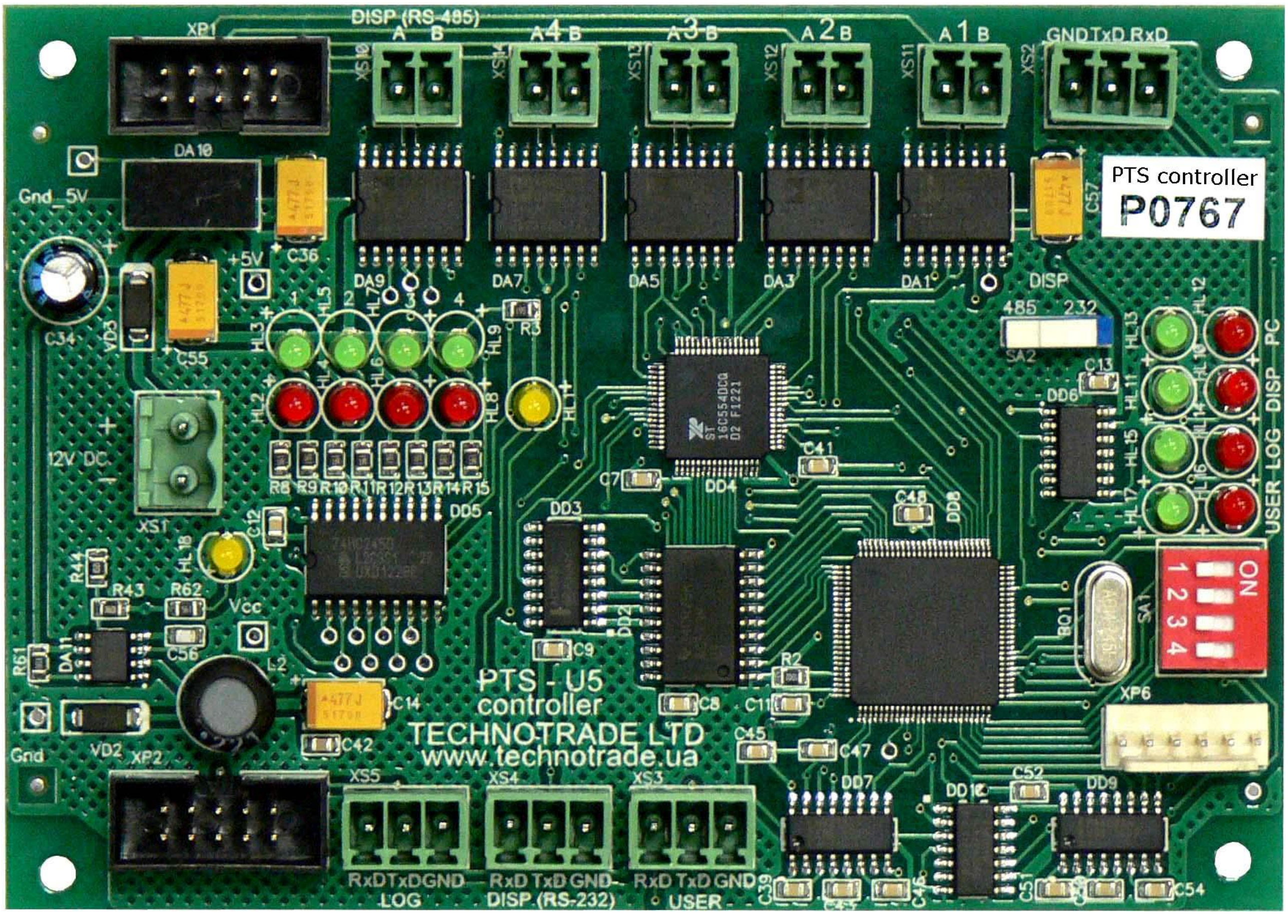
PTS controller

over fuel dispensers and ATG systems for petrol stations



TECHNICAL GUIDE

(PTS controller PCB board modification: PTS-U5)

Review date: 22 October, 2014

TECHNOTRADE LTD

CONTENT

REVISION HISTORY

REV	DATE	BY	SECTION	DESCRIPTION
R03	22.10.2014	Evgeniy Vasyliev	Installation requirements for petrol station	Requirements to power supply, requirements to grounding, requirements to laying of cable communications
R02	04.07.2014	Evgeniy Vasyliev	PTS controller configuration utility	Introduction of new PTS controller configuration and testing utility
R01	30.10.2013	Evgeniy Vasyliev	All	First release of PTS-U5 board revision

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-U5). Older versions of the PTS controller board (PTS-U3, PTS-U2 and PTS-U) and their corresponding connection schemes can be found in technical guide for PTS-U3 controller, which can be downloaded from PTS controller web-page: *http://www.technotrade.ua/fuel_pump_controller.html*.

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*

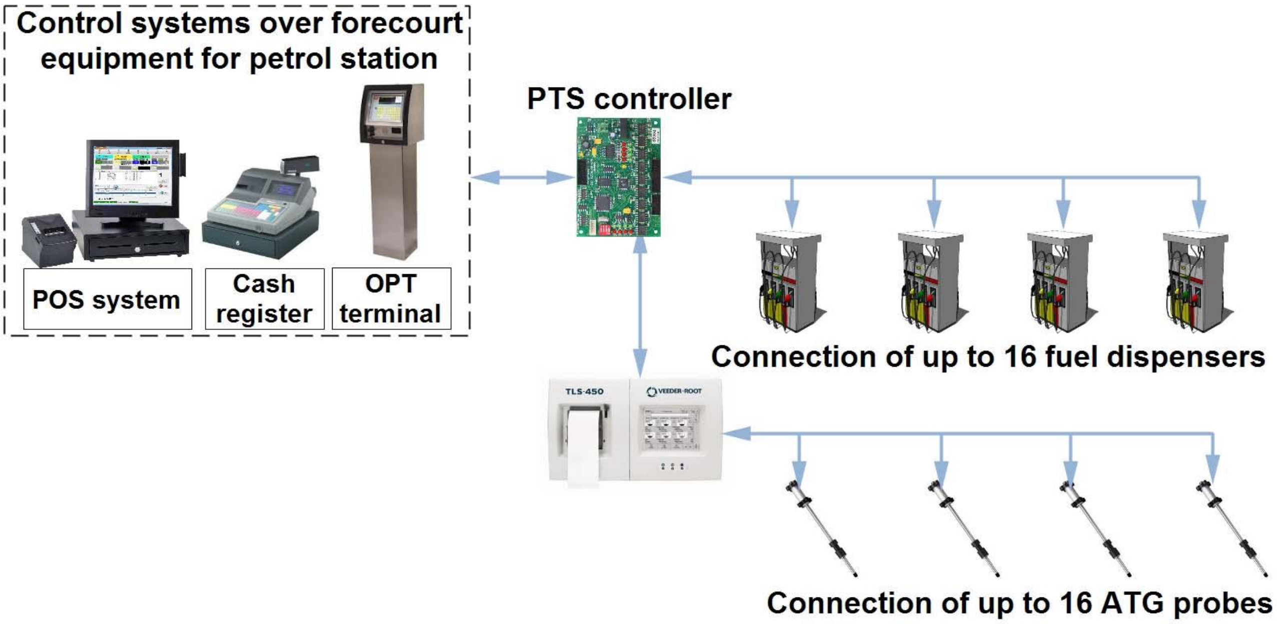
TECHNICAL FEATURES

Appointment

PTS controller over fuel dispensers and ATG (automatic tank gauge) systems for petrol stations serves as a protocols converter. It allows conversion of proprietary communication protocols of dispensers and ATG systems to a single communication protocol of PTS controller. Thus using a single communication protocol of PTS controller a control system (POS system, cash register or payment terminal) is able to provide control over any of the supported dispensers and ATG systems in exactly the same way. A work for a developer of the control system is completely simplified - he does not need to implement a great variety of the communication protocols in the control system (which are difficult to get from manufacturers and difficult to implement), developer needs to implement only protocol of PTS controller and PTS controller will itself undertake conversion to proprietary communication protocols of dispensers and ATG systems. Thus time required for development of the control system is sufficiently reduced.

PTS controller can be called a forecourt controller and is intended to be used in connection with a 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 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).


Control systems over forecourt equipment for petrol station
POS system
Cash register
OPT terminal
PTS controller
Connection of up to 16 fuel dispensers
TLS-450
VEEDER-ROOT
Connection of up to 16 ATG probes

Specification

##	PARAMETER	VALUE
1	**Power supply voltage**	12 V DC (24 V DC under request)
2	**Current consumption**	200 mA max
3	**Temperature range**	-40°C ÷ +80°C
4	**Weight**	120 g
5	**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	2A	S4-Dart
2	ADAST (ADAMOV SYSTEMS)	ADAST EasyCall
3	AGIRA	PumpControl GC21
4	ASPRO	PumpControl GC21
5	BARANSAY	GILBARCO Two-Wire
6	BATCHEN	GILBARCO Electroline
7	BATCHEN	GILBARCO MPP
8	BENNETT	BENNETT pump dispenser protocol (current loop)
9	BENNETT	BENNETT pump dispenser protocol (RS-485)
10	BLUE SKY	Blue Sky
11	DEVELCO	DEVELCO
12	DONG HWA PRIME	Prime pump interface
13	EMGAZ DRAGON	Marconi PumaLAN
14	EPCO	EPCO protocol specification
15	EUROPUMP	S4-Dart
16	FALCON LPG	FALCON
17	FUELSIS	S4-Dart
18	GALILEO	PumpControl GC21
19	GILBARCO	GILBARCO Two-Wire
20	GREENFIELD	GILBARCO Two-wire
21	HONG YANG	HongYang FZ-protocol
22	HONG YANG	HongYang 886 communication protocol
23	HONG YANG	HongYang 886 MPD communication protocol
24	IMW	PumpControl GC21
25	IMW	Tokheim UDC
26	KALVACHA	Kalvacha
27	KIEVNIIGAZ	TIT UniGaz
28	KOREA ENE	EnE Dispenser POS protocol
29	KPG-2	TIT UniGaz
30	KRAUS	MNET design specification
31	LANFENG	RS–232 Protocol of Dispenser
32	LG ENE	EnE Dispenser POS protocol
33	LOGITRON	Marconi PumaLAN
34	MEKSAN / WAYNE SU86	WAYNE Dart
35	MEKSER	S4-Dart
36	MEPSAN	S4-Dart
37	MIDCO	MIDCO Proprietary Pump Protocol

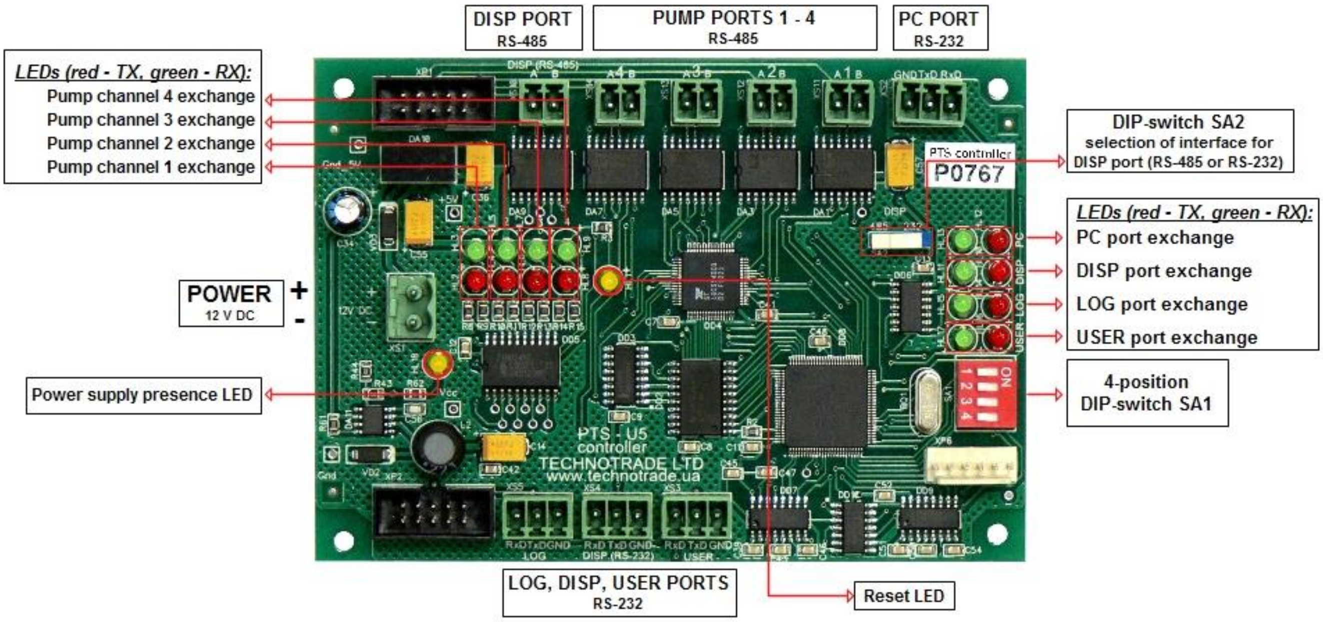
38	**NUOVO PIGNONE**	Nuovo Pignone
39	**PEC (GALLAGHER FUEL SYSTEMS)**	PEC Pump Communication Protocol
40	**PETPOSAN**	S4-Dart
41	**PETROLMECCANICA**	WAYNE Dart
42	**PETROTEC**	GILBARCO Two-Wire
43	**PROWALCO**	SPDC-1, MPDC-1
44	**PUMP CONTROL**	PumpControl GC21
45	**REAL-TECH**	Blue Sky
46	**SAFE**	SAFE Graf
47	**SALZKOTTEN**	ER3-Data interface
48	**SALZKOTTEN**	GILBARCO Two-Wire
49	**SANKI**	Sanki communication protocol
50	**SEA BIRD**	Blue Sky
51	**SHELF**	SHELF
52	**SLAVUTICH**	SLAVUTICH FD-Link
53	**SOMO PETRO**	POS protocol
54	**TATSUNO (JAPAN)**	TATSUNO SS-LAN
55	**TATSUNO EUROPE (FORMER BENC)**	TATSUNO Benc PDE
56	**TOKHEIM**	Tokheim Controller-Dispenser Communication protocol
57	**TOKICO**	Tokico SS-LAN
58	**TOMINAGA**	Tominaga SS-LAN
59	**TOPAZ**	TOPAZ Electro
60	**UNICON-TIT**	TIT UniPump
61	**WAYNE DRESSER**	WAYNE Dart
62	**WAYNE DRESSER**	Wayne US Current Loop
63	**WAYNE PIGNONE**	WAYNE Dart
64	**YENEN**	S4-Dart
65	**ZAP / MM PETRO**	MM PETRO ZAP RS-485

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

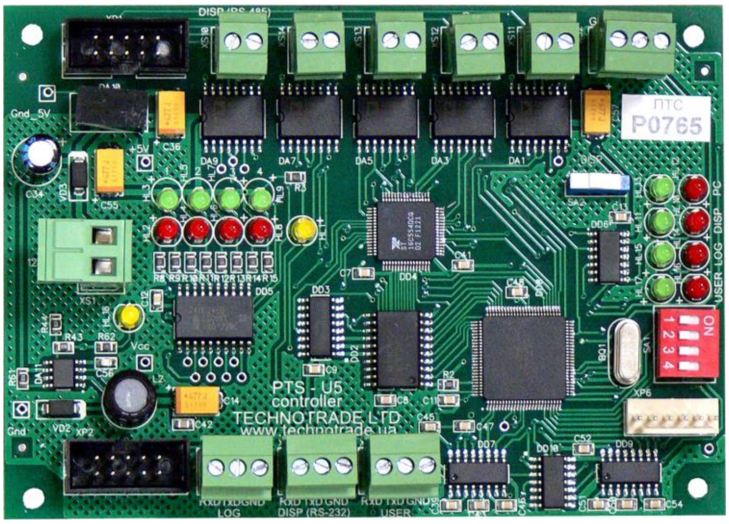
BOARD CONNECTORS AND INTERFACES



NOTE!

DIP-switch SA2 serves for selection of DISP channel interface between RS-485 and RS-232.

PTS controller is supplied together with terminal blocks for each of the connectors for screwing of connection wires.



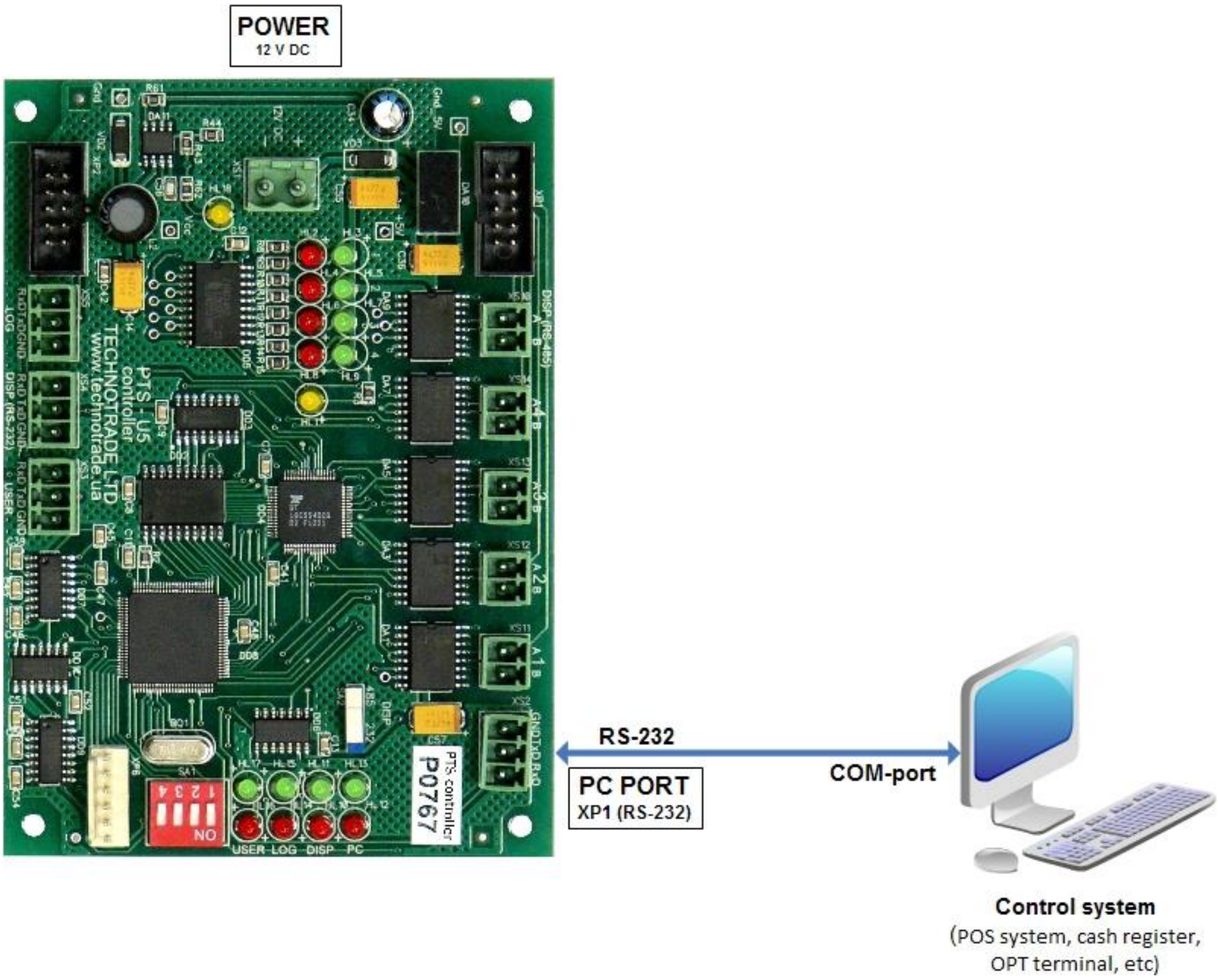
COMMUNICATION PORTS

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

Scheme of connections:



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, Ethernet-to-COM, Bluetooth-to-COM, other converters).

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.

INSTALLATION REQUIREMENTS FOR PETROL STATION

***WARNING!* Manufacturer guarantees reliable and stable operation of products only at compliance with below requirements. In case of absence of uninterruptible power supply or incorrect wiring of products to it any claims to malfunction of software are not accepted.**

1. Requirements to power supply

The described products come into structure of control system (POS) for petrol station. Power supply of the products should be done from a separate power supply with built-in filter of radio frequency interferences and limiter of high voltage pulse interferences. Power supply should have a safety factor of 1.5.

At emergency switching off the power supply or in case of power voltage exceeding its permitted ranges the products can switch off with loss or corruption of data and possible damage of hardware and software. Power supply of all electronic blocks of POS and electronic pumpheads of dispensers, which are connected through information lines, should be made from single common uninterruptible power supply source (UPS). Connection of other devices to given UPS is strictly prohibited. UPS should be of continuous action (online) and work with double conversion with output voltage regulation. UPS should have a safety factor of 1.5. Filter of radio frequency interferences and limiter of high voltage pulse interferences should be used for feeding equipment from UPS.

Supply of electronic pumpheads of dispensers should be made from the UPS unit using 3-wires scheme with isolated neutral through dedicated two-pole breaker for each dispenser. Connection of other parts of dispenser to UPS unit (expect electronic pumpheads) is strictly prohibited.

UPS unit should be connected to a separate three-pole socket fed through the three-wire feeder (phase, neutral, ground wires) with insulated neutral from a dedicated circuit breaker of switchboard. Feeder coming from the switchboard to the socket should located not closer than 0.3 meters to other feeders. The socket should be located at a distance of not more than 1 meter away from the POS. Phase wire of the feeder should not have any other consumer, which are sources of interferences (for example motors).

For protection of POS and UPS from secondary effects of atmospheric electricity it is required to install high-voltage arresters (dischargers) at the transformer substation or on poles of power lines.

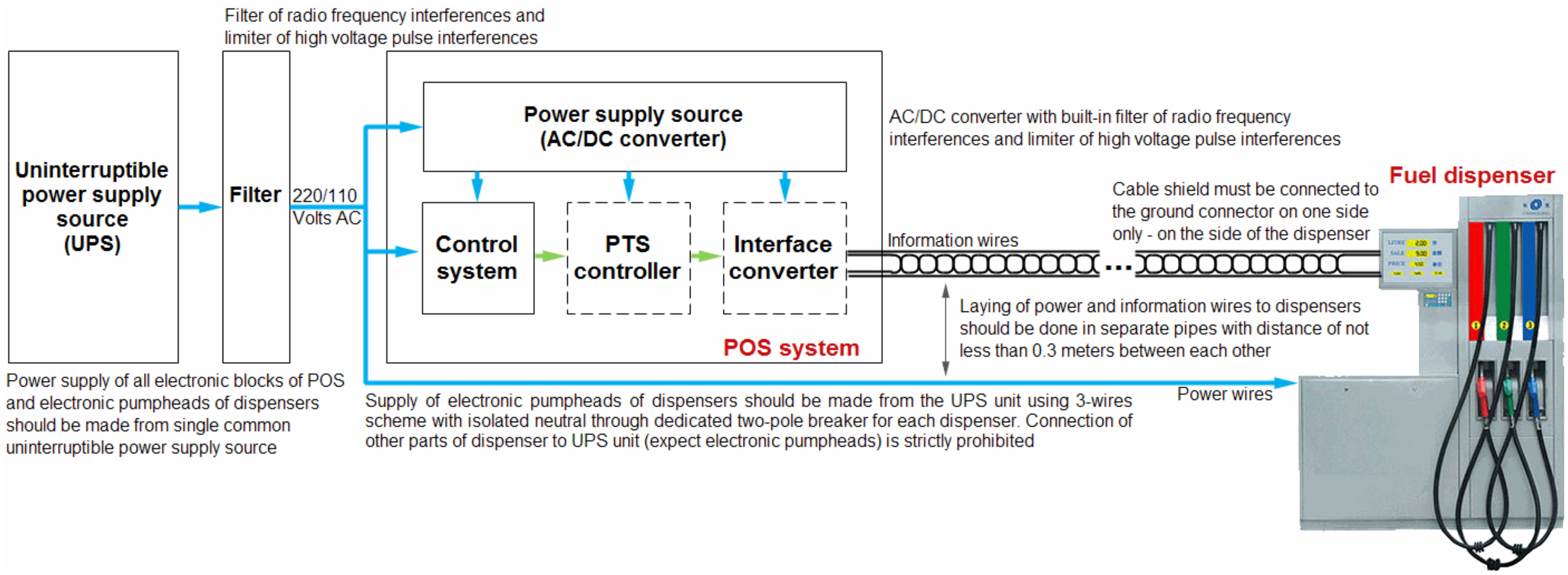
2. Requirements to grounding

In the switchboard the ground wire of feeder socket should be connected to the grounding screw, which should be connected by means of welding with a protection grounding circuit of petrol station by steel wire with a diameter of not less than 5 mm.

Protection grounding circuit of petrol station should correspond to safety requirements and be separated from the station lightning protection circuit. Distance from the nearest electrode of protection grounding circuit to electrode of lightning protection circuit must be at least 10 meters. Resistance of the protection grounding circuit should be not more than 4 Ohms and must be confirmed by the test report. Length of wires from the switchboard to the nearest electrode of protection grounding circuit should not exceed 15 meters.

3. Requirements to laying of cable communications

Laying of power and information wires to dispensers should be done in separate pipes with distance of not less than 0.3 meters between each other. For informational wires (current loops, RS-485, other interfaces) it is recommended to use shielded twisted-pair cables (recommended type – FTP CAT 5E). The cable shield must be connected to the ground connector on one side only – on the side of the dispenser.



Filter of radio frequency interferences and limiter of high voltage pulse interferences

AC/DC converter with built-in filter of radio frequency interferences and limiter of high voltage pulse interferences

Cable shield must be connected to the ground connector on one side only - on the side of the dispenser

Laying of power and information wires to dispensers should be done in separate pipes with distance of not less than 0.3 meters between each other

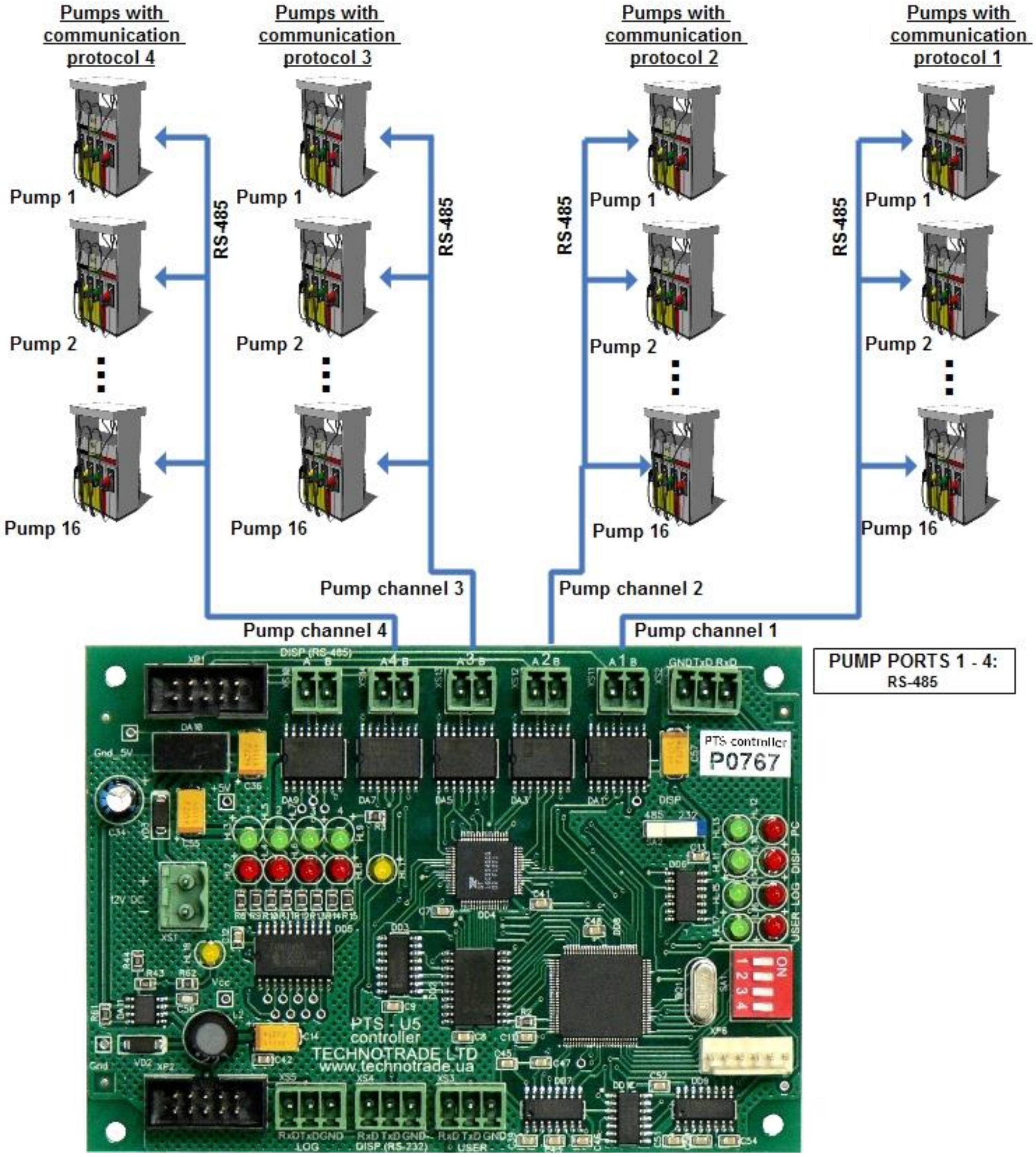
Power supply of all electronic blocks of POS and electronic pumpheads of dispensers should be made from single common uninterruptible power supply source

Supply of electronic pumpheads of dispensers should be made from the UPS unit using 3-wires scheme with isolated neutral through dedicated two-pole breaker for each dispenser. Connection of other parts of dispenser to UPS unit (expect electronic pumpheads) is strictly prohibited

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*").

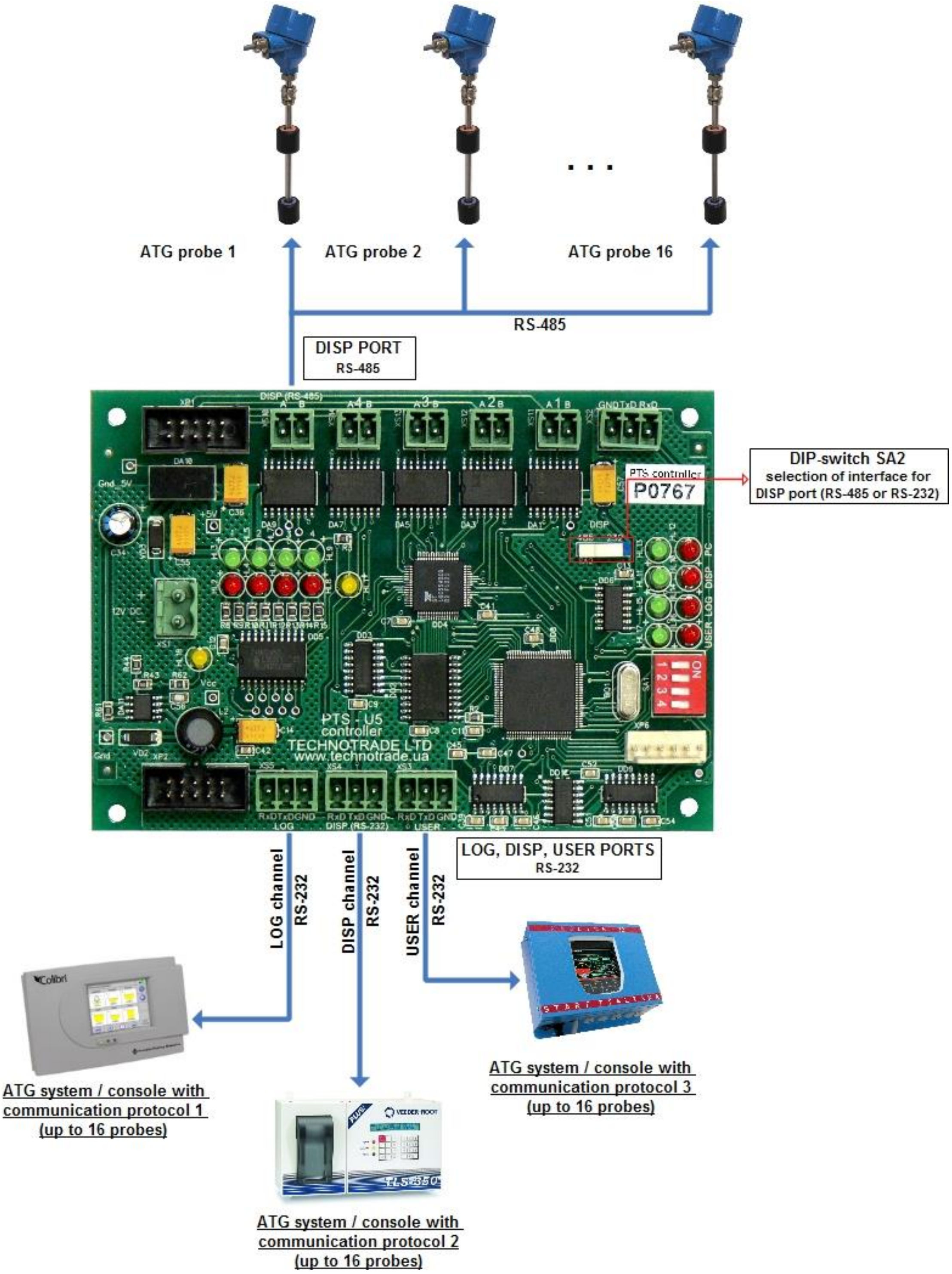
So, you can connect all the maximum 16 pumps to the same pump channel or you can distribute all the 16 pumps between the any or all pump channels. As the pump channels are working independently from each other it is recommended to distribute the connected dispensers between all the pump channels in approximately equal quantities.



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 pump channel.

CONNECTION TO ATG SYSTEMS AND PROBES

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*").

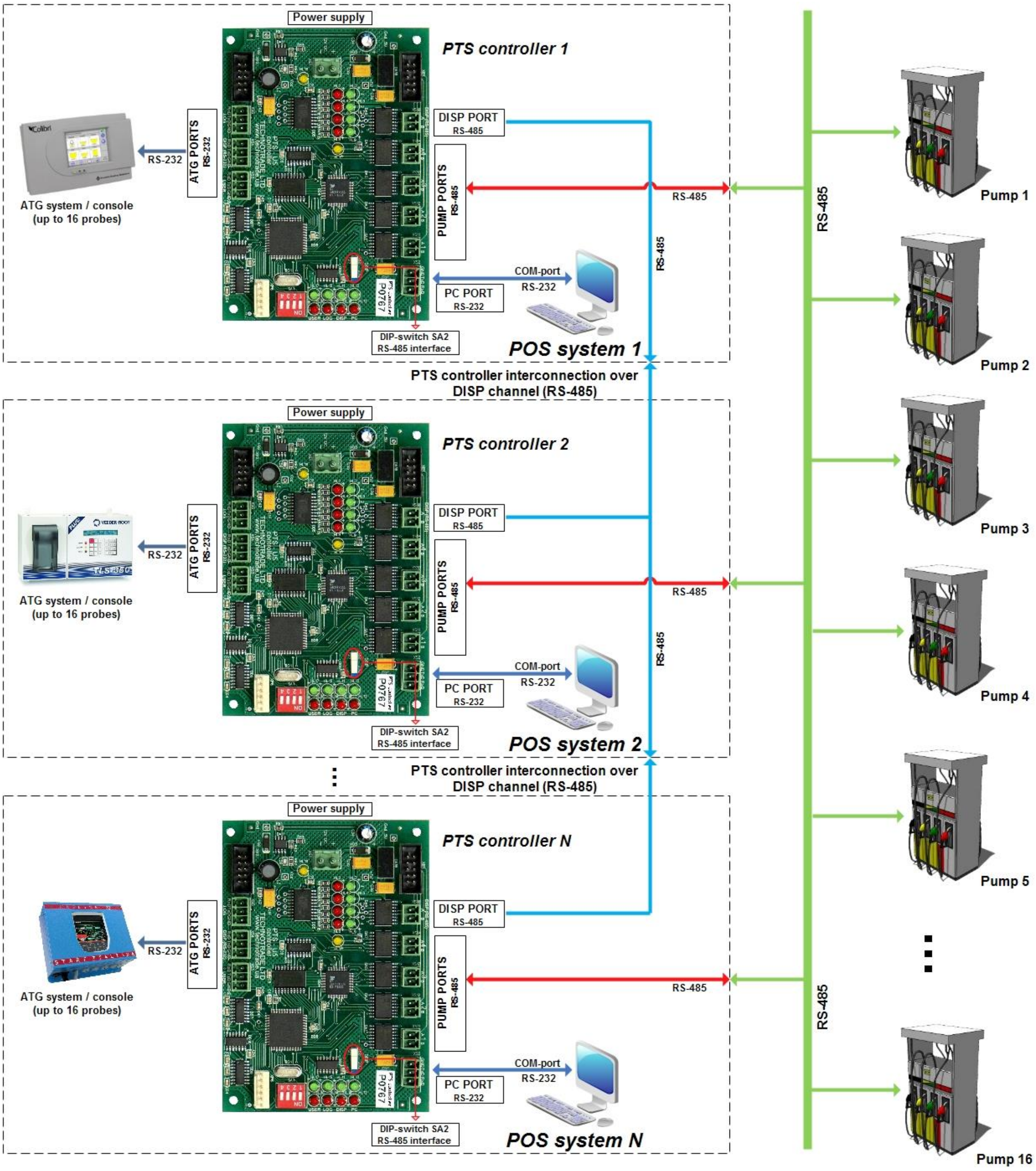


NOTE: DISP channel provides a possibility to connect ATG system (probes) over either RS-485 or RS-232 interfaces – interface is selected using a DIP-switch SA2, located on PTS PCB board.

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 controller interconnection a DIP-switch SA2 should be set in position 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.

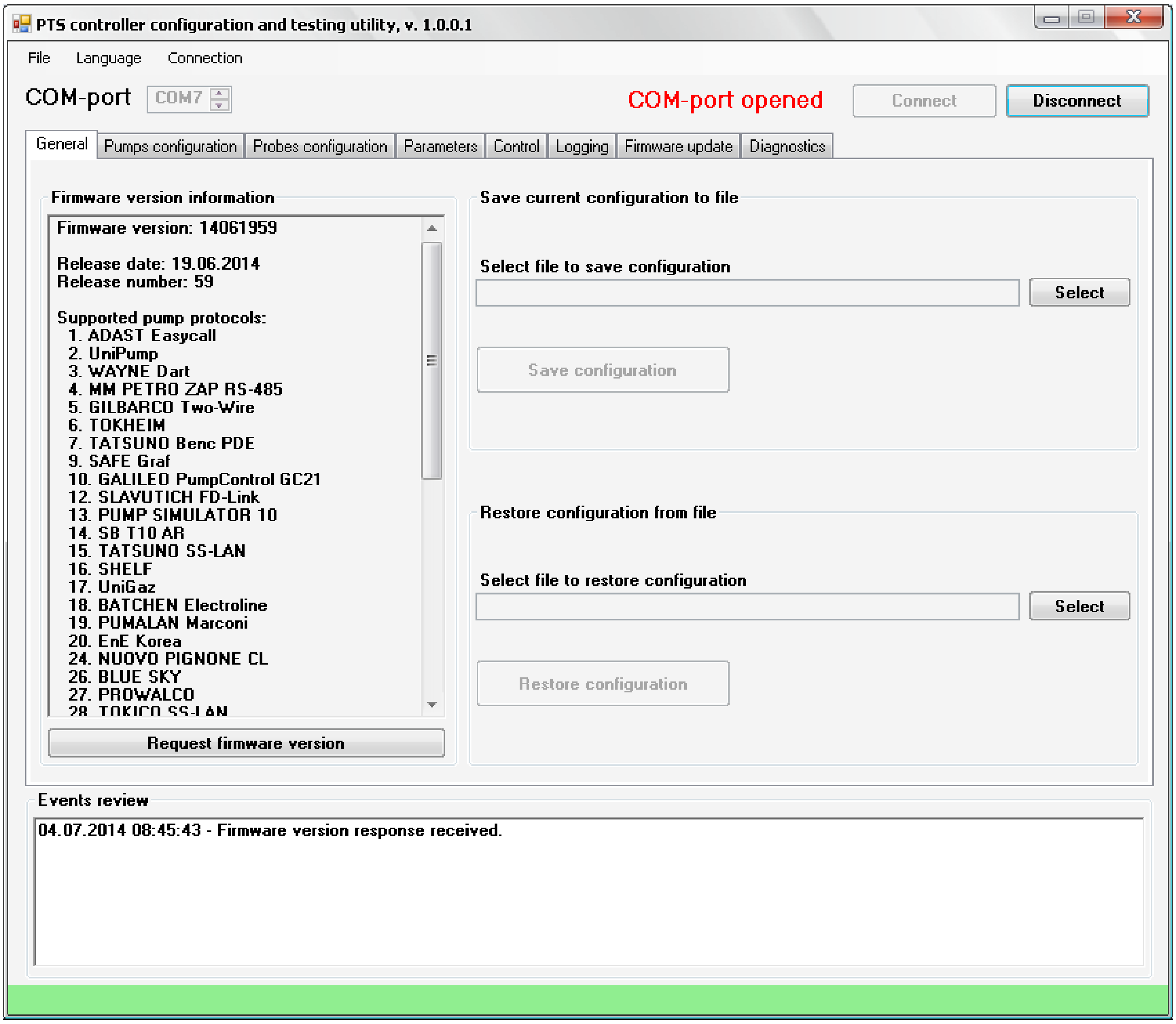
NOTE! In order to provide control over same fuel dispensers from several PTS controllers locking and unlocking of dispensers is required. For this reason in configuration of PTS controller in parameters for PTS controller "*Lock*" and "*Unlock*" commands should be set to be used (see section "*Configuration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

PTS CONTROLLER CONFIGURATION AND TESTING UTILITY

General information

PTS controller configuration and testing utility *Pts_config.exe* serves for configuration and testing of the PTS controller. This utility is open source, which allows developers to use its source code in their development. Currently utility is developed under Visual Studio using C# and Visual Basic .NET languages under Windows OS. Development of the utility is also done in other programming languages under other operating systems. For direct communication with PTS controller the utility uses UniPump communication protocol.



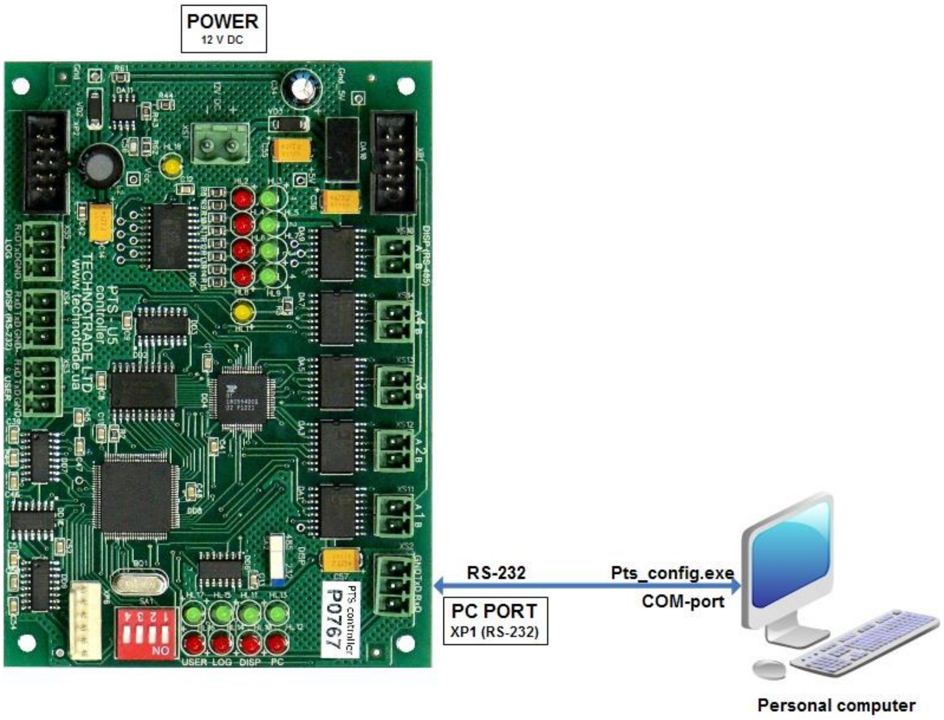
```
Firmware version: 14061959

Release date: 19.06.2014
Release number: 59

Supported pump protocols:
  1. ADAST Easycall
  2. UniPump
  3. WAYNE Dart
  4. MM PETRO ZAP RS-485
  5. GILBARCO Two-Wire
  6. TOKHEIM
  7. TATSUNO Benc PDE
  9. SAFE Graf
  10. GALILEO PumpControl GC21
  12. SLAVUTICH FD-Link
  13. PUMP SIMULATOR 10
  14. SB T10 AR
  15. TATSUNO SS-LAN
  16. SHELF
  17. UniGaz
  18. BATCHEN Electroline
  19. PUMALAN Marconi
  20. EnE Korea
  24. NUOVO PIGNONE CL
  26. BLUE SKY
  27. PROWALCO
  28. TOKICO SS-LAN
```

```
04.07.2014 08:45:43 - Firmware version response received.
```

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

- provision of control over the connected pumps (fuel dispensers) and probes (ATG systems) for testing purposes
- configuration of pumps (fuel dispensers) channels and probes (ATG systems) channels
- configuration of PTS controller parameters and pumps (fuel dispensers) protocols parameters
- update of the PTS controller firmware
- logging of PTS controller communication exchange with connected equipment
- self-diagnostics of PTS controller communication ports and switches
- saving of all configuration of PTS controller to file and loading of all configuration of PTS controller from file

Scheme of connections

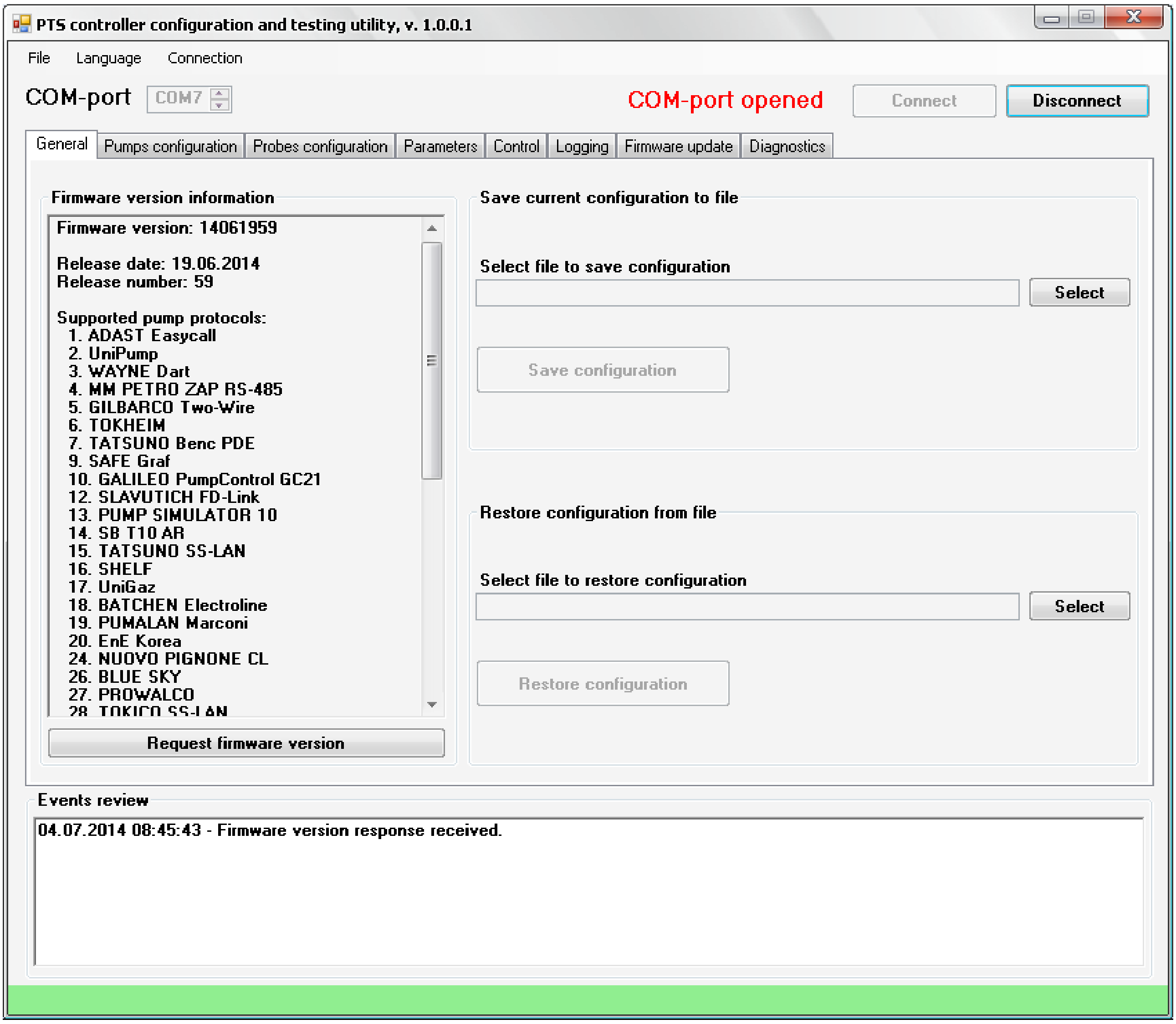


NOTE! On some PCs for correct operation COM-port FIFO settings should be adjusted to Tx: 1, Rx: 1. This may also apply to situations when a PC does not have a native COM-port and external COM-port converter are applied (like USB-to-COM, PCI-to-COM, Ethernet-to-COM, Bluetooth-to-COM etc).

Starting PTS configuration utility

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

Tab "General"



```
Firmware version: 14061959

Release date: 19.06.2014
Release number: 59

Supported pump protocols:
  1. ADAST Easycall
  2. UniPump
  3. WAYNE Dart
  4. MM PETRO ZAP RS-485
  5. GILBARCO Two-Wire
  6. TOKHEIM
  7. TATSUNO Benc PDE
  9. SAFE Graf
  10. GALILEO PumpControl GC21
  12. SLAVUTICH FD-Link
  13. PUMP SIMULATOR 10
  14. SB T10 AR
  15. TATSUNO SS-LAN
  16. SHELF
  17. UniGaz
  18. BATCHEN Electroline
  19. PUMALAN Marconi
  20. EnE Korea
  24. NUOVO PIGNONE CL
  26. BLUE SKY
  27. PROWALCO
  28. TOKICO SS-LAN
```

```
04.07.2014 08:45:43 - Firmware version response received.
```

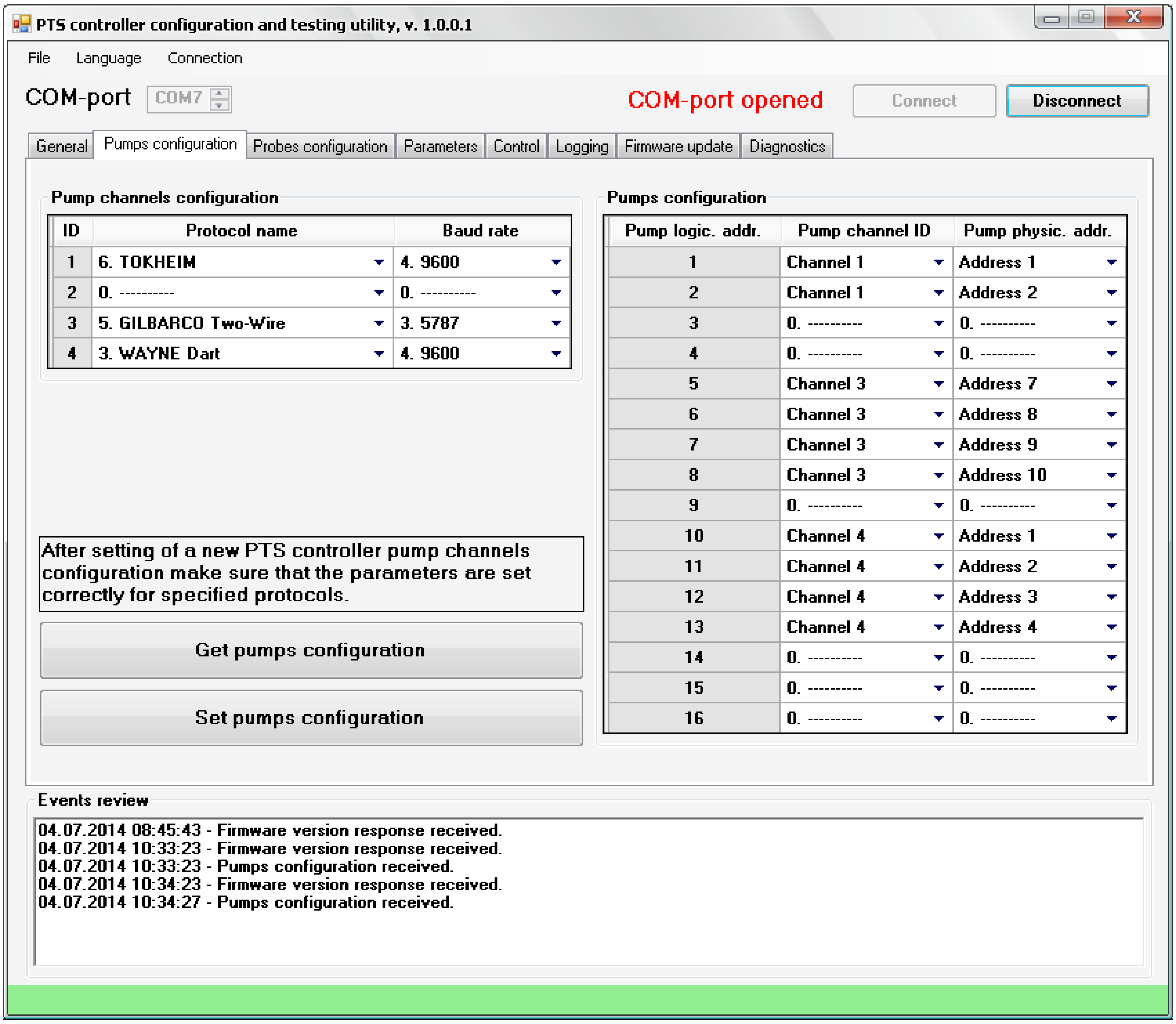
Tab "*General*" serves for 2 main purposes:

- reading of PTS controller firmware version information: firmware date issue and a list of upported communication protocols of pumps and ATG systems (probes)
- saving of all PTS controller configuration to a file on PC
- restoring of all PTS controller configuration from a file on PC

NOTE! Pay attention that the PTS controller is using latest version of the firmware. Latest version of firmware can be received upon request from TECHNOTRADE LTD company or downloaded from TECHNOTRADE LTD company website. Normally new firmware for PTS controller is issued together with PTS configuration utility update having latest features of PTS controller, so updated version of the PTS controller configuration utility *Pts_config.exe* should be also requested and downloaded. Information on how to update PTS controller firmware please find below in section "*Tab 'Firmware update'*".

A list of all supported communication protocols and baud rates by PTS controller as well as description and settings of all parameters are described in file pts_config_en.xml (file may have other name depending on the used language).

Tab "Pumps configuration"



Pump channels configuration

ID	Protocol name	Baud rate
1	6. TOKHEIM	4. 9600
2	0. ----------	0. ----------
3	5. GILBARCO Two-Wire	3. 5787
4	3. WAYNE Dart	4. 9600

After setting of a new PTS controller pump channels configuration make sure that the parameters are set correctly for specified protocols.

Pumps configuration

Pump logic. addr.	Pump channel ID	Pump physic. addr.
1	Channel 1	Address 1
2	Channel 1	Address 2
3	0. ----------	0. ----------
4	0. ----------	0. ----------
5	Channel 3	Address 7
6	Channel 3	Address 8
7	Channel 3	Address 9
8	Channel 3	Address 10
9	0. ----------	0. ----------
10	Channel 4	Address 1
11	Channel 4	Address 2
12	Channel 4	Address 3
13	Channel 4	Address 4
14	0. ----------	0. ----------
15	0. ----------	0. ----------
16	0. ----------	0. ----------

Events review

```
04.07.2014 08:45:43 - Firmware version response received.
04.07.2014 10:33:23 - Firmware version response received.
04.07.2014 10:33:23 - Pumps configuration received.
04.07.2014 10:34:23 - Firmware version response received.
04.07.2014 10:34:27 - Pumps configuration received.
```

Tab "*Pumps configuration*" is used for reading and writing of configuration of pump channels in PTS controller.

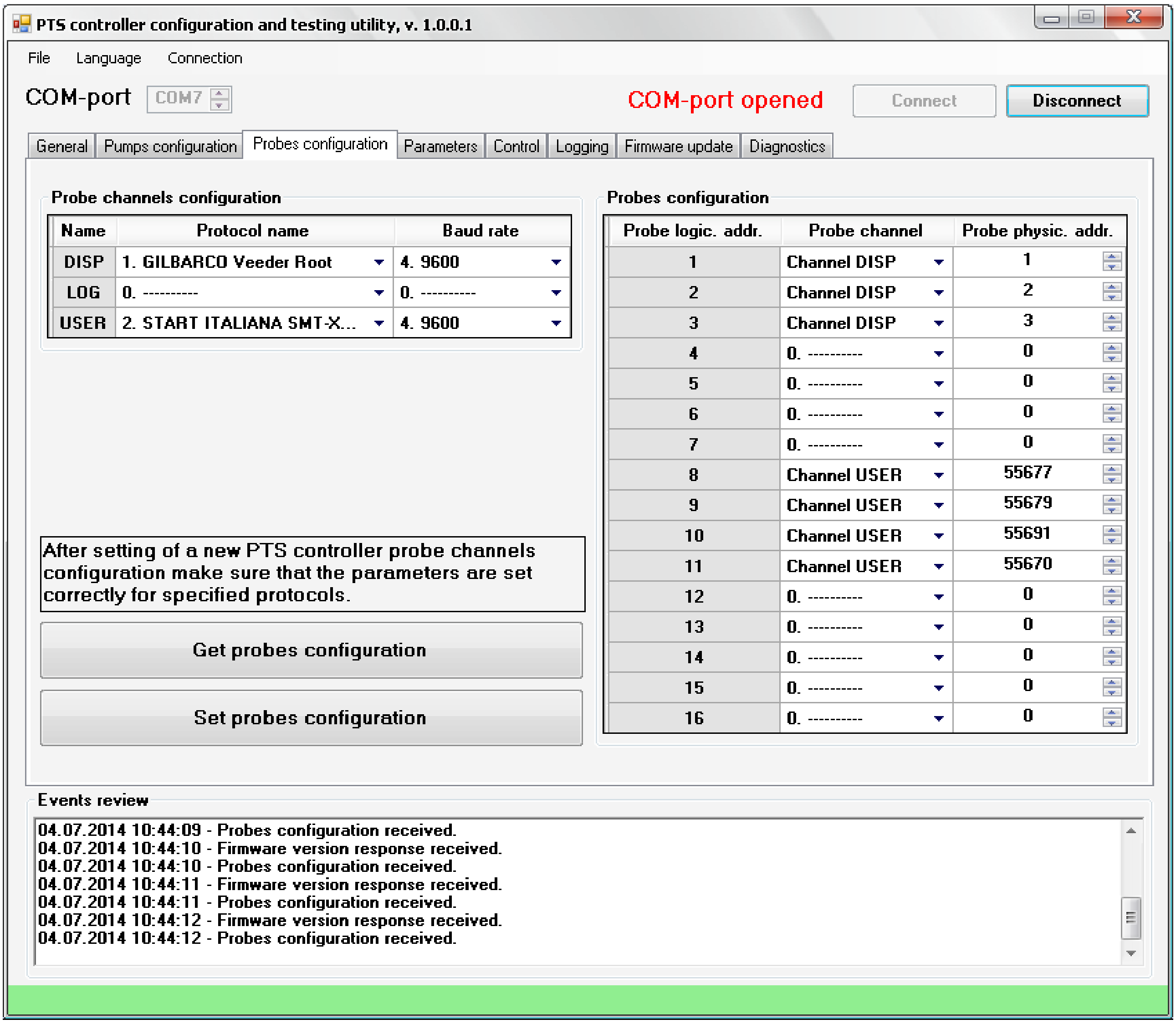
Configuration of pump channels includes setting of communication protocol and baud rate for each of the pump channels and also assigning of pumps to each of the pump channels. Each of the pumps can be assigned to any of the pump channels and requires specification of the pump physical address.

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 "*Set pumps 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 events review field on the form.

For some pump protocols it is also necessary to configure parameters, which is to be done on tab "*Parameters*" of the utility.

Tab "Probes configuration"



Probe channels configuration

Name	Protocol name	Baud rate
DISP	1. GILBARCO Veeder Root	4. 9600
LOG	0. ----------	0. ----------
USER	2. START ITALIANA SMT-X...	4. 9600

After setting of a new PTS controller probe channels configuration make sure that the parameters are set correctly for specified protocols.

Probes configuration

Probe logic. addr.	Probe channel	Probe physic. addr.
1	Channel DISP	1
2	Channel DISP	2
3	Channel DISP	3
4	0. ----------	0
5	0. ----------	0
6	0. ----------	0
7	0. ----------	0
8	Channel USER	55677
9	Channel USER	55679
10	Channel USER	55691
11	Channel USER	55670
12	0. ----------	0
13	0. ----------	0
14	0. ----------	0
15	0. ----------	0
16	0. ----------	0

Events review

```
04.07.2014 10:44:09 - Probes configuration received.
04.07.2014 10:44:10 - Firmware version response received.
04.07.2014 10:44:10 - Probes configuration received.
04.07.2014 10:44:11 - Firmware version response received.
04.07.2014 10:44:11 - Probes configuration received.
04.07.2014 10:44:12 - Firmware version response received.
04.07.2014 10:44:12 - Probes configuration received.
```

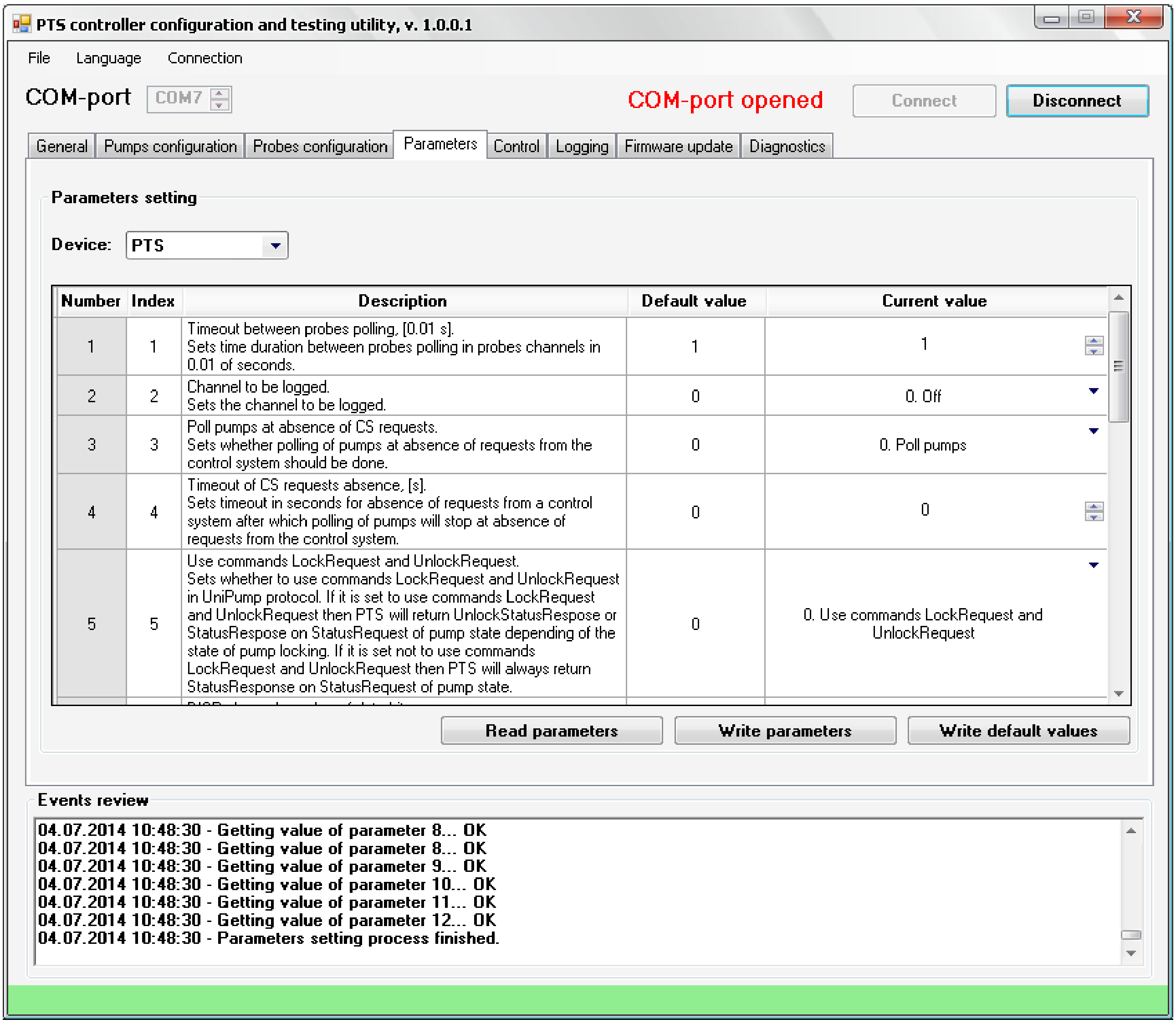
Tab "*Probes configuration*" is used for reading and writing of configuration of probe channels in PTS controller.

Configuration of probe channels includes setting of communication protocol and baud rate for each of the probe channels and also assigning of probes to each of the probe channels. Each of the probes can be assigned to any of the probe channels and requires specification of the probe physical address.

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

After configuration is finished it is necessary to click a button "Set probes configuration", which will write current probe 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 events review field on the form.

Tab "Parameters"



Number	Index	Description	Default value	Current value
1	1	Timeout between probes polling, [0.01 s]. Sets time duration between probes polling in probes channels in 0.01 of seconds.	1	1
2	2	Channel to be logged. Sets the channel to be logged.	0	0. Off
3	3	Poll pumps at absence of CS requests. Sets whether polling of pumps at absence of requests from the control system should be done.	0	0. Poll pumps
4	4	Timeout of CS requests absence, [s]. Sets timeout in seconds for absence of requests from a control system after which polling of pumps will stop at absence of requests from the control system.	0	0
5	5	Use commands LockRequest and UnlockRequest. Sets whether to use commands LockRequest and UnlockRequest in UniPump protocol. If it is set to use commands LockRequest and UnlockRequest then PTS will return UnlockStatusRespose or StatusRespose on StatusRequest of pump state depending of the state of pump locking. If it is set not to use commands LockRequest and UnlockRequest then PTS will always return StatusResponse on StatusRequest of pump state.	0	0. Use commands LockRequest and UnlockRequest

```
04.07.2014 10:48:30 - Getting value of parameter 8... OK
04.07.2014 10:48:30 - Getting value of parameter 8... OK
04.07.2014 10:48:30 - Getting value of parameter 9... OK
04.07.2014 10:48:30 - Getting value of parameter 10... OK
04.07.2014 10:48:30 - Getting value of parameter 11... OK
04.07.2014 10:48:30 - Getting value of parameter 12... OK
04.07.2014 10:48:30 - Parameters setting process finished.
```

Tab "Parameters" serves for:

- configuration of parameters for PTS controller
- configuration of parameters for pumps protocols

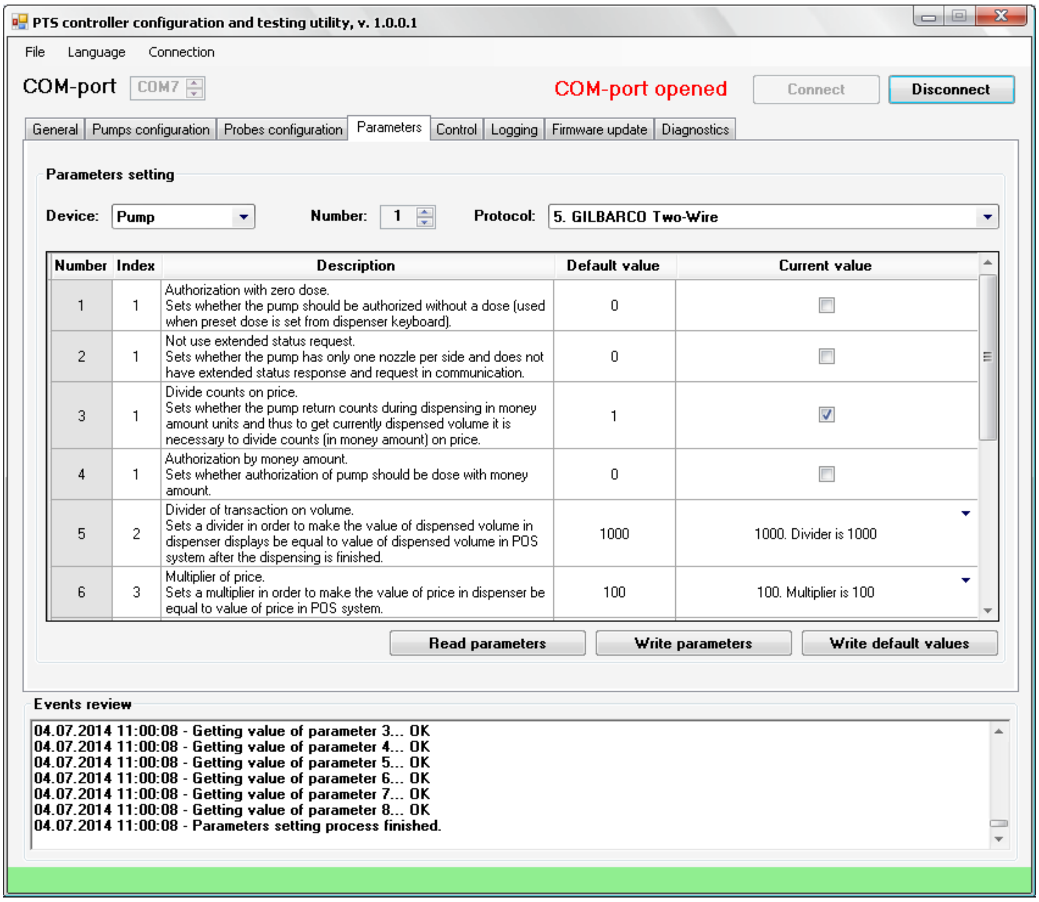
All parameters are listed in a table with detailed description. Default parameters values are specified in a separate column. To get a current value of the parameter it is necessary to read them from PTS controller.

Configuration of PTS controller parameters includes various adjustments for PTS controller operation.

Some of the pumps (or probes) communication protocols also require setting of parameters. These parameters are set for the specified logical address of the pump (or probe) and do not refer to other pumps (or probes) in the PTS controller.

Description and settings of all parameters as well as a list of all supported communication protocols and baud rates by PTS controller are described in file pts_config_en.xml (file may have other name depending on the used language).

Examples of the device protocol parameters:



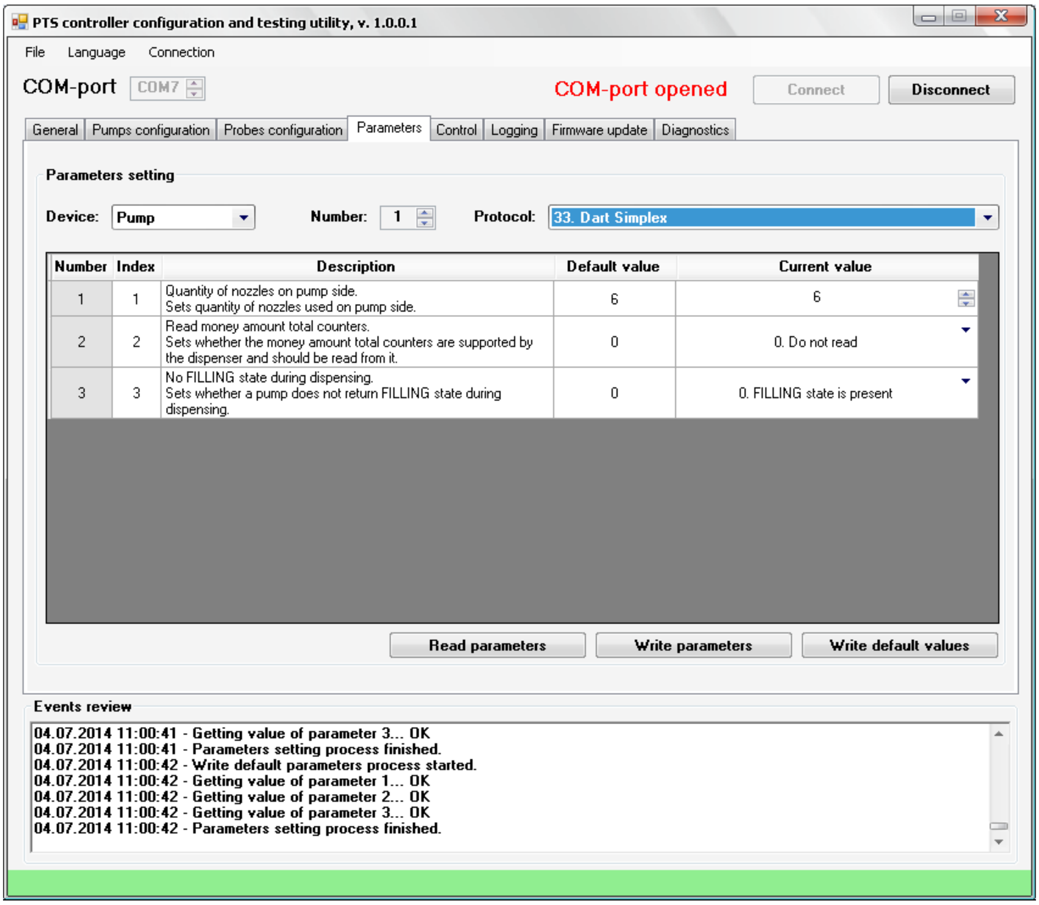
COM-port COM7 COM-port opened

Device: Pump Number: 1 Protocol: 5. GILBARCO Two-Wire

Number	Index	Description	Default value	Current value
1	1	Authorization with zero dose. Sets whether the pump should be authorized without a dose (used when preset dose is set from dispenser keyboard).	0	☐
2	1	Not use extended status request. Sets whether the pump has only one nozzle per side and does not have extended status response and request in communication.	0	☐
3	1	Divide counts on price. Sets whether the pump return counts during dispensing in money amount units and thus to get currently dispensed volume it is necessary to divide counts (in money amount) on price.	1	☑
4	1	Authorization by money amount. Sets whether authorization of pump should be dose with money amount.	0	☐
5	2	Divider of transaction on volume. Sets a divider in order to make the value of dispensed volume in dispenser displays be equal to value of dispensed volume in POS system after the dispensing is finished.	1000	1000. Divider is 1000
6	3	Multiplier of price. Sets a multiplier in order to make the value of price in dispenser be equal to value of price in POS system.	100	100. Multiplier is 100

```
04.07.2014 11:00:08 - Getting value of parameter 3... OK
04.07.2014 11:00:08 - Getting value of parameter 4... OK
04.07.2014 11:00:08 - Getting value of parameter 5... OK
04.07.2014 11:00:08 - Getting value of parameter 6... OK
04.07.2014 11:00:08 - Getting value of parameter 7... OK
04.07.2014 11:00:08 - Getting value of parameter 8... OK
04.07.2014 11:00:08 - Parameters setting process finished.
```

Gilbarco Two-wire communication protocol parameters



COM-port COM7 COM-port opened

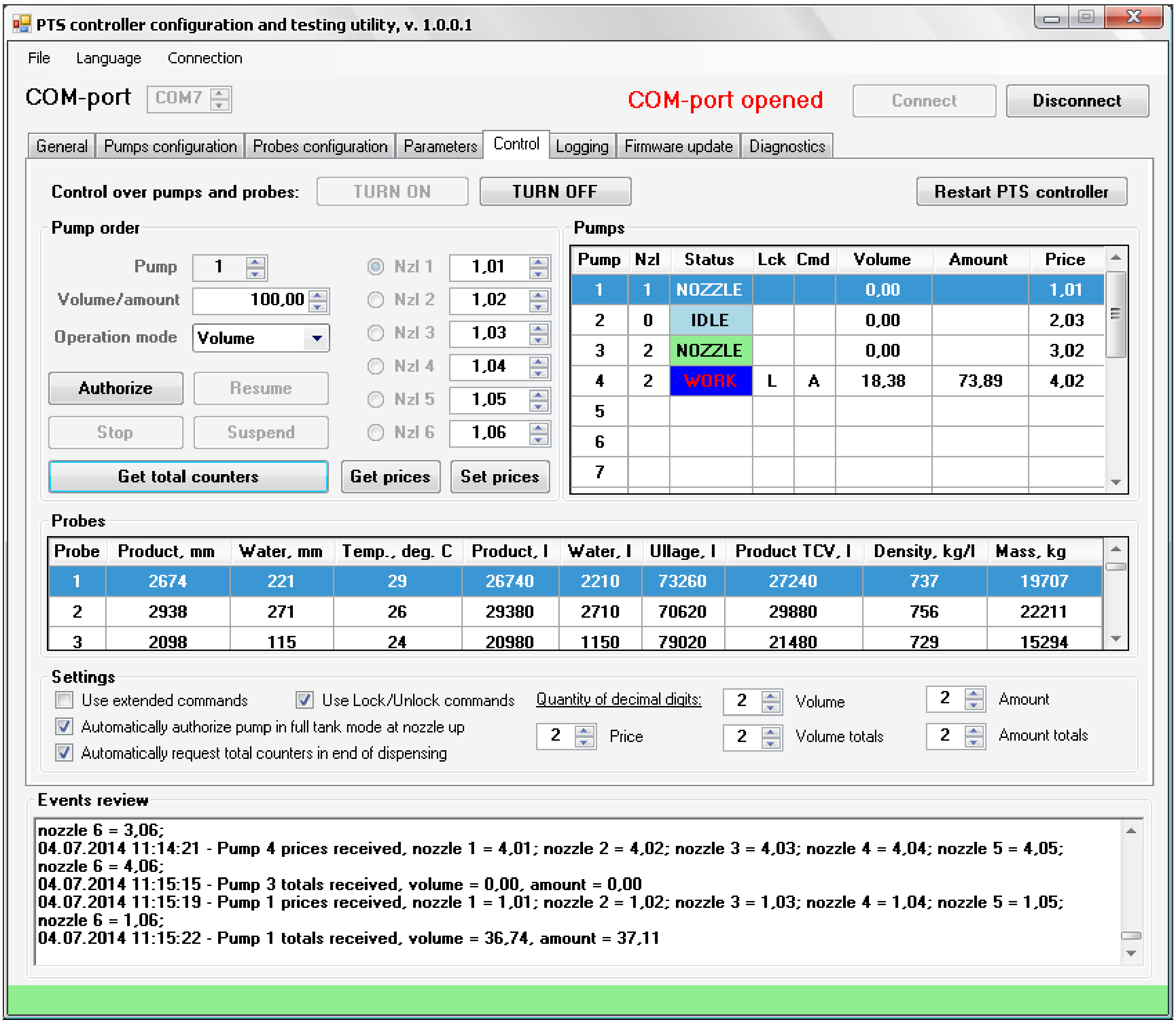
Device: Pump Number: 1 Protocol: 33. Dart Simplex

Number	Index	Description	Default value	Current value
1	1	Quantity of nozzles on pump side. Sets quantity of nozzles used on pump side.	6	6
2	2	Read money amount total counters. Sets whether the money amount total counters are supported by the dispenser and should be read from it.	0	0. Do not read
3	3	No FILLING state during dispensing. Sets whether a pump does not return FILLING state during dispensing.	0	0. FILLING state is present

```
04.07.2014 11:00:41 - Getting value of parameter 3... OK
04.07.2014 11:00:41 - Parameters setting process finished.
04.07.2014 11:00:42 - Write default parameters process started.
04.07.2014 11:00:42 - Getting value of parameter 1... OK
04.07.2014 11:00:42 - Getting value of parameter 2... OK
04.07.2014 11:00:42 - Getting value of parameter 3... OK
04.07.2014 11:00:42 - Parameters setting process finished.
```

Dart communication protocol parameters

Tab "Control"



Pump order

Pump: 1
Volume/amount: 100,00
Operation mode: Volume

● Nzl 1	1,01
○ Nzl 2	1,02
○ Nzl 3	1,03
○ Nzl 4	1,04
○ Nzl 5	1,05
○ Nzl 6	1,06

Pumps

Pump	Nzl	Status	Lck	Cmd	Volume	Amount	Price
1	1	NOZZLE			0,00		1,01
2	0	IDLE			0,00		2,03
3	2	NOZZLE			0,00		3,02
4	2	WORK	L	A	18,38	73,89	4,02
5							
6							
7							

Probes

Probe	Product, mm	Water, mm	Temp., deg. C	Product, l	Water, l	Ullage, l	Product TCV, l	Density, kg/l	Mass, kg
1	2674	221	29	26740	2210	73260	27240	737	19707
2	2938	271	26	29380	2710	70620	29880	756	22211
3	2098	115	24	20980	1150	79020	21480	729	15294

Settings

□ Use extended commands
☑ Use Lock/Unlock commands
☑ Automatically authorize pump in full tank mode at nozzle up
☑ Automatically request total counters in end of dispensing

Quantity of decimal digits:
2 Price
2 Volume
2 Volume totals
2 Amount
2 Amount totals

Events review

```
nozzle 6 = 3,06;
04.07.2014 11:14:21 - Pump 4 prices received, nozzle 1 = 4,01; nozzle 2 = 4,02; nozzle 3 = 4,03; nozzle 4 = 4,04; nozzle 5 = 4,05;
nozzle 6 = 4,06;
04.07.2014 11:15:15 - Pump 3 totals received, volume = 0,00, amount = 0,00
04.07.2014 11:15:19 - Pump 1 prices received, nozzle 1 = 1,01; nozzle 2 = 1,02; nozzle 3 = 1,03; nozzle 4 = 1,04; nozzle 5 = 1,05;
nozzle 6 = 1,06;
04.07.2014 11:15:22 - Pump 1 totals received, volume = 36,74, amount = 37,11
```

Tab "*Control*" serves for provision of control over connected pumps and probes.

Section "*Pumps*" is used for displaying state of all 16 pumps of PTS controller, current taken up nozzle, state whether the pump is locked by PTS controller, currently executed command and also volume, money amount and price of dispensed fuel. Selection of the pump is made by selecting a pumps row in a table.

Section "*Pump order*" is made for provision of control over the pumps:

- selection of pump
- field for setting a dose to be dispensed by selected pump
- selection of operation mode: volume preset, money amount preset, full tank
- commands to be given to selected pump: authorize, stop, suspend, resume, get total counters, get and set price
- fields for getting/setting prices of each nozzle of the selected pump

Section "*Probes*" is used for displaying measurement data of each of 16 probes of PTS controller.

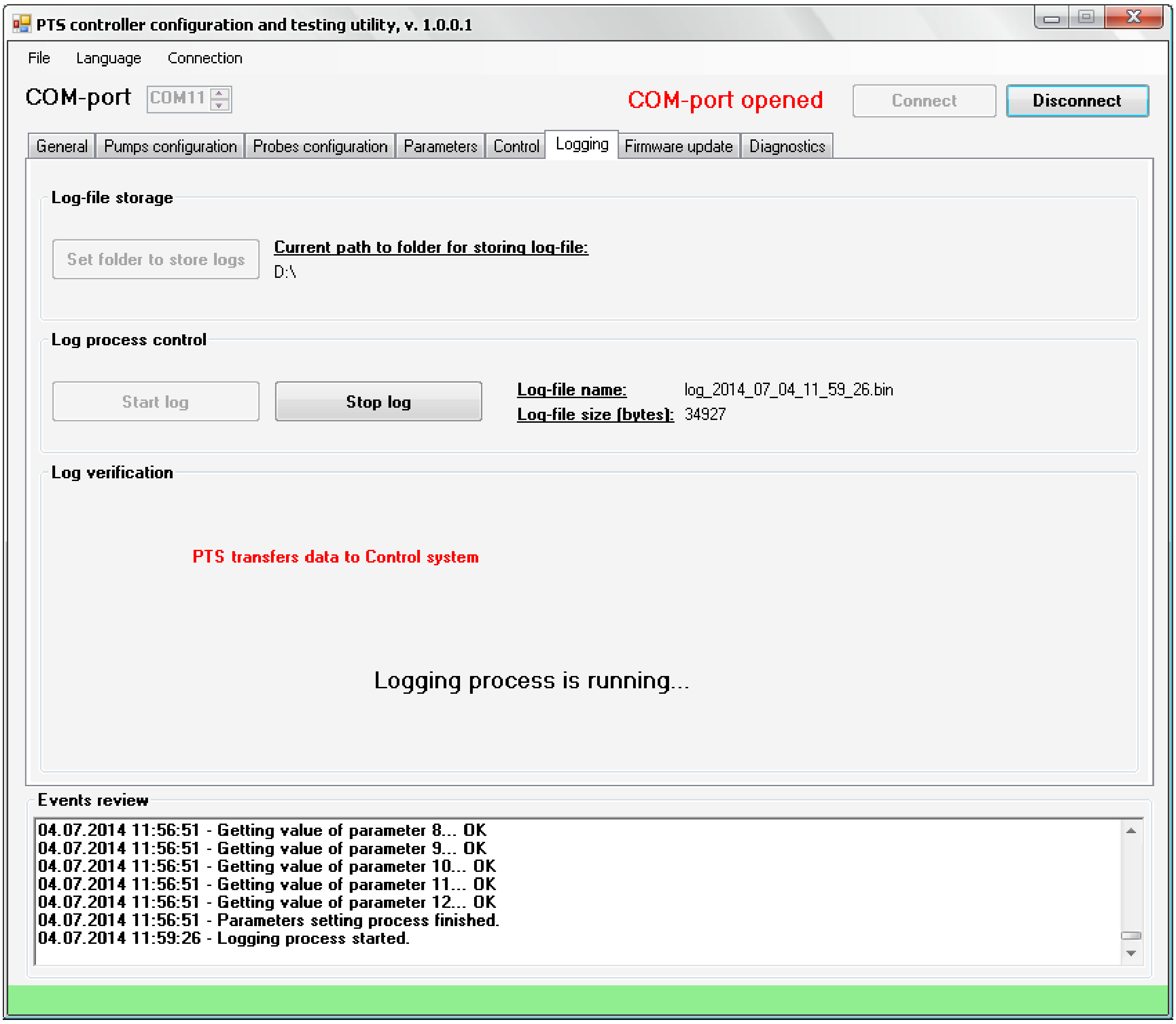
Section "*Settings*" includes the following settings:

- "*Use extended commands*" sets if communication with PTS controller should be doe using general commands of extended (extended commands are to be used instead of general commands when there is necessity to get/set values to PTS controller, which size is biger than provided by general UniPump protocol format)

- "*Use Lock/Unlock commands*" – if this option is selected then PTS controller will try to lock the pump before giving command to it and unlock it after the command is performed, it this option is not set then PTS controller will not send LockRequest and UnlockRequest commands. This option should be equal to PTS parameter "Use command LockRequest and UnlockRequest".
- "*Automatically authorize pump in full tank mode at nozzle up*" – if this option is set then PTS controller will automatically authorize the pump with volume on 999999 liters automatically at once when the nozzle is up. This is made to set dispensing in completely automatic mode without any actions to be performed on computer to start dispensing.
- "*Automatically request total counters in end of dispensing*" – I this optionis set then PTS controlle will automatically request total counters from the pump at the end of dispensing (after a transaction is closed).
- "*Quantity of decimal digits*": these options set number of decimal places used in pump in fields of volume, money amount, price, volume and money amount total counters. The quantity of decimal digits in these fields should correspond to their quantity in the pump in order to make numbers displays in pump same as in this PTS configuration utility.

Events review fields displays results of all performed operations.

Tab "*Logging*"



Current path to folder for storing log-file:
D:\

Log-file name: log_2014_07_04_11_59_26.bin
Log-file size (bytes): 34927

PTS transfers data to Control system

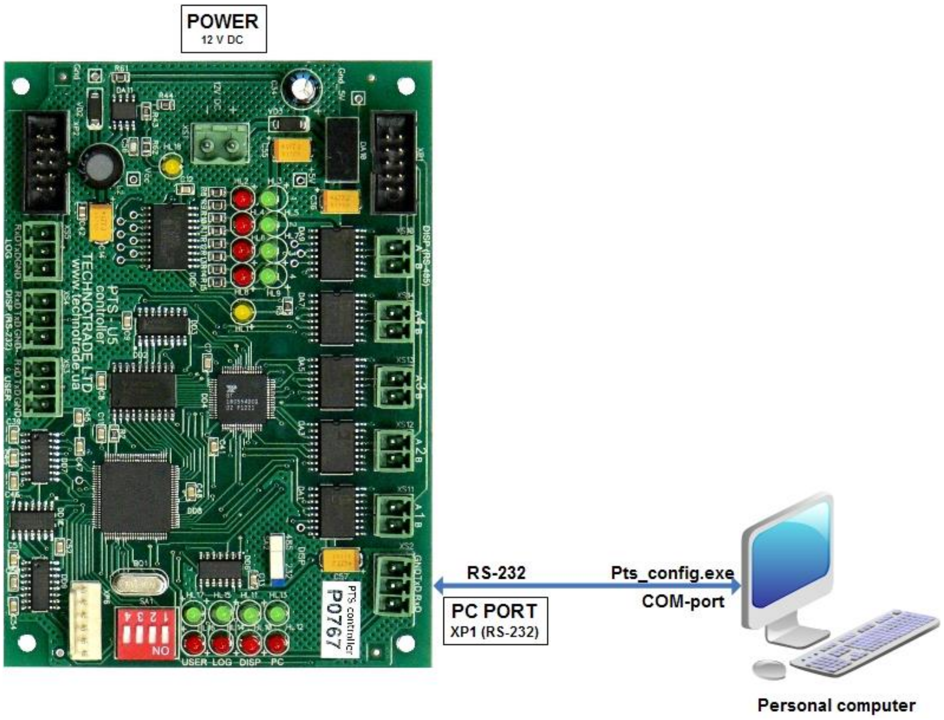
Logging process is running...

```
04.07.2014 11:56:51 - Getting value of parameter 8... OK
04.07.2014 11:56:51 - Getting value of parameter 9... OK
04.07.2014 11:56:51 - Getting value of parameter 10... OK
04.07.2014 11:56:51 - Getting value of parameter 11... OK
04.07.2014 11:56:51 - Getting value of parameter 12... OK
04.07.2014 11:56:51 - Parameters setting process finished.
04.07.2014 11:59:26 - Logging process started.
```

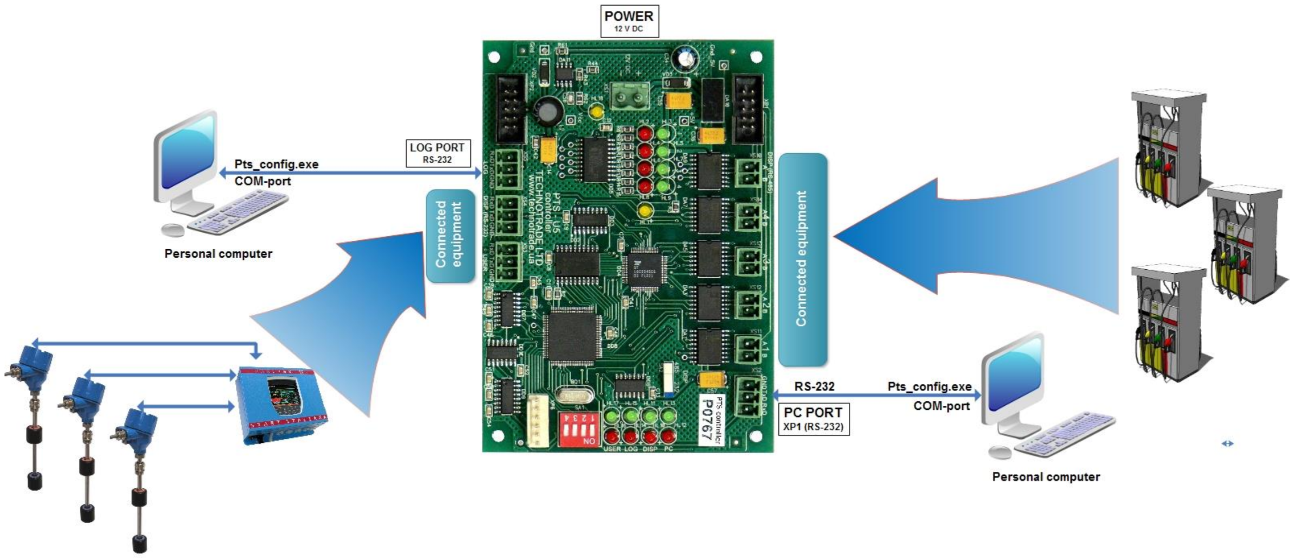
Tab "*Logging*" serves for logging of communication exchange between the PTS controller and connected pump or probe.

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_config.exe* utility to save in a control system exchange logs of PTS controller with connected equipment.

In order to get the log it is necessary first to adjust the PTS controller parameter (see tab "*Parameters*") "*Channel to be logged*", where to set a channel, from which a log should be taken. This should be done at direct connection of PTS controller to PC.



After that it is necessary to connect the PC to PTS controller port LOG as shown on the scheme below. At this PTS controller should be working with connected equipment and control system. Log is taken online – all current actions on the selected channel are written to the log-file.



On the tab “*Logging*” it is necessary to select a folder, where the log-file is to be stored and press “*Start log*” button. Log process will be displayed in the “*Log verification*” section and size of the log-file capture will be displayed on the form.

“*Log verification*” section displays log process by indicating the direction of currently taken log:

- Control system transfers data to PTS
- PTS transfers data to Control system
- PTS transfers data to Slave device
- Slave device transfers data to PTS

In case if there is no data transferred – the "*Log verification*" section displays a message "*Log is not taken, please recheck log parameters in PTS controller and connections!*". If you see this message – you need to recheck all the configuration and connections to get a log.

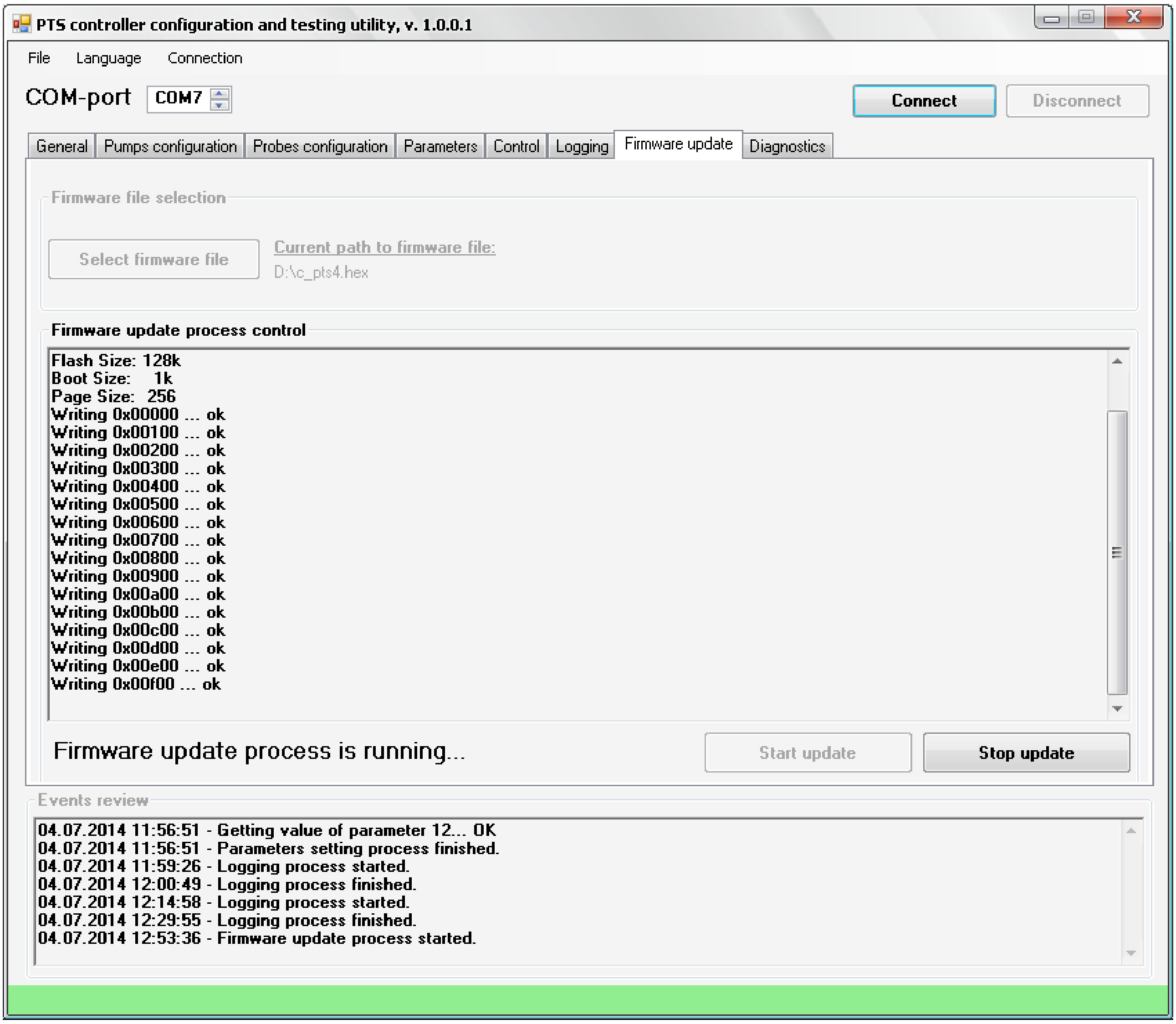
Log is accumulated to a file, size of file should be growing during the logging process, if it does not grow – then the log is not taken, you will probably see a message "*Log is not taken, please recheck log parameters in PTS controller and connections!*" in "*Log verification*" section as described above.

Log-file has a name with indication of the time of log taken, for example: log_2014_07_04_12_14_58.bin, which means that the log was taken on 4th of July, 2014 at 12:14:58.

In case if log process is switched on –a new log-file with a new name is created each 30 minutes (old log-file is saved). This allows to avoid log-file overfilling when its size comes to critical to the operating system. Thus you can switch on logging and wait until any problem happens. When it happens and you approximately know time of this problem – you can take appropriate (by name) log-file and send to TECHNOTRADE LTD for analysis.

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

Tab "Firmware update"



Current path to firmware file:
D:\c_pts4.hex

```
Flash Size: 128k
Boot Size:    1k
Page Size:  256
Writing 0x00000 ... ok
Writing 0x00100 ... ok
Writing 0x00200 ... ok
Writing 0x00300 ... ok
Writing 0x00400 ... ok
Writing 0x00500 ... ok
Writing 0x00600 ... ok
Writing 0x00700 ... ok
Writing 0x00800 ... ok
Writing 0x00900 ... ok
Writing 0x00a00 ... ok
Writing 0x00b00 ... ok
Writing 0x00c00 ... ok
Writing 0x00d00 ... ok
Writing 0x00e00 ... ok
Writing 0x00f00 ... ok
```

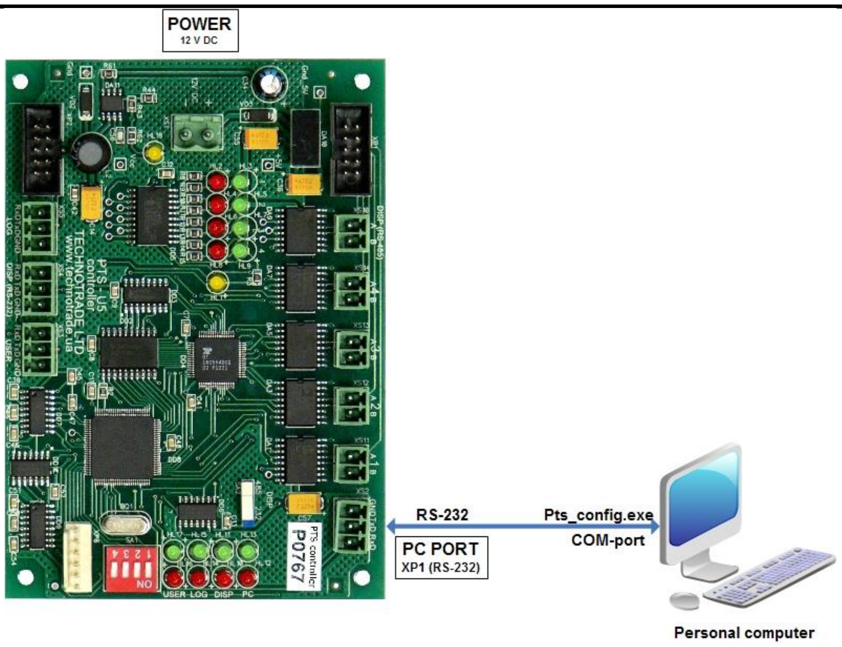
Firmware update process is running...

```
04.07.2014 11:56:51 - Getting value of parameter 12... OK
04.07.2014 11:56:51 - Parameters setting process finished.
04.07.2014 11:59:26 - Logging process started.
04.07.2014 12:00:49 - Logging process finished.
04.07.2014 12:14:58 - Logging process started.
04.07.2014 12:29:55 - Logging process finished.
04.07.2014 12:53:36 - Firmware update process started.
```

Tab "*Firmware update*" servers for updating of firmware of PTS controller and other interface converters. 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.

Connection of the PTS controller is done directly to COM-port of PC.

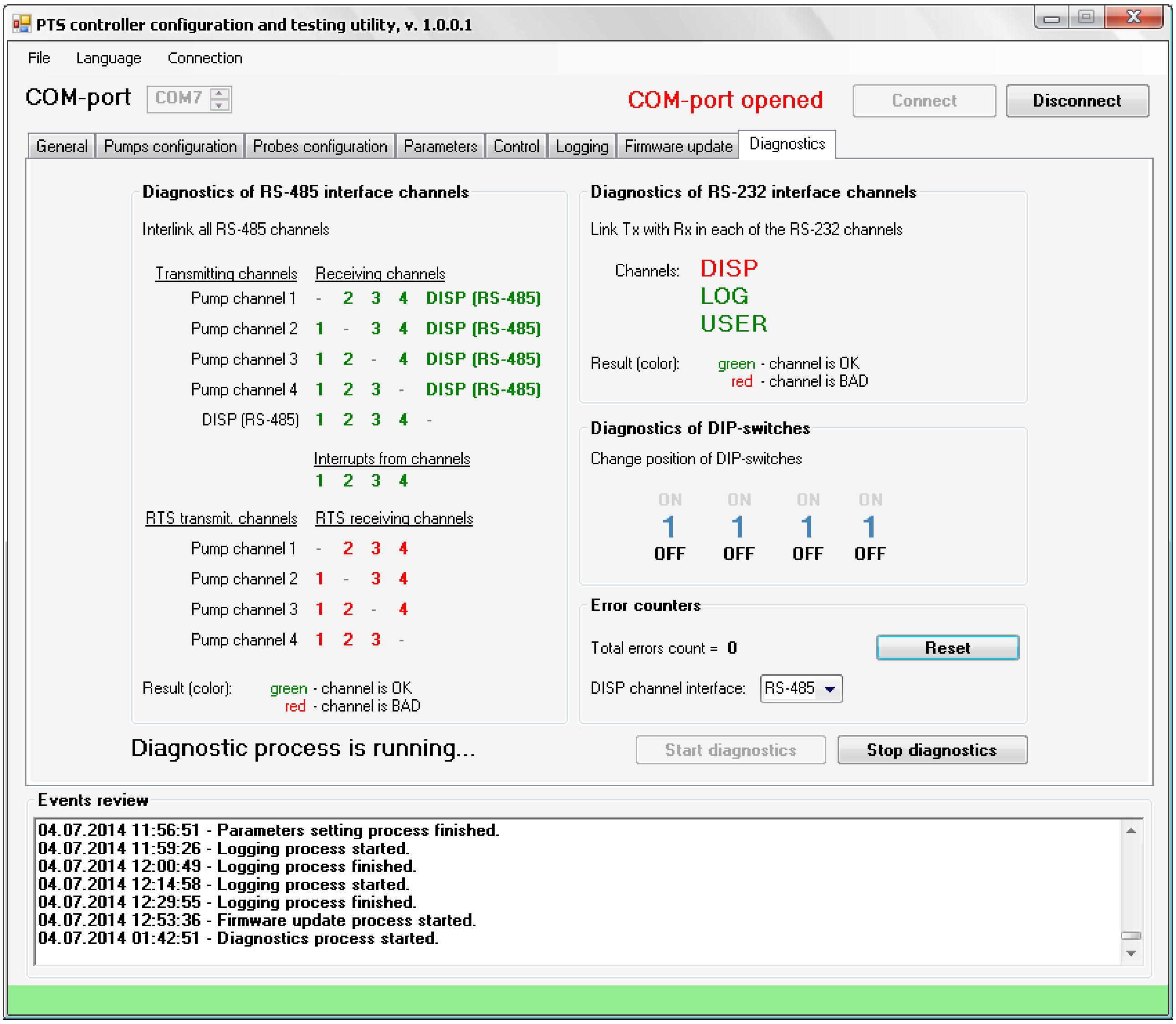
It is necessary select the COM-port number of the PC, check that connection is not opened and select the firmware file before starting the firmware update process.



NOTE! Pay attention that your PTS controller is using latest version of the firmware. Latest version of firmware can be received upon request from TECHNOTRADE LTD company or downloaded from TECHNOTRADE LTD company website. Normally new firmware for PTS controller is issued together with PTS configuration utility update having latest features of PTS controller, so updated version of the PTS controller configuration utility *Pts_config.exe* should be also requested and downloaded. Information on the present firmware version of PTS controller can be checked on tab “*General*” (mentioned above).

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 on tab “*General*” section *“Save current configuration 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 on tab “*General*” section “*Restore configuration from file*”).

Tab "Diagnostics "



COM-port COM7

COM-port opened

Diagnostics of RS-485 interface channels

Interlink all RS-485 channels

Transmitting channels	Receiving channels				
Pump channel 1	-	2	3	4	DISP (RS-485)
Pump channel 2	1	-	3	4	DISP (RS-485)
Pump channel 3	1	2	-	4	DISP (RS-485)
Pump channel 4	1	2	3	-	DISP (RS-485)
DISP (RS-485)	1	2	3	4	-

Interrupts from channels
1 2 3 4

RTS transmit. channels	RTS receiving channels			
Pump channel 1	-	2	3	4
Pump channel 2	1	-	3	4
Pump channel 3	1	2	-	4
Pump channel 4	1	2	3	-

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

Diagnostics of RS-232 interface channels

Link Tx with Rx in each of the RS-232 channels

Channels: DISP
LOG
USER

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

Diagnostics of DIP-switches

Change position of DIP-switches

ON	ON	ON	ON
1	1	1	1
OFF	OFF	OFF	OFF

Error counters

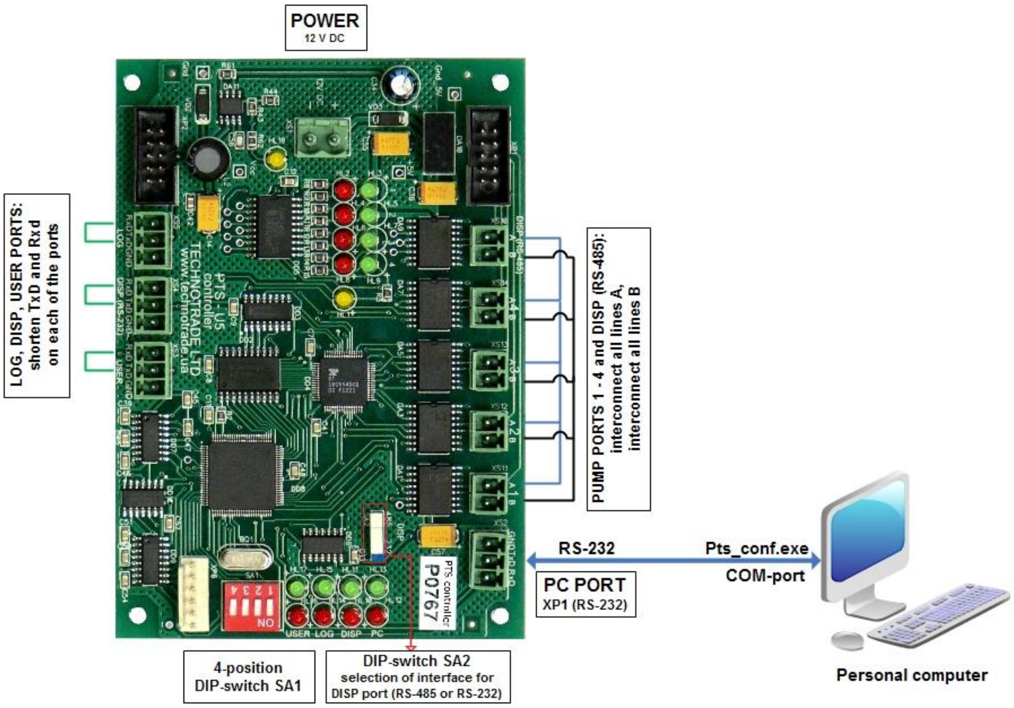
Total errors count = 0

DISP channel interface: RS-485

Diagnostic process is running...

```
04.07.2014 11:56:51 - Parameters setting process finished.
04.07.2014 11:59:26 - Logging process started.
04.07.2014 12:00:49 - Logging process finished.
04.07.2014 12:14:58 - Logging process started.
04.07.2014 12:29:55 - Logging process finished.
04.07.2014 12:53:36 - Firmware update process started.
04.07.2014 01:42:51 - Diagnostics process started.
```

Tab "*Diagnostics*" servers for self-diagnostics of PTS controller ports and DIP-switches. Scheme of connections of the PTS controller should be the following:



Pump ports 1 – 4 and DISP port (RS-485) should be interconnected: lines A of each port shortened together and separately lines B of each port shortened together.

On ATG channels LOG, USER and DISP (RS-232) pins TxD should be shortened with pins RxD.

Diagnostics of port with RS-485 interfaces:

Place DIP-switch SA2 on PTS board in position "*485*" for selection of RS-485 interface for DISP channel.

In section "Error counter" set DISP channel interface to RS-485.

In section "*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). In case if the error counter increases its value – then the errors are taking place.

For PTS controller of version PTS-U5 area RTS channel 1 – 4 should not be taken into account cause is used with previous versions of PTS controller (PTS-U, PTS-U2, PTS-U3).

Diagnostics of port with RS-232 interfaces:

Place DIP-switch SA2 on PTS board in position "*232*" for selection of RS-232 interface for DISP channel.

In section "*Error counters*" set DISP channel interface to RS-232.

In section "*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). In case if the error counter increases its value – then the errors are taking place.

Diagnostics of DIP-switch SA1:

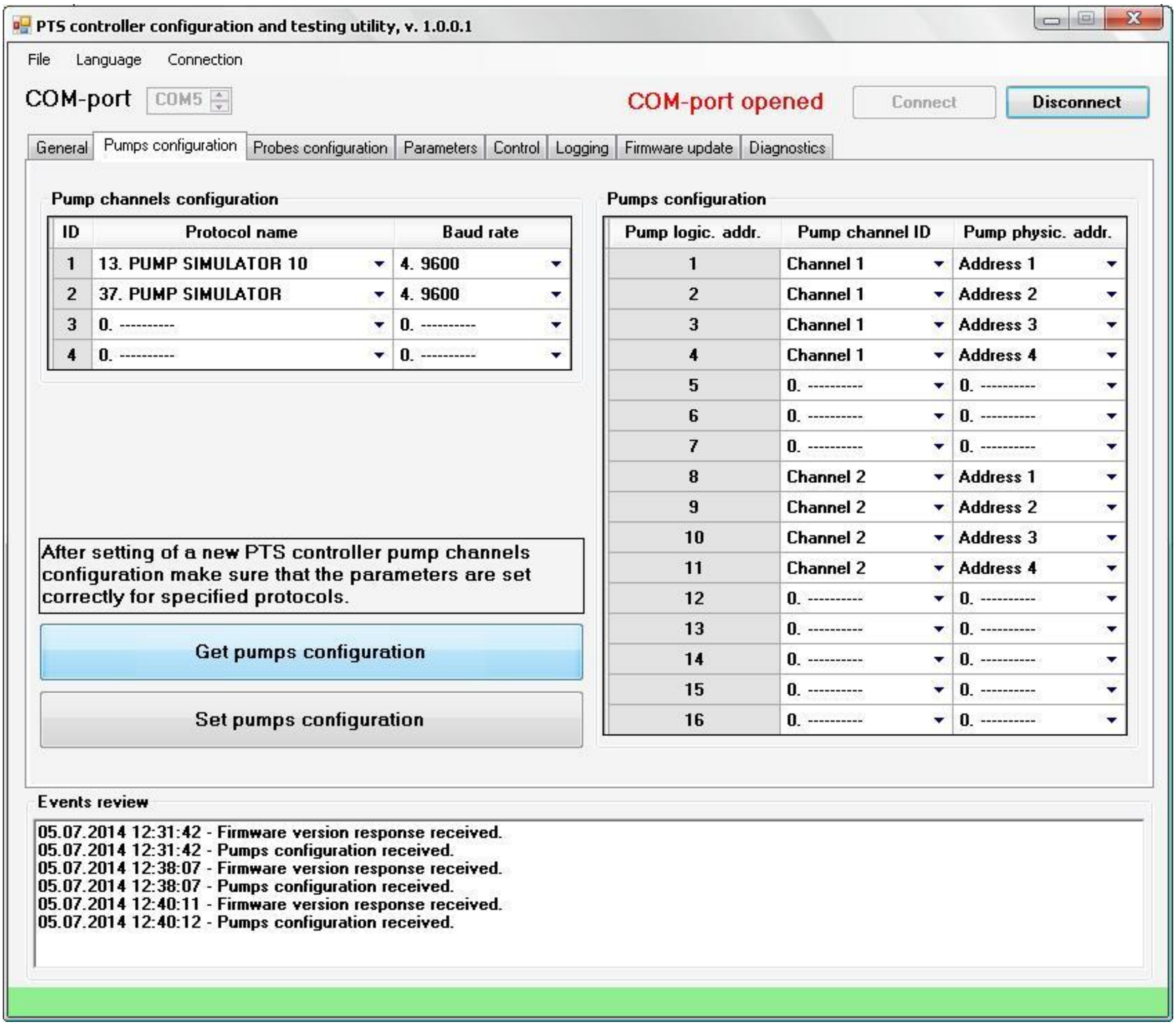
In section "*Diagnostics of DIP-switches*" current position of switches in SA1 DIP-switch on PTS board is displayed. If displayed position of switches correspond to real position of switches in DIP-switch SA1 on PTS board after checking them in various positions – then DIP-switch SA1 is working correctly.

BUILT-IN PUMPS 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.



Pump channels configuration

ID	Protocol name	Baud rate
1	13. PUMP SIMULATOR 10	4. 9600
2	37. PUMP SIMULATOR	4. 9600
3	0. ----------	0. ----------
4	0. ----------	0. ----------

Pumps configuration

Pump logic. addr.	Pump channel ID	Pump physic. addr.
1	Channel 1	Address 1
2	Channel 1	Address 2
3	Channel 1	Address 3
4	Channel 1	Address 4
5	0. ----------	0. ----------
6	0. ----------	0. ----------
7	0. ----------	0. ----------
8	Channel 2	Address 1
9	Channel 2	Address 2
10	Channel 2	Address 3
11	Channel 2	Address 4
12	0. ----------	0. ----------
13	0. ----------	0. ----------
14	0. ----------	0. ----------
15	0. ----------	0. ----------
16	0. ----------	0. ----------

After setting of a new PTS controller pump channels configuration make sure that the parameters are set correctly for specified protocols.

Events review

```
05.07.2014 12:31:42 - Firmware version response received.
05.07.2014 12:31:42 - Pumps configuration received.
05.07.2014 12:38:07 - Firmware version response received.
05.07.2014 12:38:07 - Pumps configuration received.
05.07.2014 12:40:11 - Firmware version response received.
05.07.2014 12:40:12 - Pumps configuration received.
```

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 SA1 on the PTS controller board (DIP-switch SA1 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 nozzle 1 up.

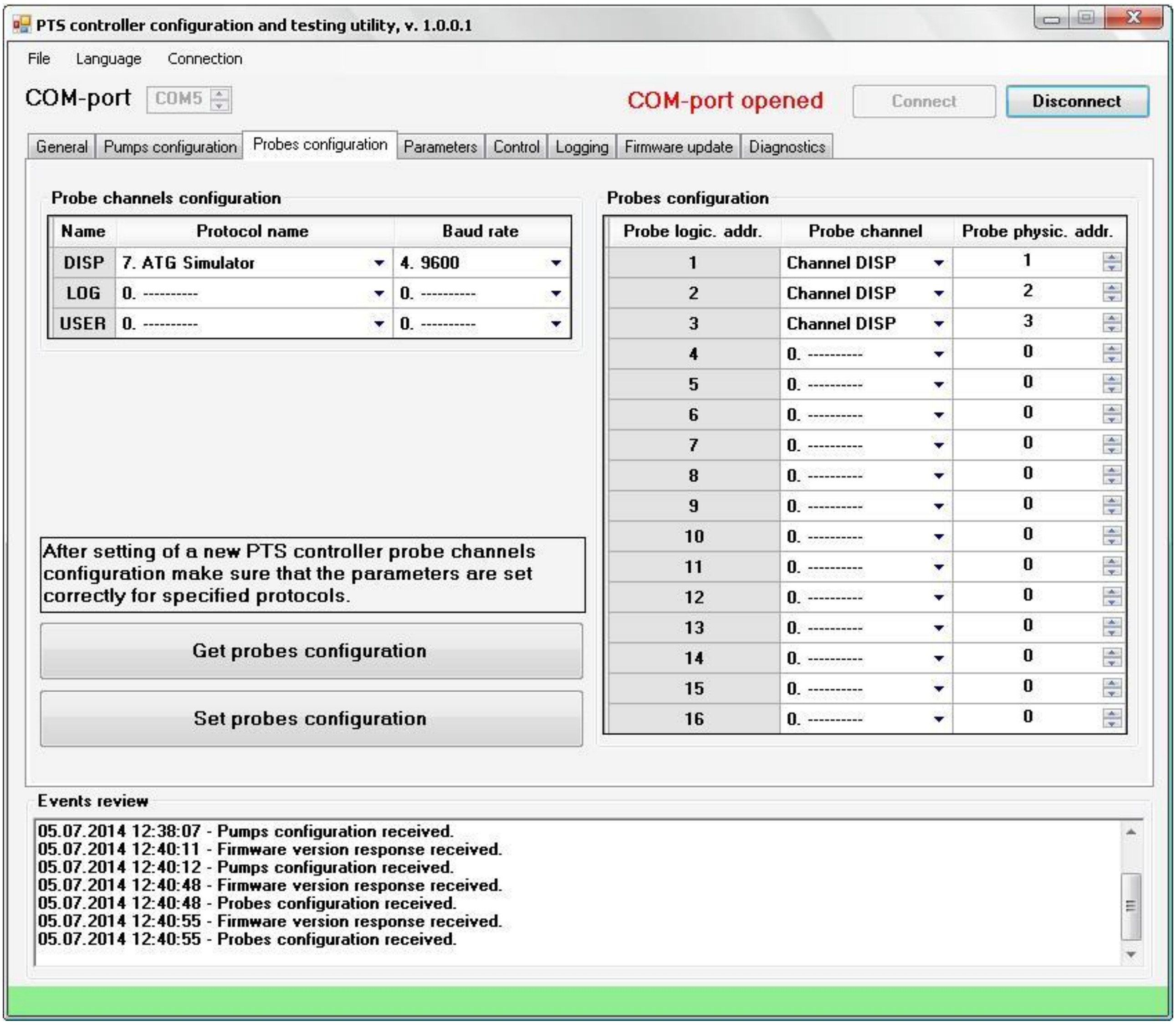
Pump simulators allow to:

- 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 PROBES 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.



Probe channels configuration

Name	Protocol name	Baud rate
DISP	7. ATG Simulator	4. 9600
LOG	0. ----------	0. ----------
USER	0. ----------	0. ----------

After setting of a new PTS controller probe channels configuration make sure that the parameters are set correctly for specified protocols.

Probes configuration

Probe logic. addr.	Probe channel	Probe physic. addr.
1	Channel DISP	1
2	Channel DISP	2
3	Channel DISP	3
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

Events review

```
05.07.2014 12:38:07 - Pumps configuration received.
05.07.2014 12:40:11 - Firmware version response received.
05.07.2014 12:40:12 - Pumps configuration received.
05.07.2014 12:40:48 - Firmware version response received.
05.07.2014 12:40:48 - Probes configuration received.
05.07.2014 12:40:55 - Firmware version response received.
05.07.2014 12:40:55 - Probes configuration received.
```

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)

PTS CONTROLLER API (APPLICATION PROGRAMMING INTERFACES)

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
  - open source PTS controller configuration and testing utility
- **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
  - open source PTS controller configuration and testing utility
- **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.

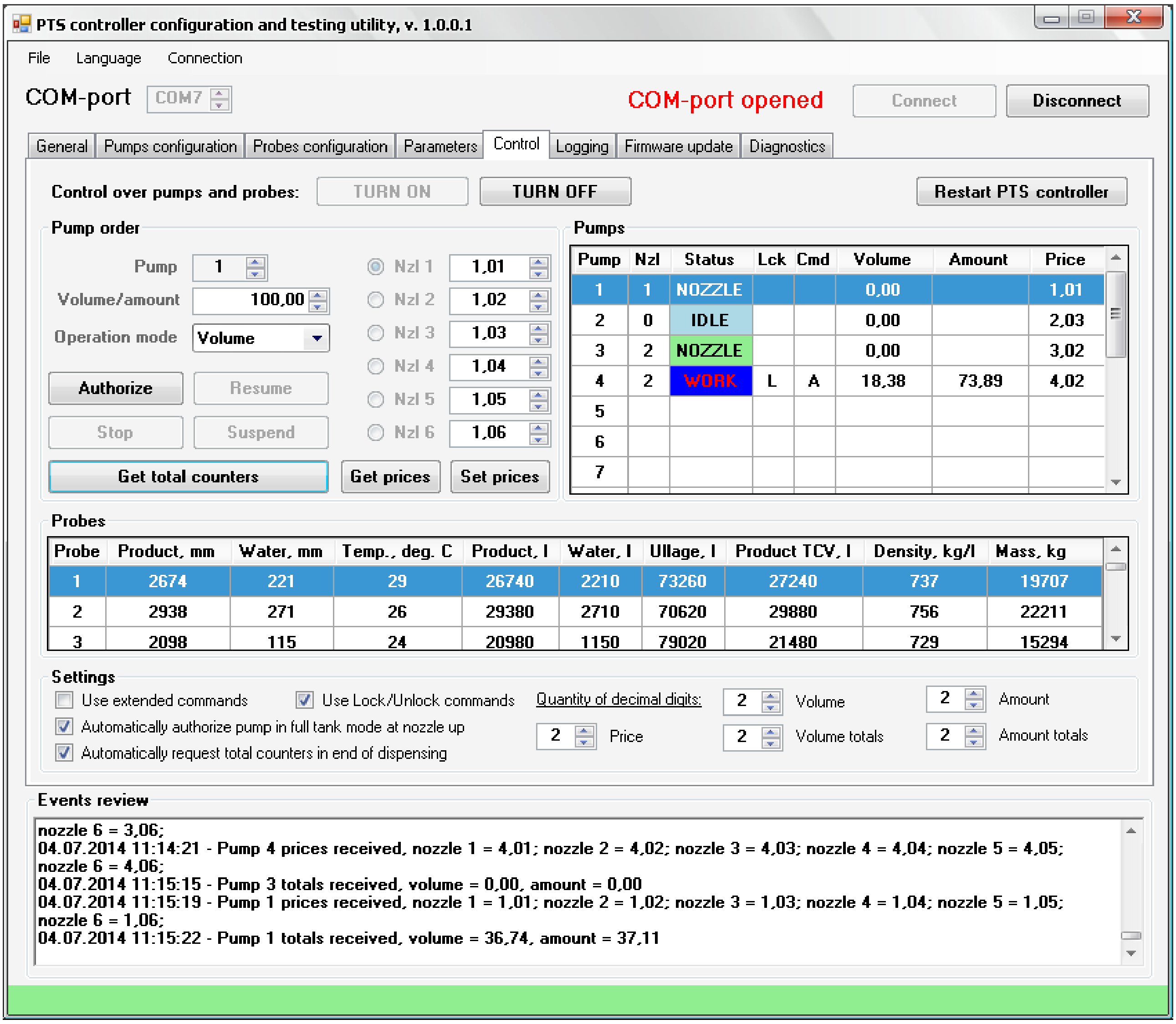
PS: If you want use *.bat files for call make, please edit it for set true path to your favorite language installation.

C# PTS controller configuration and testing utility

PTS controller configuration and testing utility is written in C# programming language in Visual Studio 2010 Express and requires .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.
- Open source utility for configuration of PTS controller, provision of control over pumps and probes, logging of operation, firmware update, self-diagnostics, others (uses classes and methods of PtsLib.dll library).



COM-port COM7

COM-port opened

Control over pumps and probes:

Pump order

Pump 1

Volume/amount 100,00

Operation mode Volume

● Nzl 1 1,01

○ Nzl 2 1,02

○ Nzl 3 1,03

○ Nzl 4 1,04

○ Nzl 5 1,05

○ Nzl 6 1,06

Pumps

Pump	Nzl	Status	Lck	Cmd	Volume	Amount	Price
1	1	NOZZLE			0,00		1,01
2	0	IDLE			0,00		2,03
3	2	NOZZLE			0,00		3,02
4	2	WORK	L	A	18,38	73,89	4,02
5							
6							
7							

Probes

Probe	Product, mm	Water, mm	Temp., deg. C	Product, l	Water, l	Ullage, l	Product TCV, l	Density, kg/l	Mass, kg
1	2674	221	29	26740	2210	73260	27240	737	19707
2	2938	271	26	29380	2710	70620	29880	756	22211
3	2098	115	24	20980	1150	79020	21480	729	15294

Settings

□ Use extended commands

☑ Use Lock/Unlock commands

☑ Automatically authorize pump in full tank mode at nozzle up

☑ Automatically request total counters in end of dispensing

Quantity of decimal digits:

2 Price

2 Volume

2 Volume totals

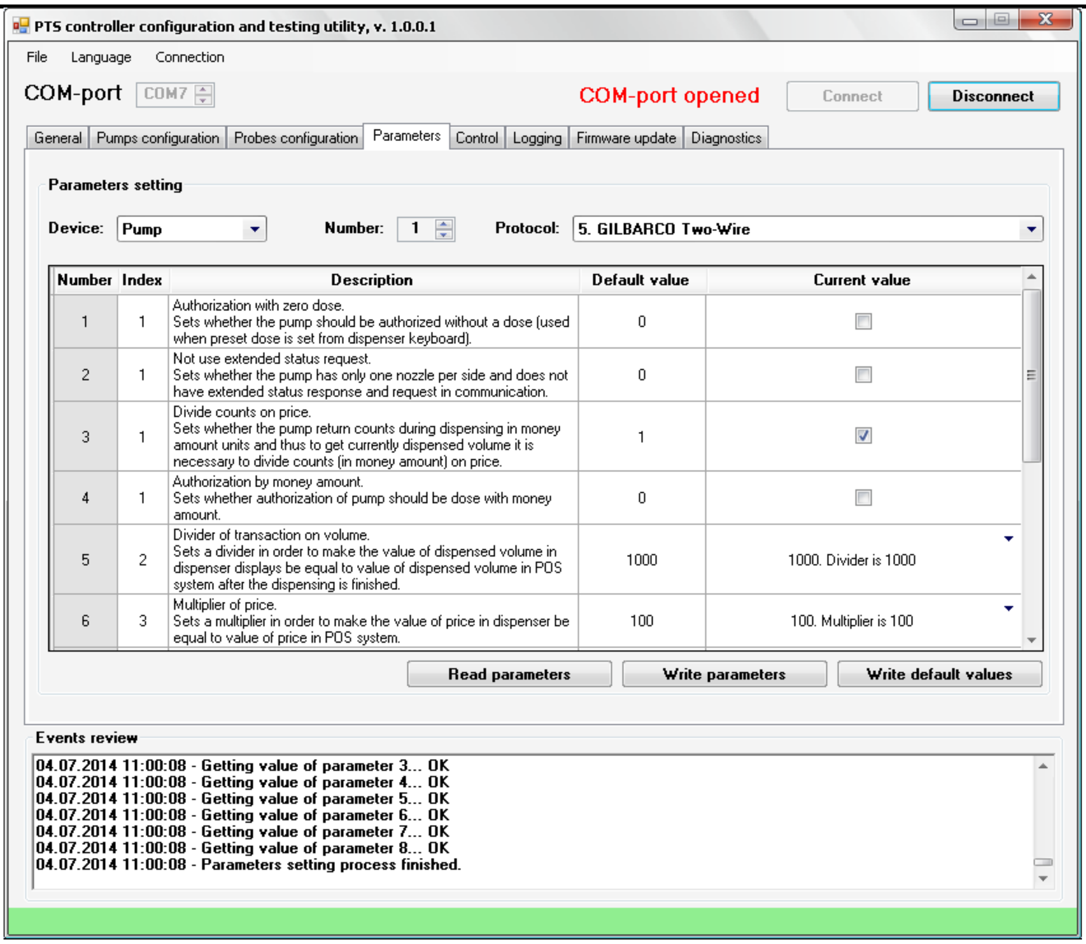
2 Amount

2 Amount totals

Events review

```
nozzle 6 = 3,06;
04.07.2014 11:14:21 - Pump 4 prices received, nozzle 1 = 4,01; nozzle 2 = 4,02; nozzle 3 = 4,03; nozzle 4 = 4,04; nozzle 5 = 4,05;
nozzle 6 = 4,06;
04.07.2014 11:15:15 - Pump 3 totals received, volume = 0,00, amount = 0,00
04.07.2014 11:15:19 - Pump 1 prices received, nozzle 1 = 1,01; nozzle 2 = 1,02; nozzle 3 = 1,03; nozzle 4 = 1,04; nozzle 5 = 1,05;
nozzle 6 = 1,06;
04.07.2014 11:15:22 - Pump 1 totals received, volume = 36,74, amount = 37,11
```

PTS controller configuration and testing utility: tab "Control"

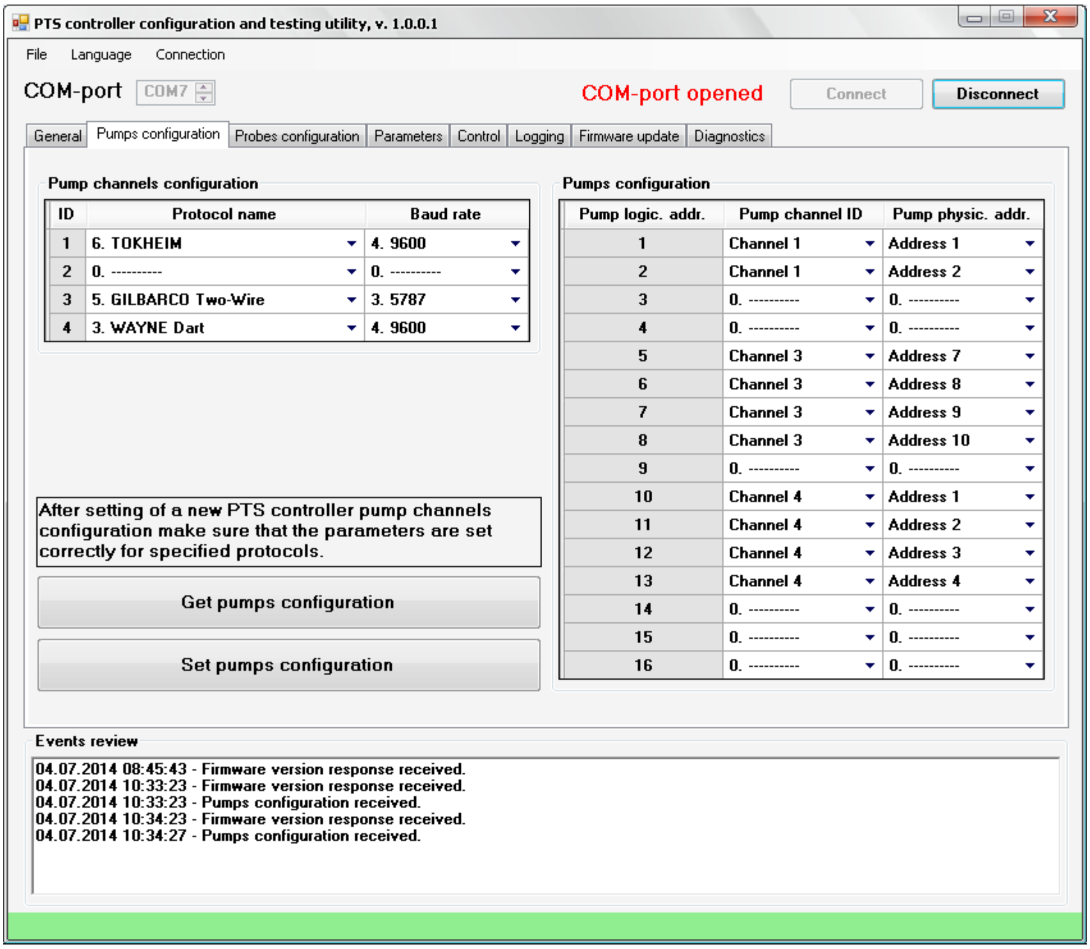


Device: Pump   Number: 1   Protocol: 5. GILBARCO Two-Wire

Number	Index	Description	Default value	Current value
1	1	Authorization with zero dose. Sets whether the pump should be authorized without a dose (used when preset dose is set from dispenser keyboard).	0	□
2	1	Not use extended status request. Sets whether the pump has only one nozzle per side and does not have extended status response and request in communication.	0	□
3	1	Divide counts on price. Sets whether the pump return counts during dispensing in money amount units and thus to get currently dispensed volume it is necessary to divide counts (in money amount) on price.	1	☑
4	1	Authorization by money amount. Sets whether authorization of pump should be dose with money amount.	0	□
5	2	Divider of transaction on volume. Sets a divider in order to make the value of dispensed volume in dispenser displays be equal to value of dispensed volume in POS system after the dispensing is finished.	1000	1000. Divider is 1000
6	3	Multiplier of price. Sets a multiplier in order to make the value of price in dispenser be equal to value of price in POS system.	100	100. Multiplier is 100

```
04.07.2014 11:00:08 - Getting value of parameter 3... OK
04.07.2014 11:00:08 - Getting value of parameter 4... OK
04.07.2014 11:00:08 - Getting value of parameter 5... OK
04.07.2014 11:00:08 - Getting value of parameter 6... OK
04.07.2014 11:00:08 - Getting value of parameter 7... OK
04.07.2014 11:00:08 - Getting value of parameter 8... OK
04.07.2014 11:00:08 - Parameters setting process finished.
```

PTS controller configuration and testing utility: tab “Parameters”



ID	Protocol name	Baud rate
1	6. TOKHEIM	4. 9600
2	0. ----------	0. ----------
3	5. GILBARCO Two-Wire	3. 5787
4	3. WAYNE Dart	4. 9600

After setting of a new PTS controller pump channels configuration make sure that the parameters are set correctly for specified protocols.

Pump logic. addr.	Pump channel ID	Pump physic. addr.
1	Channel 1	Address 1
2	Channel 1	Address 2
3	0. ----------	0. ----------
4	0. ----------	0. ----------
5	Channel 3	Address 7
6	Channel 3	Address 8
7	Channel 3	Address 9
8	Channel 3	Address 10
9	0. ----------	0. ----------
10	Channel 4	Address 1
11	Channel 4	Address 2
12	Channel 4	Address 3
13	Channel 4	Address 4
14	0. ----------	0. ----------
15	0. ----------	0. ----------
16	0. ----------	0. ----------

```
04.07.2014 08:45:43 - Firmware version response received.
04.07.2014 10:33:23 - Firmware version response received.
04.07.2014 10:33:23 - Pumps configuration received.
04.07.2014 10:34:23 - Firmware version response received.
04.07.2014 10:34:27 - Pumps configuration received.
```

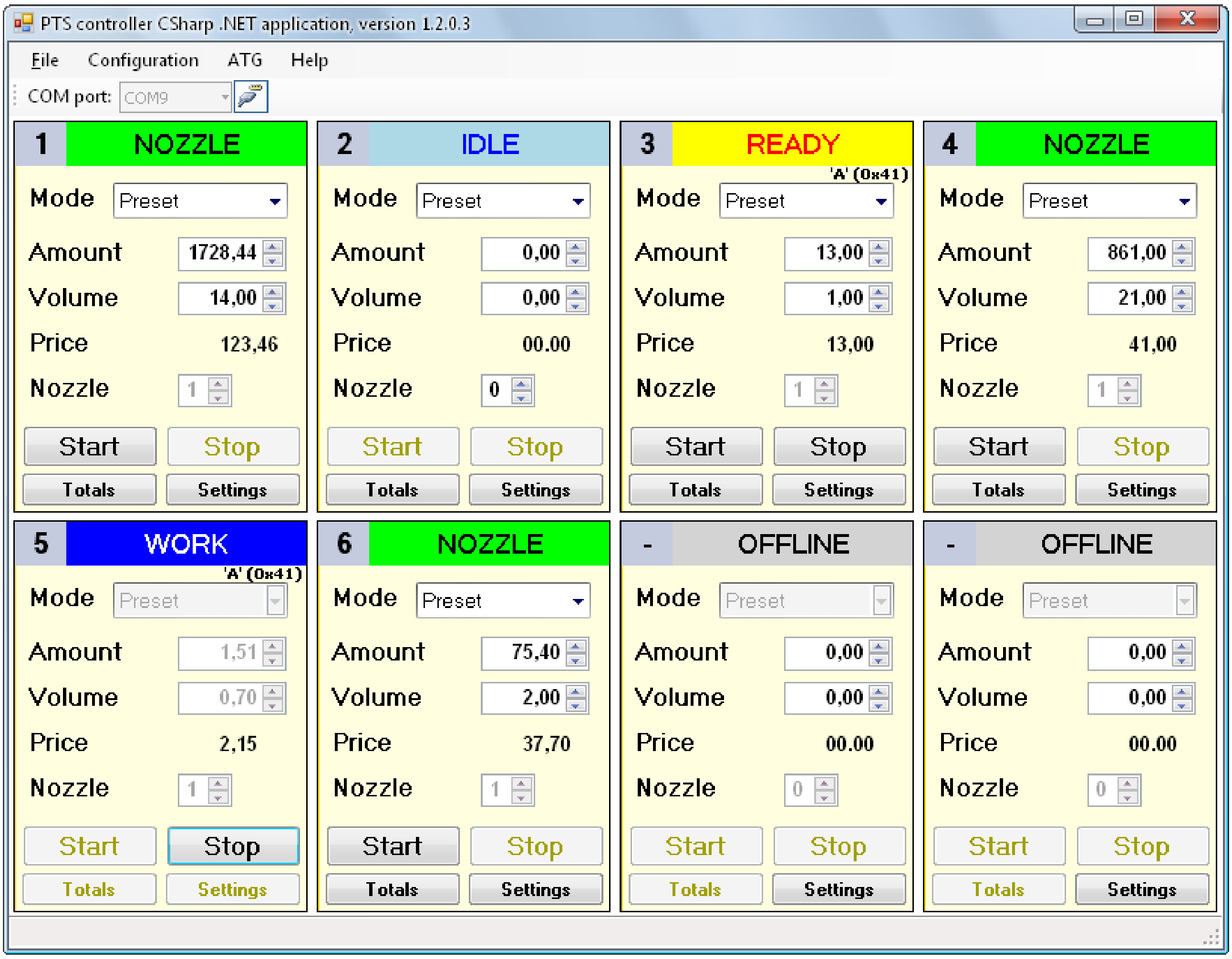
PTS controller configuration and testing utility: tab “Pumps configuration”

C# application

The application for PTS controller is written in C# programming language in Visual Studio 2010 Express and requires .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).



COM port: COM9

1 NOZZLE
Mode Preset
Amount 1728,44
Volume 14,00
Price 123,46
Nozzle 1

2 IDLE
Mode Preset
Amount 0,00
Volume 0,00
Price 00.00
Nozzle 0

3 READY
'A' (0x41)
Mode Preset
Amount 13,00
Volume 1,00
Price 13,00
Nozzle 1

4 NOZZLE
Mode Preset
Amount 861,00
Volume 21,00
Price 41,00
Nozzle 1

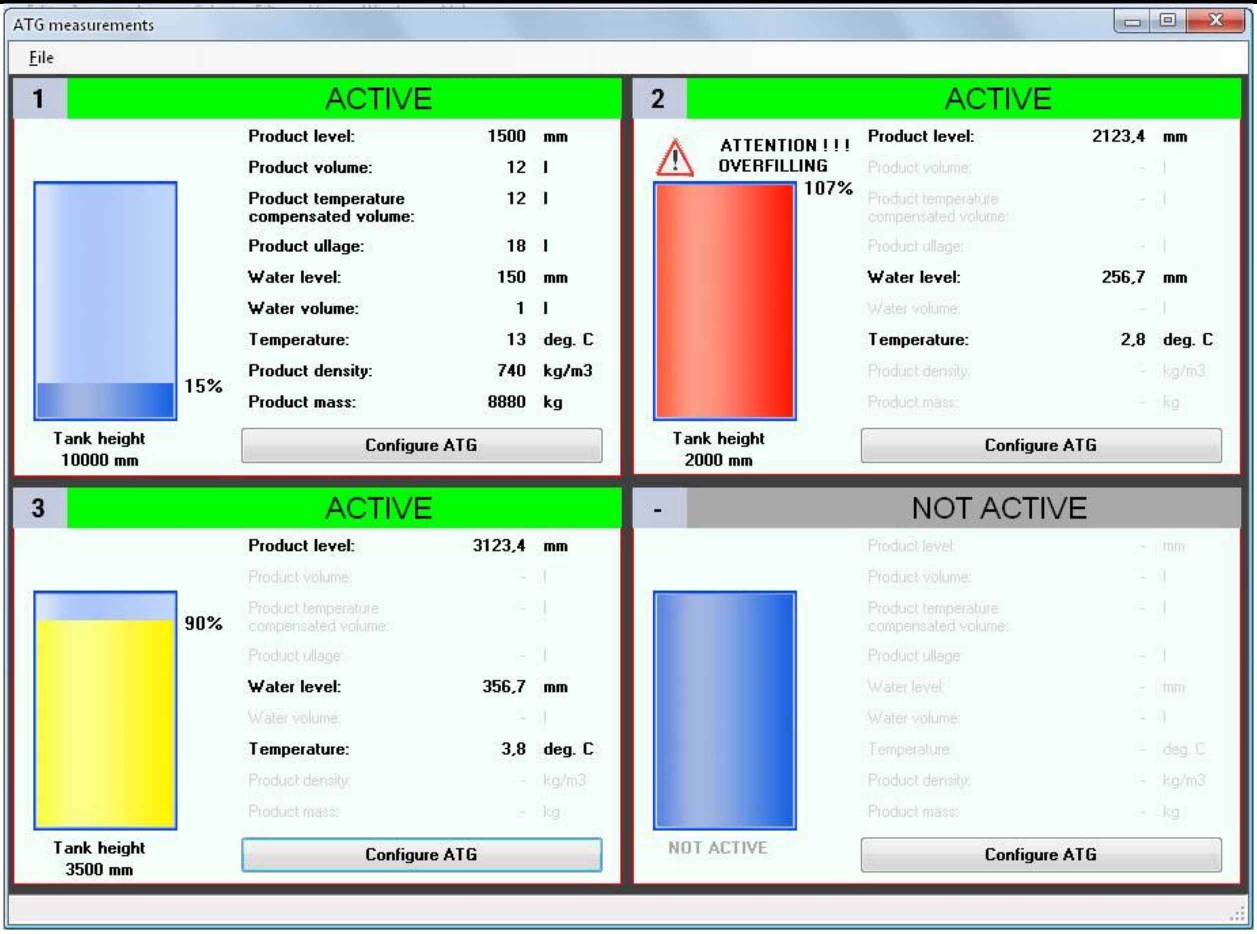
5 WORK
'A' (0x41)
Mode Preset
Amount 1,51
Volume 0,70
Price 2,15
Nozzle 1

6 NOZZLE
Mode Preset
Amount 75,40
Volume 2,00
Price 37,70
Nozzle 1

- OFFLINE
Mode Preset
Amount 0,00
Volume 0,00
Price 00.00
Nozzle 0

- OFFLINE
Mode Preset
Amount 0,00
Volume 0,00
Price 00.00
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

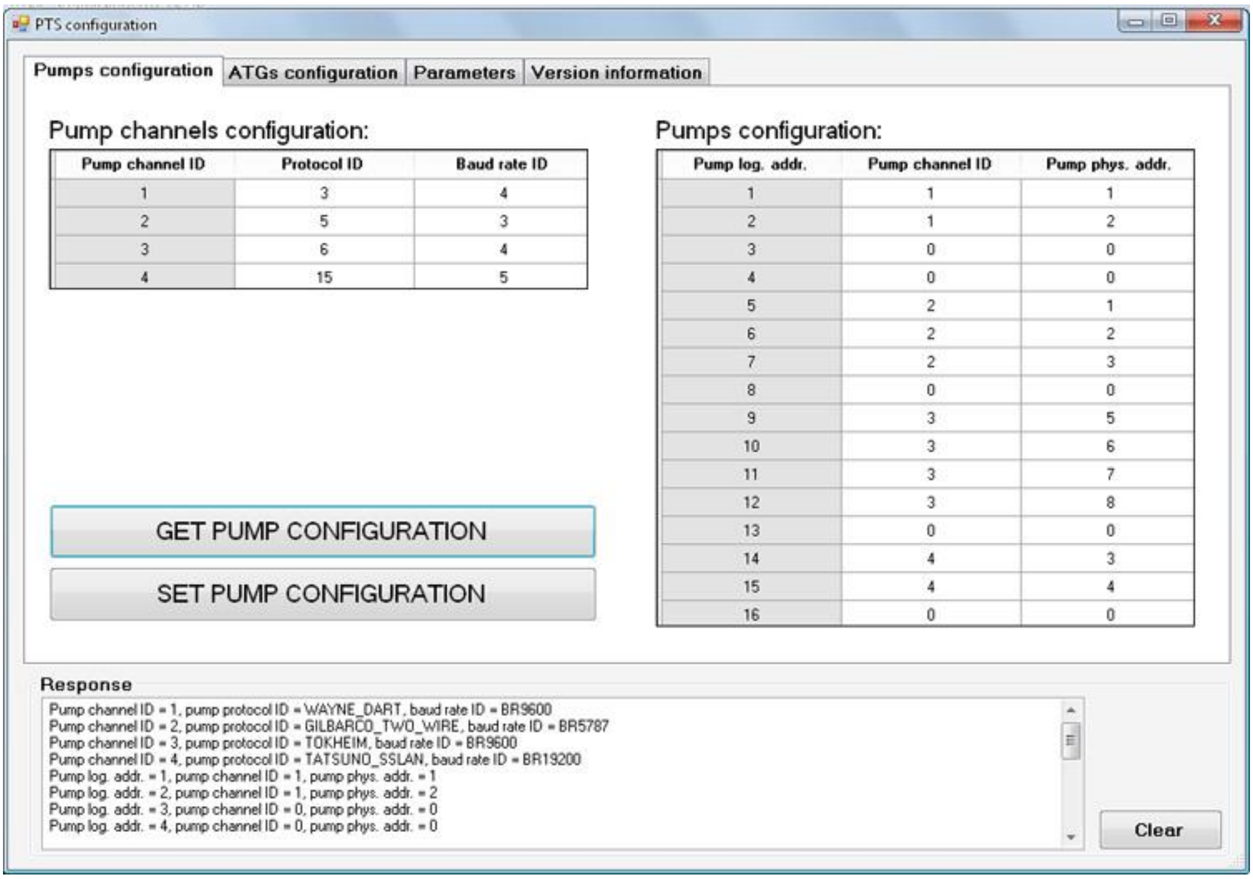
90%
Tank height
3500 mm

- 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
```

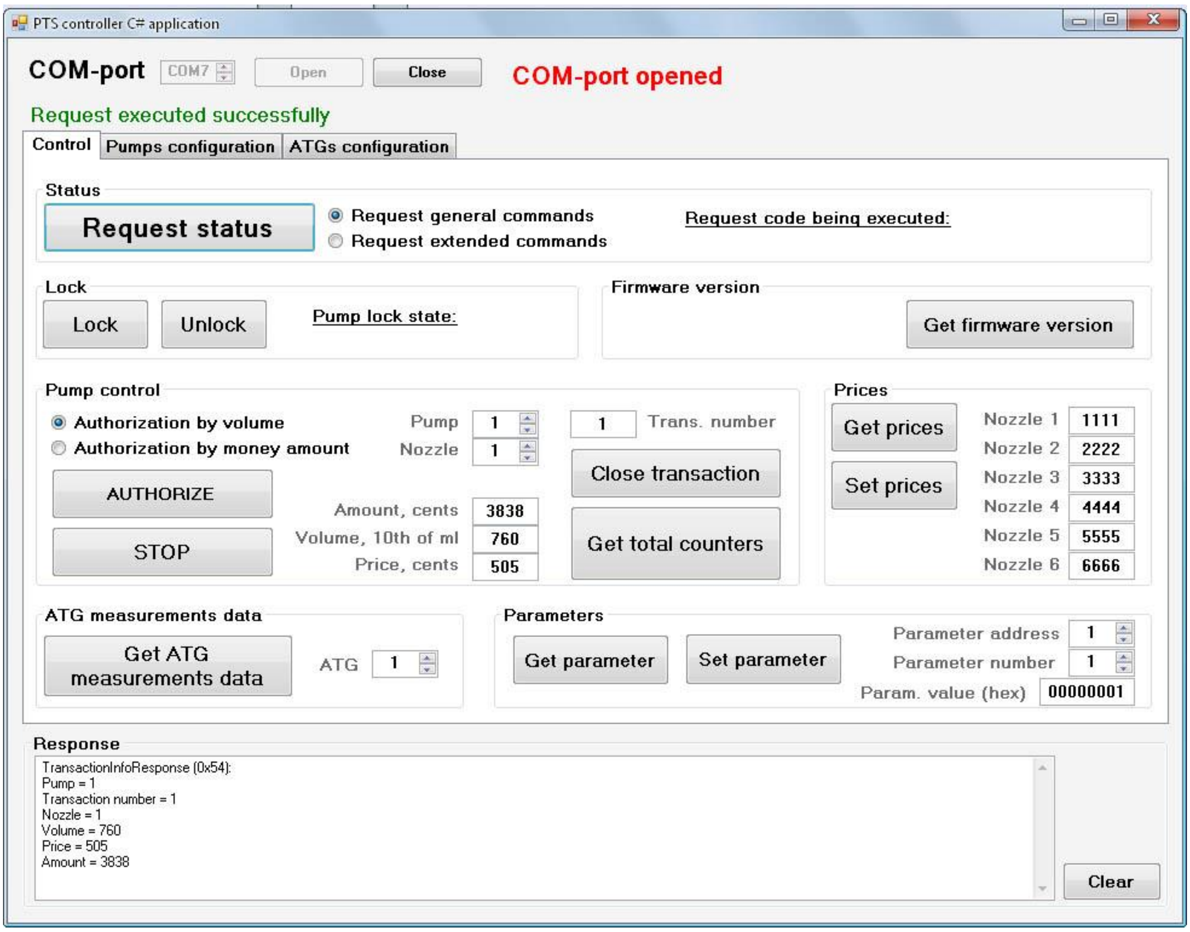
Form of PTS controller configuration: allows to set configuration for pumps and ATG systems channels and also set parameters

C# API

The API for PTS controller is written in C# programming language in Visual Studio 2010 Express and requires .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
1 Trans. number
Amount, cents 3838
Volume, 10th of ml 760
Price, cents 505

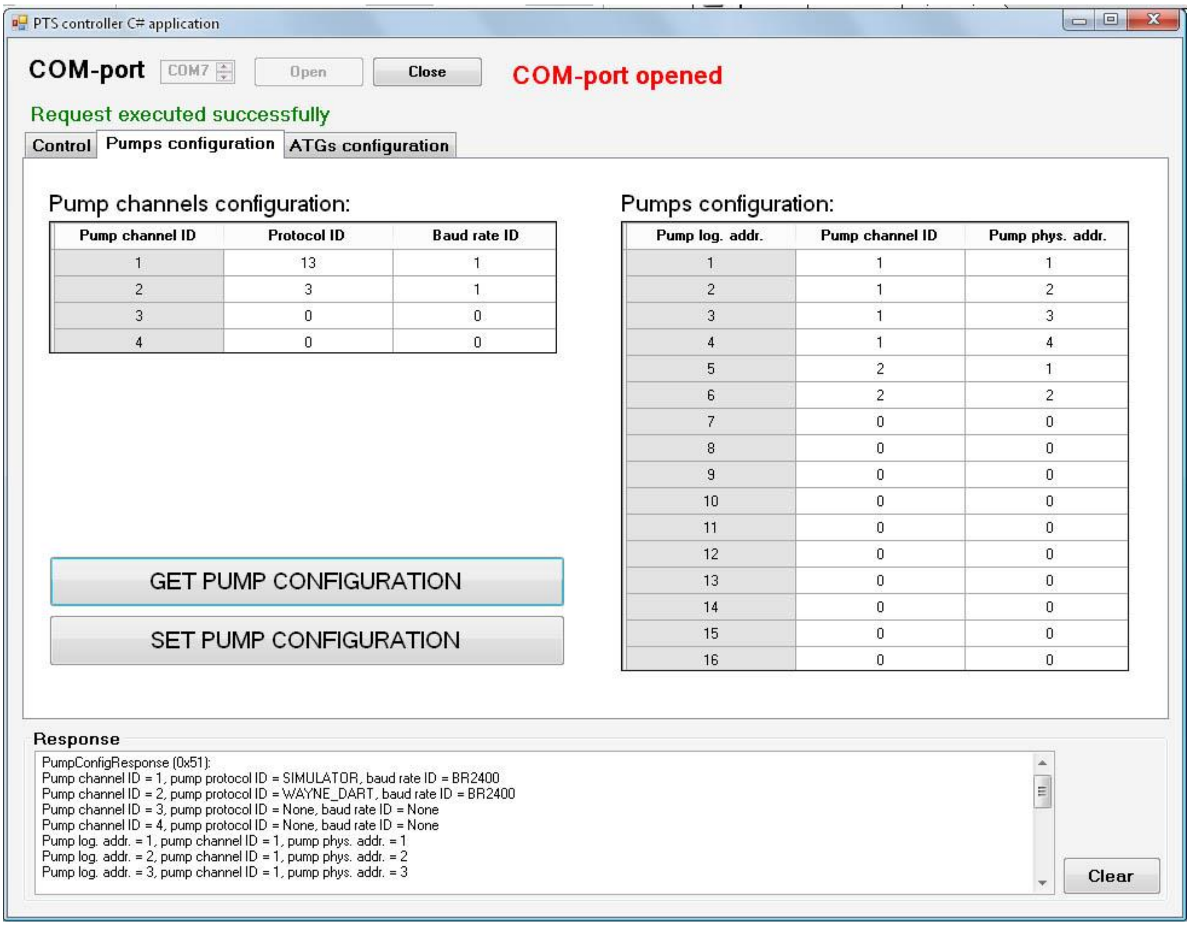
Nozzle 1	1111
Nozzle 2	2222
Nozzle 3	3333
Nozzle 4	4444
Nozzle 5	5555
Nozzle 6	6666

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

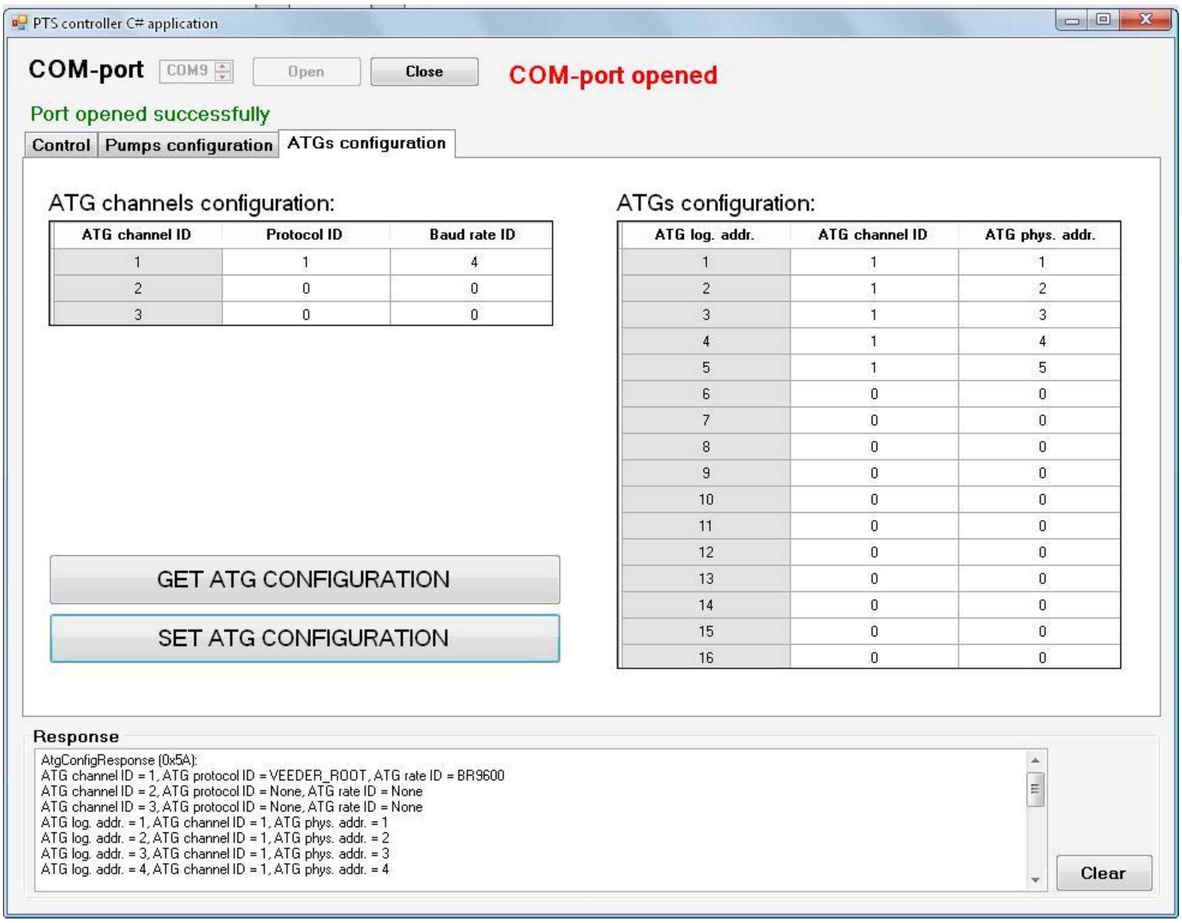
Pumps configuration:

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

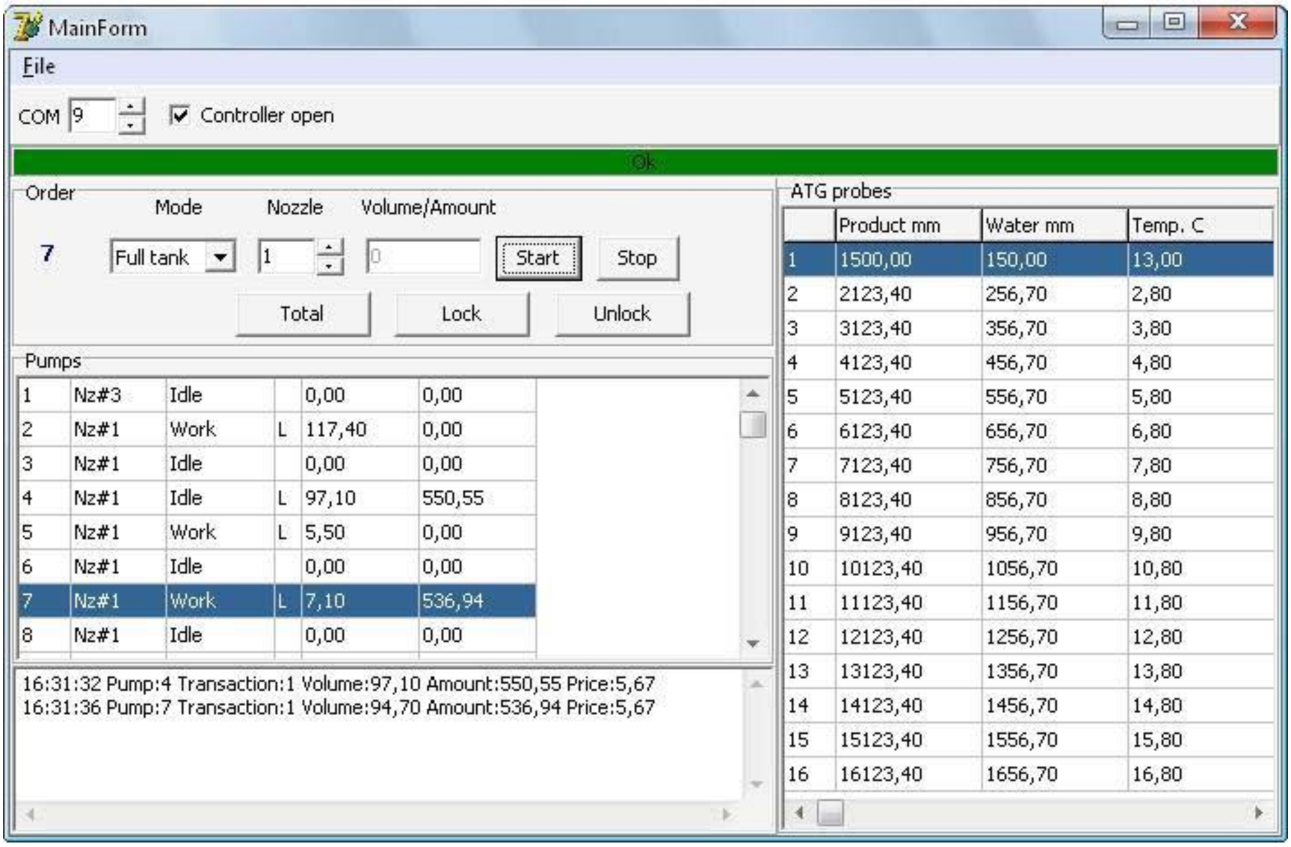
```
AtgConfigResponse (0x5A):
ATG channel ID = 1, ATG protocol ID = VEEDER_ROOT, ATG rate ID = BR9600
ATG channel ID = 2, ATG protocol ID = None, ATG rate ID = None
ATG channel ID = 3, ATG protocol ID = None, ATG rate ID = None
ATG log. addr. = 1, ATG channel ID = 1, ATG phys. addr. = 1
ATG log. addr.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

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.



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.

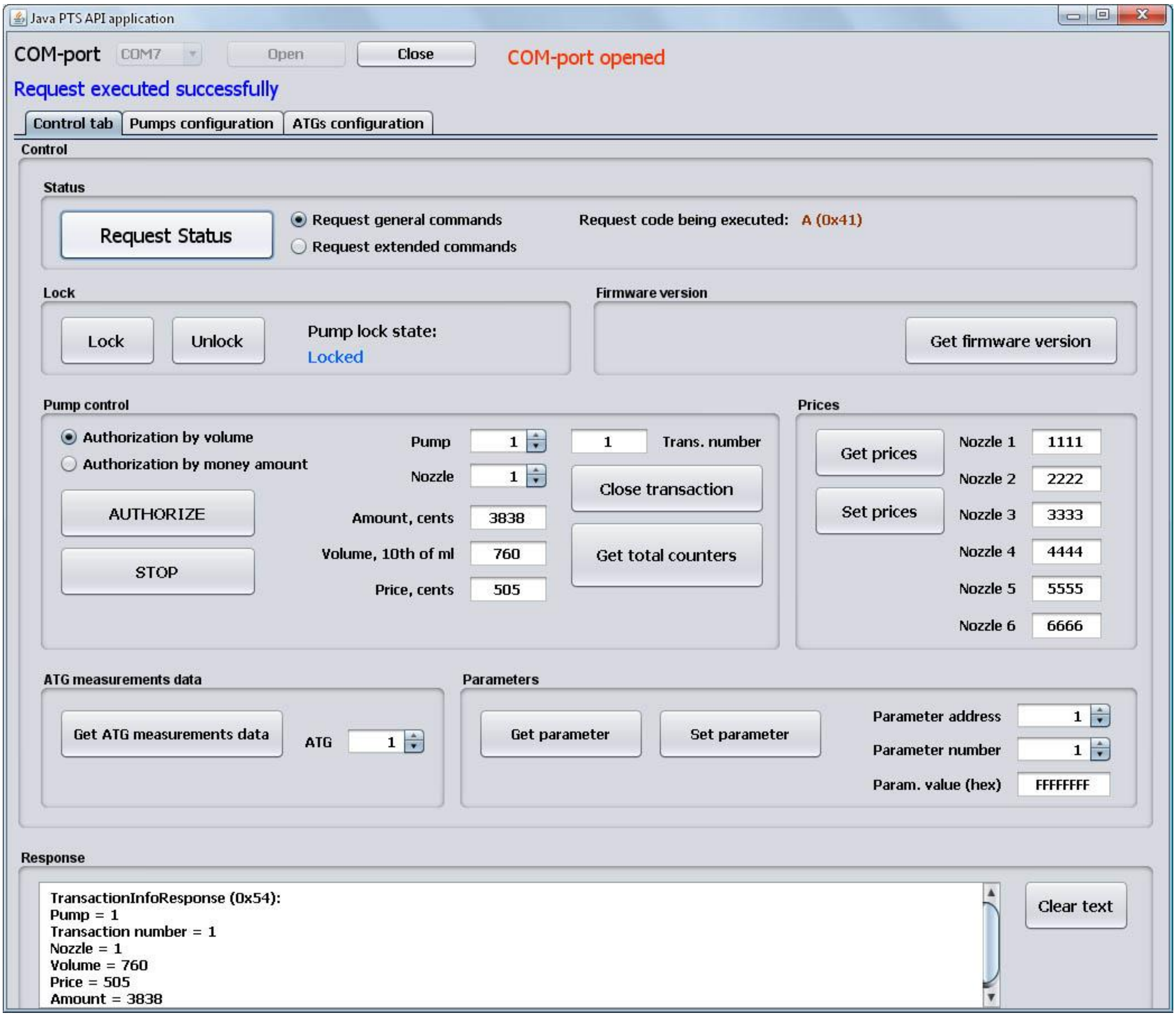
Java API

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

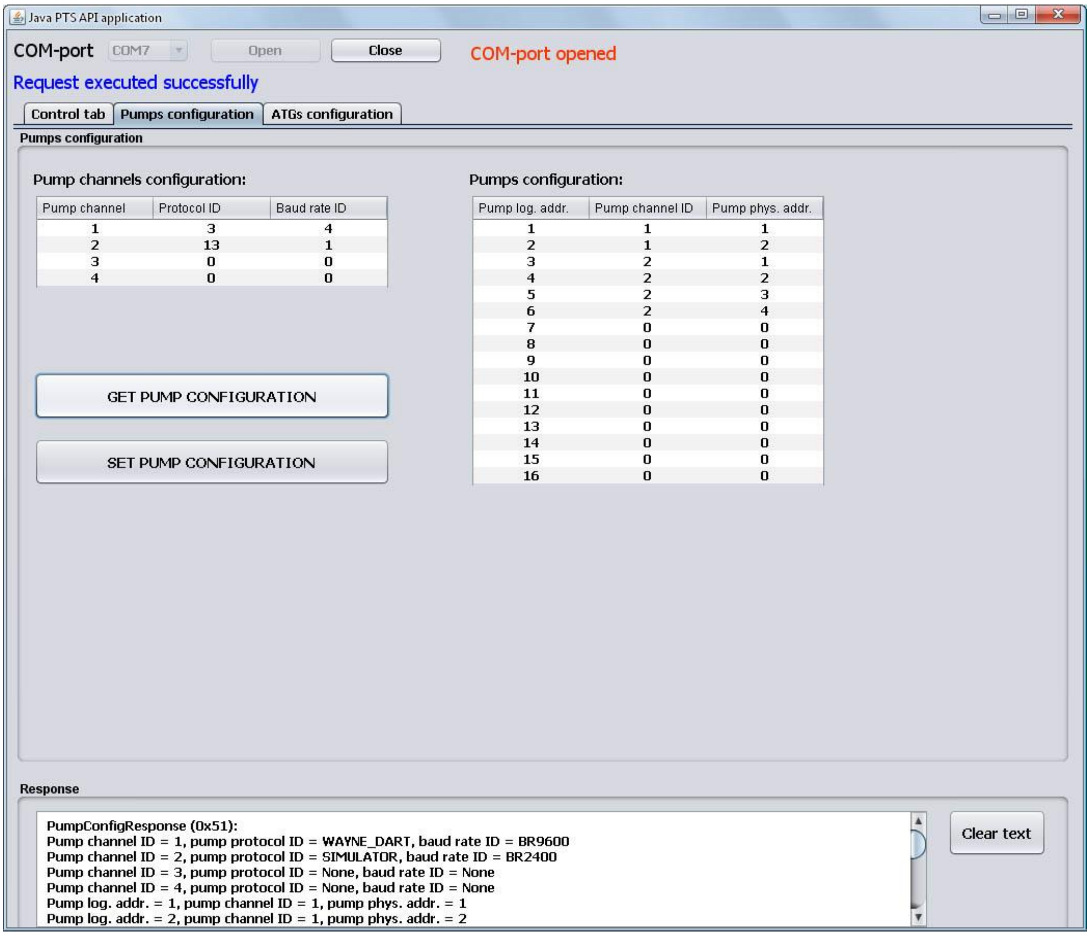
Nozzle 1 1111
Nozzle 2 2222
Nozzle 3 3333
Nozzle 4 4444
Nozzle 5 5555
Nozzle 6 6666

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:

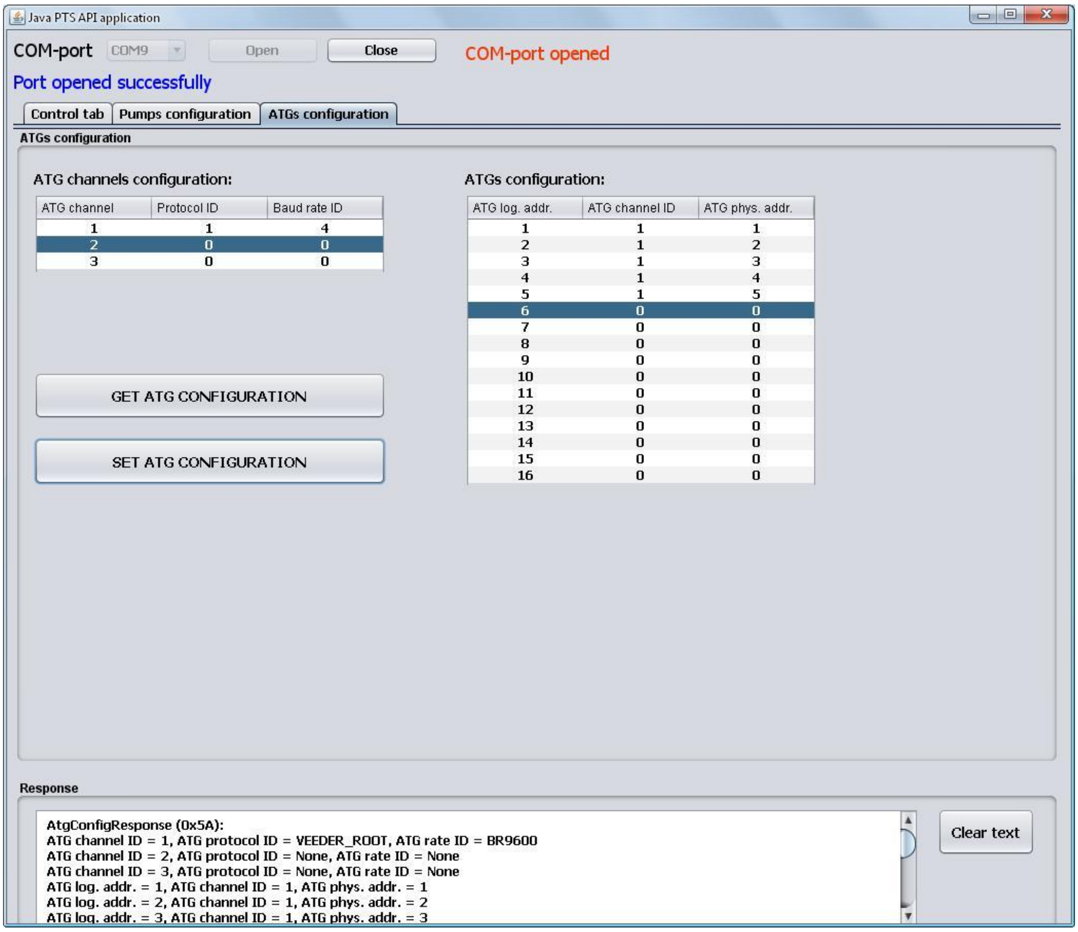
Pump channel	Protocol ID	Baud rate ID
1	3	4
2	13	1
3	0	0
4	0	0

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:

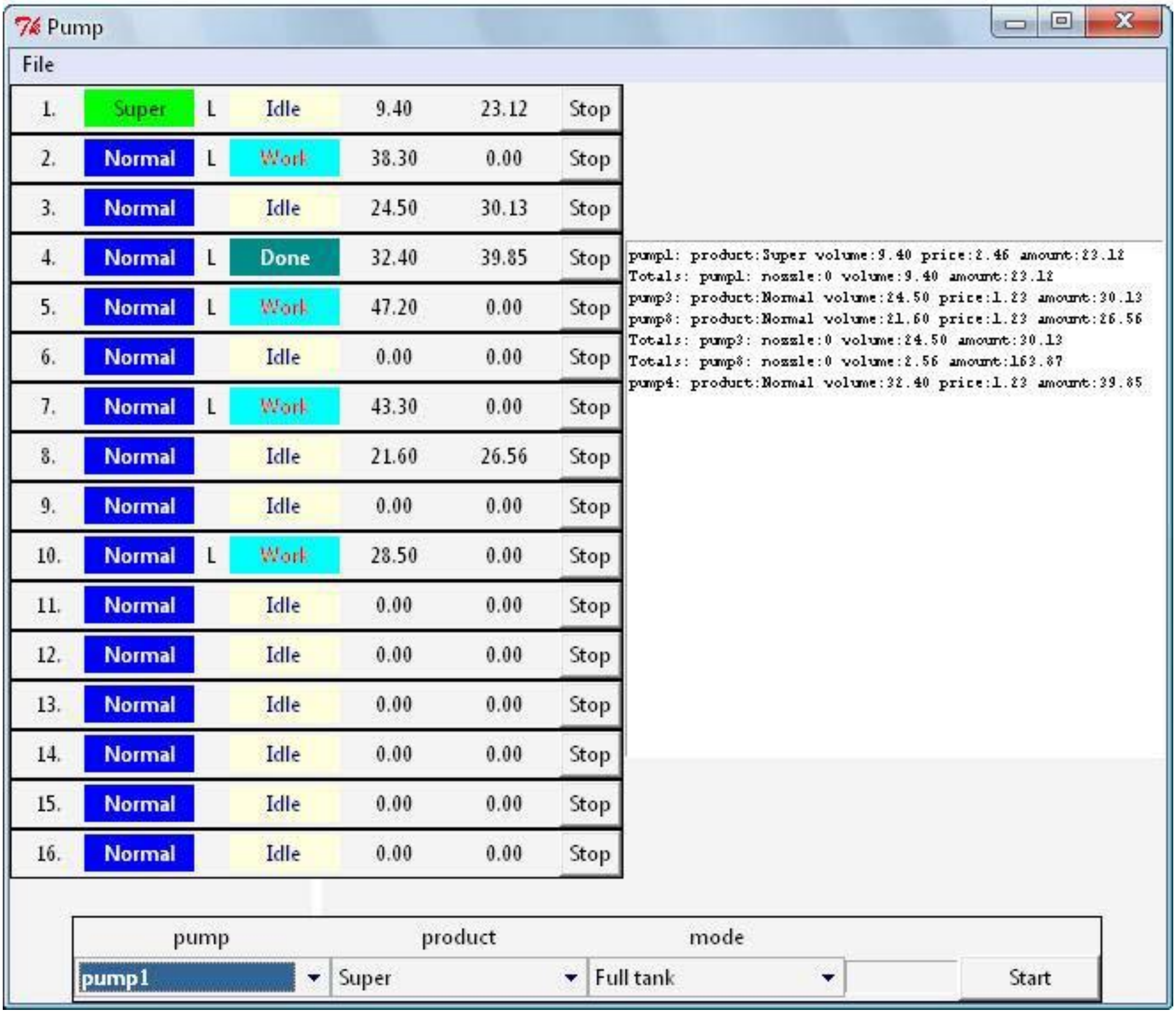
ATG log. addr.	ATG channel ID	ATG phys. addr.
1	1	1
2	1	2
3	1	3
4	1	4
5	1	5
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0
13	0	0
14	0	0
15	0	0
16	0	0

```
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

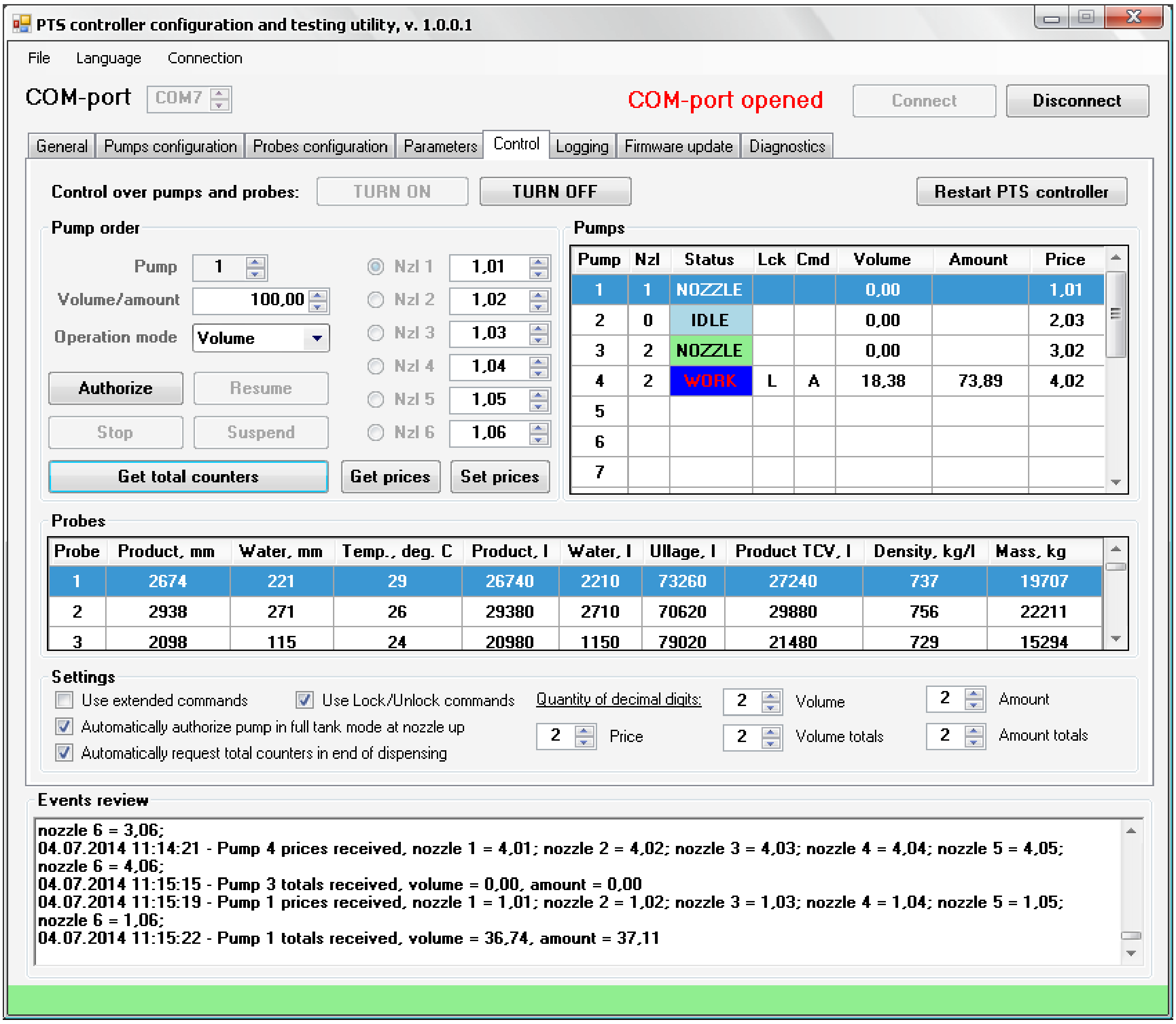
Work with the protocol can be set for work without exceptions (when error occured), and with generation of exceptions (PtsException). Parameter of initialization of use_exceptions is responsible for it.

Visual Basic .NET PTS controller configuration and testing utility

PTS controller configuration and testing utility is written in VB.NET programming language in Visual Studio 2010 Express and requires .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.
- Open source utility for configuration of PTS controller, provision of control over pumps and probes, logging of operation, firmware update, self-diagnostics, others (uses classes and methods of PtsLib.dll library).



Pump order

Pump: 1

Volume/amount: 100,00

Operation mode: Volume

● Nzl 1 1,01
○ Nzl 2 1,02
○ Nzl 3 1,03
○ Nzl 4 1,04
○ Nzl 5 1,05
○ Nzl 6 1,06

Pumps

Pump	Nzl	Status	Lck	Cmd	Volume	Amount	Price
1	1	NOZZLE			0,00		1,01
2	0	IDLE			0,00		2,03
3	2	NOZZLE			0,00		3,02
4	2	WORK	L	A	18,38	73,89	4,02
5							
6							
7							

Probes

Probe	Product, mm	Water, mm	Temp., deg. C	Product, l	Water, l	Ullage, l	Product TCV, l	Density, kg/l	Mass, kg
1	2674	221	29	26740	2210	73260	27240	737	19707
2	2938	271	26	29380	2710	70620	29880	756	22211
3	2098	115	24	20980	1150	79020	21480	729	15294

Settings

☐ Use extended commands ☑ Use Lock/Unlock commands

☑ Automatically authorize pump in full tank mode at nozzle up

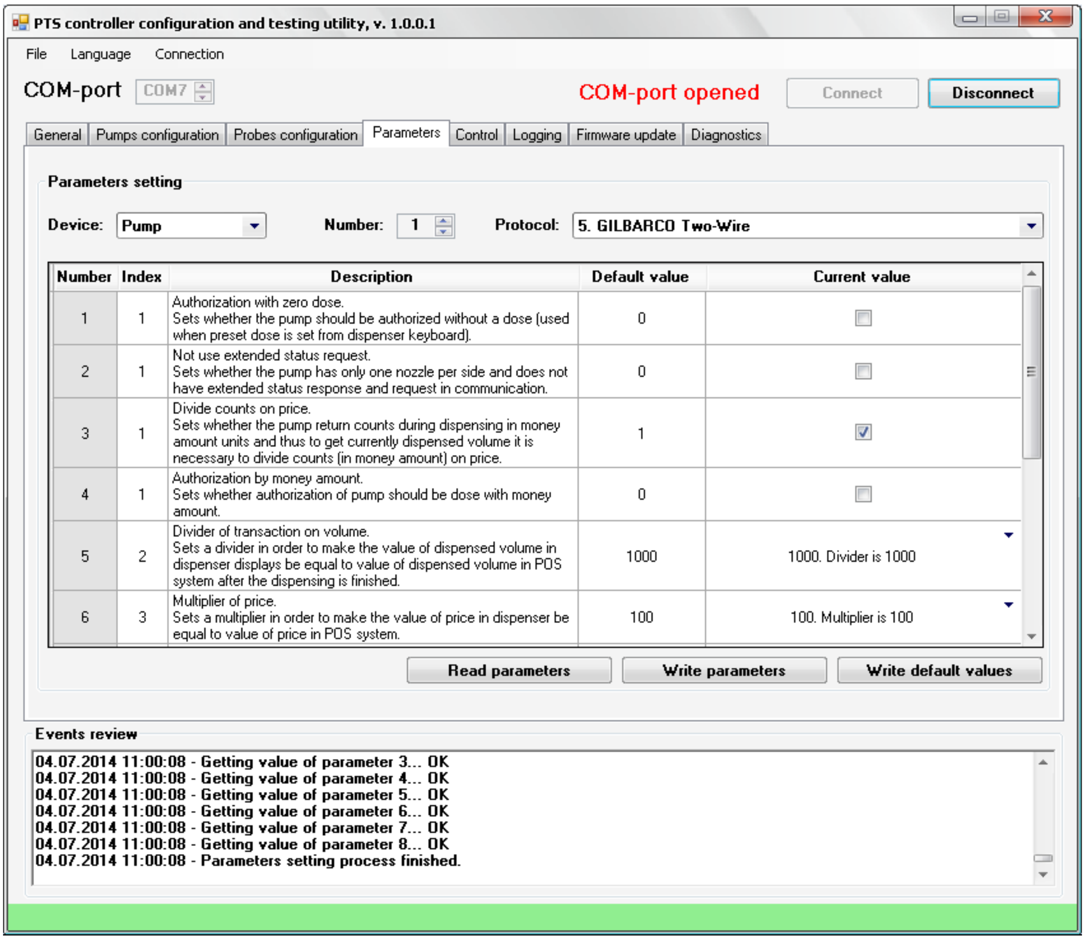
☑ Automatically request total counters in end of dispensing

Quantity of decimal digits: 2 Price; 2 Volume; 2 Volume totals; 2 Amount; 2 Amount totals

Events review

```
nozzle 6 = 3,06;
04.07.2014 11:14:21 - Pump 4 prices received, nozzle 1 = 4,01; nozzle 2 = 4,02; nozzle 3 = 4,03; nozzle 4 = 4,04; nozzle 5 = 4,05;
nozzle 6 = 4,06;
04.07.2014 11:15:15 - Pump 3 totals received, volume = 0,00, amount = 0,00
04.07.2014 11:15:19 - Pump 1 prices received, nozzle 1 = 1,01; nozzle 2 = 1,02; nozzle 3 = 1,03; nozzle 4 = 1,04; nozzle 5 = 1,05;
nozzle 6 = 1,06;
04.07.2014 11:15:22 - Pump 1 totals received, volume = 36,74, amount = 37,11
```

PTS controller configuration and testing utility: tab "Control"

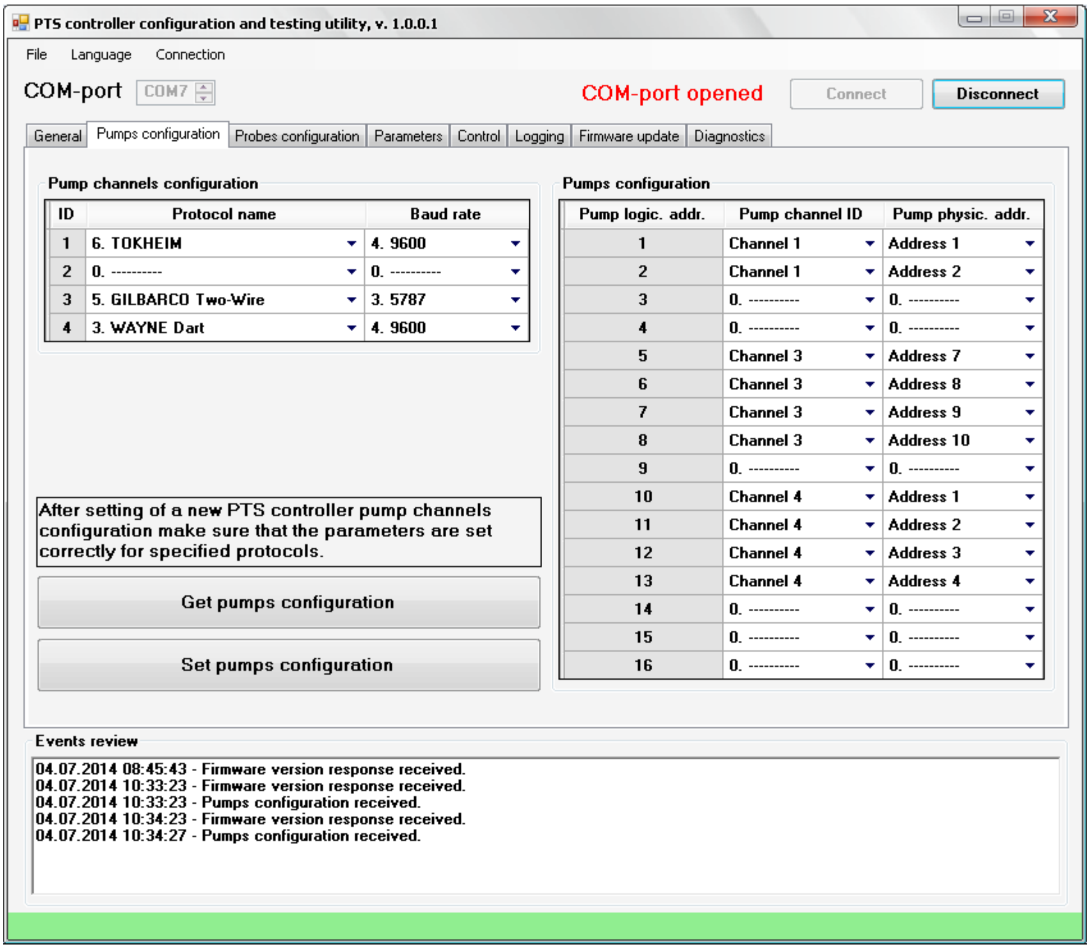


Device: Pump Number: 1 Protocol: 5. GILBARCO Two-Wire

Number	Index	Description	Default value	Current value
1	1	Authorization with zero dose. Sets whether the pump should be authorized without a dose (used when preset dose is set from dispenser keyboard).	0	☐
2	1	Not use extended status request. Sets whether the pump has only one nozzle per side and does not have extended status response and request in communication.	0	☐
3	1	Divide counts on price. Sets whether the pump return counts during dispensing in money amount units and thus to get currently dispensed volume it is necessary to divide counts (in money amount) on price.	1	☑
4	1	Authorization by money amount. Sets whether authorization of pump should be dose with money amount.	0	☐
5	2	Divider of transaction on volume. Sets a divider in order to make the value of dispensed volume in dispenser displays be equal to value of dispensed volume in POS system after the dispensing is finished.	1000	1000. Divider is 1000
6	3	Multiplier of price. Sets a multiplier in order to make the value of price in dispenser be equal to value of price in POS system.	100	100. Multiplier is 100

```
04.07.2014 11:00:08 - Getting value of parameter 3... OK
04.07.2014 11:00:08 - Getting value of parameter 4... OK
04.07.2014 11:00:08 - Getting value of parameter 5... OK
04.07.2014 11:00:08 - Getting value of parameter 6... OK
04.07.2014 11:00:08 - Getting value of parameter 7... OK
04.07.2014 11:00:08 - Getting value of parameter 8... OK
04.07.2014 11:00:08 - Parameters setting process finished.
```

PTS controller configuration and testing utility: tab "Parameters"



ID	Protocol name	Baud rate
1	6. TOKHEIM	4. 9600
2	0. ----------	0. ----------
3	5. GILBARCO Two-Wire	3. 5787
4	3. WAYNE Dart	4. 9600

After setting of a new PTS controller pump channels configuration make sure that the parameters are set correctly for specified protocols.

Pump logic. addr.	Pump channel ID	Pump physic. addr.
1	Channel 1	Address 1
2	Channel 1	Address 2
3	0. ----------	0. ----------
4	0. ----------	0. ----------
5	Channel 3	Address 7
6	Channel 3	Address 8
7	Channel 3	Address 9
8	Channel 3	Address 10
9	0. ----------	0. ----------
10	Channel 4	Address 1
11	Channel 4	Address 2
12	Channel 4	Address 3
13	Channel 4	Address 4
14	0. ----------	0. ----------
15	0. ----------	0. ----------
16	0. ----------	0. ----------

```
04.07.2014 08:45:43 - Firmware version response received.
04.07.2014 10:33:23 - Firmware version response received.
04.07.2014 10:33:23 - Pumps configuration received.
04.07.2014 10:34:23 - Firmware version response received.
04.07.2014 10:34:27 - Pumps configuration received.
```

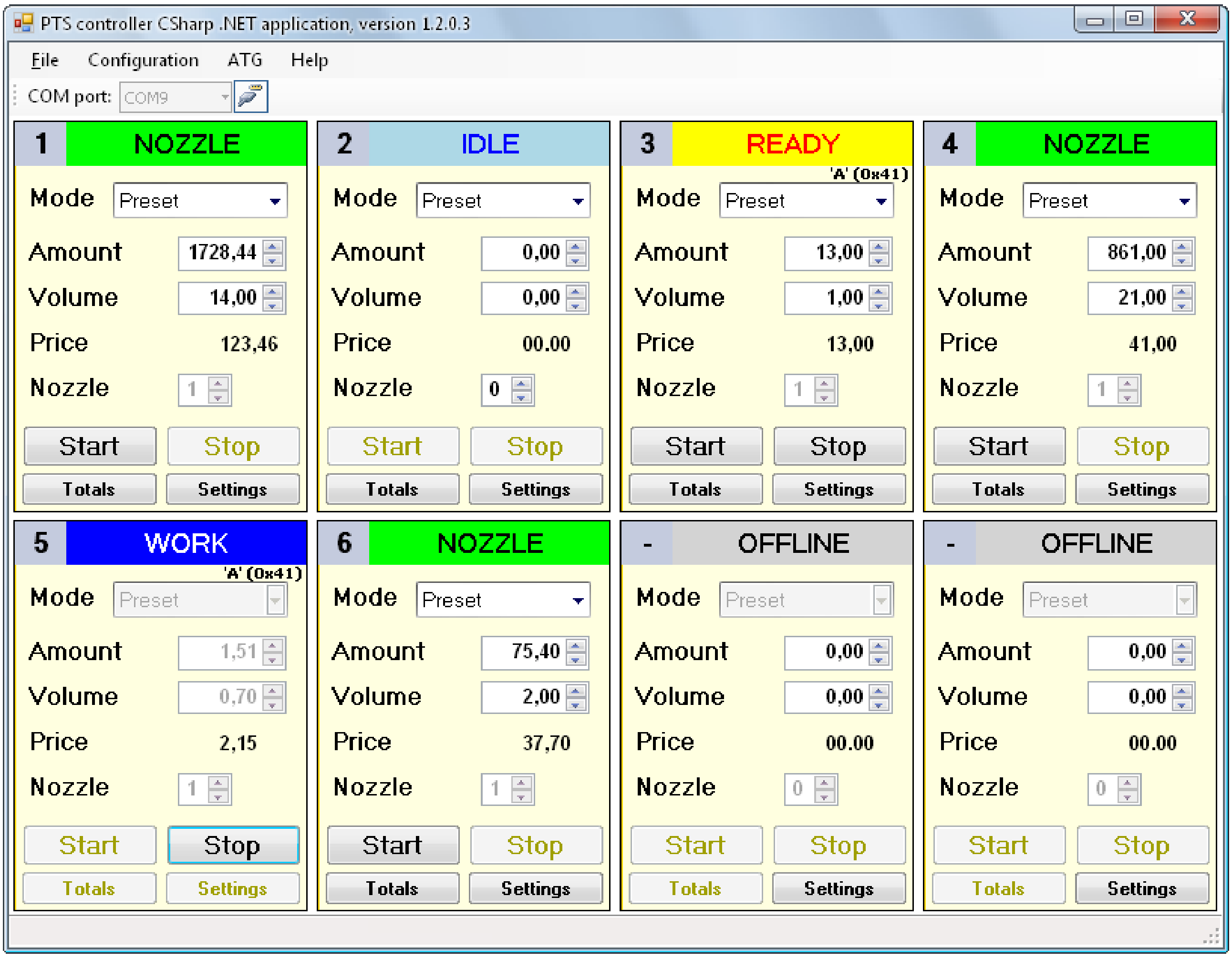
PTS controller configuration and testing utility: tab "Pumps configuration"

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).

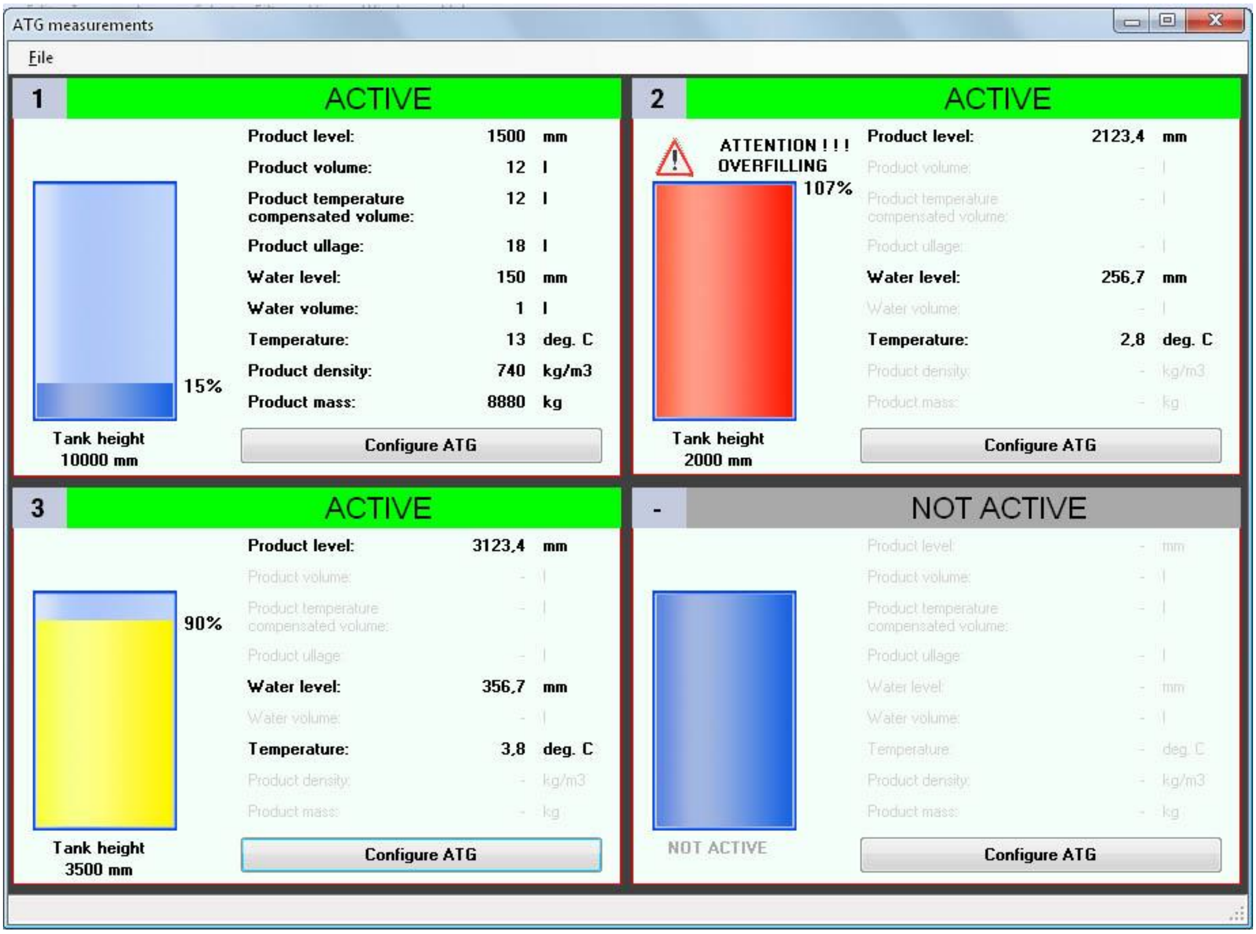


COM port: COM9

	1 NOZZLE	2 IDLE	3 READY 'A' (0x41)	4 NOZZLE
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 WORK 'A' (0x41)	6 NOZZLE	- OFFLINE	- OFFLINE
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

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



1 — ACTIVE (15%, Tank height 10000 mm)

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

2 — ACTIVE (ATTENTION ! ! ! OVERFILLING, 107%, Tank height 2000 mm)

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

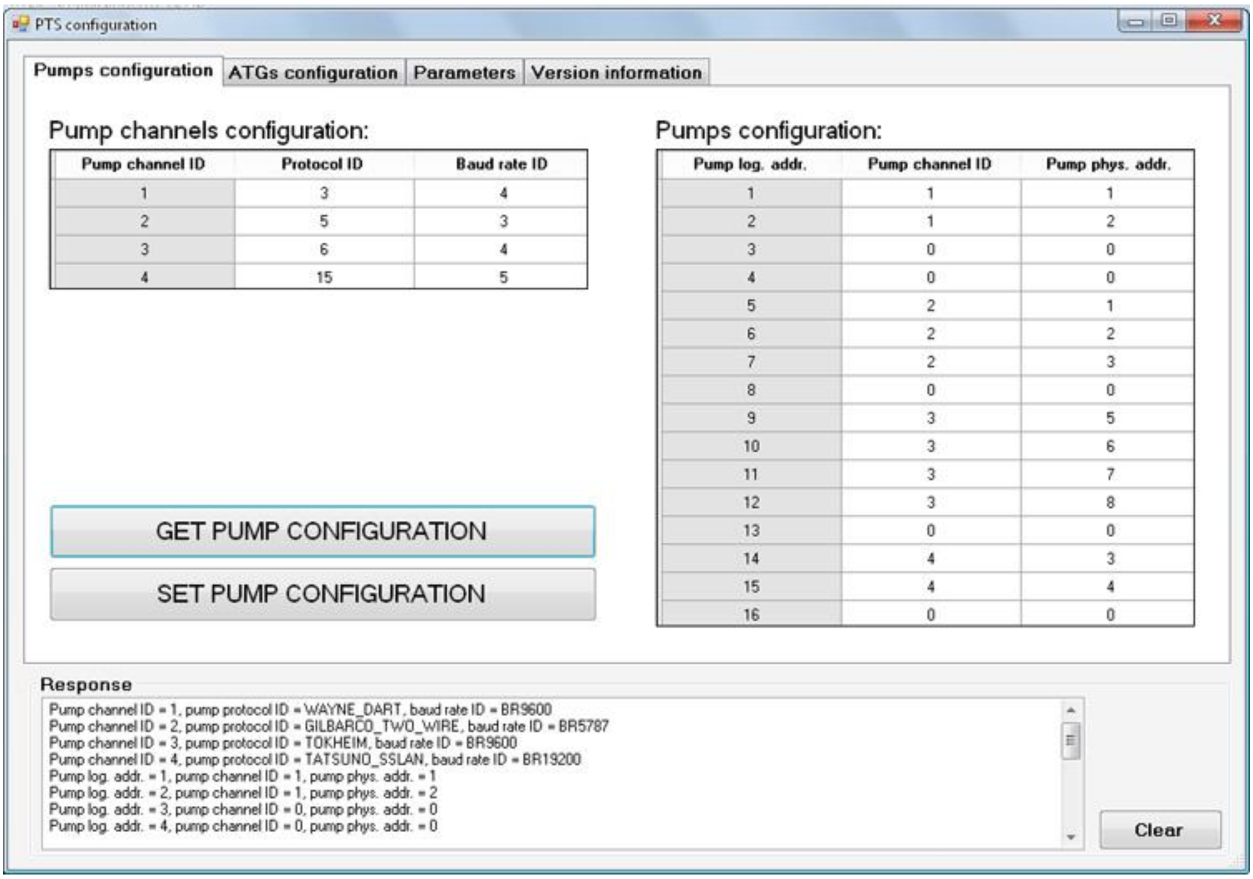
3 — ACTIVE (90%, Tank height 3500 mm)

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

- — NOT ACTIVE (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

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
```

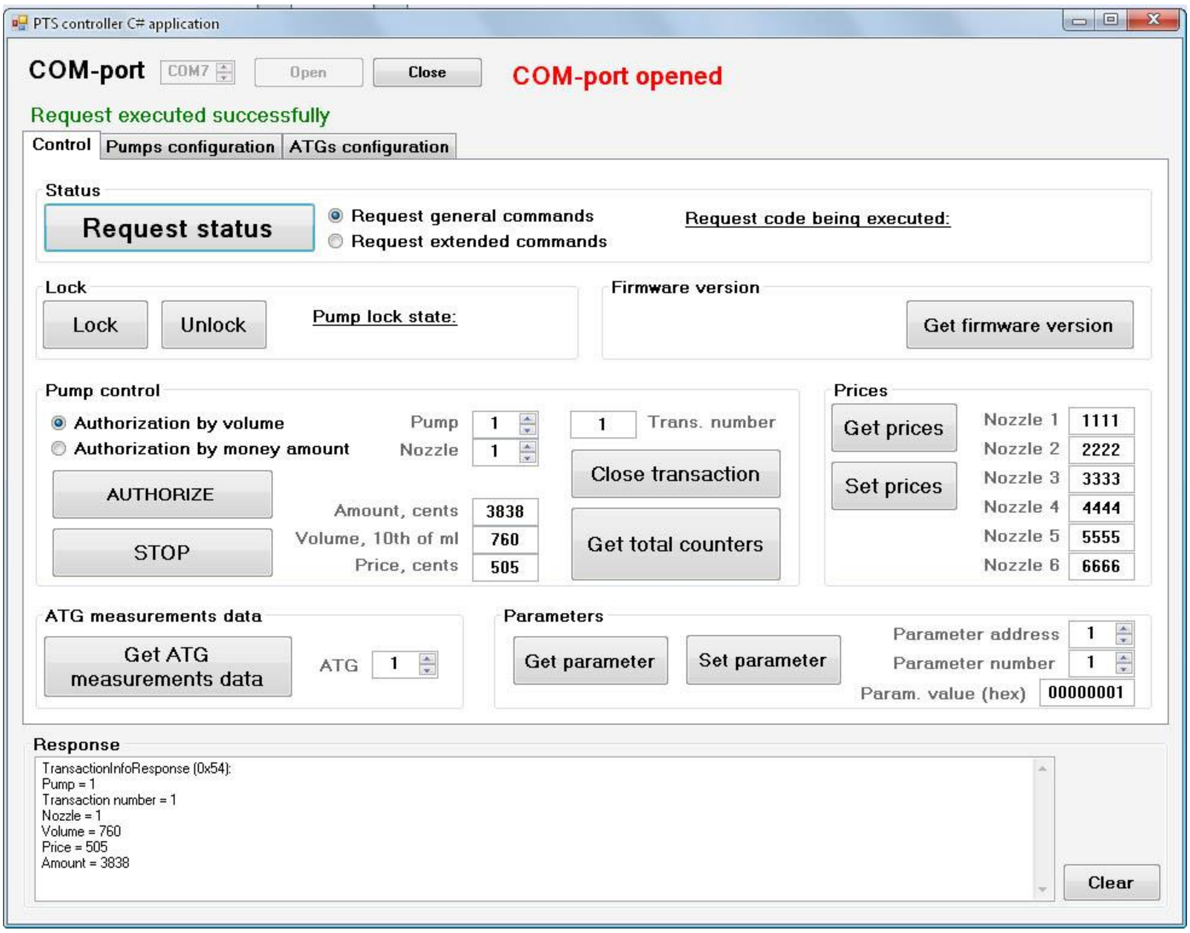
Form of PTS controller configuration: allows to set configuration for pumps and ATG systems channels and also set parameters

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

Request code being executed:

Pump lock state:

● 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

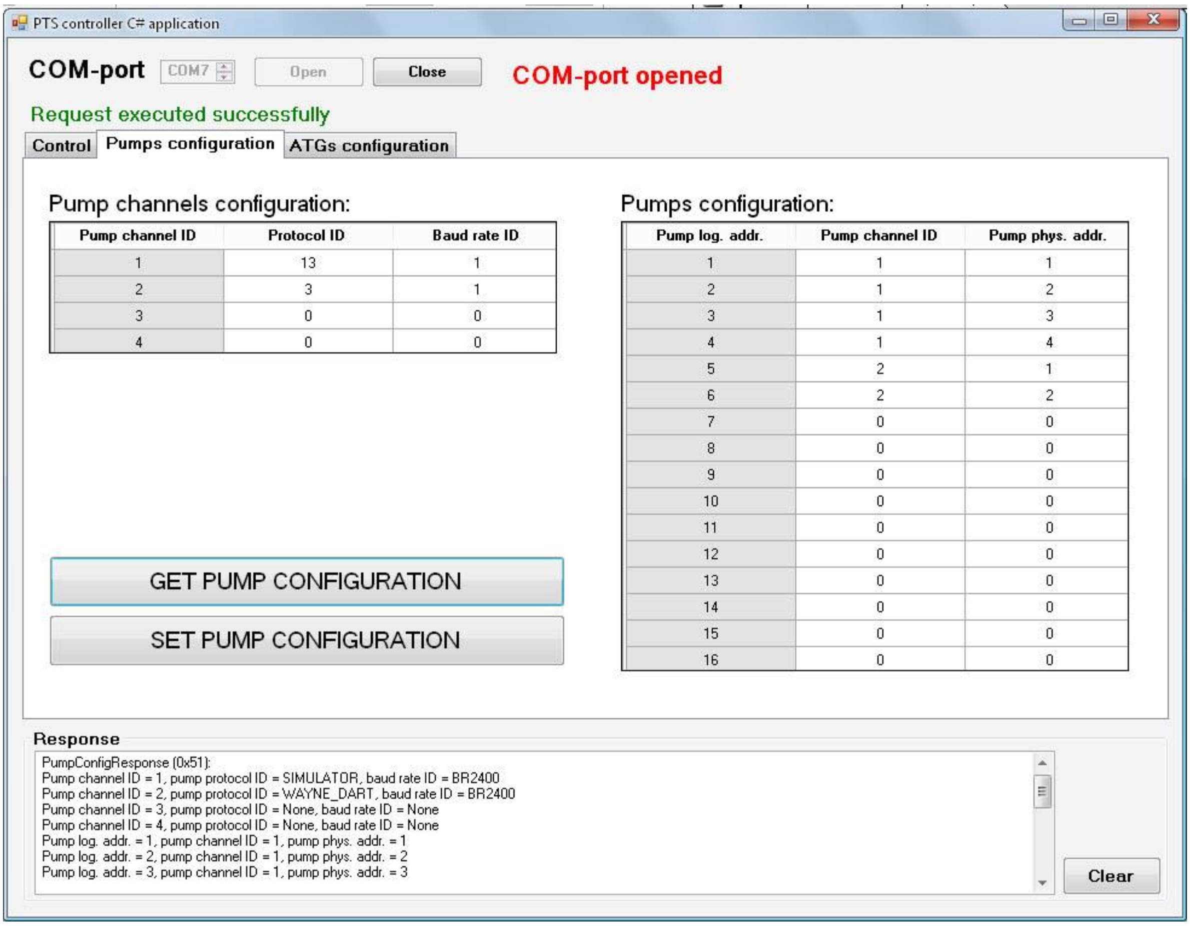
Nozzle 1	1111
Nozzle 2	2222
Nozzle 3	3333
Nozzle 4	4444
Nozzle 5	5555
Nozzle 6	6666

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

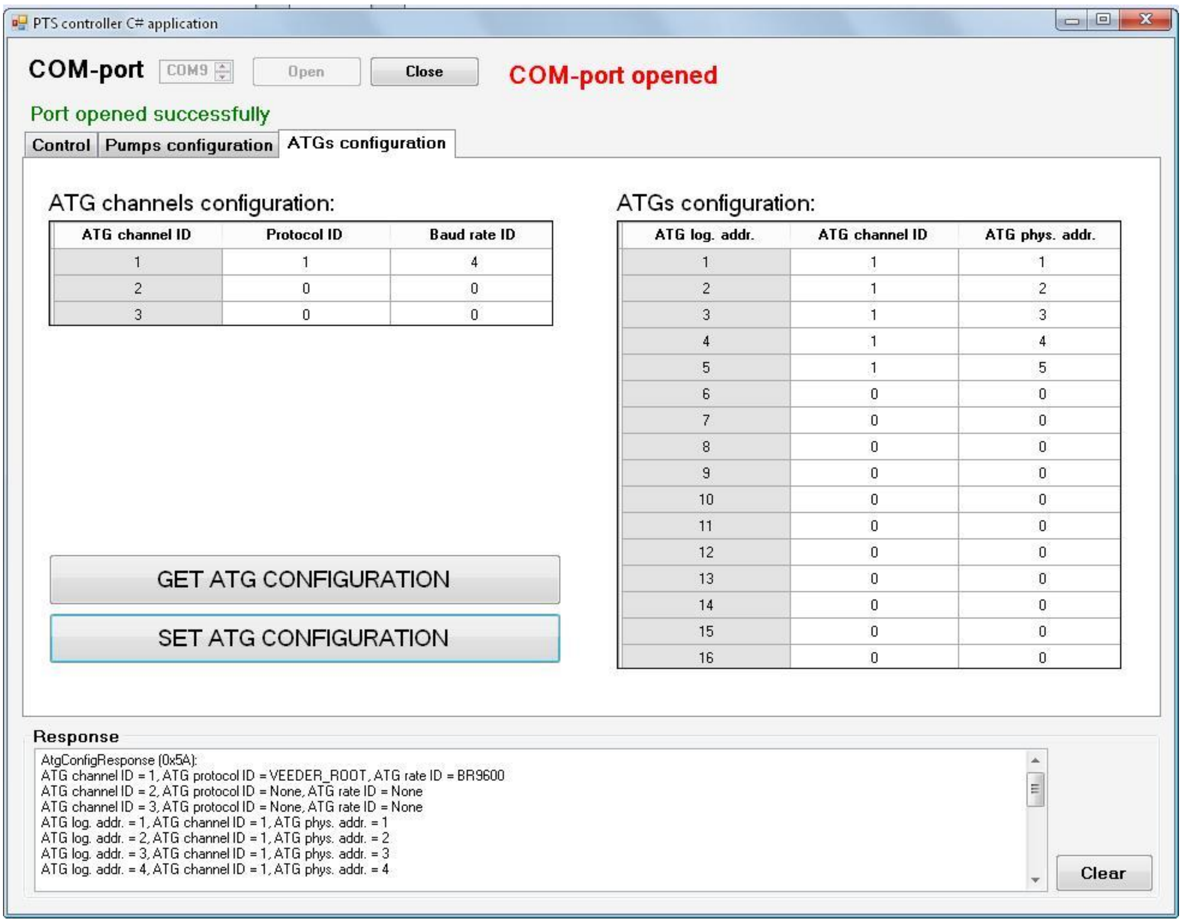
Pumps configuration:

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

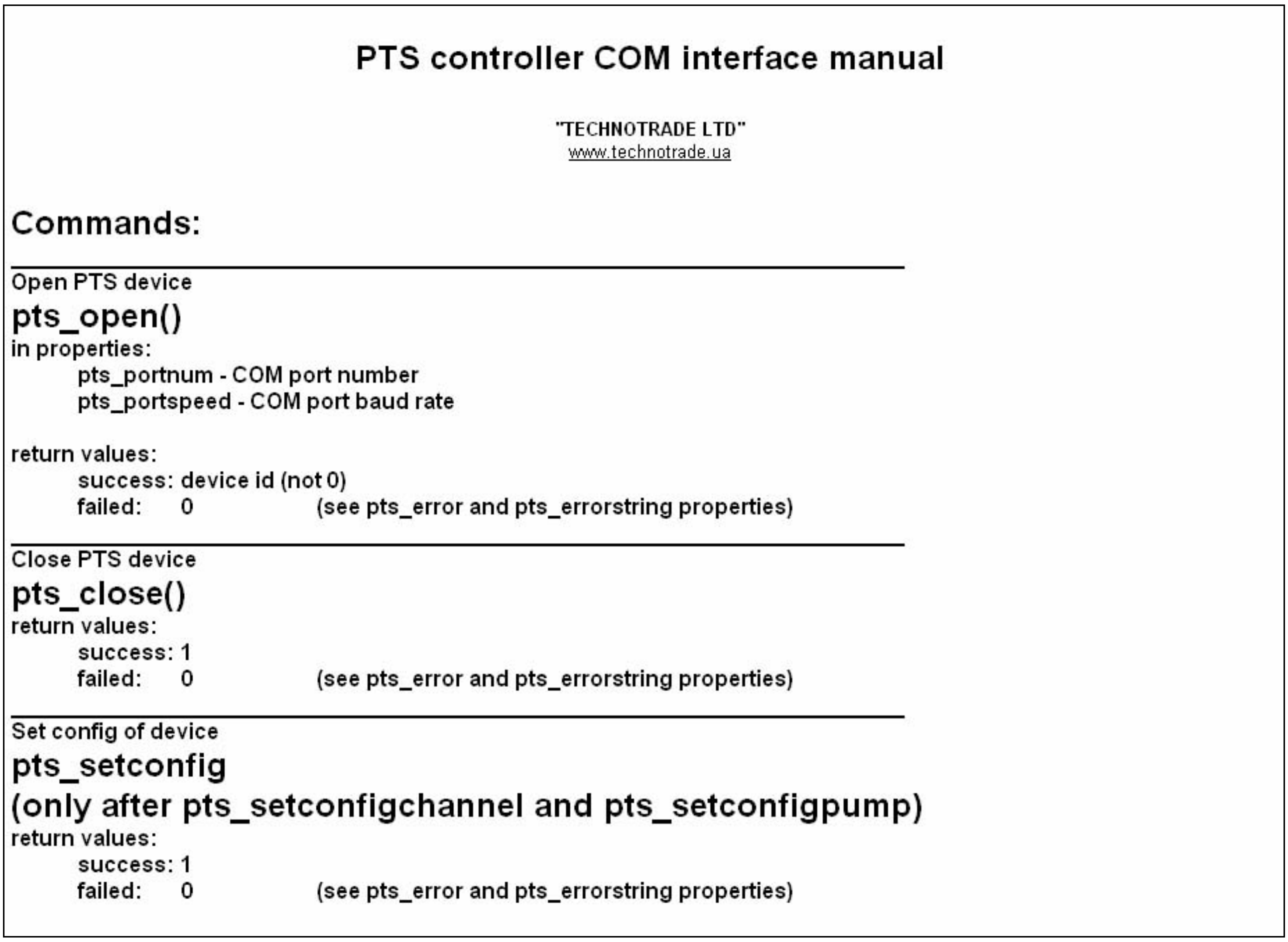
PTS controller COM-object for MS Windows OS

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).

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)

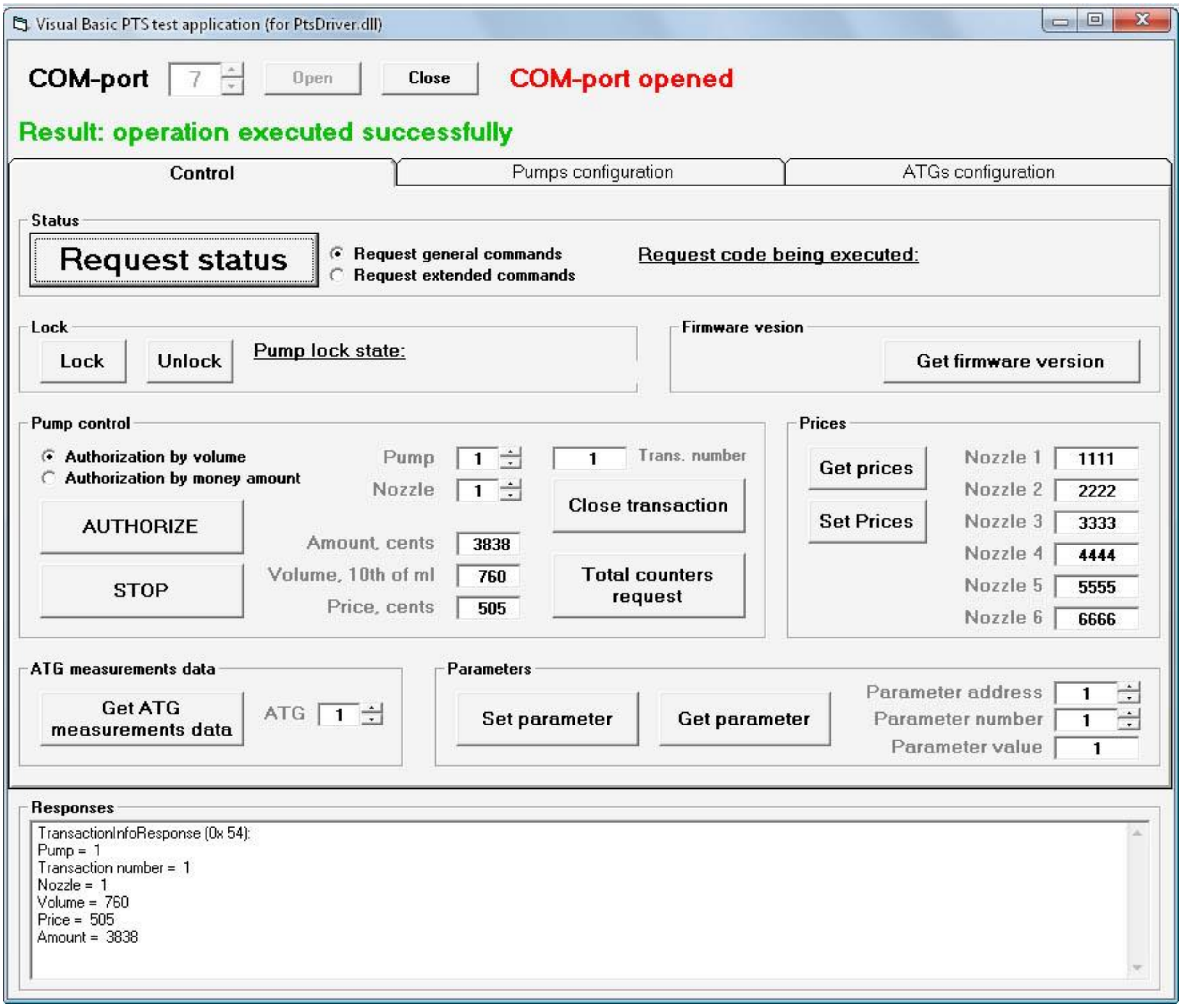
To register PTSDriver.dll library in Windows OS perform command

```
REGSVR32 PTSDriver.dll
```

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

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).



Visual Basic PTS test application (for PtsDriver.dll)

COM-port 7

COM-port opened

Result: operation executed successfully

Status: ● Request general commands ○ Request extended commands

Request code being executed:

Pump lock state:

Pump control: ● 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

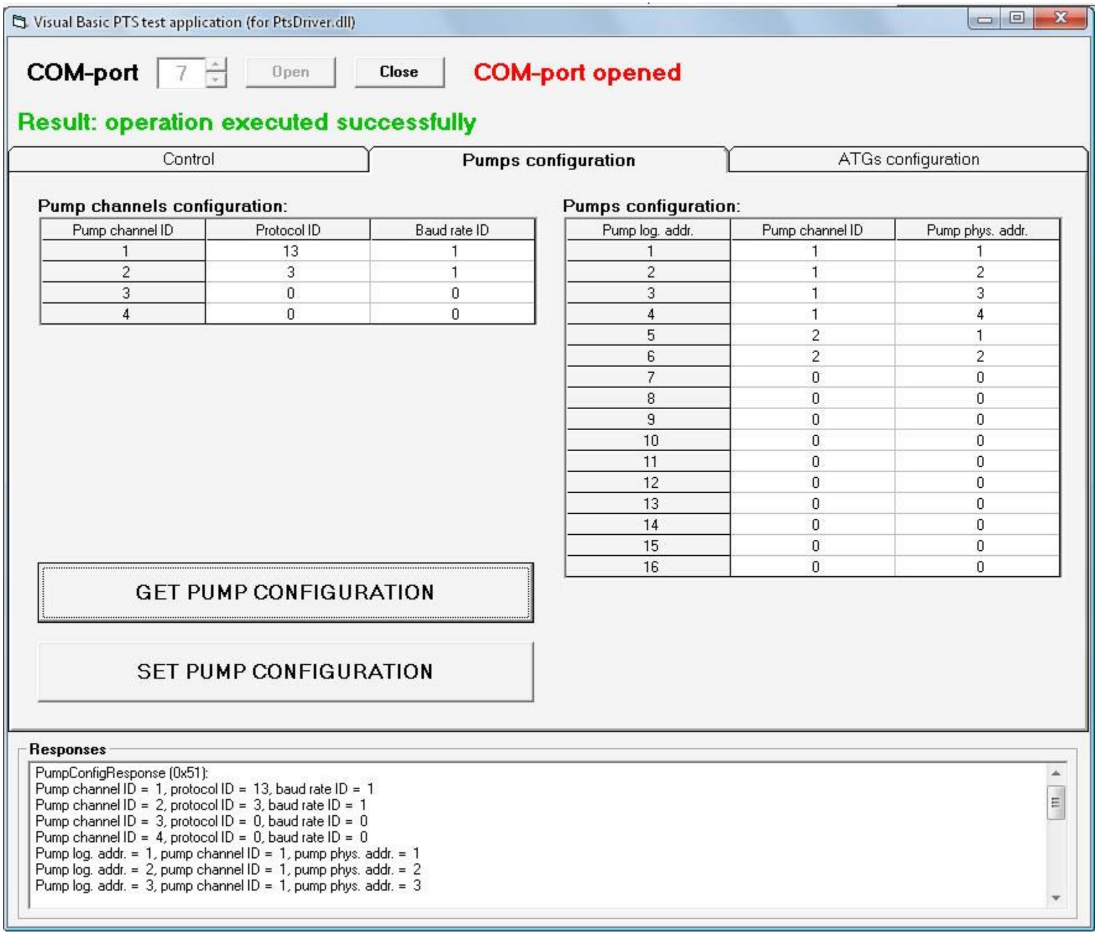
Parameter number 1

Parameter value 1

Responses

```
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



COM-port 7

COM-port opened

Result: operation executed successfully

Pump channels configuration:

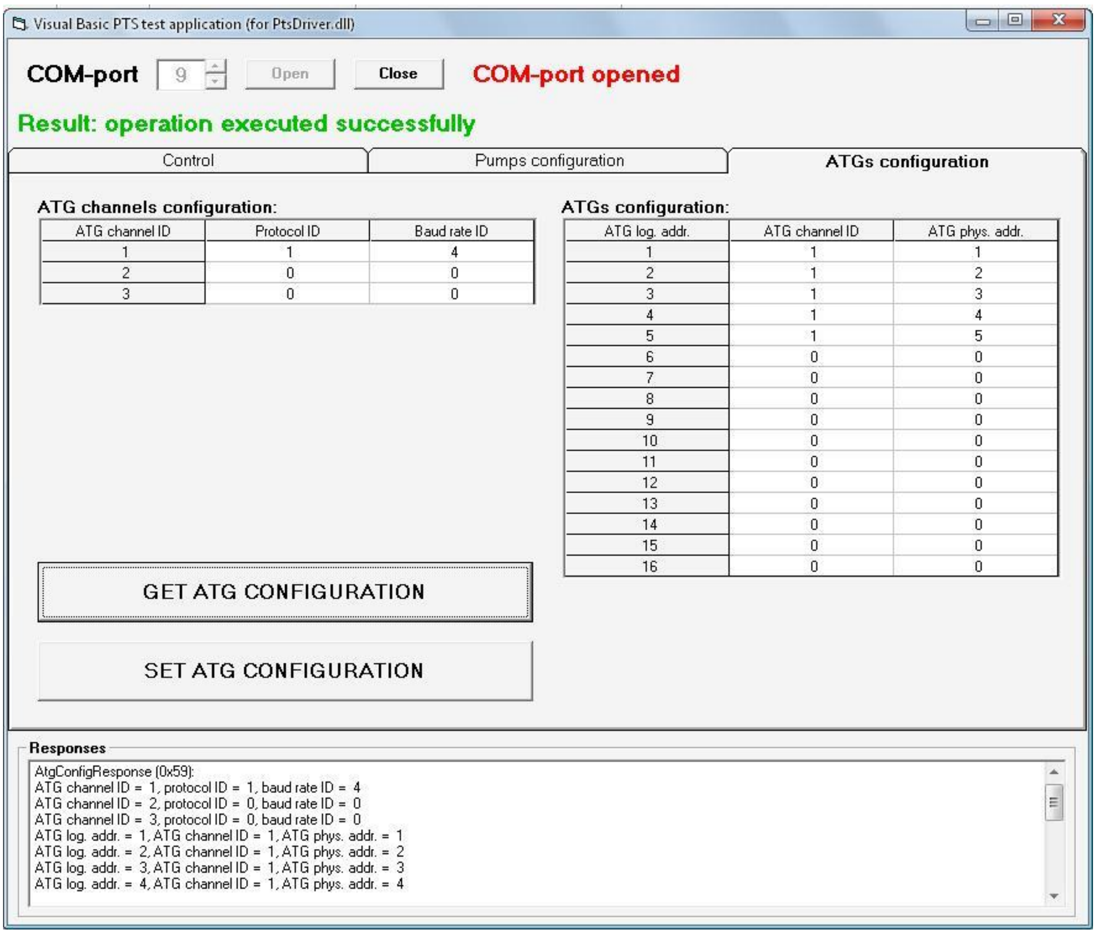
Pump channel ID	Protocol ID	Baud rate ID
1	13	1
2	3	1
3	0	0
4	0	0

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, 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



COM-port 9

COM-port opened

Result: operation executed 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

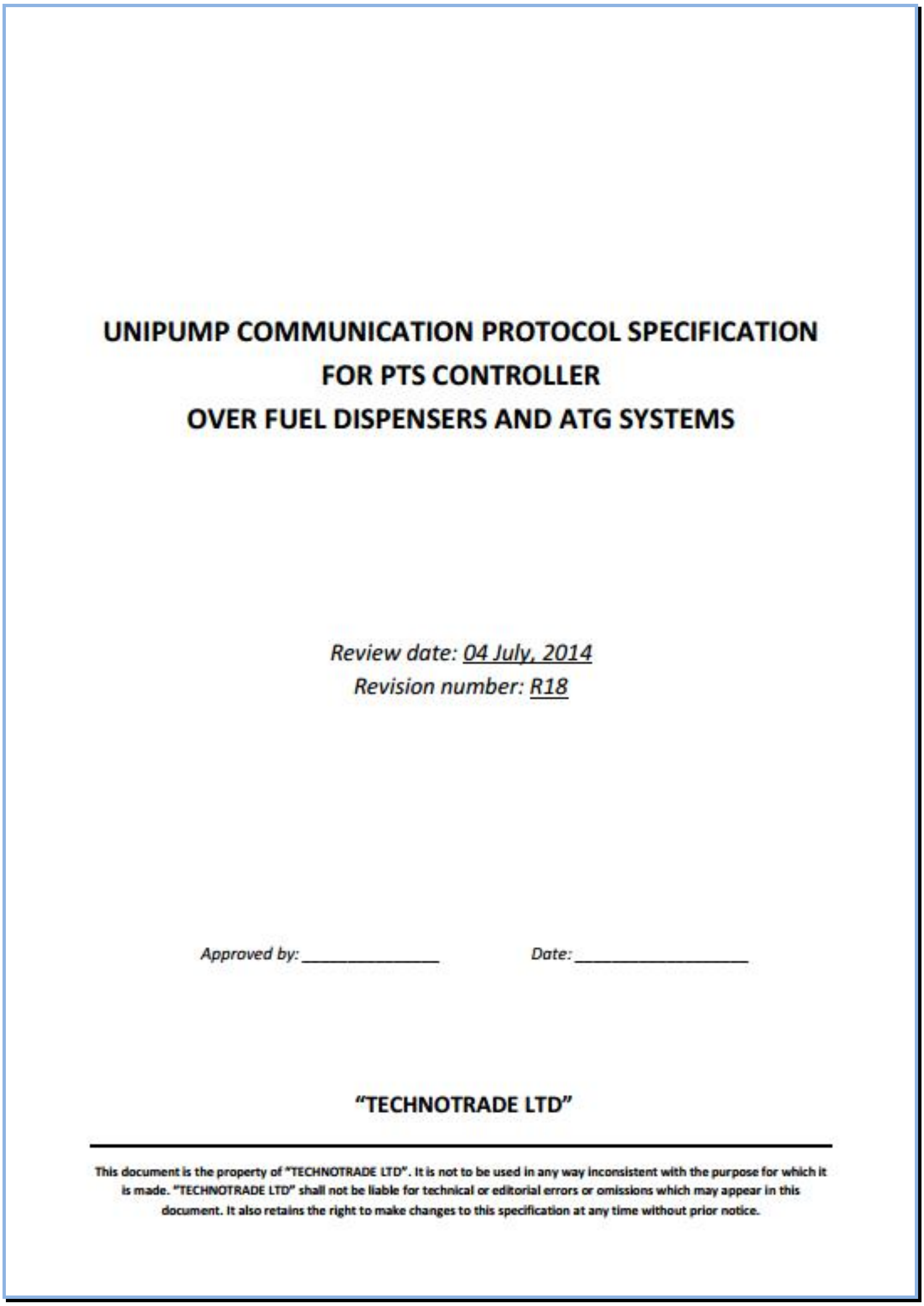
```
AtgConfigResponse (0x59):
ATG channel ID = 1, protocol ID = 1, baud rate ID = 4
ATG channel ID = 2, protocol ID = 0, baud rate ID = 0
ATG channel ID = 3, protocol ID = 0, baud rate ID = 0
ATG log. addr. = 1, ATG channel ID = 1, ATG phys. addr. = 1
ATG log. addr. = 2, ATG channel ID = 1, ATG phys.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

PTS controller is intended to be used in connection with control systems for petrol stations (POS systems, cash registers, OPT terminals, etc) to provide simultaneous control over various types of electronic fuel delivery dispensers and ATG systems of various manufactures using the single common communication protocol UniPump. PTS controller provides conversion of the common communication protocol UniPump into various proprietary communication protocols of manufacturers.

UniPump communication protocol covers a list of commands and responses for communication with PTS controller for provision of control over dispensers and ATG systems and configuration of PTS controller.



UNIPUMP COMMUNICATION PROTOCOL SPECIFICATION
FOR PTS CONTROLLER
OVER FUEL DISPENSERS AND ATG SYSTEMS

Review date: 04 July, 2014
Revision number: R18

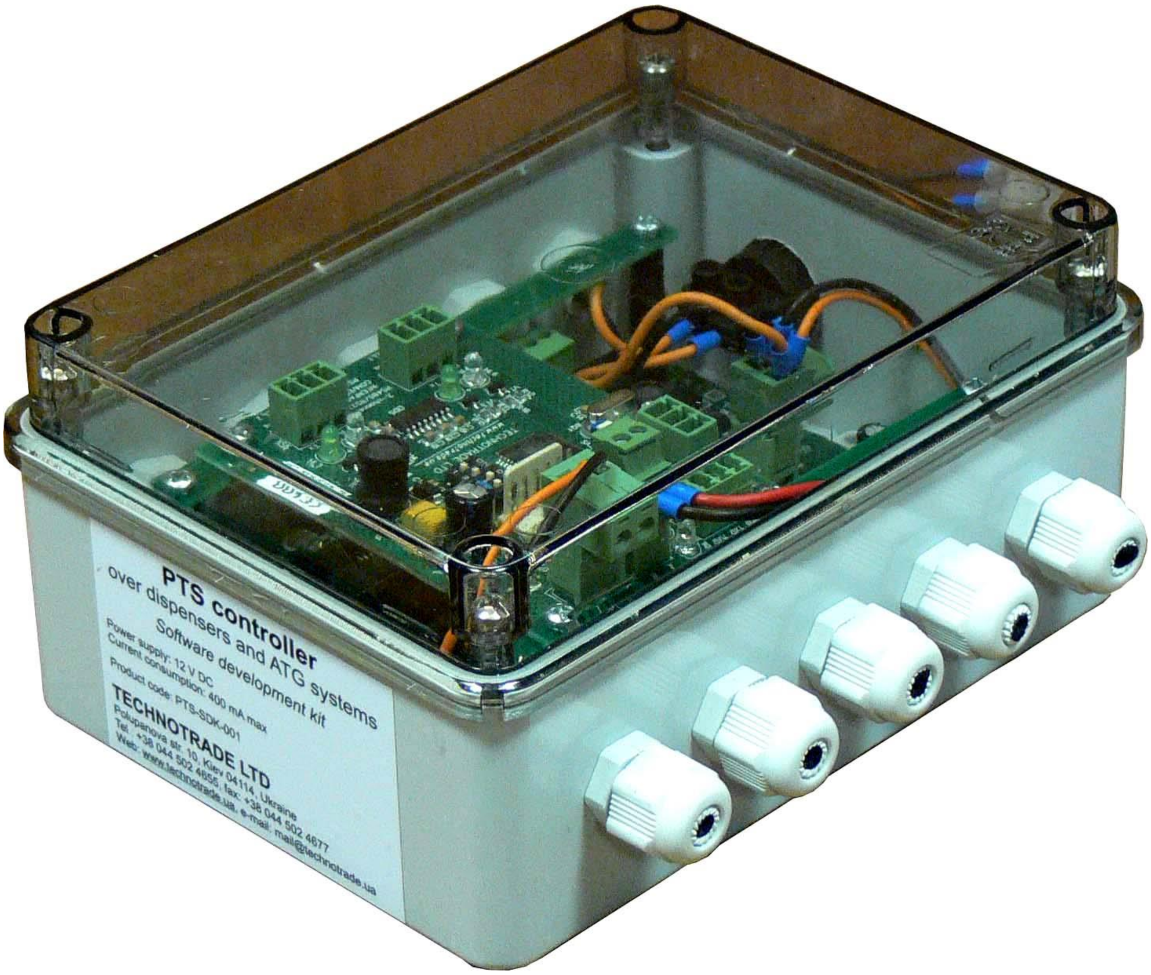
Approved by: ______________ *Date:* __________________

"TECHNOTRADE LTD"

This document is the property of "TECHNOTRADE LTD". It is not to be used in any way inconsistent with the purpose for which it is made. "TECHNOTRADE LTD" shall not be liable for technical or editorial errors or omissions which may appear in this document. It also retains the right to make changes to this specification at any time without prior notice.

PTS CONTROLLER SOFTWARE DEVELOPMENT KIT (SDK)

PTS controller SDK appointment


PTS controller
over dispensers and ATG systems
Software development kit
Power supply: 12 V DC
Current consumption: 400 mA max
Product code: PTS-SDK-001
TECHNOTRADE LTD
Polupanova str. 10, Kiev 04114, Ukraine
Tel: +38 044 502 4655, fax: +38 044 502 4677
Web: www.technotrade.ua, e-mail: mail@technotrade.ua

PTS controller SDK (Software Development Kit) is intended for developers of POS management software for petrol stations that want to control fuel dispensers and ATG systems from their software.

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 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.

4. 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.

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 (2 channel) – 1 pcs
3. USB software protection dongle Guardant (required for NaftaPOS software and fuel dispensers software simulators) – 1 pcs
4. Cabling (RS-232 to PC) – 3 pcs

SOFTWARE:

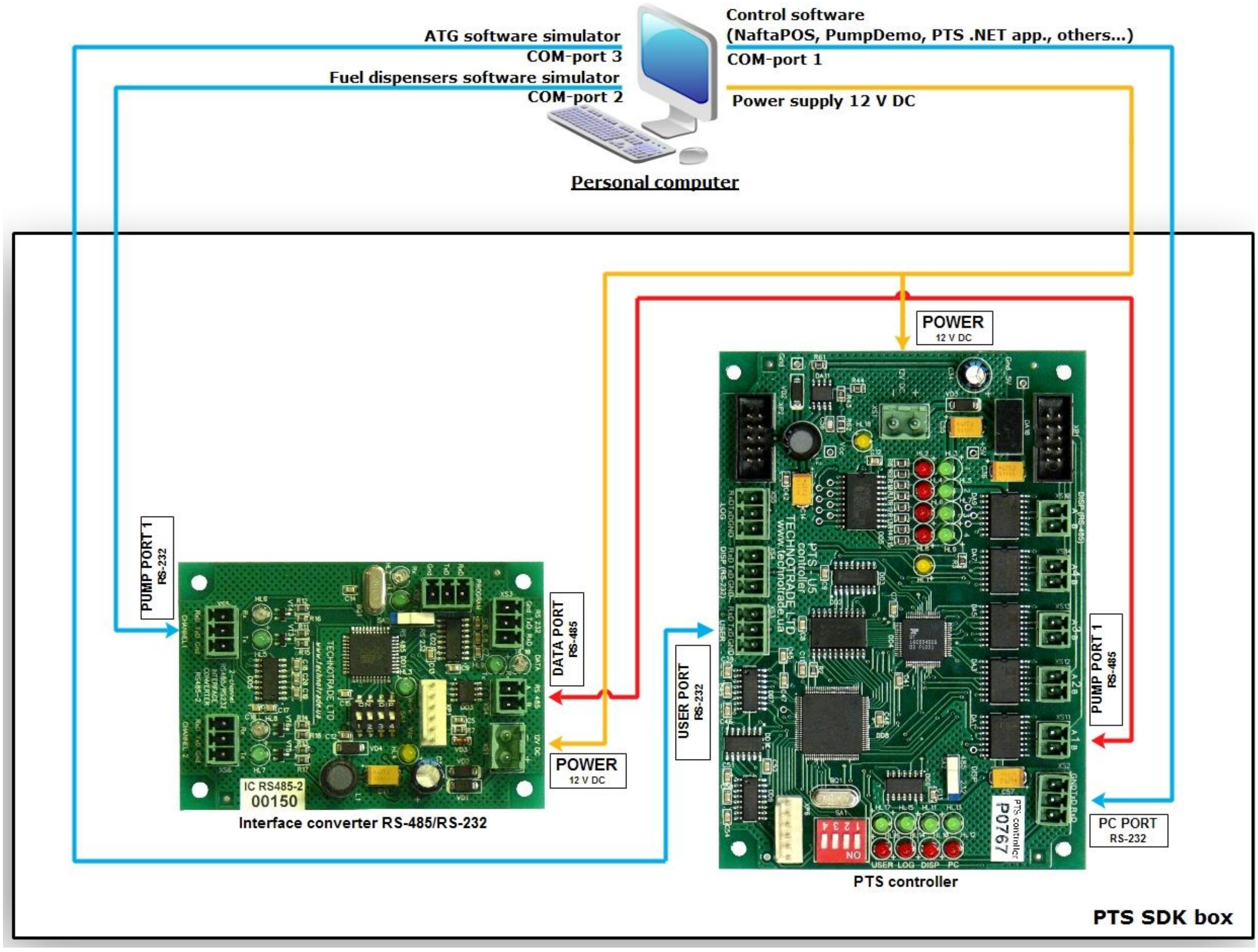
1. Software simulators of popular fuel dispensers
2. Software simulators of ATG systems (probes)
3. API for developers in various programming languages
4. NaftaPOS software for petrol stations – 1 license (protected by USB-dongle)
5. Useful utilities and applications for debugging of communication with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

PTS controller SDK technical features

Specification

##	PARAMETER	VALUE
1	**Voltage**	12 V DC
2	**Current consumption**	450 mA max
3	**Temperature range**	-40°C ÷ +80°C
4	**Weight**	600 g
5	**Overall dimensions**	160 x 160 x 85 mm

PTS controller SDK connections scheme



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

Control systems software, which uses UniPump communication protocol, connects to the PTS controller PC port.

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 personal computer and communicates through a COM-port 2.

Connection to ATG software simulators is made through USER port of PTS controller (RS-232 interface), which is connected to personal computer through a COM-port 3.

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.

2-channel RS-485 to RS-232 dispenser interface converter

Appointment

2-channel RS-485 to RS-232 dispenser interface converter (RS-485 to RS-232 interface and backwards) is intended for communication with fuel dispensers, which use RS-232 interface, through interfaces:

- RS-232
- RS-485 (2-wire)

More technical information regarding 2-channel RS-485 to RS-232 dispenser interface converter can be found in technical guide on: *http://www.technotrade.ua/rs485_to_rs232_converter.html*.

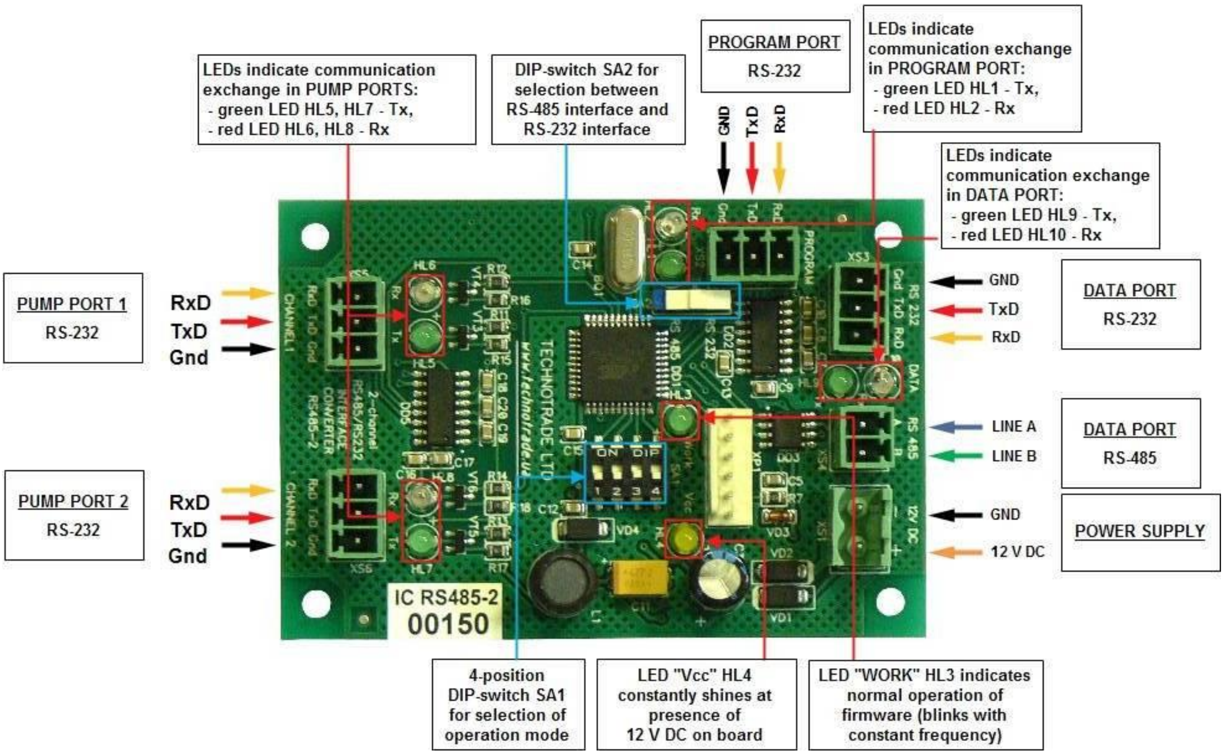
Specification

##	PARAMETER	VALUE
1	**Voltage**	12 V DC
2	**Current consumption**	75 mA max
3	**Temperature range**	-40°C ÷ +80°C
4	**Weight**	45 g
5	**Overall dimensions**	85 x58 x 25 mm

Communication ports

PORT NAME		INTERFACE	PURPOSE
DATA PORTS	**RS-232**	RS-232 (3 wires: TxD, RxD, Gnd)	Communication with a control system (POS terminal, cash register, OPT). Selection of interface is made using DIP-switch SA2
	RS-485	RS-485 (2-wires: line A, line B)	
PUMP PORTS	**Pump port 1**	RS-232 (3 wires: TxD, RxD, Gnd)	Connection with fuel dispensers using RS-232 interface. One dispenser is to be connected to each of the pump ports.
	Pump port 2		
PROGRAM PORT	**RS-232**	RS-232 (3 wires: TxD, RxD, Gnd)	Update of the interface converter firmware

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



LEDs indicate communication exchange in PUMP PORTS:
- green LED HL5, HL7 - Tx,
- red LED HL6, HL8 - Rx

DIP-switch SA2 for selection between RS-485 interface and RS-232 interface

LEDs indicate communication exchange in PROGRAM PORT:
- green LED HL1 - Tx,
- red LED HL2 - Rx

LEDs indicate communication exchange in DATA PORT:
- green LED HL9 - Tx,
- red LED HL10 - Rx

4-position DIP-switch SA1 for selection of operation mode

LED "Vcc" HL4 constantly shines at presence of 12 V DC on board

LED "WORK" HL3 indicates normal operation of firmware (blinks with constant frequency)

NOTE!

DIP-switch SA1 serves for selection of interface converter operation mode:

- switch 1 should be set in position "OFF"
- switch 2 should be set in position "OFF"
- switch 3 should be set in position "OFF"
- switch 4 should be set in position "OFF"

DIP-switch SA2 serves for selection of interface:

- RS-232
- RS-485 (2-wire)

USB software protection dongle Guardant

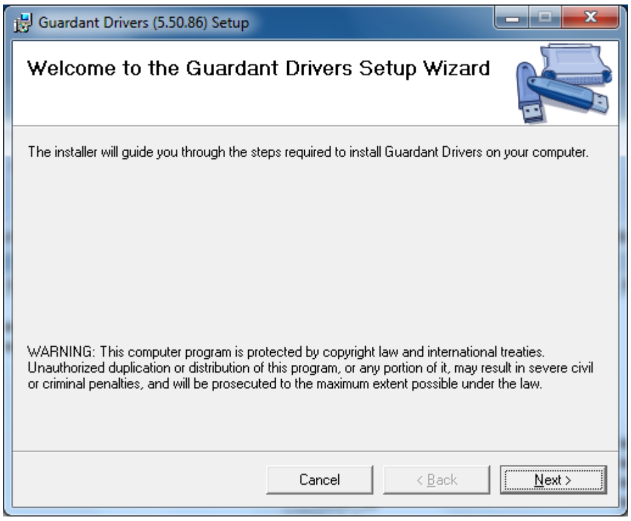
Appointment

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.

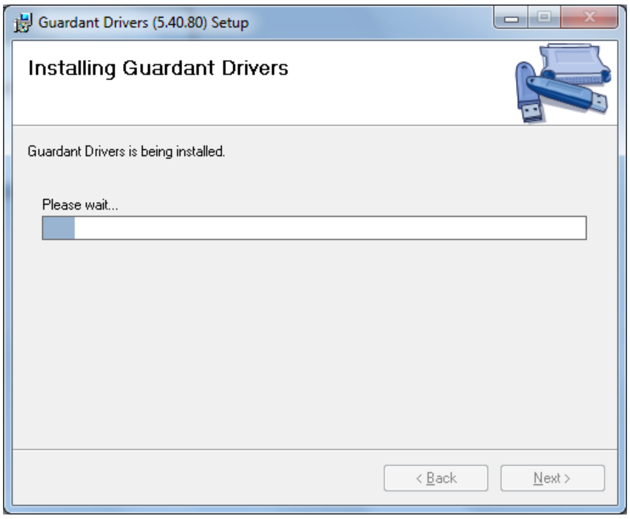
Guardant USB dongle drivers are installed during NaftaPOS software installation process or can be installed/updated any time later:


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...

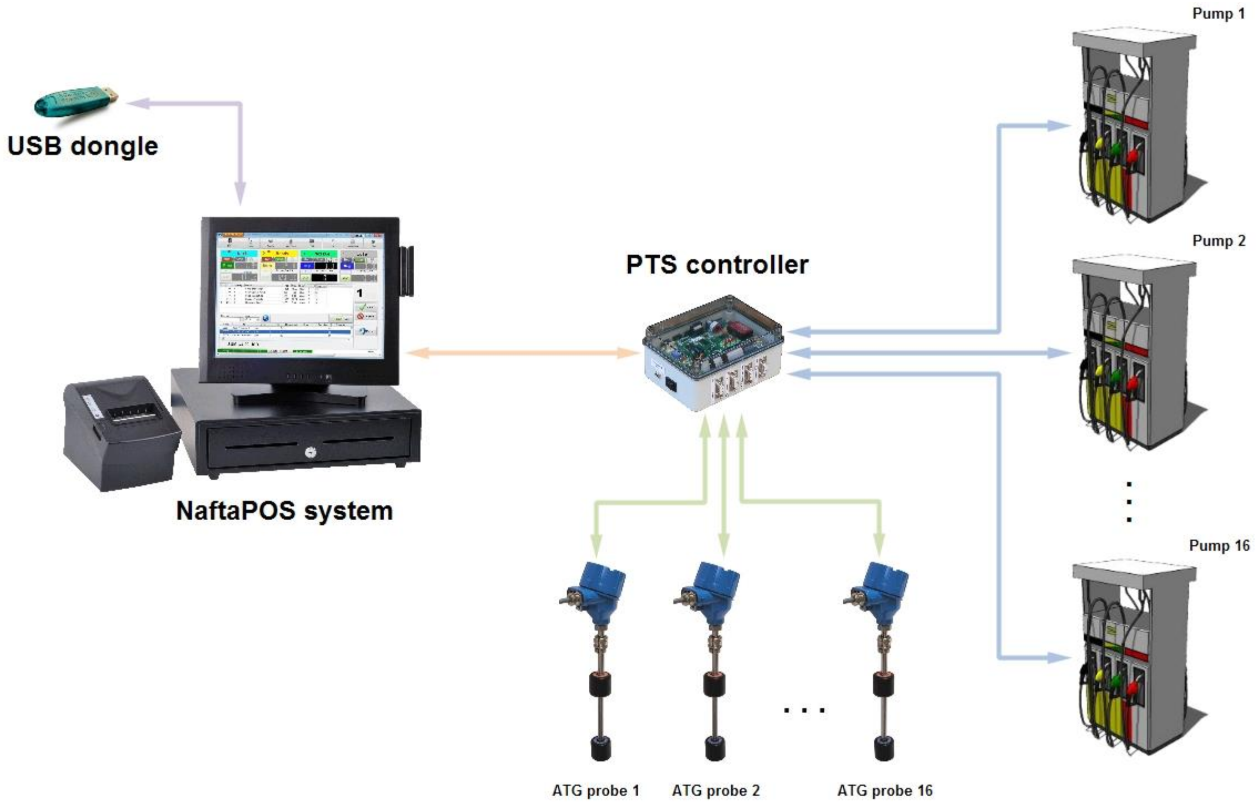
During installation process it is necessary to follow all steps in the installation wizard until installation is completed.

NaftaPOS software for petrol stations

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.

Information on installation, configuration and operation in NaftaPOS software can be found in NaftaPOS software user manual, which can be downloaded from NaftaPOS software web-page: *http://www.technotrade.ua/petrol_station_software.html*.

Description on running the NaftaPOS system in structure of PTS controller SDK is given in section "*PTS SDK step by step configuration*" below.



Open source PTS controller .NET application (C# and VB.NET)

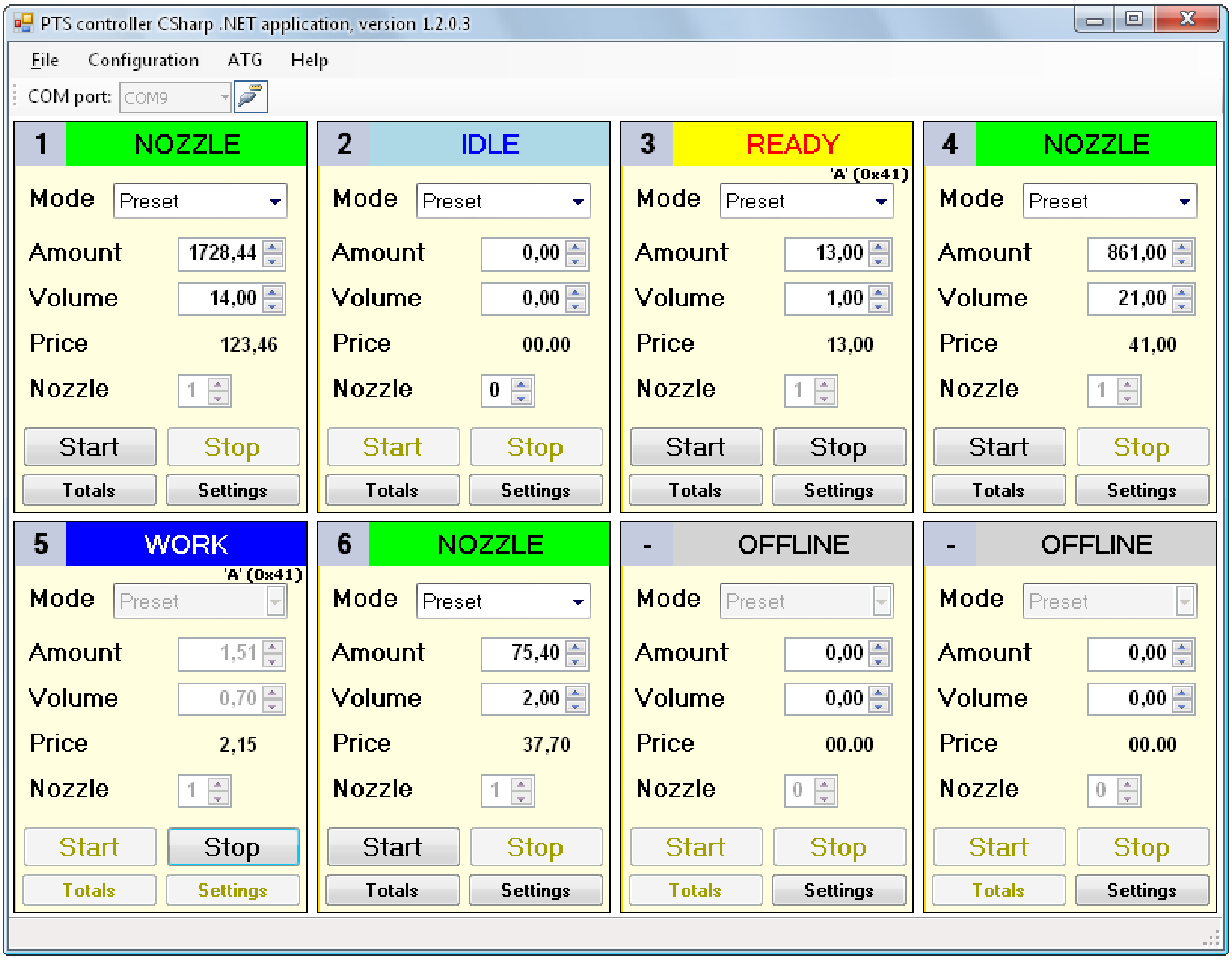
Purpose

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.



COM port: COM9

	1 NOZZLE	2 IDLE	3 READY	4 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 WORK	6 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.

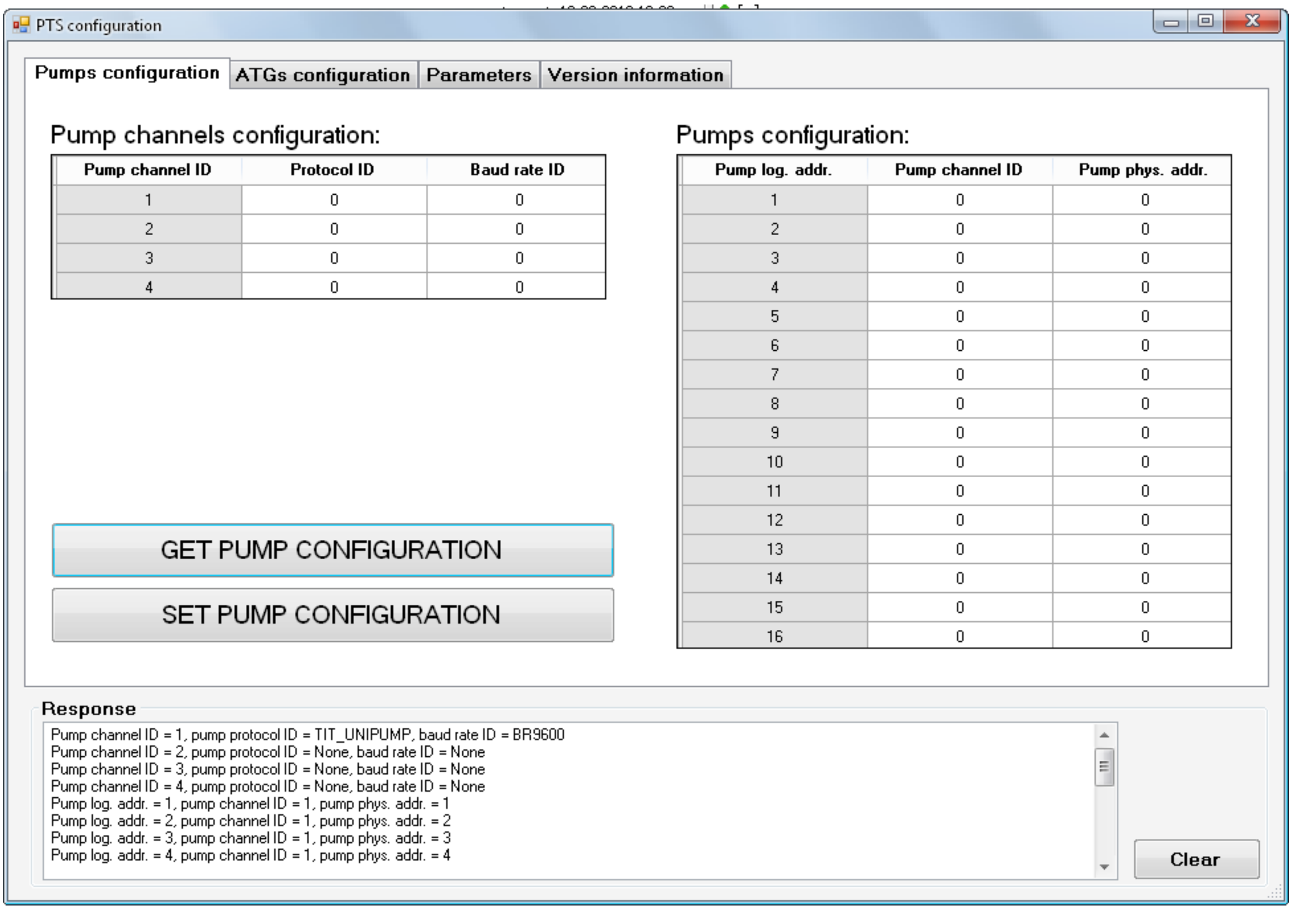
Note: connection with PTS controller is made through a 3-wire realization of RS-232 interface, an asynchronous half-duplex channel, baud rate 57600 baud without a parity check. This configuration is hardly coded inside a program.

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

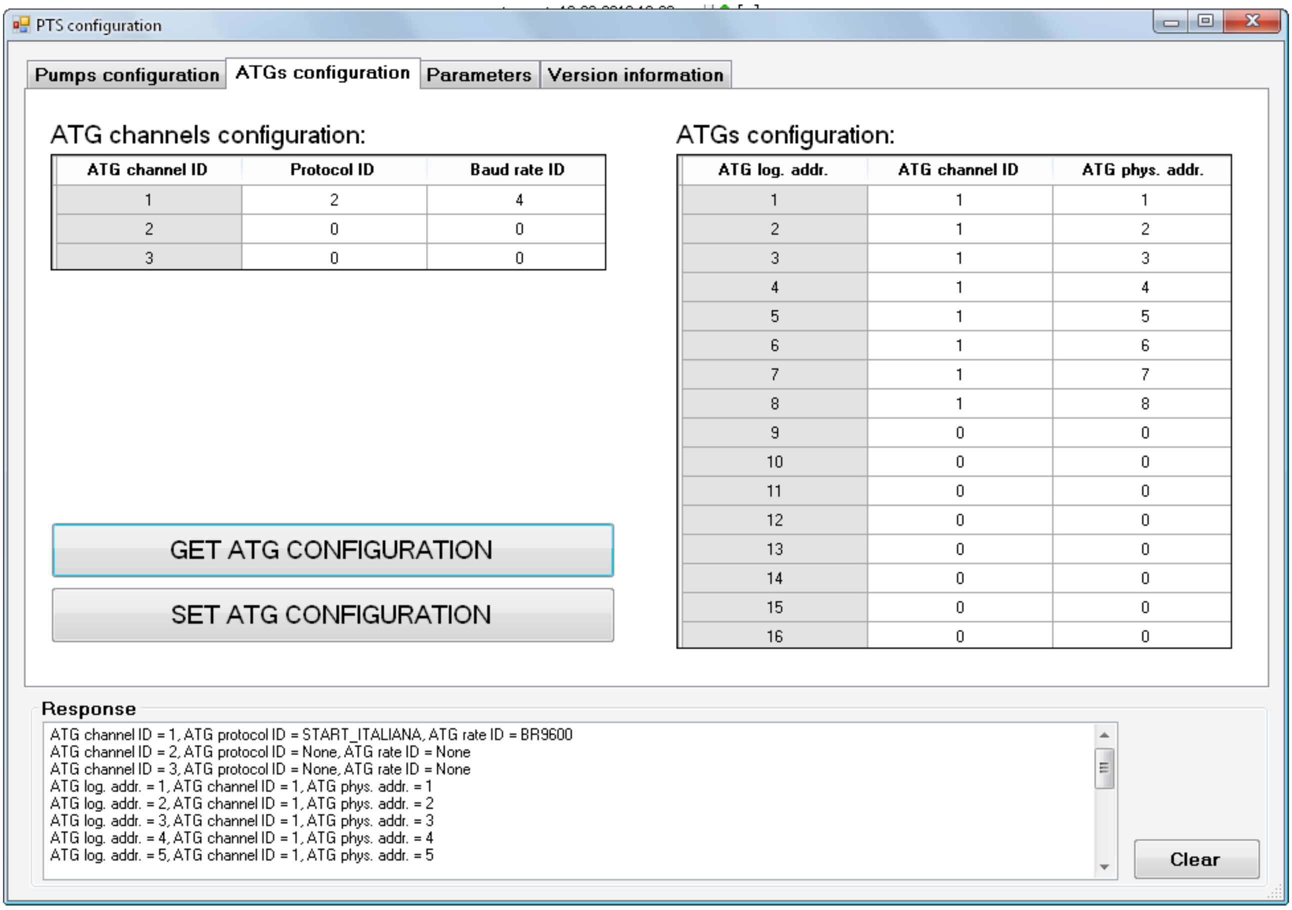
Pumps configuration:

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

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
```

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



ATG channels configuration:

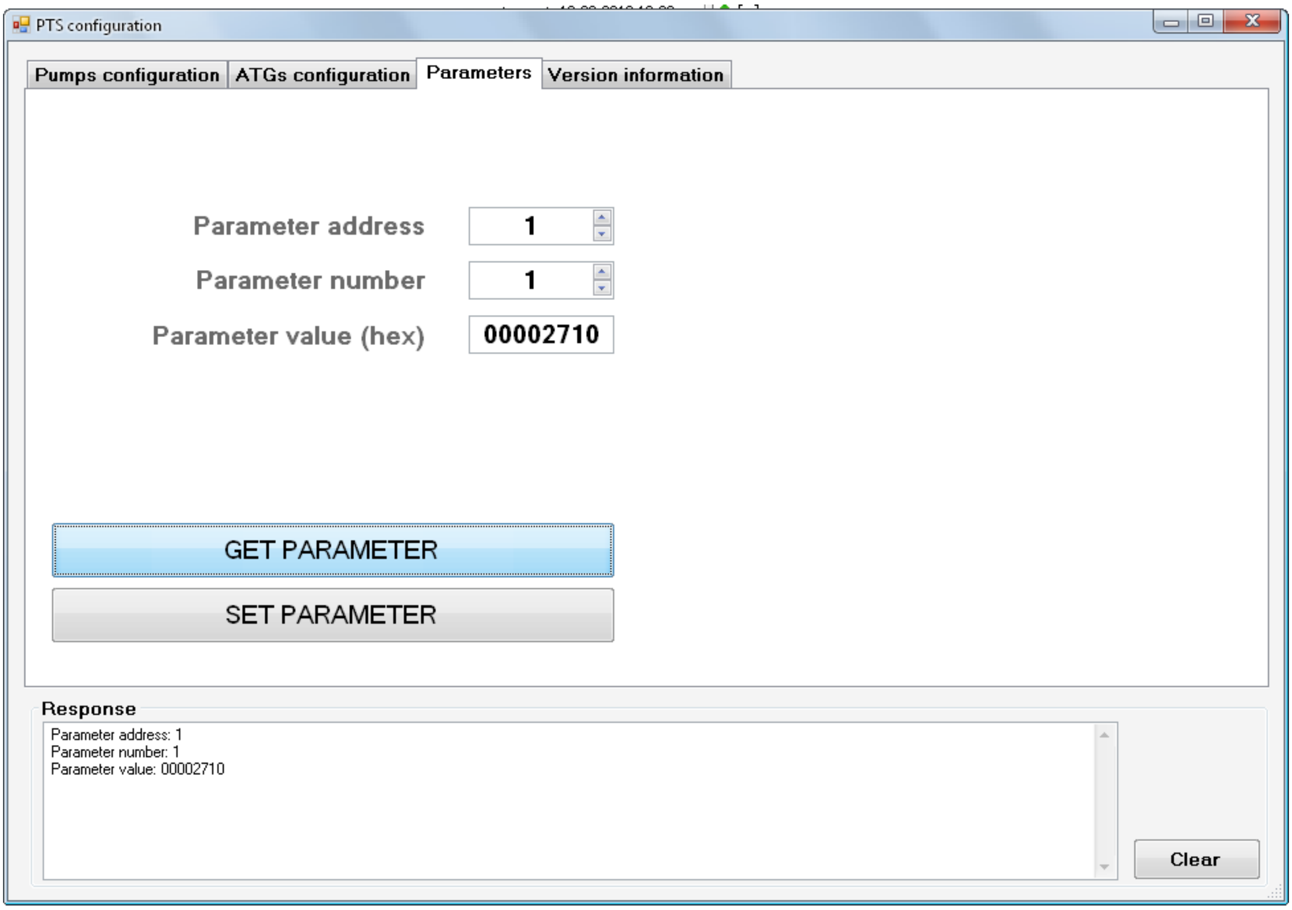
ATG channel ID	Protocol ID	Baud rate ID
1	2	4
2	0	0
3	0	0

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

```
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):



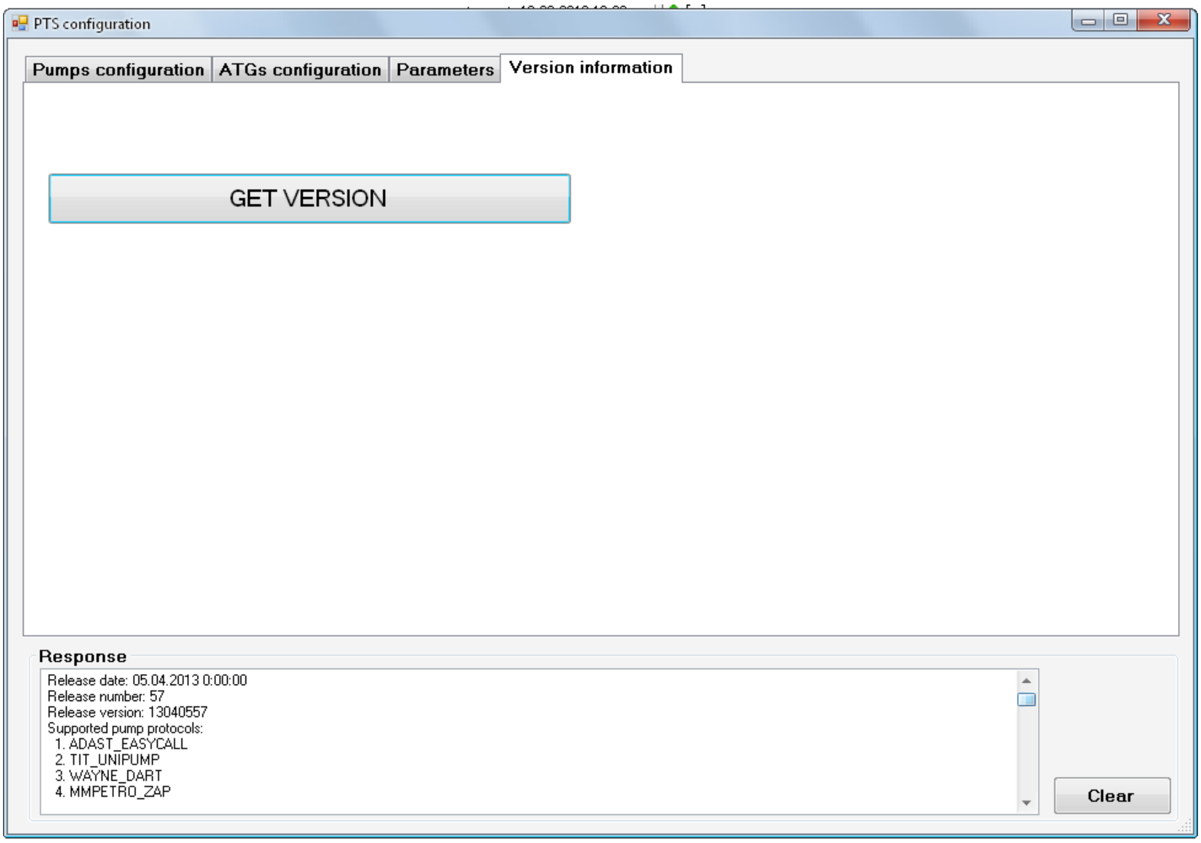
Parameter address: 1

Parameter number: 1

Parameter value (hex): 00002710

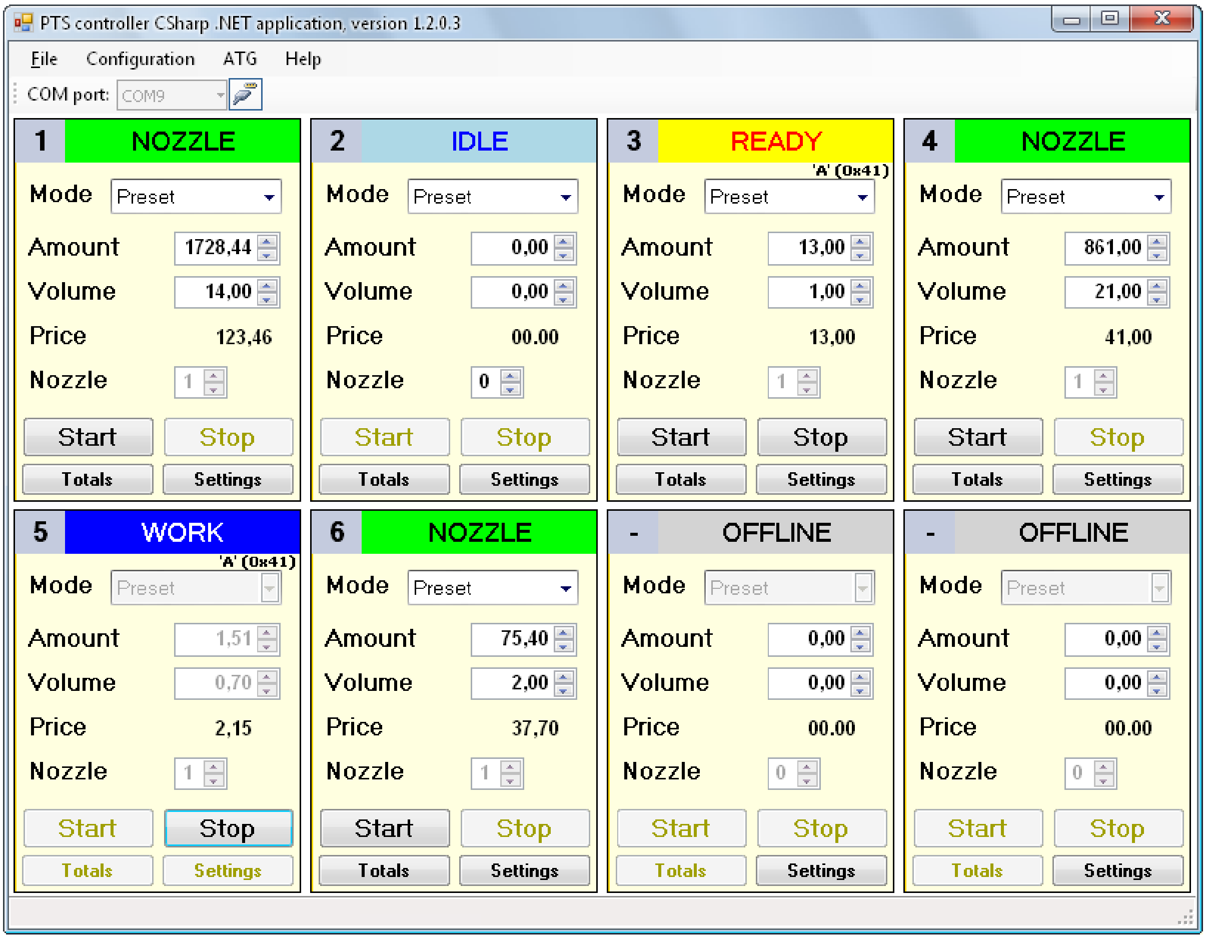
```
Parameter address: 1
Parameter number: 1
Parameter value: 00002710
```

Tab "*Version information*" allows to receive 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



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

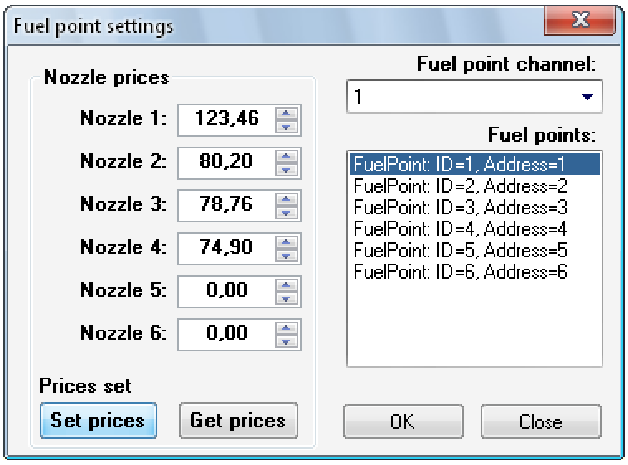
Possible statuses of fuel points:

- **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.

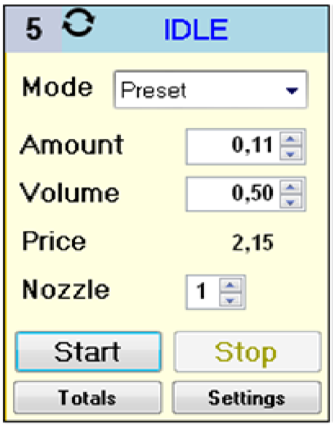
Clicking "*Set prices*" button leads to updating prices on selected fuel point. Clicking "*Get prices*" leads to receiving of prices from selected fuel point.

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.

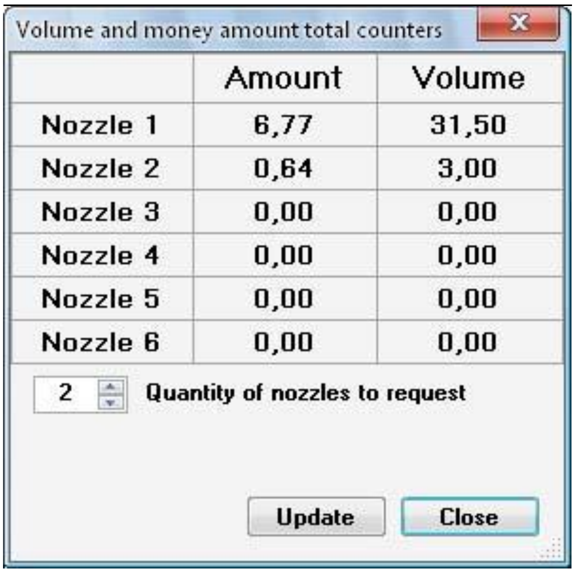
Reading of fuel point total counters

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

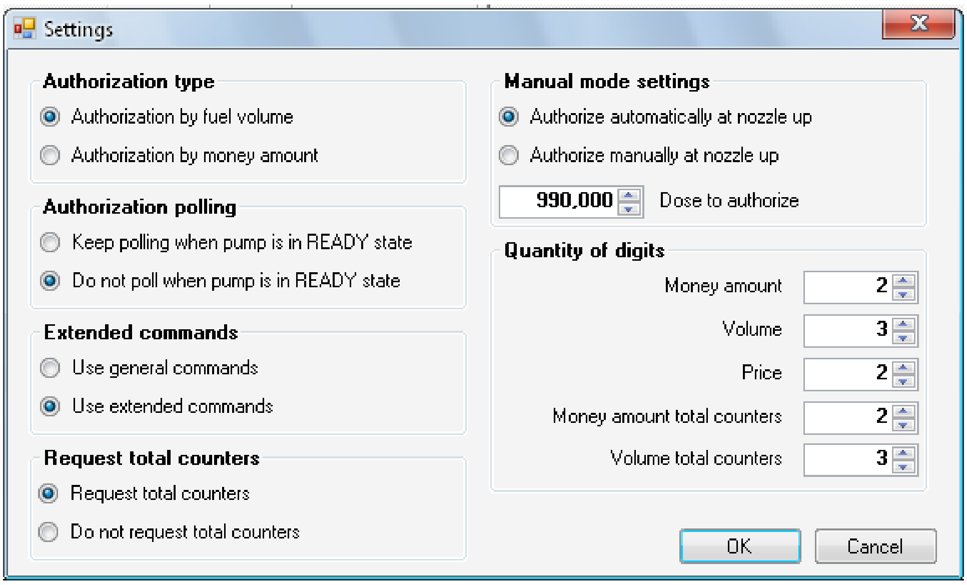

Volume and money amount total counters

	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

Quantity of nozzles to request: 2

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:

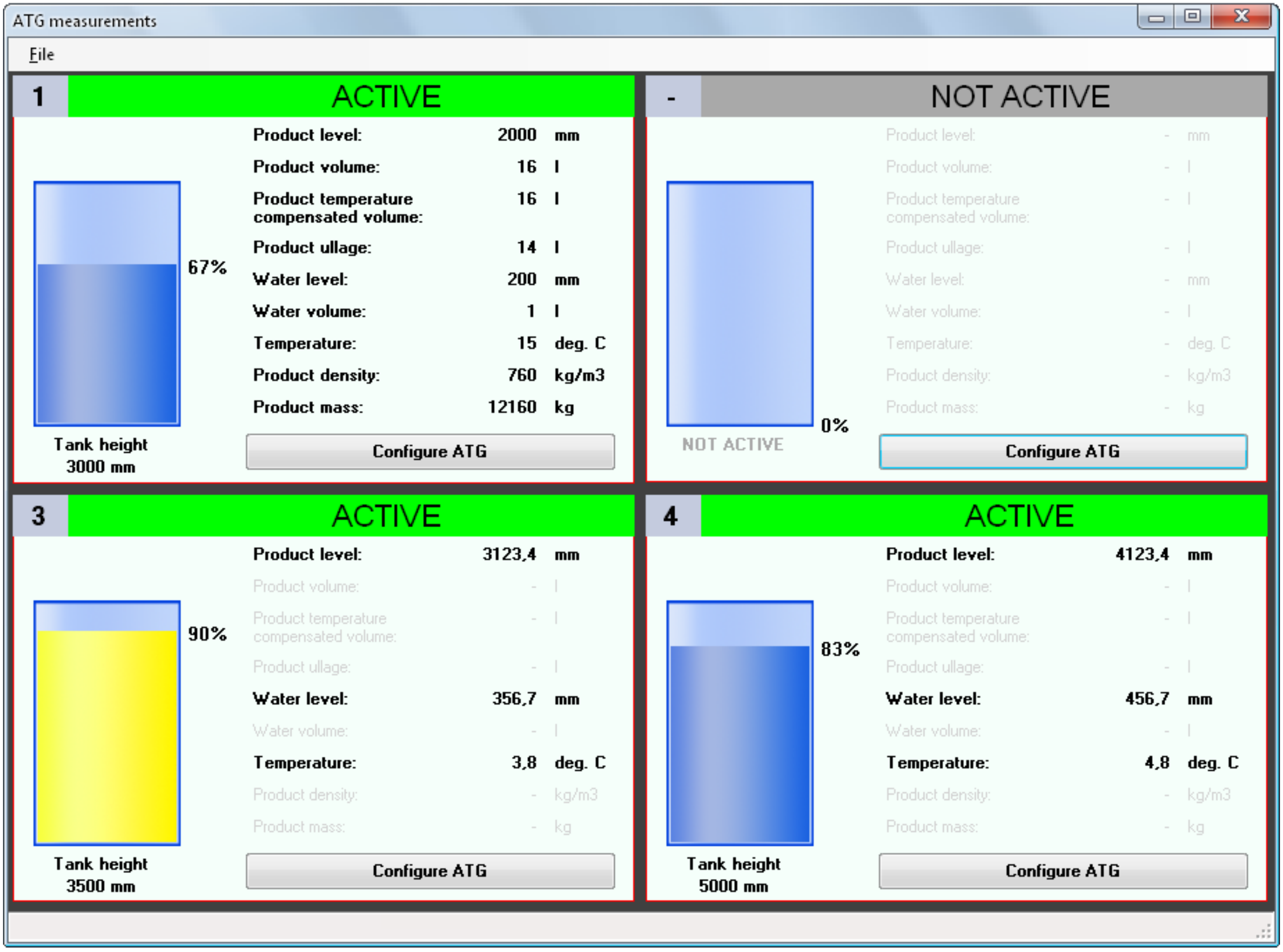
   Setting of quantity of digits in money, volume, price and total counters (volume and money amount) - this option is required in cases when dispensers may have non-standard quantity of digits (for example 3 digits in volume and 0 digits in money amount)

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

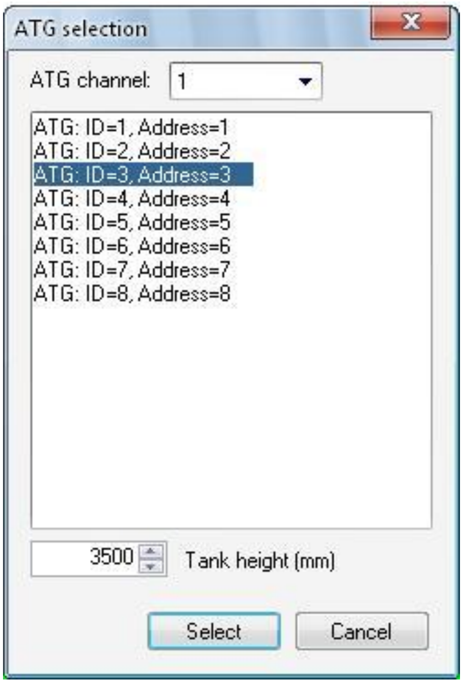
83%
Tank height
5000 mm

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).



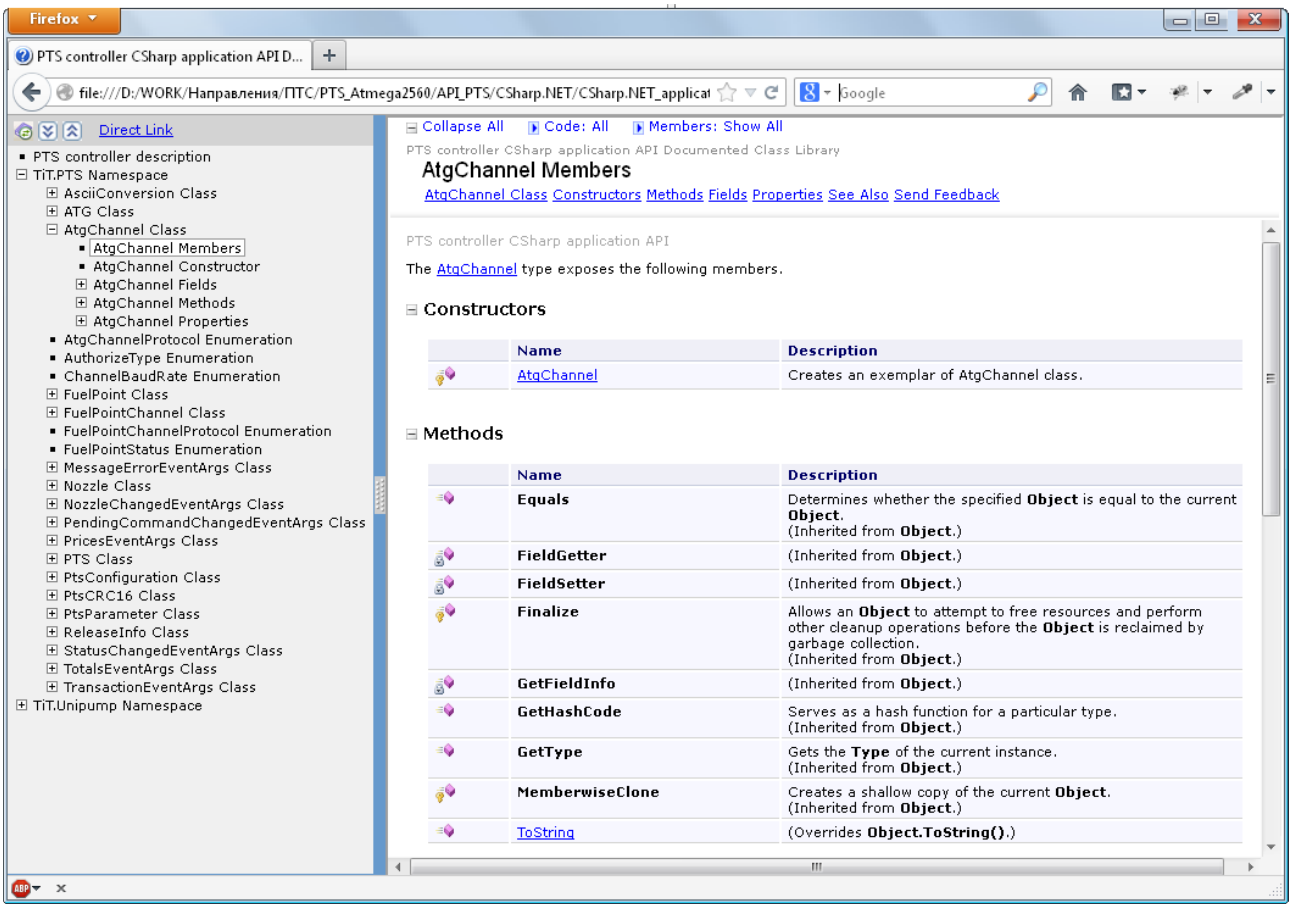
ATG channel: 1

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.

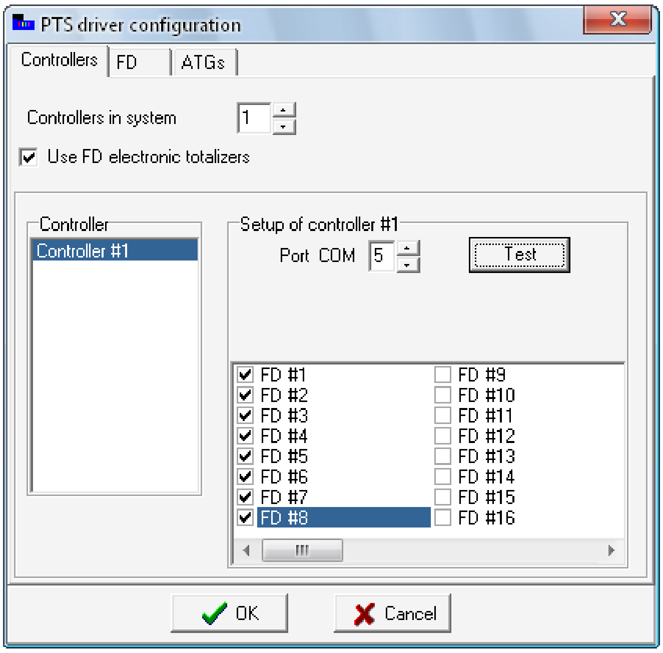
Methods

	Name	Description
	Equals	Determines whether the specified **Object** is equal to the current **Object**. (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. (Inherited from **Object**.)
	GetFieldInfo	(Inherited from **Object**.)
	GetHashCode	Serves as a hash function for a particular type. (Inherited from **Object**.)
	GetType	Gets the **Type** of the current instance. (Inherited from **Object**.)
	MemberwiseClone	Creates a shallow copy of the current **Object**. (Inherited from **Object**.)
	ToString	(Overrides **Object.ToString()**.)

PumpDemo utility

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

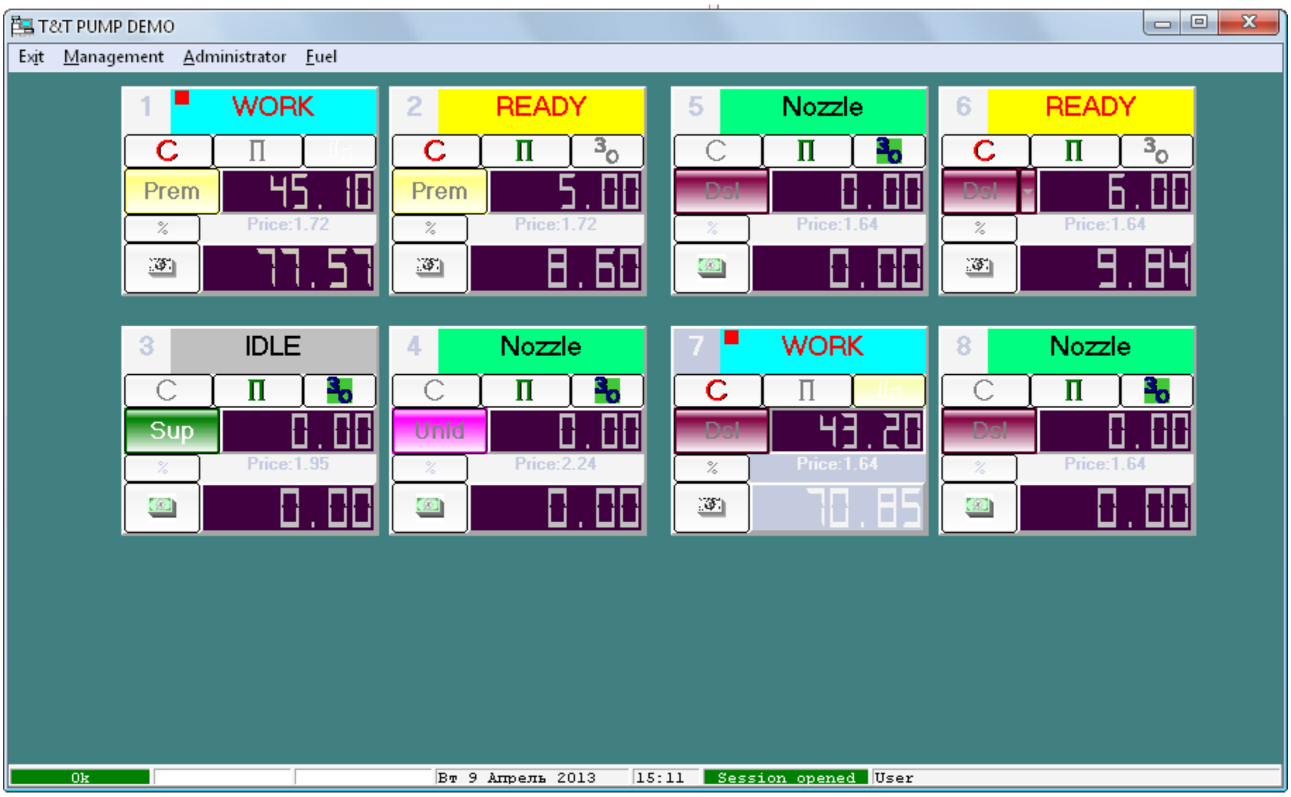
Configuration of the system is done by running a setup.bat file. Configuration is similar to configuration of NaftaPOS software for operation with PTS controller (see NaftaPOS software user manual):



Controllers in system 1
☑ Use FD electronic totalizers
Controller: Controller #1
Setup of controller #1
Port COM 5

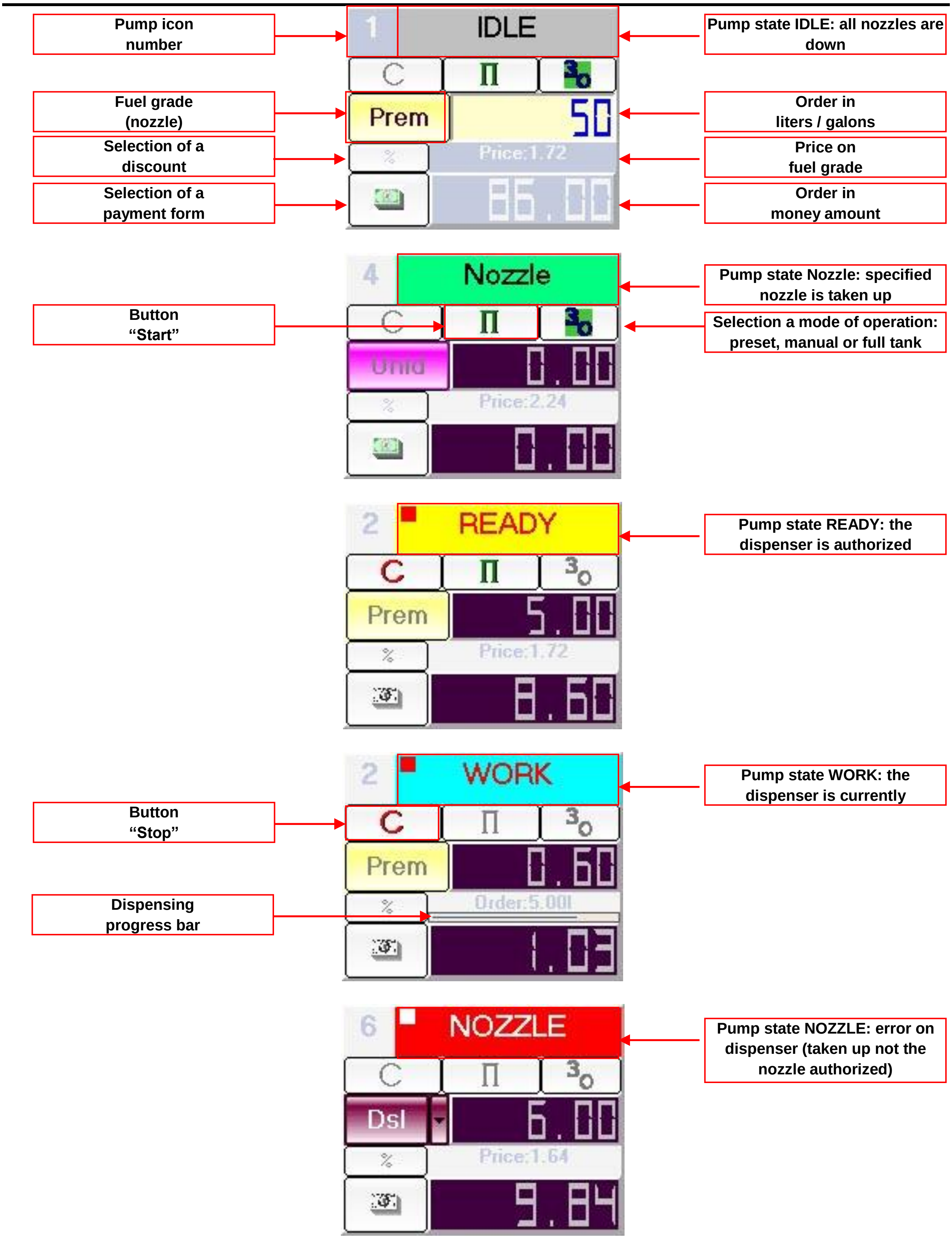
☑ FD #1 □ FD #9
☑ FD #2 □ FD #10
☑ FD #3 □ FD #11
☑ FD #4 □ FD #12
☑ FD #5 □ FD #13
☑ FD #6 □ FD #14
☑ FD #7 □ FD #15
☑ FD #8 □ FD #16

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.



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

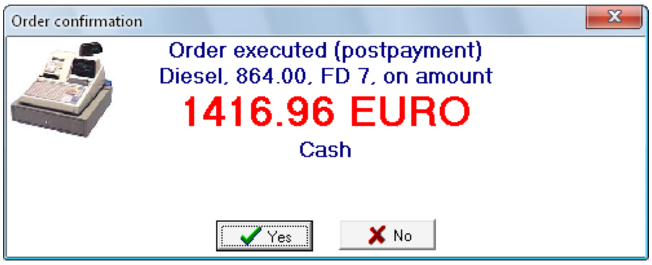
Pump state WORK: the dispenser is currently

Button “Stop”

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

PTS controller test utilities

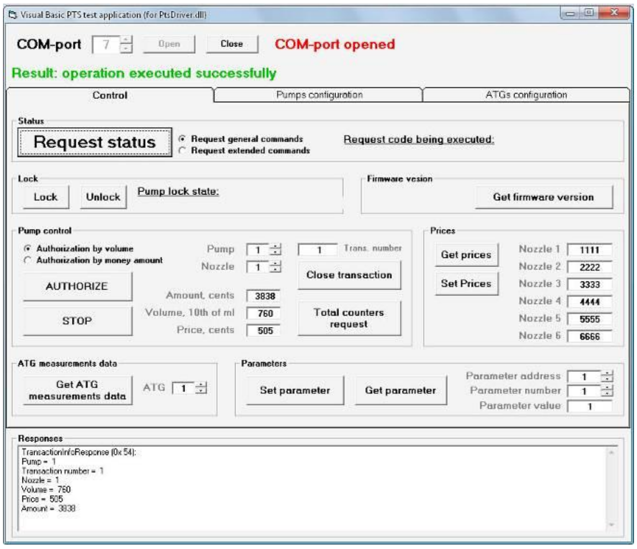
Purpose

PTS controller test utilities are written in the following languages:

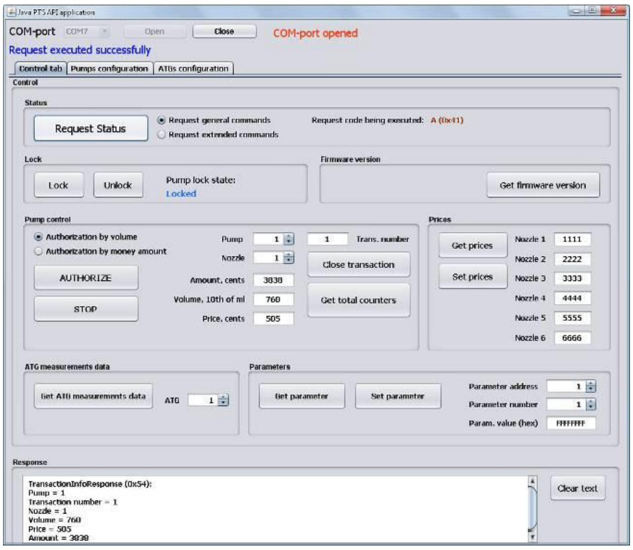
- C#
- VB 6.0
- VB.NET
- Java

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.

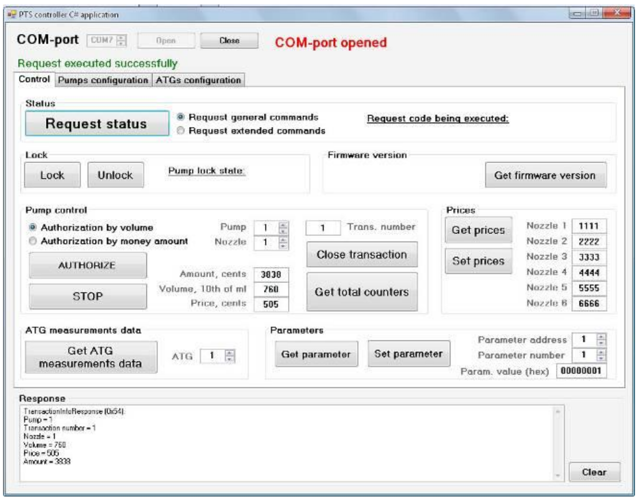
Utility provides all operations with a PTS controller. Thus developers using one of the above languages can view how control over PTS controller is provided in the utility and implement the same functions in own developed applications for petrol stations.

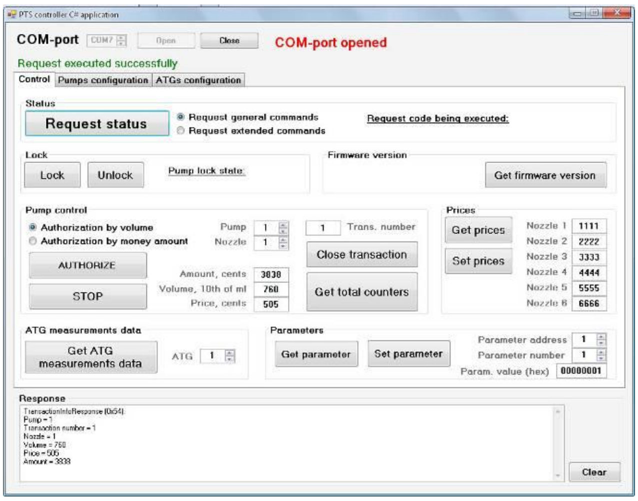

Visual Basic PTS test application (for PtsDriver.dll)
COM-port 7
COM-port opened
Result: operation executed successfully
● Request general commands
○ Request extended commands
Request code being executed:
Pump lock state:
● Authorization by volume
○ Authorization by money amount
Pump 1
Nozzle 1
Trans. 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
Responses
```
TransactionInfoResponse (0x54):
Pump = 1
Transaction number = 1
Nozzle = 1
Volume = 760
Price = 505
Amount = 3838
```

Visual Basic 6.0 (uses COM-object)


Java PTS API application
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
Response
```
TransactionInfoResponse (0x54):
Pump = 1
Transaction number = 1
Nozzle = 1
Volume = 760
Price = 505
Amount = 3838
```

Java (developed in NetBeans IDE)


PTS controller C# application
COM-port COM7
COM-port opened
Request executed successfully
● Request general commands
○ Request extended commands
Request code being executed:
Pump lock state:
● 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
Response
```
TransactionInfoResponse (0x54):
Pump = 1
Transaction number = 1
Nozzle = 1
Volume = 760
Price = 505
Amount = 3838
```


PTS controller C# application
COM-port COM7
COM-port opened
Request executed successfully
● Request general commands
○ Request extended commands
Request code being executed:
Pump lock state:
● 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
Response
```
TransactionInfoResponse (0x54):
Pump = 1
Transaction number = 1
Nozzle = 1
Volume = 760
Price = 505
Amount = 3838
```

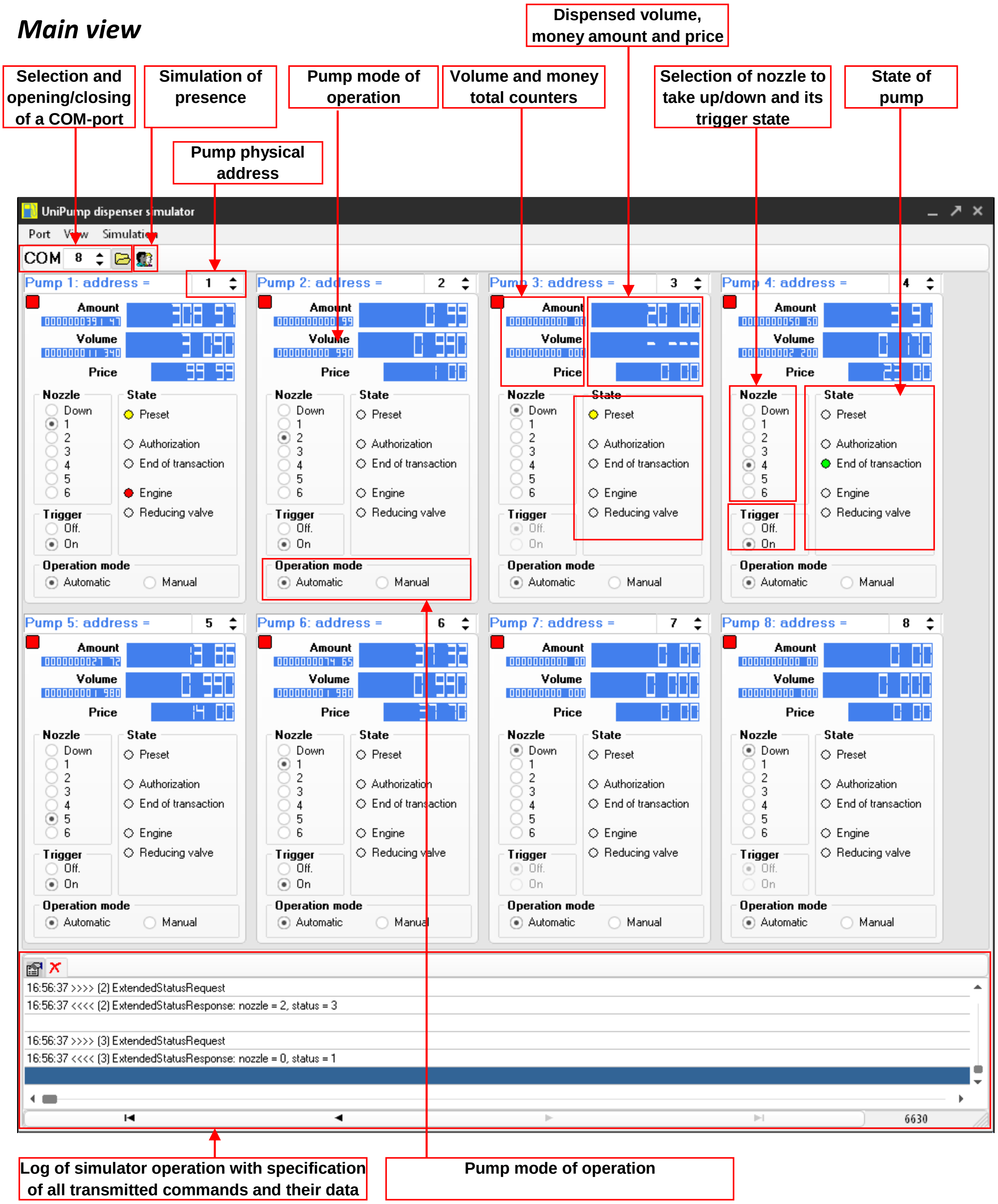
C# (developed in Visual Studio Express 2010) *VB.NET (developed in Visual Studio Express 2010)*

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

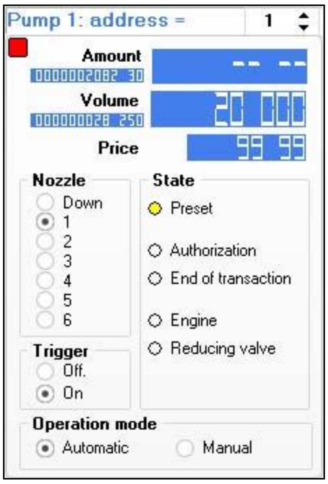


```
16:56:37 >>>> (2) ExtendedStatusRequest
16:56:37 <<<< (2) ExtendedStatusResponse: nozzle = 2, status = 3

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 1: address = 1

Amount 0000002082 30 -- --

Volume 000000028 250 20 000

Price 99 99

Nozzle
○ Down
● 1
○ 2
○ 3
○ 4
○ 5
○ 6

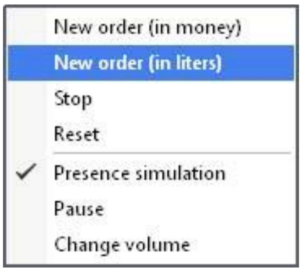
Trigger
○ Off.
● On

State
○ Preset
○ Authorization
○ End of transaction
○ Engine
○ Reducing valve

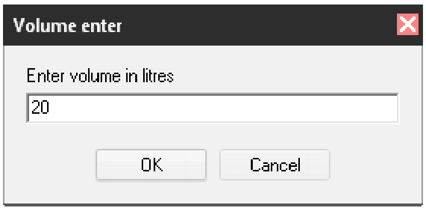
Operation mode
● Automatic ○ Manual

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



Volume enter

Enter volume in litres
20

Volume preset menu for pump

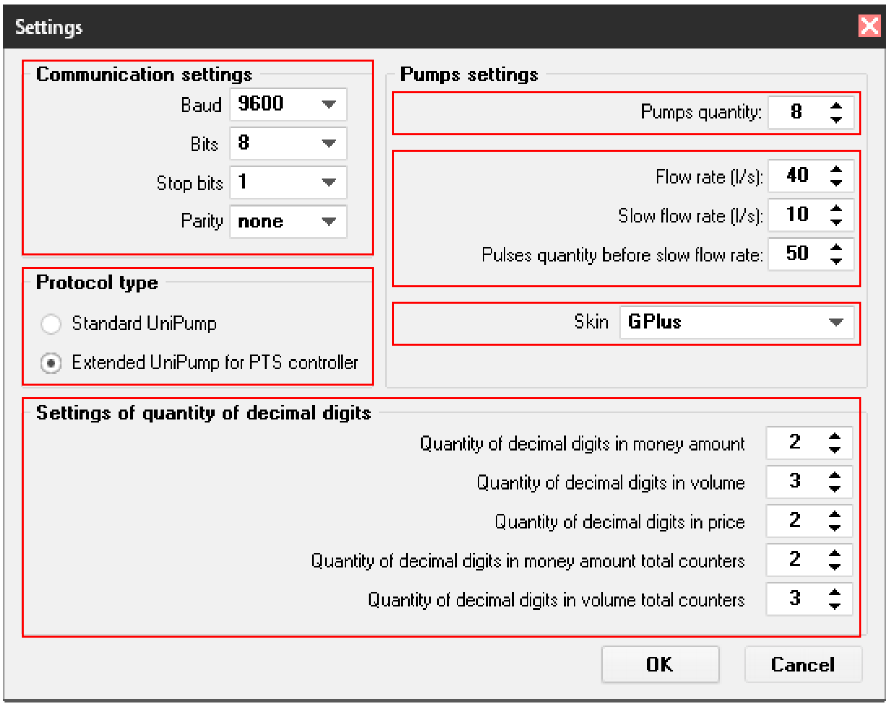
Configuration

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

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.



Settings

Communication settings
Baud 9600
Bits 8
Stop bits 1
Parity none

Protocol type
○ Standard UniPump
● Extended UniPump for PTS controller

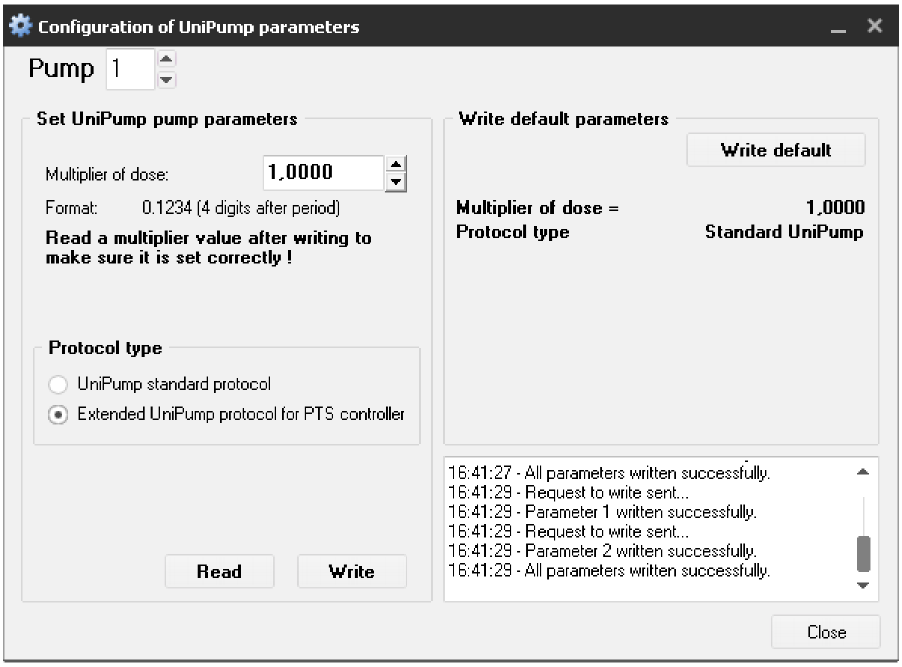
Pumps settings
Pumps quantity: 8
Flow rate (l/s): 40
Slow flow rate (l/s): 10
Pulses quantity before slow flow rate: 50
Skin GPlus

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

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

```
16:41:27 - All parameters written successfully.
16:41:29 - Request to write sent...
16:41:29 - Parameter 1 written successfully.
16:41:29 - Request to write sent...
16:41:29 - Parameter 2 written successfully.
16:41:29 - All parameters written successfully.
```

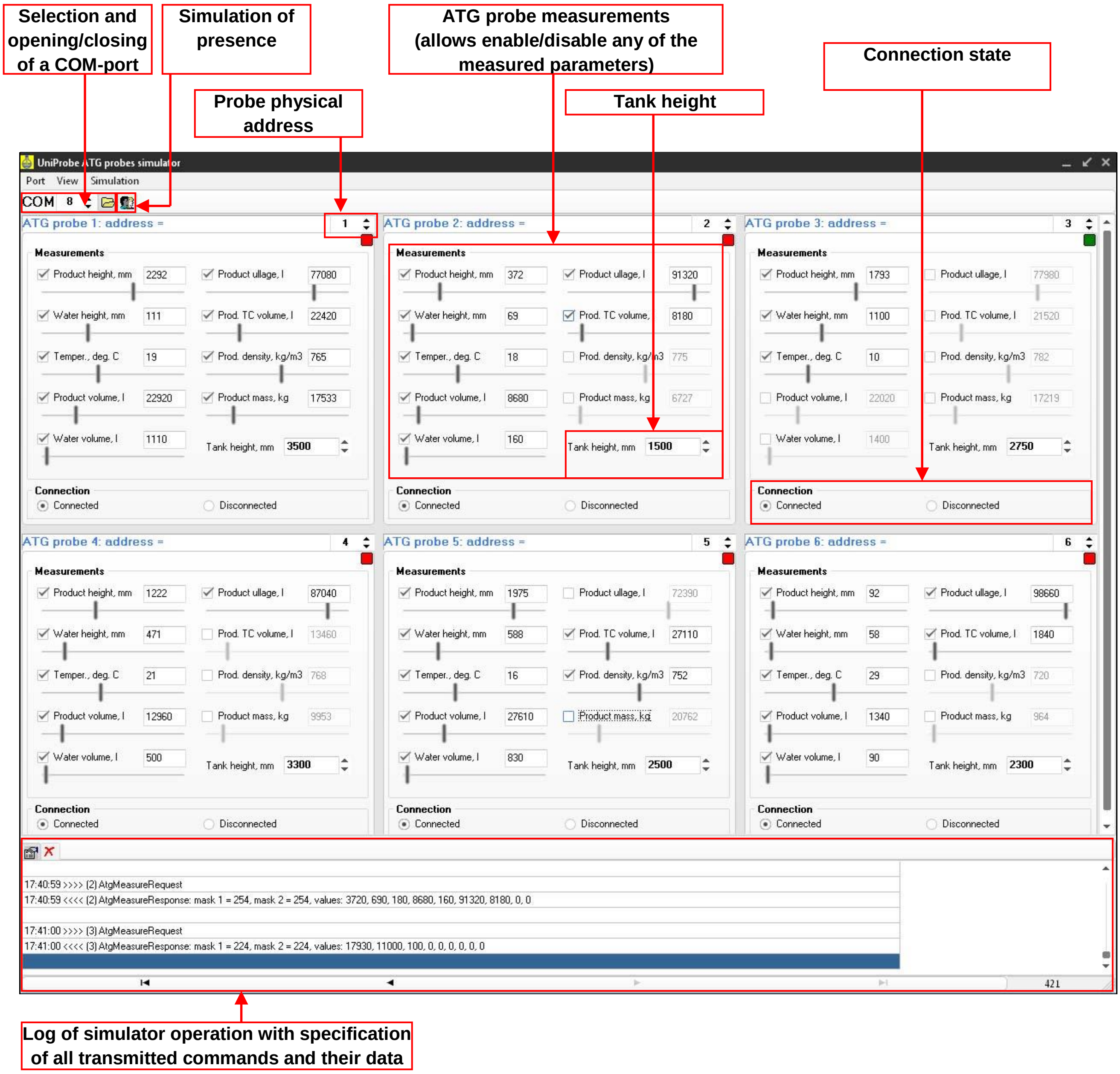
Parameters for UniPump protocol in PTS controller

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

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
```

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

ATG probes software simulator main view

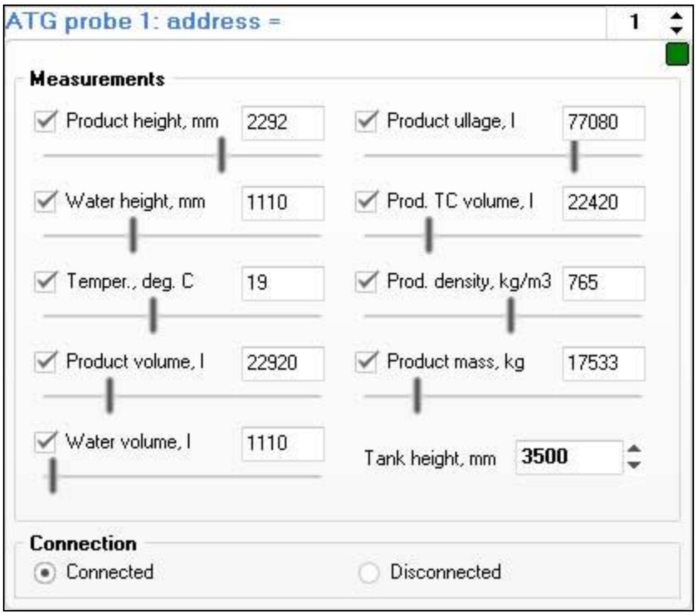
In ATG probes simulator it is possible to set measurements of the following parameters:

- 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

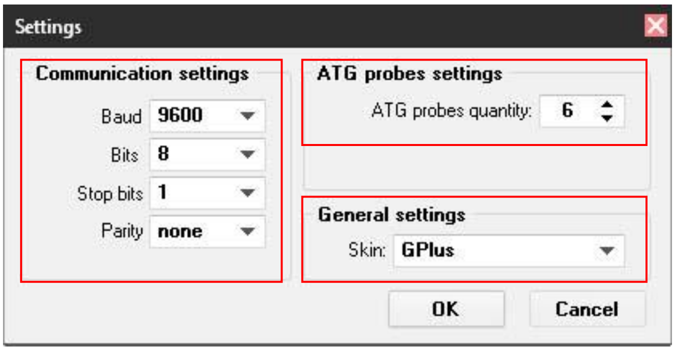
● Connected ○ Disconnected

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.

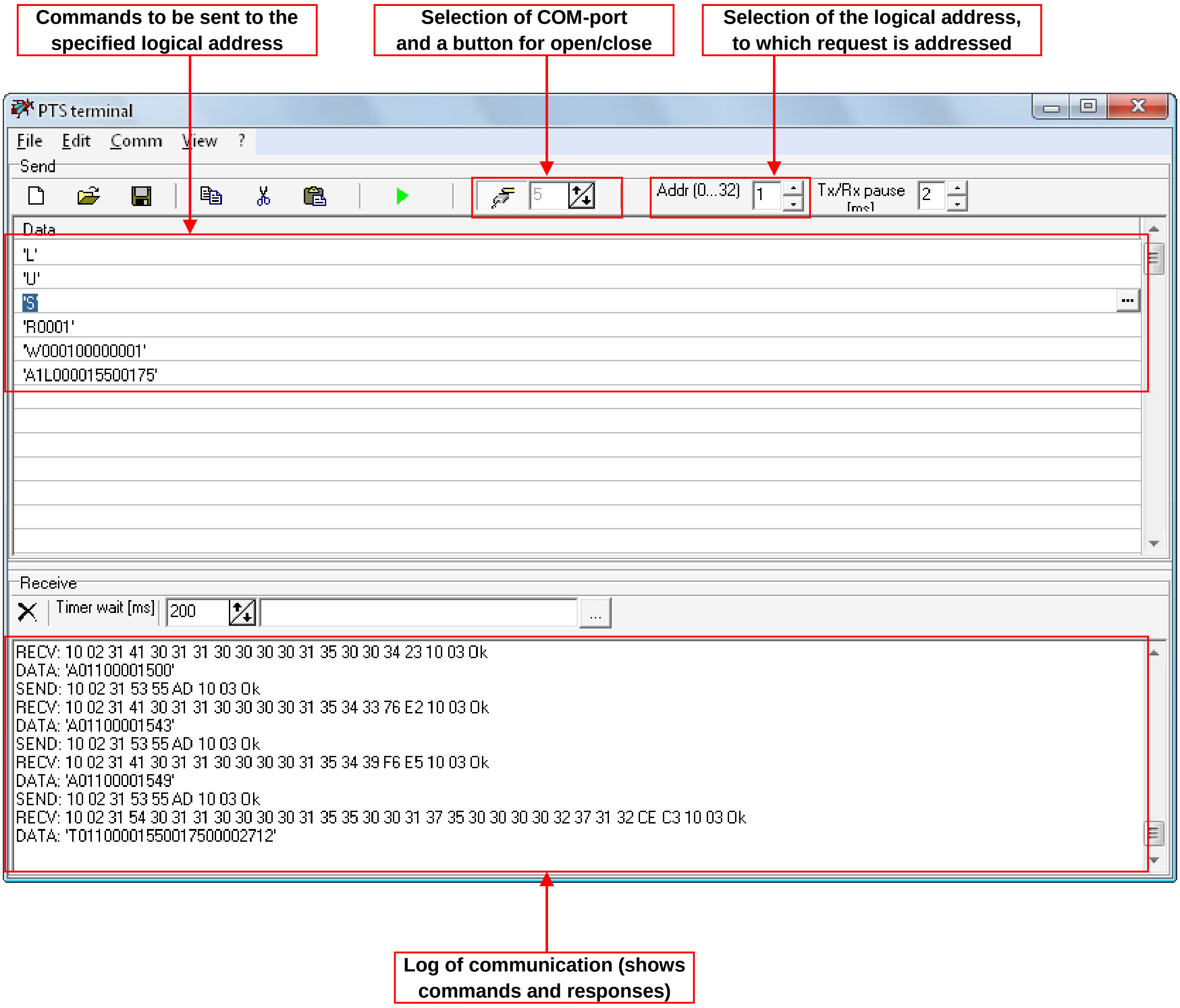
For communication of the PTS controller with UniProbe ATG probes software simulator it is necessary to select in PTS controller configuration of the ATG channel communication protocol “*9. UNIPROBE*” and select appropriate baud rate (equal to set in simulator (9600 baud rate by defaults) and set probes physical addresses equal to those, which are set for probes in simulator.

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

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

```
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)

Step-by-step configuration of the PTS controller SDK

Purpose

This step-by-step instruction is provided in order to simplify understanding of PTS controller SDK operation and assembling. 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

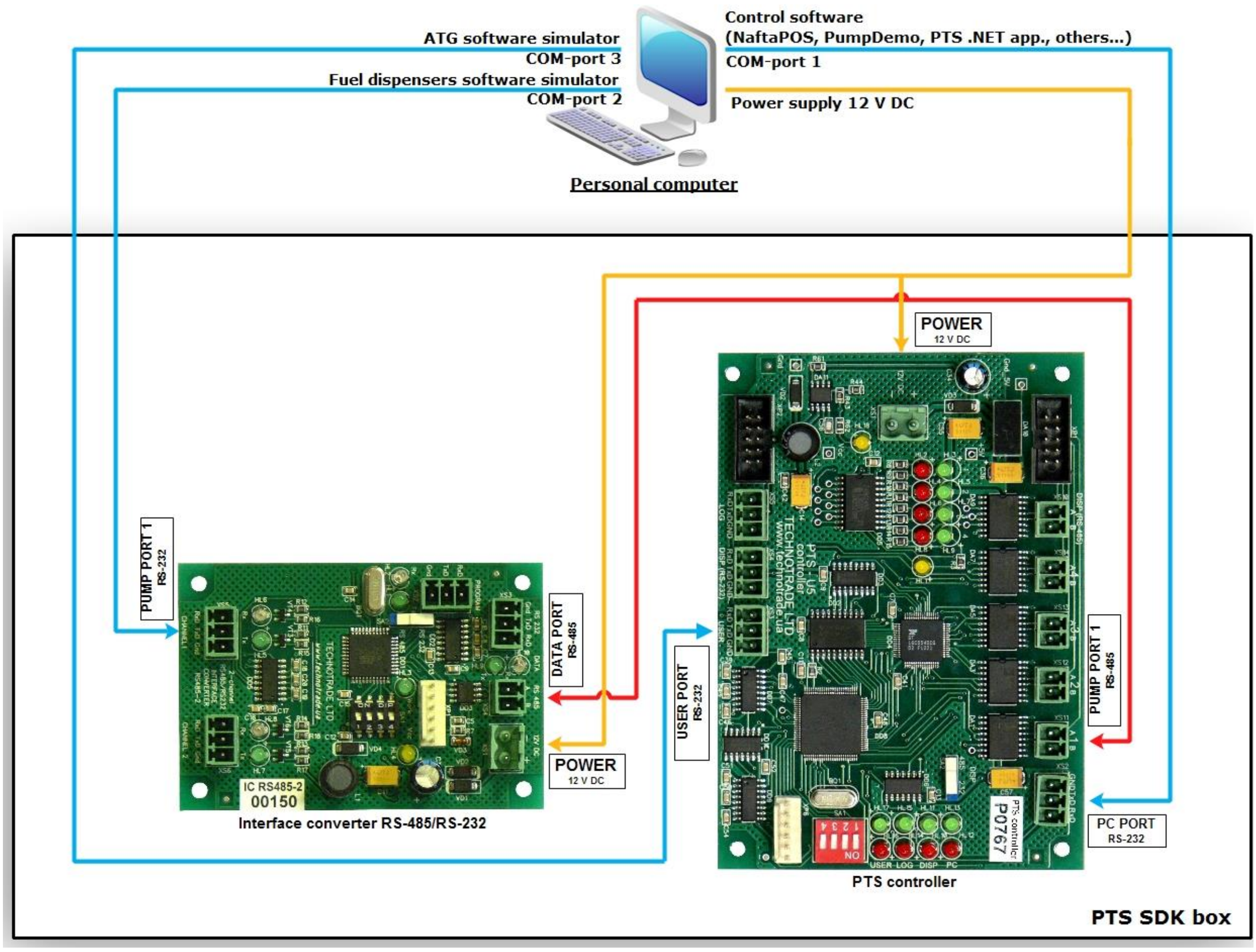
Step 2. Assembling PTS controller SDK cabling and connection to PC

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, Bluetooth-to-COM, other possible.

USB-to-COM interface converter cable

PTS controller SDK is generally supplied in a plastic box, in which the PTS controller and RS-485/RS-232 interface converter boards are mounted. All cables inside the PTS controller SDK are already connected. In order to start operation it is necessary to connect interfaces to PC COM-ports and power on the PTS controller SDK. 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:



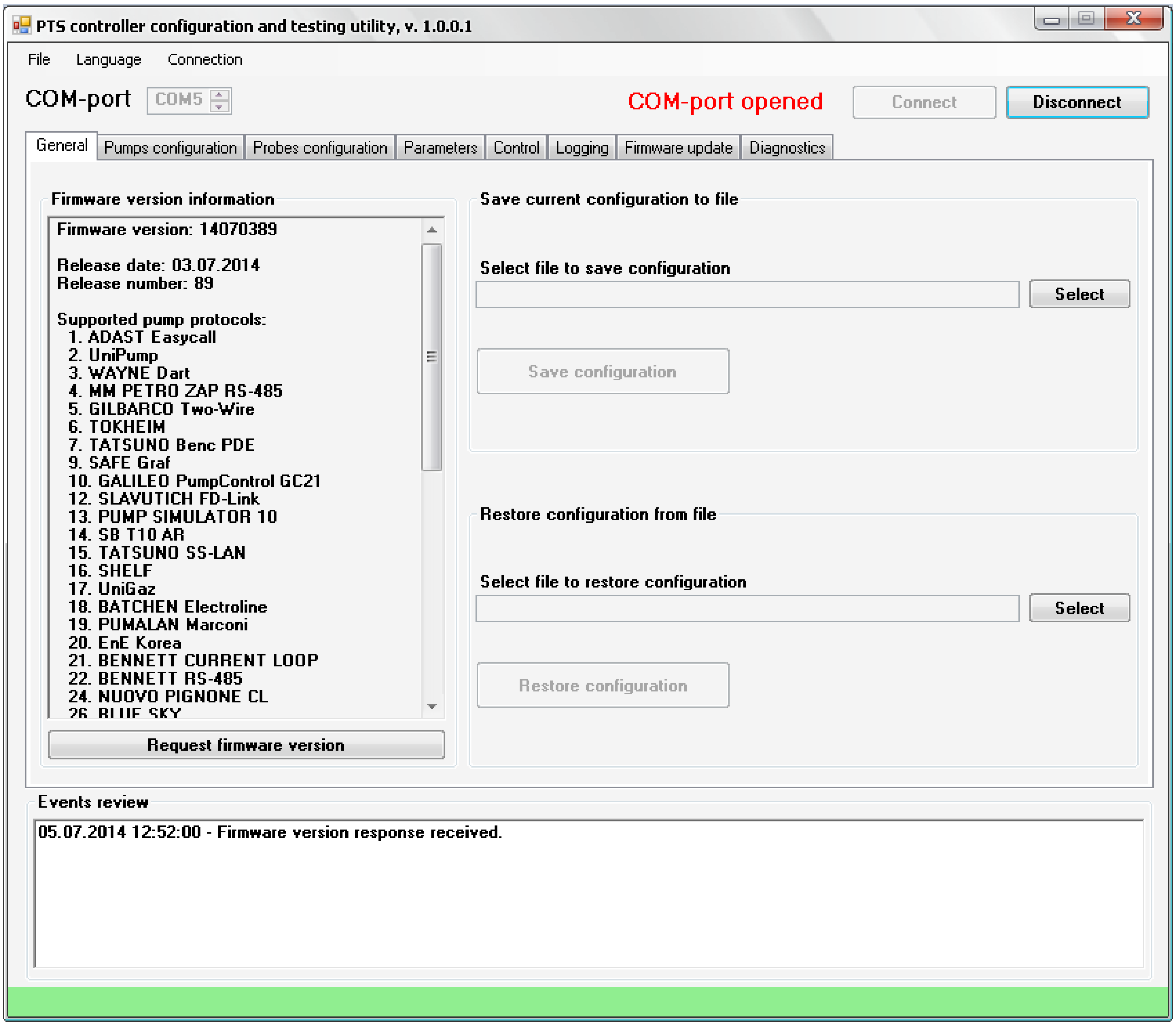
1. Connection of PC with PTS controller: connect cable coming from PC port of PTS controller to COM-port 1 of PC (where the control software is to be launched).
2. Connection of RS-485/RS-232 interface converter to PC: connect cable coming from PUMP PORT 1 of RS-485/RS-232 interface converter to COM-port 2 of PC, on which the pump software simulator is to be launched.
3. Make sure that PUMP PORT 1 of PTS controller is connected to DATA PORT (RS-485) of RS-485/RS-232 interface converter.
4. Make sure that switch SA2 of RS-485/RS-232 interface converter is set to position "*RS-485*" and all switches of DIP-switch SA1 are set into position OFF.
5. Make sure that power supply cable is connected to POWER connector of PTS controller and POWER connector of RS-485/RS-232 interface converter.
6. (Optional, may be skipped) In order to use software simulator of ATG system connect USER port of PTS controller to COM-port 3 of PC, on which the ATG system software simulator is to be launched.
7. Switch on (power on) the PTS controller SDK (requires 12 V DC power supply source).

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 3. Starting of the PTS controller configuration utility Pts_config.exe

Run *Pts_config.exe* and from the main menu select connection type to be “*Direct connection to PC COM-port*”, select COM-port with connected PTS controller and click on the “*Connect*” button. Make sure that the port is opened (red inscription “COM-port opened” should appear).

Click on button “Request firmware version” to get firmware information from PTS controller.



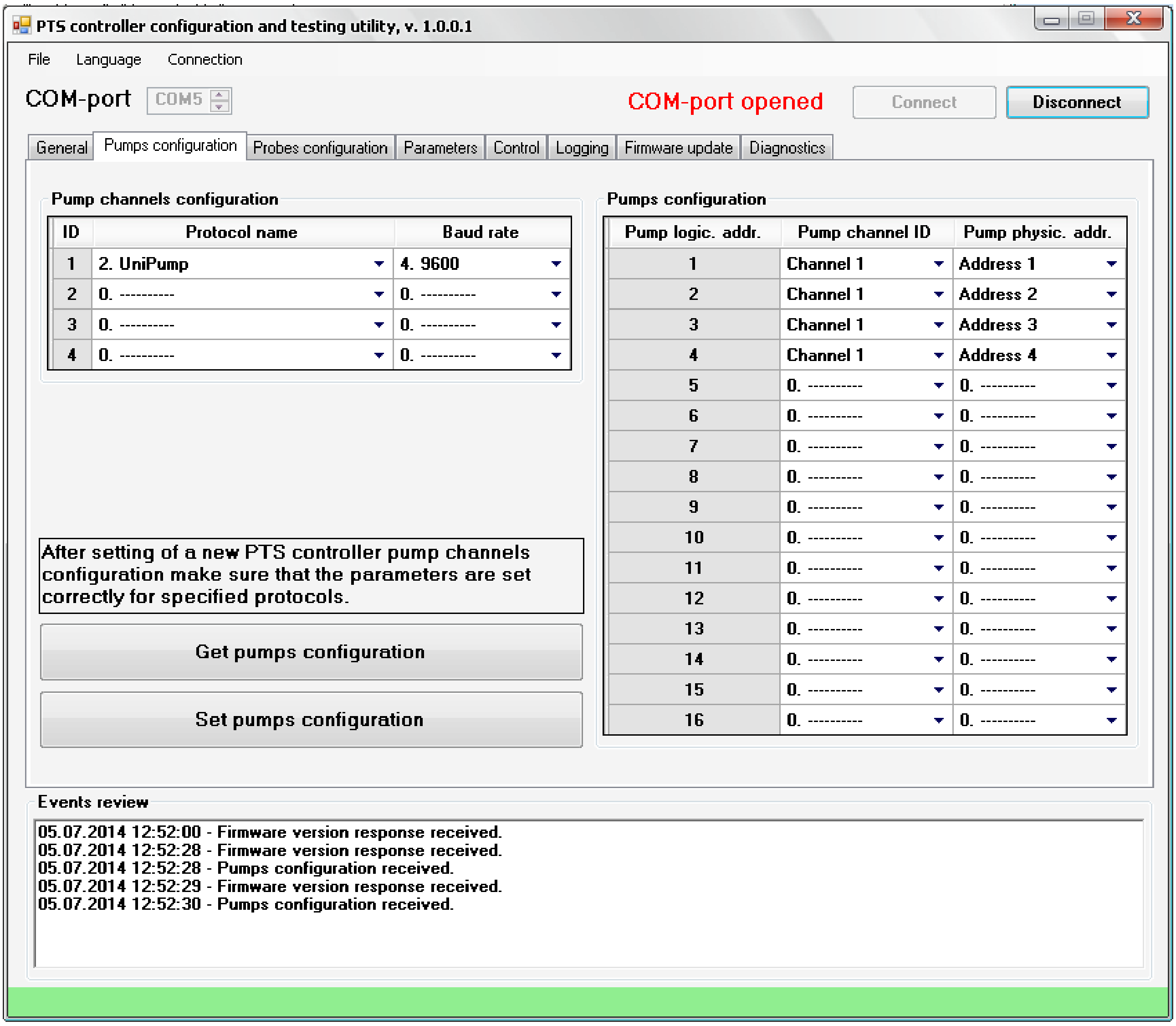
```
Firmware version: 14070389

Release date: 03.07.2014
Release number: 89

Supported pump protocols:
  1. ADAST Easycall
  2. UniPump
  3. WAYNE Dart
  4. MM PETRO ZAP RS-485
  5. GILBARCO Two-Wire
  6. TOKHEIM
  7. TATSUNO Benc PDE
  9. SAFE Graf
  10. GALILEO PumpControl GC21
  12. SLAVUTICH FD-Link
  13. PUMP SIMULATOR 10
  14. SB T10 AR
  15. TATSUNO SS-LAN
  16. SHELF
  17. UniGaz
  18. BATCHEN Electroline
  19. PUMALAN Marconi
  20. EnE Korea
  21. BENNETT CURRENT LOOP
  22. BENNETT RS-485
  24. NUOVO PIGNONE CL
  26. BLUE SKY
```

```
05.07.2014 12:52:00 - Firmware version response received.
```

Step 4. Configuration of pump channels

Go to tab "*Pumps configuration*" and set pumps channels configuration as shown on below screenshot. Set protocol "*2. UniPump*" for pump channel 1 and baud rate "*4. 9600*". Assign this pump channel to pumps 1 - 4, set the physical addresses of these pumps also 1 – 4.

Click the button "*Set pumps configuration*" to save configuration in PTS controller.



Pump channels configuration

ID	Protocol name	Baud rate
1	2. UniPump	4. 9600
2	0. ----------	0. ----------
3	0. ----------	0. ----------
4	0. ----------	0. ----------

After setting of a new PTS controller pump channels configuration make sure that the parameters are set correctly for specified protocols.

Pumps configuration

Pump logic. addr.	Pump channel ID	Pump physic. addr.
1	Channel 1	Address 1
2	Channel 1	Address 2
3	Channel 1	Address 3
4	Channel 1	Address 4
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. ----------

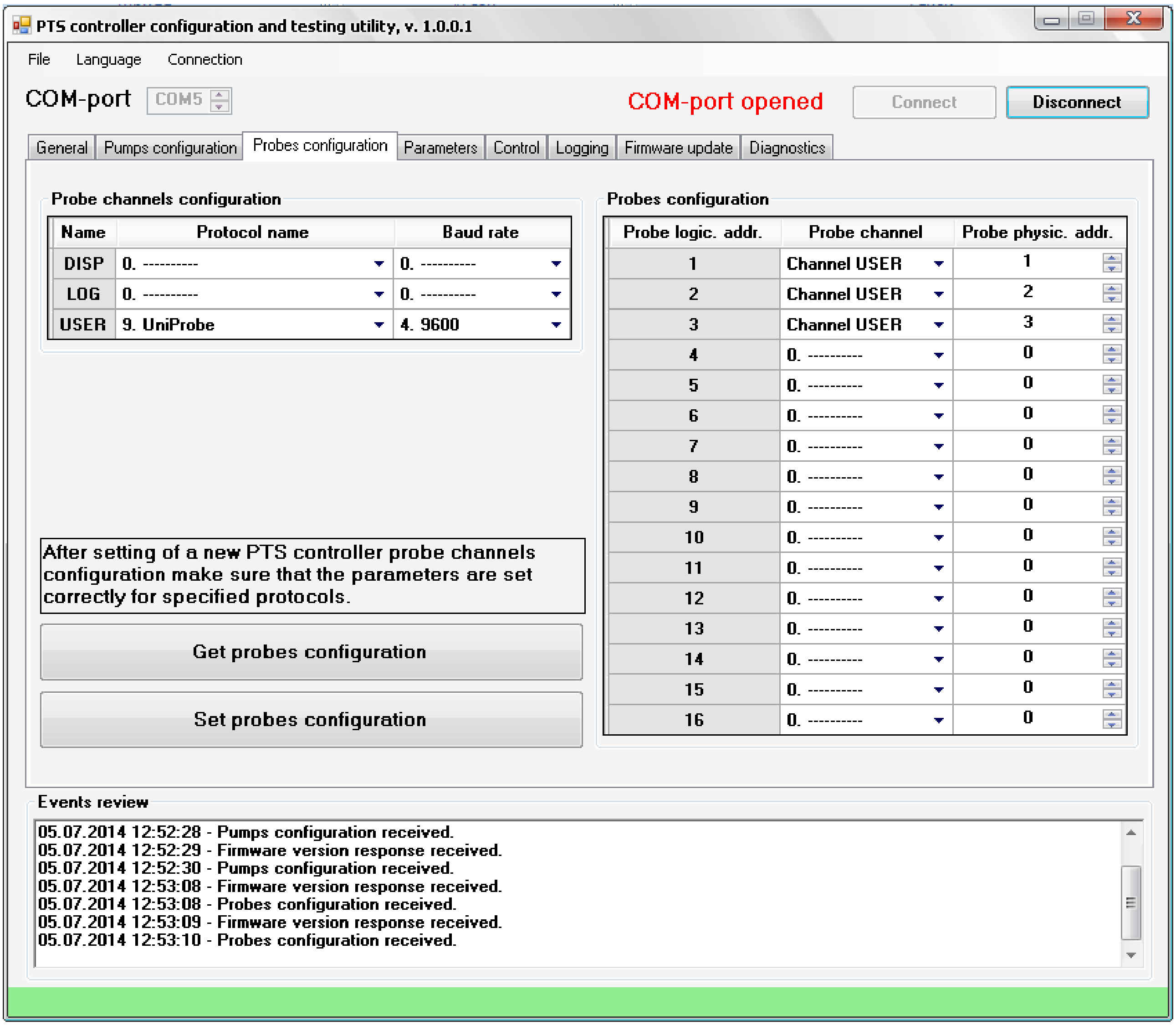
Events review

```
05.07.2014 12:52:00 - Firmware version response received.
05.07.2014 12:52:28 - Firmware version response received.
05.07.2014 12:52:28 - Pumps configuration received.
05.07.2014 12:52:29 - Firmware version response received.
05.07.2014 12:52:30 - Pumps configuration received.
```

Step 5. Configuration of probes channels

Go to tab "*Probes configuration*" and set probes channels configuration as shown on below screenshot. Set protocol "*9. UniProbe*" for probe channel DISP and baud rate "*4. 9600*". Assign this probe channel to probes 1 - 3, set the physical addresses of these probes also 1 – 3.

Click the button "*Set pumps configuration*" to save configuration in PTS controller.



Probe channels configuration

Name	Protocol name	Baud rate
DISP	0. ----------	0. ----------
LOG	0. ----------	0. ----------
USER	9. UniProbe	4. 9600

After setting of a new PTS controller probe channels configuration make sure that the parameters are set correctly for specified protocols.

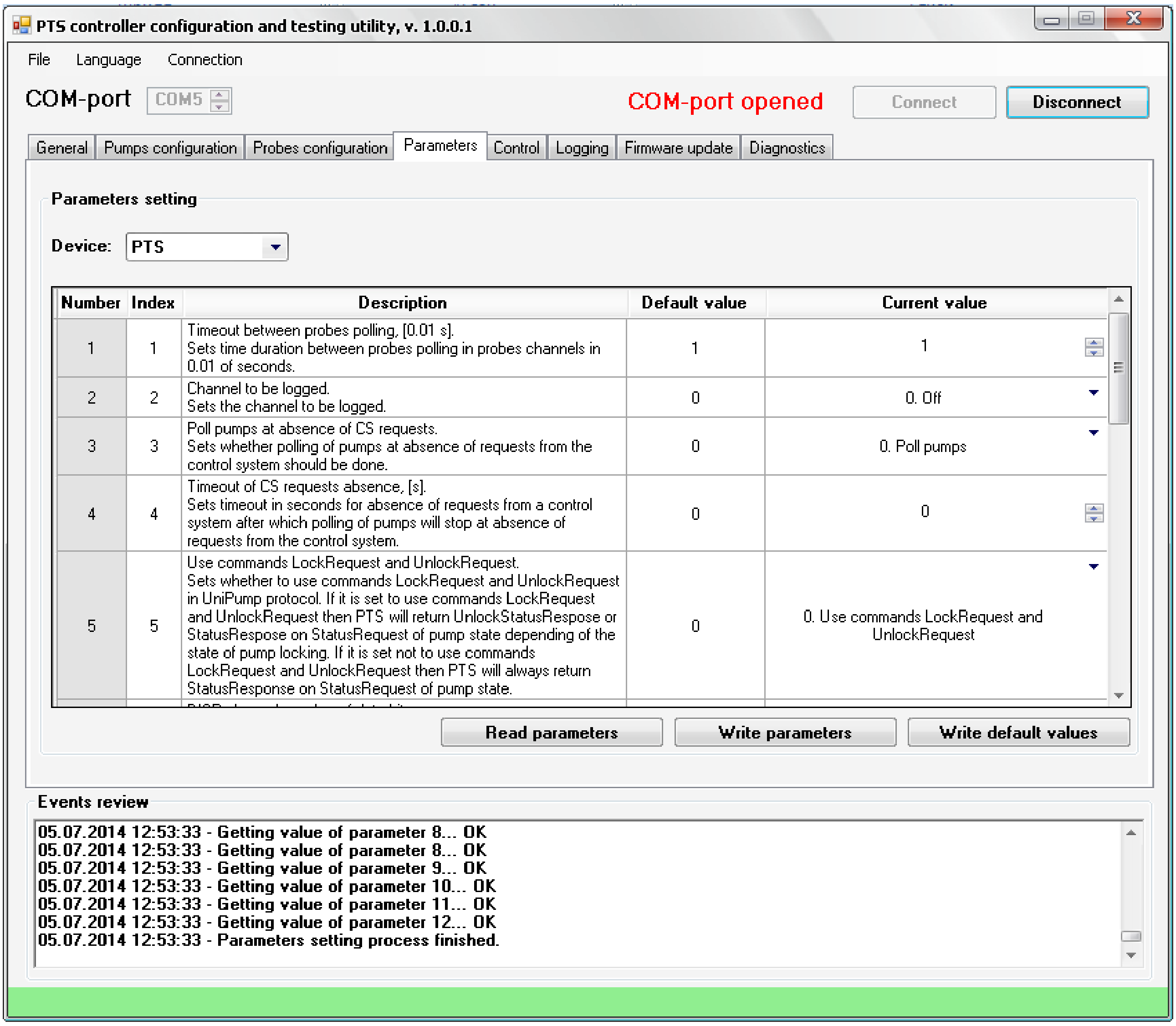
Probes configuration

Probe logic. addr.	Probe channel	Probe physic. addr.
1	Channel USER	1
2	Channel USER	2
3	Channel USER	3
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

```
05.07.2014 12:52:28 - Pumps configuration received.
05.07.2014 12:52:29 - Firmware version response received.
05.07.2014 12:52:30 - Pumps configuration received.
05.07.2014 12:53:08 - Firmware version response received.
05.07.2014 12:53:08 - Probes configuration received.
05.07.2014 12:53:09 - Firmware version response received.
05.07.2014 12:53:10 - Probes configuration received.
```

Step 6. Configuration of PTS controller parameters

Go to tab “*Parameters*” and select from a device list item “*PTS*”, click on a button “*Write default parameters*”.



COM-port COM5

COM-port opened

Parameters setting

Device: PTS

Number	Index	Description	Default value	Current value
1	1	Timeout between probes polling, [0.01 s]. Sets time duration between probes polling in probes channels in 0.01 of seconds.	1	1
2	2	Channel to be logged. Sets the channel to be logged.	0	0. Off
3	3	Poll pumps at absence of CS requests. Sets whether polling of pumps at absence of requests from the control system should be done.	0	0. Poll pumps
4	4	Timeout of CS requests absence, [s]. Sets timeout in seconds for absence of requests from a control system after which polling of pumps will stop at absence of requests from the control system.	0	0
5	5	Use commands LockRequest and UnlockRequest. Sets whether to use commands LockRequest and UnlockRequest in UniPump protocol. If it is set to use commands LockRequest and UnlockRequest then PTS will return UnlockStatusRespose or StatusRespose on StatusRequest of pump state depending of the state of pump locking. If it is set not to use commands LockRequest and UnlockRequest then PTS will always return StatusResponse on StatusRequest of pump state.	0	0. Use commands LockRequest and UnlockRequest

Events review

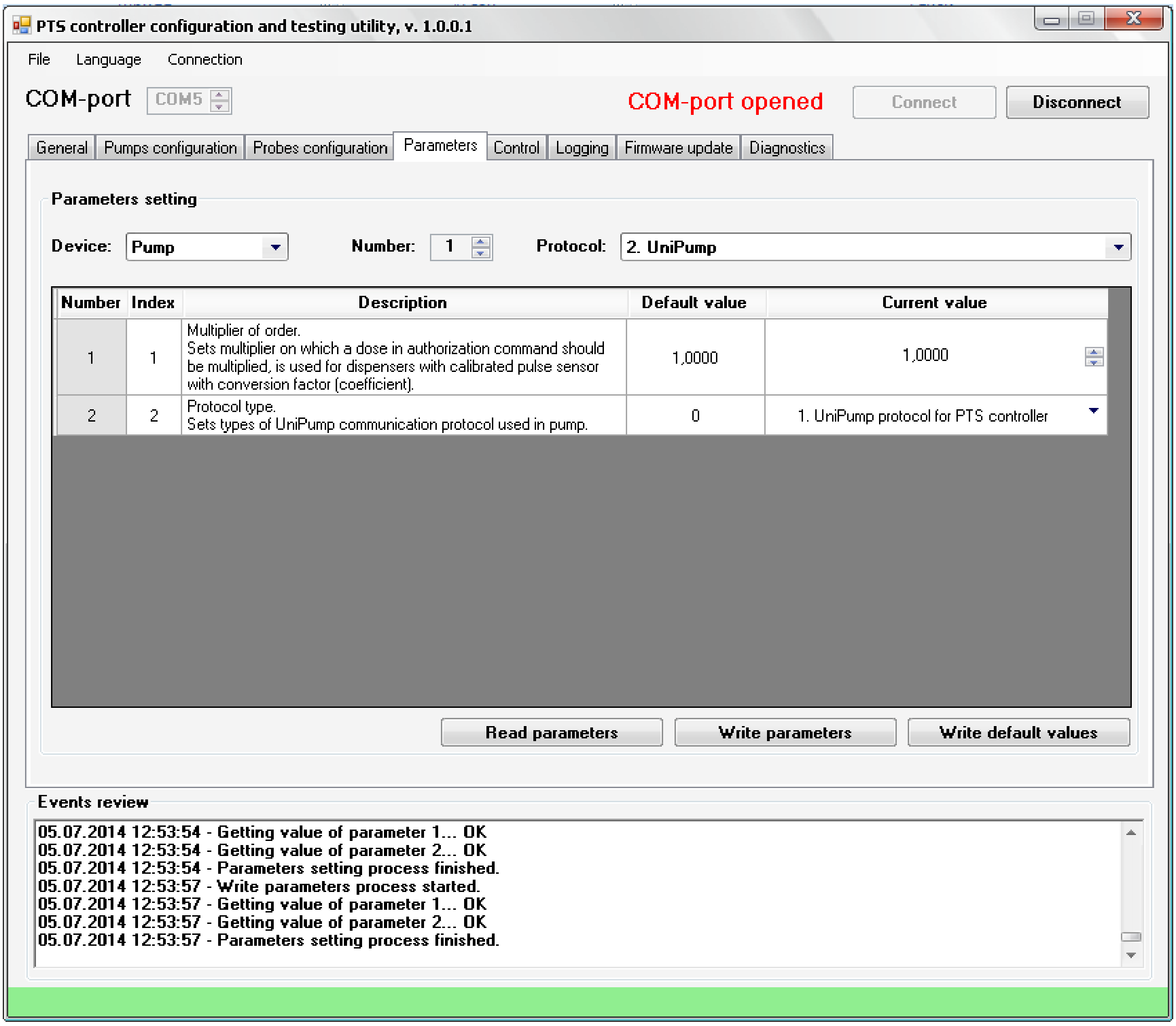
```
05.07.2014 12:53:33 - Getting value of parameter 8... OK
05.07.2014 12:53:33 - Getting value of parameter 8... OK
05.07.2014 12:53:33 - Getting value of parameter 9... OK
05.07.2014 12:53:33 - Getting value of parameter 10... OK
05.07.2014 12:53:33 - Getting value of parameter 11... OK
05.07.2014 12:53:33 - Getting value of parameter 12... OK
05.07.2014 12:53:33 - Parameters setting process finished.
```

Step 7. Configuration of pumps parameters

On tab "*Parameters*" select from a device list item "*Pump*", in a field "*Number*" select pump 1, in a protocol list – "*2. UniPump*" (which means configuration of UniPump protocol parameters for pump 1).

Click on a button "*Write default parameters*" to write default values. Change the parameter with index 2 "*Protocol type*" to value "*1. UniPump protocol for PTS*" as it is shown on below screenshot.

Save current configuration for each of 4 pumps (select pumps one by one using field "*Number*").



Device: Pump
Number: 1
Protocol: 2. UniPump

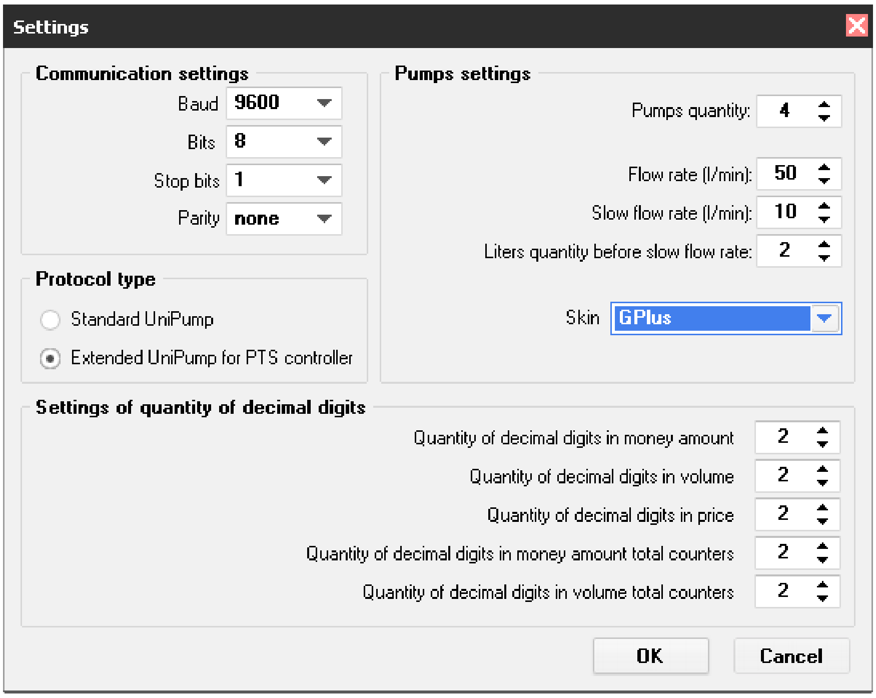
Number	Index	Description	Default value	Current value
1	1	Multiplier of order. Sets multiplier on which a dose in authorization command should be multiplied, is used for dispensers with calibrated pulse sensor with conversion factor (coefficient).	1,0000	1,0000
2	2	Protocol type. Sets types of UniPump communication protocol used in pump.	0	1. UniPump protocol for PTS controller

```
05.07.2014 12:53:54 - Getting value of parameter 1... OK
05.07.2014 12:53:54 - Getting value of parameter 2... OK
05.07.2014 12:53:54 - Parameters setting process finished.
05.07.2014 12:53:57 - Write parameters process started.
05.07.2014 12:53:57 - Getting value of parameter 1... OK
05.07.2014 12:53:57 - Getting value of parameter 2... OK
05.07.2014 12:53:57 - Parameters setting process finished.
```

At this step configuration of the PTS controller is finished.

Step 8. Configuration of pumps software simulator

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



Settings

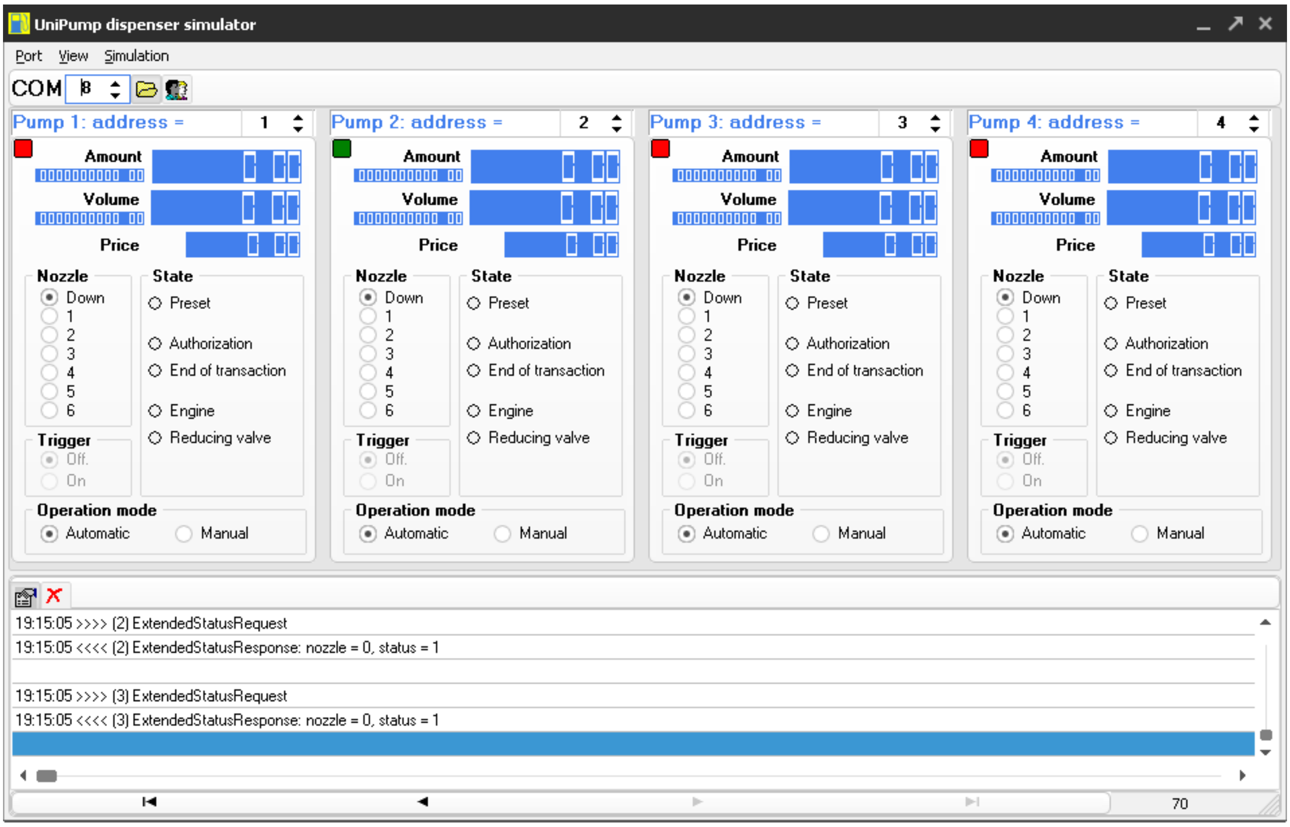
Communication settings
Baud: 9600
Bits: 8
Stop bits: 1
Parity: none

Protocol type
○ Standard UniPump
● Extended UniPump for PTS controller

Pumps settings
Pumps quantity: 4
Flow rate (l/min): 50
Slow flow rate (l/min): 10
Liters quantity before slow flow rate: 2
Skin: GPlus

Settings of quantity of decimal digits
Quantity of decimal digits in money amount: 2
Quantity of decimal digits in volume: 2
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: 2

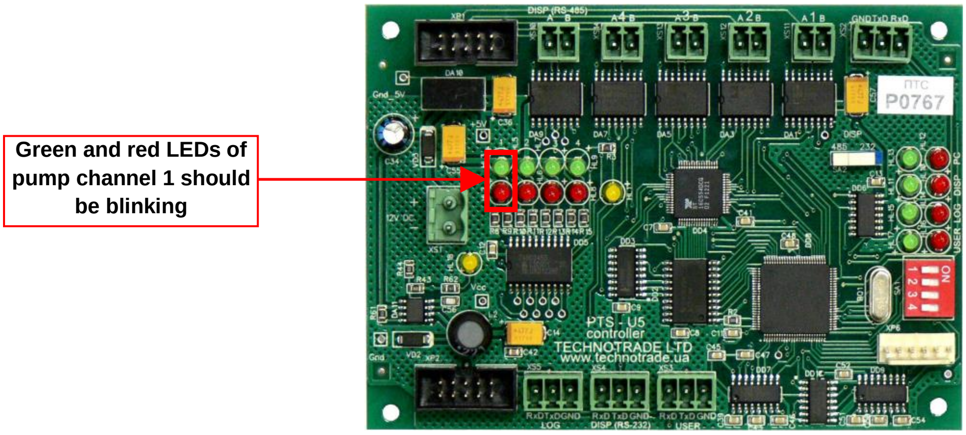
After properties are set click OK and on the main form select a COM-port, to which pump port of RS-485/RS-232 interface converter board is connected, and click “*Open*” button:



```
19:15:05 >>>> [2] ExtendedStatusRequest
19:15:05 <<<< [2] ExtendedStatusResponse: nozzle = 0, status = 1

19:15:05 >>>> [3] ExtendedStatusRequest
19:15:05 <<<< [3] ExtendedStatusResponse: nozzle = 0, status = 1
```

After COM-port is opened in case if there is communication between the PTS controller and pump simulator – you should see blinking of red squares located in left top corner of pump icons in pump simulator, and communication will be logged in log window of the simulator.

At this in PTS controller on pump channel 1 green and red LEDs will begin to blink. Blinking of the green LED means that PTS controller is sending requests to the pump, blinking of the red LED means that the PTS controller receives responses from the pump.

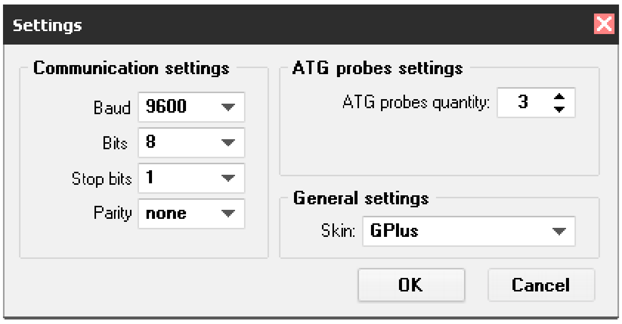


Green and red LEDs of pump channel 1 should be blinking

In case if only green LED is blinking on pump channel 1 – there is a mistake in wiring connection or configuration of the equipment (PTS controller or pump).

Step 9. Configuration of ATG probes software simulator

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



Settings

Communication settings
Baud: 9600
Bits: 8
Stop bits: 1
Parity: none

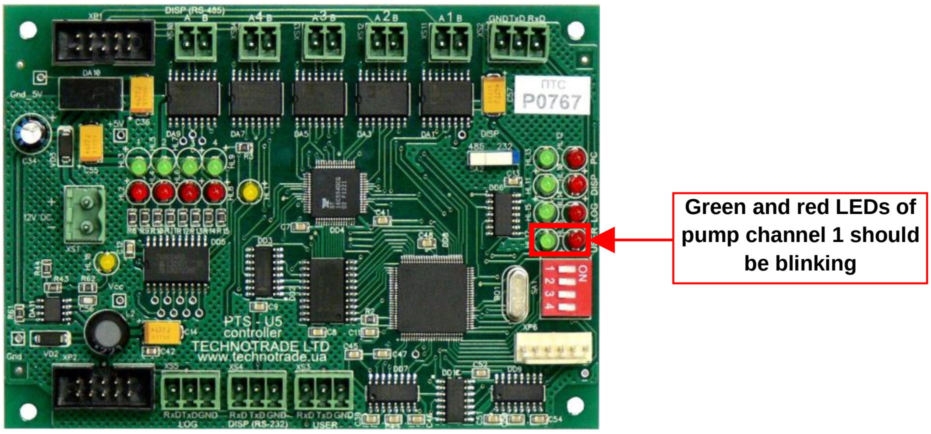
ATG probes settings
ATG probes quantity: 3

General settings
Skin: GPlus

After properties are set click OK and on the main form select a COM-port, to which USER port of PTS controller is connected, and click "*Open*" button.

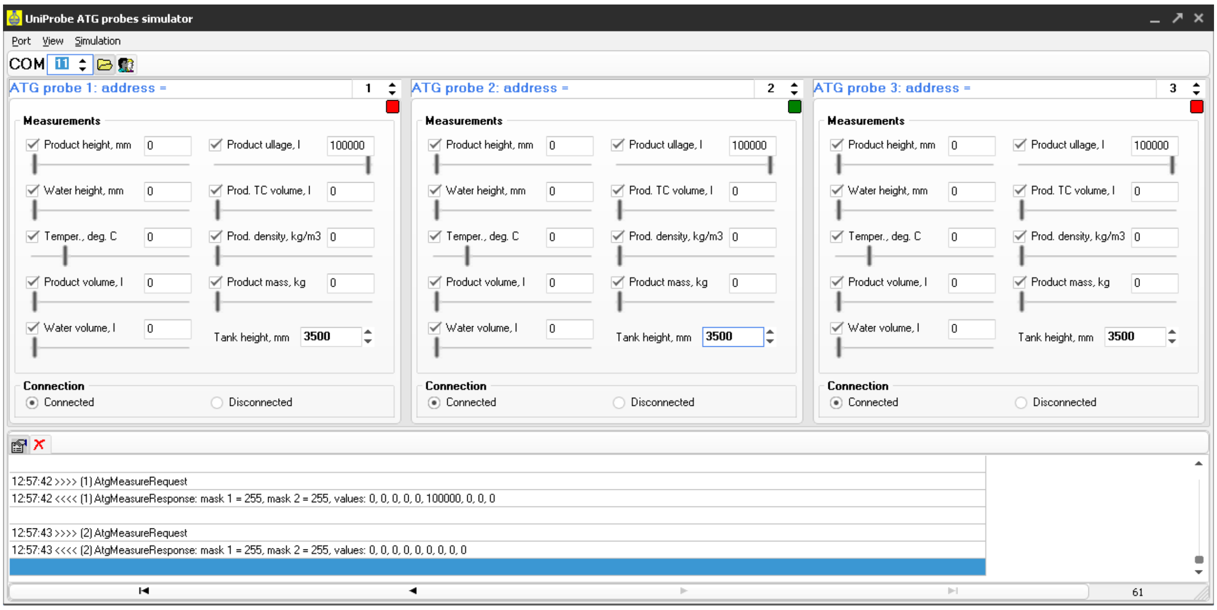
After COM-port is opened in case if there is communication between the PTS controller and ATG probes simulator – you should see blinking of red squares located in left top corner of ATG probes icons in ATG probes simulator, and communication will be logged in log window of the simulator.

At this in PTS controller on USER channel green and red LEDs will begin to blink once per second. Blinking of the green LED means that PTS controller is sending requests to the ATG, blinking of the red LED means that the PTS controller receives responses from the ATG.



Green and red LEDs of pump channel 1 should be blinking

In case if only green LED is blinking on USER channel – there is a mistake in wiring connection or configuration of the equipment (PTS controller or ATG).



UniProbe ATG probes simulator

ATG probe 1: address = 1

Measurements
☑ Product height, mm 0 ☑ Product ullage, l 100000
☑ Water height, mm 0 ☑ Prod. TC volume, l 0
☑ Temper., deg. C 0 ☑ Prod. density, kg/m3 0
☑ Product volume, l 0 ☑ Product mass, kg 0
☑ Water volume, l 0 Tank height, mm 3500

Connection
● Connected ○ Disconnected

ATG probe 2: address = 2

Measurements
☑ Product height, mm 0 ☑ Product ullage, l 100000
☑ Water height, mm 0 ☑ Prod. TC volume, l 0
☑ Temper., deg. C 0 ☑ Prod. density, kg/m3 0
☑ Product volume, l 0 ☑ Product mass, kg 0
☑ Water volume, l 0 Tank height, mm 3500

Connection
● Connected ○ Disconnected

ATG probe 3: address = 3

Measurements
☑ Product height, mm 0 ☑ Product ullage, l 100000
☑ Water height, mm 0 ☑ Prod. TC volume, l 0
☑ Temper., deg. C 0 ☑ Prod. density, kg/m3 0
☑ Product volume, l 0 ☑ Product mass, kg 0
☑ Water volume, l 0 Tank height, mm 3500

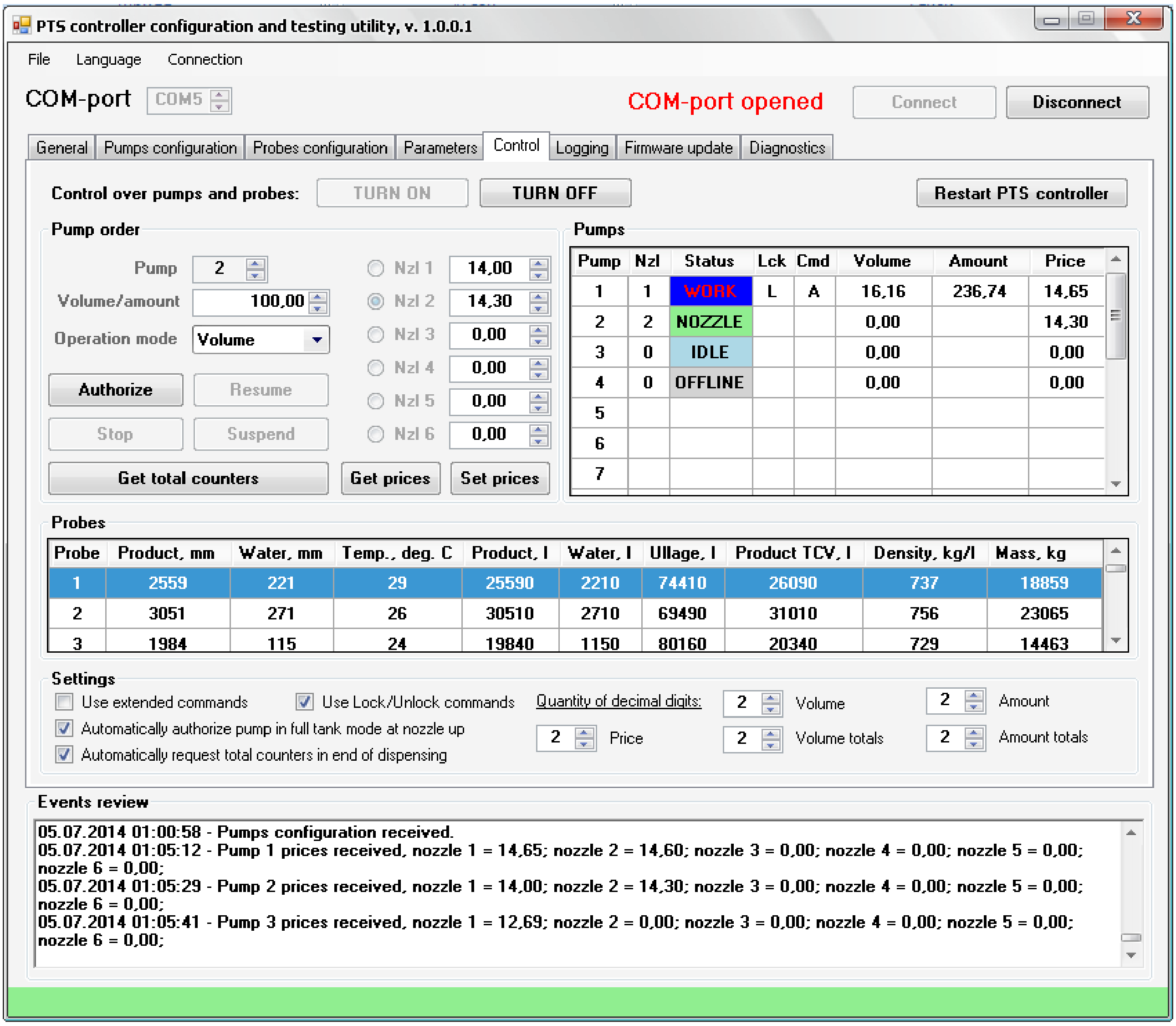
Connection
● Connected ○ Disconnected

```
12:57:42 >>>> (1) AtgMeasureRequest
12:57:42 <<<< (1) AtgMeasureResponse: mask 1 = 255, mask 2 = 255, values: 0, 0, 0, 0, 0, 100000, 0, 0, 0

12:57:43 >>>> (2) AtgMeasureRequest
12:57:43 <<<< (2) AtgMeasureResponse: mask 1 = 255, mask 2 = 255, values: 0, 0, 0, 0, 0, 0, 0, 0, 0
```

Step 10. Control over pumps and ATG probes in PTS controller configuration and testing utility

Go to tab "*Control*" in *Pts_config.exe* and click on the button "*TURN ON*".

Now it is necessary to set initial values in the pump simulator and ATG probes simulator to display them in the utility. Take up nozzles on the pumps and also set values for the ATGs (there is a possibility to emulate presence in the simulators to make them work if there was really process going on). In the *Pts_config.exe* utility you will see these changes.



Pump 2
Volume/amount 100,00
Operation mode Volume

○ Nzl 1 14,00
● Nzl 2 14,30
○ Nzl 3 0,00
○ Nzl 4 0,00
○ Nzl 5 0,00
○ Nzl 6 0,00

Pump	Nzl	Status	Lck	Cmd	Volume	Amount	Price
1	1	WORK	L	A	16,16	236,74	14,65
2	2	NOZZLE			0,00		14,30
3	0	IDLE			0,00		0,00
4	0	OFFLINE			0,00		0,00
5							
6							
7							

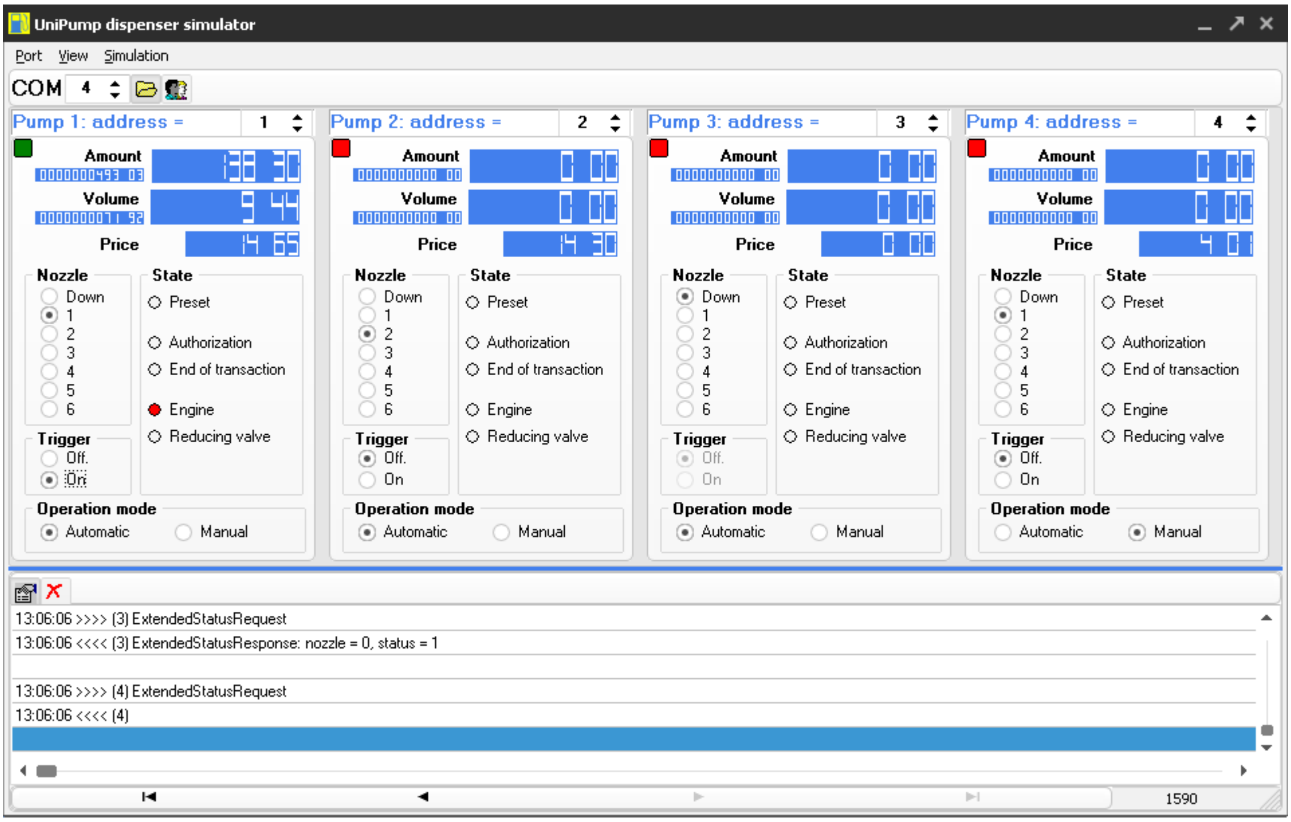
Probe	Product, mm	Water, mm	Temp., deg. C	Product, l	Water, l	Ullage, l	Product TCV, l	Density, kg/l	Mass, kg
1	2559	221	29	25590	2210	74410	26090	737	18859
2	3051	271	26	30510	2710	69490	31010	756	23065
3	1984	115	24	19840	1150	80160	20340	729	14463

☐ Use extended commands
☑ Use Lock/Unlock commands
☑ Automatically authorize pump in full tank mode at nozzle up
☑ Automatically request total counters in end of dispensing

Quantity of decimal digits:
2 Price
2 Volume
2 Volume totals
2 Amount
2 Amount totals

```
05.07.2014 01:00:58 - Pumps configuration received.
05.07.2014 01:05:12 - Pump 1 prices received, nozzle 1 = 14,65; nozzle 2 = 14,60; nozzle 3 = 0,00; nozzle 4 = 0,00; nozzle 5 = 0,00;
nozzle 6 = 0,00;
05.07.2014 01:05:29 - Pump 2 prices received, nozzle 1 = 14,00; nozzle 2 = 14,30; nozzle 3 = 0,00; nozzle 4 = 0,00; nozzle 5 = 0,00;
nozzle 6 = 0,00;
05.07.2014 01:05:41 - Pump 3 prices received, nozzle 1 = 12,69; nozzle 2 = 0,00; nozzle 3 = 0,00; nozzle 4 = 0,00; nozzle 5 = 0,00;
nozzle 6 = 0,00;
```

Now you are able to provide full control over the pumps and read receive data from the ATG probes:



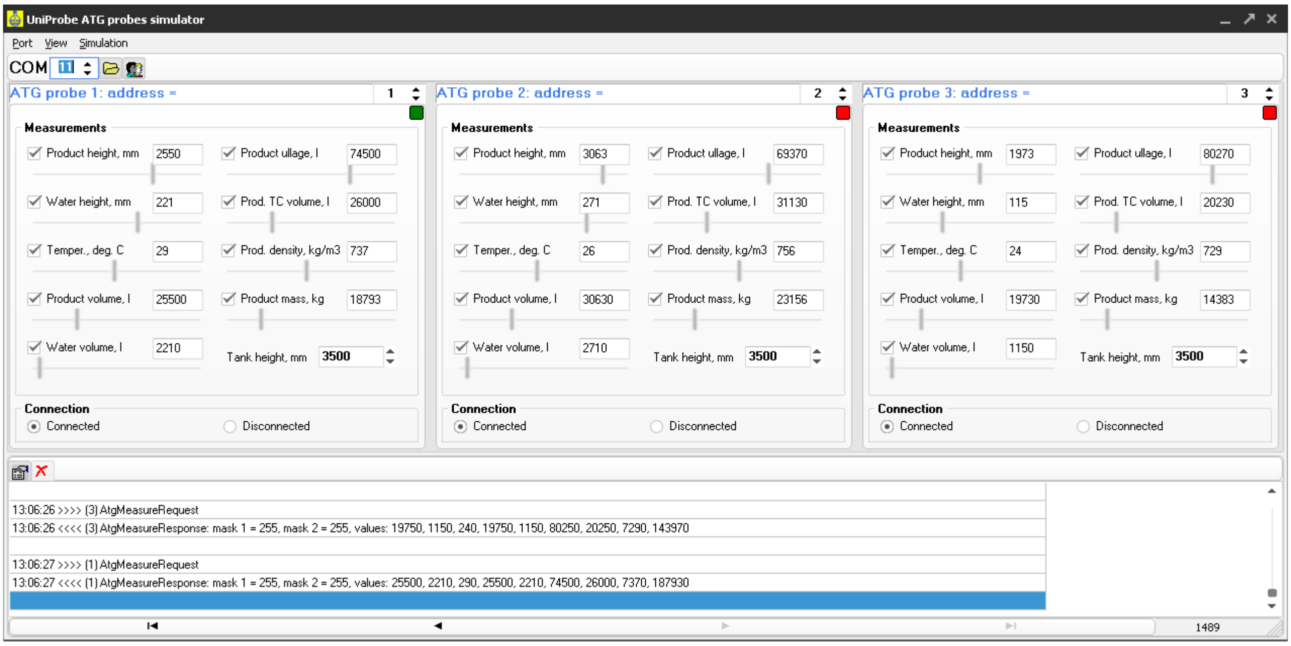
	Pump 1: address = 1	Pump 2: address = 2	Pump 3: address = 3	Pump 4: address = 4
Amount	138.30	0.00	0.00	0.00
Amount total	0000000493.03	0000000000.00	0000000000.00	0000000000.00
Volume	9.44	0.00	0.00	0.00
Volume total	0000000071.92	0000000000.00	0000000000.00	0000000000.00
Price	14.65	14.30	0.00	4.01
Nozzle	1	2	Down	1
State	Engine			
Trigger	On	Off.	Off.	Off.
Operation mode	Automatic	Automatic	Automatic	Manual

```
13:06:06 >>>> (3) ExtendedStatusRequest
13:06:06 <<<< (3) ExtendedStatusResponse: nozzle = 0, status = 1

13:06:06 >>>> (4) ExtendedStatusRequest
13:06:06 <<<< (4)
```

1590

Pumps software simulator

Try to make different operations with pumps: set/get prices, take up/down nozzles, get total counters, authorize a dispenser, others. Plays with decimal digits number to make them equal to used at your petrol stations (Note! Number of decimal digits should be also changed in pumps software simulator).



Measurements	ATG probe 1: address = 1	ATG probe 2: address = 2	ATG probe 3: address = 3
☑ Product height, mm	2550	3063	1973
☑ Product ullage, l	74500	69370	80270
☑ Water height, mm	221	271	115
☑ Prod. TC volume, l	26000	31130	20230
☑ Temper., deg. C	29	26	24
☑ Prod. density, kg/m3	737	756	729
☑ Product volume, l	25500	30630	19730
☑ Product mass, kg	18793	23156	14383
☑ Water volume, l	2210	2710	1150
Tank height, mm	3500	3500	3500
Connection	Connected	Connected	Connected

```
13:06:26 >>>> (3) AtgMeasureRequest
13:06:26 <<<< (3) AtgMeasureResponse: mask 1 = 255, mask 2 = 255, values: 19750, 1150, 240, 19750, 1150, 80250, 20250, 7290, 143970

13:06:27 >>>> (1) AtgMeasureRequest
13:06:27 <<<< (1) AtgMeasureResponse: mask 1 = 255, mask 2 = 255, values: 25500, 2210, 290, 25500, 2210, 74500, 26000, 7370, 187930
```

1489

ATG probes software simulator

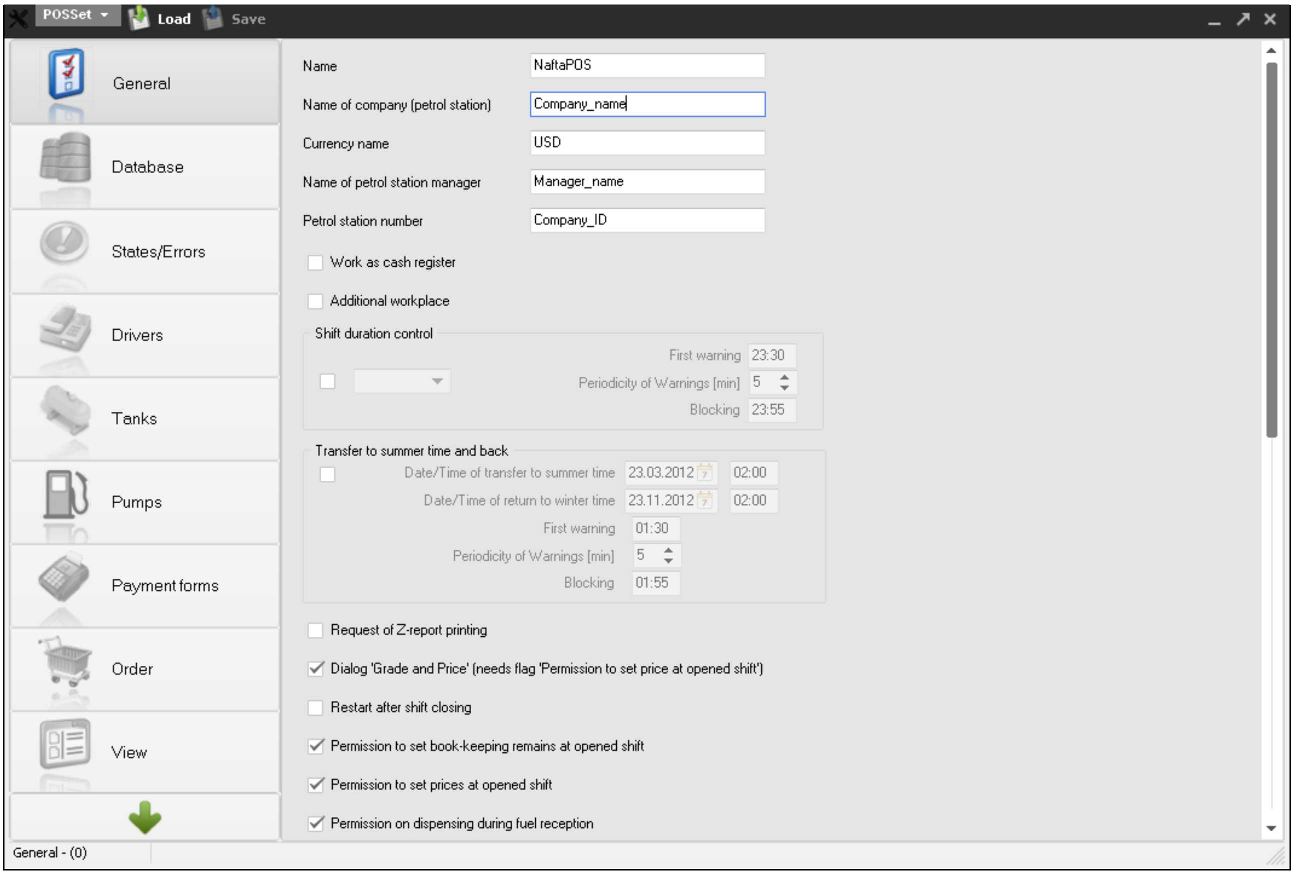
Try to change the measured values of fuel in tanks and see how soon they are updated in *Pts_config.exe* utility.

Step 11. Configuration of NaftaPOS software and control over pumps and ATG probes

Using instructions given in user manual of NaftaPOS software for petrol stations (can be downloaded from NaftaPOS software web-page: *http://www.technotrade.ua/petrol_station_software.html*) it is necessary to download and install NaftaPOS software. During installation of NaftaPOS software Guardant USB dongle drivers are to be installed.

When PTS controller is connected to the PC (COM-port 1 for communication with PTS controller, COM-port 2 for communication with pumps software simulator, COM-port 3 for communication with ATG probes software simulator) and NaftaPOS software is installed it is necessary to run POSSet.exe configuration utility of NaftaPOS software (as a rule located in root folder of NaftaPOS software *Drive:\NaftaPOS\POSSet.exe*).

Go to tab "*General*" and set there configuration as follows:



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')

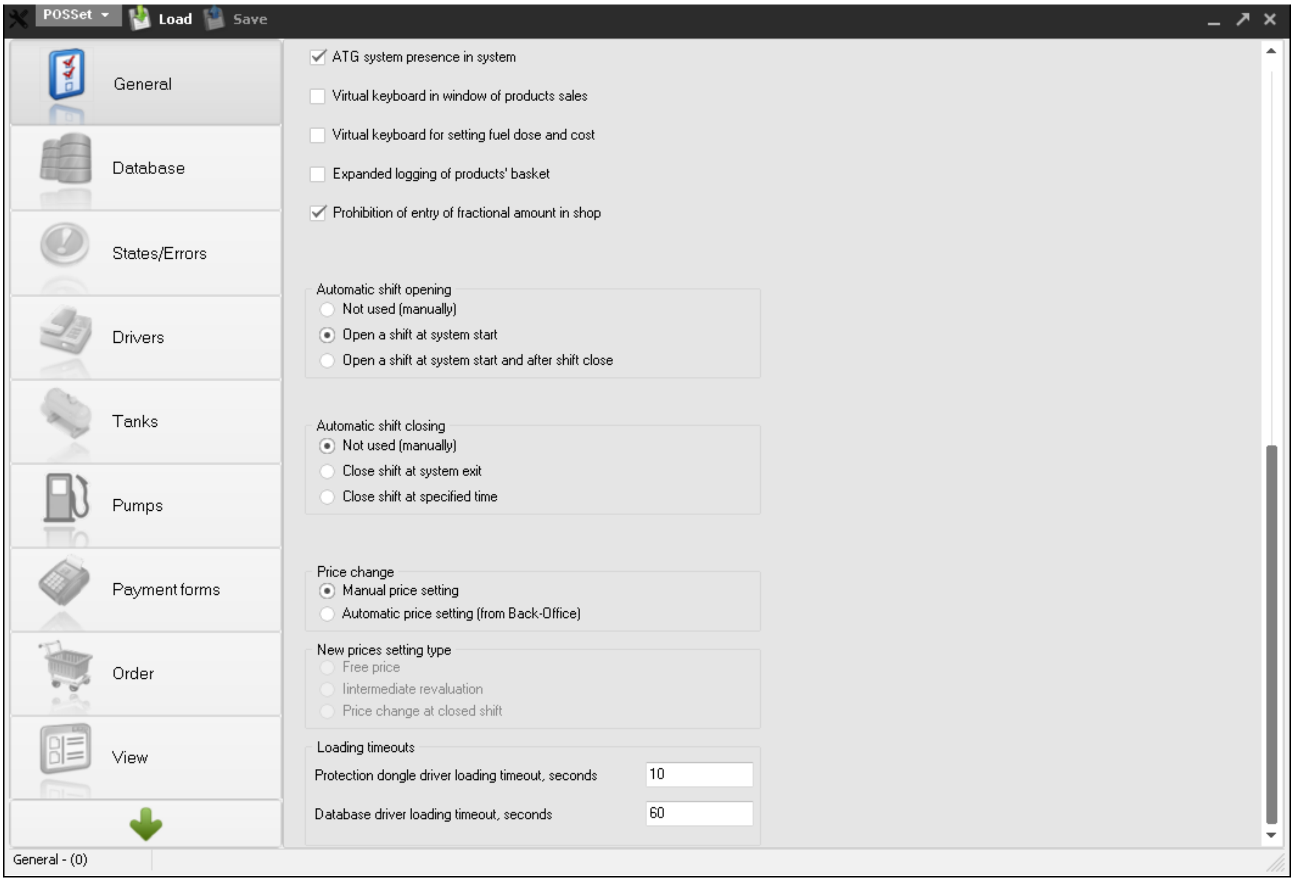
□ Restart after shift closing

☑ Permission to set book-keeping remains at opened shift

☑ Permission to set prices at opened shift

☑ Permission on dispensing during fuel reception

Checkbox "*ATG system presence in system*" should be checked



☑ ATG system 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

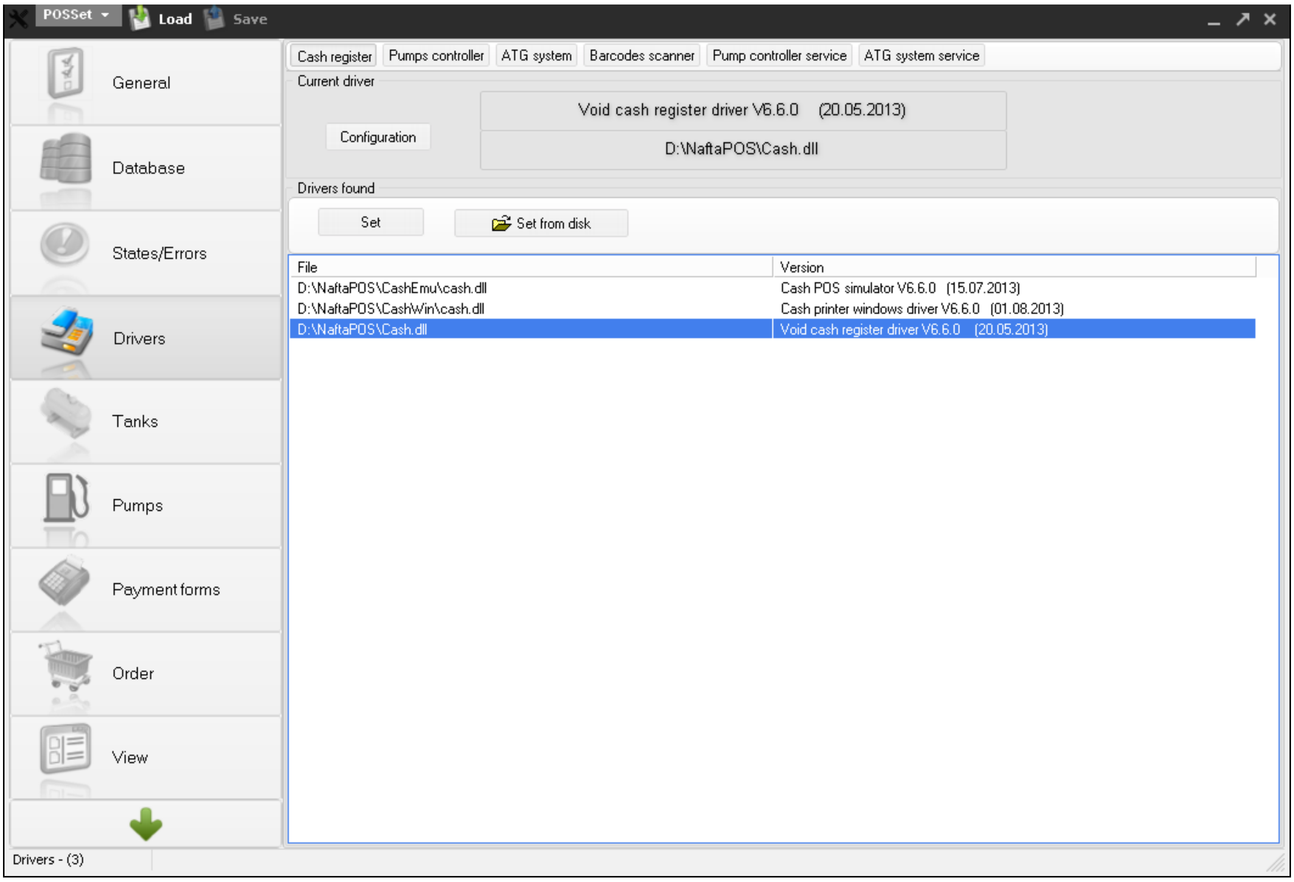
Price change
● Manual price setting
○ Automatic price setting (from Back-Office)

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

Loading timeouts

Protection dongle driver loading timeout, seconds	10
Database driver loading timeout, seconds	60

Go to tab "*Drivers*" and set for a cash register a driver "*Void cash register driver*":



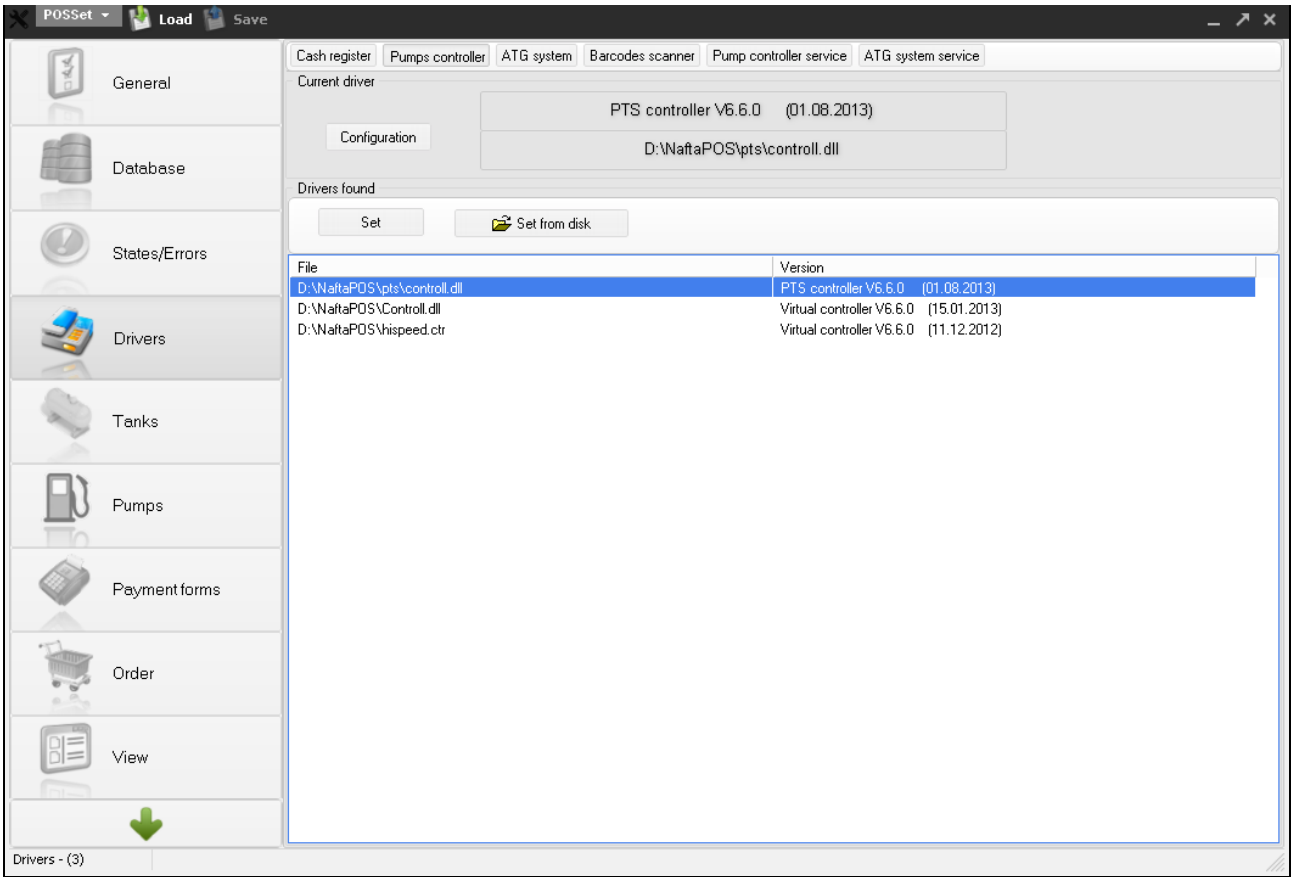
Current driver

Void cash register driver V6.6.0 (20.05.2013)
D:\NaftaPOS\Cash.dll

Drivers found

File	Version
D:\NaftaPOS\CashEmu\cash.dll	Cash POS simulator V6.6.0 (15.07.2013)
D:\NaftaPOS\CashWin\cash.dll	Cash printer windows driver V6.6.0 (01.08.2013)
D:\NaftaPOS\Cash.dll	Void cash register driver V6.6.0 (20.05.2013)

As a driver of "*Pumps controller*" select a "*PTS controller*" driver:



Current driver

PTS controller V6.6.0 (01.08.2013)

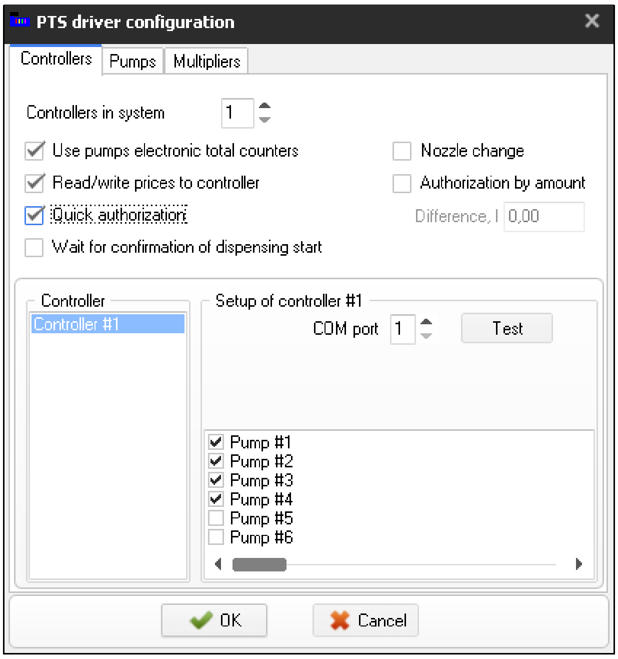
D:\NaftaPOS\pts\controll.dll

Drivers found

File	Version
D:\NaftaPOS\pts\controll.dll	PTS controller V6.6.0 (01.08.2013)
D:\NaftaPOS\Controll.dll	Virtual controller V6.6.0 (15.01.2013)
D:\NaftaPOS\hispeed.ctr	Virtual controller V6.6.0 (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 – 4
- Set checkboxes "*Use pumps electronic totalizers*", "*Read/write prices to controller*", "*Quick authorization*"



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

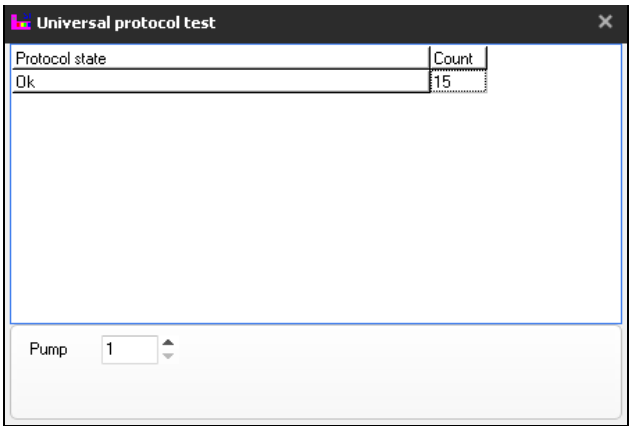
Controller: Controller #1

Setup of controller #1

COM port 1

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

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	15

Pump 1

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

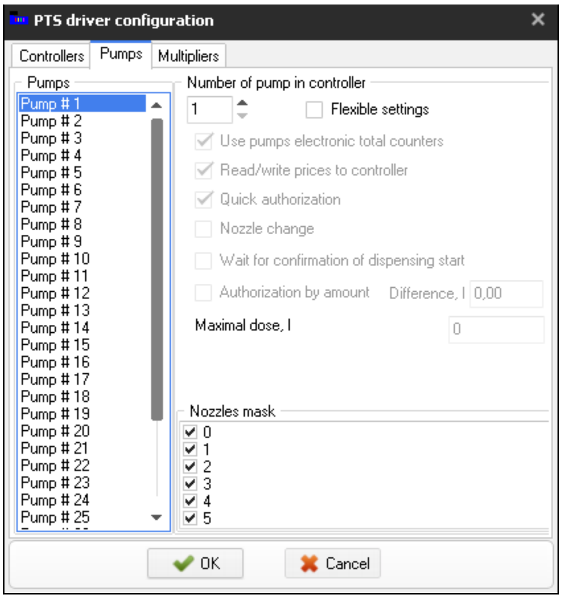
Pump #1 → 1

Pump #2 → 2

Pump #3 → 3

Pump #4 → 4

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


PTS driver configuration

Controllers | Pumps | Multipliers

Pumps: Pump # 1 (selected), 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: 1 ☐ Flexible settings

☑ Use pumps electronic total counters

☑ Read/write prices to controller

☑ Quick authorization

☐ Nozzle change

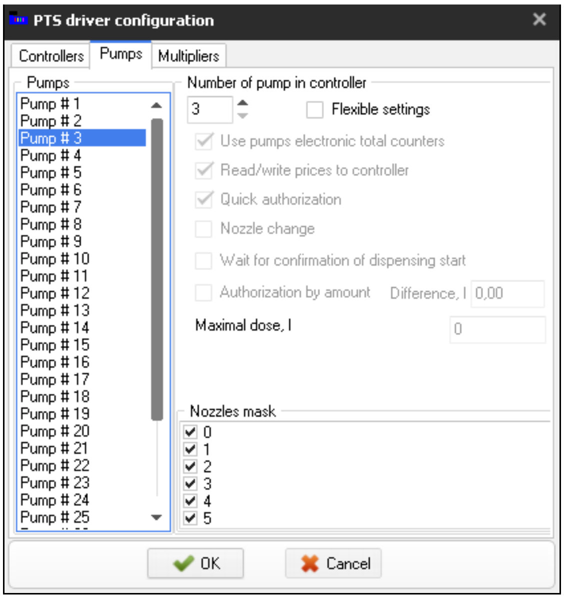
☐ Wait for confirmation of dispensing start

☐ Authorization by amount Difference, l 0,00

Maximal dose, l 0

Nozzles mask

☑ 0
☑ 1
☑ 2
☑ 3
☑ 4
☑ 5


PTS driver configuration

Controllers | Pumps | Multipliers

Pumps: Pump # 1, Pump # 2, Pump # 3 (selected), 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: 3 ☐ Flexible settings

☑ Use pumps electronic total counters

☑ Read/write prices to controller

☑ Quick authorization

☐ Nozzle change

☐ Wait for confirmation of dispensing start

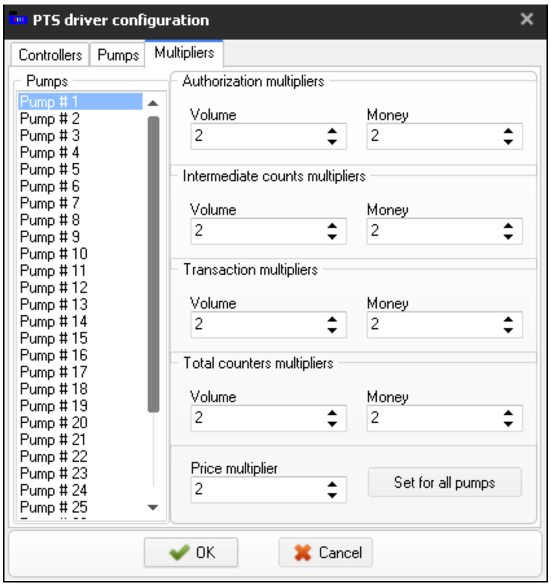
☐ Authorization by amount Difference, l 0,00

Maximal dose, l 0

Nozzles mask

☑ 0
☑ 1
☑ 2
☑ 3
☑ 4
☑ 5

All multipliers set equal to 2 digits on tab “*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

Authorization multipliers
Volume: 2
Money: 2

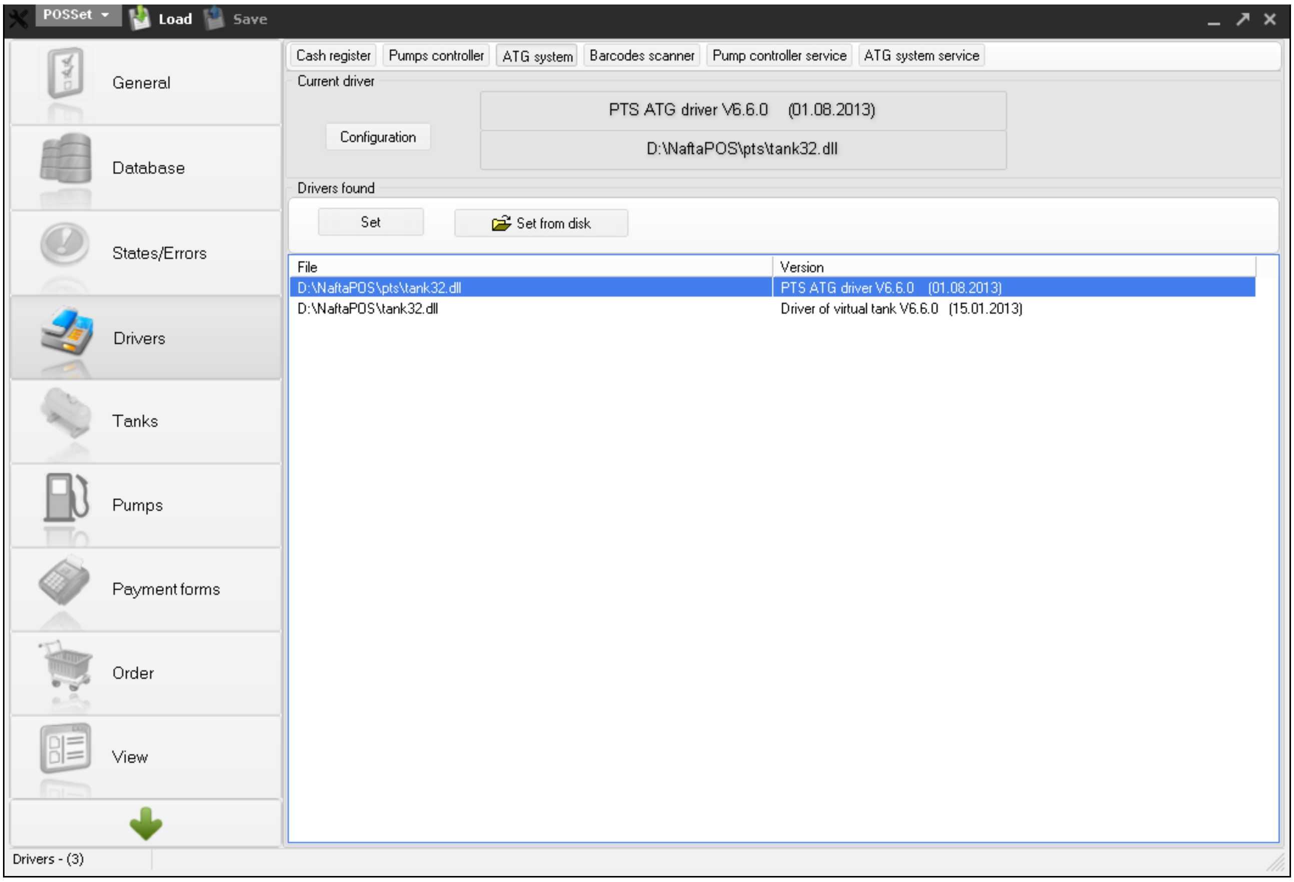
Intermediate counts multipliers
Volume: 2
Money: 2

Transaction multipliers
Volume: 2
Money: 2

Total counters multipliers
Volume: 2
Money: 2

Price multiplier: 2

As a driver of “*Tanks*” select a “*PTS ATG driver*”:



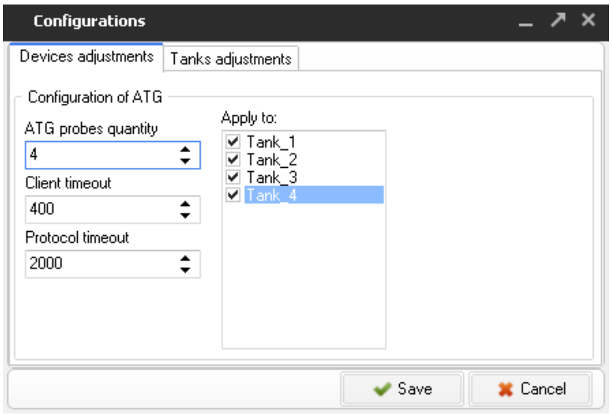
Current driver
PTS ATG driver V6.6.0 (01.08.2013)
D:\NaftaPOS\pts\tank32.dll

Drivers found

File	Version
D:\NaftaPOS\pts\tank32.dll	PTS ATG driver V6.6.0 (01.08.2013)
D:\NaftaPOS\tank32.dll	Driver of virtual tank V6.6.0 (15.01.2013)

Drivers - (3)

Go to configuration of “*PTS ATG driver*” and set there the following configuration:



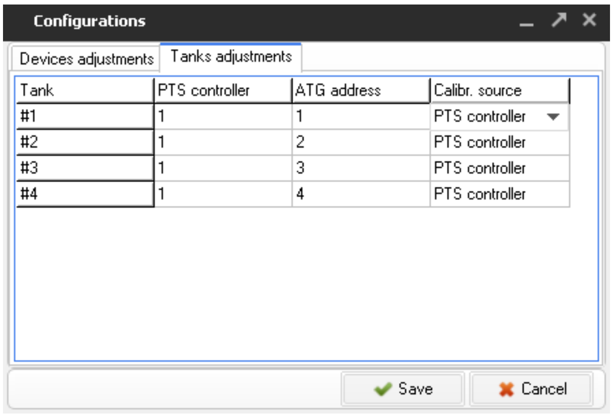
Configuration of ATG

ATG probes quantity: 4

Client timeout: 400

Protocol timeout: 2000

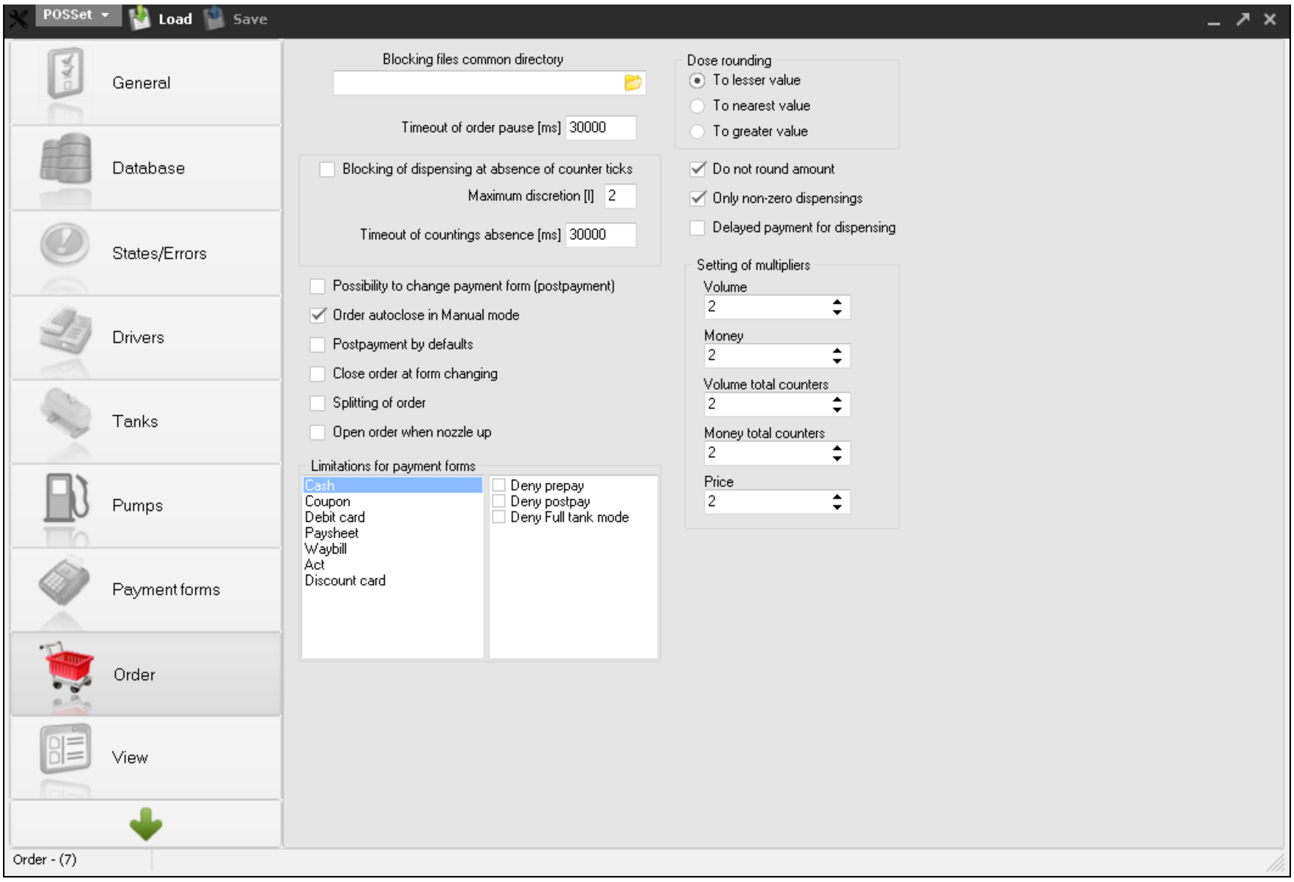
Apply to:
- ☑ Tank_1
- ☑ Tank_2
- ☑ Tank_3
- ☑ Tank_4



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

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

□ 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	□ Deny prepay
Coupon	□ Deny postpay
Debit card	□ Deny Full tank mode
Paysheet	
Waybill	
Act	
Discount card	

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

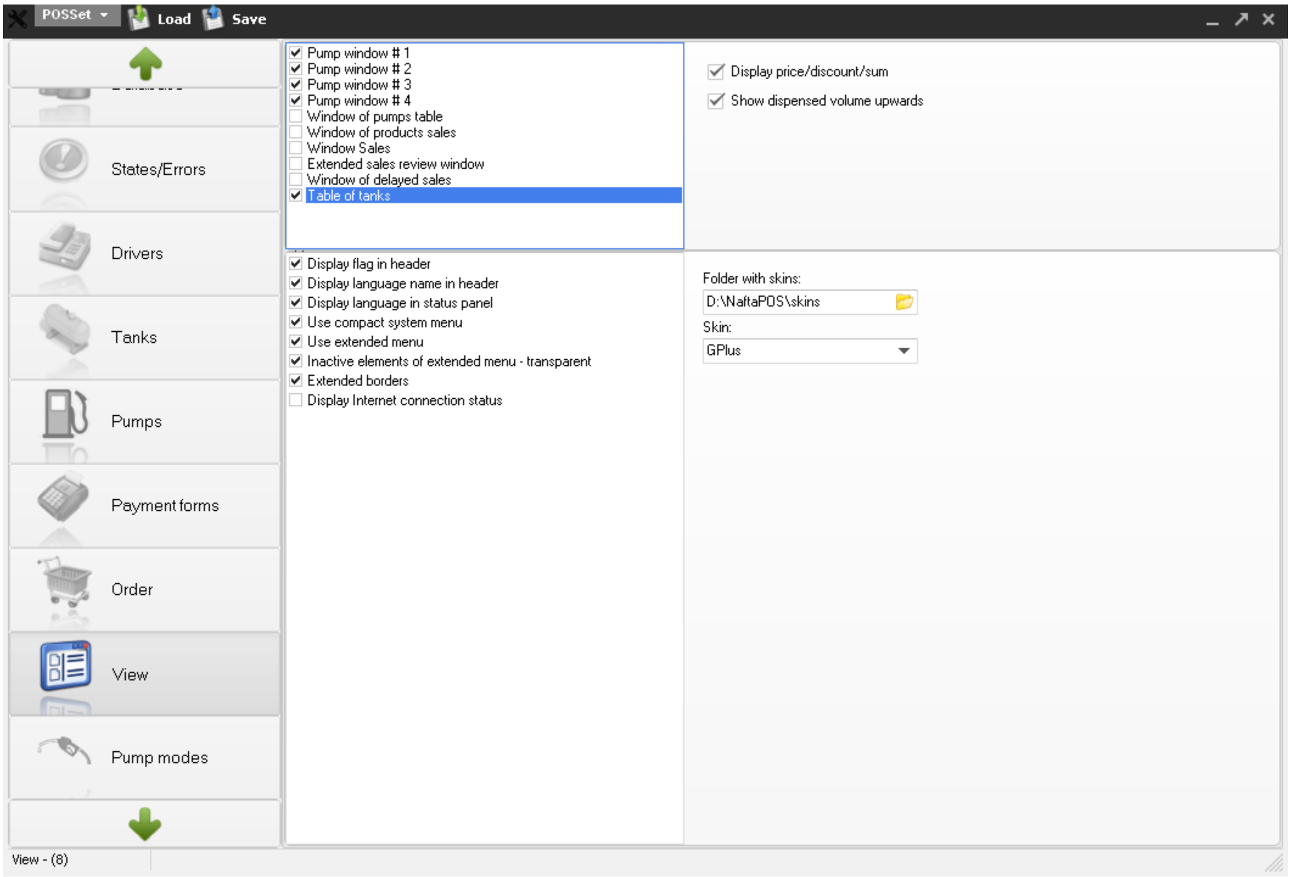
Money: 2

Volume total counters: 2

Money total counters: 2

Price: 2

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



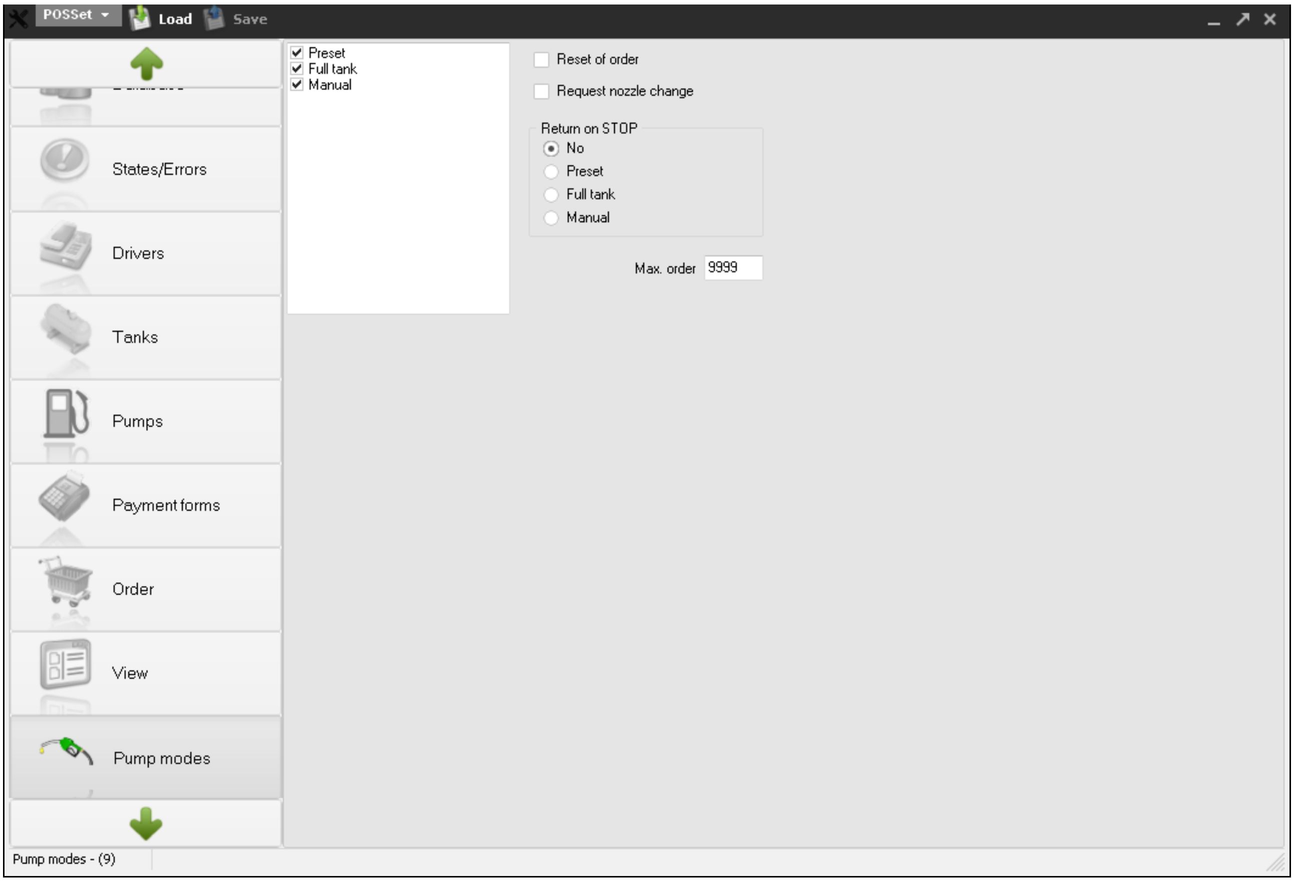
☑ Pump window # 1
☑ Pump window # 2
☑ Pump window # 3
☑ Pump window # 4
□ Window of pumps table
□ Window of products sales
□ Window Sales
□ Extended sales review window
□ Window of delayed sales
☑ Table of tanks

☑ Display price/discount/sum
☑ Show dispensed volume upwards

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

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

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 NaftaPOS software is configured to work.

Make sure that Guardant USB dongle is inserted into USB-port of PC and LED on it is shining. Run Drive:\NaftaPOS\NaftaPOS.exe.

On the main panel there will be pump icons and table of tanks located (they were selected to be displayed in configuration), these windows can be dragged along the main panel and their sizes can be changes.

You can provide control over the pumps and monitor states of every controlled parameter of each tank:

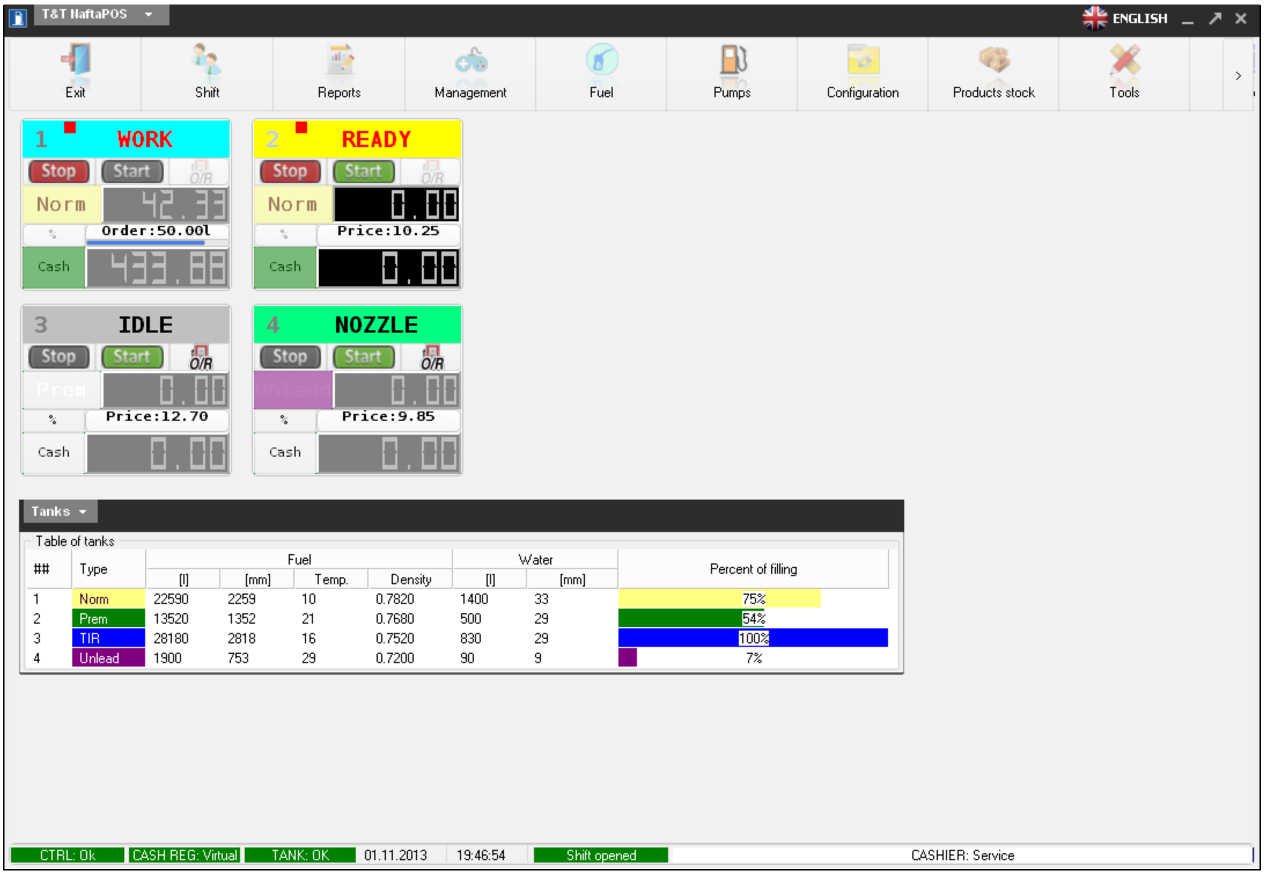
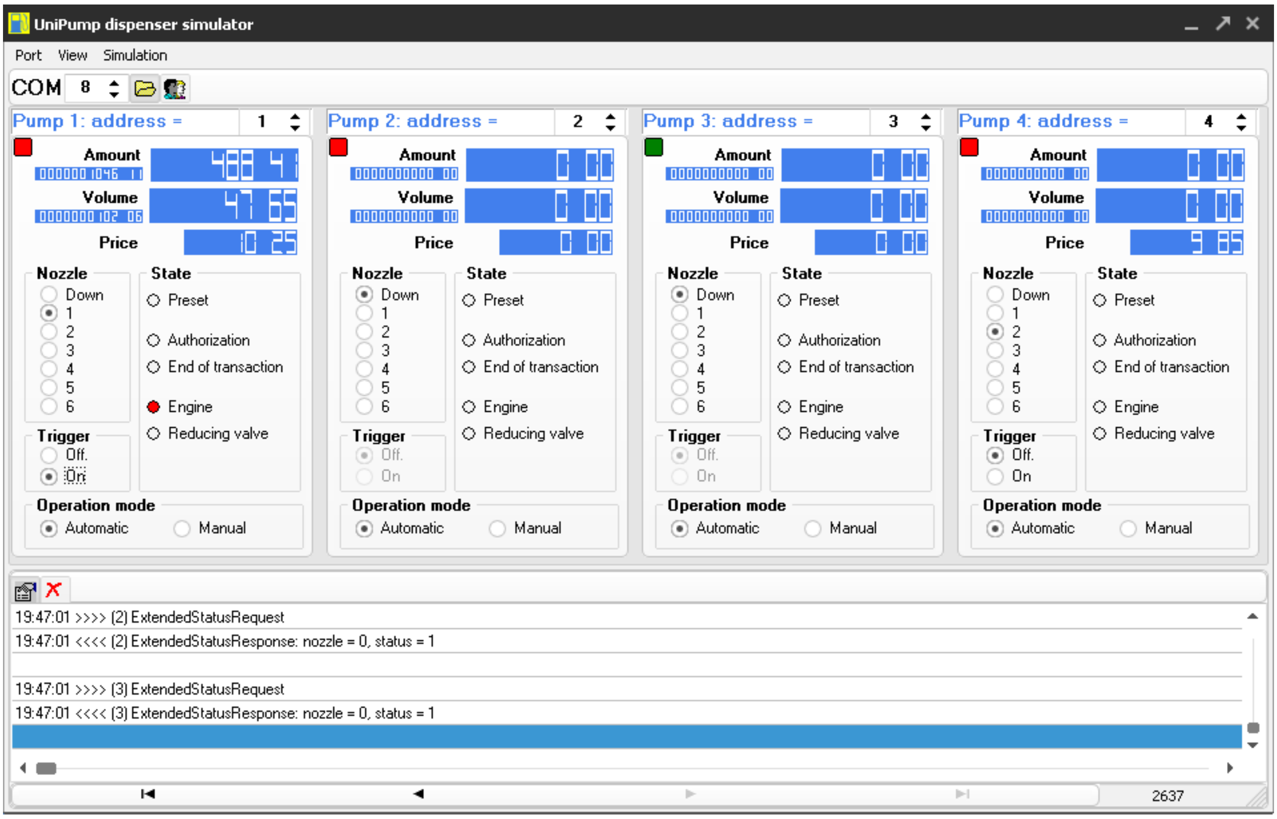


Table of tanks

##	Type	Fuel				Water		Percent of filling
		[l]	[mm]	Temp.	Density	[l]	[mm]	
1	Norm	22590	2259	10	0.7820	1400	33	75%
2	Prem	13520	1352	21	0.7680	500	29	54%
3	TIR	28180	2818	16	0.7520	830	29	100%
4	Unlead	1900	753	29	0.7200	90	9	7%

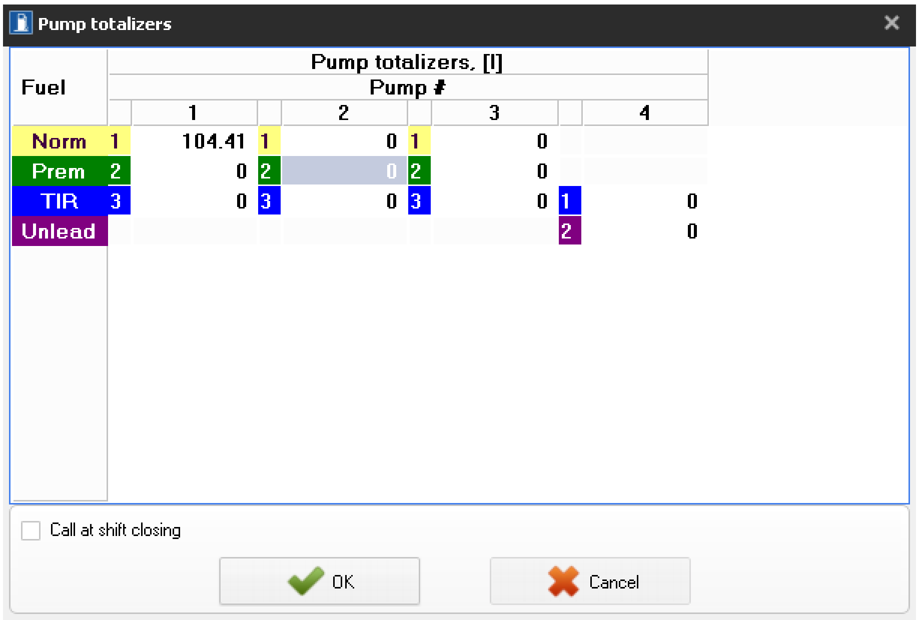
CTRL: Ok | CASH REG: Virtual | TANK: OK | 01.11.2013 | 19:46:54 | Shift opened | CASHIER: Service

You can play with dispensers and their nozzles simulating real fuel dispenser activity.



```
19:47:01 >>>> (2) ExtendedStatusRequest
19:47:01 <<<< (2) ExtendedStatusResponse: nozzle = 0, status = 1

19:47:01 >>>> (3) ExtendedStatusRequest
19:47:01 <<<< (3) ExtendedStatusResponse: nozzle = 0, status = 1
```

NaftaPOS will read pump total counters after each dispensing is performed. Total counter values from dispensers can be read using a main menu item “*Pumps*” → “*Totalizers*”:

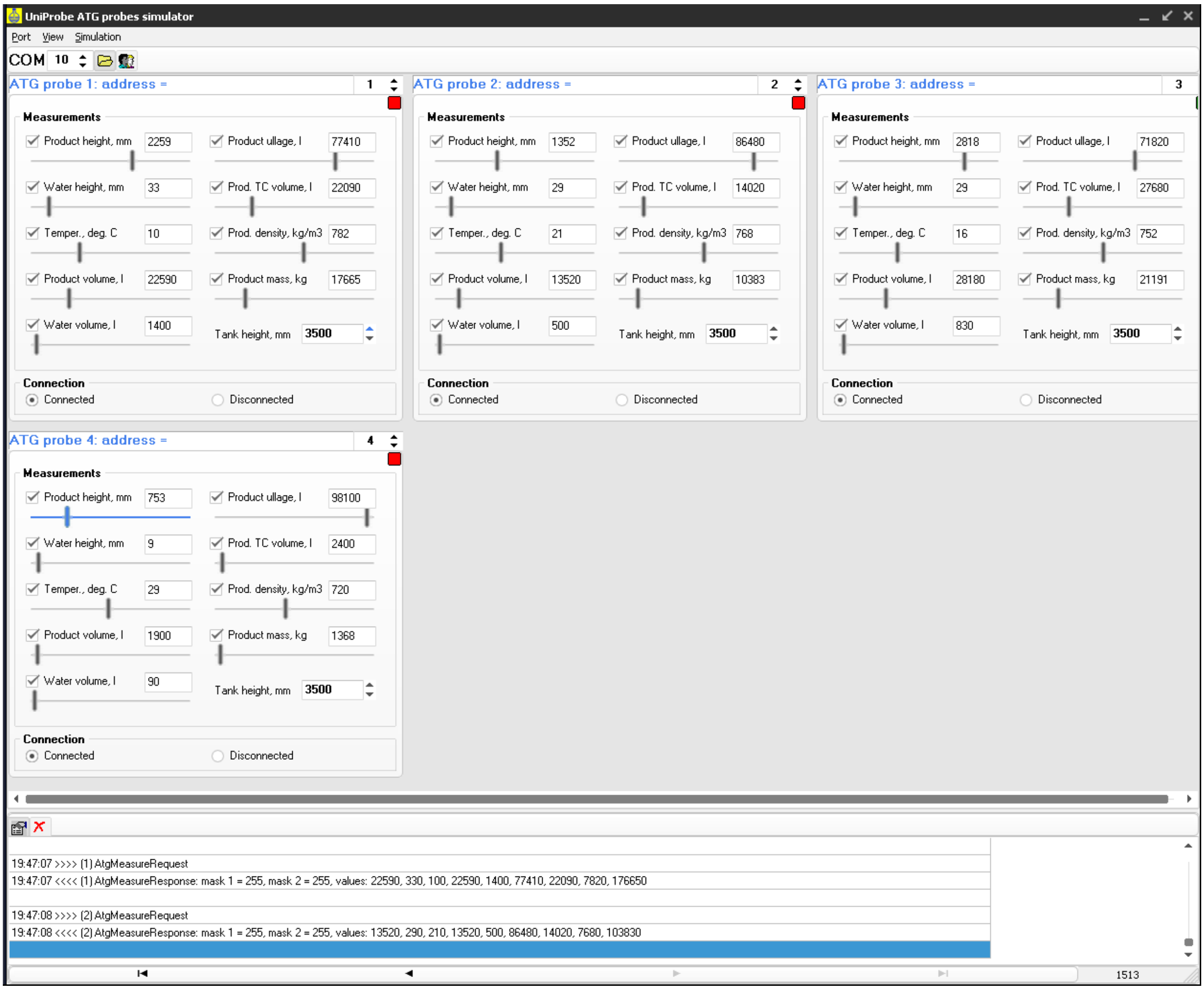


Pump totalizers

Fuel	Pump totalizers, [l]							
	Pump #							
		1		2		3		4
Norm	1	104.41	1	0	1	0		
Prem	2	0	2	0	2	0		
TIR	3	0	3	0	3	0	1	0
Unlead							2	0

□ Call at shift closing

You can change values measured by ATG systems and these changes will be displayed in NaftaPOS:



COM 10

ATG probe 1: address = 1

Measurements

☑ Product height, mm	2259	☑ Product ullage, l	77410
☑ Water height, mm	33	☑ Prod. TC volume, l	22090
☑ Temper., deg. C	10	☑ Prod. density, kg/m3	782
☑ Product volume, l	22590	☑ Product mass, kg	17665
☑ Water volume, l	1400	Tank height, mm	3500

Connection
● Connected ○ Disconnected

ATG probe 2: address = 2

Measurements

☑ Product height, mm	1352	☑ Product ullage, l	86480
☑ Water height, mm	29	☑ Prod. TC volume, l	14020
☑ Temper., deg. C	21	☑ Prod. density, kg/m3	768
☑ Product volume, l	13520	☑ Product mass, kg	10383
☑ Water volume, l	500	Tank height, mm	3500

Connection
● Connected ○ Disconnected

ATG probe 3: address = 3

Measurements

☑ Product height, mm	2818	☑ Product ullage, l	71820
☑ Water height, mm	29	☑ Prod. TC volume, l	27680
☑ Temper., deg. C	16	☑ Prod. density, kg/m3	752
☑ Product volume, l	28180	☑ Product mass, kg	21191
☑ Water volume, l	830	Tank height, mm	3500

Connection
● Connected ○ Disconnected

ATG probe 4: address = 4

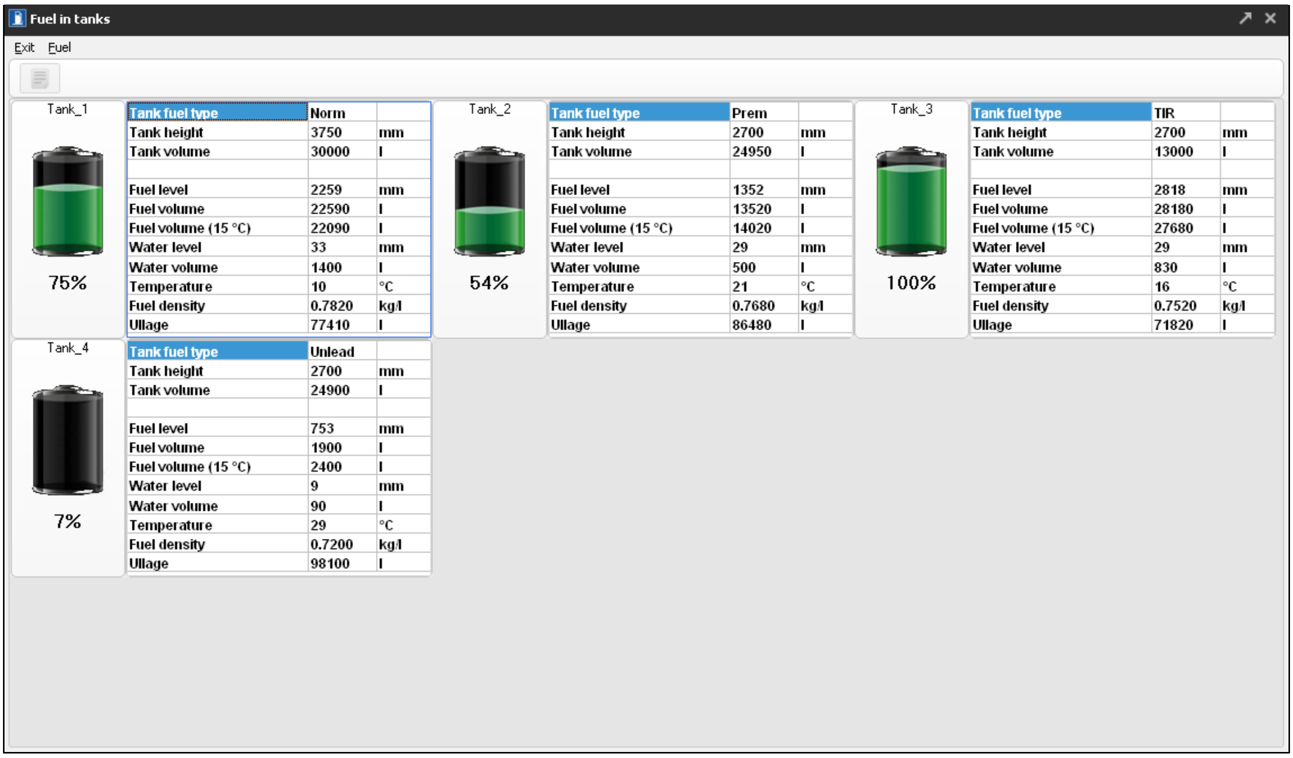
Measurements

☑ Product height, mm	753	☑ Product ullage, l	98100
☑ Water height, mm	9	☑ Prod. TC volume, l	2400
☑ Temper., deg. C	29	☑ Prod. density, kg/m3	720
☑ Product volume, l	1900	☑ Product mass, kg	1368
☑ Water volume, l	90	Tank height, mm	3500

Connection
● Connected ○ Disconnected

```
19:47:07 >>>> (1) AtgMeasureRequest
19:47:07 <<<< (1) AtgMeasureResponse: mask 1 = 255, mask 2 = 255, values: 22590, 330, 100, 22590, 1400, 77410, 22090, 7820, 176650

19:47:08 >>>> (2) AtgMeasureRequest
19:47:08 <<<< (2) AtgMeasureResponse: mask 1 = 255, mask 2 = 255, values: 13520, 290, 210, 13520, 500, 86480, 14020, 7680, 103830
```

NaftaPOS displays measurements of ATG systems in tanks table and also in a separate window (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):



Tank_1

75%

Tank fuel type	Norm	
Tank height	3750	mm
Tank volume	30000	l
Fuel level	2259	mm
Fuel volume	22590	l
Fuel volume (15 °C)	22090	l
Water level	33	mm
Water volume	1400	l
Temperature	10	°C
Fuel density	0.7820	kg/l
Ullage	77410	l

Tank_2

54%

Tank fuel type	Prem	
Tank height	2700	mm
Tank volume	24950	l
Fuel level	1352	mm
Fuel volume	13520	l
Fuel volume (15 °C)	14020	l
Water level	29	mm
Water volume	500	l
Temperature	21	°C
Fuel density	0.7680	kg/l
Ullage	86480	l

Tank_3

100%

Tank fuel type	TIR	
Tank height	2700	mm
Tank volume	13000	l
Fuel level	2818	mm
Fuel volume	28180	l
Fuel volume (15 °C)	27680	l
Water level	29	mm
Water volume	830	l
Temperature	16	°C
Fuel density	0.7520	kg/l
Ullage	71820	l

Tank_4

7%

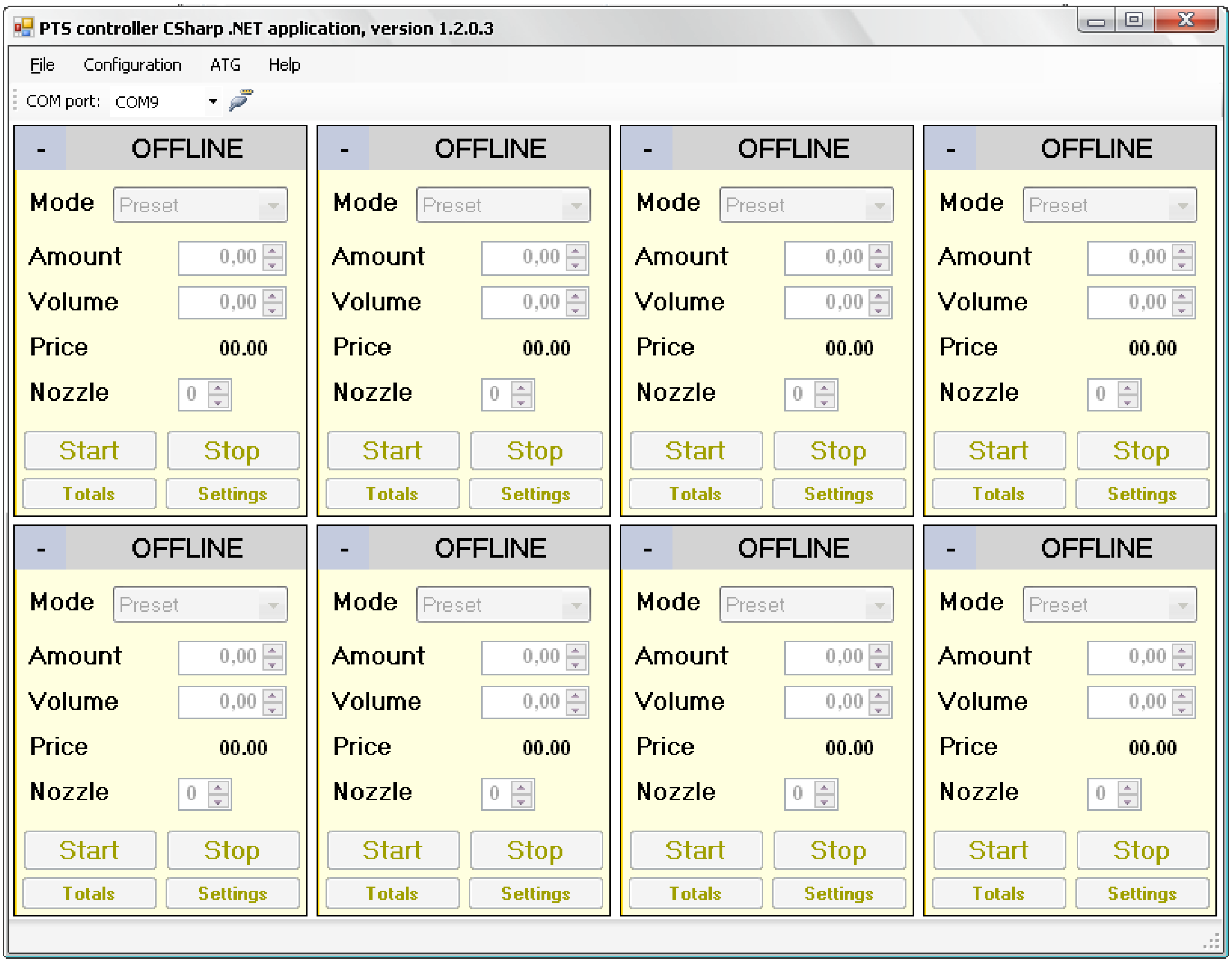
Tank fuel type	Unlead	
Tank height	2700	mm
Tank volume	24900	l
Fuel level	753	mm
Fuel volume	1900	l
Fuel volume (15 °C)	2400	l
Water level	9	mm
Water volume	90	l
Temperature	29	°C
Fuel density	0.7200	kg/l
Ullage	98100	l

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

Step 12. Configuration of PTS controller .NET application for control over pumps and ATG probes

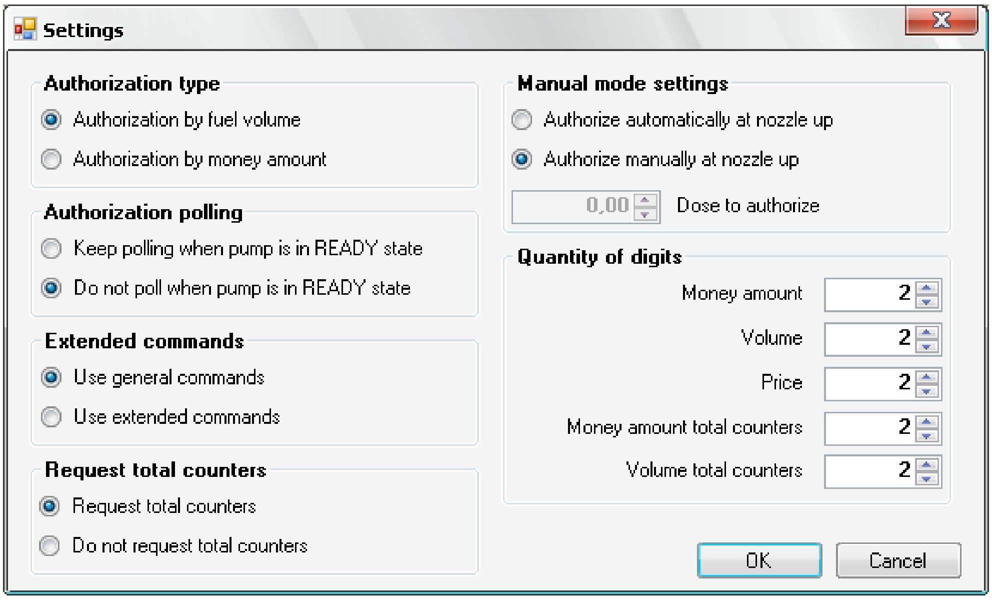
The open-source application written in C# and VB.NET is a good place to start development of new control software for petrol stations. In this section we will see how it is easy to configure and run it.

Compile "*PTS application*" project or simply run *PtsApplication.exe* (application is located in PTS SDK software kit and is located in API TOOLS folder for C# and VB.NET languages):



COM port: COM9

Go to main menu item "*Configuration*" → "*Settings*" and set configuration as shown below:



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

0,00 Dose to authorize

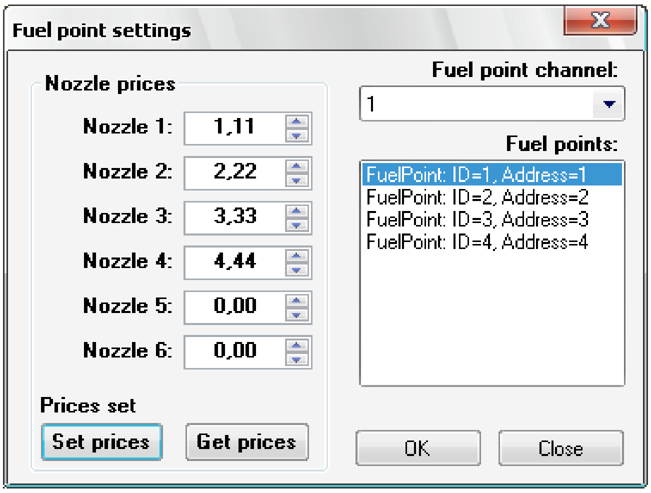
Quantity of digits

Money amount	2
Volume	2
Price	2
Money amount total counters	2
Volume total counters	2

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 #3 → FuelPoint ID = 3, Address = 3
- Fuel point icon #4 → FuelPoint ID = 4, Address = 4

Also here set prices for each of the nozzles:



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

Prices set

Set prices Get prices

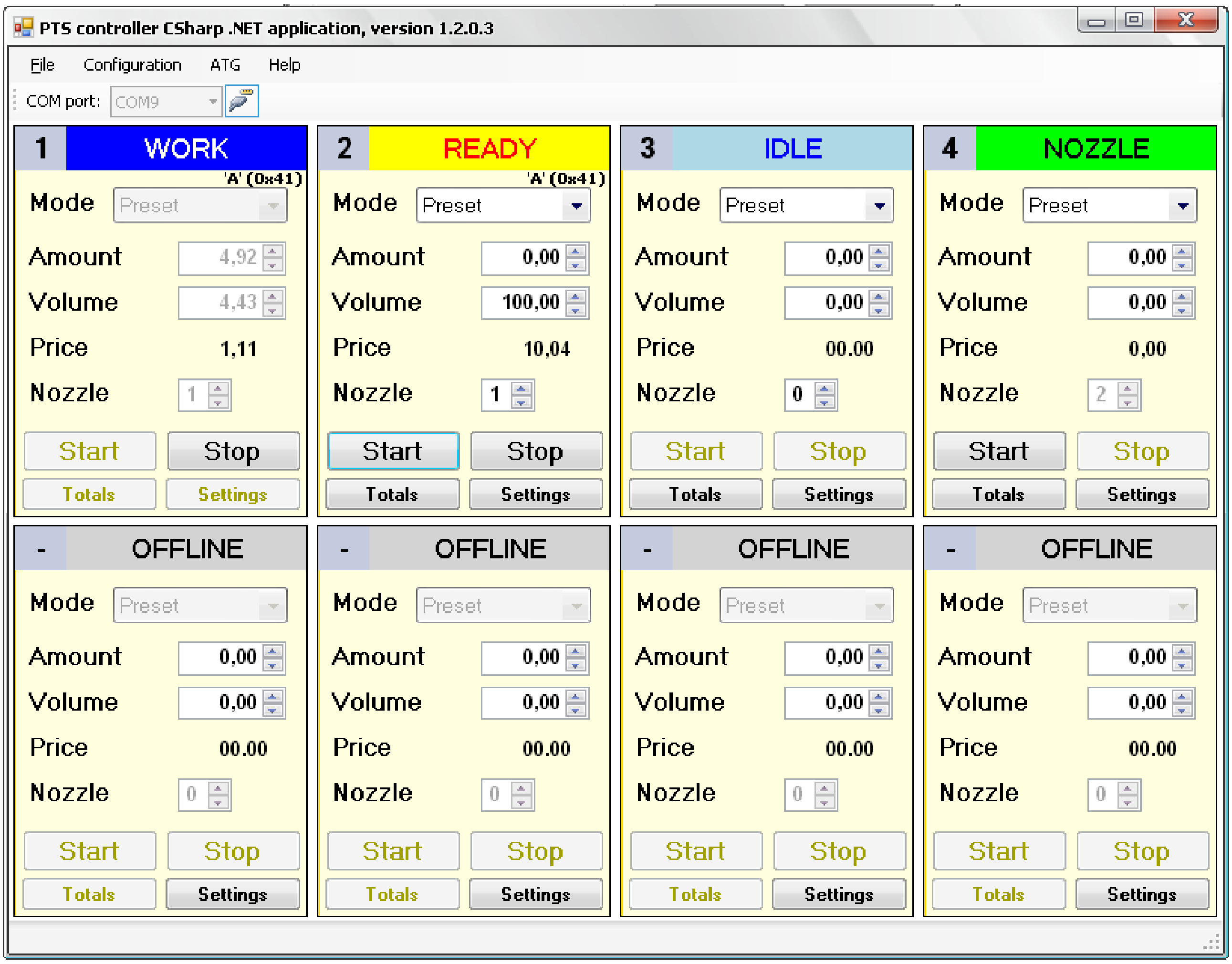
Fuel point channel: 1

Fuel points:

FuelPoint: ID=1, Address=1
FuelPoint: ID=2, Address=2
FuelPoint: ID=3, Address=3
FuelPoint: ID=4, Address=4

OK Close

Now the main form of the application allows to provide control over pumps in pump software simulator (see configuration of the pumps simulator in step 7 above):



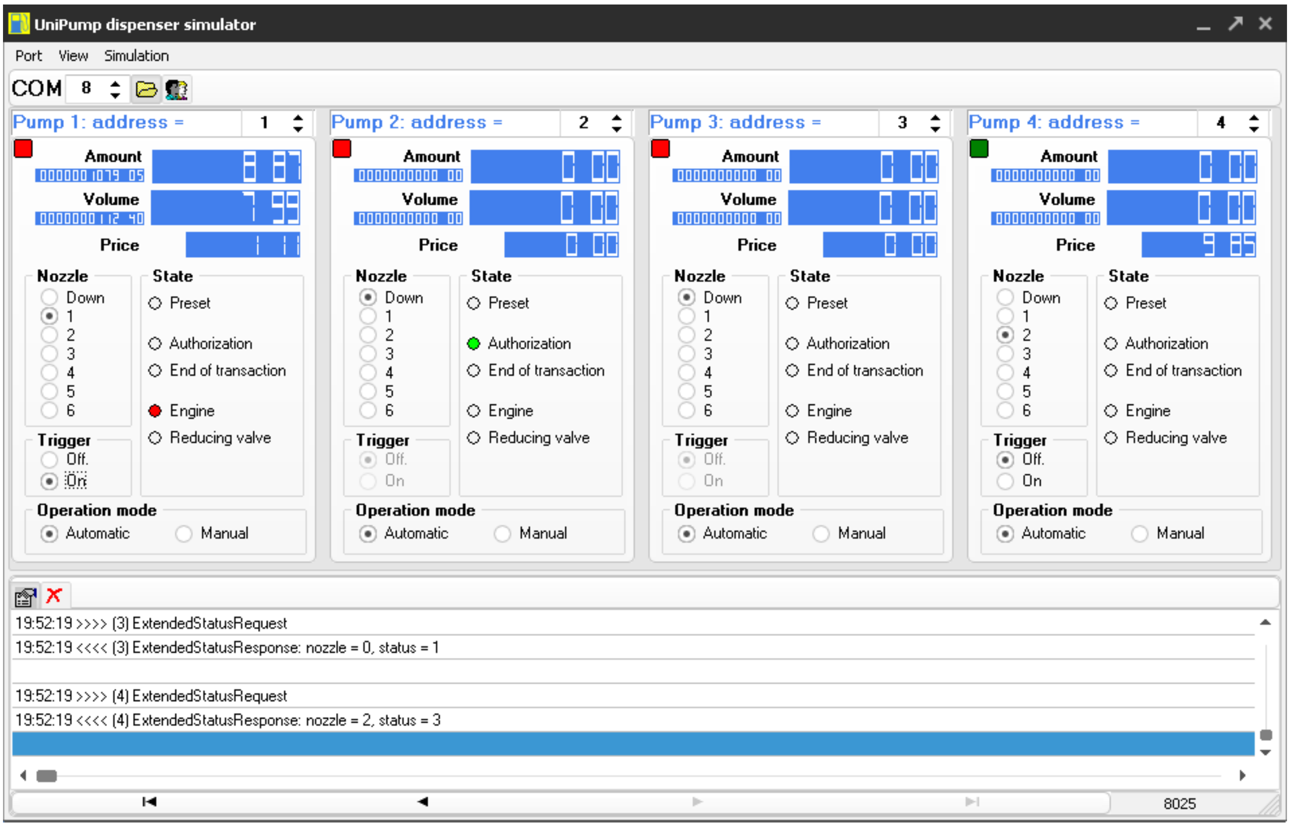
PTS controller CSharp .NET application, version 1.2.0.3

File Configuration ATG Help

COM port: COM9

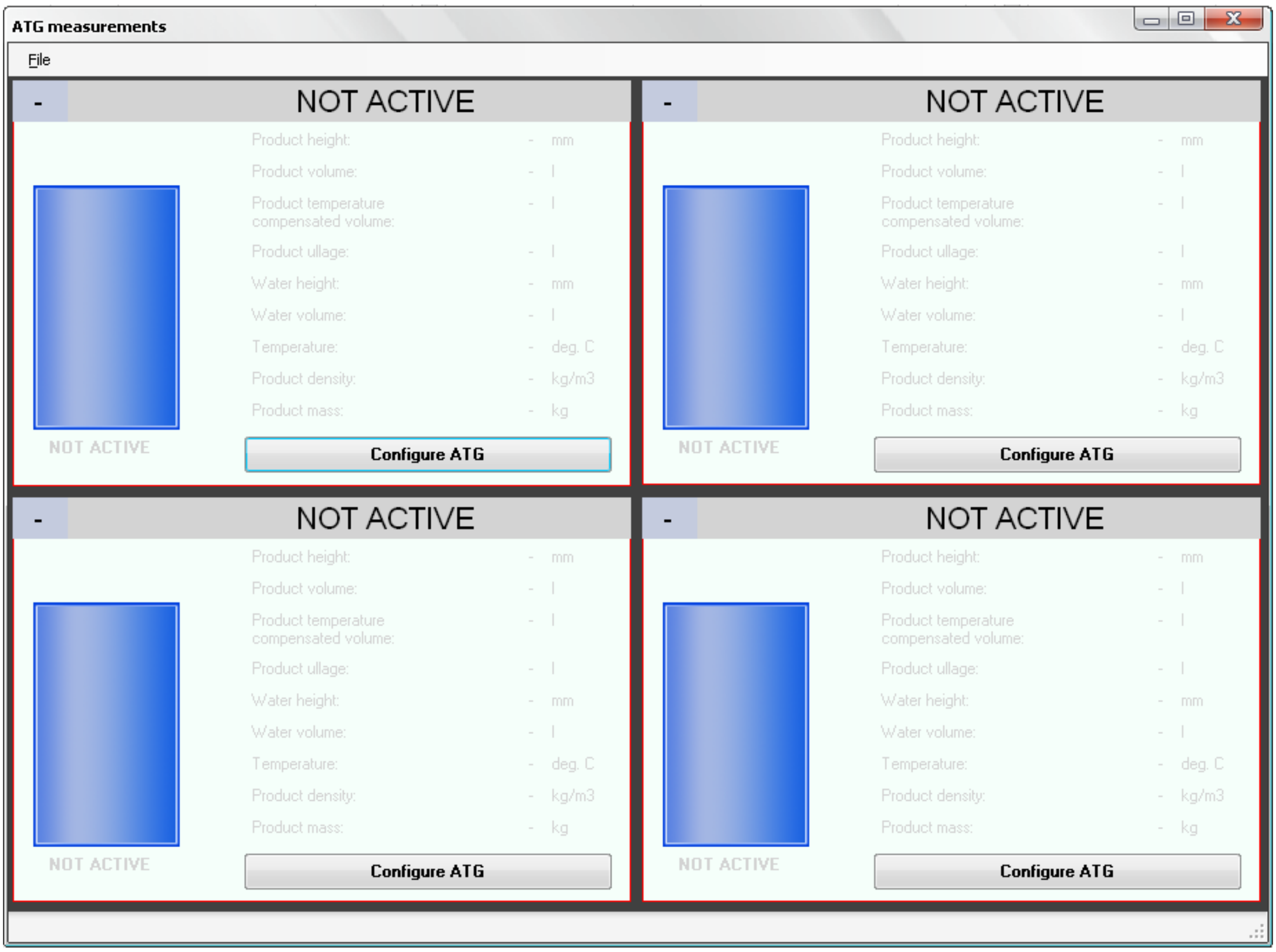
	1	2	3	4
Status	WORK	READY	IDLE	NOZZLE
	'A' (0x41)	'A' (0x41)		
Mode	Preset	Preset	Preset	Preset
Amount	4,92	0,00	0,00	0,00
Volume	4,43	100,00	0,00	0,00
Price	1,11	10,04	00.00	0,00
Nozzle	1	1	0	2

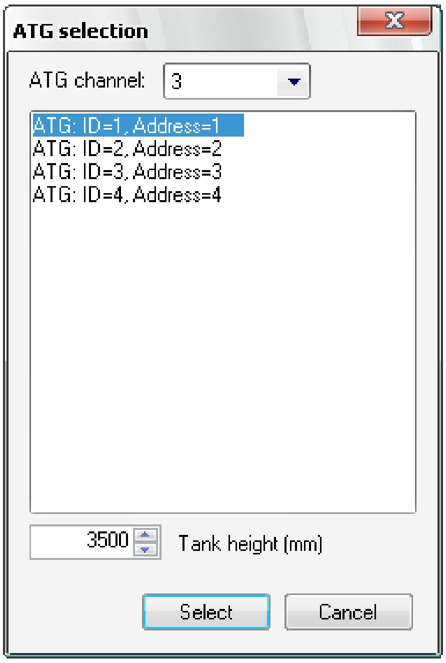
	-	-	-	-
Status	OFFLINE	OFFLINE	OFFLINE	OFFLINE
Mode	Preset	Preset	Preset	Preset
Amount	0,00	0,00	0,00	0,00
Volume	0,00	0,00	0,00	0,00
Price	00.00	00.00	00.00	00.00
Nozzle	0	0	0	0



```
19:52:19 >>>> (3) ExtendedStatusRequest
19:52:19 <<<< (3) ExtendedStatusResponse: nozzle = 0, status = 1

19:52:19 >>>> (4) ExtendedStatusRequest
19:52:19 <<<< (4) ExtendedStatusResponse: nozzle = 2, status = 3
```

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:





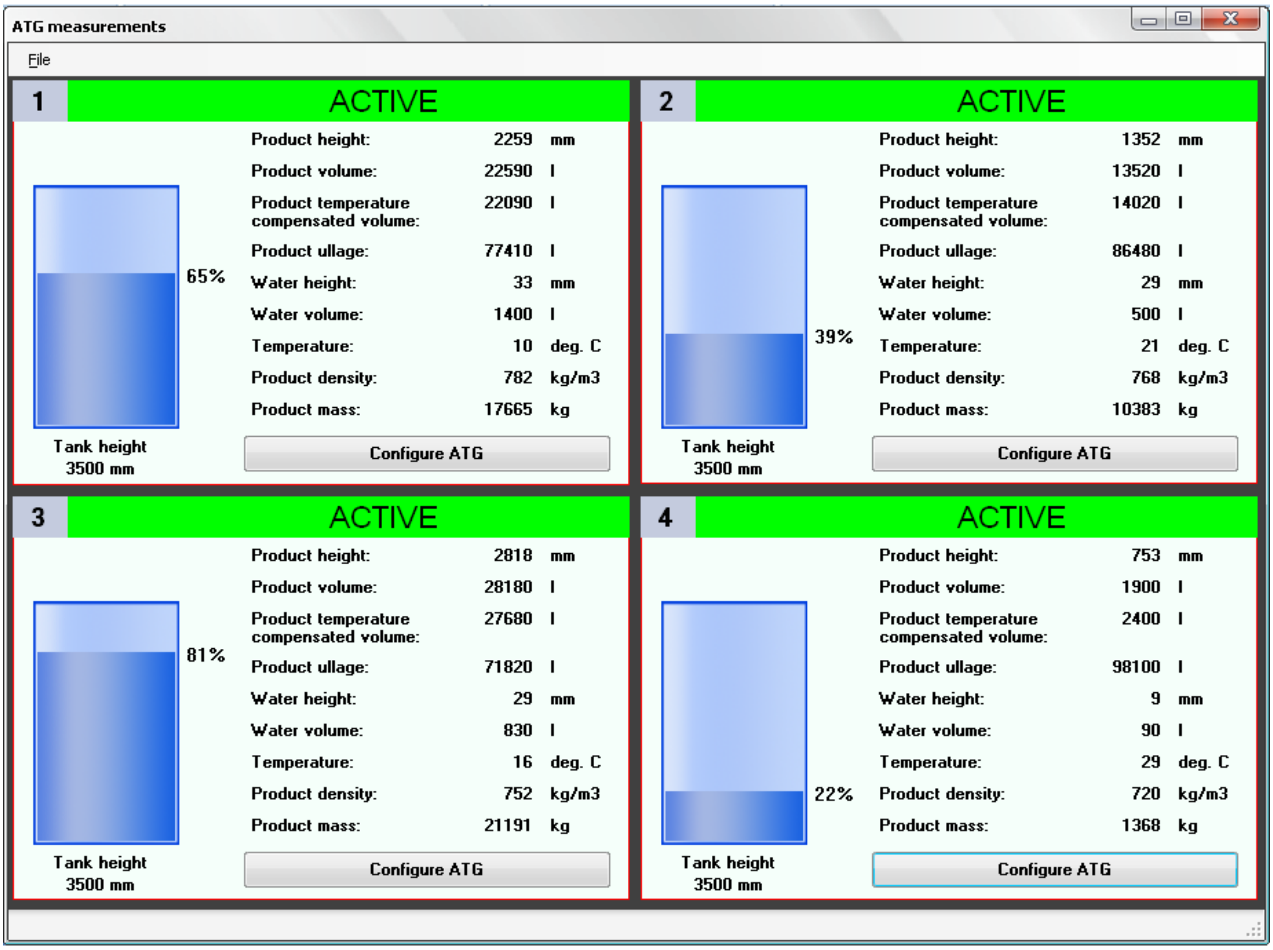
ATG selection

ATG channel: 3

ATG: ID=1, Address=1
ATG: ID=2, Address=2
ATG: ID=3, Address=3
ATG: ID=4, Address=4

3500 Tank height (mm)

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



ATG measurements

1 ACTIVE

65%

Tank height 3500 mm

Product height:	2259	mm
Product volume:	22590	l
Product temperature compensated volume:	22090	l
Product ullage:	77410	l
Water height:	33	mm
Water volume:	1400	l
Temperature:	10	deg. C
Product density:	782	kg/m3
Product mass:	17665	kg

2 ACTIVE

39%

Tank height 3500 mm

Product height:	1352	mm
Product volume:	13520	l
Product temperature compensated volume:	14020	l
Product ullage:	86480	l
Water height:	29	mm
Water volume:	500	l
Temperature:	21	deg. C
Product density:	768	kg/m3
Product mass:	10383	kg

3 ACTIVE

81%

Tank height 3500 mm

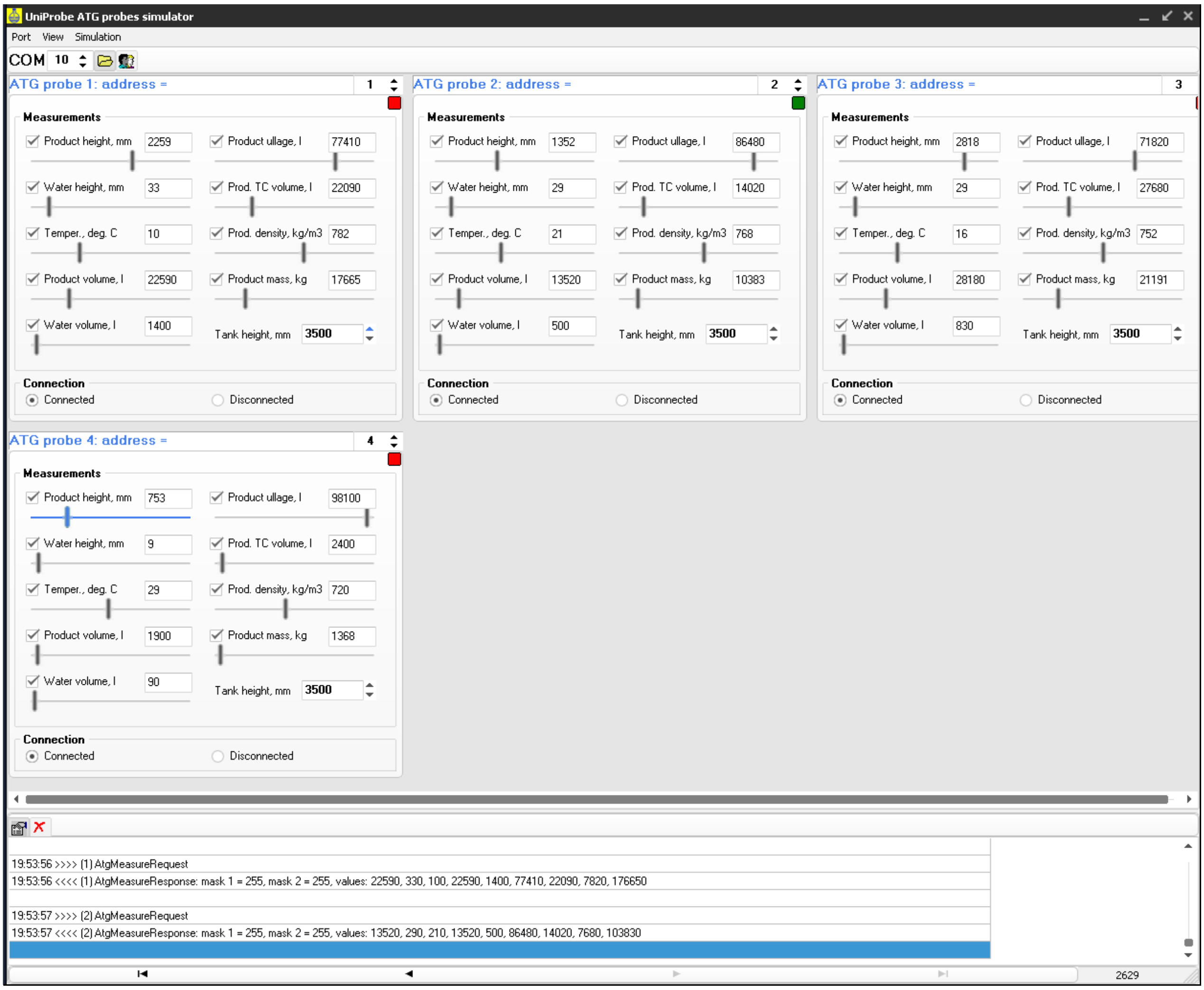
Product height:	2818	mm
Product volume:	28180	l
Product temperature compensated volume:	27680	l
Product ullage:	71820	l
Water height:	29	mm
Water volume:	830	l
Temperature:	16	deg. C
Product density:	752	kg/m3
Product mass:	21191	kg

4 ACTIVE

22%

Tank height 3500 mm

Product height:	753	mm
Product volume:	1900	l
Product temperature compensated volume:	2400	l
Product ullage:	98100	l
Water height:	9	mm
Water volume:	90	l
Temperature:	29	deg. C
Product density:	720	kg/m3
Product mass:	1368	kg



COM 10

ATG probe 1: address = 1

Measurements

- ☑ Product height, mm 2259
- ☑ Product ullage, l 77410
- ☑ Water height, mm 33
- ☑ Prod. TC volume, l 22090
- ☑ Temper., deg. C 10
- ☑ Prod. density, kg/m3 782
- ☑ Product volume, l 22590
- ☑ Product mass, kg 17665
- ☑ Water volume, l 1400
- Tank height, mm 3500

Connection: ● Connected ○ Disconnected

ATG probe 2: address = 2

Measurements

- ☑ Product height, mm 1352
- ☑ Product ullage, l 86480
- ☑ Water height, mm 29
- ☑ Prod. TC volume, l 14020
- ☑ Temper., deg. C 21
- ☑ Prod. density, kg/m3 768
- ☑ Product volume, l 13520
- ☑ Product mass, kg 10383
- ☑ Water volume, l 500
- Tank height, mm 3500

Connection: ● Connected ○ Disconnected

ATG probe 3: address = 3

Measurements

- ☑ Product height, mm 2818
- ☑ Product ullage, l 71820
- ☑ Water height, mm 29
- ☑ Prod. TC volume, l 27680
- ☑ Temper., deg. C 16
- ☑ Prod. density, kg/m3 752
- ☑ Product volume, l 28180
- ☑ Product mass, kg 21191
- ☑ Water volume, l 830
- Tank height, mm 3500

Connection: ● Connected ○ Disconnected

ATG probe 4: address = 4

Measurements

- ☑ Product height, mm 753
- ☑ Product ullage, l 98100
- ☑ Water height, mm 9
- ☑ Prod. TC volume, l 2400
- ☑ Temper., deg. C 29
- ☑ Prod. density, kg/m3 720
- ☑ Product volume, l 1900
- ☑ Product mass, kg 1368
- ☑ Water volume, l 90
- Tank height, mm 3500

Connection: ● Connected ○ Disconnected

```
19:53:56 >>>> (1) AtgMeasureRequest
19:53:56 <<<< (1) AtgMeasureResponse: mask 1 = 255, mask 2 = 255, values: 22590, 330, 100, 22590, 1400, 77410, 22090, 7820, 176650

19:53:57 >>>> (2) AtgMeasureRequest
19:53:57 <<<< (2) AtgMeasureResponse: mask 1 = 255, mask 2 = 255, values: 13520, 290, 210, 13520, 500, 86480, 14020, 7680, 103830
```

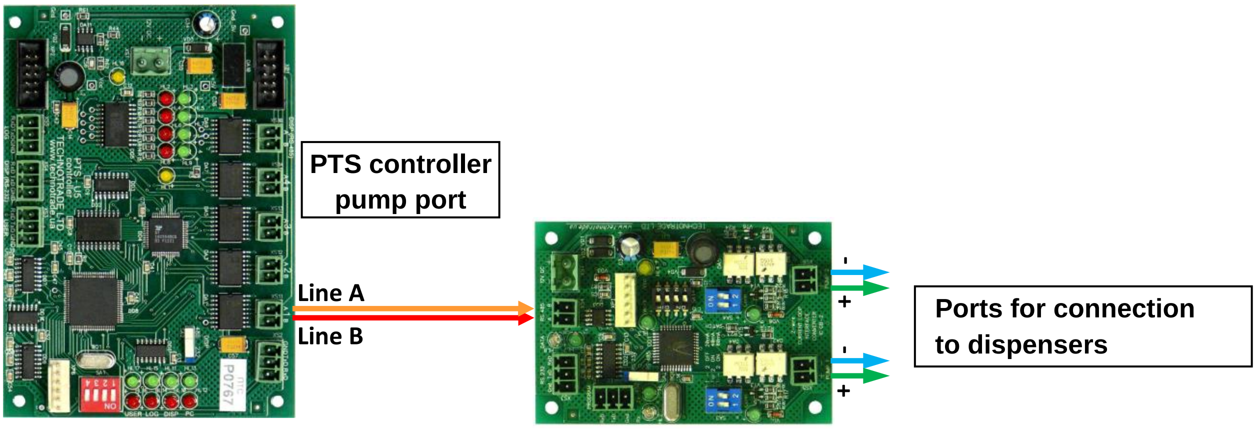
Now the PTS application correctly works with pumps and ATGs software simulators.

EXAMPLES OF FUEL DISPENSERS CONNECTION SCHEMES

Gilbarco dispenser connection scheme

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

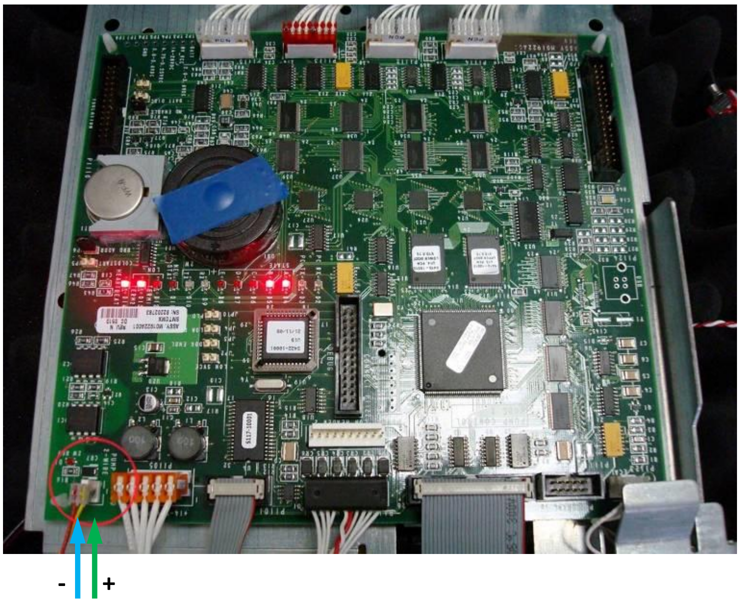
Configuration of PTS controller pump channel: protocol "*5. GILBARCO Two-Wire*", baud rate "*3. 5787*".



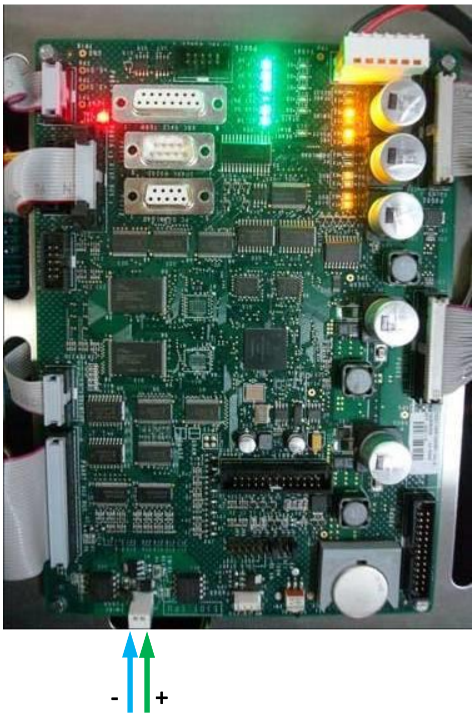
PTS controller

2-wire GB fuel dispenser interface converter
2-channel and 8-channel versions available.
More info can be found on:
http://www.technotrade.ua/2_wire_gb_interface_converter.html

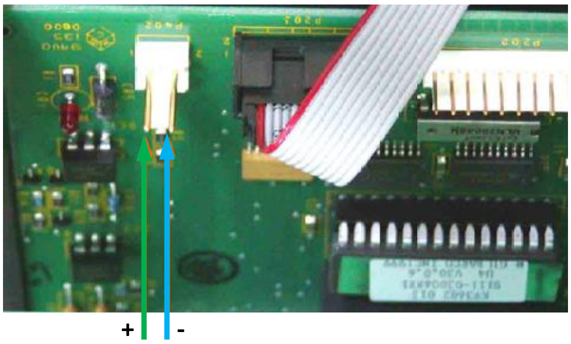
Gilbarco Encore 500 dispenser board



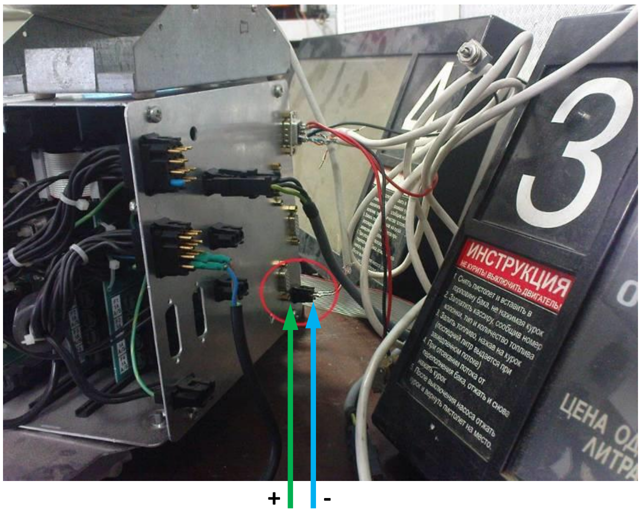
Gilbarco dispenser ASSY M06104A001 rev. B board



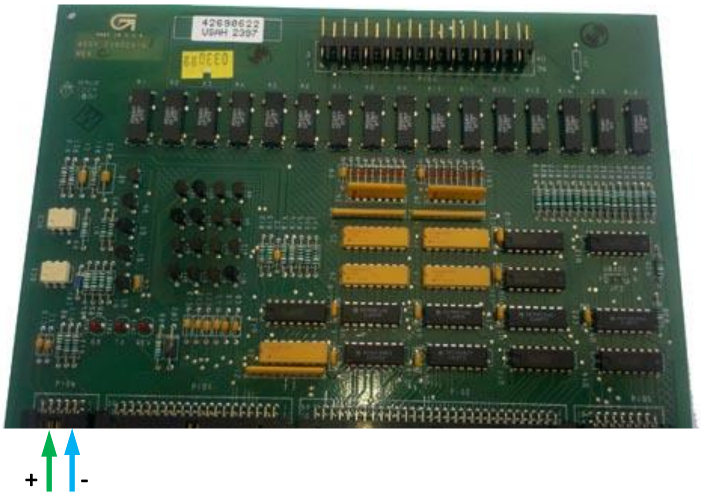
Gilbarco dispenser board



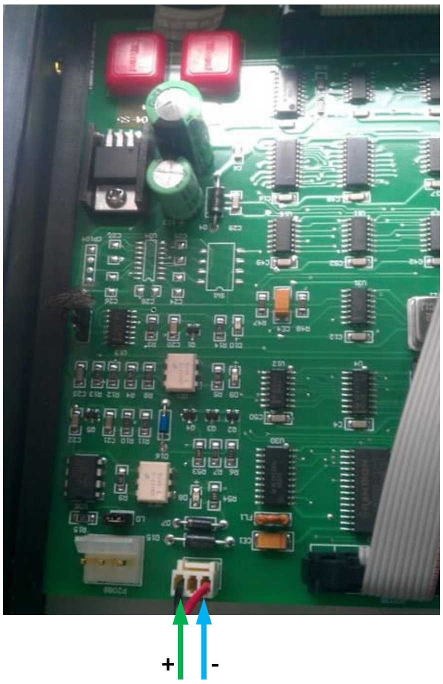
Gilbarco Euroline dispenser board



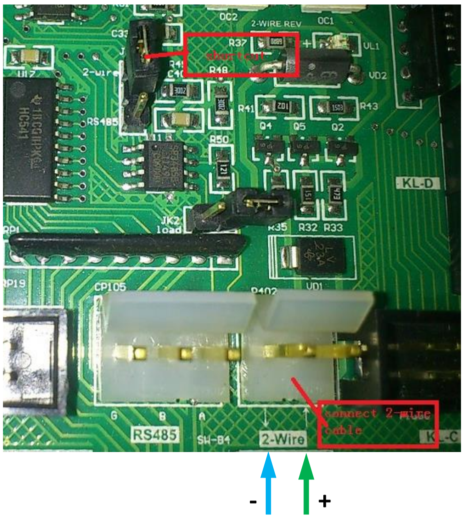
Gilbarco Highline / Dimension Assy dispenser board

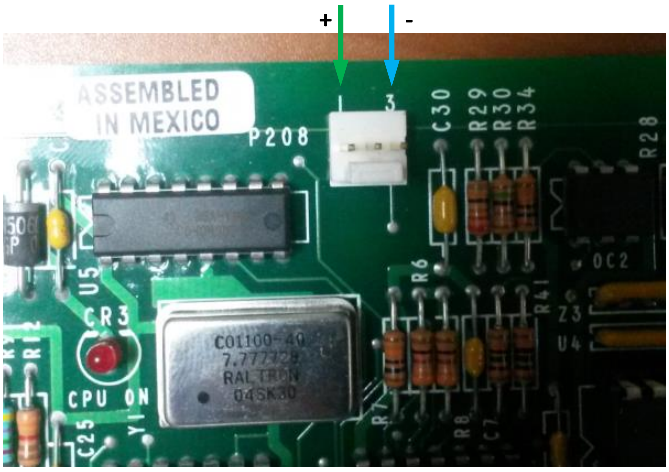


Gilbarco Endeavor dispenser board

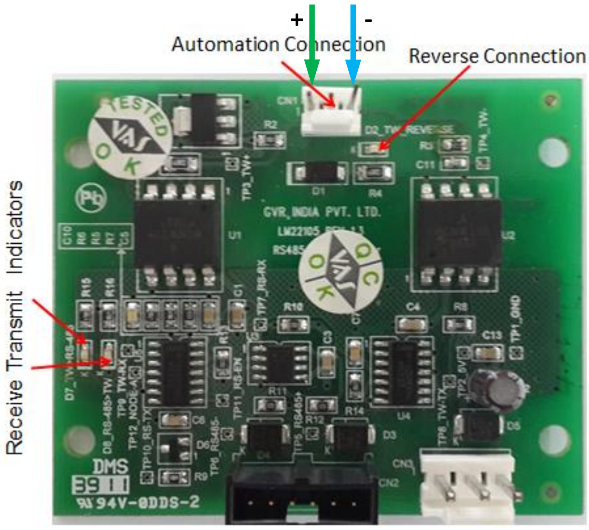


Gilbarco 3202 series dispenser board





Gilbarco Endeavor dispenser board

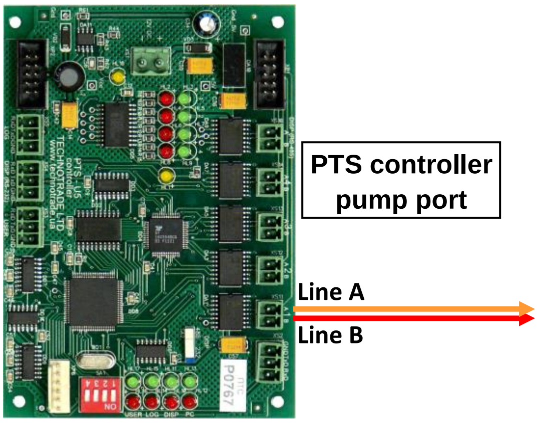


Gilbarco Frontier dispenser board

Wayne Dresser dispenser connection scheme (RS-485 interface)

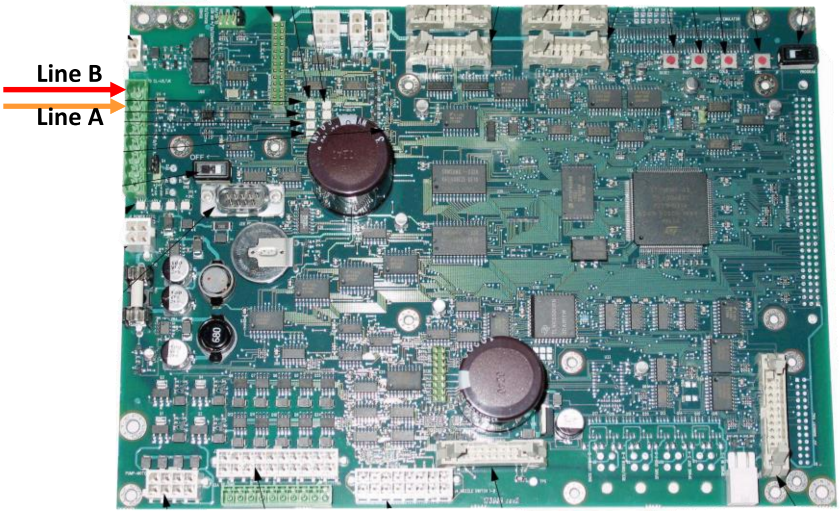
Connection to Wayne Dresser dispenser is made directly without any interface converter.

Configuration of PTS controller pump channel: protocol "*3. WAYNE Dart*", baud rate "*4. 9600*".

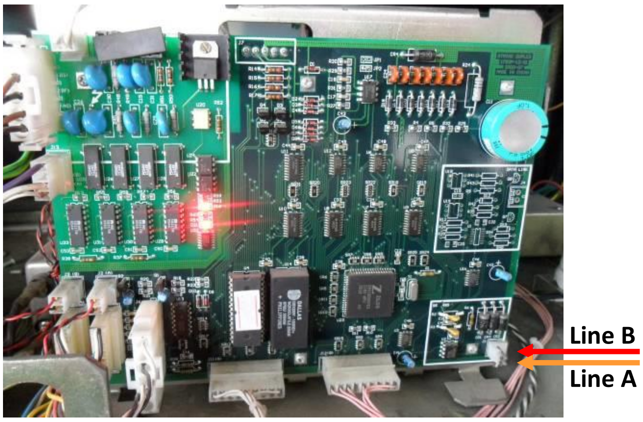


PTS controller

Wayne Dresser iGem dispenser board



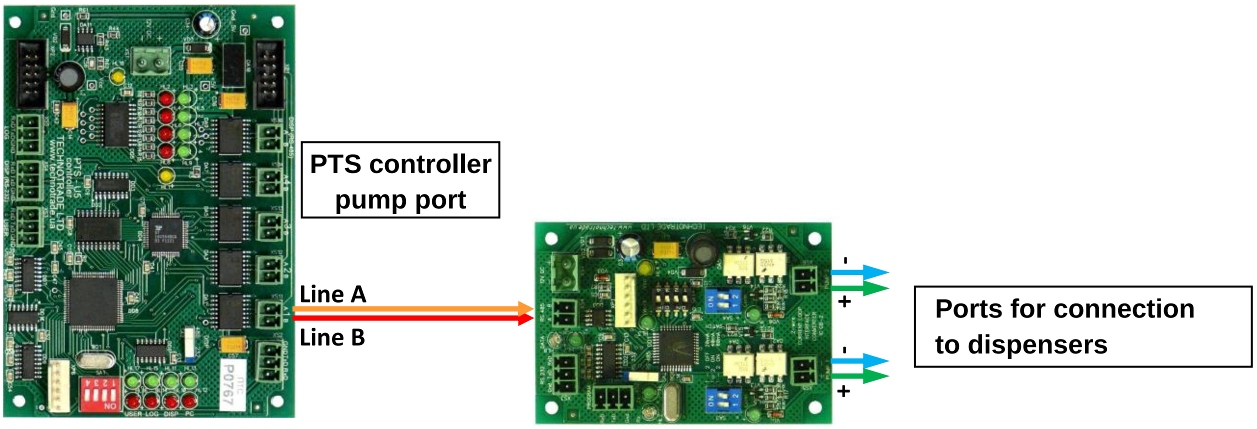
Wayne Dresser STMTAX Duplex dispenser



Wayne Dresser dispenser connection scheme (current loop interface)

Connection to Wayne Dresser dispenser is made through 2-wire GB interface converter, which provides connection of RS-232/RS-485 interfaces to 2-wire current loop interface.

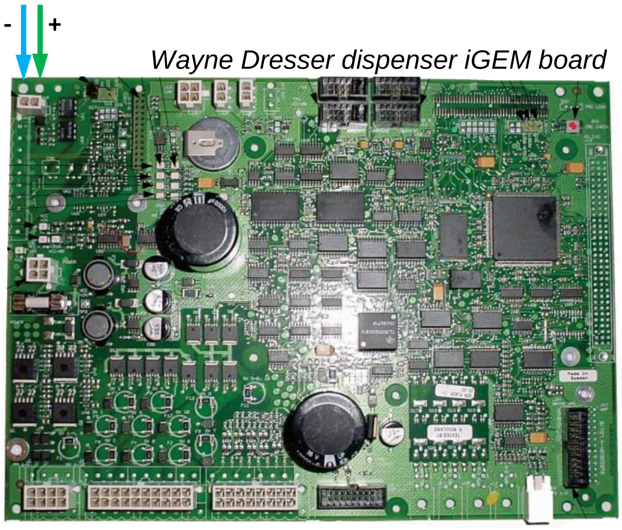
Configuration of PTS controller pump channel: protocol "*23. WAYNE USCL*", baud rate "*4. 9600*".



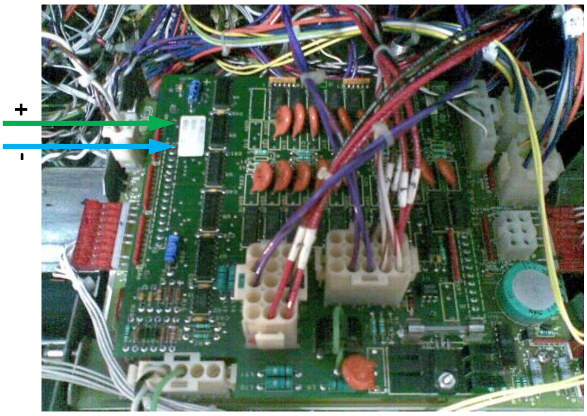
PTS controller

2-wire GB fuel dispenser interface converter
2-channel and 8-channel versions available.
More info can be found on:
http://www.technotrade.ua/2_wire_gb_interface_converter.html

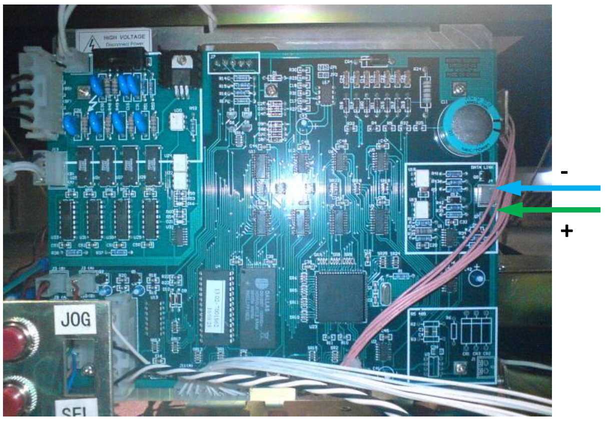
Wayne Dresser dispenser iGEM board



Wayne Dresser dispenser board



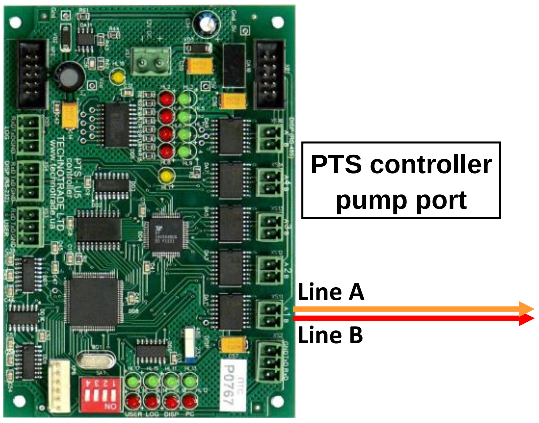
Wayne Dresser STMTAX Duplex dispenser



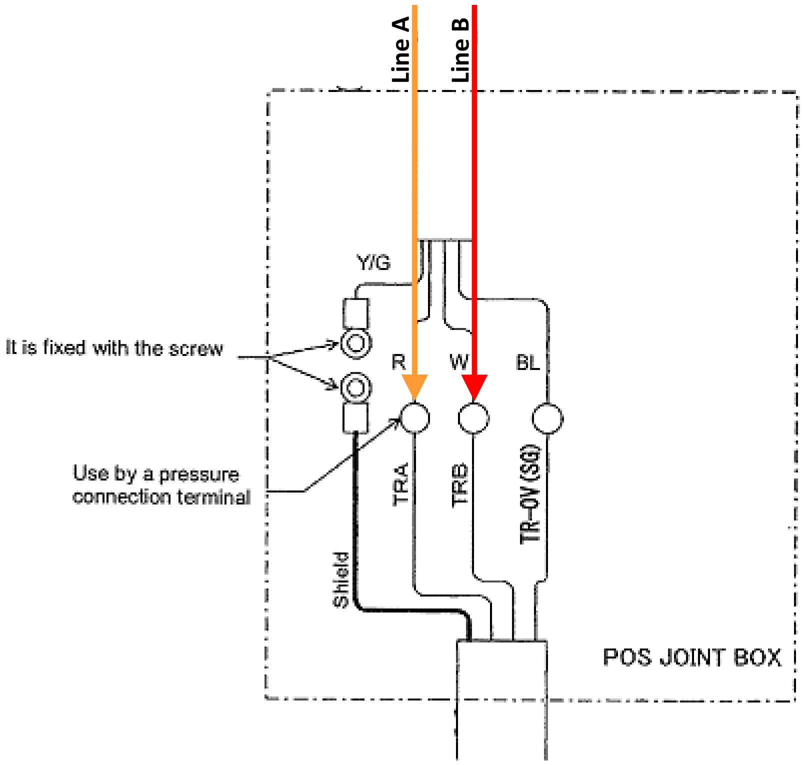
TATSUNO (Japan) dispenser connection scheme

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

Configuration of PTS controller pump channel: protocol “*15. TATSUNO SS-LAN*”, baud rate “5. *19200*”.

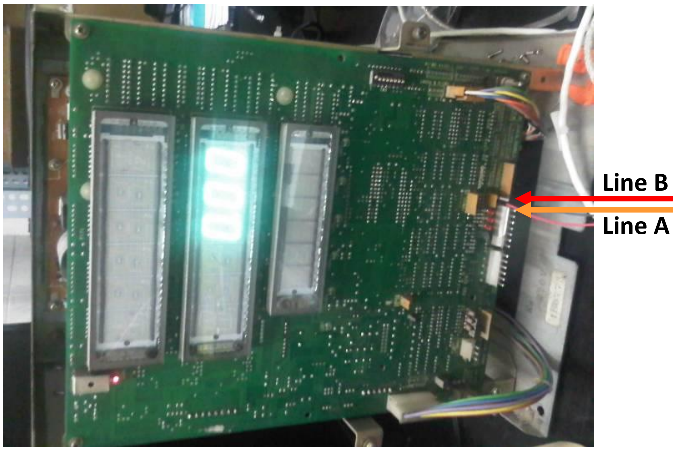


PTS controller

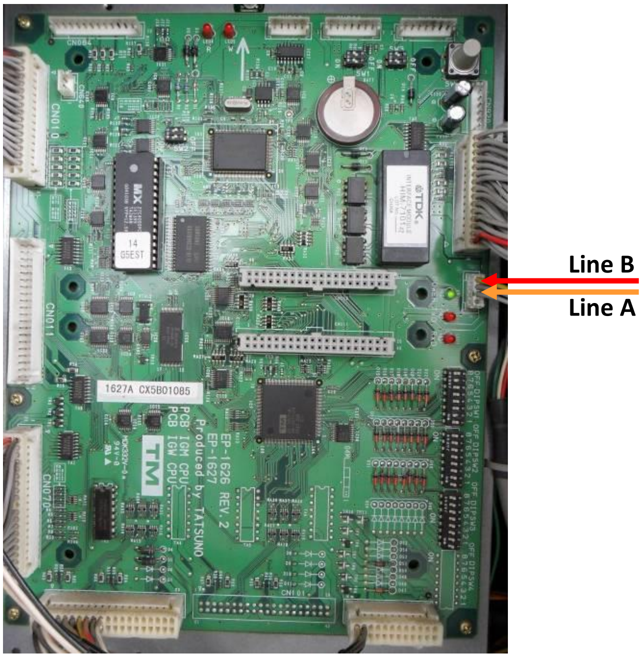

Line A
Line B
Y/G
It is fixed with the screw
R
W
BL
Use by a pressure connection terminal
TRA
TRB
TR–0V (SG)
Shield
POS JOINT BOX

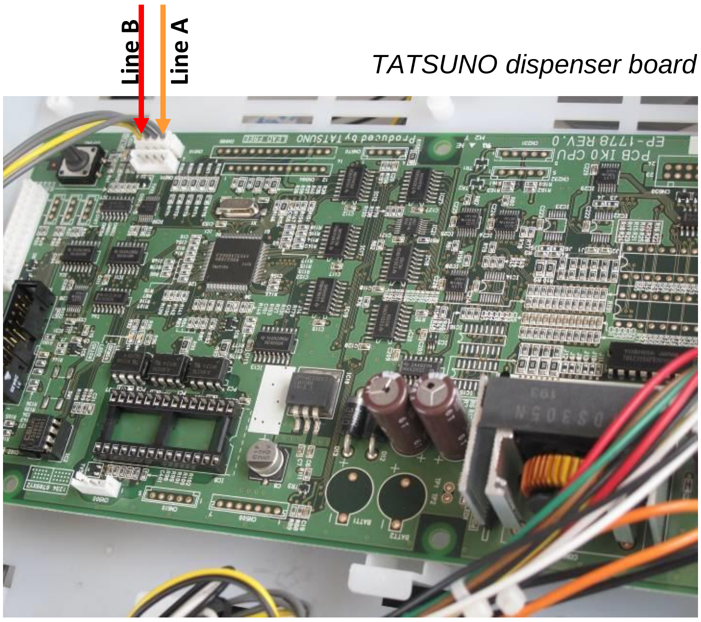
TATSUNO POS joint box

TATSUNO dispenser board



TATSUNO dispenser board

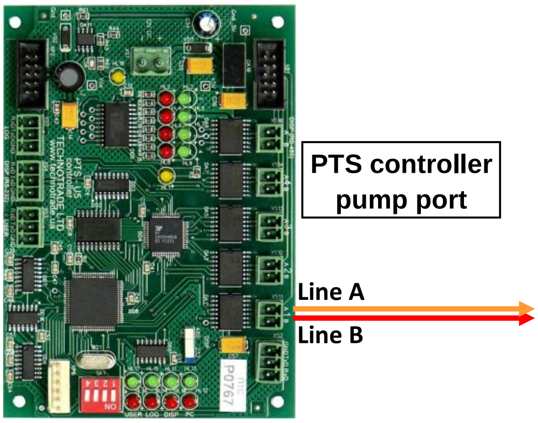




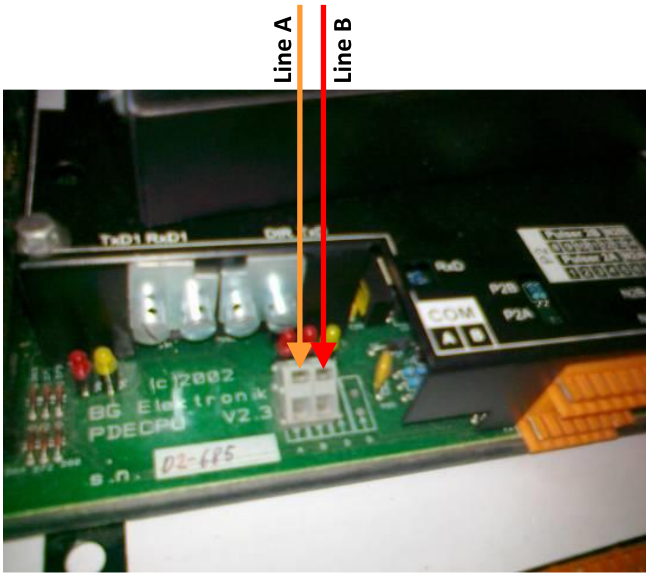
TATSUNO dispenser board

TATSUNO Benc dispenser connection scheme

Connection to TATSUNO Benc (Europe) dispenser is made directly without any interface converter. Configuration of PTS controller pump channel: protocol “*7. TATSUNO Benc PDE*”, baud rate “5. *19200*”.



PTS controller

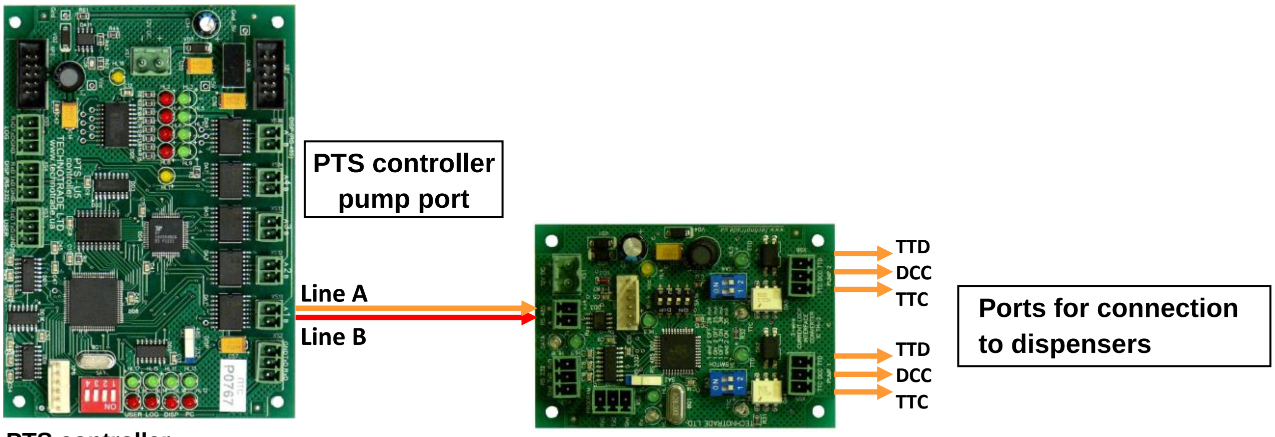


TATSUNO Benc pumphead

Tokheim dispenser connection scheme

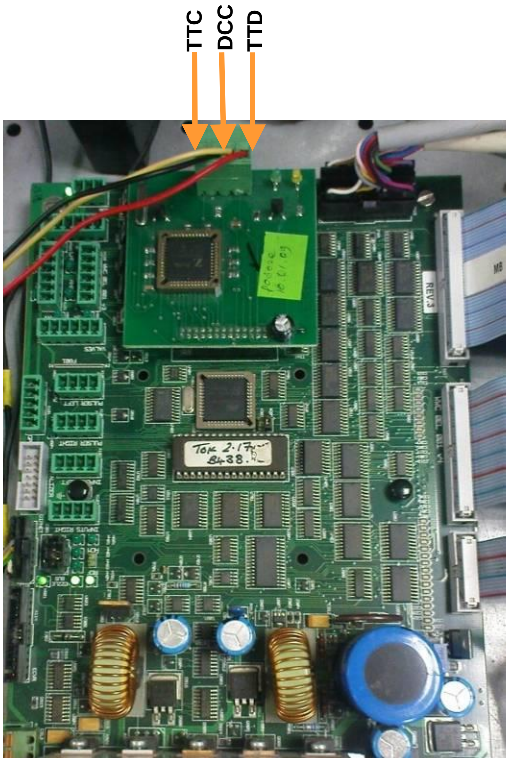
Connection to Tokheim dispenser is made through 3-wire TH interface converter, which provides connection of RS-232/RS-485 interfaces to 3-wire current loop interface.

Configuration of PTS controller pump channel: protocol “*6. TOKHEIM*”, baud rate “*4. 9600*”.

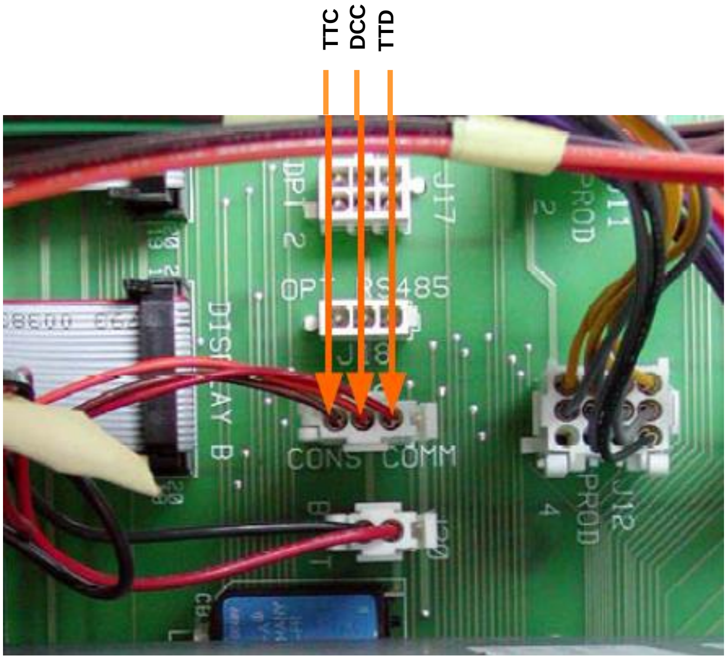


PTS controller

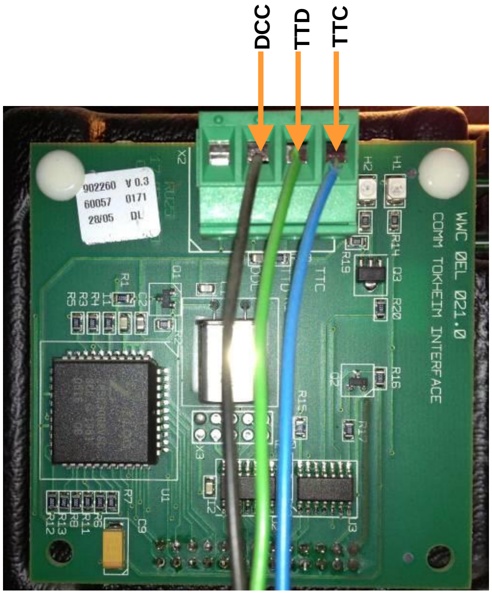
3-wire TH fuel dispenser interface converter
2-channel and 8-channel versions available.
More info can be found on:
http://www.technotrade.ua/3_wire_th_interface_converter.html



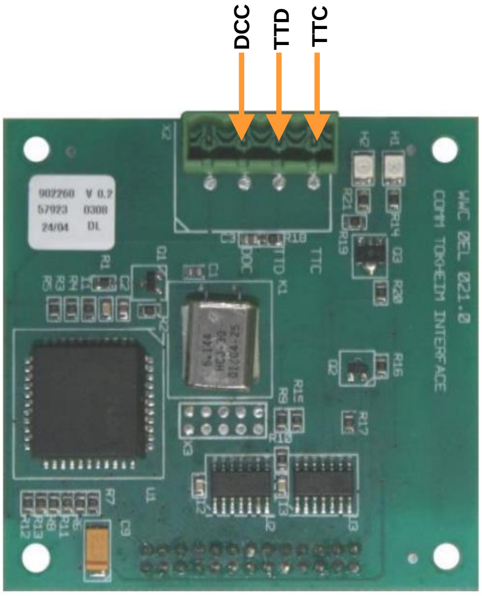
Tokheim dispenser board



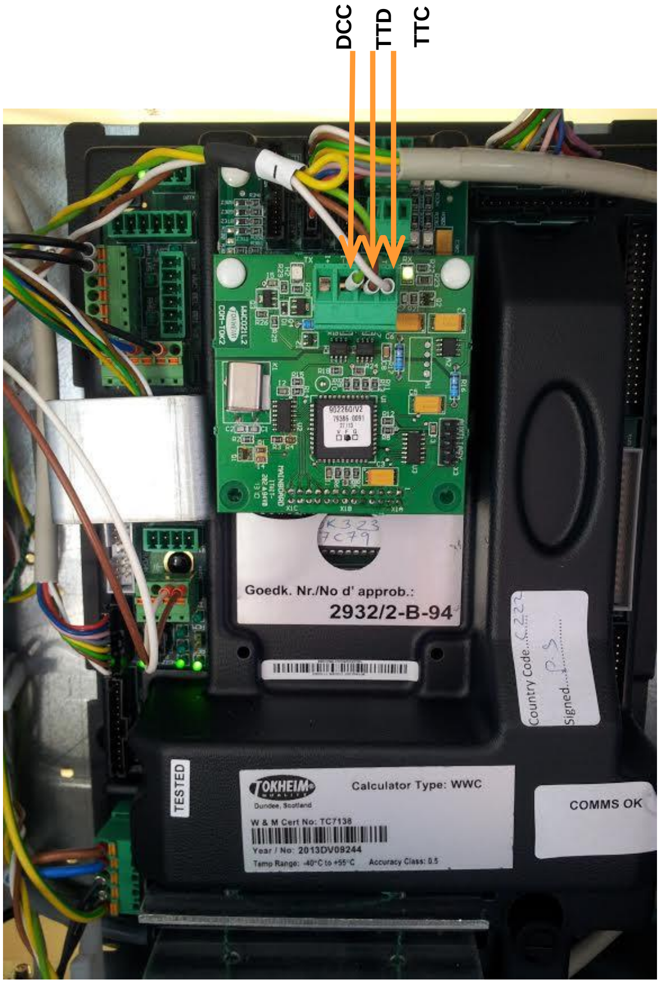
Tokheim dispenser board



Tokheim dispenser interface board



Tokheim dispenser interface board

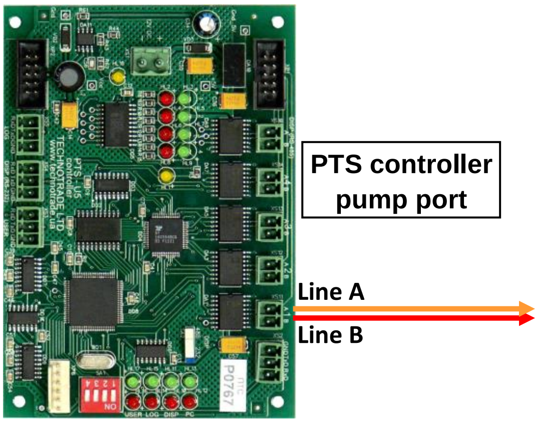


Tokheim dispenser calculator with interface board

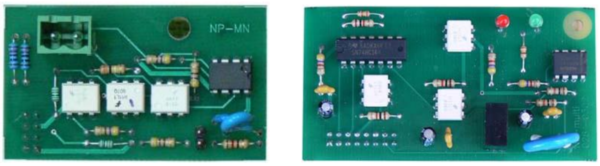
Nuovo Pignone dispenser connection scheme (RS-485 interface)

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 board for RS-485 interface and should be adjusted to have Wayne Dart protocol.

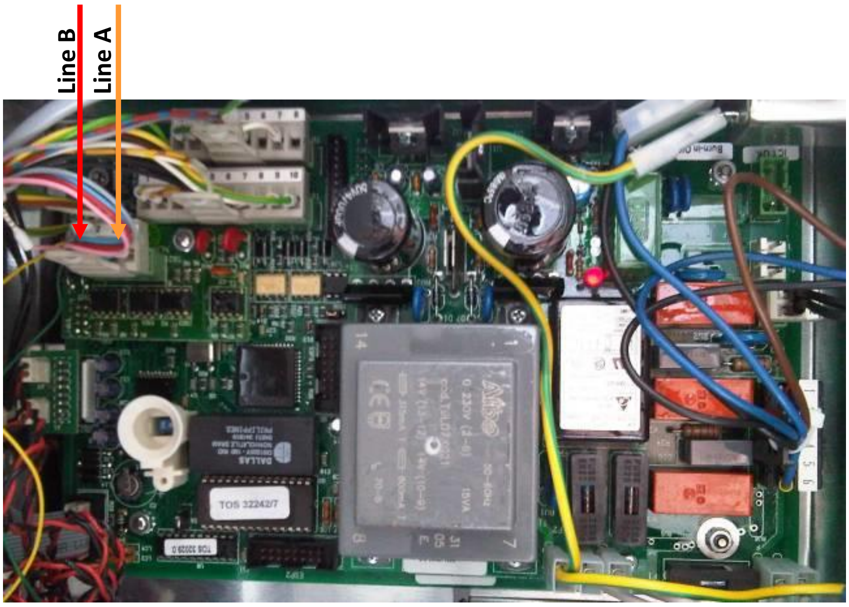
Configuration of PTS controller pump channel: protocol "*3. WAYNE Dart*", baud rate "4. *9600*".



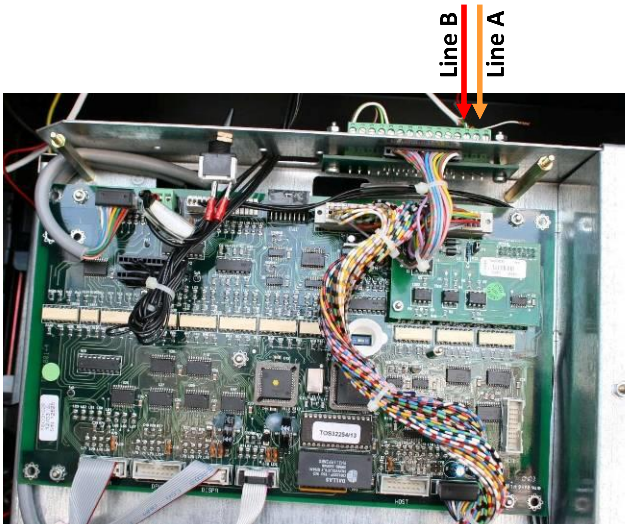
PTS controller



Nuovo Pignone interface boards for RS-485 interface for monoproduct and multiproducts dispensers
More info can be found on:
http://www.technotrade.ua/nuovo_pignone_interface_converter.html



Nuovo Pignone monoproduct dispenser board

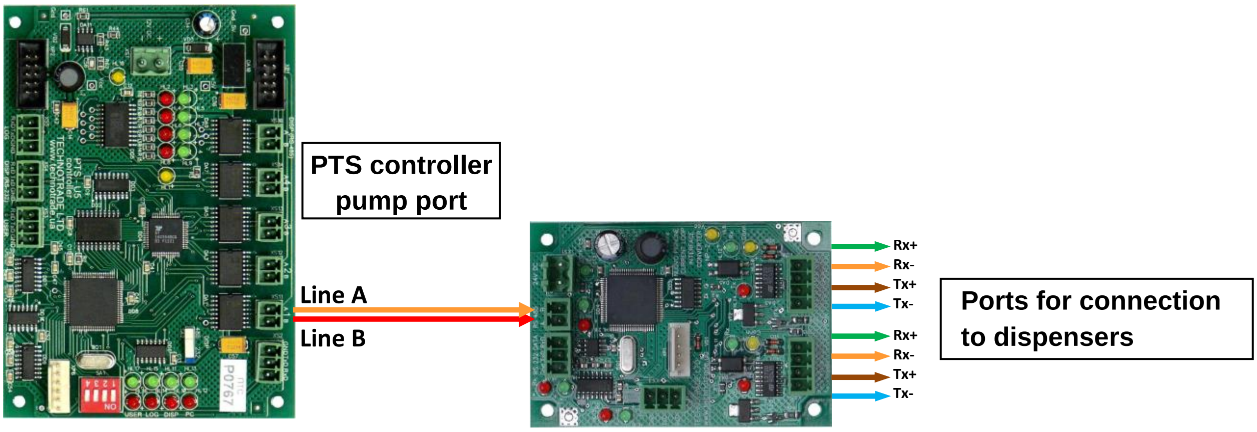


Nuovo Pignone multiproduct dispenser board

Nuovo Pignone dispenser connection scheme (4-wire current loop interface)

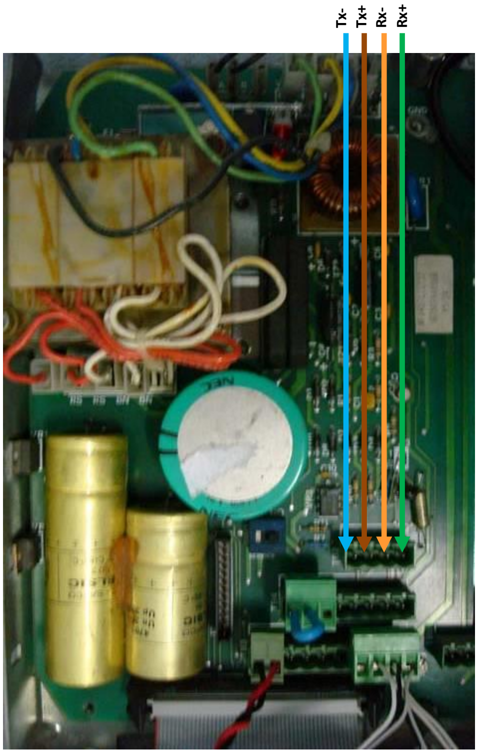
Connection to Nuovo Pignone dispensers with 4-wire current loop interface is made through 4-wire NP interface converter, which provides connection of RS-232/RS-485 interfaces to 4-wire current loop interface.

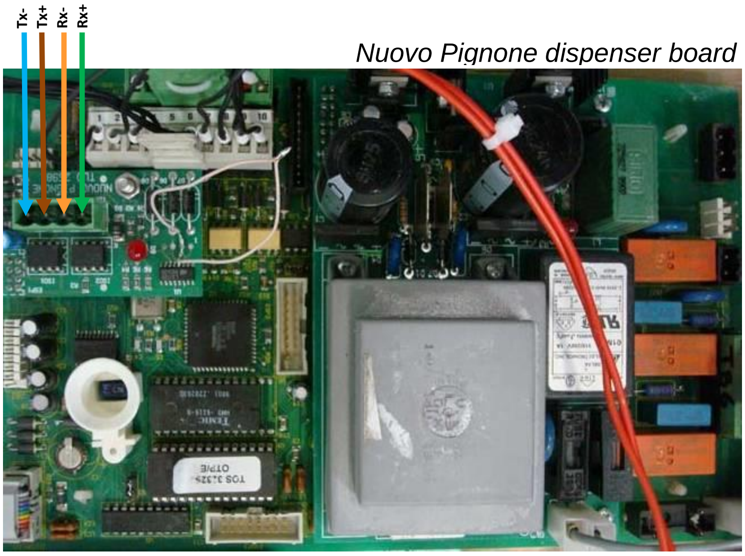
Configuration of PTS controller pump channel: protocol "*24. NUOVO PIGNONE CL*", baud rate "*1. 2400*".



PTS controller

4-wire NP fuel dispenser interface converter
2-channel and 8-channel versions available.
More info can be found on:
http://www.technotrade.ua/4_wire_np_interface_converter.html





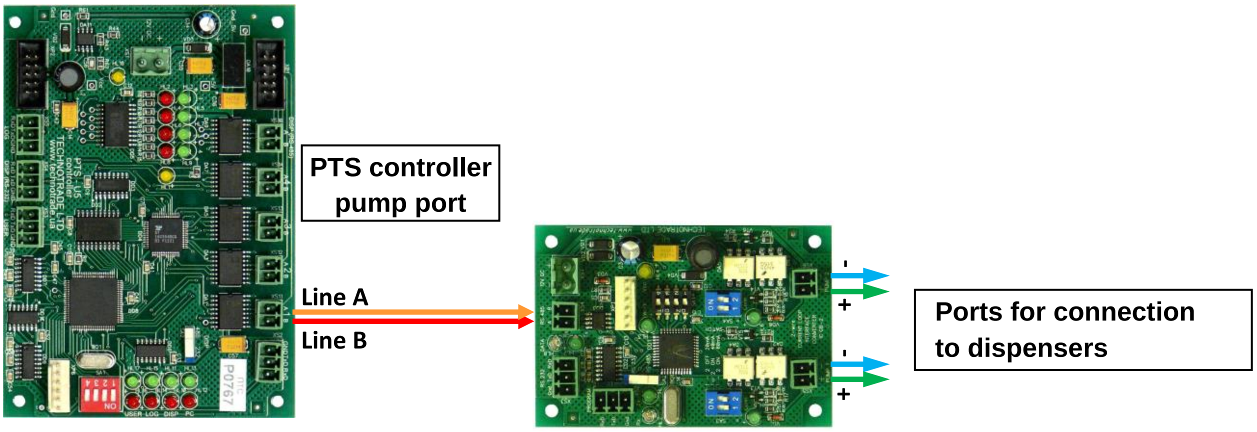
Nuovo Pignone dispenser board

Nuovo Pignone dispenser board

Batchen Email dispenser connection scheme

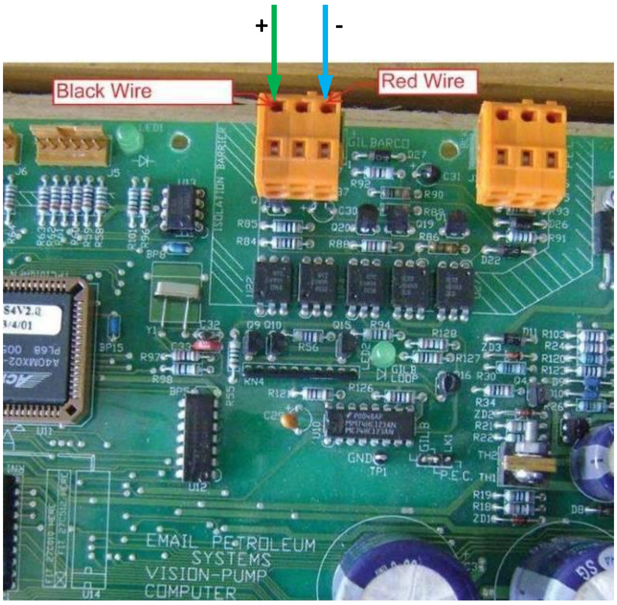
Connection to Batchen dispenser is made through 2-wire GB interface converter, which provides connection of RS-232/RS-485 interfaces to 2-wire current loop interface.

Configuration of PTS controller pump channel: protocol "*18. BATCHEN Electroline*" (for single-product dispensers) or "*39. BATCHEN MPP*" (for multi-product dispensers), baud rate "*2. 4800*".

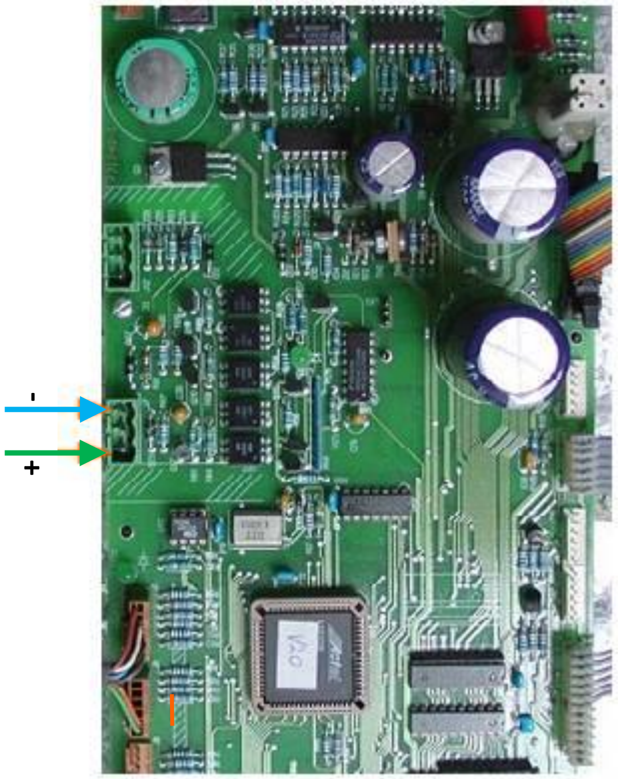


PTS controller

2-wire GB fuel dispenser interface converter
2-channel and 8-channel versions available.
More info can be found on:
http://www.technotrade.ua/2_wire_gb_interface_converter.html



Batchen dispenser board

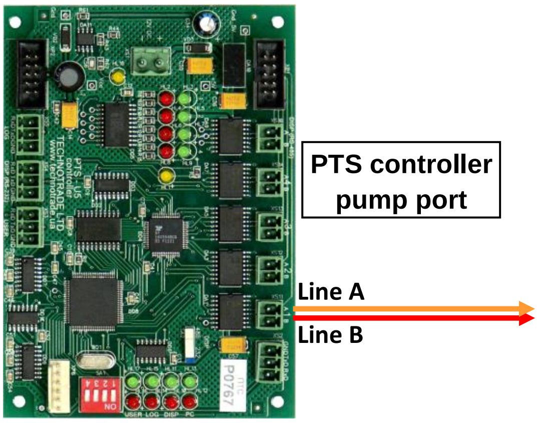


Batchen dispenser board

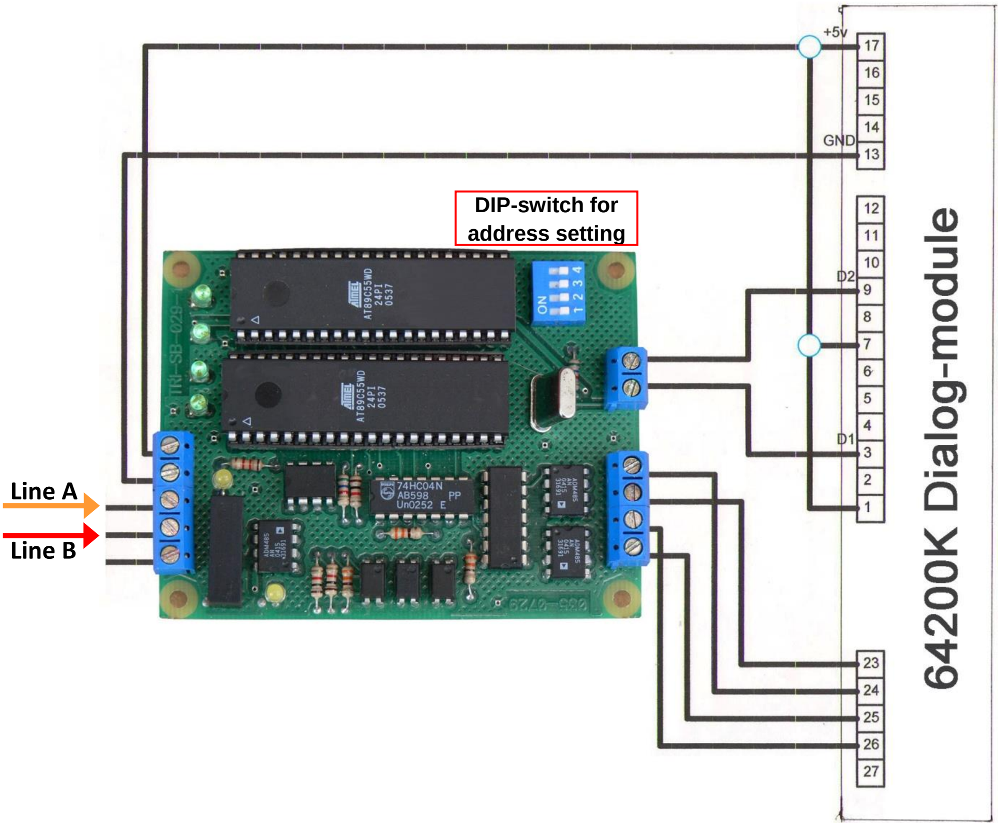
Scheidt & Bachmann T20 dispenser connection scheme

Connection to Scheidt&Bachmann T20 dispenser is made through S&B T20 interface converter board.

Configuration of PTS controller pump channel: protocol “*2. UniPump*”, baud rate “*4. 9600*”.



PTS controller



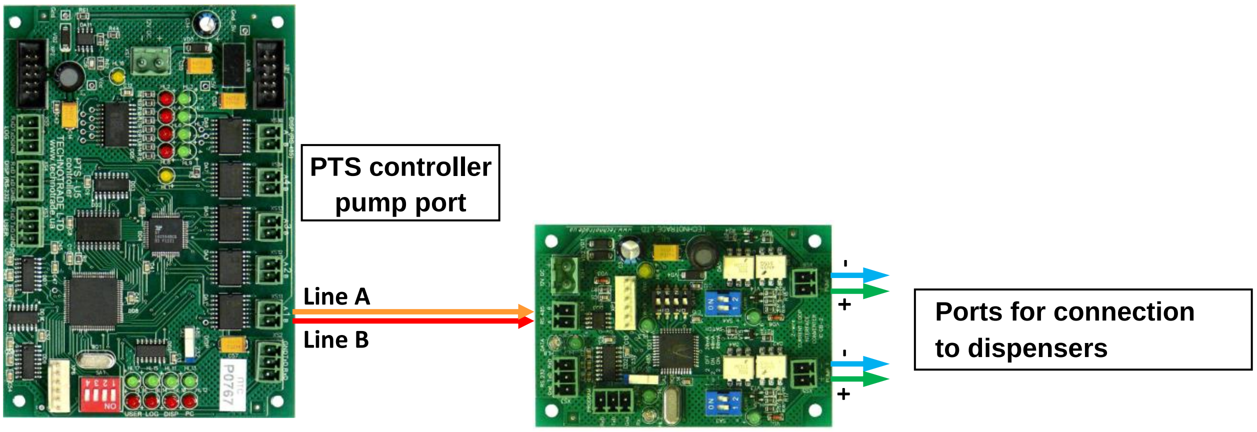
Address setting on S&B T20 interface converter board using a DIP-switch

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

Galileo dispenser connection scheme

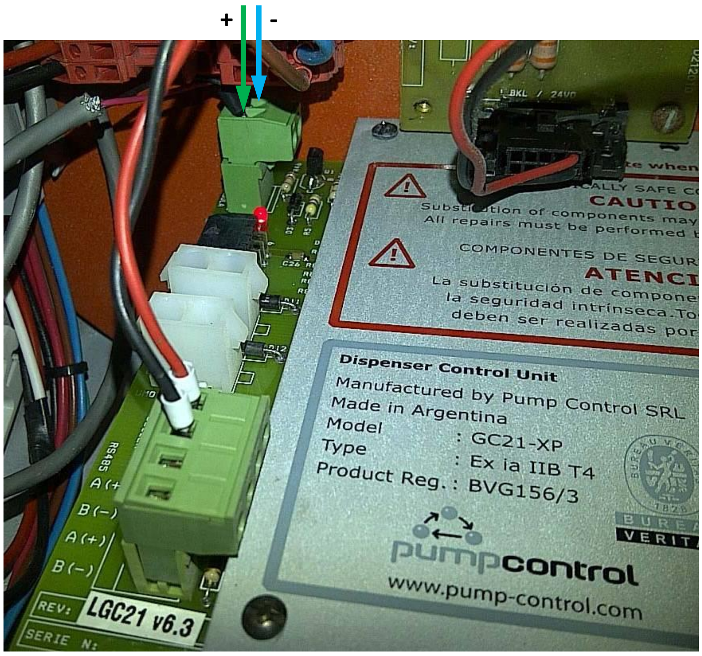
Connection to Galileo dispenser is made through 2-wire GB interface converter, which provides connection of RS-232/RS-485 interfaces to 2-wire current loop interface.

Configuration of PTS controller pump channel: protocol "*10. GALILEO PumpControl GC21*", baud rate "*2. 4800*".



PTS controller

2-wire GB fuel dispenser interface converter
2-channel and 8-channel versions available.
More info can be found on:
http://www.technotrade.ua/2_wire_gb_interface_converter.html

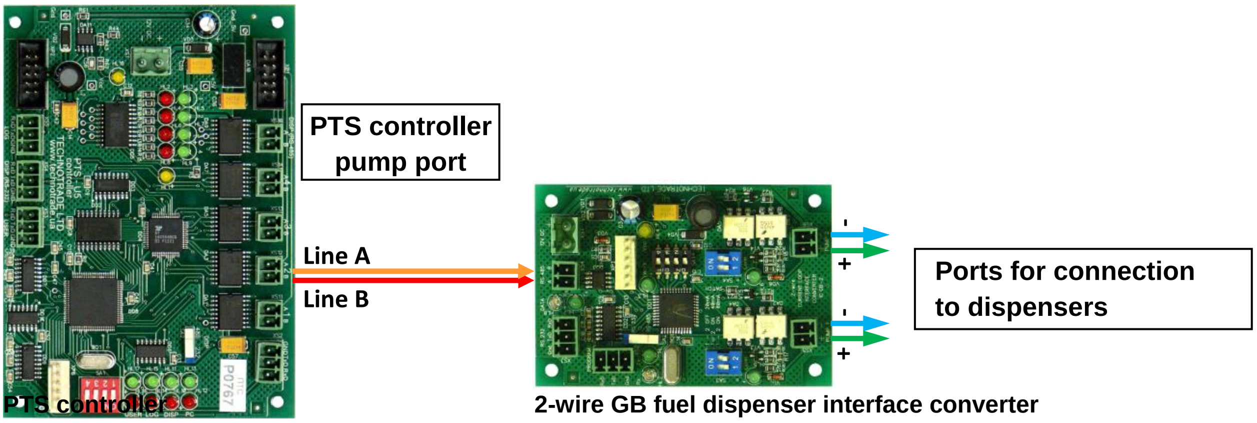


Galileo dispenser board

Prowalco dispenser connection scheme

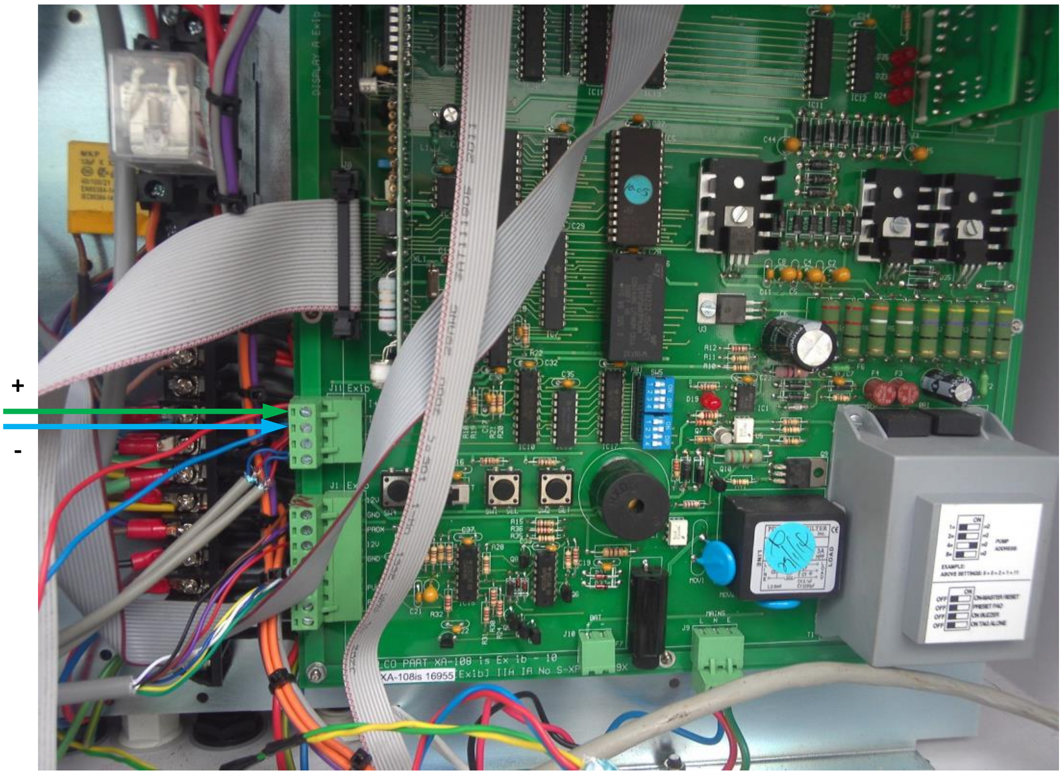
Connection to Prowalco dispenser is made through 2-wire GB interface converter, which provides connection of RS-232/RS-485 interfaces to 2-wire current loop interface.

Configuration of PTS controller pump channel: protocol "*27. PROWALCO*", baud rate "*2. 4800*".



PTS controller

2-wire GB fuel dispenser interface converter
2-channel and 8-channel versions available.
More info can be found on:
http://www.technotrade.ua/2_wire_gb_interface_converter.html

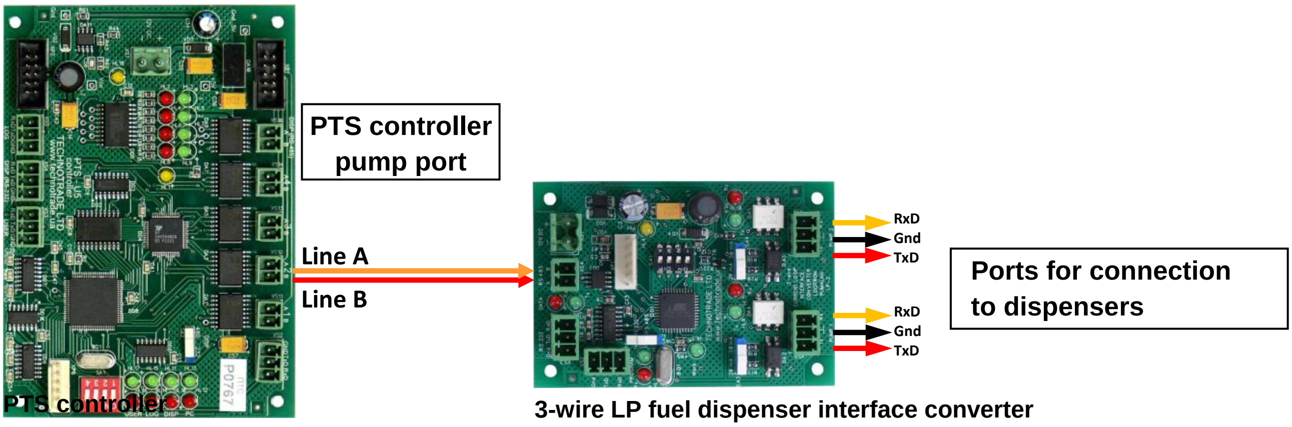


Prowalco dispenser board

Emgaz Dragon dispenser with EsiWelma pumphead connection scheme

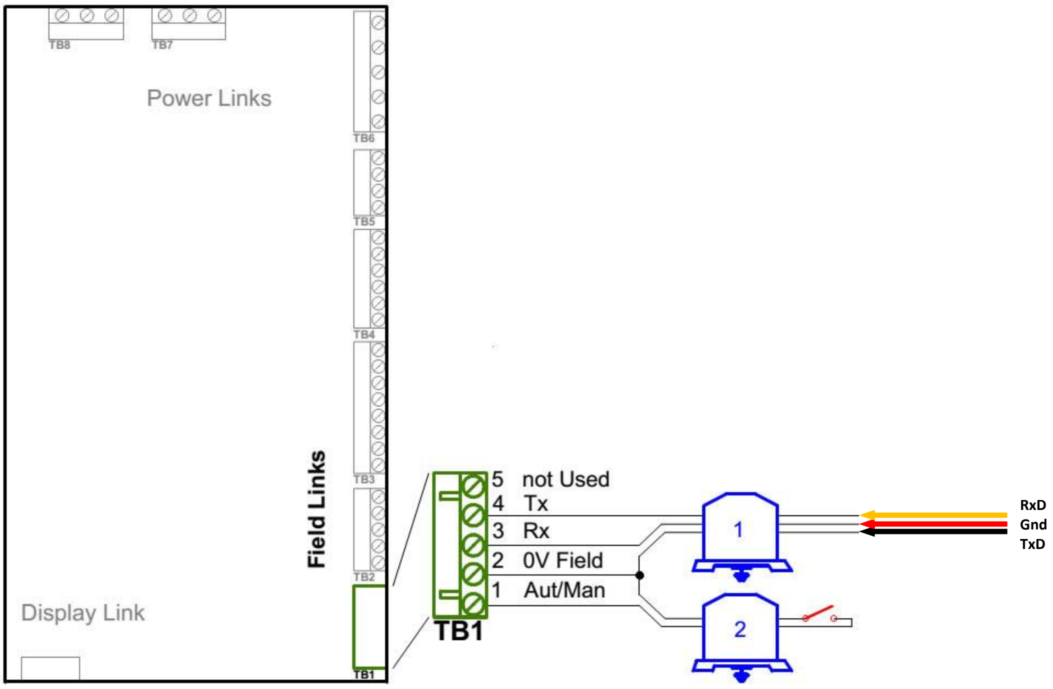
Connection to EsiWelma pumphead is made through 3-wire LP interface converter, which provides connection of RS-232/RS-485 interfaces to 3-wire current loop interface.

Configuration of PTS controller pump channel: protocol "*19. PUMALAN Marconi*", baud rate "*2. 4800*".



PTS controller

3-wire LP fuel dispenser interface converter
2-channel and 8-channel versions available.
More info can be found on:
http://www.technotrade.ua/3_wire_lp_interface_converter.html

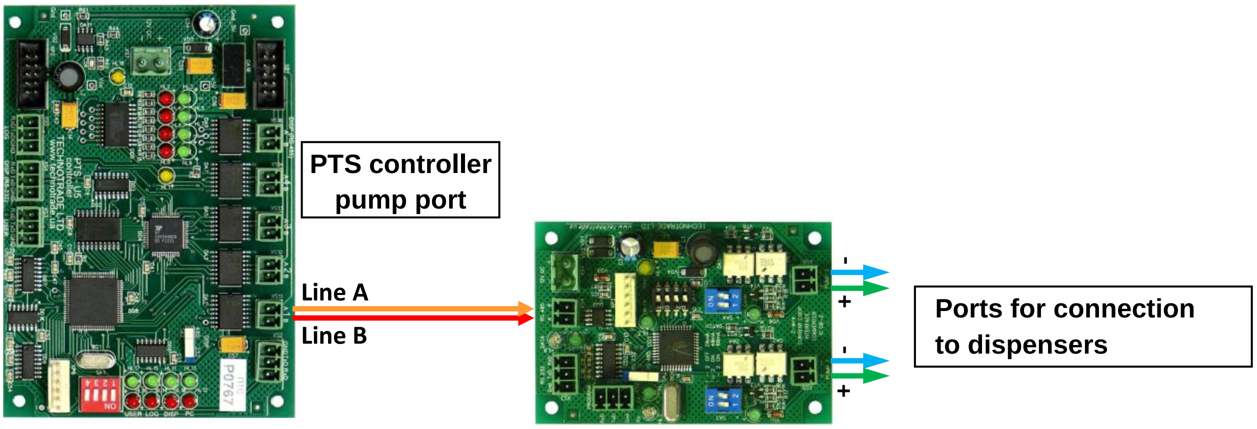


EsiWelma pumphead

PETPOSAN-S4 / MEKSAN-S4 / EUROPUMP-S4 / YENEN dispensers connection scheme

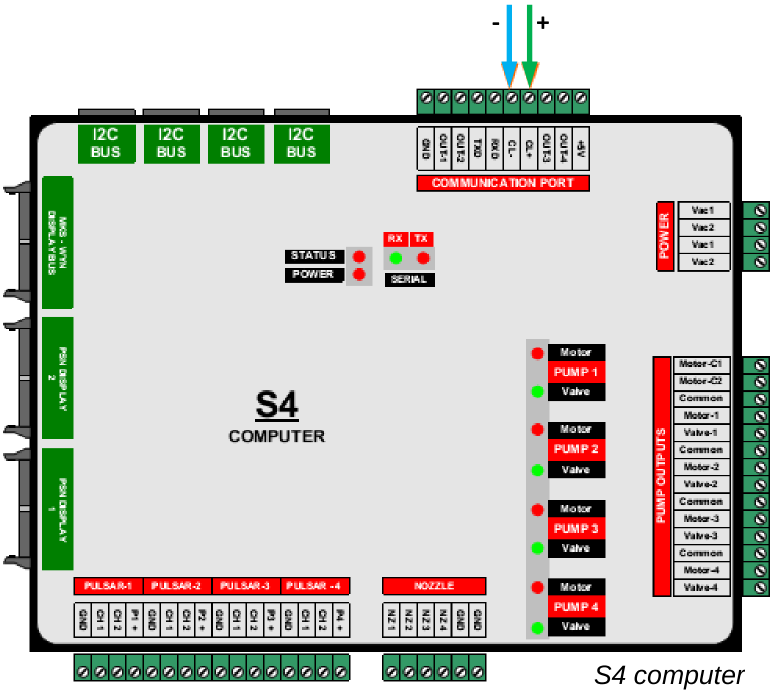
Connection to S4 computer is made through 2-wire GB interface converter, which provides connection of RS-232/RS-485 interfaces to 2-wire current loop interface.

Configuration of PTS controller pump channel: protocol “*33. Dart Simplex*”, baud rate “*4. 9600*”.



PTS controller

2-wire GB fuel dispenser interface converter
2-channel and 8-channel versions available.
More info can be found on:
http://www.technotrade.ua/2_wire_gb_interface_converter.html

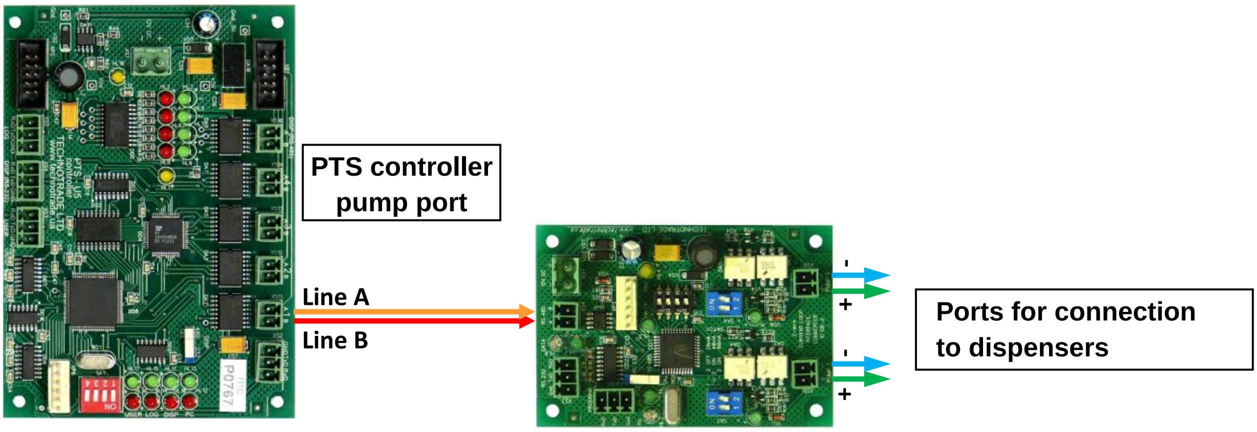


S4 computer

PETPOSAN-Beta / EUROPUMP-Beta dispensers connection scheme

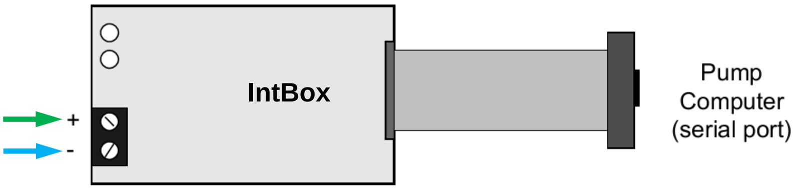
Connection to Beta computer is made through 2-wire GB interface converter, which provides connection of RS-232/RS-485 interfaces to 2-wire current loop interface.

Configuration of PTS controller pump channel: protocol "*33. Dart Simplex*", baud rate "*4. 9600*".



PTS controller

2-wire GB fuel dispenser interface converter
2-channel and 8-channel versions available.
More info can be found on:
http://www.technotrade.ua/2_wire_gb_interface_converter.html



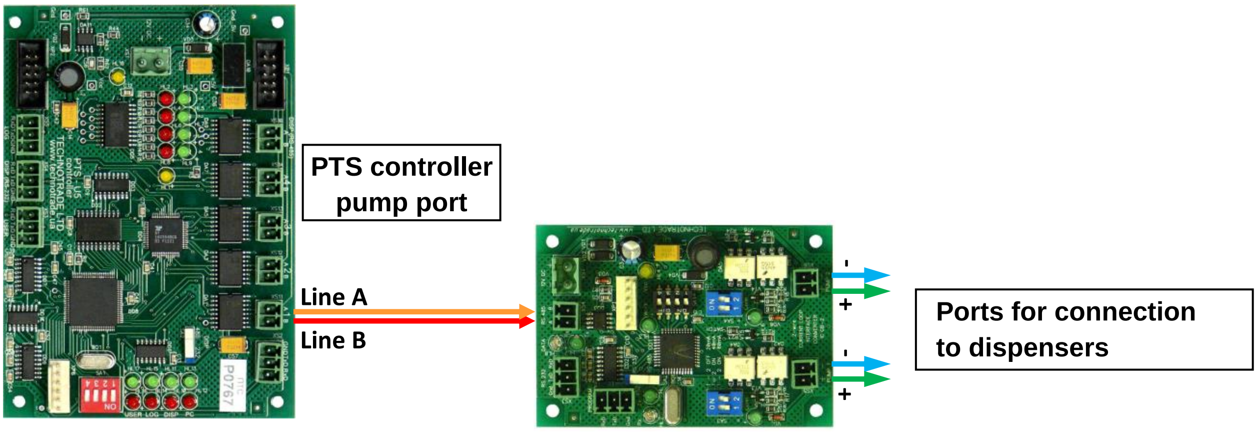
PETPOSAN
HPC - 04 ELECTRONIC PUMP COMPUTER
KEYBOARD
PRINTER
SERIAL PORT
DIKKAT!
CIHAZDA GERILIM VARKEN KONNEKTÖRLERI SÖKUP TAKMAYINIZ
WARNING!
EITHER PLUG OR UNPLUG WHILE POWERED
PUMP-B
PUMP-A
PULSER-B
PULSER-A

Petposan-Beta CPU

EuroPump dispenser connection scheme

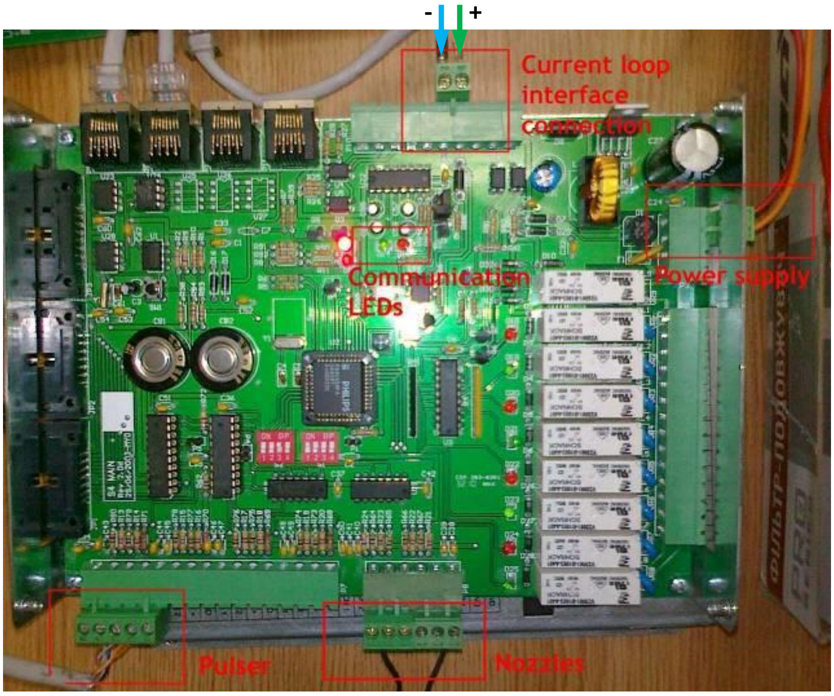
Connection to EuroPump dispenser is made through 2-wire GB interface converter, which provides connection of RS-232/RS-485 interfaces to 2-wire current loop interface.

Configuration of PTS controller pump channel: protocol "*33. Dart Simplex*", baud rate "*4. 9600*".



PTS controller

2-wire GB fuel dispenser interface converter
2-channel and 8-channel versions available.
More info can be found on:
http://www.technotrade.ua/2_wire_gb_interface_converter.html

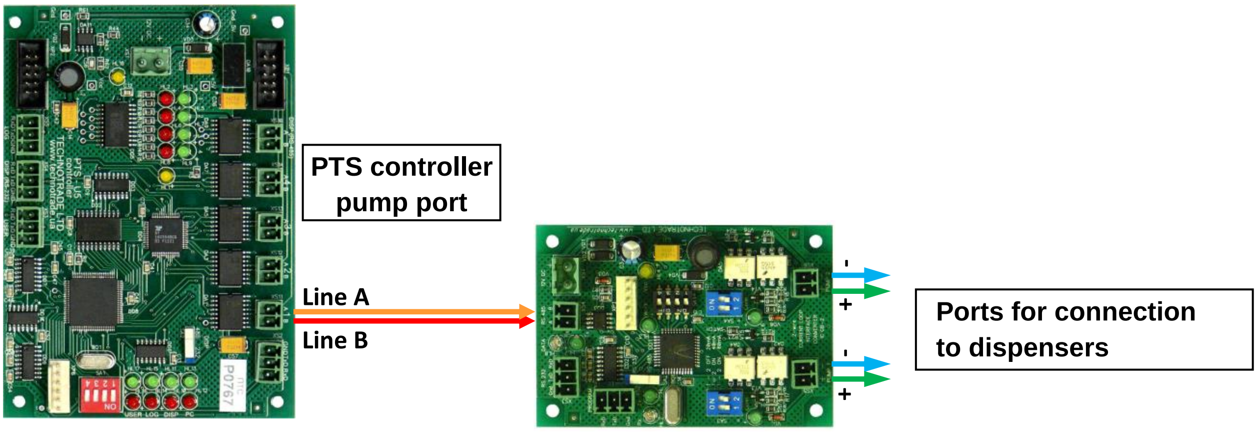


EuroPump dispenser board

Mekser dispenser connection scheme

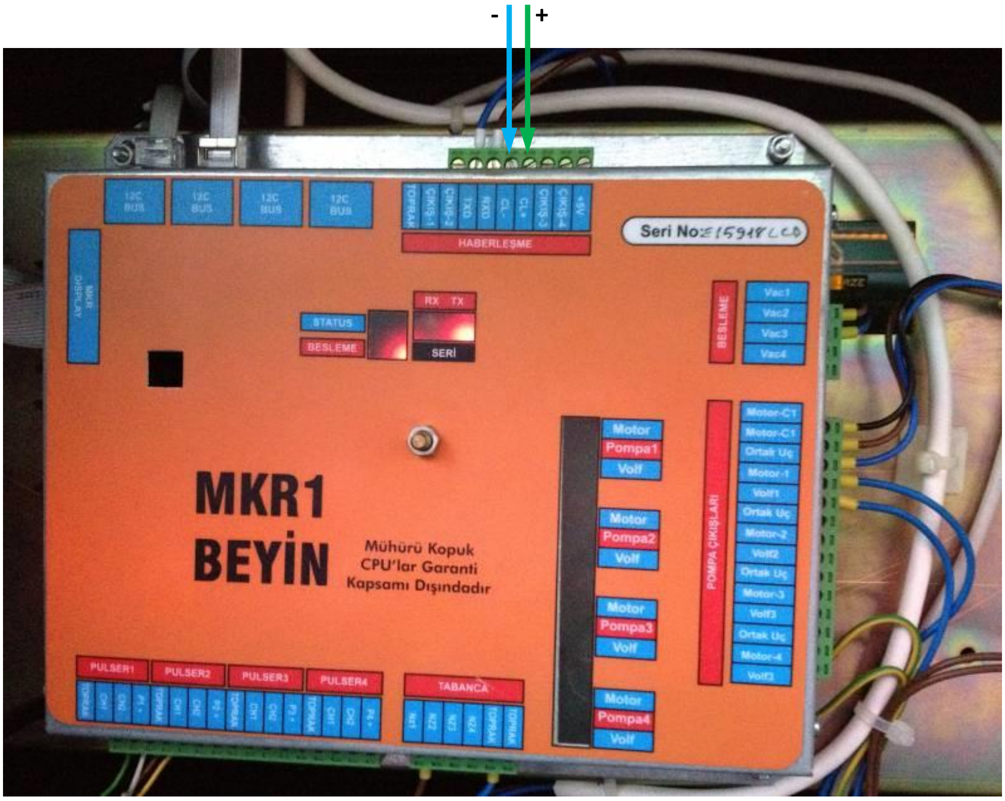
Connection to Mekser dispenser is made through 2-wire GB interface converter, which provides connection of RS-232/RS-485 interfaces to 2-wire current loop interface.

Configuration of PTS controller pump channel: protocol "*33. Dart Simplex*", baud rate "*4. 9600*".



PTS controller

2-wire GB fuel dispenser interface converter
2-channel and 8-channel versions available.
More info can be found on:
http://www.technotrade.ua/2_wire_gb_interface_converter.html

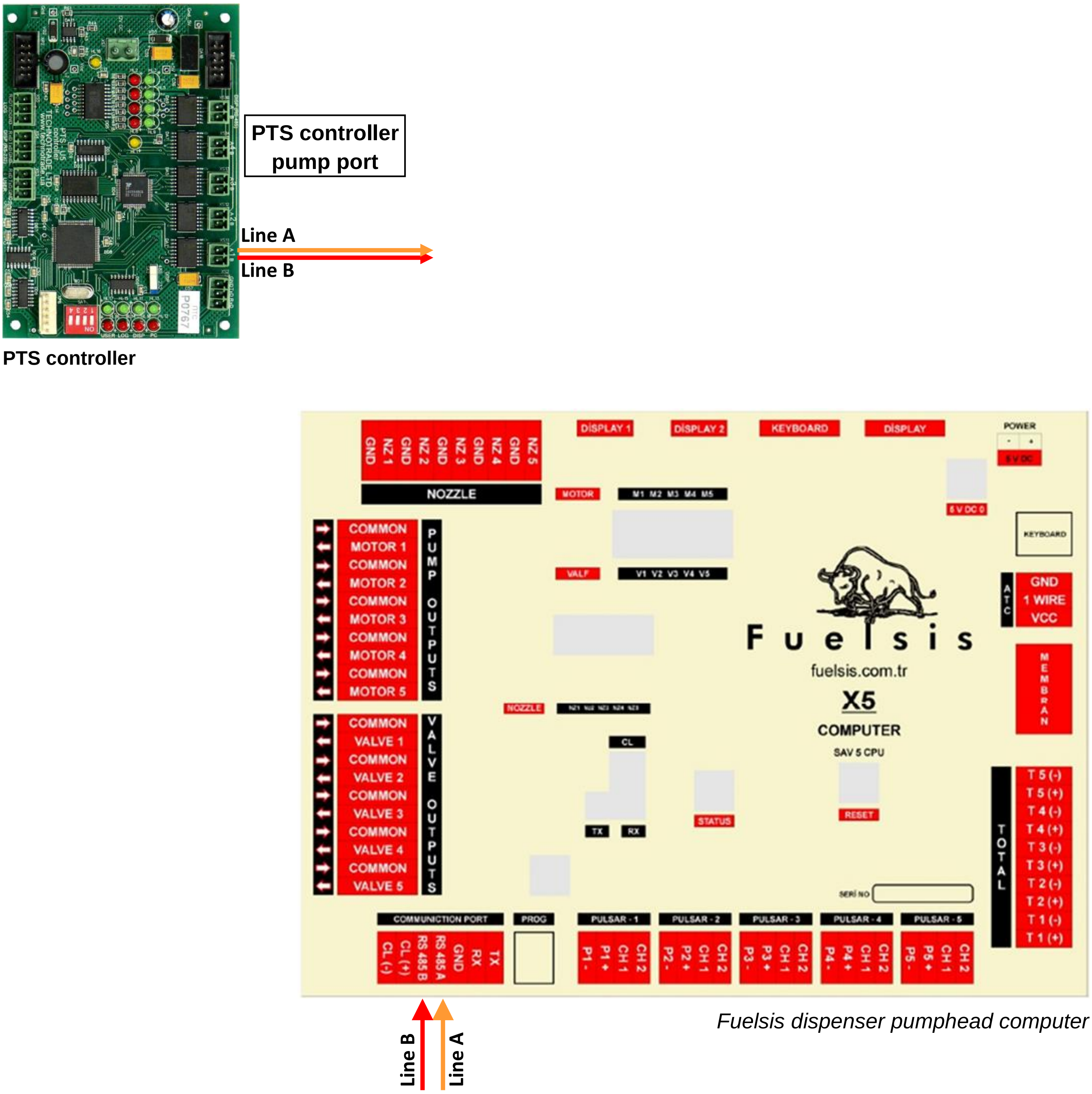


Mekser dispenser board

Fuelsis dispenser connection scheme

Connection to Fuelsis dispenser is made directly without any interface converter.

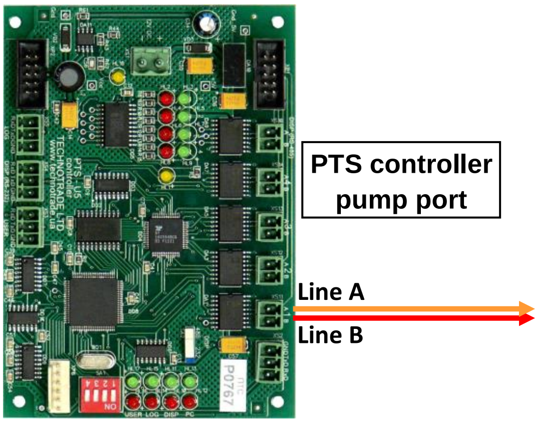
Configuration of PTS controller pump channel: protocol “*33. Dart Simplex*”, baud rate “*4. 9600*”.



Mepsan UNIMEP dispenser connection scheme

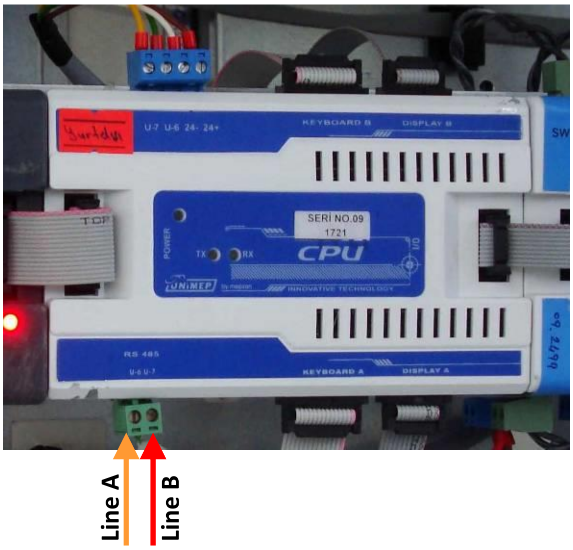
Connection to Mepsan dispenser is made directly without any interface converter.

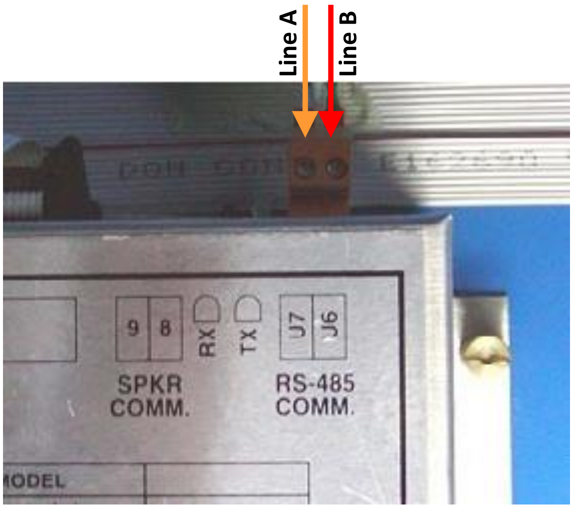
Configuration of PTS controller pump channel: protocol “*33. Dart Simplex*”, baud rate “*4. 9600*”.



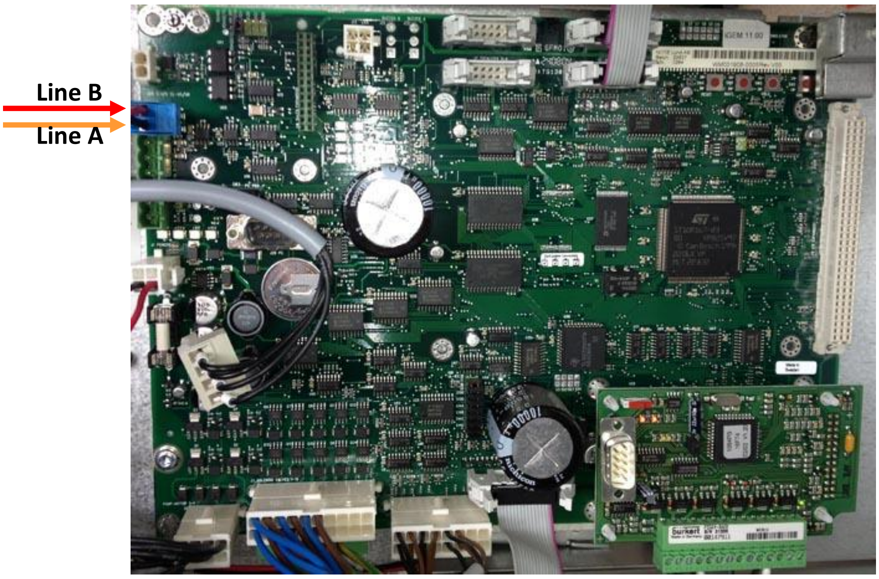
PTS controller

Mepsan Unimep dispenser calculator





Mepsan Unimep dispenser calculator

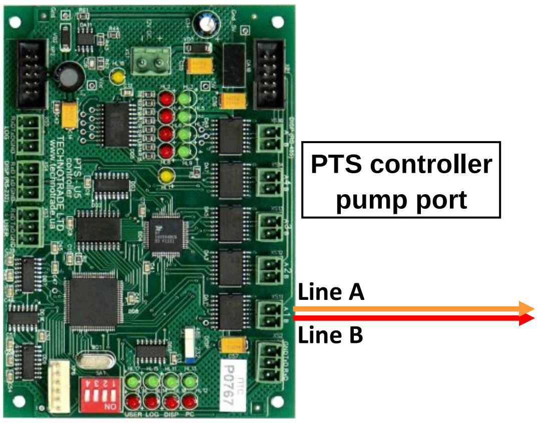


Mepsan dispenser board

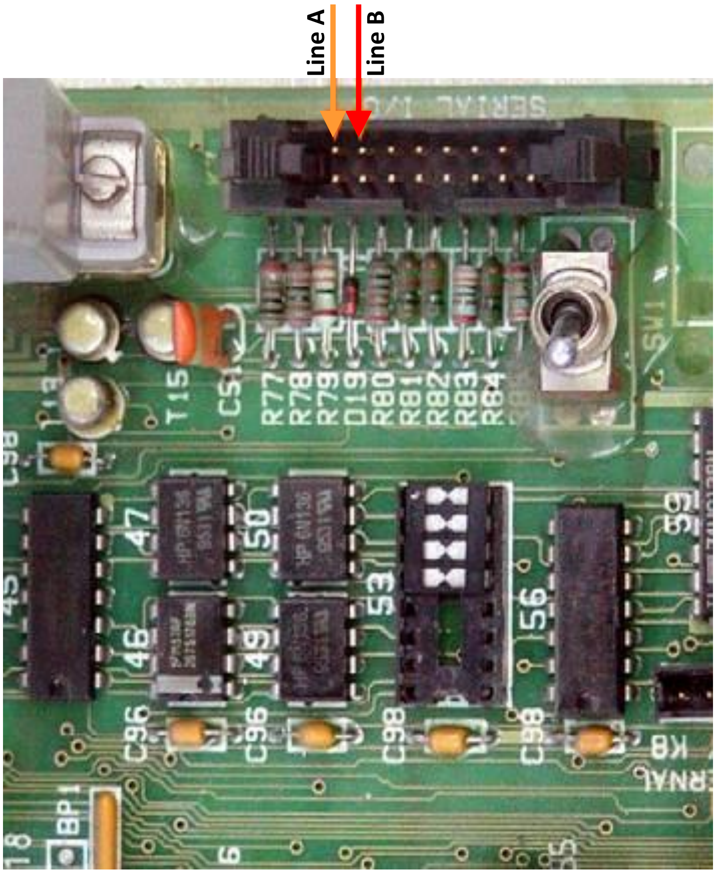
Meksan / Wayne SU86 dispenser connection scheme

Connection to Meksan / Wayne SU86 dispenser is made directly without any interface converter.

Configuration of PTS controller pump channel: protocol “*33. Dart Simplex*”, baud rate “*4. 9600*”.



PTS controller

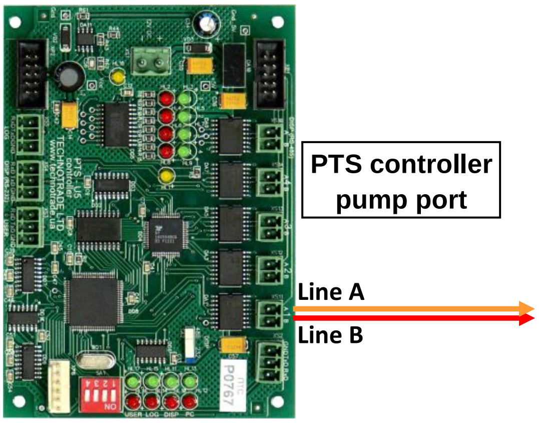


MEKSAN / WAYNE SU86 dispenser board

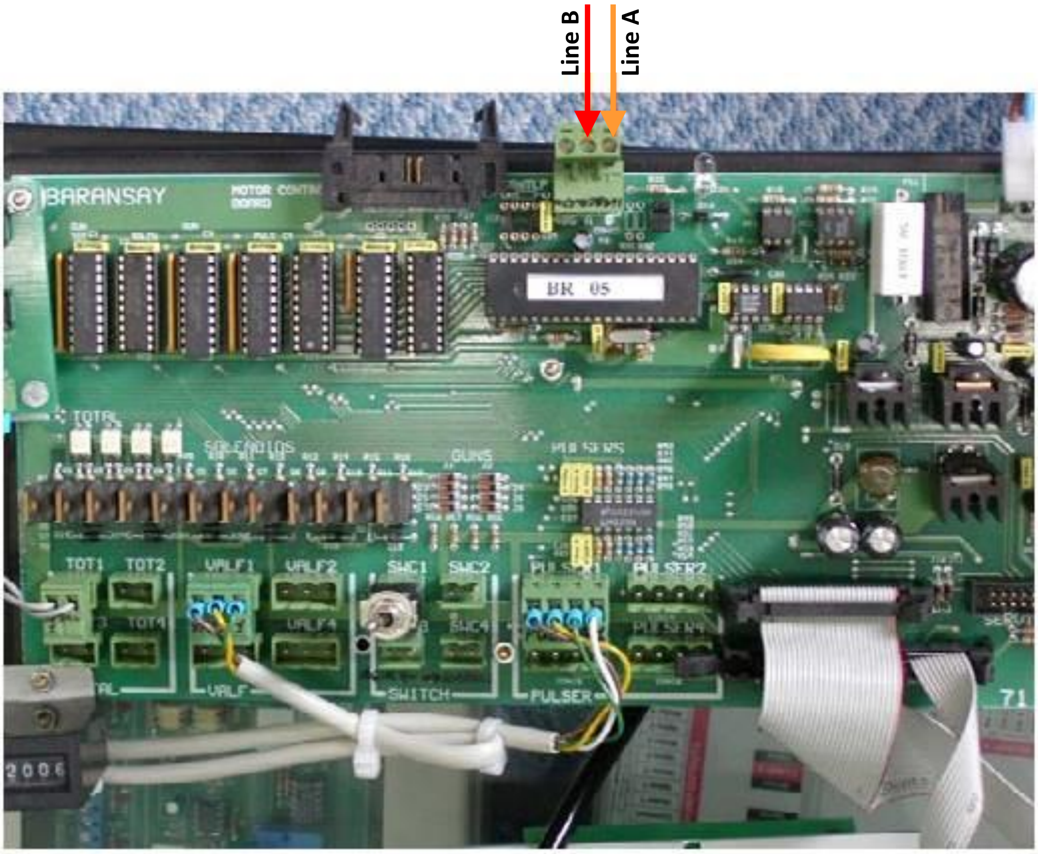
Baransay dispenser connection scheme

Connection to Baransay dispenser is made directly without any interface converter.

Configuration of PTS controller pump channel: protocol “*33. Dart Simplex*”, baud rate “*4. 9600*”.



PTS controller

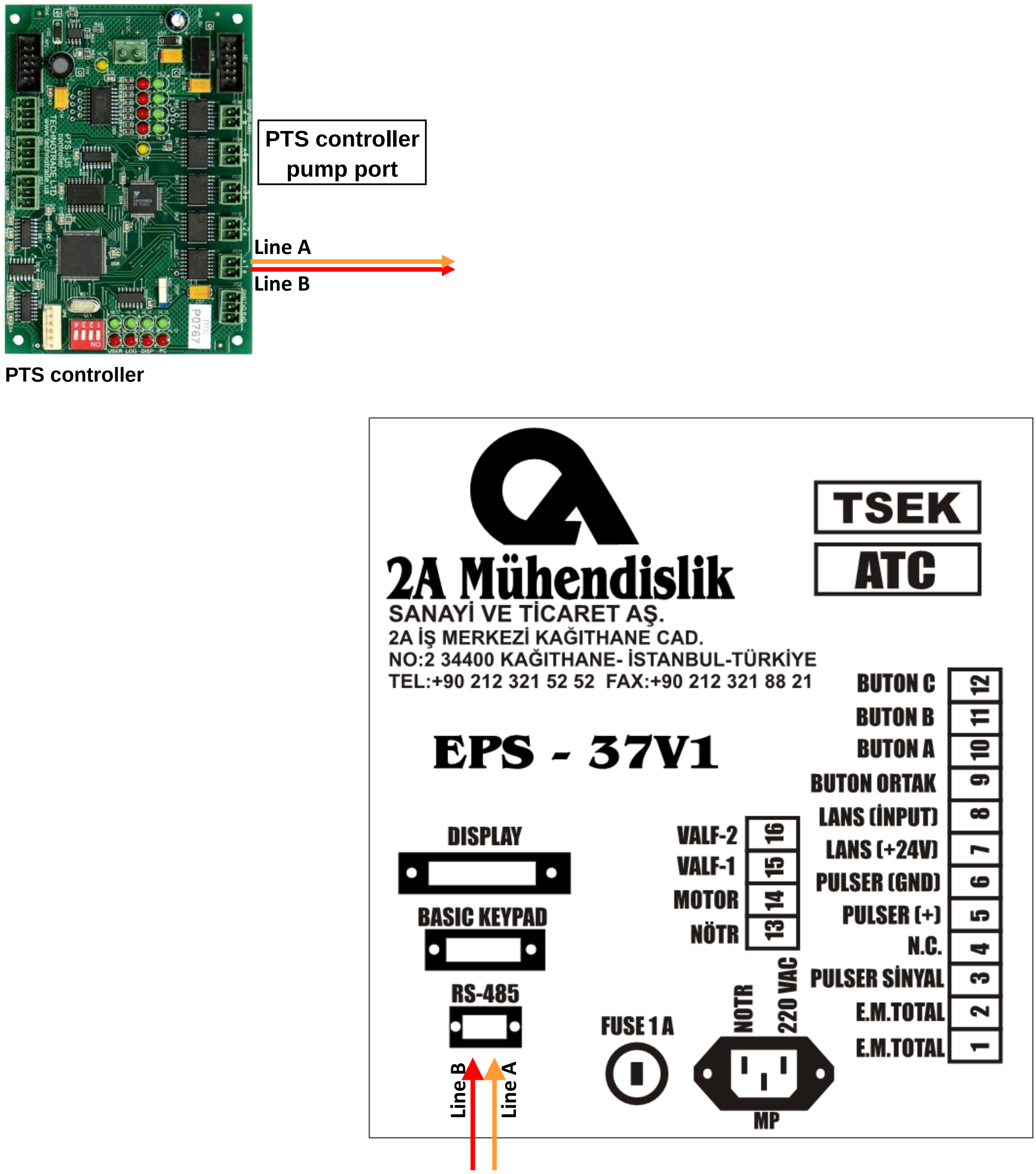


Baransay dispenser board

2A LPG dispenser connection scheme

Connection to 2A dispenser is made directly without any interface converter.

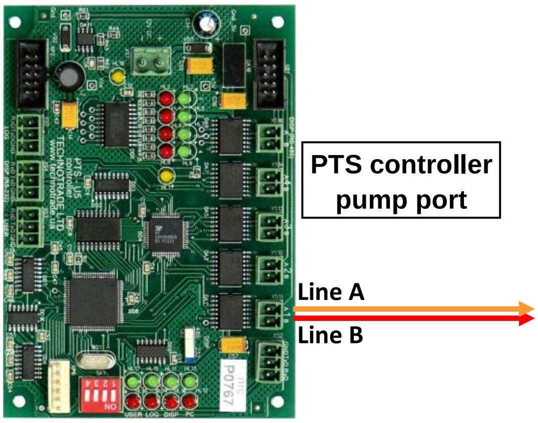
Configuration of PTS controller pump channel: protocol “*33. Dart Simplex*”, baud rate “*4. 9600*”.



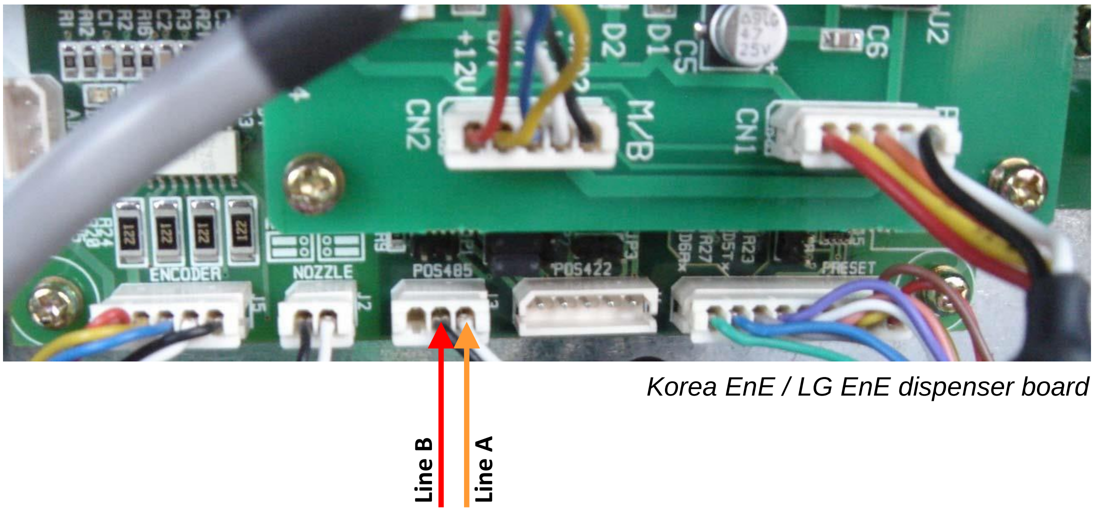
Korea EnE (LG EnE) dispenser connection scheme

Connection to Korea EnE (LG EnE) dispenser is made directly without any interface converter.

Configuration of PTS controller pump channel: protocol “*20. EnE Korea*”, baud rate “*4. 9600*”.



PTS controller

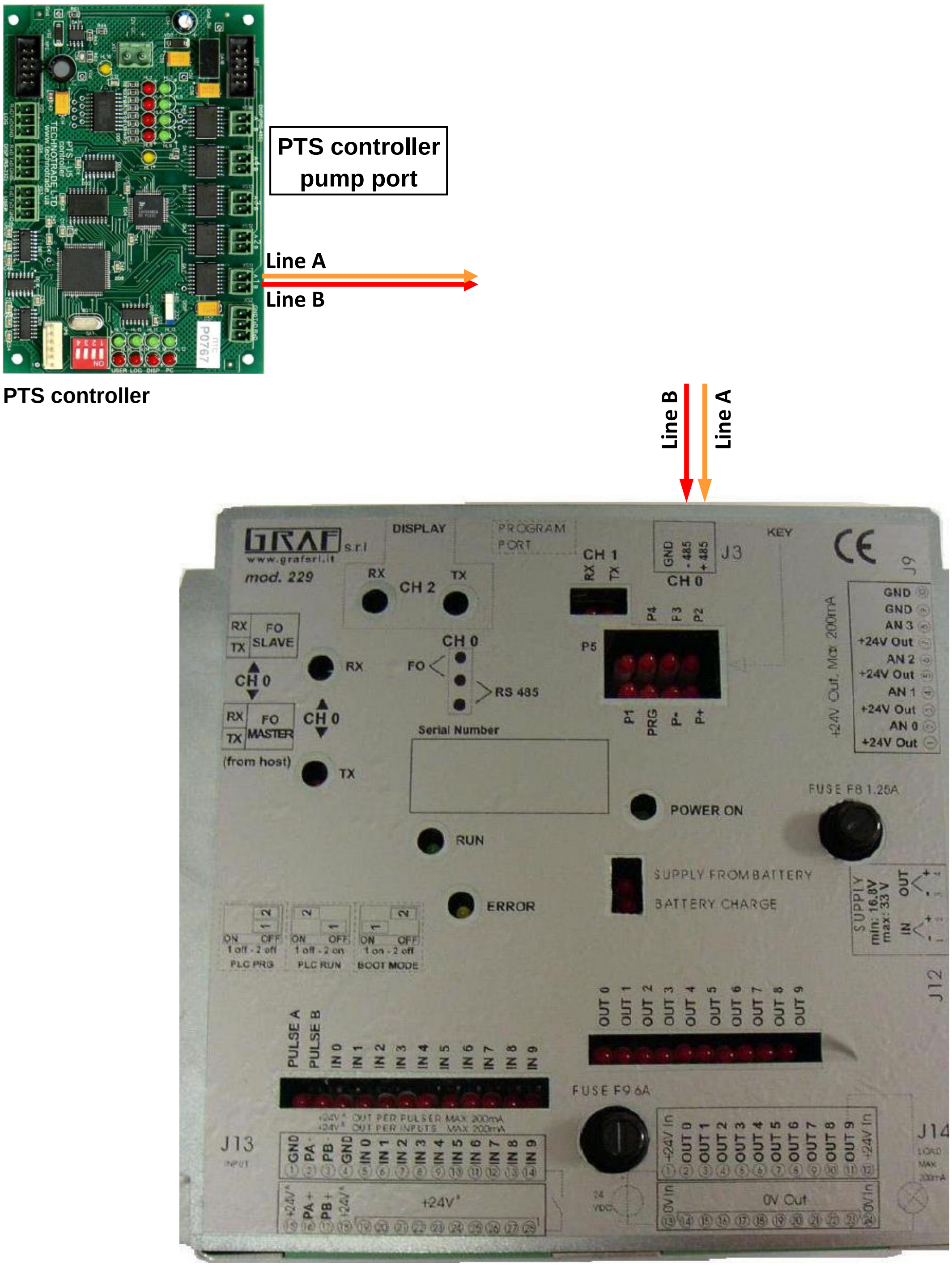


Korea EnE / LG EnE dispenser board

SAFE dispenser connection scheme

Connection to SAFE dispenser is made directly without any interface converter.

Configuration of PTS controller pump channel: protocol "*9. SAFE Graf*", baud rate "*4. 9600*".

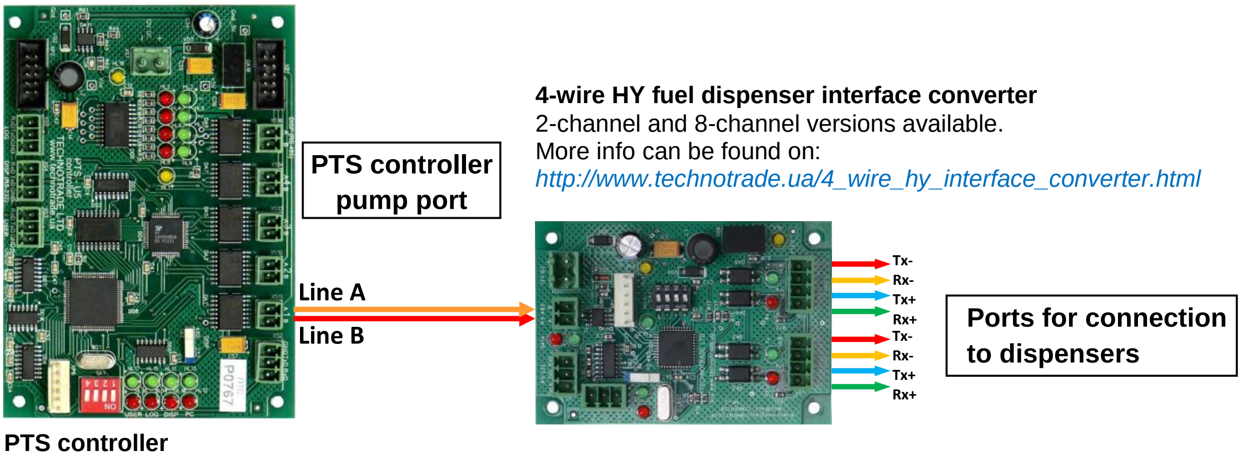


SAFE Graf electronic head PMII

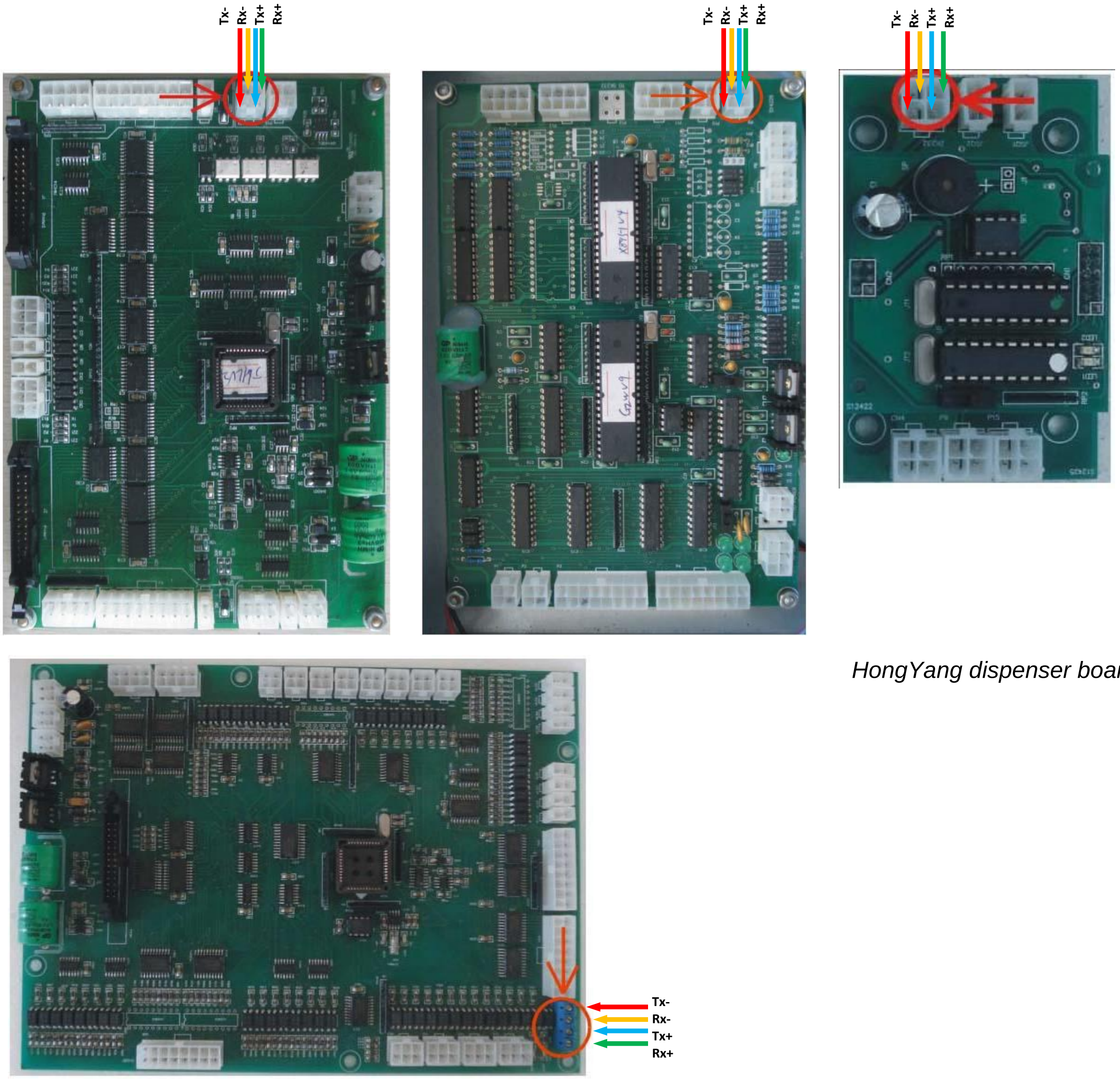
HongYang dispenser connection scheme

Connection to HongYang dispenser dispenser is made through 4-wire HY interface converter, which provides connection of RS-232/RS-485 interfaces to 4-wire current loop interface.

Configuration of PTS controller pump channel: protocol "*38. HongYang 886*" (in case of single-product dispenser) or "*42. HongYang MPD 886*" (in case of multi-product dispenser), baud rate "*1. 2400*".



4-wire HY fuel dispenser interface converter
2-channel and 8-channel versions available.
More info can be found on:
http://www.technotrade.ua/4_wire_hy_interface_converter.html

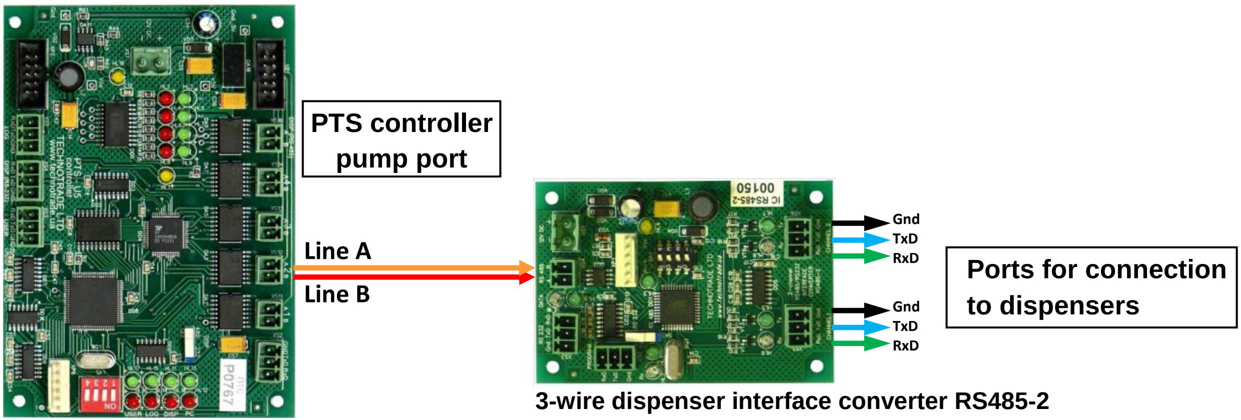


HongYang dispenser boards

Lanfeng dispenser connection scheme

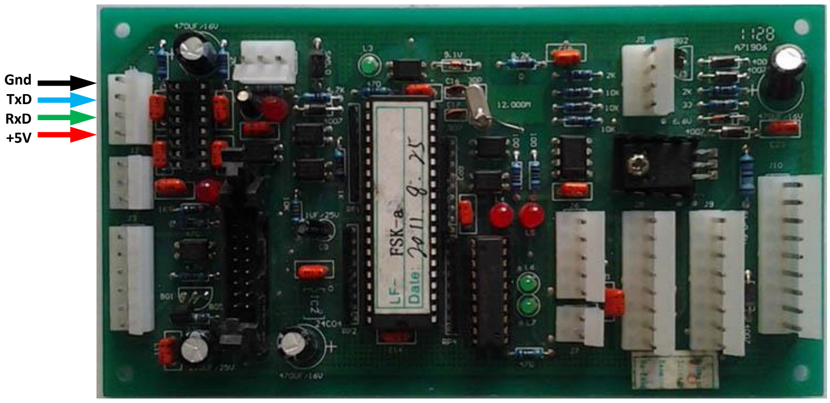
Connection to Lanfeng dispenser dispenser is made through 3-wire interface converter, which provides connection of RS-485 interfaces to RS-232 interface.

Configuration of PTS controller pump channel: protocol “*30. LANFENG*”, baud rate “*4. 9600*”.



PTS controller

3-wire dispenser interface converter RS485-2
More info can be found on:
http://www.technotrade.ua/3_wire_485_interface_converter.html

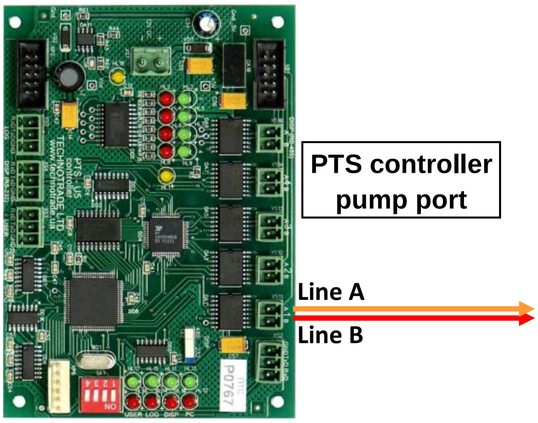


Lanfeng RS-232 dispenser board

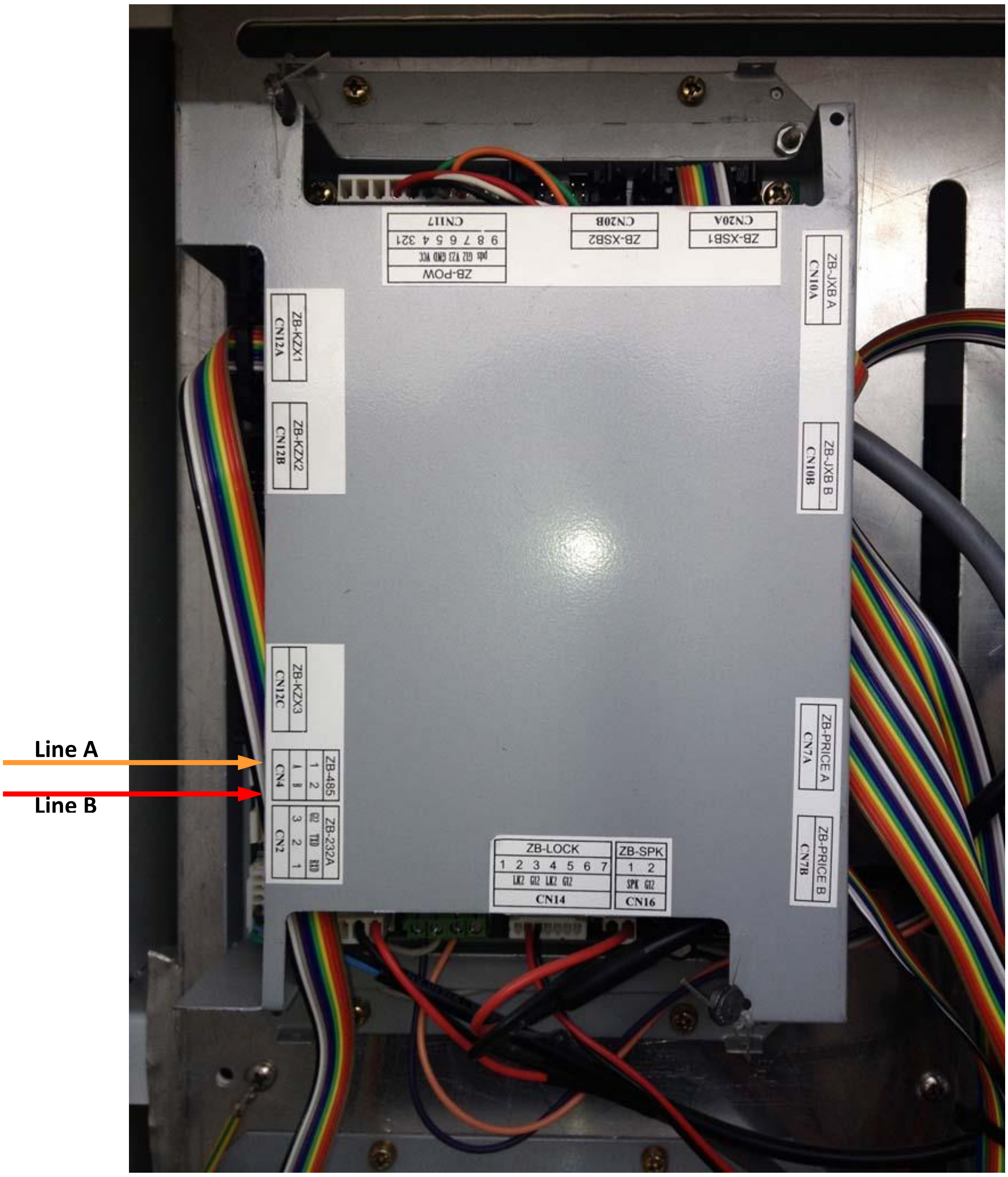
Sanki dispenser connection scheme

Connection to Sanki dispenser is made directly without any interface converter.

Configuration of PTS controller pump channel: protocol “*26. Blue Sky*”, baud rate “*2. 4800*”.



PTS controller

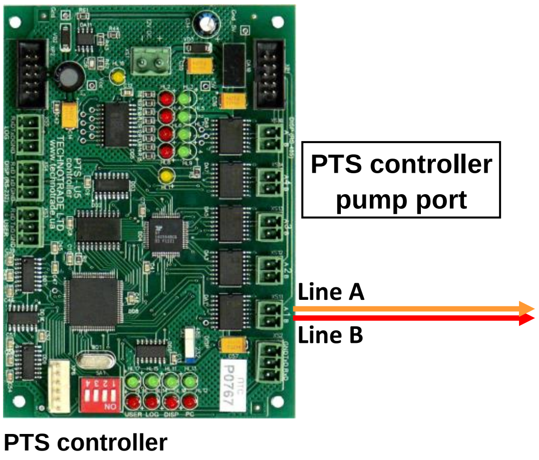


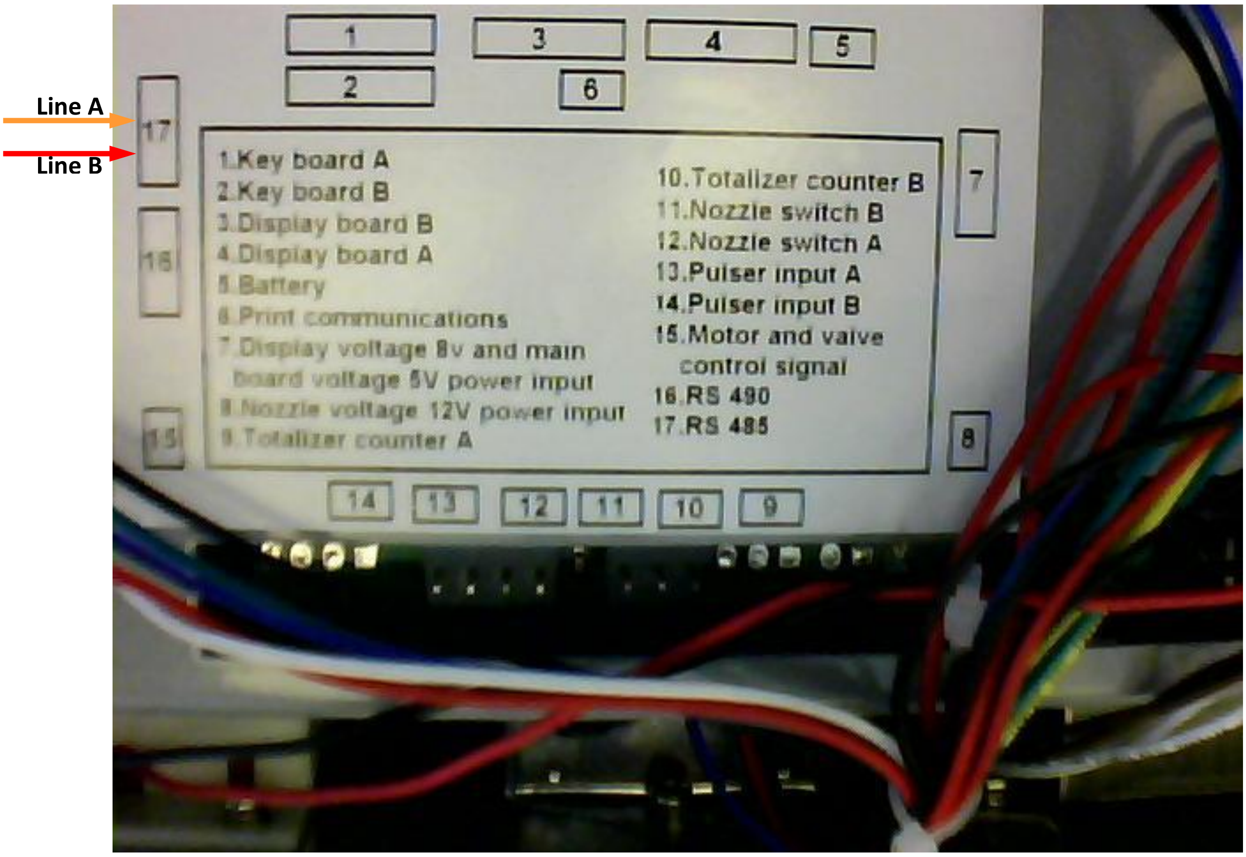
Sanki dispenser board

Blue Sky dispenser connection scheme

Connection to Blue Sky dispenser is made directly without any interface converter.

Configuration of PTS controller pump channel: protocol “*26. Blue Sky*”, baud rate “*2. 4800*”.





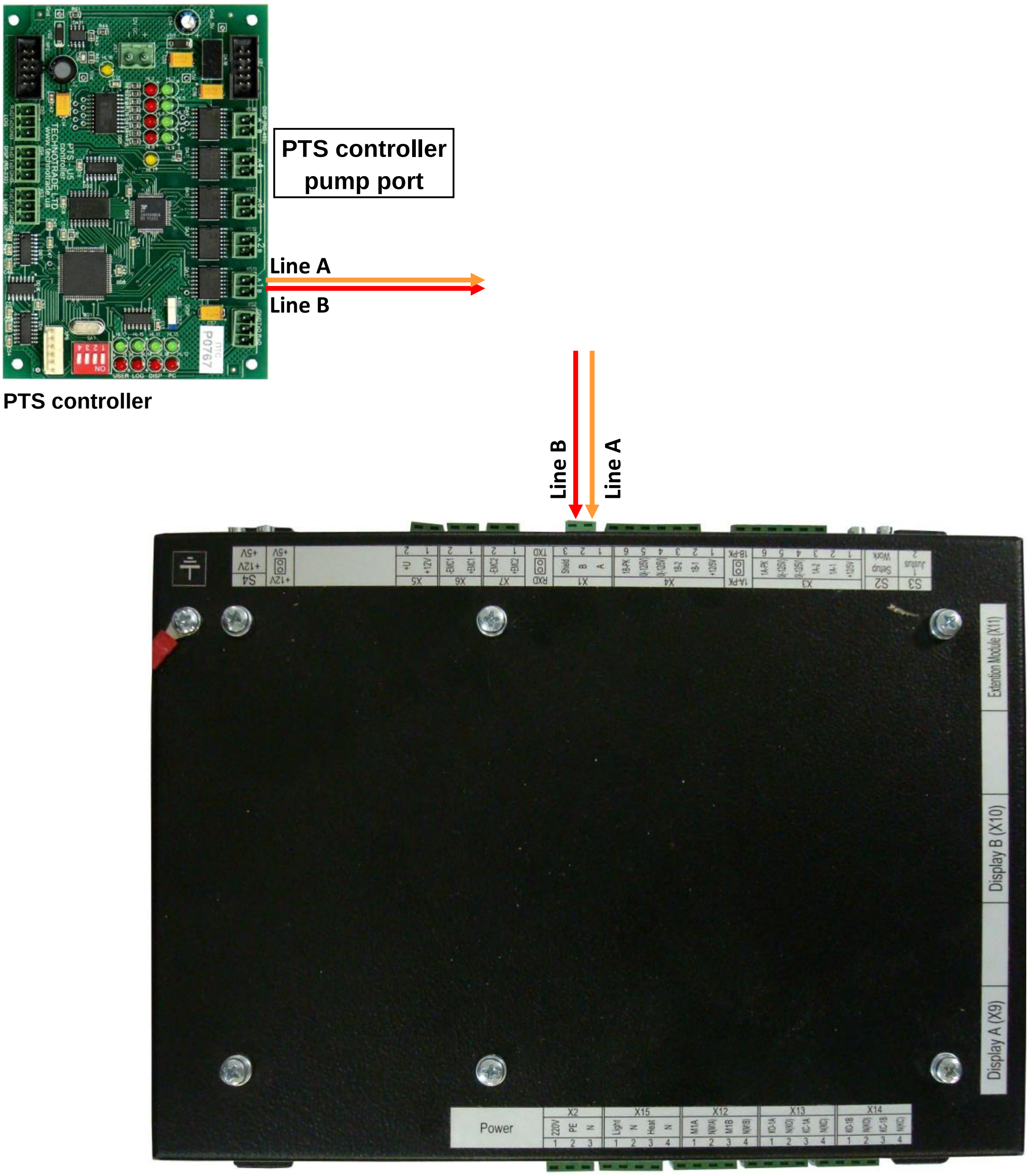
1.Key board A
2.Key board B
3.Display board B
4.Display board A
5.Battery
6.Print communications
7.Display voltage 8v and main board voltage 5V power input
8.Nozzle voltage 12V power input
9.Totalizer counter A
10.Totalizer counter B
11.Nozzle switch B
12.Nozzle switch A
13.Pulser input A
14.Pulser input B
15.Motor and valve control signal
16.RS 490
17.RS 485

Blue Sky dispenser board

Topaz dispenser connection scheme

Connection to TOPAZ dispenser is made directly without any interface converter.

Configuration of PTS controller pump channel: protocol “*41. TOPAZ*”, baud rate “*2. 4800*”.

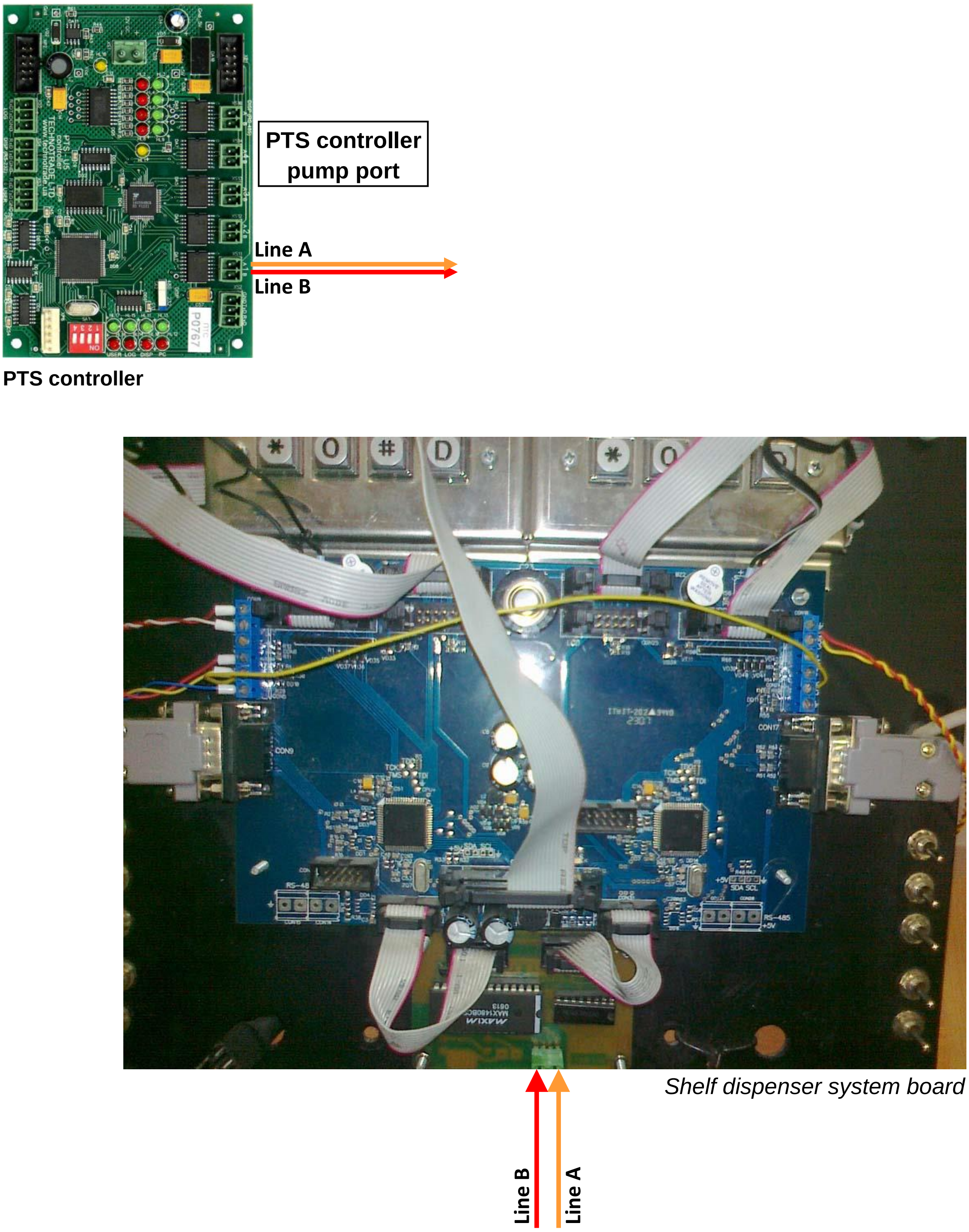


Topaz dispenser system board

Shelf dispenser connection scheme

Connection to SHELF dispenser is made directly without any interface converter.

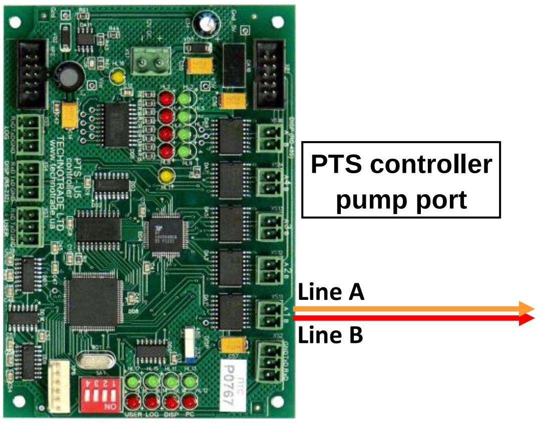
Configuration of PTS controller pump channel: protocol “*16. SHELF*”, baud rate “*4. 9600*”.



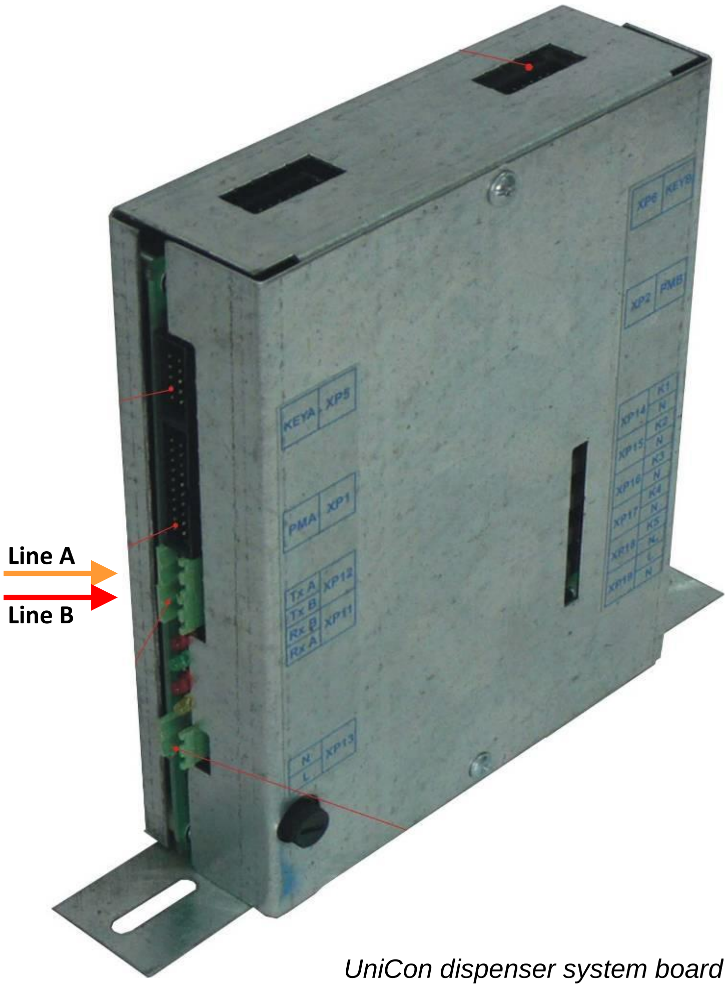
UniCon dispenser connection scheme

Connection to UniCon dispenser is made directly without any interface converter.

Configuration of PTS controller pump channel: protocol “*16. UniPump*”, baud rate “*4. 9600*”.



PTS controller



UniCon dispenser system board

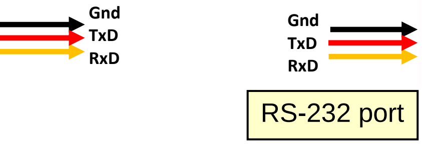
EXAMPLES OF ATG SYSTEMS CONNECTION SCHEMES

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 ATG ports of PTS controller (RS-232 interface).

Configuration of PTS controller – protocol "*1. GILBARCO Veeder Root*", baud rate is selected to be equal to set in TLS ATG system.

PTS controller



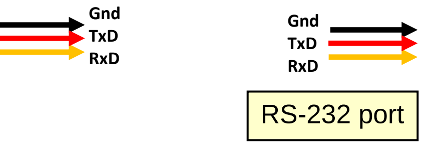
Gilbarco Veeder Root TLS-350 ATG console

Start Italiana console connection scheme

Connection to Start Italiana console is made directly to one of ATG ports of PTS controller (RS-232 interface).

Configuration of PTS controller – protocol "*1. GILBARCO Veeder Root*", baud rate is selected to be equal to set in Start Italiana console.

PTS controller

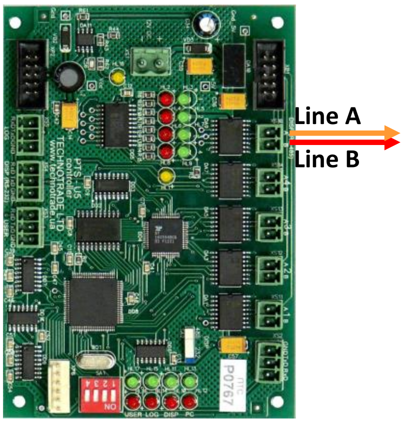


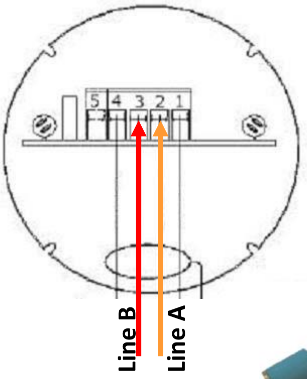
Start Italiana ATG console

Start Italiana wired probes connection scheme

Connection to Start Italiana probes is made directly to DISP (RS-485) port of PTS controller (RS-485 interface).

Configuration of PTS controller – protocol "*2. START ITALIANA SMT-XMT*", baud rate "*4. 9600*".



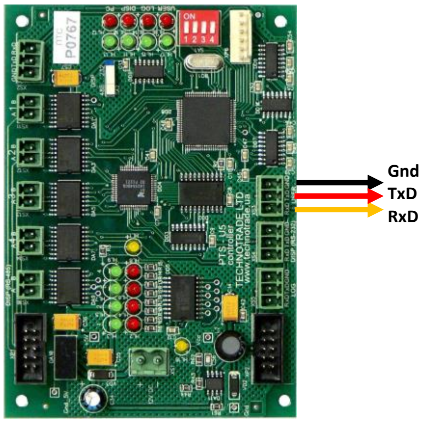


Start Italiana probe (RS-485)

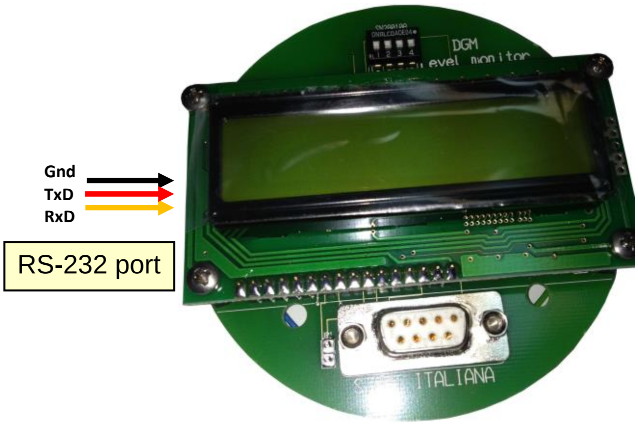
Start Italiana wireless probes connection scheme

Connection to Start Italiana wireless is made through Start Italiana RF protocol converter board to one of channels of ATG port of PTS controller (RS-232 interface).

Configuration of PTS controller – protocol "*2. START ITALIANA SMT-XMT*", baud rate "*4. 9600*".



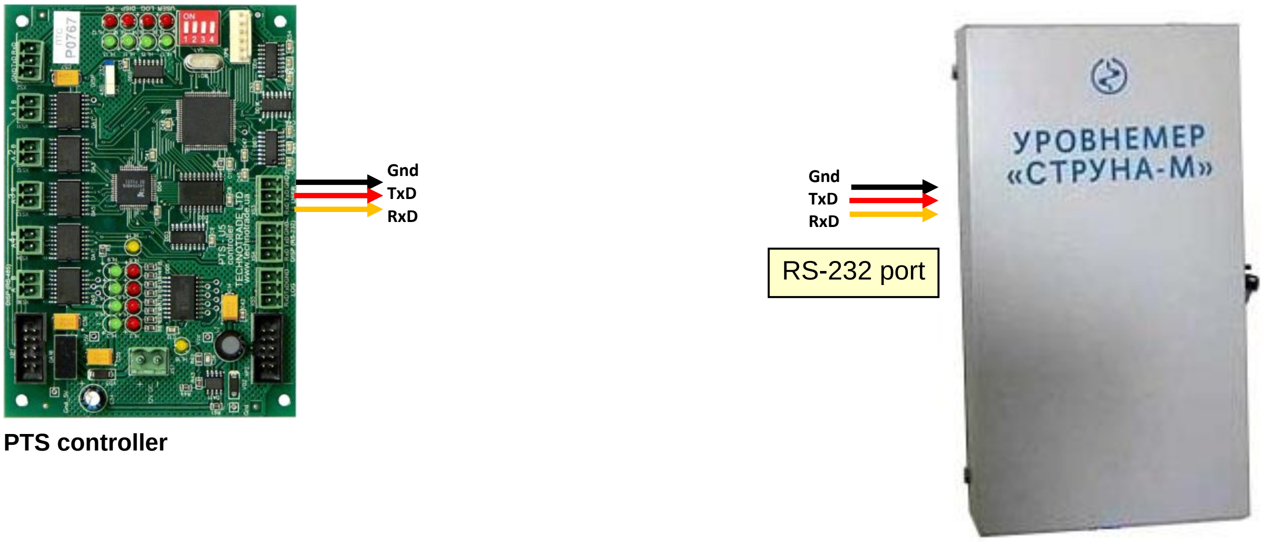
PTS controller



Start Italiana RF protocol converter board

Struna ATG system connection scheme

Connection to Struna system is made directly to one of ATG ports of PTS controller (RS-232 interface). Configuration of PTS controller – protocol *"4. STRUNA Kedr spec. 1.4"*, baud rate is selected to be equal to set in Struna ATG system.

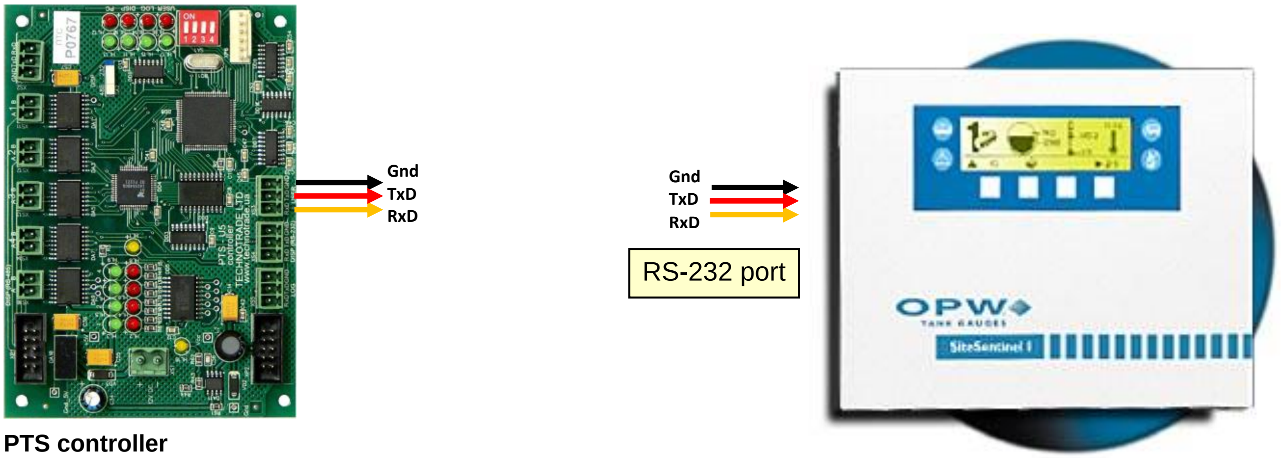


PTS controller

Struna ATG calculation unit

OPW Site Sentinel ATG system connection scheme

Connection to OPW system is made directly to one of ATG ports of PTS controller (RS-232 interface). Configuration of PTS controller – protocol "*3. PETROVEND4*", baud rate is selected to be equal to set in OPW ATG system.

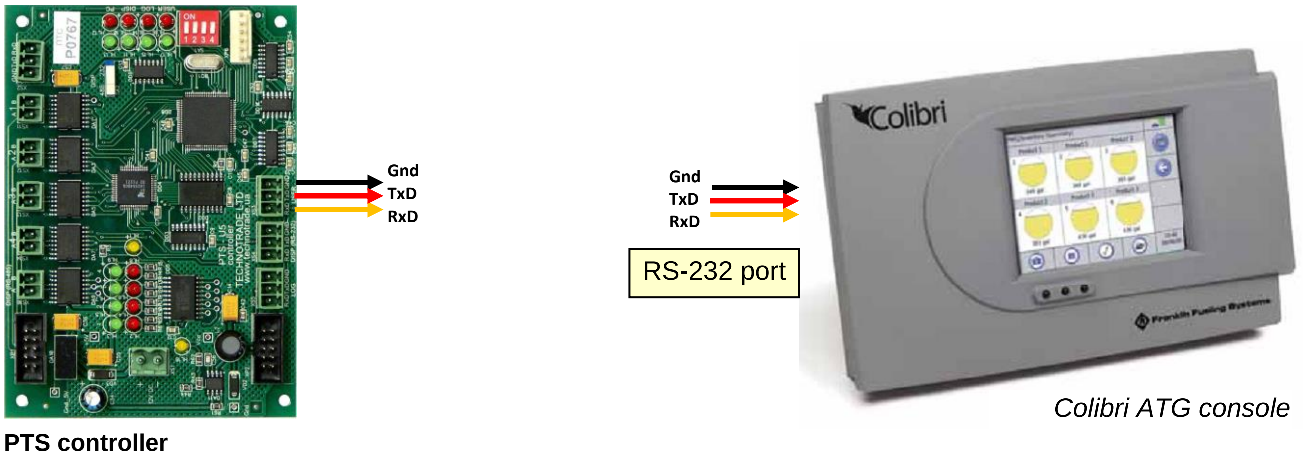


PTS controller

OPW Site Sentinel ATG console

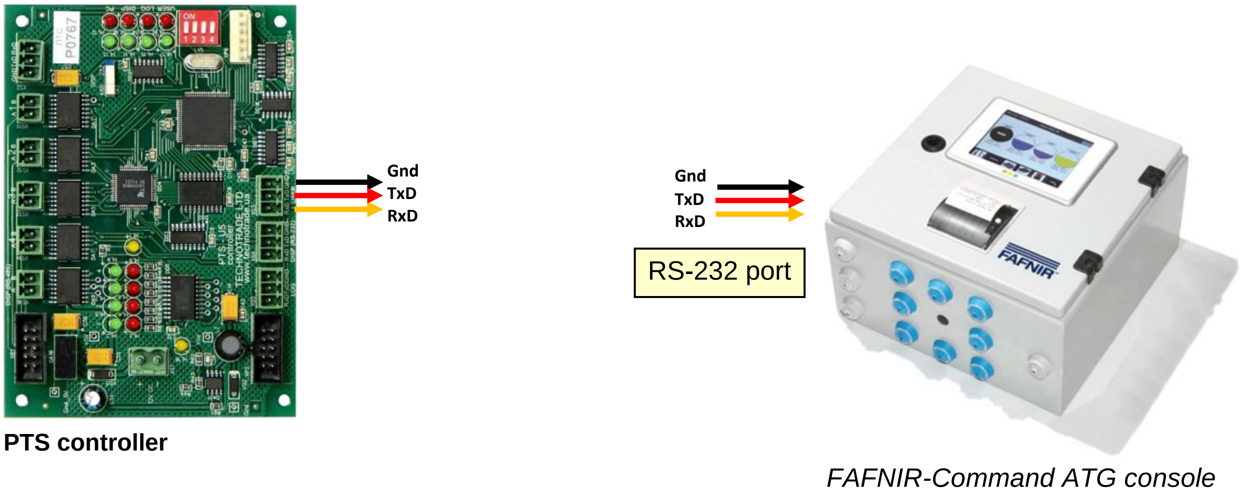
Colibri ATG system connection scheme

Connection to Colibri system is made directly to one of ATG ports of PTS controller (RS-232 interface). Configuration of PTS controller – protocol "*1. GILBARCO Veeder Root*", baud rate is selected to be equal to set in Colibri ATG system.



FAFNIR ATG system connection scheme

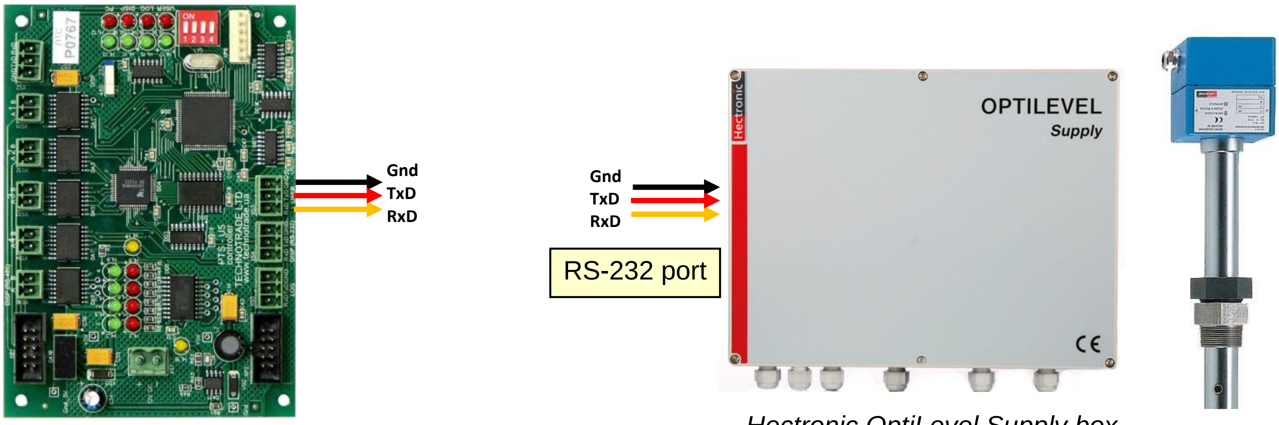
Connection to FAFNIR system is made directly to one of ATG ports of PTS controller (RS-232 interface). Configuration of PTS controller – protocol "*5. FAFNIR Visy-Quick*", baud rate is selected to be equal to set in FAFNIR ATG system.



Hectronic ATG probes connection scheme

Connection to Hectronic probes is made directly to one of ATG ports of PTS controller (RS-232 interface) to Hectronic OptiLevel Supply box.

Configuration of PTS controller – protocol “*8. HECTRONIC HLS*”, baud rate “*4. 9600*”.

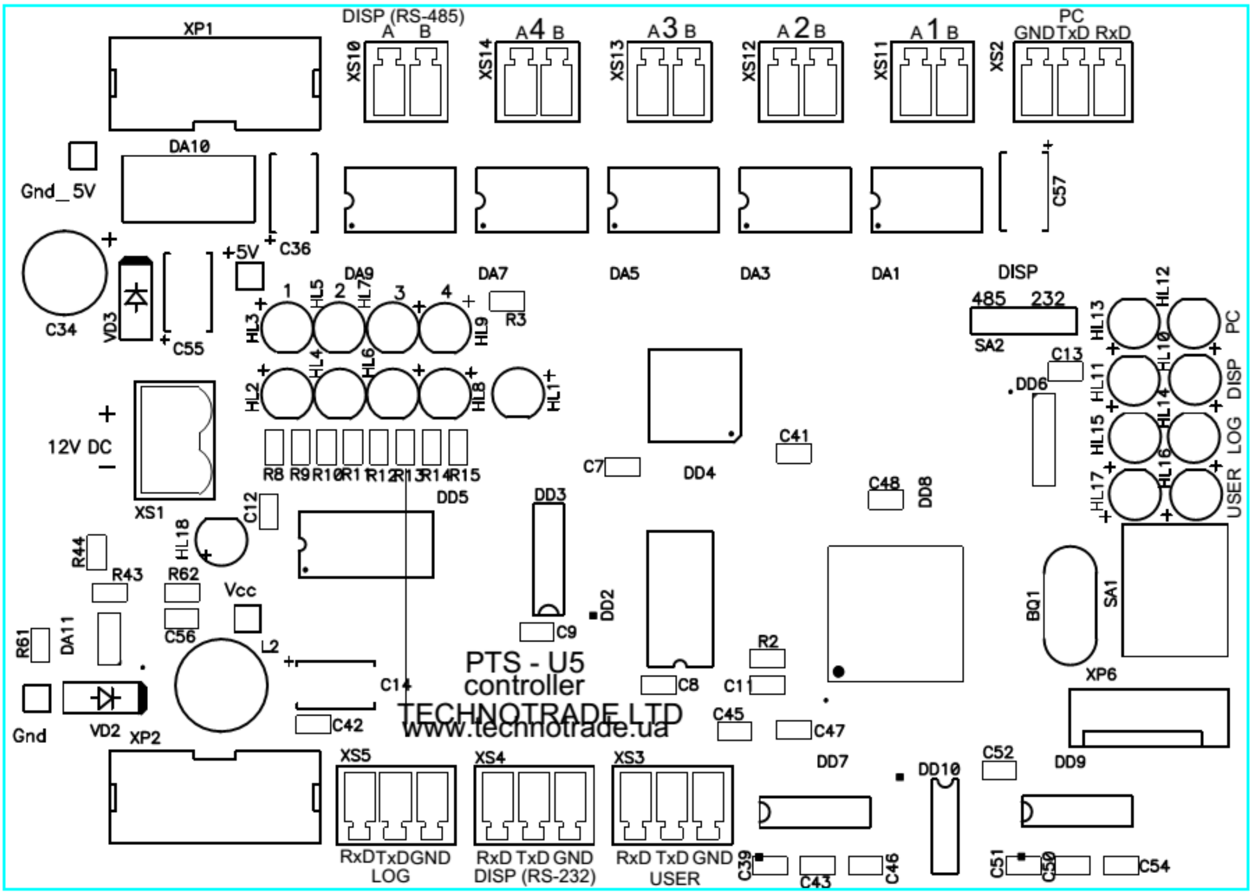


PTS controller

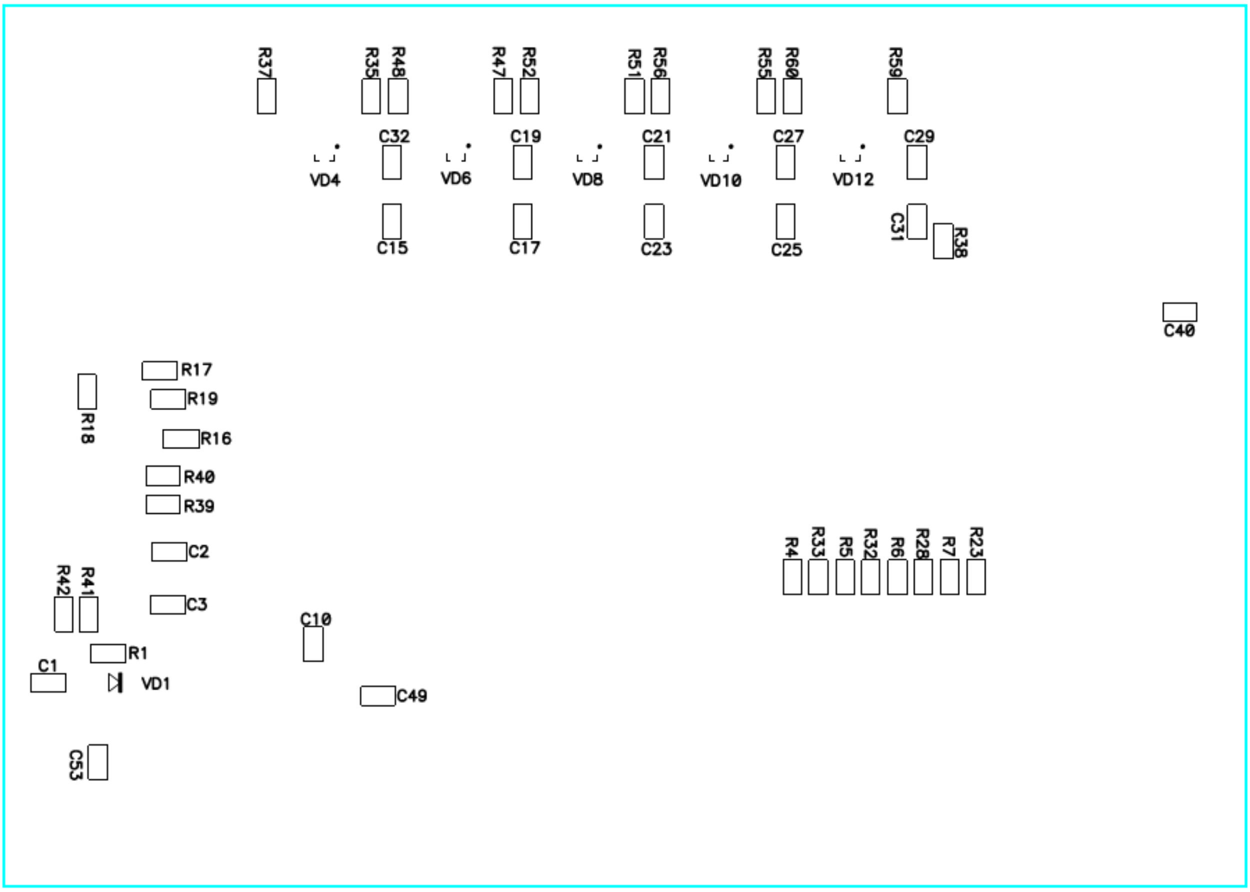
Hectronic OptiLevel Supply box

PTS CONTROLLER PCB BOARD

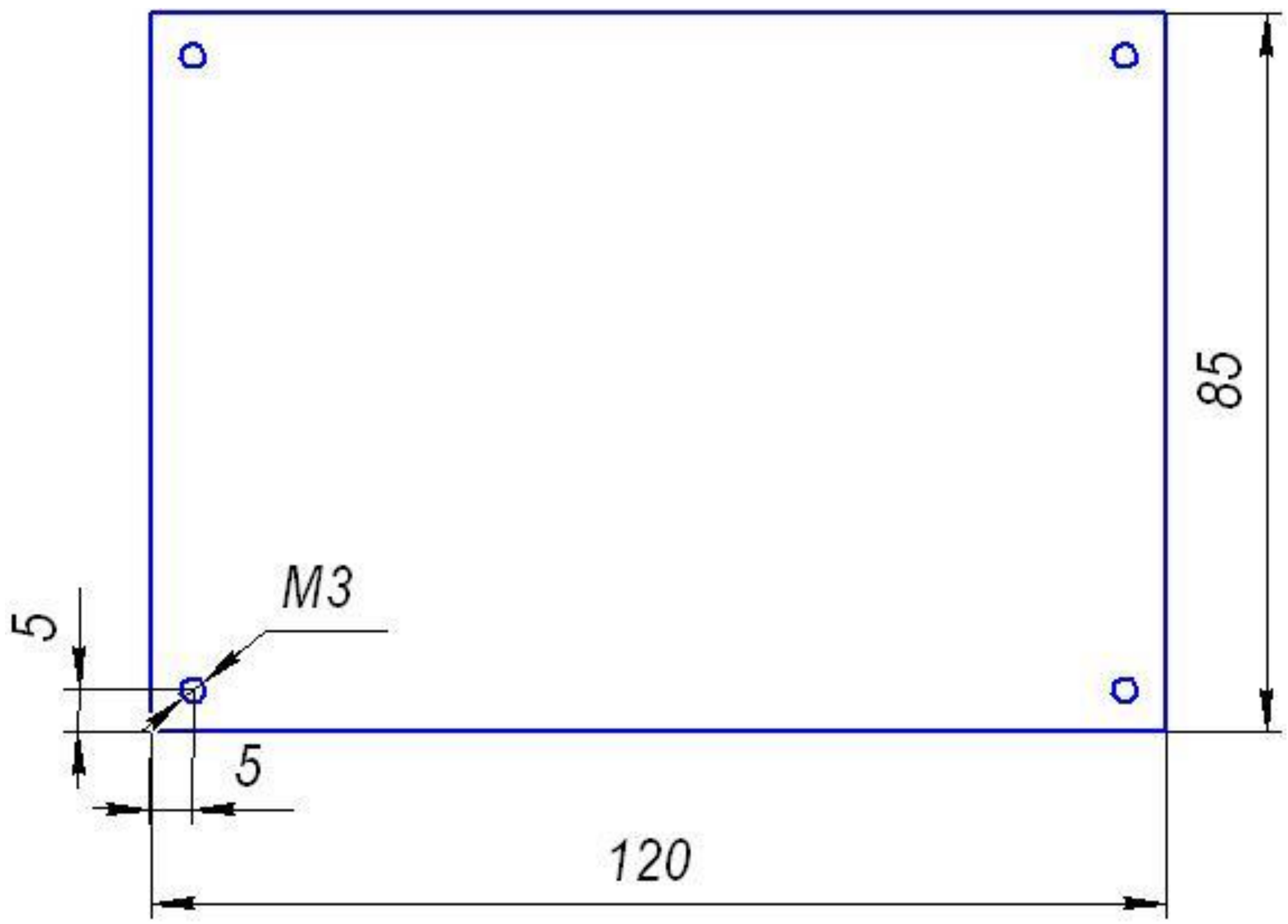
PCB board top view



PCB board bottom view

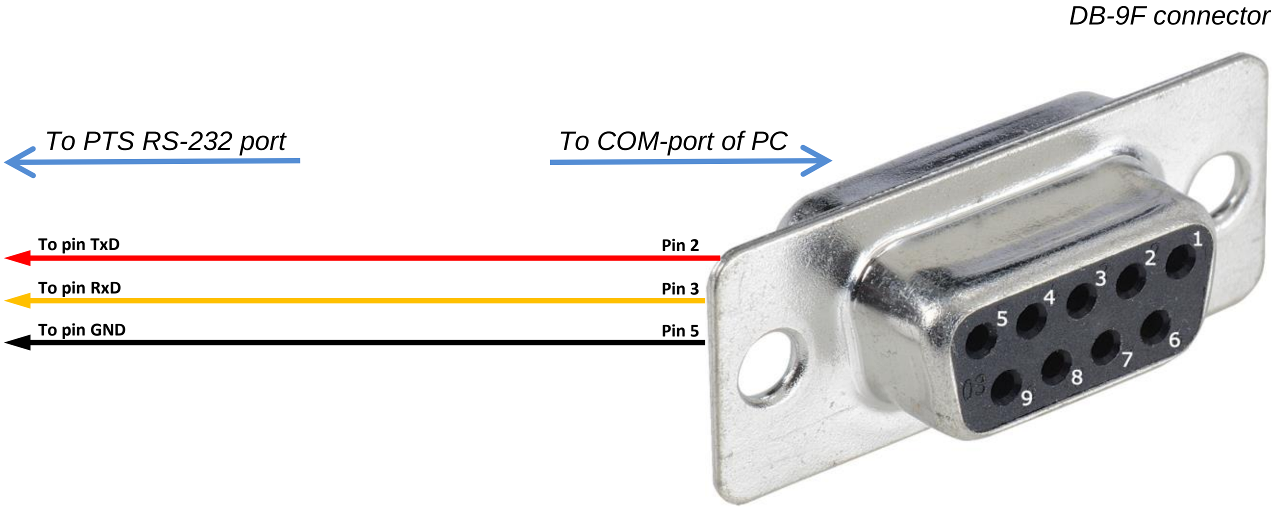


Board dimensions



PTS CONTROLLER RS-232 PORT CABLE CONNECTOR

Pinout of cable used for connection RS-232 ports of PTS controller to PC:


DB-9F connector
To PTS RS-232 port
To COM-port of PC
To pin TxD
Pin 2
To pin RxD
Pin 3
To pin GND
Pin 5
1
2
3
4
5
6
7
8
9