SMART WEIGHING SOLUTIONS

32x Series Digital Indicator Communications Manual

003R-632-200-M10

Copyright

Disclaimer

Rinstrum Pty Ltd reserves the right to make changes to the products contained in this manual in order to improve design, performance or reliability.

The information in this manual is believed to be accurate in all respects at the time of publication, but is subject to change without notice. Rinstrum Pty Ltd assumes no responsibility for any errors or omissions and disclaims responsibility for any consequences resulting from the use of the information provided herein.

SPECIAL NOTE
Trade Use of the Instrument

This manual may occasionally make reference to Trade Use settings of the instrument.

Some individual settings may not be legal for trade use. Please check regulations with the appropriate Weights and Measures Authority.

Table of Contents

1. Introduction

This manual details the communications protocol (**COMM**) of the indicator.

This protocol is used by the **Viewer** program to calibrate and configure the instrument. The **Viewer TEST** tab can be used as a convenient terminal to test the instrument communications.

The indicator is fitted with **opto-LINK** communications as standard. This allows a temporary isolated communications link to be established with a PC permitting the instrument software to be upgraded. It also allows the use of computerised setup and calibration via the **Viewer** software. The instrument also has an RS-232 port, which may be used for printing, connection to a remote display or control using **COMM**. The instrument can utilise the protocol to fully control the indicator.

Figure 1: Weight Indicator

1.1. Intended Audience

This manual is aimed at the designer or installer who is familiar with the operation and setup of the instrument.

1.2. Scope

This manual lists the commands and registers for the protocol. In practice, only a small subset of these commands would be used to control operational parameters.

1.3. The Manuals Set

This manual is part of a set of manuals covering the setup and operation of the instrument. The set includes the following:

- **Reference Manual** - Contains detailed information on calibration and setup. This manual is intended for use by Scale Technicians who are installing the instrument.
- **Operator Manual** - Aimed at the operation of the instrument, and covers the day-to-day operation of the instrument.
- **Quick Start Manual** - Intended for Scale Technicians who are familiar with the instrument and simply need a quick reference to menu options and connection diagrams, etc.
- **Communications Manual** - Contains details on the extended networking capabilities (communications protocol).

1.4. Document Conventions

The following document conventions (typographical) are used throughout this Communications Manual.

Bold Text	Bold text denotes words and phrases to note.
^	This symbol denotes one space (used in Commands)
…	Ellipses indicate an incomplete listing. For space considerations in this Manual complete listings of returned Command responses may not be shown.
1234_H	The subscript $_H$ indicates numerical values expressed in hexadecimal (radix 16).
1010_b	The subscript $_b$ indicates a numerical value expressed in binary (radix 2). Also, typically, spaces are used every 4 bits to break the string up visually, e.g., $1100\ 1101_b$.
[…]	Items enclosed in square brackets are optional.
<char>	<char> denotes a special character value. eg. <CR> is carriage return.
↵	This symbol indicates a <CR><LF> pair of characters.
1234	This font indicates message **sent** to an indicator.
1234	This font indicates a **response** from an indicator.
“string”	String values are enclosed in double quotes “”
⊗	Items marked with ⊗ indicate that the setting is available only in Full Setup and is trade critical. When trade critical settings are changed the calibration counter will be incremented.
REG_...	Register name. Refer to Register Identifiers page 37

2. Connecting to the Instrument

2.1. opto-LINK

A temporary infrared communications link can be established between any instrument and a PC using an optional opto-LINK cable. The opto-LINK cable can be used to transfer setup information from a PC and download software upgrades to the instrument. This setup information can be stored for later use and/or transferred to other instruments.

The PC end of the opto-LINK cable is a standard COM port (female DB9) or USB connector. The instrument end of the cable consists of an infrared transceiver, which attaches to the left side of the instrument display. To facilitate a quick and simple connection, the infrared transceiver is secured in place by a permanent magnet located within the head of the opto-LINK.

WARNING
The opto-LINK head contains a strong magnet and care should be taken with its proximity to electronic media (eg. credit cards, floppy disks, etc.) and/or other electronic instrumentation.

Note: The serial connection for the opto-LINK is standard to this instrument.

2.1.1. opto-LINK Activation

This feature is used to temporarily connect a PC to the instrument for calibration and setup purposes.

A long press of the **GROSS/NET** key will toggle the opto-LINK infrared communications On/Off.

When the opto-LINK has been enabled the following will occur:

- The instrument briefly displays the prompt **opto-L**.
- The editing annunciators (ie. GRP, ITM, etc.) will flash while the instrument searches for activity. During this period, the instrument also disables the RS-232 communications.
- **Activity Located:** If the instrument is **successful** in locating activity, the editing annunciators will continue to flash during the entire period of communications.
- **No Activity Located:** If the instrument **fails** to locate activity in five minutes, the opto-LINK will be disabled and the editing annunciators will stop flashing. The instrument will also revert back to the normal RS-232 communications (ie. the SERIAL:TYPE setting will be re-activated).

2.1.2. opto-LINK Communications Settings

The communications settings for opto-LINK are **9600 baud**, **no parity**, **8 data bits** and **1 stop bit**. This is often referred to as **9600 N81**.

2.2. RS-232 Connection

2.2.1. RS-232 Communications Settings

The communications settings for RS-232 are configured using the SERIAL:BAUD and SERIAL:BITS items in the instrument setup menus.

Network: One Instrument to PC (RXD, TXD, GND)

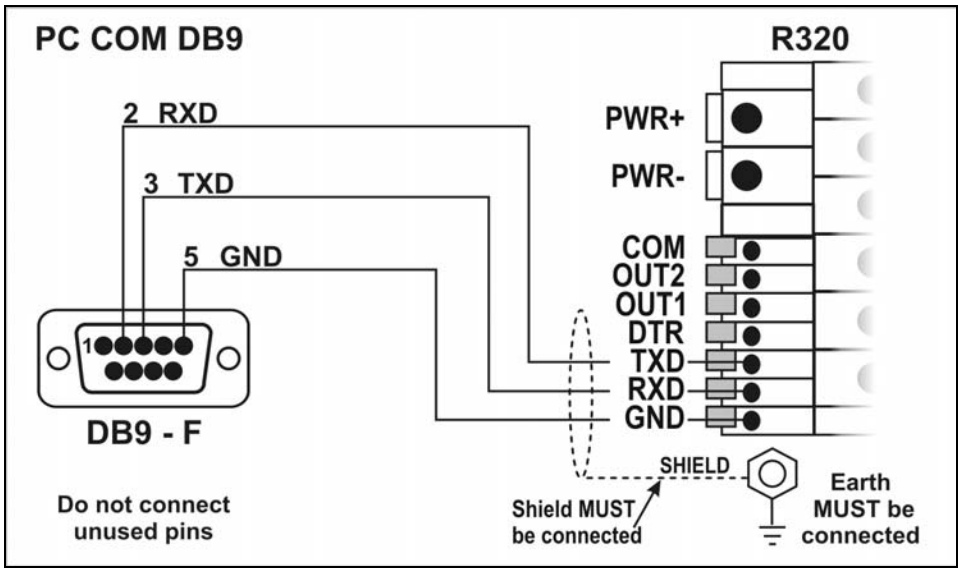


Figure 2: RS-232 — One Instrument to PC using COM Port (DB9)

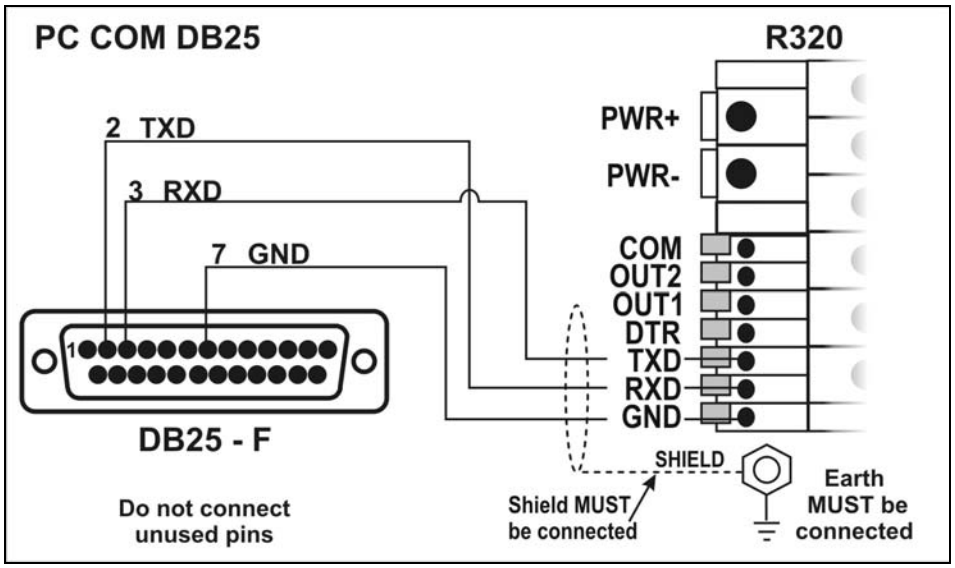


Figure 3: RS-232 – One Instrument to PC using COM Port (DB25)

Ring Networks: Multiple Instruments to PC (RXD, TXD, GND)

Instruments with K302 software revision V3.82+ can be configured in a Ring Network. This also requires an enhancement in the PC's software.

The Short Ring Network layout can be used in situations up to a total cable run length of about 150 m (500 ft) at 9600 baud in a clean EMC environment. If there are communications errors, or for longer cable runs, lower the baud rate to 4800 or 2400, and/or use the Long Ring Network in Figure 5 below, which uses a separate return path from the 'Last Instrument' to the PC.

For DB25 connections at the PC connector, refer to Figure 3 above.

When operating in a Ring Network, the Instruments must have:
- **same** serial port options, i.e., baud, parity, data bits, stop bits;
- **unique** addresses.

Short Ring Network: Multiple Instruments to PC (RXD, TXD, GND)

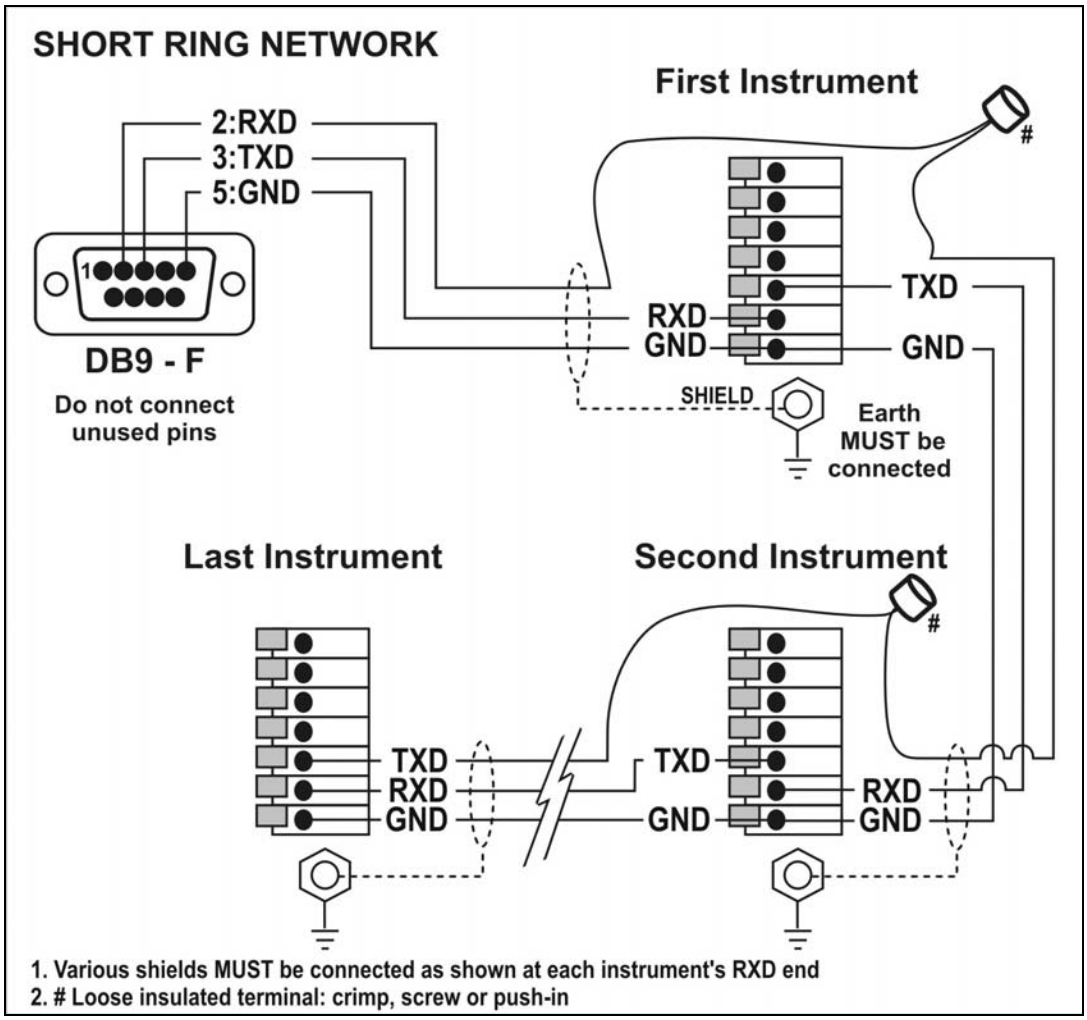


Figure 4: RS-232 Short Cable Runs: Ring Network using COM Port (DB9)

Long Ring Network: Multiple Instruments to PC (RXD, TXD, GND)

The Long Ring Network layout can be used in situations where each leg of the cable run can be up to about 150 m (500 ft) at 9600 baud. If there are communications errors, lower the baud rate to 4800 or 2400.

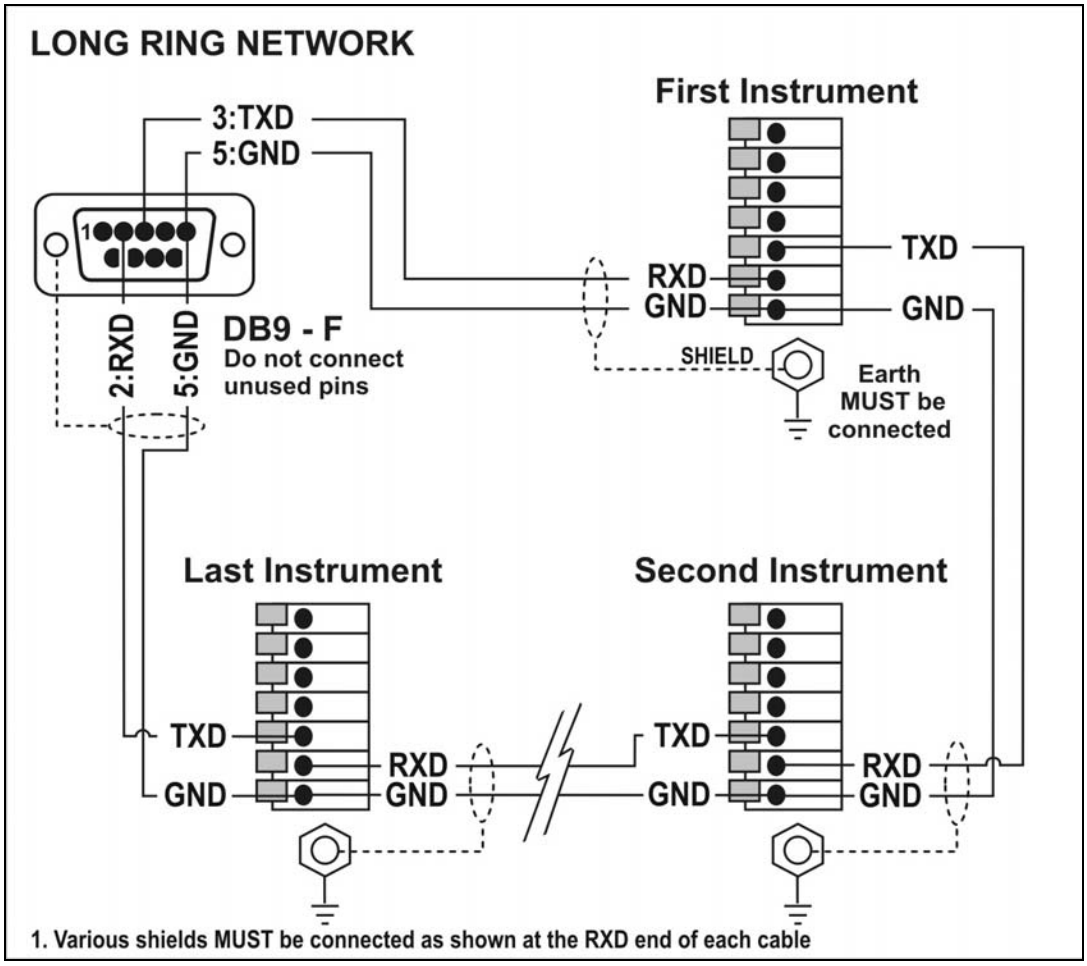


Figure 5: RS-232 Long Cable Runs: Ring Network using COM Port (DB9)

2.3. opto-LINK vs RS-232

Table 1 provides a summary of the differences between the two communications links:

	opto-LINK	RS-232
Supports **COMM** protocol	Yes	Yes
Supported indicators	All R300 Series	R320, R323 Only
Permanent connection	No	Yes
Communications settings	9600, N, 8, 1	Configurable via SERIAL:BAUD, SERIAL:BITS
Timeout with no use	5 minutes	None
Software upgrade using FLASH	Yes	Yes
Remote display (Master)	No	Yes
Printing (Print)	No	Yes
Auto outputs (Auto.1, Auto.2)	No	Yes
NET mode (Network)	When activated as described in Opto-Link Activation page 5.	SERIAL:TYPE Auto switch to NET when command received
Supports **Ring Network** enhancement	No	Yes: set SERIAL:TYPE to NET

Table 1: opto-LINK vs RS-232 port

3. Getting Started

This section gives a brief introduction to using the **COMM** protocol on the instrument. The examples below use the broadcast address and assume that a PC is connected to the RS-232 port as described in RS-232 Connection page 6. The indicator is assumed to be at address **01**.

From K302 software revision V3.82+, the instruments can be installed in a Ring Network, see §3.7 below.

3.1. Using Viewer

- Start **Viewer**.
- Connect to the indicator.
- Select the **Test** page.
- The examples listed in this section may be entered directly in the edit box at the top left of the screen.
- Press the **Send** button or press the **ENTER** key to send a command to the indicator. The command that is sent is shown on the terminal screen after the symbols **>>>**.

Note: The ↵ symbol below indicates <CR><LF>. Using **ViewR300**, this is generated by pressing **Send**.

3.2. COMM Protocol Summary

NOTE: The values used with the Read Final and Write Final commands are hexadecimal. Refer to Table 7: Decimal, Binary and Hexadecimal Conversion page 41 for information on conversions.

Figure 6 lists registers and commands for performing common tasks on the instrument.

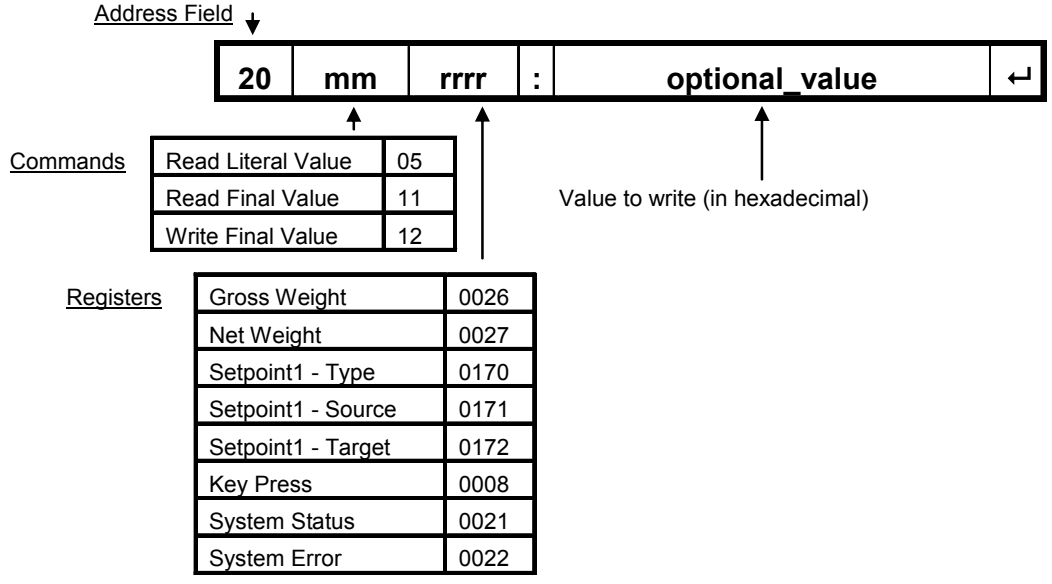


Figure 6: Commonly used Registers and Commands

3.3. Reading Gross Weight as Literal Value

Type the following (without the "" quotes) and then press **Send** (which appends <CR><LF>):

"**20050026:**"

The indicator will response with a message like:

"81050026:^^10.00^kg^G" *(along with a trailing <CR><LF>)*

3.4. Read Gross Weight as Final Value

Type the following and then press **Send**:

"**20110026:**"

The indicator will response with a message like:

"81110026:000003E8"

Where the value after the ":" is the hexadecimal value of the gross weight without any decimal places or units. In decimal, this value is 1000. For conversion between decimal and hexadecimal, refer to Table 7: Decimal, Binary and Hexadecimal Conversion page 41.

3.5. Setting Values

For example, change Set Point Target 1 (overweight) to 500 kg in final form for an indicator with units of kg.

Type the following and then press **Send**:

"**20120171:1F4**" *(1F4 in hexadecimal = 500 decimal)*

The indicator will respond with a message like:

"81120171:0000"

Where the "0000" value after the ":" indicates no errors.

3.6. Remote Key Press

For example, remotely press the **TARE** key on an indicator.

Type the following and then press **Send**:

"**20120008:8003**"

The indicator will respond with a message like:

"81120008:0000"

Note: For the **ZERO** key type "**20120008:8002**".

3.7. RS-232 Ring Network Enhancement

From K302 software revision V3.82+, the instruments can be installed in a Ring Network. The central computer's software is required to send additional framing characters, 'Echo-On' (=<DC2> =ASCII 12_H) and 'Echo-Off' (=<DC4> =ASCII 14_H) around each command.

The responses from the instruments echo the command (which is why <DC2> has been called 'Echo-On'), which is passed by one instrument on to the next.

Each instrument's response is then added to the end of the incoming message ahead of the 'Echo-Off' =<DC4> character.

3.7.1. RS-232 Ring Network Example

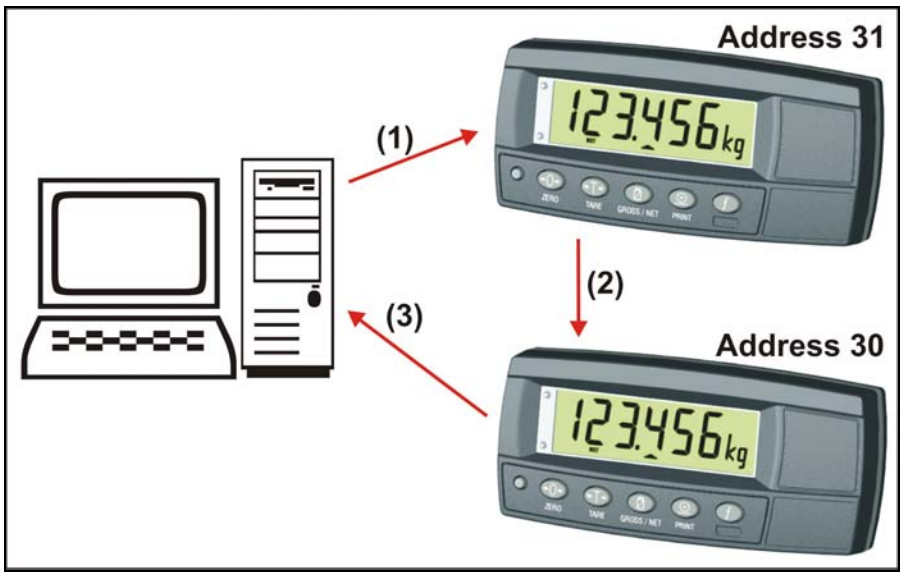


Figure 7: RS-232 Ring Network: Example with Two Instruments

A typical command from a PC, shown as (1) in the example Ring Network in Figure 7 above, might be:

Message (1):

```
<DC2>20110150:<CR><LF>
<DC4>
```

The first instrument, address 31 in our example, would send the echoed command and add its own following response on to the second instrument (2):

Message (2):

```
<DC2>20110150:<CR><LF>
9F110150:07/01/2030 17:29<CR><LF>
<DC4>
```

The second instrument, address 30 in our example, would send the echoed command and 31's response and add its own following response on to the PC (3):

Message (3):

```
<DC2>20110150:<CR><LF>
9F110150:07/01/2030 17:29<CR><LF>
9E110150:07/01/2030 17:30<CR><LF>
<DC4>
```

4. Protocol Overview

The instrument contains a number of registers. The indicator is configured by reading and writing the information stored in these registers. Information such as gross weight is obtained by reading these registers. The instrument has registers of different types to hold weights, menu options, etc.

4.1. COMM Message Structure

The messages use ASCII characters. Almost all numeric values are transmitted in hexadecimal form using uppercase HEX characters (ie. 0 to 9, A to F).

Commands from a Master (typically a PC) to a Slave (typically an Indicator) are of the form:

Address Field	Command Field	Register Id		[Optional Parameter]	
xx	**mm**	**rrrr**	":"	[ppppp]	↵

Table 2 COMM Command Format

Responses from Slave to Master are of the form:

Address Field	Command Field	Register Id		Return value	
yy	mm	rrrr	":"	vvvvv	↵

Table 3 COMM Response Format

Where:

xx is the Address Field sent to the indicator (Response bit = 0).

yy is the Address Field in the response from the indicator (Response bit = 1).

mm is the Command Field.

rrrr is the Register Id Field.

"**:**" is the separator between the command header and the parameter or return value.

ppppp is the optional parameter value sent to the indicator.

vvvvv is the return value from the indicator.

↵is the two characters <CR><LF> (Carriage-Return, Line-Feed).

4.1.1. Address Field

The address field is a two character hexadecimal field where:

Bit	Field Name	Description
7	Response	Set by the slave to indicate that this message is a response to a previous command.
6	Error	Set to indicate that the data in this message is an error code and not a normal response. Refer to Error Handling page 16.
5	Reply Required	Set by the master to indicate that a reply to this message is required by any slave to which it is addressed. If not set, the slave should silently perform the command.
4 to 0	Indicator Address	Valid unit addresses are 01_H to $1F_H$ (1 to 31 in decimal). 00_H is the broadcast address. All slaves must process broadcast commands. When replying to a broadcast, slaves replace the broadcast address with their own address in this field. Set in the indicator menus with SERIAL:ADDRES.

4.1.2. Command Field

The command field is a two character, hexadecimal field holding the id of command to be performed. The following commands are defined to read register values or properties:

Command	Description
CMD_READ_TYPE	Read Type of register.
CMD_READ_RANGE_MIN	Read Minimum valid register value.
CMD_READ_RANGE_MAX	Read Maximum valid register value.
CMD_READ_LITERAL	Read Literal value of register. Note: Value will be converted to final value as required (eg. 2.000 kg G for gross weight).
CMD_READ_DEFAULT	Read Raw default register value.
CMD_READ_MENU_TEXT	Read Menu Text name shown on indicator display during setup menus.
CMD_READ_ITEM	Return item from list. Only valid for TYPE_MENU, TYPE_OPTION and TYPE_BITFIELD.
CMD_READ_PERMISSION	Read Permission string.
CMD_READ_FINAL	Read Value of register converted to final value.

The following command is defined to write values to registers:

Command	Description
CMD_WRITE_FINAL	Write Final Value to register.

The following command is defined to execute on registers of TYPE_EXECUTE:

Command	Description
CMD_EXECUTE	Execute the function associated with a register.

4.1.3. Register Id Field

This is the four character hexadecimal register identifier. For the list of registers in the instrument refer to Register Identifiers page 37.

4.1.4. Optional Parameters

The value here is dependent on the command and register being accessed. Not all commands require a parameter (eg. CMD_READ_FINAL). Commands that write to a register have the new value as the parameter.

4.1.5. Return Value

The value here is dependent on the command and register being accessed. If a command does not return a value (eg. CMD_WRITE_FINAL), then the return value will be an error code (eg. "0000" for no errors).

4.1.6. RS-232 Ring Network Enhancement

When the instruments are installed in a Ring Network, the central computer's software is required to send additional framing characters around each command.

Referring to Table 2 COMM Command Format above, the enhanced commands from a Master (typically a PC) to the instruments are of the format:

Echo-On	COMM Message					Echo-Off
<DC2>	Command:					
	Address Field	Command Field	Register Id		[Optional Parameter]	
	xx	mm	rrrr	":"	[ppppp]	↵
						<DC4>

Table 4 COMM Command Format with Ring Network Enhancement

Where:

<DC2> and **<DC4>** are the characters ASCII 12_H and ASCII 14_H respectively, here called 'Echo-On' and 'Echo-Off'.

Each instrument echoes the command, which is passed on to the next instrument.

If the command's Address Field is addressed to one of the instruments, or is a broadcast (Address Field = 0), then the instrument will append a response ahead of the 'Echo-Off'-<DC4> character.

If the command is broadcast, every instrument will append a response: eg., 7 instruments will generate 7 responses.

Referring to Table 3 COMM Response Format above, the responses from each instrument that are sent onto the next instrument, and so on up to the PC, are of the format:

Echo-On	COMM Message(s)						Echo-Off
<DC2>	Command:						
	Address Field	Command Field	Register Id		[Optional Parameter]		
	xx	mm	rrrr	":"	[ppppp]	↵	
	'n' Responses:						
	Address Field	Command Field	Register Id		Return value		
	yy	mm	rrrr	":"	vvvvv	↵	<DC4>

Table 5 COMM Response Format with Ring Network Enhancement

4.2. Error Handling

If a command cannot be processed, the indicator returns an error code. Typical errors include:

- Register not implemented on indicator
- Register value out of range
- Command not implemented for register

An error response is indicated by setting the Error bit in the Address Field of the reply.

The message returned by the indicator in response to an error will be of the form:

Address Field	Command Field	Register Id		Error Code	
zz	mm	rrrr	":"	wwwww	↵

An error code will consist of a hexadecimal value representing the specific error code.

Note: When an error occurs, the most significant bit is **1** and other bits indicate the specific problem. More than one error bit may set if more than one error is present.

Refer to Error Codes page 37.

For example, attempting to read the type of the register with Id = 0000_H (not implemented) will return:

```
C1010000:A000
```

which is interpreted as:

$C1_H$ = 1100 0001 $_b$

- Response to a command
- Error
- Unit responding is address 01_H

01_H = REG_READ_TYPE (the command that was sent).

0000_H = Register Id.

$A000_H$ = 1010 0000 0000 0000 $_b$ is the error code which is interpreted as

- REG_ERR_ERROR = 8000_H (*most significant bit set)*
- REG_NOT_IMPLEMENTED = 4000_H

4.3. Register Types

The following types are defined:

Type	Data Size (bytes)	Minimum Value	Maximum Value	Description
TYPE_BYTE	1	0	2^8-1	Small numeric values.
TYPE_OPTION	1	0	2^8-1	Lists of strings.
TYPE_USHORT	2	0	2^{16}-1	Unsigned numeric values.
TYPE_SHORT	2	-2^{15}	2^{15}-1	Signed numeric values.
TYPE_LONG	4	-2^{31}	2^{31}-1	Large numeric values.
TYPE_ULONG	4	0	2^{32}-1	Large unsigned numeric values.
TYPE_STRING	--	--	--	<NUL> terminated string.
TYPE_BITFIELD	1, 2 or 4	0	2^8-1, 2^{16}-1, or 2^{32}-1	Composite setting.
TYPE_MENU	1	0	2^8-1	Menu structure.
TYPE_BLOB	--	--	--	Binary Large OBject. A block of data, interpretation is inferred from Register Id.

4.4. Permissions

4.4.1. Register Access

Each register has permission levels required for reading, writing and executing. These permission levels can be read using the CMD_READ_PERMISSION command. The communications link has a permission level. Changing the current permission level is achieved by writing a passcode to a permission register. If the correct passcode has been given, future messages use that permission level. The passcodes for SAFE and FULL are also used for SPEC:SAFE.PC and SPEC:FULL.PC in the indicator's setup menus. If more than one passcode register has the correct passcode, the indicator uses the highest level (most powerful). The instrument has the following permission levels:

Permission Level	Description	Passcode Register
None	Lowest level permission. Operation is always available (eg. reading the gross weight).	NONE
Safe	Operation is available if the safe passcode has been given. Used for indicator settings that do not affect trade certification of indicator (eg. baud rate).	REG_ENTER_PASS_SAFE
Full	Operation is available if the FULL passcode has been given. Used for trade critical indicator settings (eg. decimal places).	REG_ENTER_PASS_FULL
Factory	Operation is available only to the indicator's internal operations (eg. writing to the gross weight).	N/A

If a command cannot be performed because the register needs a higher permission, the command will return an error code of REG_ERR_ACCESS_DENIED.

4.4.2. Calibration and Configuration Counters

Within the instrument there are a number of trade critical registers that can affect the calibration and/or legal for trade performance of the instrument. If any of these registers are altered, the trade certification of the scale will be voided.

The instrument provides built-in calibration counters to monitor the number of times the critical registers are altered. The value of the counter is stored within the unit and can only be reset at the factory. Each time a critical register is altered, the counter will increase. Whenever the instrument is powered up, or setup mode is entered/exited, the current value in the counter is displayed briefly (eg. C00010).

Note: When the Scale Use is set to NTEP two counters will display. The table below describes when the counter(s) will increment for Industrial, OIML or NTEP modes.

Industrial	OIML	NTEP
The **Calibration Counter** increments when trade critical settings, marked with ⊗**,** are changed. An example of the counter is **C.00019**.	The **Calibration Counter** increments when trade critical settings, marked with ⊗**,** are changed. An example of the counter is **C.00019**	The **Calibration Counter** increments when trade critical settings in the Calibration (**CAL**) menu, marked with ⊗**,** are changed. An example of the counter is **C.00010**. The **Configuration Counter** increments when other trade critical settings (ie. **not** in the **CAL** menu), marked with ⊗**,** are changed. An example of the counter is **F.00009**.

Internally, the instrument operates with two counters - the Calibration Counter (REG_CAL_COUNT_NTEP) and the Configuration Counter (REG_CFG_COUNT_NTEP). These correspond directly to the NTEP counters. The Industrial/OIML Calibration Counter (REG_CAL_COUNT_OIML) is the sum of these NTEP counters. Each register has a bit in the Permission property to indicate if a counter will be incremented when the register's value is modified. This counter is incremented even if the values are not saved to EEPROM.

The Calibration counter is incremented for trade critical registers that affect calibration parameters. The Configuration counter is incremented for trade critical registers that affect configuration parameters.

The value(s) of the counter(s) is written on the tamperproof trade label on the front of the indicator for trade-certified applications and functions as an electronic seal. If any legal for trade settings are changed on the instrument, the current value of the calibration counter will be different from the recorded value and the seal is broken. In this manual, items marked with ⊗ indicate that the setting is a legal for trade critical settings.

5. Register Operations and Properties

5.1. Register Operations

The basic operations that can be performed on values in registers are:

- Read the current value of the register as Final or Literal.
- Write a new value to the register as Final.
- Execute a function associated with a register.

If a command fails, the returned message will have the Error bit set in the Address field to show that the return value is an error code. Not all indicators implement all registers. If an indicator does not support a register or a particular command on a register, it will return an error code of REG_ERR_NOT_IMPLEMENTED. If an attempt is made to write to a register while the indicator is in the setup menus, the indicator will return REG_ERR_MENU_IN_USE. If an attempt is made to access a register without sufficient permission (refer to Register Access page 18), the indicator will return REG_ERR_ACCESS_DENIED.

5.2. Common Properties

Each register stores information about itself. This information is referred to as properties. The register properties differ depending on the type of the register.

Registers implement the following common properties, regardless of the register's type.

5.2.1. Type

Each indicator supports a finite set of register types. It is recommended that this command be performed on a register (to check that it is supported) before performing any other operation.

Command:	CMD_READ_TYPE
Return:	Register Type as a hexadecimal value eg. 01 => TYPE_BYTE
Error Code:	REG_ERR_NOT_IMPLEMENTED

5.2.2. Menu Text

This property is the text shown on the indicator's display while traversing the menus.

Command:	CMD_READ_MENU_NAME
Return:	The indicator will return a string. Some registers may return an empty string in response to this command. The text may contain leading or trailing spaces.

eg. The Menu Text for REG_DECIMAL_PLACES is "DP" and when accessed from the **BUILD** menu the display shows **DP**.

5.2.3. Permission

Command: CMD_READ_PERMISSION
Return: This property returns a string of characters, indicating the permission bits for the register. The characters are (in order):

- Read Permission
- Write Permission
- Calibration Counter
- Configuration Counter

Read	Write	Calibration Counter	Configuration Counter
"-" None (any)	"-" None (any)	"-" None	"-" None
"S" Safe	"S" Safe	"C" Affects	"F" Affects
"F" Full	"F" Full		
"f" Factory (Internal)	"f" Factory (Internal)		

eg. The Permission property of REG_DECIMAL_PLACES returns "-F-F" which means:

- Read with any permission (always available).
- Write with Full permission.
- Does not affect Calibration Counter.
- Affects Configuration Counter.

5.3. Type Specific Properties

The behaviour of the following properties depends on the type of register.

5.3.1. RangeMin, RangeMax

All Registers have a concept of RangeMin and RangeMax. The register must contain a value that is in the range RangeMin <= Value <= RangeMax.

Command: CMD_READ_RANGE_MIN
Return: Hexadecimal

Register Type	RangeMin	RangeMax
TYPE_SHORT, TYPE_USHORT, TYPE_LONG, TYPE_ULONG	Minimum value	Maximum value
TYPE_STRING, TYPE_BLOB	0	(No. of elements -1)
TYPE_OPTION, TYPE_MENU	0	(No. of Elements-1)
TYPE_BITFIELD	0	(No. of character positions – 1)

5.3.2. Final Value

The final value is the numeric value converted to user units (eg. weight in kg), but without unit or decimal places.

Note: These values are generally the same as the displayed value with the decimal point and units removed.

Command: CMD_READ_FINAL
Return: Hexadecimal value of register without units or decimal points.
Errors: REG_ERR_ACCESS_DENIED if permission not valid.

Command: CMD_WRITE_FINAL
Parameter: hexadecimal value to write to register without units or decimal points.
Errors: REG_ERR_UNDER_RANGE if new value < RangeMin.
REG_ERR_OVER_RANGE if new value > RangeMax.
REG_ERR_ACCESS_DENIED if permission not valid.
REG_ERR_MENU_IN_USE if menu in use and attempting to write.

eg. An indicator displaying 2.345 kg Gross would return "929" when reading REG_WEIGHT_GROSS. Converting 929_H to decimal gives 2345.

5.3.3. Literal Value

This is a formatted version of the Final value in a human readable form. It may be the same as the Final Value.

Command: CMD_READ_LITERAL

eg. The Literal value of a TYPE_OPTION register is the string to show the current value.

5.3.4. Default Value

Registers can have a factory default value associated with them.

Command: CMD_READ_DEFAULT

5.3.5. Execute

This command has optional parameters that are specific to each register.

Command: CMD_EXECUTE
Return: Dependent on register being executed on.

For more information, refer to Execute Functions page 28.

5.3.6. Read Items

This command reads one item from the list for the register. The item to be read is given as a parameter to the command. Each value is returned as a string.

Command:	CMD_READ_ITEM
Valid Types:	TYPE_OPTION, TYPE_MENU, TYPE_BITFIELD
Parameter:	RangeMin ≤ parameter ≤ RangeMax
Return:	String value
Error Code:	If no parameter is present, the indicator returns ERR_BAD_PARAMETER.

eg. Read items 0 and 1 from REG_DECIMAL_PLACES

Read item 0	**200D0128:0↵**
Response	810D0128:000000↵
Read item 1	**200D0128:1↵**
Response	810D0128:00000.0↵

6. Reading Status and Error Codes

6.1. REG_SYSTEM_STATUS

This register contains status information about the indicator.

Bit(s)	Field	Description
31 to 18, 8, 5, 4	RESERVED	Reserved for future use.
17	OverLoad	The weight is above the maximum allowable weight reading.
16	UnderLoad	The weight is below the minimum allowable weight reading.
15	Error	Diagnostic error. Check REG_SYSTEM_ERROR.
14	MenuActive	Setup menus are active. Registers cannot be written to using **COMM**.
13	Calibrating	Unit is busy calibrating.
12	Motion	Weight is not stable.
11	isCOZ	The gross reading is within ± ¼ of a division of true zero.
10	isZero	The displayed weight is within the zero 'dead' band setting.
9	DisplayNet	Display is showing NET value.
7	Output1	The setpoint weight is over the setpoint target.
6	Output2	The setpoint weight is under the setpoint target.
3 .. 0	InternalErrorCode	Used for Calibration Errors, etc.

Table 6: Register REG_SYSTEM_STATUS Interpretation

The following table lists the values that the InternaErrorCode field may take. It lists the error message that would be displayed on the indicator's display if the operation was being performed using the setup menus.

Display Message	Code Value	Description
no error	00	Last operation was successful.
(SPAN) (LO)	01	The load cell signal range (span) is too small for these settings.
(SPAN) (HI)	02	The load cell signal range (span) is too large for these settings.
(RES) (LO)	03	The scale build is configured for less than 100 graduations.
(RES) (HIGH)	04	The scale build is configured for more than 30,000 graduations.
(PT.TOO) (CLOSE)	05	An attempt has been made to place a calibration point too close to an existing calibration point.
	06	The calibration (linearisation) point does not exist.
(LIN.PT) (LO)	07	An attempt has been made to place a linearisation point below zero.
(LIN.PT) (HI)	08	An attempt has been made to place a linearisation point above fullscale.
(RES) (LO)		The scale build is configured for less than 100 graduations.

Display Message	Code Value	Description
(RES) (HIGH)		The scale build is configured for more than 30,000 graduations.
(ZERO) (LO)		An attempt has been made to calibrate zero below -2mV/V.
(ZERO) (HI)		An attempt has been made to calibrate zero above +2mV/V.

6.2. REG_SYSTEM_ERROR

This register contains the error codes as displayed on the indicator's LCD display.

The instrument continually monitors the condition of the internal circuits. Any faults or out-of-tolerance conditions are shown on the display as an **E** type error message.

In the table below the following terms are used:

- Check: This item can be checked on site by service personnel.
- Return for Service: The instrument must be returned to the manufacturer for factory service.

Error	Description	Resolution
(E0001)	The power supply voltage is too low.	Check supply
(E0002)	The power supply voltage is too high.	Check scale / cables
(E0010)	The temperature is outside of allowable limits.	Check location
(E0020)	Scale build is incorrect. The number of graduations has been set too low or too high.	Fix up scale build
(E0100)	The digital setup information has been lost.	Re-enter setup
(E0200)	The calibration information has been lost.	Re-calibrate
(E0300)	All setup information has been lost	Enter setup and calibrate
(E0400)	The factory information has been lost.	Return for Service
(E0800)	The EEPROM memory storage chip has failed	Return for Service
(E2000)	ADC Out of Range Error. This may be caused from a broken load cell cable.	Check BUILD:CABLE setting. Check load cell cable, wiring, etc.
(E4000)	The battery backed RAM data has lost data.	Re-enter setup
(E8000)	The FLASH program memory is incorrect	Return for Service

The **E** type error messages are additive. For example if instrument is running off batteries and the temperature drops, the battery voltage may be too low. The resulting error messages will be **E 0011** (0001 + 0010). The numbers add in hexadecimal as follows:

1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - A - B - C - D - E - F

(For example, 2 + 4 = 6, or 4 + 8 = C)

7. Remote Key Interface

7.1. Operation

The indicator polls the REG_KEYBOARD register looking for new key presses to act upon. No new key press is indicated by a value of KEY_NONE in this register. After reading the value, the unit sets REG_KEYBOARD to its default value (KEY_NONE). Remote control of the indicator is achieved by writing to this register with the new key-code. Priority is given to the local keyboard and the external key-code is lost.

7.2. Key-Codes

Key-codes are divided into physical, logical and ASCII codes and are represented by an unsigned 16-bit value:

Code Type	RangeMin	RangeMax	Notes
ASCII	0000_H	$007F_H$	Normal ASCII characters including <NUL> 00_H
Reserved	**0080_H**	**$6FFF_H$**	**Reserved for future use.**
Logical	7000_H	$7FFF_H$	
Physical	8000_H	$FFFF_H$	

Physical codes represent the physical keys on an indicator. Logical codes represent an indicator function. ASCII codes are used to enter data using device independent codes (eg. a numeric keypad could generate ASCII codes). Logical characters can be considered to include the ASCII characters as a subset. Refer to Key Codes page 40 for the list of codes.

7.2.1. Physical Codes

Physical keys are numbered from 1 to 63.

Physical codes are interpreted by the indicator in a **device dependent manner**.

Physical keys consist of the KEY_PHYSICAL bit **ORed** with the numeric value of the key (1 to 63).
eg. (KEY_PHYSICAL + 1) = 1st physical key, . (KEY_PHYSICAL + 5) = 5th physical key.

A long press is indicated by the KEY_PHYSICAL_LONG bit being set. If not set, it is a short press.

To handle two simultaneous physical keys being pressed (to increase the number of keys available), one key number is **ORed** into the upper byte of the keycode and the other into the lower byte. The lower numbered key is shifted into the MSByte.

eg. Physical key 1 and Physical key 2: `(KEY_PHYSICAL | (1 << 8) | (2))`

7.2.2. Example

Tare an indicator by remotely sending the **TARE** key

Command Field	12_H	*CMD_WRITE_FINAL*
Parameter Field	8003_H	*KEY_PHYSICAL_3*
Register Field	0008_H	*REG_KEYBOARD*
Complete Message	20120008:8003↵	
Sample Response	81120008:0000↵	

7.2.3. Logical Keys

Logical codes maintain their meaning across a range of indicators.

eg. KEY_SETUP_FULL

8. Execute Functions

Registers that respond to CMD_EXECUTE may use the optional parameter field depending on the function involved. Execute functions are used to perform actions (eg. calibration) within the indicator, as opposed to register values that can simply be read or written (eg. changing the full-scale value).

8.1. Calibration

Calibration functions may take a number of seconds to complete. During this period the indicator will display a message to show that it is calibrating. When the calibration is complete, the indicator's display will return to displaying the current weight. To determine the progress of the calibration, read REG_SYSTEM_STATUS until the Calibrating bit is 0 (refer to REG_SYSTEM_STATUS page 24). Then check the InternalErrorCode field. A value of 0 indicates no errors with the last calibration. Non-zero values indicate various errors such as span high or low, etc.

Zero calibration is performed using REG_CALIBRATE_ZERO as shown in the table below.

Type of Calibration	Parameter	Other Registers
Zero Calibration using Test Weights	none	none
Direct mV/V Zero Calibration	(mV/V * 10,000) as hexadecimal value	none

eg. Execute a Zero calibration using test weights.

Command = 10_H *CMD_EXECUTE*

Register Id = 0102_H *REG_CALIBRATE_ZERO*

Parameter Field = "" *none*

Remove all weight from scale		
Complete Message	**20100102:**↵	*Indicator shows "Z. in P"*
Sample Response	81100102:0000↵	
Check Status by reading REG_SYSTEM_STATUS		
Complete Message	**20040021:**↵	
Sample Response while calibrating	81040021:00002000↵	*Indicator is Calibrating*
Sample Response when complete	81040021:00000C00↵	*Calibration finished, no errors*

The return value of $0C00_H$ from REG_SYSTEM_STATUS is described in REG_SYSTEM_STATUS page 24. In binary this value is $0000\ 1100\ 0000\ 0000_b$ which has isCOZ=1 and isZero=1 showing that the indicator has been zeroed. InternalErrorCode field = 0 showing that the calibration was successful.

8.1.1. Span

Span calibration is performed using REG_CALIBRATE_SPAN as shown in the table below.

Type of Calibration	Parameter	Other Registers
Span Calibration using test weights	none	REG_WEIGHT_CALIBRATION
Direct mV/V Span Calibration	(mV/V * 10,000) as hexadecimal value	REG_FULLSCALE

eg. Execute a Span calibration using a scale base of 2500 g. Assume that the indicator has Units = g, Decimal Places = "000000".

Command = 10_H *CMD_EXECUTE*

Register Id = 0103_H *REG_CALIBRATE_SPAN*

Set span weight	**20120100:9C4**	*REG_WEIGHT_CALIBRATION*
Add test weights		
Execute	**20100103:**↵	*REG_CALIBRATE_SPAN*
Sample Response	81100103:0000↵	*Indicator shows "S. in P"*
Check Status by reading	**20040021:**↵	
Sample Response while calibrating	81040021:00002000↵	*Indicator is Calibrating*
Sample Response when complete	81040021:00000000↵	*Calibration finished, no errors*

InternalErrorCode field = 0 showing that the calibration was successful.

eg. Execute a Direct Span calibration at 3.0 mV/V. Assume that REG_FULLSCALE already contains 3000 decimal.

Command = 10_H *CMD_EXECUTE*

Register Id = 0103_H *REG_CALIBRATE_SPAN*

Parameter Field = 7530_H *(3.0 * 10000 = 30000 decimal)*

Execute with direct mV/V	**20100103:7530**↵	*REG_CALIBRATE_SPAN*
Sample Response	81100103:0000↵	
Check Status	**20040021:**↵	*REG_SYSTEM_STATUS*
Sample Response	81040021:00000C00↵	

InternalErrorCode field = 0 showing that the calibration was successful.

8.1.2. Linearisation (K302 and K305 only)

Adding linearisation points is similar to span calibration using test weights.

The desired weight value is written to REG_WEIGHT_CALIBRATION. To delete a point, set this value to zero.

Place the test weight on the scale.

Execute REG_CALIBRATE_LIN**x** to add the point "x" to the calibration points (1 ≤ x ≤ 5).

Check REG_SYSTEM_STATUS for any errors.

eg. Assume an indicator that has been calibrated for zero and span at 3000 g. Add a linearisation point L1 at 1000g.

Write calibration weight	**20120100:3E8**↵	*REG_WEIGHT_CALIBRATION*
	81060100:0000↵	
Add calibration weight		
Execute linearisation L1	**20040021:**↵	*REG_CALIBRATION_LIN1*
	81100104:0000↵	*Unit shows "L. in P"*
Check status	**20040021:**↵	
	81040021:00002000↵	*Busy Calibrating*
	20040021:↵	
	81040021:00000000↵	*Completed successfully*

eg. Deleting Linearisation point L1.

Write calibration weight	**20120100:0**↵	*REG_WEIGHT_CALIBRATION*
	81060100:0000↵	
Execute linearisation L1	**20040021:**↵	*REG_CALIBRATION_LIN1*
	81100104:0000↵	
Check status	**20040021:**↵	
	81040021:00000000↵	*Completed successfully*

8.2. Save Settings to EEPROM

When an indicator is switched on, it retrieves the last saved settings from EEPROM. When settings are modified by writing to or executing a register, the value is not automatically saved to EEPROM. If the unit is switched off or power is interrupted, the new setting will be lost. Executing REG_SAVE_SETTINGS will save the current settings to EEPROM.

eg. Save settings to EEPROM.

Save Settings	**20100010:**↵	*REG_SAVE_SETTINGS*
	81100010:0000↵	

9. Streaming

9.1. Basic Operation

The streaming facility in the instrument allows a master (PC) to quickly read a number of registers (up to three) from a slave (indicator) with one command. REG_STREAM_DATA contains the **final** value for the three registers. When new weight data is available, REG_STREAM_DATA is updated with the latest **final** values pointed to by REG_STREAM_REG1 to 3.

There is a list of registers that may be streamed. REG_STREAM_REG1, REG_STREAM_REG2 and REG_STREAM_REG3 are used to select three registers from this list. The raw value of each REG_STREAM_REG1 to 3 is an index into this list.

9.2. Registers Available to Stream

The following table lists the registers able to be streamed:

Index into list	Reg Id
0000_H	REG_NONE
0001_H	REG_ADC_SAMPLE_NUMBER
0002_H	REG_SYSTEM_STATUS
0003_H	REG_SYSTEM_ERROR
0004_H	REG_ABSOLUTE_MVV
0005_H	REG_WEIGHT_DISPLAY
0006_H	REG_WEIGHT_USER
0007_H	REG_WEIGHT_GROSS
0008_H	REG_WEIGHT_NET
0009_H	REG_WEIGHT_TARE
$000A_H$	REG_WEIGHT_PEAK
$000B_H$	REG_WEIGHT_HOLD
$000C_H$	REG_WEIGHT_TOTAL
$000D_H$	REG_WEIGHT_LIVESTOCK
$000E_H$	REG_WEIGHT_PT_TARE
$000F_H$	REG_FULLSCALE

9.3. Example

Figure 8 below, demonstrates the use of the streaming registers. In this example, REG_STREAM_REG1 contains the value 0003_H. Looking up this index into the list of registers yields REG_SYSTEM_ERROR. So the 1st position in REG_STREAM_DATA is filled in with the contents of REG_SYSTEM_ERROR. Similarly, REG_STREAM_REG2 contains 0004_H, which refers to REG_ABSOLUTE_MVV. So the 2nd position in REG_STREAM_DATA is filled in with the contents of REG_ABSOLUTE_MVV. The 3rd position in REG_STREAM_DATA is filled in with the contents of REG_ADC_SAMPLE_NUMBER.

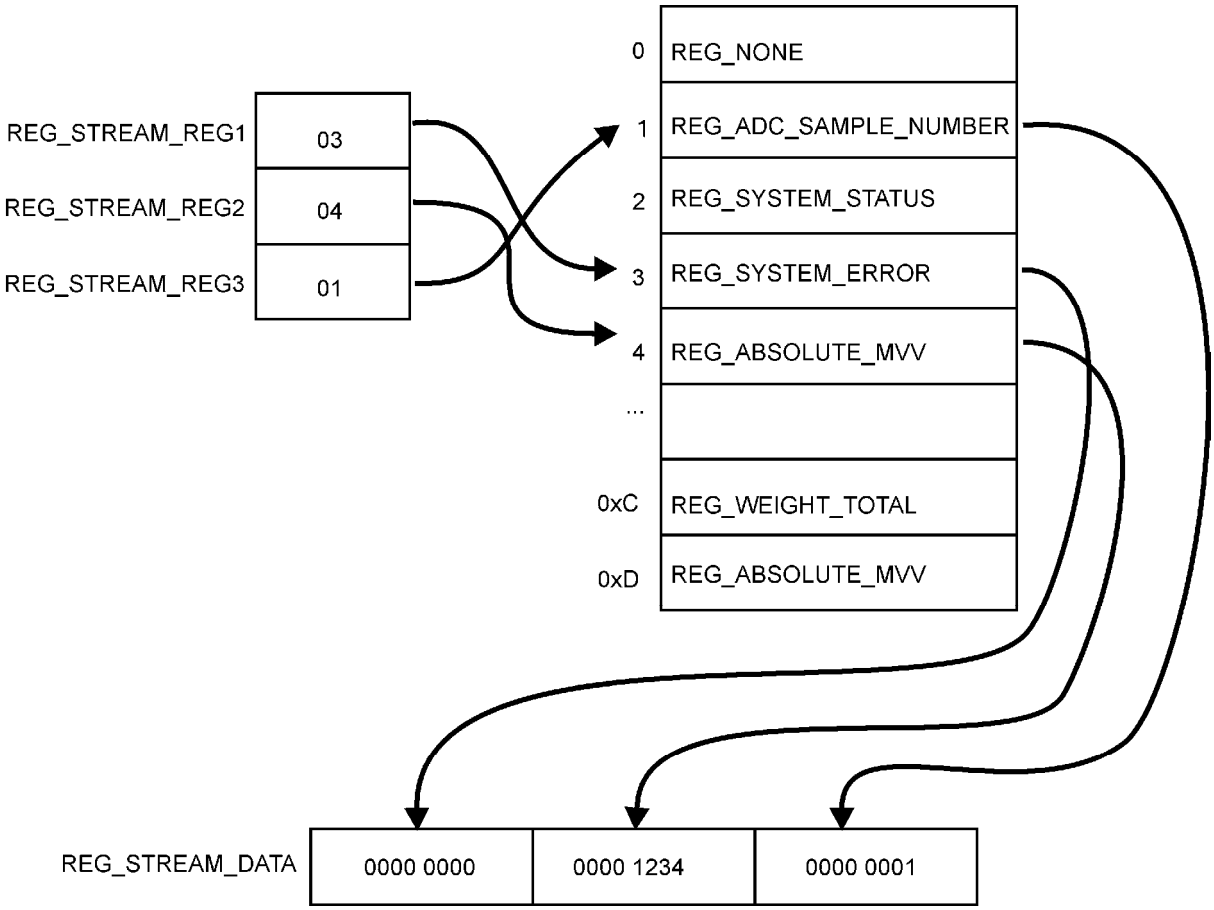


Figure 8: Streaming Registers Example

Read Stream Register	**20110040:**↵
Final Response	81110040:00000000000123400000001↵
Read Stream Register Literal	**20050040:**↵
Response	81050040:E0000,0.4660,1↵

10. Printing

10.1. Basic Operation

There are two non-volatile registers available for user programmable print strings. REG_PROG_PRINT with a capacity 80 ascii(160 hex) characters including tokens. REG_PROG_PRINT_SUMMARY with a capacity of 20 ascii(40 hex) characters including tokens. The REG_PROG_PRINT register is used if SERIAL:FORMAT:CUSTOM is selected.when prints are activated for SERIAL:TYPE settings of AUTO, SINGLE and AUTO.PR.

When SERIAL:TYPE:PRINT is selected a short press of the print key uses REG_PROG_PRINT as the source for printing while the REG_PROG_PRINT_SUMMARY register is used when prints are activated by a long press of the print key.

FACTRY:DEFLT resets REG_PROG_PRINT to the standard short print press printing, while REG_PROG_PRINT_SUMMARY is reset to the standard long print press printing. Alternatively the standard formats can be set by writing the default print tokens (indicated below) to the registers REG_PROG_PRINT and REG_PROG_PRINT_SUMMARY respectively.

10.2. Tokens Available for Printing

The following table lists the tokens available for printing:

Token	Corresponding Print Feature
$7F_H$	Default Print String (short print press)
$8F_H$	Default Print String (long print press)
CF_H	Gross Weight
$D0_H$	Net Weight
$D8_H$	Tare_Weight
$D9_H$	Display Weight
$F0_H$	Count (if counting is turned on)
$C0_H$	Time
BF_H	Date
DA_H	Print Identifier
$E8_H$	Streaming Register Literal
$E9_H$	Streaming Register Raw
DB_H	Total Weight (total of Display Weight)
DD_H	Total Count (if counting is turned on)
DC_H	Items (No. of prints since last summary)

10.3. Examples of Operation

The following data loaded :

```
20120146:D020617420C0206F6E20BF0D0A↵
```

Would produce the printout below:

```
525 kg N at 11:04 on 03.02.2005<CR><LF>
```

To use the streaming tokens, the streaming register is first set up as described above.

Streaming to print the REG_SYSTEM_ERROR, REG_ABSOLUTE_MVV, and REG_ADC_SAMPLE_NUMBER registers:

Write Stream Register 1 **20120042:03**↵
Write Stream Register 2 **20120043:04**↵
Write Stream Register 3 **20120044:01**↵

The “Streaming Register Raw” print token when loaded :

```
20120146:E9↵
```

Would produce the printout below on a short press of the print button, auto, single print, or auto print options:

```
00000000000138700024AED<CR><LF>
```

The “Streaming Register Literal” print token when loaded :

```
20120146:E8↵
```

Would produce the printout below on a short press of the print button, auto, single print, or auto print options:

```
E0000,0.4999,148113<CR><LF>
```

11. Advanced Type Handling (TYPE_MENU)

The menu items can be accessed from the **COMM** port. Figure 9 illustrates examples of the menu levels.

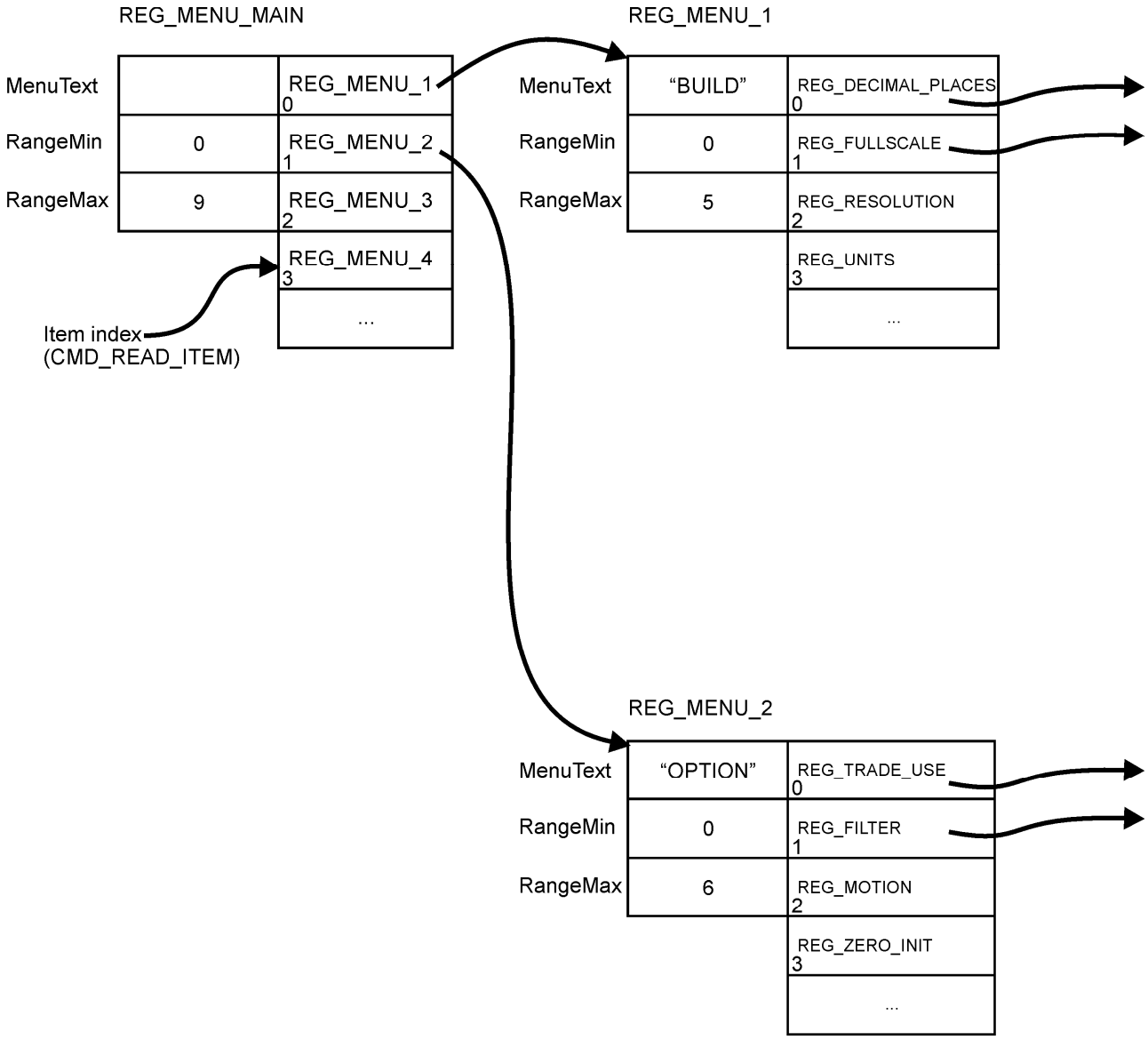


Figure 9: TYPE_MENU Example

12. Appendix – Program Constants

12.1. Register Types

Type	Code	Description
TYPE_CHAR	00_H	8-bit signed value (typically used for ASCII characters)
TYPE_UCHAR	01_H	8-bit unsigned value
TYPE_SHORT	02_H	Signed 16-bit value
TYPE_USHORT	03_H	Unsigned 16-bit value
TYPE_LONG	04_H	Signed 32-bit value
TYPE_ULONG	05_H	Unsigned 32-bit value
TYPE_STRING	06_H	NULL terminated string of CHAR
TYPE_OPTION	07_H	Selectable option.
TYPE_MENU	08_H	Menu type
TYPE_WEIGHT	09_H	Contains weight and status information
TYPE_BLOB	$0A_H$	Untyped block of memory
TYPE_EXECUTE	$0B_H$	Executable commands
TYPE_BITFIELD	$0C_H$	Options combined into a compressed field

12.2. Commands

Command	Code	Description
CMD_NONE	00_H	
CMD_READ_TYPE	01_H	Read the type of a register
CMD_READ_RANGE_MIN	02_H	Read the minimum value allowed for this register
CMD_READ_RANGE_MAX	03_H	Read the maximum value allowed for this register
CMD_READ_RAW	04_H	Read internal raw value (typically for factory use only)
CMD_READ_LITERAL	05_H	Read literal value.
CMD_WRITE_RAW	06_H	Write internal raw value (typically for factory use only)
CMD_READ_DEFAULT	07_H	Read default value
CMD_READ_MENU_TEXT	09_H	Read Menu Text as used in the instrument menus
CMD_READ_ITEM	$0D_H$	Read Item from Option list
CMD_READ_PERMISSION	$0F_H$	Read register permission settings
CMD_EXECUTE	10_H	Execute a command on a TYPE_EXECUTE register
CMD_READ_FINAL	11_H	Read final register value
CMD_WRITE_FINAL	12_H	Write final register value

12.3. Error Codes

The error code returned when a command is not processed correctly is a hexadecimal value that is a combination of the following constants.

Error	Code	Description
REG_ERR_NONE	0000_H	
REG_ERR_ERROR	8000_H	Always set to show an error
REG_ERR_UNKNOWN	4000_H	Unknown error – internal use only
REG_ERR_NOT_IMPLEMENTED	2000_H	Feature not implemented on this device
REG_ERR_ACCESS_DENIED	1000_H	Action not permitted for the current access rights
REG_ERR_UNDER_RANGE	0800_H	Value less than RangeMin
REG_ERR_OVER_RANGE	0400_H	Value greater than RangeMax
REG_ERR_ILLEGAL_VALUE	0200_H	Value not compatible with data type
REG_ERR_ILLEGAL_OPERATION	0100_H	Operation not defined
REG_ERR_CANNOT_SAVE	0080_H	Write Operation failed
REG_ERR_BAD_PARAMETER	0040_H	Parameter not valid
REG_ERR_MENU_IN_USE	0020_H	Cannot modify register values while SETUP menus are active
REG_ERR_RESERVED_4	0010_H	
REG_ERR_RESERVED_3	0008_H	
REG_ERR_RESERVED_2	0004_H	
REG_ERR_RESERVED_1	0002_H	
REG_ERR_DATA_ERROR	0001_H	Internal data error

12.4. Register Identifiers

Register	Code	Type	Example	Description
REG_REGISTER_VERSION	0001_H	STRING	V1.0	Version of the protocol
REG_COPYRIGHT	0002_H	STRING	(C) Rinstrum 2003	Copyright message
REG_UNIT_MODEL	0003_H	STRING	R320	Model of the instrument
REG_SOFTWARE_VERSION	0004_H	STRING	V1.2	Software version
REG_UNIT_SERIAL_NO	0005_H	ULONG	3106432	Instrument Serial Number
REG_KEYBOARD	0008_H	USHORT	KEY_TARE	Write to simulate key presses
REG_DISPLAY_RAW	0009_H	BLOB	0100005B6D3F…	Contents of LCD display memory
REG_SAVE_SETTINGS	0010_H	EXECUTE		Save Settings command
REG_MENU_MAIN	0011_H	MENU		Start of menu definitions
REG_CAL_COUNT_OIML	0012_H	USHORT	00000033	OIML Calibration counter
REG_CAL_COUNT_NTEP	0013_H	USHORT	0000000D	NTEP Calibration counter
REG_CFG_COUNT_NTEP	0014_H	USHORT	00000014	NTEP Configuration counter
REG_ENTER_PASS_FULL	0019_H	ULONG	000004D2	Write to attempt Passcode Read is only allowed if correct Write 0 to lock instrument again
REG_ENTER_PASS_SAFE	$001A_H$	ULONG	000009A4	

Register	Code	Type	Example	Description
REG_ADC_SAMPLE_NUMBER	0020_H	ULONG	00169196	Counts each new reading
REG_SYSTEM_STATUS	0021_H	ULONG	00000200	Instrument Status
REG_SYSTEM_ERROR	0022_H	ULONG	00000000	Instrument Error Status
REG_ABSOLUTE_MVV	0023_H	WEIGHT	00006187	Absolute signal in mV/V
REG_WEIGHT_DISPLAY	0024_H	WEIGHT	000005DC	Displayed weight reading
REG_WEIGHT_USER	0025_H	WEIGHT	000005DC	Gross or Net weight
REG_WEIGHT_GROSS	0026_H	WEIGHT	000005DC	Gross weight
REG_WEIGHT_NET	0027_H	WEIGHT	000001F4	Net weight
REG_WEIGHT_TARE	0028_H	WEIGHT	000003E8	Tare weight
REG_WEIGHT_PEAK	0029_H	WEIGHT	000009C4	Peak held weight
REG_WEIGHT_HOLD	$002A_H$	WEIGHT	00000000	Manual held weight
REG_WEIGHT_TOTAL	$002B_H$	WEIGHT	000009C4	Total weight reading
REG_WEIGHT_LIVESTOCK	$002D_H$	WEIGHT	000001F4	Livestock weight
REG_WEIGHT_PT_TARE	$002E_H$	WEIGHT	000003E8	Preset Tare weight
REG_FULLSCALE	$002F_H$	LONG	00000BB8	Fullscale setting
REG_WEIGHT_NET_TOTAL	0030_H	LONG	00000208	Total Net Weight Reading (K303 & K307 only)
REG_WEIGHT_GROSS_TOTAL	0031_H	LONG	0000020C	Total Gross Weight Reading (K303 & K307 only)
REG_STREAM_DATA	0040_H	BLOB	0017ABC0000A…	Block of data selected by STREAM_REGs
REG_STREAM_MODE	0041_H	OPTION		
REG_STREAM_REG1	0042_H	MENU	00000001	Index for first register data
REG_STREAM_REG2	0043_H	MENU	00000002	Index for second register data
REG_STREAM_REG3	0044_H	MENU	00000005	Index for third register data
REG_PASSCODE_FULL	$00D0_H$	ULONG	000004D2	Actual Full Passcode
REG_PASSCODE_SAFE	$00D1_H$	ULONG	000009A4	Actual Safe Passcode
REG_MENU_1 .. REG_MENU_10	$00E0_H$ $00E9_H$	MENU		Sub-menu items
REG_CAL_STAGE0	$00F0_H$	BLOB	3D570025000EF…	Data block for stage 0
REG_CAL_STAGE1	$00F1_H$	BLOB	0BA302000FFFF…	Data block for stage 1
REG_CAL_STAGE2	$00F2_H$	BLOB	08E0000000000…	Data block for stage 2
REG_CAL_STAGE3	$00F3_H$	BLOB	00D8000D000E…	Data block for stage 3
REG_WEIGHT_CALIBRATION	0100_H	WEIGHT	000005DC	Weight used for calibration
REG_CALIBRATE_ZERO	0102_H	EXECUTE		Calibrate Zero command
REG_CALIBRATE_SPAN	0103_H	EXECUTE		Calibrate Span command

Register	Code	Type	Example	Description
REG_CALIBRATE_LIN1 .. REG_CALIBRATE_LIN10	0104_H $010D_H$	EXECUTE		Linearisation commands (K302 & K304 only)
REG_ZERO_MVV	0111_H	WEIGHT	FFFFFFFF	Calibrated Zero mV/V
REG_SPAN_WEIGHT	0112_H	WEIGHT	000009C4	Calibrated Span Weight
REG_SPAN_MVV	0113_H	WEIGHT	000061A8	Calibrated Span mV/V
REG_LIN1_WEIGHT .. REG_LIN10_WEIGHT	0114_H $011D_H$	WEIGHT	08000001	Weight at Linearisation points > 00100000 means no point (K302 & K304 only)
REG_RESOLUTION	0122_H	OPTION	00000000	Resolution setting
REG_GRADS	0123_H	OPTION	00000BB8	Graduations setting (K305 & K306 only)
REG_DECIMAL_PLACES	0128_H	OPTION	00000000	Decimal Point position
REG_UNITS	0129_H	OPTION	00000000	Units selection
REG_CABLE_MODE	$012A_H$	OPTION	00000000	4/6 wire operation
REG_HIRES_MODE	$012B_H$	OPTION	00000000	High resolution (x10) mode
REG_TRADE_USE	0130_H	OPTION	00000000	Trade or Industrial operation
REG_FILTER	0131_H	OPTION	00000000	Filtering setting
REG_MOTION	0132_H	OPTION	00000000	Motion setting
REG_ZERO_RANGE	0133_H	OPTION	00000000	Zero range setting
REG_ZERO_TRACKING	0134_H	OPTION	00000000	Zero tracking setting
REG_ZERO_INIT	0135_H	OPTION	00000000	Initial zero operation
REG_ZERO_BAND	0136_H	LONG	00000000	Zero band setting
REG_AUTO_TARE_THRESH	0138_H	LONG	00000000	Automatic tare threshold
REG_SERIAL_TYPE	0140_H	OPTION	00000000	Type of serial output
REG_SERIAL_FORMAT	0141_H	OPTION	00000000	Format of serial output
REG_SERIAL_BAUD	0142_H	OPTION	00000000	Baud rate
REG_SERIAL_BITS	0143_H	BITFIELD	00000000	Parity and Stop bits
REG_SERIAL_ADDRESS	0144_H	BYTE	0000001F	Network Address
REG_PRINTER_SEQUENCE	0145_H	EXECUTE		Reset Printer Seq. command
REG_PROG_PRINT	0146_H	BLOB	00D8000D000E…	Programmable Printing Short
REG_PROG_PRINT_SUMMARY	0147_H	BLOB	00D8000D000E…	Programmable Printing Long
REG_CLOCK	0150_H	STRING	16/10/2003 10:32	Full Date/Time string
REG_CLOCK_FORMAT	0151_H	OPTION	00000000	Date format selection
REG_DATA_DD	0152_H	USHORT	00000011	Date
REG_DATA_MM	0153_H	USHORT	0000000A	Month
REG_DATE_YYYY	0154_H	USHORT	000007D3	Year
REG_TIME_HH	0155_H	USHORT	0000000A	Hour
REG_TIME_MM	0156_H	USHORT	00000020	Minute
REG_TIME_SS	0157_H	USHORT	00000015	Second

Register	Code	Type	Example	Description
REG_KEY_LOCK	0160_H	BITFIELD	00000000	Key locking
REG_USER_KEY_FUNC	0161_H	OPTION	00000000	Special Function setting
REG_AUTO_OFF_TIME	0162_H	OPTION	00000000	Auto power off setting
REG_BACKLIGHT	0163_H	OPTION	00000000	Backlight options
REG_REMOTE_KEY_FUNC	0164_H	OPTION	00000000	Remote key function setting
REG_BAT_VOLT	0165_H	OPTION	00000000	Battery voltage selection
REG_WD_KEY_LOCK	0166_H	OPTION	00000000	Washdown Key locking (K304 & K306 only)
REG_SETPT_TYPE_1	0170_H	OPTION	00000000	Setpoint 1 Type
REG_SETPT_SRC_1	0171_H	OPTION	00000000	Data Source Setpoint 1
REG_SETPT_TARGET_1	0172_H	LONG	000003E8	Setpoint 1 Target
REG_SETPT_TYPE_2	0173_H	OPTION	00000000	Setpoint 2 Type
REG_SETPT_SRC_2	0174_H	OPTION	00000000	Data Source Setpoint 2
REG_SETPT_TARGET_2	0175_H	LONG	000003E8	Setpoint 2 Target
REG_COUNT_QTY	0180_H	ULONG	0000000A	Counting sample quantity
REG_OVERLOAD_COUNT	0181_H	ULONG	0000000D	Input Overload Count
REG_CLEAR_OVERLOAD	0182_H	EXECUTE		Reset Input Overload Count

12.5. Key Codes

Key Code	Code	Description
KEY_NONE	0000_H	No key pressed
KEY_SETUP_FULL	7001_H	Access to Full setup menu
KEY_SETUP_SAFE	7002_H	Access to Safe setup menu
KEY_ZERO	7201_H	Zero function
KEY_TARE	7202_H	Tare function
KEY_GROSS_NET	7203_H	Gross/Net function
KEY_PRINT	7204_H	Print function
KEY_USER_FN1	7205_H	User F1 function
KEY_POWER_ON	7301_H	Power key press
KEY_POWER_OFF	7302_H	Power Off function
KEY_PHYSICAL_1 .. KEY_PHYSICAL_20	8001_H 8014_H	Physical key codes. Eg. Key 1 = Power, Key 2 = Zero etc.
KEY_PHYSICAL_1 .. KEY_PHYSICAL_2	8081_H 8094_H	Long presses (> 2 seconds) of the Physical keys

12.6. Decimal, Binary and Hexadecimal Conversion

Decimal (radix 10)	Binary (radix 2)	Hexadecimal (radix 16)
0	0000 0000 $_b$	00 $_H$
1	0000 0001 $_b$	01 $_H$
2	0000 0010 $_b$	02 $_H$
3	0000 0011 $_b$	03 $_H$
4	0000 0100 $_b$	04 $_H$
5	0000 0101 $_b$	05 $_H$
6	0000 0110 $_b$	06 $_H$
7	0000 0111 $_b$	07 $_H$
8	0000 1000 $_b$	08 $_H$
9	0000 1001 $_b$	09 $_H$
10	0000 1010 $_b$	0A $_H$
11	0000 1011 $_b$	0B $_H$
12	0000 1100 $_b$	0C $_H$
13	0000 1101 $_b$	0D $_H$
14	0000 1110 $_b$	0E $_H$
15	0000 1111 $_b$	0F $_H$
16	0001 0000 $_b$	10 $_H$
17	0001 0001 $_b$	11 $_H$
18	0001 0010 $_b$	12 $_H$
...	...	...
254	1111 1110 $_b$	FE $_H$
255	1111 1111 $_b$	FF $_H$

Table 7: Decimal, Binary and Hexadecimal Conversion

12.7. Setup and Calibration Errors

These messages show status messages or errors that may occur during the instrument setup and calibration.

Error	Description	Resolution
(ENTRY) (DENIED)	The instrument may be in Safe Setup and an item that needs Full Setup has been selected for editing.	Access Full Setup to edit the item.
	When accessing setup, more than three attempts have been made with the incorrect passcode.	Turn the instrument off. When the instrument is turned back on, enter the correct passcode to access setup.
(LIN.PT) (LO)	An attempt has been made to place a linearisation point below zero.	Incorrect linearisation point entered (must be between zero and full scale).
(PT.TOO) (CLOSE)	An attempt has been made to place a calibration point too close to an existing calibration point.	Re-enter the calibration point. Points must be spaced by at least 2% of full scale from each other.
(RES) (LO)	The scale build is configured for less than 100 graduations.	Check the resolution (count-by) and capacity settings.
(RES) (HIGH)	The scale build is configured for more than 30,000 graduations.	Check the resolution (count-by) and capacity settings.
(SPAN) (LO)	The load cell signal range (span) is too small for these settings.	Incorrect span weight entered (must be between zero and full scale). Scale wiring incorrect. Wrong load cell capacity (too large). Wrong or no calibration weight added to scale.
(SPAN) (HI)	The load cell signal range (span) is too large for these settings.	Incorrect span weight entered (must be between zero and full scale). Scale wiring incorrect. Load cell capacity too small for application.
(ZERO) (LO)	An attempt has been made to calibrate zero below -2mV/V.	Scale wiring incorrect
(ZERO) (HI)	An attempt has been made to calibrate zero above +2mV/V.	Remove all weight from scale. Scale wiring incorrect.

12.8. Glossary Terms

Term	Definition
Calibration Parameter	An adjustable parameter that can affect measurement or performance accuracy and, due to its nature, needs to be updated on an ongoing basis to maintain device accuracy, eg., span adjustments, linearisation factors, and coarse zero adjustments
COMM	The communications protocol used to communicate with the R300 Series
Configuration Parameter	An adjustable or selectable parameter for a device feature that can affect the accuracy of a transaction or can significantly increase the potential for fraudulent use of the device and, due to its nature, needs to be updated only during device installation or upon replacement of a component, eg., division value (increment), sensor range, and units of measurement.
Count-by	The smallest change in weight units that the display can show. See also Resolution.
CRC	Cyclic Redundancy Check
Division	A single graduation.
EEPROM	Electrically Erasable Programmable Read-Only Memory
FIR	Finite Impulse Response
Full Scale	The maximum gross weight allowed on the scale. This is used to detect overload and underload conditions, etc.
Graduations	The maximum number of display steps between zero gross load and full capacity gross load. It is equal to the full scale divided by the resolution.
LED	Light Emitting Diode
NTEP	National Type Evaluation Program
OIML	International Organization of Legal Metrolology
opto-LINK Cable	opto-isolated infrared communications link cable
PLC	Programmable Logic Controller
Range	Total change in weight between zero gross load and full capacity gross load (ie. the nominated total capacity of the scale). It is always given in displayed weight units.
Register	
Resolution	The smallest change in weight units that the display can show. See also Count-by.
Ring Network	A network of up to 31 Instruments connected to a central computer
RS-232	Standard for communications hardware layers.
Step-Response	The step-response is the time between placing a weight on the scale and the correct weight reading being displayed.
Transients	A temporary voltage oscillation or spike caused by a sudden change of load (or other external influence).
Units	The actual units of measurement (kilograms, tonnes, pounds, etc.).

12.9. List of Figures

12.10. List of Tables

13. Index

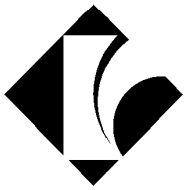
Notes:

Notes:

Notes:

Notes:

SMART WEIGHING SOLUTIONS



rinstrum