SUPMETER

BST100-A/B/C MODBUS

Communication Protocol

V3.0

（RS232/RS485/CANBUS/Ethernet）

ООО «Промвесоборудование»
Отдел Пусконаладки

Changsha Supmeter Technological Co.,Ltd.

Contents

1. HOST-SLAVE MODE MODBUS ASCII PROTOCOL 3

1.1 HOST-SLAVE NETWORK 3
1.2 CODING SYSTEM 3
1.3 SLAVE WEIGHING CONTROLLER SETTING 3
1.4 HOST SETTING 3
1.5 MODBUS ASCII COMMUNICATION FORMATS 4
1.5.1 Command 0x03 for Reading Registers 4
Example1.5.1.1 Command 0x03 for Reading Totalized Weight, Speed, Flow and Flow Set Point 5
Example1.5.1.2 Command 0x03 for Reading DI1 State 6
Example1.5.1.3 Command 0x03 for Reading Working State 7
Example1.5.1.4 Command 0x03 for Reading Parameter P112 Cal. Revs (Method A) 8
Example1.5.1.5 Command 0x03 for Reading Displayed Weight During Calibrating 9
1.5.2 Command 0x06 for Writing Single Register 10
Example1.5.2.1 Command 0x06 for Turning DO1 ON 10
Example1.5.2.2 Command 0x06 for Clearing Screen 10
Example1.5.2.3 Command 0x06 for Key-locking 11
Example1.5.2.4 Command 0x06 for Writing Key Value 11
Example1.5.2.5 Command 0x06&0x03 for Reading Prarmeter P112 Cal. Revs (Method B) 12
1.5.3 Command 0x10 for Writing Multiple Registers 13
Example1.5.3.1 Command 0x10 for Writing Flow Set Point 14
Example1.5.3.2 Command 0x06 for Writing Date&Time 15
Example1.5.3.3 Command 0x10 for Writing Parameter P112 Cal. Revs (Method A) 16
Example1.5.3.4 Command 0x06&0x10 for Writing Parameter P112 Cal. Revs (Method B) 17
Example1.5.3.5 Command 0x10 for Writing Real Weight for Calibrarion 18
1.6 MODBUS RTU COMMUNICATION FORMAT 19
1.6.1 Command 0x03 for Reading Registers 19
Example 1.6.1.1 Command 0x03 for Reading Totalized Weight, Speed, Flow and Flow Set Point 20
1.6.2 Command 0x06 for Writing Single Register 21
Example1.6.2.1 Command 0x06 for Turning DO1 ON 21
1.6.3 Command 0x10 for Writing Multiple Registers 22
Example1.6.3.1 Command 0x10 for Writing Flow Set Point 23

2. CONTINUOUS TRANSMITTING MODE ASCII PROTOCOL 24

2.1 POINT-TO-POINT NETWORK 24

2.2 CODING SYSTEM 24
2.3 TRANSMITTER (WEIGHING CONTROLLER) SETTING 24
2.4 RECEIVER SETTING 24
2.5 CONTINUOUS TRANSMITTING FORMAT 25

APPENDIX 1. LRC GENERATION 26

REFERENCES: *JUNE 1996, MODICON MODBUS PROTOCOL REFERENCE GUIDE, MODICON, INC., INDUSTRIAL AUTOMATION SYSTEMS* 26

APPENDIX 2. ETHERNET CONFIGURATION FOR SLAVE 27

STEP4: SET THE NETWORK CONFIGURATION OF THE SLAVE (WEIGHING CONTROLLERS). 27

APPENDIX 3. REGISTER DESCRIPTION 28

A 3.1 REGISTERS FOR REAL-TIME DATA AND PARAMETER WRITING&READING 28

A3.1.1 Working State Register 29
A3.1.2 Bit Format of DI&DO Register 29
A3.1.3 Date&Time Register Format 29
A 3.2 REGISTERS FOR KEYPAD OPERATION 30
A 3.3 REGISTERS FOR REMOTE DISPLA 30
A3.3.1 Bit Format of Registers for Character Color per Line 31
A3.3.2 Bit Format of LED-Light Register 31

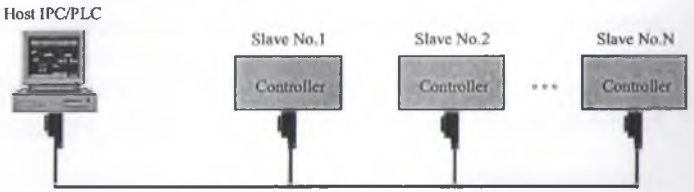
APPENDIX 4. SYSTEM CALIBRATION DESCRIPTION 32

A4.1 ZERO CALIBRATION 32
A4.2 DYNAMIC SPAN CALIBRATION 32
A4.3 STATIC SPAN CALIBRATION 32

APPENDIX 5. CRC GENERATION 33

1. Host-Slave Mode Modbus ASCII Protocol

1.1 Host-Slave Network



RS232: N=1; RS485: N≤31; Ethernet; CANBUS

1.2 Coding System

- ASCII code
  - Hexadecimal, ASCII characters 0–9(0x30-0x39), A–F(0x41-0x46).
  - Each 8–bit byte is sent as two ASCII characters.
- RTU
  - Send by Hexadecimal.
- Byte Format
  - 1 start bit.
  - 8 data bits (least significant bit sent first).
  - 1 bit for parity check (even/odd/none).
  - 1 stop bit.
- Error Check Field
  - ASCII adopt Longitudinal Redundancy Check (LRC). See Appendix 1;
  - RTU adopt Cyclic Redundancy Check (CRC). See Appendix 5.
- Baud Rate
  - 1200/2400/4800/9600/19200bps.

1.3 Slave Weighing Controller Setting

- Set Address to 00-99.
- Set Communication Mode to Host-Slave.
- Set Baud Rate to 1200/2400/4800/9600/19200bps.
- Set Parity Check to Even/Odd/None.
- For Ethernet, Set IP Address via the configuration software. See Appendix 2.

1.4 Host Setting

- Set Slave Addresses of Host Software in accordance with those of slave controllers.
- Set the Coding System parameters in accordance with those of slave weighing controllers.

1.5 Modbus ASCII Communication Formats

1.5.1 Command 0x03 for Reading Registers

QUERY FORMAT:

Field Name	ASCII [Hi-Lo]	
Start	1 Char	[0x3A]
Slave Address	2 Chars	
Command	2 Chars	[0x30 33]
Starting Register Address	4 Chars	[= Starting Register No. minus 1]
Quantity of Registers	4 Chars	
LRC Check	2 Chars	
End	2 Chars	[0x0D 0A]

RESPONSE FORMAT:

Field Name	ASCII [Hi-Lo]	
Start	1 Char	[0x3A]
Slave Address	2 Chars	
Command	2 Chars	[0x30 33]
Byte Count	2 Chars	
Starting Register Data	2 Chars	
...		
Last Register Data	2 Chars	
LRC Check	2 Chars	
End	2 Chars	[0x0D 0A]

Example1.5.1.1 Command 0x03 for Reading Totalized Weight, Speed, Flow and Flow Set Point

Read data from Registers No.0001-0008 (Quantity of Registers: 8; Byte Count: 16) of slave controller No.01. See Appendix 3.

Data Name	Type	Register No.	Value[DEC]	[HEX]	[ASCII]
Totalized Weight	Long	0001-0002	19491001kg	01 29 68 B9	30 31 32 39 36 38 42 39
Speed	Float	0003-0004	0.35m/s	3E B3 33 33	33 45 42 33 33 33 33 33
Flow	Float	0005-0006	19.9t/h	41 9F 33 33	34 31 39 46 33 33 33 33
Flow Set Point	Float	0007-0008	20.0t/h	41 A0 00 00	34 31 41 30 30 30 30 30

QUERY:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	30 33	
Starting Register Address	30 30 30 30	[Starting Register No.0001 minus 1=0x00 00]
Quantity of Registers	30 30 30 38	
LRC Check	- -	
End	0D 0A	

RESPONSE:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	30 33	
Byte Count	31 30	[DEC:16; HEX:10]
Starting Register Data	30 31 32 39	Totalized Weight [Hi]
	36 38 42 39	Totalized Weight [Lo]
	33 45 42 33	Speed [Hi]
	33 33 33 33	Speed [Lo]
	34 31 39 46	Flow [Hi]
	33 33 33 33	Flow [Lo]
	34 31 41 30	Flow Set Point [Hi]
Last Register Data	30 30 30 30	Flow Set Point [Lo]
LRC Check	- -	
End	0D 0A	

Example1.5.1.2 Command 0x03 for Reading DI1 State

Read data from Register No.0033 (Quantity of Registers: 1; Byte Count: 2) of slave controller No.01. See Appendix 3.

Data Name	Type	Register No.	Value[HEX]	[ASCII]
DI1 State	Int	0033	00 FF [00 00: OFF; 00 FF: ON]	30 30 46 46

QUERY:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	30 33	
Starting Register Address	30 30 32 30	[Starting Register No.0033 minus1=0x00 20]
Quantity of Registers	30 30 30 31	
LRC Check	- -	
End	0D 0A	

RESPONSE:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	30 33	
Byte Count	30 32	
Starting Register Data	30 30 46 46	[DI1 State: 0x00 FF]
LRC Check	- -	
End	0D 0A	

Example1.5.1.3 Command 0x03 for Reading Working State

Read data from Register No.0009 (Quantity of Registers: 1; Byte Count: 2) of slave controller No.01.
See Appendix 3.

Data Name	Type	Register No.	Value[HEX]	[ASCII]
Working State	Int	0009	00 FF [Weighing]	30 30 46 46

QUERY:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	30 33	
Starting Register Address	30 30 30 38	[Starting Register No.0009 minus 1=0x00 08]
Quantity of Registers	30 30 30 31	
LRC Check	- -	
End	0D 0A	

RESPONSE:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	30 33	
Byte Count	30 32	
Starting Register Data	30 30 46 46	[0x00 FF: Weighing]
LRC Check	- -	
End	0D 0A	

Example1.5.1.4 Command 0x03 for Reading Parameter P112 Cal. Revs (Method A)

Read data from Register No.0112 (Quantity of Registers: 2; Byte Count: 4) of slave controller No.01.
See Appendix 3.

Data Name	Type	Register No.	Value[DEC]	[HEX]	[ASCII]
Cal. Revs	Long	0112	10[Revs]	00 00 00 0A	30 30 30 30 30 30 30 41

QUERY:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	30 33	
Starting Register Address	30 30 36 46	[Starting Register No.0112 minus 1=0x00 6F]
Quantity of Registers	30 30 30 32	
LRC Check	- -	
End	0D 0A	

RESPONSE:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	30 33	
Byte Count	30 34	
Starting Register Data	30 30 30 30	P112 [Hi]
Last Register Data	30 30 30 41	P112[Lo]
LRC Check	- -	
End	0D 0A	

Example1.5.1.5 Command 0x03 for Reading Displayed Weight During Calibrating

Read data from Register No. 0051-0052 (Quantity of Registers: 2; Byte Count: 4) of slave controller No.01. See Appendix 3.

Data Name	Type	Register No.	Value[DEC]	[HEX]	[ASCII]
Displayed Weight During Calibrating	Long	0051-0052	1000[100.0kg]	00 00 03 E8	30 30 30 30 30 33 45 38

QUERY:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	30 33	
Starting Register Address	30 30 33 32	[Starting Register No.0051 minus 1=0x00 32]
Quantity of Registers	30 30 30 32	
LRC Check	- -	
End	0D 0A	

RESPONSE:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	30 33	
Byte Count	30 34	
Starting Register Data	30 30 30 30	Displayed Weight During Calibrating [Hi]
Last Register Data	30 33 45 38	Displayed Weight During Calibrating [Lo]
LRC Check	- -	
End	0D 0A	

1.5.2 Command 0x06 for Writing Single Register

QUERY&RESPONSE:

Field Name	ASCII [Hi-Lo]	
Start	1 Char	[0x3A]
Slave Address	2 Chars	
Command	2 Chars	[0x30 36]
Register Address	4 Chars	[= Register No. minus 1]
Register Data	4 Chars	
LRC Check	2 Chars	
End	2 Chars	[0x0D 0A]

Example1.5.2.1 Command 0x06 for Turning DO1 ON

Write 0x00FF into Register No.0036 of slave controller No.01.
See Appendix 3.

Data Name	Type	Register No.	Value[HEX]	[ASCII]
DO1 State&Control	Int	0036	00 FF [0x0000：OFF；0x00FF：ON]	30 30 46 46

QUERY&RESPONSE:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	30 36	
Register Address	30 30 32 33	[Register No.0036 minus 1=0x00 23]
Register Data	30 30 46 46	
LRC Check	- -	
End	0D 0A	

Example1.5.2.2 Command 0x06 for Clearing Screen

Write 0x00A5 into Register No.0009 of slave controller No.01.
See Appendix 3.

Operation Name	Type	Register No.	Value[HEX]	[ASCII]
Clear Screen	Int	0009	00 A5	00 00 41 35

QUERY&RESPONSE:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	30 36	
Register Address	30 30 30 38	[Register No.0009 minus 1=0x08]
Register Data	30 30 41 35	
LRC Check	- -	
End	0D 0A	

Example1.5.2.3 Command 0x06 for Key-locking

Write 0x0002 into Register No.0027 of slave controller No.01.
See Appendix 3.

Operation Name	Type	Register No.	Value[HEX]	[ASCII]
Key-locking	Int	0027	00 00	30 30 30 30
			[00 00: Locking; 00 02: Unlocking]	

QUERY&RESPONSE:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	30 36	
Register Address	30 30 31 41	[Register No.0027 minus 1=0x00 1A]
Register Data	30 30 30 30	
LRC Check	- -	
End	0D 0A	

Example1.5.2.4 Command 0x06 for Writing Key Value

Write Key Value into Register No.0059 of slave controller No.01.
See Appendix 3.

Operation Name	Type	Register No.	Value[HEX]	[ASCII]
【ENT】	Int	0059	00 40	30 30 34 30
【ZERO】	Int	0059	00 21	30 30 32 31
【CAL1】	Int	0059	00 22	30 30 32 32
【CAL2】	Int	0059	00 23	30 30 32 33

QUERY&RESPONSE:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	30 36	
Register Address	30 30 33 41	[Register No.0059 minus 1=0x00 3A]
Register Data	30 30 34 30	【ENT】
LRC Check	- -	
End	0D 0A	

Example1.5.2.5 Command 0x06&0x03 for Reading Prarmeter P112 Cal. Revs (Method B)

Read data from Register No. 0048-0049 (Quantity of Registers: 2; Byte Count: 4) of slave controller No.01. See Appendix 3.

Data Name	Type	Register No.	Value[DEC]	[HEX]	[ASCII]
Parameter No.	Int	0047	0112	00 70	30 30 37 30
Parameter Value	Long	0048-0049	10[Revs]	00 00 00 0A	30 30 30 30 30 30 30 41

□ Step1. Command 0x06 for Writing Parameter No.0112 into Register No.0047.

QUERY&RESPONSE:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	30 36	
Register Address	30 30 32 45	[Register No.0047 minus 1=0x00 2E]
Register Data	30 30 37 30	
LRC Check	- -	
End	0D 0A	

□ Step2. Command 0x03 for Reading Value of Parameter P112 from Register No.0048-0049.

QUERY:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	30 33	
Starting Register Address	30 30 32 46	[Starting Register No.0048 minus 1=0x00 2F]
Quantity of Registers	30 30 30 32	
LRC Check	- -	
End	0D 0A	

RESPONSE:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	30 33	
Byte Count	30 34	
Starting Register Data	30 30 30 30	P112 [Hi]
Last Register Data	30 30 30 41	P112[Lo]
LRC Check	- -	
End	0D 0A	

1.5.3 Command 0x10 for Writing Multiple Registers

QUERY FORMAT:

Field Name	ASCII [Hi-Lo]	
Start	1 Char	[0x3A]
Slave Address	2 Chars	
Command	2 Chars	[0x31 30]
Starting Register Address	4 Chars	[= Starting Register No. minus 1]
Quantity of Registers	4 Chars	
Byte Count	2 Chars	
Starting Register Data	4 Chars	
...		
Last Register Data	4 Chars	
LRC Check	2 Chars	
End	2 Chars	[0x0D 0A]

RESPONSE FORMAT:

Field Name	ASCII [Hi-Lo]	
Start	1 Char	[0x3A]
Slave Address	2 Chars	
Command	2 Chars	[0x31 30]
Starting Register Address	4 Chars	[= Starting Register No. minus 1]
Quantity of Registers	4 Chars	
LRC Check	2 Chars	
End	2 Chars	[0x0D 0A]

Example1.5.3.1 Command 0x10 for Writing Flow Set Point

Write 20.0t/h into Registers No.0007-0008 (Quantity of Registers: 2; Byte Count: 4) of slave controller No.01. See Appendix 3.

Data Name	Type	Register No.	Value[DEC]	[HEX]	[ASCII]
Flow Set Point	Float	0007–0008	20.0t/h	41 A0 00 00	34 31 41 30 30 30 30 30

QUERY:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	31 30	
Starting Register Address	30 30 30 36	[Starting Register No. 0007 minus 1=0x00 06]
Quantity of Registers	30 30 30 32	
Byte Count	30 34	
Starting Register Data	34 31 41 30	
Last Register Data	30 30 30 30	
LRC Check	- -	
End	0D 0A	

RESPONSE:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	31 30	
Starting Register Address	30 30 30 36	[Starting Register No. 0007 minus 1=0x00 06]
Quantity of Registers	30 30 30 32	
LRC Check	- -	
End	0D 0A	

Example1.5.3.2 Command 0x06 for Writing Date&Time

Write '09-05-17 22:36:45-7' into Registers No. 0040-0046 (Quantity of Registers: 7; Byte Count: 14) of slave controller No.1.
See Appendix 3.

Data Name	Type	Register No.	Value[HEX]
Date&Time	7-Int	0040-0046	0009 0005 0011 0016 0024 002D 0007
			[DEC: 09 05 17 22 36 45 07]

QUERY:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	31 30	
Starting Register Address	30 30 32 37	[Starting Register No. 0040 minus 1=0x0027]
Quantity of Registers	30 30 30 37	
Byte Count	30 30 30 45	
Starting Register Data	30 30 30 39	
	30 30 30 35	
	30 30 31 31	
	30 30 31 36	
	30 30 32 34	
	30 30 32 44	
Last Register Data	30 30 30 37	
LRC Check	- -	
End	0D 0A	

RESPONSE:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	31 30	
Starting Register Address	30 30 32 37	[Starting Register No. 0040 minus 1=0x0027]
Quantity of Registers	30 30 30 37	
LRC Check	- -	
End	0D 0A	

Example1.5.3.3 Command 0x10 for Writing Parameter P112 Cal. Revs (Method A)

Write 10 [Revs] into Register No.0112 of slave controller No. 01 (Quantity of Registers: 2; Byte Count: 4). See Appendix 3.

Data Name	Type	Register No.	Value [DEC]	[HEX]	[ASCII]
Cal. Revs	Long	0112	10[Revs]	00 00 00 0A	30 30 30 30 30 30 30 41

QUERY:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	31 30	
Starting Register Address	30 30 36 46	[Starting Register No. 0112 minus 1=0x00 6F]
Quantity of Registers	30 30 30 32	
Byte Count	30 34	
Starting Register Data	30 30 30 30	
Last Register Data	30 30 30 41	
LRC Check	- -	
End	0D 0A	

RESPONSE:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	31 30	
Starting Register Address	30 30 36 46	[=Starting Register No. 0112 minus 1=0x00 6F]
Quantity of Registers	30 30 30 32	
LRC Check	- -	
End	0D 0A	

Example1.5.3.4 Command 0x06&0x10 for Writing Parameter P112 Cal. Revs (Method B)

Write 10[Revs] into Registers No.0048-0049 of slave controller No. 01 (Quantity of Registers: 2; Byte Count: 4). See Appendix 3.

Data Name	Type	Register No.	Value [DEC]	[HEX]	[ASCII]
Parameter No.	Int	0047	0112	00 70	30 30 37 30
Parameter Value	Long	0048-0049	10 [Revs]	00 00 00 0A	30 30 30 30 30 30 30 41

□ Step1. Command 0x06 for Writing Parameter No.0112 into Register No.0047.

QUERY&RESPONSE:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	30 36	
Register Address	30 30 32 45	[Register No.0047 minus 1=0x00 2E]
Register Data	30 30 37 30	
LRC Check	- -	
End	0D 0A	

□ Step2. Command 0x10 for Writing 10[Revs] into Registers No. 0048-0049.

QUERY :

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	31 30	
Starting Register Address	30 30 32 46	[Starting Register No.0048 minus 1=0x00 2F]
Quantity of Registers	30 30 30 32	
Byte Count	30 34	
Starting Register Data	30 30 30 30	
Last Register Dat	30 30 30 41	
LRC Check	- -	
End	0D 0A	

RESPONSE:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	31 30	
Starting Register Address	30 30 32 46	[Starting Register No.0048 minus 1=0x00 2F]
Quantity of Registers	30 30 30 32	
LRC Check	- -	
End	0D 0A	

Example1.5.3.5 Command 0x10 for Writing Real Weight for Calibrarion

Write 1000[100.0kg] into Registers No. 0051-0052 of slave controller No. 01 (Quantity of Registers: 2; Byte Count: 4). See Appendix 3.

Data Name	Type	Register No.	Value [DEC]	[HEX]	[ASCII]
Real Weight for Calibrarion	Long	0051-0052	1000[100.0kg]	00 00 03 E8	30 30 30 30 30 33 45 38

QUERY :

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	31 30	
Starting Register Address	30 30 33 32	[Starting Register No.0051 minus 1=0x00 32]
Quantity of Registers	30 30 30 32	
Byte Count	30 34	
Starting Register Data	30 30 30 30	
Last Register Dat	30 33 45 38	
LRC Check	- -	
End	0D 0A	

RESPONSE:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	31 30	
Starting Register Address	30 30 33 32	[Starting Register No.0051 minus 1=0x00 32]
Quantity of Registers	30 30 30 32	
LRC Check	- -	
End	0D 0A	

1.6 Modbus RTU Communication Format

1.6.1 Command 0x03 for Reading Registers

QUERY FORMAT:

Field Name	
Slave Address	1 Char
Command	1 Char [0x03]
Starting Register Address	2 Char s[= Starting Register No. minus 1]
Quantity of Registers	2 Chars
CRC Check	2 Chars

RESPONSE FORMAT:

Field Name	
Slave Address	1 Char
Command	1 Char [0x03]
Byte Count	1 Char
Starting Register Data	2 Chars
...	
Last Register Data	2 Chars
CRC Check	2 Chars

Example 1.6.1.1 Command 0x03 for Reading Totalized Weight, Speed, Flow and Flow Set Point

Read data from Registers No.0001-0008 (Quantity of Registers: 4; Byte Count: 8) of slave controller No.01. See Appendix 3.

Data Name	Type	Register No.	Value[DEC]	[HEX]
Totalized Weight	Long	0001–0002	19491001kg	01 29 68 B9
Speed	Float	0003–0004	0.35m/s	3E B3 33 33
Flow	Float	0005–0006	19.9t/h	41 9F 33 33
Flow Set Point	Float	0007–0008	20.0t/h	41 A0 00 00

QUERY:

Field Name		
Slave Address	01	
Command	03	
Starting Register Address	00 00	[Starting Register No.0001 minus 1=0x00 00]
Quantity of Registers	00 08	
CRC Check	..	

RESPONSE:

Field Name		
Slave Address	01	
Command	03	
Byte Count	10	[DEC:16; HEX:10]
Starting Register Data	01 29	Totalized Weight [Hi]
	68 B9	Totalized Weight [Lo]
	3E B3	Speed [Hi]
	33 33	Speed [Lo]
	41 96	Flow [Hi]
	33 33	Flow [Lo]
	41 40	Flow Set Point [Hi]
Last Register Data	00 00	Flow Set Point [Lo]
CRC Check	..	

1.6.2 Command 0x06 for Writing Single Register

QUERY&RESPONSE:

Field Name	
Slave Address	1 Char
Command	1 Char [0x06]
Register Address	2 Chars [= Register No. minus 1]
Register Data	2 Char s
CRC Check	2 Char s

Example1.6.2.1 Command 0x06 for Turning DO1 ON

Write 0x00FF into Register No.0036 of slave controller No.01.
See Appendix 3.

Data Name	Type	Register No.	Value[HEX] [ASCII]
DO1 State&Control	Int	0036	00 FF
			[00 00: OFF; 00 FF: ON]

QUERY&RESPONSE:

Field Name	
Slave Address	01
Command	06
Register Address	00 [Register No.0036 minus 1=0x00 23]
Register Data	30 30 46 46
CRC Check	- -
End	0D 0A

1.6.3 Command 0x10 for Writing Multiple Registers

QUERY FORMAT:

Field Name	
Slave Address	1 Char
Command	1 Char [0x10]
Starting Register Address	2 Chars [= Starting Register No. minus 1]
Quantity of Registers	2 Chars
Starting Register Data	2 Chars
...	
Last Register Data	2 Chars
CRC Check	2 Chars

RESPONSE FORMAT:

Field Name	
Slave Address	1 Char
Command	1 Char [0x10]
Starting Register Address	2 Chars [= Starting Register No. minus 1]
Quantity of Registers	2 Chars
CRC Check	2 Chars

Example1.6.3.1 Command 0x10 for Writing Flow Set Point

Write 20.0t/h into Registers No.0007-0008 (Quantity of Registers: 2; Byte Count: 4) of slave controller No.01. See Appendix 3.

Data Name	Type	Register No.	Value[DEC]	[HEX]
Flow Set Point	Float	0007–0008	20.0t/h	41 A0 00 00

QUERY:

Field Name		
Slave Address	01	
Command	10	
Starting Register Address	00 06	[Starting Register No. 0007 minus 1=0x00 06]
Quantity of Registers	00 02	
Byte Count	00 04	
Starting Register Data	41 A0	
Last Register Data	00 00	
CRC Check	--	

RESPONSE:

Field Name		
Slave Address	01	
Command	10	
Starting Register Address	00 06	[Starting Register No. 0007 minus 1=0x00 06]
Quantity of Registers	00 02	
CRC Check	--	

2. Continuous Transmitting Mode ASCII Protocol

2.1 Point-to-Point Network



2.2 Coding System

- ASCII Code
  - Hexadecimal, ASCII characters 0–9 (0x30-0x39), A–F (0x41-0x46).
  - Each 8–bit byte is sent as two ASCII characters.
- Byte Format
  - 1 start bit.
  - 8 data bits (least significant bit sent first).
  - 1 bit for parity check (even/odd/none).
  - 1 stop bit.
- Error Check Field
  - Longitudinal Redundancy Check (LRC). See Appendix 1.
- Baud Rate
  - 1200, 2400, 4800, 9600, 19200bps.

2.3 Transmitter (Weighing Controller) Setting

- Set Connection Mode to Continuous.
- Set Baud Rate to 1200/2400/4800/9600/19200bps.
- Set Parity Check to Even/Odd/None.
- For Ethernet, set IP Address via the configuration software. See Appendix 2.

2.4 Receiver Setting

- Set the 'Coding System' Parameters in accordance with those of Weighing Controller.

2.5 Continuous Transmitting Format

Transmitting Frequency: 2Hz.
The controller transmits the following data:

Data Name	Type	Register No.	Value[DEC]	[HEX]	[ASCII]
Totalized Weight	Long	0001–0002	19491001kg	01 29 68 B9	30 31 32 39 36 38 42 39
Speed	Float	0003–0004	0.35m/s	3E B3 33 33	33 45 42 33 33 33 33 33
Flow	Float	0005–0006	19.9t/h	41 9F 33 33	34 31 39 46 33 33 33 33

DATA FORMAT:

Field Name	ASCII [Hi-Lo]	
Start	3A	
Slave Address	30 31	
Command	30 33	
Byte Count	31 30	[DEC:16; HEX:10]
Starting Register Data	30 31 32 39	Totalized Weight [Hi]
	36 38 42 39	Totalized Weight [Lo]
	33 45 42 33	Speed [Hi]
	33 33 33 33	Speed [Lo]
	34 31 39 46	Flow [Hi]
Last Register Data	33 33 33 33	Flow [Lo]
LRC Check	- -	
End	0D 0A	

Appendix 1. LRC Generation

The Longitudinal Redundancy Check (LRC) field is one byte long, containing an 8–bit binary value. The LRC value is calculated by the transmitting device, which appends the LRC to the message. The receiving device recalculates an LRC on receipt of the message, and compares the calculated value with the actual value it received in the LRC field. If the two values are not equal, it indicates an error.

□ The Procedure for Generating an LRC

Step1. Add all bytes in the message, excluding the starting colon(0x3A)and the ending CRLF(0x0D 0A). Add them to an 8–bit field, so that carries will be discarded automatically.
Step2. Subtract the final field value from 0xFF.
Step3. Add 1 to produce the LRC value.
Step4. Place the LRC into the message.

For example, if the LRC value is 0x61:

Start	Address	Command	Data	...	Data	LRC Hi	LRC Lo	End
0x3A						0x36	0x31	0x0D 0A

□ LRC Generation

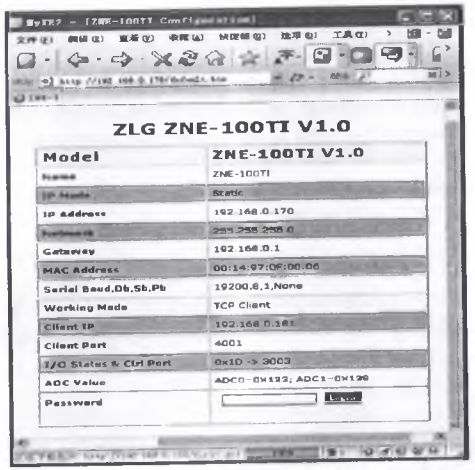
```
static unsigned char LRC(auchMsg, usDataLen)
unsigned char *auchMsg ;          /* message to calculate LRC */
unsigned char usDataLen ;         /* quantity of bytes in message */
{
    unsigned char uchLRC = 0 ;    /* LRC char initialized */
    while (usDataLen--)           /* pass through message buffer */
    uchLRC += *auchMsg++ ;        /* add buffer byte without carry */
    return((~uchLRC)++);
}
```

References: *June 1996, Modicon Modbus Protocol Reference Guide, MODICON, Inc., Industrial Automation Systems.*

Appendix 2. Ethernet Configuration for Slave

Step1: Open Internet Explorer.

Step2: Input the default IP address of the slave (weighing controller): http://192.168.0.178.



Step3: Input password 88888, and click Login button to enter Configuration menu.

Configuration
- Change Password
- IP Configuration
- Serial Configuration
- Working Option
- I/O Configuration
- Exit
- RESET ZNE-100TI

Step4: Set the network configuration of the slave (weighing controllers).

Appendix 3. Register Description

A 3.1 Registers for Real-Time Data and Parameter Writing&Reading

Data&Operation Name	Type	Register No. [DEC]	Attribute	Note
Totalized Weight	Long	0001-0002	R	kg
Speed	Float	0003-0004	R	m/s
Flow	Float	0005-0006	R	t/h
Flow Set Point	Float	0007-0008	R/W	t/h
Clear Screen	Int	0009	W	0x00 A5
Working State	Int	0009	R	See A3.1.1
Totalized Weight of Current Shift	Long	0010-0011	R	kg
Totalized Weight of Current Day	Long	0012-0013	R	kg
Totalized Weight of Current Month	Long	0014-0015	R	kg
Batch Weight Set Point	Long	0016-0017	R/W	kg
Current Batch Weight	Long	0018-0019	R	kg
Load	Long	0020-0021	R	kg/m
Load Set Point	Long	0022-0023	R/W	kg/m
Belt Angle	Int	0024	R	度（°）
Control Current	Int	0025	R/W	mA
AD Value	Int	0026	R	
Key-locking&Key-unlocking State&Operation	Int	0027	R/W	0x00 02: Key-unlocking 0x00 00: Key-locking
DI &DO State&Control	Int	0028	R/W	See A3.1.2
AI1 Value	Int	0029	R	mA
AO1 Value	Int	0030	R/W	mA
AO2 Value	Int	0031	R/W	mA
AO3 Value	Int	0032	R/W	mA
DI1 State	Int	0033	R	0x00 FF：ON 0x00 00：OFF
DI2 State	Int	0034	R	
DI3 State	Int	0035	R	
DO1 State&Control	Int	0036	R/W	
DO2 State&Control	Int	0037	R/W	
DO3 State&Control	Int	0038	R/W	
DO4 State&Control	Int	0039	R/W	
Date&Time	7-Int	0040-0046	R/W	参见 A3.1.3
Parameter No. (Method B for Parameter Reading&Writing)	Int	0047	R/W	See 6.3.1Parameter List of the Weighing Controller Manual
Parameter Value (Method B for Parameter Reading&Writing)	Long	0048-0049	R/W	
Unused	Int	0050	R/W	
Displayed Weight During Calibrating	Long	0051-0052	R	Display Value of Totolized Weight during Span Calibration and Segmenting Correction
Real Weight for Calibration	Long	0051-0052	W	Real weight (unit: kg) to be inputted after the last cycle of Dynamic Span Calibration or Segmenting Correction ends .
Software Version	Long	0053-0054	R	
Unused	Int	0055-0058	R/W	
Parameter (Method A for Parameter Reading&Writing)	Long	100-999	R/W	See 6.3.1 Parameter List of the Weighing Controller Manual.

Note: R－Readable; W－Writable.

A3.1.1 Working State Register

Register Value [HEX]	Working State
0x00 FF	Weighing
0x00 01	Dynamic Span Calibration
0x00 02	Static Span Calibration
0x00 03	Segmenting Correction
0x00 04	Zero Calibration
0x00 05	Speed Calibration
0x00 06	Belt Length Calibration
0x00 07-0x00 0F	Unused
0x00 10	Running for the last cycle of Dynamic Span Calibration or Segmenting Correction
0x00 11	Waiting for inputting Real Weight for Calibration

A3.1.2 Bit Format of DI&DO Register

BIT15-8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
		DI3	DI2	DI1	DO4	DO3	DO2	DO1
		R	R	R	R/W	R/W	R/W	R/W

BIT=0: OFF; BIT=1: ON

A3.1.3 Date&Time Register Format

Int 0	Int 1	Int 2	Int 3	Int 4	Int 5	Int 6
YEAR	MONTH	DATE	HOUR	MINUTE	SECOND	WEEKDAY
00-99	01-12	01-31	00-23	00-59	00-59	1-7 7: Sunday

A 3.2 Registers for Keypad Operation

Operation Name	Type	Register No. [DEC]	Attribute	Key Value [HEX]
【MENU】	Int	0059	W	0x00 51
【◄】				0x00 52
【►】				0x00 49
【▲】				0x00 48
【▼】				0x00 41
【ENT】				0x00 40
【ZERO】Zero Calibration				0x00 21
【CAL1】Dynamic Span Calibration				0x00 22
【CAL2】Static Span Calibration				0x00 23
【SEG】Segmenting Correction				0x00 24
【SPD】Speed Calibration				0x00 25
【LEN】Belt Length Calibration				0x00 26
【TIME】Date&Time Setting				0x00 27
【CLS】Clear Screen				0x00 28
【REC】Weight Record				0x00 29
【PRINT】Print				0x00 20
【1】				0x00 31
【2】				0x00 32
【3】				0x00 33
【4】				0x00 34
【5】				0x00 35
【6】				0x00 36
【7】				0x00 37
【8】				0x00 38
【9】				0x00 39
【0】				0x00 30

Note: R—Readable; W—Writable. Write key Value into Register No.0059 via Command 0x06.

A 3.3 Registers for Remote Displa

Data Name	Type	Register No. [DEC]	Attribute	Note
1st Line's Characters on LCD	8-Int	0060-0067	R	16 ASCII Chars
2nd Line's Characters on LCD	8-Int	0068-00075	R	16 ASCII Chars
3rd Line's Characters on LCD	8-Int	00076-0083	R	16 ASCII Chars
4th Line's Characters on LCD	8-Int	0084-0091	R	16 ASCII Chars
1st Line's Character Color	Int	0092	R	See A3.3.1
2nd Line's Character Color	Int	0093	R	
3rd Line's Character Color	Int	0094	R	
4th Line's Character Color	Int	0095	R	
LED-Light (on Controller's Panel)	Int	0096	R	See A3.3.2

Note: R—Readable; W—Writable. Read data from Registers via Command 0x10.

A3.3.1 Bit Format of Registers for Character Color per Line

Character No.	Bit No.	Note
Left.16	Bit15	
Left.15	Bit14	
Left.14	Bit13	
Left.13	Bit12	
Left.12	Bit11	
Left.11	Bit10	
Left.10	Bit9	
Left.9	Bit8	Bit=0: Black
Left.8	Bit7	Bit=1: Background Color
Left.7	Bit6	
Left.6	Bit5	
Left.5	Bit4	
Left.4	Bit3	
Left.3	Bit2	
Left.2	Bit1	
Left.1	Bit0	

A3.3.2 Bit Format of LED-Light Register

LED Light	Bit No.	Note
	Bit15	
	Bit14	
	Bit13	
	Bit12	
	Bit11	
	Bit10	
[PV]	Bit9	
[SV]	Bit8	Bit=0: OFF
[kg/m]	Bit7	Bit=1: ON
[t/h]	Bit6	
[t]	Bit5	
[kg]	Bit4	
[ALARM]	Bit3	
[EXT]	Bit2	
[AUTO]	Bit1	
[RUN]	Bit0	

Appendix 4. System Calibration Description

A4.1 Zero Calibration

Step No.	Description	Example	Note
1	Writing Parameter P112 Cal. Revs	1.5.3.3	or 1.5.3.4
2	Writing Key Value of【ZERO】	1.5.2.4	For starting Zero Calibration
3	Reading Working State	1.5.1.3	0x00 04: During Zero Calibration 0x00 FF: End (Correct Zero Value automatically)

A4.2 Dynamic Span Calibration

Step No.	Description	Example	Note
1	Writing Key Value of【CAL1】	1.5.2.4	For starting Dynamic Span Calibration
2	Reading Displayed Weight During Calibrating	1.5.1.5	
3	Writing Key Value of【ENT】	1.5.2.4	After loading ends
4	Reading Working State	1.5.1.3	0x00 10: Running for the last cycle 0x00 11: Waiting for inputting Real Weight
5	Writing Real Weight for Calibration	1.5.3.5	For correcting Span Coefficient

A4.3 Static Span Calibration

Step No.	Description	Example	Note
1	Writing Parameter P112 Cal. Revs	1.5.3.3	or Example1.5.3.4
2	Writing Key Value of【CAL2】	1.5.2.4	For starting Static Span Calibration
3	Reading Displayed Weight During Calibrating	1.5.1.5	
4	Reading Working State	1.5.1.3	0x00 02: During Static Span Calibration 0x00 FF: End (Correct Span Coefficient automatically)

Appendix 5. CRC Generation

The Cyclic Redundancy Check (CRC) field is one byte long, containing an 8–bit binary value. The CRC value is calculated by the transmitting device, which appends the CRC to the message. The receiving device recalculates an CRC on receipt of the message, and compares the calculated value with the actual value it received in the CRC field. If the two values are not equal, it indicates an error.

Address	Command	Data	...	Data	CRC[Hi]	CRC[Lo]
					0x36	0x31

Establish CRC Check Reminder Table firstly:

```
// CRC , High-bit byte Surplus
unsigned   char code CRCHI[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40
} ;
//CRC , Low-bit byte Surplus
unsigned   char code CRCLO[] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06,
0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04, 0xCC, 0x0C, 0x0D, 0xCD,
0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A,
0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC, 0x14, 0xD4,
0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3,
0xF2, 0x32, 0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4,
0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29,
0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED,
0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60,
0x61, 0xA1, 0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67,
0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
```



```
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68,
0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA, 0xBE, 0x7E,
0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71,
0x70, 0xB0, 0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92,
0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B,
0x99, 0x59, 0x58, 0x98, 0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B,
0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42,
0x43, 0x83, 0x41, 0x81, 0x80, 0x40
} ;
```

- **CRC Generation**

```
// CRC High-bit Byte Initial Value
HData_tmp=0xFF;
// CRC Low-bit Byte Initial Value
LData_tmp=0xFF;
// nend     Check byte number
//Lrc_Data[0]     CRC Check High-Bit
//Lrc_Data[1]     CRC Check Low-Bit
//pSendBuf[]      The data going to be sent
for(i=0;i<nend;i++)
{
    RetData_Tmp       = HData_tmp ^ pSendBuf[i];
    HData_tmp     = LData_tmp ^ CRCHI[RetData_Tmp] ;
    LData_tmp     = CRCLO[RetData_Tmp] ;
}
Lrc_Data[0]=HData_tmp;
Lrc_Data[1]=LData_tmp;
```

SUPMETER

Changsha Supmeter Technological Co.,Ltd.

Address: Building A6, Lugu International Industrial Park,
Changsha. 410205, China
Tel: +86 731 85115100
Fax: +86 731 85158100
Website: www.supmeter.com
E-mail: supmeter@supmeter.com