Product Manual

Sensor

SDMS40

Multipoint Scanning Snowfall Sensor

01/2019

RELIABLE SINCE 1974 MONITORING

CAMPBELL SCIENTIFIC

Guarantee

This equipment is guaranteed against defects in materials and workmanship. We will repair or replace products which prove to be defective during the guarantee period as detailed on your invoice, provided they are returned to us prepaid. The guarantee will not apply to:

- Equipment which has been modified or altered in any way without the written permission of Campbell Scientific
- Batteries
- Any product which has been subjected to misuse, neglect, acts of God or damage in transit.

Campbell Scientific will return guaranteed equipment by surface carrier prepaid. Campbell Scientific will not reimburse the claimant for costs incurred in removing and/or reinstalling equipment. This guarantee and the Company's obligation thereunder is in lieu of all other guarantees, expressed or implied, including those of suitability and fitness for a particular purpose. Campbell Scientific is not liable for consequential damage.

Please inform us before returning equipment and obtain a Repair Reference Number whether the repair is under guarantee or not. Please state the faults as clearly as possible, and if the product is out of the guarantee period it should be accompanied by a purchase order. Quotations for repairs can be given on request. It is the policy of Campbell Scientific to protect the health of its employees and provide a safe working environment, in support of this policy a "Declaration of Hazardous Material and Decontamination" form will be issued for completion.

When returning equipment, the Repair Reference Number must be clearly marked on the outside of the package. Complete the "Declaration of Hazardous Material and Decontamination" form and ensure a completed copy is returned with your goods. Please note your Repair may not be processed if you do not include a copy of this form and Campbell Scientific Ltd reserves the right to return goods at the customers' expense.

Note that goods sent air freight are subject to Customs clearance fees which Campbell Scientific will charge to customers. In many cases, these charges are greater than the cost of the repair.

CAMPBELL® SCIENTIFIC
WHEN MEASUREMENTS MATTER

Campbell Scientific Ltd,
80 Hathern Road,
Shepshed, Loughborough, LE12 9GX, UK
Tel: +44 (0) 1509 601141
Fax: +44 (0) 1509 270924
Email: support@campbellsci.co.uk
www.campbellsci.co.uk

PLEASE READ FIRST

About this manual

Please note that this manual was originally produced by Campbell Scientific Inc. primarily for the North American market. Some spellings, weights and measures may reflect this origin.

Some useful conversion factors:

Area:	1 in^2 (square inch) = 645 mm^2	**Mass:**	1 oz. (ounce) = 28.35 g 1 lb (pound weight) = 0.454 kg
Length:	1 in. (inch) = 25.4 mm 1 ft (foot) = 304.8 mm 1 yard = 0.914 m 1 mile = 1.609 km	**Pressure:**	1 psi (lb/in^2) = 68.95 mb
		Volume:	1 UK pint = 568.3 ml 1 UK gallon = 4.546 litres 1 US gallon = 3.785 litres

In addition, while most of the information in the manual is correct for all countries, certain information is specific to the North American market and so may not be applicable to European users.

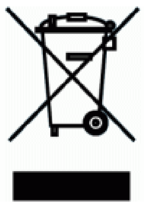
Differences include the U.S standard external power supply details where some information (for example the AC transformer input voltage) will not be applicable for British/European use. *Please note, however, that when a power supply adapter is ordered it will be suitable for use in your country.*

Reference to some radio transmitters, digital cell phones and aerials may also not be applicable according to your locality.

Some brackets, shields and enclosure options, including wiring, are not sold as standard items in the European market; in some cases alternatives are offered. Details of the alternatives will be covered in separate manuals.

Part numbers prefixed with a "#" symbol are special order parts for use with non-EU variants or for special installations. Please quote the full part number with the # when ordering.

Recycling information



At the end of this product's life it should not be put in commercial or domestic refuse but sent for recycling. Any batteries contained within the product or used during the products life should be removed from the product and also be sent to an appropriate recycling facility.

Campbell Scientific Ltd can advise on the recycling of the equipment and in some cases arrange collection and the correct disposal of it, although charges may apply for some items or territories.

For further advice or support, please contact Campbell Scientific Ltd, or your local agent.

CAMPBELL SCIENTIFIC
WHEN MEASUREMENTS MATTER

Campbell Scientific Ltd, 80 Hathern Road, Shepshed, Loughborough, LE12 9GX,
UK Tel: +44 (0) 1509 601141 Fax: +44 (0) 1509 270924
Email: support@campbellsci.co.uk
www.campbellsci.co.uk

Safety

DANGER — MANY HAZARDS ARE ASSOCIATED WITH INSTALLING, USING, MAINTAINING, AND WORKING ON OR AROUND **TRIPODS, TOWERS, AND ANY ATTACHMENTS TO TRIPODS AND TOWERS SUCH AS SENSORS, CROSSARMS, ENCLOSURES, ANTENNAS, ETC**. FAILURE TO PROPERLY AND COMPLETELY ASSEMBLE, INSTALL, OPERATE, USE, AND MAINTAIN TRIPODS, TOWERS, AND ATTACHMENTS, AND FAILURE TO HEED WARNINGS, INCREASES THE RISK OF DEATH, ACCIDENT, SERIOUS INJURY, PROPERTY DAMAGE, AND PRODUCT FAILURE. TAKE ALL REASONABLE PRECAUTIONS TO AVOID THESE HAZARDS. CHECK WITH YOUR ORGANIZATION'S SAFETY COORDINATOR (OR POLICY) FOR PROCEDURES AND REQUIRED PROTECTIVE EQUIPMENT PRIOR TO PERFORMING ANY WORK.

Use tripods, towers, and attachments to tripods and towers only for purposes for which they are designed. Do not exceed design limits. Be familiar and comply with all instructions provided in product manuals. Manuals are available at www.campbellsci.eu or by telephoning +44(0) 1509 828 888 (UK). You are responsible for conformance with governing codes and regulations, including safety regulations, and the integrity and location of structures or land to which towers, tripods, and any attachments are attached. Installation sites should be evaluated and approved by a qualified engineer. If questions or concerns arise regarding installation, use, or maintenance of tripods, towers, attachments, or electrical connections, consult with a licensed and qualified engineer or electrician.

General

- Prior to performing site or installation work, obtain required approvals and permits. Comply with all governing structure-height regulations, such as those of the FAA in the USA.
- Use only qualified personnel for installation, use, and maintenance of tripods and towers, and any attachments to tripods and towers. The use of licensed and qualified contractors is highly recommended.
- Read all applicable instructions carefully and understand procedures thoroughly before beginning work.
- Wear a **hardhat** and **eye protection**, and take **other appropriate safety precautions** while working on or around tripods and towers.
- **Do not climb** tripods or towers at any time, and prohibit climbing by other persons. Take reasonable precautions to secure tripod and tower sites from trespassers.
- Use only manufacturer recommended parts, materials, and tools.

Utility and Electrical

- **You can be killed** or sustain serious bodily injury if the tripod, tower, or attachments you are installing, constructing, using, or maintaining, or a tool, stake, or anchor, come in **contact with overhead or underground utility lines**.
- Maintain a distance of at least one-and-one-half times structure height, or 20 feet, or the distance required by applicable law, **whichever is greater**, between overhead utility lines and the structure (tripod, tower, attachments, or tools).
- Prior to performing site or installation work, inform all utility companies and have all underground utilities marked.
- Comply with all electrical codes. Electrical equipment and related grounding devices should be installed by a licensed and qualified electrician.

Elevated Work and Weather

- Exercise extreme caution when performing elevated work.
- Use appropriate equipment and safety practices.
- During installation and maintenance, keep tower and tripod sites clear of un-trained or non-essential personnel. Take precautions to prevent elevated tools and objects from dropping.
- Do not perform any work in inclement weather, including wind, rain, snow, lightning, etc.

Maintenance

- Periodically (at least yearly) check for wear and damage, including corrosion, stress cracks, frayed cables, loose cable clamps, cable tightness, etc. and take necessary corrective actions.
- Periodically (at least yearly) check electrical ground connections.

WHILE EVERY ATTEMPT IS MADE TO EMBODY THE HIGHEST DEGREE OF SAFETY IN ALL CAMPBELL SCIENTIFIC PRODUCTS, THE CUSTOMER ASSUMES ALL RISK FROM ANY INJURY RESULTING FROM IMPROPER INSTALLATION, USE, OR MAINTENANCE OF TRIPODS, TOWERS, OR ATTACHMENTS TO TRIPODS AND TOWERS SUCH AS SENSORS, CROSSARMS, ENCLOSURES, ANTENNAS, ETC.

Table of contents

1. Introduction

The SDMS40 Multipoint Scanning Snowfall Sensor is a two dimensional (2D) multipoint-scanning snow gauge, which scans its laser in a circular path on the snow's surface and measures the distance from each point on the path. Once it completes a set of measurements, the SDMS40 takes an intelligent average of the depths to provide a representative average snow depth of the target area. Communications options include SDI-12 and RS-232.

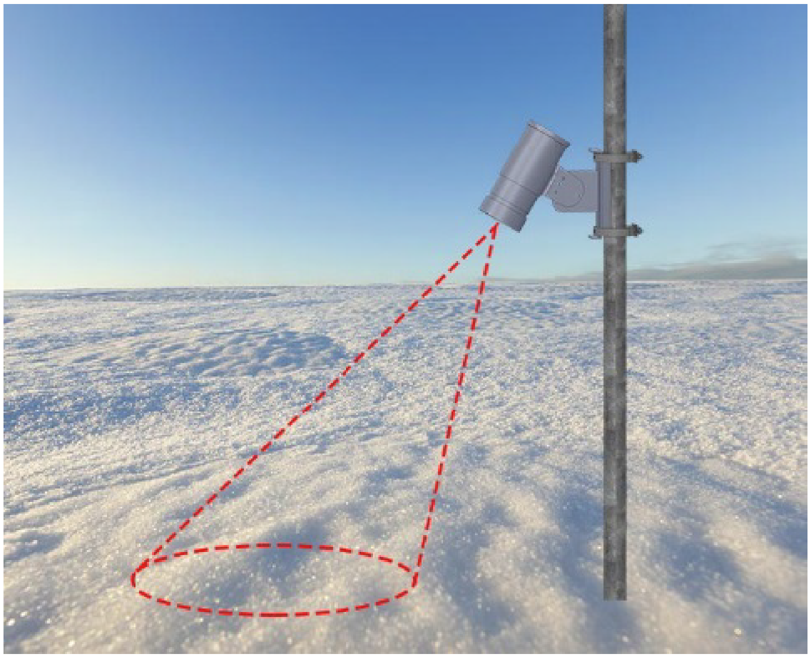


FIGURE 1-1. Laser area

FIGURE 1-1 (p. 1) demonstrates the oval pattern scanned by the sensor. Sophisticated filtering algorithms are implemented to provide reliable measurements in various weather and surface conditions.

The size of the target area varies depending on the height and tilt angle of the SDMS40.

2. Precautions

- READ AND UNDERSTAND the Safety (p. iii) section at the front of this manual.
- Although the SDMS40 is rugged, it should be handled as precision scientific instrument.
- To avoid shock or damage to the instrument, never apply power while working on wiring and connections.
- Never open the sensor when the power is turned on.
- The SDMS40 uses a Class 2 laser. Do not stare into the laser beam.

3. Initial inspection

- Upon receipt of the SDMS40, inspect the packaging and contents for damage. File any damage claims with the shipping company.
- Immediately check package contents against the shipping documentation. Contact Campbell Scientific about any discrepancies.
- Model number and cable length are printed on a label at the connection end of the cable (if a cable was purchased). Check the model number information against the shipping documents to ensure the expected product and cable length are received.
- Shipped with 4 screws, 2 lock washers, 2 band clamps, and mounting bracket.
- Ensure that the cable has a 1-Position Terminal Connector attached to the blue wire and another 1-Position Terminal Connector attached to the yellow wire. These connectors are used to individually isolate unused wires.

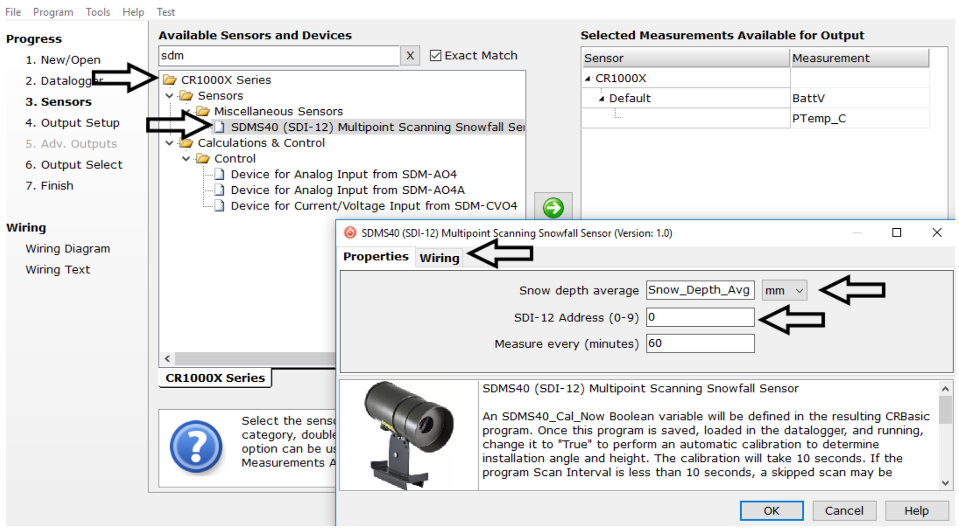
4. Quickstart

A video that describes data logger programming using Short Cut is available at: www.campbellsci.eu/videos/cr1000x-datalogger-getting-started-program-part-3. Short Cut is an easy way to program your data logger to measure the sensor and assign data logger wiring terminals. Short Cut is available as a download on www.campbellsci.eu. It is included in installations of LoggerNet, PC200W, PC400, or RTDAQ.

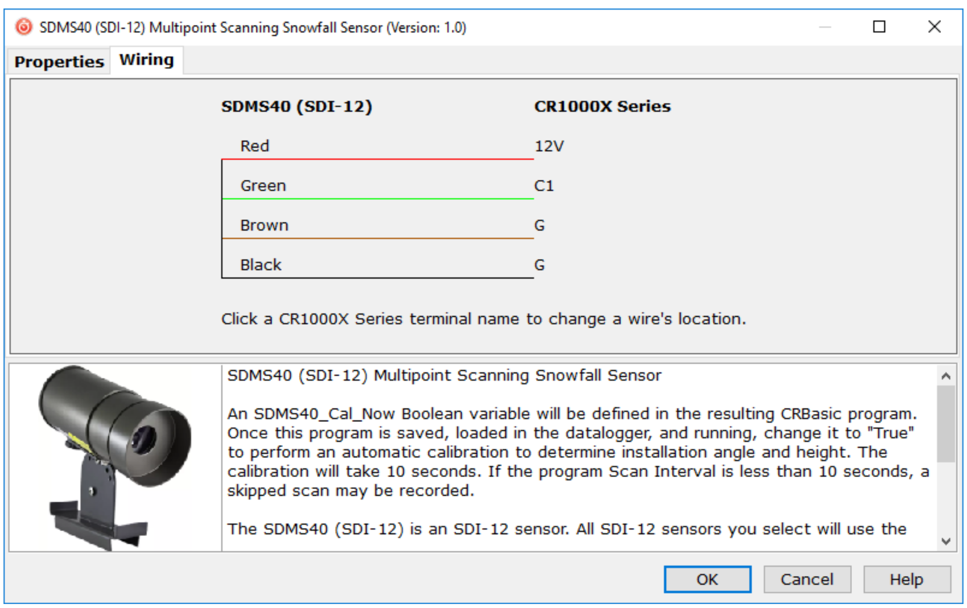
The following procedure also shows using Short Cut to program the sensor.

1. Open Short Cut and click **Create New Program**.
2. Double-click the data logger model.

3. In the **Available Sensors and Devices** box, type SDMS40 or locate the sensor in the **Sensors | Miscellaneous Sensors** folder. Double-click **SDMS40 (SDI-12) Multipoint Scanning Snowfall Sensor**. The default average snow depth units are millimetres. This can be changed by clicking the **Snow depth average** box and selecting different units. Type the correct **SDI-12 Address**.

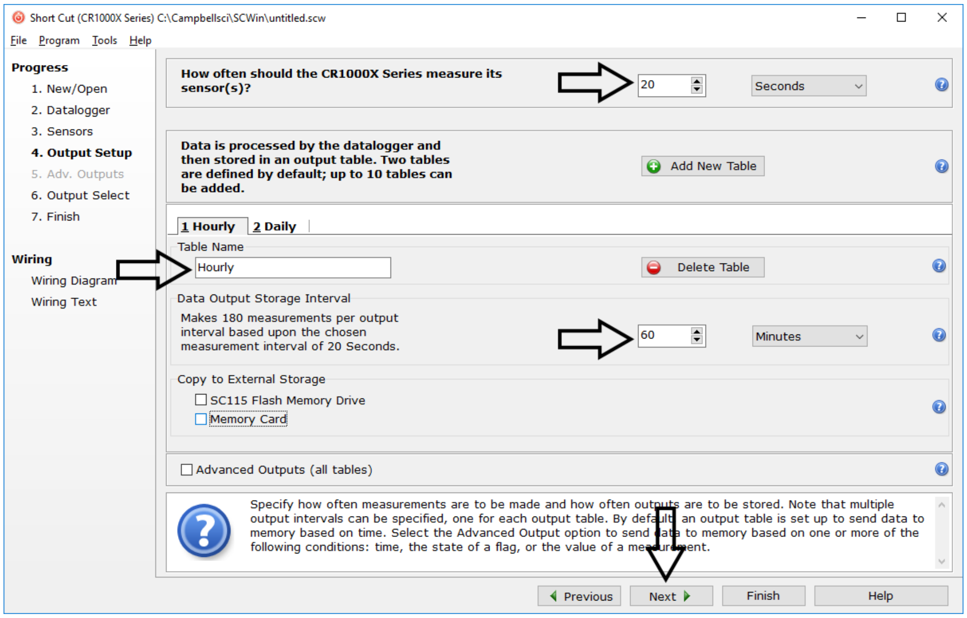


4. Click the **Wiring** tab to see how the sensor is to be wired to the data logger. Click **OK** after wiring the sensor.

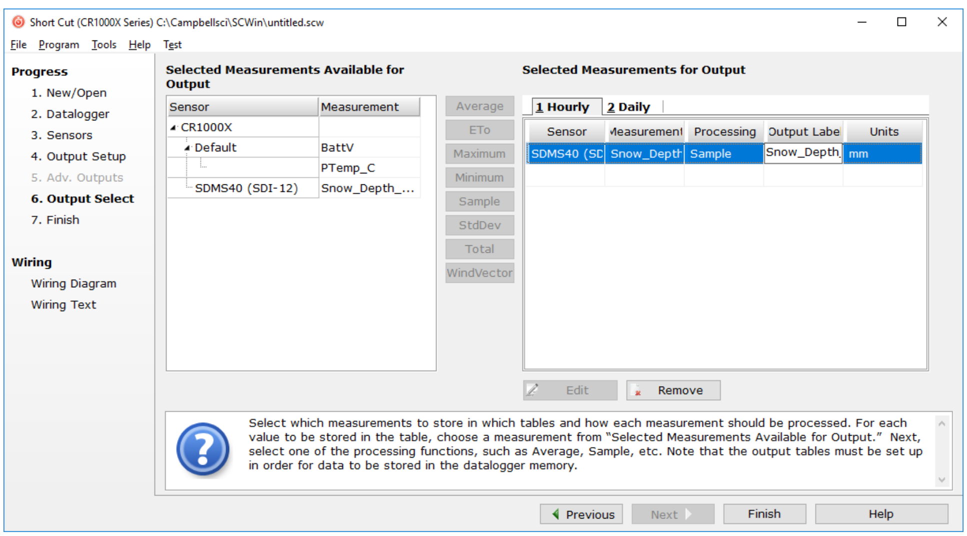


5. Repeat step three for other sensors.

6. In **Output Setup**, type the scan rate, meaningful table names, and **Data Output Storage Interval**.



7. Select the measurement and its associated output option.



8. Click **Finish** and save the program. Send the program to the data logger if the data logger is connected to the computer.

9. If the sensor is connected to the data logger, check the output of the sensor in the data logger support software data display in LoggerNet, PC400, RTDAQ, or PC200W to make sure it is making reasonable measurements.

5. Features

- Provides representative average snow depth of the target area
- Filters out erroneous measurement data caused by noise or foreign materials
- Detects new snowfall quickly and reliably
- Operates on natural ground or snow plate
- Compact and light
- Simple installation process
- After mounting, the sensor performs a fully automatic calibration process to calculate install angle and height
- Output data on SDI-12 or RS-232 serial data interface
- Compatible with Campbell Scientific CRBasic data loggers: CR200(X) series, CR300 series, CR6 series, CR800 series, CR1000, CR1000X series, CR3000, and CR5000

6. Specifications

Power supply:	Requires 12 to 15 Vdc power supply capable of providing up to 2 A continuously.
Current draw	
Standby:	50 mA
Active:	250 mA
Heater:	1300 mA
Cable:	
Type:	3-Cond 20 AWG, 2 Pair 24 AWG
Maximum length:	14 m (45 ft)

Sensor	
Method:	Multipoint laser scanning
Number of scanning points:	36
Range:	1 to 5 m
Target area diameter:	30 cm to 200 cm, depending on installation height and angle
Gauge pointing angle: Half angle:	0 to 45° from vertical
Resolution:	6°
Accuracy:	1 mm ±3 mm
Communications protocols:	SDI-12, RS-232
General	
Operating temperature:	–40 to 50 °C (with sensor heat on)
Weight:	1.8 kg (3.9 lb)
Enclosure protection class:	IP67
Laser safety:	Class 2
Dimensions	
Height:	12 cm (4.72 in)
Length:	28 cm (11.02 in)
Width:	10 cm (3.94 in)
Compliance documents:	View at www.campbellsci.eu/sdms40

7. Installation

If you are programming your data logger with Short Cut, skip Wiring (p. 8) and Programming (p. 10). Short Cut does this work for you. See Quickstart (p. 2) for a tutorial.

7.1 Mounting

The SDMS40 is environmentally sealed for outdoor installations. The enclosure provides protection from moisture and high humidity. It is not intended for operation under water. All that is required is an appropriate mounting fixture.

Position the SDMS40 about one metre above the maximum seasonal snow depth height (FIGURE 7-1 (p. 7), FIGURE 7-2 (p. 7)). This provides adequate height for required accuracy and resolution. Mounting procedure is provided in Table 7-1 (p. 8).

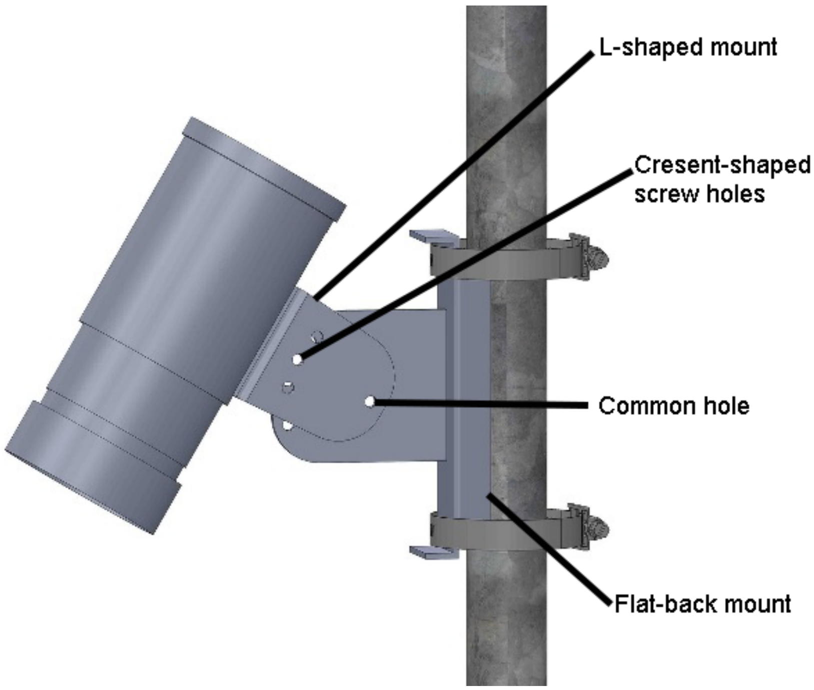


FIGURE 7-1. SDMS40 mounting holes

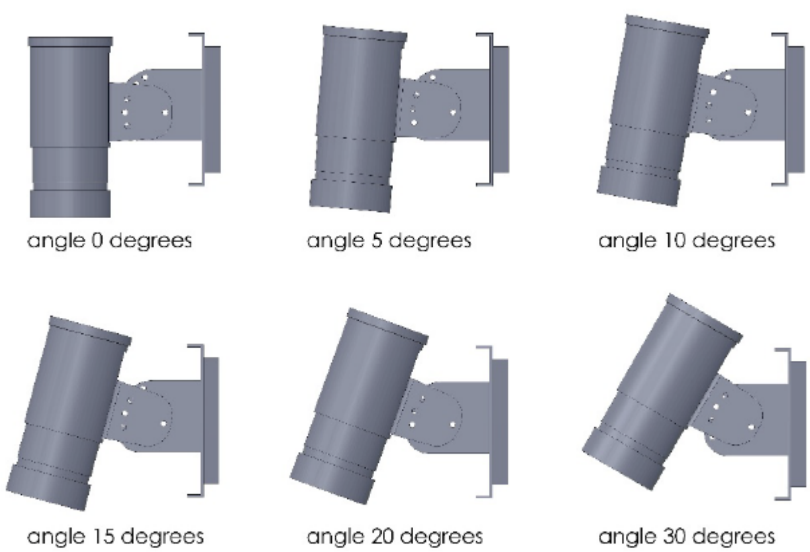


FIGURE 7-2. SDMS40 mounting angles

Table 7-1: SDMS40 mounting procedure	
Step	**Procedure**
1	Attach the L-shaped mount to the flat back mount using the common hole and crescent shaped screw holes.
2	Using FIGURE 7-2 (p. 7), decide which angle your sensor is to be mounted at.
3	Bolt the L-shaped mounting piece to the underside of the sensor. The big middle circle should line up with the cable connector.
4	Install sensor and mount 1 m above the maximum seasonal snow depth height. For mounting to poles, use the provided hose clamps.
5	Line up the connector end of the cable to the cable connector on the sensor. Lightly push the connector into place and screw the connector to secure.

7.1.1 Adjusting inclination angle or direction of the SDMS40

The SDMS40 can be installed at any angle between 0 and 45 degrees from the pole. After loosely tightening the screw on the common hole (FIGURE 7-1 (p. 7)), the inclination angle can be adjusted in 5 degree increments by matching one of the six holes on the flat backed mount attached to the pole (FIGURE 7-2 (p. 7)). Use the second screw to fix the inclination angle by tightening the screw through the SDMS40 part and the bracket. Completely tighten the common hole screw. See FIGURE 7-2 (p. 7) for mounting angle options.

7.2 Wiring

The SDMS40 requires a 12 to 15 Vdc power supply capable of providing up to 2 A continuously. Ensure the power is turned off before connecting the sensor.

> CAUTION:
> To avoid damage to the instrument, ensure that the power is turned off while wiring.

Align markers on the male and female connectors to plug in the cable to the sensor. The cable comes with a 1-Position Terminal Connector attached to the blue wire and another 1-Position Terminal Connector attached to the yellow wire (FIGURE 7-3 (p. 9)).

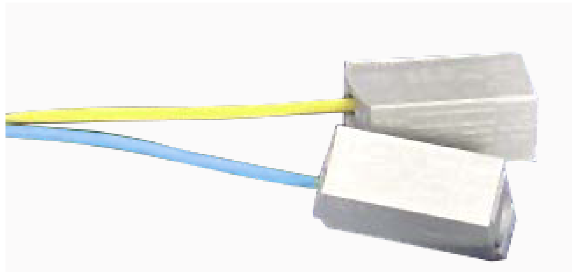


FIGURE 7-3. 1-Position Terminal Connectors individually isolate wires not used for SDI-12 measurements

If using RS-232, remove the connectors from the blue and yellow wires and attach the green wire to one of the connectors (FIGURE 7-4 (p. 9)).

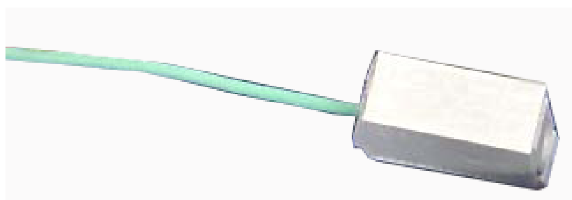


FIGURE 7-4. Position Terminal Connector isolates wire not used for RS-232 measurements

Table 7-2 (p. 9) provides SDI-12 wiring and Table 7-3 (p. 10) provides RS-232 wiring. Once mounting and wiring of the SDMS40 are complete, apply power to the SDMS40.

Table 7-2: SDI-12 sensor wire colour, function, and data logger connection

Wire colour	Function	Power supply terminal	Data logger connection terminal
Black	Power ground	**G**	
Red	Power	**12V**	
Green	SDI-12 signal		**C** or **U**[1] terminal configured for SDI-12
Brown	Signal ground		⏚ (analogue ground)

[1] **U** or **C** terminals are automatically configured by the measurement instruction.

If multiple SDI-12 sensors are connected to a data logger, Campbell Scientific recommends using separate terminals when possible. However, multiple SDI-12 sensors can connect to the same data logger control or **U** terminal. Each must have a unique SDI-12 address. Valid addresses are 0 through 9, a through z, and A through Z.

For the CR6 and CR1000X, triggering conflicts may occur when a companion terminal is used for a triggering instruction such as `TimerInput()`, `PulseCount()`, or `WaitDigTrig()`. For example, if the SDMS40 is connected to C3 on a CR1000X, C4 cannot be used in the `TimerInput()`, `PulseCount()`, or `WaitDigTrig()` instructions.

Table 7-3: RS-232 sensor wire colour, function and data logger connections

Wire colour	Function	Power supply terminal	Data logger connection terminal
Red	Power	**12V**	
Black	Power ground	**G**	
Brown	Signal ground		⏚ (analogue ground)
Blue	RS-232 Rx		**C** (odd numbered)
Yellow	RS-232 Tx		**C** (even numbered)

8. Programming

Short Cut is the best source for up-to-date data logger programming code. If your data acquisition requirements are simple, you can probably create and maintain a data logger program exclusively with Short Cut. If your data acquisition needs are more complex, the files that Short Cut creates are a great source for programming code to start a new program or add to an existing custom program.

> **NOTE:**
> Short Cut cannot edit programs after they are imported and edited in CRBasic Editor.

A Short Cut tutorial is available in Quickstart (p. 2). If you wish to import Short Cut code into CRBasic Editor to create or add to a customized program, follow the procedure in Importing Short Cut code into CRBasic Editor (p. 22). Programming basics for CRBasic data loggers are provided in the following section. Complete program examples for select CRBasic data loggers can be found in Example programs (p. 24).

8.1 SDI-12 programming

The `SDI12Recorder()` instruction is used to measure an SDMS40 configured for SDI-12 measurements. This instruction sends a request to the sensor to make a measurement and then retrieves the measurement from the sensor. See SDI-12 sensor measurements (p. 12) for more information. SDI-12 program (p. 24) provides a complete CRBasic program that measures the sensor.

For most data loggers, the `SDI12Recorder()` instruction has the following syntax:

```
SDI12Recorder(Destination, SDIPort, SDIAddress, "SDICommand", Multiplier,
Offset, FillNAN, WaitonTimeout)
```

For the `SDIAddress`, alphabetical characters need to be enclosed in quotes (for example, "A"). Also enclose the `SDICommand` in quotes as shown. The `Destination` parameter must be an array. The required number of values in the array depends on the command (see Table 9-1 (p. 12)).

`FillNAN` and `WaitonTimeout` are optional parameters (refer to CRBasic Help for more information).

8.2 RS-232 programming

The RS-232 instruction sequence is shown in the following table. Example programs are provided in RS-232 program (p. 27).

Table 8-1: RS-232 instruction sequence

Instruction	Function
`SerialOpen()`	Set up a data logger terminal for serial communications.
`SerialFlush()`	Clears the buffer.
`Scan()`	Establish a scan rate.
`SerialOut()`	Send RS-232 command to the sensor Table 10-2 (p. 21).
`SerialIn()`	Set up the COM terminal to receive the incoming serial data. Please note that in the beginning of the CRBasic program, the variable used in the `SerialIn()` instruction needs to be declared as an ASCII string format.
`SplitStr()`	Split out digital count value from the input string.

9. Operation

9.1 SDI-12 sensor measurements

The SDMS40 responds to the SDI-12 commands shown in Table 9-1 (p. 12). When using an `M!` command, the data logger waits for the time specified by the sensor, sends the `D!` command, pauses its operation, and waits until either it receives the data from the sensor or the sensor timeout expires. If the data logger receives no response, it will send the command a total of three times, with three retries for each attempt, or until a response is received. Because of the delays this command requires, it is only recommended in measurement scans of 10 seconds or more or use `SlowSequence`.

A `C!` command follows the same pattern as an `M!` command with the exception that it does not require the data logger to pause its operation until the values are ready. Rather, the data logger picks up the data with the `D!` command on the next pass through the program. Another measurement request is then sent so that data is ready on the next scan.

The `R0!` command directly reads the sensor measurements and outputs all of its values.

NOTE:
This section briefly describes using the SDI-12 commands. Additional SDI-12 information is available in SDI-12 sensor support (p. 31), or at www.sdi-12.org.

Table 9-1: SDI-12 command list

Commands[1]	Responses	Remarks
`a!`	a<CR><LF>	Acknowledge active.
`aI!`	"system info" <CR><LF>	SDI012 version, manufacturer, model, firmware version (e.g. 013wtherpiaSDMS40v6.111-24-2016).
`aV!`	"test result" <CR><LF>	System verification details.
`aAB!`	B<CR><LF>	Change address.
`A?!`	a<CR><LF>	Query sensor address.
`aM!`	0501<CR><LF>	Start measurement. Average depth value will be provided by **`aD0!`** following a service request.

Table 9-1: SDI-12 command list		
Commands[1]	Responses	Remarks
aC1!	05041<CR><LF>	Start concurrent measurement. Average depth and individual depth data at each sample point will be provided by **aD0!** through **aD8!**.
aD0!	depth<CR><LF>	Average depth.
aD1! thru aD8!	36 individual sample data	Grouped in 8 packets.
aR0!	+depth<CR><LF>	Similar to **aD0** for continuous measurement mode.
aXA!	0601	Perform automatic calibration to determine installation angle and height.
aXTxx! Where xx is the desired threshold	xx<CR><LF>	Set the heater threshold value (default is 0 °C). The heater will turn on when the internal temperature drops below this value and will remain on until the temperature climbs above the threshold. The allowed values range from –40 to 10 °C.
aXHxx Where xx is the height in mm	xx<CR><LF>	Manually set the current sensor height in mm. This option would only be used if the automatic calibration fails due to problems in the target area.
aXGxx!	0501	Reset the sensor ground level. This command would be used with the offset if there is existing snow on the ground when the sensor is installed.

[1] "a" refers to the address of the sensor.

9.2 RS-232 sensor measurements

The SDMS40 measures the current snow depth at a user-programmable interval (minutes) and transmits data on its RS-232 serial data lines to an external device, such as a data logger. By default, the sensor is in polling mode, where measurements are triggered by request from a data logger.

Table 9-2 (p. 14) is a list of RS-232 commands in the command mode.

Table 9-2: RS-232 command list		
Command usage	**Default value**	**Function**
@v (x) ⏎	1	Set the verbose level: 0 – none/1 – show information. Keep the verbose level at 1 if using the RS-232 example program.
@i (x) ⏎	0 (polling mode)	Set measurement interval in minutes. The sensor should be kept in polling mode for use with the sample RS-232 data logger program. Allowed values: 1, 2, 3, 4, 5, 6, 10, 20, 30, and 60.
@h (x)⏎	2000 mm	Manually set the current gauge height in millimeters. Use this command only if automatic calibration fails.
@g (x)⏎ Where (x) is the existing snow depth	n/a	Reset the sensor ground level. This command would be used with the offset if there is existing snow on the ground when the sensor is installed.
@m⏎	n/a	Run a round of measurements immediately.
@s⏎	n/a	SDMS40 status. This shows various system information such as current firmware version and installation angle and height.
@ac⏎	n/a	Perform automatic calibration to determine installation height and angle.
@history⏎	n/a	Show measurement data from the last 24 hours.
@lowtempx⏎ Where x is the desired threshold	0°C	To check the current threshold value, type "@lowtemp" ⏎.To modify the heater threshold value, include the value x. The heater will turn on when the internal temperature drops below this value and will remain on until the temperature climbs above the threshold. Allowed values range from -40°C to 10°C.
@b (x)⏎ Where x is an index for the desired baud rate	0 (57,600 bps)	Check or modify the baud rate of the serial port. To check current baud rate, type "@b"⏎. To modify the baud rate, include the desired index that follows. 0: 57600, 1: 38400, 2: 19200, 3: 9600, 4: 4800, 5: 2400, 6: 1200.
@d⏎	n/a	Check and modify the current SDMS-40 date * to modify the SDMS-40 date, type "@d" ⏎ and follow instructions.

Table 9-2: RS-232 command list		
Command usage	**Default value**	**Function**
@t⏎	n/a	Check and modify the current SDMS-40 time * to modify the SDMS-40 time, type "@t" ⏎ and follow the instructions.
@ct xx ⏎ Where XX is the new threshold	25	**Change Threshold CT** represents the difference in depth for two measurements in the same point
@vt⏎ Where XX is the new threshold	15	**Variance threshold VT** is the difference in depth between two adja-cent points in the same measurement

10. Maintenance and calibration

10.1 Maintenance

Regular cleaning and inspection is required:

- Check to make sure the target area is free from any obstacles or foreign material.
- Remove any dust or foreign deposits from the window of the SDMS40. Clean the window glass with soft cleaning fabric or tissues, water, and soft cleaning detergents.
- Inspect the bracket and other mounting clamps for loosened screws or clamps.

10.2 Calibration

Once the SDMS40 is fully installed, calibrate it for proper operation. Calibration sets the height and angle of the sensor to ensure accurate measurements. This occurs automatically (Automatic calibration (p. 15)) or manually (Manual calibration (p. 16)).

10.2.1 Automatic calibration

SDMS40 supports a fully automatic calibration process, which calculates the height and inclination angle of the SDMS40. Calibration is required when you first set up the sensor and when you move the sensor. This is done by issuing a calibration request command (SDI-12 **aXA!** or RS-232 **@ac**). When using the RS-232 command, the sensor will ask to confirm the request. Enter **y** to proceed.

10.2.2 Manual calibration

Manual calibration is only required if automatic calibration fails. After installation, enter the height of the sensor and run a ground level resetting procedure.

Entering the height can be done by sending the appropriate command (SDI-12 **aXHxxxx!** or RS-232 **@h xxxx**, where xxxx is the sensor height). When using the RS-232 command, the sensor will ask to confirm the height. Enter **y** to proceed.

After entering the sensor height, initiate a ground level reset (SDI-12 **aXG!** or RS-232 **@g**). When using the RS-232 command, the sensor will ask to confirm the request. Enter **y** to proceed.

10.3 Updating operating system

To update the sensor operating system, download the operating system available from: www.campbellsci.eu/sdms40. A terminal emulator software such as PuTTY Portable is required to do this update. PuTTY Portable is available at no charge from https://portableapps.com/apps/internet/putty_portable.

Table 10-1: Updating operating system procedure

Step	Procedure
1	Wire the SDMS40 sensor to a DB9 female terminal block (see Table 10-2 (p. 21)).
2	Connect the DB9 female to your computer RS-232 port using a standard serial cable or to a USB port using a serial-to-USB adapter.
3	Extract the operating system .zip file (sdms-firmware-update-utility.zip) downloaded from the website and open PuTTYPortable.exe.

Table 10-1: Updating operating system procedure

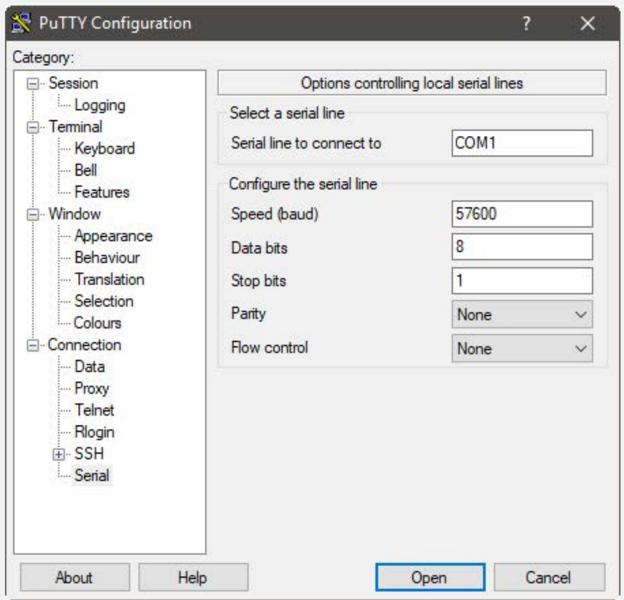
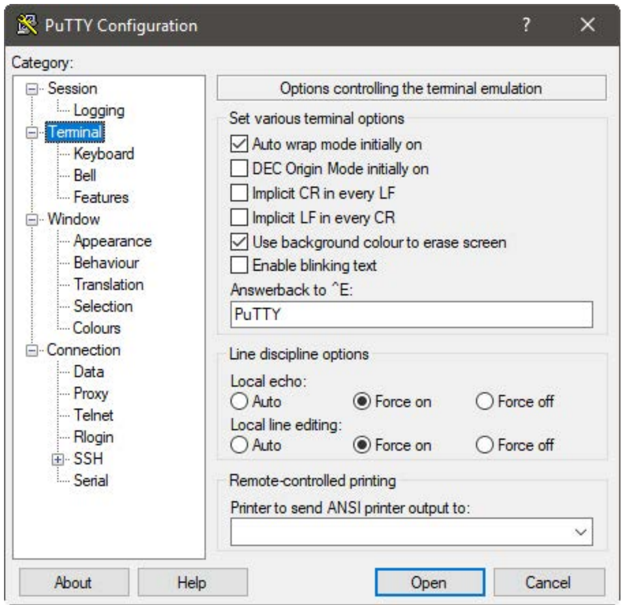
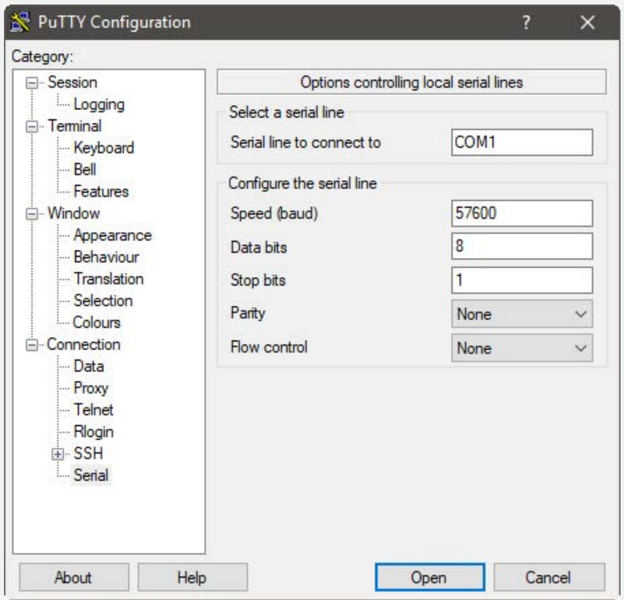
Step	Procedure
4	Select the **Session** Category and set the parameters to the following: a. **Select a serial line:** Use the COM port assigned to your serial cable. b. **Speed (baud):** 57600 c. **Data bits:** 8 d. **Stop bits:** 1 e. **Parity:** None f. **Flow control:** None 

Table 10-1: Updating operating system procedure

<table>
<tr><th>Step</th><th>Procedure</th></tr>
<tr><td>5</td><td>Select the Terminal Category and select the Force on option for both the Local echo and Local line editing parameters.



</td></tr>
</table>

Table 10-1: Updating operating system procedure	
Step	**Procedure**
6	Select the **Serial** Category and set the parameters to the following: a. **Select a serial line:** Use the COM port assigned to your serial cable. b. **Speed (baud):** 57600 c. **Data bits:** 8 d. **Stop bits:** 1 e. **Parity:** None f. **Flow control:** None 
7	Click the **Open** button.
8	Once the PuTTY terminal emulator opens, apply power to the sensor.
9	Let the sensor initialize and perform its initial measurements. Proceed to step 10 after **@@systemready@** is displayed.
10	Type **@b 0** and press the return key.

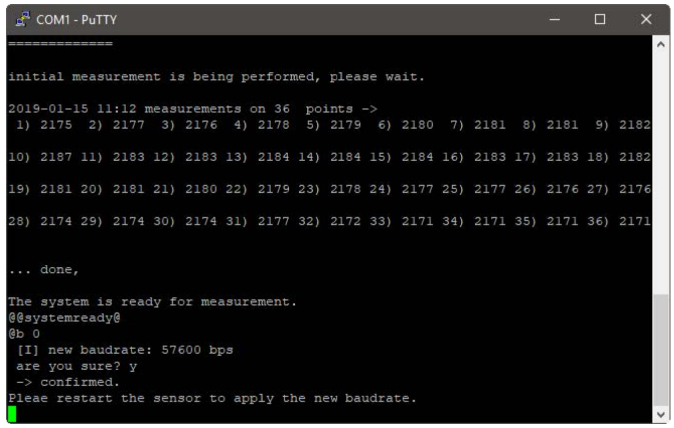
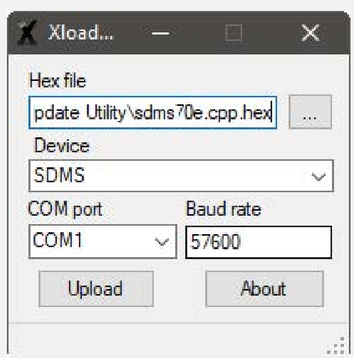
<table>
<tr><th colspan="2">Table 10-1: Updating operating system procedure</th></tr>
<tr><th>Step</th><th>Procedure</th></tr>
<tr><td>11</td><td>Confirm the new baud rate of 57600 bps by typing y (lower case) then pressing the return key. If successful, the message -> confirmed. is displayed in the terminal window.

</td></tr>
<tr><td>12</td><td>Run the Xloader.exe program from the folder and set the parameters to the following:
a. Hex file: press the ... button and select the *.cpp.hex operating system file from the folder.
b. Speed (baud): 57600
c. Device: SDMS
d. Com port: Use the COM port assigned to your serial cable.
e. Baud rate: 57600

</td></tr>
<tr><td>13</td><td>Click the Upload button. The message Uploading ... will appear at the bottom of Xloader.</td></tr>
<tr><td colspan="2">The operating system update may take a few minutes. Upon successful completion, an XXXXXX bytes uploaded message will appear.</td></tr>
</table>

Table 10-2: Operating system update wiring		
Colour	**Function**	**Connection**
White*	Operating system reset	Pin 4
Blue	RX	Pin 3
Yellow	TX	Pin 2
Brown	Ground	Pin 5
*Only use when resetting the operating system.		

Table 10-3: Power wiring		
Colour	**Function**	**Connection**
Red	Power	**12 V**
Black	Power Ground	**G**

Appendix A. Importing Short Cut code into CRBasic Editor

This tutorial shows:

- Importing a Short Cut program into a program editor for additional refinement
- Importing a wiring diagram from Short Cut into the comments of a custom program

Short Cut creates files, which can be imported into CRBasic Editor. Assuming defaults were used when Short Cut was installed, these files reside in the C:\campbellsci\SCWin folder:

- .DEF (wiring and memory usage information)
- .CR2 (CR200(X)-series datalogger code)
- .CR300 (CR300-series datalogger code)
- .CR6 (CR6-series datalogger code)
- .CR8 (CR800-series datalogger code)
- .CR1 (CR1000 datalogger code)
- .CR1X (CR1000X-series datalogger code)
- .CR3 (CR3000 datalogger code)
- .CR5 (CR5000 datalogger code)

Import Short Cut code and wiring diagram into CRBasic Editor:

1. Create the Short Cut program following the procedure in Quickstart (p. 2). Finish the program. On the **Advanced** tab, click the **CRBasic Editor** button. The program opens in CRBasic with the name **noname.CR_**. Provide a name and save the program.

> **NOTE:**
> Once the file is edited with CRBasic Editor, Short Cut can no longer be used to edit the program it created.

2. The program can now be edited, saved, and sent to the data logger.
3. Import wiring information to the program by opening the associated .DEF file. By default, it is saved in the c:\campbellsci\SCWin folder. Copy and paste the section beginning with heading "–Wiring for CRXXX–" into the CRBasic program, usually at the head of the file. After pasting, edit the information such that an apostrophe (') begins each line. This character instructs the data logger compiler to ignore the line when compiling. You can

highlight several lines of CRBasic code then right-click and select **Comment Block**. (This feature is demonstrated at about 5:10 in the CRBasic | Features video.)

Appendix B. Example programs

B.1 SDI-12 program

Wiring for this example program is shown in Table B-1 (p. 24).

Table B-1: Wiring for SDI-12 example program

Wire colour	Function	CR1000X connection terminal
Black	Power ground	**G**
Red	Power	**SW12-1**
Green	SDI-12 signal	**C1**
Brown	Signal ground	⏚ (analogue ground)

CRBasic Example 1: CR1000X SDI-12 program to measure the SDMS40

```
'****************************************************************************
'SDMS40 SDI-12 Sample Program (CR1000X)
'****************************************************************************

SequentialMode

'****************************************************************************
' Constants
'****************************************************************************
Const SDMS40_Interval = 1 'in minutes
Const SDI12_PORT1 = C1
' C1 is the Communications port used for connection to SDMS40

'****************************************************************************
' Diagnostic variables
'****************************************************************************
Public PTemp, batt_volt
Units PTemp = deg C
Units batt_volt = volts

'****************************************************************************
' Sensor Variables
'****************************************************************************
Public Snow_Depth_Avg As Float
```

CRBasic Example 1: CR1000X SDI-12 program to measure the SDMS40

```
Units Snow_Depth_Avg = mm
Dim SDI_Calibrate_Return
Public Calibrate_Flag As Boolean
Public SDMS40_Snow_Depth_Points(36)
Units SDMS40_Snow_Depth_Points() = mm

'*************************************************************************
' Diagnostic Data Table
'*************************************************************************
'Daily diagnostic data table for troubleshooting purposes
DataTable (Diagnostic,True,-1)
  DataInterval (0,1,day,10)
  Sample (1,status.OSversion,String)
  Sample (1,status.ProgName,String)
  Sample (1,status.LithiumBattery,FP2)
  Sample (1,status.PakBusAddress,UINT2)
  Sample (1,status.Low12VCount,UINT2)
  Maximum (1,status.Battery,FP2,False,False)
  Minimum (1,status.Battery,FP2,False,False)
  Sample (1,status.CompileResults,String)
  Maximum (1,status.PanelTemp,FP2,False,False)
  Minimum (1,status.PanelTemp,FP2,False,False)
  Sample (1,status.ProgSignature,UINT4)
  Sample (1,status.StartTime,String)
  Sample (1,status.SkippedScan,UINT2)
  Sample (1,status.SkippedSystemScan,UINT2)
  Sample (1,status.VarOutOfBound,UINT2)
  Sample (1,status.WatchdogErrors,UINT2)
EndTable

'*************************************************************************
' Snow Depth Data Table
'*************************************************************************
DataTable(SnowDepth,1,-1)
  DataInterval (0,SDMS40_Interval,min,10)
  Sample(1, Snow_Depth_Avg,IEEE4)
  Sample(36,SDMS40_Snow_Depth_Points(),IEEE4)
EndTable

'*************************************************************************
' Subroutine: CalibrateSensors
' Description: Sends an auto-calibration command to the SDMS40.
'*************************************************************************
Sub CalibrateSensors
  SDI12Recorder(SDI_Calibrate_Return,SDI12_PORT1,0,"XA!",1,0,-1,1)
  Delay (1,10,Sec)
EndSub
```

CRBasic Example 1: CR1000X SDI-12 program to measure the SDMS40

```
'******************************************************************************
' Main Program
'******************************************************************************
BeginProg

  Scan (10,Sec,5,0)
    PanelTemp (PTemp,_60Hz)
    Battery (batt_volt)
    CallTable Diagnostic

    ' The user's programming for other sensors would go here in the main scan

  NextScan

  SlowSequence
  Scan (SDMS40_Interval,min,5,0)

    If Calibrate_Flag = true

      Calibrate_Flag = false
      Call CalibrateSensors

    Else
      ' Measure average snow depth
      SDI12Recorder (Snow_Depth_Avg,SDI12_PORT1,0,"M!",1.0,0,-1,1)
      ' Returning SDMS40 36 snow Depth Points by sending 0D1! thru 0D8! commands
      SDI12Recorder (SDMS40_Snow_Depth_Points(1) ,SDI12_PORT1,0,"D1!" _
      ,1.0,0,-1,1)
      SDI12Recorder (SDMS40_Snow_Depth_Points(6) ,SDI12_PORT1,0,"D2!", _
      1.0,0,-1,1)
      SDI12Recorder (SDMS40_Snow_Depth_Points(11) ,SDI12_PORT1,0,"D3!", _
      1.0,0,-1,1)
      SDI12Recorder (SDMS40_Snow_Depth_Points(16) ,SDI12_PORT1,0,"D4!", _
      1.0,0,-1,1)
      SDI12Recorder (SDMS40_Snow_Depth_Points(21) ,SDI12_PORT1,0,"D5!", _
      1.0,0,-1,1)
      SDI12Recorder (SDMS40_Snow_Depth_Points(26) ,SDI12_PORT1,0,"D6!", _
      1.0,0,-1,1)
      SDI12Recorder (SDMS40_Snow_Depth_Points(31) ,SDI12_PORT1,0,"D7!", _
      1.0,0,-1,1)
      SDI12Recorder (SDMS40_Snow_Depth_Points(36) ,SDI12_PORT1,0,"D8!", _
      1.0,0,-1,1)

    EndIf

    CallTable(SnowDepth)
```

CRBasic Example 1: CR1000X SDI-12 program to measure the SDMS40

```
  NextScan
EndProg
```

B.2 RS-232 program

Wiring for this example program is shown in Table B-2 (p. 27).

Table B-2: Wiring for RS-232 example program

Wire colour	Function	CR1000X connection terminal
Red	Power	SW12-1
Black	Power ground	G
Brown	Signal ground	⏚ (analogue ground)
Blue	RS-232 Tx (output)	C1
Yellow	RS-232 Rx (input)	C2

CRBasic Example 2: CR1000X RS-232 program to measure the SDMS40

```
'***************************************************************************
'SDMS40 RS232 Sample Program (CR1000X)
'***************************************************************************
SequentialMode
'***************************************************************************
'User entered constants
'***************************************************************************
Const SDMS40_Interval = 1 'measurement and data output interval (in minutes)
Const SDMS40_COMport = ComC1
'Communications port used for connection to SDMS40
Const SDMS40_baud_rate = 57600
'Diagnostic variables
'***************************************************************************
Public PTemp, batt_volt
Units PTemp = deg C
Units batt_volt = volts
'***************************************************************************
'Variables for WeatherPia SDMS40 Scanning Laser Sensor
'***************************************************************************
Public SDMS40_Measure_Now As Boolean
'the user can set this to TRUE to request a measurement
```

CRBasic Example 2: CR1000X RS-232 program to measure the SDMS40

```
Public SDMS40_Calibrate_Now As Boolean
'the user can set this to TRUE to calibrate the sensor
Public SDMS40_Install_Height
Units SDMS40_Install_Height = mm
ReadOnly SDMS40_Install_Height
Public SDMS40_Install_Angle
Units SDMS40_Install_Angle = degrees
ReadOnly SDMS40_Install_Angle
Public SDMS40_Depth_Avg
Units SDMS40_Depth_Avg = mm
Public SDMS40_Temperature(3)
Units SDMS40_Temperature () = Deg C
Alias SDMS40_Temperature (1) = SDMS40_Board_Temperature
Alias SDMS40_Temperature (2) = SDMS40_Heater_Low_Threshold_Temperature
Alias SDMS40_Temperature (3)= SDMS40_Laser_Temperature
Public Verbose_Mode
Public SDMS40_Depth_Points(36)
Units SDMS40_Depth_Points() = mm
Public SDMS40_Distance_Points(36)
Units SDMS40_Distance_Points() = mm
Dim SDMS40_string As String * 2000
'string to hold data string received from SDMS40
Dim SDMS40_string_temp As String * 2000
Dim SDMS40_Serial_Check
'**************************************************************************
'Snow depth data table
'**************************************************************************
DataTable(SnowDepth,1,-1)
DataInterval (0,SDMS40_Interval,Min,10)
Sample(1,SDMS40_Depth_Avg,FP2)
Sample(3,SDMS40_Temperature(),FP2)
Sample(36,SDMS40_Depth_Points(),FP2)
Sample(36,SDMS40_Distance_Points(),FP2)
EndTable
'Main Program
BeginProg
'Open COM port for SDMS40
SerialOpen (SDMS40_COMport,SDMS40_baud_rate,0,10,2000)
SerialFlush (SDMS40_COMport)
' Force the verbose mode to 1
SerialOut (SDMS40_COMport,"@v 1" + CHR(13),"",0,0)
SerialIn (SDMS40_string,SDMS40_COMport,10,"",2000)
SplitStr (Verbose_Mode,SDMS40_string,"",1,0)
SerialFlush (SDMS40_COMport)
'Retrieve install angle and height from the sensor
SerialOut (SDMS40_COMport,"@s" + CHR(13),"",0,0)
SerialIn (SDMS40_string,SDMS40_COMport,1000,"",2000)
```

CRBasic Example 2: CR1000X RS-232 program to measure the SDMS40

```
SplitStr(SDMS40_string_temp,SDMS40_string,"angle:",1,4)
SplitStr(SDMS40_Install_Angle,SDMS40_string_temp,"",1,0)
SplitStr(SDMS40_string_temp,SDMS40_string,"Height:",1,4)
SplitStr(SDMS40_Install_Height,SDMS40_string_temp,"",1,0)
Scan (10,Sec,5,0)
  PanelTemp (PTemp,_60Hz)
  Battery (batt_volt)
  'The user's programming for other sensors would go here in the main scan
NextScan
SlowSequence
Scan (1,min,5,0)
  If SDMS40_Calibrate_Now = true
    'Calibration process
    SDMS40_Calibrate_Now = false
    SerialFlush (SDMS40_COMport)
    ' Force the verbose mode to 1
    SerialOut (SDMS40_COMport,"@v 1" + CHR(13),"",0,10)
    SerialIn (SDMS40_string,SDMS40_COMport,10,"",2000)
    SplitStr (Verbose_Mode,SDMS40_string,"",1,0)
    SerialFlush (SDMS40_COMport)
    SDMS40_Serial_Check = SerialOut (SDMS40_COMport,"@ac" + CHR(13), _
    "are you sure?",2,50)
    If SDMS40_Serial_Check = 13 Then
      SDMS40_Serial_Check = SerialOut (SDMS40_COMport,"y" + CHR(13), _
      "confirmed.",2,50)
      SerialIn (SDMS40_string,SDMS40_COMport,1000,"",2000)
      SplitStr(SDMS40_string_temp,SDMS40_string,"Angle:",1,4)
      SplitStr(SDMS40_Install_Angle,SDMS40_string_temp,"",1,0)
      SplitStr(SDMS40_string_temp,SDMS40_string,"Height:",1,4)
      SplitStr(SDMS40_Install_Height,SDMS40_string_temp,"",1,0)
    EndIf
  Else
    If TimeIntoInterval(0,SDMS40_Interval,min)
    SDMS40_Measure_Now = true
    EndIf
    If SDMS40_Measure_Now = true Then
      SDMS40_Measure_Now = false
      SerialFlush (SDMS40_COMport)
      ' Force the verbose mode to 1
      SerialOut (SDMS40_COMport,"@v 1" + CHR(13),"",0,10)
      SerialIn (SDMS40_string,SDMS40_COMport,10,"",2000)
      SplitStr (Verbose_Mode,SDMS40_string,"",1,0)
      SerialFlush (SDMS40_COMport)
      'Send the measurement command
      SDMS40_Serial_Check = SerialOut (SDMS40_COMport,"@m" + CHR(13), _
      "measurements",2,50)
      'Receive and parse the response from the sensor
```

CRBasic Example 2: CR1000X RS-232 program to measure the SDMS40

```
      SerialIn (SDMS40_string,SDMS40_COMport,1000,"",2000)
      SplitStr (SDMS40_Depth_Avg,SDMS40_string,"[M]",1,4)
      SplitStr (SDMS40_string_temp,SDMS40_string,"[t]",1,4)
      SplitStr(SDMS40_Temperature(),SDMS40_string_temp,"",3,0)
      SplitStr(SDMS40_string_temp,SDMS40_string,"[P]",1,4)
      SplitStr(SDMS40_Depth_Points(),SDMS40_string_temp,"",36,0)
      SplitStr(SDMS40_string_temp,SDMS40_string,"[R]",1,4)
      SplitStr(SDMS40_Distance_Points(),SDMS40_string_temp,"",36,0)
    EndIf
  EndIf
  CallTable SnowDepth
 NextScan
EndProg
```

Appendix C. SDI-12 sensor support

C.1 Introduction

SDI-12, Serial Data Interface at 1200 baud, is a protocol developed to simplify sensor and data logger compatibility. Only three wires are necessary — serial data, ground, and 12 V. With unique addresses, multiple SDI-12 sensors can connect to a single SDI-12 terminal on a Campbell Scientific data logger.

This appendix discusses the structure of SDI-12 commands and the process of querying SDI-12 sensors. For more detailed information, refer to version 1.4 of the SDI-12 protocol, available at www.sdi-12.org.

For additional information, refer to the SDI-12 Sensors | Transparent Mode and SDI-12 Sensors | Watch or Sniffer Mode videos.

C.2 SDI-12 command basics

SDI-12 commands have three components:

- **Sensor address (a)** – a single character and the first character of the command. Use the default address of zero (0) unless multiple sensors are connected to the same port.
- **Command body** – an upper case letter (the "command"), optionally followed by one or more alphanumeric qualifiers.
- **Command termination (!)** – an exclamation mark.

An active sensor responds to each command. Responses have several standard forms and always terminate with <CR><LF> (carriage return and line feed). Standard SDI-12 commands are listed in Table C-1 (p. 31).

Table C-1: Campbell Scientific sensor SDI-12 command and response set

Name	Command	Response[1]
Acknowledge Active	`a!`	a<CR><LF>
Send Identification	`aI!`	allccccccccmmmmmmvvvxxx...xx <CR><LF>

Table C-1: Campbell Scientific sensor SDI-12 command and response set		
Name	**Command**	**Response[1]**
Start Verification	`aV!`	atttn <CR><LF>
Address Query	`?!`	a<CR><LF>
Change Address	`aAb!`	b<CR><LF>
Start Measurement	`aM!` `aM1!...aM9!`	atttn<CR><LF>
Start Measurement and Request CRC	`aMC!` `aMC1!...aMC9!`	atttn <CR><LF>
Start Concurrent Measurement	`aC!` `aC1!...aC9!`	atttnn<CR><LF>
Start Concurrent Measurement and Request CRC	`aCC!` `aCC1!...aCC9!`	atttnn<CR><LF>
Send Data	`aD0!...aD9!`	a<values><CR><LF> or a<values><CRC><CR><LF>
Continuous Measurement	`aR0!...aR9!`	a<values><CR><LF>
Continuous Measurement and Request CRC	`aRC0!...aRC9!`	a<values><CRC><CR><LF>
Extended Commands	`aXNNN!`	a<values><CR><LF>

[1] Information on each of these commands is given in the following sections.

C.2.1 Acknowledge active command (a!)

The Acknowledge Active command (`a!`) is used to test a sensor on the SDI-12 bus. An active sensor responds with its address.

C.2.2 Send identification command (aI!)

Sensor identifiers are requested by issuing command `aI!`. The reply is defined by the sensor manufacturer but usually includes the sensor address, SDI-12 version, manufacturer's name, and sensor model information. Serial number or other sensor specific information may also be included.

`aI!`	allccccccccmmmmmmvvvxxx...xx<CR><LF>
a	Sensor SDI-12 address

ll	SDI-12 version number (indicates compatibility)
cccccccc	8-character vendor identification
mmmmmm	6 characters specifying the sensor model
vvv	3 characters specifying the sensor version (operating system)
xxx…xx	Up to 13 optional characters used for a serial number or other specific sensor information that is not relevant for operation of the data logger
<CR><LF>	Terminates the response
Source: *SDI-12: A Serial-Digital Interface Standard for Microprocessor-Based Sensors* (see References).	

C.2.3 Start verification command (aV!)

The response to a Start Verification command can include hardware diagnostics, but like the **aI!** command, the response is not standardized.

Command: **aV!**

Response: *atttn<CR><LF>*

a = sensor address

ttt = time, in seconds, until verification information is available

n = the number of values to be returned when one or more subsequent **D!** commands are issued

C.2.4 Address query command (?!)

Command **?!** requests the address of the connected sensor. The sensor replies to the query with the address, *a*. This command should only be used with one sensor on the SDI-12 bus at a time.

C.2.5 Change address command (aAb!)

Multiple SDI-12 sensors can connect to a single SDI-12 terminal on a data logger. Each device on a single terminal must have a unique address.

A sensor address is changed with command **aAb!**, where *a* is the current address and *b* is the new address. For example, to change an address from *0* to *2*, the command is **0A2!**. The sensor responds with the new address *b*, which in this case is 2.

> **NOTE:**
> Only one sensor should be connected to a particular terminal at a time when changing addresses.

C.2.6 Start measurement commands (aM!)

A measurement is initiated with the `M!` command. The response to each command has the form *atttn<CR><LF>*, where

a = sensor address

ttt = time, in seconds, until measurement data is available. When the data is ready, the sensor notifies the data logger, and the data logger begins issuing **D** commands.

n = the number of values returned when one or more subsequent **D** commands are issued. For the **aM!** command, n is an integer from 0 to 9.

When the **aM!** is issued, the data logger pauses its operation and waits until either it receives the data from the sensor or the time, *ttt*, expires. Depending on the scan interval of the data logger program and the response time of the sensor, this may cause skipped scans to occur. In this case make sure your scan interval is longer than the longest measurement time (*ttt*).

Table C-2: Example aM! sequence

`0M!`	The data logger makes a request to sensor 0 to start a measurement.
00352<CR><LF>	Sensor 0 immediately indicates that it will return two values within the next 35 seconds.
0<CR><LF>	Within 35 seconds, sensor 0 indicates that it has completed the measurement by sending a service request to the data logger.
`0D0!`	The data logger immediately issues the first **D** command to collect data from the sensor.
0+.859+3.54<CR><LF>	The sensor immediately responds with the sensor address and the two values.

C.2.7 Start concurrent measurement commands (aC!)

A concurrent measurement (**aC!**) command follows the same pattern as the **aM!** command with the exception that it does not require the data logger to pause its operation, and other SDI-12 sensors may take measurements at the same time. The sensor will not issue a service request to notify the data logger that the measurement is complete. The data logger will issue the **aD0!** command during the next scan after the measurement time reported by the sensor has expired. To use this command, the scan interval should be 10 seconds or less. The response to each command has the form *atttn<CR><LF>*, where

a = the sensor address

ttt = time, in seconds, until the measurement data is available

nn = the number of values to be returned when one or more subsequent **D** commands are issued.

See the following example. A data logger has three sensors wired into terminal **C1**. The sensors are addresses X, Y, and Z. The data logger will issue the following commands and receive the following responses:

Table C-3: Example aC! sequence	
`XC!`	The data logger makes a request to sensor X to start a concurrent measurement.
X03005<CR><LF>	Sensor X immediately indicates that it will have 5 (05) values ready for collection within the next 30 (030) seconds.
`YC!`	The data logger makes a request to sensor Y to start a concurrent measurement.
Y04006<CR><LF>	Sensor Y immediately indicates that it will have 6 (06) values ready for collection within the next 40 (040) seconds.
`ZC!`	The data logger makes a request to sensor Z to start a concurrent measurement.
Z02010<CR><LF>	Sensor Z immediately indicates that it will have 10 values ready for collection within the next 20 (020) seconds.
`ZD0!`	After 20 seconds have passed, the data logger starts the process of collecting the data by issuing the first **D** command to sensor Z.
Z+1+2+3+4+5+6+7+8+9+10<CR><LF>	Sensor Z immediately responds with the sensor address and the 10 values.
`XD0!`	10 seconds later, after a total of 30 seconds have passed, the data logger starts the process of data from sensor X by issuing the first **D** command.
X+1+2+3+4+5<CR><LF>	The sensor immediately responds with the sensor address and the 5 values.

Table C-3: Example aC! sequence	
YD0!	Ten seconds later, after a total of 40 seconds have passed, the data logger starts the process of data from sensor Y by issuing the first **D** command.
Y+1+2+3+4+5+6<CR><LF>	The sensor immediately responds with the sensor address and the 6 values.

C.2.8 Start measurement commands with cyclic redundancy check (aMC! and aCC!)

Error checking is done by using measurement commands with cyclic redundancy checks (**aMC!** or **aCC!**). This is most commonly implemented when long cable lengths or electronic noise may impact measurement transmission to the data logger. When these commands are used, the data returned in response to **D** or **R** commands must have a cyclic redundancy check (CRC) code appended to it. The CRC code is a 16-bit value encoded within 3 characters appended before the <CR><LF>. This code is not returned in the data table but checked by the data logger as it comes. The code returned is based on the SDI-12 protocol. See the SDI-12 communication specification for version 1.3 available at www.sdi-12.org to learn more about how the CRC code is developed.

C.2.9 Stopping a measurement command

A measurement command (**M!**) is stopped if it detects a break signal. A break signal is sent by the data logger before most commands.

A concurrent measurement command (**C!**) is aborted when another valid command is sent to the sensor before the measurement time has elapsed.

C.2.10 Send data command (aD0! ... aD9!)

The Send Data command requests data from the sensor. It is issued automatically with every type of measurement command (**aM!**, **aMC!**, **aC!**, **aCC!**). When the measurement command is a**M!** or a**MC!**, the data logger issues the **aD0!** command once a service request has been received from the sensor. When the data logger is issuing concurrent commands (**aC!** or **aCC!**), the Send Data command is issued after the required time has elapsed (no service request will be sent by the sensor). In transparent mode (SDI-12 transparent mode (p. 37)), the user asserts this command to obtain data.

Depending on the type of data returned and the number of values a sensor returns, the data logger may need to issue **aD0!** up to **aD9!** to retrieve all data. A sensor may return up to 35 characters of data in response to a **D** command that follows an **M!** or **MC!** command. A sensor may return up to 75 characters of data in response to a **D** command that follows a **C!** or **CC!** command.

Command: **aD0!** (**aD1!** ... **aD9!**)

Response: *a*<*values*><*CR*><*LF*> or *a*<*values*><*CRC*><*CR*><*LF*>

where:

a = the sensor address

<*values*> = values returned with a polarity sign (+ or –)

<*CR*><*LF*> = terminates the response

<*CRC*> = 16-bit CRC code appended if data was requested with a**MC!** or **aCC!**.

C.2.11 Continuous measurement command (aR0! ... aR9!)

Sensors that are able to continuously monitor the phenomena to be measured can be read directly with the **R** commands (**R0!** ... **R9!**). The response to the **R** commands mirrors the Send Data command (**aD0!**). A maximum of 75 characters can be returned in the <*values*> part of the response to the **R** command.

C.2.12 Extended commands

Many sensors support extended SDI-12 commands. An extended command is specific to a make of sensor and tells the sensor to perform a specific task. They have the following structure. Responses vary from unit to unit. See the sensor manual for specifics.

Command: **aXNNNN!**

The command will start with the sensor address (**a**), followed by an **X**, then a set of optional letters, and terminate with an exclamation point.

Response: *a*<*optional values*><*CR*><*LF*>

The response will start with the sensor address and end with a carriage return/line feed.

C.3 SDI-12 transparent mode

System operators can manually interrogate and enter settings in probes using transparent mode. Transparent mode is useful in troubleshooting SDI-12 systems because it allows direct communication with probes. Data logger security may need to be unlocked before activating the transparent mode.

Transparent mode is entered while the PC is communicating with the data logger through a terminal emulator program. It is accessed through Campbell Scientific data logger support software or other terminal emulator programs. Data logger keyboards and displays cannot be used.

The terminal emulator is accessed by navigating to the **Datalogger** list in PC200W, the **Tools** list in PC400, or the **Datalogger** list in the **Connect** screen of LoggerNet.

Watch the video: SDI-12 Sensors | Transparent Mode.

The following examples show how to enter transparent mode and change the SDI-12 address of an SDI-12 sensor. The steps shown in Changing an SDI-12 address (p. 38) are used with most Campbell Scientific data loggers. Changing an SDI-12 address — CR200(X) Series (p. 39) lists the steps used for CR200(X)-series dataloggers.

C.3.1 Changing an SDI-12 address

The example below was done with a CR1000, but the steps are only slightly different for CR1000X-series, CR300-series, CR6-series, CR800-series, and CR3000 dataloggers. For CR200(X)-series dataloggers, see Changing an SDI-12 address — CR200(X) Series (p. 39).

1. Connect an SDI-12 sensor to the CR1000.
2. In **LoggerNet Connect**, under **Datalogger**, click **Terminal Emulator**. The terminal emulator window opens.
3. Under **Select Device**, located in the lower left side of the window, select the **CR1000** station.
4. Click **Open Terminal**.
5. Select **All Caps Mode**.
6. Press Enter until the data logger responds with the **CR1000>** prompt.
7. Type **SDI12** and press Enter.
8. At the **Select SDI12 Port** prompt, type the number corresponding to the control port where the sensor is connected and press Enter. The response **Entering SDI12 Terminal** indicates that the sensor is ready to accept SDI-12 commands.
9. To query the sensor for its current SDI-12 address, type **?!** and press Enter. The sensor responds with its SDI-12 address. If no characters are typed within 60 seconds, the mode is exited. In that case, simply type **SDI12** again, press Enter, and type the correct control port number when prompted.
10. To change the SDI-12 address, type **aAb!**, where **a** is the current address from the above step and **b** is the new address (see FIGURE C-1 (p. 39). Press Enter. The sensor changes its address and responds with the new address.
11. To exit SDI-12 transparent mode, click **Close Terminal**.

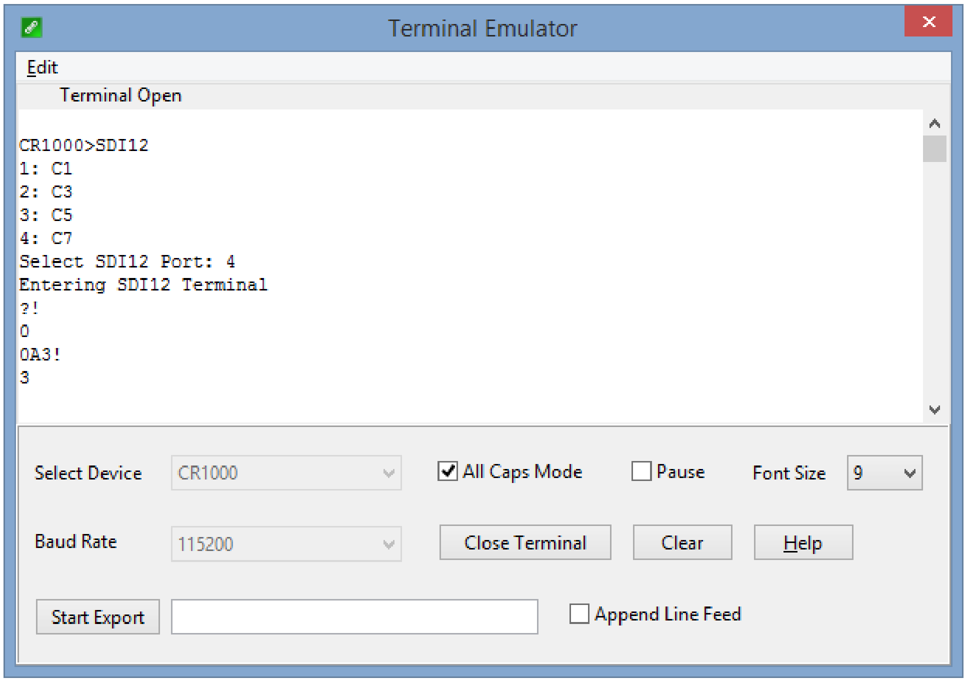


FIGURE C-1. CR1000 example of using the SDI-12 transparent mode to change the SDI-12 address from 0 to 3. Sensor is connected to control port 1.

C.3.2 Changing an SDI-12 address — CR200(X) Series

1. Connect a single SDI-12 sensor to the CR200(X).
2. In **LoggerNet Connect**, under **Datalogger**, click **Terminal Emulator**. The terminal emulator window opens.
3. Under **Select Device**, located in the lower left side of the window, select the **CR200Series** station.
4. Click **Open Terminal**.
5. Select **All Caps Mode**.
6. Press Enter until the data logger responds with the **CR2XX>** prompt.
7. Type **SDI12** and press Enter.
8. The response **SDI12>** indicates that the sensor is ready to accept SDI-12 commands.
9. To query the sensor for its current SDI-12 address, type **?!** and press Enter. The sensor responds with its SDI-12 address. If no characters are typed within 60 seconds, the mode is exited. In that case, simply type **SDI12** again and press Enter.
10. To change the SDI-12 address, type **aAb!**, where **a** is the current address from the above step and **b** is the new address (see FIGURE C-2 (p. 40)). Press Enter. The sensor changes its address and responds with the new address.
11. To exit SDI-12 transparent mode, click **Close Terminal**.

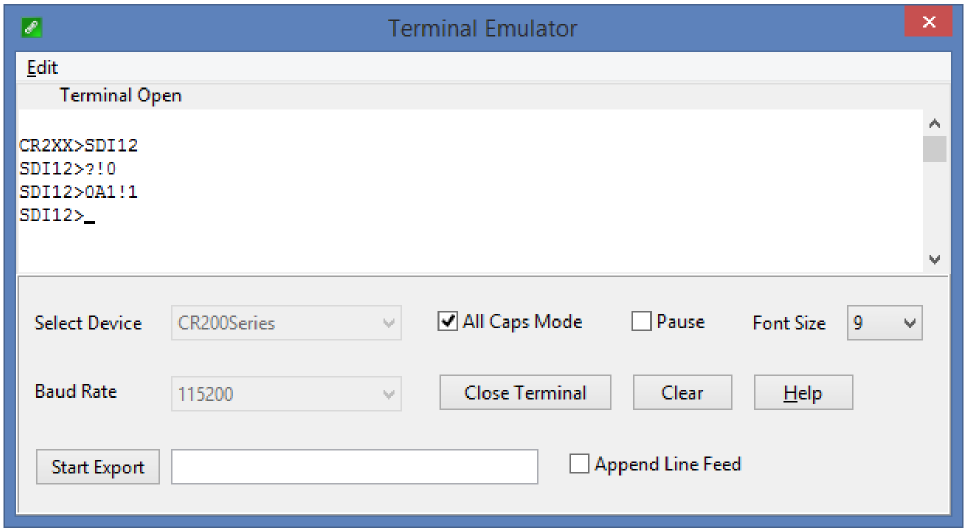


FIGURE C-2. CR200(X) example of using the SDI-12 transparent mode to change the SDI-12 address from 0 to 1

C.4 References

SDI-12 Support Group. SDI-12: A Serial-Digital Interface Standard for Microprocessor-Based Sensors – Version 1.4. River Heights, UT: SDI-12 Support Group, 2017. http://www.sdi-12.org/current_specification/SDI-12_version-1_4-Dec-1-2017.pdf.

Campbell Scientific Worldwide Offices

Australia
Location: Garbutt, QLD Australia
Email: *info@campbellsci.com.au*
Website: *www.campbellsci.com.au*

Brazil
Location: São Paulo, SP Brazil
Email: *andread@campbellsci.com.br*
Website: *www.campbellsci.com.br*

Canada
Location: Edmonton, AB Canada
Email: *dataloggers@campbellsci.ca*
Website: *www.campbellsci.ca*

China
Location: Beijing, P. R. China
Email: *info@campbellsci.com.cn*
Website: *www.campbellsci.com.cn*

Costa Rica
Location: San José, Costa Rica
Email: *info@campbellsci.cc*
Website: *www.campbellsci.cc*

France
Location: Antony, France
Email: *info@campbellsci.fr*
Website: *www.campbellsci.fr*

Germany
Location: Bremen, Germany
Email: *info@campbellsci.de*
Website: *www.campbellsci.de*

South Africa
Location: Stellenbosch, South Africa
Email: *sales@csafrica.co.za*
Website: *www.campbellscientific.co.za*

Southeast Asia
Location: Bangkok, Thailand
Email: *info@campbellsci.asia*
Website: *www.campbellsci.asia*

Spain
Location: Barcelona, Spain
Email: *info@campbellsci.es*
Website: *www.campbellsci.es*

UK
Location: Shepshed, Loughborough, UK
Email: *sales@campbellsci.co.uk*
Website: *www.campbellsci.co.uk*

USA
Location: Logan, UT USA
Email: *info@campbellsci.com*
Website: *www.campbellsci.com*

Please visit *www.campbellsci.com/contact* to obtain contact information
for your local US or international representative.