Simple motion sequence control of Lexium 28 via Modbus

Modbus control of Lexium 28 was not implemented in the design of Lexium 28.

Full Modbus control is not possible because there are no Modbus objects to access the operating mode and the control word for control of Lexium 28.

However, it is possible to make position moves of Lexium 28 via Modbus by using the position sequence mode.

What makes this possible is the ability to “force” the digital inputs via Modbus.

Lexium 28 digital inputs can be forced on or off via Modbus. This makes it possible to start position moves of Lexium 28 via Modbus even though Modbus control was not implemented in the design of Lexium 28.

In addition, the parameters for motion sequence mode are also accessible via Modbus which means the position, velocity, acceleration and deceleration can be accessed and written. Thereby providing a means to command position moves of Lexium 28 via Modbus.

Overview of functionality

The manual for Lexium 28 provides the Modbus address for each parameter but the value is provided in Hex. The hex value must be converted to decimal format and the resulting value is the Modbus address to be written to.

For example, the manual indicates that parameter P3-06 address is 40C.

40C hex converted to decimal is 1036.

Parameter name	Description	Unit Minimum value Factory setting Maximum value HMI Format	Data type R/W Persistent	Parameter address via field-bus
P3-06 SDI	Digital Inputs - Forcing Settings Applicable operating mode: PT, PS, V, T This parameter determines whether or not a digital input can be forced. Bits 0 ... 7: Digital input DI1 ... digital input DI8 Bit settings: Value 0: Digital input cannot be forced Value 1: Digital input can be forced To actually start forcing, you must write P4-07. See P2-10 ... P2-17 for the assignment of signal input functions to the digital inputs.	- 0_h 0_h $7FF_h$ Hexadecimal	u16 RW -	Modbus $40C_h$ CANopen 4306_h

Parameter P3-06 is the parameter that is used to determine which digital inputs can be “forced” via Modbus.

There are 8 digital inputs which are assigned to bits 1 – 8.

Bit 1 = 1

Bit 2 = 2

Bit 3 = 4

Bit 4 = 8

Bit 5 = 16

Bit 6 = 32

Bit 7 = 64

Bit 8 = 128

When assigning the inputs that will be available for forcing you will write to Modbus address 1036 the bit count for the inputs you wish to access. If you want to have forcing access to all inputs then a value of 255 will be written to Modbus address 1036.

1 + 2 + 4 + 8 + 16 + 32 + 64 + 128 = 255

If, for example, you only wanted access to digital input 1 and input 6 a value of 33 would be written to Modbus address 1036. 1 + 32 = 33

Next comes parameter P4-07. This is the parameter which permits you to alter the state of the digital inputs.

The parameter has a hex address of 50E, when converted to decimal is 1294

Parameter name	Description	Unit Minimum value Factory setting Maximum value HMI Format	Data type R/W Persistent	Parameter address via field-bus
P4-07 ITST	State of Digital Inputs / Activate Forcing Applicable operating mode: PT, PS, V, T A read access to this parameter indicates the state of the digital inputs in the form of a bit pattern. Example: Read value 0x0011: Digital inputs 1 and 5 are activated By writing this parameter, you can change the state of the inputs provided that the setting for the corresponding input in P3-06 allows for forcing (value 1 for the bit corresponding to the input). Example: Write value 0x0011: Digital inputs 1 and 5 are activated Read value 0x0011: Digital inputs 1 and 5 are activated See P3-06 for permitting forcing of individual digital inputs. See P2-10 ... P2-17 for the assignment of signal input functions to the digital inputs.	- 0_h 0_h FF_h Hexadecimal	u16 RW -	Modbus $50E_h$ CANopen 4407_h

The bit assignment is the same as before.

Bit 1 = 1

Bit 2 = 2

Bit 3 = 4

Bit 4 = 8

Bit 5 = 16

Bit 6 = 32

Bit 7 = 64

Bit 8 = 128

When you want to force on an input you simply write to Modbus address 1294 the bit you want active (on state).

For example, to force on all inputs again a value of 255 would be written to address 1294.

To force on input 5 and 6 a value of 48 would be written to address 1294.

Note that only the inputs which were selected in parameter P3-06 can be forced. You cannot force an input unless it was included in those selected when writing to P3-06.

Also note, the input state will remain as written until a new value is written.

To force the inputs off again you need to write a zero to the bits you want to force off.

For example, to force off input 5 but leave on input 6 a value of 32 would be written to address 1294.

First setup

Begin by commissioning the LXM28A for Motion Sequence Mode. Parameter P1-01 is set to a 1.

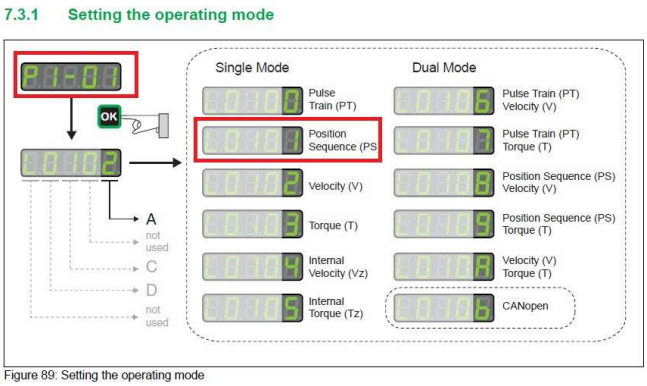
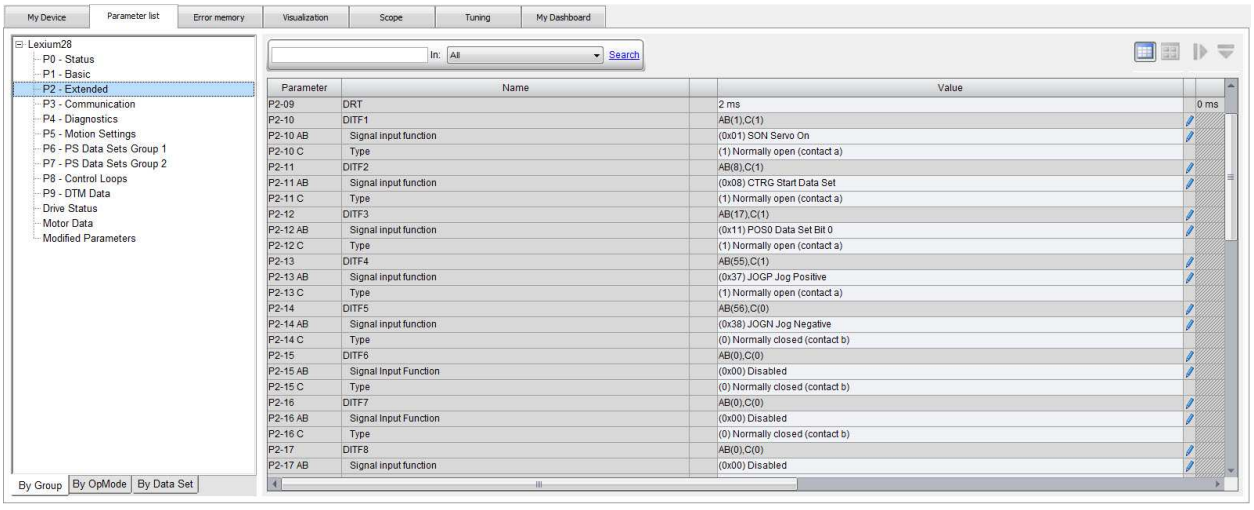
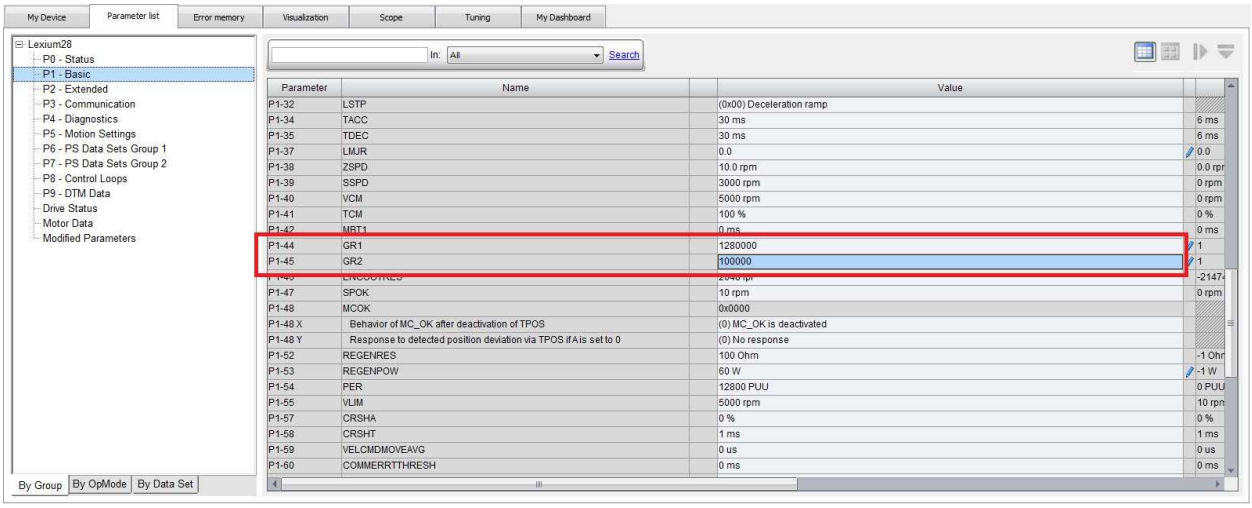


Figure 89: Setting the operating mode

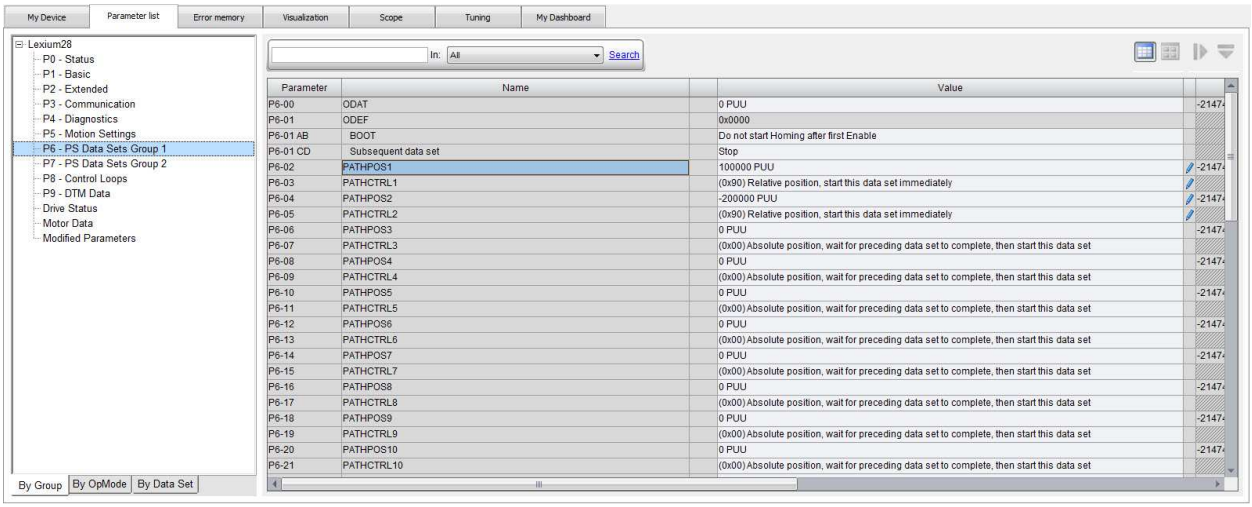
Next assign the inputs for the functionality you prefer. Using SoMove in the example seen below the inputs were assigned as Servo on, CTRG Start dataset, Data Set Bit 0, Jog Positive, and Jog Negative.



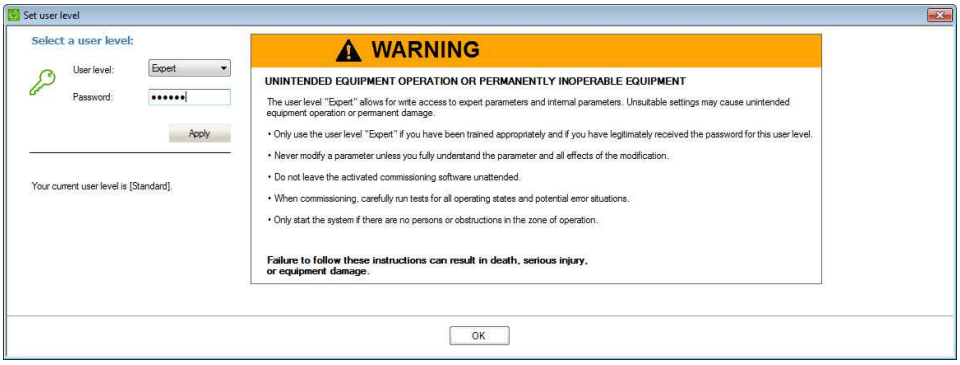
Set your preferred scaling factor for motion. Using SoMove In the example seen below the factory default scaling was used. 100000 counts = 1 motor revolution.



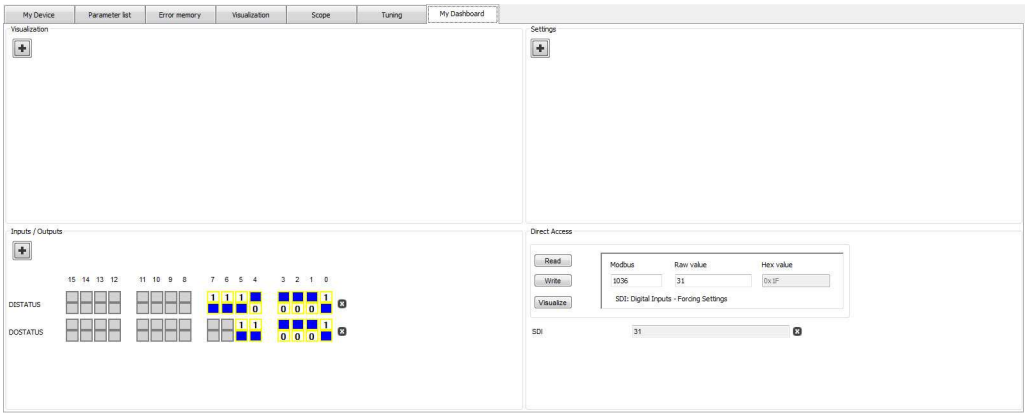
Assign basic motion sequence parameters for testing. Using SoMove in the example seen below task 1 rotates motor one full rotation clockwise and task 2 rotates the motor 2 full rotations counterclockwise.



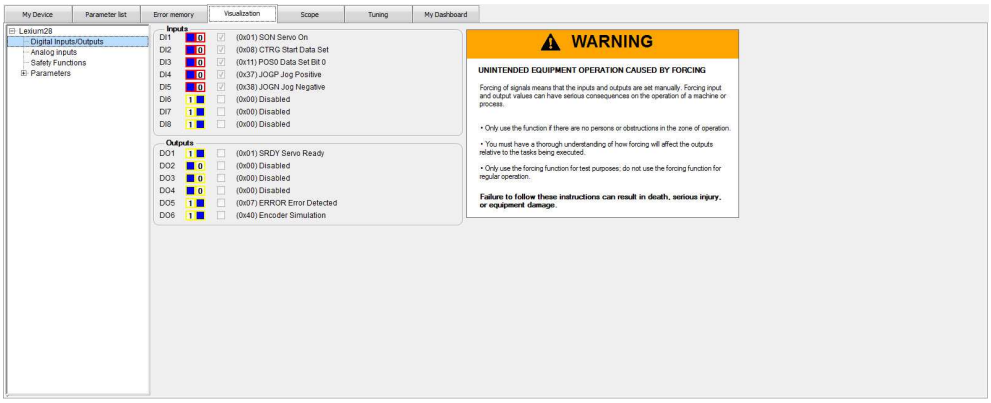
Now enter expert mode in SoMove. Password is 696969



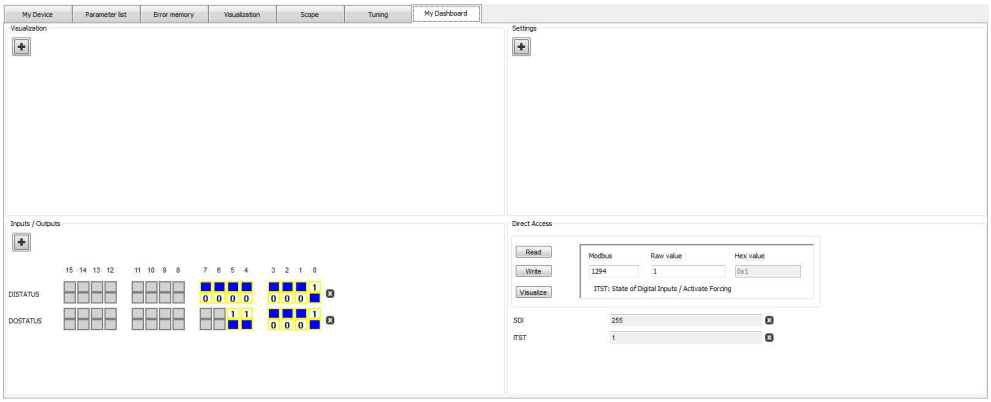
Using SoMove “My Dashboard”, write to Modbus address 1036 which inputs you want to be able to force. In the example shown, inputs 1 - 5 are going to be forced, so a value of 31 is written. Click the visualize button so the parameter can be viewed constantly.



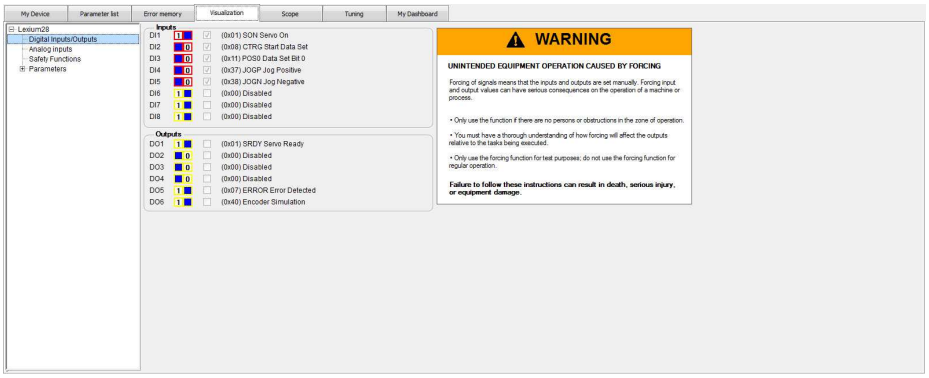
Shown below is the native state of the inputs prior to forcing. Notice that inputs 1 – 5 are accessible because those were selected in the previous step.



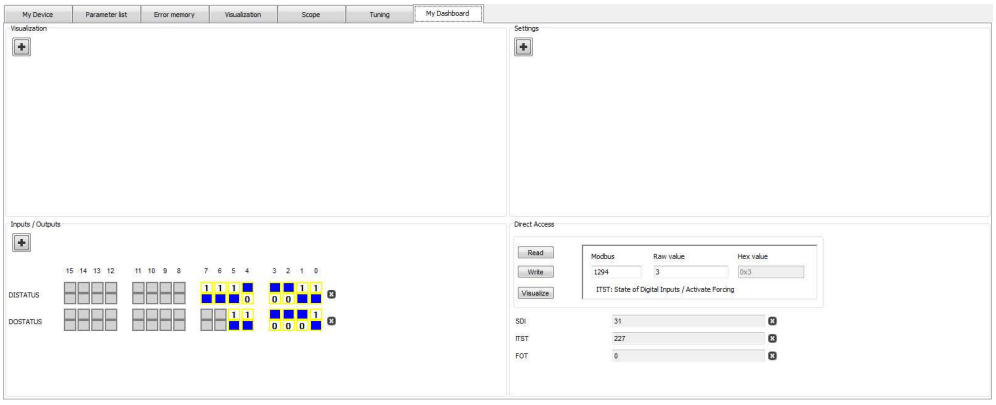
Next decide which inputs you wish to force on or off by writing to address 1294. In the example see below a value of 1 was written to 1294 thereby forcing on input 1. Click the visualize button so this parameter can also be seen constantly.



Below we can see the state of input 1 has changed after writing a 1 to address 1294 and the servo drive is now switched on.



Next we can initiate a relative move by forcing on the CTRG Start dataset input in addition to the Servo On input by writing a value of 3 to address 1294. Servo on bit = 1 + CTRG Start dataset bit = 2 for a total of 3. Immediately upon writing a value of 3 the motor will move 1 full rotation clockwise based on the previous PUU value that we stored in dataset 1.

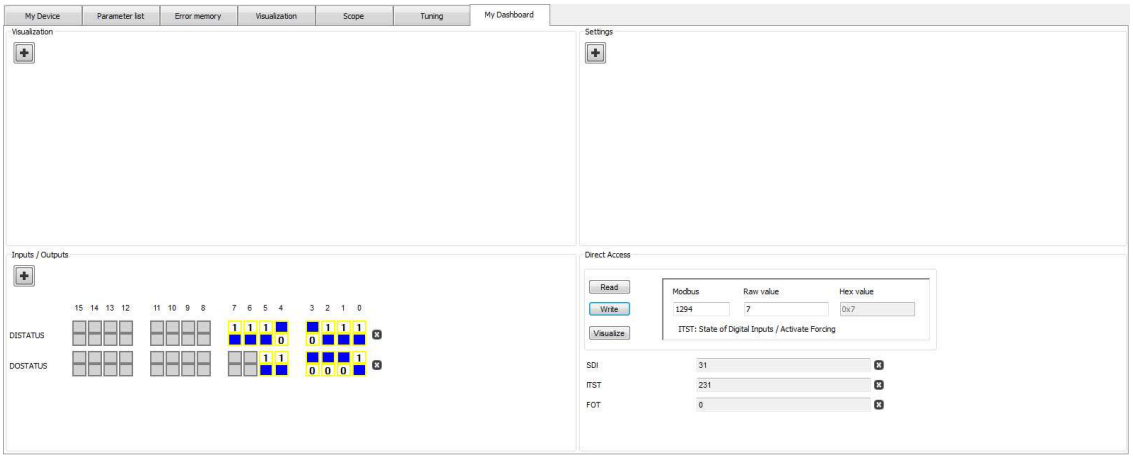


Note that the CTRG Start dataset bit is edge sensitive so another movement cannot be made until the bit is turned off and then turned on again.

If you want to keep repeating the movement profile of dataset 1, send 1 to address 1294 to turn off the start bit of CTRG Start dataset but still leave the Servo on bit true and then send 3 to address 1294 to activate CTRG Start dataset bit again while still leaving the servo on bit true.

Each time 1 and then 3 is written to address 1294 the CTRG Start dataset bit will toggle off then back on again while leaving the Servo On bit true thereby making the movement profile stored in dataset 1 every time this process is repeated.

If we wanted to execute dataset 2 a value of 7 will be written to address 1294. Servo On bit = 1 + CTRG Start dataset bit = 2 + Data Set Bit 0 = 4 for a total of 7.



As before, to repeat dataset 2 a value of 1 (servo on) is sent, then a value of 7 (Servo on + Data Set Bit 0 + CTRG Start dataset).

Switching between datasets

To toggle between dataset 1 and dataset 2 the written sequence to address 1294 would be, 1 for servo on, then 3 for servo on and CTRG Start dataset (no data bits are active so dataset 1 is performed), next write a 1 (servo remains on but CTRG Start dataset is toggled off) then write 7 (servo remains on, dataset select bit is true so dataset 2 is called, CTRG Start dataset is true so dataset 2 is started).

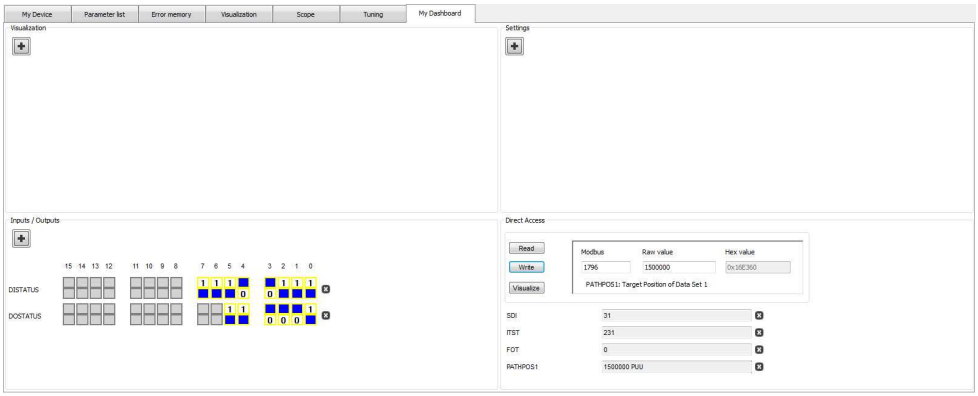
Writing new position distance

The next step is to write directly to the motion sequence tasks thereby making it possible to move to any position desired.

The target path for dataset 1 has a hex address of 704 which translates to 1796 decimal.

P6-02 PATHPOS1	Target Position of Data Set 1 Applicable operating mode: PS Bits 0 ... 31: Target position	PUU -2147483647 0 2147483647 Decimal	s32 RW per.	Modbus 704_h CANopen 4602_h

Using Modbus address 1796 write a new target path distance. The example below shows 1500000, which is 15 motor revolutions. 100000 (scaling for 1 rev) x 15 = 1500000



Sending a 3 to Modbus address 1294 starts dataset 1 with the new target distance and the motor proceeds to rotate 15 turns in a clockwise direction.

Using this process you can write to each and every dataset and make as many difference movement profiles as you would like.

Summary

It is possible to use Modbus control of LXM28 by operating the drive is position sequence mode and writing variables to the appropriate Modbus addresses.