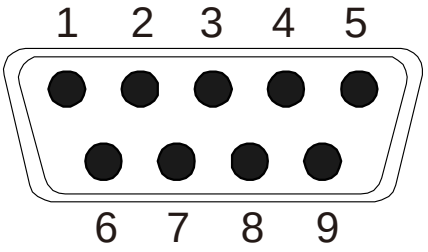
TVD, TVD-D RS-232 OUTPUT

1. RS-232 (9pin D type connector)



Pin 2	RXD	Input	Receiving data
Pin 3	TXD	Output	Transmission data
Pin 5	GND	—	SIgnal ground

9pin D Connecter:

Scale	**Computer**
Pln 2:	Pln 3
Pln 3:	Pln 2
Pln 5:	Pln 5

2. Specifications:

RS-232 output of weighing data
ASCII code
600~9600 Baud
8 data blts
No Parity

3. Continuously output protocol

S	T	,	G	S	,	-/␣							k	g	,	CR	LF
H1			H2			WEIGHT							W.U				

H1=HEADER1: ST=STABLE, US=UNSTABLE
H2=HEADER2: GS=GROSS, NT=NET
W.U=WEIGHT UNIT
TERMINATOR= CR LF

4. Toledo

W = 57 <STX> = 02 <CR> = 0D ? = 3F

Register	Scale	Action
W---------->		The register requests the weight from the scale.
<---- <STX> $X_1X_2X_3X_4X_5$ <CR>		***If (Weight ≠ 0) && (Weight > 0) && (Weight <= Capacity + 9 divisions) && (Weight is Stable) then:*** The scale transmits the weight in 7 bytes where <STX> is the first byte, X_1 Is the 2nd byte and represents the MSD (Most Significant Digit) of the weight value, etc, X_5 Is the 6th byte and represents the LSD (Least Slgnlflcant Dlglt) of the weight, and <CR> is the 7th and final byte. In this protocol the register decides the decimal place and units (lb or kg.) Must use leading zeros whenever necessary.
<---- <STX> ? Y_1 <CR>		***Else:*** The scale transmlts a status response in 4 bytes where <STX> is the first byte, 3F is the 2nd byte, Y_1 Is the 3rd byte and represents the status byte, and <CR> is the 4th and final byte. ***End if***

Status Byte Definition:

Motion	Y_1 = 61	a	01100001
~~**Scale at ZERO**~~ **Weight < 0**	Y_1 = 70	p	01110000
	Y_1 = 64	d	01100100
Weight > Capacity	Y_1 = 62	b	01100010
Weight < 0 & Motion	Y_1 = 65	e	01100101
Weight > Capacity & Motion	Y_1 = 63	c	01100011

Status Byte Formation		
Bit	**Description**	
Bit 7 (MSB)	Parity Bit.	
Bit 6	(Not used.)	**Always = 1**.
Bit 5	Net Weight Bit.	**Gross = 0**, Net = 1.
Bit 4	Zero Bit.	Zero = 1, Not Zero = 0.
Bit 3	Outside Zero Range Bit.	**Within = 0**, Outside = 1.
Bit 2	Negative Weight Bit.	Negative = 1, Non-Negative = 0.
Bit 1	Overload Bit.	Overload = 1, Non-Overload = 0.
Bit 0 (LSB)	Motion Bit.	Motion = 1, Stable = 0.

Example 1:

If weight is 21.30 lb and the scale is stable, it will transmit the following 7 bytes:

02	30	32	31	33	30	0D
<STX>	0	2	1	3	0	<CR>

Example 2:

If weight is unstable, it will transmit the following 4 bytes:

02	3F	61	0D
<STX>	?	a	<CR>

Exceptions: If Weight = 12345.6 then $X_1X_2X_3X_4X_5X_6$ = 12345€

5. NCI-ECR

W = 57 <CR> = 0D <LF> = 0A L = 4C B = 42 K = 4B G = 47 S = 53 <ETX> = 03 . = 2E

Register	Scale	Action
W---------->		The register requests the weight from the scale by sending a W followed by a <CR>.
<CR>---------->		
<--<LF> $X_1X_2X_3X_4X_5X_6U_1U_2$ <CR>		<LF> $S_1S_2S_3$ <CR> <ETX> **The scale always responds to a valid request.** The scale transmits the weight in 16 bytes where:

<LF> is the first byte.

X_1 to X_6 are bytes 2 thru 7 and represent the weight value **including the decimal point.** Therefore, a weight of 3.02 will be 003.02 = X_1 to X_6, or a weight of 3.002 will be 03.002 = X_1 to X_6.

U_1 to U_2 are bytes 8 & 9 and represent the unit. For pounds U_1 = 4C = "L" and U_2 = 42 = "B"; for kilos U_1 = 4B = "K" and U_2 = 47 = "G".

<CR> Is the 10th byte.

<LF> Is the 11th byte.

S_1 Is the 12th byte and is always S_1 = 53 = "S".

S_2 to S_3 are the 13th & 14th bytes. They form the status word.

<CR> Is the 15th byte.

<ETX> Is the 16th byte.

Status Word Definition:

OK (Stable)	S_2 = 30 S_3 = 30	"00"
Motion	S_2 = 31 S_3 = 30	"10"
Scale at ZERO	S_2 = 32 S_3 = 30	"20"
Weight < 0	S_2 = 30 S_3 = 31	"01"
Weight > Capacity	S_2 = 30 S_3 = 32	"02"
Motion & Weight < 0*	S_2 = 31 S_3 = 30	"11"
Motion & Weight > Capacity*	S_2 = 31 S_3 = 30	"12"

Note*: Whenever the Weight > **Max Capacity + 9 divisions** then the scale must transmit a "zero" weight value. This is 0.00 or 0.000 or whatever the "zero" weight needs to be for that scale's capacity/resolution/unit setting.

Bit	S_2 Bit Description	S_3 Bit Description
7 (MSB)	Parity Bit.	Parity Bit.
6	(Not used) **Always = 0**.	(Not used) **Always = 0**.
5	(Not used) **Always = 1**.	(Not used) **Always = 1**.
4	(Not used) **Always = 1**.	(Not used) **Always = 1**.
3	(Not used) **Always = 0**.	(Not used) **Always = 0**.
2	(Not used) **Always = 0**.	(Not used) **Always = 0**.
1	Zero Bit: Zero = 1, Non-Zero = 0.	Overload Bit: Overload = 1, Non-Overload = 0.
0 (LSB)	Motion Bit: Motion = 1, Stable = 0.	Negative Weight Bit: Negative = 1, Non-Negative = 0.

Example:

If weight is 21.30 lb and the scale is **Stable**, it will transmit the following 16 bytes:

0A	30	32	31	2E	33	30	4C	42	0D	0A	53	30	30	0D	03
<LF>	0	2	1	.	3	0	L	B	<CR>	<LF>	S	0	0	<CR>	<ETX>

6. NCI-General

W = 57 <CR> = 0D <LF> = 0A L = 4C B = 42 K = 4B G = 47 <ETX> = 03 . = 2E

Register	Scale	Action
W---------->		The register requests the weight from the scale by sending a W followed by a <CR>.
<CR>---------->		
<--<LF> $X_1X_2X_3X_4X_5X_6U_1U_2$ <CR>		<LF> S_1S_2 <CR> <ETX> **The scale always responds to a valid request.** The scale transmits the weight in 15 bytes where:

<LF> is the first byte.

X_1 to **X_6** are bytes 2 thru 7 and represent the weight value **including the decimal point.** Therefore, a weight of 3.02 will be 003.02 = **X_1** to **X_6**, or a weight of 3.002 will be 03.002 = **X_1** to **X_6**.

U_1 to **U_2** are bytes 8 & 9 and represent the unit. For pounds **U_1** = 4C = "L" and **U_2** = 42 = "B"; for kilos **U_1** = 4B = "K" and **U_2** = 47 = "G".

<CR> Is the 10th byte.

<LF> Is the 11th byte.

S_1 to **S_2** are the 12th & 13th bytes. They form the status word.

<CR> is the 14th byte.

<ETX> Is the 15th byte.

Status Word Definition:

OK (Stable)	S_2 = 30 S_3 = 30	"00"
Motion	S_2 = 31 S_3 = 30	"10"
Scale at ZERO	S_2 = 32 S_3 = 30	"20"
Weight < 0	S_2 = 30 S_3 = 31	"01"
Weight > Capacity*	S_2 = 30 S_3 = 32	"02"
Motion & Weight < 0	S_2 = 31 S_3 = 30	"11"
Motion & Weight > Capacity*	S_2 = 31 S_3 = 30	"12"

Note*: Whenever the Weight > **Max Capacity + 9 divisions** then the scale must transmit a "zero" weight value. This is 0.00 or 0.000 or whatever the "zero" weight needs to be for that scale's capacity/resolution/unit setting.

Bit	S_1 Bit Description	S_2 Bit Description
7 (MSB)	Parity Bit.	Parity Bit.
6	(Not used) **Always = 0**.	(Not used) **Always = 0**.
5	(Not used) **Always = 1**.	(Not used) **Always = 1**.
4	(Not used) **Always = 1**.	(Not used) **Always = 1**.
3	(Not used) **Always = 0**.	(Not used) **Always = 0**.
2	(Not used) **Always = 0**.	(Not used) **Always = 0**.
1	Zero Bit: Zero = 1, Non-Zero = 0.	Overload Bit: Overload = 1, Non-Overload = 0.
0 (LSB)	Motion Bit: Motion = 1, Stable = 0.	Negative Weight Bit: Negative = 1, Non-Negative = 0.

Example:

If weight is 11.300 kg and the scale is **Stable**, it will transmit the following 15 bytes:

0A	31	31	2E	33	30	30	4B	47	0D	0A	30	30	0D	03
<LF>	1	1	.	3	0	0	K	G	<CR>	<LF>	0	0	<CR>	<ETX>

7. TEC

<ENQ> = 05, <ACK> = 06, <NAK> = 15, <BEL> = 07, <DC1> = 11, <DC2> = 12, <STX> = 02, <ETX> = 03, <DEL> = 7F, <NUL> = 00

Register	Scale	Action
<ENQ>_____>		1) The register establishes communications by sending <ENQ>.
		2) **If (Weight is stable) then**
	<_____<ACK>	The scale will transmit <ACK>.
		Else
	<_____<BEL>	The scale will transmit <BEL>. Go back to step 1.
		End If
<DC2>_____>		3) The register requests the weight by sending <DC2>.
		4) Scale will transmit the following 9 bytes:
	<_____<STX>	Start of text.
	<_____<ID>	Identification byte defined bellow.
	<_____<W_5>	MSD (Most significant digit) of weight data.
	<_____<W_4>	2nd MSD (Most significant digit) of weight data.
	<_____<W_3>	3rd MSD (Most significant digit) of weight data.
	<_____<W_2>	4th MSD (Most significant digit) of weight data.
	<_____<W_1>	LSD (Least significant digit) of weight data.
	<_____<BCC>	Block check character defined below.
	<_____<ETX>	End of text.
		5) **If (Register verified data correctly) then**
<ACK>_____>		The register will transmit <ACK>. Go back to step 1.
		Else
		Go back to step 1.
		End If

Identifier Byte Definition

7F	<DEL>	If (Weight < 0) OR (Weight > Max Capacity + 9 divisions) then
		<ID> = 7F
		$W_5 = W_4 = W_3 = W_2 = W_1 = 30$; however, W_1 or W_5 can be <NUL> sometimes (see below.)
		Else
		Follow the <ID> codes below.
		End If

41	A	**Not Used**
42	B	**Not Used**
43	C	**Not Used**
44	D	**Not Used**
45	E	For 120 lb or 300 lb scales with 2 decimal places: Format **0 00 |b**
46	F	**Not Used**

<BCC> Definition

The BCC character is formed by performing an XOR operation on the following 6 bytes: <ID>, W_5, W_4, W_3, W_2, and W_1

<BCC> = <ID> XOR **W_5** XOR **W_4** XOR **W_3** XOR **W_2** XOR **W_1**

See the examples below for the formation of the BCC character.

Example 1:

If weight is 250.05 lb, you are using a 300 x 0.05 lb scale, the scale is **Stable**, and F16 = Even, it will transmit the following 9 bytes:

02	45	32	35	30	30	35	77	03
<STX>	<ID>	2	5	0	0	5	<BCC>	<ETX>

Example 2:

If weight is 39.55 lb, you are using a 300 x 0.01 lb scale, the scale is **Stable**, and F16 = Even, it will transmit the following 9 bytes:

02	45	0	33	39	35	35	4F	03
<STX>	<ID>	<NUL>	3	9	5	5	<BCC>	<ETX>

Example 3:

If weight is -5.01 lb, you are using a 300 x 0.05 lb scale, the scale is **Stable**, and F16 = Even, it will transmit the following 9 bytes:

02	7F	30	30	30	30	30	4F	03
<STX>	<ID>	0	0	0	0	0	<BCC>	<ETX>

8. Easy

R = 52 <STX> = 02 <CR> = 0D R = 52 L = 4C B = 42 K = 4B G = 47 . = 2E F = 46 W = 57 Z = 5A
<DC1> = 11 <DC2> = 12 <DC3> = 13 <DC4> = 14 S = 53 <EOT> = 04

Register	Scale	Action
R---------->		The register requests the Raw A/D Counts from the scale.
<---- <STX> $X_1X_2X_3X_4X_5X_6$ <CR>		The scale transmits the Raw A/D Counts in 8 bytes where <STX> is the first byte, X_1 Is the 2nd byte and represents the MSD (Most Significant Digit) of the data, etc, X_6 Is the 7th byte and represents the LSD (Least Significant Digit) of the data, and <CR> is the 8th and final byte. **The Raw A/D Counts are like the counts displayed in F9 CAL mode.**

Example:

If the Raw A/D counts are 22,130 counts then it will transmit the following 8 bytes:

02	30	32	32	31	33	30	0D
<STX>	0	2	2	1	3	0	<CR>

<DC1>---------->		The register requests the CZP data from the scale. This is the Raw A/D counts with no load on the scale.
<---- <STX> $X_1X_2X_3X_4X_5X_6$ <CR>		The scale transmits the CZP in 8 bytes where <STX> is the first byte, X_1 is the 2nd byte and represents the MSD (Most Significant Digit) of the data, etc, X_6 Is the 7th byte and represents the LSD (Least Significant Digit) of the data, and <CR> is the 8th and final byte.

Example:

If the CZP is 2,542 counts then it will transmit the following 8 bytes:

02	30	30	32	35	34	32	0D
<STX>	0	0	2	5	4	2	<CR>

<DC2>---------->		The register requests the CSP data from the scale. This is the Calibrated Span Point of the scale and is expressed as the Raw A/D counts with the full capacity load on the scale. The CSP should not be Zero Adjusted.
<---- <STX> $X_1X_2X_3X_4X_5X_6$ <CR>		The scale transmits the CSP in 8 bytes where <STX> is the first byte, X_1 is the 2nd byte and represents the MSD (Most Significant Digit) of the data, etc, X_6 Is the 7th byte and represents the LSD (Least Significant Digit) of the data, and <CR> is the 8th and final byte.

Example:

If the CSP is 202,542 counts then it will transmit the following 8 bytes:

02	32	30	32	35	34	32	0D
<STX>	2	0	2	5	4	2	<CR>