TECHNICAL NOTE

DIRIS Digiware Communication

When **energy** matters

socomec
Innovative Power Solutions

Summary

Introduction 4

Operation and main architecture of DIRIS Digiware communication and M/D gateway/screen....5

Digiware fieldbus (Modbus RTU)....6

LAN/Ethernet network (Modbus TCP)....7

Preferred limitations and approaches to Modbus RTU/TCP communication....8

The response time for a Modbus RTU request will depend on 9

the overall response time "Modbus TCP response time + other elements" 10

Examples....11

Impact of communication speed and pause time between each question/reply and the next cycle....11

Impact of meter/PMD configuration and amount of data to be monitored 12

Impact of network traffic (other than Modbus TCP IP of Socomec devices, but also email, web surfing, YouTube browsing, etc.) 14

Impact of configuring "retries/timeout" setting 15

Other communication and cybersecurity protocols....16

Diagnostics 17

Identifying the nature of the communication failure 17

Fault diagnostics 17

Summary and Socomec support....18

Glossary....19

Introduction

The energy revolution and the digital revolution have arrived.
These two trends mean that the connectivity of electrical monitoring systems will inevitably increase over time.
By being aware of these trends and developments, and by fostering our expertise not only in power switching, protection and energy control, but increasingly in automation, monitoring and IT communication systems that are now inherent in these solutions, Socomec will continue to support its customers in these present and future challenges.

This technical note is intended to describe how internal/external communication works in a control, monitoring and metering system in a DIRIS Digiware electrical facility.
By extension, this also applies to any communicating ISOM Digiware, COUNTIS, DIRIS A and ATyS connected system.

It also sets out the rules to determine the bandwidths and interrogation frequencies, and to adapt the definition of the system to the need for true communication.

Operation and main architecture of Diris Digiware

An all-in-one monitoring system can be mapped as follows:

- DIRIS Digiware monitoring systems
- a TCP/IP Ethernet communication network
- a monitoring system that centralises all the data

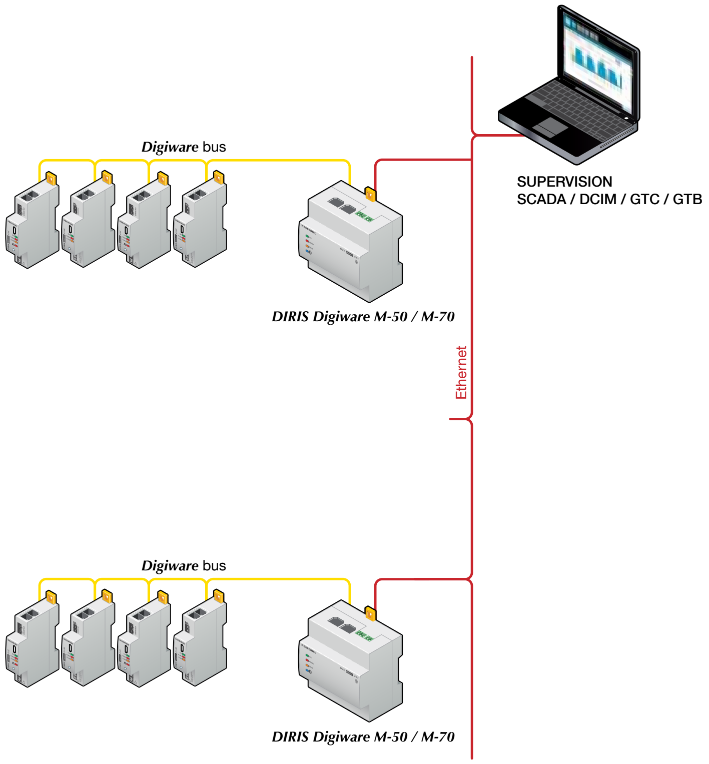


Fig. 1 – Power Monitoring DIRIS Digiware architecture.

A DIRIS Digiware system installed in an electrical panel or in a PDU system will consist of the following elements:

- A main gateway and/or screen: DIRIS Digiware M/D.
  This element acts as a Modbus RTU gateway to Modbus TCP.
- Voltage and current monitoring modules: DIRIS Digiware Uxx/Ixx/Sxx among others
  These modules communicate via the Digiware Bus, which embeds, among other things, the Modbus RTU communication transmission between the screen and the main gateway. The Modbus RTU is embedded in the Yellow RJ45 Digiware bus, which also transmits the digital voltage (patented technology) and the power supply for the modules.

Fig. 2 –.

A LAN Ethernet network on which Modbus TCP exchanges take place with the monitoring/SCADA/DCIM/BMS system acts differently from the Modbus RTU fieldbus on which Modbus exchanges take place downstream in the Digiware section. The main differences are shown below.

Digiware fieldbus (Modbus RTU):

- communication between Digiware system devices via Modbus RTU
- works in master-slave mode
- the gateway/screen is the Modbus RTU master
- the current and voltage monitoring modules DIRIS Digiware Uxx/Ixx/Sxx are the slaves

According to the Modbus RTU standard, the gateway/screen interrogates 1 monitoring module, waits for its reply, interrogates the 2nd monitoring module, waits for its reply, interrogates the 3rd monitoring module, waits for its reply, and so on.

LAN/Ethernet network (TCP Modbus):

Principle:

- Modbus embedded via Ethernet TCP/IP
- Operates in client/server mode
- The gateway/screen is the Modbus TCP to Modbus RTU interface

In this technical note, we will mainly refer to the Modbus TCP which links to the Modbus RTU downstream of the gateway/screen.

There are other communication protocols available, however, which will be discussed a little later and more succinctly in this note.

At TCP/IP level, it is possible to send several questions simultaneously and/or asynchronously without waiting for the reply to the question each time.
For example, it is possible for the monitoring unit to send 10 questions at the same time and then receive the 10 replies.
It is also possible for a monitoring unit to send 5 questions, then 5 more questions the next second, while still waiting for the replies to the first 5 questions.

Let's take an example using 4 DIRIS Digiware meters to interrogate. The monitoring unit sends the 4 questions at the same time at t0 in Modbus TCP. The communication gateway still sends and receives the questions to the slaves A,B,C,D in sequence one after the other:

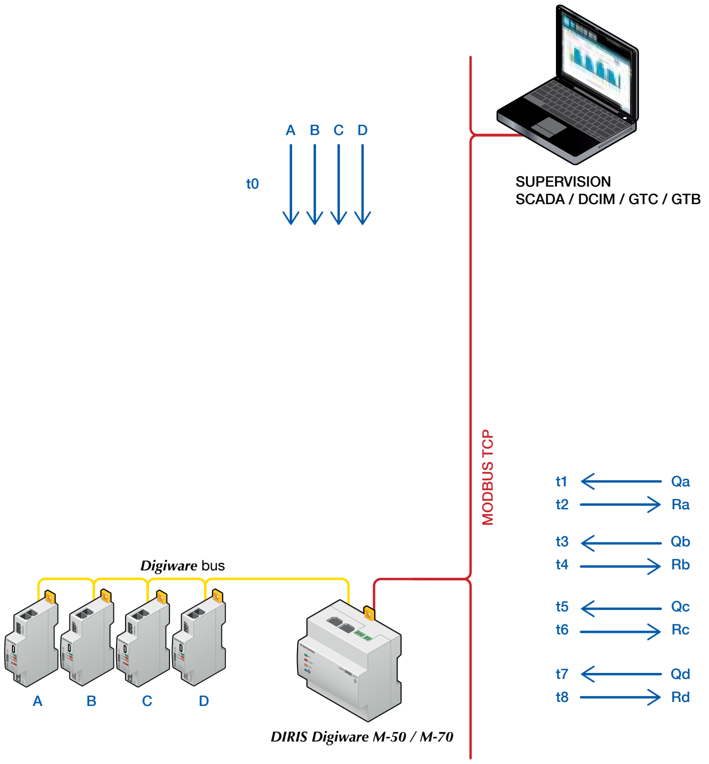


Fig. 4 – Power monitoring DIRIS Digiware architecture communication principle.

This results in a stack of questions at the gateway or screen level, ordered in a sequence determined by a Modbus TCP / IP Client. So, the screen will first stack the questions from the client(s) (in this case, 32 questions in the buffer), then it translates the questions into Modbus RTU format.
Each question received by the gateway or the Digiware screen corresponds to a question sent from the gateway or screen to a Modbus RTU slave (modules I, U, I / O, etc.).

This is the principle of a communication gateway that allows one input communication protocol and one output communication protocol.

Preferred limitations and approaches to Modbus TCP/RTU communication

A DIRIS Digiware D/M screen or gateway can handle 32 TCP sockets in parallel.

A socket is a connection. By way of analogy, if you open 10 web pages on your browser, you open 10 http sockets in parallel.

It is therefore possible for the monitoring system (master/client) to interrogate 32 Digiware modules (slaves) at the same time from the Ethernet monitoring system.
Similarly, you could connect 32 independent monitoring units, each using only one socket.

According to the Modbus RTU rules, however, these requests will be processed one after the other on the serial bus:
- one question to the first Digiware module
- one reply from the first Digiware module
- one question to the second Digiware module
- one reply from the second Digiware module
- and so on: Question/Reply/Question/Reply/Question/Reply

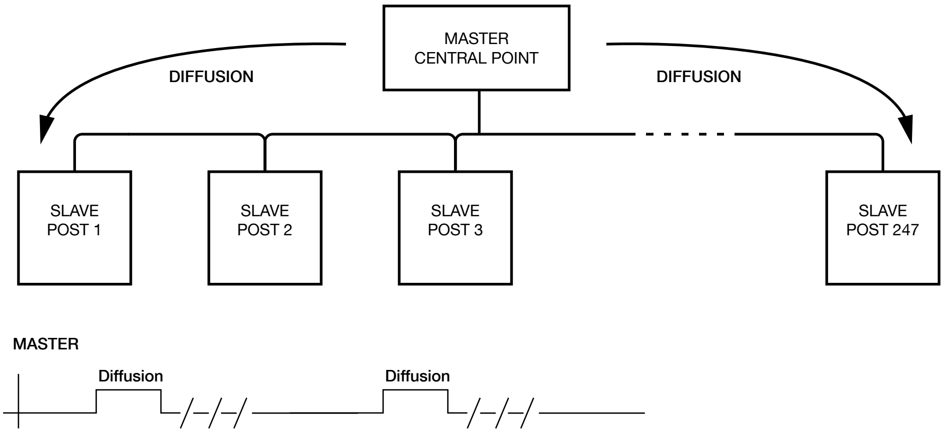


Fig. 5 – RS485 master transmission principle.

We can therefore assimilate these 32 TCP sockets to a "queue" of 32 questions located on the screen/gateway side.

The bandwidth limit will therefore be at the Modbus RTU level downstream.

Below we can see which parameters have impacts on:

- the response time for a Modbus RTU request
- the overall cycle time of an all-in-one DIRIS Digiware system

The response time for a Modbus RTU request will depend on:

- **The communication speed**
  (38400 bds by default on Digiware but configurable up to 500 kbs)

- **The configuration of the turnabout delay**
  Modbus pause time between the end of a reply and the start of the next question on the Digiware internal bus. 10ms by default

- **Retry/Timeout at Modbus RTU level**
  These terms are detailed in the glossary, but the timeout is the waiting time of the gateway/screen if a DIRIS Digiware module does not respond to a question. After this time, the question will be sent back to the DIRIS Digiware monitoring module to try communicating again with the DIRIS Digiware module. The number of these attempts before considering the module permanently unreachable corresponds to the number of "retries".

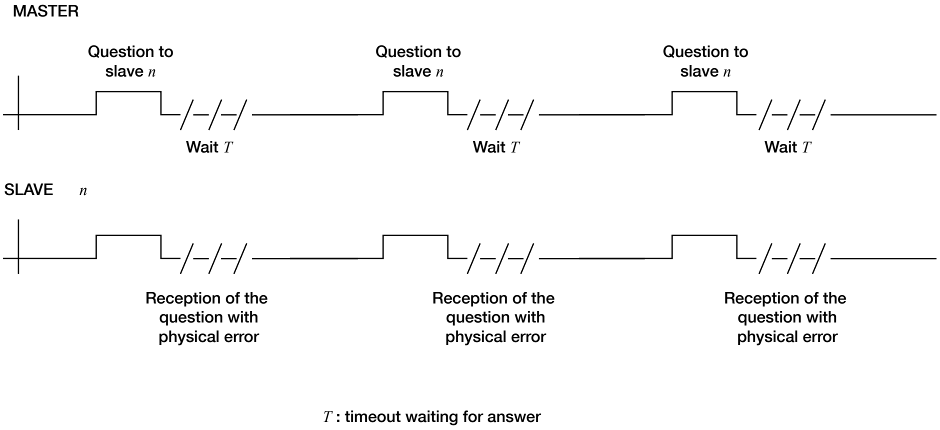

MASTER
Question to slave n
Question to slave n
Question to slave n
Wait T
Wait T
Wait T
SLAVE n
Reception of the question with physical error
Reception of the question with physical error
Reception of the question with physical error
T : timeout waiting for answer

Fig. 6 – RS485 slave timeout / retry principle.

- **The amount and frequency of the requested data**
  (if the interrogation requests 1 reading or 20 readings such as currents, voltages, power, etc.) The 'refresh rate' is also important.

- **Internal use of the system and the queue status at time t:**
  To view the values on the DIRIS Digiware D screen directly, to log and view on the WEBVIEW-M embedded webserver; requests that will be added to the requests of the monitoring unit.
- An interrogation that arrives at time t can therefore be processed right away or can be second or third in the queue and will be processed after 10/20/30 ms.

- **Banning an absent device to avoid overloading the network with unsuitable frames:**
  To protect against attacks and/or communication requests that are not compatible (contact from a non-existent meter or a request from a non-existent Modbus register on an existing meter), the Dxx/Mxx have a built-in banning mechanism. When they receive a number of non-compliant Modbus TCP/IP frames (3 in a row), this allows them to not transmit them downstream on the Digiware bus, to avoid overloading it. Like all the elements described in this technical note, this is a configurable parameter that can be enabled/disabled.

The overall response time: the Modbus RTU response time + all of the following:

The response times at the gateway - which are related to the data traffic on the Ethernet network this time - are added to this:

- **Total number of requests required: the number and configuration of DIRIS Digiware monitoring modules**
  For example, if an I-30 module is configured as three-phase, there will be a kWh value to read, whereas if it is configured as 3x single-phase, it will be the equivalent of 3 meters which means there will be 3 kWh values to read.
  All this is multiplied by the number of measuring modules that make up the system.
  The number of requests will also depend on the ability to interrogate multiple measurements in a single Modbus request when the registers are contiguous, follow each other and are in the same table / Modbus table.

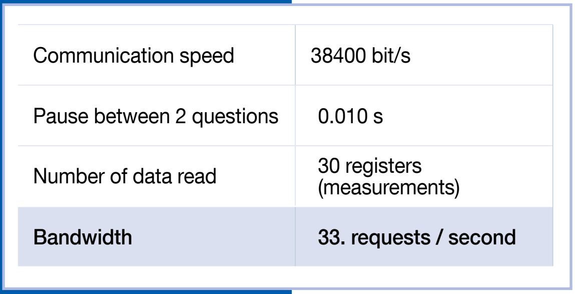
- **All traffic present on the Ethernet network (email exchange, data, web surfing, etc.):**
  The global use of the Ethernet network, a network not dedicated to Modbus TCP monitoring of electrical readings and meters, can also be the cause of "traffic jams" on the Ethernet network and justify overly long response times and/or communication errors.
  Socomec D/M screens/gateways are sized to support TCP/IP traffic levels not affecting them, to a maximum of 6000 packets per second.
  If the network becomes more loaded than that, it might make sense to use a TCP/IP network dedicated to electrical monitoring.
  Local use of the screen (measurements displayed on the panel door) or use of the WEBVIEW webserver also occupy bandwidth at Digiware communication level. The bandwidth limits below take into account the entire bandwidth, not only that dedicated to the monitoring system. To optimise communication for monitoring as much as possible, favour a blind system without a webserver (DIRIS Digiware M-50).
  You can also completely disable the datalogging/storage function of the gateway to free up the bandwidth for monitoring as much as possible.

- **Synchronising the config retry/timeout with those of the monitoring:**
  When a question arrives via Ethernet on a D50 display for example, it will transfer it to the Digiware bus.
  If the slave has not replied after XX seconds, the D50 will try NN times to resend the question.
  XX is the timeout and NN is the number of retries

Examples

Below are some examples of the impacts that the elements listed above can have.

Impact of communication speed and pause time between each question/reply and the next cycle:



Communication speed	38400 bit/s
Pause between 2 questions	0.010 s
Number of data read	30 registers (measurements)
Bandwidth	**33. requests / second**

Fig. 7 – Response time bandwidth calculation example #1.

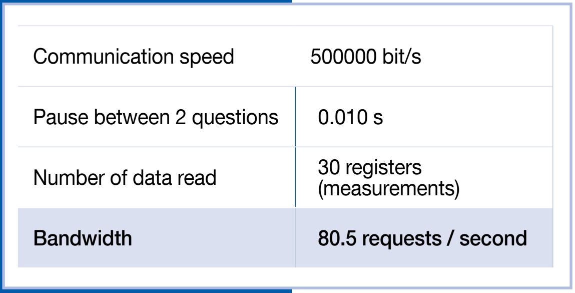
The communication speed (38400 bit/s) can be set up to 500 kbit/s on the DIRIS Digiware system. This does not multiply the bandwidth proportionally, as you need to take into account the pause times between each request (turnabout delay 10 ms).

For example, if we have 32 monitoring modules to interrogate 1x every 500 ms, it would take a bandwidth of 64 resquests per second (32 requests per 500 ms).

Below is the calculation of the default bandwidth (number of questions per second that a screen/gateway can process on the DIRIS Digiware downstream bus). This simplified approach can be documented in detail if required (we have hidden some parameters that allow the calculation, to simplify understanding):

This means that the default speed allows 33.3 requests to be processed every second.

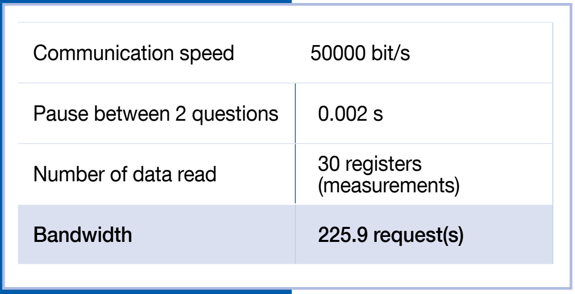
So, in this example, we would need to increase the speed.



Communication speed	500000 bit/s
Pause between 2 questions	0.010 s
Number of data read	30 registers (measurements)
Bandwidth	**80.5 requests / second**

Fig. 8 – Response time bandwidth calculation example #2.

Another example:
System consisting of a gateway or screen and 32 DIRIS Digiware I60 modules configured as 6x single phase.



Communication speed	50000 bit/s
Pause between 2 questions	0.002 s
Number of data read	30 registers (measurements)
Bandwidth	**225.9 request(s)**

Fig. 9 – Response time bandwidth calculation example #3.

We may therefore need 32x I60 modules that each have 6 single-phase meters.
This means 6 questions will need to be sent to the I-60 rather than just one (see following section).
We want to read values every second, so we will need a bandwidth of 32x6 = 182 requests per second.

In this case we need to switch the turnabout delay to 2 ms, which gives us 225.9 requests per second (greater than our need of 182 requests per second).

Notes:

This bandwidth only takes into account the intrinsic limit of the DIRIS Digiware system.
Other delays can be induced by the Ethernet network (existing traffic and/or number of intermediate switches/firewalls/routers) or the monitoring PC, which can have the impact of reducing the bandwidth by up to 30%.

A gateway/screen has a maximum of 32 sockets, but in this last example we need to extend it to 225.9 requests per second every second.
Let's say we send 22.59 requests every 100 ms, rather than 225 requests at the same time every 1 second (because the queue/sockets are limited to 32, we would "ditch" 225-32 = 193 requests).

Impact of meter/PMD configuration and amount of data to be monitored:

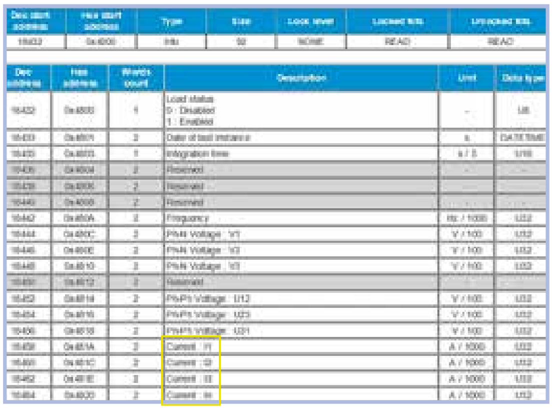
If we take the example of an I30 configured in 2 different ways:

It takes 2 requests to read this information

Configured as three-phase:
- Current/voltage/power reading => 1 request because the registers are concatenated
- Energy reading => 1 request

It therefore takes 6 requests to read this information (3x more)

Configured as 3x single phase:
- Reading currents/voltages/power => 3 requests because the registers are contiguous but independent for the 3 loads
- SCREENSHOTS OF BOTH
- Independent energy reading for 3 loads => 3 requests



Dec start address	Hex start address	Type	Size	Lock level	Locked R/W	Unlocked R/W
18432	0x4800	Info	50	NONE	READ	READ

Dec address	Hex address	Words count	Description	Unit	Data type
18432	0x4800	1	Load status 0 : Disabled 1 : Enabled	-	U8
18433	0x4801	2	Date of last instance	s	DATETIME
18435	0x4803	1	Integration time	s / 3	U16
18436	0x4804	2	Reserved	-	-
18438	0x4806	2	Reserved	-	-
18440	0x4808	2	Reserved	-	-
18442	0x480A	2	Frequency	Hz / 1000	U32
18444	0x480C	2	Ph-N Voltage : V1	V / 100	U32
18446	0x480E	2	Ph-N Voltage : V2	V / 100	U32
18448	0x4810	2	Ph-N Voltage : V3	V / 100	U32
18450	0x4812	2	Reserved	-	-
18452	0x4814	2	Ph-Ph Voltage : U12	V / 100	U32
18454	0x4816	2	Ph-Ph Voltage : U23	V / 100	U32
18456	0x4818	2	Ph-Ph Voltage : U31	V / 100	U32
18458	0x481A	2	Current : I1	A / 1000	U32
18460	0x481C	2	Current : I2	A / 1000	U32
18462	0x481E	2	Current : I3	A / 1000	U32
18464	0x4820	2	Current : In	A / 1000	U32

Fig. 10 - 1 Modbus request to read currents/voltages/power on a 3-phase configured module

If we take 2 examples of all-in-one systems:

We can see that if we have:
- 5x I30 modules
- Setup as 1 x three-phase
- You want to read all their main measurements 1x per second: 2 requests required
  - › 1 request to read currents/power (contiguous Modbus registers in the same Modbus table)
  - › 1 request to read the energy readings (independent Modbus table)

→ Sum of 5 x 1 x 2 requests = 10 requests every second
→ OK on a single DIRIS Digiware system with a single screen because it is less than 33 requests per second, which is the limit at the default speed of 38.4 kbds

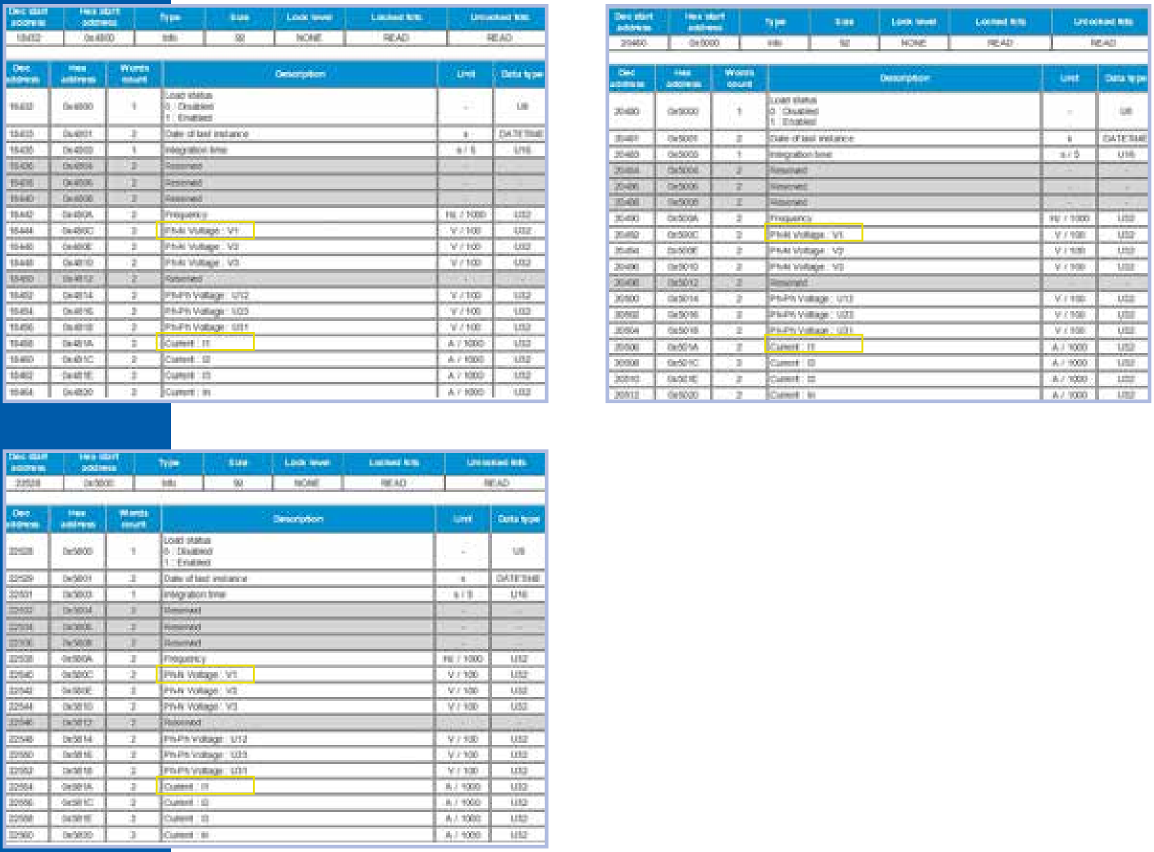
Another example – let's say:

- 10x I30 modules
- configured as 3x single-phase
- you want to read all their main measurements 1x per second: 2 requests required
  - › 1 request to read currents/power (contiguous Modbus registers in the same Modbus table)
  - › 1 request to read the energy readings (independent Modbus table)

→ This gives us a sum of 10 x 3 x 2 requests = 60 requests every second => exceeds the limit of a single screen because it is greater than 33 requests per second, which is the limit at the default speed of 38.4 kbds

→ 3 Solutions:

- read the values once every 2s instead of 1x per second.
- increase the default communication speed to 500 kbds instead of 38 kbds
- always read 1x per second but distribute the system over 2 remote screens/gateways to "double" the communication capacity



Dec start address	Hex start address	Type	Size	Lock level	Locked lots	Unlocked lots
18432	0x4800	Info	32	NONE	READ	READ

Dec address	Hex address	Words count	Description	Unit	Data type
18432	0x4800	1	Load status 0 : Disabled 1 : Enabled	-	U8
18433	0x4801	2	Date of last instance	s	DATETIME
18435	0x4803	1	Integration time	s / 5	U16
18436	0x4804	2	Reserved	-	-
18438	0x4806	2	Reserved	-	-
18440	0x4808	2	Reserved	-	-
18442	0x480A	2	Frequency	Hz / 1000	U32
18444	0x480C	2	Ph-N Voltage : V1	V / 100	U32
18446	0x480E	2	Ph-N Voltage : V2	V / 100	U32
18448	0x4810	2	Ph-N Voltage : V3	V / 100	U32
18450	0x4812	2	Reserved	-	-
18452	0x4814	2	Ph-Ph Voltage : U12	V / 100	U32
18454	0x4816	2	Ph-Ph Voltage : U23	V / 100	U32
18456	0x4818	2	Ph-Ph Voltage : U31	V / 100	U32
18458	0x481A	2	Current : I1	A / 1000	U32
18460	0x481C	2	Current : I2	A / 1000	U32
18462	0x481E	2	Current : I3	A / 1000	U32
18464	0x4820	2	Current : In	A / 1000	U32

Dec start address	Hex start address	Type	Size	Lock level	Locked lots	Unlocked lots
20480	0x5000	Info	32	NONE	READ	READ

Dec address	Hex address	Words count	Description	Unit	Data type
20480	0x5000	1	Load status 0 : Disabled 1 : Enabled	-	U8
20481	0x5001	2	Date of last instance	s	DATETIME
20483	0x5003	1	Integration time	s / 5	U16
20484	0x5004	2	Reserved	-	-
20486	0x5006	2	Reserved	-	-
20488	0x5008	2	Reserved	-	-
20490	0x500A	2	Frequency	Hz / 1000	U32
20492	0x500C	2	Ph-N Voltage : V1	V / 100	U32
20494	0x500E	2	Ph-N Voltage : V2	V / 100	U32
20496	0x5010	2	Ph-N Voltage : V3	V / 100	U32
20498	0x5012	2	Reserved	-	-
20500	0x5014	2	Ph-Ph Voltage : U12	V / 100	U32
20502	0x5016	2	Ph-Ph Voltage : U23	V / 100	U32
20504	0x5018	2	Ph-Ph Voltage : U31	V / 100	U32
20506	0x501A	2	Current : I1	A / 1000	U32
20508	0x501C	2	Current : I2	A / 1000	U32
20510	0x501E	2	Current : I3	A / 1000	U32
20512	0x5020	2	Current : In	A / 1000	U32

Dec start address	Hex start address	Type	Size	Lock level	Locked lots	Unlocked lots
22528	0x5800	Info	32	NONE	READ	READ

Dec address	Hex address	Words count	Description	Unit	Data type
22528	0x5800	1	Load status 0 : Disabled 1 : Enabled	-	U8
22529	0x5801	2	Date of last instance	s	DATETIME
22531	0x5803	1	Integration time	s / 5	U16
22532	0x5804	2	Reserved	-	-
22534	0x5806	2	Reserved	-	-
22536	0x5808	2	Reserved	-	-
22538	0x580A	2	Frequency	Hz / 1000	U32
22540	0x580C	2	Ph-N Voltage : V1	V / 100	U32
22542	0x580E	2	Ph-N Voltage : V2	V / 100	U32
22544	0x5810	2	Ph-N Voltage : V3	V / 100	U32
22546	0x5812	2	Reserved	-	-
22548	0x5814	2	Ph-Ph Voltage : U12	V / 100	U32
22550	0x5816	2	Ph-Ph Voltage : U23	V / 100	U32
22552	0x5818	2	Ph-Ph Voltage : U31	V / 100	U32
22554	0x581A	2	Current : I1	A / 1000	U32
22556	0x581C	2	Current : I2	A / 1000	U32
22558	0x581E	2	Current : I3	A / 1000	U32
22560	0x5820	2	Current : In	A / 1000	U32

Fig. 11 – Modbus requests to read current / voltage / power on a module configured as 3x single-phase.

Summary:

	Example #1	Example #2
System size	5x I30 modules	10x I30 modules
Number of loads/modules	Configured as 1x three-phase	Configured as 3x single-phase
Expected readings and reading frequency	Measurement + energy readings 1x per second: 2 requests required	Measurement + energy readings 1x per second: 2 requests required
Bandwidth required	5 x 1 x 2 requests = 10 requests every second	10 x 3 x 2 requests = 60 requests every second
Comparison to theoretical max. bandwidth:	< at 33.5 requests per second, which is the default bandwidth of a DIRIS Digiware gateway/screen	> at 33.5 requests per second, which is the default bandwidth of a DIRIS Digiware gateway/screen
Conclusion	OK	NOK
Actions:	1 screen will suffice	Reduce the reading frequency, or increase the communication speed, or use 2 gateways/screens to "distribute" the communication load

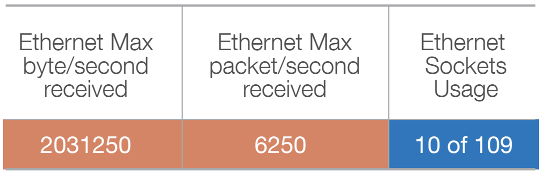
Note:

The larger the amount of loads on each system, the more messages there will be and the bandwidth reduces as a result: access to 10x DIRIS Digiware I-30 devices with three-phase loads will be three times faster than the same 10x I30 devices with 3 single-phase loads (3x as many messages to read)

Socomec teams are here to help you make a detailed simulation / case study specific to your project.

Impact of network traffic (other than Modbus TCP IP of Socomec devices, but also email, web surfing, YouTube browsing, etc.):

Example of an overloaded network exceeding 6000 packets per second seen by the DIRIS Digiware D-50 screen:



Ethernet Max byte/second received	Ethernet Max packet/second received	Ethernet Sockets Usage
2031250	6250	10 of 109

This traffic is seen by the screen or gateway, even if this traffic is not intended for it.

It is recommended not to use a DIRIS Digiware system on a TCP network exceeding 2000 TCP/IP packets of existing network traffic (this would be like trying to add traffic on an already congested motorway).

Impact of configuring "retries/timeout" settings:

When a question arrives via Ethernet on a D50 display for example, it will transfer it to the Digiware bus. If there is no error of communication the data flow will work like this:

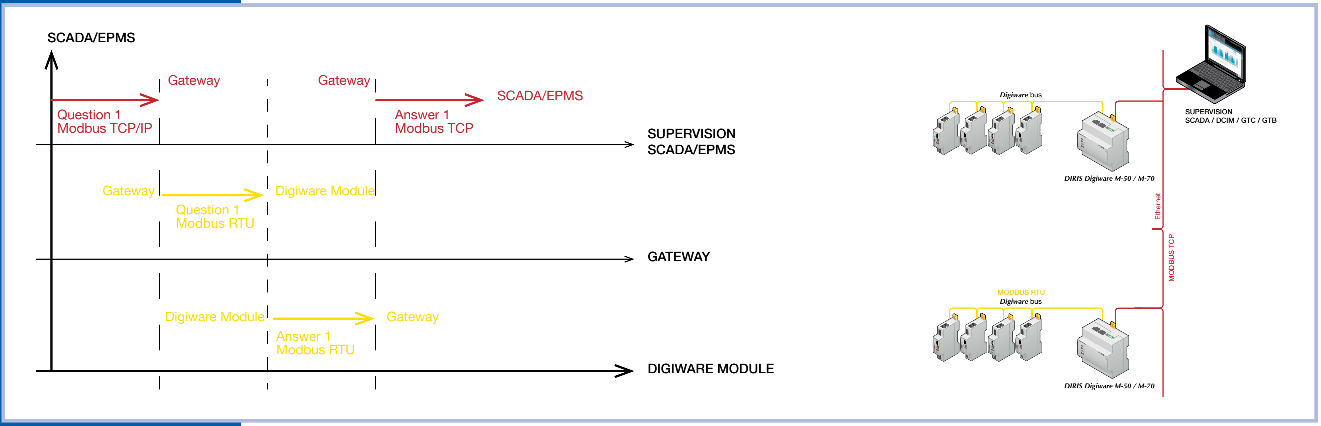


Fig. 12 – Impact of configuring "retries/timeout" settings

If the slave has not replied after XX seconds, the gateway will try NN times to resend the question. XX is the timeout and NN the number of retries.

Example - gateway is set up as follows:

TIMEOUT: 500 ms
RETRY: 3

The gateway will retry after 500 ms 3x in a row to ask the question again. If after 1.5 seconds (3 retry questions waiting 500 ms in between each time) there is still no reply, the D50 will respond to the monitoring unit that it failed to communicate with the slave and will return a Modbus 0B error code.

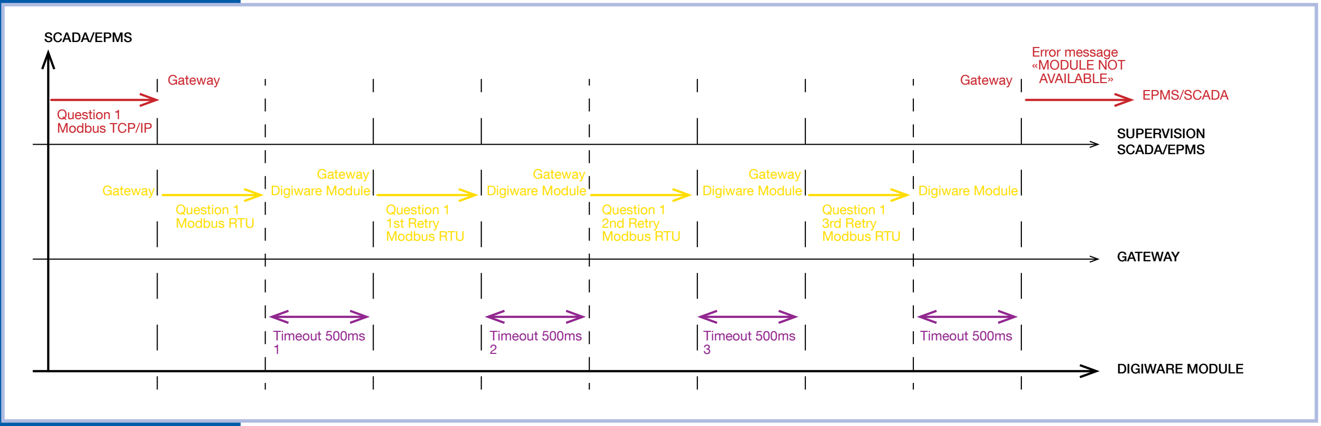


Fig. 12 – Impact of configuring "retries/timeout" settings:

This can be complicated if the monitoring system also handles retries/timeouts at its level.

Suppose the EPMS/SCADA is configured as follows:

TIMEOUT: 1 second
RETRY: 2

This means that the EPMS/SCADA will send a question to the display; if there is no reply it will send the question again after 1s.

This will generate a new question even if the “internal gateway retry cycle” is not yet finished and will generate a risk of communication collision :

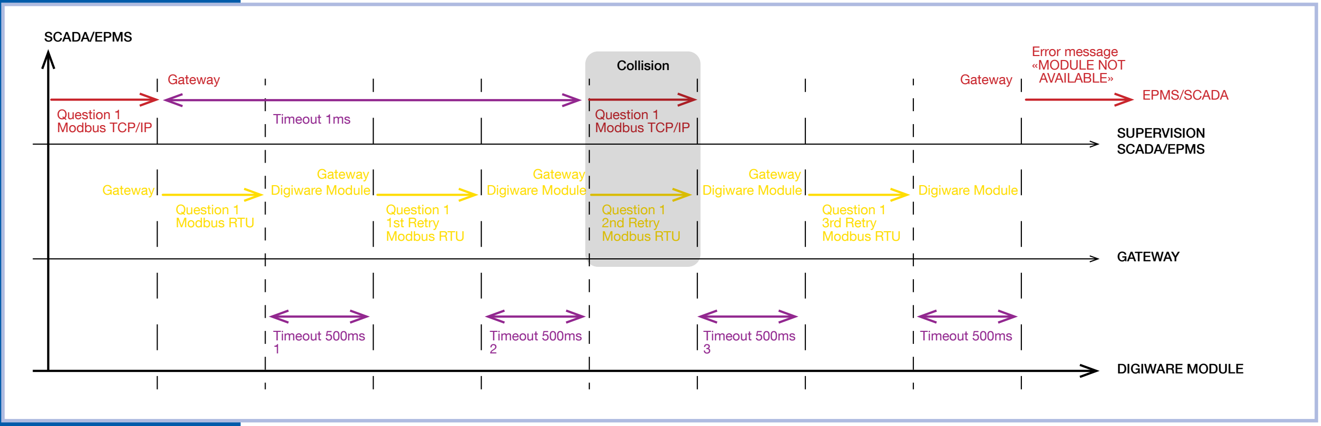


Fig. 13 –

The rule is as follows:

(Number of D50 retries+1) x D50 timeout < 150% monitoring unit timeout

In our case, we would have to switch the timeout of the monitoring unit to 3s, or reduce the display timeout to 200 ms.

Other communication and cybersecurity protocols

The DIRIS Digiware system, downstream of the gateway / DIRIS Digiware M/D screen, operates in Modbus RTU.
The gateway/screen makes the data available upstream in Modbus TCP.

At Ethernet level it is also possible to recover the measurements and consumptions of the DIRIS Digiware modules under other protocols:

SNMP v1/v2/v3:

Protocol suited to monitoring IT components.

It is possible to retrieve the measurements in OID format (SNMP equivalent of a Modbus register).
It is possible to retrieve alarms in TRAP format (push alarms, unlike the permanent and periodic system of requests in Modbus); this optimises the traffic / your use of the network bandwidth.
A MIB file is available on the Socomec website to use this protocol (MIB file is the equivalent of a driver).

Bacnet IP

Protocol more suited to buildings / BMS. It is an object-oriented protocol, meaning that it is plug & play and that the monitoring system will automatically recognise which reported data are voltage readings and which reported data are current readings, and so on.
A document called "PICS" is available on the Socomec website, which details how you can use the Bacnet IP with the Socomec system.

FTP(s)

CSV file transfer protocol, more suitable for sending monitoring archives/logs in a database, rather than for real-time monitoring.
This protocol is particularly suitable for systems specialising in energy optimisation.

HTTP(s)

A web server embedded in the gateway/screen lets you view real-time and historical monitoring/consumption data in a web browser.

At Ethernet level, the system gateways/screens comply with Cybersecurity Standard IEC62443.
There is a separate technical note available on this subject, for more information.

Diagnostic

A dedicated technical guide on the diagnostics of these communication systems is available from Socomec. Below is a summary.

A communication problem in a Modbus TCP monitoring architecture (GTC/GTB/SCADA/DCIM/BMS) reporting real-time electrical monitoring data may seem difficult to diagnose, but the use of a structured approach ensures a smooth operation:

1. Identifying the nature of the communication failure:

A communication fault can have several characteristics. Identifying it accurately is the first step to solving the problem.

If data is received but is not consistent with the value read on the device locally, we can suppose that there was a problem in interpreting the data: if the data was formatted on 1 or 2 Modbus registers (16/32 bits), a signed/unsigned value, an inversion of heavyweight/lightweight registers, offsetting of the communication table (bad Modbus register interrogated) are the general paths to be explored.

Or are we talking about a loss of communication? If so, is it temporary or permanent? If temporary, is there a cyclical event (every day between 12:00 pm and 12:10 pm, for example)? When we talk about communication loss, did the monitoring unit not receive any reply at all, or a reply giving a current/voltage/kWh measurement value that seems inconsistent, or a Modbus error code emitted either by the gateway or by the PMD/meter itself?
It is also important at this stage to identify what the expectations of the monitoring unit are in terms of the number of values to be reported, data reporting frequency and the typology of the Modbus TCP interrogations (number of sockets used in series and/or parallel).
Does the device start again when you restart the screen and/or the gateway, or just when you restart the Digiware bus?

2. Fault diagnostics:

There are a number of LAN/MODBUS TCP network diagnostics tools (e.g. Wireshark).
They are a great help in finding the source component of the communication limitation on the network:

- The monitoring system itself (GTB/BMS/SCADA/BMS/DCIM)
- Elements of the network infrastructure (firewall, hub, switch, router, VPN)
- The electrical monitoring and TCP/IP communication system of the meters/PMD

The IT maintenance teams at the customer site can perform a more detailed diagnosis when a malfunction is detected on the monitoring unit and/or elements of the network infrastructure.
The detection of a malfunction on the electrical monitoring system (gateway and/or meters/PMD) can be diagnosed with specially designed tools such as DigiwareDiag and/or ModbusVision.

There is a complete technical guide dedicated to the use of these diagnostic tools.

Summary

This technical note clarifies and demystifies the advanced Modbus TCP/IP communication functions and limitations of a DIRIS Digiware system accompanied by a gateway Mxx or an Dxx display screen.
Following the recommendations and rules for implementing communication set out in this note will help you optimise the performance of your monitoring system.
Of course, this is in addition to complying with the applicable rules in terms of configuring and wiring the entire monitoring and communication chain.

As experts happy to support our partners, Socomec is on hand for any case studies and custom-sizing of communication architecture.

Glossary

Socket:

A TCP/IP connection. It is possible to open several connections in parallel, similar to when you open several web pages on your computer at the same time. This is not possible at the DIRIS Digiware / Modbus RTU downstream bus, or only one connection is possible so the connections are sequenced one after the other.

Request/Question:

This is a Modbus message sent by the Modbus RTU master to the DIRIS Digiware monitoring modules.

Modbus register:

This is the address at which a value is stored. The register will be different if we want to read the voltage, current or power, etc.
You can liken the address of the DIRIS Digiware monitoring module to a street name, and the specific measurement that we want to read with its register to the house or apartment number.

Some documentation will use the term "word" instead of "register".

TCP packet:

Under the IP protocol, all data sent over the Internet is broken down into smaller pieces called packets.
For example, when a Web page is sent from a Web server to a user's laptop, the data within the Web page travels online in the form of a series of packets.
The packets are then reassembled by the laptop in order to then build the web page on the screen.
For example, a TCP/IP packet can contain 1 Modbus TCP request.

Master/Slave:

At Modbus RTU level, the gateway/screen is the master; the one that asks the questions.
Slaves are the monitoring modules and speak only to answer a question put to them. A question must be followed by a reply before you can send a new question.

Client/Server:

This equates to the notion of master/slave. The client requests the data from the server that provides it. So the gateway/screen is the Modbus RTU client and the DIRIS Digiware monitoring modules are the Modbus RTU servers.
It is important to note that a gateway/screen has a dual role:

- Client/master at Modbus RTU level
- But server/slave at Modbus TCP level

Turnabout delay:

This is the pause time expected by a gateway/screen when it has received a reply from a DIRIS Digiware module, before sending a question to the next DIRIS Digiware module.

Timeout:

This is a countdown timer that determines the amount of time during which the Modbus RTU master is likely to receive the reply from a Modbus RTU slave. If the slave replies within this allotted time, then the reply is taken into account. If the slave does not reply in this period of time, then it is assumed that the slave is no longer responding and at that time the master can repeat the same question to the same slave (see number of retries).

Retries:

When the timeout is reached, the retries are the number of times a gateway/screen will try to ask the same question to the DIRIS Digiware module that is not currently responding.

Digibus:

This is the yellow RJ45 serial bus that connects all DIRIS Digiware modules.
It is composed of a Modbus RTU bus, as well as a patented proprietary digital transmission protocol of the single voltage measurement for the entire bus.
The power supply of the modules also travels on this Digibus.

Conclusion

This technical note clarifies and demystifies the advanced features and limits of the **Modbus TCP** communication with a **DIRIS Digiware** system headed by a display or gateway **Dxx/Mxx**.
By following the guidelines expressed in this technical note you will be able to optimise the performance of a supervision monitoring system.
Of course, the electrical standards and safety rules of a specific country must be taken into account.

As an expert and with the will to support our partners, Socomec remains available for any questions or specific case study analysis about communication architecture.

Socomec: our innovations supporting your energy performance

1 independent manufacturer

3,600 employees worldwide

10 % of sales revenue dedicated to R&D

400 experts dedicated to service provision

Your power management expert

POWER SWITCHING

POWER MONITORING

POWER CONVERSION

ENERGY STORAGE

EXPERT SERVICES

The specialist for critical applications

- Control, command of LV facilities
- Safety of persons and assets

- Measurement of electrical parameters
- Energy management

- Energy quality
- Energy availability
- Energy storage

- Prevention and repairs
- Measurement and analysis
- Optimisation
- Consultancy, commissioning and training

A worldwide presence

12 production sites

- France (x3)
- Italy (x2)
- Tunisia
- India
- China (x2)
- USA (x3)

28 subsidiaries and commercial locations

• Algeria • Australia • Belgium • China • Canada
• Dubai (United Arab Emirates) • France • Germany
• India • Indonesia • Italy • Ivory Coast • Netherlands
• Poland • Portugal • Romania • Serbia • Singapore
• Slovenia • South Africa • Spain • Switzerland
• Thailand • Tunisia • Turkey • UK • USA

80 countries
where our brand is distributed

HEAD OFFICE

SOCOMEC GROUP
SAS SOCOMEC capital 10 646 360 €
R.C.S. Strasbourg B 548 500 149
B.P. 60010 - 1, rue de Westhouse
F-67235 Benfeld Cedex
Tel. +33 3 88 57 41 41 - Fax +33 3 88 57 78 78
info.scp.isd@socomec.com

www.socomec.com

YOUR DISTRIBUTOR / PARTNER

100 years OF SHARED ENERGY 1922 — 2022

socomec
Innovative Power Solutions

DOT 00373 02 en - 09/21 - Photo: Martin Bernhart - Réalisation: Socomec - Gyss Imprimeur Obernai