PC Can Console user guide

Version: 1.0
Status: DRAFT

Issued by	Role	Date
Davide Nobili		09/03/2020

Approved by	Role	Date

Document Change History			
Date	Version	Changed By	Change Description
09/03/2020	1.0	Davide Nobili	First release for ZpCanConsole 0.97

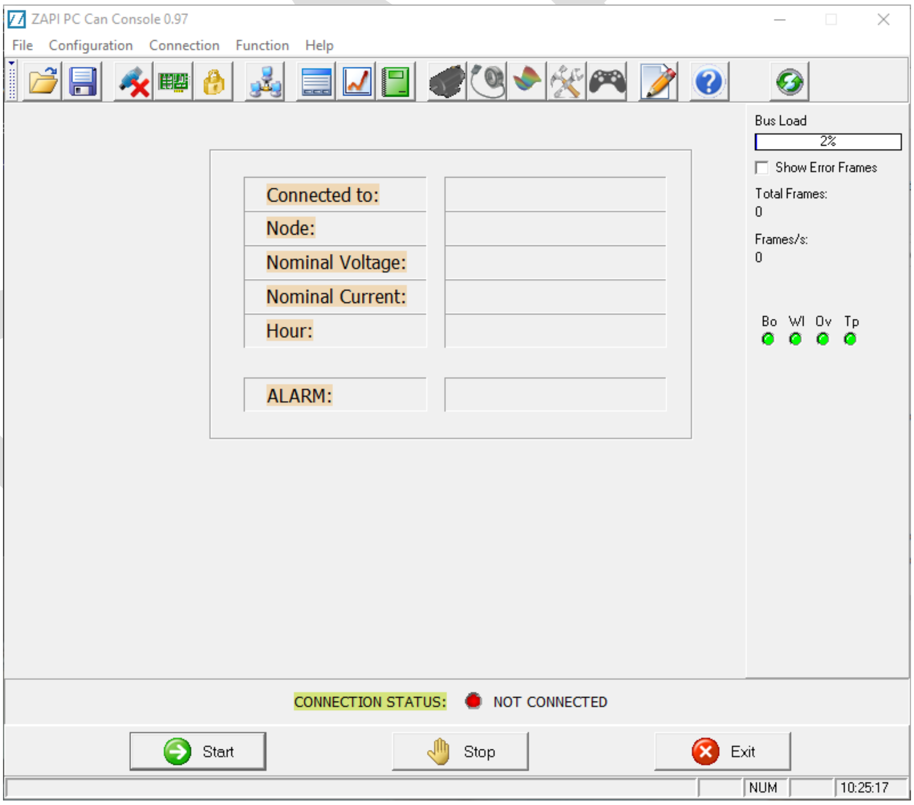
Index

1. Introduction

Zapi Can Console software permit to test and adjust the setup configuration of the Zapi Controller.

Functionalities:

- Error check: actual errors, log book;
- Tester: check motor controller internal value and input/output status;
- Parameter: read and write motor controller parameter. Full parameter list download to *.csv and load from *.csv;
- EEPROM: read and write EEPROM data;
- RAM: read and write RAM data;
- *.z86 software upload: upload motor controller firmware (only via CANbus);
- Error frame counter: check error frame over the CANbus network (only with IXXAT dongle);
- Motor autotuning: estimate motor parameter onsite;
- Autoteching sensor: estimate absolute sensor offset;
- CANopen module: simulate CANopen command.



Bus Load 2%

□ Show Error Frames

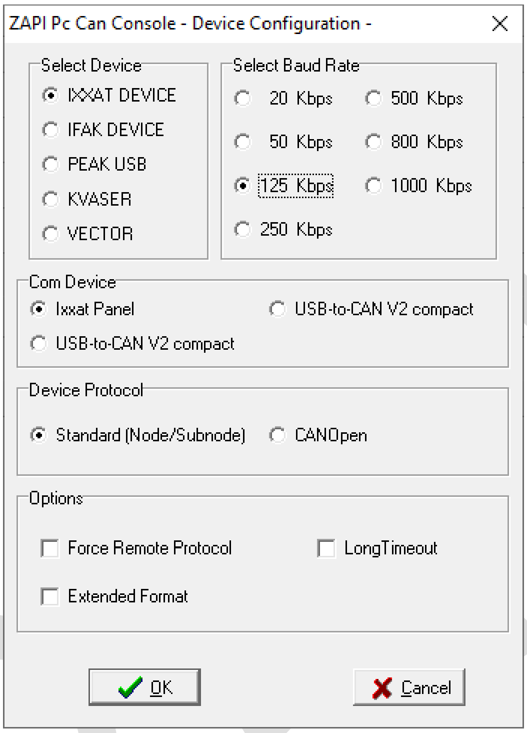
Total Frames: 0

Frames/s: 0

CONNECTION STATUS: ● NOT CONNECTED

2. CAN Device configuration

Click on or “Configuration” → “Can Device” or press CTRL+C to select the USB-to-CAN device, the Baud Rate and the Communication protocol to connect the Zapi Can Console with the Zapi Controller.



ZAPI Pc Can Console - Device Configuration -

Select Device
● IXXAT DEVICE
○ IFAK DEVICE
○ PEAK USB
○ KVASER
○ VECTOR

Select Baud Rate
○ 20 Kbps
○ 50 Kbps
● 125 Kbps
○ 250 Kbps
○ 500 Kbps
○ 800 Kbps
○ 1000 Kbps

Com Device
● Ixxat Panel
○ USB-to-CAN V2 compact
○ USB-to-CAN V2 compact

Device Protocol
● Standard (Node/Subnode)
○ CANOpen

Options
□ Force Remote Protocol
□ LongTimeout
□ Extended Format

Device Protocol

Set the node discovery method. For newer application set CANOpen, former application should use Standard mode. This option affect also the way the node are displayed in the list of the available node

- Standard: Node.Subnode (e.g. 2.0 → Traction µC master, 2.1 → Traction µC slave);
- CANopen: Canopen Node (e.g. 8 → Traction µC master, 9 → Traction µC slave).

NOTE: not all the target inverter support both protocol

Select Baud Rate

In this example, it is used an IXXAT Device set with a 125 kbps Baud Rate.

NOTE: the CANbus speed is machine driven, the same controller and SW can be used at different CANbus speed.

Com Device

Two different selection method are possible:

- Selecting “Ixxat Panel” the IXXAT Select VCI Device window will open in order to select the specific IXXAT dongle.
- Other options allow a direct selection of the CAN device to use, without opening IXXAT Select VCI Device windows.

Force Remote Protocol

It is recommended to keep this option always unselected.

Controller communicate with the Zapi zpCANconsole trough two different protocol.

- Legacy protocol (ULTRA protocol or Remote Protocol)
- SDO based protocol (faster).

Force Remote Protocol flag force the communication protocol. Suggested to use only in case of communication problem or in case of old controller. Do not use for double microcontroller (2uC) Zapi Controller.

Note: if remote protocol is not selected, the communication start with the SDO protocol. If this communication fails it automatically switch to the remote protocol and automatically set this flag.

Long Timeout

Increase the timeout time in case of no answer (useful for remote use of the Console). For standard purpose keep it not selected.

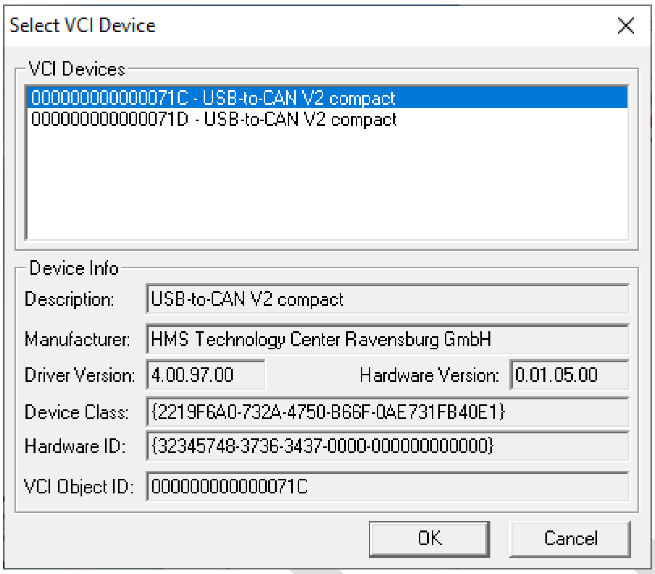
Extended Format

Legacy mode, do not use.

Confirm with the OK button.

If selected the “Ixxat Panel” option, will open the Select VCI Device window.
Using windows 10 following installation from IXXAT official internet site are needed:

- VCI V4 driver
- VCI V2 API-Addon (VCI2 on VCI4)



Select VCI Device

VCI Devices

000000000000071C - USB-to-CAN V2 compact
000000000000071D - USB-to-CAN V2 compact

Device Info

Description: USB-to-CAN V2 compact

Manufacturer: HMS Technology Center Ravensburg GmbH

Driver Version: 4.00.97.00 Hardware Version: 0.01.05.00

Device Class: {2219F6A0-732A-4750-B66F-0AE731FB40E1}

Hardware ID: {32345748-3736-3437-0000-000000000000}

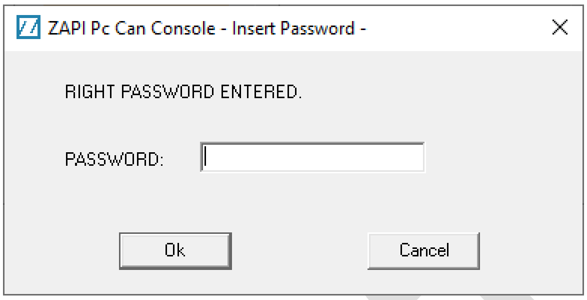
VCI Object ID: 000000000000071C

Confirm with the OK button.

By pressing the button it is possible to release the CAN Device for other use.

3. Enter password

Click on 🔒 button or “Configuration” → “Enter Password”.



ZAPI Pc Can Console - Insert Password -

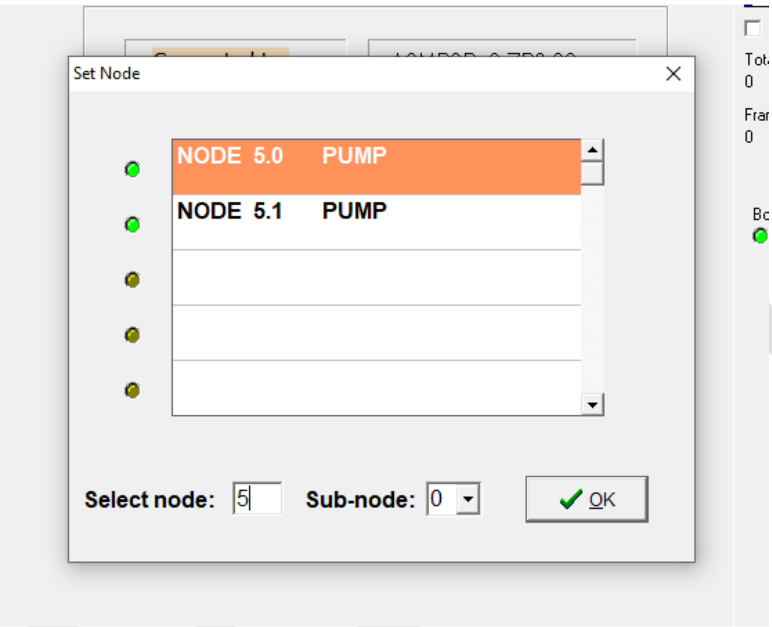
RIGHT PASSWORD ENTERED.

PASSWORD:

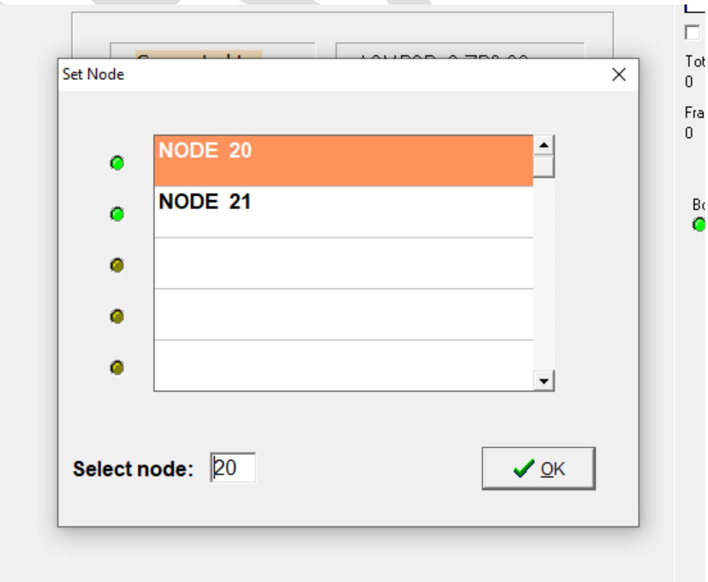
4. Select Node

Click on button or "Connection" → "Set Node" to select the Zapi Controller to connect.

E.g. Standard Protocol: NODE 5.0 → Pump µC master, NODE 5.1 → Pump µC slave:



E.g. CANopen Protocol: NODE 20 → Pump µC master, NODE 21 → Pump µC slave:

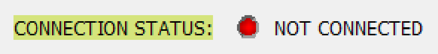


Once selected the node and the sub-node click on the "OK" button.

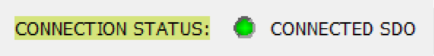
5. Connect with Zapi Controller

Click on button in the main window.

If the connection is successful, the status LED in the lower part of the main window change from red to green,

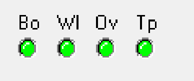


CONNECTION STATUS: 🔴 NOT CONNECTED

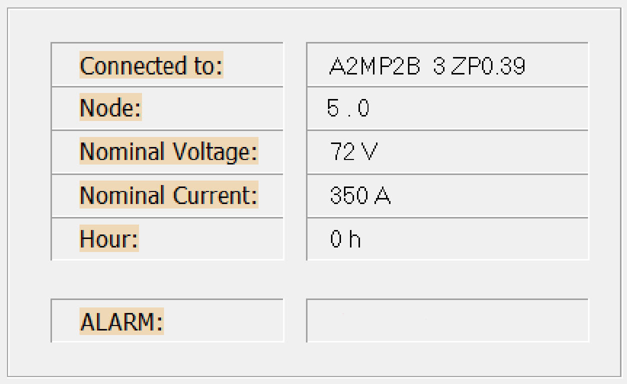


CONNECTION STATUS: 🟢 CONNECTED SDO

on the right central side of the window 4 green LEDs will light.



and on the main window will appear the information related to the target controller: software version, CAN bus node number, nominal battery voltage and current, hour counter, and if present the actual alarm.



Connected to:	A2MP2B 3 ZP0.39
Node:	5 . 0
Nominal Voltage:	72 V
Nominal Current:	350 A
Hour:	0 h
ALARM:	

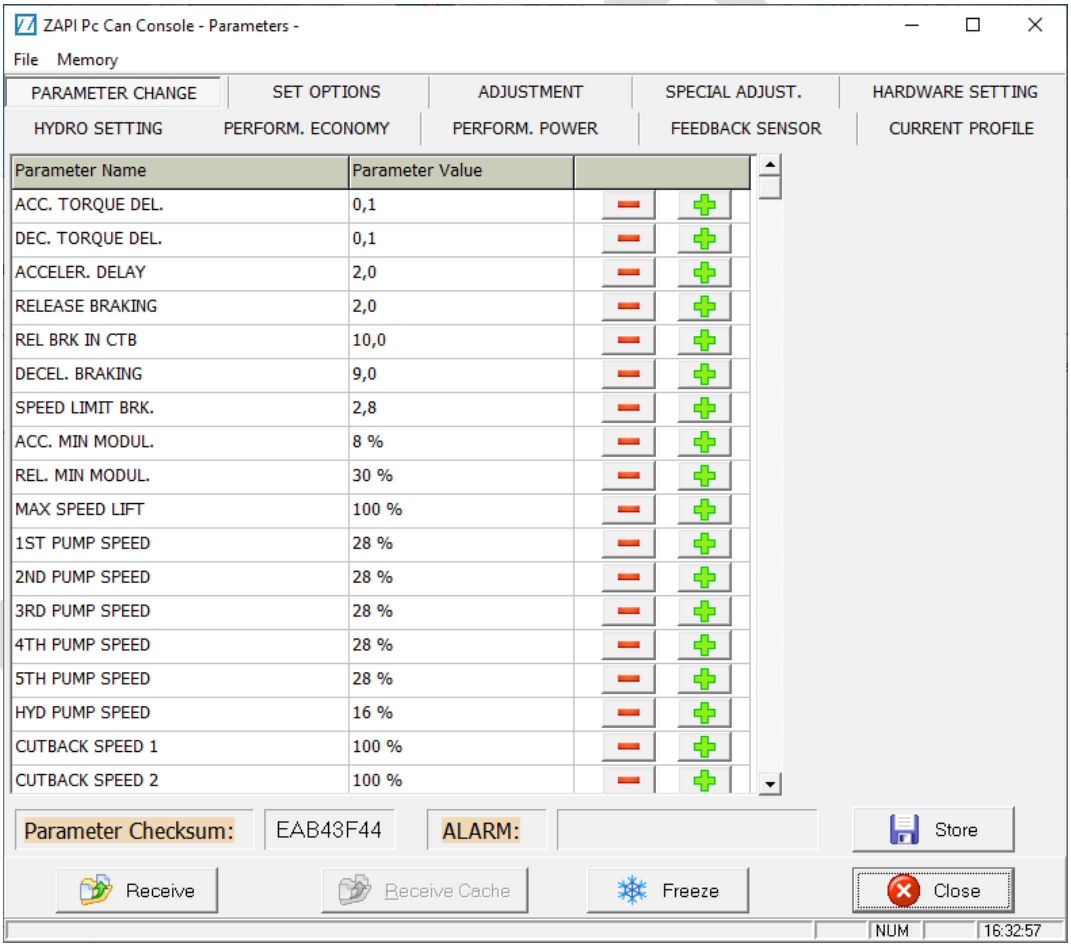
Note: if you are attempting an SDO communication but the speed is slow, check in Device Configuration (Chapter 2) and de-select Force Remote Protocol.

6. Parameters management

Click on button or "Function" → "Parameters" or press CTRL+P to open the parameters window and on Receive button to download from Zapi Controller actual parameters configuration.

If the password inserted is right (Chapter 3), parameter modification is allowed and plus/minus buttons − + are displayed.

If connected with SDO protocol, it is possible to double click con parameter value and type the correct value directly by keyboard. Please note that in some cases this is the only way to set the correct parameter value.



Parameter Name	Parameter Value	
ACC. TORQUE DEL.	0,1	− +
DEC. TORQUE DEL.	0,1	− +
ACCELER. DELAY	2,0	− +
RELEASE BRAKING	2,0	− +
REL BRK IN CTB	10,0	− +
DECEL. BRAKING	9,0	− +
SPEED LIMIT BRK.	2,8	− +
ACC. MIN MODUL.	8 %	− +
REL. MIN MODUL.	30 %	− +
MAX SPEED LIFT	100 %	− +
1ST PUMP SPEED	28 %	− +
2ND PUMP SPEED	28 %	− +
3RD PUMP SPEED	28 %	− +
4TH PUMP SPEED	28 %	− +
5TH PUMP SPEED	28 %	− +
HYD PUMP SPEED	16 %	− +
CUTBACK SPEED 1	100 %	− +
CUTBACK SPEED 2	100 %	− +

Parameter Checksum: EAB43F44 ALARM:

Once changed one or more parameters click on Store button and wait until it return bright.

Most parameters modification need a key cycle to be active.

NOTE: always change parameter with motor stopped (0 rpm and no direction active) if "Store" button is pressed with motor active, the motor is stopped and then restarted.

6.1 Manage parameters in *.csv extension

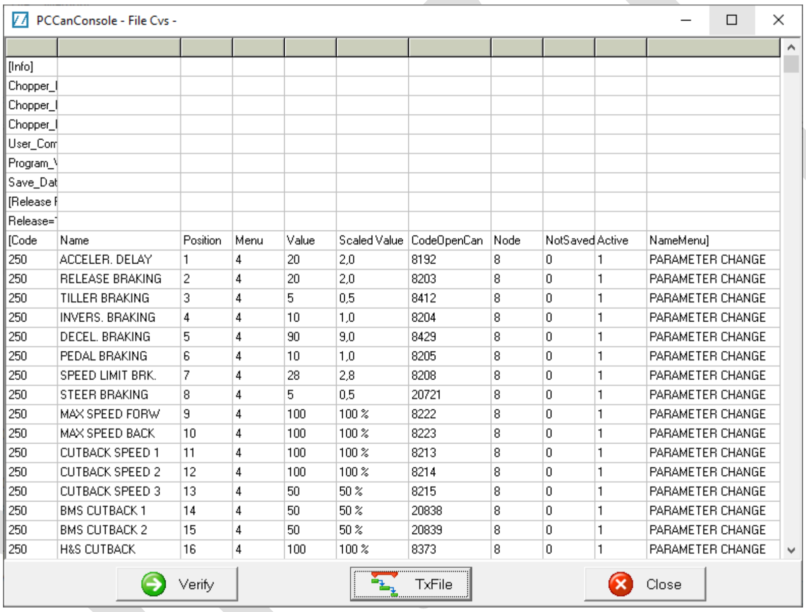
6.1.1 Download parameters

During the SW development and on field troubleshooting is often requested the parameters configuration.

On the parameters window click on “File” → “Save” to download the parameters configuration in *.csv extension.

6.1.2 Verify parameters

On the parameters window click on “File” → “Open” to load a parameters configuration previously saved (*.csv extension).



[Info]										
Chopper_I										
Chopper_I										
Chopper_I										
User_Com										
Program_\										
Save_Dat										
[Release F										
Release=										
[Code	Name	Position	Menu	Value	Scaled Value	CodeOpenCan	Node	NotSaved	Active	NameMenu]
250	ACCELER. DELAY	1	4	20	2,0	8192	8	0	1	PARAMETER CHANGE
250	RELEASE BRAKING	2	4	20	2,0	8203	8	0	1	PARAMETER CHANGE
250	TILLER BRAKING	3	4	5	0,5	8412	8	0	1	PARAMETER CHANGE
250	INVERS. BRAKING	4	4	10	1,0	8204	8	0	1	PARAMETER CHANGE
250	DECEL. BRAKING	5	4	90	9,0	8429	8	0	1	PARAMETER CHANGE
250	PEDAL BRAKING	6	4	10	1,0	8205	8	0	1	PARAMETER CHANGE
250	SPEED LIMIT BRK.	7	4	28	2,8	8208	8	0	1	PARAMETER CHANGE
250	STEER BRAKING	8	4	5	0,5	20721	8	0	1	PARAMETER CHANGE
250	MAX SPEED FORW	9	4	100	100 %	8222	8	0	1	PARAMETER CHANGE
250	MAX SPEED BACK	10	4	100	100 %	8223	8	0	1	PARAMETER CHANGE
250	CUTBACK SPEED 1	11	4	100	100 %	8213	8	0	1	PARAMETER CHANGE
250	CUTBACK SPEED 2	12	4	100	100 %	8214	8	0	1	PARAMETER CHANGE
250	CUTBACK SPEED 3	13	4	50	50 %	8215	8	0	1	PARAMETER CHANGE
250	BMS CUTBACK 1	14	4	50	50 %	20838	8	0	1	PARAMETER CHANGE
250	BMS CUTBACK 2	15	4	50	50 %	20839	8	0	1	PARAMETER CHANGE
250	H&S CUTBACK	16	4	100	100 %	8373	8	0	1	PARAMETER CHANGE

Click on Verify button to show the differences between the parameters configuration on the Zapi Controller and the loaded parameters configuration.

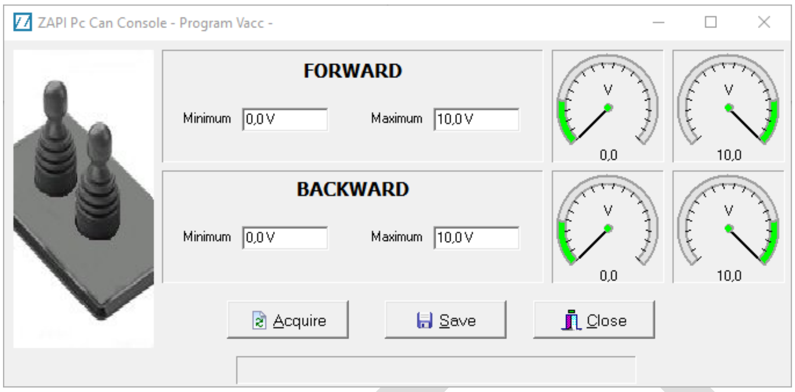
6.1.3 Restore parameters

Once loaded the parameter configuration (*.csv extension) click on TxFile button to restore the loaded parameters configuration on the Zapi Controller.

Click on Close button to come back in the parameters window.

7. Accelerator potentiometer acquisition

Click “Function” → “Program Vacc” or press CTRL+V to open the accelerator acquisition window.



ZAPI Pc Can Console - Program Vacc -

FORWARD

Minimum 0,0 V Maximum 10,0 V

BACKWARD

Minimum 0,0 V Maximum 10,0 V

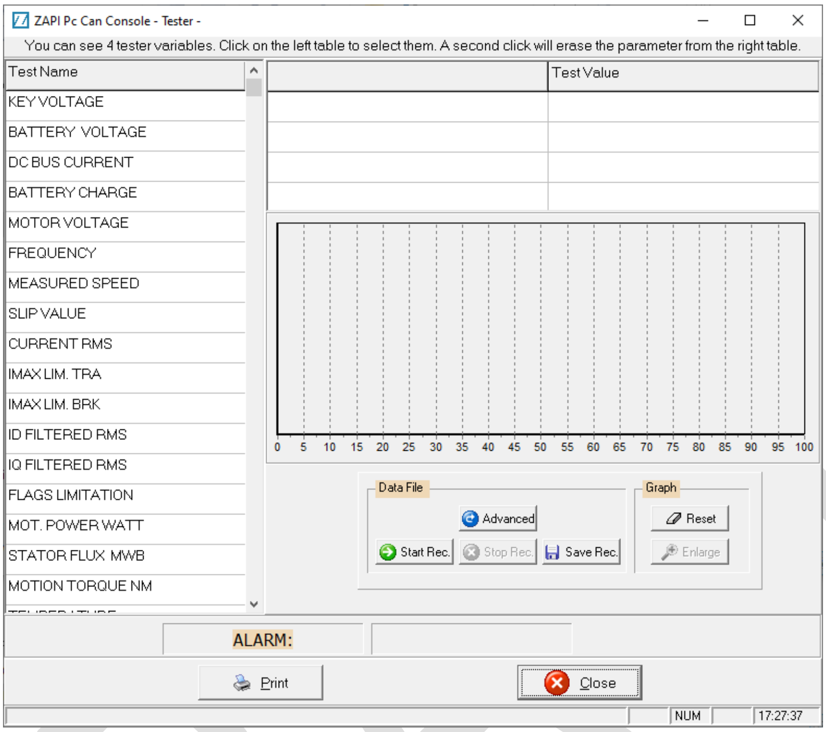
Click on Acquire button to start the acquisition procedure. By pressing the accelerator (combined with the direction switch) the referred voltage box follow the accelerator voltage. Press the accelerator over its full range for both direction (forward/backward).

Click on Save button to save the values acquired.

Click on Close button to close the accelerator acquisition window.

8. Tester

Click on button or “Function” → “Tester” or press CTRL+T to open the tester window.



ZAPI Pc Can Console - Tester -

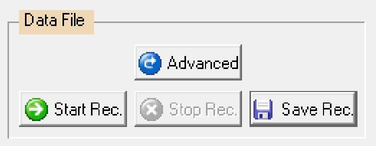
You can see 4 tester variables. Click on the left table to select them. A second click will erase the parameter from the right table.

Test Name
KEY VOLTAGE
BATTERY VOLTAGE
DC BUS CURRENT
BATTERY CHARGE
MOTOR VOLTAGE
FREQUENCY
MEASURED SPEED
SLIP VALUE
CURRENT RMS
IMAX LIM. TRA
IMAX LIM. BRK
ID FILTERED RMS
IQ FILTERED RMS
FLAGS LIMITATION
MOT. POWER WATT
STATOR FLUX MWB
MOTION TORQUE NM

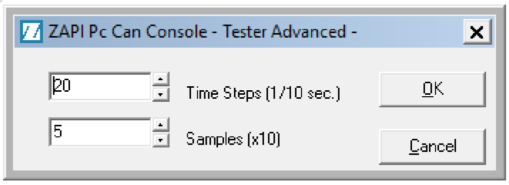
With this window is possible to view up to four variables on real time on the table and on the graphic chart.

Data file

With the following box is possible to save a variable acquisition on a *.csv file.



Click on Advanced button to set the sampling time and the samples acquired on each period.



ZAPI Pc Can Console - Tester Advanced -

20 Time Steps (1/10 sec.)

5 Samples (x10)

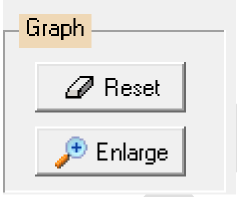
Click on Start Rec. button to start the recording.

Click on Stop Rec button to stop the recording.

Click on Save Rec. button to save the *.csv file.

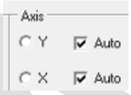
Graph options

With the following box is possible to manage the graph view.

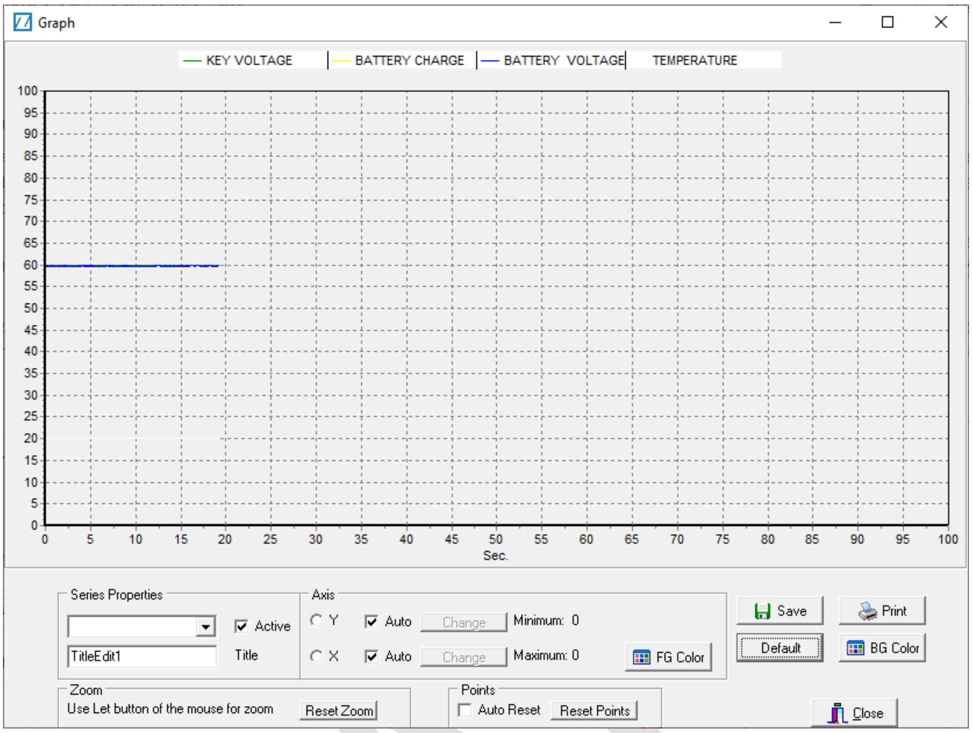


Click on “Reset” button to clear the graph and restart the measure.

Click on “Enlarge” button to open the graph window.

In order to see the full range of acquisition  :

- Y axis and then check both Auto flag.
- X axis and then check both Auto flag.



☑ Active

TitleEdit1

○ Y ☑ Auto Minimum: 0

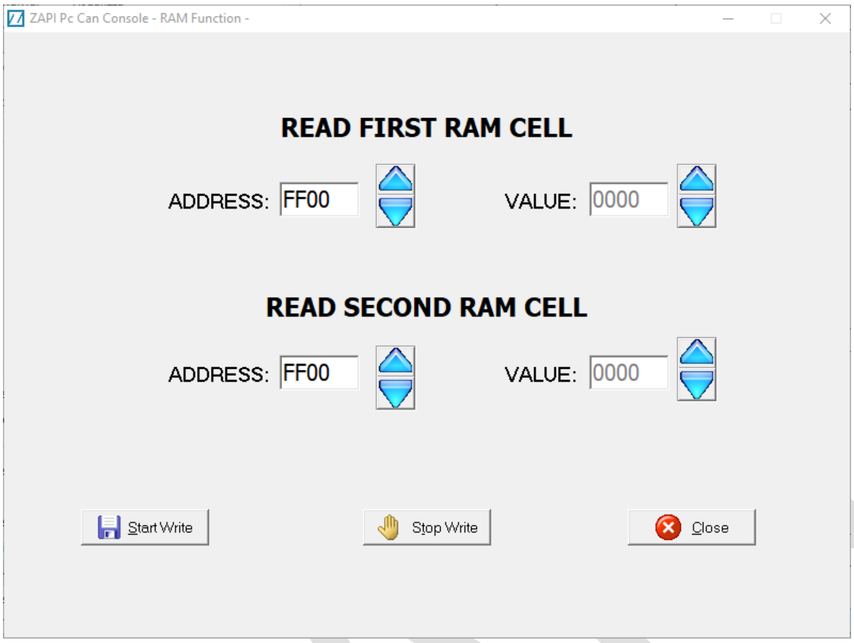
○ X ☑ Auto Maximum: 0

Use Let button of the mouse for zoom

□ Auto Reset

9. RAM management

Click on “Function” → “RAM Function” to open the following window.



ZAPI Pc Can Console - RAM Function -

READ FIRST RAM CELL

ADDRESS: FF00 VALUE: 0000

READ SECOND RAM CELL

ADDRESS: FF00 VALUE: 0000

Two RAM cells are available at the same time.

9.1 Read RAM

Write on the address field to view the desired RAM cell.

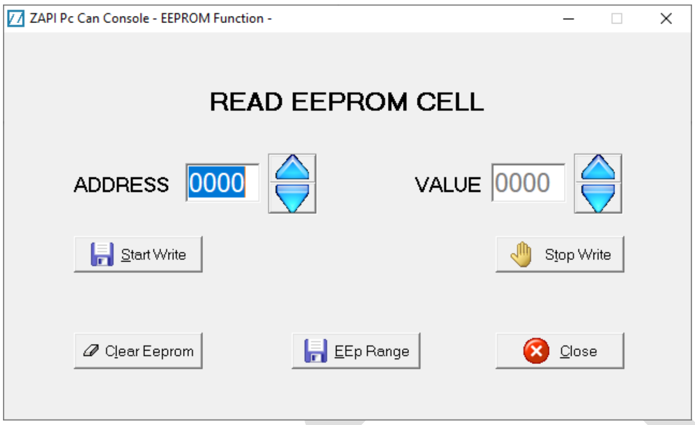
The value read is related to two RAM cells, the right byte is referred to the even address, left byte is referred to the odd address.

9.2 Write RAM

Click on Start Write button to enable the RAM writing (the value fields characters changes from grey to black). Write the wanted value. Click on Stop Write button to validate the new value.

10. EEPROM management

Click on “Function” → “EEPROM Function” to open the following window.



ZAPI Pc Can Console - EEPROM Function -

READ EEPROM CELL

ADDRESS 0000

VALUE 0000

10.1 Read EEPROM

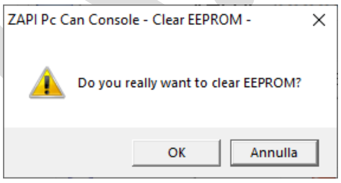
As described for Read RAM.

10.2 Write EEPROM

As described for Write RAM.

10.3 Clear EEPROM

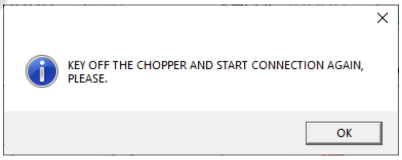
Click on Clear Eeprom button to start the clear EEPROM procedure. This procedure restore the default values for the unlocked parameters.



ZAPI Pc Can Console - Clear EEPROM -

Do you really want to clear EEPROM?

Click on “OK”.



KEY OFF THE CHOPPER AND START CONNECTION AGAIN, PLEASE.

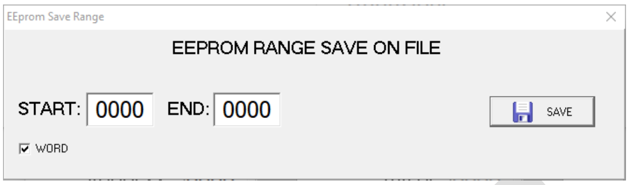
Click on “OK”.

Recycle the key.

After a CLEAR EEPROM procedure is done, at the next key on “PARAM CONFIG 01” warning is shown.

10.4 Save EEPROM Range

Click on EEp Range button to open the EEPROM Save Range window.



EEprom Save Range

EEPROM RANGE SAVE ON FILE

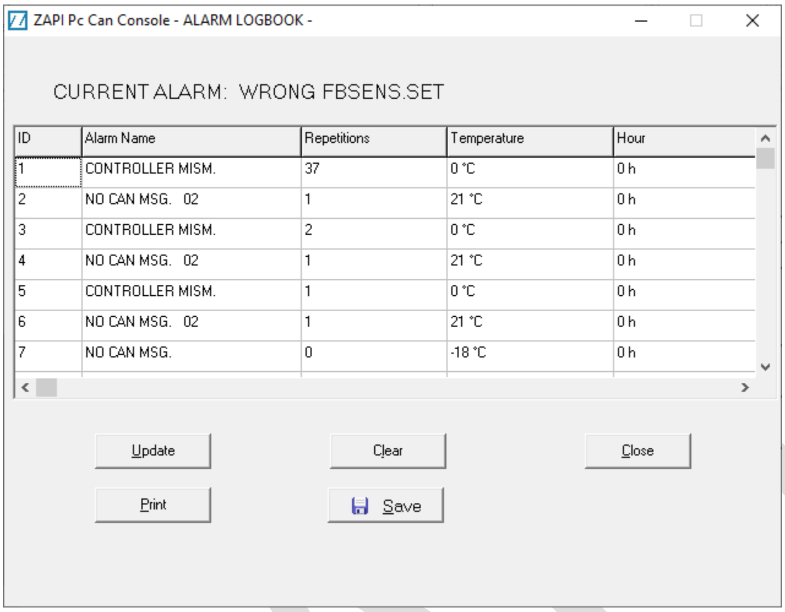
START: 0000 END: 0000

☑ WORD

Write the addresses range desired and click on SAVE button to save on *.csv extension the EEPROM values.

11. Alarms logbook

Click on button or “Function” → “Alarm Logbook” or press CTRL+A to open the following window.



CURRENT ALARM: WRONG FBSENS.SET

ID	Alarm Name	Repetitions	Temperature	Hour
1	CONTROLLER MISM.	37	0 °C	0 h
2	NO CAN MSG. 02	1	21 °C	0 h
3	CONTROLLER MISM.	2	0 °C	0 h
4	NO CAN MSG. 02	1	21 °C	0 h
5	CONTROLLER MISM.	1	0 °C	0 h
6	NO CAN MSG. 02	1	21 °C	0 h
7	NO CAN MSG.	0	-18 °C	0 h

It shows all the alarms present or that has been occurred during operation of the controller. The alarms are listed with the number of occurrences, the inverter temperature and the time at which the event occurs.

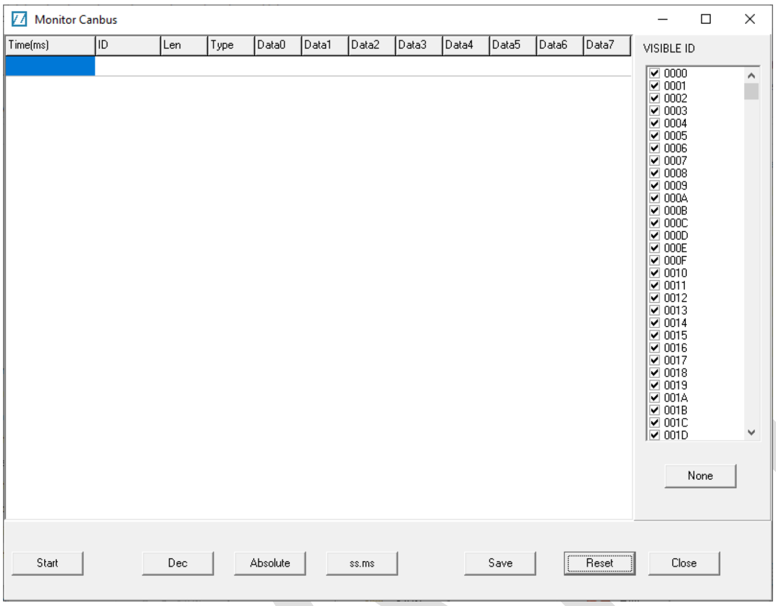
Click on Update button to refresh the alarms list.

Click on Clear button to reset the alarms list.

Click on Print or Save to print or export the alarms list in *.pdf or *.csv extension.

12. Monitor Canbus

This section implement a simple can bus message viewer. To activate it click to “Function” → “Monitor Canbus”.



On the right field is possible to select the desired messages ID. Click on None or All buttons to deselect/select all the IDs.

12.1 Record/view a CAN trace

Click on Start button to begin the recording of the CAN messages allowed on the right field.

Click on Stop button to stop the recording.

Click on Save button to export the CAN trace recorded on a *.txt file.

Click on Reset button to delete all CAN messages on the window.

12.2 Data format

Click on Dec or Hex buttons to change decimal/hexadecimal data format.

When the button Hex is present the data format is decimal.

When the button Dec is present the data format is hexadecimal.

12.3 Time format

Click on Relative or Absolute buttons to swap respectively the time format to relative and absolute and on ss.ms or hh:mm:ss.ms to swap respectively the time notation:

- Relative → Time delay from the previous CAN message;
- Absolute → Time delay from the configuration of the CAN device;
- ss:ms → seconds:milliseconds;
- hh:mm:ss:ms → hours:minutes:seconds:milliseconds.

13. Download the SW with *.z86 file

13.1 *.z86 file package

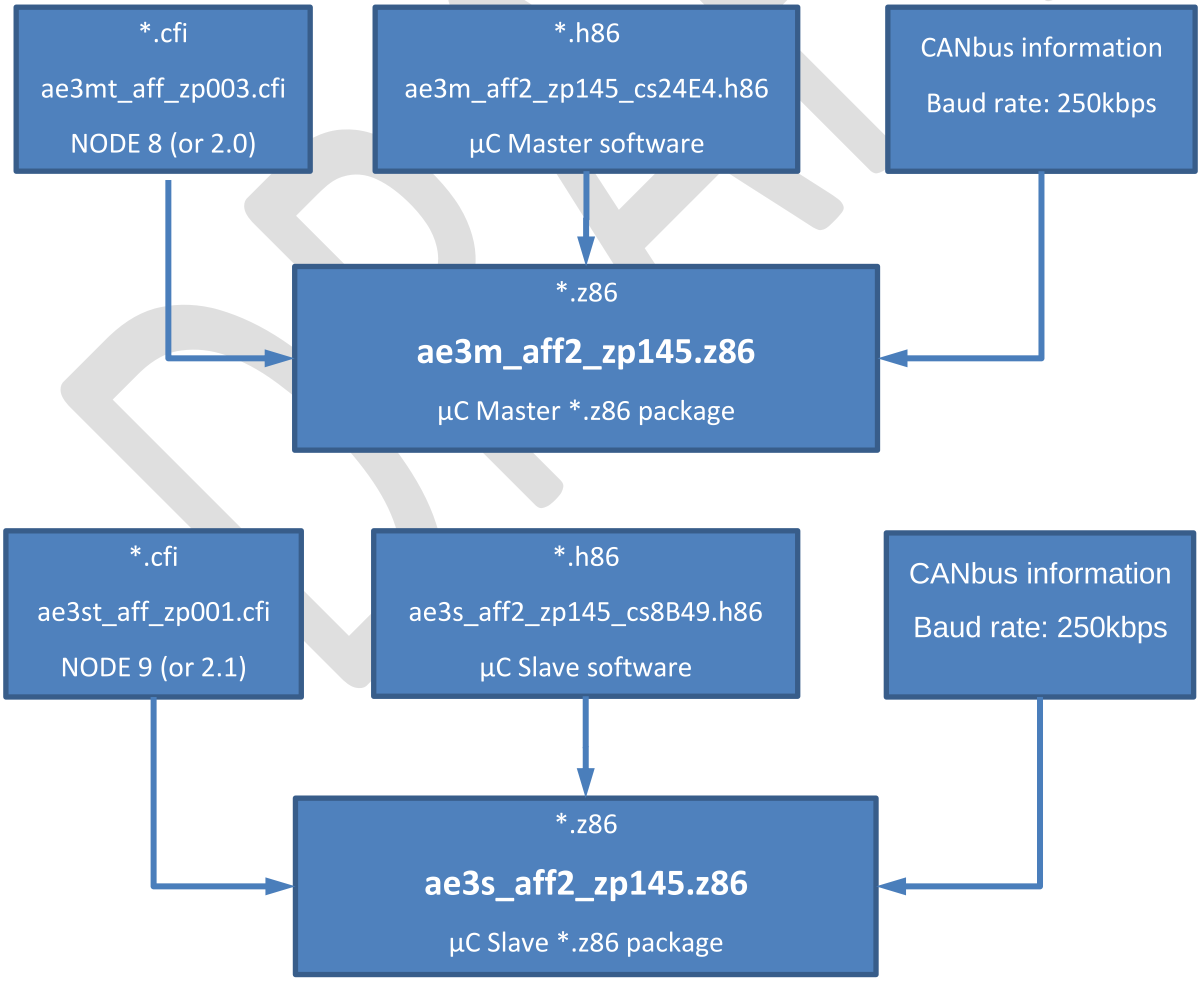
*.z86 file is a software package that include *.cfi file, *.h86 file and CANbus speed information.

*.z86 is needed to load a new firmware on a Zapi motor controller by using Zapi Smart Console or Zapi Can Console.

*.z86 package is used only in software upload via CANbus, the legacy serial firmware upload it is not possible.

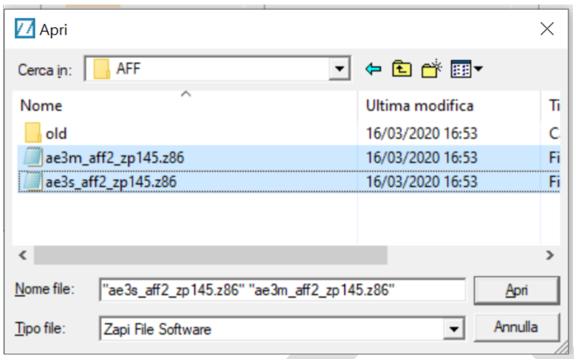
In new double microcontroller inverter, two different *.z86 packages are needed to successfully update the inverter software.

Following an example of *.z86 package composition, the package will be directly provided by Zapi beside standard *.h86 software.



13.2 Software upload over CANbus network

Click on , and the file selection windows will appear.



Apri

Cerca in: AFF

Nome	Ultima modifica	Ti
old	16/03/2020 16:53	C
ae3m_aff2_zp145.z86	16/03/2020 16:53	Fi
ae3s_aff2_zp145.z86	16/03/2020 16:53	Fi

Nome file: "ae3s_aff2_zp145.z86" "ae3m_aff2_zp145.z86"

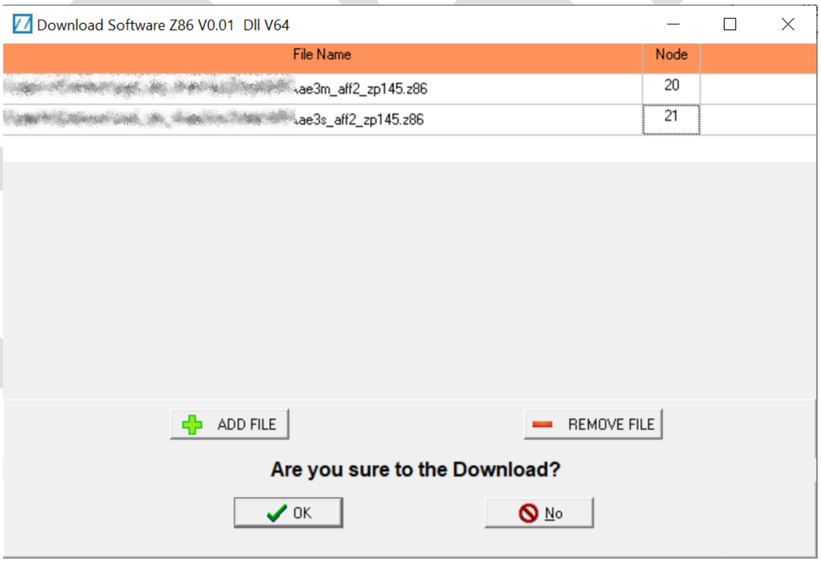
Tipo file: Zapi File Software

Select all the desired file in *.z86 format and click “OPEN”, Download Software windows will appear.

Selected software file are listed, and implicit node ID is shown.

The connection speed is the same used at Zapi Can Console device configuration, chapter “2. CAN Device configuration”.

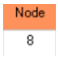
It is mandatory that all the devices to be uploaded communicate at the same CANbus speed.



Download Software Z86 V0.01 Dll V64

File Name	Node
[redacted]\ae3m_aff2_zp145.z86	20
[redacted]\ae3s_aff2_zp145.z86	21

Are you sure to the Download?

Select the correct node ID by double-click  .

Once the node ID configuration is done click ✓ OK to start the upload or ⦸ No to exit.

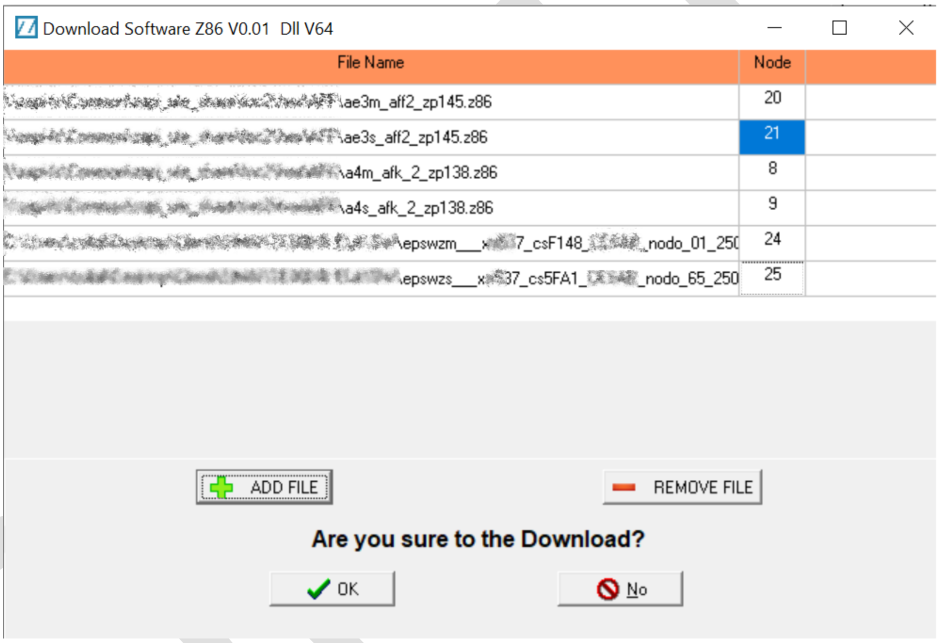
After clicking ✓ OK a green bar that represent the percentage of the upload will be shown over the file name.

When the upload will be completed the windows will be automatically close.

It is possible to add more file in the upload list by clicking ✚ ADD FILE.

Select a row from the list and click ▬ REMOVE FILE to remove any unwanted file in the list.

Below and example with six file to be loaded on three different inverter.



Download Software Z86 V0.01 Dll V64

File Name	Node
[redacted]\ae3m_aff2_zp145.z86	20
[redacted]\ae3s_aff2_zp145.z86	21
[redacted]\a4m_afk_2_zp138.z86	8
[redacted]\a4s_afk_2_zp138.z86	9
[redacted]\epswzm__x[redacted]7_csF148_[redacted]_nodo_01_250	24
[redacted]\epswzs__x[redacted]37_cs5FA1_[redacted]_nodo_65_250	25

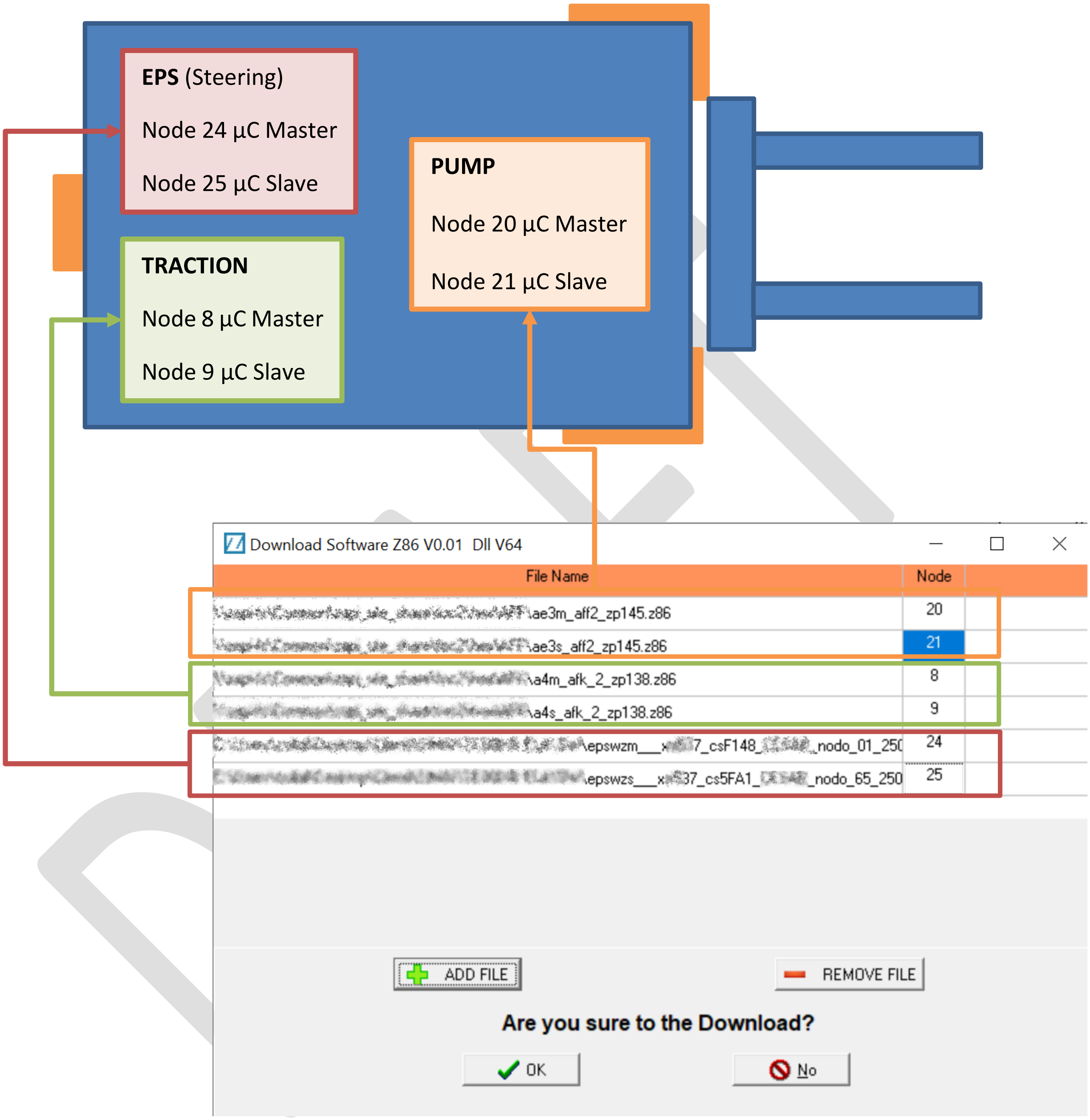
Are you sure to the Download?

Possible errors:

- TIMEOUT: it is impossible to reach the selected node. Check ether the selected file and the correspondent node. Check if the node is connectable with Zapi Can Console in order to diagnostic the main connection problem.

13.3 Example

Below explained the relation between *.z86 file and selected Node ID in a graphical example of a complete machine.



File Name	Node
[redacted]\ae3m_aff2_zp145.z86	20
[redacted]\ae3s_aff2_zp145.z86	21
[redacted]\a4m_afk_2_zp138.z86	8
[redacted]\a4s_afk_2_zp138.z86	9
[redacted]\epswzm___x[redacted]7_csF148_[redacted]_nodo_01_250	24
[redacted]\epswzs___x[redacted]37_cs5FA1_[redacted]_nodo_65_250	25

Are you sure to the Download?

14. Motor tuning - Sensor acquisition

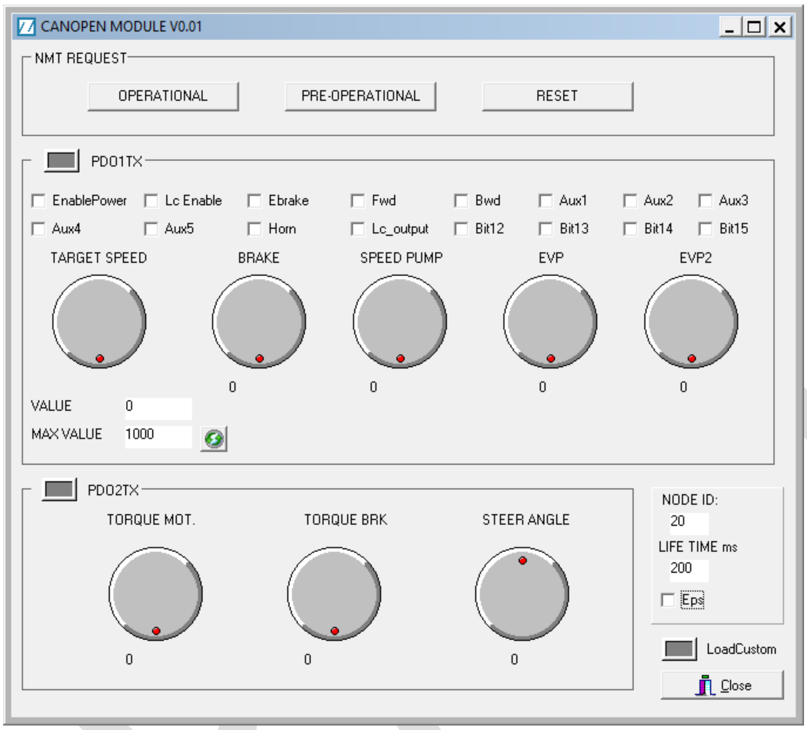
View the following files:

- "Autotuning procedure.pdf";
- "Absolute sensor acquisition.pdf".

15. CANopen module

With this module is possible to manage the CANopen commands by using a graphic interface.

Click on button to open the CANopen Joypad window.



VALUE 0

MAX VALUE 1000

NODE ID: 20

LIFE TIME ms 200

By using this window is possible to test a Zapi Controller configured as CANopen system.

Write on the NODE ID field the decimal value (e.g. Traction → 8, Pump → 20).

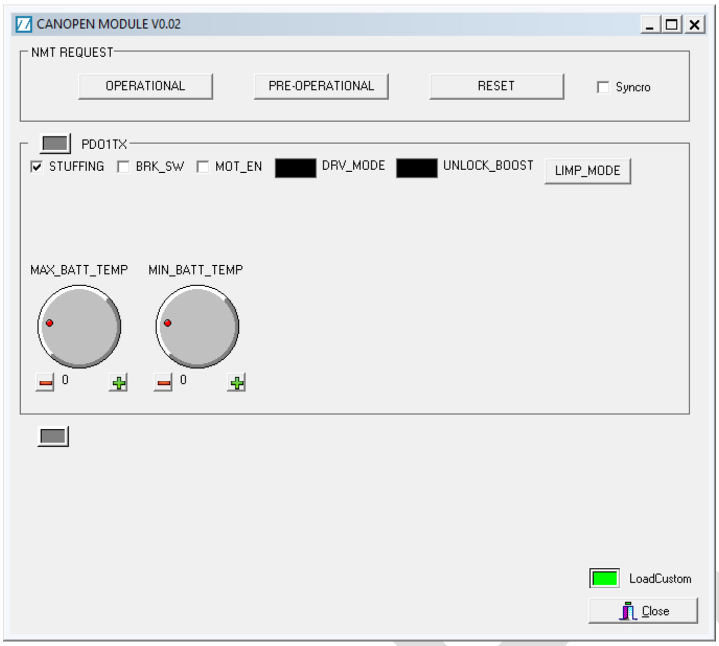
Click on PDO1TX button to start to send the CAN message.

Once pressed the OPERATIONAL button, the controller will carry out the motor speed and the outputs selected ("SPEED PUMP" throttle is referred to the DC pump of the Zapi Combi Controllers).

15.1 Custom Joypad

Click on LoadCustom button to open a custom joypad in *.txt extension.

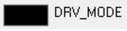
Example 1.

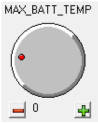


*.txt file:

```
[RELASE_FILE]
0001

[RELASE]
0001

[PDO1TX]
ID            ; 0832
TRANSMIT      ; 38      // ms

//element     ; label         ; start bit; size; min; max; resolution; offset;
STUFFING_BIT  ; STUFFING      ;        31;    1;   0;   1;          1;      0;
TCHECKBOX     ; BRK_SW        ;        19;    1;   0;   1;          1;      0;
TCHECKBOX     ; MOT_EN        ;        28;    1;   0;   1;          1;      0;
TVRNUMEDIT    ; DRV_MODE      ;        32;    3;   0;   4;          1;      0;
TVRNUMEDIT    ; UNLOCK_BOOST  ;        35;    2;   0;   2;          1;      0;

//element     ; label         ;    ID; byte0; byte1; byte2; byte3; byte4; byte5; byte6; byte7;
TBUTTON       ; LIMP_MODE     ; 0000;     64;      0;      0;      0;      0;      0;      0;      0;

//element     ; label         ; start bit;  size; min; max; resolution; offset;
TVRWHEEL      ; MAX_BATT_TEMP;        37;    11; -40; 100;        0,2;    -40;
TVRWHEEL      ; MIN_BATT_TEMP;        48;    11; -40; 100;        0,2;    -40;

[END]
```

- [PDO1TX] → Defines a new CAN message and its name → PDO1TX. The following elements are included in the PDO1RX box;
- ID; 0832 → Defines the CAN message ID (decimal);
- TRANSMIT; 38 → Defines the CAN message period in milliseconds (sent once pressed OPERATIONAL);
- STUFFING_BIT; STUFFING; → Defines a stuffing bit checkbox named "STUFFING" → ☑ STUFFING. Toggles the bit 31 (bit 7 of the byte3) on each message;
- TCHECKBOX; BRK_SW; → Defines a checkbox named "BRK_SW" → □ BRK_SW. ;

- TVRNUMEDIT; DRV_MODE; → Defines a editable number box named “DRV_MODE” → ;

- TVRWHEEL; MAX_BATT_TEMP; → Define a potentiometer named “MAX_BATT_TEMP” → ;

- TBUTTON; LIMP_MODE; → Define a button named “LIMP_MODE” → ;

- [END] → Define the end of the PDO1TX box.

By adding STUFFING_BIT, TCHECKBOX, TVRNUMEDIT and TVRWHEEL elements is possible to create the CAN message. Rules of the *.txt file for STUFFING_BIT, TCHECKBOX, TVRNUMEDIT and TVRWHEEL elements:

- Start bit: starting bit of the data (from 0 to 63);
- Size: size of the data (for STUFFING_BIT and TCHECKBOX is fixed at 1);
- Min: minimum value of the data (only for TVRNUMEDIT and TVRWHEEL);
- Max: maximum value of the data (only for TVRNUMEDIT and TVRWHEEL);
- Resolution: divider of the data output;
- Offset: offset of the output data (only for TVRNUMEDIT and TVRWHEEL).

TBUTTON element create a button that send one CAN message with fixed ID and datas for each click. Rules of the *.txt file to build a custom joypad for TBUTTON element:

- ID: CAN message ID (decimal);
- Byte 0-7: data byte (decimal);

Example 2.

```
[RELASE_FILE]
0001

[RELASE]
0001

[PDO1TX]
ID; 0520
TRANSMIT      ; 38   //ms
//componente  ; label       ;  start bit;  size;  min;  max;resolution;offset;
STUFFING_BIT  ; STUFFING    ;         31;     1;    0;    1;       1;         0;
TCHECKBOX     ; EN_POWER    ;         16;     1;    0;    1;       1;         0;
TCHECKBOX     ; EN_LC       ;         17;     1;    0;    1;       1;         0;
TCHECKBOX     ; OUT_NB      ;         18;     1;    0;    1;       1;         0;
TCHECKBOX     ; FW          ;         19;     1;    0;    1;       1;         0;
TCHECKBOX     ; RW          ;         20;     1;    0;    1;       1;         0;
//TCHECKBOX     ; BIT_2_5     ;         21;     1;    0;    1;       1;         0;
//TCHECKBOX     ; BIT_2_6     ;         22;     1;    0;    1;       1;         0;
//TCHECKBOX     ; BIT_2_7     ;         23;     1;    0;    1;       1;         0;
//TCHECKBOX     ; BIT_3_0     ;         24;     1;    0;    1;       1;         0;
//TCHECKBOX     ; BIT_3_1     ;         25;     1;    0;    1;       1;         0;
//TCHECKBOX     ; BIT_3_2     ;         26;     1;    0;    1;       1;         0;
//TCHECKBOX     ; BIT_3_3     ;         27;     1;    0;    1;       1;         0;
TCHECKBOX     ; AGV_REQ     ;         28;     1;    0;    1;       1;         0;
//TCHECKBOX     ; BIT_3_5     ;         29;     1;    0;    1;       1;         0;
//TCHECKBOX     ; BIT_3_6     ;         30;     1;    0;    1;       1;         0;
TVRWHEEL      ; SETPNT_PMP  ;         40;     8;    0;  255;       1;         0;
TVRWHEEL      ; EVP1        ;         56;     8;    0;  255;       1;         0;
TVRWHEEL      ; TARG_SPEED  ;         0;      16;    0;  1000;      1;        0;

[END]

[PDO2TX]
ID; 0536
TRANSMIT      ; 38   //ms
//componente  ; label       ;  start bit;  size;  min;  max;resolution;offset;
STUFFING_BIT  ; STUFFING    ;         15;     1;    0;    1;       1;        0;
TCHECKBOX     ; AGV_REG     ;         14;     1;    0;    1;       1;        0;
TCHECKBOX     ; BIT_2_0     ;         16;     1;    0;    1;       1;        0;
TCHECKBOX     ; BIT_2_1     ;         17;     1;    0;    1;       1;        0;
TCHECKBOX     ; BIT_2_2     ;         18;     1;    0;    1;       1;        0;
TCHECKBOX     ; BIT_2_3     ;         19;     1;    0;    1;       1;        0;
TCHECKBOX     ; BIT_2_4     ;         20;     1;    0;    1;       1;        0;
//TCHECKBOX     ; BIT_2_5     ;         21;     1;    0;    1;       1;        0;
//TCHECKBOX     ; BIT_2_6     ;         22;     1;    0;    1;       1;        0;
//TCHECKBOX     ; BIT_2_7     ;         23;     1;    0;    1;       1;        0;
TVRWHEEL      ; STEER_ANGLE ;         24;     16;    -255;  255;      1;        0;
[END]
```