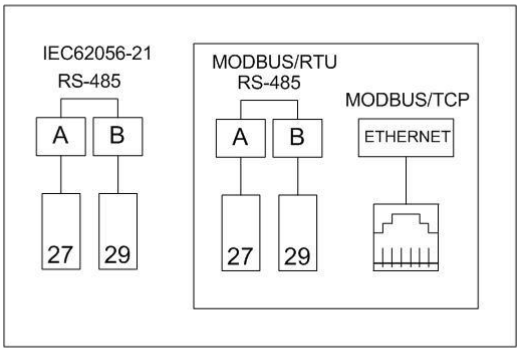
Plug&play module MKMB-3-e-3

Modbus/RTU – Modbus/TCP interface for Iskraemeco MT831 / MT860 meters



Fw rev. Feb. 23, 2013 – Manual rev. 1.4

NEW FEATURES:

- ***totalizers available as 32 bit registers***
- ***updated registers set for MT860 meters***

Telematica Sistemi s.r.l.
Via Vigentina, 71 - 27010 Zeccone (PV)
Tel. 0382.955051 Fax. 0382.957591
info@telematicasistemi.it   www.telematicasistemi.it

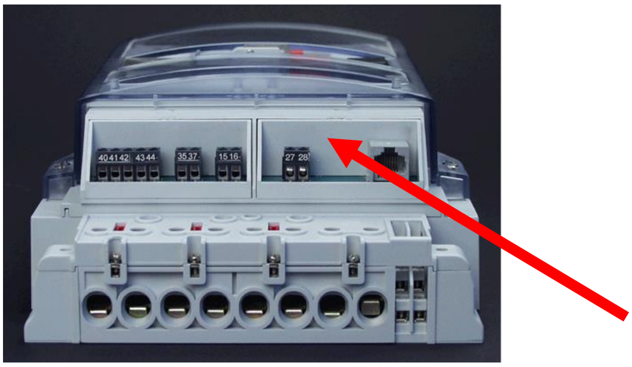
General informations

The MKMB-3-e-3 module provides the following communication interfaces:

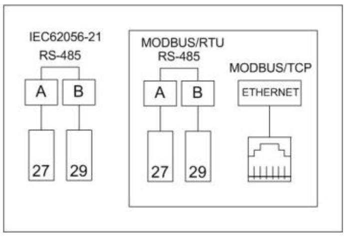
- RS485 serial interface with the IEC 62056-21 communication protocol
- RS485 serial interface with the Modbus/RTU communication protocol
- RJ45 Ethernet interface with the Modbus/TCP communication protocol

A Web Server with the standard HTTP protocol is also available on the RJ45 Ethernet interface.
The MKMB modules are delivered from the factory with the default IP address: 10.3.11.119

Module installation



The MKMB-3-e-3 module fits into the right slot of the meter, as for all the other communication modules



Communication parameters

Modbus server address: 1 ÷ 247
<u>Default</u>: 100 + last two digits of the meter factory number

Modbus/RTU serial communication:

Speed:	2400, 4800, 9600, 19200
Data bits:	8
Parity:	None, Even, Odd
Stop bits:	1, 2
<u>Default</u>:	9600 8 N 1

All registers are available through the Modbus function "*03: Read Holding Registers*"

Data types are compliant with standard data types of the C programming language:

- Short 2 bytes 16 bit two's complement
- integer 4 bytes 32 bit two's complement
- float 4 bytes 32 bit floating point IEEE 754 format
- double 8 bytes 64 bit floating point IEEE 754 format

All values are expressed according with the engineering units programmed into the meter.
We suggest you to check on the meter display the actual settings, as they could differ from the following default settings:

Active energy	kWh	Reactive energy	kvarh		
Active power	kW	Reactive power	kvar	Apparent power	VA
Frequency	Hz	Current	A	Voltage	V

Modbus/RTU registers mapping

Instantaneous values

OBIS code	Modbus register	Type	Size in bytes	Description
0.0.1	0	long	4	Meter factory number
0.9.1	2	integer	6	Actual time: Bytes 0-1 hours (0-24) Bytes 2-3 minutes (0-59) Bytes 4-5 seconds (0-59)
0.9.2	5	integer	6	Actual date Bytes 0-1 year (AA) Bytes 2-3 month Bytes 4-5 day
1.8.0	8	double	8	Active energy A+ totalizer register
2.8.0	12	double	8	Active energy A- totalizer register
3.8.0	16	double	8	Reactive Q+ = Q1 + Q2 totalizer register
4.8.0	16	double	8	Reactive Q- = Q3 + Q4 totalizer registers
5.8.0	24	double	8	Reactive Q1 totalizer register
6.8.0	28	double	8	Reactive Q2 totalizer register
7.8.0	32	double	8	Reactive Q3 totalizer register
8.8.0	36	double	8	Reactive Q4 totalizer register
9.8.0	40	double	8	Apparent energy S+ totalizer register
10.8.0	44	double	8	Apparent energy S- totalizer register
13.7.0	48	float	4	Average power factor
14.7.0	50	float	4	Average power factor
15.7.0	52	float	4	Σ*Li* active power (abs(QI+QIV)+(abs(QII+QIII))
130.7.0	54	float	4	Instantaneous reactive power (Q1+Q2+Q3+Q4)
131.7.0	56	float	4	Instantaneous apparent power (Q1+Q2+Q3+Q4)

In order to have access to all the instantaneous values, phase values and load profiles, the MT831 meters must be ordered with the special "Modbus enabled firmware", otherwise only the data with the green background on the above table will be available.
Data available on MT860 meters are limited to the green background items.

On meters with the special "Modbus enabled firmware", the register values are updated every 10 seconds.

Phase values

OBIS code	Modbus register	Type	Size in bytes	Description
81.7.40	58	float	4	Phase angle in phase R
81.7.51	60	float	4	Phase angle in phase S
81.7.62	62	float	4	Phase angle in phase T
81.7.1	64	float	4	Phase angle V2 -> V1
81.7.2	66	float	4	Phase angle V3 -> V1
31.7.0	68	float	4	Average current RMS in phase R
31.7.3	70	float	4	Average 3rd current harmonics, phase R
31.7.5	72	float	4	Average 5th current harmonics, phase R
31.7.7	74	float	4	Average 7th current harmonics, phase R
51.7.0	76	float	4	Average current RMS in phase S
51.7.3	78	float	4	Average 3rd current harmonics, phase S
51.7.5	80	float	4	Average 5th current harmonics, phase S
51.7.7	82	float	4	Average 7th current harmonics, phase S
71.7.0	84	float	4	Average current RMS in phase T
71.7.3	86	float	4	Average 3rd current harmonics, phase T
71.7.5	88	float	4	Average 5th current harmonics, phase T
71.7.7	90	float	4	Average 7th current harmonics, phase T
32.7.0	92	float	4	Average voltage RMS in phase R
32.7.3	94	float	4	Average 3rd voltage harmonics, phase R
32.7.5	96	float	4	Average 5th voltage harmonics, phase R
32.7.7	98	float	4	Average 7th voltage harmonics, phase R
52.7.0	100	float	4	Average voltage RMS in phase S
52.7.3	102	float	4	Average 3rd voltage harmonics, phase S
52.7.5	104	float	4	Average 5th voltage harmonics, phase S
52.7.7	106	float	4	Average 7th voltage harmonics, phase S
72.7.0	108	float	4	Average voltage RMS in phase T
72.7.3	110	float	4	Average 3rd voltage harmonics, phase T
72.7.5	112	float	4	Average 5th voltage harmonics, phase T
72.7.7	114	float	4	Average 7th voltage harmonics, phase T

In order to have access to all the instantaneous values, phase values and load profiles, the MT831 meters must be ordered with the special "Modbus enabled firmware", otherwise only the data with the green background on the above table will be available.
Data available on MT860 meters are limited to the green background items.

Load profiles, last recorded values

	Modbus register	Type	Size in bytes	Description
P.01 (TST)	116	integer	12	Last recorded profile P.01 value timestamp Bytes 0-1 hour (0-24) Bytes 2-3 minute (0-59) Bytes 4-5 second (0-59) Bytes 6-7 year (YYYY) Bytes 8-9 month Bytes 10-11 day
P.01 (C1)	122	double	8	Profile P.01 - Last recorded value channel 1
P.01 (C2)	126	double	8	Profile P.01 - Last recorded value channel 2
P.01 (C3)	130	double	8	Profile P.01 - Last recorded value channel 3
P.01 (C4)	134	double	8	Profile P.01 - Last recorded value channel 4
P.01 (C5)	138	double	8	Profile P.01 - Last recorded value channel 5
P.01 (C6)	142	double	8	Profile P.01 - Last recorded value channel 6
P.02 (TST)	146	integer	12	Last recorded profile P.02 value timestamp Bytes 0-1 hour (0-24) Bytes 2-3 minute (0-59) Bytes 4-5 second (0-59) Bytes 6-7 year (YYYY) Bytes 8-9 month Bytes 10-11 day
P.02 (C1)	152	double	8	Profile P.02 - Last recorded value channel 1
P.02 (C2)	156	double	8	Profile P.02 - Last recorded value channel 2
P.02 (C3)	160	double	8	Profile P.02 - Last recorded value channel 3
P.02 (C4)	164	double	8	Profile P.02 - Last recorded value channel 4
P.02 (C5)	168	double	8	Profile P.02 - Last recorded value channel 5
P.02 (C6)	172	double	8	Profile P.02 - Last recorded value channel 6

P.01 and P.02 are the first and the second load profiles.
Data stored in each load profile channel depend on the meter configuration.
On default factory settings, load profile P.01 holds the average demand (both positive and negative directions) and reactive demand (divided by quadrant) at 15 minutes integration period, with the following channels sequence: P+, Q1, Q4, P-, Q2, Q3.

Replication of double 64 bit registers into couples of 16/32 bit registers

Starting from firmware release of November 2011, available for sale since January 2012, floating point values of 64 bit double type are available also in 32bit format, encoded into long format.

Firmware version of the MKMB-3-e-3 module is visible at the HTML starting page of the module's web interface.

Double values have been splitted into two components:

- data expressed in integer format
- number of decimal digits of the original data

Original value can be obtained by applying the formula: (long value) / ($10^{\text{number of associated decimal digits}}$)

Example: 1.8.0 totalizer with value 01485,652

Register content 176:	***3***
Content of registers 177 ÷ 178 (long format):	***1485652***
Original value of the 1.8.0 totalizer:	***1485652 / 10 ^{3} = 1485652 / 1000 = 1485,652***

OBIS Code	Modbus register	Type	Size in bytes	Description
	176	short	2	Number of decimal digits of the totalizer registers 1.8.0 ÷ 10.8.0
1.8.0	177	long	4	32 bit encoding of the A+ active energy totalizer
2.8.0	179	long	4	32 bit encoding of the A- active energy totalizer
3.8.0	181	long	4	32 bit encoding of the Q+ = Q1+ Q2 reactive energy totalizer
4.8.0	183	long	4	32 bit encoding of the Q- = Q3+ Q4 reactive energy totalizer
5.8.0	185	long	4	32 bit encoding of the Q1 reactive energy totalizer
6.8.0	187	long	4	32 bit encoding of the Q2 reactive energy totalizer
7.8.0	189	long	4	32 bit encoding of the Q3 reactive energy totalizer
8.8.0	191	long	4	32 bit encoding of the Q4 reactive energy totalizer
9.8.0	193	long	4	32 bit encoding of the S+ apparent energy totalizer
10.8.0	195	long	4	32 bit encoding of the S- apparent energy totalizer

OBIS Code	**Modbus register**	**Type**	**Size in bytes**	**Description**
	197	short	2	Number of decimal digits of P.01 (C1)
	198	short	2	Number of decimal digits of P.01 (C2)
	199	short	2	Number of decimal digits of P.01 (C3)
	200	short	2	Number of decimal digits of P.01 (C4)
	201	short	2	Number of decimal digits of P.01 (C5)
	202	short	2	Number of decimal digits of P.01 (C6)
P.01 (TST)				Date/time last period of P.01 profile
	203	short	2	Byte 0-1 hour (0-24)
	204	short	2	Byte 2-3 minutes (0-59)
	205	short	2	Byte 4-5 seconds (0-59)
	206	short	2	Byte 6-7 year (XXXX)
	207	short	2	Byte 8-9 month
	208	short	2	Byte 10-11 day
P.01 (C1)	209	long	4	1st channel of the P.01 profile's last period 32 bit encoding
P.01 (C2)	211	long	4	2nd channel of the P.01 profile's last period 32 bit encoding
P.01 (C3)	213	long	4	3rd channel of the P.01 profile's last period 32 bit encoding
P.01 (C4)	215	long	4	4th channel of the P.01 profile's last period 32 bit encoding
P.01 (C5)	217	long	4	5th channel of the P.01 profile's last period 32 bit encoding
P.01 (C6)	219	long	4	6th channel of the P.01 profile's last period 32 bit encoding

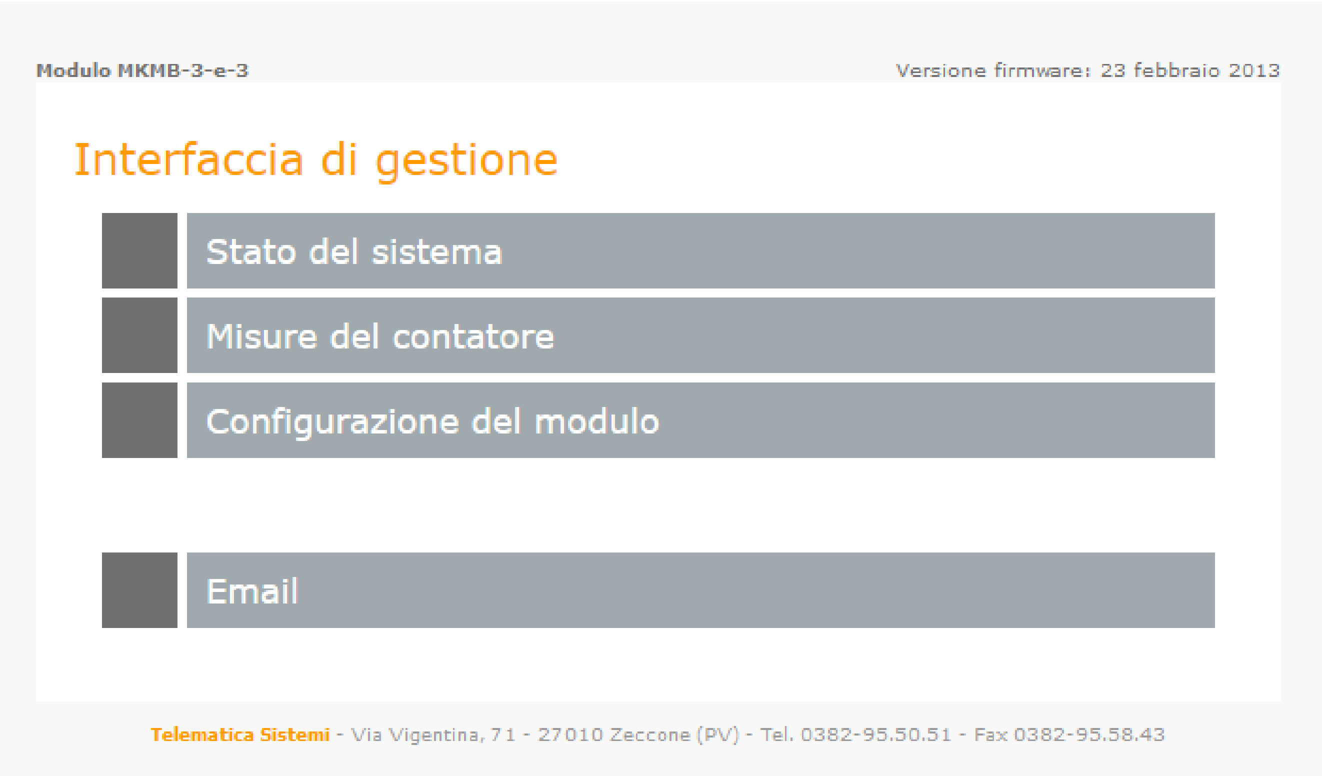
OBIS code	Modbus register	Type	Size in bytes	Description
	221	short	2	Number of decimal digits of P.02 (C1)
	222	short	2	Number of decimal digits of P.02 (C2)
	223	short	2	Number of decimal digits of P.02 (C3)
	224	short	2	Number of decimal digits of P.02 (C4)
	225	short	2	Number of decimal digits of P.02 (C5)
	226	short	2	Number of decimal digits of P.02 (C6)
P.02 (TST)				Date/time last period of P.01 profile
	227	short	2	Byte 0-1 hour (0-24)
	228	short	2	Byte 2-3 minutes (0-59)
	229	short	2	Byte 4-5 seconds (0-59)
	230	short	2	Byte 6-7 year (XXXX)
	231	short	2	Byte 8-9 month
	232	short	2	Byte 10-11 day
P.02 (C1)	233	long	4	1st channel of the P.02 profile's last period 32 bit encoding
P.02 (C2)	235	long	4	2nd channel of the P.02 profile's last period 32 bit encoding
P.02 (C3)	237	long	4	3rd channel of the P.02 profile's last period 32 bit encoding
P.02 (C4)	239	long	4	4th channel of the P.02 profile's last period 32 bit encoding
P.02 (C5)	241	long	4	5th channel of the P.02 profile's last period 32 bit encoding
P.02 (C6)	243	long	4	6th channel of the P.02 profile's last period 32 bit encoding

Setting the Modbus server address and the communication parameters

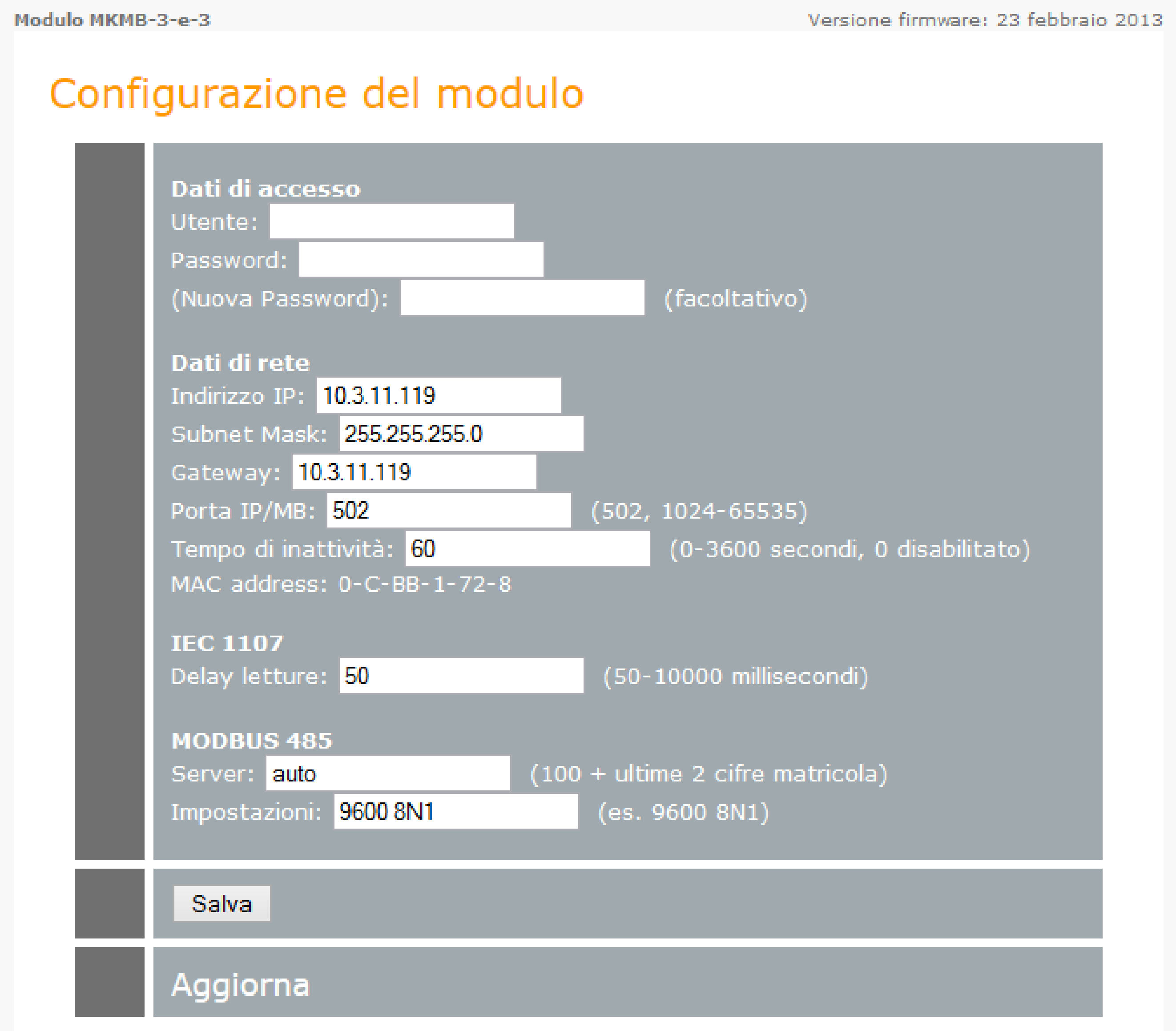
MKMB modules are factory programmed with default settings for RS485 and TCP/IP communication.

The module settings can be changed using an Internet browser.
The default address of the embedded Web server is http://10.3.11.119.

In order to access the module the PC must be on the same network subnet of the module.
Instructions on how to change the IP address of the PC can be found in the Windows help.



Home page of the embedded Web server



To change the parameters you must first enter the Username ("Utente") and the password. To save the changes press the "Salva" button.

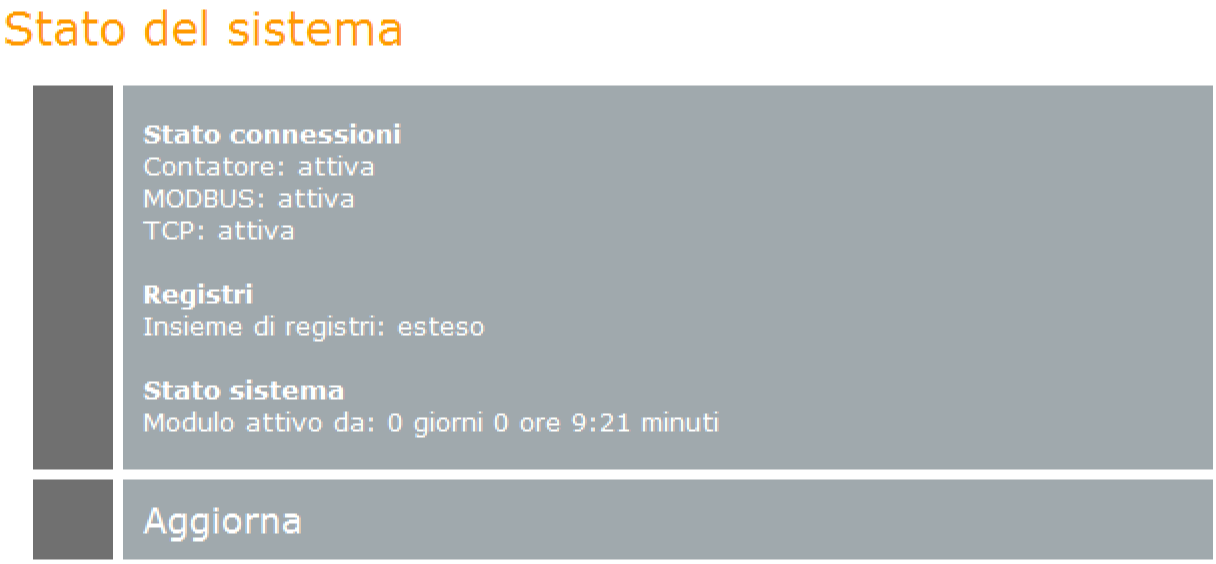
Pay attention when changing the network parameters and/or the password: in case you forget your settings it could be necessary to send the module back to the factory to restore the default settings.

The Modbus/RTU address is automatically set during the startup of the module adding 100 to the last two digits of the factory number of the meter hosting the module.

E.g.:	meter factory number:	35 705 236
	Last two digits:	36
	Default Modbus/RTU address =	136

Access	**Default value**	**Description**
Utente (User)	***user***	Requested to save the changes
Password	***pwd***	Requested to save the changes
Nuova (new) Password		**Password length is limited to three characters**
Network	**Default value**	**Description**
IP address	10.3.11.119	Module TCP/IP address
Subnet Mask	255.255.255.0	
Gateway	10.3.11.119	
IP/MB port	502	Modbus/TCP port
Tempo di inattività	60	Modbus/TCP inactivity timeout in seconds. If no requests are sent to the module within the specified timeout the open TCP/IP connection will be closed. Useful to prevent orphan TCP connections when the client PC hangs without closing the TCP socket.
MAC address	Factory defined	
IEC 1107	**Default value**	**Description**
Poll delay	500	Meaningful only for meters not running the “Modbus enabled firmware”. It holds the communication delay between the MKMB and the meter.
MODBUS/RTU	**Default value**	**Description**
Server	auto	When set to “auto” the Modbus/RTU server address is automatically defined during the module’s power on
Impostazioni	9600 8N1	Modbus/RTU RS485 serial communication setting

MKMB module status



The information showed on the status page are summarized on the following table.

Stato connessioni	**Description**
Contatore	Connection status between the meter and the module. attiva = active
MODBUS	Status of the MODBUS server
TCP	Status of the network interface
Registri	**Description**
Depending on the firmware in the meter, two status are possible: • ridotto (reduced) • esteso (extended)	 only a subset of data is available (see page 3) the whole set of data is available
Stato Sistema	**Description**
Time elapsed (approx.) since the startup of the module	

Testing the operation of the Modbus module

The Web interface can be used to read some of the main registers of the meter.

On meters not programmed with the "Modbus-enabled" firmware some registers will not be available.

MKMBConf – Software for testing the Modbus/RTU RS485 interface

<u>Prerequisites</u>

A Personal Computer running WindowsXp / Vista / Windows 7 equipped with:

- an RS485 half duplex (2 wires) serial port

  or

- an RS232 + RS232/RS485 interface converter (eg. Iskra CON1H)

Checking the module operation

The "<u>Controllo</u>" page is intended to check the correctness of the module configuration and its proper operation.
The control is done through a client Modbus/RTU procedure that communicates continuously with the MKMB module.
First, you must connect your PC to the primary RS485 port of the module (the rightmost RS485 labelled "Modbus/RTU").

On the upper left panel of the " <u>Controllo</u>" page, set the communication parameters and the Modbus server address (using the predefined values or the custom values previously set into the meter).
Click on the "<u>Avvia</u>" button to start the communication (polling) with the meter using the Modbus/RTU protocol.
The received data will be shown in tabular and graphical formats.
Although the request is sent every second, the received informations are updated every 10 seconds.
The trace of the communication is shown for diagnostic purpose on the bottom area of the window.
Click on the "<u>Arresta</u>" button to stop the communication.

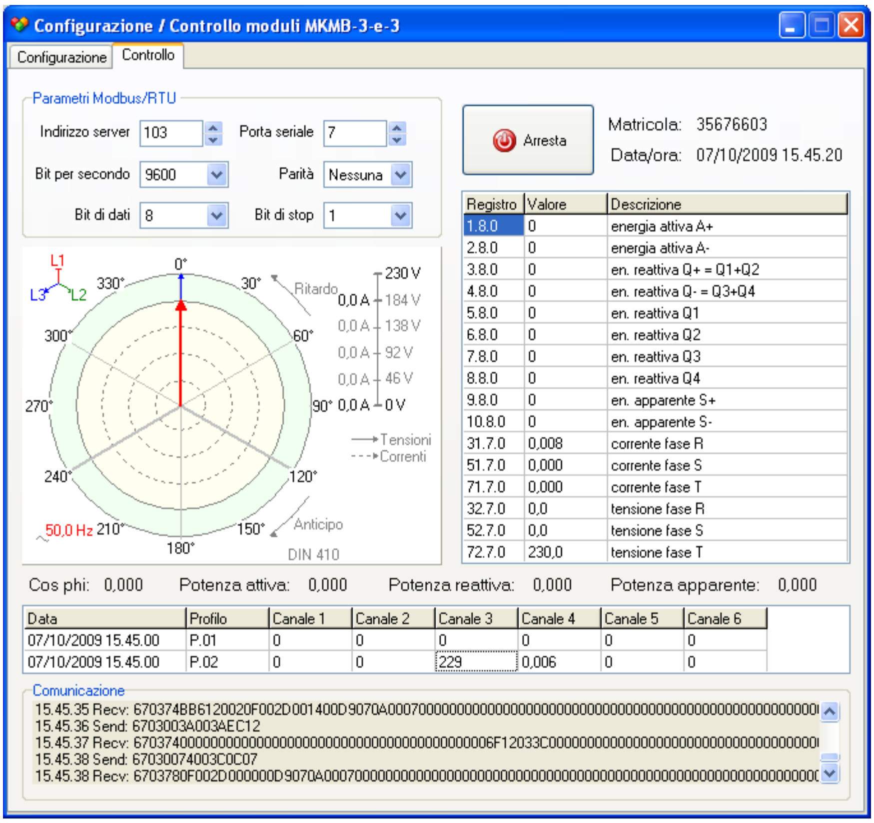


Figure 1

By selecting an incorrect or an already in use serial port, an error message will be shown.
By selecting a valid serial port but not connected to the module, or using incorrect format settings, the message "Nessuna risposta ricevuta!" will be repeatedly shown on the trace area to alert the user that no answer has been received for the sent request. In this case, check the correctness of the communication parameters, of the parameters stored into the meter and the physical connection between the PC and the module.

Parameters changed using the "Configurazione" page will have no effect on modules on which Modbus/TCP protocol is present.

How to decode the MODBUS/RTU data packets

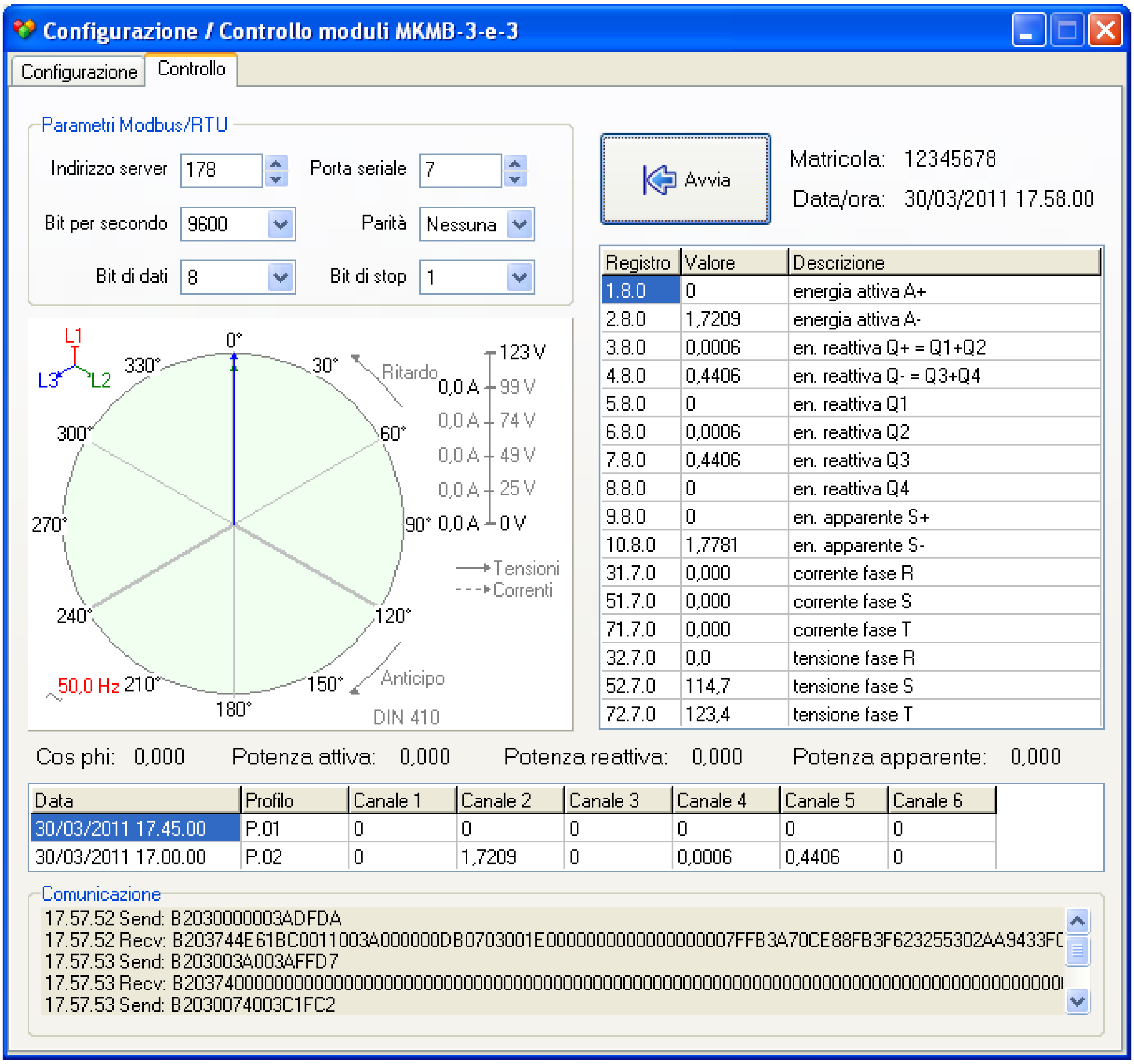


Figure 2

The above figure is showing a real example of multiple registers reading.

On "*Comunicazione*" section in the bottom part of the screen, is shown the trace of the data exchanged between the PC and the module.

To the request B2030000003ADFDA the module answers with B20374 followed by the 16 bit registers:
4E61BC0011003A000000DB0703001E000000000000000007FFB3A70CE88FB3F623255302…
where **4E61** is the register 0, **BC00** is the register 1, **1103** is the register 2 and so on.

***NB**: In order to decode the values you must swap the bytes, the words and the dwords.*

Example 1 (meter factory number –32 bit integer)
The meter factory number is stored in the Modbus registers 0 and 1, on the example: **4E61 BC00**
Swapping the words and the bytes inside the words will result in **00BC614E**
By converting to decimal the hex value “0x00BC614E”, the value 12345678 is obtained, that is the factory number of the meter.

Example 2 (cumulative register – double IEEE 754)
Decode the four consecutive Modbus registers 12 to 15, **7FFB 3A70 CE88 FB3F**
By swapping the dwords, then the words and finally the bytes inside the words, the value **3FFB88CE703AFB7F** is obtained.
Converting the value according to the IEEE 754 double format, the obtained value is 1.720, that is the value of the A- cumulative register 2.8.0 of the example shown in ***Figure 2***.

An interesting calculator can be found at:
http://babbage.cs.qc.edu/IEEE-754/64bit.html

Example 3 (voltage – float IEEE 754)
Decode the two Modbus registers 100 and 101: **6666 E542**
Swapping the words and the bytes inside the words results will result in **42E56666**
Converting the value according to the IEEE 754 format, the obtained value is 114.7, that is the value of the phase-S voltage (OBIS register code 52.7.0).

IEEE 754-1985 *(from Wikipedia)*

IEEE 754–1985 was an industry standard for representing floating-point numbers in computers, officially adopted in 1985 and superseded in 2008 by IEEE 754-2008. During its 23 years, it was the most widely used format for floating-point computation. It was implemented in software, in the form of floating-point libraries, and in hardware, in the instructions of many CPUs and FPUs. The first chip to implement IEEE 754–1985 was the Intel 8087.

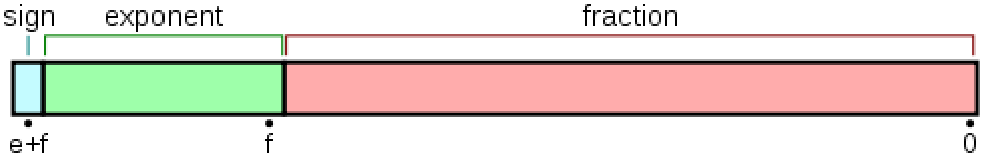
IEEE 754–1985 represents numbers in binary, providing definitions for four levels of precision, of which the two most commonly used are:

level	width	range	Precision
single precision	32 bits	$\pm1.18\times10^{-38}$ to $\pm3.4\times10^{38}$	approx. 7 decimal digits
double precision	64 bits	$\pm2.23\times10^{-308}$ to $\pm1.80\times10^{308}$	approx. 15 decimal digits

The standard also defines representations for positive and negative infinity, a "negative zero", five exceptions to handle irregular results like division by zero, special values called NaNs for representing those exceptions, denormal numbers to represent numbers outside the ranges shown above, and four rounding modes.

Structure of a floating-point number

General layout



The three fields in an IEEE 754 float

Binary floating-point numbers in IEEE 754 format consist of three fields: sign bit, exponent, and fraction. The fraction is the significand without its most significant bit.

Exponent biasing

Main article: Exponent bias

Instead of being stored in two's complement format, the exponent is stored in "biased format" (offset binary): a constant, called the *bias*, is added to it so that the lowest representable exponent is represented as 1, and there is no sign bit. The bias is equal to $2^{n-1}-1$, where n is the number of bits in the exponent field. Thus the actual number stored in the exponent field is the true exponent plus the bias.

For example, in single-precision format, where the exponent field is 8 bits long, the bias is $2^{8-1}-1 = 128 - 1 = 127$. So, for example, an exponent of 17 would be represented as 144 in single precision ($144 = 17 + 127$).

Cases

The most significant bit of the significand (not stored) is determined by the value of *biased exponent*. If $0 < \textit{biased exponent} < 2^n - 1$ (where n is the number of bits in the exponent field), the most significant bit of the *significand* is 1, and the number is said to be *normalized*. If *biased exponent* is 0 and fraction is not 0, the most significant bit of the *significand* is 0 and the number is said to be *de-normalized*. Three other special cases arise:

1. If *biased exponent* is 0 and *fraction* is 0, the number is ±0 (depending on the sign bit)
2. If *biased exponent* = $2^n - 1$ and *fraction* is 0, the number is ±infinity (again depending on the sign bit), and
3. If *biased exponent* = $2^n - 1$ and *fraction* is not 0, the number being represented is not a number (NaN).

This can be summarized as:

Type	Biased Exponent	Fraction
Zeroes	0	0
Denormalized numbers	0	non zero
Normalized numbers	1 to $2^n - 2$	any
Infinities	$2^n - 1 =$ all 1's	0
NaNs	$2^n - 1 =$ all 1's	non zero

Single-precision 32-bit

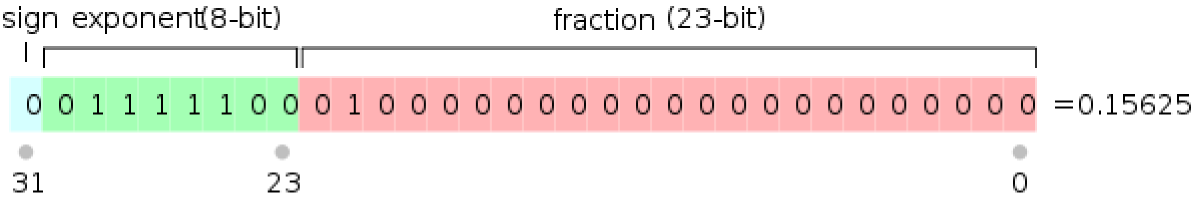


Figure 3 The number 0.15625 represented as a single-precision IEEE 754-1985 floating-point number

A single-precision binary floating-point number is stored in 32 bits, as shown above.

If the number is normalized (the most common case), it has this value:

$$(-1)^{sign} \times 1.fraction \times 2^{exp-127}$$

where:

sign = the sign bit. (0 makes the number positive or zero; 1 makes it negative.)
exp = the 8-bit number stored in the exponent field. *exp* = biased exponent = the true exponent + 127. (See above for why.)
fraction = the 23 bits of the fraction field.

In the example, the *sign* is 0, so the number is positive; *exp* is 124, so the true exponent is –3; and *fraction* is .01. So, the represented number is:

$$1.01_2 \times 2^{-3} = 1.25_{10} \times \frac{1}{8} = 0.15625$$

(Subscripts indicate the base of the number: base 2 or base 10.)

Notes

1. Denormalized numbers are the same except that exp = −126 and m is 0.fraction. (exp is *not* −127 : The fraction has to be shifted to the right by one more bit, in order to include the leading bit, which is not always 1 in this case. This is balanced by incrementing the biased exponent to −126 for the calculation.)
2. −126 is the smallest exponent for a normalized number
3. There are two Zeroes, +0 (sign bit is 0) and −0 (sign bit is 1)
4. There are two Infinities +∞ (sign bit is 0) and −∞ (sign bit is 1)
5. NaNs may have a sign and a fraction, but these have no meaning other than for diagnostics; the first bit of the fraction is often used to distinguish *signaling NaNs* from *quiet NaNs*
6. NaNs and Infinities have all 1s in the exponent field.
7. The positive and negative numbers closest to zero (represented by the denormalized value with all 0s in the exponent field and the binary value 1 in the fraction field) are

   $\pm 2^{-149} \approx \pm 1.4012985 \times 10^{-45}$

8. The positive and negative normalized numbers closest to zero (represented with the binary value 1 in the exponent field and 0 in the fraction field) are

   $\pm 2^{-126} \approx \pm 1.175494351 \times 10^{-38}$

9. The finite positive and finite negative numbers furthest from zero (represented by the value with 254 in the exponent field and all 1s in the fraction field) are

   $\pm(1-2^{-24}) \times 2^{128} \approx \pm 3.4028235 \times 10^{38}$

Here is the summary table from the previous section with some 32-bit single-precision examples:

Type	Sign	Actual Exponent	Exp (biased)	Exponent field	Significand (fraction field)	Value
Zero	0	-127	0	0000 0000	000 0000 0000 0000 0000 0000	0.0
Negative zero	1	-127	0	0000 0000	000 0000 0000 0000 0000 0000	−0.0
One	0	0	127	0111 1111	000 0000 0000 0000 0000 0000	1.0
Minus One	1	0	127	0111 1111	000 0000 0000 0000 0000 0000	−1.0

Smallest denormalized number	*	-127	0	0000 0000	000 0000 0000 0000 0000 0001	$\pm2^{-23} \times 2^{-126} = \pm2^{-149} \approx \pm1.4\times10^{-45}$
"Middle" denormalized number	*	-127	0	0000 0000	100 0000 0000 0000 0000 0000	$\pm2^{-1} \times 2^{-126} = \pm2^{-127} \approx \pm5.88\times10^{-39}$
Largest denormalized number	*	-127	0	0000 0000	111 1111 1111 1111 1111 1111	$\pm(1-2^{-23}) \times 2^{-126} \approx \pm1.18\times10^{-38}$
Smallest normalized number	*	-126	1	0000 0001	000 0000 0000 0000 0000 0000	$\pm2^{-126} \approx \pm1.18\times10^{-38}$
Largest normalized number	*	127	254	1111 1110	111 1111 1111 1111 1111 1111	$\pm(2-2^{-23}) \times 2^{127} \approx \pm3.4\times10^{38}$
Positive infinity	0	128	255	1111 1111	000 0000 0000 0000 0000 0000	$+\infty$
Negative infinity	1	128	255	1111 1111	000 0000 0000 0000 0000 0000	$-\infty$
Not a number	*	128	255	1111 1111	non zero	NaN
* Sign bit can be either 0 or 1 .						

Range and Precision Table

Precision is defined as the minimum difference between two successive mantissa representations; thus it is a function only in the mantissa; while the gap is defined as the difference between two successive numbers.

Some example range and gap values for given exponents:

Actual Exponent (unbiased)	Exp (biased)	Minimum	Maximum	Gap
0	127	1	1.999999880791	1.19209289551e-7
1	128	2	3.99999976158	2.38418579102e-7
2	129	4	7.99999952316	4.76837158203e-7
10	137	1024	2047.99987793	1.220703125e-4
11	138	2048	4095.99975586	2.44140625e-4
23	150	8388608	16777215	1
24	151	16777216	33554430	2
127	254	1.7014e38	3.4028e38	2.02824096037e31

As an example, 16,777,217 cannot be encoded as a 32-bit float as it will be rounded to 16,777,216. This shows why floating point arithmetic is unsuitable for accounting software. However, all integers within the representable range that are a power of 2 can be stored in a 32-bit float without rounding.

A more complex example

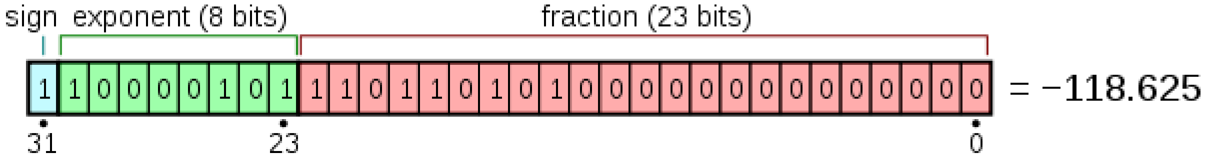
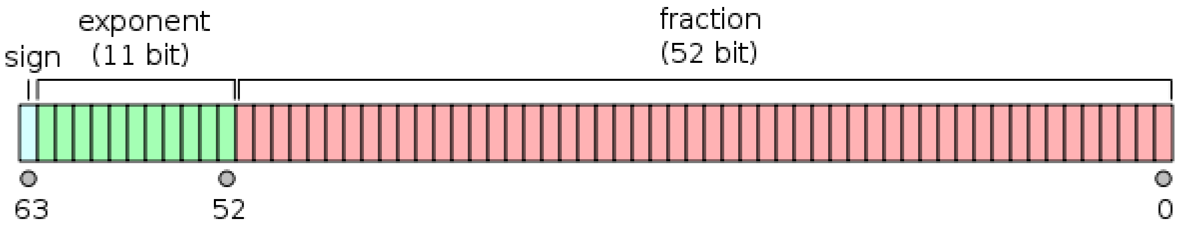


Figure 4 Bit values for the IEEE 754 32bit float -118.625

The decimal number −118.625 is encoded using the IEEE 754 system as follows:

1. The sign, the exponent, and the fraction are extracted from the original number. Because the number is negative, the sign bit is "1".
2. Next, the number (without the sign; i.e., unsigned, no two's complement) is converted to binary notation, giving 1110110.101. The 101 after the binary point has the value 0.625 because it is the sum of:
    1. $(2^{-1}) \times 1$, from the first bit after the binary point
    2. $(2^{-2}) \times 0$, from the second bit
    3. $(2^{-3}) \times 1$, from the third bit.
3. That binary number is then *normalized*; that is, the binary point is moved left, leaving only a 1 to its left. The number of places it is moved gives the (power of two) exponent: 1110110.101 becomes 1.110110101×2^6. After this process, the first binary digit is always a 1, so it need not be included in the encoding. The rest is the part to the right of the binary point, which is then padded with zeros on the right to make 23 bits in all, which becomes the significand bits in the encoding: That is, 11011010100000000000000.
4. The exponent is 6. This is encoded by converting it to binary and biasing it (so the most negative encodable exponent is 0, and all exponents are non-negative binary numbers). For the 32-bit IEEE 754 format, the bias is +127 and so 6 + 127 = 133. In binary, this is encoded as 10000101.

Double-precision 64 bit



The three fields in a 64bit IEEE 754 float

Double precision is essentially the same except that the fields are wider:

The fraction part is much larger, while the exponent is only slightly larger. NaNs and Infinities are represented with Exp being all 1s (2047). If the fraction part is all zero then it is Infinity, else it is NaN.

For Normalized numbers the exponent bias is +1023 (so actual exponent is Exp − 1023). For Denormalized numbers the exponent bias is −1022 (so actual exponent is Exp −1022) because normalized numbers have a leading 1 digit before the binary point and denormalized numbers do not. As before, both infinity and zero are signed.

Notes:

1. The positive and negative numbers closest to zero (represented by the denormalized value with all 0s in the Exp field and the binary value 1 in the Fraction field) are

$$\pm 2^{-1074} \approx \pm 5\times 10^{-324}$$

2. The positive and negative normalized numbers closest to zero (represented with the binary value 1 in the Exp field and 0 in the fraction field) are

$$\pm 2^{-1022} \approx \pm 2.2250738585072020\times 10^{-308}$$

3. The finite positive and finite negative numbers furthest from zero (represented by the value with 2046 in the Exp field and all 1s in the fraction field) are

$$\pm((1-(1/2)^{53})2^{1024}) \approx \pm 1.7976931348623157\times 10^{308}$$

Some example range and gap values for given exponents:

Actual Exponent (unbiased)	Exp (biased)	Minimum	Maximum	Gap
0	1023	1	1.9999999999999997	2.2204460492503130808472633e-16
1	1024	2	3.9999999999999995	8.8817841970012523233890533447266e-16
2	1025	4	7.9999999999999990	3.5527136788005009293556213378906e-15
10	1033	1024	2047.9999999999997	2.27373675443232059478759765625e-13
11	1034	2048	4095.9999999999995	4.547473508864641189575195312 5e-13
52	1075	4503599627370496	9007199254740991	1
53	1076	9007199254740992	18014398509481982	2
1023	2046	8.9884656743115800e+307	1.7976931348623157e308	1.9958403095347198116563727130368e292