DCell

DSC

DCell & DSC
Strain Gauge or Load Cell Embedded Digitiser Module – 2nd Generation

Software Version 3 onwards

User Manual

mantracourt.com

mantracourt

Contents

Chapter 1 Introduction 4
Overview 4
Key Features 4
Special Facilities 5
Version 3 Additions and Enhancements 5
Version 4 Additions and Enhancements 6
The Product Range 6
Which Device To Use 6
Additional DCell & DSC Variants Available 7
Some Application Examples 8
Chapter 2 Getting Started with the Evaluation Kit 9
The Evaluation Kit 9
Contents 9
Checking the Device Protocol Type and Station Number 10
Connecting Up The Evaluation Kit For RS485 10
Connecting Up The Evaluation Kit For RS232 10
Initial Checks 11
Instrument Explorer 11
What Can Instrument Explorer Do? 11
Installing Instrument Explorer 11
Running the Instrument Explorer Software 11
Instrument Explorer Icon 12
Instrument Explorer Window 12
Instrument Settings 13
Viewing Device Data 14
Instrument Explorer Parameter List 14
Connecting a Load Cell 15
DSJ1 Evaluation Board Sensor Connections 16
Performing A System Calibration 17
Sys Calibration, Table method 18
Sys Calibration, Auto Method 19
Chapter 3 Explanation of Category Items 22
Communications 22
Station Number, STN 22
Baud rate Control, BAUD 23
Reset: New For Version 4 (DSC Only) 23
DSC Issue 3 pcb or Higher 23
Communications Failure Count, CFCT 24
Output Format Controls, DP and DPB (ASCII ONLY) 24
Information 24
Software Version, VER 24
Serial Number, SERL and SERH 24
Strain Gauge 25
mV/V output, MVV 25
Nominal mV/V level, NMVV 25
mV/V Output In Percentage Terms, ELEC 25
Temperature Value, TEMP 25
Output Rate Control, RATE 25
Dynamic Filtering, FFST and FFLV 25
Cell 27
Temperature Compensation In Brief 27
Cell Scaling, CGAI, COFS 27
Cell Limits, CMIN, CMAX 28
Linearisation In Brief 29
System 29
System Scaling, SGAI, SOFS 29

System Limits, SMIN, SMAX 30
System Zero, SZ 30
System Outputs, SYS, SOUT 30
Reading Snapshot, SNAP, SYSN 30
Control 31
Shunt Calibration Commands, SCON and SCOF 31
Digital Output, OPON and OPOF 31
Flags 31
Diagnostics Flags, FLAG and STAT 31
Latched Warning Flags (FLAG) 31
Dynamic Status Flags (STAT) 32
Output Update Tracking 33
User Storage 33
USR1…USR9 33
Reset 33
The Reset command, RST 33
WARNING: Finite Non-Volatile Memory Life 33
Chapter 4 The Readings Process 34
Flow diagram 34
Cell and System Scaling 35
Calibration Parameters Summary and Defaults 36
Chapter 5 Temperature Compensation 37
Purpose and Method of Temperature Compensation 37
Temperature Module Connections and Mounting (DTEMP) 37
DCELL connections to DTEMP 37
DSC connections to DTEMP 37
Control Parameters 38
Internal Calculation 38
The Temperature Measurement 39
How to Set Up a Temperature Compensation 39
Parameter Calculations 40
Chapter 6 Linearity Compensation 41
Purpose and Method of Linearisation 41
Control Parameters 41
Internal Calculation 41
How to Set Up Linearity Compensation 42
Parameter Calculations and Example 42
Chapter 7 Self-Diagnostics 44
Diagnostics Flags 44
Diagnostics LED 44
Chapter 8 Communication Protocols 45
Bus Standards 45
Serial Data Format 45
Communications Flow Control 45
Communications Errors 45
Choice of Bus Formats 45
The RS232 Bus Standard 45
The RS485 Bus Standard 46
Communications Protocols 46
Choosing a Protocol 46
Communications Software for the Different Protocols 46
Common Features of All Protocols 47
Data Type Conversions and Rounding 47
The ASCII Protocol 48
Continuous Output Stream (ASCII ONLY) 50
New for Version 4 50
The MODBUS-RTU Protocol 50
The Mantrabus-II Protocol 52

Chapter 9 Software Command Reference **55**
Commands in Access Order 55
Chapter 10 Installation **57**
Before Installation 57
Physical Mounting 57
Electrical Protection 57
Moisture Protection 57
Soldering Methods 58
Power Supply Requirements 58
Cable Requirements 58
Strain Gauge Input (DSC) 58
Power and Communication 58
Temperature Sensor 59
Identifying Strain Gauge Connections 59
DCell Input Connections 59
DSC Input Connections 59
Identifying Bus-End Connections 60
DCell Bus Connections 60
DSC4-RS485 Versions-Bus Connections 60
DSC2-RS232-Bus Connections 60
Strain Gauge Cabling and Grounding Requirements 61
DCell Strain Gauge Wiring 61
DCell Strain Gauge Wiring Arrangement 61
DSC Strain Gauge Cabling Arrangement 62
Communications Cabling and Grounding Requirements 62
DCell Power and Communications Wiring 62
DSC4-RS485 Versions- Power and Communications Wiring 63
DSC2-RS232 Versions Bus-End Arrangement 63
Suitable Cable Types 64
DCell/DSC4-RS485 Versions-RS485 Bus Cable 64
Warning: Special Problems with Portable Computers 64
To Avoid These Problems 64
RS232 Bus Layout 64
RS485 Bus Layout 64
RS485 Bus Connections for Multiple DCells 65
RS485 Bus Connections for Multiple DSC RS485 Versions 65
RS232 & RS485 Bus Converters 66
Strain Gauge Sensitivity Adjustment (DSC ONLY) 67
Identifying the DSC 'Rg' Resistor 67
Chapter 11 Troubleshooting **68**
LED Indicator 68
No Communications 68
Bad Readings 69
Unexpected Warning Flags 69
Problems With Bus Baud Rate 70
Recovering a "lost" DCell/DSC 70
Chapter 12 Specifications **71**
Technical Specifications DSC/DCELL High Stability 71
Technical Specifications DSC/DCELL Industrial Stability 72
Mechanical Specification for DSC 73
Mechanical Specification for DCell 73
CE Approvals 73
Warranty 74

Chapter 1 Introduction

This chapter provides an introduction to DCell/DSC products, describing the product range, main features and application possibilities.

Overview

The DCell and DSC products are miniature, high-precision Strain Gauge Converters; converting a strain gauge sensor input to a **digital** serial output. They allow multiple high precision measurements to be made over a low-cost serial link. Outputs can be accessed directly by PLCs or computers, or connected via various types of network, telephone or radio modem, all without compromising accuracy.

Key Features

Ultra-miniature
The DCell 'puck' format can be fitted inside most load cell pockets, and similar restricted spaces. The DSC cards are similarly very small, optimised for mounting as a component onto custom PCBs.

High-precision

Industrial Version.
25ppm basic accuracy (equates to 16 bit resolution)

High Stability
5ppm basic accuracy (equates to 18 bit resolution) with comparable stability – far exceeds standard instrument performance.

Low-power
Low-voltage DC supply (5.6V min), typically 40mA for RS485 &.52mA for RS232 (including 350R strain gauge).

Adjustable sensitivity
Configured for standard 2.5mV/V full-scale strain gauges as supplied.
A single additional resistor configures the input between 0.5 and 100 mV/V full-scale.

Temperature sensing and compensation (optional)
An optional temperature sensor module is available and advanced 5-point temperature-compensation of measurement.

Linearity compensation
Advanced 7-point linearity compensation.

Serial output
Lower-cost cabling, improved noise immunity, and longer cable runs with no accuracy penalty.
Device addressing allows up to 253 devices on a single bus, drastically reducing cabling cost and complexity.
Two-way communications allow in-situ re-calibration, multiple outputs and diagnostics.
No separate measuring instruments needed.

Digital calibration
Completely drift-free, adjustable in-system and/or in-situ via standard communications link.
Two independent calibration stages for load cell-and-system-specific adjustments.
Programmable compensation for non-linearity and temperature corrections.
Calibration data is also transferable between devices for in-service replacement.

Self-diagnostics
Continuous monitoring for faults such as strain overload, over/under-temperature, broken sensors or unexpected power failure.
All fault warnings are retained on power-fail.

Multiple output options
Choice of two communications standards: RS232 (DSC only) or RS485.
Choice of three different protocols: ASCII, MODBUS or MANTRABUS, for ease of integration.
All variants provide identical features and performance.

Special Facilities

Output Capture Synchronisation
A single command instructs all devices on a bus to sample their inputs simultaneously, for synchronised data capture.

Output Tare Value
An internal control allows removal of an arbitrary output offset, enabling independent readings of net and gross measurement values.

Dynamic Filtering
Gives higher accuracy on stable inputs, without increased settling time.

Programmable Output Modes
Output rate control enables speed/accuracy trade-off.
ASCII output version provides decimal format control and continuous output mode for 'dumb terminal' output.

Unique Device Identifier
Every unit carries a unique serial-number tag, readable over the communications link.

Communications Error Detection
An interruption of normal communications due to drop-outs or noise is detected as badly formatted receive data, which triggers a communications failure counter to be incremented.

External Temperature Sensing (optional)
An external temperature module for improved accuracy (especially tracking changing temperature conditions).

Software Reset
A special communications command forces a device reboot, as a failsafe to ensure correct operation.

Version 3 Additions and Enhancements

The following are an outline only more detail will be found further on in this manual

DCell

- Easy mounting via a 2mm screw
- Connection via solder holes to either side of PCB
- Lower profile, single PCB construction

DSC

- Additional I/O
- Easier shielding connection at load cell connector end

DCell & DSC

- Baud Rates to 230400. Higher baud rate allow faster polling of groups of devices on a single bus, also faster measurement sampling requires faster communications for individual devices allowing the higher sampling to be realised over the communications link.
- Higher sampling rates. Sampling to 500Hz can now be achieved. Also more sampling rates are available as follows 1, 2, 5, 10, 20, 50, 60, 100, 200, 300 & 500Hz.
- Lower cost. With new technology and further use of miniaturisation the cost is now lower.
- ESD protection to +/-15KV (RS485 version). Higher ESD voltage rating reduces the possibility of damage by static.
- Real mV/V calibration. Instead of % full scale the base measurement is in mV/V and is factory calibrated to within 0.1%. the % of FS output "ELEC" is still available.
- Extreme Noise Immunity, 5 x heavy industrial level.
- Diagnostics LED. An LED is used to indicate that the device is powered and working correctly. The LED is also used to indicate which protocol the device is.
- Remote shunt cal. A 100K 1% 50ppm/Deg C resistor can be switched across the bridge to allow load cell integrity to be established.
- Peak & Trough Measurements. Added to allow the faster rates to hold a peak or trough readings. These are stored in volatile memory & are therefore reset on power up.

- Programmable dynamic filtering. The filtering is the same as used on Version 2 but with the advantage of being able to set the characteristics using the communications.
- Wide Operating Voltage. The operating voltage is now 5.6 to 18V allowing the device to be powered from a wider range of available system supplies.
- DC Excitation. DC excitation has now been employed allowing longer cable lengths for the load cells which is particularly useful for DSC. This is a 4 wire measurement.
- Defaults have been changed so that the device is shipped with a baud rate of 115200 and station number of 1.
- Scaling implementation has been changed for both "CELL" and "SYS". The gain is applied before the offset thus following the more standard approach. This allows for an offset change to be made easily as the offset is not a component of gain.

Version 4 Additions and Enhancements

DSC

- Reset to default communication parameters. A pair of pads on the underside of the DSC, when shorted together at power up will reset Station no. to 1 and baudrate to 115200

Note: This is only for issue 3 pcb and higher. Please refer to underside for issue marking.

DCell & DSC

- For ASCII versions. Setting Station No. to 998 gives continuous output streaming as version 3, except that, on powering on streaming starts without requiring an XON character

The Product Range

Devices are available in two physical formats.

The **DCell** (puck) products consist of a Digital Strain Gauge Signal Conditioner with RS485 bus output in double sided component population format.
This is suitable for installation in very small spaces, including load cell pockets.
External connections are made by wiring to through hole pads. Mounting is via a 2mm mounting hole to accept M2 screw or American equivalent #0-80.
Important Note: DO NOT USE #2 screw size.

The **DSC** (card) products are very similar to the DCell but in a different physical form for mounting stand-alone or on a board.
The DSC is also available with an RS232 output.
External connections are via header pins which can plug into connectors, or be soldered to wires or into a host PCB. DSC has an open collector output and volt free digital input.

Which Device To Use

It is important to select the correct product for your application.

- First choose DCell or DSC based on your physical installation needs
- Choose the communications protocol depending on performance/integration requirements
- the RS232 output option may be simpler if your system only uses a *single* DSC card

Common Features

Both physical formats offer identical control and near-identical measurement performance
Both are available in all three output protocols: MODBUS, ASCII or MANTRABUS

Differences
Only the DSC (card) is available with the RS232 output option and has the digital Input & output.

Special Aspects To Consider
The DCell fits neatly into a strain gauge pocket
The DSC lends itself to PCB mounting
The RS485 output version *must* be used for multiple devices on the same bus

Additional DCell & DSC Variants Available

A separate variant is available with CANbus output. Refer to DCell & DSC CAN - 2nd Generation - Manual. (These variants are sufficiently different to require their own manuals).

The following order codes are supported by an earlier version manual 'DCell & DSC Version 2'
DLCPKASC, DLCPKMAN, DLCPKMOD, DSC2AS, DSC2MA, DSC2MB, DSC4AS, DSC4MA, DSC4MB

Some Application Examples

Simple Distributed Measurement
Pressure loads are taken at a number of keys points in a manufacturing process, distributed over a large area. Each pressure sensor contains a DCell unit, and all the sensors are connected by a single cable carrying power and RS485 communications. A central PC allows continuous display, monitoring and logging of all values from a central control room. This displays a control-panel and current display window, and logs information to an Excel spreadsheet for future analysis.
Further monitoring checks and displayed information can easily be added when required to the system where up to 253 'nodes' can be installed.

Low Cost Dedicated Weighing Station
A basic load cell weighing-pad device has a cable leading to a wall mounted weight display.

Digital Load Cell
Load cell products are offered with a high-precision digital communications option.
A DCell is fitted into the gauge pocket of each load cell in manufacture. During product testing, each unit undergoes a combined load test and temperature cycle. Each unit is then programmed with individually calculated gain, offset, linearity and temperature compensation tables. All units perform to a very tight specification without the use of any trimming components.

High Reliability Load sensing
A road bridge has a dedicated load monitoring and active control computer system. System calibration adjustments are only established during construction, so sensors must be replaceable without recalibration.
Each load monitoring point has a digital load cell fitted, with calibration values set during construction. Self-diagnostics aid detection of failures.
When a failed load cell is replaced it will produce identical force measurements. The old load cell set-up data values are programmed into the separate user-level calibration store in the unit, to produce an identically performing replacement.

Remote Radio Weigher
A variety of lifting machines in a loading yard can be used with a weighing link to display weight in tonnes on a remote hand-held readout.
A heavy duty strain gauge load-link is fitted with a battery-powered radio modem and DCell. The independent handheld display unit communicates with the DCell over a transparent radio link, providing a simple LCD readout and tare button operation.

Load Balance Monitor
A lorry loading weighpoint monitors left/right load balance and sounds a warning if loading is too uneven for safety.
A drive-on weighing platform is provided with load cells at each of four corners. Each cell is wired to a DSC unit, and these are cabled to a 3rd party LCD display and control unit, producing a complete turnkey system. A digital I/O card is wired to the same bus to control the warning alarm. Application software running on the control unit provides a % left/right balance readout with a graphical tipping display, and a total weight indication.
The balance indication is calculated by comparing the different corner readings. If it exceeds a programmed limit, a command to the I/O card turns the relay on.
Total weight is calculated by summing the individual results mathematically.
Automatic re-zeroing occurs when the total is near zero for more than a few seconds.
A control button enables a set-up mode for recalibration (protected by operator password), which displays individual readings and total. Corner compensation can be checked by observing the changing total as a weight is moved around. Simple button presses control two point recalibration for any cell.

Weighing Subsystem For Process Control
Several strain gauge loads are monitored as part of a larger data acquisition/monitoring system, based around a high-speed Profibus network. The load measurements occur in groups of physically related signals which relate to specific 'area modules' along with a number of other measurements and control outputs. The strain gauges are wired to DSC cards, controlled and interrogated via MODBUS protocol commands on an RS485 bus. The DSCs and other 3rd party MODBUS-compliant devices which govern the area module are all connected to a single RS485 'spur'. The devices in each area-module spur are controlled from the main Profibus backbone, using an off-the-shelf bus gateway unit.

Chapter 2 Getting Started with the Evaluation Kit

This chapter explains how to connect up a DCell/DSC for the first time and how to get it working. For simplicity, this chapter is based on the standard DCell/DSC Evaluation Kit, which contains everything needed to communicate with a puck or card from your PC.
It is advised that first time users wishing to familiarise themselves with the product use the Mantracourt Evaluation Kit. This provides a low cost, easy way to get started.
If you do not have an Evaluation Kit, the instructions in this chapter mostly still apply, but you will need to wire up the device (and possible bus-converter) and have some means of communicating with it. See *Communications* Protocols in *chapter 10* as appropriate to the protocol type.

The Evaluation Kit

Contents

- An Evaluation PCB (DSJ1) which comprises of
  - A 8 way screw connector for the strain gauge & Temperature sensor
  - A 5 way screw connector for power & RS485 comms
  - A 9 way 'D'Type for direct RS232 connection to PC, CAN & RS485
  - Link headers for RS232 or RS485 comms selection
  - Terminating resistor for CAN & RS485
  - ON board Temperature Sensor for easy evaluation of Temperature compensation
  - LED for power indication
  - LED for digital output (DSC only)
  - Push Switch for digital input (DSC only)

- An Evaluation DCell or DSC of your choice
- A CD ROM containing Instrument Explorer software
- A 9 to 25 way 'D'Type adaptor for the PC comms port
- A 9 way 'D'Type extension lead
- **For RS485 ONLY** an RS232 to RS485 converter and connecting cable
- **For RS232 ONLY** a power connection cable

Other Things you will need

- A regulated power supply, capable of providing 10 –18V at 100mA (10V is minimum requirement for RS485 converter)
- A PC running Windows 98 or above, with a spare RS232 communications port and 45Mb free disk space

and, ideally

- A strain gauge, load cell or simulator typically 350-5000 ohms impedance. (Refer to specifications Chapter 12)

Checking the Device Protocol Type and Station Number

Before running the communications application, you will need to know both the protocol to use and the station number of the device.

For DSC the product label shows the product code which determines the protocol and its serial number.

For the DCell the serial number is it's only means of identification. The serial number must be used to cross reference the dispatch note to identify the protocol of the device.

For a DCell, the Product Code is one of the following 6 types

DLCSASC	Industrial Stability ASCII output
DLCSMAN	Industrial Stability MANTRABUS output
DLCSMOD	Industrial Stability MODBUS output
DLCHASC	High Stability ASCII output
DLCHMAN	High Stability MANTRABUS output
DLCHMOD	High Stability MODBUS output

For a DSC card, the Product Code is one of the following 12 types

DSCS4ASC	Industrial Stability RS485 output card with ASCII protocol
DSCS4MAN	Industrial Stability RS485 output card with MANTRABUS protocol
DSCS4MOD	Industrial Stability RS485 output card with MODBUS protocol
DSCS2ASC	Industrial Stability RS232 output card with ASCII protocol
DSCS2MAN	Industrial Stability RS232 output card with MANTRABUS protocol
DSCS2MOD	Industrial Stability RS232 output card with MODBUS protocol
DSCH4ASC	High Stability RS485 output card with ASCII protocol
DSCH4MAN	High Stability RS485 output card with MANTRABUS protocol
DSCH4MOD	High Stability RS485 output card with MODBUS protocol
DSCH2ASC	High Stability RS232 output card with ASCII protocol
DSCH2MAN	High Stability RS232 output card with MANTRABUS protocol
DSCH2MOD	High Stability RS232 output card with MODBUS protocol

NOTE: For evaluation purposes, the electrical output standard RS485 or RS232 is not important: Your kit should contain the correct equipment to connect the device to a PC.
The product code should match your original order.
The serial number of the device is also shown.

The *station number* of a New DCell/DSC device is factory set to 1
This can be changed if required but for ease of evaluating it is suggested that you use this default value.

Connecting Up The Evaluation Kit For RS485

Connect the PC using the 9 way 'D'Type to the RS232/RS485 converter. Plug the cable provided into the converter and connect the other end to the 5 way connector marked J1 on the DSJ1 PCB.
Ensure LK1 & LK5 are set to "CAN/RS485". Fit LK2 which terminates the RS485 comms. Connect the power cable (Red & Black twisted) to your power supply, which has been set to deliver between 10 & 18 volts. Switch on, the Green Power LED should be on.

Connecting Up The Evaluation Kit For RS232

Connect the supplied power cable (Red & Black twisted) to the 5 way connector marked J1. Connect the 9 way 'D'Type extension lead to the 9 way 'D'Type socket marked J3 of the DSJ1 and the other end to the comms port of the PC.
Ensure LK1 & LK5 are set to "RS232" see PCB ident for the markings of these links.
Now connect power cable to your power supply, which has been set to deliver between 10 & 18 volts. Switch on, the Green Power LED should be on.

Note that if your PC serial port has a 25 way serial port connector, you should use the 9 to 25 way 'D' type adaptor provided to connect to the evaluation hardware.

Initial Checks

With no load cell connected The LED of the DCell or DSC should flash OFF for 100ms every 0.5, 1 or 2 seconds depending on protocol. See following table.

Protocol	LED Flash Period
ASCII	0.5 seconds
MODBUS	1 second
MANTRABUS II	2 seconds

Note: If a Load cell is connected and there are no errors then the LED will Flash ON for 100mS then Off for the above period. This being the normal healthy state

Another check that the device is working okay is by noting the current drawn from the supply. For RS232 variants this should be about 33mA for RS485 Variant about 30mA.

Instrument Explorer

Instrument Explorer is Mantracourts own communication interface for our range of standard products. It provides communications drivers for the DCell/DSC products. A complimentary copy is provided on CD-ROM with the DCell/DSC Evaluation Kit. Instrument Explorer can also be downloaded from Mantracourts website www.mantracourt.co.uk/products_software.html

Instrument Explorer is a software application that enables communication with Mantracourt Electronics instrumentation for configuration, calibration, acquisition and testing purposes.
The clean, contemporary interface allows full customisation to enable your Instrument Explorer to be moulded to your individual requirements.

What Can Instrument Explorer Do?

- Save and restore customisable user workspace
- Read and Write individual instrument parameters
- Save and restore parameter configurations
- Log data to a window or file
- Perform calibration and compensation

Installing Instrument Explorer

Install the Instrument Explorer software by inserting the CD in the CD ROM drive. This should start the 'AutoRun' process, unless this is disabled on your computer.
(If the install program does not start of its own accord, run SETUP.EXE on the CD by selecting '*Run*' from the '*Start Menu*' and then entering *D:\SETUP*, where *D* is the drive letter of your CD-ROM drive).

The install program provides step-by-step instructions. The software will install into a folder called *InstrumentExplorer* inside the *Program Files* folder. You may change this destination if required.
Shortcut icons can be created on your desktop and shortcut bar. After installation you may be asked to restart the computer. This should be done before proceeding with communications.

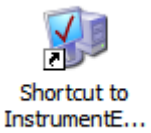
Note about CAN drivers

It is not necessary to install the optional CAN drivers which are selected by default.

Running the Instrument Explorer Software

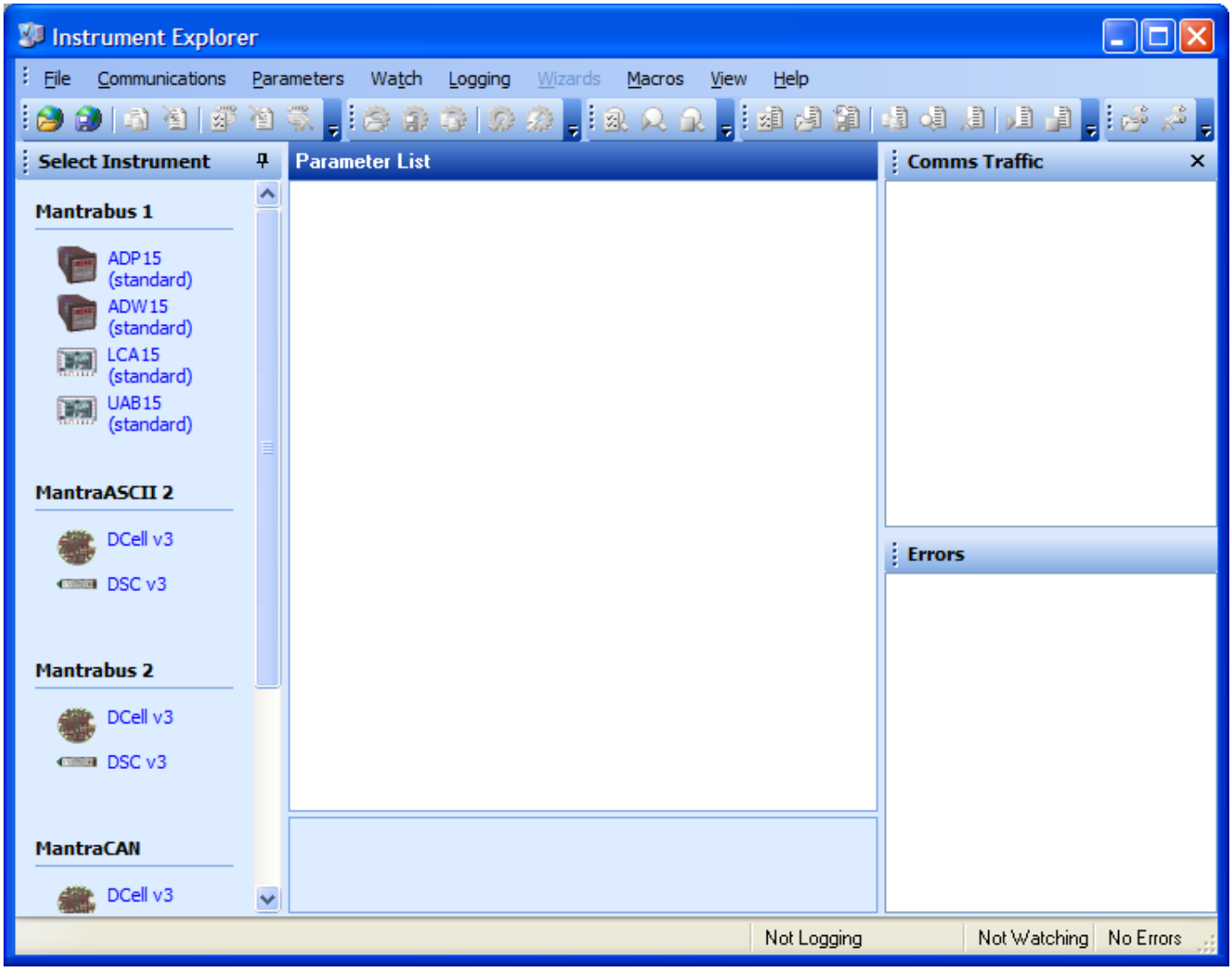
Having installed Instrument Explorer you can now run the application, which the rest of this chapter is based around.
From the Windows 'Start' button, select *Programs*, then *Instrument Explorer* or click on the shortcut on your desktop.

Instrument Explorer Icon



The application should open and look like the following screen shot.

Instrument Explorer Window



The layout of Instrument Explorers Window and child windows allows the user full customisation to their requirements. If the application show a different arrangement of child windows than the above screen shot, then load one of the default workspaces as follows:

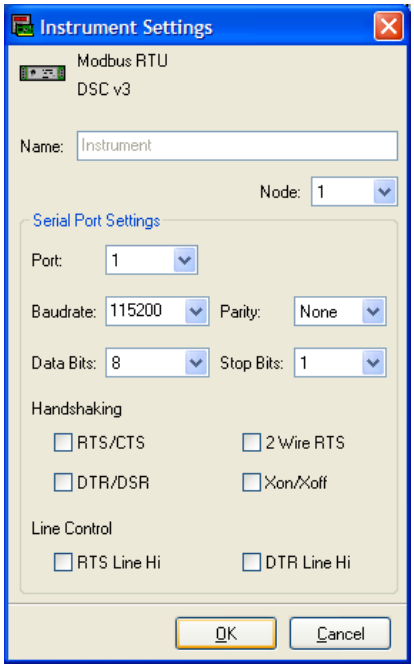
Click **File** on the menu and select **Open Workspace**. From the file dialogue window select **Layout - Standard.iew**. This will ensure your application layout matches this document.

A list of available instruments is displayed in the **Select Instrument** pane of Instrument Explorer. Select the relevant device and protocol, to match the device you are working with, by clicking on the required device icon.

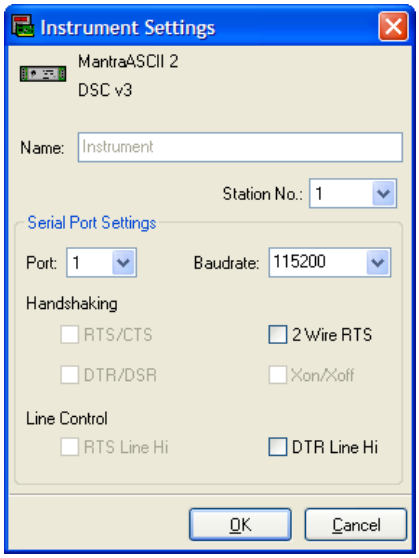
Instrument Settings

One of the following dialogue windows will be displayed:

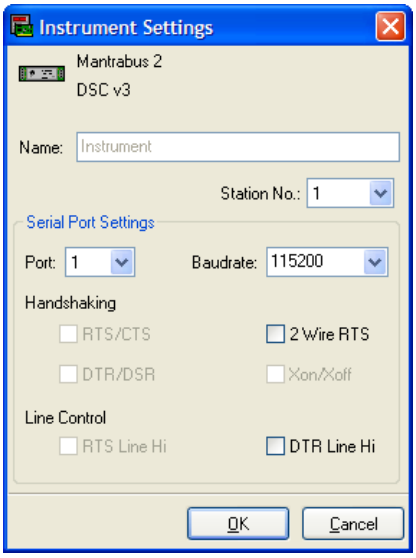
Modbus



MantraASCII



MantraBus



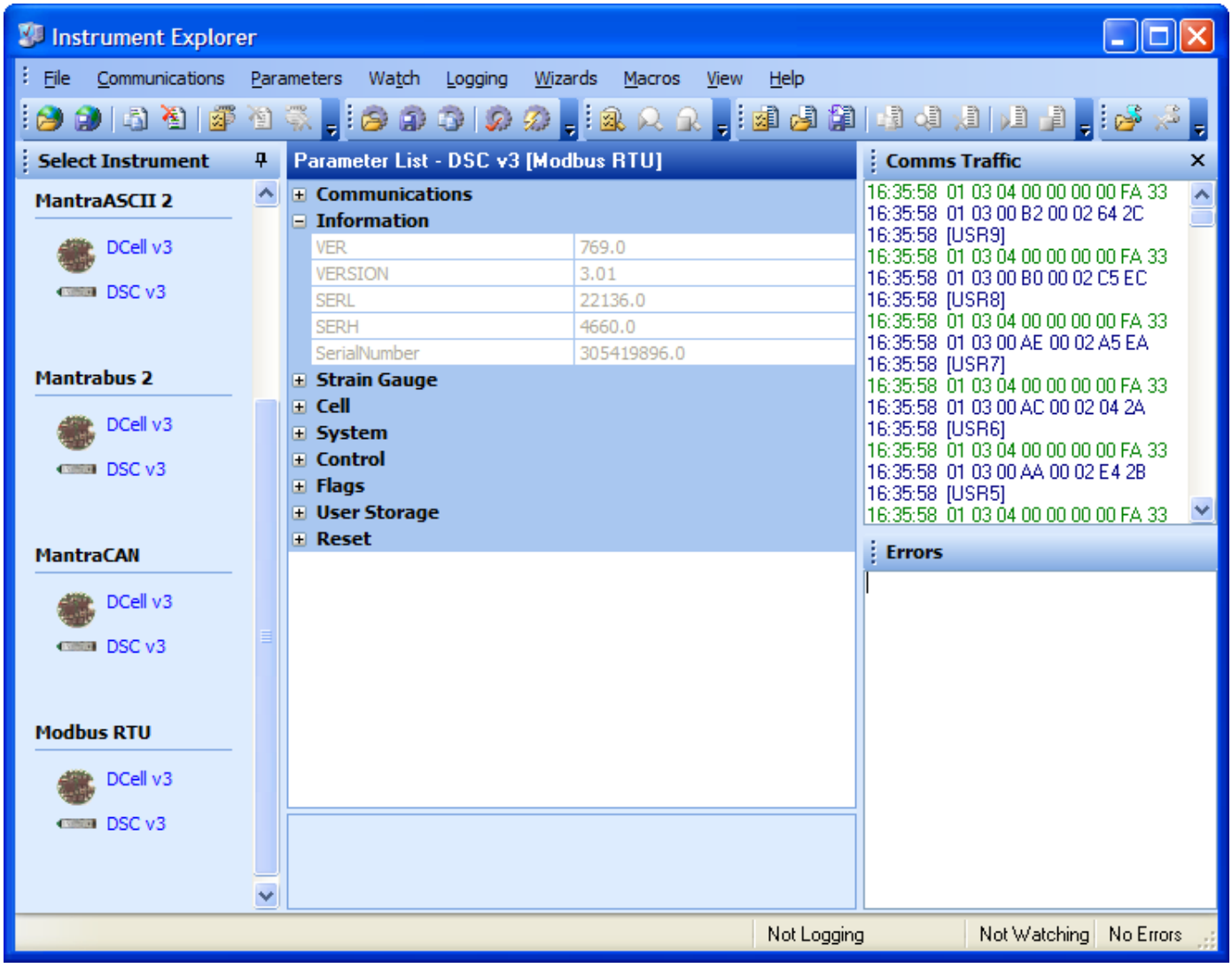
- Select the serial port to which the device is connected.
- Select the baud rate to which the device is set. The factory default is **115200.**
- Select the Station No. / Node. The factory default is **1**.
- Now click the 'OK' button…

The above assumes factory defaults. If your device is known to have different settings use these instead of the ones stated above.

Viewing Device Data

The following main parameter list should now appear in the central pane.

Instrument Explorer Parameter List

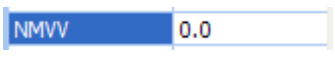


When an instrument has been selected from the Select Instrument Window this parameter list window will become populated.

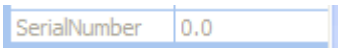
The parameters and commands which are available for the selected device will appear in this list in a structured hierarchic manner enabling the user to expand or contract categories by clicking the ⊞ and ⊟ buttons on the left of the list.

There are four types of parameters and commands:

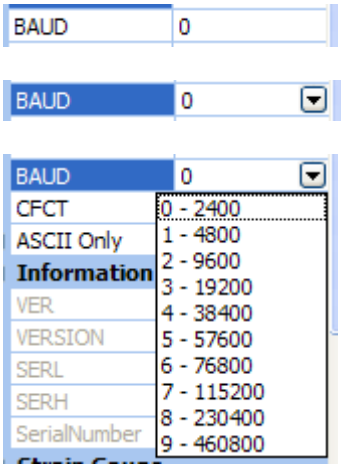
Read/write Numeric – These parameter values are displayed in the right hand column and can be edited by clicking the value.
The value can then be changed and pressing the Enter key or moving away from the edited value will cause the new value to be written to the device. There are no checks on the data entered and it is up to the user to enter the correct data.



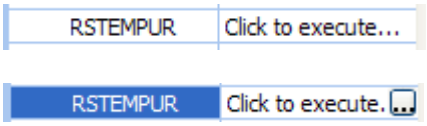
Read-Only – These parameter values are displayed 'greyed out' and cannot be changed.



Read/write Enumerated – These parameters can only be changed by selecting the new value from a drop down list.
Clicking in the right hand column will display a down arrow button which when clicked will display the parameter value options in a list.
Note that all enumerated data (apart from on/off) will be displayed with a numeric value, hyphen then the description of the value.
The numeric value is the value of the parameter and the description is just there to help.



Commands – These commands have 'Click to execute…' displayed in the right hand column. Clicking here will display a … button. Click this to issue the command to the device.



As parameters are changed the communications traffic is displayed in the Traffic Pane.
If any errors occur these will be shown in red in the Error Pane. Once an error occurs it will need to be reset before any more communications can take place. Reset errors by either right-clicking the Error Pane and selecting **Reset Errors** from the pop-up menu or select the **Communications** menu and click the **Reset Errors** item.

To manually refresh the parameter list click the button on the toolbar or select **Sync Now** from the **Parameters** menu.

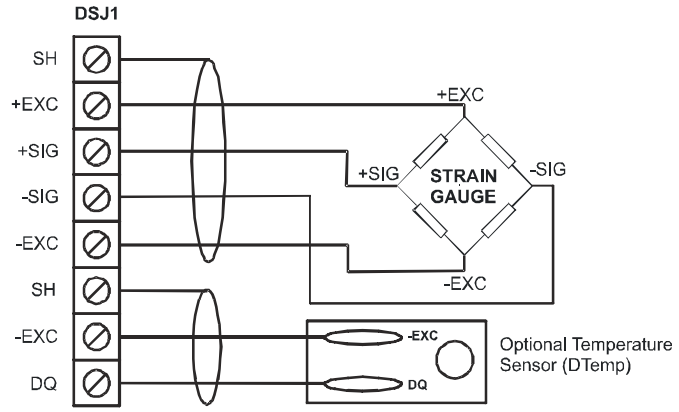
Now you have successfully established communications with your evaluation device the next step is to perform a simple calibration.

Connecting a Load Cell

You can now connect a strain gauge bridge, load cell or simulator to the DCell/DSC.

A suitable strain gauge should have an impedance of 350-5000ohms and (at least for now) a nominal output of around 2.5mV/V.

DSJ1 Evaluation Board Sensor Connections



See Chapter 10 for Connection Method to Strain Gauge
See Chapter 5 for Connections to Temperature Sensor Module

Next, we will set Instrument Explorer to automatically update dynamic parameters from the device so that we can see values such as SYS changing on the screen. To do this either click the button on the toolbar or click on the **Parameters** menu and select the **Auto Sync** item. Note that these options toggle so be sure to leave your selection in the active state.

From the Parameter List click the ⊞ next to the **System** heading to expand this level. The Parameter List should look as follows:

System	
⊞ Scaling	
⊞ Limits	
SRAW	-2.000592
SZ	0.0
SYS	-2.000592
SOUT	-2.000592
SNAP	Click to execute...
SYSN	0.0
⊞ Peak + Trough	

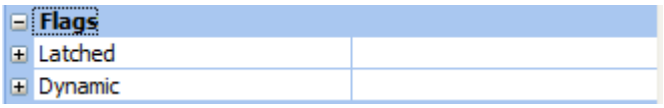
This now exposes more levels that can be expanded as required by clicking the ⊞ next to the heading name.

System	
⊟ Scaling	
SGAI	1.0
SOFS	0.0
⊟ Limits	
SMIN	-100.0
SMAX	100.0
SRAW	-2.000592
SZ	0.0
SYS	-2.000592
SOUT	-2.000592
SNAP	Click to execute...
SYSN	0.0
⊟ Peak + Trough	
PEAK	-2.000592
TROF	-2.000719
RSPT	Click to execute...

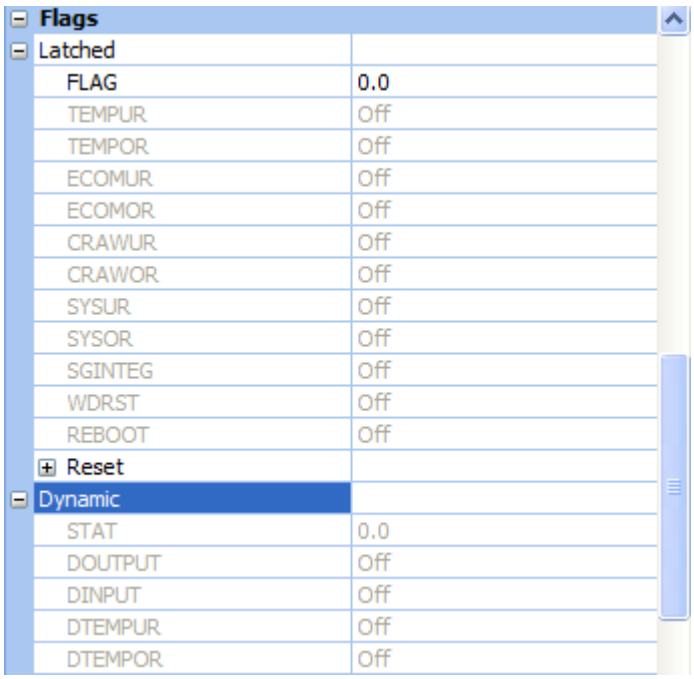
Dynamic values (such as SYS and SRAW) will now be updating in real-time from the device.

Once you have connected the load cell, you should see 'believable' output values, in the "SYS" parameter displayed in the parameter list pane. These values should correspond to mV/V assuming the device is in it's factory default state.

For diagnostics the device has two sets of flags. One being latched and held within the devices non volatile memory (FLAG parameter), the other being dynamic and volatile (STAT parameter).



Instrument Explorer provides a simple method of displaying and resetting of individual flags although these are held within the device in FLAG and STAT parameters.



To quickly clear all the flags simply write zero to the FLAG parameter. Now if no problems exist all flags should in their off state.
If any flags remain on then refer to Chapter 8 for flag definitions.

Performing A System Calibration

The values obtained so far are in mV/V units, these are factory calibrated and fixed to within 0.1% accuracy.

The device also contains two separate user-adjustable calibration parameter groups, these are termed **"Cell"** and **"System"**. Cell being used to convert from mV/V to a calibration value and System to convert this calibration value to the required engineering units. The use of "CELL" is optional. We shall be using "System" for the following exercise where we rescale the output value to read in units of your choice, and to calibrate precisely to your load cell / system hardware.

Instrument Explorer provides Wizards to allow quick and simple calibration operations to be undertaken without the use of a calculator. Wizards can be activated by simply selecting the required item from the Wizard menu.
Since we are now calibrating at system level we have a choice of two calibration methods:

Sys Calibration Table – This technique is used when a manufacturers calibration document is available for the connected strain gauge. This normally gives mV/V to engineering unit values.

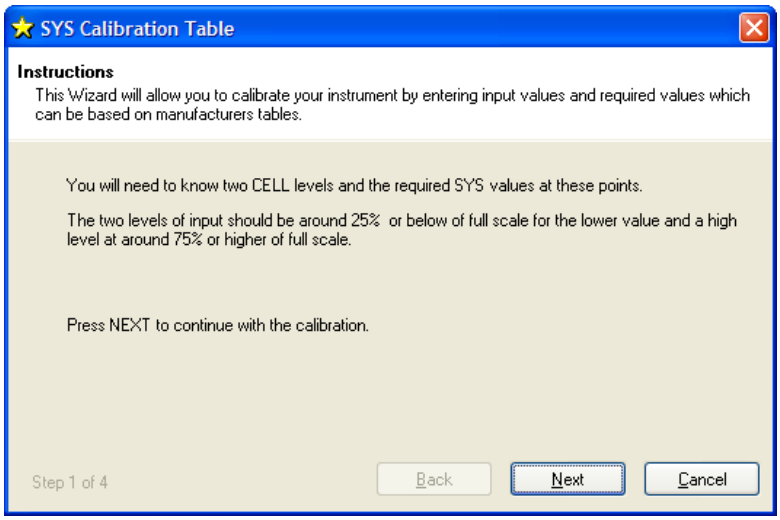
Sys Calibration Auto – This technique is used when the input can be stimulated with real input values. For example you have access to test weight / forces.
We will now describe each of these techniques with an example.

Sys Calibration, Table method

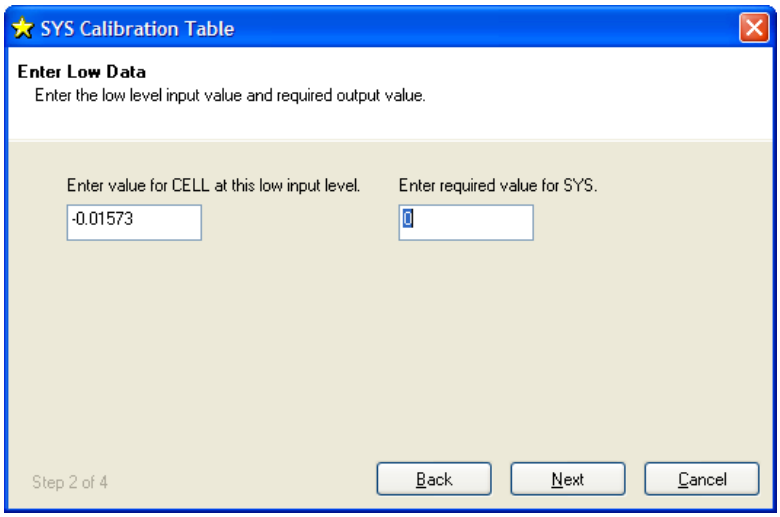
A 10 tonne load cell manufacturer gives the following data:

mV/V output	Force
2.19053	10 tonne
-0.01573	0 tonne

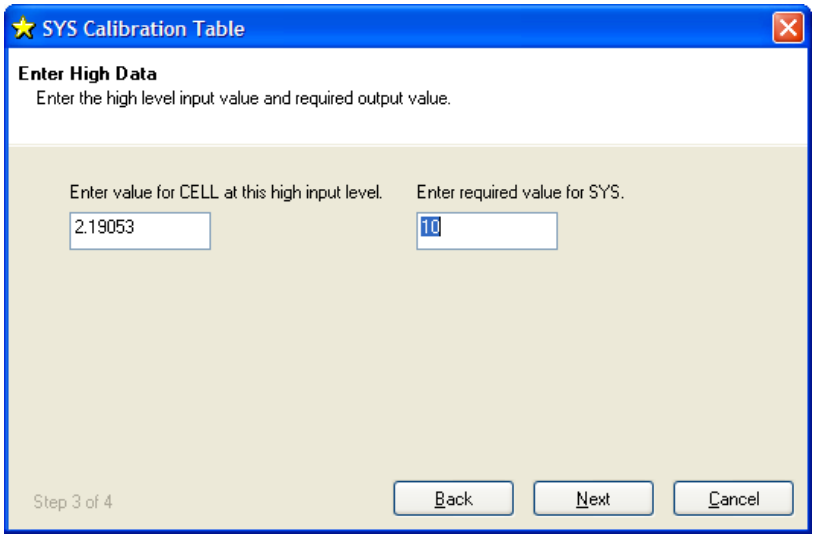
Start the wizard by selecting **Sys Calibration Table** from the **Wizard** menu



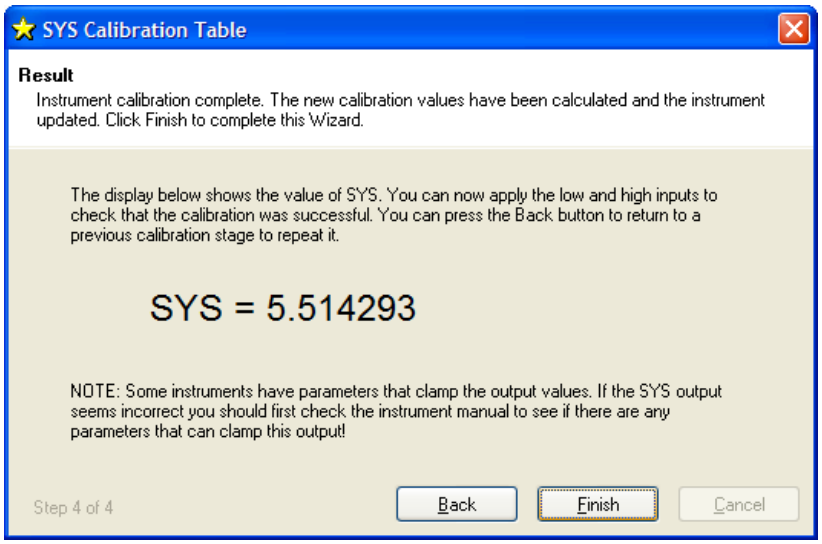
Click the **Next** button and enter the low values as shown below.



Click the **Next** button and enter the high values as shown below.



Click **Next** the following window will be displayed showing the calibrated SYS value which is dependent on the current input values.

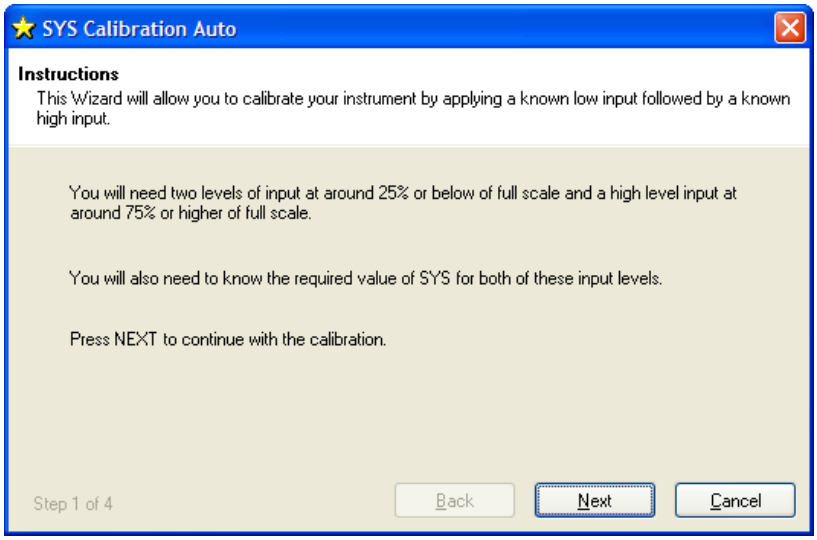


The device is now calibrated. However you may find SYS has been 'clamped' if the resultant SYS is greater than SMAX or less than SMIN. If this is the case then change these values to suitable limits. In this example we may set SMIN to -0.5 (tonne) and SMAX to 12.0 (tonne). This would then provide clamping of SYS to these values and also a flags being set in FLAG and STAT.

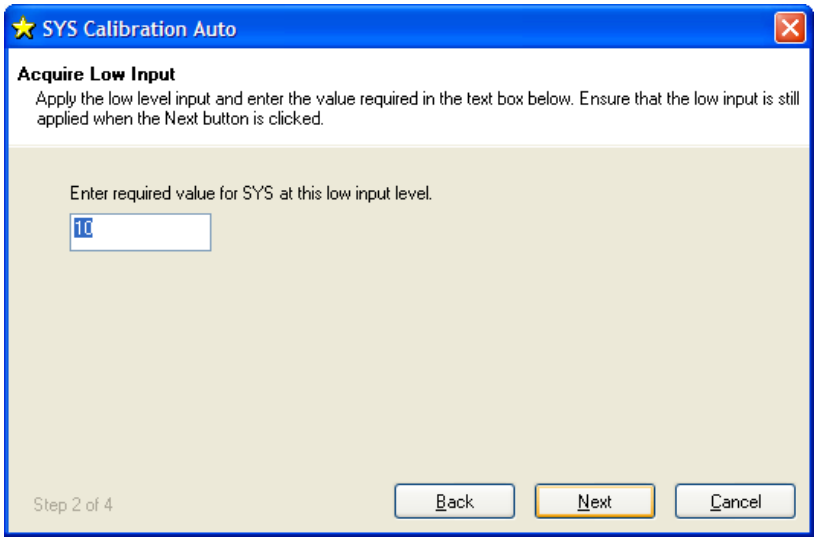
Sys Calibration, Auto Method

Assume we require to calibrate for Kg output and we have available a known accurate 10 Kg and 100 Kg test weights.

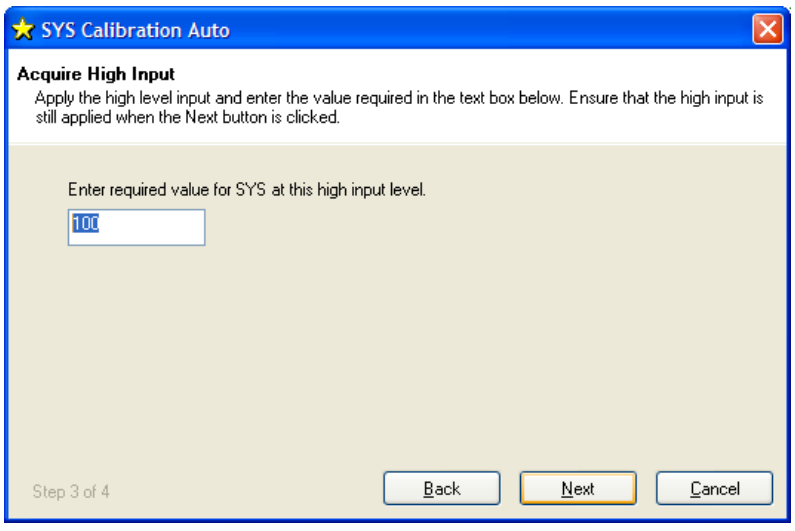
Start the wizard by selecting **Sys Calibration Auto** from the **Wizard** menu



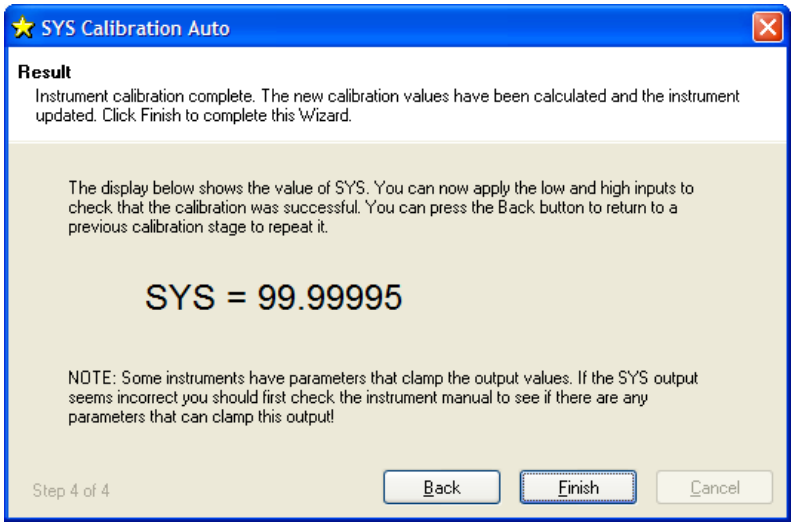
Click Next.



Apply the low known test weight and enter the required SYS value for this weight. In this case it will be 10 as we want the units of SYS to be Kg. Click **Next** to continue



Apply the high known test weight and enter the required SYS value for this weight. In this case it will be 100. Click **Next** to continue.



The device is now calibrated. However you may find SYS has been 'clamped' if the resultant SYS is greater than SMAX or less than SMIN. If this is the case then change these values to suitable limits. In this example we may set SMIN to –0.5 (Kg) and SMAX to 110.0 (Kg). This would then provide clamping of SYS to these values and also a flags being set in FLAG and STAT.

For detailed information about calibration calculations please refer to chapter 3.

Chapter 3 Explanation of Category Items

Instrument Explorer shows the categories to which parameters and generated variables belong. This provides a convenient method for describing the functionality and purpose of each. The categories can be seen from Instrument Explorers Parameter List pane and are as follows.

Communications

This allows the devices station number / node and baud rate to be changed and also to monitor and reset the communications failure counter **CFCT**. For the ASCII protocol there are **DP** and **DPB** controls which set the format of the ASCII string returned by the device. See Chapter 12

Care should be taken when changing the station number or baud rate as communications can be lost with the host. Also note that some commands require the reset (**RST**) command to be sent or a power cycle before the new values take effect. **STN**, **BAUD, DP** and **DPB** are such commands.
When using Instrument Explorer to change either the **STN** or **BAUD** parameter, communications with the device will be lost after the **RST** command has been issued as the software will be using the previous settings. In this case you need to change the device settings in Instrument Explorer by selecting **Change Settings** from the **Communications** menu.

Station Number, STN

The STN parameter controls the 'station number', which specifies the device address for bus communications.
As supplied, devices have the station-number factory set to 1, as described previously in Version 3 Additions and Enhancements, Chapter 1.

Checking the Device Protocol Type and Station Number.
To connect *multiple* devices on the same bus, it is first vital to set all the station numbers to different values. This is because if two devices with the same station number are connected to the same bus, it is not possible to talk to them individually: So in particular, you cannot correct the problem by changing the station number of one of them!
If a bus connects to two devices with the same station number, the only solution is to remove one of them and connect it to a one-to-one link to change its station number.

The new value of STN does not take effect until the RST command is issued or the device is power cycled.
To Change the Station Number of your Device
1. First set **STN** to a suitable new value (making sure that no other device of the same number is also connected!)
2. Send a **RS**T command or power cycle the device.
3. Change the device settings in Instrument Explorer by selecting **Change Settings** from the **Communications** menu.

NOTES:
- The valid range of **STN** depends on the protocol, but it is always at least 1-253
- All the protocol types have a 'bus address' type device identifier, which is known as the 'station number' for MANTRABUS, 'address' for ASCII and 'node id' for MODBUS.
- The valid ranges for different protocols are: 1-253 for MANTRABUS, 1-999 for ASCII and 1-255 for MODBUS.
- In all cases, if **STN** is set outside the valid range, it behaves as if set to a default of 1.

Baud rate Control, BAUD

The **BAUD** parameter is a read/write byte value specifying a standard communications baud rate according to the following table –

BAUD value	0	1	2	3	4	5	6	7	8	9
baud rate (bps)	2400	4800	9600	19K2	38K4	57K6	76K8	115K2	230K4	460K8

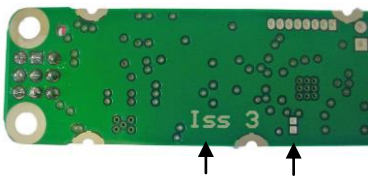
BAUD can only take the values shown above. If set <0 or >9, the baud rate defaults to 9600

Warning: When changing this setting it is possible to lose communication with the device. As well as keeping track of the correct baud rate, it is also essential in this case to be sure that your hardware supports the rate you are changing to.
The evaluation kit supports all possible DCell/DSC baud rate settings UPTO 115200

ReSet: (New For Version 4 (DSC Only))

- With version 3 pcb and higher. A reset to comms defaults is available. This will return Station No. to 1 and baudrate to 115200. This is achieved by powering off the DSC shorting the 2 pads together whilst powering on.

DSC Issue 3 pcb or Higher



Ensure issue 3 or Higher 2 unmasked pads

The new value of BAUD does not take effect until the RST command is issued or the device is power cycled.

To Change The Baud Rate, follow a similar sequence to changing the STN value
1. First set **BAUD** to the new value

2. Send a **RST** command or power cycle the device.
3. Change the device settings in Instrument Explorer by selecting **Change Settings** from the **Communications** menu. Select new baud rate.

Communications Failure Count, CFCT

The Communications Failure Count, **CFCT**, can be used to indicate the quality off the communications bus. Each time there is a UART framing error or a data overrun error the count is incremented. This is a read/ Write parameter so resetting is by writing the value 0. This value is not saved on power fail and is reset to zero on power up.

Output Format Controls, DP and DPB (ASCII ONLY)

The parameters **DP** and **DPB** are used to control the formatting of floating-point values in the ASCII protocol.

DP controls the number of decimal places after the point and **DPB** controls the number of decimal places before the point. Values of 1..8 are appropriate in both cases.
All output values are then transmitted in this same format. As values are limited to a normal 4-byte accuracy (about 7 digits), it may sometimes be necessary to alter the formatting for best accuracy in reading/writing values.

eg. if DP=5 and DPB=2, the value 1.257 is output as "+01.25700"

The new value of DP and DPB does not take effect until the RST command is issued or the device is power cycled.

It is important to ensure that when settings DpB the maximum number expected to be formatted does not exceed the number of integer digits. i.e. DpB = 3 cannot display numbers greater than or equal to 1000. Failure to observe this will lead to non-numeric ASDCII codes after the polarity character.

Information

Reports the current version of the devices software and the devices unique serial number. Note that VERSION is the read able item derived from the devices internal value of **VER** and SerialNumber is derived from **SERL** and **SERH**.

Software Version, VER

The **VER** parameter (read-only byte) returns a value identifying the software release number, coded as 256*(major-release)+(minor-release)

eg. current version 3.1 returns VER=769

Serial Number, SERL and SERH

SERL and **SERH** are read-only integer parameters returning the device serial-number.
This is decoded as = 65536***SERH** + **SERL**.

Strain Gauge

This is where the measurement process starts. If the optional temperature module is fitted then **TEMP** will display actual temperature in Degree C. Otherwise **TEMP** will display 125 Degree C.
RATE is the parameter that selects measurement cycle update rate.

mV/V output, MVV

MVV is the factory calibrated mV/V output and it is this value that all other measurement output values are derived from. Factory calibration is within 0.05%.

Nominal mV/V level, NMVV

This is used to represent the nominal mV/V value representing 100% of full scale. This value is used solely for the generation of **ELEC**. It is factory set for 2.5mV/V. If the electronic gain is adjusted by changing the gain resistor then if **ELEC** is used **NMVV** value must be changed to represent the new nominal mV/V.

mV/V Output In Percentage Terms, ELEC

This is mainly for backwards compatibility with Version 2. It is the mV/V value represented in percentage terms, 100% being the value set by **NMVV.**

Temperature Value, TEMP

If the optional temperature module is fitted, DTEMP then **TEMP** will display actual temperature in Degree C. Otherwise **TEMP** will display 125 Degree C. **TEMP** is used by the temperature compensation. See chapter 5

Output Rate Control, RATE

The **RATE** parameter is used to select the output update rate, according to the following table of values –

RATE value	0	1	2	3	4	5	6	7	8	9	10
update rate (readings per second)	1	2	5	10	20	50	60	100	200	300	500

The default rate is 10Hz (**RATE**=3): The other settings give a different speed/accuracy trade-off.
Invalid **RATE** values are treated as if it was set to 3.

The underlying analogue to digital conversion rate is 4.8Khz. These results are block averaged to produce the required output rate.

To Change The Output Rate
1. Set **RATE** to the new value
2. Click on the '**RST**' button to reboot the device
3. Wait for one second for the reset procedure to complete and measure cycle to start

With **RATE** set to 0, you should be able to see the **SYS** update rate slow to once a second, and the noise level should also noticeably decrease.

All the main-reading output values are updated at this rate. Rate does not change the rate at which temperature output **TEMP** is updated.

Dynamic Filtering, FFST and FFLV

The Dynamic filter is basically a recursive filter and therefore behaves like an "RC" circuit. It has two user settings, a level set in mV/V by **FFLV** and a maximum number of steps set by **FFST**, maximum value **FFST** can be is 255. If a difference between a new input value (**RMVV**) and the current filtered value (**MVV**) is greater than **FFLV** then the fractional amount of the new reading added to the current reading is reset to 1, that is to say that output of the filter will be equal to the new input reading. If the difference is less than **FFLV** then the fractional amount added is

incremented until it reaches the maximum level set by **FFST**.IE if **FFST** = 10 then after a step change the fractional part of a new reading is incremented as follows
1/1, 1/2, 1/3, 1/4, 1/5, 1/6…. 1/10, 1/10, 1/10
This allows the Filter to respond rapidly to a fast moving input signal.

With a step change, which does not exceed **FFLV**, the calculated new filtered value can be calculated as follows

New Filter Output value = Current Filter Output Value + ((Input Value - Current Filter Output Value) / FFST)

The time taken to reach 63% of a step change input (which is less than **FFLV**) is the frequency at which values are passed to the dynamic filter, set in **RATE**, multiplied by **FFST**.

The table below gives an indication of the response to a step input less than **FFLV**.
Update Rate is 1/table value of **RATE** see Chapter 3 Output Rate Control.

% Of Final Value	**Time To settle**
63%	Update Rate * **FFST**
99%	Update Rate * **FFST** * 5
99.9%	Update Rate * **FFST** * 7

For example, If **RATE** is set to 7 = 100Hz = 0.01s and **FFST** is set to 30 then the time taken to reach a % of step change value is as follows.

% Of Final Value	**Time To settle**
63%	0.01 x 30 = 0.3 seconds
1%	0.01 x 30 x 5 = 1.5 seconds
0.1%	0.01 x 30 x 7 = 2.1 seconds

The following table shows the number of updates x FFST and the error % **New Filter Output value** will differ from a constant **Input Value**.

x FFST	% Error
1	36.78794412
2	13.53352832
3	4.97870684
4	1.83156389
5	0.67379470
6	0.24787522
7	0.09118820
8	0.03354626
9	0.01234098
10	0.00453999
11	0.00167017
12	0.00061442
13	0.00022603
14	0.00008315
15	0.00003059
16	0.00001125
17	0.00000414
18	0.00000152
19	0.00000056
20	0.00000021

Remember that if the step change in mV/V is greater than the value set in **FFLV** then
New Filter Output value = New Input Value.

And the internal working value of **FFST** is reset to 1, being incremented each update set by **RATE** until it reaches the user set value of **FFST**.

Cell

Provides the level where the integration between the DCell/DSC and the strain gauge bridge takes place. Features include, when the optional temperature module is fitted, 5-point temperature compensation to produce a temperature compensated value **CMVV**. Scaling using a gain and offset, **CGAI** and **COFS** respectively, producing a value known as **CRAW**. Linearisation, using up to 7-points, producing the final output from this section known as **CELL**. Over load and under load values can be set in **CMIN** & **CMAX** to alert the user of forces less or greater than the integrator has intended the unit to be operated. These features allow the output **CELL** to be in force units which can be used by 'System' to convert to units of weight.

Temperature compensation and Linearisation are covered in detail in their own chapters.

Temperature Compensation In Brief

When the optional temperature hardware module DTEMP is connected the temperature compensation is available. The temperature compensation facility can remove the need for the fitting of compensation resistors to the strain gauges. This compensation can apply for gain and offset with up to 5 temperature points.
The input for the temperature compensation is **MVV** and the output from the process is **CMVV**. If not temperature compensation is invoked the **CMVV** is equal to **MVV**

A Detailed explanation is given in chapter 5

Cell Scaling, CGAI, COFS

The temperature compensated value **CMVV** is scaled with gain and offset using **CGAI** and **COFS** respectively. The gain is applied first and the offset then subtracted. This would be used to give a force output in the chosen units, this output being termed **CRAW**.

CRAW = (CMVV X **CGAI) – COFS**

Two Point Calibration Calculations and Examples

Examples are given here for two point calibration, as this is by far the most common method.

Cell Calibration
The scaling parameters are **CGAI** and **COFS**
CGAI is in cell-units per mV/V'
COFS is in cell units

The cell output calculation is (in the absence of temperature and linearity corrections) –
CRAW = (**CMVV** × **CGAI**) – **COFS**
If we have two electrical-output (**MVV**) readings for two known force loads, we can convert the output to the required range. So if –

test load = fA → **CMVV** reading = cA
test load = fB → **CMVV** reading = cB

– then calculate the following gain value

CGAI = (fB – fA) / (cB – cA)

and the offset is

COFS = (cA x **CGAI**) – fA

The outputs should then be **CELL** = fA, fB true force values, as required.

Calibration Methods

There are a number of ways of establishing the correct control values.

Method 1 - Nominal (data sheet) Performance Values

This is the simplest method, where the given nominal mV/V sensor output is used to calculate an approximate value for **CGAI**.

Example.

A 50 kN load cell has nominal sensitivity of 2.2mV/V full-scale.

So to get 50.0 for an input of 2.2mV/V, we set CGAI to 50/2.2≈22.7273. This assumes the output for 0kN is 0mV/V.

Method 2 - Device Standard (Calibration) Values
With some load cells you may have a manufacturer's calibration document. This gives precise cell-output gain and offset specifications for the individual cell. These values can be used to set the SGAI and SOFS values to be used.

Example.

A 10 tonne load cell has a calibration sheet specifying 2.19053mV/V full-scale output, and -0.01573mV/V output offset.
CGAI is set to 10 / (2.19053- -0.01573) ≈ 4.532557.

COFS is set to -0.01573 x 4.532557≈ -0.0071297

NOTE:
Methods 1 and 2 require no load tests. This means that systematic installation errors cannot be removed, such as cells not being mounted exactly vertical. The accuracy is also limited by the DCell/DSC electrical calibration accuracy, which is about 0.1%.
The remaining methods require testing with known loads, but are therefore inherently more reliable in practice, as they can remove unexpected complicating factors relating to installation.

Method 3 - Two-Point Calibration Method
This is a simple in-system calibration procedure, and probably the commonest method in practice (as in the previous example).
Two known loads are applied to the system, and reading results noted, then calibration parameters are set to provide exactly correct readings for these two conditions.

eg. a 10kN (1-tonne) load cell has a CELL reading of +0.120721mV/V with no load, and -2.21854mV/V with a known 100Kg test-weight.
To calibrate this to read in a -1.0 to +1.0 tonne range,

Calculate CGAI as 0.1 / (2.21854 - +0.120721) = 0.047669.

Set COFS= 0.120721 x 0.047669 = 0.005755.

Method 4 - Multi-point Calibration Test
For ultimate accuracy to a whole series of point measurements may be taken to determine the best linear scaling of input output: Effectively, a 'best line' through the data is then chosen, and the calibration is set up to follow the line.
Testing of this sort is also used to establish linearity corrections, and similar tests at different temperatures are used to set up temperature compensation (see Chapters on Temperature Compensation *and* Linearity *Compensation*).

Note: Instrument Explorer provides "wizards" for easy calibration of the Cell stage. There are two wizards, 'Cell Calibration Auto' and 'Cell Calibration Table' these can be found under the menu item "Wizards".

Cell Limits, CMIN, CMAX

These are used to indicate that the desired maximum and minimum value of **CRAW** have been exceeded. They are set in Force units. On **CRAW** being greater than the value set in **CMAX** the **CRAWOR** flag is set in both **FLAG** and **STAT**, the value of **CRAW** is also clamped to this value. On **CRAW** being less than the value set in **CMIN** the **CRAWUR** flag is set in both **FLAG** and **STAT**, the value of **CRAW** is also clamped to this value.

Linearisation In Brief

Linearisation allows for any non-linearity in the strain gauge measurement to be removed. Up to 7 points can be set using **CLN**. The principle of operation is that the table holds a value at which an offset is added. The point in the table that refer to **CRAW** are named **CLX1..CLX7**. The offsets added at these point are named **CLK1.. CLK7** and are set in thousandths of a cell unit. The output from the Linearisation function is **CELL**. If no Linearisation is used (**CLN** < 2) the **CELL** is equal to **CRAW**.

A Detailed explanation is given in chapter 6

System

System is where the "Force" output, **CELL**, is converted to weight when installed into a system. Other features such as **SZ** offers a means of zeroing the system output **SYS**. Peak and Trough values are also recorded against the value of **SYS,** these are volatile and reset on power up. A command **SNAP** records the next **SYS** value and stores in **SYSN**, this is useful where more than 1 device in a system and to prevent measurement skew across the system the **SNAP** command can be broadcast to all devices ready for polling of their individual **SYSN** values.

System Scaling, SGAI, SOFS

The cell output value **CELL** is scaled with gain and offset using **SGAI** and **SOFS** respectively. The gain is applied first and the offset the subtracted. This would be used to give a force output in the chosen units, this output being termed **SRAW**.

SRAW = (CELL X **SGAI) - SOFS**

If we have two cell-output (CELL) readings for two known test loads, we can convert the output to the required range. So if –
Test load = xA → CELL reading = cA
Test load = xB → CELL reading = cB

Then we calculate the following gain value
SGAI = (xB – xA) / (cB – cA)
And then the offset
SOFS = (cA x SGAI) - xA

The outputs should now be SRAW = xA, xB true load values, as required.

Example of calculations for SGAI and SOFS

Example:

A 2500Kgf load cell installation is to be calibrated by means of test weights.

The cell calibration gives an output in Kgf ranging 0–2000.

A system calibration is required to give an output reading in the range 0–1.0 tonnes.

Calculations

Take readings with two known applied loads, such as –

For test load of xA = 99.88Kg : CELL reading cA = 100.0112

For test load of xB = 500.07Kg: CELL reading cB = 498.7735

Calculate gain value. In this case put SGAI = (xB - xA) / (cB - cA)

= (0.50007 - 0.09988) / (498.7735 - 100.0112)

≈ 0.001003580 = 1.003580×10^{-3}

Calculate offset value. In this case SOFS = (cA x SGAI) – xA

= (100.0112 x 1.003580x10^{-3}) – 0.09988

≈ 0.00048924

Check

Putting the values back into the equation, results for the two test loads should then be —

For x = 99.88Kg, CELL = 100.0112, so

SRAW ≈ (100.0112 × 1.003580x10$^{-3)}$ – 0.00048924 ≈ 0.09988

For x = 500.07Kg, CELL = 498.7735, so

SRAW ≈ (498.7735 × 1.003580x10^{-3}) – 0.00048924 ≈ 0.5006987

The remaining errors are due to rounding the parameters to 7 figures.
Internal parameter storage is only accurate to about 7 figures, so errors of about this size can be expected in practice.

System Limits, SMIN, SMAX

These are used to indicate that the desired maximum and minimum value of **SRAW** have been exceeded. They are set in weight units. On **SRAW** being greater than the value set in **SMAX** the **SRAWOR** flag is set in both **FLAG** and **STAT**, the value of **SRAW** is also clamped to this value. On **SRAW** being less than the value set in **SMIN** the **SRAWUR** flag is set in both **FLAG** and **STAT**, the value of **SRAW** is also clamped to this value.

System Zero, SZ

SZ provides a means of applying a zero to **SYS and SOUT.** This could be used to generate an Net value making **SRAW** in effect a gross value.

SYS = SRAW – SZ

Care should be taken on how often **SZ** is written to, see "WARNING: Finite Non-Volatile Memory Life" later in this chapter.

System Outputs, SYS, SOUT

SYS is considered to be the main output value and it is this value that would be mainly used by the master. **SOUT** is for backwards compatibility with Version 2

Reading Snapshot, SNAP, SYSN

The action command **SNAP** samples the selected output by copying **SYS** to the special result parameter **SYSN**.
The main use of this is where a number of different inputs need to be sampled at the same instant.
Normally, multiple readings are staggered in time because of the need to read back results from separate devices in sequence: By broadcasting a **SNAP** command at the required time, all devices on the bus will sample their inputs within a few milliseconds. The resulting values can then be read back in the normal way from all the devices **SYSN** parameters.

Note: Instrument Explorer provides "wizards" for easy calibration of the System stage. There are two wizards, 'Sys Calibration Auto' and 'Sys Calibration Table' these can be found under the menu item "Wizards".

Control

Shunt Calibration Commands, SCON and SCOF

The Device is fitted with a "Shunt" calibration resistor whose value is 100K.This can be switched across the bridge, using **SCON**, giving an approximate change of 0.8mV/V at nominal 2.5mV/V. The command **SCOF** removes the resistor from across the bridge. It is important for the user to remember to switch out the shunt calibration resistor after calibration has been confirmed.

Digital Output, OPON and OPOF

For DSC **ONLY** an open collector output is available. This can be switched on using **OPON** and off by the command **OPOF**. This output is capable of switching 100mA at 30v (TBC)

Flags

Diagnostics Flags, FLAG and STAT

All the self-diagnostics rely on the **FLAG** & **STAT** parameters, which are 16-bit integer register in which different bits of the value represent different diagnostic warnings. **FLAG** is stored in EEPROM and is therefore non-volatile, **STAT** is stored in RAM and reset on power-up to 0. **FLAG** is latching requiring reset by the user where as **STAT** is non-latching showing current error status.

Latched Warning Flags (FLAG)

The flags are normally used as follows:-
FLAG is read at regular intervals by the host (like the main output value, but generally at longer intervals)
If some warnings are active, i.e. **FLAG** is non-zero, then the host tries to cancel the warnings found by writing **FLAG**= 0
The host then notes whether the error then either remains (i.e. couldn't be cancelled), or if it disappears, or if it re-occurs within a short time, and will take action accordingly.

The warning flags are generally latched indicators of transient error events: By resetting the register, the host both signals that it has seen the warning, and readies the system to detect any re-occurrence (i.e. it resets the latch).

What the host should actually do with warnings depends on the type and the application: Sometimes a complete log is kept, sometimes no checking at all is needed.
Often, some warnings can be ignored unless they recur within a short time.

Warning flags survive power-down, i.e. they are backed up in non-volatile (EEPROM) storage.
Though useful, this means that repeatedly cancelling errors which then shortly recur can wear out the device non-volatile storage – see Chapter 3 Basic Set-up and Calibration.

Meaning and Operation of Flags

The various bits in the FLAG value are as follows

Bit	Value	Description	Name
0	1	(unused – reserved)	Unused
1	2	(unused – reserved)	Unused
2	4	Temperature under range (TEMP)	TEMPUR
3	8	Temperature over-range (TEMP	TEMPOR
4	16	Strain gauge input under-range	ECOMUR
5	32	Strain gauge input over-range	ECOMOR
6	64	Cell under-range (CRAW)	CRAWUR
7	128	Cell over-range (CRAW)	CRAWOR
8	256	System under-range (SRAW)	SYSUR
9	512	System over-range (SRAW)	SYSOR
10	1024	(unused – reserved)	Unused
11	2048	Load Cell Integrity Error (LCINTEG)	LCINTEG
12	4096	Watchdog Reset	WDRST

11	8192	(unused – reserved)	Unused
14	16384	Brown-Out Reset	BRWNOUT
15	32768	Reboot warning (Normal Power up)	REBOOT

NOTE:
The mnemonic names are used by convenience properties in Instrument Explorer, but are otherwise for reference only –the flags can only be accessed via the FLAG parameter.

The various warning flags have the following meanings

TEMPUR and TEMPOR indicate temperature under-and over-range. The temperature minimum and maximum settings are part of the temperature calibration, fixed at –50.0 and +90.0 ºC. Only active when optional Temperature module fitted.

ECOMUR and ECOMOR are the basic electrical output range warnings. These are tripped when the electrical reading goes outside fixed ±120% limits: This indicates a possible overload of the input circuitry, i.e. the input is too big to measure.
The tested value, ECOM is an un-filtered precursor of ELEC

CRAWUR and CRAWOR are the cell output range warnings. These are tripped when the cell value goes outside programmable limits CMIN or CMAX.
The tested value, CRAW is the cell output prior to linearity compensation.

SYSUR and SYSOR are the system output range warnings. These are triggered if the SYS value goes outside the SMIN or SMAX limits.

LCINTEG indicates a missing or a problem with the Load cell. It is based on the common mode of the –SIG being correct. NOTE This flag will also be set when the shunt calibration has been switched on.

WDRST indicates that the Watchdog has caused the device to re-boot. If this error continually occurs consult factory.

BRWNOUT indicates that the device has re-booted due to the supply voltage falling below 4.1V, the minimum spec for supply voltage is 5.6V and this must include any troughs in the AC element of this supply.

REBOOT is set whenever the DCell/DSC is powered up and is normal for a power up condition. This flag can be used to warn of power loss to device.

Dynamic Status Flags (STAT)

Status are "live" flags, indicating current status of the device. Some of these flags have the same bit value & description from "FLAG".

Meaning and Operation of Flags

The various bits in the STAT value are as follows

Bit	Value	Description	Name
0	1	Setpoint output status	SPSTAT
1	2	Digital Input status (DSC ONLY)	IPSTAT
2	4	Temperature under range (TEMP)	TEMPUR
3	8	Temperature over-range (TEMP	TEMPOR
4	16	Strain gauge input under-range	ECOMUR
5	32	Strain gauge input over-range	ECOMOR
6	64	Cell under-range (CRAW)	CRAWUR
7	128	Cell over-range (CRAW)	CRAWOR
8	256	System under-range (SRAW)	SYSUR
9	512	System over-range (SRAW)	SYSOR
10	1024	(unused – reserved)	Unused
11	2048	Load Cell Integrity Error (LCINTEG)	LCINTEG
12	4096	Shunt Calibration Resistor ON	SCALON

11	8192	Stale output value (previously read)	OLDVAL
14	16384	(unused – reserved)	Unused
15	32768	(unused – reserved)	Unused

SPSTAT indicates the state of the Open collector output, 1 being output on 0 being output off.

IPSTAT indicates the state of the digital input (Only available on DSC model). Bit set indicates input is 'closed' to 0v (-V or GND).

SCALON Used to indicate that the Shunt Calibration command, **SCON**, has been issued & therefore the shuntcal resistor is now in circuit with the strain gauge bridge. **SCOF** command resets this bit. Note that when Shunt Calibration is active the "Load Cell Integrity Error" will also be generated.

OLDVAL is set when the device is read via the communications. Thus indicating this value has already been sampled. It is reset when a new result has been made available.

Output Update Tracking

The **OLDVAL** flag can be used for output update tracking
This allows sampling each result exactly once: To achieve this poll the **STAT** value until **OLDVAL** is cleared to indicate a new output is ready, then read **SYS**, this reading will set the **OLDVAL** flag in **STAT.**

This scheme works as long as the communications speed is fast enough to keep up. With faster update rates and slower baud rates, it may not be possible to read out the data fast enough.

User Storage

USR1…USR9

There are nine storage locations **USR1** to **USR9**. These are floating point numbers which can be used for storage of data. This data could be calibration time and date, operator number, customer number etc.
This data is not used in anyway by the DCell or DSC.

Reset

The Reset command, RST

This command is used to reset the device. This command **MUST** be issued if the following parameters are changed before the change will take effect. Alternatively the power maybe cycled.

RATE, STN, BAUD, DP & DPB.

The reset action may take up to about a second to take effect, followed by the normal start-up pause of 1 second.

WARNING: Finite Non-Volatile Memory Life

The DCell and DSC use EEPROM-type memory as the storage for non-volatile controls (i.e. all the settings that are retained even when powered down).
The device EEPROM itself is specified for 100,000 write cycles (for any one storage location), although typically this is 1,000,000. Therefore –

When automatic procedures may write to stored control parameters, it is important to make sure this does not happen too frequently.

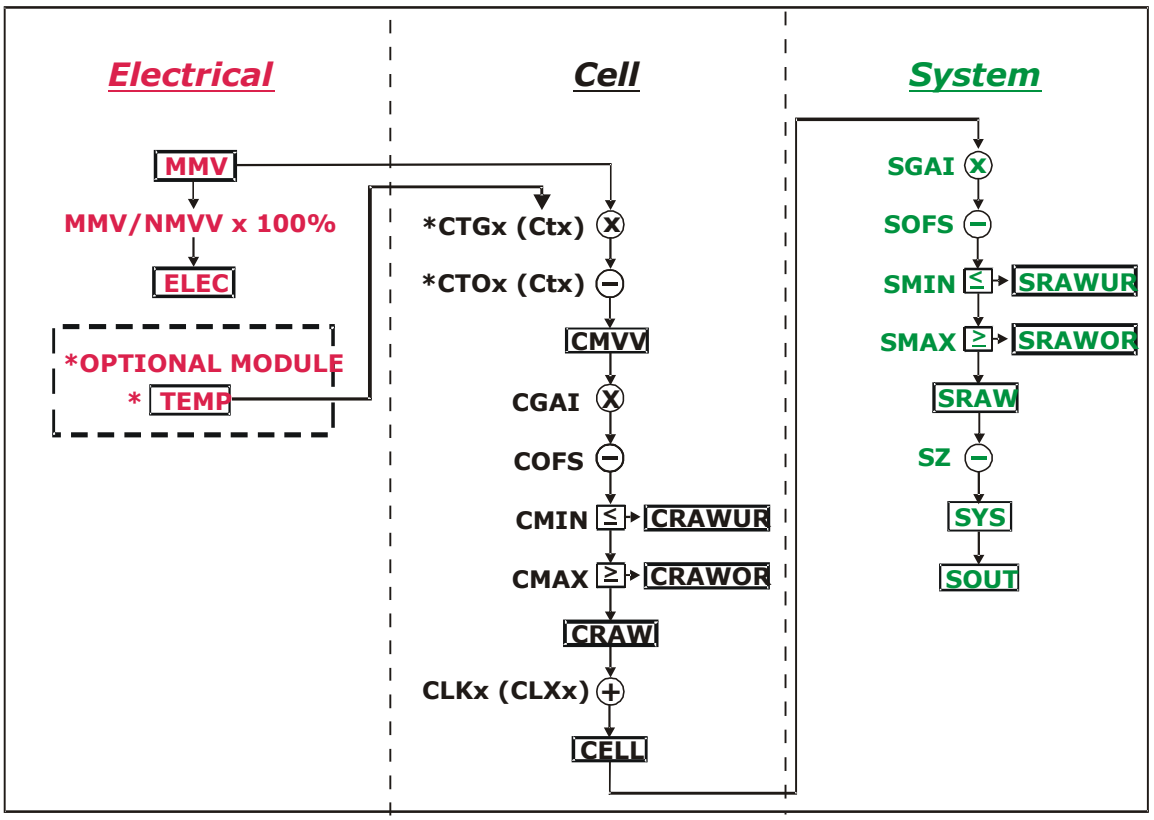
So you should not, for example, *on a regular basis* adjust an offset calibration parameter to zero the output value. However, it *is* reasonable to use this if the zeroing process is initiated by the operator, and won't normally be used repeatedly.

For the same reason, automatically cancelling warning flags must also be implemented with caution: It is okay as long as you are not getting an error recurring *repeatedly*, and resetting it every few seconds.

Chapter 4 The Readings Process

This chapter gives an account of the reading process *except* for the linearity-and temperature-compensation processes (which have their own chapters later on).

Flow diagram



The underlying analogue to digital conversion rate is 4.8Khz. These results are block averaged to produce the required output rate set by the **RATE** control This block averaged result is then passed through the dynamic filter at the same rate and then into the 'chain' of above calculations.

The named values shown in the boxes are all output parameters, which can be read back over the comms link.
The diagram shows ***three separate calibration stages***, called the 'Electrical', 'Cell' and 'System'.
This allows independent calibrations to be stored for the device itself, the load cell and the installed system characteristics –

Electrical
The 'Electrical' calibration produces corrected electrical readings from the internal measurements.
This is factory-set by Mantracourt during the production process.

The main outputs from this are –

- **MVV** is the factory calibrated output, in mV/V units.
- **ELEC** is the mV/V in % terms. Where the 100% value is set using **NMMV**. This is for backwards compatibility only.
- **TEMP** is a device temperature measurement, in °C and requires an optional module.

There are also two flags, ECOMUR and ECOMOR (not shown on the diagram), which indicate an input electrical under- or over-range.

Cell
The 'Cell' calibration converts the mV/V output into a cell-force reading.
This can be used by an OEM sensor manufacturer to provide a standard, calibrated output in force units, which could be based on either typical or device-specific calibration data.
(This stage also includes the temperature- and linearity-corrections, not covered here)

The outputs from this are

- **CMVV** is the temperature compensated mV/V (**MVV**).
- **CRAW** is the scaled temperature compensated value **CMVV**
- **CELL** is a load cell force reading in "Force" units (e.g. kN)
- **CRAWUR** and **CRAWOR** are two flags indicating under or, over range for the force measurement.

System
The 'System' calibration converts the Cell output into a final output value, in the required engineering units.
This is normally be set up by a systems installer or end user, to provide whatever kind of output is needed, independently of device-specific information in the Cell calibration.
(Making this split allows in-service replacement without re calibration).

The Outputs From This Are

- **SRAW** is a re-scaled and offset adjusted output derived from **CELL**
- **SYS** is the final output value, after removing a final user output offset value (**SZ**) from **SRAW**
- **SRAWUR** and **SRAWOR** are output warning limit flags.

In practice, **SRAW** and **SYS** can be used to represent something like gross and net values.

Cell and System Scaling

Both the Cell and System calibrations are simply linear rescaling calculations –i.e. they apply a gain and offset.

In both cases, four parameters define the scaling, offset and min and max limit values. These calculations are applied in the following way:
Output = (Input × **GAI**) – **OFS**
Output = min (output, **MAX**)
Output = max (output, **MIN**)
(In addition, if the value exceeds either limit, one of two dedicated error flags is set)

The control parameters thus have the following characteristics: –

- **GAI** is the multiplying factor, set in "output-units per input-unit"
- **OFS** is the value that gives zero output, set in "output units"
- **MAX** and MIN are output limit values, set in "output units"

The units and functions of the main scaling controls can thus be summarised as –

Cell Calibration

CGAI	Force/mV/V	mV/V gain factor
COFS	Force	**CELL** Offset Value
CMIN	Force	Minimum value for **CRAW**
CMAX	Force	Maximum value for **CRAW**

System Calibration

SGAI	Eng/ Force	**SYS**/**CELL** gain factor
SOFS	Eng	**SRAW** value offset
SMIN	Eng	Minimum value for **SRAW**
SMAX	Eng	Maximum value for **SRAW**
SZ	Eng	**SYS** value offset

(MVV is mV/V, "force" is force units, and "eng" is engineering units)

Calibration Parameters Summary and Defaults

The various control parameters are listed for each stage.
This also includes the compensation parameters, not covered in this chapter, but shown in the flow diagram
The 'default' values shown set the device back to its nominal default calibration (mV/V)

Cell Control Defaults

Command	Action	Default Values
FFLV	Filter dynamic level	0.001
FFST	Filter Steps (max)	100
NMMV	Nominal 2.5mV/V	2.5
RATE	Rate 10Hz	3
CGAI	basic cell gain	1.0
COFS	basic cell offset	0.0
CTN	number of temp points	0
CT1..5	temp points (Deg C)	0.0, 0.0, 0,0…
CTO1..5	offset adjusts	0.0, 0.0, 0,0…
CTG1..5	gain adjusts	1.0, 1.0, 1.0, 1.0…
CMIN	**CRAW** min limit	-3.0
CMAX	**CRAW** max limit	+3.0
CLN	number of linearity points	0
CLX1..7	linearity raw-value points	0.0, 0.0, 0,0…
CLK1..7	linearity adjusts	0.0, 0.0, 0,0…

System Control Defaults

Command	Action	Default Values
SGAI	basic gain	1.0
SOFS	basic offset	0.0
SMIN	**SRAW** min limit	-100.0
SMAX	**SRAW** max limit	+100.0
SZ	output zero offset	0.0

Chapter 5 Temperature Compensation

This chapter explains how to use the Temperature Compensation facilities, to compensate for changes in the measurement with ambient temperature.

Temperature compensation is only provided when an optional module consisting of a digital temperature sensor is wired to the DCell or DSC.

Purpose and Method of Temperature Compensation

Most measurement methods are affected by changes in temperature, and (uncompensated) load cells are especially sensitive, having a large overall temperature coefficient.

Temperature compensation adjusts the measured value in a way that depends on a temperature measurement, so that (ideally) the output does not depend on the current temperature.
In practice, it is usual to refer to a calibration 'reference' temperature: The ideal output value is then what the reading 'would have been' if made at the reference temperature.

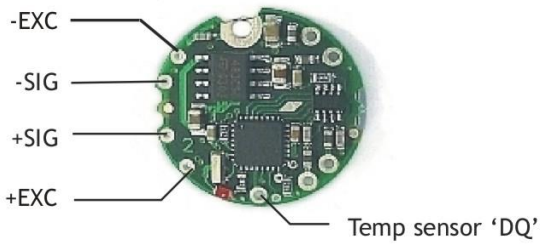
The DCell/DSC temperature compensation facilities make adjustments to the 'Cell' calibration parameters (i.e. gain and offset) which depend on temperature, according to a digitally programmed curve.
These adjustments are automatically applied, based on the current device temperature measurement. With some care, this can remove the need for the usual electrical compensation components altogether.

Note that the temperature compensation will also remove the temperature drift of the DCell/DSC itself if the temperature compensation data is collected when the DCell/DSC and strain gauges are tested together as a system.

Temperature Module Connections and Mounting (DTEMP)

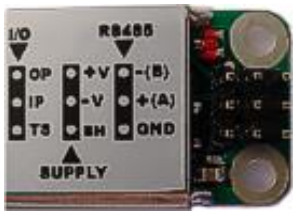
The temperature module is a connected using only two wires. The temperature sensor is the Dallas "1-Wire" digital device DS18S20.

DCELL connections to DTEMP



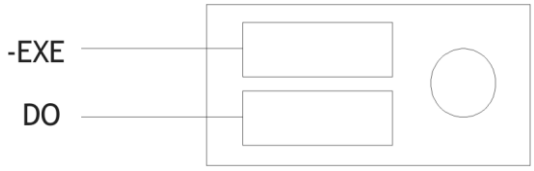
One connection is ground for which –EXC is used and the other is the 1 wire "DQ" connection which provides the bi-directional data line.

DSC connections to DTEMP



For the DSC "TS" on the I/O connector is connected to "DQ" of the DTEMP module and "GND" of the DSC is connected to "-EXC"

The temperature module is a small double sided PCB with an 8 pin SOIC integrated circuit mounted to it. The dimensions are 10.5 x 7.6 x 2.5mm. There are two solder pads for connection to the DSC or DCell. A 2mm hole is used for fixing the temperature module to the body of the load cell. The module should, ideally, be positioned as close as possible to the strain gauges. The IC on the temperature module must also be in good thermal contact to the load cell body so the strain gauges and temperature sensor see the same temperature.



Control Parameters

The temperature compensation parameters define a pair of lookup tables that contain adjustments to the cell calibration gain and offset over temperature.

The parameters concerned are the following

CTN	[-]	Number of temperature table points
CT1…CT5	[°C]	Indicated TEMP value at table point
CTO1…CTO5	mV/V	MV/V ($x10^4$) Offset adjustment at Offset table point
CTG1…CTG5	ppm	Gain ($x10^{6)}$ at Gain table point

CTN sets the number of points in the gain & offset table's. A **CTN** value of less than two effectively switches off temperature compensation. The maximum number of point is 5, values greater than 5 reset **CTN** to 0 switching off temperature compensation.

CT1 to **CT5** sets the temperature in degree C of the correction points. The table must be filled from **CT1** up to & including CT(**CTN**) and must be entered in order of increasing temperature value.

CTO1 to **CTO5** provide the offset adjustment in mV/V $x10^4$. The reason for the multiplication is due to the limitation set by the ASCII protocol of only being able to enter up to 6 decimal places. The **CTOx** value is **subtracted** from the uncompensated value.

CTG1 to **CTG5** provide the gain adjustment in ppm terms. The actual gain value used is calculated as
1 + **CTGn** x 10^{-6}

Internal Calculation

The temperature compensation calculation is described as follows:-

The GAIN correction is applied first. The current measured temperature is checked against the table values CT1 to CT(CTN) to establish an index value if the measured temperature is below that of CT1 then CT1 is used as the index, If the temperature is above CT(CTN-1) then CT(CTN-1) is used.

This can be represented as follows:-
A working table index, i, is derived from the current measured temperature, T, as follows —

(n = number of points used, as set by CTN)

When	(T < CT1)	then i = 1
When	(T > CTn-1)	then i = (n–1)
Otherwise		i is chosen so that $Ti \leq T \leq Ti+1$

Once an index into the table has been established the gain value to be used is extrapolated between the index value and the value above. If the temperature is above CT(CTN) or below CT1 then the calculated temperature gain value is extrapolated from CT(CTN-1) to CT(CTN) or CT1 to CT2 respectively.

This can be represented mathematically as follows:-
CALC_CTG = CTGi + (CTGi+1 – CTGi) × (T – CTi) / (CTi+1 – CTi)

The actual gain value used is 1 + CALC_CTGO x 10-6 and is multiplied by the uncompensated value MVV.

The offset correction is then applied. Using the same temperature index "i" as found for the GAIN index above.

The Offset value is extrapolated between the same two temperature points.

This can be represented mathematically as follows:-

$CALC_CTO = CTO_i + (CTO_{i+1} - CTO_i) \times (T - CT_i) / (CT_{i+1} - CT_i)$

The actual offset value used is CALC_CTO x 10^{-4} and is subtracted from the above gain adjusted value.

The output from the temperature compensation (CMVV) is then calculated as
CMVV = MVV x (1 + CALC_CTGO x 10^{-6}) - (CALC_CTO x 10^{-4})

The Temperature Measurement

The temperature sensor used is a Dallas (MAXIM) DS18S20 Digital Thermometer using the "1-Wire" bus technology. This gives a temperature measurement accuracy of +/-0.5 Degree C over the temperature range –10 to +85 degree C and +/-2.0 Degree C over temp range –55 to +125 Degree C. The resolution of the measurement is 0.0625 Degree C. The temperature is sampled and the **TEMP** variable updated every 5 seconds.

How to Set Up a Temperature Compensation

There are a number of ways of obtaining a temperature compensation curve.
The best possible compensation for a given piece of physical hardware can only be achieved by performing experiments on that particular unit (DCell/DSC and associated strain gauges), to characterise the measurement output at a variety of different, stable temperatures in the required operating range.
The basic choice of methods depends on trading off ideal accuracy against the complexity of the calibration procedure.

Method 1
Apply a simple linear drift correction (i.e. for known constant gain and offset changes per degree), by specifying zero correction at the calibration temperature, and appropriately adjusted correction values at extreme temperatures above and below this.
This can be used when the measurement or sensor has known temperature coefficients.

Method 2
Where the temperature characteristics of the measurement are known, but not linear, a similar scheme to Method 1 can be used, with a multi-point table defining an approximation to the known, ideal temperature curves of offset and gain variations.

NOTE: Both of the above methods are based on 'known' characteristics, which could come from datasheets but these methods would not compensate for the DCell/DSC.

Method 3
Do a series of measurements at different temperatures and install the appropriate correction values to give exactly correct results at those same temperatures –i.e. calculate ideal gain and offset corrections at the tested temperatures.
(*This method is the most common)*. There is a 'wizard' available in Instrument Explorer which will enable this method to be easily completed by calculating the gain and offset corrections for you.

Method 4
Use a set of test results to plan a 'best correction' curve (not necessarily perfect at test temperatures, but slightly better overall).

NOTES:
All of these methods can be applied *either* to data from individual devices *or* to an 'average' correction for a particular type of sensor hardware.
During testing, temperatures should be measured using the internal TEMP measurement, as this is the measurement used to do the corrections.
For in-system tests, the environment of the DCell/DSC must always be as near as possible to the exact conditions of the eventual in-system use.

Parameter Calculations

Instrument Explorer provides a "Wizard" for the calculation of the parameters required by the DSC/DCell. This is based on Method 3 where data is collected. The wizard allows for small changes in the sampled temperature point that may occur when taking a set of results for gain and offset. Also taken into account is any variation in the test weights at different temperatures. This is a complex mathematical procedure which is best solved by a PC programme such as the wizard.

Chapter 6 Linearity Compensation

This chapter describes the Linearity Compensation features and how to use them.

Purpose and Method of Linearisation

Load cell sensor outputs are never precisely proportional to the input (applied load).
If the graph of the measurement output against the true value shows slight deviations from the ideal straight-line, then slight errors remain even when the basic calibration (offset and gain) is as good as possible.
Linearity compensation adjusts the raw measurement by a small amount that is calculated as a function of the raw measurement value itself. Ideally this will adjust the output response, for any given input load, by exactly the right amount to place the final result onto the ideal straight line.

The DCell/DSC non-linearity compensation uses a single 'lookup table', similar to those used for temperature compensation (see previous chapter). This provides a linearly-interpolated compensating value with up to 7 control points, which is then added to the output result.

Generally, linearisation is a finer level of compensation than temperature compensation.
It should only be applied after the basic Cell calibration and temperature compensation (if any) have been set up.
Although the tests are generally simpler than testing over temperature, the accuracy requirement is often greater.
See below for notes of possible difficulties to be avoided.

Control Parameters

Refer to Chapter 9 for command numbers
The lookup table (based on parameters **CLXi**, **CLKi**) defines an offset adjustment based on the **CRAW** value, which is then added in to give the final **CELL** output.
(So linearity correction is applied after any temperature compensation.)

The Parameters Involved Are :

CLN	Sets the number of linearisation points (from 2 up to 7)
CLX1..7	Raw input (**CRAW**) value points
CLK1..7	Output (**CELL**) adjustments to apply at these points

They are used like this :

- The number of calibration points is set by CLN (from 2 up to 7)
- Raw input value points are set by CLX1, CLX2 .. CLX7 (or up to the number set by CLN)
  These ***must*** be arranged in order of increasing input value.
- The output corrections at these points are set by CLK1, CLK2 .. CLK7
- **Corrections are specified in "thousandths of a cell unit".** So a CLKi of 1.0 actually adds 0.001 to the CELL output). This due to a limitation in the ASCII conversion to floating point numbers.

Internal Calculation

This uses the same basic 'interpolated table lookup' method as for temperature compensation.
First, a working table index, i, is derived from the current raw input **CRAW**=x, as follows :
(n = number of points used, as set by **CLN**)

When $(x < CLX_1)$ then $i = 1$

When $(x > CLX_{n-1})$ then $i = (n-1)$

Otherwise i is chosen so that $CLX_i \le x \le CLX_{i+1}$

The resulting interpolated adjustment value is then calculated as –

$ofs = CLK_i + (CLK_{i+1} - CLK_i) \times (x - CLX_i) / (CLX_{i+1} - CLX_i)$

Then the compensated cell value is calculated as –
CELL = **CRAW** + ofs

How to Set Up Linearity Compensation

A linearity correction can be set up either from sensor specification/calibration data, or more commonly from in-system testing results.
Assuming we do not have any prior information on linearity errors, the usual approach is to do a series of controlled tests with accurately known test loads.
Just as with temperature compensation, it is *possible* to obtain a detailed graph of linearity error and then choose a 'best-fit' piecewise linear curve for the compensation table.
However, it is generally good enough, and much simpler, to simply test at several different points and then apply an exact correction at those points. If the error curve is reasonably smooth, this should give exact results at the test points, and reasonably accurate values in between.

NOTES:
Linearisation tests should only be done ***after*** the cell calibration is set, because the correction values are dependent on the cell calibration.
Similarly, linearisation testing should only be done at the calibration 'reference' temperature, or after temperature compensation is installed, to avoid temperature effects from distorting the results.
The linearisation tests should not reveal any significant remaining linear trend in the errors.
If errors do appear to lie on a definite line, this could drastically reduce the accuracy of the correction.
If this does happen, it shows that the cell calibration is wrong and should be redone.
The table points must always cover more-or-less the whole range of output values to be used, because corrections are extrapolated outward beyond the first and last points.
It is always worthwhile including more test-points than will be used in the correction table, because this gives confidence that no regions of rapidly changing error have been missed.
Tests should be done both with steadily increasing ***and*** decreasing load values, as hysteresis effects (for load cells) are often of a similar size to non-linearities.

Parameter Calculations and Example

Based on the simple method outlined above, we suppose that we have obtained test results for a series of precisely known load values –

test loads X_i give readings of CRAW = C_i, for (i = 1..n)

Then calculate the errors that need to be removed at these points –

$E_i = X_i - C_i$

Now just enter these values into the correction table, remembering to scale the errors –

CLN = n
$CLX_i = X_i$

$CLK_i = 1000 \bullet E_i$

Example

Suppose we have a load cell and Cell calibration giving a result in the range 0-500 KgF.

The following test results were obtained using a series of known test loads -

For test load of x1 = 0Kg : CELL reading c1 = 0.0010

For test load of x2 = 100.13Kg : CELL reading c2 = 100.44

For test load of x3 = 199.72Kg : CELL reading c3 = 200.57

For test load of x4 = 349.97Kg : CELL reading c4 = 349.75

For test load of x5 = 450.03Kg : CELL reading c5 = 449.98

We choose these precise test points as our linearisation reference points, so

CLN = 5

CLX1 = 0.0010

CLX2 = 100.44

CLX3 = 200.57

CLX4 = 349.75

CLX5 = 449.98

(Note that these are the raw reading values, not the known true values.)

Now calculate all the residual errors, and set up the correction factors -

CLK1 = 10^3×(x1 - c1) ≈ 1000×(000.00-0.0010) = -1.0

CLK2 = 10^3×(x2 - c2) ≈ 1000×(100.13-100.44) = -310.0

CLK3 = 10^3×(x3 - c3) ≈ 1000×(199.72-200.57) = -850.0

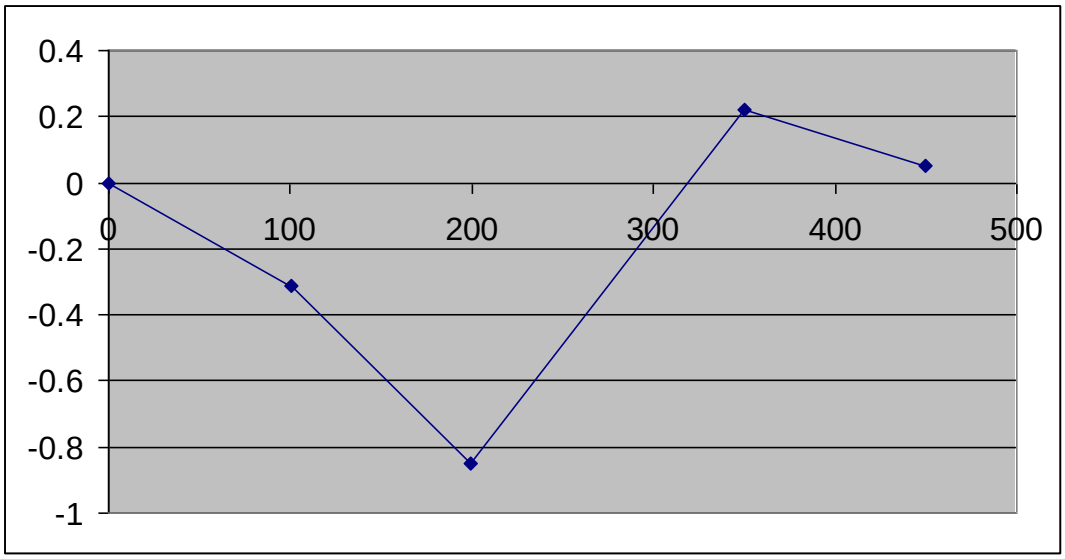
CLK4 = 10^3×(x4 - c4) ≈ 1000×(349.97-349.75) = +220.0

CLK5 = 10^3×(x5 - c5) ≈ 1000×(450.03-449.98) = +320.0

The CELL output values will now have the required values at all these 5 points.

Note on the Example

If you graph the errors from the above example, the results look like this



This doesn't show any very definite linear trend, so the calibration is okay.
However there ***is*** a big jump between points 3 and 4, which might be worth a more detailed investigation: Some important features of the error curve could have been missed by the test.

Chapter 7 Self-Diagnostics

Diagnostics Flags

The main diagnostics facilities are by means of the flags. See chapter 3 Flags for a full description of the flags and their meaning.
The flags are normally used something like this.
FLAG is read at regular intervals by the host (like the main output value, but generally at longer intervals)
If some warnings are active, i.e. **FLAG** is non-zero, then the host tries to cancel the warnings found by writing **FLAG**= 0
The host then notes whether the error then either remains (i.e. couldn't be cancelled), or if it disappears, or if it re-occurs within a short time, and will take action accordingly.

The warning flags are latched indicators of transient error events: By resetting the register, the host both signals that it has seen the warning, and readies the system to detect any re-occurrence (i.e. it resets the latch).

What the host should actually do with warnings depends on the type and the application: Sometimes a complete log is kept, sometimes no checking at all is needed.
Often, some warnings can be ignored unless they recur within a short time.

Warning flags survive power-down, i.e. they are backed up in non-volatile (EEPROM) storage.
Though useful, this means that repeatedly cancelling errors which then shortly recur can wear out the device non-volatile storage – see *WARNING: Finite Non-Volatile Memory Life* in *chapter 3*.

Diagnostics LED

A new feature for Version 3 is the addition of an LED to indicate the current status of the device and also it's protocol. If all is healthy the LED should flash ON for a period of 100mS , The rate at which the LED flashes is used to indicate the protocol as indicated in the table below.

Protocol	LED Flash Period
ASCII	0.5 seconds
MODBUS	1 second
MANTRABUS II	2 seconds

If an error conditions occurs which is based on any of the following error flags being set then the operation of the LED will invert. IE the LED will flash off for 100mS at the rate set in the table above.
These flags Being, TEMPUR, TEMPOR, ECOMUR, ECOMOR, CRAWUR, CRAWOR, SYSUR, SYSOR & LCINTEG.

Chapter 8 Communication Protocols

This chapter gives details of communication protocols and bus connections.

There are effectively three 'layers' to DCell/DSC communications
1. Internally, all devices support the same 'command set', as described in earlier sections.
2. Command-accesses are coded into actual byte sequences according to a communications protocol. Several different protocols are available.
3. Serial communications are carried out over a bus, which operates according to the usual RS232 or RS485 standards.

The bus standard and protocol type are fixed for a given device during production.
The communications baud rate and station number (the "bus address") are configured for each device by the ordinary control parameters STN and BAUD (see *Chapter 2 Getting Started with the Evaluation Kit*).

Bus Standards

Serial Data Format

Serial data formatting is the same for all the protocols and is fixed to
- one start-bit
- one stop-bit
- 8 data bits
- no parity

Communications Flow Control

Bus flow control is managed as part of the protocol (managed differently by each).
No hardware or software flow control signals are to be used for any of the bus standards.

Communications Errors

Serial data which does not conform to the expected format causes a 'serial framing error' to be registered, which increments the Communication Failure Counter **CFCT**.
What this actually means is that, following a starting (1 to 0) transition, a 'stop' (1) bit was not seen in the expected place. This is obviously baud rate dependent; the commonest cause being data transmitted at a lower baud rate than the unit was configured for.

Choice of Bus Formats

Essentially, the RS232 bus is *only* suitable where the application will only ever need a single DCell/DSC device.
The only real advantage of RS232 output is that no data converter is required to connect the device to a PC.

The RS485 bus is the simplest and most flexible choice from the wiring point of view. It enables multi-drop operation and much longer cable lengths.
However, it usually requires a bus converter, which must be chosen to suit the equipment in use.

The RS232 Bus Standard

RS232 has separate, dedicated transmit (Tx) and receive (RX) wires. Each wire is permanently driven by the sending end, with no transmit enable/disable controls, so RS232 is only suitable for one-to-one communications.

The connection is basically three-wire, in that the Tx and RX signals are related to a common ground potential. For DCell/DSC connections, this ground is shared with the power return (VIN–).
(It is therefore especially important for RS232 devices to connect the ground ***at the device end***, to avoid noise from communications degrading the reading accuracy).

The RS232 standard specifies operation over line lengths of 15M (50 ft) or less, more or less independent of baud rate.
The drive capability of RS232 compatible hardware varies a great deal, but most exceed this comfortably, at least at the baud rates used by DCell/DSC devices.

The RS485 Bus Standard

RS485 differs from RS232 in two important respects:

1. It uses differential signalling on pairs of signal wires. With small voltage detection thresholds and a large amount of common-mode rejection, this improves performance over longer distances.
2. It uses driver enable/disable controls to allow the same wires to be shared between several devices. This enables multi-drop operation.

The RS485 standard is a two-wire bus. For good noise immunity, the two wires are normally twisted together and shielded.
There is no defined ground connection: All attached devices load the A and B lines so as to normally pull the grounds of floating devices to within a few volts of each other. The common-mode tolerance (effectively about +/–7 Volts) then allows communications without any further grounding provision.

An RS485 bus of any length also behaves like a transmission line, and so must be terminated to avoid reflections. This is done by connecting a 120 Ohm resistor between the A and B lines at each end of the bus.

> The RS485 standard is specified for operation up to 4000 feet (1200 metres) at any rate below 9600 baud, and proportionately less at higher rates.

However, this assumes an ideal straight cable run with termination at both ends.
In practice, a bus with many devices along its length can fall short of this, especially if any connected devices are on long 'stubs' leading off the main bus. This may mean that the bus has to run at a slower speed than expected for reliable communications.

Communications Protocols

All communications take place according to a specific protocol. All devices on a bus must use the same protocol to avoid confusion.
The protocol specifies the structure and meaning of data exchanged, and how access is controlled so as to avoid collisions.
The various DCell/DSC variants support a variety of different protocols, allowing integration with various other types of devices on a network.

Choosing a Protocol

The current choices are

ASCII	printable characters, easy to drive, direct output to printers/displays
MODBUS RTU	binary industry standard, inter-compatible with other devices such as PCBs
MANTRABUS-II	efficient binary protocol, checksums give better security than ASCII

Communications Software for the Different Protocols

To access a DCell/DSC, you will need a communications application running on your PC or PLC, in addition to the appropriate hardware connections.
The simplest approach for initial experiments is to use the Instrument Explorer evaluation application.

Other Simple Ways Of Using The Different Protocols Are As Follows :

1. The ASCII protocol only uses basic printable characters, and so can be accessed with a simple terminal program like Windows 'HyperTerminal'.
2. The MODBUS protocol can be accessed via a proprietary generic MODBUS application.
   For evaluation purposes, we suggest the free shareware demo of 'ModScan32' from Win-Tech software (visit www.win-tech.com).

Common Features of All Protocols

The communications protocols are all of the master/slave type: A central 'host' device (normally a PC or PLC) acts as a bus master, in control of all communications.
Communications consist of the master transmitting **command messages** addressed to particular DCell/DSC 'slave' devices. The target slave may then transmit a 'response message' back to the master.
Because there is only one bus master, and slaves never initiate communications, the master is in control of all communications. This enables multi-drop operation, i.e. a single master can control any number of slave devices at a time.
In practice, the master usually 'polls' the attached slaves, interrogating them in a fixed rotation.

Command Types
A single command sent to a device can instruct it to read from or write to an internal 'parameter' value, or to execute a 'Control Action'.
The device responds by returning data (for a parameter read) or a simple acknowledge (for write or action): Precise details depend on the protocol in use.

Each Command Message Contains the Following Information:
1. The intended slave address (or broadcast)
2. The command to access
3. The access type, one of read / write /execute
4. (For write accesses only) the write value

The Command Response will then be one of the Following Three Types:
1. An error indication
2. A simple acknowledge
3. A read data-value (implicit acknowledge)

The various protocols differ quite a lot in the available types of error and acknowledgements. The distinction between different access types is also protocol-dependent; in that some use a dummy read or write command for execute access.

Slave Addressing and Broadcast
Every slave device on the bus is identified by a unique address value, known variously as its 'station number', 'node id' etc. depending on the protocol.
Each command message contains an address specifying to which slave device it is directed. A slave will ignore all communications that are not addressed to it.
All the protocols also define a special address value (normally 0) which is reserved for broadcast commands, which all slaves act on.
No response is allowed to broadcast commands, as multiple replies would collide with one another.

Parameters
Parameters are the values used for all control settings and output values. They have an associated storage type (byte, integer or real-value), and may be either read/write, read-only or write-only.
Output or 'result' values are mostly read-only.
Configurable parameters are held in non-volatile storage, so control settings are retained permanently even when power is removed.

Data Type Conversions and Rounding

Type Conversion
Depending on the protocol, an integer/byte parameter may need to be converted to or from a floating-point representation for reading or writing.
The rules are as follows
For reading, integer and byte parameters are treated as unsigned, and never read negative
– i.e. read value ranges are 0 to 65535.0 and 0 to 255.0
For writing, values written to integer and byte parameters are truncated to the nearest integer, and negative or positive values are acceptable
NOTE: Floating-point data is not always exact, even when reading integral data

E.G. could get 3.999974 instead of 4
eg. for a byte write 240, 240.1 and 239.66 are all the same value

Rounding
Although rounding is applied when writing to integral values, data read from a device is ***not*** rounded off.

The ASCII Protocol

The ASCII protocol uses only printable characters and carriage-return ('<CR>'), which allows a "dumb" terminal device or a PC programme like Hyper-Terminal to interrogate the device. Host Command Message Format
The basic command request structure is shown in the following example, illustrating the message "!001:SGAI=123.456<CR>", meaning write 123.456 to parameter SGAI on station 1:

Framing Character	Station Address	Separator	Command Identifier	Access Code	Data	End of frame
!	001	:	SGAI	=	-99.9999	<CR>

An Explanation of Each Field is as Follows:

- **Framing Character** A "!" character is used to signal the start of a new message. This character is only ever transmitted by the host, for framing purposes
- **Station Address** A three-digit ASCII decimal number (0-999), determining which slave device(s) the command is intended for. All three digits must be sent.
  Address 000 is reserved for broadcast addressing.
- **Separator** always present. As no checksum or message verification technique is used, slaves use this as an extra check on message validity
- **Command Identifier.** Up to 4 alpha-numeric characters, case insensitive, giving the name of the required command.
- **Access Code.** Defines what sort of response is expected :-
  '**=**' means write data is expected to follow
  '**?**' means the host is expecting to receive read data back
  <CR> (i.e. nothing more before end) means the command is an action type (execute)
- **Data.** An ASCII decimal-formatted number, can include 0..9, '+','-', '**.**' and spaces.
  This field can have a maximum length of 15 characters
- **End of frame.** A <CR> is always present to indicate the end of the message

Summary

- A command message begins with '**!**', followed by a three-digit station address, then a '**:**', and finishes with a <CR>.
- The '**!**' and <CR> *only* appear at the beginning and end of commands respectively
- From the '**:**' to the final <CR> is the command 'instruction' (of read, write or execute type)
- All instructions begin with an alphanumeric command identifier of up to 4 characters, and end with a non-alphanumeric (which may be the final <CR>).

Slave Response Message Formats
Each slave monitors the bus for command messages. It responds to any message that is addressed to it by sending a response message.

To be accepted by a slave device, a message must start with '**!**', the correct three-digit slave address and '**:**', and end with <CR>, with no intervening extra '**!**'.
The slave will then *always* respond.

There are Three Possible Types of Response: acknowledge (ACK), acknowledge with data (for a read), and not-acknowledge (NAK)
ACK is a single <CR> character. This confirms an execute or write command.
ACK-with-data is a decimal number, followed by <CR>. This confirms a read and returns the data value.
NAK is an '?' <CR> sequence. The device rejected the command.

There Are Several Possible Reasons For A NAK Response
Command identifier not recognised
Badly formatted command: Missing command identifier, unrecognised access-code character, or unexpected character somewhere else
Access attempted not supported by this command.

NOTES:

- From receipt of the host's terminating <CR> to a response from the device (if any) will be at most 50mS. After this, it can be assumed there is no response.
- There is no value-checking: A slave can *not* NAK a command because a write data value is 'unsuitable' in some way, only if write access itself is disallowed.

For the Ack-with-data (i.e. a successful read command), the returned value consists of printable ASCII characters finishing with a <CR>, formatted according to the DP and DPB settings as follows

Write Command
If the device accepts the command then a <CR> is transmitted. There is no error checking on the data received by the device.

Example:

A command to set the BAUD parameter to three on station 1 could look like this –

!001:BAUD=3<CR>

assuming a device with STN=1 is present, it will respond with –

<CR>

Read Command
Returns the requested value specified by the command.
The returned value is formatted according to the DP and DPB values: The response consists of a sign character (±), DPB decimal digits before a decimal point, DP digits after the point and a terminating <CR>:
The length of the response is thus fixed at DP+DPB+3 characters.

Example:

A command to read the SOUT output could look like this –

!001:SOUT?<CR>

if the value=32.1, and format settings are DP=3 and DPB=5, the response string will be –

+00032.100<CR>

Action Command
If the device accepts the command then a <CR> is transmitted.

Example:

A command to reset device 14 would look like –

!014:RST<CR>

the response string will be just –

<CR>

Broadcast Commands
If the station address in a command message is "000", this means a broadcast command:.
All slaves act as normal on a broadcast command, but do not respond.

Example:

A command to all devices on the bus to sample their inputs would look like this –

!000:SNAP<CR>

- there is no response

Bad Commands
If any command is not understood by the device then a "?" is transmitted followed by a <CR>.

Example:

A unrecognised command, correctly addressed to station 173 –

!173:XYWR?<CR>

produces the general error response –

?<CR>

Continuous Output Stream (ASCII ONLY)

For the ASCII protocol only, there is a 'continuous output' mode:

The SOUT value is continually broadcast at the output rate. The maximum output rate is 300Hz with minimum baudrate of 115200.

The output is switched on and off by sending the standard ASCII 'XON/XOFF' control bytes (ctrl-Q = 0x11 and ctrl-S = 0x13).

This feature is intended for output to a single, simple serial display devices and printers.

It Has Certain Special Limitations As Follows:

- It can only be used in one-to-one operation, i.e. only one unit on a bus otherwise collisions will occur.
- On a RS485 bus it is not always easy to switch off, as the stop instruction must be transmitted when the device itself is not transmitting: If the output rate is limited by communications speed, then output is virtually continuous and may be impossible to stop.
  (N.B. this does not apply to RS232, which has separate transmit and receive connections)
- The operation does not start automatically, i.e. an initial Ctrl-Q must be sent. This means that if there is (for instance) a brief power interruption, output will stop.

To avoid possible problems, continuous-output operation is only enabled when the station-number (STN) is set to the special value of 999.

New for Version 4

- When Station No. is set for 998 the continuous output streams starts automatically from power up.

The MODBUS-RTU Protocol

MODBUS is a proprietary standard of Modicom Inc.
The full specification is quite complex, including a timeout-based framing strategy and polynomial CRC calculation, so full details are not given here. Refer to Modicom documentation. Knowledge of the MODBUS protocol is therefore assumed.

The MODBUS protocol is a partial implementation of the "RTU" (binary) form of the MODBUS standard, sufficient to allow DCell/DSC units to coexist on a serial bus with other MODBUS-compliant devices.

NOTE:
Third-party applications for MODBUS communications are readily available (e.g. ModScan from Win-Tech software, www.win-tech.com, who offer a free trial version).

Modbus Messages
All messages and responses are formatted and checksummed according to the normal RTU rules.
The slave number is the device station number: Slave '0' may also be used for broadcast writes.

The device command-set is mapped into the MODBUS 'Output' or 'Holding Registers' –

Parameters (read or write) are mapped onto a pair of registers containing a 4-byte floating-point value

Action Commands are implemented as dummy parameters: Writing activates the command and reading returns a dummy value (with no action)

Only Two Valid Message Function Codes Are Supported

Function 03	'Read Holding Registers'	- to read a register-pair
Function 16	'Preset Multiple Registers'	- to write a register-pair

The start address must always be a valid parameter address, which is always an odd number (see the following).

The ***only*** permitted data length is two registers, i.e. 4 bytes.
Registers ***cannot*** be read or written singularly, in larger groups or using other addresses (i.e. even-numbered registers cannot be addressed directly).

Parameter Addresses
All MODBUS parameter addresses are derived from the equivalent MANTRABUS register number by a simple "times 2 plus 1" calculation.
For example, the 'FLAG' parameter is Mantrabus register#14, so this occupies MODBUS holding registers 29 and 30 (because 2×14+1 = 29).

See the command table in for the starting register numbers.

Parameter Values
All exchanged values (read and write parameters) are in the standard IEEE 4-byte floating-point format:
The 32 Bits Of The Number Are Distributed As Follows

MSB	31	**Sign** bit ,	1=negate
	30-23	**Exponent**,	7-bit excess-127
LSB	22-0	**Mantissa**,	23-bit fraction with implicit 1

The value of the number is thus
$\{ (-1)^{Sign} * 2^{(Exponent-127)} * 1.Mantissa \}$,

Note the 'assumed 1' before the mantissa. The exception to this is the special value 0.0, which is represented as 4 zeroes.

eg. a floating point number of -12345.678 is represented as - [hex] C6 40 E6 B6
These 32 bits are mapped onto a register pair in the following way: The lower register holds bits 15-0 and the upper register bits 31-16.
These values are coded according to normal Modbus conventions, so the actual byte sequence in a read/write message is thus -
$R1_{hi}$, $R1_{lo}$, $R2_{hi}$, $R2_{lo}$
- Which in terms of bits is-
15:8, 7:0, 31:24, 23:16

Error Codes
Only Three Modbus Error Codes Are Supported, Which Are Used As Follows

01	'Illegal Function'	request for function other than 3/16
02	'Illegal Data Address'	attempt to read an unsupported register address
03	'Illegal Data Value'	attempt to write a read-only parameter, or message too long for buffer (valid messages have a known maximum length)

Write Command Example
Write value 1.23 (represents as hex 3F9D70A4) to registers 57,58 on slave#4
by sending

[hex]	04	station address
	10	function code
	00 38	start-reg hi,lo (NB h38=56 addresses register 57)
	00 02	quantity = 2 registers
	04	byte-count = 4
	70 A4	first=lower register (17) value = hi,lo
	3F 9D	second=upper register (18) value = hi,lo
	6B AB	checksum = hi,lo

A Correct Response Would Then Be

[hex]	04	station address
	10	info copied from command ...
	00 38	
	00 02	
	C0 50	checksum = hi,lo

Read Command Example
Read a value from registers 13,14 on slave#52
by sending –
[hex] 34 station address
03 function code
00 0C start-reg hi,lo (NB h14=20 addresses register 21)
00 02 quantity = 2 registers
01 AD checksum = hi,lo

A Correct Response, With A Value -55.2317 (Hex C25CED51) Would Then Be
[hex] 34 station response
03 function
04 byte-count
ED 51 C2 5C data
AA D4 checksum = hi,lo

Execute Command Example
Execute command 101 on slave#17
by sending –
[hex] 11 station address
10 function code
00 64 start-addr h64=100 → register 101
00 02 quantity = 2 registers
04 byte-count
00 00 00 00 data (value irrelevant)
A0 B4 checksum = hi,lo

The Acknowledge Response Is Then
[hex] 11 station response
10 info copied from command ...
00 64
00 02
02 87 checksum = hi,lo

The Mantrabus-II Protocol

Mantrabus-II is a two-wire system where data is transmitted & received over a common pair of wires. For this reason the framing character must not be sent in a reply from the responding DCell.
The protocol ensures this does not occur by splitting byte data into nibbles with the exception of the framing character & station number.

Framing Character
The framing character for Mantrabus-II is **FEh**, (this being different from the older Mantrabus-I **FFh** to allow the two protocols to be mixed on one bus).

Checksum
Both Host & Device send their EXOR checksum of all data sent, excluding framing character, in nibble format the MS nibble being first.

eg. EXOR Checksum of data is A7h. Checksum characters sent = 0Ah, 07h

Data Transfer
Data is both sent and received as 4 bytes split into 8 nibbles following the station number, plus two nibbles of checksum.

Floating-Point Data Format
All data sent & received in Mantrabus-II is in the IEEE floating-point format, this being a 4-byte floating-point number. The byte containing the sign & exponent is sent first, with the LS byte of the mantissa being last.
The Memory Layout Of The 4-Byte Floating-Point Numbers Is:

MSB	31	**Sign** bit ,	1=negate
	30-23	**Exponent**,	7-bit excess-127
LSB	22-0	**Mantissa**,	23-bit fraction with implicit 1

The value of the number is thus

$$\{ (-1)^{Sign} * 2^{(Exponent-127)} * 1.Mantissa \},$$

Note the 'assumed 1' before the mantissa. The exception to this is the special value 0.0, which is represented as 4 zeroes.

eg. a floating-point number of -12345.678 is represented as - [hex] C6, 40, E6, B6
This is transmitted in nibble format as - [hex] 0C, 06, 04, 00, 0E, 06, 0B, 06.

End of Data Identifier

As the protocol has no fixed length or length identifiers the ***last*** nibble of ***data*** sent to the device has its ***MS nibble set***. This indicates to the device that all data has been received & the next two bytes will be checksum data.

ACK & NAK

Mantrabus-II supports ACK & NAK, sending ACK (06h) at the end of a successful operation, and NAK (15h) for an unknown command or failed operation. These are always preceded by the station number (see examples below). N.B. Mantrabus-II will ***not*** transmit a NAK for invalid checksum data, but instead remains silent. (This is different from the behaviour of the older Mantrabus-I).

Writing to Variables

Station number and command number are followed by 8 bytes of nibble data (the last having its MS bit set), followed by the two checksum nibbles.

eg. To write the value 100.0 (Floating point 100.0 = 42h, C8h, 00h, 00h) to variable CGAI **(command number 40) at station 20, send the following**

FEh, 14h, 28h, 04h, 02h, 0Ch, 08h, 00h, 00h, 00h, 80h, 0Bh, 0Eh
| | | [DATA |] [EXOR CS]
Frame number | Station number | Cmd | MS bit of last Byte set | char

the response is then -
14h, 06h i.e. 'station number', 'ACK'.

Reading of Variables

To read an individual variable, the command number is sent with the MS bit set (i.e. no data following).

eg. To read *CGAI* (command number 40) from station number 20, send the following -

FEh, 14h, A8h, 0Bh, 0Ch
| | [|] [EXOR CS]
Frame char | Station number | Cmd with MS bit set of last byte

Assuming the value was -12345.678 (coded as C640E6B6h, representing -1 * 2^{13}* 12345.678 / 8192), the response will be -

14h, 0Ch, 06h, 04h, 00h, 0Eh, 06h, 0Bh, 06h, 01h, 0Fh
| [DATA] [EXOR CS]
Station number

Action Commands

These are transmitted like read commands, i.e. no data sent. The response is as for write commands.

eg. To reset station three (command 100), send the following.-

FEh,	03h,	E4h,	0Eh, 07h
|	|	|	[EXOR CS]
Frame char	Station number	Cmd with MS bit set	

The response is then -

03h, 06h

Chapter 9 Software Command Reference

This chapter contains tables of all DCell/DSC commands, with brief details of each.

Commands in Access Order

ASCII name	description	datatype	access	MB reg	MD reg
CMVV	Temp Compensated mV/V	float	RO	5	11
STAT	Status	int	RO	6	13
MVV	Filtered & factory calibrated mV/V	float	RO	8	17
SOUT	selected output	float	RO	9	19
SYS	main output	float	RO	10	21
TEMP	temperature	float	RO	11	23
SRAW	raw system output	float	RO	12	25
CELL	cell output	float	RO	13	27
FLAG	error flags	int	RW	14	29
CRAW	raw cell output	float	RO	15	31
ELEC	electrical output	float	RO	16	33
SZ	system zero	float	RW	22	45
SYSN	snapshot result	float	RO	23	47
PEAK	Peak value	Float	RO	24	49
TROF	Trough value	Float	RO	25	51
CFCT	Communications failure count	Float	RW	26	53
VER	software version	byte	RO	30	61
SERL	serial number low	int	RO	31	63
SERH	serial number high	int	RO	32	65
STN	station number	int	RW	33	67
BAUD	baud rate select	byte	RW	34	69
OPCL	Output Control (Value select)	byte	RW	35	71
RATE	reading rate select	byte	RW	36	73
DP	digits after point	byte	RW	37	75
DPB	digits before point	byte	RW	38	77
NMVV	Nominal mV/V for scaling ELEC	float	RW	39	79
CGAI	cell gain	float	RW	40	81
COFS	cell offset	float	RW	41	83
CMIN	cell range min	float	RW	44	89
CMAX	cell range max	float	RW	45	91
CLN	lin n-points	byte	RW	50	101
CLX1..7	lin raw-values	float	RW	51..57	103..115
CLK1..7	lin corrections	float	RW	61..67	123..135
SGAI	system gain	float	RW	70	141
SOFS	system offset	float	RW	71	143
SMIN	system range min	float	RW	74	145
SMAX	system range max	float	RW	75	151
USR1..9	g.p. storage values	float	RW	81..89	163..179
FFLV	Dynamic Filter Level	Float	RW	92	185
FFST	Dynamic filter steps	Float	RW	93	187
RST	reboot	-	X	100	201
SNAP	take snapshot	-	X	103	207
RSPT	Reset peak & trough	-	X	104	209
SCON	Shunt cal ON	-	X	105	211
SCOF	Shunt cal OFF	-	X	106	213
OPON	Digital Output on	-	X	107	215
OPOF	Digital output off	-	X	108	217

CTN	tempco n-points	byte	RW	110	221
CT1..5	tempco Temp points	float	RW	111..115	223..231
CTG1..5	tempco gain-adjust	float	RW	116..120	233..241
CTO1..5	tempco offset-adjust	float	RW	121..125	243..251

Table Key
".." - Denotes a range (e.g. CLK1..7 means "CLK1" to "CLK7")
Access RW/RO/WO/X = read-write / read-only / write-only / execute
Datatype float/int/byte/- = 4-byte real / two-byte integer / 1-byte integer / none
MB reg = register number for MANTRABUS protocol
MD reg = start register address (always odd) for MODBUS protocol

NOTES:
All Modbus accesses are in register pairs, Modbus addresses are just (2*MANTRABUS)+1

Chapter 10 Installation

This chapter gives detailed information on integrating DCell and DSC devices into a production system – including mounting, protection, adjustments, wiring and electrical requirements.

Before Installation

Carefully remove the DCell/DSC device from its shipment box. Check that the device is complete and undamaged. Check the Product Type Code – on the product label is that which you ordered.

The DCell/DSC can operate in any industrial environment providing the following limits are not exceeded

Operating Temperature	-40 ºC to +85 ºC
Humidity	95% non condensing
Storage temperature	-40 ºC to +85 ºC

For precise details of Environmental Approvals, see chapter 15.

It Is Advisable To Follow The Following Installation Practice Where Possible

- Minimise vibration
- Do not mount next to strong electrical or magnetic fields (transformers, power cables)
- Install electrical protection device as the unit is not internally fused
- Always ensure the package is secure and protected

Physical Mounting

DCell is normally sealed in the pocket of the load cell, which provides mechanical and moisture protection, and electrical shielding.
The DCell should be mounted using a 2mm screw to the body of the load cell. This should be a "Good" electrical connection to obtain maximum performance.
Connecting wires are soldered directly to the pads on the top and/or bottom of the PCB. Care must be taken to electrically insulate the connection pads from the surrounding metal.

DSC is normally installed in a protective enclosure, such as a metal box.
The pins can be plugged into standard (0.1" pitch) PCB header sockets, or soldered directly into a host board or to connecting wires.
It can be mounted either way up. Unwanted pins projecting on one side may be cropped off.
For extra vibration resistance, the 3 mounting holes provided can be used.
If not required, the protruding end with the single hole can be cut off to make the board smaller.

Electrical Protection

No additional electrical shielding is normally needed.
Electrostatic protection is sufficient for installation purposes only.
Devices are protected against shorting of communications lines to power supply, and shorting of sensor inputs.
No over-current protection is provided in case of faults, so the supply arrangements should ensure adequate power limiting or fusing.

NOT PROTECTED AGAINST REVERSE POLARITY OF SUPPLY.

Moisture Protection

Both DCell and DSC must only be operated in a dry environment, as moisture can dramatically degrade the measurement performance.

DCell
Will normally be sealed into a load cell pocket.
While flexible silicone sealant can be used to completely embed the unit, this is not adequate moisture protection in itself.

If required, the entire unit can be embedded in a potting compound: A two-part epoxy compound can be used, but bubbles and gaps must be avoided to prevent mechanical stresses which could break the device.
The compound used must be specified for electrical use, and have sufficient thermal conductivity to cope with the heat given off (up to 1W on a 15V supply).

DSC no additional electrical shielding is required, but moisture and/or mechanical protection are often required. Any simple box or enclosure can be used. If metal, the enclosure should be grounded to the SH connection (see *Communications Cabling and Grounding Requirements* below).

Soldering Methods

Take care soldering cables to the pads. Use a temperature controlled soldering iron set to a maximum 330ºC, for no longer than 2 seconds per pad. Excessive heat or increased soldering time may result in damage to the PCB.

NOTES:

1. Solder with water-soluble flux should not be used (even low-residue), as this can leave a surface film which attracts atmospheric moisture, degrading measurement performance.

2. DCell units can be damaged by poor soldering, due to the small nature of the circuit boards: Overheating or applying any pressure to a pad can de solder components on the other side of the board, or cause the pad itself to become detached.

Power Supply Requirements

The power supply needed is nominally 5.6 to 18V DC, but any possible droop or ripple must be included: The devices contain 'brown-out' detection, which may trigger if the supply voltage at the device drops below the 5.6 volts.

A single device consumes typically 40mA with a 350Ω gauge connected (except RS232-output units, which use about 10mA more).
An installation should therefore assume at least 60mA per unit, and allow for extra current being taken at power-on (though supply voltage can safely drop temporarily), and for possible voltage drops in long cables.

Any power-supply ripple should be below 100mV, and supply arrangements should provide current limiting for fault conditions (see *Electrical Protection,* above).

Cable Requirements

Strain Gauge Input (DSC)

For optimal performance twin twisted pair with individual shields is recommended, this gives good noise immunity. Maximum length should not exceed 20m. Normal 4 core shielded cable can be used in areas of low electromagnetic noise.

Power and Communication

Again, twin twisted pair cable is recommended for this. The cable should have the following characteristics.

- Twisted-pair with independent shields
- Characteristic impedance 50-150 ohms
- Core to core and core to shield capacitance below 300pF/m

A suitable type is BICC Brand-Rex BE56723 (also equivalent to Belden type 8723).
In the UK, this is available from Farnell, part number 118-2117

Normal 4 core shielded cable can be used in areas of **low** electromagnetic noise.

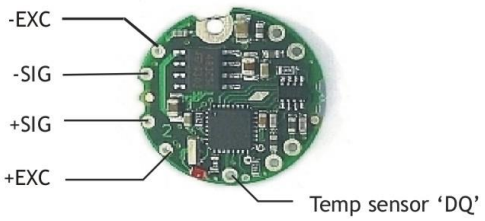
For RS232 operating at high baud rate on longer distances a triple twisted pair cable is recommended. A suitable type is BICC Brand-Rex PD3003 (also equivalent to Belden type 8777).
In the UK, this is available from Farnell (enter 'Belden 8777' for real length options).

Temperature Sensor

A shielded twisted pair is recommended, with a maximum length of 10m the shield being connected to the load cell body or (SH if DSC). For short lengths (< 2m) in a low noise environment (inside load cell body for example) then normal cable can be used.

Identifying Strain Gauge Connections

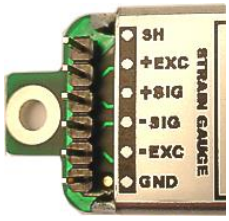
DCell Input Connections



The Optional Temperature sensor has two connections, "DQ" which provides the bi-directional data line and –EXC which provides the 0v connection.

Care must be taken with connections to the temperature sensor as erroneous values, typically 85°C, can occur if the cables are too long or in noisy electrical environments.
Please ensure, if using own supplied temperature sensor is either DS18S20Z or BS18S20.
DS18B20 MUST NOT be used.

DSC Input Connections



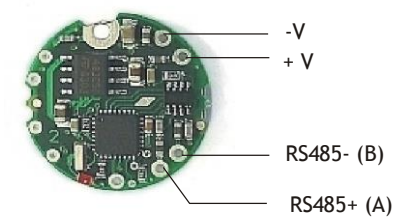
Strain Gauge Bridge Connections Are As Follows

- **EXC+/-** positive/negative excitation (supply)
- **SIG+/-** positive/negative output signal
- **SH** Shield connection for shield of strain gauge cable
- **GND** can be used as the ground connection for the temperature sensor

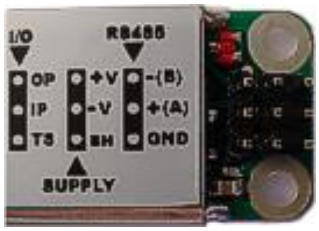
Other Connections (DSC ONLY)
Resistor **RG** and track-cut **TC** are used to adjust the mV/V sensitivity (see *Strain* Gauge Sensitivity Adjustment, next page).

Identifying Bus-End Connections

DCell Bus Connections

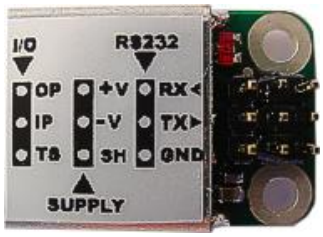


DSC4-RS485 Versions-Bus Connections



- **+V** and **-V** are the DC power-supply and return connections
- **RS485+ (A)** and RS485- **(B)** are *RS485* communications connections
- **GND** is a communications ground connection
- **SH** is the shield connection
- **OP** is the open collector digital output
- **IP** is the volt free digital input
- **TS** connects to DQ of the optional temperature sensor

DSC2-RS232-Bus Connections



- **+V** and **-V** are the DC power-supply and return connections
- **RX** and **TX** are *RS232* communications connections. RX being the DSC's receive
- **GND** is a communications ground connection
- **SH** is the shield connection

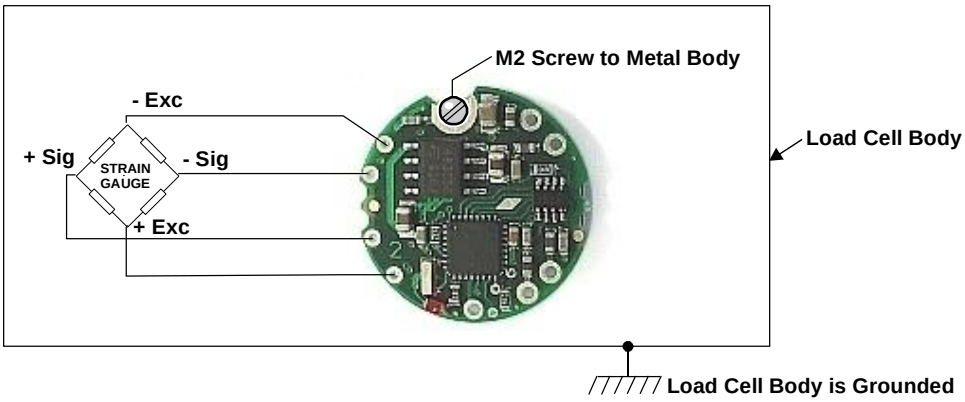
Strain Gauge Cabling and Grounding Requirements

To achieve full performance specifications and conform to environmental approvals, it is important to follow the wiring procedures outlined in this section.

DCell Strain Gauge Wiring

The following diagram illustrates how to wire up a DCell to a strain gauge

DCell Strain Gauge Wiring Arrangement

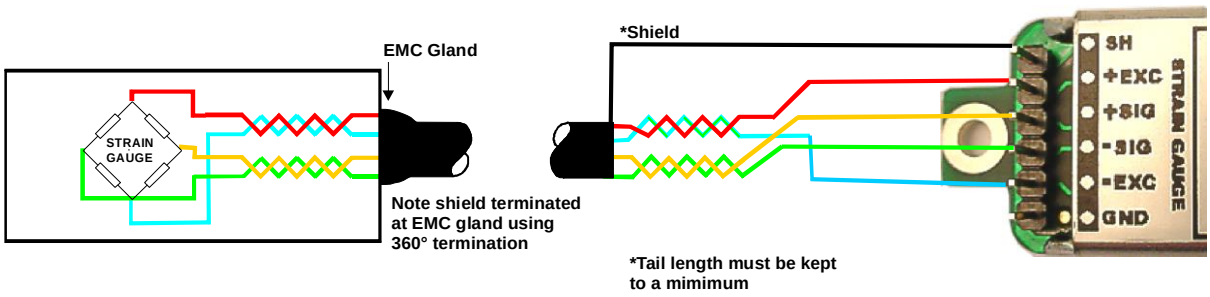


Key Requirements

All the load cell wires should be kept as short as feasible, at most 20cm.
The EXC+/– wires should be twisted together, also the SIG+/– pair, and the two pairs kept apart. It is also recommended to secure the wires from moving due to shock or vibration. If the DCell is mounted outside the body of the load cell then, for optimal performance, twin twisted cable should be used, although standard 4 core shielded cable can be used in **low** noise environments.

The M2 mounting hole **must be grounded** via an M2 screw to the load cell body for specified performance to be met. (The 2mm mounting hole to accept M2 screw or American equivalent #0-80. Important Note: DO NOT USE #2 screw size.)

DSC Strain Gauge Cabling Arrangement



Note: The specified cable above shows a yellow wire this will be replaced with a white wire.

Key Requirements

The Strain Gauge cable should be a twin twisted pair with independent shields, with the two pairs used for the EXC and SIG signal-pairs.
For specified performance, the load cell must be grounded to the SH.

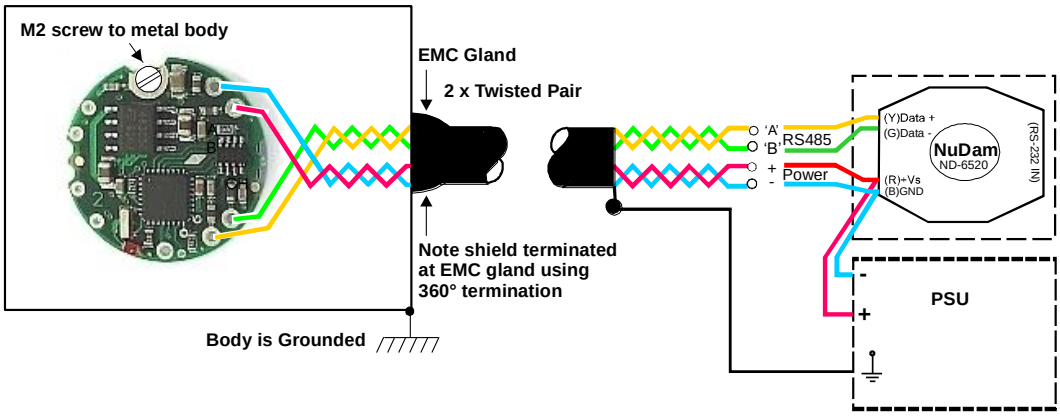
Communications Cabling and Grounding Requirements

To achieve full performance specifications and conform to environmental approvals, it is important to follow the wiring procedures outlined in this section.

DCell Power and Communications Wiring

The following diagram illustrates how to connect a puck to the communications and power supply (“bus”) cable.

DCell Bus-End Arrangement



Note: The specified cable above shows a yellow wire this will be replaced with a white wire.

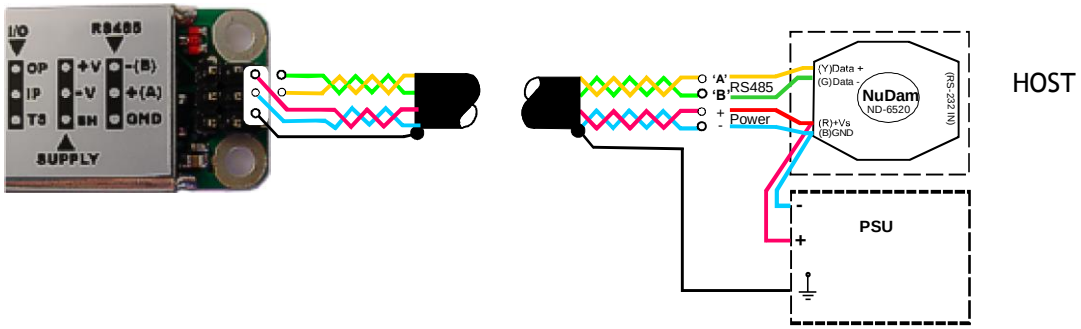
Key Requirements

The cable must enter the load cell via an EMC cable gland, which connects the cable shield to the load cell body. This must be a 360 Degree connection
The cable should be a twin twisted pair with independent shields, with one pair used for the communications and the other for the power wires.
There MUST be a common connection from the PSU and the RS485 ground to ensure the RS485 stays within the required common mode voltage of +/- 7v.
The shield should be connected to the grounded enclosure of the power supply.

DSC4-RS485 Versions- Power and Communications Wiring

The following diagram illustrates how to connect a DSC4 card to the communications and power supply (“bus”) cable.

DSC4-RS485 Versions-Bus-End Arrangement



Note: The specified cable above shows a yellow wire this will be replaced with a white wire.

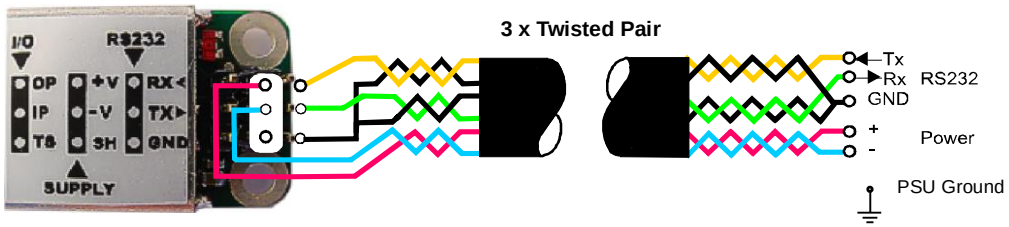
Key Requirements

The cable should be a twin twisted pair with independent shields, with one pair used for the communications and the other for the power wires.
The cable shield must be connected to the SH pin at the DSC end, and ***not*** at the host end.
Any further metal housing should also be connected to the DSC SH pin, and should ***not*** be connected to the bus cable shield (or the sensor cable).
The cable shield should also be connected to the grounded enclosure of the power supply.
There MUST be a common connection from the PSU and the RS485 ground to ensure the RS485 stays within the required common mode voltage of +/- 7v

DSC2-RS232 Versions (RS232 output) Power and Communications Wiring

The following diagram illustrates how to connect a DSC2 card to the communications and power supply (“bus”) cable.

DSC2-RS232 Versions Bus-End Arrangement



Note: The specified cable above shows a yellow wire this will be replaced with a white wire.

Key Requirements

The cable should be a triple twisted pair with independent shields. Each of the Rx, Tx and VIN+ wires is paired with a VIN- wire. For short cable lengths at low baud rates in a **low** noise environment then 4 core shielded cable can be used with the RS232 GND and PSU –V being common.

The cable shield should also be connected to the grounded enclosure of the power supply. The cable shield must be grounded to the SH pin at the DSC end, and ***not*** at the host end.
Any further metal housing should also be grounded to the DSC CH pin, and should ***not*** be connected to the bus cable shield (or the sensor cable).

Suitable Cable Types

DCell/DSC4-RS485 Versions-RS485 Bus Cable

Requires

- 2 × twisted-pair with independent shields
- Characteristic impedance 50-150 ohms
- Core to core and core to shield capacitance below 300pF/m

A suitable type is BICC Brand-Rex BE56723 (also equivalent to Belden type 8723)
In the UK, this is available from Farnell, part number 118-2117

DSC Sensor Cable and DSC2-RS232 Versions RS232 Cable
Both of these require 3 × twisted-pair version, otherwise similar to the above.
A suitable type is BICC Brand-Rex PD3003 (also equivalent to Belden type 8777)
In the UK, this is available from Farnell, part number 148-540

Warning: Special Problems with Portable Computers

Many portables use double insulated power supplies with no ground connection, and in this case a considerable voltage can appear on the port pins/chassis when the PC is powered off the mains.
When such a system hosts a DCell/DSC bus, the whole arrangement may be connected to the mains ground only via the external power-supply or the ground connections to the data-converter or the DCell/DSC units.

Electrical damage due to the brief presence of high voltages can easily occur when such a system is connected up.

Permanent harm can easily be done to the PC serial port, the data converter and/or DCell/DSC devices.

This is not simply a theoretical risk. We have seen several converters, and some DCell devices, destroyed in this way.

Also note that this kind of damage may often not be immediately obvious, appearing as erratic operation rather than outright failure.

To Avoid These Problems

1. Any portable should be separately grounded (e.g. via the converter supply) before connecting it to the mains supply, or to a DCell/DSC bus.
2. We always recommend the use of externally powered rather than port powered data converters (see below for details of suitable converters)

RS232 Bus Layout

Essentially, the only limitation here is on the cable length.
As described in *The RS232 Bus Standard in chapter 8*, this is supposedly up to **15m independent of baud rate**.

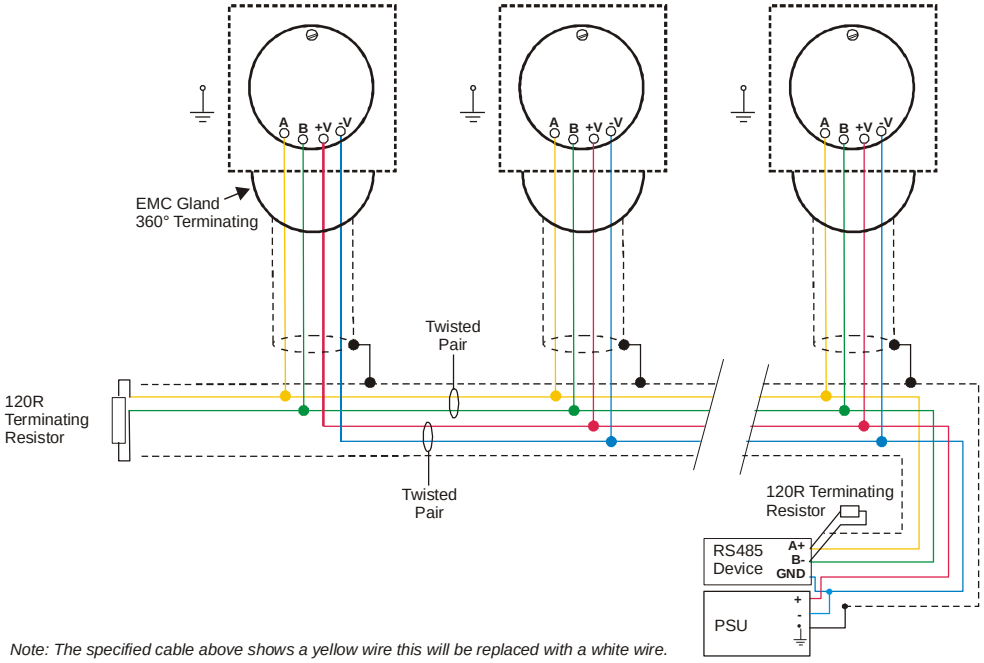
This is not a very realistic figure for typical modern hardware: For genuine RS232 compatible hardware, the length might be at least twice this at 9600 baud, and perhaps more at lower speeds.

However, some PCs have serial ports that are *not* truly RS232 hardware compatible, and may not have sufficient drive for specified operation.
In these cases, the port will probably still be usable with a "short enough" cable. It would be far safer, though, to replace the suspect hardware with something more suitable.

RS485 Bus Layout

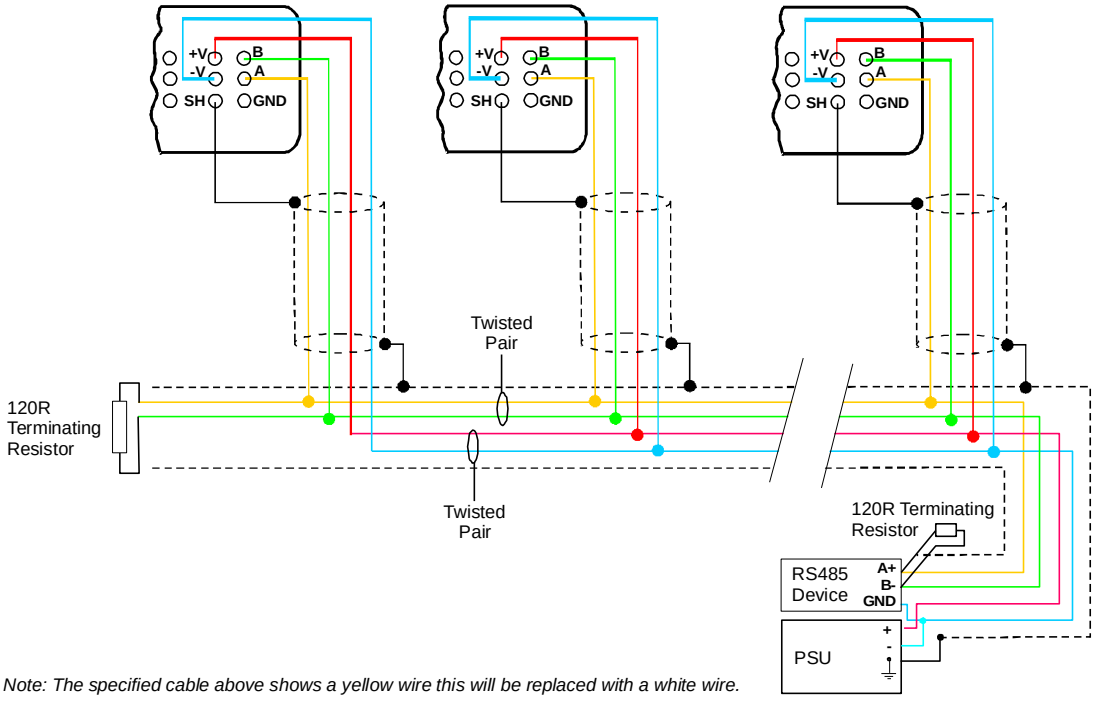
See also the general discussion of RS485 characteristics, *The RS485 Bus Standard* in chapter 8
Multiple devices are connected "in parallel" to the communications and power supply wire pairs, as shown in the following diagrams.

RS485 Bus Connections for Multiple DCells



Note: The specified cable above shows a yellow wire this will be replaced with a white wire.

RS485 Bus Connections for Multiple DSC RS485 Versions



Note: The specified cable above shows a yellow wire this will be replaced with a white wire.

Key Requirements

- The main bus cable must be terminated at either end.
- Where the bus does not go directly to each attached device, each 'stub' cable connecting to the bus should have just one device on it.
- Stub branches should be kept as short as possible, less than 10m at most.
- The 'stub' cables should be grounded at each device, and the main cable grounded at the data-converter end. *None of these should connect*.
- Stubs are not terminated.

Bus Layout and Termination

The ideal bus is a single length of cable, terminated at either end. Each end connects to a communicating device; while other devices are connected as near as possible directly to the main bus as it passes them (i.e. not on long side-branches).
The bus must be terminated at both ends to avoid reflections. Connecting a 120 Ohm resistor between the A and B lines does this.

Loading

In addition to reflections, each connected device places a load on the bus. According to the RS485 standard, a maximum of 32 'standard-load' devices can be simultaneously connected. The DCell/DSC devices are each one quarter standard load, so ***a maximum of 128 devices may be connected at once to a single length of bus***.
It is possible to increase this with bus repeaters, but the bi-directional nature of RS485 means it is usually simpler to add extra communications ports at the host, driving completely separate busses.

Grounding

The RS485 standard does not specify any particular ground connection.
If two external devices are both externally grounded (i.e. not floating), the grounds must be within the bus common-mode range (±7 volts) to connect safely and communicate. Beyond this, an isolating converter may be needed.

2 Wire & 4 Wire Connections

DCell/DSC devices only use two wire RS485 connections.
A 2 wire RS485 connection uses the same pair for transmit and receive, so the master has no special rights to the bus, and only the rules of the protocol used prevent two devices transmitting at once.

RS232 & RS485 Bus Converters

Typical DCell/DSC applications use a PC or PLC host, connecting via an RS232 port. This then requires an RS232 to RS485 bus converter to communicate with the bus.
The following features are of importance-

Two Wire RS485 Connection

Baud rate Support

Must support at least the rates to be used

Driver-enable Control

Can be ***either*** hardware control-line driven (normally via DTR), ***or*** automatic (host transmitting enables driver).

For Automatic Control

The converter detects host transmissions: It usually needs to be set to the correct baud rate, and may only support certain specific baud rates.

Power Options

Can be self-powered (i.e. from the RS232 port, maybe RTS, DTR or TX data-pin), or require an external power-supply.
Although a self powered converter seems attractive, it will usually have a limited drive capability: It will thus only drive a reasonably short bus with a few devices on it.

The RS232 & RS485 bus converter Mantracourt supplies as standard is an externally powered device, supporting all the DCell/DSC data rates up to 115200.
It features completely automatic data-rate detection (no set-up switches), and automatic enable-control switching with opto-isolation.

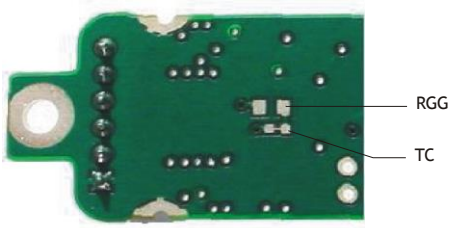
GND PSU must be connected to GND of the RS485 device.

Strain Gauge Sensitivity Adjustment (DSC ONLY)

Standard unit suitable for inputs up to 3m/V. Beyond this strain gauge sensitivity adjustment please consult factory.

If your strain gauge does not deliver a 2.5mV/V full scale output, you may want to adjust the sensitivity of the electronics (hardware) and/or the software gain controls.
If you want to test with an input of *more* than 2.5mV/V, you will have to adjust the hardware sensitivity to avoid saturating the input. If it is less, you can correct in software alone, but increasing sensitivity will generally improve accuracy.
To adjust the mV/V for DSC, an extra resistor 'Rg' is fitted across the pads RG, as shown above, in

Identifying the DSC 'Rg' Resistor



Identifying Sensor End Connections
The link across TC can be cut to disconnect the internal 100R gain resistor: This is needed for lowering the sensitivity.

The resistor is 0805 size surface mount chip.
A 0.1%, 5ppm/Deg C resistor for the high stability version & 25ppm/ Deg C for the Industrial must be used to maintain performance.

Reducing Sensitivity
To accommodate a maximum sensor output *larger* than 2.5mV/V, it is necessary to reduce the electrical sensitivity of the input circuitry.

To decrease sensitivity, the link **TC** is cut, and the value of the resistor fitted, in Ω, should be –

R_g = (required mV/V) x 40

Example: For 10mV/V

R_g = 10 x 40 = 400Ω

Increasing Sensitivity
When the full-scale output is *smaller* than 2.5 mV/V, it may be desired to increase sensitivity. However, it is often possible instead to compensate partly or entirely in software, by increasing a software gain control CGAI or SGAI

To *increase* sensitivity, **TC** is left in place, so that the fitted Rg appears in parallel (this gives better temperature stability). Its value should then be

RG = 1 / ((0.025 / rqd m V/V) – 0.01)

Two effects should be noted
1. The purpose of increasing sensitivity is to reduce reading *noise*, which governs the effective resolution. Using software gain alone obviously gives reduced performance.
2. The sensitivity should, however, not be set greater than typically 1mV/V: Beyond this, input noise usually dominates and no extra benefit can be achieved.

Chapter 11 Troubleshooting

This chapter gives a quick guide to problem solving for DCell/DSC devices.
Bear in mind that the quickest way to pin down problems is to usually replace items with 'known good' alternatives. This also applies to cables, power supplies, devices etc.

LED Indicator

The LED is used to indicate the protocol selection, the device is powered and the Device is
Operating. For correct operation the LED should Flash ON for 100mS then repeat at a rate depending on the protocol. See table below. If this is not the case then follow the instructions below.

Protocol	LED Flash Period
ASCII	0.5 seconds
MODBUS	1 second
MANTRABUS II	2 seconds

If the LED is OFF check the power supply polarity & voltage. The voltage including any troughs should be above 5.6V and less than 18V. If the supply polarity and voltage is correct then switch off, remove the strain gauge from circuit and switch on. If the LED is still off check the current drawn by the device. With no strain gauges should be approx 30mA.

If the LED is permanently ON then contact factory.

If The LED is ON for the majority of time then Flashes OFF for 100mS then a fault exists. This Fault can be read back using the communications. Likely causes of this are Strain Gauge Integrity error or Limits reached for MVV, CRAW, SRAW or TEMP. First check the connections to the strain gauge are correct. Next check the input is not over-range or the limits set for CMIN, CMAX, SMIN or SMAX have not been exceeded.

No Communications

The majority of problems involve a failure to communicate, as there are a fair number of optional settings that must be set the same at both ends of the link.
For this reason, any communications application should always check command responses, and flag a problem when there these responses are not activated.

Possible problems can be categorised according to where in the 'chain' of communication the problem may be. The typical chain runs as follows,

- PC software (port connection, baud rate, station number, protocol)
- PC serial port (working)
- Serial lead to converter
- RS232/RS485 data converter (power supply, PC port, wiring, transmit enable, baud rate setting)
- Bus wiring
- Device (wiring, station number, baud rate, protocol, working)

A quick checklist elaborates on these requirements, in the case that you are using the Instrument Explorer evaluation software (other software may have different requirements at the PC end): Check that –

PC End

1. PC software settings: right serial port, baud rate and protocol
   (standard data setup is 1 start bit+ 1 stop bit, no flow control)
2. PC serial port okay: check with other serial device, e.g. wire 2 PCs together with Hyperterminal running on both

(RS232 Versions) RS232 Evaluation Cable

3. Evaluation Kit cable: on right serial port; end *without* power supply wires connects to PC

(RS485 Versions) Cable + Data Converter

4. Cable to data converter: right serial port, standard 9 pin straight-through cable (***not*** 'null modem' type, with Tx/Rx crossed).
5. Power connected to data converter

6. Data converter baud rate setting ***(if not Evaluation Kit type)***: DIL switches for this, may be essential for automatic enable-switching
7. Data converter enable switching ***(if not Evaluation Kit type)***: If done by control line, check the serial port connects this correctly. If by transmission detect, how is baud rate set?

Evaluation Board or Device

8. Power reaches the device
9. ***(RS485)*** connections the right way round.
   With comms idle, 'B' should be a few tenths of a volt higher than 'A'
10. Device settings: correct station-number, baud rate
    (How do you know these are correct? A substitute device is very useful here!)
11. Device protocol: double-check product label
12. Device running okay:
    LED is flashing.
    devices take 20-30mA supply current without sensor attached, 30-50 with.

Bad Readings

The cause can be either hardware or software related.

Software

1. Check the MVV reading first and ensure it is correct. This figure is the RAW input and is not affected by the user configurable calibration settings.
2. If MVV looks correct, check the calibration settings step by step.
   Consider resetting all the calibration controls to default values – see *Chapter 4 The Readings Process*. This should make SOUT=MVV at all times.

Hardware

3. Load Cell problem should be indicated by flags in STAT, LCINTEG
4. Genuine hardware problems usual show up as **total** failure – i.e. no reading = always unchanging, usually near zero, sometimes always full-scale.
   Check wiring, take voltage level readings and again if possible use a known good device and set up.
5. Check the sensor is connected properly, and has some resistance across excitation wires, and around 350 Ohms across output wires (when disconnected from device).
6. Check for damaged DCell/DSC device by replacement

Unexpected Warning Flags

Remember that all warning flags in **FLAG** must be explicitly reset –they do not clear themselves when a problem is resolved.
If a flag cannot be cleared, the cause must be persistent –i.e. it keeps happening again. This can be immediate, regular (every few seconds) or irregular (occasional).
See *Self Diagnosis* for precise details of how the individual warnings operate.

Bear In Mind The Following Possible Problems

1. REBOOT or an increasing CFCT may indicate intermittent connections.
2. Where ECOMUR/OR or EXCUR/EXCOR are triggered, suspect input wiring.
3. Various 'range' errors (CRAWUR/OR, SRAWUR/OR) are also likely to be set if the excitation was interrupted (EXCUR/OR).
4. For range errors, check the associated limit parameters (CMIN/MAX, SMIN/MAX).
5. Problems are likely if any calibration MIN/MAX parameters are set the wrong side of zero (i.e. MIN>0 or MAX<0).

Problems With Bus Baud Rate

There are a number of special difficulties to be considered here

- Systems with very long cabling may not work with higher baud rates
- When using an RS232/RS485 converter, it may be necessary to change some converter settings when changing baud rate. Certain converters may not support some baud rates.
- Always remember you need to reboot devices before the change takes effect
- A bus with two devices talking at different baud rates may become unusable.
  So always change ***all at once*** by powering down or issuing a broadcast reset command (RST).

Difficult problems can always be overcome, if necessary, by isolating individual devices and trying the different baud rates in turn. This deals with all possible problems, as long as your hardware can deliver all the supported baud rates.

Recovering a "lost" DCell/DSC

For baud rate problems, see previous section.

You can try all 3 protocols if confused (but this should be indicated by the product code on the product label, if it has not been removed).
If a station number is unknown, it can be reset via *broadcast* command, as long as the device is the only one on the bus.

If two devices on the same bus end up set to the same 'bus address', they can no longer be commanded separately: The only solution is to remove one device from the bus and connect it exclusively to a PC to change its **STN**.

Always remember that a reboot (power-off or RST command) is needed to change STN and BAUD settings!

Chapter 12 Specifications

Technical Specifications DSC/DCELL High Stability

Set for 2.5mV/V sensitivity.

Parameter	**Min**	**Typical**	**Max**	**Units**
Strain Gauge Excitation System		4 Wire		
Strain Gauge Excitation Voltage	4. 5	5	5.25	VDC
Strain Gauge Drive Capability	320	-	5000	Ohms
Strain Gauge Sensitivity	-3	2.5	3	mV/V
Offset Temperature Stability		1	4	ppm/C
Gain Temperature Stability		3	5	ppm/C
Offset Stability with Time		20	80	ppm of FR (1)
Gain Stability with Time			30	ppm of FR (2)
Non Linearity before Linearization		5	25	ppm of FR
Internal Resolution		16 Million		Counts/divs
Resolution @ 1Hz readings (Noise stable) over 100s		200,000		Counts/divs
Resolution @ 10Hz readings (Noise stable) over 100s		120,000		Counts/divs
Resolution @ 100Hz readings (Noise stable) over 100s		50,000		Counts/divs
Resolution @ 500Hz readings (Noise stable) over 100s		18,000		Counts/divs
Signal Filter	Dynamic recursive type user programmable			
Optional Temperature Resolution				
Temperature Measurement Resolution		0.0625		Deg C
Temperature Measurement Accuracy (-10 to 85)		0.5		Deg C
Temperature Measurement Accuracy (-55 to 125)		2.0		Deg C
Temperature update Speed		5		Seconds
Electrical				
Power Supply voltage	5.6	12	18	V dc
Power Supply ripple			100	mV ac pk-pk
Power Supply current (350R Bridge)		45	60	mA
Power @ 10v (350R Bridge)		450		mW
Environmental				
Output Data terminal	RS485 (DSC offers RS232)			
Data transmission rate	2,400	-	230,400	bps
Output cable length (speed dependant)			1000	m
Operating temperature range	-40		+85	C
Storage temperature	-40		+85	C
Humidity	0		95	%RH
PCB Dimensions DSC	87.4 x 20 x 8.5mm			
PCB Dimensions DCell	Diameter 20mm, Height 5.3mm			

Notes.

1. From original offset at any time
2. 1st Year
3. Dependent on cable type and is for RS485, For RS232 max length is 20m

The DSC digital output is an open collector transistor rated at 100mA @ 40v

Technical Specifications DSC/DCELL Industrial Stability

DSC Conditioner is nominally set for 2.5mV/V sensitivity.

Parameter	Min	Typical	Max	Units
Strain Gauge Excitation System		4 Wire		
Strain Gauge Excitation Voltage	4. 5	5	5.25	VDC
Strain Gauge Drive Capability	320	-	5000	Ohms
Strain Gauge Sensitivity	-3	2.5	3	mV/V
Offset Temperature Stability		5	10	ppm/C
Gain Temperature Stability		30	50	ppm/C
Offset Stability with Time		35	160	ppm of FR (1)
Gain Stability with Time			300	ppm of FR (2)
Non Linearity before Linearization		5	25	ppm of FR
Internal Resolution		16 Million		Counts/divs
Resolution @ 1Hz readings (Noise stable) over 100s		66,000		Counts/divs
Resolution @ 10Hz readings (Noise stable) over 100s		40,000		Counts/divs
Resolution @ 100Hz readings (Noise stable) over 100s		10,000		Counts/divs
Resolution @ 500Hz readings (Noise stable) over 100s		5,000		Counts/divs
Signal Filter	Dynamic recursive type user programmable			
Optional Temperature Resolution				
Temperature Measurement Resolution		0.0625		Deg C
Temperature Measurement Accuracy (-10 to 85)		0.5		Deg C
Temperature Measurement Accuracy (-55 to 125)		2.0		Deg C
Temperature update Speed		5		Seconds
Electrical				
Power Supply voltage	5.6	12	18	V dc
Power Supply ripple			100	mV ac pk-pk
Power Supply current (350R Bridge)		45	60	mA
Power @ 10v (350R Bridge)		450		mW
Environmental				
Output Data terminal	RS485 (DSC offers RS232)			
Data transmission rate	2,400	-	230,400	bps
Output cable length (speed dependant) Note 3			1000	m
Operating temperature range	-40		+85	C
Storage temperature	-40		+85	C
Humidity	0		95	%RH
PCB Dimensions DSC	87.4 x 20 x 8.5mm			
PCB Dimensions DCell	Diameter 20mm, Height 5.3mm			

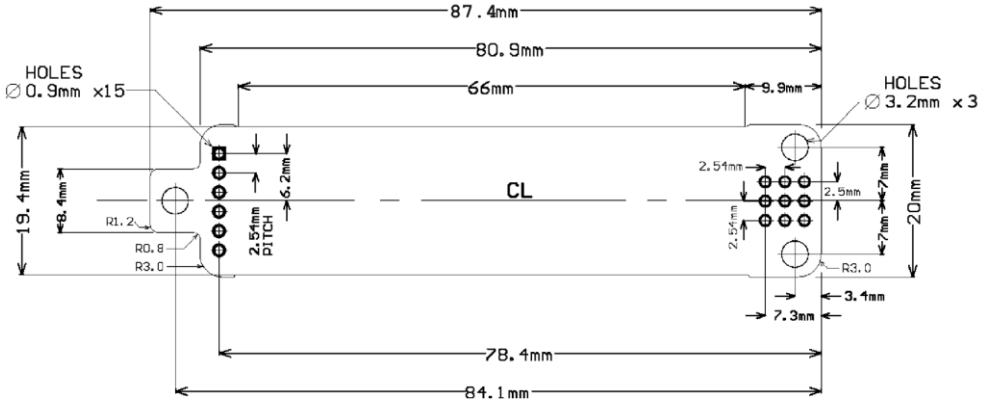
Notes.

1. From original offset at any time
2. 1st Year
3. Dependent on cable type and is for RS485, For RS232 max length is 20m

The DSC digital output is an open collector transistor rated at 100mA @ 40v

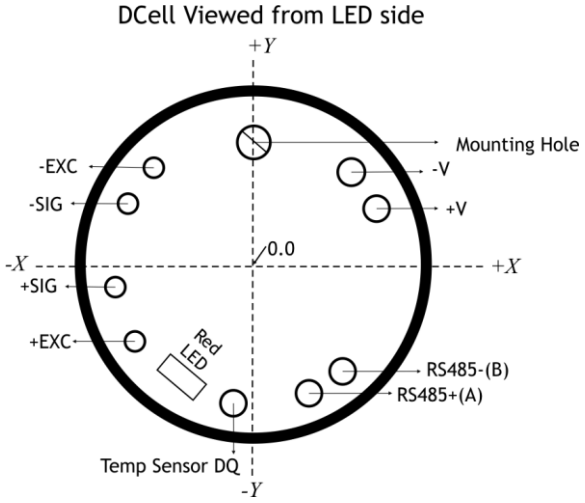
Mechanical Specification for DSC

Note: Viewed from top-side (CAN/label as photo on front cover of manual)



Mechanical Specification for DCell

DCell : Diameter 20mm, Height 5.3mm and has an 2mm mounting hole to accept M2 screw or American equivalent #0-80. Important Note: DO NOT USE #2 screw size



Hole centre positions taken from pcb centre.
All dimensions are in mm.

	x	y	Hole diameter
Mounting hole	0.0	+7.9	2.0
-V	+6.2	+6.1	1.0
+V	+7.9	+3.8	1.0
RS485-(B)	+5.8	-6.6	1.0
RS485+ (A)	+3.6	-8.1	1.0
Temp sensor DQ	-1.3	-8.8	1.0
+EXC	-7.5	-4.9	0.7
+SIG	-8.8	-1.5	0.7
-SIG	-8.0	+3.9	0.7
-EXC	-6.3	+6.1	0.7

CE Approvals

European EMC Directive	2004/108/EC
	BS EN 61326-1:2006
	BS EN 61326-2-3:2006

Warranty

All DCell & DSC products from Mantracourt Electronics Ltd., ('Mantracourt') are warranted against defective material and workmanship for a period of (3) three years from the date of dispatch.
If the 'Mantracourt' product you purchase appears to have a defect in material or workmanship or fails during normal use within the period, please contact your Distributor, who will assist you in resolving the problem. If it is necessary to return the product to 'Mantracourt' please include a note stating name, company, address, phone number and a detailed description of the problem. Also, please indicate if it is a warranty repair.
The sender is responsible for shipping charges, freight insurance and proper packaging to prevent breakage in transit.
'Mantracourt' warranty does not apply to defects resulting from action of the buyer such as mishandling, improper interfacing, operation outside of design limits, improper repair or unauthorised modification.
No other warranties are expressed or implied. 'Mantracourt' specifically disclaims any implied warranties of merchantability or fitness for a specific purpose. The remedies outlined above are the buyer's only remedies. 'Mantracourt' will not be liable for direct, indirect, special, incidental or consequential damages whether based on the contract, tort or other legal theory.
Any corrective maintenance required after the warranty period should be performed by 'Mantracourt' approved personnel only.

Designed, manufactured and supported in the UK

ISO 9001 REGISTERED FIRM

In the interests of continued product development, Mantracourt Electronics Limited reserves the right to alter product specifications without prior notice.

Code No 517-153	Issue 3.3	17.04.15

Distribuidor

Brasil e América do Sul

CONTATO

Endereço
Rua Sete de Setembro, 2656 - Centro
13560-181 - São Carlos - SP - Brasil

Telefone
+ 55 (16) 3371-0112
+ 55 (16) 3372-7800

Internet
www.metrolog.net
metrolog@metrolog.net

Metrolog Controles de Medição