects take this as a limit. But, this object is not defined in some unit, but it is defined as „per thousand of rated torque“.

If the Max Torque defines 1200 to be the maximum torque, it means that the motor is allowed to output 120% of rated torque (for a limited amount of time).

Additionally, maximum torque can be limited by the „Limit Max Torque“ slider.

Torque can be modulated by the „Torque“ slider in CW or CCW directions. Moving the slider has an immediate effect on generated torque!

The „Torque Slope“ determines how quick the motor will ramp up the torque.

It is recommended to first adjust the slope, and then modulate torque.

As in other modes, START and STOP buttons must be used to start/stop the motor in torque mode.

5.1 Target Torque Reached flag

This is the only flag associated with torque mode. This flag will be active, if the internally measured torque is within the range defined by Torque Window.

6.0 Profile Position Mode

Profile Position mode can be switched on by choosing „Profile Position“ mode in the „Mode Of Operation“ box, and clicking the „Set“ button.

Upon setting the operation mode to „Profile Position“, the field in the middle of the GUI should jump to tab „Position“ , or the user can click on it manually, to access all the controls and parameters used during „Profile Position“ mode.

Notice ! Switching mode of operation during when the motor is executing a previous move is not recommended ! The system has a safety feature that will try to stop any move in case of switching modes, but It is preferred to use the STOP button to stop the motor before changing modes.

Position control is the most complex part of the GUI. While in other modes, the drive will execute the move as soon as the new target (velocity, torque..) is written into the object, position mode requires a handshaking protocol to be followed by the drive and the controller.

The new target will become effective only if the handshaking procedure was successful. This procedure is mostly hidden by the GUI, but you can follow it in the Message window.

Things to setup before using the Profile Position mode :

- speed at which the motor runs during the move is defined by the „Profile Velocity“ (6081h) object
- acceleration / deceleration to that speed is defined by the „Profile Acceleration“ and „Profile Deceleration“ objects.
- The allowable max. current during the move is defined by the „Run Current“ object

Usually, the already loaded default values are working well.

Software position limit objects will prevent the user to set target position outside of these limits.

Hold Current Object : Hold Current defines how much current the motor will use to maintain the position at the end of the move.

Other settings:

Absolute/Relative position switch : In absolute mode, the target position will be processed as an absolute target, while in relative mode, the target position will be relative to the previous target position.

Multipoint positioning section: This section is active only when „multipoint positioning“ checkbox is switched on. This mode will be described as a separate topic (7)

6.1 Starting a move in „Profile Position“ mode:

Enter a new target position (in encoder counts) in the edit box, and click the „Set“ button.

NOTICE ! In this mode, it's not required to press the START button. The motor will start automatically when setting a new position. This is a GUI function only. In real application, the CANOpen controller has to control the START bit.

The motor should begin to move to the given target position immediately. Position can be verified in the Motor Monitor section.

If the motor can't reach the commanded position by just 1 or 2 encoder counts, try to increase the „Hold Current“ value.

Before commanding another position, it is recommended to check the Message Window if the „Ready for new setpoint!“ message has appeared. This is the indication that the system has processed the previous target and it is ready for the new one. In real application, this flag can be monitored in program to determine when to send the next target. In the GUI however, sending the new target too early can confuse and potentially crash the GUI.

If reaching the target takes too much time, the STOP button must be used to abort the move !

6.2 Flags associated with Profile Position mode:

6.2.1 Position Reached Flag (Position Window Settings section):

In a similar way to other window-type flags, the Position Reached flag also requires a window and a window-time to be set.

A window is defined for the accepted position range symmetrically around the **target** position. If a drive is situated in

the accepted position range for more than the window-time, the Position Reached flag becomes active.

It is not recommended to set zero for the window-time.

NOTICE ! When using multipoint positioning mode , this flag will become active only when the last processed target position is reached.

6.2.1 Following Error Flag (Following Tolerance Settings):

In a similar way to other window-type flags, the Following Error flag also requires a window and a window-time to be set.

Following Error can be used to check positioning errors during positioning, when target position is not yet reached.

A window is defined for the accepted following error tolerance symmetrically around the reference position. If a drive is situated out of the accepted position range for more than following error time out time, the Following Error flag becomes active.

It is not recommended to set zero for the window-time.

6.3 Multipoint Positioning

If the multipoint positioning checkbox is checked, the drive enters the Multipoint Positioning mode. This mode is a special extension of the single point position mode, and requires a different approach from the user to provide target points.

IDEA drive has a 4-point target-position ring buffer in this mode. Users can send target positions into this buffer, as long as there is an empty slot in the buffer available. As targets are processed, the drive system will free-up the processed slot in the buffer automatically, and it will accept a new target position. If the buffer is full, the user (application) has to wait for a free slot to become available.

This process requires a complex handshaking procedure to be followed. The GUI is handling this handshaking, but the buffer must be filled with target points by the user.

How to use this mode :

In this mode, the „Target Position“ is not processed, instead the user can enter 5 different positions before starting the move, and send those positions to the drive, filling up the point-buffer (Multipoint positioning section).

When clicking the “Set Next” button, the GUI will send the first position to the drive, and the drive will begin to process the point(s) in the buffer. Clicking the “Set Next” button again, will automatically send the next position (position #2)...and so on. If the drive has a free slot in the point-buffer it will store the position there, and it will process it when the previous target is reached (processed). This way, the user can quickly set 4 points and wait for the drive to process those, and then set another set of points...etc. If the buffer is full, the new target cannot be placed in the buffer, so the drive will reject the new position.

The “Set Next” button will go through all 5 points automatically, and after the 5th position is transferred, it will circle back to the first. This is done to be able to demonstrate the ringbuffer-like behavior in this mode. The next point will be always placed in the next free slot in the buffer. If there is a free slot in the buffer, the controller (user) can periodically and continuously send new targets to the drive without having to wait for the previous move to be finished.

NOTICE ! When using multipoint positioning mode , this flag will become active only when the last processed target position is reached.

6.4 Special settings

End of move position correction

- Open loop mode only
- Encoder monitoring required
- At end of move checks if measured position within target position error window (End of move position correction deadband)
- Will automatically initiate move to correct for position if outside threshold window

Stall Detection

- Open loop mode only
- Encoder monitoring required

- Tracks position over entire move profile. If Position Error Threshold exceeded anytime during the move it assumes a stall condition and will initiate a new move to the target position. Move retry attempts not to exceed Stall Recovery Attempts.
- Typically Position Error Threshold > End of move position correction deadband

Position Validation

- Servo Mode (inherent tracking)
- Will not indicate move completed until the measured position is within the Position Error Threshold window of the target position

7.0 Menus

7.1 Utils menu

The „Utils“ menu contains utilities for some special drive operations. Utilities like the EEPROM dump, and EMCY test are only for testing, and will not be covered by this manual.

7.1.1 Change NodeID

In this menu, it is possible to change NodeID and Baud rate of the connected drive.

After setting a new NodeID and/or Baud rate, the new settings will take effect only after resetting or power-cycling the drive. The GUI needs to be restarted also.

If the NodeID is changed from the default (32), mPLC can not detect the bootup message anymore, because it is filtered to detect only the NodeID of 32. The GUI should work fine without this, but the GUI will take longer to find the new NodeID at startup, as it is expecting the default one, and has to search the CAN bus for the new NodeID.

In case the Baud rate is changed, the miCAN stick configuration also has to be changed in mPLC's control center-> Config Hardware-> CAN-USB-> Baudrate, otherwise the drive cannot be detected due to speed settings difference.

Using the “Detect” button, the current NodeID of the connected drive can be read.

7.1.2 Update Firmware

Opens up a new window for the Firmware update process. **Firmware update is only possible in “OPERATION ENABLED” state!** Firmware files must have .bin extension and must be exactly 128.000 bytes long.

On the right side of the window, the available drives can be listed with “Scan network” button. One of the available drives must be selected before flashing the new firmware. After the update process is done, simply select another drive to update next.

The GUI must be restarted after all the desired drives are successfully updated.

7.1.3 Load Defaul Parameters

Loads a default-parameter .bin file to the drive. File must have a .bin extension. When finished do a power-cycle on the drive and restart the GUI. After the GUI restart, choose „Restore Firmware Defaults + Reset” from Save/Restore menu. Do a power cycle and a GUI restart to finalize the parameter update.

7.1.4. Hard Reset

Performs a hard reset of the drive (same as power cycle). The GUI must be restarted after this event, as the state machines will not be in the expected state.

7.1.5 Firmware Version

This menu will display the firmware version.

7.1.6 List motor parameters

This menu will list the motor parameters in the message window on the right side of the GUI. These parameters include motor model, drive number and various physical characteristics.

7.1.7 Communication

Choosing this option, the GUI will re-scan the Baud Rate and Node ID, the same process is run at the GUI startup. Currently implemented for debug purpose only.

7.2 Save/Restore Menu

In this menu, it's possible to save/restore object dictionary values into the internal non-volatile memory. List of saved/restored objects is predefined in the firmware and can not be changed.

Refer to the Knowledgebase on which object has the save/restore capability.

If the object dictionary is saved, the saved values will be loaded automatically at system startup as long as the saved data is valid.

The „Restore firmware defaults + reset“ option will restore every object's value to firmware-defined defaults.

This option must be used after every firmware update to load the correct startup values.

7.3 Control Gains Menu

This menu allows the adjustment of the control gains. Use with precaution as wrong control gain values can cause unexpected motor behaviour. It is recommended to alter the values only gradually.

These parameters are part of the non-volatile save list, so when saving the object dictionary, these values are preserved across power cycles.

7.4 Homing menu

The Homing menu will open a separate window for homing-control and homing setup.

Supported homing modes are described in the Knowledgebase and in the CiA402 standard document.

Requirements for this mode :

- Prior to using this option, Homing Mode must be selected and set in the Mode Of Operation dropdown menu.
- Input mapping must be set correctly according to the physical layout of the system > Depending on the homing mode used, limit switches has to be mapped to real inputs, paying extra attention to correct „high-active“ flag setting which determines the switch's default state. Negative limit switch must always located in CCW direction, and positive limit switch must be located in CW direction according to the move-range of the motor.

In addition to the standard, IDEA drive supports some more IDEA Drive-specific functions/objects :

As the IDEA drive relies on the „Run Current“ object in every mode, this is true for the Homing mode also. Run current sufficient enough to execute homing moves must be set correctly before using the homing mode.

Hold current is optional, but it is recommended to use a non-zero value for the hold current.

Move to origin after homing /Move to origin speed object :

If this option is selected, the motor will move to the origin position after the homing is finished successfully. This position is defined by the „Home Offset“ object, measured in encoder counts.

The IDEA-drive specific „Move to origin speed“ determines the speed at which the motor will move to the origin.

7.5 PDOs menu

This menu allows reading / setting PDO setup information. The GUI itself does not support use of PDOs.

Not all buttons have functions yet.

The list of mappable objects can be seen by loading the EDS file (Load EDS file button). This file is part of the GUI installation and it's detected automatically. The EDS file must be updated if any firmware change affects the „PDO mappable“ flag of any object, or mappable objects are added/deleted.

It is possible to setup single or multiple PDO's in GUI memory, and later to transfer the complete setup to the motor.

The „PDO source selection“ option determines if the GUI is showing PDOs in the memory or in the motor.

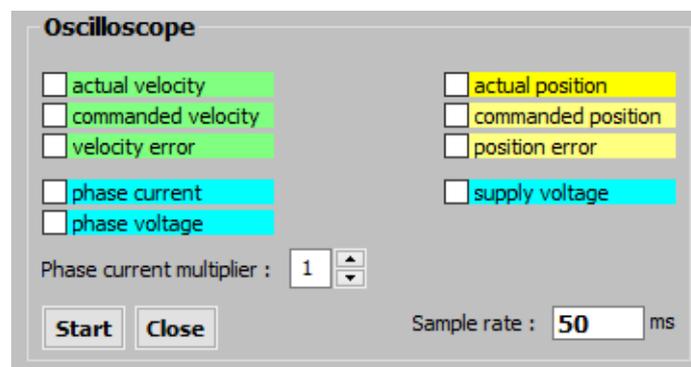
8 The „Status Register“ section

The 1002h Status Register contains the last error occurred, and in the GUI the corresponding name of the current error will light up. Its a visual representation of the most common errors. If an error is light continues to stay on, check the State Machine State, and if Fault State is present, use the Fault Reset button, then Set SM to Operation Enabled Mode again. If the error is still present, it is recommended to reset the drive and the GUI.

9 Oscilloscope

The oscilloscope offers visual representation of some values including velocity and position in real time.

The oscilloscope setup part and the available objects are shown on the picture bellow.



Upon checking the parameters the user would like to visualise, set the phase current multiplier and the sample rate, press „Start“ button, to turn the oscilloscope on, it will be opened in another window. The oscilloscope is „gnuplot“ based, thus the user can look up for shortcuts, functions and usage (eg.: right click + drag for zoom).

Phase current multiplier can be useful, when phase current is displayed in the same time as other values with high numbers on the same graph, for a better overall view.

Sample rate can be changed from 5ms to 5000ms. Higher rate can result in more detailed graph but it may cause gnuplot a hard time to generate. Lower rate is much easier to draw, but may lose some data inbetween two samples.

The oscilloscope sampling can be stopped using „Stop“ button on the GUI (in the place of „Start“), after it can be started again, to continue.

To turn off oscilloscope properly, press „Close“ button on the GUI, and not the default window close (X) button. If done improperly, the gnuplot co`re might stuck in the background.

