

INSTRUCTION MANUAL



SC115 CS I/O 2G Flash Memory Drive with USB Interface

Revision: 11/11



Copyright © 2010-2011
Campbell Scientific, Inc.

WARRANTY AND ASSISTANCE

This equipment is warranted by CAMPBELL SCIENTIFIC (CANADA) CORP. ("CSC") to be free from defects in materials and workmanship under normal use and service for **twelve (12) months** from date of shipment unless specified otherwise. ***** **Batteries are not warranted.** ***** CSC's obligation under this warranty is limited to repairing or replacing (at CSC's option) defective products. The customer shall assume all costs of removing, reinstalling, and shipping defective products to CSC. CSC will return such products by surface carrier prepaid. This warranty shall not apply to any CSC products which have been subjected to modification, misuse, neglect, accidents of nature, or shipping damage. This warranty is in lieu of all other warranties, expressed or implied, including warranties of merchantability or fitness for a particular purpose. CSC is not liable for special, indirect, incidental, or consequential damages.

Products may not be returned without prior authorization. To obtain a Return Merchandise Authorization (RMA), contact CAMPBELL SCIENTIFIC (CANADA) CORP., at (780) 454-2505. An RMA number will be issued in order to facilitate Repair Personnel in identifying an instrument upon arrival. Please write this number clearly on the outside of the shipping container. Include description of symptoms and all pertinent details.

CAMPBELL SCIENTIFIC (CANADA) CORP. does not accept collect calls.

Non-warranty products returned for repair should be accompanied by a purchase order to cover repair costs.



CAMPBELL SCIENTIFIC
C A N A D A C O R P .

11564 - 149 street - edmonton - alberta - T5M 1W7
tel 780.454.2505 fax 780.454.2655

www.campbellsci.ca

PLEASE READ FIRST

About this manual

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

Some useful conversion factors:

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

In addition, part ordering numbers may vary. For example, the CABLE5CBL is a CSI part number and known as a FIN5COND at Campbell Scientific Canada (CSC). CSC Technical Support will be pleased to assist with any questions.

SC115 Table of Contents

PDF viewers: These page numbers refer to the printed version of this document. Use the PDF reader bookmarks tab for links to specific sections.

1. Cautionary Statements	1
2. Quickstart	1
3. General Description	1
3.1 Memory Drive Function	2
3.2 Communications Interface Function.....	2
4. Specifications	2
5. Setup	3
5.1 SC115 Settings (Optional).....	3
5.2 PC Preparation.....	3
5.3 Datalogger CRBasic Programming	5
6. Data Storage Modes	6
6.1 Data Collection Modes	6
6.1.1 Programming for the Data-Collection Modes.....	6
6.1.2 Multiple Dataloggers	6
6.2 Resident Mode.....	7
6.2.1 Programming for the Resident Modes	7
7. Retrieving Data from SC115	7
8. Datalogger Programming	8
8.1 TableFile() Instruction	8
8.1.1 FileName.....	10
8.1.2 Options.....	10
8.1.3 MaxFiles	11
8.1.4 NumRecs/TimeIntoInterval Parameter	11
8.1.5 Interval Parameter	12
8.1.6 Units Parameter.....	12
8.1.7 OutStat Parameter	12
8.1.8 LastFileName Parameter.....	12
8.2 CardFlush Instruction	12
8.3 CardOut Instruction	13
8.4 Powerup.ini.....	13

8.5 Example Programs	13
8.5.1 Data Collection Mode Examples.....	13
8.5.1.1 Standard Data Collection Mode.....	13
8.5.1.2 Enhanced Data Collection Mode Example.....	14
8.5.2 Resident Interval Mode Example.....	14
8.5.3 Resident Record Number Mode Examples	15
8.5.4 CardFlush Example.....	16
8.5.5 CardOut Example.....	16
9. Serial Communications Interface Function.....	17
10. Troubleshooting Memory Drive.....	17
10.1 Preventing Data Corruption	17
10.2 Skipped Scans.....	18
10.3 SC115 with Large Data Compliment.....	18
10.4 Slow Data Collection.....	18
10.5 Slow PC Boot-up	18
10.6 Write Failure.....	18
11. Troubleshooting Communications Interface.....	19
11.1 Breaking the Physical Comms Link.....	19
12. Technical Reference	19
12.1 SDC Addressing	19
12.2 Formatting Memory	19
12.3 Operating System.....	19
13. Glossary.....	20

List of Tables

1. PC Driver Installation for SC115 as Memory Device.....	4
2. PC Driver Installation for SC115 as Communications Interface	5
3. Function of NumRec, TFInterval, and TFUnits Parameters	9
4. Synopsis of MaxFiles Parameter Options.....	11

SC115 CS I/O 2G Flash Memory Drive with USB Interface

1. Cautionary Statements

- Corruption of multiple data files may occur if the SC115 is removed from the host datalogger or PC during data transfer.
- Do not disconnect the SC115 from the host while the LED is flashing or lit. Always use the “Safely Remove Hardware” utility provided in the Windows operating system prior to removal from a PC.
- Some legitimate data collection modes increase the risk of inadvertently disconnecting the SC115 from the datalogger during data transfer. See Section 10.1 Preventing Data Corruption to assess the risk in a particular application.
- Always click *Disconnect* in the datalogger support software prior to removing the SC115 when it is a communications interface. See Section 9. Serial Interface Communications Function for communication interface details and precautions.

2. Quickstart

The primary function of the SC115 is as a portable memory drive (thumb drive) to collect data from compatible Campbell Scientific dataloggers (see Section 4. Specifications for compatibility). Simply connect the CS I/O connector of the SC115 to the CS I/O connector of a properly programmed datalogger, and data are transferred automatically.

Memory Drive Setup (see Section 5. Setup for complete instructions):

- SC115: shipped from the factory ready to use in most applications.
- PC: install software drivers on the PC. See Section 5.3 PC Preparation. Install drivers BEFORE plugging in the SC115. When drivers are installed, the SC115 acts just like any other thumb drive on the PC.
- Datalogger: insert the `TableFile()` instruction into the datalogger CRBasic program, as shown below, immediately following the `DataInterval()` instruction. This configuration will enable the SC115 to collect the newest data in the data collection mode (see Section 6.1). Note that “USB:” is the correct drive name.

```
TableFile ("USB:FileName",8,-1,0,0,Hr,0,0)
```

3. General Description

The SC115 is a portable 2-GB memory drive (thumb drive) compatible with CRBasic dataloggers that have a CS I/O port. It can be used in tandem with a CompactFlash (CF) card. It shuttles data, OS, and program files between

Campbell Scientific dataloggers and a PC, or it remains connected to the datalogger to augment data memory.

The SC115 can also be used as a CS I/O to USB communications interface.

The SC115 has a USB 2.0 compatible connector used to attach the SC115 to a PC USB port. It also has a CS I/O 9-pin connector that attaches the SC115 to a Campbell Scientific datalogger CS I/O port. The SC115 may be connected directly to a PC or datalogger, or connected through the supplied extension cables.

The 5V (pin 1) of the CS I/O interface determines whether the SC115 is used as a memory drive or as a CS I/O-to-USB communications interface. If the 5V pin is not driven high, it acts as a memory drive. If the 5V pin is driven high by the datalogger, the SC115 will serve as a communications interface.

3.1 Memory Drive Function

As a memory drive, when the SC115 is attached to the CS I/O port of a datalogger, the LED (Light Emitting Diode) first flickers as the SC115 and datalogger negotiate communications. The datalogger determines if the TableFile() instruction is set to write to an SC115. The LED flickers as the datalogger copies data files to the SC115. The LED quits flickering when data transfer is complete, and the SC115 can safely be disconnected.

As a memory drive, when the SC115 is attached to a PC, the LED first lights solidly as it communicates with the PC to obtain a unique address. Once negotiations are finished, the LED extinguishes and the SC115 appears as a removable drive in Windows® Explorer. Data files on the SC115 can be copied or moved to other drives. Files, such as CRBasic program files or operating system files, can be copied or moved from PC drives to the SC115.

3.2 Communications Interface Function

As a communication device, the SC115 provides pass-through communications.

4. Specifications

Compatible Dataloggers	CR800, CR850, CR1000, CR3000, CR5000, CR9000X
Storage Capacity	2 GB
Dimensions	105.41 x 43.18 x 21.2 mm (4.15 x 1.7 x 0.8 in)
Weight	63.79 g (2.25 oz)
Case	Sealed over-molded package. No serviceable components.
Power Requirements	12 VDC through the datalogger CS I/O port, or 5 VDC through PC USB port
Current Drain	200 µA (quiescent), 35 mA (active)
Operating Temperature Range	-25° to +50°C
Media	Flash
Mean Time Between Failure	> 1,000,000 Hours

5. Setup

A PC requires software drivers to support a SC115. The datalogger requires addition of some CRBasic code. In rare instances, the SC115 may require a change to its default configuration through Campbell Scientific's Device Configuration Utility Software v. 1.5 or higher (DevConfig).

5.1 SC115 Settings (Optional)

The default SDC addressing in the SC115 is adequate for most applications. For exceptional circumstances, the CS I/O SDC address can be altered using the SC115 service in DevConfig v. 1.5 or higher (see Section 12.1 SDC Addressing).

5.2 PC Preparation

The following procedures install required drivers to the PC. Install drivers BEFORE connecting the SC115 to the PC.

1. Obtain software drivers from one of the following sources.

Source 1: Insert the CD supplied with a new SC115. The CD should auto run and present a menu (if not, open AutoRun.exe). Click on [Install Drivers] button.

Source 2: Drivers can be accessed through the SC115 service of DevConfig.

Source 3: Obtain the SC115_Drivers.exe file at www.campbellsci.com, and copy it to a PC drive. Open the file and follow the prompts.

2. Follow the on-screen prompts to copy driver files to the PC.
3. Install drivers for SC115 as outlined in Table 1. PC Driver Installation for SC115 as Memory Device **or** Table 2. PC Driver Installation for SC115 as Communications Interface. Procedures differ for different Windows® operating systems and assume driver files have been copied to the PC.

TABLE 1. PC Driver Installation for SC115 as Memory Device		
Windows® XP	Windows® Vista	Windows® 7
<ol style="list-style-type: none"> 1. Insert the SC115 into a computer USB port, 2. The <i>Found New Hardware Wizard</i> window will open. Select “No, not this time,” when asked to connect to Windows® Update, then click <i>Next</i>. 3. Select “Install the software automatically,” then click <i>Next</i>. 4. If the <i>Windows® Security</i> window appears, select “Install this driver software anyway” to continue. 5. Click <i>Finish</i> to close the <i>found New Hardware Wizard</i> window. 	<ol style="list-style-type: none"> 1. Insert the SC115 into a computer USB port, 2. The <i>Found New Hardware</i> window will open. Click on “Locate and install driver software.” 3. If the <i>Windows® Security</i> window appears, select “Install this driver software anyway.” 	<ol style="list-style-type: none"> 1. Insert the SC115 into a computer USB port, 2. Windows® 7 typically configures the driver automatically. 3. If the drivers are not automatically installed, do steps 4 through 6. 4. Go to Windows Start Menu Control Panel Hardware and Sounds Device Manager (under the Devices and Printers category). 5. Under Other Devices, right click on the SC115. 6. Choose Update Drivers.

Windows® XP	Windows® Vista	Windows® 7
<ol style="list-style-type: none"> 1. Apply power to a Campbell Scientific datalogger, then attach the SC115 to the datalogger's CS I/O port. 2. Using the included USB extension cable, attach the SC115 to an open USB port on the computer. 3. The <i>Found New Hardware Wizard</i> window will open. Select "No, not this time," when asked to connect to Windows® Update, then click <i>Next</i>. 4. Select "Install the software automatically," then click <i>Next</i>. 5. If the <i>Windows® Security</i> window appears, select "Install this driver software anyway." 6. Click <i>Finish</i> to close the <i>found New Hardware Wizard</i> window. 	<ol style="list-style-type: none"> 1. Apply power to a Campbell Scientific datalogger, then attach the SC115 to the datalogger's CS I/O port. 2. Using the included USB extension cable, attach the SC115 to an open USB port on the computer. 3. The <i>Found New Hardware</i> window will open. Click on "Locate and install driver software." 4. If the <i>Windows® Security</i> window appears, select "Install this driver software anyway." 	<ol style="list-style-type: none"> 1. Apply power to a Campbell Scientific datalogger, then attach the SC115 to the datalogger's CS I/O port. 2. Using the included USB extension cable, attach the SC115 to an open USB port on the computer. 3. Windows® 7 typically configures the driver automatically. 4. If the drivers are not automatically installed, do steps 5 through 7. 5. Go to Windows Start Menu Control Panel Hardware and Sounds Device Manager (under the Devices and Printers category). 6. Under Other Devices, right click on the SC115. 7. Choose Update Drivers.

5.3 Datalogger CRBasic Programming

The datalogger CRBasic program needs to include a TableFile() instruction to store the datalogger's data on the SC115. The TableFile() instruction is entered in the declaration of the data table after the DataInterval() instruction. The parameters entered for the TableFile() instruction depend on the data storage mode used (refer to Section 6). Section 8 discusses, in detail, datalogger programming, and provides several datalogger programming examples.

6. Data Storage Modes

The datalogger CRBasic program determines the SC115 data storage mode through the TableFile() instruction. The TableFile() instruction in the datalogger CRBasic program determines what data the SC115 will receive from the datalogger. Data collection modes and residential modes are available.

6.1 Data Collection Modes

With these modes, data collection is automatically initiated by connecting the SC115 to the datalogger. The SC115 only collects the data already contained in datalogger's memory at the time of connection. Additional data will not be collected until the SC115 is disconnected and then reconnected to the datalogger.

Both standard and enhanced data-collection modes are available. With the standard mode, the SC115 collects the newest data (data written to the datalogger memory since the last connection). With the enhanced mode, the SC115 collects all of the data stored in the datalogger's memory every time the SC115 is reconnected.

NOTE To use the enhanced data collection mode, the datalogger must have a newer operating system (OS 22 or higher).

6.1.1 Programming for the Data-Collection Modes

The standard mode is enabled in the TableFile() instruction by entering 0 for both the *NumRecs* (Number of Records) parameter and the *Interval* parameter. The enhanced mode is enabled by entering 0 for the *NumRecs* parameter and entering -1 for the *Interval* parameter.

Below are TableFile() instructions that enable data collection modes. The first TableFile() instruction enables the standard data collection mode. The second TableFile() instruction enables the enhanced data collection mode. See Section 8.5.1 Data Collection Mode Example for more detailed program code.

```
TableFile ("USB:"+Status.SerialNumber+"_FileName",8,-1,0,0,Hr,0,0)
```

```
TableFile ("USB:"+Status.SerialNumber+"_FileName",8,-1,0,-1,Hr,0,0)
```

6.1.2 Multiple Dataloggers

The data-collection modes allow one SC115 to be used with multiple dataloggers. When using multiple dataloggers, specify a unique file name for each instance of the TableFile() instruction (see Section 8.1.1 FileName). This avoids overwriting data files from other dataloggers.

6.2 Resident Mode

NOTE Resident modes increase the risk of data corruption. Review Section 10.1 Preventing Data Corruption before employing a resident mode.

In the resident modes, the SC115 remains attached to a single datalogger allowing it to be used as resident external memory. The datalogger can be programmed to bale data to the SC115 at regular intervals or at uniform bale sizes.

When data retrieval is required, 1) a second SC115 is swapped with the first, or 2) the SC115 is removed, milked of data (perhaps by copying or moving data to a laptop computer), and then reconnected to the datalogger.

NOTE To avoid losing data when the SC115 is disconnected from the datalogger, the datalogger must have sufficient storage memory allocated in `DataTable()` / `EndTable` declarations in the form of internal or CompactFlash®¹ memory to cover the period the SC115 is not present.

6.2.1 Programming for the Resident Modes

A resident mode that collects data at regular intervals is enabled in the `TableFile()` instruction by entering zero or a positive value for the *NumRecs/TimeIntoInterval* parameter and entering a non-zero, positive value for the *Interval* parameter. To collect data at uniform bale sizes, enter a non-zero, positive value for the *NumRecs* parameter and enter zero for the *Interval* parameter.

Below are `TableFile()` instructions that enable resident modes. The first `TableFile()` instruction sets the datalogger to bale data to the SC115 at midnight each day. The second `TableFile()` instruction sets the datalogger to send bales of 24 records to the SC115. See Section 8.5.2 Resident Interval Mode Example for more detailed program code.

```
TableFile ("USB:FileName",8,-1,0,24,Hr,0,0) 'Interval trigger
```

```
TableFile ("USB:FileName",8,-1,24,0,Hr,0,0) 'Number of records trigger
```

7. Retrieving Data from SC115

Install PC software drivers BEFORE connecting the SC115. See Section 5.3 PC Preparation.

To collect data from the SC115, plug it into a PC USB port. The SC115 becomes a memory drive on the PC, and data files can be copied, moved, or deleted with the common file handling functions of Windows®. Multiple SC115s can be connected to a PC simultaneously and data passed between them. Connecting a second SC115 may require installing the device driver a second time. When used as a communications interface, however, only one SC115 should be connected to a PC.

Before disconnecting the SC115 from the PC, run the *Safely Remove Hardware* utility in Windows.

8. Datalogger Programming

When used as a memory drive, the SC115 requires a TableFile() instruction in the datalogger CRBasic program. TableFile() creates a file from data table records, and writes the file to the SC115. The TableFile() instruction must be within the DataTable() / EndTable declaration.

8.1 TableFile() Instruction

TableFile() syntax is reviewed below with comments specific to SC115 applications. For more detail, consult CRBasic Editor Help. TableFile() is included in the DataTable() / EndTable declaration as shown in the programming examples in Section 8.5 Example Programs.

CAUTION

DataTable() size parameter must be declared large enough to support TableFile() output to the SC115. TableFile() does not have its own memory. It can only use what is allocated for the table by the DataTable() instruction. For example, making the DataTable() size parameter = 0 to store data only to the SC115 will result in files with only 0 to 2 records and many missing records in the file.

TableFile (a,b,c,d,e,f,g,h)

- a** = "Drive:FileName." Drive always = USB; for SC115. Set a filename unique to each instance of TableFile() instruction. See Section 8.1.1 for required syntax.
- b** = Format. Most common format is code 8 (TOA5). See CRBasic Help for details.
- c** = MaxFiles. -1 invokes auto allocation.
- d** = NumRec / Time Into Interval. Data Collection Modes: enter 0. Resident Modes: enter number of records (integer > 0) or enter time into interval (integer ≥ 0). See Table 3 and Section 8.1.4 for explanation.
- e** = TFInterval. Data Collection Modes: enter 0 to collect newest data or enter -1 to collect all of the datalogger's data. Resident mode: enter 0 if parameter d provides number of records, otherwise enter time interval (integer > 0). See Table 3 and Section 8.1.5 for explanation.
- f** = TFUnits. Enter units for time interval. Ignored at compilation if e = 0 or -1.
- g** = OutStat. Optional. Set to 0 or see CRBasic Help.
- h** = LastFileName, Optional. Set to 0 or see CRBasic Help.

TABLE 3. Function of *NumRec*, *TFInterval*, and *TFUnits* Parameters

<i>NumRec/Time Into Interval</i> (d) Entry	<i>TFInterval</i> (e) Entry	<i>TFUnits</i> (f) Entry	Function
0	0	N/A (compiler ignores)	When the SC115 is connected to the datalogger, the datalogger automatically sends to the SC115 only the data collected after the SC115 was last connected. The datalogger will not send additional data if the SC115 remains connected to the datalogger.
0	-1	N/A (compiler ignores)	When the SC115 is connected to the datalogger, the datalogger automatically sends to the SC115 all of the data contained in the datalogger's memory. The datalogger will not send additional data if the SC115 remains connected to the datalogger.
number of data records (value > 0)	0	N/A (compiler ignores)	The datalogger sends the specified number of data records to the SC115 when the data records become available. For example, if the <i>NumRec</i> parameter is set to 20 and <i>TFInterval</i> is set to 0, the datalogger will write 20 records to the SC115 when 20 records become available.
time into interval (value ≥ 0)	interval (value > 0)	mSec, uSec, Sec, Min, Hr, or Day	The datalogger sends data to the SC115 at the specified interval. For example, when the <i>NumRec/Time Into Interval</i> parameter is set to 0, the <i>TFInterval</i> parameter is set to 60 and the <i>TFUnit</i> parameter is set to Min, the datalogger will write data to the SC115 every 60 minutes.

The following sections explain each TableFile() parameter in detail.

8.1.1 FileName

The FileName parameter must be a string declared as Const, such as

```
Const FileName = "USB:FileName",
```

or as an expression that evaluates to a constant, such as

```
"USB:"+Variable+"FileName"
```

Quotation marks are required. The created file will have a suffix of *X.dat*, where *X* is a number that increments each time a new file is written.

When using a single SC115 to collect data from several dataloggers, the TableFile() *FileName* parameter must be unique for each data table being collected. Otherwise, data may be overwritten on the SC115. Using a TableFile() instruction with parameters as shown below creates a naming scheme wherein the station name or serial number are part of the file name. This allows the source of data to be easily identified during post-processing.

```
TableFile ("USB:"+Status.SerialNumber+"Filename", 8, -1, 0, 0, Min, 0, 0)
```

When a program is compiled with “USB:” in the file path, it searches the attached SC115 for any file names in the series. If it finds any, it uses the highest numbered file name, increments it by 1, uses the result as the name for a new file, and writes the new data to the new file.

When multiple SC115s are used to retrieve data from multiple dataloggers, it is often desirable to set the interval parameter to -1. If the interval parameter is set to zero instead of -1, the complete set of records from a single datalogger will likely be spread across all SC115s used to retrieve data.

CAUTION

If data are retrieved from a datalogger using a second SC115 containing filenames matching those created by the current datalogger's TableFile(), the old files are likely to be overwritten. Using unique filenames for each datalogger, such as including the station name or serial number, is the best practice. A rarely used alternative is to recompile the datalogger program with the second SC115 connected to retrieve its directory information. This is done by powering down the datalogger, connecting the SC115, then powering up the datalogger.

8.1.2 Options

The Options parameter specifies the type of file to be saved and whether to include the header information, timestamp, and/or record number. Options 0, 8, 16, and 20 correspond to Campbell Scientific formats for TOB1, TOA5, CSIXML, and CSIJSON, respectively. Choosing an option not among these four may make the file incompatible with Campbell Scientific software that reads or writes data files. For example, Option 7 results in a TOB1 file that cannot be read by CardConvert or View Pro. Refer to the CRBasic Editor Help system for a complete listing of option codes.

8.1.3 MaxFiles

The MaxFiles parameter specifies the maximum number of files to retain on an SC115. Table 4. Synopsis of MaxFiles Parameter Options, reviews the options.

TABLE 4. Synopsis of MaxFiles Parameter Options	
MaxFiles Entry	Function
X	Create Maximum of X files, ring memory (overwrite oldest file with newest file when full).
0	The filename will remain fixed with no number appended. The old file if it exists will be overwritten at each output.
-1	No limit to number of files, ring memory (overwrite oldest file with newest file when full)
-2	No limit to number of files, fill and stop memory (datalogger stops writing to SC115 when SC115 is full)

When MaxFiles is X, SC115 memory may fill before X number of files is reached. If this occurs, the datalogger internally reduces X to the current number of files and overwrites the oldest file with new data.

Refer to the CRBasic Help system for detailed information on this parameter.

8.1.4 NumRecs/TimeIntoInterval Parameter

Data Collection Modes

For Data Collection Modes, the *NumRecs/TimeIntoInterval* parameter is set to zero, and the *Interval* parameter is set to either zero or -1 (see Section 8.1.5). With these modes, the datalogger begins writing to the SC115 as soon as it is connected to the datalogger. All new records are written to a single file. See Table 3 and Section 8.5.1 Data Collection Mode Example.

Resident Modes

NOTE

Resident Mode increases the risk of data corruption. Review Section 10.1 Preventing Data Corruption before employing a Resident Mode.

For Resident Modes, the *NumRecs/TimeIntoInterval* parameter determines when files are written to the SC115. The function of *NumRecs/TimeIntoInterval* is linked to the *Interval* parameter. If *Interval* is set to zero, enter the number of records to be included in each new file sent to the SC115. The program will create a new file each time *NumRecs* is reached. If *Interval* is set to non-zero, *NumRecs/TimeIntoInterval* becomes the time into *Interval* that the program writes the next file. Table 3 and CRBasic Editor Help provide further explanation of the interplay between *TimeIntoInterval* and *Interval*.

8.1.5 Interval Parameter

Data Collection Modes

For Data Collection Modes the *Interval* parameter determines what data are written to the SC115. Set this parameter to zero to collect only the data written to the datalogger memory since it was last connected. Set this parameter to -1 to collect all of the data stored in the datalogger's memory. See Table 3.

Resident Mode

NOTE

Resident Mode increases the risk of data corruption. Review Section 10.1 Preventing Data Corruption before employing a Resident Mode.

For Resident Mode, the *Interval* parameter is used to determine how frequently data are written to the SC115. By setting this parameter to a non-zero number, the datalogger writes a new file to the SC115 at the interval based on this number and the *Units* parameter.

If *Interval* is set to zero, but *NumRecs/TimeIntoInterval* is set to a non-zero value, the datalogger writes data to the SC115 whenever the number of new records matches the *NumRecs/TimeIntoInterval* value.

8.1.6 Units Parameter

The *Units* parameter specifies the units used by the *Interval* parameter. The compiler ignores this parameter if the *Interval* parameter is set to zero or -1.

8.1.7 OutStat Parameter

Set to 0 (no *OutStat*) if desired. The *OutStat* parameter is a variable that indicates whether or not a new file was stored the last time *TableFile()* executed. *OutStat* is updated the next time *CallTable()* instruction is executed. If files take longer to write than the *CallTable()* interval, then this variable will still not be updated.

8.1.8 LastFileName Parameter

Set to 0 (no *LastFileName*) if desired. The *LastFileName* parameter is a variable that contains the name of the last file written. *LastFileName* is updated the next time *CallTable()* instruction is executed. If files take longer to write than the *CallTable()* interval, then this variable will still not be updated.

8.2 CardFlush Instruction

If a program *Scan() / NextScan Count* parameter is non-zero, the latest data may fail to write to the SC115 unless the *CardFlush* instruction is used. For example, with a scan count of 600, a scan interval of 100 ms, and writing a file with *TableFile()* after every 10 seconds (every 100 records), only 400–575 records may be written to the SC115 due to the program ending at the 600th scan prior to transferring all the records to the SC115. To transfer the remaining data to the SC115, use the *CardFlush* instruction placed between the *NextScan* and *EndProg* instructions. See Section 8.5.4 *CardFlush* Example.

8.3 CardOut Instruction

A datalogger that supports CompactFlash® (CF) can be programmed to simultaneously store data to internal memory, a CF card, and an SC115. The datalogger program needs to include a CardOut() instruction in the DataTable() declaration. By including CardOut(), the datalogger can write data to the SC115 that is present only on the CF card.. See Section 10.4 Slow Data Collection for more information on SC115 and CF card interactions. See Section 8.5.6 CardOut Example.

8.4 Powerup.ini

The SC115 supports the use of a powerup.ini file, which allows the host datalogger to be loaded with a specific program or operating system at datalogger power-up. Consult the Power-up.ini section in the datalogger manual for more information.

8.5 Example Programs

8.5.1 Data Collection Mode Examples

8.5.1.1 Standard Data Collection Mode

In this example, the SC115 is connected to the datalogger to retrieve the data written to the datalogger memory after the last connection. It only retrieves data that were already in datalogger memory when it was connected. To retrieve subsequent data, the SC115 must be disconnected then re-connected to the datalogger.

```
Public PanelTempC, BattVolt

DataTable (Test,1,-1)
  DataInterval (0,60,Min,0)
  TableFile ("USB:"+Status.SerialNumber+"_Filename",8,-1,0,0,Min,0,0)
  Sample (1,PanelTempC,FP2)
  Minimum (1,BattVolt,FP2,0,False)
EndTable

BeginProg
  Scan (10,Sec,3,0)
  PanelTemp (PanelTempC,250)
  Battery (BattVolt)
  CallTable Test
  NextScan
EndProg
```

8.5.1.2 Enhanced Data Collection Mode Example

'In this example, the SC115 is connected to the datalogger to retrieve all of the data stored in datalogger memory. It only retrieves data that were already stored in the datalogger memory when it was connected. To retrieve subsequent data, the SC115 must be disconnected and then reconnected to the datalogger.'

```
Public PanelTempC, BattVolt

DataTable (Test,1,-1)
    DataInterval (0,60,Min,0)
    TableFile ("USB:"+Status.SerialNumber+"_Filename",8,-1,0,-1,Min,0,0)
    Sample (1,PanelTempC,FP2)
    Minimum (1,BattVolt,FP2,0,False)

EndTable

BeginProg
    Scan (10,Sec,3,0)
        PanelTemp (PanelTempC,250)
        Battery (BattVolt)
        CallTable Test
    NextScan
EndProg
```

8.5.2 Resident Interval Mode Example

'In this example, the SC115 remains at the datalogger as resident memory. This program avoids data corruption by setting TableFile() Interval to 60. In this case, data are written only once an hour to the SC115.'

```
Public PanelTempC, BattVolt

DataTable (Test,1,-1)
    DataInterval (0,1,Min,0)
    TableFile ("USB:Test",8,-1,0,60,Min,0,0)   'SC115 resident mode with interval
    Sample (1,PanelTempC,FP2)
    Minimum (1,BattVolt,FP2,0,False)

EndTable

BeginProg
    Scan (10,Sec,3,0)
        PanelTemp (PanelTempC,250)
        Battery (BattVolt)
        CallTable Test
    NextScan
EndProg
```

8.5.3 Resident Record Number Mode Examples

*'In this example, the SC115 remains at the datalogger as resident memory.
'This program avoids data corruption by setting TableFile() NumRecs to 60
'In this case, data are written only once an hour to the SC115.*

```
Public PanelTempC, BattVolt
```

```
DataTable (Test,1,-1)
```

```
  DataInterval (0,1,Min,0)
```

```
  TableFile ("USB:Test",8,-1,60,0,Min,0,0)  'SC115 resident mode with NumRec
```

```
  Sample (1,PanelTempC,FP2)
```

```
  Minimum (1,BattVolt,FP2,0,False)
```

```
EndTable
```

```
BeginProg
```

```
  Scan (10,Sec,3,0)
```

```
    PanelTemp (PanelTempC,250)
```

```
    Battery (BattVolt)
```

```
    CallTable Test
```

```
  NextScan
```

```
EndProg
```

*'In this example, the SC115 remains at the datalogger as resident memory.
'This program avoids data corruption by setting DataTable() Size to equal
'TableFile() NumRecs. In this case, data are written only once every two hours
'to the SC115.*

```
Public PanelTempC, BattVolt
```

```
DataTable (Test,1,120)
```

```
  DataInterval (0,1,Min,0)
```

```
  TableFile ("USB:Test",8,-1,120,0,Min,0,0)  'SC115 resident mode with NumRec
```

```
  Sample (1,PanelTempC,FP2)
```

```
  Minimum (1,BattVolt,FP2,0,False)
```

```
EndTable
```

```
BeginProg
```

```
  Scan (10,Sec,3,0)
```

```
    PanelTemp (PanelTempC,250)
```

```
    Battery (BattVolt)
```

```
    CallTable Test
```

```
  NextScan
```

```
EndProg
```

8.5.4 CardFlush Example

```

'In this example, the SC115 remains at the datalogger as resident memory.
'The Scan count, usually left at 0, is set to 600 in this application.
'To ensure complete collection of data, CardFlush instruction is included.

Public PanelTempC, BattVolt

DataTable (Test,1,-1)
  DataInterval (0,10,Sec,0)
  TableFile ("USB:Test",8,-1,100,0,Hr,0,0)   'SC115 resident mode with NumRec
  Sample (1,PanelTempC,FP2)
  Minimum (1,BattVolt,FP2,0,False)
EndTable

BeginProg
  'Scan stops at count 600 – Not enough time to transfer all data!
  Scan (100,mSec,3,600)
    PanelTemp (PanelTempC,250)
    Battery (BattVolt)
    CallTable Test
  NextScan
  CardFlush                               'Included to ensure complete data transfer to SC115
EndProg

```

8.5.5 CardOut Example

```

'In this example, the SC115 and a CompactFlash® card are used as external memory.

Public PanelTempC, BattVolt

DataTable (Test,1,-1)
  DataInterval (0,1,Min,0)
  CardOut (0,-1)
  TableFile ("USB:Test",8,-1,0,60,Min,0,0)   'SC115 resident mode with interval
  Sample (1,PanelTempC,FP2)
  Minimum (1,BattVolt,FP2,0,False)
EndTable

BeginProg
  Scan (10,Sec,3,0)
    PanelTemp (PanelTempC,250)
    Battery (BattVolt)
    CallTable Test
  NextScan
EndProg

```

9. Serial Communications Interface Function

The SC115 uses the Campbell Scientific SDC protocol to facilitate communications between the datalogger and a PC. The SDC address can be set to 7, 8, 10, or 11 (See Section 5.1 SC115 Settings).

Consider the following points when using the SC115 as a communications interface:

- Use the provided extension cables to establish the physical connections between datalogger, SC115, and PC **BEFORE** opening LoggerNet.
- When selecting a port in the LoggerNet Network Map for USB to CS I/O communication, the COM port *SC115 (COM#)* must be selected. *SC115 MSD (COM#)* may also appear as a choice, but must not be selected as it does not support the communications interface mode.
- Always click on *Disconnect* within the datalogger support software Connect screen prior to breaking the physical connection between the PC, SC115 and datalogger. If this rule is not observed, and communication problems develop, follow the procedure in Section 11.1 Breaking the Physical Comms Link.
- Do not connect two SC115s to a PC at the same time if both are attached to powered dataloggers. This can cause confusion of virtual com ports on the PC. As memory devices, two SC115s may be attached to a PC at the same time.

10. Troubleshooting Memory Drive

10.1 Preventing Data Corruption

To avoid corruption of SC115 memory, never remove the SC115 from the datalogger during data transfer. The flashing LED indicates when data are being written to the SC115. Should the SC115 be removed while the LED is flashing, the most likely result is that the current data file will be corrupted. In addition, a FAT sector or the directory link sector may become corrupted, requiring the SC115 memory to be reformatted and all data lost.

When either the *NumRecs/TimeIntoInterval* or *Interval* parameters are set to a non-zero positive value (Resident Modes), there is a risk that the datalogger will begin writing data to the SC115 at the exact moment it is being removed from the datalogger, resulting in data corruption. To prevent this, set these parameters to values that allow the time between writing data to be easily discerned (such as an interval of 10 seconds) by watching the LED. The user then times the removal of the SC115 to occur when the datalogger is not writing data. See Section 8.5.2 Resident Interval Mode Example.

Another method is to set the *NumRecs/TimeIntoInterval* parameter in *TableFile()* to match the *Size* parameter in the associated *DataTable()* instruction. Depending on the rate at which records are written, this can cause data to be written to the SC115 at long intervals, greatly reducing the chance of removing the SC115 from the datalogger while transferring data. See Section 8.5.3 Resident Record Number Mode Example.

If SC115 data becomes corrupted, first attempt to retrieve all files from the SC115. Reformat the SC115 per Section 12.2 Formatting Memory.

10.2 Skipped Scans

To avoid skipped scans, ensure that the scan interval in the datalogger program is long enough to include writing to the SC115. For example, if the program has a single TableFile() command, add 100 ms to the scan rate to accommodate each added TableFile() command.

Compile datalogger programs in pipeline mode when possible. Datalogger programs compiled in sequential mode require a longer scan interval than programs compiled in pipeline mode to avoid skipped scans. In pipeline mode, the *Scan/NextScan* instruction buffer option can be increased to prevent skipped scans. In sequential mode, the TableFile() instruction must finish before continuing to the next instruction, resulting in skipped scans unless the scan interval is long enough to handle all communication, measurement processing, and TableFile() tasks. Sequential mode ignores any scan buffers that may be assigned. Programs may run as much as three times faster in pipeline mode than in sequential mode.

10.3 SC115 with Large Data Compliment

An SC115 with a large compliment of data (either a large number of files, or a few very large files) may respond slower than an SC115 with less data. Compiling datalogger programs from a full SC115 may take longer than normal.

Opening the SC115 directory in support software File Control also takes longer if there is a large number of files within the directory. File control can access approximately 20 files per second when opening the directory.

10.4 Slow Data Collection

When a CompactFlash card (CF) is resident to the datalogger, data are transferred automatically from the CF to a SC115 used in data collection mode (plug and pull). The speed of this data transfer can be very slow. A resident SC115 will avoid the slow transfer. CF cards can also be swapped.

10.5 Slow PC Boot-up

Leaving an SC115 attached to a PC while it is booting up may cause a delay of several minutes to the boot-up process on some computers. The PC will continue to boot normally if the SC115 is removed. The SC115 can then be inserted and used as normal.

10.6 Write Failure

A write failure (after retries) is interpreted to mean the memory is full. When a write failure occurs, *Maximum Number of Files* parameter in the TableFile() instruction is adjusted to its current number. In addition, the datalogger enters ring mode, overwriting the oldest file on the SC115 with the new data (as if the *MaxFile* parameter was set to -1).

The timeout period between a failed write command and a retry is 8 seconds.

11. Troubleshooting Communications Interface

11.1 Breaking the Physical Comms Link

To avoid a communications error, always click on *Disconnect* prior to physically breaking the SC115 interface link.

When the SC115 interface link between the PC and the datalogger is physically broken before *Disconnect* is clicked in the support software, the software requires about one minute to process the broken link. If the link is reconnected before the break is processed, a communications error may occur. If an error occurs, use the following procedure to re-establish communications.

1. Click on the support software Cancel button to stop the connection attempt.
2. Disconnect the SC115 from the PC for at least one minute.
3. After one minute, reconnect the SC115 to the PC and datalogger.
4. Close the support software, then reopen it, then click on *Connect*.

12. Technical Reference

12.1 SDC Addressing

SDC addressing suitable for most applications is set at the factory.

The SC115 responds to two SDC addresses. SDC Address 0 is used when data files are transferred from the datalogger to the SC115. This setting cannot be changed.

When plugged into a USB port, addresses SDC7, SDC8, SDC10, and SDC11 are available to support communication interface mode on the CS I/O port. This setting defaults to SDC7, but can be changed using DevConfig.

12.2 Formatting Memory

The SC115 is formatted at the factory and is ready to use out of the box. Should the SC115 require formatting, such as if the memory becomes corrupted, it can be formatted through datalogger File Control (PC400, LoggerNet, RTDAQ software), datalogger keypad (File | Format | USB), or Windows® Explorer (right click | Format). When given an option between formatting as FAT or FAT32, always choose FAT32.

NOTE

Formatting will erase all data files on the SC115.

12.3 Operating System

The SC115 is configured at the factory with an operating system. Unless notified by Campbell Scientific, the operating system does not need to be reloaded. Downloading a new OS to the SC115 does not affect files in memory.

13. Glossary

DevConfig – Device Configuration Utility. Campbell Scientific PC software used for configuring dataloggers and peripherals.

LED – Light Emitting Diode

LoggerNet – Campbell Scientific top-level datalogger support software.

PC400 – Campbell Scientific mid-level datalogger support software.

RTDAQ – Campbell Scientific datalogger support software for industrial applications.

SDC – Synchronous Devices Communication. A proprietary communications protocol between Campbell Scientific dataloggers and some peripherals.

Campbell Scientific Companies

Campbell Scientific, Inc. (CSI)

815 West 1800 North
Logan, Utah 84321
UNITED STATES
www.campbellsci.com • info@campbellsci.com

Campbell Scientific Africa Pty. Ltd. (CSAf)

PO Box 2450
Somerset West 7129
SOUTH AFRICA
www.csafrica.co.za • cleroux@csafrica.co.za

Campbell Scientific Australia Pty. Ltd. (CSA)

PO Box 444
Thuringowa Central
QLD 4812 AUSTRALIA
www.campbellsci.com.au • info@campbellsci.com.au

Campbell Scientific do Brazil Ltda. (CSB)

Rua Luisa Crapsi Orsi, 15 Butantã
CEP: 005543-000 São Paulo SP BRAZIL
www.campbellsci.com.br • suporte@campbellsci.com.br

Campbell Scientific Canada Corp. (CSC)

11564 - 149th Street NW
Edmonton, Alberta T5M 1W7
CANADA
www.campbellsci.ca • dataloggers@campbellsci.ca

Campbell Scientific Centro Caribe S.A. (CSCC)

300 N Cementerio, Edificio Breller
Santo Domingo, Heredia 40305
COSTA RICA
www.campbellsci.cc • info@campbellsci.cc

Campbell Scientific Ltd. (CSL)

Campbell Park
80 Hathern Road
Shepshed, Loughborough LE12 9GX
UNITED KINGDOM
www.campbellsci.co.uk • sales@campbellsci.co.uk

Campbell Scientific Ltd. (France)

3 Avenue de la Division Leclerc
92160 ANTONY
FRANCE
www.campbellsci.fr • info@campbellsci.fr

Campbell Scientific Spain, S. L.

Avda. Pompeu Fabra 7-9, local 1
08024 Barcelona
SPAIN
www.campbellsci.es • info@campbellsci.es

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