MODBUS RTU

Modbus RTU communication bus user guide available for
NTT, TOMCAT EVO e DGFOX EVO drive.

USER GUIDE

Version data	Changes Description
09/09/2020 Rev: 4.0	• First release

Thanks for choosing this H.D.T. product.

www.hdtlovato.com

Read carefully this manual before using this product.

For continuous improvement, H.D.T. reserves the right to change features and specifications to manual and product without notice for the customer.

Details of images, contained here in, may be different from real product.

Doc.	Modbus RTU
Rev. N°	4.0
Date	09/09/2020

Summary

Ch. 1 Modbus RTU introduction

The purpose of this document is to provide a detailed description of the structure and parameters related to the Modbus, developed for the HDT's drives Tomcat, DGFOX and NTT. Detailed explanations about specific aspect of the Modbus protocol can be downloaded from the website

www.modbus.org

1.01 Configuration

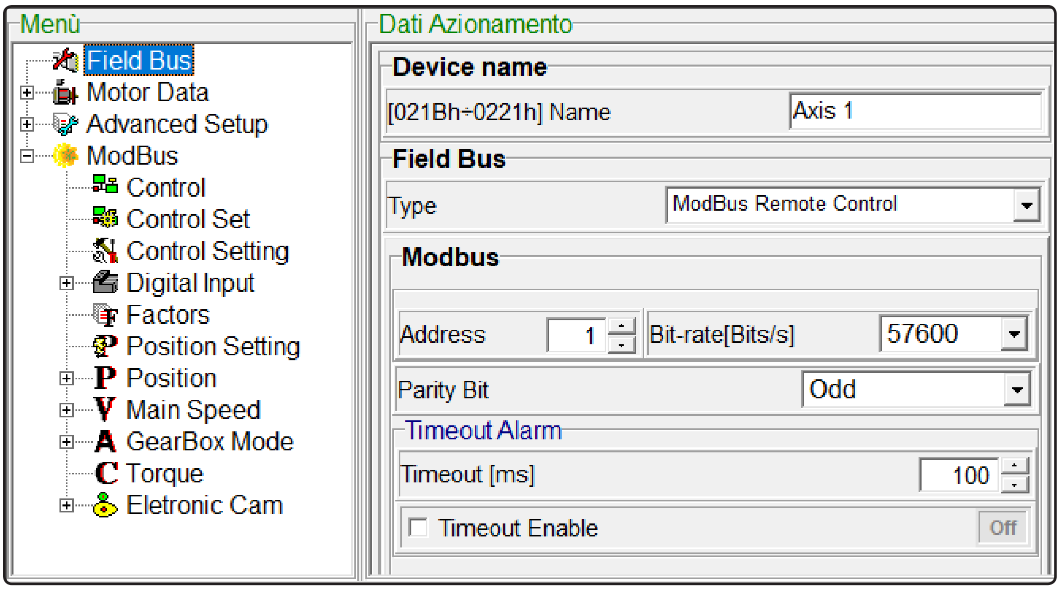
Modbus protocol is a messaging structure based on a serial bus transmission system. In every Modbus network all nodes must be assigned with a unique address, and the baudrate must be the same for all the connected devices.

The address and the baudrate can be set using a dip switch and a rotative selector for **TOMCAT EVO** and **DGFOX EVO**, or usiung a keypad for **NTT**. Alternatively, these parameters can be set using the configuration software **CALIPER** (in the menu "***Field Bus***").

From Caliper it is also possible to enable or disable a "*time-out alarm*", which monitor the Modbus activity: if this function is activated, whenever the data traffic stops for a period longer than the time set in the "*time-out*" panel, the drive goes into alarm.

A terminating resistor can be inserted externally, please refer to the respective user manuals of the drives to get more detailed information.

settable Baudrate from CALIPER	settable Baudrate from dip-switch
9.6 Kbit/s	9.6 Kbit/s
14.4 Kbit/s	-
19.2 Kbit/s	19.2 Kbit/s
38.4 Kbit/s	38.4 Kbit/s
57.6 Kbit/s	57.6 Kbit/s



Ch. 2 Theory of Modbus RTU protocol

2.01 Introduction

MODBUS protocol defines the format and the communication mode between a “master” who manages the system and one or more “slaves” that respond to the master. This protocol define how the master and the slaves start and stop their communication, how the transmitter and the receiver are identified, how messages need to be delivered and how errors are detected.

It’s possible to connect a master and up to 247 slaves on a common line. It should be noted that this is a logical limit of the Protocol, the physical interface may further restrict the number of devices, for example the standard interface of a RS-485 provides a maximum of 31 slaves connected to a line. By replacing the last element of the line with a special “bridge or repeater” you can connect other 31 slaves, and so on until the maximum logical number of appliable devices is reached.

Only the master can start a communication. There are two possible type of communication: it could be a question from the master directed to a single slave followed by the answer of the latter, or it could be a message from the master broadcasted to all the connected devices, which shall not be answered.

Some features of the protocol are :

- interface’s standard
- parity
- number of stop bits
- RTU’s size (binary).

2.02 Message format

In order to make two devices communicate, the message must be contained in a “box”. This box leave the transmitter via a “port” and is brought along the line to a similar “port” of the receiver.

MODBUS enstablish the format of this box that, for both the master and the slave,includes:

- The address of the device with which the master started the comunication (the address 0 corresponds to a broadcast message sent to all the slaves);
- The function code that should be or has been performed;
- the data to be exchanged;
- The error control according to the CRC16 algorithm.

If a device detects an error in the received message (of format, parity or CRC16 nature), the message is considered invalid and discarded. When a slave detects an error in the received message it will not execute the action required and it will not answer to the question, the same happen if the address in the message doesn’t match with an online device.

Address (1Byte)	Function code (1 Byte)	Data (1-252 Byte)	CRC16 (2 Byte)

2.03 Character format

The devices that use the MODBUS protocol normally use size “8, E, 1”(That is: 8 data bits, also parity check and 1 stop bit) or size “8, N, 1” (That is: 8 data bits, No parity and 1 stop bit), or the size “8,O,1” (That is: 8 data bits, Odd parity check and 1 stop bit).

2.04 The address

As already mentioned above, MODBUS' communications always involve the master, which manages the line, and one slave at a time (with the exception of the broadcast messages). To identify the target of the message a byte is transmitted, as the first character, that contains the numeric address of the selected slave.

Therefore all the slaves will be assigned a different address number that uniquely identifies it. The allowable addresses are those from 1 to 247, while address 0, which can not be assigned to a slave,when is put as first character of the message sent by the master indicates that it is "broadcast", that is directed to all slaves simultaneously. Only the messages that don't require any answer to carry out their function can be broadcasted, so they're only assegnation messages.

2.05 The function code

The second character of the message sent by the master identifies the function to perform, to which the slave responds with the same code to indicate that the function has been performed.

The MODBUS functions supported by **DGFOX EVO**, **TOMCAT EVO** and **NTT** are reported below:

Function	Description
01	Read Coil Status
02	Read Input Status
03	Read Holding Registers
04	Read Input Registers
06	Write Single Registers
16	Write Multiple Registers

2.06 the CRC16

The last two message's characters contains the CRC16 Code (Cyclic Redundancy Check) , which is calculated using the CRC16 algorithm. In order to calculate these last two characters the message(address, function code and data,excluding the start bit, the stop bit and the eventual parity) is considered as one continuos binary number whose most significant bit (MSB) is transmitted first.

See the appropriate documentation on the site www.modbus.org to check the calculation procedure of the CRC

2.07 Message synchronization

The synchronization of the messages between transmitter and receiver is obtained by interposing a pause between messages equal to at least 3.5 times the time of a character. If the receiver device doesn't get any data for a period as long as 3.5 characters, it will assume that the previous message is completed and that the next byte received is the first character of a new message and therefore an address.

2.08 Modbus functions

The detailed descriptions of the Modbus' Functions used by **DGFOX EVO**, **TOMCAT EVO** and **NTT** are explained below**.**

2.08.1 Read output status (01)

This function code is used to read the status of 1-2000 contiguous outputs of a device. The request defines the starting address, ie the address of the first output and the total number of outputs. The outputs are addressed starting from zero, so the outputs numbered 1-16 are addressed as 0-15.

The outputs in the response message are packed together, one for each bit of the data field and their status is shown as 1 = ON and 0 = OFF. The least significant bit of the first byte of the answer contains the first output addressed in the request. The other outputs are reported in ascending order in the following bits of the answer, from the least significant to the most significant.

If the number of the outputs returned is not a multiple of 8, the remaining bits of final byte of the answer will be fillled with 0. The ***Byte Count*** field of the answer return the number of Byte occupied with Data.

Here is an example of a request to read the status of outputs 1 and 2 of a Drive:

Question		Answer	
Name of the field	**Value (hex)**	**Name of the field**	**Value (hex)**
Function code	01	Function code	01
Starting address (MSB)	00	Byte count	01
Starting address (LSB)	00	Output 1 and 2 status	03
Number of output (MSB)	00		
Number of output (LSB)	02		

2.08.2 Read input status (02)

This function code is used to read the status of 1-2000 contiguous inputs of a device. The request defines the starting address, ie the address of the first entry and the total number of inputs. The inputs are addressed starting at zero, so inputs numbered 1-6 are addressed as 0-5.

The inputs in the response message are packed together, one for each bit of the data field and their status is shown as 1 = ON and 0 = OFF. The least significant bit of the first byte of the answer contains the first output addressed in the request. The other outputs are reported in ascending order in the following bits of the answer, from the least significant to the most significant.

If the number of the inputs returned is not a multiple of 8, the remaining bits of final byte of the answer will be fillled with 0. The ***Byte Count*** field of the answer return the number of Byte occupied with Data.

Here is an example of a request to read inputs 1 to 6 of Drive:

Question		Answer	
Name of the field	**Value (hex)**	**Name of the field**	**Value (hex)**
Function code	02	Function code	02
Starting address (MSB)	00	Byte count	01
Starting address (LSB)	00	Input status from 1 to 6	17
Number of input (MSB)	00		
Number of input (LSB)	06		

2.08.3 Read holding registers (03)

This function allows you to request the value of contiguous blocks of 16-bit(word) registers containing numeric variables.The master specifies the starting register address and the amount of contiguous registers to be read.

Here's an example to read the status of the 2 registers 061Ch and 061Dh (*measured position*), it is assumed that the return value is 128500dec (1F5F4h):

Question		Answer	
Name of the field	**Value (hex)**	**Name of the field**	**Value (hex)**
Function code	03	Function code	03
Starting address (MSB)	06	Byte count	04
Starting address (LSB)	1C	Register Value 061C (MSB)	00
Quantity of registers (MSB)	00	Register Value 061C (LSB)	01
Quantity of registers (LSB)	02	Register Value 061D (MSB)	F5
		Register Value 061D (LSB)	F4

2.08.4 Read input registers (04)

This function allows you to read from 1 to 125 contiguous input registers of a device. The master specifies the starting address and the number of registers to be read. The first register starts at address 0. The response data are packaged in two bytes per register with the binary contents right justified for each byte. For each register, the first byte contains the most significant bits and the second the least significant byte.

Here's an example to read only register inputs present on the drive:

Question		Answer	
Name of the field	**Value (hex)**	**Name of the field**	**Value (hex)**
Function code	04	Function code	04
Starting address (MSB)	00	Byte count	02
Starting address (LSB)	00	Register Value 00 (MSB)	00
Quantity of registers (MSB)	00	Register Value 00 (LSB)	17
Quantity of registers (LSB)	01		

2.08.5 Write single register (06)

This function allows to set the value of a single 16-bit register.

In addition to the address of the slave and the function code (06) the message contains the address of the variable expressed in two bytes, and the value to be assigned.The address numbering starts from zero (word1 = 0). In response the slave retransmits the message received after the variable has been changed.

Here's an example of the value of a write address register 0300h to 0003h:

Question		Answer	
Name of the field	**Value (hex)**	**Name of the field**	**Value (hex)**
Function code	06	Function code	06
Address of the register (MSB)	03	Address of the register (MSB)	03
Address of the register (LSB)	00	Address of the register (LSB)	00
Value of register (MSB)	00	Value of register (MSB)	00
Value of register (LSB)	03	Value of register (LSB)	03

Broadcast mode is allowed.

2.08.6 Preset Multiple Registers (16)

This function allows you to set the value of a consecutive block of 16-bit registers.

In addition to the address of the slave and the function code (16) the message contains the starting address (starting address), the number of words to be written, the number of bytes that contain the data and the data characters.

Here is an example of writing two registers starting from 'address 0601h: These 2 registers represent the 32 bit variable of the "*target position*". You want to write a total value of 655618 dec. Converted to hexadecimal corresponds to A0102h, then on the two registers are written respectively: 000Ah and 0102h

Question		**Answer**	
Name of the field	**Value (hex)**	**Name of the field**	**Value (hex)**
Function code	10	Function code	10
Starting address (MSB)	06	Starting address (MSB)	06
Starting address (LSB)	01	Starting address (LSB)	01
Quantity of registers (MSB)	00	Quantity of registers (MSB)	00
Quantity of registers (LSB)	02	Quantity of registers (LSB)	02
Number of Byte	04		
Register Value (MSB)	00		
Register Value (LSB)	0A		
Register Value (MSB)	01		
Register Value (LSB)	02		

Broadcast mode is allowed.

2.09 Error Management

In MODBUS there are two types of errors, handled differently : **transmission errors** and **operational errors** . Transmission errors alter the message changing its format, its parity (if used) or its CRC16 . A device that detects this type of errors in a message will treat the latter as invalid and won't answer.

If the message format is correct, but the required function cannot be performed for whatever reason , then there is an operational error. When this error appear the slave sends an exception message. This message is composed by an address, the code of the required function, an error code and the CRC. To denote that the answer sent is an error notification the function code is sent back with its most significant bit set to "1". this means that the range of values from 128 to 255 is reserved for the error notification "***exception responses***".

Hereafter a list of the "***function code***", with the related "***exception code***", implemented in the drives **DgFox EVO**, **TomCat EVO** and **NTT**. The format of all the "***function code***" and all the related answers follow the Modbus standard (see the document ***Modbus_Application_Protocol V1 1a.pdf***). In the event that the drive receives a "***function code***" different from those present, it returns an "***exception code***" response with error code 0x01.In the exception code's description some of the used variables are wrote in italics (like *address*, *quantity_coils* , etc. .) , since they refer to the explanation of the "***exception code***" contained in the Modbus document indicated above .

2.09.1 Exception codes

1. Function Code : **0x01** - (**Read Coil**) allows you to read the outputs of the drive.
   - Exception code **0x02**: when "*address*" is out of range (addresses allowed: 0 ÷ 1), or when the sum (*address* + *quantity_coils*) >2
   - Exception code **0x03**: when (*quantity_ coils* = 0) or (*quantity_ coils* > 2000)
2. Function Code : **0x02** - (**Read Discrete Inputs**) allows you to read the input of the drive.
   - Exception code **0x02**: when "*address*" is out of range (addresses allowed: 0 ÷ 5), or when the sum (*address* + *quantity_inputs*) >5
   - Exception code **0x03**: when (*quantity_ inputs* = 0) or (*quantity_ inputs* > 2000)

3. Function Code : **0x03** - (**Read Multiple Registers**) allows you to read a a contiguous block of registers.
   - Exception code **0x02**: when "*address*" is out of range, ie the address of the register or of the registers doesn't exist
   - Exception code **0x03**: when (*num_registers*= 0) or (*num_registers*> 125)
4. Function Code : **0x04** - (**Read Input Registers**) allows you to read the input registers.
   - Exception code **0x02**: when "*address*" is out of range (addresses allowed: 0), or when the sum (*address* + *num_registers*)>1
   - Exception code **0x03**: when (*num_registers*= 0) or (*num_registers*> 125)
5. Function Code : **0x06** - (**Write Single Register**) allows you to write a single register in RAM. When the drive is shut down the values written are lost; when the drive is turned on again the registers are loaded with the values saved before with the cofiguration software "**CALIPER**".
   - Exception code **0x02**: when the "*address*" is out of range, ie the Address of the register doesn't exist.
   - Exception code **0x04**: writing error because:
     a. The value is outside of the range allowed by the parameter
     b. The parameter is read-only
   - Exception code **0x06**: The register can not be written for safety reasons because:
     c. The drive is in state "***Switch On***" or "***Operation Enabled***"
     d. The drive is currently busy processing the data
     e. Because the local control from the configuration program "**CALIPER**" is active.
   - Exception code **0x07**: error occurred writing a 32 bit variable, because the two 16 bit-registers that compose this parameter were not sent consecutively (these two registers must be wrote starting from the lower Modbus address and then with the next one)
6. Function code: **0x16** - (**Write Multiple Registers**) allows you to write a block of contiguous registers in RAM. When the drive is shut down the values written are lost; when the drive is turned on again the registers are loaded with the values saved before with the cofiguration software "**CALIPER**".
   - Exception code **0x02**: when "*address*" is out of range, ie the address of the register or of the registers doesn't exist
   - Exception code **0x03**: when (*num_registers* = 0) or (*num_registers* > 123), or when (*num_byte* ≠ (2**num_registers*)).
   - Exception code **0x04**, **0x06** e **0x07**: the same as the "Function Code **0x06**" (**Write Single Register**).

Ch. 3 Modbus protocol parameters

3.01 Managing 32 bits variables

3.01.1 Writing 32 bits variables

The 32-bit variables are always made up of two 16-bit registers with consecutive Modbus addresses (see the parameter "*Target position*" accessible by the addresses 601H and 602h). In order to write a 32-bit parameter the register with the lower Modbus address must be written first (601H in the case of the parameter "*Target position*") and then the register with the higher address (602h in the case of the parameter "*Target position*"). If this sequence is not complied the drive send the **exception code 0x07**.

When the second register with the higher address has been received the resulting 32-bit parameter is verified: if it falls in range set for the parameter the latter is updated, otherwise the drive send an **exception code 0x04** to indicate that the parameter is out of range.

To write these variables we recommend the use of the **function code 16** (see *"2.08.6 Preset Multiple Registers (16)" pag. 14*).

3.01.2 Reading 32 bits variables

In order to read a 32-bit parameter, the register with the lower Modbus address must be requested first (61Ch in the case of the parameter "*Measured position*") and then the register with the higher address (61Dh in the case of the parameter "*Measured position*"). To read these registers use the **function code 03**.

3.02 Automatic Recovery undervoltage DC BUS

Depending on the value of the bit 1 of the register 201h "*alarm mode*" (*automatic reset from undervoltage*), the drive can behave in two different ways when there is an undervoltage event:

- Bit 1 set to "0 = saved": in this case the drive status remains in *fault* even when the voltage returns to a normal value; to switch the status to "*Switch off*" you must send a reset.
- Bit 1 set to "1 = auto reset": when the voltage returns to a normal value the undervoltage alarm is cancelled and the drive status automatically goes to "*Switch off*".

3.03 Automatic Recovery overvoltage DC BUS

Depending on the value of the bit 0 of the register 201h "*alarm mode*" (*automatic reset from overvoltage*), the drive can behave in two different ways when there is an overvoltage event:

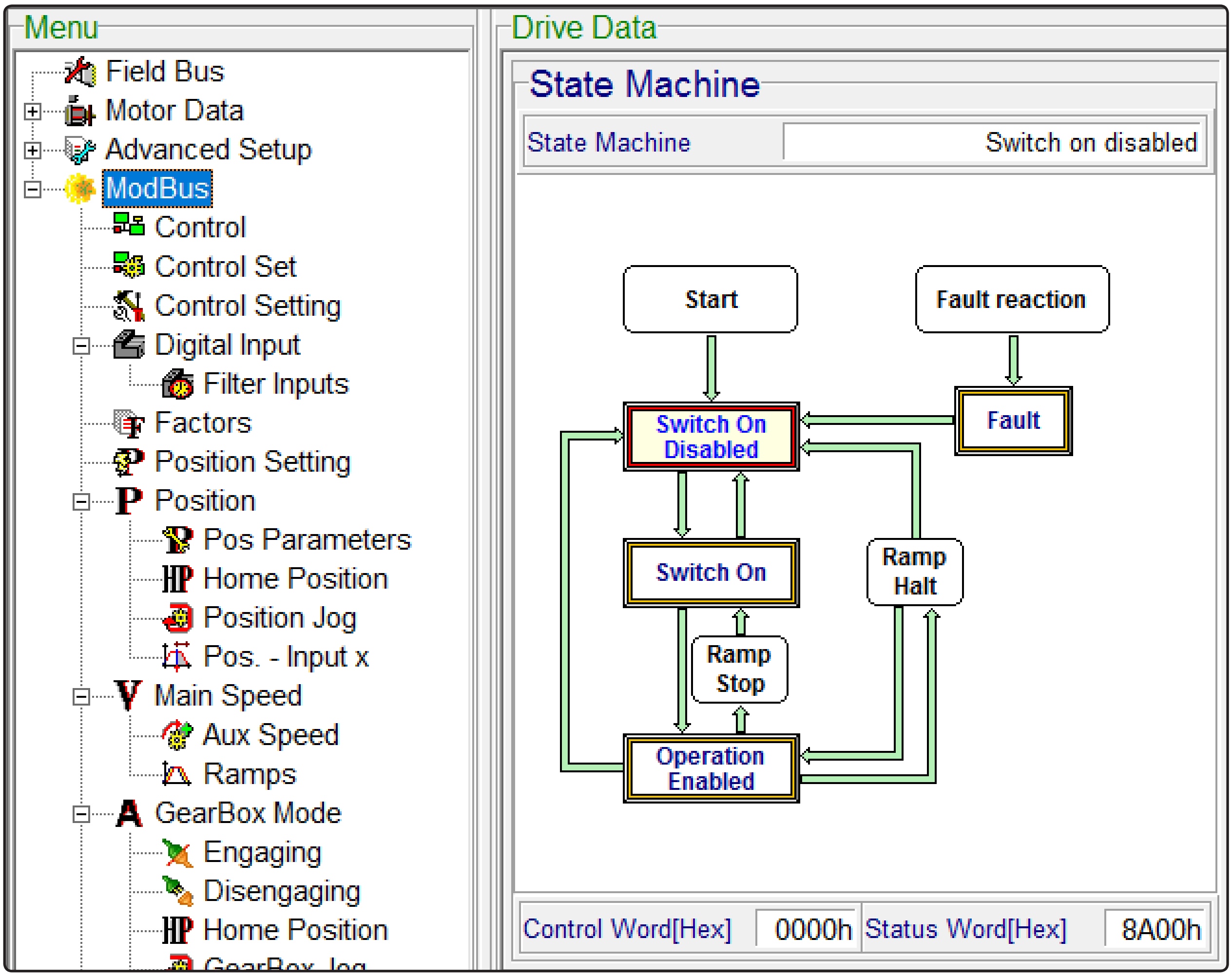
- Bit 0 set to "0 = saved": in this case the drive status remains in *fault* even when the voltage returns under the threshold value; to switch the status to "*Switch off*" you must send a reset.
- Bit 0 set to "1 = auto reset": when the voltage returns under the threshold value the overvoltage alarm is cancelled and the drive status automatically goes to "*Switch off*".

3.04 State Machine

The status of the drive is managed using a *finite state machine,* which control when the drive should be enabled or disabled, or if it should be turned to *fault* as a result of an alarm of the drive. All the various stages that involve a change of status of the drive are shown in the **status word**, a16-bit read-only variable located in the register 0301h. Reading the status word you can check at any time what is the actual state of the state machine.

The commands that allow you to switch from one state to another are instead managed by the master and are set through the **control word**, a16-bit variable located in the register 0300h.

State machine	
Status	**Description**
SWITCH OFF	• There is no alarm • Power supply disabled • values can be changed
SWITCH ON	• There is no alarm • Power enabled, the motor is stopped in torque • values can be changed
OPERATION ENABLED	• There is no alarm • Power enable and motor stand still • The drive is ready to execute commands on the operating mode • values can be changed
FAULT	• There is an alarm in the drive and the function of fault reaction is finished • values can be changed • Power supply disabled

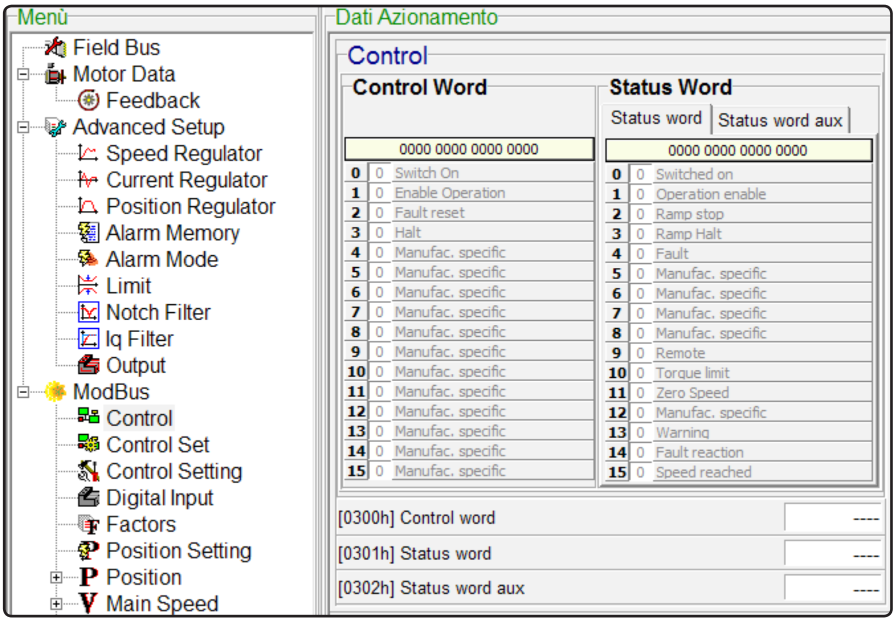


3.05 Control word and status word

This section describes the two control registers "Control word " and " Status word".

Indirizzo	Nome	Tipo	Read	Write
0300h	Control word	UINT16	always	always
0301h	Status word	UINT16	always	No
0302h	Status word aux	UINT16	always	No

The control word and status word can be displayed using Caliper Modbus clicking in the menu under "***Control***" in which the following screen appears:



3.05.1 Register 0300h (768 dec) - Control word

Address	Name	Type	Range	Default	Unit	Read	Write
0300h	**Control word**	UINT16	0..65535	0	Bit	Always	Always

Description bit **control word**:

15..4	3	2	1	0
Manufacturer specific / Operation mode specific	Halt	Fault reset	Enable operation	Switch on

3.05.1.a Control word Bits 0...3

The bit 0..3 manage the commands for switching the state of the state machine according to the following table :

Commands	Bits of the Control word				Description
	3	2	1	0	
	Halt	**Fault reset**	**Enable operation**	**Switch on**	
Switch off	x	0	0	0	The motor is free
Switch on	0	0	0	1	The motor is holding the torque but the references are disabled and the motor does not move
Enable operation	0	0	1	1	The motor is holding the torque and it can move according to the selected control
Halt	1	0	x	x	It provides a shutdown procedure, with the Halt's ramp
Fault reset	x		x	x	It provides an alarms' reset

3.05.1.b Control word Bits 4...15

The following bits are related with the operative mode:

Bit	Operative Mode					
	Speed	**Torque**	**Gearbox**	**Positioner**	**Electronic cam**	**Press**
4	reserved	reserved	Enable Axis	Start Positioning	Start Cam	Start press
5	reserved	reserved	Reset encoder	Jog -	Jog -	reserved
6	reserved	reserved	Reset position	Jog +	Jog +	reserved
7	reserved	reserved	Position shift	Homing	Homing	reserved
8	reserved	reserved	Speed shift +	Absolute/ relative	Bit0 - Select	reserved
9	reserved	reserved	Speed shift -	reserved	Bit1 - Select	reserved
10	reserved	reserved	Jog -	reserved	Bit2 - Select	reserved
11	reserved	reserved	Jog +	reserved	Speed shift +	reserved
12	reserved	reserved	Homing	reserved	Speed shift -	reserved
13	reserved	reserved	reserved	reserved	reserved	reserved
14	reserved	reserved	reserved	reserved	reserved	reserved
15	reserved	reserved	reserved	reserved	reserved	reserved

3.05.2 Register 0301h (769 dec)- Status word

Address	Name	Type	Range	Default	Unit	Read	Write
0301h	**Status word**	UINT16	0..65535	0	Bit	Always	No

Description of the bit of the **Status word**:

Bit	Description
0	Switch on
1	Enable operation
2	Ramp stop
3	Ramp Halt
4	Fault
5	Operation mode specific
6	Operation mode specific
7	Operation mode specific
8	Operation mode specific
9	Remote
10	Torque limit
11	Speed zero
12	Operation mode specific
13	Warning
14	Fault reaction
15	Manufacturer specific

3.05.2.a Status word- Bits related with the drive's status

Status	Bit status word	Description
SWITCH OFF	xxxx xxxx xxx0 0000	The motor is free
SWITCH ON	xxxx xxxx xxxx 0001	The motor is holding the torque, but the references are disabled so it doesn't move
OPERATION ENABLED	xxxx xxxx xxxx 0011	The motor is holding the torque and it can move according to the selected control
RAMPA STOP	xxxx xxxx xxxx 0111	the drive is executing the deceleration ramp of "***operational***". At the end of the ramp the motor remains in "Switch On"
RAMPA HALT	xxxx xxxx xxxx1011	The motor is executing the deceleration ramp of "***Halt***". At the end of the ramp the motor is free.
FAULT	xxxx xxxx xxx1 0000	The drive has detected an alarm and stops the motor in the selected mode in the "***fault reaction***"
FAULT REACTION	x1xx xxxx xxxx xx11	The drive is executing the function of fault reaction. When the motor is stationary the drive pass the status of "***Fault***"

The bits indicated by the x value are irrelevant

3.05.2.b Status word bits 5-8 e 12 - modo operativo

Bit	Operative Mode					
	Speed	Torque	Gearbox	Positioner	Electronic cam	Press
5	reserved	reserved	reserved	Position reached	Cam execution	Pressure reached
6	reserved	reserved	Gearbox enabled	Set point acknowleged	reserved	Underpressure
7	reserved	reserved	reserved	Homing executed	Homing executed	Overpressure
8	reserved	reserved	reserved	Position error	Position error	Power-limit
12	reserved	reserved	reserved	Position limit	reserved	Mode Press-velocity

3.05.2.c Status word bits 9,10,11,13 - Bit di segnalazione

Bit	Name	Description of the Status word's signaling bits
9	REMOTE	0: The commands are handled by Caliper and not by the MODBUS fieldbus 1: The commands are handled by the MODBUS fieldbus
10	TORQUE LIMIT	0: The drive is not in a torque limit 1: The drive is in torque limit
11	SPEED ZERO	0: Motor is rotating at a speed greater than the minimum speed threshold, set at 0210h, 0211h. 1: The motor is below the minimum speed threshold, set at 0210h, 0211h.
13	WARNING	0: The drive has no warning to be reported. 1: The drive found an warning. The drive can continue to operate. In the variables "Alarms" you can read the type of warning.

3.05.2.d Register 0302h (770 dec)- Status word Aux

Address	Name	Type	Range	Default	Unit	Read	Write
0302h	**Status word Aux**	UINT16	0..65535		Bit	Always	No

Description of the bits of the status word aux:

Currently it doesn't report any information.

Bit	Operative Mode		
	Gearbox	Positioner	Electronic cam
0	reserved	Position	reserved
1	reserved	Home Position	reserved
2	reserved	Jog Mode	reserved
3	reserved	Pos-Input1 reached	reserved
4	Measuring position	Measuring position	Measuring position
5 ... 15	reserved	reserved	reserved

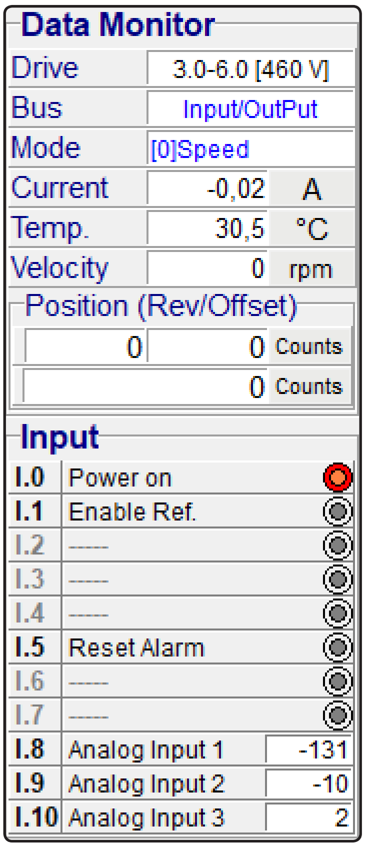
This read-only register is an extension of the main status word.
Currently is used only when the operative mode is set on "***Gearbox***", "***Positioner*** or "***Electronic Cam***".

3.06 Read-only variables of "Data Monitor"

This section describes the read-only variables that indicate the status of the drive:

Address	Name	Type	Read	Write
0303h	Variable alarms 1 (MSB)	UINT32	Always	No
0304h	Variable alarms 2 (LSB)			
0305h	Measured position (MSB)	INT32	Always	No
0306h	Measured position (LSB)			
0307h	Measured speed (MSB)	INT32	Always	No
0308h	Measured speed (LSB)			
0309h	Measured Current	INT16	Always	No
030Ah	Measured Voltage (DC bus)	INT16	Always	No
030Bh	State of inputs	UINT16	Always	No
030Ch	Multi-turns position	INT32	Always	No
030Dh	Single-turn position			
030Eh	Measured position (MSB)	INT32	Always	No
030Fh	Measured position (LSB)			
0310h	Measured speed (MSB)	INT32	Always	No
0311h	Measured speed (LSB)			
0312h	Electrical angle	UINT16	Always	No
0313h	Temperature of Drive's radiator	INT16	Always	No

The variables reporting the position, speed, current, temperature and the input status can be displayed using Caliper in "***Data Monitor***", as you can see from the illustration on the right:



3.06.1 Registers 0303h (771 dec) and 0304h (772 dec) Alarms Variable

The registers 0303h and 0304h are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0303h (MSB) **0304h (LSB)**	**Alarms variable**	UINT16	0..65535		Bit	Always	No

Address	Bit	Alarm code	Description of the alarms' Bits in the register 0303h
0304h	0	FA_01	Error parameters storage
	1	FA_02	Error offset currents
	2	FA_03	Power overcurrent
	3	FA_04	Over-voltage DC Bus
	4	FA_05	mancanza tensione
	5	FA_06	mancanza fase
	6	FA_07	Error position sensor of the motor
	7	FA_08	Secure disable
	8	FA_09	Over temperature motor
	9	FA_10	Braking resistor (only for Tomcat and NTT)
	10	FA_11	24 [V] absence
	11	FA_12	Under voltage DC Bus
	12	FA_13	Position error
	13	FA_14	Error home position
	14	F_15	Warinig I2t inverter
	15	F_16	Warning I2t Motor
0303h	0	F_17	Warning over speed
	1	FA_18	Secure disable error 1
	2	FA_19	(reserved)
	3	F_20	Fieldbus communication error
	4	FA_21	Fault memory
	5	FA_22	Motor phases Error
	6	FA_23	Alarm Secure disable
	7	FA_24	Heatsink overtemperature
	8	FA_25	Error sequence sensors hall motor
	9	FA_26	Speed error
	10	FA_27	Alarm I2t inverter
	11	FA_28	Alarm I2t motor
	12	FA_29	Alarm Simmetry DC Bus
	13	FA_30	Alarm Overpressure
	14	FA_31	Alarm Underpressure
	15	-	reserved

.

3.06.2 Registers 0305h (773 dec) and 0306h (774 dec) Measured position

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

These registers contain the position of the motor. The register "*MSB*" represents the most significative part of the measured position , while the register "*LSB*" represents the least significative part.

Address	Name	Type	Range	Default	Unit	Read	Write
0305h (MSB) **0306h (LSB)**	**Measured position**	INT32	±7FFFFFFFh			Always	No

This variable is reset to zero when the drive is turned on and also after the **Homing** procedure.

This variable is affected by the conversion factors, so the unit of measure is affected by them too.

3.06.3 Registers 0307h (775 dec) and 0308h (776 dec) Measured speed

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0307h (MSB) **0308h (LSB)**	**Measured speed**	INT32	±7FFFFFFFh		rpm	Always	No

These registers contain the measured speed of the motor. The unit of measure of this speed is affected by the conversion factors set.

3.06.4 Register 0309h (776 dec) Measured current

Address	Name	Type	Range	Default	Unit	Read	Write
0309h	**Measured current**	INT16	-32768.. +32767		A/100	Always	No

This register contains the average value of the Iq component of the measured current (ie the component that actually produces torque). This value is updated every 60 [ms]. The ***Measured current*** is reported in hundredths of an Ampere.

3.06.5 Register 030Ah (777 dec) Measured voltage

Address	Name	Type	Range	Default	Unit	Read	Write
030Ah	**Measured voltage**	INT16	-32768.. +32767		Volt	Always	No

This read-only register contains the voltage of the power capacitor bank. It is expressed in *Volts*.

3.06.6 Register 030Bh (779 dec) Digital inputs

This read-only register contains the logic state of the digital inputs

Address	Name	Type	Range	Default	Unit	Read	Write
030Bh	**Digital inputs**	UINT16	0..65535		Logic Level	Always	No

Common digital inputs for NTT, DGFOX and TOMCAT	
Bit	**Description**
0	Input I0 (1= high; 0= low): Generic input
1	Input I1 (1= high; 0= low): Generic input
2	Input I2 (1= high; 0= low): Generic input/ limitswitch CW (if used)
3	Input I3 (1= high; 0= low): Generic input/ limitswitch CCW (if used)
4	Input I4 (1= high; 0= low): Generic input/ Home sensor (if used)
5	Input I5 (1= high; 0= low): Generic input

Common digital inputs for DGFOX and TOMCAT	
Bit	**Description**
6	Input I6 (1= high; 0= low): Generic analog input, is also usable as a digital input
7	Input I7 (1= high; 0= low): Generic analog input, is also usable as a digital input

digital inputs only for NTT	
Bit	**Description**
6	Input I6 (1= high; 0= low): Generic input
7	Input I7 (1= high; 0= low): Generic input
8	Input I8 (1= high; 0= low): Generic analog input, is also usable as a digital input
9	Input I9 (1= high; 0= low): Generic analog input, is also usable as a digital input
10	Input I10 (1= high; 0= low): Generic analog input, is also usable as a digital input
11... 15	(reserved)

3.06.7 Registers 030Ch (780 dec) and 030D (781 dec) Enc SSI Absolute position

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

The 030Ch read-only register contains the most significant part of the encoder position: it represents the multi-turn component of the position read by the absolute encoder.

The 030Dh read-only register contains the least significant part of the encoder position: it represents the part of a turn read by the absolute encoder. A single turn is divided into 65535 counts.

Address	Name	Type	Range	Default	Unit	Read	Write
030Ch (MSB) **030Dh (LSB)**	**Absolute position**	INT32	±7FFFFFFFh		Giri	Always	No

These registers contain the position read by the absolute encoder SSI without any kind of conversion. This position can be displayed by the Caliper in the menu "***Motor data - Feedback***".

3.06.8 Registers 030Eh (782 dec) and 030Fh (783 dec) measured position

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

These registers contain the measured position of the motor. The MSB register contains the number of turns performed by the motor , the LSB register instead contains the fraction of turn performed by the motor (1 turn= 65535)

Address	Name	Type	Range	Default	Unit	Read	Write
030Eh (MSB) **030Fh (LSB)**	**Measured position**	INT32	±7FFFFFFFh		Giri	Always	No

This variable is reset to zero when the drive is turned on and also after the **Homing** procedure.

This variable is not affected by the conversion factors.

3.06.9 Registers 0310h (784 dec) and 0311h (785 dec) Measured speed

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0310h (MSB) **0311h (LSB)**	**Measured speed**	INT32	±7FFFFFFFh		Rpm o mm/sec	Always	No

These registers contain the measured speed of the motor. this speed is expressed in [rpm] if it's selected "***rotary motor***" or in [mm/s] if it's selected "***linear motor***". This variable is not affected by the conversion factors..

3.06.10 Register 0312h (786 dec) Electrical angle

Address	Name	Type	Range	Default	Unit	Read	Write
0312h	**Electrical angle**	INT16	0..65535		count	Always	No

This read-only register contains the absolute value of the angle of the electric motor standardized to 65535 (65535 corresponds to 360 degrees). In one mechanical revolution of the motor the electrical angle is repeated for a number of times equal to the number of pole pairs.

3.06.11 Register 0313h (787 dec) Drive's temperature

Address	Name	Type	Range	Default	Unit	Read	Write
0313h	**Drive's temperature**	INT16	0..1500		°/10	Always	No

this read-only register contains the drive's temperature expressed in tenths of degree.

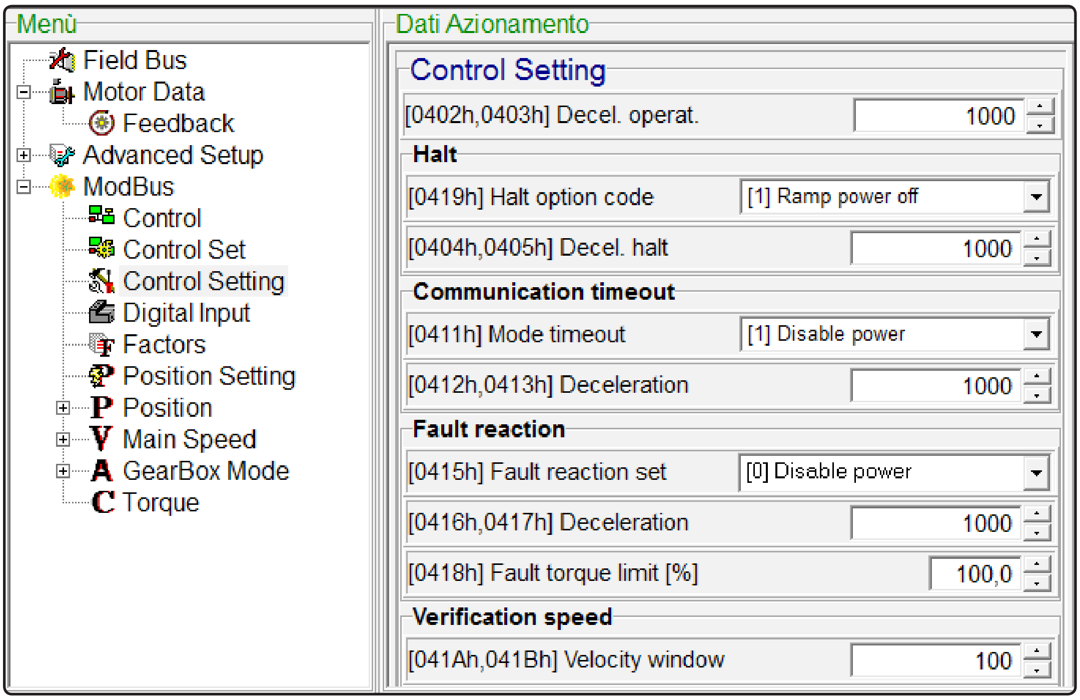
3.07 Modbus control variables

This section describes the control variables used in general MODBUS commands

Indirizzo	Nome	Tipo	Read	Write
0401h	Control type	UINT16	always	Switch On
0402h	Operational Deceleration (MSB)	UINT32	always	always
0403h	Operational Deceleration (LSB)			
0404h	Halt Deceleration (MSB)	UINT32	always	always
0405h	Halt Deceleration (LSB)			
040Eh	Error position (MSB)	UINT32	always	always
040Fh	Error position (LSB)			
0410h	Error position time	UINT16	always	always
0411h	Loss fieldbus timeout mode	UINT16	always	always
0412h	Deceleration timeout mode (MSB)	UINT32	always	always
0413h	Deceleration timeout mode (LSB)			
0415h	Fault reaction mode	INT16	always	always
0416h	Fault reaction deceleration (MSB)	UINT32	always	always
0417h	Fault reaction deceleration (LSB)			
0418h	Fault reaction torque limit	UINT16	always	always
0419h	Halt option	INT16	always	always
041Ah	Speed window (MSB)	UINT32	always	always
041Bh	Speed window (LSB)			
041Ch	Speed window 's time (ms)	UINT16	always	always

In the next paragraph is explained in what menu of Caliper is displyed the register 0401h.

In the paragraphs *"3.07.5 Register 0410h (1040 dec) - Error position time" pag. 30* are explained in what menu of Caliper is displyed the registers 040Eh, 040Fh and 0410h. All the other variables can be displayed using Caliper in the menu " ***Modbus-Control setting*** ", as you can see in the picture below:



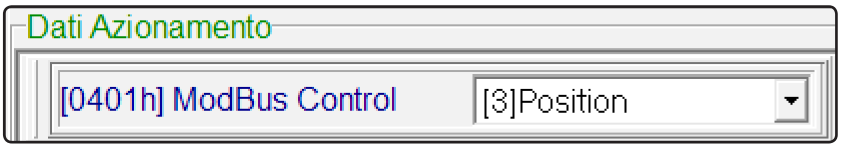
3.07.1 Register 0401h (1025 dec) Control type

Address	Name	Type	Range	Default	Unit	Read	Write
0401h	**Control type**	UINT16	0..5	0		Always	Switch On

The value in this register is used to select the type of control that you want, depending on your application. In the table below are reported the control types (with the associated value of 0401h) implemented in ***DGFOX EVO***, ***TOMCAT EVO*** and ***NTT***.

Value	Description
0	***Speed Control***: the current loop and the speed loop are activated, the drives follow the speed reference sent by the master or by an analog signal
1	***Torque Control***: in this mode you control directly the torque using only the current loop (because the torque is proportional with the quadrature current)
2	***Gearbox Control***: in this mode the motor follows the position reference obtained making the product between the number of impulses from an external encoder and the ratio of the gearbox
3	***Positioner Control***: in this mode the current loop, speed loop and position loop are abilitated; once you have set the desired position the drive moves towards it. The movement can be performed following a speed profile with trapezoidal shape or with an "S" shape; the latters are smoothed by the jerk set.
4	***Electronic Cam Control***: In this mode the "slave "motor follows a position reference which depends the number of impulses from an external encoder (the "Master") and the drawed cam profile. The movement can be performed following a speed profile with trapezoidal shape or with an "S" shape; the latters are smoothed by the jerk set.
5	***Pressure Control***: this mode is used to control the pressure of an hydraulic control unit; this is done controlling the servo pump installed on it. the current loop, speed loop and pressure loop are abilitated. The pressure loop takes as a reference an analogic signal, and confronts it with the analogic signal produced by a transducer that measures the real pressure. The pressure loop then produces the speed reference for the speed loop. The speed profile and the pressure profile can be performed with trapezoidal shape or with an "S" shape.

The setting of this control parameter can be displayed using Caliper clicking on menu "***Modbus-Control Set***" in which the following window appears:



3.07.2 Registers 0402h (1026 dec) and 0403 (1027) Operational Deceleration

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0402h (MSB) **0403h (LSB)**	**Operational Deceleration**	UINT32	1..7FFFFFFFh	1000 (dec)		Always	Always

These registers are used to set the value of deceleration used to stop the motor when it changes from ***Operational*** state to ***Switch-on*** state. This value is converted into an internal unit of measure of the drive (increments on second squared [increments/s^2]) using the ***Acceleration's factors***: the registers 040Ah-040Bh (*Numerator*) and the registers 040Ch-040Dh (*Denominator*). the conversion is done following this formula:

$$drive\_operational\_deceleration = \frac{(Numerator \times Operational\_deceleration)}{Denominator}$$

Considering that the drive divides a single turn in 65536 counts, using the default values of the factors, ie *Numerator* = 65536 and and *Denominator* = 60, the ***Operational deceleration*** is expressed in [rpm/s]

3.07.3 Registers 0404h (1028 dec) and 0405h (1029 dec) Halt Deceleration

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0404h (MSB) **0405h (LSB)**	**Halt** **Deceleration**	UINT32	1..7FFFFFFFh	1000 (dec)		Always	Always

These registers are used to set the value of deceleration used to stop the motor when it receives the ***Halt*** command. This value is converted into an internal unit of measure of the drive (increments on second squared [increments/s^2]) using the ***Acceleration's factors***: the registers 040Ah-040Bh (*Numerator*) and the registers 040Ch-040Dh (*Denominator*). the conversion is done following this formula:

$$drive\_Halt\_deceleration = \frac{(Numerator \times Halt\_deceleration)}{Denominator}$$

Considering that the drive divides a single turn in 65536 counts, using the default values of the factors, ie *Numerator* = 65536 and and *Denominator* = 60, the ***Halt deceleration*** is expressed in [rpm/s]

3.07.4 Registers 040Eh (1038 dec) and 040Fh (1039 dec) - Max Error position

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

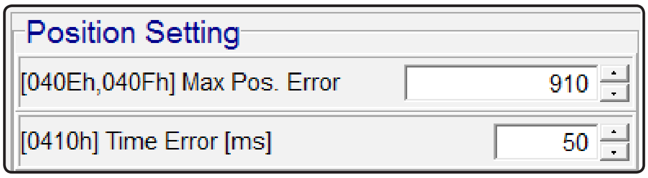
Address	Name	Type	Range	Default	Unit	Read	Write
040Eh (MSB) **040Fh (LSB)**	**Max Error** **position**	INT32	1..FFFFFFFFh	910 (dec)		Always	Always

These registers contain the maximum acceptable position error when the drive is working with the ***positioner control***. The ***Max Error position*** has the same unit of measure of the ***Target position*** (registers 0601h, 0602h). This value is converted into an internal unit of measure of the drive (increments) using the ***Position factor numerator*** (registers: 0609h,060Ah) and the ***Position factor denominator*** (registers: 060Bh,060Ch). Considering that by default the ***Position factor numerator*** and the ***Position factor denominator*** are set to 1 and that a single turn corresponds to 65536 increments, by default 910 increments correspond to approximately 5°.

3.07.5 Register 0410h (1040 dec) - Error position time

Address	Name	Type	Range	Default	Unit	Read	Write
0410h	**Error position time**	INT16	10..4000	50	msec	Always	Always

This perameter sets the maximum time interval for which the position error may exceeds the ***Max Error position***; if this happens, the drive passes to the fault state and signals it in the appropriate bits of status word. This variable can be displayed by the Caliper in the menu "***Position setting***" in which the following screen appears:



3.07.6 Register 0411h (1041 dec) - Loss fieldbus timeout mode

Address	Name	Type	Range	Default	Unit	Read	Write
0411h	**Loss fieldbus timeout mode**	UINT16	0..2	1		Always	Always

Value	Description
0	No action: a warning is signaled on bit 13 of the status word and the bit 3 of ***Variable alarm 2*** is rised
1	the power is disabled (the motor coasts to stop) and the ***Fault*** is signaled
2	The motor stops with the deceleration ramp set to "***deceleration timeout mode***". Once stopped the power is disabled and the ***Fault*** is signaled

Using the Software "***Caliper"*** it's possible to set a maximum time interval in which is accepted a communication loss in the Modbus channel. if this time interval is surpassed the Drive reacts as selected in this register.

3.07.7 Registers 0412h (1042 dec) and 0413h (1043 dec) - timeout mode deceleration

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.* The register 0412h contains the most significant part of the deceleration.

Address	Name	Type	Range	Default	Unit	Read	Write
0412h (MSB) **0413h (LSB)**	**timeout mode** **Deceleration**	INT32	10.. FFFFFFFFh	1000 (dec)		Always	Always

These registers contain the value of the deceleration used to stop the motor when there is a communication loss in the Modbus channel. This value is converted into an internal unit of measure of the drive (increments on second squared [increments/s^2]) using the ***Acceleration's factors***: the registers 040Ah-040Bh (*Numerator*) and the registers 040Ch-040Dh (*Denominator*). The conversion is done following this formula:

$$drive\_timeout\_mode\_deceleration = \frac{(Numerator \times timeout\_mode\_deceleration)}{Denominator}$$

Considering that the drive divides a single turn in 65536 counts, using the default values of the factors, ie *Numerator* = 65536 and and *Denominator* = 60, the ***timeout mode deceleration*** is expressed in [rpm/s]

3.07.8 Register 0415h (1045 dec) - Fault reaction set

Address	Name	Type	Range	Default	Unit	Read	Write
0415h	**Fault reaction set**	INT16	0..2	1		Always	Always

Valore	Descrizione
0	the power is disabled (the motor coasts to stop) and the ***Fault*** is signaled
1	Deceleration ramp: The motor stops with the deceleration ramp sets in "***fault reaction deceleration***". Once stopped the power is disabled and the ***Fault*** is signaled
2	Torque limit: the motor stops with the torque set on "***Fault torque limit***" and the ***Fault*** is signaled

Using the parameter ***"Fault reaction set***" is possible to select how the drive will respond when some alarms occur. The drive performs the selected reaction remaining in "***Fault reaction"*** state (bit14 of the status word=1); once it stops , the power is disabled and the Drive switch to "***Fault"*** state. The reaction selected by ***"Fault reaction set***" will be performed only for this alarms:

Alarms managed by Fault reaction set	
Alarm	**Code**
Under Voltage DC BUS	FA - 12
Position error	FA - 13

3.07.9 Registers 0416h (1046 dec) and 0417h (1047 dec) - fault reaction deceleration

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.* the register 0416h contains the most significant part of the deceleration.

Address	Name	Type	Range	Default	Unit	Read	Write
0416h (MSB) **0417h (LSB)**	**fault reaction deceleration**	UINT32	10.. FFFFFFFFh	1000 (dec)		Always	Always

These registers contain the value of the deceleration used to stop the motor when there is a ***Fault reaction*** and ***Fault reaction deceleration*** is selected on register 0415h. This value is converted into an internal unit of measure of the drive (increments on second squared [increments/s^2]) using the ***Acceleration's factors***: the registers 040Ah-040Bh (*Numerator*) and the registers 040Ch-040Dh (*Denominator*). the conversion is done following this formula:

$$drive\_fault\_reaction\_deceleration = \frac{(Numerator \times fault\_reaction\_deceleration)}{Denominator}$$

Considering that the drive divides a single turn in 65536 counts, using the default values of the factors, ie *Numerator* = 65536 and and *Denominator* = 60, the ***Fault reaction deceleration*** is expressed in [rpm/s]

3.07.10 Registro 0418h (1048 dec) Torque limit fault

Address	Name	Type	Range	Default	Unit	Read	Write
0418h	**Torque limit fault**	UINT16	0..3000	1000		Always	Always

This register is used to set the value of the torque current (Iq) used to stop the motor when there is a ***Fault reaction*** and ***Torque limit*** is selected in register 0415h. The current value is expressed in thousandths of the ***Nominal motor current*** (register 0153h) using this formula:

$$Torque\_current = \frac{(Torque\_limit\_fault) * (Nominal\_current)}{(1000)}$$

3.07.11 Register 0419h (1049 dec) Halt option code

Address	Name	Type	Range	Default	Unit	Read	Write
0419h	**Halt option code**	UINT16	0..2	0		Always	Always

The parameter ***Halt Option code*** is used to select how to react when the bit 3 of the control word (assigned to the ***Halt***) is raised to 1. According to the value of this parameter the drive will react with one of the ways reported below:

Value	Halt option code	Description
0	Turn off the power, the motor is free to rotate	Power is disabled and the motor stops for inertia. The drive switch to "***Switch Off***" state
1	Power Off Ramp	The drive switches to speed control, and the motor is commanded a stop in ramp with deceleration set on the parameters 0404h e 0405h. At the end of the ramp the servodrive switch to "***Switch off***" state (power disabled)
2	Power On Ramp	The drive switches to speed control and the motor is commanded a stop in ramp with deceleration set on the parameters 0404h e 0405h.At the end of the ramp the servodrive remains in "***Switch on***" (power enabled)

3.07.12 Registers 041Ah (1050 dec) and 041B (1051 dec) -Velocity window

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
041Ah (MSB) **041Bh (LSB)**	**Velocity window**	UINT16	1..FFFFFFFFh	100 (dec)		Always	Always

The ***Velocity window*** defines a speed range around the reference speed from which the measured speed shouldn't exit:

[Reference speed – Velocity window ÷ Reference speed + Velocity window]

When the measured speed remains inside the range defined by "***Velocity window*** " for the time set with register 041Ch, the bit 15 of the status word is raised to 1.

This value is converted into an internal unit of measure of the drive using the ***speed factor numerator*** (registers 0406h, 0407h) and ***speed factor denominator*** (registers 0408h,0409h). The conversion is done following this formula:

$$drive\_velocity\_window = \frac{(Speed\_factor\_Numerator \times velocity\_window)}{(Speed\_factor\_Denominator)}$$

Using the default values of the factors, ie *Speed_factor_Numerator* = 65536 and and *Speed_factor_Denominator* = 60, the ***velocity window*** is expressed in [rpm].

3.07.13 Registro 041Ch (1052 dec) Vel. window time

Address	Name	Type	Range	Default	Unit	Read	Write
041Ch	**Vel. window time**	UINT16	1..10000	200	ms	Always	Always

"Vel. window time" is used to verify if the measured speed is inside the speed range defined by "***Velocity window***" (registri 041Ah, 041Bh). If the measured speed remains inside the range cited above for a time longer than **"Vel. window time"**, than the bit 15 of the status word is raised to 1.

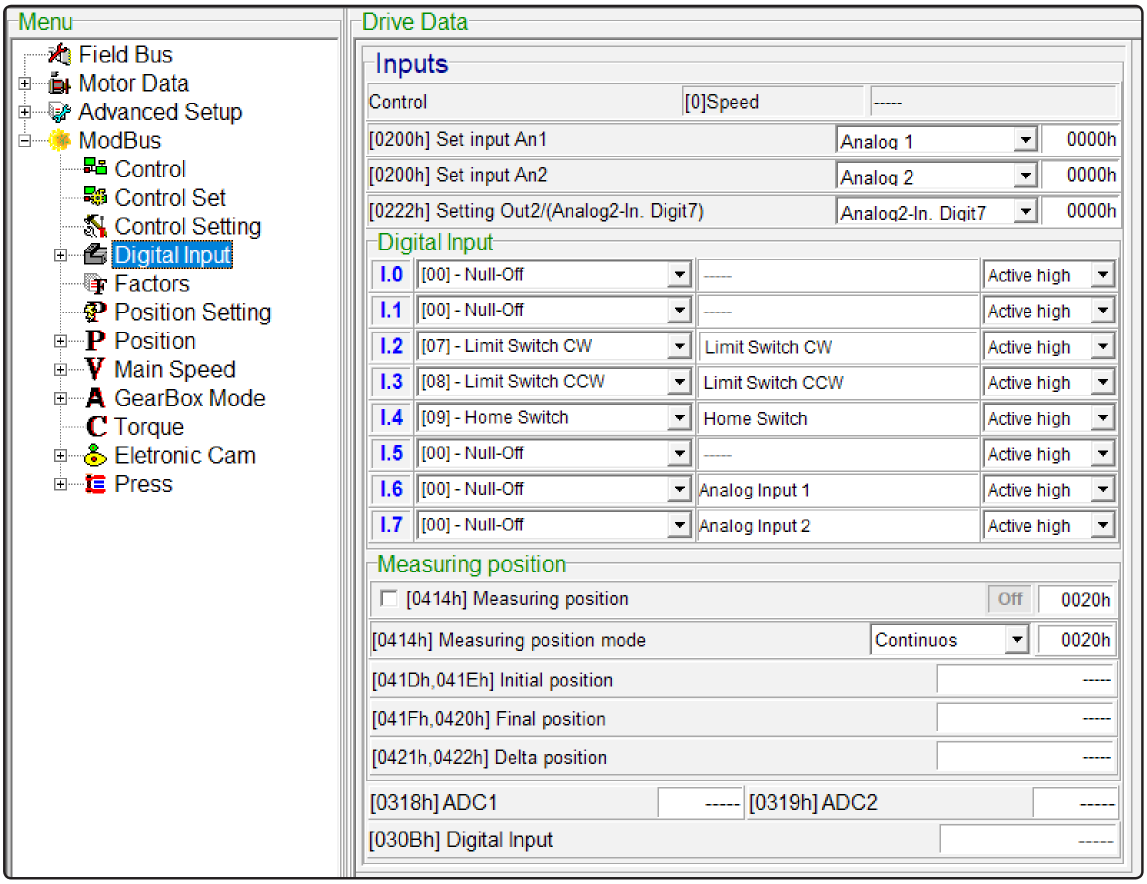
3.08 Digital inputs

This section describes the control variables used to set the digital and analogical inputs. In the table below there is a list of all the used registers.

Address	Name	Type	Read	Write
0200h	Miscellaneous Flag	INT16	always	always
0318h	ADC1	INT16	Always	No
0319h	ADC2	INT16	Always	No
031Ah	ADC3	INT16	Always	No
0414h	Digital Input (flag)	UINT16	Always	Always
041Dh 041Eh	Initial Position	INT32	Always	No
041Fh 0420h	Final Position	INT32	Always	No
0421h 0422h	Delta position	INT32	Always	No
0423h	Filter time input 0	UINT16	Always	Always
0424h	Filter time input 1	UINT16	Always	Always
0425h	Filter time input 2	UINT16	Always	Always
0426h	Filter time input 3	UINT16	Always	Always
0427h	Filter time input 4	UINT16	Always	Always
0428h	Filter time input 5	UINT16	Always	Always
0429h	Filter time input 6	UINT16	Always	Always
042Ah	Filter time input 7	UINT16	Always	Always
042Bh	Digital input I.0	UINT16	Always	Switch off
042Ch	Digital input I.1	UINT16	Always	Switch off
042Dh	Digital input I.2	UINT16	Always	Switch off

Address	Name	Type	Read	Write
042Eh	Digital input I.3	UINT16	Always	Switch off
042Fh	Digital input I.4	UINT16	Always	Switch off
0430h	Digital input I.5	UINT16	Always	Switch off
0431h	Digital input I.6	UINT16	Always	Switch off
0432h	Digital input I.7	UINT16	Always	Switch off
0433h	Digital Input (flag)	UINT16	Always	Switch off
0448h	Filter time input 8	UINT16	Always	Always
0449h	Filter time input 9	UINT16	Always	Always
044Ah	Filter time input 10	UINT16	Always	Always
044Bh	Digital input I.8	UINT16	Always	Switch off
044Ch	Digital input I.9	UINT16	Always	Switch off
044Dh	Digital input I.10	UINT16	Always	Switch off

These registers can be seen and modified in the menu "***Modbus - digital Input***" in which the following screen appears:



3.08.1 Register 0200h (512 dec) , Bit 2, 3 ,11 – Set input An1,An2 e An3

Indirizzo	Nome	Tipo	Range	Default	Unit	Read	Write
0200h		UINT16	0...65536	0		Always	Always

This register is recalled in several paragraphs of this manual, because its bits are used as flags in various parts of the firmware. Its bits 2,3 and 11 are used as flags,with which it is decided whether to use the three analog inputs as such or whether to use them as additional digital inputs.

<table>
<tr><th>Bit</th><th colspan="2">Description</th></tr>
<tr><td rowspan="4">2</td><td colspan="2">Set Input An1</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Analogic 1</td></tr>
<tr><td>1</td><td>Digital input 8</td></tr>
<tr><td rowspan="4">3</td><td colspan="2">Set Input An2</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Analogic 2</td></tr>
<tr><td>1</td><td>Digital input 9</td></tr>
<tr><td rowspan="4">11</td><td colspan="2">Set Input An3 (only NTT)</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Analogic 3</td></tr>
<tr><td>1</td><td>Digital input 10</td></tr>
</table>

3.08.2 Register 0318h (1057 dec) - ADC1

Address	Name	Type	Range	Default	Unit	Read	Write
0318h	**ADC1**	INT16	-32768.. +32767			Always	No

This register contains the voltage value read by the external ad-converter of the main differential input. The voltage value can varies in the range ±10 [V], and has a resolution of 16 bit for **NTT,** and12 bit for **DGFox EVO** and **TomCat EVO**.

3.08.3 Register 0319h (1058 dec) - ADC2

Address	Name	Type	Range	Default	Unit	Read	Write
0319h	**ADC2**	INT16	-32768.. +32767			Always	No

This register contains the voltage value read by the external ad-converter of the first auxiliary differential input. The voltage value can varies in the range ±10 [V], and has a resolution of 12 bit.

3.08.4 Register 031Ah (1059 dec) - ADC3

Address	Name	Type	Range	Default	Unit	Read	Write
031Ah	**ADC3**	INT16	-32768.. +32767			Always	No

This register contains the voltage value read by the external ad-converter of the second auxiliary differential input. The voltage value can varies in the range ±10 [V], and has a resolution of 12 bit. This differential input is present only in the **NTT.**

3.08.5 Register 0414h (1044 dec) Digital Input (flag)

Address	Name	Type	Range	Default	Unit	Read	Write
0414h	**Digital Input**	UINT16	0..007Fh	003Ah		Always	Always

This register contains some flags. these flags are used to set how the digital Inputs are used.

Bit	**Digital input (flag) -bit description**	
0	Reserved	not used
1	Reserved	not used
2	Reserved	not used
3	Reserved	not used
4	Reserved	not used
5	Input power mode	0 = the drive turns on when the input commutes
		1 = the drive turns on when the input is high
6	abilita inversione input	0 =
		1 =
7	Measure of position (enable)	0 = measure of position is not enabled
		1 = measure of position is enabled
8	Reserved	not used
9	Measure of position (mode)	0 = continuous. The position is measured every time that the input switch from 0 V to 24 V. See paragraphs 4.8.6, 4.8.7 and 4.8.8
		1 = single. The position is measured only once, to make another measure you must switch off and switch on the bit that enable the measure of positions. See paragraphs 4.8.6, 4.8.7 and 4.8.8
10..15	Reserved	not used

3.08.6 Registers 041Dh (1053 dec) and 041Eh (1054 dec) - Initial Position

Address	Name	Type	Range	Default	Unit	Read	Write
041Dh (MSB) **041Eh (LSB)**	**Initial position**	INT32	± 7FFFFFFFh			Always	No

These registers are used only if the position measure is enabled by the Bit 7 of register 0414h (*"3.08.5 Register 0414h (1044 dec) Digital Input (flag)" pag. 36)*. When one digital input is set on "***pos. measure In x***", we assume that it's connected with a sensor or a photo cell used to measure the length of an object moving nearby. When the object arrives in front of the sensor the latter sends a signal to the digital input that switch from 0 to 1. The position measured on the rising edge of this input is saved on registers 041Dh and 041Eh as the "***Initial position***".

The measured initial position is then converted from the Drive's unit of measure to the Caliper's using the "***Position factor numerator***" (0609h, 060Ah), "***Position factor denominator***" (060Bh,060Ch) and "***Position Home offset***" (060Dh,060Eh), following the formula below:

$$initial\_position = \frac{pos\_factor\_denominator * drive\_initial\_position}{pos\_factor\_numerator} + pos\_home\_offset$$

If "***Measure of position mode***" is set on "***continuous***" (bit 9 del registro 0414h = 0) then the initial position is updated continuously at every rising edge coming to the digital input.

If "***Measure of position mode***" is set on "***single***" instead (bit 9 del registro 0414h = 1) then the initial position is measured only once at the first rising edge coming in the digital input and will not be updated, unless the measure of position is disabled and then enabled again.

3.08.7 Registers 041Fh (1055 dec) and 0420h (1056 dec) - Final position

Address	Name	Type	Range	Default	Unit	Read	Write
041Fh (MSB) **0420h (LSB)**	**Final position**	INT32	± 7FFFFFFFh			Always	No

These registers are used only if the position measure is enabled by the Bit 7 of register 0414h (*"3.08.5 Register 0414h (1044 dec) Digital Input (flag)" pag. 36).* When one digital input is set on "***pos. measure In x***", we assume that it's connected with a sensor or a photo cell used to measure the length of an object moving nearby. When the object has passed beyond the sight of the sensor the latter sends a signal to the digital input that switch from 1 to 0. The position measured on the falling edge of this input is saved on registers 041Fh and 0420h as the "***Final position***".

The measured final position is then converted from the Drive's unit of measure to the Caliper's using the "***Position factor numerator***" (0609h, 060Ah), "***Position factor denominator***" (060Bh,060Ch) and "***Position Home offset***" (060Dh,060Eh), following the formula below:

$$final\_position = \frac{pos\_factor\_denominator * drive\_final\_position}{pos\_factor\_numerator} + pos\_home\_offset$$

If "***Measure of position mode***" is set on "***continuous***" (bit 9 del registro 0414h = 0) then the final position is updated continuously at every falling edge coming to the digital input.

If "***Measure of position mode***" is set on "***single***" instead (bit 9 del registro 0414h = 1) then the final position is measured only once at the first falling edge coming in the digital input and will not be updated, unless the measure of position is disabled and then enabled again.

3.08.8 Registers 0421h (1057 dec) and 0422h (1058 dec) - Delta position

Address	Name	Type	Range	Default	Unit	Read	Write
0421h (MSB) **0422h (LSB)**	**Delta Position**	INT32	± 7FFFFFFFh			Always	No

These registers are used only if the position measure is enabled by the Bit 7 of register 0414h (*"3.08.5 Register 0414h (1044 dec) Digital Input (flag)" pag. 36).* They contain the value of the difference between the "***final position***" (041Dh,041Eh) and the "***initial position***" (041Fh,0420h), which corresponds to the length of the object that should be measured (see *"3.08.6 Registers 041Dh (1053 dec) and 041Eh (1054 dec) - Initial Position" pag. 37* and *"3.08.7 Registers 041Fh (1055 dec) and 0420h (1056 dec) - Final position" pag. 37*).

The measured value is then converted from the Drive's unit of measure to the Caliper's using the "***Position factor numerator***" (0609h, 060Ah), "***Position factor denominator***" (060Bh,060Ch) and "***Position Home offset***" (060Dh,060Eh), following the formula below:

$$delta\_position = \frac{pos\_factor\_denominator * drive\_delta\_position}{pos\_factor\_numerator} + pos\_home\_offset$$

If "***Measure of position mode***" is set on "***continuous***" (bit 9 del registro 0414h = 0) then the delta position is updated every time that the final position is updated.

If "***Measure of position mode***" is set on "***single***" instead (bit 9 del registro 0414h = 1) then the delta position is measured only once at the first measure of th final position and will not be updated, unless the measure of position is disabled and then enabled again.

3.08.9 Registers 0423h (1059 dec) ... 042Ah (1066 dec) andRegisters 0448h (1096 dec) ... 044Ah (1098 dec) - Digital input filters

Indirizzo	Nome	Tipo	Range	Default	Unit	Read	Write
0423h	**Filter time input 0**	UNT16	0 ... 6000	0	ms/10	Always	Always
0424h	**Filter time input 1**						
0425h	**Filter time input 2**						
0426h	**Filter time input 3**						
0427h	**Filter time input 4**						
0428h	**Filter time input 5**						
0429h	**Filter time input 6**						
042Ah	**Filter time input 7**						
0448h	**Filter time input 8**						
0449h	**Filter time input 9**						
044Ah	**Filter time input 10**						

All the digital inputs have their own filter, which is used to filter out all the external disturbances from the signals. These filters are regulated using using a single parameter , "***Filter time input ...***" , which is set in one of the registers reported above (there is one for every filter,ie for every input). the value of these registers are used as "***reference points***": if an input receives a signal with a different logic value it will not switch istantly, instead it will wait a period of time as long as the time assigned to its filter. If the signal last longer, the input switch consequently, otherwise the signal will be ignored and considered as a disturbance.

3.08.10 Registers 042Bh (1067 dec) ... 0433h (1075 dec) and 044Bh (1053 dec) ... 044Dh (1054 dec) - Digital input

Address	Name	Type	Range	Default	Unit	Read	Write
042Bh	**Digital In. I.0**	UNT16	0 ... 32	Null-off		Always	switch off
042Ch	**Digital In. I.1**			Null-off			
042Dh	**Digital In. I.2**			Limit switch CW			
042Eh	**Digital In. I.3**			Limit switch CCW			
042Fh	**Digital In. I.4**			Home switch			
0430h	**Digital In. I.5**			Null-off			
0431h	**Digital In.I.6**			Null-off			
0432h	**Digital In. I.7**			Null-off			
044Bh	**Digital In. I.8 (NTT)**			Null-off			
044Ch	**Digital In. I.9 (NTT)**			Null-off			
044Dh	**Digital In. I.10 (NTT)**			Null-off			

The registers from 42Bh to 432h and from 44Bh and 44Dh are used to assign at every input a specific function, selectable from the menu "***Modbus - digital input***". The description of these functions is reported in the table below:

Name	Description
[00] = Null-Off	No command selected
[01] = Switch On	when this command is activated the power is turned on, but the reference signal for the control remains to 0. (ex. if the speed control is selected, with the ***Switch On*** the motor is powered, but the speed reference remains to 0, so the motor will excert a torque to hold its position). On the drive's display is showed the word "*Enable*"
[02] = Enable Operation	when this command is activated the motor receives the reference signal, which will change depending on the selected control type (it could be a current, a speed ...). The motor switchs from "***Switch on***" state to "***Operation enabled***", and starts following the reference. On the drive's display is showed the word "*Run*"
[03] = Fault reset	when this command is activated all drive's alarms are reset
[04] = Power/Operat.	Combined command of ***Switch On*** + ***Enable Operation.*** On the drive's display is showed the word "*Run*"
[05] = Power/Operat./ Reset	Combined command of ***Reset***+ ***Switch On*** + ***Enable Operation***
[06] = Halt	Halt command. The reference signal is stopped while the Drive was in "***Operation Enabled***", so the state change to "***Switch on***". It's possible to select how the halt stop will occur (sudden stop or with a ramp)
[07] = Limit switch CW	the input is abilitated for the function of limit switch CW (clockwise). Whenever this input is activated the motor stops.
[08] = Limit switch CCW	the input is abilitated for the function of limit switch CCW (counterclockwise). Whenever this input is activated the motor stops.
[09] = Home switch	The input is assigned to the home sensor
[10] = Start Reference	This command abilitate the creation of the reference signal (current, speed, pos, ecc...)
[11] = Position Abs./Rel.	This command is used to convert the position from absolute to relative and viceversa
[12] = Home position	This command starts the homing position procedure
[13] = Jog +	Command of positive speed jog
[14] = Jog -	Command of negative speed jog
[15] = Rif Select-In.0	input used to select the positioner quote using the Modbus
[16] = Rif Select-In.1	input used to select the positioner quote using the Modbus
[17] = Rif Select-In.2	input used to select the positioner quote using the Modbus

Name	Description
[18] = Rif Select-In.3	input used to select the positioner quote using the Modbus
[19] = Rif Select-In.4	input used to select the positioner quote using the Modbus
[20] = Rif Select-In.5	input used to select the positioner quote using the Modbus
[21] = Reset Index	the index of the quote of the cyclic positioner gets resetted
[22] = Pos. on Input X	this command stops the motor on input X signaled position + the relative position setted
[23] = Meas. Pos. Input X	This input is used to measure the position
[24] = Cam Select-In. 0	input used to select the cam between the 8 avaiable
[25] = Cam Select-In. 1	input used to select the cam between the 8 avaiable
[26] = Cam Select-In. 2	input used to select the cam between the 8 avaiable
[27] = Input Sync Cam	Input used to receive the synchronization signal for the electronic cam
[28] = Input Sync Slave	input used to receive the synchronization signal for the slave module
[29] = Pos. phase shift	input used to enable the position shift of the axis
[30] = Vel shift +	input used to enable the positive speed shift of the axis
[31] = Vel shift -	input used to enable the negative speed shift of the axis
[32] = Select/Start	input used to select the positioner quotes and to start the positioner "***Input-start***"

Address	Name	Type	Range	Default	Unit	Read	Write
0433h	**Digital input flag**	UINT16	0...65535	Attivo alto		Always	Switch off

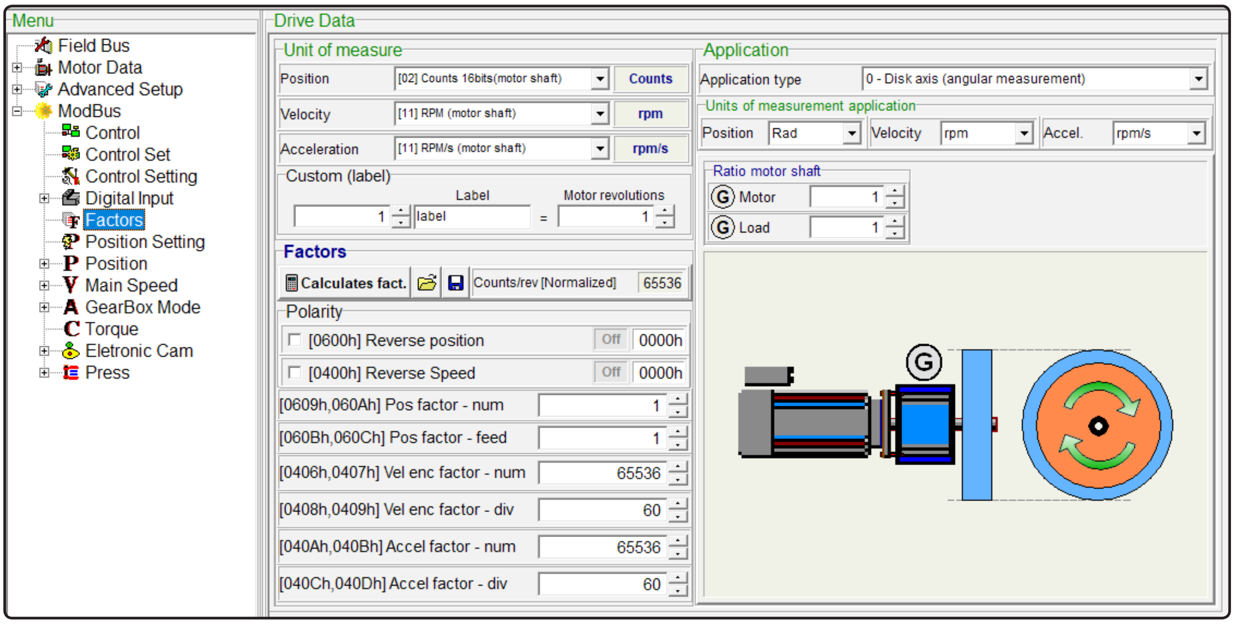
The register 433h instead contains 11 bits, one for each digital input. This bits are used to select if the assigned input must be considered active when it's equal to 1 ("***active high***") or when it's equal to 0 ("***active low***").

3.09 Conversion factors

This paragraph lists the conversion factors used by the drive to convert the positions, speeds, and accelerations variables. These factors are used only when the control type is set on ***Speed*** or ***Positioner***

Address	Name	Type	Read	Write
0400h	Modbus speed flag	UINT16	always	always
0406h	Speed factor numerator (MSB)	UINT32	always	always
0407h	Speed factor numerator (LSB)			
0408h	Speed factor denominator (MSB)	UINT32	always	always
0409h	Speed factor denominator (LSB)			
040Ah	Acc, factor numerator (MSB)	UINT32	always	always
040Bh	Acc, factor numerator(LSB)			
040Ch	Acc, factor denominator (MSB)	UINT32	always	always
040Dh	Acc, factor denominator (LSB)			
0600h	Modbus position flag	UINT16	always	always
0609h	Pos. Factor numerator (MSB)	UINT32	always	always
060Ah	Pos. Factor numerator (LSB)			
060Bh	Pos. Factor denominator (MSB)	UINT32	always	always
060Ch	Pos. Factor denominator (LSB)			

these variables can be set in the Caliper menu "***Modbus- Factors***" in which the following screen appears:



3.09.1 Register 0400h (1024 dec) Modbus flag reverse speed

Address	Name	Type	Range	Default	Unit	Read	Write
0400h	**Modbus flag reverse speed**	UINT16	0..1	0	Bit	Always	Always

Bit	Name	Description
0	reverse speed (speed polarity)	Setting this bit to 1, using the same speed reference the direction of rotation is reversed
1... 16	reserved	

3.09.2 Registers 0406h (1030 dec) ,0407h (1031 dec) - Speed factor numerator

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0406h (MSB) 0407h (LSB)	**Speed factor numerator**	UINT32	0..FFFFFFFFh	65536 (dec)		Always	Always

The ***speed factor numerator*** works together with the ***speed factor denominator.***

The setted reference speed is multiplied with the "***Speed factor Numerator***" and then is divided by the "***speed factor denominator***" in order to convert it to the internal unit of measure of the Drive (increments per second):

$$drive\_reference\_speed = \frac{(SpeedfactorNumerator \times Reference\_speed)}{(SpeedfactorDenominator)}$$

Considering that the drive divides a single turn in 65536 counts, using the default values of the factors, ie *Numerator* = 65536 and and *Denominator* = 60, the reference speed is expressed in [rpm].

3.09.3 Registers 0408h (1032 dec), 0409h (1033 dec) Speed factor denominator

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0408h (MSB) **0409h (LSB)**	**Speed factor denominator**	UINT16	1..FFFFFFFFh	60 (dec)		Always	Always

Look in the previous paragraph for the description.

3.09.4 Registers 040Ah (1034 dec), 040Bh (1035 dec) Acceleration factor numerator

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
040Ah (MSB) **040Bh (LSB)**	**Acceleration factor numerator**	UINT32	1..FFFFFFFFh	65536 (dec)		Always	Always

The ***Acceleration factor numerator*** works together with the Acceleration ***factor denominator.***

All the accelerations and decelerations values are multiplied with the "***Acceleration factor Numerator***" and then are divided by the "***Acceleration factor denominator***" in order to convert them to the internal unit of measure of the Drive (increments per second squared):

$$drive\_acceleration = \frac{(AccelerationfactorNumerator \times Acceleration)}{(AccelerationfactorDenominator)}$$

Considering that the drive divides a single turn in 65536 counts, using the default values of the factors, ie *Numerator* = 65536 and and *Denominator* = 60, the acceleration is expressed in [rpm/sec].

3.09.5 Registers 040Ch (1036 dec), 040Dh (1037 dec) Acceleration factor denominator

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
040Ch (MSB) **040Dh (LSB)**	**Acceleration factor denominator**	UINT32	1..FFFFFFFFh	60 (dec)		Always	Always

Look in the previous paragraph for the description.

3.09.6 Register 0600h (1536 dec) Modbus flag position

Address	Name	Type	Range	Default	Unit	Read	Write
0600h	**Modbus flag position**	UINT16	0..7	0	Bit	Always	Always

Bit	Name	Modbus flag Position - Bit Description
0	Reverse position (pos. polarity)	Setting this bit to 1, using the same position reference the target position and the measured position are reversed. The motor rotates in the other direction.
1	Enable position limits	Setting this bit to1, the position limits set on registers 0618h, 0619h, 061Ah, 061Bh are enabled.
2	Enable torque limit	Setting this bit to 1, whenever the drive reaches the torque limit the position profile adapts to the measured position. When this approach is used, the error threshold value set on registers 040Eh, 040Fh must be very large.
3	rounding mode	0 = rounding mode "jerk". The acceleration does not change istantly, but with a ramp. We can set direclty the gradient of this ramp, ie the jerk, via Caliper
		1= rounding mode "Time". The acceleration does not change istantly, but with a ramp. We can set direclty the duration of this ramp via Caliper
4	enable positioning - input x	0= Off
		1= On
5	position type (Input). used only with bus Input/output	0 = Absolute
		1 = relative
6 ... 16	(reserved)	

3.09.7 Registers 0609h (1545 dec), 060Ah (1546 dec) - Position factor numerator

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0609h (MSB) **060Ah (LSB)**	**Position factor numerator**	UINT32	0..FFFFFFFFh	1		Always	Always

The ***position factor numerator*** works together with the ***position factor denominator.***

These registers are used to convert the target position value in the internal unit of measure of the drive. The ***Target position*** (registro 0601h 0602h) is multiplied by the ***Numerator*** and then divided by the ***Denominator:***

$$Drive\_Target\_Position = \frac{(PositionfactorNumerator \times Target\_Position)}{(Positionfactordenominator)}$$

Usually the ***Numerator*** is equal to the number of increments per turn, which is 65536.

The ***denominator*** represents instead the space covered with one turn of the axis, the value of which is expressed with the unit of measure of the ***Target position***.

The ***Measured Position***, which is contained in registers 061Ch e 061Dh, is influenced by these two factors too:

$$drive\_measured\_position = \frac{(PositionfactorNumerator \times measured\_position)}{(Positionfactordenominator)}$$

the default values of the ***Position factor numerator*** and ***position factor denominator*** is set to 1, so the Target Position and the Measured Position are expressed in increments (65536 increments = 1 turn of the motor).

3.09.8 Registers 060Bh(1547 dec), 060Ch(1548 dec) -Position factor denominator

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
060Bh (MSB) **060Ch (LSB)**	**Position factor denominator**	UINT32	1..FFFFFFFFh	60 (dec)		Always	Always

Look in the previous paragraph for the description.

3.10 Positioner

The control type ***Positioner*** is selected when the parameter "***control type***" (Register 0401h) is set to 3.

3.10.1 Control word and Status word

In this paragraph are described the structures of the *Control word* and the *Status word* when the ***Positioner*** mode is selected.

3.10.1.a Impostazione bit Control word in position mode

<table>
<tr><th>Bit</th><th colspan="2">Description</th></tr>
<tr><td>0</td><td colspan="2">Switch on</td></tr>
<tr><td>1</td><td colspan="2">Enable operation</td></tr>
<tr><td>2</td><td colspan="2">Fault reset</td></tr>
<tr><td>3</td><td colspan="2">Halt</td></tr>
<tr><td rowspan="4">4</td><td colspan="2">Start Position</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0->1</td><td>At the rising edge of this bit the target position previously set is acquired, and the motor starts to move</td></tr>
<tr><td>1->0</td><td>No effetct</td></tr>
<tr><td rowspan="4">5</td><td colspan="2">Jog -</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Jog- disabled</td></tr>
<tr><td>1</td><td>Jog- enabled: the motor starts to move with the speed setted on registers 0612h, 0613h. Note: the Jog function is excecuted only if the motor is not excecuting a profile position.</td></tr>
</table>

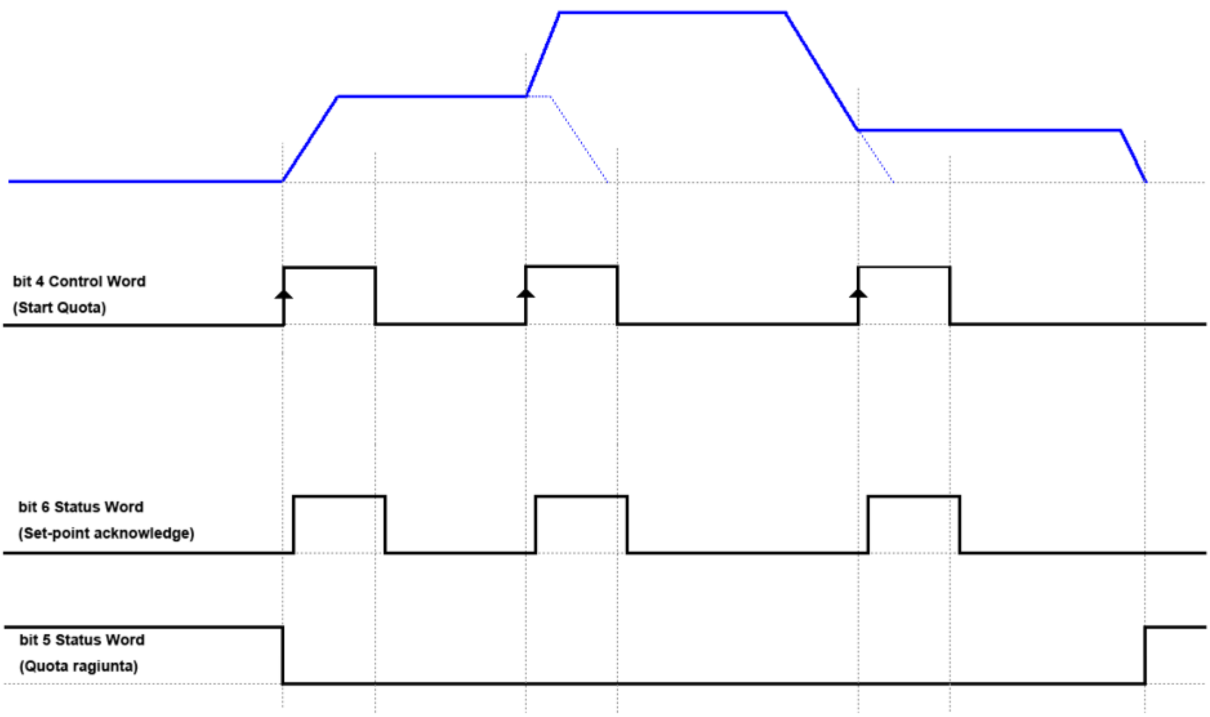
<table>
<tr><th>Bit</th><th colspan="2">Description</th></tr>
<tr><td rowspan="4">6</td><td colspan="2">Jog +</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Jog+ disabled</td></tr>
<tr><td>1</td><td>Jog+ enabled: the motor rotates in the opposite direction of Jog- with the speed setted on registers 0612h, 0613h. Note: the Jog function is excecuted only if the motor is not excecuting a profile position.</td></tr>
<tr><td rowspan="4">7</td><td colspan="2">Homing</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0->1</td><td>At the rising edge of this bit the research of the home position is enabled</td></tr>
<tr><td>0</td><td>the research of the home position is stopped</td></tr>
<tr><td rowspan="4">8</td><td colspan="2">Position absolute/relative (A/R)</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>The target position is considered as an absolute value</td></tr>
<tr><td>1</td><td>The target position is considered as a relative value of the current position</td></tr>
<tr><td>9 ... 15</td><td colspan="2">Manufacturer specific</td></tr>
</table>

3.10.1.b Description of the Status word Bits in Position mode

<table>
<tr><th>Bit</th><th colspan="2">Description</th></tr>
<tr><td>0</td><td colspan="2">switch on</td></tr>
<tr><td>1</td><td colspan="2">Enable position</td></tr>
<tr><td>2</td><td colspan="2">Stop Ramp</td></tr>
<tr><td>3</td><td colspan="2">Halt Ramp</td></tr>
<tr><td>4</td><td colspan="2">Fault</td></tr>
<tr><td rowspan="4">5</td><td colspan="2">Target reached</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>set target is not reached</td></tr>
<tr><td>1</td><td>set target reached</td></tr>
<tr><td rowspan="4">6</td><td colspan="2">Target acknowledged</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Target from the trajectory generator is not yet acquired. When the Start position bit of the control word is set to 0 this bit is set to 0 too.</td></tr>
<tr><td>1</td><td>Target from the trajectory generator is acquired.</td></tr>
<tr><td rowspan="4">7</td><td colspan="2">Homing performed</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Research of the home position not executed</td></tr>
<tr><td>1</td><td>Research of the home position executed</td></tr>
<tr><td rowspan="4">8</td><td colspan="2">Position Error</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Position error not detected</td></tr>
<tr><td>1</td><td>Position error detected</td></tr>
<tr><td rowspan="4">9</td><td colspan="2">Remote</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Remote control disabled</td></tr>
<tr><td>1</td><td>Remote control enabled</td></tr>
</table>

<table>
<tr><th>Bit</th><th colspan="2">Description</th></tr>
<tr><td rowspan="4">10</td><td colspan="2">Torque limit</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Torque limit not reached</td></tr>
<tr><td>1</td><td>Torque limit reached</td></tr>
<tr><td rowspan="4">11</td><td colspan="2">Speed Zero</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>The motor is at standstill</td></tr>
<tr><td>1</td><td>the motor is moving</td></tr>
<tr><td rowspan="4">12</td><td colspan="2">Position Limit</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Position limit not found</td></tr>
<tr><td>1</td><td>The motor has reached a position limit setted by software. The motor stops and hold the torque. If the motor reaches a limit switch then it stops, but Position limit Bit is not rised.</td></tr>
<tr><td>13</td><td colspan="2">Warning</td></tr>
<tr><td>14</td><td colspan="2">Fault reaction</td></tr>
<tr><td>15</td><td colspan="2">Speed reached</td></tr>
</table>

On the rising edge of bit 4 (***Start Position***) of the *control word* the position's profile is executed and the Bit 6 (***Target acknowledged***) of the *status word* is rised to 1 to indicate that the setpoint was acquired. When the bit 4 of the *control word* returns to 0, the bit 6 of the *status word* returns to 0 too, to indicate that the drive is ready to get and reach another setpoint even if it hasn't reached the last one yet. The Bit 5 (***Target Reached***) of the *status word* is raised to 1 when the last of the setpoints is reached.

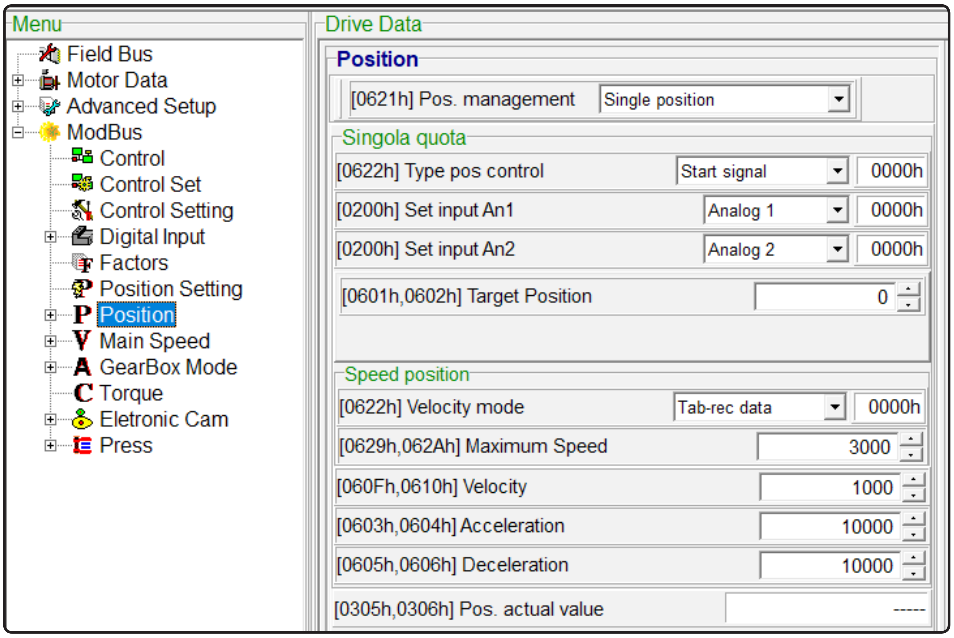


3.10.2 Position Profile parameters

This paragraph lists the parameters used to generate the trajectory of the position profile:

Address	Name	Type	Read	Write
0314h	Index select	UINT16	always	No
0315h	Setpoint number	UINT16	always	No
0601h	Ref. position (MSB)	INT32	always	always
0602h	Ref. position (LSB)			
0603h	Pos. - acceleration (MSB)	UINT32	always	always
0604h	Pos. - acceleration (LSB)			
0605h	Pos. - deceleration (MSB)	UINT32	always	always
0606h	Pos. - deceleration (LSB)			
060Fh	Pos. - speed (MSB)	UINT32	always	always
0610h	Pos. - speed (LSB)			
061Ch	Measured pos. (MSB)	INT32	always	No
061Dh	Measured pos. (LSB)			
0621h	Pos. management	INT16	always	switch off
0622h	Flag controllo pos.	INT16	always	switch off
0623h	Index	INT16	always	always
0624h	Max cycles	INT16	always	always
0625h	Pos. analog. max (MSB)	INT16	always	always
0626h	Pos. analog. max (LSB)			
0627h	Pos. analog. min (MSB)	INT16	always	always
0628h	Pos. analog. min (LSB)			
0629h	Maximum speed (MSB)	UINT32	always	always
062Ah	Maximum speed (LSB)			

these variables can be set in the Caliper menu "***Modbus-Position***" in which the following screen appears:



3.10.2.a Register 0314h (788 dec) - Selection index

Address	Name	Type	Range	Default	Unit	Read	Write
0314h	**selection index**	UINT16	0... 65535			Always	No

This register at the moment is not used.

3.10.2.b Register 0315h (789 dec) - Pos. number

Address	Name	Type	Range	Default	Unit	Read	Write
0315h	**Pos. number**	UINT16	0... 65535			Always	No

This register contains the index of the position, along the ones listed in the position table, that is the setpoint of the drive at the moment.

3.10.2.c Registers 0601h (1537 dec) and 0602h (1538 dec) - Target Position

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0601h (MSB) **0602h (LSB)**	**Target Position**	INT32	±7FFFFFFFh	0		Always	Always

Target Position indicates the setpoint that the drive must reach. This value is converted in the internal unit of measure of the drive using the ***Position factors*** (registers 609h 60Ah; 60Bh 60Ch). The formula for the conversion is showed below:

$$Drive\_Target\_Position = \frac{(PositionfactorNumerator \times Target\_Position)}{(Positionfactordenominator)}$$

Note: One turn of the motor axis corresponds to 65536 internal increments.

3.10.2.d Registers 0603h (1539 dec) and 0604h (1540 dec) - Position -Acceleration

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0603h (MSB) **0604h (LSB)**	**Position -** **Acceleration**	UINT32	0..,5DFFA2h	10000 (dec)		Always	Always

These parameters contain the value of the acceleration reached while the motor moves to the target position. This value is converted in internal unit of the drive (increments per second squared [increment/s^2]) using the ***Acceleration factor numerator*** (registri 040Ah, 040Bh) and the ***Acceleration factor denominator*** (registri 040Ch, 040Dh). The formula for the conversion is showed below:

$$Drive\_acceleration = \frac{(AccelerationfactorNumerator \times acceleration)}{(Accelerationfactordenominator)}$$

Using the default values of the ***Acceleration factors***, ie Numerator=65536 and Denominator=60, the ***Position-acceleration*** is expressed in [rpm/s]

Note: one turn of the motor correspond to 65536 internal increments.

3.10.2.e Registers 0605h (1541 dec) and 0606h (1542 dec) Position- Deceleration

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0605h (MSB) **0606h (LSB)**	**Position - Deceleration**	UINT32	1... 7FFFFFFFh	10000 (dec)		Always	Always

These registers contain the value of deceleration reached while the motor moves to the target position. This value is converted in internal unit of the drive (increments per second squared [increment/s^2]) using the ***Acceleration factor numerator*** (registri 040Ah, 040Bh) and the ***Acceleration factor denominator*** (registri 040Ch, 040Dh). The formula for the conversion is showed in the previous paragraph. Using the default values of the ***Acceleration factors***, ie Numerator=65536 and Denominator=60, the ***Position deceleration*** is expressed in [rpm/s].

Note: one turn of the motor correspond to 65536 internal increments.

3.10.2.f Registers 060Fh (1551 dec) and 0610h (1552 dec) Position-speed

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
060Fh (MSB) **0610h (LSB)**	**Position- speed**	UINT32	1... 7FFFFFFFh	1000 (dec)		Always	Always

These registers contain the maximum absolute value of speed tha could be reached while the motor moves to the target position. This value is converted to the internal unit of the drive (increments per second [increments/s]) using the conversion factors ***Speed Factor Numerator*** (registers 0406h, 0407h) and ***Speed Factor Denominator*** (registers 0408h e 0409h).The formula for the conversion is showed below:

$$drive\_reference\_speed = \frac{(SpeedfactorNumerator \times Reference\_speed)}{(SpeedfactorDenominator)}$$

Using the default values of the ***Speed factors***, ie Numerator=65536 and Denominator=60, the ***Position-speed*** is expressed in [rpm/s].

Note: one turn of the motor correspond to 65536 internal increments.

3.10.2.g Registers 061Ch (1564 dec) and 061D (1565 dec) measured position

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
061Ch (MSB) **061Dh (LSB)**	**Measured position**	INT32	±7FFFFFFFh			Always	No

These registers contains the ***Measured position*** of the motor. This value is reset at the Drive turn on and after the homing procedure. This value is obtained using the position measured by the sensor, the position factor numerator, the position factor denominator and the ***Home offset*** (060Dh, 060Eh) as reported in the formula below:

$$Measured\_position = \frac{(PositionfactorDenominator \times Sensor\_position)}{(Positionfactornumerator)} - Home\_Offset$$

3.10.2.h Register 0621h (1569 dec) - Pos. management

Address	Name	Type	Range	Default	Unit	Read	Write
0621h	**Pos. management**	INT16	0... 4	single position		Always	Switch off

Type	Description
single position Analogic position	In this mode the drive generates a position trajectory for only one target, using the values set of speed, acceleration, deceleration and jerk. the position could be absolute or relative. For the other fieldbuses these parameters can be set exclusively using a telegram; the Modbus RTU instead allows you to use both the Modbus commands and the analogical/digital inputs. If a fieldbus is not available, position and speed can be set sending Analogic signals to the appropriate inputs, while the other parameters can be set using the Caliper.
Pos - table input selection	In this mode a maximum of 64 target positions can be stored. Like in the ***Single position Mode***, for every setpoint stored the values of speed, acceleration, deceleration and jerk can be set. the positions could be absolute or relative. The target positions are stored in a table of the drive using the Caliper or the fieldbus. These target positions can be reached individually or they can be concatenated to obtain a more complex trajectory.
Pos-table cyclic	This mode is similar to the ***Pos-table input section Mode***, with the difference that the target positions are all rigidly reached one after the other. This positions can be enabled manually using the Caliper or with the Modbus RTU. It's possible to choose if the sequence of targets has to be cyclic (the sequence is repeated in a never-ending loop) or acyclic (the sequence is repeated for a finite number of cycles)
Pos-table input start	This mode allows you to synchronize one axis start with the arrival of another axis to its target position, without using a PLC. This mode differs from the previous one because the input that choose the target point (or the sequence of target points) also starts the execution of the position trajectory. The signal of "***Target reached***" can be sent by all the output of the drive. Therefore, the connection between the output of one drive to the input of another drive permits the synchronized start of the latter. This mode is allowed with digital/analogic inputs or with the fieldbus Modbus.

3.10.2.i Register 0622h (1570 dec) - Positioner - flag

Address	Name	Type	Range	Default	Unit	Read	Write
0622h	**Positioner - flag**	INT16	0 ... 65535			Always	Switch off

This register contain a series of flags used by the drive when the control type is set on "***Positioner***". They are listed in the table below:

<table>
<tr><th>Bit</th><th colspan="2">Description</th></tr>
<tr><td rowspan="4">0</td><th colspan="2">Mode</th></tr>
<tr><th>value</th><th>description</th></tr>
<tr><td>0</td><td>manual cycle</td></tr>
<tr><td>1</td><td>automatic cycle</td></tr>
<tr><td rowspan="4">1</td><th colspan="2">Cycle</th></tr>
<tr><th>value</th><th>description</th></tr>
<tr><td>0</td><td>cyclic</td></tr>
<tr><td>1</td><td>acyclic</td></tr>
<tr><td rowspan="4">2</td><th colspan="2">speed mode</th></tr>
<tr><th>value</th><th>description</th></tr>
<tr><td>0</td><td>tab-rec data</td></tr>
<tr><td>1</td><td>from analog 1</td></tr>
</table>

<table>
<tr><th>Bit</th><th colspan="2">Description</th></tr>
<tr><td rowspan="4">3</td><td colspan="2">Type pos control</td></tr>
<tr><td>value</td><td>description</td></tr>
<tr><td>0</td><td>start signal</td></tr>
<tr><td>1</td><td>continuos</td></tr>
<tr><td>4 ... 15</td><td colspan="2">reserved speed</td></tr>
</table>

3.10.2.j Register 0623h (1571 dec) - Index

Address	Name	Type	Range	Default	Unit	Read	Write
0623h	**Index**	UINT16	2...64	2		Always	Always

This register is used when the positioner is set on ***Pos-table cyclic mode.*** It's used to set the index of the last setpoint position to reach; the positioner will reach all the targets of the table until it arrives to the last.

3.10.2.k Register 0624h (1572 dec) - Max cycles

Address	Name	Type	Range	Default	Unit	Read	Write
0624h	**Max cycles**	UINT16	1...65530	1		Always	Always

This register is used when the positioner is set on ***Pos-table cyclic mode***. It's used to set the number of times that the sequence of position trajectories must be repeated.

3.10.2.l Registers 0625h (1573 dec) and 0626h (1574 dec) - Pos. Analog. max

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0625h (MSB) **0626h (LSB)**	**Pos. analog. max**	INT32	±7FFFFFFFh	65536		Always	Always

These registers contain the "***Maximum analogic position***". When the positioner is set on ***Analogic position*** the measured position is read by the analogic signal entering on the auxiliary analogic input 1. This value must remain inside the range delimited by the "***Maximum analogic position***" and the "***Minimum analogic position***" set.

3.10.2.m Registers 0627h (1575 dec), 0628h (1576 dec) - Pos. Analog. min

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0627h (MSB) **0628h (LSB)**	**Pos. analog. min**	INT32	±7FFFFFFFh	0		Always	Always

These registers contain the "***Minimum analogic position***". When the positioner is set on ***Analogic position*** the measured position is read by the analogic signal entering on the auxiliary analogic input 1. This value must remain inside the range delimited by the "***Maximum analogic position***" and the "***Minimum analogic position***" set.

3.10.2.n Registers 0629h (1577 dec), 062Ah (1578 dec) - Full scale analogic speed

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0629h (MSB) **062Ah (LSB)**	**Full scale analogic speed**	UINT32	1... 7FFFFFFFh	0		Always	Always

These registers contain the ***full scale of the analogic speed***. The measured speed is read from the main analogic input, and then is converted in an analogic value which cannot exceed the full scale set in this registers.

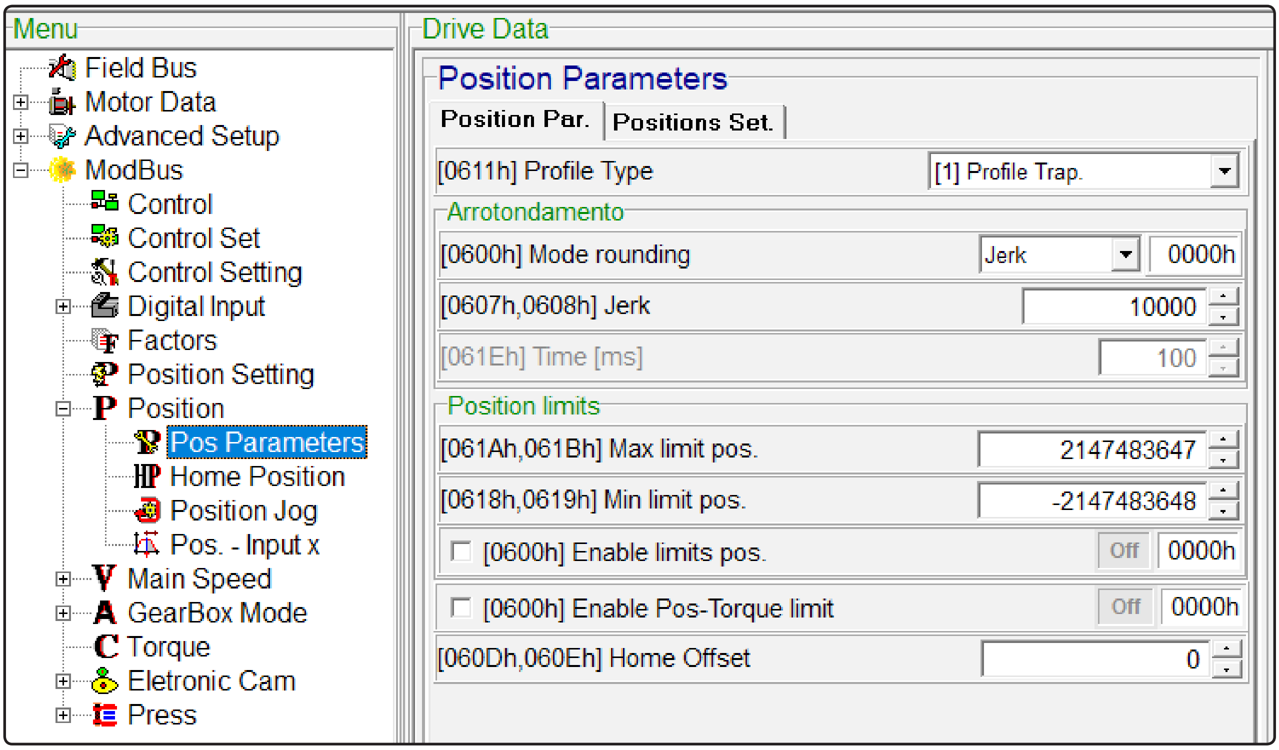
3.10.3 Position parameters

This paragraph lists the parameters used to control the Drive when it's working on ***Positioner*** mode:

Address	Name	Type	Read	Write
0600h	Modbus positioner flag	UINT16	always	always
0607h	Jerk (MSB)	UINT32	always	always
0608h	Jerk (LSB)			
060Dh	Home offset (MSB)	INT16	always	Switch on
060Eh	Home offset (LSB)			
0611h	Position type	INT16	always	Switch on
0618h	Min Position Limit (MSB)	INT32	always	always
0619h	Min Position Limit (LSB)			
061Ah	Max Position Limit (MSB)	INT32	always	always
061Bh	Max Position Limit (LSB)			
061Eh	Ramp Time	INT16	always	always
062Bh	Position recovery	INT16	always	always
062Ch	End position mode	INT16	always	always
062Dh	Threshold target reached (MSB)	UINT32	always	always
062Eh	Threshold target reached (LSB)			
0631h	Time target reached	UINT16	always	always

-

these variables can be set in the Caliper menu "***Position-Pos. parameters***" in which the following screen appears:



3.10.3.a Registro 0600h (1536 dec) bit 1 e 2 - Modbus positioner flag

Address	Name	Type	Range	Default	Unit	Read	Write
0600h	**Modbus position flag**	UINT16	0..7	0	Bit	Always	Always

This register was already described in the paragraph "***Conversion factors***". From the menu "***positioner parameters***" we can access only to the bit 1 and 2, that are used as flags.

Bit	Name	Modbus position flag -Bit description
1	Enable position limits	When this bit is set on 1 all the position limits set on registers 0618h, 0619h, 061Ah and 061Bh are enabled. If the rotor is moving to a position that is beyond the range determined by these limits, it will be stopped once it reached one of them and the bit 12 of the *Status word* will be raised to notify the event.
2	Enable Torque limit	When this bit is set on 1, if the drive reaches the torque limit the position trajectory calculated will adapt to the measured position. In practice, when the motor is blocked the algorithm that generate the position trajectory stops, but it still applies a torque (not beyond the ***Current Limit*** set on 0207h) because it's trying to finish its task. Once the rotor is free to move the position trajectory will be continued in order to reach the target position. In this mode the position error is neglected since the reference position is set equal to the measured position. However, it's suggested to set the ***Error Threshold*** (registers 040Eh, 040Fh) to a higher value to be sure that no alarm occurs once the current limit is reached.

3.10.3.b Register 0607h (1543 dec) 0608h (1544 dec) Position Jerk

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0607h (MSB) **0608h (LSB)**	**Position Jerk**	UINT32	1 ... 7FFFFFFFh	10000 (dec)		Always	Always

The ***Position jerk*** is used when ***Position Type*** (register 0611h) is set on "***S profile"*** and the parameter "***Rounding mode***" (register 0600h) is set on "***jerk***". When "***S profile***" is enabled the position trajectory is calculated in such a way that there are no sudden variations of the acceleration. in this case the acceleration changes with a linear ramp, the gradient of which is equal to the jerk set on these registers.

The unit of measure of this value is "acceleration divided by time"; inside the drive it's expressed as [increments/s^3]. This value is converted in the internal unit of measure of the drive using the ***Acceleration factor numerator*** (registers 040Ah, 040Bh) and the ***Acceleration factor denominator*** (registers 040Ch, 040Dh) as showed in the formula below:

$$Drive\_position\_jerk = \frac{(AccelerationFactorNumerator \times Position\_Jerk)}{(AccelerationFactorDenominator)}$$

Using the default value of this factors, i.e. *Numerator*=65536 and *Denominator*=60, the jerk is expressed on the Caliper in [rpm/s^2].

3.10.3.c Registers 060Dh (1549 dec), 060Eh (1550 dec) - Position Home offset

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
060Dh (MSB) **060Eh (LSB)**	**Position home offset**	INT32	0 ... ±7FFFFFFFh	0		Always	Always

These registers contain the ***Home offset***, i.e. the difference between the home position of the application and the home position of the drive found after the Homing procedure. This parameter affects the ***Measured position*** (registers 061Ch,061Dh), as shown in the formula below:

$$Measured\_Position = sensor\_position + home\_offset$$

The position value is acquired by the drive from the external encoder, it's a signed value of 32 bit. One turn of the motor corresponds to 65536 increments. That's because the position value is the sum of two terms:

- the position of the rotor expressed as a fraction of a turn, which range from 0 to 65535;
- the numer of turns made by the axis, which range from -32767 to 32767.

The measured position read from the sensor is reset to 0 at every turn on of the drive and after the research of the Home position.

You bring the value of ***Measured Position*** (registers 061Ch, 061Dh) at a certain point to 0 following these steps:

1. Put ***Home offset*** to 0;
2. Read the value of ***Measured position*** reported on registers 061Ch, 061Dh at the point of interest;
3. If the bit 0 of ***Modbus position-flag*** (register 0600h) is 0, set ***Home offset*** to the opposite of the value read in the previous step. If the bit 0 of ***Modbus position-flag*** (register 0600h) is 1, set ***Home offset*** to the same of the value read in the previous step. With this correction the sum of the ***Sensor Position*** and the ***Home Offset*** will be equal to 0.

3.10.3.d Register 0611h (1553 dec) Positioner type

Address	Name	Type	Range	Default	Unit	Read	Write
0611h	**Positioner type**	UINT16	0..,1	0		Always	Switch on

Value	Description
0	***S profile enabled .*** The position reference's acceleration doesn't change abruptly, but following a ramp which depends on the ***Position Jerk*** (Registers 0607h,0608h) or on the ***Ramp time*** (register 61Eh). Therefore the speed profile obtained is curved (hence the name ***S profile***). The mechanical stress is reduced, but it will take longer to reach the target position (the drive is less "aggressive")
1	***Trapezoidal profile enabled.*** The position reference's acceleration changes abruptly. Therefore the speed profile obtained looks like a broken line (hence the name ***Trapezoidal profile***). The mechanical stress could be high

3.10.3.e Registers 0618h (1560 dec), 0619h (1561 dec) Min. limit Position

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0618h (MSB) **0619h (LSB)**	**Min. limit** **Position**	INT32	0 ... ±7FFFFFFFh	-7FFFFFFFh		Always	Always

These registers contains the minimum limit of the absolute position used when the drive is working on ***Positioner*** mode. The unit of measure is the same of the ***Target position*** (060h,0602h) and the ***Measured position*** (061Ch, 061Dh). The position limits are enabled by switching to 1 the bit 1 of register 0600h. When the position reference, expressed with its absolute value with respect to the origin, reach one of these limits the bit 12 of the status word ("position limit") is raised.

3.10.3.f Registers 061Ah (1562 dec), 061Bh (1563 dec) Max limit Position

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
061Ah (MSB) **061Bh (LSB)**	**Max. limit** **Position**	INT32	0 ... ±7FFFFFFFh	7FFFFFFFh		Always	Always

These registers contains the maximum limit of the absolute position used when the drive is working on ***Positioner*** mode. The unit of measure is the same of the ***Target position*** (060h,0602h) and the ***Measured position*** (061Ch, 061Dh). The position limits are enabled by switching to 1 the bit 1 of register 0600h. When the position reference, expressed with its absolute value with respect to the origin, reach one of these limits the bit 12 of the status word ("position limit") is raised.

3.10.3.g Register 061Eh (1566 dec) - Ramp time

Address	Name	Type	Range	Default	Unit	Read	Write
061Eh	**Ramp time**	UINT16	1..65535	100	ms	Always	Always

This register contains the value of the "***ramp time***". When you choose to smooth the position profile by changing the acceleration with linear ramps, you can also fix one parameter between the gradient of these ramps (i.e. the jerk) or the duration of these ramps. If you choose the second option, all the ramps will last for the time period that you set in this register

3.10.3.h Register 062Bh (1579 dec) - Position Recovery

Address	Name	Type	Range	Default	Unit	Read	Write
062Bh	**Position Recovery**	INT16	1..65535	Nessuna Azione		Always	Always

In some situations it could be necessary to stop momentarily the drive while it's following a position trajectory. With this register it's possible to decide what the drive will do when it'll return in ***Operation Enabled*** state. Some of these recovery modes will work only with certain types of Positioner control.

Pos. recovery	Description
No action (Works only in ***Single/Analogic position*** mode)	**Relative pos.=** the motor restarts moving, the position trajectory generated has as target the Current position, i.e. the target that it was reaching before being stopped. it doesn't consider the movement already made, so it will make again all the relative movement planned. (for example, if the motor has to do 10 turns, but it's stopped after 3, when it restarts it will make another 10 , for a total of 13 turns)
	Absolute pos.=the motor restarts moving, the position trajectory generated has as target the Current position, i.e. the target that it was reaching before being stopped. Since the position has an absolute value, it will stop exactly in the planned position: it's the same as the ***Current Position Recovery*** with absolute position.
Wait home position	the Drive will be in alarm, it's necessary to make another Homing procedure
Current position (Works only in ***Single/Analogic position*** mode)	**Relative pos.=** the motor restarts moving, the position trajectory generated has as target the Current position, i.e. the target that it was reaching before being stopped. it considers the movement already made, so it will make only the remaining movement necessary to complete the planned movement. (for example, if the motor has to do 10 turns, but it's stopped after 3, when it restarts it will make another 7, for a total of 10 turns)
	Absolute pos.= the motor restarts moving, the position trajectory generated has as target the Current position, i.e. the target that it was reaching berfore being stopped. Since the position has an absolute value, it will stop exactly in the planned position
Next position (Works only in ***Single/Analogic position*** mode)	**Relative pos.=** the motor restarts moving, the position trajectory generated has as target the Next position, i.e. the target after the one that it was reaching before being stopped. It considers the movement already made, so it will make the remaining movement necessary to reach the target position and then it will move to reach the next target. (for example, if the motor has to do 10 turns first and then another 5, but it's stopped after 3, when it restarts it will make another 7 to reach the first target and then the remaining 5 of the next position)
	Absolute pos.= the motor restarts moving, the position trajectory generated has as target the Next position, i.e. the target after the one that it was reaching before being stopped. Since the position has an absolute value, it will stop exactly in the planned position.
Reset home position	The Home Position is reset to the actual position of the Drive

3.10.3.i Register 062Ch (1580 dec) - End position mode

Address	Name	Type	Range	Default	Unit	Read	Write
062Ch	**End position mode**	INT16		Stop immediately		Always	Always

In some situations it could be necessary to stop momentarily the drive while it's following a position trajectory. With this register it's possible to decide how the drive will stop.

Type	Description
Stop immediately	The Drive will stop abruptly, with a ramp of speed as short as possible
Position deceleration	The Drive will stop with a ramp of speed. The deceleration applied is the one set of registers 0605h and 0606h.
Ending position	The Drive will reach the target position before stopping.
Ending cycle	The Drive will conclude the cycle that is doing before stopping.

3.10.3.j Registers 062Dh (1581 dec),062Eh (1582 dec) Threshold position reached

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
062Dh (MSB) **062Eh (LSB)**	**Threshold position reached**	UINT32	0... 7FFFFFFFh	910		Always	Always

These registers contain a position value that we named "***Threshold position reached***". When the position error drops below this threshold value and stays there for a time longer than the one set in "***Time target reached***" (register 0631h), then we assume that the target position has been reached; the bit 5 of the status word is set to 1. To reset this bit to 0, the position error must rise beyond the "***Threshold position reached***" for a time longer than the one set in "***Time target reached***" .

3.10.3.k Register 0631h (1585 dec) Time target reached

Address	Name	Type	Range	Default	Unit	Read	Write
0631h	**Time target reached**	UINT16	0..10000	0		Always	Always

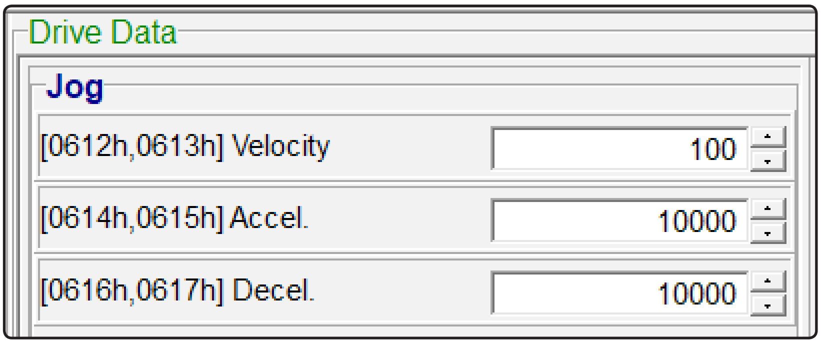
This register contains a time interval that we named "***Time target reached***". When the position error drops below the "***Threshold position reached***" (registers 062Dh,062Eh) and stays there for a time longer than the one set in this register, then we assume that the target position has been reached; the bit 5 of the status word is set to 1. To reset this bit to 0, the position error must rise beyond the "***Threshold position reached***" for a time longer than the one set in "***Time target reached***" .

3.10.4 Position Jog Parameters

This paragraph lists all the the Jog commands used with the ***Positioner*** control.

Address	Name	Type	Read	Write
0612h	Jog speed (MSB)	UINT32	always	always
0613h	Jog speed (LSB)			
0614h	Jog Acceleration (MSB)	UINT32	always	always
0615h	Jog Acceleration (LSB)			
0616h	Jog Deceleration (MSB)	UINT32	always	always
0617h	Jog Deceleration (LSB)			

these variables can be set in the Caliper menu "***Modbus-Position jog***" in which the following screen appears:



3.10.4.a Registers 0612h (1554 dec), 0613h (1555 dec) Jog speed

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0612h (MSB) **0613h (LSB)**	**Jog speed**	UINT32	1... 7FFFFFFFh	100 (dec)		Always	Always

These registers contain the absolute value of speed applied when the Drive is working in Jog mode. inside the drive it's expressed as [increments/ s]. This value is converted in the internal unit of measure of the drive using the ***Speed factor numerator*** (registers 0406h, 0407h) and the ***Speed factor denominator*** (registers 0408h, 0409h) as showed in the formula below:

$$drive\_jog\_speed = \frac{(SpeedfactorNumerator \times Jog\_speed)}{(Speedfactordenominator)}$$

Using the default values of the ***Speed factors***, ie *Numerator*=65536 and *Denominator*=60, the ***Jog speed*** is expressed in [rpm/s].

Note: one turn of the motor correspond to 65536 internal increments.

3.10.4.b Registers 0614h (1556 dec), 0615h (1557 dec) Jog Acceleration

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0614h (MSB) **0615h (LSB)**	**Jog** **Accelerazione**	UINT32	1 ... 7FFFFFFFh	10000 (dec)		Always	Always

These registers contain the absolute value of acceleration applied when the Drive is working in Jog mode. inside the drive it's expressed as [increments/s^2]. This value is converted in the internal unit of measure of the drive using the ***Acceleration factor numerator*** (registers 040Ah, 040Bh) and the ***Acceleration factor denominator*** (registers 040Ch, 040Dh) as showed in the formula below:

$$drive\_jog\_acceleration = \frac{(Acceleration factor Numerator \times Jog\_acceleration)}{(Acceleration factor denominator)}$$

Using the default values of the ***Acceleration factors***, ie *Numerator*=65536 and *Denominator*=60, the ***Jog acceleration*** is expressed in [rpm/s]

Note: one turn of the motor correspond to 65536 internal increments.

3.10.4.c Registers 0616h (1558 dec), 0617h (1559 dec) Jog Deceleration

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0616h (MSB) **0617h (LSB)**	**Jog** **decelerazione**	UINT32	1 ... 7FFFFFFFh	10000 (dec)		Always	Always

These registers contain the absolute value of deceleration applied when the Drive is working in Jog mode. inside the drive it's expressed as [increments/s^2]. This value is converted in the internal unit of measure of the drive using the ***Acceleration factor numerator*** (registers 040Ah, 040Bh) and the ***Acceleration factor denominator*** (registers 040Ch, 040Dh) as showed in the formula below:

$$drive\_jog\_deceleration = \frac{(Acceleration factor Numerator \times Jog\_deceleration)}{(Acceleration factor denominator)}$$

Using the default values of the ***Acceleration factors***, ie *Numerator*=65536 and *Denominator*=60, the ***Jog deceleration*** is expressed in [rpm/s]

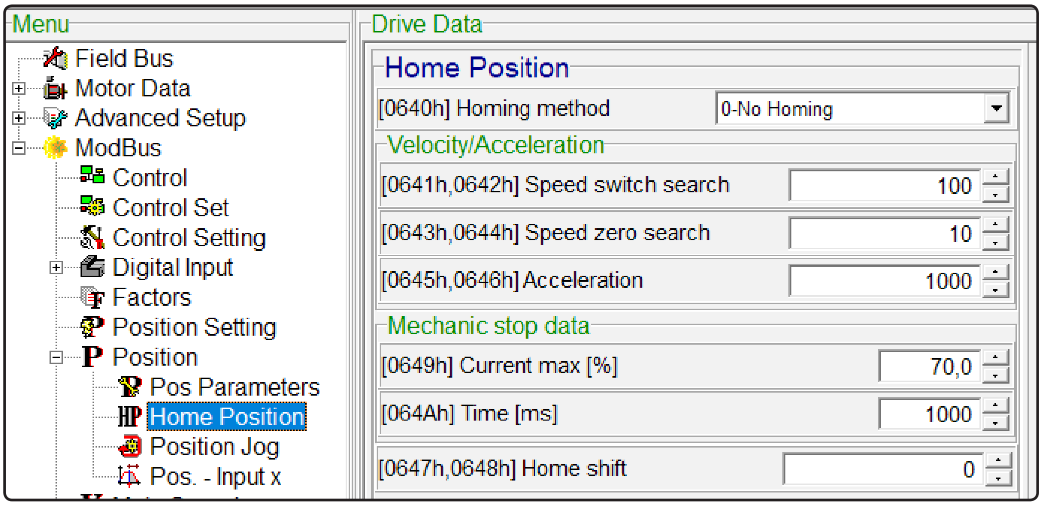
Note: one turn of the motor correspond to 65536 internal increments.

3.10.5 Home Position

This paragraph lists all the commands and modes which control the searching procedure of the position zero, i.e. the *home* of the Drive. Limit switches could be introduced, as appropriate, to costrain the searching of the home

Address	Name	Type	Read	Write
0640h	Homing method	UINT16	always	always
0641h	switch search speed (MSB)	UINT32	always	always
0642h	switch search speed (LSB)			
0643h	zero search speed (MSB)	UINT32	always	always
0644h	zero search speed (LSB)			
0645h	Homing acceleration (MSB)	UINT32	always	always
0646h	Homing acceleration (LSB)			
0647h	Home shift (MSB)	INT32	always	always
0648h	Home shift (LSB)			
0649h	mechanical stop current	UINT16	always	always
064Ah	mechanical stop time	UINT16	always	always

These variables can be set in the Caliper menu "***Position-Home position***" in which the following screen appears:



3.10.5.a Register 0640h (1600 dec) Homing method

Address	Name	Type	Range	Default	Unit	Read	Write
0640h	**Homing method**	UINT16	0..,35 eccetto 15,16,31,32	0		Always	Always

This register, which is referred as ***Homing method,*** is used to select in which way the homing procedure has to be excecuted. All the methods avaiable has an assigned number which range from 0 to 35, with the exception of the number 15,16,31 and 32 which are reserved.

3.10.5.b Method -4 - Mechanical stop CW

The drive excecutes the homing moving clockwise until it reaches the mechanical stop. When the drive assumes to have reached it (see registers 0649h,064Ah), the position reached by the motor become the homing position.

3.10.5.c Method -3 -Mechanical stop CCW

The drive excecutes the homing moving counterclockwise until it reaches the mechanical stop. When the drive assumes to have reached it (see registers 0649h,064Ah), the position reached by the motor become the homing position.

3.10.5.d Method -2 - Mechanical stop CW and index pulse

The drive excecutes the homing moving clockwise until it reaches the mechanical stop. Once the drive assumes to have reached it (see registers 0649h,064Ah), the motor changes direction and slowly moves counterclockwise until it find an index pulse of the encoder. The position reached by the motor becomes the homing position.

3.10.5.e Method -1 Mechanical stop CCW and index pulse

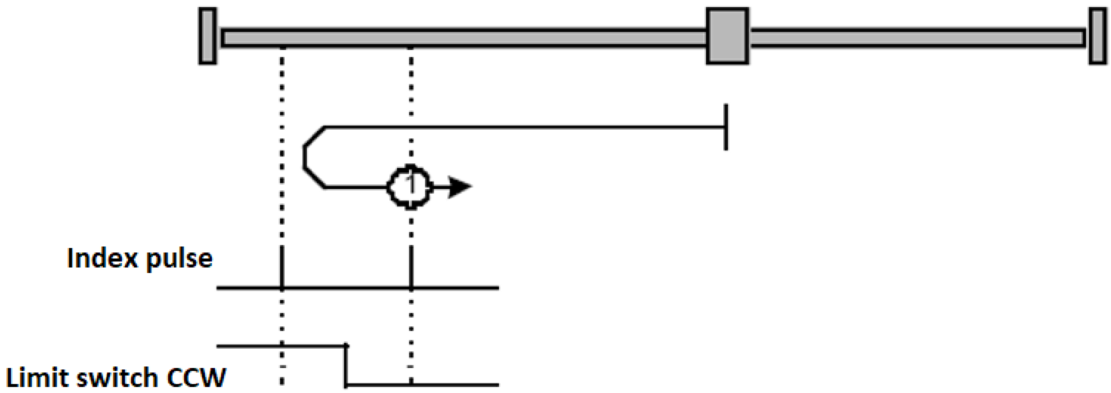
The drive excecutes the homing moving counterclockwise until it reaches the mechanical stop. Once the drive assumes to have reached it (see registers 0649h,064Ah), the motor changes direction and slowly moves clockwise until it find an index pulse of the encoder. The position reached by the motor becomes the homing position.

3.10.5.f Method 0 - No homing

When the drive is switched on the value of the measured position is reset to zero and is set as the home position of the drive.

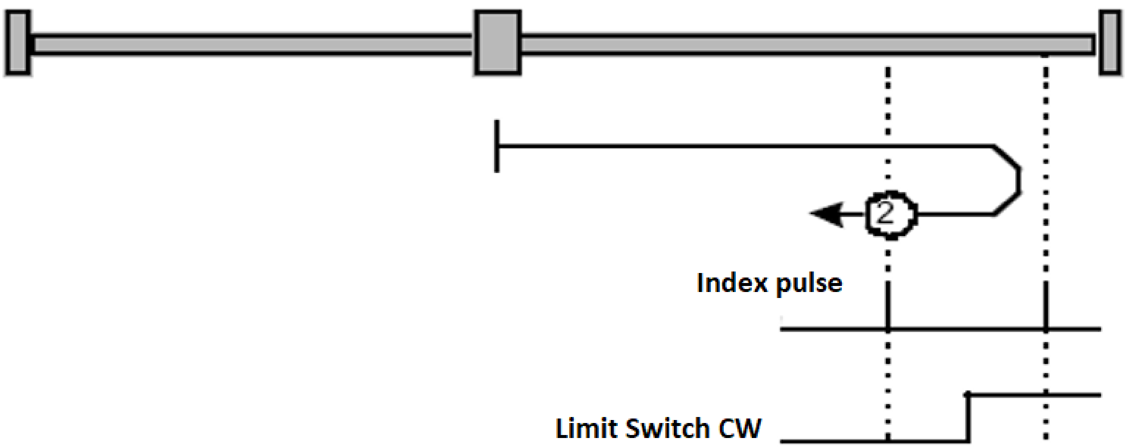
3.10.5.g Method 1 - Homing on negative limit switch and index pulse

The drive excecute the homing by moving the motor counterclockwise untill it find the limit switch sensor CCW. Once it has reached the sensor, it changes direction and moves at low speed to exit from the limit switch sensor and then it continues until it find an index pulse of the encoder, that point therefore will become the home position of the drive.



3.10.5.h Method 2 - Homing on positive limit switch and index pulse

The drive excecute the homing by moving the motor clockwise untill it find the limit switch sensor CW. Once it has reached the sensor, it changes direction and moves at low speed to exit from the limit switch sensor and then it continues until it find an index pulse of the encoder, that point therefore will become the home position of the drive.

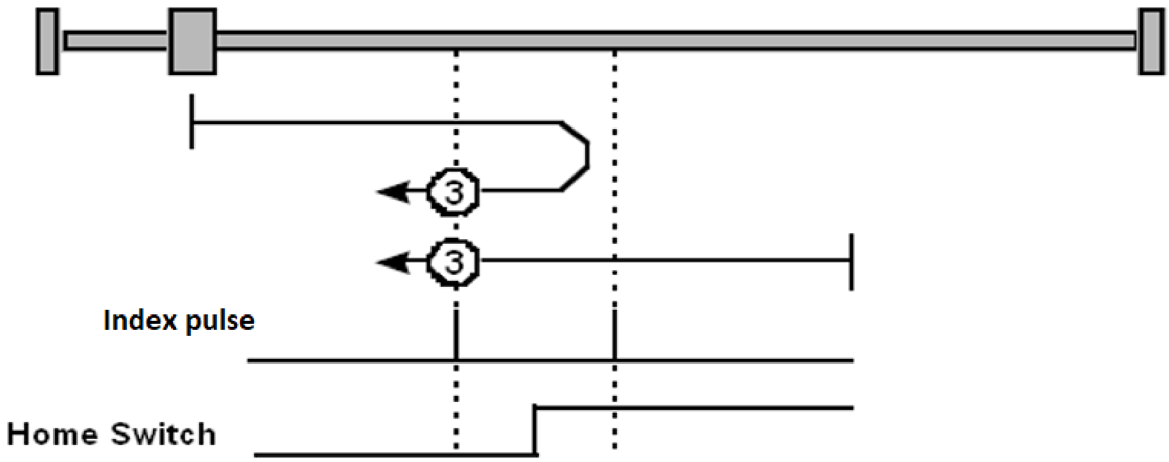


3.10.5.i Method 3 – Homing on positive home switch and index pulse

The logic value of the input assigned to the home sensor determines the direction of the search of the home sensor:

» If the assigned input is low, the motor rotates clockwise and when the commutation of the home sensor is revealed the motor stops and slowly moves counterclockwise until it finds an index pulse of the encoder, which will become the home position.

» If the assigned input is high, the motor rotates counterclockwise and when the commutation of the home sensor is revealed the motor slows down but keeps moving counterclockwise until it finds an index pulse of the encoder, which will become the home position.

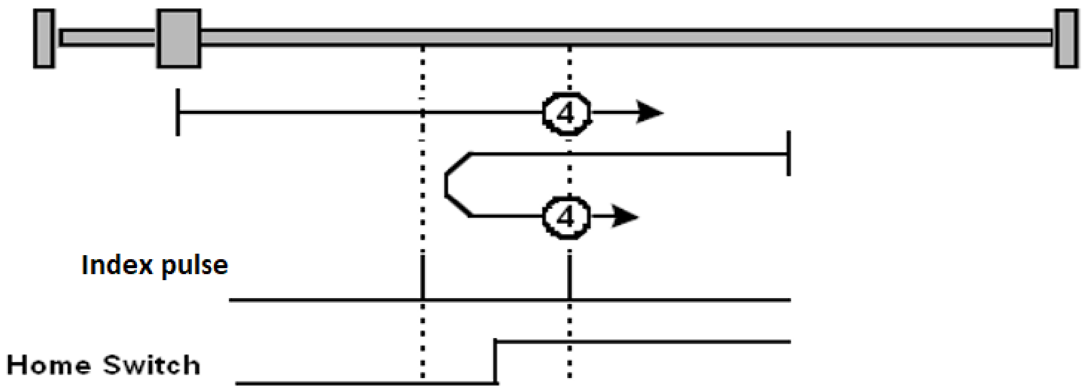


3.10.5.j Method 4 - Homing on positive home switch and index pulse

The logic value of the input assigned to the home sensor determines the direction of the search of the home sensor:

» If the assigned input is high, the motor rotates counterclockwise and when the commutation of the home sensor is revealed the motor stops and slowly moves clockwise until it finds an index pulse of the encoder, which will become the home position.

» If the assigned input is low, the motor rotates clockwise and when the commutation of the home sensor is revealed the motor slows down but keeps moving clockwise until it finds an index pulse of the encoder, which will become the home position.

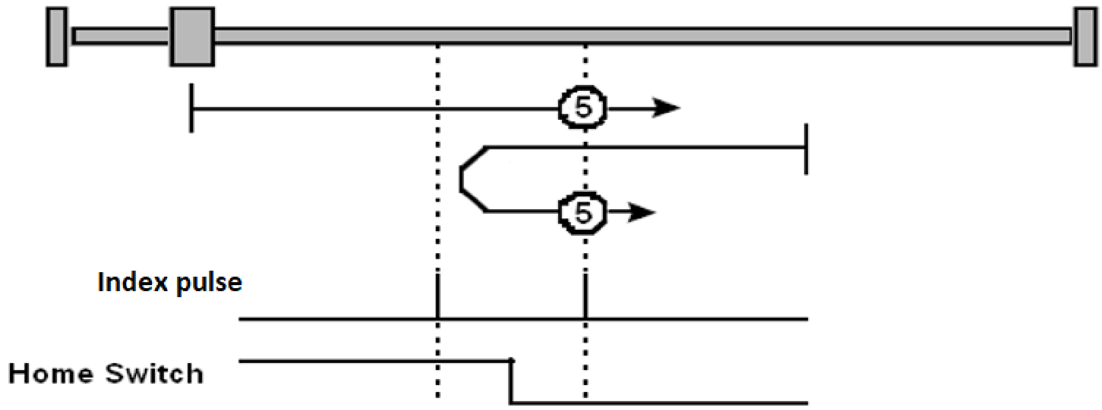


3.10.5.k Method 5 - Homing on negative home switch and index pulse

The logic value of the input assigned to the home sensor determines the direction of the search of the home sensor:

» If the assigned input is low, the motor rotates counterclockwise and when the commutation of the home sensor is revealed the motor stops and slowly moves clockwise until it finds an index pulse of the encoder, which will become the home position.

» If the assigned input is high, the motor rotates clockwise and when the commutation of the home sensor is revealed the motor slows down but keeps moving clockwise until it finds an index pulse of the encoder, which will become the home position.

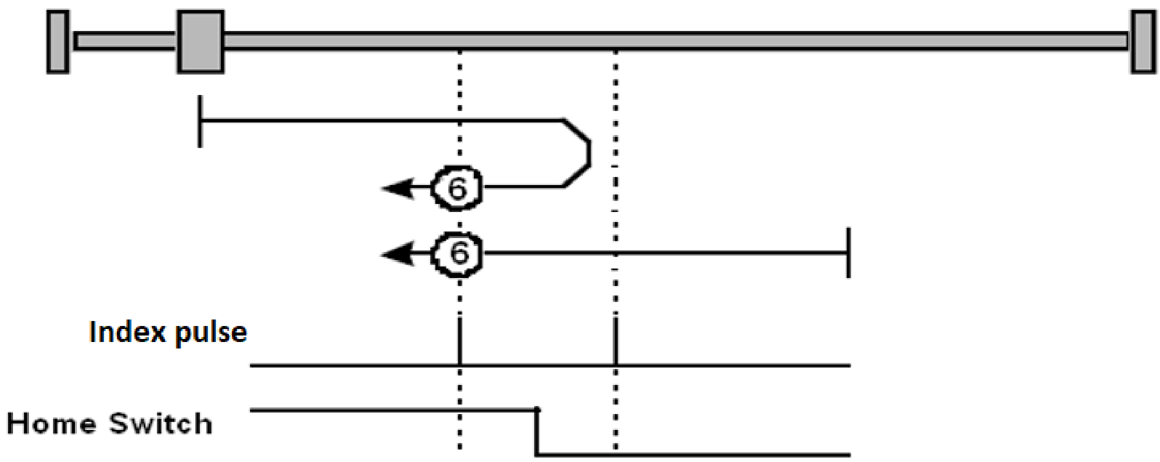


3.10.5.l Method 6 - Homing on negative home switch and index pulse

The logic value of the input assigned to the home sensor determines the direction of the search of the home sensor:

» If the assigned input is high, the motor rotates clockwise and when the commutation of the home sensor is revealed the motor stops and slowly moves counterclockwise until it finds an index pulse of the encoder, which will become the home position.

» If the assigned input is low, the motor rotates counterclockwise and when the commutation of the home sensor is revealed the motor slows down but keeps moving counterclockwise until it finds an index pulse of the encoder, which will become the home position.



3.10.5.m Method 7 - Homing on home switch and index pulse

The drive moves the motor clockwise searching for the home sensor. Once it finds it the motor stops and slowly moves counterclockwise to exit the sight of the sensor, and after that it continues moving counterclockwise until it finds an index pulse of the encoder. The position reached will become the home position. If in the beginning of this search the motor reachs the limit switch CW, the sense of rotation is switched to counterclockwise to guide the motor to the home sensor.

3.10.5.n Method 8 - Homing on home switch and index pulse

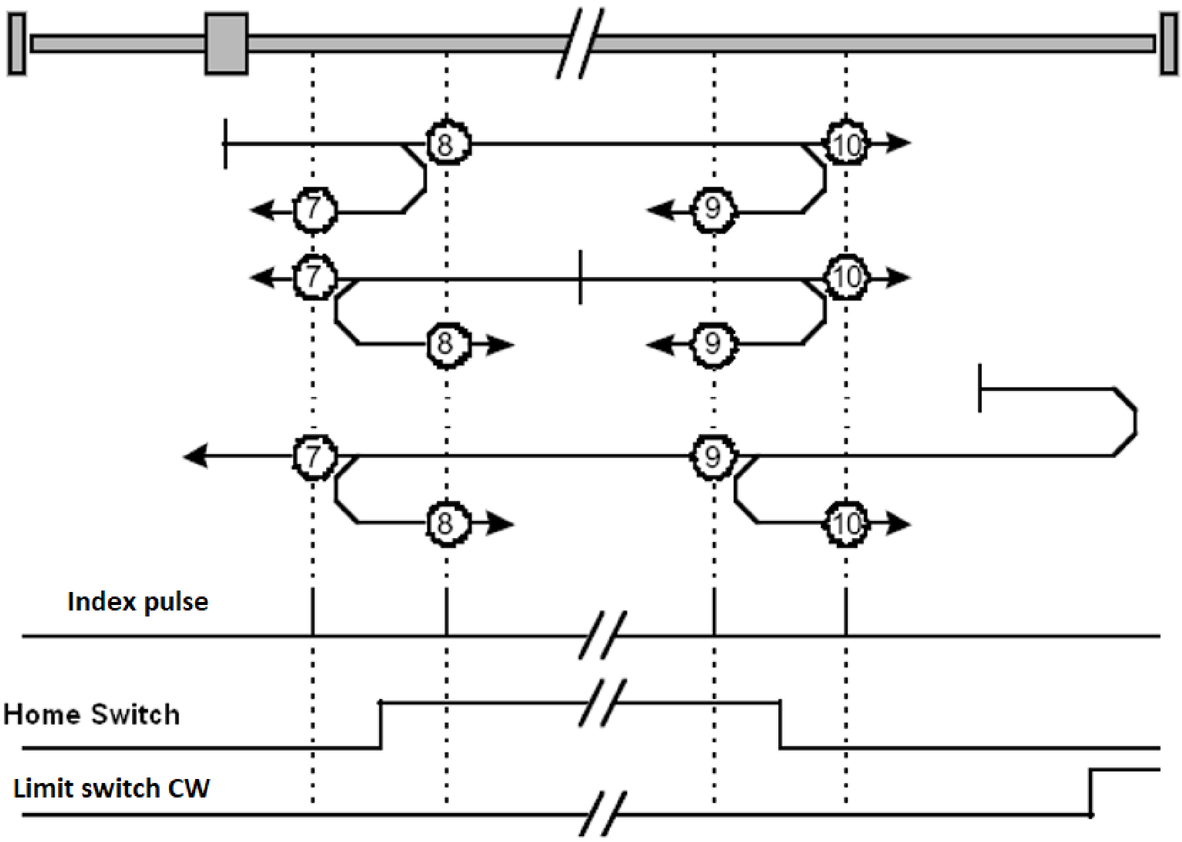
The drive moves the motor clockwise searching for the home sensor. Once it finds it the motor stops and slowly moves counterclockwise to exit the sight of the sensor, and after that it returns moving clockwise until it finds an index pulse of the encoder. The position reached will become the home position. If in the beginning of this search the motor reachs the limit switch CW, the sense of rotation is switched to counterclockwise to guide the motor to the home sensor.

3.10.5.o Method 9 - Homing on home switch and index pulse

The drive moves the motor clockwise searching for the home sensor. Once it finds it the motor slows down but keeps moving clockwise to exit the sight of the sensor, and after that it starts moving counterclockwise until it finds an index pulse of the encoder. The position reached will become the home position. If in the beginning of this search the motor reachs the limit switch CW, the sense of rotation is switched to counterclockwise to guide the motor to the home sensor.

3.10.5.p Method 10 - Homing on home switch and index pulse

The drive moves the motor clockwise searching for the home sensor. Once it finds it the motor slows down but keeps moving clockwise to exit the sight of the sensor, and after that it continues moving clockwise until it finds an index pulse of the encoder. The position reached will become the home position. If in the beginning of this search the motor reachs the limit switch CW, the sense of rotation is switched to counterclockwise to guide the motor to the home sensor.



3.10.5.q Method 11 - Homing on home switch and index pulse

The drive moves the motor counterclockwise searching for the home sensor. Once it finds it the motor stops and slowly moves clockwise to exit the sight of the sensor, and after that it continues moving clockwise until it finds an index pulse of the encoder. The position reached will become the home position. If in the beginning of this search the motor reachs the limit switch CCW, the sense of rotation is switched to clockwise to guide the motor to the home sensor.

3.10.5.r Method 12 - Homing on home switch and index pulse

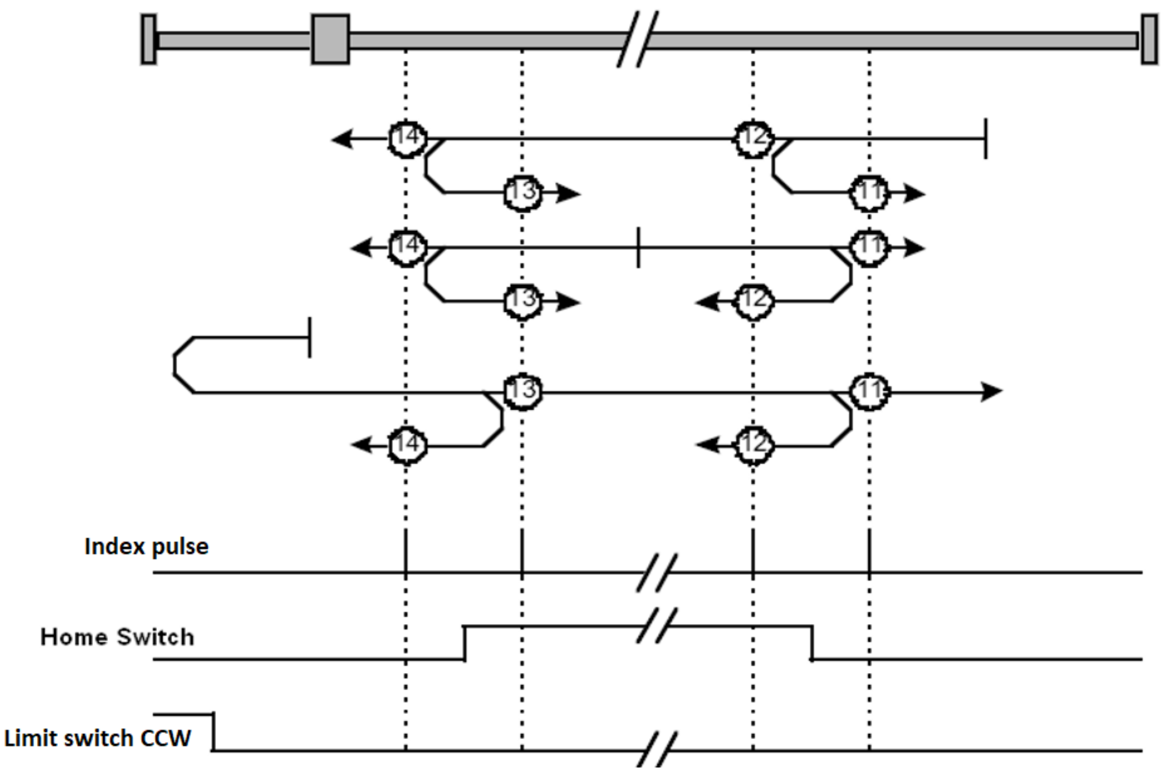
The drive moves the motor counterclockwise searching for the home sensor. Once it finds it the motor stops and slowly moves clockwise to exit the sight of the sensor, and after that it returns moving counterclockwise until it finds an index pulse of the encoder. The position reached will become the home position. If in the beginning of this search the motor reachs the limit switch CCW, the sense of rotation is switched to clockwise to guide the motor to the home sensor.

3.10.5.s Method 13 - Homing on home switch and index pulse

The drive moves the motor counterclockwise searching for the home sensor. Once it finds it the motor slows down but keep moving counterclockwise to exit the sight of the sensor, and after that it starts moving clockwise until it finds an index pulse of the encoder. The position reached will become the home position. If in the beginning of this search the motor reachs the limit switch CCW, the sense of rotation is switched to clockwise to guide the motor to the home sensor.

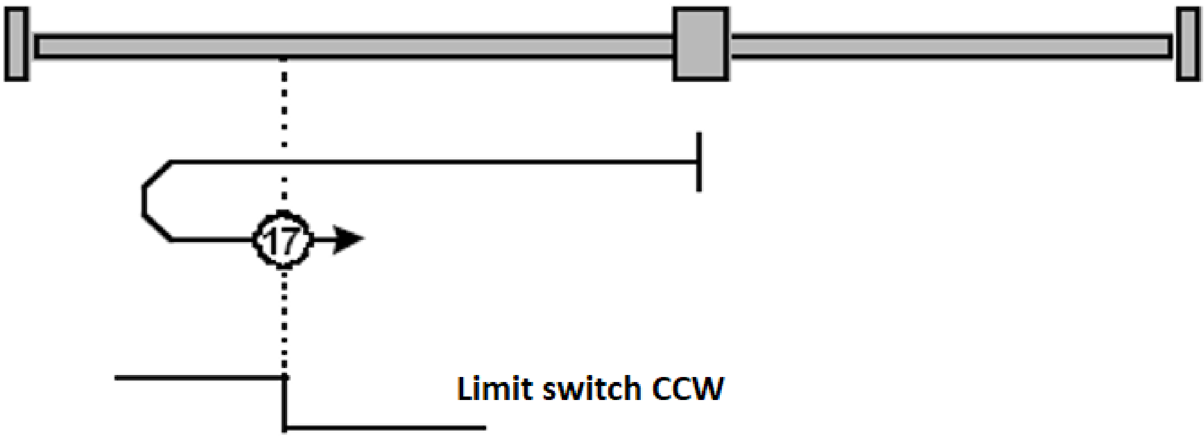
3.10.5.t Method 14 - Homing on home switch and index pulse

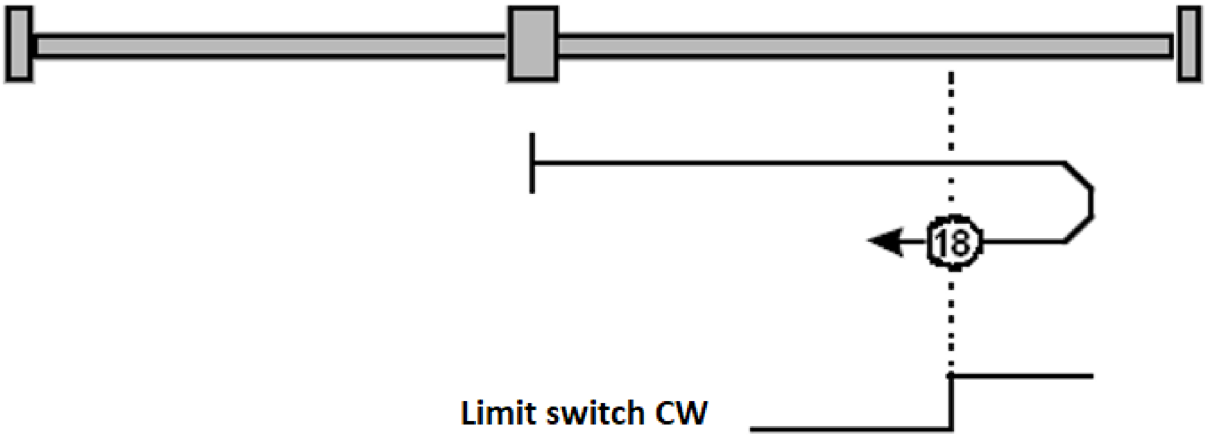
The drive moves the motor counterclockwise searching for the home sensor. Once it finds it the motor slows down but keep moving counterclockwise to exit the sight of the sensor, and after that it continues moving counterclockwise until it finds an index pulse of the encoder. The position reached will become the home position. If in the beginning of this search the motor reachs the limit switch CCW, the sense of rotation is switched to clockwise to guide the motor to the home sensor.



3.10.5.u Homing without index pulse

The homing methods ranging from 17 to 30 correspond with the respective methods ranging from 1 to 14, the only difference is that in this case the search of the index pulse of the encoder is not excecuted. For example, in the picture below we can see how the homing methods 17 and 18 are excecuted.



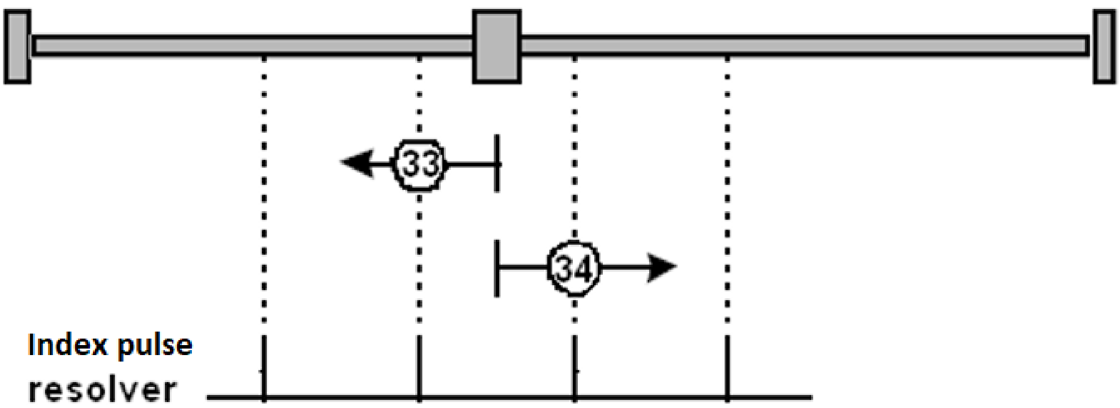


3.10.5.v Method 33 – Homing on index pulse (counterclockwise)

Starting from the position where it is, the drive moves counterclockwise to reach an index pulse of the encoder; the position reached will become the home position.

3.10.5.w Method 34 -Homing on index pulse (clockwise)

Starting from the position where it is, the drive moves clockwise to reach an index pulse of the encoder; the position reached will become the home position.



3.10.5.x Method 35 - Homing on the current position

When this method is escecuted the current position will become the home position.

3.10.5.y Registers 0641h (1601 dec) 0642h (1602 dec) Switch search speed

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0641h (MSB) **0642h (LSB)**	**Switch search speed**	UINT32	1... 7FFFFFFFh	100 (dec)		Always	Always

These registers contain the ***Switch search speed***, i.e. the speed reached by the motor while it's searching for the home switch.This value is converted to the internal unit of the drive (increments per second [increments/s]) using the conversion factors ***Speed Factor Numerator*** (registers 0406h, 0407h) and ***Speed Factor Denominator*** (registers 0408h e 0409h).The formula for the conversion is showed below:

$$drive\_switch\_search\_speed = \frac{(Speedfactor Numerator \times switch\_search\_speed)}{(Speedfactordenominator)}$$

Using the default values of the ***Speed factors***, ie *Numerator*=65536 and *Denominator*=60, the ***switch search speed*** is expressed in [rpm/s].

Note: one turn of the motor correspond to 65536 internal increments.

3.10.5.z Registers 0643h (1603 dec) 0644h (1604 dec) Zero search speed

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0643h (MSB) **0644h (LSB)**	**Zero search speed**	UINT32	1... 7FFFFFFFh	10 (dec)		Always	Always

These registers contain the ***Zero search speed***, i.e. the speed reached by the motor while it's exiting from the sight of the home switch (or the limit switch) to search for the index impulse of the encoder (*Note: depending on the method used there may not be the need to search an index impulse*). This value is converted to the internal unit of the drive (increments per second [increments/s]) using the conversion factors ***Speed Factor Numerator*** (registers 0406h, 0407h) and ***Speed Factor Denominator*** (registers 0408h e 0409h).The formula for the conversion is showed below:

$$Drive\_zero\_search\_speed = \frac{(Speed\_factor\_Numerator \times Zero\_search\_speed)}{(Speed\_factor\_denominator)}$$

Using the default values of the ***Speed factors***, ie *Numerator*=65536 and *Denominator*=60, the ***zero search speed*** is expressed in [rpm/s].

Note: one turn of the motor correspond to 65536 internal increments.

3.10.5.aa Registers 0645h (1605 dec),0646h (1606 dec) Homing Acc/Dec

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0645h (MSB) **0646h (LSB)**	**Homing Acc/Dec**	UINT32	1 ... 7FFFFFFFh	1000 (dec)		Always	Always

These registers contain the ***Homing Acceleration,*** i.e. the absolute value of acceleration reached when the homing procedure is excecuted. inside the drive it's expressed as [increments/s^2]. This value is converted in the internal unit of measure of the drive using the ***Acceleration factor numerator*** (registers 040Ah, 040Bh) and the ***Acceleration factor denominator*** (registers 040Ch, 040Dh) as showed in the formula below:

$$drive\_homing\_acc = \frac{(Acceleration factor Numerator \times homing\_acc)}{(Acceleration factor denominator)}$$

Using the default values of the ***Acceleration factors***, ie *Numerator*=65536 and *Denominator*=60, the ***Homing acceleration*** is expressed in [rpm/s]

Note: one turn of the motor correspond to 65536 internal increments.

3.10.5.ab Registers 647h (1607 dec), 648h (1608 dec) - Home shift

Address	Name	Type	Range	Default	Unit	Read	Write
0647h (MSB) **0648h (LSB)**	**Home shift**	INT32	±7FFFFFFFh	0		Always	Always

These registers contain the ***Home shift***. At the end of the homing procedure the motor stops over an index impulse of the encoder or a limit switch, depending on the procedure used. When ***Home shift*** differs from 0, the motor makes another shift clockwise or counterclockwise from the point where it stopped; the value and the direction of this shift depends on the ***Home shift***.

3.10.5.ac Register 649h (1609 dec) - Mechanical stop current

Address	Name	Type	Range	Default	Unit	Read	Write
649h	**Mechanical stop current**	UINT16	10...950	700	%/10	Always	Always

This register contains the limit value of the current used to verify if a mechanical stop has been reached. When the motor is absorbing this current we assume that it's actually pushing on a mechanical stop. This current is expressed as a percentage relative to the nominal current.

3.10.5.ad Register 64Ah (1610 dec) - Mechanical stop time

Address	Name	Type	Range	Default	Unit	Read	Write
64Ah	**Mechanical stop time**	UINT16	10... 10000	1000	ms	Always	Always

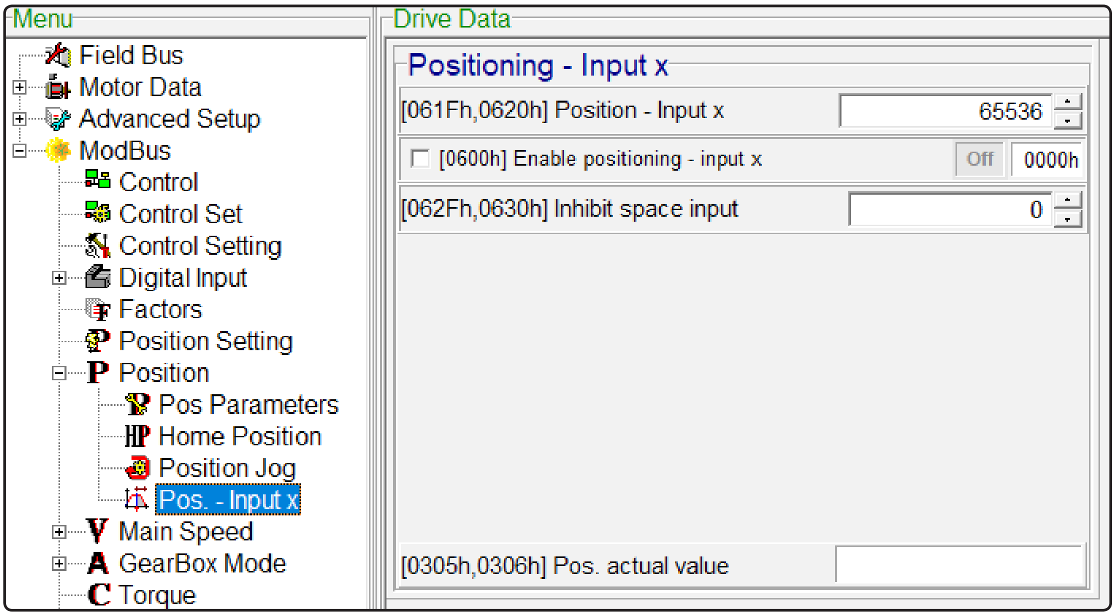
This register contains a time value used to verify if a mechanical stop has been reached. When the motor is absorbing a current greater than or equal to the one set on register 649h for a time longer than the one sets in this register, we assume that the motor is actually pushing on a mechanical stop.

3.10.6 Positioning- Input x

This paragraph describes the function ***Positioning-Input X***, which can be used only when the Drive is working as a ***Positioner***. When this function is launched firstly the current position of the motor is stored. Then a certain space (expressed as increments) is added to the previous stored position; the obtained result becomes the new target position of the Drive. Now the motor will start moving to the new target and it will stop once it has reached it. The bit 5 of the *status word* will be raised to indicate that the target has been reached. The command to start the ***Positioning-Input X*** function is assigned to a digital input; when this input is high the procedure described above will be excecuted. Usually in the applications where this function is used the digital input that commands its start is connected to a sensor, which observes material passing in front of it. When the sensor reveals a mark on the observed material the ***Positioning-Input X*** is launched, and the controlled motor will move as requested to excecute a certain task. The parameters used to set the ***Positioning-Input X*** function are listed in the table below:

Address	Name	Type	Read	Write
0600h	Enable pos. input X	UINT16	always	always
061Fh	Position - Input X (MSB)	INT32	always	always
0620h	Position - Input X (LSB)			
062Fh	Inhibition space (MSB)	UINT32	always	always
0630h	Inhibition space (LSB)			

The parameters listed above can be set in the Caliper menu "***Position-Pos. input X***" in which the following screen appears:



3.10.6.a Register 0600h (1536 dec) - Enable pos. Input x

Address	Name	Type	Range	Default	Unit	Read	Write
0600h	**Enable input x**	INT16	0 ... 65535 (bit)	0		Always	Always

This register contains a flag which is used to enable the ***Positioning-Input X*** function.

3.10.6.b Registers 061Fh (1567 dec), 0620h (1568 dec) - Position - Input x

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
061Fh (MSB) **0620h (LSB)**	**Position - Input x**	INT32	± 7FFFFFFFh	65536		Always	Always

These registers contain the value of space that the drive will add to the position measured when the ***Positioning-Input X*** function is launched (see *"3.10.6 Positioning- Input x" pag. 69*). The obtained result is the new target position that the motor is going to reach. The value of this variable has no sign, because the direction of the movement will be calculated by the drive in a way that its sign is equal to the ones of the measured speed.

3.10.6.c Registers 062Fh (1583 dec), 0630h (1584 dec) - Inhibition space

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
062Fh (MSB) **0630h (LSB)**	**Inhibition space**	UINT32	0 ... 10000000h	0		Always	Always

These registers contain a threshold value of space used by ***Positioning-Input X*** function. At paragraph *"3.10.6 Positioning- Input x" pag. 69* it was explained that the command that launch this function usually is a digital input connected to a sensor, which observes material passing in front of it. When the sensor reveals a mark on the material the ***Positioning-Input X*** function is launched. It may happens that the material has some imperfections that the sensor could mistake for a mark, which would cause an anticipated execution of this positioning. To solve this problem the variable ***Inhibition space*** is introduced: the sensor is going to consider as an intended mark only the ones seen after the motor has moved for a space longer than the ***Inhibition space*** from the last target; the ones seen before will be regarded as imperfection.

3.11 Speed mode

The control type ***Speed*** is selected when the parameter "***control type***" (Register 0401h) is set to 0.

3.11.1 Control word and Status word

In this paragraph are described the structures of the *Control word* and the *Status word* when the ***Speed*** mode is selected.

3.11.1.a Setting of the Control word bits in Speed mode

Bit	Description
0	***Switch on***
1	***Enable operation***
2	***Fault reset***
3	***Halt***
4 ... 15	***Manufacturer specific***

3.11.1.b Description of the Status word bits in Speed mode

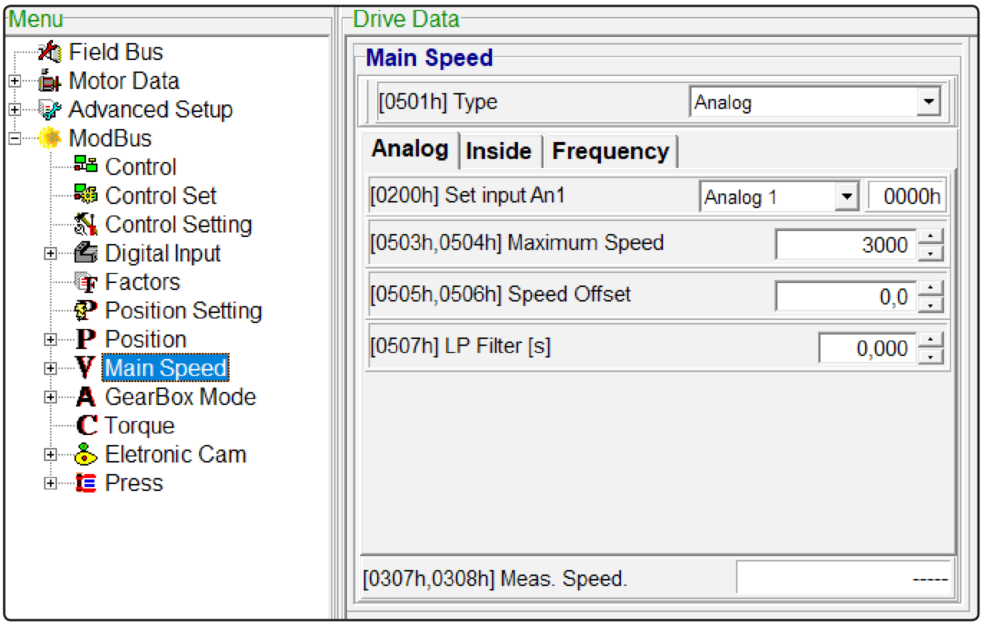
<table>
<tr><th>Bit</th><th colspan="2">Description</th></tr>
<tr><td>0</td><td colspan="2">Switch on</td></tr>
<tr><td>1</td><td colspan="2">Enable operation</td></tr>
<tr><td>2</td><td colspan="2">Ramp Stop</td></tr>
<tr><td>3</td><td colspan="2">Ramp Halt</td></tr>
<tr><td>4</td><td colspan="2">Fault</td></tr>
<tr><td>5..8</td><td colspan="2">Manufacturer specific</td></tr>
<tr><td rowspan="4">9</td><td colspan="2">Remote</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Remote Control disabled</td></tr>
<tr><td>1</td><td>Remote Control enabled</td></tr>
<tr><td rowspan="4">10</td><td colspan="2">Torque Limit</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Torque limit not reached</td></tr>
<tr><td>1</td><td>Torque limit reached</td></tr>
<tr><td rowspan="4">11</td><td colspan="2">Zero speed</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Motor is at a standstill</td></tr>
<tr><td>1</td><td>Motor is moving</td></tr>
<tr><td>12</td><td colspan="2">Manufacturer specific</td></tr>
<tr><td>13</td><td colspan="2">Warning</td></tr>
<tr><td>14</td><td colspan="2">Fault reaction</td></tr>
<tr><td>15</td><td colspan="2">Speed reached</td></tr>
</table>

3.11.2 Main Speed Parameters

This paragraph lists the parameters used to set the Main reference speed when the drive is working with the ***Speed*** control.

Address	Name	Type	Read	Write
0500h	Speed Flag	UINT16	always	always
0501h	Main speed type	INT16	always	always
0503h	Maximum speed (MSB)	UINT32	always	always
0504h	Maximum speed (LSB)			
0505h	Speed offset(MSB)	INT32	always	always
0506h	Speed offset(LSB)			
0507h	Low pass filter	UINT16	always	always
0508h	Target speed (MSB)	UINT32	always	always
0509h	Target speed (LSB)			
050Ah	pulses per revolution	UINT16	always	always
051Dh	Measured speed (MSB)	INT32	always	no
051Eh	Measured speed (LSB)			

These parameters can be set in the Caliper menu "***Main speed***" in which the following screen appears:



3.11.2.a Register 0500h (1280 dec) Modbus speed flag

Address	Name	Type	Range	Default	Unit	Read	Write
0500h	**Modbus speed flag**	UINT16	0..000Fh	0	Bit	Always	Always

This register contains some flags used when the Drive is working on ***Speed*** mode. The Bits 0 and 1 of this register defines what kind of digital inputs are used for the frequency reference, according to the table below:

Bit 1	Bit 0	Modbus speed flag- Description of bits 0 and 1
0	0	***Channel A-B*** : the reference is generated using signlas in quadrature
0	1	***Frequency-Direction***: One channels provides informations on the direction, the other provides informations on the speed
1	0	***Impulses CW CCW***: Depending on which channel receives the impulses the motor will rotate in a sense or the other
1	1	Not allowed

The bits 2,3 and 4 have the functions listed below:

Bit	*Name*	Modbus speed flag- Description of bits 2,3 and 4
2	Enable auxiliary reference	0 = auxiliary reference disabled
		1 = auxiliary reference enabled
3	Enable ramps	0 = Ramps disabled
		1 = Ramps enabled
4	Enable S ramps	0 = S Ramps disabled
		1 = S Ramps enabled
5 ... 16	reserved	

3.11.2.b Register 0501h (1281 dec) Main speed type

Address	Name	Type	Range	Default	Unit	Read	Write
0501h	**Main speed type**	UINT16	0...2	0		Always	Switch on

The parameter "***Main speed type***" selects the main reference speed used when the drive is working with the ***Speed*** control.

Value	Description
0	***Analogic reference***: the drive uses a reference speed obtained by an analogic signal with range +-10V
1	***Inside reference***: the drive uses a reference speed which is equal to the valued set on registers 0508h, 0509h.
2	***Frequency reference***:the drive uses a reference speed obtained by the frequency signal obtained by the inputs Pulse-Dir.
3	***Speed table reference***: the drive uses a reference speed obtained by a speed table stored on the drive. In this table can be saved at most 64 values of speed. The acceleration and deceleration can be set for each of these values.

3.11.2.c Registers 0503h (1283 dec), 0504h (1284 dec) Maximum speed

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0503h (MSB) **0504h (LSB)**	**Maximum speed**	UINT32	0 ... 7FFFFFFFh	3000 (dec)	rpm	Always	Always

These registers contain the full scale of speed,expressed in [rpm], for the main analogical speed input. This value is matched to 10 [V], which is the full scale of an analogical signal expressed in [Volt]. For the calculation of the reference speed in [rpm] the drive will do a proportion using the full scale in [rpm], the full scale in [Volt] and the measured analogic value in [Volt] that corresponds to the reference speed. The obtained result is converted to the internal unit of the drive (increments per second [increments/s]) using the conversion factors ***Speed Factor Numerator*** (registers 0406h, 0407h) and ***Speed Factor Denominator*** (registers 0408h e 0409h).The formula for these calculation is showed below:

$$drive\_reference\_speed = \frac{[SpeedFactorNumerator \times (\frac{(Maximum\_speed)}{10}) \times volt\_input]}{(SpeedFactordenominator)}$$

3.11.2.d Registers 0505h (1285 dec),0506h (1286 dec) Speed offset

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0505h (MSB) **0506h (LSB)**	**Speed offset**	INT32	±7FFFFFFFh	0		Always	Always

These registers contain an offset of speed that could be added or subtracted from the main analogical speed reference. This value is converted to the internal unit of the drive (increments per second [increments/s]) using the conversion factors ***Speed Factor Numerator*** (registers 0406h, 0407h) and ***Speed Factor Denominator*** (registers 0408h e 0409h).The formula for the conversion is showed below:

$$Drive\_speed\_offset = \frac{[SpeedFactorNumerator \times (Speed\_Offset)]}{(SpeedFactordenominator)}$$

Using the default values of the ***Speed factors***, ie *Numerator*=65536 and *Denominator*=60, the ***speed offset*** is expressed in [rpm/s].

Note: one turn of the motor correspond to 65536 internal increments.

3.11.2.e Register 0507h (1287 dec) Low pass filter

Address	Name	Type	Range	Default	Unit	Read	Write
0507h	**Low pass filter**	UINT16	0...1000	0		Always	Always

this register contains the time costant of the low pass filter applied to the main analogical input of speed. It's expressed in [msec]. This filter is used to eliminate the potential noise that could affects the analogic signals.

3.11.2.f Registers 0508h (1288 dec), 0509h (1289 dec) Target speed

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0508h (MSB) 0509h (LSB)	**Target speed**	UINT32	±7FFFFFFFh	0		Always	Always

These registers contain the ***Target Speed***, which corresponds to the speed reference when the register 0501h is set on "***Inside reference***". This value is converted to the internal unit of the drive (increments per second [increments/s]) using the conversion factors ***Speed Factor Numerator*** (registers 0406h, 0407h) and ***Speed Factor Denominator*** (registers 0408h e 0409h).The formula for the conversion is showed below:

$$Drive\_target\_speed = \frac{(SpeedFactorNumerator \times Target\_speed)}{(SpeedFactordenominator)}$$

Using the default values of the ***Speed factors***, ie *Numerator*=65536 and *Denominator*=60, the ***Target speed*** is expressed in [rpm/s].

Note: one turn of the motor correspond to 65536 internal increments.

3.11.2.g Register 050Ah (1290 dec) Pulses per revolution

Address	Name	Type	Range	Default	Unit	Read	Write
050Ah	**Pulses per revolution**	UINT16	128..,16384	1024		Always	Switch off

This register contains the number of pulses that we want to match to one turn of the motor.

3.11.2.h Registers 051Dh (1309 dec),051Eh (1310 dec) Measured speed

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
051Dh (MSB) 051Eh (LSB)	**Measured speed**	INT32	±7FFFFFFFh			Always	No

These registers contain the measured speed of the motor. This value is converted to the internal unit of the drive (increments per second [increments/s]) using the conversion factors ***Speed Factor Numerator*** (registers 0406h, 0407h) and ***Speed Factor Denominator*** (registers 0408h e 0409h).The formula for the conversion is showed below:

$$drive\_measured\_speed = \frac{(Speedfactordenominator \times measured\_speed)}{(Speedfactornumerator)}$$

Using the default values of the ***Speed factors***, ie *Numerator*=65536 and *Denominator*=60, the ***Measured speed*** is expressed in [rpm/s].

Note: one turn of the motor correspond to 65536 internal increments.

3.11.3 Parameters of the Auxiliary speed reference

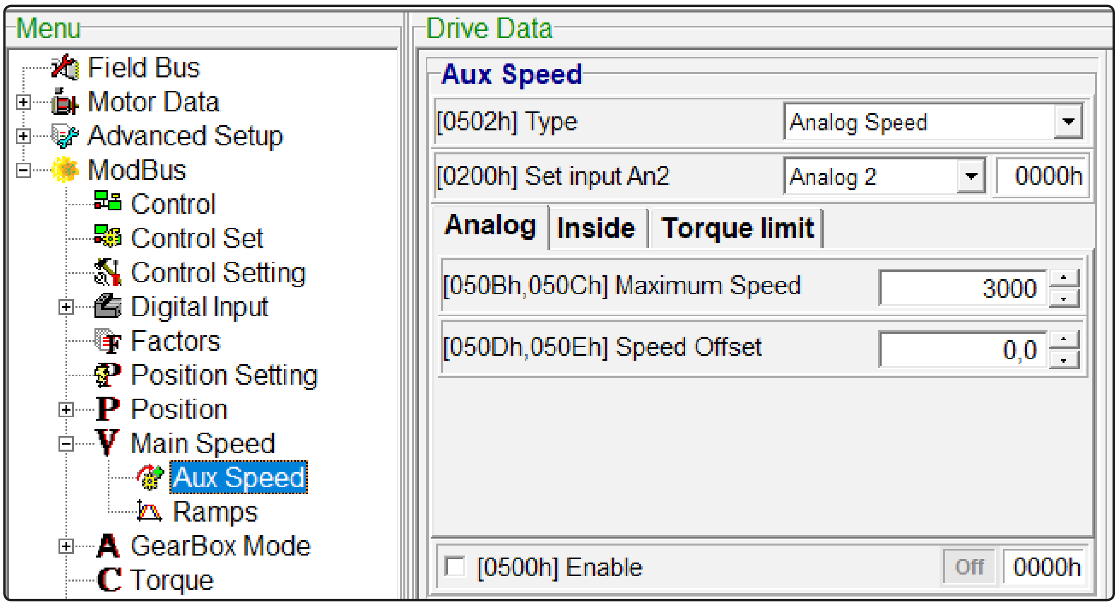
This paragraph lists the parameters used to set the Auxiliary reference of speed. this is an additional reference that could be used when the drive works on ***Speed*** mode. It can have two functions:

» it could add or subtract a value, obtained from an anaogical signal or via Caliper, to the Main speed reference. This allow you to adjust manually the reference of speed calculated by the Drive;

» it could be used to change manually the torques limit of the Drive.

Address	Name	Type	Read	Write
0500h	Modbus speed Flag	UINT16	always	always
0502h	Auxiliary reference type	INT16	always	always
050Bh	Maximum speed aux (MSB)	UINT32	always	always
050Ch	Maximum speed aux (LSB)			
050Dh	Offset speed aux (MSB)	INT32	always	always
050Eh	Offset speed aux (LSB)			
050Fh	Inside speed aux (MSB)	INT32	always	always
0510h	Inside speed aux (LSB)			
0511h	Max torque limit	UINT16	always	always
0512h	Offset torque limit	INT16	always	always

These parameters can be set in the Caliper menu "***Main speed- aux speed***" in which the following screen appears:



3.11.3.a Register 0500h (1280 dec) Modbus speed flag

Address	Name	Type	Range	Default	Unit	Read	Write
0500h	**Modbus speed flag**	UINT16	0..000Fh	0	Bit	Always	Always

This register was already described in paragraph *"3.11.1.a Setting of the Control word bits in Speed mode" pag. 71*. In this paragraph for convenience we report only the description of the Bit 2 that enables the auxiliary reference.

Bit	Name	Modbus speed flag - Description Bit 2
2	Enable auxiliary reference	0= Auxiliary reference disabled
		1= Auxiliary reference enabled

3.11.3.b Register 0502h (1282 dec) Auxiliary reference type

Address	Name	Type	Range	Default	Unit	Read	Write
0502h	**Auxiliary reference type**	UINT16	0..,2	0		Always	Switch on

The parameter "***Auxiliary reference type***" allows you to choose what type of auxiliary reference (if another speed reference or a torque limit) the Drive is going to use when it's set on ***Speed*** control:

Value	Description
0	***Analogical speed***: the drive adds the reference of speed received from the auxiliary analogical input (ranging from 0 to 10 [V]) to the main speed reference
1	***Inside speed***: the drive adds the reference of speed wrote on registers 050Fh, 0510h to the main speed reference
2	***Torque limit***: the drive uses the signal obtained from the auxiliary analogical input (ranging from 0 to 10 [V]) to configure the torque limit

3.11.3.c Registers 050Bh (1291 dec),050Ch (1292 dec) Maximum speed aux

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
050Bh (MSB) 050Ch (LSB)	**Maximum speed aux**	UINT32	0 ...7FFFFFFFh	3000 (dec)		Always	Always

These registers contain the full scale of speed,expressed in [rpm], for the auxiliary analogical speed input. This value is matched to 5 [V], which is the full scale of an analogical signal expressed in [Volt]. For the calculation of the auxiliary reference speed in [rpm] the drive will do a proportion using this full scale and the measured analogic value in [Volt] reduced by 5 [V] (this is done because the analog signal can only range from 0 to 10 [V], so we have to center the 0 of the speed to 5 [V] if we want to represents both positive and negative speeds).The obtained result is converted to the internal unit of the drive (increments per second [increments/s]) using the conversion factors ***Speed Factor Numerator*** (registers 0406h, 0407h) and ***Speed Factor Denominator*** (registers 0408h e 0409h).The formula for these calculation is showed below:

$$Drive\_ref\_speed\_aux = \frac{[SpeedfactorNumerator \times (\frac{(Maximum\_speed\_aux)}{5}) \times (volt\_input - 5)]}{(Speedfactordenominator)}$$

3.11.3.d Registers 050Dh (1293 dec) 050Eh (1294 dec) Offset speed aux

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
050Dh (MSB) 050Eh (LSB)	**Offset speed aux**	INT32	±7FFFFFFFh	0		Always	Always

These registers contain an offset of speed that is added or removed from the speed obtained from the auxiliary analog input.This value is converted to the internal unit of the drive (increments per second [increments/s]) using the conversion factors ***Speed Factor Numerator*** (registers 0406h, 0407h) and ***Speed Factor Denominator*** (registers 0408h e 0409h).The formula for the conversion is showed below:

$$drive\_Offset\_speed\_aux = \frac{[SpeedfactorNumerator \times (Offset\_speed\_aux)]}{(SpeedFactorDenominator)}$$

Using the default values of the ***Speed factors***, ie *Numerator*=65536 and *Denominator*=60, the ***Offset speed aux*** is expressed in [rpm/s].

Note: one turn of the motor correspond to 65536 internal increments.

3.11.3.e Registers 050Fh (1295 dec),0510h (1296 dec) Inside speed aux

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
050Fh (MSB) **0510h (LSB)**	**Inside speed aux**	INT32	±7FFFFFFFh	0		Always	Always

The ***Inside speed aux*** is a value of speed that we can write directly on Caliper. When register 0502h is set on "***Inside speed***" and the auxiliary reference is enabled, the ***Inside speed aux*** is added to the main speed reference.This value is converted to the internal unit of the drive (increments per second [increments/s]) using the conversion factors ***Speed Factor Numerator*** (registers 0406h, 0407h) and ***Speed Factor Denominator*** (registers 0408h e 0409h).The formula for the conversion is showed below:

$$drive\_target\_speed\_aux = \frac{(SpeedfactorNumerator \times Target\_speed\_aux)}{(SpeedFactorDenominator)}$$

Using the default values of the ***Speed factors***, ie *Numerator*=65536 and *Denominator*=60, the ***Inside speed aux*** is expressed in [rpm/s].

Note: one turn of the motor correspond to 65536 internal increments.

3.11.3.f Register 0511h (1297 dec) Max torque limit

Address	Name	Type	Range	Default	Unit	Read	Write
0511h	**Max torque limit**	UINT16	0..2500	1000		Always	Always

This register is used when the auxiliary reference type is set on "***Torque limit***". It contains the torque full scale for the auxiliary analogic input, which is used to limits the maximum torque that the controlled motor could reach. This value is expressed as thousandths of the ***Nominal current*** (register 0153h). We can calculate the Torque limt current in [A] using the formula below:

$$Torque\_limit\_current = \frac{(Fullscale\_torque\_limit\ xI\_nom\_motor)}{(1000 \times 5)} \times (Volt\_input\_aux - 5)$$

3.11.3.g Register 0512h (1298 dec) Offset Torque limit

Address	Name	Type	Range	Default	Unit	Read	Write
0512h	**Offset torque limit**	INT16	-1000... +1000	0		Always	Always

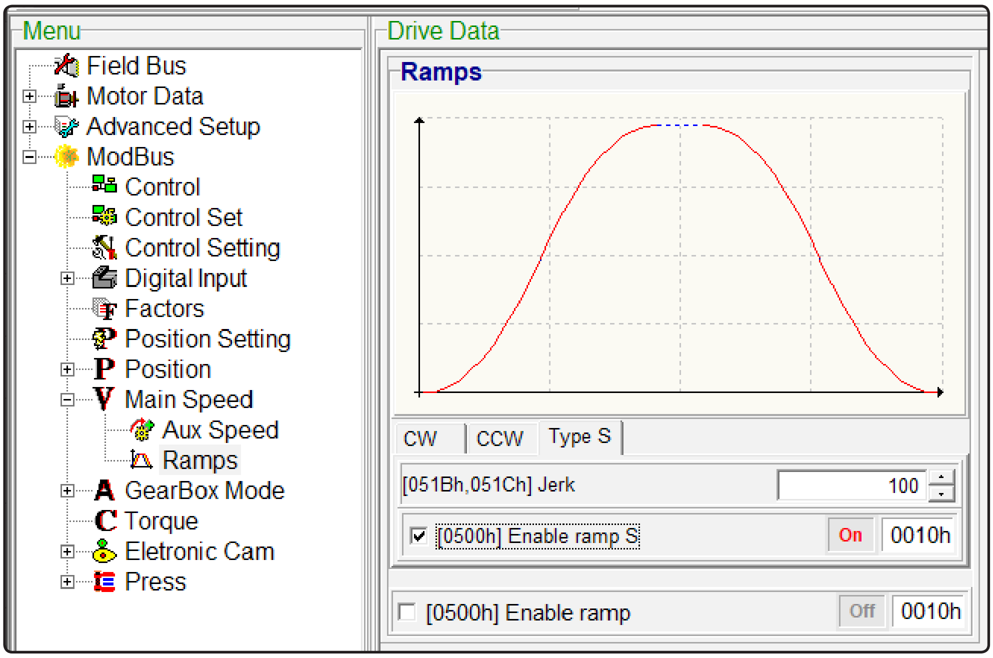
This register is used when the auxiliary reference type is set on "***Torque limit***". It contains an offset that is added or removed from the auxiliary analogic input, that transmits to the drive the torque full scale (register 0511h).This value is expressed as thousandths of the ***Nominal current*** (register 0153h).

3.11.4 Ramps parameters

This parameters lists the parameters used to set the speed ramps. They're used when the drive works with the ***Speed*** control.

Address	Name	Type	Read	Write
0500h	Modbus speed flag	UINT16	always	always
0513h	Acceleration CW (MSB)	UINT32	always	always
0514h	Acceleration CW (LSB)			
0515h	Deceleration CW (MSB)	UINT32	always	always
0516h	Deceleration CW (LSB)			
0517h	Acceleration CCW (MSB)	UINT32	always	always
0518h	Acceleration CCW (LSB)			
0519h	Deceleration CCW (MSB)	UINT32	always	always
051Ah	Deceleration CCW (LSB)			
051Bh	Jerk (MSB)	UINT32	always	always
051Ch	Jerk (LSB)			

These parameters can be set in the Caliper menu "***Main speed- ramps***" in which the following screen appears:



3.11.4.a Register 0500h (1280 dec) Modbus speed flag

Address	Name	Type	Range	Default	Unit	Read	Write
0500h	**Modbus speed flag**	UINT16	0..000Fh	0	Bit	Always	Always

This register was already described in paragraph *"3.11.1.a Setting of the Control word bits in Speed mode" pag. 71*. In this paragraph for convenience we report only the description of the Bits 3 and 4 that enables the speed ramps.

Bit	Name	Modbus speed flag - Description Bits 3,4
3	Enable ramps	0= Ramps disabled
		1= Ramps enabled
4	Enable S ramps	0=S Ramps disabled
		1= S Ramps enabled (only when Bit 3 =1)

3.11.4.b Registers 0513h (1299 dec), 0514h (1300 dec) Acceleration CW

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0513h (MSB) **0514h (LSB)**	**Acceleration CW**	UINT32	0 ... 7FFFFFFFh	1000 (dec)		Always	Always

These registers are used when the Drive is set on ***Speed*** control and the speed ramps are enabled. They contain the value of the acceleration applied when the motor is moving clockwise.Inside the drive it's expressed as [increments/ s^2]. This value is converted in the internal unit of measure of the drive using the ***Acceleration factor numerator*** (registers 040Ah, 040Bh) and the ***Acceleration factor denominator*** (registers 040Ch, 040Dh) as showed in the formula below:

$$Drive\_accel\_CW = \frac{(AccelerationFactorNumerator \times Accel\_CW)}{(AccelerationFactorDenominator)}$$

Using the default values of the ***Acceleration factors***, ie *Numerator*=65536 and *Denominator*=60, the ***acceleration CW*** is expressed in [rpm/s]

Note: one turn of the motor correspond to 65536 internal increments.

3.11.4.c Registers 0515h (1301 dec) 0516h (1302 dec) Deceleration CW

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0515h (MSB) **0516h (LSB)**	**Deceleration CW**	UINT32	0 ... 7FFFFFFFh	1000 (dec)		Always	Always

These registers are used when the Drive is set on ***Speed*** control and the speed ramps are enabled. They contain the value of the deceleration applied when the motor is moving clockwise.Inside the drive it's expressed as [increments/s^2]. This value is converted in the internal unit of measure of the drive using the ***Acceleration factor numerator*** (registers 040Ah, 040Bh) and the ***Acceleration factor denominator*** (registers 040Ch, 040Dh) as showed in the formula below:

$$Drive\_decel\_CW = \frac{(AccelerationFactorNumerator \times decel\_CW)}{(AccelerationFactorDenominator)}$$

Using the default values of the ***Acceleration factors***, ie *Numerator*=65536 and *Denominator*=60, the ***deceleration CW*** is expressed in [rpm/s]

Note: one turn of the motor correspond to 65536 internal increments.

3.11.4.d Registers 0517h (1303 dec) 0518h (1304 dec) Acceleration CCW

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0517h (MSB) **0518h (LSB)**	**Acceleration CCW**	UINT32	0 ... 7FFFFFFFh	1000 (dec)		Always	Always

These registers are used when the Drive is set on ***Speed*** control and the speed ramps are enabled. They contain the value of the acceleration applied when the motor is moving counterclockwise.Inside the drive it's expressed as [increments/s^2]. This value is converted in the internal unit of measure of the drive using the ***Acceleration factor numerator*** (registers 040Ah, 040Bh) and the ***Acceleration factor denominator*** (registers 040Ch, 040Dh) as showed in the formula below:

$$Drive\_accel\_CCW = \frac{(AccelerationFactorNumerator \times Accel\_CCW)}{(AccelerationFactorDenominator)}$$

Using the default values of the ***Acceleration factors***, ie *Numerator*=65536 and *Denominator*=60, the ***acceleration CCW*** is expressed in [rpm/s]

Note: one turn of the motor correspond to 65536 internal increments.

3.11.4.e Registers 0519h (1305 dec) 051Ah (1306 dec) Deceleration CCW

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0519h (MSB) **051Ah (LSB)**	**Deceleration CCW**	UINT32	0 ... 7FFFFFFFh	1000 (dec)		Always	Always

These registers are used when the Drive is set on ***Speed*** control and the speed ramps are enabled. They contain the value of the deceleration applied when the motor is moving counterclockwise.Inside the drive it's expressed as [increments/s^2]. This value is converted in the internal unit of measure of the drive using the ***Acceleration factor numerator*** (registers 040Ah, 040Bh) and the ***Acceleration factor denominator*** (registers 040Ch, 040Dh) as showed in the formula below:

$$Drive\_decel\_CCW = \frac{(AccelerationFactorNumerator \times decel\_CCW)}{(AccelerationFactorDenominator)}$$

Using the default values of the ***Acceleration factors***, ie *Numerator*=65536 and *Denominator*=60, the ***deceleration CCW*** is expressed in [rpm/s]

Note: one turn of the motor correspond to 65536 internal increments.

3.11.4.f Registers 051Bh (1307 dec) 051Ch (1308 dec) Jerk.

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
051Bh (MSB) **051Ch (LSB)**	**Jerk**	UINT32	0 …980000h	1000 (dec)		Always	Always

These registers contain the ***Jerk***, a value used to give to the speed a parabolic profile at the start and the end of the speed ramps. An high value of Jerk means that the acceleration variation are bigger, which means a lower accleration time but with higher mechanical stress; vice versa, A low value of Jerk means that the acceleration variation are lower, which means an higher accleration time but with lower mechanical stress. The unit of measure of the jerk is "acceleration divided by time"; inside the drive it's expressed as [increments/ s^3]. This value is converted in the internal unit of measure of the drive using the ***Acceleration factor numerator*** (registers 040Ah, 040Bh) and the ***Acceleration factor denominator*** (registers 040Ch, 040Dh) as showed in the formula below:

$$Drive\_Jerk = \frac{(AccelerationfactorNumerator \times Jerk)}{(AccelerationFactorDenominator)}$$

Using the default value of this factors, i.e. *Numerator*=65536 and *Denominator*=60, the ***jerk*** is expressed on the Caliper in [rpm/s^2].

3.12 Gearbox mode

The control type ***Gearbox*** is selected when the parameter "***control type***" (Register 0401h) is set to 2.

3.12.1 Control word and Status word In Gearbox mode

In this paragraph are described the structures of the *Control word* and the *Status word* when the ***Gearbox*** mode is selected.

3.12.1.a Setting of the Control word bits in Gearbox mode

<table>
<tr><th>Bit</th><th colspan="2">Description</th></tr>
<tr><td>0</td><td colspan="2">Switch on</td></tr>
<tr><td>1</td><td colspan="2">Enable operation</td></tr>
<tr><td>2</td><td colspan="2">Fault reset</td></tr>
<tr><td>3</td><td colspan="2">Halt</td></tr>
<tr><td rowspan="4">4</td><td colspan="2">Enable Axis</td></tr>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>The drive is not engaged with the reference of the master</td></tr>
<tr><td>1</td><td>The drive is commanded to engage with the reference of the master</td></tr>
<tr><td rowspan="4">5</td><td colspan="2">Reset encoder</td></tr>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>No function</td></tr>
<tr><td>1</td><td>Reset the counter of the pulses of the master encoder (it can’t be used in Operational state)</td></tr>
<tr><td rowspan="4">6</td><td colspan="2">Reset position</td></tr>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>No function</td></tr>
<tr><td>1</td><td>Reset the measured position of the motor (it can’t be used in Operational state)</td></tr>
<tr><td rowspan="3">7</td><td colspan="2">Position shift</td></tr>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0 ->1</td><td>when this bit switches from 0 to 1 the motor is commanded to shift position with respect the master encoder . This shift depends on the parameter Phase Shift (registers 0444h,0445h)</td></tr>
<tr><td rowspan="4">8</td><td colspan="2">Shift speed +</td></tr>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Function shift speed + disabled</td></tr>
<tr><td>1</td><td>Function shift speed + enabled: we can manually increase the number of the encoder pulses that the slave receives from the master. The slave “will think” that the master is moving faster,then it will also speed up and therefore their mutual position will shift</td></tr>
<tr><td rowspan="4">9</td><td colspan="2">Shift speed -</td></tr>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Function shift speed - disabled</td></tr>
<tr><td>1</td><td>Function shift speed - enabled: we can manually decrease the number of the encoder pulses that the slave receives from the master. The slave “will think” that the master is moving slower,then it will also slow down and therefore their mutual position will shift</td></tr>
<tr><td rowspan="4">10</td><td colspan="2">Jog -</td></tr>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Function Jog - disabled</td></tr>
<tr><td>1</td><td>Function Jog - enabled: the motor will rotate at the speed set on registers 0612h, 0613h</td></tr>
<tr><td rowspan="4">11</td><td colspan="2">Jog +</td></tr>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Function Jog + disabled</td></tr>
<tr><td>1</td><td>Function Jog - enabled: the motor will rotate at the speed set on registers 0612h, 0613h as Jog-, but in the opposite direction</td></tr>
<tr><td rowspan="4">12</td><td colspan="2">Home position</td></tr>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>The Homing procedure is stopped</td></tr>
<tr><td>0 ->1</td><td>On the rising edge of this Bit the Homing procedure is started</td></tr>
<tr><td>13 ... 15</td><td colspan="2">Reserved</td></tr>
</table>

3.12.1.b Description of the Status word Bits in Gearbox mode

<table>
<tr><th>Bit</th><th colspan="2">Description</th></tr>
<tr><td>0</td><td colspan="2">Switch on</td></tr>
<tr><td>1</td><td colspan="2">Enable operation</td></tr>
<tr><td>2</td><td colspan="2">Rampa Stop</td></tr>
<tr><td>3</td><td colspan="2">Rampa Halt</td></tr>
<tr><td>4</td><td colspan="2">Fault</td></tr>
<tr><td rowspan="4">6</td><td colspan="2">Gearbox enabled</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>The drive is not engaged with the reference of the master</td></tr>
<tr><td>1</td><td>The drive could be in one of these mode: Engagement, Excecution or Disengagement</td></tr>
<tr><td rowspan="4">9</td><td colspan="2">Remote</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Remote control disabled</td></tr>
<tr><td>1</td><td>Remote control enabled</td></tr>
<tr><td rowspan="4">10</td><td colspan="2">Torque limit</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Torque limit not reached</td></tr>
<tr><td>1</td><td>Torque limit reached</td></tr>
<tr><td rowspan="4">11</td><td colspan="2">Zero speed</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>The motor is stopped</td></tr>
<tr><td>1</td><td>The motor is moving</td></tr>
<tr><td>13</td><td colspan="2">Warning</td></tr>
<tr><td>14</td><td colspan="2">Fault reaction</td></tr>
<tr><td>5,7,8,
12,15</td><td colspan="2">Reserved</td></tr>
</table>

On the rising edge of bit 4 (***Enable axis***) of the *control word* the Drive starts the engagement phase with the reference of the master and the bit 6 (***Gearbox enabled***) of the *status word* is set to 1 to show that the motor is moving. When the engagement phase is over the Drive follows the speed and the position of the master reference. When the bit 4 of the *control word* returns to 0, the bit 6 (***Gearbox enabled***) of the *status word* also will return to 0, but only when the disengagement phase is over, in order to show that the Drive is really disengaged from the master reference.

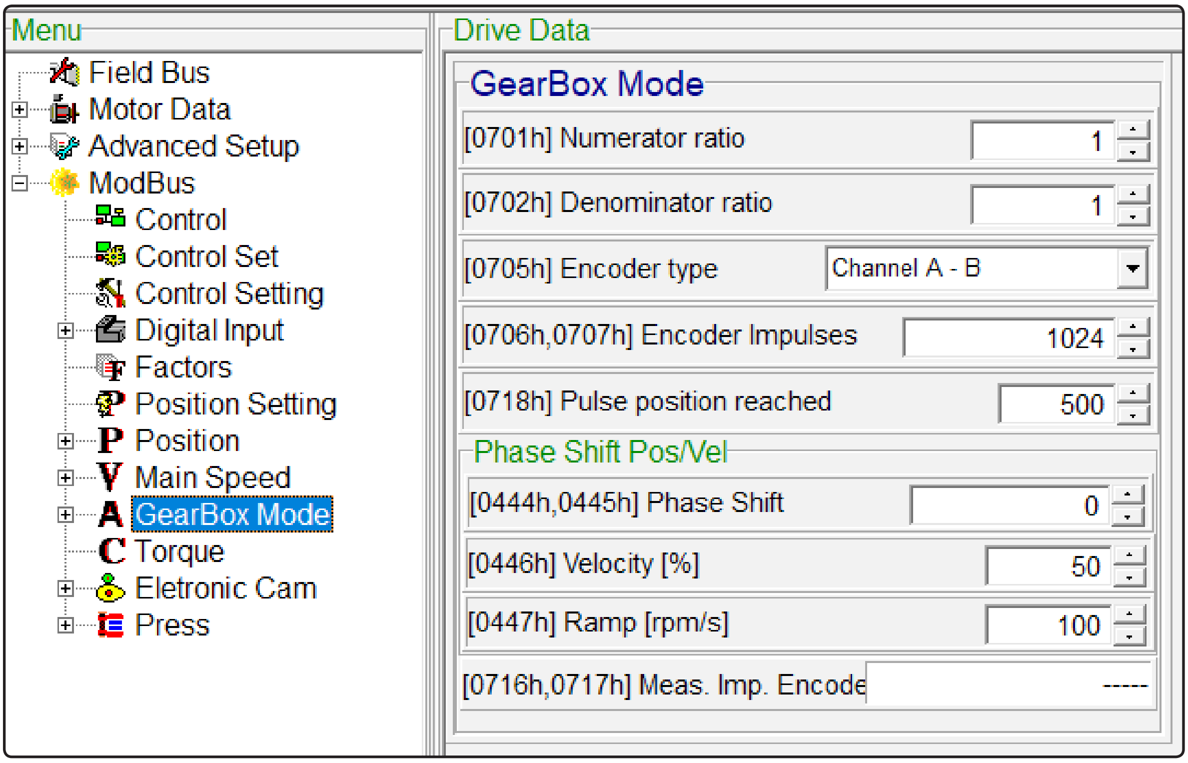
3.12.2 Gearbox parameter

This paragraph lists the control prameters used with the ***Gearbox*** mode. In ***Gearbox*** mode the Drive is controlling the motor position. The pulses received from the external encoder are multiplied by the ***Gearbox ratio numerator*** (register 0701h) and then divided by the ***Gearbox ratio denominator*** (register 0702h); the obtained result becomes the reference for the position control loop, which works with a measured position that is converted in internal increments of the drive, with the ratio 1 turn=65536 increments. To switch to the ***Gearbox*** mode the parameter ***Control type*** (register 0401h) must be set on 2. After that, to enable the ***Gearbox*** control the bit 4 of the *control word* must be switched from 0 to 1. The rising edge of this bit starts the engagement phase, which brings the motor to the same speed of the position reference calculated before.

When the engagement phase is over the Drive starts the phase of excecution, where it follows the position reference obtained by the external encoder. In this phase the parameters ***Max position error*** (registers 040Eh, 040Fh) and ***error time*** (register 0410h) set the error limits for the position control. The ***Gearbox*** control is disabled when the bit 4 of the *control word* from 1 to 0; this falling edge starts the disengagement phase whose task is to stop the motor axis in the selected way. The Bit 6 of the *status word* shows the operational status of the ***Gearbox*** control: if this bit is high the ***Gearbox*** control is active and it could be in the phase of engagement, execution or disengagement.

Address	Name	Type	Read	Write
0444h	Phase shift (MSB)	INT32	always	always
0445h	Phase shift (LSB)			
0446h	shift speed	UINT16	always	always
0447h	shift acceleration	UINT16	always	always
0701h	Gearbox ratio numerator	INT16	always	always
0702h	Gearbox ratio denominator	UINT16	always	always
0705h	Encoder type	UINT16	always	always
0706h	Encoder impulses (MSB)	UINT32	always	always
0707h	Encoder impulses (LSB)			
0716h	Measured impulses (MSB)	UINT32	always	No
0717h	Measured impulses (LSB)			
0718h	Pulses position reached	UINT16	always	always

These parameters can be set in the Caliper menu "***Gearbox Mode***" in which the following screen appears:



3.12.2.a Registers 0444h (1092 dec), 0445h (1093 dec) Phase shift

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0444h (MSB) **0445h (LSB)**	**Phase shift**	INT32	±FFFFFFFh	0	Count	Always	Always

These registers contain the number of pulses used to change the relative position between the motor and the reference of the master's encoder. the phase shift is perfromed on the rising edge of bit 7 of the *control word*. Setting this parameter to 65535 would lead to a phase shift of a turn of the slave's axis.

3.12.2.b Register 0446h (1094 dec) Shift speed

Address	Name	Type	Range	Default	Unit	Read	Write
0446h	**Shift speed**	UINT16	1..200	50	%	Always	Always

This register contains the speed with which the phase shift of the slave is performed. This value is expressed in percent of the speed measured in the instant when the phase shift was commanded..

3.12.2.c Register 0447h (1095 dec) Shift acceleration

Address	Name	Type	Range	Default	Unit	Read	Write
0447h	**Shift acceleration**	UINT16	1..60000	100	rpm/s	Always	Always

This register contains the acceleration reached by the slave in the initial and final stages of the phase shift.

3.12.2.d Register 0701h (1793 dec) Gearbox ratio numerator

Address	Name	Type	Range	Default	Unit	Read	Write
0701h	**Gearbox ratio numerator**	INT16	-32768... +32767	1		Always	Always

This register contains the numerator of the Gearbox ratio. This parameter is used together with the ***Gearbox ratio Denominator*** (register 0702h) to define the speed ratio between the master axis and the slave axis. In fact, when the ***Gearbox*** mode is used the number of pulses aquired from the master's encoder are multiplied for the ***Gearbox ratio numerator*** (register 0701h) and divided by the ***Gearbox ratio Denominator*** (register 0702h), and the result give us a position reference with a speed which could differs from the real one of the master.

3.12.2.e Register 0702h (1794 dec) Gearbox ratio denominator

Address	Name	Type	Range	Default	Unit	Read	Write
0702h	**Gearbox ratio denominator**	INT16	1.. +32767	1		Always	Always

This register contains the numerator of the Gearbox ratio (see the previous paragraph).

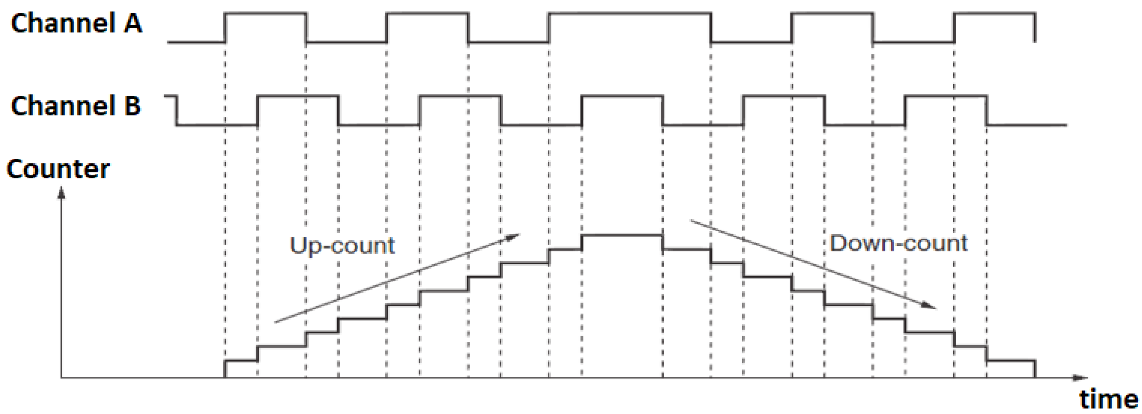
3.12.2.f Register 0705h (1797 dec) Encoder type

Address	Name	Type	Range	Default	Unit	Read	Write
0705h	**Encoder type**	INT16	0..2	0		Always	Always

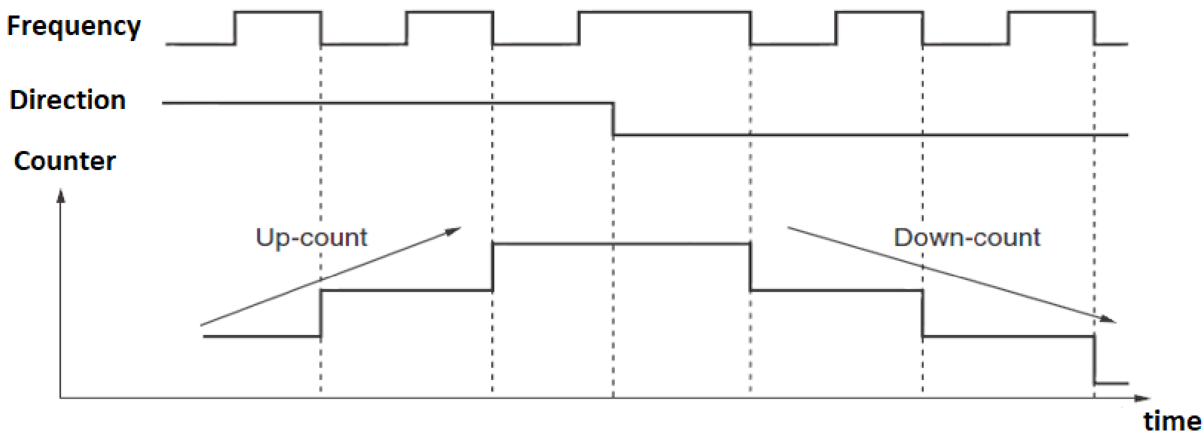
This register is used when the ***Gearbox*** mode is selected. The position and speed references are sent by the master as digital signals that could be of different types; this register defines what is the type of the signal received from the external encoder.

Value	***Encoder type Description***
0	***Channel A-B*** : the reference is obtained by two signals in quadrature
1	***Frequency-Direction***: One channel tells the direction of rotation, the other tells the speed
2	***Impulses CW CCW***: Depending on which channel is receiving the impulses, the motor will rotate in a sense or the other

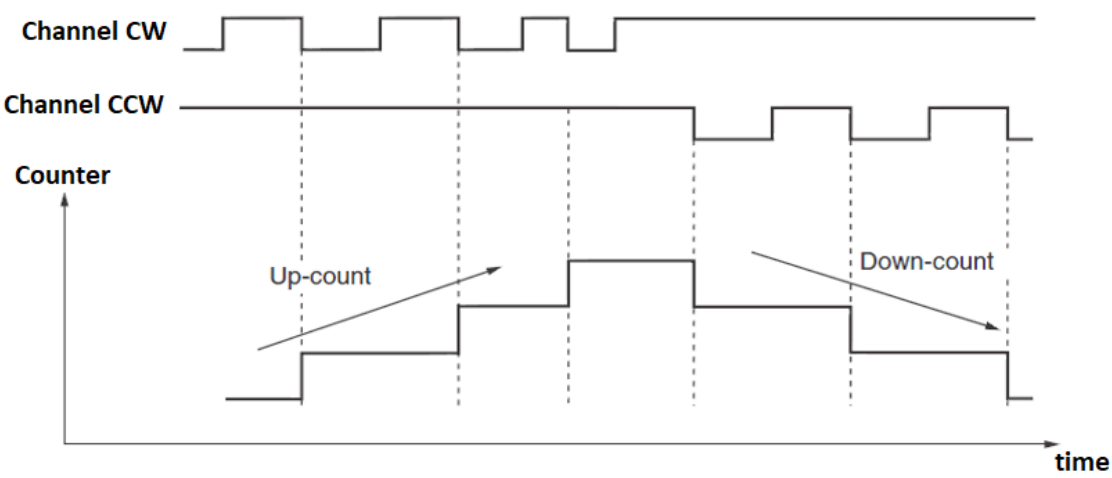
Encoder type "Channel A-B"



Encoder type "Frequency-Direction"



Encoder type "Impulses CW CCW"



If the "***channel A-B***" is selected, it must be taken into account that the Drive will count all the impulses of the two channels of the encoder, so the resolution on a single turn is increased 4 times with respect to the resolution declared by the encoder. For example, if the encoder declares to have 1024 impulses per turn, the Drive will count 4096 impulses per turn; this must be taken into accout when the ***Gearbox ratio numerator*** (register 0701h) and the ***Gearbox ratio denominator*** (registro 0702h) values are decided.

3.12.2.g Registers 0706h (1798 dec),0707h (1799 dec) Encoder impulses

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0706h (MSB) **0707h (LSB)**	**Encoder Impulses**	UINT32	64h … FFFFFFFFh	1024 (dec)		Always	Always

These registers contain the number of impulses per turn of the encoder. If the "***channel A-B***" is selected, inside the Drive the "***Encoder Impulses***" will be multiplied by 4, because in this mode the Drive can acquire a number of impulses from the encoder 4 times larger.

3.12.2.h Registers 0716h (1814 dec), 0717h (1815dec) Measured impulses

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0716h (MSB) **0717h (LSB)**	**Measured impulses**	INT32	±7FFFFFFFh		Impulsi encoder motore	Always	No

These registers contain the number of impulses read by the external encoder; this value can be reset to zero switching the bit 5 of the *control word* from 0 to 1, when the ***Gearbox*** function is not operational.

3.12.2.i Register 0718h (1816 dec) Pulses position reached

Address	Name	Type	Range	Default	Unit	Read	Write
0718h	**Pulses position reached**	UINT16	1..65535	500	Impulsi encoder motore	Always	Always

This registers contains a threshold expressed as a number of pulses. If the difference between the position reference of the master and the measured position of the slaves is smaller than this threshold, than the Output function "***Position reached***" is set to 1.The "***Position reached***" function can be associated to one of the Drive outputs in the menu "***Advanced setup-output***" of the Caliper. The pulses expressed on this register obey the proportion

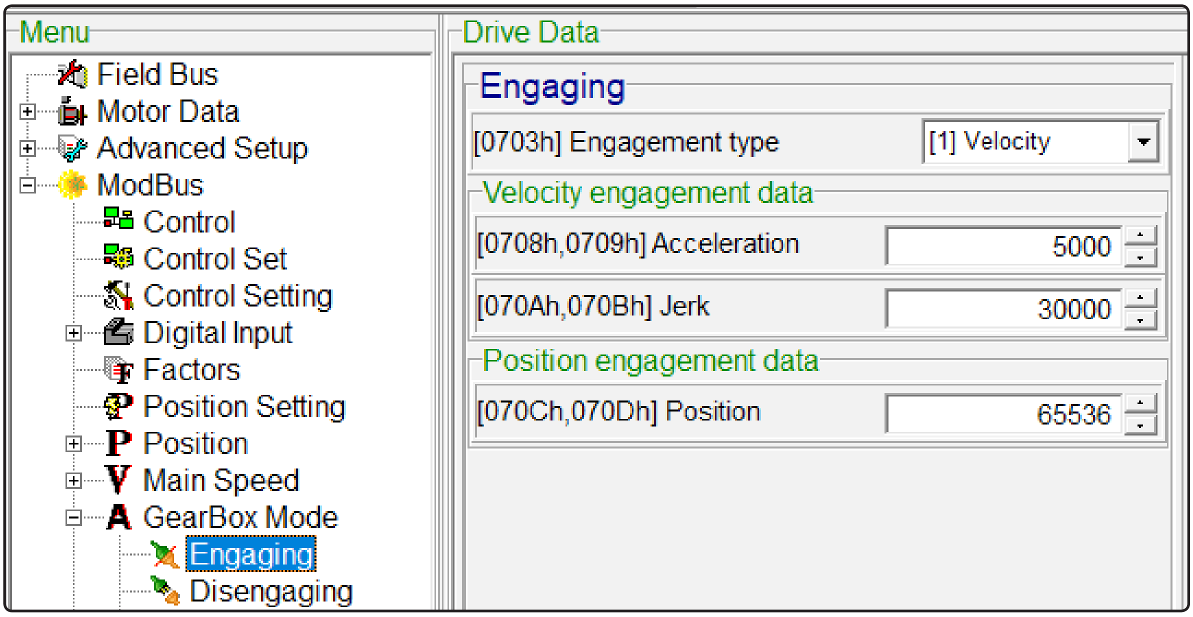
"*1 turn=65536 pulses*".

3.12.3 Gearbox-engagement parameters

This paragraph lists the parameters used during the engagement phase of the slave. This phase starts on the raising edge of the bit 4 of the *control word*, and is used to bring the slave at the same speed of the calculated position reference. Once this phase is terminated the Drive goes in the phase of Excecution, where the slaves rigidly follow the position reference of the master.

Address	Name	Type	Read	Write
0703h	engaging type	INT16	always	always
0708h	engaging acceleration (MSB)	UINT32	always	always
0709h	engaging acceleration (LSB)			
070Ah	engaging jerk (MSB)	UINT32	always	always
070Bh	engaging jerk (LSB)			
070Ch	engaging position (MSB)	INT32	always	always
070Dh	engaging position (LSB)			

These parameters can be set in the Caliper menu "***Gearbox Mode-Engaging***" in which the following screen appears:



3.12.3.a Register 0703h (1795 dec) Engaging type

Address	Name	Type	Range	Default	Unit	Read	Write
0703h	**Engaging type**	INT16	0..2	1		Always	Always

The ***Engaging type*** is used to select how the Drive will perform the phase of engagement to the calculated position reference.

Value	Engaging type	Description
0	Immediate engagement	The position control starts immediately to follow the position reference sent by the external encoder. This mode is suggested when the ***Measured impulses*** (0716h,0717h) of the external encoder at the starting of the ***Gearbox*** Control are zero.
1	Speed engagement	The Drive speeds up with a ramp of speed until it reaches the speed of the position reference. The engagement is performed according to the parameters ***Engaging Acceleration*** (registers 0708h, 0709h) and ***Engaging Jerk*** (registers 070Ah, 070Bh).
2	Position engagement	The Drive reaches the speed of the position reference with a speed ramp. During this speed ramp the motor covers the space set on the parameter ***Engaging position*** (registers 070Ch, 070Dh); The position curve followed by the Drive durign the phase of engagement is calculated using a polinomial function of the 5th order.

3.12.3.b Registers 0708h (1800 dec) 0709h (1801 dec) Engaging acceleration

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0708h (MSB) **0709h (LSB)**	**Engaging acceleration**	UINT32	1 ... 7FFFFFFFh	5000 (dec)		Always	Always

These registers contains the acceleration reached by the slave during the "***Speed engagement***". Inside the drive it's expressed as [increments/s^2]. This value is converted in the internal unit of measure of the drive using the ***Acceleration factor numerator*** (registers 040Ah, 040Bh) and the ***Acceleration factor denominator*** (registers 040Ch, 040Dh) as showed in the formula below:

$$Drive\_engagement\_accel = \frac{(AccelerationFactorNumerator \times engagement\_accel)}{(AccelerationFactorDenominator)}$$

Using the default values of the ***Acceleration factors***, ie *Numerator*=65536 and *Denominator*=60, the ***Engaging acceleration*** is expressed in [rpm/s]

Note: one turn of the motor correspond to 65536 internal increments.

3.12.3.c Registers 070Ah (1802 dec) 070Bh (1803 dec) engaging jerk

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
070Ah (MSB) **070Bh (LSB)**	**engaging jerk**	UINT32	0..,5BFFFFh	30000 (dec)		Always	Always

These registers contain the ***Engaging jerk***, a value used to give to the speed a parabolic profile at the start and the end of the speed ramp during the "***Speed engagement***". An high value of Jerk means that the acceleration variation are bigger, which means a lower accleration time but with higher mechanical stress; vice versa, A low value of Jerk means that the acceleration variation are lower, which means an higher accleration time but with lower mechanical stress. The unit of measure of the jerk is "acceleration divided by time"; inside the drive it's expressed as [increments/ s^3]. This value is converted in the internal unit of measure of the drive using the ***Acceleration factor numerator*** (registers 040Ah, 040Bh) and the ***Acceleration factor denominator*** (registers 040Ch, 040Dh) as showed in the formula below:

$$Drive\_Engagement\_Jerk = \frac{(AccelerationFactorNumerator \times Engagement\_Jerk)}{(AccelerationFactorDenominator)}$$

Using the default value of this factors, i.e. *Numerator*=65536 and *Denominator*=60, the ***engaging jerk*** is expressed on the Caliper in [rpm/s^2].

3.12.3.d Registers 070Ch (1804 dec) 070Dh (1805 dec) engaging position

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
070Ch (MSB) **070Dh (LSB)**	**engaging position**	INT32	0 + 7FFFFFFFh	65536 (dec)		Always	Always

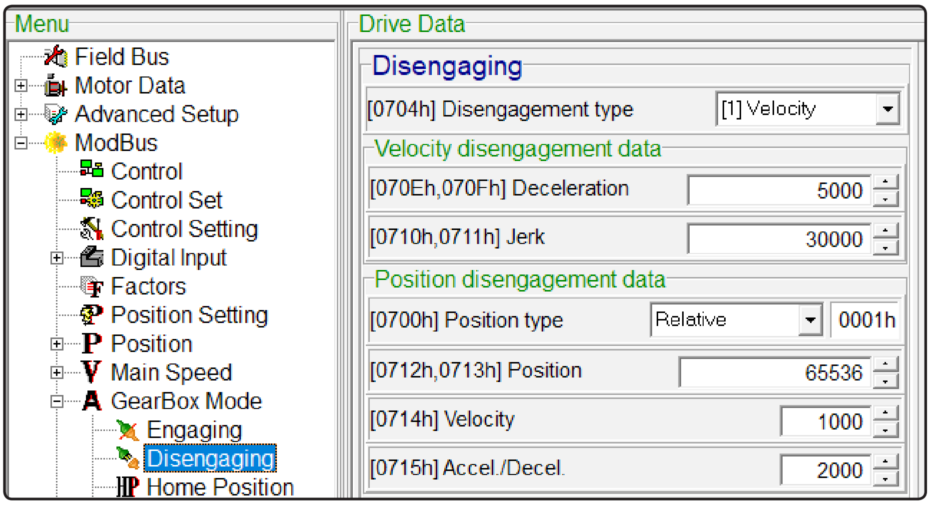
When ***Engaging type*** (register 0703h) is set on "***position engagement***", this parameter represents the space that the slave is going to cover while it's accelerating up to the speed of the position reference. This space corresponds to the one covered by the master during the phase of engagement multiplied by the Gearbox ratio

3.12.4 Gearbox- disengagement parameters

This paragraph lists the parameters used during the disengagement phase of the slave. This phase is started by the falling edge (from 1 to 0) of bit 4 of the *control word*, and is used to slow down the slave until it stops using the selected modality. Once this phase is over the bit 6 of the *Status word* is set to 0 and the motor remains still, holding the torqe.

Address	Name	Type	Read	Write
0700h	Position disengaging type	INT16	Always	Always
0704h	Disengaging type	INT16	Always	Always
070Eh	Disengaging deceleration (MSB)	UINT32	Always	Always
070Fh	Disengaging deceleration (LSB)			
0710h	Disengaging jerk (MSB)	UINT32	Always	Always
0711h	Disengaging jerk (LSB)			
0712h	Disengaging position (MSB)	INT32	Always	Always
0713h	Disengaging position (LSB)			
0714h	Pos. disengaging speed	UINT16	Always	Always
0715h	Pos. disengaging deceleration	UINT16	Always	Always

These parameters can be set in the Caliper menu "***Gearbox Mode-disengaging***" in which the following screen appears:



3.12.4.a Register 0700h (1792 dec) Position disengaging type

Address	Name	Type	Range	Default	Unit	Read	Write
0700h	**Position disengaging type**	INT16	0 = absolute	1		Always	Always
			1 = relative				

This register sets if the ***disengaging position*** (registers 0712h,0713h) should be considered an absolute value with respect to the ***home position*** or a ***relative*** value with respect to the position measured when the disengagement started.

3.12.4.b Register 0704h (1796 dec) Disengaging type

Address	Name	Type	Range	Default	Unit	Read	Write
0704h	**disengaging type**	INT16	0..2	1		Always	Always

This register sets the ***disengaging type*** , which decide how the Drive will pass from the phase where the motor is executing the position reference to phase where the motor stands still. The disengagement starts when the ***Gearbox*** function is disabled, i.e. when the bit 4 of the *controlword* is switched from 1 to 0.

Value	disengaging type	Description
0	Immediate disengagement	the motor will stop immediately when the ***Gearbox*** function is disabled. *Note: this disengagement is really abrupt, the motor is going to stop with all its torque, which will lead to high mechanical stress in the apparatus*
1	Speed disengagement	The motor will stop with a ramp of speed. the disengagement is perfomed using the set values of ***Disengaging deceleration*** (registers 070Eh,070Fh) and ***Disengaging Jerk*** (registers 0710h,0711h)
2	Position disengagement	The Drive stops following a position trajectory obtained by a polynomial function of the 4th order. The parameters used to set the disengagement phase are ***Position disengaging type*** (register 0700h) which decide if the ***disengaging position*** is absolute or relative, ***Pos. disengaging speed*** (register 0714h) which sets the maximum speed reached in the disengagement phase, ***Pos. disengaging deceleration*** (register 0715h) which sets the maximum acceleration and the ***disengaging position*** (registers 0712h, 0713h). If ***Disengaging position*** is an ***absolute*** value, then it represents the position that the motor will reach before stopping. If ***Disengaging position*** is a ***relative*** value, then it represents the space traveled during the disengagement phase.

3.12.4.c Registers 070Eh (1806 dec) 070Fh (1807 dec) Disengaging deceleration

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
070Eh (MSB) **070Fh (LSB)**	**Disengaging deceleration**	UINT32	1 ... 7FFFFFFFh	5000 (dec)		Always	Always

These registers contains the deceleration reached by the slave during the "***Speed disengagement***". Inside the drive it's expressed as [increments/s^2]. This value is converted in the internal unit of measure of the drive using the ***Acceleration factor numerator*** (registers 040Ah, 040Bh) and the ***Acceleration factor denominator*** (registers 040Ch, 040Dh) as showed in the formula below:

$$drive\_disengagement\_decel = \frac{(AccelerationfactorNumerator \times disengagement\_decel)}{(AccelerationFactorDenominator)}$$

Using the default values of the ***Acceleration factors***, ie *Numerator*=65536 and *Denominator*=60, the ***Disengaging deceleration*** is expressed in [rpm/s]

Note: one turn of the motor correspond to 65536 internal increments.

3.12.4.d Registers 0710h (1808 dec) 0711h (1809 dec) Disengaging jerk

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0710h (MSB) **0711h (LSB)**	**Disengaging jerk**	UINT32	1 ... 7FFFFFFFh	30000 (dec)		Always	Always

These registers contain the ***Disengaging jerk***, a value used to give to the speed a parabolic profile at the start and the end of the speed ramp during the "***Speed disengagement***". An high value of Jerk means that the acceleration variation are bigger, which means a lower accleration time but with higher mechanical stress; vice versa, A low value of Jerk means that the acceleration variation are lower, which means an higher accleration time but with lower mechanical stress. The unit of measure of the jerk is "acceleration divided by time"; inside the drive it's expressed as [increments/ s^3]. This value is converted in the internal unit of measure of the drive using the ***Acceleration factor numerator*** (registers 040Ah, 040Bh) and the ***Acceleration factor denominator*** (registers 040Ch, 040Dh) as showed in the formula below:

$$drive\_disengagement\_Jerk = \frac{(AccelerationFactorNumerator \times disengagement\_Jerk)}{(AccelerationFactorDenominator)}$$

Using the default value of this factors, i.e. *Numerator*=65536 and *Denominator*=60, the ***disengaging jerk*** is expressed on the Caliper in [rpm/s^2].

3.12.4.e Registers 0712h (1810 dec) 0713h (1811 dec) disengaging position

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0712h (MSB) **0713h (LSB)**	**Disengaging position**	INT32	0 .. 7FFFFFFFh	65536 (dec)		Always	Always

Disengaging Position represents the position reached at the end of the ***Position disengagement*** phase. The value of this parameter could be considered absolute or relative, depending on the value set on parameter "***Position disengaging type***" (register 0700h). If ***Disengaging position*** is considered relative, then the motor will stop on the position obtained by the sum of ***Disengaging position*** and the position measured when the disengagement started. If ***Disengaging position*** is considered absolute, then the motor will stop exactly on the ***Disengaging position***.

Note: the value of the Measured position could be reset to 0 switching the bit 6 of the controlword from 0 to 1,when the Gearbox function is not active.

3.12.4.f Register 0714h (1812 dec) Pos. disengaging speed

Address	Name	Type	Range	Default	Unit	Read	Write
0714h	**Pos. disengaging speed**	UINT16	0..65536	1000		Always	Always

This register contains the ***Pos. disengaging speed***, i.e. the maximum speed reached while the ***Position disengagement*** is performed.

3.12.4.g Register 0715h (1813 dec) Pos. disengaging deceleration

Address	Name	Type	Range	Default	Unit	Read	Write
0715h	**Pos. disengaging deceleration**	UINT16	0..65536	2000		Always	Always

This register contains the ***Pos. disengaging deceleration***, i.e. the deceleration applied while the ***Position disengagement***.

3.12.5 Gearbox-Jog parameters

The ***Jog*** functioning when the drive is set on ***Gearbox*** Mode is the same as the one described on paragraph *"3.10.4 Position Jog Parameters" pag. 58* when the drive is set on ***Positioner*** Mode. The used parameters are the same.

3.12.6 Gearbox- Home position

The ***Home position*** functioning when the drive is set on ***Gearbox*** Mode is the same as the one described on paragraph *"3.10.5 Home Position" pag. 60* when the drive is set on ***Positioner*** Mode. The used parameters are the same.

3.13 Torque mode

The control type ***Torque*** is selected when the parameter "***control type***" (Register 0401h) is set to 1. This control use only the quadrature current loop regulator,which controls directly the motor torque.

3.13.1 Controlword and Statusword

In this paragraph are described the structures of the *Control word* and the *Status word* when the ***Torque*** mode is selected.

3.13.1.a Setting of theControlword bits in Torque mode

Bit	Description
0	***Switch on***
1	***Enable operation***
2	***Fault reset***
3	***Halt***
4..15	***Manufacturer specific***

3.13.1.b Description of the Statusword bits in Torque mode

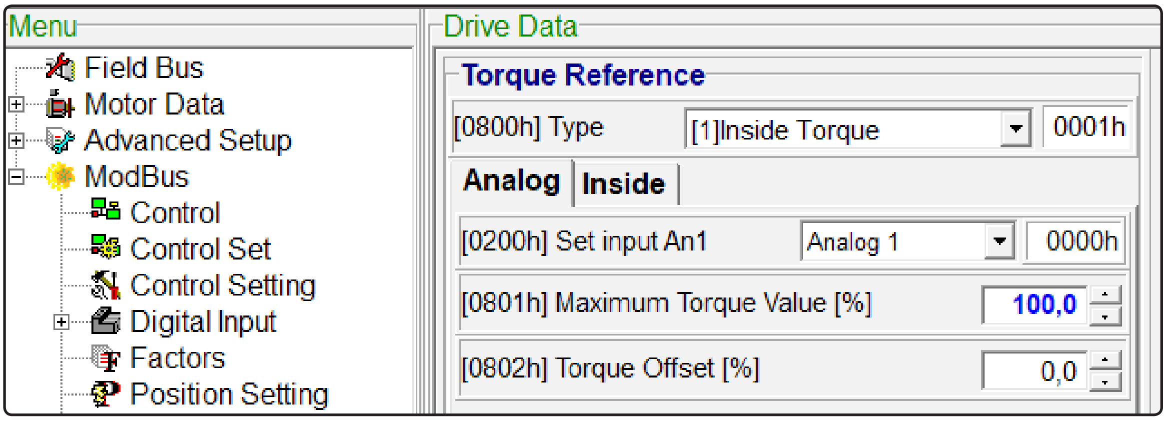
<table>
<tr><th>Bit</th><th colspan="2">Description</th></tr>
<tr><td>0</td><td colspan="2">Switch on</td></tr>
<tr><td>1</td><td colspan="2">Enable operation</td></tr>
<tr><td>2</td><td colspan="2">Rampa Stop</td></tr>
<tr><td>3</td><td colspan="2">Rampa Halt</td></tr>
<tr><td>4</td><td colspan="2">Fault</td></tr>
<tr><td>5..8</td><td colspan="2">Manufacturer specific</td></tr>
<tr><td rowspan="4">9</td><td colspan="2">Remote</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Remote control disabled</td></tr>
<tr><td>1</td><td>Remote control enabled</td></tr>
<tr><td rowspan="4">10</td><td colspan="2">Torque Limit</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Torque limit not reached</td></tr>
<tr><td>1</td><td>Torque limit reached</td></tr>
<tr><td rowspan="4">11</td><td colspan="2">Zero speed</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>The motor is stopped</td></tr>
<tr><td>1</td><td>The motor is moving</td></tr>
<tr><td>12</td><td colspan="2">Manufacturer specific</td></tr>
<tr><td>13</td><td colspan="2">Warning</td></tr>
<tr><td>14</td><td colspan="2">Fault Reaction</td></tr>
<tr><td>15</td><td colspan="2">Manufacturer specific</td></tr>
</table>

3.13.2 Torque control

This paragraph lists the control parameters with the ***Torque*** mode.

Address	Name	Type	Read	Write
0800h	Torque reference Type	INT16	Always	Always
0801h	Maximum torque value	UINT16	Always	Always
0802h	Torque offset	INT16	Always	Always
0803h	Inside torque	INT16	Always	Always

These parameters can be set in the Caliper menu "***Torque***" in which the following screen appears:



Note: in this menu you can also set the bit 2 of register 200h, the use of which was already described in paragraph *"3.08.1 Register 0200h (512 dec) , Bit 2, 3 ,11 – Set input An1,An2 e An3" pag. 35*

3.13.2.a Register 0800h (2048 dec) Torque reference type

Address	Name	Type	Range	Default	Unit	Read	Write
0800h	**Torque reference type**	INT16	0,1	1		Always	Always

The parameter "***Torque reference type***" allows you to choose what reference of torque will be used by the drive when it's set on "***Torque***" mode

Value	Description
0	***Analog Torque***: the drive takes as torque reference the signal received by the analogic input +-10 [V]
1	***Inside Torque***: the drive takes as torque reference the value wrote on register 0803h

3.13.2.b Register 0801h (2049 dec) Maximum torque value

Address	Name	Type	Range	Default	Unit	Read	Write
0801h	**Maximum torque value**	UINT16	0..,2500	1000		Always	Always

This register contains the ***Maximum Torque Value***, i.e. the Full scale of the main analog input which receives the torque reference as a signal. The received torque reference is converted inside of the Drive in a current reference (in Ampere) using the formula below:

$$Drive\_reference\_current = (Nominal\_current) \times (\frac{(Max\_torque\_value)}{1000}) \times (\frac{(volt\_input)}{10})$$

3.13.2.c Register 0802h (2050 dec) Torque offset

Address	Name	Type	Range	Default	Unit	Read	Write
0802h	**Torque offset**	INT16	-1000 ... +1000	0		Always	Always

This register contains the ***Torque offset***, i.e. a Torque value that is added or subtracted form the torque reference received by the main analog input. The torque reference received is then converted inside the drive in a current reference (in Ampere). The formula to convert this torque offset in a current offset is showed below:

$$Drive\_current\_offset = \frac{[Nominal\_current \times (analog\_torque\_offset)]}{1000}$$

3.13.2.d Register 0803h (2051 dec) Inside torque

Address	Name	Type	Range	Default	Unit	Read	Write
0803h	**Inside torque**	INT16	-2500 ... + 2500	0		Always	Always

This register contains the value used as the torque reference for the control when the Drive was set on "***Inside Torque***" mode. The torque reference is expressed in thousandth of the ***Nominal Current*** (register 0153h): for example, if the ***Inside torque*** (register 0803h) is equal to 1000, then the motor will receives a torque current equal to the nominal current. To calculate the current reference it's used the formula below:

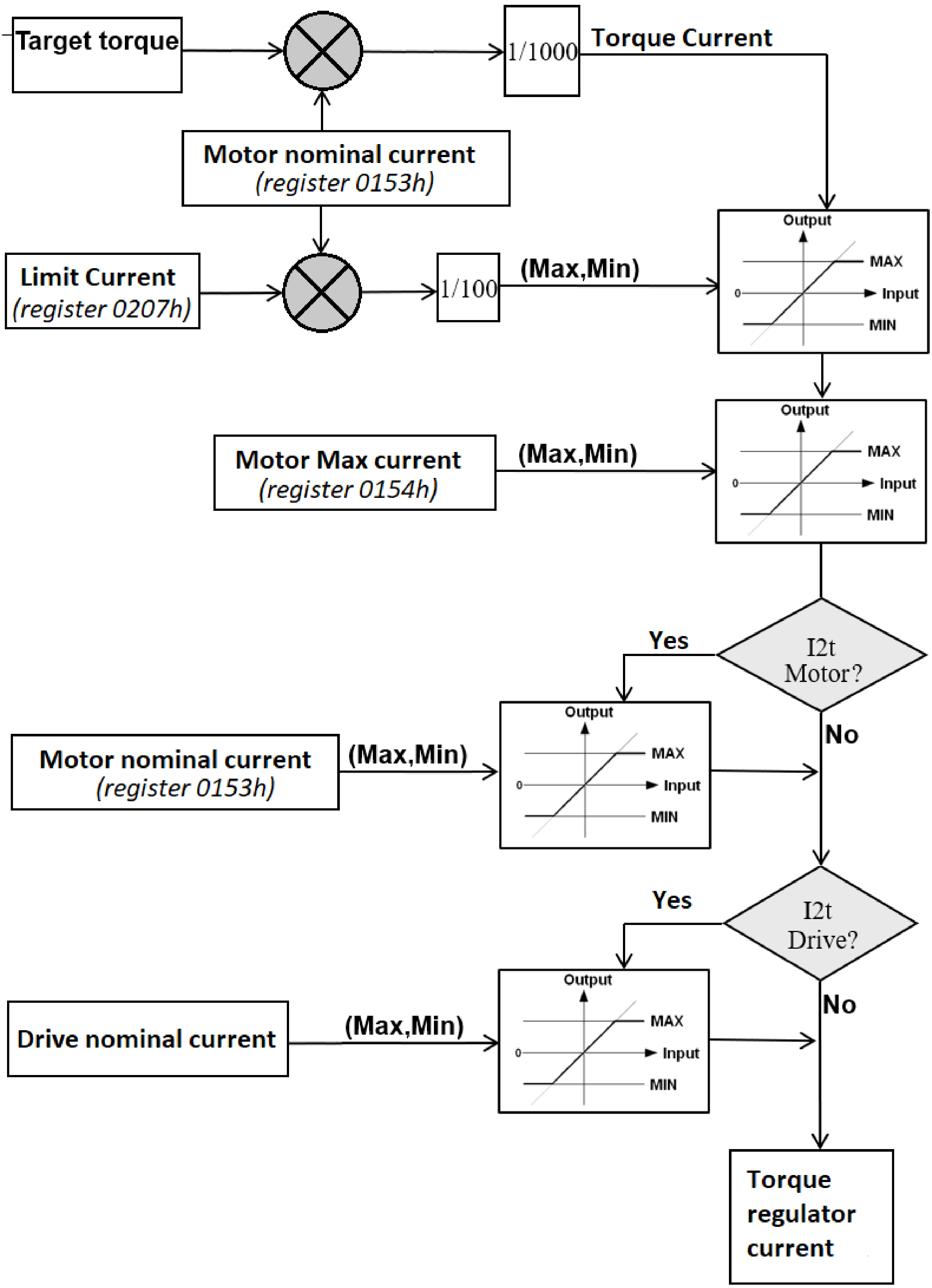
$$drive\_reference\_current = (Nominal\_current) \times (\frac{Inside\_torque}{1000})$$

3.13.2.e Limits of the torque current

The value of the torque current has several limits imposed by other parameters inside the Drive.

The first limit is imposed by the current obtained multiplying the ***Current Limit*** (register 0207h) with the ***Motor Nominal Current*** (register 0153h) and then dividing the result by 1000. The current obtained with this formula represents the maximum limit that the torque current must not exceed, while the minimum limit is equal to the negative of the maximum limit. After that the resulting torque current must not exceed the limits imposed by the ***Motor max current*** (register 0154h) and by the maximum current that the Drive could erogate, which is an unchangeable parameter inside of the drive.

If the I2t alarm of the motor occur, the maximum torque current is limited to the ***Motor nominal current*** (register 0153h) ; in a similar way, If the I2t alarm of the Drive occur, the maximum torque current is limited to the maximum current that the Drive could erogate.


Target torque
1/1000
Torque Current
Motor nominal current
(register 0153h)
Limit Current
(register 0207h)
1/100
(Max,Min)
Output
MAX
0
Input
MIN
Output
MAX
0
Input
MIN
Motor Max current
(register 0154h)
(Max,Min)
Yes
I2t
Motor?
No
Motor nominal current
(register 0153h)
(Max,Min)
Output
MAX
0
Input
MIN
Yes
I2t
Drive?
No
Drive nominal current
(Max,Min)
Output
MAX
0
Input
MIN
Torque
regulator
current

3.14 Electronic Cam mode

The control type ***Electronic Cam*** is selected when the parameter "***control type***" (Register 0401h) is set to 4.

3.14.1 Controlword and Statusword

In this paragraph are described the structures of the *Control word* and the *Status word* when the ***Electronic cam*** mode is selected.

3.14.1.a Setting of the Controlword bits in Electronic Cam mode

<table>
<tr><th>bit</th><th colspan="2">description</th></tr>
<tr><td>0</td><td colspan="2">Switch on</td></tr>
<tr><td>1</td><td colspan="2">Enable operation</td></tr>
<tr><td>2</td><td colspan="2">Fault reset</td></tr>
<tr><td>3</td><td colspan="2">Halt</td></tr>
<tr><td rowspan="4">4</td><td colspan="2">Start cam</td></tr>
<tr><td>value</td><td>description</td></tr>
<tr><td>0</td><td>the Drive is not engaged to the master reference</td></tr>
<tr><td>1</td><td>the Drive is commanded to engage to the master reference</td></tr>
<tr><td rowspan="4">5</td><td colspan="2">Jog -</td></tr>
<tr><td>value</td><td>description</td></tr>
<tr><td>0</td><td>Function Jog- disabled</td></tr>
<tr><td>1</td><td>Function Jog- enabled: the motor moves with the speed set on registers 0612h, 0613h. Note: the function Jog is active only when the motor is not excecuting a position curve</td></tr>
<tr><td rowspan="4">6</td><td colspan="2">Jog +</td></tr>
<tr><td>value</td><td>description</td></tr>
<tr><td>0</td><td>Function Jog+ disabled</td></tr>
<tr><td>1</td><td>Function Jog+ enabled: the motor moves in the opposite direction of Jog-, with the speed set on registers 0612h, 0613h. Note: the function Jog is active only when the motor is not excecuting a position curve</td></tr>
<tr><td rowspan="4">7</td><td colspan="2">Home position</td></tr>
<tr><td>value</td><td>description</td></tr>
<tr><td>0->1</td><td>The research of the home position starts on the rising edge of this bit</td></tr>
<tr><td>0</td><td>The research of the home position stops</td></tr>
<tr><td rowspan="4">8</td><td colspan="2">Bit0 - selection</td></tr>
<tr><td>value</td><td>description</td></tr>
<tr><td>0</td><td rowspan="2">this bit is used together with another 2 bits to select the index of the cam that we want to excecute</td></tr>
<tr><td>1</td></tr>
<tr><td rowspan="4">9</td><td colspan="2">Bit1 - selection</td></tr>
<tr><td>value</td><td>description</td></tr>
<tr><td>0</td><td rowspan="2">this bit is used together with another 2 bits to select the index of the cam that we want to excecute</td></tr>
<tr><td>1</td></tr>
<tr><td rowspan="4">10</td><td colspan="2">Bit 2 - selection</td></tr>
<tr><td>value</td><td>description</td></tr>
<tr><td>0</td><td rowspan="2">this bit is used together with another 2 bits to select the index of the cam that we want to excecute</td></tr>
<tr><td>1</td></tr>
</table>

bit	description	
11	***speed shift +***	
	value	**description**
	0	Function shift speed + disabled
	1	Function shift speed + enabled: we can manually increase the number of the encoder pulses that the slave receives from the master. The slave “will think” that the master is moving faster,then it will also speed up and therefore their mutual position will shift
12	***speed shift -***	
	value	**description**
	0	Function shift speed - disabled
	1	Function shift speed - enabled: we can manually decrease the number of the encoder pulses that the slave receives from the master. The slave “will think” that the master is moving slower, then it will also slow down and therefore their mutual position will shift
13... 15	***Manufacturer specific***	

3.14.1.b Meaning of the statusword bits in Electronic Cam mode

bit	Description	
0	***Switch on***	
1	***Enable operation***	
2	***Ramp Stop***	
3	***Ramp Halt***	
4	***Fault***	
5	***Cam execution***	
	Value	**Description**
	0	The Drive is not engaged with the master reference
	1	The drive coul be in one of this phase with respect to the master reference: engagement, excecution, disengagement
6	***Manufacturer specific***	
7	**Homing performed**	
	Value	**Description**
	0	Research of the home position not executed
	1	Research of the home position executed
8	**Position Error**	
	Value	**Description**
	0	Position error not detected
	1	Position error detected
9	**Remote**	
	Value	**Description**
	0	Remote control disabled
	1	Remote control enabled
10	***Torque limit***	
	Value	**Description**
	0	Torque limit not reached
	1	Torque limit reached

<table>
<tr><th>bit</th><th colspan="2">Description</th></tr>
<tr><td rowspan="4">11</td><td colspan="2">Zero speed</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>The motor is stopped</td></tr>
<tr><td>1</td><td>The motor is moving</td></tr>
<tr><td>12</td><td colspan="2">Manufacturer specific</td></tr>
<tr><td>13</td><td colspan="2">Warning</td></tr>
<tr><td>14</td><td colspan="2">Fault reaction</td></tr>
<tr><td>15</td><td colspan="2">Speed reached</td></tr>
</table>

On the rising edge of the bit 4 (***Start cam***) of the *control word* the engagement phase to the master reference is started and the bit 5 (***Cam execution***) of the *status word* is raised to 1 to show that the axis is moving. Once the engagement phase is over the drive excecutes the position curve obtained by the drawed cam and the master reference.when the bit 4 of the *status word* returns to 0, the bit 5 (***Cam execution***) of the *status word* returns to 0, but only after the end of the disengagement phase, to show that the drive is truly disengaged from the master reference.

3.14.2 Electronic cam parameters

Essentially, the ***Electronic cam*** functioning can be divided in 3 phases:

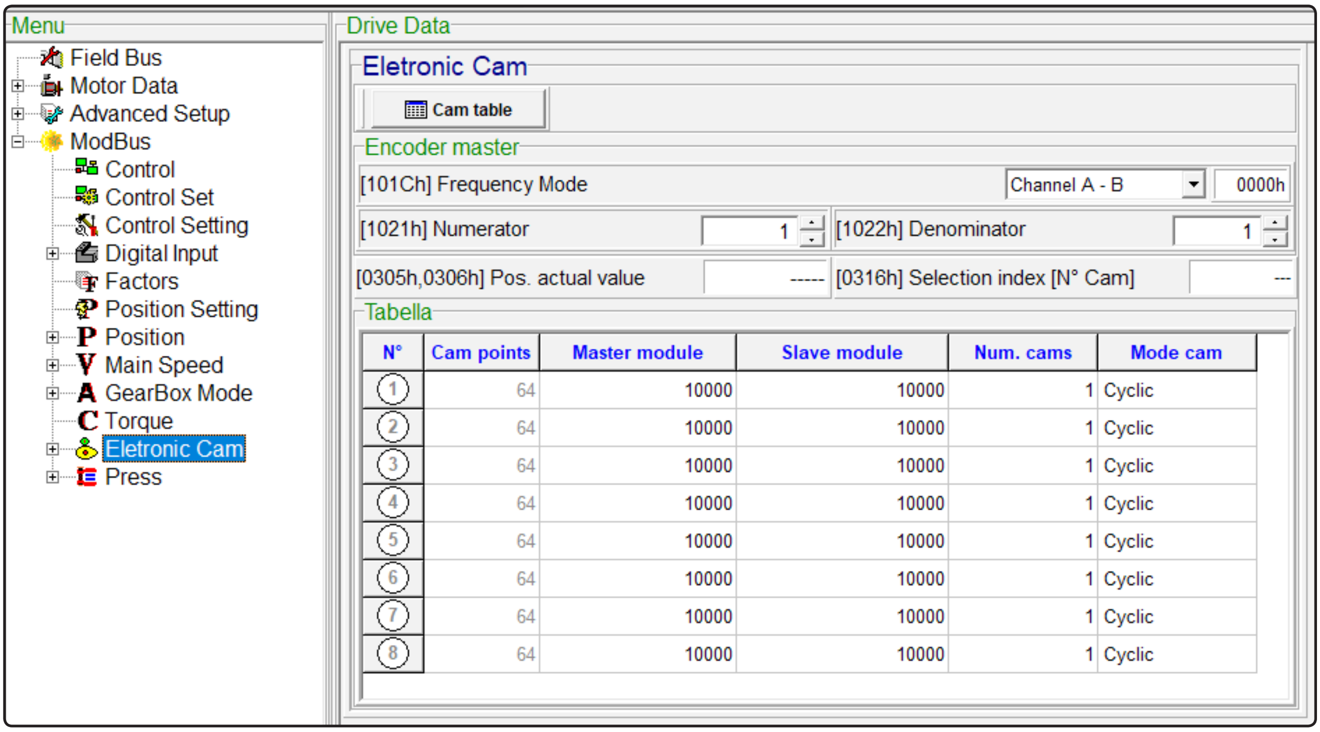
1. ***Cam engagement***: during this phase the slave speeds up until it reaches the speed of the master reference, and after that it could start excecuting the electronic cam;
2. ***Cam execution***: during this phase the slave executes the position profile obtained by points set on the cam table. This table maps the reference position received by the master with the target position that the slave must reach. By putting several points together, the profile of the cam is drawn. The electronic cam could be executed in one of the following ways:
   - "***Cyclic***": the electronic cam will be endlessly repeated, to stop it the bit 5 of the status word has to be lowered to 0. After the stop the disengagement phase starts, the axis will be stopped with the method set on the parameter "***Cyclic disengage mode***".
   - "***Acyclic***": in this mode the electronic cam will be repeated only for the number of cycles set on the parameter "***Num cams***". After the end of the last cam, the disengagement phase starts automatically in the way set on the parameter "***Acyclic disengage mode***". To make the slave start again you have to switch the bit 5 of the *Status word* to 0 and then switch it again to 1.
3. ***Cam disengagement***: during this phase the slave is disengaged from the master reference and is stopped with the mode set on the parameters previously cited.

With the ***electronic cam*** mode you can also perform the functions of ***Home position*** and ***Jog***, in order to repostion the slave axis. This table lists the parameters used to command the motor in the ***Electronic Cam*** mode.

Address	Name	Type	Read	Write
316	Selection index (N° cam)	UINT16	Always	No
101C	Electronic Cam - flag	INT16	Always	Always
101D	Mode cam - flag	INT16	Always	Always
1021	Encoder numerator	INT16	Always	Always
1022	Encoder denominator	INT16	Always	Always
1023	Num cams (1)	UINT16	Always	Always
1024	Num cams (2)	UINT16	Always	Always
1025	Num cams (3)	UINT16	Always	Always
1026	Num cams (4)	UINT16	Always	Always
1027	Num cams (5)	UINT16	Always	Always
1028	Num cams (6)	UINT16	Always	Always
1029	Num cams (7)	UINT16	Always	Always

Address	Name	Type	Read	Write
102A	Num cams (8)	UINT16	Always	Always
102B	Master module (1) (MSB)	UINT32	Always	Always
102C	Master module (1) (LSB)	UINT32	Always	Always
102D	Slave module (1) (MSB)	UINT32	Always	Always
102E	Slave module (1) (LSB)	UINT32	Always	Always
102F	Master module (2) (MSB)	UINT32	Always	Always
1030	Master module (2) (LSB)	UINT32	Always	Always
1031	Slave module (2) (MSB)	UINT32	Always	Always
1032	Slave module (2) (LSB)	UINT32	Always	Always
1033	Master module 3) (MSB)	UINT32	Always	Always
1034	Master module (3) (LSB)	UINT32	Always	Always
1035	Slave module (3) (MSB)	UINT32	Always	Always
1036	Slave module (3) (LSB)	UINT32	Always	Always
1037	Master module (4) (MSB)	UINT32	Always	Always
1038	Master module (4) (LSB)	UINT32	Always	Always
1039	Slave module (4) (MSB)	UINT32	Always	Always
103A	Slave module (4) (LSB)	UINT32	Always	Always
103B	Master module (5) (MSB)	UINT32	Always	Always
103C	Master module (5) (LSB)	UINT32	Always	Always
103D	Slave module (5) (MSB)	UINT32	Always	Always
103E	Slave module (5) (LSB)	UINT32	Always	Always
103F	Master module (6) (MSB)	UINT32	Always	Always
1040	Master module (6) (LSB)	UINT32	Always	Always
1041	Slave module (6) (MSB)	UINT32	Always	Always
1042	Slave module (6) (LSB)	UINT32	Always	Always
1043	Master module (7) (MSB)	UINT32	Always	Always
1044	Master module (7) (LSB)	UINT32	Always	Always
1045	Slave module (7) (MSB)	UINT32	Always	Always
1046	Slave module (7) (LSB)	UINT32	Always	Always
1047	Master module (8) (MSB)	UINT32	Always	Always
1048	Master module (8) (LSB)	UINT32	Always	Always
1049	Slave module (8) (MSB)	UINT32	Always	Always
104A	Slave module (8) (LSB)	UINT32	Always	Always
1100	Cam points (1)	UINT16	Always	Always
1101	Cam points (2)	UINT16	Always	Always
1102	Cam points (3)	UINT16	Always	Always
1103	Cam points (4)	UINT16	Always	Always
1104	Cam points (5)	UINT16	Always	Always
1105	Cam points (6)	UINT16	Always	Always
1106	Cam points (7)	UINT16	Always	Always
1107	Cam points (8)	UINT16	Always	Always

These parameters can be set in the Caliper menu "***Electronic Cam***" in which the following screen appears:



3.14.2.a Register 316h (790 dec) - Selection index (N° cam)

Address	Name	Type	Range	Default	Unit	Read	Write
316h	**Selection index (N° cam)**	UINT16	0 ... 65535			Always	No

This register contains the index of the selected cam profile.

3.14.2.b Register 101Ch (4124 dec) Electronic Cam - flag

Address	Name	Type	Range	Default	Unit	Read	Write
101Ch	**Electronic Cam - flag**	INT16	0 ... 65535			Always	Always

this register contains some flags used when the the control type is set on ***Electronic Cam***. The list of this flags is reported in the table below:

Bit	Description	
0	***Frequency mode***	
	Value	**Description**
	0	Channel A-B
	1	Frequency direction
1	***Type pos. - cyclic cam disengagement***	
	Value	**Description**
	0	Absolute
	1	Relative
2	***Type pos. - acyclic cam disengagement***	
	Value	**Description**
	0	Absolute
	1	Relative
3	***Enable synchronism***	
	Value	**Description**
	0	Synchronism disabled
	1	Synchronism enabled
4	***Correction slave module***	
	Value	**Description**
	0	Correction disabled
	1	Correction enabled
5 ... 15	***Reserved***	

3.14.2.c Register 101Dh (4125 dec) - Mode cam- flag

Address	Name	Type	Range	Default	Unit	Read	Write
101Dh	**Mode cam - flag**	INT16	0 ... 65535			Always	Always

This register contains 8 flags, one for every possible cam (there may not be more than 8 cams). They're used to decide if the assigned cam must be executed cyclically or acyclically:

- If the cam is cyclic (assigned bit = 0), it will be endlessly repeated;
- If the cam is acyclic (assigned bit = 1), it will be repeated for a preset number of cycles, after which it will stop.

Cam mode description			
BIt	**n° cam**	**value**	**Description**
0	cam mode 1	0	cyclic
1	cam mode 2		
2	cam mode 3		
3	cam mode 4		
4	cam mode 5	1	acyclic
5	cam mode 6		
6	cam mode 7		
7	cam mode 8		
8 ... 15	***reserved***		

3.14.2.d Register 1021h (4129 dec) Encoder numerator

Address	Name	Type	Range	Default	Unit	Read	Write
1021h	**Encoder numerator**	INT16	-32768 ... 32767	1		Always	Always

The number of encoder pulses that the slave receives from the master can be multiplied by a costant greater or lesser than 1. In this way we can modify the position reference used for the cam, making the slave move faster or slower according to the costant used. This costant is equal to the ratio between the "***Encoder Numerator***" (register 1021h) and the "***Encoder Denominator***" (register 1022h).

3.14.2.e Register 1022h (4130 dec) Encoder denominator

Address	Name	Type	Range	Default	Unit	Read	Write
1022h	**Encoder denominator**	INT16	1 ... 65535	1		Always	Always

The number of encoder pulses that the slave receives from the master can be multiplied by a costant greater or lesser than 1. In this way we can modify the position reference used for the cam, making the slave move faster or slower according to the costant used. This costant is equal to the ratio between the "***Encoder Numerator***" (register 1021h) and the "***Encoder Denominator***" (register 1022h).

3.14.2.f Registers 1023h (4131 dec) ... 102Ah (4138 dec) - Num cams

Address	Name	Type	Range	Default	Unit	Read	Write
1023h **1024h** **1025h** **1026h** **1027h** **1028h** **1029h** **102Ah**	**Num cams**	UINT16	1 ... 60000	1		Always	Always

These 8 registers are used to set for every cam how many cycles it should be repeated (there may not be more than 8 cams). These registers are used only when the "***Cam mode***" of the selectd cam is "***acyclic***" .

3.14.2.g Registers 102Bh (4139 dec) and 102Ch (4140 dec)... Registers 1047h (4167 dec) and 1048h (4168 dec)- Master mode

Address	Name	Type	Range	Default	Unit	Read	Write
102Bh 102Ch **102Fh 1030h** **1033h 1034h** **1037h 1038h** **103Bh 103Ch** **103Fh 1040h** **1043h 1044h** **1047h 1048h**	**Master mode**	UINT32	0 ... 1048575	10000 (dec)		Always	Always

These 16 registers form 8 32-bit variables, and therefore they should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

These 8 variables set the "***Master module***" for every stored cam. The "***Master module***" corresponds to the "period" of the cam profile, i.e it represents the number of pulses that the slave has to receive from the master encoder to complete one cam profile. When the number of pulses counted by the slave reaches the "***Master module***", it gets reset to 0. The slave associate the "***Master module***" with the x-coordinate of the last point in the table of the drawed cam. The x-coordinates of the other points are derived using a proportion. Using this proportion the slave can map directly the number of pulses received by the master with the target position that it has to reach. The ratio between the "***Master module***" and the "***Cam points***" equals the number of master pulses found between two points of the cam table.

*Note: the number of master pulses received by the slave is modified by the parameters "**Encoder numerator**" (1021h) and "**Encoder denominator**" (1022h).*

$$cam\_step\_length = \frac{(Master\_Module)}{(N°\_point\_cam - 1)}$$

3.14.2.h Registers 102Dh (4141 dec) and 102Eh (4142 dec)... Registers 1049h (4169 dec) and 104Ah (4170 dec)- Slave module

Address	Name	Type	Range	Default	Unit	Read	Write
102Dh 102Eh **1031h 1032h** **1035h 1036h** **1039h 103Ah** **103Dh 103Eh** **1041h 1042h** **1045h 1046h** **1049h 104Ah**	**Slave Module**	UINT32	0 ... 4194303	10000 (dec)		Always	Always

These 16 registers form 8 32-bit variables, and therefore they should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

These variables contain the "***Slave Module***" of every cam stored on the Drive. The "***Slave Module***" represents the maximum excursion that the cam could perfeorm, expressed in increments; 65536 increments correspond to 1 turn of the motor. When the cam is drawed all the y-coordinates of its points could range only from 0 to 65535. The Drive multiplicates all the y-coordinates of the cam table by the "***Slave Module***" and then divides them by 65536. Now the y-coordinates of the cam will be able to reach higher values (depending on the slave module), which means that the slave will be able to do more turns in a single cam.

$$Slave\_value = \frac{(Slave\_Module \times Table\_slave\_value)}{(65536)}$$

3.14.2.i Registers 1100h (4352 dec) ... 1107h (4359 dec) - Cam points

Address	Name	Type	Range	Default	Unit	Read	Write
1100h **1101h** **1102h** **1103h** **1104h** **1105h** **1106h** **1107h**	**Cam points**	UINT16	16 ... 912	64		Always	Always

These 8 registers store the number of points that compose every cam stored in the Drive. There may not be more than 8 cams stored in the Drive, and the sum of the points that compose all the cams cannot be more than 912.

3.14.3 Electronic cam Mode - Synchronism and Shift parameters

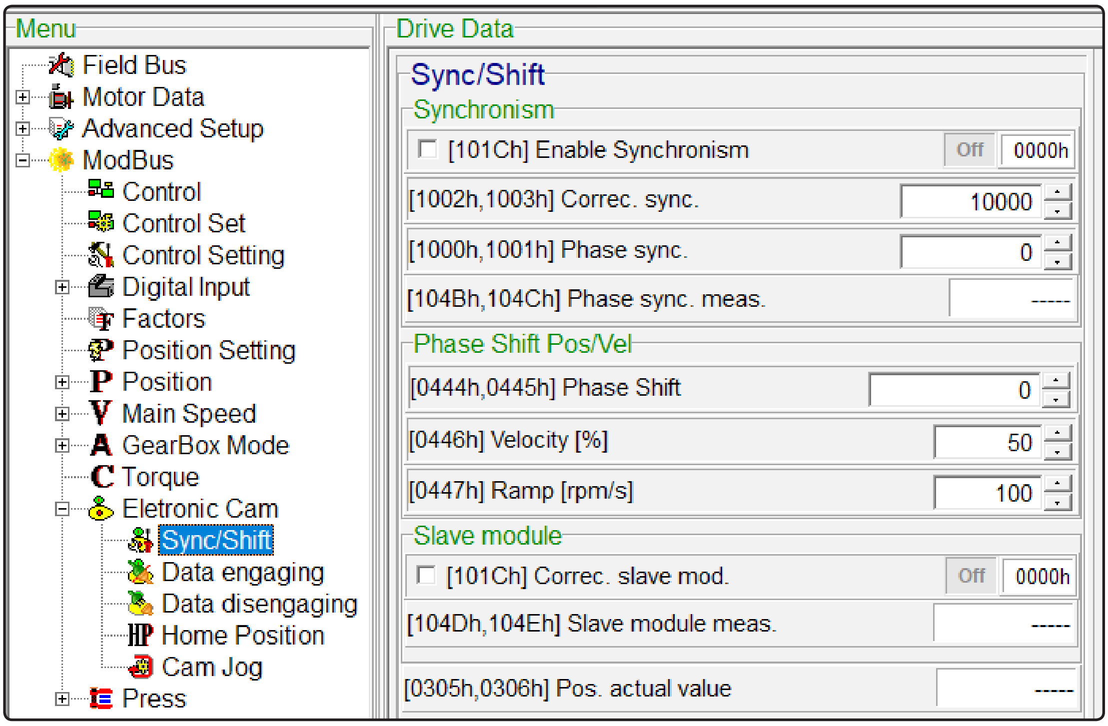
This paragraph describes the ***Synchronism function*** and the ***Shift function*** of the ***Electronic Cam*** control.

The ***Synchronism function*** is used to prevent the cam profile executed by the Slave from increasingly shifting from the Master's reference after every cycle. This problem is not always present, but it may appear if there are rounding errors in the values of the ***Master module*** or the ***Slave module***. Because of this rounding errors the final position of the cam profile won't be exactly where it was intended, therefore the next cam profile will start from a wrong position and its final position will be affected again by the same rounding error. These errors, even if they're small, add up over the time, increasing the phase shift between the cam profile executed and the master reference. In order to avoid that, you have to use an external sensor, which send a ***sync pulse*** to the Slave when the axis of the latter pass in front of it. After that you have to set on the Drive the expected value of the counter of the Master reference pulses in the ***Phase sync*** parameter, that you expect to have when the Drive receives the ***sync pulse***. The counter of the Master reference's pulses ranges from 0 to the ***Master Module*** value. If ,as time passes, this counter "shifts" because the set ***Master Module*** is a rounded value of the real ***Master module***, the ***sync pulse*** will not arrive when the counting matches with the ***Phase sync***: therefore the Slave will know that it is in the wrong position, and it will correct the reference pulses counter in order to cancel this error for the next cam profile. The Slave will do this correction only if the ***Synchronism function*** is enabled, and only if one of the input is assigned to receive the ***sync pulse***.

Address	Name	Type	Read	Write
1000h	Phase sync (MSB)	UINT32	Always	Always
1001h	Phase sync (LSB)			
1002h	Correction sync (MSB)	UINT32	Always	Always
1003h	Correction sync (LSB)			
101Ch	Cam - flag	INT16	Always	Always
104Bh	Phase sync measured (MSB)	INT32	Always	Always
104Ch	Phase sync measured (LSB)			
104Dh	Slave module meas. (MSB)	INT32	Always	No
104Eh	Slave module meas. (LSB)			

The ***Shift function***, instead, is specifically used to shift the relative position between the Slave and the Master. With this function you can increase or decrease of a certain value the counter of the Master reference's pulses. The cam profile therefore will shift of a certain "phase", and after that it will remain in phase with the master's reference. The registers *0444h, 0445h,0446h* e *0447h* used for the ***Shift function*** were already described for the ***Gearbox*** control on the paragraphs *"3.12.2.a Registers 0444h (1092 dec), 0445h (1093 dec) Phase shift" pag. 86* , *"3.12.2.b Register 0446h (1094 dec) Shift speed" pag. 86* and *"3.12.2.c Register 0447h (1095 dec) Shift acceleration" pag. 86.*

The parameters used by these two functions can be set in the Caliper menu "***Electronic Cam- Sync/Shift***" in which the following screen appears:



3.14.3.a Registers 1000h (4096 dec), 1001h (4097 dec) Phase sync

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
1000h (MSB) **1001h (LSB)**	**Phase sync**	UINT32	0... 80000000h	0		Always	Always

The parameter ***Phase sync*** represents the value that the reference's pulses counter should have when it receives the ***Sync Pulse*** from the assigned input. If the ***synchronism function*** is enabled, when the ***sync pulse*** occur the Drive calculates the difference between the measured value of the counter and the ***Phase sync***, and using the obtained result it gradually corrects the counter of the Master's reference pulses to cancel this difference for the next time the ***sync pulse*** occur.

3.14.3.b Registers 1002h (4098 dec), 1003h (4099 dec) Correction sync

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
1002h (MSB) **1003h (LSB)**	**Correction sync**	UINT32	10... 80000000h	10000		Always	Always

These registers contain the "***Correction sync***", i.e. the maximum value used for the correction of the position error. When the ***sync pulse*** occurs,the Drive calculates the position error as the difference between the measured value of the reference position of the Master (expressed in pulses) and the value of ***Phase sync*** (registers 1000h,1001h). If this 2 values differ then the Drive will make some corrections to reduce their difference to zero when the next ***sync pulse*** occurs. The "***Correction sync***" limits the value of position that the Drive could recover in single cam cycle; If the position error were cancelled all at once the the resulting cam profile would be too much distorted with respect to the drawed cam profile.

3.14.3.c Registers 104Bh (4171 dec), 104Ch (4172 dec) - Phase sync measured

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
104Bh (MSB) **104Ch (LSB)**	**Phase sync measured**	INT32	± 7FFFFFFFh			Always	No

These registers report on a text box the value of the measured s***ynchronism phase***, i.e the value of the reference's pulses counter read when the ***sync pulse*** occured.

3.14.3.d Registers 104Dh (4173 dec), 104Eh (4174 dec) - Slave module meas.

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
104Dh (MSB) **104Eh (LSB)**	**Slave modul meas.**	INT32	± 7FFFFFFFh			Always	No

These registers contain the value of the ***Measured Slave Module***.

3.14.4 Data engaging

This paragraph lists all the parameters used during the engagement phase when the control of the Drive is set on ***Electronic cam***. This phase starts when the bit 4 of the *control word* is switched from 0 to 1. The engagement phase purpose is to speed up the Slave axis until it reaches the same speed of the cam profile obtained by tha cam table and the reference signal of the Master. When this speed is reached the engagement phase ends, and the Slave will be " in Phase" with the Master and it will rigidly execute the drawed cam profile.

Address	Name	Type	Read	Write
1004h	Start ramp phase (MSB)	UINT32	Always	Always
1005h	Start ramp phase (LSB)			
1006h	Start engage phase (MSB)	UINT32	Always	Always
1007h	Start engage phase (LSB)			
1008h	Master space (MSB)	UINT32	Always	Always
1009h	Master space (LSB)			
100Ah	Slave space (MSB)	UINT32	Always	Always
100Bh	Slave space (LSB)			
101Eh	Engagement type	UINT16	Always	Always

3.14.4.a Registers 1004h (4100 dec), 1005h (4101 dec) - start ramp phase

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
1004h (MSB) **1005h (LSB)**	**Start ramp phase**	UINT32	0... 100000000h	0		Always	Always

The parameter "***Start ramp phase***" is used when the parameter "***Engagement type***" is set on " ***Engagement on ramp***". In this mode the engagement phase will be executed with a speed ramp and it will start when the Master reference equals the value of "***Start ramp phase***".

3.14.4.b Registers 1006h (4102 dec) , 1007h (4103 dec) - Start engage phase

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
1006h (MSB) **1007h (LSB)**	**Start engage phase**	UINT32	0... 100000000h	0		Always	Always

The parameter "***Start engage phase***" is used when the parameter "***Engagement type***" is set on " ***Engagement on master phase***". In this mode the execution of the cam profile starts when the Master reference equals the value of "***Start engage phase***".

3.14.4.c Registers 1008h (4104 dec), 1009h (4105 dec) - Master space

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
1008h (MSB) **1009h (LSB)**	**Master space**	UINT32	1000... 100000000h	65536		Always	Always

The parameter is used when the parameter "***Engagement type***" is set on " ***Engagement on ramp***". It defines the space covered by the master,expressed as a number of pulses, during the speed ramp of the engagement phase.

3.14.4.d Registers 100Ah (4106 dec) e 100Bh (4107 dec) - Slave space

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
100Ah (MSB) **100Bh (LSB)**	**Slave space**	UINT32	1000... 100000000h	65536		Always	Always

The parameter is used when the parameter "***Engagement type***" is set on " ***Engagement on ramp***". It defines the space covered by the slave axis,expressed as a number of pulses, during the speed ramp of the engagement phase (65536 pulses corresponds to one turn of the axis).

3.14.4.e Register 101Eh (4126 dec) - Engagement type

Address	Name	Type	Range	Default	Unit	Read	Write
101Eh	**Engagement type**	UINT16	0 ... 2	0		Always	Always

This register contains the "***Engagement type***". This parameter is used to select how the Slave will engage with the Master reference before executing the Cam's profile.

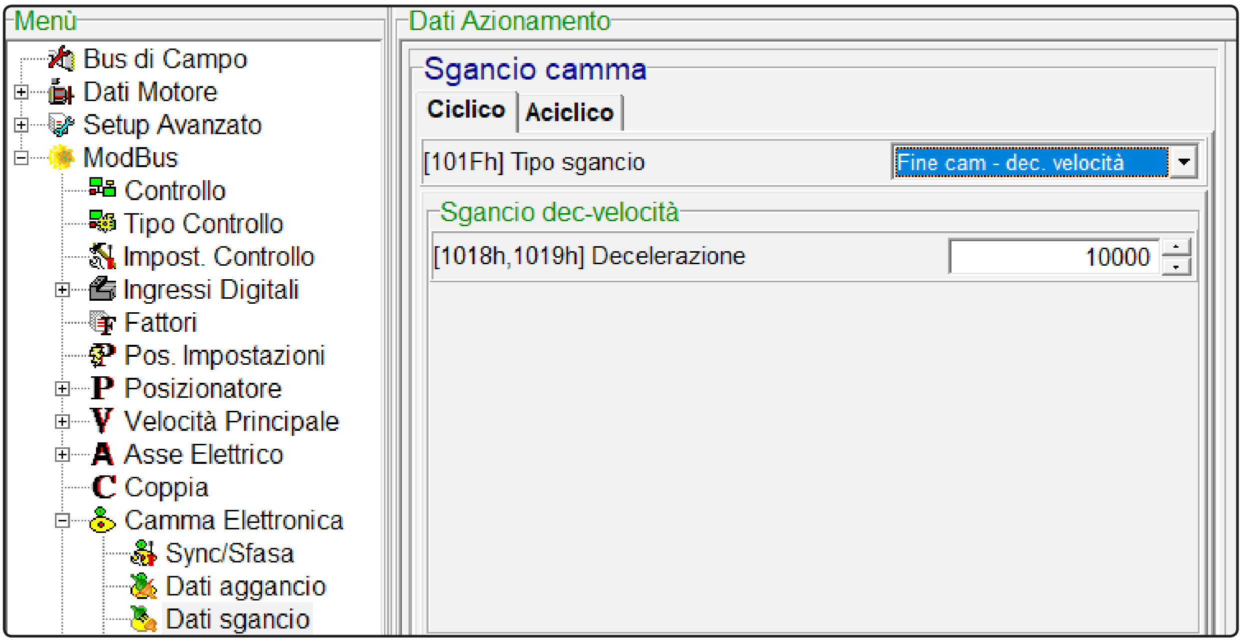
Value	Type	Description
0	Immediate engagement	After the command of start the slave will start executing immediately the cam's profile
1	Engagement on master phase	After the command of start, The slave will start executing the cam's profile when the master's reference has reached the value set on "***start engage phase***" (1006h,1007h)
2	Engagement on ramp	The Slave follows a speed ramp in order to engage with the speed of the cam's profile that has to be executed. The parameter "***Master space***" (1008h,1009h) defines the space that the Master will cover during this ramp, the "***Slave space***" (100Ah,100Bh) defines the space that the Slave will cover during this ramp, and the "***Start ramp phase***" (1004h,1005h) defines the position that the master's reference has to reach in order to start the engagement phase.

3.14.5 Data disengaging

This paragraph lists all the parameters used during the disengagement phase when the control of the Drive is set on ***Electronic cam***. When the bit 4 of the *control word* is switched from 1 to 0 the Slave stops executing the cam's profile and starts the disengagement phase, which is used to bring the speed of the Slave axis to 0 with the selected mode. Once the disengagement phase is over, the bit 5 of the *Status word* is lowered and the motor stands still holding the torque. Switching the bit 4 of the *control word* to 0 starts the disengagement phase for both the cyclic cams and the acyclic cams. It should be remembered that for the acyclic cams the disengagement phase will starts automatically when th Slave has executed the cam's profile for the requested number of times (defined on the parameter "***Num cams"***, see *"3.14.2.f Registers 1023h (4131 dec) ... 102Ah (4138 dec) - Num cams" pag. 105*).

Address	Name	Type	Read	Write
100Ch	Pos - cyclic disengage (MSB)	INT32	Always	Always
100Dh	Pos -cyclic disengage (LSB)			
100Eh	Speed-cyclic disengage (MSB)	UINT32	Always	Always
100Fh	Speed-cyclic disengage (LSB)			
1010h	Accel./decel.-cyclic disengage (MSB)	UINT32	Always	Always
1011h	Accel./decel.-scyclic disengage (LSB)			
1012h	Pos - acyclic disengage (MSB)	INT32	Always	Always
1013h	Pos -acyclic disengage (LSB)			
1014h	Speed - acyclic disengage (MSB)	UINT32	Always	Always
1015h	Speed - acyclic disengage (LSB)			
1016h	Accel./decel.-acyclic disengage (MSB)	UINT32	Always	Always
1017h	Accel./decel.-acyclic disengage (LSB)			
1018h	Deceleration-cyclic disengage (MSB)	UINT32	Always	Always
1019h	Deceleration-cyclic disengage (MSB)			
101Ah	Deceleration-acyclic disengage (MSB)	UINT32	Always	Always
101Bh	Deceleration-acyclic disengage (MSB)			
101Fh	Cyclic disengage type	UINT16	Always	Always
1020h	Acyclic disengage type	UINT16	Always	Always

These parameters can be set in the Caliper menu "***Electronic Cam- Data disengaging***" in which the following screen appears:



3.14.5.a Registers 100Ch (4108 dec), 100Dh (4109 dec) - Pos - cyclic disengage

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
100Ch (MSB) **100Dh (LSB)**	**Pos-cyclic disengage**	INT32	± 7FFFFFFFh	65536		Always	Always

"Pos - cyclic disengage" represents the position reached at the end of the disengagement phase of a cyclic cam's profile, when the parameter "***cyclic disengage type"***(register 101Fh) is set on "***Immediate-position***" or"***End cam-position***". This parameter could be considered as an absolute value or a relative value, depending on the value of the Bit 1 of the parameter ***Cam-flag*** (bit 1 of register 101Ch ="***Type pos. - cyclic cam disengagement***"). If ***"Pos-cyclic disengage"*** is considered as a relative value, the motor stops on the position obtained by the sum of ***"Pos - cyclic disengage"*** and the position measured when the disengagement phase was commanded.If ***"Pos-cyclic disengage"*** is considered as an absolute value, the motor stops exactly on ***"Pos - cyclic disengage" .***

3.14.5.b Registers 100Eh (4110 dec), 100Fh (4111 dec) - Speed- cyclic disengage

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
100Eh (MSB) **100Fh (LSB)**	**Speed - cyclic disengage**	UINT32	1... 7FFFFFFFh	500		Always	Always

These registers contain the "**Speed - cyclic disengage** ", i.e. the maximum speed that could be reached during the disengagement phase of a cyclic cam, when the parameter "***cyclic disengage type"***(register 101Fh) is set on "***Immediate-position***" or"***End cam-position***".

3.14.5.c Registers 1010h (4112 dec) , 1011h (4113 dec) - Accel./decel. - cyclic disengage

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
1010h (MSB) **1011h (LSB)**	**Accel./decel. - cyclic disengage**	UINT32	1... 7FFFFFFFh	10000		Always	Always

These registers contain the absolute value of the accelerations and decelerations applied during the disengagement phase of a cyclic cam, when the parameter "***cyclic disengage type"***(register 101Fh) is set on "***Immediate-position***" or"***End cam-position***".

3.14.5.d Registers 1012h (4114 dec), 1013h (4115 dec) - Pos - acyclic disengage

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
1012h (MSB) **1013h (LSB)**	**Pos -acyclic disengage**	INT32	± 7FFFFFFFh	65536		Always	Always

"Pos - acyclic disengage" represents the position reached at the end of the disengagement phase of an acyclic cam's profile, when the parameter "***acyclic disengage type"***(register 1020h) is set on "***Immediate-position***" or"***End cam-position***". This parameter could be considered as an absolute value or a relative value, depending on the value of the Bit 2 of the parameter ***Cam-flag*** (bit 2 of register 101Ch ="***Type pos. - acyclic cam disengagement***"). If ***"Pos-acyclic disengage"*** is considered as a relative value, the motor stops on the position obtained by the sum of ***"Pos - acyclic disengage"*** and the position measured when the disengagement phase was commanded. If ***"Pos-acyclic disengage"*** is considered as an absolute value, the motor stops exactly on ***"Pos - acyclic disengage"*** .

3.14.5.e Registers 1014h (4116 dec), 1015h (4117 dec) - Speed - acyclic disengage

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
1014h (MSB) **1015h (LSB)**	**Speed- acyclic disengage**	UINT32	1... 7FFFFFFFh	500		Always	Always

These registers contain the "**Speed - acyclic disengage** ", i.e. the maximum speed that could be reached during the disengagement phase of an acyclic cam, when the parameter "a***cyclic disengage type"***(register 1020h) is set on "***Immediate-position***" or"***End cam-position***".

3.14.5.f Registers 1016h (4118 dec), 1017h (4119 dec) - Accel./decel. - acyclic disengage

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
1016h (MSB) **1017h (LSB)**	**Accel./decel. acyclic disengage**	UINT32	1... 7FFFFFFFh	10000		Always	Always

These registers contain the absolute value of the accelerations and decelerations applied during the disengagement phase of an acyclic cam, when the parameter "***acyclic disengage type"***(register 1020h) is set on "***Immediate-position***" or"***End cam-position***".

3.14.5.g Registers 1018h (4120 dec), 1019h (4121 dec) - Deceleration - cyclic disengage

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
1018h (MSB) **1019h (LSB)**	**Deceleration - cyclic disengage**	UINT32	1... 7FFFFFFFh	10000		Always	Always

These registers are used to set the deceleration applied during the disengagement phase with a speed ramp of a cyclic cam, when the parameter "***cyclic disengage type"*** (register 101Fh) is set on "***end cam - dec. speed***" or "***immediate- dec. speed***".

3.14.5.h Registers 101Ah (4122 dec), 101Bh (4123 dec) - Deceleration - acyclic disengage

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
101Ah (MSB) **101Bh (LSB)**	**Deceleration - acyclic disengage**	UINT32	1... 7FFFFFFFh	10000		Always	Always

These registers are used to set the deceleration applied during the disengagement phase with a speed ramp of an acyclic cam, when the parameter "***acyclic disengage type"*** (register 1020h) is set on "***end cam - dec. speed***" or "***immediate- dec. speed***".

3.14.5.i Register 101Fh (4127 dec) - Cyclic disengage type

Address	Name	Type	Range	Default	Unit	Read	Write
101Fh	**cyclic disengage type**	UINT16	0 ... 3	2		Always	Always

This register is used to set how the Slave will disengage from a cyclic cam's profile that it's executing; the disengagement is started when the bit 4 of the *Control word* is switched to 0

Value	Type	Description
0	end cam - dec. speed	Once the running cam is completed, the disengagement phase starts. The motor is stopped with a ramp of speed, whose deceleration is set on "**Deceleration-cyclic disengage** " (1018h,1019h)
1	End cam-position	Once the running cam is completed, a positioning of the Slave is performed with the target set on "**Pos-cyclic disengage**" (100Ch,100Dh)
2	immediate- dec. speed	When the bit 4 of the C*ontrol word* switches to 0, the running cam is interrupted and it's immediately started the disengagement, which stops the motor with a ramp of speed whose deceleration is set on "**Deceleration-cyclic disengage** " (1018h,1019h)
3	Immediate-position	When the bit 4 of the C*ontrol word* switches to 0, the running cam is interrupted and it's immediately started a positioning of the Slave with the target set on "**Pos-cyclic disengage**" (100Ch,100Dh)

3.14.5.j Register 1020h (4128 dec) - Acyclic disengage type

Address	Name	Type	Range	Default	Unit	Read	Write
1020h	**Acyclic disengage type**	UINT16	0 ... 1	0		Always	Always

This register is used to set how the Slave will disengage from an acyclic the cam's profile that it's executing; the disengagement is started when the bit 4 of the *Control word* is switched to 0.It should be remembered that for the acyclic cams the disengagement phase will starts automatically when th Slave has executed the cam's profile for the requested number of times (defined on the parameter "***Num cams"***, see *"3.14.2.f Registers 1023h (4131 dec) ... 102Ah (4138 dec) - Num cams" pag. 105*).

Value	Type	Description
0	end cam - dec. speed	Once the running cam is completed for the number of times set on "***Num Cams***", the disengagement phase starts. The motor is stopped with a ramp of speed, whose deceleration is set on "**Deceleration-acyclic disengage** " (101Ah,101Bh)
1	End cam-position	Once the running cam is completed for the number of times set on "***Num Cams***", a positioning of the Slave is performed with the target set on "**Pos-acyclic disengage**" (1012h,1013h)

3.14.6 Electronic cam - Jog

The ***Jog*** functioning when the drive is set on ***Electronic Cam*** Mode is the same as the one described on paragraph *"3.10.4 Position Jog Parameters" pag. 58* when the drive is set on ***Positioner*** Mode. The used parameters are the same.

3.14.7 Electronic cam - Home Position

The ***Home position*** functioning when the drive is set on ***Electronic Cam*** is the same as the one described on paragraph *"3.10.5 Home Position" pag. 60* when the drive is set on ***Positioner*** Mode. The used parameters are the same.

3.15 Press Mode

The control type ***Press*** is selected when the parameter "***control type***" (Register 0401h) is set to 5. With this mode, in its ordinary functioning, the Drive regulates the speed of a servopump in order to bring its pressure to the reference value.This is its normal usage, wich is applied when the bit 5 of the *Control word* is set on 0. However, if the bit 5 of the *Control word* is set on 1, then the Drive will actually switch to a ***Speed*** control, without changing the parameter "***control type***" (Register 0401h). This particular usage could be useful if you want to control with one Drive both the servopump of an hydraulic unit in ***Press*** mode and another motor in ***Speed*** mode, connecting the Drive first with one, and then with the other (you can't control both motors with the two previously descripted modes at the same time).

3.15.1 Control word and status word

In this paragraph are described the structures of the *Control word* and the *Status word* when the ***Press*** mode is selected.

3.15.1.a Setting of the Controlword bits in Press mode

<table>
<tr><th>Bit</th><th colspan="2">Description</th></tr>
<tr><td>0</td><td colspan="2">Switch on</td></tr>
<tr><td>1</td><td colspan="2">Enable operation</td></tr>
<tr><td>2</td><td colspan="2">Fault reset</td></tr>
<tr><td>3</td><td colspan="2">Halt</td></tr>
<tr><td>4</td><td colspan="2">Enable Setpoint</td></tr>
<tr><td rowspan="4">5</td><td colspan="2">Set mode Press-vel</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>The drive works in Press mode</td></tr>
<tr><td>1</td><td>The drive works in Speed mode</td></tr>
<tr><td>6 ... 15</td><td colspan="2">reserved</td></tr>
</table>

3.15.1.b Meaning of the statusword bits in Press mode

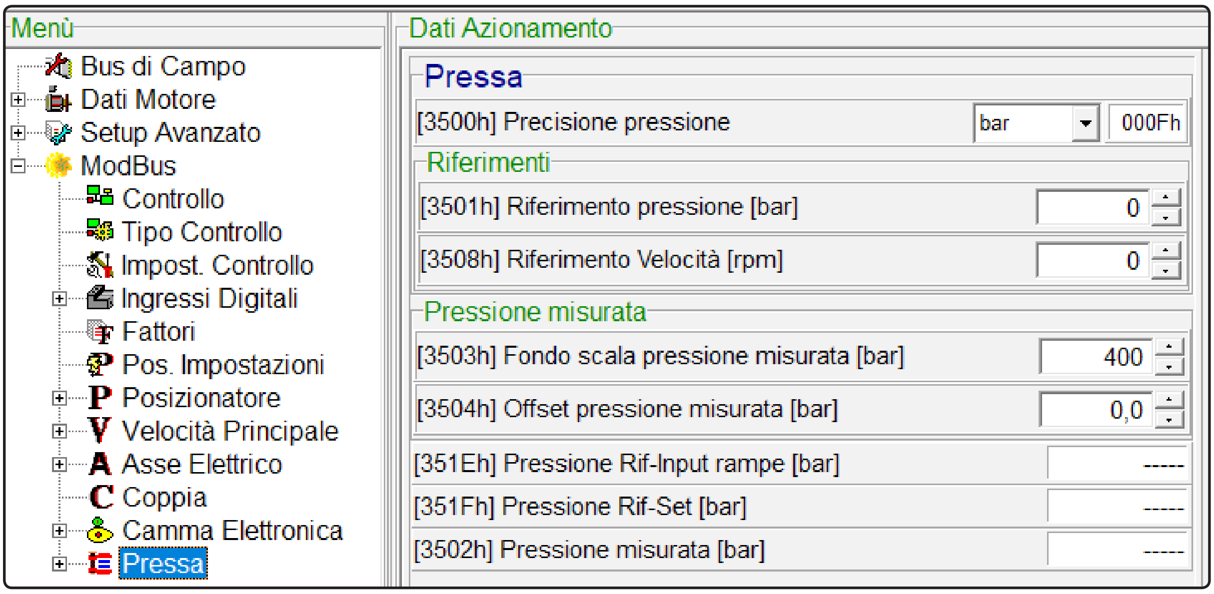
<table>
<tr><th>bit</th><th colspan="2">description</th></tr>
<tr><td>0</td><td colspan="2">Switch on</td></tr>
<tr><td>1</td><td colspan="2">Enable operation</td></tr>
<tr><td>2</td><td colspan="2">Ramp Stop</td></tr>
<tr><td>3</td><td colspan="2">Ramp Halt</td></tr>
<tr><td>4</td><td colspan="2">Fault</td></tr>
<tr><td rowspan="4">5</td><td colspan="2">Pressure reached</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Reference pressure not reached</td></tr>
<tr><td>1</td><td>Reference pressure reached</td></tr>
<tr><td rowspan="4">6</td><td colspan="2">Underpressure</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>The Drive is not in underpressure alarm</td></tr>
<tr><td>1</td><td>Underpressure Alarm</td></tr>
<tr><td rowspan="4">7</td><td colspan="2">Overpressure</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>The Drive is not in overpressure alarm</td></tr>
<tr><td>1</td><td>Overpressure Alarm</td></tr>
<tr><td rowspan="4">8</td><td colspan="2">Power-Limit</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>the Drive hasn’t reached the power limit</td></tr>
<tr><td>1</td><td>the Drive has reached the power limit</td></tr>
<tr><td rowspan="4">9</td><td colspan="2">Remote</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Remote control disabled</td></tr>
<tr><td>1</td><td>Remote control enabled</td></tr>
<tr><td rowspan="4">10</td><td colspan="2">Torque-Limit</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>the Drive hasn’t reached the torque limit</td></tr>
<tr><td>1</td><td>the Drive has reached the torque limit</td></tr>
<tr><td rowspan="4">11</td><td colspan="2">Zero speed</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>the motor is stopped</td></tr>
<tr><td>1</td><td>the motor is moving</td></tr>
<tr><td>12</td><td colspan="2">Manufacturer specific</td></tr>
<tr><td>13</td><td colspan="2">Warning</td></tr>
<tr><td>14</td><td colspan="2">Fault reaction</td></tr>
<tr><td>15</td><td colspan="2">Manufacturer specific</td></tr>
</table>

3.15.2 Press Parameters

This paragraph lists the main parameters used to control the motor in the ***Press*** mode

Address	Name	Type	Read	Write
3500h	Press - flag	UINT16	Always	Switch off
3501h	Pressure reference	UINT16	Always	Always
3502h	Measured pressure	UINT16	Always	No
3503h	Maximum measured pressure	UINT16	Always	Always
3504h	Offset measured pressure	UINT16	Always	Always
3508h	Speed reference	UINT16	Always	Always
351Eh	Pressure ref-input ramps	UINT16	Always	No
351Fh	Pressure ref-set	UINT16	Always	No

These parameters can be set in the Caliper menu "***Press***" in which the following screen appears:



3.15.2.a Register 3500h (13568 dec) - Press-flag

Address	Name	Type	Range	Default	Unit	Read	Write
3500h	**Press- flag**	UINT16				Always	Switch off

this register contains some flags used when the the control type is set on ***Press***. The list of this flags is reported in the table below:

bit	description	
0	***Enables speed ramp***	
	Value	**Description**
	0	Speed ramps disabled
	1	Speed ramps enabled
1	***Enables pressure ramp***	
	Value	**Description**
	0	Pressure ramps disabled
	1	Pressure ramps enabled
2	***Overpressure alarm***	
	Value	**Description**
	0	Overpressure alarm disabled
	1	Overpressure alarm enabled
3	***Underpressure alarm***	
	Value	**Description**
	0	Underpressure alarm disabled
	1	Underpressure alarm enabled
4	***Pressure accuracy***	
	Value	**Description**
	0	precision of 1 bar
	1	precisio of 1/10 of bar
5 ...15	***Manufacturer specific***	

3.15.2.b Register 3501h (13569 dec) - Pressure reference

Address	Name	Type	Range	Default	Unit	Read	Write
3501h	**Pressure reference**	UINT16	0... 1000	0	bar	Always	Always

This register is used to set the pressure reference, in [Bar], that the servopump must reach.

3.15.2.c Register 3502h (13570 dec) - Measured pressure

Address	Name	Type	Range	Default	Unit	Read	Write
3502h	**Measured pressure**	UINT16	-32768... 32767		bar	Always	No

This register contains the value of the pressure measured by the transducer. The transducer converts the pressure in a voltage proportional with its value, which is read by the main analogical input of the Drive. The Drive will then calculate the value in [Bar] of the measured pressure using a proportion (see *"3.15.2.d Register 3503h (13571 dec) - Maximum measured pressure" pag. 118*).

3.15.2.d Register 3503h (13571 dec) - Maximum measured pressure

Address	Name	Type	Range	Default	Unit	Read	Write
3503h	**Maximum measured pressure**	UINT16	0... 1000	400	bar	Always	Always

This register contains the full scale of pressure, in [Bar]. The transducer converts the measured pressure in a voltage proportional with its value, which is read by the main analogical input of the Drive.The Drive will then calculate the value in [Bar] of the measured pressure using a proportion,assuming that a voltage of 10 [V] corresponds to the pressure set on this register.

The formula used to convert the measured pressure in [Bar] is showed below:

$$measured\_pressure = \frac{Maximum\_pressure}{10} * volt\_input$$

Note: we assume that the pressure of 0 [bar] corresponds to a voltage of 0 [V], therefore we are using only half of the measurable range of voltage, which ranges from -10 [V] to + 10 [V]

3.15.2.e Register 3504h (13572 dec) - Offset measured pressure

Address	Name	Type	Range	Default	Unit	Read	Write
3504h	**Offset measured pressure**	UINT16	-1000... 1000	0	bar/ 10	Always	Always

This register contains a constant value which is added to (or subtracted from) the value of the measured pressure. It's used to eliminate the potential offset of the transducer.

3.15.2.f Register 3508h (13576 dec) - Speed reference

Address	Name	Type	Range	Default	Unit	Read	Write
3508h	**Speed reference**	UINT16	1... 9999	1	rpm	Always	Always

This register contains a value of speed expressed in [rpm]. When the bit 5 of the *Control word* is set to 0,this register represents the maximum limit that the speed produced by the pressure regulator cannot exceed. When the bit 5 of the *Control word* is set to 1,this register represents the speed reference that the Drive is trying to reach; in this case the control is changed in fact from ***Press*** mode to ***Speed*** mode (see paragraph *"3.15 Press Mode" pag. 115*).

3.15.2.g Register 351Eh (13598 dec) - Pressure ref-input ramps

Address	Name	Type	Range	Default	Unit	Read	Write
351Eh	**Pressure ref-input ramps**	UINT16	-32768... 32767		bar	Always	No

This register contains the value of the pressure reference, in [Bar], that the Drive is trying to reach. If the Drive is controlled with a Fieldbus (Modbus included), then this register has the same value of register 3501h (see *"3.15.2.b Register 3501h (13569 dec) - Pressure reference" pag. 118*). Instead,if the Drive is controlled with the ***Input/Output remote control***, then the value of this register will corresponds to the signal received by the analogical input. If the pressure ramps are enabled, this pressure reference will be "filtered" prior to being passed to the pressure regulator, in order to prevent it from any sharp variation.

3.15.2.h Register 351Fh (13599 dec) - Pressure ref-set

Address	Name	Type	Range	Default	Unit	Read	Write
351Fh	**Pressure ref-set**	UINT16	-32768... 32767		bar	Always	No

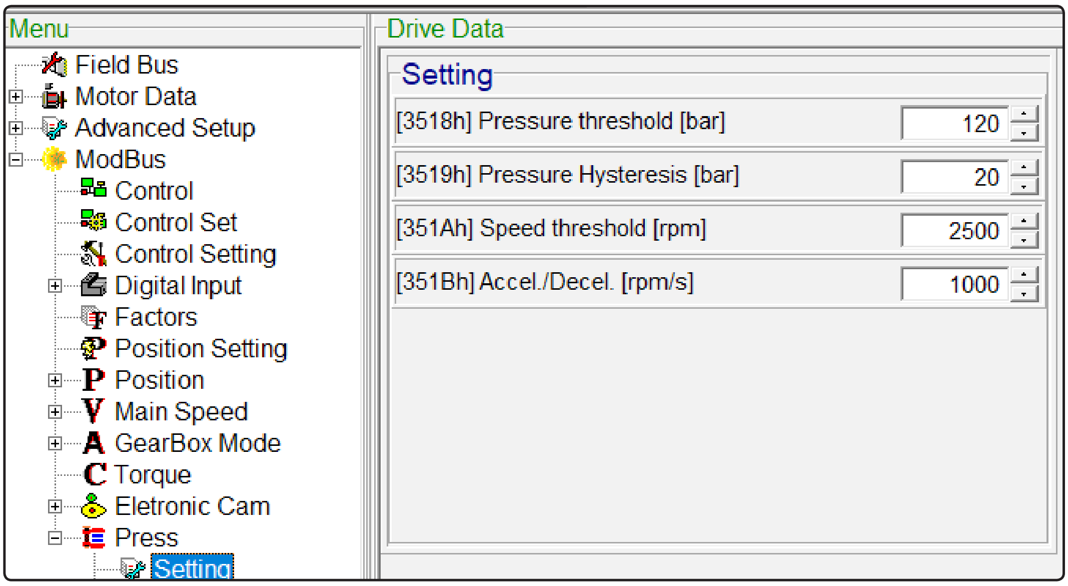
This register contains the pressure reference value after it has been "filtered" by the pressure ramps. These ramps are used to avoid the speed reference from changing abruptly when it's enabled or disabled.

3.15.3 Setting

When the Drive is working in ***Press*** mode it's possible to limit the maximum power that the servopump could absorb. In oder to set this limit of power you must use the parameters listed in the table below. Note that the power is never explicitly mentioned, because the Drive works with a value proportional with the power which is obtained by the product of pressure and speed, considering that the latter is proportional with the flow rate of the pump.

Address	Name	Type	Read	Write
3518h	Pressure threshold	UINT16	Always	Always
3519h	Pressure Hysteresis	UINT16	Always	Always
351Ah	Speed threshold	UINT16	Always	Always
351Bh	Accel./ decel.	UINT16	Always	Always

These parameters can be set in the Caliper menu "***Press-setting***" in which the following screen appears:



3.15.3.a Register 3518h (13592 dec) - Pressure threshold

Address	Name	Type	Range	Default	Unit	Read	Write
3518h	**Pressure threshold**	UINT16	0 ... 1000	120	bar	Always	Always

This register contains a pressure threshold. If the measured pressure surpasses the ***Pressure threshold*** (i.e. this register) and the measured speed surpasses the ***Speed threshold*** (register 0351Ah) at the same time, then the Power limit will be activated. When this limit is enabled the maximum speed that the motor cannot surpass is the one set on ***Speed threshold***.

3.15.3.b Register 3519h (13593 dec) - Pressure Hysteresis

Address	Name	Type	Range	Default	Unit	Read	Write
3519h	**Pressure Hysteresis**	UINT16	1 ... 50	20	bar	Always	Always

This register contains a pressure value, named ***Pressure Hysteresis***. If the Power limit for the ***Press*** mode is enabled,in order to disable it is necessary that the measured pressure or the measured speed or both decrease under a certain threshold. The measured speed must go under the ***Speed threshold*** (register 351Ah); the measured pressure must go under a value lower than the ***Pressure threshold*** (register 3518h), obtained by decreasing the ***Pressure threshold*** of the value of ***Pressure Hysteresis***.

3.15.3.c Register 351Ah (13594 dec) - Speed threshold

Address	Name	Type	Range	Default	Unit	Read	Write
351Ah	**Speed threshold**	UINT16	0 ... 9999	2500	rpm	Always	Always

This register contains a speed threshold. If the measured pressure surpasses the ***Pressure threshold*** (register 3518h) and the measured speed surpasses the ***Speed threshold*** (i.e. this register) at the same time, then the Power limit will be activated. When this limit is enabled the maximum speed that the motor cannot surpass is the one set on ***Speed threshold***.

3.15.3.d Register 351Bh (13595 dec) - Accel./decel.

Address	Name	Type	Range	Default	Unit	Read	Write
351Bh	**Accel./ decel.**	UINT16	1 ... 60000	1000	rpm/s	Always	Always

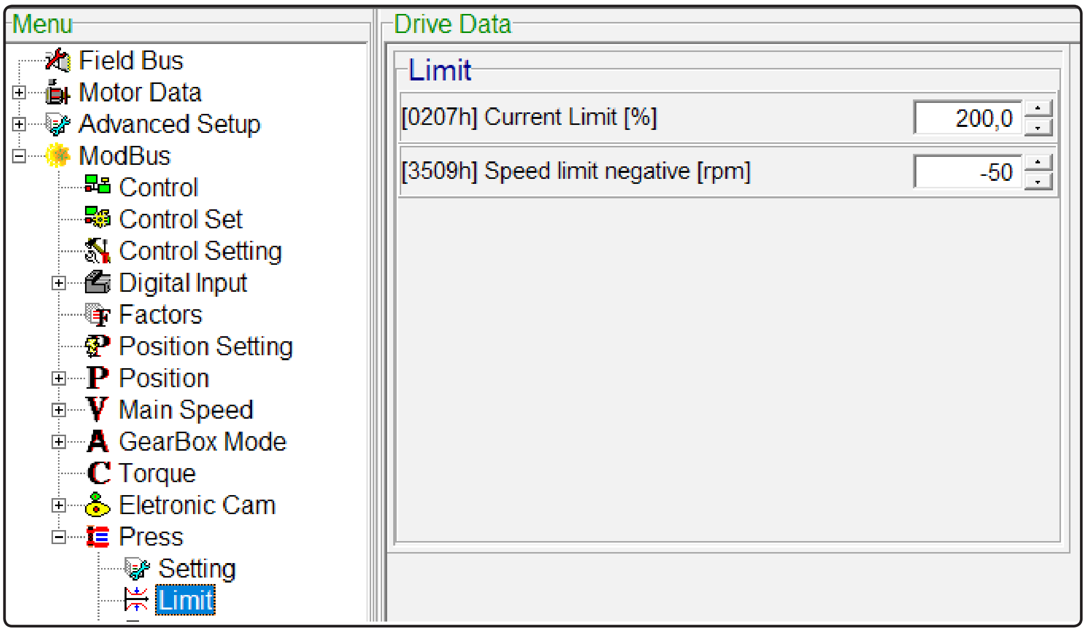
When the Drive has reached the Power limit there is still a speed ramp working, which is used to avoid sharp changes of the maximum limit of speed at the exit of the pressure regulator. This register contains the absolute value of the accelerations and decelerations applied by this speed ramp.

3.15.4 Limit

This paragraph lists the parameters used to set additional limits to the Drive when it's working on ***Press*** mode.

Address	Name	Type	Read	Write
0207h	Current limit	UINT16	Always	Always
3509h	Speed limit negative	UINT16	Always	Always

These parameters can be set in the Caliper menu "***Press-Limit***" in which the following screen appears:



3.15.4.a Register 0207h (519 dec) – Current limit

Address	Name	Type	Range	Default	Unit	Read	Write
0207h	**Current limit**	UINT16	0...3000	2000	‰	Always	Always

This register is used to set the upper limit of the absolute value of the current, expressed in thousandth (‰) with respect to the nominal current of the motor;for example, if we have a nominal current of 2,00 [A] and the current limit is set on 2000, then the maximum current allowed will be equal to 4,00 [A].

3.15.4.b Register 3509h (13577 dec) - Speed limit negative

Address	Name	Type	Range	Default	Unit	Read	Write
3509h	**Speed limit negative**	UINT16	-9999 ... 0	-50	rpm	Always	Always

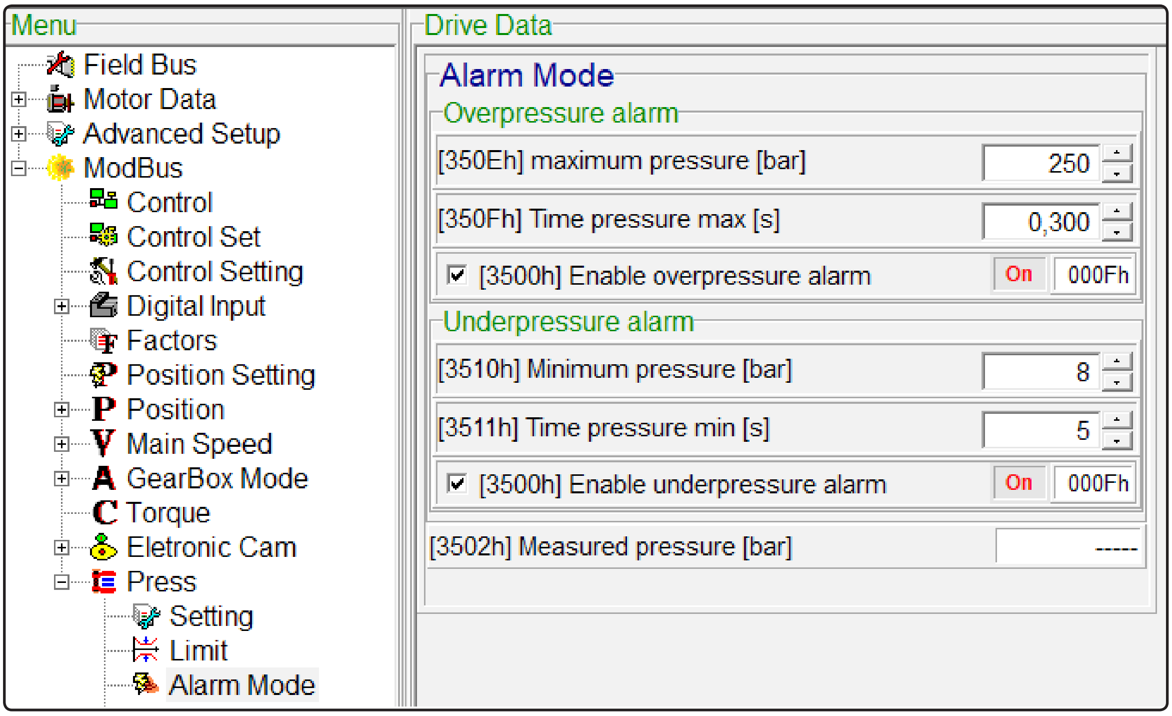
This register contains the maximum limit of speed with which the servopump can rotate in the opposite direction.

3.15.5 Alarm mode

This paragraph lists the parameters used to set the Underpressure alarm and the Overpressure alarm when the Drive is working with the ***Press*** mode.

Address	Name	Type	Read	Write
350Eh	**Maximum pressure**	UINT16	Always	Always
350Fh	**Time pressure max**	UINT16	Always	Always
3510h	**Minimum pressure**	UINT16	Always	Always
3511h	**Time pressure min**	UINT16	Always	Always

These parameters can be set in the Caliper menu "***Press- Alarm Mode***" in which the following screen appears:



3.15.5.a Register 350Eh (13582 dec) - Maximum pressure

Address	Name	Type	Range	Default	Unit	Read	Write
350Eh	**Maximum pressure**	UINT16	1 ... 1000	250	bar	Always	Always

This register contains the upper limit of the pressure, expressed in [Bar]. If the measured pressure surpass this limit for a period of time longer than the one set on the parameter "***Time pressure max***" (register 350Fh),then Drive will signal the ***Overpressure Alarm "FA30"***.

3.15.5.b Register 350Fh (13583 dec) - Time pressure max.

Address	Name	Type	Range	Default	Unit	Read	Write
350Fh	**Time pressure max**	UINT16	1 ... 60000	300	s/ 1000	Always	Always

This register contains a time period, expressed in [ms]. If the measured pressure surpass the limit ***Maximum pressure*** (register 350Eh) for a period of time longer than the one set on this parameter, then the Drive will signal the ***Overpressure Alarm "FA30"***.

3.15.5.c Register 3510h (13584 dec) - Minimum pressure

Address	Name	Type	Range	Default	Unit	Read	Write
3510h	**Minimum pressure**	UINT16	1 ... 1000	8	bar	Always	Always

This register contains the lower limit of the pressure, expressed in [Bar]. If the measured pressure drops below this limit for a period of time longer than the one set on the parameter "***Time pressure min***" (register 3511h),then Drive will signal the ***Underpressure Alarm "FA31"***.

3.15.5.d Register 3511h (13585 dec) - Time pressure min.

Address	Name	Type	Range	Default	Unit	Read	Write
3511h	**Time pressure min**	UINT16	1 ... 60000	5	s	Always	Always

This register contains a time period, expressed in [ms]. If the measured pressure drops below the limit ***Minimum pressure*** (register 3510h) for a period of time longer than the one set on this parameter, then the Drive will signal the ***Underpressure Alarm "FA31"***.

3.15.6 Pressure regulator

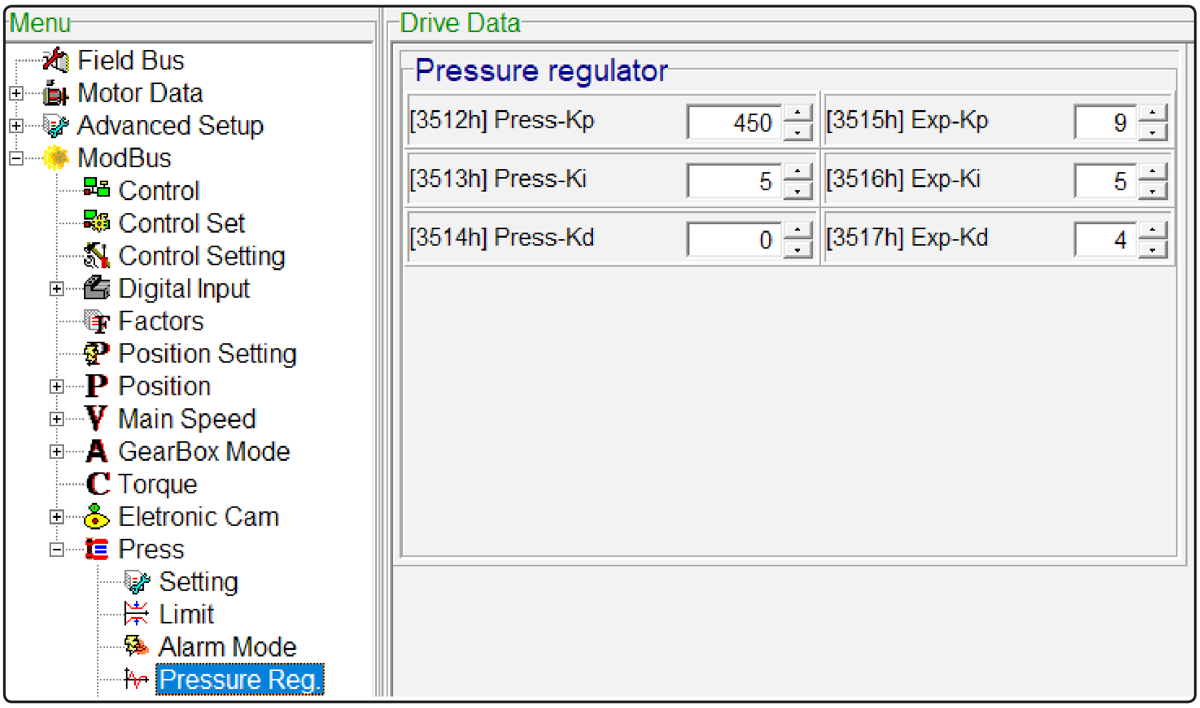
The pressure is regulated using a PID. The proportional, integral and derivative gains of this PID are calculated usign the formula below:

$$Gain_x = K_x 2^{Exponent}$$

There is a constant value ***Kx*** and an ***Exponent*** for each of the PID's gains.We opted for this solution because it allows to make a coarse regulation by changing the exponents, and a fine regulation by changing the constant values.

Address	Name	Type	Read	Write
3512h	**Press-Kp**	UINT16	Always	Always
3513h	**Press-Ki**	UINT16	Always	Always
3514h	**Press-Kd**	UINT16	Always	Always
3515h	**Esp Kp**	UINT16	Always	Always
3516h	**Esp Ki**	UINT16	Always	Always
3517h	**Esp Kd**	UINT16	Always	Always

The constant values ***Kx*** and the ***Exponents*** used to calculate this 3 gains can be set in the Caliper menu "***Press-Pressure regulator***" in which the following screen appears:



3.15.6.a Register 3512h (13586 dec) - Press-Kp

Address	Name	Type	Range	Default	Unit	Read	Write
3512h	**Press-Kp**	UINT16	1 ... 32767	450		Always	Always

This register contains the constant value ***Kx*** of the proportional gain of the Pressure regulator, see formula on paragraph *“3.15.6 Pressure regulator” pag. 123*

3.15.6.b Register 3513h (13587 dec) - Press-Ki

Address	Name	Type	Range	Default	Unit	Read	Write
3513h	**Press-Ki**	UINT16	0 ... 32767	5		Always	Always

This register contains the constant value ***Kx*** of the integral gain of the Pressure regulator, see formula on paragraph *“3.15.6 Pressure regulator” pag. 123*

3.15.6.c Register 3514h (13588 dec) - Press-Kd

Address	Name	Type	Range	Default	Unit	Read	Write
3514h	**Press-Kd**	UINT16	0 ... 32767	0		Always	Always

This register contains the constant value ***Kx*** of the derivative gain of the Pressure regulator, see formula on paragraph *“3.15.6 Pressure regulator” pag. 123*

3.15.6.d Register 3515h (13589 dec) - Esp Kp

Address	Name	Type	Range	Default	Unit	Read	Write
3515h	**Esp Kp**	UINT16	-16 ... 16	9		Always	Always

This register contains the ***Exponent*** of the proportional gain of the Pressure regulator, see formula on paragraph *“3.15.6 Pressure regulator” pag. 123*

3.15.6.e Register 3516h (13590 dec) - Esp Ki

Address	Name	Type	Range	Default	Unit	Read	Write
3516h	**Esp Ki**	UINT16	-16 ... 16	5		Always	Always

This register contains the ***Exponent*** of the integral gain of the Pressure regulator, see formula on paragraph *"3.15.6 Pressure regulator" pag. 123*

3.15.6.f Register 3517h (13591 dec) - Esp Kd

Address	Name	Type	Range	Default	Unit	Read	Write
3517h	**Esp Kd**	UINT16	-16 ... 16	4		Always	Always

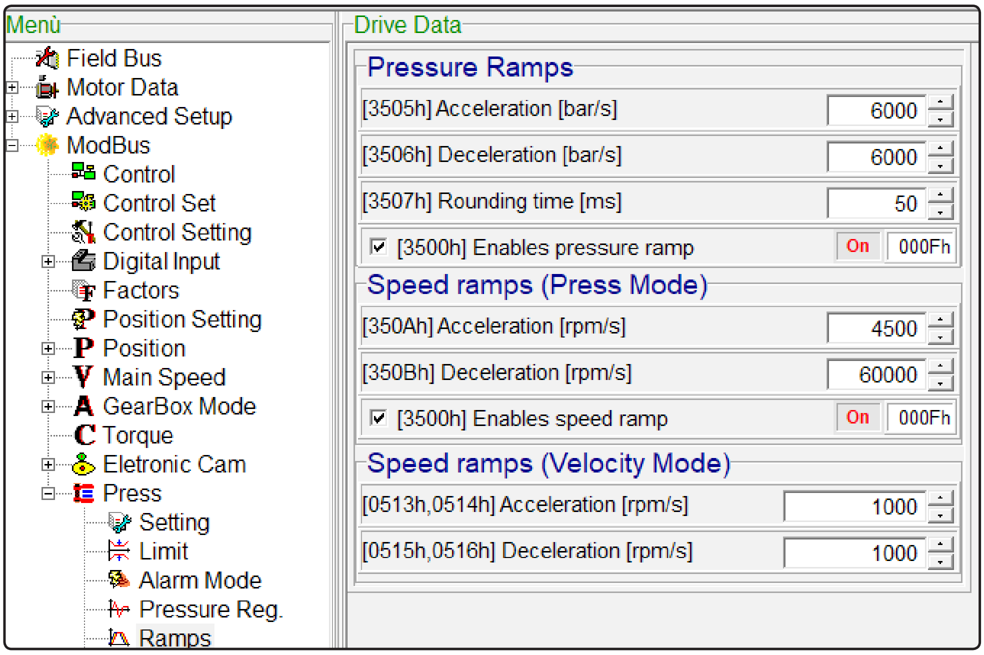
This register contains the ***Exponent*** of the derivative gain of the Pressure regulator, see formula on paragraph *"3.15.6 Pressure regulator" pag. 123*

3.15.7 Ramps

This paragraph lists the parameters used to set the pressure ramp and the speed ramp. These ramps are used to smooth the abrupt variations of the pressure reference and the upper limit of speed at the exit of the pressure regulator , which could occur when these values are inserted, removed or modified.

Address	Name	Type	Read	Write
3505h	**Pressure ramps - acceleration**	UINT16	Always	Always
3506h	**Pressure ramps - deceleration**	UINT16	Always	Always
3507h	**Pressure ramps - Rounding time**	UINT16	Always	Always
350Ah	**Speed ramps - acceleration**	UINT16	Always	Always
350Bh	**Speed ramps - deceleration**	UINT16	Always	Always

The registers 0513h, 0514h, 0515h and 0516h were already described on paragraphs *"3.11.4.b Registers 0513h (1299 dec), 0514h (1300 dec) Acceleration CW" pag. 80* and *"3.11.4.c Registers 0515h (1301 dec) 0516h (1302 dec) Deceleration CW" pag. 80*. They are used when the bit 5 of the *Control Word* is set on 1, their task is to filter the speed reference of the Drive when the latter switched from the Press mode to the Speed mode (see paragraph *"3.15 Press Mode" pag. 115*). All the other parameters can be set in the Caliper menu "***Press- ramps***" in which the following screen appears:



3.15.7.a Register 3505h (13573 dec) - Pressure ramps - acceleration

Address	Name	Type	Range	Default	Unit	Read	Write
3505h	**Pressure ramps - acceleration**	UINT16	1 ... 60000	6000	bar/s	Always	Always

This register is used to set the gradient of the rising phase of the pressure ramp. Using this ramp, when the pressure reference is increased (or inserted) it will not rise immediately, instead it will rise with the gradient set on this register. This gradient is expressed in [bar/s].

3.15.7.b Register 3506h (13574 dec) - Pressure ramps - deceleration

Address	Name	Type	Range	Default	Unit	Read	Write
3506h	**Pressure ramps - deceleration**	UINT16	1 ... 60000	6000	bar/s	Always	Always

This register is used to set the gradient of the descending phase of the pressure ramp. Using this ramp, when the pressure reference is idecreased (or removed) it will not drop immediately, instead it will decrease with the gradient set on this register. This gradient is expressed in [bar/s].

3.15.7.c Register 3507h (13575 dec) - Pressure ramps - rounding time

Address	Name	Type	Range	Default	Unit	Read	Write
3507h	**Pressure ramps - rounding time**	UINT16	0 ... 10000	50	ms	Always	Always

This register is used to set a period of time, expressed in [ms]. At the end of the rising phase or the descending phase of the pressure ramp the pressure reference is smoothed to cancel its discontinuity; this smoothing curve will last for the period of time set on this register.

3.15.7.d Register 350Ah (13578 dec) - Speed ramps - acceleration

Address	Name	Type	Range	Default	Unit	Read	Write
350Ah	**Speed ramps - acceleration**	UINT16	1 ... 60000	4500	rpm/s	Always	Always

This register contains the acceleration applied on the rising phase of the speed ramp; when the speed reference is increased (or inserted) it will not rise immediately, instead it will rise with the acceleration set on this register.

3.15.7.e Register 350Bh (13579 dec) - Speed ramps - deceleration

Address	Name	Type	Range	Default	Unit	Read	Write
350Bh	**Speed ramps - deceleration**	UINT16	1 ... 60000	60000	rpm/s	Always	Always

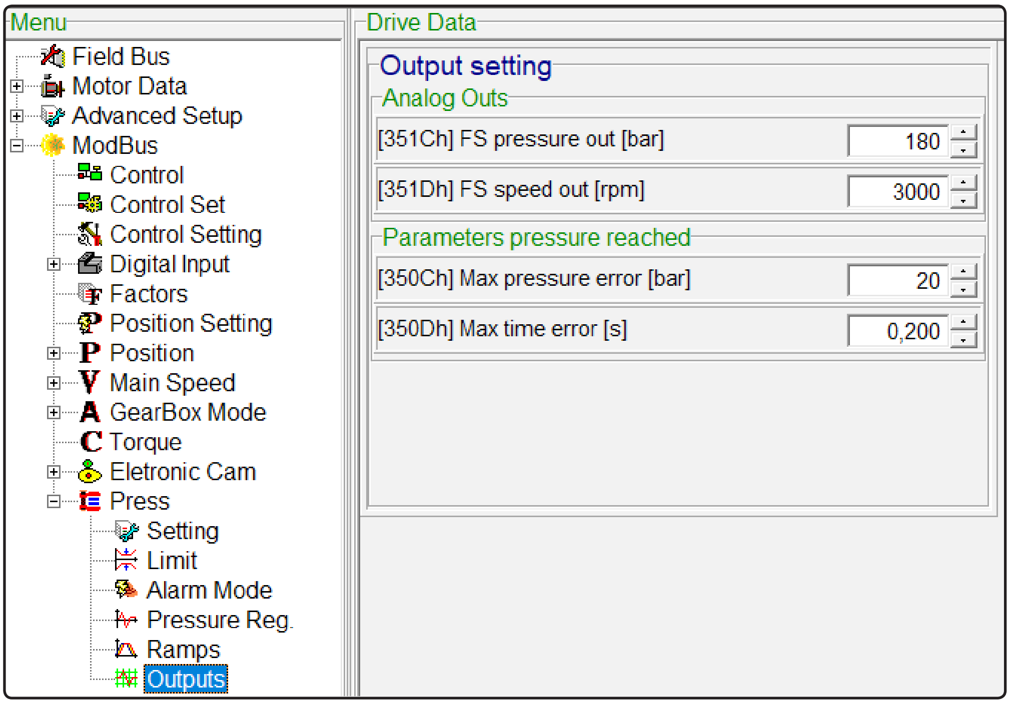
This register contains the deceleration applied on the descending phase of the speed ramp; when the speed reference is decreased (or removed) it will not drop immediately, instead it will decrease with the deceleration set on this register.

3.15.8 Outputs

The parameters listed below are used to set the full scale of the analogic outputs of speed and pressure. They're also used to signal through the *Status word* if the pressure reference has been reached.

Address	Name	Type	Read	Write
351Ch	**FS pressure out**	UINT16	Always	Always
351Dh	**FS speed out**	UINT16	Always	Always
350Ch	**Max pressure error**	UINT16	Always	Always
350Dh	**Max time error**	UINT16	Always	Always

These parameters can be set in the Caliper menu "***Press- Outputs***" in which the following screen appears:



3.15.8.a Register 351Ch (13596 dec) - FS pressure out

Address	Name	Type	Range	Default	Unit	Read	Write
351Ch	**FS pressure out**	UINT16	0 ... 1000	180	bar	Always	Always

This register contains the full scale of pressure (expressed in [Bar]) applied to the analog output which sends the measured pressure. This value is matched with the maximum voltage of the analog output, i.e. 10 [V]. The other values of pressure are converted to voltage using a proportion.

3.15.8.a Register 351Dh (13579 dec) - FS speed out

Address	Name	Type	Range	Default	Unit	Read	Write
351Dh	**FS speed out**	UINT16	1 ... 9999	3000	rpm	Always	Always

This register contains the full scale of speed (expressed in [rpm]) applied to the analog output which sends the measured speed. This value is matched with the maximum voltage of the analog output, i.e. 10 [V]. The other values of speed are converted to voltage using a proportion.

3.15.8.b Register 350Ch (13580 dec) - Max pressure error

Address	Name	Type	Range	Default	Unit	Read	Write
350Ch	**Max pressure error**	UINT16	1 ... 1000	20	bar	Always	Always

This register contains a pressure value,named ***Max pressure error***. When the difference between the measured pressure and the pressure reference is smaller than ***Max pressure error*** for a period of time longer than the set on ***Max time error*** (register 350Dh), the bit 5 of the *Status word* is switched to 1; otherwise is switched to 0.

3.15.8.c Register 350Dh (13581 dec) - Max time error

Address	Name	Type	Range	Default	Unit	Read	Write
350Dh	**Max time error**	UINT16	1 ... 60000	200	s/ 1000	Always	Always

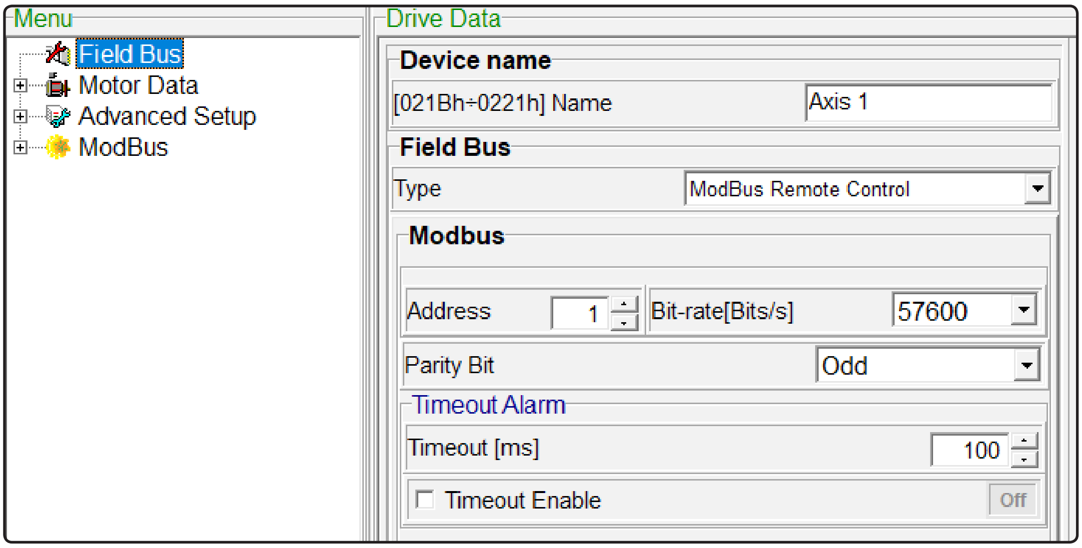
This register contains a time period,named ***Max time error***. When the difference between the measured pressure and the pressure reference is smaller than ***Max pressure error*** (register 350Ch) for a period of time longer than ***Max time error***, the bit 5 of the *Status word* is switched to 1; otherwise is switched to 0.

Ch. 4 INTERNAL PARAMETERS OF THE DRIVE

This chapters lists some specific parameters concerning the Drive. These parameters are not involved with the type of control that you want to use, instead they describe how the Drive operates, how it is identified, how its regulators are tuned, what kind of feedback it use, when it will go in alarm, etc.

4.01 Drive's name

The parameter ***Drive's name*** can be set with the Caliper in the Modbus menù "***Field Bus***", where the following screen appears:



Address	Name	Type	Range	Default	Unit	Read	Write
021Bh	**Drive's name**	UINT16	0 ... 65535			Always	Switch off
021Ch							
021Dh							
021Eh							
021Fh							
0220h							

This parameter, consisting of 6 registers, allows you to change the name of a specific Drive. When you're connected to a group of Drives via a HUB USB this feature allows you to name each of them as you like, making it easier to distinguish them. The size of this parameter cannot exceed 12 Bytes.

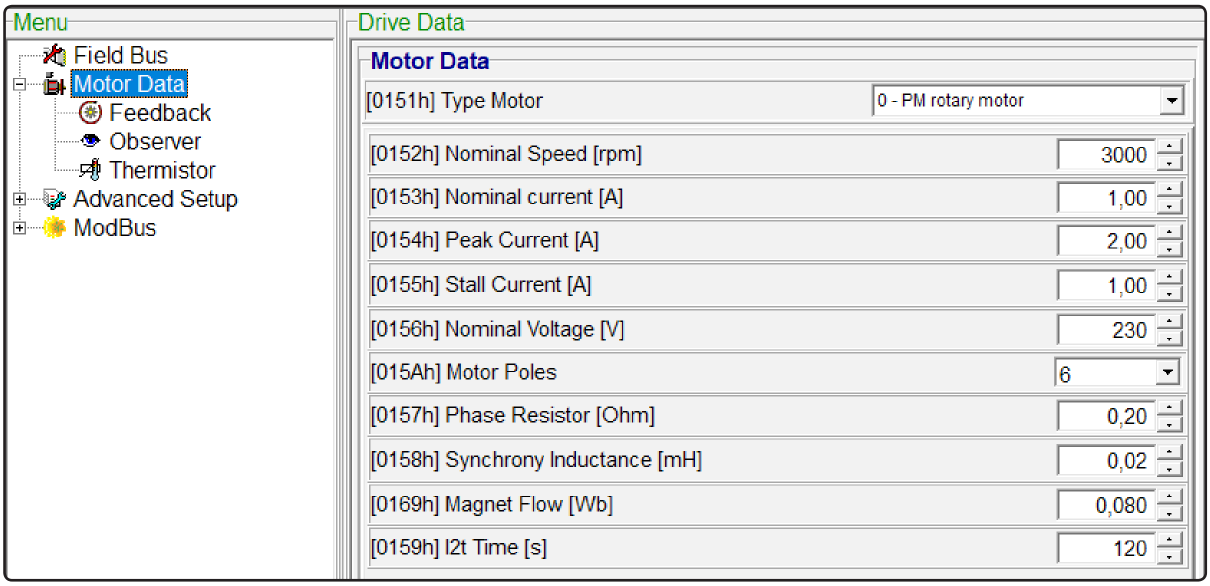
*Note: the other parameters related to the field bus were already described on paragraphs**"1.01 Configuration" pag. 9* *and* *"2.04 The address" pag. 11* *.*

4.02 Motor data

This paragraph lists the parameters used to describe the motor controlled by the Drive.

Address	Name	Type	Read	Write
0151h	Motor type	UINT16	Always	Switch off
0152h	Nominal speed	UINT16	Always	Switch off
0153h	Nominal current	UINT16	Always	Switch off
0154h	Peak current	UINT16	Always	Switch off
0155h	Stall current	UINT16	Always	Switch off
0156h	Nominal Voltage	UINT16	Always	Switch off
0157h	Phase resistance	UINT16	Always	Switch off
0158h	Synchronous inductance	UINT16	Always	Switch off
0159h	I2T time	UINT16	Always	Switch off
015Ah	Motor poles	UINT16	Always	Switch off
0165h	Pole pitch (MSB)	UINT32	Always	Switch off
0166h	Pole pitch (LSB)			
0169h	Magnet flow	UINT16	Always	Always
0178h	Nominal frequency	UINT16	Always	Switch off
0179h	Nominal power factor	UINT16	Always	Switch off
017Ah	Mechanical power	UINT16	Always	Switch off
017Bh	Freq. 1	UINT16	Always	Switch off
017Ch	Freq. 2	UINT16	Always	Switch off
017Dh	Freq. 3	UINT16	Always	Switch off
017Eh	Freq. 4	UINT16	Always	Switch off
017Fh	D.freq. 1	UINT16	Always	Switch off
0180h	D.freq. 2	UINT16	Always	Switch off
0181h	D.freq. 3	UINT16	Always	Switch off
0182h	D.freq. 4	UINT16	Always	Switch off
018Bh	Freq. 1	UINT16	Always	Switch off
018Ch	Freq. 2	UINT16	Always	Switch off
018Dh	Freq. 3	UINT16	Always	Switch off
018Eh	Freq. 4	UINT16	Always	Switch off
018Fh	Freq. 5	UINT16	Always	Switch off
0190h	Volt. 1	UINT16	Always	Switch off
0191h	Volt. 2	UINT16	Always	Switch off
0192h	Volt. 3	UINT16	Always	Switch off
0193h	Volt. 4	UINT16	Always	Switch off
0194h	Volt. 5	UINT16	Always	Switch off
0199h	Current limit Kp	UINT16	Always	Always
019Ah	Current limit Ki	UINT16	Always	Always
019Bh	Slip compensation	UINT16	Always	Always

These parameters can be set in the Caliper menu "***Motor data***" in which the following screen appears:



4.02.1 Register 0151h (337 dec) – Motor type

Address	Name	Type	Range	Default	Unit	Read	Write
0151h	**Motor type**	UINT16	0...2	0		Always	Switch off

This register is used to select what kind of motor is used.

Value	Motor type	Description
0	Rotary motor	This type is used for the SPM brushless motors. When this type is selected, the values reported in the menu "***Data monitor***" of the Caliper will adopt "***rpm***" as unit of measure of the speed and "increments" as unit of measure of the position. In addition, the parameters used by the "***Conversion Factors***" menu (see *"3.09 Conversion factors" pag. 40*) are set up for a rotary motor.
1	Linear motor	When this type is selected, the values reported in the menu "***Data monitor***" of the Caliper will adopt "***mm/s***" as unit of measure of the speed and "***mm***" as unit of measure of the position. In addition, the parameters used by the "***Conversion Factors***" menu (see *"3.09 Conversion factors" pag. 40*) are set up for a linear motor. In the Motor data menu the parameter "***pole pitch***" (registers 0165h,0166h) are included and the number of poles of the motor is set to 2.
2	Brushed DC motor	When this type is selected, the brushed DC motor control is enabled; the unit of measures of speed and position are the same used with the rotary SPM motor. In this mode the positive terminal of the motor must be connected with the ***U*** phase and the negative terminal must be connected with the ***W*** phase of the Drive
3	Induction V-Hz	This type is used for the asynchronous motors. The Drive controls directly the output voltage in order to create the rotating magnetic field necessary to bring the motor to the reference speed. There are no current nor speed feedback; the torque is not controlled
4	Induction FOC	This type is used for the asynchronous motors. Using a model of the motor and the current and speed feedbacks, the Field Oriented Control (FOC) allows you to control the torque of the induction motor, thus achieving better performance than the V-Hz control. *Note: this control is still being implemented*

4.02.2 Register 0152h (338 dec) – Nominal speed

Address	Name	Type	Range	Default	Unit	Read	Write
0152h	**Nominal speed**	UINT16	0...20000	3000	Rpm o mm/s	Always	Switch off

This register contains the nominal speed of the motor expressed in "***rpm***" or "***mm/s***", depending on the type of the controlled motor.

4.02.3 Register 0153h (339 dec) – Nominal current

Address	Name	Type	Range	Default	Unit	Read	Write
0153h	**Nominal current**	UINT16	0...60000	100	A/100	Always	Switch off

This register contains the nominal current of the motor expressed in hundredths of Ampere. The least significant decimal places are intended as decimal fractions of Ampere; for example, the value 100 is read as 1,00 [A].

4.02.4 Register 0154h (340 dec) – Peak current

Address	Name	Type	Range	Default	Unit	Read	Write
0154h	**Peak current**	UINT16	0...60000	200	A/100	Always	Switch off

This register contains the peak current of the motor expressed in hundredths of Ampere. The least significant decimal places are intended as decimal fractions of Ampere; for example, the value 200 is read as 2,00 [A].

4.02.5 Register 0155h (341 dec) – Stall current

Address	Name	Type	Range	Default	Unit	Read	Write
0155h	**Stall current**	UINT16	0...60000	100	A/100	Always	Switch off

This register contains the value of the current absorbed by the motor when the rotor is blocked, expressed in hundredths of Ampere. The least significant decimal places are intended as decimal fractions of Ampere; for example, the value 100 is read as 1,00 [A].

4.02.6 Register 0156h (342 dec) – Nominal voltage

Address	Name	Type	Range	Default	Unit	Read	Write
0156h	**Nominal voltage**	UINT16	0...460	60	V	Always	Switch off

This register contains the value of the nominal motor voltage, expressed in [Volt]. It is not currently used by the firmware.

4.02.7 Register 0157h (343 dec) – Phase resistance

Address	Name	Type	Range	Default	Unit	Read	Write
0157h	**Phase resistance**	UINT16	1...25000	20	Ohm/100	Always	Switch off

This register contains the resistance of a single phase of the motor, expressed in hundredths of [Ohm].

4.02.8 Register 0158h (344 dec) – Synchronous inductance

Address	Name	Type	Range	Default	Unit	Read	Write
0158h	**Synchronous inductance**	UINT16	1...65000	2	mH/ 100	Always	Switch off

This register contains the synchronous inductance of a single phase of the motor, expressed in hundredths of [mH].

4.02.9 Register 0159h (345 dec) – I2T time

Address	Name	Type	Range	Default	Unit	Read	Write
0159h	**I2T time**	UINT16	1...3000	120	sec	Always	Switch off

This register is used to set the maximum time interval, expressed in seconds, in which the motor could absorb a current with twice the value of the ***Nominal current*** (register 0153h). If this current is absorbed for a time interval longer than the one set on this register, the current will be automatically limited to the nominal value and the bit 10 of the "***Alarm Variable***" (register 0303h) is set to 1.

4.02.10 Register 015Ah (346 dec) – Motor poles

Address	Name	Type	Range	Default	Unit	Read	Write
015Ah	**Motor poles**	UINT16	2...50	6		Always	Switch off

This register is used to set the number of poles of the motor. If the controlled motor has an encoder with a number of pulses per turn that is set correctly in the Drive, then the Drive is able to automatically count the number of poles of the motor during the phasing procedure.

4.02.11 Registers 0165h (357 dec) and 0166h (358 dec) – Pole pitch

Address	Name	Type	Range	Default	Unit	Read	Write
0165h **0166h**	**Pole pitch**	UINT32	1 ... 100000	10000	mm/ 1000	Always	Switch off

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.* These registers contain the pole pitch of the motor, measured in thousandth of millimeter. This parameter is used only when you chose to control a linear motor (register 0151h set to 1).

4.02.12 Register 0169h (361 dec) – Magnetic flux

Address	Name	Type	Range	Default	Unit	Read	Write
0169h	**Magnetic flux**	UINT16	1...65000	80	Wb/ 1000	Always	Always

This register is used only when the Drive is controlling a brushless motor (rotative or linear). It contains the magnetic flux linked with a motor phase. It's expressed in hundredths of [Weber].

4.02.13 Register 0178h (376 dec) – Nominal frequency

Address	Name	Type	Range	Default	Unit	Read	Write
0178h	**Nominal frequency**	UINT16	100...3000	500	Hz/ 10	Always	Switch off

This register is used only when the Drive is controlling an induction motor. It contains the nominal frequency of the motor. It's expressed in tenths of [Hertz].

4.02.14 Register 0179h (377 dec) – Nominal power factor

Address	Name	Type	Range	Default	Unit	Read	Write
0179h	**Nominal power factor**	UINT16	500... 10000	850	1/ 10000	Always	Switch off

This register is used only when the Drive is controlling an induction motor. It contains the nominal power factor - cos(phi) - of the motor. It's expressed in tenths of thousandths.

4.02.15 Register 017Ah (378 dec) – Mechanical power

Address	Name	Type	Range	Default	Unit	Read	Write
017Ah	**Mechanical power**	UINT16	1...65536	1000	kW/ 100	Always	Switch off

This register is used only when the Drive is controlling an induction motor. It contains the mechanical power -i.e. the nominal power - of the motor. It's expressed in hundredths of [kW].

4.02.16 Registers 017Bh (379 dec), 017Ch (380 dec), 017Dh (381 dec) and 017Eh (382 dec) – Frequency jumps

Address	Name	Type	Range	Default	Unit	Read	Write
017Bh	**Freq. 1**	UINT16	100... 3000	100	Hz/ 10	Always	Switch off
017Ch	**Freq. 2**	UINT16	100... 3000	200	Hz/ 10	Always	Switch off
017Dh	**Freq. 3**	UINT16	100... 3000	400	Hz/ 10	Always	Switch off
017Eh	**Freq. 4**	UINT16	100... 3000	600	Hz/ 10	Always	Switch off

These registers are used only when the Drive is controlling an induction motor. They're used to set 4 frequency values that the Drive must avoid, because they would create dangerous resonances on the motor. When the Drive is changing the output voltage frequency it will "skip" these values, keeping the frequency to at least a minimum difference from them which could be set on the parameters "***frequency delta***" (see 017Fh,0180h,0181h,0182h).

4.02.17 Registers 017Fh (383 dec), 0180h (384 dec), 0181h (385 dec) and 0182h (386 dec) – Frequency delta

Address	Name	Type	Range	Default	Unit	Read	Write
017Fh	**D.freq. 1**	UINT16	0... 100	0	Hz/ 10	Always	Switch off
0180h	**D.freq. 2**	UINT16	0... 100	0	Hz/ 10	Always	Switch off
0181h	**D.freq. 3**	UINT16	0... 100	0	Hz/ 10	Always	Switch off
0182h	**D.freq. 4**	UINT16	0... 100	0	Hz/ 10	Always	Switch off

These registers are used only when the Drive is controlling an induction motor. They're used to set 4 frequency values,named ***frequency delta***. there is one ***frequency delta*** assigned for every resonance frequencies set on registers 017Bh,017Ch,017Dh e 017Eh. When the Drive has to skip one of the chosen resonance frequencies, it will bring the voltage frequency to a value with a minimum difference from the resonance frequency equals to its ***frequency delta***

4.02.18 Registers 018Bh (395 dec) ... 018Fh (399 dec) and Registers 0190h (400 dec) ... 0194h (404 dec) - V/Hz ramp

Address	Name	Type	Range	Default	Unit	Read	Write
018Bh	**Freq. 1**	UINT16	0...249	0	%/10	Always	Switch off
018Ch	**Freq. 2**	UINT16	1...499	250	%/10	Always	Switch off
018Dh	**Freq. 3**	UINT16	251...749	500	%/10	Always	Switch off
018Eh	**Freq. 4**	UINT16	501...999	750	%/10	Always	Switch off
018Fh	**Freq. 5**	UINT16	751...1000	1000	%/10	Always	Switch off
0190h	**Volt. 1**	UINT16	0...1000	0	%/10	Always	Switch off
0191h	**Volt. 2**	UINT16	0...1000	250	%/10	Always	Switch off
0192h	**Volt. 3**	UINT16	0...1000	500	%/10	Always	Switch off
0193h	**Volt. 4**	UINT16	0...1000	750	%/10	Always	Switch off
0194h	**Volt. 5**	UINT16	0...1000	1000	%/10	Always	Switch off

These registers are used only when the Drive is controlling an induction motor via the "***V-Hz***" control. They constitute the x and y coordinates of 5 points, (Freq.1, Volt.1) ... (Freq.5, Volt.5). Interpolating these points the Drive calculates a function that reduces the voltage supplied to the motor with the decrease of the frequency. With the default values, the Drive will reduce the voltage proportionally to the frequency, keeping the ratio between them costant during the defluxing stage.

4.02.19 Register 0199h (409 dec) - Current limit Kp

Address	Name	Type	Range	Default	Unit	Read	Write
0199h	**Current limit Kp**	UINT16	0...65535	1200		Always	Always

This register is used only when the Drive is controlling an induction motor. It's the proportional gain value of a PID regulator, used to bring the current to its nominal value

4.02.20 Register 019Ah (410 dec) - Current limit Ki

Address	Name	Type	Range	Default	Unit	Read	Write
019Ah	**Current limit Ki**	UINT16	0...65535	100		Always	Always

This register is used only when the Drive is controlling an induction motor. It's the integral gain value of a PID regulator, used to bring the current to its nominal value

4.02.21 Register 019Bh (411 dec), Slip compensation

Address	Name	Type	Range	Default	Unit	Read	Write
019Bh	**Slip compensation**	UINT16	0...2000	1000	%/10	Always	Always

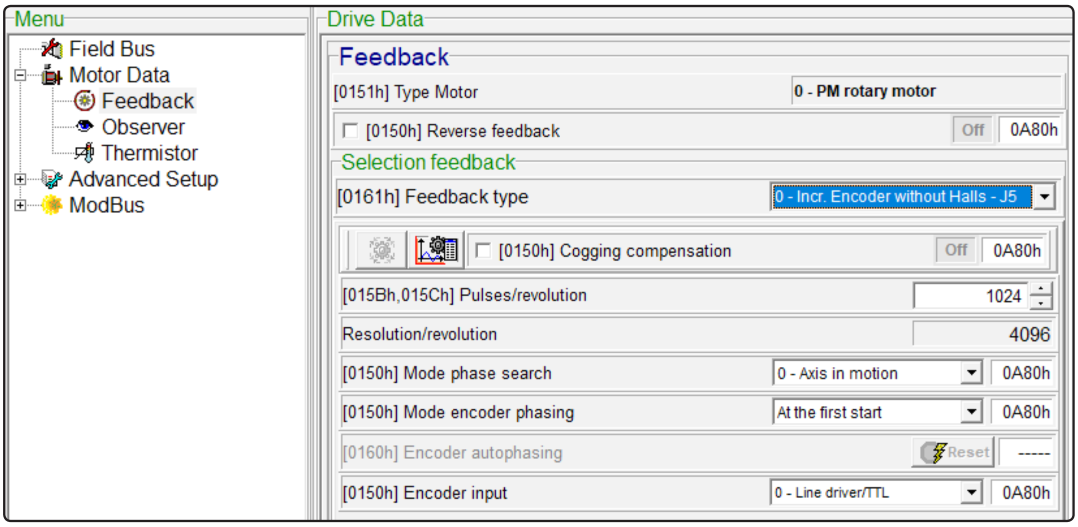
This register is used only when the Drive is controlling an induction motor. It's a coefficient that is multiplied with the rotating magnetic field generated from the stator, in order to estimate the mechanical speed of the rotor

4.03 Feedback parameters

This paragraph lists the parameters related to the position sensor of the motor.

Address	Name	Type	Read	Write
0150h	**Flags**	UINT16	Always	Switch off
015Bh	**Pulses/revolution (MSB)**	UINT32	Always	Switch off
015Ch	**Pulses/revolution (MSB)**			
0160h	**Encoder autophasing reset**	INT16	Always	Always
0161h	**Feedback type**	UINT16	Always	Switch off
0162h	**Offset**	UINT16	Always	Always
0163h	**Bit single-turn**	UINT16	Always	Switch off
0164h	**Bit multi-turn**	UINT16	Always	Switch off
0167h	**Application offset (MSB)**	INT32	Always	Always
0168h	**Application offset (LSB)**			
016Ah	**Covariance current**	UINT16	Always	Always
016Bh	**Covariance speed**	UINT16	Always	Always
016Ch	**Covariance phase**	UINT16	Always	Always
016Dh	**Covarianza noise**	UINT16	Always	Always
016Eh	**Zero crossing**	UINT16	Always	Always
016Fh	**Starting current**	UINT16	Always	Always
0170h	**Current ramp**	UINT16	Always	Always
0171h	**Stationing current**	UINT16	Always	Always
0172h	**Initial speed max**	UINT16	Always	Always
0173h	**Speed hysteresis**	UINT16	Always	Always
0174h	**Speed ramp**	UINT16	Always	Always
0175h	**Fault measured speed**	UINT16	Always	Always
0176h	**Initial delay**	UINT16	Always	Always
0177h	**Resolver poles**	UINT16	Always	Switch off
01ACh	**Serial frequency encoder (MSB)**	UINT32	Always	No
01ADh	**Serial frequency encoder (LSB)**			
030Ch	**Encoder position (MSB)**	INT32	Always	No
030Dh	**Encoder position (LSB)**			
031Bh	**Resolver state**	INT16	Always	No
031Ch	**Resolver phase**	INT16	Always	No

These parameters can be set in the Caliper menu "***Motor data- Feedback***" in which the following screen appears:



4.03.1 Register 0150h (336 dec)- Flags

Address	Name	Type	Range	Default	Unit	Read	Write
0150h	**Flags**	UINT16	0..12	0	Bit	Always	Switch off

This register contains various flags, most of which are used to set the motor feedback. These flags are listed in the table below:

Bit	Name	0150h - Bit description
0	***Connection type***	0: star 1: triangle
1	***Reverse Feedback***	0: normal feedback 1: reversed feedback
2	***Cogging compensation***	0: no cogging compensation 1: cogging compensation
3	***Encoder input***	0: Line driver/TTL 1: Open collector/Push Pull
4,5	***Modo encoder phasing (without Hall)***	00: At the first start 01: At every start 10: After each reset
6,7	***Resolver precision***	00: 10 Bit 01: 12 Bit 10: 14 Bit 11: 16 Bit
8,9	***Encoder precision***	00: 256 01: 1024 10: 4096 11: 16384
10	***Reverse output encoder***	0: normal simulated encoder 1: reversed simulated encoder
11	***Check electrical angle***	0: Off 1: On
12	***mode phase search (without Hall)***	0: Axis in motion 1: Standstill axis
13...15	***reserved***	

- The parameter ***resolver precision*** allows you to set the resolver resolution in a range from 10 to 16 bit. It's default value is 14 bit, which is suited for most of the applications. For applications where the nominal speed reach up to 5000 [rpm] you can set the resolution to 16 bit in order to increase the performance of the position control, at the expence of the speed regulator's bandwidth which will be reduced (it will become 4 times smaller with respect of the bandwidth obtained with a resolution of 14 bit).
- The parameter ***encoder precision*** allows you to set the number of pulses per (mechanical) rotation (ppr) produced by the output of the simulated encoder with index pulse. It's possible to set a number of pulses per rotation less than or equal to the number chosen for the feedback.

4.03.2 Registers 015Bh (347 dec) 015Ch (348 dec) Pulses per revolution (or pole pitch)

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
015Bh 015Ch	**Pulses/ revolution**	UINT32	64 ... 100FFFFh	1024 (dec)		Always	Switch off

These registers are used to set the number of pulses produced by the encoder when the motor complete one revolution or, in case of a linear motor, when it moves for a length equal to the pole pitch.

4.03.3 Register 0160h (352 dec) – Encoder autophasing reset

Address	Name	Type	Range	Default	Unit	Read	Write
0160h	**Encoder auto-phasing reset**	INT16	0...1	0		Always	Always

This register is used only when the Feedback type is set on "***Incremental Encoder without Halls***". At the end of the phasing procedure this register is set to 1. Bringing back this register to 0 the previuous phasing of the motor will be resetted; the Drive will excecute another phasing procedure at the next start command.

4.03.4 Register 0161h (353 dec) – Feedback type

Address	Name	Type	Range	Default	Unit	Read	Write
0161h	**Feedback type**	UINT16	0...3	0		Always	Switch off

This register is used to set what type of position sensor is used as feedback by the motor.

Feedback type
0 - Incremental encoder without Halls
Using this type you have to set the number of pulses of the encoder on the parameter ***Pulses per revolution*** (registers 015Bh, 015Ch). At the first start command the phasing procedure will be executed for few seconds; this procedure is necessary to align the rotor magnet in the correct position in order to obtain the best torque control. This procedure involves a rotation of a certain angle (max 180 °/ n° of poles pair) or a shift if the motor is linear, so there must be no obstacles limiting the motor movement because they would cause an error in the calculation of the optimal position for the motor control. At the end of the procedure the parameter ***Encoder autophasing reset*** (register 0160h) is set to 1; to repeat the procedure at the next power on just reset this last parameter.
1 - Incremental encoder with Halls
Using this type you have to set the number of pulses of the encoder on the parameter ***Pulses per revolution*** (registers 015Bh, 015Ch). The rotor magnet phase must be set in degrees [°] in the parameter ***Offset*** (register 0162h) or it could be automatically calculated by pushing the button "***sensor pos autophasing*** ", which starts the phasing procedure of the position sensor, counts the number of poles and calculates the offset of the magnetic field of the rotor.
2 - Halls only
This type means that the position feedback consists of hall sensors only. In this situation the performance of the control are very limited because without the encoder the motor must be controlled with a Trapezoidal current instead of a Sinusoidal one.

Feedback type
3 - Encoder SSI
With this type of sensor the position is acquired in a serial mode. You have to set on the Drive the number of bits of the position resolution provided by the sensor manufacturer, which have to be set in the parameters "***Bit single turn"*** (register 0163h), "***Bit multi turn"*** (register 0164h). The rotor magnet phase could be manually set, in degrees [°], on parameter "***Offset"*** (register 0162h) or it could be automatically calculated by pushing the button "***sensor pos autophasing*** ". The absolute position measured by the encoder is reported on the parameter "***Encoder position"*** (registers 030Ch, 030Dh); this value is reduced by the value of "***Application offset***" (registri 0167h, 0168h), and the obtained result represents the value of the measured position used as feedback for the positioner applications. by acting on the "***Application offset***" is possible to move the position 0 of the application.
4 - Sensorless
This type is used when the motor has no position sensor. The Drive will use the ***Kalman Filter*** in order to estimate the rotor position. The functioning of this Filter and the parameters required to set it will be explained apart in paragraph *"4.03.13 Parametri Feedback- Filtro di Kalman" pag. 138*
5 - Resolver
With this type you have to set the ***Resolver precision*** (register 0150h). The rotor magnet phase must be set in degrees [°] in the parameter ***Offset*** (register 0162h) or it could be automatically calculated by pushing the button "***sensor pos autophasing*** ", which starts the phasing procedure of the position sensor, counts the number of poles and calculates the offset of the magnetic field of the rotor.
6- Encoder Biss
This feedback is an absolute encoder which comunicate using the ***Biss*** protocol. There is no need to set the parameters "***Bit single turn"*** and "***Bit multi turn"*** , because they're automatically read when the Drive turn on. The rotor magnet phase could be manually set, in degrees [°], on parameter "***Offset"*** (register 0162h) or it could be automatically calculated by pushing the button "***sensor pos autophasing*** ".
7-Encoder Endat
This feedback is an absolute encoder which comunicate using the ***Endat*** protocol. There is no need to set the parameters "***Bit single turn"*** and "***Bit multi turn"*** , because they're automatically read when the Drive turn on. The rotor magnet phase could be manually set, in degrees [°], on parameter "***Offset"*** (register 0162h) or it could be automatically calculated by pushing the button "***sensor pos autophasing*** ".
8-Encoder Biss Sin/Cos
This feedback converts the measured position in two analog sinusoidal signals, the "***Sin/Cos***", instead of converting it in two digital signals. This change increase the position resolution. Using only these two analogical signals you can't obtain the absolute position. To get the latter the Drive will read the absolute position from the encoder ***Biss*** when it turns on, and next it will start using the "***Sin/Cos***" : summing the relative position read by the "***Sin/Cos***" with the absolute position initially read by the encoder ***Biss*** the Drive obtains the current absolute position of the motor
9-Encoder Endat Sin/Cos
This feedback converts the measured position in two analog sinusoidal signals, the "***Sin/Cos***", instead of converting it in two digital signals. This change increase the position resolution. Using only these two analogical signals you can't obtain the absolute position. To get the latter the Drive will read the absolute position from the encoder ***Endat*** when it turns on, and next it will start using the "***Sin/Cos***" : summing the relative position read by the "***Sin/Cos***" with the absolute position initially read by the encoder ***Endat*** the Drive obtains the current absolute position of the motor
10-Incr. Encoder Sin/Cos
This feedback converts the measured position in two analog sinusoidal signals, the "***Sin/Cos***", instead of converting it in two digital signals. This change increase the position resolution. Using only these two analogical signals you can't obtain the absolute position. Therefore the Drive has to execute the phasing procedure every times it turns on, as the one described when we use an incremental encoder without Hall as feedback.
11-Encoder SSI Sin/cos
This feedback converts the measured position in two analog sinusoidal signals, the "***Sin/Cos***", instead of converting it in two digital signals. This change increase the position resolution. Using only these two analogical signals you can't obtain the absolute position. To get the latter the Drive will read the absolute position from the encoder ***SSI*** when it turns on, and next it will start using the "***Sin/Cos***" : summing the relative position read by the "***Sin/Cos***" with the absolute position initially read by the encoder ***SSI*** the Drive obtains the current absolute position of the motor

4.03.5 Register 0162h (354 dec) – Offset

Address	Name	Type	Range	Default	Unit	Read	Write
0162h	**Offset**	UINT16	0...36000	0	°/100	Always	Always

This register contains the offset of the position sensor, expressed in degrees[°]. The least significant digits are interpreted as decimal fractions of a degree, for example the value 18000 is intended as 180,00°. The position sensor offset can be automatically calculated by pushing the button "***sensor pos autophasing*** ", (it's in the toolbar above the screen), which starts the phasing procedure of the position sensor and counts the number of poles.

4.03.6 Register 0163h (355 dec) – Bit single turn

Address	Name	Type	Range	Default	Unit	Read	Write
0163h	**Bit single-turn**	UINT16	4...31	12	Count	Always	Switch off

This register contains the number of bits of resolution of the absolute encoder for a single turn of the motor,i.e. a single turn is splitted in a number of sectors equal to the maximum value set on this register, the obtained sectors are the minimum "variation" that the encoder can detect

4.03.7 Register 0164h (356 dec) – Bit Multi turn

Address	Name	Type	Range	Default	Unit	Read	Write
0164h	**Bit multi-turn**	UINT16	0...31	0	Count	Always	Switch off

This register contains the number of bits of resolution of the absolute encoder for many turns of the motor,i.e. it allows you to choose the maximum value of turns of the motor that the Drive will be able to count.

4.03.8 Registers 0167h (359 dec), 0168h (360 dec) – Application offset

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0167h **0168h**	**Application offset**	INT32	± 7FFFFFFFh	0		Always	Always

When an absolute encoder is used, using the "application offset" parameter is possible to shift the position of the index pulse of the application (i.e. the position 0). When you're working with a position control, in fact, the value of the measured position is obtained by subtracting the value of the ***Encoder position*** (registers 030Ch, 030Dh) with this offset:

$$Measured\_position = Encoder\_position - Application\_Offset$$

Pushing the button "***Home***" on the menù "***Feedback***", the actual position will be reset depending on the value read from the parameter "***Encoder position***" (registers 030Ch, 030Dh).

4.03.9 Register 0177h (375 dec) – Resolver poles

Address	Name	Type	Range	Default	Unit	Read	Write
0177h	**Resolver poles**	UINT16	1...8	2		Always	Switch off

This register contains the number of pole couples of the resolver.

4.03.10 Registers 01ACh (428 dec), 01ADh (429 dec) - Serial frequency encoder

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Indirizzo	Nome	Tipo	Range	Default	Unit	Read	Write
01ACh 01ADh	**Serial frequency encoder**	UINT32	0 .... FFFFFFFFh	0		Always	No

This register contains the frequency of the clock used by the encoder to communicate with the Drive.

4.03.11 Registers 030Ch (780 dec), 030Dh (781 dec) – Encoder position

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
030Ch 030Dh	**Encoder position**	INT32	± 7FFFFFFFh	0		Always	No

This read-only parameter shows the absolute value of the measured position read by the serial encoder SSI.

4.03.12 Register 031Bh (795 dec) – Resolver state

Address	Name	Type	Range	Default	Unit	Read	Write
031Bh	**Resolver state**	UINT16	0 ... 65535	0		Always	No

This read-only parameter shows the current state of the resolver.

Value	Description
0	Configuration parity error
1	Phase error lock range
2	Velocity max tracking rate
3	Tracking error threshold
4	Sin/cos mismatch threshold
5	Sin/cos ovverrange threshold
6	Sin/cos below threshold
7	Sin/cos clipped

4.03.13 Register 031Ch (796 dec)– Resolver phase

Address	Name	Type	Range	Default	Unit	Read	Write
031Ch	**Resolver phase**	INT16	0 ... 65535	0		Always	No

This register contains the current position angle (expressed in increments) of the resolver.

4.03.14 Feedback parameters- Kalman filter

When the Drive has to perform a "***sensorless***" control of the motor (there aren't any position feedbacks), in order to estimate the speed and the position it uses the ***Kalman filter***.

The Kalman filter is an ***Observer*** which use a model of the motorto estimate its state variables from its inputs. In our case the state variables are the currents, the speed and the position, while the inputs are the supplied voltages. The estimation of this filter is corrected at each step using a sort of "feedback" based on the estimation errors between the measured currents and the estimated currents (you can use as feedback only the measurable state variables).

In order to make this filter works you need an accurate model of the motor, therefore you have to set correctly its parameters (resistance, inductance, magnetic flux, number of pole pairs).

The kalman filter resembles the Luenberger observer, but it has three major differences:

» The Kalman filter allows you to take into account the "***measurement errors***". In this application they refer only to the noisein the measurement of the current. On register 016Dh you have to write the error that you expect to find when you're measuring the current.

» The Kalman filter allows you to take into account the "***model errors***", which inevitably appear when the real parameters of the motor differ form the ones set Via Caliper. These errors will affect the estimated values of currents, speed and position. On registers 016Ah, 016Bh,016Ch you have to write the model errors that you expect will appear estimating the state variables.

» The "gains" that multiplies with the "feedbacks" of the Kalman filter change over time, in order to make the best possible estimation. The expected model's and measurement's errors that you set via Caliper represent the reliability of our estimations and measurements. Depending on their value, the Kalman filter will change its gains to achieve the best possible estimation.

At very low speed the estimation made by the Kalman Filter will not be very reliable. Therefore when the speed drops below a certain threshold the Drive will switch to a "***V-Hz***" control, i.e. it will directly produce the voltage necessary to bring (at regime) the motor to the selected reference speed, without estimating it's position.

4.03.14.a Register 016Ah (362 dec) – Covariance current

Address	Name	Type	Range	Default	Unit	Read	Write
016Ah	**Covariance current**	UINT16	1 ... 65000	1000	1/100	Always	Always

This register is used by the Kalman filter, i.e. when the Drive uses a sensorless control (see *"4.03.14 Feedback parameters- Kalman filter" pag. 141*).

The "***current covariance*** " represents the current estimation error that is expected to appear due to of an approximate model of the motor (for example the set resistance and the inductance could be inaccurate). An high covariance means that you expect a rough estimation of the current, on the contrary a low covariance means that you expect an accurate estimation.

4.03.14.b Register 016Bh (363 dec) – Covariance speed

Address	Name	Type	Range	Default	Unit	Read	Write
016Bh	**Covariance speed**	UINT16	1 ... 65000	100	1/100	Always	Always

This register is used by the Kalman filter, i.e. when the Drive uses a sensorless control (see *"4.03.14 Feedback parameters- Kalman filter" pag. 141*).

The "***speed covariance*** " represents the speed estimation error that is expected to appear due to an approximate model of the motor (for example the set resistance and the inductance could be inaccurate). An high covariance means that you expect a rough estimation of the speed, on the contrary a low covariance means that you expect an accurate estimation.

4.03.14.c Register 016Ch (364 dec) – Covariance phase (position)

Address	Name	Type	Range	Default	Unit	Read	Write
016Ch	**Covariance phase (position)**	UINT16	1 ... 65000	80	1/10000	Always	Always

This register is used by the Kalman filter, i.e. when the Drive uses a sensorless control (see *"4.03.14 Feedback parameters- Kalman filter" pag. 141*).

The "***phase covariance*** " represents the position estimation error that is expected to appear due to an approximate model of the motor (for example the set resistance and the inductance could be inaccurate). An high covariance means that you expect a rough estimation of the position, on the contrary a low covariance means that you expect an accurate estimation.

4.03.14.d Register 016Dh (365 dec) – Covariance noise

Address	Name	Type	Range	Default	Unit	Read	Write
016Dh	**Covariance noise**	UINT16	1 ... 65000	100	1/100	Always	Always

This register is used by the Kalman filter, i.e. when the Drive uses a sensorless control (see *"4.03.14 Feedback parameters- Kalman filter" pag. 141*).

The "***noise covariance*** " represents the value of disturbance that is expected to affect your current measurement. An high covariance means that you expect an innacurate measurement of the current, on the contrary a low covariance means that you expect an accurate measurement.

4.03.14.e Register 016Eh (366 dec) – Zero crossing

Address	Name	Type	Range	Default	Unit	Read	Write
016Eh	**Zero crossing**	UINT16	0 ... 1	0		Always	Always

This register is used when the Drive uses a sensorless control (see *"4.03.14 Feedback parameters- Kalman filter" pag. 141*). It's used to set how the Drive will switch from the Kalman filter control to the "***V-Hz***" control.

Bit	description
[0] = Stop/Start	If the absolute value of the reference speed is lower than the difference between the "***initial speed max***" (0172h) and the "***speed hysteresis***" (0173h) then the Drive switches immediately to the ***V/Hz*** control.
[1] = Continuous	If the absolute value of the reference speed is lower than the difference between the "***initial speed max***" (0172h) and the "***speed hysteresis***" (0173h) then a countdown starts; when it terminates the Drive switches to the ***V/Hz*** control.

4.03.14.f Register 016Fh (367 dec) – Starting current

Address	Name	Type	Range	Default	Unit	Read	Write
016Fh	**Starting current**	UINT16	1 ... 2000	1000	%/10	Always	Always

This register is used when the Drive uses a sensorless control (see *"4.03.14 Feedback parameters- Kalman filter" pag. 141*). When starting the motor with a sensorless control, the Drive uses the "***V/Hz***" control instead the Kalman filter because the latter is not reliable at low speed. this register contains the value of the current (expressed in % with respect to the nominal motor current) that the "***V/Hz***" control will apply during this time.

4.03.14.g Register 0170h (368 dec) – Current ramp

Address	Name	Type	Range	Default	Unit	Read	Write
0170h	**Current ramp**	UINT16	1 ... 60000	400	s/1000	Always	Always

This register is used when the Drive uses a sensorless control (see *"4.03.14 Feedback parameters- Kalman filter" pag. 141*). When starting a motor with a sensorless control, the current applied doesn't switch immediately to the value set on register 016Fh, but it rises with a linear ramp. The value of register 0170h , expressed in seconds, represents how long this ramp will last.

4.03.14.h Register 0171h (369 dec) – Stationing current

Address	Name	Type	Range	Default	Unit	Read	Write
0171h	**Stationing current**	UINT16	0 ... 2000	500	%/10	Always	Always

This register is used when the Drive uses a sensorless control (see *"4.03.14 Feedback parameters- Kalman filter" pag. 141*). It contains the current absorbed by the motor when the latter is standstill and the Drive is at ***SWITCH ON*** state. It's expressed as a percentage of the nominal current.

4.03.14.i Register 0172h (370 dec) – Initial speed max

Address	Name	Type	Range	Default	Unit	Read	Write
0172h	**Initial speed max.**	UINT16	1 ... 10000	100	rpm	Always	Always

This register is used when the Drive uses a sensorless control (see *"4.03.14 Feedback parameters- Kalman filter" pag. 141*). When starting the motor with a sensorless control, the Drive uses the "***V/Hz***" control instead the Kalman filter because the latter s not reliable at low speed. This register contains the speed threshold (expressed in [rpm]) that the motor must reach before switching from the "***V/Hz***" control to the Kalman filter control.

4.03.14.j Register 0173h (371 dec) – Speed hysteresis

Address	Name	Type	Range	Default	Unit	Read	Write
0173h	**Speed hysteresis**	UINT16	1 ... 1000	20	rpm	Always	Always

This register is used when the Drive uses a sensorless control (see *"4.03.14 Feedback parameters- Kalman filter" pag. 141*). The Drive doesn't change immediately from the Kalman filter control to the ***V/Hz*** control when the motor speed drops below the threshold set on register 0172h, but at a lower speed. Therefore there is an hysteresis cycle when the Drive switches from one control type to the other. The register 0173h represents indeed the width of this hysteresis cycle, i.e. the difference between "***Initial speed max.***" of register 0172h and the reduced speed below which the Drive change to the ***V/Hz*** control.

4.03.14.k Register 0174h (372 dec) – Speed ramp

Address	Name	Type	Range	Default	Unit	Read	Write
0174h	**Speed ramp**	UINT16	1...60000	1000	rpm/s	Always	Always

This register is used when the Drive uses a sensorless control (see *"4.03.14 Feedback parameters- Kalman filter" pag. 141*). When starting a motor with a sensorless control, the speed applied doesn't switch immediately to the value set on register 0172h, but it rises with a linear ramp. The value of register 0174h , expressed in [rpm/s],represents the acceleration of this ramp.

4.03.14.l Register 0175h (373 dec) – Fault measured speed

Address	Name	Type	Range	Default	Unit	Read	Write
0175h	**Fault measured speed**	UINT16	1...65000	2000	rpm	Always	Always

This register is used when the Drive uses a sensorless control (see *"4.03.14 Feedback parameters- Kalman filter" pag. 141*). It represents the maximum permissible speed error when the Drive uses the Kalman Filter, above which the Drive switches directly to the ***V/Hz*** control.

4.03.14.m Register 0176h (374 dec) – Initial delay

Address	Name	Type	Range	Default	Unit	Read	Write
0176h	**Initial delay**	UINT16	1...10000	200	s/1000	Always	Always

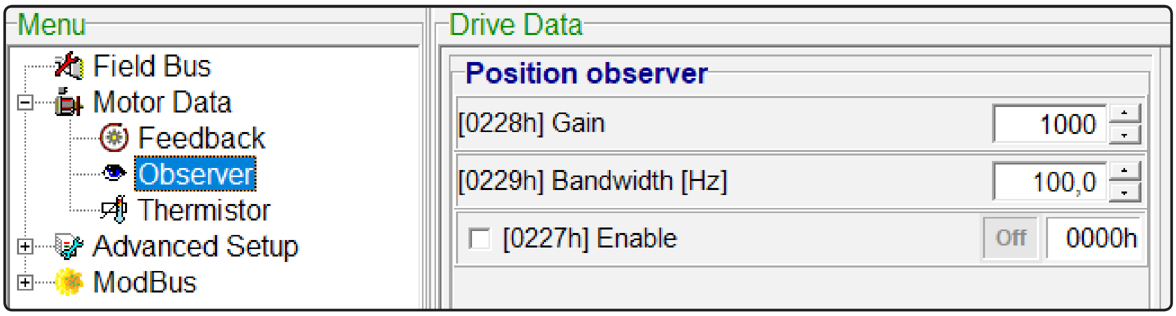
This register is used when the Drive uses a sensorless control (see *"4.03.14 Feedback parameters- Kalman filter" pag. 141*). It represents the period of time the Drive has to wait before starting to follow the speed reference. This delay is necessary to permit to the motor to get in the initial orientation.

4.04 Observer parameters

In this section is described the Observer, a system used by our Drives to improves the resolution of the position measured by the feedback.

Address	Name	Type	Read	Write
0227h	**Enable Observer**	UINT16	Always	Always
0228h	**Gain**	UINT16	Always	Always
0229h	**Bandwidth**	UINT16	Always	Always

The parameters related to the Observer can be set in the Caliper menu "***Motor data-Observer***" in which the following screen appears:



The Observer can be used with every kind of feedback, and it can increment the position resolution on a single turn up to 16 bit. This system could be useful to reduce the motor noise caused by the speed regulator in certain applications, while maintaining good static and dynamic performances.

We recommend its use when:

- the position feedback is provided by an encoder with a resolution below 1000 ppr or 12 bit;
- the position feedback is provided by hall sensors only

4.04.1 Register 0227h (551 dec) – Enable Observer

Address	Name	Type	Range	Default	Unit	Read	Write
0227h	**Enable Observer**	UINT16	0...65535	0		Always	Always

This register contains a flag used to enable or disable the Observer.

4.04.2 Register 0228h (552 dec) – Gain

Address	Name	Type	Range	Default	Unit	Read	Write
0228h	**Gain**	UINT16	1...65535	1000		Always	Always

This register contains the value of the Observer gain.

4.04.3 Register 0229h (553 dec) – Bandwidth

Address	Name	Type	Range	Default	Unit	Read	Write
0229h	**Bandwidth**	UINT16	1...60000	1000	Hz/ 10	Always	Always

This register is used to set the Observer bandwidth, a sort of filter used to eliminate the truncation error from the measured position, which could be remarkable when a sensor with low resolution is used.

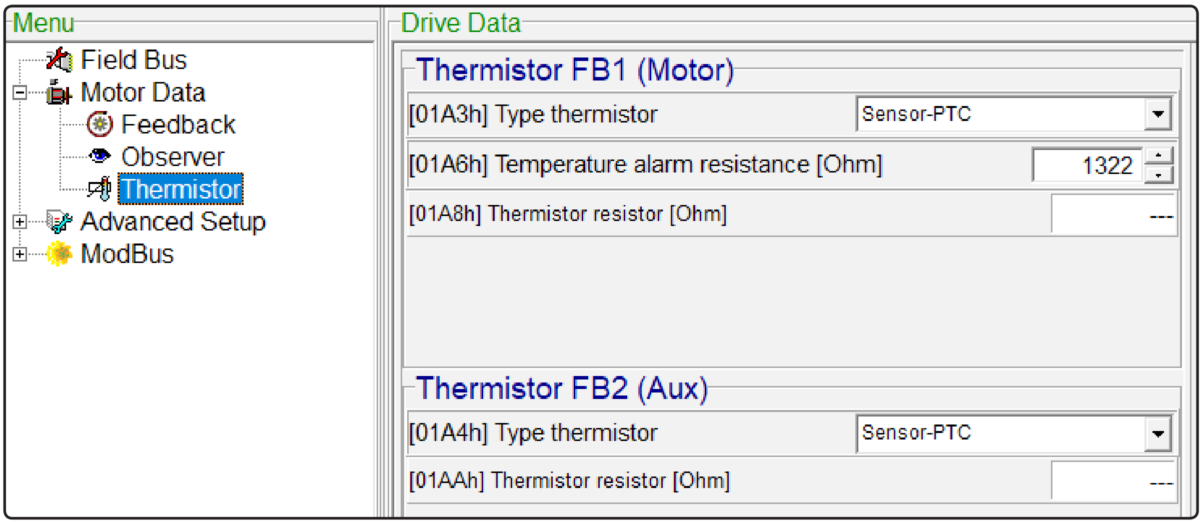
N.B: excessively decreasing the bandwidth of the Observer could cause instability in the speed regulator with the loss of control of the motor, with the risk of damaging the application where it's mounted.

4.05 Thermistor parameters

This paragraph lists the parameters related to the thermistor.

Address	Name	Type	Read	Write
01A3h	**Type thermistor FB1 (motor)**	UINT16	Always	Always
01A4h	**Type thermistor FB2 (Aux)**	UINT16	Always	Always
01A5h	**Temperature alarm threshold FB1 (motor)**	UINT16	Always	Always
01A6h	**Temperature alarm resistance FB1 (motor)**	UINT16	Always	Always
01A7h	**Thermistor temperature FB1 (motor)**	UINT16	Always	No
01A8h	**Thermistor resistance FB1 (motor)**	UINT16	Always	No
01A9h	**Thermistor temperature FB2 (Aux)**	UINT16	Always	No
01AAh	**Thermistor resistance FB2 (Aux)**	UINT16	Always	No

These parameters can be set in the Caliper menu "***Motor data-Thermistor***" in which the following screen appears:



4.05.1 Register 01A3h (419 dec) – Type thermistor FB1 (motor)

Address	Name	Type	Range	Default	Unit	Read	Write
01A3h	**Type thermistor FB1 (motor)**	UINT16	0 ... 2	0		Always	Always

This register is used to set what type of thermistor is mounted on the motor, if present.

Type -thermistor FB1 (motor): description	
0	No thermal sensor
1	Sensor PTC
2	Sensor KTY84

4.05.2 Register 01A4h (420 dec) – Type thermistor FB2 (Aux)

Address	Name	Type	Range	Default	Unit	Read	Write
01A4h	**Type thermistor FB2 (Aux)**	UINT16	0 ... 2	0		Always	Always

This register is used to set what type of auxiliary thermistor is used, if present.

Type -thermistor FB2 (aux): description	
0	No thermal sensor
1	Sensor PTC
2	Sensor KTY84

4.05.3 Register 01A5h (421 dec) – Temperature alarm threshold FB1 (Motor)

Address	Name	Type	Range	Default	Unit	Read	Write
01A5h	**Temperature alarm threshold FB1 (motor)**	UINT16	1...10000	1300	°/10	Always	Always

This register is used to set a temperature threshold for the KTY84 sensor mounted on the motor, if the measured temperature exceeds this threshold an alarm appears.

4.05.4 Register 01A6h (422 dec) – Temperature alarm resistance FB1 (Motor)

Address	Name	Type	Range	Default	Unit	Read	Write
01A6h	**Temperature alarm resistance FB1 (motor)**	UINT16	1...65535	1640	Ohm	Always	Always

This register is used to set the maximum resistance allowed for the PTC sensor mounted on the motor, if its resistance exceeds this value an alarm appears.

4.05.5 Register 01A7h (423 dec) – Thermistor temperature FB1 (Motor)

Address	Name	Type	Range	Default	Unit	Read	Write
01A7h	**Thermistor temperature FB1 (motor)**	INT16	-32768 ... 32767		°	Always	No

This register reports the temperature measured by the Kty84 sensor mounted on the motor.

4.05.6 Register 01A8h (424 dec) – Thermistor resistance FB1 (Motor)

Address	Name	Type	Range	Default	Unit	Read	Write
01A8h	**Thermistor resistance FB1 (motor)**	UINT16	0...65535	---	Ohm	Always	No

This register reports the measured resistance of the PTC sensor mounted on the motor.

4.05.7 Register 01A9h (425 dec) – Thermistor temperature FB2 (Aux)

Address	Name	Type	Range	Default	Unit	Read	Write
01A9h	**Thermistor temperature FB2 (Aux)**	INT16	-32768 ... 32767		°	Always	No

This register reports the temperature measured by the auxiliary Kty84 sensor.

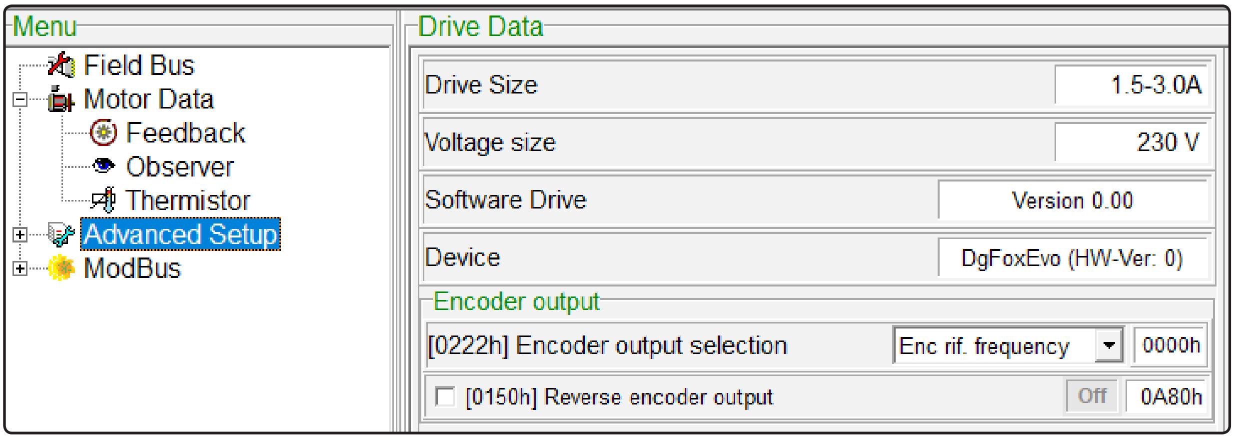
4.05.8 Register 01AAh (426 dec) – Thermistor resistance FB2 (Aux)

Address	Name	Type	Range	Default	Unit	Read	Write
01AAh	**Thermistor resi-stance FB2 (Aux)**	UINT16	0...65535	---	Ohm	Always	No

This register reports the measured resistance of the auxiliary PTC sensor.

4.06 Advanced Setup

Selecting the Caliper menù "***Advanced setup***" the following screen appears:



In this screen you can read some useful information regarding the used Drive, such as its rated voltage and current, its model and the installed firmware.

You can also set 2 flag, contained in the registers 0150h and 0222h. These flags are described below.

4.06.1 Register 0150h (336 dec), bit 10 – Reverse encoder output

Address	Name	Type	Range	Default	Unit	Read	Write
0150h	**Reverse encoder output**	UINT16	0..65535		Bit	Always	Switch off

This flag is used to reverse the encoder signal that will be transmitted in output.

value	description
0	normal simulated encoder
1	reversed simulated encoder

4.06.2 Register 0222h (546 dec), bit 4,5 – Encoder output selection

Address	Name	Type	Range	Default	Unit	Read	Write
0222h	**Encoder output selection**	UINT16	0...65535		Bit	Always	Switch off

This flag is used to select what "encoder signal" you want to transmit on the output:

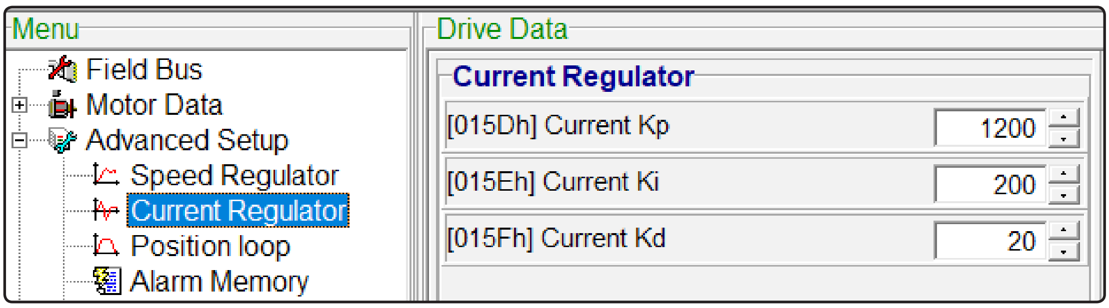
Description of register 222h- bit 4 and 5	
value	**Description**
0	***Enc. rif. frequency***: the Drive transmits on the output a copy of the position reference signal
1	***Enc. sim. resolver***: the Drive transmits on the output the signal of a simulated resolver
2	***Enc. increment+Hall***: the Drive transmits on the output a copy of the measured position signal

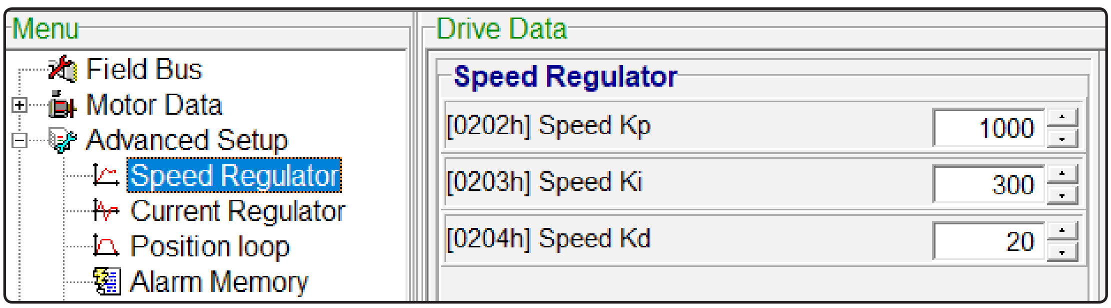
4.07 Speed's and current's regulators

This paragraph describes the parameters used for the tuning of the current's and speed's regulators. The regulators currently used are PI (proportional-integral) regulators. The derivative gains are not used at the moment; they were included because they will be implemented in the future.

Address	Name	Type	Read	Write
015Dh	**Kp current regulator**	UINT16	Always	Always
015Eh	**Ki current regulator**	UINT16	Always	Always
015Fh	**Kd current regulator**	UINT16	Always	Always
0202h	**Kp speed regulator**	UINT16	Always	Always
0203h	**Ki speed regulator**	UINT16	Always	Always
0204h	**Kd speed regulator**	UINT16	Always	Always

These parameters can be set in the Caliper menus "***Advanced setup-Current regulator***" and "***Advanced setup-Speed regulator***" in which the following screens appear:





4.07.1 Register 015Dh (349 dec) – Kp current regulator

Address	Name	Type	Range	Default	Unit	Read	Write
015Dh	**Kp current regulator**	UINT16	1...2000	1000		Always	Always

This register contains the proportional gain value of the current regulator.

4.07.2 Register 015Eh (350 dec) – Ki current regulator

Address	Name	Type	Range	Default	Unit	Read	Write
015Eh	**Ki current regulator**	UINT16	1...2000	400		Always	Always

This register contains the integral gain value of the current regulator.

4.07.3 Register 015Fh (351 dec) – Kd current regulator

Address	Name	Type	Range	Default	Unit	Read	Write
015Fh	**Kd current regulator**	UINT16	1...1000	20		Always	Always

This register contains the derivative gain value of the current regulator. Currently it's not used by the firmware. Non gestito dal firmware attuale.

4.07.4 Register 0202h (514 dec) – Kp speed regulator

Address	Name	Type	Range	Default	Unit	Read	Write
0202h	**Kp speed regulator**	UINT16	1...3000	200		Always	Always

This register contains the proportional gain value of the speed regulator.

4.07.5 Register 0203h (515 dec) – Ki speed regulator

Address	Name	Type	Range	Default	Unit	Read	Write
0203h	**Ki speed regulator**	UINT16	0...3000	100		Always	Always

This register contains the integral gain value of the speed regulator.

4.07.6 Register 0204h (516 dec) – Kd speed regulator

Address	Name	Type	Range	Default	Unit	Read	Write
0204h	**Kd speed regulator**	UINT16	1...1000	20		Always	Always

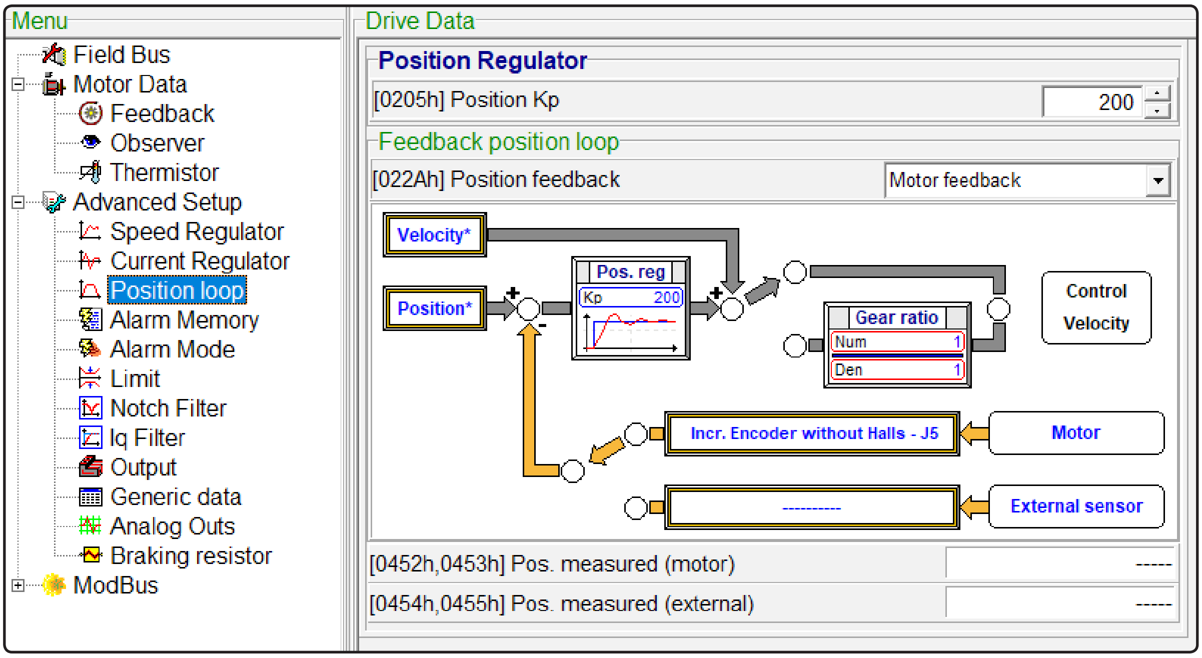
This register contains the derivative gain value of the speed regulator. Currently it's not used by the firmware.

4.08 Position loop

This paragraph lists the parameter used for the tuning of the position regulator.

Address	Name	Type	Read	Write
0205h	**Kp Position regulator**	UINT16	Always	Always
022Ah	**Position feedback**	UINT16	Always	Switch off
022Bh	**Pulses/revolution (MSB)**	UINT32	Always	Switch off
022Ch	**Pulses/revolution (LSB)**			
044Eh	**Gear ratio - num (MSB)**	UINT32	Always	Always
044Fh	**Gear ratio - num (LSB)**			
0450h	**Gear ratio - den (MSB)**	UINT32	Always	Always
0451h	**Gear ratio - den (LSB)**			
0452h	**Pos. measured (motor) (MSB)**	UINT32	Always	No
0453h	**Pos. measured (motor) (LSB)**			
0454h	**Pos. measured (extern) (MSB)**	UINT32	Always	No
0455h	**Pos. measured (extern) (LSB)**			

These parameters can be set in the Caliper menu "***Advanced setup-position loop***" in which the following screen appears:



4.08.1 Register 0205h (517 dec) – Kp position regulator

Address	Name	Type	Range	Default	Unit	Read	Write
0205h	**Kp position regulator**	UINT16	0...4000	200		Always	Always

This register contains the proportional gain value of the position regulator.

4.08.2 Register 022Ah (554 dec)– Position feedback

Address	Name	Type	Range	Default	Unit	Read	Write
022Ah	**Position feedback**	UINT16	0...2	0		Always	Switch off

This register is used to select what kind of position feedback you want to use

Position feedback		
0	motor feedback	The Drive uses the measured position of the motor as feedback
1	Encoder reference -J7	The Drive uses as feedback the position measured by an external sensor, received on input J7
2	Encoder feedback -J5	The Drive uses as feedback the position measured by an external sensor, received on input J5

4.08.3 Registers 022Bh (555 dec), 022Ch (556 dec)– Pulses/revolution

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
022Bh **022Ch**	**Pulses/ revolution**	UINT32	1...65536	1024		Always	Switch off

These registers contain the number of pulses per revolution received by the external encoder. If an encoder of type "***Channel A-B***" is used, the value of "***pulses per revolution***" will be internally multiplied by 4, because in this mode the Drive can read the encoder pulses with a resolution 4 times higher.

4.08.4 Registers 044Eh (1102 dec), 044Fh (1103 dec)– Gear ratio - num

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
044Eh **044Fh**	**Gear ratio - num**	UINT32	0 ... 7FFFFFFh	1		Always	Always

These two registers constitute the numerator of a reduction ratio which multiplies with the speed produced by the position regulator, which will later become the speed reference of the Drive. This speed is calculated as follow:

$$Reference\_speed = \frac{gear\_ratio\_numerator}{gear\_ratio\_denominator} * Output\_speed$$

4.08.5 Registers 0450h (1104 dec), 0451h (1105 dec)– Gear ratio - den

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0450h **0451h**	**Gear ratio - den**	UINT32	1 ... 7FFFFFFh	1		Always	Always

These two registers constitute the denominator of a reduction ratio which multiplies with the speed produced by the position regulator, which will later become the speed reference of the Drive. This speed is calculated using the formula on paragraph *"4.08.4 Registers 044Eh (1102 dec), 044Fh (1103 dec)– Gear ratio - num" pag. 152*

4.08.6 Registers 0452h (1106 dec), 0453h (1107 dec)– Pos. measured (motor)

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0452h **0453h**	**Pos. measured (motor)**	UINT32	±7FFFFFFFh			Always	No

These two registers contain the value of the normalized measured position of the motor, with a resolution of 16 bit.

4.08.7 Registers 0454h (1108 dec), 0455h (1109 dec)– Pos. measured (extern)

These registers are a unique 32-bit variable, and therefore should be managed as described in Section *"3.01 Managing 32 bits variables" pag. 16.*

Address	Name	Type	Range	Default	Unit	Read	Write
0454h **0455h**	**Pos. measured (extern)**	UINT32	±7FFFFFFFh			Always	No

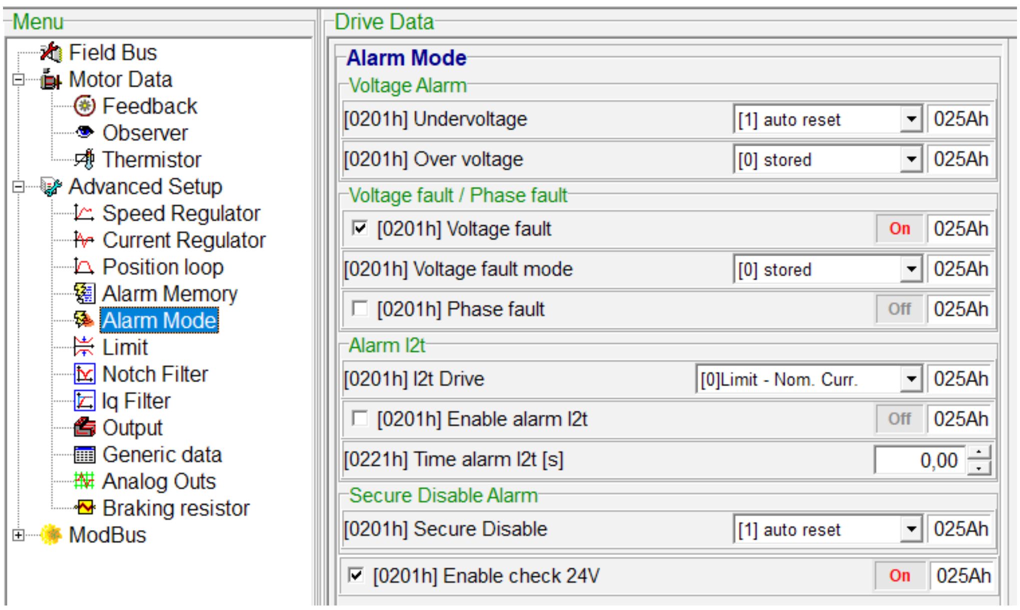
These two registers contain the value of the normalized measured position of an absolute external encoder, with a resolution of 16 bit.

4.09 Alarm mode

This paragraph lists the parameters used to manage separately the alarms of the Drive.

Address	Name	Type	Read	Write
0201h	**Alarms mode**	UINT16	Always	Always
0221h	**Time alarm I2t**	UINT16	Always	Always

These parameters can be set in the Caliper menu "***Advanced setup-Alarm mode***" in which the following screen appears:



4.09.1 Register 0201h (513 dec) – Alarms mode

Address	Name	Type	Range	Default	Unit	Read	Write
0201h	**Alarms mode**	UINT16	0...65535	0A		Always	Always

These register contains various flags used to set the Drive’s alarms, as showed in the following table

Bit	Value	Description
0		**Alarm mode over voltage**
	Value	**Description**
	0	***Stored***: the “***over voltage***” alarm is signaled and the Drive goes in “***Fault***” state. You have to command a reset or restart the Drive if you want to restore its state.
	1	***Auto reset***: the “***over voltage***” alarm is signaled and the Drive goes in “***Fault***” state. When the voltage drops below the “***over voltage***” threshold, the Drive exits from the “***Fault***” state without the need of an external reset.
1		**Alarm mode undervoltage**
	Value	**Description**
	0	***Stored***: the “***undervoltage***” alarm is signaled and the Drive goes in “***Fault***” state. You have to command a reset or restart the Drive if you want to restore its state.
	1	***Auto reset***: the “***undervoltage***” alarm is signaled and the Drive goes in “***Fault***” state. When the voltage rises over the “***undervoltage***” threshold, the Drive exits from the “***Fault***” state without the need of an external reset.
2		**Alarm mode I2t drive**
	Value	**Description**
	0	***Limit to nominal current***: when the Drive “***I2t alarm***” occurs, the maximum current deliverable is limited to the value of the Drive’s rated current, and remains there regardless of the evolution of the estimated temperature of the Drive.
	1	***Auto reset cyclic***: if the Drive “***I2t alarm***” occurs, it will be automatically cancelled as soon as the estimated temperature of the Drive drops below the fixed threshold.
3		**Alarm mode secure disable (Only TOMCAT)**
	Value	**Description**
	0	***Stored***: the “***secure disable***” alarm is signaled and the Drive goes in “***Fault***” state. You have to command a reset or restart the Drive if you want to restore its state.
	1	***Auto reset***: the “***secure disable***” alarm is signaled and the Drive goes in “***Fault***” state. When the voltage is restored on the inputs assigned to this function, the Drive exits from the “***Fault***” state without the need of an external reset.
4		**Alarms save**
	Value	**Description**
	0	***off:*** The Drive doesn’t store the alarms have occurred
	1	***on:*** The Drive stores the alarms that have occurred on a list
5		**Enable alarm I2t**
	Value	**Description**
	0	***off***: alarm not enabled
	1	***on***: alarm enabled
6		**Enable check 24 V**
	Value	**Description**
	0	***off***: alarm not enabled
	1	***on***: alarm enabled

<table>
<tr><th>Bit</th><th colspan="2">Description</th></tr>
<tr><td rowspan="4">7</td><td colspan="2">Alarm mode voltage fault</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Stored: the “voltage fault” alarm is signaled and the Drive goes in “Fault” state. You have to command a reset or restart the Drive if you want to restore its state</td></tr>
<tr><td>1</td><td>Auto reset: the “voltage fault” alarm is signaled and the Drive goes in “Fault” state. When the voltage is restored to its normal value, the Drive exits from the “Fault” state without the need of an external reset.</td></tr>
<tr><td rowspan="4">8</td><td colspan="2">Phase fault</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>off: alarm not enabled</td></tr>
<tr><td>1</td><td>on: alarm enabled</td></tr>
<tr><td rowspan="4">9</td><td colspan="2">Voltage fault</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>off: alarm not enabled</td></tr>
<tr><td>1</td><td>on: alarm enabled</td></tr>
<tr><td>10 ... 15</td><td colspan="2">Reserved</td></tr>
</table>

4.09.2 Register 0221h (545 dec) – Time alarm I2t

Address	Name	Type	Range	Default	Unit	Read	Write
0221h	**Time alarm I2t**	UINT16	0...60000	0	s/100	Always	Always

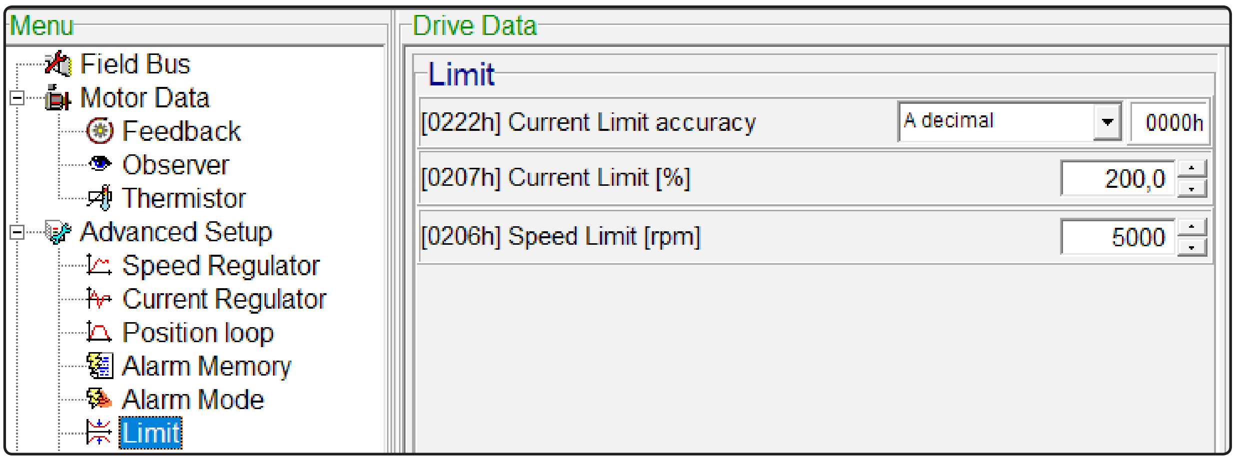
This register stores the reference time “***time alarm I2t***”, used to control both the Drive’s and the motor’s current. If the drive current rises over twice its rated value for a period longer than the one set on register 0221h, then the alarm 27 occurs. If the motor current rises over twice its rated value for a period longer than the one set on register 0221h, then the alarm 28 occurs.

4.10 Limit

This paragraph lists the parameters used to set the speed’s and current’s maximum limits used by the Drive.

Address	Name	Type	Read	Write
0206h	**Speed limit**	UINT16	Always	Always
0207h	**Current limit**	UINT16	Always	Always
0222h	**Various flags**	UINT16	Always	Switch off

These parameters can be set in the Caliper menu “***Advanced setup-Limit***” in which the following screen appears:



4.10.1 Register 0206h (518 dec) – Speed limit

Address	Name	Type	Range	Default	Unit	Read	Write
0206h	**Speed limit**	UINT16	0...20000	5000	rpm	Always	Always

This register contains the maximum limit, expressed in [rpm], of the speed reference's absolute value.

4.10.2 Register 0207h (519 dec) – Current limit

Address	Name	Type	Range	Default	Unit	Read	Write
0207h	**Current limit**	UINT16	0...3000	2000	‰	Always	Always

This register contains the maximum limit of the current reference's absolute value, expressed in [‰] with respect to the rated motor current. For example, if the rated motor current is of 2,00 [A] and the current's limit is set on 2000, then the maximum current allowed is equal to 4,00 [A].

4.10.3 Register 0222h (546 dec) – Various Flags

Address	Name	Type	Range	Default	Unit	Read	Write
0222h	**Various flags**	UINT16	0...65535	0		Always	Switch off

This register contains more flags, which are used in different part of the Firmware. In this menù it's used only the flag "***Current limit accuracy***", but for the sake of completeness the following table shows all the flags:

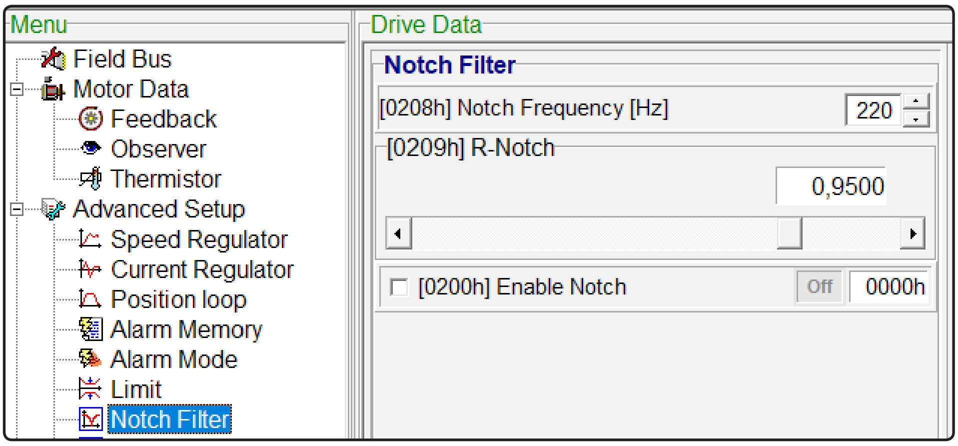
Bit	Description	
0,1	**Frequency mode**	
	Value	**Description**
	0	Channel A-B
	1	Frequency direction
	2	Pulses CW/CCW
2	**Mode end position**	
	Value	**Description**
	0	End current quote using the position control
	1	End current quote using the speed control (reference speed=0)
3	**Compatibility PID (only DgFox/TomCat)**	
	Value	**Description**
	0	***Off***: The PID gains are taken as they are
	1	***On***:To mantain the compatibility with the tuning of PIDs made in older applications the PID gains are converted inside the Firmware
4,5	**Select encoder out (only NTT)**	
	Value	**Description**
	0	Enc. rif. frequency
	1	Enc. sim. resolver
	2	Enc. increment+Hall
6	**set_tmc_out2_analog2 (only Tomcat)**	
	Value	**Description**
	0	the analog input 2 is used as a digital input (n°7)
	1	the analog input 2 is used as a digital input (n°7)
7	**Current limit accuracy**	
	Value	**Description**
	0	A decimal
	1	No decimal
8 ... 15	***Reserved***	

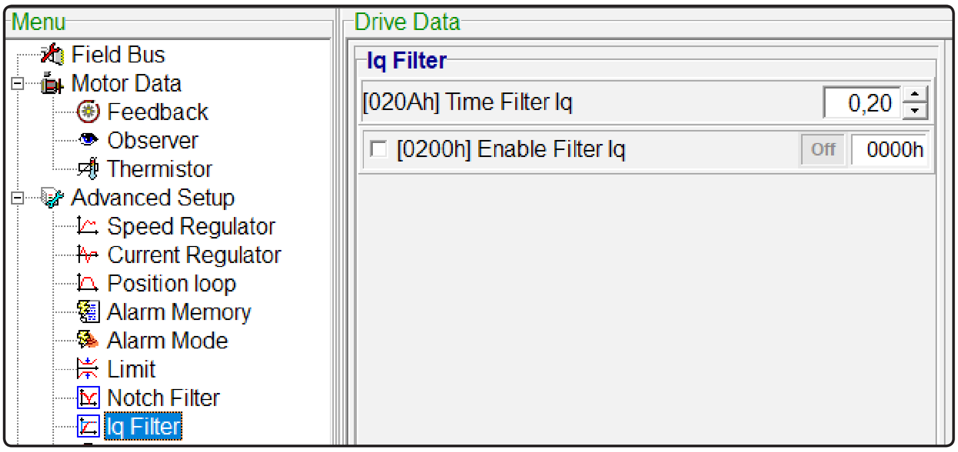
4.11 Filters

This paragraph lists the parameters used to tune the ***Notch filter*** and the ***Iq filter*** of the Drive. The Notch filter is used to eliminate a certain frequency from the Torque current (Iq) produced by the speed regulator. This Filter could be useful to avoid undesired resonances in the system composed by the motor and its load.On the other hand the Iq filter is used to filter the Iq reference used by the current regulator. This filter could be useful to avoid current's overshoot,but the current regulator will become less dynamic.

Address	Name	Type	Read	Write
0200h	**Filters Flags**	UINT16	Always	Always
0208h	**Notch frequency**	UINT16	Always	Always
0209h	**R-Notch (bandwidth)**	UINT16	Always	Always
020Ah	**Time filter Iq**	UINT16	Always	Always

These parameters can be set in the Caliper menus "***Advanced setup-Notch filter***" and "***Advanced setup-Iq filter***" in which the following screens appear:





4.11.1 Register 0200h (512 dec) Bit 0,1 – Filters flags

Address	Name	Type	Range	Default	Unit	Read	Write
0200h	**Filters flags**	UINT16	0...65536	0		Always	Always

This register is referred in several paragraphs of this manual, because it contains various flags used in different points of the firmware. The bits 0 and 1 of this register are two flags used to enable or disable the Notch filter or the iq filter, as showed in the following table:

<table>
<tr><th>Bit</th><th colspan="2">Description</th></tr>
<tr><td rowspan="4">0</td><td colspan="2">Enable/disable Notch filter</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Notch filter disabled</td></tr>
<tr><td>1</td><td>Notch filter enabled</td></tr>
<tr><td rowspan="4">1</td><td colspan="2">Enable/disable Low-pass Iq filter</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Low-pass Iq filter disabled</td></tr>
<tr><td>1</td><td>Low-pass Iq filter enabled</td></tr>
<tr><td>2 ... 15</td><td colspan="2">Reserved</td></tr>
</table>

4.11.2 Register 0208h (520 dec) – Notch frequency

Address	Name	Type	Range	Default	Unit	Read	Write
0208h	**Notch frequency**	UINT16	2...400	220	Hz	Always	Always

This register is used to set the frequency that the Notch filter should eliminate.

4.11.3 Register 0209h (521 dec) – R-Notch (bandwidth)

Address	Name	Type	Range	Default	Unit	Read	Write
0209h	**R-Notch (bandwidth)**	UINT16	8000...9900	9500		Always	Always

This register is used to set an adimensional parameter, R. R is used to define the bandwidth around the Notch frequency. All the frequencies inside this range will still be partially reduced by the Notch filter.

4.11.4 Register 020Ah (522 dec) – Time filter Iq

Address	Name	Type	Range	Default	Unit	Read	Write
020Ah	**Time filter Iq**	UINT16	1...3000	20	Ms/100	Always	Always

This parameter represents the time costant of the low-pass Iq filter and is expressed in hundredths of [ms] (the two least significant digits are interpreted as decimal fractions of [ms], for example the value 20 is intended as 0,20 [ms].

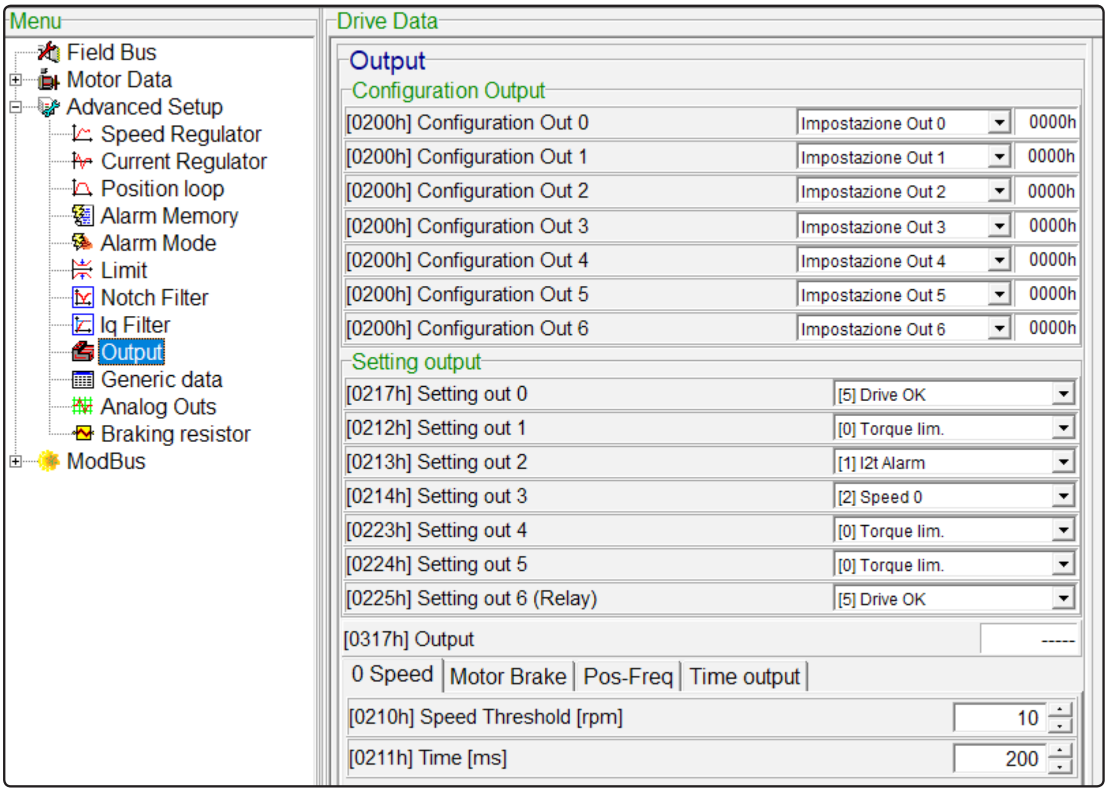
4.12 Output

This paragraph describes the registers used to set the Drive's outputs.

Address	Name	Type	Read	Write
0200h	**Configuration Outputs**	UINT16	Always	Always
020Bh	**Time brake enable**	UINT16	Always	Always
020Ch	**Time brake disable**	UINT16	Always	Always
020Dh	**Current offset**	INT16	Always	Always
020Eh	**Deceleration**	UINT16	Always	Always
020Fh	**Speed brake enable**	UINT16	Always	Always
0210h	**Speed threshold**	UINT16	Always	Always
0211h	**Time**	UINT16	Always	Always
0212h	**Setting Out1**	UINT16	Always	Always
0213h	**Setting Out2**	UINT16	Always	Always
0214h	**Setting Out3 (NTT only)**	UINT16	Always	Always
0217h	**Setting Out0**	UINT16	Always	Always

Address	Name	Type	Read	Write
0218h	**Divisor count encoder**	UINT16	Always	Always
0223h	**Setting Out4 (NTT only)**	UINT16	Always	Always
0224h	**Setting Out5 (NTT only)**	UINT16	Always	Always
0225h	**Setting Out6 (relay) (NTT only)**	UINT16	Always	Always
0317h	**Output**	UINT16	Always	Always
0456h	**Time out0**	UINT16	Always	Always
0457h	**Time out1**	UINT16	Always	Always
0458h	**Time out2**	UINT16	Always	Always
0459h	**Time out3**	UINT16	Always	Always
045Ah	**Time out4**	UINT16	Always	Always
045Bh	**Time out5**	UINT16	Always	Always
045Ch	**Time out6**	UINT16	Always	Always

These parameters can be set in the Caliper menu "***Advanced setup-Output***" in which the following screen appears:



4.12.1 Register 0200h (512 dec) bit 4,5,6,7,8,9,10 – Configuration outputs

Address	Name	Type	Range	Default	Unit	Read	Write
0200h	**configuration outputs**	UINT16	0...65536	0		Always	Always

This register is referred in several paragraphs of this manual, because it contains various flags used in different points of the firmware. The bits 4,5,6,7,8,9 and 10 of this register are used to set how the Drive's outputs will be controlled:

- When the bit assigned to an output is set to 0, the output in question will report the value chosen via Caliper, as described on paragraph *"4.12.12 Register 0217h (535 dec) – Setting Out0" pag. 162*
- When the bit assigned to an output is set to 1, the output in question will be controlled via Bus. This function allows you to use the Drive as an extension module for the outputs of a PLC to which it is connected

Bit	Name	Range	Default
4	Configuration Out0	0 = Setting Out 0 1 = Bus control	Setting Out 0
5	Configuration Out1	0 = Setting Out 1 1 = Bus control	Setting Out 1
6	Configuration Out2	0 = Setting Out 2 1 = Bus control	Setting Out 2
7	Configuration Out3	0 = Setting Out 3 1 = Bus control	Setting Out 3
8	Configuration Out4	0 = Setting Out 4 1 = Bus control	Setting Out 4
9	Configuration Out5	0 = Setting Out 5 1 = Bus control	Setting Out 5
10	Configuration Out6	0 = Setting Out6 1 = Bus control	Setting Out 6

4.12.2 Register 020Bh (523 dec) – Time brake enable

Address	Name	Type	Range	Default	Unit	Read	Write
020Bh	**Time brake enable**	UINT16	10..2000	200	msec	Always	Always

This register is used to set the time delay between the brake de-energizing command and the actual mechanical stop.

4.12.3 Register 020Ch (524 dec) – Time brake disable

Address	Name	Type	Range	Default	Unit	Read	Write
020Ch	**Time brake disable**	UINT16	10..2000	200	msec	Always	Always

This register is used to set the time delay between the brake energizing command and its actual releasing.

4.12.4 Register 020Dh (525 dec) – Current offset

Address	Name	Type	Range	Default	Unit	Read	Write
020Dh	**Current offset**	INT16	-32000 ...32000	0	A/100	Always	Always

In the vertical axis applications the gravity excert a continuous torque on the motor. As a result of this torque, when the brake is released the motor doesn't remain in place, but it makes a little movement before the PID starts working and bring it back to its initial position. To avoid this problem you can set a "***current offset***", which produces an ever-present feedforward torque which cancels the torque caused by the gravity. The value of this current has to be set manually. To find this value you have to follow these steps:

» Select "***Speed***" as control type, and give the ***Power on*** command without enabling the reference. This is equivalent to set the speed reference equal to 0;
» release the brake;
» Wait until the motor ends its transitory movement and stops moving;
» Read from Caliper the value of the current used by the Drive to blocks the motor;
» Set the current's value read in the previous step as the "***current offset***".

4.12.5 Register 020Eh (526 dec) – Deceleration

Address	Name	Type	Range	Default	Unit	Read	Write
020Eh	**Deceleration**	UINT16	1..60000	1000	Rpm/s	Always	Always

The Drive, before disabling the motor torque and closing the brake, checks whether the motor speed is lower than the threshold value set on the parameter "***Speed brake enable***" (register 020Fh). If the motor speed is higher than this threshold, the Drive starts a deceleration ramp with the deceleration set on this register. Once the motor has reached the speed set in "***Speed brake enable***" the Drive closes the brake and removes the torque. However, If the Drive goes in the ***Fault*** state, the brake is closed immediately without checking the speed.

4.12.6 Register 020Fh (527 dec) – Speed brake enable

Address	Name	Type	Range	Default	Unit	Read	Write
020Fh	**Speed brake enable**	UINT16	1..500	4	Rpm	Always	Always

This register contains the minimum speed threshold below which the motor brake is closed during the stopping phase.

4.12.7 Register 0210h (528 dec) – Speed threshold

Address	Name	Type	Range	Default	Unit	Read	Write
0210h	**Speed threshold**	UINT16	1..1500	10	rpm	Always	Always

This register is used to set a speed threshold. If the absolute value of the measured speed remains under the value set on this register for a time longer than the one set on the parameter ***Time*** (Register 0211h), than the output assigned to the function ***speed 0*** rises to the logical value 1, i.e. it's signaled that the motor has reached the "zero speed". Vice versa, in the opposite case (***measured speed*** > ***Speed threshold***),the same output shows the logical value 0.

4.12.8 Register 0211h (529 dec) – Time

Address	Name	Type	Range	Default	Unit	Read	Write
0211h	**Time**	UINT16	1..10000	100	msec	Always	Always

This register is used to set a time period, named "***time***" on Caliper. If the absolute value of the measured speed remains under the value set on the register "***speed threshold"*** (Register 0210h) for a time longer than the one set on this register, than the output assigned to the function ***"speed 0"*** rises to the logical value 1, i.e. it's signaled that the motor has reached the "zero speed".

4.12.9 Register 0212h (530 dec) – Setting Out1

Address	Name	Type	Range	Default	Unit	Read	Write
0212h	**Setting Out1**	UINT16	0...9	0		Always	Always

This register contains the parameter "***Setting Out 1"***. Depending on the value of this parameter a certain function is assigned to the output 1. The possible assignable functions are the same for all the outputs;they're described in paragraph *"4.12.12 Register 0217h (535 dec) – Setting Out0" pag. 162*

4.12.10 Register 0213h (531 dec) – Setting Out2

Address	Name	Type	Range	Default	Unit	Read	Write
0213h	**Setting Out2**	UINT16	0...9	1		Always	Always

This register contains the parameter "***Setting Out 2"***. Depending on the value of this parameter a certain function is assigned to the output 2. The possible assignable functions are the same for all the outputs;they're described in paragraph *"4.12.12 Register 0217h (535 dec) – Setting Out0" pag. 162*

4.12.11 Register 0214h (532 dec) – Setting Out3 ((NTT ONLY)

Address	Name	Type	Range	Default	Unit	Read	Write
0214h	**Setting Out3**	UINT16	0...9	2		Always	Always

This register contains the parameter "***Setting Out 3"***. Depending on the value of this parameter a certain function is assigned to the output 3. The possible assignable functions are the same for all the outputs;they're described in paragraph *"4.12.12 Register 0217h (535 dec) – Setting Out0" pag. 162*. This register is not used by the DgFox and TomCat Drives.

4.12.12 Register 0217h (535 dec) – Setting Out0

Address	Name	Type	Range	Default	Unit	Read	Write
0217h	**Setting Out0**	UINT16	0...9	5		Always	Always

This register contains the parameter "***Setting Out 0"***. Depending on the value of this parameter a certain function is assigned to the output 0. The description of the possible assignable functions are reported in the following table:

Value	Name	Description
0	***Torque limit***	The output turns on when the motor reaches the torque limit.
1	***I2t alarm***	The output turns on when there is an overcurrent
2	***Speed 0***	The output turns on when the motor has reached the "zero speed". This event is defined using the parameters In questo caso viene segnalato il raggiungimento della velocità nulla in base ai parametri "***Speed threshold***"(register 0210h) and "***Time***"(register 0211h)
3	***Target reached***	This function is used when the Drive works with a ***Positioner*** control. The output turns on when the motor has reached the target position.
4	***Motor brake***	The output turns on when the motor brakes is activated, which depends on the parameters ***Time brake enable*** (Register 020Bh), ***Time brake disable*** (Register 020Ch), ***Deceleration*** (Register 020Eh), ***Speed brake enable*** (Register 020Fh)
5	***Drive OK***	The output turns on when there is no alarms. The Drive can work
6	***Secure Disable***	The output turns on when the Drive is in the "***secure disable***" state
7	***Pos-freq out*** ***(only for DGFox and TomCat)***	The output produces a square waveform with a frequency which depends on a divider, which is a power of 2, and the position (in increments)of the motor on a single turn.
8	***Pos output***	This function is used when the Drive works with a "***Positioner Input-start***" control, the output produces a pulse of the set duration when the motor reach the current target position.
9	***Homing attained***	When any of the "***Homing***" procedure is performed, the output turns on and remains at 1 untill another "***Homing***" procedure is requested (or untill the Drive is turn off).

4.12.13 Register 0218h (536 dec) – Divisor count encoder

Address	Name	Type	Range	Default	Unit	Read	Write
0218h	**Divisor count encoder**	UINT16	0...9	1		Always	Always

This register contains the parameter "***divisor count encoder***", which is used by the function "***Pos-freq. out***" (see paragraph *"4.12.12 Register 0217h (535 dec) – Setting Out0" pag. 162*).

"***Pos-Freq. Out***" produces a square waveform with a frequency which depends on a divider, which is a power of 2, and the position (in increments)of the motor on a single turn. To set the resolution of this output signal expressed in pulses per turn the Drive follows the following formula:

$$Enc_{out} = \frac{FB_{RESOLUTION}}{2\ DIV}$$

where:

» *Enc_out* is the resolution of the output square waveform;
» *FB_resolution* is the resolution of the motor feedback;
» *DIV* is the value of "***divisor count encoder***"

4.12.14 Register 0223h (547 dec) – Setting Out4 (NTT ONLY)

Address	Name	Type	Range	Default	Unit	Read	Write
0223h	**Setting Out4**	UINT16	0...9	2		Always	Always

This register contains the parameter "***Setting Out 4***". Depending on the value of this parameter a certain function is assigned to the output 4. The possible assignable functions are the same for all the outputs;they're described in paragraph *"4.12.12 Register 0217h (535 dec) – Setting Out0" pag. 162.* This register is not used by the DgFox and TomCat Drives.

4.12.15 Register 0224h (548 dec) – Setting Out5 (NTT ONLY)

Address	Name	Type	Range	Default	Unit	Read	Write
0224h	**Setting Out5**	UINT16	0...9	2		Always	Always

This register contains the parameter "***Setting Out 5"***. Depending on the value of this parameter a certain function is assigned to the output 5. The possible assignable functions are the same for all the outputs;they're described in paragraph *"4.12.12 Register 0217h (535 dec) – Setting Out0" pag. 162.* This register is not used by the DgFox and TomCat Drives.

4.12.16 Register 0225h (549 dec) – Setting Out6 (relay) (NTT ONLY)

Address	Name	Type	Range	Default	Unit	Read	Write
0225h	**Setting Out6**	UINT16	0...9	2		Always	Always

This register contains the parameter "***Setting Out 6"***. Depending on the value of this parameter a certain function is assigned to the output 6. The possible assignable functions are the same for all the outputs;they're described in paragraph *"4.12.12 Register 0217h (535 dec) – Setting Out0" pag. 162.* This register is not used by the DgFox and TomCat Drives.

4.12.17 Register 0317h (791 dec) – Output

Address	Name	Type	Range	Default	Unit	Read	Write
0317h	**Output**	INT16	0 ...65535			Always	Always

This register works as a buffer between the Field Bus and the 7 digital outputs of the Drive.

If one or more outputs are set on "***Bus control*** " (see paragraph *"4.12.1 Register 0200h (512 dec) bit 4,5,6,7,8,9,10 – Configuration outputs" pag. 160)*, the register 0317h will receives from the Field Bus the values that these outputs have to show, and then it updates their values.

4.12.18 Registers 0456h (1110 dec) … 045Ch (1116 dec) – Time outputs

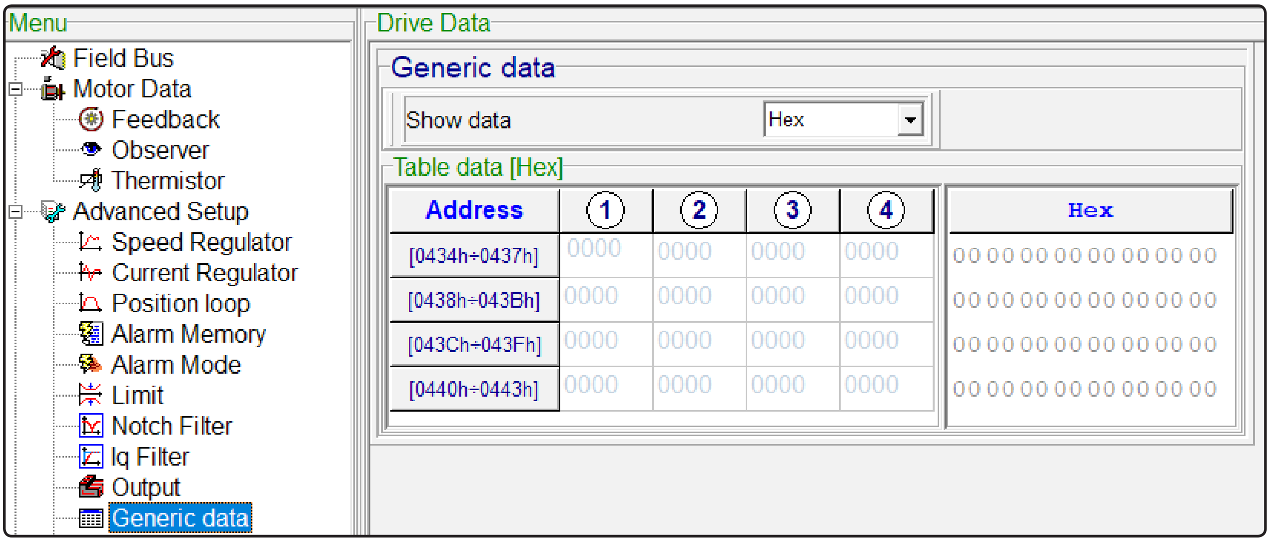
Address	Name	Type	Range	Default	Unit	Read	Write
0456h	**Time out0**	UINT16	1 ... 6000	1	S/100	Always	Always
0457h	**Time out1**						
0458h	**Time out2**						
0459h	**Time out3**						
045Ah	**Time out4**						
045Bh	**Time out5**						
045Ch	**Time out6**						

These registers contain the time intervals used by the function "***Pos output***" (see paragraph *"4.12.12 Register 0217h (535 dec) – Setting Out0" pag. 162).* "***Pos output***" makes the assigned outputs produce a pulse when the current target position has been reached; the width of this pulse is set on these registers.

4.13 Generic data

Clicking on the Caliper menu "***Advanced setup- Generic Data***" you can see the registers from 0434h to 0443h.

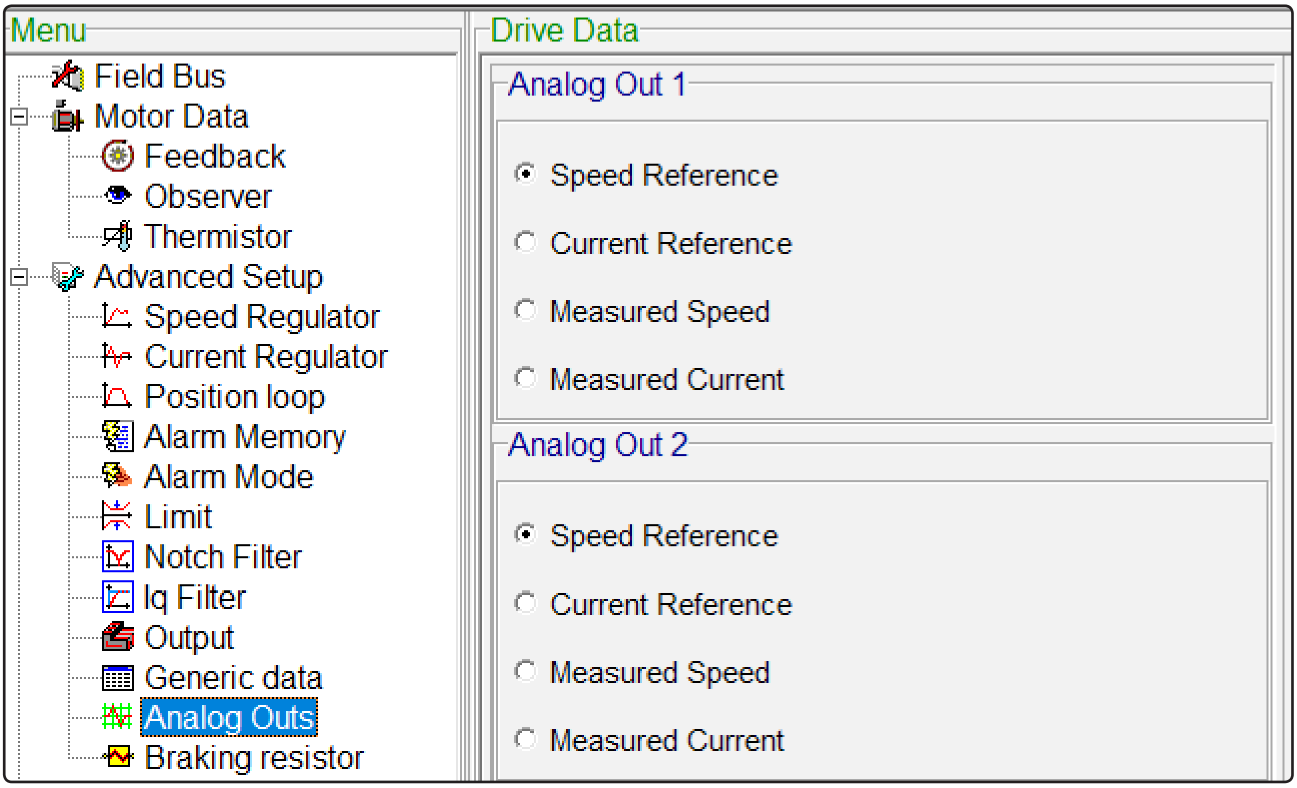
Address	Name	Type	Read	Write
0434h	**Generic data**	UINT16	Always	Always
0435h	**Generic data**	UINT16	Always	Always
0436h	**Generic data**	UINT16	Always	Always
0437h	**Generic data**	UINT16	Always	Always
0438h	**Generic data**	UINT16	Always	Always
0439h	**Generic data**	UINT16	Always	Always
043Ah	**Generic data**	UINT16	Always	Always
043Bh	**Generic data**	UINT16	Always	Always
043Ch	**Generic data**	UINT16	Always	Always
043Dh	**Generic data**	UINT16	Always	Always
043Eh	**Generic data**	UINT16	Always	Always
043Fh	**Generic data**	UINT16	Always	Always
0440h	**Generic data**	UINT16	Always	Always
0441h	**Generic data**	UINT16	Always	Always
0442h	**Generic data**	UINT16	Always	Always
0443h	**Generic data**	UINT16	Always	Always



These are addresses of the flash memory, non volatile, which are made avaiable to the customer and which can only be accessed via Field Bus. The data stored on these registers cannot be modified by the Drive nor be cancelled if the firmware is updated. Therefore they can be used to identify the Drive, setting on them a unique value which can be used as an "electronic label".

4.14 Analog outs

In this paragraph is described the menu "***Advanced setup -Analog outs***" in which the following screen appears:



This menù is used to select the variables that we want to send via the Drive's analog output. The resolution of the output analog signals are of 10 bit, therefore it's advised to use them only for a "monitoring" purpose.

4.14.1 Register 0200h (533 dec) Bit 12,13,14,15 – Analog outputs

Address	Name	Type	Range	Default	Unit	Read	Write
0200h	**Analog outputs**	UINT16	0...65536	0		Always	Always

The NTT drives can produce 2 analog signals proportional to one of the variables listed in the table below, witha resolution of 10 Bit. The last 4 Bits of register 0200h are used to select the variables that we want to observe:

Analog output 1		
Bit register 200h		**Observable variables**
12	**13**	
0	0	Speed reference
0	1	Current reference
1	0	Measured speed
1	1	Measured current

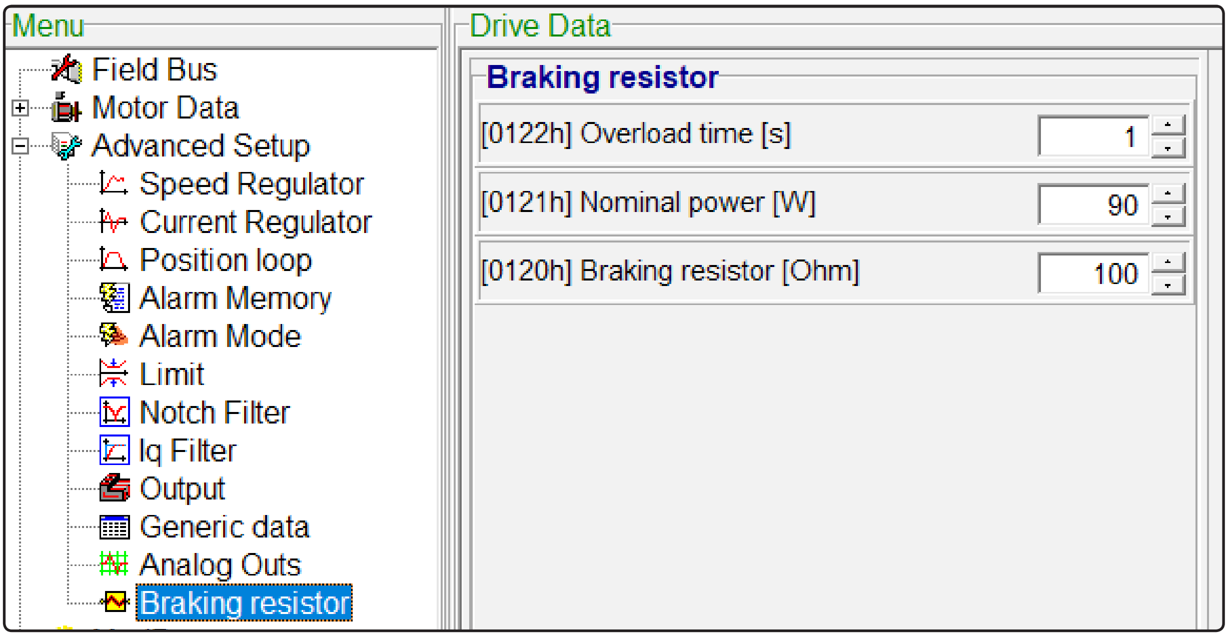
Analog output 2		
Bit register 200h		Observable variables
14	15	
0	0	Speed reference
0	1	Current reference
1	0	Measured speed
1	1	Measured current

4.15 Braking resistor

This paragraph describes the registers used to set the characteristics of the braking resistor eventually connected to the Drive.

Address	Name	Type	Read	Write
0120h	**Braking resistor**	UINT16	Always	Always
0121h	**Nominal power**	UINT16	Always	Always
0122h	**Overload time**	UINT16	Always	Always

These parameters can be set in the Caliper menu "***Advanced setup-Braking resistor***" in which the following screen appears:



4.15.1 Register 0120h (288dec) – Braking resistor

Address	Name	Type	Range	Default	Unit	Read	Write
0120h	**Braking resistor**	UINT16	10..10000	100	ohm	Always	Always

This register is used to set the resistance, expressed in [Ohm], of the Braking resistor connected to the Drive.

4.15.2 Register 0121h (289dec) – Nominal power

Address	Name	Type	Range	Default	Unit	Read	Write
0121h	**Nominal power**	UINT16	30..10000	90	Watt	Always	Always

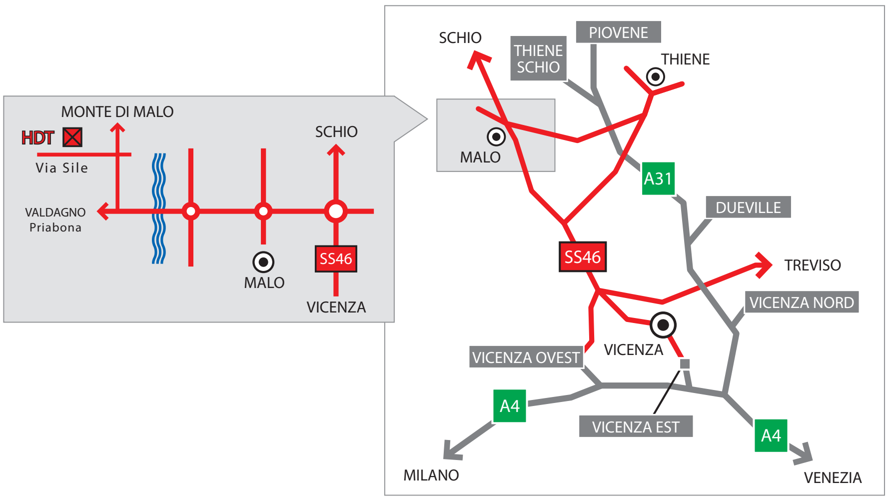
This register is used to set the rated power, expressed in [Watt], of the braking resistor connected to the Drive.

4.15.3 Register 0122h (290dec) – Overload time

Address	Name	Type	Range	Default	Unit	Read	Write
0122h	**Overload time**	UINT16	1..255	1	s	Always	Always

This register is used to set the overload time allowed by the braking resistor connected to the Drive.

High Digital Technology



H.D.T. s.r.l.

www.hdtlovato.com

via Sile 8, 36030 Monte di Malo (VI)
Tel. +39. (0) 445.602744 r.a. – Fax +39. (0) 445.602668