

Protocols

TW/VW

Protocol index

1	TPV0 (\$ Protocol).....	2
1.1	TYPE A (TPV0_A or 7Pu_A).....	2
1.1.1	Command Mode (7rAn0).....	2
1.1.2	Continuous Mode (7rAn1).....	3
1.1.3	Manual Mode (7rAn2)	3
1.2	TYPE B: (TPV0_B or 7Pu_B).....	4
1.2.1	Command Mode.....	4
1.2.2	Continuous Mode.....	6
1.2.3	Manual Mode	7
2	TPV2	8
2.1	TPV 2A (Uniwell-ICL Protocol).....	8
2.2	TPV 2B (Uniwell-W Protocol):	10
3	TPV 4 (Berkel Protocol)	11
4	TPV 7 (Samsung Protocol)	12
5	TEDC1 (CAS Protocol)	15
6	TOLEDO Protocol	17
7	NCIECR AND NCIGEN Protocol	18
8	TEC Protocol	19
9	EASY Protocol.....	20
10	TISA.....	22
10.1	TISA cash register protocol(tisa) A.....	22
10.2	Protocol TISA with sending of stable weight(tisa_S).....	23
10.3	Protocol VD TISA(vd_tisa)	24

Protocols Information

This modified version has several protocols used to connect to different POS (Point of Sale) systems. Details for the protocols can be found below:

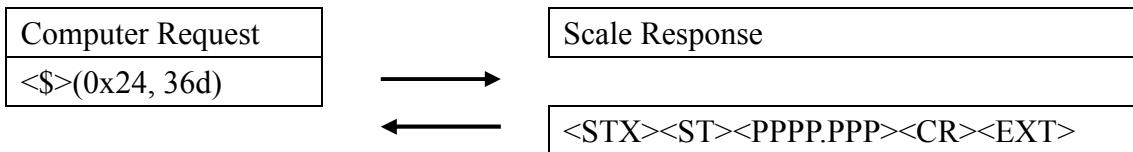
1 TPV0 (\$ Protocol)

From the menu, select the **TPV0_A** or **TPV0_B** (on the scale it is displayed as **7Pu_A** or **7Pu_B**) protocol depending on the type needed. Then select the transmission mode in the F8 function: 7rAn0 (command mode), 7rAn1 (continuous mode) or 7rAn2 (manual mode).

1.1 TYPE A (TPV0_A or 7Pu_A)

1.1.1 Command Mode (7rAn0)

Communication format:



Scale Response:

<STX>	Character 0x02, 2d
<ST>	Status byte whose value is obtained by adding the weight flags' status (0x08 if zero weight, 0x20 if stable weight, 0x02 if net weight and 0x01 if gross weight) to 0x20.
<PPPP.PPP>	Value of the weight in ASCII and with 8 bytes, including the weight's decimal point and the '-' sign when the net weight is negative. The weight field is left-justified with blank spaces (0x20, 32d). When the weight is out of range, dashes are sent (character 0x2d, 45d) occupying all 8 bytes of the weight field.
<CR>	Character 0x0d, 13d.
<ETX>	Character 0x03, 3d.

Values of <ST>

zero 8d	stable 32d	net 2d	gross 1d	0x20 32d	total of adding all flags	Char
8	32	0	1	32	73	I
8	0	0	1	32	41)
8	32	2	0	32	74	J
8	0	2	0	32	42	*
0	32	0	1	32	65	A
0	0	0	1	32	33	!
0	32	2	0	32	66	B
0	0	2	0	32	34	"

1.1.2 Continuous Mode (7rAn1)

The scale automatically sends the information described in the section Command Mode.

1.1.3 Manual Mode (7rAn2)

If the weight is stable, the scale can send the information described in the section Command Mode by pressing the M+ key. If the weight does not stabilize within a few seconds, the weight is not sent and the scale will display an unstable weight error (Error 14).

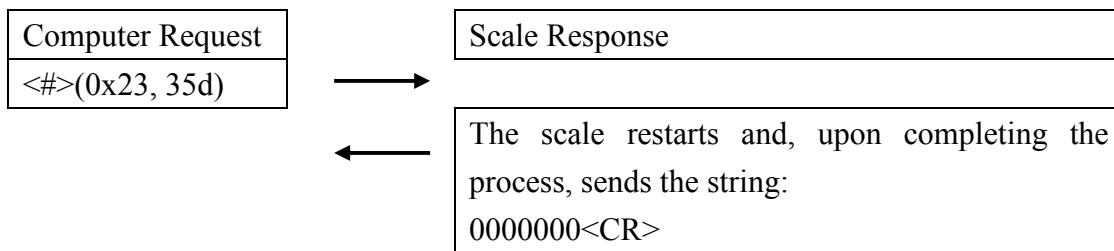
1.2 TYPE B: (TPV0_B or 7Pu_B)

1.2.1 Command Mode

The scale can perform a series of functions if a certain request is received:

- a) Resetting the scale::

Communication format:

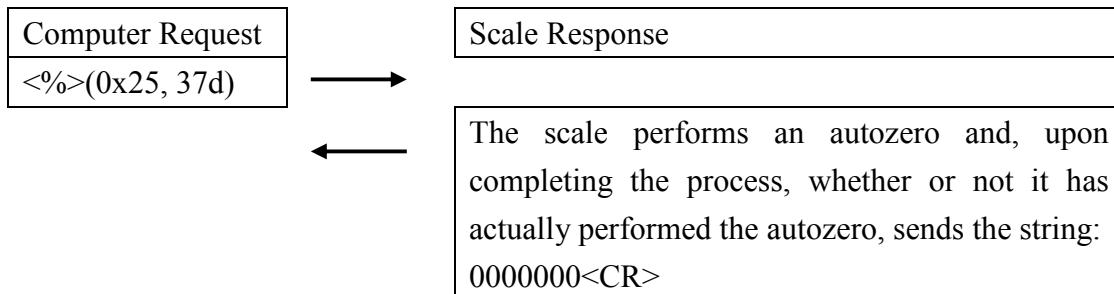


Scale response:

0000000	0x30, 48d
<CR>	Character 0x0d, 13d

- b) Autozero request:

Communication format:

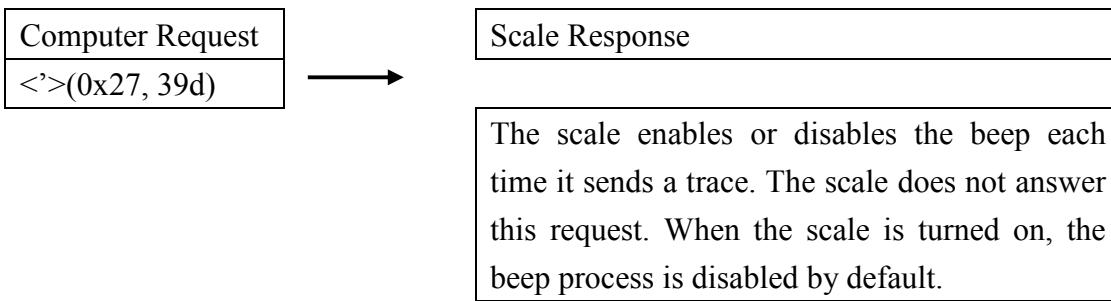


Scale response:

0000000	0x30, 48d
<CR>	Character 0x0d, 13d

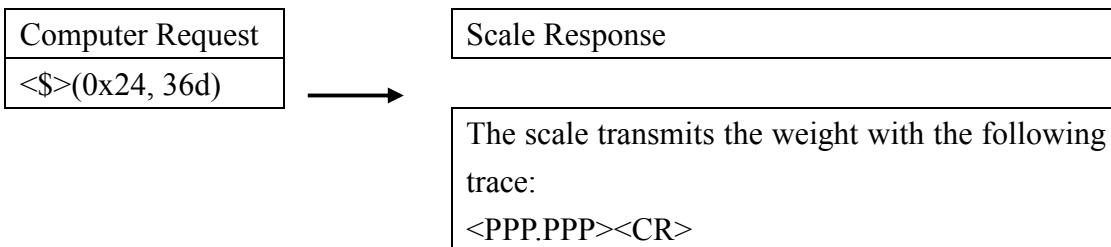
c) Disabling / enabling the transmission beep:

Communication format:



d) Weight request:

Communication format:



Scale Answer:

<PPP.PPP>	Value of the net weight with 7 bytes including the decimal point and left-justified with zeros (0x30)
<CR>	Character 0x0d, 13d

Special conditions:

- 1) Net or gross weight zero and stable: Transmits 0000000<CR>
- 2) Net or gross weight over or under: Transmits AAAAAAAA<CR>
- 3) Net or gross weight in range and stable: Sends PPP.PPP<CR> (the value of the weight with 7 bytes including the decimal point and left-justified with zeros).
- 4) Net or gross weight in range, stable and negative: Sends the string described in item 2.
- 5) Net or gross weight in range and unstable: the scale retains the weight request indefinitely until it becomes stable, when the scale operates as described in item 3.

Transmission started by the scale at the beginning

- 6) When the scale is turned on and the initial sequence is completed, it indicates that it is ready by sending the sequence 0000000<CR>.

1.2.2 Continuous Mode

After selecting this option, the continuous transmission protocol type is chosen. The possible values are: command mode or manual mode. The scale automatically sends the weight depending on the trace selected:

Command Mode trace: <PPP.PPP><CR>

<PPP.PPP>	Value of the weight with 7 bytes including the decimal point and left-justified with zeros (0x30). If the weight is out of range (over or under), or there is a negative net weight, a string composed of 7 characters A (0x41, 65d) is sent.
<CR>	Character 0x0d, 13d.

Manual Mode trace: <STX><ST><PPP.PPP><CR>

<STX>	Character 0x02, 2d.
<ST>	Tare status. The possible values are: blank (0x20, 32d) if the weight is gross, T (0x54, 84d) if there is a tare, and F (0x46, 70d) if there is a fixed tare.
<PPP.PPP>	Value of the weight with 7 bytes including the decimal point and left-justified with zeros (0x30). If the weight is Over, Under or Negative, <AAAAAAA> is sent.
<CR>	Character 0x0d, 13d.

1.2.3 Manual Mode

The scale will transmit the weight with the corresponding trace if the weight is stable by pressing the M+ key. If the weight does not stabilize in a present time (TIME LIMIT 3s), the weight is not sent and the scale displays an unstable weight error (Error 14).

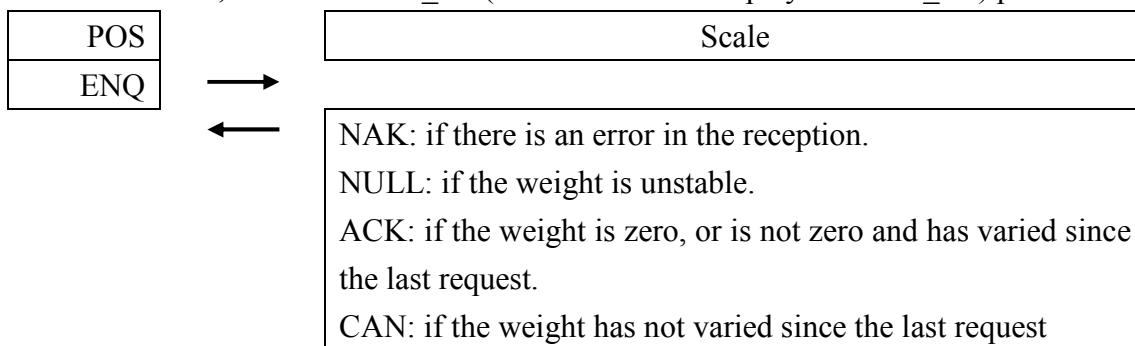
The trace used in this format is that described in the Continuous Mode with Manual Mode trace.

After sending the weight, the scale waits for an answer before the end of the duration (TIME OUT about 7s). If no answer is received at the end of this duration, the scale displays (Error 9). If the character ACK is received before the duration has elapsed (TIME OUT), the scale beeps, displays the indicator TXD OK for 2 seconds and, finally, beeps again, returning to the weight mode. If a negative answer is received, NAK, the scale displays (Error 10).

2 TPV2

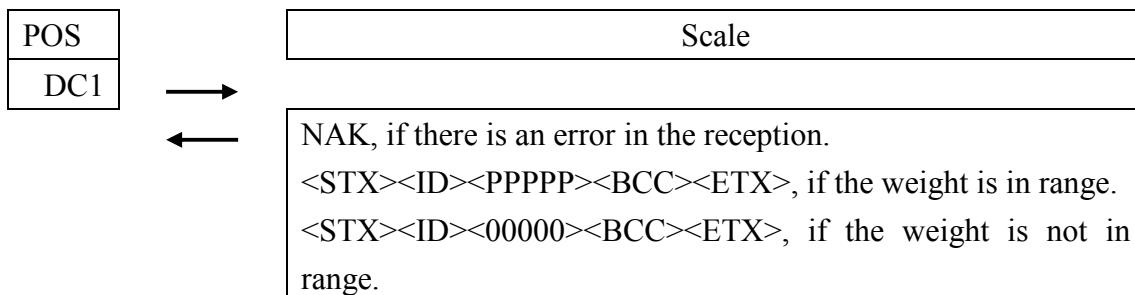
2.1 TPV 2A (*Uniwell-ICL Protocol*)

From the menu, select the TPV_2A (on the scale it is displayed as 7Pu_2A) protocol.



ENQ	- Character 0x05, 5d.
NACK	- Character 0x15, 21d.
NULL	- Character 0x00, 0d.
ACK	- Character 0x06, 6d.
CAN	- Character 0x18, 24d.

When the scale sends ACK to the POS:



DC1	- Character 0x11, 17d
NACK	- Character 0x15, 21d
STX	- Character 0x02, 2d
ID (weight in range)	- Scale type depending on the full-scale value. The scale always sends (0x69)
ID (weight not in range)	- Scale type depending on the OR full0scale value with 0x10. The scale always sends (0x79)
PPPPP	- Value of the weight in ASCII without decimal point and left-justified with zeros (0x30, 48d)
00000	- Five zeros in ASCII (0x30)

BCC	- XOR of all the frame's characters except STX and ETX
ETX	- Character (0x03, 3d)

ID value:

Hexadecimal	Binary	ASCII	Scale type
0x68	011x1000	h	25 lb 1/8 oz
0x69	011x1001	i	15 kg 5 g
0x6a	011x1010	j	30 lb 0.01g
0x6b	011x1011	k	6 kg 2 g
0x6c	011x1100	l	50 lb 0.01 lb

If the POS receives the frame <STX><ID><PPPPP><BCC><ETX> (weight in range), it resends the same frame to the scale so that it can compare the frame sent and received.

When the scale has received the frame for comparison, it returns:

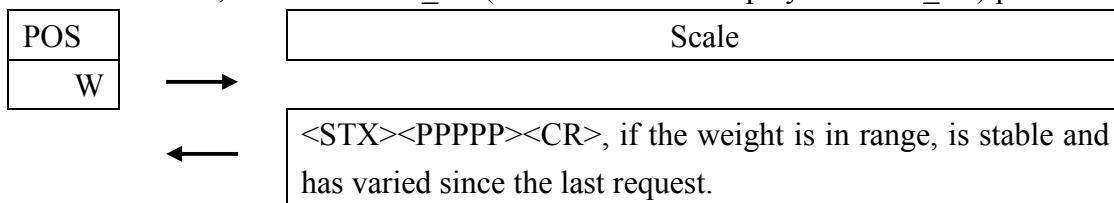
- | |
|--|
| - NAK (0x15, 21d), if there is an error in the reception. |
| - ACK (0x06, 6d), if the frame received does not match the frame sent. |
| - CR (0xd, 13d), if the frame received matches the frame sent. |

End of sequence.

If scale doesn't receive the frame, it must to wait

2.2 TPV 2B (*Uniwell-W Protocol*):

From the menu, select the TPV_2B (on the scale it is displayed as 7Pu_2B) protocol.

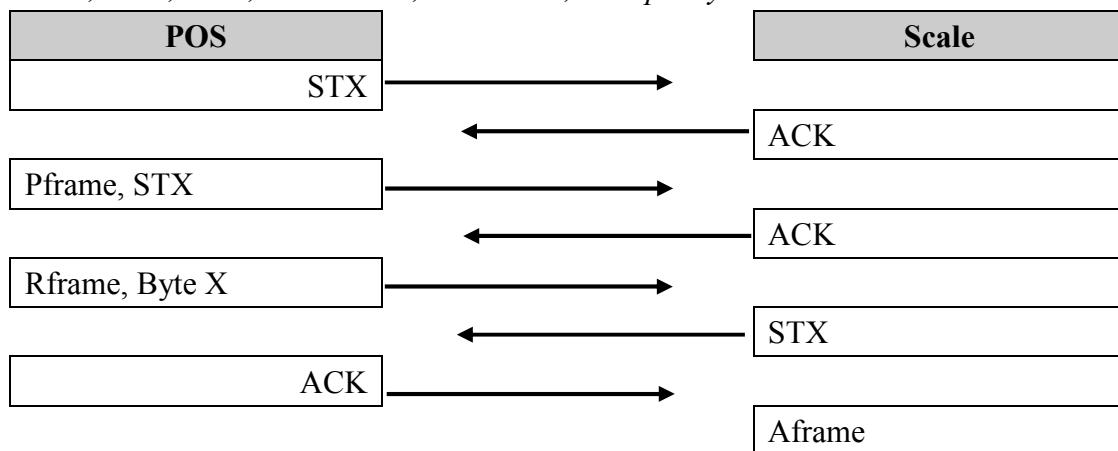


W	- Character 0x57, 87d
STX	- Character 0x02, 2d
PPPPP	- Value of the weight in ASCII without decimal point and left-justified with zeros (0x30, 48d)
CR	- Character (0x0d, 13d)

3 TPV 4 (Berkel Protocol)

From the menu, select the TPV_4 (on the scale it is displayed as 7Pu_4) protocol.

1200, 2400, 4800, 9600 bauds, 7 data bits, even parity.



Description of the programming frames

Pframe

@	1	Pu5 Pu4 Pu3 Pu2 Pu1 Unit price	CR	LF
---	---	-----------------------------------	----	----

MSB or highest weight digit is placed in Pu5. For example, 325 would be placed as 00325.

Rframe

1	2	CR	LF
---	---	----	----

Second byte (Byte X) can be any ASCII character between ‘0’ and ‘9’ (30H and 39H).

Description of the answer frame:

Aframe (weight, price and amount)

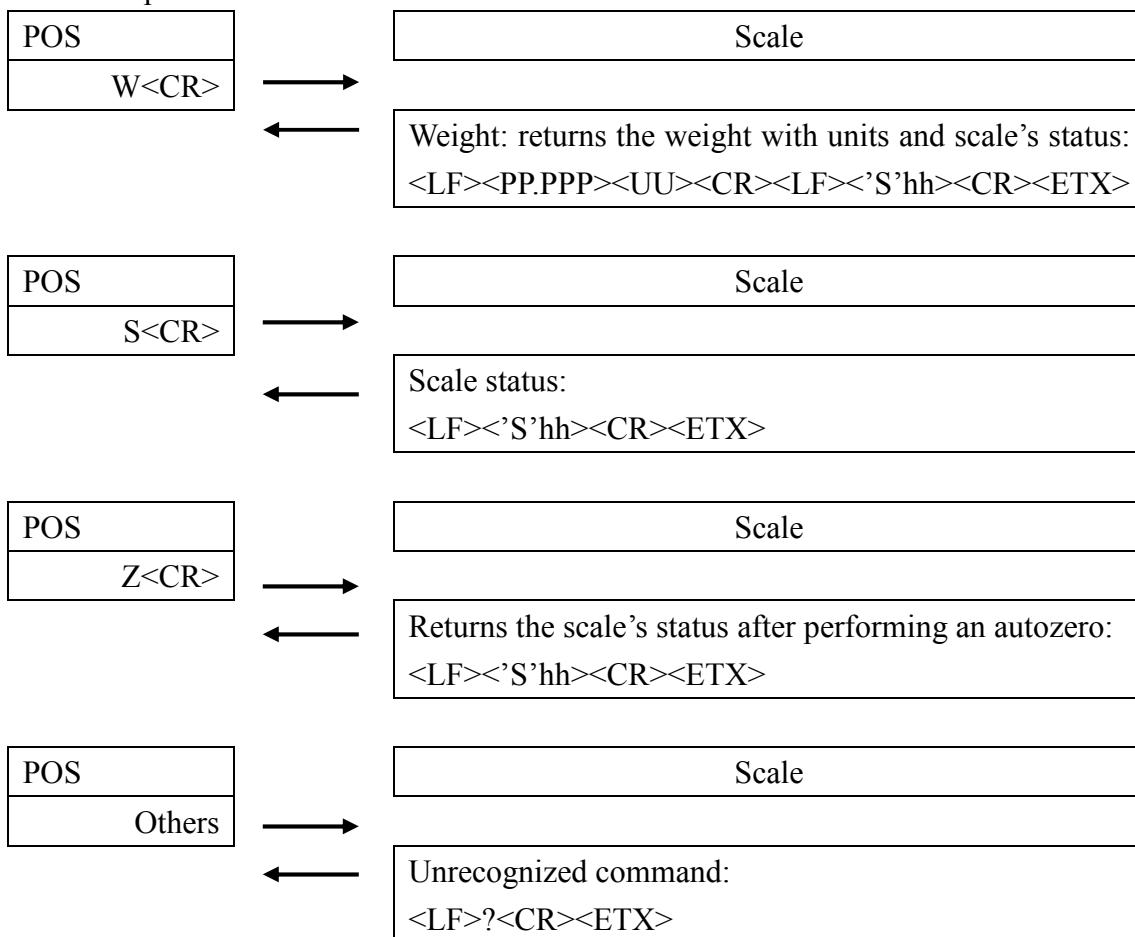
W5 W4 W3 W2 W1 WEIGHT	Pu5 Pu4 Pu3 Pu2 Pu1 PRICE	I6 I5 I4 I3 I2 I1 AMOUNT	0	CR	LF
--------------------------	------------------------------	-----------------------------	---	----	----

- 1- The data are numeric and are filled with leading zeros.
- 2- If the data are not numeric, it must be understood that it is impossible to perform the weight (weight out of range, unstable weight or zero weight). On the other hand, the scale does not return the answer to the TPV if there has been no weight variation since the last OK transaction.

4 TPV 7 (Samsung Protocol)

From the menu, select the TPV_7 (on the scale it is displayed as 7Pu_7) protocol.

This accepts 4 commands:



ETX	- Character (0x03, 3d)
LF	- Character (0x0a, 10d)
CR	- Character (0x0d, 13d)
PP.PPP	- Value of the weight with decimal point and left-justified with zeros (0x30)
UU	- Measuring units in upper case (KG, LB, OZ...)
'S'hh	- Character 'S' + 2 status bytes
?	- Character 0x3F

Coding of the status bytes:

Bit	First status byte	Second status byte
LSB 0	1 Scale in movement 0 Weight stable	1 Weight under 0 No Weight under
1	1 Scale at zero 0 Weight <> zero	1 Weight over 0 No Weight over
2	1 Ram error 0 No Ram error	1 Rom Error 0 No Rom error
3	1 Eprom error 0 No Eprom error	1 Calibration failure 0 Calibration OK
4	Always 1	Always 1
5	Always 1	Always 1
6	Always 0	1 There is a third status byte 0 Last byte
MSB 7	Always 0	Always 0

Most usually states:

Status	first status byte h ₁								second status byte h ₂								In ASCII
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
weigh 0, stable	0	0	1	1	0	0	1	0	0	0	1	1	0	0	0	0	20
weigh 0, unstable	0	0	1	1	0	0	1	1	0	0	1	1	0	0	0	0	30
weigh no 0, stable	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	00
weigh no 0, unstable	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	0	10
OVERLOAD, stable	0	0	1	1	0	0	0	0	0	0	1	1	0	0	1	0	02
OVERLOAD, unstable	0	0	1	1	0	0	0	1	0	0	1	1	0	0	1	0	12
under, stable	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	1	01
under, unstable	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	1	11

5 TEDC1 (CAS Protocol)

From the menu, select the TEDC1 (on the scale it is displayed as 7eDC1) protocol.

Communication Agreement

1. Baud Rate → 9,600 BPS
2. Data Bit → 8 BIT
3. Stop Bit → 1 BIT
4. Parity Bit → NO
5. Communication Level → RS-232C level
6. Data Format → ASCII
7. Command Definitions:

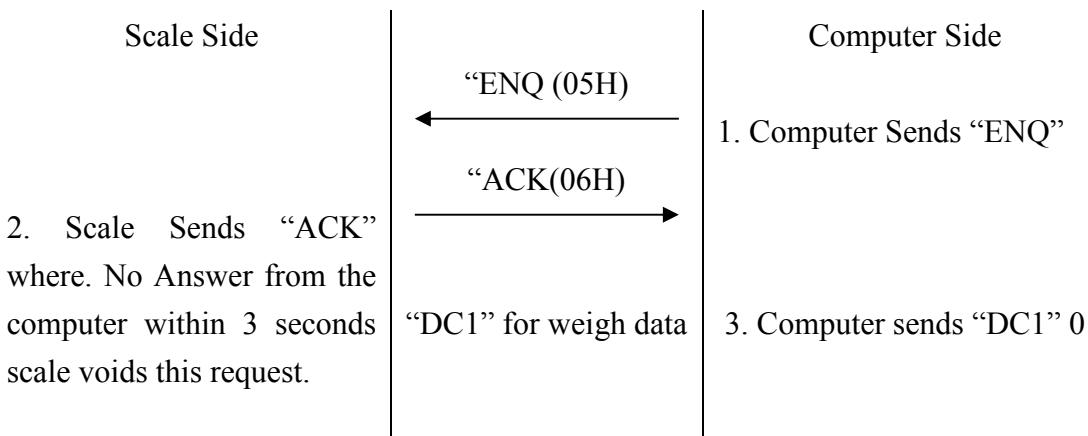
- | | |
|------------------|-------------------|
| 6-1. "ENQ" → 05H | 6-7. "EOT" → 04H |
| 6-2. "ACK" → 06H | 6-8. "DC1" → 11H |
| 6-3. "NAK" → 15H | 6-9. "DC2" → 12H |
| 6-4. "SOH" → 01H | 6-10. "DC3" → 13H |
| 6-5. "STX" → 02H | 6-11. "DC4" → 14H |
| 6-6. "ETX" → 03H | |

Wire Connections

The wire connections of the D-SUB 9 PIN connector of a computer side

Scale	Computer
TXD (PIN 2)	RXD (PIN3)
RXD (PIN 3)	TXD (PIN2)
GND (PIN 5)	GND (PIN7)
	<ul style="list-style-type: none"> └ RTS (PIN4) └ CTS (PIN5) └ DCD (PIN8) └ DSR (PIN6) └ DTR (PIN20)

The Protocol



Data Trains

The Data Trains for the “DC1”

<u>SO</u> <u>H</u>	<u>ST</u> <u>X</u>	<u>ST</u> <u>A</u>	<u>SIG</u> <u>N</u>	<u>W</u> <u>5</u>	<u>W</u> <u>4</u>	<u>W</u> <u>3</u>	<u>W</u> <u>2</u>	<u>W</u> <u>1</u>	<u>W</u> <u>0</u>	<u>UN</u> <u>1</u>	<u>UN</u> <u>0</u>	<u>BCC</u>	<u>ET</u> <u>X</u>	<u>EO</u> <u>T</u>
<u>Commands</u>	<u>Data Block</u>												Chec k	<u>Command</u> <u>s</u>

Remarks:

.STA → a weight status of the scale
Scale is stable → “S”. Not stable → “U”

.SIGN → sign of the weight data
Zero and positive weight → “+” Negative weight → “-“
Overload → “F”

.W5 through W0 → Weight data including decimal point <WW.WWW>
But all “F”s when the scale is put on overload

.Un1 through Un0 → Unit of weight (kg or lb)

.BCC → block check character
BCC is created by exclusive order(XOR) of all data block.

6 TOLEDO Protocol

The PC send character 'W' or 'w', The scale is automatically configured in this protocol and sends the following sequence:

if the weight is in range, is stable, is not zero, is positive;

<STX><PPPPP><CR>

else

<STX><'?'><H | 0x60><CR>

STX: Character 0x02;

PPPPP: The value of weight and with 5 bytes, without the weight's decimal point, field is left-justified with zero(0x30);

CR: Character 0x0d;

'?': Character ?; (0x3f)

H: Bit-----status byte

H₁ stable 0 / unstable 1

H₂ in range 0 / out of range 1

H₃ positive 0 / negative 1

H₄ weight is not zero 0 / weight is zero 1

<H | 0x60> → H XOR 0x60 → 0 0 0 0 H₄ H₃ H₂ H₁ XOR 1100000

0 0 0 0 H ₄ H ₃ H ₂ H ₁	0110 0000	ASCII
---	-----------	-------

Positive, unstable →	0000 0001	xor	0110 0000 = a
Overload, stable →	0000 0010	xor	0110 0000 = b
Overload, unstable →	0000 0011	xor	0110 0000 = c
Negative, stable →	0000 0100	xor	0110 0000 = d
Negative, unstable →	0000 0101	xor	0110 0000 = e
Zero, stable →	0000 1000	xor	0110 0000 = h
Zero, unstable →	0000 1001	xor	0110 0000 = i

7 NCIECR AND NCIGEN Protocol

The PC send character <W><CR> or <w><CR>, The scale is automatically configured in this protocol and sends the following sequence, if the weight is in range:
case NCIECR

protocol:<LF><PP.PPP><UU><CR><LF><'S'><s1><s2><CR><ETX>

case NCIGEN Protocol:<LF><PP.PPP><UU><CR><LF><s1><s2><CR><ETX>

else

case NCIECR protocol:<LF><00.000><UU><CR><LF><'S'><s1><s2><CR><ETX>

case NCIGEN Protocol:<LF><00.000><UU><CR><LF><s1><s2><CR><ETX>

LF: Character 0x0a;

PP.PPP: The value of weight and with 6 bytes, include the weight's decimal point, field is left-justified with zero(0x30);

CR: Character 0x0d;

UU: Measuring units in upper case(the scale always sends "KG");

'S': Character S

ETX: Character 0x03;

s1: Bit-----status byte

s2: Bit-----status byte

1 → stable 0 / unstable 1

1 → positive 0 / negative 1

2 → weight is not zero 0 / weight is zero 1

2 → in range 0 / out of range 1

Not zero, stable, in range, positive → 00

Not zero, stable, in range, negative → 01

Not zero, stable, out of range, positive → 02

Not zero, stable, out of range, negative → 03

Not zero, unstable, in range, positive → 10

Not zero, unstable, in range, negative → 11

Not zero, unstable, out of range, positive → 12

Not zero, unstable, out of range, negative → 13

Zero, stable, in range, positive → 20

Zero, stable, in range, negative → 21

~~Zero, stable, out of range, positive → 22~~ no possible condition

~~Zero, stable, out of range, negative → 23~~ no possible condition

Zero, unstable, in range, positive → 30

Zero, unstable, in range, negative → 31

~~Zero, unstable, out of range, positive → 32~~ no possible condition

~~Zero, unstable, out of range, negative → 33~~ no possible condition

8 TEC Protocol

The PC send character <ENQ> or <FF>, The scale is automatically configured in this protocol and sends the following sequence:

Computer Request: ENQ

Scale Response: if the weight is stable, returns ACK
if the weight is unstable, returns BEL

Computer Request: FF

Scale Response: if the weight is zero or negative ,Returns

<STX><0x7f><00000><BCC><ETX>

if the weight in range ,Returns <STX><N><PPPPP><BCC><ETX>

STX: Character 0x02;

N: if cap=6kg, res=6000,desc=3, then N='C' and the weight is positive N=NUL;

if cap=15kg, res=6000,desc=3, then N='A'

if cap=30kg, res=6000,desc=3, then N='B'

None of the above N='G'

PPPPP: The value of weight and with 5 bytes, without the weight's decimal point, field is left-justified with zero(0x30);

BCC: XOR checksum is created by exclusive order of a data block (from the second to BCC)

ENQ: Character 0x05;

ACK: Character 0x06;

BEL: Character 0x07;

FF: Character 0x0C;

NUL: Character 0x00;

9 EASY Protocol

The PC send some characters, The scale is automatically configured in this protocol and sends the following sequence:

Computer Request: 'R'

Scale Response: <STX><AD_OUT><CR>

Computer Request: 'F'

Scale Response:

<STX><WWWW.WW><UU><PPP.PPP><AAA.AAA><TTTT.TT><CR>

Computer Request: DC1

Scale Response: the value of calibration zero point

Computer Request: DC2

Scale Response: the value of current AD out minus value of calibration zero point

Computer Request: DC3

Scale Response: NULL

Computer Request: DC4

Scale Response: NULL

Computer Request: 'Z'

Scale Response: current zero point

Computer Request: 'S'

Scale Response: if in zero range, returns <ACK> after performing an autozero;
else Returns <NAK> only;

Computer Request: EOT

Scale Response: Returns <ACK> after stop continuum model ;

Computer Request: 'W'

Scale Response: start continuum model,

Returns <STX><Y><PPPPP><UU><H | 0x40><CR>;

'R': Character R;

STX: Character 0x02;

AD_OUT:the value of AD out (6 bytes)

CR: Character 0x0d;

'F': Character F;

'W': Character W

WWWW.WW:The value of weight and with 7 bytes,include the weight's decimal point,The weight field is left-justified with zero(0x30);

UU: Measuring units in upper case(the scale always sends "KG");

PPP.PPP:The value of price and with 7 bytes,include the weight's decimal point,The weight field is left-justified with zero(0x30);

AAA.AAA:The value of amount and with 7 bytes,include the weight's decimal point,The weight field is left-justified with zero(0x30);

TTTT.TT:The value of tare and with 7 bytes,include the weight's decimal point,The weight field is left-justified with zero(0x30); DC1:Character 0x11;

DC2:Character 0x12;

DC3:Character 0x13;

DC4:Character 0x14;

NAK:Character 0x15;

ACK:Character 0x06;

EOT:Character 0x04;

For computer request: 'W'

Y: positive Y='+' / negative Y='-'

PPPPPP: The value of weight and with 6 bytes,without the weight's decimal point,The weight field is left-justified with zero(0x30);

UU: Measuring units in upper case(the scale always sends "KG");

H: Bit-----status byte

1 stable 0 / unstable 1

2 in range 0 / out of range 1

3 positive 0 / negative 1

4 gross is not zero 0 / gross is zero 1

5 net is not zero 0 / net is zero 1

8 tare is not zero 1 / tare is zero 0

10 TISA

10.1 TISA cash register protocol(tisa) A

Protocol

Cash register

Scales



Where:

98: 0x39h & 0x38h

PPPPP: 5 digits for the price

C: checksum, logical sum (XOR) of all previous characters

Example:

If PPPPP = 00005

then C = 9 XOR 8 XOR 0 XOR 0 XOR 0 XOR 5 = 4

Frame 98000054C_RL_F

C_R: 0x0Dh

L_F: 0x0Ah

99: 0x39h & 0x39h

S: weight status

S: 0x30h correct (weigh>0, stable)

S: 0x31h error

WWWWW: 5 digits for WEIGHT Without decimal point

E: sum status

E: 0x30h correct

E: 0x31h error

IIIII: 6 digits for SUM

10.2 Protocol TISA with sending of stable weight(tisa_S)

Protocol

Cash register

Scales

Where:

98: 0x39h & 0x38h

PPPPP: 5 digits for the price

C: checksum, logical sum (XOR) of all previous characters

Example:

If PPPPP = 00005

then $C = 9 \text{ XOR } 8 \text{ XOR } 0 \text{ XOR } 0 \text{ XOR } 0 \text{ XOR } 5 = 4$

Frame 98000054C_RL_F

C_R: 0x0Dh

L_F 0x0Ah

99: 0x39h & 0x39h

S: weight status

S: 0x30h OK

S: 0x31h error

gits for WEIGHT

E: sum status

E: 0x0

E·0x31h error

Units for SUM

APPENDIX B: ARGUMENTS FOR DETERMINING THE NUMBER OF CLUSTERS

Note. once the price is received, the scale sends the data which the weight is stable, it is not negative and there is no error.

10.3 Protocol VD TISA(vd_tisa)

Protocol

Cash register

Scales

← 99SWWWWWWEIIIIICCR_RL_F

Where:

99:

0x39h & 0x39h

C:

checksum, logical sum (XOR) of all previous characters

C_R:

0x0Dh

L_F

0x0Ah

S:

weight status

S: 0x30h OK

S: 0x31h error

WWWWWW: 5 digits for WEIGHT. Without decimal point

E: amount status

E: 0x30h OK

E: 0x31h error

IIIII: 6 digits for AMOUNT

Note: the scale sends the data when the weight is stable, it is not negative, it is bigger than 0 and there is not error.