

## Sommario

### ***AN002 - Example of use of Danaher drives with MicroQMove connected to CANOpen***

<b><i>bus</i></b> .....	3
Bus expansion .....	3
Short description of CANOpen .....	3
Creation and configuration of a network .....	4
Parameterization and drive configuration .....	6
Drive calibration .....	10
DEMO_CANOPEN_DRIVE application .....	11
Mode of operation .....	12
CAN network diagnostic .....	12



## AN002 - Example of use of Danaher drives with MicroQMove connected to CANOpen bus

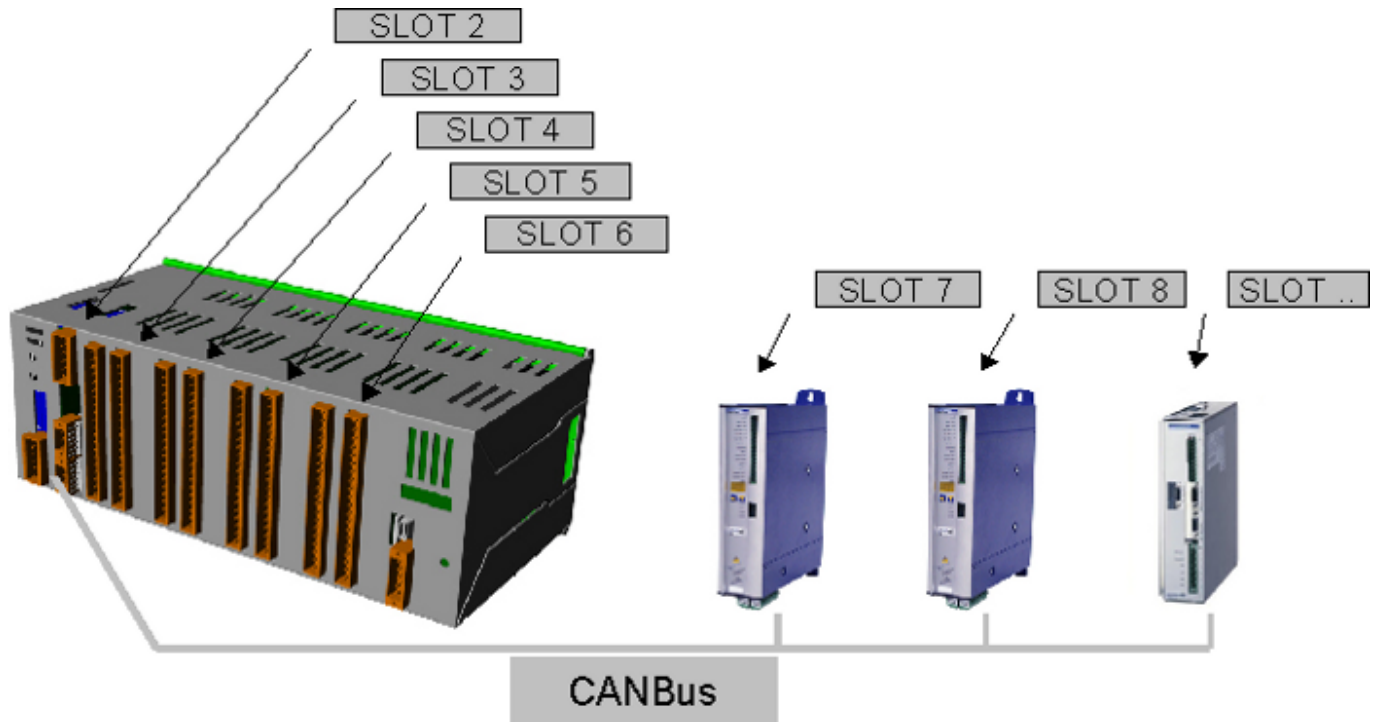
In this “application note” explains how to control the Danaher drives connected with a MicroQMove device using the CANOpen bus. In this way the user can figure out how to navigate at the time of construction of a communication network. The MicroQMove device in this example is an R502BF.

### Bus expansion

The 502BF product is a programmable system consists of a main processing unit (CPU) and a series of cards that manage local inputs and outputs internally connected via a data bus. On this internal bus identifies a number (depends to the model) of card slots and numbered starting to 2. For example the 502BF product allows you to install up to 5 cards, then the slot numbering on the bus will be from 2 to 6.

When you create a CANOpen network, the number of slots is extended up to 12 thus making available additional I/O present on devices that implement the DS-401 profiles of communication “Communication profile for generic I/O modules” and DS-402 “Device Profile Drives and Motion Control”. The CANOpen network only supports a configuration with devices that belong to the same profile, so it is not possible to have generic I/O modules with the same network drives. Also when you use a network configured for devices by DS-402 type (driver), These must all be of the same manufacturer. In this document, we'll discuss connecting and setting up a network in which all devices are drives Danaher.

The product is equipped with a CANbus port that implements a CANOpen standard communication protocol. This Protocol is used to implement a process communication We can define simplistically a “internal bus expansion”. This means that for all intents are used process data communication to present to the system I/O information coming from the CANOpen bus and this information is addressed as if they belonged to additional card slots present “virtually” on the position from 7 to 12. This solution makes it easy and intuitive CANOpen network configuration.



### Short description of CANOpen

The CANOpen is a communication protocol based on CAN serial bus. The CANOpen expects the hardware of a device connected has a CAN receiver (CAN-transceiver) and a CAN controller as established by the ISO 11898. The CANOpen communication profile, CiA DS-301, implements both cyclic communications that event-oriented communication, thus reducing the load on the bus while maintaining low reaction times. You can achieve high performance in terms of the amount of Exchange data even at relatively low speeds, reducing problems related to EMC rules and wiring costs. The physical medium used by the CANOpen Protocol is a line bus differential pair with common return complies with ISO 11898. The bus length is limited by the speed of communication as follows:

Bitrate	Maximum bus length
1 Mbps	25 m
500 kbps	100 m
250 kbps	250 m
125 kbps	1000 m

## Creation and configuration of a network

Here are the steps to the creation and configuration of a network:

- connect the drives to the QEM instrument. For details about the choice of the connecting cable, the structure of the network, etc. refer to the specifications available from the site Can-Cia
- design the network type
- insert the terminating resistors on the first and last node in the network. To do this on the drive connector Danaher connect a resistor to 120 Ohm between the pins 9 and 6, while for the instrument QEM, the resistance is already internally provided and then turn it on (consult QEM product installation manuals for the operations)
- with Qview development environment to create a new project by defining a device of CANOPEN type that execute all configuration operations, diagnostics and network management.

```

-----
; Internal device declaration
-----
INTDEVICE
FieldB1 CANOPEN 0004 500 0

```

In the above example was declared a device with sample time 4ms and 500kbps communication rate

- insert into configuration unit, in the Declaration section of the BUS, a card type C402A on one of the available slots for the CANOpen network. The following example shows the configuration for a 502BF product connected with a driver:

```

-----
; Qmove Bus declaration
-----
BUS
1 502BF 99
2 . .
3 . .
4 . .
5 . .
6 . .
7 C402A .

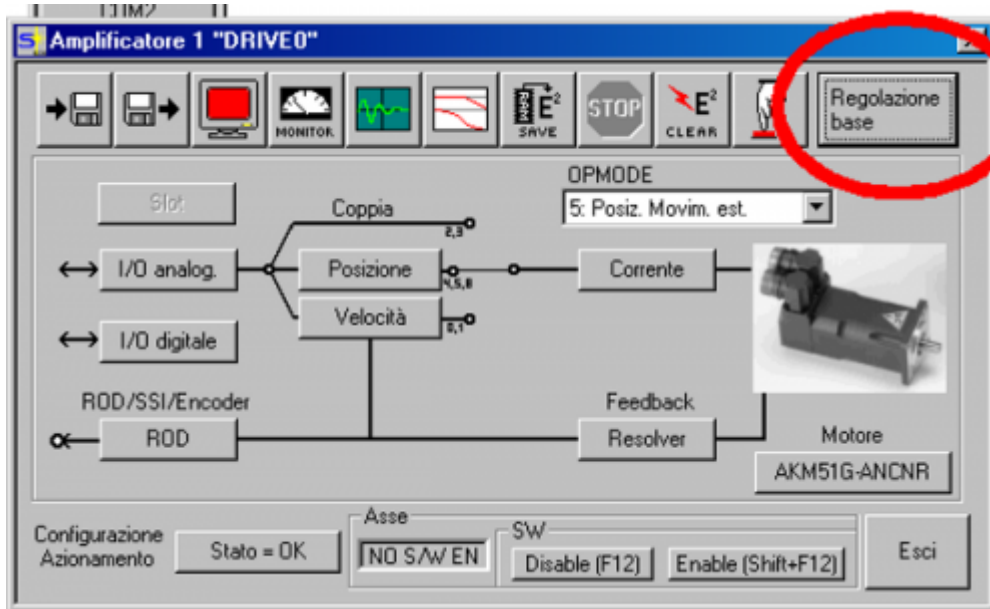
```

- add a Qcl unit without any additional code
- set the rate of communication. To set the rate of the 502BF product You must define the value in kbps in the definition of the CANOPEN device. Accepted values are 1000, 500, 250 and 125.

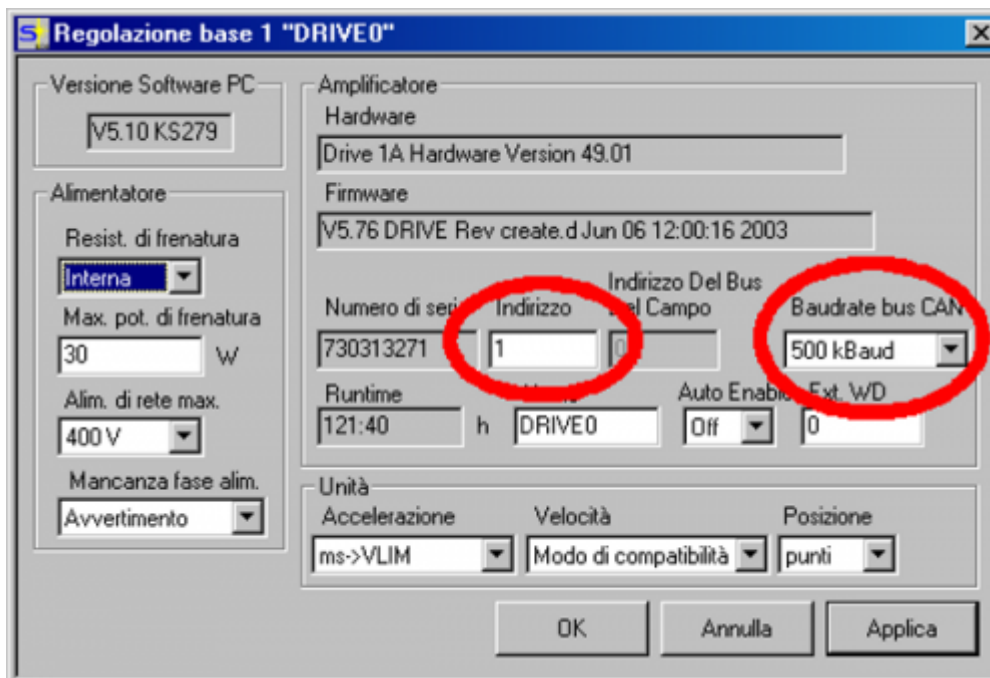
In the example we have chosen the speed 500 kbps. There is also another way of setting the rate to be used when you think would be necessary to change the speed even without intervention in the Qcl application. In this case you set zero in the speed during the declaration of the device and then choose the speed through the dip-switches:

Dip 5	Dip 7	Bitrate
off	off	125 kbps
off	on	250 kbps
on	off	500 kbps
on	on	1 Mbps

\* use the software provided by the manufacturer of the drive for setting parameters. Open the Basic setting by clicking on the button:



- set the "indirizzo" field at the 1 value and the CAN baudrate field bus to the 500 kBaud value:



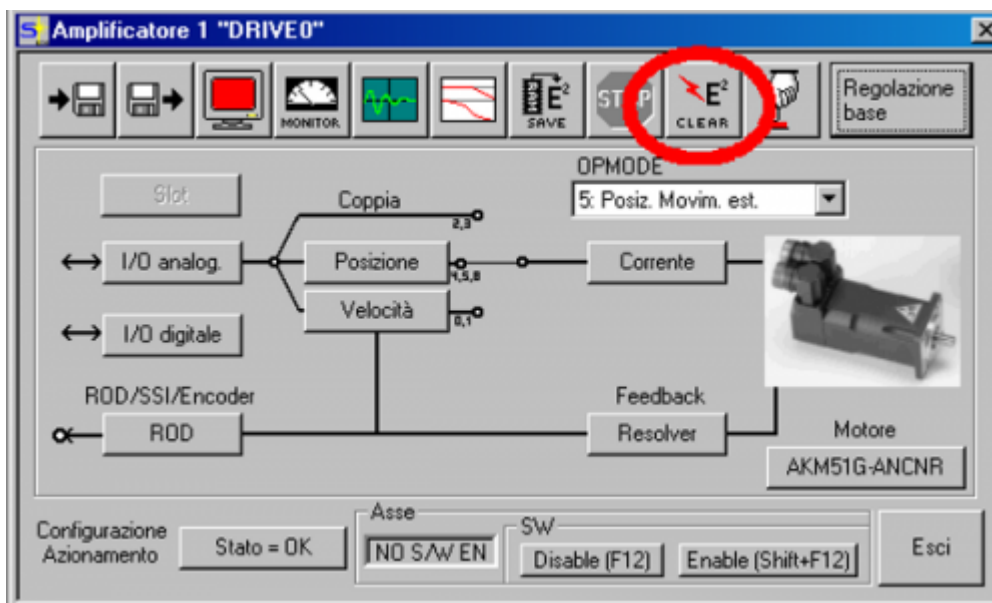
- from the QView, compile and download the application you previously created. Verify that there is not present a *Bus Error* or a *Watchdog Error*. Finally, by opening the bus monitor, verify that the driver is recognized. The BUS information window should look like this:

Detected Target BUS configuration			
SLOT	Card	Version	Watchdog BUS
1	502BF	99.0.2	
2			
3			
4			
5			
6			
7	C402A		
8			
9			
10			
11			
12			

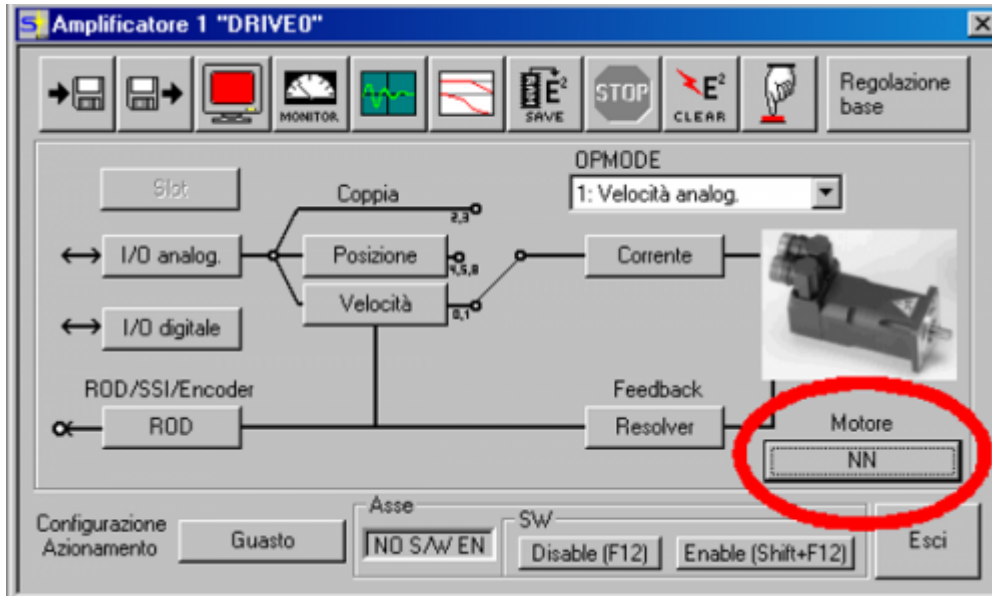
## Parameterization and drive configuration

For correctly manage from the QEM device, the Danaher driver must be properly configured. Here are the steps to do this, This refers to the software that comes with the drive:

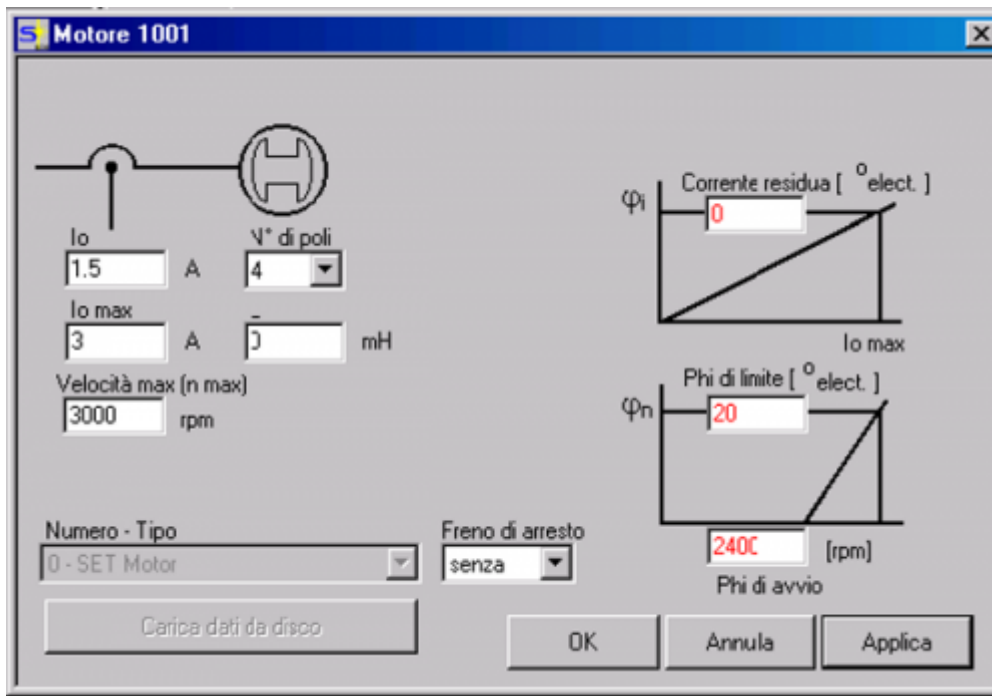
- restore the default values of the drive to start the parameterization from a known situation and repeatable. To do this press the button shown in the picture:



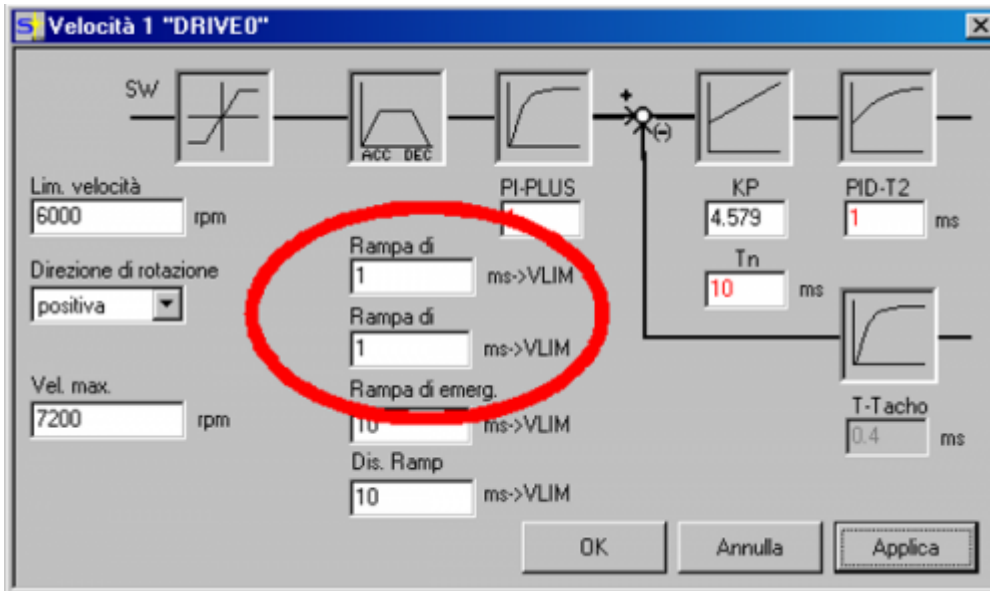
- set the motor data. Press the Motor button:



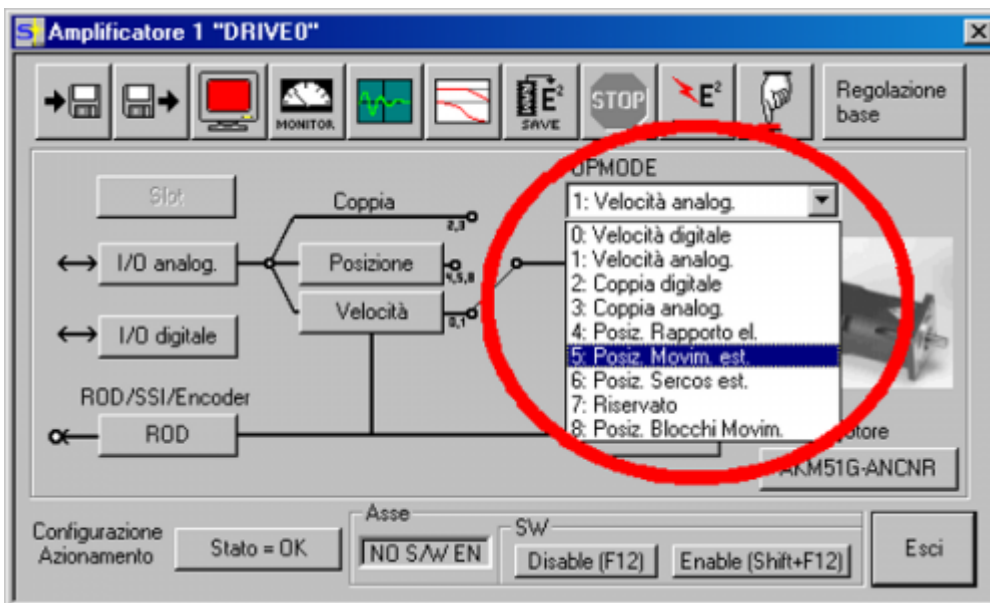
- manually set the parameters, or load the values using an external file supplied by the manufacturer:



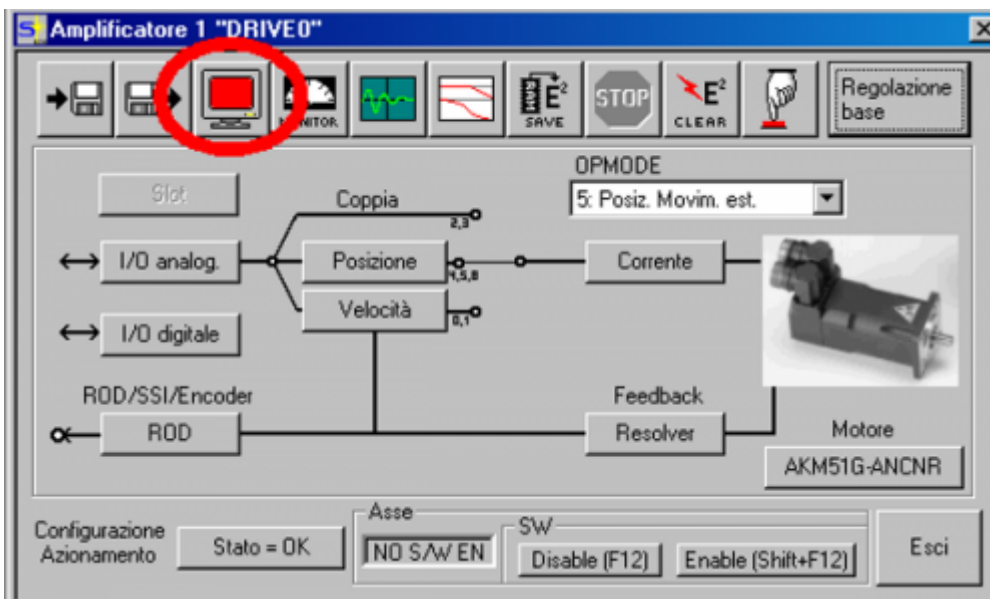
- press the speed button and set the acceleration and deceleration ramps at 1 ms:



- in the main screen AMPLIFIER, set OPMODE = 5) Placing External Movement:

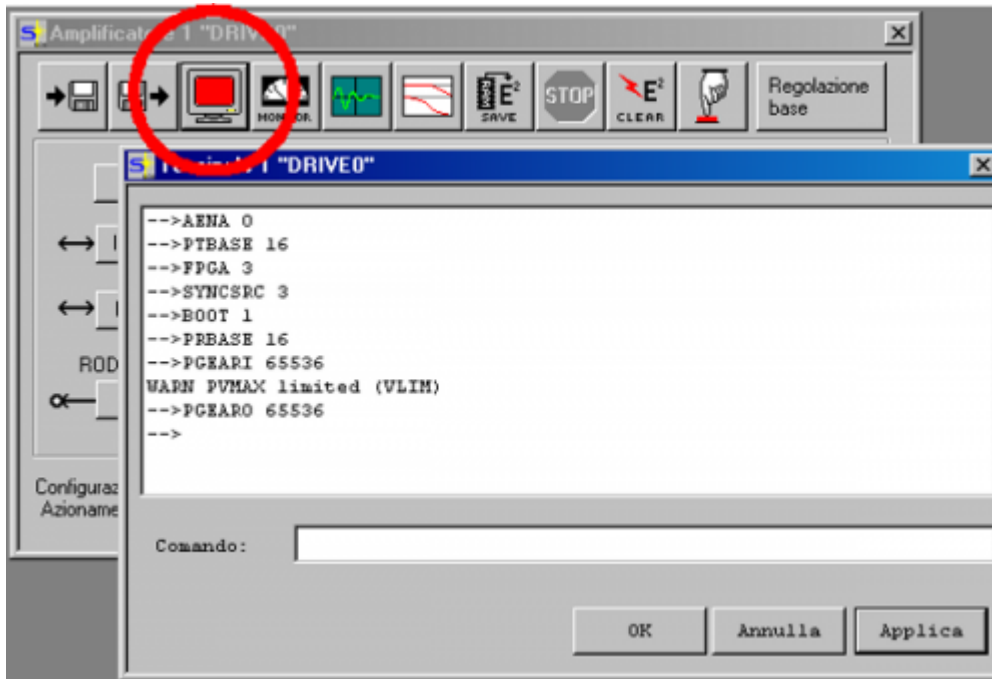


- Open the TERMINAL screen by pressing the button:



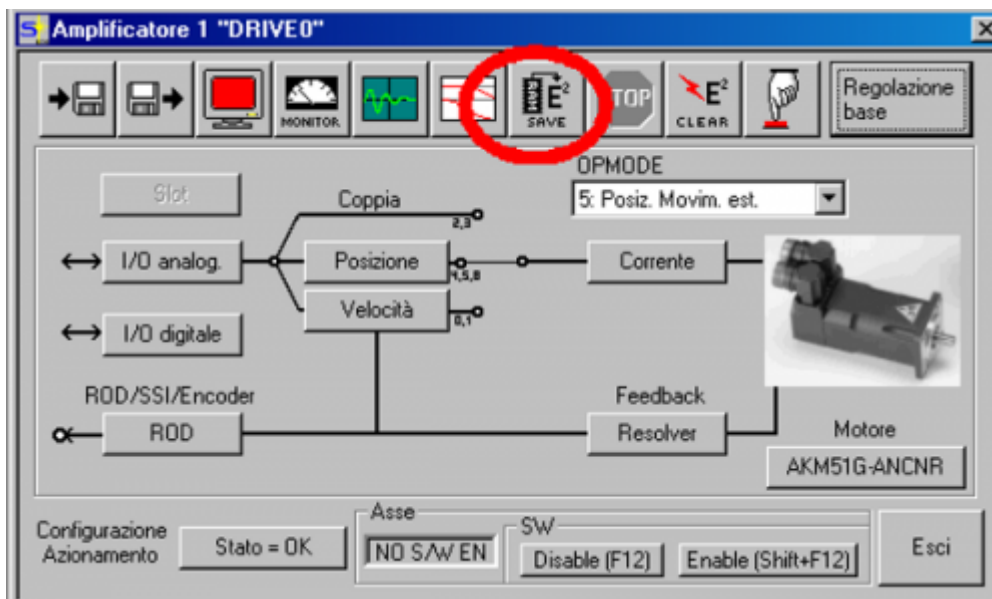


- Enter the following commands:



1. AENA 0: defines the State of the art software enabling drive ignition, setting it to 0 will be provided by 502BF and not automatically enabled
2. PTBASE 16: defines the interval between two SYNC message for for the implementation of the set-point position and sending the position of axles and is calculated using the follow formula:  $PTBASE = 4 * (t_{camp} \text{ CANOPEN device})$
3. FPGA 3: enables handling of synchronizing multiple drives via CANOpen
4. SYNCSRC 3: abilita la sincronizzazione di più azionamenti via CANOpen
5. BOOT 1: enable quick Boot mode on power up. The drive will take about 2 seconds to get ready instead of 15 seconds
6. PRBASE 16: imposta la risoluzione di posizione interna a  $2^{16}$  per essere compatibile con il prodotto 502BF
7. PGEARI 65536: numerator of position resolution
8. PGEARO 65536: denominator of position resolution

- Save the settings on the drive:



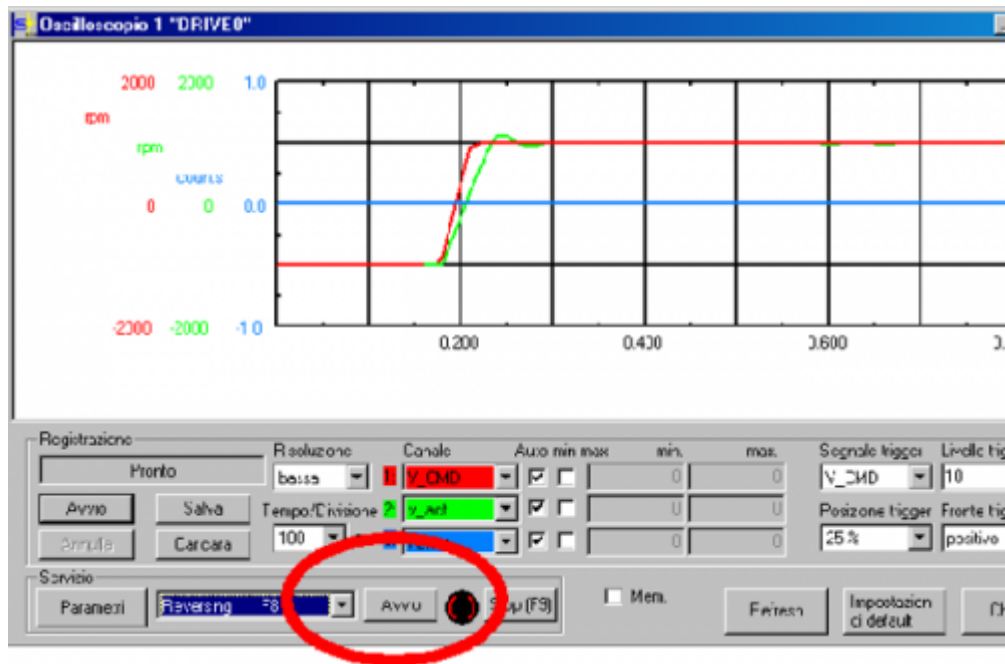
## Drive calibration

After setting the parameters on driving we can execute the axis calibration, which you should perform in two stages:

- calibration of the speed loop (non-communicating with the drive in CANOpen to 502BF)
- calibration of the position loop executing movements with the 502BF

### Speed loop calibration

To calibrate the speed loop you should: set on the screen OPMODE 0 = Digital Speed. Open the SPEED screen and set the acceleration and deceleration ramps at a value of approximately 100ms (depending on the load connected to the motor). To execute the motor movements to execute using the OSCILLOSCOPE screen functions Speed F6 or F8 by clicking on the Start button in the Service panel.



Now click the Start button Oscilloscope showing the positioning of the motor. Act on the parameters of the SPEED screen:

- $K_p$ : speed proportional gain
- $T_n$ : integration time speed proportional gain

Increase the gain  $K_p$  until the motor starts to vibrate, then reduce it to a value that is stable (too low a value representing a slow response of the motor, while too high value makes the motor too brusque and swinging). The  $T_n$  parameter, in milliseconds, represents the integral time of speed proportional gain: This value increases with the increase of the size of the motor and with increasing the load inertia. Too low value of  $T_n$  with large inertia involve sudden movements of the motor and strong overshoot, while too high values result in a slow response of the engine. If changing  $K_p$  and  $T_n$  you don't get a good calibration you can use the  $PID-T2$  parameter: a low-pass filter on the proportional gain of speed: too low or too high makes the answer too rapid or too slow axis respectively.

### Calibration of the position loop

At the end of the calibration of the speed loop switches to position ring calibration, by setting the acceleration and deceleration Ramp parameters to 1ms and OPMODE at the value 5)Placing external movement to work mode CanOpen with 502BF. Open the Qview application named DEMO\_CANOPEN\_DRIVE. This application monitors a drive with address nr.1 with rate 500 kbps. Upload the application to the device and give RUN to the CPU. By setting the  $gbtest$  variable = 3, the drive starts making a series of movements in two dimensions. To change the acceleration times refer to an01 (EANPOS) device while locating dimensions refer to code in the MOVEMENT unit.



**Note:**  
If the direction of movement is incorrect actuate the ASCII DIR control of the drive. Do not actuate the CNTREV control of the device that must remain in the default state ( $st\_cntrev = 0$ ).

Open the position screen and go to modify the parameters of the position ring  $K_v$  (position proportional gain) and  $F_f$  (Feed Forward factor) in order to test the behavior of the axis.

The Feed Forward factor determines the percentage by which the speed factor is considered in the calculation of the position feedback.

Even in this case you can check the error values with Oscilloscope tool.

These designations for calibration must be understood only as a trace. Refer to the official documentation of the drive for more precise informations.

## DEMO\_CANOPEN\_DRIVE application

This sample application implements all the necessary code to manage a drive controlled via CANOpen. It can be used as sample code or as a starting point to develop your own application.

The application consists of two units: CANOPEN and MOVEMENT.

MOVEMENT is a simple unit that implements a sequence of movements. The gbtest variable can implement the following sequences depending on the value:

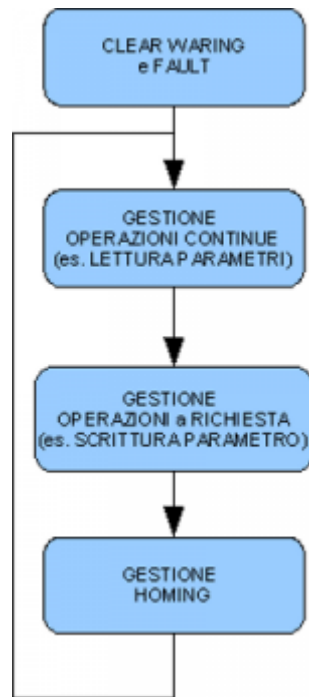
0 = No operation

1 = continuous motion in two dimensions (0 and 50000). Towards zero normal movement, to 50000 quote torque limiting movement and with error tracking disabled

2 = start homing procedure

3 = continuous motion in two dimensions (0 and 10000)

CANOPEN is destined for network management CANOpen. It is the only unit that makes use of the device; it executes the following macro operations:



The managing continuing operations is implemented in Qcl procedure named DR\_GET\_VALUES, is used to read some values from the drive through the SDO. These values are put on variables and made available for other drives. Are always reading operations. More variables are read and lower will be the refresh rate of the data.

The operations management on demand is implemented in the QCL DR\_REQUEST procedure, is used for all the operations that must be performed to event (for example, a parameter write). The event is to be triggered to another drive.

To unleash the event you have to set a value in the *gbDrRequest* variable, to wait for the termination of the operation wait until the same value is set in the *gbDrAnswer* variable. Each task can have one or more variables of exchange value. We see a summary table of operations now implemented.

Operation	Qcl variables	Description
sets the drive in trajectory mode	no	This request sets the drive in trajectory mode. By this time the location is controlled by the 502BF tool.
start homing procedure	setting variables: <i>gbHomType</i> , <i>gbHomDir</i> , <i>glHomVel</i> , <i>glHomAcc</i> , <i>glHomDec</i> , <i>glHomOff</i> State variables: <i>gfHomActive</i> , <i>gfHomOk</i> , <i>gfHomError</i>	This request starts a homing procedure. Before you start this request set the corresponding variables. During the procedure the state variable <i>gfHomActive</i> is set to 1. At the end of the procedure if <i>gfHomOk</i> is set to 1 the procedure ended correctly, if instead <i>gfHomError</i> is set to 1 then the procedure ended incorrectly. The procedure uses the homing system drive implemented by ASCII NREF parameter. Refer to the description of this command.

Operation	Qcl variables	Description
sets current limit motor	<i>glCurrLim</i>	this procedure sets the maximum current in the motor to the value contained in the <i>glCurrLim</i> variable
set tracking error limit	<i>glMaxFerr</i>	This procedure sets the maximum following error with the value contained in the <i>glMaxFerr</i> variable

## Mode of operation

The drive is commanded by sending via CANOpen of location informations. This means that the drive is delegated the task of creating the current setting, speed and position.

The information passing via CANOpen are always relative and not absolute. This means that the drive moves the engine without knowing the absolute position and then without limits software (that are implemented in the 502BF product device). The actual position of the motor is accessed via CANOpen and available in the *encoder* and *posit* parameters of the device. The resolution of this information depends on transducer used in the drive. If you use a resolver with the *PRBASE* parameter to the value 16, for each revolution of the motor will be counted a value of 65536. In practice, however, by default there is a factor of division of 4 between the information provided by the drive and read by the 502BF. This factor can be programmed through a QDO command of the CANOPEN device. The division factor was inserted to prevent overflow when reading. It is necessary to (as with all other internal device) that at each sampling time of the position is changed to a value greater than of  $2^{15}$  (32768) bit encoder.

It is important to work properly that the sampling time of the CANOPEN device coincides with sampling time of all devices that use this bus. Due to restrictions of the Danaher drive, the sampling time shall be equal to one of the following values: 2,3,4,5,8,10,20 msec.

## CAN network diagnostic

Communication errors can be monitored with parameters *errflags*, *maxrxerr*, *maxtxerr*, *busload* and *maxtraffic* of the CANOPEN device.

The *busload* and *maxtraffic* parameters should be used to verify the actual use of the bus. The value must not exceed the 80.0% of percentage. This data determines the number of drives that you can use. The use of the bus depends on the following factors:

Factor	Description
Sample time device	at each sampling time are sent the movement information and it reads the actual motor position
CAN bus speed	the speed defines the duration of the message so inversely proportional
number drives	There are several drives and multiple emails to be sent into the sample time
SDO use	the use of SDO involves sending messages more up to a maximum of one message for each sample time
communication errors	communication errors may alter the content of a message that will immediately be retransmitted increasing traffic

Documento generato automaticamente da **Qem Wiki** - <https://wiki.qem.it/>

Il contenuto wiki è costantemente aggiornato dal team di sviluppo, è quindi possibile che la versione online contenga informazioni più recenti di questo documento.