

Web Server – WS10

INSTRUCTIONS MANUAL V2.0x B

5000702

INDEX

Index	1
Basic Functions	2
Electrical Connections	4
WS10 configuration	6
Local inputs and outputs configurations	9
Modbus RTU network configuration – WS10 as master	12
Alarms supervision configuration	14
Data logger configuration	15
HTML Page server configuration	17
Periodical data transmission configuration	20
E-mail configuration	22
Modbus RTU Network configuration – WS10 as slave	24
Modbus TCP network configuration – WS10 as server or gateway	25
Configuring the modem to start a data call – PPP client	27
Configuring the modem to receive a data call – PPP Server	29
DNS Configuration	30
Security issues on the WS10	31
Specifications	32
Appendix A	33

BASIC FUNCTIONS

The WS10 Web Server by Novus is an equipment capable of acquiring and delivering data using the internet technologies. Local and remote systems supervision, alarm notification and data logging are easy to configure and operate, with no dedicated software.

DATA ACQUISITION

The WS10 gets data from the system or process through:

- 4 built-in signal inputs
- RS232 or RS485 serial communication interface

Data acquisition over the RS485 communication interface uses the Modbus RTU protocol. The WS10 acts as a master, polling the Modbus network nodes to get the specified data to its memory.

All the data acquired from its local inputs and from external Modbus devices are stored on the WS10 memory. The user can assign identifiers for each memory position. On the configurations of the data access functions, all registers are referenced by their names.

REGISTERS

Registers are memory locations in the WS10 that contain:

- Instantaneous input or output values read or written to the WS10 built-in signal inputs.
- Instantaneous values, read from or written to, the equipments connected to the Modbus RTU network.;
- Instantaneous memory values, used to control WS10's functions, not directly connected to the physical inputs or outputs;
- The instantaneous values corresponding to the current alarms situation.

Each register has an identifier composed by a Device Name and a Tag Name. This identifier is used as a reference to the register when configuring the WS10 functions:



For registers access using ModbusTCP and ModbusRTU-Slave, numerical addresses must be assigned to the registers. These addresses may be defined in file **MAP.CFG**, described on chapter ModbusTCP.

DATA LOGGING

The WS10 can be configured to periodically log the values of a group of registers on its Flash memory. This historical logging can be sent out by e-mail or directly to a computer. This function allows the WS10 to act as a Data Logger for the variables read from the equipments in the Modbus network or from its built-in signal inputs. On a remote supervision system, this function enables the WS10 to monitor the process and send the acquired data periodically or when the communication link is established.

REMOTE DATA ACCESS

The instantaneous and historical values acquired from the system or process can be accessed by different physical ways:

- Ethernet interface;
- RS232, RS485 serial interface or Modem (conventional or mobile).

The WS10 offers five basic applications for remote data access of its registers.

- WebServer;
- Modbus TCP or Modbus RTU;
- Alarm conditions supervision;
- E-mail transmission;
- Periodical data transmission.

The WS10's multi-task operating system allows for the simultaneous utilization of all these applications

HTML PAGE SERVER APPLICATION

In this application, the WS10 presents the acquired data of the process in dynamic user-created HTML pages, that can be visualized by any computer using a Internet browser (Netscape, Internet Explorer, etc).

The WS10, connected to the company's intranet by its Ethernet interface, shares the information (application data) to any computer connected to the intranet, allowing both readings and writings of data, with or without password authorization.

In a remote application, a WS10 connected to a conventional or mobile modem can answer a "dial-up" call connection and present a supervision page to the internet browser of the computer that originated the call, allowing the supervision of remote units.

MODBUS TCP

The WS10 can be configured as a ModbusTCP server and gateway, enabling integration to SCADA supervisory systems. As a server, WS10 accepts connections from multiple SCADA clients, allowing read and write access to its internal registers. This registers are associated to WS10's local I/O, memory positions or external ModbusRTU devices. As a gateway, WS10 forwards messages between ModbusTCP and ModbusRTU networks, acting as a protocol converter. Both modes may be used simultaneously, while all other WS10 functions remain fully functional. ModbusTCP communication can be used over Ethernet or phone line (fixed or mobile).

ALARM SUPERVISION

In this application, the WS10 continuously compares the content of its registers against user programmed limits. The result of this comparison is assigned to a new register that can be used to trigger a local output, write in a remote equipment through the Modbus RTU, send an e-mail or send information to a remote server.

TRANSMISSION OF E-MAIL OR MOBILE PHONE MESSAGES

The WS10 can be configured to send e-mails automatically to an address list. The transmission can be periodical or by exception, based on register values. The e-mails are sent through the Ethernet or Modem and have a dynamic content, sending process data to the recipient. The message sending to cell phones can be done by email (depending on the routing service offered by the mobile phone operator) or directly to a GSM modem connected to the WS10. The e-mail may include, as an attached file, the historical data stored in its memory.

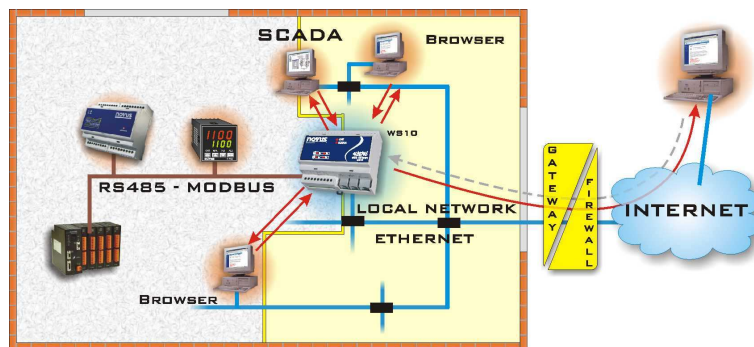
PERIODICAL DATA TRANSMISSION

The data acquired by the WS10 can be sent periodically to any accessible computer in the Ethernet or Internet (using a conventional or mobile Modem), being received by a TCP server application and stored in a place and format defined by the user. Both instantaneous and historical register values can be sent to the remote server.

APPLICATION TOPOLOGY

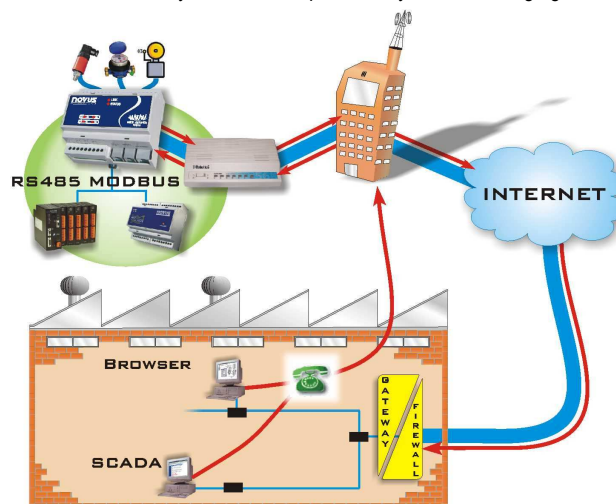
LOCAL OPERATION

Connected to a local network, WS10 can bridge a ModbusRTU network to existing SCADA using Ethernet, expanding data accessibility to virtually all networked computers using internet browser. WS10 can also log historical data, monitor alarm conditions and send e-mail messages in periodical or exception basis. WS10 is an alternative to SCADA in small applications. When the network connection to the internet is properly configured, WS10 data can be accessed from external locations, both by SCADA systems and browsers. The following figure shows WS10 in local application.



REMOTE OPERATION

Installed on a remote site, WS10 can connect through a modem to a fixed (optional internal modem) or cellular phone line. Using this connection, WS10 can access internet and send e-mail messages on abnormal conditions or periodically. WS10 can also transfer historical data to a networked computer. Its ability to answer data calls allows dial-up access using SCADA or browser. Remotely installed, WS10 allows additional functionalities when compared to conventional remote units that can only transfer data periodically. The following figure shows WS10 in a remote application.



CONFIGURATION

The WS10 configuration is accomplished with usual Internet tools available in most computers: Telnet and FTP (File Transfer Protocol). Telnet is used for IP address configuration, and FTP it is used for transferring the configuration files to the WS10.

The various functions of the WS10 are enabled by the presence of the correspondent configuration files, containing the parameters required for configuring an application.

ELECTRICAL CONNECTIONS

POWER SUPPLY

Terminals 17 and 18 are the power supply inputs of the WS10, accepting any voltage input between 100 and 240 Vac. A few seconds after power-on the STATUS indicator blinks, indicating the WS10 is running.

The mains power should be one proper for instrumentation or computers.

INPUTS

The built-in WS10 inputs can be used to measure the following types of signals:

- Analog voltage between 0 to 5 V
- Analog current between 0 to 20 mA (or 4 to 20 mA with 20 % resolution loss)
- Digital logic level: Level 0 for signals between 0 and 1V, Level 1 for signals between 4 and 5 V.
- Dry contact digital signal.

The connections are done on terminals 1 to 8, observing voltage limits and polarities. Sensor cables should run separately from power signals, in grounded conduits.

The selection between digital and analog inputs, as well as the setting of the ranges for these inputs, are described in the section *Configuration of local inputs*. For current type inputs, it is required to change internal jumpers in the WS10.

For the digital inputs, it is possible to associate a function for pulse integration, useful with flow transducers with pulse output. For more information, refer to *Configuration of local inputs and outputs*.

HARDWARE CONFIGURATION FOR CURRENT INPUT

For the mA inputs, it is necessary to short circuit internal jumpers in the WS10. There is a separate jumper for each input.

Attention: The following actions require the work of a skilled hardware technician. Use appropriate soldering iron and solder.

1. Open the WS10 back cover.
2. Find the locations of jumpers J1, J2, J3 e J4. The jumpers are identified as follows.

Jumper	Input	Opened	Closed
J2	Input 1	Voltage or switch	Current
J3	Input 2		
J4	Input 3		
J1	Input 4		

3. Using solder, close or open the jumper that corresponds to the desired input and function.
4. Re-install the WS10 back cover.

OUTPUTS

The WS10 has 2 relay outputs with SPST contacts: Output 1 (terminals 10 and 12) and Output 2 (terminals 13 and 15). Wiring and loads must comprise the specification.

Noise suppressors (RC filters) should be installed in parallel to contactors and solenoids coils driven by the WS10 or placed close to it.

SERIAL INTERFACES

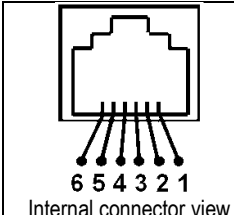
The WS10 provides up to 2 serial communication interfaces. The label placed in the WS10 lateral side identifies the installed serial interfaces.

The *SERIAL 1* interface is used to communicate with remote equipment using the master Modbus RTU protocol. It is usually an RS485 interface.

The *SERIAL 2* interface is intended for communicating with Modems, either internal (provided as an optional) or external. This interface can be configured to allow access to the WS10 configuration functions. In the standard WS10 configuration, *SERIAL 2* is a RS232 interface.

The figure below shows the RJ12 terminal assignments for the *SERIAL 1* and *SERIAL 2* interfaces.

Terminal	RS232	RS485	Modem interno
1	+5 V / 20 mA Output	+5V / 20 mA Output	-
2	Signal GND	Signal GND	-
3	CTS	-	Telephone lime
4	RxD	-	Telephone lime
5	TxD	D1 (or D)	-
6	RTS	D0 (or D\)	-



USING THE RS232 WITH RS232/RS485 CONVERTERS

The WS10 is ready to accept an external RS232 / 485 converter on its *SERIAL 2* (RS232) communication port. Converters like the Novus ISO485-1 and ISO485-2, in the RTS mode, can be used. The DTR signal is kept fixed in +5V and used to power the RS232 side of the converter. The CTS signal is also kept fixed in +5 V to enable the WS10 for transmitting. The RJ12 to DB9 or DB25 cable adaptor is described in the table below. It is necessary to wire all the signals as shown.

RJ12	Male DB9	Male DB25	Description
1	4	20	DTR – Fixed in +5 V
2	5	7	Comum
3	4	20	CTS – Fixed in +5 V
4	2	3	RXD
5	3	2	TXD
6	7	4	RTS

USING THE RS232 WITH AN EXTERNAL MODEM

External Modems can be connected to the RS232 serial port of the WS10. A cable connecting the RJ12 to the DB9 and DB25 connectors needs to be assembled, following the descriptions of the table below. The configuration shown implements the RTS/CTS flow control.

RJ12	Male DB9	Male DB25	Description
1	4	20	DTR – Fixed in +5 V
2	5	7	Comum
3	8	5	CTS
4	2	3	RXD
5	3	2	TXD
6	7	4	RTS

ETHERNET

The Ethernet connection follows the RJ45 Standard. It is recommended to use category 5 cables. For direct connection of the WS10 to a computer, use cross wiring. When a hub or switch is used, use conventional wiring.

WS10 CONFIGURATION

Installing and configuring the WS10 require networking expertise. Consult your company's network administrator.

GENERAL INFORMATION

The WS10 configuration is accomplished through the Ethernet interface making use of ordinary network programs: **FTP** (File Transfer Protocol – for transferring files to/from the WS10) and **Telnet** (to access the configuration console). The FTP is normally provided with the Internet navigators (Internet Explorer or Netscape); other FTP programs can also be used. The Telnet application is available in the operating system of your computer, although other Telnet client applications can be used.

In order to configure the WS10, it must be powered and connected to the Ethernet through a hub or switch, or directly to the Ethernet interface of a computer (in this case a cross cable must be used). The WS10 IP address must be known to get FTP or Telnet access to the WS10.

The WS10 factory settings are:

IP = 192.168.200.200
NetMask = 255.255.255.0
Gateway = 192.168.200.1
DHCP disabled

The computer used to access the WS10 must be on the same sub-network of the WS10. To access the WS10 with the factory settings, the IP addressing configuration of the computer must be:

IP = 192.168.200.xxx (where xxx can be any value except 200, already assigned to the WS10)
 NetMask = 255.255.255.0

TELNET CONFIGURATION

To change IP configurations, it is necessary to establish a Telnet connection with the WS10 to get access to configuration console. Run your preferred Telnet application and connect your computer to the WS10 IP address, using the Telnet standard port. When the WS10 is found, the message "Novus WS10 Telnet Session" is presented. The WS10 prompts for Username and password. The factory values are:

Username: telnet
Password: telnet

Once connected, the message "User logged in" is showed indicating the connection success. With the Telnet application running, the WS10 gives access to commands similar to the ones of the DOS operating system. The commands are executed when the Enter key is pressed

Through the Telnet console, a user can modify important configurations in the WS10. Only capable users should use this feature. Use only the commands listed on this section.

The following commands are related to the configuration of the WS10 IP address and may be needed in the configuration process.

IPCFG: Shows current IP address.

IP aaa.bbb.ccc.ddd: Defines a new IP address for the WS10.

NETMASK aaa.bbb.ccc.ddd: Defines a new subnet mask for the WS10.

GATEWAY aaa.bbb.ccc.ddd: Defines the gateway address.

DHCP n: n = 0 disables the dynamic addressing through DHCP; n = 1 enables dynamic addressing.

REBOOT: forces the WS10 reboot. A new IP address is effective only after the reboot. Following the reboot, the Telnet connection is lost. To re-establish the connection, use the new IP address. In the case of the new IP address falls in another subnet, it will be necessary to change the address of the computer used in the configuration to this same subnet.

WS10 -v: Shows the Web Server software version.

The same console commands may be available on the SERIAL 2 communication port, provided it is RS232 and is not being used as a PPP client or Server. Factory settings for the communication parameters are: Baud Rate 19200 bps, no parity, 8 bits, 1 stop bit. Use a RS232 cable and PC serial communication program to communicate over this port. This feature may be factory disabled depending on the WS10 version and hardware options.

FILE TRANSFER USING FTP

The configuration of the WS10 is defined by text files stored in its memory. These files can be transferred to a computer, modified, and copied back into the WS10. The new configurations will be effective when the WS10 is rebooted. The WS10 incorporates a FTP Server compatible to the majority of the FTP clients available, like the ones provided with the navigators Internet Explorer and Netscape. The Windows Explorer can also be used for this purpose.

To establish a FTP connection with the WS10, write the WS10 IP address in the navigator address bar, as shown below:



If prompted, inform username and password. The FTP factory settings are:

Username: ftp
Password: ftp

The WS10 factory settings allow anonymous FTP connection also.

The WS10 internal files and directories are shown. It is possible to drag the files and directories from the WS10 to any other local directory or network location, and the other way around.

Configuration files that require modification can be copied to a computer temporary directory, edited, saved and stored back in the WS10. The modifications performed on the configuration files will become effective in the next boot.

In some navigators, the FTP access may refuse connection even with the correct password. The way around is to inform username and password along with the IP address. The example below illustrates this procedure for username *user* and password *pass*:

ftp://user:pass@192.168.200.200

If the error persists, not allowing the FTP connection, perform a WS10 reboot and try again using the username and password in the address line as suggested above.

To solve further FTP connection errors, we recommend the usage of dedicated FTP client software. There are many shareware and freeware options available for download on the Internet. In the WS10 CD are included some of these FTP softwares.

CONFIGURATION OF THE IP ADDRESS

The IP address is used in every communication with the WS10. The IP address must be known and properly kept.

The WS10 with an unknown IP address is useless. Novus provides a low level IP detection program to scan the network looking for a WS10 with unknown IP address.

Once in the Ethernet, the WS10 can only be accessed by computers in the same subnet. So, the computer used to access the WS10 must have its address in the same subnet address range of the WS10. If not, the computer must be temporarily addressed to the same WS10 subnet, returning to its original configuration after the configuration is completed.

PROCEDURES FOR GIVING THE WS10 A FIXED IP ADDRESS

The following steps are required to assign a new IP address to the WS10:

- Get from the network administrator a IP address for the WS10 and the configurations for the Gateway and subnet mask.
- Establish a Telnet connection with the WS10 using its current IP address. The computer must belong to the same WS10 subnet.
- Disable the WS10 DHCP client typing: `DHCP 0` and Enter.
- Type a new IP address and Enter. Example: `IP 10.1.1.80`
- Configure the subnet mask typing `NETMASK` followed by the mask and Enter. Example: `NETMASK 255.255.255.0`
- Configure the gateway address by typing `GATEWAY` followed by the address. Example: `GATEWAY 10.1.1.1`
- Re-initialize the WS10 by typing `REBOOT` followed by Enter.
- Upon reboot, the WS10 will assume the new network configuration, and will only be reached by computers in the same subnet.

PROCEDURE FOR DYNAMIC IP ADDRESSING

The WS10 incorporates a DHCP client for automatic network parameters assignment (IP, gateway and subnet mask). This feature requires that:

- A DHCP Server is available in the same WS10 subnet. Check this condition with your network administrator.
- The DHCP client must be enabled in the WS10. The factory setting for this function is **disabled**.

To enable the DHCP client in the WS10, follow the steps below:

- Connect the Ethernet to the WS10
- Power-on the WS10
- Open a Telnet session with the WS10 using the current IP address (probably the factory setting address). The computer must be in the same WS10 subnet.
- Enable the DHCP client in the WS10, typing: `DHCP 1` and Enter.
- Type `REBOOT` followed by the Enter.
- After reboot, the WS10 will request a IP address to the network DHCP Server. This address is obtained from your network DHCP Server. Ask your network administrator.

The address assigned to the WS10 is retained even if it is powered off, as long as it comes back to the network before the expiration date configured in the DHCP Server, typically a few weeks. If the WS10 is kept away from the network beyond the IP expiration date, then it will probably get a new IP address when it returns to operation. To avoid this possibility of having a new IP address assigned to the WS10, the DHCP Server can be configured to keep a permanent IP address for the WS10, based on its MAC address. Consult your network administrator.

CONFIGURATION FILES

All WS10 configurations are based on text files which contain the parameters that control all of the WS10 functions.

CFG FILES

The configuration of the WS10 functionalities is defined in text files stored in the disk **A** root. The configuration files carry the extension `CFG`, and, in order to be modified, they must be transferred using FTP to a computer, edited with a text editor with no formatting (as Windows Notepad) and transferred back to the WS10. The new configurations will be effective in the next WS10 boot. These configuration files have a rigid syntax, and the rules for editing the files must be followed strictly to assure success in the configuration.

These files are separated by sections (names between `[]`), sub-sections (names between `< >`) and parameters, as the following example:

```
[ DEVICES ]           Section name: DEVICES
< TT101 >           Sub-section name: TT101
ADDRESS=1           Assigns 1 to the ADDRESS parameter.
```

When a Section has sub-sections, these must be placed in the end of the section, after all the parameters that are not related to sub-sections. No distinction is made between capital and low case characters. It is not recommended to use accentuated characters in the configuration files. When defining the tag names, make sure not to repeat names in the same device, as no warning will be generated and only the first defined tag will be accessed.

To define a comment line on a CFG configuration file, start the line with the `#` character.

It is recommended to keep in a computer a complete copy of all the WS10 files. Always edit the configuration files and HTML pages in this same computer. This way a backup copy with the full WS10 contents is saved, protecting the application against accidental file corruption.

Observe the correct spelling for the file names. If a file with the wrong name is included in the WS10, the corresponding function will not be installed.

Parameter names inside all CFG files must be written exactly as specified in this manual.

The WS10 configuration files are:

MODBUS.CFG	Data acquisition configuration for the ModbusRTU network (WS10 as the network master)
LOCALIO.CFG	Configuration and calibration of local inputs
WEBS.CFG	Configuration of the HTML page server
PPP.CFG	Modem configuration for originating data calls (PPP client)
MODBUS2.CFG	Configuration of the WS10 as a slave in a ModbusRTU network
CLITCP.CFG	Configuration for the periodic send of data to a Server.
DATALOG.CFG	Configuration of the WS10 for the historical logging in internal Flash memory
ALARMS.CFG	Configuration of the alarm condition supervision or exceptions
MAIL.CFG	WS10 configuration for e-mail transmission
MODBTCP.CFG	ModbusTCP configuration
MAP.CFG	Registers memory map definition for ModbusTCP or ModbusRTU-Slave access
DNS.CFG	Addresses for DNS servers.

CHIP.INI FILE

The CHIP.INI file belongs to the WS10 operating system and shall not be altered, except in particular situations, listed below. Tampering on this file may result on WS10 failure.

Only in the cases listed below should the CHIP.INI file be modified:

- Change of the security level and passwords for the Telnet console access.
- Change in the security level and passwords of the FTP server.
- Configuration of the WS10 for receiving a data phone call (PPP Server).

Further details can be obtained in the sections that deal with security and modem configuration for data transfer.

LOCAL INPUTS AND OUTPUTS CONFIGURATIONS

The built-in inputs and outputs are configured on the **LOCALIO.CFG** file. Only the selection between voltage and current input is accomplished in hardware. All other configurations are done on this file, exemplified below.

```
[General]
AChannels=1
DecimalSeparator=.
[Input1]
Tag=LIN1
Logic=1
PullUp=2
Debounce=1
Callow=0
CalHigh=1023
[Input2]
Tag=LIN2
Logic=1
PullUp=1
Debounce=0
Callow=0
CalHigh=1023
<Flow>
TotTag=TOT2
Interval=10 [Input3]
...
[Input4]
...
[Output1]
Tag=OUT1
[Output2]
Tag=OUT2
[Memory]
MEM01=0
MEM02="NOVUS"
[Memory-Hold]
MEM03=50
[Aliases]
PRES1=_INTERNAL_/LIN1
LEVEL=TT101/PV,0,0,1000,3000,1
```

On **[General]** section are defined the number of analog input channels (the remaining channels are digital inputs) and the character to be used as decimal separator in numbers calculated by the WS10. In the example, one input is defined as analog, and the decimal separator is a dot.

All parameters for each of the 4 inputs are defined on sections **[Input1]**, **[Input2]**, **[Input3]** and **[Input4]**. On the end of each of this sections can be defined a sub-section named **<Flow>**, that activates flow measurement for this input. The instantaneous flow is defined as the number of pulses on the defined **Interval**. The total flow is also calculated and identified by a separate tag named on the parameter **TotTag**.

On sections **[Output1]** and **[Output2]** tag names are assigned to both outputs.

On **[Memory]** section the user can define tag names for memory registers. This memory registers are not associated to any input or output, and may be used to store values used as email triggers, historical log control and any other function that require volatile storage of values. A name and initial value is defined for each memory register. Changes to the content of a memory register will be lost when the WS10 is powered-down. On power-up the defined initial value will be assigned to each memory register. To assign text to a memory register, set the initial value as a quoted string (maximum 200 characters).

On section **[Memory-Hold]** the user can define tag names for non-volatile memory registers. The initial value of these registers can be changed by a write operation to the register using an HTML page, for example. They can also hold text values when defined with a quoted string as initial value (maximum 200 characters). These register can be used, for example, to hold setpoints for the alarms of the WS10.

On the **[Aliases]** section, new registers can be defined from the value of an existing register. It can be used to create a new register, calculated as a linear transformation of an existing register. It is useful to convert the numerical range of a register. In the example, register LEVEL (in the range 0 to 300.0) is calculated from register TT101/PV, which range is 0 to 1000.

The input, output and memory registers belong to the **_INTERNAL_** device.

Local inputs are automatically read by the WS10 at 100ms interval. If the configuration file **MODBUS.CFG** exists, the local inputs scan will be done on the same **Scanrate** of the Modbus network.

Section	Parameter	Values	Description
General	AChannels	0, 1, 2 or 4	Number os analog channels. 0: All 4 channels are digital inputs 1: Input 1 is analog 2: Input 1 and 2 are analog 4: All 4 channels are analog inputs
Input1	Tag	Up to 6 characters or digits	Name for the input channel
	Logic	0 or 1	For digital inputs, define the active state of the input. Has no effect on analog channels. 0: Input state is active (1) when the input is in low level. 1: Input state is inactive (0) when the input is in low level.
	PullUp	0, 1 or 2	Define the type of internal bias resistor (10K) for the input. Select 2 for analog inputs. 0: Bias resistor tied to ground (PullDown) 1: Bias resistor tied to +5V (PullUp) 2: No bias resistor. Use this setting for analog channels
	Debounce	0 or 1	Enable (1) or disable (0) the digital input filter. Enable the debounce filter to debounce contact signals. This parameter has no effect on analog channels. Filter time is 200ms.
	CalLow	See section <i>Local Inputs Calibration</i>	Low value for calibration of the analog channel.
	CalHigh		High value for calibration of the analog channel.
	Flow Sub-section TotTag	Up to 6 characters or digits	Define a tag name for the flow total, if the flow function is used. Remove the Flow sub-section if pulse flow measurement is not necessary.
Flow Sub-section Interval	1 to 30000	Time in seconds for the flow measurement. The register identified by the input tag will contain the number of pulses counted on each Interval .	
Input2	The same parameters described for Input1 are repeated under the sections for the 3 other inputs.		
Input3			
Input4			
Output1	Tag	Up to 6 characters or digits	Output channel name.
Output2	The same parameters described for Output1 are repeated under the section for the second output		
Memory	Name for the register with up to 6 characters or digits	Initial value for the memory register. Numbers up to 32 bits and text up to 200 characters	Names and initial values for a list of volatile memory registers. To define a text register, set the initial value as a quoted text string.
Memory-Hold	Name for the register with up to 6 characters or digits	Initial value for the memory register. Numbers up to 32 bits and text up to 200 characters	Names and initial values for a list of non-volatile memory register. To define a text register, set the initial value as a quoted text string. The initial value can be changed by a write operation to the register.
Aliases	Name for the new register with up to 6 characters or digits	<p>$RegB, A_L, B_L, A_H, B_H, P$</p> <p>where:</p> <p>$RegB$: Name for the register with up to 6 characters</p> <p>A_L, A_H, B_L, B_H: numbers in the range -2147483647 to 2147483648</p> <p>P: 0 to 9</p>	<p>In section Aliases new registers can be defined. The value of the new register will be calculated using a linear transformation of the value of an existing register. In this section multiple new registers can be defined, one in each line of the section. Syntax:</p> <p style="text-align: center;">$RegB=RegA, A_L, B_L, A_H, B_H, P$</p> <p>$RegB$: Name for the new register, with up to 6 characters. $RegA$: Name of an existing register. A_L: Minimum Value of $RegA$. A_H: Maximum value of $RegA$. B_L: Minimum value for $RegB$. B_H: Maximum value for $RegB$. P: Number of decimal places in the calculated $RegB$ value. $(B_H-B_L), (A_H-A_L)$ must be in the range -2147483648 to +2147483647 $RegB$ value will be continuously calculated using the following formula:</p> $RegB = \left(\frac{B_H - B_L}{A_H - A_L} \right) \cdot (RegA - A_L) + B_L$

LOCAL INPUTS CALIBRATION

Before using a local input as analog input, it must be calibrated. The local inputs have a 10 bits resolution and a measurement range from 0 to 5V or 0 to 20mA. The calibration procedure allows configuration of the measured value range, to achieve the desired result in engineering units.

Calibration is configured in 2 parameters for each channel on the **LOCALIO.CFG** file: **CalLow** and **CalHigh**. During calibration the values of these parameters need to be changed using the described FTP procedures. Remember to reboot the WS10 after the transfer of a new configuration file. The measurement result of each channel can be viewed using the browser to display the default HTML file installed on the WS10.

The goal of the calibration procedure is to get the minimum and maximum desired values when the minimum and maximum signals are applied to each input. On the following procedure, this indication limits are identified as **ResultLow** and **ResultHigh**.

The calibration procedure should be executed to each analog input, following the next steps.

- Configure the J1 to J5 jumpers according to the type of signal – V or mA.
- Edit the **LOCALIO.CFG** file to set the initial values for the calibration parameters: **CalLow** = 0 and **CalHigh** = 1023. Transfer the file to the WS10 and reboot.
- Apply the lowest signal value to the input (for example, 0V for a 0-5V signal or 4mA for a 4-20mA signal).
- Check the measurement of the input – This value is the **ADLow**.
- Apply the highest signal value to the input (for example, 5V for a 0-5V signal or 20mA for a 4-20mA signal).
- Check the measurement of the input – This value is the **ADHigh**
- Calculate the first calibration parameter of the channel – **CalLow**:

$$CalLow = \frac{(ADLow \times ResultHigh) - (ADHigh \times ResultLow)}{(ResultHigh - ResultLow)}$$
- Calculate the second calibration parameter of the channel – **CalHigh**:

$$CalHigh = \frac{CalLow \times (ResultHigh - 1023) + 1023 \times ADHigh}{ResultHigh}$$
- Edit the calibration parameters **CalLow** and **CalHigh** on the calibrated channel section of the **LOCALIO.CFG**. Transfer the file to the WS10 and reboot.
- Check the calibration result on the extreme input values and at least one intermediate value.
- Repeat the above procedure for each analog input.

If **CalLow** or **CalHigh** values result out of the –32768 to +32767 range, the desired calibration is not achievable. Try different values for **ResultLow** and **ResultHigh**.

PREDEFINED TAGS OF THE WS10

In addition to the user defined tags, the WS10 has a set of predefined tags, listed on the following table.

Internal Tag name	Description
_VER	Integer value containing the firmware version of the WS10. Read only.
_BOOT	WS10 reboot. Write 1 to this tag to force the reboot. Write only.
_DAY	Day value of the internal real time clock. Read/Write.
_MONTH	Month value of the internal real time clock. Read/Write.
_YEAR	Year value of the internal real time clock. Read/Write.
_HOUR	Hour value of the internal real time clock. Read/Write.
_MIN	Minutes value of the internal real time clock. Read/Write.
_SEC	Seconds value of the internal real time clock. Read/Write.
_LOG	Identify the content of the data logger function. Used on the configuration of the email or of the data delivery functions, to enable transmission of the data logger memory content. Read only
_LOG_E	Identify the content of the data logger function. Used on the configuration of the email or of the data delivery functions, to enable transmission of the data logger memory content, followed by an automatic log erase. Read only

REFERENCE TO THE _INTERNAL_ DEVICE TAGS

The **_INTERNAL_** device contains all internal tags of the WS10:

- Predefined tags
- Tags assigned to the local inputs and outputs
- Tags assigned to memory registers

To reference internal tags on the creation of HTML files and other configurations, use the syntax exemplified below:

INTERNAL/_day

INTERNAL/MyOUT

MODBUS RTU NETWORK CONFIGURATION – WS10 AS MASTER

The WS10 can be configured as a Modbus master, and can read or write on memory locations of other Modbus devices on the same network. If the **MODBUS.CFG** file exists on the WS10, this function is enabled and the WS10 starts scanning the network according to the configuration of devices and registers. Values read from the external devices are stored on the WS10 memory and are identified by a device/tag name. Follows an example of content of the **MODBUS.CFG** file:

<pre>[Config] Scanrate=50 Serial=1 Flowctrl=2 De485=1 Baudrate=19200 Parity=0 Wordlen=8 Stopbits=1 Respondedelay=100 Bytetimeout=80 FrameDelay=10 Retries=2 [Devices] <TT101> Address=1 PV=5 OUT=24,1 <TT201> Address=2 TVALS=0,3,8 INVALS=32,4,16</pre>	<p>On section [Config] the communication parameters and data acquisition interval are defined.</p> <p>On section [Devices] the external Modbus devices are identified. A new sub-section is defined and named for each Modbus device. The Address parameter defines the slave address and a list of tags and register addresses define the registers of interest on the device.</p> <p>In this example there are two Modbus devices: TT101 (slave address 1) and TT201 (slave address 2).</p> <p>On the TT101 device, two registers are defined. The register identified as PV is a Modbus holding register located at address 6 of the device. The register identified as OUT is a Modbus coil status register located at address 24 of the device, and is read using Modbus command 1 (<i>Read Coil Status</i>).</p> <p>On the TT201 device, two block registers are defined. Register TVALS are read from the Modbus slave using command 3 (<i>Read Holding Register</i>), and 8 registers starting at address 0 are read. Registers INVALS are read using command 4 (<i>Read Input Register</i>), and 16 registers are read starting at address 32.</p> <p>If the Modbus command is not specified, it defaults to the <i>Read Holding Register (3)</i> for read operation and <i>Preset Single Register</i> for write operation. If the number of registers is not specified, a single Modbus register is read.</p> <p>Valid Modbus commands on a tag definition are: 1, 2, 3 or 4. If a write operation is issued on a tag defined with commands 1 or 3, the WS10 executes the write operation using command 5 or 6. The WS10 can't issue block write commands to the Modbus network.</p>
--	---

Section	Sub-Section	Parameter	Values	Description
Config		Scanrate	1 to 30000	Data acquisition interval, in 100ms units. The same interval will be used for acquisition from local inputs.
		Serial	1 or 2	Number of the serial port assigned to the Modbus RTU function. Factory setting is 1.
		Flowctrl	0 or 2	Serial port flow control configuration: 0: Now flow control 2: RTS/CTS flow control (typical when RS485 or RS232 with external RS232/485 converter. Factory setting)
		De485	0, 1 or 2	RS485 mode configuration. 0: RS232 with external RS232/485 converter. 1: RS485(factory setting) 2: RS232
		Baudrate	300, 600, 1200, 2400, 4800, 9600, 19200, 38400	Serial port baud rate. Factory setting is 9600.
		Parity	0, 1 or 2	Serial port parity. 0: No parity (factory setting); 1: Even; 2: Odd.
		Wordlen	7 or 8	Number of bits per byte. Factory setting is 8.
		Stopbits	1 or 2	Number of stop bits. Factory setting is 1.
		Respondedelay	1 to 30000	Maximum response wait time, in ms. Factory setting is 100.
		Bytetimeout	1 to 30000	Maximum interval between bytes of the received frame, in ms. Factory setting is 80.
		FrameDelay	0 a 30000	Time in ms between the reception of an answer frame and transmission of a new request frame. Typical value is 10.
		Retries	0 to 100	Number of communication retries in case of error. Factory setting is 2.
Devices	Device name with up to 10 characters (1 sub-section for each device)	Address	1 to 247	Address of the slave device named on this section.
		Tag name with up to 6 characters or digits	<i>Addr, Cmd, Num</i> <i>Addr: 0 to 9999</i> <i>Cmd: 1 to 4</i> <i>Num: 1 to 125</i>	Tag name, first Modbus address (<i>Addr</i>), Modbus read command (<i>Cmd</i>) and number of registers for each variable of interest on the device of this sub-section. Specifying the Modbus command (<i>Cmd</i>) is optional, and defaults to command 3 (<i>Read Holding Register</i>). Valid commands are 1 to 4. Specifying the number of registers (<i>Num</i>) is optional, and defaults to a single register. When the number of registers is specified, the Modbus command (<i>Cmd</i>) must be specified. Registers with <i>Num</i> greater than 1 are called Block Registers. Multiple Modbus devices can be defined, one in each sub-section. Multiple registers can be defined for each Modbus device.

REFERENCE TO THE MODBUS ACQUIRED TAGS

Reference to the WS10 registers associated to external Modbus RTU devices are done through the device and tag name defined at the **MODBUS.CFG** file, using one of the syntax options below. Only registers read with commands 3 (default command) and 1 accepts write operation.

Reference to the value of a single register	DeviceName/TagName TT101/PV TT101/PV.Val Same result as above. ".Val" means the tag value and can be omitted.
Reference to a member of a Block Register	DeviceName/TagName[n] TT201/TVALS[0] 1 st position of the block register TT201/INVALS[5] 6 th position of the block register
Reference to the status of the last read operation on a Modbus register	DeviceName/TagName.Sts TT101/PV.Err Register error status TT201/TVALS[0].Err Block Register error status. The status is the same for all members of the block
Reference to the error counter of the read operations on a Modbus register	DeviceName/TagName.Sts TT101/PV.ErrCnt Number of reading errors of the register TT201/TVALS[0].ErrCnt Number of reading errors of the block. The number is the same for all members of the block

The values read by the WS10 from the external Modbus slaves are stored on the WS10 memory with additional status information to allow validation of the last Modbus network operation. An communication error counter is also stored for each register.

The communication status of the last reading (.Err) is read-only and can contain the following values:

- 0: The current value was read with no error on the last scan interval.
- 1: The current value is out-dated due to a Invalid Operation error on the last scan interval.
- 2: The current value is out-dated due to a CRC error on the last scan interval.
- 3: The current value is out-dated due to a time-out error on the last scan interval.

The error counter (.ErrCnt) is a read-write value that contains the accumulated number of communication errors detected for the associated register or block register.

The error and error counter values are not updated on write operations to the Modbus network. The WS10 doesn't offer any function to validate the success of a write operation into a Modbus slave.

ALARMS SUPERVISION CONFIGURATION

The WS10 can compare the value of any register against limits defined as constants or other registers. The result of the comparison is assigned to a new alarm register. Based on the result of the alarm, registers can be written to, allowing the WS10 to take actions based on the detected alarm condition. Some examples of actions that can be executed by the WS10 when an alarm condition is detected:

- Write to a local output register, allowing local signaling of the alarm condition.
- Write to registers of external Modbus devices, taking alarm actions on other parts of the system.
- Enable the WS10 data logger function to start logging critical values of the system under abnormal conditions.
- Trigger email messages containing process information and informing the start or end of an alarm condition.

The alarm function is enabled when the **ALARMS.CFG** file is installed on the WS10. Follows an example of the **ALARMS.CFG** file:

<pre>[Alarm1] Condition=TT101/PV > 200 Hysteresis=2 <SetList> _INTERNAL_/OUT1=1 TT101/RUN=0 <ResetList> _INTERNAL_/OUT1=0 [Emerg1] Condition=TT201/TVALS[3]<=_INTERNAL_/MEM01 Hysteresis=1 <SetList> _INTERNAL_/OUT2=1 <ResetList> _INTERNAL_/OUT2=0</pre>	<p>Each section defines a new alarm, and the name of the section defines a tag name on the <code>_ALARMS_</code> device. This read-only tag contains alarm status.</p> <p>Under each alarm section, an alarm condition is defined from the comparison of a register value and a constant value or from the comparison of 2 registers values. Many comparison operators can be used.</p> <p>At the <code><SetList></code> sub-section of each alarm a list of write operations can be defined. These write operations will be executed when the alarm condition starts.</p> <p>At the <code><ResetList></code> sub-section of each alarm a list of write operations can be defined. These write operations will be executed when the alarm condition ends.</p> <p>Multiple alarm sections can be defined.</p>
--	---

Section	Sub-section	Parameter	Values	Description
Alarm Name With up to 6 characters or digits		Condition	Device/Tag Operator Device/Tag or Device/Tag Operator Numerical constant	Defines the alarm activation condition. The alarm is activated when the condition results true. The comparison may be between 2 tag values or one tag and one constant value. The allowed comparison operators are: >, <, >=, <=, = e <>. In a typical application, an local input register is compared against a memory register. The value of this memory register is programmed by the operator through an HTML page on the browser screen.
		Hysteresis	0 to 30000	Hysteresis for the end of the alarm condition. Example. If the alarm is activated above 200 and the hysteresis is 2, then the alarm will be deactivated below 198.
	SetList	Device/Tag	Value for the register	List of register identifications and values. The values will be written to the registers when the alarm condition begins.
	ResetList	Device/Tag	Value for the register	List of register identifications and values. The values will be written to the registers when the alarm condition ends.
Multiple alarms can be defined, one in each section as the one described above.				

When the **ALARMS.CFG** file is installed, a new device named `_ALARMS_` is created on the WS10. This device contains tags with the same names of the alarm sections on the file. The values of these alarm registers correspond to the status of each alarm, with value zero for an inactive alarm and one for an active alarm condition. Typically these alarm registers are used to trigger email messages and displayed on the browser screen.

REFERENCE TO THE WS10 ALARM TAGS

To reference the alarm status tags under the `_ALARMS_` device, use the syntax of the example below:

`_ALARMS_/HiPres`

Alarm registers are read only. A value of **0** indicates an inactive alarm, and a value of **1** indicates an active alarm condition.

DATA LOGGER CONFIGURATION

The WS10 can be programmed to periodically save a set of register values and time stamps on its flash memory as a file. This file can be transferred to the PC using FTP or can be automatically sent as an email attachment.

Attention: The data logger is stopped during the transmission of the data.

The data logger function is enabled when the **DATALOG.CFG** is installed on the WS10. Follows an example of the **DATALOG.CFG** file:

<pre>[General] ID=LOGGER1 File=A:\LOG\DATALOG.TXT Interval=600 ReserveDisk=10000 WrapAround=0 FileMode=TXT DateFormat=DMY LogEnable=_INTERNAL_/IN3 LogClear=_INTERNAL_/MEM1 [DatalogServer] Enable=0 Port=2001 [VarList] N1100/PV N1100/PV.Err=PVStatus _ALARMS_/AL01</pre>	<p>At the [General] section all log control parameters are defined.</p> <p>At the [VarList] section all registers that will be logged are identified. An alternative name (alias) can be defined for each register. The register names or alias are printed on the first line of the log file. The following lines contains the time-stamps and values of each register.</p> <p>Section [DatalogServer] is necessary only when a Java application supplied by Novus is used to plot the content of the data logger into an HTML page. If this function is not being used, omit this section.</p>
---	---

Section	Parameter	Values	Description
General	ID	Log identification, with up to 10 characters	Name for the Log, to allow the recipient of the log file to identify its origin. The ID is printed on the first line of the log file. Maximum 10 characters.
	File	Name and path to the log file	Name of the log file. Must start with the drive letter (A: for internal flash, B: for external flash or compact flash card), a folder name and a file name. To allow browser access to the log file, store at the A:\WEB folder. If browser access to the log file is not necessary, store on a separate folder.
	Interval	1 to 65535	Interval between logs in 100ms units. At each interval a new line is appended to the log file, containing a time stamp and values of the listed registers. Avoid setting interval to values below 100 (10 second), to avoid premature failure of the Flash memory and the WS10 to fail.
	WrapAround	0 or 1	Action to be executed when no more memory is available for the log file. If 0, the data logger is stopped. If 1 the log continues on the beginning of the file, replacing the older lines with the newer lines.
	ReserveDisk	5000 to 2000000	Number of bytes to be reserved on the flash memory. The data logger will not use the specified amount of memory for storage of the log file.
	FileMode	TXT	Type of the log file. At the present WS10 version only the text format TXT is available.
	DateFormat	DMY, MDY or YMD	Date format of the time-stamp logged on each line of the log file. Define the sequence of day (D), month (M) and year (Y). The separator is /.
	LogEnable	Device/Tag	Register that enable or disable the data logger. If the register value is 0, the data logger is suspended. If the register value is different from 0, the data logger is enabled. To keep the data logger always running, omit this parameter line.
DatalogServer	LogClear	Device/Tag	Register that clear the data logger file. When the register value changes from 0 to a non 0 value, the log file is erased. After the erase completion, the WS10 write 0 to this register. To disable the erase of the log file under register control, omit this parameter line. If the register value is 0, the data logger is suspended. If the register value is different from 0, the data logger is enabled. To keep the data logger always running, omit this parameter line.
	Enable	0 or 1	Enable TCP communication with an external application for reading the data logger file content.
DatalogServer	Port	1 to 65535	TCP port used to communicate with an external application for reading the data logger file content. Value above 1024 should be used.
	Section DatalogServer should only be created if the Java application supplied by Novus is being used to display a plot of the data logger content into an HTML page. If this function is not being used, omit this section.		
VarList	Device/Tag of the registers to be logged	Alternative name for the register, with up to 24 characters.	List of registers to be included on the log file. The device and tag names or the alternative name will be printed at the first line of the file. The registers values will be printed on every line of the log file.

DATA LOGGER FILE FORMAT

A new line is appended to the log file at each interval. The first line contains the identifiers, and all following lines contain the values of the listed registers.

The first 2 columns of a log line contain the date and time, and the following columns contain the values for each register. Columns are separated by the comma character.

File layout:

	Log ID	Empty field	Identifier 1	Identifier 2	...	Identifier n
If Interval ≥ 10 (1s)	MM/DD/YY	HH:MM:SS	Value 1	Value 2	...	Value n
If Interval < 10 (1s)	MM/DD/YY	HH:MM:SS.S	Value 1	Value 2	...	Value n

File example:

```
OvenRoom, ,Oven1Temp,Oven2Temp,AmbientTemp,TT101/PV,
12/24/04,10:10:10,180,200,28,121,
12/25/04,11:11:11,182,245,28,121,
12/26/04,22:22:22,182,286,30,120,
```

When data wrap around is enabled (*WrapAround=1*), one or more of the file lines may contain the character ^, indicating that this line has no data and can be discarded. These lines are automatically excluded when the log file is sent as an email attachment.

PLOTTING THE DATA LOGGER CONTENT INTO AN HTML PAGE

Novus offers a Java application that can read the content of the data logger file from the WS10 flash memory and plot its data in an HTML page, with zoom and print capability. The Java application must be installed in all PCs that will display this plot. These PCs must also have installed the Java Virtual Machine from Sun – JRE.

To enable communication between the Java application and the WS10, the following parameters must be programmed in section [DatalogServer] of the DATALOG.CFG file:

```
[DatalogServer]
Enable=1
Port=2001
```

The TCP port 2001 is the default for this function. The same port must be programmed for the WS10 and in the Java application configuration.

The Java application, documentation and examples are included in the WS10 CD.

REFERENCE TO THE LOG RELATED TAGS

To integrate the data logger to the email and periodical data transmission functions of the WS10, two tags are defined on the _INTERNAL_ device of the WS10 to identify the data logger content. If one of these tags is included on the list of tags for delivering, the content of the log file will be attached to the email message or sent to the specified server. The two log related tags are:

```
_INTERNAL_ / _LOG           Send the log content and don't erase the log file
_INTERNAL_ / _LOG_E       Send the log content and erase the log file
```

During the log delivery, the data logger is suspended. Use the _LOG_E tag with caution, since there is no guarantee of the email delivery and data can be lost.

To get access to the log file through the browser, add a FTP link to the log file on the HTML page, or address directly the file path at the browser address line (the log file must be located at the WEB folder to allow http access to it).

HTML PAGE SERVER CONFIGURATION

The WS10 can store and serve HTML files. All files accessed through http must be stored in the *WEB* folder or in its sub-folders. Use FTP to include and exclude files. It is not necessary to reboot the WS10 when new or updated files are transferred to the WS10. The new page version will be displayed on the next browser refresh.

The default page of the WS10 is named *MAIN.HTM*. If a page with this name is stored in the WS10, it will be exhibited on the browser when only the WS10 IP address is typed at the address line. All web server files must be located at the *WEB* folder or in its subfolders.

The *WEBS.CFG* configuration file define some parameters to the web server of the WS10. Follows an example:

<pre>[Auth] PostPage=procform User=Novus Pass=suvoN [PagesList] forno1.htm forno2.htm</pre>	<p>At section [Auth] are defined the user name and password to log on all HTML files served by the WS10. The CGI script name is also defined at this section.</p> <p>At section [PagesList] all HTML files containing dynamic data must be listed. Dynamic pages are HTML files that contain tag names of WS10 registers, which will be replaced by the corresponding values when the page is served by the WS10. Them <i>MAIN.HTM</i> page can not contain dynamic data.</p>
---	---

Section	Parameter	Values	Description
<i>Auth</i>	<i>PostPage</i>	CGI script name	Name of the CGI script executed by the WS10 when the method POST is invoked on an HTML formulary. Must match to the name assigned to the ACTION attribute of the formulary.
	<i>User</i>	User name	Name of the user that can access the pages served by the WS10. If this parameter is defined, the browser will prompt for a user name and password before presenting the HTML files. Omit this parameter to disable password on each page served by the WS10.
	<i>Pass</i>	User password	Password assigned to the user. Omit this parameter to disable passwords.
<i>PagesList</i>	<i>HTML file names</i>		List of dynamic HTML files. Dynamic pages are HTML files that contain tag names of WS10 registers, which will be replaced by the corresponding values when the page is served by the WS10. Attention: Type the name of the files exactly the same way here and at the browser address line. The web server differentiates lower case and upper case characters, and a mistyped file name will display with the tag names instead of the tag values.

CREATING HTML FILES

There are no special restrictions on the design of pages hosted by the WS10. All the usual techniques such as frames, tables, Java script, images, animation, etc. can be used, limited only by the available memory.

Pages that should present dynamic information must include a special marker at the dynamic data position. This marker identifies the device and tag name of the value to be displayed, and should be enclosed between 2 percent characters, as exemplified below:

%DeviceName/TagName%

Attention:

Only pages listed at the [PagesList] section of the WEBS.CFG file will be served by the WS10 with the tag names substituted by the tag values. When invoking these pages from the browser address line, the same name and capitalization defined at the WEBS.CFG file must be used.

The WS10 can not handle references to internal registers in the main page – MAIN.HTM. Register names will not be replaced with the respective values for this page.

To include the percent (%) character at the page, type it doubled – %%.

The maximum size for HTML and associated files (images, flash, etc) is 64K for each file.

If a register name is unknown, it will be substituted by the string "N/A" when the page is served.

The page at Example 1 presents the content of 3 registers in a table. The register identifications are highlighted. When serving this page, the WS10 will replace the tag names with the corresponding values stored on its memory.

```
<html>
<head>
<title>Monitor</title>
  <META HTTP-EQUIV="Refresh" CONTENT="10">
</head>
<body>
<font face="Arial" color="navy" size=+1><p align="center"><b>

<table border=1>
<tr>
      <td colspan=2 align=center> Heat Exchanger 1 </td>
</tr><tr>
      <td> Input </td><td> %TT101/PV% </td>
</tr><tr>
      <td> Output </td><td> %TT201/INVALS[0]% </td>
</tr><tr>
      <td> Flow </td><td> %_internal_/FT101% </td>
</tr>
</table>

</b></p></font>
</body>
</html>
```

Example 1 – Dynamic HTML file with table

The *MAIN.HTM* page can not display register values. To display dynamic data in the start page, define frames inside *MAIN.HTM* and include register references in separate HTML files. The *MAIN.HTM* page can also switch automatically to the page which actually displays the register values. See example of redirection to *DATA.HTM* below:

```
<html>
<head>
  <meta http-equiv="REFRESH" content="0;URL=data.htm">
</head>
</html>
```

The example page 2 uses Java Script to format date and time information from the WS10 real time clock, and to execute mathematical operations before presenting the data. This is an important feature, since sometimes the data measured or acquired by the WS10 is not in the desired engineering unit for exhibition. The usage of Java scripts allows formatting of the data before presenting the page. In this example a link to other page is also created.

```
<html>
<head>
<title>Monitor</title>
  <META HTTP-EQUIV="Refresh" CONTENT="3">
</head>
<body>
<center><font face="Arial" color="navy" size=+1><b>
<script language="JavaScript" type="text/JavaScript">
<!--
var month = new Array('jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug', 'sep', 'oct', 'nov', 'dec');

function cMonth(x)
{
  if (x > 0 && x < 13)
    return month[x - 1];
  else
    return '';
}

function cNum(x)
{
  if (x >= 0 && x < 10)
    return '0' + x;
  else
    return '' + x;
}
document.write(cMonth(%_INTERNAL__/_MONTH%)+', %_INTERNAL__/_DAY%, '+ cNum(%_INTERNAL__/_YEAR%)+ '<br>');
document.write(cNum(%_INTERNAL__/_HOUR%)+':'+cNum(%_INTERNAL__/_MIN%)+':'+cNum(%_INTERNAL__/_SEC%));
document.write('<br><br>FT1: '+%_INTERNAL__/_FT1%/9+'<br>');
document.write('FT1 Total: '+Math.round(%_INTERNAL__/_FT1TOT%/10)+'<br>');
/-->
</script>
</b></font>
<br>
<br>
<a href=modif.htm>Modify</a></center>
</body>
</html>
```

Example 2 – Usage of Java Script for formatting and math operations on tag values

USAGE OF CGI TO SEND VALUES TO THE WS10

An HTML formulary can write to a WS10 register using the POST method. The WS10 built in CGI script associate the formulary values to the WS10 registers. The example 3 show the use of the POST method to write on the WS10 real time clock registers and on 3 other registers.

The CGI script name defined at the **[Auth]** section of the **WEBS.CFG** file must match to the name used at the page Forms. The default name **procform** is used on the example and shown below:

```
<FORM ACTION=procform METHOD=POST>
```

The name of the page to be presented after the form post can be defined. This definition is optional, and the same form page will be reloaded if not defined. At example 3, the following hidden field is used to define the next page to be presented:

```
<INPUT TYPE=HIDDEN NAME=nextpage VALUE=data.htm>
```

It is important to examine the following field definition extracted from example 3:

```
<INPUT TYPE=TEXT NAME=_INTERNAL_/FT1TOT SIZE=7 VALUE=%_INTERNAL_/FT1TOT%>
```

In this field, the first usage of the tag `_INTERNAL_/FT1TOT` is not enclosed by the percent characters, since we don't want the WS10 to replace the name by the corresponding value. The tag name must remain untouched to inform to the WS10 CGI script the name of the destination register for the written data. The second usage of the same tag is enclosed by the percent characters to allow the WS10 to replace the tag name with the corresponding value, which will be used as the initial value for the form field.

```
Date/Time<br><br>
<script language="JavaScript" type="text/JavaScript">
<!--
function goback()
{
  document.location='data.htm';
  return false;
}

function cNum(x)
{
  if (x >= 0 && x < 10)
    return '0' + x;
  else
    return '' + x;
}

document.write('<FORM ACTION=procform METHOD=POST>\
<INPUT TYPE=HIDDEN NAME=nextpage VALUE=data.htm>\
<INPUT TYPE=TEXT NAME=_INTERNAL_/_MONTH SIZE=1 VALUE='+cNum(_INTERNAL_/_MONTH%)+ '>\/\
<INPUT TYPE=TEXT NAME=_INTERNAL_/_DAY SIZE=1 VALUE='+cNum(_INTERNAL_/_DAY%)+ '>\/\
<INPUT TYPE=TEXT NAME=_INTERNAL_/_YEAR SIZE=1 VALUE='+cNum(_INTERNAL_/_YEAR%)+ '><br>\
<INPUT TYPE=TEXT NAME=_INTERNAL_/_HOUR SIZE=1 VALUE='+cNum(_INTERNAL_/_HOUR%)+ '>:\
<INPUT TYPE=TEXT NAME=_INTERNAL_/_MIN SIZE=1 VALUE='+cNum(_INTERNAL_/_MIN%)+ '>:\
<INPUT TYPE=TEXT NAME=_INTERNAL_/_SEC SIZE=1 VALUE='+cNum(_INTERNAL_/_SEC%)+ '><br><br>\
<INPUT TYPE=SUBMIT NAME=button VALUE="Set Date/Time" ></FORM>');
//-->
</script>

<FORM ACTION=procform METHOD=POST>
  <INPUT TYPE=HIDDEN NAME=nextpage VALUE=data.htm>
  FT1 Totalization: <INPUT TYPE=TEXT NAME=_INTERNAL_/FT1TOT SIZE=7 VALUE=%_INTERNAL_/FT1TOT%>
  <INPUT TYPE=SUBMIT NAME=button VALUE="Apply">
</FORM>

<FORM ACTION=procform METHOD=POST>
  <INPUT TYPE=HIDDEN NAME=nextpage VALUE=data.htm>
  OUT1: <INPUT TYPE=TEXT NAME=_INTERNAL_/OUT1 SIZE=1 VALUE=%_INTERNAL_/OUT1%>
  <INPUT TYPE=SUBMIT NAME=button VALUE="Apply"><br>
</FORM>

<FORM ACTION=procform METHOD=POST>
  <INPUT TYPE=SUBMIT NAME=button VALUE="Back" onClick='return goback();'>
</FORM>
</b></center></font>

</body>
</html>
```

Example 3 – Write to the WS10 using formula with the POST method

The WS10 memory registers are useful to control some internal functions using HTML pages. Pages can write to memory registers used for:

- Enable, disable or clear the data logger file
- Trigger an email
- Define alarms setpoints
- Activate an alarm

Screenshots of the 3 examples are presented at the appendix A.

PERIODICAL DATA TRANSMISSION CONFIGURATION

The WS10 can be configured to periodically send data to a host computer. This host computer must run a dedicated application that accepts a socket connection requested by the WS10 and receive and store the sent data on files or database. The periodical data transmission function of the WS10 is enabled by the inclusion of the **CLITCP.CFG** file at the WS10.

There are 2 output paths for the TCP data: Ethernet or PPP (modem). The Ethernet interface is used when the host computer is on the same network or can be reached through the network gateway. PPP connection is used when the WS10 is installed on a remote location without available Ethernet connection. The WS10 can dial to a internet service provider and reach the host IP address through the internet, or can dial directly to the host.

Before using PPP and modem, read the section: "Configuring the modem to start a data call". To use a domain name instead of a IP address for the server, refer to section "DNS Configuration".

After the establishment of the socket connection between WS10 and host, the data transfer is started as bellow:

WS10		Host	Comment
ID WS10Name<LF>	→		WS10 presents itself to the server, using the name defined in CLITCP.CFG
	←	02<LF>	Server acknowledges. <LF> is the Line Feed character
{Data Block}	→		WS10 transfers data. Details bellow.
	←	03<LF>	Server acknowledges reception and storage of data.

The data block sent by the WS10 to the host computer is pure text with the following layout:

{S;N;IdData1;IdData2;...;IdDataN;ddmmyyhhmmss;ValData1;ValData2;...;ValDataN;}

or

{L;N;IdData1;IdData2;...;IdDataN;ddmmyyhhmmss;ValData1;ValData2;...;ValDataN;ddmmyyhhmmss;ValData1;ValData2;...;}

where:

- S or L** Data type identifier. **S** for a single value for each register
 L for the whole content of data logger file.
- N** Number of registers on the data block. In case of data logger file content, is the number of registers values stored on each log file line.
- IdData0...N** Identifier of the Nth register sent. Register name or alias defined at the **CLITCP.CFG** file.
- ddmmyyhhmmss** Day, month, year, hour, minute, second when the data was acquired by the WS10.
- ValData0...N** Value of the Nth register.
- { }** Block delimiter
- ;** Field separator

Example of the **CLITCP.CFG** file:

<pre>[General] Address=192.168.200.10 Port=2001 SendRate=60 KeepAlive=1 LinkType=ETH ID=Oven1 [VarList] N1100/PV TF201/INVALS[3] [TriggerList] _INTERNAL_/MEM5 _ALARMS_/AL22</pre>	<p>In [General] section the connection parameters are defined.</p> <p>In [VarList] section all registers that will be sent are identified. An alternative name (alias) can be defined for each register.</p> <p>In [TriggerList] section a list of tags can be included. When any of these tags change from a zero to a non-zero value, transmission of data is started.</p>
--	---

Section	Parameters	Values	Description
General	Address	Host IP address	IP address of the host computer. Must be accessible through the specified connection. If a DNS configuration exists, a domain name can be specified in substitution of the IP address.
	Port	0 to 65535	Number of the listening TCP port at the host computer.
	SendRate	0 to 65535	Interval in seconds of the periodical data transmission.
	KeepAlive	0 or 1	Define if the connection is maintained or interrupted after the conclusion of the communication. 0: After sending all data, the connection is aborted by the WS10. A new connection will be established for the next data transmission. This is the typical option for modem connection. 1: The connection to the host computer is established only once. This is the typical option for Ethernet connection.
	LinkType	ETH or PPP	Type of connection. ETH: Ethernet. PPP: PPP connection configured at the PPP.CFG file.
	ID	WS10 identification with up to 10 characters	Identification of the WS10, to allow the host computer to differentiate the source of data. Maximum 10 characters.
VarList	Device/Tag of the registers to be sent	Alternative name for the register, with up to 24 characters.	List of registers to be sent to the host computer. Both the name and value are sent. If an alternative name for the tag is defined, this alternative name is sent. To send the content of the data logger file, use <code>_INTERNAL_/_LOG</code> or <code>_INTERNAL_/_LOG_E</code> . When the content of the log file is selected, both register identifiers and data are sent from the content of the log file.
TriggerList		Device/Tag list	List of registers that can trigger the data transmission. When any of the listed registers change from zero to non-zero value, transmission is triggered. If the register is of the memory type, the WS10 write 0 to it after the message is sent.

Novus offers at no cost a Windows application to run at the host computer to accept connection and data from multiple WS10. Received data is saved on text files, one for each WS10. Source code in C language is included.

E-MAIL CONFIGURATION

The WS10 can send multiple e-mail messages, containing dynamic information. Messages can be sent periodically or by exception, based on register values. Both the subject and body of the message can contain dynamic data. The data logger file can be included on any message as an attachment. Some typical applications of the e-mail function:

- Periodically send system data, including data logger file.
- Send notification messages when alarm conditions are detected or terminated.
- Send SMS to mobile phones.

To send e-mails, the WS10 must have access to a SMTP (*Simple Message Transfer Protocol*) server. The SMTP server can be reached through the Ethernet or PPP (modem) interface. The Ethernet is used when the SMTP server is located on the network or at the internet accessed through the gateway. PPP connection is used when the WS10 is installed on remote locations where the network is not available. The WS10 can dial to a internet server provider and use its SMTP server to route the e-mail messages. Before using PPP and modem, read the section: "Configuring the modem to start a data call". To use a domain name instead of a IP address for the SMTP server, refer to section "DNS Configuration".

E-mails are enabled when the **MAIL.CFG** file is installed. Follows an example of the **MAIL.CFG** file.

<pre>[General] LinkType=PPP SMTPServer=192.168.200.2 SMTPPort=25 Login=novus Pass=suvon FromAddress=ws10@xyz.com Name=WebServer TimeZone=-3 [Message 1] <DestList> john@xyz.com mary@uvw.com <Subject> Furnace 3. Temperature = %TT101/PV% <MessageBody> Current furnace conditions: Temperature = %TT101/PV% Pressure = %_INTERNAL_/IN1% Alarm 2 = %_ALARMS_/ALARM2 . <Attachment> _INTERNAL_/_LOG <Scheduler> Interval = 240 TimeFilter = 30 <TriggerList> _ALARMS_/ALARM1 _INTERNAL_/MEM12</pre>	<p>At [General] section are defined the SMTP server connection parameters.</p> <p>Each following section define a new e-mail. The section name should be different for each message, and has no other usage.</p> <p>The sub-section <DestList> contains a list of addressees. One e-mail address can be defined at each line.</p> <p>The sub-section <Subject> contains a single line for the message subject. Dynamic data can be included by using register names enclosed on 2 percent characters.</p> <p>The sub-section <MessageBody> contains the body of the message, with multiple lines. Dynamic data can be included by using register names enclosed on 2 percent characters. The final line must contain a single period character (.) to define the end of the message body. The message can be defined using HTML.</p> <p>The sub-section <Attachment> allows the attachment of the data logger file to the message. Only tags _LOG and _LOG_E of the _INTERNAL_ device can be referenced here.</p> <p>At sub-section <Scheduler> an interval can be defined for the message periodic transmission. A time filter can be defined to avoid excess of messages.</p> <p>At the <TriggerList> sub-sections a list of tags can be included. When any of these tags change from a zero to a non-zero value, the message is sent.</p>
--	--

Section	Sub-Section	Parameter	Values	Description
General		LinkType	ETH or PPP	Type of connection. ETH : Ethernet. PPP : PPP connection configured at the <i>PPP.CFG</i> file.
		SMTPServer	IP address	IP address of the SMTP server. Must be accessible through the specified connection. If a DNS configuration exists, a domain name can be specified in substitution of the IP address.
		SMTPPort	SMTP Port	Port used to connect the SMTP Server. If this parameter is not set the Port 25 will be assumed as standard.
		Login	User name	Login name to the SMTP server. Omit this parameter line if the server doesn't require authentication.
		Pass	User password	Login password to the SMTP server. Omit this parameter line if the server doesn't require authentication.
		FromAddress	E-mail address	Return address for the e-mail message. Set to a valid e-mail address to allow reception of error messages on delivery failures.
		Name	Name	Name of the message sender.
		TimeZone	Time zone value	Time zone where the WS10 is installed. Examples: -3, +2:30.
Nome do e-mail	DestList		List of e-mail addresses	List of e-mail addresses to whom this message will be sent. One address per line.
	Subject		Line of text	Subject line of the message. May contain references to device/tag, enclosed by 2 % characters.
	MessageBody		Lines of text or HTML	Multiple lines of message body. May contain references to device/tag, enclosed by 2 % characters. If HTML code is detected, the email is sent in HTML format. The last line must contain a single period character to define the end of message body.
	Attachment		Device/Tag	If the message should contain the content of the data logger file, include this sub-section and one of the references to the log registers below: _INTERNAL / _LOG to send and don't erase the log file _INTERNAL / _LOG_E to send and erase the log file Usage of LOG_E is unsafe.
	Scheduler	Interval	0 to 65535	Interval in minutes of the periodic transmission of this message. Program 0 to disable periodic transmission.
		TimeFilter	0 to 65535	Minimum interval in minutes between consecutive transmissions of this message when triggered by registers. Used to limit the frequency of exception email messages.
	TriggerList		Device/Tag list	List of registers that can trigger the transmission of this message. When any of the listed registers change from zero to non-zero value, the message is triggered. If the register is of the memory type, the WS10 write 0 to it after the message is sent.
Multiple e-mail messages can be defined, one at each section as the one described above.				

Depending on the type of connection to the SMTP server, a long time should be required to transfer the message. This is more critical with PPP (modem) connection, where connection negotiation may take one minute or more. The message is constructed by the WS10 after the connection to the SMTP server is established, and all dynamic information will be acquired at this moment. Sometimes the condition that triggered the email will be finished when the message is finally posted, and this possibility should be considered on the design of the supervisory system.

MODBUS RTU NETWORK CONFIGURATION – WS10 AS SLAVE

The WS10 can be configured to act as a Modbus RTU slave on a serial port. When the WS10 is a Modbus RTU master on one serial port and slave on the other serial port, it can be used as a Modbus RTU concentrator.

Assignment of unique addresses to the WS10 registers are defined by the user in the **MAP.CFG** file, as described in section “Registers Address Map for Modbus TCP and Modbus RTU-Slave”.

Read requests are promptly answered by the WS10 from the data in its memory. When the WS10 receives a write command to a memory register that is associated to an external Modbus device (on the network that the WS10 is the master), it executes the write operation on this device. The WS10 will only confirm the write operation to the master when the confirmation from the slave on the other network is received.

The slave Modbus function is enabled when the file **MODBUS2.CFG** is installed. Follows an example of this file:

<pre>[Config] Serial=2 Flowctrl=2 De485=0 Baudrate=19200 Parity=0 Wordlen=8 Stopbits=1 Address=1 Timeout=20</pre>	<p>At the [Config] section the serial port parameters are defined. The <i>Address</i> parameter defines the slave address of the WS10 on the network.</p>
---	--

Section	Parameter	Values	Description
Config	Serial	1 or 2	Number of the serial port assigned to the slave Modbus RTU function.
	Flowctrl	0 or 2	Serial port flow control configuration: 0: Now flow control 2: RTS/CTS flow control (typical when RS485 or RS232 with external RS232 / 485 converter).
	De485	0, 1 or 2	RS485 mode configuration. 0: RS232 with external RS232 / 485 converter. 1: RS485 2: RS232
	Baudrate	300, 600, 1200, 2400, 4800, 9600, 19200, 38400	Serial port baud rate.
	Parity	0, 1 or 2	Serial port parity. 0: No parity; 1: Even; 2: Odd.
	Wordlen	7 or 8	Number of bits per byte.
	Stopbits	1 or 2	Number of stop bits.
	Address	1 to 247	WS10 slave address.
	Timeout	1 to 30000	Silent interval between Modbus frames, in ms. When a frame is received, it is considered complete after a silent interval with at least this value. Typical value: 20. Consider adjusting this value if WS10 is not responding a significant number of requests from master.

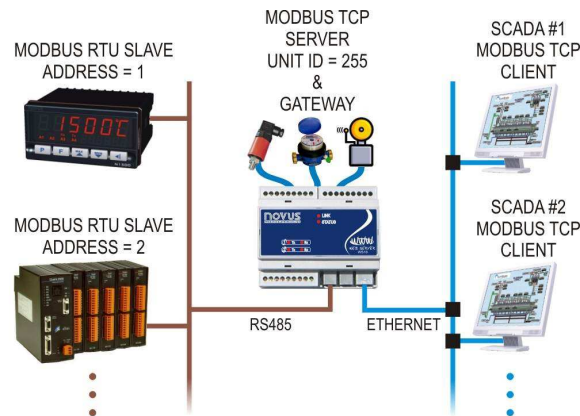
MODBUS TCP NETWORK CONFIGURATION – WS10 AS SERVER OR GATEWAY

Configuration of this function requires knowledge on ModbusRTU and ModbusTCP protocols. Full protocol documentation can be obtained at www.modbus.org.

Before including WS10 in a SCADA system, check if SCADA supports ModbusTCP.

OVERVIEW

WS10 can be configured to operate on a ModbusTCP network as a **Server** or **gateway**. A **Server** accepts connection requests issued by a client, and answers to its messages. A **gateway** accepts connection requests issued by a client, and forwards all messages between the client and a ModbusRTU sub-network. SCADA software are examples of ModbusTCP clients. The following figure shows a typical ModbusTCP network, including a ModbusRTU sub-network.



In a ModbusRTU network, all slaves have unique addresses from 1 to 247. In a ModbusTCP network, both **servers** and **gateways** are identified by their IP addresses. A Unit Identifier is also assigned to these devices, with value 255.

If a ModbusTCP client connects to the WS10 IP address and issues a message to unit 255, this message will be interpreted, executed and answered by the WS10. In this case, WS10 is acting as a **server**.

If the message received from the client is for a unit other than 255, WS10 will forward that message to the ModbusRTU sub-network. If in this sub-network exists a ModbusRTU slave whose address matches the specified unit, the message will be interpreted, executed and answered by this slave. The WS10 will receive and promptly forward this message to the client. In this case, WS10 is acting as a **gateway** between the ModbusTCP and ModbusRTU networks.

WS10 can act simultaneously as a **server** and **gateway**. To access WS10 as a **server**, the client should issue messages to unit 255. To use WS10 as a **gateway**, the client should issue messages to valid addresses on the ModbusRTU sub-network.

MODBUS TCP SERVER

When a new supervisory system is implemented, it is preferable to use WS10 as a ModbusTCP **Server**. This will assure better communication performance, since **Server** answers are always faster than **gateway** answers. If the system includes ModbusRTU devices, configure WS10 as a master of this network and set a polling interval to all variables of interest. By doing that, a ModbusTCP client may access all these variables directly from the WS10 registers, that will act as a **server** and concentrator. This solution brings other benefits, since registers read from the ModbusRTU networks may be used with other functionalities of WS10 (e-mail, HTML, ...).

When configuring the SCADA client, set the WS10 IP address as the server address, and define the Modbus address (or unit identifier) as 255. SCADA messages sent to the WS10 allow read or write operation on its registers. If a write operation is received and the written register is assigned to a ModbusRTU device, WS10 will automatically issue the appropriate write command to this device.

MODBUS TCP GATEWAY

Using WS10 as a ModbusTCP **gateway** is only recommended when WS10 is included in an existing system using ModbusRTU devices and SCADA software. When WS10 is included as a **gateway**, there is no need to change addresses of the ModbusRTU devices in the SCADA configuration. To adapt the SCADA for using WS10 as a **gateway**, it is only necessary to configure the IP address of WS10 in the SCADA ModbusTCP communication driver.

Performance of ModbusTCP network will be limited by the performance of the ModbusRTU sub-network. WS10 will master the ModbusRTU sub-network according to **MODBUS.CFG** configuration file, section *[Config]*.

When configuring the SCADA client, set the WS10 IP address as the server address, and keep sending messages to the original ModbusRTU slave addresses (1 to 247). WS10 will not verify contents of forwarded messages.

SERVER AND GATEWAY SIMULTANEOUSLY

WS10 can be used as a **server** and **gateway** simultaneously. In fact, the only difference is the unit identifier within the messages sent by the client. In resume, there are two distinct manners of accessing information in a ModbusRTU sub-network using the ModbusTCP network: accessing a WS10 register that contains the value of the last poll of the ModbusRTU network or accessing the ModbusRTU device directly.

PARAMETERS

ModbusTCP function is enabled when file **MODBTCP.CFG** is included in the WS10. Follows an example of this file:

<pre>[General] ServerPort = 502 ServerID = 255 IdleTimeOut = 300 MaxConnections = 4 MultipleConnections = true [Rules] Allow 10.1.1.0 mask 255.255.255.0 Allow 200.200.200.200 Deny 192.168.0.0 mask 255.255.255.0</pre>	<p>Connection parameters are defined in section [General]. Values shown for <i>ServerPort</i> and <i>ServerID</i> parameters are specified by the protocol, and should not be modified.</p> <p>In section [Rules] the allowed and blocked IP addresses are listed. <i>Deny</i> clauses have higher priority than <i>Allow</i> clauses. By defining an <i>IP mask</i>, one can allow or deny access for an IP range. To allow access from any IP, use <i>Allow All</i>.</p>
--	--

Section	Parameter	Values	Description
General	ServerPort	1 to 65535	TCP port for the ModbusTCP connection. Always set to 502.
	ServerID	1 to 255	Unit identifier for the WS10. Always set to 255.
	IdleTimeOut	0 to 65535	Maximum time with no data flow in seconds. After this time the connection is aborted by the WS10. If value 0 is defined, the connection will not be aborted due to inactivity. It is strongly recommended to set a non-zero value.
	MaxConnections	1 to 10	Maximum number of clients that can be simultaneously connected to the WS10.
	MultipleConnections	True or False	Permission for multiple connections from the same IP address. If true , a single client may establish multiple ModbusTCP connections to WS10. If false , a single connection is allowed from each client.
Rules	<p>Allowed Syntaxes:</p> <p>Allow xxx.xxx.xxx.xxx</p> <p>Allow xxx.xxx.xxx.xxx mask yyy.yyy.yyy.yyy</p> <p>Allow all</p> <p>Deny xxx.xxx.xxx.xxx</p> <p>Deny xxx.xxx.xxx.xxx mask yyy.yyy.yyy.yyy</p> <p>Deny all</p>		<p>Rules for connection authorization. Allow clauses define external IP addresses from which ModbusTCP connections will be allowed. Deny clauses define external IP addresses from which no connections will be allowed. By using a mask clause, an IP range can be defined. All clause means all IP values. Deny rules have higher priority. If no rule is defined, all IP addresses are blocked and ModbusTCP function can't be used by any client. At least one Allow clause must exist.</p> <p>xxx.xxx.xxx.xxx and yyy.yyy.yyy.yyy are IP and mask values, respectively.</p>

REGISTERS ADDRESS MAP FOR MODBUS TCP AND MODBUS RTU-SLAVE

WS10 registers are defined by its device and tag name. To externally access these registers using ModbusTCP or ModbusRTU-Slave, numerical addresses must be assigned. These addresses correspond to Holding Registers Addresses, as defined in Modbus specifications.

All Modbus registers defined in this map can contain values up to 16 bits long. To address an WS10 register containing a 32 bits value, it is necessary to create 2 Modbus registers, one associated to the upper 16 bits and the other to the lower 16 bits of the WS10 register, as in the following example:

```
_INTERNAL_/IN1CNT.Hi = 5
_INTERNAL_/IN1CNT.Lo = 6
```

by doing this, the Modbus register with address 5 is associated to the upper 16 bits of the IN1CNT register of the device `_INTERNAL_`, and the Modbus register with address 6 is associated to the lowest 16 bits.

Assignment of addresses to these registers is only necessary if one of the above protocols is in use, and is implemented in file **MAP.CFG**, to be included in the WS10. Follows an example for this file.

<pre>[RegMap] TT101/PV = 0 TT201/PV = 1 _ALARMS_/ALARM1 = 2 _INTERNAL_/IN1 = 3 _INTERNAL_/OUT1 = 4 _INTERNAL_/IN1CNT.Hi = 5 _INTERNAL_/IN1CNT.Lo = 6</pre>	<p>In section [RegMap] addresses are associated to already defined register names. Physical Modbus addresses are used (zero based). Addresses don't need to be contiguous, but it is recommended to improve performance and allow block operations. All addresses are Holding Registers, accessible using Modbus commands 3, 6 and 16.</p>
--	--

CONFIGURING THE MODEM TO START A DATA CALL – PPP CLIENT

When a modem is connected to one of the serial ports (usually the serial port 2), the WS10 can be configured to start a data call using PPP (Point to Point Protocol). The WS10 will act as a PPP client and at the other side a PPP server will respond to the connection request. The **PPP.CFG** file contains the configuration of the PPP client function.

Two WS10 functions may initiate a PPP connection:

- Periodical data transmission
- E-mail transmission

If any of these functions uses a PPP connection, a properly configured **PPP.CFG** file must be installed on the WS10.

Usually the WS10 dials to a Internet service provider to gain access to the internet, where the IP address of a SMTP server or host computer is reached to accomplish the desired function. The WS10 may also dial directly to a host computer.

The configuration of the **PPP.CFG** file depends on the modem type, and requires modem expertise to be successfully accomplished. Follows an example containing the configurations required when the internal WS10 modem (optional) is used:

<pre>[General] COM=2 Auth=1 Flow=2 Baud=38400 IdleTimeout=60 RetryDelay=10 [Connection] String0=ATZ Answer0=OK TimeOut0=5 Retries0=0 String1=ATX1E0 Answer1=OK TimeOut1=5 Retries1=0 [HangUp] String=ATH Answer=OK TimeOut=5 Retries=1 [Dial] Dial0=33334444 User0=novus Pwd0=suvon Timeout0=60 Retries0=0</pre>	<p>At [General] section the serial port connected to the modem is configured.</p> <p>[Connection]: At this section the AT commands required to establish the modem connection are defined. Up to 6 line of commands can be configured, containing the command, the expected answer, answer timeout and number of retries. These commands depend on the modem type.</p> <p>[HangUp]: At this section the command for terminating the connection is defined.</p> <p>[Dial]: Number to be dialed, username and password for authentication are defined. Two other phone numbers and users information can be defined, and will be used in case of failure on connection to the previous number.</p>
--	--

Section	Parameter	Values	Description
General	COM	1 or 2	Number of the serial port connected to the modem.
	Auth	0 to 4	PPP authentication: 0: No authentication. 1: PAP, user and password sent by the client– Usual option 2: CHAP, user and password sent by the client. 3: PAP, user and password sent by the server. 4: CHAP, user and password sent by the server.
	Flow	0, 1 or 2	Type of flow control between the WS10 and Modem. 0: No flow control. 1: XON/XOFF 2: RTS/CTS – Usual option
	Baud	300, 600, 1200, 2400, 4800, 9600, 19200, 38400	Baud rate in bits per second.
	IdleTimeout	0 to 65535	Maximum time with no data flow, in seconds. After this time the connection is aborted by the WS10. If the value 0 is defined, the connection will not be aborted due to inactivity. It is strongly recommended to set a non-zero value. Usual setting is 60 seconds.
	RetryDelay	0 to 65535	Time in minutes between connection retry after a connection error. Don't set a low value for this parameter if the WS10 is configured to receive data calls. Suggested value is 5 minutes.
Connection	String0	Modem command	First command sent to the modem to start the connection.
	Answer0	Modem Answer	At least one word of the expected response to the previous command.
	Timeout0	0 to 65535	Timeout in seconds for the modem to respond with the expected answer. If 0, the answer is waited forever. Avoid programming 0 to this parameter.
	Retries0	0 to 65535	Number of retries on the previous command.
	...		Up to 6 command blocks may be sent to the modem to establish the modem connection. Use only the necessary blocks, always from 0 to 5.
	String5, Answer5 Timeout5, Retries5		Last block of commands.
HangUp	String	Modem command	Command issued to the modem to terminate the connection. Usually ATH.
	Answer	Modem answer	Expected modem answer to the disconnect command. Usually OK.
	Timeout	0 to 65535	Timeout in seconds for the modem to respond with the expected answer.
	Retries	0 to 65535	Number of retries on the disconnect command.
Dial	Dial0	Phone number	First number to be dialed when the connection needs to be established. May be a internet access number or the modem number of a host computer.
	User0	User name	User name for the PPP connection.
	Pwd0	User password	User password for the PPP connection.
	Timeout0	0 to 65535	Maximum time in seconds for the PPP connection negotiation.
	Retries0	0 to 65535	Number of PPP connections retries if not established on the maximum defined time.
	...		Up to 3 PPP connection blocks may be defined. The WS10 tries to connect to each number, on the 0 to 3 sequence.
	Dial2, User2 Pwd2, Timeout2, Retries2		Last PPP connection block. The second and third connections are optional.

CONFIGURING THE MODEM TO RECEIVE A DATA CALL – PPP SERVER

The PPP server is implemented on the WS10 operating system, and is configured on the system file CHIP.INI. The WS10 can act as a PPP server and answer a phone call, negotiate modem and PPP connection and assign IP parameters to the calling computer. The WS10 may also be configured to answer data calls, as described in *Configuring the modem to start a data call*.

A remote computer may dial to the WS10 to:

- Access the pages hosted by the WS10.
- Establish a Modbus TCP connection
- Change configuration or upgrade firmware using FTP.
- Connect to the Telnet console for configuration and diagnostic purposes.

Follows an example of the PPP server configuration on the CHIP.INI file. The example presents the configuration used with the WS10 internal modem (optional) at serial port 2. Modem and IP network expertise are necessary to change these configurations.

<pre>[PPPSERVER] ENABLE=0 MODEMTRACE=0 COMPORT=EXT ADDRESS=192.168.1.1 REMOTEADDRESS=192.168.1.2 GATEWAY=192.168.1.0 AUTH=1 IDLETIME=60 FLOWCTRL=2 MODEM=1 USER0=ppp PASSWORD0=ppp BAUD=38400 INITCMD0=ATZ INITANSWER0=OK INITTIMEOUT0=3 INITCMD1=AT\N4\Q3\V2 INITANSWER1=OK INITTIMEOUT1=3</pre>	<p>The PPP server function is configured at the [PPPSERVER] section. Only make changes to this section!</p> <p>ENABLE: 0 to inhibit and 1 to enable the PPP server. Don't enable the PPP server if this function will not be used.</p> <p>MODEMTRACE: Usually 0. Set to 1 to debug modem commands through the Telnet console.</p> <p>COMPORT: Serial port connected to the modem. EXT for serial port 2 or COM for serial port 1. Usually set to EXT.</p> <p>ADDRESS: WS10 IP address for the PPP connection.</p> <p>REMOTEADDRESS: IP address that the WS10 will assign to the calling computer.</p> <p>GATEWAY: Defines the IP address of the gateway for PPP interface.</p> <p>AUTH: Type of authentication. 0 for none, 1 for PAP (usual option), 2 for CHAP.</p> <p>IDLETIME: Maximum idle time before the connection aborts. In seconds. Usual value is 60 seconds.</p> <p>FLOWCTRL: Flow control between modem and WS10. 0 for none, 1 for XON/XOFF and 2 for RTS/CTS. Usually 2.</p> <p>MODEM: Connection device. 1 for modem or 0 for serial cable. Usually 1.</p> <p>USER0 & PASSWORD0: If AUTH=1, defines the first authorized user for PPP connection.</p> <p>USER1 & PASSWORD1: If AUTH=1, defines the second authorized user for PPP connection. It is safer to always define the two users, to disable the default user (ppps/ppps).</p> <p>BAUD: Baud rate of the PPP serial connection. Maximum is 38400bps.</p> <p>INITCMDn, INITANSWERn & INITTIMEOUTn: Modem initialization commands, issued at the WS10 boot and after finishing the phone call. 3 block of 3 parameters defining the modem command, expected answer and answer timeout in seconds. (n = 0, 1 or 2).</p>
---	---

If the internal modem of the WS10 answers a data call but the calling computer reports that a connections could not be established, try to replace at CHIP.INI file the line `INITCMD1=AT\N4\Q3\V2` by `INITCMD1=AT\N0\Q3\V2`, and reboot the WS10. This change may solve some protocol incompatibilities detected on some PC modem cards.

DNS CONFIGURATION

When configuring IP addresses for e-mail and periodical data transmission functions, domain names can be defined for the servers. To resolve the domain names, WS10 needs access to a DNS Server.

If WS10 is using dynamic IP (DHCP enabled), possibly a DNS address has been assigned by the DHCP server. If WS10 is using a fixed IP or if the assigned DNS address is invalid, a DNS configuration can be established for the WS10 by including a file DNS.CFG, similar to the following example. If the WS10 has established a PPP connection to an external server, the DNS configuration assigned by this server will prevail, and the configuration described here will be ignored.

<pre>[General] PrimaryServer = 200.200.200.200 SecondaryServer = 200.200.200.201</pre>	<p>All DNS configuration is under the [General] section. The IP address of the main DNS is assigned to <i>PrimaryServer</i>. A secondary DNS IP may be assigned to <i>SecondaryServer</i>.</p>
--	---

Section	Parameter	Values	Description
General	PrimaryServer	IP Address	IP address for the primary DNS
	SecondaryServer	IP Address	IP address for the secondary DNS

SECURITY ISSUES ON THE WS10

TELNET ACCESS

Up to 2 authorized telnet users can be configured on the WS10. With factory settings, the WS10 has one configured telnet user (telnet/telnet). Since the second user is not configured, a default user is also enabled, with user name and password *tel*. The telnet console can be disabled, but it is not recommended since it is the main configuration tool for the WS10. To increase the security level, configure both telnet users.

Telnet configurations are located at the *[TELNET]* section of the system file *CHIP.INI* exemplified below:

<pre>[TELNET] ENABLE=1 TIMEOUT=10 LOGINDELAY=0 LOGINRETRIES=3 USER0=TELNET PASSWORD0=TELNET USER1=SUPERVISOR PASSWORD1=SUPER</pre>	<p>The Telnet function is configured at the [TELNET] section. Only make changes to this section!</p> <p>ENABLE: 1 to enable and 0 to disable the Telnet console.</p> <p>TIMEOUT: Time in minutes of inactivity before the telnet connection be closed by the WS10.</p> <p>LOGINDELAY: Enable progressive delay on the next login when an invalid username or password is submitted. 1 to activate and 0 to deactivate this feature.</p> <p>LOGINRETRIES: Maximum number of login retries before the WS10 closes the connection.</p> <p>USER0 & PASSWORD0: First Telnet user and password. Up to 19 characters each.</p> <p>USER1 & PASSWORD1: Second Telnet user and password. Up to 19 characters each. Always define the 2 users to disable the default user and password.</p>
--	---

FTP ACCESS

Up to 2 authorized FTP users can be configured on the WS10. With factory settings, the WS10 has one configured FTP user (ftp/ftp). Since the second user is not configured, the anonymous ftp user is enabled. It is not recommended to disable the FTP server, since it is the only path to access and change the configuration files. Always define the 2 FTP users to increase the security.

FTP configurations are located at the *[FTP]* section of the system file *CHIP.INI*, exemplified below:

<pre>[FTP] ENABLE=1 TIMEOUT=300 LOGINDELAY=0 USER0=TELNET PASSWORD0=TELNET ACCESSRIGHT0=0 USER1=SUPERVISOR PASSWORD1=SUPER ACCESSRIGHT1=0</pre>	<p>The FTP function is configured at the [FTP] section. Only make changes to this section!</p> <p>ENABLE: 1 to enable and 0 to disable the FTP server.</p> <p>TIMEOUT: Time in seconds of inactivity before the FTP connection be closed by the WS10.</p> <p>LOGINDELAY: Enable progressive delay on the next login when an invalid username or password is submitted. 1 to activate and 0 to deactivate this feature.</p> <p>USER0, PASSWORD0 & ACCESSRIGHT0: First FTP user and password. Up to 19 characters each. ACCESSRIGHT 0 allows read/write rights for the user, 1 allows read only rights.</p> <p>USER1, PASSWORD1 & ACCESSRIGHT1: Defines the second FTP user. Always define the 2 users to disable the default user and password.</p>
---	--

HTML PAGE SERVER

Authentication to the served pages can be enabled at the *[AUTH]* section of the *WEBS.CFG* file. Additional configurations are located at the *[WEB]* section of the system file *CHIP.INI*, exemplified below:

<pre>[WEB] ENABLE=1 ROOTDIR=A:\WEB HTTPPORT=80 USER0=WEB PASSWORD0=WEB</pre>	<p>The WEB function is configured at the [WEB] section. Only make changes to this section!.</p> <p>ENABLE: 1 to enable or 0 to disable the WS10 page server service.</p> <p>ROOTDIR: Defines the root folder of the web server files. All files located at this folder and its sub-folders are accessible using the web browser.</p> <p>HTTPPORT: HTTP listening port. Usually 80.</p> <p>USER0 & PASSWORD0: User name and password to access the PUT method of transferring files to the server root folder. Always define a user and password for this function.</p>
--	---

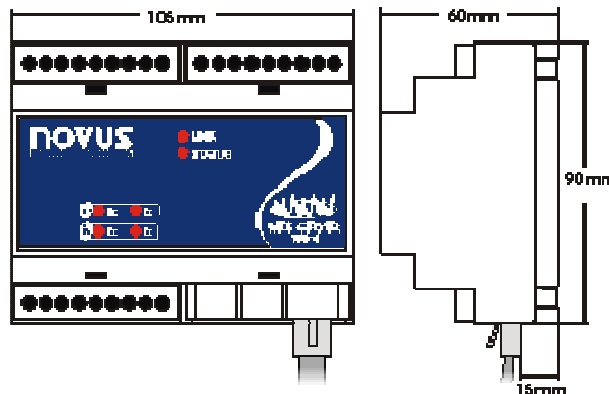
SPECIFICATIONS

GENERAL

Switching power supply, 100 to 240 Vac / 4VA.

Operating environment: 0 to 55 °C, 35 % to 85 % RH, no condensation.

ABS housing for DIN rail mount. Dimensions: 105 x 90 x 60 mm



COMMUNICATION INTERFACES

Two serial interfaces with 6 pin RJ12 connection. Options:

- RS232.
- RS485 with optical isolation. Modbus RTU master or slave protocol.
- Internal V34+ Modem

Ethernet 10/100. RJ45 connection.

Protocols: TCP/IP, PPP (client and server), HTTP, FTP, SMTP, DHCP, DNS, ModbusRTU (master and slave), ModbusTCP (server and gateway).

INPUTS

Four configurable inputs:

- Digital inputs for voltage pulse or dry contact. Flow measurement and totalizer function.
- Analog inputs for 0 to 5 V, 0 to 20 mA or 4 to 20 mA.

Resolution: 10 bits (1024 levels) for 0 to 5 V and 0 to 20 mA. 800 levels for 4 to 20 mA

Analog input accuracy: Better than 15 mV or 0,06 mA.

Pulse flow measurement time base accuracy: Better than 1s.

Pulse input maximum frequency: 200 Hz with debounce disabled.

Input impedance: >1 M Ω for analog voltage, 230 Ω for current.

Configurable input bias resistor: 10 K Ω

Allowable input configurations:

- 4 digital inputs
- 1 analog and 3 digital inputs
- 2 analog and 2 digital inputs
- 4 analog inputs

OUTPUTS

Two 250V/3A SPST relays.

REAL TIME CLOCK

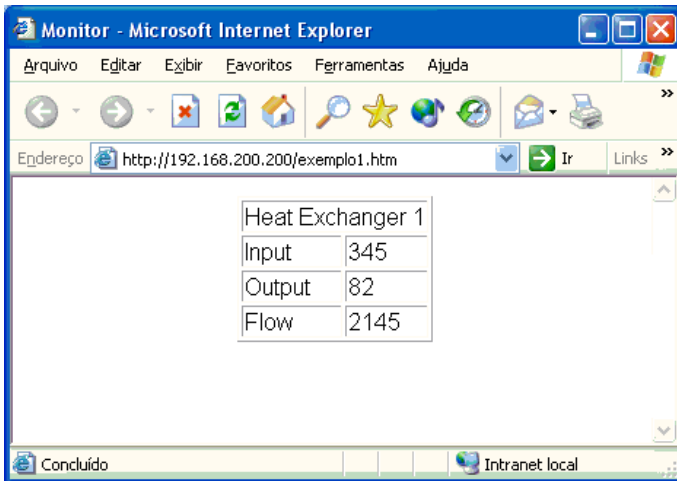
Accuracy: ± 3 minutes / month @ 25 °C

Battery: CR2032 lithium cell.

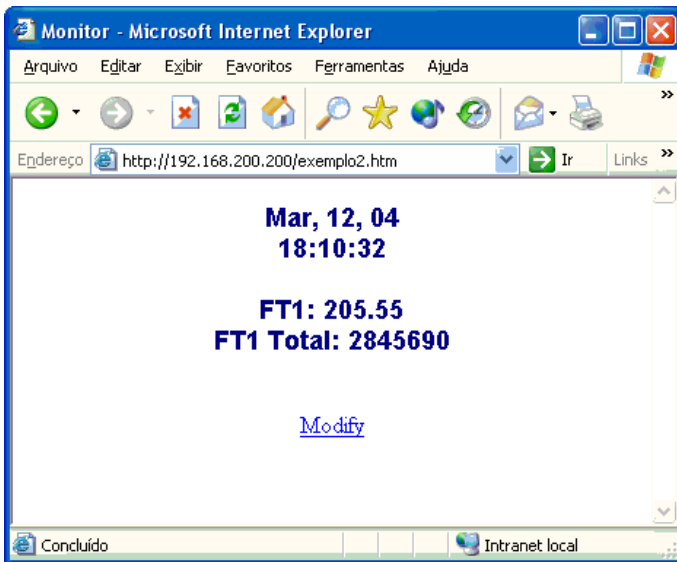
Battery life: 10 years

APPENDIX A

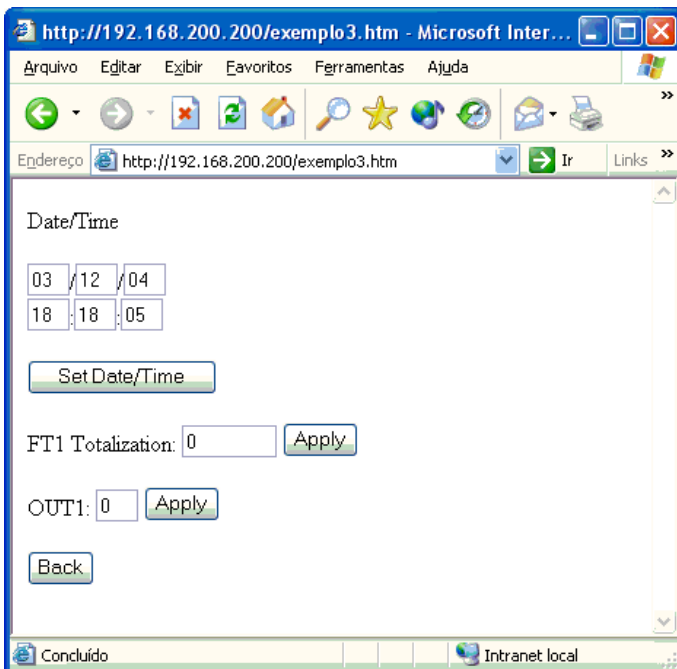
Screenshots of the HTML examples.



Example 1



Example 2



Example 3