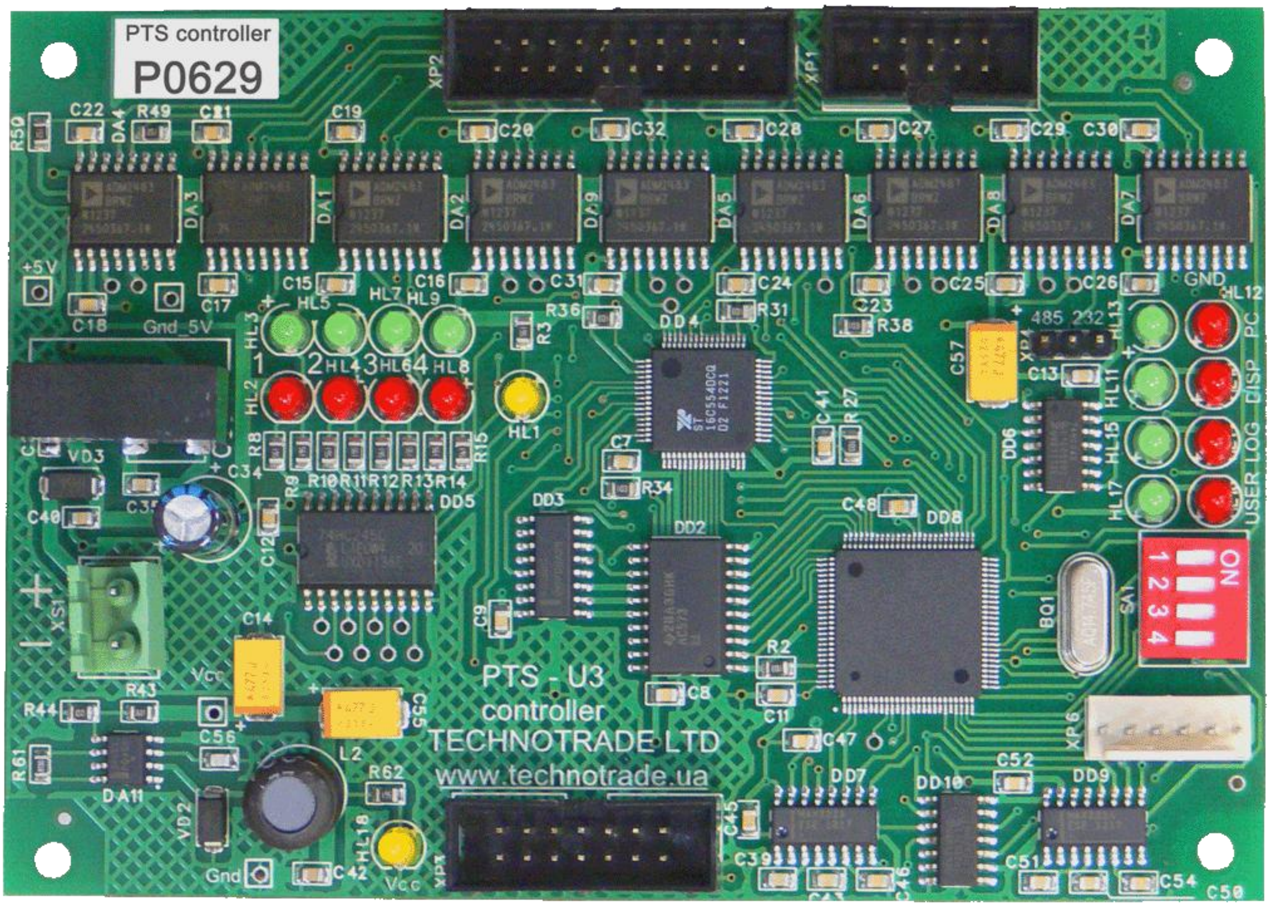
# PTS controller

## over fuel dispensers and ATG systems for petrol stations



# TECHNICAL GUIDE

(PTS controller PCB board modification: PTS-U3)

*Review date: 29 July, 2013*

**TECHNOTRADE LTD**

# CONTENT

# REVISION HISTORY

| REV | DATE | BY | SECTION | DESCRIPTION |
|---|---|---|---|---|
| **R01** | 12.04.2013 | Eugene Vasylyev | All | First release of PTS-U3 board revision |
| **R02** | 29.07.2013 | Evgeniy Vasyliev | UniPump pumps software simulator<br>UniProbe ATG probes software simulator | Updated UniPump pumps software simulator<br>Updated UniProbe ATG probes software simulator |

# PURPOSE OF THE DOCUMENT

This Technical Guide is intended for studying of PTS controller over fuel dispensers and ATG systems for petrol stations. It contains basic information regarding its

- technical characteristics
- supported communication protocols of fuel dispensers and ATG systems
- board interfaces and connectors
- configuration
- supplied application programming interfaces (API)
- description, configuration and connection of PTS controller software development kit (SDK)
- schemes of connection to fuel dispensers and ATG systems
- board schematics
- cabling

Information regarding connection to specific fuel dispensers and correspondent configuration of PTS controller can be received upon request to TECHNOTRADE LTD company.

Given technical manual describes the latest hardware version of the PTS controller board (PTS controller PCB board modification PTS-U3), older versions of the PTS controller (PTS-U2 and PTS-U) and their corresponding connection schemes are found in the end of this document.

Due to a reason that PTS controller firmware is constantly being developed in direction of improvement of its possibilities, changes are possible in final version, which are not described in given Technical Guide.

During the system development process given Technical Guide is also expanded and updated and new chapters are added. Latest version of this Technical Guide can be downloaded from the PTS controller web-page: http://www.technotrade.ua/fuel_pump_controller.html.

TECHNOTRADE LTD hereby permits reproduction of this document as may be required by any of the customers or OEMs wishing to use it.

This document has been carefully prepared and is believed to be accurate. However TECHNOTRADE LTD, its employees and its agents do not assume responsibility for its use either directly or indirectly. TECHNOTRADE LTD shall not be liable for technical or editorial errors or omissions which may appear in this document. TECHNOTRADE LTD reserves a right to make changes to this document at any time without notice. Prospective users of this document should contact TECHNOTRADE LTD at the time they wish to use PTS controller together with their products to become aware of any updates that may apply.

In case if you find any mistakes, omissions in this document or have any suggestions on improvements to this document, please feel free to e-mail them to our support mailbox: support_1a@technotrade.ua. We will be grateful to you for this valuable information.

All technical questions regarding the PTS controller are welcome to be asked on support mailbox: support_1a@technotrade.ua. Our support team will be glad to help you.

Also you can call to us or visit us on:

## TECHNOTRADE LTD

Ukraine, 04114 Kiev, Polupanova str. 10, office 1
Tel: +38-044-502-46-55, +38-044-502-46-77
Web: www.technotrade.ua
Mail: mail@technotrade.ua

# TECHNICAL FEATURES

## *Appointment*

**PTS controller over fuel dispensers and ATG systems for petrol stations** is a specialized controller intended to be used in connection with third party control system for petrol station (POS system, cash register, OPT terminal, etc) to provide simultaneous remote control over petrol, diesel, CNG and LPG dispensers and automatic tank gauge (ATG) systems installed at petrol, CNG and LPG stations and storage depots.

PTS controller provides control using the single common communication protocol. PTS controller provides conversion of the common communication protocol into various proprietary communication protocols of fuel dispensers and ATG systems.

PTS controller is supplied with a rich set of API (application programming interfaces) developed under most popular programming languages and environments in order to make comfortable and quick implementation of PTS controller into control systems for petrol stations (POS system, cash register, OPT terminal, etc).

## *Specification*

| PARAMETER | VALUE |
|---|---|
| **Power supply voltage** | 12 V DC |
| **Current consumption** | 200 mA max |
| **Temperature range** | -40°C ÷ +80°C |
| **Weight** | 120 g |
| **Overall dimensions** | 120 x 85 x 30 mm |

# SUPPORTED FUEL DISPENSERS COMMUNICATION PROTOCOLS

*Baud rate is configured for pump channels in PTS controller independently on the used communication protocol.*

*Some dispensers may demand using interface converter boards to RS-485 interface (depends on electronics of used pumphead in dispenser). Read more information on interface converters on: http://www.technotrade.ua/dispenser_interface_converters.html*

| ## | FUEL DISPENSER BRAND | PROTOCOL NAME |
|---|---|---|
| 1 | GILBARCO | GILBARCO Two-Wire |
| 2 | WAYNE DRESSER | WAYNE Dart |
| 3 | WAYNE DRESSER | Wayne US Current Loop |
| 4 | WAYNE PIGNONE | WAYNE Dart |
| 5 | NUOVO PIGNONE | Nuovo Pignone |
| 6 | TOKHEIM | Tokheim Controller-Dispenser Communication protocol |
| 7 | TATSUNO (JAPAN) | TATSUNO SS-LAN |
| 8 | TATSUNO EUROPE (FORMER BENC) | TATSUNO Benc PDE |
| 9 | LOGITRON | Marconi PumaLAN |
| 10 | EUROPUMP | S4-Dart |
| 11 | MEPSAN | S4-Dart |
| 12 | MEKSAN / WAYNE SU86 | WAYNE Dart |
| 13 | MEKSER | S4-Dart |
| 14 | 2A | S4-Dart |
| 15 | PETROLMECCANICA | WAYNE Dart |
| 16 | TOKICO | Tokico SS-LAN |
| 17 | TOMINAGA | Tominaga SS-LAN |
| 18 | SALZKOTTEN | GILBARCO Two-Wire |
| 19 | ADAST (ADAMOV SYSTEMS) | ADAST EasyCall |
| 20 | PEC (GALLAGHER FUEL SYSTEMS) | PEC Pump Communication Protocol |
| 21 | PETROTEC | GILBARCO Two-Wire |
| 22 | ZAP / MM PETRO | MM PETRO ZAP RS-485 |
| 23 | BENNETT | BENNETT pump dispenser protocol (current loop) |
| 24 | BENNETT | BENNETT pump dispenser protocol (RS-485) |
| 25 | EMGAZ DRAGON | Marconi PumaLAN |
| 26 | KALVACHA | Kalvacha |
| 27 | GALILEO | PumpControl GC21 |
| 28 | KRAUS | MNET design specification |
| 29 | BATCHEN | GILBARCO Electroline |
| 30 | BATCHEN | GILBARCO MPP |
| 31 | PROWALCO | SPDC-1, MPDC-1 |
| 32 | BARANSAY | GILBARCO Two-Wire |
| 33 | FALCON LPG | GILBARCO Two-Wire |
| 34 | KOREA ENE | EnE Dispenser POS protocol |
| 35 | LG ENE | EnE Dispenser POS protocol |
| 36 | SOMO PETRO | POS protocol |
| 37 | HONG YANG | HongYang 886 communication protocol |

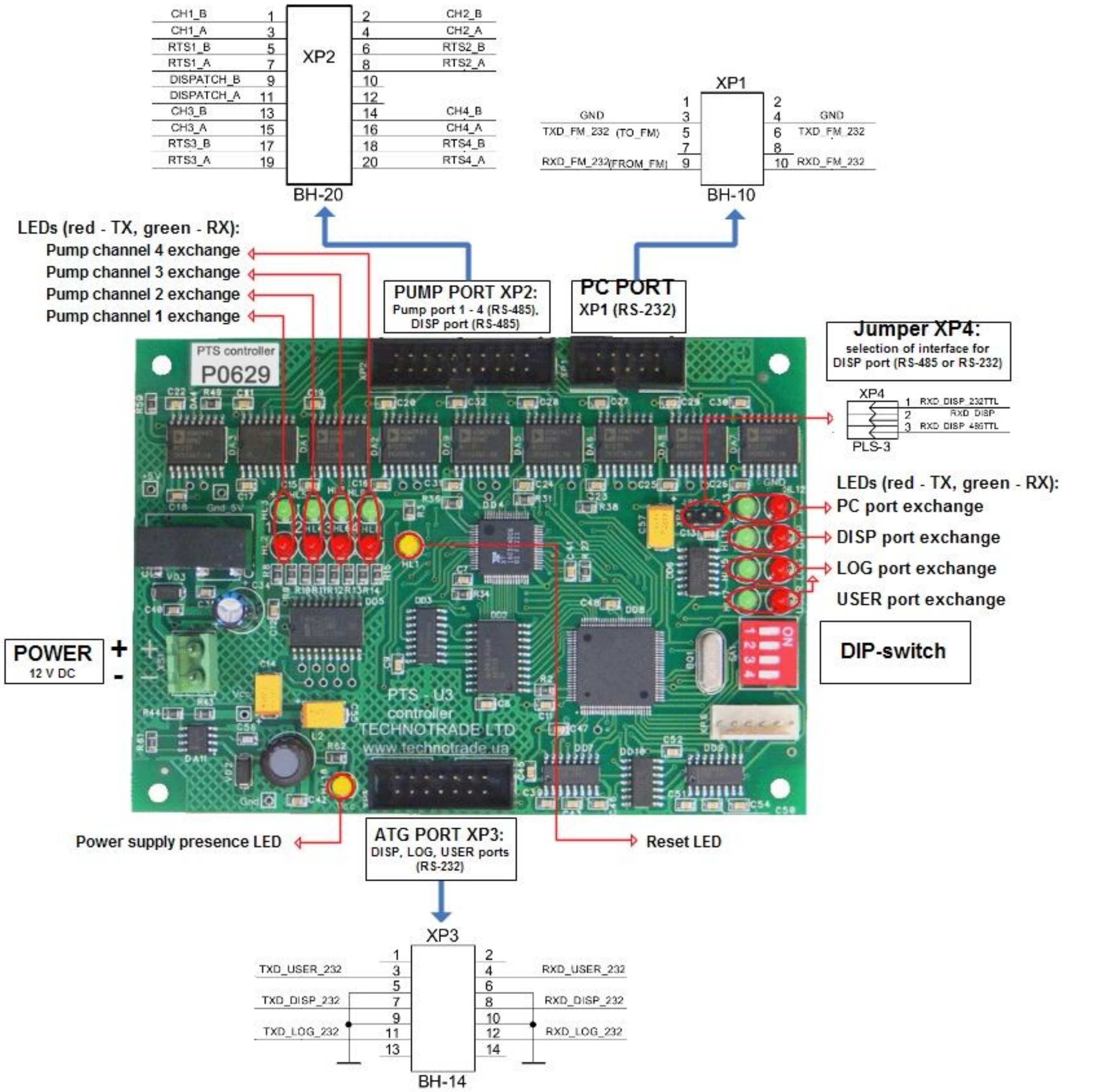
| | | |
|---|---|---|
| **38** | **HONG YANG** | HongYang FZ-protocol |
| **39** | **LANFENG** | RS–232 Protocol of Dispenser |
| **40** | **SANKI** | Sanki communication protocol |
| **41** | **BLUE SKY** | Blue Sky |
| **42** | **REAL-TECH** | Blue Sky |
| **43** | **SEA BIRD** | Blue Sky |
| **44** | **DONG HWA PRIME** | Prime pump interface |
| **45** | **EPCO** | EPCO protocol specification |
| **46** | **GREENFIELD** | GILBARCO Two-wire |
| **47** | **SAFE** | SAFE Graf |
| **48** | **DEVELCO** | DEVELCO |
| **49** | **PUMP CONTROL** | PumpControl GC21 |
| **50** | **AGIRA** | PumpControl GC21 |
| **51** | **ASPRO** | PumpControl GC21 |
| **52** | **IMW** | PumpControl GC21 |
| **53** | **IMW** | Tokheim UDC |
| **54** | **SLAVUTICH** | SLAVUTICH FD-Link |
| **55** | **SHELF** | SHELF |
| **56** | **UNICON-TIT** | TIT UniPump |
| **57** | **KPG-2** | TIT UniGaz |
| **58** | **KIEVNIIGAZ** | TIT UniGaz |

# SUPPORTED ATG SYSTEMS COMMUNICATION PROTOCOLS

*Communication parameters (baud rate, parity control, data and stop bits) are configured for ATG channels in PTS controller independently on the used communication protocol.*

| ## | ATG SYSTEM BRAND | PROTOCOL NAME |
|---|---|---|
| **1** | **GILBARCO Veeder Root**<br>**(TLS-2, TLS-300, TLS-350, TLS-450)** | GILBARCO Veeder Root |
| **2** | **OPW** | GILBARCO Veeder Root |
| **3** | **INCON** | GILBARCO Veeder Root |
| **4** | **LABKO** | GILBARCO Veeder Root |
| **5** | **OMNTEC** | GILBARCO Veeder Root |
| **6** | **ENRAF** | ENRAF Height-, Volume-protocol |
| **7** | **START ITALIANA** | START ITALIANA SMT/XMT |
| **8** | **PETRO VEND** | PETROVEND4 |
| **9** | **STRUNA** | STRUNA Kedr spec. 1.4 |
| **10** | **FAFNIR** | FAFNIR VISY-Quick |
| **11** | **FAFNIR** | Gilbarco Veeder Root |
| **12** | **ASSYTECH** | Assytech |
| **13** | **HECTRONIC** | HECTRONIC HLS |
| **14** | **MTS ATG SENSORS** | MTS USTD |
| **15** | **UNIPROBE** | UNIPROBE |

# BOARD CONNECTORS AND INTERFACES



***NOTE!*** Jumper XP4 serves for selection of DISP channel interface between RS-485 and RS-232.

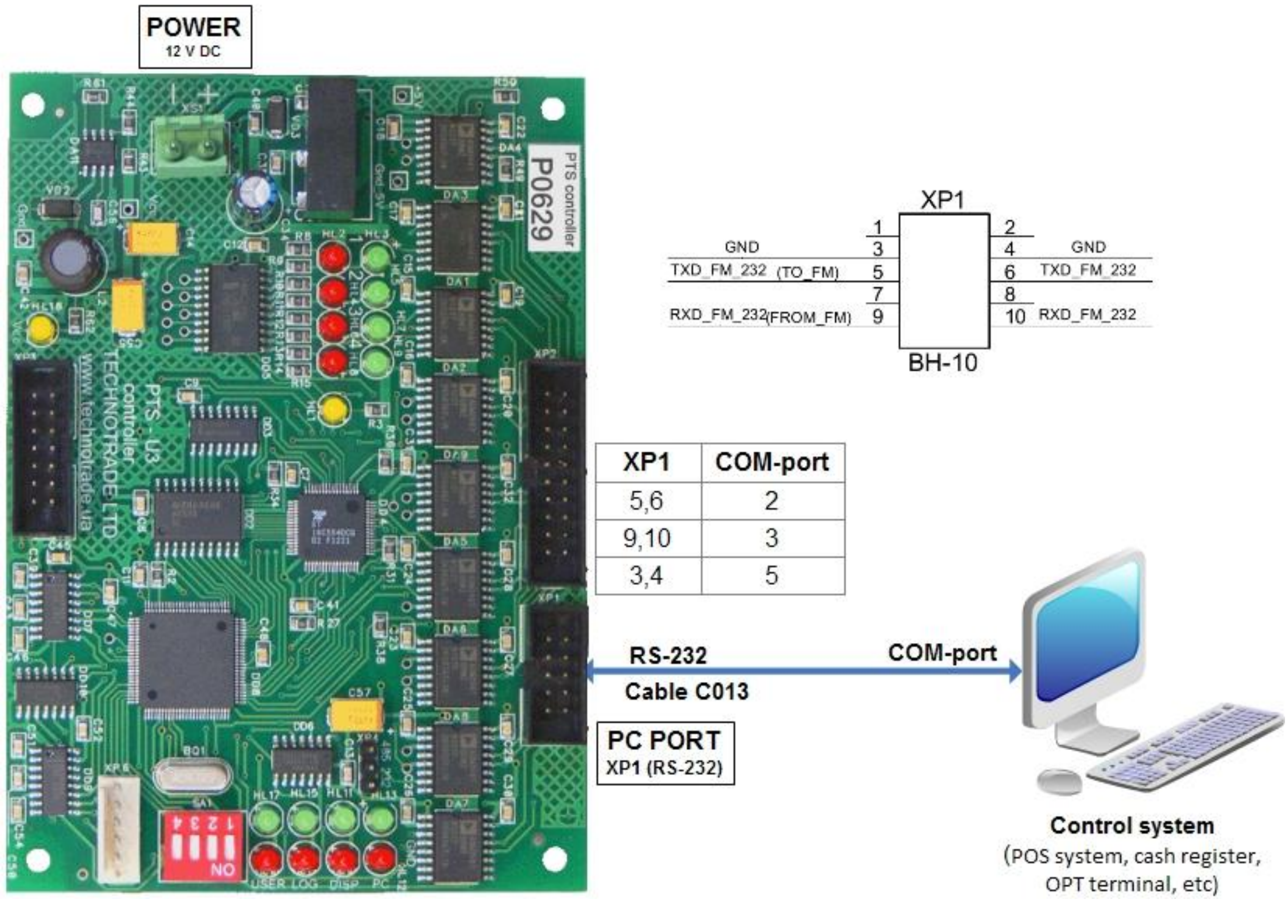
# COMMUNICATION PORTS

| PORT NAME | | INTERFACE | PURPOSE |
|---|---|---|---|
| **PC PORT** | | RS-232 | Connection with control system (POS system, cash register, OPT terminal, etc) |
| **PUMP PORTS** | **Pump port 1** | Optically isolated RS-485: 2 wires for lines A and B, 2 wires for RTS control | Connection with fuel dispensers using configurable proprietary communication protocol (up to 16 fuel dispensers) |
| | **Pump port 2** | Optically isolated RS-485: 2 wires for lines A and B, 2 wires for RTS control | Connection with fuel dispensers using configurable proprietary communication protocol (up to 16 fuel dispensers) |
| | **Pump port 3** | Optically isolated RS-485: 2 wires for lines A and B, 2 wires for RTS control | Connection with fuel dispensers using configurable proprietary communication protocol (up to 16 fuel dispensers) |
| | **Pump port 4** | Optically isolated RS-485: 2 wires for lines A and B, 2 wires for RTS control | Connection with fuel dispensers using configurable proprietary communication protocol (up to 16 fuel dispensers) |
| | **DISP port (RS-485)** | Optically isolated RS-485: 2 wires for lines A and B | 1. PTS controllers interconnection (up to 16 PTS controllers) for simultaneous control over the same fuel dispensers and ATG systems<br>2. Connection with ATG systems (probes) using configurable proprietary com. protocol (up to 16 ATG probes) |
| **ATG PORTS** | **DISP port (RS-232)** | RS-232 | Connection with ATG system (console) using configurable proprietary communication protocol (up to 16 ATG probes) |
| | **LOG port** | RS-232 | 1. Connection with ATG system (console) using configurable proprietary communication protocol (up to 16 ATG probes)<br>2. Writing of operation log of PTS controller interaction with fuel dispensers, ATG systems, PTS interconnection |
| | **USER port** | RS-232 | Connection with ATG system (console) using configurable proprietary communication protocol (up to 16 ATG probes) |

# CONNECTION TO CONTROL SYSTEM

Connection to the control system (POS system, cash register, OPT terminal, etc) is made through a PC port, which has RS-232 interface.

Scheme of connections:



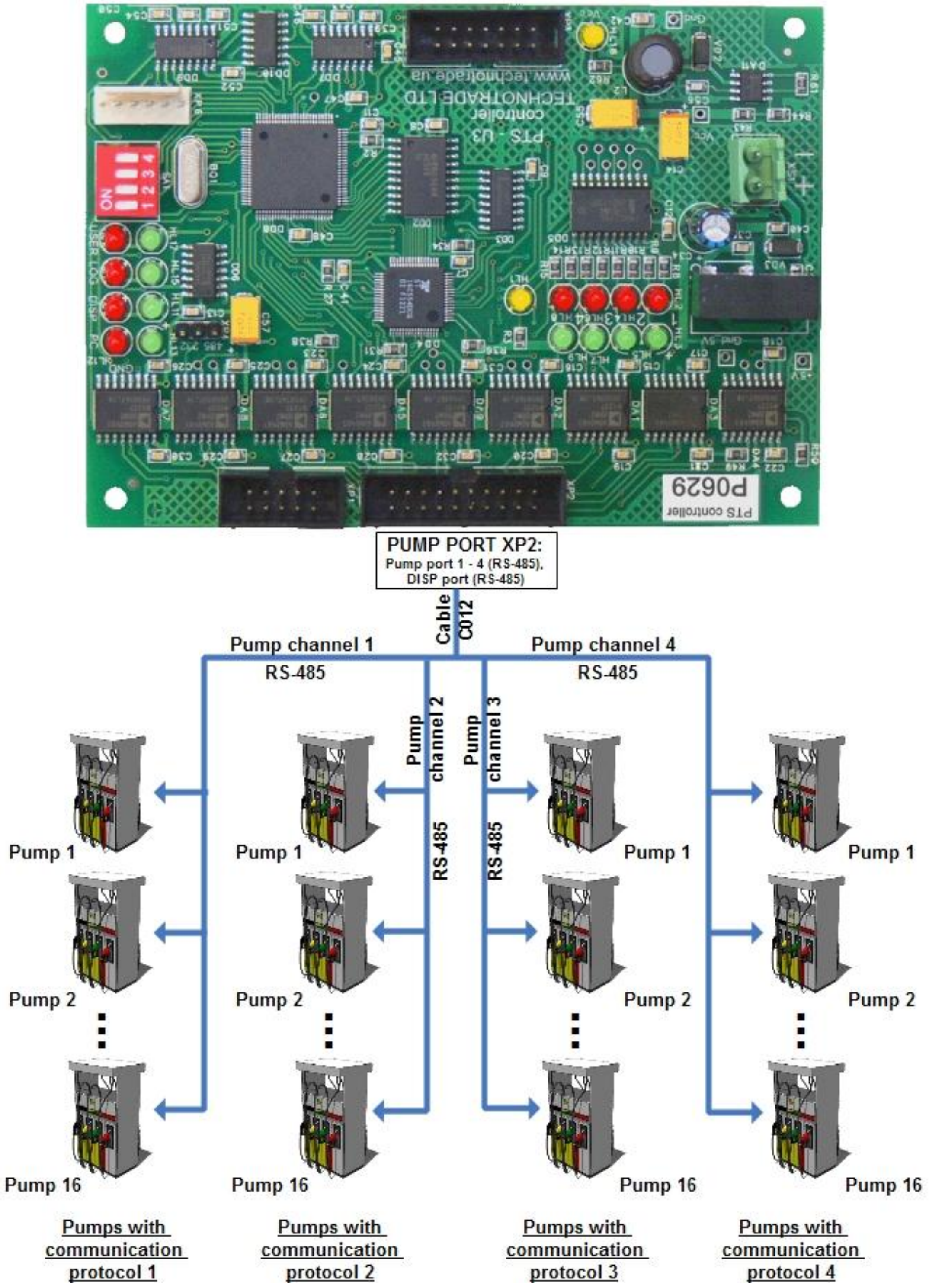
| XP1 | COM-port |
|---|---|
| 5,6 | 2 |
| 9,10 | 3 |
| 3,4 | 5 |

In case if the control system does not have a COM-port – it is possible to use any type of converter to COM-port (like USB-to-COM converter, Ethernet-to-COM, others).

Communication with PTS controller is made using commands and responses described in UniPump communication protocol (own proprietary protocol of TECHNOTRADE LTD) – see document “UniPump communication protocol specification for PTS controller over fuel dispensers and ATG systems” for more information. Thus PTS controller provides conversion of the common communication protocol UniPump into various proprietary communication protocols of fuel dispensers and ATG systems.

# CONNECTION TO FUEL DISPENSERS

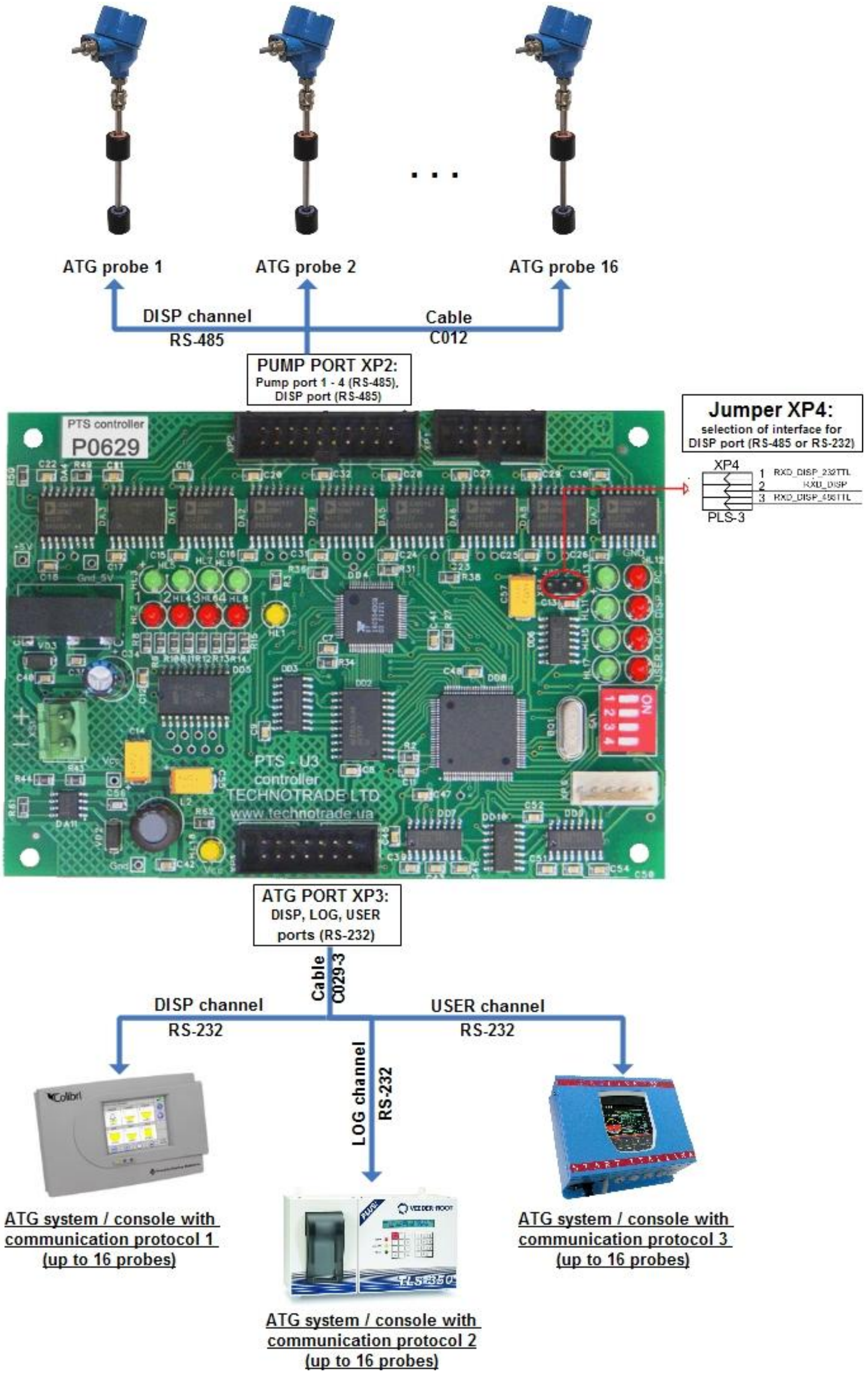
PTS controller can simultaneously control up to 16 fueling places (16 sides of 1-sided dispensers or 8 sides of 2-sided fuel dispensers) that use up to 4 various communication protocols (each of the pump channels can be adjusted to a separate communication protocol and baud rate and connect up to 16 fueling places) (see section “Examples of connection to fuel dispensers”).



***NOTE:*** if there are less than 4 various types of fuel dispensers at petrol station (which use various exchange protocols) – it is recommended to distribute fuel dispensers between 4 PTS controller channels in approximately equal quantities in order to minimize delays between fuel dispensers querying in the same channel.

# CONNECTION TO ATG SYSTEMS

PTS controller can simultaneously control up to 16 ATG probes (gauges) (separate probes or probes connected to ATG systems / consoles) that use up to 3 various communication protocols (each of the ATG channels can be adjusted to a separate communication protocol and baud rate and connect up to 16 ATG probes) (see section "Examples of connection to ATG systems").

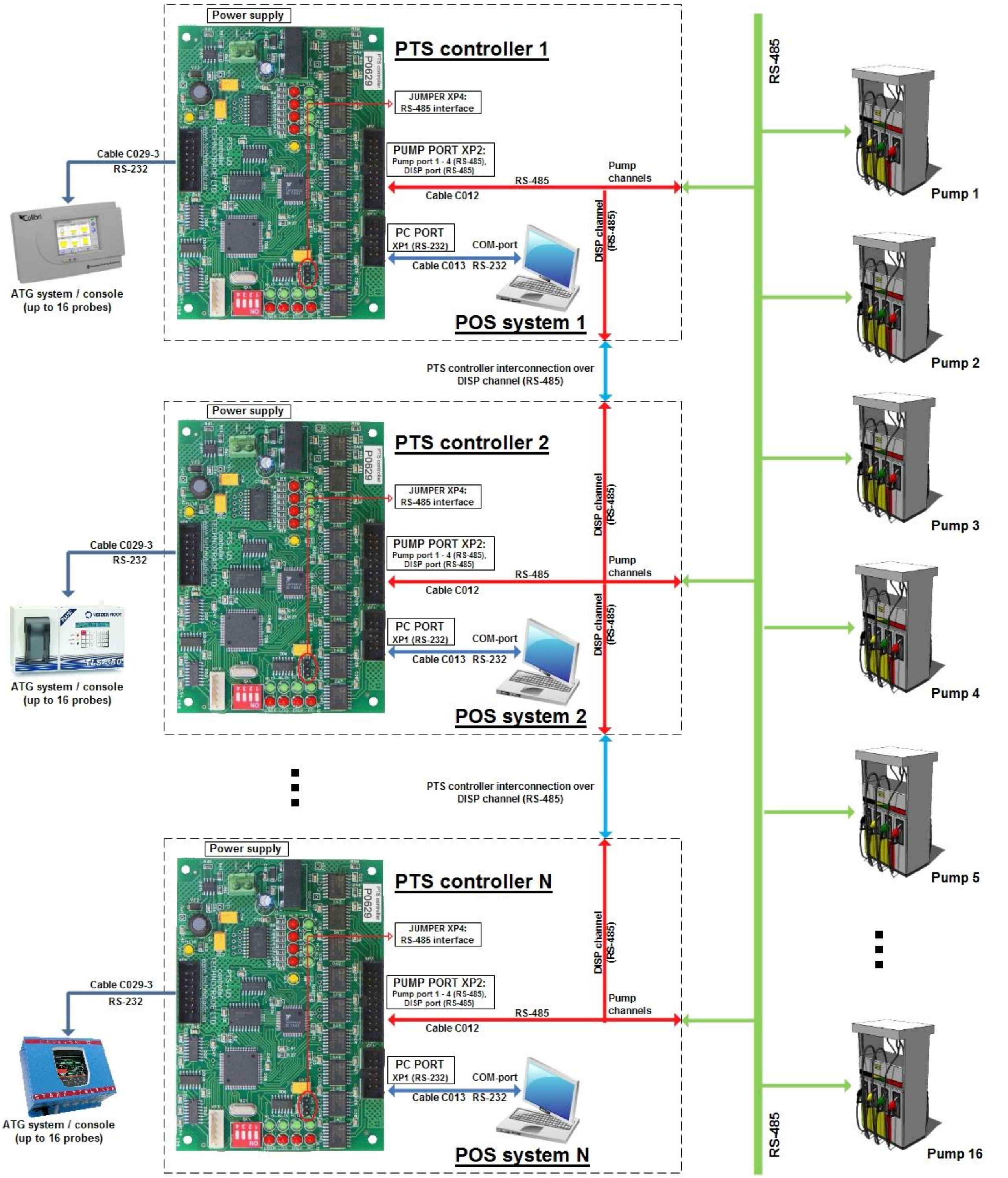


***NOTE:*** DISP channel provides a possibility to connect ATG system (probes) over either RS-485 or RS-232 interfaces – interface is selected using a jumper XP4, located on PTS PCB board.

# PTS CONTROLLER MULTI POS SYSTEMS OPERATION

PTS controller allows a possibility to lead management over the same fuel dispensers from several POS systems and share ATG probes measurement values between several interconnected PTS controllers. Thus every interconnected PTS controller is able to provide control over any of the connected fuel dispensers and know ATG system measurement data of every other interconnected PTS controller.

Scheme of PTS controllers interconnections:



***NOTE!*** In order to enable PTS interconnection a jumper XP4 should be selected for DISP RS-485 interface and DISP channel should not be configured for any ATG system (see section “PTS controller configuration”).

## Configuration:

At necessity to organize several working places at petrol station (several POS systems) each of the POS systems should have its own PTS controller connected. PTS controllers should be interconnected with each other through a dispatcher PTS port (DISP channel on RS-485 interface) and with fuel dispensers. Thus control over each fuel dispenser at petrol station can be made from every POS system and each PTS controller will know measurement data of every ATG probe connected to any of the interconnected PTS controllers. At authorization of a fuel dispenser from one POS system the fuel dispenser becomes locked by PTS controller of this POS system and while it is locked all other POS systems can not control over given fuel dispenser, they can only monitor current state of the fuel dispenser. When operation of given POS system is finished with given fuel dispenser, PTS controller of this POS system unlocks the fuel dispenser and it becomes commonly available for all other PTS controllers, which makes it possible to be controlled (locked) by any of the PTS controllers connected to other POS systems.

***NOTE!*** In order to provide control over same fuel dispensers from several PTS controllers locking and unlocking of dispensers is required. For this reason in configuration of PTS controller in parameters for PTS controller ‘Lock’ and ‘Unlock’ commands should be set to be used (see section “Configuration of parameters for PTS controller”).

***NOTE!*** In order to provide correct exchange of ATG systems measurement data between interconnected PTS controllers logical addresses of connected ATG probes in configuration of interconnected PTS controllers should not intersect.

# PTS CONTROLLER CONFIGURATION

## General information

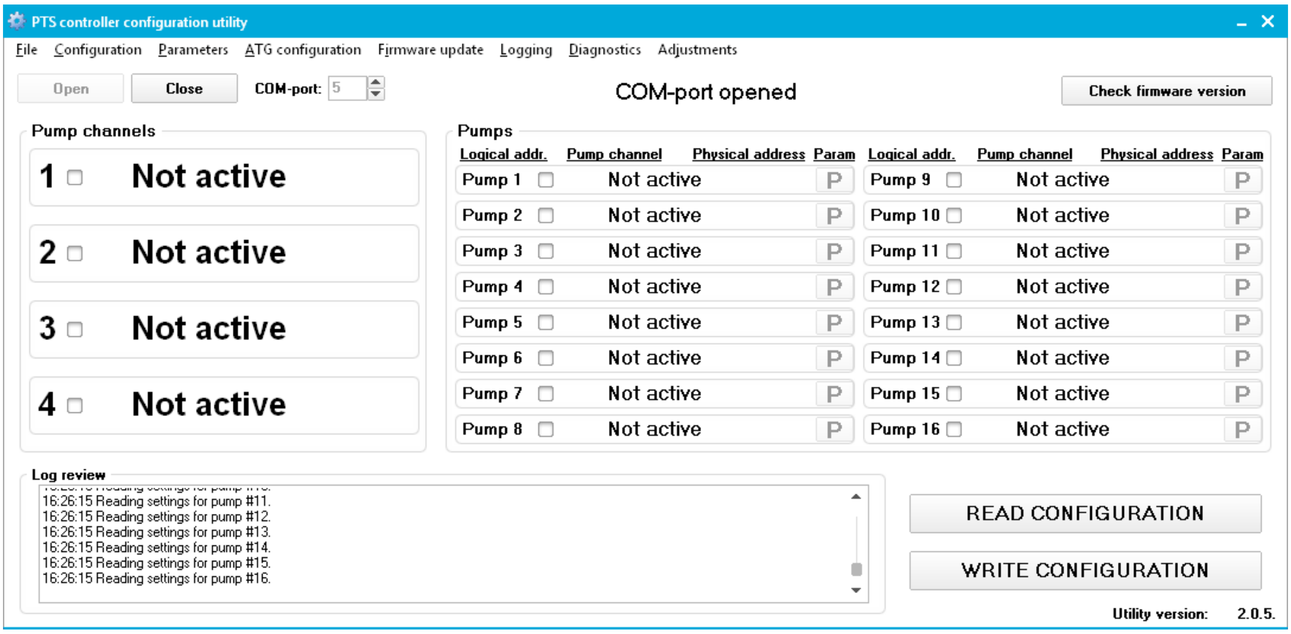
PTS controller configuration utility *Pts_conf.exe* serves for configuration of the PTS controller.

Currently utility is made under Windows OS. It is based on PTS controller communication protocol UniPump.

PTS controller configuration utility *Pts_conf.exe* is a multipurpose utility for PTS controller. It provides the following possibilities:

- configuration of pumps (fuel dispensers) channels
- configuration of ATG systems (probes) channels
- configuration of pumps (fuel dispensers) parameters
- configuration of PTS controller parameters
- configuration of logging parameters
- update of the PTS controller firmware
- taking logs from PTS controller
- self-diagnostics of PTS controller interfaces and switches
- saving of all configuration of PTS controller to file
- loading of all configuration of PTS controller from file

*Pts_conf.exe* utility is developed to be multilanguage and supports various skins in order to suit to various control systems graphical interfaces.



COM-port: 5

COM-port opened

Pump channels

1 ☐ Not active

2 ☐ Not active

3 ☐ Not active

4 ☐ Not active

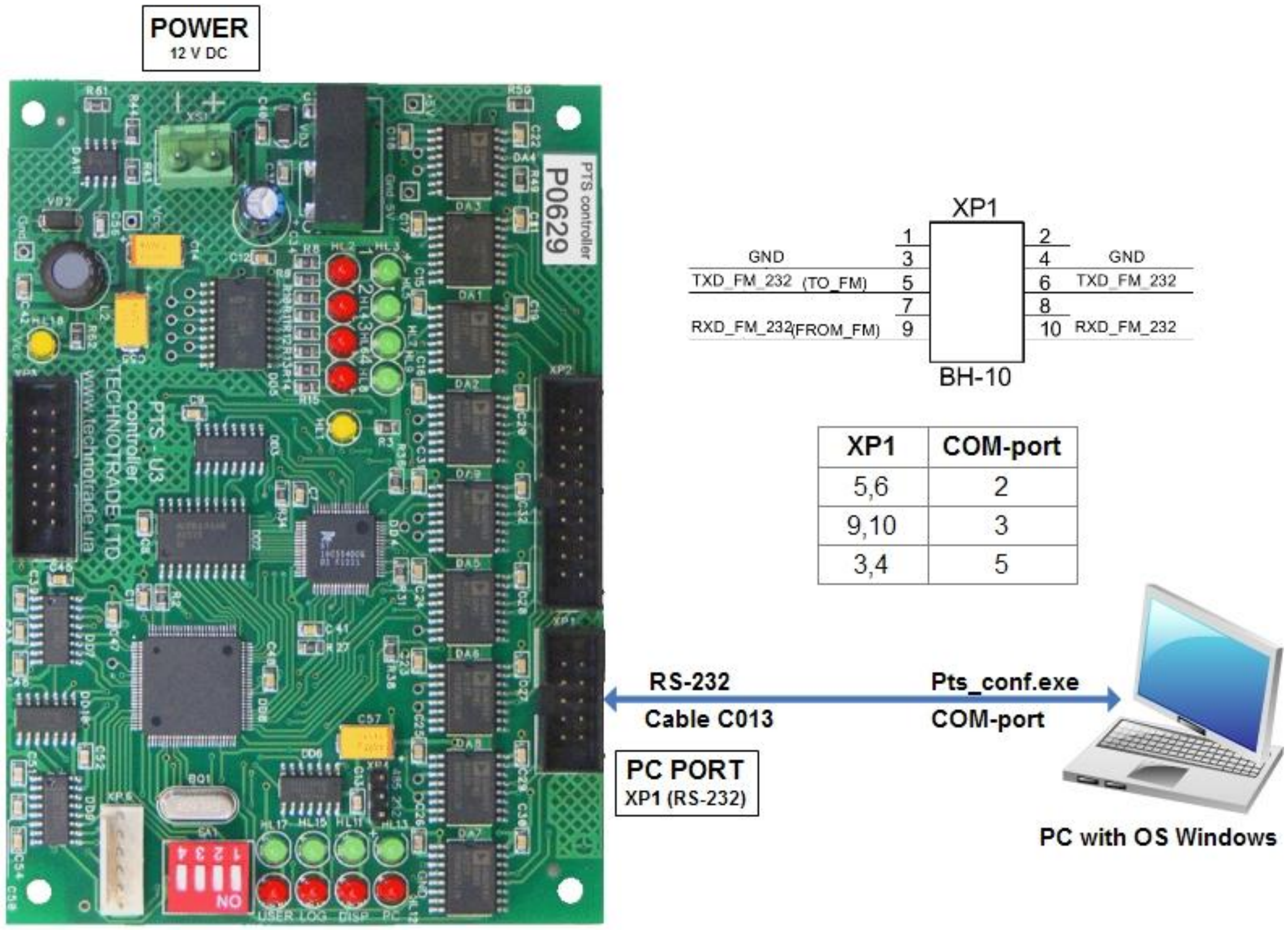
| Logical addr. | Pump channel | Physical address | Param | Logical addr. | Pump channel | Physical address | Param |
|---|---|---|---|---|---|---|---|
| Pump 1 | ☐ | Not active | P | Pump 9 | ☐ | Not active | P |
| Pump 2 | ☐ | Not active | P | Pump 10 | ☐ | Not active | P |
| Pump 3 | ☐ | Not active | P | Pump 11 | ☐ | Not active | P |
| Pump 4 | ☐ | Not active | P | Pump 12 | ☐ | Not active | P |
| Pump 5 | ☐ | Not active | P | Pump 13 | ☐ | Not active | P |
| Pump 6 | ☐ | Not active | P | Pump 14 | ☐ | Not active | P |
| Pump 7 | ☐ | Not active | P | Pump 15 | ☐ | Not active | P |
| Pump 8 | ☐ | Not active | P | Pump 16 | ☐ | Not active | P |

```
16:26:15 Reading settings for pump #11.
16:26:15 Reading settings for pump #12.
16:26:15 Reading settings for pump #13.
16:26:15 Reading settings for pump #14.
16:26:15 Reading settings for pump #15.
16:26:15 Reading settings for pump #16.
```

Utility version: 2.0.5.

Use PTS controller configuration utility *Pts_conf.exe* to make configuration of the PTS controller.

## *Scheme of connections*



| XP1 | COM-port |
|---|---|
| 5,6 | 2 |
| 9,10 | 3 |
| 3,4 | 5 |

***NOTE!*** On some PCs for correct operation COM-port FIFO settings should be adjusted to Tx: 1, Rx: 1. This may also apply to situations when a PC does not have a native COM-port and external COM-port converter are applied (like USB-to-COM, PCI-to-COM, Ethernet-to-COM, etc).

## *Launching pts_conf utility*

Run *Pts_conf.exe* utility. Make sure that in main menu in tab “Adjustments” type of PTS controller connection “Direct connection of controller to PC” is selected (other types of connection serve for connection of PTS controller through fiscal modules). Set up a correct COM-port number and press “Open” button.

## *Configuration of pump channels*

Pump channels are configured on main form of the utility.

To read current configuration of the PTS controller use a button “READ CONFIGURATION”.

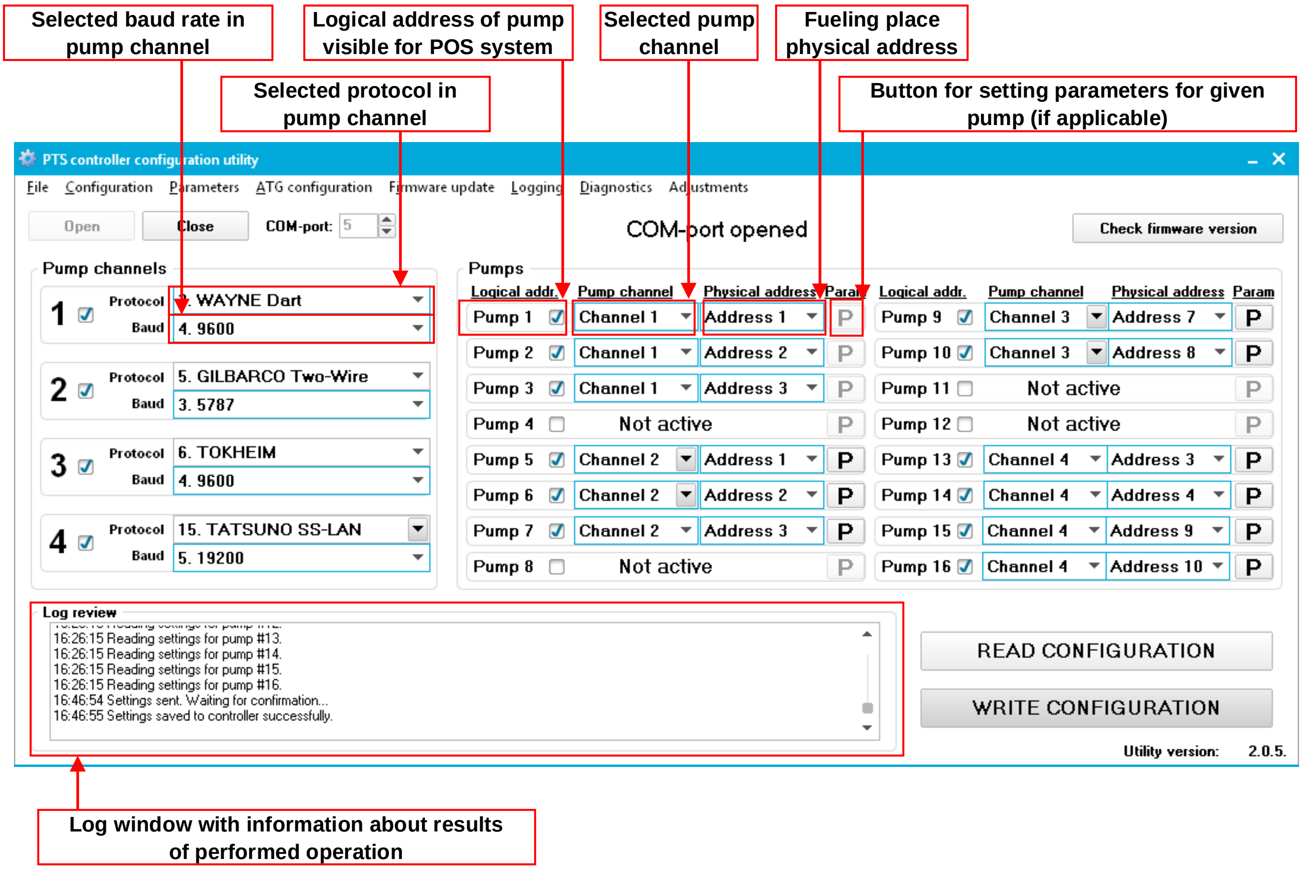
Selection of the pump channel is made by activating of a checkbox near it (there are 4 pump channels available). When pump channel is activated it is necessary to select a protocol to be used on pump channel and baud rate from the dropdown lists.

Selection of the connected pump is made by activating of a checkbox near it (there are 16 pumps available). When pump is activated it is necessary to select:

- a pump channel, to which this pump belongs to, from the dropdown list
- physical address of the fueling place from the dropdown list

Logical address of the pump means the address on which the control system (POS system, cash register, OPT, etc) will see given fueling place. Physical address of the pump means address of the real fueling place, which is programmed or set in configuration of the fuel dispenser fueling place.

After configuration is finished it is necessary to click a button "WRITE CONFIGURATION", which will write current pump channels configuration to controller. Information about result of operation of writing of configuration to controller (whether it is made successfully or there was some problem) will be written in the log window on the form.


Selected baud rate in pump channel
Logical address of pump visible for POS system
Selected pump channel
Fueling place physical address
Selected protocol in pump channel
Button for setting parameters for given pump (if applicable)
PTS controller configuration utility
File Configuration Parameters ATG configuration Firmware update Logging Diagnostics Adjustments
Open
Close
Check firmware version
READ CONFIGURATION
WRITE CONFIGURATION
Log window with information about results of performed operation

COM-port: 5

COM-port opened

Pump channels

| | | | |
|---|---|---|---|
| 1 ☑ | Protocol | 2. WAYNE Dart | |
| | Baud | 4. 9600 | |
| 2 ☑ | Protocol | 5. GILBARCO Two-Wire | |
| | Baud | 3. 5787 | |
| 3 ☑ | Protocol | 6. TOKHEIM | |
| | Baud | 4. 9600 | |
| 4 ☑ | Protocol | 15. TATSUNO SS-LAN | |
| | Baud | 5. 19200 | |

Pumps

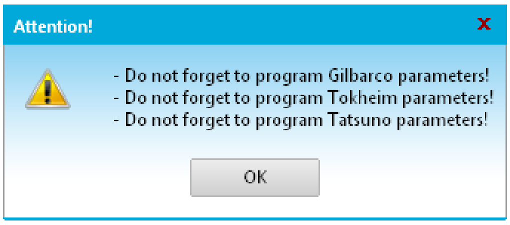
| Logical addr. | Pump channel | Physical address | Param | Logical addr. | Pump channel | Physical address | Param |
|---|---|---|---|---|---|---|---|
| Pump 1 ☑ | Channel 1 | Address 1 | P | Pump 9 ☑ | Channel 3 | Address 7 | P |
| Pump 2 ☑ | Channel 1 | Address 2 | P | Pump 10 ☑ | Channel 3 | Address 8 | P |
| Pump 3 ☑ | Channel 1 | Address 3 | P | Pump 11 □ | Not active | | P |
| Pump 4 □ | Not active | | P | Pump 12 □ | Not active | | P |
| Pump 5 ☑ | Channel 2 | Address 1 | P | Pump 13 ☑ | Channel 4 | Address 3 | P |
| Pump 6 ☑ | Channel 2 | Address 2 | P | Pump 14 ☑ | Channel 4 | Address 4 | P |
| Pump 7 ☑ | Channel 2 | Address 3 | P | Pump 15 ☑ | Channel 4 | Address 9 | P |
| Pump 8 □ | Not active | | P | Pump 16 ☑ | Channel 4 | Address 10 | P |

Log review

```
16:26:15 Reading settings for pump #13.
16:26:15 Reading settings for pump #14.
16:26:15 Reading settings for pump #15.
16:26:15 Reading settings for pump #16.
16:46:54 Settings sent. Waiting for confirmation...
16:46:55 Settings saved to controller successfully.
```

Utility version: 2.0.5.

For some protocols it is also required to configure parameters. In this case after a button "WRITE CONFIGURATION" is clicked a warning dialog will appear:



- Do not forget to program Gilbarco parameters!
- Do not forget to program Tokheim parameters!
- Do not forget to program Tatsuno parameters!

At this a button with a letter "P" on it will be activated. By clicking on this button a corresponding window for parameters configuration of the selected pump channel protocol will be shown (see section 'Parameters configuration' below).

Current configuration of pump channels can be saved to a file on a hard drive for future usage (select in main menu a tab 'File' → 'Save as pump configuration' to save configuration or 'File' → 'Open pump configuration' to open configuration from file).

## *Configuration of ATG channels*

ATG channels are configured on form, which is called from the main menu in tab “ATG configuration” by selecting an item “Configuration of ATG channels”.

To read current configuration of the PTS controller use a button “READ CONFIGURATION”.

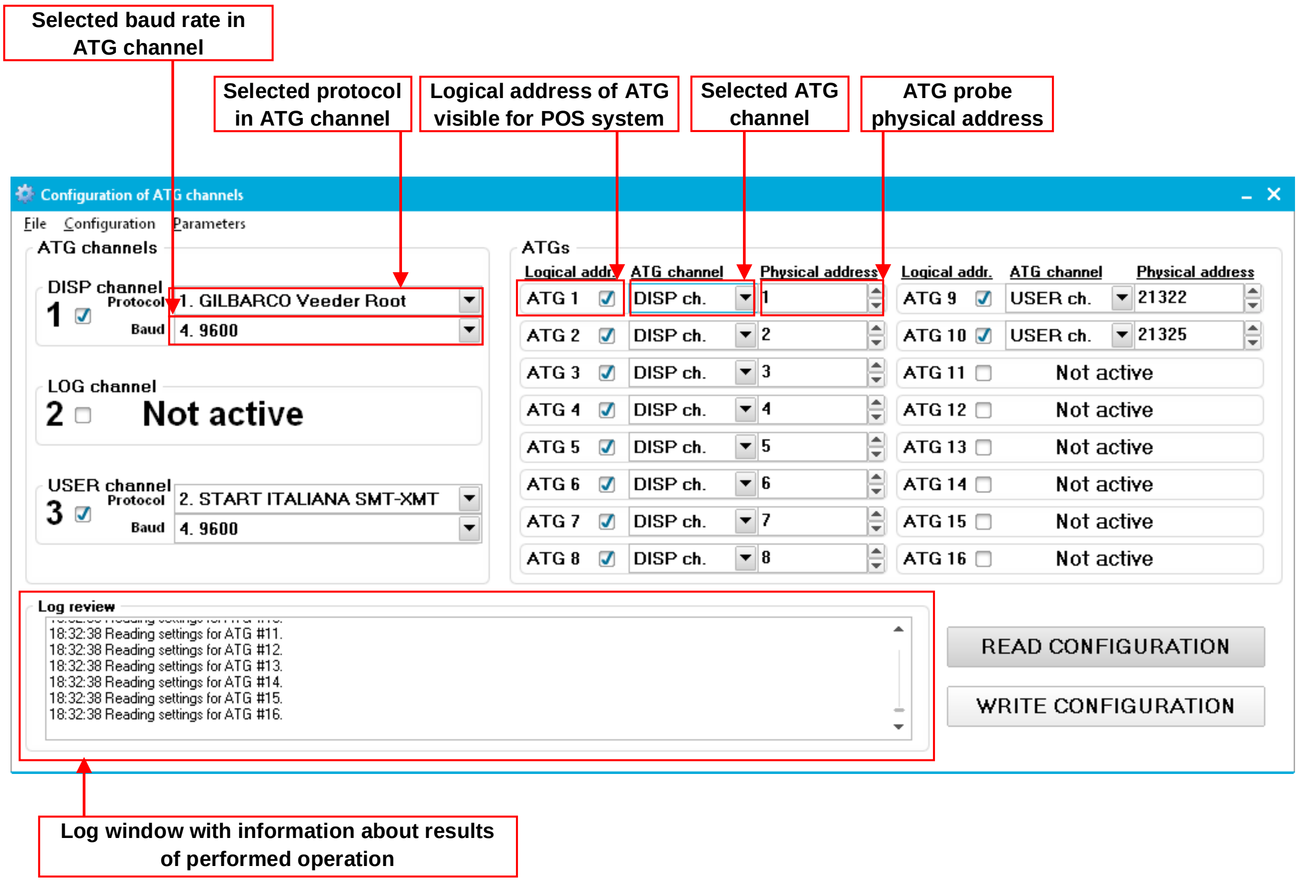
Selection of the ATG channel is made by activating of a checkbox near it (there are 4 ATG channels available). When ATG channel is activated it is necessary to select a protocol to be used on ATG channel and baud rate from the dropdown lists.

Selection of the connected ATG is made by activating of a checkbox near it (there are 16 ATG available). When ATG is activated it is necessary to select:

- ATG channel, to which this ATG belongs to, from the dropdown list
- physical address of the ATG probe from the dropdown list

Logical address of the ATG means the address on which the control system (POS system, cash register, OPT, etc) will see given ATG probe. Physical address of the ATG means address of the real ATG probe, which is programmed or set in configuration of the ATG console (probe).

After configuration is finished it is necessary to click a button “WRITE CONFIGURATION”, which will write current ATG channels configuration to controller. Information about result of operation of writing of configuration to controller (whether it is made successfully or there was some problem) will be written in the log window on the form.


Selected baud rate in ATG channel
Selected protocol in ATG channel
Logical address of ATG visible for POS system
Selected ATG channel
ATG probe physical address
Configuration of ATG channels
File Configuration Parameters
ATG channels
ATGs
Log review
READ CONFIGURATION
WRITE CONFIGURATION
Log window with information about results of performed operation

DISP channel 1 ☑ Protocol: 1. GILBARCO Veeder Root; Baud: 4. 9600

LOG channel 2 □ Not active

USER channel 3 ☑ Protocol: 2. START ITALIANA SMT-XMT; Baud: 4. 9600

| Logical addr. | ATG channel | Physical address | Logical addr. | ATG channel | Physical address |
|---|---|---|---|---|---|
| ATG 1 ☑ | DISP ch. | 1 | ATG 9 ☑ | USER ch. | 21322 |
| ATG 2 ☑ | DISP ch. | 2 | ATG 10 ☑ | USER ch. | 21325 |
| ATG 3 ☑ | DISP ch. | 3 | ATG 11 □ | Not active | |
| ATG 4 ☑ | DISP ch. | 4 | ATG 12 □ | Not active | |
| ATG 5 ☑ | DISP ch. | 5 | ATG 13 □ | Not active | |
| ATG 6 ☑ | DISP ch. | 6 | ATG 14 □ | Not active | |
| ATG 7 ☑ | DISP ch. | 7 | ATG 15 □ | Not active | |
| ATG 8 ☑ | DISP ch. | 8 | ATG 16 □ | Not active | |

```
18:32:38 Reading settings for ATG #11.
18:32:38 Reading settings for ATG #12.
18:32:38 Reading settings for ATG #13.
18:32:38 Reading settings for ATG #14.
18:32:38 Reading settings for ATG #15.
18:32:38 Reading settings for ATG #16.
```

Current configuration of ATG channels can be saved to a file on a hard drive for future usage (select in main menu a tab ‘File’ → ‘Save as ATG configuration’ to save configuration or ‘File’ → ‘Open ATG configuration’ to open configuration from file).

# *Parameters configuration*

Configuration of parameters in PTS controller includes:

- configuration of parameters for pumps protocols
- configuration of parameters for ATG systems
- configuration of parameters for PTS controller
- configuration of parameters for interface converters
- configuration of parameters for logging

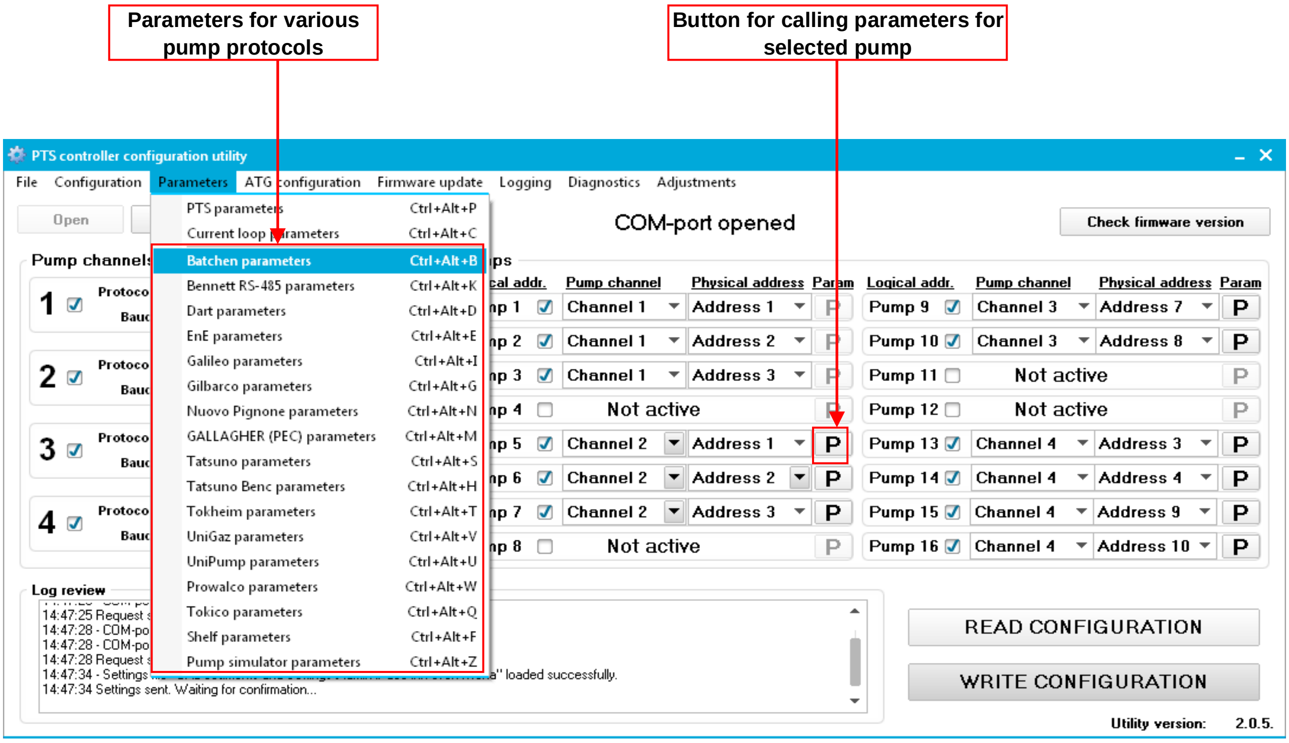
All parameters are described in details in UniPump communication protocol description.

## *Configuration of parameters for pumps protocols*

Due to a reason that various fuel dispensers are supplied worldwide with various configurations like

- single product and multi product
- quantity of digits in price and cost per liter/gallon,
- mode of dispenser operation and dispenser type, quantity of nozzles per side
- others

For flexibility of adjustment of PTS controller operation with fuel dispensers special parameters for pump protocols in PTS controllers are foreseen. Adjustment of parameters for every separate communication protocol is made using a main menu tab “Parameters” or by clicking a button with a letter “P” located in a main form near each of the pumps (where it is activated).


Parameters for various pump protocols
Button for calling parameters for selected pump
PTS controller configuration utility
File
Configuration
Parameters
ATG configuration
Firmware update
Logging
Diagnostics
Adjustments
Open
PTS parameters Ctrl+Alt+P
Current loop parameters Ctrl+Alt+C
Batchen parameters Ctrl+Alt+B
Bennett RS-485 parameters Ctrl+Alt+K
Dart parameters Ctrl+Alt+D
EnE parameters Ctrl+Alt+E
Galileo parameters Ctrl+Alt+I
Gilbarco parameters Ctrl+Alt+G
Nuovo Pignone parameters Ctrl+Alt+N
GALLAGHER (PEC) parameters Ctrl+Alt+M
Tatsuno parameters Ctrl+Alt+S
Tatsuno Benc parameters Ctrl+Alt+H
Tokheim parameters Ctrl+Alt+T
UniGaz parameters Ctrl+Alt+V
UniPump parameters Ctrl+Alt+U
Prowalco parameters Ctrl+Alt+W
Tokico parameters Ctrl+Alt+Q
Shelf parameters Ctrl+Alt+F
Pump simulator parameters Ctrl+Alt+Z
COM-port opened
Check firmware version
Pump channels
Protoco
Bauc
Log review
READ CONFIGURATION
WRITE CONFIGURATION
Utility version: 2.0.5.

| cal addr. | Pump channel | Physical address | Param | Logical addr. | Pump channel | Physical address | Param |
|---|---|---|---|---|---|---|---|
| np 1 ☑ | Channel 1 | Address 1 | P | Pump 9 ☑ | Channel 3 | Address 7 | P |
| np 2 ☑ | Channel 1 | Address 2 | P | Pump 10 ☑ | Channel 3 | Address 8 | P |
| np 3 ☑ | Channel 1 | Address 3 | P | Pump 11 □ | Not active | | P |
| np 4 □ | Not active | | P | Pump 12 □ | Not active | | P |
| np 5 ☑ | Channel 2 | Address 1 | P | Pump 13 ☑ | Channel 4 | Address 3 | P |
| np 6 ☑ | Channel 2 | Address 2 | P | Pump 14 ☑ | Channel 4 | Address 4 | P |
| np 7 ☑ | Channel 2 | Address 3 | P | Pump 15 ☑ | Channel 4 | Address 9 | P |
| np 8 □ | Not active | | P | Pump 16 ☑ | Channel 4 | Address 10 | P |

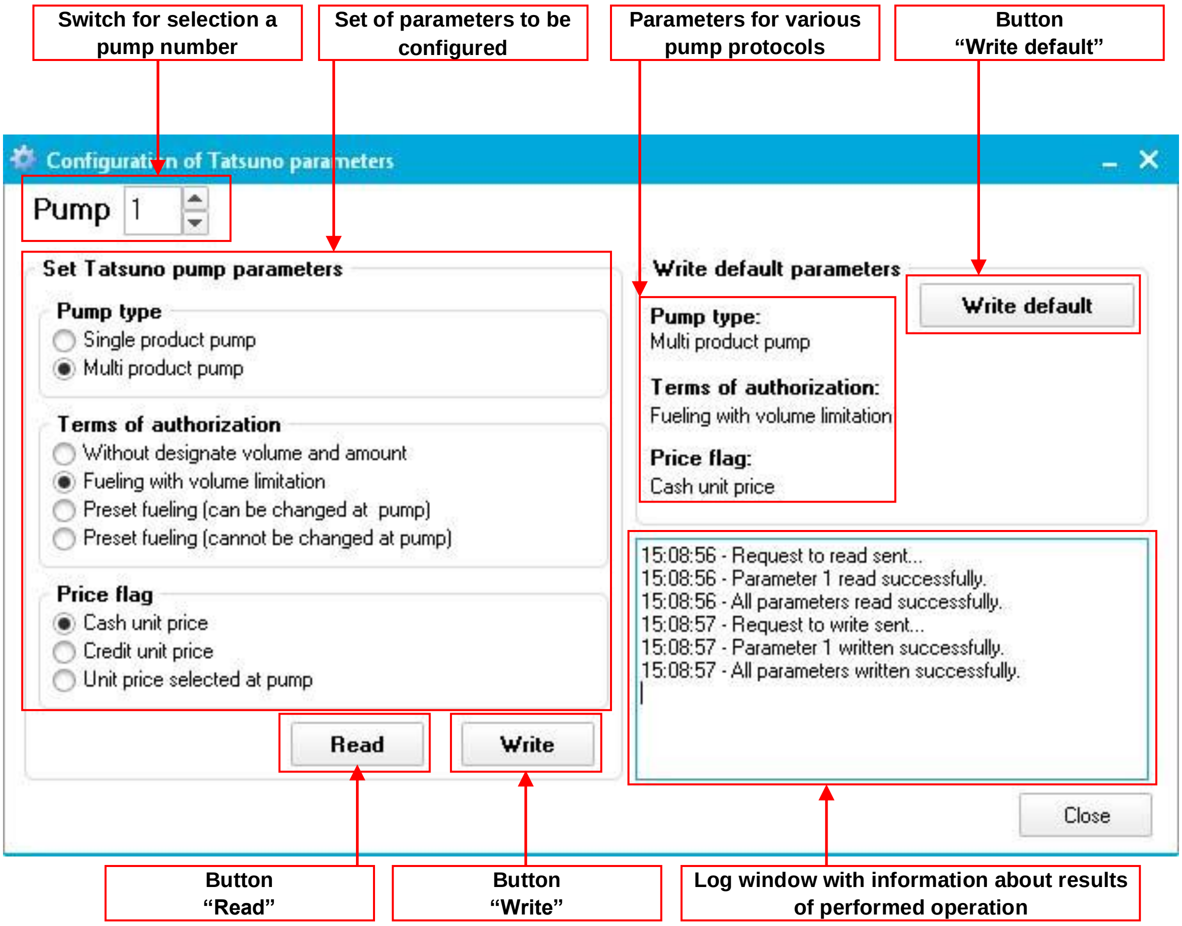
```
14:47:25 Request s
14:47:28 - COM-po
14:47:28 - COM-po
14:47:28 Request s
14:47:34 - Settings                                  a" loaded successfully.
14:47:34 Settings sent. Waiting for confirmation...
```

All forms for configuration of pump parameters are similar and include:

- switch for selection a pump number, which parameters are configured
- set of parameters to be configured
- button “Read” to read parameters for selected pump from PTS
- button “Write” to write parameters for selected pump into PTS
- button “Write default” to write default parameters for selected pump from PTS
- description of default parameters, which are to be written at clicking a button “Write default”

- log window with information about results of performed operation

As an example a form for configuration of parameters for Tatsuno SS-LAN protocol look as shown below:



Switch for selection a pump number
Set of parameters to be configured
Parameters for various pump protocols
Button “Write default”
Configuration of Tatsuno parameters
Read
Write
Write default
Close
Button “Read”
Button “Write”
Log window with information about results of performed operation

Pump 1

Set Tatsuno pump parameters

Pump type
○ Single product pump
● Multi product pump

Terms of authorization
○ Without designate volume and amount
● Fueling with volume limitation
○ Preset fueling (can be changed at  pump)
○ Preset fueling (cannot be changed at pump)

Price flag
● Cash unit price
○ Credit unit price
○ Unit price selected at pump

Write default parameters

Pump type:
Multi product pump

Terms of authorization:
Fueling with volume limitation

Price flag:
Cash unit price

```
15:08:56 - Request to read sent...
15:08:56 - Parameter 1 read successfully.
15:08:56 - All parameters read successfully.
15:08:57 - Request to write sent...
15:08:57 - Parameter 1 written successfully.
15:08:57 - All parameters written successfully.
```

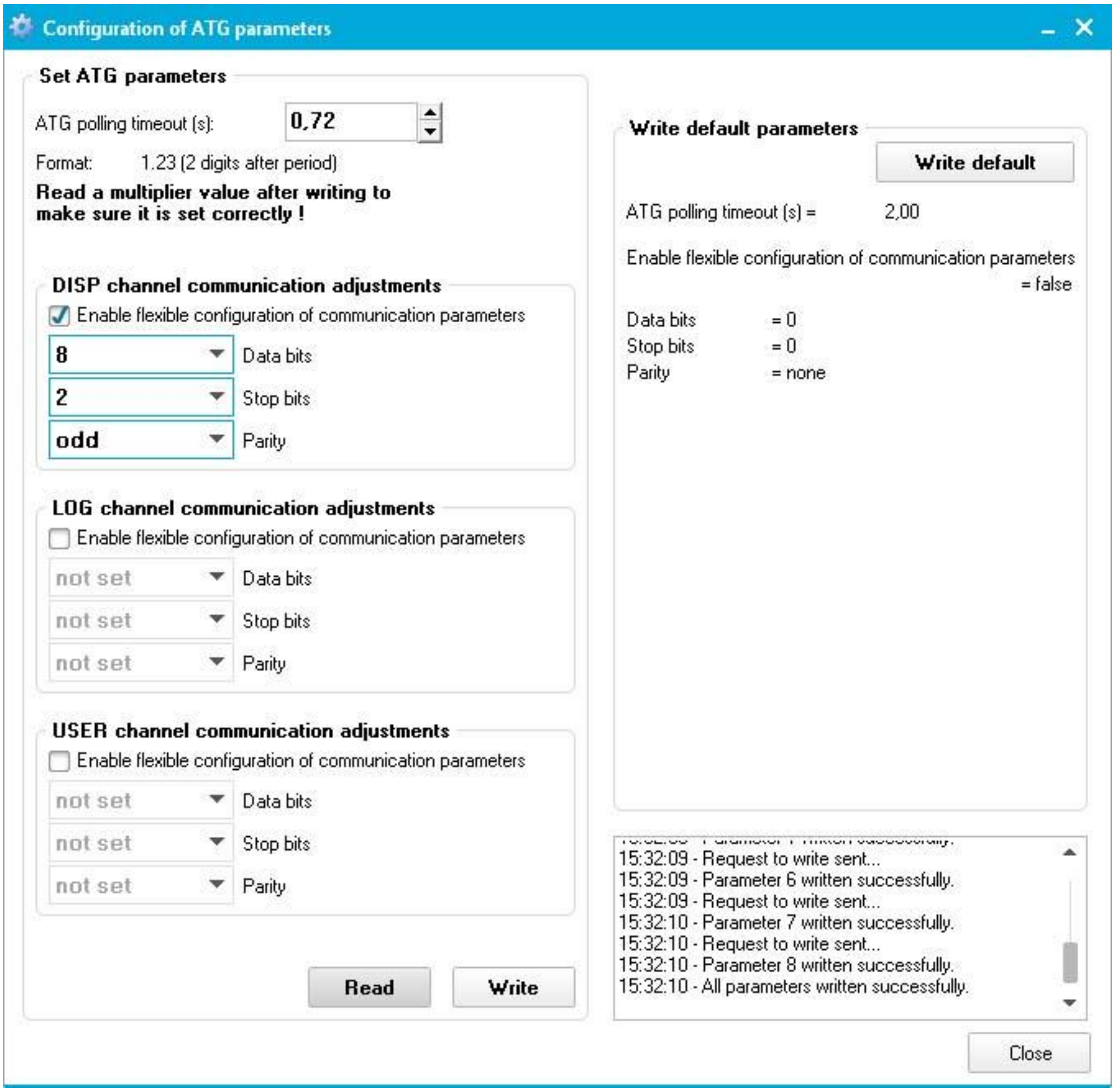
## *Configuration of parameters for ATG systems*

Form for configuration of parameters for ATG systems is called from main form main menu tab ‘ATG configuration’ → ‘ATG parameters’ or from a form for configuration of ATG channels main menu “Parameters” → ‘ATG parameters’.

Parameters for ATG systems include:

- setting of speed of polling the ATG probes (delay between pollings ATG probes)
- setting of flexible configuration of communication adjustments (data bits, stop bits, parity) for each ATG channels

Parameter for setting of flexible configuration of communication adjustments is necessary for cases when different ATG systems using the same communication protocol may have different communication settings.


Configuration of ATG parameters
Set ATG parameters
Write default parameters
Write default
DISP channel communication adjustments
LOG channel communication adjustments
USER channel communication adjustments
Read
Write
Close

ATG polling timeout (s): 0,72

Format: 1.23 (2 digits after period)

Read a multiplier value after writing to make sure it is set correctly !

DISP channel communication adjustments

☑ Enable flexible configuration of communication parameters

8 Data bits

2 Stop bits

odd Parity

LOG channel communication adjustments

☐ Enable flexible configuration of communication parameters

not set Data bits

not set Stop bits

not set Parity

USER channel communication adjustments

☐ Enable flexible configuration of communication parameters

not set Data bits

not set Stop bits

not set Parity

ATG polling timeout (s) = 2,00

Enable flexible configuration of communication parameters = false

Data bits = 0
Stop bits = 0
Parity = none

```
15:32:09 - Request to write sent...
15:32:09 - Parameter 6 written successfully.
15:32:09 - Request to write sent...
15:32:10 - Parameter 7 written successfully.
15:32:10 - Request to write sent...
15:32:10 - Parameter 8 written successfully.
15:32:10 - All parameters written successfully.
```

## *Configuration of parameters for PTS controller*

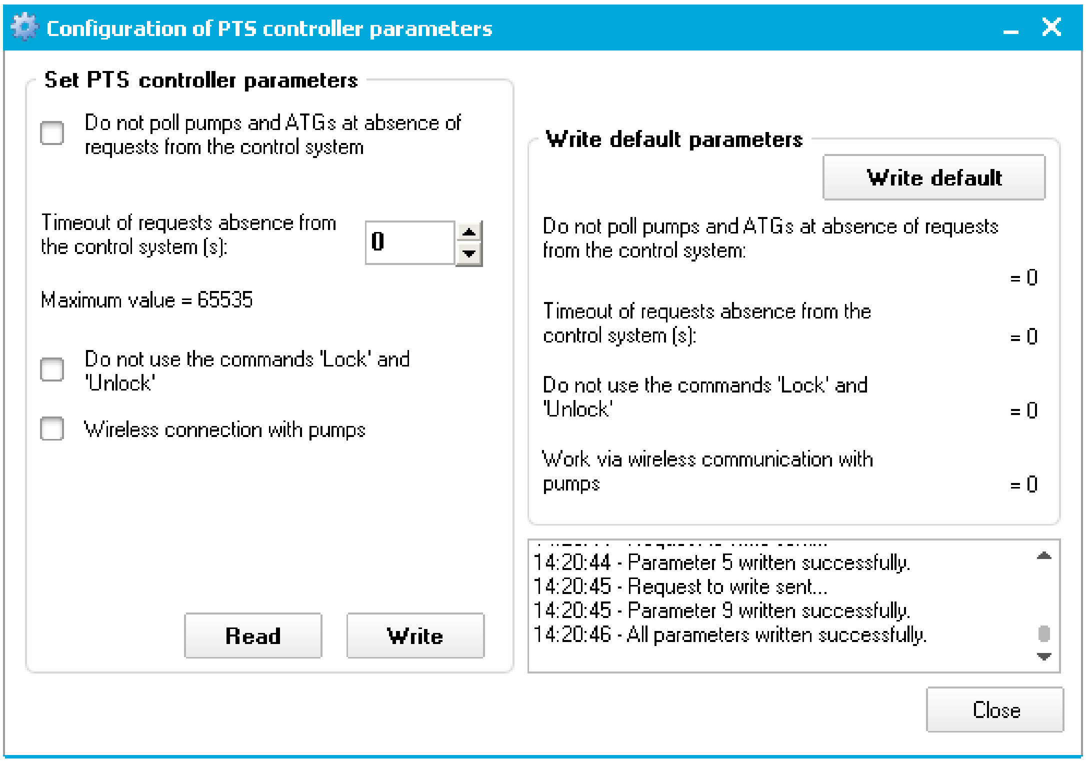
Form for configuration of parameters for PTS controller is called from main form main menu tab 'Parameters' → 'PTS controller parameters'.

Parameters for PTS controller include:

- setting of the way PTS controller should work: poll pumps and ATG systems at absence of requests from the control system or not
- in case if the PTS controller should not poll pumps and ATG systems at absence of requests from the control system – timeout in seconds of requests absence from the control system after which polling should not be performed
- setting of the way PTS controller should work: use commands 'Lock' and 'Unlock' or not
- setting of the way PTS controller is linked with pumps: wireless or wired connection

In accordance with UniPump communication protocol of PTS controller commands 'Lock' and 'Unlock' are used every time control system wants to take control over fuel dispenser – it should first lock it with command 'Lock' and in the end of control – unlock it with command 'Unlock'. This behavior is required for cases when there are several control systems installed at the same sight (petrol station) each having its own PTS controller connected and each of these control systems should be able to provide control over any of the fuel dispensers on the sight. Using these commands 'Lock' and 'Unlock' gives a possibility for each control system to provide control over any of the connected fuel dispensers, Read more about it below in section "PTS controller multi POS systems operation".

In case if parameters are set not to use commands 'Lock' and 'Unlock' – then PTS will always respond that pump is locked.


Configuration of PTS controller parameters
Set PTS controller parameters
Write default parameters
Write default
Read
Write
Close

□ Do not poll pumps and ATGs at absence of requests from the control system

Timeout of requests absence from the control system (s): 0

Maximum value = 65535

□ Do not use the commands 'Lock' and 'Unlock'

□ Wireless connection with pumps

Do not poll pumps and ATGs at absence of requests from the control system: = 0

Timeout of requests absence from the control system (s): = 0

Do not use the commands 'Lock' and 'Unlock' = 0

Work via wireless communication with pumps = 0

```
14:20:44 - Parameter 5 written successfully.
14:20:45 - Request to write sent...
14:20:45 - Parameter 9 written successfully.
14:20:46 - All parameters written successfully.
```

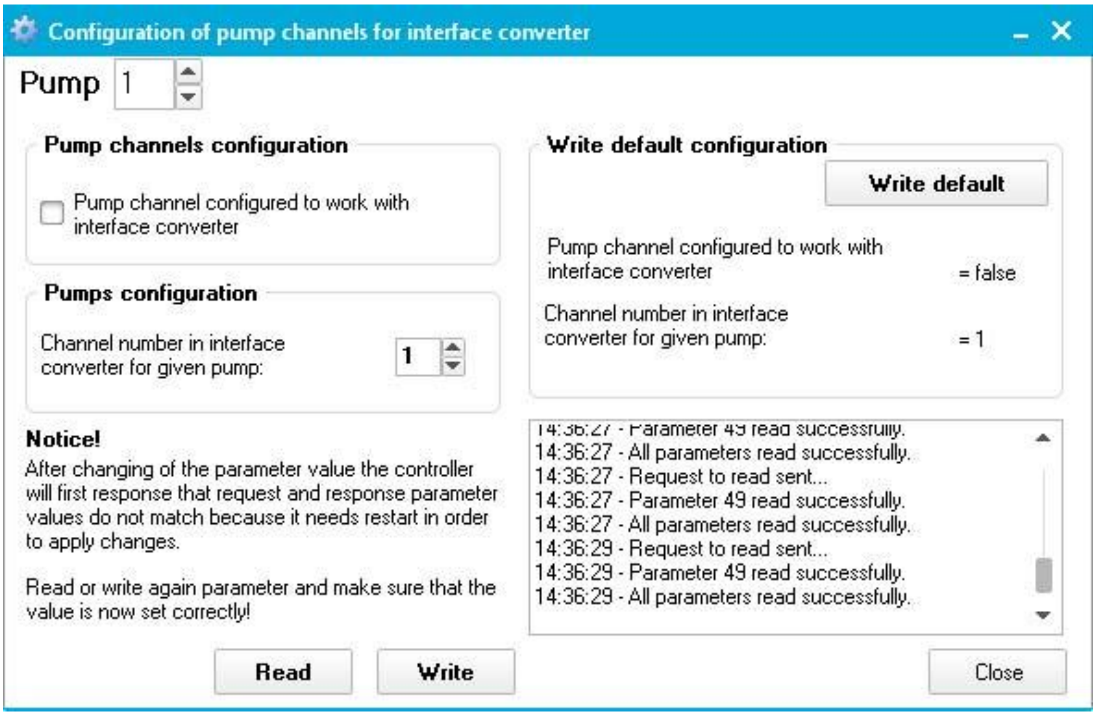
## *Configuration of parameters for interface converters*

Form for configuration of parameters for PTS controller is called from main form main menu tab 'Parameters' → 'Interface converter parameters'.

Parameters for interface converters are used for some of manufactured interface converters working together with PTS controller, which are used for fuel dispensers, which do not have addressing in communication protocol. So, addressing of the fuel dispenser is made by a number of a pump channel of the interface converter, to which a fuel dispenser is connected.

Parameters for interface converters include:

- setting of the logical pump number, which is connected through an interface converter to PTS controller
- setting if the pump channel should be configured to work with interface converter or in standard way
- channel number in interface converter for given pump (substitutes address of the fuel dispenser)


Configuration of pump channels for interface converter
Pump 1
Pump channels configuration
Write default configuration
Write default
Pumps configuration
Read
Write
Close

□ Pump channel configured to work with interface converter

Channel number in interface converter for given pump: 1

Pump channel configured to work with interface converter = false

Channel number in interface converter for given pump: = 1

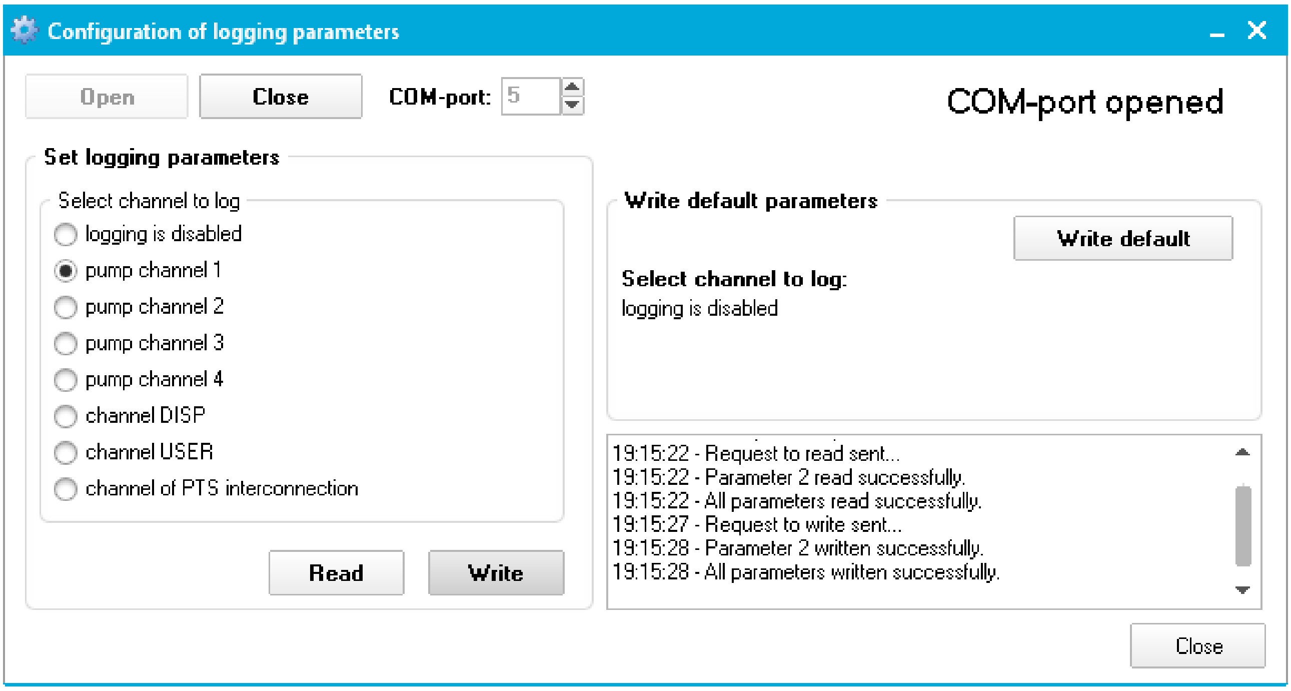
**Notice!**
After changing of the parameter value the controller will first response that request and response parameter values do not match because it needs restart in order to apply changes.

Read or write again parameter and make sure that the value is now set correctly!

```
14:36:27 - Parameter 49 read successfully.
14:36:27 - All parameters read successfully.
14:36:27 - Request to read sent...
14:36:27 - Parameter 49 read successfully.
14:36:27 - All parameters read successfully.
14:36:29 - Request to read sent...
14:36:29 - Parameter 49 read successfully.
14:36:29 - All parameters read successfully.
```

## *Configuration of parameters for logging*

Logging is one of the possibilities of PTS controller. It is described in section “Logging of communication” below.



COM-port: 5

COM-port opened

Select channel to log
○ logging is disabled
● pump channel 1
○ pump channel 2
○ pump channel 3
○ pump channel 4
○ channel DISP
○ channel USER
○ channel of PTS interconnection

**Select channel to log:**
logging is disabled

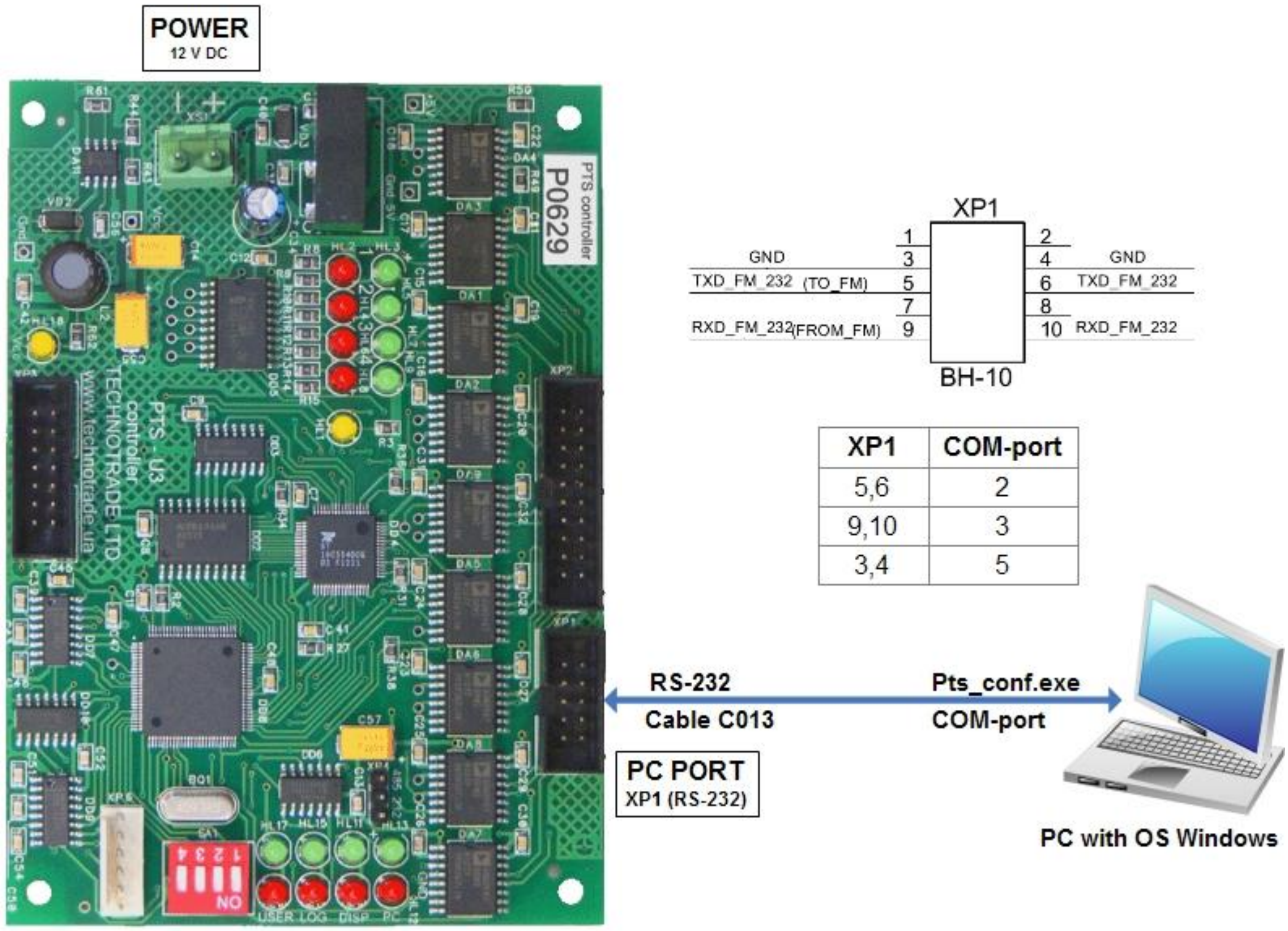
```
19:15:22 - Request to read sent...
19:15:22 - Parameter 2 read successfully.
19:15:22 - All parameters read successfully.
19:15:27 - Request to write sent...
19:15:28 - Parameter 2 written successfully.
19:15:28 - All parameters written successfully.
```

## *Update of the PTS controller firmware*

PTS controller firmware is constantly being improved and new versions of firmware with new added communication protocols and fixed bugs of the previous firmware versions are proposed to be applied. New versions of PTS controller's firmware are always available for downloading for customers.

PTS controller firmware update is made using a built-in updater in *Pts_conf.exe* utility.

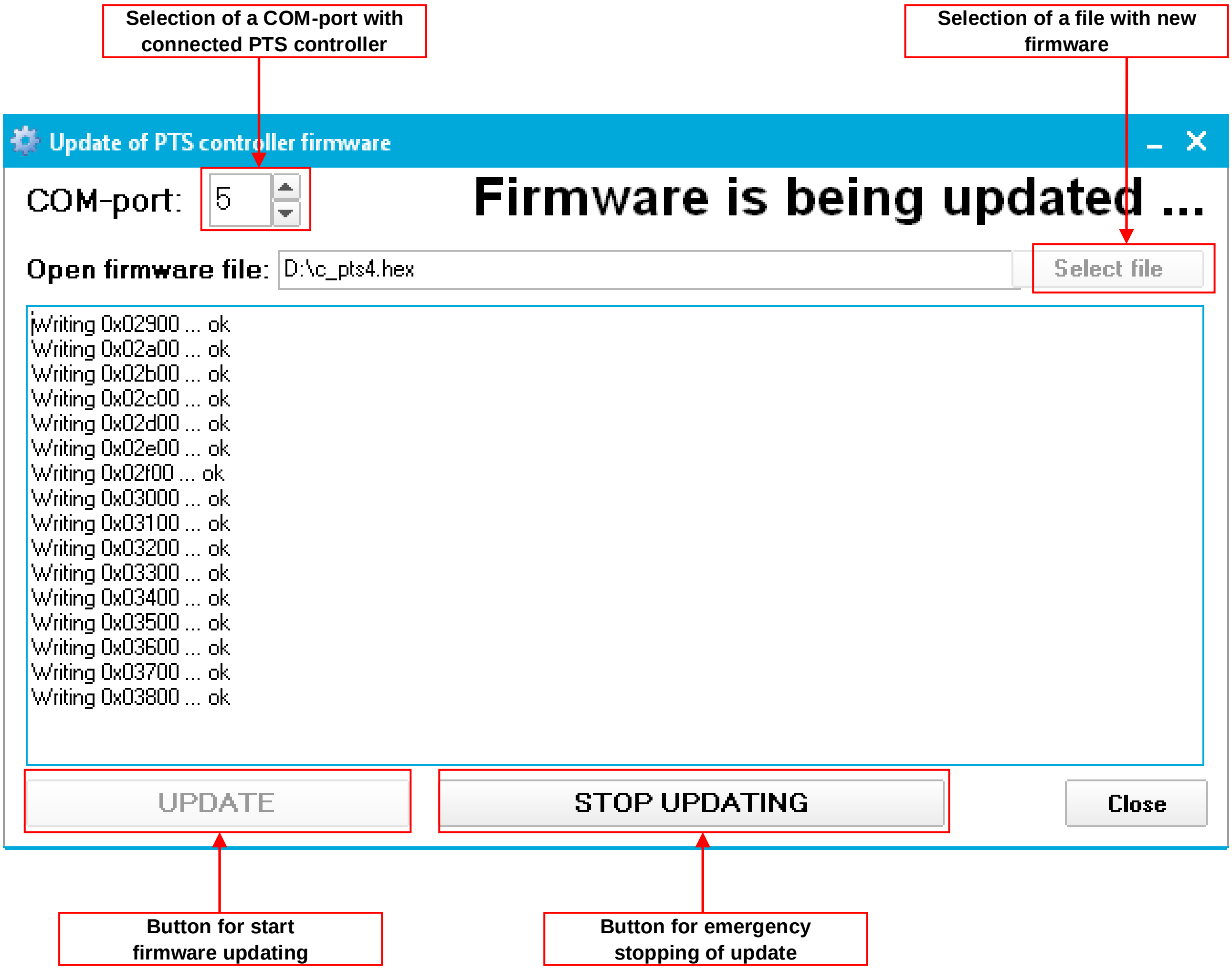
Scheme of connections of the PTS controller should be same as for configuration:



| XP1 | COM-port |
|---|---|
| 5,6 | 2 |
| 9,10 | 3 |
| 3,4 | 5 |

In *Pts_conf.exe* utility leave a COM-port closed. Form for updating of PTS controller firmware is called from main form main menu tab ‘Firmware update’ → ‘Firmware update’.

On the opened form select a COM-port of connected PTS controller, path to a file with a new firmware and press a button “UPDATE”. Firmware will start to be updated. Wait until the process is finished. In case of any errors - restart a tool and try again.


Selection of a COM-port with connected PTS controller
Selection of a file with new firmware
Update of PTS controller firmware
Firmware is being updated ...
Select file
UPDATE
STOP UPDATING
Close
Button for start firmware updating
Button for emergency stopping of update

COM-port: 5

Open firmware file: D:\c_pts4.hex

```
Writing 0x02900 ... ok
Writing 0x02a00 ... ok
Writing 0x02b00 ... ok
Writing 0x02c00 ... ok
Writing 0x02d00 ... ok
Writing 0x02e00 ... ok
Writing 0x02f00 ... ok
Writing 0x03000 ... ok
Writing 0x03100 ... ok
Writing 0x03200 ... ok
Writing 0x03300 ... ok
Writing 0x03400 ... ok
Writing 0x03500 ... ok
Writing 0x03600 ... ok
Writing 0x03700 ... ok
Writing 0x03800 ... ok
```

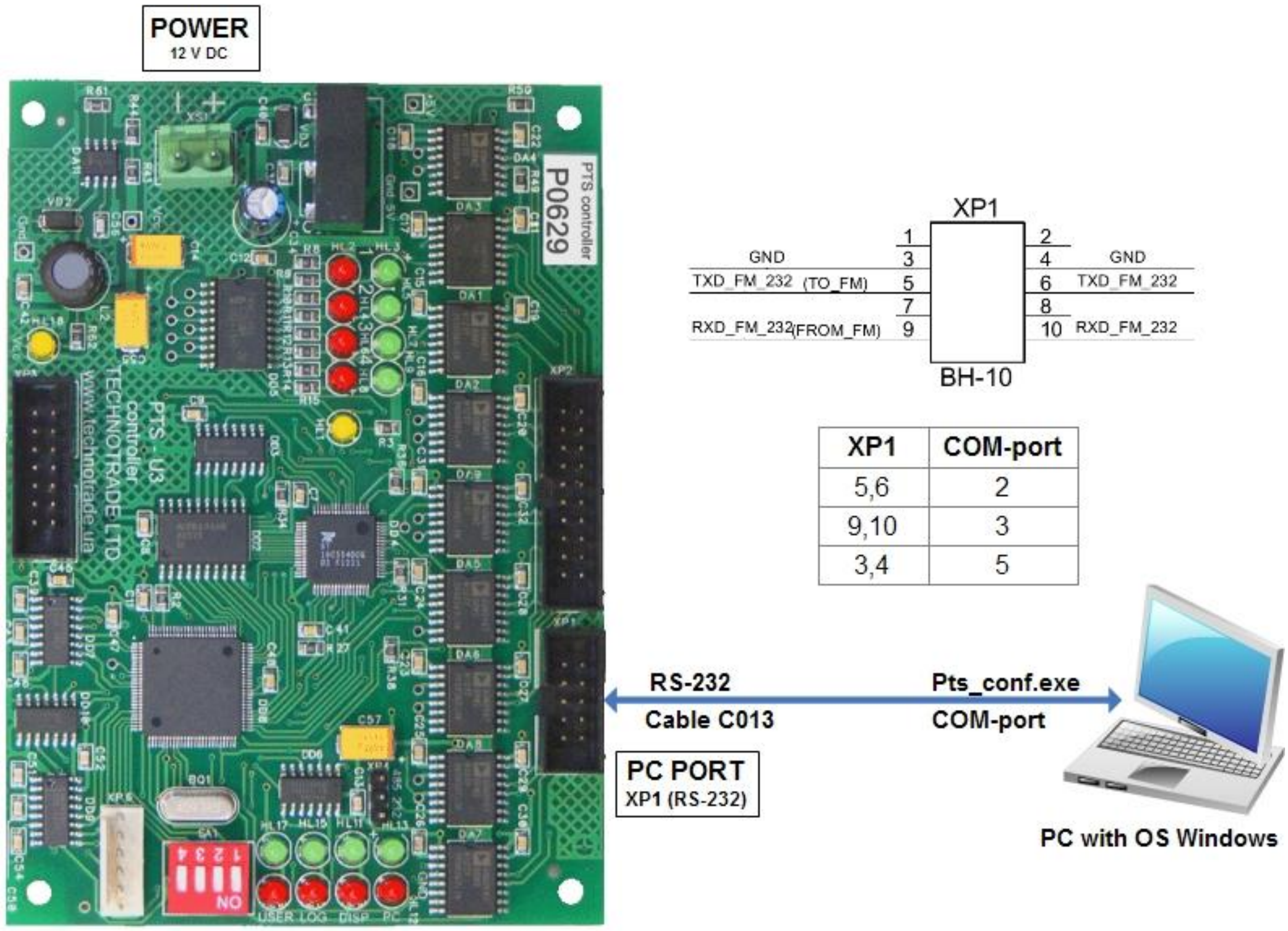
After the firmware update is completed - check PTS controller configuration. Some of the new firmwares (in case if it contains sufficient changes of PTS controller operation) may erase the PTS controller configuration.

Due to a reason that some firmware versions may erase all configuration of the PTS controller it recommended to save all configuration of PTS controller to file before making an update (see section ‘Saving of all configuration of PTS controller to file’) and in case if after the update the PTS controller configuration turns out to be erased – restore all the configuration from the previously saved file (see section ‘Loading of all configuration of PTS controller from file’).

## *Logging of PTS controller operation*

With a reason to quickly locate and remove possible bugs in PTS controller communication with connected equipment (fuel dispensers, ATG systems, other PTS controllers in PTS interconnection) PTS controller has a possibility of logging of its exchange with connected equipment. Use *Pts_conf.exe* utility to save in a control system exchange logs of PTS controller with connected equipment.

Before taking a log from PTS controller it is necessary to configure a channel, from which a log will be taken. Scheme of connections of the PTS controller should be same as for configuration:

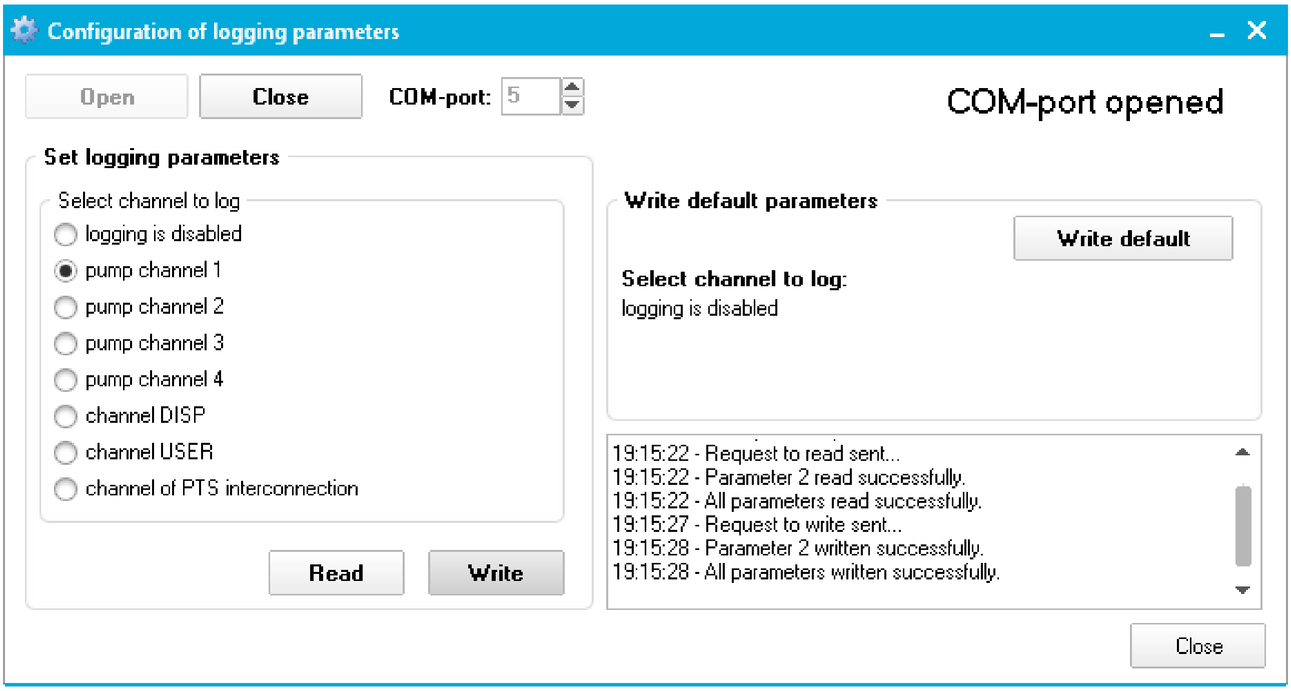


| XP1 | COM-port |
|---|---|
| 5,6 | 2 |
| 9,10 | 3 |
| 3,4 | 5 |

In *Pts_conf.exe* utility leave a COM-port closed. Form for logging is called from main form main menu tab ‘Logging’ → ‘Logging’.

On the opened form click on a button “Select a channel to log” to open a form of selection of the channel, from which a log will be taken.

On the opened form open a COM-port, to which PTS controller is connected and select a channel, from which a log will be taken. Click a button “Write” to save configuration and close a form.



COM-port opened

Set logging parameters

Select channel to log
- ○ logging is disabled
- ● pump channel 1
- ○ pump channel 2
- ○ pump channel 3
- ○ pump channel 4
- ○ channel DISP
- ○ channel USER
- ○ channel of PTS interconnection

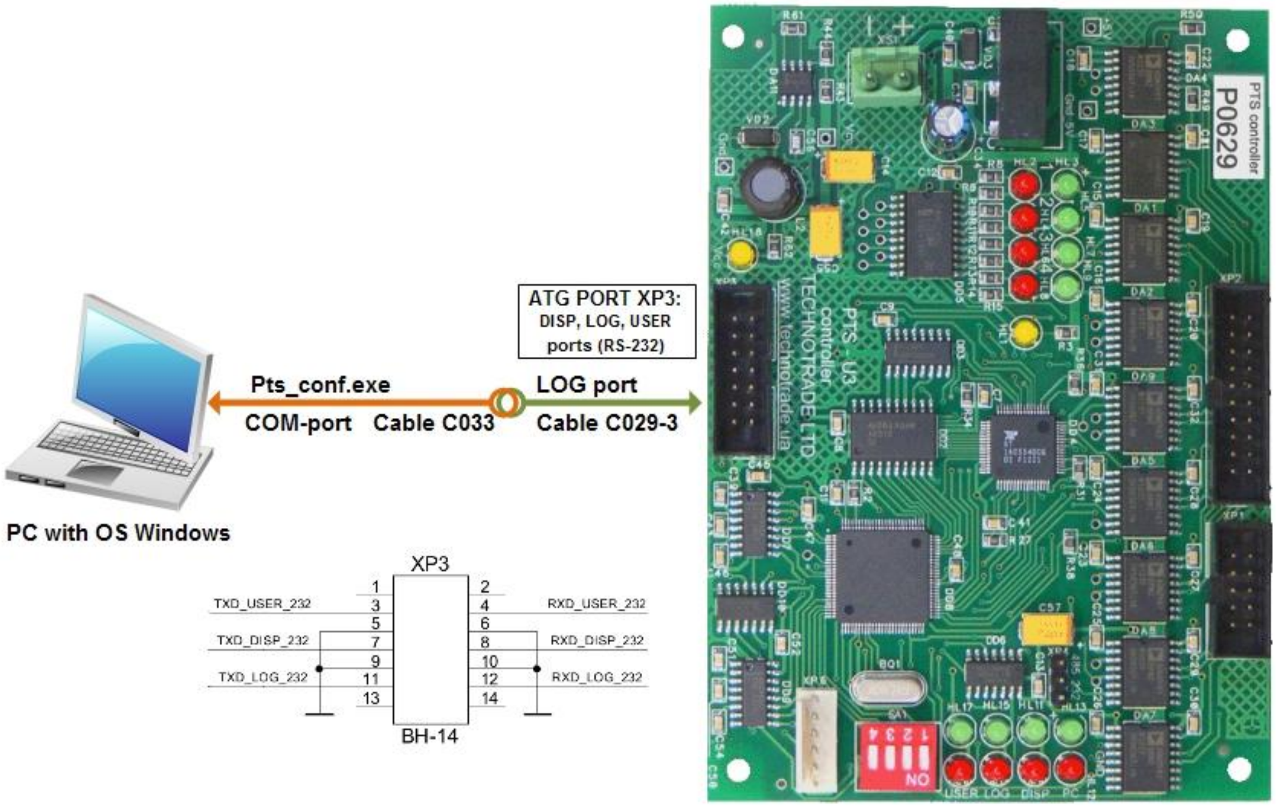
Write default parameters

Select channel to log:
logging is disabled

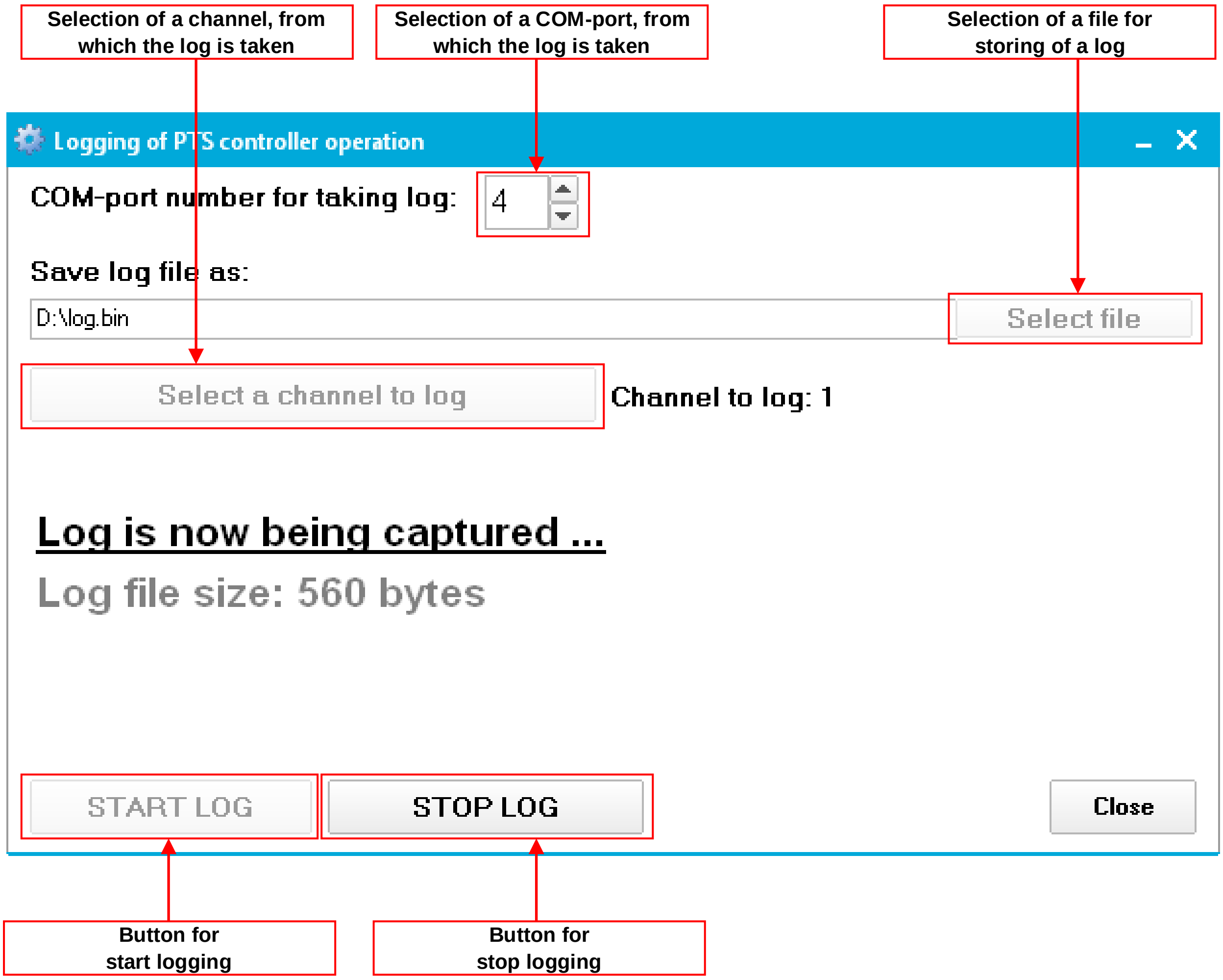
```
19:15:22 - Request to read sent...
19:15:22 - Parameter 2 read successfully.
19:15:22 - All parameters read successfully.
19:15:27 - Request to write sent...
19:15:28 - Parameter 2 written successfully.
19:15:28 - All parameters written successfully.
```

After the channel, from which a log will be taken, is configured it is possible to get a log from it. PTS controller sends a LOG to a LOG-port, so it is necessary to connect it to COM-port of the control system.

Scheme of connections is the following:



In *Pts_conf.exe* utility on form for logging it is necessary to select a COM-port, to which a LOG-port of PTS controller is connected and to select a .bin file, in which a log will be accumulated. After this clicking on a button “START LOG” will start saving of the log into the file, which size is indicated on the form. If the file size stays equal to zero – then log is being taken incorrectly. In case of any problem to take the log information about it will be shown on the screen.



Selection of a channel, from which the log is taken
Selection of a COM-port, from which the log is taken
Selection of a file for storing of a log
Logging of PTS controller operation
Select file
Select a channel to log
START LOG
STOP LOG
Close
Button for start logging
Button for stop logging

COM-port number for taking log: 4

Save log file as:

D:\log.bin

Channel to log: 1

Log is now being captured ...

Log file size: 560 bytes

Log will be accumulated to the selected file. As an evidence of it logged data size of the accumulated file will be displayed on the screen. To stop logging press a button ”STOP LOG”. In case of any errors – restart a tool and try again.

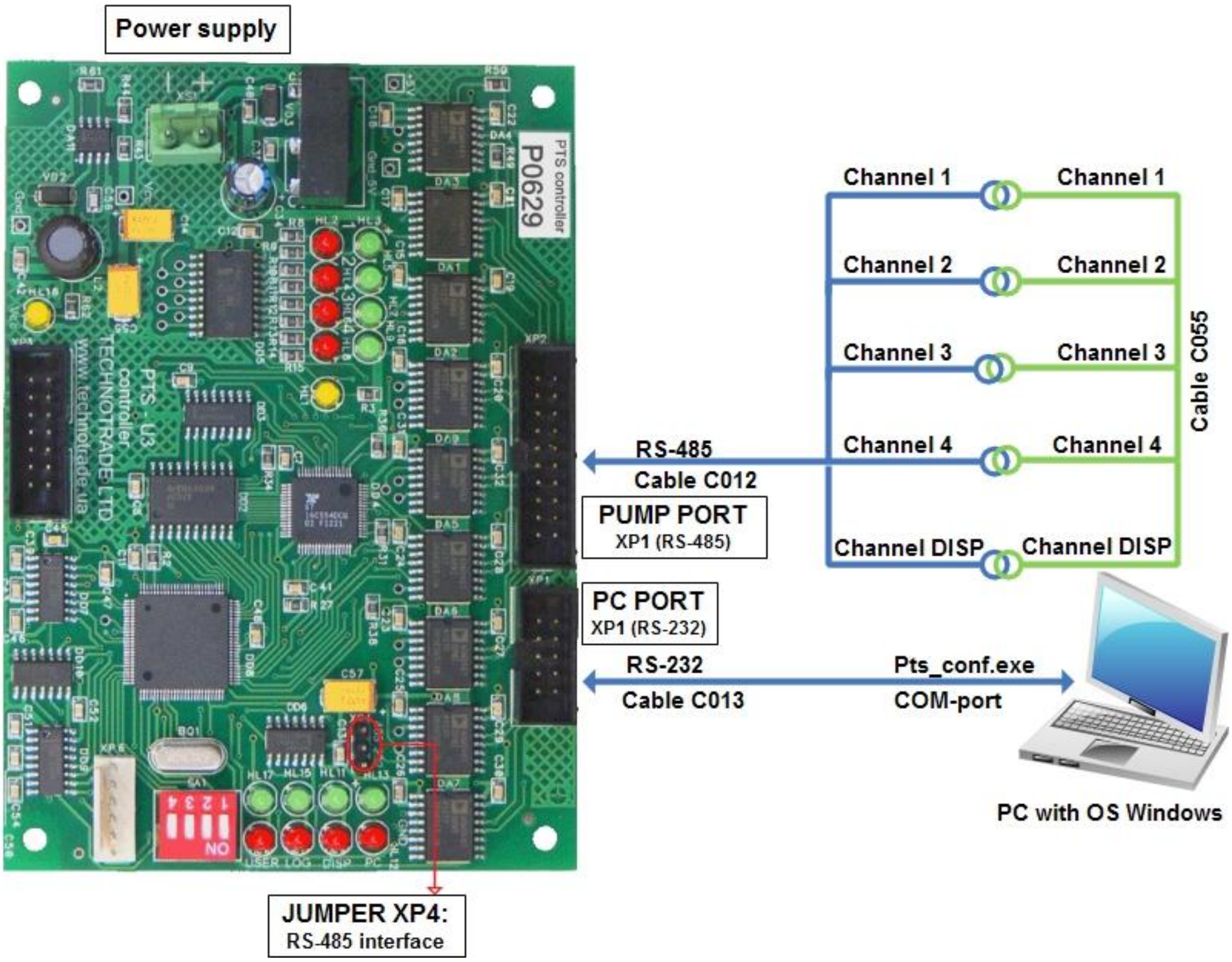
Log in transferred in the encrypted form. After a log of exchange is taken it is required to pass a received .bin file to “TECHNOTRADE LTD” for examining and elimination of possible problems, for reasons of which it was taken.

## *Self-diagnostics of PTS controller interfaces and DIP-switch*

With a reason to quickly provide diagnostics of PTS controller's RS-485 and RS-232 interfaces and DIP-switches a special diagnostics of PTS controller is used. Use *Pts_conf.exe* utility to provide diagnostics of PTS controller.

In *Pts_conf.exe* utility open a COM-port with connected PTS controller. Form for PTS controller diagnostics is called from main form main menu tab 'Diagnostics' → 'Self-diagnostics'.

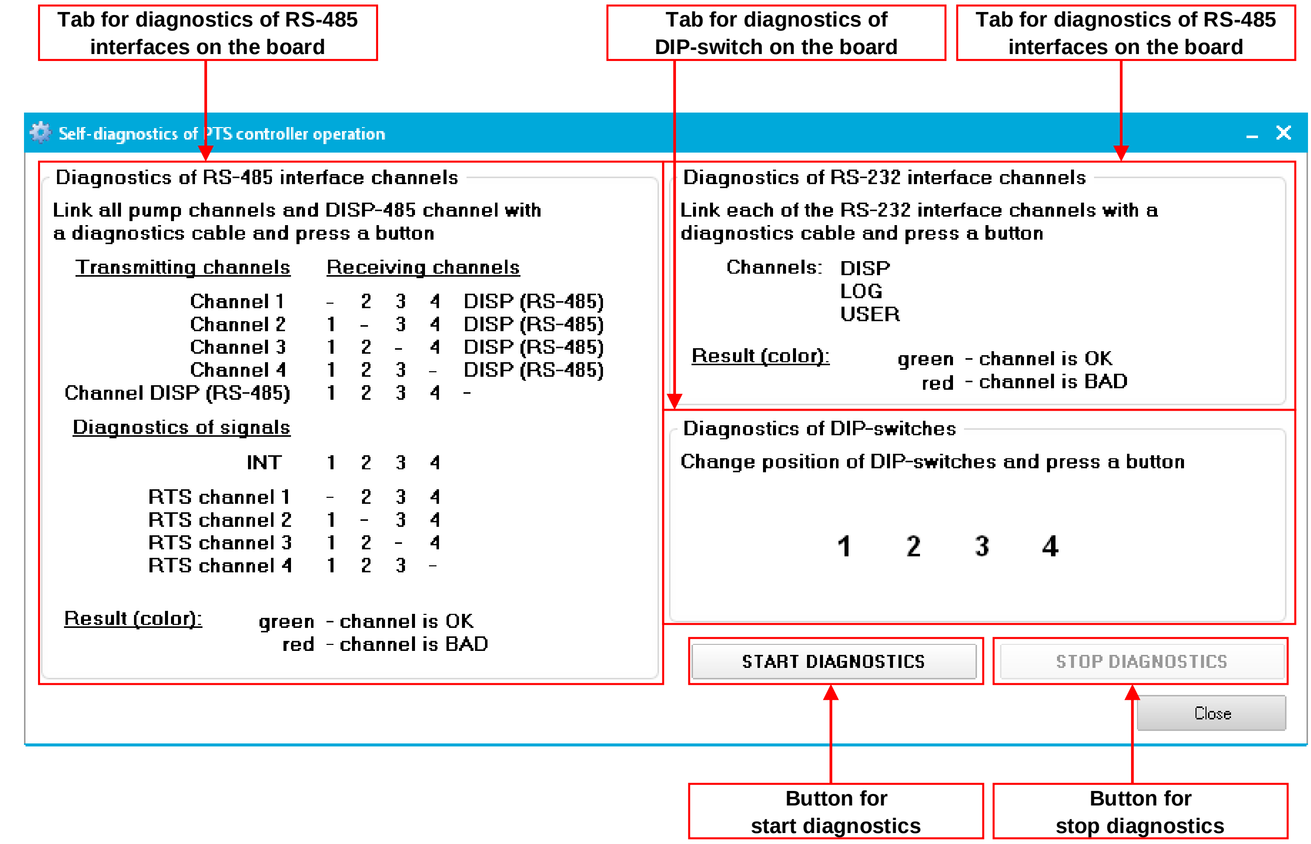
Diagnostics of 485 interfaces:



Configuration:

In order to check RS-485 interfaces connect all RS-485 connectors of C012 cable with appropriate connectors of diagnostics cable C055 (connector X1 to connector X1, X2 – to X2, X3 – to X3 and so on). Close XP4 jumper on the board of PTS controller to position RS-485.

On form for PTS controller self-diagnostics press a button "START DIAGNOSTICS" in order to start self-diagnostics. On the tab "Diagnostics of RS-485 interface channels" results of diagnostics will be shown by color of labels. Green color means that correspondent channel is working correctly (OK), red color – correspondent channel is working incorrectly (BAD).



Tab for diagnostics of RS-485 interfaces on the board

Tab for diagnostics of DIP-switch on the board

Tab for diagnostics of RS-485 interfaces on the board

Self-diagnostics of PTS controller operation

Diagnostics of RS-485 interface channels

Link all pump channels and DISP-485 channel with a diagnostics cable and press a button

| Transmitting channels | Receiving channels | | | | |
|---|---|---|---|---|---|
| Channel 1 | - | 2 | 3 | 4 | DISP (RS-485) |
| Channel 2 | 1 | - | 3 | 4 | DISP (RS-485) |
| Channel 3 | 1 | 2 | - | 4 | DISP (RS-485) |
| Channel 4 | 1 | 2 | 3 | - | DISP (RS-485) |
| Channel DISP (RS-485) | 1 | 2 | 3 | 4 | - |

Diagnostics of signals

| | | | | |
|---|---|---|---|---|
| INT | 1 | 2 | 3 | 4 |
| RTS channel 1 | - | 2 | 3 | 4 |
| RTS channel 2 | 1 | - | 3 | 4 |
| RTS channel 3 | 1 | 2 | - | 4 |
| RTS channel 4 | 1 | 2 | 3 | - |

Result (color): green - channel is OK
red - channel is BAD

Diagnostics of RS-232 interface channels

Link each of the RS-232 interface channels with a diagnostics cable and press a button

Channels: DISP
LOG
USER

Result (color): green - channel is OK
red - channel is BAD

Diagnostics of DIP-switches

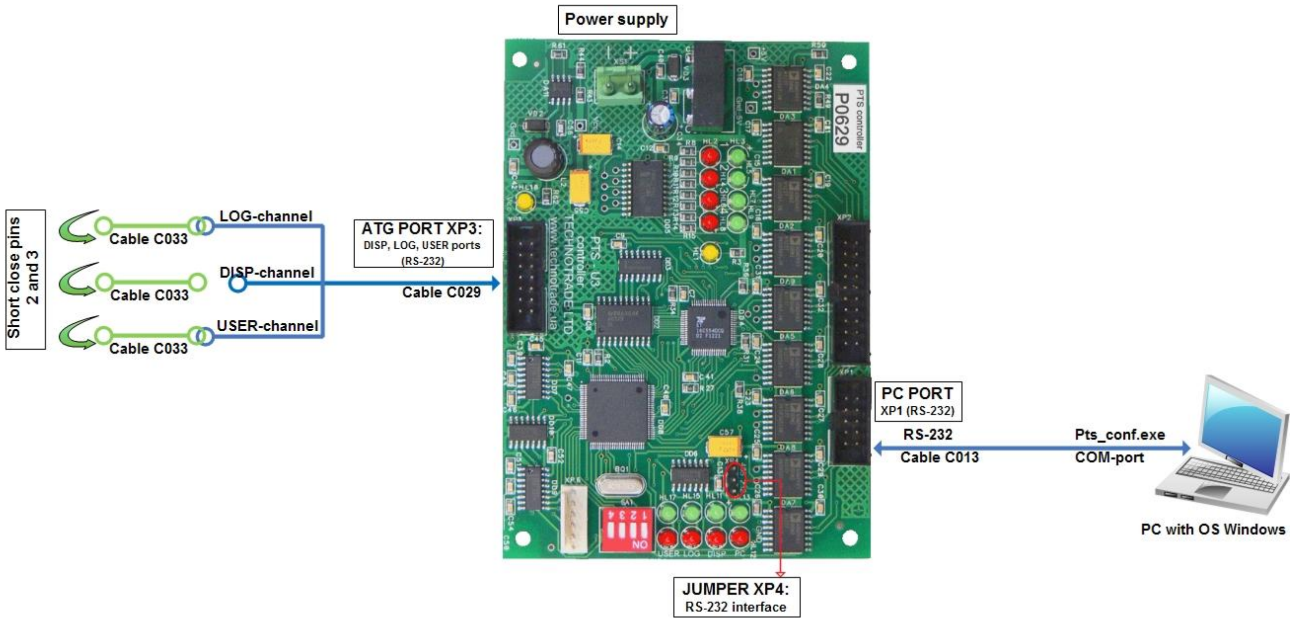
Change position of DIP-switches and press a button

1 2 3 4

Button for start diagnostics

Button for stop diagnostics

Diagnostics of 232 interfaces:



In order to check RS-232 interfaces (ports USER, LOG, DISP on cable C029 or on cable C033, which is connected to cable C029) connect pins 2 (Tx) with pins 3 (Rx) on each of these ports. Close XP4 jumper on the board of PTS controller to position RS-232. On the tab “Diagnostics of RS-232 interface channels” results of diagnostics will be shown by color of labels. Green color means that correspondent channel is working correctly (OK), red color – correspondent channel is working incorrectly (BAD).

Diagnostics of DIP-switch:

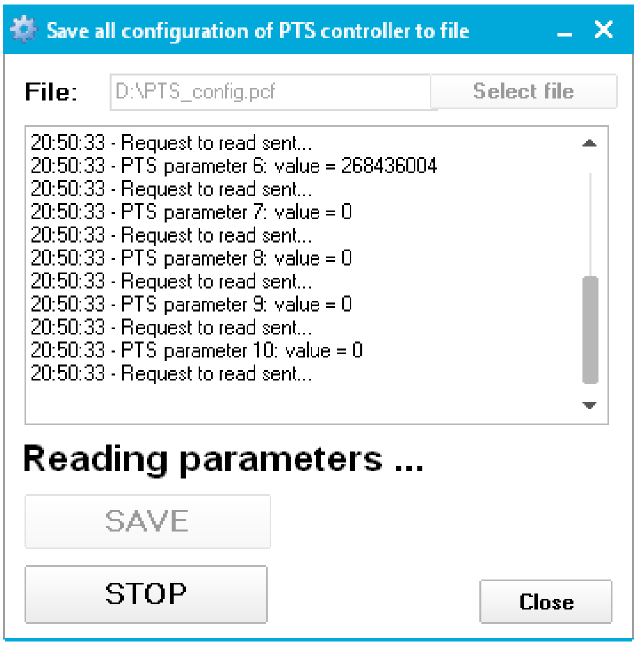
Tab “Diagnostics of DIP-switches” will display current position of switches in a DIP-switch on PTS board. If displayed position of switches correspond to real position of switches in a DIP-switch on PTS board after checking them in various positions – then DIP switch is working correctly.

## *Saving of all configuration of PTS controller to file*

Form for saving of all configuration of PTS controller (pumps channels configuration, ATG systems channels configuration and all parameters) is called from main form main menu tab 'File' → 'Save all configuration of PTS controller to file'.

On the opened form it is necessary to specify a file, to which configuration of PTS controller will be saved, and click "Save" button. The utility will save one by one:

- pumps channels configuration
- ATG systems channels configuration
- parameters



File: D:\PTS_config.pcf

```
20:50:33 - Request to read sent...
20:50:33 - PTS parameter 6: value = 268436004
20:50:33 - Request to read sent...
20:50:33 - PTS parameter 7: value = 0
20:50:33 - Request to read sent...
20:50:33 - PTS parameter 8: value = 0
20:50:33 - Request to read sent...
20:50:33 - PTS parameter 9: value = 0
20:50:33 - Request to read sent...
20:50:33 - PTS parameter 10: value = 0
20:50:33 - Request to read sent...
```

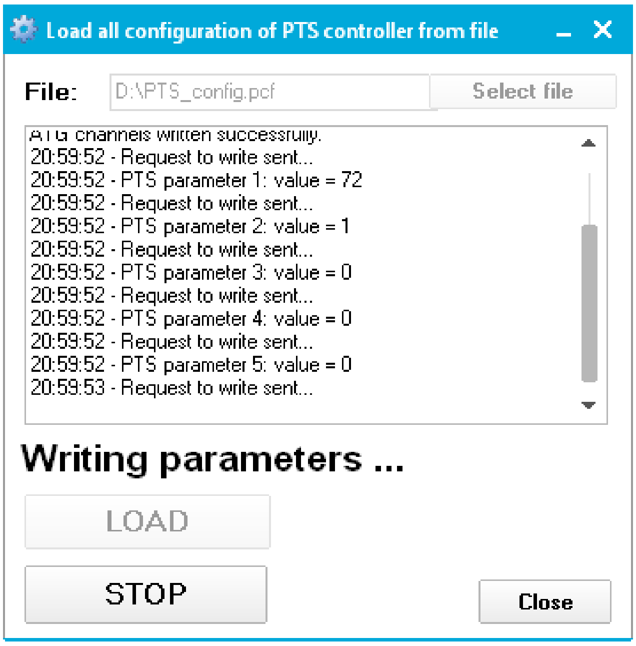
Reading parameters ...

## *Loading of all configuration of PTS controller from file*

Form for loading of all configuration of PTS controller (pumps channels configuration, ATG systems channels configuration and all parameters) is called from main form main menu tab 'File' → 'Load all configuration of PTS controller from file'.

On the opened form it is necessary to specify a file, from which configuration of PTS controller will be loaded, and click "Load" button. The utility will load one by one:

- pumps channels configuration
- ATG systems channels configuration
- parameters



File: D:\PTS_config.pcf

```
ATG channels written successfully.
20:59:52 - Request to write sent...
20:59:52 - PTS parameter 1: value = 72
20:59:52 - Request to write sent...
20:59:52 - PTS parameter 2: value = 1
20:59:52 - Request to write sent...
20:59:52 - PTS parameter 3: value = 0
20:59:52 - Request to write sent...
20:59:52 - PTS parameter 4: value = 0
20:59:52 - Request to write sent...
20:59:52 - PTS parameter 5: value = 0
20:59:53 - Request to write sent...
```

Writing parameters ...

# *Additional possibilities of PTS controller configuration utility*

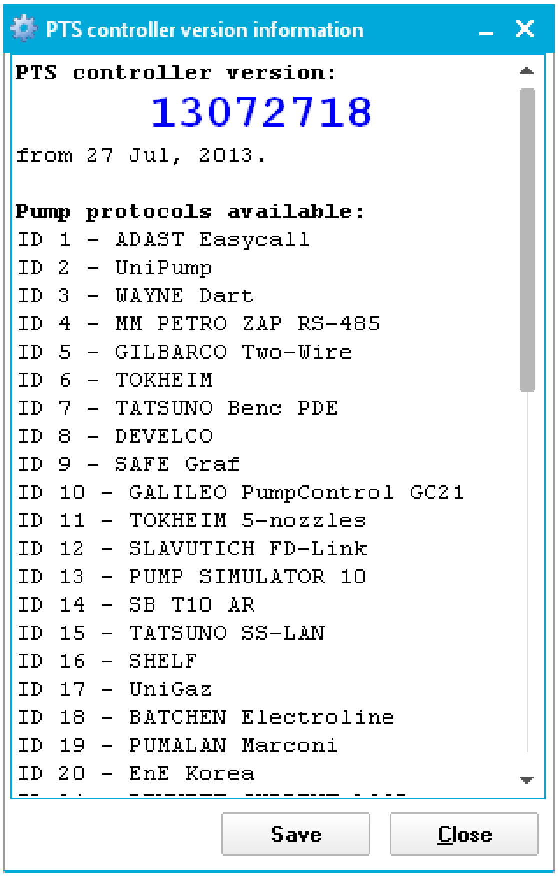
## *Reading of PTS controller firmware version*

*Pts_conf.exe* utility is able to read current firmware version from the PTS controller and also all supported communication protocols with fuel dispensers and ATG systems.

In order to make it on the main form of the utility open a COM-port, to which PTS controller is connected and click a button "Check firmware version".

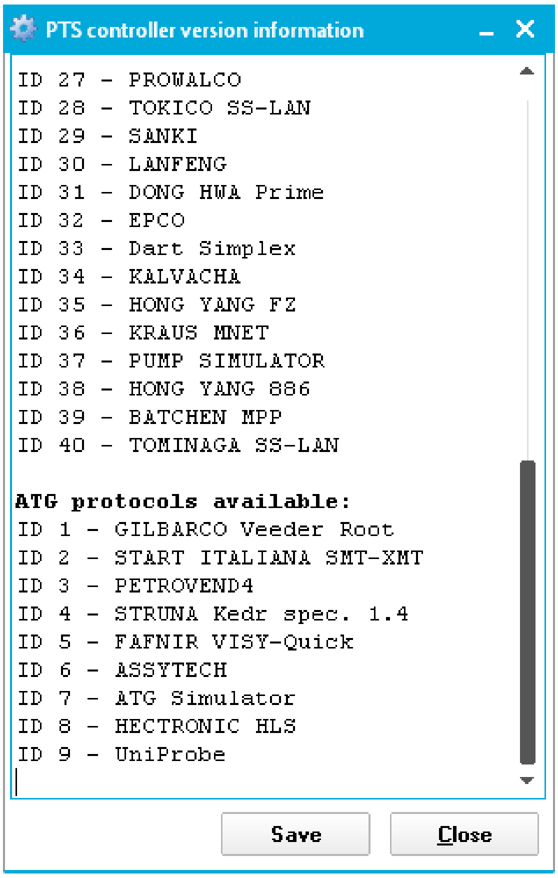
Opened form will contain information about:

- date of firmware release (including number of release)
- list of supported communication protocols with fuel dispensers in given firmware
- list of supported communication protocols with ATG systems in given firmware



```
PTS controller version:
        13072718
from 27 Jul, 2013.

Pump protocols available:
ID 1 - ADAST Easycall
ID 2 - UniPump
ID 3 - WAYNE Dart
ID 4 - MM PETRO ZAP RS-485
ID 5 - GILBARCO Two-Wire
ID 6 - TOKHEIM
ID 7 - TATSUNO Benc PDE
ID 8 - DEVELCO
ID 9 - SAFE Graf
ID 10 - GALILEO PumpControl GC21
ID 11 - TOKHEIM 5-nozzles
ID 12 - SLAVUTICH FD-Link
ID 13 - PUMP SIMULATOR 10
ID 14 - SB T10 AR
ID 15 - TATSUNO SS-LAN
ID 16 - SHELF
ID 17 - UniGaz
ID 18 - BATCHEN Electroline
ID 19 - PUMALAN Marconi
ID 20 - EnE Korea
```

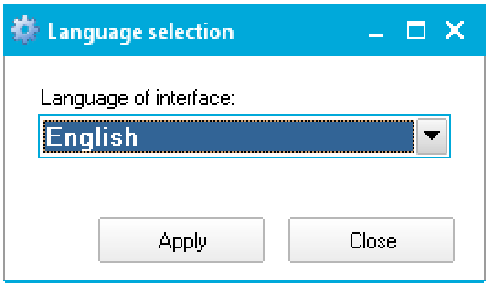


```
ID 27 - PROWALCO
ID 28 - TOKICO SS-LAN
ID 29 - SANKI
ID 30 - LANFENG
ID 31 - DONG HWA Prime
ID 32 - EPCO
ID 33 - Dart Simplex
ID 34 - KALVACHA
ID 35 - HONG YANG FZ
ID 36 - KRAUS MNET
ID 37 - PUMP SIMULATOR
ID 38 - HONG YANG 886
ID 39 - BATCHEN MPP
ID 40 - TOMINAGA SS-LAN

ATG protocols available:
ID 1 - GILBARCO Veeder Root
ID 2 - START ITALIANA SMT-XMT
ID 3 - PETROVEND4
ID 4 - STRUNA Kedr spec. 1.4
ID 5 - FAFNIR VISY-Quick
ID 6 - ASSYTECH
ID 7 - ATG Simulator
ID 8 - HECTRONIC HLS
ID 9 - UniProbe
```

## *Setting of PTS controller configuration utility language*

PTS controller configuration utility is supports multiple languages and is easily localizable to new languages. Form for selection of language of user interface is called from main form main menu tab 'Adjustments' → 'Language selection'. For calling of this form COM-port on main form should be closed.

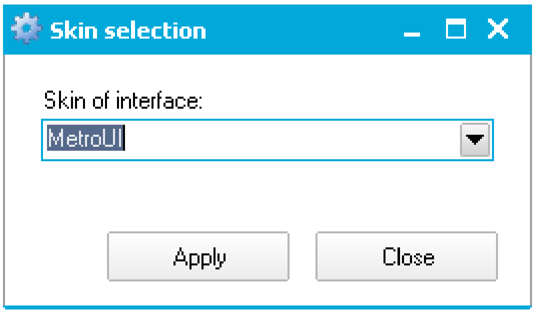


Language of interface: English

Adding of a new language to the utility is also easy. All language phrases are located in a file lang.xml, which is located in root folder of *Pts_conf.exe* utility. Adding a new translation of the phrases to this file will add a new language to the utility.

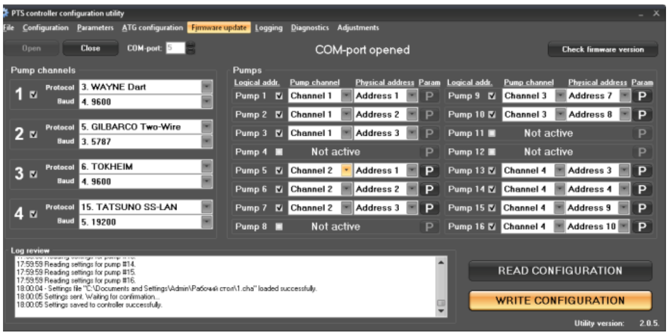
## *Setting of PTS controller configuration utility skin for graphical user interface*

PTS controller configuration utility in order to be used together with various POS systems and applications has a possibility to change its skins to suit to various graphical user interfaces. Selection of the skin is made form main menu tab ‘Adjustments’ → ‘Skin selection’:

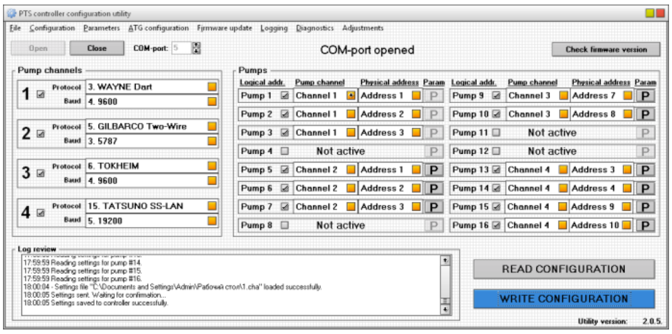


Skin of interface: MetroUI

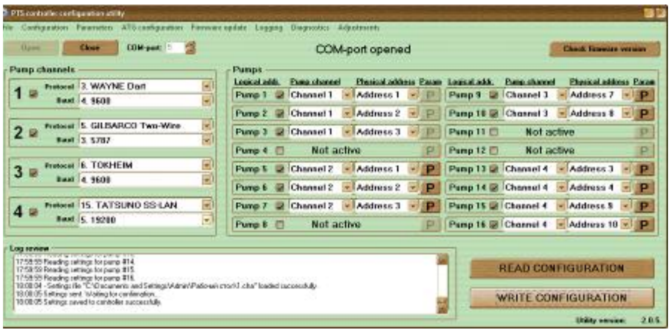
Examples of skins:



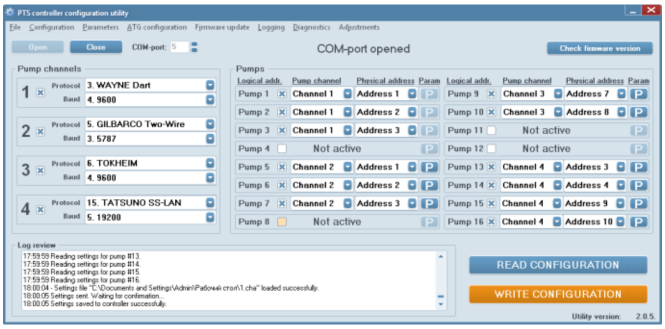
Office 2007 Black



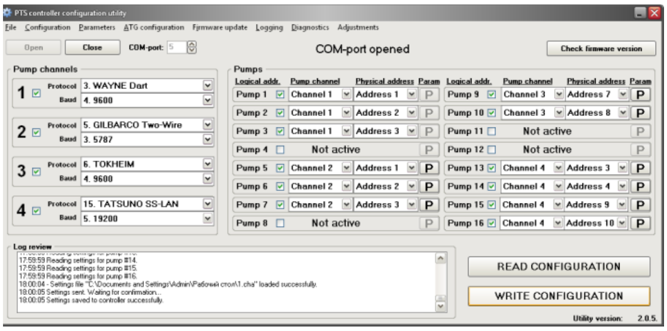
Neutral 3



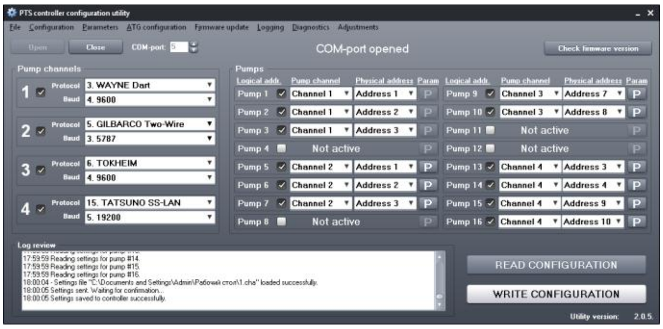
Retro



WEB2



LongHorn



iOS dark

## *Logging of Pts_conf utility operations*

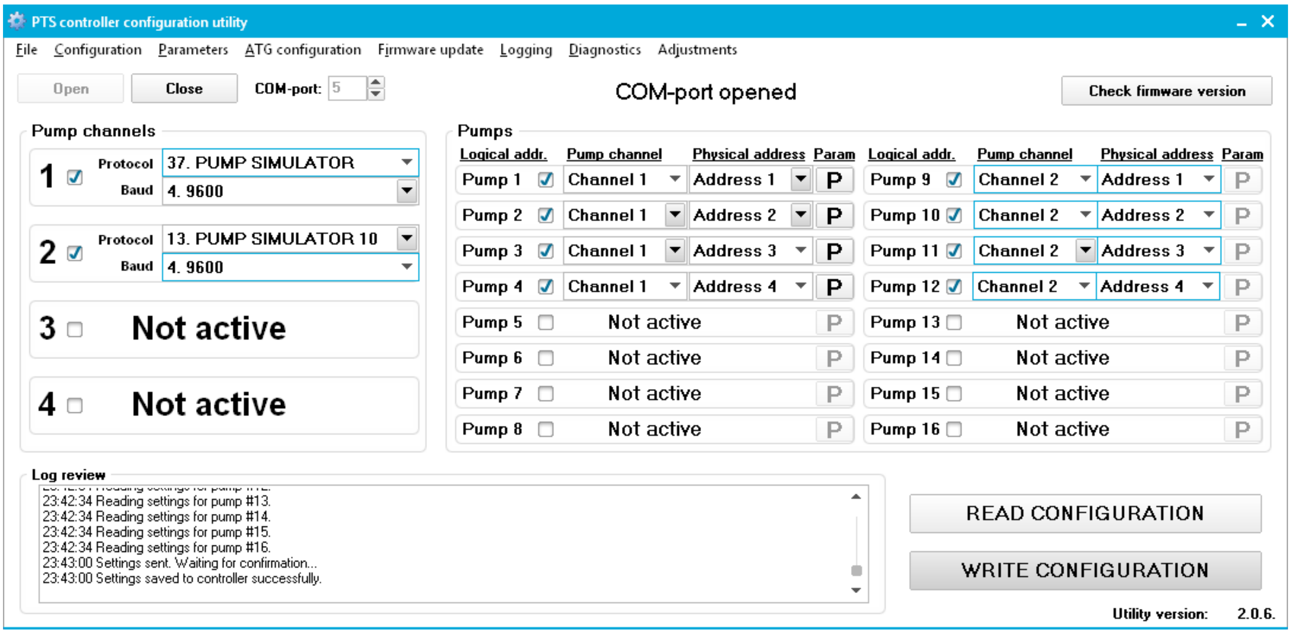
With a reason to give developers a possibility to show what commands PTS controller configuration utility sends to PTS controller the utility has an option for saving of a log of all its actions to a file. This option is configured in main menu in tab “Adjustments” → “Write log of operations” (possibility to switch it on and off).

# BUILT-IN PUMP SIMULATOR

In order to make easy debugging of control systems without a fuel dispenser connected to PTS controller a simulator of fuel dispenser is present in PTS controller firmware. There are 2 pump simulators present:

- pump protocol “13. PUMP SIMULATOR 10”
- pump protocol “37. PUMP SIMULATOR”

When any of these protocols is selected in configuration of the PTS controller – baud rate can be set any.


PTS controller configuration utility
File Configuration Parameters ATG configuration Firmware update Logging Diagnostics Adjustments
Open
Close
Check firmware version
READ CONFIGURATION
WRITE CONFIGURATION

COM-port: 5

COM-port opened

Pump channels

| | | |
|---|---|---|
| 1 ☑ | Protocol | 37. PUMP SIMULATOR |
| | Baud | 4. 9600 |
| 2 ☑ | Protocol | 13. PUMP SIMULATOR 10 |
| | Baud | 4. 9600 |
| 3 ☐ | Not active | |
| 4 ☐ | Not active | |

Pumps

| Logical addr. | Pump channel | Physical address | Param | Logical addr. | Pump channel | Physical address | Param |
|---|---|---|---|---|---|---|---|
| Pump 1 ☑ | Channel 1 | Address 1 | P | Pump 9 ☑ | Channel 2 | Address 1 | P |
| Pump 2 ☑ | Channel 1 | Address 2 | P | Pump 10 ☑ | Channel 2 | Address 2 | P |
| Pump 3 ☑ | Channel 1 | Address 3 | P | Pump 11 ☑ | Channel 2 | Address 3 | P |
| Pump 4 ☑ | Channel 1 | Address 4 | P | Pump 12 ☑ | Channel 2 | Address 4 | P |
| Pump 5 ☐ | Not active | | P | Pump 13 ☐ | Not active | | P |
| Pump 6 ☐ | Not active | | P | Pump 14 ☐ | Not active | | P |
| Pump 7 ☐ | Not active | | P | Pump 15 ☐ | Not active | | P |
| Pump 8 ☐ | Not active | | P | Pump 16 ☐ | Not active | | P |

Log review

```
23:42:34 Reading settings for pump #13.
23:42:34 Reading settings for pump #14.
23:42:34 Reading settings for pump #15.
23:42:34 Reading settings for pump #16.
23:43:00 Settings sent. Waiting for confirmation...
23:43:00 Settings saved to controller successfully.
```

Utility version: 2.0.6.

Purpose of the pump simulator protocol is to help developers in debugging of the control system software over PTS controller at absence of real dispenser pumphead for connection or software simulators of fuel dispenser.

When protocol “37. PUMP SIMULATOR” is selected then for pump with logical address 1 position of first 4 nozzles is selected using a DIP switch on the PTS controller board (DIP-switch has 4 switches: switch 1 is responsible for position of nozzle 1, switch 2 – for nozzle 2 and so on). For pumps with logical addresses 2 – 16 the pumps always have nozzle 1 up (nozzle can not be placed down).

In protocol “13. PUMP SIMULATOR 10” all pumps always have nozzle1 up.

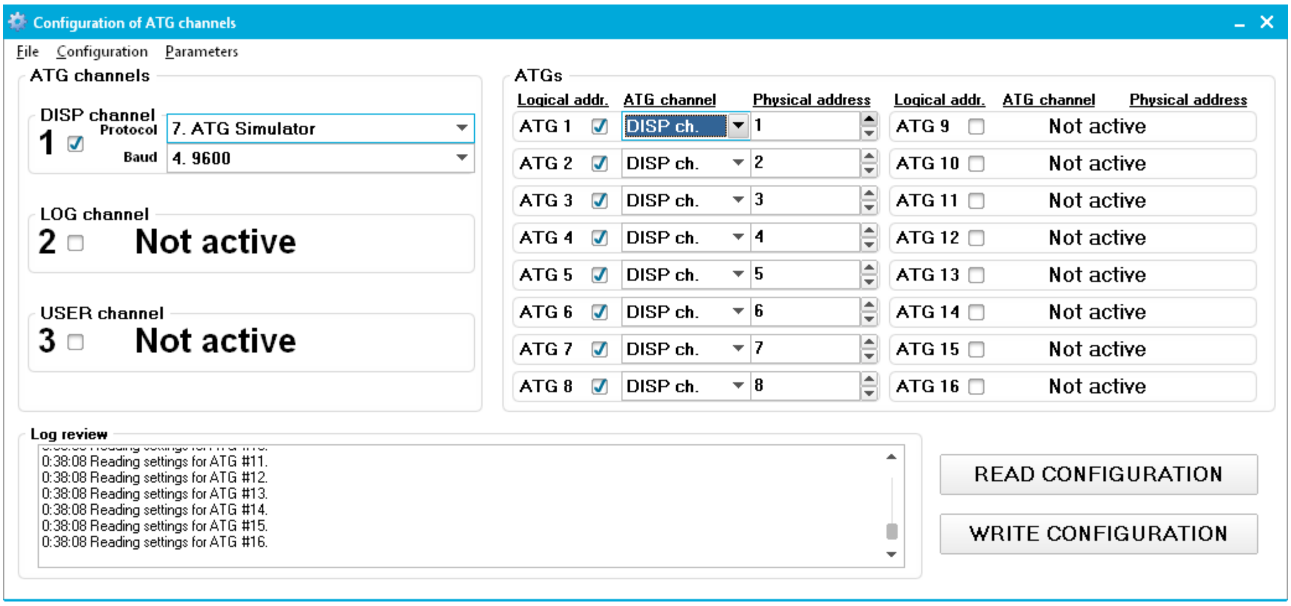
Pump simulators allow to:

- authorize a dispenser with preset volume or amount values
- simulate dispensing of fuel through this dispenser and stop it at necessity
- set prices to dispenser and get prices from dispenser
- read total counters values (both amount and volume) (total counters are lead on each of the nozzles separately and are dropped to zero at restart of the PTS controller)

Difference of protocol “13. PUMP SIMULATOR 10” from protocol “37. PUMP SIMULATOR” is that when an order is given in protocol “37. PUMP SIMULATOR” – dispensing is made from zero till the ordered dose. When an order is given in protocol “13. PUMP SIMULATOR 10” – then dispensing is made from ordered dose minus ten liters till the ordered dose, so always only last ten liters are dispensed (this is useful for purposes when there is a need to make quick void sales for book keeping account at control systems).

# BUILT-IN ATG PROBE SIMULATOR

PTS controller firmware between its ATG probes protocols has a protocol “7 ATG SIMULATOR”, which allows to simulate presence of connected ATG probe. Baud rate at this can be set to any possible.



ATG channels

DISP channel
1 ☑ Protocol 7. ATG Simulator
Baud 4. 9600

LOG channel
2 □ Not active

USER channel
3 □ Not active

ATGs

| Logical addr. | ATG channel | Physical address | Logical addr. | ATG channel | Physical address |
|---|---|---|---|---|---|
| ATG 1 ☑ | DISP ch. | 1 | ATG 9 □ | Not active | |
| ATG 2 ☑ | DISP ch. | 2 | ATG 10 □ | Not active | |
| ATG 3 ☑ | DISP ch. | 3 | ATG 11 □ | Not active | |
| ATG 4 ☑ | DISP ch. | 4 | ATG 12 □ | Not active | |
| ATG 5 ☑ | DISP ch. | 5 | ATG 13 □ | Not active | |
| ATG 6 ☑ | DISP ch. | 6 | ATG 14 □ | Not active | |
| ATG 7 ☑ | DISP ch. | 7 | ATG 15 □ | Not active | |
| ATG 8 ☑ | DISP ch. | 8 | ATG 16 □ | Not active | |

Log review

```
0:38:08 Reading settings for ATG #11.
0:38:08 Reading settings for ATG #12.
0:38:08 Reading settings for ATG #13.
0:38:08 Reading settings for ATG #14.
0:38:08 Reading settings for ATG #15.
0:38:08 Reading settings for ATG #16.
```

At this depending on the selected address of the ATG probe output measurement values will be the following:

For ATG probes with logical address 1 – dynamically changing all measurement parameters looping in the following sequence (measurement numbers coming from 1 to 9 and back from 9 to 1):

| | | Measurement number | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Measurement parameter | Product level, mm | 1000 | 1250 | 1500 | 1750 | 2000 | 2250 | 2500 | 2750 | 3000 |
| | Water level, mm | 100 | 125 | 150 | 175 | 200 | 225 | 250 | 275 | 300 |
| | Product volume, l | 8000 | 10000 | 12000 | 14000 | 16000 | 18000 | 20000 | 22000 | 24000 |
| | Water volume, l | 800 | 1000 | 1200 | 1400 | 1600 | 1800 | 2000 | 2200 | 2400 |
| | Temperature, deg. C | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| | Product temperature compensated volume, l | 8500 | 10500 | 12500 | 14500 | 16000 | 17500 | 19500 | 21500 | 23500 |
| | Product ullage, l | 22000 | 20000 | 18000 | 16000 | 14000 | 12000 | 10000 | 8000 | 6000 |
| | Product density, kg/m$^3$ | 720 | 730 | 740 | 750 | 760 | 770 | 780 | 790 | 800 |
| | Product mass, kg | 5760 | 7300 | 8880 | 10500 | 12160 | 13860 | 15600 | 17380 | 19200 |

Duration between changes equals 2 seconds.

For ATG probes with logical addresses 2-16: statically fixed measurement values:

- product level – logical address + 123.4 mm (example - for address 6: 6123.4 mm)
- water level – logical address + 56.7 mm (example - for address 9: 956.7 mm)

temperature – logical address + .8 deg. C (example - for address 12: 12.8 deg. C)

# APPLICATION PROGRAMMING INTERFACES (API)

PTS controller for developers is supplied together with rich set of API (application programming interfaces) for mostly used programming languages and operation platforms:

- **C and C++:** API for communication with a PTS controller through a COM-port, documentation and examples of application in source codes under Windows OS and Linux operation systems
- **C#:**
    - open source library, which includes classes, methods and properties for communication with PTS controller for provision of control over dispensers and ATG systems and configuration of PTS controller
    - open-source utility for testing and indication of all operations provided by the PTS controller, allows to perform all operations with the PTS controller
    - open source fully-functional application for provision of control over up to 8 pumps and 4 ATG systems by the PTS controller, allows to perform all operations with the PTS controller
- **Delphi:** API for communication with a PTS controller through a COM-port, documentation and examples of application in source codes
- **Java:**
    - open source library, which includes classes, methods and properties for communication with PTS controller for provision of control over dispensers and ATG systems and configuration of PTS controller
    - open-source utility for testing and indication of all operations provided by the PTS controller, allows to perform all operations with the PTS controller
- **Python:** API for communication with a PTS controller through a COM-port, documentation and examples of application in source codes
- **Visual Basic .NET:**
    - open source library, which includes classes, methods and properties for communication with PTS controller for provision of control over dispensers and ATG systems and configuration of PTS controller
    - open-source utility for testing and indication of all operations provided by the PTS controller, allows to perform all operations with the PTS controller
    - open source fully-functional application for provision of control over up to 8 pumps and 4 ATG systems by the PTS controller, allows to perform all operations with the PTS controller
- **COM-object for PTS controller:**
    - open-source COM-object (written in Microsoft Visual C++ 6.0 IDE in C++ programming language) with methods and properties for communication with PTS controller for provision of control over dispensers and ATG systems and configuration of PTS controller
    - open-source application in Microsoft Visual Basic 6.0 for testing and indication of all operations provided by the PTS controller, allows to perform all operations with the PTS controller
- **Description of communication protocol of PTS controller:** describes communication parameters, messages structure, commands and responses of communication between POS management system and PTS controller, typical flowchart and diagram of interoperation

## *C and C++ API*

API represents a set of for communication with a PTS controller through a COM-port, documentation and examples of application in source codes under Windows OS and Linux operation systems

API can be compiled as a static or dynamic library. The library can be compiled by different compilers for different operating systems: POSIX compatible (tested with GNU Linux only) and Microsoft Windows.

Source code of library are located in ./src directory.
Headers of library located in ./include directory.
Directoty ./examples contains several examples of using library functions.
Directoty ./doc contains html documentation generated by ccdoc programm.

Conditionally, API functions can be divided into groups:

- Manage API: Open/close, receiving error, set/get API options
  - pts_open
  - pts_open_dev
  - pts_close
  - pts_last_error
  - pts_option_get
  - pts_option_set
- Handle of a controller configuration
  - pts_pump_config_get
  - pts_atg_config_get
  - pts_pump_config_set
  - pts_atg_config_set
  - pts_version_get
- Functions of a read and write parameters of the controller or filling places
  - pts_param_get
  - pts_param_set
- Functions of management by filling places
  - pts_authorize
  - pts_pts_authorize_by_amount
  - pts_halt
  - pts_lock
  - pts_unlock
  - pts_preset_prices
  - pts_prices_request
  - pts_status_request
  - pts_total_request
  - pts_trans_close
- Functions of reading measurements of Automatic Tank Gauge (ATG)
  - pts_atg_measure_request

Library functions (except for : pts_open, pts_open_dev, pts_close) have common features:

- return value: functions return 0 (FM_OK) if no error and error code if error (see include/pts_errors.h for
- determine the type of error)

- first of parameter - pts_api_descriptor - integer obtained by the open function: pts_open or pts_open_dev
- last parameter - pointer to PTS response structure - see include/pts_data.h for determine the data type.

All commands in protocol involve getting a response.

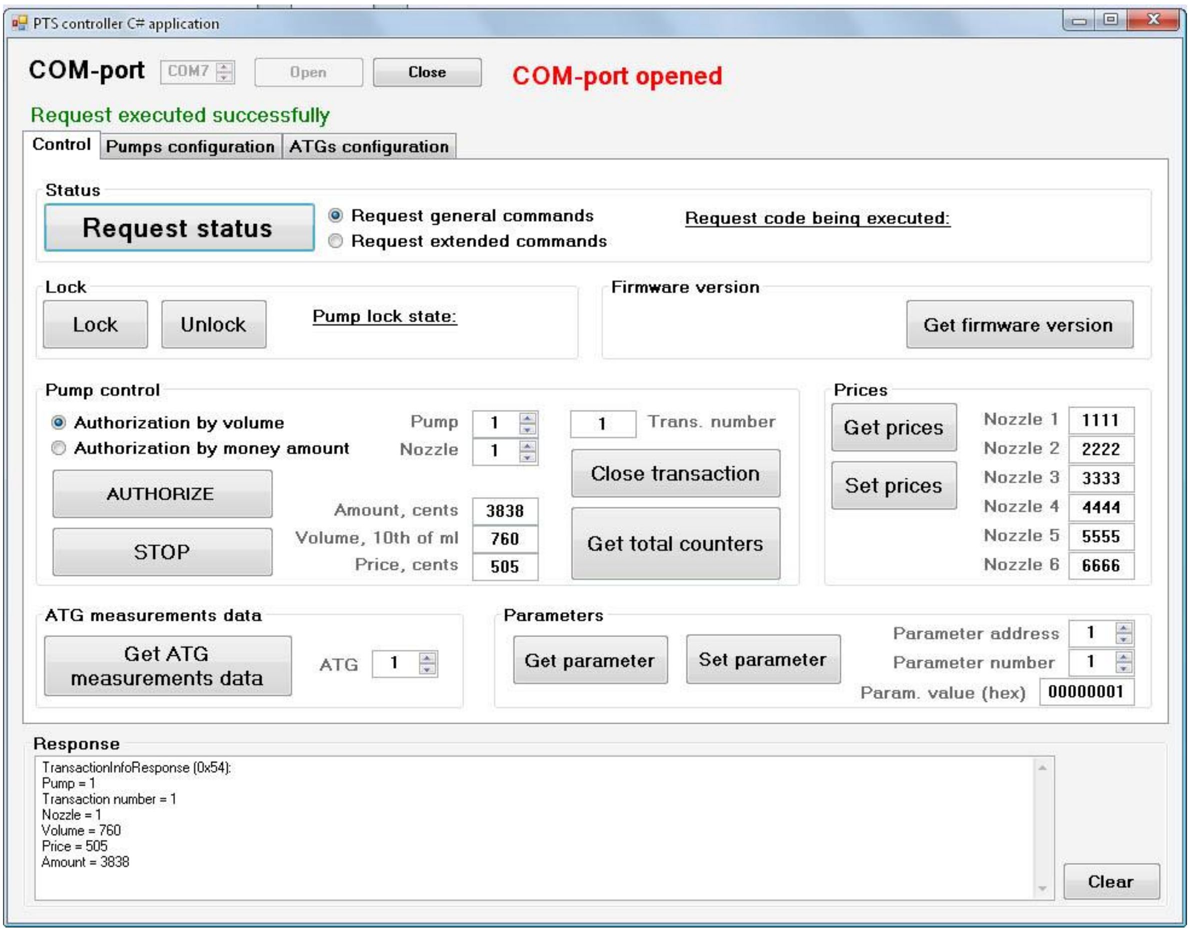
PS: If you want use *.bat files for call make, please edit it for set true path to your favorite language installation.

## *C# API*

The API for PTS controller is written in C# programming language in Visual Studio 2010 Express and requirs .NET Framework 2.0.

Given API includes 2 projects:

- PtsLib - project of PtsLib_CSharp.dll library for operation with a PTS controller. Includes classes, methods and properties for communication with PTS controller using requests and responses described in UniPump communication protocol for communication with PTS controller for provision of control over dispensers and ATG systems and configuration of PTS controller.
- Open-source utility for testing and indication of all operations provided by the PTS controller, allows to perform all operations with the PTS controller (uses classes and methods of PtsLib_CSharp.dll library).



COM-port: COM7

COM-port opened

Request executed successfully

● Request general commands
○ Request extended commands

● Authorization by volume
○ Authorization by money amount

Pump: 1
Nozzle: 1
Trans. number: 1
Amount, cents: 3838
Volume, 10th of ml: 760
Price, cents: 505

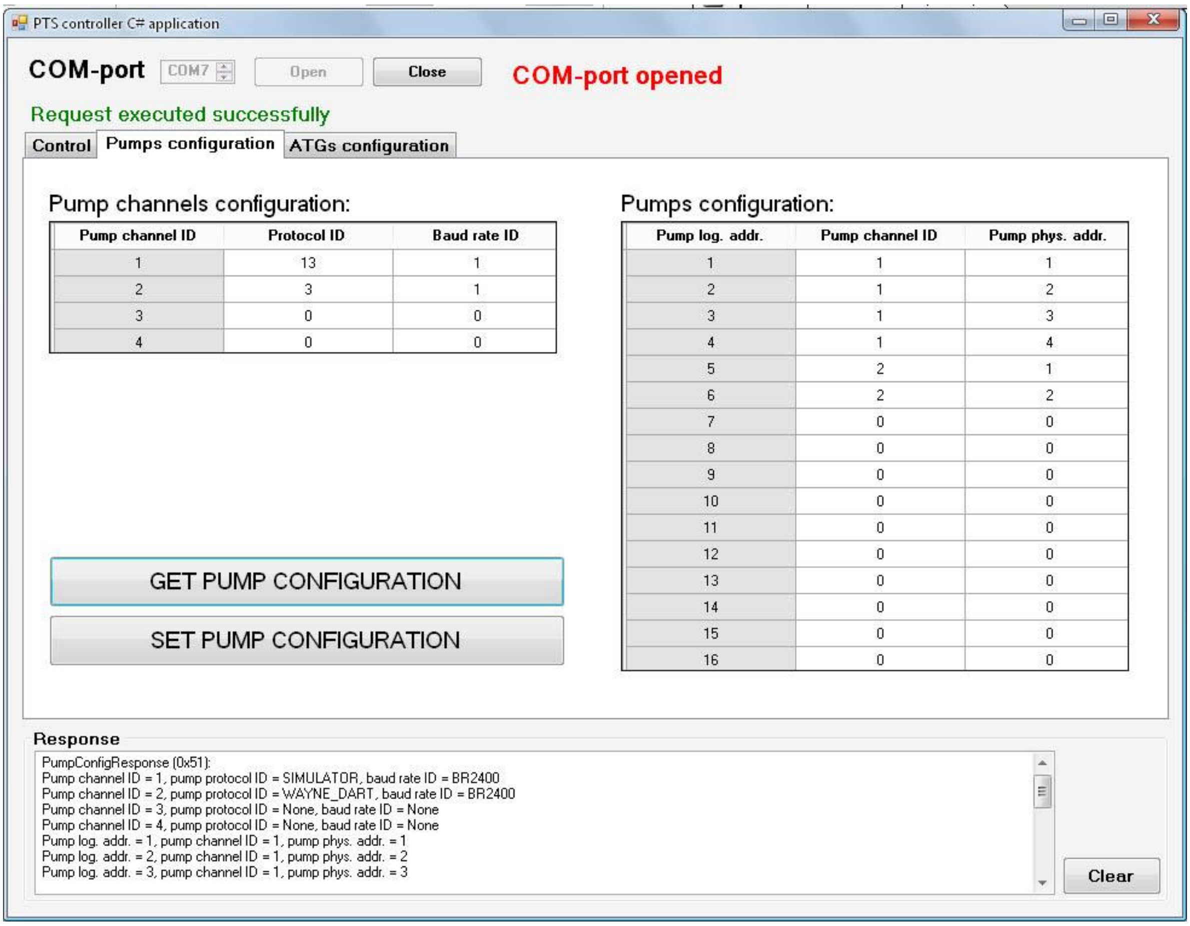
Nozzle 1: 1111
Nozzle 2: 2222
Nozzle 3: 3333
Nozzle 4: 4444
Nozzle 5: 5555
Nozzle 6: 6666

ATG: 1

Parameter address: 1
Parameter number: 1
Param. value (hex): 00000001

```
TransactionInfoResponse (0x54):
Pump = 1
Transaction number = 1
Nozzle = 1
Volume = 760
Price = 505
Amount = 3838
```

*Form for testing and indication of all operations provided by the PTS controller*



COM-port opened

Request executed successfully

Pump channels configuration:

| Pump channel ID | Protocol ID | Baud rate ID |
|---|---|---|
| 1 | 13 | 1 |
| 2 | 3 | 1 |
| 3 | 0 | 0 |
| 4 | 0 | 0 |

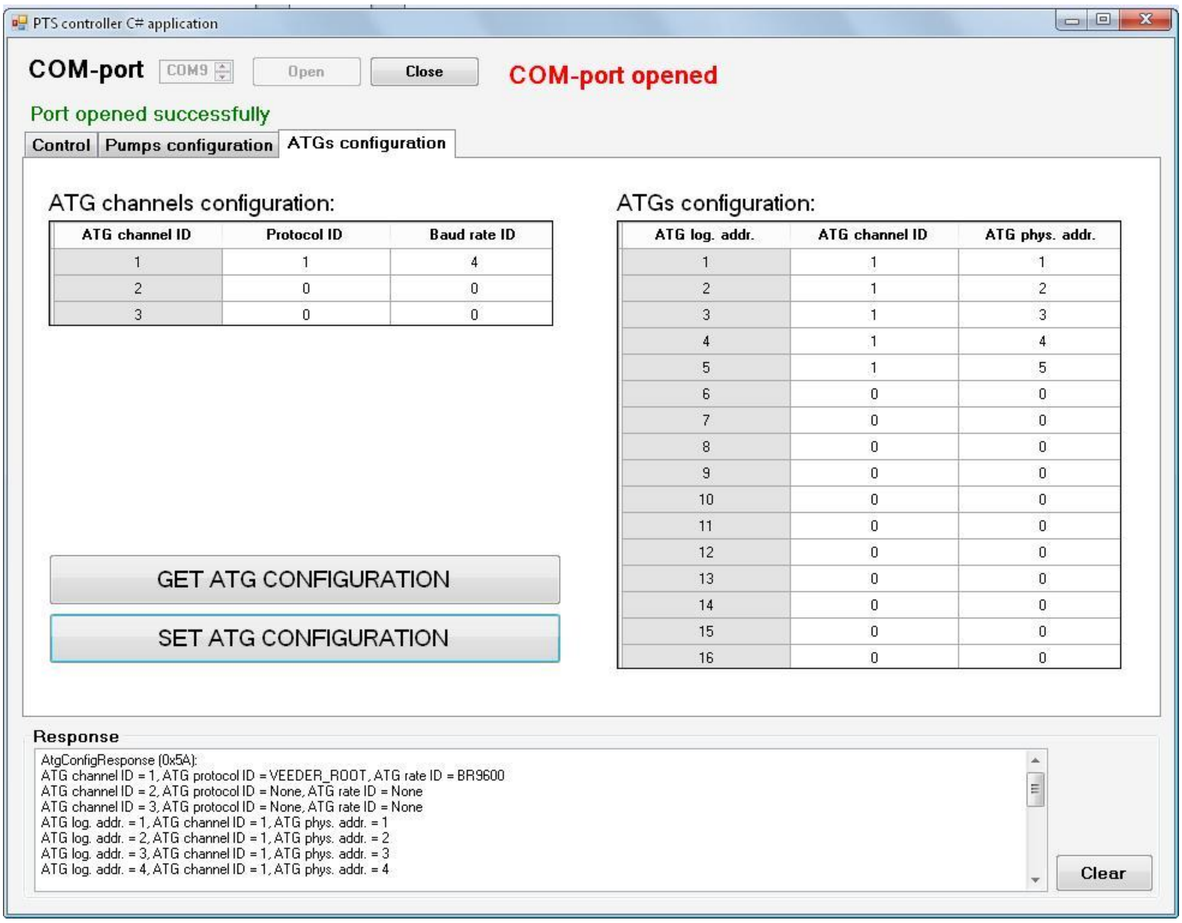
Pumps configuration:

| Pump log. addr. | Pump channel ID | Pump phys. addr. |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 1 | 2 |
| 3 | 1 | 3 |
| 4 | 1 | 4 |
| 5 | 2 | 1 |
| 6 | 2 | 2 |
| 7 | 0 | 0 |
| 8 | 0 | 0 |
| 9 | 0 | 0 |
| 10 | 0 | 0 |
| 11 | 0 | 0 |
| 12 | 0 | 0 |
| 13 | 0 | 0 |
| 14 | 0 | 0 |
| 15 | 0 | 0 |
| 16 | 0 | 0 |

Response

```
PumpConfigResponse (0x51):
Pump channel ID = 1, pump protocol ID = SIMULATOR, baud rate ID = BR2400
Pump channel ID = 2, pump protocol ID = WAYNE_DART, baud rate ID = BR2400
Pump channel ID = 3, pump protocol ID = None, baud rate ID = None
Pump channel ID = 4, pump protocol ID = None, baud rate ID = None
Pump log. addr. = 1, pump channel ID = 1, pump phys. addr. = 1
Pump log. addr. = 2, pump channel ID = 1, pump phys. addr. = 2
Pump log. addr. = 3, pump channel ID = 1, pump phys. addr. = 3
```

*Form for configuration of pumps channels of PTS controller*



COM-port opened

Port opened successfully

ATG channels configuration:

| ATG channel ID | Protocol ID | Baud rate ID |
|---|---|---|
| 1 | 1 | 4 |
| 2 | 0 | 0 |
| 3 | 0 | 0 |

ATGs configuration:

| ATG log. addr. | ATG channel ID | ATG phys. addr. |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 1 | 2 |
| 3 | 1 | 3 |
| 4 | 1 | 4 |
| 5 | 1 | 5 |
| 6 | 0 | 0 |
| 7 | 0 | 0 |
| 8 | 0 | 0 |
| 9 | 0 | 0 |
| 10 | 0 | 0 |
| 11 | 0 | 0 |
| 12 | 0 | 0 |
| 13 | 0 | 0 |
| 14 | 0 | 0 |
| 15 | 0 | 0 |
| 16 | 0 | 0 |

Response

```
AtgConfigResponse (0x5A):
ATG channel ID = 1, ATG protocol ID = VEEDER_ROOT, ATG rate ID = BR9600
ATG channel ID = 2, ATG protocol ID = None, ATG rate ID = None
ATG channel ID = 3, ATG protocol ID = None, ATG rate ID = None
ATG log. addr. = 1, ATG channel ID = 1, ATG phys. addr. = 1
ATG log. addr. = 2, ATG channel ID = 1, ATG phys. addr. = 2
ATG log. addr. = 3, ATG channel ID = 1, ATG phys. addr. = 3
ATG log. addr. = 4, ATG channel ID = 1, ATG phys. addr. = 4
```

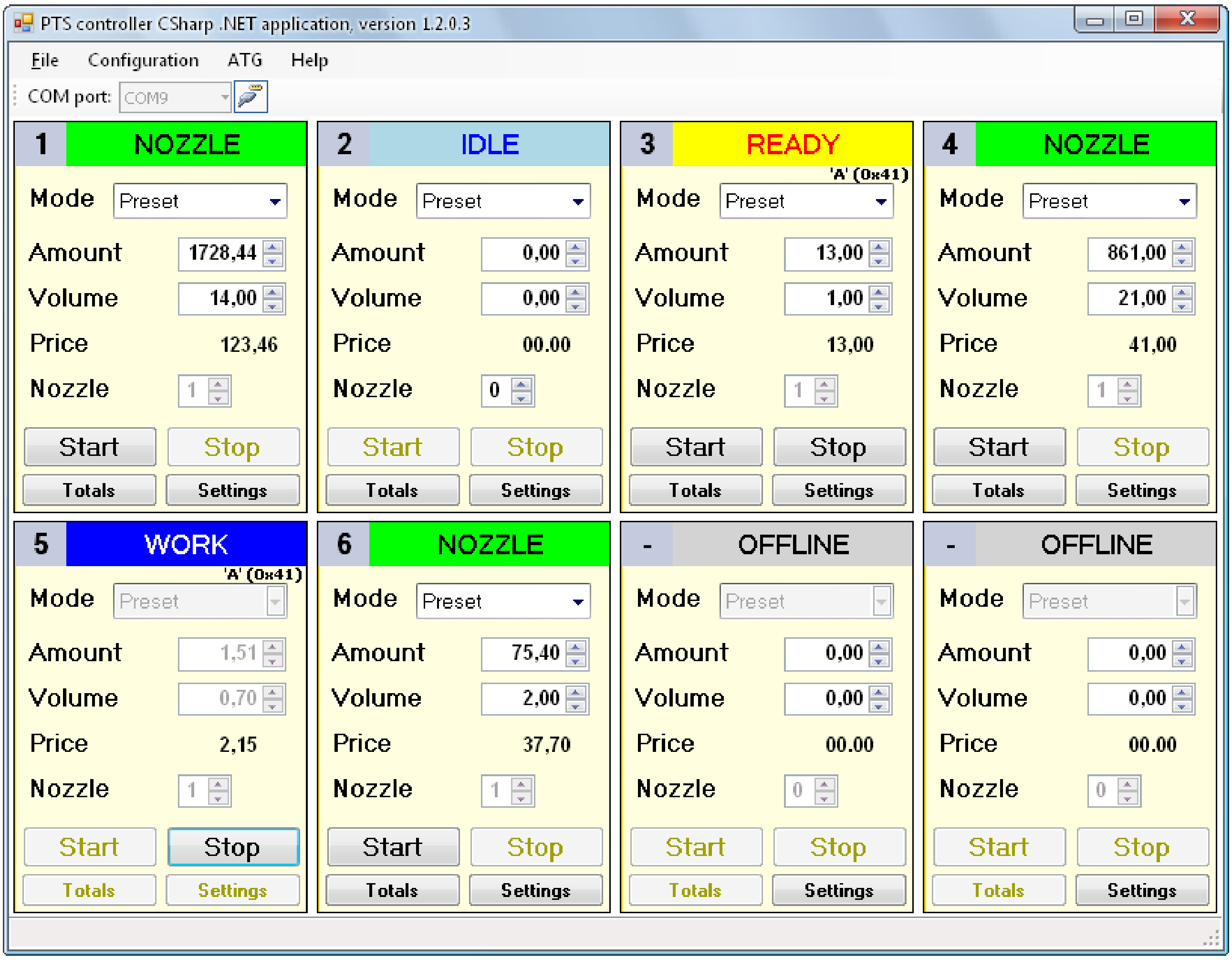
*Form for configuration of ATG systems channels of PTS controller*

## *C# application*

The application for PTS controller is written in C# programming language in Visual Studio 2010 Express and requirs .NET Framework 2.0.

Given application includes 2 projects:

- PtsLib - project of PtsLib.dll library for operation with a PTS controller. Includes classes, methods and properties for communication with PTS controller using requests and responses described in UniPump communication protocol for communication with PTS controller for provision of control over dispensers and ATG systems and configuration of PTS controller.
- Fully-functional open source application for provision of control over up to 8 pumps and 4 ATG systems by the PTS controller, allows to perform all operations with the PTS controller (uses classes and methods of PtsLib.dll library).

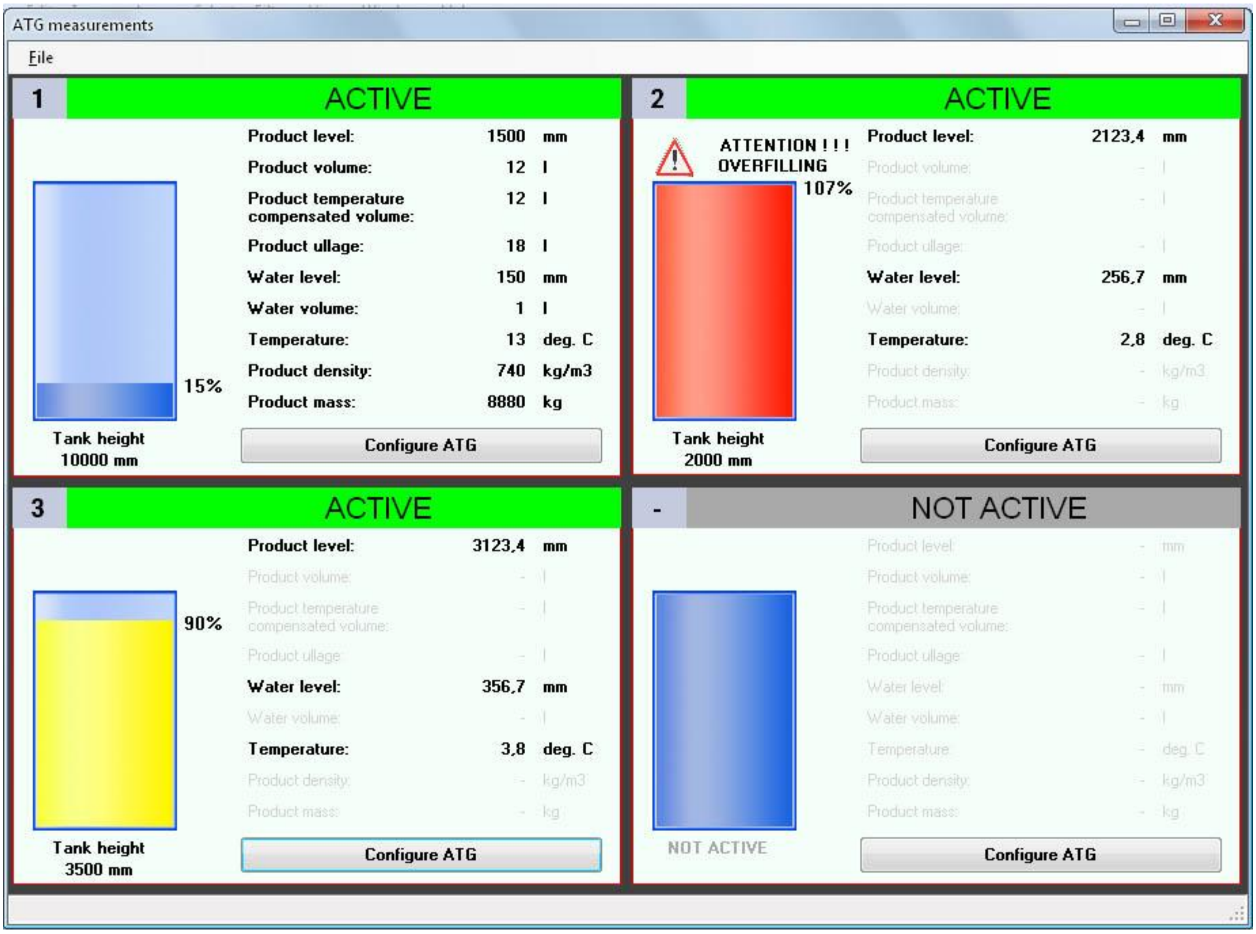


PTS controller CSharp .NET application, version 1.2.0.3

COM port: COM9

| 1 | 2 | 3 | 4 | 5 | 6 | - | - |
|---|---|---|---|---|---|---|---|
| NOZZLE | IDLE | READY | NOZZLE | WORK | NOZZLE | OFFLINE | OFFLINE |
|  |  | 'A' (0x41) |  | 'A' (0x41) |  |  |  |
| Mode: Preset | Mode: Preset | Mode: Preset | Mode: Preset | Mode: Preset | Mode: Preset | Mode: Preset | Mode: Preset |
| Amount: 1728,44 | Amount: 0,00 | Amount: 13,00 | Amount: 861,00 | Amount: 1,51 | Amount: 75,40 | Amount: 0,00 | Amount: 0,00 |
| Volume: 14,00 | Volume: 0,00 | Volume: 1,00 | Volume: 21,00 | Volume: 0,70 | Volume: 2,00 | Volume: 0,00 | Volume: 0,00 |
| Price: 123,46 | Price: 00.00 | Price: 13,00 | Price: 41,00 | Price: 2,15 | Price: 37,70 | Price: 00.00 | Price: 00.00 |
| Nozzle: 1 | Nozzle: 0 | Nozzle: 1 | Nozzle: 1 | Nozzle: 1 | Nozzle: 1 | Nozzle: 0 | Nozzle: 0 |

*General view of the application: allows to provide control over up to 8 pumps*



1 ACTIVE

| | | |
|---|---|---|
| Product level: | 1500 | mm |
| Product volume: | 12 | l |
| Product temperature compensated volume: | 12 | l |
| Product ullage: | 18 | l |
| Water level: | 150 | mm |
| Water volume: | 1 | l |
| Temperature: | 13 | deg. C |
| Product density: | 740 | kg/m3 |
| Product mass: | 8880 | kg |

15%

Tank height
10000 mm

2 ACTIVE

ATTENTION ! ! !
OVERFILLING
107%

| | | |
|---|---|---|
| Product level: | 2123,4 | mm |
| Product volume: | - | l |
| Product temperature compensated volume: | - | l |
| Product ullage: | - | l |
| Water level: | 256,7 | mm |
| Water volume: | - | l |
| Temperature: | 2,8 | deg. C |
| Product density: | - | kg/m3 |
| Product mass: | - | kg |

Tank height
2000 mm

3 ACTIVE

| | | |
|---|---|---|
| Product level: | 3123,4 | mm |
| Product volume: | - | l |
| Product temperature compensated volume: | - | l |
| Product ullage: | - | l |
| Water level: | 356,7 | mm |
| Water volume: | - | l |
| Temperature: | 3,8 | deg. C |
| Product density: | - | kg/m3 |
| Product mass: | - | kg |

90%

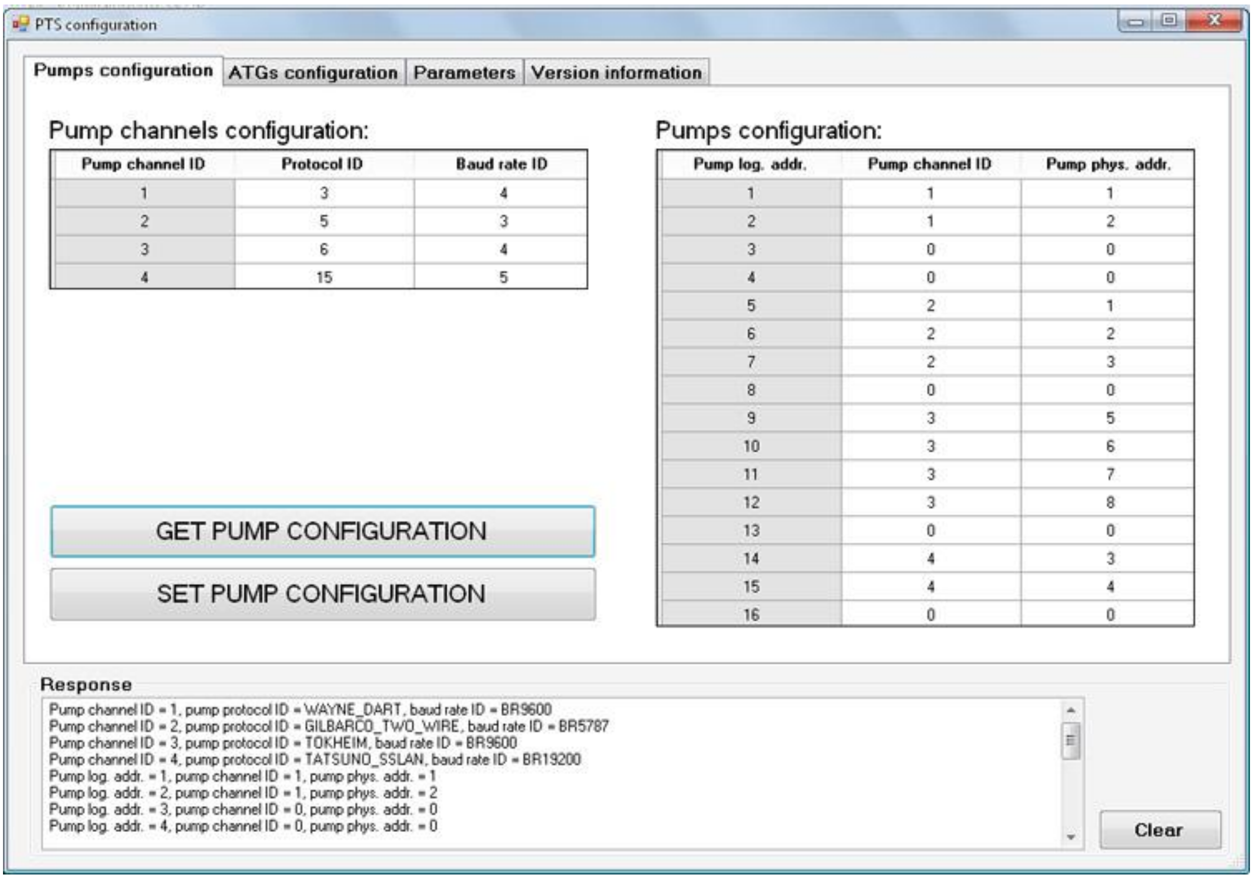
Tank height
3500 mm

- NOT ACTIVE

| | | |
|---|---|---|
| Product level: | - | mm |
| Product volume: | - | l |
| Product temperature compensated volume: | - | l |
| Product ullage: | - | l |
| Water level: | - | mm |
| Water volume: | - | l |
| Temperature: | - | deg. C |
| Product density: | - | kg/m3 |
| Product mass: | - | kg |

NOT ACTIVE

*Form of ATG systems monitoring: allows to monitor up to 4 ATG systems*



Pump channels configuration:

| Pump channel ID | Protocol ID | Baud rate ID |
|---|---|---|
| 1 | 3 | 4 |
| 2 | 5 | 3 |
| 3 | 6 | 4 |
| 4 | 15 | 5 |

Pumps configuration:

| Pump log. addr. | Pump channel ID | Pump phys. addr. |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 1 | 2 |
| 3 | 0 | 0 |
| 4 | 0 | 0 |
| 5 | 2 | 1 |
| 6 | 2 | 2 |
| 7 | 2 | 3 |
| 8 | 0 | 0 |
| 9 | 3 | 5 |
| 10 | 3 | 6 |
| 11 | 3 | 7 |
| 12 | 3 | 8 |
| 13 | 0 | 0 |
| 14 | 4 | 3 |
| 15 | 4 | 4 |
| 16 | 0 | 0 |

Response

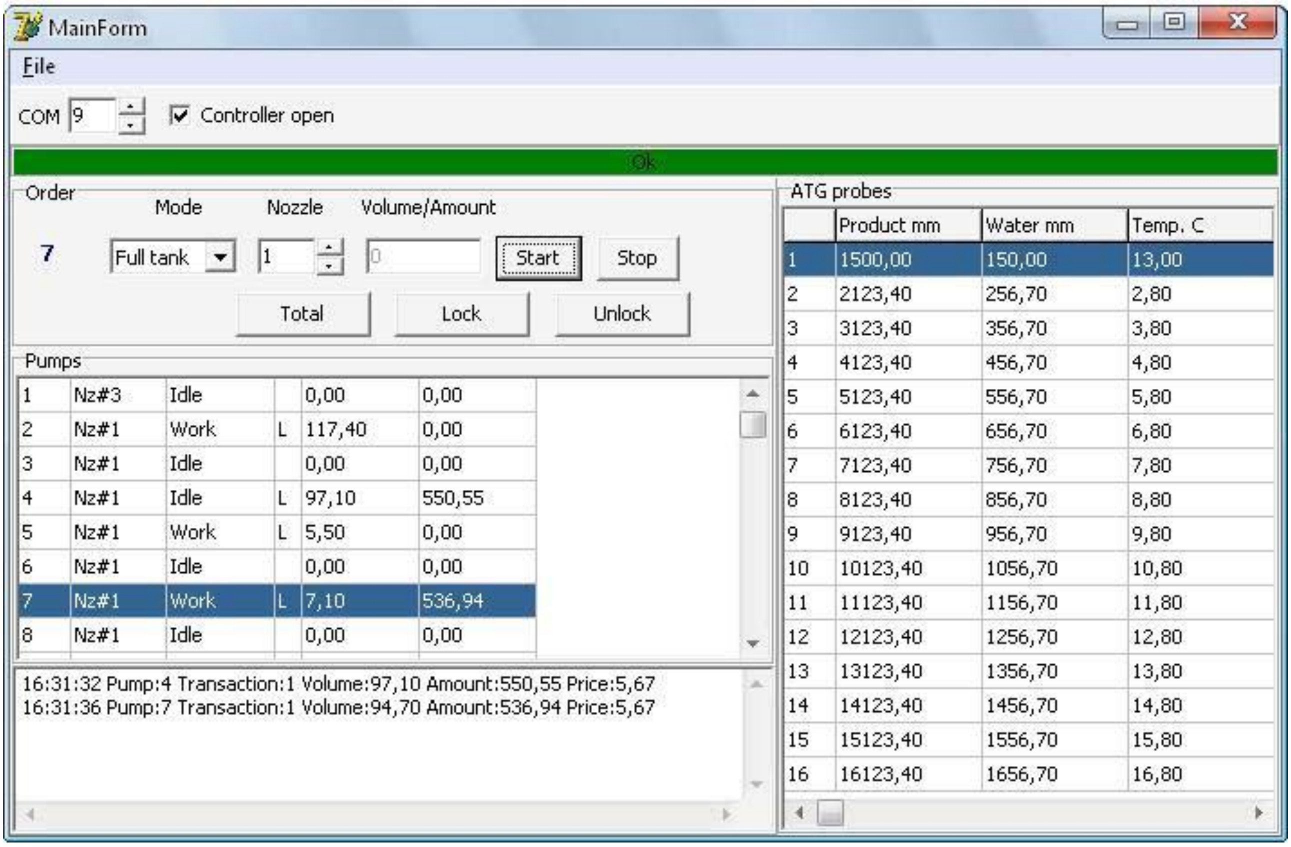
```
Pump channel ID = 1, pump protocol ID = WAYNE_DART, baud rate ID = BR9600
Pump channel ID = 2, pump protocol ID = GILBARCO_TWO_WIRE, baud rate ID = BR5787
Pump channel ID = 3, pump protocol ID = TOKHEIM, baud rate ID = BR9600
Pump channel ID = 4, pump protocol ID = TATSUNO_SSLAN, baud rate ID = BR19200
Pump log. addr. = 1, pump channel ID = 1, pump phys. addr. = 1
Pump log. addr. = 2, pump channel ID = 1, pump phys. addr. = 2
Pump log. addr. = 3, pump channel ID = 0, pump phys. addr. = 0
Pump log. addr. = 4, pump channel ID = 0, pump phys. addr. = 0
```

*Form of PTS controller configuration: allows to set configuration for pumps and ATG systems channels and also set parameters*

## *Delphi API*

Delphi API implements shared library libpts.dll calls.
Library libpts.dll was build with Borland compiler C PTS API (source C files you can find in libpts.zip).

Source code of library are located in ./src directory. Directoty ./examples contains several examples of using library functions.



COM 9 ☑ Controller open

7 Mode: Full tank Nozzle: 1 Volume/Amount: 0

| | | | | | |
|---|---|---|---|---|---|
| 1 | Nz#3 | Idle | | 0,00 | 0,00 |
| 2 | Nz#1 | Work | L | 117,40 | 0,00 |
| 3 | Nz#1 | Idle | | 0,00 | 0,00 |
| 4 | Nz#1 | Idle | L | 97,10 | 550,55 |
| 5 | Nz#1 | Work | L | 5,50 | 0,00 |
| 6 | Nz#1 | Idle | | 0,00 | 0,00 |
| 7 | Nz#1 | Work | L | 7,10 | 536,94 |
| 8 | Nz#1 | Idle | | 0,00 | 0,00 |

```
16:31:32 Pump:4 Transaction:1 Volume:97,10 Amount:550,55 Price:5,67
16:31:36 Pump:7 Transaction:1 Volume:94,70 Amount:536,94 Price:5,67
```

| | Product mm | Water mm | Temp. C |
|---|---|---|---|
| 1 | 1500,00 | 150,00 | 13,00 |
| 2 | 2123,40 | 256,70 | 2,80 |
| 3 | 3123,40 | 356,70 | 3,80 |
| 4 | 4123,40 | 456,70 | 4,80 |
| 5 | 5123,40 | 556,70 | 5,80 |
| 6 | 6123,40 | 656,70 | 6,80 |
| 7 | 7123,40 | 756,70 | 7,80 |
| 8 | 8123,40 | 856,70 | 8,80 |
| 9 | 9123,40 | 956,70 | 9,80 |
| 10 | 10123,40 | 1056,70 | 10,80 |
| 11 | 11123,40 | 1156,70 | 11,80 |
| 12 | 12123,40 | 1256,70 | 12,80 |
| 13 | 13123,40 | 1356,70 | 13,80 |
| 14 | 14123,40 | 1456,70 | 14,80 |
| 15 | 15123,40 | 1556,70 | 15,80 |
| 16 | 16123,40 | 1656,70 | 16,80 |

*Graphical form for testing API for PTS controller written in Delphi*

Conditionally, API functions can be divided into groups:

- Manage API: Open/close, receiving error, set/get API options
  - pts_open
  - pts_open_dev
  - pts_close
  - pts_last_error
  - pts_option_get
  - pts_option_set
- Handle of a controller configuration
  - pts_pump_config_get
  - pts_atg_config_get
  - pts_pump_config_set
  - pts_atg_config_set
  - pts_version_get
- Functions of a read and write parameters of the controller or filling places
  - pts_param_get
  - pts_param_set
- Functions of management by filling places
  - pts_authorize

  - pts_pts_authorize_by_amount
  - pts_halt
  - pts_lock
  - pts_unlock
  - pts_preset_prices
  - pts_prices_request
  - pts_status_request
  - pts_total_request
  - pts_trans_close
- Functions of reading measurements of Automatic Tank Gauge (ATG)
  - pts_atg_measure_request

Library functions (except for : pts_open, pts_open_dev, pts_close) have common features:

- return value: functions return 0 (FM_OK) if no error and error code if error (see include/pts_errors.h for
- determine the type of error)
- first of parameter - pts_api_descriptor - integer obtained by the open function: pts_open or pts_open_dev
- last parameter - pointer to PTS response structure - see include/pts_data.h for determine the data type.

All commands in protocol involve getting a response.

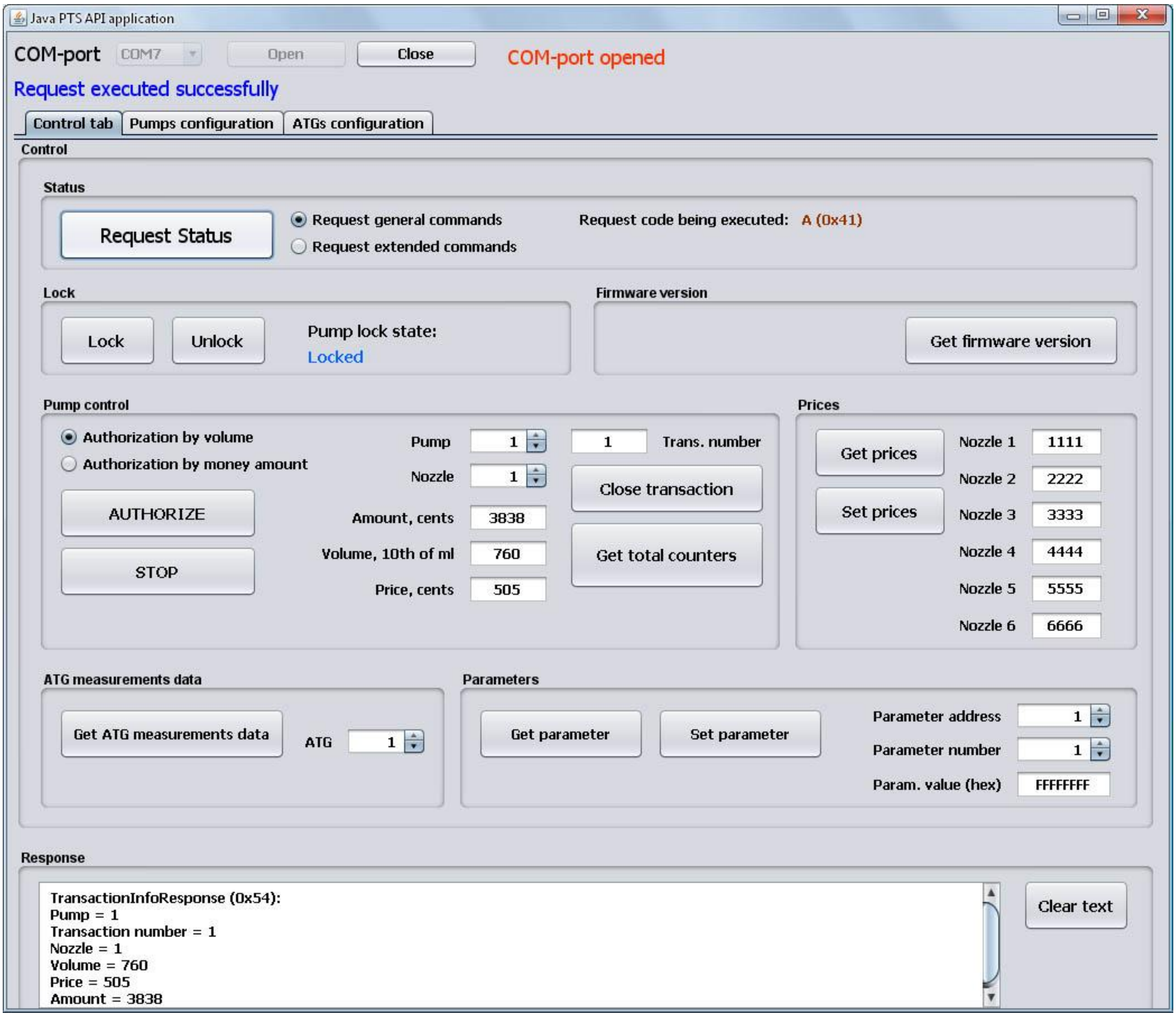
# *Java API*

Java API for PTS controller is written in Java programming language in NetBeans IDE 7.2.1.

Given API includes:

- JavaPtsLib: library for operation with a PTS controller. Includes classes, methods and properties for communication with PTS controller using requests and responses described in UniPump communication protocol for communication with PTS controller for provision of control over dispensers and ATG systems and configuration of PTS controller.
- JavaPtsApp: open-source utility for testing and indication of all operations provided by the PTS controller, allows to perform all operation with the PTS controller (uses classes and methods of JavaPtsLib).

Communication over the COM-port (RS-232 interface) is done using RXTX Java library (visit www.rxtx.qbang.org for downloads and information): RXTX is a Java library, using a native implementation (via JNI), providing serial and parallel communication for the Java Development Toolkit (JDK). It is based on the specification for Sun's Java Communications API, though while many of the class descriptions are the same the package used it not, since gnu.io is used instead. For more information please visit the rxtx wiki at http://rxtx.qbang.org/wiki (latest releases of the library can be downloaded there).



COM-port: COM7

COM-port opened

Request executed successfully

● Request general commands
○ Request extended commands

Request code being executed: A (0x41)

Pump lock state: Locked

● Authorization by volume
○ Authorization by money amount

| | |
|---|---|
| Pump | 1 |
| Trans. number | 1 |
| Nozzle | 1 |
| Amount, cents | 3838 |
| Volume, 10th of ml | 760 |
| Price, cents | 505 |

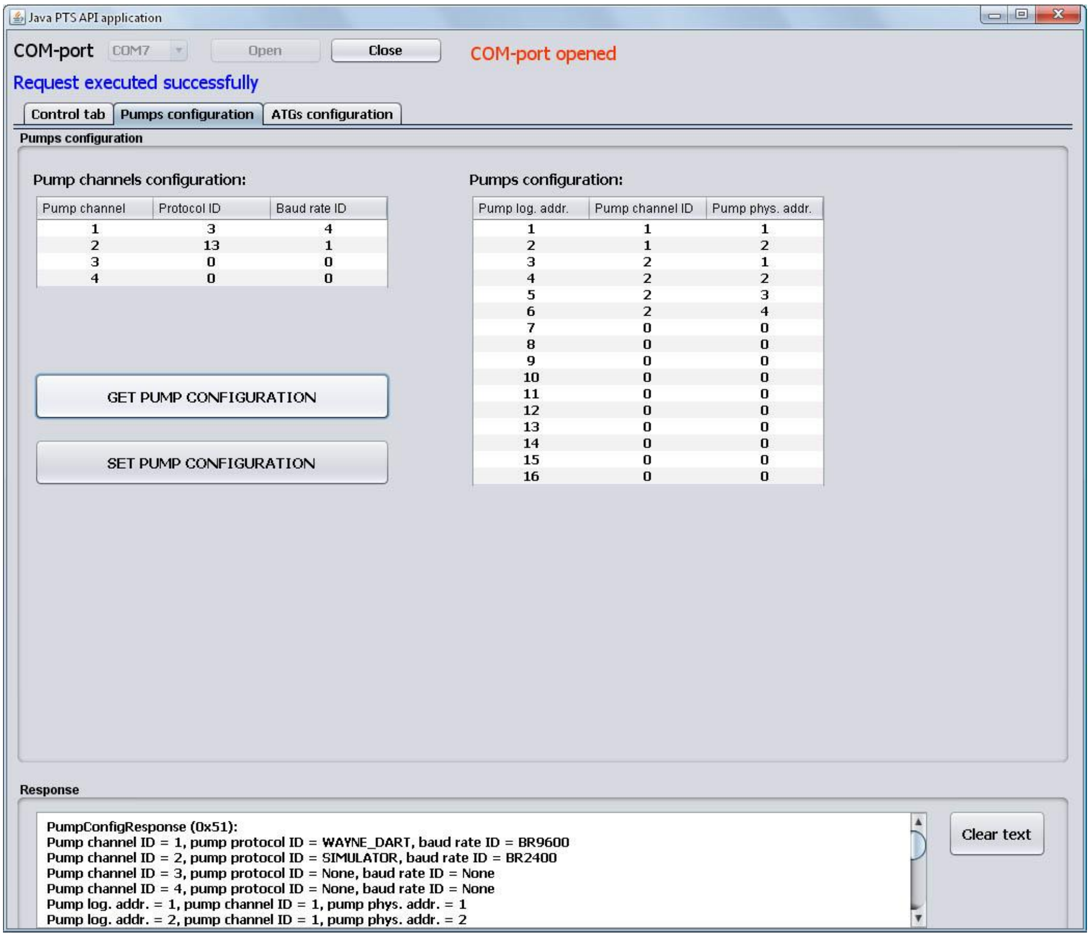
| | |
|---|---|
| Nozzle 1 | 1111 |
| Nozzle 2 | 2222 |
| Nozzle 3 | 3333 |
| Nozzle 4 | 4444 |
| Nozzle 5 | 5555 |
| Nozzle 6 | 6666 |

ATG: 1

| | |
|---|---|
| Parameter address | 1 |
| Parameter number | 1 |
| Param. value (hex) | FFFFFFFF |

```
TransactionInfoResponse (0x54):
Pump = 1
Transaction number = 1
Nozzle = 1
Volume = 760
Price = 505
Amount = 3838
```

*Form for testing and indication of all operations provided by the PTS controller*



COM-port opened

Request executed successfully

Pump channels configuration:

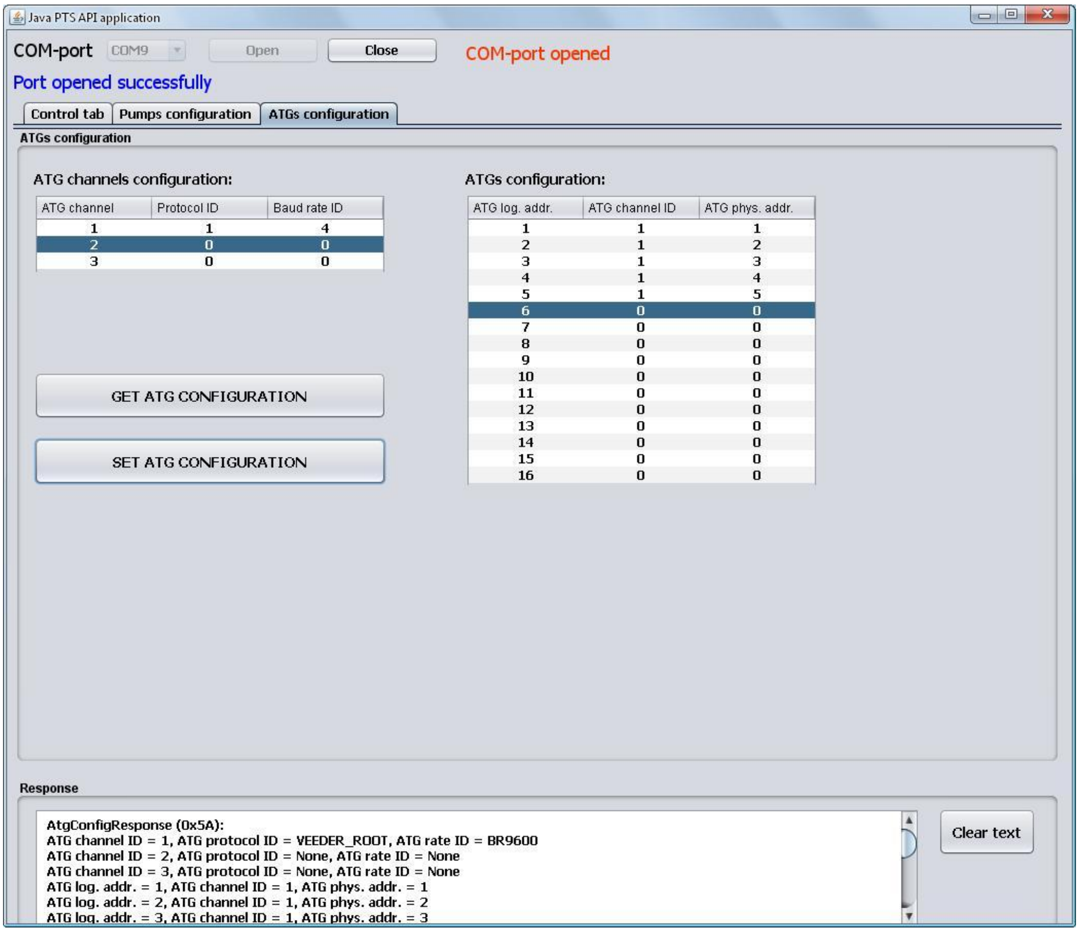
| Pump channel | Protocol ID | Baud rate ID |
|---|---|---|
| 1 | 3 | 4 |
| 2 | 13 | 1 |
| 3 | 0 | 0 |
| 4 | 0 | 0 |

Pumps configuration:

| Pump log. addr. | Pump channel ID | Pump phys. addr. |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 1 | 2 |
| 3 | 2 | 1 |
| 4 | 2 | 2 |
| 5 | 2 | 3 |
| 6 | 2 | 4 |
| 7 | 0 | 0 |
| 8 | 0 | 0 |
| 9 | 0 | 0 |
| 10 | 0 | 0 |
| 11 | 0 | 0 |
| 12 | 0 | 0 |
| 13 | 0 | 0 |
| 14 | 0 | 0 |
| 15 | 0 | 0 |
| 16 | 0 | 0 |

```
PumpConfigResponse (0x51):
Pump channel ID = 1, pump protocol ID = WAYNE_DART, baud rate ID = BR9600
Pump channel ID = 2, pump protocol ID = SIMULATOR, baud rate ID = BR2400
Pump channel ID = 3, pump protocol ID = None, baud rate ID = None
Pump channel ID = 4, pump protocol ID = None, baud rate ID = None
Pump log. addr. = 1, pump channel ID = 1, pump phys. addr. = 1
Pump log. addr. = 2, pump channel ID = 1, pump phys. addr. = 2
```

*Form for configuration of pumps channels of PTS controller*



COM-port opened

Port opened successfully

ATG channels configuration:

| ATG channel | Protocol ID | Baud rate ID |
|---|---|---|
| 1 | 1 | 4 |
| 2 | 0 | 0 |
| 3 | 0 | 0 |

ATGs configuration:

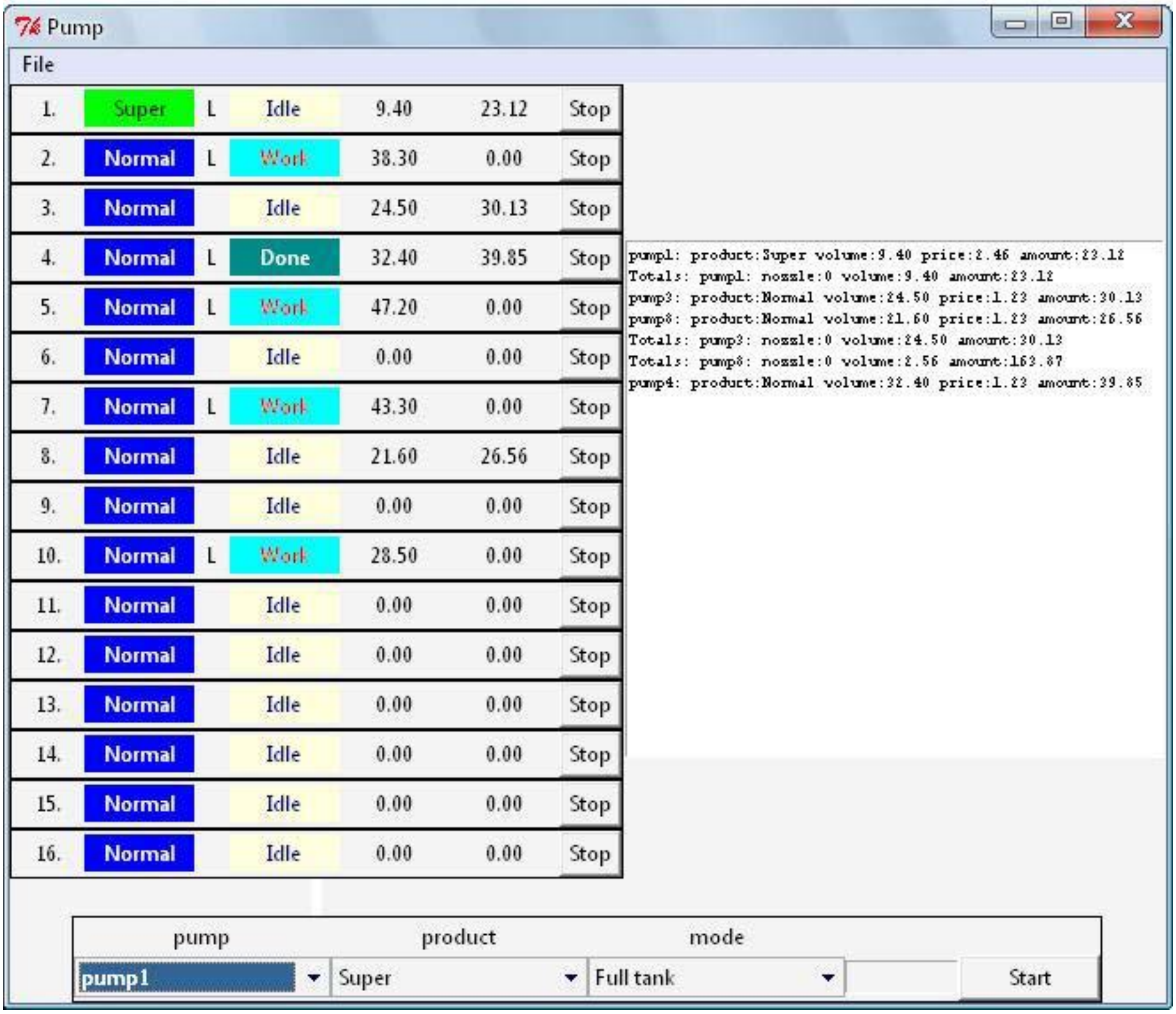
| ATG log. addr. | ATG channel ID | ATG phys. addr. |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 1 | 2 |
| 3 | 1 | 3 |
| 4 | 1 | 4 |
| 5 | 1 | 5 |
| 6 | 0 | 0 |
| 7 | 0 | 0 |
| 8 | 0 | 0 |
| 9 | 0 | 0 |
| 10 | 0 | 0 |
| 11 | 0 | 0 |
| 12 | 0 | 0 |
| 13 | 0 | 0 |
| 14 | 0 | 0 |
| 15 | 0 | 0 |
| 16 | 0 | 0 |

```
AtgConfigResponse (0x5A):
ATG channel ID = 1, ATG protocol ID = VEEDER_ROOT, ATG rate ID = BR9600
ATG channel ID = 2, ATG protocol ID = None, ATG rate ID = None
ATG channel ID = 3, ATG protocol ID = None, ATG rate ID = None
ATG log. addr. = 1, ATG channel ID = 1, ATG phys. addr. = 1
ATG log. addr. = 2, ATG channel ID = 1, ATG phys. addr. = 2
ATG log. addr. = 3, ATG channel ID = 1, ATG phys. addr. = 3
```

*Form for configuration of ATG systems channels of PTS controller*

## Python API

Before installation of this package it is necessary to install PySerial (http://pyserial.sourceforge.net/).
To install PTS module, run python setip.py install, Where python = your desired python version executable.



| | | | | | | |
|---|---|---|---|---|---|---|
| 1. | Super | L | Idle | 9.40 | 23.12 | Stop |
| 2. | Normal | L | Work | 38.30 | 0.00 | Stop |
| 3. | Normal | | Idle | 24.50 | 30.13 | Stop |
| 4. | Normal | L | Done | 32.40 | 39.85 | Stop |
| 5. | Normal | L | Work | 47.20 | 0.00 | Stop |
| 6. | Normal | | Idle | 0.00 | 0.00 | Stop |
| 7. | Normal | L | Work | 43.30 | 0.00 | Stop |
| 8. | Normal | | Idle | 21.60 | 26.56 | Stop |
| 9. | Normal | | Idle | 0.00 | 0.00 | Stop |
| 10. | Normal | L | Work | 28.50 | 0.00 | Stop |
| 11. | Normal | | Idle | 0.00 | 0.00 | Stop |
| 12. | Normal | | Idle | 0.00 | 0.00 | Stop |
| 13. | Normal | | Idle | 0.00 | 0.00 | Stop |
| 14. | Normal | | Idle | 0.00 | 0.00 | Stop |
| 15. | Normal | | Idle | 0.00 | 0.00 | Stop |
| 16. | Normal | | Idle | 0.00 | 0.00 | Stop |

```
pump1: product:Super volume:9.40 price:2.46 amount:23.12
Totals: pump1: nozzle:0 volume:9.40 amount:23.12
pump3: product:Normal volume:24.50 price:1.23 amount:30.13
pump8: product:Normal volume:21.60 price:1.23 amount:26.56
Totals: pump3: nozzle:0 volume:24.50 amount:30.13
Totals: pump8: nozzle:0 volume:2.56 amount:163.87
pump4: product:Normal volume:32.40 price:1.23 amount:39.85
```

| pump | product | mode | | |
|---|---|---|---|---|
| pump1 | Super | Full tank | | Start |

*Graphical form for testing API for PTS controller written in Python*

The protocol of exchange is realized in the pts.protocol module.
The exchange through a serial port is realized with use PySerial module (http://pyserial.sourceforge.net/) (therefore, before installation of this package it is necessary to install PySerial).

The package was tested with python of versions 2.7 and 3.0

The pts.constants module contains values of various constants
The pts.config module - structures of data, for a controler configuration
The pts.response module - structures of the data returned by the controler
The pts.protocol module - protocol commands

Conditionally, API functions can be divided into groups:

- Manage API: Open/close, receiving error, set/get API options
  - pts_open
  - pts_open_dev

  - pts_close
  - pts_last_error
  - pts_option_get
  - pts_option_set
- Handle of a controller configuration
  - pts_pump_config_get
  - pts_atg_config_get
  - pts_pump_config_set
  - pts_atg_config_set
  - pts_version_get
- Functions of a read and write parameters of the controller or filling places
  - pts_param_get
  - pts_param_set
- Functions of management by filling places
  - pts_authorize
  - pts_pts_authorize_by_amount
  - pts_halt
  - pts_lock
  - pts_unlock
  - pts_preset_prices
  - pts_prices_request
  - pts_status_request
  - pts_total_request
  - pts_trans_close
- Functions of reading measurements of Automatic Tank Gauge (ATG)
  - pts_atg_measure_request

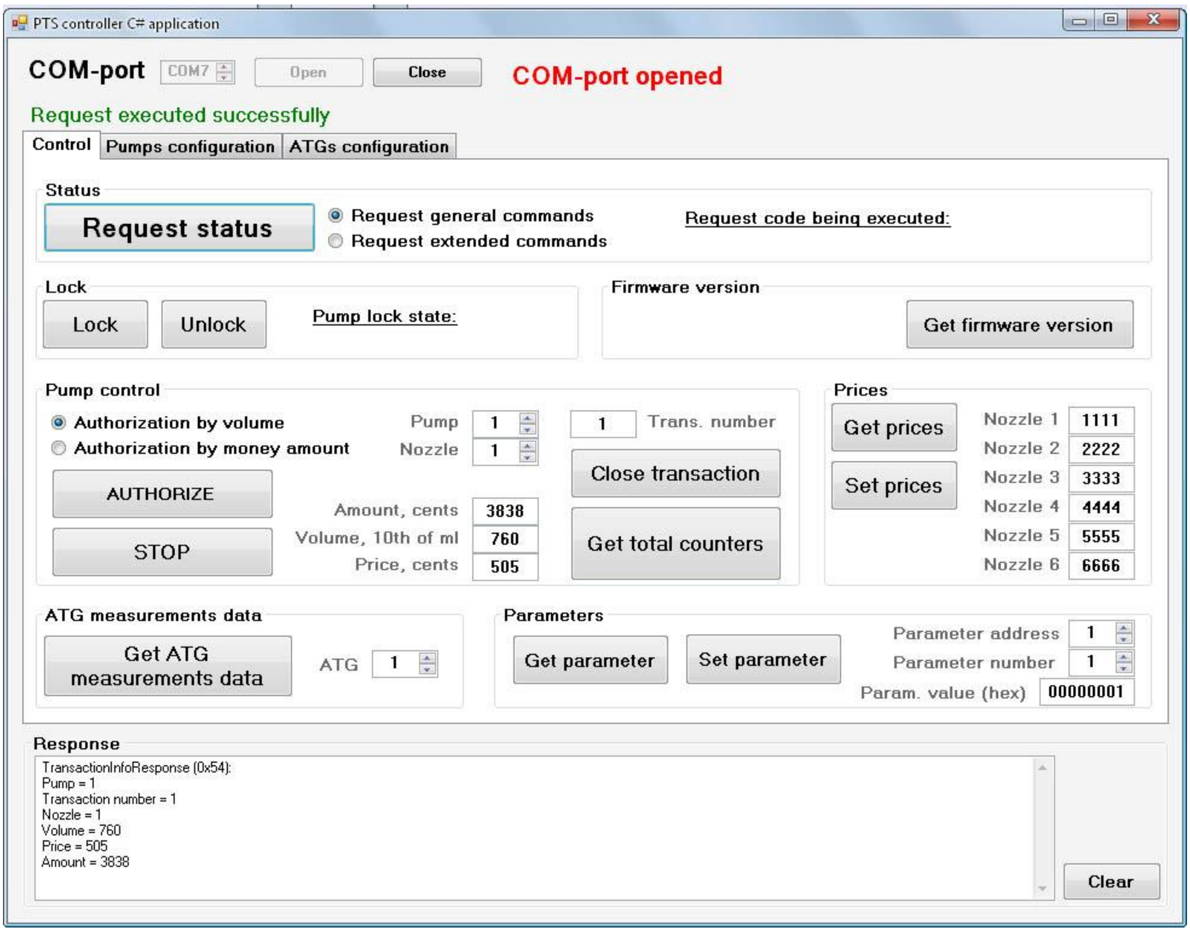
Work with the protocol can be set for work without exceptions (when error occured), and with generation of exceptions (PtsException). Parameter of initialization of use_exceptions is responsible for it.

## *Visual Basic .NET API*

Visual Basic .NET API for PTS controller is written in VB.NET programming language in Visual Studio 2010 Express and requirs .NET Framework 2.0.

Given API includes 2 projects:

- PtsLib - project of PtsLib_VBNET.dll library for operation with a PTS controller. Includes classes, methods and properties for communication with PTS controller using requests and responses described in UniPump communication protocol for communication with PTS controller for provision of control over dispensers and ATG systems and configuration of PTS controller.
- PTS - open-source utility for testing and indication of all operations provided by the PTS controller, allows to perform all operations with the PTS controller (uses classes and methods of PtsLib_VBNET.dll library).



COM-port: COM7

COM-port opened

Request executed successfully

● Request general commands
○ Request extended commands

● Authorization by volume
○ Authorization by money amount

Pump: 1
Nozzle: 1
Trans. number: 1
Amount, cents: 3838
Volume, 10th of ml: 760
Price, cents: 505

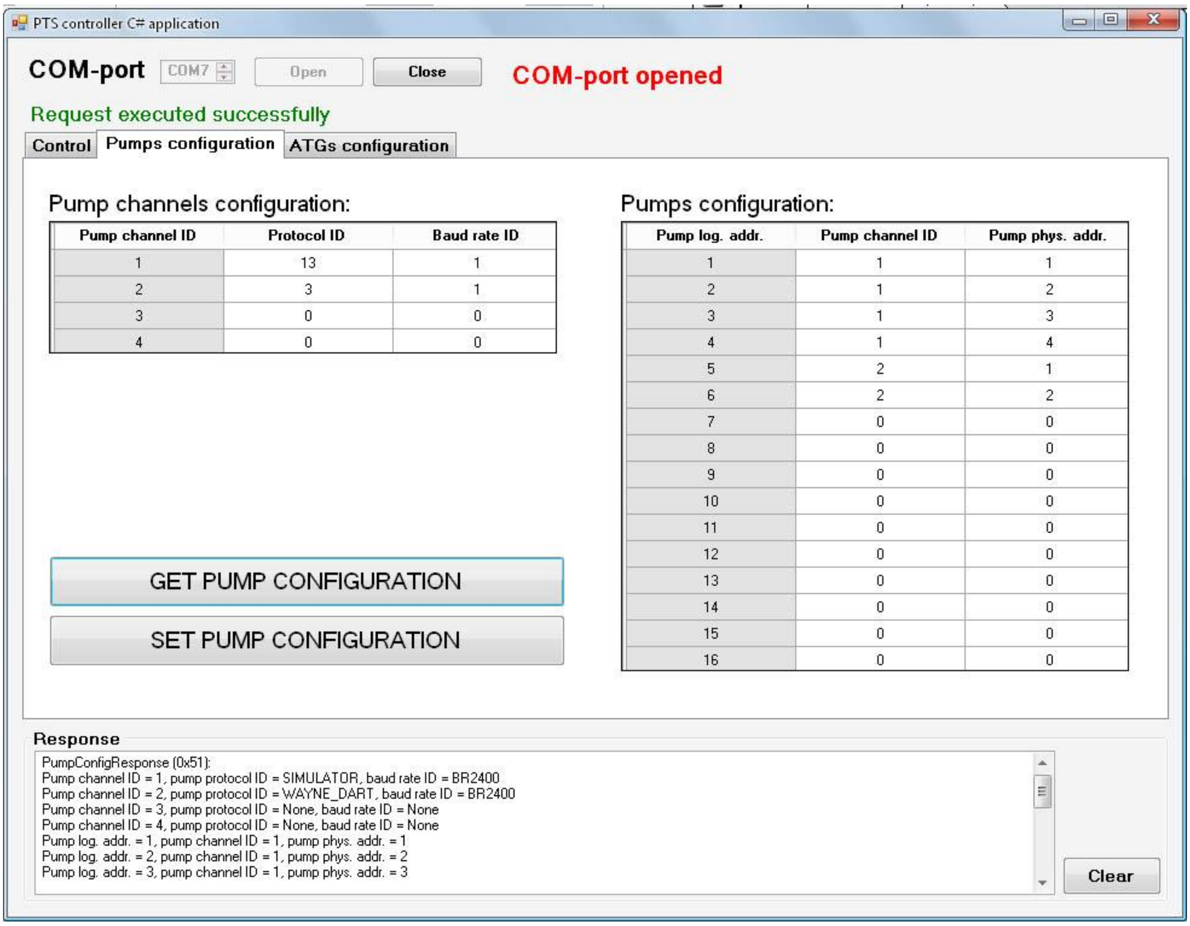
Nozzle 1: 1111
Nozzle 2: 2222
Nozzle 3: 3333
Nozzle 4: 4444
Nozzle 5: 5555
Nozzle 6: 6666

ATG: 1

Parameter address: 1
Parameter number: 1
Param. value (hex): 00000001

```
TransactionInfoResponse (0x54):
Pump = 1
Transaction number = 1
Nozzle = 1
Volume = 760
Price = 505
Amount = 3838
```

*Form for testing and indication of all operations provided by the PTS controller*



COM-port opened

Request executed successfully

Pump channels configuration:

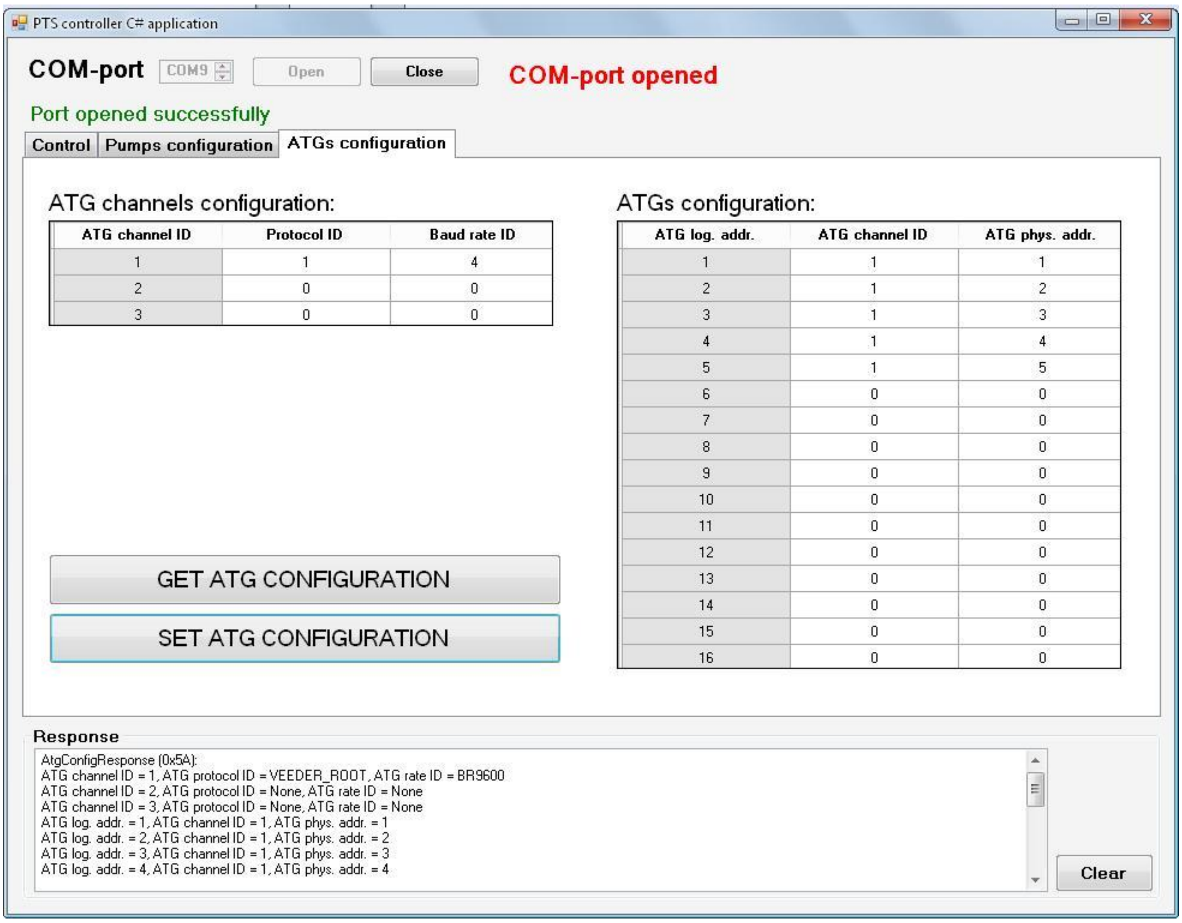
| Pump channel ID | Protocol ID | Baud rate ID |
|---|---|---|
| 1 | 13 | 1 |
| 2 | 3 | 1 |
| 3 | 0 | 0 |
| 4 | 0 | 0 |

Pumps configuration:

| Pump log. addr. | Pump channel ID | Pump phys. addr. |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 1 | 2 |
| 3 | 1 | 3 |
| 4 | 1 | 4 |
| 5 | 2 | 1 |
| 6 | 2 | 2 |
| 7 | 0 | 0 |
| 8 | 0 | 0 |
| 9 | 0 | 0 |
| 10 | 0 | 0 |
| 11 | 0 | 0 |
| 12 | 0 | 0 |
| 13 | 0 | 0 |
| 14 | 0 | 0 |
| 15 | 0 | 0 |
| 16 | 0 | 0 |

```
PumpConfigResponse (0x51):
Pump channel ID = 1, pump protocol ID = SIMULATOR, baud rate ID = BR2400
Pump channel ID = 2, pump protocol ID = WAYNE_DART, baud rate ID = BR2400
Pump channel ID = 3, pump protocol ID = None, baud rate ID = None
Pump channel ID = 4, pump protocol ID = None, baud rate ID = None
Pump log. addr. = 1, pump channel ID = 1, pump phys. addr. = 1
Pump log. addr. = 2, pump channel ID = 1, pump phys. addr. = 2
Pump log. addr. = 3, pump channel ID = 1, pump phys. addr. = 3
```

*Form for configuration of pumps channels of PTS controller*



COM-port opened

Port opened successfully

ATG channels configuration:

| ATG channel ID | Protocol ID | Baud rate ID |
|---|---|---|
| 1 | 1 | 4 |
| 2 | 0 | 0 |
| 3 | 0 | 0 |

ATGs configuration:

| ATG log. addr. | ATG channel ID | ATG phys. addr. |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 1 | 2 |
| 3 | 1 | 3 |
| 4 | 1 | 4 |
| 5 | 1 | 5 |
| 6 | 0 | 0 |
| 7 | 0 | 0 |
| 8 | 0 | 0 |
| 9 | 0 | 0 |
| 10 | 0 | 0 |
| 11 | 0 | 0 |
| 12 | 0 | 0 |
| 13 | 0 | 0 |
| 14 | 0 | 0 |
| 15 | 0 | 0 |
| 16 | 0 | 0 |

```
AtgConfigResponse (0x5A):
ATG channel ID = 1, ATG protocol ID = VEEDER_ROOT, ATG rate ID = BR9600
ATG channel ID = 2, ATG protocol ID = None, ATG rate ID = None
ATG channel ID = 3, ATG protocol ID = None, ATG rate ID = None
ATG log. addr. = 1, ATG channel ID = 1, ATG phys. addr. = 1
ATG log. addr. = 2, ATG channel ID = 1, ATG phys. addr. = 2
ATG log. addr. = 3, ATG channel ID = 1, ATG phys. addr. = 3
ATG log. addr. = 4, ATG channel ID = 1, ATG phys. addr. = 4
```

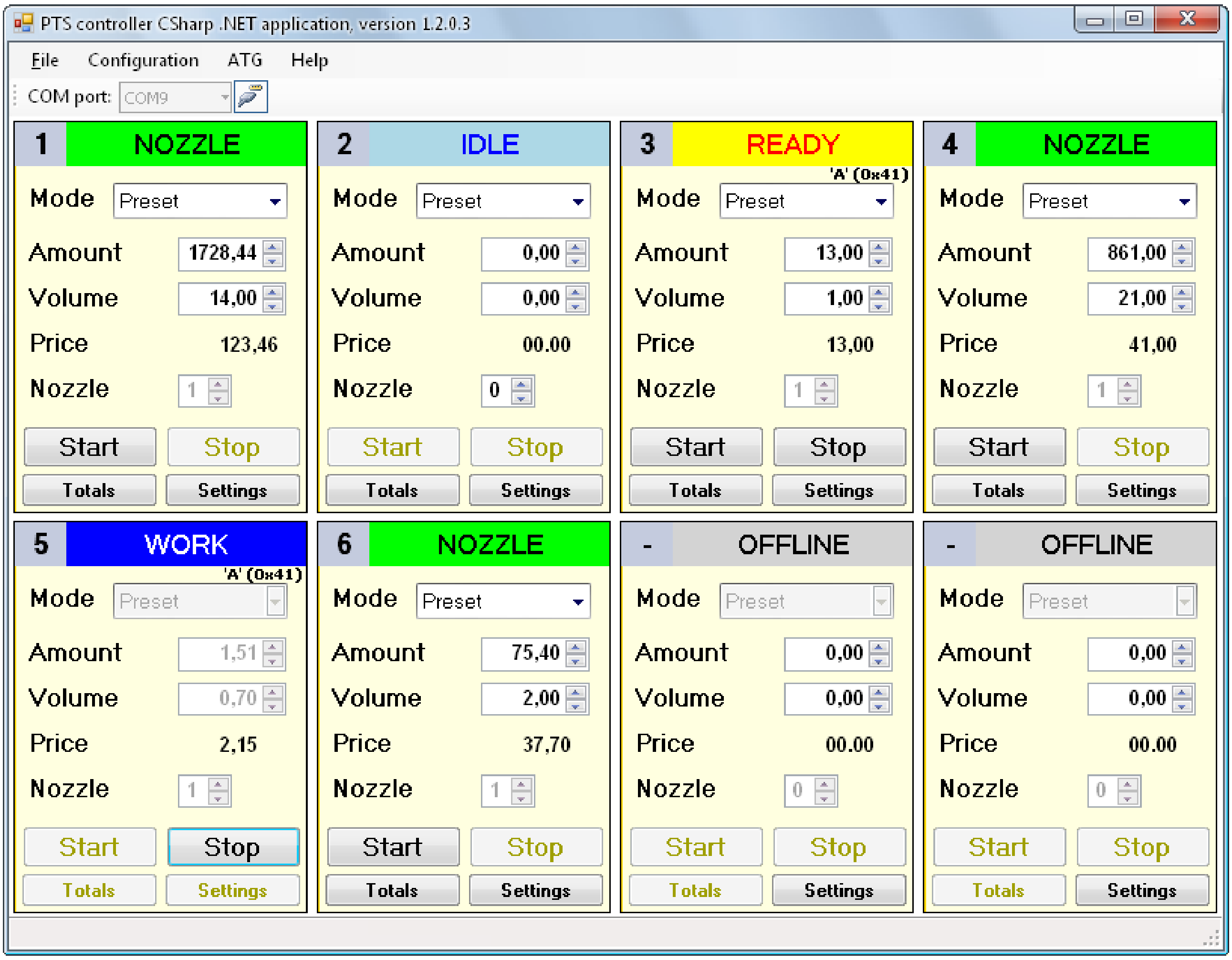
*Form for configuration of ATG systems channels of PTS controller*

## *Visual Basic .NET application*

The Visual Basic .NET application for PTS controller is written in VB.NET programming language in Visual Studio 2010 Express and requirs .NET Framework 2.0.

Given application includes 2 projects:

- PtsLib: project of PtsLib.dll library for operation with a PTS controller. Includes classes, methods and properties for communication with PTS controller using requests and responses described in UniPump communication protocol for communication with PTS controller for provision of control over dispensers and ATG systems and configuration of PTS controller.
- PTS: fully-functional open source application for provision of control over up to 8 pumps and 4 ATG systems by the PTS controller, allows to perform all operations with the PTS controller (uses classes and methods of PtsLib.dll library).

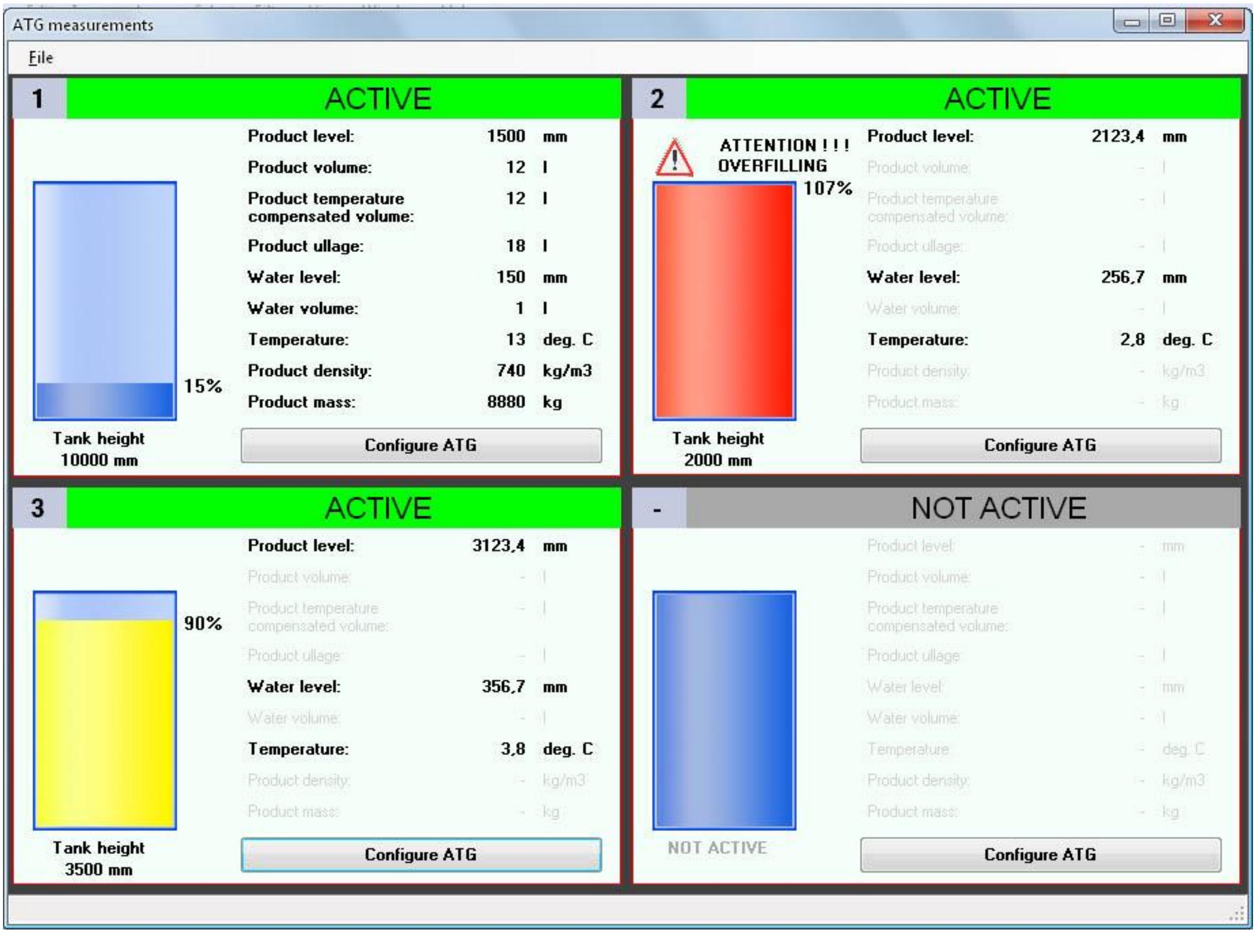


PTS controller CSharp .NET application, version 1.2.0.3

COM port: COM9

| | 1 | 2 | 3 | 4 | 5 | 6 | - | - |
|---|---|---|---|---|---|---|---|---|
| | NOZZLE | IDLE | READY | NOZZLE | WORK | NOZZLE | OFFLINE | OFFLINE |
| | | | 'A' (0x41) | | 'A' (0x41) | | | |
| Mode | Preset | Preset | Preset | Preset | Preset | Preset | Preset | Preset |
| Amount | 1728,44 | 0,00 | 13,00 | 861,00 | 1,51 | 75,40 | 0,00 | 0,00 |
| Volume | 14,00 | 0,00 | 1,00 | 21,00 | 0,70 | 2,00 | 0,00 | 0,00 |
| Price | 123,46 | 00.00 | 13,00 | 41,00 | 2,15 | 37,70 | 00.00 | 00.00 |
| Nozzle | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |

*General view of the application: allows to provide control over up to 8 pumps*



| 1 | ACTIVE | |
|---|---|---|
| Product level: | 1500 | mm |
| Product volume: | 12 | l |
| Product temperature compensated volume: | 12 | l |
| Product ullage: | 18 | l |
| Water level: | 150 | mm |
| Water volume: | 1 | l |
| Temperature: | 13 | deg. C |
| Product density: | 740 | kg/m3 |
| Product mass: | 8880 | kg |

15%
Tank height
10000 mm

| 2 | ACTIVE | |
|---|---|---|
| Product level: | 2123,4 | mm |
| Product volume: | - | l |
| Product temperature compensated volume: | - | l |
| Product ullage: | - | l |
| Water level: | 256,7 | mm |
| Water volume: | - | l |
| Temperature: | 2,8 | deg. C |
| Product density: | - | kg/m3 |
| Product mass: | - | kg |

ATTENTION ! ! !
OVERFILLING
107%
Tank height
2000 mm

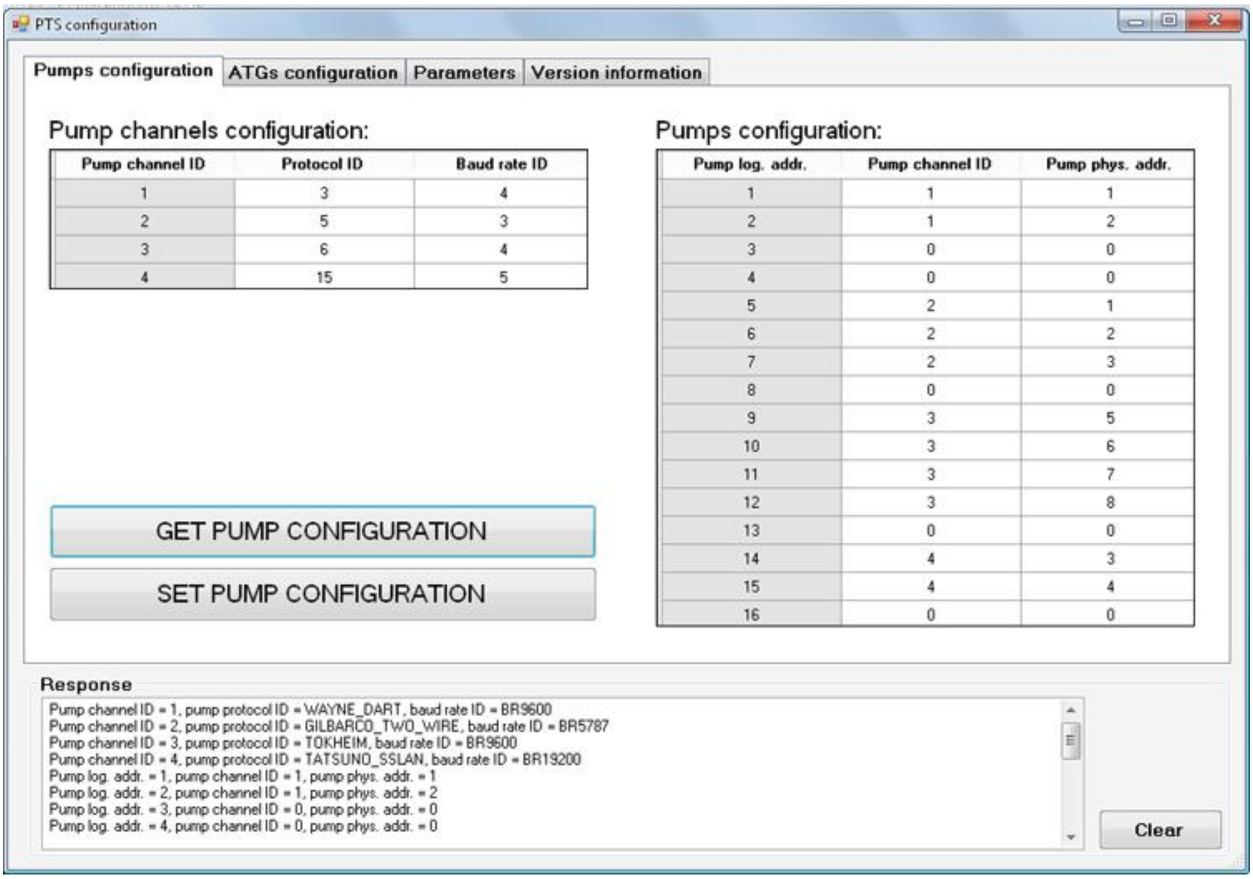
| 3 | ACTIVE | |
|---|---|---|
| Product level: | 3123,4 | mm |
| Product volume: | - | l |
| Product temperature compensated volume: | - | l |
| Product ullage: | - | l |
| Water level: | 356,7 | mm |
| Water volume: | - | l |
| Temperature: | 3,8 | deg. C |
| Product density: | - | kg/m3 |
| Product mass: | - | kg |

90%
Tank height
3500 mm

| - | NOT ACTIVE | |
|---|---|---|
| Product level: | - | mm |
| Product volume: | - | l |
| Product temperature compensated volume: | - | l |
| Product ullage: | - | l |
| Water level: | - | mm |
| Water volume: | - | l |
| Temperature: | - | deg. C |
| Product density: | - | kg/m3 |
| Product mass: | - | kg |

NOT ACTIVE

*Form of ATG systems monitoring: allows to monitor up to 4 ATG systems*



Pump channels configuration:

| Pump channel ID | Protocol ID | Baud rate ID |
|---|---|---|
| 1 | 3 | 4 |
| 2 | 5 | 3 |
| 3 | 6 | 4 |
| 4 | 15 | 5 |

Pumps configuration:

| Pump log. addr. | Pump channel ID | Pump phys. addr. |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 1 | 2 |
| 3 | 0 | 0 |
| 4 | 0 | 0 |
| 5 | 2 | 1 |
| 6 | 2 | 2 |
| 7 | 2 | 3 |
| 8 | 0 | 0 |
| 9 | 3 | 5 |
| 10 | 3 | 6 |
| 11 | 3 | 7 |
| 12 | 3 | 8 |
| 13 | 0 | 0 |
| 14 | 4 | 3 |
| 15 | 4 | 4 |
| 16 | 0 | 0 |

Response

```
Pump channel ID = 1, pump protocol ID = WAYNE_DART, baud rate ID = BR9600
Pump channel ID = 2, pump protocol ID = GILBARCO_TWO_WIRE, baud rate ID = BR5787
Pump channel ID = 3, pump protocol ID = TOKHEIM, baud rate ID = BR9600
Pump channel ID = 4, pump protocol ID = TATSUNO_SSLAN, baud rate ID = BR19200
Pump log. addr. = 1, pump channel ID = 1, pump phys. addr. = 1
Pump log. addr. = 2, pump channel ID = 1, pump phys. addr. = 2
Pump log. addr. = 3, pump channel ID = 0, pump phys. addr. = 0
Pump log. addr. = 4, pump channel ID = 0, pump phys. addr. = 0
```

*Form of PTS controller configuration: allows to set configuration for pumps and ATG systems channels and also set parameters*

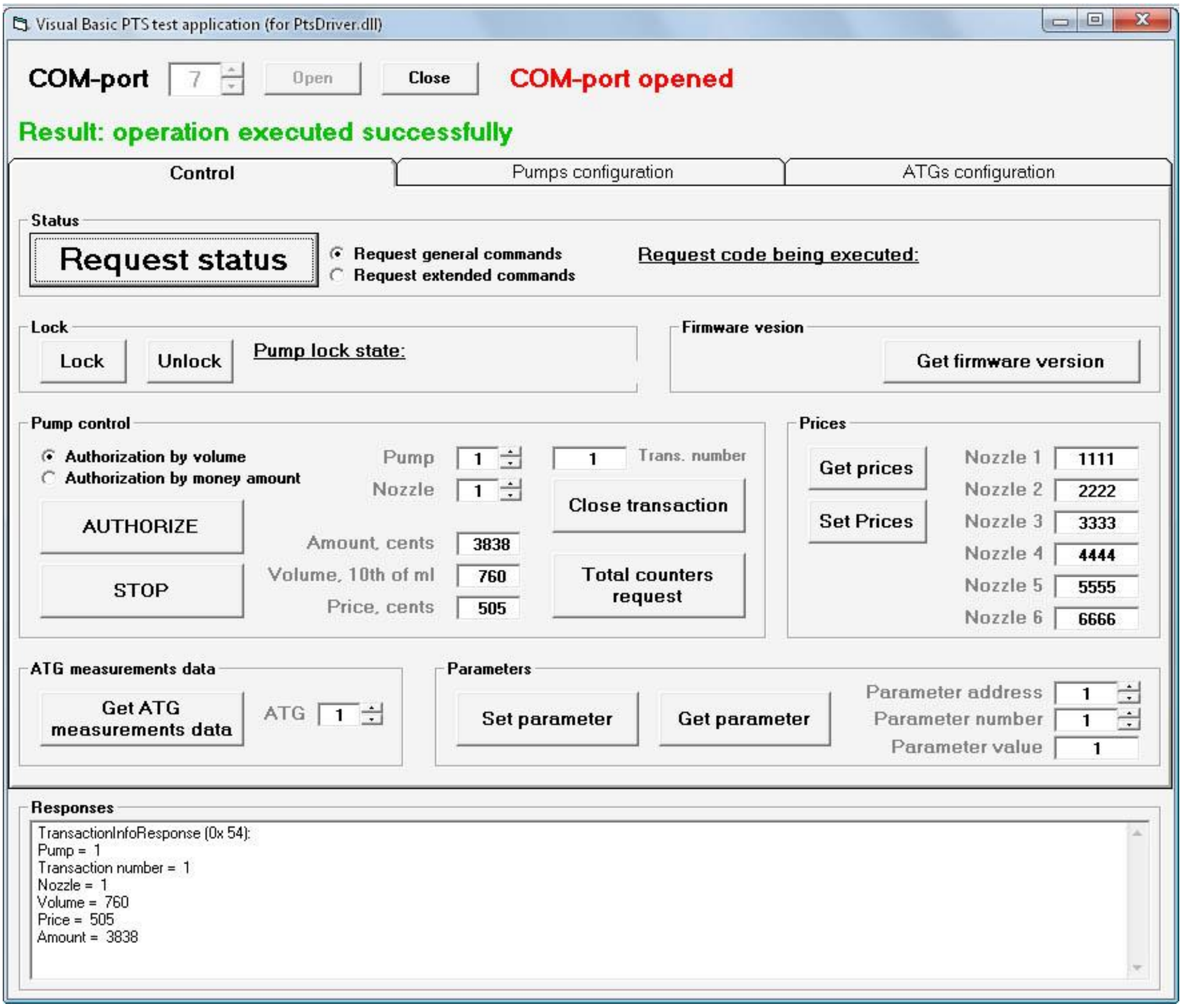
## COM-object for MS Windows OS

COM-object for PTS controller is written in Microsoft Visual C++ 6.0 IDE in C++ programming language. Project is located in PTSDriver folder. It builds PTSDriver.dll on output.

For communication with a PTS controller it uses API (application programming interface), written in C language (located in C_API folder).

## Visual Basic 6.0 application

For testing of the COM-object an open-source application in Microsoft Visual Basic 6.0 is written (located in VB_6.0_app folder). It provides GUI form for testing and indication of all operations provided by the PTS controller, allows to perform all operation with the PTS controller (uses classes and methods of PTSDriver).



COM-port 7

COM-port opened

Result: operation executed successfully

● Request general commands
○ Request extended commands

● Authorization by volume
○ Authorization by money amount

Pump 1
Nozzle 1
Trans. number 1
Amount, cents 3838
Volume, 10th of ml 760
Price, cents 505

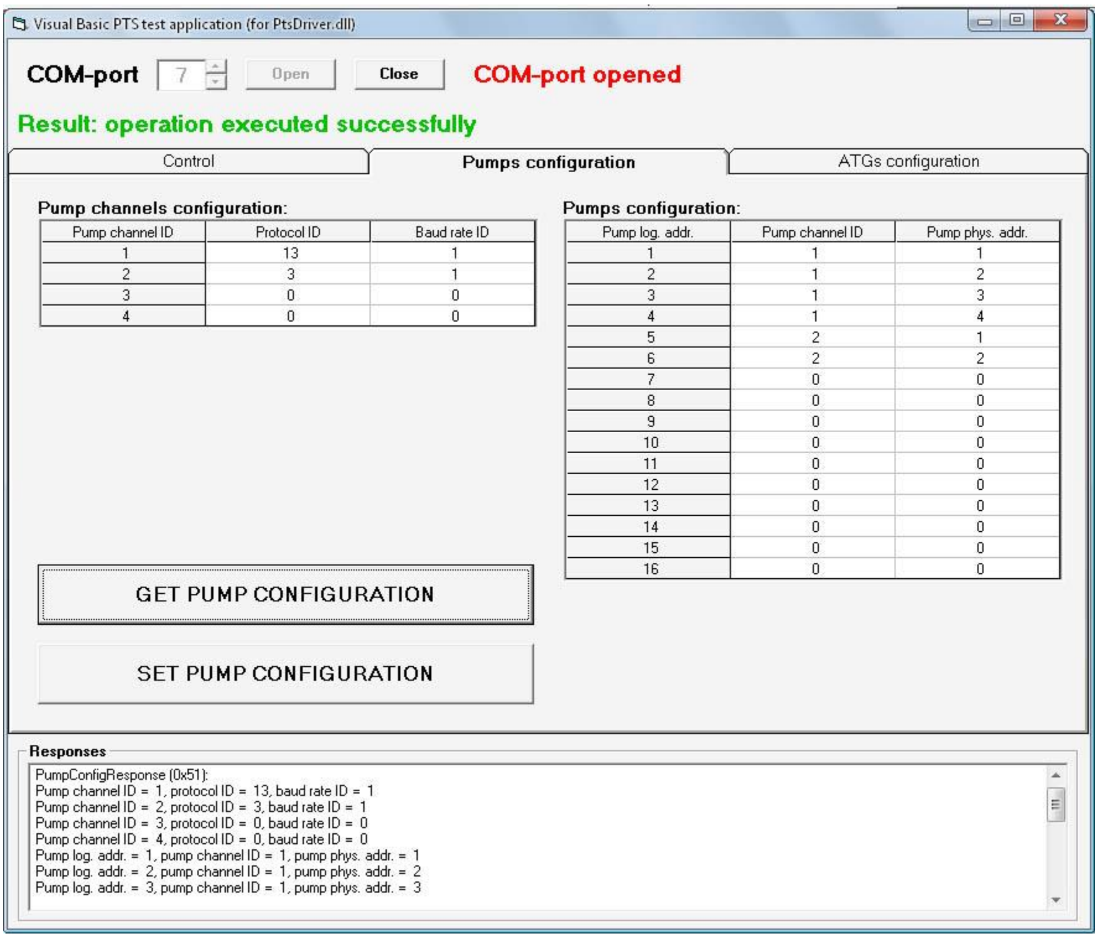
Nozzle 1 1111
Nozzle 2 2222
Nozzle 3 3333
Nozzle 4 4444
Nozzle 5 5555
Nozzle 6 6666

ATG 1

Parameter address 1
Parameter number 1
Parameter value 1

```
TransactionInfoResponse (0x 54):
Pump = 1
Transaction number = 1
Nozzle = 1
Volume = 760
Price = 505
Amount = 3838
```

*Form for testing and indication of all operations provided by the PTS controller*



Pump channels configuration:

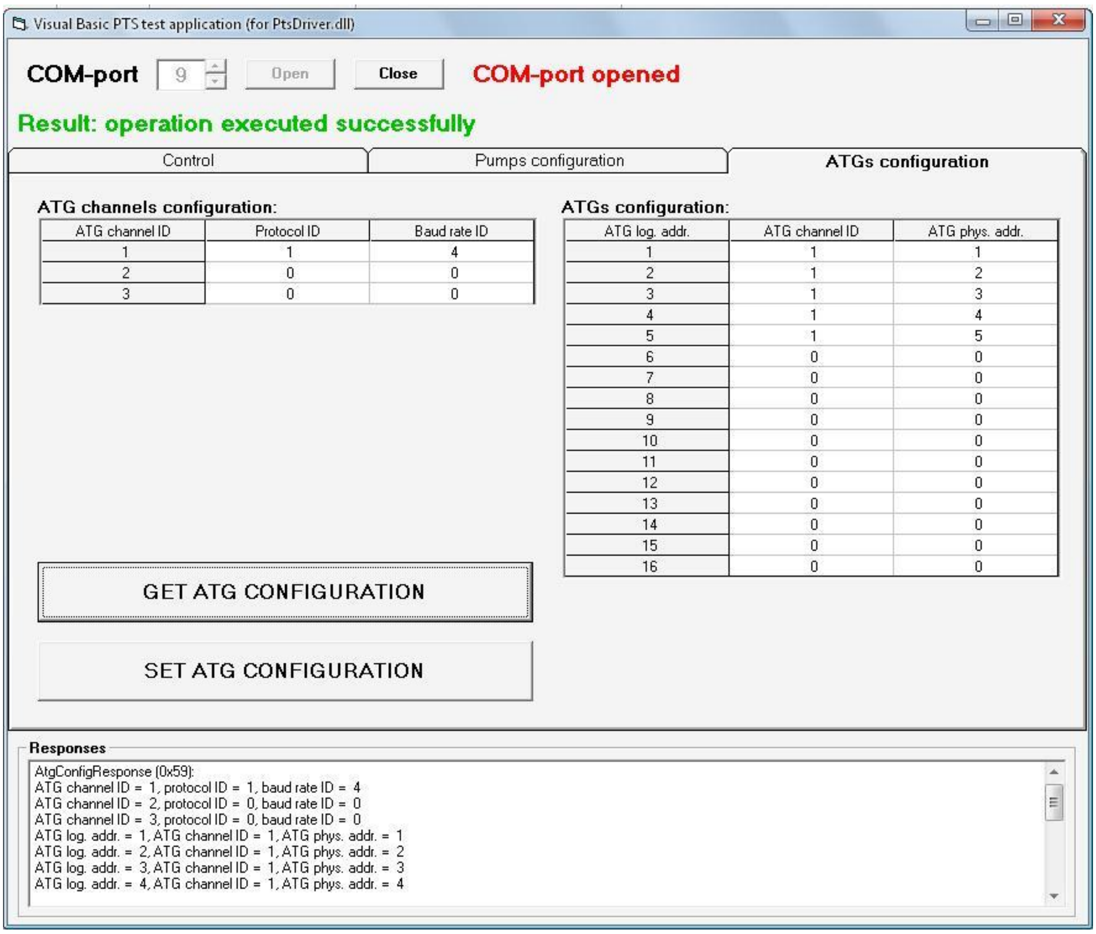
| Pump channel ID | Protocol ID | Baud rate ID |
|---|---|---|
| 1 | 13 | 1 |
| 2 | 3 | 1 |
| 3 | 0 | 0 |
| 4 | 0 | 0 |

Pumps configuration:

| Pump log. addr. | Pump channel ID | Pump phys. addr. |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 1 | 2 |
| 3 | 1 | 3 |
| 4 | 1 | 4 |
| 5 | 2 | 1 |
| 6 | 2 | 2 |
| 7 | 0 | 0 |
| 8 | 0 | 0 |
| 9 | 0 | 0 |
| 10 | 0 | 0 |
| 11 | 0 | 0 |
| 12 | 0 | 0 |
| 13 | 0 | 0 |
| 14 | 0 | 0 |
| 15 | 0 | 0 |
| 16 | 0 | 0 |

```
PumpConfigResponse (0x51):
Pump channel ID = 1, protocol ID = 13, baud rate ID = 1
Pump channel ID = 2, protocol ID = 3, baud rate ID = 1
Pump channel ID = 3, protocol ID = 0, baud rate ID = 0
Pump channel ID = 4, protocol ID = 0, baud rate ID = 0
Pump log. addr. = 1, pump channel ID = 1, pump phys. addr. = 1
Pump log. addr. = 2, pump channel ID = 1, pump phys. addr. = 2
Pump log. addr. = 3, pump channel ID = 1, pump phys. addr. = 3
```

*Form for configuration of pumps channels of PTS controller*



ATG channels configuration:

| ATG channel ID | Protocol ID | Baud rate ID |
|---|---|---|
| 1 | 1 | 4 |
| 2 | 0 | 0 |
| 3 | 0 | 0 |

ATGs configuration:

| ATG log. addr. | ATG channel ID | ATG phys. addr. |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 1 | 2 |
| 3 | 1 | 3 |
| 4 | 1 | 4 |
| 5 | 1 | 5 |
| 6 | 0 | 0 |
| 7 | 0 | 0 |
| 8 | 0 | 0 |
| 9 | 0 | 0 |
| 10 | 0 | 0 |
| 11 | 0 | 0 |
| 12 | 0 | 0 |
| 13 | 0 | 0 |
| 14 | 0 | 0 |
| 15 | 0 | 0 |
| 16 | 0 | 0 |

```
AtgConfigResponse (0x59):
ATG channel ID = 1, protocol ID = 1, baud rate ID = 4
ATG channel ID = 2, protocol ID = 0, baud rate ID = 0
ATG channel ID = 3, protocol ID = 0, baud rate ID = 0
ATG log. addr. = 1, ATG channel ID = 1, ATG phys. addr. = 1
ATG log. addr. = 2, ATG channel ID = 1, ATG phys. addr. = 2
ATG log. addr. = 3, ATG channel ID = 1, ATG phys. addr. = 3
ATG log. addr. = 4, ATG channel ID = 1, ATG phys. addr. = 4
```

*Form for configuration of ATG systems channels of PTS controller*

# PTS CONTROLLER SOFTWARE DEVELOPMENT KIT (SDK)

## *PTS controller SDK appointment*

**PTS controller SDK (Software Development Kit)** is appointed on developers of control system software applications (POS system, cash register, OPT terminal, etc) for petrol stations providing control over fuel dispensers and ATG systems through PTS controller.

PTS controller SDK includes a PTS controller and RS-232/RS-485 interface converter board, which allow to run software simulators of fuel dispensers and ATG systems on personal computer and debug operation of control systems software with PTS controller on it without a necessity to connect to real fuel dispensers and ATG systems. Thus it is convenient to work in office or at home rather than on a working petrol station being connected to real equipment.

**PTS controller SDK allows developers to:**

1. Study operation with the PTS controller.
2. Implement the PTS controller into own developed POS system or software application for control over fuel dispensers and ATG level measurement systems for petrol stations.
3. Debug own POS system or software application to work correctly with PTS controller by watching how NaftaPOS software provides it, implement the same behavior in own system.
4. Debug own software application to correctly provide control over various popular fuel dispensers locally (on the workplace) without a necessity to go to the petrol station and connect to real fuel dispensers using supplied with PTS SDK fuel dispensers software simulators, which simulate operation of various popular fuel dispensers. Through an interface converter RS-485 / RS-232, supplied together with PTS SDK, connect a PTS controller output RS-485 to a PC COM-port with fuel dispensers software simulators.

## *PTS controller SDK structure*

*HARDWARE*:

1. PTS controller over fuel dispensers and ATG systems for petrol stations - 1 pcs
2. Interface converter RS-485 / RS-232 - 1 pcs
3. USB software protection dongle Guardant (required for NaftaPOS software and fuel dispensers software simulators) - 1 pcs
4. Cabling

*SOFTWARE*:

1. NaftaPOS software for petrol stations – 1 license (protected by USB-dongle)
2. PumpDemo - free contracted version of NaftaPOS software for petrol stations
3. Software simulators of popular fuel dispensers
4. Software simulators of ATG systems (probes)
5. API for developers in various programming languages
6. Useful tools for debugging of communication with PTS controller

## *PTS controller SDK technical features*

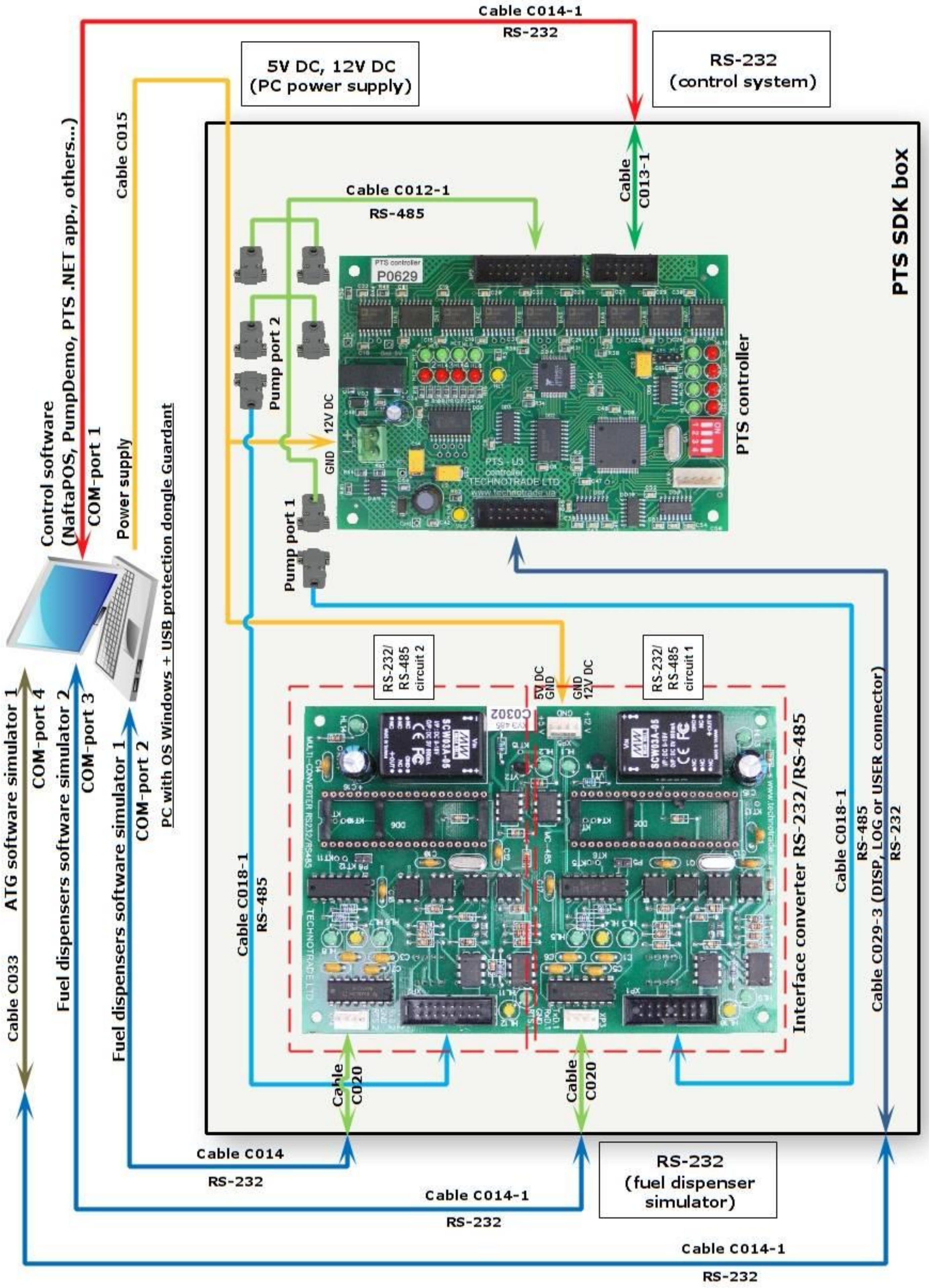
### *Specification*

| PARAMETER | VALUE |
|---|---|
| **Voltage** | +5 V, +12 V DC |
| **Current consumption** | 450 mA max |
| **Temperature range** | -40°C ÷ +80°C |
| **Weight** | 1700 g |
| **Overall dimensions** | 240 x 200 x 100 mm |

### *Communication ports*

| PARAMETER | VALUE |
|---|---|
| **PC port** | RS-232 |
| **Pump port** | RS-232 |

## *PTS controller SDK connections scheme*



PTS SDK box includes 2 boards mounted inside of a plastic case (PTS controller board and RS-232/RS-485 interface converter board) and all required cablings for power supply and communication of these boards.

Control systems software, which uses UniPump communication protocol, connects to the PTS controller SDK box through a connector with inscription “RS-232 (control system)” is connected.

Fuel dispenser software simulator is connected to the PTS controller SDK box through a connector with inscription “RS-232 (fuel dispenser simulator)”.

RS-232/RS-485 interface converter board is used in order to convert signals coming from PTS controller pump channels in RS-485 interface into signals of RS-232 interface for the fuel dispenser software simulator, which runs on PC and communicates through a COM-port. RS-232/RS-485 interface converter board has 2 independent conversion circuits, so it is possible to simultaneously connect 2 fuel dispenser’s simulators to PTS controller pump channels using the same RS-232/RS-485 interface converter board. Connection of pump port to RS-232/RS-485 interface converter board is made through cable C029 and C018-1. Connection of the RS-232/RS-485 interface converter board to PC COM-port is made through cables C020.

For connection to software simulators of ATG systems PTS controller does not need the RS-232/RS-485 interface converter board because ATG port is already in RS-232 interface and can be connected through cables C029-3 and C033 to the COM-port of PC, where the ATG systems software simulator is launched.

Cable C014 serves as a standard prolonger of RS-232 interface.

Thus it is possible to debug a correct operation of software, being developed, with the PTS controller without a necessity to connect to real fuel dispensers and ATG systems, but using fuel dispenser’s software simulators and ATG system’s software simulators.

# *RS-232/RS-485 interface converter purpose*

## *Appointment*

**RS-232 to RS-485 interface converter** is intended for conversion of signals from RS-232 interface to signals of RS-485 interface and backwards. Converter contains 2 separate independent conversion channels.

More technical information regarding interface converter RS-232/RS-485 can be found in technical guide on: http://www.technotrade.ua/rs232_to_rs485_converter.html.

## *Specification*

| PARAMTER | VALUE |
|---|---|
| **Voltage** | +12 V DC, +5 V DC |
| **Current consumption** | 250 mA max |
| **Temperature range** | 0°C ÷ +40°C |
| **Weight** | 120 g |
| **Dimensions** | 145 x 100 x 20 mm |

## *Communication ports*

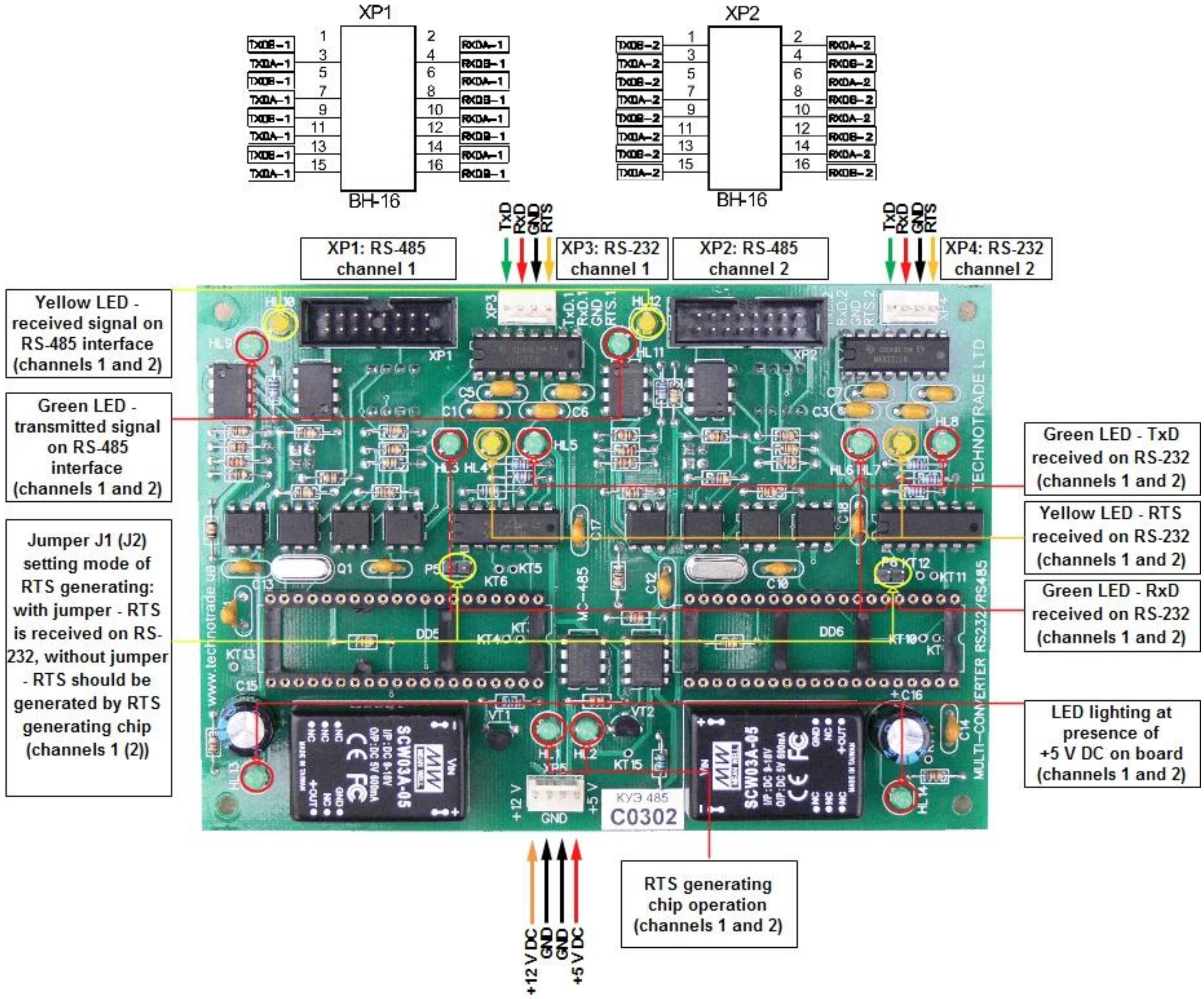
| PARAMTER | VALUE |
|---|---|
| **RS-232 port** | RS-232 (can be with RTS control) |
| **RS-485 port** | Half-duplex (2-wire) or full-duplex (4-wire) |

## *Features*

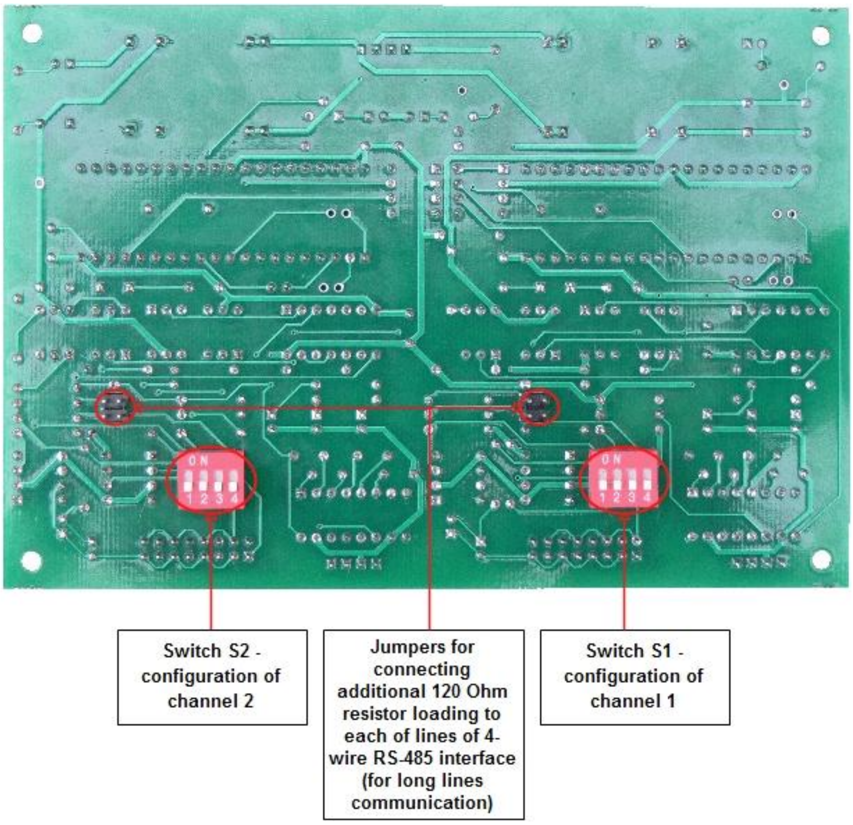
| PARAMTER | VALUE |
|---|---|
| **Channels** | Converter contains 2 separate independent channels |
| **Galvanic isolation** | Receiver and transmitter are galvanically isolated |
| **RTS control** | RS-232 port can receive RTS control, for cases when it does not receive RTS control a RTS generating chip should be installed |

## *RS-232/RS-485 interface converter PCB board connectors overview*

### *Top view*


XP1
1 TXDB-1
3 TXDA-1
5 TXDB-1
7 TXDA-1
9 TXDB-1
11 TXDA-1
13 TXDB-1
15 TXDA-1
2 RXDA-1
4 RXDB-1
6 RXDA-1
8 RXDB-1
10 RXDA-1
12 RXDB-1
14 RXDA-1
16 RXDB-1
BH-16
XP2
1 TXDB-2
3 TXDA-2
5 TXDB-2
7 TXDA-2
9 TXDB-2
11 TXDA-2
13 TXDB-2
15 TXDA-2
2 RXDA-2
4 RXDB-2
6 RXDA-2
8 RXDB-2
10 RXDA-2
12 RXDB-2
14 RXDA-2
16 RXDB-2
BH-16
TxD
RxD
GND
RTS
XP1: RS-485 channel 1
XP3: RS-232 channel 1
XP2: RS-485 channel 2
TxD
RxD
GND
RTS
XP4: RS-232 channel 2
Yellow LED - received signal on RS-485 interface (channels 1 and 2)
Green LED - transmitted signal on RS-485 interface (channels 1 and 2)
Jumper J1 (J2) setting mode of RTS generating: with jumper - RTS is received on RS-232, without jumper - RTS should be generated by RTS generating chip (channels 1 (2))
Green LED - TxD received on RS-232 (channels 1 and 2)
Yellow LED - RTS received on RS-232 (channels 1 and 2)
Green LED - RxD received on RS-232 (channels 1 and 2)
LED lighting at presence of +5 V DC on board (channels 1 and 2)
RTS generating chip operation (channels 1 and 2)
+12 V DC
GND
GND
+5 V DC

### *Back view*


Switch S2 - configuration of channel 2
Jumpers for connecting additional 120 Ohm resistor loading to each of lines of 4-wire RS-485 interface (for long lines communication)
Switch S1 - configuration of channel 1
ON
1 2 3 4
ON
1 2 3 4

## RS-232/RS-485 interface converter operation

Signals of RS-232 interface enter the MAX232 chip, where are converted into signals of TTL logics, and after conversion go through optocouplers H11L1, that serve for galvanic isolation of transmitter and receiver, enter the TTL input of MAX485 chip, from the output of which signals of RS-485 interface are received. Analogous behavior takes places in contrary direction of signals.

The board of the Converter has 2 separate channels, the only thing which unites them is source voltage +5 V, which feeds interface RS-232. Voltage +5 V for circuits of RS-485 interface of the channels is taken from separate DC-DC converters with galvanical isolation for the channels.

Depending on the state of switches Sn (where n=1,2 – number of channel) output of RS-485 interface can be adjusted for operation in half-duplex or full-duplex modes (2-wire or 4-wire schemes of connection accordingly). When switches Sn:1, Sn:3, Sn:4 (where n=1,2 – number of channel) are in position “OFF” – the Converter works in full-duplex (4-wire) mode of RS-485 interface, in position “ON” - the Converter works in half-duplex (2-wire) mode of RS-485 interface. Switch Sn:2 (where n=1,2 – number of channel) is intended for switching on (position “OFF”) or turning off (position “ON”) of echo mode in half-duplex (2-wire) mode of RS-485 interface (repeating of all signals, received on input).

Half-duplex (2-wire) mode of RS-485 interface requires presence of RTS (Request To Send) signal on the input of RS-232 interface for synchronization of transmit/receive. In case if RTS signal is impossible to receive from the source of signal of RS-232 interface, then it is possible to generate it with a help of RTS generating chips of each channel, which are installed only in case of such necessity. If input of RS-232 interface contains RTS signals and RTS generating chips are not installed then it is necessary to place a jumper Jn (where n=1,2 – number of channel) for the required channel, in contrary case the jumper should be removed.

# *USB software protection dongle Guardant*

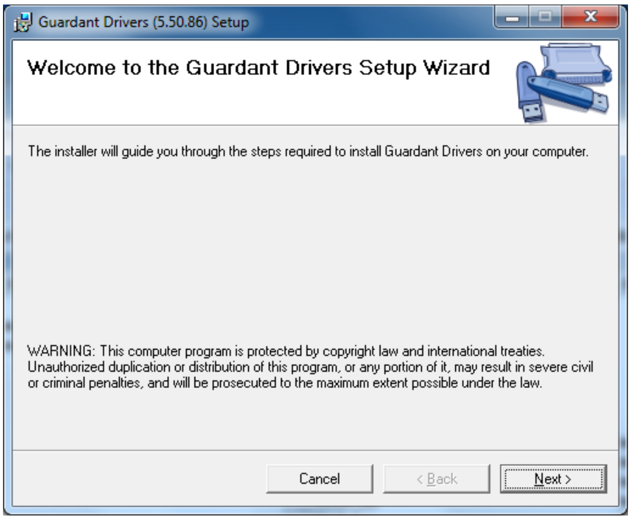
## *Appointment*

**USB protection dongle Guardant** is required for running of NaftaPOS software and fuel dispensers software simulators. To run this software it is necessary to have the dongle inserted in the USB port. Electronic protection dongle Guardant – is a device that is connected to USB port (or a parallel port LPT in older versions), located on a panel of a PC motherboard. Electronic protection dongle Guardant is necessary for protection of NaftaPOS software and fuel dispensers software simulators from unauthorized copying and illegal distribution. Scope of protection is that the dongle contains part of the code of protected software. Without a dongle being inserted to PC protected software will not launch and correspondent error will be shown to user.

## *Installation*

To install Guardant dongle drivers run GrdDriversEN.msi.

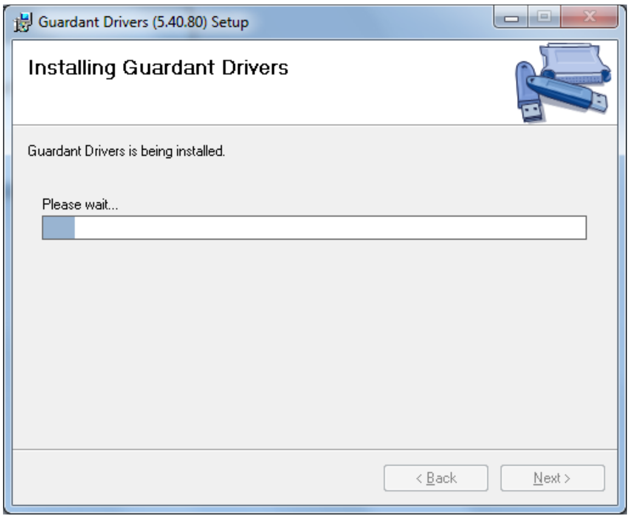
Guardant USB dongle drivers are installed during NaftaPOS software installation process or can be installed/updated any time later:


Guardant Drivers (5.50.86) Setup

Welcome to the Guardant Drivers Setup Wizard

The installer will guide you through the steps required to install Guardant Drivers on your computer.

WARNING: This computer program is protected by copyright law and international treaties. Unauthorized duplication or distribution of this program, or any portion of it, may result in severe civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.


Guardant Drivers (5.40.80) Setup

Installing Guardant Drivers

Guardant Drivers is being installed.

Please wait...

During installation process it is necessary to follow all steps in the installation wizard until installation is completed.

# *NaftaPOS software for petrol stations*

## *Appointment*

Front-Office software NaftaPOS is intended for control over technological processes, automated measurement and commercial account at petrol stations (also called fuel stations or gas stations) and petroleum storage depots.

Area of application – petrol stations for dispensing of light petroleum products, diesel fuel, liquified petroleum gas (LPG), compressed natural gas (CNG), multifuel petrol stations, petroleum storage depots.

NaftaPOS system is developed with account of all modern requirements to systems for automation of petrol stations, at this it is oriented as on small petrol stations, so as on big petrol complexes with organization of several working places of operators and administrator (Back-Office) of a petrol station.

User Manual on NaftaPOS software system can be downloaded from NaftaPOS software web-page http://www.technotrade.ua/petrol_station_software.html. it is strongly recommended to read User Manual on NaftaPOS software prior to installation.

More technical information regarding NaftaPOS software can be found on NaftaPOS software web-page: http://www.technotrade.ua/petrol_station_software.html

## *Installation*

There are 2 types of installation files for NaftaPOS system:

- 32-bit version
- 64-bit version

Installation file, which corresponds to hardware characteristics of the POS terminal, should be used for installation of the system.

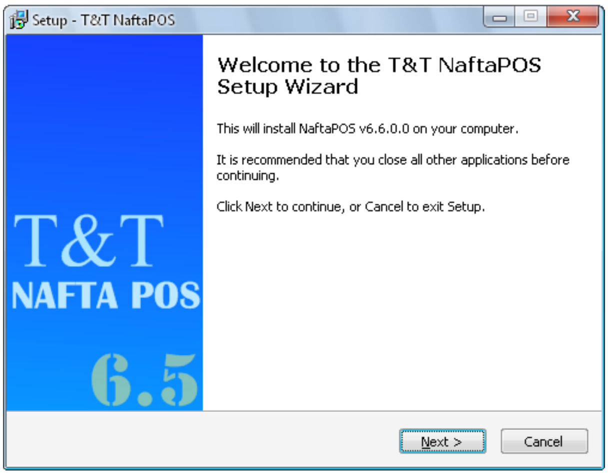
NaftaPOS software is to be installed on a POS terminal (or a standard PC) with Microsoft Windows operation system (supported OS Windows 2000, XP, Embedded POSReady 2009, Vista, 7, 8).

At using NaftaPOS you should in Windows Control Panel in configuration of regional parameters on tabs "Numbers" and "Currency unit" as a "Separator of integer and fractional parts" set '.' (point, dot).

Minimal requirements to a PC include the following:

- 1 GHz CPU
- 1 GB RAM
- 60 GB HDD

Installation of NaftaPOS software is made by running an installation file following instructions, specified in installation wizard windows:



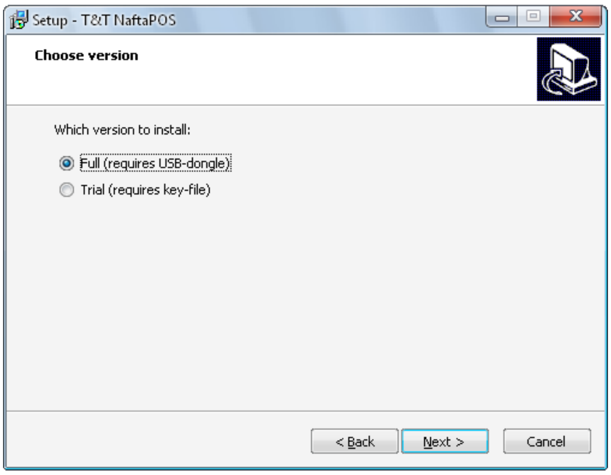
Setup - T&T NaftaPOS

Welcome to the T&T NaftaPOS Setup Wizard

This will install NaftaPOS v6.6.0.0 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.



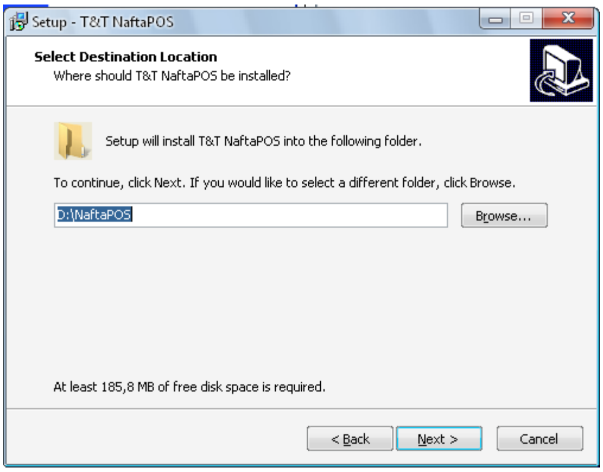
Setup - T&T NaftaPOS

Choose version

Which version to install:

● Full (requires USB-dongle)

○ Trial (requires key-file)



Setup - T&T NaftaPOS

Select Destination Location

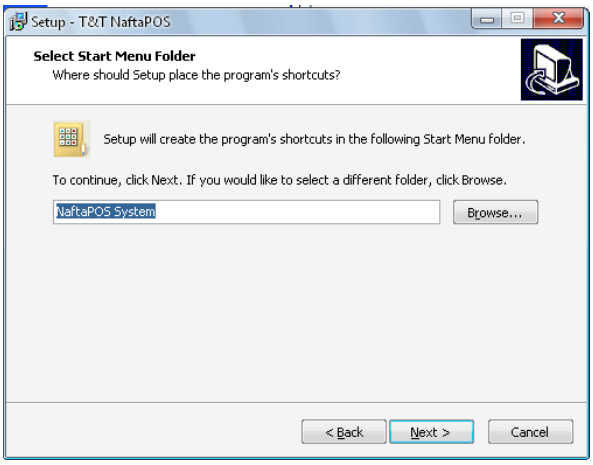
Where should T&T NaftaPOS be installed?

Setup will install T&T NaftaPOS into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

D:\NaftaPOS

At least 185,8 MB of free disk space is required.



Setup - T&T NaftaPOS

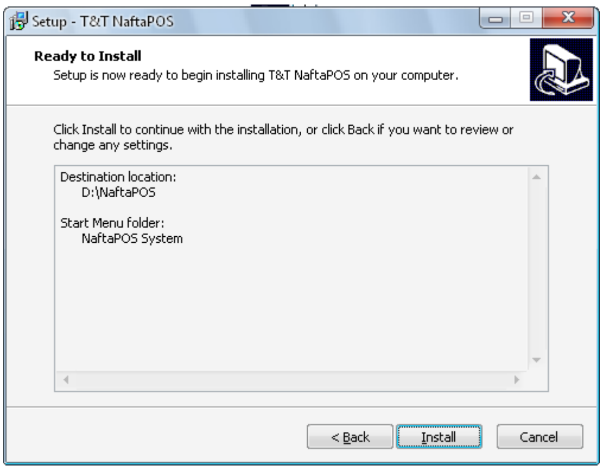
Select Start Menu Folder

Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

NaftaPOS System



Setup - T&T NaftaPOS

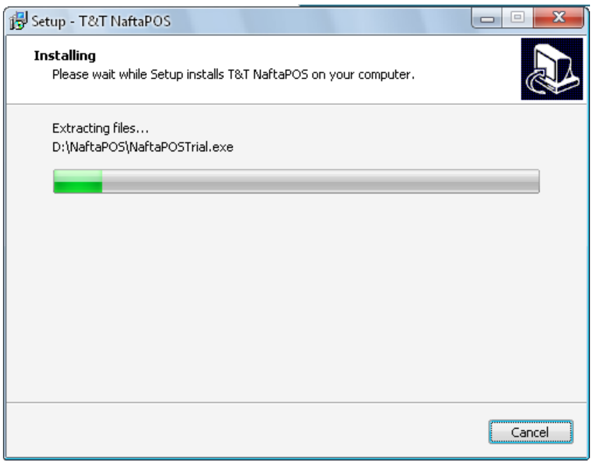
Ready to Install

Setup is now ready to begin installing T&T NaftaPOS on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

Destination location:
D:\NaftaPOS

Start Menu folder:
NaftaPOS System

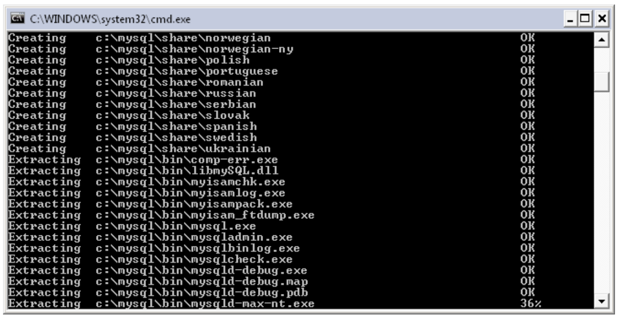


Setup - T&T NaftaPOS

Installing

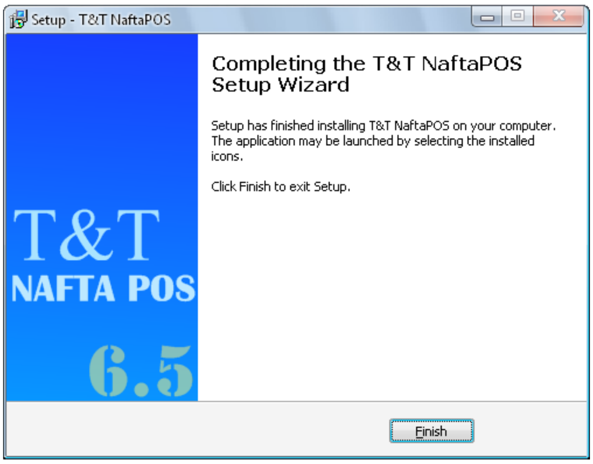
Please wait while Setup installs T&T NaftaPOS on your computer.

Extracting files...
D:\NaftaPOS\NaftaPOSTrial.exe



C:\WINDOWS\system32\cmd.exe

```
Creating    c:\mysql\share\norwegian                 OK
Creating    c:\mysql\share\norwegian-ny              OK
Creating    c:\mysql\share\polish                    OK
Creating    c:\mysql\share\portuguese                OK
Creating    c:\mysql\share\romanian                  OK
Creating    c:\mysql\share\russian                   OK
Creating    c:\mysql\share\serbian                   OK
Creating    c:\mysql\share\slovak                    OK
Creating    c:\mysql\share\spanish                   OK
Creating    c:\mysql\share\swedish                   OK
Creating    c:\mysql\share\ukrainian                 OK
Extracting  c:\mysql\bin\comp-err.exe                OK
Extracting  c:\mysql\bin\libmySQL.dll                OK
Extracting  c:\mysql\bin\myisamchk.exe               OK
Extracting  c:\mysql\bin\myisamlog.exe               OK
Extracting  c:\mysql\bin\myisampack.exe              OK
Extracting  c:\mysql\bin\myisam_ftdump.exe           OK
Extracting  c:\mysql\bin\mysql.exe                   OK
Extracting  c:\mysql\bin\mysqladmin.exe              OK
Extracting  c:\mysql\bin\mysqlbinlog.exe             OK
Extracting  c:\mysql\bin\mysqlcheck.exe              OK
Extracting  c:\mysql\bin\mysqld-debug.exe            OK
Extracting  c:\mysql\bin\mysqld-debug.map            OK
Extracting  c:\mysql\bin\mysqld-debug.pdb            OK
Extracting  c:\mysql\bin\mysqld-max-nt.exe           36%
```



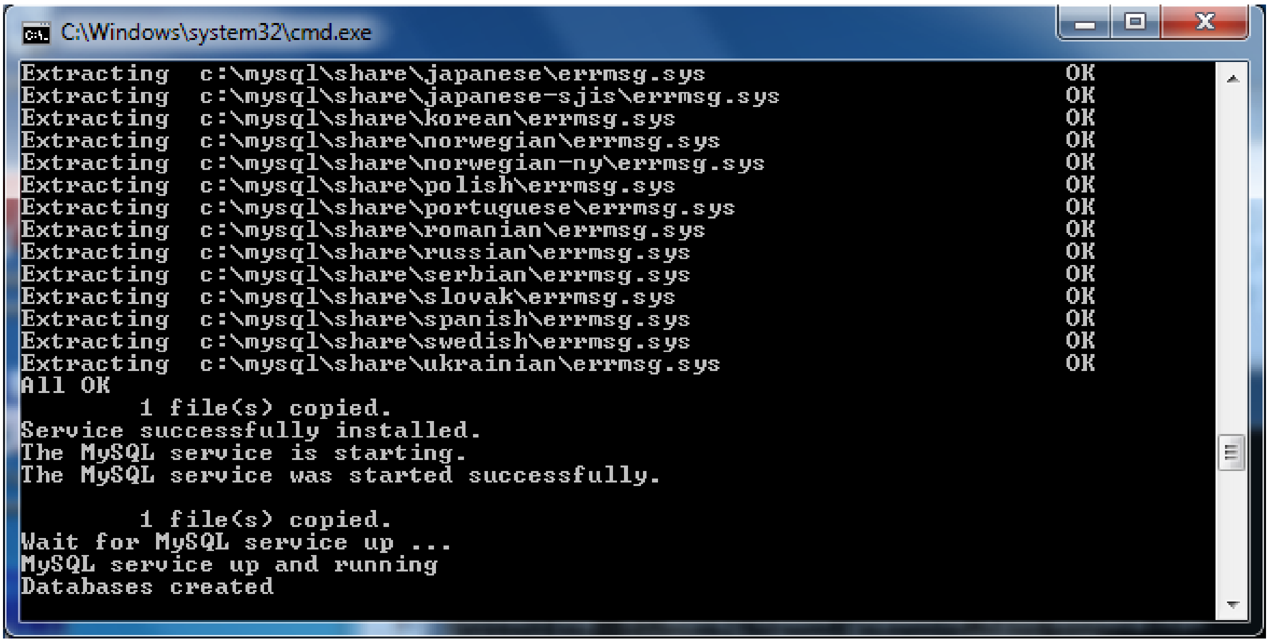
Setup - T&T NaftaPOS

Completing the T&T NaftaPOS Setup Wizard

Setup has finished installing T&T NaftaPOS on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

During installation process MySQL RDMS will be installed (starting MySQL service):

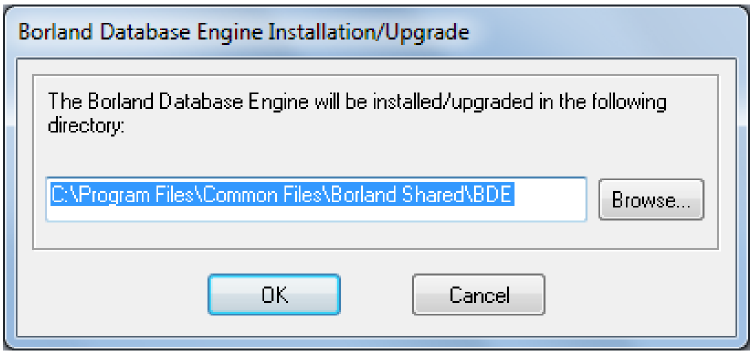


```
Extracting  c:\mysql\share\japanese\errmsg.sys                  OK
Extracting  c:\mysql\share\japanese-sjis\errmsg.sys             OK
Extracting  c:\mysql\share\korean\errmsg.sys                    OK
Extracting  c:\mysql\share\norwegian\errmsg.sys                 OK
Extracting  c:\mysql\share\norwegian-ny\errmsg.sys              OK
Extracting  c:\mysql\share\polish\errmsg.sys                    OK
Extracting  c:\mysql\share\portuguese\errmsg.sys                OK
Extracting  c:\mysql\share\romanian\errmsg.sys                  OK
Extracting  c:\mysql\share\russian\errmsg.sys                   OK
Extracting  c:\mysql\share\serbian\errmsg.sys                   OK
Extracting  c:\mysql\share\slovak\errmsg.sys                    OK
Extracting  c:\mysql\share\spanish\errmsg.sys                   OK
Extracting  c:\mysql\share\swedish\errmsg.sys                   OK
Extracting  c:\mysql\share\ukrainian\errmsg.sys                 OK
All OK
        1 file(s) copied.
Service successfully installed.
The MySQL service is starting.
The MySQL service was started successfully.

        1 file(s) copied.
Wait for MySQL service up ...
MySQL service up and running
Databases created
```

NaftaPOS installation process – MySQL DBMS installation

Borland Database Engine will be installed during NaftaPOS software installation process, so it is important to agree with a request on its installation:

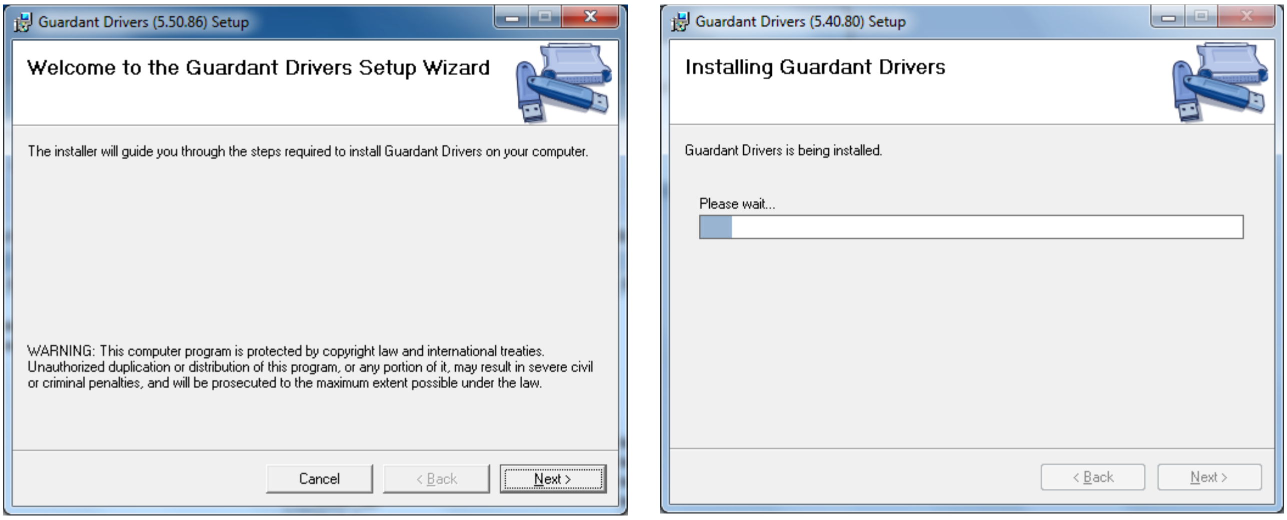


The Borland Database Engine will be installed/upgraded in the following directory:

C:\Program Files\Common Files\Borland Shared\BDE

NaftaPOS installation process – Borland Database Engine installation

Guardant dongle drivers are installed during NaftaPOS software installation process, so it is important to agree with a request on its installation:


Guardant Drivers (5.50.86) Setup
Cancel
< Back
Next >
Guardant Drivers (5.40.80) Setup
< Back
Next >

Welcome to the Guardant Drivers Setup Wizard

The installer will guide you through the steps required to install Guardant Drivers on your computer.

WARNING: This computer program is protected by copyright law and international treaties. Unauthorized duplication or distribution of this program, or any portion of it, may result in severe civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Installing Guardant Drivers

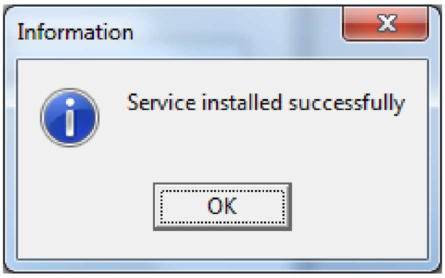
Guardant Drivers is being installed.

Please wait...

Installation process – Guardant dongle drivers installation

During installation the following NaftaPOS services will be installed:

- T&T database server
- T&T Scheduler


Information

Service installed successfully

*NaftaPOS installation process – service installation*

After installation of NaftaPOS software a logical drive, on which the system is installed, will contain a directory “NaftaPOS” with all files of NaftaPOS software and NaftaPOS system files that track its operation (log-files):

- main_logfile.txt
- controller.txt
- CashErr.txt,

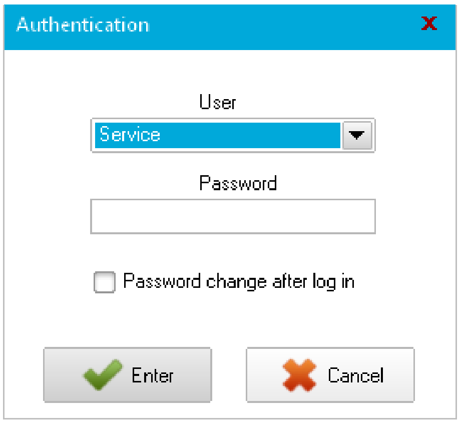
Others log files may be created automatically during the system operation. By content of these log-files it is possible to track reasons of possible errors and misoperations arisen during software operation and lead a log of actions performed by operator of the petrol station. This information is written in logs that at necessity can be sent to TECHNOTRADE LTD company for defining of found errors and their initial reasons with subsequent fixing of these errors.

## *Configuration of NaftaPOS software to work with PTS controller*

For configuration of NaftaPOS software it is necessary to have administrator rights in NaftaPOS system (these rights are set in NaftaPOS configuration, person without rights can not configure NaftaPOS).

Configuration of NaftaPOS software is made in configuration utility POSSet.exe (located in NaftaPOS\POSSet.exe folder), which can also be run from NaftaPOS software by selecting an item from main menu ‘Configuration’ (in case if the configuration utility was started from inside the NaftaPOS software – then for the made adjustments to enter into force it is necessary to restart NaftaPOS system).

At entering the configuration utility a user is requested to authenticate in order to prove his permission to make configurations of the system:


Authentication

User

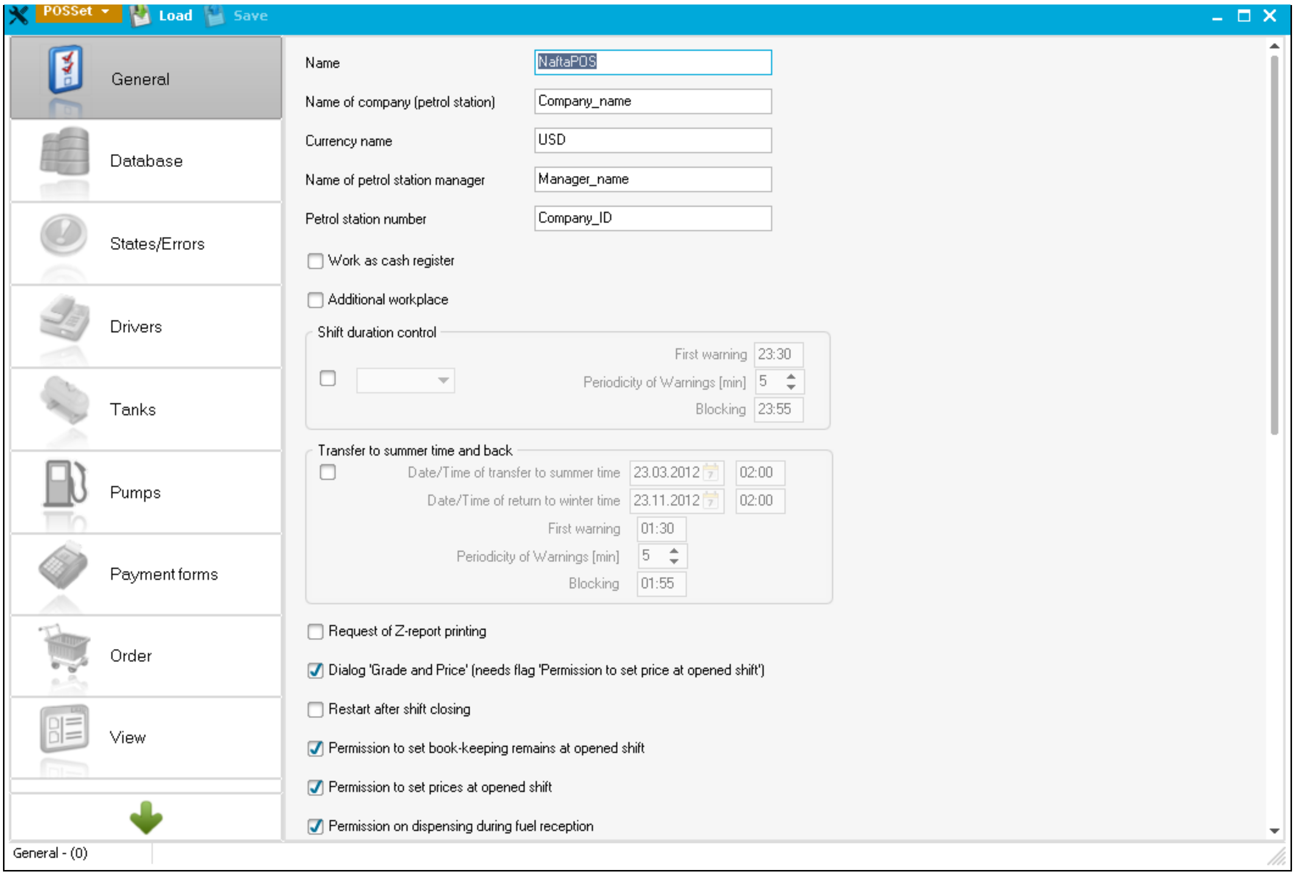
Service

Password

☐ Password change after log in

By defaults after installation there is only 1 user present with login ‘Service’ and no password (adding of other users and changing of passwords is made in configuration utility).

After logging inside the configuration utility the following window is shown, where on the left side – there is a selector of tabs to be configured, and on the right side – configurable parameters of the selected tab.



Name: NaftaPOS

Name of company (petrol station): Company_name

Currency name: USD

Name of petrol station manager: Manager_name

Petrol station number: Company_ID

□ Work as cash register

□ Additional workplace

Shift duration control

□ First warning 23:30; Periodicity of Warnings [min] 5; Blocking 23:55

Transfer to summer time and back

□ Date/Time of transfer to summer time 23.03.2012 02:00

Date/Time of return to winter time 23.11.2012 02:00

First warning 01:30

Periodicity of Warnings [min] 5

Blocking 01:55

□ Request of Z-report printing

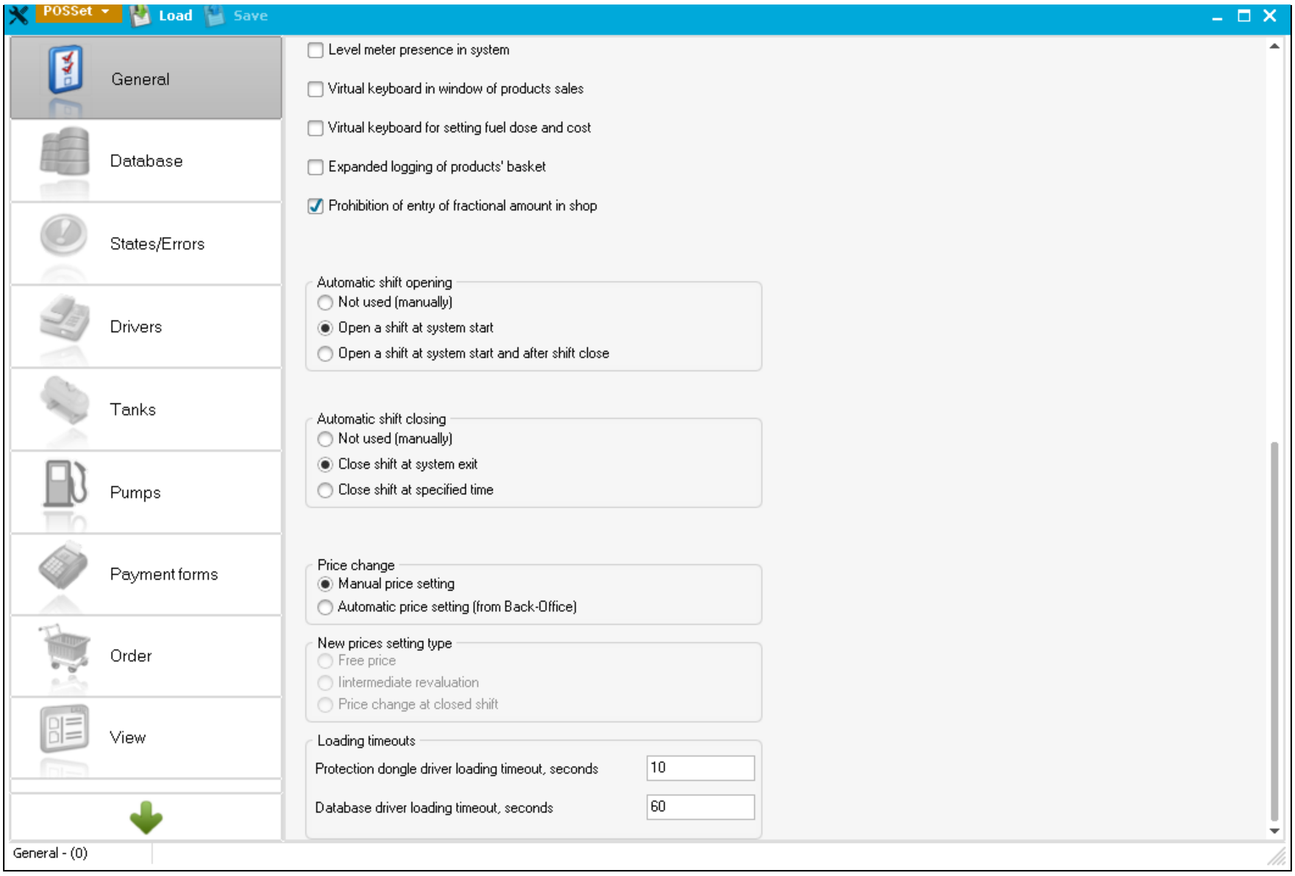
☑ Dialog 'Grade and Price' (needs flag 'Permission to set price at opened shift')

□ Restart after shift closing

☑ Permission to set book-keeping remains at opened shift

☑ Permission to set prices at opened shift

☑ Permission on dispensing during fuel reception



□ Level meter presence in system

□ Virtual keyboard in window of products sales

□ Virtual keyboard for setting fuel dose and cost

□ Expanded logging of products' basket

☑ Prohibition of entry of fractional amount in shop

Automatic shift opening

○ Not used (manually)

● Open a shift at system start

○ Open a shift at system start and after shift close

Automatic shift closing

○ Not used (manually)

● Close shift at system exit

○ Close shift at specified time

Price change

● Manual price setting

○ Automatic price setting (from Back-Office)

New prices setting type

○ Free price

○ Iintermediate revaluation

○ Price change at closed shift

Loading timeouts

Protection dongle driver loading timeout, seconds: 10

Database driver loading timeout, seconds: 60

Initial configuration of NaftaPOS for operation with PTS controller should include the following steps:

## 1. Configuration of tab 'General'

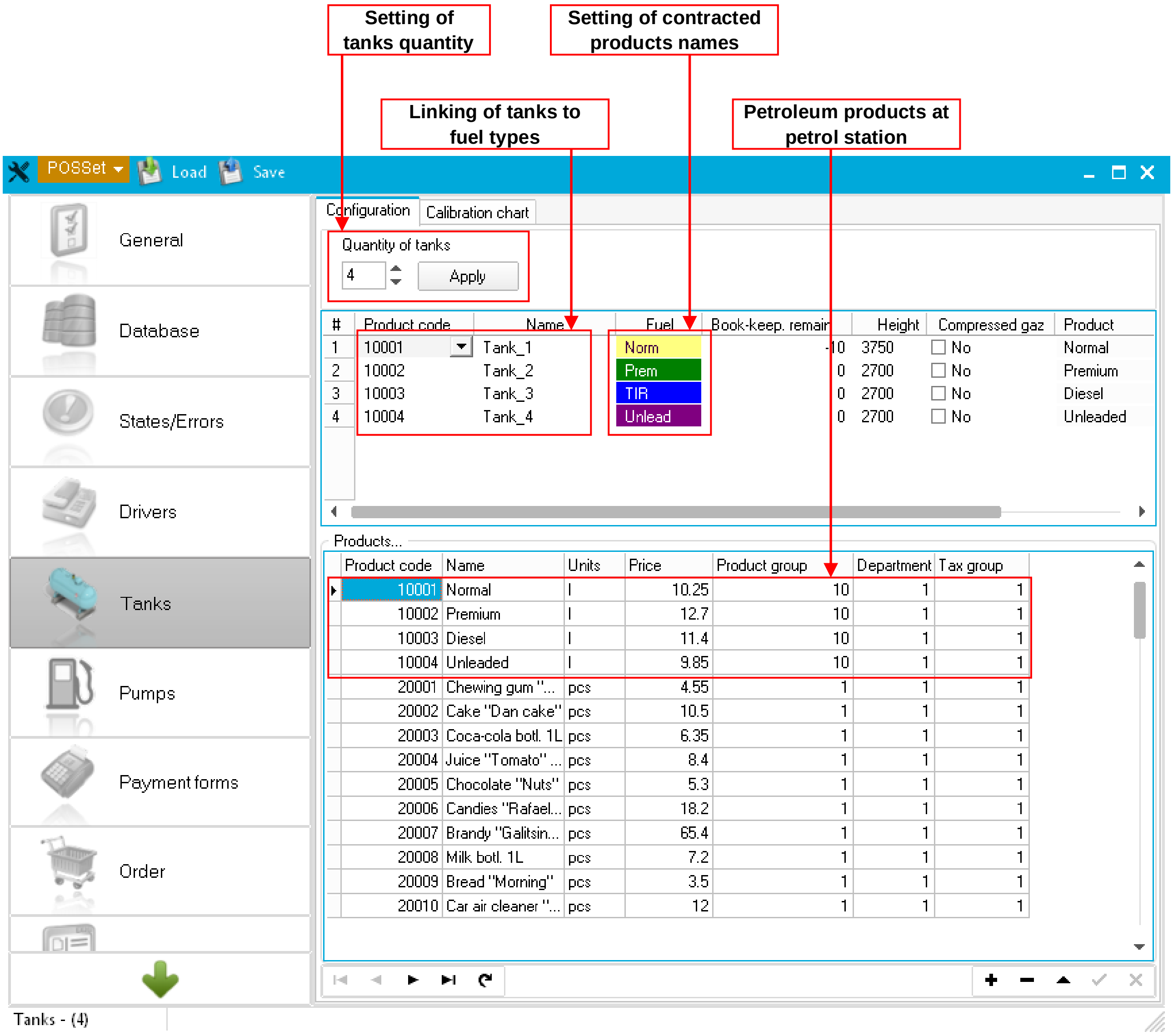
Make sure that the following checkboxes are checked:

- Dialog 'Grades and Prices' (needs flag 'Permission to set prices at opened session')
- Permission to set book-keeping remains at opened shift
- Permission to set prices at opened session
- Permission on dispensing during fuel reception
- Level meter presence in system

Other checkboxes can be left unchecked.
Additionally check 'Manual price setting' in Price change panel.

## 2. Configuration of tab "Tanks"

Make sure that you have the tanks quantity set and tanks are linked to fuel types:



Setting of tanks quantity

Setting of contracted products names

Linking of tanks to fuel types

Petroleum products at petrol station

Quantity of tanks: 4 Apply

| # | Product code | Name | Fuel | Book-keep. remain | Height | Compressed gaz | Product |
|---|---|---|---|---|---|---|---|
| 1 | 10001 | Tank_1 | Norm | -10 | 3750 | □ No | Normal |
| 2 | 10002 | Tank_2 | Prem | 0 | 2700 | □ No | Premium |
| 3 | 10003 | Tank_3 | TIR | 0 | 2700 | □ No | Diesel |
| 4 | 10004 | Tank_4 | Unlead | 0 | 2700 | □ No | Unleaded |

Products...

| Product code | Name | Units | Price | Product group | Department | Tax group |
|---|---|---|---|---|---|---|
| 10001 | Normal | l | 10.25 | 10 | 1 | 1 |
| 10002 | Premium | l | 12.7 | 10 | 1 | 1 |
| 10003 | Diesel | l | 11.4 | 10 | 1 | 1 |
| 10004 | Unleaded | l | 9.85 | 10 | 1 | 1 |
| 20001 | Chewing gum "... | pcs | 4.55 | 1 | 1 | 1 |
| 20002 | Cake "Dan cake" | pcs | 10.5 | 1 | 1 | 1 |
| 20003 | Coca-cola botl. 1L | pcs | 6.35 | 1 | 1 | 1 |
| 20004 | Juice "Tomato" ... | pcs | 8.4 | 1 | 1 | 1 |
| 20005 | Chocolate "Nuts" | pcs | 5.3 | 1 | 1 | 1 |
| 20006 | Candies "Rafael... | pcs | 18.2 | 1 | 1 | 1 |
| 20007 | Brandy "Galitsin... | pcs | 65.4 | 1 | 1 | 1 |
| 20008 | Milk botl. 1L | pcs | 7.2 | 1 | 1 | 1 |
| 20009 | Bread "Morning" | pcs | 3.5 | 1 | 1 | 1 |
| 20010 | Car air cleaner "... | pcs | 12 | 1 | 1 | 1 |

Tanks - (4)

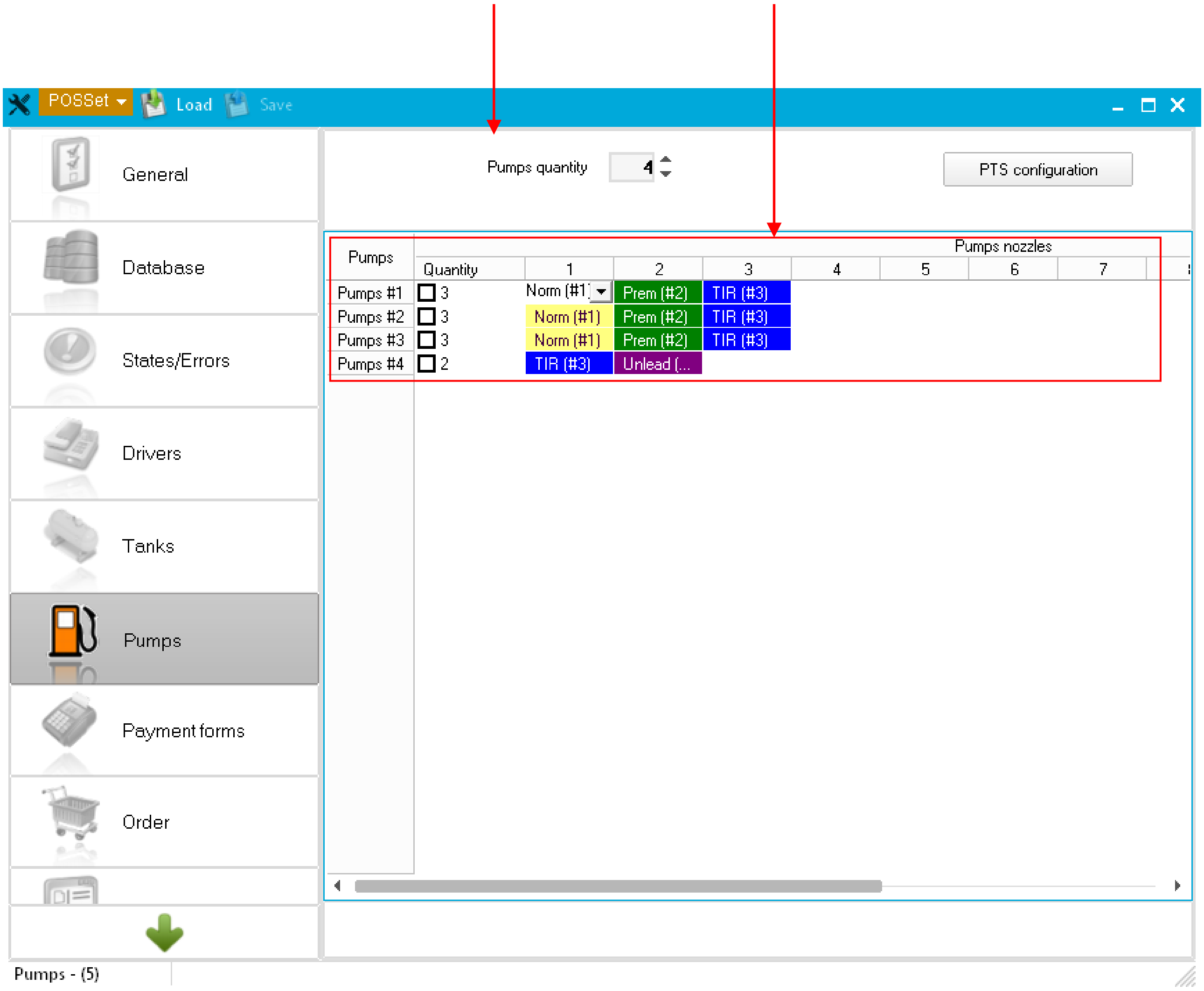
## 3. Configuration of tab 'Pumps'

Make sure that you have the pumps quantity set and pumps' nozzles are linked to fuel types:

Setting of pumps quantity

Linking of pumps' nozzles to fuel types

Run PTS configuration utility



Pumps quantity 4

| Pumps | Quantity | Pumps nozzles 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Pumps #1 | □ 3 | Norm (#1 | Prem (#2) | TIR (#3) | | | | |
| Pumps #2 | □ 3 | Norm (#1) | Prem (#2) | TIR (#3) | | | | |
| Pumps #3 | □ 3 | Norm (#1) | Prem (#2) | TIR (#3) | | | | |
| Pumps #4 | □ 2 | TIR (#3) | Unlead (... | | | | | |

***NOTE!***

Due to a reason of constant development of PTS controller firmware and adding to it communication protocols of new fuel dispensers and ATG systems PTS controller configuration tool is also constantly being developed. That is why in order to have the latest version of the configuration tool and the latest firmware of the PTS controller – request it from “TECHNOTRADE LTD” company. Location of PTS controller configuration tool (in case of installation of NaftaPOS system on drive C:\) is “C:\NaftaPOS\pts_conf\”.

## 4. Configuration of tab ‘View’

Make sure that you have the windows to be displayed on main form to be checked here:

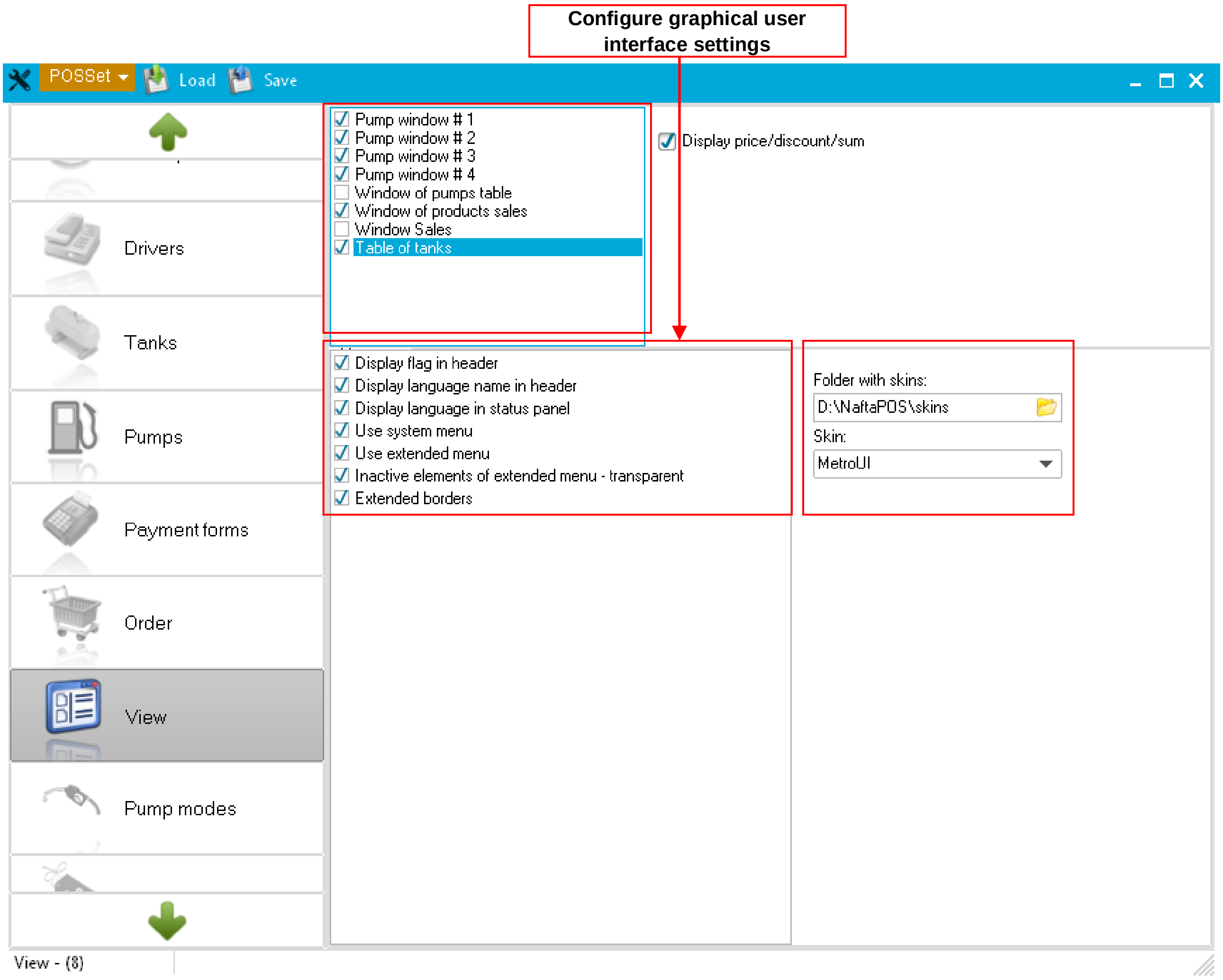
- Pump window #... (for each of configured pumps quantity set on tab “Pumps” above)
- Window of products sales (for a reason to be able to provide sales of retail products from main form)
- Table of tanks

Check all the checkboxes in the options for graphical user interface adjustments.

Also select a skin for graphical user interface.

**Configure windows to be displayed on the main form**

**Configure a skin for graphical user interface**



Configure graphical user interface settings

☑ Pump window # 1
☑ Pump window # 2
☑ Pump window # 3
☑ Pump window # 4
☐ Window of pumps table
☑ Window of products sales
☐ Window Sales
☑ Table of tanks

☑ Display price/discount/sum

☑ Display flag in header
☑ Display language name in header
☑ Display language in status panel
☑ Use system menu
☑ Use extended menu
☑ Inactive elements of extended menu - transparent
☑ Extended borders

Folder with skins: D:\NaftaPOS\skins

Skin: MetroUI

## 5. Configuration of tab ‘Drivers’

Tab ‘Drivers’ serves for configuration of drivers of connected equipment to NaftaPOS software, namely:

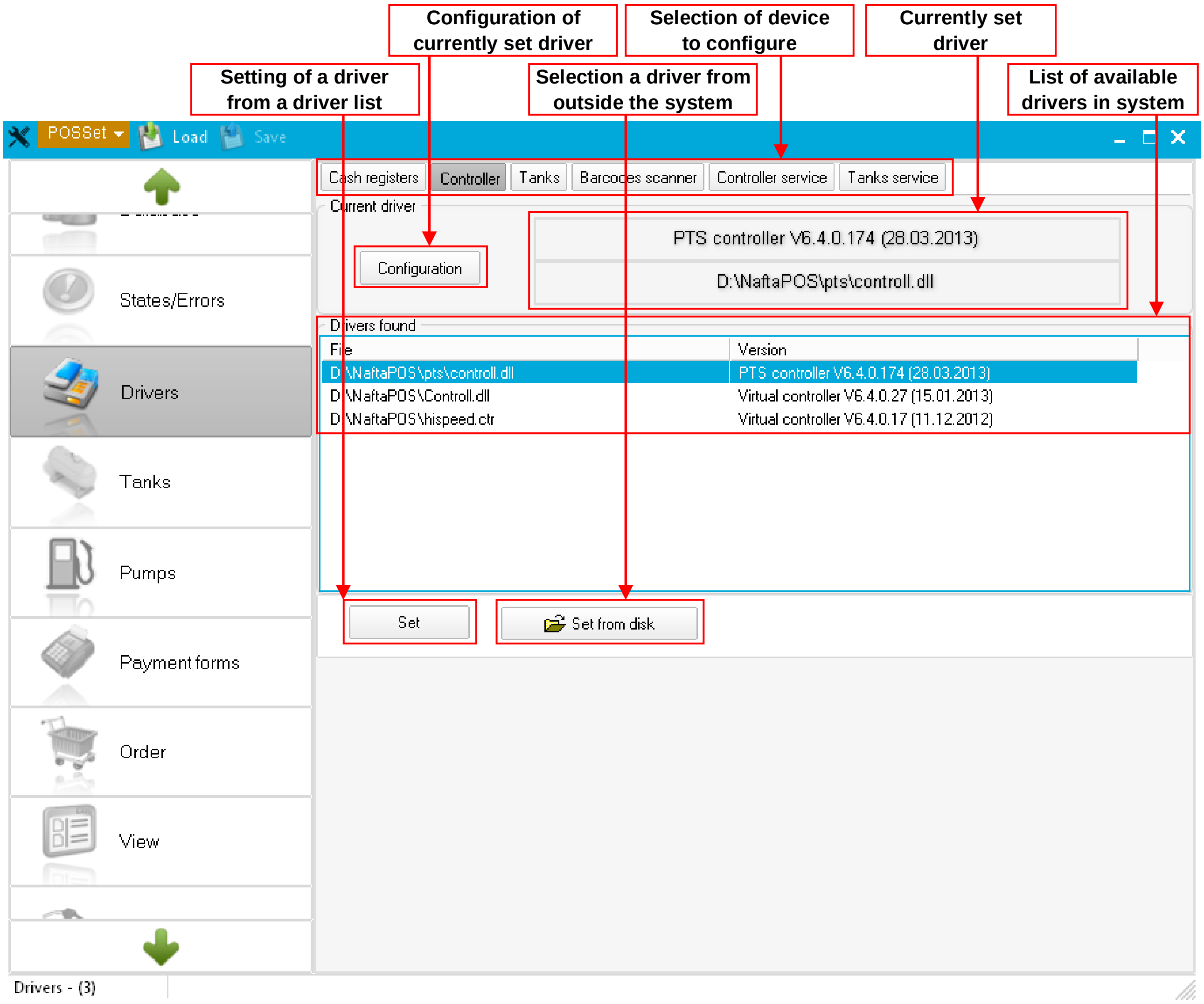
- Cash registers (receipt printers)
- Forecourt controllers
- ATG systems
- Barcode scanners
- Controller service
- Tanks service

By default all drivers are set as void drivers (simulators).

For test operation it is recommended to set the following configuration:

- As driver of cash register select “Void cash register driver”
- As driver of controller select “PTS controller”
- As driver of tanks select “PTS ATG driver”

For operation of NaftaPOS software with PTS controller it is necessary to click on button “Controller” from the top menu in this tab “Drivers” and there select from the list of drivers a driver for PTS controller and click on a button “Set “ to make it applied in the system.



PTS controller V6.4.0.174 (28.03.2013)

D:\NaftaPOS\pts\controll.dll

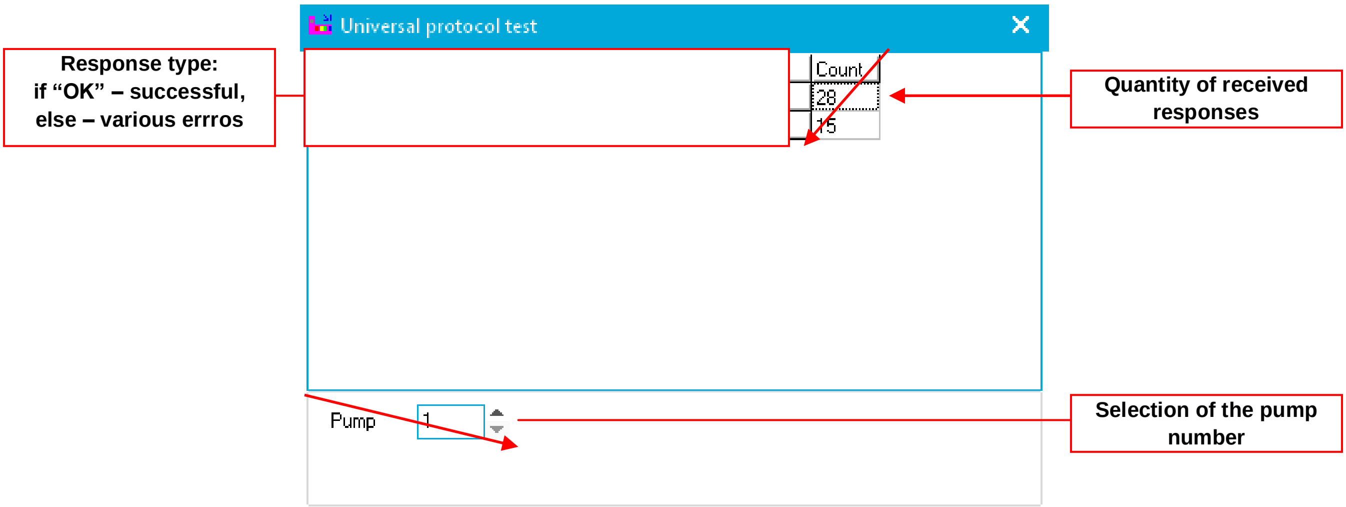
| File | Version |
|---|---|
| D:\NaftaPOS\pts\controll.dll | PTS controller V6.4.0.174 (28.03.2013) |
| D:\NaftaPOS\Controll.dll | Virtual controller V6.4.0.27 (15.01.2013) |
| D:\NaftaPOS\hispeed.ctr | Virtual controller V6.4.0.17 (11.12.2012) |

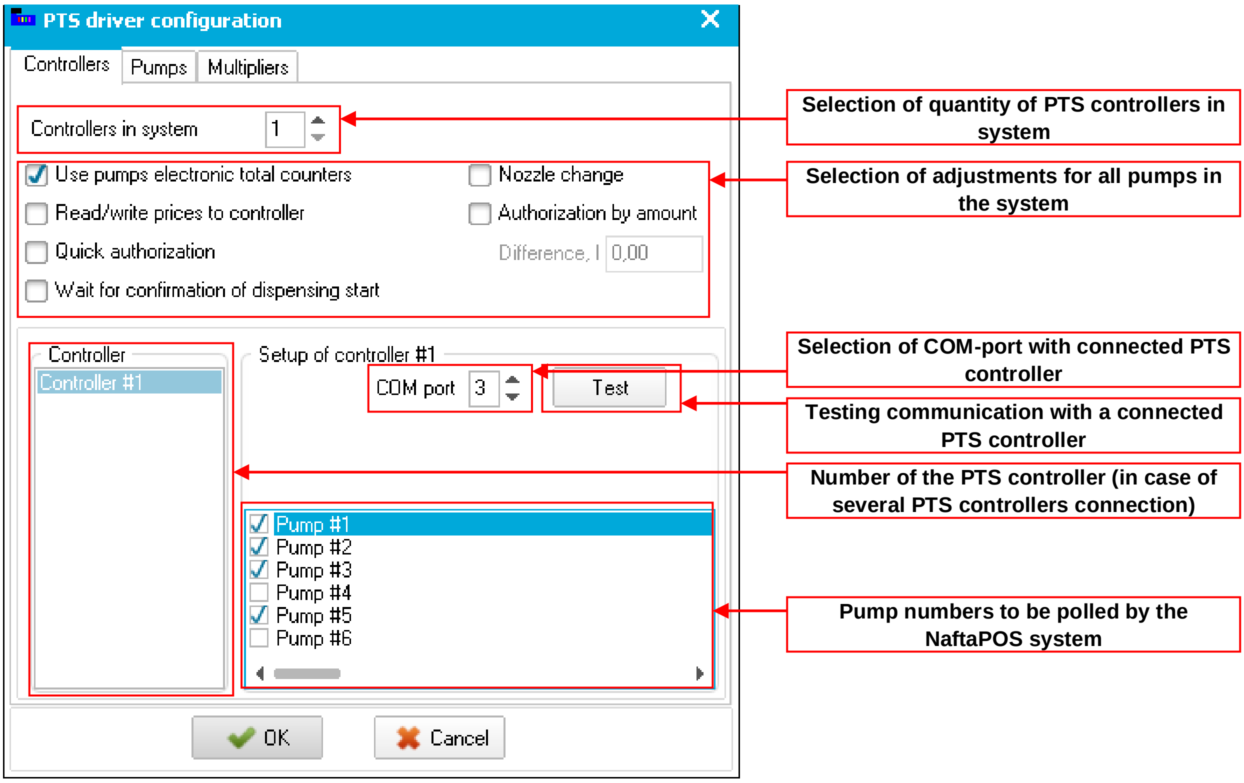
## Configuration of pump controller

Driver “PTS controller” should be selected as a driver for “Controller”.

Tab “Controller” specifies the pumps in NaftaPOS system, with which NaftaPOS works. To each of these pumps on tab “Pumps” can be linked any pump logical address, which is configured in PTS controller configuration.

Clicking of button “Test” opens a window for checking communication with the current PTS controller on specified COM-port: If response is OK – communication is successful, otherwise various errors will be displayed.


Universal protocol test
Response type: if “OK” – successful, else – various errros
Quantity of received responses
Selection of the pump number
Count
28
15
Pump
1



PTS driver configuration

Controllers | Pumps | Multipliers

Controllers in system: 1

☑ Use pumps electronic total counters
□ Read/write prices to controller
□ Quick authorization
□ Wait for confirmation of dispensing start
□ Nozzle change
□ Authorization by amount
Difference, l: 0,00

Controller: Controller #1

Setup of controller #1
COM port: 3
Test

☑ Pump #1
☑ Pump #2
☑ Pump #3
□ Pump #4
☑ Pump #5
□ Pump #6

OK | Cancel

Selection of quantity of PTS controllers in system

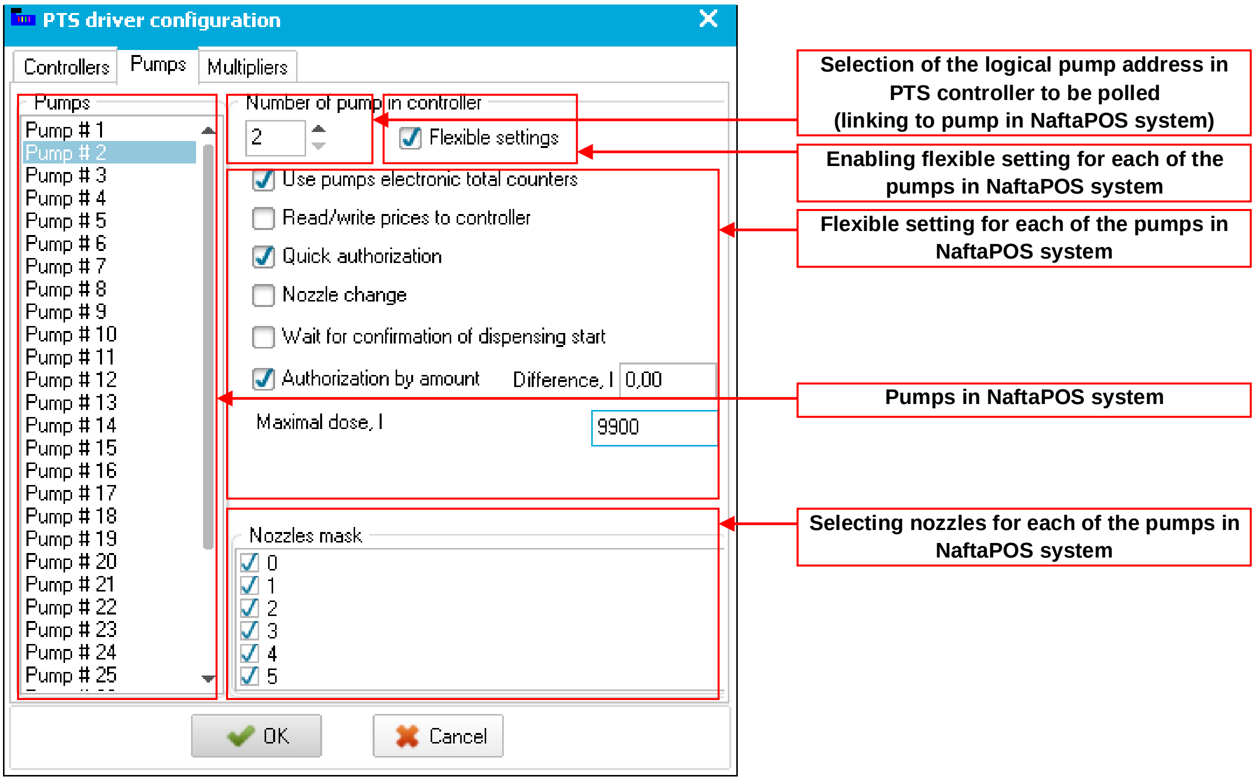
Selection of adjustments for all pumps in the system

Selection of COM-port with connected PTS controller

Testing communication with a connected PTS controller

Number of the PTS controller (in case of several PTS controllers connection)

Pump numbers to be polled by the NaftaPOS system



PTS driver configuration

Controllers | Pumps | Multipliers

Pumps: Pump # 1, Pump # 2, Pump # 3, Pump # 4, Pump # 5, Pump # 6, Pump # 7, Pump # 8, Pump # 9, Pump # 10, Pump # 11, Pump # 12, Pump # 13, Pump # 14, Pump # 15, Pump # 16, Pump # 17, Pump # 18, Pump # 19, Pump # 20, Pump # 21, Pump # 22, Pump # 23, Pump # 24, Pump # 25

Number of pump in controller: 2

☑ Flexible settings

☑ Use pumps electronic total counters
□ Read/write prices to controller
☑ Quick authorization
□ Nozzle change
□ Wait for confirmation of dispensing start
☑ Authorization by amount Difference, l: 0,00
Maximal dose, l: 9900

Nozzles mask
☑ 0
☑ 1
☑ 2
☑ 3
☑ 4
☑ 5

OK | Cancel

Selection of the logical pump address in PTS controller to be polled (linking to pump in NaftaPOS system)

Enabling flexible setting for each of the pumps in NaftaPOS system

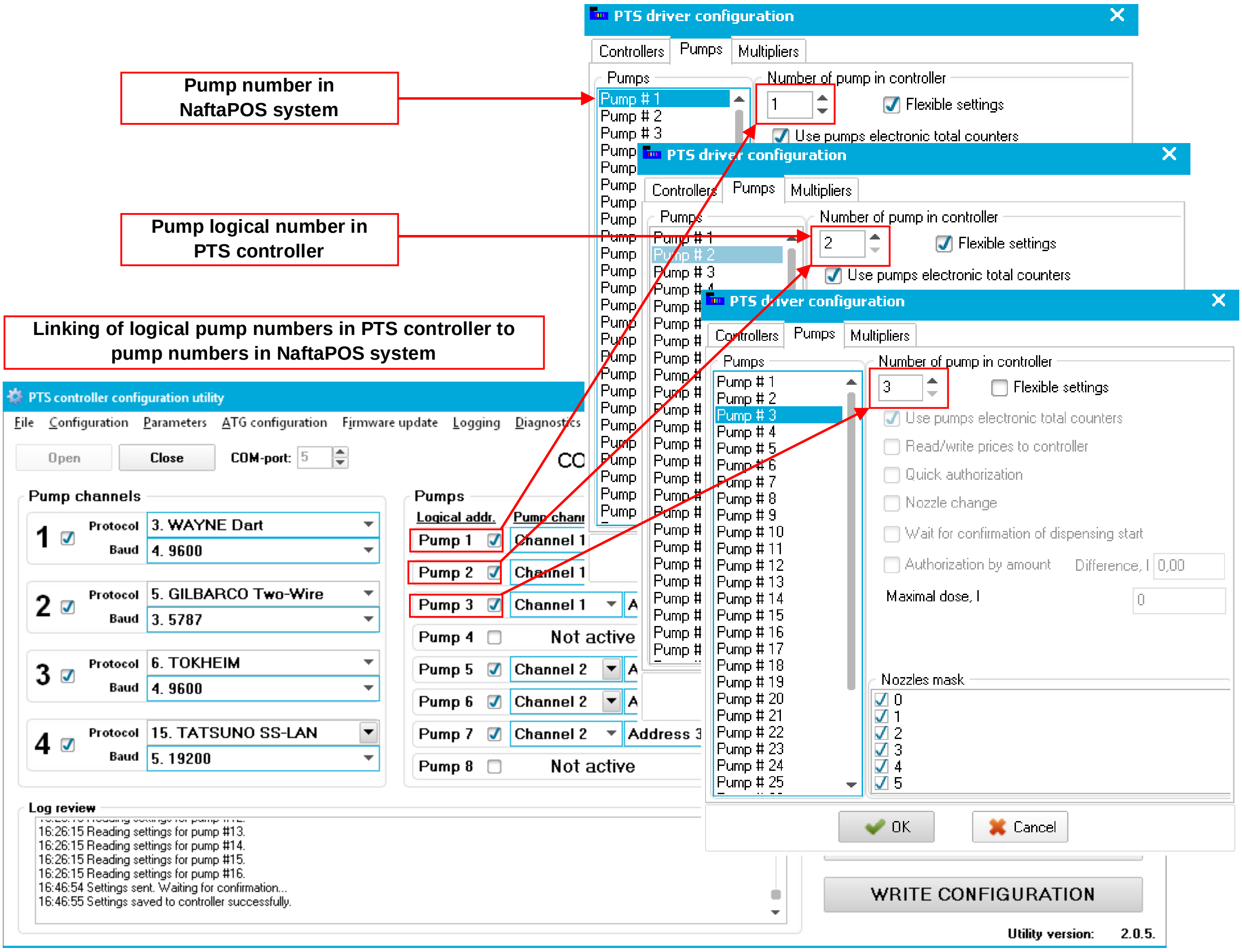
Flexible setting for each of the pumps in NaftaPOS system

Pumps in NaftaPOS system

Selecting nozzles for each of the pumps in NaftaPOS system

In configuration of “PTS controller” driver only logical addresses of dispensers are set, the physical address of the connected fueling place is specified in PTS controller configuration utility.

Linking of pump windows numbers to logical addresses of fuel dispensers set in PTS controller configuration tool is made on tab “Pumps” of PTS controller driver:



Pump number in NaftaPOS system

Pump logical number in PTS controller

Linking of logical pump numbers in PTS controller to pump numbers in NaftaPOS system

PTS driver configuration – Pumps tab: Pump # 1 selected, Number of pump in controller: 1, ☑ Flexible settings, ☑ Use pumps electronic total counters

PTS driver configuration – Pumps tab: Pump # 2 selected, Number of pump in controller: 2, ☑ Flexible settings, ☑ Use pumps electronic total counters

PTS driver configuration – Pumps tab: Pump # 3 selected

Number of pump in controller: 3

□ Flexible settings
☑ Use pumps electronic total counters
□ Read/write prices to controller
□ Quick authorization
□ Nozzle change
□ Wait for confirmation of dispensing start
□ Authorization by amount Difference, l 0,00

Maximal dose, l 0

Nozzles mask: ☑ 0 ☑ 1 ☑ 2 ☑ 3 ☑ 4 ☑ 5

PTS controller configuration utility – COM-port: 5

Pump channels

| Channel | | Protocol | Baud |
|---|---|---|---|
| 1 | ☑ | 3. WAYNE Dart | 4. 9600 |
| 2 | ☑ | 5. GILBARCO Two-Wire | 3. 5787 |
| 3 | ☑ | 6. TOKHEIM | 4. 9600 |
| 4 | ☑ | 15. TATSUNO SS-LAN | 5. 19200 |

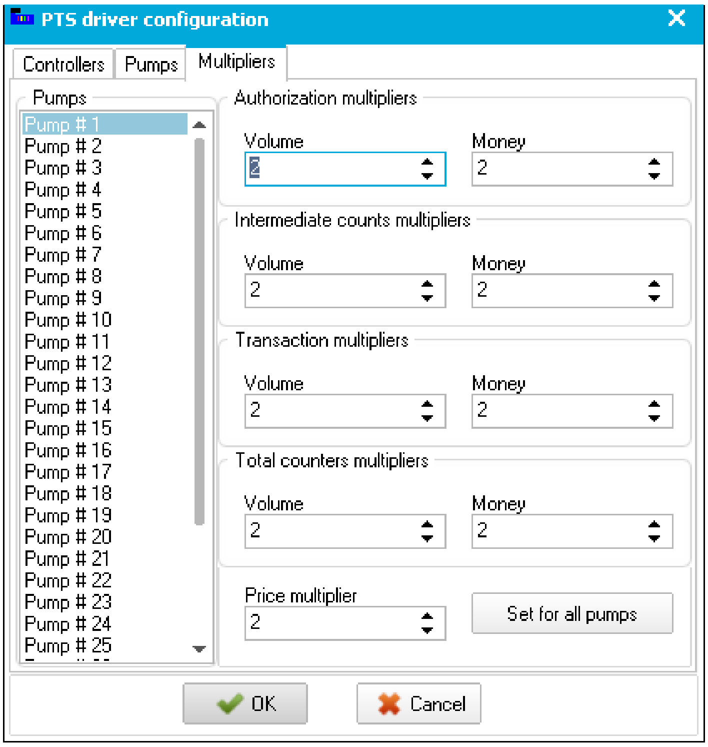
Pumps

| Logical addr. | | Pump chann |
|---|---|---|
| Pump 1 | ☑ | Channel 1 |
| Pump 2 | ☑ | Channel 1 |
| Pump 3 | ☑ | Channel 1 |
| Pump 4 | □ | Not active |
| Pump 5 | ☑ | Channel 2 |
| Pump 6 | ☑ | Channel 2 |
| Pump 7 | ☑ | Channel 2 |
| Pump 8 | □ | Not active |

Log review

```
16:26:15 Reading settings for pump #13.
16:26:15 Reading settings for pump #14.
16:26:15 Reading settings for pump #15.
16:26:15 Reading settings for pump #16.
16:46:54 Settings sent. Waiting for confirmation...
16:46:55 Settings saved to controller successfully.
```

Multipliers can be set separately for each of the pumps. Clicking on button “*Set for all pumps*” leads to setting of the current multipliers on currently selected pump for all other pumps.



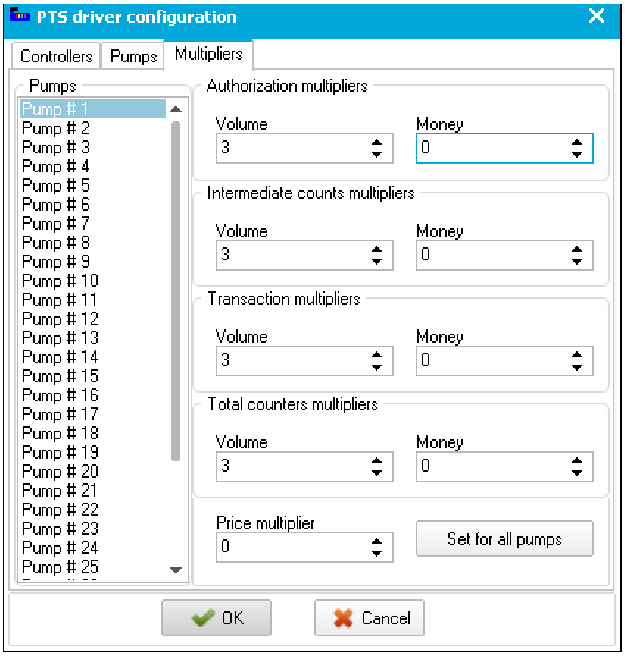
PTS driver configuration – Multipliers tab: Pump # 1 selected

| | Volume | Money |
|---|---|---|
| Authorization multipliers | 2 | 2 |
| Intermediate counts multipliers | 2 | 2 |
| Transaction multipliers | 2 | 2 |
| Total counters multipliers | 2 | 2 |

Price multiplier: 2

Shown on image above multipliers set that there are 2 digits on the pump in fields of monay amount, volume and price and also pump gives out total counters (volume and money) with 2 digits after decimal separator.

Other examples of multipliers:



Authorization multipliers
Volume: 3
Money: 0

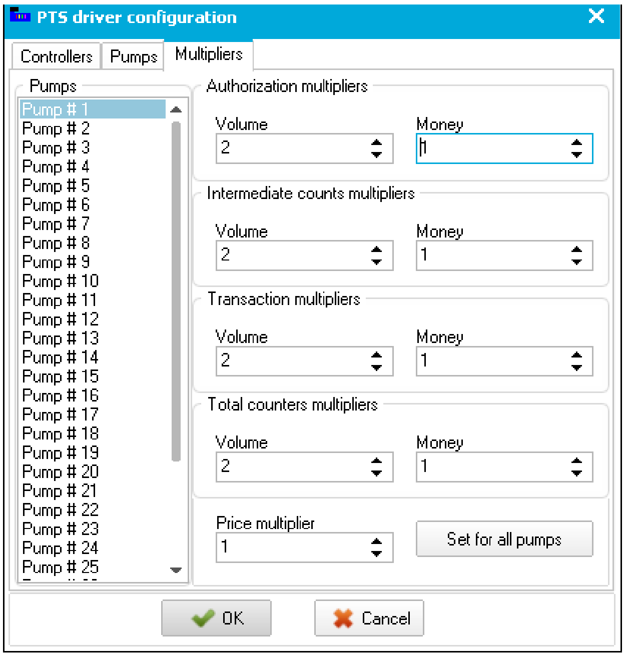
Intermediate counts multipliers
Volume: 3
Money: 0

Transaction multipliers
Volume: 3
Money: 0

Total counters multipliers
Volume: 3
Money: 0

Price multiplier: 0

*Setting multipliers for pump with 0 digit in money amount and 3 digits in volume*



Authorization multipliers
Volume: 2
Money: 1

Intermediate counts multipliers
Volume: 2
Money: 1

Transaction multipliers
Volume: 2
Money: 1

Total counters multipliers
Volume: 2
Money: 1

Price multiplier: 1

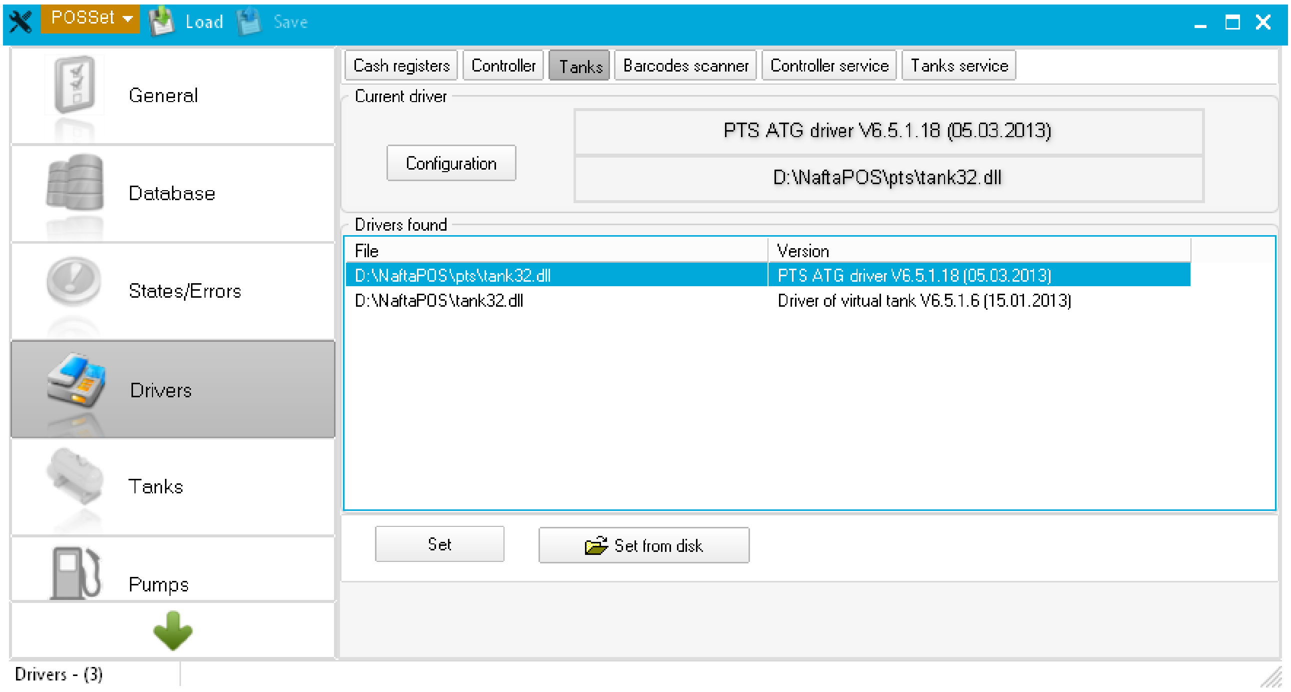
*Setting multipliers for pump with 1 digit in money amount and 2 digits in volume*

Except setting of multipliers for each of the pumps it is necessary to set multipliers in NaftaPOS system in order to set how data should be accounted in system, stored in database and displayed in reports. Setting of multipliers for NaftaPOS system is made on tab "*Order*" in configuration utility of NaftaPOS.

Driver "Virtual controller" is used for simulation of fuel dispensers. At using of this driver NaftaPOS can simulate presence of actual fuel dispensers, but at this no real dispensers are connected to the system.
Driver "Virtual controller (HIGHSPEED)" is used for simulation of fuel dispensers like driver "Virtual controller", but unlike it dispensing is made very quickly and there is no need to wait until dispensing is finished no matter what is the order value.

## Configuration of ATG controller

For operation of NaftaPOS software with PTS controller it is necessary to click on button “Tanks” from the top menu in tab “Drivers” and there select from the list of drivers a driver for PTS ATG and click on a button “Set “ to make it applied in the system.



Current driver

PTS ATG driver V6.5.1.18 (05.03.2013)

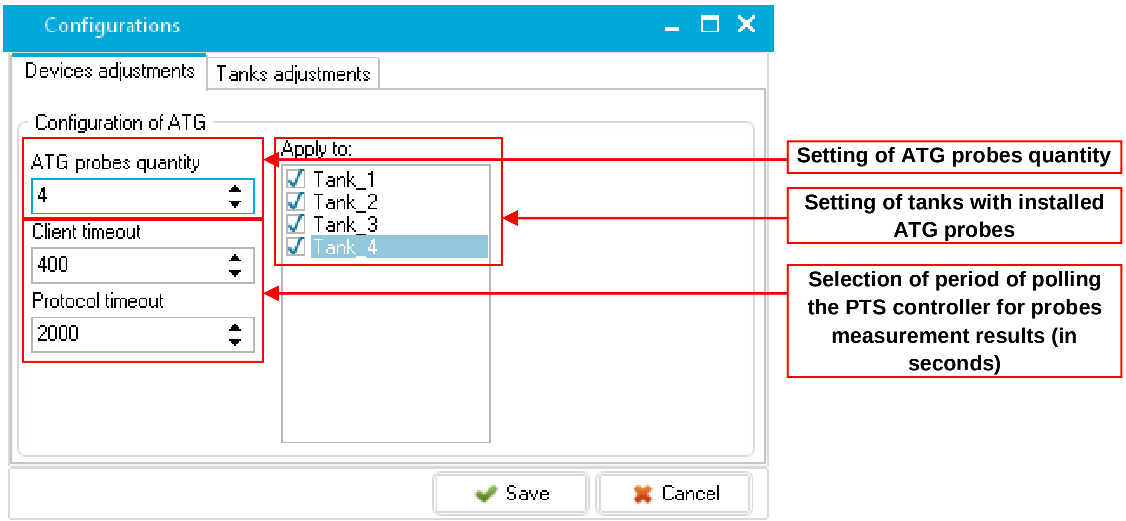
D:\NaftaPOS\pts\tank32.dll

Drivers found

| File | Version |
|---|---|
| D:\NaftaPOS\pts\tank32.dll | PTS ATG driver V6.5.1.18 (05.03.2013) |
| D:\NaftaPOS\tank32.dll | Driver of virtual tank V6.5.1.6 (15.01.2013) |

Driver “PTS ATG driver” is used for communication with various ATG systems using the PTS controller.

In configuration of “PTS ATG driver” driver only logical addresses of ATG system probes (sensors) are set, the original physical addresses of probes are specified in PTS controller configuration utility.



Configuration of ATG

ATG probes quantity: 4

Client timeout: 400

Protocol timeout: 2000

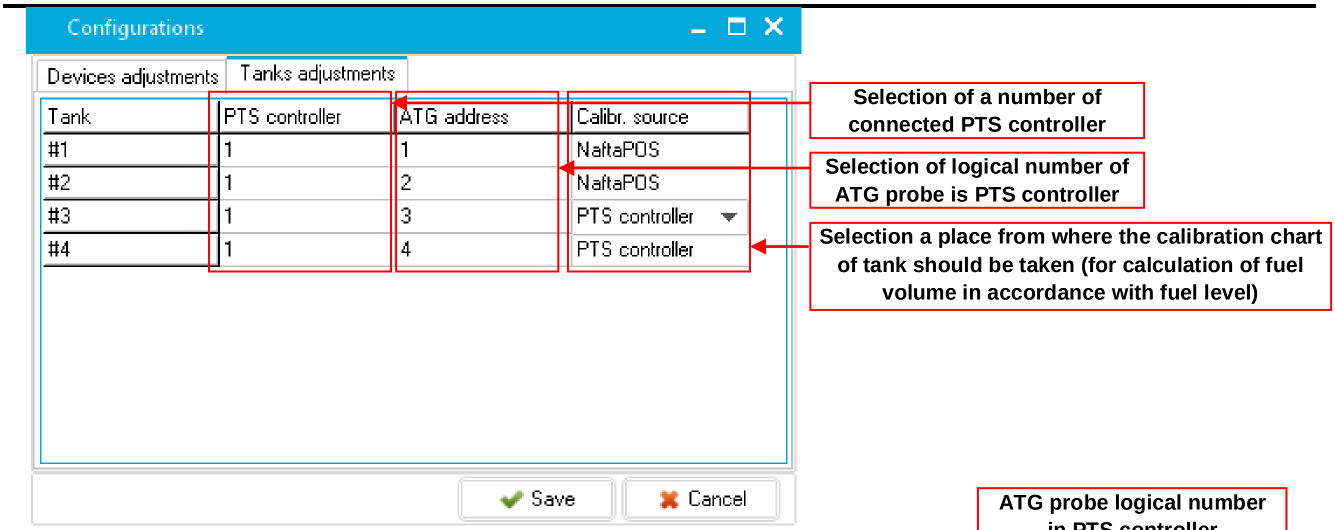
Apply to:

☑ Tank_1
☑ Tank_2
☑ Tank_3
☑ Tank_4

Setting of ATG probes quantity

Setting of tanks with installed ATG probes

Selection of period of polling the PTS controller for probes measurement results (in seconds)



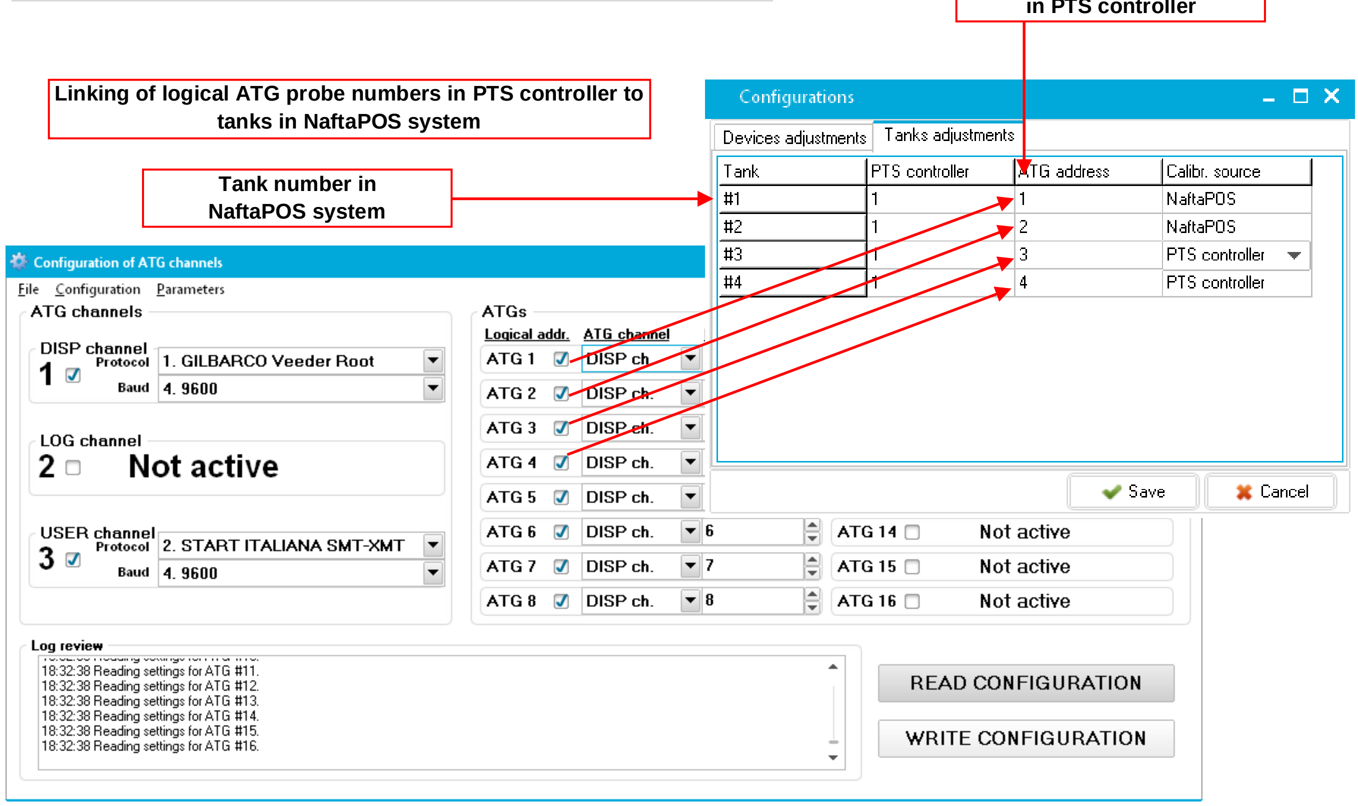
| Tank | PTS controller | ATG address | Calibr. source |
|---|---|---|---|
| #1 | 1 | 1 | NaftaPOS |
| #2 | 1 | 2 | NaftaPOS |
| #3 | 1 | 3 | PTS controller |
| #4 | 1 | 4 | PTS controller |

Selection of a number of connected PTS controller

Selection of logical number of ATG probe is PTS controller

Selection a place from where the calibration chart of tank should be taken (for calculation of fuel volume in accordance with fuel level)

ATG probe logical number in PTS controller



Linking of logical ATG probe numbers in PTS controller to tanks in NaftaPOS system

Tank number in NaftaPOS system

| Tank | PTS controller | ATG address | Calibr. source |
|---|---|---|---|
| #1 | 1 | 1 | NaftaPOS |
| #2 | 1 | 2 | NaftaPOS |
| #3 | 1 | 3 | PTS controller |
| #4 | 1 | 4 | PTS controller |

ATG channels

DISP channel 1 ☑ Protocol: 1. GILBARCO Veeder Root; Baud: 4. 9600

LOG channel 2 □ Not active

USER channel 3 ☑ Protocol: 2. START ITALIANA SMT-XMT; Baud: 4. 9600

ATGs

| Logical addr. | | ATG channel | | | |
|---|---|---|---|---|---|
| ATG 1 | ☑ | DISP ch. | | | |
| ATG 2 | ☑ | DISP ch. | | | |
| ATG 3 | ☑ | DISP ch. | | | |
| ATG 4 | ☑ | DISP ch. | | | |
| ATG 5 | ☑ | DISP ch. | | | |
| ATG 6 | ☑ | DISP ch. | 6 | ATG 14 □ | Not active |
| ATG 7 | ☑ | DISP ch. | 7 | ATG 15 □ | Not active |
| ATG 8 | ☑ | DISP ch. | 8 | ATG 16 □ | Not active |

Log review

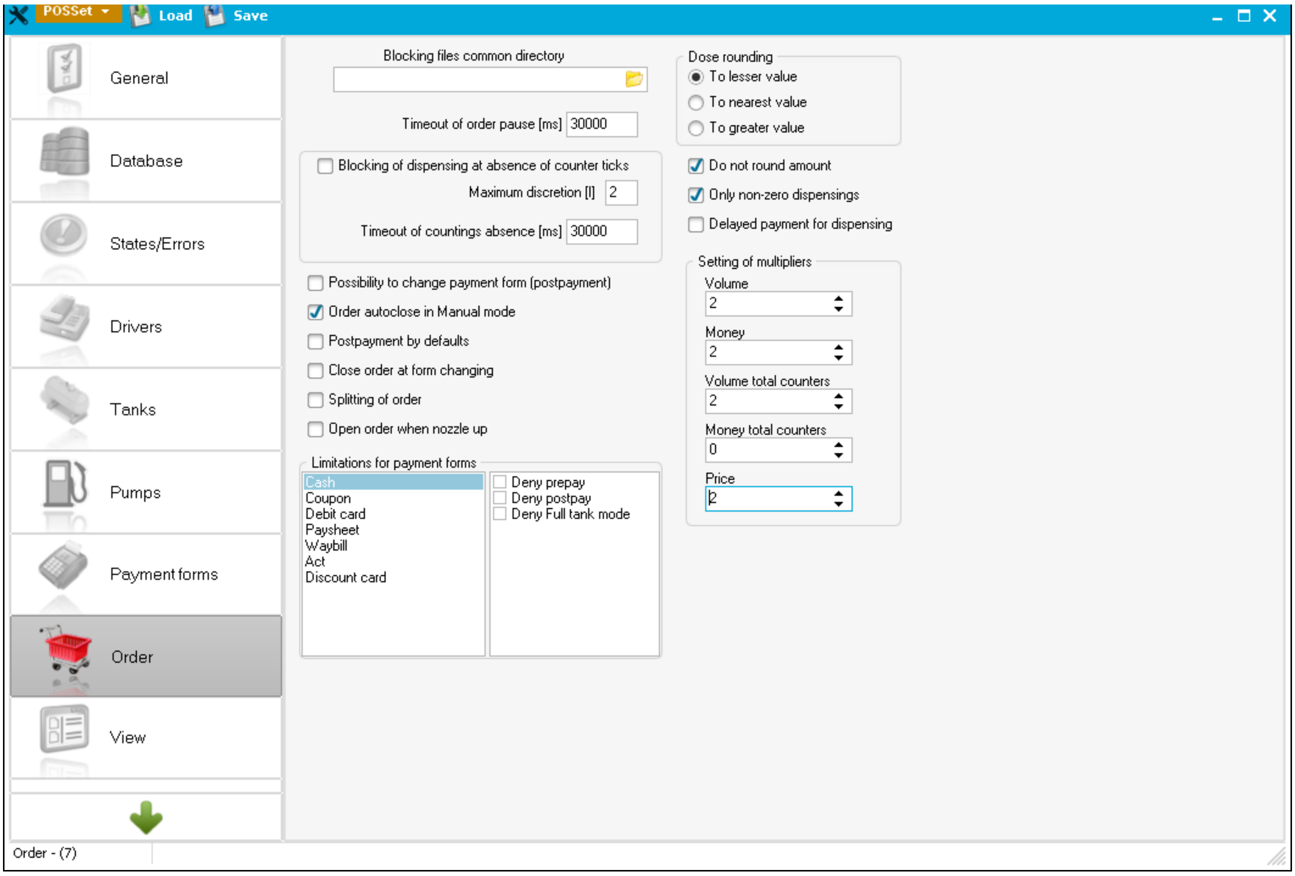
```
18:32:38 Reading settings for ATG #11.
18:32:38 Reading settings for ATG #12.
18:32:38 Reading settings for ATG #13.
18:32:38 Reading settings for ATG #14.
18:32:38 Reading settings for ATG #15.
18:32:38 Reading settings for ATG #16.
```

Driver “Driver of virtual tank” is used for simulation of ATG system presence in system. This driver should be set, when there is no ATG system connected. In this case a checkbox “Level meter presence in system” on tab “General” should be unchecked.

## 6. Configuration of tab ‘Order’

Tab ‘Drivers’ serves for configuration of several things of NaftaPOS system connected with order placing. Make sure that the following things are checked:

- Checkbox “order autoclose in Manual mode”
- Only-non-zero dispensings



Blocking files common directory

Timeout of order pause [ms] 30000

☐ Blocking of dispensing at absence of counter ticks

Maximum discretion [l] 2

Timeout of countings absence [ms] 30000

☐ Possibility to change payment form (postpayment)

☑ Order autoclose in Manual mode

☐ Postpayment by defaults

☐ Close order at form changing

☐ Splitting of order

☐ Open order when nozzle up

Limitations for payment forms

Cash
Coupon
Debit card
Paysheet
Waybill
Act
Discount card

☐ Deny prepay
☐ Deny postpay
☐ Deny Full tank mode

Dose rounding

● To lesser value
○ To nearest value
○ To greater value

☑ Do not round amount

☑ Only non-zero dispensings

☐ Delayed payment for dispensing

Setting of multipliers

Volume 2

Money 2

Volume total counters 2

Money total counters 0

Price 2

Area “Setting of multipliers” sets how many digits after decimal separator in values of money amount, volume, price and total counter values received from pumps should be accounted in NaftaPOS system for calculations, storing in database and displaying in reports.

All the described steps are basic configuration of NaftaPOS software to operate with PTS controller. NaftaPOS has more options to configure, more details about them please read on NaftaPOS web-page: http://www.technotrade.ua/petrol_station_software.html. For testing purposes of NaftaPOS software operation with PTS controller these adjustments are enough.

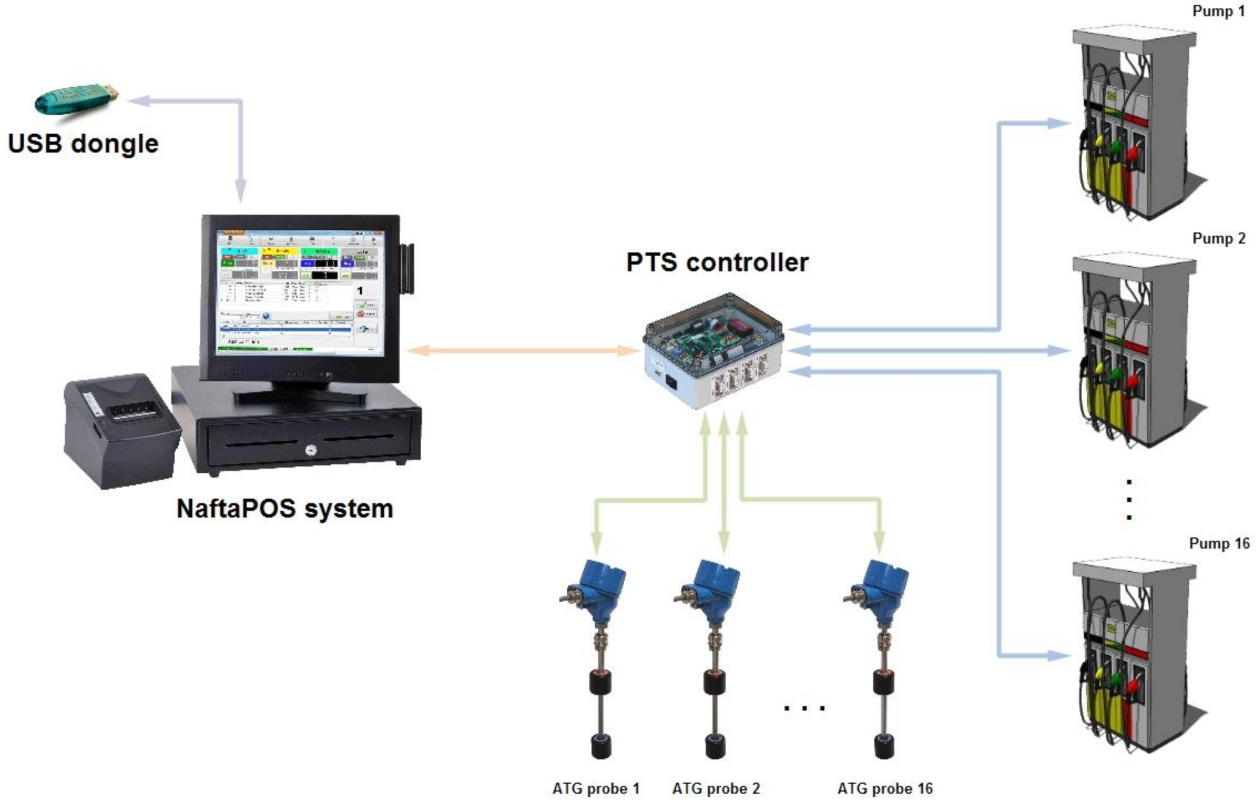
## *Running NaftaPOS software*

NaftaPOS software is to be started by running NaftaPOS.exe. After checking of connected equipment you will see a desktop of NaftaPOS (graphical user interface is adjustable in right-mouse button menu).

In “NaftaPOS” shift mode of operation is applied. Cashier at beginning of operation should open a new shift (in main menu tab ‘Shift’ → item ‘New shift’). Shift control is required in order to make the system operate under responsibility of the logged cashier (under his personal login and password) and thus make him bear responsibility for all actions happening at petrol station. Shift control can be switched off in configuration of NaftaPOS on tab ‘General’.

Detailed description on running the NaftaPOS system is given in section “Step by step configuration” below.

To understand better how to work in NaftaPOS and lead control over fuel dispensers, sales of products and management over a petrol station – please read documentation and see online videos on the NaftaPOS software web-page: http://www.technotrade.ua/petrol_station_software.html.

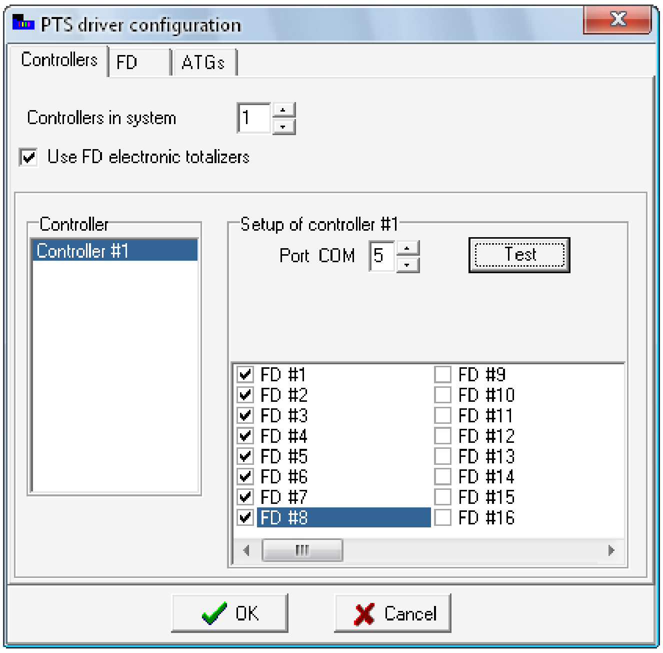


Localization of NaftaPOS software to any language is quite easy. All text strings and their correspondent translation is located in XML file lang.xml in root folder of NaftaPOS software.

## *PumpDemo utility*

PumpDemo utility is a contracted version of NaftaPOS software used for debugging purposes. It does not require installation.

Configuration of the system is done by running a setup.bat file. Configuration is similar to configuration of NaftaPOS software for operation with PTS controller (see section “Configuration of NaftaPOS software to work with PTS controller” above):



PTS driver configuration

Controllers in system 1

☑ Use FD electronic totalizers

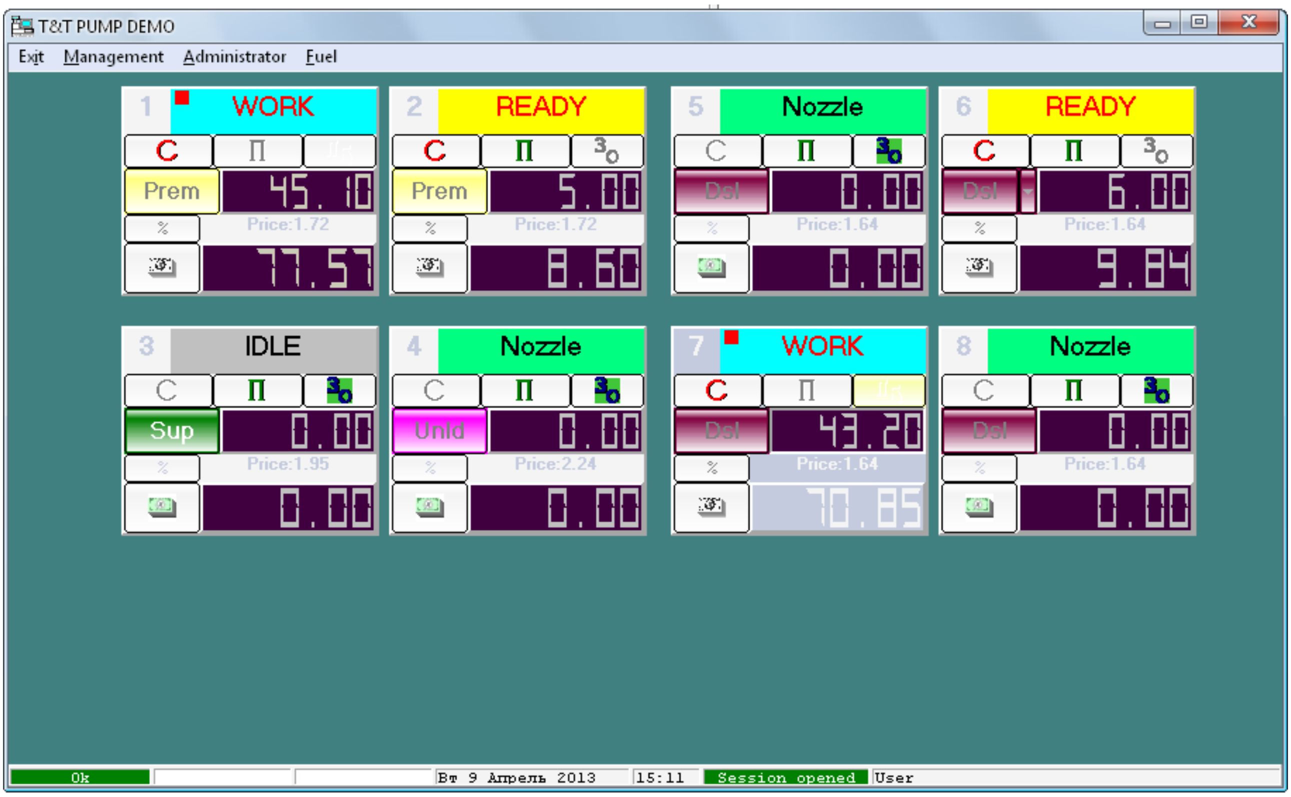
Controller: Controller #1

Setup of controller #1

Port COM 5

☑ FD #1 □ FD #9
☑ FD #2 □ FD #10
☑ FD #3 □ FD #11
☑ FD #4 □ FD #12
☑ FD #5 □ FD #13
☑ FD #6 □ FD #14
☑ FD #7 □ FD #15
☑ FD #8 □ FD #16

This allows to set configuration for communicating with PTS controller.
Graphical user interface is adjusted in a file AZSDemo.dat, where all information is stored in a view of text.



T&T PUMP DEMO

1 WORK
C П
Prem 45.10
%
Price:1.72
77.57

2 READY
C П З₀
Prem 5.00
%
Price:1.72
8.60

5 Nozzle
C П З₀
Dsl 0.00
%
Price:1.64
0.00

6 READY
C П З₀
Dsl 6.00
%
Price:1.64
9.84

3 IDLE
C П З₀
Sup 0.00
%
Price:1.95
0.00

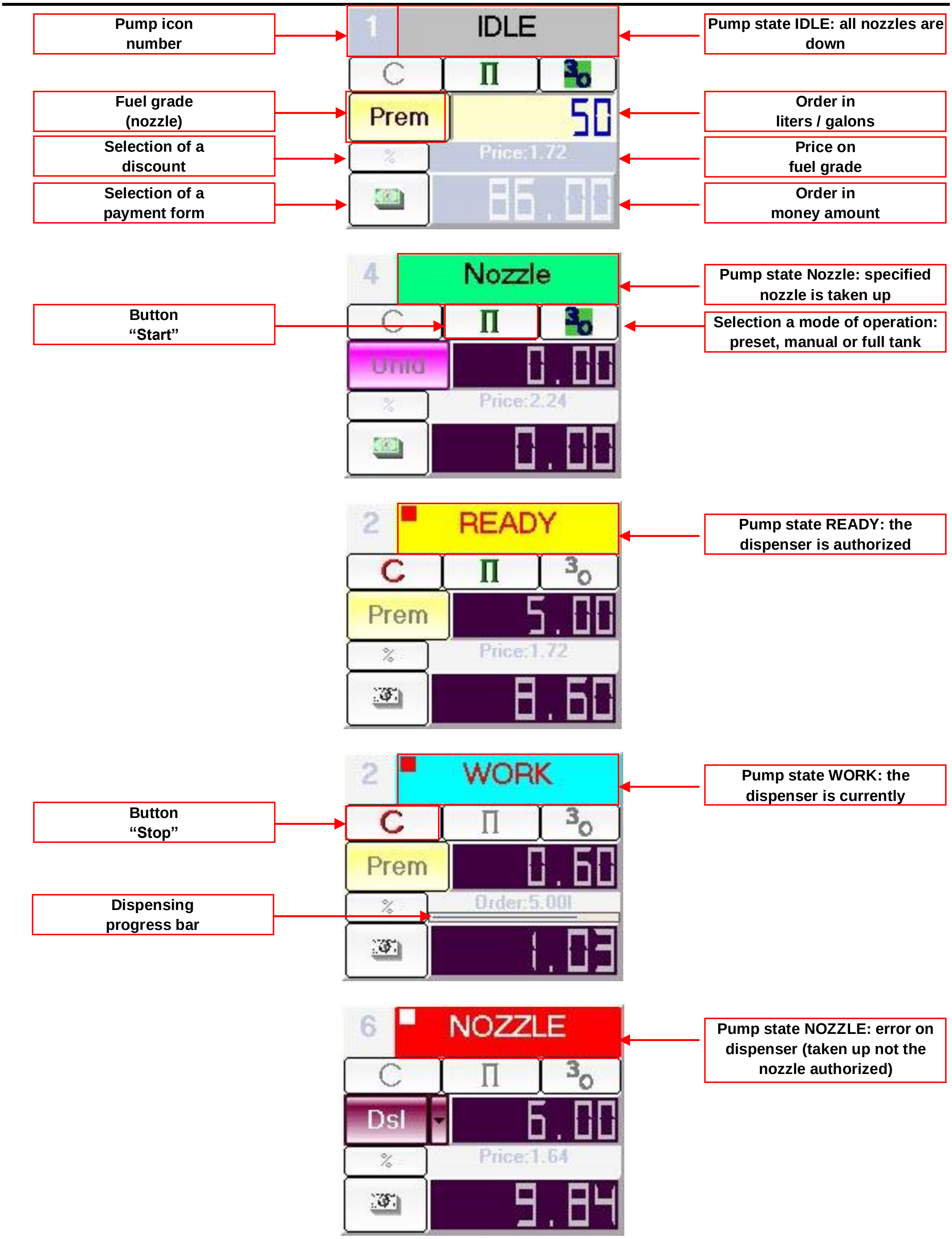
4 Nozzle
C П З₀
Unld 0.00
%
Price:2.24
0.00

7 WORK
C П
Dsl 43.20
%
Price:1.64
70.85

8 Nozzle
C П З₀
Dsl 0.00
%
Price:1.64
0.00

Ok | Вт 9 Апрель 2013 | 15:11 | Session opened | User

Operation of the PumpDemo utility is made same as operation of NaftaPOS software.



Pump icon number

Pump state IDLE: all nozzles are down

Fuel grade (nozzle)

Order in liters / galons

Selection of a discount

Price on fuel grade

Selection of a payment form

Order in money amount

Pump state Nozzle: specified nozzle is taken up

Button "Start"

Selection a mode of operation: preset, manual or full tank

Pump state READY: the dispenser is authorized

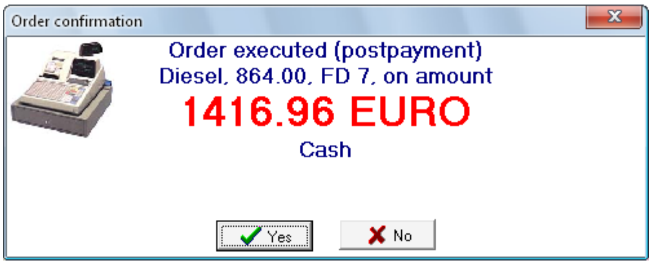
Button "Stop"

Pump state WORK: the dispenser is currently

Dispensing progress bar

Pump state NOZZLE: error on dispenser (taken up not the nozzle authorized)

As a confirmation of the fuel dispensing ordered in preset mode or performed in manual or full tank modes a correspondent window is shown:


Order confirmation

Order executed (postpayment)
Diesel, 864.00, FD 7, on amount
1416.96 EURO
Cash

# *Open source PTS controller .NET application*

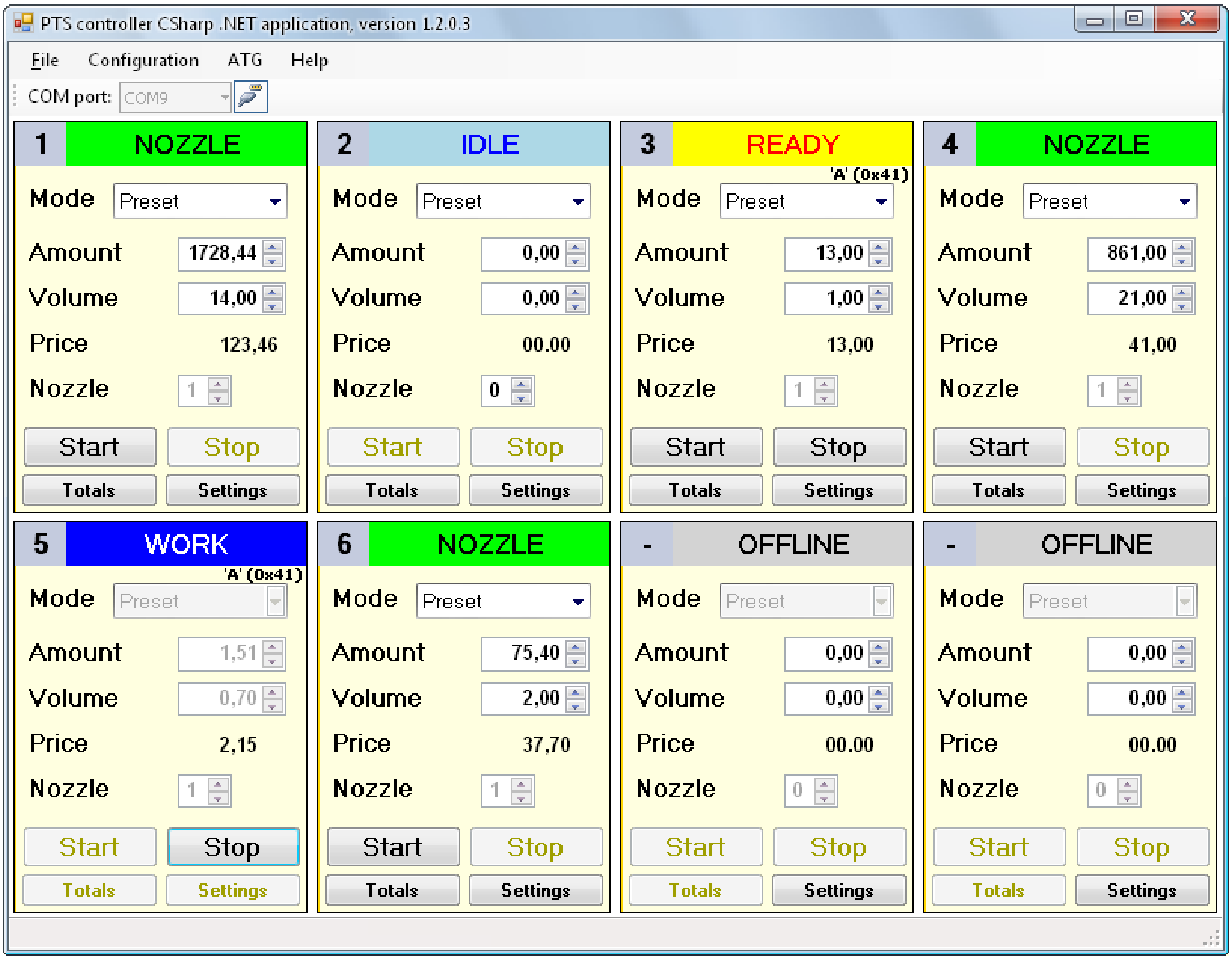
## *Purpose*

In order to provide control over PTS controller and simplify work of developers, writing on Microsoft .NET Framework in implementation of UniPump communication protocol of PTS controller and give an open source example of application for control over fuel dispensers and ATG systems at petrol station a .NET library (with open source in C# and VB.NET languages and documentation on methods and properties) and PTS controller .NET application (with open source in C# and VB.NET languages) are provides.

Given application includes 2 projects:

- PtsLib - project of PtsLib.dll library for operation with a PTS controller. Includes classes, methods and properties for communication with PTS controller using requests and responses described in UniPump communication protocol for communication with PTS controller for provision of control over dispensers and ATG systems and configuration of PTS controller.
- Fully-functional open source application for provision of control over up to 8 pumps and 4 ATG systems by the PTS controller, allows to perform all operations with the PTS controller (uses classes and methods of PtsLib.dll library).

PTS controller .NET application is provided on both C# and VB.NET languages and requires .NET Framework 2.0 for operation.



PTS controller CSharp .NET application, version 1.2.0.3

COM port: COM9

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| | NOZZLE | IDLE | READY | NOZZLE |
| | | | 'A' (0x41) | |
| Mode | Preset | Preset | Preset | Preset |
| Amount | 1728,44 | 0,00 | 13,00 | 861,00 |
| Volume | 14,00 | 0,00 | 1,00 | 21,00 |
| Price | 123,46 | 00.00 | 13,00 | 41,00 |
| Nozzle | 1 | 0 | 1 | 1 |

| | 5 | 6 | - | - |
|---|---|---|---|---|
| | WORK | NOZZLE | OFFLINE | OFFLINE |
| | 'A' (0x41) | | | |
| Mode | Preset | Preset | Preset | Preset |
| Amount | 1,51 | 75,40 | 0,00 | 0,00 |
| Volume | 0,70 | 2,00 | 0,00 | 0,00 |
| Price | 2,15 | 37,70 | 00.00 | 00.00 |
| Nozzle | 1 | 1 | 0 | 0 |

## *Start*

To launch PTS controller .NET application run PtsApplication.exe, select from a dropdown list a COM port name, to which PTS controller is connected, and press a button to open a selected COM port.

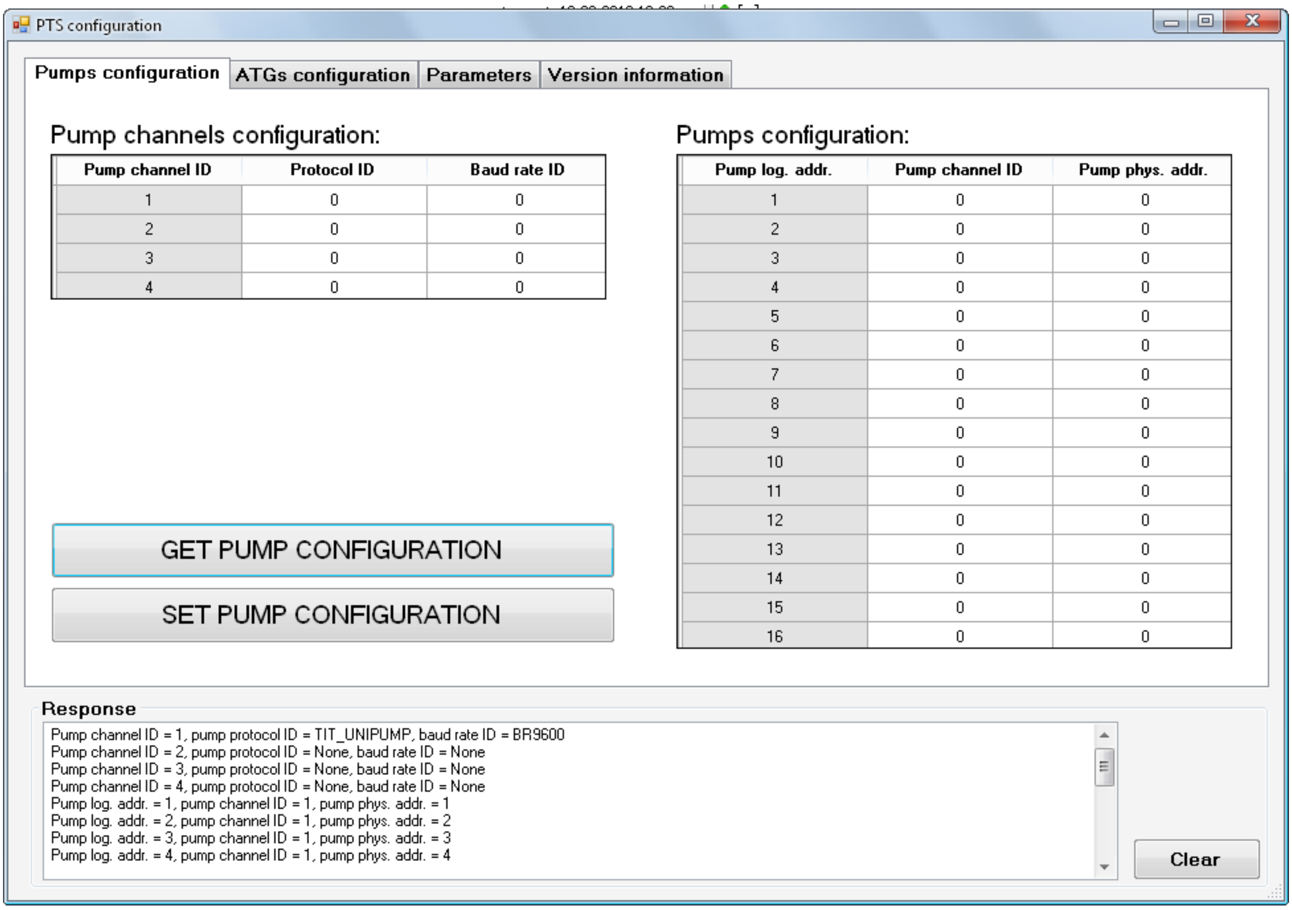
***Note***: connection with PTS controller is made through a 3-wire realization of RS-232 interface, an asynchronous half-duplex channel, baud rate 57600 baud without a parity check. This configuration is hardly coded inside a program.

## *Configuration of PTS controller*

To open configuration form go on the main form to menu item "Configuration" → "Fuel point configuration settings".

Configuration allows to set pump channels, ATG channels, parameters in PTS controller. It is quite convenient to use and open source code helps to understand better how the PTS controller works.

Pumps configuration tab allows to configure pumps channels of PTS controller (like *PTS_conf.exe* utility):



Pump channels configuration:

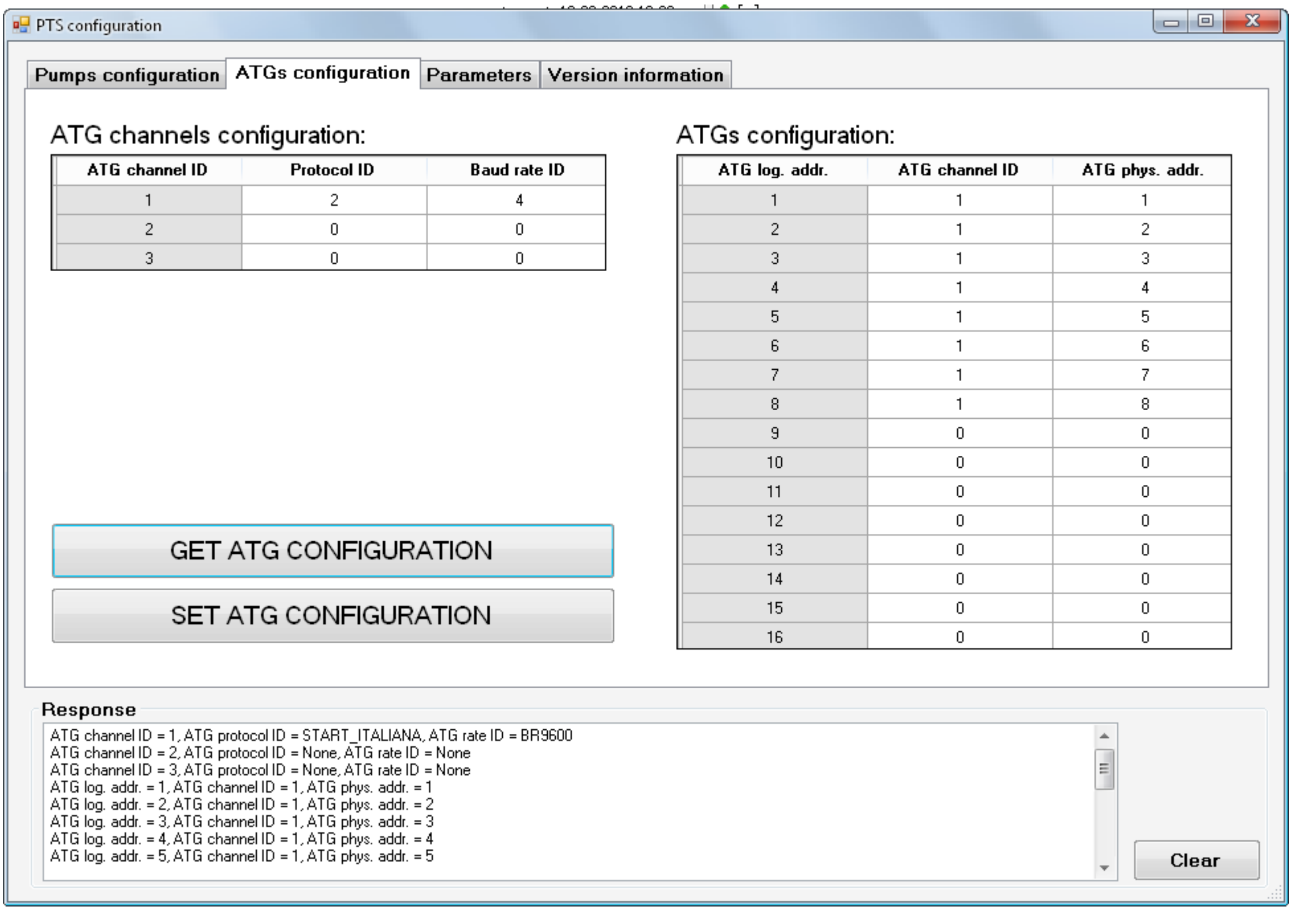
| Pump channel ID | Protocol ID | Baud rate ID |
|---|---|---|
| 1 | 0 | 0 |
| 2 | 0 | 0 |
| 3 | 0 | 0 |
| 4 | 0 | 0 |

Pumps configuration:

| Pump log. addr. | Pump channel ID | Pump phys. addr. |
|---|---|---|
| 1 | 0 | 0 |
| 2 | 0 | 0 |
| 3 | 0 | 0 |
| 4 | 0 | 0 |
| 5 | 0 | 0 |
| 6 | 0 | 0 |
| 7 | 0 | 0 |
| 8 | 0 | 0 |
| 9 | 0 | 0 |
| 10 | 0 | 0 |
| 11 | 0 | 0 |
| 12 | 0 | 0 |
| 13 | 0 | 0 |
| 14 | 0 | 0 |
| 15 | 0 | 0 |
| 16 | 0 | 0 |

```
Pump channel ID = 1, pump protocol ID = TIT_UNIPUMP, baud rate ID = BR9600
Pump channel ID = 2, pump protocol ID = None, baud rate ID = None
Pump channel ID = 3, pump protocol ID = None, baud rate ID = None
Pump channel ID = 4, pump protocol ID = None, baud rate ID = None
Pump log. addr. = 1, pump channel ID = 1, pump phys. addr. = 1
Pump log. addr. = 2, pump channel ID = 1, pump phys. addr. = 2
Pump log. addr. = 3, pump channel ID = 1, pump phys. addr. = 3
Pump log. addr. = 4, pump channel ID = 1, pump phys. addr. = 4
```

ATGs configuration tab allows to configure ATGs channels of PTS controller (like *PTS_conf.exe* utility):



ATG channels configuration:

| ATG channel ID | Protocol ID | Baud rate ID |
|---|---|---|
| 1 | 2 | 4 |
| 2 | 0 | 0 |
| 3 | 0 | 0 |

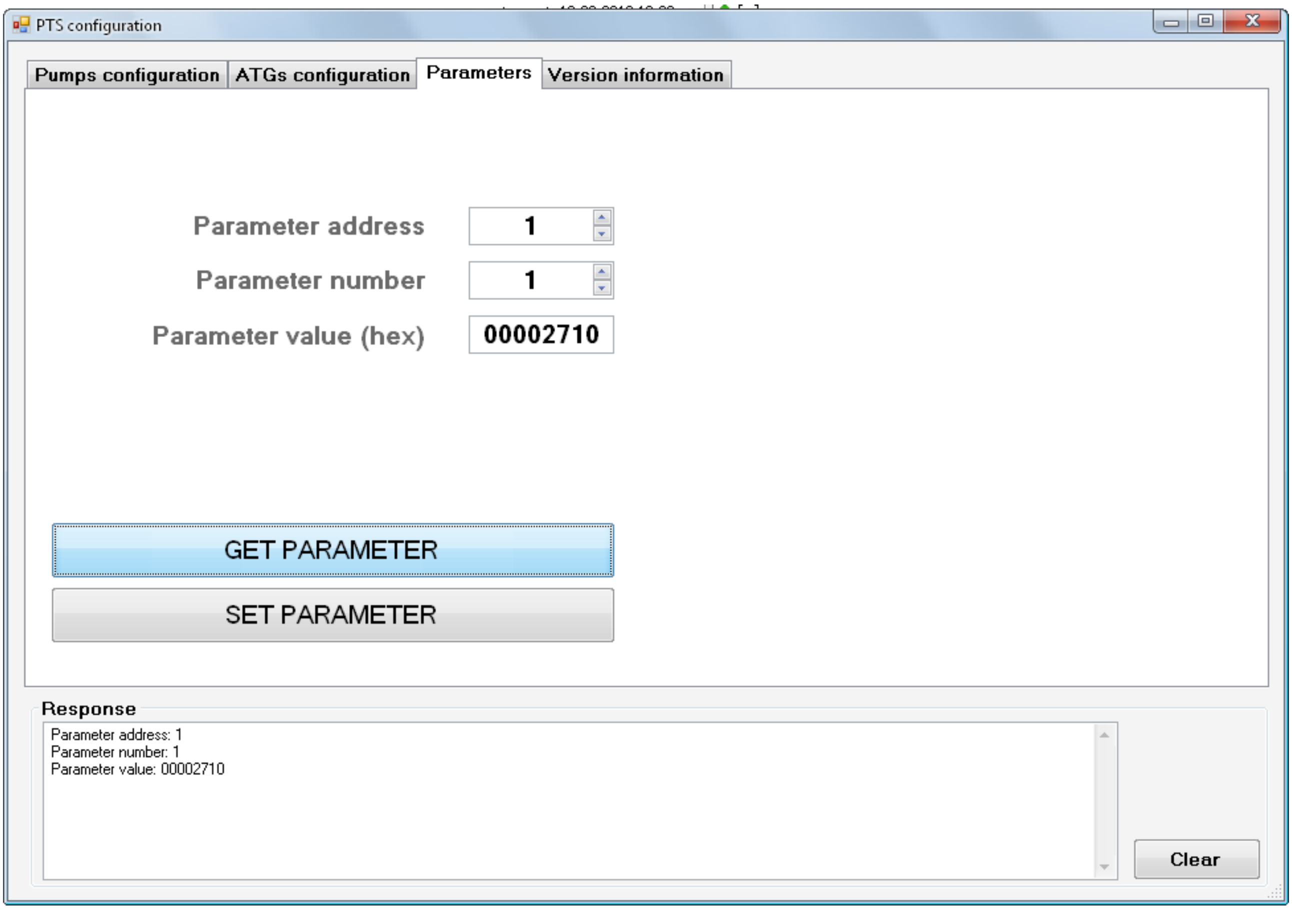
ATGs configuration:

| ATG log. addr. | ATG channel ID | ATG phys. addr. |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 1 | 2 |
| 3 | 1 | 3 |
| 4 | 1 | 4 |
| 5 | 1 | 5 |
| 6 | 1 | 6 |
| 7 | 1 | 7 |
| 8 | 1 | 8 |
| 9 | 0 | 0 |
| 10 | 0 | 0 |
| 11 | 0 | 0 |
| 12 | 0 | 0 |
| 13 | 0 | 0 |
| 14 | 0 | 0 |
| 15 | 0 | 0 |
| 16 | 0 | 0 |

Response

```
ATG channel ID = 1, ATG protocol ID = START_ITALIANA, ATG rate ID = BR9600
ATG channel ID = 2, ATG protocol ID = None, ATG rate ID = None
ATG channel ID = 3, ATG protocol ID = None, ATG rate ID = None
ATG log. addr. = 1, ATG channel ID = 1, ATG phys. addr. = 1
ATG log. addr. = 2, ATG channel ID = 1, ATG phys. addr. = 2
ATG log. addr. = 3, ATG channel ID = 1, ATG phys. addr. = 3
ATG log. addr. = 4, ATG channel ID = 1, ATG phys. addr. = 4
ATG log. addr. = 5, ATG channel ID = 1, ATG phys. addr. = 5
```

Parameters tab allows to read/write parameters in PTS controller (like *PTS_conf.exe* utility):

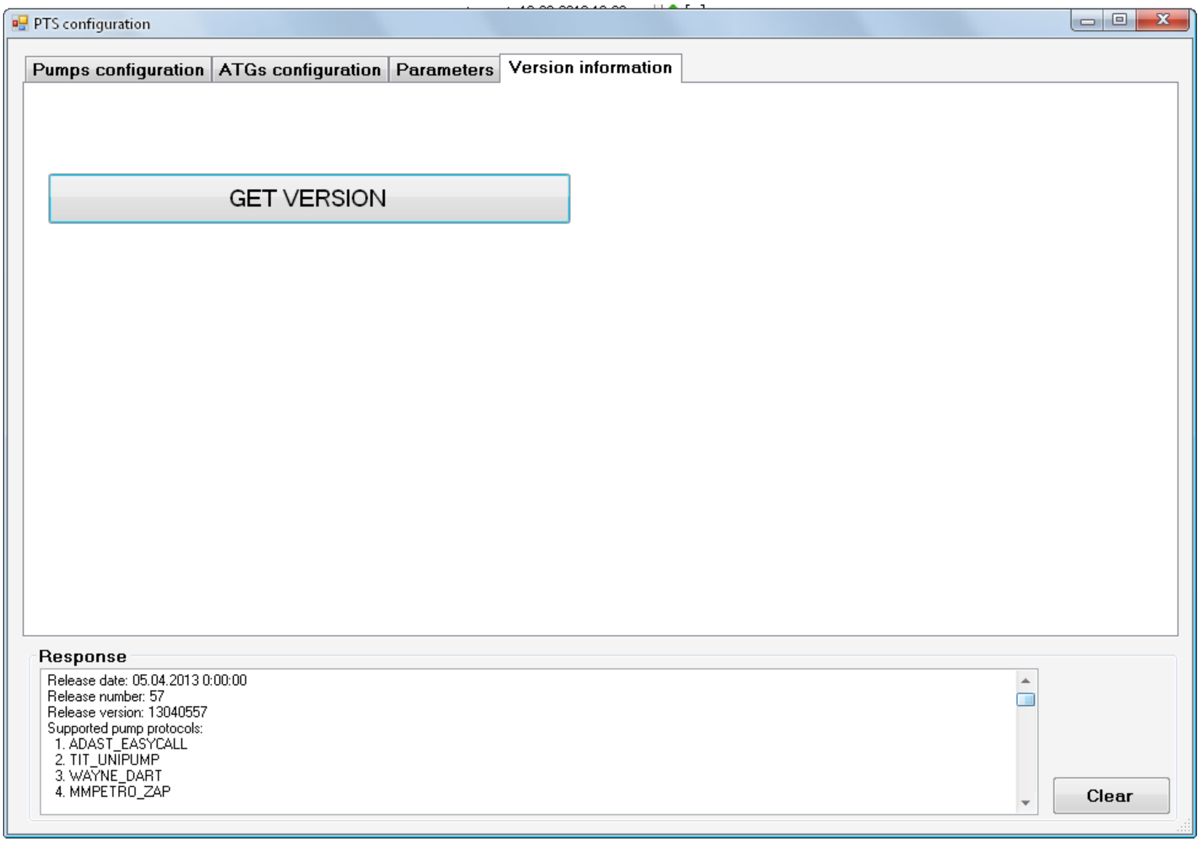


Parameter address: 1
Parameter number: 1
Parameter value (hex): 00002710

Response

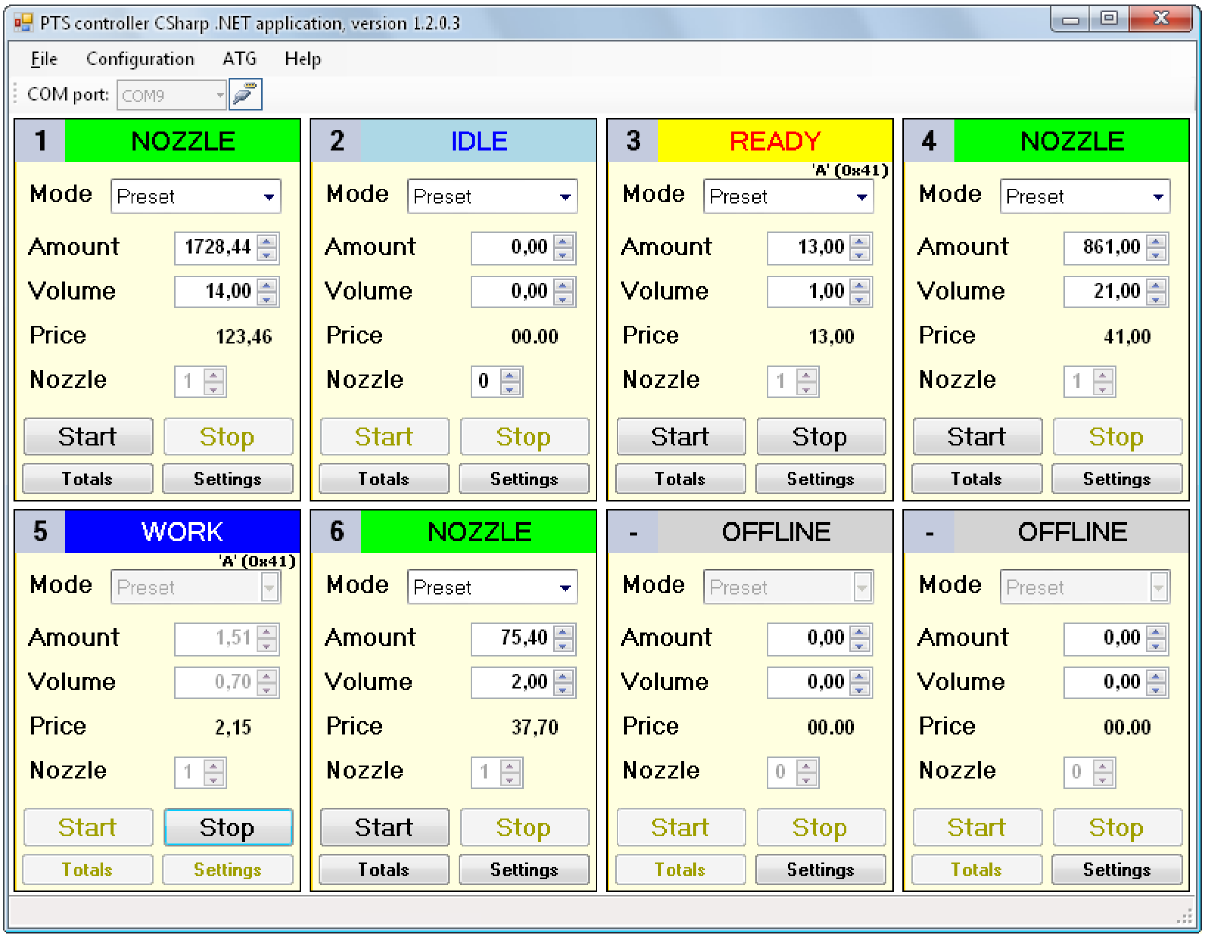
```
Parameter address: 1
Parameter number: 1
Parameter value: 00002710
```

Tab “Version information” allows to reseave information about date and version of PTS controller firmware release and also a list of supported communication protocols of pumps and ATG systems (like *PTS_conf.exe* utility):



```
Release date: 05.04.2013 0:00:00
Release number: 57
Release version: 13040557
Supported pump protocols:
 1. ADAST_EASYCALL
 2. TIT_UNIPUMP
 3. WAYNE_DART
 4. MMPETRO_ZAP
```

## *Main view*



COM port: COM9

| | 1 NOZZLE | 2 IDLE | 3 READY | 4 NOZZLE | 5 WORK | 6 NOZZLE | - OFFLINE | - OFFLINE |
|---|---|---|---|---|---|---|---|---|
| | | | 'A' (0x41) | | 'A' (0x41) | | | |
| Mode | Preset | Preset | Preset | Preset | Preset | Preset | Preset | Preset |
| Amount | 1728,44 | 0,00 | 13,00 | 861,00 | 1,51 | 75,40 | 0,00 | 0,00 |
| Volume | 14,00 | 0,00 | 1,00 | 21,00 | 0,70 | 2,00 | 0,00 | 0,00 |
| Price | 123,46 | 00.00 | 13,00 | 41,00 | 2,15 | 37,70 | 00.00 | 00.00 |
| Nozzle | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |

In main view 8 fuel point icons are shown on the screen, which are to be linked to physical fuel points and show its state.

Each of the displayed fuel point icons contains information about:

- **Fuel point ID** - a logical number of a fuel point in PTS controller (number displayed in top left corner, displaying with a symbol "-" means a fuel point not set)
- **Status** – status of a fuel point (displayed in a top line)
- **Mode** – selection between Preset mode (required to specify money amount or volume of fuel to be dispensed) and Manual mode (does not require to specify money amount or volume of fuel to be dispensed)
- **Amount** – money amount for which a fuel dispensing should be made (in a range 0,00 – 9999,99)
- **Volume** – fuel volume for which a fuel dispensing should be made (in a range 0,00 – 9999,99)
- **Price** – price per 1 liter/gallon of fuel
- **Nozzle** – number of an active nozzle that is taken up (displaying with a symbol "-" means no nozzle is taken up)

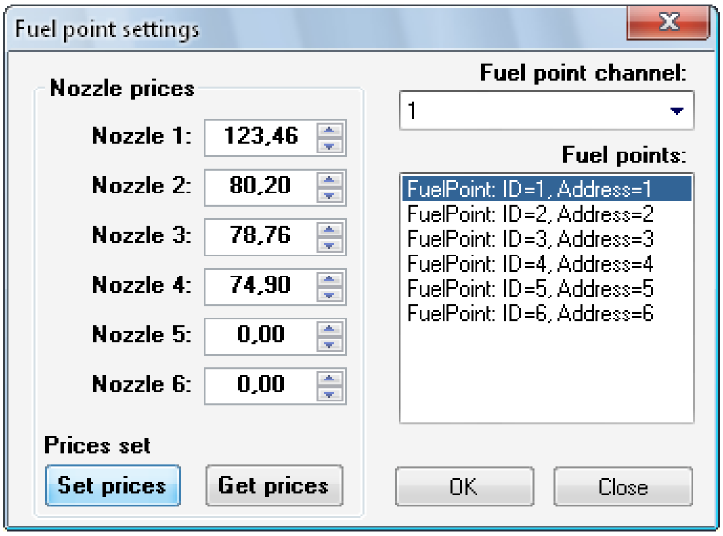
Possible statuses of fuel points:

- **OFFLINE** – fuel point is not active, is not set or is controlled in a manual mode
- **IDLE** – fuel point is in idle state (all nozzles are hang down)
- **NOZZLE** – nozzle is taken up, waiting for allowance on fuel dispensing
- **READY** – fuel dispensing is allowed, test of indicator is running
- **WORK** – fuel dispensing is in process
- **TransactionCompleted** – transaction is finished normally, waiting for a nozzle to be hang down
- **TransactionStopped b**
- **ERROR** – Status of an error

Each of the fuel point icons contains the following buttons:

1. **Start** – start of fuel dispensing (fuel point authorization)
2. **Stop** – stop of fuel dispensing
3. **Settings** – opens a window for setting of fuel point icon configurations
4. **Update prices** – clicking leads to setting of prices per 1 liter/gallon of fuel for all nozzles of a fuel point with which current fuel point icon is linked
5. **Totals** – opens a window for reading of fuel point totalizers

## *Settings of fuel point icons*


Fuel point settings

Nozzle prices

Nozzle 1: 123,46
Nozzle 2: 80,20
Nozzle 3: 78,76
Nozzle 4: 74,90
Nozzle 5: 0,00
Nozzle 6: 0,00

Fuel point channel: 1

Fuel points:
FuelPoint: ID=1, Address=1
FuelPoint: ID=2, Address=2
FuelPoint: ID=3, Address=3
FuelPoint: ID=4, Address=4
FuelPoint: ID=5, Address=5
FuelPoint: ID=6, Address=6

Given window contains a dropdown list with all 4 pump channels of the PTS controller, selecting each of which leads to displaying of all fuel points' IDs and corresponding fuel points' addresses configured to given pump channel for a PTS controller in a field below. Selecting one of the fuel point IDs and pressing OK button leads to linking given fuel point icon to selected fuel point ID and displaying its status. Selecting of a PTS pump channel with value 0 will lead to switching off a fuel point icon.

Given window for each of the fuel points also contains a list of 6 nozzles for setting of prices per 1 liter/gallon of fuel, dispensed through these nozzles, in a range 0,00 – 999,99. If a fuel point has less than 6 nozzles – prices will be set only for first number of nozzles, which a fuel point has.

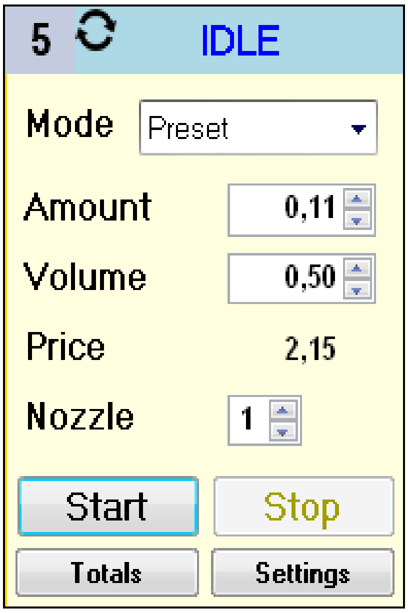
Clicking 'Set prices' button leads to updating prices on selected fuel point. Clicking 'Get prices' leads to receiving of prices from selected fuel point.

Clicking OK button leads to updating a fuel point icon and updating (creating) a file "Config.xml" in root folder with configuration of fuel point icons. At launching of a PTS controller .NET application next time configuration of fuel point icons will be read from this file "Config.xml" in root folder.

## *Reading of fuel point total counters*

In the end of every dispensing application itself reads total counters from the pump on the nozzle, whih was dispensing. At this a sign of totals reading appears in the left top corner of the pump icon (and under the header of the 'T' command is displayed, which means that currently PTS controller processes TotalInfoRequest command).

Total counters can be updated at any moment from any of the pumps. Selecting of a button "Totals" on a fuel point icon and clicking a button Update in it leads to reading of total counters:



5 IDLE

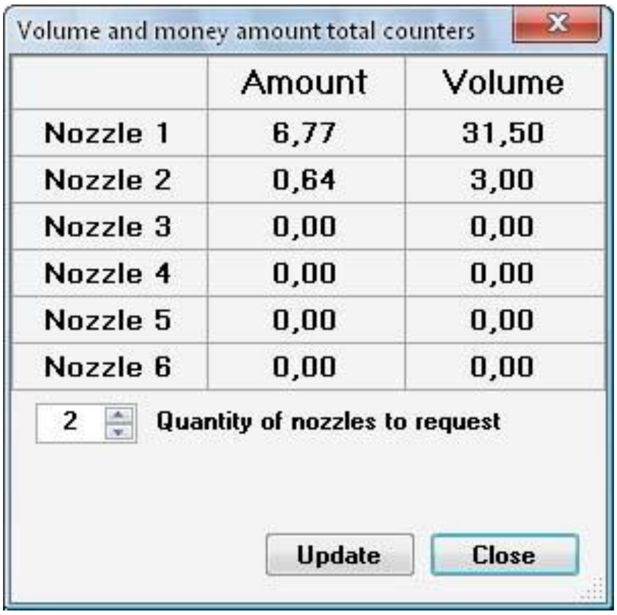
Mode: Preset

Amount: 0,11

Volume: 0,50

Price: 2,15

Nozzle: 1



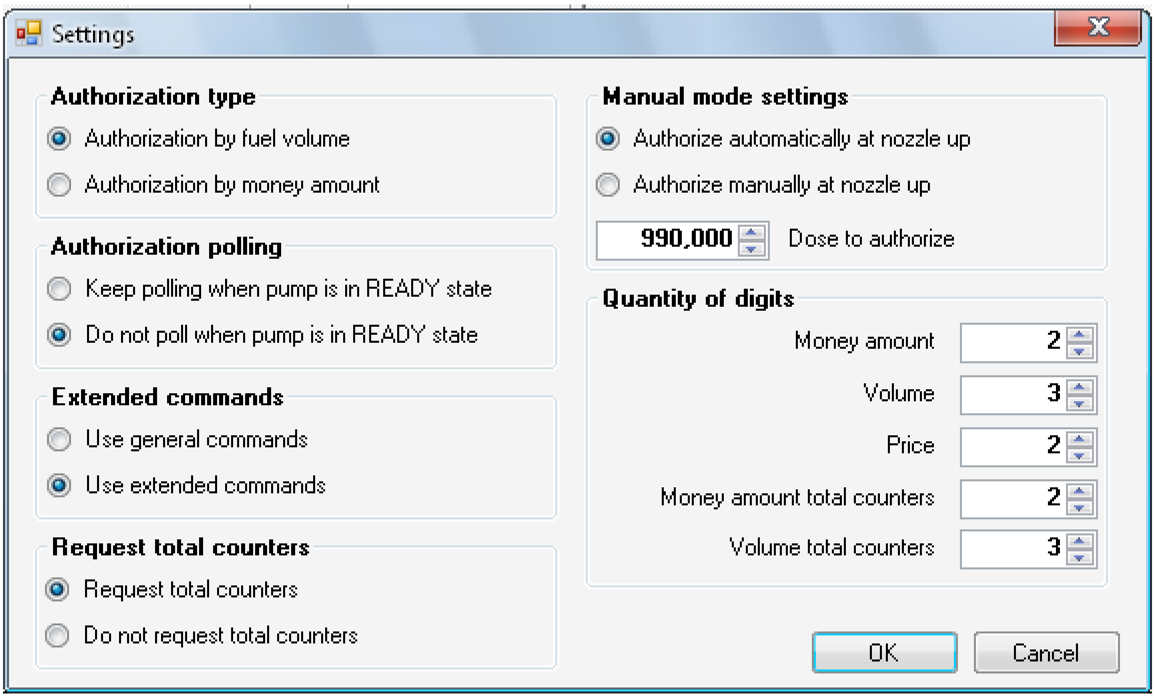
Volume and money amount total counters

| | Amount | Volume |
|---|---|---|
| Nozzle 1 | 6,77 | 31,50 |
| Nozzle 2 | 0,64 | 3,00 |
| Nozzle 3 | 0,00 | 0,00 |
| Nozzle 4 | 0,00 | 0,00 |
| Nozzle 5 | 0,00 | 0,00 |
| Nozzle 6 | 0,00 | 0,00 |

2 Quantity of nozzles to request

## *Settings*

Selection from top menu "Configuration" an item "Settings" leads to opening a new window, which allows adjustment of



Settings

**Authorization type**
- ● Authorization by fuel volume
- ○ Authorization by money amount

**Authorization polling**
- ○ Keep polling when pump is in READY state
- ● Do not poll when pump is in READY state

**Extended commands**
- ○ Use general commands
- ● Use extended commands

**Request total counters**
- ● Request total counters
- ○ Do not request total counters

**Manual mode settings**
- ● Authorize automatically at nozzle up
- ○ Authorize manually at nozzle up

990,000 Dose to authorize

**Quantity of digits**

| | |
|---|---|
| Money amount | 2 |
| Volume | 3 |
| Price | 2 |
| Money amount total counters | 2 |
| Volume total counters | 3 |

Settings include the following:

1. Authorization type of fuel dispensers:
   - **Authorization by fuel volume** – dispensers are to be authorized with volume
   - **Authorization by money amount** – dispensers are to be authorized with money amount

2. Authorization polling:
   - **Keep polling when pump is in READY state** – after the Start button is clicked to authorize a dispenser, but nozzle is not taken up on the dispenser, the fuel point displays READY state; this option enables to keep constantly sending authorization command in READY state (until nozzle is taken up of fuel point and it is authorized)
   - **Do not poll when pump is in READY state** – after the Start button is clicked to authorize a dispenser, but nozzle is not taken up on the dispenser, the fuel point displays READY state; this option enables to send authorization command only once at clicking on a 'Start' button of fuel point icon

3. Extended commands:
   - **Use general commands** – PTS controller can work with general commands (for prices containing up to 4 digits) and also with extended commands (for more quantity of digits in price values); this option sets to use general commands in communication with PTS controller (prices containing with up to 4 digits)
   - **Use extended commands** – PTS controller can work with general commands (for prices containing up to 4 digits) and also with extended commands (for prices containing with up to 9 digits); this option sets to use extended commands in communication with PTS controller (prices containing with up to 9 digits)

4. Request total counters:
   - **Request total counters** – this option sets that application should automatically request total counters after each dispensing on dispenser is performed

- **Do not request total counters** – this option sets that application should not automatically request total counters after each dispensing on dispenser is performed

5. Manual mode settings:
   - **Authorize automatically at nozzle up** – this option sets that once a nozzle is taken up on the dispenser – application should at once authorize it with a value set in field “Dose to authorize”; this options enables the application to set in a way, when order is preset from dispenser keyboard and application simply monitors the dispensers operation; in order this feature to work a fuel point should be switched to Manual mode of operation (see above), in Preset mode this feature does not work (preset is supposed to be given from application)
   - **Authorize manually at nozzle up** – this option sets that dispensers are to be controlled fully from the application by clicking on Start and Stop buttons both in preset and Manual modes of operation

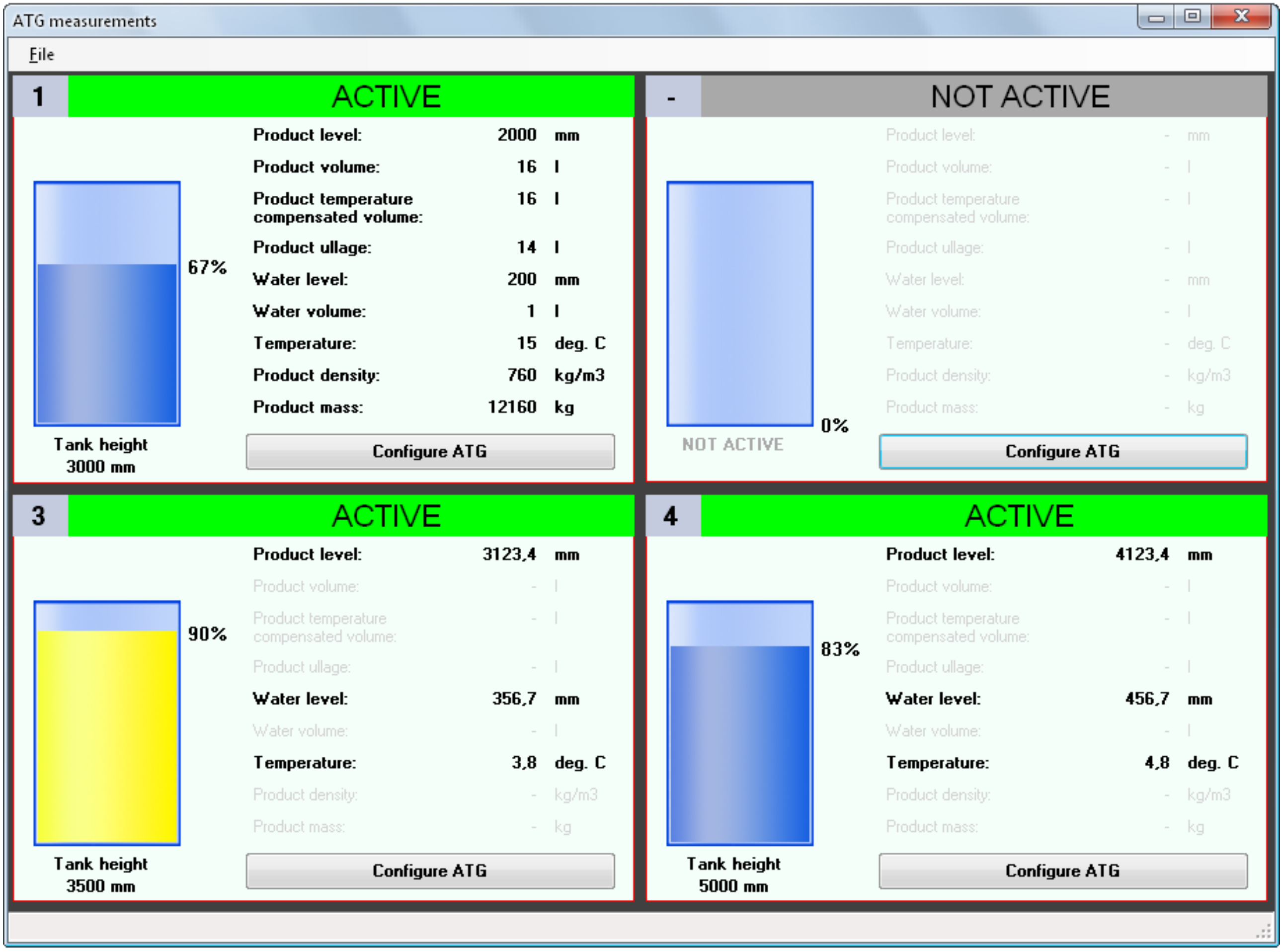
6. Quantity of digits:
   **Setting of quantity of digits in money, volume, price and total counters (volume and money amount)** - this option is required in cases when dispensers may have non-standard quantity of digits (for example 3 digits in volume and 0 digits in money amount)

## *Displaying ATG systems probes states*

In ATG view 4 ATG icons are shown on the screen, which are to be linked to physical ATG systems (probes) and show its state.

Each of the displayed ATG icons contains information about measured by ATG system (probe) parameters of petroleum product inside tanks. Depending on the type (communication protocol) of ATG system (probe) the following data may be displayed:

- **Product level** (accuracy 0.1 mm)
- **Product volume** (accuracy 1 liter)
- **Product temperature compensated volume** (accuracy 1 liter)
- **Product ullage (volume)** (accuracy 1 liter)
- **Water level** (accuracy 0.1 mm)
- **Water volume** (accuracy 1 liter)
- **Temperature** (accuracy 0.1 degree Celcium)
- **Product density** (accuracy 0.1 kg/m$^3$)
- **Product mass** (accuracy 0.1 kg)



1 ACTIVE

Product level: 2000 mm
Product volume: 16 l
Product temperature compensated volume: 16 l
Product ullage: 14 l
Water level: 200 mm
Water volume: 1 l
Temperature: 15 deg. C
Product density: 760 kg/m3
Product mass: 12160 kg

67%
Tank height 3000 mm

- NOT ACTIVE

Product level: - mm
Product volume: - l
Product temperature compensated volume: - l
Product ullage: - l
Water level: - mm
Water volume: - l
Temperature: - deg. C
Product density: - kg/m3
Product mass: - kg

0%
NOT ACTIVE

3 ACTIVE

Product level: 3123,4 mm
Product volume: - l
Product temperature compensated volume: - l
Product ullage: - l
Water level: 356,7 mm
Water volume: - l
Temperature: 3,8 deg. C
Product density: - kg/m3
Product mass: - kg

90%
Tank height 3500 mm

4 ACTIVE

Product level: 4123,4 mm
Product volume: - l
Product temperature compensated volume: - l
Product ullage: - l
Water level: 456,7 mm
Water volume: - l
Temperature: 4,8 deg. C
Product density: - kg/m3
Product mass: - kg

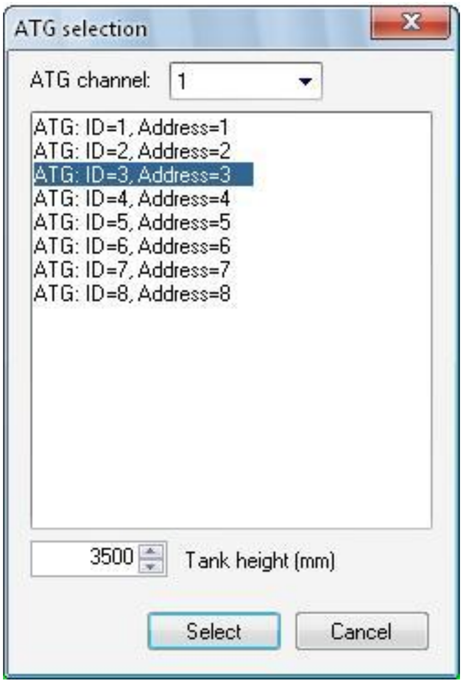
83%
Tank height 5000 mm

If ATG system does not support any the parameters – a correspondent value label on ATG icon stays disabled. Picture of a tank on the ATG icon shows visually level of product in tank compared to tank height (which should be equal to ATG probe height). If ATG icon is linked with an ATG system – its state is "ACTIVE", else – "NOT ACTIVE".

## *Settings of ATG probe icons*

Given window contains a dropdown list with all 3 ATG channels of the PTS controller, selecting each of which leads to displaying of all ATG probes' IDs and corresponding ATG probes' addresses configured to given ATG channel for a PTS controller in a field below. Selecting one of the ATG probe IDs and pressing OK button leads to linking given ATG probe icon to selected ATG probe ID and displaying its state. Selecting of a PTS ATG channel with value 0 will lead to switching off a ATG probe icon.

Given window for each of the ATG probes also contains a field for entering height of a tank with installed probe (which is considered to be a maximum level of product allowed for given tank).



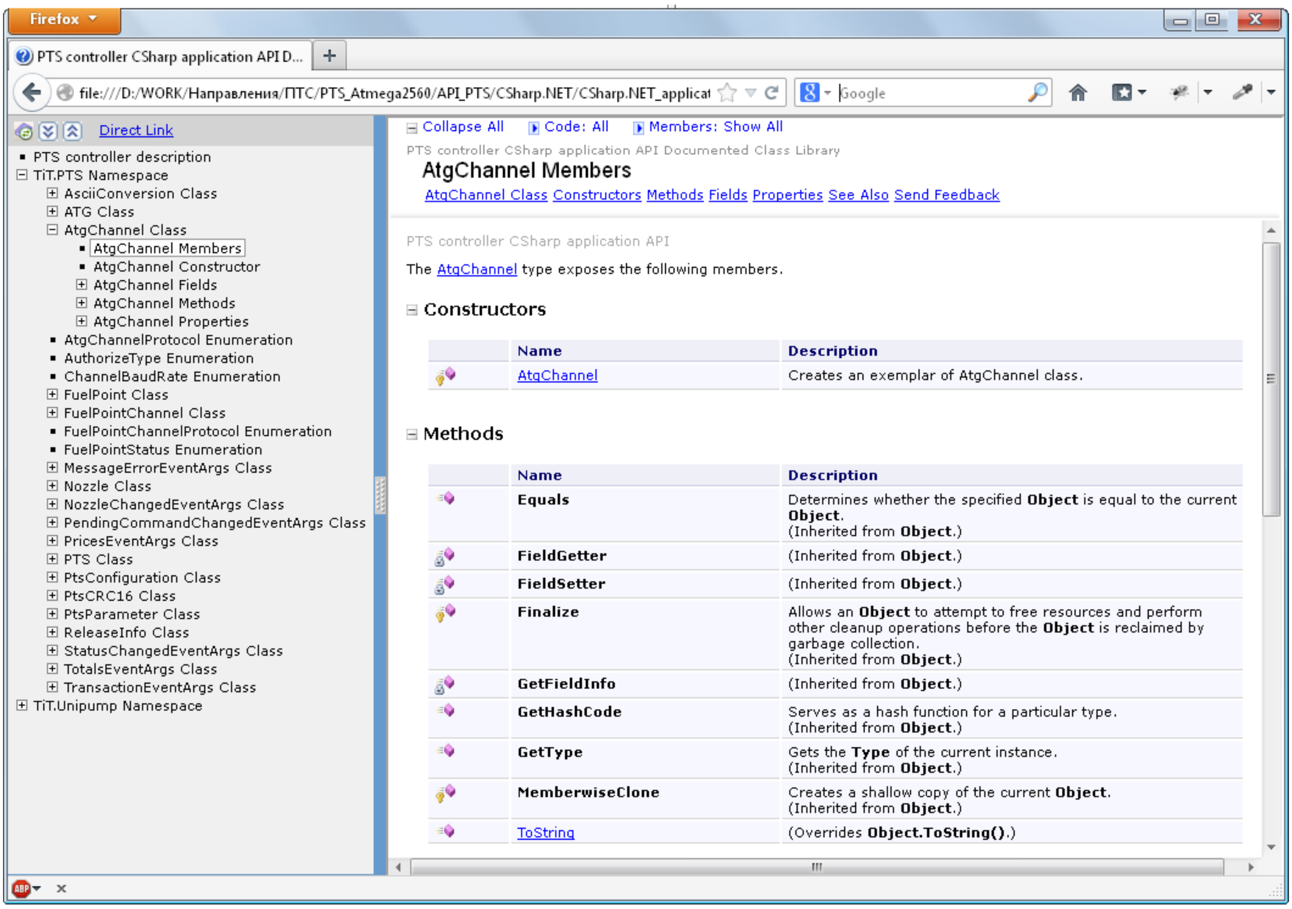
ATG channel: 1

ATG: ID=1, Address=1
ATG: ID=2, Address=2
ATG: ID=3, Address=3
ATG: ID=4, Address=4
ATG: ID=5, Address=5
ATG: ID=6, Address=6
ATG: ID=7, Address=7
ATG: ID=8, Address=8

3500 Tank height (mm)

## *TiT.PTS classes help file*

PTS controller .NET application is based on a number of classes which documentation describing methods and properties is supplied together with it. Documentation is generated in a form of html-files.


Firefox
PTS controller CSharp application API D...
file:///D:/WORK/Направления/ПТС/PTS_Atmega2560/API_PTS/CSharp.NET/CSharp.NET_applicat
Google
Direct Link
Collapse All
Code: All
Members: Show All

- PTS controller description
- TiT.PTS Namespace
  - AsciiConversion Class
  - ATG Class
  - AtgChannel Class
    - AtgChannel Members
    - AtgChannel Constructor
    - AtgChannel Fields
    - AtgChannel Methods
    - AtgChannel Properties
  - AtgChannelProtocol Enumeration
  - AuthorizeType Enumeration
  - ChannelBaudRate Enumeration
  - FuelPoint Class
  - FuelPointChannel Class
  - FuelPointChannelProtocol Enumeration
  - FuelPointStatus Enumeration
  - MessageErrorEventArgs Class
  - Nozzle Class
  - NozzleChangedEventArgs Class
  - PendingCommandChangedEventArgs Class
  - PricesEventArgs Class
  - PTS Class
  - PtsConfiguration Class
  - PtsCRC16 Class
  - PtsParameter Class
  - ReleaseInfo Class
  - StatusChangedEventArgs Class
  - TotalsEventArgs Class
  - TransactionEventArgs Class
- TiT.Unipump Namespace

PTS controller CSharp application API Documented Class Library

AtgChannel Members

AtgChannel Class Constructors Methods Fields Properties See Also Send Feedback

PTS controller CSharp application API

The AtgChannel type exposes the following members.

Constructors

| | Name | Description |
|---|---|---|
| | AtgChannel | Creates an exemplar of AtgChannel class. |

Methods

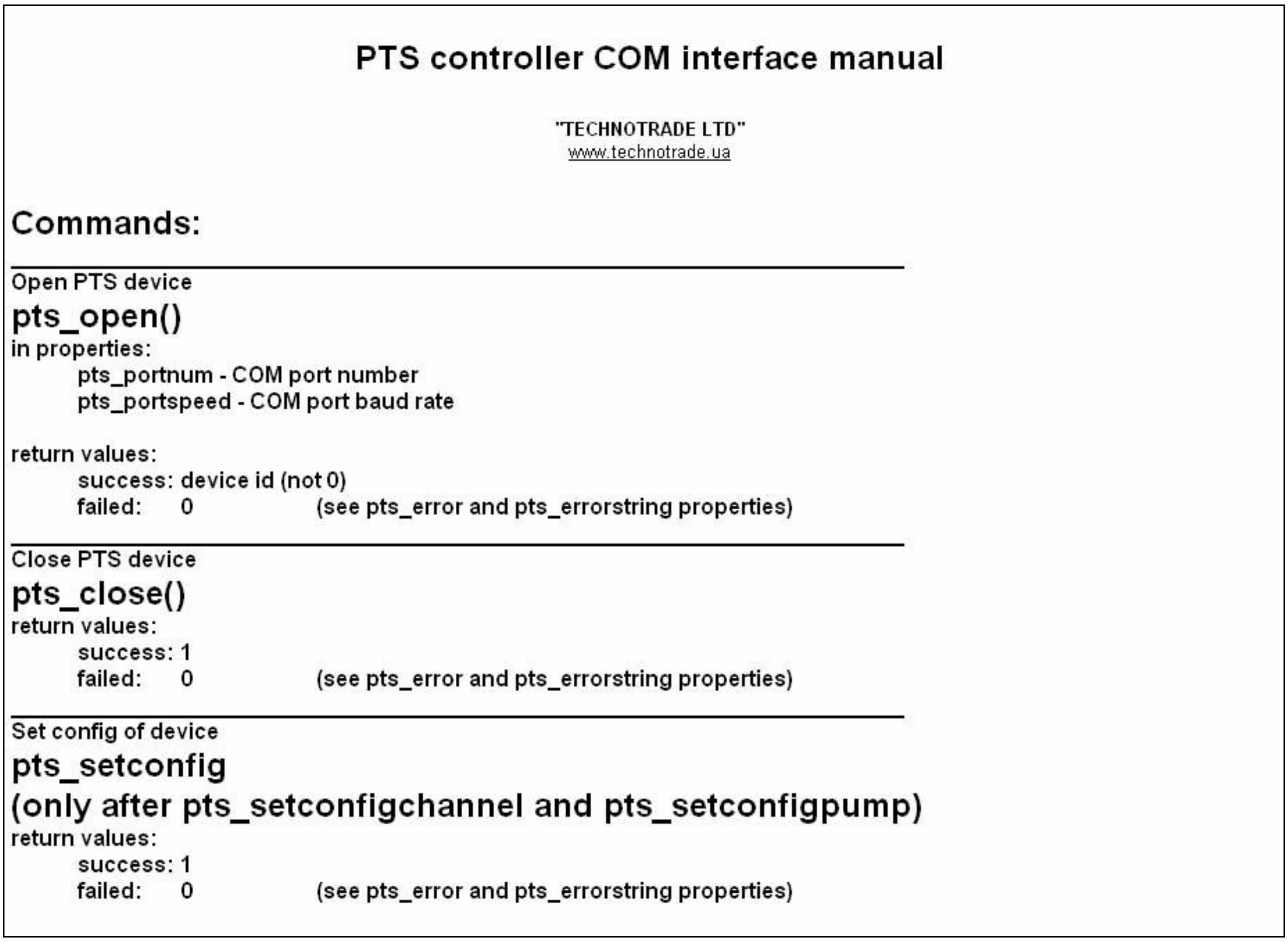
| | Name | Description |
|---|---|---|
| | **Equals** | Determines whether the specified **Object** is equal to the current **Object**.<br>(Inherited from **Object**.) |
| | **FieldGetter** | (Inherited from **Object**.) |
| | **FieldSetter** | (Inherited from **Object**.) |
| | **Finalize** | Allows an **Object** to attempt to free resources and perform other cleanup operations before the **Object** is reclaimed by garbage collection.<br>(Inherited from **Object**.) |
| | **GetFieldInfo** | (Inherited from **Object**.) |
| | **GetHashCode** | Serves as a hash function for a particular type.<br>(Inherited from **Object**.) |
| | **GetType** | Gets the **Type** of the current instance.<br>(Inherited from **Object**.) |
| | **MemberwiseClone** | Creates a shallow copy of the current **Object**.<br>(Inherited from **Object**.) |
| | ToString | (Overrides **Object.ToString()**.) |

## *PTS COM-object*

In order to provide control over PTS controller and simplify work of developers, who are working in Windows OS in implementation of UniPump communication protocol of PTS controller, a COM-object with methods and properties for operation with PTS controller is provided.

COM-object is compiled on Visual C++ 6.0 and is supplied together with a test application, written on Visual Basic 6.0.

Documentation of COM-object describes all methods and properties of the COM-object and its application.



**PTS controller COM interface manual**

"TECHNOTRADE LTD"
www.technotrade.ua

**Commands:**

Open PTS device
**pts_open()**
in properties:
pts_portnum - COM port number
pts_portspeed - COM port baud rate

return values:
success: device id (not 0)
failed: 0 (see pts_error and pts_errorstring properties)

Close PTS device
**pts_close()**
return values:
success: 1
failed: 0 (see pts_error and pts_errorstring properties)

Set config of device
**pts_setconfig**
**(only after pts_setconfigchannel and pts_setconfigpump)**
return values:
success: 1
failed: 0 (see pts_error and pts_errorstring properties)

To register PTSDriver.dll library in Windows OS perform command

```
REGSVR32 PTSDriver.dll
```

from
PTS_SDK\API TOOLS\COM_object_API_plus_VB6.0_appilcation.rar\PTSDriver\ReleaseMinDependency\ folder.

# *PTS controller test utility*

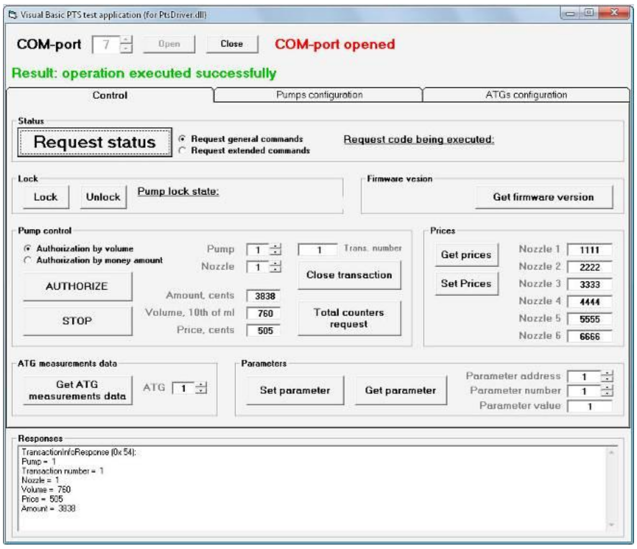
## *Purpose*

PTS controller test utility is written in the following languages:

- C#
- VB 6.0
- VB.NET
- Java

It is written for developers in order to give open source test application, which shows how to communicate with PTS controller using commands from UniPump communication protocol without a necessity to implement this protocol from zero.

Utility provides all operations with a PTS controller. Thus developers using one of the above languages can view how control over PTS controller is provided in the utility and implement the same functions in own developed applications for petrol stations.



COM-port 7 — COM-port opened

Result: operation executed successfully

● Request general commands ○ Request extended commands

● Authorization by volume ○ Authorization by money amount

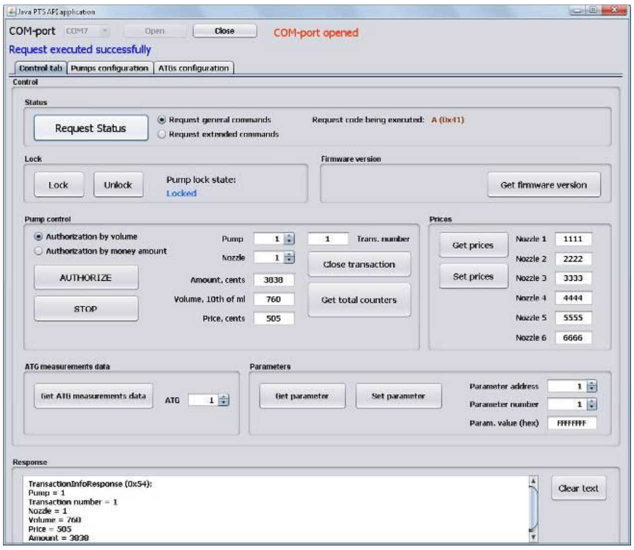
Pump 1, Nozzle 1, Trans. number 1, Amount, cents 3838, Volume, 10th of ml 760, Price, cents 505

Nozzle 1 1111, Nozzle 2 2222, Nozzle 3 3333, Nozzle 4 4444, Nozzle 5 5555, Nozzle 6 6666

ATG 1; Parameter address 1, Parameter number 1, Parameter value 1

```
TransactionInfoResponse (0x 54):
Pump = 1
Transaction number = 1
Nozzle = 1
Volume = 760
Price = 505
Amount = 3838
```

Visual Basic 6.0 (uses COM-object)



COM-port COM7 — COM-port opened

Request executed successfully

● Request general commands ○ Request extended commands

Request code being executed: A (0x41)

Pump lock state: Locked

● Authorization by volume ○ Authorization by money amount

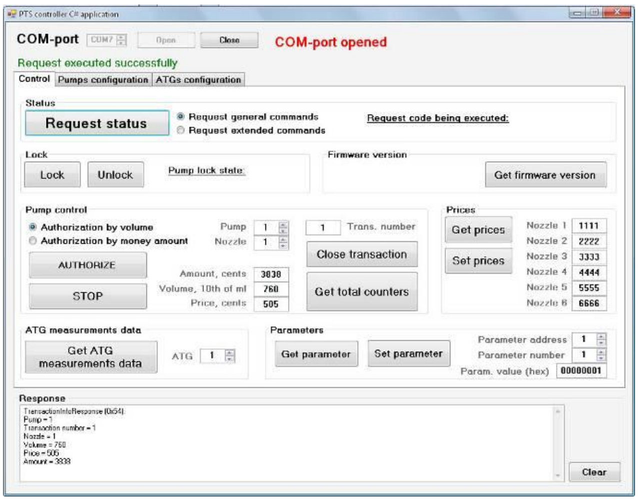
Pump 1, Nozzle 1, Trans. number 1, Amount, cents 3838, Volume, 10th of ml 760, Price, cents 505

Nozzle 1 1111, Nozzle 2 2222, Nozzle 3 3333, Nozzle 4 4444, Nozzle 5 5555, Nozzle 6 6666

ATG 1; Parameter address 1, Parameter number 1, Param. value (hex) FFFFFFFF

```
TransactionInfoResponse (0x54):
Pump = 1
Transaction number = 1
Nozzle = 1
Volume = 760
Price = 505
Amount = 3838
```

Java (developed in NetBeans IDE)



COM-port COM7 — COM-port opened

Request executed successfully

● Request general commands ○ Request extended commands

● Authorization by volume ○ Authorization by money amount

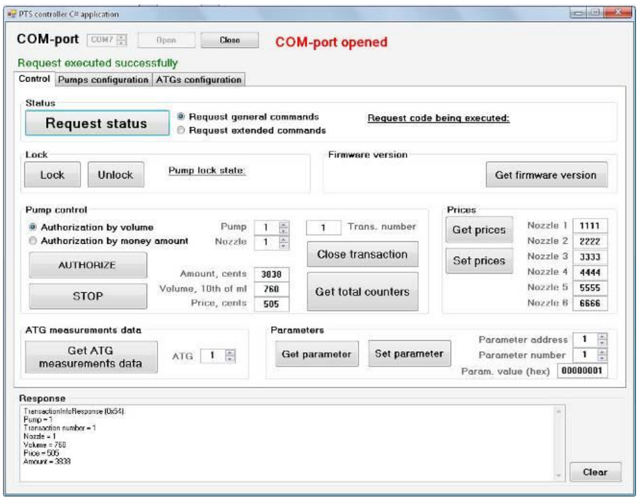
Pump 1, Nozzle 1, Trans. number 1, Amount, cents 3838, Volume, 10th of ml 760, Price, cents 505

Nozzle 1 1111, Nozzle 2 2222, Nozzle 3 3333, Nozzle 4 4444, Nozzle 5 5555, Nozzle 6 6666

ATG 1; Parameter address 1, Parameter number 1, Param. value (hex) 00000001

```
TransactionInfoResponse (0x54):
Pump = 1
Transaction number = 1
Nozzle = 1
Volume = 760
Price = 505
Amount = 3838
```

C# (developed in Visual Studio Express 2010)



COM-port COM7 — COM-port opened

Request executed successfully

● Request general commands ○ Request extended commands

● Authorization by volume ○ Authorization by money amount

Pump 1, Nozzle 1, Trans. number 1, Amount, cents 3838, Volume, 10th of ml 760, Price, cents 505

Nozzle 1 1111, Nozzle 2 2222, Nozzle 3 3333, Nozzle 4 4444, Nozzle 5 5555, Nozzle 6 6666

ATG 1; Parameter address 1, Parameter number 1, Param. value (hex) 00000001

```
TransactionInfoResponse (0x54):
Pump = 1
Transaction number = 1
Nozzle = 1
Volume = 760
Price = 505
Amount = 3838
```

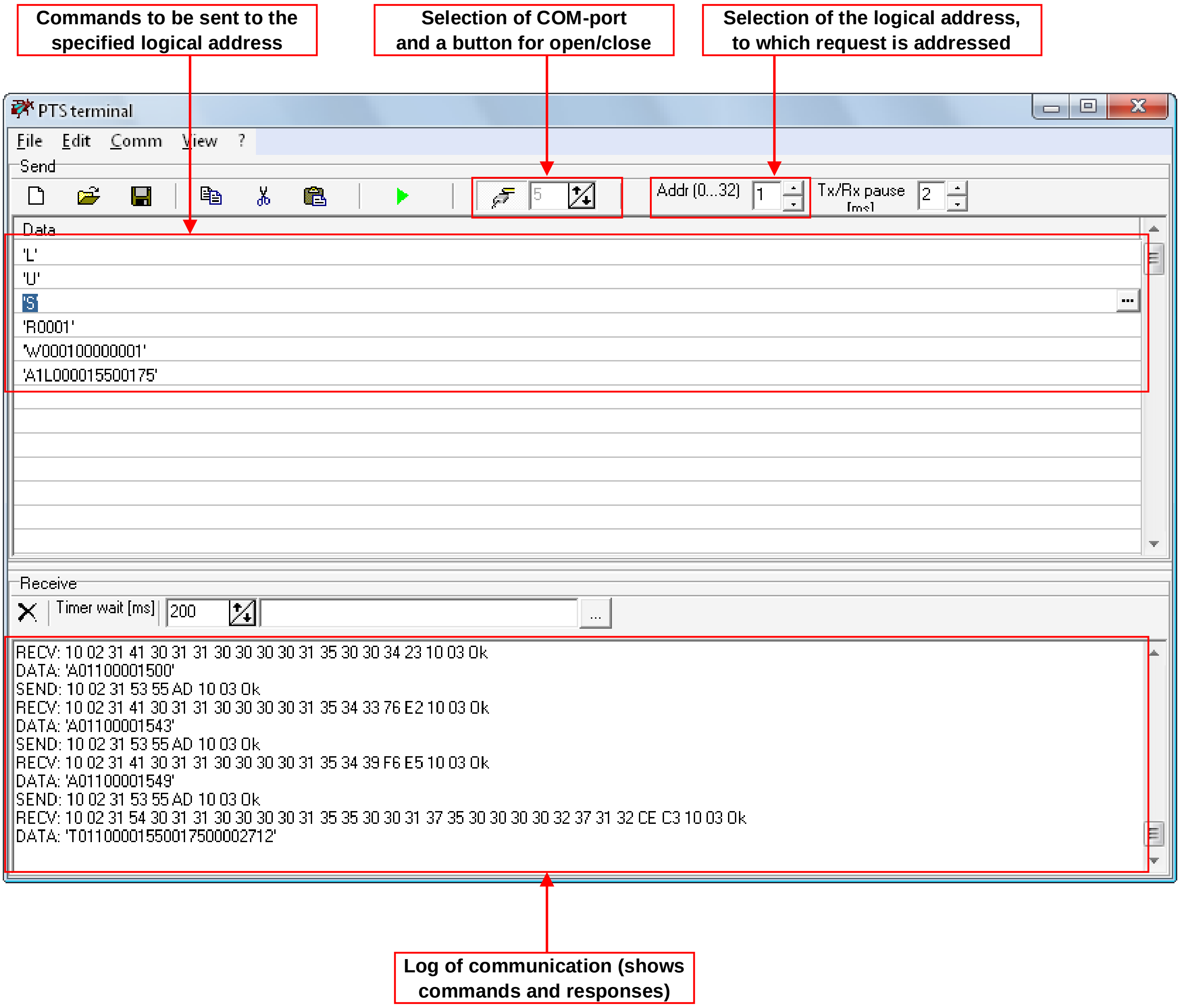
VB.NET (developed in Visual Studio Express 2010)

# *PTS terminal utility*

## *Purpose*

PTS terminal is used in order to provide debugging of PTS controller operation. It allows to send to PTS controller requests formed with bytes in accordance with UniPump communication protocol commands without a necessity to add framing bytes and CRC.

This utility can be useful during development of own control software, when developers can test sending of commands to PTS controller and receive its responses using this utility and in such a way compare then with responses processed in own developed control software.



Commands to be sent to the specified logical address

Selection of COM-port and a button for open/close

Selection of the logical address, to which request is addressed

Addr (0...32) 1

Tx/Rx pause [ms] 2

```
'L'
'U'
'S'
'R0001'
'W000100000001'
'A1L000015500175'
```

Timer wait [ms] 200

```
RECV: 10 02 31 41 30 31 31 30 30 30 30 31 35 30 30 34 23 10 03 Ok
DATA: 'A01100001500'
SEND: 10 02 31 53 55 AD 10 03 Ok
RECV: 10 02 31 41 30 31 31 30 30 30 30 31 35 34 33 76 E2 10 03 Ok
DATA: 'A01100001543'
SEND: 10 02 31 53 55 AD 10 03 Ok
RECV: 10 02 31 41 30 31 31 30 30 30 30 31 35 34 39 F6 E5 10 03 Ok
DATA: 'A01100001549'
SEND: 10 02 31 53 55 AD 10 03 Ok
RECV: 10 02 31 54 30 31 31 30 30 30 30 31 35 35 30 30 31 37 35 30 30 30 30 32 37 31 32 CE C3 10 03 Ok
DATA: 'T01100001550017500002712'
```

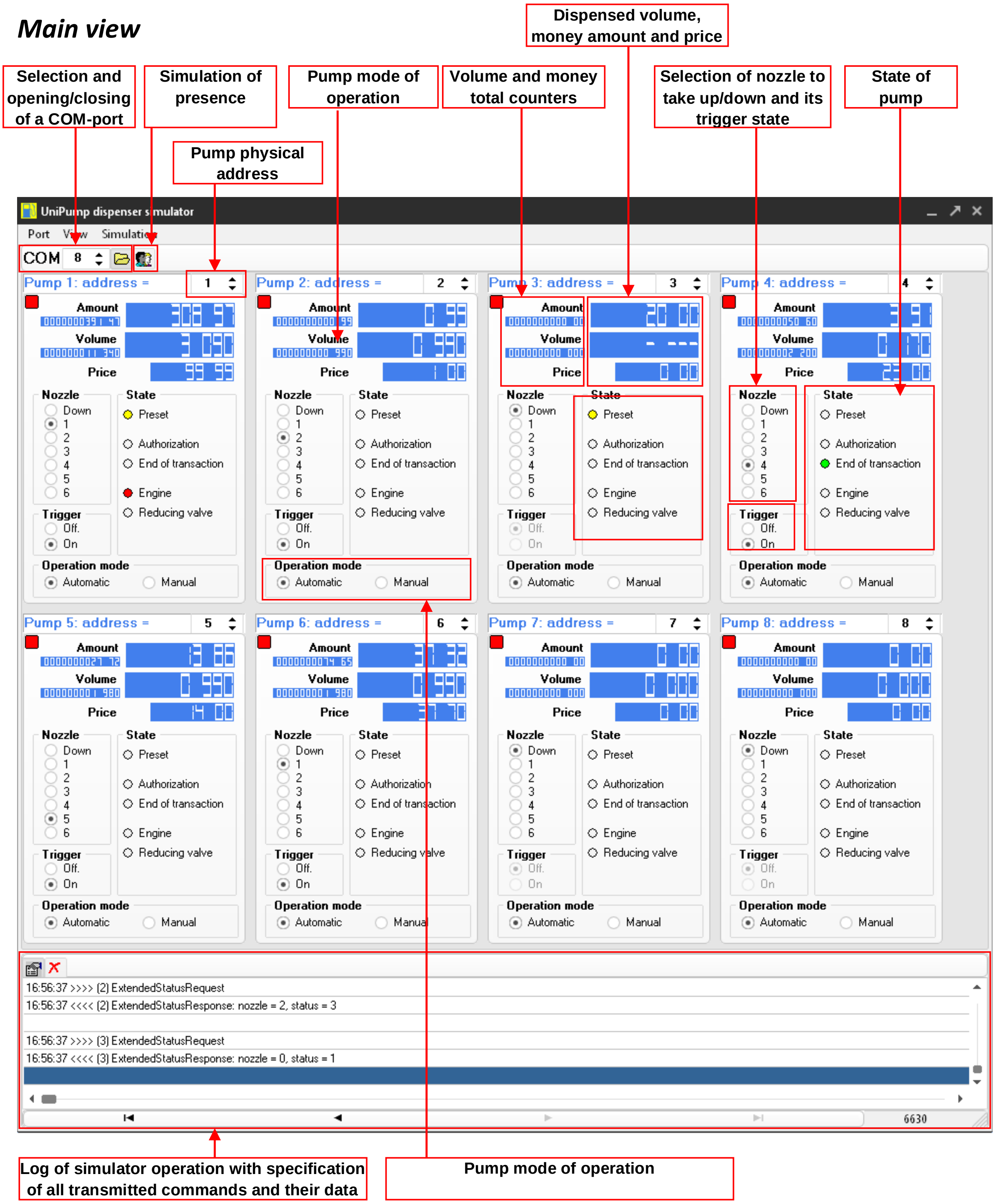
Log of communication (shows commands and responses)

# UniPump pumps software simulator

## Purpose

PTS controller SDK includes a pumps software simulator with a purpose to enable debugging of PTS controller implementation in third party software. Purpose of software simulators is to debug operation of control system software (cash register, POS system, OPT, etc) over fuel dispensers through PTS controller without a necessity to connect to real fuel dispensers.

## Main view



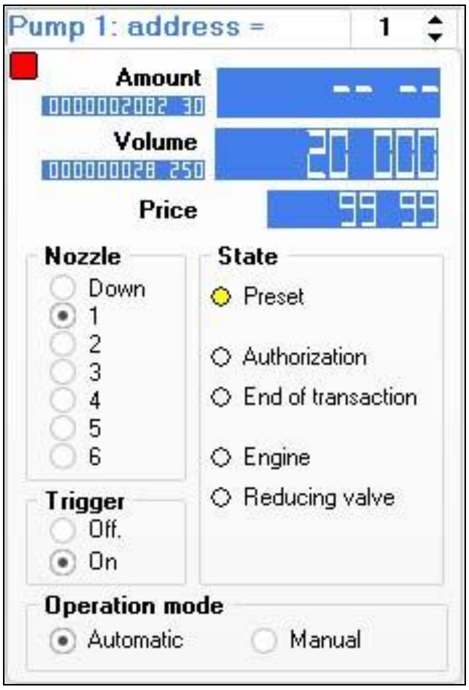
```
16:56:37 >>>> [2] ExtendedStatusRequest
16:56:37 <<<< [2] ExtendedStatusResponse: nozzle = 2, status = 3

16:56:37 >>>> [3] ExtendedStatusRequest
16:56:37 <<<< [3] ExtendedStatusResponse: nozzle = 0, status = 1
```

*Pumps software simulator main view*

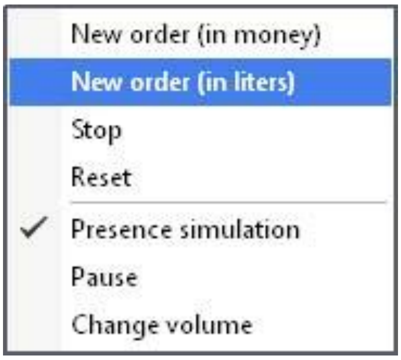
In pumps simulator it is possible to add up to 99 pumps giving each a unique physical address. Each of the pumps leads separately its total counters (volume and money amount) and prices for each of its 6 nozzles. Total counter values as well as prices are stored in Windows system register. Each of the nozzles has a trigger.

Each of the pumps can be transferred to automatic mode of operation (controlled from the control system) or manual mode of operation (controlled manually and does not response to control system).

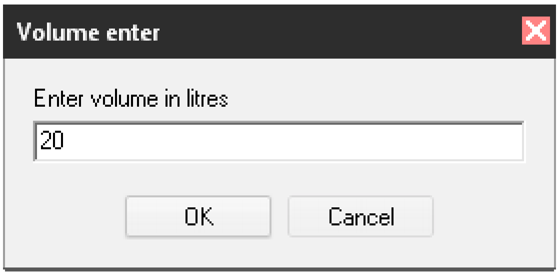


*Pump control in pumps software simulator*

In right-click mouse menu it is possible to preset the pump with money amount or volume value, stop the pump, reset, pause and change volume:



*Pump right-button mouse menu*



Enter volume in litres: 20

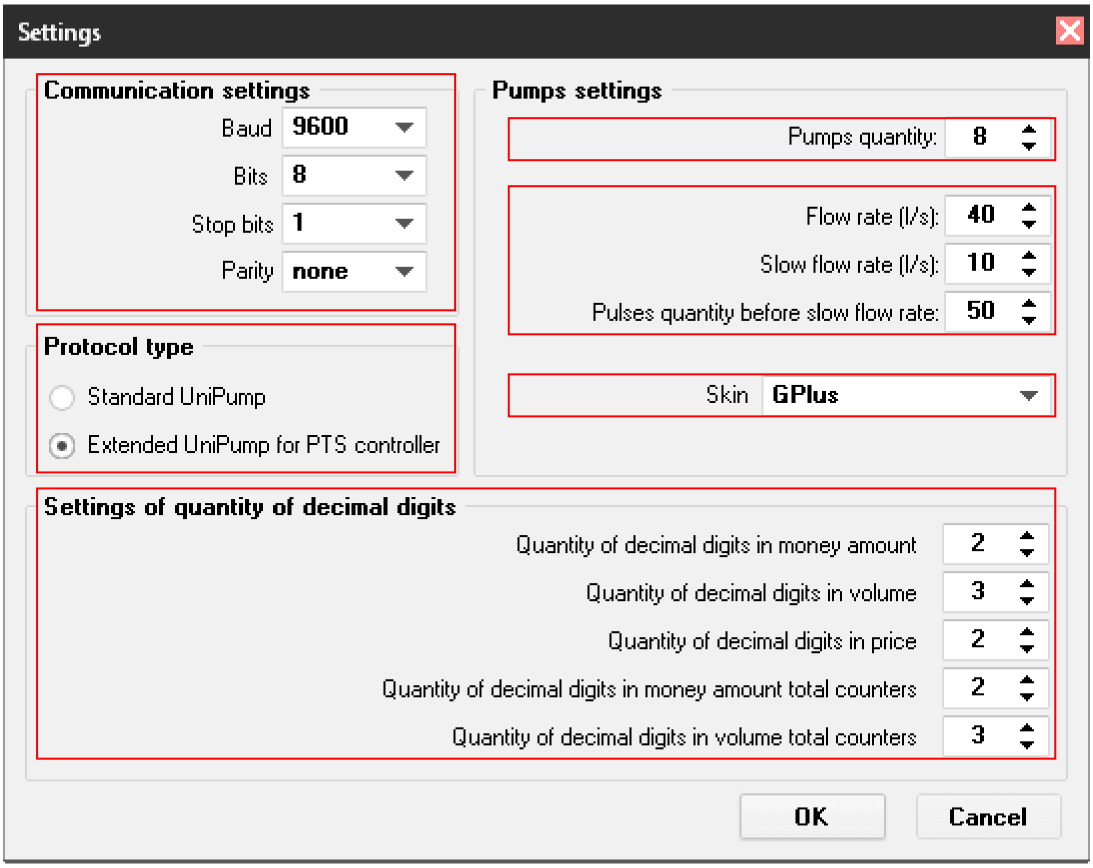
*Volume preset menu for pump*

## *Configuration*

Configuration of the pumps simulator includes configuration of:

- communication settings, which include setting of baud rate, number of data bits and stop bits, parity control
- quantity of pumps (physical address is set on each pump control independently)
- technical characteristics of the pumps in simulator (flow rate, slow flow rate (when flowdown valve is on), quantity of pulses before the flowdown valve is switched on

- protocol type: selection between standard UniPump communication protocol and extended UniPump communicatin protocol for PTS controller
- setting quantity of digits in values of volume, money amount, price and total counters
- selection of graphical skin



Settings

Communication settings

Baud 9600

Bits 8

Stop bits 1

Parity none

Protocol type

○ Standard UniPump

● Extended UniPump for PTS controller

Pumps settings

Pumps quantity: 8

Flow rate (l/s): 40

Slow flow rate (l/s): 10

Pulses quantity before slow flow rate: 50

Skin GPlus

Settings of quantity of decimal digits

Quantity of decimal digits in money amount 2

Quantity of decimal digits in volume 3

Quantity of decimal digits in price 2

Quantity of decimal digits in money amount total counters 2

Quantity of decimal digits in volume total counters 3

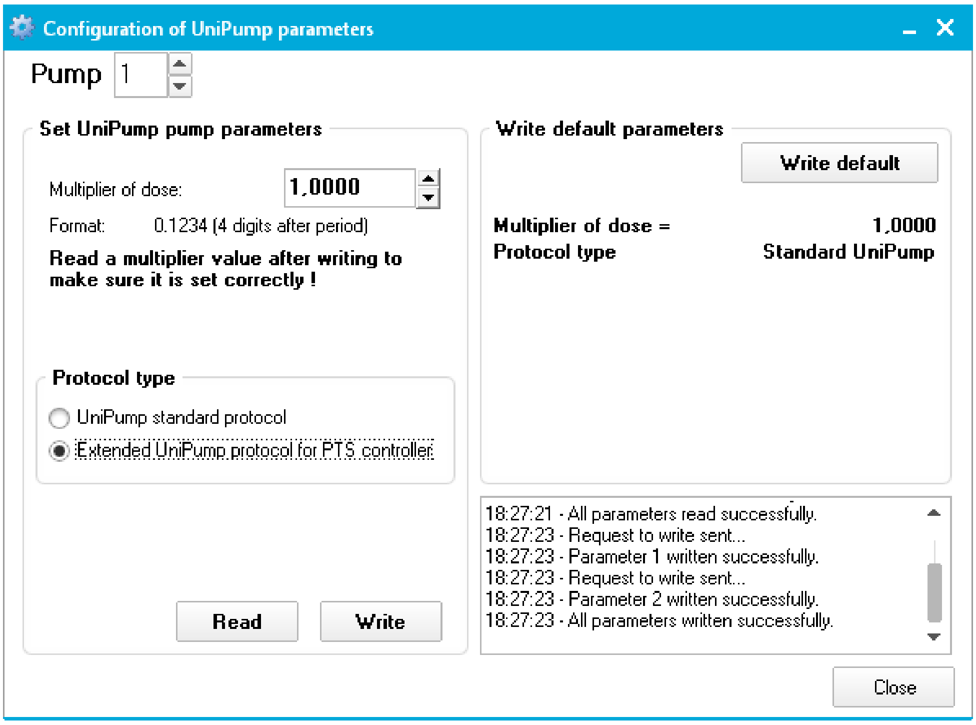
*Settings of pumps simulator*

Pumps simulator can simulate presence at the pumps – take up nozzles and make dispensing in random way thus making it possible to debug control system software as if it was operating on real petrol station.

Log window of the pumps simulator enables to see all the requests and responses in communication and their respective data.

For communication of PTS controller with the UniPump pumps software simulator it is necessary to select in PTS controller configuration of the pump channel communication protocol “2. UNIPUMP” and select appropriate baud rate (equal to set in simulator (9600 baud rate by defaults) and set pumps physical addresses equal to those, which are set for pumps in simulator. Also it is necessary to set parameters to UniPump protocol for each of the pumps in PTS controller configuration (see image below):

1. Parameter “Multiplier of dose” set equal to 1.
2. Parameter “Protocol type” set equal to “UniPump standard protocol” in case if protocol type is set to “Standard UniPump” in pumps simulator or “Extended protocol for PTS controller” in case if protocol type is set to “Extended UniPump for PTS controller” in pumps simulator.



Configuration of UniPump parameters

Pump 1

Set UniPump pump parameters

Multiplier of dose: 1,0000

Format: 0.1234 (4 digits after period)

Read a multiplier value after writing to make sure it is set correctly !

Protocol type

○ UniPump standard protocol

● Extended UniPump protocol for PTS controller

Write default parameters

Multiplier of dose = 1,0000

Protocol type Standard UniPump

```
18:27:21 - All parameters read successfully.
18:27:23 - Request to write sent...
18:27:23 - Parameter 1 written successfully.
18:27:23 - Request to write sent...
18:27:23 - Parameter 2 written successfully.
18:27:23 - All parameters written successfully.
```

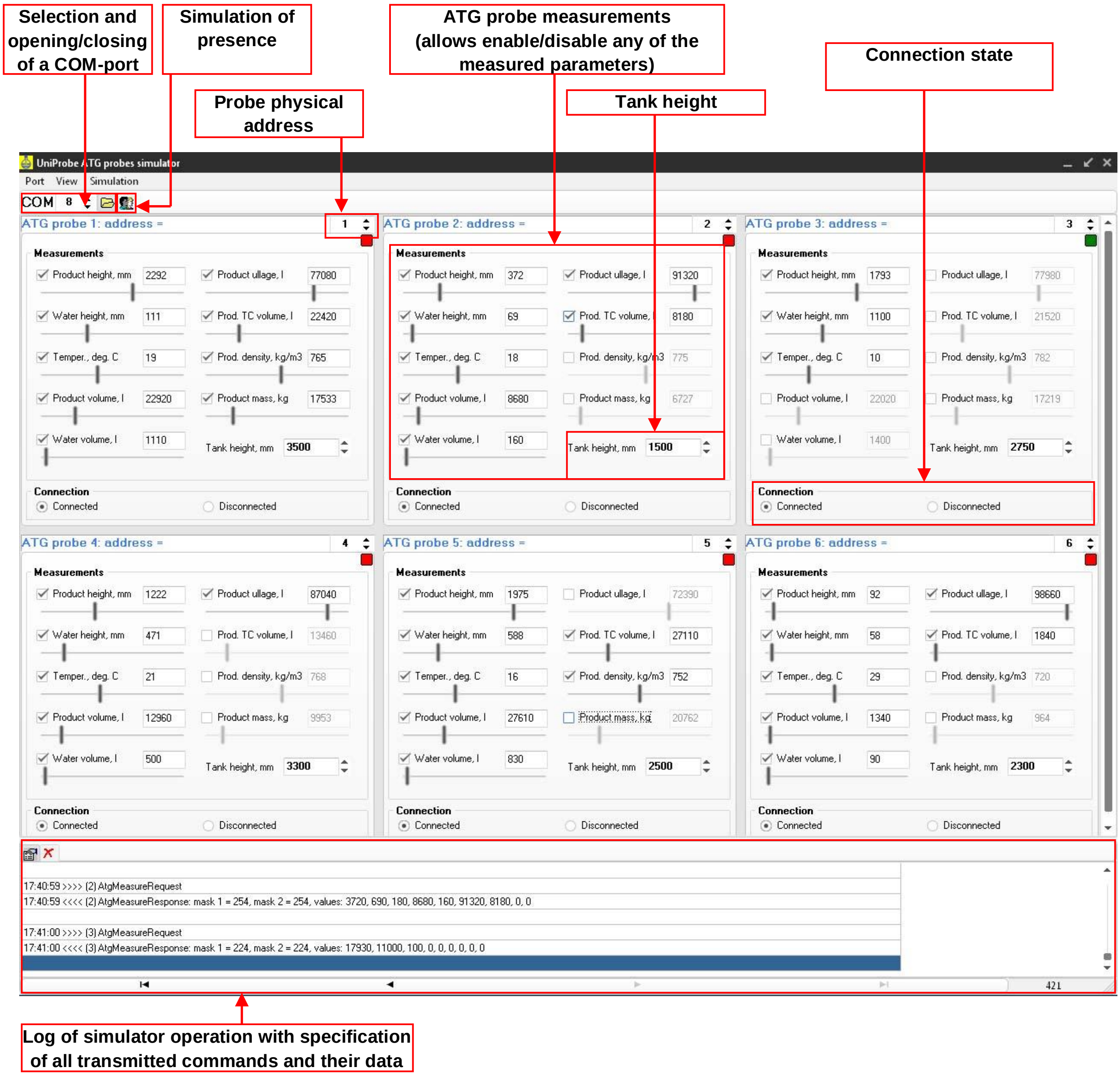
*Parameters for UniPump protocol in PTS controller*

# *UniProbe ATG probes software simulator*

## *Purpose*

PTS controller SDK includes a ATG probes software simulator with a purpose to enable debugging of PTS controller implementation in third party software. Purpose of software simulators is to debug operation of control system software (cash register, POS system, OPT, etc) over ATG systems through PTS controller without a necessity to connect to real ATG systems or probes.

## *Main view*



Selection and opening/closing of a COM-port

Simulation of presence

ATG probe measurements (allows enable/disable any of the measured parameters)

Connection state

Probe physical address

Tank height

Log of simulator operation with specification of all transmitted commands and their data

ATG probe 1: address = 1
Measurements
☑ Product height, mm 2292 ☑ Product ullage, l 77080
☑ Water height, mm 111 ☑ Prod. TC volume, l 22420
☑ Temper., deg. C 19 ☑ Prod. density, kg/m3 765
☑ Product volume, l 22920 ☑ Product mass, kg 17533
☑ Water volume, l 1110 Tank height, mm 3500
Connection
● Connected ○ Disconnected

ATG probe 2: address = 2
Measurements
☑ Product height, mm 372 ☑ Product ullage, l 91320
☑ Water height, mm 69 ☑ Prod. TC volume, l 8180
☑ Temper., deg. C 18 ☐ Prod. density, kg/m3 775
☑ Product volume, l 8680 ☐ Product mass, kg 6727
☑ Water volume, l 160 Tank height, mm 1500
Connection
● Connected ○ Disconnected

ATG probe 3: address = 3
Measurements
☑ Product height, mm 1793 ☐ Product ullage, l 77980
☑ Water height, mm 1100 ☐ Prod. TC volume, l 21520
☑ Temper., deg. C 10 ☐ Prod. density, kg/m3 782
☐ Product volume, l 22020 ☐ Product mass, kg 17219
☐ Water volume, l 1400 Tank height, mm 2750
Connection
● Connected ○ Disconnected

ATG probe 4: address = 4
Measurements
☑ Product height, mm 1222 ☑ Product ullage, l 87040
☑ Water height, mm 471 ☐ Prod. TC volume, l 13460
☑ Temper., deg. C 21 ☐ Prod. density, kg/m3 768
☑ Product volume, l 12960 ☐ Product mass, kg 9953
☑ Water volume, l 500 Tank height, mm 3300
Connection
● Connected ○ Disconnected

ATG probe 5: address = 5
Measurements
☑ Product height, mm 1975 ☐ Product ullage, l 72390
☑ Water height, mm 588 ☑ Prod. TC volume, l 27110
☑ Temper., deg. C 16 ☑ Prod. density, kg/m3 752
☑ Product volume, l 27610 ☐ Product mass, kg 20762
☑ Water volume, l 830 Tank height, mm 2500
Connection
● Connected ○ Disconnected

ATG probe 6: address = 6
Measurements
☑ Product height, mm 92 ☑ Product ullage, l 98660
☑ Water height, mm 58 ☑ Prod. TC volume, l 1840
☑ Temper., deg. C 29 ☐ Prod. density, kg/m3 720
☑ Product volume, l 1340 ☐ Product mass, kg 964
☑ Water volume, l 90 Tank height, mm 2300
Connection
● Connected ○ Disconnected

```
17:40:59 >>>> (2) AtgMeasureRequest
17:40:59 <<<< (2) AtgMeasureResponse: mask 1 = 254, mask 2 = 254, values: 3720, 690, 180, 8680, 160, 91320, 8180, 0, 0

17:41:00 >>>> (3) AtgMeasureRequest
17:41:00 <<<< (3) AtgMeasureResponse: mask 1 = 224, mask 2 = 224, values: 17930, 11000, 100, 0, 0, 0, 0, 0, 0
```

*ATG probes software simulator main view*

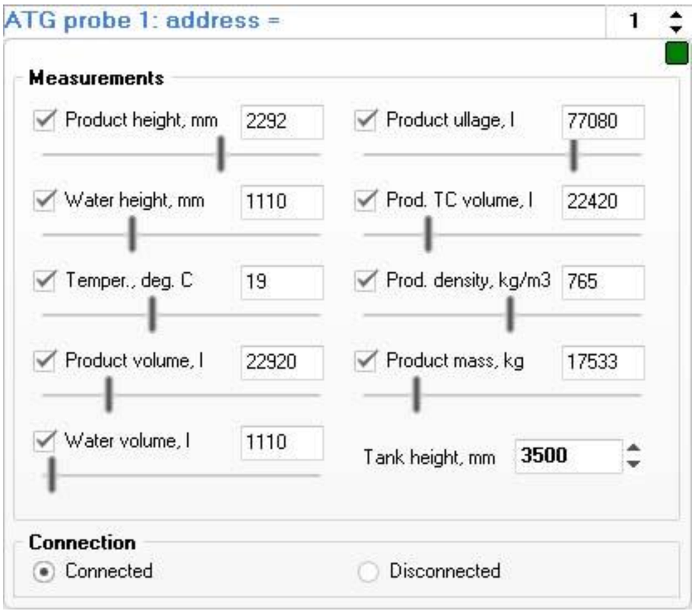
In ATG probes simulator it is possible to set measurements of the following parameters:

- Products height
- Water height
- Temperature
- Product volume

- Water volume
- Product ullage
- Product temperature compensated volume
- Product density
- Product mass

Any of the specified parameters can be switched on or off.
Also it is possible to set height of tank in the system.

Each of the ATG probes can be transferred to connected (responses to PTS controller) or disconnected (does not response to PTS controller) state.



ATG probe 1: address = 1

**Measurements**

☑ Product height, mm 2292
☑ Product ullage, l 77080
☑ Water height, mm 1110
☑ Prod. TC volume, l 22420
☑ Temper., deg. C 19
☑ Prod. density, kg/m3 765
☑ Product volume, l 22920
☑ Product mass, kg 17533
☑ Water volume, l 1110
Tank height, mm 3500

**Connection**

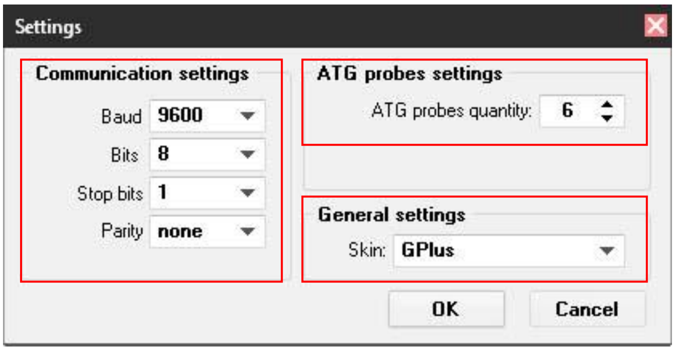
● Connected ○ Disconnected

*Probe control in ATG probes software simulator*

## Configuration

Configuration of the pumps simulator includes configuration of:

- communication settings, which include setting of baud rate, number of data bits and stop bits, parity control
- quantity of ATG probes (physical address is set on each probe control independently)
- selection of graphical skin



Settings

**Communication settings**

Baud 9600
Bits 8
Stop bits 1
Parity none

**ATG probes settings**

ATG probes quantity: 6

**General settings**

Skin: GPlus

*Settings of probes simulator*

Probes simulator can simulate presence of liquids in tanks – change height and volume of fuel, temperature mass in random way thus making it possible to debug control system software as if it was operating on real petrol station.

Log window of the probes simulator enables to see all the requests and responses in communication and their respective data.

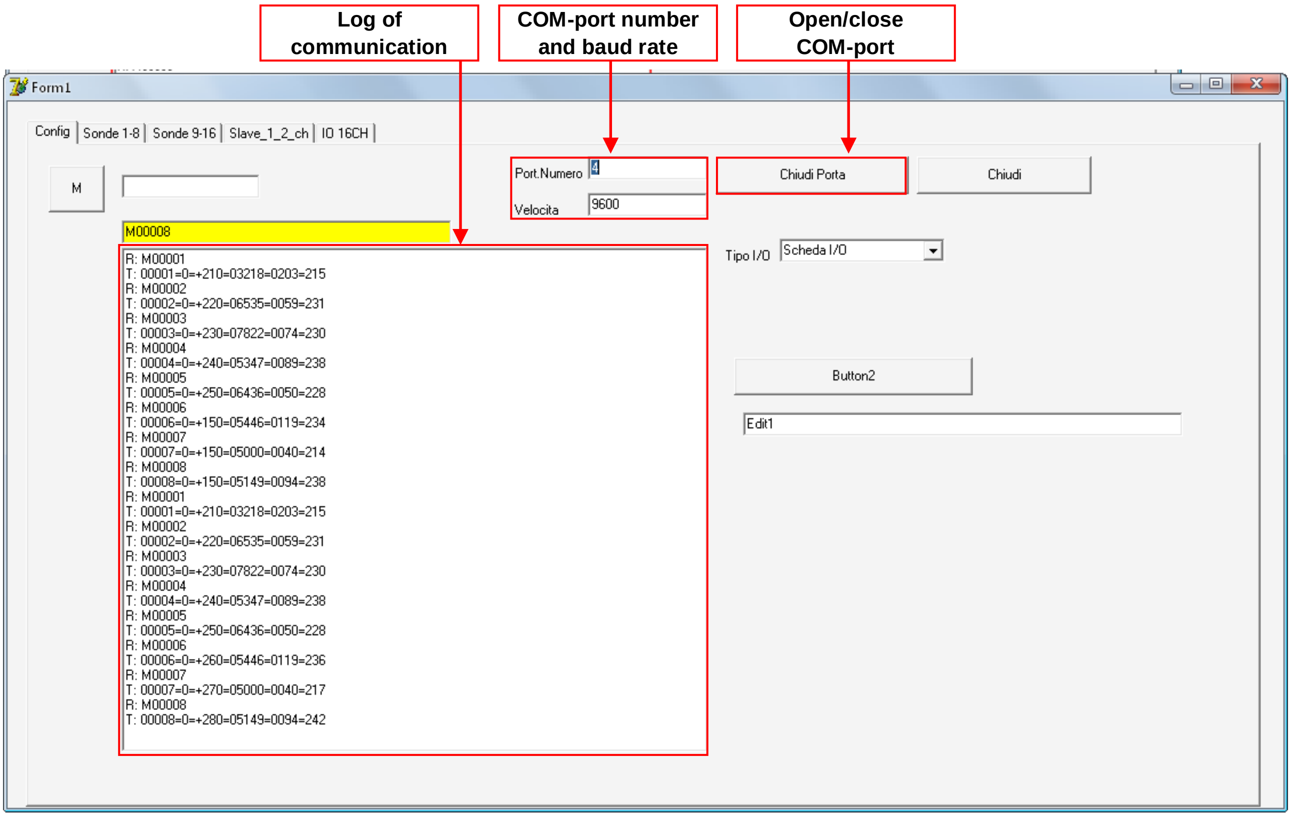
For communication of the PTS controller with UniProbe ATG probes software simulator it is necessary to select in PTS controller configuration of the ATG channel communication protocol "9. UNIPROBE" and select appropriate baud rate (equal to set in simulator (9600 baud rate by defaults) and set probes physical addresses equal to those, which are set for probes in simulator. Also it is necessary to set parameters to UniPump protocol for each of the pumps in PTS controller configuration (see image below):

3. Parameter "Multiplier of dose" set equal to 1.
4. Parameter "Protocol type" set equal to "UniPump standard protocol" in case if protocol type is set to "Standard UniPump" in pumps simulator or "Extended protocol for PTS controller" in case if protocol type is set to "Extended UniPump for PTS controller" in pumps simulator.

# *Start Italiana ATG probes software simulator*

## *Purpose*

PTS controller SDK include ATG probes software simulator of Start Italiana. Purpose of the software simulator is to enable debugging of PTS controller and ATG probe at implementation of PTS controller in third party software. Purpose of software simulators is to debug operation of control system software (cash register, POS system, OPT, etc) over ATG systems through PTS controller without a necessity to connect to ATG system or probes.


Log of communication

COM-port number and baud rate

Open/close COM-port

M00008

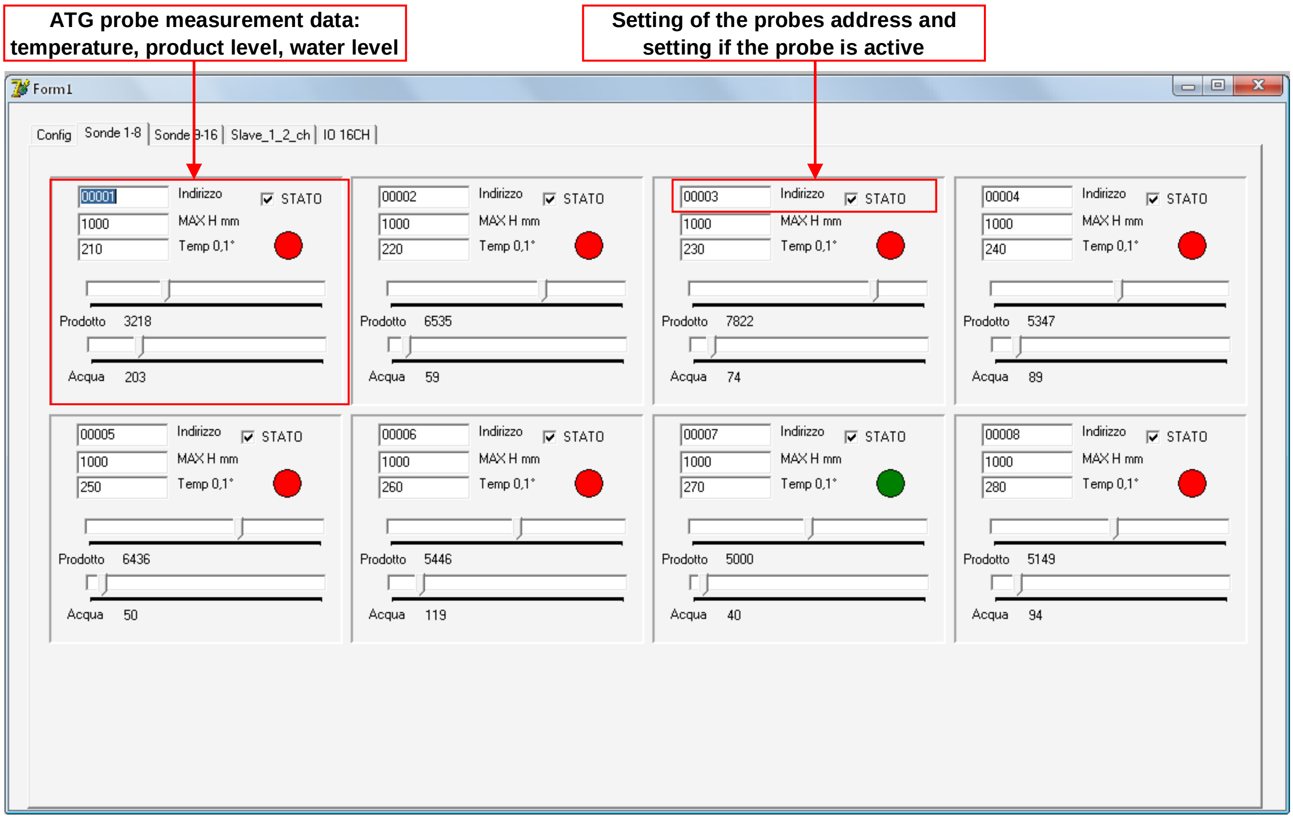
```
R: M00001
T: 00001=0=+210=03218=0203=215
R: M00002
T: 00002=0=+220=06535=0059=231
R: M00003
T: 00003=0=+230=07822=0074=230
R: M00004
T: 00004=0=+240=05347=0089=238
R: M00005
T: 00005=0=+250=06436=0050=228
R: M00006
T: 00006=0=+150=05446=0119=234
R: M00007
T: 00007=0=+150=05000=0040=214
R: M00008
T: 00008=0=+150=05149=0094=238
R: M00001
T: 00001=0=+210=03218=0203=215
R: M00002
T: 00002=0=+220=06535=0059=231
R: M00003
T: 00003=0=+230=07822=0074=230
R: M00004
T: 00004=0=+240=05347=0089=238
R: M00005
T: 00005=0=+250=06436=0050=228
R: M00006
T: 00006=0=+260=05446=0119=236
R: M00007
T: 00007=0=+270=05000=0040=217
R: M00008
T: 00008=0=+280=05149=0094=242
```

Port.Numero 4

Velocita 9600

Tipo I/O Scheda I/O

Edit1


ATG probe measurement data: temperature, product level, water level

Setting of the probes address and setting if the probe is active

| Indirizzo | STATO | MAX H mm | Temp 0,1° | | Prodotto | Acqua |
|---|---|---|---|---|---|---|
| 00001 | ☑ | 1000 | 210 | 🔴 | 3218 | 203 |
| 00002 | ☑ | 1000 | 220 | 🔴 | 6535 | 59 |
| 00003 | ☑ | 1000 | 230 | 🔴 | 7822 | 74 |
| 00004 | ☑ | 1000 | 240 | 🔴 | 5347 | 89 |
| 00005 | ☑ | 1000 | 250 | 🔴 | 6436 | 50 |
| 00006 | ☑ | 1000 | 260 | 🔴 | 5446 | 119 |
| 00007 | ☑ | 1000 | 270 | 🟢 | 5000 | 40 |
| 00008 | ☑ | 1000 | 280 | 🔴 | 5149 | 94 |

# *Step-by-step configuration of the PTS controller SDK*

## *Purpose*

In order to simplify understanding of PTS controller SDK operation and assembling this step-by-step instruction is provided. It describes basic steps to be made with PTS controller SDK to assemble it correctly and also to install and configure software coming in its structure.

## *Step 1. Downloading of PTS controller SDK software*

Using a link for download of PTS controller SDK software it is necessary to get all files from it. The files should include:

- API tools:
    - Open source application programming interfaces, libraries, test utilities, applications and documentation for development the third party software for communication with PTS controller using various programming languages and environments
    - UniPump communication protocol description for PTS controller
- Software tools
    - NaftaPOS software for petrol stations
    - PumpDemo util
    - PTS controller configuration tool *Pts_conf.exe*
    - PTS terminal utility
    - USB dongle drivers
    - Latest stable PTS controller firmware
- Software simulators
    - Fuel dispensers software simulators
    - ATG systems (probes) software simulators
- Documentation
    - User manuals
    - Technical guides

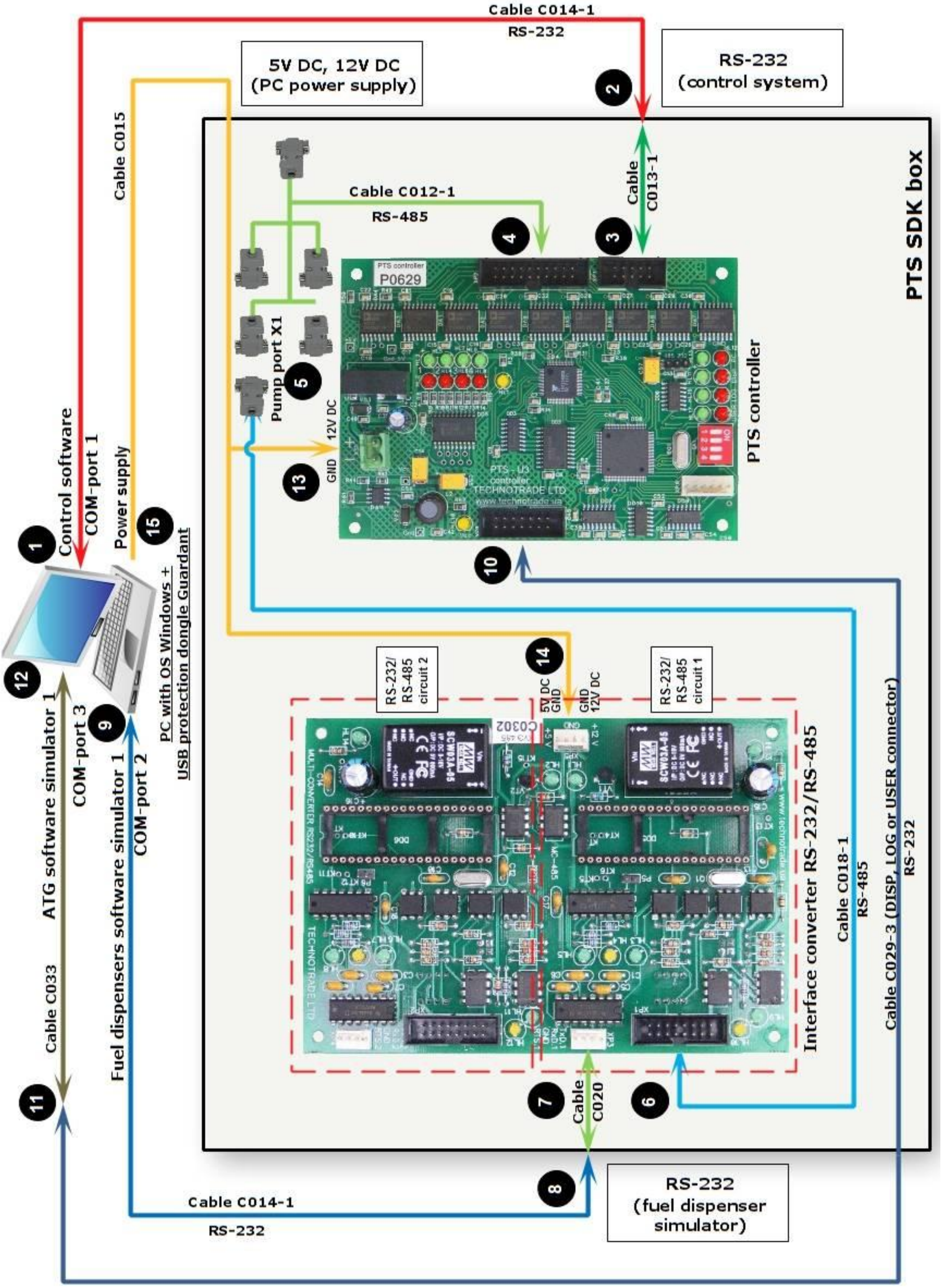
## *Step 2. Assembling PTS controller SDK cabling and connection to PC*

For connection of PTS controller SDK cabling it is necessary to have at least 2 COM-ports available in PC (one COM-port – for control system, other COM-port – for fuel dispenser software simulator or ATG probe software simulator). In case of absence of native COM-ports in the PC motherboard it is possible to use any standard interface converters to RS-232 interface: USB-to-COM, Ethernet-to-COM, PCI-to-COM, etc.

*USB-to-COM interface converter cable*

PTS controller SDK is generally supplied in a plastic box, where the PTS controller and RS-232/RS-485 interface converter boards are mounted in the box. Cabling in order to avoid damage of the boards during transportation is supplied separately. So, it is necessary to make all the cablings connections correctly for the system to start operation.

In this step-by-step instruction we will connect 1 pump simulator and 1 ATG probes simulator. Scheme of connections should be the following:



## *Step 3. Connection of cabling*

Refer to the scheme of connections above for connection of cabling:

1. Cabling for communication of PC with PTS controller:
   - Connect cable C014-1 from one side to one of the COM-ports of PC (where the control software is to be launched) and from the other – to connector “RS-232 (control system)”, which is located on the box of PTS SDK (see connections 1 and 2 on the scheme above). This connector “RS-232 (control system)” is a connector of cable C013-1, mounted inside the PTS SDK box.
   - Make sure that the cable C013-1 is mounted inside the PTS SDK box and is connected from one side to cable C014-1 and from the other – to PC port of PTS controller (connections 2 and 3).
2. Cabling for connection to pump software simulator:
   - Connect cable C012-1 to the pump port of PTS controller (connection 4).
   - Connect connector X1 of cable C012-1 to the one side of cable C018-1 (connection 5).
   - Connect the other side of cable C018-1 to XP1 connector (RS-485 channel 1) of RS-232/RS-485 interface converter board (connection 6).
   - Make sure that cable C020 is mounted inside the PTS SDK box to connector “RS-232 (fuel dispenser simulator)” with one of its sides (connection 8).
   - Connect the other side of cable C020 to connector XP3 (RS-232 channel 1) of RS-232/RS-485 interface converter board (connection 7).
   - Using a cable C014-1 link the connector “RS-232 (fuel dispenser simulator)”, located on PTS SDK box, with a COM-port of the PC, where pumps software simulator is to be launched (connections 8 and 9).
3. Cabling for connection to ATG software simulator (optional, may be skipped):
   - Connect cable C029-3 to ATG port of PTS controller (connection 10).
   - Connect connector USER of cable C029-3 to one side of cable C033 (connection 11).
   - Connect the other side of cable C033 to COM-port of the PC, where ATG software simulator is to be launched (connection 12).
4. Cabling for power supply of PTS controller and RS-232/RS-485 interface converter:
   - Connect a cable C015-1, which is mounted inside the PTS SDK box, to PTS controller board (voltage 12 V DC) and to RS-232/RS-485 interface converter board (voltages 5 V DC, 12 V DC) (connections 13 and 14).
   - Connect external side of cable C015-1 to PC power supply or to any other power supply source having 2 voltages: 5 V DC and 12 V DC (connection 15).

So, PTS controller is to be connected to a COM-port of the PC using cable C013-1 (cable C013-1 comes mounted to the PTS SDK box with inscription “RS-232 (control system)”). Connection can be made through a cable C014-1, which serves as a prolonger of the COM-port.

RS-232/RS-485 interface converter board serves for connecting of PTS controller pump channels (in RS-485 interface) with COM-port of PC (which is in RS-232 interface), on which pumps software simulator is to be launched. RS-232/RS-485 interface converter board has 2 independent conversion circuits. Input cable is C018-1 (RS-485 interface), output cable is C020 (RS-232 interface).

Due to a reason that in this description we will configure only channel 1 of PTS controller to communicate with fuel dispensers – it is necessary to connect the connector X1 of cable C012-1 coming from the pump port of the PTS controller to a cable C018-1 of the RS-232/RS-485 interface converter board. Output cable C020 of the same conversion circuit of the RS-232/RS-485 interface converter (one of cables C020 comes mounted to the PTS SDK box with inscription "RS-232 (fuel dispenser simulator)") is to be connected to the COM-port of PC, on which pumps software simulator is launched. Connection can be made through a cable C014-1, which serves as a prolonger of the COM-port.

In order to connect ATG systems (probes) software simulators it is necessary to connect one of the connectors of cable C029-3 through a cable C033 to the COM-port of PC, where ATG system (probes) software simulator is launched. Connection can be made through the cable C014-1, which serves as a prolonger of the COM-port. In this description we will use USER-port of PTS controller to connect ATG probes software simulator, so it is necessary to connect USER connector of cable C029-3 to cable C033 (directly or through cable C014-1). C033 cable should be connected to COM-port of the PC.

It is worth to mention that only 1 application can work with a specific COM-port at the time, 2 applications can not share the same COM-port.

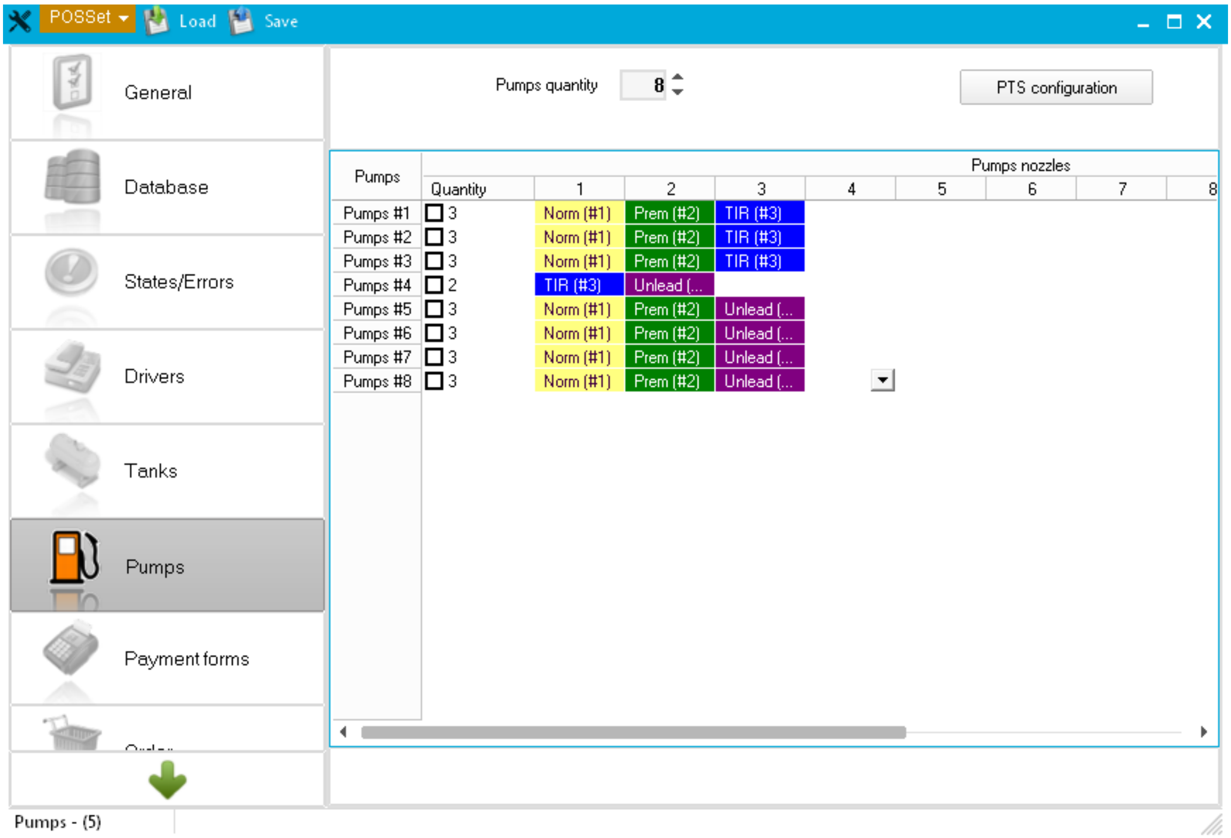
## *Step 4. Installation of NaftaPOS software for petrol stations*

Using instructions given in point "NaftaPOS software for petrol stations. Installation." of given technical guide it is necessary to download and install NaftaPOS software. During installation of NaftaPOS software Guardant USB dongle drivers are to be installed.

## *Step 5. Configuration of NaftaPOS software to work with PTS controller for communication with pumps and ATG systems*

After the cabling is duly connected and NaftaPOS is installed run POSSet configuration utility of NaftaPOS software (*POSSet.exe*, located in root folder of NaftaPOS software).

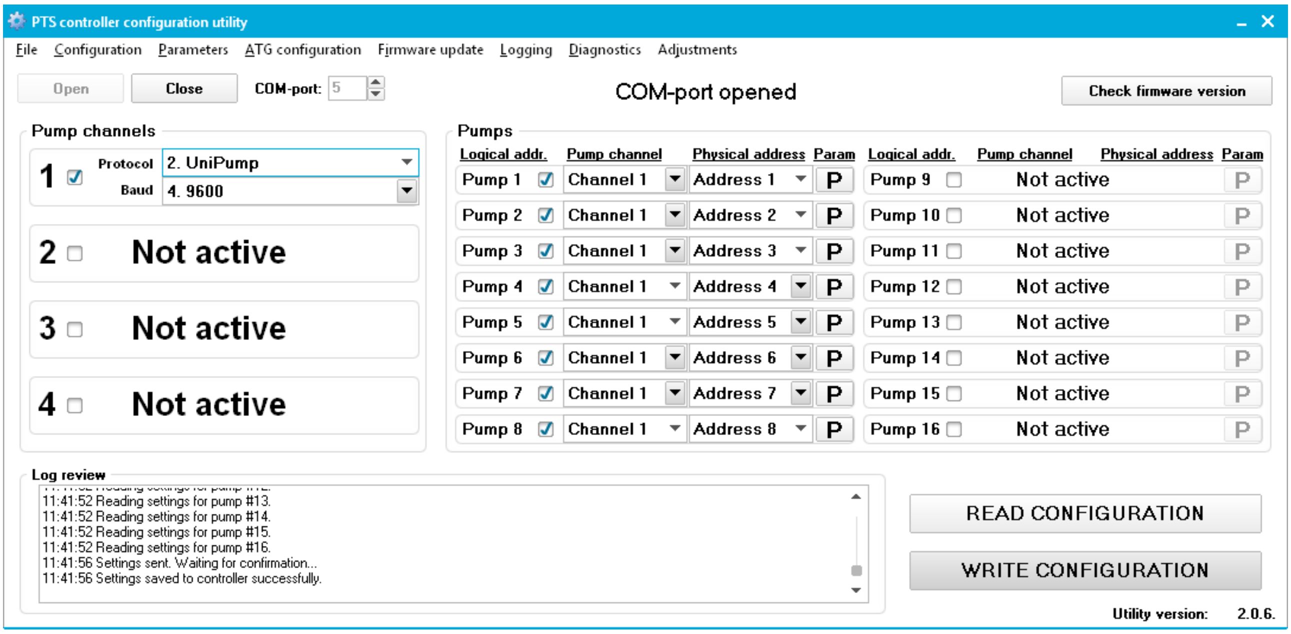
Go to tab "Pumps" and set there quantity to pumps equal to 8:



Pumps quantity 8

| Pumps | | Pumps nozzles | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Quantity | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Pumps #1 | □ 3 | Norm (#1) | Prem (#2) | TIR (#3) | | | | | |
| Pumps #2 | □ 3 | Norm (#1) | Prem (#2) | TIR (#3) | | | | | |
| Pumps #3 | □ 3 | Norm (#1) | Prem (#2) | TIR (#3) | | | | | |
| Pumps #4 | □ 2 | TIR (#3) | Unlead (... | | | | | | |
| Pumps #5 | □ 3 | Norm (#1) | Prem (#2) | Unlead (... | | | | | |
| Pumps #6 | □ 3 | Norm (#1) | Prem (#2) | Unlead (... | | | | | |
| Pumps #7 | □ 3 | Norm (#1) | Prem (#2) | Unlead (... | | | | | |
| Pumps #8 | □ 3 | Norm (#1) | Prem (#2) | Unlead (... | ▾ | | | | |

On tab “Pumps” click on a button “PTS configuration” and set “2. UniPump” protocol and “4. 9600” baud rate for pump channel 1. Adjust pumps 1 – 8 for pump channel 1 and write configuration.


PTS controller configuration utility
File Configuration Parameters ATG configuration Firmware update Logging Diagnostics Adjustments
Open
Close
Check firmware version
READ CONFIGURATION
WRITE CONFIGURATION

COM-port: 5

COM-port opened

Pump channels

1 ☑ Protocol 2. UniPump
Baud 4. 9600

2 □ Not active

3 □ Not active

4 □ Not active

Pumps

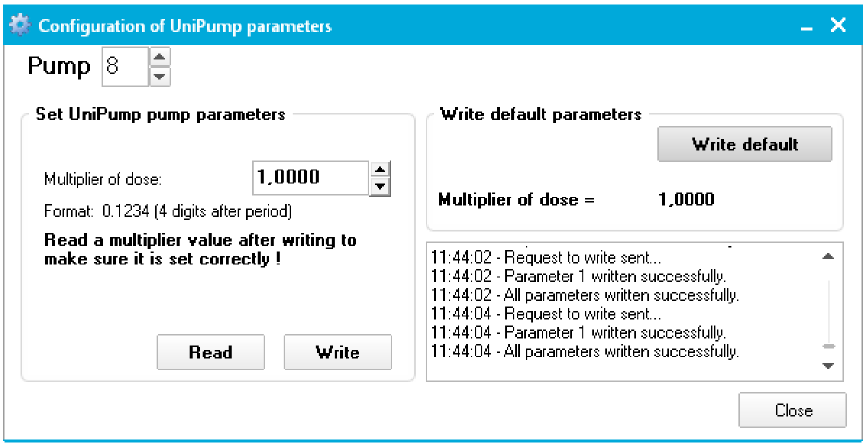
| Logical addr. | Pump channel | Physical address | Param | Logical addr. | Pump channel | Physical address | Param |
|---|---|---|---|---|---|---|---|
| Pump 1 ☑ | Channel 1 | Address 1 | P | Pump 9 □ | Not active | | P |
| Pump 2 ☑ | Channel 1 | Address 2 | P | Pump 10 □ | Not active | | P |
| Pump 3 ☑ | Channel 1 | Address 3 | P | Pump 11 □ | Not active | | P |
| Pump 4 ☑ | Channel 1 | Address 4 | P | Pump 12 □ | Not active | | P |
| Pump 5 ☑ | Channel 1 | Address 5 | P | Pump 13 □ | Not active | | P |
| Pump 6 ☑ | Channel 1 | Address 6 | P | Pump 14 □ | Not active | | P |
| Pump 7 ☑ | Channel 1 | Address 7 | P | Pump 15 □ | Not active | | P |
| Pump 8 ☑ | Channel 1 | Address 8 | P | Pump 16 □ | Not active | | P |

Log review

```
11:41:52 Reading settings for pump #13.
11:41:52 Reading settings for pump #14.
11:41:52 Reading settings for pump #15.
11:41:52 Reading settings for pump #16.
11:41:56 Settings sent. Waiting for confirmation...
11:41:56 Settings saved to controller successfully.
```

Utility version: 2.0.6.

As parameters for UniPump protocol set default parameters for each of the enabled pump:



Pump 8

Set UniPump pump parameters

Multiplier of dose: 1,0000

Format: 0.1234 (4 digits after period)

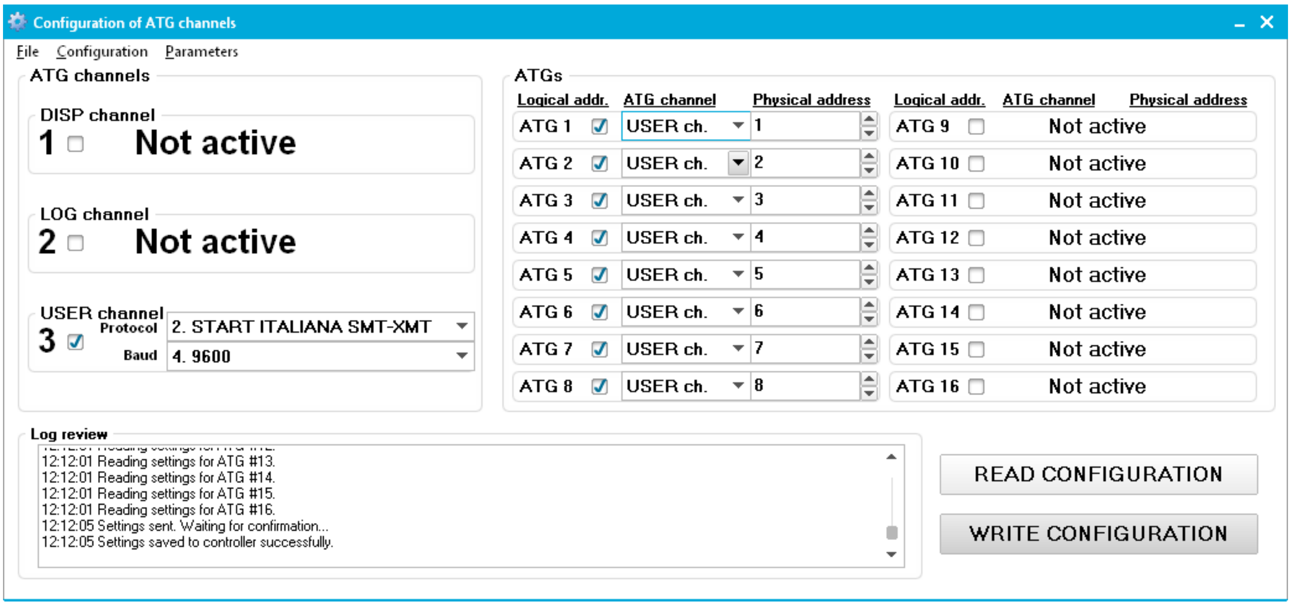
Read a multiplier value after writing to make sure it is set correctly !

Write default parameters

Multiplier of dose = 1,0000

```
11:44:02 - Request to write sent...
11:44:02 - Parameter 1 written successfully.
11:44:02 - All parameters written successfully.
11:44:04 - Request to write sent...
11:44:04 - Parameter 1 written successfully.
11:44:04 - All parameters written successfully.
```

In configuration of ATG channels set “2. START_ITALIANA SMT-XMT” protocol and “4. 9600” baud rate for USER ATG channel. Adjust ATGs 1 – 8 for USER pump channel and write configuration.



ATG channels

DISP channel
1 □ Not active

LOG channel
2 □ Not active

USER channel
3 ☑ Protocol 2. START ITALIANA SMT-XMT
Baud 4. 9600

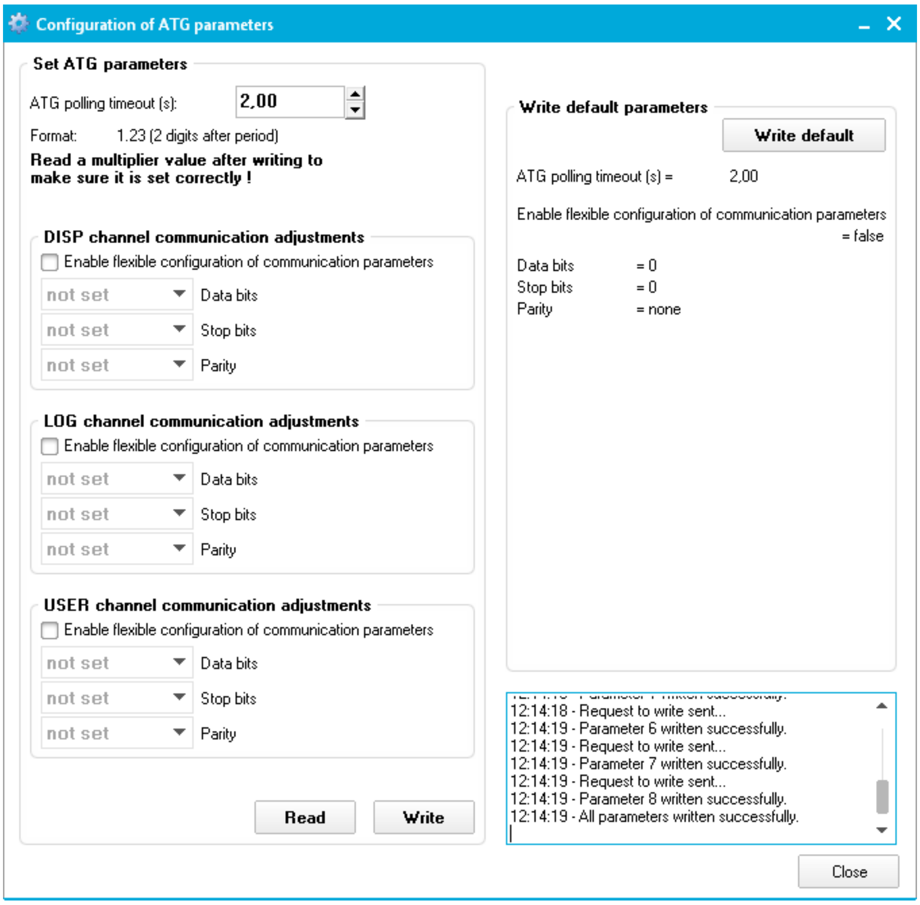
ATGs

| Logical addr. | ATG channel | Physical address | Logical addr. | ATG channel | Physical address |
|---|---|---|---|---|---|
| ATG 1 ☑ | USER ch. | 1 | ATG 9 □ | Not active | |
| ATG 2 ☑ | USER ch. | 2 | ATG 10 □ | Not active | |
| ATG 3 ☑ | USER ch. | 3 | ATG 11 □ | Not active | |
| ATG 4 ☑ | USER ch. | 4 | ATG 12 □ | Not active | |
| ATG 5 ☑ | USER ch. | 5 | ATG 13 □ | Not active | |
| ATG 6 ☑ | USER ch. | 6 | ATG 14 □ | Not active | |
| ATG 7 ☑ | USER ch. | 7 | ATG 15 □ | Not active | |
| ATG 8 ☑ | USER ch. | 8 | ATG 16 □ | Not active | |

Log review

```
12:12:01 Reading settings for ATG #13.
12:12:01 Reading settings for ATG #14.
12:12:01 Reading settings for ATG #15.
12:12:01 Reading settings for ATG #16.
12:12:05 Settings sent. Waiting for confirmation...
12:12:05 Settings saved to controller successfully.
```

Parameters for ATGs set as default:



Configuration of ATG parameters

**Set ATG parameters**

ATG polling timeout (s): 2,00

Format: 1.23 (2 digits after period)

**Read a multiplier value after writing to make sure it is set correctly !**

**DISP channel communication adjustments**

☐ Enable flexible configuration of communication parameters

not set Data bits

not set Stop bits

not set Parity

**LOG channel communication adjustments**

☐ Enable flexible configuration of communication parameters

not set Data bits

not set Stop bits

not set Parity

**USER channel communication adjustments**

☐ Enable flexible configuration of communication parameters

not set Data bits

not set Stop bits

not set Parity

Read Write

**Write default parameters**

Write default

ATG polling timeout (s) = 2,00

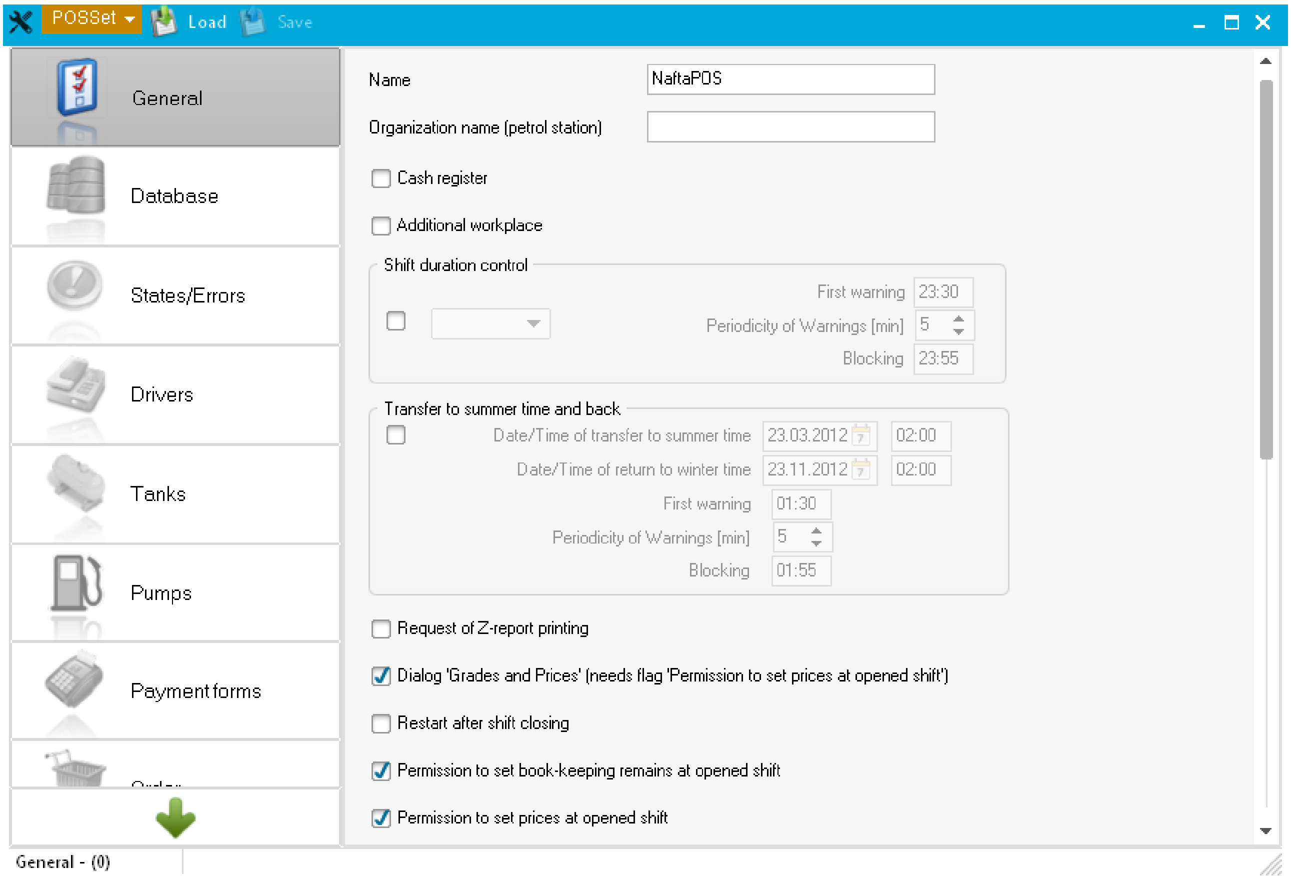
Enable flexible configuration of communication parameters = false

```
Data bits   = 0
Stop bits   = 0
Parity      = none
```

```
12:14:18 - Request to write sent...
12:14:19 - Parameter 6 written successfully.
12:14:19 - Request to write sent...
12:14:19 - Parameter 7 written successfully.
12:14:19 - Request to write sent...
12:14:19 - Parameter 8 written successfully.
12:14:19 - All parameters written successfully.
```

Close

Close PTS controller configuration utility and go to POSSet tab “General”, where set configuration as shown on screen shots:



Name: NaftaPOS

Organization name (petrol station):

☐ Cash register

☐ Additional workplace

Shift duration control

☐ First warning 23:30

Periodicity of Warnings [min] 5

Blocking 23:55

Transfer to summer time and back

☐ Date/Time of transfer to summer time 23.03.2012 02:00

Date/Time of return to winter time 23.11.2012 02:00

First warning 01:30

Periodicity of Warnings [min] 5

Blocking 01:55

☐ Request of Z-report printing

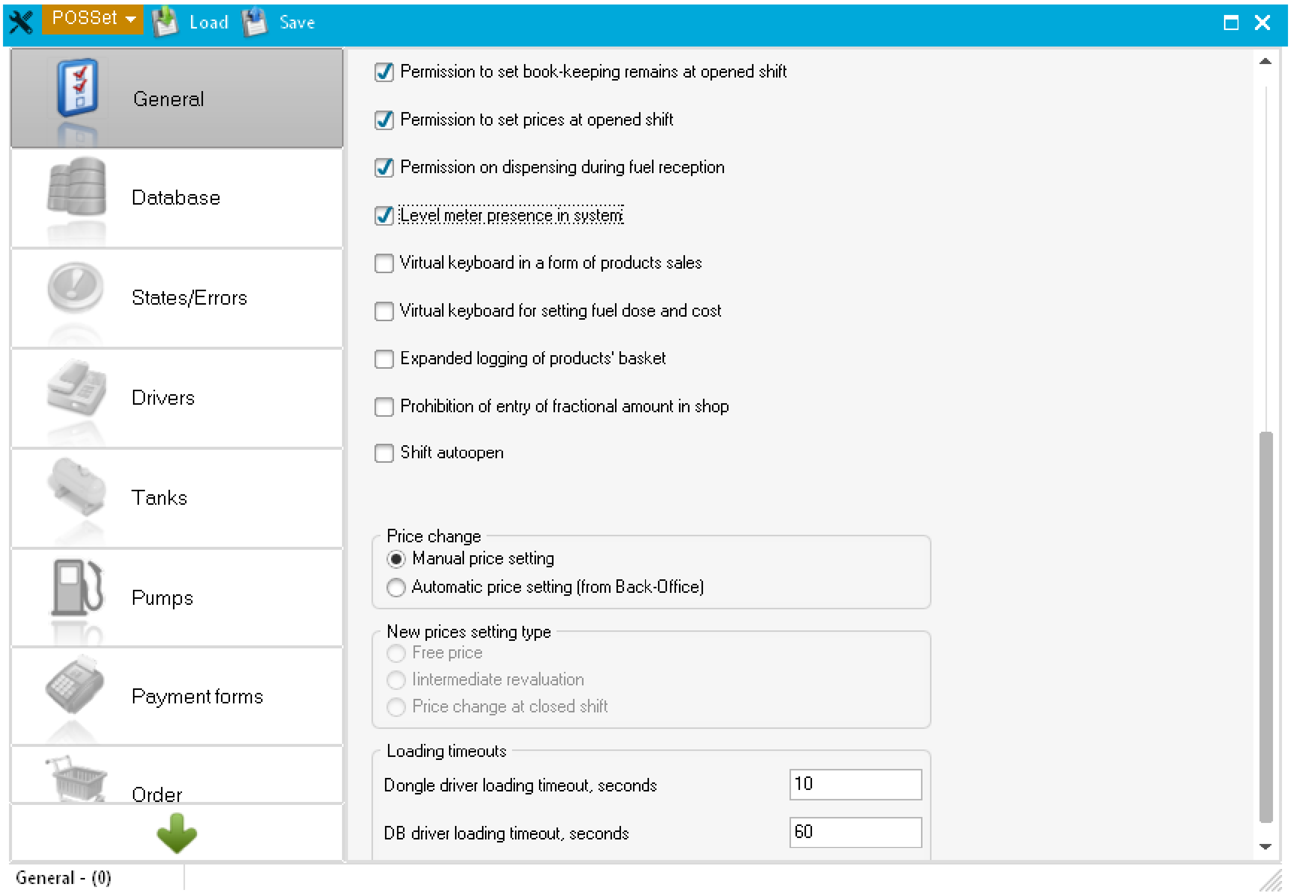
☑ Dialog 'Grades and Prices' (needs flag 'Permission to set prices at opened shift')

☐ Restart after shift closing

☑ Permission to set book-keeping remains at opened shift

☑ Permission to set prices at opened shift

General - (0)



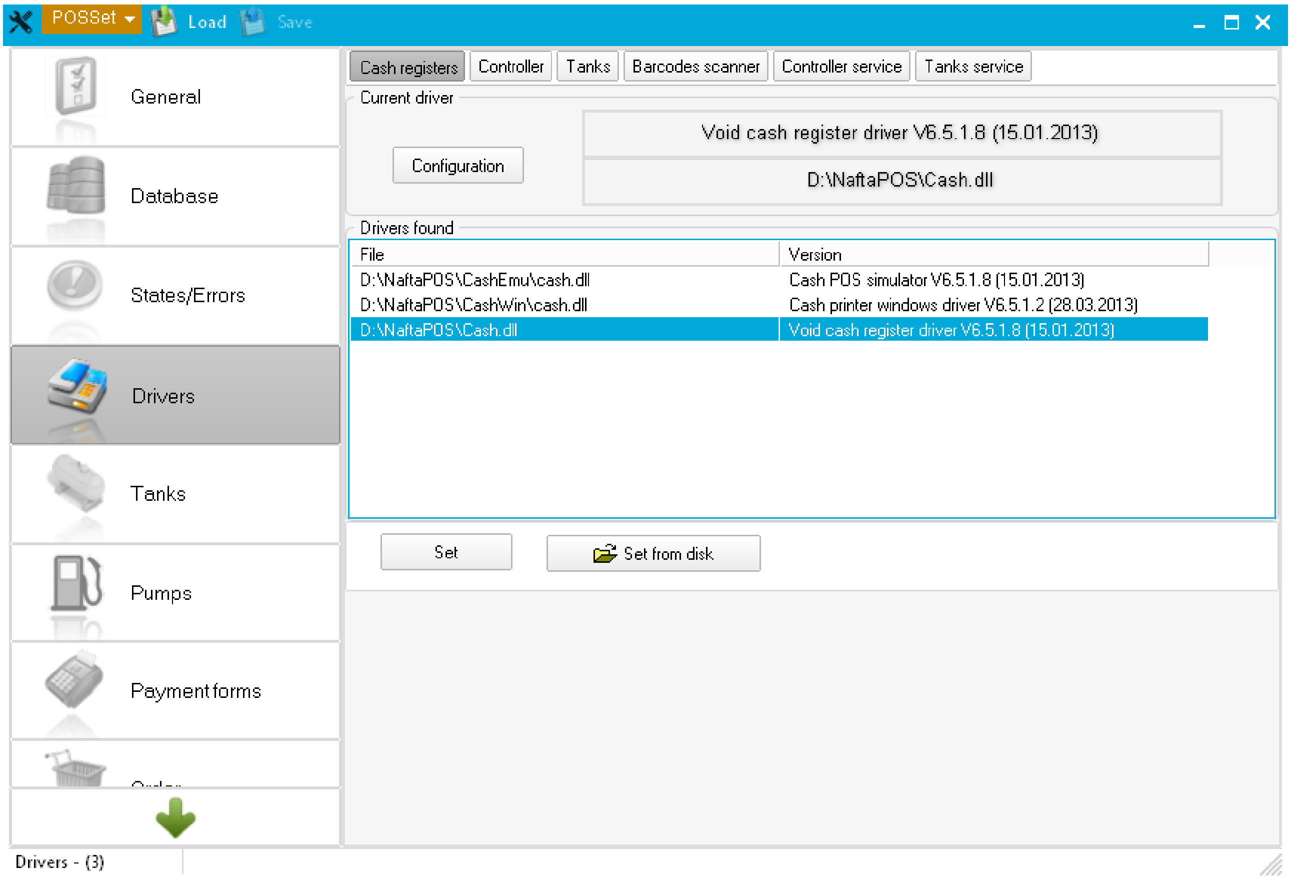
☑ Permission to set book-keeping remains at opened shift
☑ Permission to set prices at opened shift
☑ Permission on dispensing during fuel reception
☑ Level meter presence in system
□ Virtual keyboard in a form of products sales
□ Virtual keyboard for setting fuel dose and cost
□ Expanded logging of products' basket
□ Prohibition of entry of fractional amount in shop
□ Shift autoopen

Price change
● Manual price setting
○ Automatic price setting (from Back-Office)

New prices setting type
○ Free price
○ Iintermediate revaluation
○ Price change at closed shift

Loading timeouts
Dongle driver loading timeout, seconds: 10
DB driver loading timeout, seconds: 60

Go to tab “Drivers” and set for a cash register a driver “Void cash register driver”:

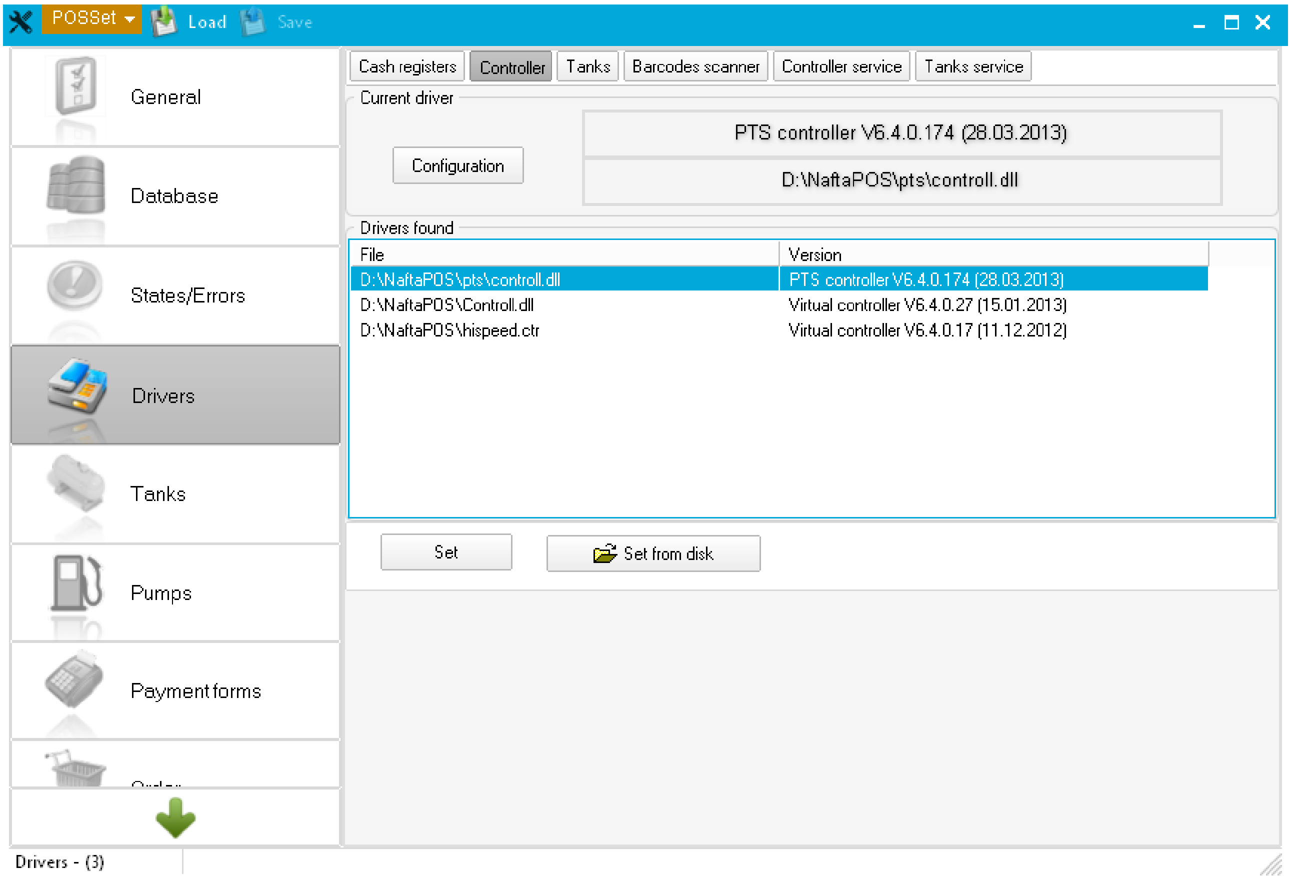


Current driver
Void cash register driver V6.5.1.8 (15.01.2013)
D:\NaftaPOS\Cash.dll

Drivers found

| File | Version |
|---|---|
| D:\NaftaPOS\CashEmu\cash.dll | Cash POS simulator V6.5.1.8 (15.01.2013) |
| D:\NaftaPOS\CashWin\cash.dll | Cash printer windows driver V6.5.1.2 (28.03.2013) |
| D:\NaftaPOS\Cash.dll | Void cash register driver V6.5.1.8 (15.01.2013) |

As a driver of “Controller” select a “PTS controller” driver:



Current driver

PTS controller V6.4.0.174 (28.03.2013)

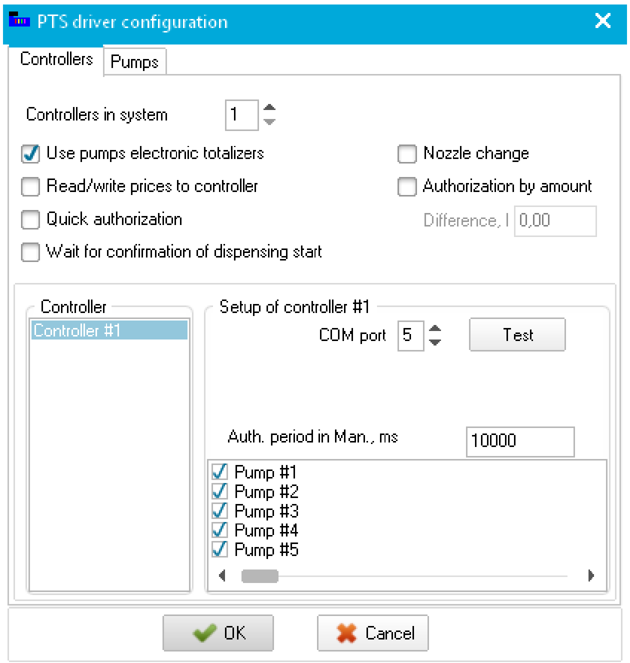
D:\NaftaPOS\pts\controll.dll

Drivers found

| File | Version |
|---|---|
| D:\NaftaPOS\pts\controll.dll | PTS controller V6.4.0.174 (28.03.2013) |
| D:\NaftaPOS\Controll.dll | Virtual controller V6.4.0.27 (15.01.2013) |
| D:\NaftaPOS\hispeed.ctr | Virtual controller V6.4.0.17 (11.12.2012) |

Go to configuration of “PTS controller” driver and set there the following configuration on tab “Controllers”:

- Number of COM-port, to which PTS controller is connected
- Enable pumps 1 – 8
- Set “Use pumps electronic totalizers”



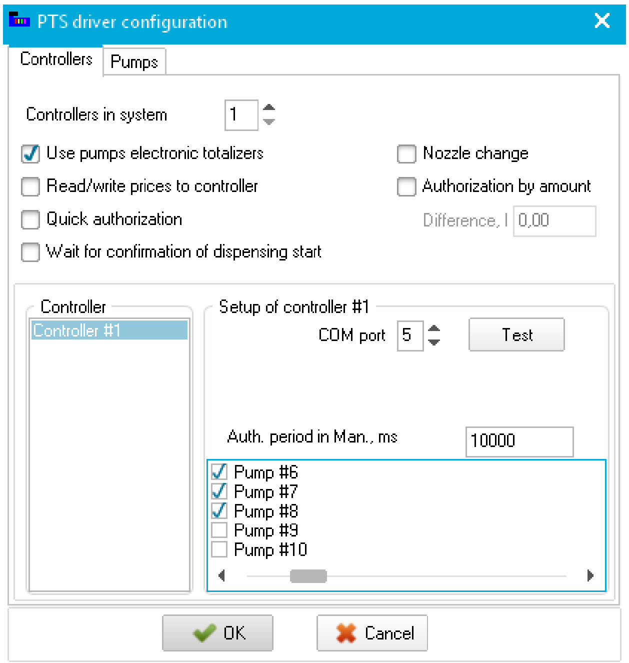
Controllers in system 1

☑ Use pumps electronic totalizers
□ Read/write prices to controller
□ Quick authorization
□ Wait for confirmation of dispensing start
□ Nozzle change
□ Authorization by amount
Difference, l 0,00

Controller: Controller #1

Setup of controller #1
COM port 5
Auth. period in Man., ms 10000

☑ Pump #1
☑ Pump #2
☑ Pump #3
☑ Pump #4
☑ Pump #5



Controllers in system 1

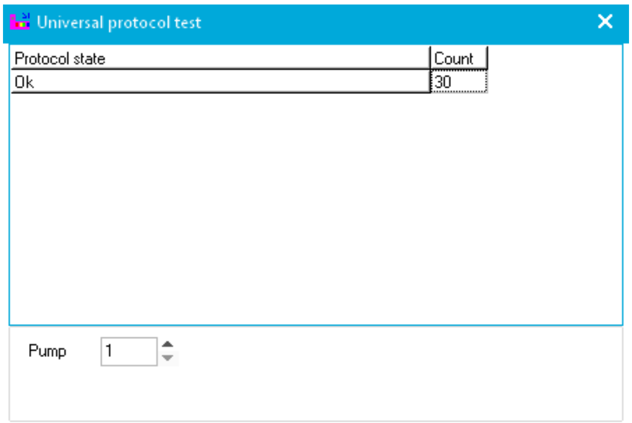
☑ Use pumps electronic totalizers
□ Read/write prices to controller
□ Quick authorization
□ Wait for confirmation of dispensing start
□ Nozzle change
□ Authorization by amount
Difference, l 0,00

Controller: Controller #1

Setup of controller #1
COM port 5
Auth. period in Man., ms 10000

☑ Pump #6
☑ Pump #7
☑ Pump #8
□ Pump #9
□ Pump #10

After specifying the COM-port number click on the button “Test”. The opened Window should show “OK” statuses, which say that PTS controller is connected to the system and is responding correctly. In case if you see any errors – there are mistakes in connection or configuration of the equipment.



Universal protocol test

| Protocol state | Count |
|---|---|
| Ok | 30 |

Pump: 1

On tab “Pumps” it is necessary to set for each of the configured pumps a correspondent number of the pump in controller:

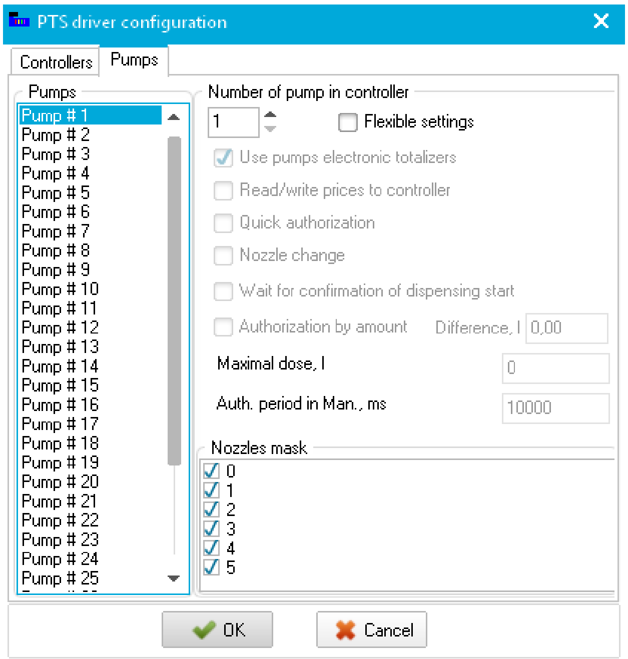
Pump #1 → 1

Pump #2 → 2

…………………

Pump #8 → 8

Also for each of the selected pump nozzle mask should include all nozzles (from 0 to 5).



PTS driver configuration

Controllers | Pumps

Pumps: Pump # 1 (selected)

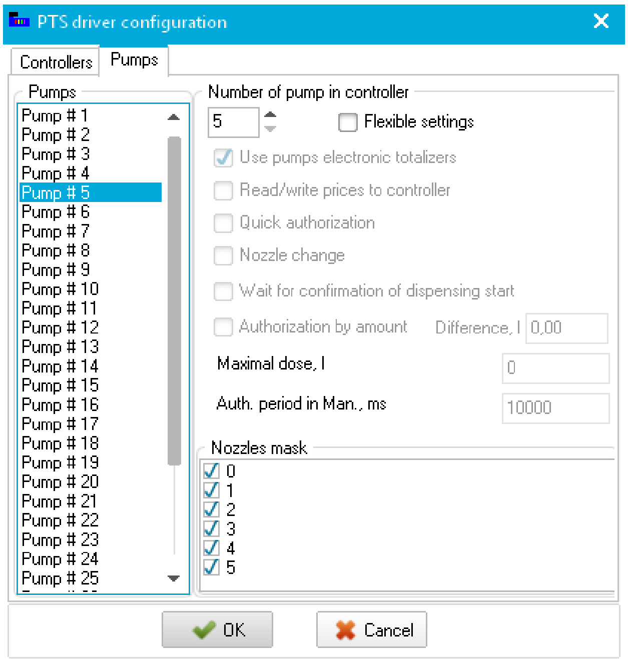
Number of pump in controller: 1

□ Flexible settings
☑ Use pumps electronic totalizers
□ Read/write prices to controller
□ Quick authorization
□ Nozzle change
□ Wait for confirmation of dispensing start
□ Authorization by amount Difference, l 0,00

Maximal dose, l: 0

Auth. period in Man., ms: 10000

Nozzles mask: ☑ 0 ☑ 1 ☑ 2 ☑ 3 ☑ 4 ☑ 5



PTS driver configuration

Controllers | Pumps

Pumps: Pump # 5 (selected)

Number of pump in controller: 5

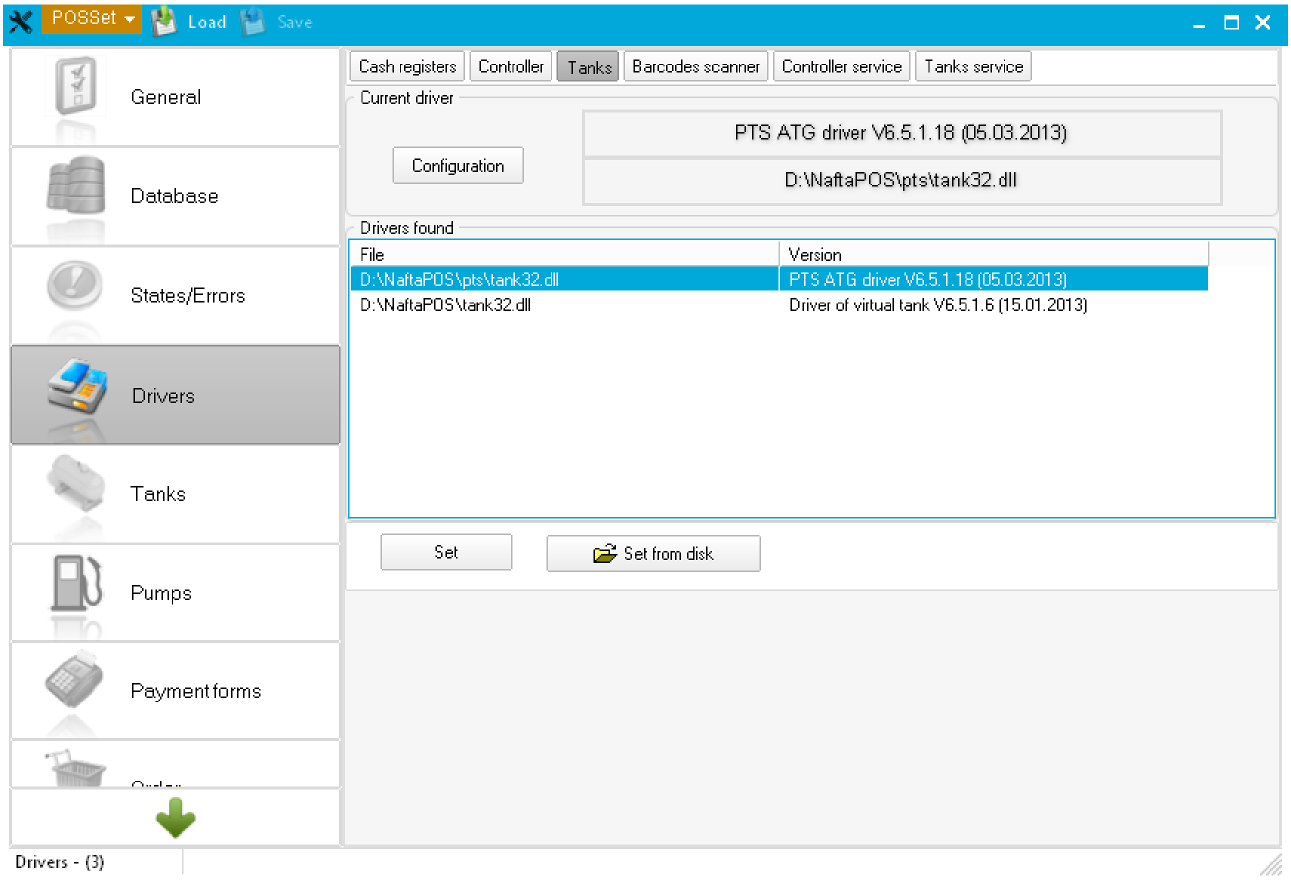
□ Flexible settings
☑ Use pumps electronic totalizers
□ Read/write prices to controller
□ Quick authorization
□ Nozzle change
□ Wait for confirmation of dispensing start
□ Authorization by amount Difference, l 0,00

Maximal dose, l: 0

Auth. period in Man., ms: 10000

Nozzles mask: ☑ 0 ☑ 1 ☑ 2 ☑ 3 ☑ 4 ☑ 5

As a driver of "Tanks" select a "PTS ATG" driver:



Current driver

PTS ATG driver V6.5.1.18 (05.03.2013)

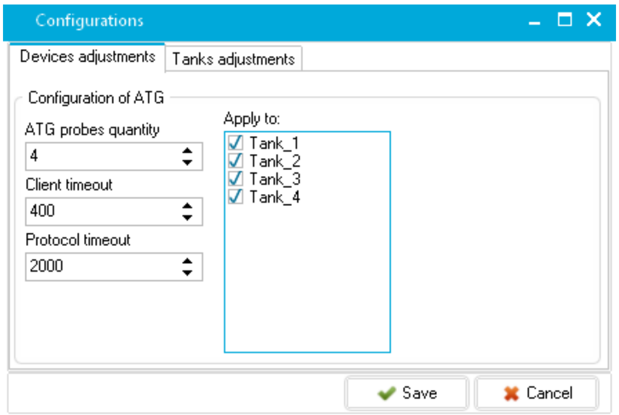
D:\NaftaPOS\pts\tank32.dll

Drivers found

| File | Version |
|---|---|
| D:\NaftaPOS\pts\tank32.dll | PTS ATG driver V6.5.1.18 (05.03.2013) |
| D:\NaftaPOS\tank32.dll | Driver of virtual tank V6.5.1.6 (15.01.2013) |

Drivers - (3)

Go to configuration of "PTS ATG" driver and set there the following configuration:



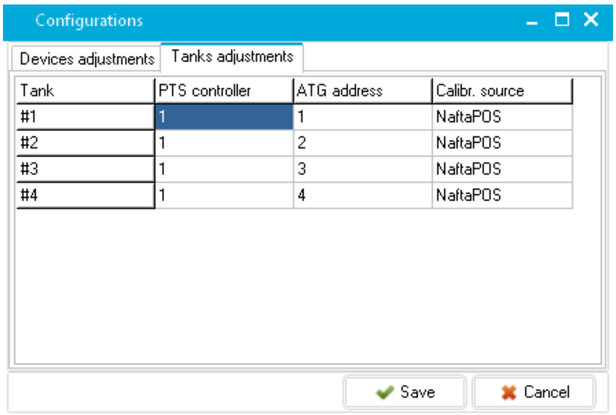
Configuration of ATG

ATG probes quantity: 4

Client timeout: 400

Protocol timeout: 2000

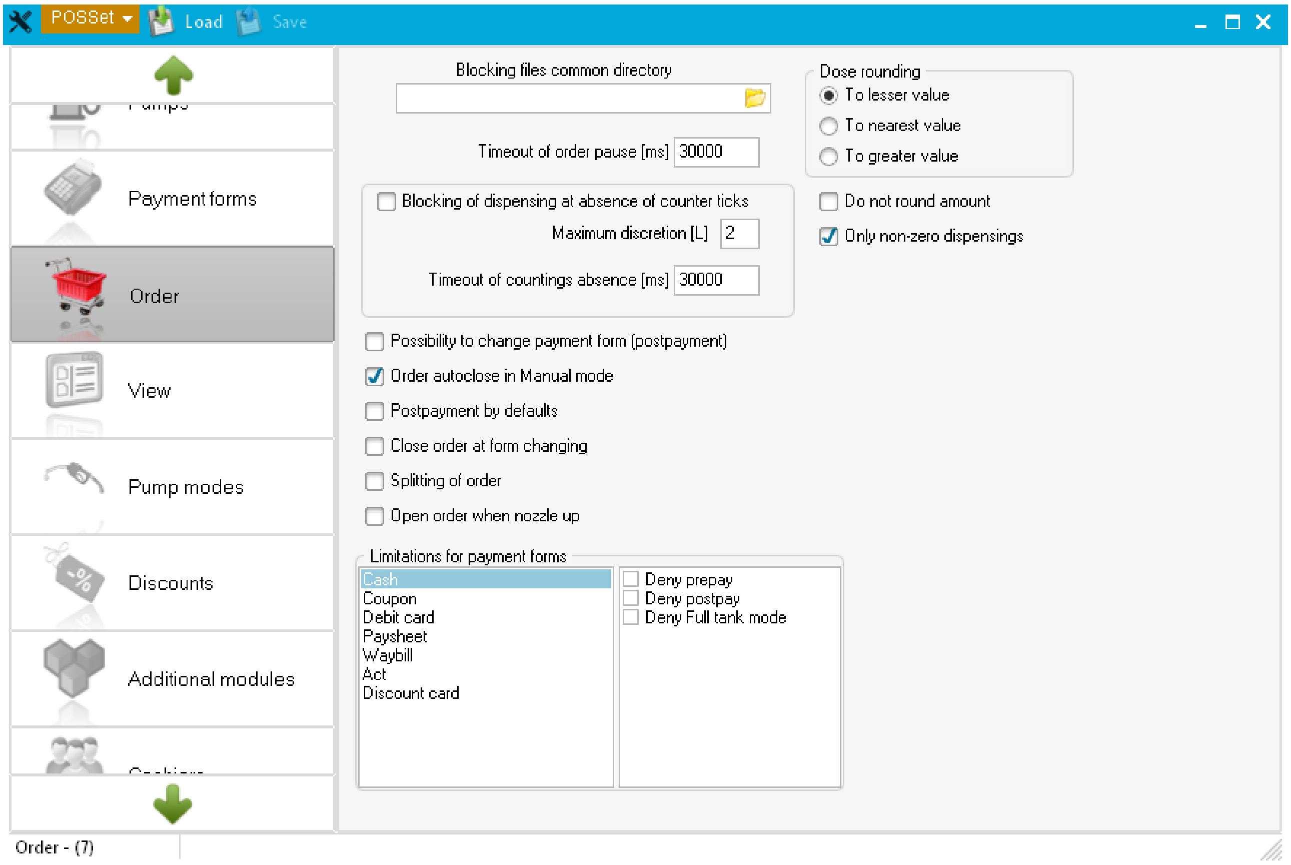
Apply to:
☑ Tank_1
☑ Tank_2
☑ Tank_3
☑ Tank_4



| Tank | PTS controller | ATG address | Calibr. source |
|---|---|---|---|
| #1 | 1 | 1 | NaftaPOS |
| #2 | 1 | 2 | NaftaPOS |
| #3 | 1 | 3 | NaftaPOS |
| #4 | 1 | 4 | NaftaPOS |

As drivers of "Barcode scanner", "Controller service", "Tanks service" leave the drivers, which are set as default.

Go to tab “Order” and check that configuration is set as shown on the screenshot:



Blocking files common directory

Timeout of order pause [ms] 30000

Dose rounding
● To lesser value
○ To nearest value
○ To greater value

□ Blocking of dispensing at absence of counter ticks
Maximum discretion [L] 2
Timeout of countings absence [ms] 30000

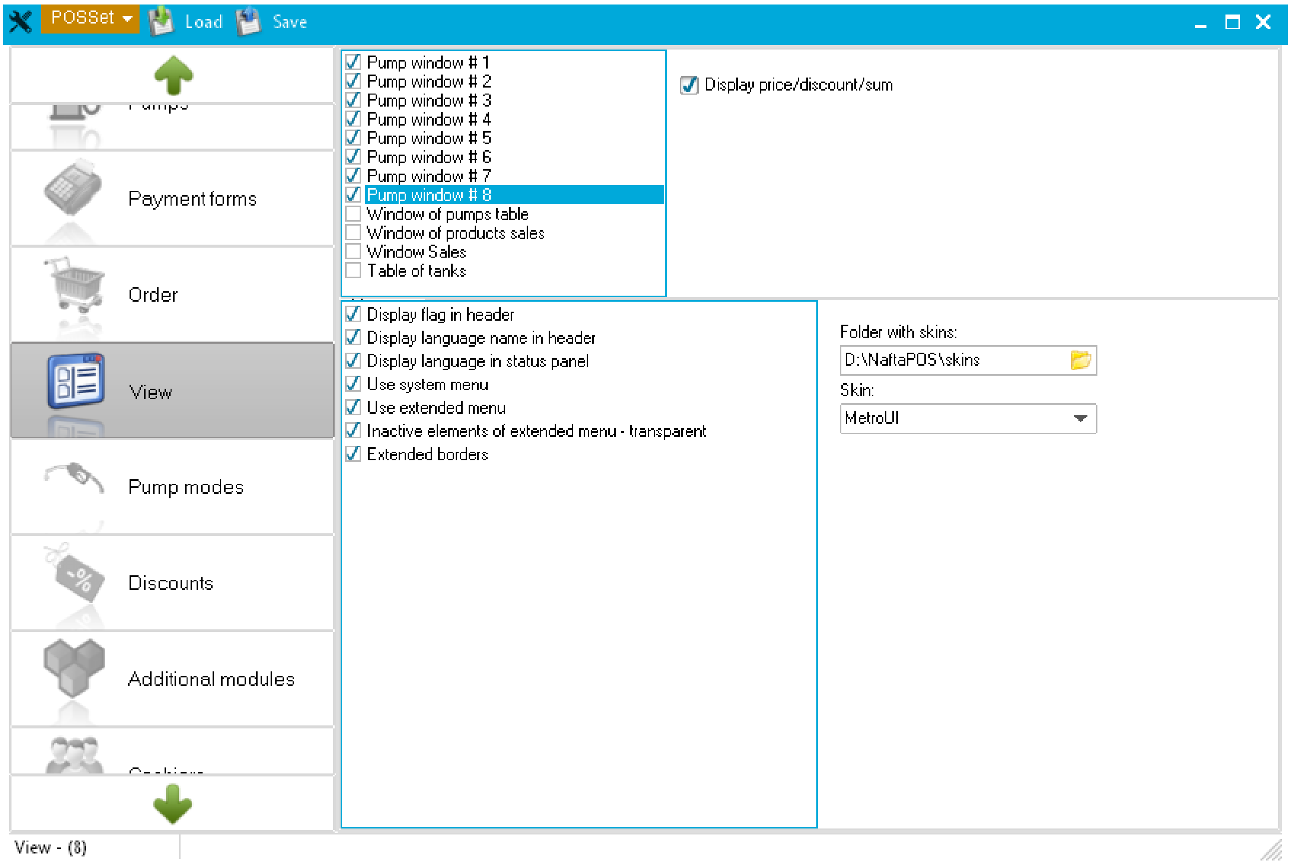
□ Do not round amount
☑ Only non-zero dispensings

□ Possibility to change payment form (postpayment)
☑ Order autoclose in Manual mode
□ Postpayment by defaults
□ Close order at form changing
□ Splitting of order
□ Open order when nozzle up

Limitations for payment forms
Cash
Coupon
Debit card
Paysheet
Waybill
Act
Discount card

□ Deny prepay
□ Deny postpay
□ Deny Full tank mode

On tab “View” check windows to be displayed (pumps 1 – 8 and table of tanks):



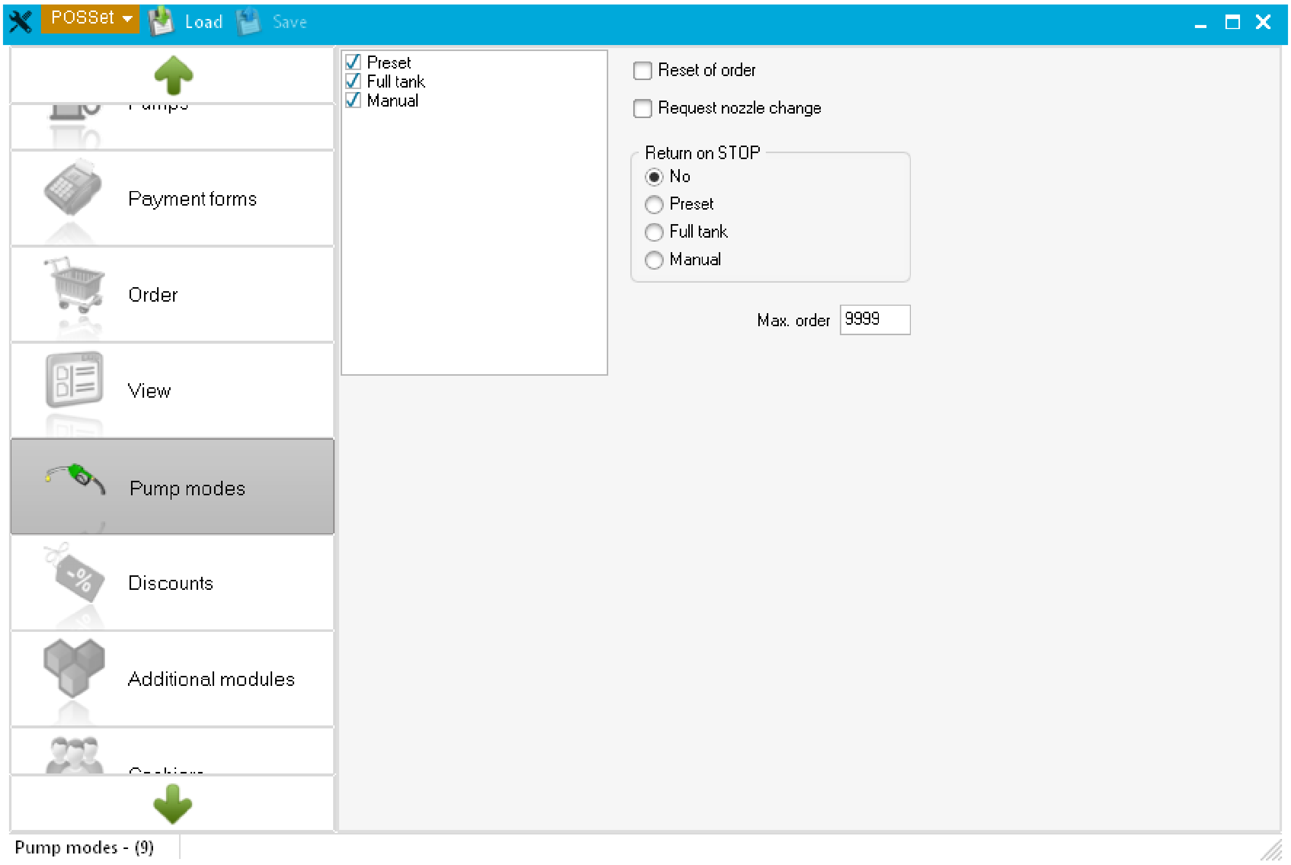
☑ Pump window # 1
☑ Pump window # 2
☑ Pump window # 3
☑ Pump window # 4
☑ Pump window # 5
☑ Pump window # 6
☑ Pump window # 7
☑ Pump window # 8
□ Window of pumps table
□ Window of products sales
□ Window Sales
□ Table of tanks

☑ Display price/discount/sum

☑ Display flag in header
☑ Display language name in header
☑ Display language in status panel
☑ Use system menu
☑ Use extended menu
☑ Inactive elements of extended menu - transparent
☑ Extended borders

Folder with skins:
D:\NaftaPOS\skins
Skin:
MetroUI

On tab “Pump modes” set the check that all modes are switched on:



☑ Preset
☑ Full tank
☑ Manual

□ Reset of order

□ Request nozzle change

Return on STOP
● No
○ Preset
○ Full tank
○ Manual

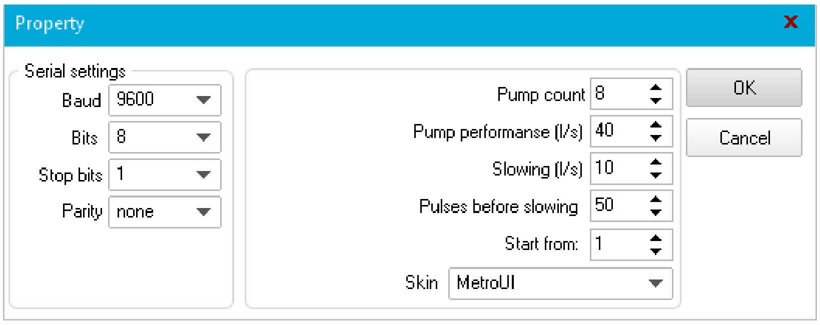
Max. order 9999

Now PTS controller is configured to work with a pumps software simulator and also ATG software simulator.

## *Step 6. Configuration of pumps software simulator*

As a pumps software simulator we will use UniPump software simulator in this step-by-step instruction.

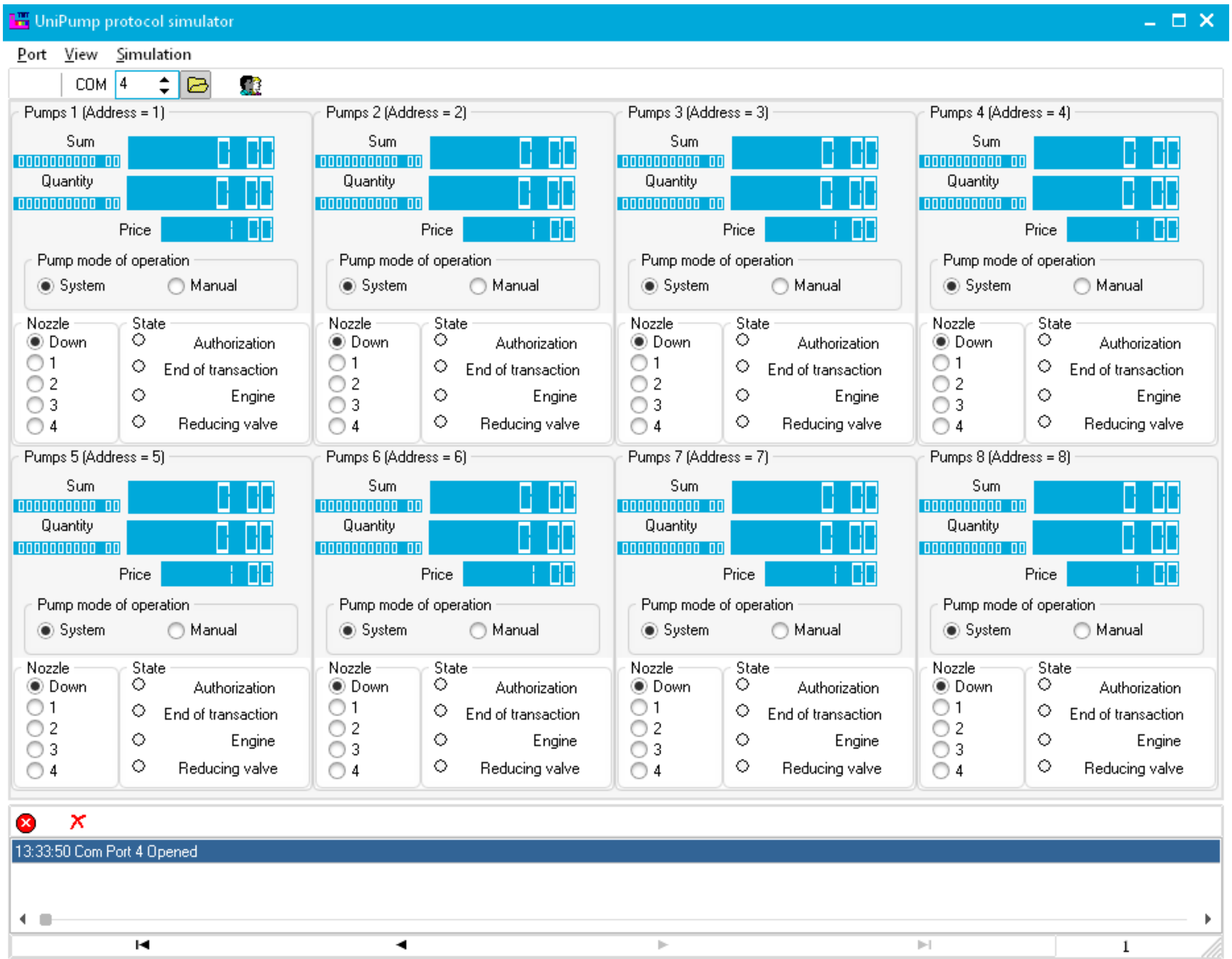
Run *SimUniPump.exe* and go to configuration of its properties. Set properties as shown on screenshot:



Serial settings
Baud 9600
Bits 8
Stop bits 1
Parity none

Pump count 8
Pump performanse (l/s) 40
Slowing (l/s) 10
Pulses before slowing 50
Start from: 1
Skin MetroUI

After properties are set click OK and on the main form select a COM-port, to which the output of RS-232/RS-485 interface converter board is connected, and click “Open” button:



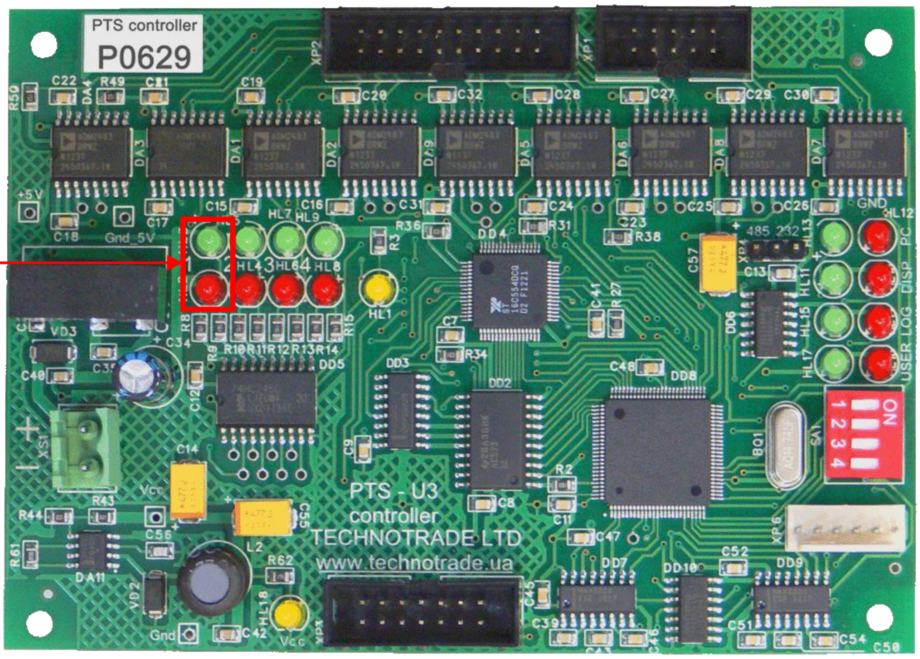
COM 4

Pump mode of operation: ● System ○ Manual

Nozzle: ● Down ○ 1 ○ 2 ○ 3 ○ 4

```
13:33:50 Com Port 4 Opened
```

After COM-port is opened in case all equipment is correctly connected and configured – you should see blinking of the green and red LEDs on pump channel 1 of the PTS controller:



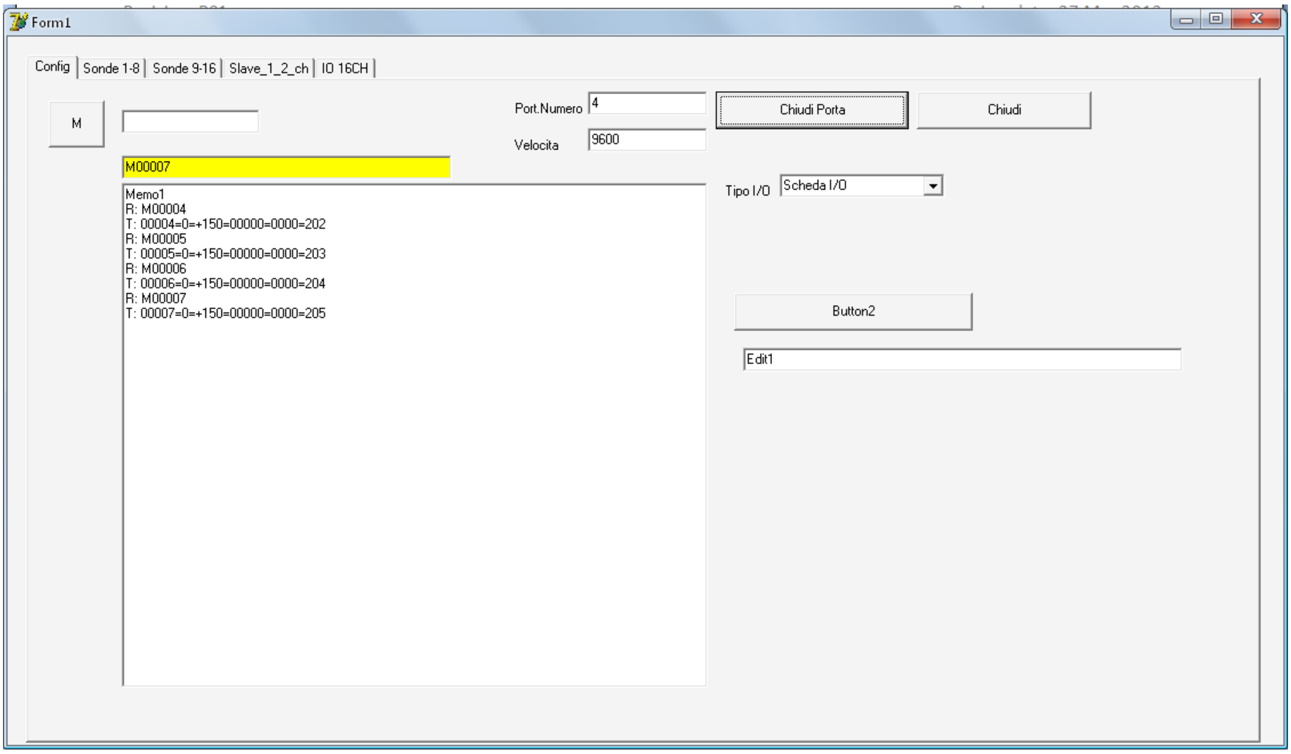
Green and red LEDs of pump channel 1 should be blinking

In case if only green LED is blinking on pump channel 1 – there is a mistake in connection or configuration of equipment.

## Step 7. Configuration of ATGs software simulator

As a pumps software simulator we will use Start Italiana software simulator in this step-by-step instruction.

Run *probe_simulator.exe* and go to tab “Config. Select a COM-port, to which the cable C029-3 is connected, and click “Apri porta” button as shown on screenshot:

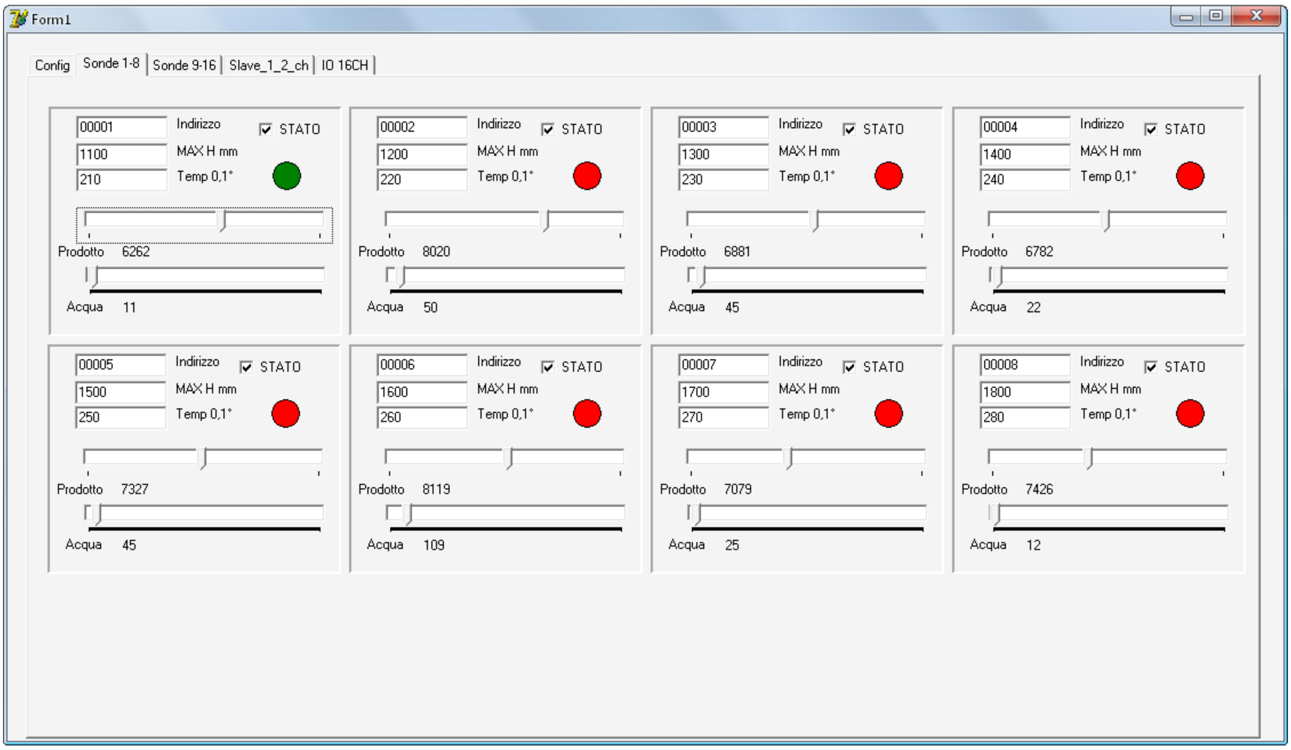


Port.Numero 4
Velocita 9600
Tipo I/O Scheda I/O
Edit1

M00007

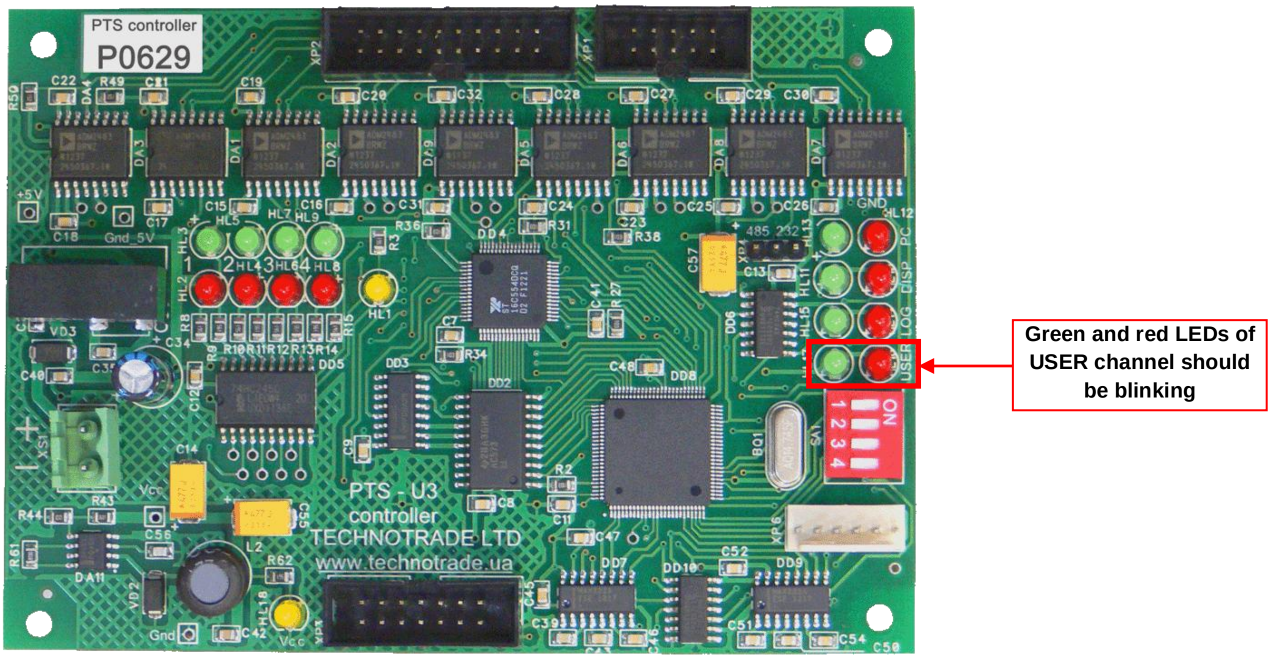
```
Memo1
R: M00004
T: 00004=0=+150=00000=0000=202
R: M00005
T: 00005=0=+150=00000=0000=203
R: M00006
T: 00006=0=+150=00000=0000=204
R: M00007
T: 00007=0=+150=00000=0000=205
```

Go to tab “Sonde 1-8”, where set some values for measurement of temperature, product and water levels for each of the probes:



| Indirizzo | MAX H mm | Temp 0,1° | STATO | Prodotto | Acqua |
|---|---|---|---|---|---|
| 00001 | 1100 | 210 | ☑ ● | 6262 | 11 |
| 00002 | 1200 | 220 | ☑ ● | 8020 | 50 |
| 00003 | 1300 | 230 | ☑ ● | 6881 | 45 |
| 00004 | 1400 | 240 | ☑ ● | 6782 | 22 |
| 00005 | 1500 | 250 | ☑ ● | 7327 | 45 |
| 00006 | 1600 | 260 | ☑ ● | 8119 | 109 |
| 00007 | 1700 | 270 | ☑ ● | 7079 | 25 |
| 00008 | 1800 | 280 | ☑ ● | 7426 | 12 |

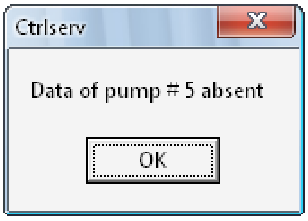
After COM-port is opened in case all equipment is correctly connected and configured – you should see blinking of the green and red LEDs on USER channel of the PTS controller happening once per 2 seconds:



In case if only green LED is blinking on USER channel – there is a mistake in connection or configuration of equipment.

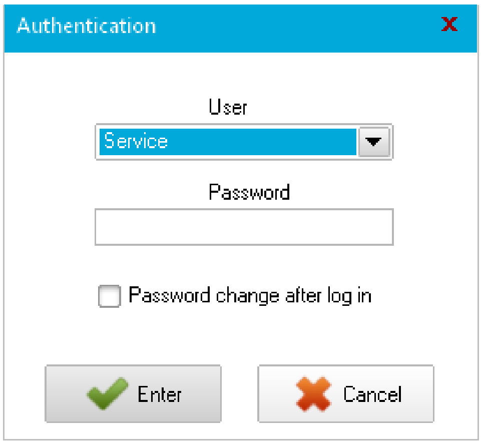
## Step 8. Running NaftaPOS software

Make sure that Guardant USB dongle is inserted into USB-port of PC and LED on it is shining.
Run NaftaPOS.exe. NaftaPOS software can show warning messages at first time for the pumps, which were added and has a number bigger than 4 (there are 4 pumps by default set in NaftaPOS). Click OK button.


Ctrlserv

Data of pump # 5 absent

After the system loads open a new shift (“Shift” → “New shift”) using your login / password (if default user credentials were not changed – then by defaults after installation there is only 1 user present with login ‘Service’ and no password (adding of other users and changing of passwords is made in configuration utility POSSet).


Authentication

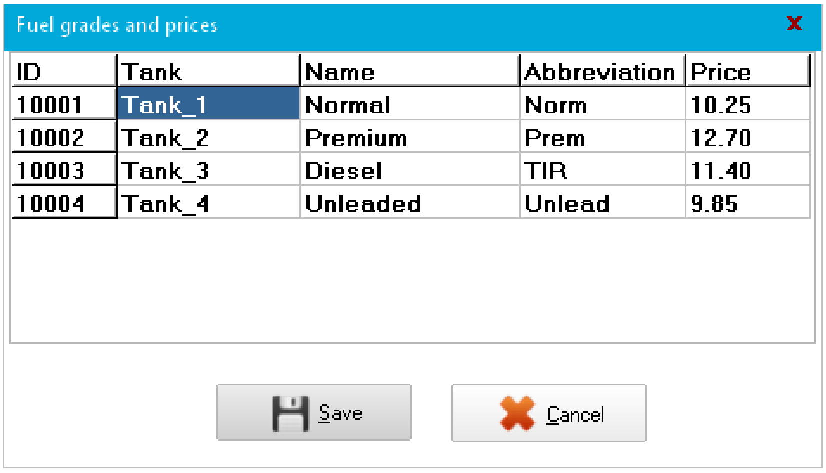
User

Service

Password

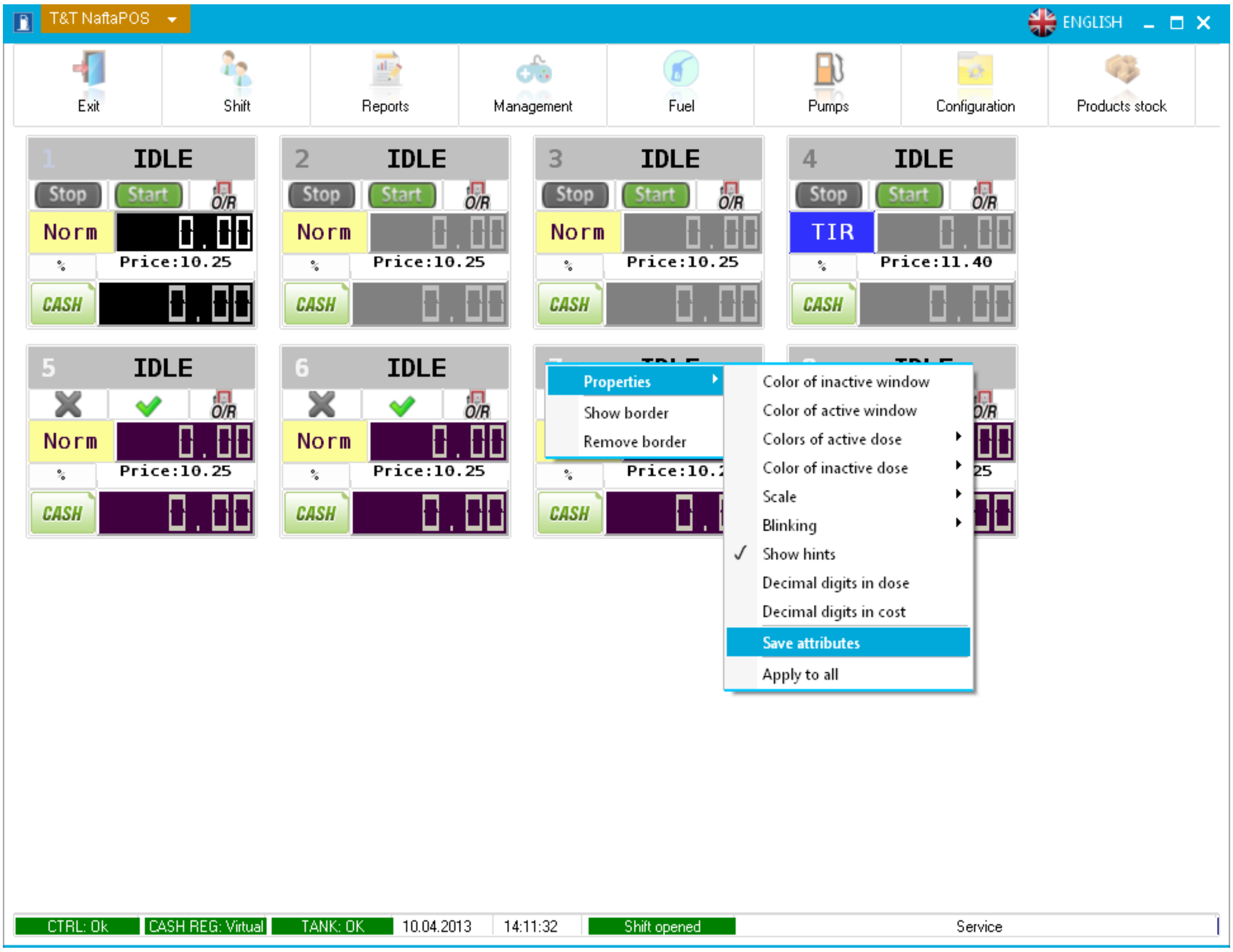
□ Password change after log in

After start the system will ask you to adjust the prices for fuel in tanks:

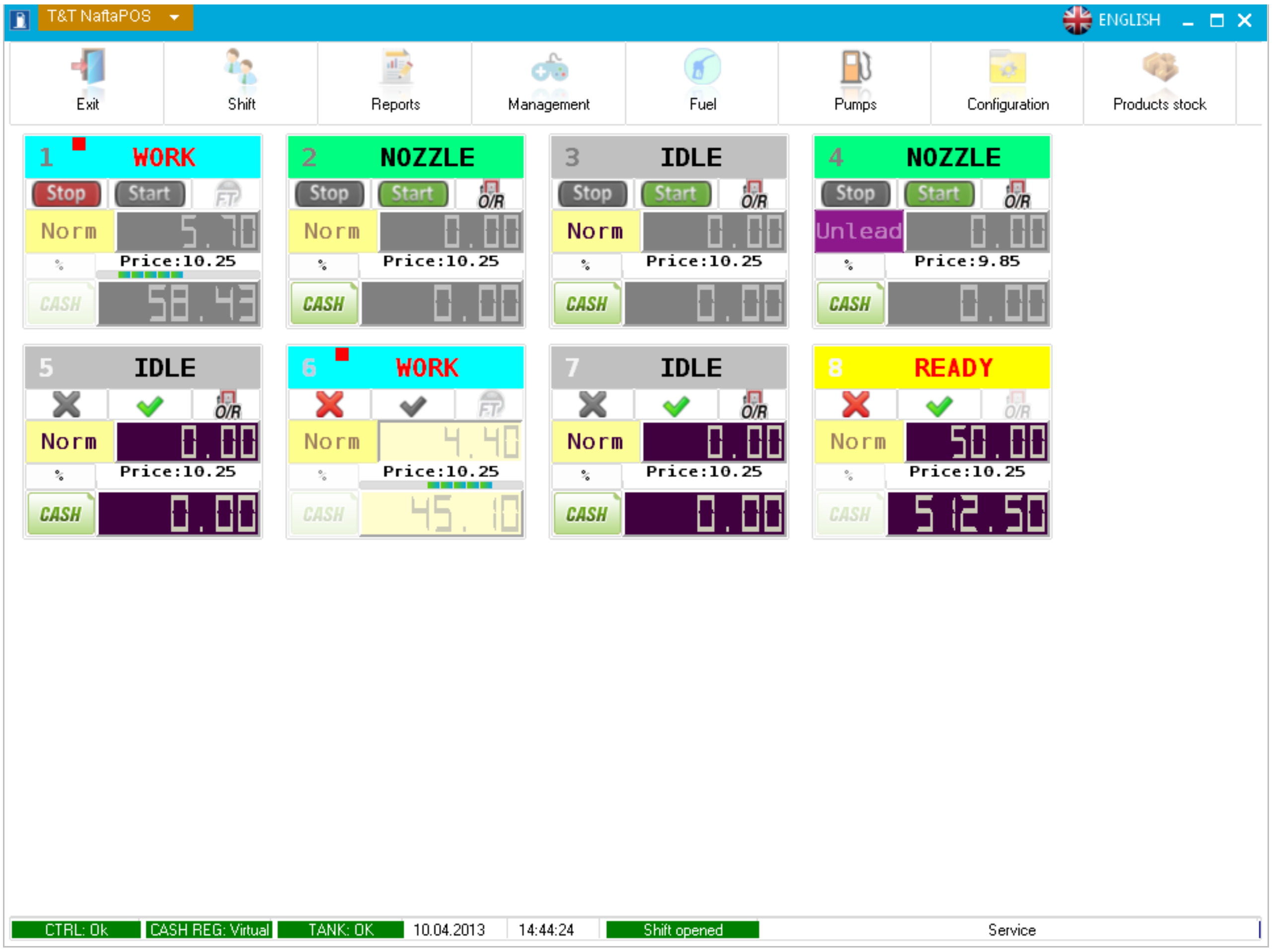

Fuel grades and prices

| ID | Tank | Name | Abbreviation | Price |
|---|---|---|---|---|
| 10001 | Tank_1 | Normal | Norm | 10.25 |
| 10002 | Tank_2 | Premium | Prem | 12.70 |
| 10003 | Tank_3 | Diesel | TIR | 11.40 |
| 10004 | Tank_4 | Unleaded | Unlead | 9.85 |

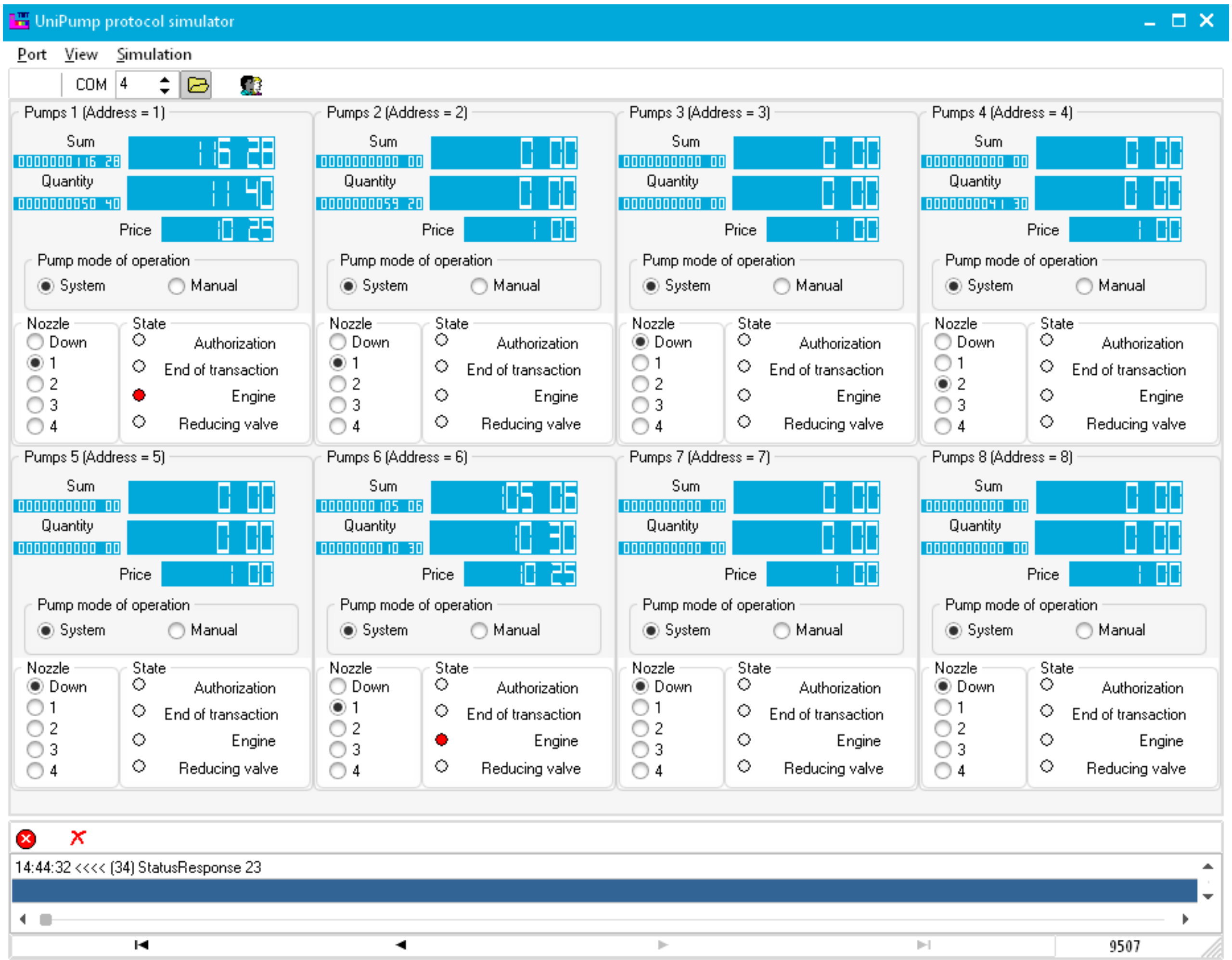
After that you will need to place icons of pumps on the screen as it is comfortable for you and make their scale as you need. This is made only once. After that in right-button menu of each of pump icons select "Save attributes":



And now it is possible to make control over pumps from the NaftaPOS software:



CTRL: Ok | CASH REG: Virtual | TANK: OK | 10.04.2013 | 14:44:24 | Shift opened | Service



| | Pumps 1 (Address = 1) | Pumps 2 (Address = 2) | Pumps 3 (Address = 3) | Pumps 4 (Address = 4) |
|---|---|---|---|---|
| Sum | 116.28 | 0.00 | 0.00 | 0.00 |
| Quantity | 11.40 | 0.00 | 0.00 | 0.00 |
| Price | 10.25 | 1.00 | 1.00 | 1.00 |
| Pump mode of operation | ● System ○ Manual | ● System ○ Manual | ● System ○ Manual | ● System ○ Manual |
| Nozzle | ● 1 | ● 1 | ● Down | ● 2 |
| State | ● Engine | | | |

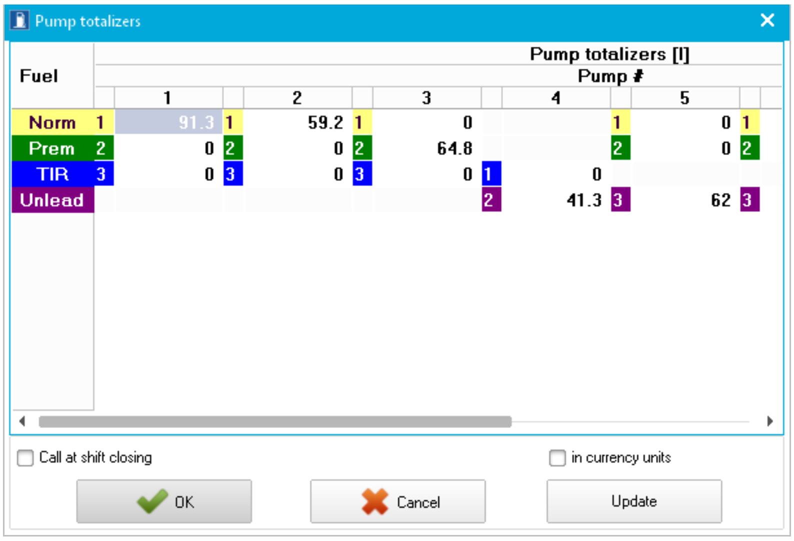
| | Pumps 5 (Address = 5) | Pumps 6 (Address = 6) | Pumps 7 (Address = 7) | Pumps 8 (Address = 8) |
|---|---|---|---|---|
| Sum | 0.00 | 105.06 | 0.00 | 0.00 |
| Quantity | 0.00 | 10.30 | 0.00 | 0.00 |
| Price | 1.00 | 10.25 | 1.00 | 1.00 |
| Pump mode of operation | ● System ○ Manual | ● System ○ Manual | ● System ○ Manual | ● System ○ Manual |
| Nozzle | ● Down | ● 1 | ● Down | ● Down |
| State | | ● Engine | | |

Nozzle: Down 1 2 3 4
State: Authorization, End of transaction, Engine, Reducing valve

```
14:44:32 <<<< (34) StatusResponse 23
```

9507

Total counter values from dispensers can be read using a main menu item “Pumps” → “Totalizers”

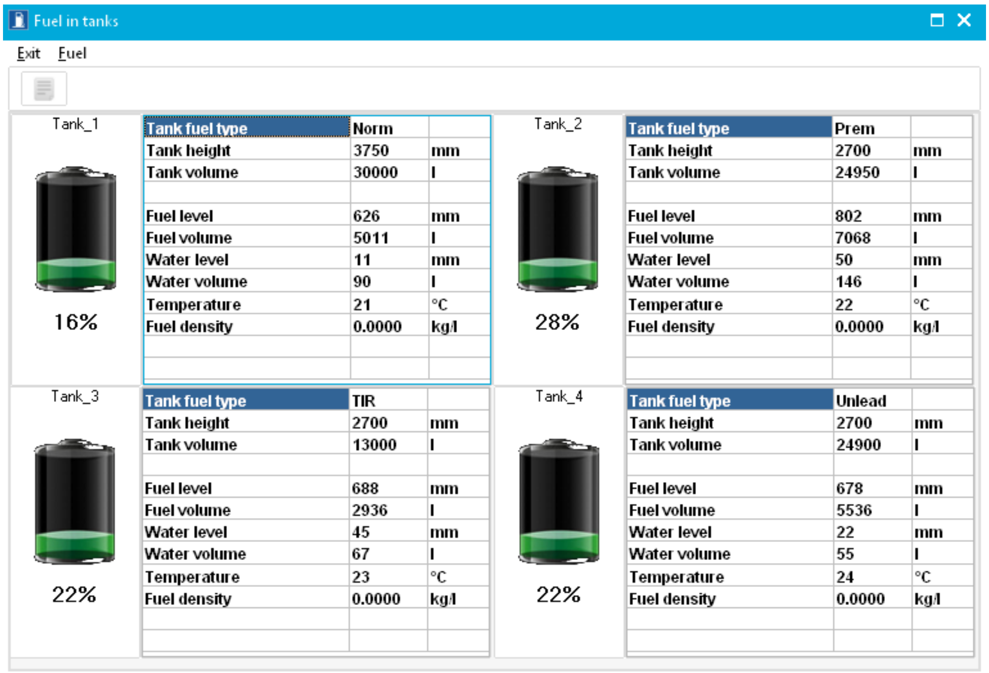


Pump totalizers

| Fuel | Pump totalizers [l] | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Pump # | | | | | | | | | | |
| | | 1 | | 2 | | 3 | | 4 | | 5 | |
| Norm | 1 | 91.3 | 1 | 59.2 | 1 | 0 | | | 1 | 0 | 1 |
| Prem | 2 | 0 | 2 | 0 | 2 | 64.8 | | | 2 | 0 | 2 |
| TIR | 3 | 0 | 3 | 0 | 3 | 0 | 1 | 0 | | | |
| Unlead | | | | | | | 2 | 41.3 | 3 | 62 | 3 |

☐ Call at shift closing ☐ in currency units

For viewing of fuel parameters in tank select a main menu item “Fuel” → “Fuel in tanks” and you will be able to view measurements of the ATG probes:



Fuel in tanks

Tank_1 16%

| Tank fuel type | Norm | |
|---|---|---|
| Tank height | 3750 | mm |
| Tank volume | 30000 | l |
| | | |
| Fuel level | 626 | mm |
| Fuel volume | 5011 | l |
| Water level | 11 | mm |
| Water volume | 90 | l |
| Temperature | 21 | °C |
| Fuel density | 0.0000 | kg/l |

Tank_2 28%

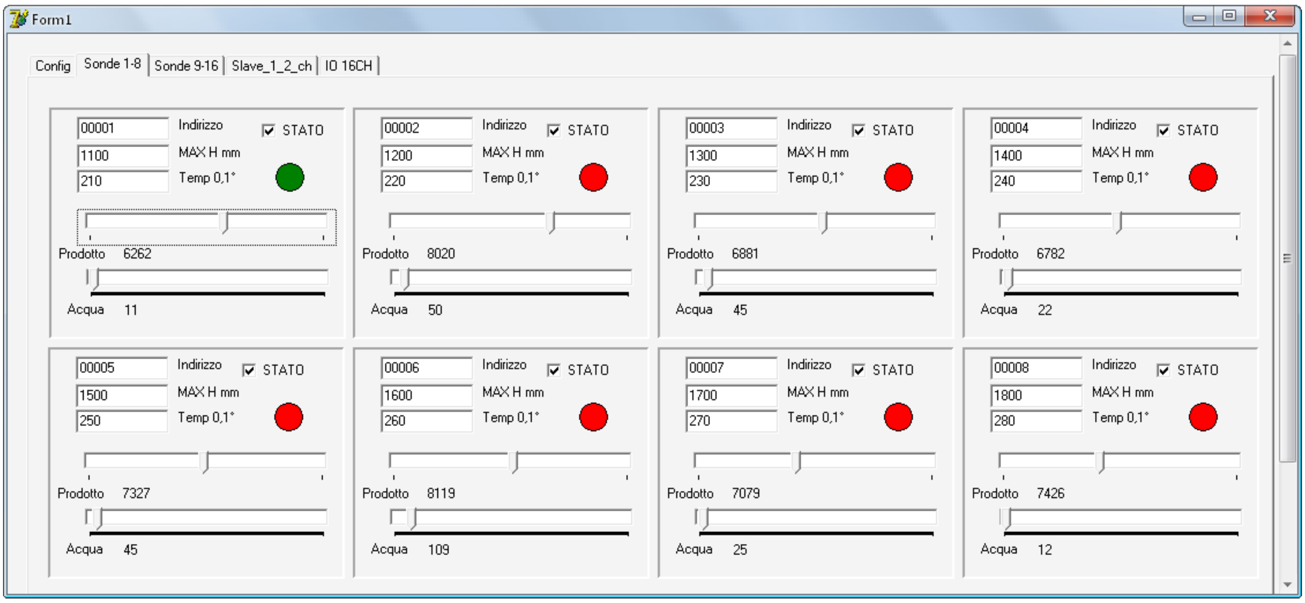
| Tank fuel type | Prem | |
|---|---|---|
| Tank height | 2700 | mm |
| Tank volume | 24950 | l |
| | | |
| Fuel level | 802 | mm |
| Fuel volume | 7068 | l |
| Water level | 50 | mm |
| Water volume | 146 | l |
| Temperature | 22 | °C |
| Fuel density | 0.0000 | kg/l |

Tank_3 22%

| Tank fuel type | TIR | |
|---|---|---|
| Tank height | 2700 | mm |
| Tank volume | 13000 | l |
| | | |
| Fuel level | 688 | mm |
| Fuel volume | 2936 | l |
| Water level | 45 | mm |
| Water volume | 67 | l |
| Temperature | 23 | °C |
| Fuel density | 0.0000 | kg/l |

Tank_4 22%

| Tank fuel type | Unlead | |
|---|---|---|
| Tank height | 2700 | mm |
| Tank volume | 24900 | l |
| | | |
| Fuel level | 678 | mm |
| Fuel volume | 5536 | l |
| Water level | 22 | mm |
| Water volume | 55 | l |
| Temperature | 24 | °C |
| Fuel density | 0.0000 | kg/l |



Form1

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Indirizzo | 00001 | 00002 | 00003 | 00004 | 00005 | 00006 | 00007 | 00008 |
| MAX H mm | 1100 | 1200 | 1300 | 1400 | 1500 | 1600 | 1700 | 1800 |
| Temp 0,1° | 210 | 220 | 230 | 240 | 250 | 260 | 270 | 280 |
| STATO | ☑ ● | ☑ ● | ☑ ● | ☑ ● | ☑ ● | ☑ ● | ☑ ● | ☑ ● |
| Prodotto | 6262 | 8020 | 6881 | 6782 | 7327 | 8119 | 7079 | 7426 |
| Acqua | 11 | 50 | 45 | 22 | 45 | 109 | 25 | 12 |

Also remains of fuel in tanks is possible to view from main form of NaftaPOS. For this select a main menu item “View” → “Table of tanks”:

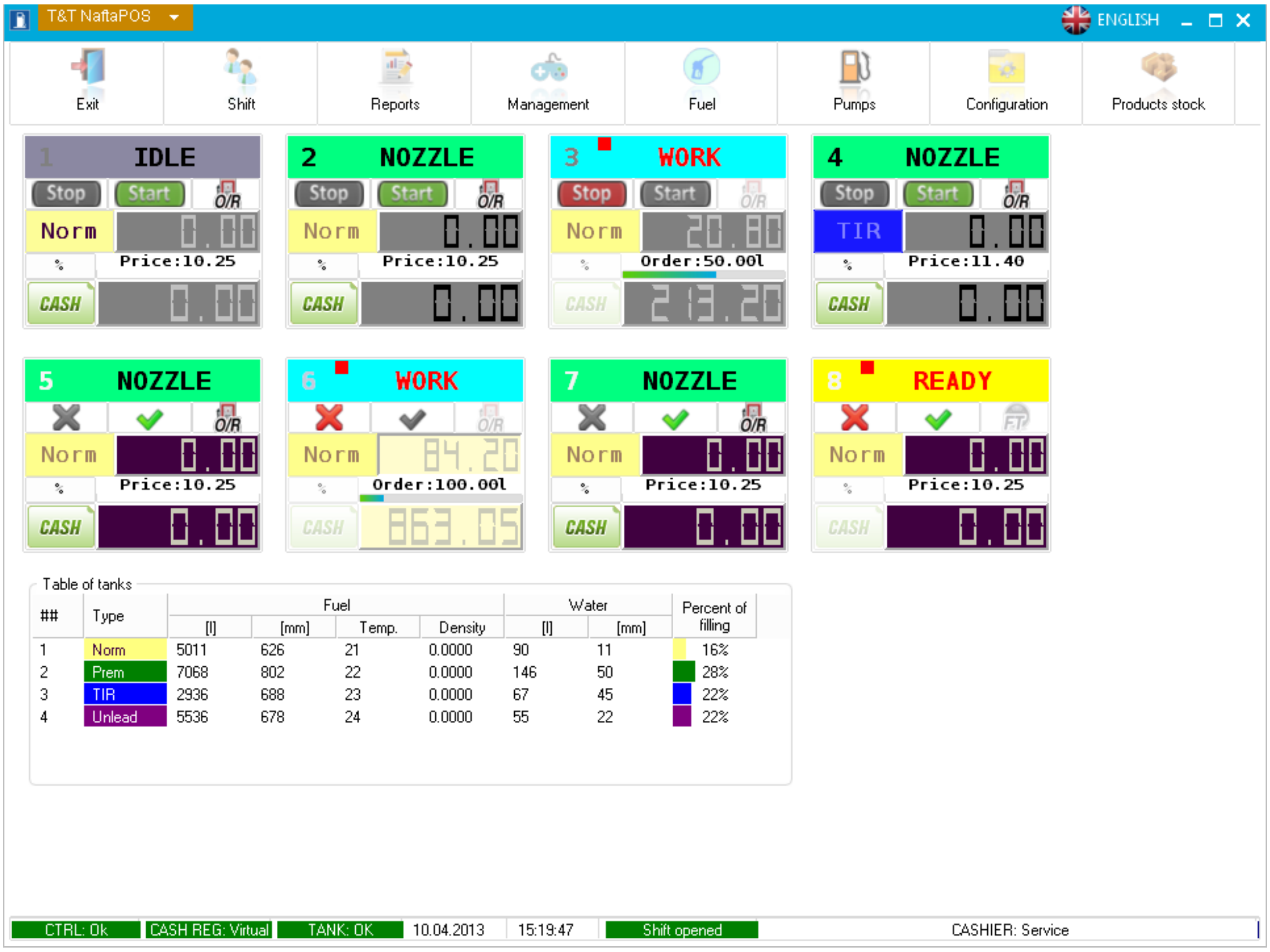


Table of tanks

| ## | Type | Fuel [l] | Fuel [mm] | Fuel Temp. | Fuel Density | Water [l] | Water [mm] | Percent of filling |
|---|---|---|---|---|---|---|---|---|
| 1 | Norm | 5011 | 626 | 21 | 0.0000 | 90 | 11 | 16% |
| 2 | Prem | 7068 | 802 | 22 | 0.0000 | 146 | 50 | 28% |
| 3 | TIR | 2936 | 688 | 23 | 0.0000 | 67 | 45 | 22% |
| 4 | Unlead | 5536 | 678 | 24 | 0.0000 | 55 | 22 | 22% |

CTRL: Ok | CASH REG: Virtual | TANK: OK | 10.04.2013 | 15:19:47 | Shift opened | CASHIER: Service

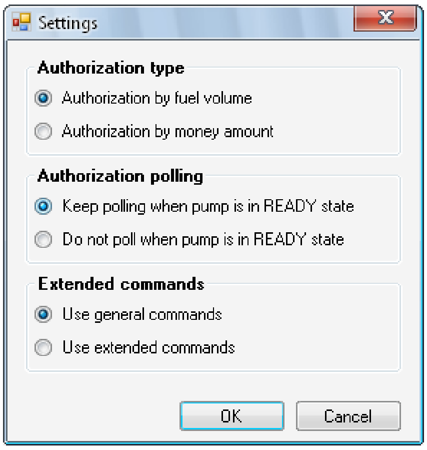
Now the NaftaPOS system correctly works with pumps and ATGs software simulators.

## *Step 9. Configuration and running of open-source application for PTS controller*

The open-source application written in C# and VB.NET is a good place to start development of a new control software for petrol stations. In this section we will see how it is easy to configure and run it.

Compile or simply run PTS Application (application is given in PTS SDK software kit and is located in API TOOLS folder.

Go to main menu item “Configuration” → “Settings” and set the following:



Settings

**Authorization type**

- ● Authorization by fuel volume
- ○ Authorization by money amount

**Authorization polling**

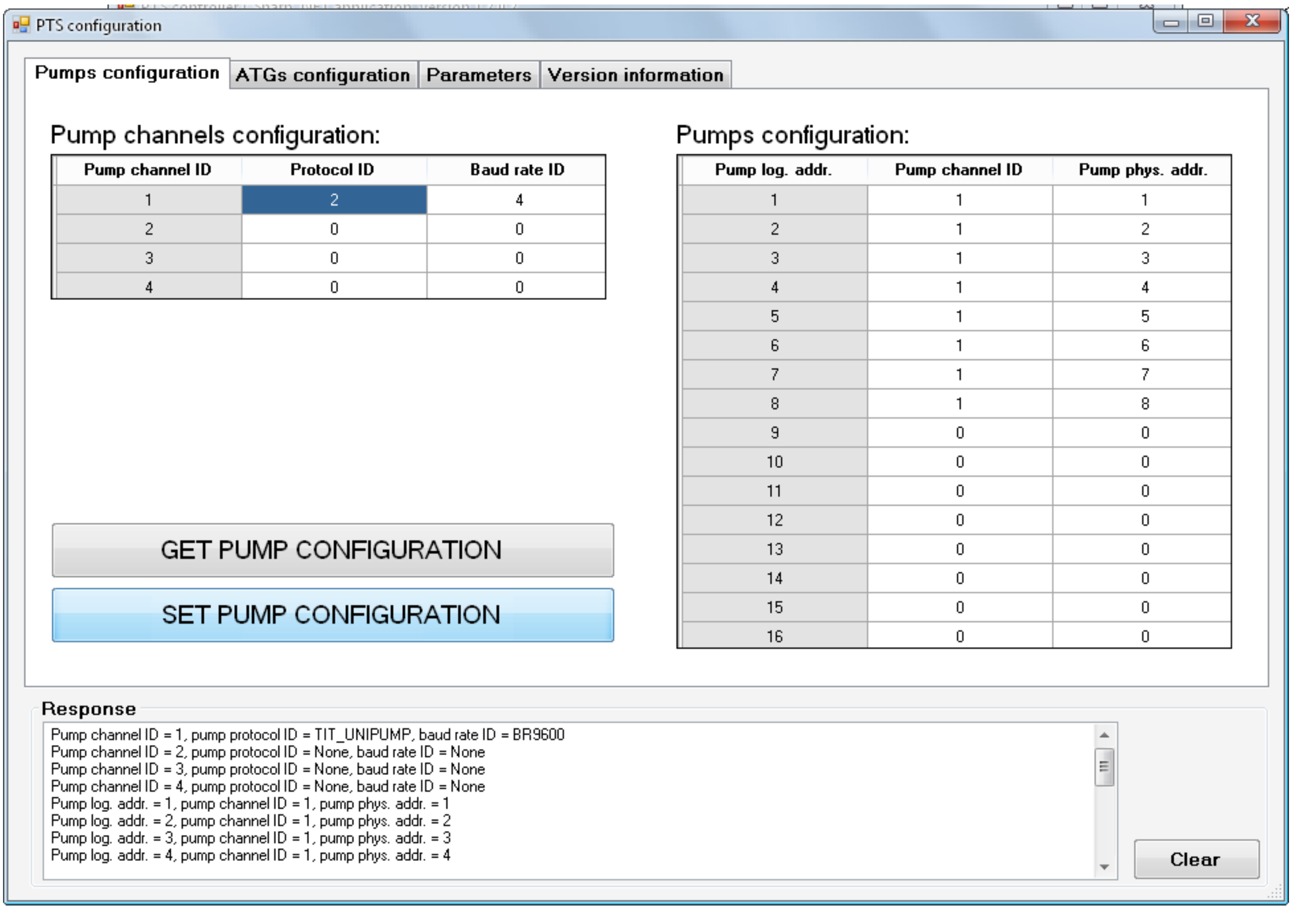
- ● Keep polling when pump is in READY state
- ○ Do not poll when pump is in READY state

**Extended commands**

- ● Use general commands
- ○ Use extended commands

OK Cancel

Open a COM-port and go to main menu item “Configuration” → “Fuel point configuration settings”. Make sure that all pumps channels and ATG channels are configured correctly:



Pump channels configuration:

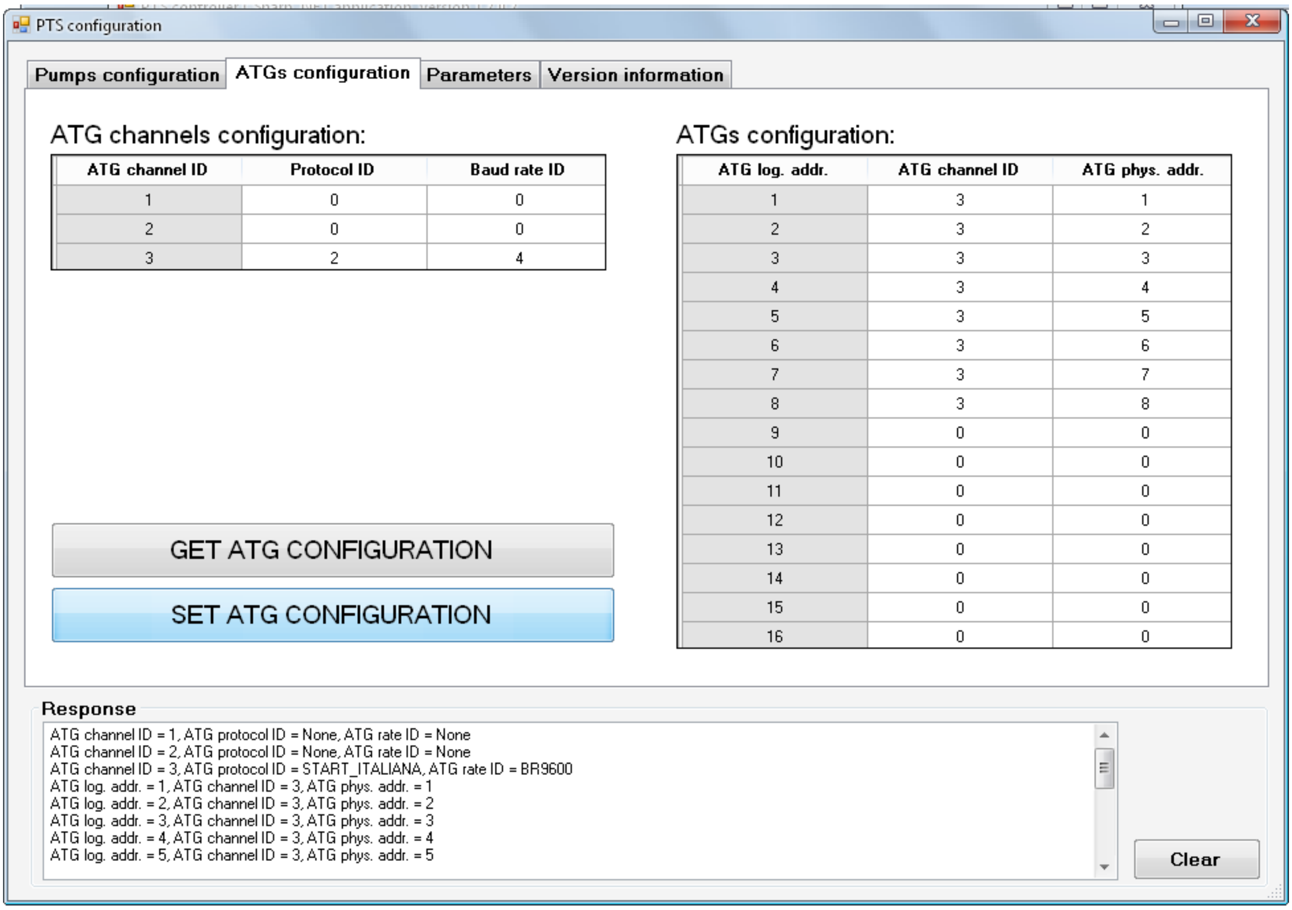
| Pump channel ID | Protocol ID | Baud rate ID |
|---|---|---|
| 1 | 2 | 4 |
| 2 | 0 | 0 |
| 3 | 0 | 0 |
| 4 | 0 | 0 |

Pumps configuration:

| Pump log. addr. | Pump channel ID | Pump phys. addr. |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 1 | 2 |
| 3 | 1 | 3 |
| 4 | 1 | 4 |
| 5 | 1 | 5 |
| 6 | 1 | 6 |
| 7 | 1 | 7 |
| 8 | 1 | 8 |
| 9 | 0 | 0 |
| 10 | 0 | 0 |
| 11 | 0 | 0 |
| 12 | 0 | 0 |
| 13 | 0 | 0 |
| 14 | 0 | 0 |
| 15 | 0 | 0 |
| 16 | 0 | 0 |

Response

```
Pump channel ID = 1, pump protocol ID = TIT_UNIPUMP, baud rate ID = BR9600
Pump channel ID = 2, pump protocol ID = None, baud rate ID = None
Pump channel ID = 3, pump protocol ID = None, baud rate ID = None
Pump channel ID = 4, pump protocol ID = None, baud rate ID = None
Pump log. addr. = 1, pump channel ID = 1, pump phys. addr. = 1
Pump log. addr. = 2, pump channel ID = 1, pump phys. addr. = 2
Pump log. addr. = 3, pump channel ID = 1, pump phys. addr. = 3
Pump log. addr. = 4, pump channel ID = 1, pump phys. addr. = 4
```



ATG channels configuration:

| ATG channel ID | Protocol ID | Baud rate ID |
|---|---|---|
| 1 | 0 | 0 |
| 2 | 0 | 0 |
| 3 | 2 | 4 |

ATGs configuration:

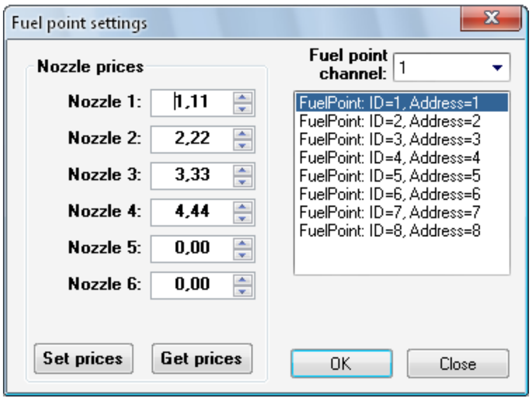
| ATG log. addr. | ATG channel ID | ATG phys. addr. |
|---|---|---|
| 1 | 3 | 1 |
| 2 | 3 | 2 |
| 3 | 3 | 3 |
| 4 | 3 | 4 |
| 5 | 3 | 5 |
| 6 | 3 | 6 |
| 7 | 3 | 7 |
| 8 | 3 | 8 |
| 9 | 0 | 0 |
| 10 | 0 | 0 |
| 11 | 0 | 0 |
| 12 | 0 | 0 |
| 13 | 0 | 0 |
| 14 | 0 | 0 |
| 15 | 0 | 0 |
| 16 | 0 | 0 |

Response

```
ATG channel ID = 1, ATG protocol ID = None, ATG rate ID = None
ATG channel ID = 2, ATG protocol ID = None, ATG rate ID = None
ATG channel ID = 3, ATG protocol ID = START_ITALIANA, ATG rate ID = BR9600
ATG log. addr. = 1, ATG channel ID = 3, ATG phys. addr. = 1
ATG log. addr. = 2, ATG channel ID = 3, ATG phys. addr. = 2
ATG log. addr. = 3, ATG channel ID = 3, ATG phys. addr. = 3
ATG log. addr. = 4, ATG channel ID = 3, ATG phys. addr. = 4
ATG log. addr. = 5, ATG channel ID = 3, ATG phys. addr. = 5
```

In main form of the application for each of the fuel point icons go to setting ("Settings" button) and set a correspondent fuel point number (from fuel point channel 1) and prices for each of the nozzles:

- Fuel point icon #1 → FuelPoint ID = 1, Address = 1
- Fuel point icon #2 → FuelPoint ID = 2, Address = 2
- ……………………………………………………………………………
- Fuel point icon #1 → FuelPoint ID = 8, Address = 8



Fuel point settings

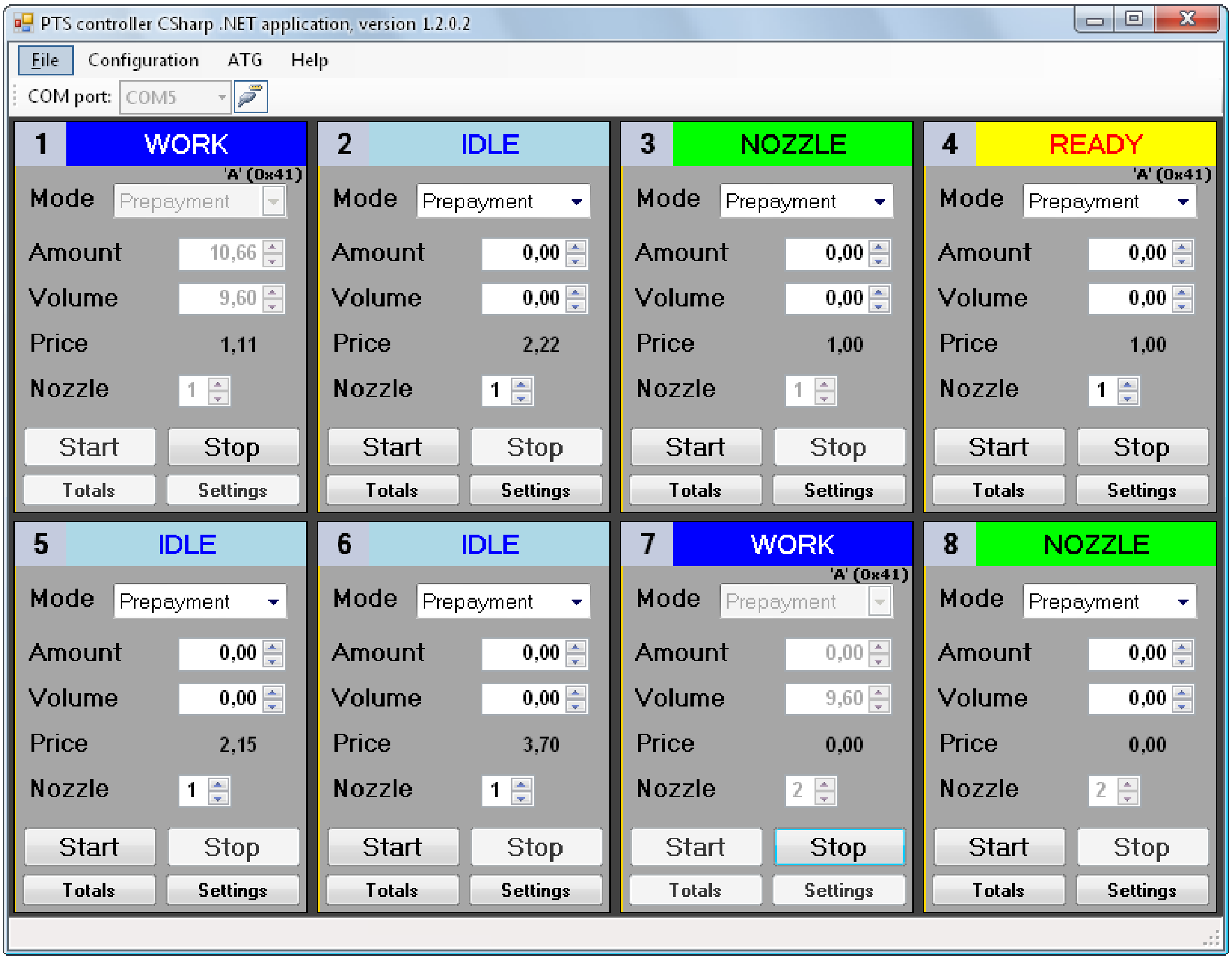
Nozzle prices

Nozzle 1: 1,11
Nozzle 2: 2,22
Nozzle 3: 3,33
Nozzle 4: 4,44
Nozzle 5: 0,00
Nozzle 6: 0,00

Fuel point channel: 1

FuelPoint: ID=1, Address=1
FuelPoint: ID=2, Address=2
FuelPoint: ID=3, Address=3
FuelPoint: ID=4, Address=4
FuelPoint: ID=5, Address=5
FuelPoint: ID=6, Address=6
FuelPoint: ID=7, Address=7
FuelPoint: ID=8, Address=8

Now the main form of the application allows to provide control over pumps in pump software simulator:



PTS controller CSharp .NET application, version 1.2.0.2

COM port: COM5

1 WORK 'A' (0x41)
Mode Prepayment
Amount 10,66
Volume 9,60
Price 1,11
Nozzle 1

2 IDLE
Mode Prepayment
Amount 0,00
Volume 0,00
Price 2,22
Nozzle 1

3 NOZZLE
Mode Prepayment
Amount 0,00
Volume 0,00
Price 1,00
Nozzle 1

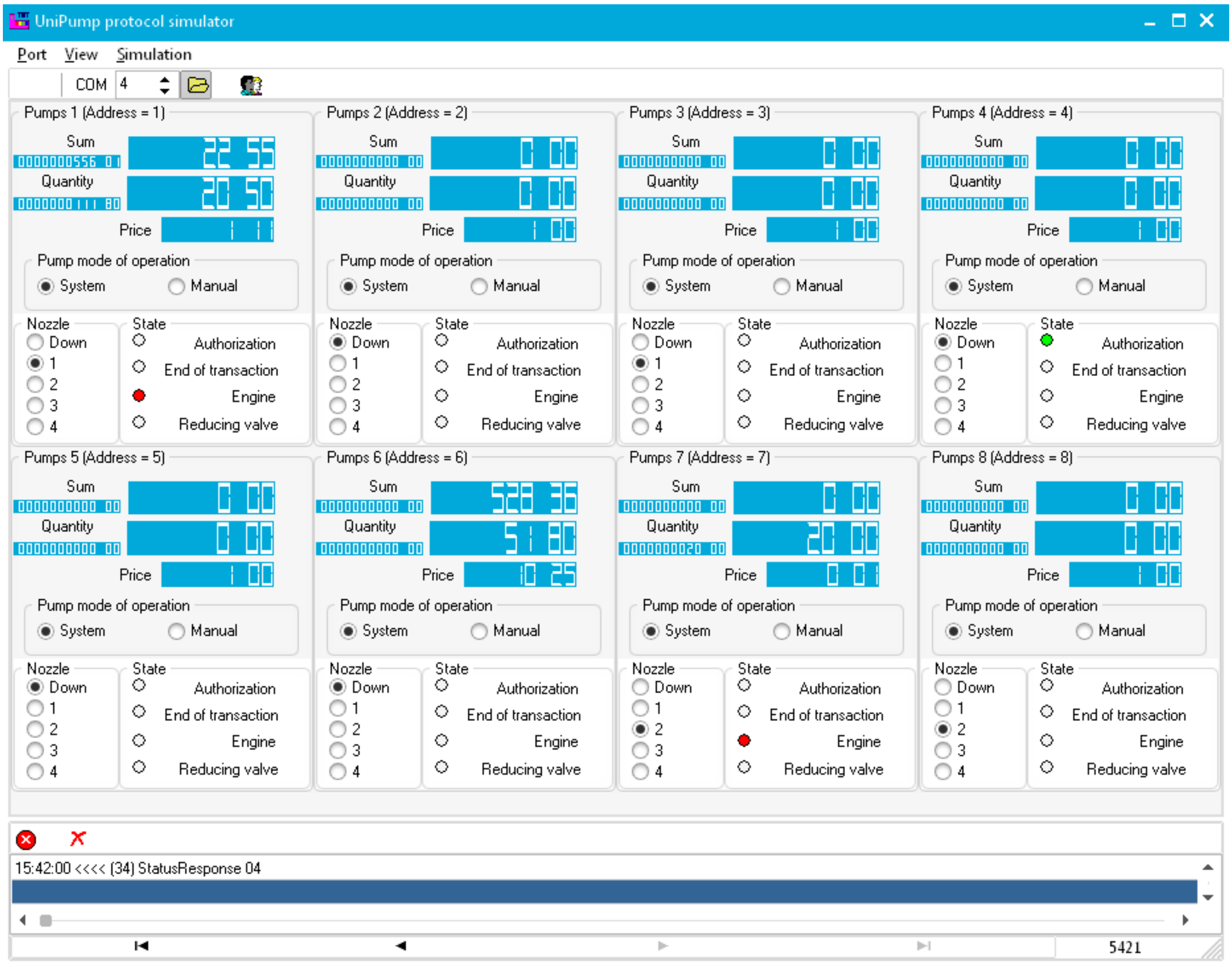
4 READY 'A' (0x41)
Mode Prepayment
Amount 0,00
Volume 0,00
Price 1,00
Nozzle 1

5 IDLE
Mode Prepayment
Amount 0,00
Volume 0,00
Price 2,15
Nozzle 1

6 IDLE
Mode Prepayment
Amount 0,00
Volume 0,00
Price 3,70
Nozzle 1

7 WORK 'A' (0x41)
Mode Prepayment
Amount 0,00
Volume 9,60
Price 0,00
Nozzle 2

8 NOZZLE
Mode Prepayment
Amount 0,00
Volume 0,00
Price 0,00
Nozzle 2

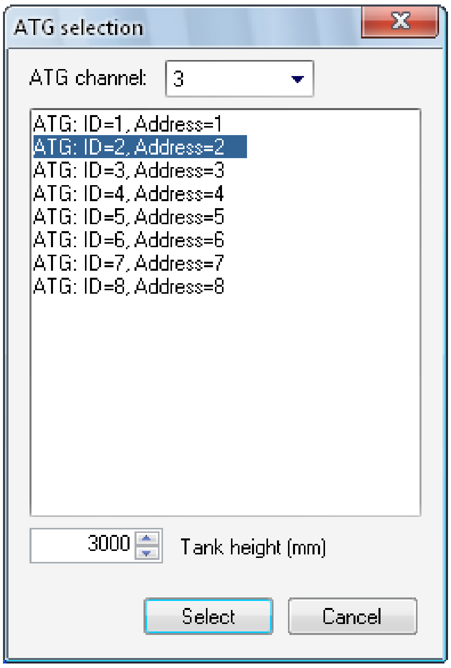


COM 4

```
15:42:00 <<<< (34) StatusResponse 04
```

5421

To view the measurement by the ATGs in PTS application go to main menu item “ATG” → “ATG measurement data”, where for each of the ATG icons in settings set configuration of ATG to get measurement values and also set tank’s height:

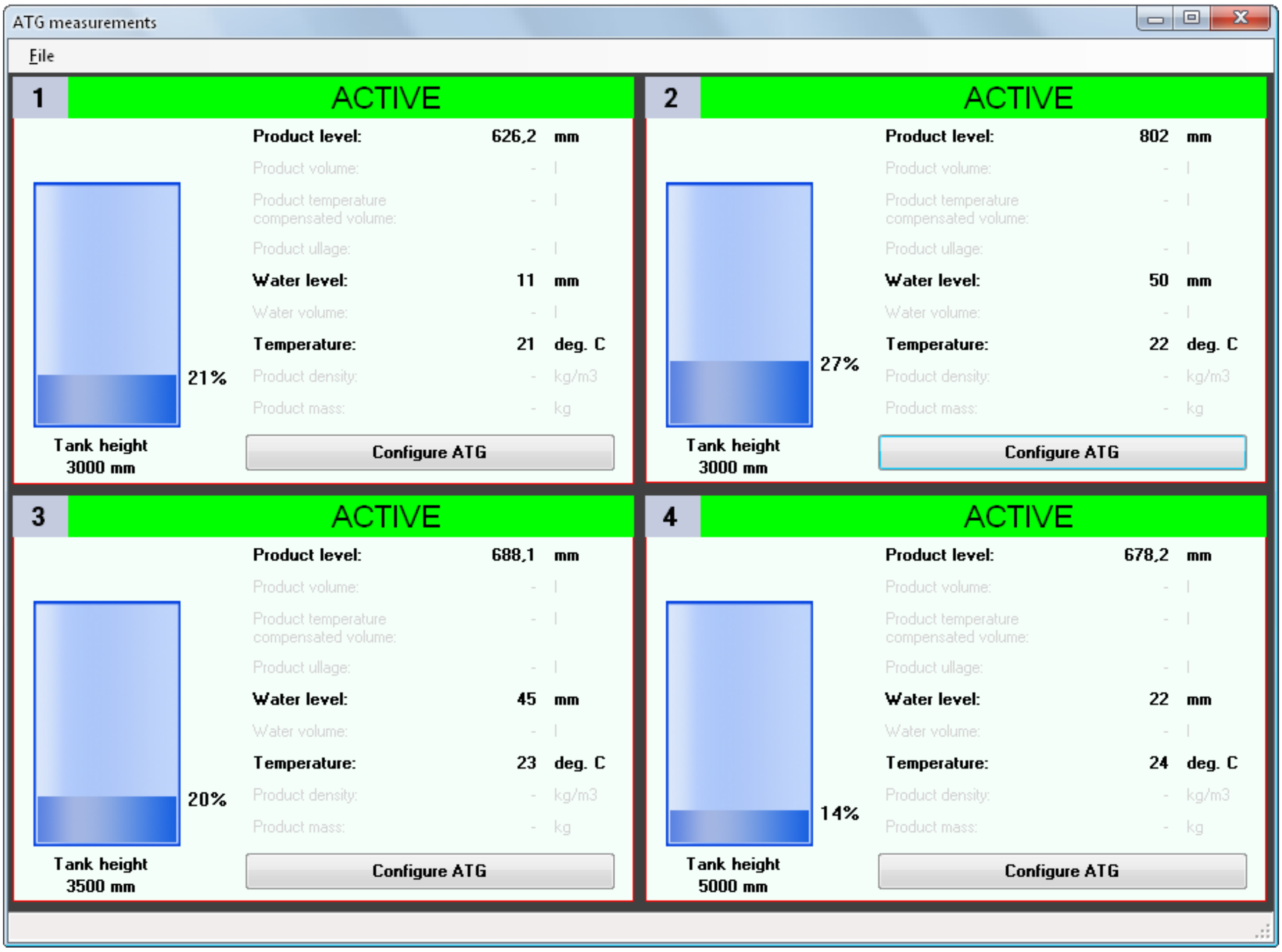


ATG channel: 3

ATG: ID=1, Address=1
ATG: ID=2, Address=2
ATG: ID=3, Address=3
ATG: ID=4, Address=4
ATG: ID=5, Address=5
ATG: ID=6, Address=6
ATG: ID=7, Address=7
ATG: ID=8, Address=8

3000 Tank height (mm)

Now PTS application is able to indicate current level measurement data from ATG system:



**1 ACTIVE**

| | | |
|---|---|---|
| Product level: | 626,2 | mm |
| Product volume: | - | l |
| Product temperature compensated volume: | - | l |
| Product ullage: | - | l |
| Water level: | 11 | mm |
| Water volume: | - | l |
| Temperature: | 21 | deg. C |
| Product density: | - | kg/m3 |
| Product mass: | - | kg |

21%
Tank height
3000 mm

**2 ACTIVE**

| | | |
|---|---|---|
| Product level: | 802 | mm |
| Product volume: | - | l |
| Product temperature compensated volume: | - | l |
| Product ullage: | - | l |
| Water level: | 50 | mm |
| Water volume: | - | l |
| Temperature: | 22 | deg. C |
| Product density: | - | kg/m3 |
| Product mass: | - | kg |

27%
Tank height
3000 mm

**3 ACTIVE**

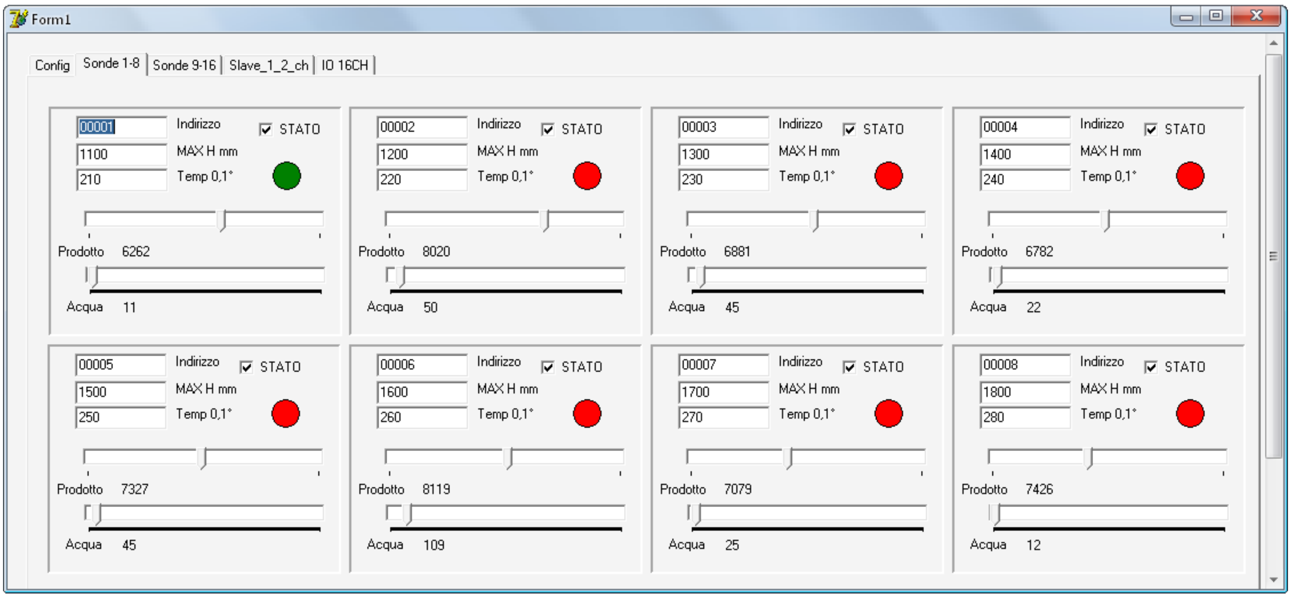
| | | |
|---|---|---|
| Product level: | 688,1 | mm |
| Product volume: | - | l |
| Product temperature compensated volume: | - | l |
| Product ullage: | - | l |
| Water level: | 45 | mm |
| Water volume: | - | l |
| Temperature: | 23 | deg. C |
| Product density: | - | kg/m3 |
| Product mass: | - | kg |

20%
Tank height
3500 mm

**4 ACTIVE**

| | | |
|---|---|---|
| Product level: | 678,2 | mm |
| Product volume: | - | l |
| Product temperature compensated volume: | - | l |
| Product ullage: | - | l |
| Water level: | 22 | mm |
| Water volume: | - | l |
| Temperature: | 24 | deg. C |
| Product density: | - | kg/m3 |
| Product mass: | - | kg |

14%
Tank height
5000 mm



| Indirizzo | MAX H mm | Temp 0,1° | STATO | Prodotto | Acqua |
|---|---|---|---|---|---|
| 00001 | 1100 | 210 | ☑ ● | 6262 | 11 |
| 00002 | 1200 | 220 | ☑ ● | 8020 | 50 |
| 00003 | 1300 | 230 | ☑ ● | 6881 | 45 |
| 00004 | 1400 | 240 | ☑ ● | 6782 | 22 |
| 00005 | 1500 | 250 | ☑ ● | 7327 | 45 |
| 00006 | 1600 | 260 | ☑ ● | 8119 | 109 |
| 00007 | 1700 | 270 | ☑ ● | 7079 | 25 |
| 00008 | 1800 | 280 | ☑ ● | 7426 | 12 |

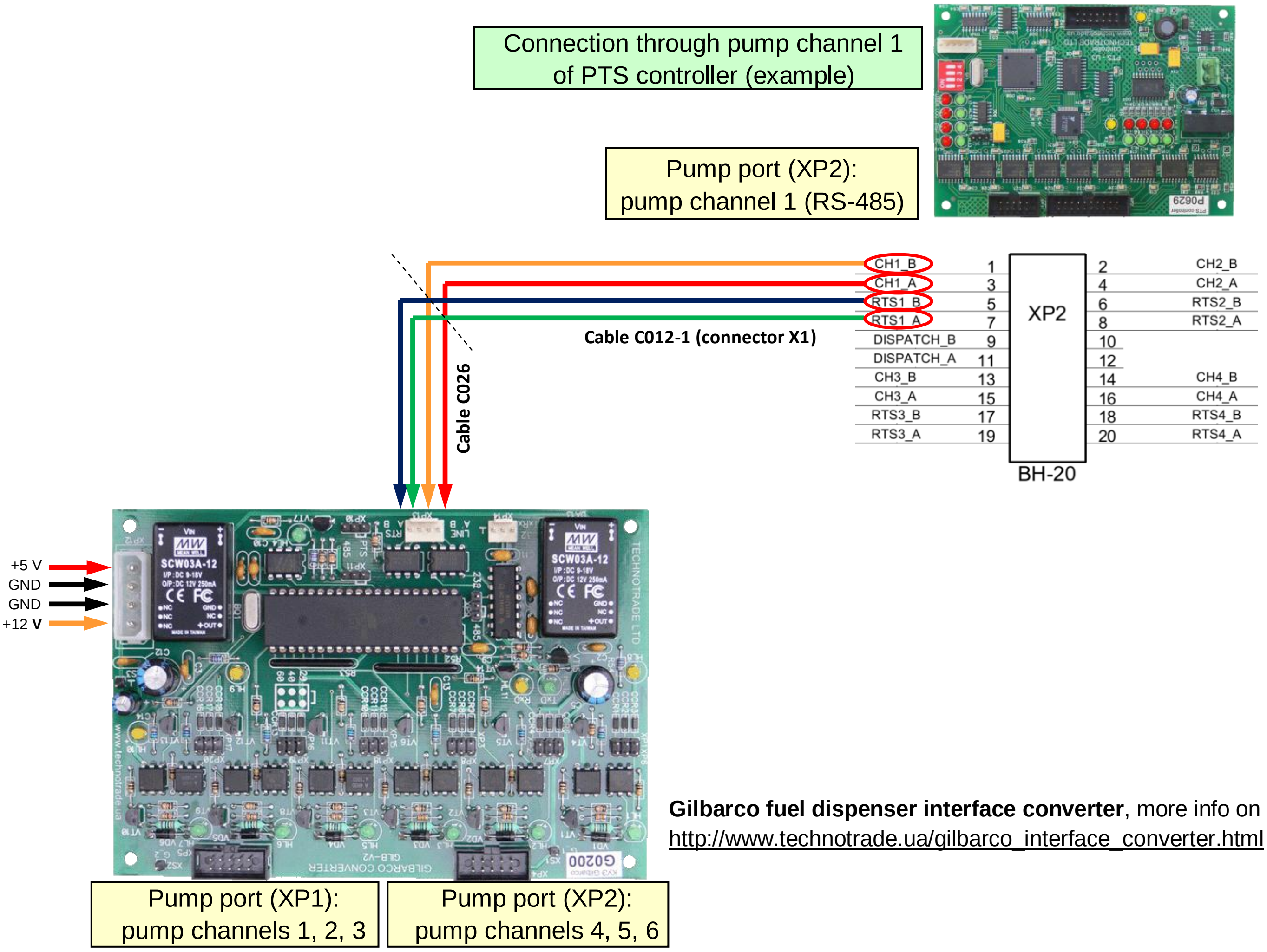
***NOTE!*** In case of necessity to debug operation with PTS controller and there are no software simulators present (or equipment to do this) it is comfortable to use built-in simulators of pumps and ATGs in PTS controller (read more in sections “BUILT-IN PUMP SIMULATOR” and “BUILT-IN ATG PROBE SIMULATOR”.

# EXAMPLES OF FUEL DISPENSERS CONNECTION SCHEMES

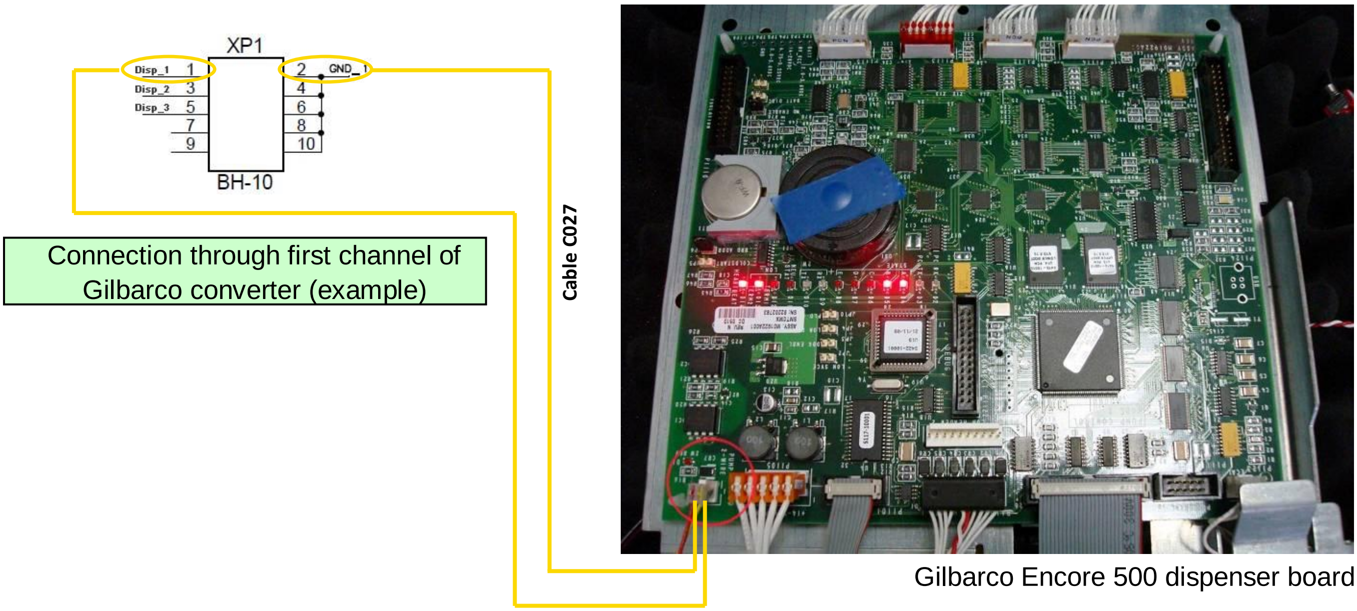
## *Gilbarco dispenser connection scheme*

Connection to Gilbarco dispenser is made through Gilbarco interface converter, which provides connection RS-232/RS-485 interface to Gilbarco 2-wire current loop interface.

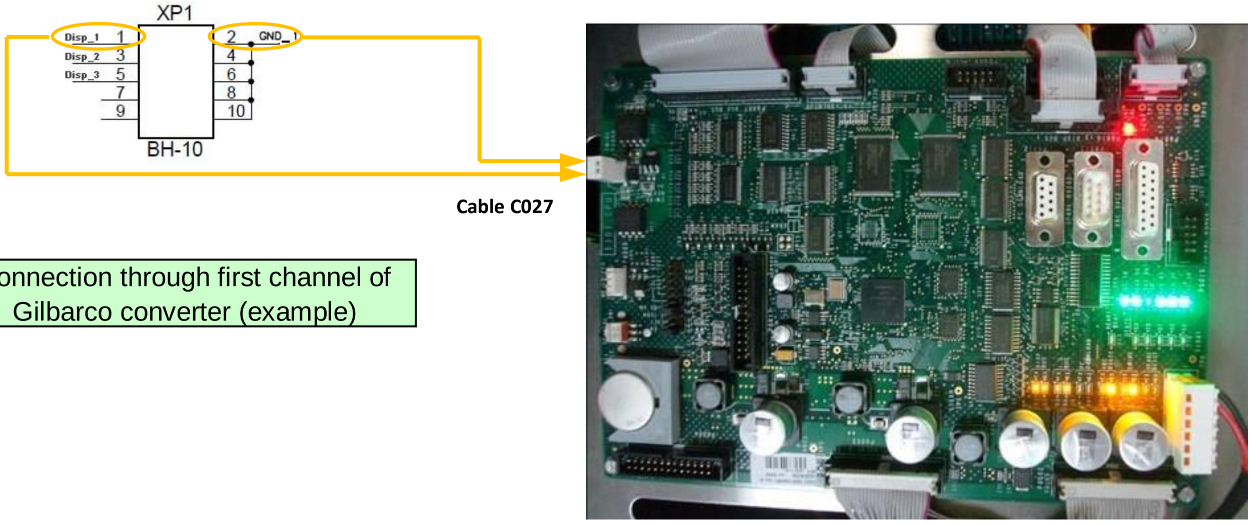
Configuration of PTS controller: protocol **GILBARCO Two-Wire**, baud rate **5787**.


Connection through pump channel 1 of PTS controller (example)
Pump port (XP2): pump channel 1 (RS-485)
CH1_B 1
CH1_A 3
RTS1_B 5
RTS1_A 7
DISPATCH_B 9
DISPATCH_A 11
CH3_B 13
CH3_A 15
RTS3_B 17
RTS3_A 19
XP2
2 CH2_B
4 CH2_A
6 RTS2_B
8 RTS2_A
10
12
14 CH4_B
16 CH4_A
18 RTS4_B
20 RTS4_A
BH-20
Cable C012-1 (connector X1)
Cable C026
+5 V
GND
GND
+12 V
Pump port (XP1): pump channels 1, 2, 3
Pump port (XP2): pump channels 4, 5, 6

**Gilbarco fuel dispenser interface converter**, more info on http://www.technotrade.ua/gilbarco_interface_converter.html

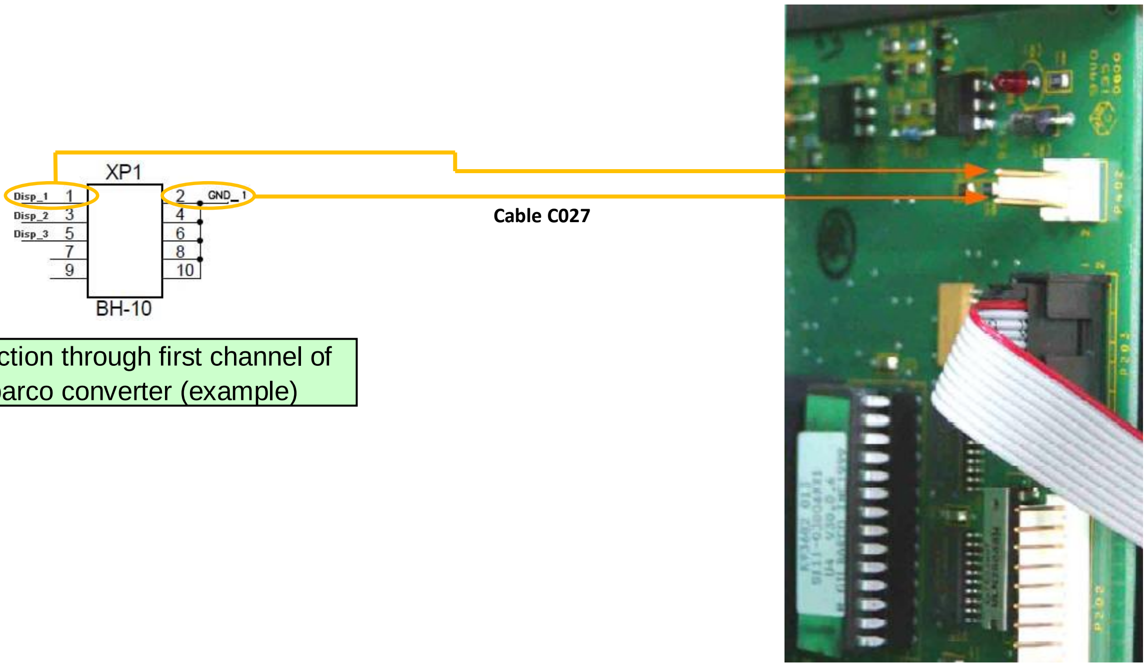

XP1
Disp_1 1
Disp_2 3
Disp_3 5
7
9
2 GND_1
4
6
8
10
BH-10
Connection through first channel of Gilbarco converter (example)
Cable C027

Gilbarco Encore 500 dispenser board



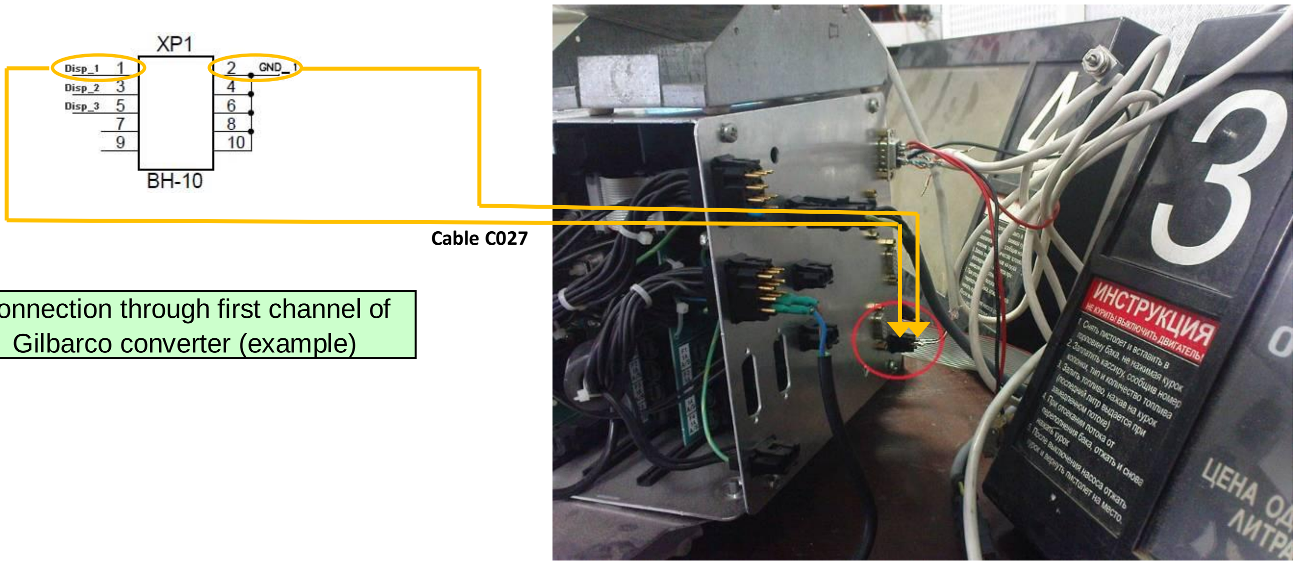
Connection through first channel of Gilbarco converter (example)

Gilbarco dispenser ASSY M06104A001 rev. B board



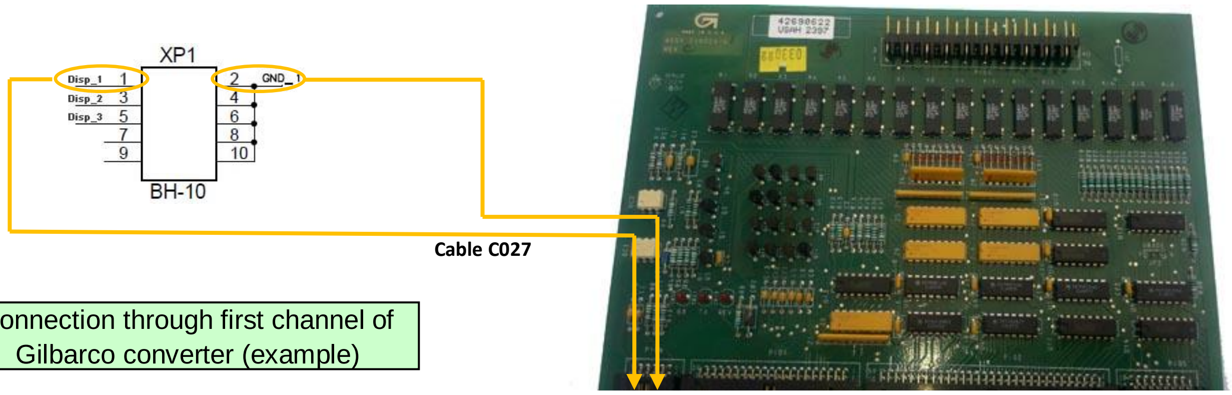
Connection through first channel of Gilbarco converter (example)

Gilbarco dispenser board



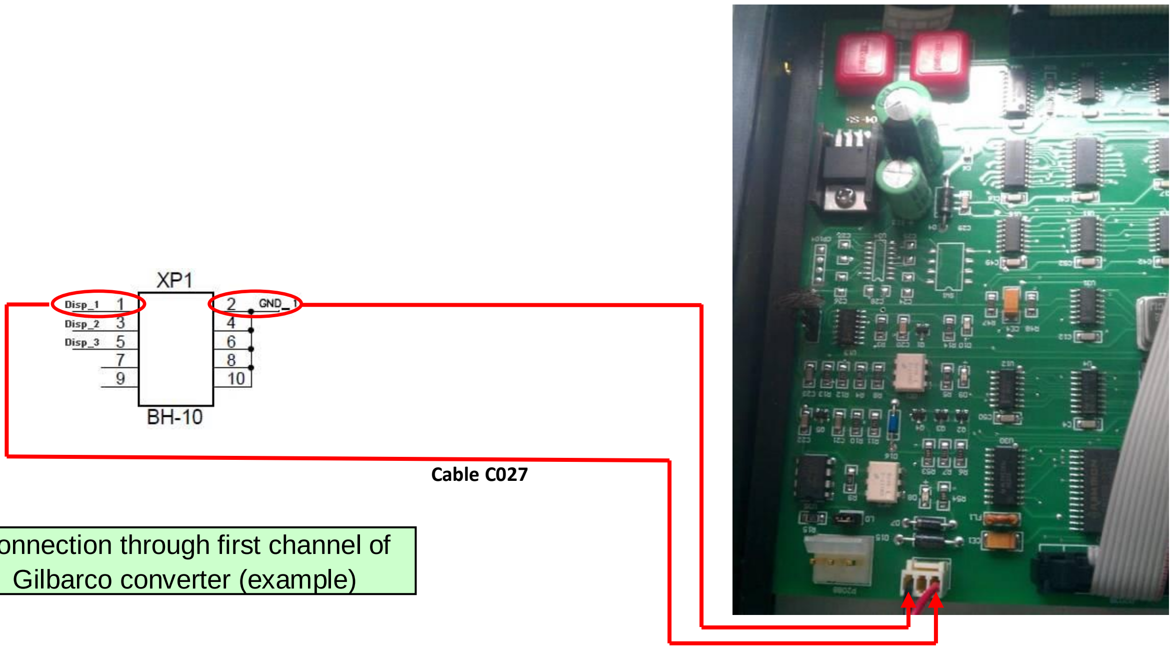
Connection through first channel of Gilbarco converter (example)

Gilbarco Euroline dispenser board



Connection through first channel of Gilbarco converter (example)

Gilbarco Highline / Dimension Assy dispenser board

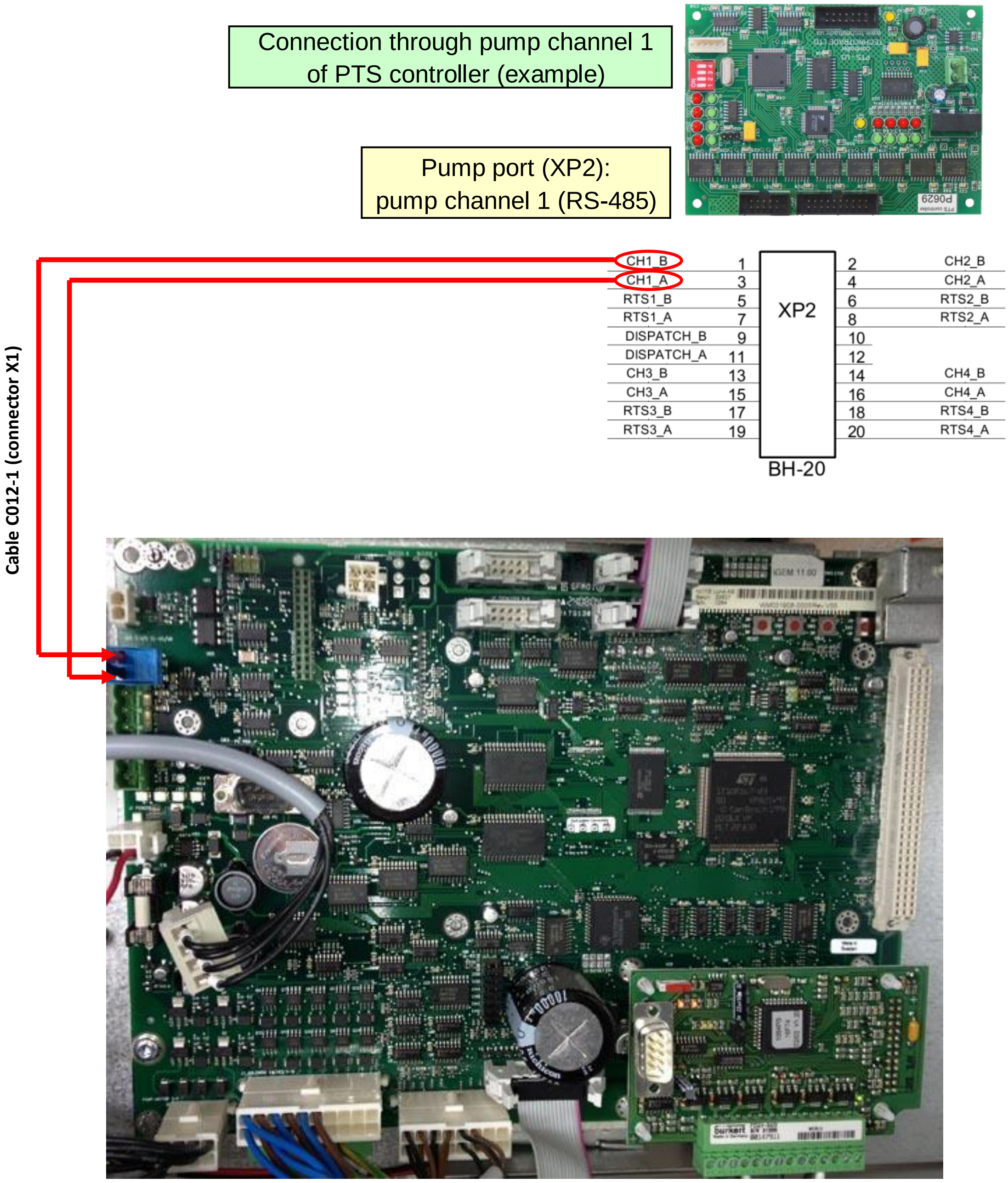


Connection through first channel of Gilbarco converter (example)

Gilbarco Endeavor dispenser board

# *Wayne Dresser dispenser connection scheme (RS-485 interface)*

Connection to Wayne Dresser dispenser is made directly without any interface converter.
Configuration of PTS controller: protocol **WAYNE Dart**, baud rate **9600**.

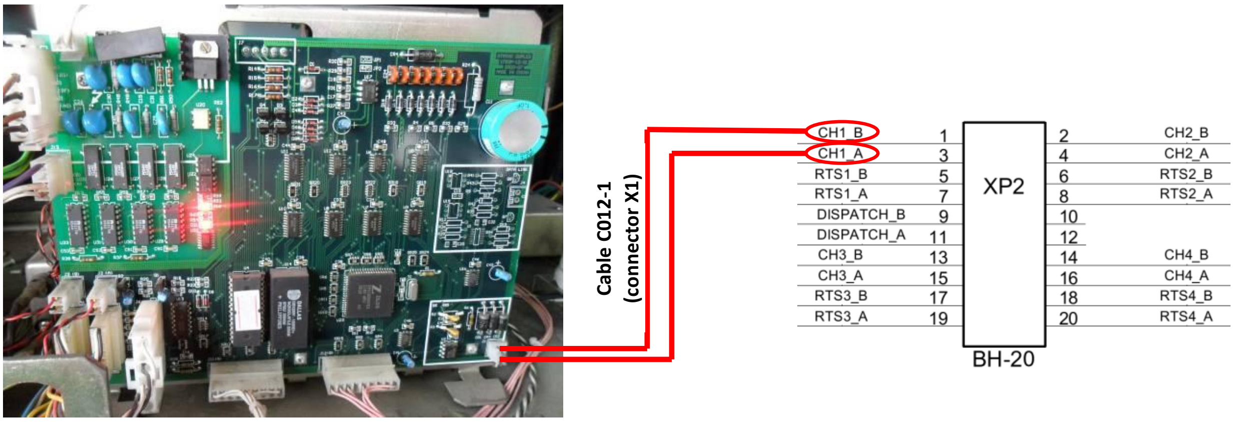


Connection through pump channel 1 of PTS controller (example)

Pump port (XP2): pump channel 1 (RS-485)

| | | | |
|---|---|---|---|
| CH1_B | 1 | 2 | CH2_B |
| CH1_A | 3 | 4 | CH2_A |
| RTS1_B | 5 | 6 | RTS2_B |
| RTS1_A | 7 | 8 | RTS2_A |
| DISPATCH_B | 9 | 10 | |
| DISPATCH_A | 11 | 12 | |
| CH3_B | 13 | 14 | CH4_B |
| CH3_A | 15 | 16 | CH4_A |
| RTS3_B | 17 | 18 | RTS4_B |
| RTS3_A | 19 | 20 | RTS4_A |

Wayne Dresser iGem dispenser board



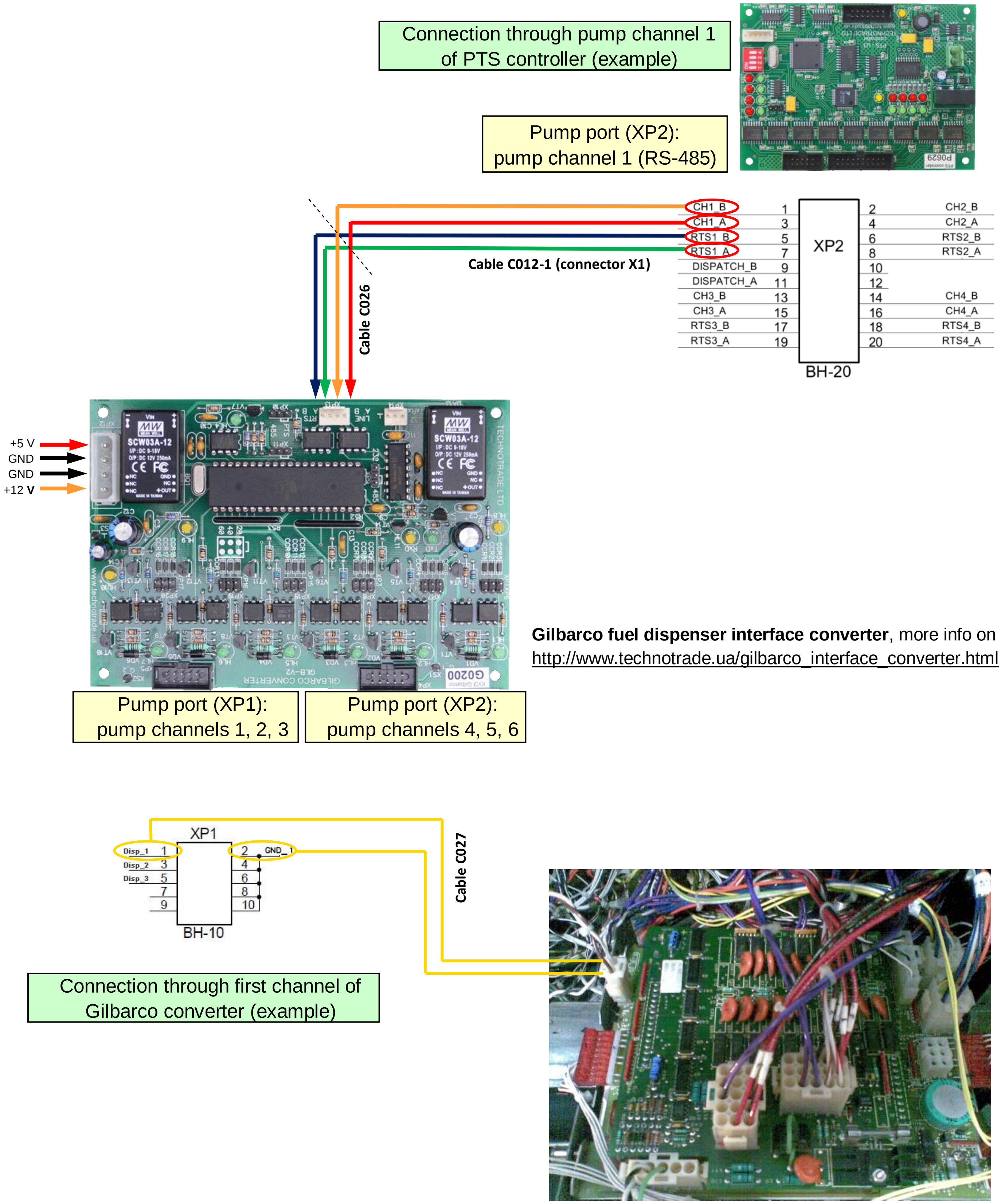
| | | | |
|---|---|---|---|
| CH1_B | 1 | 2 | CH2_B |
| CH1_A | 3 | 4 | CH2_A |
| RTS1_B | 5 | 6 | RTS2_B |
| RTS1_A | 7 | 8 | RTS2_A |
| DISPATCH_B | 9 | 10 | |
| DISPATCH_A | 11 | 12 | |
| CH3_B | 13 | 14 | CH4_B |
| CH3_A | 15 | 16 | CH4_A |
| RTS3_B | 17 | 18 | RTS4_B |
| RTS3_A | 19 | 20 | RTS4_A |

Wayne Dresser dispenser board

## *Wayne Dresser dispenser connection scheme (current loop interface)*

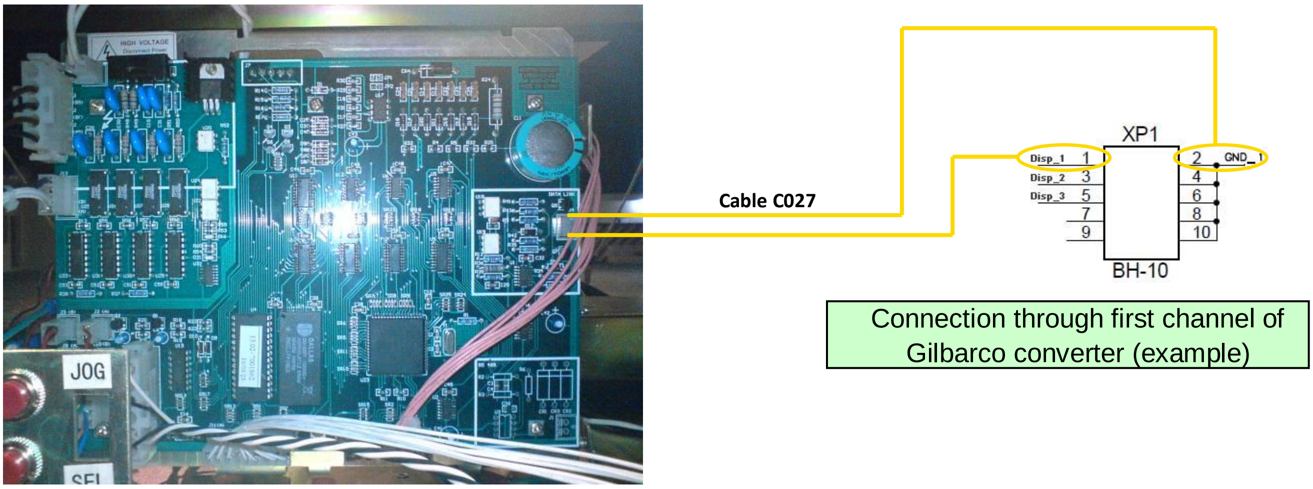
Connection to Gilbarco dispenser is made through Gilbarco interface converter, which provides connection RS-232/RS-485 interface to Gilbarco 2-wire current loop interface.

Configuration of PTS controller: protocol **WAYNE USCL**, baud rate **9600**.


Connection through pump channel 1 of PTS controller (example)
Pump port (XP2): pump channel 1 (RS-485)
CH1_B 1
CH1_A 3
RTS1_B 5
RTS1_A 7
DISPATCH_B 9
DISPATCH_A 11
CH3_B 13
CH3_A 15
RTS3_B 17
RTS3_A 19
XP2
2 CH2_B
4 CH2_A
6 RTS2_B
8 RTS2_A
10
12
14 CH4_B
16 CH4_A
18 RTS4_B
20 RTS4_A
BH-20
Cable C012-1 (connector X1)
Cable C026
+5 V
GND
GND
+12 V
Pump port (XP1): pump channels 1, 2, 3
Pump port (XP2): pump channels 4, 5, 6
XP1
Disp_1 1
Disp_2 3
Disp_3 5
7
9
2 GND_1
4
6
8
10
BH-10
Cable C027
Connection through first channel of Gilbarco converter (example)

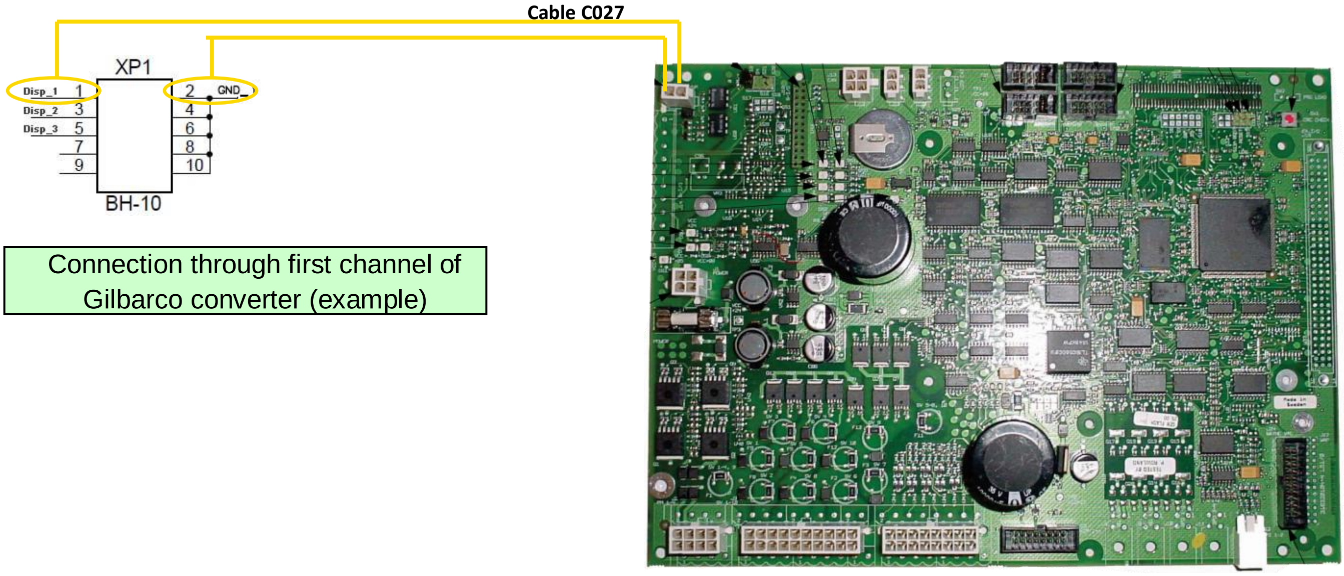
**Gilbarco fuel dispenser interface converter**, more info on http://www.technotrade.ua/gilbarco_interface_converter.html

Wayne Dresser dispenser board



Connection through first channel of Gilbarco converter (example)

Wayne Dresser dispenser board



Connection through first channel of Gilbarco converter (example)

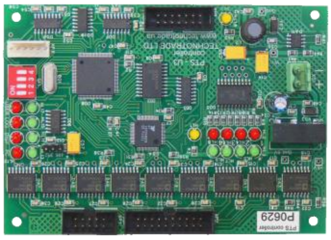
Wayne Dresser dispenser iGEM board

## *TATSUNO (Japan) dispenser connection scheme*

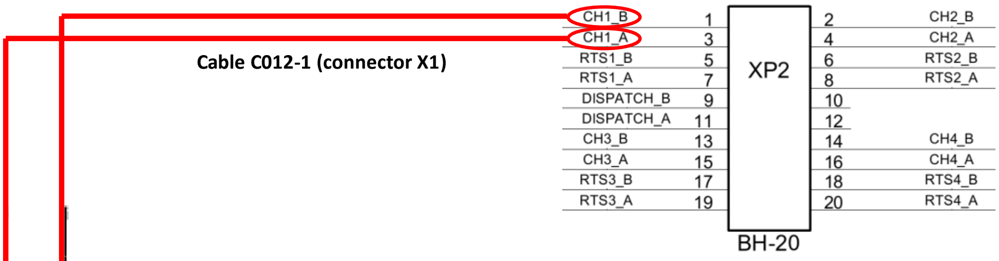
Connection to TATSUNO (Japan) dispenser is made directly without any interface converter.

Configuration of PTS controller: protocol **TATSUNO SS-LAN**, baud rate **19200**.

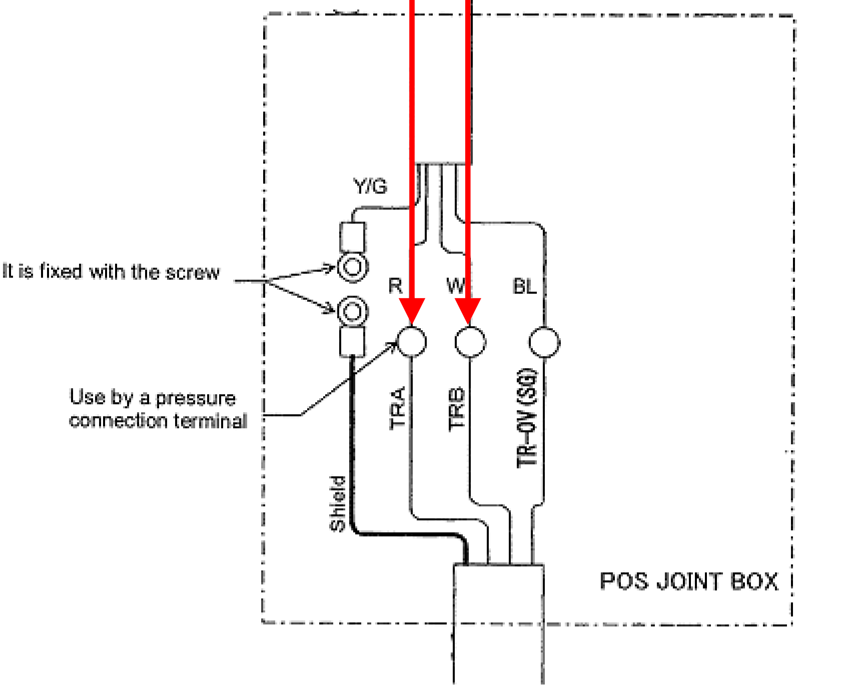
Connection through pump channel 1 of PTS controller (example)

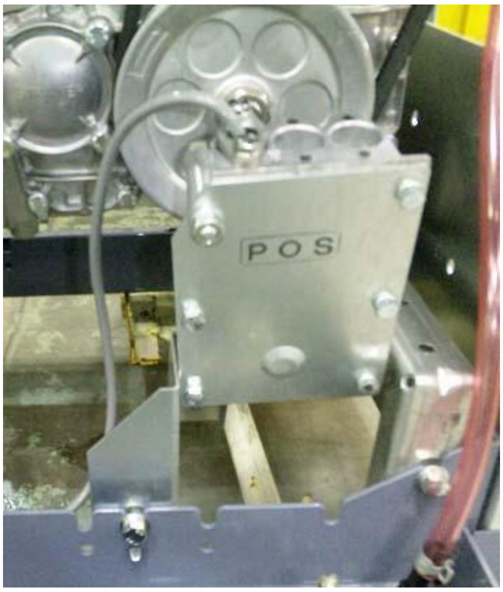


Pump port (XP2): pump channel 1 (RS-485)

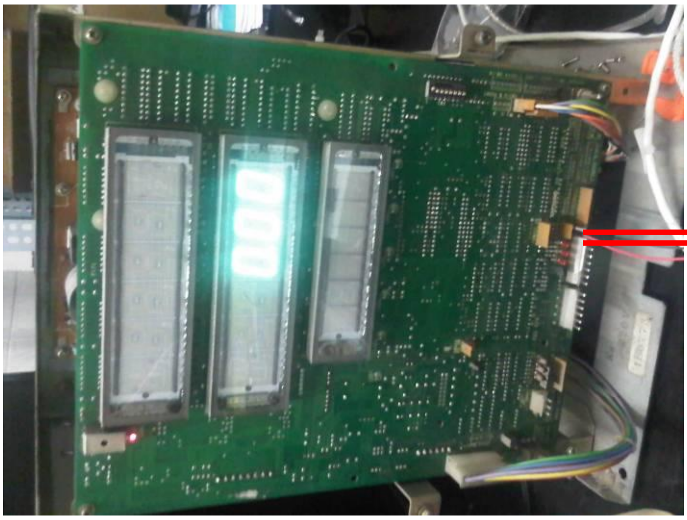


| | | | |
|---|---|---|---|
| CH1_B | 1 | 2 | CH2_B |
| CH1_A | 3 | 4 | CH2_A |
| RTS1_B | 5 | 6 | RTS2_B |
| RTS1_A | 7 | 8 | RTS2_A |
| DISPATCH_B | 9 | 10 | |
| DISPATCH_A | 11 | 12 | |
| CH3_B | 13 | 14 | CH4_B |
| CH3_A | 15 | 16 | CH4_A |
| RTS3_B | 17 | 18 | RTS4_B |
| RTS3_A | 19 | 20 | RTS4_A |

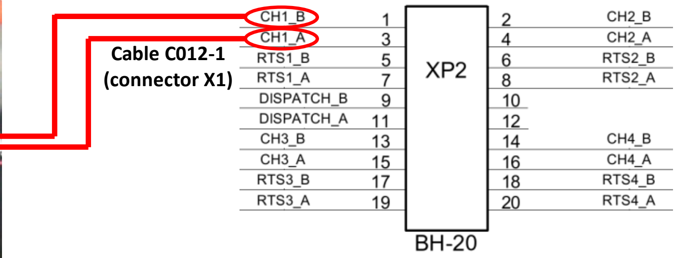

Y/G
It is fixed with the screw
R
W
BL
Use by a pressure connection terminal
TRA
TRB
TR-0V (SG)
Shield
POS JOINT BOX



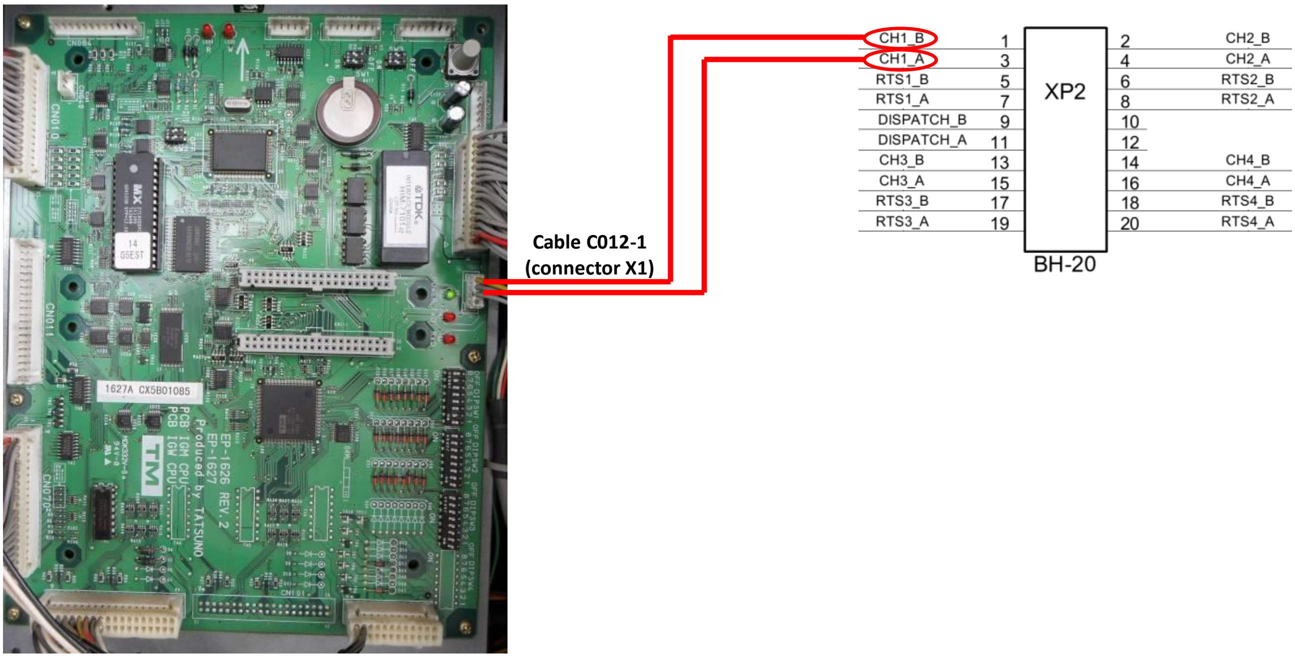
TATSUNO POS joint box



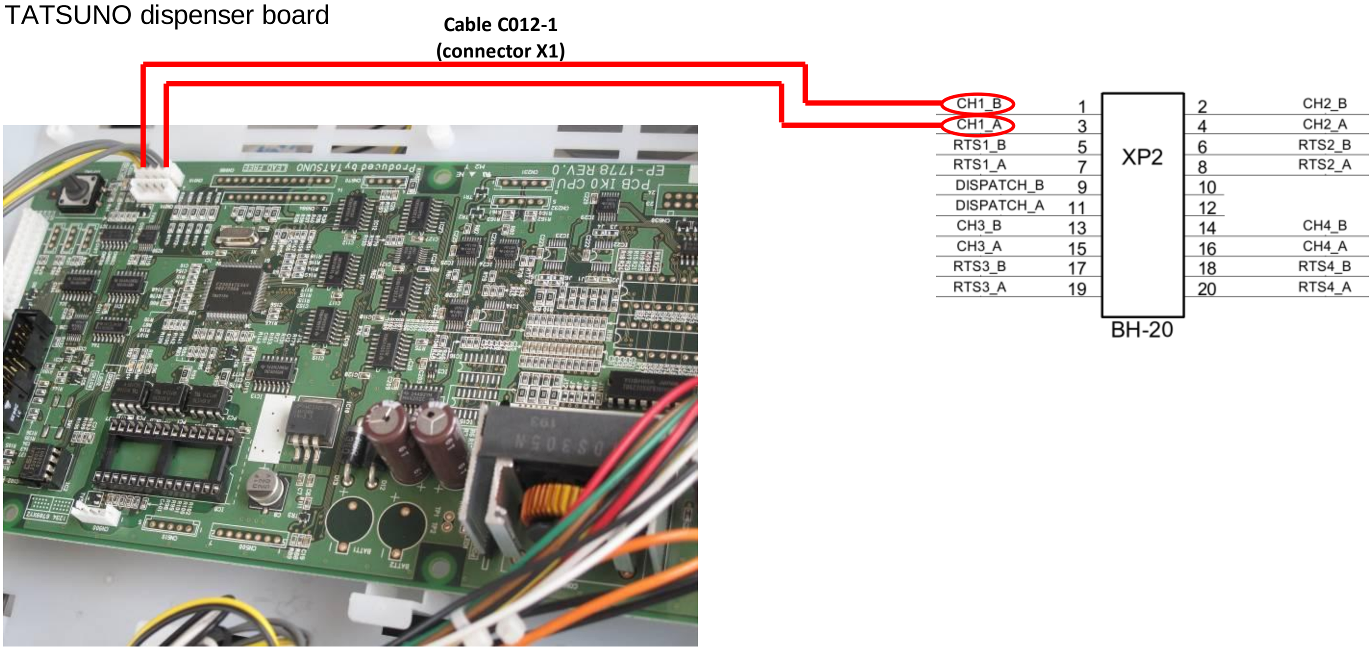
TATSUNO dispenser board



| | | | |
|---|---|---|---|
| CH1_B | 1 | 2 | CH2_B |
| CH1_A | 3 | 4 | CH2_A |
| RTS1_B | 5 | 6 | RTS2_B |
| RTS1_A | 7 | 8 | RTS2_A |
| DISPATCH_B | 9 | 10 | |
| DISPATCH_A | 11 | 12 | |
| CH3_B | 13 | 14 | CH4_B |
| CH3_A | 15 | 16 | CH4_A |
| RTS3_B | 17 | 18 | RTS4_B |
| RTS3_A | 19 | 20 | RTS4_A |



TATSUNO dispenser board

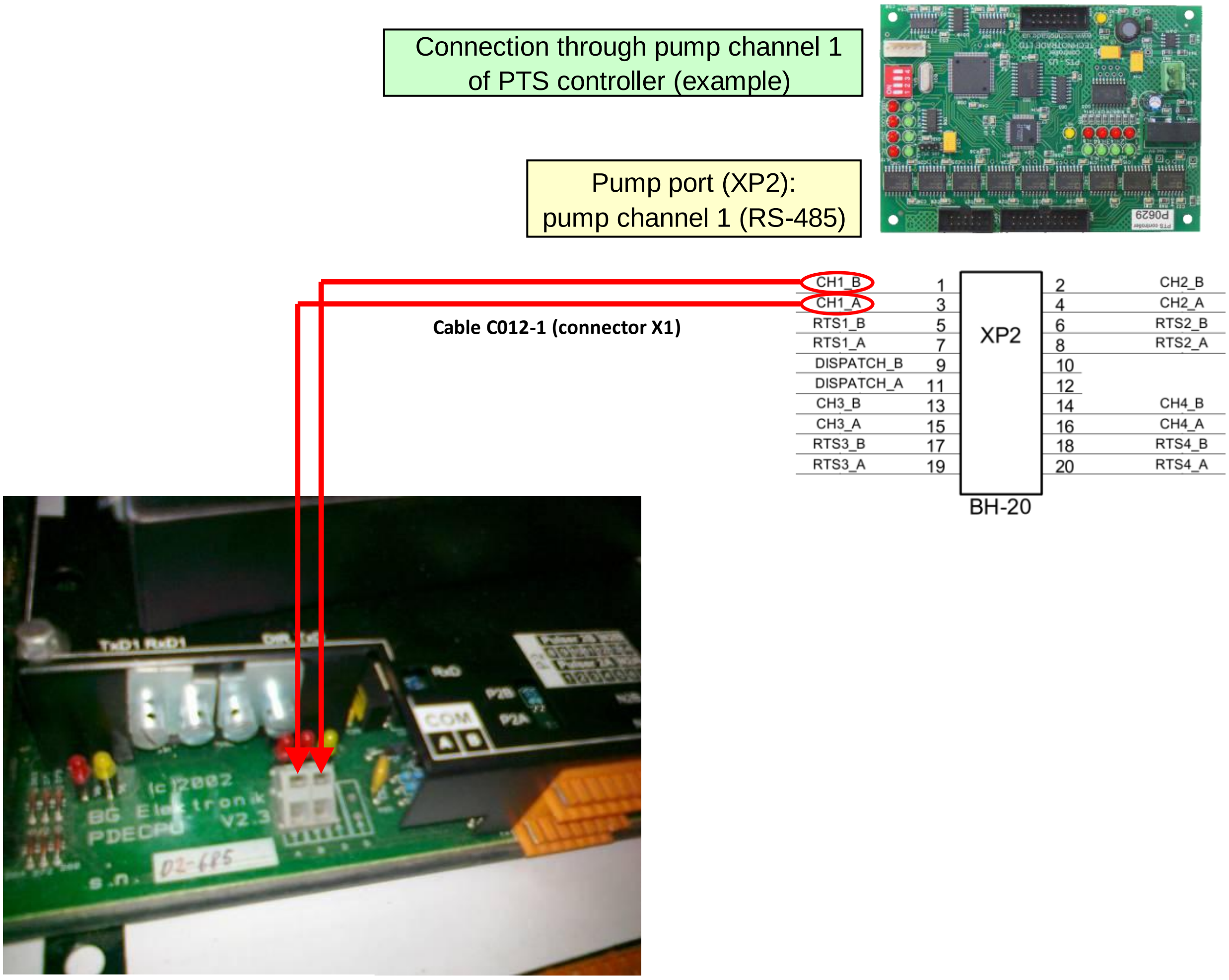


TATSUNO dispenser board

## *TATSUNO Benc dispenser connection scheme*

Connection to TATSUNO Benc dispenser is made directly without any interface converter.

Configuration of PTS controller: protocol **TATSUNO Benc PDE**, baud rate **19200**.



Connection through pump channel 1 of PTS controller (example)

Pump port (XP2): pump channel 1 (RS-485)

**Cable C012-1 (connector X1)**

XP2

| | | | |
|---|---|---|---|
| CH1_B | 1 | 2 | CH2_B |
| CH1_A | 3 | 4 | CH2_A |
| RTS1_B | 5 | 6 | RTS2_B |
| RTS1_A | 7 | 8 | RTS2_A |
| DISPATCH_B | 9 | 10 | |
| DISPATCH_A | 11 | 12 | |
| CH3_B | 13 | 14 | CH4_B |
| CH3_A | 15 | 16 | CH4_A |
| RTS3_B | 17 | 18 | RTS4_B |
| RTS3_A | 19 | 20 | RTS4_A |

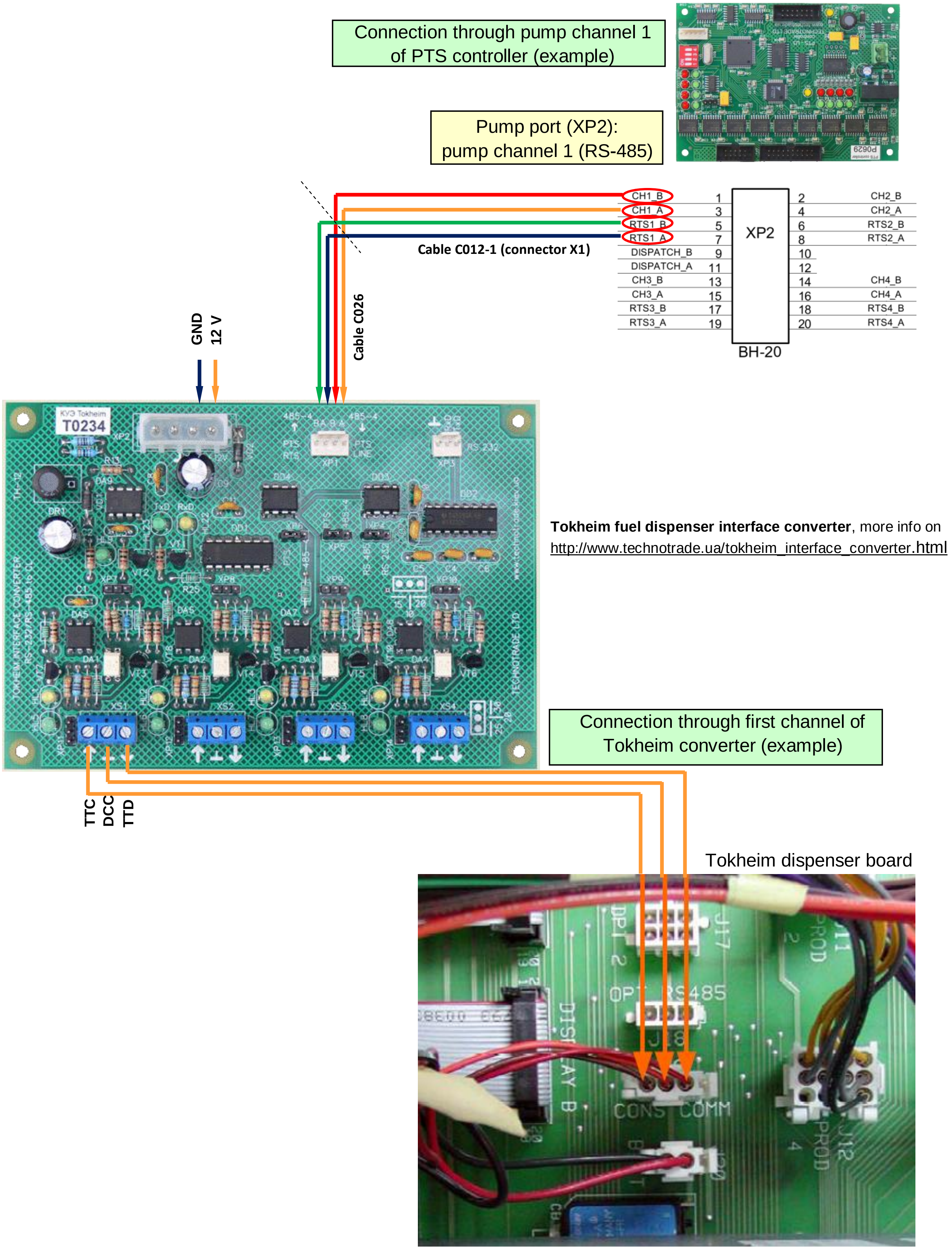
BH-20

TATSUNO Benc pumphead

## *Tokheim dispenser connection scheme*

Connection to Gilbarco dispenser is made through Gilbarco interface converter, which provides connection RS-232/RS-485 interface to Gilbarco 2-wire current loop interface.

Configuration of PTS controller: protocol **TOKHEIM**, baud rate **9600**.



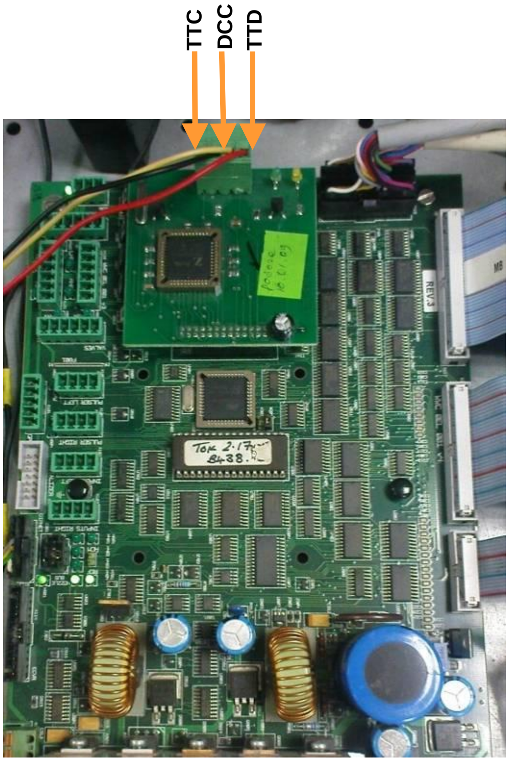
Connection through pump channel 1 of PTS controller (example)

Pump port (XP2): pump channel 1 (RS-485)

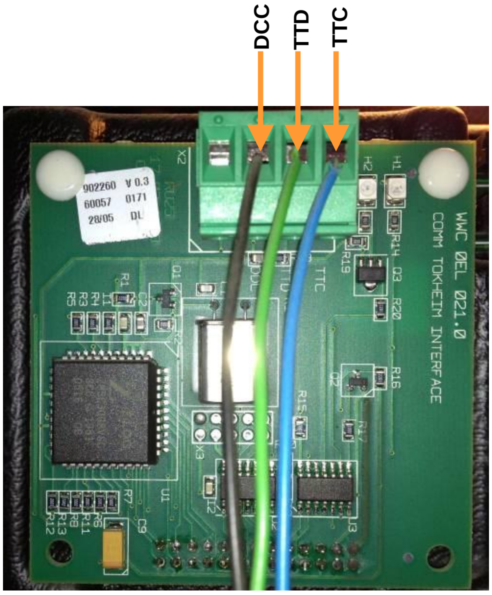
**Tokheim fuel dispenser interface converter**, more info on http://www.technotrade.ua/tokheim_interface_converter.html

Connection through first channel of Tokheim converter (example)

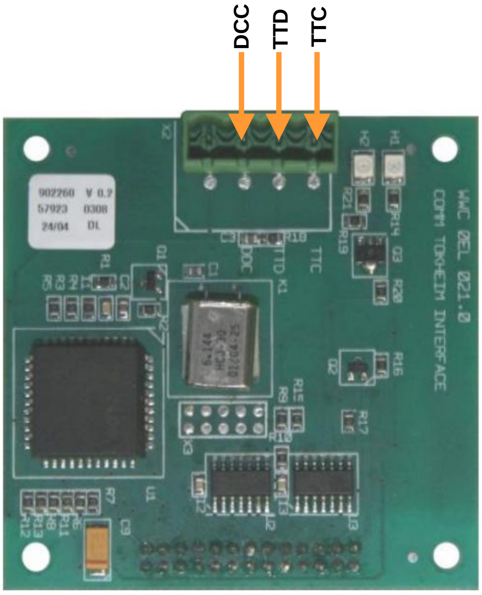
Tokheim dispenser board



Tokheim dispenser board



Tokheim dispenser interface board



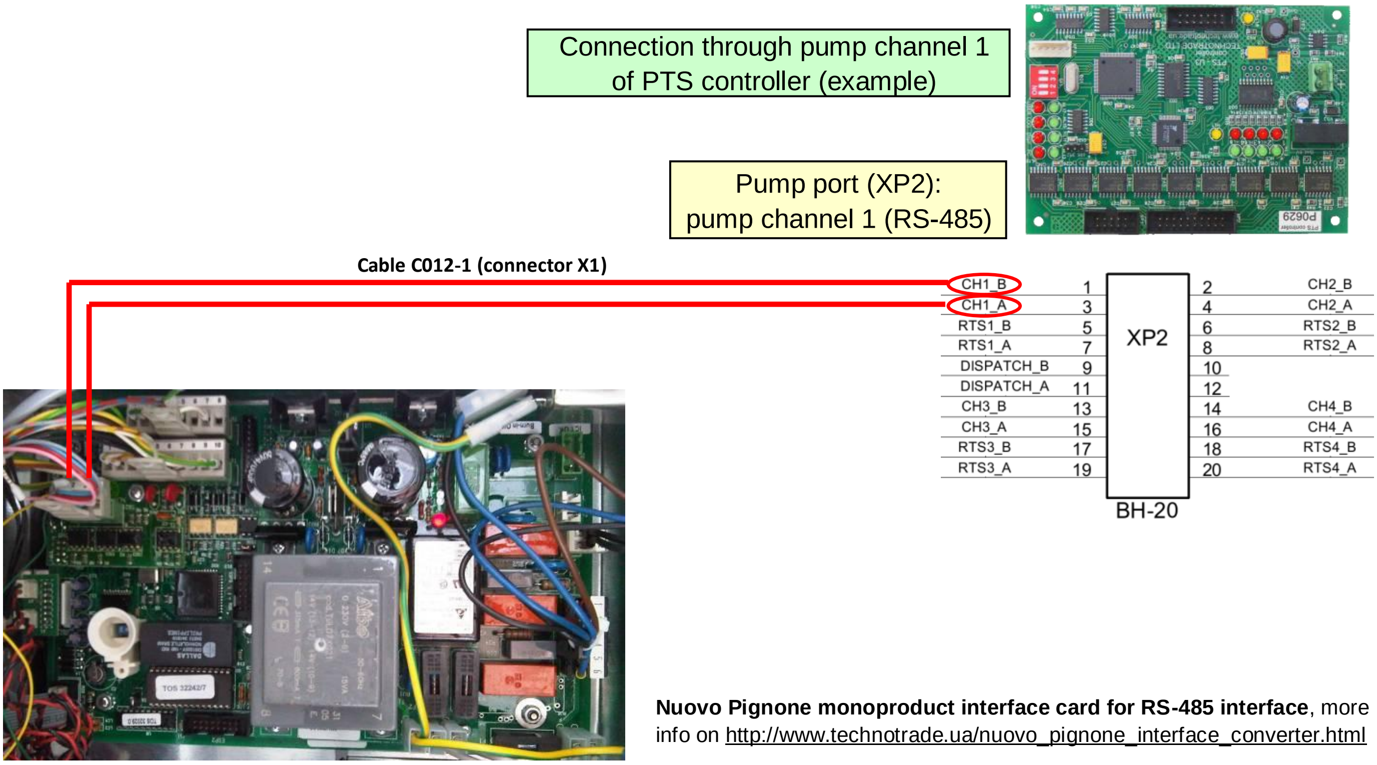
Tokheim dispenser interface board

# *Nuovo Pignone dispenser connection scheme (RS-485 interface)*

Connection to Nuovo Pignone dispensers with RS-485 interface is made directly without any interface converter. At this Nuovo Pignone dispenser should have an interface card for RS-485 interface and should be adjusted to have Wayne Dart protocol.

Configuration of PTS controller: protocol **WAYNE Dart**, baud rate **9600**.

## Nuovo Pignone monoproduct dispenser



Connection through pump channel 1 of PTS controller (example)

Pump port (XP2): pump channel 1 (RS-485)

**Cable C012-1 (connector X1)**

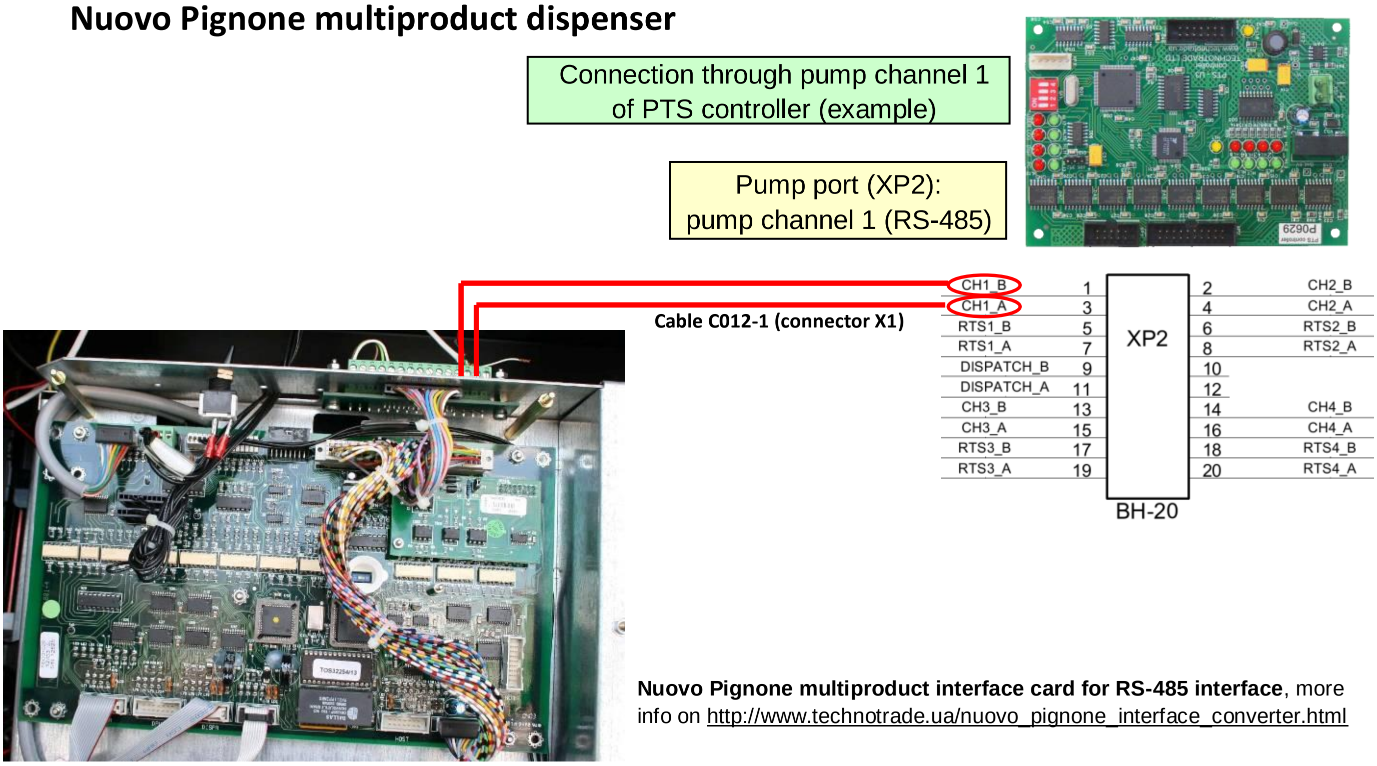
| | | XP2 | | |
|---|---|---|---|---|
| CH1_B | 1 | | 2 | CH2_B |
| CH1_A | 3 | | 4 | CH2_A |
| RTS1_B | 5 | | 6 | RTS2_B |
| RTS1_A | 7 | | 8 | RTS2_A |
| DISPATCH_B | 9 | | 10 | |
| DISPATCH_A | 11 | | 12 | |
| CH3_B | 13 | | 14 | CH4_B |
| CH3_A | 15 | | 16 | CH4_A |
| RTS3_B | 17 | | 18 | RTS4_B |
| RTS3_A | 19 | | 20 | RTS4_A |

BH-20

**Nuovo Pignone monoproduct interface card for RS-485 interface**, more info on http://www.technotrade.ua/nuovo_pignone_interface_converter.html

Nuovo Pignone monoproduct dispenser board

## Nuovo Pignone multiproduct dispenser



Connection through pump channel 1 of PTS controller (example)

Pump port (XP2): pump channel 1 (RS-485)

**Cable C012-1 (connector X1)**

| | | XP2 | | |
|---|---|---|---|---|
| CH1_B | 1 | | 2 | CH2_B |
| CH1_A | 3 | | 4 | CH2_A |
| RTS1_B | 5 | | 6 | RTS2_B |
| RTS1_A | 7 | | 8 | RTS2_A |
| DISPATCH_B | 9 | | 10 | |
| DISPATCH_A | 11 | | 12 | |
| CH3_B | 13 | | 14 | CH4_B |
| CH3_A | 15 | | 16 | CH4_A |
| RTS3_B | 17 | | 18 | RTS4_B |
| RTS3_A | 19 | | 20 | RTS4_A |

BH-20

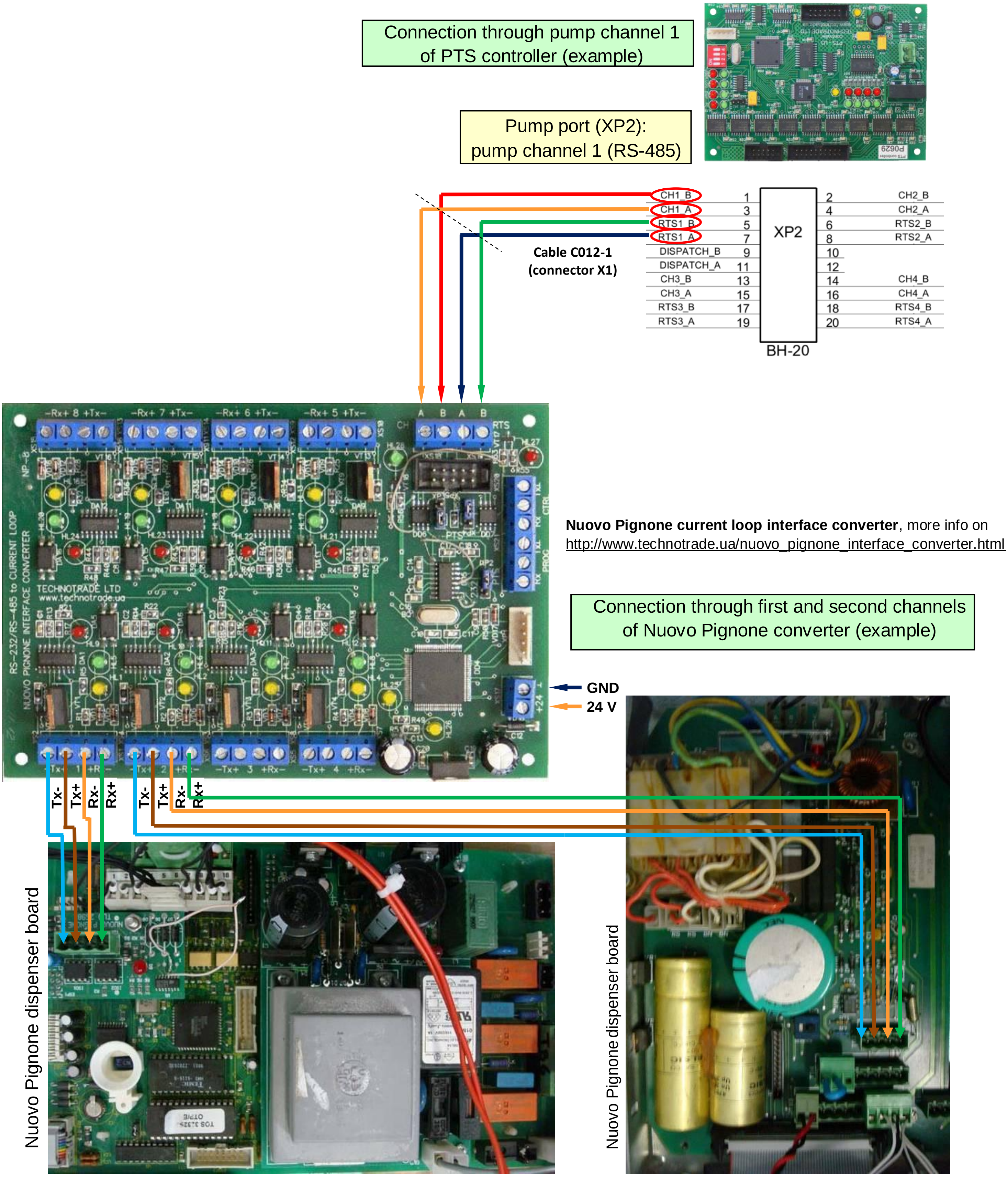
**Nuovo Pignone multiproduct interface card for RS-485 interface**, more info on http://www.technotrade.ua/nuovo_pignone_interface_converter.html

Nuovo Pignone multiproduct dispenser board

## *Nuovo Pignone dispenser connection scheme (4-wire current loop interface)*

Connection to Nuovo Pignone dispensers with 4-wire current loop interface is made through uovo Pignone current loop interface converter, which provides connection of RS-232/RS-485 interface to Nuovo Pignone 4-wire interface.

Configuration of PTS controller: protocol **NUOVO PIGNONE CL**, baud rate **2400**.



Connection through pump channel 1 of PTS controller (example)

Pump port (XP2): pump channel 1 (RS-485)

Cable C012-1 (connector X1)

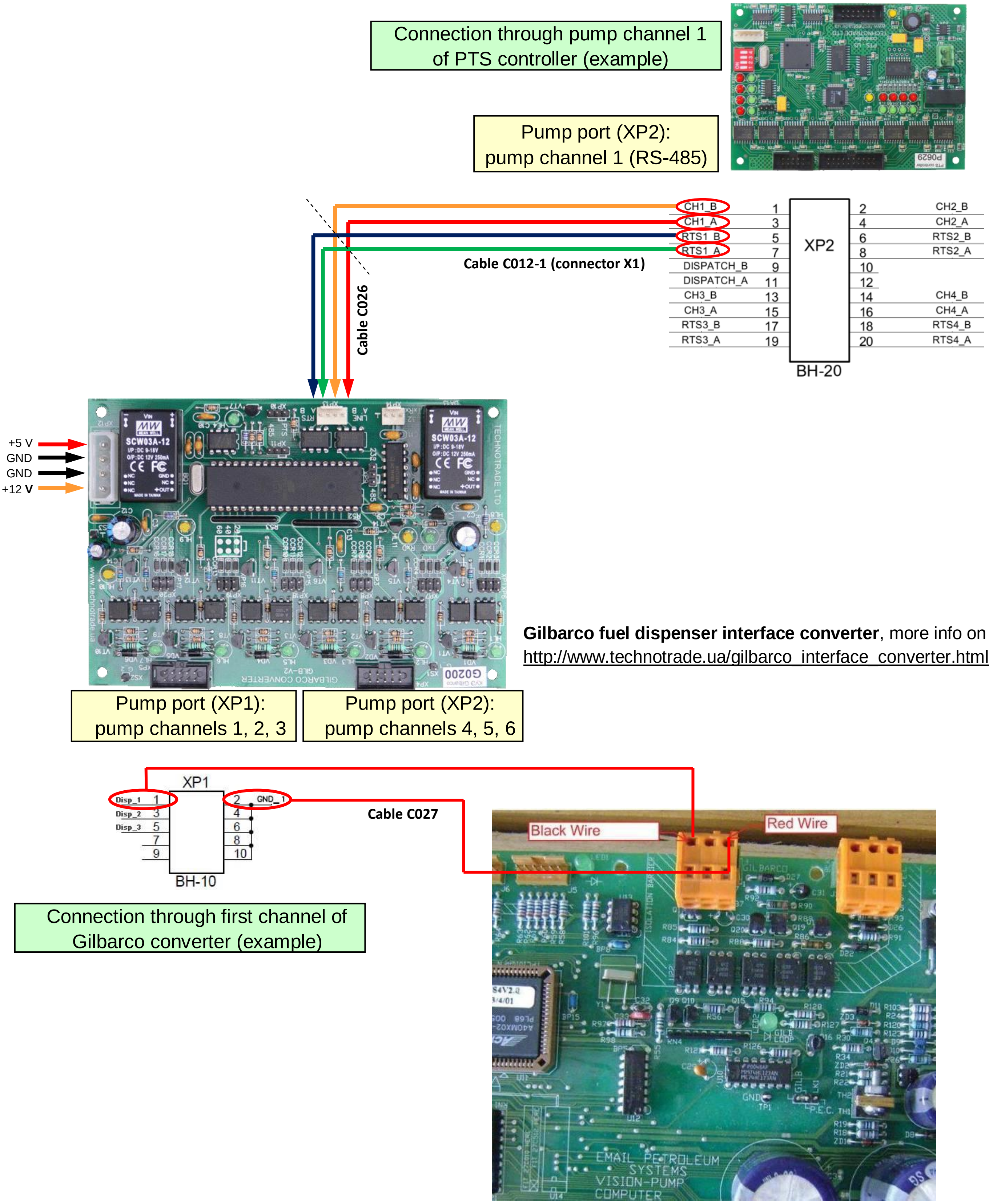
**Nuovo Pignone current loop interface converter**, more info on http://www.technotrade.ua/nuovo_pignone_interface_converter.html

Connection through first and second channels of Nuovo Pignone converter (example)

## *Batchen Email dispenser connection scheme*

Connection to Batchen dispenser is made through Gilbarco interface converter, which provides connection RS-232/RS-485 interface to Gilbarco 2-wire current loop interface.

Configuration of PTS controller: protocol **BATCHEN Email**, baud rate **4800**.



Connection through pump channel 1 of PTS controller (example)

Pump port (XP2): pump channel 1 (RS-485)

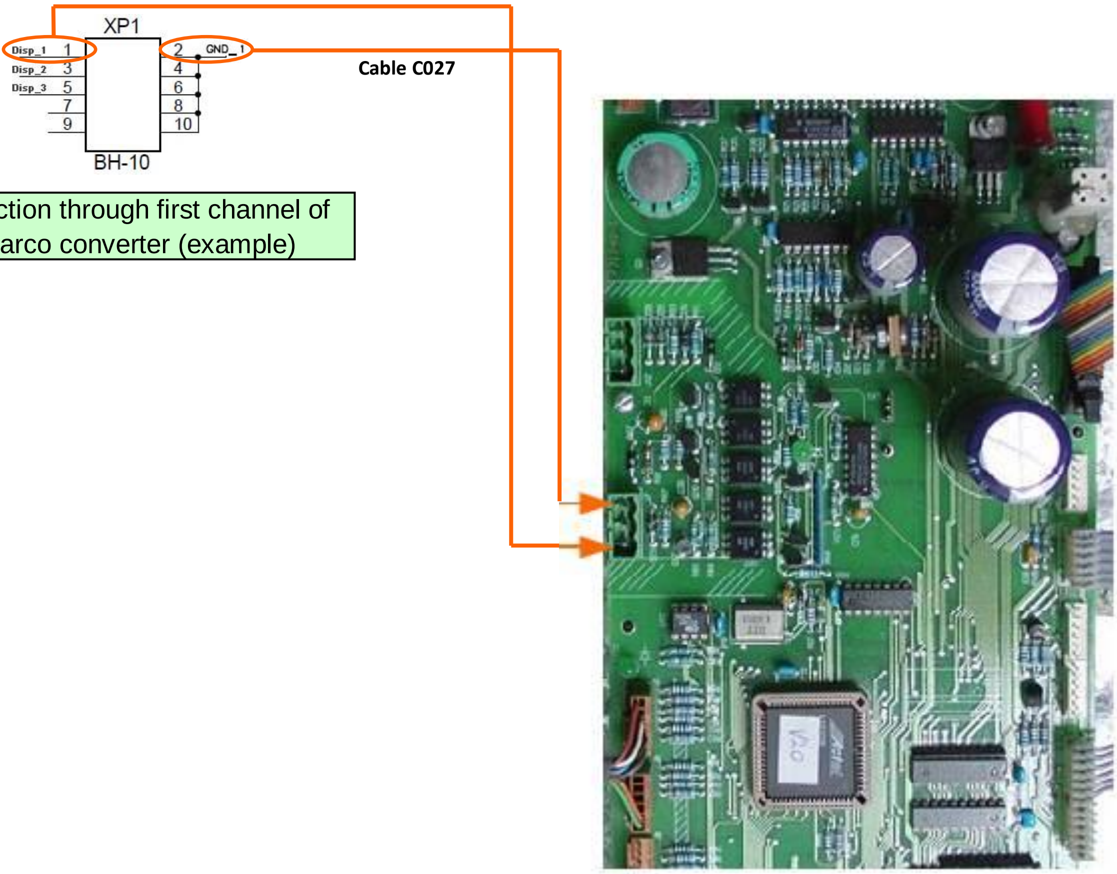
Pump port (XP1): pump channels 1, 2, 3

Pump port (XP2): pump channels 4, 5, 6

Connection through first channel of Gilbarco converter (example)

**Gilbarco fuel dispenser interface converter**, more info on http://www.technotrade.ua/gilbarco_interface_converter.html

Batchen dispenser board

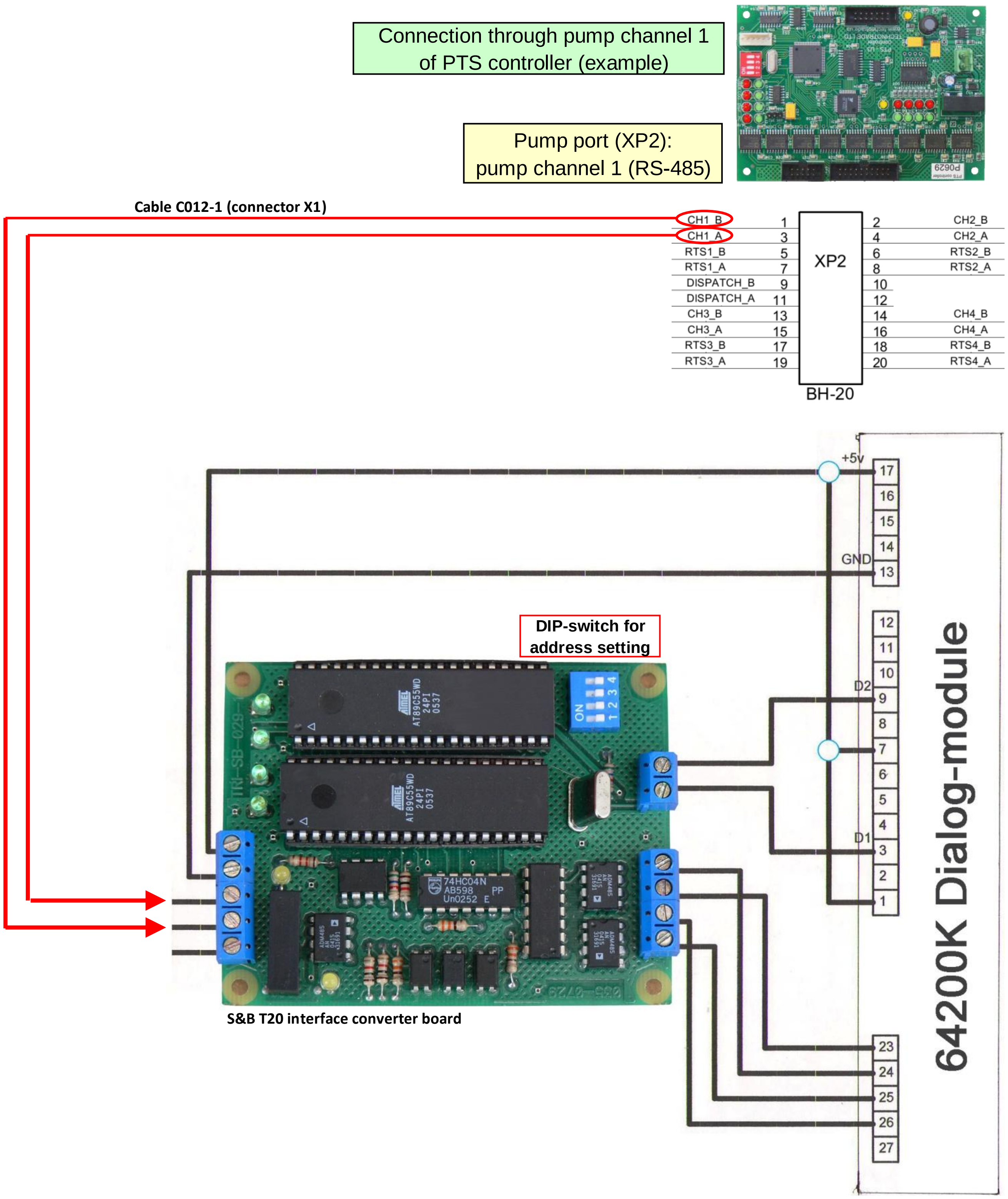


Connection through first channel of Gilbarco converter (example)

Batchen dispenser board

## *Scheidt & Bachmann T20 dispenser connection scheme*

Connection to Scheidt&Bachmann T20 dispenser is made through a S&B T20 interface converter board.
Configuration of PTS controller: protocol **UNIPUMP**, baud rate **9600**



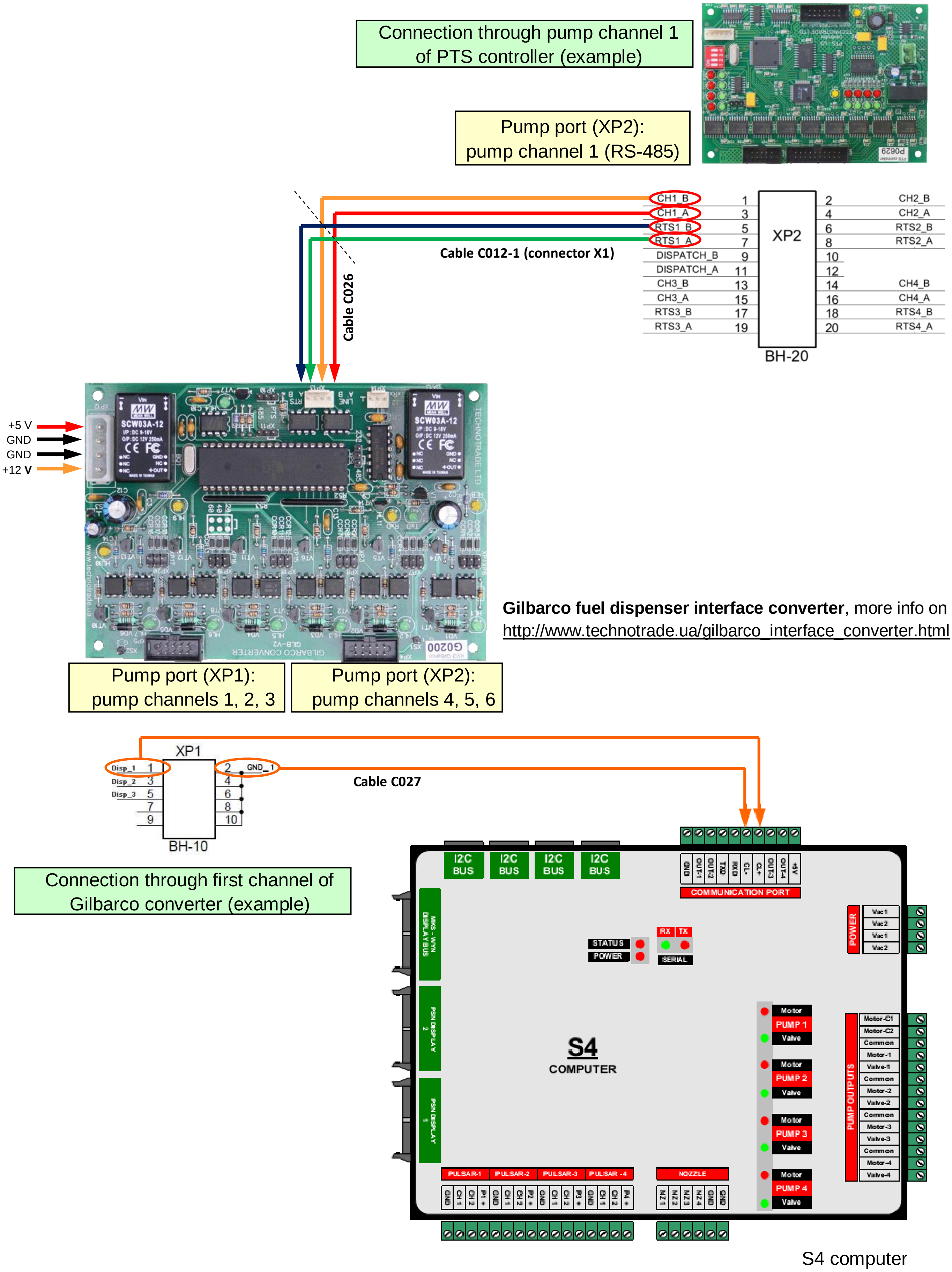
Address setting on S&B T20 interface converter board using a DIP switch

| Address | 1 & 2 | 3 & 4 | 5 & 6 | 7 & 8 | 9 & 10 | 11 & 12 | 13 & 14 | 15 & 16 |
|---|---|---|---|---|---|---|---|---|
| **DIP 1** | OFF | OFF | OFF | OFF | ON | ON | ON | ON |
| **DIP 2** | OFF | OFF | ON | ON | OFF | OFF | ON | ON |
| **DIP 3** | OFF | ON | OFF | ON | OFF | ON | OFF | ON |
| **DIP 4** | OFF | OFF | OFF | OFF | OFF | OFF | OFF | OFF |

## *PETPOSAN-S4 / MEKSAN-S4 / EUROPUMP-S4 dispensers connection scheme*

Connection to S4 computer is made through Gilbarco interface converter, which provides connection RS-232/RS-485 interface to Gilbarco 2-wire current loop interface.

Configuration of PTS controller: protocol **DART Simplex**, baud rate **9600**.



Connection through pump channel 1 of PTS controller (example)

Pump port (XP2): pump channel 1 (RS-485)

**Gilbarco fuel dispenser interface converter**, more info on http://www.technotrade.ua/gilbarco_interface_converter.html

Pump port (XP1): pump channels 1, 2, 3

Pump port (XP2): pump channels 4, 5, 6

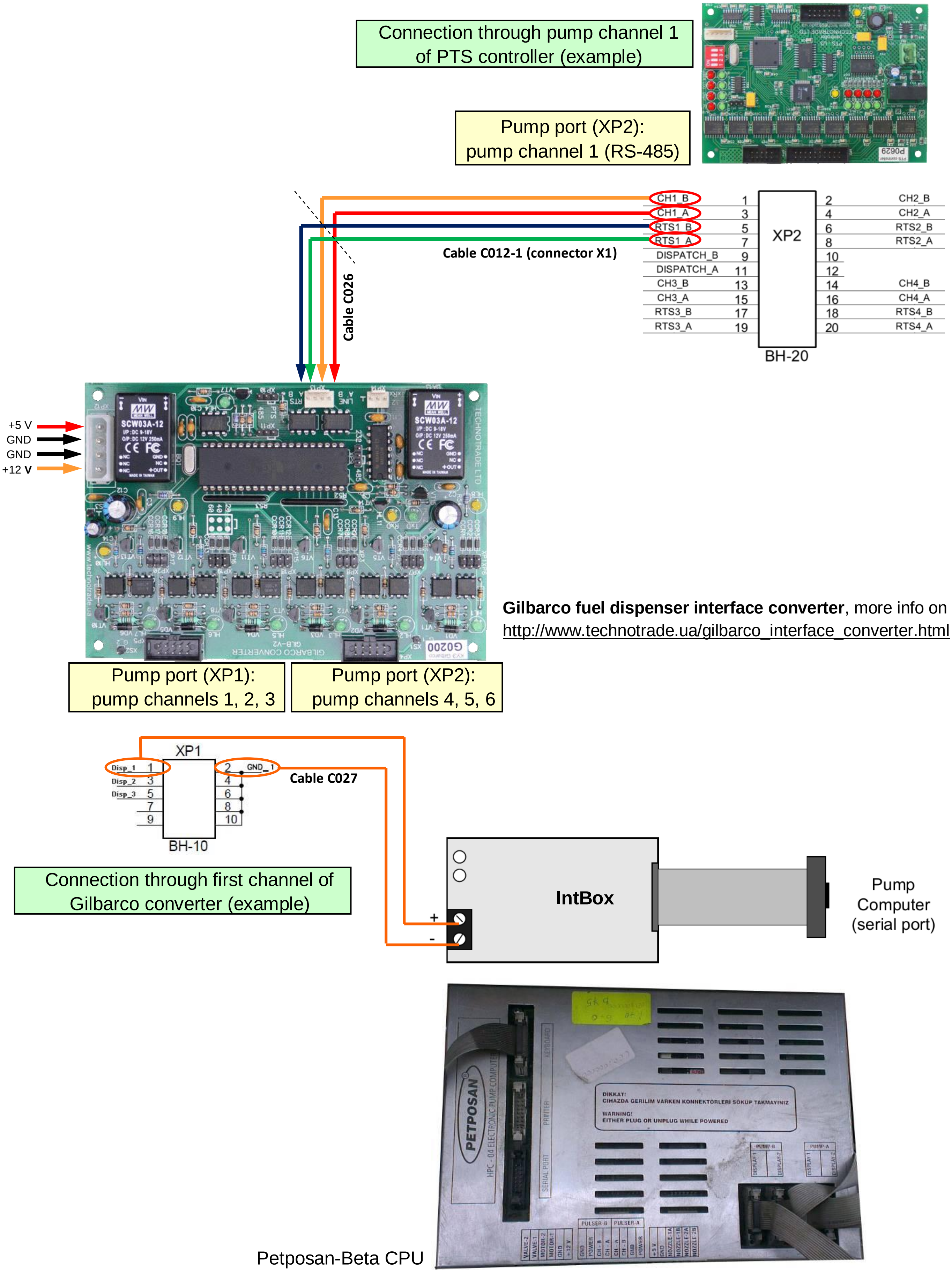
Connection through first channel of Gilbarco converter (example)

S4 computer

## *PETPOSAN-Beta / EUROPUMP-Beta dispensers connection scheme*

Connection to Beta computer is made through Gilbarco interface converter, which provides connection RS-232/RS-485 interface to Gilbarco 2-wire current loop interface.

Configuration of PTS controller: protocol **DART Simplex**, baud rate **9600**.

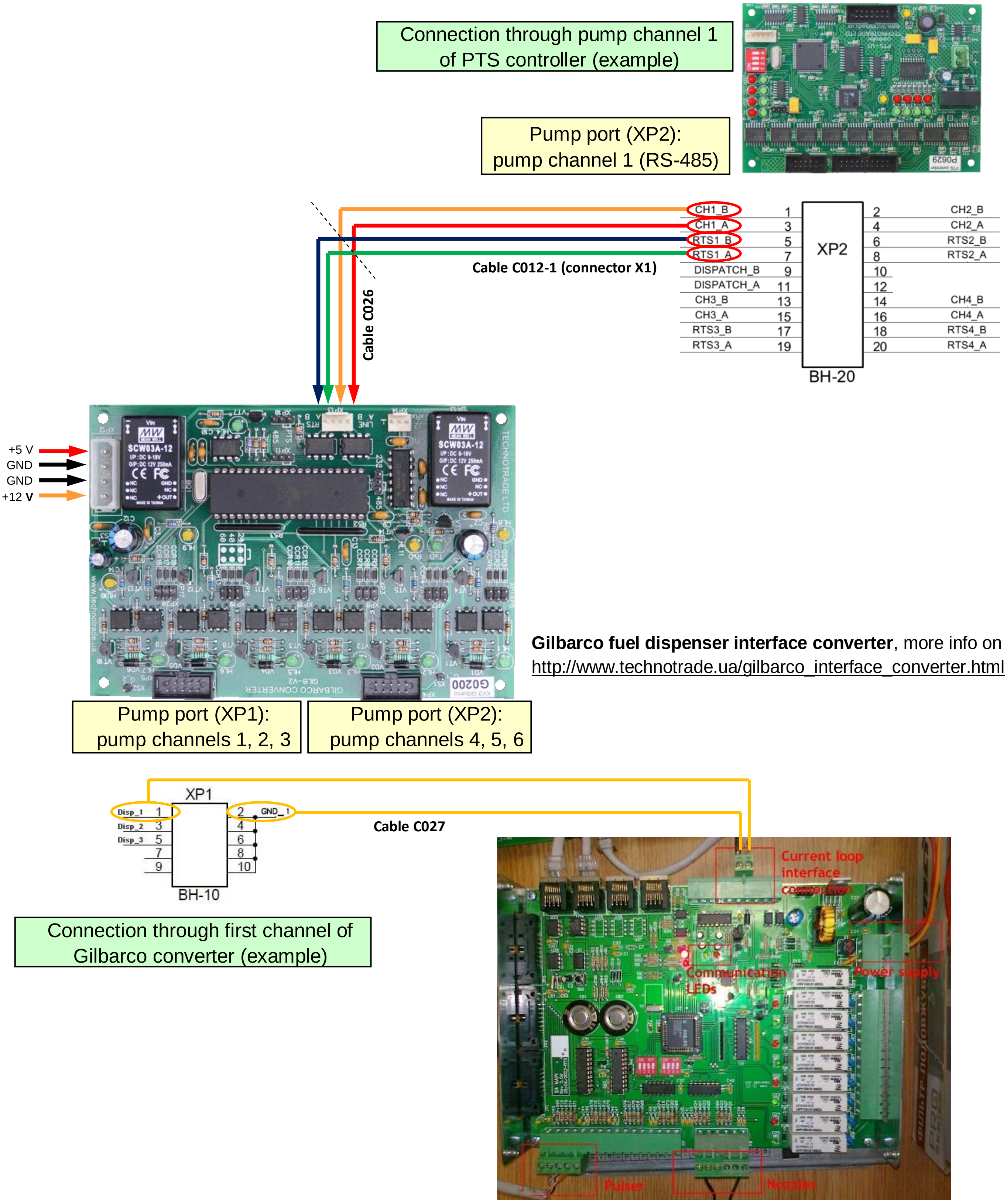

Connection through pump channel 1 of PTS controller (example)
Pump port (XP2): pump channel 1 (RS-485)
CH1_B 1
CH1_A 3
RTS1_B 5
RTS1_A 7
DISPATCH_B 9
DISPATCH_A 11
CH3_B 13
CH3_A 15
RTS3_B 17
RTS3_A 19
XP2
2 CH2_B
4 CH2_A
6 RTS2_B
8 RTS2_A
10
12
14 CH4_B
16 CH4_A
18 RTS4_B
20 RTS4_A
BH-20
Cable C012-1 (connector X1)
Cable C026
+5 V
GND
GND
+12 V
Pump port (XP1): pump channels 1, 2, 3
Pump port (XP2): pump channels 4, 5, 6
XP1
Disp_1 1
Disp_2 3
Disp_3 5
7
9
2 GND_1
4
6
8
10
BH-10
Cable C027
Connection through first channel of Gilbarco converter (example)
IntBox
+
-
Pump Computer (serial port)
Petposan-Beta CPU

**Gilbarco fuel dispenser interface converter**, more info on http://www.technotrade.ua/gilbarco_interface_converter.html

# *EuroPump dispenser connection scheme*

Connection to EuroPump dispenser is made through Gilbarco interface converter, which provides connection RS-232/RS-485 interface to Gilbarco 2-wire current loop interface.

Configuration of PTS controller: protocol **DART Simplex**, baud rate **9600**.



Connection through pump channel 1 of PTS controller (example)

Pump port (XP2): pump channel 1 (RS-485)

**Gilbarco fuel dispenser interface converter**, more info on http://www.technotrade.ua/gilbarco_interface_converter.html

Pump port (XP1): pump channels 1, 2, 3

Pump port (XP2): pump channels 4, 5, 6

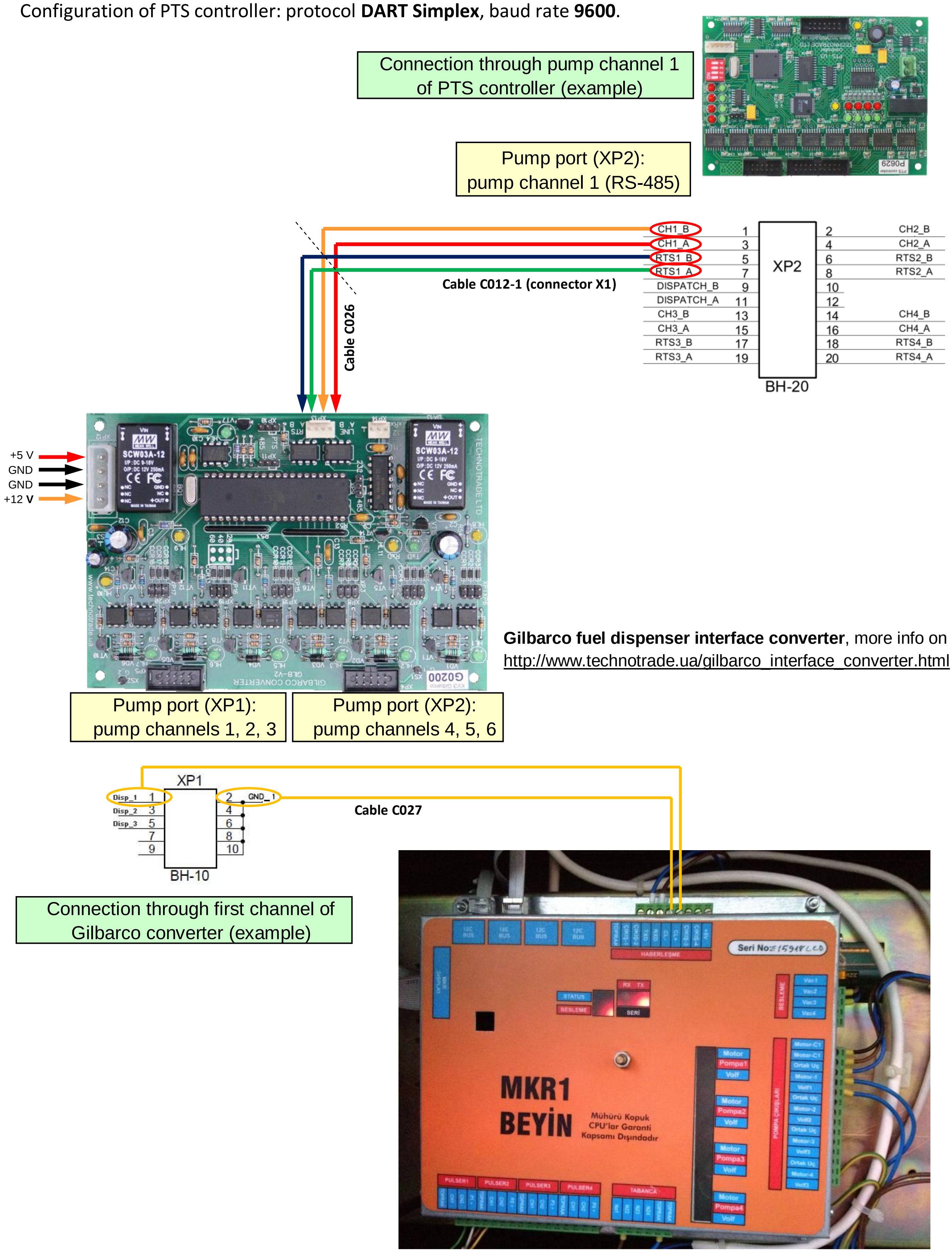
Connection through first channel of Gilbarco converter (example)

EuroPump dispenser board

# *Mekser dispenser connection scheme*

Connection to Mekser dispenser is made through Gilbarco interface converter, which provides connection RS-232/RS-485 interface to Gilbarco 2-wire current loop interface.

Configuration of PTS controller: protocol **DART Simplex**, baud rate **9600**.


Connection through pump channel 1 of PTS controller (example)
Pump port (XP2): pump channel 1 (RS-485)
CH1_B 1
CH1_A 3
RTS1_B 5
RTS1_A 7
DISPATCH_B 9
DISPATCH_A 11
CH3_B 13
CH3_A 15
RTS3_B 17
RTS3_A 19
XP2
BH-20
2 CH2_B
4 CH2_A
6 RTS2_B
8 RTS2_A
10
12
14 CH4_B
16 CH4_A
18 RTS4_B
20 RTS4_A
Cable C012-1 (connector X1)
Cable C026
+5 V
GND
GND
+12 V
Pump port (XP1): pump channels 1, 2, 3
Pump port (XP2): pump channels 4, 5, 6
XP1
Disp_1 1
Disp_2 3
Disp_3 5
7
9
2 GND_1
4
6
8
10
BH-10
Cable C027
Connection through first channel of Gilbarco converter (example)

**Gilbarco fuel dispenser interface converter**, more info on http://www.technotrade.ua/gilbarco_interface_converter.html

Mekser dispenser board

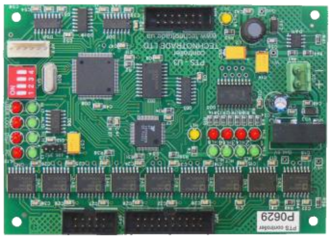
## *Mepsan UNIMEP dispenser connection scheme*

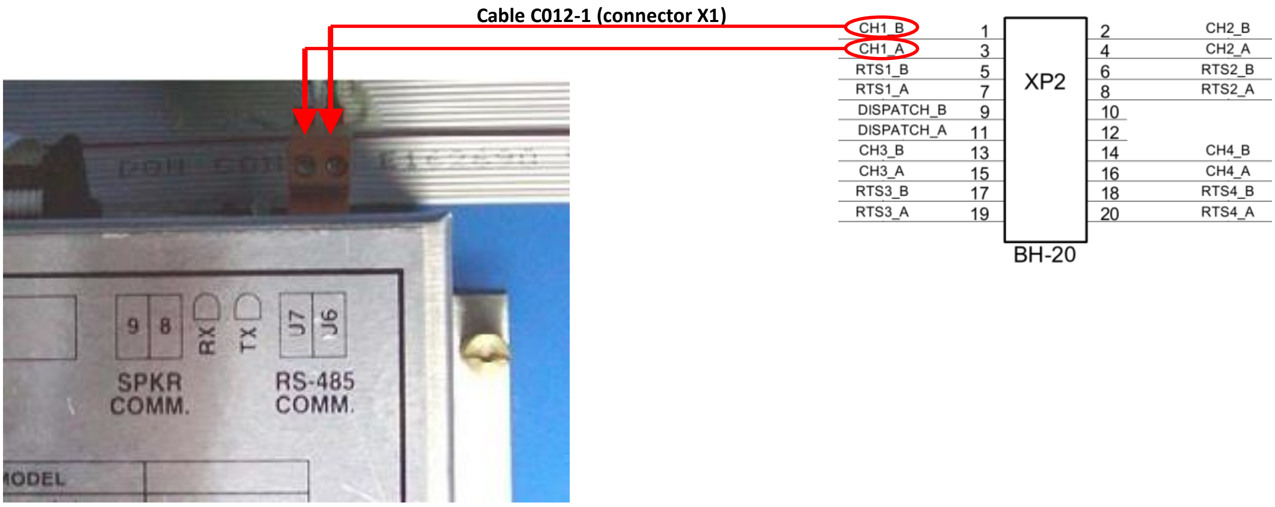
Connection to Mepsan dispenser is made directly without any interface converter.

Configuration of PTS controller: protocol **DART Simplex**, baud rate **9600**.

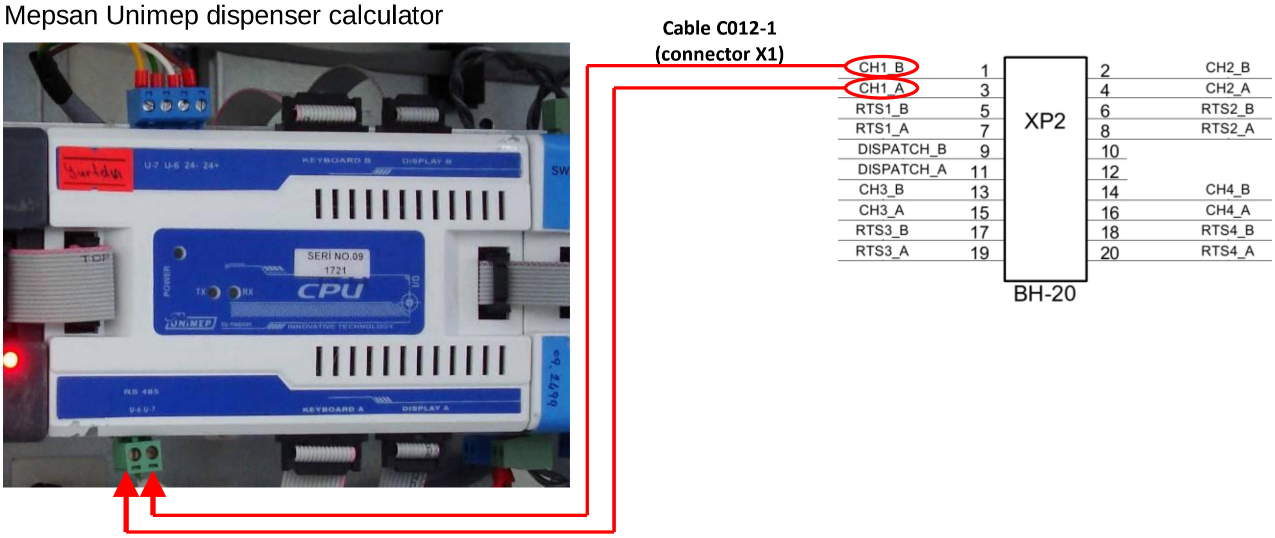
Connection through pump channel 1 of PTS controller (example)

Pump port (XP2): pump channel 1 (RS-485)

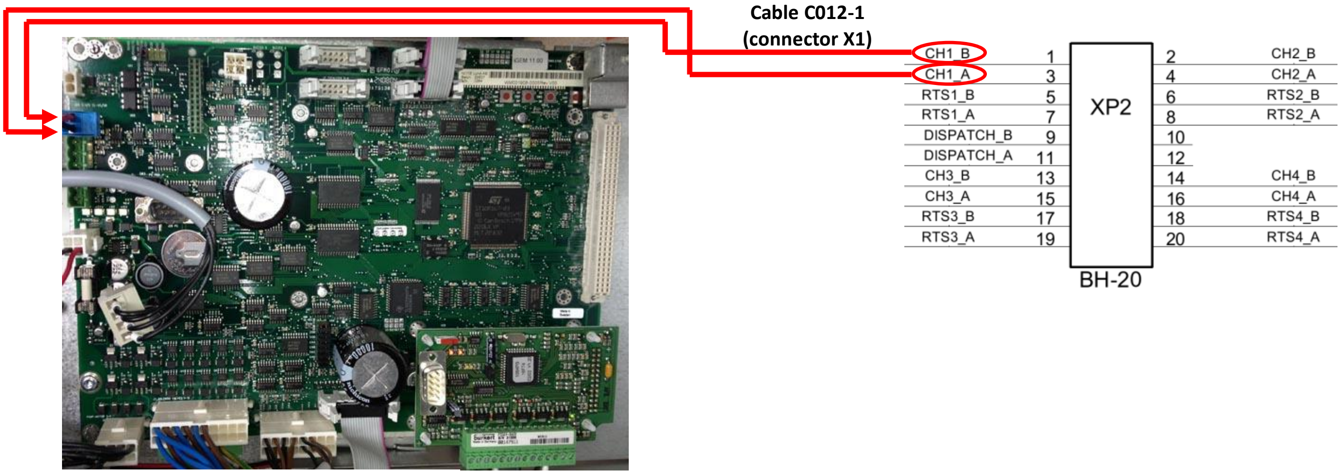




Mepsan Unimep dispenser calculator



Mepsan Unimep dispenser calculator

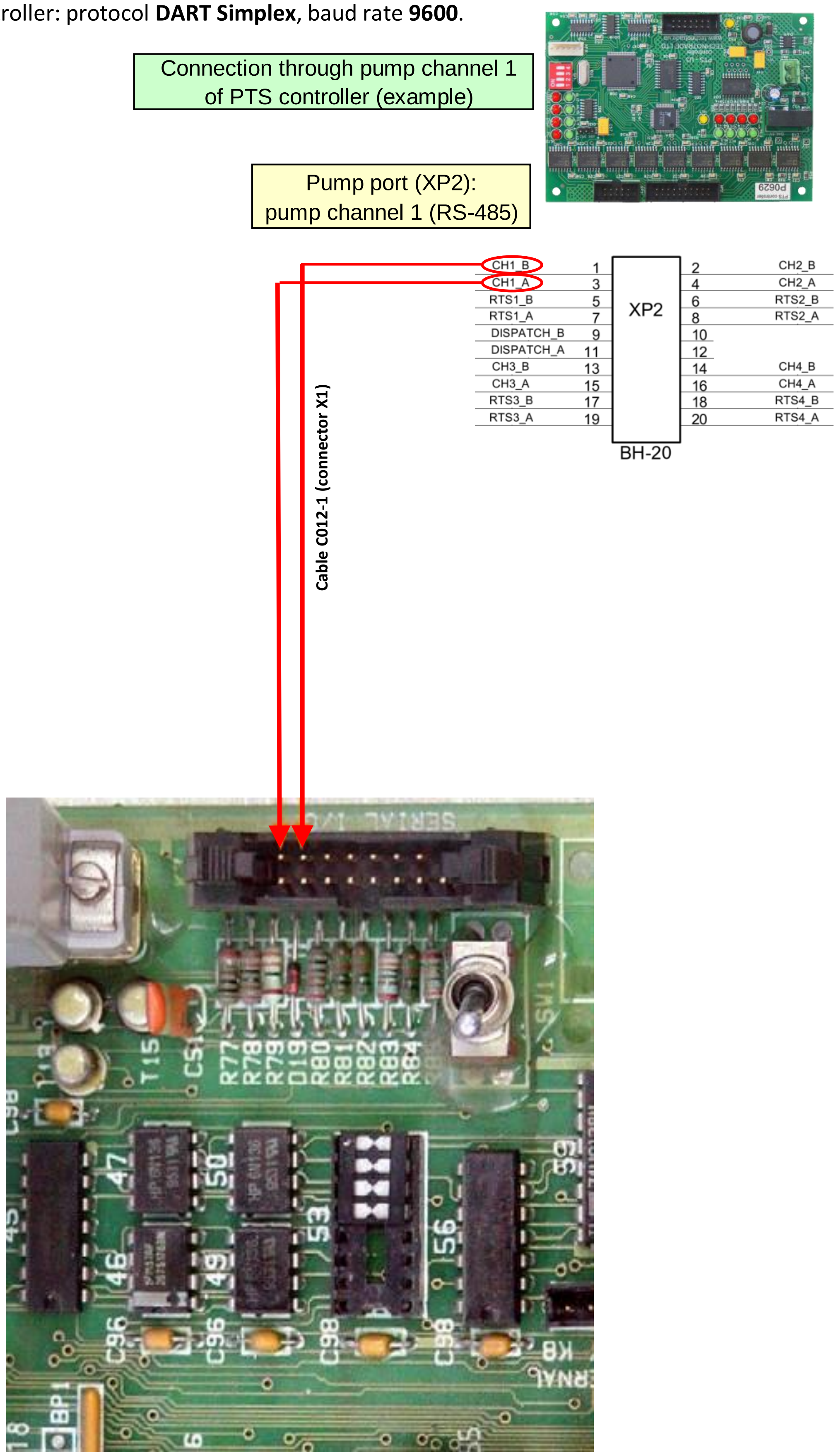


Mepsan dispenser board

## *Meksan / Wayne SU86 dispenser connection scheme*

Connection to Meksan / Wayne SU86 dispenser is made directly without any interface converter.

Configuration of PTS controller: protocol **DART Simplex**, baud rate **9600**.

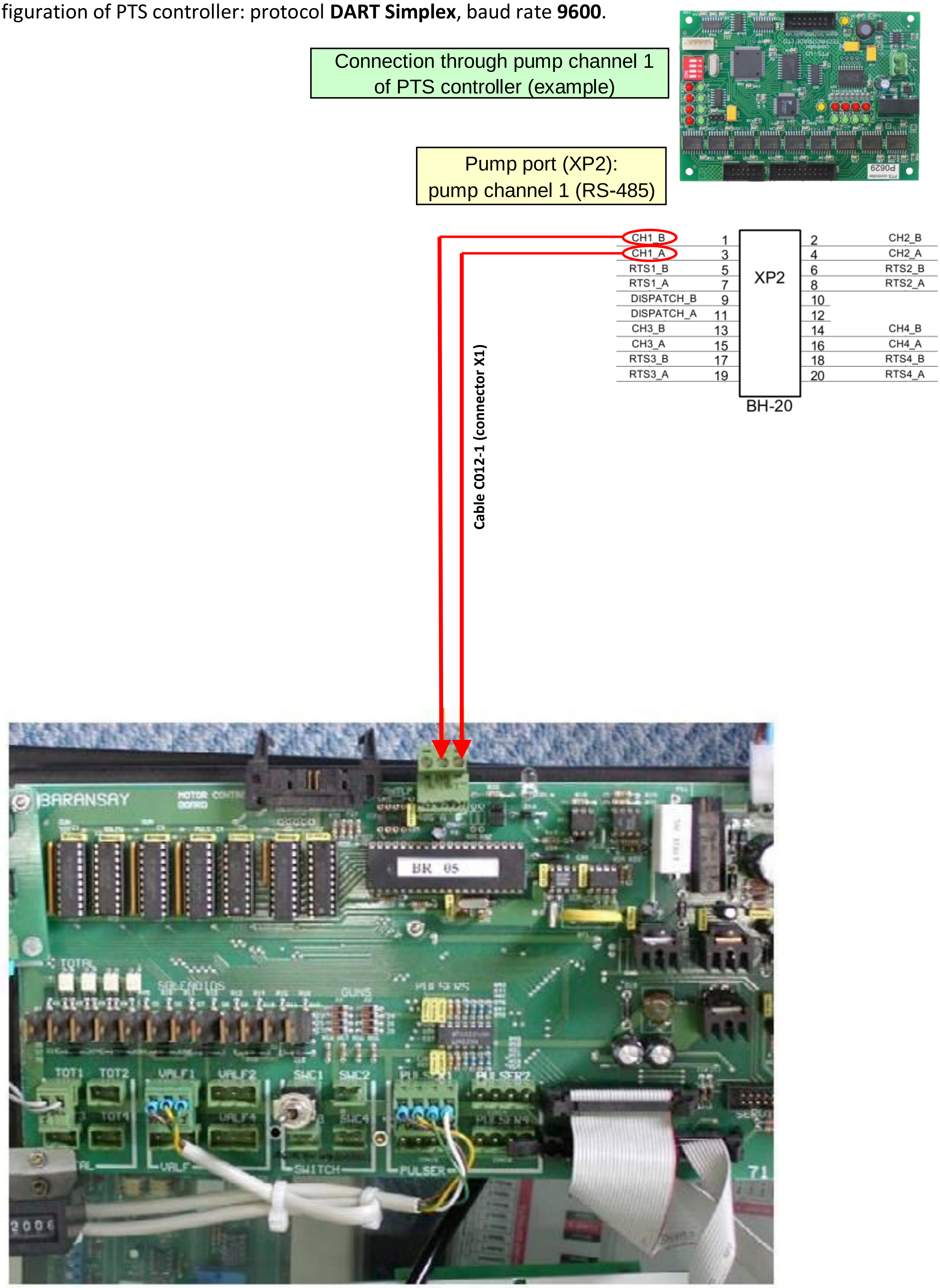


MEKSAN / WAYNE SU86 dispenser board

## *Baransay dispenser connection scheme*

Connection to Baransay dispenser is made directly without any interface converter.

Configuration of PTS controller: protocol **DART Simplex**, baud rate **9600**.

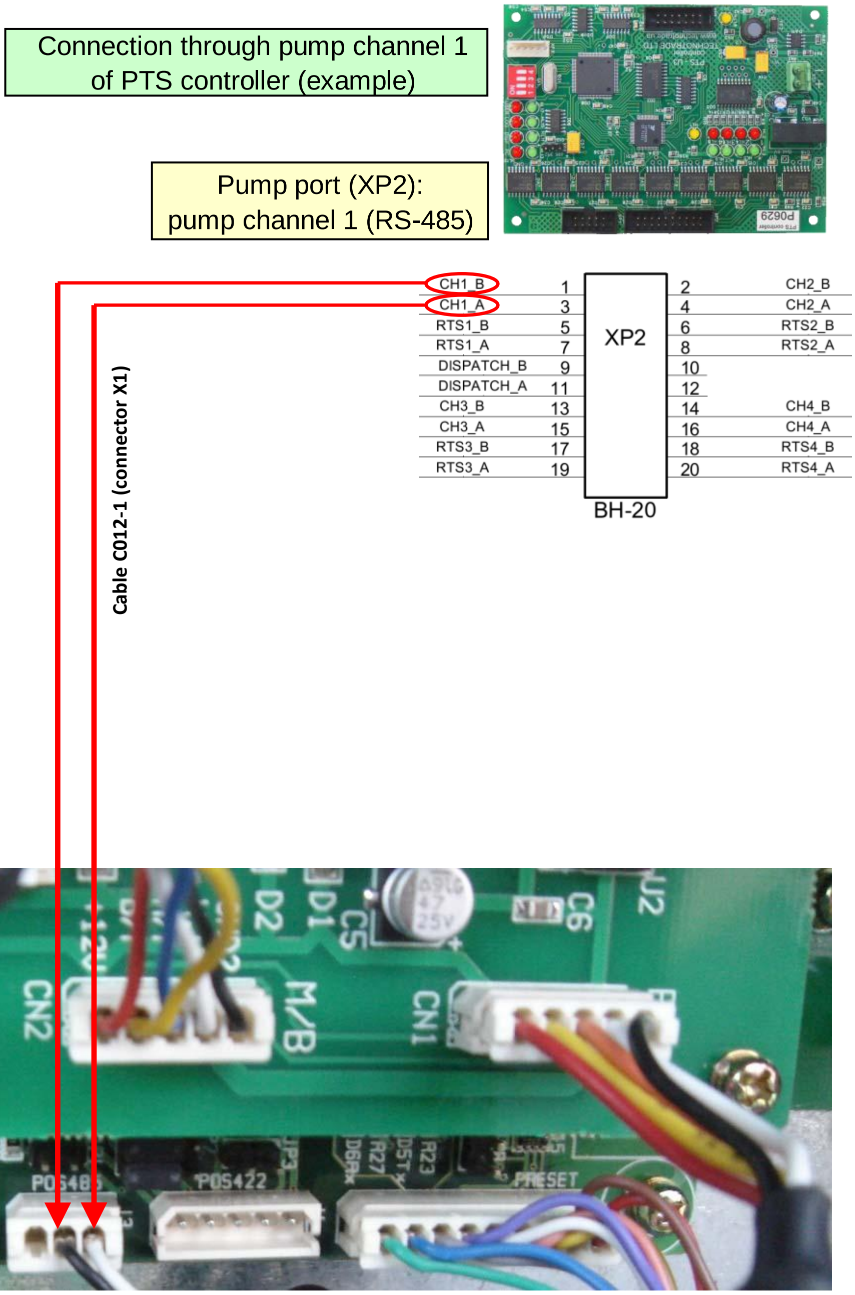


Bransay dispenser board

## *Korea EnE (LG EnE) dispenser connection scheme*

Connection to Korea EnE (LG EnE) dispenser is made directly without any interface converter.

Configuration of PTS controller: protocol **KOREA EnE**, baud rate **4800** or **9600**.



Connection through pump channel 1 of PTS controller (example)

Pump port (XP2): pump channel 1 (RS-485)

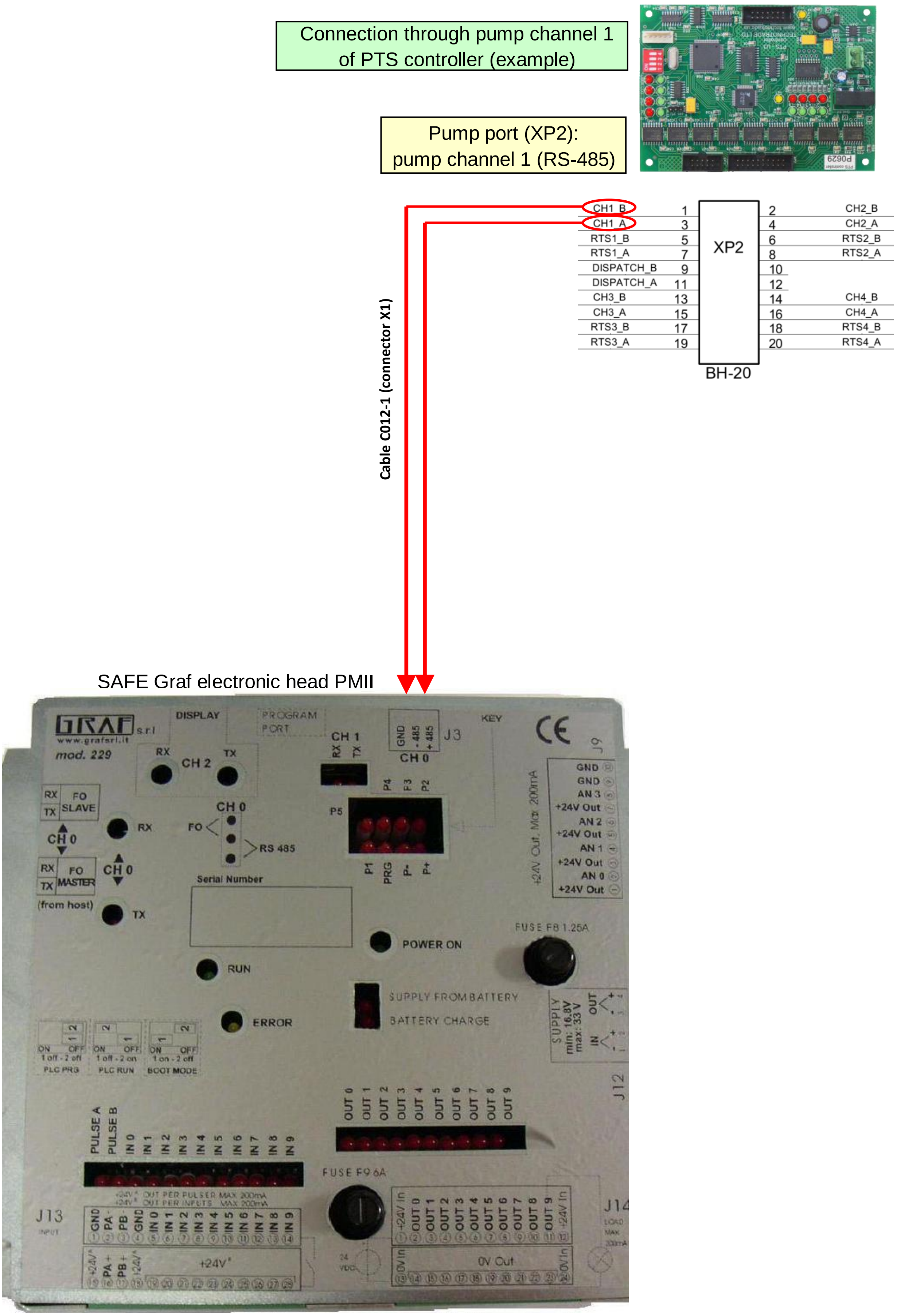
Cable C012-1 (connector X1)

Korea EnE / LG EnE dispenser board

## *SAFE dispenser connection scheme*

Connection to SAFE dispenser is made directly without any interface converter.

Configuration of PTS controller: protocol **SAFE Graf**, baud rate **9600**

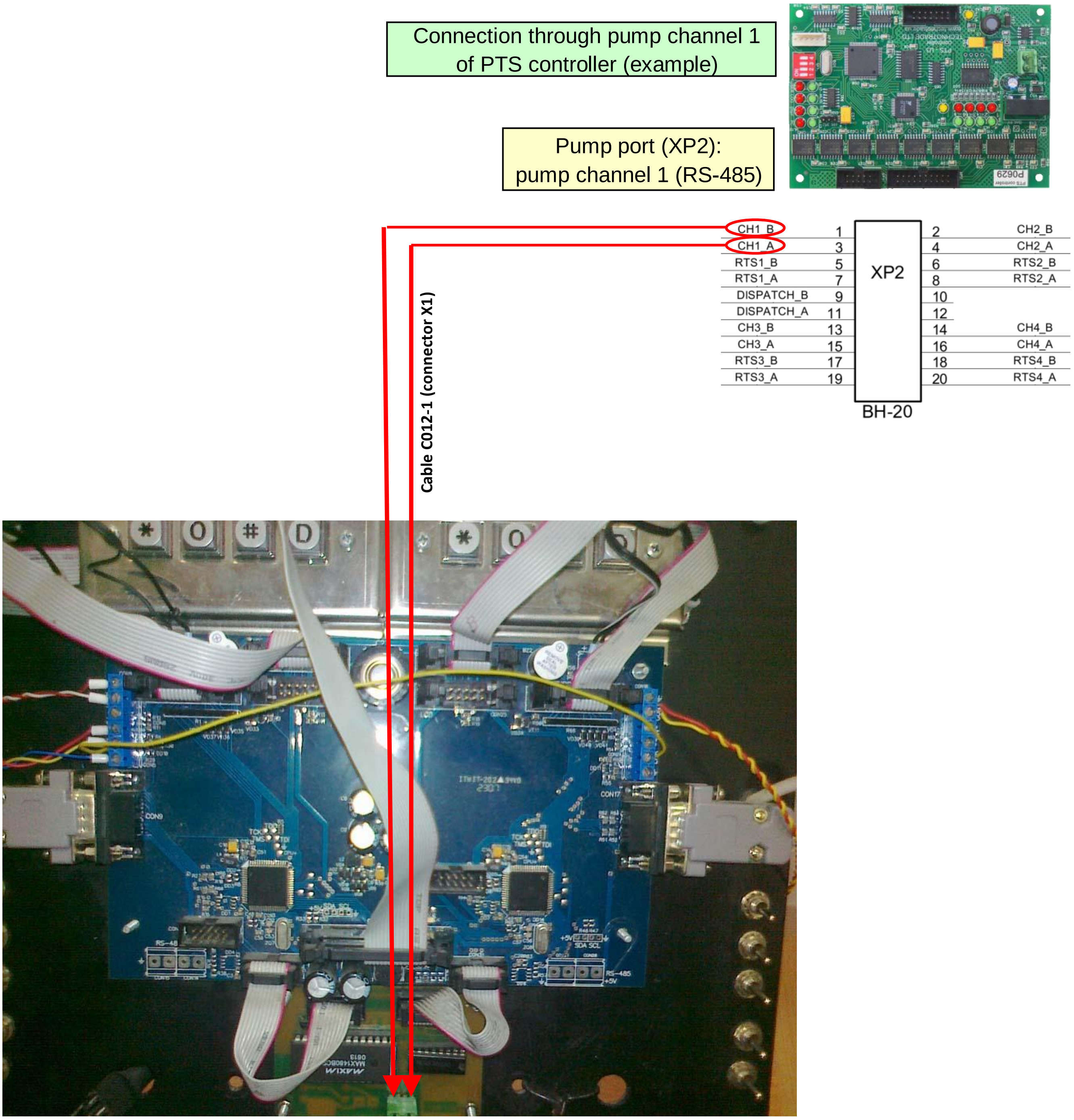


| | | | |
|---|---|---|---|
| CH1_B | 1 | 2 | CH2_B |
| CH1_A | 3 | 4 | CH2_A |
| RTS1_B | 5 | 6 | RTS2_B |
| RTS1_A | 7 | 8 | RTS2_A |
| DISPATCH_B | 9 | 10 | |
| DISPATCH_A | 11 | 12 | |
| CH3_B | 13 | 14 | CH4_B |
| CH3_A | 15 | 16 | CH4_A |
| RTS3_B | 17 | 18 | RTS4_B |
| RTS3_A | 19 | 20 | RTS4_A |

## *Shelf dispenser connection scheme*

Connection to Shelf dispenser is made directly without any interface converter.

Configuration of PTS controller: protocol **Shelf**, baud rate **9600**

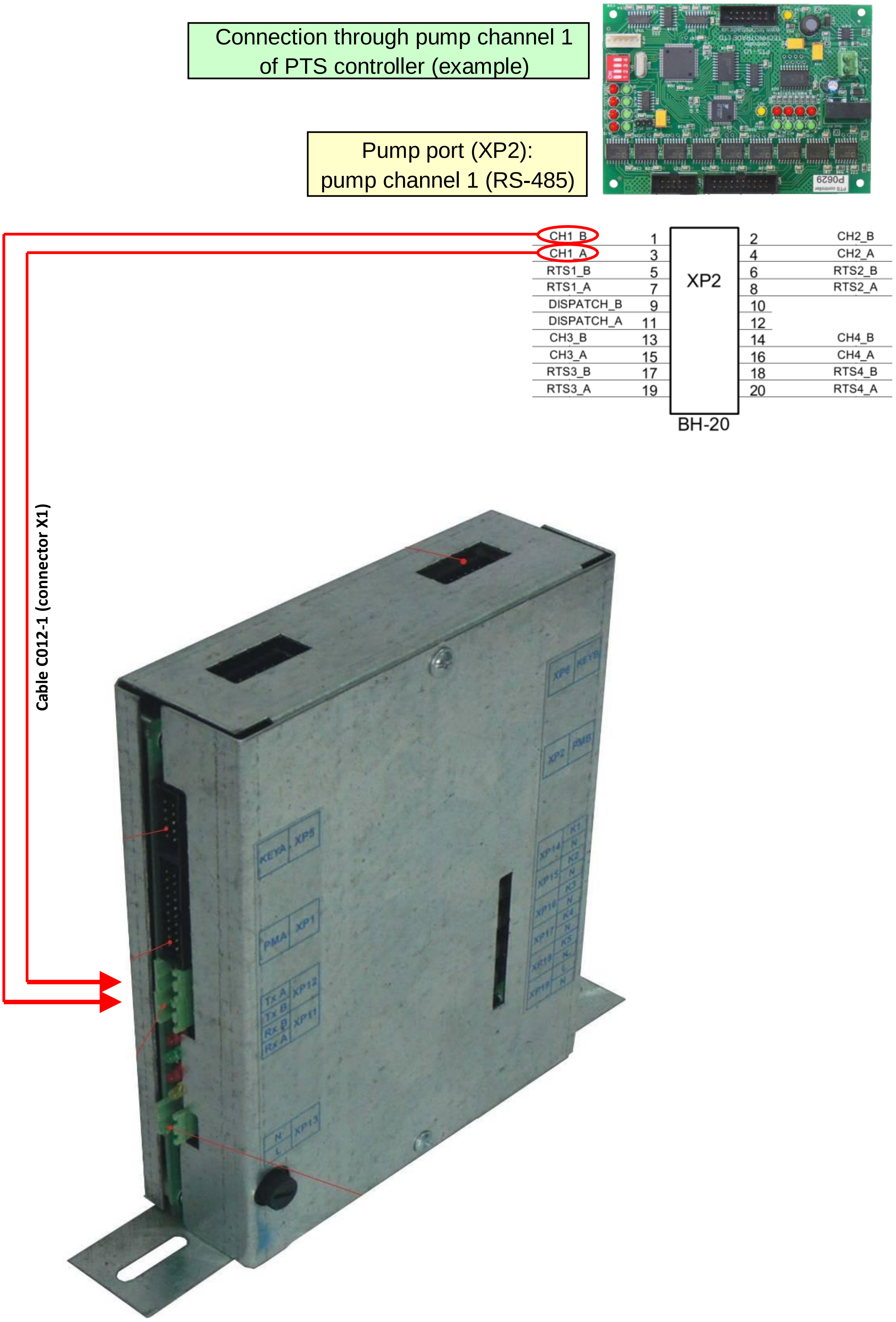


Shelf dispenser system board

## *UniCon dispenser connection scheme*

Connection to UniCon dispenser is made directly without any interface converter.

Configuration of PTS controller: protocol **UniPump**, baud rate **9600**



Connection through pump channel 1 of PTS controller (example)

Pump port (XP2): pump channel 1 (RS-485)

| | | XP2 | | |
|---|---|---|---|---|
| CH1_B | 1 | | 2 | CH2_B |
| CH1_A | 3 | | 4 | CH2_A |
| RTS1_B | 5 | | 6 | RTS2_B |
| RTS1_A | 7 | | 8 | RTS2_A |
| DISPATCH_B | 9 | | 10 | |
| DISPATCH_A | 11 | | 12 | |
| CH3_B | 13 | | 14 | CH4_B |
| CH3_A | 15 | | 16 | CH4_A |
| RTS3_B | 17 | | 18 | RTS4_B |
| RTS3_A | 19 | | 20 | RTS4_A |

BH-20

Cable C012-1 (connector X1)

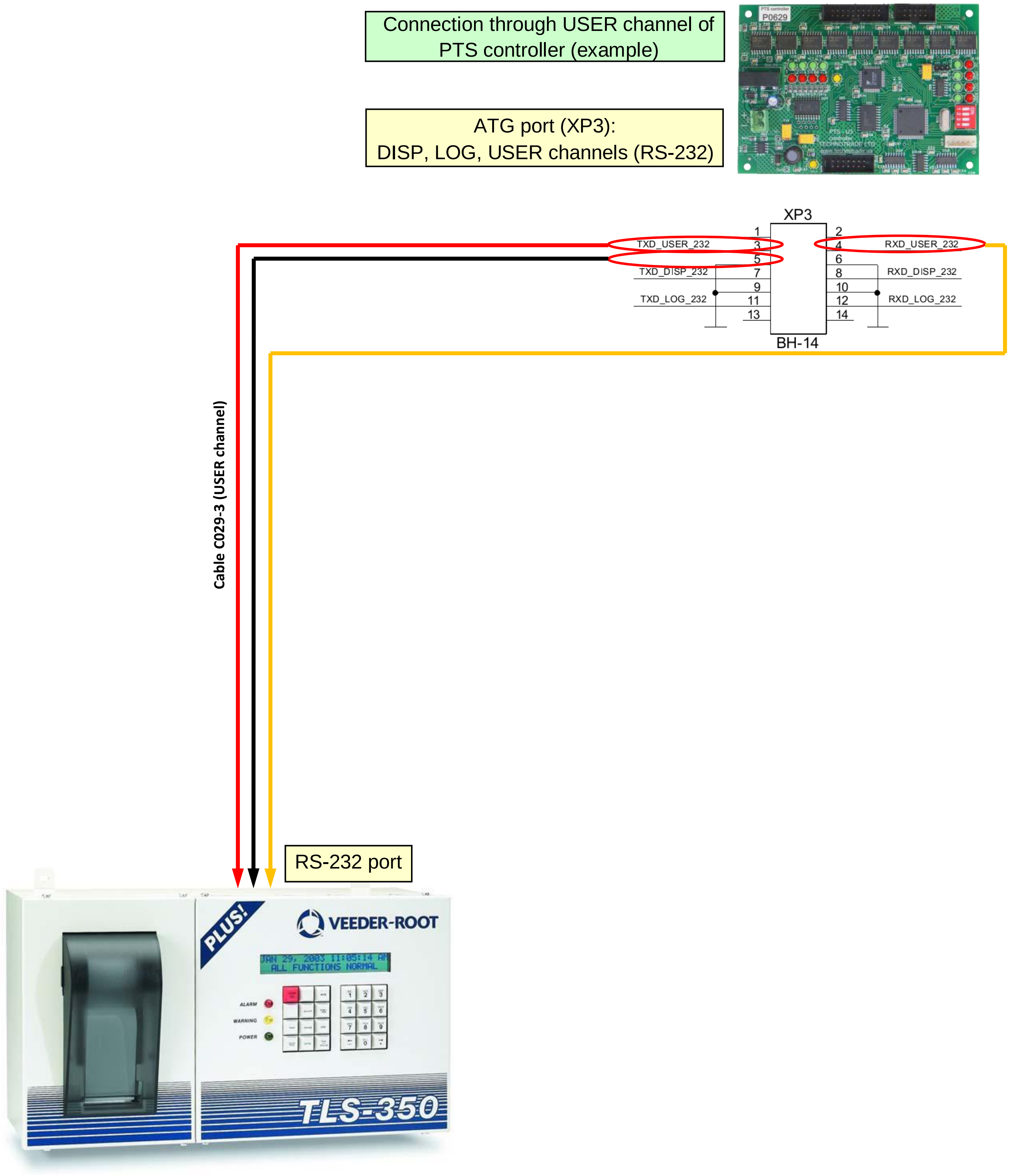
UniCon dispenser system board

# EXAMPLES OF ATG SYSTEMS CONNECTION SCHEMES

## *Gilbarco Veeder Root ATG systems TLS2, TLS-250, TLS-300, TLS-350, TLS-450 connection scheme*

Connection to Gilbarco Veeder Root TLS system is made directly to one of channels of ATG port of PTS controller (RS-232 interface).

Configuration of PTS controller – protocol **GILBARCO Veeder Root**, baud rate is selected to be equal to set in TLS ATG system.



Connection through USER channel of PTS controller (example)

ATG port (XP3): DISP, LOG, USER channels (RS-232)

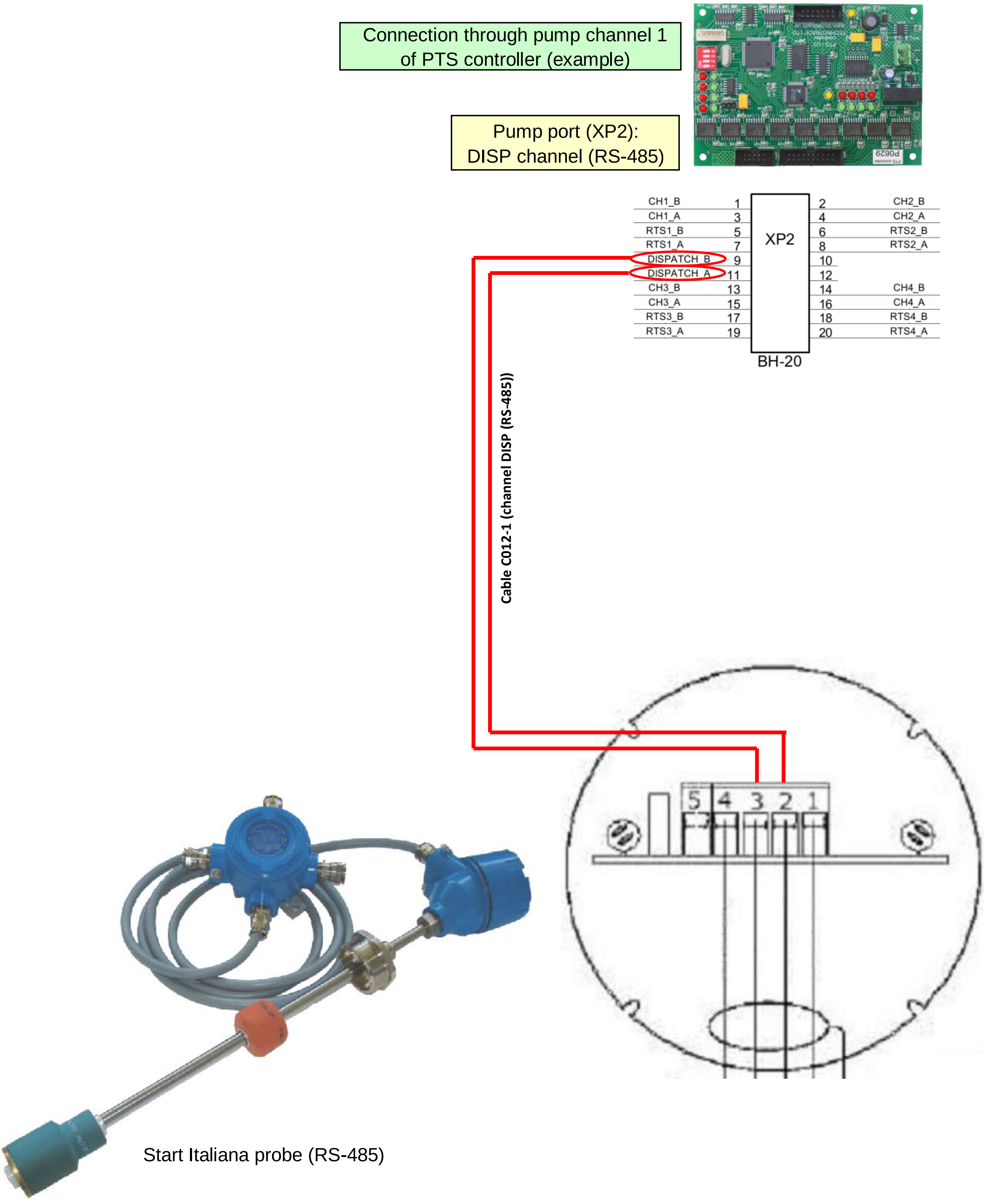
RS-232 port

Gilbarco Veeder Root TLS-350 ATG console

## *Start Italiana wired probes connection scheme*

Connection to Start Italiana probes is made directly to DISP (RS-485) channels of pump port of PTS controller (RS-485 interface).

Configuration of PTS controller – protocol **START ITALIANA SMT-XMT**, baud rate **9600**.

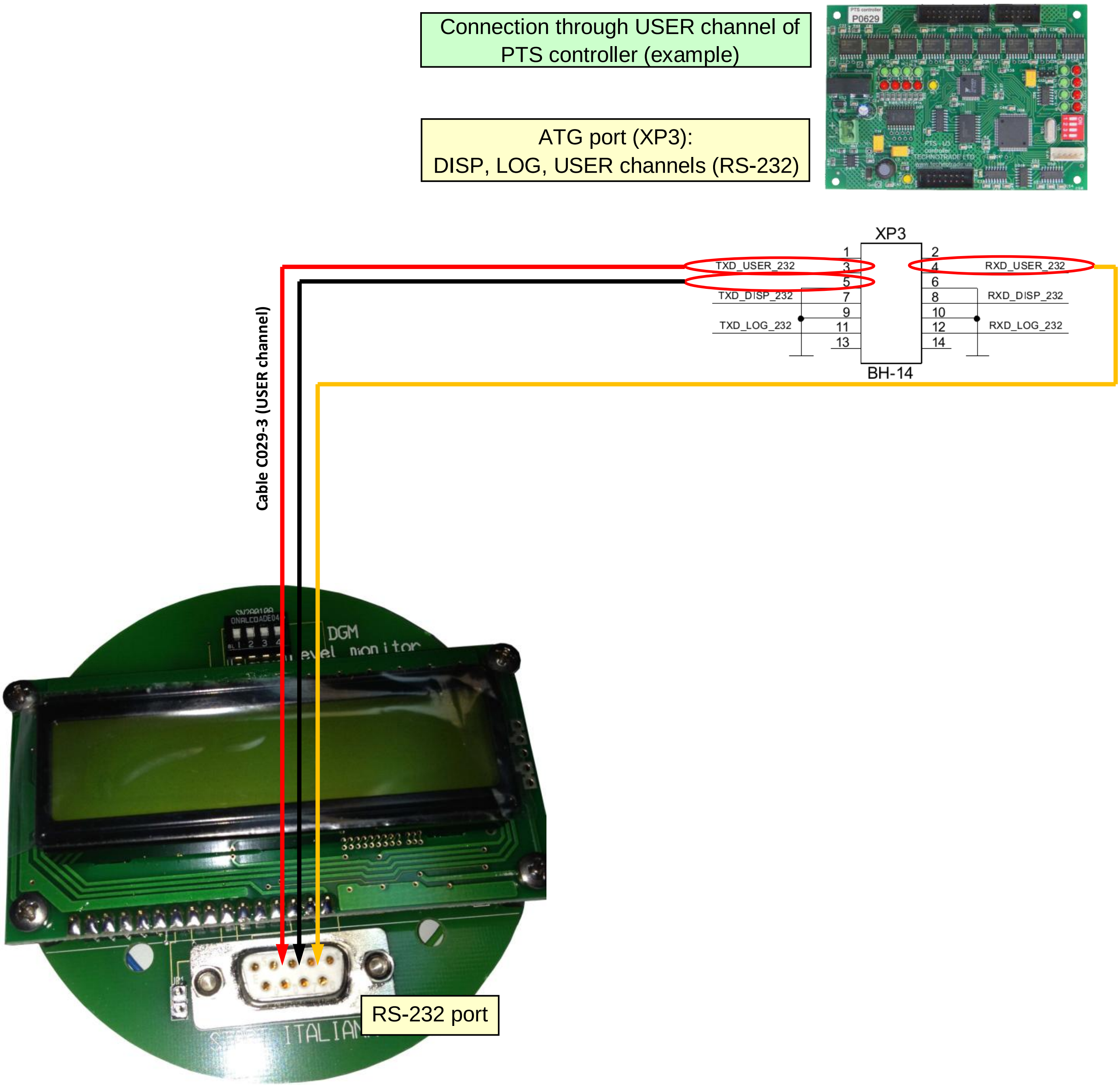

Connection through pump channel 1
of PTS controller (example)
Pump port (XP2):
DISP channel (RS-485)
XP2
BH-20
Cable C012-1 (channel DISP (RS-485))
5 4 3 2 1
Start Italiana probe (RS-485)

| | | | |
|---|---|---|---|
| CH1_B | 1 | 2 | CH2_B |
| CH1_A | 3 | 4 | CH2_A |
| RTS1_B | 5 | 6 | RTS2_B |
| RTS1_A | 7 | 8 | RTS2_A |
| DISPATCH_B | 9 | 10 | |
| DISPATCH_A | 11 | 12 | |
| CH3_B | 13 | 14 | CH4_B |
| CH3_A | 15 | 16 | CH4_A |
| RTS3_B | 17 | 18 | RTS4_B |
| RTS3_A | 19 | 20 | RTS4_A |

## *Start Italiana wireless probes connection scheme*

Connection to Start Italiana system is made directly to one of channels of ATG port of PTS controller (RS-232 interface).

Configuration of PTS controller – protocol **START ITALIANA SMT-XMT**, baud rate **9600**.



Connection through USER channel of PTS controller (example)

ATG port (XP3): DISP, LOG, USER channels (RS-232)

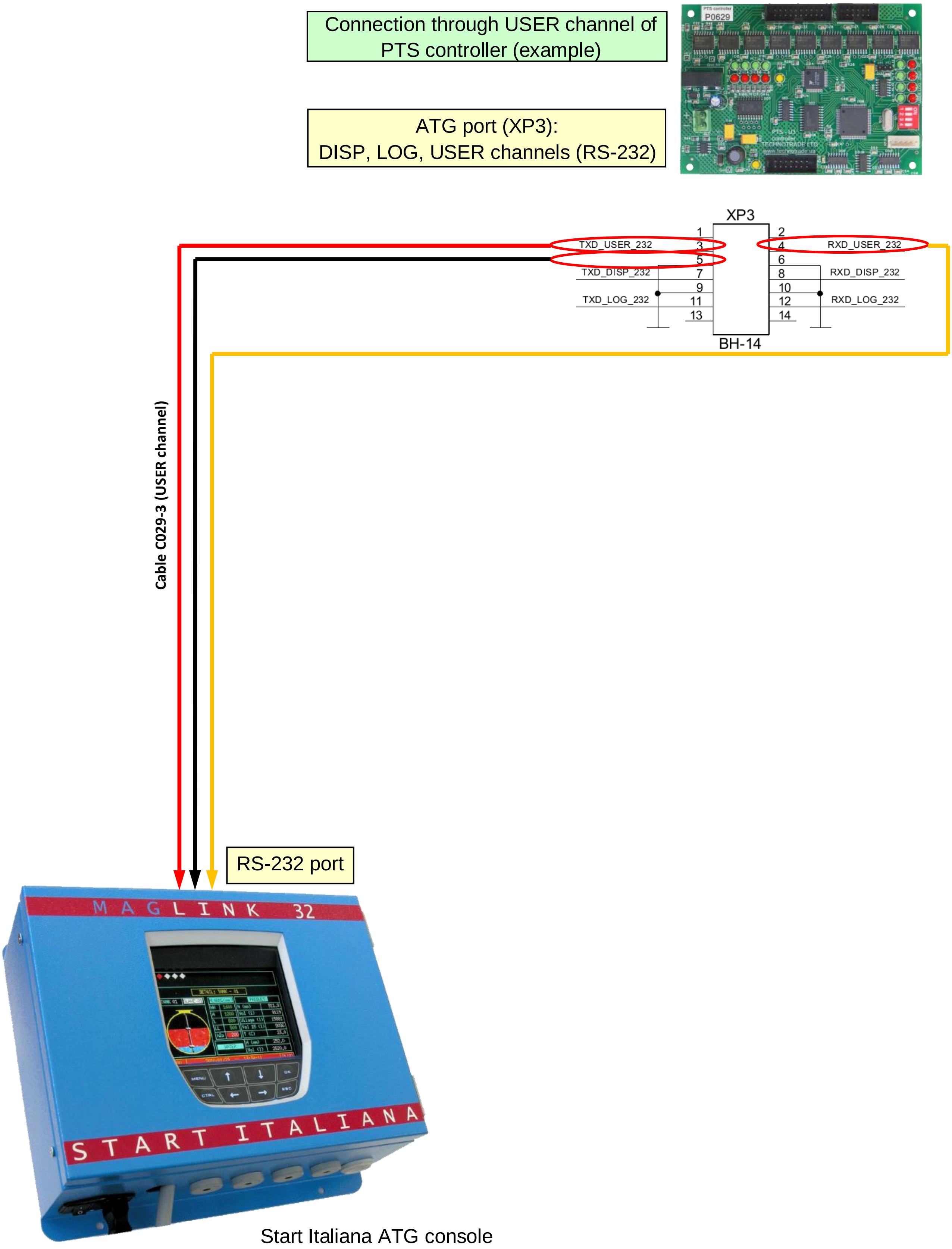
RS-232 port

Start Italiana RF protocol converter board

## *Start Italiana ATG system connection scheme*

Connection to Start Italiana system is made directly to one of channels of ATG port of PTS controller (RS-232 interface).

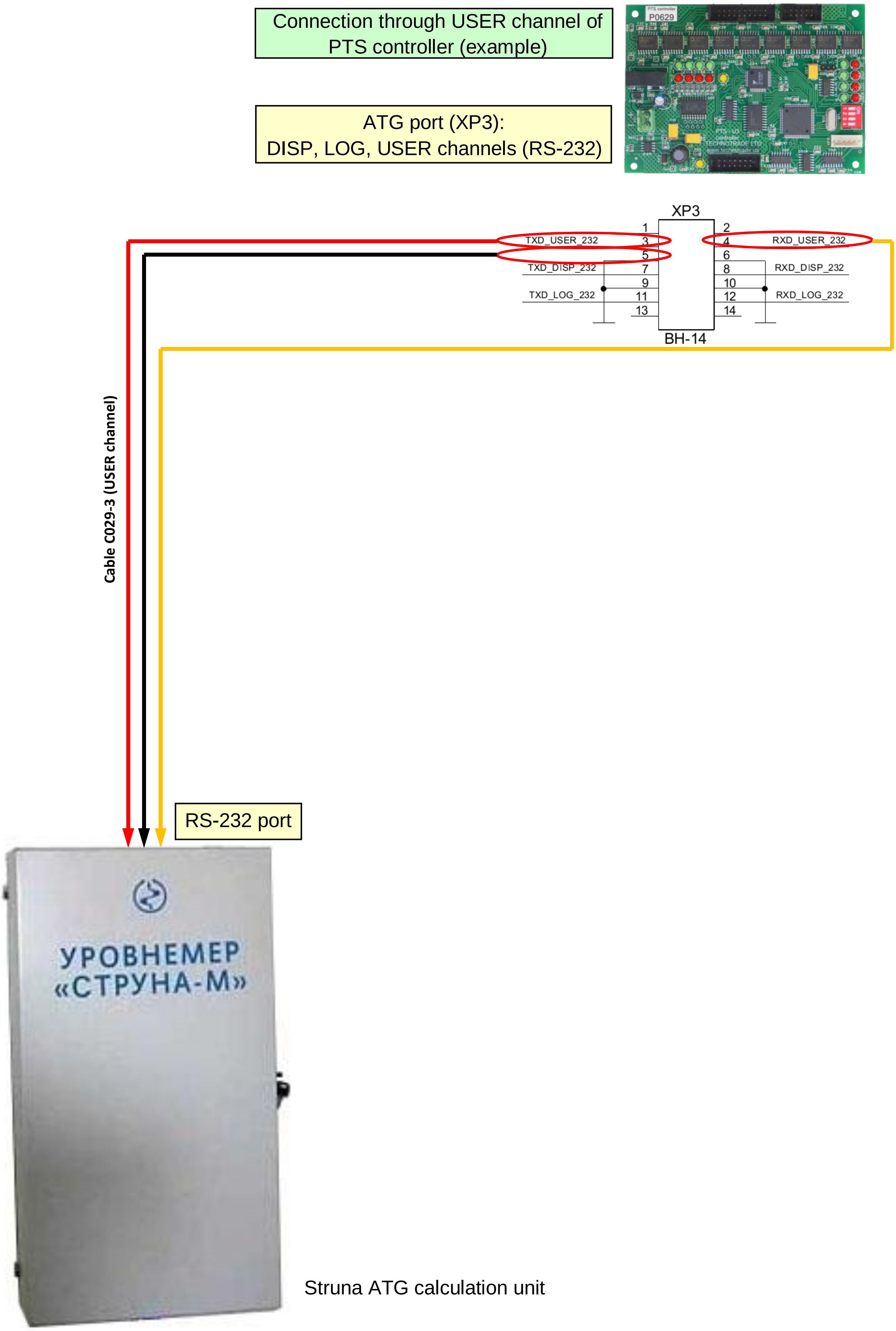
Configuration of PTS controller – protocol **GILBARCO Veeder Root**, baud rate is selected to be equal to set in Start Italiana ATG system.



## *Struna ATG system connection scheme*

Connection to Struna system is made directly to one of channels of ATG port of PTS controller (RS-232 interface).

Configuration of PTS controller – protocol **STRUNA Kedr 1.4**, baud rate is selected to be equal to set in Struna ATG system.

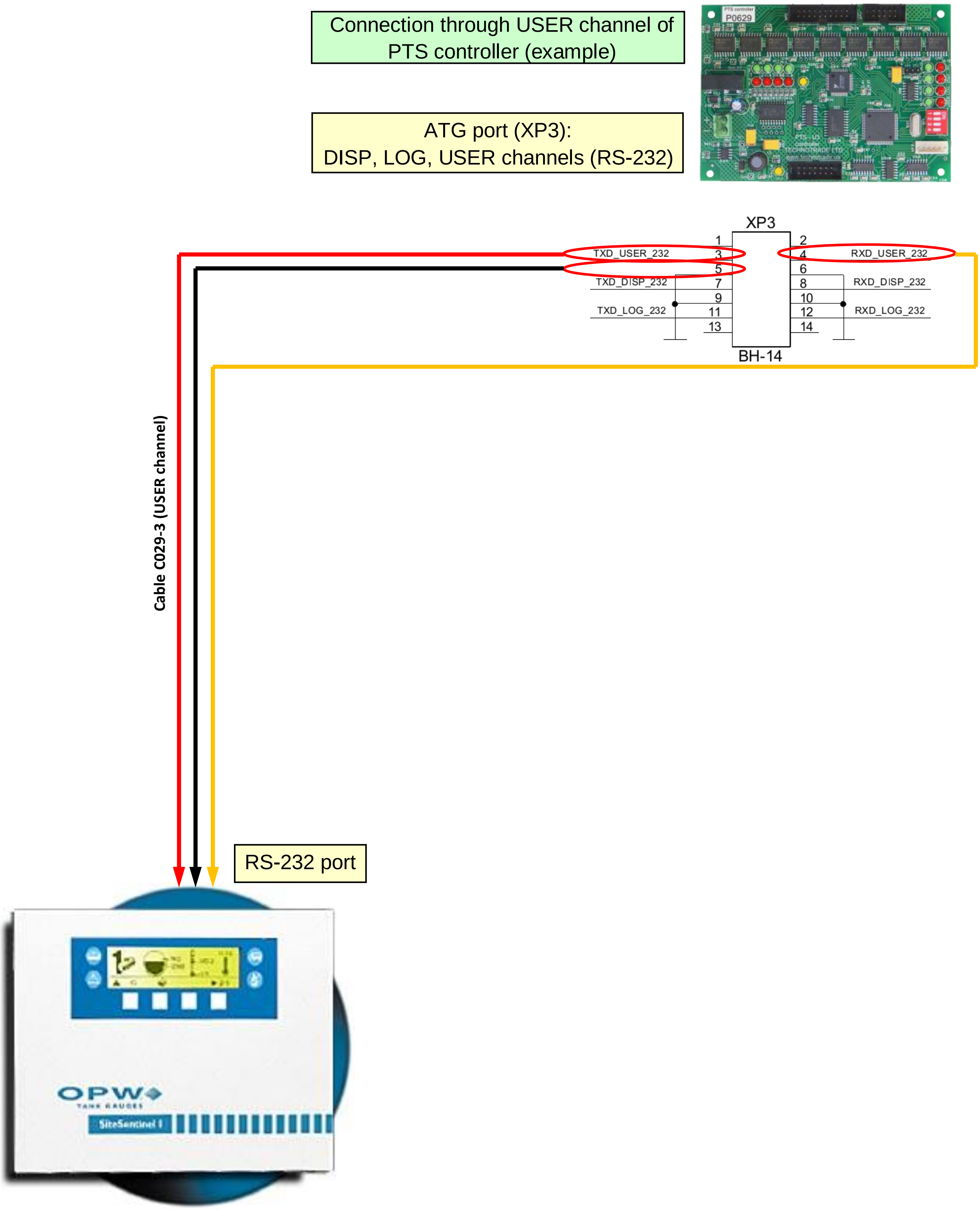


Struna ATG calculation unit

## *OPW Site Sentinel ATG system connection scheme*

Connection to OPW system is made directly to one of channels of ATG port of PTS controller (RS-232 interface).

Configuration of PTS controller – protocol **PETROVEND 4**, baud rate is selected to be equal to set in OPW ATG system.



Connection through USER channel of PTS controller (example)

ATG port (XP3): DISP, LOG, USER channels (RS-232)

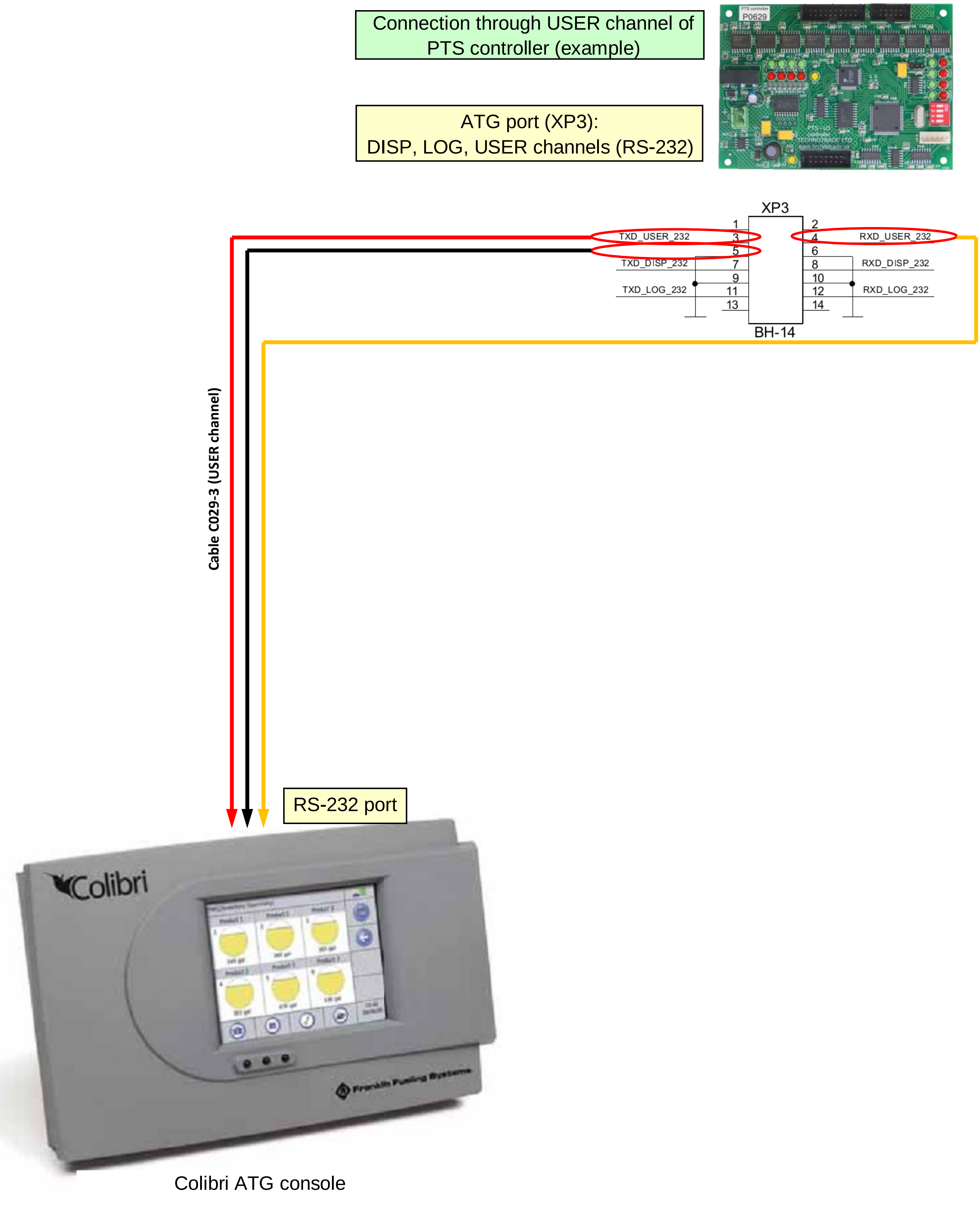
RS-232 port

OPW Site Sentinel ATG console

## *Colibri ATG system connection scheme*

Connection to Colibri system is made directly to one of channels of ATG port of PTS controller (RS-232 interface).

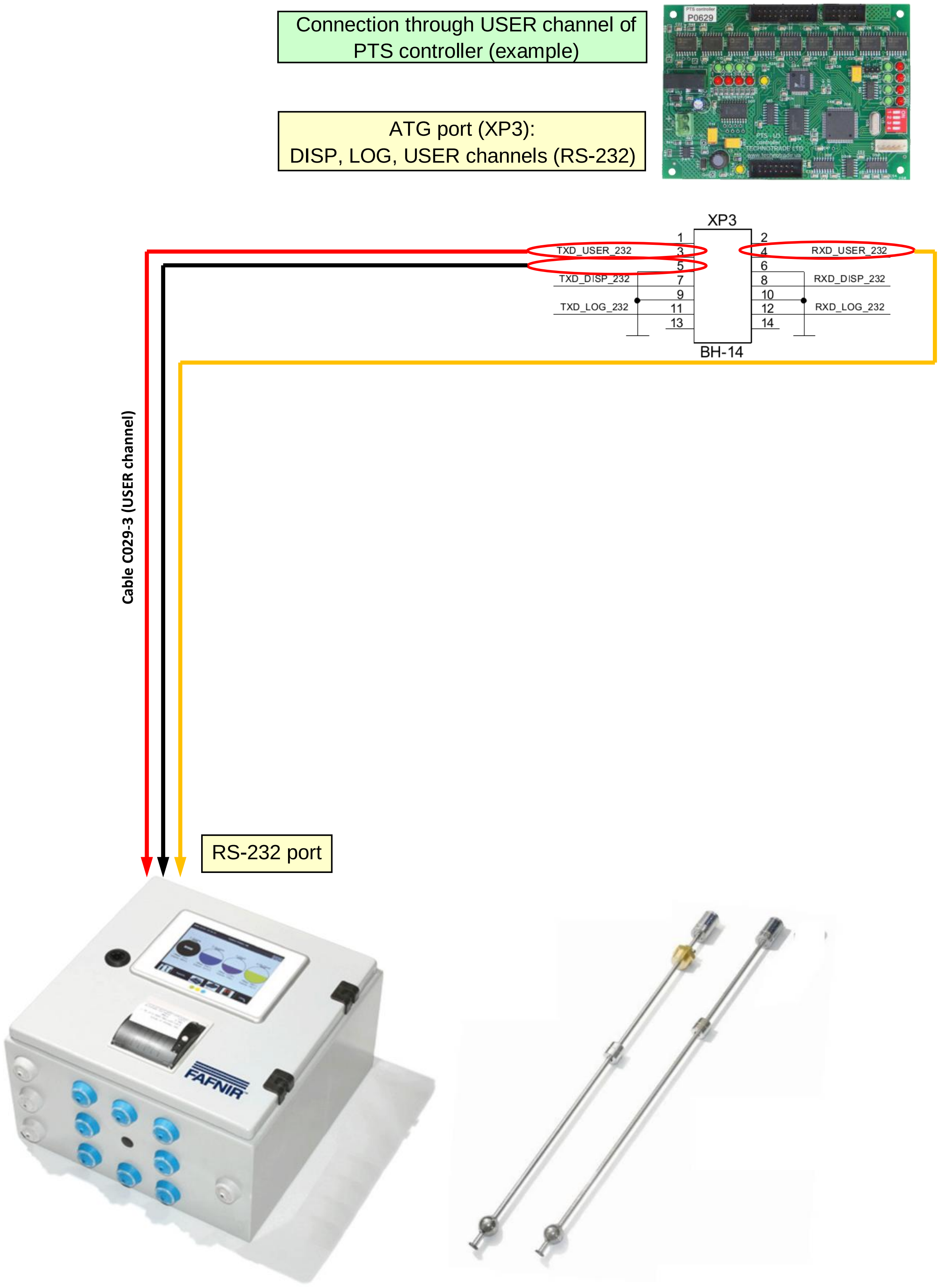
Configuration of PTS controller – protocol **GILBARCO Veeder Root**, baud rate is selected to be equal to set in Start Italiana ATG system.



## *FAFNIR ATG system connection scheme*

Connection to FAFNIR system is made directly to one of channels of ATG port of PTS controller (RS-232 interface).

Configuration of PTS controller – protocol **FAFNIR VISY-Quick**, baud rate is selected to be equal to set in FAFNIR ATG system.

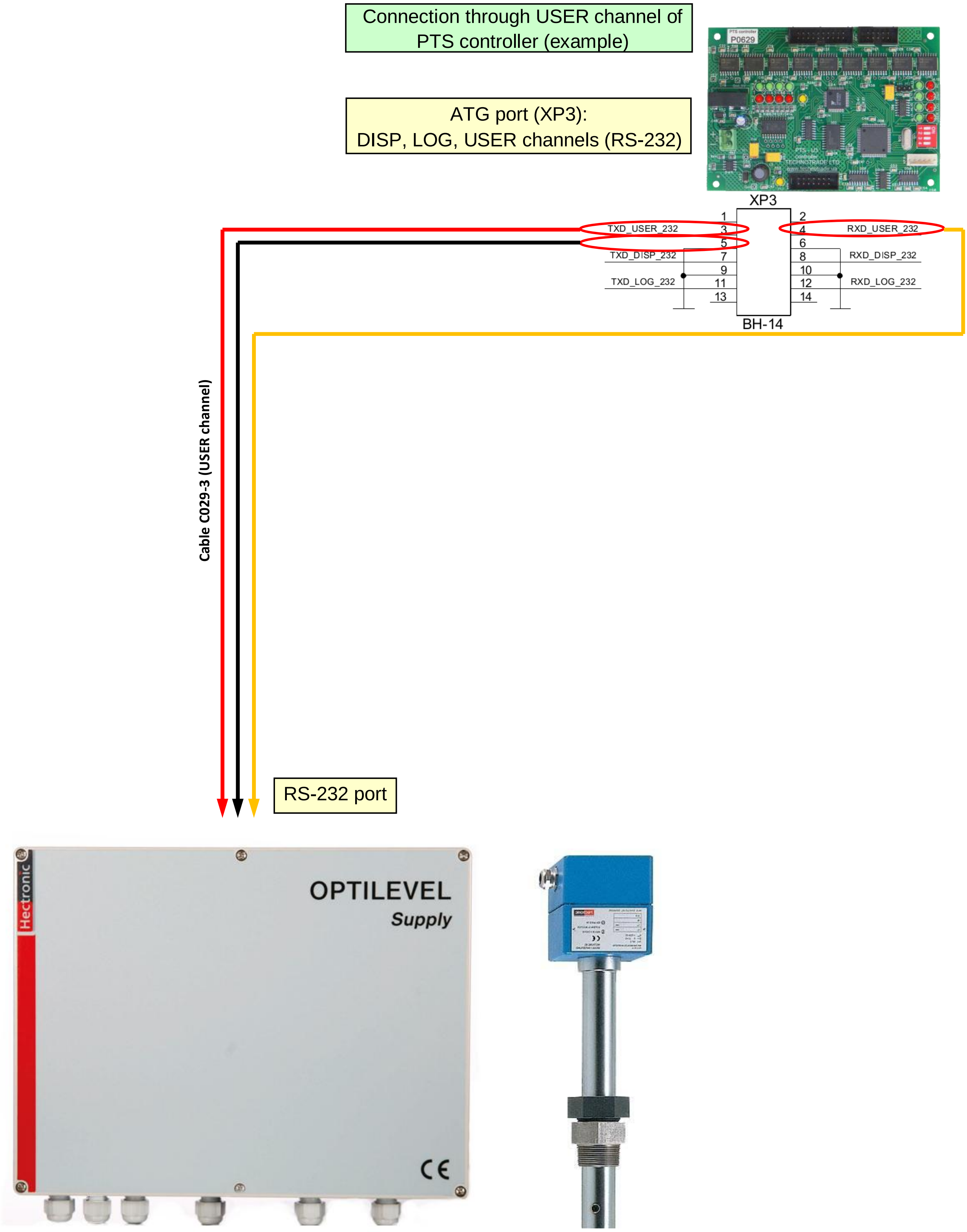


FAFNIR-Command ATG console

## *Hectronic ATG probes connection scheme*

Connection to Hectronic probes is made directly to one of channels of ATG port of PTS controller (RS-232 interface) to Hectronic OptiLevel Supply box.

Configuration of PTS controller – protocol **HECTRONIC HLS**, baud rate **9600**.



Connection through USER channel of PTS controller (example)

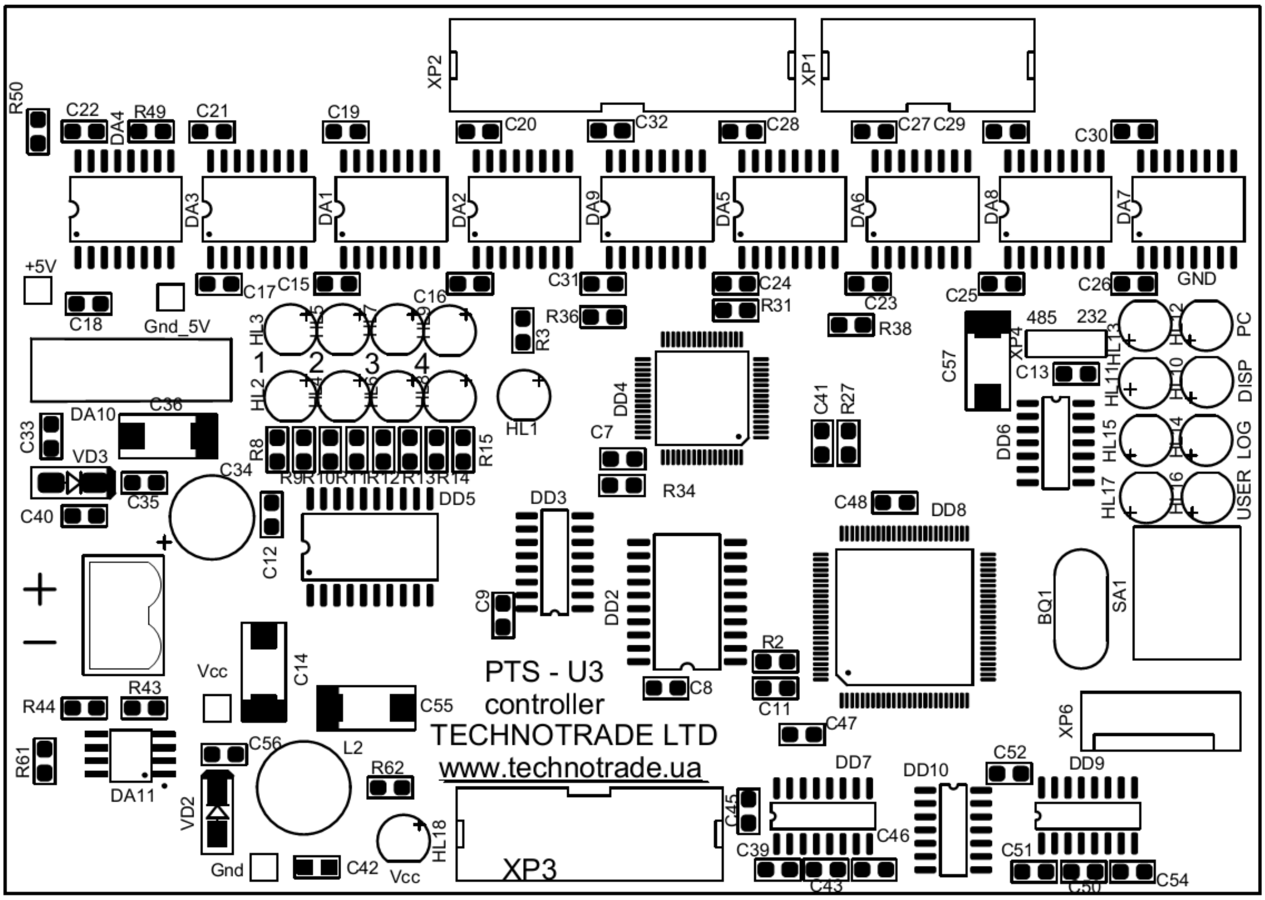
ATG port (XP3):
DISP, LOG, USER channels (RS-232)

RS-232 port

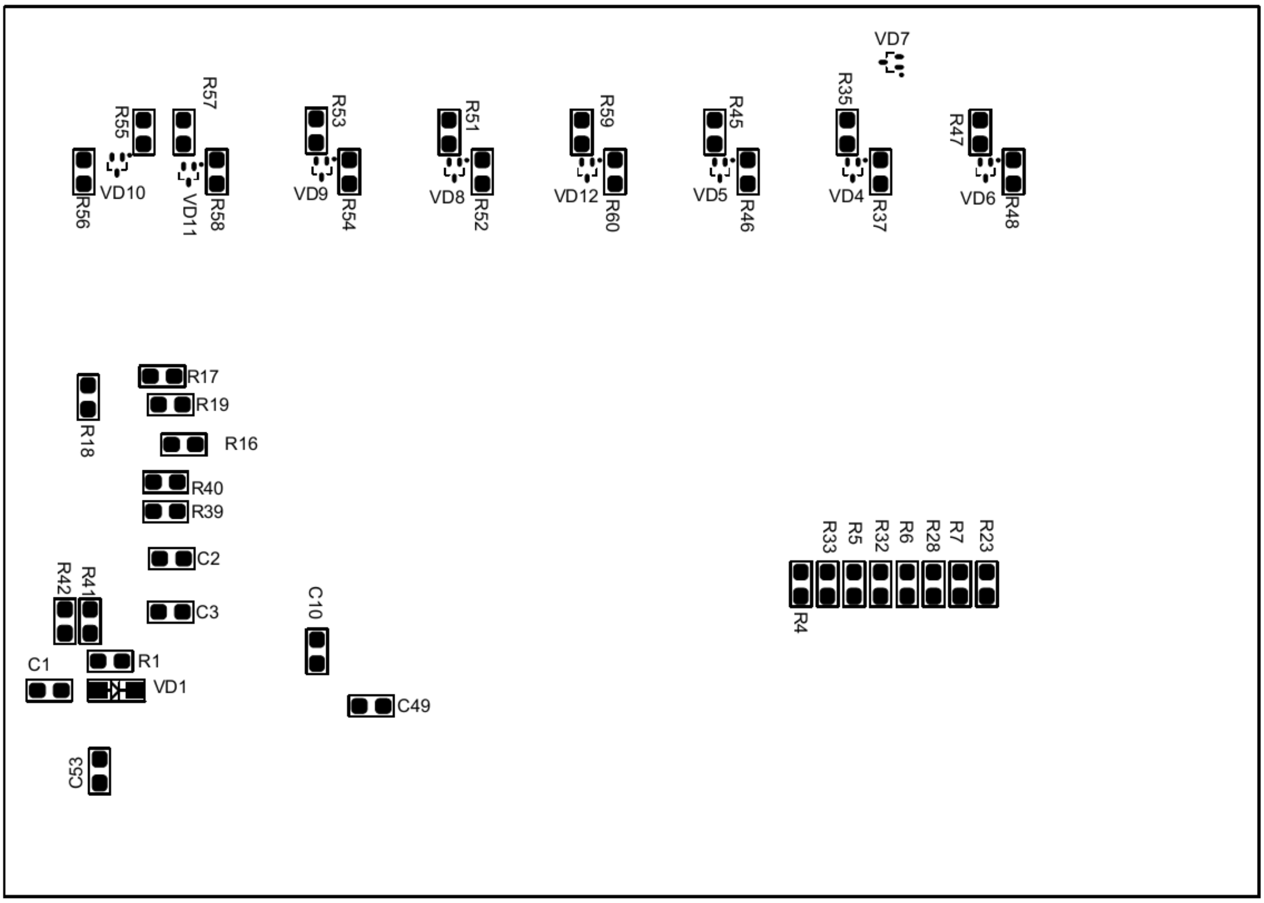
Hectronic OptiLevel Supply box

# PTS CONTROLLER PCB BOARD

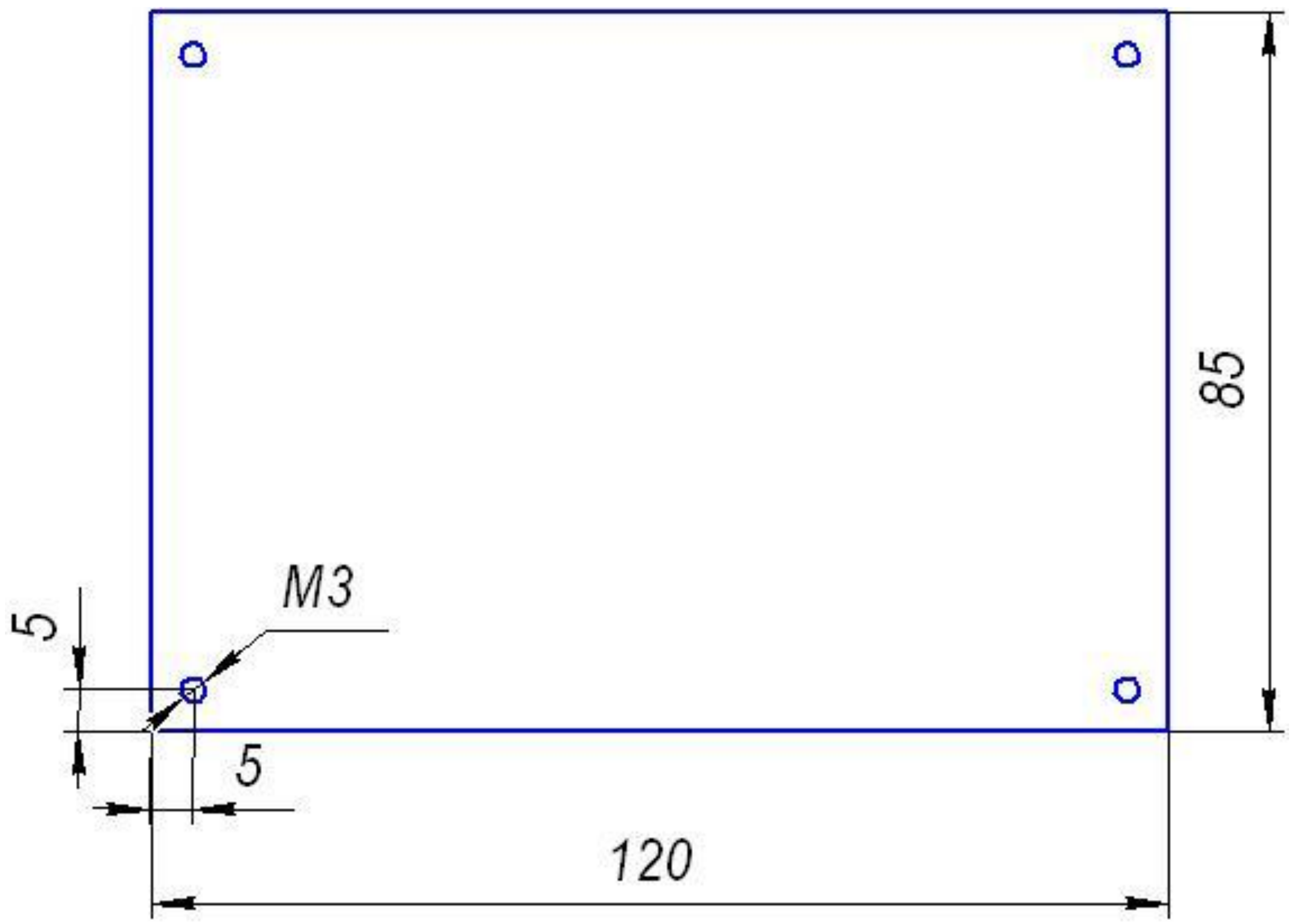
## *PCB board top view*



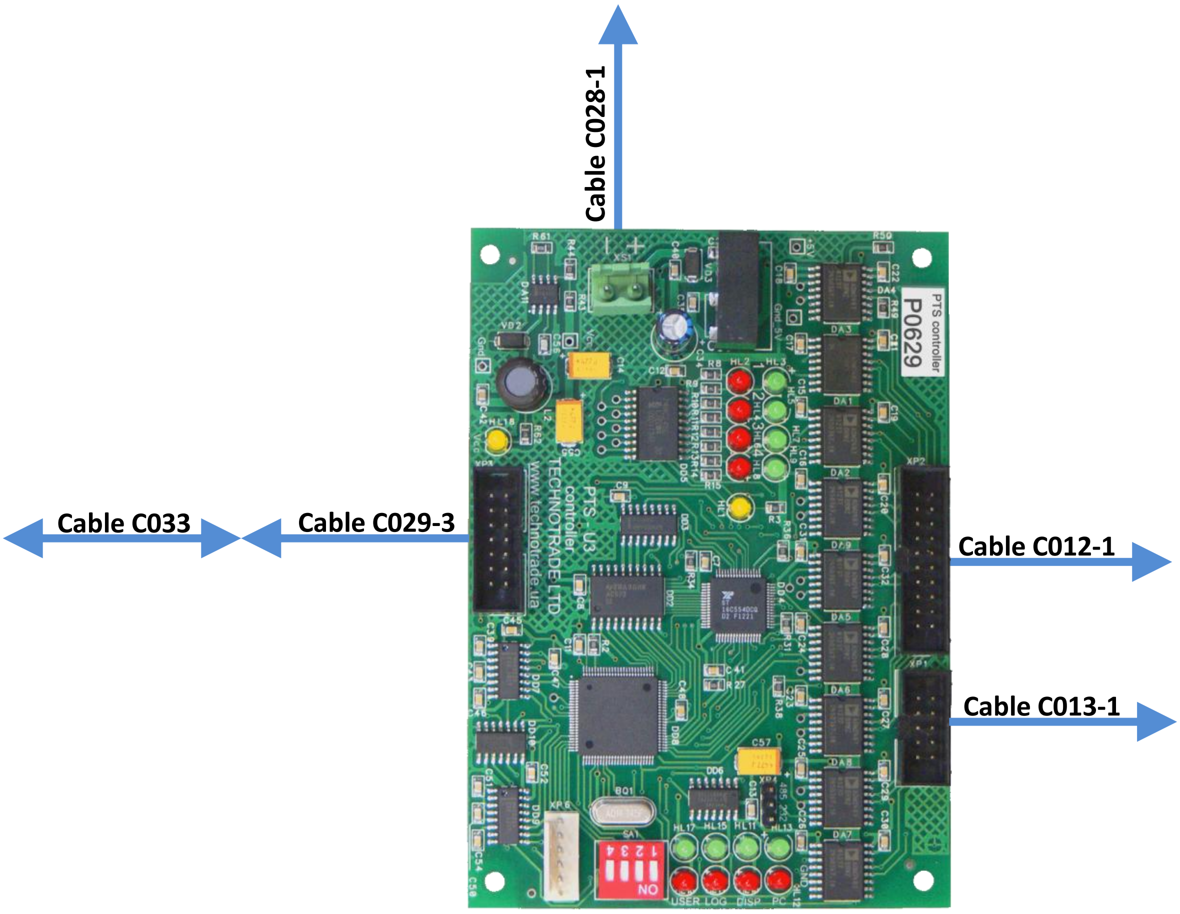
## *PCB board bottom view*



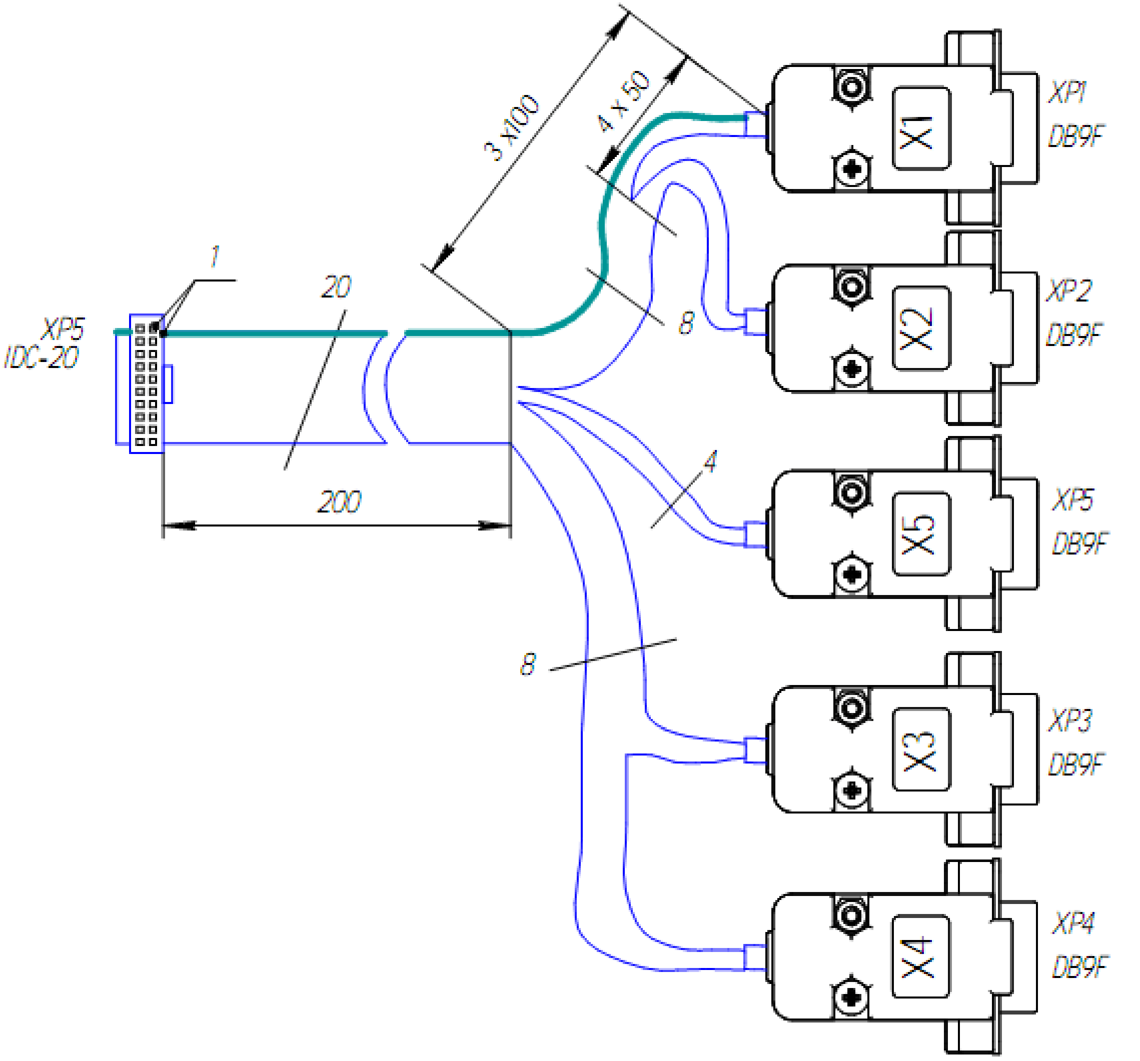
## Board dimensions



# PTS CONTROLLER CABLINGS

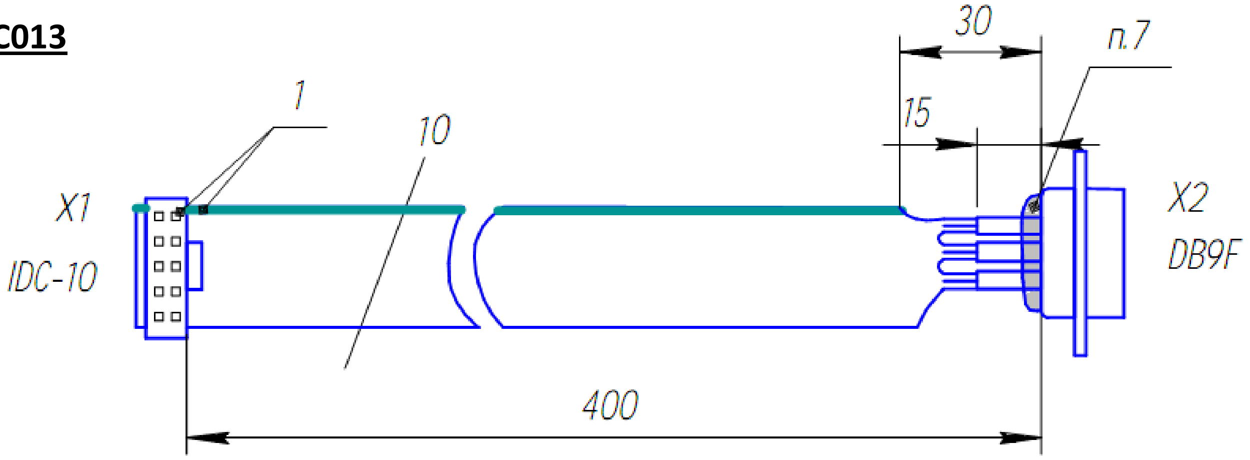


## CABLE C012-1

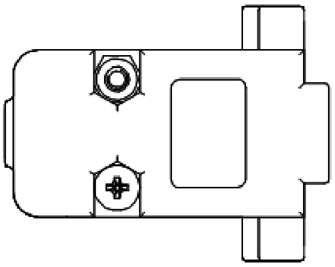


| XP5 | XP1 |
|---|---|
| 1 | 1 |
| 3 | 3 |
| 5 | 5 |
| 7 | 7 |
| | XP2 |
| 2 | 1 |
| 4 | 3 |
| 6 | 5 |
| 8 | 7 |
| | XP3 |
| 13 | 1 |
| 15 | 3 |
| 17 | 5 |
| 19 | 7 |
| | XP4 |
| 14 | 1 |
| 16 | 3 |
| 18 | 5 |
| 20 | 7 |
| | XP5 |
| 9 | 1 |
| 10 | |
| 11 | 3 |
| 12 | |

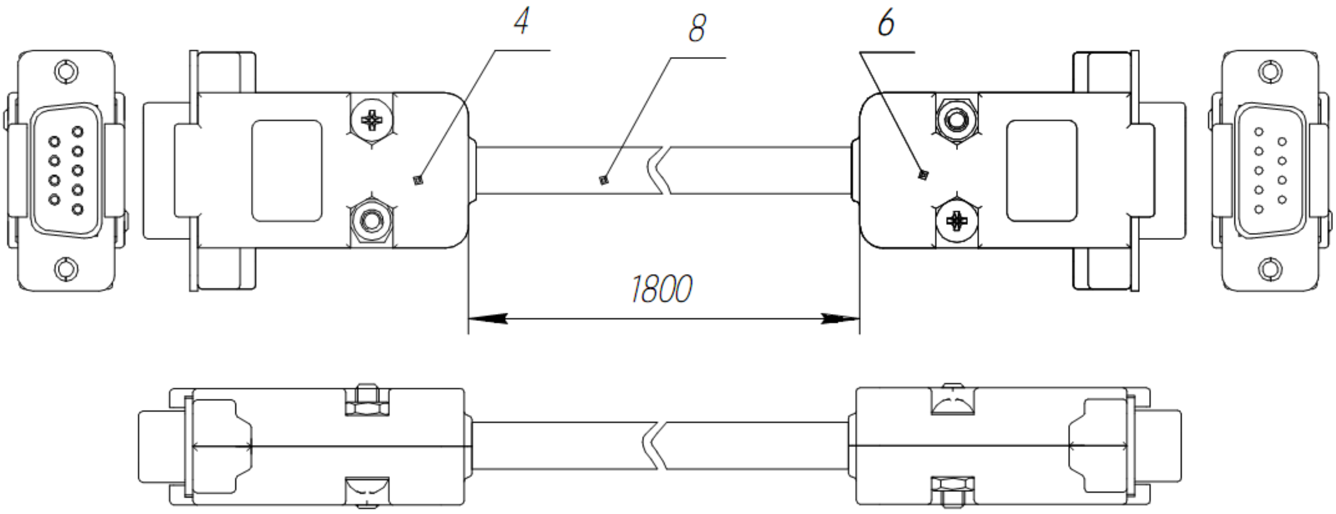
## CABLE C013



| XP1 | XP2 |
|---|---|
| 3, 4 | 5 |
| 5, 6 | 2 |
| 9, 10 | 3 |
| 1, 2 | - |
| 7, 8 | - |

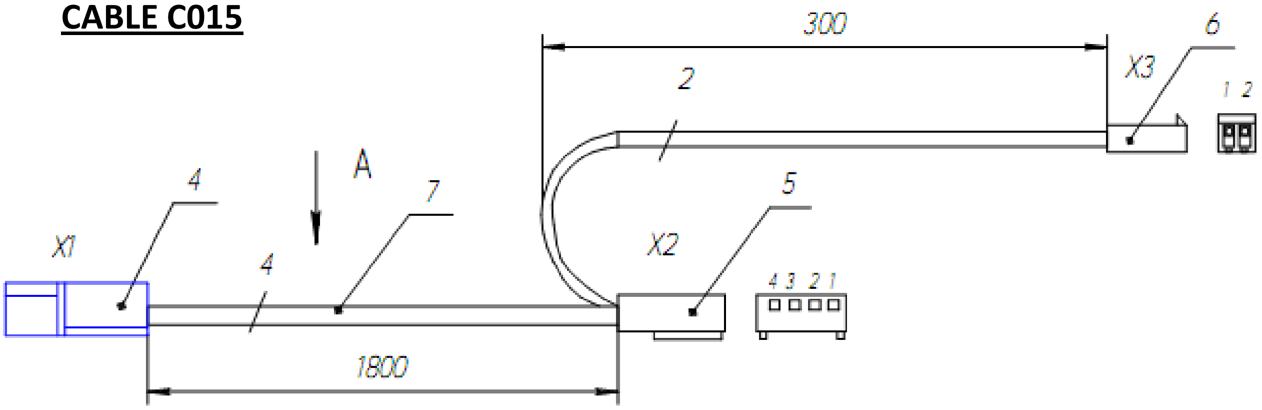


## CABLE C014-1

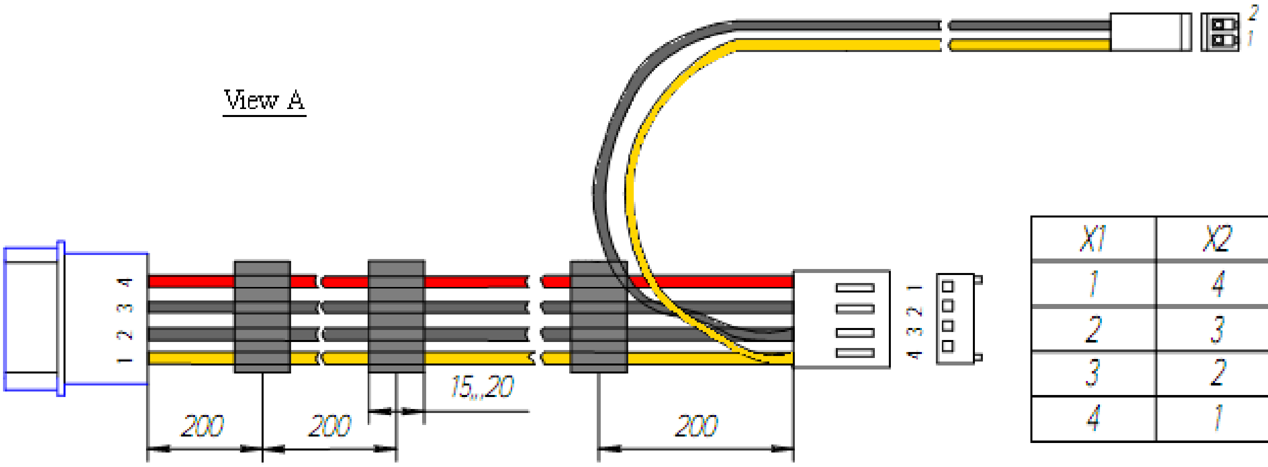


| X1 - DB-9F | X2 - DB-9M |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 5 | 5 |
| 7 | 7 |

## CABLE C015

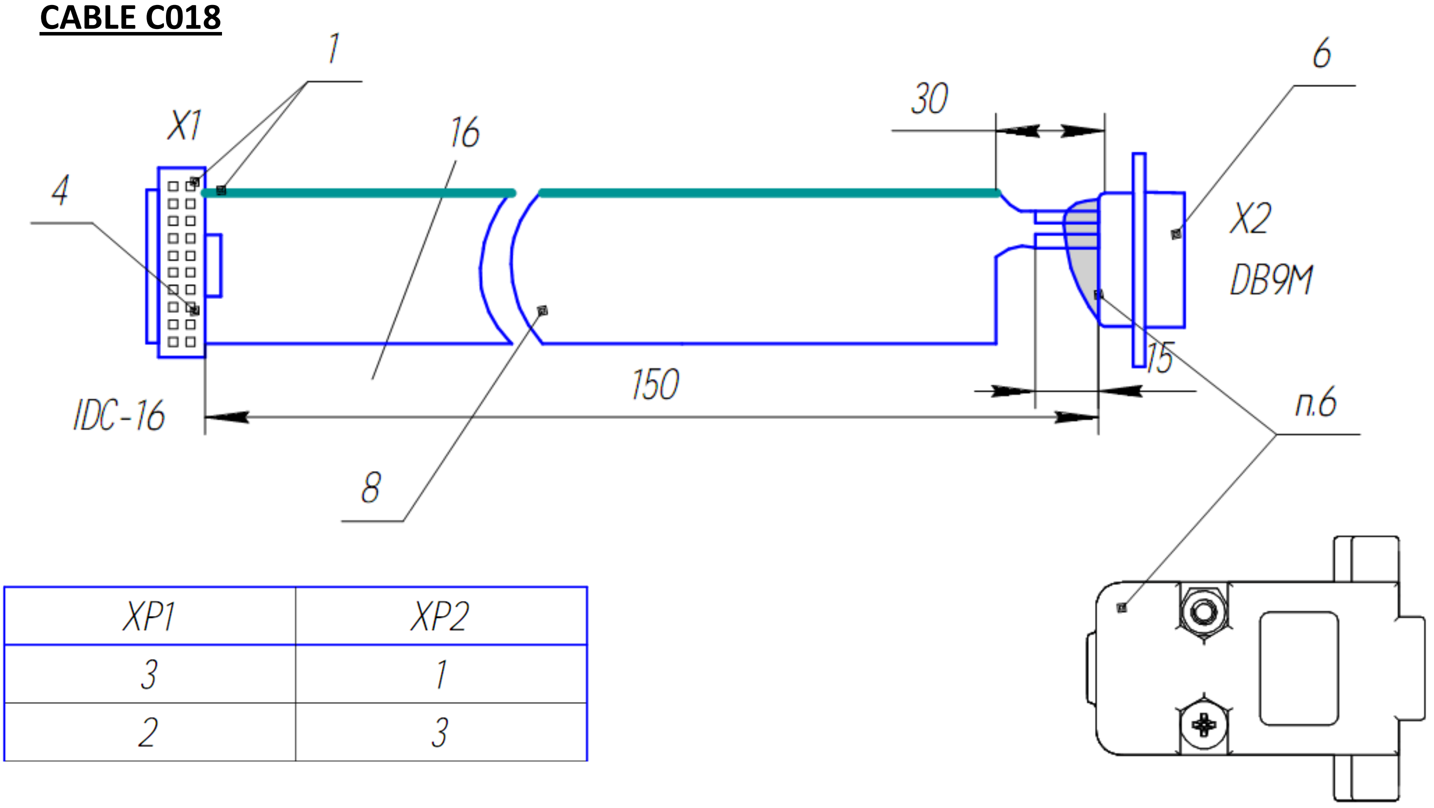


View A



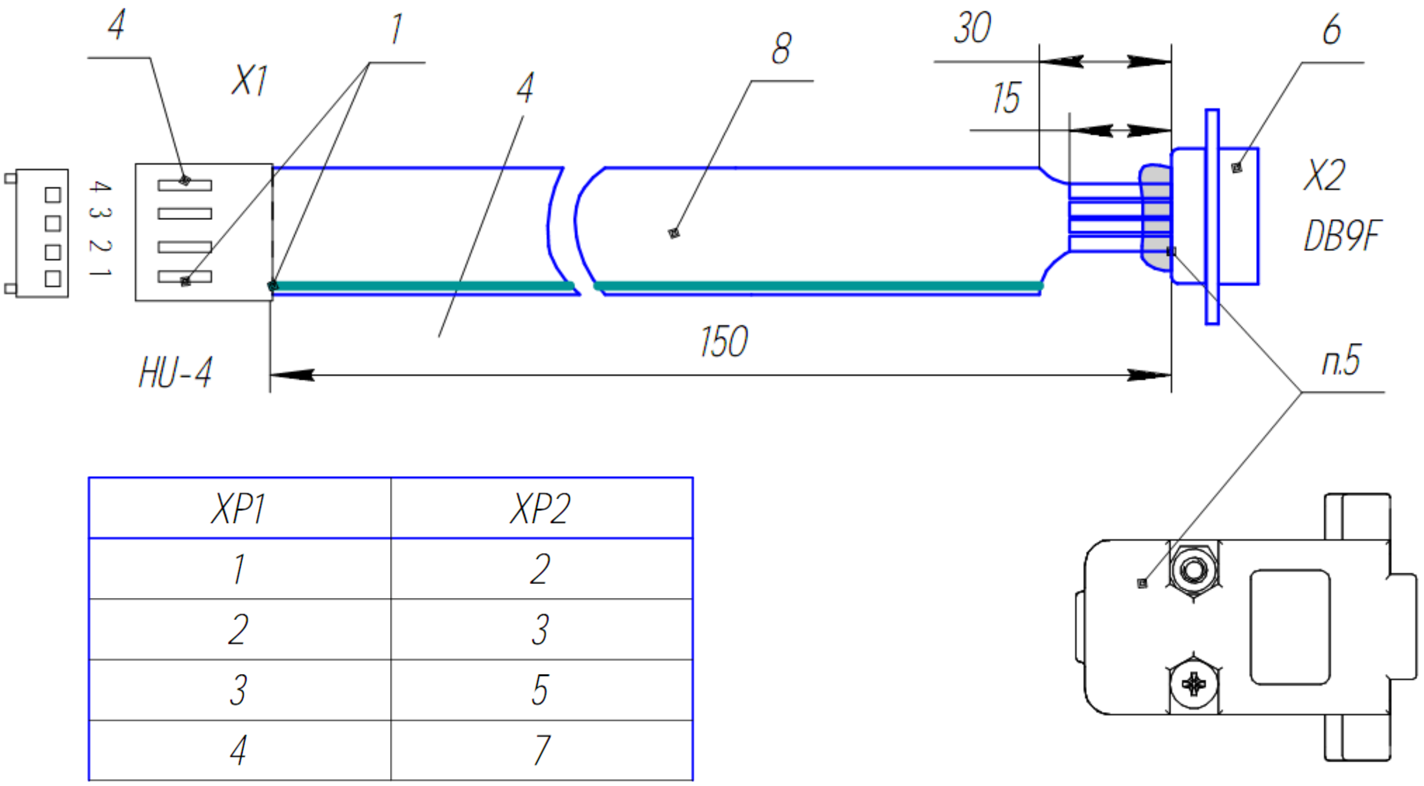
| X1 | X2 | X3 |
|---|---|---|
| 1 | 4 | 1 |
| 2 | 3 | 2 |
| 3 | 2 | |
| 4 | 1 | |

## CABLE C018



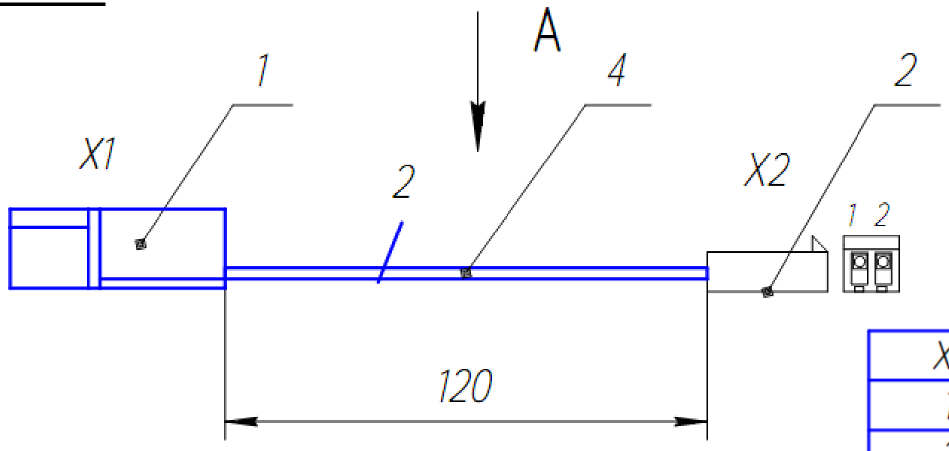
| XP1 | XP2 |
|---|---|
| 3 | 1 |
| 2 | 3 |

## CABLE C020

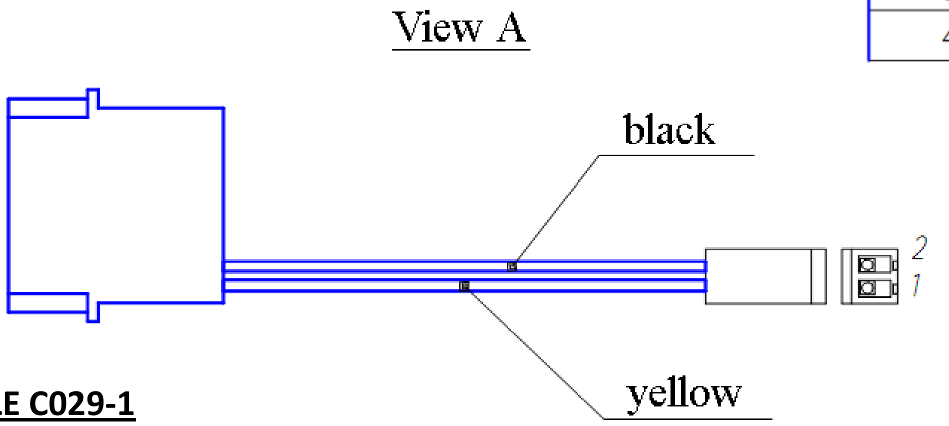


| XP1 | XP2 |
|---|---|
| 1 | 2 |
| 2 | 3 |
| 3 | 5 |
| 4 | 7 |

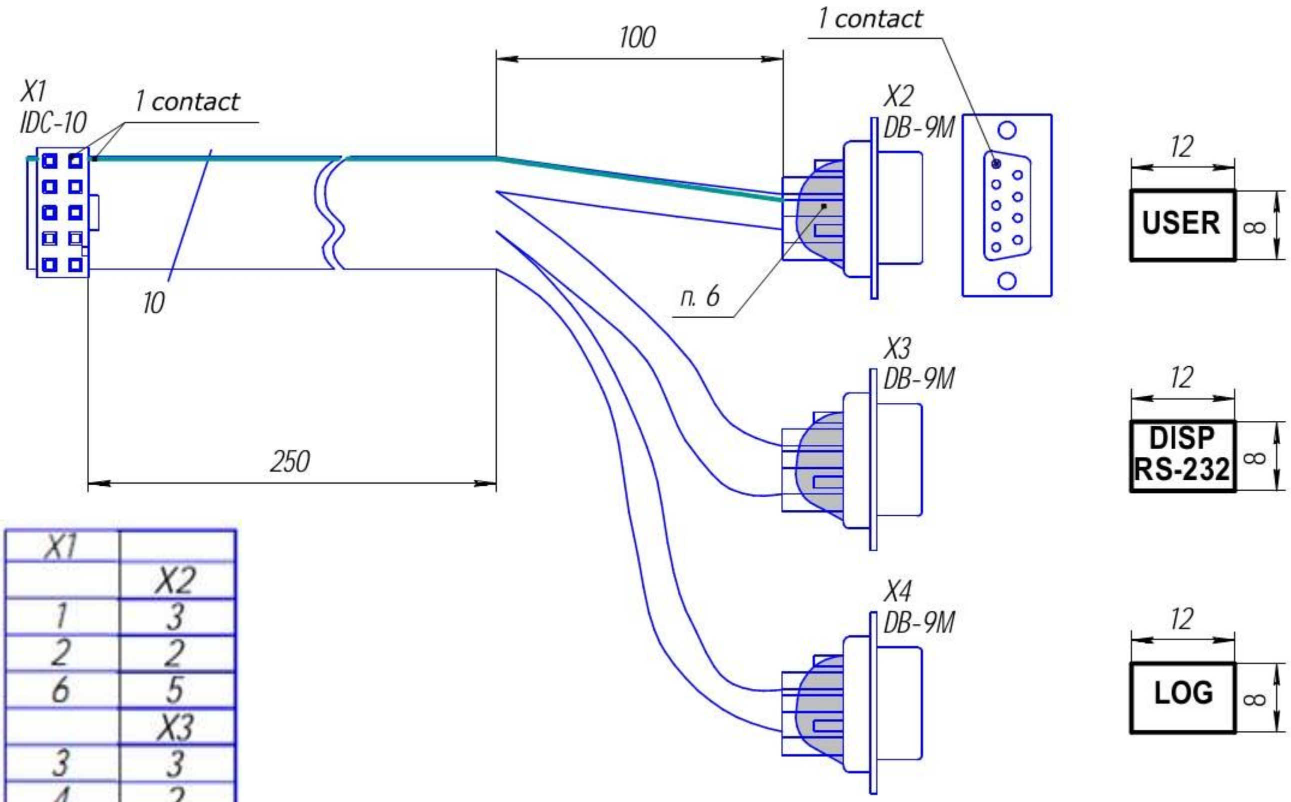
## CABLE C028



| X1 | X2 |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | |
| 4 | |

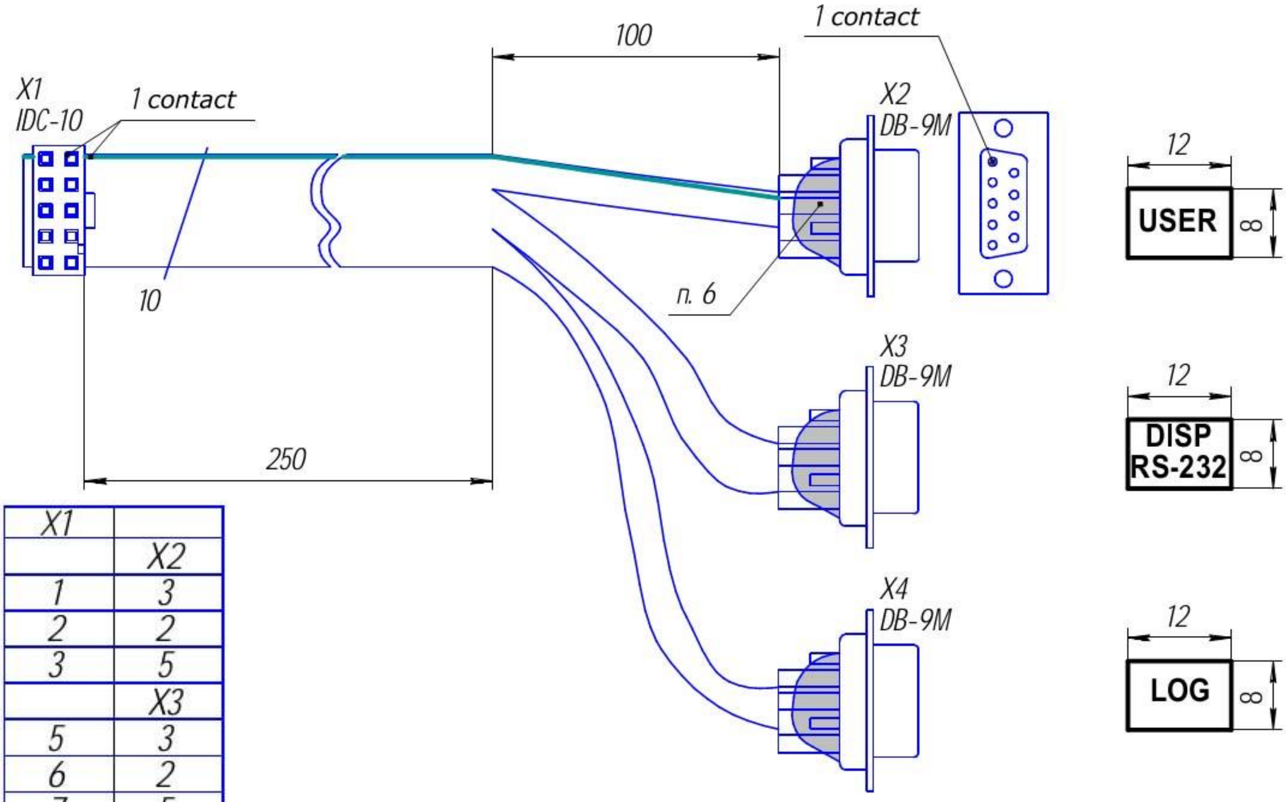


## CABLE C029-1



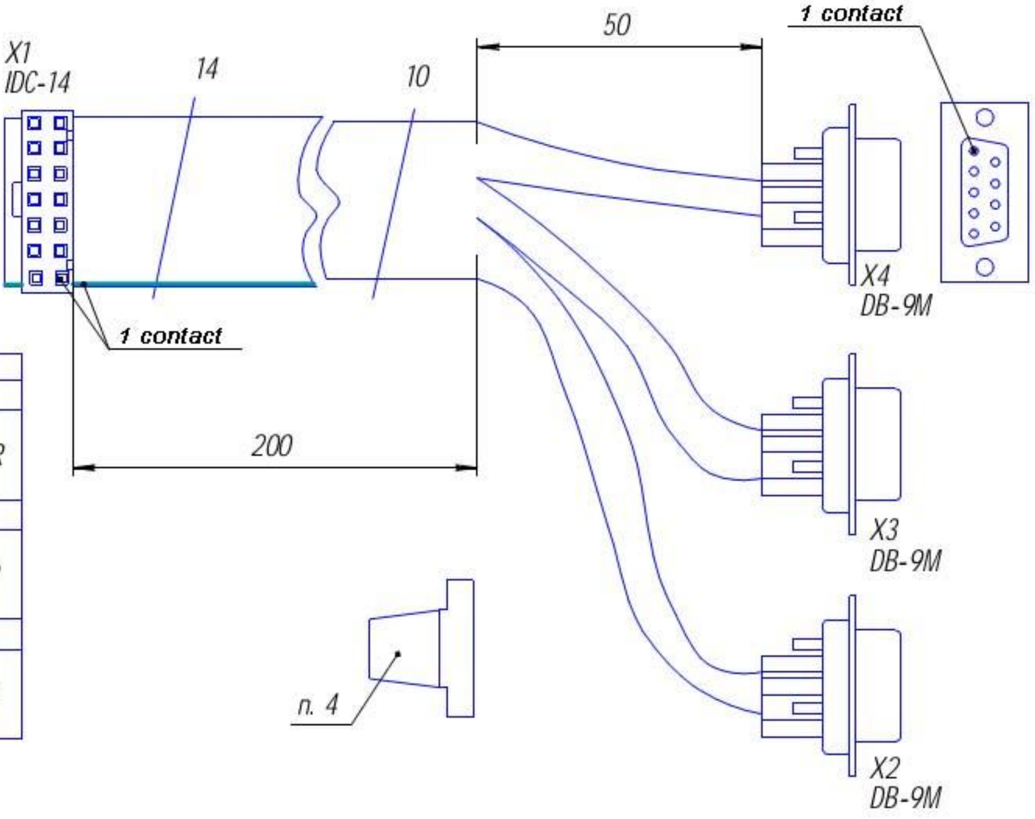
| X1 | |
|---|---|
| | X2 |
| 1 | 3 |
| 2 | 2 |
| 6 | 5 |
| | X3 |
| 3 | 3 |
| 4 | 2 |
| 9 | 5 |
| | X4 |
| 7 | 3 |
| 8 | 2 |
| 10 | 5 |

## CABLE C029-2



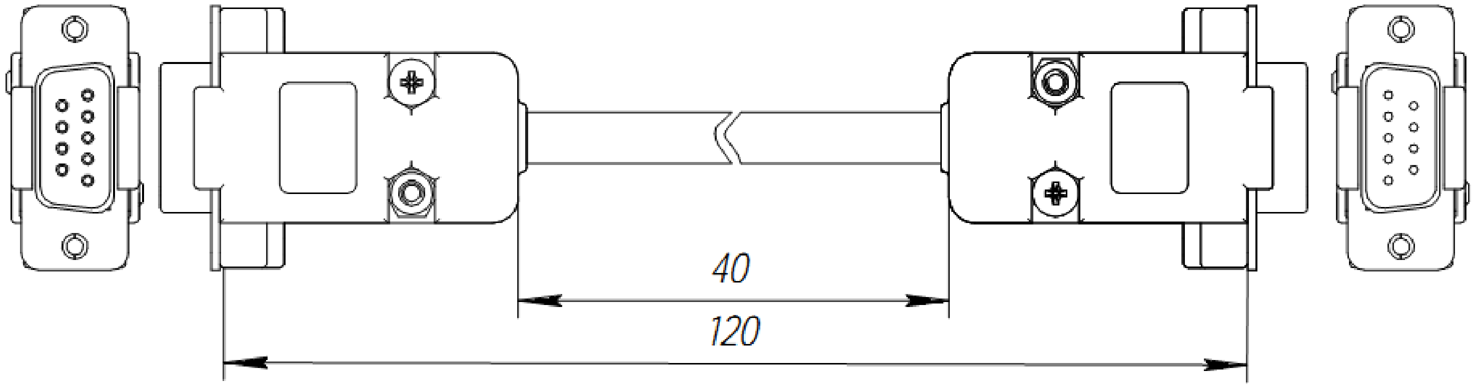
| X1 | |
|---|---|
| | X2 |
| 1 | 3 |
| 2 | 2 |
| 3 | 5 |
| | X3 |
| 5 | 3 |
| 6 | 2 |
| 7 | 5 |
| | X4 |
| 9 | 3 |
| 10 | 2 |
| 8 | 5 |

## CABLE C029-3



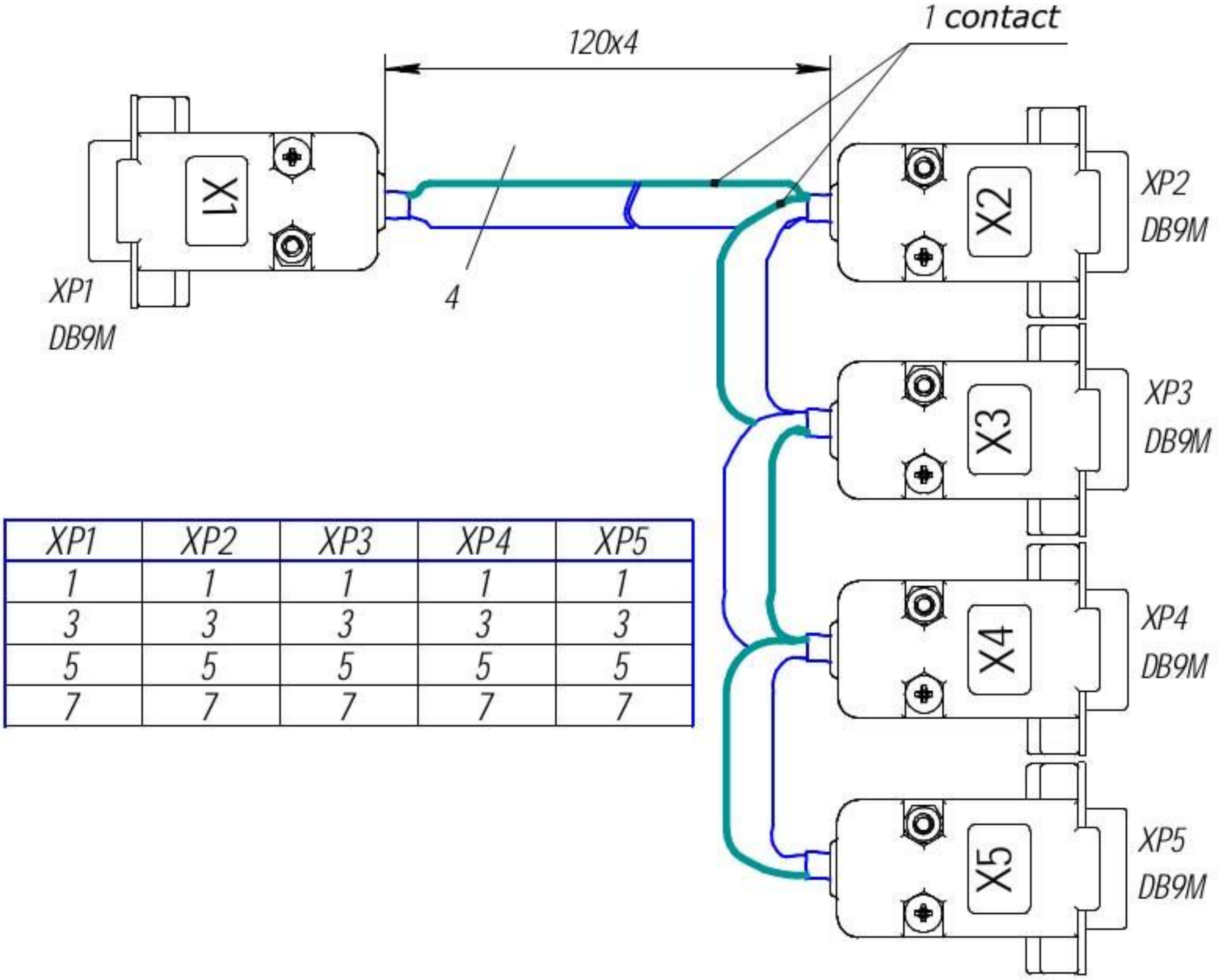
| X1 | | |
|---|---|---|
| | X2 | |
| 3 | 3 | USER |
| 4 | 2 | |
| 5 | 5 | |
| | X3 | |
| 7 | 3 | DISP |
| 8 | 2 | |
| 9 | 5 | |
| | X4 | |
| 11 | 3 | LOG |
| 12 | 2 | |
| 10 | 5 | |

## CABLE C033



| X1 - DB-9**F** | X2 - DB-9**F** |
|---|---|
| 2 | 3 |
| 3 | 2 |
| 5 | 5 |

## CABLE C055

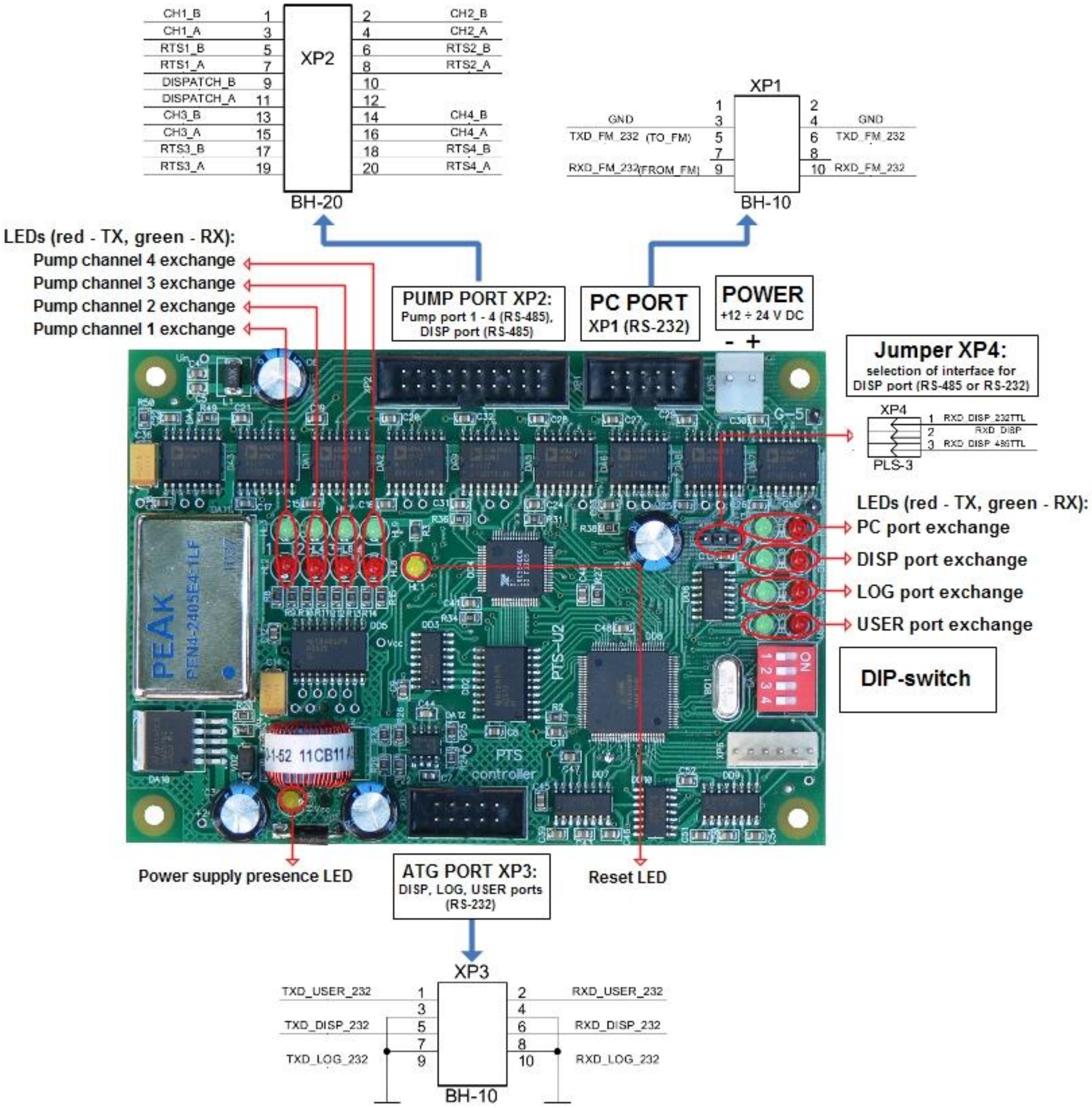


| XP1 | XP2 | XP3 | XP4 | XP5 |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 3 | 3 | 3 | 3 | 3 |
| 5 | 5 | 5 | 5 | 5 |
| 7 | 7 | 7 | 7 | 7 |

# PREVIOUS HARDWARE VERSIONS OF PTS CONTROLLER BOARD

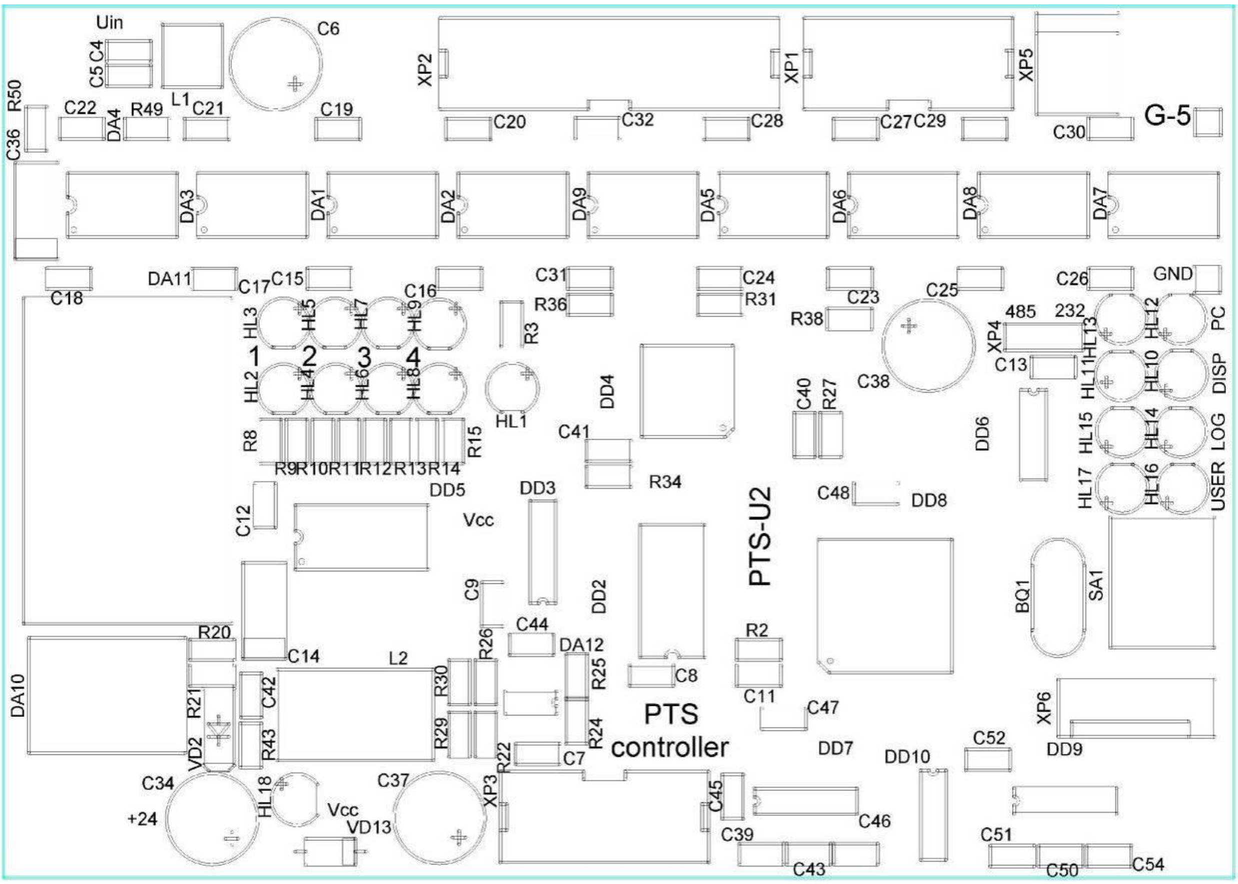
## *PTS-U2 board modification*

### *PTS-U2 board connectors and interfaces*

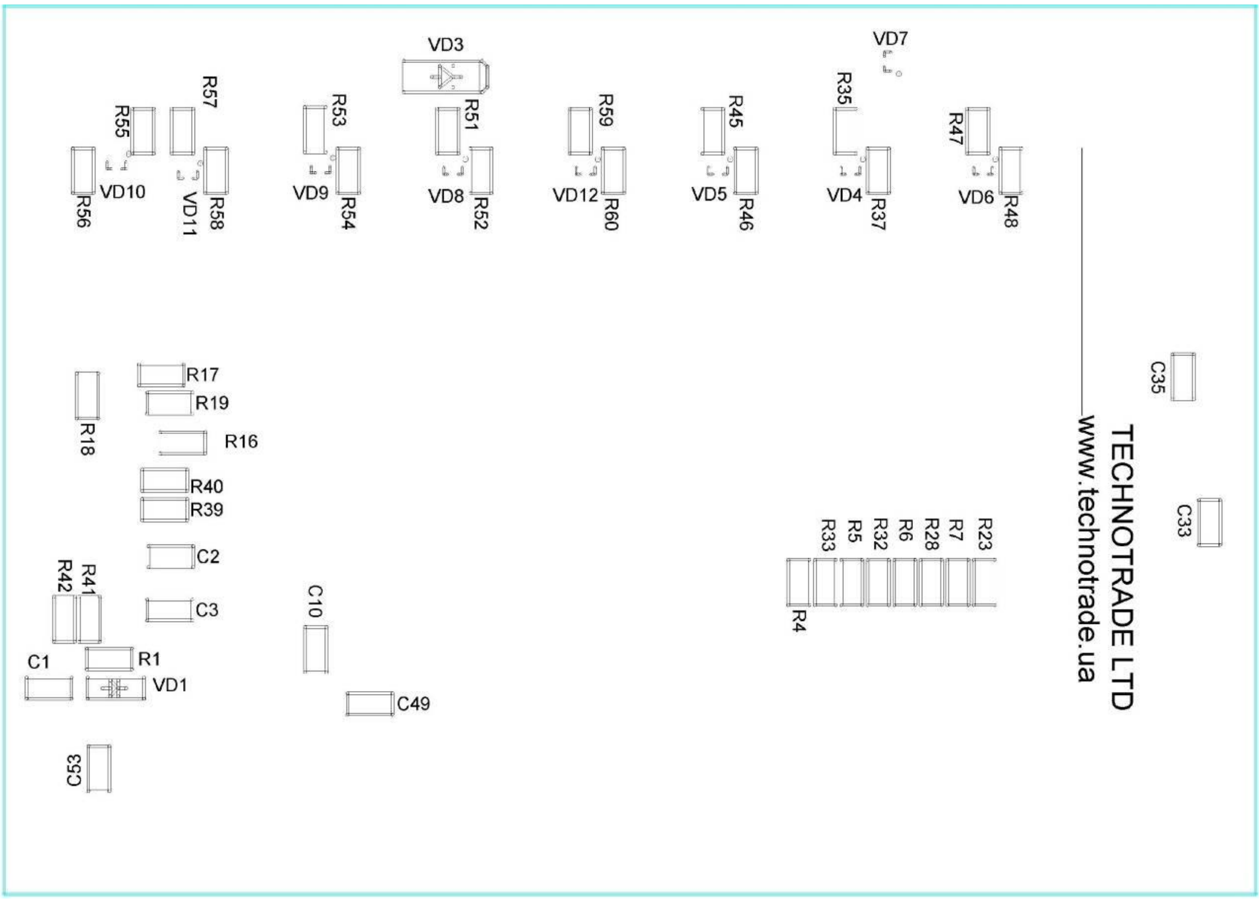


## *PTS-U2 board*

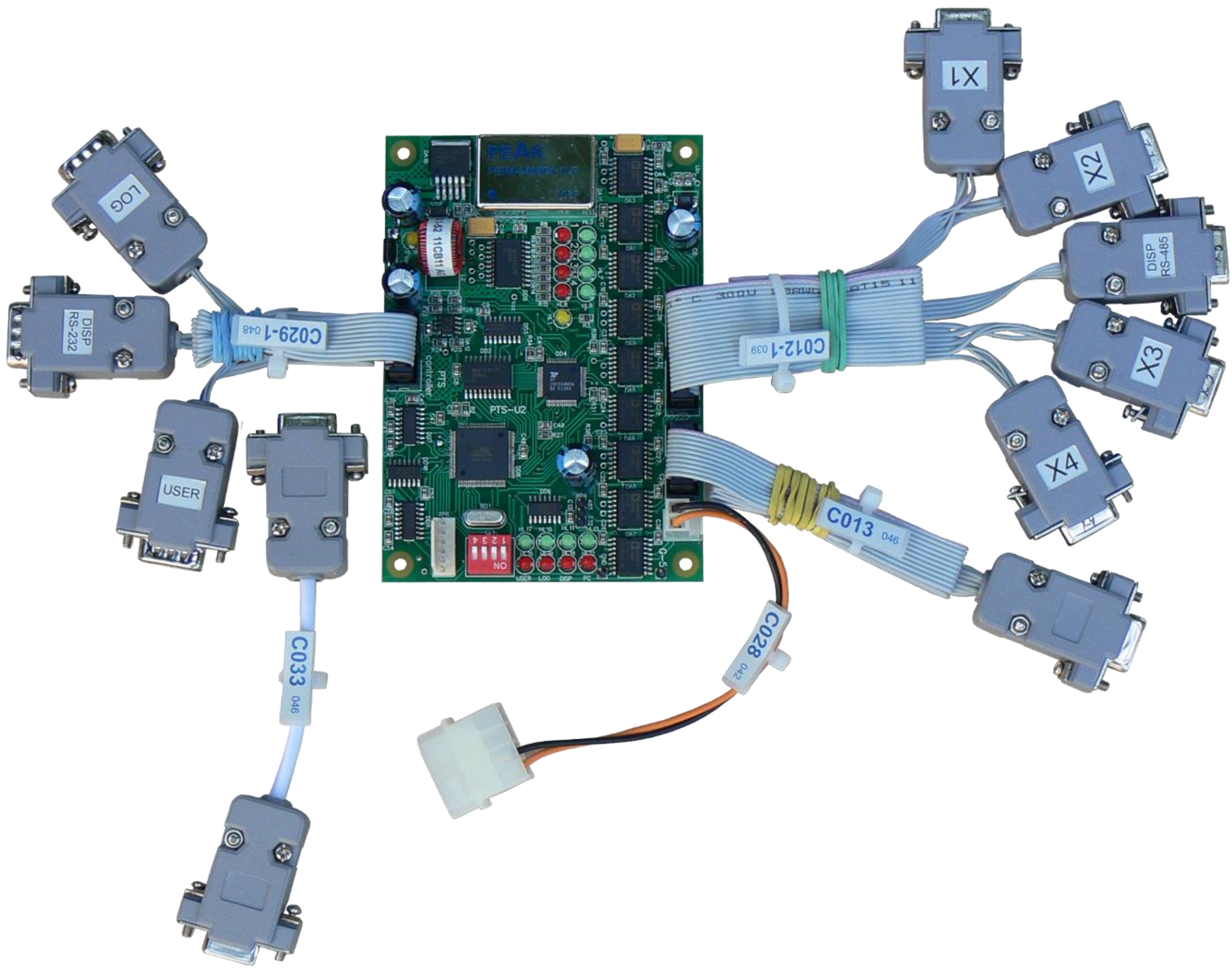
### *Top view*

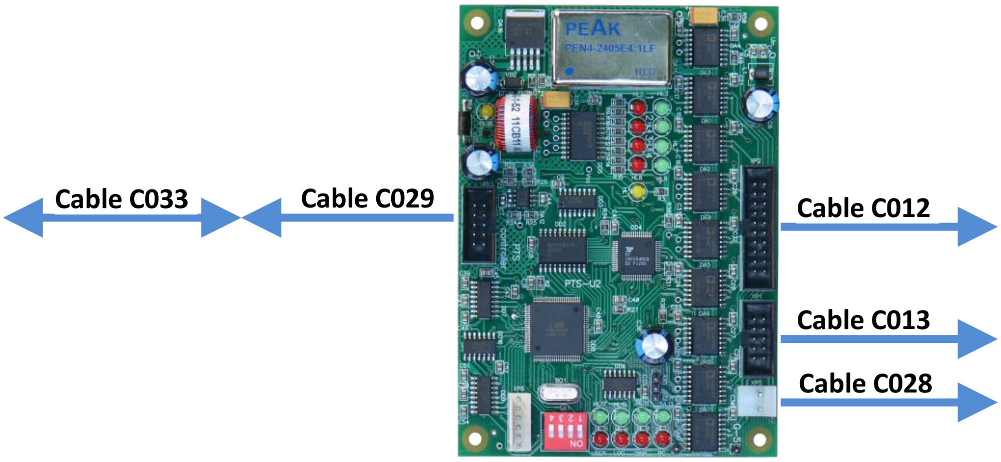


### *Bottom view*



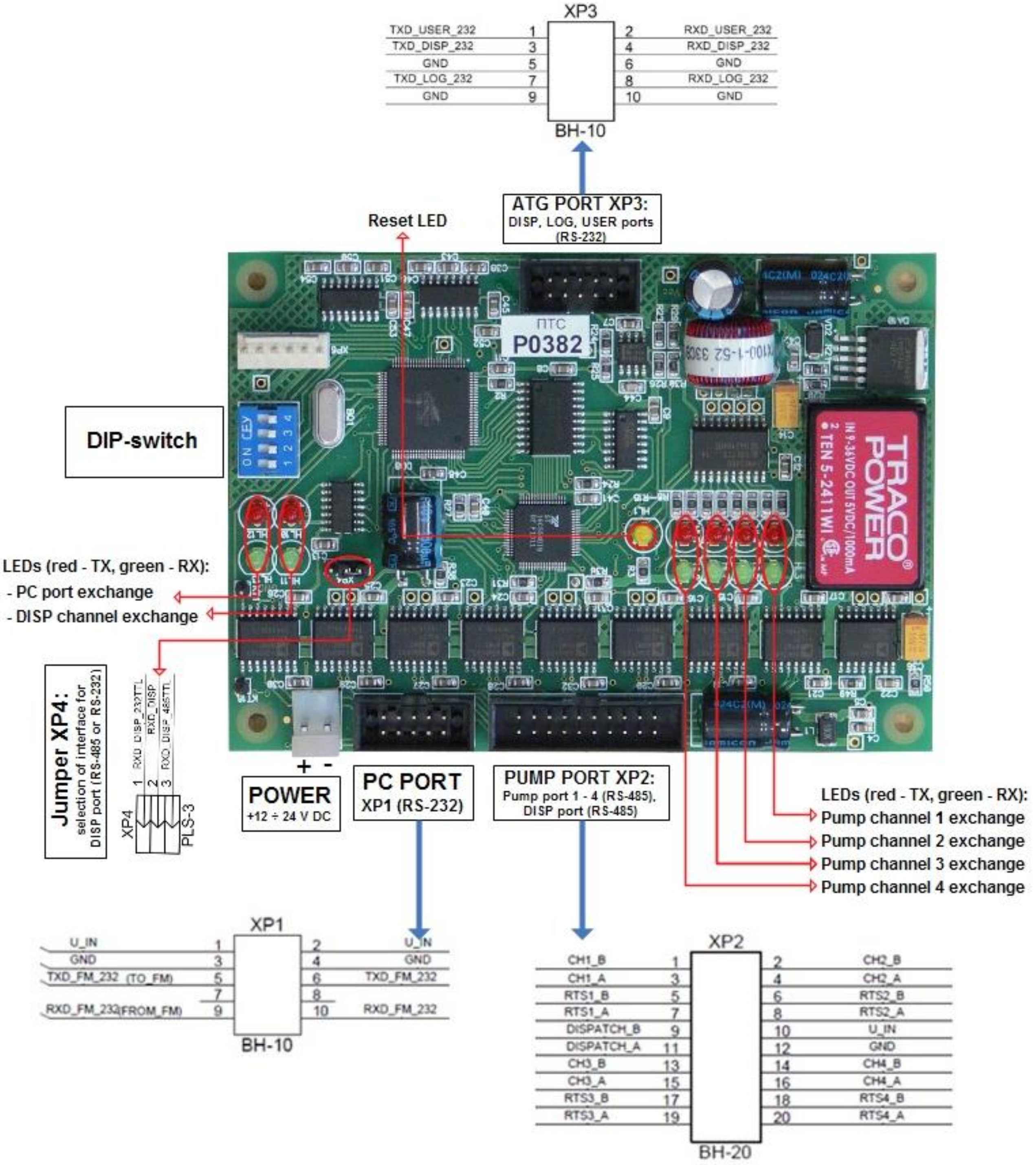
## *PTS-U2 board cabling*





## *PTS-U board modification*

### *PTS-U board connectors and interfaces*



XP3 (BH-10)

| | | | |
|---|---|---|---|
| TXD_USER_232 | 1 | 2 | RXD_USER_232 |
| TXD_DISP_232 | 3 | 4 | RXD_DISP_232 |
| GND | 5 | 6 | GND |
| TXD_LOG_232 | 7 | 8 | RXD_LOG_232 |
| GND | 9 | 10 | GND |

XP1 (BH-10)

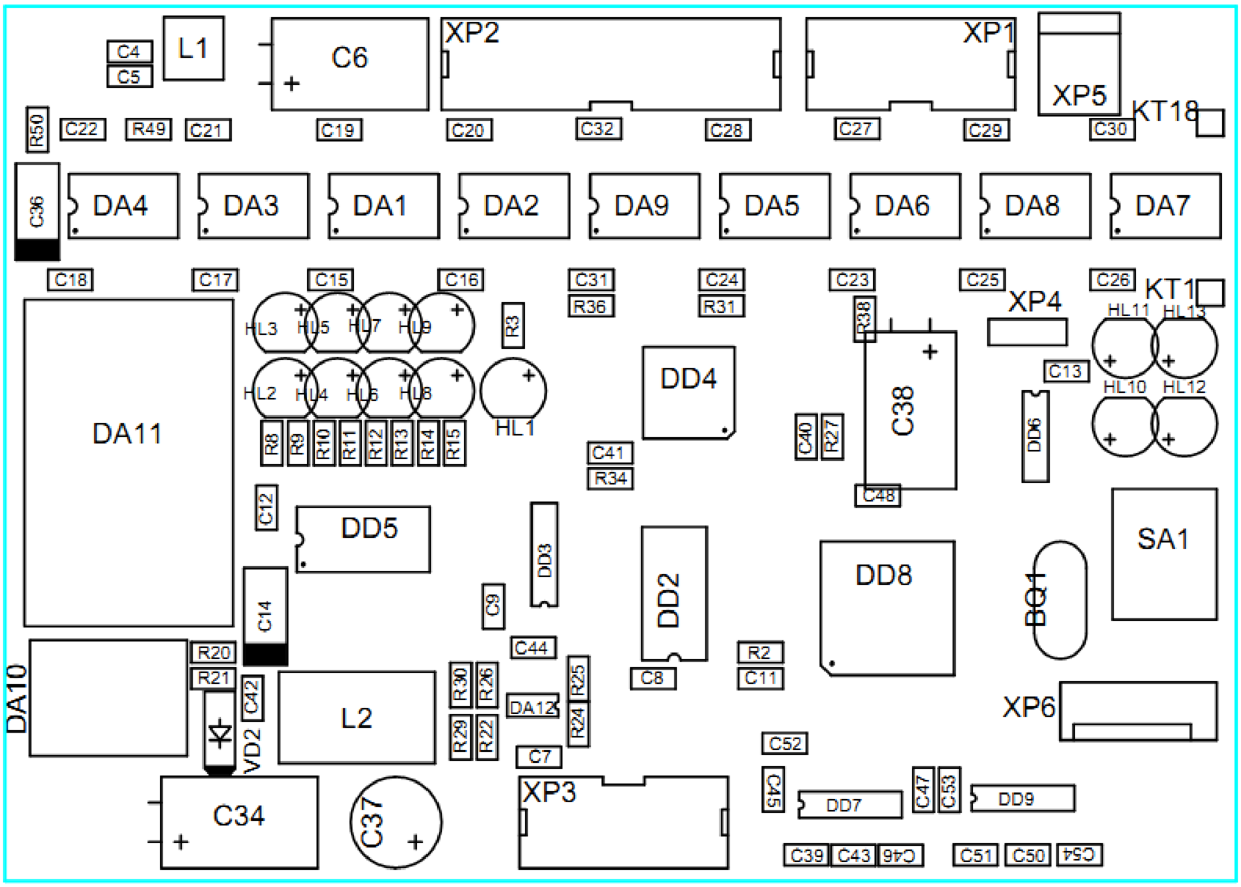
| | | | |
|---|---|---|---|
| U_IN | 1 | 2 | U_IN |
| GND | 3 | 4 | GND |
| TXD_FM_232 (TO_FM) | 5 | 6 | TXD_FM_232 |
| | 7 | 8 | |
| RXD_FM_232(FROM_FM) | 9 | 10 | RXD_FM_232 |

XP2 (BH-20)

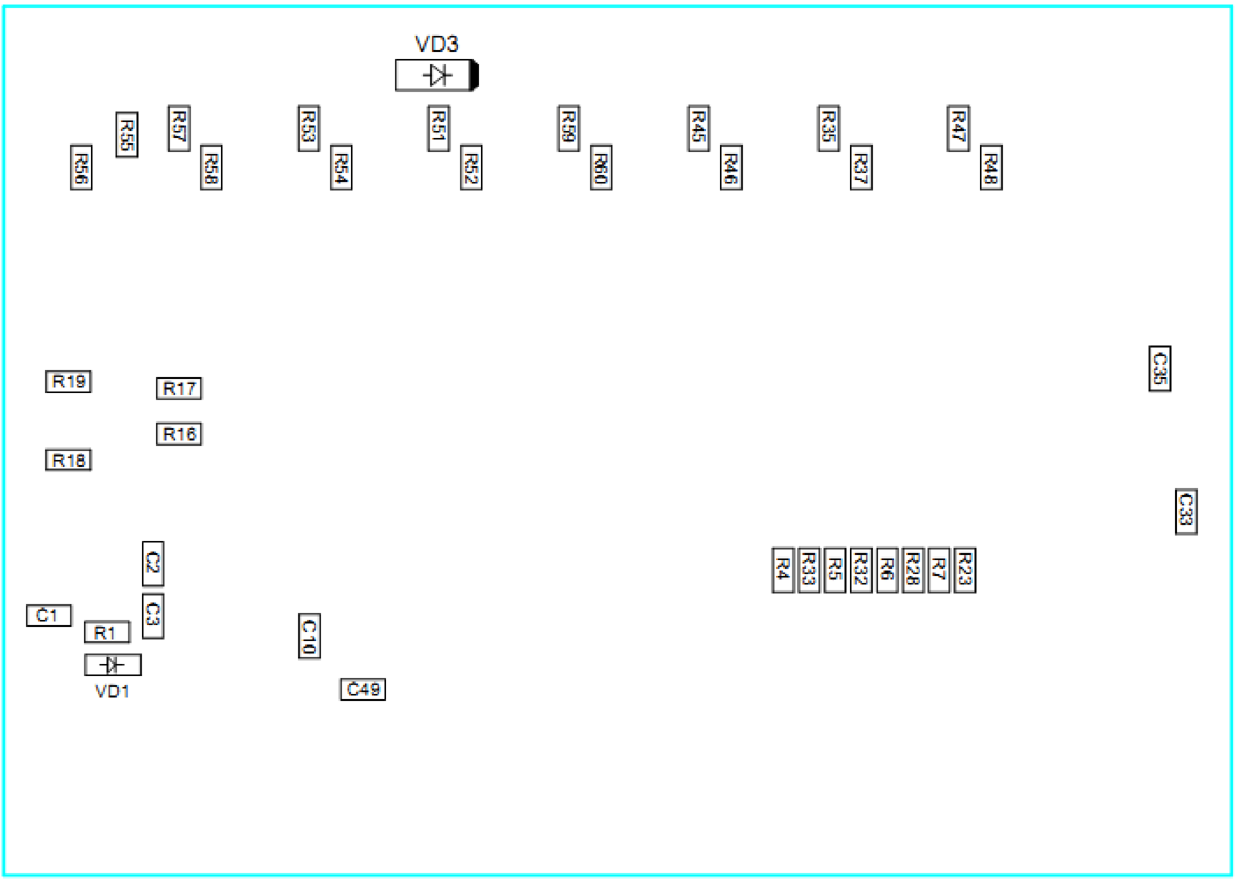
| | | | |
|---|---|---|---|
| CH1_B | 1 | 2 | CH2_B |
| CH1_A | 3 | 4 | CH2_A |
| RTS1_B | 5 | 6 | RTS2_B |
| RTS1_A | 7 | 8 | RTS2_A |
| DISPATCH_B | 9 | 10 | U_IN |
| DISPATCH_A | 11 | 12 | GND |
| CH3_B | 13 | 14 | CH4_B |
| CH3_A | 15 | 16 | CH4_A |
| RTS3_B | 17 | 18 | RTS4_B |
| RTS3_A | 19 | 20 | RTS4_A |

## *PTS-U board*

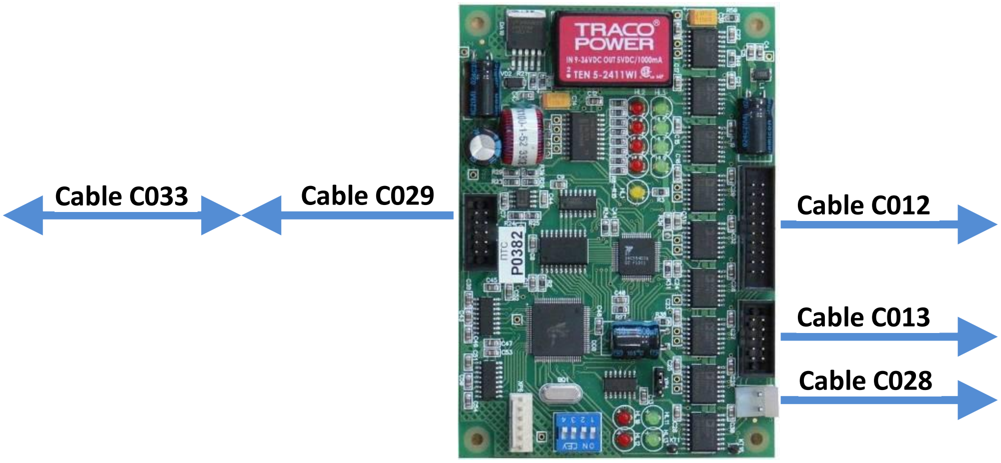
### *Top view*



### *Bottom view*



## PTS-U board cabling


Cable C033
Cable C029
Cable C012
Cable C013
Cable C028
TRACO POWER
P0382