

SERIAL PROTOCOL

8408-14788-02

SCP-02

WEIGH-TRONIX

SCP-02

This document defines a special purpose serial communications protocol that is used to interface Weigh-Tronix/NCI POS (point-of-sale) scales to ECR (electronic cash registers), or other computing equipment (eg. personal computers).

It also defines extensions to "OPOS-enable" the scale for use with the Weigh-Tronix OPOS-Compliant Service Object (SO) ActiveX DLL.

TITLE:
SERIAL COMMUNICATIONS PROTOCOL:
SCP-02 (ECR STANDARD and OPOS EXTENSIONS)



PART: 8408-1478802	REV: B	ECO:	APPV'D: BA	08-JUN-01
--------------------	--------	------	------------	-----------

OVERVIEW

The purpose of this document is to define a special purpose serial communications protocol that may be used to interface Weigh-Tronix/NCI POS (Point-of-Sale) scales to ECR (Electronic Cash Registers), or other computing equipment (e.g. personal computers).

It also defines extensions to the basic ECR protocol that then enables the scale to be used in OPOS applications when used with a Weigh-Tronix OPOS Service-Object ActiveX control.

Weigh-Tronix provides an OPOS compliant Service-Object (SO) as an in-process ActiveX DLL component. This SO software communicates with a scale using the serial communications protocol defined in this document. The SO also communicates with another piece of software called a Common Control-Object (CCO). This CCO (for scales) is provided by the OPOS committee and is the only allowable interface that the POS application may use to access data and status from the scale hardware. The application makes a request to the CCO which passes it along to the SO, which formats it and transmits the appropriate command to the scale. The scale transmits a response back which is read by the SO, reformatted and passed up to the CO which then passes it along to the application level.

GOAL

A goal of this protocol document (and the associated Weigh-Tronix OPOS Service Object), is to provide a full robust OPOS interface for scales that support all OPOS Mandatory commands, as well as at least a limited OPOS interface for legacy scales that support only the ECR (standard) protocol subset.

COMMANDS

There are two sub-sets of commands that make up the “ECR” protocol. The first is the standard ECR subset used for legacy applications and the second is the set of extensions required for use in OPOS applications. Even though in combination they define a single “ECR” protocol, this document may make reference to individual commands as an “ECR” command or and “OPOS” command to distinguish their typical use.

There are two types of serial commands: *mandatory* and *optional*. Mandatory commands must be supported by all products. However, technical feasibility (e.g., memory constraints) and application requirements will determine the implementation of optional commands.

This protocol will support the serial communications requirements of our scales that must interface with electronic cash registers (ECR). This type of interface has several differences from the *NCI Standard Protocol* used in industrial bench scales.

The main differences between the two are as follows:

- If weight is negative, in motion, over capacity or under capacity, or if a zero error exists, the ECR protocol causes only the scale's status to be returned in response to a weight (**W**) command. In the general protocol the contents of the display are returned (WYSIWYG) as well as units and status.
- Status bytes in the ECR protocol are prefaced by the letter '**S**' but not in the standard protocol.
- Units of measure abbreviations in the ECR protocol are always upper case. The standard protocol adheres to the ANSI standard for abbreviations.
- In the ECR protocol, the weight is always positive, therefore the polarity byte is absent. In the general protocol, the weight field is WYSIWYG so polarity is handled automatically.

Summary of commands used in this protocol:

A, C, D, H, M, P, S, T, U, W, Z
m, p, s, t, u, X
<ENQ>

NCI ECR/OPOS SERIAL COMMUNICATIONS PROTOCOL

TITLE:	NCI ECR/OPOS Serial Communications Protocol
Desc:	NCI's standard ECR with OPOS extensions.
Comm:	RS-232, No-Handshake, Bi-Directional
Prot:	ECr
Baud:	Selectable
Note:	Used in scales only for interface to ECR.

Key to symbols used:

<ETX>	End of text control character	(0x03)
<LF>	Line Feed control character	(0x0A)
<CR>	Carriage Return control character	(0x0D)
<SP>	Space character	(0x20)
x	Weight digit characters	
S	A fixed single-character ('S') prefix for the status string.	
hh	Two-character status string (see note 5)	
UU	Two-character units-of-measure string in uppercase (e.g. "LB", "KG", "OZ", "GM")	

Notes:

- Unless otherwise specified, responses to serial commands will be immediate, or within one weight meter cycle of the scale.
- The scale accepts serial commands of the type: **x<CR>**
- The scale responds to unknown commands with: **<LF>?<CR><ETX>**
- Baud rate and parity will be configurable.
- Data bits will be fixed at seven (7).
- Start and stop bits will each be fixed at one (1).
- Modem control lines will not be supported.
- Baud rates supported will be 1200, 2400, 4800 and 9600

Command	Scale Response
---------	----------------

Name: Request weight

Used: ECR:Mandatory, OPOS:Mandatory

OPOS Ref: *ReadWeight* (Method)

Command: **W<CR>**

Response: Returns decimal weight, units and status. (see note 1 and 2)

<LF>xx . xxxUU<CR><LF>Shh<CR><ETX>

-or-

Returns lb-oz weight with units plus scale status.

<LF>xLB<SP>xx . xOZ<CR><LF>Shh<CR><ETX>

-or-

Scale status only if wt < 0, initial zero error, in motion or out of capacity.

<LF>Shh<CR><ETX>

OPOS Note: *The transmitted weight string will include the decimal point (if appropriate) and will be to the resolution as it is displayed on the scale. The Service Object will subsequently convert this ASCII text value to an equivalent long value (with an assumed fractional part of three digits) as required by the OPOS specification.*

Name: Request status

Used: ECR:Mandatory, OPOS:Mandatory

OPOS Ref: N/A

Command: **S<CR>**

Response: Returns scale status.

<LF>Shh<CR><ETX>

Name: Zero the scale

Used: ECR:Mandatory, OPOS:Mandatory

OPOS Ref: *ZeroScale* (Method)

Command: **Z<CR>**

Response: Scale is zeroed, returns scale status. (see note 4)

<LF>Shh<CR><ETX>

Command	Scale Response
---------	----------------

Name: Request high-resolution weight

Used: ECR:Optional, OPOS:N/A

OPOS Ref: N/A

Command: **H<CR>**

Response: Returns decimal wt in 10x with units plus scale status. (see notes 2 and 3)
<LF>xxx.xxxxUU<CR><LF>Shh<CR><ETX>

-or-

Returns lb-oz wt in 10x with units plus scale status.

<LF>xLB<SP>xx.xxOZ<CR><LF>Shh<CR><ETX>

-or-

Scale status only if wt < 0, initial zero error, in motion or out of capacity.

<LF>Shh<CR><ETX>

Name: Change units of measure

Used: ECR:Optional, OPOS:N/A

OPOS Ref: N/A

Command: **U<CR>**

Response: Changes units of measure, returns new units and scale status.

<LF>uu<CR><LF>Shh<CR><ETX>

Name: Request metrology raw counts

Used: ECR:Optional, OPOS:N/A

OPOS Ref: N/A

Command: **M<CR>**

Response: Returns normalized raw counts and scale status.

<LF>xxxxxxMM<CR><LF>Shh<CR><ETX>

Name: unrecognized command

Used: ECR:Mandatory, OPOS:Mandatory

OPOS Ref: N/A

Command: all others

Response: Unrecognized command

<LF>?<CR><ETX>

Command	Scale Response
---------	----------------

Name: Request Units-of-Measure

Used: ECR:N/A, OPOS:Mandatory

OPOS Ref: *WeightUnits* (Property)

Command: **u**<CR> (lowercase 'u')

Response: Returns current units-of-measure as a single numeric character.
<LF>x<CR><ETX>

Where: **x** = '1' : Grams
= '2' : Kilograms
= '3' : Ounces
= '4' : Pounds

OPOS Note: *This function is provided for the sake of completeness since the units-of-measure can be extracted from the standard weight response string.*

Name: Request Scale Capability Attributes

Used: ECR:N/A, OPOS:Mandatory

OPOS Ref: Combined *CapDisplay*, *CapDisplayText*, *CapPriceCalculating*, *CapTareWeight*, *CapZeroScale* (Properties)

Command: **A**<CR>

Response: Returns scale OPOS capabilities as a string of five flag-characters.
<LF>vwxyz<CR><ETX>

Where: **v** = 'T' Scale has a weight display (*CapDisplay*)
v = 'F' Scale does not have a weight display

w = 'T' Scale has a text display (*CapDisplayText*)
w = 'F' Scale does not have a text display

x = 'T' Scale can calculate unit price (*CapPriceCalculating*)
x = 'F' Scale cannot calculate unit price

y = 'T' Scale allows setting tare value (*CapTareWeight*)
y = 'F' Scale does not allow setting tare value

z = 'T' Scale may be zeroed (*CapZeroScale*)
z = 'F' Scale cannot be zeroed

Command	Scale Response
---------	----------------

Name: Request Maximum Weight
Used: ECR:N/A, OPOS:Mandatory
OPOS Ref: *MaximumWeight* (Property)
Command: **m**<CR> (lowercase 'm')
Response: Returns current maximum weight supported by the scale as a string value, without decimal point, and in the currently enabled units-of-measure.
 <LF>**xx**<CR><ETX>

Name: Request Maximum Text Display Size
Used: ECR:N/A, OPOS:Optional
OPOS Ref: *MaxDisplayTextChars* (Property)
Command: **C**<CR>
Response: Returns the maximum number of text characters that may be displayed on the scales dedicated message display as a variable length string value (e.g. 1 – 999).
 <LF>**x**<CR><ETX> or
 <LF>**xx**<CR><ETX> or
 <LF>**xxx**<CR><ETX> or

Name: Request Computed Sales Price
Used: ECR:N/A, OPOS:Optional
OPOS Ref: *SalesPrice* (Property)
Command: **s**<CR> (lowercase 's')
Response: Returns the calculated sales price as a string value.
 <LF>**xxx.xx**<CR><ETX> (typical response)
Note: Please see the section entitled 'Sales Price and Unit Price Commands' on page 11 for details of price string value (i.e. currency) limitations.

Name: Display Message Text
Used: ECR:N/A, OPOS:Optional
OPOS Ref: *DisplayText* (Method)
Command: **Dccccccc . . .**<CR>
Response: Displays the desired text message "cccccc...", then returns:
 <LF><ACK><CR><ETX>

Name: Set Tare
Used: ECR:N/A, OPOS:Optional
OPOS Ref: *TareWeight* (Property)
Command: **Txx . xxx**<CR>

Response: Set the specified tare value, then return status.
<LF><ACK><CR><ETX>

Command	Scale Response
---------	----------------

Name: Read Tare
Used: ECR:N/A, OPOS:Optional
OPOS Ref: *TareWeight* (Property) support to allow reading currently set tare weight.
Command: **t**<CR> (lowercase 't')
Response: Return the current tare value, units and status.
<LF>**xx . xxxUU**<CR><LF>**Shh**<CR><ETX>

Name: Set Unit Price
Used: ECR:N/A, OPOS:Optional
OPOS Ref: *UnitPrice* (Property)
Command: **Pxxx . xx**<CR> (typical value)
Response: Set the specified unit-price value, then return status.
<LF><ACK><CR><ETX>
Note: *Please see the section entitled 'Sales Price and Unit Price Commands' on page 11 for details of price string value (i.e. currency) limitations.*

Name: Read Unit Price
Used: ECR:N/A, OPOS:Optional
OPOS Ref: *UnitPrice* (Property) support to allow reading currently set UnitPrice value.
Command: **p**<CR> (lowercase 'p')
Response: Returns the currently active sales price as a string value.
<LF>**xxx . xx**<CR><ETX> (typical response)
Note: *Please see the section entitled 'Sales Price and Unit Price Commands' on page 11 for details of price string value (i.e. currency) limitations.*

Name: Scale Enquiry
Used: ECR:N/A, OPOS:Mandatory
OPOS Ref: N/A
Command: <ENQ><CR>
Response: Depends on the scale (and protocol).
<LF>**OPOS**<CR><ETX> *OPOS-enabled scale (OPOS protocol)*
-or-
<LF>**?**<CR><ETX> *Other scale (e.g. ECR protocol)*

Name: Request Price Computing Transaction
Used: ECR:N/A, OPOS:Optional
OPOS Ref: WT_MFGCMD_ReadTransaction (Vendor-Specific Method)
Command: X<CR>
Response: Returns decimal weight, units-of-measure, scale status, unit-price and total sales-price in a single consolidated string response. This guarantees that all of the price computing data values are associated in time. (see note 1 and 2)

<LF>xx.xxxUU<CR><LF>Shh<CR><LF>xxx.xx<CR><LF>xxx.xx<CR><ETX>

-or-

Scale status only if wt < 0, initial zero error, in motion or out of capacity.

<LF>Shh<CR><ETX>

Note: *Please see the section entitled 'Sales Price and Unit Price Commands' on page 11 for details of price string value (i.e. currency) limitations.*

OPOS Note: *The transmitted weight string will include the decimal point (if appropriate) and will be to the resolution as it is displayed on the scale. The Service Object will subsequently convert this ASCII text value to an equivalent long value (with an assumed fractional part of three digits) as required by the OPOS specification.*

NOTES:

- 1) Weight field is always six characters (5 for weight, 1 for decimal point), regardless of display size.
- 2) Leading zeroes are not suppressed.
- 3) High resolution weight field is seven characters (6 for weight, 1 for decimal point), regardless of display size.
- 4) If scale is outside zero range or in motion the 'Z' command is ignored and scale status is returned without zero status flag set, i.e., scale not at zero.
- 5) Status bits are defined the same as for the NCI Standard protocol.

Weight Command: W

This command causes the scale to return the displayed weight string in ECR format. The scale will usually return the displayed weight with leading zeroes (i.e. no leading zero suppression). However, if weight is negative, in motion, overcapacity or undercapacity, or if a zero error exists, the ECR protocol causes only the scale's status to be returned in response to a weight (**w**) command.

The returned weight string will include decimal point plus units of measure. The length of the weight field will be equal to the length of the scale's display plus three (one for the decimal point, and two for the units, e.g., "LB"). For pounds-ounce weight, the length of the weight field will be equal to the length of the scale's display plus five (one for the decimal point, two for the "LB" and two for the "OZ"). Units of measure will appear in their ANSI standard abbreviated form ("LB" for pounds, "KG" for kilograms, etc.) in uppercase characters.

Zero Command: Z

If zeroing criteria are met, the scale is zeroed. In any case, scale status is returned.

High-Resolution Command: H

This is the same as the **w** command except that when weight is returned, it is returned with ten times the scale's displayed resolution. Thus, for decimal weight, the length of the weight field is equal to the length of the scale's display plus three or four, and for pounds-ounce it will be the display length plus five or six (depending on the presence of a decimal point).

Change Units Command: U

The effect of this command will be the same as if the *UNITS* key on the scale were pressed, causing the scale to change its units of measure. In response, the scale will return the new units of measure and scale status.

Metro Command: M

This command is optional. It will cause the scale to return normalized raw counts used for metrology verification. The count value is the same number displayed in the scale's division test in its diagnostics mode. For example, this number ranges up to 120,000 for the 67xx, 100,000 for the 78xx and 1,000,000 for QDT™ scales.

Sales Price ('s') and Unit Price ('P'/'p') Commands:

These commands are used to set/read the unit price and to read the sales price. The string values passed (and/or returned) are considered 'currency' values as defined in the OPOS specification.

Depending on the scale application and country of use, the currency values for unit price and sales price may or may not have a decimal point. In addition, it may have from one to four places to the right of a decimal point. Typical usage will be two places to the right of the decimal point. Currency values can support up to fifteen places to the left of the decimal point, but in all practical uses it is usually limited to about seven places maximum. Negative values (i.e. strings with '-' preceding the value) are not allowed.

The following are examples of valid price strings:

X
X.
X.XX
X.XXX
X.XXXX
XX.XXXX
:
:
XXXXXXXX.XXXX

The following are examples of invalid price strings:

X.XX	(multiple decimal points not allowed)
-X.XX	(negative sign not allowed)
+X.XX	(positive sign is assumed)
\$X.XX	(currency sign not allowed)

Scale Status Command: S

There will be at least two status bytes. If bit 6 of the second status byte is set then there will be a third byte. Bit 6 of each successive byte will indicate whether or not another status byte follows. The bit will be clear (0) in the last status byte. At this time only the first three bytes are defined. Others may be added in the future. Bit 0 is the least significant bit in the byte while bit 7 is the most significant bit.

The status bits are defined as follows:

Bit	Status Byte 1	Status Byte 2	Status Byte 3 (opt)
0	1 = Scale in motion 0 = Stable	1 = Under capacity 0 = Not under capacity	} 00 = Low range 01 = (undefined) } 10 = (undefined) 11 = High range
1	1 = Scale at zero 0 = Not at zero	1 = Over capacity 0 = Not over capacity)
2	1 = RAM error 0 = RAM okay	1 = ROM error 0 = ROM okay	1 = Net weight 0 = Gross weight
3	1 = EEPROM error 0 = EEPROM okay	1 = Faulty calibration 0 = Calibration okay	Initial zero error
4	Always 1	Always 1	Always 1
5	Always 1	Always 1	Always 1
6	Always 0	1 = Byte follows 0 = Last byte	1 = Byte follows 0 = Last byte
7	Parity	Parity	Parity