INSTRUCTION MANUAL

PQF-Manager communication

How to use RS485 – USB – Ethernet connections

Table of contents

1 Introduction to this manual

1.1 Intended audience

This manual is intended for programmers, commissioning people, supervision people who need to start communication, access data, and to develop supervision software which will interact with the PQF-Manager and PQF active filters.

1.2 Before you start

This manual describes the PQF-Manager data table. These data are accessible through RS485, USB or Ethernet using Modbus RTU and TCP protocols or PQ-Link software.

All information available from the keyboard of the PQF-Manager will be available through the data table. Addresses, access levels and protocol information are of concerns.

To be able to access data of the PQF-Manager and PQF active filters consistently, a basic knowledge of these products is needed. Functionality of the PQF-Manager, meaning of various measurements and logging of data are some particular aspects that should be familiar. Look in the PQF-Manager and PQF operating manuals to know more about that.

1.3 How to use this manual

Chapter 2 gives details concerning the Modbus protocol.

Chapter 3 describes Modbus functions and how Modbus is implemented in the controller.

Chapter 4 describes USB / TCP/IP protocol and how it is implemented in the controller.

Chapter 5 contains the table reference and formats to access measurements / settings data.

Chapter 6 describes the Windows DLL to handle USB / TCP/IP requests in a user specific application.

1.4 Software protocols and physical interface

The PQF-Manager supports three communication protocols:

Modbus RTU and Modbus TCP (Chapter 2 and 3) and PQ-Link protocol (Chapter 6).

Three physical connections are available:

RS485 with the Modbus Adapter option (Chapter 2), USB (Chapter 4), Ethernet-RJ45

(Chapter 4).

The table below resumes the availability of communication protocols depending on the connection provided.

	Available connection		
Communication protocol	RS485 Modbus Adapter (option)	USB	Ethernet-RJ45
	Modbus RTU	PQ-Link	PQ-Link and Modbus TCP

2 Modbus protocol overview

2.1 Overview

MODBUS RTU

MODBUS RTU is a non-proprietary serial communications protocol that is widely used in the process control industry. The protocol was developed by Modicon for PLC communications and later released for public use.

This protocol is available in all major Human Machine Interface (HMI) software packages and terminals. Many of the major controller and PLC manufacturers also offer MODBUS protocol as a standard or optional protocol in their instrumentation.

The hardware over which MODBUS RTU communications are performed is not defined by the protocol. MODBUS RTU is supported on RS-232, RS-422, RS-485, Ethernet and other electrical standards. It should be noted that MODBUS RTU, MODBUS ASCII and MODBUS Plus are unique communication formats, and are not compatible with each other. This document will discuss MODBUS RTU only.

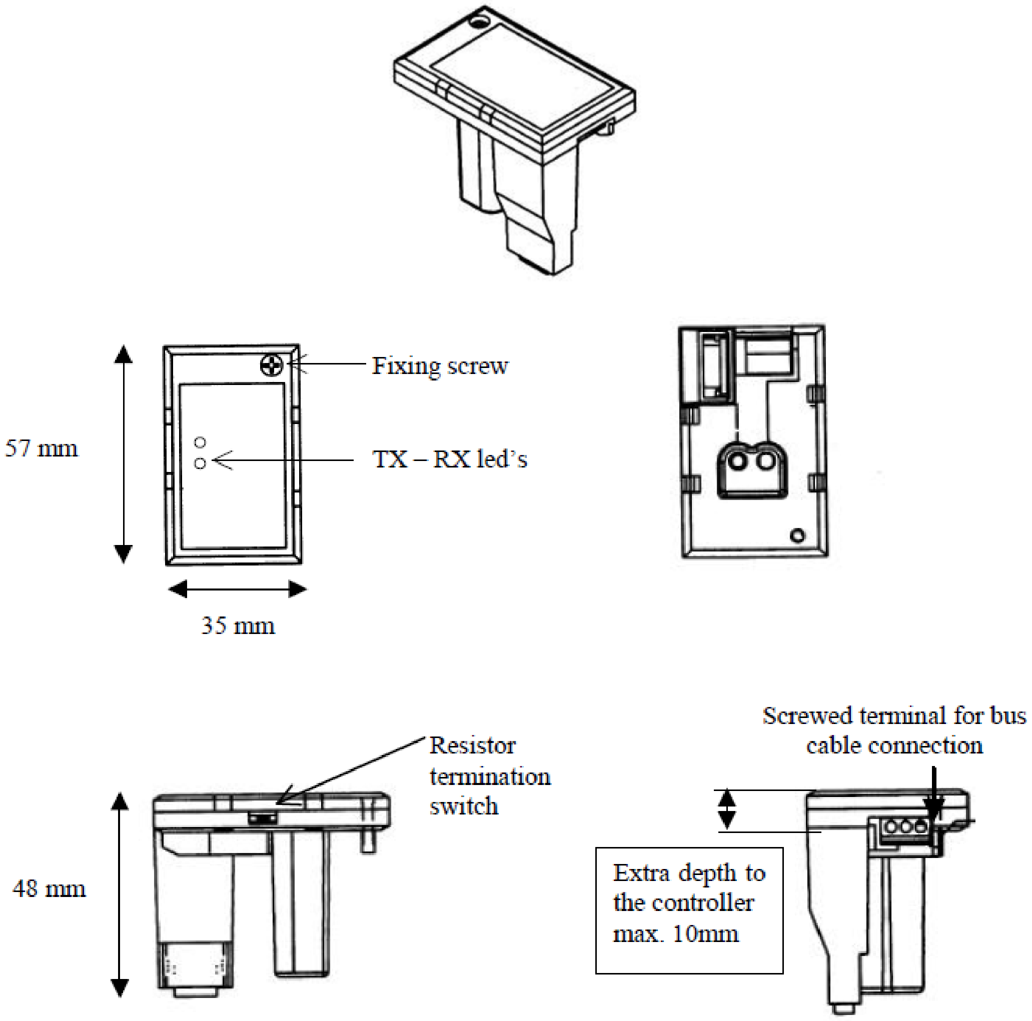
MODBUS TCP

MODBUS TCP is a MODBUS RTU message transmitted with a TCP/IP wrapper and sent over an Ethernet network instead of serial lines. The Server does not have a SlaveID as in RTU since it uses an IP Address instead.

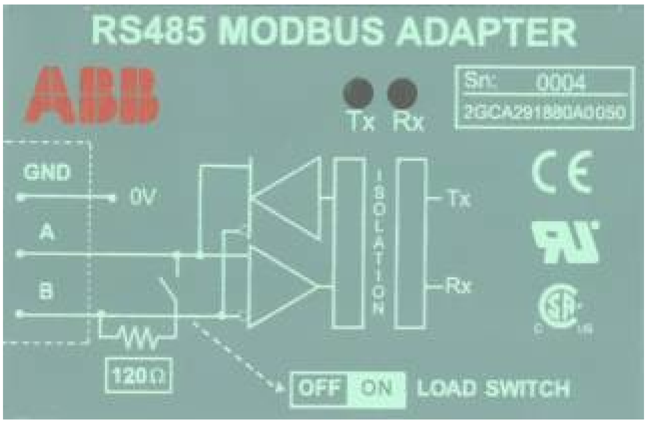
2.2 RS485 Modbus Adapter

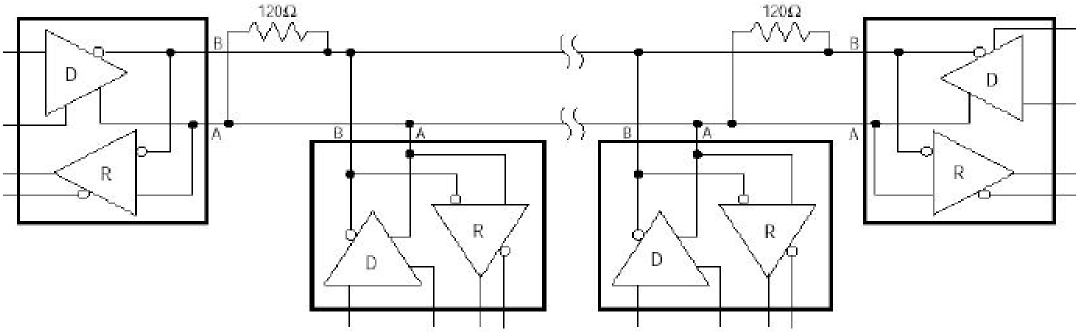
The Modbus protocol communicates with the instrumentation by means of an industry standard serial interface. This interface may be RS-232, RS-422 or RS-485. Some systems may also support the protocol over other busses or networks, such as Ethernet. An RS-232 interface allows only two devices to be connected together. RS-422 supports 1 driver and up to 10 receivers on a single network. For bi-directional communications, special tri-state circuitry is provided. RS-485 supports up to 32 driver/receiver pairs. With special hardware, the RS-422 and RS-485 limits can be expanded to allow as many as 248 devices on a single network. Each device on a network must have a unique address, which may be soft configured. Address zero is reserved for broadcast messages from the host to all slaves.

The RS485 Modbus Adapter is an option to the PQF-Manager. It enables the connection of the PQF-Manager controller to an RS485 Modbus network.


Fixing screw
57 mm
TX – RX led's
35 mm
Resistor termination switch
48 mm
Screwed terminal for bus cable connection
Extra depth to the controller max. 10mm

CAUTION: Be careful that the RS485 MODBUS ADAPTER is the one with a GREEN text colour (3.3V power supply). The one with a WHITE text colour is reserved for the old model (5V power supply).





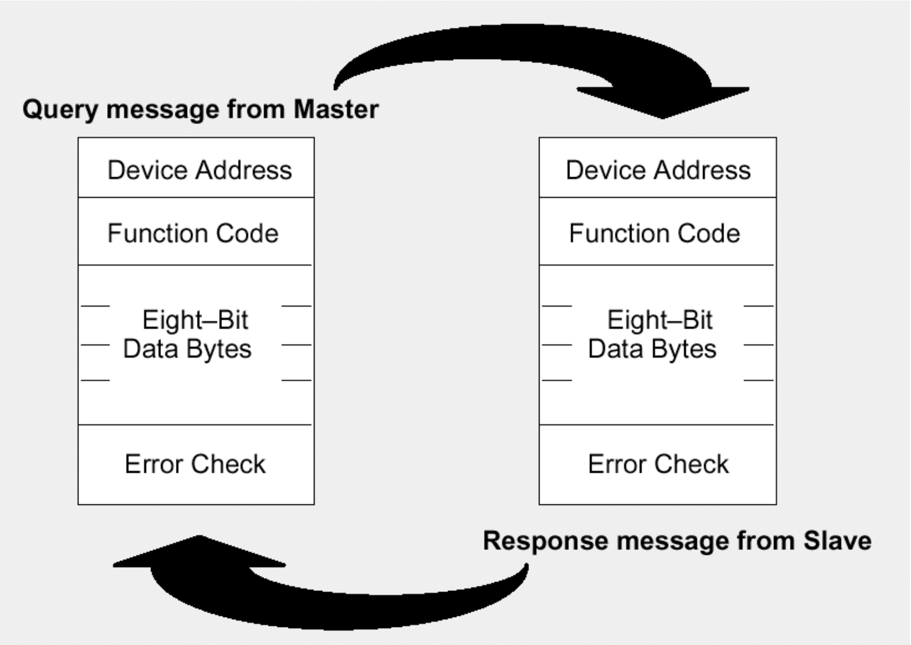
2.3 Transactions on Modbus Networks

Modbus protocol uses a master–slave technique, in which only one device (the master) can initiate transactions (called ‘queries’). The other devices (the slaves) respond by supplying the requested data to the master, or by taking the action requested in the query. Typical master devices include host processors and programming panels. Typical slaves include programmable controllers.

The master can address individual slaves, or can initiate a broadcast message to all slaves.

Slaves return a message (called a ‘response’) to queries that are addressed to them individually. Responses are not returned to broadcast queries from the master.

The Modbus protocol establishes the format for the master’s query by placing into it the device (or broadcast) address, a function code defining the requested action, any data to be sent, and an error–checking field. The slave’s response message is also constructed using Modbus protocol. It contains fields confirming the action taken, any data to be returned, and an error–checking field. If an error occurred in receipt of the message, or if the slave is unable to perform the requested action, the slave will construct an error message and send it as its response.



The Query:

The function code in the query tells the addressed slave device what kind of action to perform. The data bytes contain any additional information that the slave will need to perform the function.

The data field must contain the information telling the slave which register to start at and how many registers to read.

The error check field provides a method for the slave to validate the integrity of the message contents.

The Response:

If the slave makes a normal response, the function code in the response is an echo of the function code in the query. The data bytes contain the data collected by the slave, such as register values or status. If an error occurs, the function code is modified to indicate that the response is an error response, and the data bytes contain a code that describes the error.

The error check field allows the master to confirm that the message contents are valid.

2.4 Serial Transmission Mode

The transmission mode defines the bit contents of message fields transmitted serially on the networks. It determines how information will be packed into the message fields and decoded.

Modbus defines two transmission modes: ASCII or RTU.

Only RTU mode will be used here. The mode and serial parameters must be the same for all devices on a Modbus network.

RTU Mode

The main advantage of this mode is that its greater character density allows better data throughput than ASCII for the same baud rate.

Each message must be transmitted in a continuous stream.

The format for each byte in RTU mode is:

Bits per Byte:

1 start bit

8 data bits, least significant bit sent first

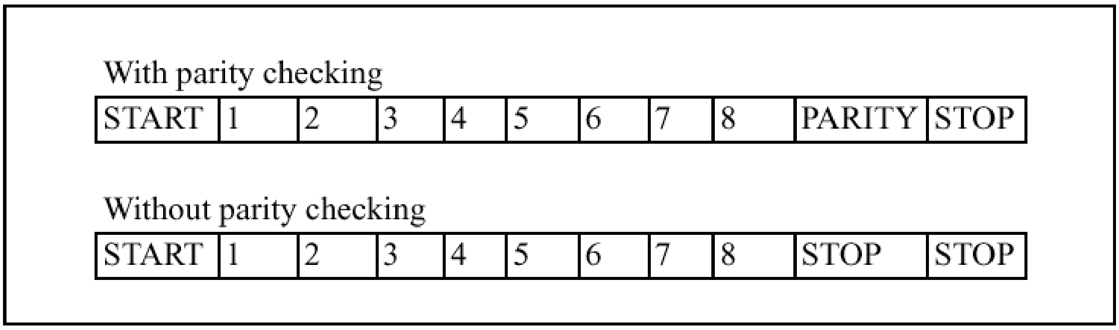
1 bit for even/odd parity; no bit for no parity

1 stop bit if parity is used; 2 bits if no parity

Error Check Field: Cyclical Redundancy Check (CRC)

The messages are transmitted in the network from left to right, i.e. the Least

Significant Bit (LSB) first and the Most Significant Bit (MSB) last.



With parity checking

START	1	2	3	4	5	6	7	8	PARITY	STOP

Without parity checking

START	1	2	3	4	5	6	7	8	STOP	STOP

Description of the bit sequence for the RTU mode

2.5 Modbus Message Framing

A Modbus message is placed by the transmitting device into a frame that has a known beginning and ending point. This allows receiving devices to begin at the start of the message, read the address portion and determine which device is, and to know when the message is completed.

Partial messages can be detected and errors can be set as a result.

Modbus RTU Framing

In RTU mode, messages start with a silent interval of at least 3.5 character times.

This is most easily implemented as a multiple of character times at the baud rate that is being used on the network (shown as T1–T2–T3–T4 in the figure below).

Another factor to consider is that each device has its own response time. This response time can be anywhere from a few milliseconds to a few hundred milliseconds. The Host must be configured to allow adequate time for the slowest device to respond.

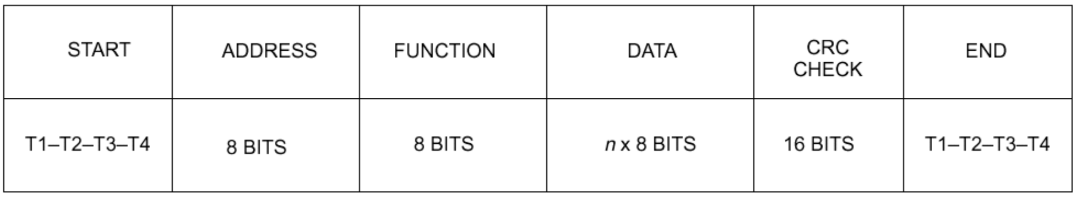
The first field then transmitted is the device address.

Networked devices monitor the network bus continuously, including during the ‘silent’ intervals. When the first field (the address field) is received, each device decodes it to find out if it is the addressed device.

Following the last transmitted character, a similar interval of at least 3.5 character times marks the end of the message. A new message can begin after this interval.

The entire message frame must be transmitted as a continuous stream. If a silent interval of more than 1.5 character times occurs before completion of the frame, the receiving device flushes the incomplete message and assumes that the next byte will be the address field of a new message.

Similarly, if a new message begins earlier than 3.5 character times following a previous message, the receiving device will consider it is a continuation of the previous message. This will set an error, as the value in the final CRC field will not be valid for the combined messages. A typical message frame is shown below.



START	ADDRESS	FUNCTION	DATA	CRC CHECK	END
T1–T2–T3–T4	8 BITS	8 BITS	*n* x 8 BITS	16 BITS	T1–T2–T3–T4

For a complete description of the Modbus protocol, please look at the Modicon Modbus Protocol Reference Guide (PI–MBUS–300 Rev. J).

Modbus TCP Framing

Modbus TCP/IP (also Modbus-TCP) is simply the Modbus RTU protocol with a TCP interface that runs on Ethernet.

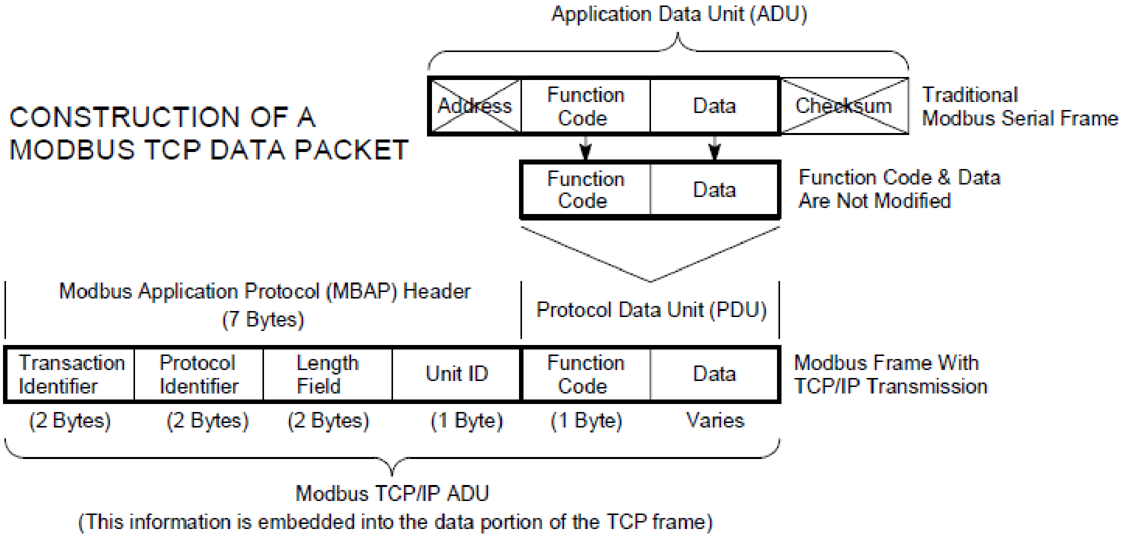
The Modbus messaging structure is the application protocol that defines the rules for organizing and interpreting the data independent of the data transmission medium.

TCP/IP refers to the Transmission Control Protocol and Internet Protocol, which provides the transmission medium for Modbus TCP/IP messaging.

Simply stated, TCP/IP allows blocks of binary data to be exchanged between computers. It is also a world-wide standard that serves as the foundation for the World Wide Web. The primary function of TCP is to ensure that all packets of data are received correctly, while IP makes sure that messages are correctly addressed and routed. Note that the TCP/IP combination is merely a transport protocol, and does not define what the data means or how the data is to be interpreted (this is the job of the application protocol, Modbus in this case).

So in summary, Modbus TCP/IP uses TCP/IP and Ethernet to carry the data of the Modbus message structure between compatible devices. That is, Modbus TCP/IP combines a physical network (Ethernet), with a networking standard (TCP/IP), and a standard method of representing data (Modbus as the application protocol). Essentially, the Modbus TCP/IP message is simply a Modbus communication encapsulated in an Ethernet TCP/IP wrapper.

In practice, Modbus TCP embeds a standard Modbus data frame into a TCP frame, without the Modbus checksum, as shown in the following diagram.


Application Data Unit (ADU)
CONSTRUCTION OF A MODBUS TCP DATA PACKET
Address
Function Code
Data
Checksum
Traditional Modbus Serial Frame
Function Code
Data
Function Code & Data Are Not Modified
Modbus Application Protocol (MBAP) Header (7 Bytes)
Protocol Data Unit (PDU)
Transaction Identifier
Protocol Identifier
Length Field
Unit ID
Function Code
Data
Modbus Frame With TCP/IP Transmission
(2 Bytes)
(2 Bytes)
(2 Bytes)
(1 Byte)
(1 Byte)
Varies
Modbus TCP/IP ADU
(This information is embedded into the data portion of the TCP frame)

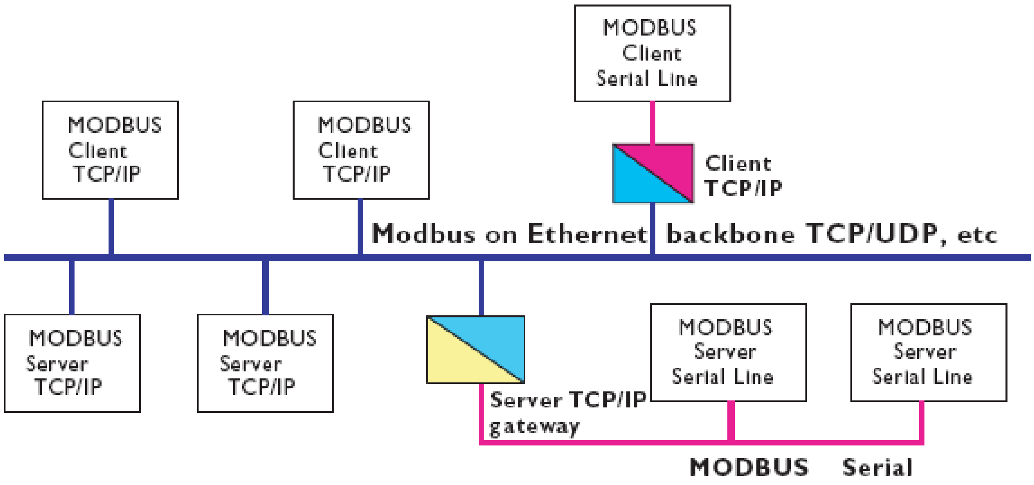
The complete Modbus TCP/IP Application Data Unit is embedded into the data field of a standard TCP frame and sent via TCP to well-known system port 502, which is specifically reserved for Modbus applications. Modbus TCP/IP clients and servers listen and receive Modbus data via port 502.

Modbus TCP must establish a connection before transferring data, since it is a connection-based protocol. The Master (or Client in Modbus TCP) establishes a connection with the Slave (or Server). The Server waits for an incoming connection from the Client. Once a connection is established, the Server then responds to the queries from the Client until the client closes the connection.

The number of Clients connected to 1 specific RVT is limited to 5.

In summary:

- Modbus TCP allows the user to connect to a RVT12-3P through Ethernet or Internet using Modbus standard protocol (with HMI, SCADA…)
- The slave address of Modbus RTU specification is replaced by the IP address via TCP port 502.
- Multiple Clients may access multiple RVT Servers.



3 Modbus function codes

3.1 Data Addresses in Modbus Messages

Modbus defines 4 address spaces: 2 address spaces for bit addressable data and 2 address spaces for 16 bits addressable data.

Address space	Data	Readable/writable	Modbus name	Address space
0XXXX	Output bit	Read & write	Coil status	0XXXX
1XXXX	Input bit	Read	Input status	1XXXX
3XXXX	Input word	Read	Input register	3XXXX
4XXXX	Output word	Read & write	Holding register	4XXXX

Input register address space will be mainly used for measurements.

Holding register address space will contain settings.

All data addresses in Modbus messages are referenced to zero.

For example:

The coil known as 'coil 1' in a programmable controller is addressed as coil 0000 in the data address field of a Modbus message.

Coil 127 decimal is addressed as coil 007E hex (126 decimal).

Holding register 40001 is addressed as register 0000 in the data address field of the message.

The function code field already specifies a 'holding register' operation. Therefore the '4XXXX' reference is implicit.

Holding register 40108 is addressed as register 006B hex (107 decimal).

3.2 Supported function codes

The following table gives the Modbus functions which are implemented and supported.

The code is the one used in function field of the Modbus message.

The address space concerned and the purpose of the function are given below.

Code	Function	Address range/remark
1	Read coil status	0XXXX reads the on/off status of discrete outputs
2	Read input status	1XXXX reads the on/off status of discrete inputs
3	Read holding registers	4XXXX reads contents of output registers
4	Read input registers	3XXXX reads contents of input registers
5	Force single coil	0XXXX sets the status of a discrete output
6	Preset single register	4XXXX sets the value of a holding register
7	Read exception status	Device specific
8	Diagnostics	Checks the communication system between the master and the slave
11	Fetch comm. event ctr.	Returns the amount of successful read/write operations on data points
12	Fetch comm. event log	Returns log registers of communication events
15	Force multiple coils	0XXXX sets the status of multiple discrete outputs
16	Preset multiple registers	4XXXX sets the value of multiple holding registers
17	Report slave ID	Device specific
22	Mask write 4X registers	4XXXX and/or write of a holding register
23	Read/write 4X registers	4XXXX reads a set of holding registers and writes a set of holding registers in one query

Remark: please note that for security reasons broadcast is not supported by the PQF-Manager.

3.3 Master's queries and Slave's responses

When a master device sends a query to a slave device it expects a normal response. One of four possible events can occur from the master's query:

- If the slave device receives the query without a communication error, and can handle the query normally, it returns a normal response.
- If the slave does not receive the query due to a communication error, no response is returned. The master program will eventually process a timeout condition for the query.
- If the slave receives the query, but detects a communication error (parity or CRC), no response is returned. The master program will eventually process a timeout condition for the query.
- If the slave receives the query without a communication error, but cannot handle it (for example, if the request is to read a non–existent coil or register), the slave will return an exception response informing the master of the nature of the error.

3.4 Reads and writes to Modbus addresses (functions 1, 2, 3, 4, 5, 6, 15, 16, 22, 23)

The format of a read function (read coil status (01), read input status (02), read input registers (04), read holding registers (03)) is as follows:

QUERY		RESPONSE	
Slave address	1 byte	Slave address	1 byte (echo of master's query)
Function	1 byte	Function	1 byte (echo of master's query)
Starting data address	2 bytes	Byte count	1 byte
Quantity of points	2 bytes	Data values	N bytes
Error check field CRC	2 bytes	Error check field CRC	2 bytes

The format of a force single coil (05) or a preset single register (06) function is as follows:

QUERY		RESPONSE	
Slave address	1 byte	Slave address	1 byte (echo of master's query)
Function	1 byte	Function	1 byte (echo of master's query)
Data address	2 bytes	Data address	2 bytes
Data value	2 bytes	Data value	2 bytes
Error check field CRC	2 bytes	Error check field CRC	2 bytes

The format of a force multiple coil (15) or a preset multiple registers (16) function is as follows:

QUERY		RESPONSE	
Slave address	1 byte	Slave address	1 byte (echo of master's query)
Function	1 byte	Function	1 byte (echo of master's query)
Data address	2 bytes	Data address	2 bytes
Quantity of points	2 bytes	Quantity of points	2 bytes
Byte count	1 byte	Error check field CRC	2 bytes
Data values	N bytes		
Error check field CRC	2 bytes		

The format of a read/write multiple registers (23) function is as follows:

QUERY		RESPONSE	
Slave address	1 byte	Slave address	1 byte (echo of master's query)
Function	1 byte	Function	1 byte (echo of master's query)
Read data address	2 bytes	Byte count	1 byte
Read quantity of points	2 bytes	Data values	N bytes
Write data address	2 bytes	Error check field CRC	2 bytes
Write quantity of points	2 bytes		
Byte count	1 byte		
Write data values	N bytes		
Error check field CRC	2 bytes		

The format of a Mask/write register (22) function is as follows:

QUERY		RESPONSE	
Slave address	1 byte	Slave address	1 byte (echo of master's query)
Function	1 byte	Function	1 byte (echo of master's query)
Data address	2 bytes	Data address	2 bytes
And mask	2 bytes	And mask	2 bytes
Or mask	2 bytes	Or mask	2 bytes
Error check field CRC	2 bytes	Error check field CRC	2 bytes

3.5 Fetch comm event counter (function 11)

The controller’s event counter is incremented once for each successful message completion. It is not incremented for exception responses, poll commands, or fetch event counter commands. It returns amount of successful read/write operations on data points.

The format of a Fetch comm event counter (11) function query is as follows:

QUERY	
Slave address	1 byte
Function	1 byte
Error check field CRC	2 bytes

RESPONSE	
Slave address	1 byte (echo of master's query)
Function	1 byte (echo of master's query)
Status word	2 bytes (0)
Event counter	2 bytes
Error check field CRC	2 bytes

3.6 Fetch comm event log (function 12)

Returns a status word, the comm event counter (see function 11) , the bus message counter (see function 08 subfunction 11), and a field of event bytes from the slave.

The format of a Fetch comm event log (12) function query is as follows:

QUERY	
Slave address	1 byte
Function	1 byte
Error check field CRC	2 bytes

RESPONSE	
Slave address	1 byte (echo of master's query)
Function	1 byte (echo of master's query)
Byte count	1 byte
Status word	2 bytes (0)
Event counter	2 bytes
Bus message counter	2 bytes
Event log buffer	N bytes
Error check field CRC	2 bytes

The 64 bytes wide Event log buffer is filled with communication events. The most recent communications event is shown in the Event 0 byte.

Event bytes are stored in the Even log buffer for 4 different reasons.

The bit will be set to logic ‘1’ if the corresponding condition is TRUE.

Slave Modbus Receive Event

This type of event byte is stored by the slave when a query message is received.

It is stored before the slave processes the message.

Bit	Contents
0	Not Used
1	Communications Error
2	Not Used
3	Not Used

4	Character Overrun
5	Currently in Listen Only Mode
6	Broadcast Received
7	1

Slave Modbus Send Event

This type of event byte is stored by the slave when it finishes processing a query message.

It is stored if the slave returned a normal or exception response, or no response.

Bit	Contents
0	Read Exception Sent (Exception Codes 1-3)
1	Slave Abort Exception Sent (Exception Code 4)
2	Not used
3	Not used
4	Write Timeout Error Occurred
5	Currently in Listen Only Mode
6	1
7	0

Slave Entered Listen Only Mode

This type of event byte is stored by the slave when it enters the Listen Only Mode.

The event is defined by a content of ‘04’ hex.

Slave Initiated Communication Restart

This type of event byte is stored by the slave when its communications port is restarted. The slave can be restarted by the Diagnostics function (code 08), with subfunction Restart Communications Option (code 01).

The event is defined by a contents of ‘00’ hex.

3.7 Diagnostics function and subfunctions (function 8)

The format of a diagnostics (08) function query is as follows:

QUERY	
Slave address	1 byte
Function	1 byte
Subfunction	2 bytes
Data field	2 bytes
Error check field CRC	2 bytes

The format of a response to a diagnostics function query is an echo of the query itself.

If the request is directed to a counter, however, the slave returns the counter's value in the data field.

00 Return Query Data

The data in the query data field is to be returned (looped back) in the response. The entire response should be identical to the query.

01 Restart Communication Option

The slave's peripheral port is to be initialized and restarted, and all of its communication event counters are to be cleared. If the port is currently in the Listen Only Mode, no response will be sent. If the port is not currently in the Listen Only Mode, a normal response will be sent. This occurs before the restart is executed.

02 Return Diagnostic Register (Not supported)

03 (Not supported)

04 Force Listen Only Mode

Forces the addressed slave to enter the Listen Only Mode for Modbus communications.

10 Clear Counters and Diagnostic Register

Clears all counters and the diagnostic register.

11 Return Bus Message Count

The response data field returns the total quantity of messages that the slave has detected in the communications system since its last restart, clear counters operation, or power-up.

12 Return Bus Communication Error Count

The response data field returns the quantity of CRC errors encountered by the slave since its last restart, clear counters operation, or power-up.

13 Return Bus Exception Error Count

The response data field returns the quantity of Modbus exception responses returned by the slave since its last restart, clear counters operation, or power-up.

14 Return Slave Message Count

The response data field returns the quantity of messages addressed to the slave, or broadcast that the slave has processed since its last restart, clear counters operation, or power-up.

15 Return Slave No Response Count

The response data field returns the quantity of messages addressed to the slave for which it sent no response (neither a normal response nor an exception response) since its last restart, clear counters operation, or power-up.

16 Return Slave NACK Response Count (Not supported)

17 Return Slave Busy Response Count (Not supported)

18 Return Bus Character Overrun Count

The response data field returns the quantity of messages addressed to the slave that it could not handle due to a character overrun condition since its last restart, clear counters operation, or power-up

19 (Not supported)

20 (Not supported)

21 (Not supported)

Diagnostic counters

Bus Message Counter

The total number of messages that the slave device has detected in the communications system since its last restart, clear counters operation, or power-up.

Bus Communication Error Counter

The number of CRC or LRC errors encountered by the slave device since its last restart, clear counters operation, or power-up.

Bus Exception Error Counter

The number of Modbus exception responses sent by the slave device since its last restart, clear counters operation, or power-up.

Slave Message Counter

The number of messages addressed to the slave device or broadcast that the slave device has processed since its last restart, clear counters operation, or power-up.

Slave No Response Counter

The number of messages addressed to the slave device for which it sent no response (neither a normal response nor an exception response) since its last restart, clear counters operation, or power-up.

Bus Character Overrun Counter

The number of messages addressed to the slave device that it could not handle due to a character overrun condition since its last restart, clear counters operation, or power-up .

3.8 Exception responses

Exception responses are sent when the slave device cannot handle the query. The format of an exception response to a master's query is as follows:

01 ILLEGAL FUNCTION

The function code received in the query is not an allowable action for the slave device (see paragraph 2.2).

02 ILLEGAL DATA ADDRESS	The data address or number of items received in the query is not allowable or correct for the slave device. The slave device will send this exception response if an attempt to read or write part of a multiple register database object is detected. Possible objects are time, strings and counters
03 ILLEGAL DATA VALUE	A value contained in the query data field is out of range. The contents of the register or the status of the coil has not changed (see paragraph 4.3).
04 SLAVE DEVICE ABORT	An unrecoverable error occurred while the slave was attempting to perform the requested action. This may happen when the access level for changing a parameter is not reached (see paragraph 4.2).
05 ACKNOWLEDGE	Not supported
06 SLAVE DEVICE BUSY	Not supported
07 NEGATIVE ACKNOWLEDGE	Not supported
08 MEMORY PARITY ERROR	Not supported

An application program in the master is responsible for handling exception responses. Typical processes include successive attempts to send a query, sending diagnostic messages to the slave, and notifying the operators.

3.9 CRC generation

The Cyclical Redundancy Check (CRC) field is two bytes, containing a 16–bit binary value. The CRC value is calculated by the transmitting device, which appends the CRC to the message. The receiving device recalculates a CRC during receipt of the message, and compares the calculated value to the actual value it received in the CRC field. If the two values are not equal, an error results.

Placing the CRC into the Message:

When the 16–bit CRC (two 8–bit bytes) is transmitted in the message, the low-order byte will be transmitted first, followed by the high-order byte.

Example: here is an example of calculating directly the CRC.

```
/*---------------------------------------------------------------
FUNCTION                 : This routine calculates the crc high and low byte of a message.
 ---------------------------------------------------------------
INPUT PARAMETERS     :        buf  ->  Array containing message to be sent to controller
                              start ->  Start of loop in crc counter, usually 0.
                              cnt  ->  Amount of bytes in message being sent to controller
 ---------------------------------------------------------------
```

```
OUTPUT                     :           temp  ->  Returns crc byte for message.
 --------------------------------------------------------------
*/
word crc(byte *buf,word start,word cnt)
{
word      i,j;
word      temp,temp2,flag;
  temp=0xFFFF;
  for (i=start; i<cnt; i++)
  {
                    temp=temp ^ buf[i];
                    for (j=1; j<=8; j++)
                    {
                             flag=temp & 0x0001;
                             temp=temp >> 1;
                             if (flag) temp=temp ^ 0xA001;
                    }
          }
          /* Reverse byte order. */
          temp2=temp >> 8;
          temp=(temp << 8) | temp2;
          temp &= 0xFFFF;
          return(temp);
  }
```

4 Ethernet / RJ45 and USB connections for PQ-Link protocol

4.1 General overview

The Data of the PQF-Manager can be accessed by different means:

- TCP/IP connection from a local client or from a remote client
- USB seen as a USB UART interface

The server will allow local and distant access to the PQF-Manager. Different access levels will be implemented to restrict certain functionality to given users. A login and password will therefore be required.

The format of the messages transferred via those two medium will be the same.

4.2 Physical layer

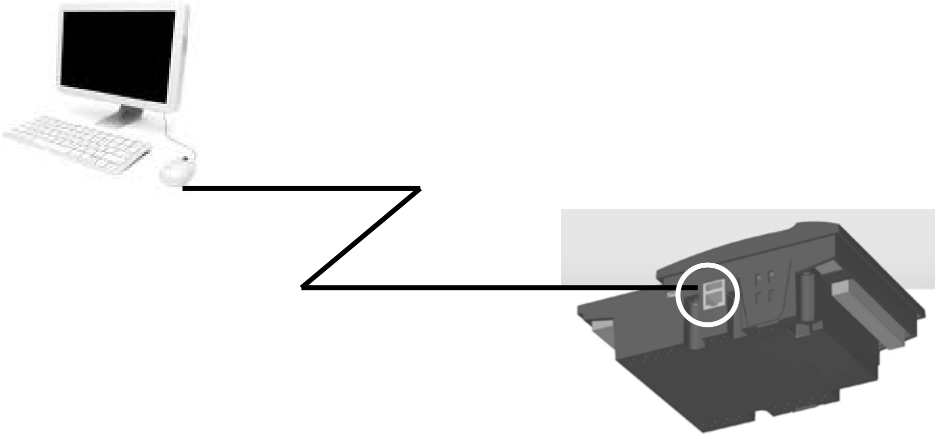
4.2.1 TCP/IP

TCP/IP connections can be indifferently initiated locally or remotely. As the local connection is used by the UI, it will have extended access rights to parameters compared to a remote connection.

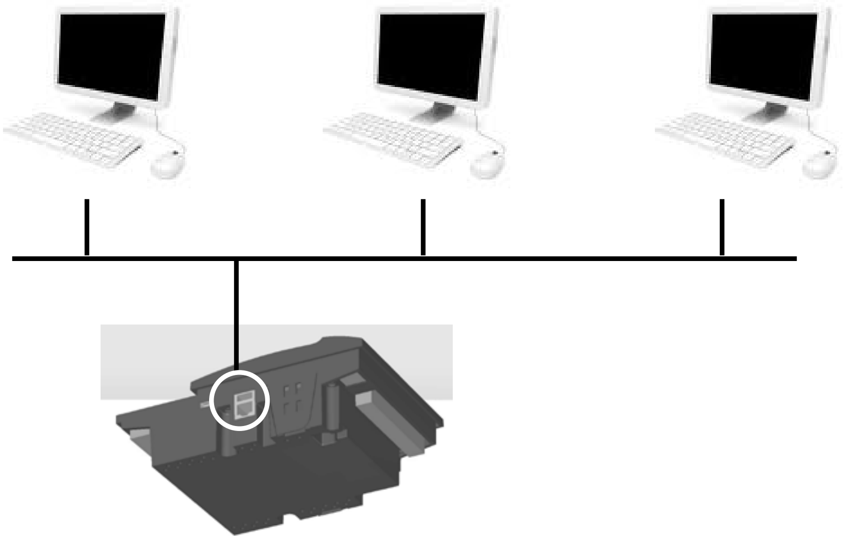
The TCP port used by default is 4250.

The maximum number of TCP/IP clients to the PQF-Manager is 2.

The connection to the PQF-Manager is an RJ45 Cat5e Ethernet cable



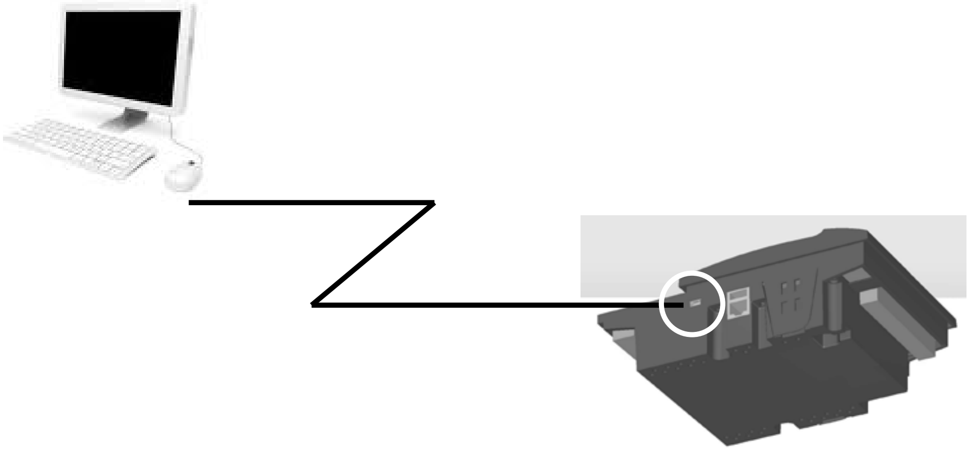
The PQF-Manager can be connected directly to a LAN or through Internet



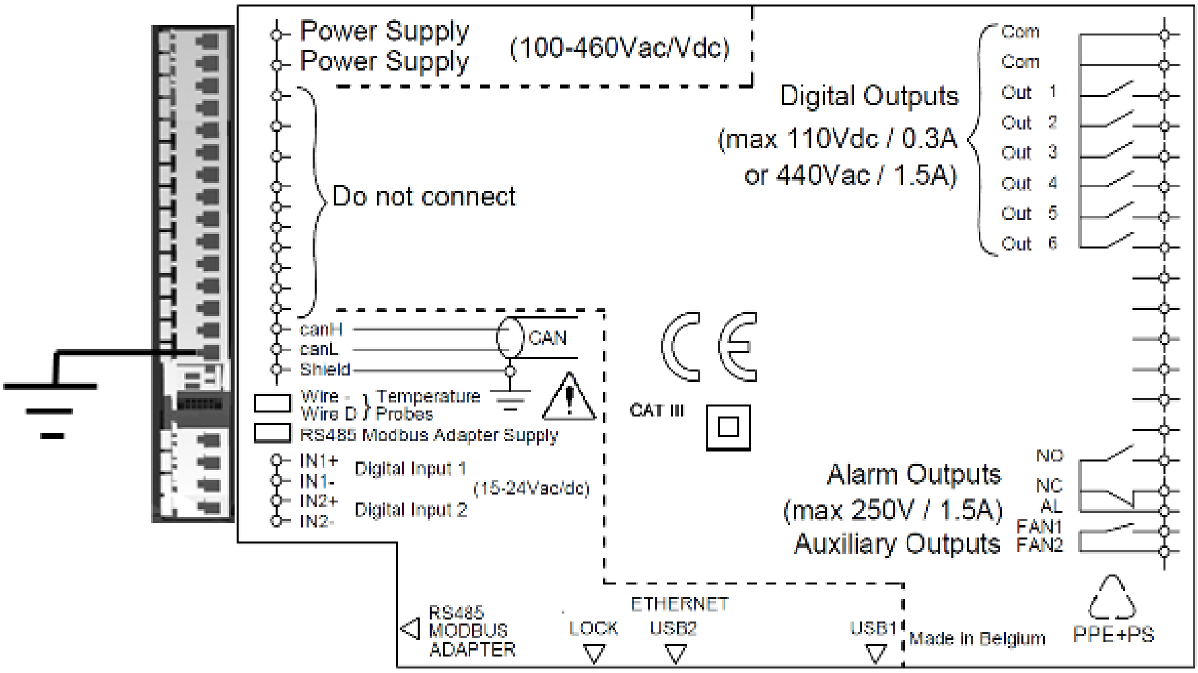
4.2.2 USB

The USB interface is used to present the PQF-Manager as a serial interface on its USB port.

The computer is connected through a USB-A male to USB-Mini B male.



Caution: The USB connection to the PQF-Manager is not isolated. It is mandatory to connect the protective EARTH connection when using the USB.



4.3 Framing layer & Command layer

The data can be accessed by different means:

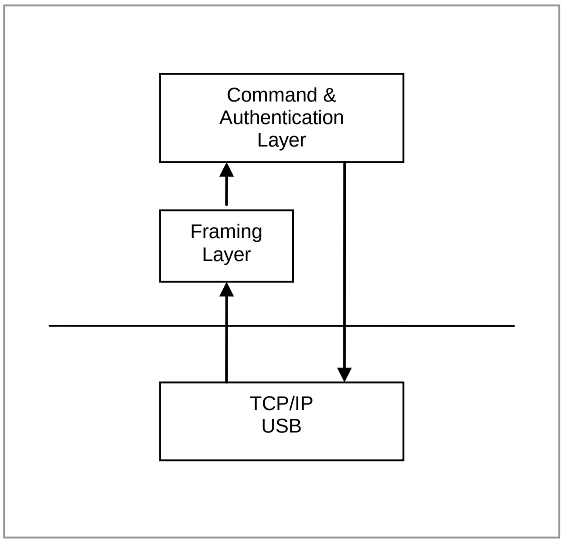
- TCP/IP connection from a local client or from a remote client
- USB seen as a USB UART interface

The server will allow local and distant access to the PQF-Manager. Different access levels will be implemented to restrict certain functionality to given users. A login and password will therefore be required.

The format of the messages transferred via those two medium will be the same.

Two layers will be put on top of them:

Framing Layer	This layer is taking care of receiving the frames. The byte stream is decoded and frames generated are passed to the layers above.
Command Layer	This is the upper layer taking care of the commands. It will also take care of the authentication of the client during the connection via some specific commands.



Those two layers will use the same format regardless of the actual “transmission medium” used.

The data is arranged in packets with integrated error checking.

The Windows DLL (chapter 6) incorporates all framing and command issues needed to communicate with the PQF-Manager.

5 Data table

5.1 Overview

There's quite an extensive set of parameters available in the PQF-Manager.

The individual parameters have been put together in groups to ease manipulation and transfers between the different layers of the application.

The parameters will not be individually accessible to the application. Only groups of parameters will be exchanged between the application and the lower layers.

This will allow the lower layer to be quite independent from the parameters contained within the group.

The parameter groups will be split in two types:

- Groups that are needed by the lower layers. They will have known and fixed group ID's in all different applications.
- Groups which are specific to an application. Their IDs will be known by the applicative layer. The lower-layers won't know the internals of these parameter groups.

5.2 Parameter Types

There are three basic types of parameter groups:

Configuration parameters are defining the behaviour of the system.

- Measurement parameters are generated as data.
- Info parameters are kind of internal information.

The parameters IDs allowed are set as follows:

Type	Sub-type	Access type	Allowed IDs	
			Lower	Upper
Configuration	Universal	R/W	0x0000	0x0FF
	Access protected	R/W	0x0100	0x07FF
	Application Specific	R/W	0x0800	0x0FFF
Measurement	Application Specific	R	0x1000	0x1FFF
Info	Universal	R	0x2000	0x20FF
	Application Specific	R	0x2100	0x2FFF

The access type is given from the perspective of the local or distant user interface.

The "Configuration – Universal" parameters are considered as system parameters. As such, they are only modifiable by users that have at least "Administrator" rights.

The "Configuration – Access Protected" parameters can be modified by users that have at least "Configurator" rights so that their value can be reset or modified.

The “Configuration – Application Specific” parameters can be modified by users that have at least “Configurator” rights.

The other read-only parameters are accessible by any kind of user.

Values of single parameters within a group can be of different types. Here is a list of these types and the associated memory usage:

Type	Size in bytes	Remarks
Byte	1	Unsigned
Signed char	1	Signed
Word	2	Unsigned
Dword	4	Unsigned
64-bits	8	Unsigned
Time	6	Same format as used in the RTC parameter
Float	4	IEEE-754 floating-point number
String	Variable	NULL-terminated ASCII string

5.3 Parameter Changes and access

Please note that the PQF-Manager is fitted with some locking function, independently than the administrator rights.

- LOCK SWITCH: the lock switch have to be released
- INSTALLATION SETTINGS: the parameter Installation Settings must be set to Unlocked.
- The parameter COMMUNICATION LOCK is used to add an access level to users. When locked, all parameter settings modifications (except the Communication lock item setting) from the PQF-Manager touchscreen are forbidden. Parameters may meanwhile be modified by the communication access only (provided all others access levels are fulfilled).

Variable	Group ID	Locked	Unlocked
Lock switch status	0x0014	0 : Lock switch pushed	1 : Lock switch released
Installation Settings	0x0807	1 : Installation Settings are locked	0 : Installation Settings are unlocked
Communication Lock	0x0809	1 : Communication lock	0 : Communication unlocked

5.4 Parameter Groups

5.4.1 Configuration

5.4.1.1 Universal

These are the different groups with their size, type and assigned group IDs:

Group ID	Description	Size (in bytes)	Modbus Base address

0x0001	Modbus Data	4	40100
0x0002	Ethernet Data	21	31700
0x0004	Touchscreen Calibration Data	17	40400
0x0013	Backlight Settings	1	41900
0x0014	Inputs Information	5	31400

5.4.1.2 Access Protected

These are the different groups with their size, type and assigned group IDs:

Group ID	Description	Size (in bytes)	Modbus Base Address
0x0100	Min-Max Logging Voltage-Current	90	45000
0x0101	Min-Max Logging Powers-F-T	70	45100
0x0102	Min-Max Logging Temperature	160	45200
0x0103	RTC from CoBo	12	42100
0x0104	Temperature Probe Addresses	128	42200
0x0106	Installation settings	79	42300
0x0107	Function settings / Function	28	44400
0x0105	Main User Settings	131	44000
0x0109	Auxiliary User Settings	131	44100
0x010A	Filters parameter Read/Write	56	44600
0x010C	Mailbox Control	143	44800

5.4.1.3 Application specific

These are the different groups with their size, type and assigned group IDs:

Group ID	Description	Size (in bytes)	Modbus Base Address
0x0801	Password	2	42800
0x0802	User interface	4	42900
0x0803	Protection (Alarm Relay)	56	43000
0x0804	Warning/Fan (Auxiliary Relay)	40	43100
0x0805	Min-Max Logging Settings	140	43200
0x0807	User Settings	23	43400
0x0808	ID	121	43500
0x0809	Communication control	9	43600

5.4.2 Measurement

5.4.2.1 Application specific

These are the different groups with their size, type and assigned group IDs:

Group ID	Description	Size (in bytes)	Modbus Base Address
0x1000	Voltages	52	30500
0x1001	Line Currents	72	30600

0x1002	Temperature	55	30700
0x1003	Powers	20	30800
0x1005	PQF status	98	30900
0x1006	Status information	55	31000

5.4.3 Info

5.4.3.1 Universal

These are the different groups with their size, type and assigned group IDs:

Group ID	Description	Size (in bytes)	Modbus Base Address
0x2001	Ethernet current settings	20	32600
0x2002	Events	14	32700
0x2003	Errors	132	32800
0x2004	Curve Samples 1-45	180	34000
0x2005	Curve Samples 46-90	180	34100
0x2006	Curve Samples 91-135	180	34200
0x2007	Curve Samples 136-180	180	34300
0x2008	Spectrum Samples 1-25	100	34400
0x2009	Spectrum Samples 26-50	100	34500
0x2081	LED control	1	31100
0x2082	PQF Settings	32	33000

5.5 Filter’s data access for read and write

Each filter can be accessed by the GUI or by the communication (no matter who is the client, User interface of the PQF-Manager, TCP/IP, USB, Modbus).

We need 2 types of additional data to access a data in a specific filter:

- Number of the filter to access, master or slave (1 byte)
- Command to initiate a read or write into the filter of the Group ID containing the specific data

5.5.1 Specifying the filter to access (Group ID 0x0809)

The number of the filter accessed by the user interface:

FilterNumberAccessedGUI is the number of the filter to access, master or slave (1 byte).

This byte will be used by the PQF-Manager to access the right filter (for filter specific data).

If this byte is changed, all data related to the new filter number will be invalid until the data list is refilled with up-to-date values.

5.5.2 Specific Group ID access (Group ID 0x010A)

For each specific Group ID of variables needing a ‘Read’ call into the filter, there is a control byte to start a read into the filter:

FilterReadAckWXYZGUI is set to 0 by the user to ask a read into the filter of the data contained in a specific Group ID, and to 1 by the PQF-Manager when all data of the Group ID are available for the user.

For each specific Group ID of variables needing a 'Write' call into the filter, there is a control byte to start a write into the filter(s):

FilterWriteAckWXYZGUI is set to 0 by the user to ask a write into the filter of the data contained in a specific Group ID to modify specific filter parameters, and to 1 by the PQF-Manager when the data of the Group ID are changed into the filter(s).

Where WXYZ is the number of the GroupID (ex: 0106)

5.5.3 Procedure for parameter access

The procedure will then be:

- after selecting a filter Nr X by the FilterNumberAccessedGUI byte, data are automatically collected by the PQF-Manager
- each FilterReadAckWXYZ are successively set to 1 as the data arrive to the PQF-Manager
- then each FilterReadAckWXYZ can be set to 0 by the user to request an update of the data contained in the Group ID
- when updating a data in a Group ID, the FilterWriteAckWXYZ is set to 0 by the user for this Group ID.
- when the data is updated in the memory of the filter, the FilterWriteAckWXYZ is set to 1 by the PQF-Manager

5.5.4 Global Group ID access (Group ID 0x010A)

A global 'Read' access to all the filter's parameters is available by the variable FilterReadAckGUI available in Group ID 0x010A.

Setting this byte to '0' will start a global Read of all filter's Parameter's / Measurements in the selected filter. Once the global read is done, the variable FilterReadAckGUI is set to '1'.

The user should manage the use of this variable as the communication on the CAN (between the PQF-Manager and the filter) as a limited bandwidth.

5.5.5 Continuous read access (Group ID 0x010A)

FilterReadAckWXYZGUI is intended to perform a single read call into the filter's parameters. Once this variable is set to '0' it will start the filter's read call and return to '1' when the read call is done.

This is true specifically for parameters, not needing multiple read calls as the data doesn't change.

In order to get a quick and easy access to filter's measurements, a continuous read access to these filter's data is provided. So when setting a FilterReadAckWXYZGUI to '0' for a Group ID containing measurements or quickly changing data, it will not return automatically to ‘1’ but will stay to ‘0’, allowing a continuous read access to filter’s measurements.

Group ID’s concerned by this continuous read access are:

- Group ID 0x0103 (RTC from the filter)
- Group ID 0x1000 (Voltages measurements)
- Group ID 0x1001 (Currents measurements)
- Group ID 0x1002 (Temperatures measurements)
- Group ID 0x1003 (Powers measurements)

The user should set this variable to ‘1’ in order to terminate the read call access to these specific Group IDs. Otherwise the PQF-Manager will continue to read the measurements contained in these Group IDs.

The user should manage the use of these variables as the communication on the CAN (between the PQF-Manager and the filter) as a limited bandwidth.

5.6 Parameter List

The parameter list is organized in several group of parameter.

Each group of parameter is identified by a Group ID.

A few data specifies how and where the data is available or can be programmed.

Parameters settings values have a limited range. If a written value exceeds the minimum and maximum allowable values, the written group of parameter will be omitted.

1/ Byte offset - Offset in bytes of the data into the Group of parameters

2/ Description - General description

3/ Units - Units depending on the type of data

4/ Data type - Format

5/ Size in bytes - Depending on the data type

6/ Default value - Default value programmed as factory settings

7/ Min value - Minimum level allowed by this data

8/ Max value - Maximum level allowed by this data

9/ Src - Source of the data (MGR or PQF)

If ‘MGR’ than the data is located in the PQF-Manager (no need to start a filter read or write).

If ‘PQF’ than the data is located in the filter’s control board, than a FilterReadAckWXYZ is needed to refresh the data or a FilterWriteAckWXYZ is required to modify the parameter.

10/ Modb @ - Base Modbus address where the data is located while accessing through Modbus protocol

5.6.1 Configuration (Universal)

5.6.1.1 Modbus Data (GroupID = 0x0001)

Byte Offset	Description	Units	Data type	Size in bytes	Default value	Min value	Max value	Src	Modb @
0	Modbus address		Byte	1	1	1	247	MGR	40101
1	NOT USED		Byte	1	0			MGR	40102
2	Modbus baud rate	Bits/second	Byte	1	9	1	9	MGR	40103
3	Parity		Byte	1	0	0	2	MGR	40104
4	Stop bits		Byte	1	0	0	1	MGR	40105

Modbus address: Enter a slave address here which will identify the filter on the Modbus network. Make sure that the address chosen is not used by any other equipment present in the network.

Modbus baud rate: Set up the Baud rate used by the Modbus communication interface here. The ‘Baud rate’ is defined as follows:

Value	Description
1	300 bauds
2	600 bauds
3	1200 bauds
4	2400 bauds
5	4800 bauds
6	9600 bauds
7	19200 bauds
8	38400 bauds
9	57600 bauds

Parity: Set up the parity used by the Modbus communication interface here. The ‘Parity’ is defined as follows:

Parity	Signification
0	No parity
1	Even
2	Odd

Stop bits: Set up the number of stopbits used by the Modbus communication interface here. The ‘Stop Bits’ are defined as follows:

Value	Description
0	1 stop bit
1	2 stop bit

5.6.1.2 Ethernet Data (GroupID = 0x0002)

Byte Offset	Description	Units	Data type	Size in bytes	Default value	Min value	Max value	Src	Modb @
0	Static IP address		Dword	4	192.168.1.40	0	0xFFFFFFFF	MGR	31701
4	Static IP mask		Dword	4	255.255.255.0	0	0xFFFFFFFF	MGR	31703
8	Static Gateway IP address		Dword	4	0	0	0xFFFFFFFF	MGR	31705
12	NOT USED		Dword	4	0			MGR	31707
16	NOT USED		Dword	4	0			MGR	31709
20	DHCP client enabled		Byte	1	1	0	1		31711

Static IP address: Numerical label assigned to each device participating in a computer network that uses Internet Protocol. PQF-Manager uses IPv4.

Static IP mask: Subnet mask indicating the part of the IP address used for routing and the part used for numbering the devices.

Static Gateway IP address: IP address identifies the router that serves as an access point to another network. If not known, 0:0:0:0 is used as the default gateway IP address.

DHCP client enabled: Setting DHCP to enable or disable determines if the IP address is provided by the gateway or is fixed.

The 'DHCP client enabled' is defined as follows:

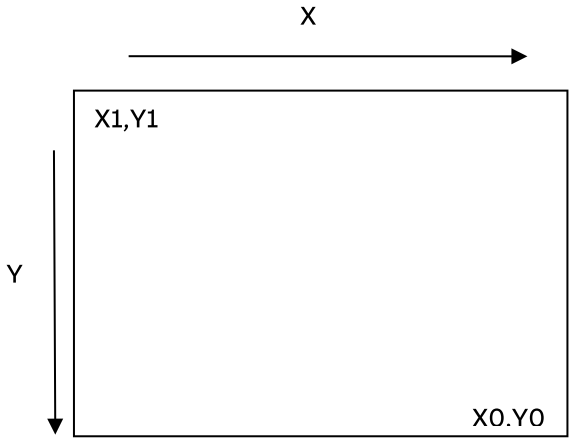
Value	Description
0	DHCP disabled
1	DHCP enabled

The IP addresses are expected to be provided in network order (big endian).

5.6.1.3 Touchscreen Calibration Data (GroupID = 0x0004)

Byte Offset	Description	Units	Data type	Size in bytes	Default value	Min value	Max value	Src	Modb @
0	X factor 0		Dword	4	3868	0	0xFFFFFFFF	MGR	40401
4	Y factor 0		Dword	4	3686	0	0xFFFFFFFF	MGR	40403
8	X factor 1		Dword	4	162	0	0xFFFFFFFF	MGR	40405
12	Y factor 1		Dword	4	334	0	0xFFFFFFFF	MGR	40407
16	Calibration Done		Byte	1	1	0	1	MGR	40409

X factor and Y factor are defined as follows:



Calibration Done: Setting it to 'Undone' will initiate a call to the Manual Touchscreen Calibration at the next boot of the PQF-Manager.

The 'Calibration Done' is defined as follows:

Value	Description
0	Undone
1	Done

5.6.1.4 Backlight settings (GroupID = 0x0013)

Byte Offset	Description	Units	Data type	Size in bytes	Default value	Min value	Max value	Src	Modb @
0	Backlight percentage	%	Byte	1	100	10	100	MGR	41901

5.6.1.5 Input information (GroupID = 0x0014)

Byte Offset	Description	Units	Data type	Size in bytes	Min value	Max value	Src	Modb @
0	NOT USED		Word	2			MGR	31401
2	NOT USED		Word	2			MGR	31402
4	Lock switch status		Byte	1	0	1	MGR	31403

Lock switch status: hardware lock to avoid modification the parameters. The 'Lock switch status' parameter is defined as follows:

Value	Description
1	Unlocked
0	Locked

5.6.2 Configuration (Access Protected)

5.6.2.1 Min-Max Logging Voltages-Currents (GroupID = 0x0100)

Byte Offset	Description	Units	Data type	Size in bytes	Default value	Min value	Max value	Src	Modb @
0	UL1L2rmsPeak	V	Float	4	0	0	9e7	MGR	45001
4	UL1L2rmsDuration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45003
10	UL2L3rmsPeak	V	Float	4	0	0	9e7	MGR	45006
14	UL2L3rmsDuration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45008
20	UL3L1rmsPeak	V	Float	4	0	0	9e7	MGR	45011
24	UL3L1rmsDuration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45013
30	InL1rmsPeak	A	Float	4	0	0	9e7	MGR	45016
34	InL1rmsDuration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45018
40	InL2rmsPeak	A	Float	4	0	0	9e7	MGR	45021
44	InL2rmsDuration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45023
50	InL3rmsPeak	A	Float	4	0	0	9e7	MGR	45026
54	InL3rmsDuration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45028
60	THDUL1L2Peak	%	Float	4	0	0	1000	MGR	45031
64	THDUL1L2Duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45033
70	THDUL2L3Peak	%	Float	4	0	0	1000	MGR	45036
74	THDUL2L3Duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45038
80	THDUL3L1Peak	%	Float	4	0	0	1000	MGR	45041
84	THDUL3L1Duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45043

The communication from the user to the PQF-Manager is limited to the reset of data, sending all the Group ID with some data set to 0 or 0:0:0:0:0:0.

Peak max recording: Maximum of the measured corresponding value.

Duration max recording: Duration during which the measured corresponding value is higher than the threshold referred as a maximum in Group ID 0x0805.

5.6.2.2 Min-Max Logging Powers-F-T (GroupID = 0x0101)

Byte Offset	Description	Units	Data type	Size in bytes	Default value	Min value	Max value	Src	Modb @
0	peak active power	W	Float	4	0	-1e10	1e10	MGR	45101
4	ActivePower Duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45103
10	peak reactive power	var	Float	4	0	-1e10	1e10	MGR	45106
14	ReactivePower Duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45108
20	peak apparent power	VA	Float	4	0	-1e10	1e10	MGR	45111
24	ApparentPower Duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45113
30	peak frequency max	Hz	Float	4	40	40	70	MGR	45116
34	FMax Duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45118
40	peak frequency min	Hz	Float	4	70	40	70	MGR	45121
44	FMin Duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45123
50	TemperatureIGBT MaxPeak	°C/°F	Float	4	-40	-40	150	MGR	45126
54	TemperatureIGBT MaxDuration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45128
60	TemperatureControl Max Peak	°C/°F	Float	4	-40	-40	150	MGR	45131
64	TemperatureControl Max Duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45133

The communication from the user to the PQF-Manager is limited to the reset of data, sending all the Group ID with some data set to 0 or 0:0:0:0:0:0.

Peak max recording: Maximum of the measured corresponding value.

Duration max recording: Duration during which the measured corresponding value is higher than the threshold refered as a maximum in Group ID 0x0805.

Peak min recording: Minimum of the measured corresponding value.

Duration min recording: Duration during which the measured corresponding value is lower than the threshold refered as a minimum in Group ID 0x0805.

Note:

TemperatureIGBT MaxPeak and TemperatureControl Max Peak are the peak value of the Max of all filters Temperature Value for a Master.

TemperatureIGBT MaxPeak and TemperatureControl Max Peak are the peak value of the corresponding filter Temperature Value for a Slave.

5.6.2.3 Max-Min Logging Temperatures (GroupID = 0x0102)

Byte Offset	Description	Units	Data type	Size in bytes	Default value	Min value	Max value	Src	Modb @
0	peak Temperature max T1	°C/°F	Float	4	-40	-40	150	MGR	45201
4	Accumulated Temperature max T1 duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45203
10	peak Temperature min T1	°C/°F	Float	4	150	-40	150	MGR	45206
14	Accumulated Temperature min T1 duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45208
20	peak Temperature max T2	°C/°F	Float	4	-40	-40	150	MGR	45211
24	Accumulated Temperature max T2 duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45213
30	peak Temperature min T2	°C/°F	Float	4	150	-40	150	MGR	45216
34	Accumulated Temperature min T2 duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45218
40	peak Temperature max T3	°C/°F	Float	4	-40	-40	150	MGR	45221
44	Accumulated Temperature max T3 duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45223
50	peak Temperature min T3	°C/°F	Float	4	150	-40	150	MGR	45226

54	Accumulated Temperature min T3 duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45228
60	peak Temperature max T4	°C/°F	Float	4	-40	-40	150	MGR	45231
64	Accumulated Temperature max T4 duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45233
70	peak Temperature min T4	°C/°F	Float	4	150	-40	150	MGR	45236
74	Accumulated Temperature min T4 duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45238
80	peak Temperature max T5	°C/°F	Float	4	-40	-40	150	MGR	45241
84	Accumulated Temperature max T5 duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45243
90	peak Temperature min T5	°C/°F	Float	4	150	-40	150	MGR	45246
94	Accumulated Temperature min T5 duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45248
100	peak Temperature max T6	°C/°F	Float	4	-40	-40	150	MGR	45251
104	Accumulated Temperature max T6 duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45253
110	peak Temperature min T6	°C/°F	Float	4	150	-40	150	MGR	45256
114	Accumulated Temperature min T6 duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45258
120	peak Temperature max T7	°C/°F	Float	4	-40	-40	150	MGR	45261

124	Accumulated Temperature max T7 duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45263
130	peak Temperature min T7	°C/°F	Float	4	150	-40	150	MGR	45266
134	Accumulated Temperature min T7 duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45268
140	peak Temperature max T8	°C/°F	Float	4	-40	-40	150	MGR	45271
144	Accumulated Temperature max T8 duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45273
150	peak Temperature min T8	°C/°F	Float	4	150	-40	150	MGR	45276
154	Accumulated Temperature min T8 duration	s	Time / 6 bytes	6	0:0:0:0:0:0	0:0:0:0:0:0	255:12:30:23:59:59	MGR	45278

The communication from the user to the PQF-Manager is limited to the reset of data, sending all the Group ID with some data set to 0 or 0:0:0:0:0:0.

Peak max recording: Maximum of the measured corresponding value.

Duration max recording: Duration during which the measured corresponding value is higher than the threshold referred as a maximum in Group ID 0x0805.

Peak min recording: Minimum of the measured corresponding value.

Duration min recording: Duration during which the measured corresponding value is lower than the threshold referred as a minimum in Group ID 0x0805.

5.6.2.4 RTC from CoBo (GroupID = 0x0103)

Byte Offset	Description	Units	Data type	Size in bytes	Default value	Min value	Max value	Src	Modb @
0	Year	Year	Word	2	11	0	255	PQF	42101
2	Month	Month	Word	2	1	1	12	PQF	42102
4	Day	Day	Word	2	1	1	31	PQF	42103
6	Hours	Hours	Word	2	0	0	23	PQF	42104
8	Minutes	Minutes	Word	2	0	0	59	PQF	42105
10	Seconds	Seconds	Word	2	0	0	59	PQF	42106

The Real Time Clock used as a reference is the one of the Master Control Board.

The Year 0 is defined as 2000 i.e. the year 2013 will be encoded as 13.

5.6.2.5 Temperature Probe Addresses (GroupID = 0x0104)

Byte Offset	Description	Units	Data type	Size in bytes	Default value	Min value	Max value	Src	Modb @
0	Address Low External temperature probe 1		64 bits	8	0	0	0xFFF…	MGR	42201
8	Address High External temperature probe 1		64 bits	8	0	0	0xFFF…	MGR	42205
16	Address Low External temperature probe 2		64 bits	8	0	0	0xFFF…	MGR	42209
24	Address High External temperature probe 2		64 bits	8	0	0	0xFFF…	MGR	42213
32	Address Low External temperature probe 3		64 bits	8	0	0	0xFFF…	MGR	42217
40	Address High External temperature probe 3		64 bits	8	0	0	0xFFF…	MGR	42221
48	Address Low External temperature probe 4		64 bits	8	0	0	0xFFF…	MGR	42225
56	Address High External temperature probe 4		64 bits	8	0	0	0xFFF…	MGR	42229
64	Address Low External temperature probe 5		64 bits	8	0	0	0xFFF…	MGR	42233
72	Address High External temperature probe 5		64 bits	8	0	0	0xFFF…	MGR	42237
80	Address Low External temperature probe 6		64 bits	8	0	0	0xFFF…	MGR	42241

88	Address High External temperature probe 6		64 bits	8	0	0	0xFFF…	MGR	42245
96	Address Low External temperature probe 7		64 bits	8	0	0	0xFFF…	MGR	42249
104	Address High External temperature probe 7		64 bits	8	0	0	0xFFF…	MGR	42253
112	Address Low External temperature probe 8		64 bits	8	0	0	0xFFF…	MGR	42257
120	Address High External temperature probe 8		64 bits	8	0	0	0xFFF…	MGR	42261

Each temperature probe has its own unique 8 bytes address.

Temperature probe addresses are automatically configured through the user interface of the PQF-Manager.

The addresses found during the Temperature probe commissioning can be read in this Group ID.

5.6.2.6 Installation Settings (GroupID = 0x0106)

Byte Offset	Description	Units	Data type	Size in bytes	Default value	Min value	Max value	Src	Modb @
0	CTScaleL1		Float	4	1	0	10000	PQF	42301
4	CTScaleL2		Float	4	1	0.1	10000	PQF	42303
8	CTScaleL3		Float	4	1	0.1	10000	PQF	42305
12	Input1Origin		Dword	4	1	1	6	PQF	42307
16	Input2Origin		Dword	4	2	1	6	PQF	42309
20	Input3Origin		Dword	4	3	1	6	PQF	42311
24	Derating	%	Float	4	100	0	100	PQF	42313
28	NeutralConnected		Byte	1	1	1	4	PQF	42315
29	UMin	%	Float	4	0	0	300	PQF	42316
33	UMax	%	Float	4	0	0	300	PQF	42318
37	Unbalance	%	Float	4	0	0	300	PQF	42320
41	GroundFault	%	Float	4	0	0	300	PQF	42322
45	TIGBTMax	°C/°F	Float	4	0	-40	105	PQF	42324

49	TControlMax	°C/°F	Float	4	0	-40	105	PQF	42326
53	Unominal	V	Dword	4	0	0	1000	PQF	42328
57	Fnominal	Hz	Dword	4	50	45	75	PQF	42330
61	FundamentalControlType		Dword	4	1	1	3	PQF	42332
65	NumberUnitsAtCommissioning	Units	Word	2	1	1	8	PQF	42334
67	PQFSerialNumber		Dword	4	0	0	0xFFFFFFFF	PQF	42335
71	Umax		Word	2	0	0	1000	PQF	42337
73	NOT USED		Byte	1	0	0	1	PQF	42338
74	NOT USED		Word	2	0	0	0xFFFF	PQF	42339
76	NOT USED		Word	2	0	0	1000	PQF	42340
78	NOT USED		Byte	1	0	0	0xFF	MGR	42341

CTScaleL1: The ratio of the CT connected in phase L1 of the supply system. E.g. 200 means a CT of 1000A/5A. The value must always be positive.

CTScaleL2: The ratio of the CT connected in phase L2 of the supply system. E.g. 200 means a CT of 1000A/5A. The value must always be positive.

CTScaleL3: The ratio of the CT connected in phase L3 of the supply system. E.g. 200 means a CT of 1000A/5A. The value must always be positive.

Input1Origin, Input2Origin, Input3Origin: Tells which line CT (CT in line L1, L2 or L3) is connected to the filter CT terminal 1 ('Input 1'). If the CT connection is inversed, a - sign precedes the label (i.e. -Line 1, -Line 2 or -Line 3).

Refer to the PQF Manual table 46 (PQFI) – 47 (PQFM) – 41 (PQFS) for more information on the CT input redirection.

The 'Input Origin' parameter is defined as follows:

Value	Description
1	L1
2	L2
3	L3
4	-L1
5	-L2
6	-L3

Derating: Expresses in % of the nominal filter current, the maximum output that the filter can generate. Default value is 100%. Reduce to a lower value if the ambient conditions are excessive.

NeutralConnected: Tells if the neutral wire is connected to the PQF.

The ‘NeutralConnected’ parameter is defined as follows:

Value	Description
1	3W
4	4W

UMin: Percentage of the rms value of the nominal network voltage under which an undervoltage warning is generated. The nominal voltage value is represented as 100%. The value of the nominal voltage is entered at the commissioning stage.

UMax: Percentage of the rms value of the nominal network voltage above which an overvoltage warning is generated. The nominal voltage value is represented as 100%. The value of the nominal voltage is entered at the commissioning stage.

Unbalance: Percentage of the network voltage imbalance above which an imbalance warning is generated.

GroundFault: Percentage of the ground current of the nominal filter current above which a ground current warning is generated. Units: Percent.

TIGBTMax: If the temperature of the hottest IGBT module is higher than “Warnings.TIGBTMax”, an IGBT temperature warning is generated. During normal operation IGBT module temperatures can reach 100°C.

TControlMax: If the temperature of the main controller board is higher than “Warnings.TcontrolMax”, a control board temperature warning is generated. During normal operation controller board temperatures can reach 70°C.

Unominal: When the voltage actually present differs a lot from the filter's rated voltage (cf. nameplate), the supply voltage to the filter auxiliaries may become too high/low. As a result the filter may operate incorrectly and/or may refuse to start.

Fnominal: Nominal network frequency. In case the value entered does not correspond to the network frequency actually present, the filter will refuse to start. The only valid values are 50Hz or 60Hz.

Fundamental Control type: Indicates the way the system synchronises on the fundamental frequency.

The ‘Fundamental Control type’ parameter is defined as follows:

Value	Description
1	synchronised on single phase
3	synchronised on three phases

NumberUnitsAtCommissioning: Gives the number of units that were interconnected at commissioning stage. Values are from 1 to 8.

5.6.2.7 Function settings / Function (GroupID = 0x0107)

Byte Offset	Description	Units	Data type	Size in bytes	Default value	Min value	Max value	Src	Modb @
0	StandBy		Word	2	0	0	1	PQF	44401
2	StandByDelay	Sec	Word	2	1	1	43200	PQF	44402
4	StandByLevel	%	Float	4	0	0	99	PQF	44403
8	StandByHysteresis	%	Float	4	0	0	23	PQF	44405
12	StartUpDelay	Sec	Word	2	1	1	43200	PQF	44407
14	StartRequest		Byte	1	0	0	1	PQF	44408
15	StopRequest		Byte	1	0	0	1	PQF	44409
16	ResetFault		Byte	1	0	0	1	PQF	44410
17	AutoRestart		Byte	1	0	0	1	PQF	44411
18	AutoRestartDelay	Sec	Word	2	1	1	43200	PQF	44412
20	NOT USED		Byte	1	0	0	18	MGR	44413
21	NOT USED		Byte	1	0	0	18	MGR	44414
22	NOT USED		Byte	1	0	0	18	MGR	44415
23	NOT USED		Byte	1	0	0	18	MGR	44416
24	NOT USED		Byte	1	0	0	6	MGR	44417
25	NOT USED		Byte	1	0	0	6	MGR	44418
26	NOT USED		Byte	1	0	0	6	MGR	44419
27	ActivationSetting		Byte	1	0	0	2	PQF	44420

StandBy: Shows whether the standby function is enabled (1) or disabled (0).

StandByDelay: When the standby function is enabled, determines the time that the load has to be lower than “StandByLevel” – “StandByHysteresis” before the IGBTs will be switched off.

StandByLevel: When the standby function is enabled, determines the nominal % load value around which the standby levels are offset (by means of “StandByHysteresis”). The standby level is expressed as a % of the nominal filter rating.

StandByHysteresis: When the standby function is enabled, determines the % hysteresis that is used to determine the lower and upper threshold of the standby level. The lower threshold is “StandByLevel” – “StandByHysteresis” and the upper threshold is “StandByLevel” +“ StandByHysteresis”.

StartUpDelay: When the standby function is enabled, determines the time that the load has to be higher than “StandByLevel” + “StandByHysteresis”, before the IGBTs will be switched on.

StartRequest: Must be set (1) to start filter, provided no error is currently present.

StopRequest: Must be set (1) to stop the filter or prevent if from restarting as soon as the error has disappeared.

ResetFault: Must be set (1) to acknowledge the fault that has caused the filter stop.

AutoRestart: When enabled (set to 1), the filter will restart automatically after a power supply outage. When it is disabled (set to 0), the filter will not restart automatically after a power supply outage. Default setting is 'Enabled'. The timer after which the filter restarts is programmable by means of “AutoRestartDelay”.

AutoRestartDelay: When the “AutoRestart” function is enabled, determines the time delay between the power coming back and the automatic filter restart.

ActivationSetting: Choose here whether the filter should use the Main settings, the Auxiliary settings or whether the choice should be determined by an external signal connected to a digital input (Ext. Input). In the latter case, the digital input should be set up and cabled correctly.

The ‘ActivationSetting’ parameter is defined as follows:

Value	Description
0	Main
1	Auxiliary
2	External input

5.6.2.8 Main User Settings (GroupID = 0x0105)

Byte Offset	Description	Units	Data type	Size in bytes	Default value	Min value	Max value	Src	Modb @
0	SelectedOrder1		Byte	1	0	0	102	PQF	44001
1	SelectedOrder2		Byte	1	0	0	102	PQF	44002
2	SelectedOrder3		Byte	1	0	0	102	PQF	44003
3	SelectedOrder4		Byte	1	0	0	102	PQF	44004
4	SelectedOrder5		Byte	1	0	0	102	PQF	44005
5	SelectedOrder6		Byte	1	0	0	102	PQF	44006
6	SelectedOrder7		Byte	1	0	0	102	PQF	44007
7	SelectedOrder8		Byte	1	0	0	102	PQF	44008
8	SelectedOrder9		Byte	1	0	0	102	PQF	44009
9	SelectedOrder10		Byte	1	0	0	102	PQF	44010
10	SelectedOrder11		Byte	1	0	0	102	PQF	44011
11	SelectedOrder12		Byte	1	0	0	102	PQF	44012
12	SelectedOrder13		Byte	1	0	0	102	PQF	44013
13	SelectedOrder14		Byte	1	0	0	102	PQF	44014
14	SelectedOrder15		Byte	1	0	0	102	PQF	44015
15	SelectedOrder16		Byte	1	0	0	102	PQF	44016
16	SelectedOrder17		Byte	1	0	0	102	PQF	44017
17	SelectedOrder18		Byte	1	0	0	102	PQF	44018

18	SelectedOrder19		Byte	1	0	0	102	PQF	44019
19	SelectedOrder20		Byte	1	0	0	102	PQF	44020
20	Selection1		Byte	1	0	0	1	PQF	44021
21	Selection2		Byte	1	0	0	1	PQF	44022
22	Selection3		Byte	1	0	0	1	PQF	44023
23	Selection4		Byte	1	0	0	1	PQF	44024
24	Selection5		Byte	1	0	0	1	PQF	44025
25	Selection6		Byte	1	0	0	1	PQF	44026
26	Selection7		Byte	1	0	0	1	PQF	44027
27	Selection8		Byte	1	0	0	1	PQF	44028
28	Selection9		Byte	1	0	0	1	PQF	44029
29	Selection10		Byte	1	0	0	1	PQF	44030
30	Selection11		Byte	1	0	0	1	PQF	44031
31	Selection12		Byte	1	0	0	1	PQF	44032
32	Selection13		Byte	1	0	0	1	PQF	44033
33	Selection14		Byte	1	0	0	1	PQF	44034
34	Selection15		Byte	1	0	0	1	PQF	44035
35	Selection16		Byte	1	0	0	1	PQF	44036
36	Selection17		Byte	1	0	0	1	PQF	44037
37	Selection18		Byte	1	0	0	1	PQF	44038
38	Selection19		Byte	1	0	0	1	PQF	44039
39	Selection20		Byte	1	0	0	1	PQF	44040
40	Curve1	A	Float	4	0	0	9000	PQF	44041
44	Curve2	A	Float	4	0	0	9000	PQF	44043
48	Curve3	A	Float	4	0	0	9000	PQF	44045
52	Curve4	A	Float	4	0	0	9000	PQF	44047
56	Curve5	A	Float	4	0	0	9000	PQF	44049
60	Curve6	A	Float	4	0	0	9000	PQF	44051
64	Curve7	A	Float	4	0	0	9000	PQF	44053
68	Curve8	A	Float	4	0	0	9000	PQF	44055
72	Curve9	A	Float	4	0	0	9000	PQF	44057
76	Curve10	A	Float	4	0	0	9000	PQF	44059
80	Curve11	A	Float	4	0	0	9000	PQF	44061
84	Curve12	A	Float	4	0	0	9000	PQF	44063
88	Curve13	A	Float	4	0	0	9000	PQF	44065
92	Curve14	A	Float	4	0	0	9000	PQF	44067
96	Curve15	A	Float	4	0	0	9000	PQF	44069
100	Curve16	A	Float	4	0	0	9000	PQF	44071
104	Curve17	A	Float	4	0	0	9000	PQF	44073

108	Curve18	A	Float	4	0	0	9000	PQF	44075
112	Curve19	A	Float	4	0	0	9000	PQF	44077
116	Curve20	A	Float	4	0	0	9000	PQF	44079
120	FilterMode		Byte	1	3	1	4	PQF	44081
121	ReactiveCompensationType		Byte	1	0	0	4	PQF	44082
122	UnbalanceCompensation		Byte	1	0	0	3	PQF	44083
123	StaticReactivePower	kvar	Float	4	0	0	1000	PQF	44084
127	TargetCosphi		Float	4	1	-1.0	1.0	PQF	44086

SelectedOrder: Twice the harmonic order to be considered by the filter.

Selection: When set, the corresponding harmonic order is filtered according to the Filter Mode.

Curve: The curve level defines the amount of current that is allowed to flow into the network for each harmonic.

The curve has to be defined in Arms.

Set the filter in mode 3 if the only harmonic requirement is to respect the curve settings.

FilterMode: To choose for the main settings the order in which filter resources will be allocated.

The ‘FilterMode’ parameter is defined as follows:

Value	Description
1	Mode 1: Curve filtering / Maximum filtering / Reactive power.
2	Curve filtering / Reactive power / Maximum filtering.
3	Curve filtering / Reactive power.
4	Q fast mode

Default value is 3. Refer to the PQF Manual table 41(PQFI) - 42 (PQFM) - 36 (PQFS) for more information on the filter mode.

ReactiveCompensationType: Main setting to select which kind of reactive power compensation has to be implemented (dynamic inductive, dynamic capacitive, static inductive or static capacitive).

The ‘ReactiveCompensationType’ parameter is defined as follows:

Value	Description	
0	Disabled	No reactive power injected by the filter
1	Dyn. ind.	Compensates till the inductive target co phi is reached (TargetCosphi)
2	Dyn. cap.	Compensates till the capacitive target cos phi is reached (TargetCosphi)

3	Static ind.	Injects the inductive amount of static reactive power specified (StaticReactivePower)
4	Static cap.	Injects the capacitive amount of static reactive power specified (StaticReactivePower)

Refer to the PQF Manual table 43 (PQFI) – 44 (PQFM) – 38 (PQFS) for more information on the reactive power injection.

StaticReactivePower: Enter the amount of static reactive power that the filter has to generate here.

Use ReactiveCompensationType to define whether the power should be capacitive (Static capacitive) or inductive (Static inductive).

TargetCosphi: Enter the target displacement power factor here. Use ReactiveCompensationType to define whether the compensation type is (dynamic) inductive or (dynamic) capacitive.

UnbalanceCompensation: Enable this feature if the filter has to do load balancing. When 1, the line to line loads are balanced; when 2, the line to neutral loads are balanced; when 3, full balancing, for line to line and line to neutral loads, is enabled.

Value	Description
0	Disabled
1	Line to Line
2	Line to Neutral (if 4 wires are connected)
3	Line to Line and Line to Neutral (if 4 wires are connected)

5.6.2.9 Auxiliary User Settings (GroupID = 0x0109)

Byte Offset	Description	Units	Data type	Size in bytes	Default value	Min value	Max value	Src	Modb @
0	SelectedOrder1		Byte	1	0	0	102	PQF	44001
1	SelectedOrder2		Byte	1	0	0	102	PQF	44102
2	SelectedOrder3		Byte	1	0	0	102	PQF	44103
3	SelectedOrder4		Byte	1	0	0	102	PQF	44104
4	SelectedOrder5		Byte	1	0	0	102	PQF	44105
5	SelectedOrder6		Byte	1	0	0	102	PQF	44106
6	SelectedOrder7		Byte	1	0	0	102	PQF	44107
7	SelectedOrder8		Byte	1	0	0	102	PQF	44108
8	SelectedOrder9		Byte	1	0	0	102	PQF	44109
9	SelectedOrder10		Byte	1	0	0	102	PQF	44110
10	SelectedOrder11		Byte	1	0	0	102	PQF	44111
11	SelectedOrder12		Byte	1	0	0	102	PQF	44112
12	SelectedOrder13		Byte	1	0	0	102	PQF	44113

13	SelectedOrder14		Byte	1	0	0	102	PQF	44114
14	SelectedOrder15		Byte	1	0	0	102	PQF	44115
15	SelectedOrder16		Byte	1	0	0	102	PQF	44116
16	SelectedOrder17		Byte	1	0	0	102	PQF	44117
17	SelectedOrder18		Byte	1	0	0	102	PQF	44118
18	SelectedOrder19		Byte	1	0	0	102	PQF	44119
19	SelectedOrder20		Byte	1	0	0	102	PQF	44120
20	Selection1		Byte	1	0	0	1	PQF	44121
21	Selection2		Byte	1	0	0	1	PQF	44122
22	Selection3		Byte	1	0	0	1	PQF	44123
23	Selection4		Byte	1	0	0	1	PQF	44124
24	Selection5		Byte	1	0	0	1	PQF	44125
25	Selection6		Byte	1	0	0	1	PQF	44126
26	Selection7		Byte	1	0	0	1	PQF	44127
27	Selection8		Byte	1	0	0	1	PQF	44128
28	Selection9		Byte	1	0	0	1	PQF	44129
29	Selection10		Byte	1	0	0	1	PQF	44130
30	Selection11		Byte	1	0	0	1	PQF	44131
31	Selection12		Byte	1	0	0	1	PQF	44132
32	Selection13		Byte	1	0	0	1	PQF	44133
33	Selection14		Byte	1	0	0	1	PQF	44134
34	Selection15		Byte	1	0	0	1	PQF	44135
35	Selection16		Byte	1	0	0	1	PQF	44136
36	Selection17		Byte	1	0	0	1	PQF	44137
37	Selection18		Byte	1	0	0	1	PQF	44138
38	Selection19		Byte	1	0	0	1	PQF	44139
39	Selection20		Byte	1	0	0	1	PQF	44140
40	Curve1	A	Float	4	0	0	9000	PQF	44141
44	Curve2	A	Float	4	0	0	9000	PQF	44143
48	Curve3	A	Float	4	0	0	9000	PQF	44145
52	Curve4	A	Float	4	0	0	9000	PQF	44147
56	Curve5	A	Float	4	0	0	9000	PQF	44149
60	Curve6	A	Float	4	0	0	9000	PQF	44151
64	Curve7	A	Float	4	0	0	9000	PQF	44153
68	Curve8	A	Float	4	0	0	9000	PQF	44155
72	Curve9	A	Float	4	0	0	9000	PQF	44157
76	Curve10	A	Float	4	0	0	9000	PQF	44159
80	Curve11	A	Float	4	0	0	9000	PQF	44161

84	Curve12	A	Float	4	0	0	9000	PQF	44163
88	Curve13	A	Float	4	0	0	9000	PQF	44165
92	Curve14	A	Float	4	0	0	9000	PQF	44167
96	Curve15	A	Float	4	0	0	9000	PQF	44169
100	Curve16	A	Float	4	0	0	9000	PQF	44171
104	Curve17	A	Float	4	0	0	9000	PQF	44173
108	Curve18	A	Float	4	0	0	9000	PQF	44175
112	Curve19	A	Float	4	0	0	9000	PQF	44177
116	Curve20	A	Float	4	0	0	9000	PQF	44179
120	FilterMode		Byte	1	3	1	4	PQF	44181
121	ReactiveCompensationType		Byte	1	0	0	4	PQF	44182
122	UnbalanceCompensation		Byte	1	0	0	3	PQF	44183
123	StaticReactivePower	kvar	Float	4	0	0	1000	PQF	44184
127	TargetCosphi		Float	4	1	-1.0	1.0	PQF	44186

SelectedOrder: Twice the harmonic order to be considered by the filter.

Selection: When set, the corresponding harmonic order is filtered according to the Filter Mode.

Curve: The curve level defines the amount of current that is allowed to flow into the network for each harmonic.

The curve has to be defined in Arms.

Set the filter in mode 3 if the only harmonic requirement is to respect the curve settings.

FilterMode: To choose for the main settings the order in which filter resources will be allocated. The ‘FilterMode’ parameter is defined as follows:

Value	Description
1	Mode 1: Curve filtering / Maximum filtering / Reactive power.
2	Curve filtering / Reactive power / Maximum filtering.
3	Curve filtering / Reactive power.
4	Q fast mode

Default value is 3. Refer to the PQF Manual table 41(PQFI) - 42 (PQFM) - 36 (PQFS) for more information on the filter mode.

ReactiveCompensationType: Auxiliary setting to select which kind of reactive power compensation has to be implemented (dynamic inductive, dynamic capacitive, static inductive or static capacitive).

The ‘ReactiveCompensationType’ parameter is defined as follows:

Value	Description	
0	Disabled	No reactive power injected by the filter
1	Dyn. ind.	Compensates till the inductive target co phi is reached (TargetCosphi)
2	Dyn. cap.	Compensates till the capacitive target cos phi is reached (TargetCosphi)
3	Static ind.	Injects the inductive amount of static reactive power specified (StaticReactivePower)
4	Static cap.	Injects the capacitive amount of static reactive power specified (StaticReactivePower)

Refer to the PQF Manual table 43 (PQFI) – 44 (PQFM) – 38 (PQFS) for more information on the reactive power injection.

StaticReactivePower: Enter the amount of static reactive power that the filter has to generate here.

Use ReactiveCompensationType to define whether the power should be capacitive (Static cap.) or inductive (Static ind.).

TargetCosphi: Enter the target displacement power factor here. Use ReactiveCompensationType to define whether the compensation type is (dynamic) inductive or (dynamic) capacitive.

UnbalanceCompensation: Enable this feature if the filter has to do load balancing. When 1, the line to line loads are balanced; when 2, the line to neutral loads are balanced; when 3, full balancing, for line to line and line to neutral loads, is enabled.

Value	Description
0	Disabled
1	Line to Line
2	Line to Neutral (if 4 wires are connected)
3	Line to Line and Line to Neutral (if 4 wires are connected)

5.6.2.10 Filter data access from/to Control Board (GroupID = 0x010A)

Byte Offset	Description	Units	Data type	Size in bytes	Default value	Min value	Max value	Src	Modb @
0	FilterReadAckGUI		Byte	1	1	0	1	MGR	44601
1	NOT USED		Byte	1	1	0	1	MGR	44602
2	FilterReadAck0000GUI		Byte	1	1	0	1	MGR	44603
3	NOT USED		Byte	1	1	0	1	MGR	44604
4	FilterWriteAck0000GUI		Byte	1	1	0	1	MGR	44605
5	NOT USED		Byte	1	1	0	1	MGR	44606
6	FilterReadAck0103GUI		Byte	1	1	0	1	MGR	44607
7	NOT USED		Byte	1	1	0	1	MGR	44608

8	FilterWriteAck0103GUI		Byte	1	1	0	1	MGR	44609
9	NOT USED		Byte	1	1	0	1	MGR	44610
10	FilterReadAck0106GUI		Byte	1	1	0	1	MGR	44611
11	NOT USED		Byte	1	1	0	1	MGR	44612
12	FilterWriteAck0106GUI		Byte	1	1	0	1	MGR	44613
13	NOT USED		Byte	1	1	0	1	MGR	44614
14	FilterReadAck0107GUI		Byte	1	1	0	1	MGR	44615
15	NOT USED		Byte	1	1	0	1	MGR	44616
16	FilterWriteAck0107GUI		Byte	1	1	0	1	MGR	44617
17	NOT USED		Byte	1	1	0	1	MGR	44618
18	FilterReadAck0105GUI		Byte	1	1	0	1	MGR	44619
19	NOT USED		Byte	1	1	0	1	MGR	44620
20	FilterWriteAck0105GUI		Byte	1	1	0	1	MGR	44621
21	NOT USED		Byte	1	1	0	1	MGR	44622
22	FilterReadAck0109GUI		Byte	1	1	0	1	MGR	44623
23	NOT USED		Byte	1	1	0	1	MGR	44624
24	FilterWriteAck0109GUI		Byte	1	1	0	1	MGR	44625
25	NOT USED		Byte	1	1	0	1	MGR	44626
26	FilterReadAck010CGUI		Byte	1	1	0	1	MGR	44627
27	NOT USED		Byte	1	1	0	1	MGR	44628
28	FilterWriteAck010CGUI		Byte	1	1	0	1	MGR	44629
29	NOT USED		Byte	1	1	0	1	MGR	44630
30	FilterReadAck0807GUI		Byte	1	1	0	1	MGR	44631
31	NOT USED		Byte	1	1	0	1	MGR	44632
32	FilterWriteAck0807GUI		Byte	1	1	0	1	MGR	44633
33	NOT USED		Byte	1	1	0	1	MGR	44634
34	FilterReadAck0808GUI		Byte	1	1	0	1	MGR	44635
35	NOT USED		Byte	1	1	0	1	MGR	44636
36	FilterWriteAck0808GUI		Byte	1	1	0	1	MGR	44637
37	NOT USED		Byte	1	1	0	1	MGR	44638
38	FilterReadAck1000GUI		Byte	1	1	0	1	MGR	44639
39	NOT USED		Byte	1	1	0	1	MGR	44640
40	FilterReadAck1001GUI		Byte	1	1	0	1	MGR	44641
41	NOT USED		Byte	1	1	0	1	MGR	44642
42	FilterReadAck1002GUI		Byte	1	1	0	1	MGR	44643
43	NOT USED		Byte	1	1	0	1	MGR	44644
44	FilterReadAck1003GUI		Byte	1	1	0	1	MGR	44645
45	NOT USED		Byte	1	1	0	1	MGR	44646
46	FilterReadAck1005GUI		Byte	1	1	0	1	MGR	44647

47	NOT USED		Byte	1	1	0	1	MGR	44648
48	FilterReadAck1006GUI		Byte	1	1	0	1	MGR	44649
49	NOT USED		Byte	1	1	0	1	MGR	44650
50	FilterReadAck2002GUI		Byte	1	1	0	1	MGR	44651
51	NOT USED		Byte	1	1	0	1	MGR	44652
52	FilterReadAck2003GUI		Byte	1	1	0	1	MGR	44653
53	NOT USED		Byte	1	1	0	1	MGR	44654
54	FilterReadAck2082GUI		Byte	1	1	0	1	MGR	44655
55	NOT USED		Byte	1	1	0	1	MGR	44656

Look into paragraph 5.5 to know how to use these data in order to:

- Specify the filter number to access using FilterNumberAccessedGUI (Group ID 0x0809)
- Start a read call to the filter using FilterReadAckWXYZGUI
- Start a write call to the filter using FilterWriteAckWXYZGUI
- Start a global read call to the filter using FilterReadAckGUI
- Start a continuous read call to the filter to access easily measurements using FilterReadAck0103GUI, FilterReadAck1000GUI, FilterReadAck1001GUI, FilterReadAck1002GUI, and FilterReadAck1003GUI.

5.6.2.11 Filter Mailbox control (GroupID = 0x010C)

Byte Offset	Description	Units	Data type	Size in bytes	Default value	Min value	Max value	Src	Modb @
0	Waveform Control		Word	2	-1	0	0xFFFF	PQF	44801
2	Spectrum Control		Word	2	-1	0	0xFFFF	PQF	44802
4	Event Control		Word	2	-1	0	0xFFFF	MGR	44803
6	SingleSelectedCurve		Byte	1	255	0	0xFF	MGR	44804
7	NOT USED		Byte	1	0	0	0xFFFF	MGR	44805
8	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44806
12	NOT USED		Byte	1	0	0	0xFF	MGR	44808
13	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44809
17	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44811
21	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44813
25	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44815
29	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44817
33	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44819
37	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44821
41	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44823
45	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44825

49	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44827
53	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44829
57	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44831
61	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44833
65	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44835
69	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44837
73	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44839
77	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44841
81	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44843
85	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44845
89	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44847
93	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44849
97	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44851
101	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44853
105	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44855
109	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44857
113	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44859
117	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44861
121	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44863
125	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44865
129	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44867
133	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44869
137	NOT USED		Dword	4	0	0	0xFFFFFFFF	MGR	44871
141	NOT USED		Byte	1	0	0	0xFF	MGR	44873
142	NOT USED		Byte	1	0	0	0xFF	MGR	44874

Waveform Control: Register controlling the waveform acquisition and which data has to be recorded.

This control register is automatically managed by the windows dll when requesting curves through USB or TCP/IP so that the user doesn't need to handle it. The samples are available with the GetCurve command (see paragraph 6: Windows DLL). Once the curves are selected the transfer is automatically done in a continuous way. Source ID's are documented in paragraph 5.7.

It is also automatically managed by the below explained 'SingleSelectedCurve' to access waveforms through Modbus.

As this register is automatically handled, the user doesn't have to worry about it. The values taken by the Waveform Control register are given below for reference.

When zero, the waveform recorder is free.

When set to 1-11 a record is ongoing.

When the acquisition is done it is set to -1.

Value set in "Waveform Control"	**Channel analyzed**	
	PQFI/PQFM	**PQFK or PQFS in 4 wire mode**
1	UL1-L2	UL1-N
2	UL2-L3	UL2-N
3	UL3-L1	UL3-N
4	IL1	IL1
5	IL2	IL2
6	IL3	IL3
7	IFilter1	IFilter1
8	IFilter2	IFilter2
9	IFilter3	IFilter3
10	Not available	IN
11	Not available	IFilterN

Note: The number of samples depends on the nominal frequency of the network.

For 50 Hz, 180 samples are available. For 60 Hz, 150 samples are available.

Spectrum Control: Register controlling the spectrum analysis and which data has to be analysed.

This control register is automatically managed by the windows dll when requesting curves through USB or TCP/IP so that the user doesn’t need to handle it. The harmonics (as waveform samples) are available with the GetCurve command (see paragraph 6: Windows DLL). Once the curves are selected the transfer is automatically done in a continuous way. Source ID’s are documented in paragraph 5.7.

It is also automatically managed by the below explained ‘SingleSelectedCurve’ to access spectrum through Modbus.

As this register is automatically handled, the user doesn’t have to worry about it. The values taken by the Spectrum Control register are given below for reference.

When zero, the spectrum analyser is free.

When set to 1-11 a new analysis is started.

When the analysis is done it is set to -1.

Value set in "Spectrum Control"	**Channel analyzed**	
	PQFI/PQFM	**PQFK or PQFS in 4 wire mode**
1	UL1-L2	UL1-N
2	UL2-L3	UL2-N
3	UL3-L1	UL3-N

4	IL1	IL1
5	IL2	IL2
6	IL3	IL3
7	IFilter1	IFilter1
8	IFilter2	IFilter2
9	IFilter3	IFilter3
10	Not available	IN
11	Not available	IFilterN

Event Control: Register controlling the transfer of one element from the event list to the event buffer.

When zero, the buffer is free for another transfer.

Set it to n between 1 and 200 for transferring the nth element of the event list in the event buffer.

The transfer is done when it is -1.

Don't forget to clear it after retrieving the event buffer. (See Group ID 0x2002).

SingleSelectedCurve: Selection of the Source ID's (Waveform or Spectrum) samples to call from a filter.

The easy way to get waveforms or spectrum samples is to use the windows dll through USB or TCP/IP.

Some commands are available to request, get or release single or multiple curves (see paragraph 6: Windows DLL). Once the curves are selected the transfer is automatically done in a continuous way.

Source ID's are documented in paragraph 5.7. These references are needed to select which curve(s) is or are requested.

These curves are not available in the same manner through Modbus communication.

Another mechanism is provided to call a curve from a filter.

The 'SingleSelectedCurve' parameter is set by default to 255.

Setting this data to the corresponding Source ID documented in paragraph 5.7 will initiate a transfer of the corresponding samples.

Source ID	Description
0	Voltage Spectrum L1-L2 or N
1	Voltage Spectrum L2-L3 or N
2	Voltage Spectrum L3-L1 or N
3	Line Current Spectrum L1
4	Line Current Spectrum L2
5	Line Current Spectrum L3
6	Filter Current Spectrum L1

7	Filter Current Spectrum L2
8	Filter Current Spectrum L3
9	Line Current Spectrum N
10	Filter Current Spectrum N
11	Samples Voltage L1-L2 or N
12	Samples Voltage L2-L3 or N
13	Samples Voltage L3-L1 or N
14	Samples Line Current In1
15	Samples Line Current In2
16	Samples Line Current In3
17	Samples Filter Current Ip1
18	Samples Filter Current Ip2
19	Samples Filter Current Ip3
20	Samples Line Current InN
21	Samples Filter Current IpN

The 'SingleSelectedCurve' will manage the 'Waveform Control' register to access waveforms through Modbus.

These samples will be available in Group IDs 0x2004, 0x2005, 0x2006, 0x2007, 0x2008, 0x2009.

The limitation compared to USB and TCP/IP is that only one waveform or one spectrum are available at a time with this mechanism.

5.6.3 Configuration (PQF-Manager Specific)

5.6.3.1 Password (GroupID = 0x0801)

Byte Offset	Description	Units	Data type	Size in bytes	Default value	Min value	Max value	Src	Modb @
0	Password		Word	2	1234	0	9999	MGR	42801

Password: This is the 4 digits password required to deactivate the Installation Lock. The password may be changed here.

Default value is 1234.

The Golden Password 6040 may be used to recover if the user forgot his password.

5.6.3.2 User interface (GroupID = 0x0802)

Byte Offset	Description	Units	Data type	Size in bytes	Default value	Min value	Max value	Src	Modb @
0	Language		Byte	1	0	0	5	MGR	42901

1	Temperature unit		Byte	1	0	0	1	MGR	42902
2	NOT USED		Byte	1	0	0	2	MGR	42903
3	NOT USED		Byte	1	100	10	100	MGR	42904

Language: To select the user interface language.

The 'Language' parameter is defined as follows:

Value	Description
0	English
1	French
2	Deutsch
3	Spanish
4	Chinese
5	Korean

Temperature unit: To select the temperature unit displayed on the PQF-Manager screen (not for Modbus data): zero for °C or one for °F.

The 'Temperature unit' parameter is defined as follows:

Value	Description
0	°C
1	°F

5.6.3.3 Protection (alarm relay n°1) (GroupID = 0x0803)

Byte Offset	Description	Units	Data type	Size in bytes	Default value	Min value	Max value	Src	Modb @
0	T1 max protection	°C or °F	Float	4	40	-40	150	MGR	43001
4	T2 max protection	°C or °F	Float	4	40	-40	150	MGR	43003
8	T3 max protection	°C or °F	Float	4	40	-40	150	MGR	43005
12	T4 max protection	°C or °F	Float	4	40	-40	150	MGR	43007
16	T5 max protection	°C or °F	Float	4	40	-40	150	MGR	43009
20	T6 max protection	°C or °F	Float	4	40	-40	150	MGR	43011
24	T7 max protection	°C or °F	Float	4	40	-40	150	MGR	43013
28	T8 max protection	°C or °F	Float	4	40	-40	150	MGR	43015
32	NOT USED		Byte	1	0	0	1	MGR	43017
33	NOT USED		Byte	1	0	0	1	MGR	43018
34	NOT USED		Byte	1	0	0	1	MGR	43019

35	NOT USED		Byte	1	0	0	1	MGR	43020
36	NOT USED		Byte	1	0	0	1	MGR	43021
37	NOT USED		Byte	1	0	0	1	MGR	43022
38	NOT USED		Byte	1	0	0	1	MGR	43023
39	NOT USED		Byte	1	0	0	1	MGR	43024
40	NOT USED		Word	2	0	0	65535	MGR	43025
42	NOT USED		Word	2	0	0	65535	MGR	43026
44	NOT USED		Word	2	0	0	65535	MGR	43027
46	NOT USED		Word	2	0	0	65535	MGR	43028
48	NOT USED		Word	2	0	0	65535	MGR	43029
50	NOT USED		Word	2	0	0	65535	MGR	43030
52	NOT USED		Word	2	0	0	65535	MGR	43031
54	NOT USED		Word	2	0	0	65535	MGR	43032

T1 – T8 max protection: Set the level at which the alarm relay will be activated for each temperature probe exceeding its threshold value.

A 5° hysteresis is applied.

5.6.3.4 Warning (warning/fan relay) (GroupID = 0x0804)

Byte Offset	Description	Units	Data type	Size in bytes	Default value	Min value	Max value	Src	Modb @
0	T1 max start fan	°C or °F	Float	4	40	-40	150	MGR	43101
4	T2 max start fan	°C or °F	Float	4	40	-40	150	MGR	43103
8	T3 max start fan	°C or °F	Float	4	40	-40	150	MGR	43105
12	T4 max start fan	°C or °F	Float	4	40	-40	150	MGR	43107
16	T5 max start fan	°C or °F	Float	4	40	-40	150	MGR	43109
20	T6 max start fan	°C or °F	Float	4	40	-40	150	MGR	43111
24	T7 max start fan	°C or °F	Float	4	40	-40	150	MGR	43113
28	T8 max start fan	°C or °F	Float	4	40	-40	150	MGR	43115
32	NOT USED		Byte	1	0	0	1	MGR	43117
33	NOT USED		Byte	1	0	0	1	MGR	43118
34	NOT USED		Byte	1	0	0	1	MGR	43119
35	NOT USED		Byte	1	0	0	1	MGR	43120

36	NOT USED		Byte	1	0	0	1	MGR	43121
37	NOT USED		Byte	1	0	0	1	MGR	43122
38	NOT USED		Byte	1	0	0	1	MGR	43123
39	NOT USED		Byte	1	0	0	1	MGR	43124

T1 – T8 max start fan: Set the level at which the auxiliary relay (warning/fan) will be activated for each temperature probe exceeding its threshold value.

A 5° hysteresis is applied.

5.6.3.5 Min Max Logging Settings (GroupID = 0x0805)

Byte Offset	Description	Units	Data type	Size in bytes	Default value	Min value	Max value	Src	Modb @
0	UL1L2rmsThreshold	V	Float	4	500	10	9e6	MGR	43201
4	UL2L3rmsThreshold	V	Float	4	500	10	9e6	MGR	43203
8	UL3L1rmsThreshold	V	Float	4	500	10	9e6	MGR	43205
12	InL1rmsThreshold	A	Float	4	1000	0	9e6	MGR	43207
16	InL2rmsThreshold	A	Float	4	1000	0	9e6	MGR	43209
20	InL3rmsThreshold	A	Float	4	1000	0	9e6	MGR	43211
24	ActivePowerThreshold	W	Float	4	1e6	0	9e7	MGR	43213
28	ReactivePowerThreshold	var	Float	4	1e6	0	9e7	MGR	43215
32	ApparentPowerThreshold	VA	Float	4	1e6	0	9e7	MGR	43217
36	THDVL1L2threshold	%	Float	4	10	0	1000	MGR	43219
40	THDVL2L3threshold	%	Float	4	10	0	1000	MGR	43221
44	THDVL3L1threshold	%	Float	4	10	0	1000	MGR	43223
48	THDIL1threshold	%	Float	4	30	0	1000	MGR	43225
52	THDIL2threshold	%	Float	4	30	0	1000	MGR	43227
56	THDIL3threshold	%	Float	4	30	0	1000	MGR	43229
60	frequency max threshold	Hz	Float	4	65	40	70	MGR	43231
64	frequency min threshold	Hz	Float	4	45	40	70	MGR	43233
68	TemperatureIGBTThreshold	°C or °F	Float	4	70	-40	150	MGR	43235
72	TemperatureControlMaxThreshold	°C or °F	Float	4	55	-40	150	MGR	43237
76	temperature T1 threshold max	°C or °F	Float	4	70	-40	150	MGR	43239
80	temperature T1 threshold min	°C or °F	Float	4	-20	-40	150	MGR	43241
84	temperature T2 threshold max	°C or °F	Float	4	70	-40	150	MGR	43243
88	temperature T2 threshold min	°C or °F	Float	4	-20	-40	150	MGR	43245

92	temperature T3 threshold max	°C or °F	Float	4	70	-40	150	MGR	43247
96	temperature T3 threshold min	°C or °F	Float	4	-20	-40	150	MGR	43249
100	temperature T4 threshold max	°C or °F	Float	4	70	-40	150	MGR	43251
104	temperature T4 threshold min	°C or °F	Float	4	-20	-40	150	MGR	43253
108	temperature T5 threshold max	°C or °F	Float	4	70	-40	150	MGR	43255
112	temperature T5 threshold min	°C or °F	Float	4	-20	-40	150	MGR	43257
116	temperature T6 threshold max	°C or °F	Float	4	70	-40	150	MGR	43259
120	temperature T6 threshold min	°C or °F	Float	4	-20	-40	150	MGR	43261
124	temperature T7 threshold max	°C or °F	Float	4	70	-40	150	MGR	43263
128	temperature T7 threshold min	°C or °F	Float	4	-20	-40	150	MGR	43265
132	temperature T8 threshold max	°C or °F	Float	4	70	-40	150	MGR	43267
136	temperature T8 threshold min	°C or °F	Float	4	-20	-40	150	MGR	43269

Threshold max: If the value of the corresponding measured value is higher than the threshold level programmed in this Group ID, the corresponding “Min Max Logging Duration” will increment.

Threshold min: If the value of the corresponding measured value is lower than the threshold level programmed in this Group ID, the corresponding “Min Max Logging Duration” will increment.

5.6.3.6 User Settings (GroupID = 0x0807)

Byte Offset	Description	Units	Data type	Size in bytes	Default value	Min value	Max value	Src	Modb @
0	Alarm1Setting		Byte	1	0	0	15	MGR	43401
1	Alarm2Setting		Byte	1	0	0	15	MGR	43402
2	Alarm3Setting		Byte	1	0	0	15	MGR	43403
3	Warning1Setting		Byte	1	0	0	6	MGR	43404
4	Warning2Setting		Byte	1	0	0	6	MGR	43405
5	Warning3Setting		Byte	1	0	0	6	MGR	43406
6	DigitalIn1Setting		Byte	1	0	0	6	MGR	43407
7	DigitalIn2Setting		Byte	1	0	0	6	MGR	43408
8	DigitalOut1Setting		Byte	1	1	0	15	MGR	43409

9	DigitalOut2Setting		Byte	1	2	0	15	MGR	43410
10	DigitalOut3Setting		Byte	1	3	0	15	MGR	43411
11	DigitalOut4Setting		Byte	1	8	0	15	MGR	43412
12	DigitalOut5Setting		Byte	1	15	0	15	MGR	43413
13	DigitalOut6Setting		Byte	1	4	0	15	MGR	43414
14	InstallationLocked		Byte	1	0	0	1	MGR	43415
15	AlarmResetDelay	Sec	Word	2	1	1	43200	MGR	43416
17	AlarmDelay	Sec	Word	2	180	180	43200	MGR	43417
19	WarningResetDelay	Sec	Word	2	1	1	43200	MGR	43418
21	WarningDelay	Sec	Word	2	1	1	43200	MGR	43419

Alarm1Setting, Alarm2Setting, Alarm3Setting: Any of the listed alarm conditions can be assigned to the Programmable alarm 1, 2 or 3.

The global alarm relay will also be activated in case a single alarm condition is reached.

Refer to the PQF Manual Table 44 (PQFI) – 45 (PQFM) – 39 (PQFS) for more information on the alarm conditions.

The 'AlarmSetting' parameter is defined as follows:

Value	**Description**
0	Disabled
1	Vrms max
2	Vrms min
3	Phase loss
4	Imbalance
5	Frequency change
6	Vdc max
7	Preload error
8	Overcurrent
9	Ground fault
10	IGBT fault
11	IGBT temporary fault
12	Temperature of the control board max
13	Power Supply fault
14	Control board
15	Any fault

Warning1Setting, Warning2Setting, Warning3Setting: Any of the listed warning conditions can be assigned to the Programmable warnings 1, 2 or 3. When using this function, appropriate warning levels should be defined.

The global auxiliary relay (warning/fan) will also be activated in case a single alarm condition is reached.

Refer to the PQF Manual for more information on the warning conditions.

The ‘WarningSetting’ parameter is defined as follows:

Value	Description
0	Disabled
1	Vrms max
2	Vrms min
3	Imbalance
4	Ground fault
5	IGBT temporary fault
6	Temperature of the control board max

DigitalIn1Setting, DigitalIn2Setting: Any of the listed functions can be assigned to Digital Input 1 or 2. When using this function, the inputs should be cabled appropriately.

Refer to the PQF Manual Table 8 for more information.

The ‘DigitalSetting’ parameter is defined as follows:

Value	Description
0	Disabled
1	Edge ON
2	Edge OFF
3	Edge ON/OFF
4	Remote ON
5	Activate main settings
6	Activate auxiliary settings

DigitalOut1Setting, DigitalOut2Setting, DigitalOut3Setting, DigitalOut4Setting,

DigitalOut5Setting, DigitalOut5Setting: Any of the listed functions can be assigned to Digital Outputs 1-6. When using this function, the outputs should be cabled appropriately.

Refer to the PQF Manual Table 9 for more information.

The ‘DigitalSetting’ parameter is defined as follows:

Value	Description
0	Disabled
1	Auxiliary ON
2	PQF runs

3	Full load
4	T limit
5	In standby
6	Main settings active
7	Auxiliary settings active
8	Armed
9	Pg. alarm 1
10	Pg. alarm 2
11	Pg. alarm 3
12	Warning 1
13	Warning 2
14	Warning 3
15	Unit missing

InstallationLocked: Software lock. Switch on the software lock (value 1) to prevent people from changing basic installation settings but still giving them access to high-level user settings (e.g. harmonics selection). Set this parameter to 0 to unlock the software lock.

The ‘InstallationLocked’ parameter is defined as follows:

Value	Description
0	Unlocked
1	Locked

AlarmResetDelay: Defines the time during which the programmed and/or system alarm condition has to disappear before the digital output and the alarm contact will be deactivated.

AlarmDelay: Defines the time during which the programmed and/or system alarm condition has to present before the digital output and the alarm contact will be activated.

WarningResetDelay: Defines the time during which the programmed warning condition has to disappear before the digital output will be deactivated. This setting is only relevant if the digital output has been set up to monitor the programmable warnings.

WarningDelay: Defines the time during which the programmed warning condition has to present before the digital output will be activated. This setting is only relevant if the digital output has been set up to monitor the programmable warnings.

5.6.3.7 ID (GroupID = 0x0808)

Byte Offset	Description	Units	Data type	Size in bytes	Default value	Min value	Max value	Src	Modb @
0	Serial number1		Dword	4	0	0	0xFFFFFFFF	MGR	43501
4	Serial number2		Dword	4	0	0	0xFFFFFFFF	MGR	43503
8	Article number		Dword	4	0	0	0xFFFFFFFF	MGR	43505
12	Relay number		Byte	1	12	6	12	MGR	43507
13	PQF-Manager model		Byte	1	2	2	10	MGR	43508
14	ABB IDnr1		Byte	1	0	0	0xFF	MGR	43509
15	ABB IDnr2		Word	2	0	0	0xFFFF	MGR	43510
17	ABB IDnr3		Dword	4	0	0	0xFFFFFFFF	MGR	43511
21	PQF Mgr V1 Version		Byte	1	0	0	0xFF	MGR	43513
22	PQF Mgr V1 Revision		Word	2	0	0	0xFFFF	MGR	43514
24	PQF Mgr V2 Version		Byte	1	0	0	0xFF	MGR	43515
25	PQF Mgr V2 Revision		Byte	1	0	0	0xFF	MGR	43516
26	PQF Mgr V3 Version		Byte	1	0	0	0xFF	MGR	43517
27	PQF Mgr V3 Revision		Byte	1	0	0	0xFF	MGR	43519
28	NOT USED		Float	4	85	0	120	MGR	43520
32	Product ID 0		Word	2	0	0	0xFFFF	MGR	43521
34	Product ID 1		Word	2	0	0	0xFFFF	MGR	43522
36	Product ID 2		Word	2	0	0	0xFFFF	MGR	43523
38	Product ID 3		Word	2	0	0	0xFFFF	MGR	43524
40	Product ID 4		Word	2	0	0	0xFFFF	MGR	43525
42	Product ID 5		Word	2	0	0	0xFFFF	MGR	43526
44	Product ID 6		Word	2	0	0	0xFFFF	MGR	43527
46	Product ID 7		Word	2	0	0	0xFFFF	MGR	43528
48	Product ID 8		Word	2	0	0	0xFFFF	MGR	43529
50	Product ID 9		Word	2	0	0	0xFFFF	MGR	43530
52	Product ID 10		Word	2	0	0	0xFFFF	MGR	43531
54	Product Type 0		Word	2	0	0	0xFFFF	MGR	43532
56	Product Type 1		Word	2	0	0	0xFFFF	MGR	43533
58	Product Type 2		Word	2	0	0	0xFFFF	MGR	43534
60	PQFAbbId1		Byte	1	0	0	0xFF	MGR	43535
61	PQFAbbId2		Word	2	0	0	0xFFFF	MGR	43536
63	PQFAbbId3		Dword	4	0	0	0xFFFFFFFF	MGR	43537
67	PQFType1		Word	2	0	0	0xFFFF	MGR	43539
69	PQFType2		Word	2	0	0	0xFFFF	MGR	43540

71	PQFType3		Word	2	0	0	0xFFFF	MGR	43541
73	SystemDescription1		Word	2	0	0	0xFFFF	MGR	43542
75	SystemDescription2		Word	2	0	0	0xFFFF	MGR	43543
77	SystemDescription3		Word	2	0	0	0xFFFF	MGR	43544
79	SystemDescription4		Word	2	0	0	0xFFFF	MGR	43545
81	SystemDescription5		Word	2	0	0	0xFFFF	MGR	43546
83	SystemDescription6		Word	2	0	0	0xFFFF	MGR	43547
85	SystemDescription7		Word	2	0	0	0xFFFF	MGR	43548
87	SystemDescription8		Word	2	0	0	0xFFFF	MGR	43549
89	SystemDescription9		Word	2	0	0	0xFFFF	MGR	43550
91	SystemDescription10		Word	2	0	0	0xFFFF	MGR	43551
93	SystemDescription11		Word	2	0	0	0xFFFF	MGR	43552
95	SystemDescription12		Word	2	0	0	0xFFFF	MGR	43553
97	SystemDescription13		Word	2	0	0	0xFFFF	MGR	43554
99	SystemId1		Word	2	0	0	0xFFFF	MGR	43555
101	SystemId2		Word	2	0	0	0xFFFF	MGR	43556
103	SystemId3		Word	2	0	0	0xFFFF	MGR	43557
105	SystemId4		Word	2	0	0	0xFFFF	MGR	43558
107	SystemId5		Word	2	0	0	0xFFFF	MGR	43559
109	SystemId6		Word	2	0	0	0xFFFF	MGR	43560
111	SystemId7		Word	2	0	0	0xFFFF	MGR	43561
113	SystemId8		Word	2	0	0	0xFFFF	MGR	43562
115	SystemId9		Word	2	0	0	0xFFFF	MGR	43563
117	SystemId10		Word	2	0	0	0xFFFF	MGR	43564
119	SystemId11		Word	2	0	0	0xFFFF	MGR	43565

This group ID is a read-only group of parameters.

Serial number: PQF-Manager serial number coded on 64 bits.

Serial number = (Serial number2)*0x10000 + Serial number1

Article number: PQF-Manager article number

Relay number: PQF-Manager number of relays

PQF-Manager model: Standard model is 2.

This parameter is defined as follows:

Value	Description
0	RVT version
1	RVT 3-phase version
2	PQF-Manager

ABB IDnr1: PQF-Manager manufacturer parameter. Filter identification number 1.

ABB IDnr2: PQF-Manager manufacturer parameter. Filter identification number 2.

ABB IDnr3: PQF-Manager manufacturer parameter. Filter identification number 3.

PQF Mgr V1 Version & Revision: (as shown in the About PQF Menu).

Version of the PQF-Manager Operating System.

PQF Mgr V2 Version & Revision: (as shown in the About PQF Menu).

Version of the PQF-Manager Configuration.

PQF Mgr V3 Version & Revision: (as shown in the About PQF Menu).

Version of the PQF-Manager User Interface.

Product ID: PQF-Manager manufacturer parameter.

Product Type 0: PQF-Manager manufacturer parameter. Two first ASCII characters describing the filter type MSB is first character.

Product Type 1: PQF-Manager manufacturer parameter. Two next ASCII characters describing the filter type MSB is first character.

Product Type 2: PQF-Manager manufacturer parameter. Two last ASCII characters describing the filter type MSB is first character.

PQFAbbId1: Manufacturer parameter. Filter identification number 1.

PQFAbbId2: Manufacturer parameter. Filter identification number 2.

PQFAbbId3: Manufacturer parameter. Filter identification number 3.

PQFType1: Manufacturer parameter. Two first ASCII characters describing the filter type MSB is first character.

PQFType2: Manufacturer parameter. Two next ASCII characters describing the filter type MSB is first character.

PQFType3: Manufacturer parameter. Two last ASCII characters describing the filter type MSB is first character.

SystemDescription: Parameter that can be written at installation of the filter.

SystemId: Manufacturer parameter.

5.6.3.8 Communication control (GroupID = 0x0809)

Byte Offset	Description	Units	Data type	Size in bytes	Default value	Min value	Max value	Src	Modb @
0	Communication lock status		Byte	1	0	0	1	MGR	43601
1	FilterNumberAccessedGUI		Byte	1	0	0	7	MGR	43602
2	NOT USED		Byte	1	0	0	255	MGR	43603
3	NOT USED		Byte	1	0	0	255	MGR	43604
4	NOT USED		Byte	1	0	0	255	MGR	43605

5	NOT USED		Byte	1	0	0	255	MGR	43606
6	NOT USED		Byte	1	0	0	255	MGR	43607
7	NOT USED		Byte	1	0	0	255	MGR	43608
8	NOT USED		Byte	1	0	0	255	MGR	43609

Communication lock status: This software lock is intended to let the user lock the parameters from the user interface, leaving access through the communication only.

The ‘Communication lock status’ parameter is defined as follows:

Value	Description
0	Unlocked
1	Locked

FilterNumberAccessedGUI: Specify the filter number which is accessed for read or write calls to the filter.

Look into paragraph 5.5 to know how to use these data in order to:

- Specify the filter number to access using FilterNumberAccessedGUI (Group ID 0x0809)
- Start a read call to the filter using FilterReadAckWXYZGUI
- Start a write call to the filter using FilterWriteAckWXYZGUI
- Start a global read call to the filter using FilterReadAckGUI
- Start a continuous read call to the filter to access easily measurements using
- FilterReadAck0103GUI , FilterReadAck1000GUI, FilterReadAck1001GUI, FilterReadAck1002GUI, and FilterReadAck1003GUI.

5.6.4 Measurement (PQF-Manager specific)

5.6.4.1 Voltage (GroupID = 0x1000)

Byte Offset	Description	Units	Data type	Size in bytes	Min value	Max value	Src	Modb @
0	RMS voltage L1-L2	V	Float	4	10	9e6	PQF	30501
4	RMS voltage L2-L3	V	Float	4	10	9e6	PQF	30503
8	RMS voltage L3-L1	V	Float	4	10	9e6	PQF	30505
12	Voltage THD L1-L2	%	Float	4	0	1000	PQF	30507
16	Voltage THD L2-L3	%	Float	4	0	1000	PQF	30509
20	Voltage THD L3-L1	%	Float	4	0	1000	PQF	30511
24	Fundamental voltage L1-L2	V	Float	4	10	9e6	PQF	30513
28	Fundamental voltage L2-L3	V	Float	4	10	9e6	PQF	30515
32	Fundamental voltage L3-L1	V	Float	4	10	9e6	PQF	30517

36	Frequency	Hz	Float	4	45	75	PQF	30519
40	Voltage imbalance	%	Float	4	0	300	PQF	30521
44	Udc	Vdc	Float	4	0	1000	PQF	30523
48	Udc Max	Vdc	Float	4	0	1000	PQF	30525

RMS voltage L1-L2: The rms voltage measured between the phases L1 and L2 (L1 and Neutral for PQFK).

RMS voltage L2-L3: The rms voltage measured between the phases L2 and L3 (L2 and Neutral for PQFK).

RMS voltage L3-L1: The rms voltage measured between the phases L3 and L1 (L3 and Neutral for PQFK).

Voltage THD L1-L2: The Total Harmonic Distortion (THDv) of the voltage measured between the phases L1 and L2 (L1 and Neutral for PQFK or PQFS connected in 4 wire mode).

Voltage THD L2-L3: The Total Harmonic Distortion (THDv) of the voltage measured between the phases L2 and L3 (L2 and Neutral for PQFK or PQFS connected in 4 wire mode).

Voltage THD L3-L1: The Total Harmonic Distortion (THDv) of the voltage measured between the phases L3 and L1 (L3 and Neutral for PQFK or PQFS connected in 4 wire mode).

Fundamental voltage L1-L2: The rms value of the voltage at fundamental frequency measured between the phases L1 and L2 (L1 and Neutral for 4W units PQFK or PQFS connected in 4 wire mode).

Fundamental voltage L2-L3: The rms value of the voltage at fundamental frequency measured between the phases L2 and L3 (L2 and Neutral for 4W units PQFK or PQFS connected in 4 wire mode).

Fundamental voltage L3-L1: The rms value of the voltage at fundamental frequency measured between the phases L3 and L1 (L3 and Neutral for 4W units PQFK or PQFS connected in 4 wire mode).

Frequency: Frequency of the supply system.

Voltage imbalance: The network imbalance expressed in %. The network voltage imbalance is calculated as the ratio of the nps (negative phase sequence) over the pps (positive phase sequence) component of the voltage.

Udc: The DC voltage present on the active filter DC bus.

Udc Max: The highest DC voltage present on the active filter DC bus on all units.

5.6.4.2 Line Currents (GroupID = 0x1001)

Byte Offset	Description	Units	Data type	Size in bytes	Min value	Max value	Src	Modb @
0	RMS line current L1	A	Float	4	0	9e6	PQF	30601
4	RMS line current L2	A	Float	4	0	9e6	PQF	30603
8	RMS line current L3	A	Float	4	0	9e6	PQF	30605
12	InZeroSequencerms	A	Float	4	0	9e6	PQF	30607
16	Line current THD L1	%	Float	4	0	1000	PQF	30609
20	Line current THD L2	%	Float	4	0	1000	PQF	30611
24	Line current THD L3	%	Float	4	0	1000	PQF	30613
28	Fundamental line current L1	A	Float	4	0	9e6	PQF	30615
32	Fundamental line current L2	A	Float	4	0	9e6	PQF	30617
36	Fundamental line current L3	A	Float	4	0	9e6	PQF	30619
40	RMS filter current L1	A	Float	4	0	9e6	PQF	30621
44	RMS filter current L2	A	Float	4	0	9e6	PQF	30623
48	RMS filter current L3	A	Float	4	0	9e6	PQF	30625
52	IpZeroSequencerms	A	Float	4	0	9e6	PQF	30627
56	CurrentScaling L1	A	Float	4	0	9e6	PQF	30629
60	CurrentScaling L2	A	Float	4	0	9e6	PQF	30631
64	CurrentScaling L3	A	Float	4	0	9e6	PQF	30633
68	CurrentScalingZeroSequence	A	Float	4	0	9e6	PQF	30635

RMS line current L1: The rms current measured in line L1.

RMS line current L2: The rms current measured in line L2.

RMS line current L3: The rms current measured in line L3.

InZeroSequencerms: The rms value of the current flowing in the neutral at the location of the CTs. Only valid if PQFK or PQFS is connected in 4 wire mode.

Line current THD L1: The Total Harmonic Distortion (THDi) of the current measured in line L1.

Line current THD L2: The Total Harmonic Distortion (THDi) of the current measured in line L2.

Line current THD L3: The Total Harmonic Distortion (THDi) of the current measured in line L3.

Fundamental line current L1: The rms value of the current at fundamental frequency measured in line L1.

Fundamental line current L2: The rms value of the current at fundamental frequency measured in line L2.

Fundamental line current L3: The rms value of the current at fundamental frequency measured in line L3.

RMS filter current L1: The rms value of the filter current measured in phase L1.

RMS filter current L2: The rms value of the filter current measured in phase L2.

RMS filter current L3: The rms value of the filter current measured in phase L3.

IpZeroSequencerms: The rms value of the filter current measured in the neutral N. Only valid if PQFK or PQFS is connected in 4 wire mode.

5.6.4.3 Temperatures (GroupID = 0x1002)

Byte Offset	Description	Units	Data type	Size in bytes	Min value	Max value	Src	Modb @
0	Internal Temperature	°C or °F	Float	4	-40	150	MGR	30701
4	External temperature probe 1	°C or °F	Float	4	-40	150	MGR	30703
8	External temperature probe 2	°C or °F	Float	4	-40	150	MGR	30705
12	External temperature probe 3	°C or °F	Float	4	-40	150	MGR	30707
16	External temperature probe 4	°C or °F	Float	4	-40	150	MGR	30709
20	External temperature probe 5	°C or °F	Float	4	-40	150	MGR	30711
24	External temperature probe 6	°C or °F	Float	4	-40	150	MGR	30713
28	External temperature probe 7	°C or °F	Float	4	-40	150	MGR	30715
32	External temperature probe 8	°C or °F	Float	4	-40	150	MGR	30717
36	TemperatureControl	°C or °F	Float	4	-40	150	PQF	30719
40	TemperatureControlMax	°C or °F	Float	4	-40	150	MGR	30721
44	TemperatureIGBT	°C or °F	Float	4	-40	150	PQF	30723
48	TemperatureIGBTMax	°C or °F	Float	4	-40	150	MGR	30725
52	HottestPhase		Byte	1	1	3	PQF	30727
53	NOT USED		Word	2	0	0xFFFF	MGR	30728

Internal Temperature: Temperature inside the PQF-Manager.

External temperature probe 1 to 8: Temperature measured by the corresponding external temperature probe connected to PQF-Manager.

Up to 8 temperature probes can be connected in daisy chain.

TemperatureControl: Temperature of the main controller board of the filter.

TemperatureControlMax: Highest temperature observed on all main controller boards of all units.

TemperatureIGBT: Highest temperature observed on all phases of the IGBT.

TemperatureIGBTMax: Highest temperature observed on all phases of the IGBT of all units.

HottestPhase: This number shows the hottest filter phase in the hottest filter module, 1 represents the phase L1, 2 represents the phase L2 and 3 represents the phase L3. It is always 1 for PQFK, PQFM and PQFS.

5.6.4.4 Powers (GroupID = 0x1003)

Byte Offset	Description	Units	Data type	Size in bytes	Min value	Max value	Src	Modb @
0	Active power	kW	Float	4	-1e9	1e9	PQF	30801
4	Reactive power	kvar	Float	4	-1e9	1e9	PQF	30803
8	Apparent power	kVA	Float	4	-1e9	1e9	PQF	30805
12	Power factor		Float	4	-1	1	PQF	30807
16	Displacement power factor (cos φ)		Float	4	-1	1	PQF	30809

Active power: Active power, P (kW), measured at the location of the CTs. If P > 0, the load absorbs active power, if P < 0, the load generates active power.

Reactive power: Reactive power, Q (kvar), measured at the location of the CTs. If Q > 0, the load is inductive, if Q < 0, the load is capacitive.

Apparent power: Apparent power, S (kVA), measured at the location of the CTs.

Power factor: Power factor (PF), the ratio between P(kW) and S(kVA), measured at the location of the CTs. The power factor is influenced by the harmonic content of the network.

Displacement power factor (cos φ): Displacement power factor (DPF) or cos φ, measured at the location of the CTs. The DPF is independent of the harmonic content of the network. A special coding is used for allowing four quadrant indications in only one number. The only two first decimal digits are significant and third are used for Inductive/capacitive indication as explained in table below.

The 'Power factor' or 'cos φ' parameter is defined as follows:

Value	Description
1.000	Resistive load only drawing active power.
0.710	Inductive load drawing almost the same amount of active and reactive power.

-0.710	Inductive generator injecting the same active power than the reactive power drawn.
0.711	Capacitive load drawing almost the same amount of active and reactive power.
-0.711	Capacitive generator injecting the same active power than the reactive power drawn.
-1.000	Generator only injecting active power.

The range is made of discrete values belonging to:

[-1.000;-0.991;-0.990;-0.981;-0.980;…;0.000;0.001;…;0.980;0.981;0.990;0.991;1.000].

5.6.4.5 PQF status (GroupID = 0x1005)

Byte Offset	Description	Units	Data type	Size in bytes	Min value	Max value	Src	Modb @
0	Standby		Byte	1	0	1	PQF	30901
1	NumberOfHarmonics		Byte	1	0	50	PQF	30902
2	StandbyOrder1		Byte	1	0	1	PQF	30903
3	StandbyOrder2		Byte	1	0	1	PQF	30904
4	StandbyOrder3		Byte	1	0	1	PQF	30905
5	StandbyOrder4		Byte	1	0	1	PQF	30906
6	StandbyOrder5		Byte	1	0	1	PQF	30907
7	StandbyOrder6		Byte	1	0	1	PQF	30908
8	StandbyOrder7		Byte	1	0	1	PQF	30909
9	StandbyOrder8		Byte	1	0	1	PQF	30910
10	StandbyOrder9		Byte	1	0	1	PQF	30911
11	StandbyOrder10		Byte	1	0	1	PQF	30912
12	StandbyOrder11		Byte	1	0	1	PQF	30913
13	StandbyOrder12		Byte	1	0	1	PQF	30914
14	StandbyOrder13		Byte	1	0	1	PQF	30915
15	StandbyOrder14		Byte	1	0	1	PQF	30916
16	StandbyOrder15		Byte	1	0	1	PQF	30917
17	StandbyOrder16		Byte	1	0	1	PQF	30918
18	StandbyOrder17		Byte	1	0	1	PQF	30919
19	StandbyOrder18		Byte	1	0	1	PQF	30920
20	StandbyOrder19		Byte	1	0	1	PQF	30921
21	StandbyOrder20		Byte	1	0	1	PQF	30922
22	FilterLoadIrms	%	Float	4	0	100	PQF	30923
26	FilterLoadIpeak	%	Float	4	0	100	PQF	30925
30	FilterLoadUdc	%	Float	4	0	100	PQF	30927
34	FilterLoadTemperature	%	Float	4	0	100	PQF	30929

38	Limitation		Byte	1	0	1	PQF	30931
39	PQFOperationDuration	Hours	Dword	4	0	0xFFFFFFFF	PQF	30932
43	FANOperationDuration	Hours	Dword	4	0	0xFFFFFFFF	PQF	30934
47	ActiveSetting		Byte	1	0	1	MGR	30936
48	Unit1Status		Byte	1	0	2	MGR	30937
49	Unit2Status		Byte	1	0	2	MGR	30938
50	Unit3Status		Byte	1	0	2	MGR	30939
51	Unit4Status		Byte	1	0	2	MGR	30940
52	Unit5Status		Byte	1	0	2	MGR	30941
53	Unit6Status		Byte	1	0	2	MGR	30942
54	Unit7Status		Byte	1	0	2	MGR	30943
55	Unit8Status		Byte	1	0	2	MGR	30944
56	PQF2State[0]		Byte	1	0	255	MGR	30945
57	PQF2State[1]		Byte	1	0	255	MGR	30946
58	WaitingForAck[0]		Byte	1	0	255	MGR	30947
59	WaitingForAck [1]		Byte	1	0	255	MGR	30948
60	IntendsToRestart[0]		Byte	1	0	255	MGR	30949
61	IntendsToRestart [1]		Byte	1	0	255	MGR	30950
62	MasterNode		Byte	1	0	7	MGR	30951
63	NodeRead		Byte	1	0	7	MGR	30952
64	MyNode		Byte	1	0	7	MGR	30953
65	CanStatus		Byte	1	0	3	MGR	30954
66	ComStatus		Byte	1	0	2	MGR	30955
67	RangeNode0		Word	2	0	0xFFFF	MGR	30956
69	RangeNode1		Word	2	0	0xFFFF	MGR	30957
71	RangeNode2		Word	2	0	0xFFFF	MGR	30958
73	RangeNode3		Word	2	0	0xFFFF	MGR	30959
75	RangeNode4		Word	2	0	0xFFFF	MGR	30960
77	RangeNode5		Word	2	0	0xFFFF	MGR	30961
79	RangeNode6		Word	2	0	0xFFFF	MGR	30962
81	RangeNode7		Word	2	0	0xFFFF	MGR	30963
83	RangePQF3		Byte	1	0	1	MGR	30964
84	RangePQFType		Byte	1	0	1	MGR	30965
85	IamMaster		Byte	1	0	1	MGR	30966
86	PQFType		Byte	1	0	3	MGR	30967
87	NOT USED		Byte	1	0	1	MGR	30968
88	NOT USED		Byte	1	0	1	MGR	30969
89	NOT USED		Byte	1	0	255	MGR	30970
90	NOT USED		Byte	1	0	255	MGR	30971

91	NOT USED		Byte	1	0	255	MGR	30972
92	PQF Serial Number		Dword	4	0	0xFFFFFFFF	PQF	30973
96	U Maximum		Word	2	0	1000	PQF	30975

Standby: Set when the filter has stopped switching IGBT's because it has entered the standby mode.

NumberOfHarmonics: Gives the number of harmonics that can be selected: 20 for PQFI/M and PQFS in 3 wire mode and 15 for PQFK and PQFS in 4 wires.

StandbyOrder1 to 20: Set if the filter cannot currently filter the selected order.

FilterLoadIrms: Gives the filter load expressed in % of the nominal rms current rating.

FilterLoadIpeak: Gives the filter load expressed in % of the nominal peak current rating.

FilterLoadUdc: Gives the filter load expressed in % of the nominal DC bus voltage rating.

FilterLoadTemperature: Gives the filter load expressed in % of the nominal temperature rating.

Limitation: When set, the filter is at full load on the peak current, rms current, temperature or DC bus voltage.

PQFOperationDuration: The total time that the filter has been running expressed in hours (h).

FANOperationDuration: The total time that the fan has been running expressed in hours (h).

ActiveSetting: Tells what are the currently active settings.

The 'ActiveSetting' parameter is defined as follows:

Value	Description
0	Auxiliary settings active
1	Main settings active

UnitStatus1 to 8: Reports the status of the unit.

The 'UnitStatus' parameter is defined as follows:

Value	Description
0	Not present (not connected on the CAN bus)
1	Ready (connected and alive on the CAN bus)
2	Fault (connected, but in fault)

PQF2State[0]**:** Reports where the filter controller stands in its state diagram.

The 'PQF2State[0]' parameter is defined as follows:

Value	Description
0	filter running
1	filter stopped
2	filter in error

PQF2State[1]: Reports where the selected slave filter controller stands in its state diagram. The 'PQF2State[1]' parameter is defined as follows:

Value	Description
0	Selected slave filter running
1	Selected slave filter stopped
2	Selected slave filter in error

WaitingForAck[0]: Reports whether the filter is in error and is waiting for an Acknowledge of fault. The 'WaitingForAck[0]' parameter is defined as follows:

Value	Description
0	Filter is ON or OFF
1	Waiting for Acknowledge Fault

WaitingForAck[1]: Reports whether the selected slave filter is in error and is waiting for an Acknowledge of fault. The 'WaitingForAck[1]' parameter is defined as follows:

Value	Description
0	Filter is ON or OFF
1	Waiting for Acknowledge Fault

IntendsToRestart[0]: Reports whether the filter is ON or Ready to be ON.

The 'IntendsToRestart[0]' parameter is defined as follows:

Value	Description
0	Not ready
1	Ready

IntendsToRestart[1]: Reports whether the selected slave filter is ON or Ready to be ON.

The 'IntendsToRestart[1]' parameter is defined as follows:

Value	Description
0	Not ready
1	Ready

MasterNode: Reports the id number of the master unit.

NodeRead: Reports the id number of the unit actually accessed for read and write calls.

MyNode: Reports the id number of the unit connected to this PQF-Manager.

CanStatus: Reports the status of the CAN communication between the PQF-Manager and the Control board to which it is connected. The 'CanStatus' parameter is defined as follows:

Value	Description
0	Communication is in error
1	Initializing the communication
2	Identifying the connected node
3	Communication is OK

ComStatus: Reports the status of the scanner procedure to setup the user interface through the CAN interface. The 'ComStatus' parameter is defined as follows:

Value	Description
0	Scanning and data download is done, filter's interface can start
1	CAN communication is not ready
2	Scanning filter's data in progress

RangNode[0] to [7]: Units sorted by unit number, starting with the Master unit.

RangePQF3: Tells if the PQF-Manager is a PQFS or not. The 'RangePQF3' parameter is defined as follows:

Value	Description
0	I'm not a PQFS
1	I am a PQFS

RangePQFType: Tells if the filter is connected in 3 wires or 4 wires

The 'RangePQFType' parameter is defined as follows:

Value	Description
0	3 Wires
1	4 Wires

IamMaster: Tells if the PQF-Manager is connected to a Master or a Slave filter

The 'IamMaster' parameter is defined as follows:

Value	Description
0	Connected to a Slave
1	Connected to a Master

PQFType: Industrial PQFI (if zero), PQFM (if one), Commercial PQFK (if two), or PQFS (if three). The PQFK is a 4-wire filter where the PQFI and PQFM are 3-wire filters and PQFS can work in both configurations.

The ‘PQFType’ parameter is defined as follows:

Value	Description
0	PQFI
1	PQFM
2	PQFK
3	PQFS

PQFSerialNumber: The PQF serial number. Note this number for future reference. UMaximum: Maximum rated network voltage for the components.

5.6.4.6 Status information (GroupID = 0x1006)

Byte Offset	Description	Units	Data type	Size in bytes	Min value	Max value	Src	Modb @
0	Relay status		Word	2	0	0x3FFF	MGR	31001
2	External input status		Byte	1	0	3	MGR	31002
3	Fan/Warning relay status		Byte	1	0	1	MGR	31003
4	Alarm relay status		Byte	1	0	1	MGR	31004
5	NOT USED		Byte	1	0	0XFF	MGR	31005
6	NOT USED		Byte	1	0	0xFF	MGR	31006
7	NOT USED		Byte	1	0	0xFF	MGR	31007
8	NOT USED		Byte	1	0	0xFF	MGR	31008
9	NOT USED		Byte	1	0	0xFF	MGR	31009
10	NOT USED		Byte	1	0	0xFF	MGR	31010
11	NOT USED		Byte	1	0	0xFF	MGR	31011
12	NOT USED		Byte	1	0	0xFF	MGR	31012
13	NOT USED		Byte	1	0	0xFF	MGR	31013
14	NOT USED		Byte	1	0	0xFF	MGR	31014
15	NOT USED		Byte	1	0	0xFF	MGR	31015
16	NOT USED		Byte	1	0	0xFF	MGR	31016
17	NOT USED		Byte	1	0	0xFF	MGR	31017
18	NOT USED		Byte	1	0	0xFF	MGR	31018
19	NOT USED		Byte	1	0	0xFF	MGR	31019
20	NOT USED		Dword	4	0	0xFFFFFFFF	MGR	31020
24	NOT USED		Dword	4	0	0xFFFFFFFF	MGR	31022
28	NOT USED		Dword	4	0	0xFFFFFFFF	MGR	31024
32	PQF Mgr V4 Version		Dword	4	0	0xFFFFFFFF	MGR	31026

36	UCVersion		Dword	4	0	0xFFFFFFFF	MGR	31028
40	DSPVersion		Dword	4	0	0xFFFFFFFF	MGR	31030
44	NOT USED		Byte	1	0	0xFF	MGR	31032
45	NOT USED		Byte	1	0	0xFF	MGR	31033
46	NOT USED		Byte	1	0	0xFF	MGR	31034
47	NOT USED		Byte	1	0	0xFF	MGR	31035
48	Alarm 0		Byte	1	0	0xFF	MGR	31036
49	Alarm 1		Byte	1	0	0xFF	MGR	31037
50	Alarm 2		Byte	1	0	0xFF	MGR	31038
51	Alarm 3		Byte	1	0	0xFF	MGR	31039
52	Warning 0		Byte	1	0	0xFF	MGR	31040
53	Warning 1		Byte	1	0	0xFF	MGR	31041
54	Warning 2		Byte	1	0	0xFF	MGR	31042

The ‘Relay Status’ parameter is defined as follows:

Value	Description
0	Relay open
1	Relay closed

Bit used	Output relay
Bit 0	1
Bit 1	2
Bit 2	3
Bit 3	4
Bit 4	5
Bit 5	6
Bit 6	Not used
Bit 7	Not used
Bit 8	Not used
Bit 9	Not used
Bit 10	Not used
Bit 11	Not used
Bit 12	Alarm
Bit 13	Fan
Bit 14	Not used

The ‘External input status’ parameter is defined as follows:

Value	Description
0	External input reset
1	External input set

Bit used	External Input Number
Bit 0	1
Bit 1	2
Bit 2	Not used
Bit 3	Not used
Bit 4	Not used
Bit 5	Not used
Bit 6	Not used
Bit 7	Not used

The ‘Fan/Warning relay status’ parameter is defined as follows:

Value	Description
0	Reset
1	Set

The ‘Alarm relay status’ parameter is defined as follows:

Value	Description
0	Reset
1	Set

The ‘Alarm0 to 2’ parameter is defined as follows:

Value	Description
0	Alarm not set according Alarm1Setting, Alarm2Setting, Alarm3Setting
1	Alarm set according Alarm1Setting, Alarm2Setting, Alarm3Setting

The ‘Alarm3’ parameter is defined as follows:

Value	Description
0	Global Alarm not set according Alarm1Setting, Alarm2Setting, Alarm3Setting
1	Global Alarm set according Alarm1Setting, Alarm2Setting, Alarm3Setting

The ‘Warning0 to 2’ parameter is defined as follows:

Value	Description
0	Warning not set according Warning1Setting, Warning2Setting, Warning3Setting
1	Warning set according Warning1Setting, Warning2Setting, Warning3Setting

PQF Mgr V4 Version: This is the firmware version/revision number of the software of the PQF-Manager as shown in the About PQF menu. Note this number for future reference. The double word is made of four bytes WXYZ so the software is Version X.Y revision Z.

UCVersion: This is the firmware version/revision number of the software controlling the filter microcontroller. Note this number for future reference. The double word is made of four bytes WXYZ so the software is Version X.Y revision Z.

DSPVersion: This is the firmware version/revision number of the software controlling the filter DSP controllers. Note this number for future reference. The double word is made of four bytes WXYZ so the software is Version X.Y revision Z.

5.6.5 Info – Universal

5.6.5.1 Ethernet current configuration (GroupID = 0x2001)

Byte Offset	Description	Units	Data type	Size in bytes	Min value	Max value	Src	Modb @
0	Current IP address		Dword	4	0	0xFFFFFFFF	MGR	32601
4	Current IP mask		Dword	4	0	0xFFFFFFFF	MGR	32603
8	Current Gateway IP address		Dword	4	0	0xFFFFFFFF	MGR	32605
12	NOT USED		Dword	4	0	0xFFFFFFFF	MGR	32607
16	NOT USED		Dword	4	0	0xFFFFFFFF	MGR	32609

These variables are providing information about the current network configuration.

When DHCP is disabled, those values will be the same as the one from the Ethernet Data parameter.

When DHCP is enabled, those values will be different than the static ones provided in the Ethernet Data parameter.

5.6.5.2 Events (GroupID = 0x2002)

Byte Offset	Description	Units	Data type	Size in bytes	Min value	Max value	Src	Modb @
0	Event		Byte	1	0	0xFF	PQF	32701
1	Error		Dword	4	0	0xFFFFFFFF	MGR	32702
5	Millisecond		Word	2	0	999	MGR	32704

7	Second		Byte	1	0	59	MGR	32705
8	Minute		Byte	1	0	59	MGR	32706
9	Hour		Byte	1	0	23	MGR	32707
10	Day		Byte	1	0	31	MGR	32708
11	Month		Byte	1	0	12	MGR	32709
12	Year		Byte	1	0	99	MGR	32710
13	LatestEventIndex		Byte	1	0	0xFF	PQF	32711

Event Control: (See Group ID 0x010C) Register controlling the transfer of one element from the event list to the MODBUS readable event buffer: When zero, the buffer is free for another transfer; set it to n between 1 and 200 for transferring the nth element of the event list in the event buffer; transfer is done when it is -1; don't forget to clear it after retrieving the event buffer.

Event: Event that is stored in the event list at the location indexed by the control register.

Refer to the PQF Manual table 50 (PQFI) – (51 (PQFM) – 45 (PQFS) for more information on the warning reported by controller.

The ‘Event’ parameter is defined as follows:

Value	Description
0	No event
1	Fault (DSP)
2	Fault (uC)
3	No more fault
4	n.a.
5	Start request
6	Stop request
7	Energisation
8	Power outage
9	System reset
10	Download DSP
11	DSP stop

Error: When not zero, this register contains the error (Dword) that has caused the event.

If the "Event" is 1, the current error was detected by the DSP. Please refer to the description of "CurrentDSPError" for the list of possible DSP errors.

If the "Event" is 2, the current error was detected by the UC. Please refer to the description of "CurrentUCError" for the list of possible UC errors.

For any other value of "Event", the error is zero.

Millisecond: Contains the millisecond at which the event of the event buffer took place.

Second: Absolute second at which the event of the event buffer took place.

Minute: Absolute minute at which the event of the event buffer took place.

Hour: Absolute hour at which the event of the event buffer took place.

Day: Absolute day at which the event of the event buffer took place.

Month: Absolute month at which the event of the event buffer took place.

Year: Absolute year at which the event of the event buffer took place. Add 2000 to the value read for absolute year.

LatestEventIndex: Index of the latest event. When zero, the latest event is stored at location 200 in the event list.

5.6.5.3 Errors (GroupID = 0x2003)

Byte Offset	Description	Units	Data type	Size in bytes	Min value	Max value	Src	Modb @
0	CurrentUCError		Dword	4	0	0xFFFFFFFF	PQF	32801
4	CurrentDSPError		Dword	4	0	0xFFFFFFFF	PQF	32803
8	NumberUCError1		Word	2	0	65535	PQF	32805
10	NumberUCError2		Word	2	0	65535	PQF	32806
12	NumberUCError3		Word	2	0	65535	PQF	32807
14	NumberUCError4		Word	2	0	65535	PQF	32808
16	NumberUCError5		Word	2	0	65535	PQF	32809
18	NumberUCError6		Word	2	0	65535	PQF	32810
20	NumberUCError7		Word	2	0	65535	PQF	32811
22	NumberUCError8		Word	2	0	65535	PQF	32812
24	NumberUCError9		Word	2	0	65535	PQF	32813
26	NumberUCError10		Word	2	0	65535	PQF	32814
28	NumberUCError11		Word	2	0	65535	PQF	32815
30	NumberUCError12		Word	2	0	65535	PQF	32816
32	NumberUCError13		Word	2	0	65535	PQF	32817
34	NumberUCError14		Word	2	0	65535	PQF	32818
36	NumberUCError15		Word	2	0	65535	PQF	32819
38	NumberUCError16		Word	2	0	65535	PQF	32820
40	NumberUCError17		Word	2	0	65535	PQF	32821
42	NumberUCError18		Word	2	0	65535	PQF	32822
44	NumberUCError19		Word	2	0	65535	PQF	32823
46	NumberUCError20		Word	2	0	65535	PQF	32824
48	NumberUCError21		Word	2	0	65535	PQF	32825
50	NumberUCError22		Word	2	0	65535	PQF	32826
52	NumberUCError23		Word	2	0	65535	PQF	32827

54	NumberUCError24		Word	2	0	65535	PQF	32828
56	NumberUCError25		Word	2	0	65535	PQF	32829
58	NumberUCError26		Word	2	0	65535	PQF	32830
60	NumberDSPError1		Word	2	0	65535	PQF	32831
62	NumberDSPError2		Word	2	0	65535	PQF	32832
64	NumberDSPError3		Word	2	0	65535	PQF	32833
66	NumberDSPError4		Word	2	0	65535	PQF	32834
68	NumberDSPError5		Word	2	0	65535	PQF	32835
70	NumberDSPError6		Word	2	0	65535	PQF	32836
72	NumberDSPError7		Word	2	0	65535	PQF	32837
74	NumberDSPError8		Word	2	0	65535	PQF	32838
76	NumberDSPError9		Word	2	0	65535	PQF	32839
78	NumberDSPError10		Word	2	0	65535	PQF	32840
80	NumberDSPError11		Word	2	0	65535	PQF	32841
82	NumberDSPError12		Word	2	0	65535	PQF	32842
84	NumberDSPError13		Word	2	0	65535	PQF	32843
86	NumberDSPError14		Word	2	0	65535	PQF	32844
88	NumberDSPError15		Word	2	0	65535	PQF	32845
90	NumberDSPError16		Word	2	0	65535	PQF	32846
92	NumberDSPError17		Word	2	0	65535	PQF	32847
94	NumberDSPError18		Word	2	0	65535	PQF	32848
96	NumberDSPError19		Word	2	0	65535	PQF	32849
98	NumberDSPError20		Word	2	0	65535	PQF	32850
100	NumberDSPError21		Word	2	0	65535	PQF	32851
102	NumberDSPError22		Word	2	0	65535	PQF	32852
104	NumberDSPError23		Word	2	0	65535	PQF	32853
106	NumberDSPError24		Word	2	0	65535	PQF	32854
108	NumberDSPError25		Word	2	0	65535	PQF	32855
110	NumberDSPError26		Word	2	0	65535	PQF	32856
112	NumberDSPError27		Word	2	0	65535	PQF	32857
114	NumberDSPError28		Word	2	0	65535	PQF	32858
115	NumberDSPError29		Word	2	0	65535	PQF	32859
116	NumberDSPError30		Word	2	0	65535	PQF	32860
118	NumberDSPError31		Word	2	0	65535	PQF	32861
120	TrippedPhase		Byte	1	1	3	PQF	32862
122	CurrentUCWarning		Byte	1	0	0xFF	PQF	32863
124	CurrentDSPWarning		Dword	4	0	0xFFFFFFFF	PQF	32864
128	FaultStopError		Dword	4	0	0xFFFFFFFF	PQF	32866

CurrentDSPError: Current global error status in the DSP.

When the bit is set in the double word, the corresponding error is present.

Refer to the PQF Manual table 51 (PQFI) -52 (PQFM) – 46 (PQFS) for more information on the errors reported by DSP. The 'CurrentDSPError' parameter is defined as follows:

Bit used	Description
Bit 0	Overvoltage RMS
Bit 1	Overvolt. transient (SW)
Bit 2	n.a.
Bit 3	Undervoltage RMS
Bit 4	Loss of phase
Bit 5	Wrong phase rotation
Bit 6	Unbalanced supply
Bit 7	n.a.
Bit 8	Bad CT connection
Bit 9	Out of mains freq. limit
Bit 10	Unstable mains frequency
Bit 11	No synchronisation
Bit 12	DC overvoltage (SW)
Bit 13	DC overvoltage (HW)
Bit 14	DC undervoltage (SW)
Bit 15	Preload problem
Bit 16	DC Top overvoltage
Bit 17	DC Bot overvoltage
Bit 18	Overcurrent peak (SW)
Bit 19	Overcurrent RMS
Bit 20	Overcurrent peak (HW)
Bit 21	Ground fault
Bit 22	IGBT temporary
Bit 23	IGBT permanent
Bit 24	IGBT check cooling
Bit 25	SPI timeout
Bit 26	Mismatch between units
Bit 27	n.a.
Bit 28	n.a.
Bit 29	Bad message sequence
Bit 30	Bad parameters
Bit 31	Critical

CurrentDSPWarning: Current global warning status in the DSP.

When the bit is set in the double word, the corresponding warning is present. Refer to the PQF Manual table 31 (PQFI) – 32 (PQFM) – 26 (PQFS) for more information on the warning reported by DSP.

The ‘CurrentDSPWarning’ parameter is defined as follows:

Bit used	Description
Bit 0	Overvoltage RMS
Bit 3	Undervoltage RMS
Bit 6	Unbalanced supply
Bit 21	Ground fault
Bit 24	IGBT check cooling

CurrentUCError: Current global error status in the UC.

When the bit is set in the double word, the corresponding error is present.

Refer to the PQF Manual table 31 (PQFI) – 32 (PQFM) – 26 (PQFS) for more information on the warning reported by controller.

The ‘CurrentUCError’ parameter is defined as follows:

Bit used	Description
Bit 0	n.a.
Bit 1	n.a.
Bit 2	Ctrl overtemperature
Bit 3	Real time clock problem
Bit 4	Com. problem (CAN bus)
Bit 5	Com. problem (RS232)
Bit 6	Preload time-out
Bit 7	Breaker/Cont trip
Bit 8	Power supply fault
Bit 9	Internal uC fault
Bit 10	Reactor overtemperature
Bit 11	Watchdog fault
Bit 12	PWM check cooling
Bit 13	n.a.
Bit 14	n.a.
Bit 15	Corrupted uC code
Bit 16	Corrupted DSP code
Bit 17	n.a.
Bit 18	Different firmwares

Bit 19	n.a.
Bit 20	DSP watchdog
Bit 21	SPI Timeout
Bit 22	n.a.
Bit 23	n.a.
Bit 24	Several units same id

CurrentUCWarning: Current global warning status in the UC.

When the bit is set in the byte, the corresponding warning is present.

The ‘CurrentUCWarning’ parameter is defined as follows:

Bit used	Description
Bit 0	n.a.
Bit 1	n.a.
Bit 2	Ctrl overtemperature

NumberDSPError: Number of times the error described at bit index 0 in the DSP error table (see “CurrentDSPError” description) has occurred since the filter controller has been initialised.

NumberUCError: Number of times the error described at bit index 0 in the UC error table (see “CurrentUCError” description) has occurred since the filter controller has been initialised.

TrippedPhase: This number represents the (hottest) filter phase in which the temperature exceeded the maximum permissible level. As a result the filter was stopped (tripped). 1 represents L1, 2 represents L2 and 3 represents L3. Always 1 for PQFM and PQFS.

The ‘TrippedPhase’ parameter is defined as follows:

Value	Description
1	L1
2	L2
3	L3

FaultStopError: double word depicting the reason why the filter was stopped and is actually in error state. It can be a UC or DSP error.

When the bit is set in the double word, the corresponding error is present.

Bit used	Description
Bit 0	On Bit 31 = 1 look in CurrentDSPError bit error table On Bit 31 = 0 look in CurrentUCError bit error table
Bit ….	On Bit 31 = 1 look in CurrentDSPError bit error table

	On Bit 31 = 0 look in CurrentUCError bit error table
Bit 30	On Bit 31 = 1 look in CurrentDSPError bit error table On Bit 31 = 0 look in CurrentUCError bit error table
Bit 31	When 0 UC is the reason of the error When 1 DSP is the reason of the error

5.6.5.4 Waveforms (Samples 1-45) (GroupID = 0x2004)

Byte Offset	Description	Units	Data type	Size in bytes	Src	Modb @
0	Sample 1		Float	4	MGR	34001
4	Sample 2		Float	4	MGR	34003
8	Sample 3		Float	4	MGR	34005
12	Sample 4		Float	4	MGR	34007
16	Sample 5		Float	4	MGR	34009
20	Sample 6		Float	4	MGR	34011
24	Sample 7		Float	4	MGR	34013
28	Sample 8		Float	4	MGR	34015
32	Sample 9		Float	4	MGR	34017
36	Sample 10		Float	4	MGR	34019
40	Sample 11		Float	4	MGR	34021
44	Sample 12		Float	4	MGR	34023
48	Sample 13		Float	4	MGR	34025
52	Sample 14		Float	4	MGR	34027
56	Sample 15		Float	4	MGR	34029
60	Sample 16		Float	4	MGR	34031
64	Sample 17		Float	4	MGR	34033
68	Sample 18		Float	4	MGR	34035
72	Sample 19		Float	4	MGR	34037
76	Sample 20		Float	4	MGR	34039
80	Sample 21		Float	4	MGR	34041
84	Sample 22		Float	4	MGR	34043
88	Sample 23		Float	4	MGR	34045
92	Sample 24		Float	4	MGR	34047
96	Sample 25		Float	4	MGR	34049
100	Sample 26		Float	4	MGR	34051
104	Sample 27		Float	4	MGR	34053
108	Sample 28		Float	4	MGR	34055
112	Sample 29		Float	4	MGR	34057
116	Sample 30		Float	4	MGR	34059
120	Sample 31		Float	4	MGR	34061

124	Sample 32		Float	4	MGR	34063
128	Sample 33		Float	4	MGR	34065
132	Sample 34		Float	4	MGR	34067
136	Sample 35		Float	4	MGR	34069
140	Sample 36		Float	4	MGR	34071
144	Sample 37		Float	4	MGR	34073
148	Sample 38		Float	4	MGR	34075
152	Sample 39		Float	4	MGR	34077
156	Sample 40		Float	4	MGR	34079
160	Sample 41		Float	4	MGR	34081
164	Sample 42		Float	4	MGR	34083
168	Sample 43		Float	4	MGR	34085
172	Sample 44		Float	4	MGR	34087
176	Sample 45		Float	4	MGR	34089

5.6.5.5 Waveforms (Samples 46-90) (GroupID = 0x2005)

Byte Offset	Description	Units	Data type	Size in bytes	Src	Modb @
0	Sample 46		Float	4	MGR	34101
4	Sample 47		Float	4	MGR	34103
8	Sample 48		Float	4	MGR	34105
12	Sample 49		Float	4	MGR	34107
16	Sample 50		Float	4	MGR	34109
20	Sample 51		Float	4	MGR	34111
24	Sample 52		Float	4	MGR	34113
28	Sample 53		Float	4	MGR	34115
32	Sample 54		Float	4	MGR	34117
36	Sample 55		Float	4	MGR	34119
40	Sample 56		Float	4	MGR	34121
44	Sample 57		Float	4	MGR	34123
48	Sample 58		Float	4	MGR	34125
52	Sample 59		Float	4	MGR	34127
56	Sample 60		Float	4	MGR	34129
60	Sample 61		Float	4	MGR	34131
64	Sample 62		Float	4	MGR	34133
68	Sample 63		Float	4	MGR	34135
72	Sample 64		Float	4	MGR	34137
76	Sample 65		Float	4	MGR	34139
80	Sample 66		Float	4	MGR	34141
84	Sample 67		Float	4	MGR	34143

88	Sample 68		Float	4	MGR	34145
92	Sample 69		Float	4	MGR	34147
96	Sample 70		Float	4	MGR	34149
100	Sample 71		Float	4	MGR	34151
104	Sample 72		Float	4	MGR	34153
108	Sample 73		Float	4	MGR	34155
112	Sample 74		Float	4	MGR	34157
116	Sample 75		Float	4	MGR	34159
120	Sample 76		Float	4	MGR	34161
124	Sample 77		Float	4	MGR	34163
128	Sample 78		Float	4	MGR	34165
132	Sample 79		Float	4	MGR	34167
136	Sample 80		Float	4	MGR	34169
140	Sample 81		Float	4	MGR	34171
144	Sample 82		Float	4	MGR	34173
148	Sample 83		Float	4	MGR	34175
152	Sample 84		Float	4	MGR	34177
156	Sample 85		Float	4	MGR	34179
160	Sample 86		Float	4	MGR	34181
164	Sample 87		Float	4	MGR	34183
168	Sample 88		Float	4	MGR	34185
172	Sample 89		Float	4	MGR	34187
176	Sample 90		Float	4	MGR	34189

5.6.5.6 Waveforms (Samples 91-135) (GroupID = 0x2006)

Byte Offset	Description	Units	Data type	Size in bytes	Src	Modb @
0	Sample 91		Float	4	MGR	34201
4	Sample 92		Float	4	MGR	34203
8	Sample 93		Float	4	MGR	34205
12	Sample 94		Float	4	MGR	34207
16	Sample 95		Float	4	MGR	34209
20	Sample 96		Float	4	MGR	34211
24	Sample 97		Float	4	MGR	34213
28	Sample 98		Float	4	MGR	34215
32	Sample 99		Float	4	MGR	34217
36	Sample 100		Float	4	MGR	34219
40	Sample 101		Float	4	MGR	34221
44	Sample 102		Float	4	MGR	34223
48	Sample 103		Float	4	MGR	34225

52	Sample 104		Float	4	MGR	34227
56	Sample 105		Float	4	MGR	34229
60	Sample 106		Float	4	MGR	34231
64	Sample 107		Float	4	MGR	34233
68	Sample 108		Float	4	MGR	34235
72	Sample 109		Float	4	MGR	34237
76	Sample 110		Float	4	MGR	34239
80	Sample 111		Float	4	MGR	34241
84	Sample 112		Float	4	MGR	34243
88	Sample 113		Float	4	MGR	34245
92	Sample 114		Float	4	MGR	34247
96	Sample 115		Float	4	MGR	34249
100	Sample 116		Float	4	MGR	34251
104	Sample 117		Float	4	MGR	34253
108	Sample 118		Float	4	MGR	34255
112	Sample 119		Float	4	MGR	34257
116	Sample 120		Float	4	MGR	34259
120	Sample 121		Float	4	MGR	34261
124	Sample 122		Float	4	MGR	34263
128	Sample 123		Float	4	MGR	34265
132	Sample 124		Float	4	MGR	34267
136	Sample 125		Float	4	MGR	34269
140	Sample 126		Float	4	MGR	34271
144	Sample 127		Float	4	MGR	34273
148	Sample 128		Float	4	MGR	34275
152	Sample 129		Float	4	MGR	34277
156	Sample 130		Float	4	MGR	34279
160	Sample 131		Float	4	MGR	34281
164	Sample 132		Float	4	MGR	34283
168	Sample 133		Float	4	MGR	34285
172	Sample 134		Float	4	MGR	34287
176	Sample 135		Float	4	MGR	34289

5.6.5.7 Waveforms (Samples 136-180) (GroupID = 0x2007)

Byte Offset	Description	Units	Data type	Size in bytes	Src	Modb @
0	Sample 136		Float	4	MGR	34301
4	Sample 137		Float	4	MGR	34303
8	Sample 138		Float	4	MGR	34305
12	Sample 139		Float	4	MGR	34307

16	Sample 140		Float	4	MGR	34309
20	Sample 141		Float	4	MGR	34311
24	Sample 142		Float	4	MGR	34313
28	Sample 143		Float	4	MGR	34315
32	Sample 144		Float	4	MGR	34317
36	Sample 145		Float	4	MGR	34319
40	Sample 146		Float	4	MGR	34321
44	Sample 147		Float	4	MGR	34323
48	Sample 148		Float	4	MGR	34325
52	Sample 149		Float	4	MGR	34327
56	Sample 150		Float	4	MGR	34329
60	Sample 151		Float	4	MGR	34331
64	Sample 152		Float	4	MGR	34333
68	Sample 153		Float	4	MGR	34335
72	Sample 154		Float	4	MGR	34337
76	Sample 155		Float	4	MGR	34339
80	Sample 156		Float	4	MGR	34341
84	Sample 157		Float	4	MGR	34343
88	Sample 158		Float	4	MGR	34345
92	Sample 159		Float	4	MGR	34347
96	Sample 160		Float	4	Src	34349
100	Sample 161		Float	4	MGR	34351
104	Sample 162		Float	4	MGR	34353
108	Sample 163		Float	4	MGR	34355
112	Sample 164		Float	4	MGR	34357
116	Sample 165		Float	4	MGR	34359
120	Sample 166		Float	4	MGR	34361
124	Sample 167		Float	4	MGR	34363
128	Sample 168		Float	4	MGR	34365
132	Sample 169		Float	4	MGR	34367
136	Sample 170		Float	4	MGR	34369
140	Sample 171		Float	4	MGR	34371
144	Sample 172		Float	4	MGR	34373
148	Sample 173		Float	4	MGR	34375
152	Sample 174		Float	4	MGR	34377
156	Sample 175		Float	4	MGR	34379
160	Sample 176		Float	4	MGR	34381
164	Sample 177		Float	4	MGR	34383
168	Sample 178		Float	4	MGR	34385

172	Sample 179		Float	4	MGR	34387
176	Sample 180		Float	4	MGR	34389

5.6.5.8 Spectrum (Order 1-25) (GroupID = 0x2008)

Byte Offset	Description	Units	Data type	Size in bytes	Src	Modb @
0	Order 1		Float	4	MGR	34401
4	Order 2		Float	4	MGR	34403
8	Order 3		Float	4	MGR	34405
12	Order 4		Float	4	MGR	34407
16	Order 5		Float	4	MGR	34409
20	Order 6		Float	4	MGR	34411
24	Order 7		Float	4	MGR	34413
28	Order 8		Float	4	MGR	34415
32	Order 9		Float	4	MGR	34417
36	Order 10		Float	4	MGR	34419
40	Order 11		Float	4	MGR	34421
44	Order 12		Float	4	MGR	34423
48	Order 13		Float	4	MGR	34425
52	Order 14		Float	4	MGR	34427
56	Order 15		Float	4	MGR	34429
60	Order 16		Float	4	MGR	34431
64	Order 17		Float	4	MGR	34433
68	Order 18		Float	4	MGR	34435
72	Order 19		Float	4	MGR	34437
76	Order 20		Float	4	MGR	34439
80	Order 21		Float	4	MGR	34441
84	Order 22		Float	4	MGR	34443
88	Order 23		Float	4	MGR	34445
92	Order 24		Float	4	MGR	34447
96	Order 25		Float	4	MGR	34449

5.6.5.9 Spectrum (Order 26-50) (GroupID = 0x2009)

Byte Offset	Description	Units	Data type	Size in bytes	Src	Modb @
0	Order 26		Float	4	MGR	34501
4	Order 27		Float	4	MGR	34503
8	Order 28		Float	4	MGR	34505
12	Order 29		Float	4	MGR	34507
16	Order 30		Float	4	MGR	34509
20	Order 31		Float	4	MGR	34511

24	Order 32		Float	4	MGR	34513
28	Order 33		Float	4	MGR	34515
32	Order 34		Float	4	MGR	34517
36	Order 35		Float	4	MGR	34519
40	Order 36		Float	4	MGR	34521
44	Order 37		Float	4	MGR	34523
48	Order 38		Float	4	MGR	34525
52	Order 39		Float	4	MGR	34527
56	Order 40		Float	4	MGR	34529
60	Order 41		Float	4	MGR	34531
64	Order 42		Float	4	MGR	34533
68	Order 43		Float	4	MGR	34535
72	Order 44		Float	4	MGR	34537
76	Order 45		Float	4	MGR	34539
80	Order 46		Float	4	MGR	34541
84	Order 47		Float	4	MGR	34543
88	Order 48		Float	4	MGR	34545
92	Order 49		Float	4	MGR	34547
96	Order 50		Float	4	MGR	34549

5.6.5.10 LED control (GroupID = 0x2081)

Byte Offset	Description	Units	Data type	Size in bytes	Min value	Max value	Src	Modb @
0	Led frequency Status		Byte	1	0	4	MGR	31101

The ‘LED frequency status’ parameter is defined as follows:

Value	Blinking rate
0	Always Off
1	0.5 second
2	1 second
3	2 seconds
4	Always On

5.6.5.11 PQF settings (GroupID = 0x2082)

Byte Offset	Description	Units	Data type	Size in bytes	Min value	Max value	Src	Modb @
0	RatingModule1	A	Dword	4	0	10000	PQF	33001
4	RatingModule2	A	Dword	4	0	10000	PQF	33003
8	RatingModule3	A	Dword	4	0	10000	PQF	33005
12	RatingModule4	A	Dword	4	0	10000	PQF	33007

16	RatingModule5	A	Dword	4	0	10000	PQF	33009
20	RatingModule6	A	Dword	4	0	10000	PQF	33011
24	RatingModule7	A	Dword	4	0	10000	PQF	33013
28	RatingModule8	A	Dword	4	0	10000	PQF	33015

RatingModule1 to 6: Current rating in Arms of the corresponding unit.

5.7 WAVEFORMS – SPECTRUM - Source IDs

Here are the different curves available on the PQF-Manager and their associated ID:

Source ID	Description	Units	Data type	Size in bytes	# elts	Min value	Max value
0	Voltage Spectrum L1-L2 or N	V	Float	4	50	-9e6	9e6
1	Voltage Spectrum L2-L3 or N	V	Float	4	50	-9e6	9e6
2	Voltage Spectrum L3-L1 or N	V	Float	4	50	-9e6	9e6
3	Line Current Spectrum L1	A	Float	4	50	-9e6	9e6
4	Line Current Spectrum L2	A	Float	4	50	-9e6	9e6
5	Line Current Spectrum L3	A	Float	4	50	-9e6	9e6
6	Filter Current Spectrum L1	A	Float	4	50	-9e6	9e6
7	Filter Current Spectrum L2	A	Float	4	50	-9e6	9e6
8	Filter Current Spectrum L3	A	Float	4	50	-9e6	9e6
9	Line Current Spectrum N	A	Float	4	50	-9e6	9e6
10	Filter Current Spectrum N	A	Float	4	50	-9e6	9e6
11	Samples Voltage L1-L2 or N	V	Float	4	180	-9e6	9e6
12	Samples Voltage L2-L3 or N	V	Float	4	180	-9e6	9e6
13	Samples Voltage L3-L1 or N	V	Float	4	180	-9e6	9e6
14	Samples Line Current In1	A	Float	4	180	-9e6	9e6
15	Samples Line Current In2	A	Float	4	180	-9e6	9e6
16	Samples Line Current In3	A	Float	4	180	-9e6	9e6
17	Samples Filter Current Ip1	A	Float	4	180	-9e6	9e6
18	Samples Filter Current Ip2	A	Float	4	180	-9e6	9e6
19	Samples Filter Current Ip3	A	Float	4	180	-9e6	9e6
20	Samples Line Current InN	A	Float	4	180	-9e6	9e6
21	Samples Filter Current IpN	A	Float	4	180	-9e6	9e6

These IDs will have to be used in the TCP/IP server/client command requesting to gather some curve information.

6 Windows Communication DLL for PQ-Link protocol

6.1 Introduction

This document describes the interface of the Windows Communication DLL.

The interface is heavily based on the protocol described Chapter 4 of this document.

This document will focus on the differences with the protocol. Indeed, to ease the life of DLL user, some code has been added to handle annoying parts of the protocol.

Moreover, to be compatible with Visual Basic 6.0, some types have had to be modified as VB does not support certain types (unsigned 16-bits integer for example).

The DLL allows communicating with the PQF-Manager through:

- a TCP/IP network connection
- the PQF-Manager USB serial interface

6.2 Interface

6.2.1 Introduction

All exported library functions follow some similar syntax:

```
COMMANDCLIENTDLL_API int __stdcall CommandClient_Fct(…);
```

The first item is defined as follows:

```
#ifdef COMMANDCLIENTDLL_EXPORTS
#define COMMANDCLIENTDLL_API __declspec(dllexport)
#else
#define COMMANDCLIENTDLL_API __declspec(dllimport)
#endif
```

The “COMMANDCLIENTDLL_EXPORTS” define is used within the DLL to make sure that the functions get exported. When the DLL is used in an external program, it is not defined and the functions are then imported from the DLL.

All functions return an integer providing an error code to the calling layer.

The “__stdcall” specifier is used to indicate that the calling convention to be used is the standard one.

Special care has been taken so that the function names exported by the DLL are the same as the ones defined above.

WARNING: in the rest of this chapter, “COMMANDCLIENTDLL_EXPORTS” and “__stdcall” have been removed from the documentation to ease the reading. They are naturally always present in the header file.

6.2.2 Opening and Closing

6.2.2.1 CommandClient_Init

This function opens the connection to the PQF-Manager.

It has the following prototype:

```
int CommandClient_Init(char *PQFManagerAddress,
                       int   ConnectionType,
                       void ( __stdcall *Callback_ConnectionReset)(void));
```

The PQFManagerAddress parameter is an IP address in the case of a TCP/IP connection e.g. “192.168.1.40” or a COM port in the case of a serial connection e.g. “COM11”.

The ConnectionType can take two values:

0 or CONNECTION_TYPE_TCPIP for a TCP/IP connection

1 or CONNECTION_TYPE_SERIAL for a serial connection

The Callback_ConnectionReset is a pointer to a function that will be called if a loss of connection with the PQF-Manager is detected.

If one does not wish to use the callback mechanism, this parameter can be set to NULL. Unwanted disconnections can then be detected when any of the DLL function call returns the “RVT_SYS_SOCKET_DISCONNECTED” error code.

When a disconnection is happening, it is necessary to call the clean function to free the PC resources used and try to connect to the PQF-Manager again.

6.2.2.2 CommandClient_Clean

This function closes the connection to the PQF-Manager.

It has the following prototype:

```
int CommandClient_Clean();
```

The closing allows to free resources allocated both on the PC side and on the PQF-Manager side.

6.2.3 Authentication

Please note that the user account covered here are applicative user account handled by the PQF-Manager.

Once the TCP/IP, USB or Modbus connection has been established to the server, the client must authenticate itself.

As the system can be accessed remotely, some basic authentication is put in place. This allows distinguishing users and granting them certain rights to do things.

This authentication is mentioned as the first point of the command layer for the very important reason that no command will be accepted before an authentication is performed. The only exception is for the local administrator user that is automatically detected based on its local connection to the server.

6.2.3.1 CommandClient_Authenticate

This function allows authenticating as a given user on the PQF-Manager.

It has the following prototype:

```
int CommandClient_Authenticate(char          *Login,
                               char          *Password,
                               unsigned char *AccessLevel);
```

The password is here given as a string and transformed by the DLL to be used in the protocol.

The Login and Password parameters are used to authenticate oneself and the AccessLevel is returned to indicate the associated access level.

Access levels

The following access levels are defined:

Level	Name	Description
0x00	Monitor	Very basic user allowing only reading measurements and curves from the device.
0x10	Configurator	More advanced user allowed modifying measurement parameters.
0x40	Administrator	User allowed modifying any parameter of the device and to create and delete users.
0x80	Local application	Special user created for internal use only. It has the same rights as an Administrator.
0xFF	Reserved	Reserved.

6.2.3.2 CommandClient_CreateUser

This function allows creating a new user account on the PQF-Manager.

It has the following prototype:

```
int CommandClient_CreateUser( char          *Login,
                              char          *Password,
                              unsigned char AccessLevel);
```

The password is here given as a string and transformed by the DLL to be used in the protocol.

The AccessLevel parameter specifies what access level to associate with the new user account.

6.2.3.3 CommandClient_DeleteUser

This function allows deleting a user account on the PQF-Manager.

It has the following prototype:

```
int CommandClient_DeleteUser(char *Login);
```

6.2.4 Parameter access

6.2.4.1 CommandClient_GetParameter

This function allows getting a parameter group from the PQF-Manager.

It is restricted to configuration, measurement and information parameters.

The parameter is read from the actual parameters memory.

Note that the parameters that have been set in the shadow memory but not yet applied are not returned when a “Get Parameter” command is issued.

For a reason of transmission efficiency, it is only possible to transfer parameter by groups of parameters; in other words, transmission of individual parameters is not foreseen.

One must have at least Monitor rights to perform this task.

It has the following prototype:

```
int CommandClient_GetParameter(int            ParamGroupID,
                                unsigned char *Value,
                                int           *Size);
```

The ParamGroupID is specifying which parameter group to get.

The Value and Size parameters are pointers to the buffer where the parameter group values will be stored and to the actual size of this parameter group.

6.2.4.2 CommandClient_SetParameter

This function allows setting a parameter group in the PQF-Manager.

It is restricted to configuration parameters.

The modification of application specific parameters is applied in a shadow memory. Parameter changes will only be copied to parameters memory after an “Apply Parameter Changes” command is issued.

The modification of universal parameters is applied directly i.e. no “Apply Parameter Changes” command is required.

For a reason of transmission efficiency, it is only possible to transfer parameter by groups of parameters; in other words, transmission of individual parameters is not foreseen.

One must have at least Configurator rights to set application specific parameters and at least administrator rights to set universal parameters.

It has the following prototype:

```
int CommandClient_SetParameter(int            ParamGroupID,
                                unsigned char *Value,
                                int            Size);
```

The ParamGroupID is specifying which parameter group to set.

The Value parameter is a pointer to the buffer where the data to write is stored and the Size is specifying the amount of data to be written.

6.2.4.3 CommandClient_ApplyParameterChanges

This function applies the shadowed parameter changes in the parameter memory of the PQF-Manager.

One must have at least Monitor rights to perform this task.

It has the following prototype:

```
int CommandClient_ApplyParameterChanges();
```

6.2.4.4 CommandClient_ConvertRVTtoVB

This helper function allows retrieving a single parameter value from a parameter group.

It has the following prototype:

```
int CommandClient_ConvertRVTtoVB(unsigned char *ParamGroupValue,
                                     int            Offset,
                                       unsigned char  ValueType,
                                     char          *OutputString,
                                     int           *OutputStringSize);
```

The ParamGroupValue are the parameter group value as returned by the CommandClient_GetParameter command. This is a buffer of unsigned char values.

The Offset and ValueType are specifying where to find the wanted data and what is the type of the wanted value.

The OutputString and OutputStringSize are specifying where the string should be placed and what its size is.

One should make sure that there's enough space in the output buffer for the string. Currently, the biggest parameters defined for the PQF-Manager are 180 bytes big and are of the string type. Numbers are converted to much smaller strings.

6.2.4.5 CommandClient_ConvertVBtoRVT

This helper function allows setting a single parameter value into a parameter group.

It has the following prototype:

```
int CommandClient_ConvertVBtoRVT(char          *InputString,
                                     unsigned char *ParamGroupValue,
                                     int            Offset,
                                     unsigned char  ValueType);
```

The InputString is the value to be converted and stored in the parameter group.

The ParamGroupValue is the parameter group value as returned by the CommandClient_GetParameter command. This is a buffer of unsigned char values.

The Offset is specifying where to store the converted data and the ValueType is specifying what type of data should be written to the parameter group.

One should make sure that the Offset and ValueType will not cause writing out of the ParamGroupValue buffer.

6.2.5 Curve access

6.2.5.1 CommandClient_RequestCurveEx (future use)

This function makes a request to get a curve available.

Wanted curve is identified by their Source ID.

One must have at least Monitor rights to perform this task.

The curves will be too big to be retrieved with a single command. As a consequence, the data will be split in chunk. The server is specifying the size of those chunks and is warning the client of how many of those chunks will need to be transferred to get the complete curve.

It has the following prototype:

```
int CommandClient_RequestCurveEx(int            SourceID,
                                  unsigned char  NumberOfPeriods,
                                         unsigned char  TriggerEnable,
                                  int            TriggerSource_ID,
                                         int            TriggerSource_Offset,
                                  unsigned char  TriggerComparator,
                                  unsigned char  TriggerValue_Type,
                                         char          *TriggerValue_ValueString,
                                         int           *CurveDescriptor);
```

The CurveDescriptor is made out of the NumberOfChunks and ChunkSize as defined in the protocol.

The Trigger Value to be used is here provided as a string and is converted by the DLL to the type specified by TriggerValue_Type.

It can take some time between the moment of the request and the time where the first data gets available.

IMPORTANT NOTE:

- Number of periods is “1” by default
- Trigger function is reserved for future use : “0” by default

6.2.5.2 CommandClient_RequestCurve

This is a simplified version of the CommandClient_RequestCurveEx command where triggering is disabled.

It has the following prototype:

```
int CommandClient_RequestCurve(int            SourceID,
                                        unsigned char  NumberOfPeriods,
                                        int           *CurveDescriptor);
```

6.2.5.3 CommandClient_ReleaseCurve

This function is telling the PQF-Manager to stop getting data for the given curve.

If the curve was requested by a single client, the curve is not made available anymore for the client and the slot is freed.

If the curve was requested by multiple clients, the freeing is only made when the last client releases the curve.

Note that the effect of this command may not be immediate.

One must have at least Monitor rights to perform this task.

It has the following prototype:

```
int CommandClient_ReleaseCurve(int SourceID);
```

6.2.5.4 CommandClient_GetCurve

This function is getting curve data from the PQF-Manager.

It has the following prototype:

```
int CommandClient_GetCurve(int            SourceID,
                                 int            CurveDescriptor,
                                 unsigned char *Value,
                                 int           *Size);
```

The SourceID and CurveDescriptor are identifying the curve.

The Value is a buffer where the curve will be stored and the Size is the actual size of the curve returned. One should make sure that the buffer is big enough to contain the curve.

The curves are transferred by chunks.

This function is taking care of downloading the number of chunks required to get the complete curve. It is also taking care that all chunks returned do belong to the same curve set.

One must have at least Monitor rights to perform this task.

6.2.5.5 CommandClient_GetCurveIDs

This function is getting all the SourceIDs currently in use by the DLL.

One must have at least Monitor rights to perform this task.

It has the following prototype:

```
int CommandClient_GetCurveIDs(int *SourceIDs,
                                  int *Size);
```

The Size is specifying how many SourceIDs have been returned and copied to the location pointed by the SourceIDs pointer. One must ensure that the location is big enough to contain the maximum number of curves allowed by the system.

6.2.5.6 CommandClient_GetCurveCharacteristics

This function is getting the characteristics of a given curve.

It has the following prototype:

```
int CommandClient_GetCurveCharacteristics(int            SourceID,
                                          unsigned char *NumberOfPeriods,
                                          unsigned char *TriggerEnable,
                                          int           *TriggerSource_ID,
                                          int           *TriggerSource_Offset,
                                          unsigned char *TriggerComparator,
                                          unsigned char *TriggerValue_Type,
                                          char   *TriggerValue_ValueString,
                                          int    *TriggerValue_ValueStringSize,
                                          int           *CurveDescriptor);
```

This function can be seen as a way to get back the parameters that were passed at the moment of requesting the curve.

This function along with the GetCurveIDs function allow for an easy re-populating of the user interface. The interface does not have to store the characteristics of all the curves it currently manages; it can just ask it back to the PQF-Manager.

One must have at least Monitor rights to perform this task.

6.2.6 Reset

This function allows to remotely restarting the PQF-Manager.

It has the following prototype:

```
int CommandClient_Reset();
```

After calling this command, the connection should be closed using the CommandClient_Clean function and should be re-established again.

One needs at least Configurator rights to perform this task.

6.3 Important considerations

6.3.1 Visual Basic 6.0 support

This DLL has been built with support for VB6.0 in mind.

6.3.2 Multi-threading

The DLL is not coded for multi-threaded application.

The first consequence is that all calls to the DLL should be called from a single thread. Calling from different thread could be possible but protection should then be implemented outside of the DLL.

The second consequence is that all calls to the DLL are blocking. In usual cases it is not a problem but when the connection gets lost for example, it could take a short amount of time to return from a called function.

The only alternative is to go for non-blocking behaviour but then the programming of the application will become more complex as a request issued would not have a direct answer with data to process but the answer would come at a later asynchronous stage.

6.3.3 Sequence of actions

The following sequence of action should be followed when using this DLL to communicate with a PQF-Manager:

1. Connect to the PQF-Manager
2. Authenticate on the PQF-Manager
3. Perform wanted actions (Get / Set parameters , Get curves ,…)
4. Go back to step 3. while the connection should be active
5. Disconnect from the PQF-Manager

One should not forget the authentication phase otherwise no subsequent action will be possible and the connection will be closed by the PQF-Manager. Moreover, the authentified user’s access level will make it possible to perform certain actions or not.

Unwanted TCP/IP or serial disconnections with the PQF-Manager can be monitored through the callback provided at initialization time or through the returning o of the “RVT_SYS_SOCKET_DISCONNECTED” code. When that is happening, it is necessary to call the clean function and try to connect to the PQF-Manager again.

6.4 Error codes

The following error codes can be returned by the DLL:

Error code define	Value
RVT_SUCCESS	0
RVT_SYS_MEMORY	1
RVT_SYS_TASK_CREATE	2
RVT_SYS_SEMAPHORE_CREATE	3
RVT_SYS_SEMAPHORE_FAILURE	4
RVT_SYS_PERIPHERAL_IO	5
RVT_MEMORY_CORRUPT	6
RVT_SYS_SOCKET_OPEN	10
RVT_SYS_SOCKET_BIND	11
RVT_SYS_SOCKET_LISTEN	12
RVT_SYS_SOCKET_CONNECT	13
RVT_SYS_SOCKET_DISCONNECTED	14
RVT_SYS_INVALID_OBJECT	16
RVT_SYS_BUFFER_OVERFLOW	17
RVT_SYS_BUFFER_TOO_SMALL	18
RVT_NO_MORE_OBJECT_ALLOWED	20

RVT_OBJECT_NOT_FOUND	21
RVT_OBJECT_OPEN_ERROR	22
RVT_OBJECT_IO_ERROR	23
RVT_NO_MORE_OBJECT	24
RVT_OBJECT_DISABLED	25
RVT_OBJECT_ALREADY_USED	26
RVT_PKT_MALFUNCTION	29
RVT_SECL_UNKNOWN_LOGIN	30
RVT_SECL_INVALID_LOGIN	31
RVT_SECL_LOGIN_FAILURE	32
RVT_SECL_AUTH_FAILURE	33
RVT_SECL_TOO_MANY_USERS	34
RVT_SECL_NOT_LOGGED_IN	35
RVT_SECL_NOT_ENOUGH_RIGHTS	36
RVT_PMDB_INVALID_ID	40
RVT_PMDB_UNAVAILABLE	41
RVT_PMDB_IO_FAILURE	42
RVT_PMDB_INVALID_NVRAM	43
RVT_PMDB_TYPE_MISMATCH	44
RVT_PMDB_OPERATION_DENIED	45
RVT_PMDB_INVALID_SIZE	46
RVT_PMDB_INVALID_CONTENT	47
RVT_FCT_INVALID_PARAMETER	50
RVT_FCT_NEEDS_INITIALIZATION	51
RVT_FCT_OPERATION_FAILED	52
RVT_FCT_OPERATION_DENIED	53
RVT_FCT_OPERATION_TIMEDOUT	54
RVT_FCT_INVALID_RESPONSE	55
RVT_CMD_UNKNOWN	60
RVT_CMD_UNSUPPORTED	61
RVT_CMD_MISMATCH	62
RVT_LOG_SYSLOG_INVALID_ADDR	110
RVT_LOG_OPERATION_DENIED	111

6.5 Example codes

6.5.1 Visual Basic 6.0 project

The project is built around a single form for the user interface and a module to define the interface to the DLL.

The form code contains necessary initialization steps to make the DLL available from another directory. It obviously also contain the code for the user interface and associated calls to the DLL functions.

The module contains the necessary constant and function declarations. It also contains some helper functions to be used when using helper functions to convert parameter; these functions take care of memory allocation for the returned strings.

7 Appendices

A1 List of abbreviations

Abbreviations	Meaning
ASCII	American Standard Code for Information Interchange
Baud rate	Unit for measuring transmission speed in bits/s
Bit	A binary digit, representing a one or zero
Bus	An electrical circuit over which data is transmitted
Byte	A whole number value represented by eight bits (0 to 255)
Chassis or Chassis Ground	A connection to an electrically conductive housing or frame of a device. It may or may not be connected to Earth Ground.
Coil	The telegram structure for Modbus transmission is implemented in registers (WORD) or coils (BOOL). A coil may be either 8 or 16 bits in length.
Common	The voltage reference point of a circuit. It may or may not be connected to earth ground, though it is generally assumed to be at zero volts, unless otherwise indicated. In floating circuits, the common is sometimes at a relatively high potential. This term is sometimes used interchangeably with the term “Ground” or GND
CRC	Cyclic Redundancy Check. Complex error checking on a message block.
CTS	ClearToSend hardware handshaking signal. Used with RequestToSend.
DHCP	Is an autoconfiguration protocol used on IP networks. Computers that are connected to IP networks must be configured before they can communicate with other computers on the network. DHCP allows a computer to be configured automatically, eliminating the need for intervention by a network administrator
DLL	Dynamic-link library is Microsoft’s implementation of the shared library concept in the Microsoft Windows and OS/2 operating systems. These libraries usually have the file extension DLL, OCX (for libraries containing ActiveX controls), or DRV (for legacy system drivers). The file formats for DLLs are the same as for Windows EXE files — that is, Portable Executable (PE) for 32-bit and 64-bit Windows. As with EXEs, DLLs can contain code, data, and resources, in any combination.
Earth or Earth Ground	Global zero voltage reference point. Physical connection is made to the earth through a grounding rod, water pipe or other reliable connection.
Ethernet	Is a family of frame-based computer networking technologies for local area networks (LANs). It defines a number of wiring and signalling standards for the Physical layer of the OSI networking model as well as a common addressing format and Media Access Control at Data Link Layer.
Ground Voltage reference point of a circuit.	It may or may not be connected to earth ground, though it is generally assumed to be at zero volts. Sometimes used interchangeably with the term “Common”.
Handshaking	Method of data flow control for serial communications
Hexadecimal or HEX	A number system using a decimal 16 as its base. A single digit number in HEX ranges from 0 to 15, represented by 0 to 9 and A to F.

HMI	Human-Machine Interface (formerly MMI)
Industrial^{IT}	Umbrella concept for ABB's vision for enterprise automation.
Industrial^{IT} Architecture	The architecture of the Industrial IT system. The architecture defines how the system is built, in terms of basic concepts, underlying technologies, system topology, modularity, and mechanisms for interaction between different parts of the system. It also defines concepts, rules, and guidelines that a component must comply with in order to fit in the Industrial IT system. A central feature of the IIT architecture is that information and functions are centred on Aspect Objects.
Industrial^{IT} Enabled	A product that is Industrial IT enabled has been verified according to the process of Industrial IT certification. It has the right to use the "Industrial IT enabled" symbol.
IP address	An Internet Protocol (IP) address is a numerical label that is assigned to devices participating in a computer network that uses the Internet Protocol for communication between its nodes. An IP address serves two principal functions: host or network interface identification and location addressing.
Loopback	A test used for checking functionality of a serial port, utilizing a test plug that connects send, receive and handshaking signals
Long Integer	Analog value consisting of two consecutive 16-bit registers
LRC	Longitudinal Redundancy Check
MAC address	In computer networking, a Media Access Control address (MAC address) is a unique identifier assigned to most network adapters or network interface cards (NICs) by the manufacturer for identification, and used in the Media Access Control protocol sub-layer. If assigned by the manufacturer, a MAC address usually encodes the manufacturer's registered identification number.
Measurement	A measurement is a value computed by the controller through its analog and digital inputs. Measurements can be read from the PQF-Manager front plate, or through the Modbus protocol.
Modbus adapter	It is an optional small interface module through which the PQF-Manager is connected to an external Modbus serial communication bus. It performs an optical to RS485 conversion. The communication with the Modbus adapter is activated with a PQF-Manager parameter.
OPC	OLE™ for Process Control. OPC is Plug-n-Play in the field of Automation and HMI. OLE™ for Process Control (OPC™) is the most standard way for connecting hardware and data devices with HMI client applications. OPC is a concept agreed upon by a committee of members from the OPC foundation. Most automation companies in the market place including ABB are members of this foundation. OPC uses state-of-the art technologies like COM, DCOM, ActiveX of Microsoft and makes development and programming easier. In the OPC world, there are two major types of applications: OPC Servers and OPC Clients.
OPC Servers	OPC Server applications are used to collect data from the data sources like hardware devices. At the bottom level, the servers are mainly for reading inputs and writing outputs of the data sources. At the upper level, the servers make the data available in a standard way to the OPC client applications.

OPC Clients	The OPC Client applications can communicate directly with the OPC servers and get the data. This way OPC enhances the interface between client and server applications by providing a standard mechanism to communicate data from a data source to any client application.
Parameter	A parameter is an operating data for the controller. Parameters can be read and programmed with the PQF-Manager front plate, or through the Modbus protocol.
Parity	Simple method of data error checking performed at the byte level. May be user-specified as Odd, Even or None with most equipment and software.
PC	Personal Computer
Power^{IT} Power Factor Controller	Microprocessor based controller from the ABB industrial controller range. The Power^{IT} Power Factor Controller RVT is intended to switch capacitor in order to compensate the power factor of the electrical network.
PQF-Manager	User interface of the ABB's PQF active filters product range
Receive	Incoming communication signal. (Rx)
RTS	RequestToSend hardware handshaking signal. Used with ClearToSend.
RVT	see Power^{IT} Power Factor Controller RVT
Rx	See Receive
PLC	Programmable Logic Controller
RTS	Request To Send
RTU	Remote Terminal Unit
Time-out	Parameter specifying the max. wait time in ms. Waiting for a response in the range 0..10000 ms.
Signed Integer	Whole number value represented by 16 bits (-32768 to 32767)
SMTP	Simple Mail Transfer Protocol (SMTP) is an Internet standard for electronic mail (e-mail) transmission across Internet Protocol (IP) networks. SMTP is specified for outgoing mail transport. While electronic mail servers and other mail transfer agents use SMTP to send and receive mail messages, user-level client mail applications typically use only SMTP for sending messages to a mail server for relaying. For receiving messages, client applications usually use either the Post Office Protocol (POP) or the Internet Message Access Protocol (IMAP) or a proprietary system (such as Microsoft Exchange or Lotus Notes/Domino) to access their mail box accounts on a mail server.
SNMP	Simple Network Management Protocol (SNMP) is a UDP-based network protocol. It is used mostly in network management systems to monitor network-attached devices for conditions that warrant administrative attention. SNMP is a component of the Internet Protocol Suite. It consists of a set of standards for network management, including an application layer protocol, a database schema, and a set of data objects.
TCP/IP	The Internet Protocol Suite is the set of communications protocols used for the Internet and other similar networks. It is commonly also known as TCP/IP, named from two of the most important protocols in it: the Transmission Control Protocol (TCP) and the Internet Protocol (IP), which were the first two networking protocols defined in this standard.
Transmit	Outgoing communication signal. (Tx)

Tri-State	The ability of a communications transmitter to turn its circuitry off, reducing the load on the network
Tx	see Transmit
Unsigned Integer	Positive whole number value represented by 16 bits (0 to 65535)
USB	Universal Serial Bus is a specification to establish communication between devices and a host controller (usually personal computers).
Word	A group of 16 bits
Xon/Xoff	Software implementation of data flow control

A2 References

- RS485 adapter installation and start-up guide
- RS485 adapter user guide
- PQF Installation and Operation Instructions manuals
- ABB Power Quality Link manual
- Modicon Modbus Protocol Reference Guide (PI–MBUS–300 Rev. J).

A3 Description of open ports

Port	State
502/TCP	Open
4250/TCP	Open
10022/TCP	Open

A4 Cyber Security Disclaimer note

This product is designed to be connected to and to communicate information and data via a network interface. It is User's sole responsibility to provide and continuously ensure a secure connection between the product and User's network or any other network (as the case may be). The User shall establish and maintain any appropriate measures (such as but not limited to the installation of

firewalls, application of authentication measures, encryption of data, installation of anti-virus programs, etc) to protect the product, the network, its system and the interface against any kind of security breaches, unauthorized access, interference, intrusion, leakage and/or theft of data or information.

ABB Ltd and its affiliates are not liable for damages and/or losses related to such security breaches, any unauthorized access, interference, intrusion, leakage and/or theft of data or information.

Please note that an ssh account exists for maintenance & development purposes.

—

s.a. ABB n.v.
Power Quality Products
Avenue Centrale 10
Z.I. Jumet
B-6040 Charleroi, Belgium
Phone: +32(0) 71 250 811
Fax: +32 (0) 71 344 007
E-Mail:
Marketing: power.quality@be.abb.com
Service: jumet.services@be.abb.com

http://new.abb.com/high-voltage/capacitors/lv

Additional information
We reserve the right to make technical changes or modify the contents of this document without prior notice. With regard to purchase orders, the agreed particulars shall prevail. ABB does not accept any responsibility whatsoever for potential errors or possible lack of information in this document.
We reserve all rights in this document and in the subject matter and illustrations contained therein. Any reproduction, disclosure to third parties or utilization of its contents – in whole or in parts – is forbidden without prior written consent of ABB.