Thermo
SCIENTIFIC

Ramsey
MINI CK100
Weight Integrator

MODBUS - TCP
Manual

MINI CK100
Part No.: Modbus Tcp

DOCUMENT HISTORY

DATE	REV	REASON FOR CHANGE	AUTHOR
November, 2009	X1	Issue	Mazzoni Massimo

File: \\ 100_ModbusTcp-X1

Actual Software Release: 92.00.00.04

INDEX

About this manual

The *Modbus-Tcp protocol* manual consists of three parts.

Chapter 1, Brief overview of the MODBUS-TCP protocol.

Chapter 2, Explains how MODBUS-TCP protocol has been implemented in the Ramsey Mini CK instruments. The information contained in this section should be intent as general information, they are valid for all the MINI CK Series models.

Chapter 3, Detailed description of the data that can be transmitted and received to and from the instrument. This part is specific for the instrument model.

Bibliography

- Modbus messaging on TCP/IP implementation guide V1.0b

1. MODBUS-TCP OVERVIEW

1.1 Introduction to the MODBUS-TCP

MODBUS® TCP/IP is an Internet protocol. The fact that TCP/IP is the transport protocol of the Internet automatically means that MODBUS® TCP/IP can be used over the Internet! Among other things it was designed to reach this goal, and as part of this goal the MODBUS® protocol specification has been submitted to the Internet Engineering Task Force (IETF). In practical terms, this means that a MODBUS® TCP/IP device installed in Europe can be addressed over the Internet from the USA from anywhere else in the world.
The implications for a vendor of equipment or an end-user are endless.

• Performing maintenance and repair on remote devices from the office using a PC and browser reduce support costs and improve customer service.

• Logging onto a plant's control system from home allows the maintenance engineer to maximize his plant's uptime and reduce the number of times that he is called out from home.

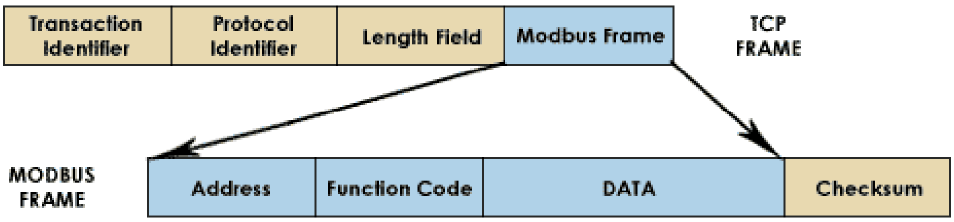
• Managing geographically distributed systems becomes easy using commercially available internet/intranet technologies.

MODBUS® TCP/IP has became an industry de facto standard because of its openness, simplicity, low cost development, and minimum hardware required to support it.
At this moment there are more than 200 MODBUS® TCP/IP devices available in the market. It is used to exchange information between devices, monitor and program them. It is also used to manage distributed I/Os, being the preferred protocol by the manufacturers of this type of devices.

Combining a versatile, scaleable, and ubiquitous physical network (Ethernet) with a universal networking standard (TCP/IP) and a vendor-neutral data representation (MODBUS®) gives a truly open, accessible network for exchange of process data

1.2 The MODBUS-TCP

Modbus/TCP basically embeds a Modbus frame into a TCP frame in a simple manner. This is a connection-oriented transaction which means every query expects a response.
This query/response technique fits well with the master/slave nature of Modbus, adding to the deterministic advantage that Switched Ethernet offers industrial users. The use of OPEN Modbus within the TCP frame provides a totally scaleable solution from ten nodes to ten thousand nodes without the risk of compromise that other multicast techniques would give.
The following figure shows a **modbus tcp** message enclosed in the TCP message:



Performance from a MODBUS TCP/IP system

The performance basically depends on the network and the hardware. If you are running MODBUS® TCP/IP over the Internet, you won't get better than typical Internet response times. However, for communicating for debug and maintenance purposes, this may be perfectly adequate.
For a high-performance Intranet with high-speed Ethernet switches to guarantee performance, the situation is completely different.
In theory MODBUS® TCP/IP carries data at up to 250/(250+70+70) or about 60% efficiency when transferring registers in bulk, and since 10 Base T Ethernet carries about 1.25 Mbytes/sec raw, the theoretical throughput is:
1.25M / 2 * 60% = 360000 registers per second and the 100 Base T speed is 10 x greater.
This assumes that you are using devices that can service Ethernet as fast as bandwidth is available.

1.3 Client – Server model

The MODBUS messaging service provides a Client/Server communication between devices connected on an Ethernet TCP/IP network.
This client / server model is based on four type of messages:

- **MODBUS Request,**
- **MODBUS Confirmation,**
- **MODBUS Indication,**
- **MODBUS Response**

A MODBUS Request is the message sent on the network by the Client to initiate a transaction,
A MODBUS Indication is the Request message received on the Server side,
A MODBUS Response is the Response message sent by the Server,
A MODBUS Confirmation is the Response Message received on the Client side

The MODBUS messaging services (Client / Server Model) are used for real time information exchange:

- between two device applications,
- between device application and other device,
- between HMI/SCADA applications and devices,
- between a PC and a device program providing on line services.

1.4 Abbreviations

ADU Application Data Unit
IETF Internet Engineering Task Force
IP Internet Protocol
MAC Medium Access Control
MB MODBUS
MBAP MODBUS Application Protocol
PDU Protocol Data Unit
PLC Programmable Logic Controller
TCP Transport Control Protocol
BSD Berkeley Software Distribution
MSL Maximum Segment Lifetime

2 IMPLEMENTATION ON THE MINI CK-100

2.1 Protocol description.

2.1.1 General communication architecture

A communicating system over MODBUS TCP/IP may include different types of device:
• A MODBUS TCP/IP Client and Server devices connected to a TCP/IP network
• The Interconnection devices like bridge, router or gateway for interconnection between the TCP/IP network and a serial line sub-network which permit connection of MODBUS Serial line Client and Server end devices.

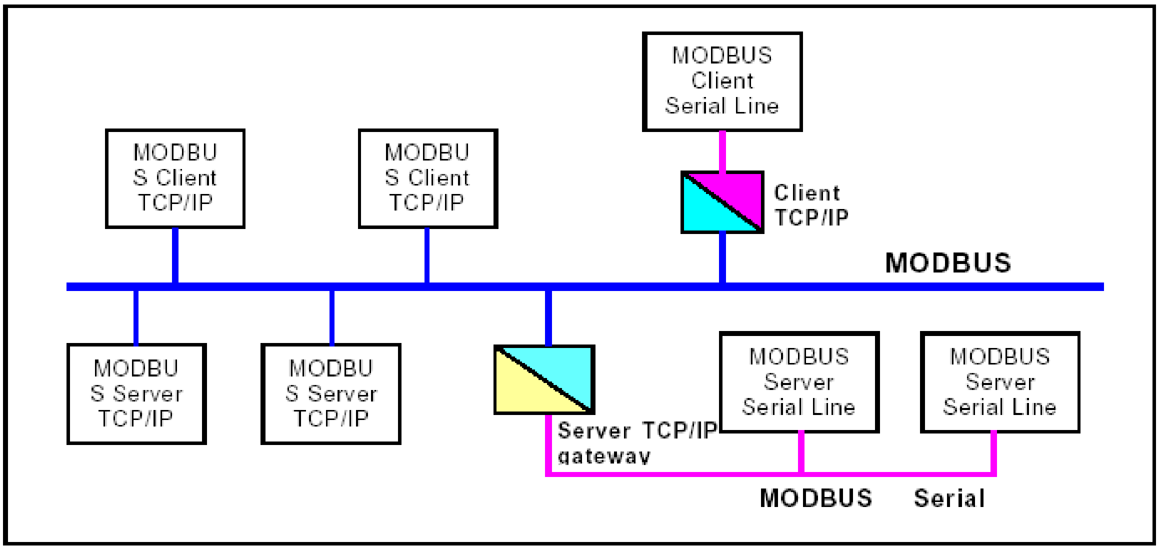
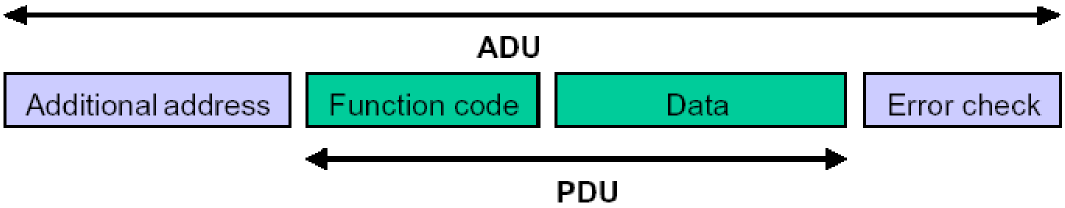


Figure 1: MODBUS TCP/IP communication architecture

The MODBUS protocol defines a **simple Protocol Data Unit (PDU)** independent of the underlaying communication layers. The mapping of MODBUS protocol on specific buses or networks can introduce some additional fields on the **Application Data Unit (ADU)**.

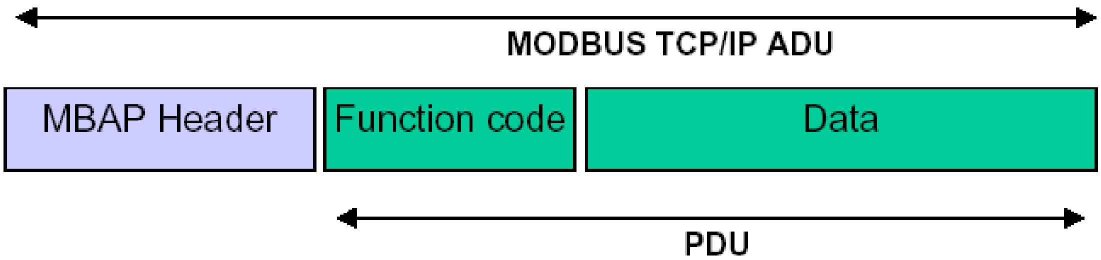


General MODBUS Frame

The client that initiates a MODBUS transaction builds the MODBUS Application Data Unit. The function code indicates to the server which kind of action to perform.

2.1.2 MODBUS on TCP/IP Application Data Unit

This section describes the encapsulation of a MODBUS request or response when it is carried on a MODBUS TCP/IP network.



Modbus request/response over TCP/IP

A dedicated header is used on TCP/IP to identify the MODBUS Application Data Unit. It is called the MBAP header (MODBUS Application Protocol header).
This header provides some differences compared to the MODBUS RTU application data

unit used on serial line:
The MODBUS ‘slave address’ field usually used on MODBUS Serial Line is replaced by a single byte ‘Unit Identifier’ within the MBAP Header. The ‘Unit Identifier’ is used to communicate via devices such as bridges, routers and gateways that use a single IP address to support multiple independent MODBUS end units.
All MODBUS requests and responses are designed in such a way that the recipient can verify that a message is finished. For function codes where the MODBUS PDU has a fixed length, the function code alone is sufficient. For function codes carrying a variable amount of data in the request or response, the data field includes a byte count.
When MODBUS is carried over TCP, additional length information is carried in the MBAP header to allow the recipient to recognize message boundaries even if the message has been split into multiple packets for transmission. The existence of explicit and implicit length rules, and use of a CRC-32 error check code (on Ethernet) results in an infinitesimal chance of undetected corruption to a request or response message.

2.1.3 MBAP header description

The MBAP Header contains the following fields:

Fields	Length	Description -	Client	Server
Transaction Identifier	2 Bytes	Identification of a MODBUS Request / Response transaction.	Initialized by the client	Recopied by the server from the received request
Protocol Identifier	2 Bytes	0 = MODBUS protocol	Initialized by the client	Recopied by the server from the received request
Length	2 Bytes	Number of following bytes	Initialized by the client (request)	Initialized by the server (Response)
Unit Identifier	1 Byte	Identification of a remote slave connected on a serial line or on other buses.	Initialized by the client	Recopied by the server from the received request

The header is 7 bytes long:

• **Transaction Identifier** - It is used for transaction pairing, the MODBUS server copies in the response the transaction identifier of the request.

- **Protocol Identifier** – It is used for intra-system multiplexing. The MODBUS protocol is identified by the value 0.
- **Length** - The length field is a byte count of the following fields, including the Unit Identifier and data fields.
- **Unit Identifier** – This field is used for intra-system routing purpose. It is typically used to communicate to a MODBUS+ or a MODBUS serial line slave through a gateway between an Ethernet TCP-IP network and a MODBUS serial line. This field is set by the MODBUS Client in the request and must be returned with the same value in the response by the server.

All MODBUS/TCP ADU are sent via TCP to registered port 502.

3 REGISTERS LIST

3.1 Description of the variables

The Table 1 of the next pages, lists the variables accessible by the Master specifically for the **MINI CK-100 Static Indicator**.

It is necessary at this point make a distinction between register and variables.
A *register* is the basic unit of the data at which the Master can access. The groups are structured in registers, they are numbered and the identification number is used by the Master to identify the portion of a group to read or write. Their dimension is always one word.
A *variable* is the format of storing of the data in the instrument memory. Its dimension can change depending by the variable type, we can have:

CHAR variable	A char is a variable of 1 byte . It contains integer values in the ranges 0 to 255 or +127 to -128. The char variables can be structured in array with various dimensions.
INTEGER variable	An integer is a variable of 1 word (1 register). It contains INTEGER values in the ranges 0 to 65535 or +32767 to -32768. The integer variables can be structured in array of integers with various dimensions.
FLOAT variable	A float is a variable of two words (2 register). It contains REAL values in the single precision IEEE format (See Appendix A at the end of this document). The single precision format can represent values in the range $3.4*10^{+38}$ to $1.18*10^{-38}$
LONG variable	A long is a variable of two words (2 register). It contains INTEGER values in the ranges 0 to 4294967295 or +2147483647 to -2147483648. The integer variables can be structured in array of integers with various dimensions.

NOTE
A lot of real values are displayed on the instrument with a number of decimals that depends by the selected division. For what regards the communication, instrument registers always contain the value with all the decimals.
For example:

Net weight	Displayed value:	10.2
	Sent value:	10.179982

LEGEND:

Register	Conventional name	
Type	Can be:	RO The register can be read but can not be written. It will be inserted only in the read groups.
		RW The register can be read or written. It will be found either in the read or in the write groups.
		WO The register can only be written. It will be inserted only in the read groups.
Low limit	Minimum acceptable value for the variable. Lower values are considered as errors.	
High limit	Maximum acceptable value for the variable. Higher values are considered as errors.	
Refresh time	Time between two updates of the variable in the instrument's memory.	
format	Can be: char , integer, float or double	
Address (word)	Number of word (decimal) in the mapping, at which the register can be found.	
Note	Comments and/or special info on use.	

3.2 List of registers

Register	Type	Low Limit	High Limit	Refresh Time [ms]	Format	Address (word)	Note
Simulation Key	WO	(**)	(**)	-	integer	0	
Write flag	RO	-	(*)	-	integer	1	
DISPLAY DATA							
Display (1)	RO	-	-	200	integer	2	
Display (2)	RO	-	-	200	integer	3	
Display (3)	RO	-	-	200	integer	4	
Display (4)	RO	-	-	200	integer	5	
Display (5)	RO	-	-	200	integer	6	
Display (6)	RO	-	-	200	integer	7	
Display (7)	RO	-	-	200	integer	8	
Display (8)	RO	-	-	200	integer	9	
Display (9)	RO	-	-	200	integer	10	
Display (10)	RO	-	-	200	integer	11	
Display (11)	RO	-	-	200	integer	12	
Display (12)	RO	-	-	200	integer	13	
Display (13)	RO	-	-	200	integer	14	
Display (14)	RO	-	-	200	integer	15	
Display (15)	RO	-	-	200	integer	16	

(*) Max limit is the number of digital inputs,it depends by the hardware configuration of the instrument.

(**) All the values are accepted but only a specific set of codes are interpreted as keys. See description of the variable.

Register	Type	Low Limit	High Limit	Refresh Time [ms]	Format	Address (word)	Note
Display (16)	RO	-	-	200	integer	17	
Display (17)	RO	-	-	200	integer	18	
Display (18)	RO	-	-	200	integer	19	
Display (19)	RO	-	-	200	integer	20	
Display (20)	RO	-	-	200	integer	21	
Display (21)	RO	-	-	200	integer	22	
Display (22)	RO	-	-	200	integer	23	
Display (23)	RO	-	-	200	integer	24	
Display (24)	RO	-	-	200	integer	25	
Display (25)	RO	-	-	200	integer	26	
Display (26)	RO	-	-	200	integer	27	
Display (27)	RO	-	-	200	integer	28	
Display (28)	RO	-	-	200	integer	29	
Display (29)	RO	-	-	200	integer	30	
Display (30)	RO	-	-	200	integer	31	
Display (31)	RO	-	-	200	integer	32	
Display (32)	RO	-	-	200	integer	33	
Display (33)	RO	-	-	200	integer	34	
Display (34)	RO	-	-	200	Integer	35	
Display (35)	RO	-	-	200	integer	36	
Display (36)	RO	-	-	200	integer	37	
Display (37)	RO	-	-	200	Integer	23	
Display (38)	RO	-	-	200	integer	38	
Display (39)	RO	-	-	200	integer	39	
Display (40	RO	-	-	200	integer	40	

Register	Type	Low Limit	High Limit	Refresh Time [ms]	Format	Address (word)	Note
STATUS AND DIAGNOSTIC DATA							
Panel Leds	RO	-	-	200	Integer	42	
Status (1)	RO	-	-	-	Integer	43	
Status (2)	RO	-	-	-	Integer	44	
Alarms (1)	RO	-	-	100	Integer	45	
Alarms (2)	RO	-	-	100	Integer	46	
In / Out image	RO	-	-	100	Integer	47	
Commands	RW	0	0xffff	-	Integer	48	
Commands 1	RW	0	0xffff	-	Integer	49	
DYNAMIC DATA							
Net weight	RO	-	-	100	float	52	
Gross weight	RO	-	-	100	float	54	
Peak weight	RO	-	-	100	float	56	
Master total	RO	-	-	100	float	58	
Reset total	RO	-	-	100	float	60	
Master total double	RO	-	-	100	double	62	
Reset total double	RO	-	-	100	double	66	
Manual zero	RW	-	-	-	long	72	
Manual span	RW	-	-	-	float	74	

Register	Type	Low Limit	High Limit	Refresh Time [ms]	Format	Address (word)	Note
Linearization							
Linear option	RW	Tab.06	Tab.06	-	integer	78	
Linear weight 1	RW	0	SCALE CAP.	-	float	79	
Linear factor 1	RW	0	1.5	-	float	81	
Linear weight 2	RW	0	SCALE CAP.	-	float	83	
Linear factor 2	RW	0	1.5	-	float	85	
Linear weight 3	RW	0	SCALE CAP.	-	float	87	
Linear factor 3	RW	0	1.5	-	float	89	
Linear weight 4	RW	0	SCALE CAP.	-	float	91	
Linear factor 4	RW	0	1.5	-	float	93	
DISPLAY DATA MENU							
Units	RW	Tab.07	Tab.07	-	integer	101	
Weight units	RW	Tab.08	Tab.08	-	integer	102	
Total units	RW	Tab.09	Tab.09	-	integer	103	
Language	RW	Tab.10	Tab.10	-	integer	104	
Time format	RW	-	-	-	integer	105	
Date format	RW	-	-	-	integer	106	
Line 2	RW	Tab.13	Tab.13	-	integer	107	
Line 3	RW	Tab.14	Tab.14	-	integer	108	
Weight damping	RW	0	16	-	integer	109	
Tare mode	RW	Tab.15	Tab.15	-	integer	110	

Register	type	Low Limit	High Limit	Refresh Time [ms]	Format	Address (word)	Note
SCALE DATA MENU							
Scale capacity	RW	1	1000000	-	float	113	
Scale div.	RW	Tab.16	Tab.16	-	integer	115	
Load cell nr.	RW	1	6	-	integer	120	
Load cell cap.	RW	1	500000	-	float	121	
Load cell sens.	RW	0.5	3.36	-	float	123	
Load cell res1	RW	10	2000	-	float	125	
Load cell res2	RW	10	2000	-	float	127	
Load cell res3	RW	10	2000	-	float	129	
Load cell res4	RW	10	2000	-	float	131	
Load cell res5	RW	10	2000	-	float	133	
Load cell res6	RW	10	2000	-	float	135	
Motion band	RW	0	3	-	integer	137	
Motion band delay	RW	0	60	-	float	139	
CALIBRATION DATA MENU							
Calibrat. mode	RW	Tab.17	Tab.17	-	integer	143	
Test weights	RW	0	SCALE CAP.	-	float	144	
Rcal res	RW	10	1000000	-	long	146	
Rcal calcon	RO	-	-	-	float	148	
Rcal factor	RW	-99.99	+99.99	-	float	150	
Calibr. interval	RW	0	365	-	integer	152	
AZT option	RW	Tab.18	Tab.18	-	integer	153	
AZT range	RW	0	10	-	float	154	
AZT dev	RW	0	10	-	float	156	
AZT duration	RW	2	60	-	Integer	158	

Register	type	Low Limit	High Limit	Refresh Time [ms]	Format	Address (word)	Note
PROTECTION LEVEL MENU							
Protection level	RW	0	2	-	Integer	161	
DIAGNOSTIC MENU							
AD gross	RO	-	-	100	long	162	
AD net	RO	-	-	100	Long	164	
Weight on Load cell (mV)	RO	0	-	100	float	166	
Load cell zero	RW	0	10000	-	Integer	168	
Load cell span	RW	0	30000	-	Integer	169	
Gain	RO	-	-	-	Integer	170	
Drift	RO	-	-	-	Long	171	
Drift reference	RO	-	-	-	Long	173	
Mech. tare	RO	-	-	-	Long	175	
Service password	RW	Tab.19	Tab.19	-	Char (5 word)	177	Ask for 5 Word (10 characters) and use only 4
Operator password	RW	Tab.20	Tab.20	-	Char (5 word)	182	Ask for 5 Word (10 characters) and use only 4
Software version	RO	-	-	-	Char (6 word)	187	12 characters
Day	RW	-	-	-	Integer	193	**Available ONLY if COM board installed**
Month	RW	-	-	-	Integer	194	
Year	RW	-	-	-	Integer	195	
Hour	RW	-	-	-	Integer	196	
Minute	RW	-	-	-	Integer	197	

Register	type	Low Limit	High Limit	Refresh Time [ms]	Format	Address (word)	Note
I/O DATA MENU							
Analog output	RO	-	-	-	Float	198	**Available ONLY if Current Output board installed**
Analog out assign	RW	-	-	-	Integer	200	
Analog out range	RW	-	-	-	Integer	201	
Analog out delay	RW	-	-	-	Integer	202	
Analog out damping	RW	-	-	-	Integer	203	
Digital Input Assignem. Function "*Ext.Alarm 1*"	RW	Tab.21	Tab.21	-	Integer	204	
Digital Input Assignem. Function "*Ext.Alarm 2*"	RW	Tab.21	Tab.21	-	Integer	205	
Digital Input Assignem. Function "*Ext.Alarm 3*"	RW	Tab.21	Tab.21	-	Integer	206	
Digital Input Assignem. Function "*Print*"	RW	Tab.21	Tab.21	-	Integer	207	
Digital Input Assignem. Function "*Reset Alarm*"	RW	Tab.21	Tab.21	-	Integer	208	
Digital Input Assignem. Function "*Reset Tare*"	RW	Tab.21	Tab.21	-	Integer	209	
Digital Input Assignem. Function "*Set Tare*"	RW	Tab.21	Tab.21	-	Integer	210	
Digital Input Assignem. Function "*Reset Total*"	RW	Tab.21	Tab.21	-	Integer	211	
Digital Input Assignem. Function "*Add Total*"	RW	Tab.21	Tab.21	-	Integer	212	
Digital Input Assignem. Function "*Hold*"	RW	Tab.21	Tab.21	-	Integer	213	
Digital Input Assignem. Function "*Peak*"	RW	Tab.21	Tab.21	-	Integer	214	

Register	type	Low Limit	High Limit	Refresh Time [ms]	Format	Address (word)	Note
Digital Output Assign. Function "*Cumulative Alarm*"	RW	Tab.22	Tab.22	-	integer	220	
Digital Output Assign. Function "*Cumulative Shut Down*"	RW	Tab.22	Tab.22	-	integer	221	
Digital Output Assign. Function "*Ready*"	RW	Tab.22	Tab.22	-	integer	222	
Digital Output Assign. Function "*Weight Stable*"	RW	Tab.22	Tab.22	-	integer	223	
Digital Output Assign. Function "*Current Output Polarity*"	RW	Tab.22	Tab.22	-	integer	224	
Digital Output Assign. Function "*Threshold 1*"	RW	Tab.22	Tab.22	-	integer	225	
Digital Output Assign. Function "*Threshold 2*"	RW	Tab.22	Tab.22	-	integer	226	
Digital Output Assign. Function "*Threshold 3*"	RW	Tab.22	Tab.22	-	integer	227	
Digital Output Assign. Function "*Threshold 4*"	RW	Tab.22	Tab.22	-	integer	228	
Digital Output Assign. Function "*Totals Updated*"	RW	Tab.22	Tab.22	-	integer	229	

Register	type	Low Limit	High Limit	Refresh Time [ms]	Format	Address (word)	Note
ALARM THRESHOLD MENU							
Threshold 1 option	RW	Tab.25	Tab.25	-	integer	238	
Threshold 1 set	RW	0	105	-	float	239	
Threshold 1 delay	RW	0	90	-	integer	241	
Threshold 1 hyst.	RW	0	105	-	float	242	
Threshold 1 mode	RW	Tab.26	Tab.26	-	integer	244	
Threshold 1 var	RW	Tab.27	Tab.27	-	integer	245	
Threshold 2 option	RW	Tab.25	Tab.25	-	integer	248	
Threshold 2 set	RW	0	105	-	float	249	
Threshold 2 delay	RW	0	90	-	integer	251	
Threshold 2 hyst.	RW	0	105	-	float	252	
Threshold 2 mode	RW	Tab.26	Tab.26	-	integer	254	
Threshold 2 var	RW	Tab.27	Tab.27	-	integer	255	
Threshold 3 option	RW	Tab.25	Tab.25	-	integer	258	
Threshold 3 set	RW	0	105	-	float	259	
Threshold 3 delay	RW	0	90	-	integer	261	
Threshold 3 hyst.	RW	0	105	-	float	262	
Threshold 3 mode	RW	Tab.26	Tab.26	-	integer	264	
Threshold 3 var	RW	Tab.27	Tab.27	-	integer	265	
Threshold 4 option	RW	Tab.25	Tab.25	-	integer	268	
Threshold 4 set	RW	0	105	-	float	269	
Threshold 4 delay	RW	0	90	-	integer	271	
Threshold 4 hyst.	RW	0	105	-	float	272	
Threshold 4 mode	RW	Tab.26	Tab.26	-	integer	274	
Threshold 4 var	RW	Tab.27	Tab.27	-	integer	275	

Register	type	Low Limit	High Limit	Refresh Time [ms]	Format	Address (word)	Note
ALARMS MENU							
Alarm define “*Clock Fail*”	RW	Tab. 28	Tab. 28	-	integer	278	
Alarm define “*Cell Fail*”	RW	Tab. 28	Tab. 28	-	integer	279	
Alarm define “*Ram Fail*”	RW	Tab. 28	Tab. 28	-	integer	280	
Alarm define “*Eeprom Fail*”	RW	Tab. 28	Tab. 28	-	integer	281	
Alarm define “*Threshoald 1*”	RW	Tab. 28	Tab. 28	-	integer	282	
Alarm define “*Threshoald 2*”	RW	Tab. 28	Tab. 28	-	integer	283	
Alarm define “*Threshoald 3*”	RW	Tab. 28	Tab. 28	-	integer	284	
Alarm define “*Threshoald 4*”	RW	Tab. 28	Tab. 28	-	integer	285	
Alarm define “*Warm Start*”	RW	Tab. 28	Tab. 28	-	integer	286	
Alarm define “*Cold Start*”	RW	Tab. 28	Tab. 28	-	integer	287	
Alarm define “*Calibration Aborted*”	RW	Tab. 28	Tab. 28	-	integer	288	
Alarm define “*Calibration Time*”	RW	Tab. 28	Tab. 28	-	integer	289	
Alarm define “*External Alarm 1*”	RW	Tab. 28	Tab. 28	-	integer	290	
Alarm define “*External Alarm 2*”	RW	Tab. 28	Tab. 28	-	integer	291	
Alarm define “*External Alarm 3*”	RW	Tab. 28	Tab. 28	-	integer	292	
Alarm define “*Auto Zero Track. Limit*”	RW	Tab. 28	Tab. 28	-	integer	293	
Alarm define “*Math Error*”	RW	Tab. 28	Tab. 28	-	integer	294	

Register	type	Low Limit	High Limit	Refresh Time [ms]	Format	Address (word)	Note
Alarm define "*Printer Error*"	RW	Tab. 28	Tab. 28	-	integer	295	
Alarm define "*Communication Error*"	RW	Tab. 28	Tab. 28	-	integer	296	
Alarm define "*Profibus Error*"	RW	Tab. 28	Tab. 28	-	integer	297	
Alarm define "*Calibration Alarm*"	RW	Tab. 28	Tab. 28	-	integer	298	
Alarm define "*Dynamic Data Lost*"	RW	Tab. 28	Tab. 28	-	integer	299	
Alarm define "*Setup Data Lost*"	RW	Tab. 28	Tab. 28	-	integer	300	
Alarm define "*Profibus Config. Error*"	RW	Tab. 28	Tab. 28	-	integer	301	
Alarm define "*Calibration Error*"	RW	Tab. 28	Tab. 28	-	integer	302	
NET MENU							
I.P. Address	RW	0.0.0.0	254.255. 255.255	-	Char (8 word)	310	
Net Mask Address	RW	0.0.0.0	255.255. 255.255	-	Char (8 word)	318	
Gateway	RW	0.0.0.0	255.255. 255.255	-	Char (8 word)	326	
Mac Address	RW	0.0.0.0	255.255. 255.255	-	Char (8 word)	334	
Com. Variable Selection	RW	Tab.28	Tab.28	-	integer	338	
Swap Int Data	R	Tab.29	Tab.29	-	integer	339	
Swap Float Data	RW	Tab.39	Tab.39	-	integer	340	
Swap Long Data	RW	Tab.39	Tab.39	-	integer	341	

Tab.4 – Register List

4.4 Description of Registers

WARNING !!!
If data READED / WROTE are outside the limits indicated on the table 4, the Function Code field of the response, is setted by CK with an “Exception Code” (MSB =1) with a value above 80 hex.

Sim_Key The master has the possibility to send a key code to the instrument. Interpreted key codes are:

Key	**Key Code**	**Key**	**Key Code**
UP ARROW	0048 H	F1	003F H
DOWN ARROW	0050 H	F2	0040 H
RUN	0052 H	F3	0041 H

Tab 5 - key codes

Write Flag Set to 0 after a message has been received and properly processed.
If a message is correctly received but cannot be processed because password protection or size errors, this flag is set to 1. The user may read this register after a write message to ensure the data have been accepted.

4.4.1 Display data

Display Contains the messages actually shown on the display of the instrument in form of an ASCII string. For example the following screen:

```
-          MENU MAIN 1        -
 PRESS UP/DOWN KEY FOR MORE
 ZERO    SPAN     MAT'L
 CAL     CAL      CAL
```

Will be stored in registers in the following way:

Display(1)	2DH	20H	Characters 1 and 2 from left of first row
Display(2)	4DH	45H	
Display(3)	4EH	55H	
Display(4)	20H	4DH	
Display(5)	41H	49H	
Display(6)	4EH	20H	
…..			
Display(10)	….	….	Characters 19 and 20 from left of first row
Display (11)	….	….	Characters 1 and 2 from left second row
…..			
Display(20)	….	….	Characters 19 and 20 from left of second row
Display(21)	….	….	Characters 1 and 2 from left of third row
…..			
Display(30)	….	….	Characters 19 and 20 from left of third row
Display(31)	….	….	Characters 1 and 2 from left of fourth row
…..			
Display(38)	4CH	20H	
Display(39)	20H	20H	
Display(40)	20H	20H	Characters 19 and 20 from left of fourth row

4.4.2 Status & Diagnostic

Panel Leds

The word below indicates the status of the 5 leds on the front panel of the instrument.

Panel Led

NU	NU	NU	NU	NU	NU	NU	NU	NU	NU	NU	04	03	02	01	00

PL1.04 Led NET (Tare) led
PL1.03 Led Stab Weight led
PL1.02 Zero led
PL1.01 Ready (rdy) led
PL1.00 Alarm (Alr) led

Status

The actual status of the instrument is resumed in four words, each bit has an own meanings, when the bit is 1 the associated status is true.

Status 1 - Generals

15	14	13	12	11	10	NU	08	07	06	05	04	NU	NU	NU	00

S1.15 Cumulative shut down
S1.14 Cumulative Alarms
S1.13 Calibration running
S1.12 Weight stable
S1.11 Center of zero
S1.10 Tare acquired
S1.09 Not used
S1.08 Ready
S1.07 Threshold 1
S1.06 Threshold 2
S1.05 Threshold 3
S1.04 Threshold 4
S1.03 not used
S1.02 not used
S1.01 not used
S1.00 success flag

Status 2 - Free

NU	NU	NU	NU	NU	NU	NU	NU	NU	NU	NU	NU	NU	NU	NU	NU

S2.15 Free
S2.14 Free
S2.13 Free
S2.12 Free
S2.11 Free
S2.10 Free
S2.09 Free
S2.08 Free
S2.07 Free
S2.06 Free
S2.05 Free
S2.04 Free
S2.03 Free
S2.02 Free
S2.01 Free
S2.00 Free

Alarms

In the alarms register, each bit represents the status of an alarm. If the alarm is active, the relevant bit will contain '1', otherwise it will contain '0'.

Alarms 1

NU	14	NU	12	11	10	09	08	07	06	05	04	03	02	01	00

A1.15 Free
A1.14 CELL FAIL
A1.13 free
A1.12 EPROM FAIL
A1.11 THRESHOLD 1
A1.10 THRESHOLD 2
A1.09 THRESHOLD 3
A1.08 THRESHOLD 4
A1.07 WARM START
A1.06 COLD START
A1.05 PWD DURING CALIBRATION
A1.04 CAL. TIME ELAPSED
A1.03 EXTERNAL ALARM 1
A1.02 EXTERNAL ALARM 2
A1.01 EXTERNAL ALARM 3
A1.00 AZT OVER LIMIT

Alarms 2

NU	NU	NU	12	11	10	09	NU	NU	NU	NU	NU	NU	NU	NU	NU

A2.15 Free
A2.14 Free
A2.13 Free
A2.12 PROFIBUS COMM.ERROR
A2.11 CALIBRATION CHANGED
A2.10 DYN DATA LOST
A2.09 SETUP DATA LOST
A2.08 Free
A2.07 Free
A2.06 Free
A2.05 Free
A2.04 Free
A2.03 Free
A2.02 Free
A2.01 Free
A2.00 Free

I/O Image

The instrument has physical inputs and outputs to which logical input and output functions are associated. As far as communication is concerned, only the status of physical inputs and outputs are transferred.

Input/Output Image – Inputs/Output installed on board

NU	NU	NU	NU	NU	NU	09	08	NU	NU	NU	04	03	02	01	00

I1.15	Not Used	O1.07	Not Used
I1.14	Not Used	O1.06	Not Used
I1.13	Not Used	O1.05	Not Used
I1.12	Not Used	O1.04	output 5 - mother board
I1.11	Not Used	O1.03	output 4 - mother board
I1.10	Not Used	O1.02	output 3 - mother board
I1.09	in 2 - mother board	O1.01	output 2 - mother board
I1.08	in 2 - mother board	O1.00	output 1 - mother board

Commands

Each bit of the commands register is specified as follows. In order to give a command, the Host must set the relevant bit to 1 and write (send) the register to the instrument. The action will be performed if the write message is accepted.

Commands

15	NU	NU	NU	NU	NU	NU	NU	NU	NU	05	04	03	02	01	00

C1.15	SAVE SETUP DATA	C1.07	Free
C1.14	Free	C1.06	Free
C1.13	Free	C1.05	CLEAR PEAK
C1.12	Free	C1.04	CLEAR RESET TOTAL
C1.11	Free	C1.03	UPDATE TOTALS
C1.10	Free	C1.02	RESET TARE
C1.09	Free	C1.01	SET TARE
C1.08	Free	C1.00	RESET ALARM

Reset Commands

When the instrument receives a ‘Clear Reset total’ or a ‘Reset alarm’ command, it executes the command for the scale :

Clear Reset total	Clear the reset total register.
Reset Alarms	Reset any pending alarm

Commands 1

NU	NU	NU	NU	NU	NU	NU	NU	NU	NU	NU	NU	NU	NU	NU	00

C1.15 Free
C1.14 Free
C1.13 Free
C1.12 Free
C1.11 Free
C1.10 Free
C1.09 Free
C1.08 Free

C1.07 Free
C1.06 Free
C1.05 Free
C1.04 Free
C1.03 Free
C1.02 Free
C1.01 Free
C1.00 RESET INSTRUMENT

In order to give a command, the Host must set the relevant bit to 1 and write (send) the register to the instrument. The action will be performed if the write message is accepted.

4.4.3 Dynamic data

Net weight The instantaneous net weight in engineering units. .
Gross weight The instantaneous gross weight in engineering units. .
Peak weight The absolute highest value of the actual net weight .
Master Total The current value of the master Totalizer of the static indicator.
Reset Total The current value of the Reset Totalizer of the static indicator.
Master Total double The current value of the master Totalizer of the static indicator in double format
Reset Total double The current value of the Reset Totalizer of the static indicator in double format.
Manual Zero The value in engineering units of the "Zero" constant of the scale.
Manual Span The value in engineering units of the "Span" constant of the scale.

4.4.4 Linearization

Linear option The enable option for linearization function.

SETTINGs	CODE
NO	00H
YES	01H

Tab 6 – Linearization option limits

Linear weight 1 This value is a known test weight(s) or a bin with pre-weighed material . Pressing the ACQuire soft key display the scale weight for the applied known weight for 1st step.
Linear factor 1 Linearization factor as applicable to the 1st weight zone .
Linear weight 2 This value is a known test weight(s) or a bin with pre-weighed material . Pressing the ACQuire soft key display the scale weight for the applied known weight for 2nd step.
Linear factor 2 Linearization factor as applicable to the 2nd weight zone .
Linear weight 3 This value is a known test weight(s) or a bin with pre-weighed material . Pressing the ACQuire soft key display the scale weight for the applied known weight for 3rd step.
Linear factor 3 Linearization factor as applicable to the 3rd weight zone .
Linear weight 4 This value is a known test weight(s) or a bin with pre-weighed material . Pressing the ACQuire soft key display the scale weight for the applied known weight for 4th step.
Linear factor 4 Linearization factor as applicable to the 4th weight zone .

4.4.5 Display Data

Units Code of the "Measure Units" displayed according the table below:

SETTINGs	CODE
METRIC	00H
ENGLISH	01H
MIXED	02H

Tab 7 – Measure units codes

Weight Units Code of the "Weight Units" displayed according the table below:

SETTINGs	CODE
PERCENT %	00H
KG	01H
TONNES	02H
POUNDS	03H
TONS	04H
LTONS	05H

Tab 8 – Weight units code

Total Units Code of the "Totalization Units" displayed according the table below:

SETTINGs	CODE
KG	00H
TONNES	01H
POUNDS	02H
TONS	03H
LTONS	04H

Tab 9 – Total units code

Language Code of the setted "Language" according the table below

SETTINGs	CODE
USA	00H
SPANISH	01H
ITALIAN	02H
FRENCH	03H
GERMAN	04H
DUTCH	05H

Tab 10 – Language units code

Line 2 Display Code of the setted "Line 3 Display" according the table below:

SETTINGs	CODE
NONE	00H
GROSS	01H
TARE	02H
PEAK	03H
RESET TOTAL	04H
MASTER TOTAL	05H
DATE & TIME	06H
BARGRAPH	07H

Tab 13 – Line 2 display format units code

Line 3 Display Code of the setted "Line 3 Display" according the table below:

SETTINGs	CODE
NONE	00H
GROSS	01H
TARE	02H
PEAK	03H
RESET TOTAL	04H
MASTER TOTAL	05H
DATE & TIME	06H
BARGRAPH	07H

Tab 14 – Line 3 display format units code

Weight Damping The Value in seconds of the displayed "Rate Damping".

Tare mode Select the mode to acquire the tare weight**:**

SETTINGs	CODE
ACQUIRED	00H
MANUAL	01H

Tab 15 – Tare mode units code

4.4.6 Scale Data

Scale Capacity The maximum weight of the indicator, entered by the user in the instrument setup. It is the reference value for the thresholds 1, 2, 3, 4 .

Scale Division Code of the Scale decimal places (Division) actually setted, according the table below:

SETTINGs	CODE
50	00H
20	01H
10	02H
5	03H
2	04H
1	05H
0,5	06H
0,2	07H
0,1	08H
0,05	09H
0,02	0AH
0,01	0BH
0,005	0CH
0,002	0DH
0,001	0EH

Tab 16 – Scale division code

Load Cell numbers The numbers of the load cells..

Load Cell Capacity The value in engineering units of the Capacity of the load cell, as it appear on the label.

Load Cell Sensitivity The value in mV/V of the load cell Sensitivity.

Load Cell Resistance 1 The value in engineering units of the Input Resistance of the load cell N° 1

Load Cell Resistance 2 The value in engineering units of the Input Resistance of the load cell N° 2

Load Cell Resistance 3 The value in engineering units of the Input Resistance of the load cell N° 3

Load Cell Resistance 4 The value in engineering units of the Input Resistance of the load cell N° 4.

Load Cell Resistance 5 The value in engineering units of the Input Resistance of the load cell N° 5

Load Cell Resistance 6 The value in engineering units of the Input Resistance of the load cell N° 6

Motion Band Define the motion band for stable weight indication.

Motion band delay Define the motion delay for stable weight indication..

4.4.7 Calibration Data

Calibration Mode Code of the “Calibration mode” selected according the table below:

SETTINGs	CODE
R-CAL	00H
WEIGHT	01H
2 POINTS	02H

Tab 17– Calibration mode code

Test Weights The value in engineering units of the sample weight used for calibration.

Calibration Resistance The value of the resistor used for the “Calibration with Resistance”.

Calibration Constant The value of the constant for the resistance calculated by instrument.

RCal Factor Number in percentage used to correct the “Calibration Constant” value to the real

Calibration Interval Number of selected days between two calibration.

Azt opt. The enable option for AZT function:

SETTINGs	CODE
NO	00H
YES	01H

Tab 18 – Azt option limits

Azt Range The value in percentage of the Range of action of AutoZero tracking with reference to the scale capacity.

Azt Deviation The value in engineering units of the Maximum amount of zero error, that the AutoZero tracking can automatically compensate.

Azt Duration The value in engineering units of the azt duration time.

4.4.8 Diagnostic data

Protection Level	Protection level of the instrument when password is entered.
AD Gross	Instantaneous value in engineering units of the AD counter converter (Gross).
AD Net	Instantaneous value in engineering units of the AD counter converter (Net Only).
Weight Load Cell	Instantaneous Value in mV of the load cell.
Zero Load Cell	Value in engineering units of the AD converter equivalent to the Zero Calibration.
Span Load Cell	Value in engineering units of the AD converter equivalent to the Span Calibration.
Gain	Gain of the ADC converter
Mechtare	Mechanical tare value.
Service Password	Numeric digit or a letter of the alphabet used by service people.The maximum length is 10 ascii characters:

VALID RANGE
a – z
A – Z
0 - 9

Tab 19- Service password limit

Operator Password Numeric digit or a letter of the alphabet used by operator people.
The maximum length is 10 ascii characters:

VALID RANGE
a – z
A – Z
0 - 9

Tab 20- Operator password limit

Software version Release of the Instrument firmware's.

4.4.9 Digital I/O

Digital inputs assign The digital inputs configuration.

N.O Edge [Code]	N.O Level [Code]	N.C. Edge [Code]	N.C. Level [Code]
0XH	20XH	10XH	30XH
"X": Stands for Number of Input			

Tab 21- Digital input assign numbers limits

Digital outputs assign The digital oputputs configuration.

N.O. [Code]	N.C. [Code]
0XH	10XH
"X": Stands for Number of Input	

Tab 22- Digital output assign numbers limits

4.4.10 Alarms

Threshold 1 opt. Enable the Threshold 1 alarm condition.

Threshold 2 opt. Enable the Threshold 2 alarm condition.

Threshold 3 opt. Enable the Threshold 3 alarm condition.

Threshold 4 opt. Enable the Threshold 4 alarm condition.

SETTINGs	CODE
NO	00H
YES	01H

Tab 25 – Thresholds option limits

Threshold 1 set Set (engineering units) value for Threshold 1 alarm condition.

Threshold 2 set	Set (engineering units) value for Threshold 2 alarm condition.
Threshold 3 set	Set (engineering units) value for Threshold 3 alarm condition.
Threshold 4 set	Set (engineering units) value for Threshold 4 alarm condition.

Threshold 1 delay	Delay time for Threshold 1 alarm condition.
Threshold 2 delay	Delay time for Threshold 2 alarm condition.
Threshold 3 delay	Delay time for Threshold 3 alarm condition.
Threshold 4 delay	Delay time for Threshold 4 alarm condition.

Threshold 1 hyst.	Set value for Threshold 1 hysteresis.
Threshold 2 hyst.	Set value for Threshold 2 hysteresis.
Threshold 3 hyst.	Set value for Threshold 3 hysteresis.
Threshold 4 hyst.	Set value for Threshold 4 hysteresis.

Threshold 1 mode	Set value for Threshold mode alarm condition
Threshold 2 mode	Set value for Threshold mode alarm condition
Threshold 3 mode	Set value for Threshold mode alarm condition
Threshold 4 mode	Set value for Threshold mode alarm condition

SETTINGs	CODE
LOW WEIGHT	00H
HIGH WEIGHT	01H

Tab 26 – Thresholds mode limits

Threshold 1 var.	Variable on which alarm 1 Threshold is managed.
Threshold 2 var.	Variable on which alarm 2 Threshold is managed.
Threshold 3 var.	Variable on which alarm 3 Threshold is managed.
Threshold 4 var.	Variable on which alarm 4 Threshold is managed.

SETTINGs	CODE
NET WEIGHT	00H
GROSS WEIGHT	01H
ABS NET WEIGHT	02H
ABS GROSS WEIGHT	03H

Tab 27 – Thresholds variable limits

Alarm defines The user can select the desired mode between:

ALARM (just a warning message),

SHUT DOWN (Warning message) and

NONE (no action).

Code of the setted “Alarm define mode” according the table below:

SETTINGs	CODE
NONE	00H
ALARM	01H
SHUTDOWN	02H

Tab 28 – Alarm define code

4.4.11 Ethernet Settings

I.P. Address Represent the value of the I.P address of the devic (default: 169.254.1.3).

SubNet Mask Represent the value of the Subnet Mask configured on the device (default : 255.255.255.000).

Gateway Represent the value of the Gateway configured on the device (default : 169.254.001.003).

Mc Address Represent the value of the Mac Address of the device (00:17:27:00:00:95)

Variable Selection Code of mode on how the variable are transmitted:

SETTINGs	WORD
NO DUMPED	00H
DUMPED	01H
DISPLAYED	02H

Tab 29– Variable selection range code

Swap Int Data Code of mode on how the INT DATA variable are swapped.

Swap Float Data Code of mode on how the FLOAT DATA variable are swapped.

Swap Long Integer Code of mode on how the LONG INTEGER DATA variable are swapped.

SETTINGs	WORD
NO SWAP	00H
SWAP BYTES	01H
SWAP WORDS	02H
SWAP BYTES+WORDS	02H

Tab 30– Variable swapped selection

4 QUERY, RESPONSE MESSAGES AND EXAMPLE

Below there are the structure of the data query from a Master (PLC) to the Salve (MINI CK 100)
Starting from the slave address field, the structure is that of the MODBUS message, except for error check field.

4.1 Read Operation Example (Function 03)

QUERY (CLIENT)

Transaction identifier	:	xx
Transaction identifier	:	xx
Protocol identifier	:	xx
Protocol identifier	:	xx
Length	:	xx
Length	:	xx
Unit identifier	:	xx
Function code	:	xx
Start address Hi	:	xx
Start address Lo	:	xx
Nr. of data Hi	:	xx
Nr. of data Lo	:	xx

RESPONSE MESSAGE (SERVER)

Transaction identifier	:	xx
Transaction identifier	:	xx
Protocol identifier	:	xx
Protocol identifier	:	xx
Length	:	xx
Length	:	xx
Unit identifier	:	xx
Function code	:	xx
Byte count	:	xx
Data Hi	:	xx
Data Lo	:	xx
Data Hi	:	xx
Data Lo	:	xx

EXAMPLE 1 – READ HOLDING REGISTER
The Client wants to receive the Belt load:

Address of slave	:	01
Register N°	:	59 (3B Hex)
Type of register	:	Float
Load value	:	100,0 Kg/m

QUERY (CLIENT)

Transaction identifier	:	0x00
Transaction identifier	:	0x00
Protocol identifier	:	0x00
Protocol identifier	:	0x00
Length	:	0x00
Length	:	0x06
Unit identifier	:	0x01
Function code	:	0x03
Start address	:	0x00
Start address	:	0x3b
Nr. of data to read	:	0x00
Nr. of data to read	:	0x02

RESPONSE MESSAGE (SERVER)

Transaction identifier	:	0x00
Transaction identifier	:	0x00
Protocol identifier	:	0x00
Protocol identifier	:	0x00
Length	:	0x00
Length	:	0x07
Unit identifier	:	0x01
Function code	:	0x03
Bytes count	:	0x04
Data Hi	:	0x00
Data Lo	:	0x00
Data Hi	:	0x42
Data lo	:	0xc8

EXAMPLE 2 – READ HOLDING REGISTER
The Client wants to receive the Scale Division Settings :

Address of slave	:	01
Register N°	:	111 (6F Hex)
Type of register	:	integer
Division	:	0,1

<u>QUERY</u> (CLIENT)

Transaction identifier	:	0x00
Transaction identifier	:	0x00
Protocol identifier	:	0x00
Protocol identifier	:	0x00
Length	:	0x00
Length	:	0x06
Unit identifier	:	0x01
Function code	:	0x03
Start address	:	0x00
Start address	:	0x6F
Nr. of data to read	:	0x00
Nr. of data to read	:	0x01

RESPONSE MESSAGE (SERVER)

Transaction identifier	:	0x00
Transaction identifier	:	0x00
Protocol identifier	:	0x00
Protocol identifier	:	0x00
Length	:	0x00
Length	:	0x05
Unit identifier	:	0x01
Function code	:	0x03
Bytes count	:	0x02
Data Hi	:	0x00
Data Lo	:	0x08

4.2 *Write Operation Example*

PRESET SINGLE REGISTER(Function 6)

QUERY (CLIENT)

Transaction identifier	:	xx
Transaction identifier	:	xx
Protocol identifier	:	xx
Protocol identifier	:	xx
Length	:	xx
Length	:	xx
Unit identifier	:	xx
Function code	:	xx
Register address Hi	:	xx
Register address Lo	:	xx
Preset data Hi	:	xx
Preset data Lo	:	xx

RESPONSE MESSAGE (SERVER)

Transaction identifier	:	xx
Transaction identifier	:	xx
Protocol identifier	:	xx
Protocol identifier	:	xx
Length	:	xx
Length	:	xx
Unit identifier	:	xx
Function code	:	xx
Register address Hi	:	xx
Register address Lo	:	xx
Preset data Hi	:	xx
Preset data Lo	:	xx

EXAMPLE 1
The Client wants to select (write) the language of the Instrument :

Address of slave	:	01
Register N°	:	100 (64 Hex)
Type of register	:	integer
Settings	:	French (03Hex)

QUERY (CLIENT)

Transaction identifier	:	0x00
Transaction identifier	:	0x00
Protocol identifier	:	0x00
Protocol identifier	:	0x00
Length	:	0x00
Length	:	0x06
Unit identifier	:	0x01
Function code	:	0x06
Register address Hi	:	0x00
Register address Lo	:	0x64
Preset data Hi	:	0x00
Preset data Lo	:	0x03

RESPONSE MESSAGE (SERVER)

Transaction identifier	:	0x00
Transaction identifier	:	0x00
Protocol identifier	:	0x00
Protocol identifier	:	0x00
Length	:	0x00
Length	:	0x04
Unit identifier	:	0x01
Function code	:	0x06
Register address Hi	:	0x00
Register address Lo	:	0x64
Preset data Hi	:	0x00
Preset data Lo	:	0x03

PRESET MULTIPLE REGISTERS (Function 16)

QUERY (CLIENT)

Transaction identifier	:	xx
Transaction identifier	:	xx
Protocol identifier	:	xx
Protocol identifier	:	xx
Length	:	xx
Length	:	xx
Unit identifier	:	xx
Function code	:	xx
Start address Hi	:	xx
Start address Lo	:	xx
Nr. of register Hi	:	xx
Nr. of register Lo	:	xx
Bytes count	:	xx
Data Hi	:	xx
Data Lo	:	xx
Data Hi	:	xx
Data Lo	:	xx

RESPONSE MESSAGE (SERVER)

Transaction identifier	:	xx
Transaction identifier	:	xx
Protocol identifier	:	xx
Protocol identifier	:	xx
Length	:	xx
Length	:	xx
Unit identifier	:	xx
Function code	:	xx
Start address Hi	:	xx
Start address Lo	:	xx
Nr. of register Hi	:	xx
Nr. of register Lo	:	xx

EXAMPLE 1

The Client wants to write the scale capacity of the Instrument :

Address of slave	:	01
Register N°	:	109 (6D Hex)
Type of register	:	float
Settings	:	100.0 t/h

QUERY (CLIENT)

Transaction identifier	:	0x00
Transaction identifier	:	0x00
Protocol identifier	:	0x00
Protocol identifier	:	0x00
Length	:	0x00
Length	:	0x09
Unit identifier	:	0x01
Function code	:	0x10
Start address Hi	:	0x00
Start address Lo	:	0x6d
Nr. of register Hi	:	0x00
Nr. of register Lo	:	0x02
Bytes count	:	0x04
Data Hi	:	0x00
Data Lo	:	0x00
Data Hi	:	0x42
Data Lo	:	0xc8

RESPONSE MESSAGE (SERVER)

Transaction identifier	:	0x00
Transaction identifier	:	0x00
Protocol identifier	:	0x00
Protocol identifier	:	0x00
Length	:	0x00
Length	:	0x05
Unit identifier	:	0x01
Function code	:	0x10
Start address Hi	:	0x00
Start address Lo	:	0x6d
Nr. of register Hi	:	0x00
Nr. of register Lo	:	0x02

APPENDIX A

FLOATING POINT NOTATION

MINI CK 101 stores floating point data types using the IEEE single precision format. The format contains a sign bit, an exponent field and a fraction field or mantissa.

- <u>The represented value</u>

  the value of the number being represented is equal to the exponent multiplied by the fractional part with the sign specified by the bit sign field:

  $(-1)^{sign} * (1.0+fraction) * s^{(exp-bias)}$

  For detailed information, refer to the *ANSI IEEE Standard for Binary Floating Point Arithmetic.*

- <u>Sign Bit</u>

  The sign of the number being represented is stored in the sign bit. If the number is positive, the sign bit contains the value 0. If it is negative, it contains the value 1. The sign bit is stored in the most significant bit of a floating point value.

- <u>Exponent Field</u>

  Using an exponent increases the range of representable numbers. The exponent field of the number contains a ‘biased’ form of the exponent. A bias is subtracted from the exponent field, letting the actual exponent represent both positive and negative exponents. The value of this bias is hexadecimal 7F therefore the effective exponent (Rexp) of the number can be obtained as:

  Rexp = Exp - 7F

  Note

  If both the exponent field and the fraction field are equal to zero, the number being represented is zero.

- <u>Fraction field (Mantissa)</u>

  IEEE floating point format stores the fractional part of a number in a “normalized” form. It assumes that all nonzero numbers are of the following form:

  1.xxxxxxxx (binary)

  The character x represent either 0 or 1 (binary).
  Since all floating point numbers being with 1, the 1 becomes the implicit normalized bit. It is the most significant bit of the fraction and is not stored in memory. The binary point is located immediately to the right of the normalized bit. All bits after the binary point represent values less the 1 (binary).

Example:

The number 1.625 (dec) can be represented as:

1.101 (bin)

This number can be represented in decimal as:

$2^0 + 2^{-1} + 2^{-3}$ (dec) Or 1 + 0.5 + 0.25

- <u>Examples</u>

some example has been added to help in the understanding of the rules explained above.

Ex 1.

The number 100.0 is represented as : 42 c8 00 00 (hex) or:

0   1 0 0 0 0 1 0 1   1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Bit 31 30 23 22 16 0

The sign is positive (bit 31 is 0).
The Exponent Field is : 1 0 0 0 0 1 0 1 (bin), 85 (hex). The *exp* is : *exp* = 85(hex) - bias (7F hex) = 6.
Thus the first implicit bit to 1 has weight 6
At this point the mantissa can be represented in this way, with the weight of each bit.

2^6 2^5 2^4 2^3 2^2 2^1 2^0 2^{-1} 2^{-2} 2^{-3} ....
1   1 0 0 1 0 0 0 0 0 0 0
Bi t Imp 22 0

Now it is enough to sum the weight of the bits at 1.

Value = 2^6(imp) + 2^5 + 2^2 = 100.0

Ex 2.

Value = 0.15, binary representation is:

```
      0    0  1  1  1  1  1  0  0    0  0  1  1  0  0  1  1  0  0  1  1  0  0  1  1  0  0  1  1  0  1  0
Bit   31   30                 23   22                                                                0
```

Sign = 0 positive
Exp = 7C hex - 7F hex = -3

Mantissa =

```
      2^-3   2^-4 2^-5 2^-6 2^-7 2^-8 2^-9 2^-10 ............
      1      0  0  1  1  0  0  1  1  0  0  1  1  0  0  1  1  0  0  1  1  0  1  0
Bit   Imp    22                                                                0
```

Value = $2^{-3} + 2^{-6} + 2^{-7} + 2^{-10} + 2^{-11}$ ...... = 0.15 (approx.)

Ex 3.

Value = -5.5, binary representation is:

```
      1    1  0  0  0  0  0  0  1    0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
Bit   31   30                 23   22                                                                0
```

Sign = 1 negative
Exp = 81hex - 7F hex = +2
Mantissa =

```
      2^2    2^1 2^0 2^-1 2^-2 2^-3 2^-4 2^-5 2^-6 2^-7 2^-8 2^-9 2^-10 ............
      1      0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
Bit   Imp    22                                                                0
```

Value = 2^2 (imp.) + 2^0 + 2^{-1} = 5.5 (negative)