

USER MANUAL
USB MODULE
8 INPUTS + 6 OUTPUTS + INCREMENTAL ENCODEUR
TC/USB-16IO



Document rev. : 15
Module rev.: 01

3C Concept S.A – 17 rue d'Ahuy, F-21121 HAUTEVILLE LES DIJON
Tel : 0380.550.130 - Fax : 0380.556.449
www.3cconcept.fr

Summary

| | |
|--|-----------|
| OVERVIEW..... | 3 |
| SPECIFICATIONS | 3 |
| ELECTRICAL SPECIFICATIONS..... | 4 |
| PINOUT | 6 |
| INPUTS / OUTPUTS WIRING | 8 |
| DRIVER INSTALLATION UNDER WINDOWS XP | 9 |
| DRIVER INSTALLATION UNDER WINDOWS 2000..... | 13 |
| TROUBLESHOOTING..... | 17 |
| DEMONSTRATION SOFTWARE | 18 |
| USING DLL TCUSB16IO.DLL..... | 20 |
| USING 64BITS DLL: TCUSB16IOX64.DLL | 21 |
| DLL FUNCTIONS DESCRIPTION..... | 22 |

The TC/USB-16IO provides a powerful way to add digital I/O to a computer.

This module add to your computer 8 opto-isolated digital inputs, 2 relays outputs, 4 opto-isolated digital outputs and 3 incremental encoder inputs.

Powered by USB bus, this module give to you a cost effective large variety of inputs/outputs.

A watchdog timer is incorporated as a failsafe feature if a particular sequence is not executed every 2 seconds in order to disable all outputs.

Specifications

- 8 opto-isolated digital inputs, 12 to 24 Vdc 10 mA Max with events counter on each input (8 bits counter) and possibility of latch of the position counter on the two last
- Lowpass filter 100 Hz on inputs
- 2 relay outputs NO (48V 1A max)
- 4 opto-isolated digital outputs (transistors 48V 30mA max)
- 3 inputs for incremental encoder A/B/Z (5V or 12/24V by jumpers), this inputs can also be used as standard digital inputs.
- 32 bits position counter.
- 2 latches of the 32 bits position counter.
- Watchdog timer
- USB to I/O isolation : 1500Vrms
- USB 2.0 Compatible (12 Mbits/s)
- USB powered (150 mA max)
- DIN 35mm mounting
- Dimensions : (L) 100 x (I) 23 x (H) 120

Electrical specifications

Power :

| Name | Description | Min | Typ | Max | Units |
|------|-------------------------|-----|-----|-----|-------|
| Iusb | USB current consumption | 40 | | 150 | mA |

10-30V Inputs :

| Name | Description | Min | Typ | Max | Units |
|------|---------------------------------|-----|------|-----|-------|
| Rin | Load impedance | | 2200 | | ohms |
| Vmax | Maximal voltage | -30 | | 30 | volts |
| Ie | Input current for logic level 1 | 3 | 5 | 15 | mA |
| Vih | Input voltage for logic level 1 | 10 | | 30 | volts |
| Vil | Input voltage for logic level 0 | | 0 | 2 | volts |

5V encoder inputs:

| Name | Description | Min | Typ | Max | Units |
|------|---------------------------------|-----|-----|-----|-------|
| Rin | Load impedance | | 470 | | ohms |
| Vmax | Maximal voltage | -7 | | 12 | volts |
| Ie | Input current for logic level 1 | 3 | | 15 | mA |
| Vih | Input voltage for logic level 1 | 3 | 5 | 12 | volts |
| Vil | Input voltage for logic level 0 | | 0 | 1 | volts |

Relay outputs :

| Name | Description | Min | Typ | Max | Units |
|------|-------------------------|-----|-----|------|-------|
| Vmax | Maximal voltage (ac/dc) | | | 48 | volts |
| Ismx | Maximal current | | | 1000 | mA |

Opto-coupler outputs:

| Name | Description | Min | Typ | Max | Units |
|------|-----------------|-----|-----|-----|-------|
| Vmax | Maximal voltage | | | 48 | volts |
| Ismx | Maximal current | | | 30 | mA |

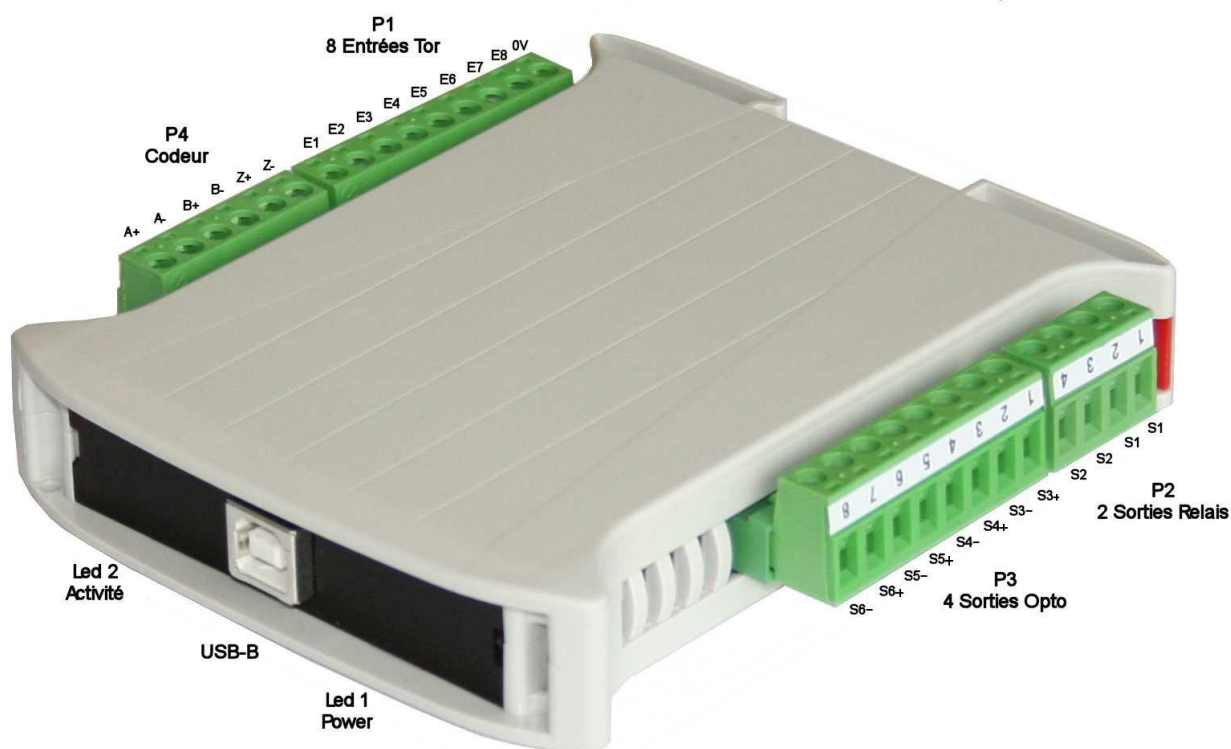
Dynamic characteristics:

| Name | Description | Min | Typ | Max | Units |
|------|------------------------------------|-----|-----------------|-----|-------|
| Tre | Inputs response time | | 1 | 10 | msec |
| Trsr | Relay outputs response time | | 6 | 12 | msec |
| Nbcy | Relay mechanical life | | 10 ⁶ | | Oper. |
| Trso | Output response time (Transistors) | | 100 | 300 | µsec |
| Twdg | Watchdog timebase | | 2 | | Sec |

* : resistive loads

USB Specifications:

- Peripherals max = 127
- Maximal length between peripherals = 5 m
- Possibility to use USB Hub
- Possibility to use extenders (ex. extremeUSB(R) www.icron.com)
- Industrial Hub (4 or 7 ports) see hubport on www.digi.com



Leds description :

Led 1 : Power supply present,(on USB)

Led 2 : Flash for normal activity

P1 : Connector 9 ways - 8 digital inputs

| Pin | Signal |
|-----|--------------|
| 1 | E1 |
| 2 | E2 |
| 3 | E3 |
| 4 | E4 |
| 5 | E5 |
| 6 | E6 |
| 7 | E7 / Latch 1 |
| 8 | E8 / Latch 2 |
| 9 | 0V |

P2 : Connector 4 ways - 2 relay outputs

| Pin | Signal |
|-----|--------|
| 1 | REL S1 |
| 2 | REL S1 |
| 3 | REL S2 |
| 4 | REL S2 |

P3 : Connector 8 ways - 4 opto-couplers digital outputs

| Pin | Signal |
|-----|----------|
| 1 | OPTO S3+ |
| 2 | OPTO S3- |
| 3 | OPTO S4+ |
| 4 | OPTO S4- |
| 5 | OPTO S5+ |
| 6 | OPTO S5- |
| 7 | OPTO S6+ |
| 8 | OPTO S6- |

ⓘ Attention : Polarized outputs, wiring polarity must be respected (see next page).

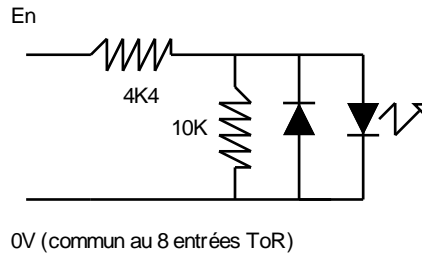
P4 : Connector 6 ways - Encoder inputs

| Pin | Signal |
|-----|--------|
| 1 | A+ |
| 2 | A- |
| 3 | B+ |
| 4 | B- |
| 5 | Z+ |
| 6 | Z- |

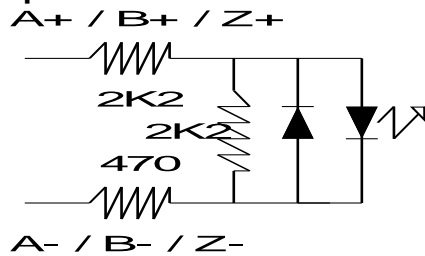
Encoder inputs configuration :

Place the 3 jumpers for 5V inputs (encoder outputs) or remove the 3 jumpers with 12/24V encoder outputs

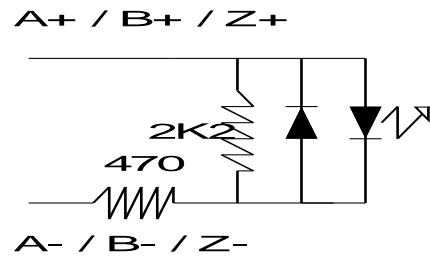
Logical inputs :



Encoder inputs :

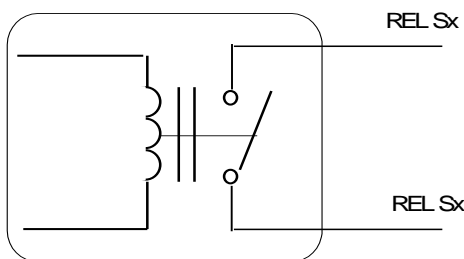


Moduleur sans cavalier (12/24V) Without jumpers (12/24V)

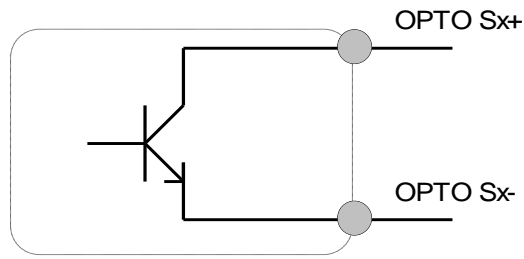


Moduleur avec cavalie With jumpers (5V)

Relay outputs :



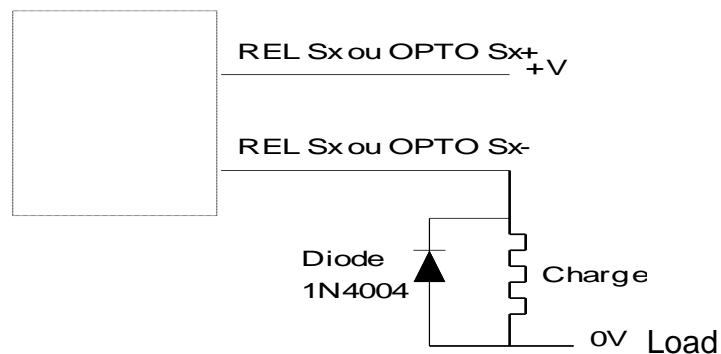
Opto-couplers outputs :



:

Output protection (relay and opto) :

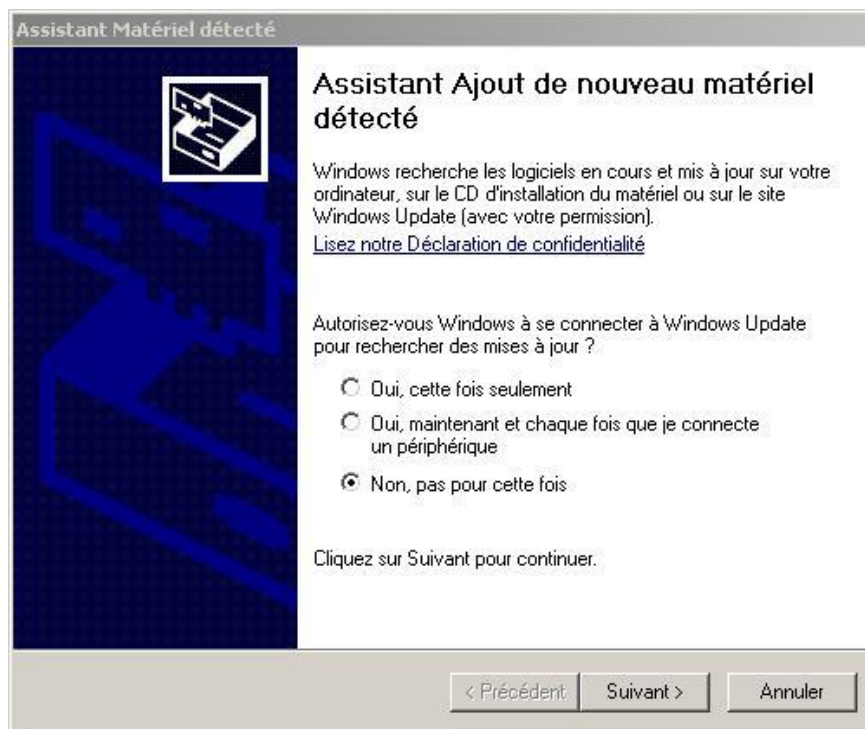
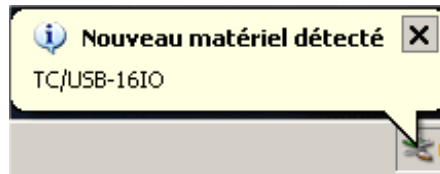
With inductive loads (relay, solenoid ...) a free wheel diode must be placed :



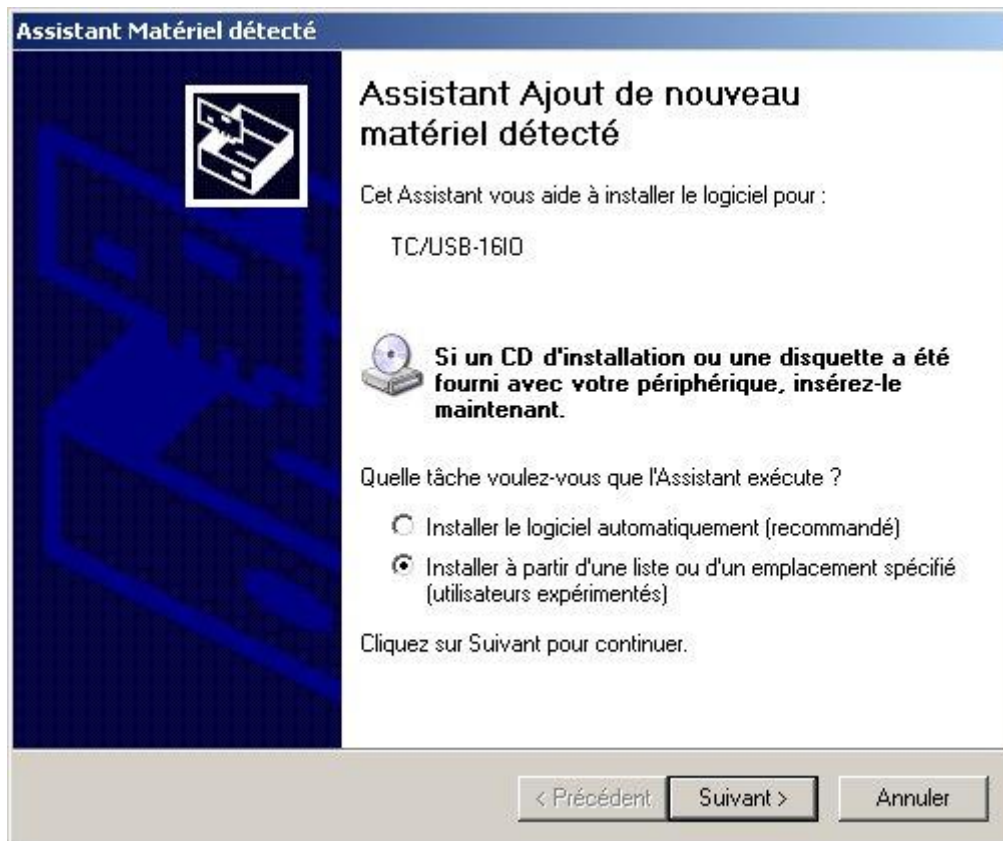
Driver installation under Windows XP

On CD-ROM you can find demonstration program, DLL (32 or 64bits) and include files to create your own programs.

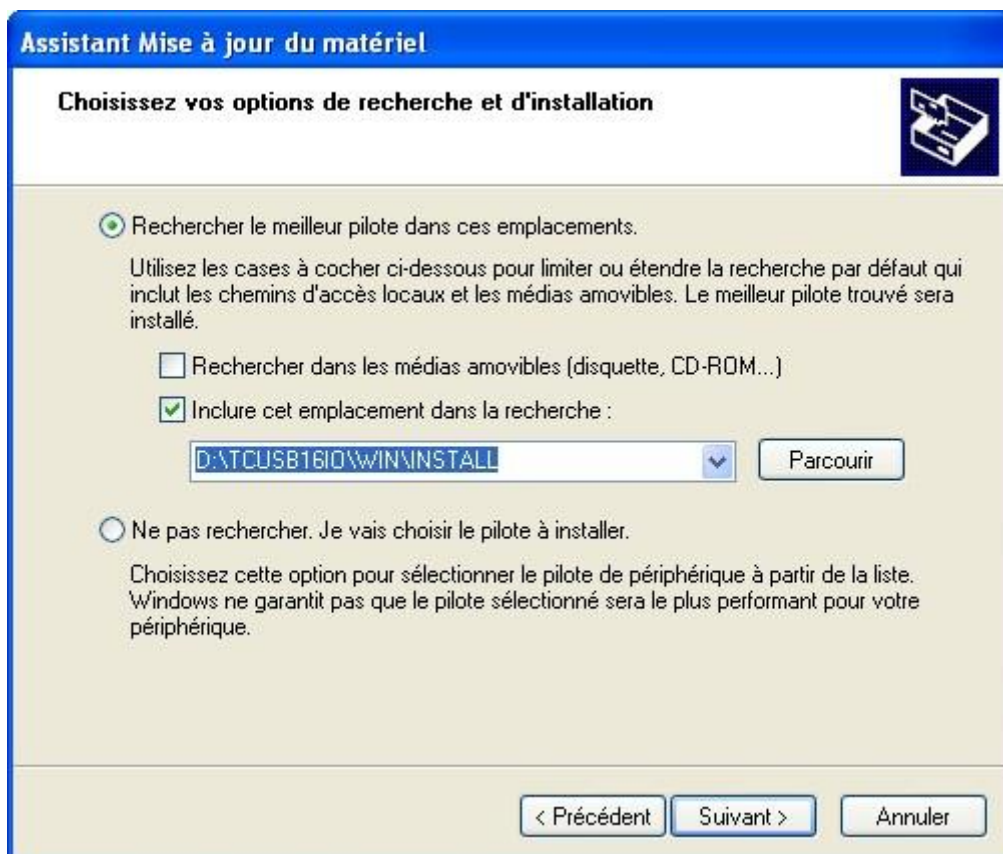
- Connect module to PC, Windows must detect it and show a message box :



- Windows want to connect to Windows Update, answer « not this time »
- Next select « install from a list.. »



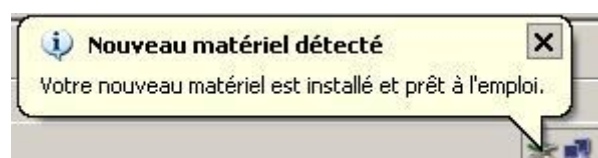
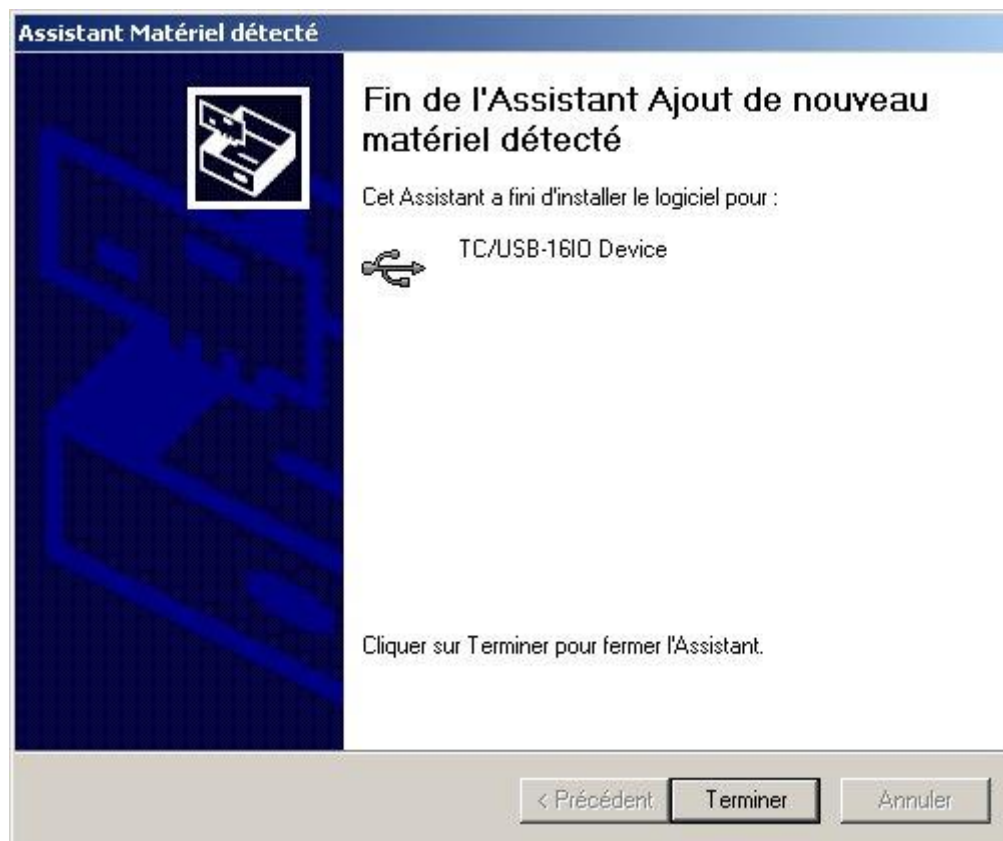
- In this window click on « Include this location » and browse to directory **TCUSB16IO\WIN\INSTALL** on CDROM



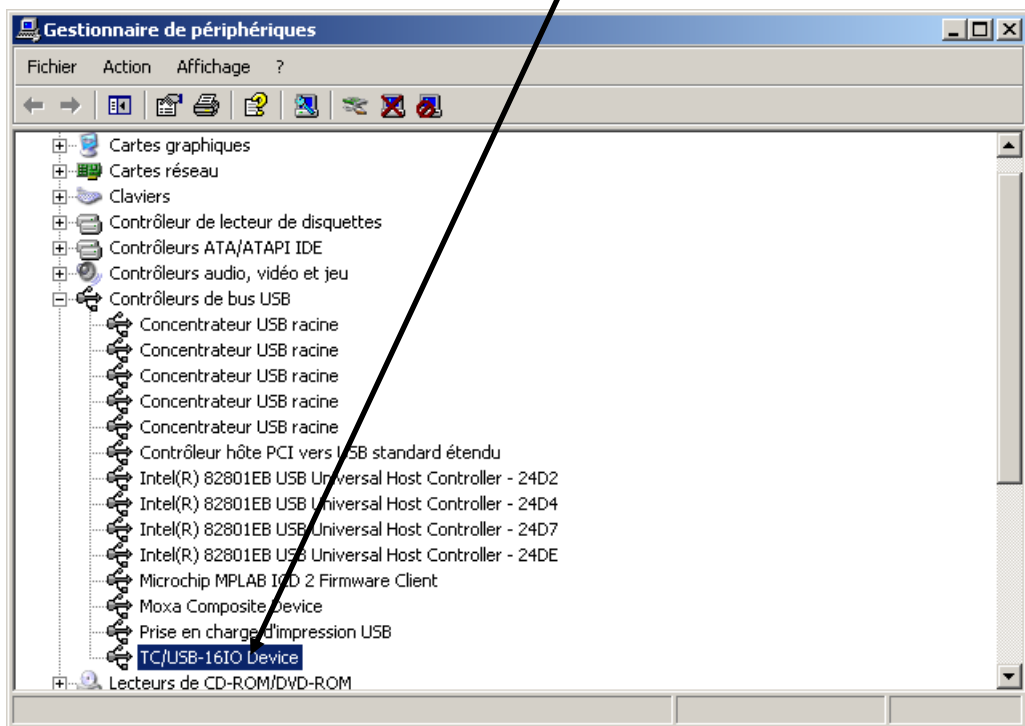
When installing, Windows say “This driver has not pass WHQL testing ...” answer “Continue anyway”



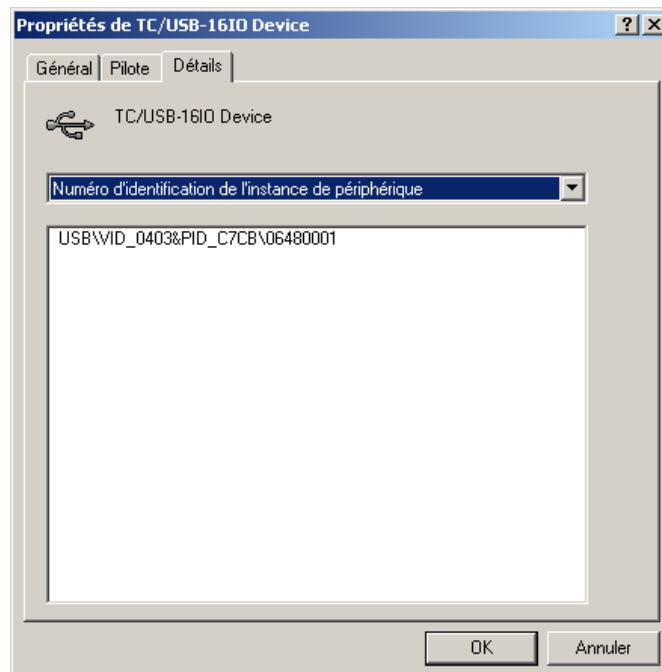
At end, Windows display the following message “Completing the found New hardware wizard”



You can verify in configuration panel that TCUSB-16IO device is installed:



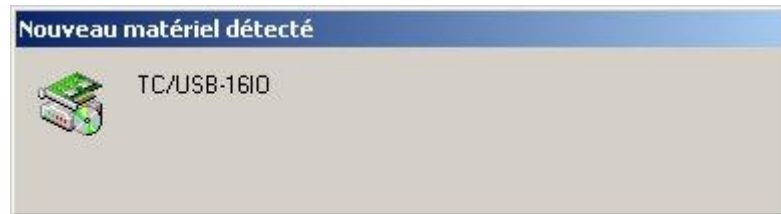
You can also read serial number on property / detail tab :



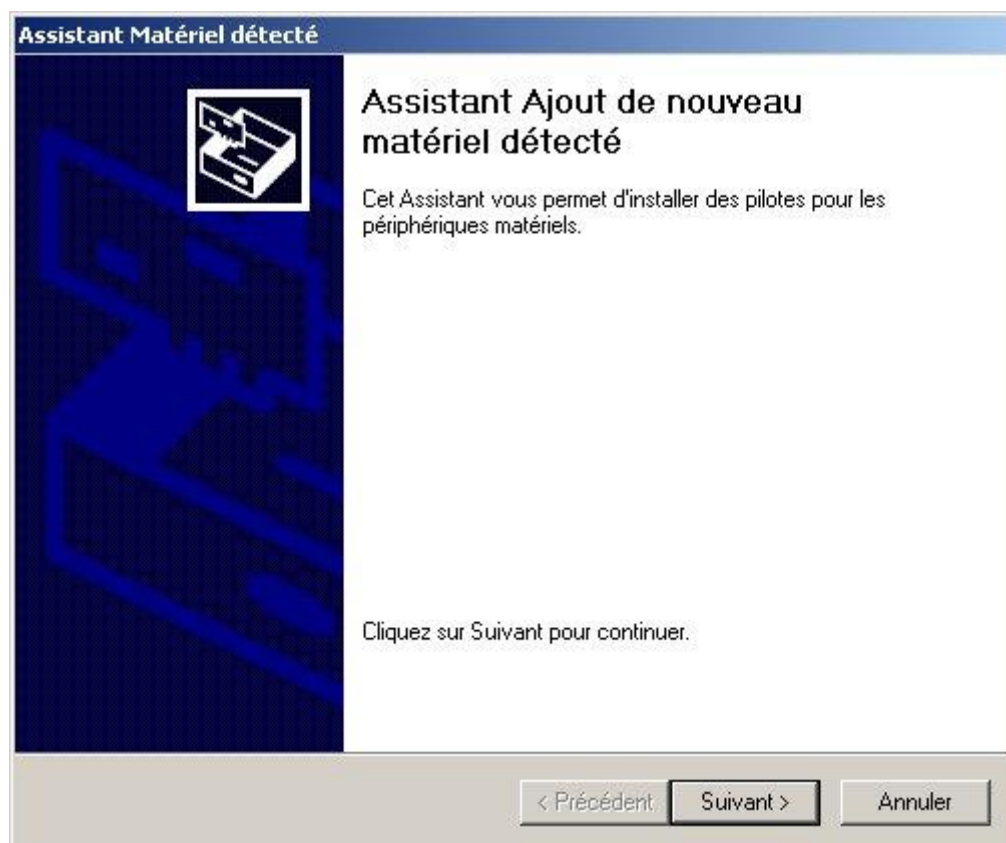
Then you can run program **TCUSB16IO\WIN\DEMO\DemoTCUSB16IO.EXE** in order to verify that module is working.

Driver installation under Windows 2000

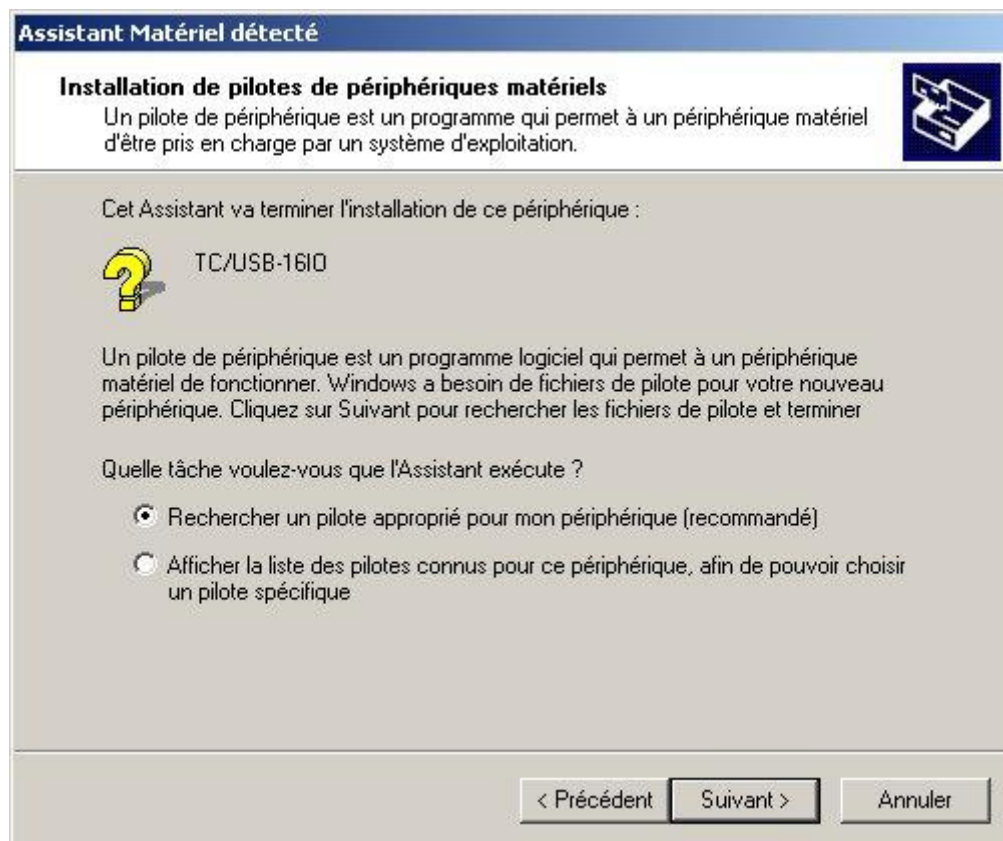
- Connect module to PC, Windows must detect it and show a message box :



- Click on Next



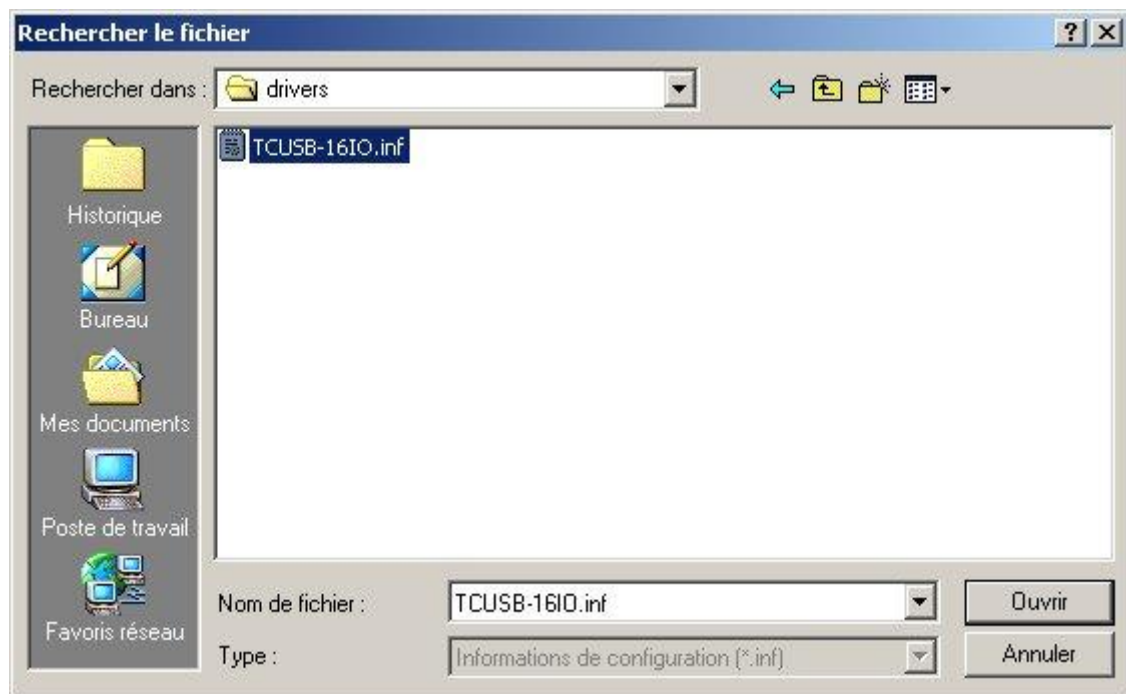
- In next screen select « Search for driver ...»



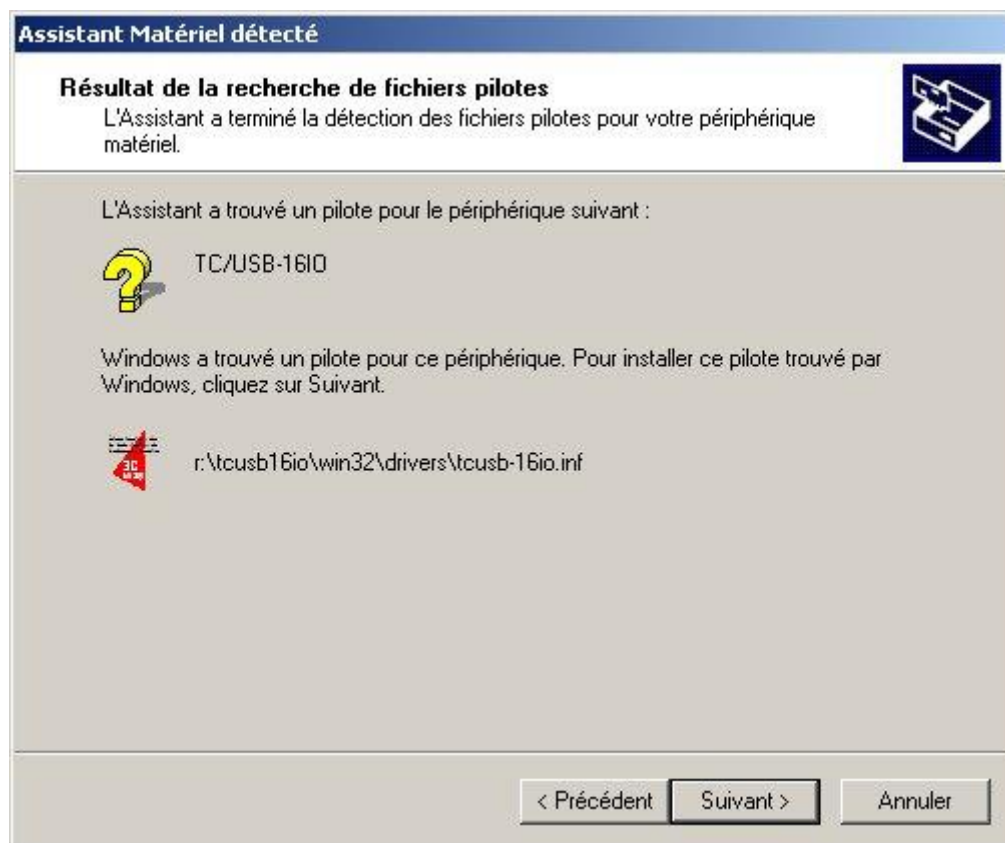
- Select specific location



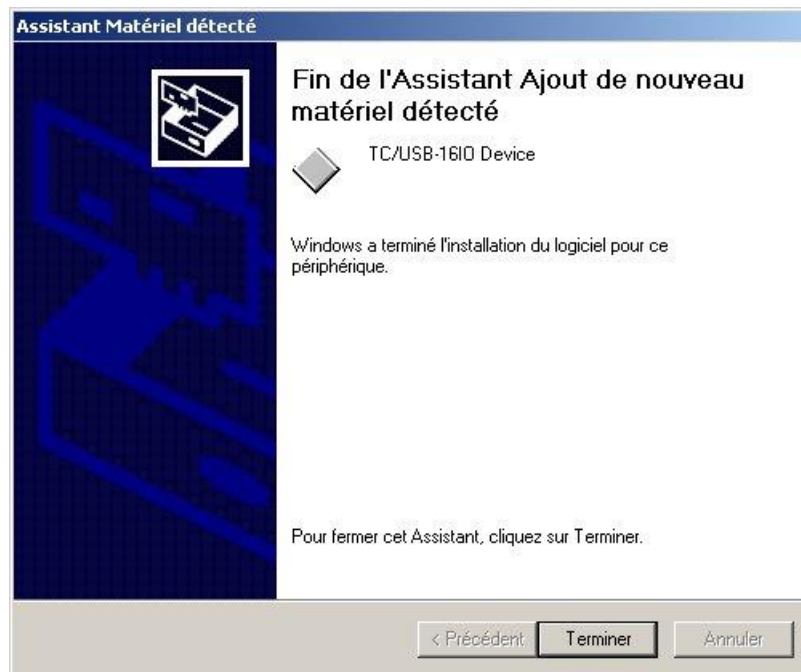
- Browse on CDROM to directory **TCUSB1610WIN\INSTALL\TCUSB.INF**, click on Open.



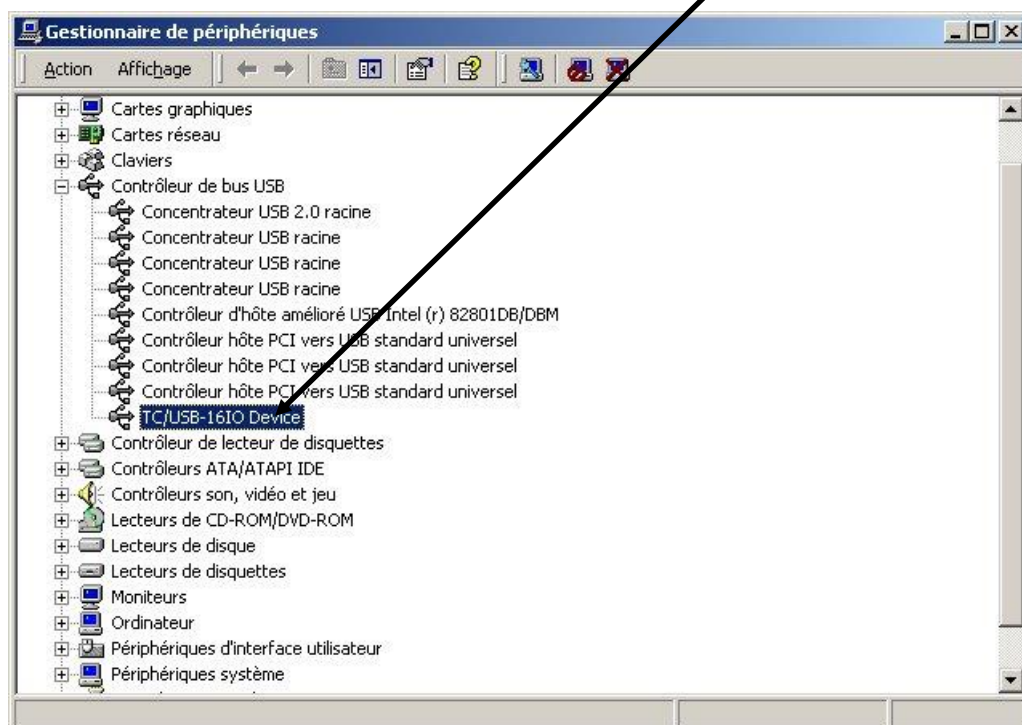
- Resume screen, click on « Next »



- Installation complete :



You can verify in configuration panel that TCUSB-16IO device is installed:



Then you can run program **TCUSB16IO\WIN\DEMO\DemoTCUSB16IO.EXE** in order to verify that module is working.

- Exclamation mark in Device manager :

An exclamation mark is display when driver is not correctly detect. Use properties tabs and update driver to restart driver installation as describe before.

- Interrupted installation :

Restart computer without module, and then connect module again. Restart installation as the first time.

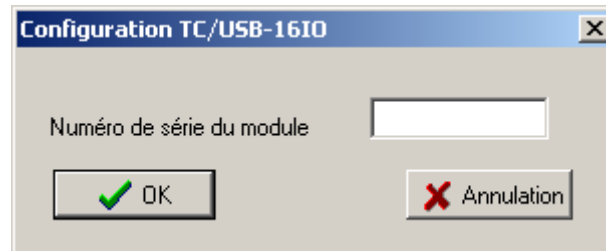
- Driver uninstallation:

To uninstall the module's driver, execute the batch file "Uninstall Driver TCUSB16IO.bat" on the cd-rom shipped with it.

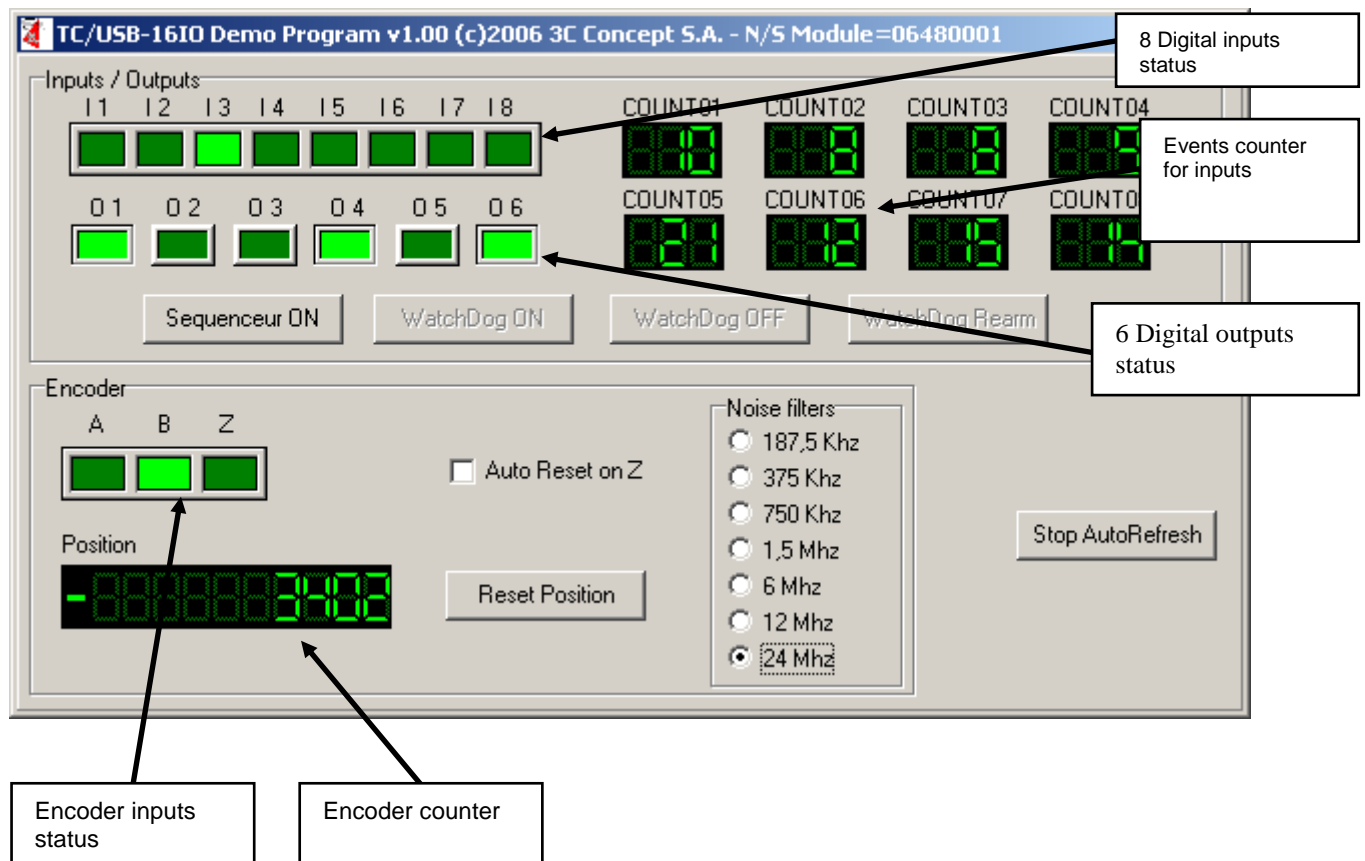
Before, you have to copy "**Uninstall Driver TCUSB16IO.bat**" et "**CDMUninstaller.exe**" from CD-ROM to HDD, in order to allow the creation of a log file for the uninstallation process.

Demonstration software

Demonstration software is on CD-ROM, TCUSB16IO\WIN\Demo\DemoTCUSB16IO.exe



Give module serial number (display on label under product) or enter OK without number if you have only one module attach to your computer.



All outputs can be drive with a click on green symbol.

Button “ Sequencur ON/OFF “ : Automatically drive outputs

Button “WatchDogOn” : Activate watchdog timer

Button “ WatchdogOff” : Stop watchdog timer

Button “ WatchDogRearm” : Manual watchdog update, must be done every 2 seconds.

Button “StopAutoRefresh” : Stop updating display

Button “Reset Position” : Clear encoder value

Using DLL TCUSB16IO.DLL

Windows 32 DLL can be use with different languages :

With PASCAL compiler (DELPHI):

Use file TCUSB16IO.INC and DLL TCUSB16IO.DLL

With C compiler:

Use file TCUSB16IO.H and DLL TCUSB16IO.DLL

With Basic compiler:

Use file TCUSB16IO.BAS and DLL TCUSB16IO.DLL

With C#:

Use file TCUSB16IO.CS and DLL TCUSB16IO.DLL

Shared object under Linux:

Look at directory « linux » on CDROM, textfile explain how to use it (TCUSB16IO\Linux\)

ⓘ Warning : All functions works on memory image update by function TCUSB16IO_Refresh. This function must be call before every function using inputs AND after every function using outputs. Do not use this function after every commands because of the transmission delay on USB bus.

An information window can appear on startup if driver is not the latest (3 seconds display). In this case, please download the new release and update your driver.

Using 64bits DLL: TCUSB16IOx64.DLL

TCUSB16IO- 64bits DLL can be used with different compilers:

C++ compiler :

Use files: TCUSB16IOx64.H, TCUSB16IOx64.LIB and DLL TCUSB16IOx64.DLL

There is one example program shipped with CD-Rom, made with following IDE :

Microsoft Visual Studio 2010 C++

TCUSB16IO 64bits DLL no longer check its version on the web (in order to keep an updated driver).
Only TCUSB16IO_**OpenN** function has been implemented ,
TCUSB16IO_**Open** has been deprecated.
Final point, calling convention in use in these 64bits DLL is "CDECL".

DLL Functions description

TCUSB16IO_OpenN(Nmodule, NumSerie)

Parameter : Nmodule: 32 bits signed integer (to 8).

Numserie : Pointer on unsigned Dword (32 bits) initialized with serial number.

Return : 32 bits signed integer with error code.

Initialize driver, assign module number to serial number send in parameter. If serial number is set to 0, variable is update with real serial number (only if one module is attached to computer).

Use this function before all other one.

Close session with TCUSB16IO_Close .

TCUSB16IO_Open(Nmodule, NumSerie)

Parameter : Nmodule: 32 bits signed integer.

Numserie : Pointer on string initialized with serial number (Text)

Return : 32 bits signed integer with error code.

Prefer use of TCUSB16IO_OpenN function (due to string configuration in many languages)

Initialize driver, assign module number (1 to 8) to serial number send in parameter. If serial number string is empty, variable is update with real serial number (only if one module is attached to computer).

Use this function before all other one.

Close session with TCUSB16IO_Close .

TCUSB16IO_Close(NModule)

Parameter : Nmodule: 32 bits signed integer.

Return : 32 bits signed integer with error code.

Close session and close driver (necessary at the end of program)

TCUSB16IO_Refresh(NModule)

Parameter : Nmodule: 32 bits signed integer.

Return : 32 bits signed integer with error code.

Update data on specified module. This function must be call BEFORE using inputs data and AFTER modifying outputs data.

TCUSB16IO_Input(Nmodule , Data) ⁽¹⁾

Parameter : Nmodule: 32 bits signed integer.

Data : Pointer on 32 bits unsigned Dword.

Return : 32 bits signed integer with error code.

Read digital inputs status (8 standard inputs and 3 encoders inputs)

| Input Variable | Bit10 | Bit9 | Bit 8 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-----------------------|-------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| State of : | Z | B | A | E8 | E7 | E6 | E5 | E4 | E3 | E2 | E1 |

TCUSB16IO_Output(Nmodule, Data) ⁽¹⁾

Parameter : Nmodule: 32 bits signed integer.

Data : Unsigned Byte 8 bits.

Return : 32 bits signed integer with error code.

Send on outputs value from variable Data.

TCUSB16IO_BitInput(Nmodule , Entree) ⁽¹⁾

Parameter : Nmodule: 32 bits signed integer (1 to 8).

Entree : 32 bits signed integer (1 to 11).

Return : 32 bits signed integer with 0 or 1 value according to designed input status

Read status of specified input / module

| Entree Variable | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|-----------------|----|----|----|----|----|----|----|----|---|----|----|
| State of : | E1 | E2 | E3 | E4 | E5 | E6 | E7 | E8 | A | B | Z |

TCUSB16IO_BitOutput(NModule,Sortie,etat) ⁽¹⁾

Parameter : Nmodule: 32 bits signed integer (1 to 8).

Sortie : 32 bits signed integer (1 to 6).

etat : 32 bits signed integer (0 or 1).

Return : None (Procedure)

Force ouput « Sortie » to state « etat » (0 or 1)

TCUSB16IO_BitReadBack(NModule,Sortie) ⁽¹⁾

Parameter : Nmodule: 32 bits signed integer.

Sortie : 32 bits signed integer (1 to 6).

Return : Signed integer 32 bits, 0 or 1 according to output status

Read output status (0 or 1)

TCUSB16IO_Counter(NModule,Entree,Data) ⁽¹⁾

Parameter : Nmodule: 32 bits signed integer.

Entree : 32 bits signed integer.

Data : Pointer on unsigned Byte (8 bits).

Return : 32 bits signed integer with error code.

Read events counter attached to specified input « Entree ».

TCUSB16IO_Config(Nmodule,Data) ⁽¹⁾

Parameter : Nmodule: 32 bits signed integer.

Data : unsigned Byte (8 bits).

Return : 32 bits signed integer with error code.

Module encoder configuration byte :

| Data | Fmax | Ref. mark |
|------|--------|-----------|
| 0 | 8Mhz | Disable |
| 1 | 4Mhz | Disable |
| 2 | 2Mhz | Disable |
| 3 | 500Khz | Disable |

| | | |
|-----|----------|----------|
| 4 | 250Khz | Disable |
| 5 | 125Khz | Disable |
| 6 | 63Khz | Disable |
| 7 | Not used | Not used |
| 8 | 8Mhz | Enable |
| 9 | 4Mhz | Enable |
| ... | ... | ... |

Encoder Frequency filter is set on bits 1 to 3, reference mark is used only if bit 4 is set (Z input can clear position counter)

TCUSB16IO_Pos(Nmodule,Data) ⁽¹⁾

Parameter : Nmodule: 32 bits signed integer.

Data : Pointer on 32 bits unsigned Dword.

Return : 32 bits signed integer with error code.

Read encoder counter (position) on 32 bits

TCUSB16IO_ResetPos(Nmodule)

Parameter : Nmodule: 32 bits signed integer.

Return : 32 bits signed integer with error code.

Clear encoder counter (position) and latches.

TCUSB16IO_ConfigLatch(Nmodule,Data)

Parameter : Nmodule : 32 bits signed integer.

Data : unsigned Byte (8 bits).

Return : 32 bits signed integer with error code.

Latch encoder configuration byte :

- 1) On the bits 1 to 3 of data, the latch 1 mode for the encoder position possible on input 7 (form 0 to 3 with 0 : No latch, 1 : Latch on the rising edge of Input 7, 2 : Latch on the falling edge of input 7, 3 : Latch on the change of state of input 7)
- 2) On the bits 5 to 7 of data, the latch 2 mode for the encoder position possible on input 8 (form 0 to 3 with 0 : No latch, 1 : Latch on the rising edge of Input 8, 2 : Latch on the falling edge of input 8, 3 : Latch on the change of state of input 8)

TCUSB16IO_Latch(Nmodule,Latch,Data) ⁽¹⁾

Parameter : Nmodule : 32 bits signed integer.

Latch: unsigned Byte (8 bits).

Data : Pointer on 32 bits unsigned Dword.

Return : 32 bits signed integer with error code.

Read specified latch value of the encoder counter (position) on 32 bits

TCUSB16IO_WdgRun(Nmodule)

Parameter : Nmodule: 32 bits signed integer.

Return : None (Procedure)

Activated watchdog timer

TCUSB16IO_WdgStop(Nmodule)

Parameter : Nmodule: 32 bits signed integer.

Return : None (Procedure)

Inactivated watchdog timer

TCUSB16IO_WdgRearm(NModule)

Parameter : Nmodule: 32 bits signed integer.

Return : None (Procedure)

Update watchdog timer (must be call less than every 2 seconds)

TCUSB16IO_ResetPort(NModule)

Parameter : Nmodule: 32 bits signed integer.

Return : None (Procedure)

USB port restart, this can be use in case of communication trouble on an active module.

TCUSB16IO_CyclePort(NModule)

Parameter : Nmodule: 32 bits signed integer.

Return : None (Procedure)

Similar to unconnect and reconnect USB port to restart communication between computer and module after serious error.

TCUSB16IO_Version(NModule,Data1,Data2) ⁽¹⁾

Parameter : Nmodule: 32 bits signed integer.

Data1 : Pointer on unsigned Byte (8 bits).

Data2 : Pointer on unsigned Byte (8 bits).

Return : 32 bits signed integer with error code.

Read firmware release number and driver release number. Values are set in data1 and data2 variables.

(1) Look at TCUSB16IO_Refresh information at the beginning of this section.

Error list :

| Number | Designation |
|--------|----------------------------|
| 0 | No error |
| 1 | Bad serial number |
| 2 | Bad module number (1 to 8) |
| 3 | No TC/USB-16IO detected |
| 4 | Non-initialized module |
| 5 | Write error |
| 6 | Read error |
| 7 | Close error |