PROLON CONTROL SYSTEMS
Herstedvesterstraede 56
DK-2620 Albertslund
Denmark
Tlf.: (+45) 43668060
Fax.: (+45) 43668061
e-mail: info@prolon.dk
Internet: www.prolon.dk

Lonbox® PID4000 Users Guide

Web Server and Data Logger with Ethernet Interface for LonWorks® Installations

Marts 2006

Document No. 02-PID4000-03

This manual is intended for use with the Lonbox® PID4000 software version 0.38

Printed in Denmark.

Prolon Control Systems
Herstedvesterstraede 56
DK-2620 Albertslund, Denmark

Contents

PROLON CONTROL SYSTEMS
Herstedvesterstraede 56
DK-2620 Albertslund
Denmark
Tlf.: (+45) 43668060
Fax.: (+45) 43668061
e-mail: info@prolon.dk
Internet: www.prolon.dk

Abstract

This manual provides detailed technical information for the Dynamic web server and Data Logger Lonbox® Series model PID4000.

The operating environment will be explained and the electrical and mechanical interface will be described.

Marts 2006

Introduction

General Overview

The Web Server and Data Logger Lonbox® Series model PID4000, is a unit used to register data in a log with an interval between each log entry. The unit is also used to monitor the logged data and to activate alarm handling, sending alarm and warning messages. The configuration and setup of the unit, is done with the embedded WEB server, serving WEB pages for each configuration step needed. The WEB server can be used with WEB pages made OEM for specific user purposes as visualization and operating equipment on a LonWorks® network.

- Data Logger
- Data Monitor
- Alarm Handler
- WEB Interface
- OEM WEB User Interfaces

Typically use

The Dynamic WEB server and Data Logger, Lonbox® Series model PID4000 is typically used to log energy consumption data as electricity, gas, water, heat and temperatures, on/off times, deviation from pre-set threshold values, among many others. The unit is also used for monitor purposes as handling of alarms and warnings. The unit has been used for quality control in building automation logging the functionality of the building and as an alarm unit sending alarms triggered be different sensors, as passive infrared detectors or other inputs. The unit is also used for OEM visualization with OEM WEB pages.

- Energy Registration
- Monitoring
- Alarm and Warning Handler
- Visualization and user interfaces
- Quality Control

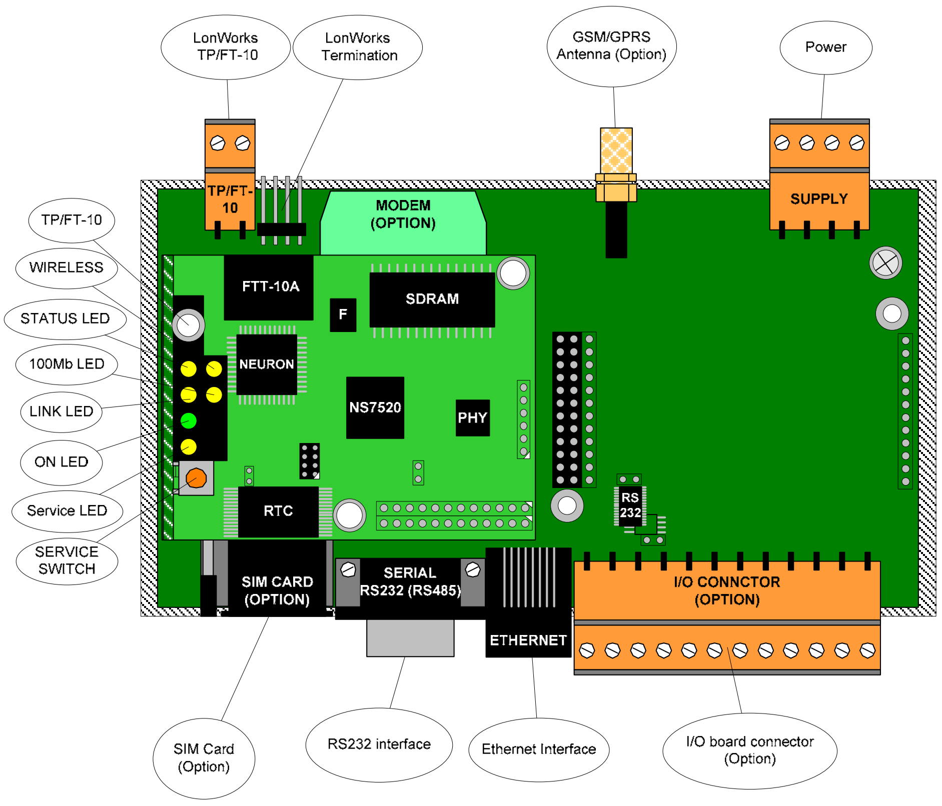
Hardware interface overview

The PID4000 Web Server can be mounted on a DIN rail. The Web Server and Data Logger, Lonbox® Series model PID4000 has several hardware interfaces:

This model is designed to log data either from a field network and the network used, is the LonWorks® TP/FT-10 or from a build-in input output module. The unit will log data from other units connected to the field network and typically it’s used for energy consumption registration and control.

To read out data from the unit, is used an Ethernet 10/100BaseT. But also a cellular mobile modem can be connected on the serial connection, using either GSM or GPRS communication technology, as you also can use a standard PSTN modem.

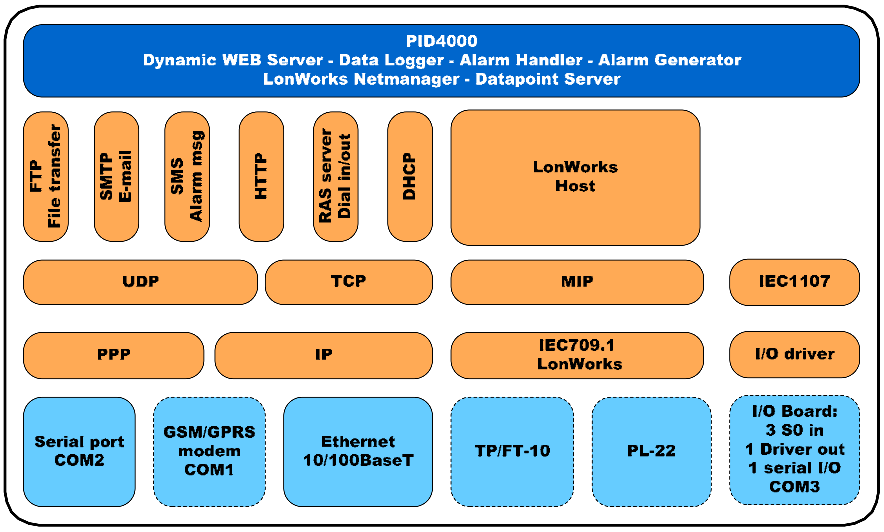
- LonWorks® TP/FT-10
- Ethernet 10/100BaseT
- Serial RS-232
- Cellular GSM / GPRS and PSTN modems



Figur 1

Embedded Software Overview

The Dynamic WEB server and Data Logger, Lonbox® Series model PID4000 has several embedded software functions.



Internet Protocol

The unit support the very common technology used on the internet and many other computer networks, known as the Internet Protocol (IP technology).

The Web Server

The embedded WEB server is a software function used for serving Web pages and these pages can be viewed with a standard WEB browser as the Microsoft Explorer browser, delivered with the any Microsoft operating system today. The WEB server is based on HTTP on IP technology.

Alarm and warning E-Mails

The unit is supporting sending e-mails using the SMTP protocol IP technology. The e-mails are used to send alarm messages and warnings, triggered by an alarm handler monitoring the logged data from the LonWorks® network.

Alarm and warning SMS on GSM

The unit is supporting sending SMS messages using the GSM cellular network. The SMS messages are used to send alarm messages and warnings, triggered by an alarm handler monitoring the logged data from the LonWorks® network.

Data files

When reading the data from the unit you can request it as a computer file. The used file is the standard XML format which is used on many computer platforms. The XML file is an open platform for file formats, including self description inside the file. Many software programs are today supporting the XML files and the Microsoft Office software suite, are today in version XP, giving you the feature reading data as WEB requests including the XML format. You can read out data from the Data Logger directly into Excel spread sheet.

Installation

Before you start

Before using the data logger you need to configure the Ethernet and LonWorks and network interfaces. After the installation, the configuration of the data logger it-self is done by using the build-in web page server.

Ethernet Network Interface setup

If your network installation allows you to use the default network address, you can directly connect to the data logger at:

Default IP address	192.168.1.100
Default subnet mask	255.255.255.0

If this address cannot be used, and it is not possible to communicate from your computer to this default address, you have to setup the IP configuration by using the RS232 port.

Initial installation of Ethernet using RS232 connection

To configure the IP port, power off the PID4000 and connect a terminal or a PC with a terminal emulator (Windows has the build-in HyperTerminal) to the serial port, using a NULL modem cable.

Set up the terminal software to match the settings of the RS232 port of the PID4000, which has default setting of **9600 baud, no parity, 8 bit data and 1 stop bit**.

During startup, no handshake is used by PID4000, and the terminal software shall therefore be configured for no handshake (flow control).

After configuring and connecting you terminal (PC), power up the Lonbox PID4000.

When the configuration screen appears, review the configuration and change it to match your LAN installation. The Lonbox PID4000 supports both static and dynamic IP (DHCP) addressing, but it is recommended to use a static IP address since the device is considered to be a server on the network.

If it is not possible to connect to the PID4000, one reason could be that the settings of the PID4000's RS232 port have been changed from the default settings (in that case, try other settings in the terminal software).

LonWorks Network Interface setup

The LonWorks network interface installation consists in assigning a network domain address and setting the data logger online.

There are two methods for this, either this is done using a traditional network management tool like *LonMaker for Windows,* or the setup can be performed using the build in web service “SystemSetDomain”. The web service is normally invoked from a custom made script.

Configuration

All configuration of the PID4000 data logger is done through the use of the embedded web server. There are two different methods available.

For normal use, the easiest way is to use a normal internet browser and to walk through the different web pages filling in the required configuration and submitting the web page forms.

An alternative way of changing the configuration is by using the web service interface. This is the best way to manage configuration if this is done automatically for example by the use of configuration scripts.

You can connect to the web server either using the Ethernet, or the RS232/PPP connection. Specify the IP address configured during installation in you browsers address line and view the description on the Lonbox PID4000 default web page.

For configuration of the data logger a default administrator password has been added to the web server:

Username: Admin
Password: admin

Configuration Overview

Before the data logging will start, the clock in the PID4000 has to be set. Click on the following link to set the clock. This step can be completed after configuring the data logger and alarm handling.

To set up the data logger; devices has to be added to the network database and the data logger channels have to be configured.

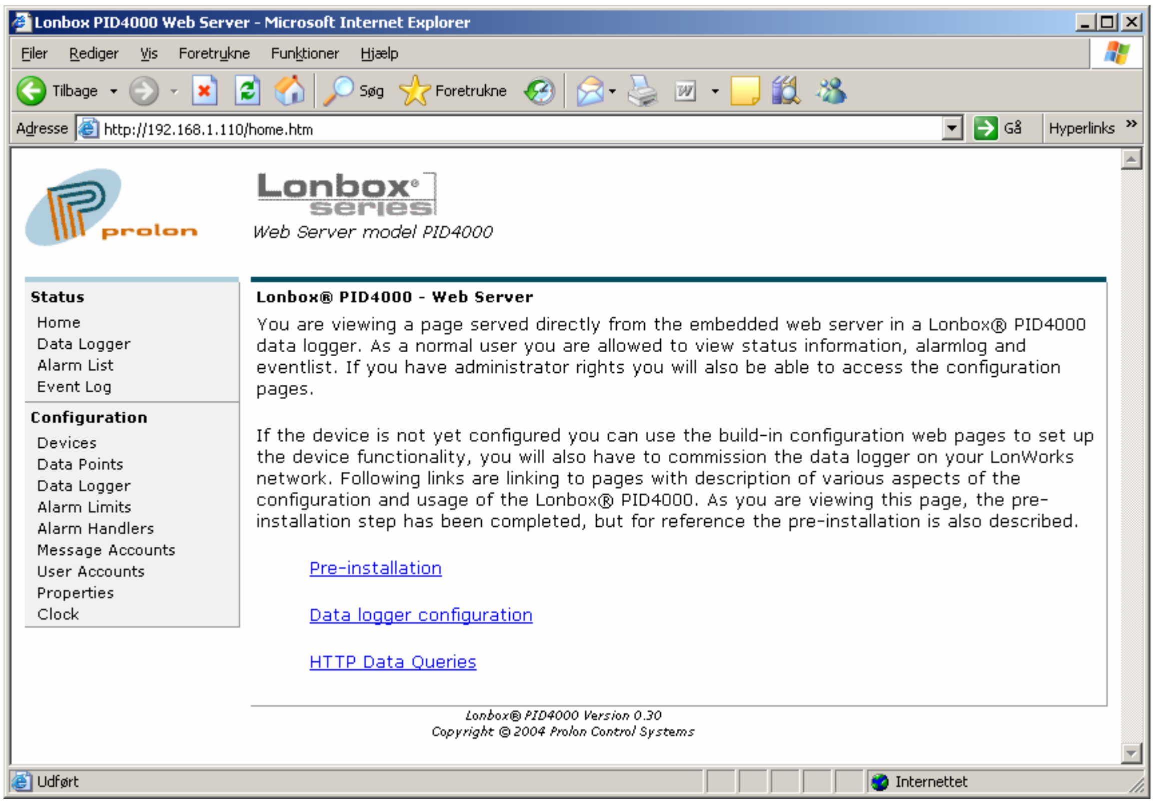
The network database contains a list of relevant device installed on the LonWorks network. The purpose of the device list is to store addresses for the devices containing data values that should be logged.

Not all devices in the installation have to be entered into the list, only devices that need to be used in the data logger setup have to be entered. After a device has been inserted into the devices list the data logger configuration page can be used to add definitions of data values that should be logged.

Note that all configuration pages are password protected. Only the *Administrator* and the *Service* users can access the configuration pages.

The Web user interface

When you are connected to the embedded WEB Server, you will get the following page on your browser screen. This is the start page giving the entry to the user interface.


Lonbox PID4000 Web Server - Microsoft Internet Explorer

http://192.168.1.110/home.htm

Lonbox® series
Web Server model PID4000

Lonbox® PID4000 - Web Server

You are viewing a page served directly from the embedded web server in a Lonbox® PID4000 data logger. As a normal user you are allowed to view status information, alarmlog and eventlist. If you have administrator rights you will also be able to access the configuration pages.

If the device is not yet configured you can use the build-in configuration web pages to set up the device functionality, you will also have to commission the data logger on your LonWorks network. Following links are linking to pages with description of various aspects of the configuration and usage of the Lonbox® PID4000. As you are viewing this page, the pre-installation step has been completed, but for reference the pre-installation is also described.

Pre-installation

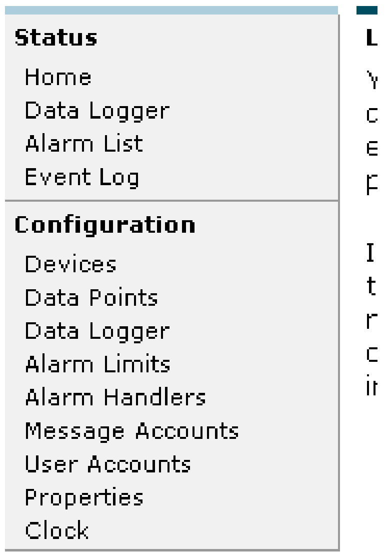
Data logger configuration

HTTP Data Queries

Lonbox® PID4000 Version 0.30
Copyright © 2004 Prolon Control Systems

In the left side is a menu and when you point and click with your left mouse button on one of the menu items, you can select that function in the menu.

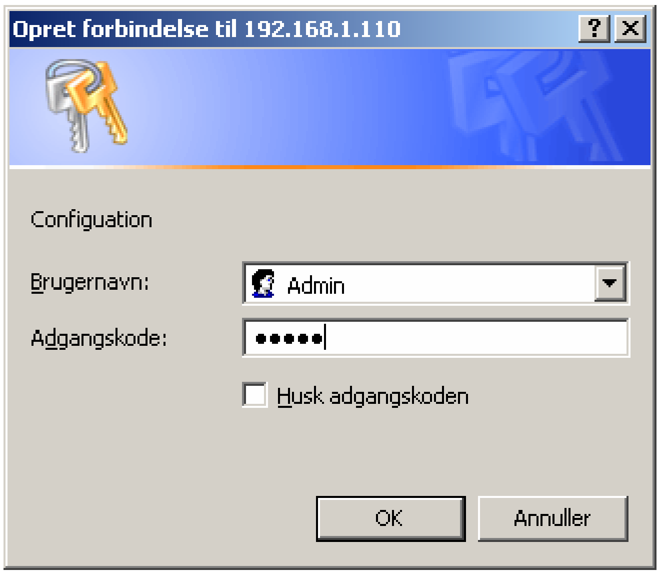
Use the web pages step by step to configure your data logger. Hereunder is the menu giving a short overview what to configure.



Please use the web pages and the help information on these pages, to operate this web based data logger interface.

Step by Step configuration, quick start

1. Start adding a device using the menu entry, DEVICE



Opret forbindelse til 192.168.1.110

Configuation

Brugernavn: Admin

Adgangskode: ●●●●●

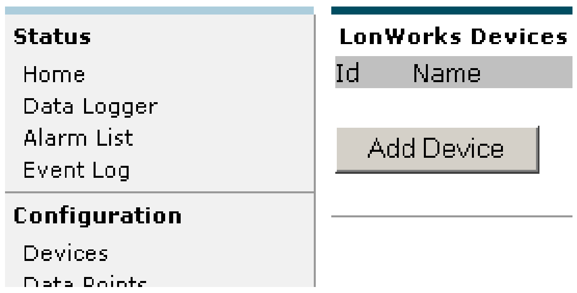
☐ Husk adgangskoden

The first time you select to use a menu entry in the configuration section you are asked for at user name and password, please use:

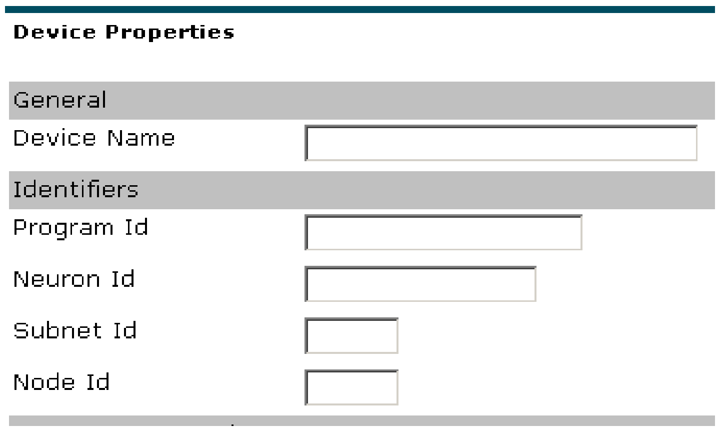
”Admin” as user name and
”admin” as the password.

The username and password will time out after a while without using the WEB pages, please re-enter the above again.

2. Hereafter you can add a new device.



3. When adding the devise it is important, that you type the correct subnet and node Id from the LonWorks network and remember also to give the device a name.



4. Hereafter add a data logger channel using that entry in the menu. Remember to give the data logger channel a name and a SNVT type index number. In here you also specify logging interval in seconds.

5. After this you need to set the clock and you are logging your first data.

Alarm Messages

If the alarm limit configuration specifies that the alarm should be send as an email, the data logger will always first send to the primary account.

If the sending of the alarm notification to the primary account fails, or if the alarm notification configuration specifies that backup should always be send, the alarm notification is send to the backup account.

The following keywords can be used in the email text and will be replaced by their values when the email is send.

Keyword	Description
<DataLoggerName>	The name of the data logger as specified in the name field on the properties page.
<HostAddress>	The IP address of the host in the dot form like "192.168.1.100".
<DeviceId>	The device id of the device where the data logger channel value is read.
<DeviceName>	The name of the device where the data logger channel value is read. The device name can be changed on the device configuration page.
<ChannelId>	The channel id of the data logger channel that caused the alarm.
<ChannelName>	The channel name of the data logger channel that caused the alarm. The channel name can be changed on the channel configuration page.
<AlarmId>	The alarm id of the alarm configuration.
<AlarmTimestamp>	The timestamp of the alarm.
<AlarmLimit>	The alarm limit that was exceeded.
<AlarmValue>	The data value that caused the alarm.
<CurrentValue>	The current value of the data logger channel.

Web Data Queries

Currently the logged data is available in XML format. There are two different XML files which contain logged data information. One containing current values, for all active channels and one containing a series of readings from a single channel.

An alarm list and an event log is also available in XML format. The alarm list and the event log can also be viewed on web pages that are updated dynamically.

Latest Logged Values

The current values file contains the last read value for all active channels. These values are in the file latest.xml. The latest data can also be viewed on the status page, where data is formatted for viewing and the values are refreshed automatically. This page requires Microsoft IE.

Data is not fetched from the LonWorks network when the latest.xml file is being read. Instead the latest.xml file contains the values from the last logging. This means that the data value is not the actual value in the device being logged, you can use the nvfetch data query to read actual real-time data.

Current Values (network variable fetch)

The current values for one or more specific network variables can be read online. When the page is processed, the values are read in real time over the LonWorks network and inserted to the page. Currently formatting of the network variables are not supported, the network variable is send as hexadecimal digits with space as a separator between the bytes.

The nvfetch page currently support only two forms of requests. You can either request the value of a single network on a device, giving the network variable index; or you can request the value of all network variables on the device.
To view current values, use the "nvfetch.xml" page and specify the device id and optionally a network variable index as parameters. The following link will display the value of the network variable with index 1 (nvindex=1) for device 0 (device=0).

“nvfetch.xml?device=0&nvindex=1“

Device Status Information

The status information for a device defined in the devices list can be read online. The device status can be use for service and maintainance purpose and is a list containing transmission statistics, node status, reset cause, firmaware and error log information for the device.

To view device status, use the "status.xml" page and specify the device id as a parameter. The following link will display status for device 0.

status.xml?device=0

Alarm List

On the alarmlist page, is the current list of alarms and their state presented. The page is a presentation of the data that also is available from the alarmlist.xml page.

Event Log

On the eventlog page, is the current list of events presented. The page is a presentation of the data that also is available from the eventlog.xml page.

The eventlog supports the CGI parameters "*ch*", "*idge*" and "*maxcount*". See the description of these parameters below.

Data Series

Logged data can be read from the file data.xml. When reading data; the channel, and optionally a data range, should be specified. Channel and data range are specified using CGI parameters "ch" and "idgt". If the file *data.xml* is read without any parameters, no data logger values will be returned, just the XML header.

Parameters are specified after a question mark (?) and each parameter is separated with an ampersand (&).

```
\\server\page?parameter&another_parameter
```

Every parameter contains a field name and a value. The field name must be specified on the left side of the equal sign and the right side specifies the value for the field.

```
field=value
```

The value can be a list. When the value is a list, each element in the list should be separated by a plus sign (+). This is used when reading several channels, such as ch=1+2+3 which means reading channel 1, 2 and 3.

The following parameter names are supported, note that names are casse sensitive.

ch	Specifies channel number for the selected data logger channel(s). This is a required field, so one or more channels should always be specified.
idge	The id greater than or equal to field is optional, The field specifies that the request data should only contain data with id greater than or equal to the specified id. When no *idge* is specified all data for a channel is specified (same as idge=0). When more than one channel is read in a single file, the idge field should contain values for all channels or no idge should be specified. Below is an example where data is requested for two channels; channel 1 data with an id greater than 1302 and channel 2 with id greater than 543. The resulting URL will be: http://192.168.1.100/data.xml?ch=1+2&idge=1302+543.
last	Specifies that only the last measures for the given channel(s). The last value limits data for all channels selected by the "ch" field. To receive the last 100 loggings for channel 1, 2 and 3 use the

	following URL: http://192.168.1.100/data.xml?ch=1+2+3&last=100
maxcount	Limits the number of values for the specified channel. When more than one channel is specified, the maxcount field applies to all channels. This parameter can be used in combination with the "idge" field to browse values in steps of for example 100 values. The URL below will read 100 values from channel 1, starting from id 1302. http://192.168.1.100/data.xml?ch=1&idge=1302&maxcount=100
date	This will return loggings where the timestamp date matches the date specified. The data should be specified in the form YYYYMMDD The URL example will read all loggings for channel 1 and 2 for the date 20-Oct-2004.. http://192.168.1.100/data.xml?ch=1+2&date=20041020

Web Services

The Lonbox PID4000 provides a http based command interface for accessing the web services available in the server. The command interface is based on simple http address entries with additional CGI parameters similar to the data query interface.

This way of implementing web services makes it possible to use these directly from a a web client address field, and also to use these from client side script languages like java script. Java script libries supporting the use of the web service interface is available from Prolon Control Systems.

The commands are invoked using the following syntax:

"http://x.x.x.x/CmdName?CmdPar1=X&CmdPar2=Y"

the above example shows an invocation if the command "CmdName" with two parameters "CmdPar1" with value X and CmdPar2 with value Y.

Note that the command name and parameters are case sensitive.

The following two examples show the use of the HTTP command interface.

Setting clock with only changing the time (not the date) will look like:

"http://192.168.1.100/SystemClockSet?hour=10&minute=24&second=12

When invoking a web service, the result of the web service is returned as a XML document, in the form as shown in the example below.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>
<LonboxWebServer VERSION="1.0.0" Name="" Ts="2004-10-22T11:27:22">
  <SystemClockSetResponse>
    <Result>OK</Result>
  </SystemClockSetResponse>
</LonboxWebServer>
```

Device Management Services

Listing devices

This service will return a response containing a XML list with all devices installed. The list contains attributes and status of each device.

Command name: DeviceList

No parameters are used for this service.

Adding a device

Command name: DeviceAdd

Name	Optional	Description
id	Yes	Identification of the device. Note that either id or name has to be specified. Use the id if possible since it is the best performing.
name	Yes	Name of the device. Note that either id or name has to be specified. Use id if possible since specifying the name requires a lookup.

Note that there is a more high level service available for installing a device in the system. This service adds the device including its predefined data points and log channels in a single service call; the behaviour is based on a XML installation template.

Deleting a device

Command name: DeviceDelete

Name	Optional	Description
id	Yes	Identification of the device. Note that either id or name has to be specified. Use the id if possible since it is the best performing.
name	Yes	Name of the device. Note that either id or name has to be specified. Use id if possible since specifying the name requires a lookup.

Updating device attributes

Command name: DeviceUpdate

Name	Optional	Description
id	Yes	Identification of the device. Note that either id or name has to be specified. Use the id if possible since it is the best performing.
name	Yes	Name of the device. Note that either id or name has to be specified. Use id if possible since specifying the name requires a lookup. When an id is specified to identify the device, the name can be specified if you want to update the device name.
nid	Yes	Update the neuron id of the device.
pid	Yes	Update the program id of the device
loc	Yes	Update the location of the device

Registering device node object interface

For the PID4000 to be able to use node object functions like file transfer, the network variables for the node object need to be registered. This service registers network variable relation to a LonMark object (function block) in the device. The service is intended for use in custom scripts automating the installation process.

Command name: DeviceSetNvInfo

Name	Optional	Description
id	Yes	Identification of the device. Note that either id or name has to be specified. Use the id if possible since it is the best performing.
name	Yes	Name of the device. Note that either id or name has to be specified. Use id if possible since specifying the name requires a lookup.
loi	No	LonMark object index
lmn	No	LonMark member number
idx	No	Network variable index on the device

Commisioning a device

Command name: DeviceCommission

Name	Optional	Description
id	Yes	Identification of the device. Note that either id or name has to be specified. Use the id if possible since it is the best performing.
name	Yes	Name of the device. Note that either id or name has to be specified. Use id if possible since specifying the name requires a lookup.
nid	Yes	Specify the neuron id of the device

Decommissioning a device

Command name: DeviceDecommission

Name	Optional	Description
id	Yes	Identification of the device. Note that either id or name has to be specified. Use the id if possible since it is the best performing.
name	Yes	Name of the device. Note that either id or name has to be specified. Use id if possible since specifying the name requires a lookup.

Loading an application image into a device

Command name: DeviceLoad

Name	Optional	Description
id	Yes	Identification of the device. Note that either id or name has to be specified. Use the id if possible since it is the best performing.
name	Yes	Name of the device. Note that either id or name has to be specified. Use id if possible since specifying the name requires a lookup.
path	No	

Reading the current attributes of a device

Command name: DeviceGet

Name	Optional	Description
id	Yes	Identification of the device. Note that either id or name has to be specified. Use the id if possible since it is the

		best performing.
Name	Yes	Name of the device. Note that either id or name has to be specified. Use id if possible since specifying the name requires a lookup.

Reading a device file using LonMart file transfer

Command name: DeviceFileRead

Name	Optional	Description
id	Yes	Identification of the device. Note that either id or name has to be specified. Use the id if possible since it is the best performing.
name	Yes	Name of the device. Note that either id or name has to be specified. Use id if possible since specifying the name requires a lookup.
index	No	The file index in the device.
path	Yes	The full local path in the web server file system where the file will be stored. If no path is specified the file content will be returned in the XML response document in packed hex format.

Writing a device file using LonMart file transfer

Command name: DeviceFileWrite

Name	Optional	Description
id	Yes	Identification of the device. Note that either id or name has to be specified. Use the id if possible since it is the best performing.
name	Yes	Name of the device. Note that either id or name has to be specified. Use id if possible since specifying the name requires a lookup.
index	No	The file index in the device.
path	Yes	The full local path in the web server file system where to store the file.
data	Yes	Optionally the content of the file can be specified in with this attribute. The data is specified in packed hex format.

Resetting a device

Command name: DeviceReset

Name	Optional	Description
id	Yes	Identification of the device. Note that either id or name has to be specified. Use the id if possible since it is the best performing.
name	Yes	Name of the device. Note that either id or name has to be specified. Use id if possible since specifying the name requires a lookup.

Changing address table of a device

Command name: DeviceUpdateAddrConfig

Name	Optional	Description
id	Yes	Identification of the device. Note that either id or name has to be specified. Use the id if possible since it is the best performing.
name	Yes	Name of the device. Note that either id or name has to be specified. Use id if possible since specifying the name requires a lookup.
index	No	Index into the devices address table
type	No	
size	Yes	Default 0
node	Yes	Has to be specified for subnet / node addressing
member	Yes	Has to be specified for group adress type.
domain	Yes	The domain index, default value is 0.
retry	Yes	Default = 3
repeat	Yes	Default = 2
transmit	Yes	Default = 5
receive	Yes	Default = 4
group	Yes	Has to be specified for group address type.
subnet	Yes	Has to be specified for subnet / node addressing.

Changing network variable configuration of a device

Command name: DeviceUpdateNvConfig

Name	Optional	Description
id	Yes	Identification of the device. Note that either id or name has to be specified. Use the id if possible since it is the best performing.
name	Yes	Name of the device. Note that either id or name has to be specified. Use id if possible since specifying the name requires a lookup.

Data Points Services

Listing data points

This service will return a XML document with a list of current data points created in the system.

Command name: DataPointList

No parameters needed for this service.

Adding a data point

Command name: DataPointAdd

Name	Optional	Description
pid	Yes	If needed this can specify a specific data point id that will be used for the add. Note that if a data point is already defined for the id specified, this data point will be overridden by the new data point.
name	Yes	The name of the new data point.
device	No	The device id of the device.
nvindex	No	The network variable index for the data point.

Deleting a data point

This service deletes the specified data point.

Command name: DataPointDelete

Name	Optional	Description
id	Yes	Identification of the data point. Note that either id or name has to be specified. Use the id if possible since it

		is the best performing.
name	Yes	Name of the data point. Note that either id or name has to be specified. Use id if possible since specifying the name requires a lookup, also data point names are not requied to be unique.

Updating the attributes of a data point

Command name: DataPointUpdate

Writing to a data point

Command name: DataPointSetValue

Name	Optional	Description
id	Yes	Identification of the data point. Note that either id or name has to be specified. Use the id if possible since it is the best performing.
name	Yes	Name of the data point. Note that either id or name has to be specified. Use id if possible since specifying the name requires a lookup, also data point names are not requied to be unique.
rawvalue	No	The value that has to be written to the specified data point. The value is in packed hex format.

Reading the current value of a data point

This service will return a XML document with a list containing all the specified data points and their values. The values are in packed hex format.

Command name: DataPointGetValue

Name	Optional	Description
id	Yes	Identification of the data point. Note that either id or name has to be specified. Use the id if possible since it is the best performing. The id parameter can be usedto specify multible data points, both using comma seperation and range syntax. Specifying a range, using '-' dash syntax. The id 3-20 will return the values of all 18 data points in the range 3 to 20. Specifying a list usiing ',' comma syntax, A comma can be used to separate the individual id list 1,3,11 will

		return the 3 data points. The two formats can be combinded like 1,3,20-41
name	Yes	Name of the data point. Note that either id or name has to be specified. Use id if possible since specifying the name requires a lookup, also data point names are not requied to be unique.

Log Channel Services

Listing log channels

Command name: LogChannelList

Name	Optional	Description
did	Yes	Device id, this can limit the query to only return log channels for a specific device. If not specified all log channels are returned.

Adding a log channel

Command name: LogChannelAdd

Deleting a log channel

Command name: LogChannelDelete

Name	Optional	Description
id	No	Log channel id.

Updating the attributes of a log channel

Command name: LogChannelUpdate

Name	Optional	Description
id	No	Log channel id.

Clearing the values logged on a log channel

Command name: LogChannelClear

Name	Optional	Description
id	No	Log channel id. This value can be set to the string “ALL” which will clear all log channels.

Adding a log value using a web service

Command name: LogChannelAddValue

Name	Optional	Description
id	No	Log channel id.
value	No	Value specified as a interfere number. The range is a 32-bit.
accumulate	Yes	Add the value to the last value logged. If not specified the value will not be accumulated.

Enableing and disabling logging

Command name: LogChannelEnable

Name	Optional	Description
id	Yes	Identification of the log channel. Note that either id or name has to be specified. Use the id if possible since it is the best performing, also log channel names are not required to be unique. The id parameter can used to specify multible log channels, both using comma seperation and range syntax. Specifying a range, using '-' dash syntax. The id 3-20 will return the values of all 18 log channels in the range 3 to 20. Specifying a list usiing ',' comma syntax, A comma can be used to separate the individual id list 1,3,11 will identify 3 log channels. The two formats can be combinded like 1,3,20-41
did	Yes	Optional device id, with the parameter all data point on a device can be enabled or disabled with a single web service action. If this is specified the id and name attributes are unused.
name	Yes	Name of the log channel. Note that either id or name has to be specified. Use id if possible since specifying the name requires a lookup, also data point names are not requied to be unique.
enable	Yes	The value 1 enables the selected log channel, and the value 0 disables the selected channel. Default value is 1

		that enables the log channel

System Level Services

Setting serial number

The set serial number command is provided for use when the data logger has been completely reset. The original serial number can be found on a label placed on the circuit board inside the PID4000 box.

Command name: SystemSetSerial.

Name	Optional	Description
serial	No	Eight digit serial number.

Warning:

Do not change the serial number it can jam the Ethernet communication. The serial number is normally set during manufacturing. This command is intend for use only, if the serial no. has been lost during a complete data reset when the battery is disconnected or removed.

Reading the current time from system real time clock

This command is used to read the current clock.

Command name: SystemClockGet

The command will return a response with the current time in an XML document.

Setting system real time clock

This command will set the internal real time system clock to an absolute value as specified.

Command name: SystemClockSet

Name	Optional	Description
year	Yes	Four digit year
month	Yes	Two digit month 1 .. 12
day	Yes	Two digit day 1 .. 31
minute	Yes	Two digit minute 0 .. 59
Sscond	Yes	Two digit second 0 .. 59

Relative adjustment of system real time clock

Adjustment of the clock with a relative value instead of specifying a specific clock value. One of the advantages of this command is that it is less dependend of the transmission times because it requires only a single transmission to verify the clock, and the adjustment command itself does not have to be taken into accound when adjusting the clock. This is especially usefull for slow PPP connection types like GSM.

Command name: SystemClockAdjust

Name	Optional	Description
second	no	Adjustment in seconds, valid range is -3600 .. 3600 seconds.

Setting IP address

This command is used for changing the Ethernet IP address. All paramters are optional so that it is possible to change only the ip address without changing or specifying the mask or default gateway. Change will not take effect until after a system reset.

Command name: SystemSetIP

Name	Optional	Description
address	Yes	Four digit year
mask	Yes	Two digit month 1 .. 12
gateway	Yes	Two digit day 1 .. 31

Communication port options

Command for setting serial communication port options.

Command name: SystemCommPort

Name	Optional	Description
type	Yes	Communication port type, can be one of the following values: - "none": port not used. - "direct" : direct PPP connection - "standard" : standard modem - "gsm" : GSM modem connection - "gprs" : GPRS modem connection
speed	Yes	Serial baudrate, valid values are 1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200.

Modem options
Command for setting modem parameters.

Command name: SystemModem

Name	Optional	Description
init	Yes	Modem initialization string
answer	Yes	Number of rings before auto answer 1 .. 10
pin	Yes	Four digit pin-code

PPP configuration
Command for changing PPP parameters.

Command name: SystemPPP

Name	Optional	Description
local	Yes	IP address of the local PPP interface.
peer	Yes	IP address assigned to peer IP interface.
compression	Yes	Van Jacobsen IP header compressio value should be ‘0’ or ‘1’. Wheren 0 means off and 1 means on.

System reset
This command can be used to reset (worm boot) the system. Issuing this command will course the system to reset and connection to the system will be temporary lost. The command has no parameters.

Command name: SystemReset

Setting LonWorks network interface domain
Configuring the LonWorks network interface with domain information and setting the device configured online.

Command name: SystemSetDomain

Name	Optional	Description
len	No	Length of network domain id (only 1 supported)
id	No	Network domain id (1-255)
subnet	No	Network subnet address (1-255)

node	No	Network node address (1-127)

Unconfiguring the LonWorks network interface
This service will cause the PID4000 to leave the configured LonWorks domain and set the interface state to unconfigured.

Command name: SystemUnconfigure

Get information about last service pin message received
This service will return an XML file with information about the last service pin message received by PID4000. The service can be used for automating installation of devices using scripts.

Command name: SystemGetLastServicePin

Reading system information
This service will return an XML file with information about the system. The file contains information about application, Ethernet and LonWorks network interface configuration, file system status etc.

Command name: SystemInformaiton

Building you own embedded web site

Lonbox web services scripts

Bacground updating of data points

There are two primary functions for automatic background update in the datapoint.js script library. The functions are used for managing an automatic update of a list of data point values.

```
StartBackgroupdUpdate(pointList, updateHandler, errorHandler)
StopBackgroundUpdate()
```

Calling StartBackgroundUpdate() update will start an automatic update of values with 5 second interval. The interval can be changed from the default 5 seconds using the function:

```
SetBackgroundUpdateInterval(interval)
```

The interval is given in milliseconds.

Reading and writing data points

Accessing single points are supported via a read and a write function. The functions are asynchronous and will not block while the action is completed. Callback handler will be called when the function completes. The script library contains a queue function so that several writes can be performed without waiting for the previous write to complete.

```
DataPointWrite(channel, value, updateHandler, errorHandler)
DataPointRead(pointList, updateHandler, errorHandler)
```

All values are in raw packed hex format, meaning that each byte in the value is represented by two hexadecimal digits.

Basic Web Visualization With Lonbox PID4000

This example demonstrates the use of scripting to visualize dynamic values from an automation system on a web page. The examples use the Lonbox PID4000 Web Server to interface the automation system. The example is available as downloadable HTML and script files by contacting “support@prolon.dk”

When delivered, the PID4000 contains a number of build-in web pages mainly for service and configuration purpose. In addition to this user defined html, script and graphic pages can be downloaded to the embedded flash disk. When customizing the web server by adding your own web pages, it is easy to read and write values from the automation system through the use the build-in service functions. In addition Service functions for system management and configuration is also available through this interface.

To interface the automation system, the PID4000 in its basic version has a LonWorks TP/FT-10 interface. The PID4000 can be integrated into a LonWorks network using the existing management system or the PID4000 can be used to install and manage a new network using build-in network management services. In addition to this, the PID4000 will be available in different variants with local I/O; this will include direct digital and analog inputs and interfaces for other bus systems.

All samples are based on an identical data point configuration. Start by configuring you PID4000 with similar data points.

Example Data Point Configuration

The data point configuration is used in the examples to demonstrates the use of scripting to visualize dynamic values from an automation system on a web page. All examples are based on the same data point configuration.

To give an example close to possible systems, data points from two different application types have been selected. The selected data points show possible use in a Lighting system and in a HVAC application.

- Lamp Actuator Input
- Lamp Actuator Feedback
- Temperature Sensor
- Temperature Setpoint

To get the full benefit of the examples you should configure a PID4000 with these data points. Pay attention that id, names and SNVT types are correct for the data points you configure.

Id	Name	SNVT	Format
0	nviLampValue	95	SNVT_switch
1	nvoLampValueFb	95	SNVT_switch
2	nvoSpaceTemp	105	SNVT_temp_p
3	nviSetpoint	105	SNVT_temp_p

It does not matter if the data points are from the same device or from several devices or what network variable index they use. To make the example work with a real device you need to configure these data points with correct network variable indexes according to the device you use. One device that can be used that supports both application types is the Lonbox PZM4148 zone controller.

Simple Text Table View

In this exampel we only monitor values from the automation system. Values are read and updated dynamically by the script code every 2 seconds and update a HTML table with the values.

Data Point	Value
Light Level	80 %
Space Temperature	22.3 C
Temperature Setpoint	22.0 C

The HTML code has identifier attributes on the elements where the value is updated dynamically this id is used from the script code.

```
<table height="96">
  <tr style="background-color:lightgrey">
    <td width="152">Data Point</td>
    <td width="116">Value</td>
  </tr>
  <tr>
    <td>Light Level</td>
    <td id="light"></td>
  </tr>
  <tr>
    <td>Space Temperature</td>
    <td id="temp"></td>
  </tr>
  <tr>
    <td>Temperature Setpoint</td>
    <td id="setp"></td>
  </tr>
</table>
```

To implement this example two script functions are created directly in the HTML page. In addition two java script libraries available in the LonboxWebTools from Prolon Control System are included. The code below is all it takes to initiate and maintain the values on the page dynamically.

```
function OnUpdate(id, name, value)
{
   // Remove possible device prefix, this makes it possible to use
   // the same page for several locations/rooms. By creating the data
   // points with a location before '.' and then the same name after
   // the '.' the code would work for all data locations.
   var offset = name.indexOf(".");
   if (offset >= 0)
      name = name.slice(offset+1);

  // In this example we simply process the value based on it's name
  // an alternative method would be to match the name of the data point
  // with a name on the HTML page or on the graphical SVG page.
  if (name == "nvoLampValueFb")
     light.innerText = FormatSwitchLevel(value);
  else if (name == "nvoSpaceTemp")
     temp.innerText = FormatTempP(value);
  else if (name == "nviSetpoint")
     setp.innerText = FormatTempP(value);
}

function InitPage()
{
   // Start the automatic background update with 5 sec. interval
   // Data points 0,1,2,3 will be dynamically updated through
   // the specified "OnUpdate()" callback function.
   StartBackgroundUpdate("0-3", OnUpdate);

   // Optional, change update interval to 2 sec.
   SetBackgroundUpdateInterval(2000);
}
```

First function "InitPage()" initializes the background update system, this function is automatically called when the page is loaded by placing an "onload" attribute on the HTML body element. To show the optional change of dynamic load interval a second call adjusting the load interval from the default 5 seconds to 2 seconds.

The second function "OnUpdate" is now automatically called by the library routines when an updated value for a data point is ready. Tree parameters are passed to the function by the library; these give the data point id, the data point name and the actual value. The value is always returned in "packed hex" format. When the value is received the function formats the value and uses the DOM objects to insert the updated value into the HTML table.

Simple Text Table View with control buttons

In this example we monitor values from the automation system and we are able to set values in the automation system. Values are read and updated dynamically by the script code every 2 seconds. The HTML control buttons are shown here as [ ] but is normal buttons on the page.

Data Point	Value	Control
Light Level	80 %	[ON] [OFF]
Space Temperature	22,3 C	
Temperature Setpoint	22,0 C	[UP] [DOWN]

The HTML code we have added a few buttons, the buttons for light control writes directly to the data point while the temperature setpoint adjustment is relative and uses a local variable to store the current value and make a relative adjustment before sending the update.

```
<input name="On" value="On" style="width:70px" type="button"
onclick="DataPointWrite(0, 'C801');">

<input name="Up" value="Up" style="width:70px" type="button"
onclick="AdjustSetpoint(-500);">
```

A few lines of code has been added to the previous example: In the OnUpdate() function the current value of the temperature setpoint is saved in the “currentSetpoint” variable. The AdjustSetpoint() function for relative setpoint adjustment method will change this setpoint and write it back to the device using the DataPointWrite function.

```
var currentSetpoint;

function AdjustSetpoint(adjustment)
{
  // If we know what the setpoint is now
  if (currentSetpoint)
  {
    // Convert existing packed hex format to a number
    // Apply the adjustment
    // Convert it back to packed hex and write the data
    var newValue = NumberFromHex(value);
    newValue += adjustment;
    DataPointWrite(3, NumberToHex(newValue));
  }
}

function OnUpdate(id, name, value)
{
   // Remove possible device prefix, this makes it possible to use
   // the same page for several locations/rooms. By creating the data
   // points with a location before '.' and then the same name after
   // the '.' the code would work for all data locations.
   var offset = name.indexOf(".");
   if (offset >= 0)
      name = name.slice(offset+1);

  // In this example we simply process the value based on it's name
  // an alternative method would be to match the name of the data
  // point with a name on the HTML page or on the graphical SVG page.
  if (name == "nvoLampValueFb")
     light.innerText = FormatSwitchLevel(value);
  else if (name == "nvoSpaceTemp")
     temp.innerText = FormatTempP(value);
  else if (name == "nviSetpoint")
  {
     // We save the current setpoint value in a variable so that we
     // can adjust it.
     currentSetpoint = value;
     setp.innerText = FormatTempP(value);
  }
}

function InitPage()
{
   // Start the automatic background update with 5 sec. interval
   // Data points 0,1,2,3 will be dynamically updated through
   // the specified "OnUpdate()" callback function.
   StartBackgroundUpdate("0-3", OnUpdate);

   // Optional, change update interval to 2 sec.
   SetBackgroundUpdateInterval(2000);
}
```

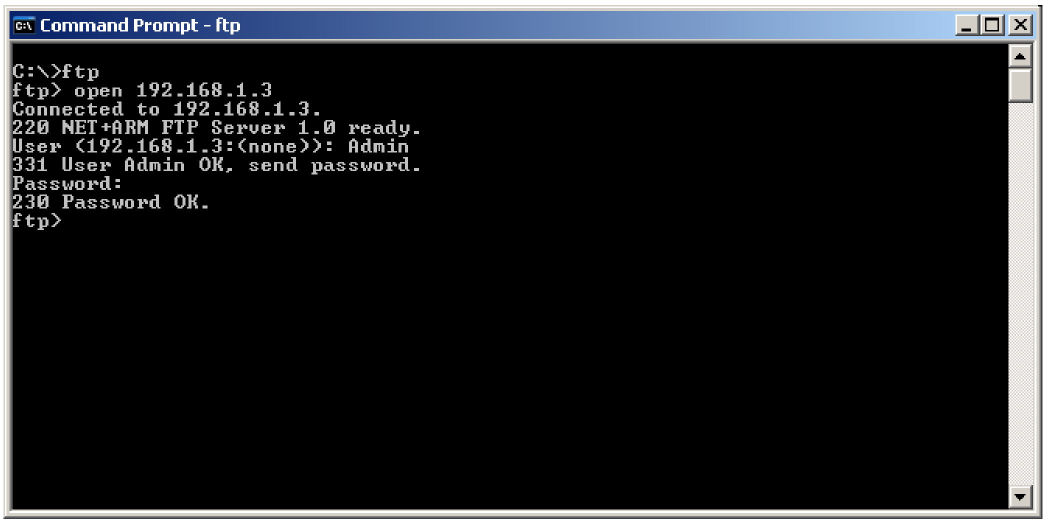
FTP Functions

The PID4000 WEB Server and data logger has a build-in FTP server. The server is used to up-date the firmware, back-up and restore the configuration, to transfer WEB sites and to send commands to the to the WEB Server.

Connection to the FTP server

To connect to the FTP server you can to use the build-in FTP client in the MS Windows operating systems, as WIN98, Windows 2000 and Windows XP. You need to be connected either via the Ethernet connector or have a PPP connection on a dial-up connection, PSTN, GSM or GPRS.

The PID4000 FTP Server do not support passive mode and advanced list commands.



```
C:\>ftp
ftp> open 192.168.1.3
Connected to 192.168.1.3.
220 NET+ARM FTP Server 1.0 ready.
User (192.168.1.3:(none)): Admin
331 User Admin OK, send password.
Password:
230 Password OK.
ftp>
```

The MS FTP client is used in the command prompt window and started using the following command:

FTP

Hereafter you need to connect to a specific data logger using the IP address, with the following FTP client command:

OPEN xxx.xxx.xxx.xxx

The *xxx.xxx.xxx.xxx*. represent an IP Address for a data logger. Hereafter the server request a username and a password. These account parameter can be changed from the PID4000 WEB pages.

The default user is:

Admin

And the password is:

admin

For more details about the FTP client in Windows, please consult the MS Windows manuals or help files.

Firmware update

To update the firmware in the PID4000 data logger you use the following FTP client commands

When first connected with the FTP client you are located in the FLASH0 partition. The firmware binary image file *image.bin* shall be put into the partition named RAM0. To change the partition, use the *CD* command with the following option:

“cd \RAM0”

With the *PUT* command you transfer a file from your local disk to the PID4000 WEB Server unit.

“put image.bin”

When this command completes, write “bye” to disconnect and wait for the PID4000 WEB Server to restart.

Upload your own WEB site

To upload your own WEB site to the PID4000 WEB Server, you use the following FTP client commands.

When first connected with the FTP client you are located in the FLASH0 partition. The WEB site files shall be located in the directory “/web/”, “/web/user/”, “/web/service/” and “/web/admin/” on the partition FLASH0. Change directory with the following command and option.

“cd \web”

With the *PUT* command you transfer the files from your local disk to the PID4000 WEB Server unit.

“put index.htm”

To see your new WEB site from the PID4000 WEB Server, use the following HTTP command in your WEB browser.

http://192.168.1.100/FS/FLASH0/web/index.htm

Electrical Specifications

Technical data:	PID4000
LonWorks® topology:	TP/FT-10
Transceiver:	FT-X1
Neuron-Chip:	FT3120E4, 10MHz
Main processor:	ARM 32-bit RISC
Flash storage:	4/8 Mbyte Flash (hardware rev. B has 8 MB) - 1 MB for application - 2 MB for datalog storage - 2 / 4 MB flash disk
Memory:	16 Mbyte SDRAM
Data Channels:	200
Ethernet:	RJ45 10/100 Mbit TCP/IP protocol
Serial Port:	RS232 Terminal and PPP protocol
Control Indicators:	Neuron state function, modem status, I/O status, Link and 100Mb in Yellow. Green Supply
Service Switch:	Neuron service Pin
Supply:	12 – 30Vdc 9 – 24Vac
Consumption	Max. with modem, 350 mA @ 24Vdc 300 mA @ 24Vac
EMC immunity:	IEC 61000-6-2
EMC emission:	EN 50081-2
Safety:	EN 60950
Operating temp.:	0 til 45°C
Storage temp.:	-20 til 70°C
Size:	L x H x W 157 x 58 x 90+con. mm
Weight	261g
Models:	
PID4000	Standard data logger
PID4000-GSM	Build-In GSM/GPRS modem
PID4025	Build-in utility metering module (I/O)
PID4025-GSM	Build-in GSM/GPRS and utility metering module (I/O)