PakBus Networking Guide

5.11.08

Contents

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

1. Introduction 1

2. PakBus Network Terminology 1

3. PakBus Concepts 3

3.1 Protocol Description 3
3.2 Device Addresses 4
3.3 Routers 5
3.4 Discovering Devices 6
3.4.1 Beacons 6
3.4.2 Allowed Neighbour List 6
3.4.3 Program Instructions 7
3.5 Datalogger to Datalogger Communication 7
3.6 Concurrent Communication 7
3.7 Keeping Track of PakBus Devices 7

4. PakBus Dataloggers 8

4.1 CRBasic Dataloggers (CR800, CR850, CR1000, and CR3000) 9
4.2 CR200 Series Datalogger 10
4.3 Edlog Dataloggers (CR510-PB, CR10X-PB and CR23X-PB) 11
4.3.1 PakBus Settings 11

5. PakBus Troubleshooting Tools 13

5.1 PakBus Graph 13
5.2 LogView 13

Appendix

A. Entering CR10X-PB PakBus Settings with a Keyboard Display A-1

Figures

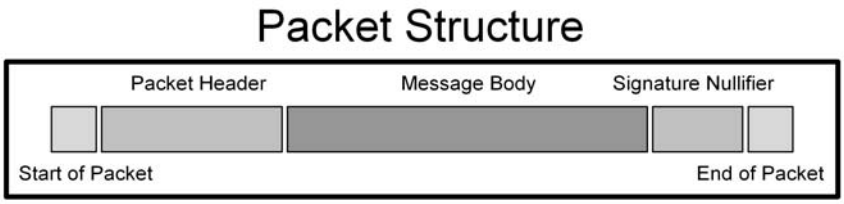
3-1. Example PakBus Protocol Packet 4
3-2. Network Map 5

PakBus Networking Guide

1. Introduction

PakBus® describes a proprietary family of protocols created by Campbell Scientific, Inc. for communication between connected devices. Similar in many ways to TCP/IP, PakBus is a packet-switched network protocol with routing capabilities.

These data communication packets contain a header, a message body, and an error-checking segment. The header of the packet describes where the packet came from, where it is going and what type of information can be expected in the message body. The message body contains the data being sent from one device to another. In addition, the packet is checked for errors to ensure that it has been transmitted without corruption.



PakBus protocol uses the basic elements of packet-switched communication to form packets when communicating between devices on a network. Some advantages of using PakBus communication between devices include the following:

- Network communication path auto-discovery capabilities
- Simple datalogger to datalogger communication
- Concurrent communication with devices

Additional information about these specific advantages along with a more detailed description of the PakBus protocol will be presented in this document.

2. PakBus Network Terminology

When discussing PakBus networks there are several common terms that are used. Learning the following terms will help increase understanding of PakBus protocol. These terms will be discussed and used in greater detail later in this document.

Allowed Neighbour List: An allowed neighbour list sets up a list of neighbours with which a device will communicate. If a device address is entered in the allowed neighbour list, a hello exchange will be initiated with that device. Any device with an address between 1 and 3999 that is not entered in the allowed neighbour list will be filtered from communicating with the device using the list.

Beacon: A broadcast message sent over the network by a node. All PakBus devices must respond to a beacon by initiating a hello exchange unless an allowed neighbour list prohibits communication with that node.

Beacon Interval: The interval specifying the rate at which beacons will be issued to the network by a PakBus node.

Destination Address: The PakBus address of the node where a packet is being sent.

Hello Exchange: The process of verifying a node as a neighbour. A hello command is sent to a neighbour node and the exchange is complete once a hello response comes back from the neighbour. To successfully send and receive packets, network devices must maintain an accurate list of currently viable links and devices using hello exchanges. If a device has not heard from a certain neighbour within a period of time greater than the calculated verification interval for those nodes, a hello exchange will be initiated to determine if the node is still a viable neighbour.

Hop: The link between two adjacent nodes.

Hop Metric: An estimate of the time required to complete a transaction between two adjacent nodes.

Leaf Node: A leaf node is a PakBus device that exists at the end of a network segment or branch. The leaf node can't route packets to other devices. It can only send and receive packets meant for that device.

Neighbour: A PakBus node that is directly connected to or, in other words, only takes one hop to communicate with another node. Before a PakBus node considers another node to be a neighbour, the nodes must successfully accomplish a hello exchange and periodically communicate in some manner. After a successful hello exchange, the nodes establish each other as neighbours.

Network: A group of two or more PakBus devices that can communicate with each other.

Node: In PakBus networking, a node is equivalent to a PakBus device and is used to reference an element in the PakBus network. A node can be a computer, datalogger, or communication peripheral that has its own PakBus address.

PakBus Address: The identification number or address of the node. The PakBus address is used to declare a source and destination node in communication packets. Every PakBus device in a network needs a unique PakBus address to receive, send, or route PakBus protocol packets.

PakBus Device: Any node that communicates using PakBus protocol.

Router: A PakBus device that forwards requests between devices that cannot communicate directly with each other. Routers maintain a routing table, which is a list of known nodes and routes. A router will only accept and forward packets that are destined for known devices. Routers pass their lists of known neighbours to other routers in order to build the network routing system.

Routing Table: A list of the most direct routes between nodes on the network. Routers calculate the most direct path between nodes based on the hop metrics of the available routes.

Source Address: The PakBus address of the node sending the packet.

Verify Interval (Communication Verification Interval or CVI): The interval at which communication must occur between established neighbours to maintain the list of known neighbour routes. If no communication has taken place during the specified interval a hello exchange is initiated. A verify interval of zero causes the device to use a default verify interval of 2.5 times the beacon interval. If the beacon interval is also zero, the default verify interval is five minutes.

3. PakBus Concepts

PakBus dataloggers rely on a proprietary packet-switched communication protocol referred to as PakBus. Formalized packets are used for communication.

3.1 Protocol Description

Like other types of packet switched communication protocols such as TCP/IP, PakBus protocol packets contain a header, a message body, and an error checking segment.

The header contains specific information that facilitates the transfer of packets between devices. Along with other details, the header of a PakBus protocol packet includes the address of the device that initiated the packet (source address), the address of the device where the packet is being sent (destination address), and a declaration of the type of packet being sent. PakBus devices on the network look at the source and destination addresses in the header to determine if a packet is meant for them and to determine where a reply message should be sent. The packet type declaration in the header describes the data contained within the body of the packet.

The body or message portion of the packet contains structured information based on the packet type declared in the header. The receiving device uses the packet type declaration to read the message data.

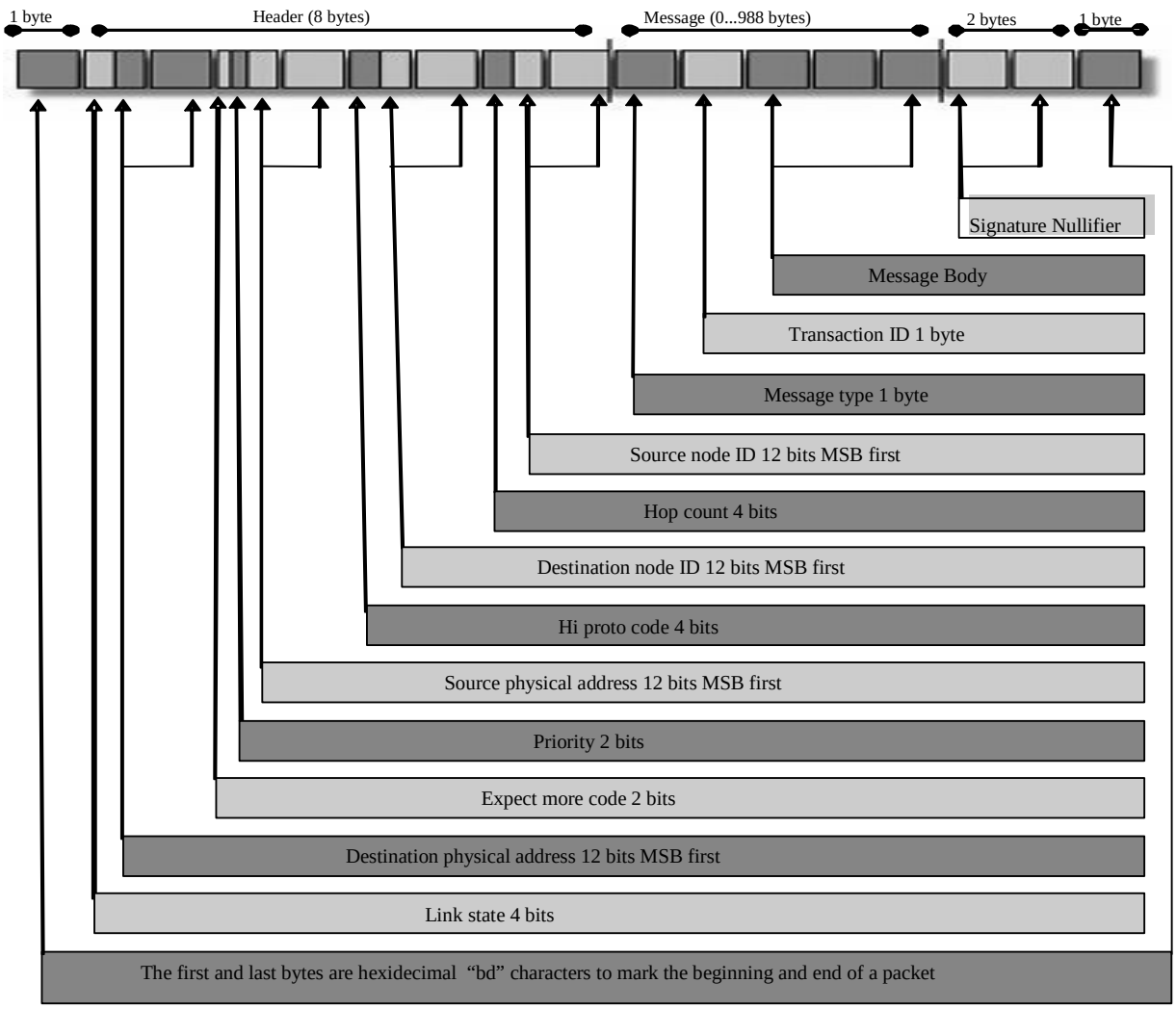


Figure 3-1. Example PakBus Protocol Packet

Each packet also contains an error checking signature nullifier used to acknowledge the validity of the packet.

Once the packet has been validated, the device receiving the packet reads the message based on the structure described by the packet type declaration and, if necessary, returns an appropriate packet to the sending device.

3.2 Device Addresses

Since each PakBus protocol packet contains a source address and a destination address, all PakBus devices on a network must have a unique PakBus address. PakBus addresses within 1 and 4094 are valid. However, by convention, addresses 1 through 3999 are typically used for dataloggers and other hardware devices on the network while PakBus addresses 4000 through 4094 are typically used for software such as LoggerNet, PC400 or PC200W.

Although addresses 1-4094 can technically be used for any PakBus capable device or software product on the network, an address filtering method called the allowed neighbour list can be set in a datalogger to limit communication between devices. Allowed neighbour lists specify neighbours to which a datalogger will respond and cause it to ignore all other devices. However, packets sent from a neighbour with a PakBus address greater than or equal to 4000 can not be filtered or ignored

even if an allowed neighbour list has been declared in a datalogger. Therefore, LoggerNet or similar software expects all nodes to respond to communication packets since the software has a default PakBus address of 4000 or greater.

The following table lists the default PakBus address assigned to specific Campbell Scientific software products:

Software	Default PakBus Address
LoggerNet	4094
PC400	4093
RTDAQ	4090
PC200W	4092
PConnect/PConnect CE	4091
Device Configuration Utility	4089

In order to communicate with devices, Campbell Scientific software such as LoggerNet must first know the network structure. The hardware and devices used to facilitate communication such as the COM port or phone modem along with their required settings are entered in the software to create a network map. Once a network map exists, the software can attempt to communicate with devices in the network.

LoggerNet uses a virtual interface called a PakBus Port when communicating with a PakBus datalogger. The PakBus Port contains the PakBus address for the software. The other PakBus devices in the PakBus network are entered as children of the PakBus port in the software network map.

Figure 3-2. Network Map

By default, all PakBus devices placed under a single PakBus port in the network map are considered part of the same network. Therefore, each device in this network branch must have a unique PakBus address. Devices that can discover each other using the physical network regardless of their location in the software network map will create a conflict unless unique PakBus address are used.

3.3 Routers

A router is a PakBus device that can accept a packet and forward it toward its ultimate network destination. Routing is fundamentally driven by the router's ability to learn about neighbours and the sharing of that information with other routers in the network. When a change occurs in a router on the network, that router will exchange the new neighbour information with other known routers.

Routers ultimately determine the best complete route to a destination by using the known routes information they maintain from all other routers. This information gives routers knowledge of every known link in the network and the ability to calculate the best complete route based on the hop metric of each link.

3.4 Discovering Devices

PakBus devices on a network may initially be unaware of each other. One benefit of using a PakBus network is that PakBus devices can automatically discover other PakBus devices on the network. There are three ways that devices can initiate a hello exchange and discover each other as neighbours:

1. Using a beacon
2. Using an allowed neighbour list
3. Using instructions in the datalogger program

3.4.1 Beacons

Beacons are a simple way for devices to discover neighbours. When devices are configured to beacon, a beacon broadcast packet is sent over the network at a regular specified interval. A PakBus device on the network that receives the beacon will respond with a hello message to the device that sent the beacon. The hello message includes the device's PakBus address. When the device that sent the beacon receives a hello message, it will answer with a hello response packet and complete the hello exchange process.

Beacons will use network resources since they broadcast over the network each time the specified beacon interval occurs. If using a metered connection or a network that needs limited traffic, carefully consider the consequences before implementing beacons as a method of discovering neighbours.

3.4.2 Allowed Neighbour List

Allowed neighbour lists can be used as another method of introducing devices to each other. Entering the address of a device in the allowed neighbour list of a datalogger will initiate discovery of the device.

Allowed neighbour lists also act as communication filters. A datalogger with a declared allowed neighbour list will only communicate with node that has an address of 4000 or greater or that has an address in the list. The receiving device will ignore communication packets from other nodes.

Allowed neighbour lists can be used to declare specific communication paths between nodes rather than having the devices discover alternate undesirable paths and possibly using marginal routes of communication. The PakBus address and the communication interface of the neighbour device in the allowed neighbour list must be entered correctly for discovery to work.

If communication fails to a node declared in an allowed neighbour list, the datalogger will continually attempt communication to that node until it is discovered with a successful hello exchange. The discovery attempt for devices in the allowed neighbour list is done with a directed hello command that occurs at two times the verify interval. If the verify interval is zero, the interval used to verify communication with devices in the allowed neighbour list will be 30 seconds by default. If the beacon interval is non-zero, discovery attempts for devices in the allowed neighbour list use the standard beacon interval and message rather than a directed hello message.

These specific intervals and procedures only apply to discovery of devices in the allowed neighbour list. Please do not confuse the discovery intervals and procedures for nodes entered in the allowed neighbour list with standard discovery

intervals and procedures used when creating the list of known routes to PakBus neighbours, which will be discussed later in this document.

If routers on the network are using allowed neighbour lists to filter communication, they must list each other as potential neighbours in their respective lists or they will not be able to communicate with each other. Do not include the datalogger's own address in the allowed neighbour list.

3.4.3 Program Instructions

A PakBus datalogger can be programmed with instructions that send packets to other PakBus devices on the network. When a program sends a packet to a specific PakBus device, the devices discover each other and become neighbours.

Although datalogger program instructions are a direct way to discover neighbours, they are not very dynamic. The program must be changed and sent to the datalogger before a new neighbour or communication route can be discovered.

3.5 Datalogger to Datalogger Communication

One distinct advantage of using PakBus devices is their ability to easily communicate directly with each other. A datalogger program can contain an instruction that will send information to a specific PakBus address on the network. The data packet will be routed through the network to the device with the address specified in the datalogger program instruction. Communication and information exchange between PakBus devices using these program instructions can happen quickly and unattended.

Some of the instructions used to transfer information between devices include GetVariables, SendGetVariables, and SendVariables. An example of datalogger to datalogger communication is a datalogger in a remote location monitoring water levels that is programmed to send a variable to a second datalogger that controls a gate or pump. Another example is a central datalogger that collects data from many remote devices and stores that data for easy retrieval by a computer from a single datalogger.

3.6 Concurrent Communication

The packets in PakBus networking are small and the communication links are only active for a short period of time while the packet is being transmitted. Longer messages can be broken into several smaller packets and sent individually between devices. Since these packets do not require a dedicated link, other related or unrelated packets can be sent over the same link at approximately the same time.

In addition, individual packets contain the necessary information to facilitate communication. Therefore, PakBus devices can simultaneously send and receive packets to and from different nodes on the network. For example, two researchers in different locations can collect data from a single datalogger simultaneously over the same PakBus connection (if permitted by the physical connection itself).

3.7 Keeping Track of PakBus Devices

Before devices acknowledge each other as neighbours, they must successfully complete a hello exchange and periodically communicate with each other. For communication between the devices to continue, they must keep an accurate list of all known neighbour routes. A healthy PakBus network continually updates the neighbour routes information between nodes. Once devices establish each other

as neighbours with a hello exchange, they expect to hear from each other within a calculated verify interval.

A verify interval can be entered in the datalogger for a specific link. If the verify interval is set to zero, the verify interval becomes 2.5 times the beacon interval. However, if both the beacon interval and verify interval are set to zero, the default verify interval for a device becomes 300 seconds.

During the hello exchange, neighbours pass each other their verify interval information. A calculated verify interval is negotiated between the neighbours from this information. The negotiated interval is the lesser of either the neighbour node's verify interval or 6/5 of the first node's verify interval. The calculated verify interval is used to determine how often neighbours must communicate with each other to keep the link active.

A good practice is to set the verify interval to a value longer than the shortest expected interval of communication inactivity. With that setting, regular communication between devices is sufficient to preserve their status as neighbour devices without causing the verify interval to expire.

If a node fails to hear from a neighbour and the calculated verify interval expires, an attempt will be made to re-establish the connection. The device will issue up to four hello messages, 2 to 12 seconds apart, directed at the neighbour node in an attempt to make contact. If communication is still unsuccessful after four hello message retries, the neighbour is removed from the list of know routes in the device. The neighbour will only be added to the list of known routes in the device after another successful hello exchange.

Devices set as routers must keep accurate lists of not only neighbours but also other routers in the network. Routers use the same verification process as neighbours when adding and removing other routers from their list. In addition, when the list of known routes in a router on the network changes by having a neighbour added or removed, the router shares that information with other routers on the network at a random interval.

When devices establish each other as neighbours, they must also know the port or interface on the device they are using for communication. This port information is included in the list of known neighbour routes. Examples of ports that might be used for communication include RS-232, modem enable (ME), synchronous device communications (SDC), or concurrent synchronous device communication (CSDC).

4. PakBus Dataloggers

Datalogger with PakBus operating systems:

- Use packet-based communication
- Create specific final storage tables to logically separate different types of data collection such as hourly, daily, rainfall, etc.
- Allow access and sharing of specific data from a table

PakBus messages are encapsulated in packets of specific size and structure. Packets include elements such as the source and destination address for the packet, markers declaring the type of message contained within the packet, transaction numbers, the message or data itself, and error checking signature nullifiers. When a packet is received, it is analyzed for errors using the signature nullifier. If it fails the error checking process, the corrupt packet is rejected and must be resent. Once

the packet is accepted, the information within the packet is processed and a response is generated if necessary. While more complex, this packet structure ensures that software such as LoggerNet can send and receive accurate commands and messages.

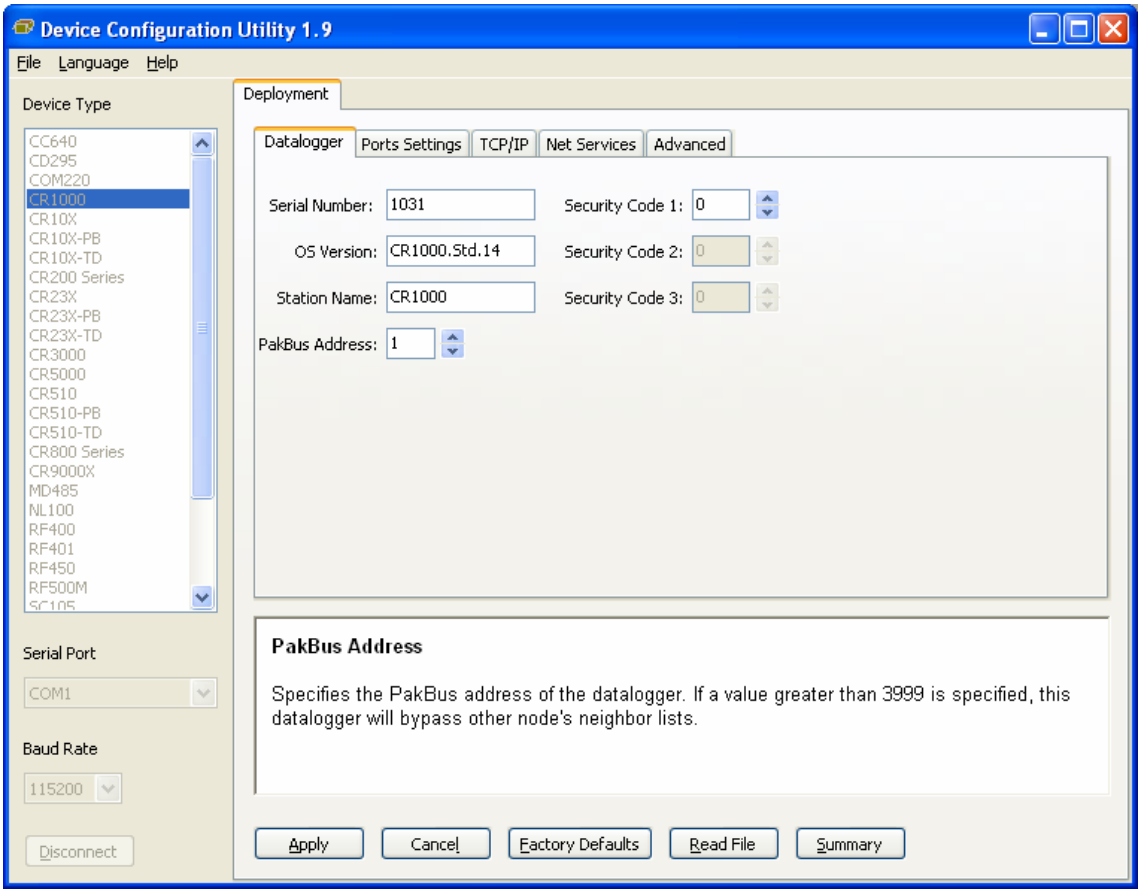
PakBus dataloggers allocate separate memory space for each final storage table when the program is compiled. The datalogger knows what output tables are required and how many fields are stored in each record at a given interval. The datalogger partitions memory so that only the data specified in the datalogger program are stored in the corresponding table. Each record within a table will contain the same number of filed and along with the values in each record, a record number and time stamp.

This table structure allows the user flexibility in terms of how much data to store in each distinct table. The simplest scheme is to allow the datalogger to automatically allocate table size so that all tables fill up and start overwriting the oldest records with the newest records at the about the same time. Alternatively, the user can create tables that contain a specified number of records. The table structure is declared in the datalogger program.

Since data is stored a specific table with a known structure, record number, and time stamp, the datalogger can access this data in its own table and send it to another node or retrieve the data from the table of a different node.

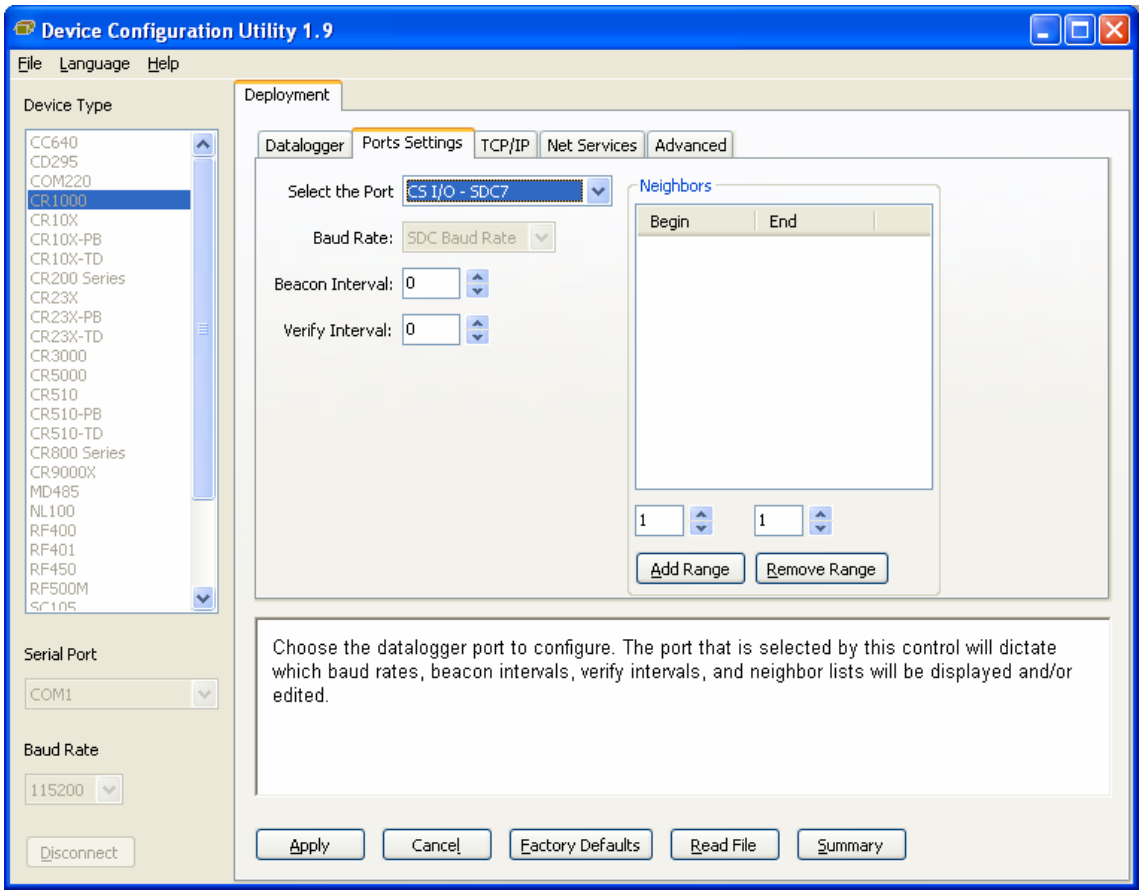
4.1 CRBasic Dataloggers (CR800, CR850, CR1000, and CR3000)

PakBus settings for these dataloggers are entered with the Device Configuration Utility, which is used to administer configurable Campbell Scientific devices.



Once connected to the datalogger with the Device Configuration Utility, the PakBus address of the datalogger can be set. Keep in mind that each PakBus device in the network must have a unique address.

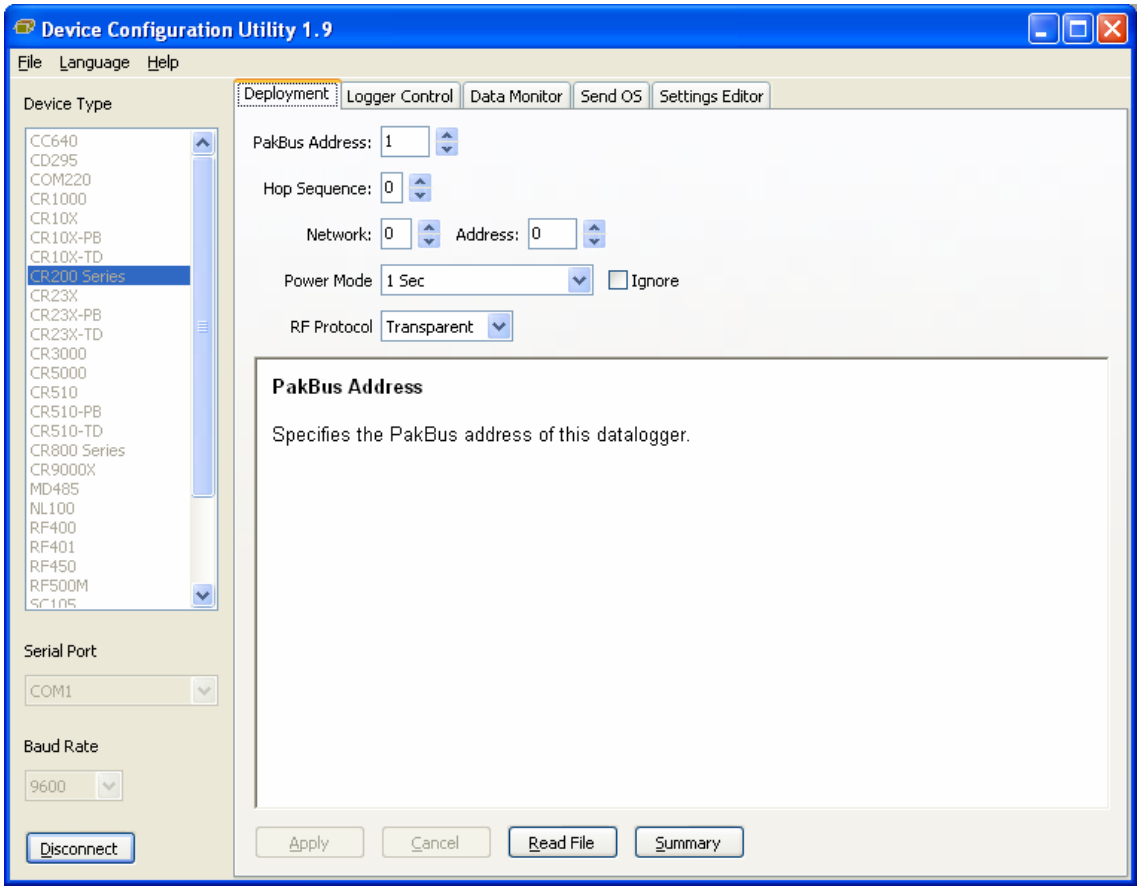
Other PakBus related settings are activated under the Port Settings tab. Select the appropriate communication port and enter the baud rate, beacon interval and verify interval. To create an allowed neighbour list, choose a range of addresses that this device will consider a neighbour on this port and click on the Add Range button. If the allowed neighbour list is activated, the device will ignore other devices in the network that have an address less than 4000 and are outside the range specified in the list.



4.2 CR200 Series Datalogger

The CR200 Series dataloggers are programmed using the CRBasic editor. However, unlike other CRBasic dataloggers, the CR200 Series has limited memory and requires the datalogger program to be compiled before it is sent to the datalogger. Additionally, the CR200 Series datalogger can only act as a leaf node in a PakBus network.

The PakBus settings in the CR200 Series datalogger are entered using the Device Configuration Utility.



Once connected to the datalogger with the Device Configuration Utility, the PakBus address of the datalogger can be set on the Deployment tab. Other radio settings can be changed on this screen, but there are no other PakBus settings that can be changed using Device Configuration Utility.

4.3 Edlog Dataloggers (CR510-PB, CR10X-PB and CR23X-PB)

A PakBus operating system is available for the CR510, CR10X and CR23X dataloggers. The operating system can be sent to the datalogger with the Device Configuration Utility.

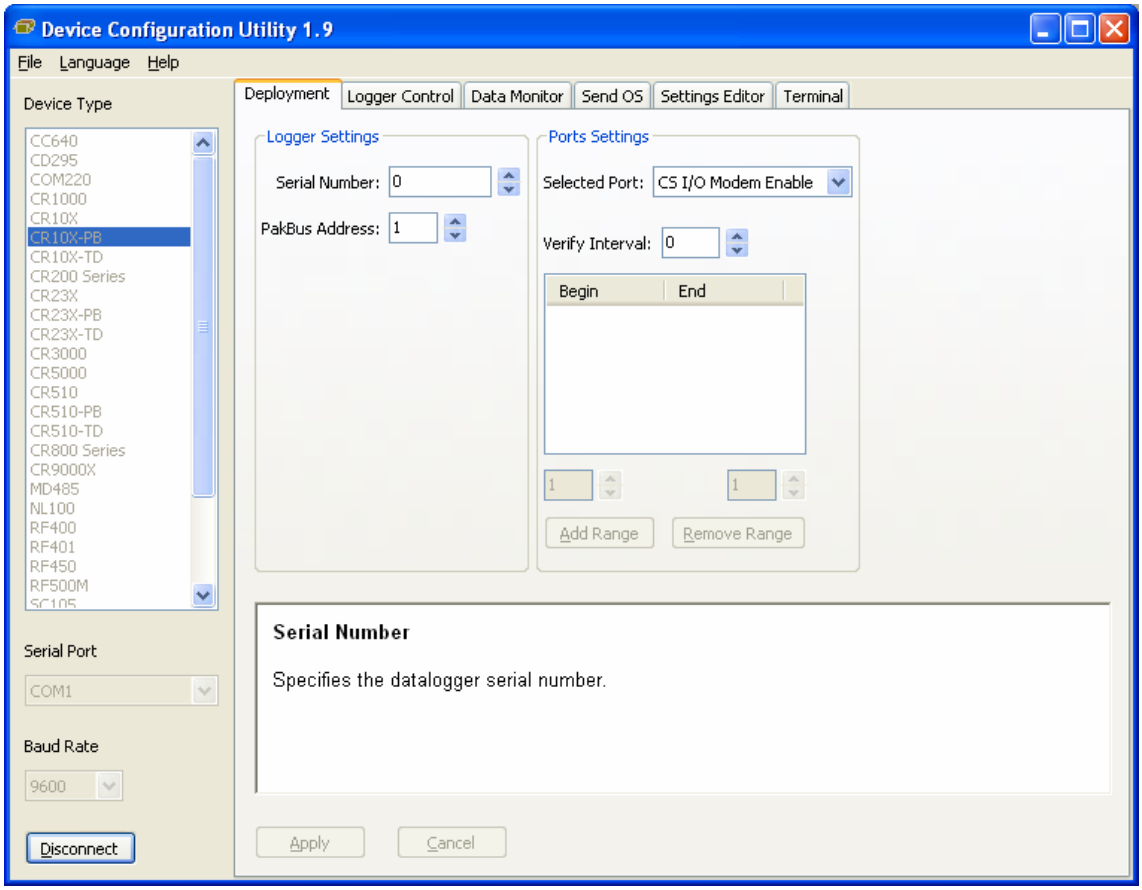
4.3.1 PakBus Settings

PakBus settings can be entered in the datalogger using the Device Configuration Utility. Alternatively, the user can enter PakBus settings in the datalogger program using the Options | PakBus Settings menu item in Edlog. Sending the program to the datalogger will configure those PakBus settings and override any settings already in the datalogger but the preferred method of entering PakBus settings is with the Device Configuration Utility.

In addition, PakBus settings can be manually configured or viewed in the CR510-PB, CR10X-PB or CR23X-PB datalogger with a keyboard display. Use *D15 and *D18 for configuration and *D17 and *D19 to view the routing table of the datalogger.

Since the preferred method of entering PakBus settings is with Device Configuration Utility, the next section describes some of the settings and choices available for these dataloggers.

Open the Device Configuration Utility software and connect to the datalogger. Once a connection is established, the PakBus settings for the datalogger can be adjusted.



Under the Deployment tab of the Device Configuration Utility, enter the PakBus address of the device in the space specified in Device Configuration Utility. Each PakBus device in the network must have a unique address.

Allowed neighbour lists can be activated by selecting the communication port, entering a Verify Interval, and choosing a range of addresses that this device will consider a neighbour. If the allowed neighbour list is activated, the device will ignore other devices in the network that have an address less than 4000 and are outside the range specified in the list.

There are no other PakBus settings to adjust with the Device Configuration Utility for these dataloggers.

5. PakBus Troubleshooting Tools

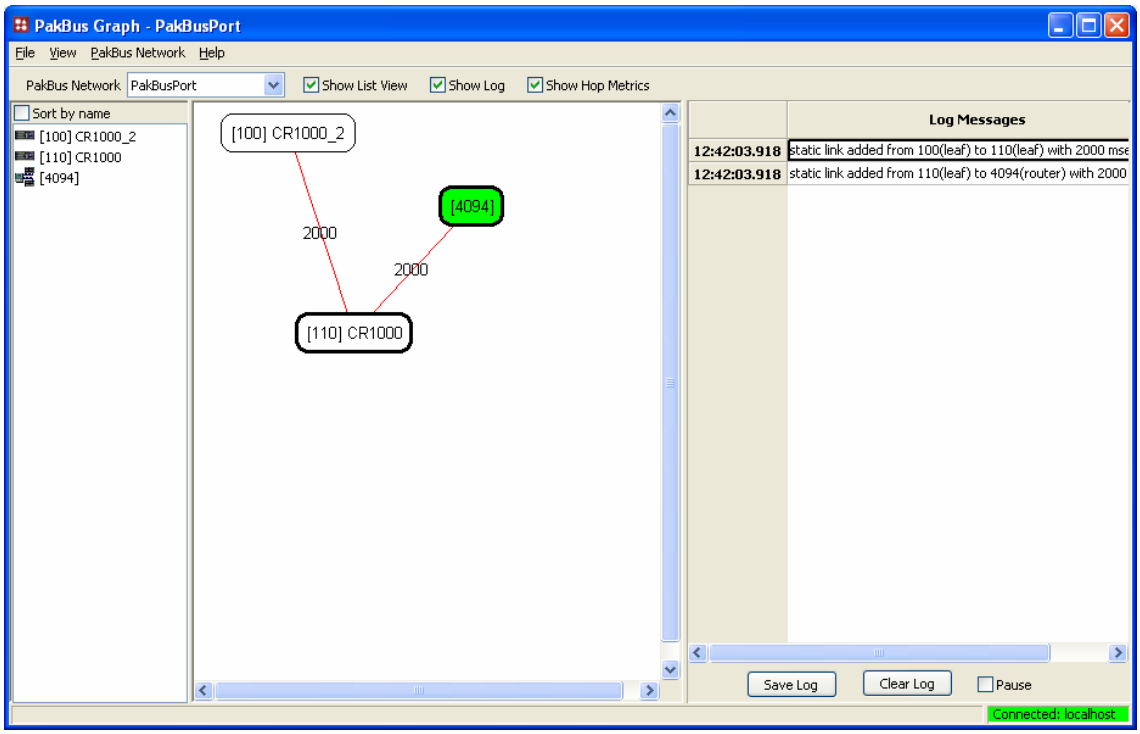
5.1 PakBus Graph

PakBus Graph is a LoggerNet utility that graphically depicts the devices and connection in a PakBus datalogger network. In PakBus graph, the PakBus devices are represented at least by their PakBus address and by their name if the device has been configured in the LoggerNet Setup screen. Some PakBus settings are viewable and editable through PakBus Graph.

Right-clicking on a device in the map will expose available options. Some of those options may include Edit Settings and Ping Node, which can be useful when troubleshooting.

Edit Settings will request the settings from the device. The settings for the device will open in a new window. Settings that are editable can be changed from this window.

Ping Node allows the user to test communication to the device from other nodes on the network. The results are displayed for the user and are useful when troubleshooting communication links.



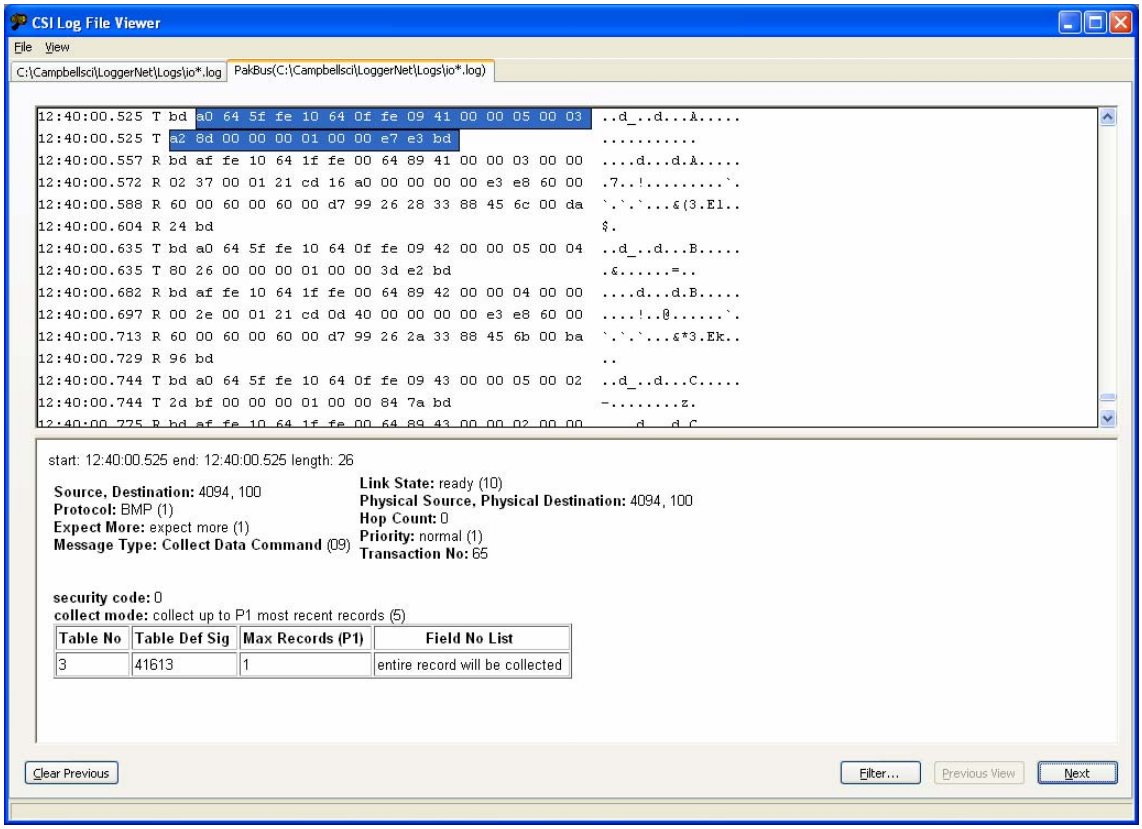
5.2 LogView

LogView is a utility installed in the LoggerNet program directory, which can be found by default in C:\Program Files\Campbellsci\LoggerNet. LogView not only has the capability of displaying log files but also includes a PakBus packet filter that can parse low-level log file packets into human readable information.

Open LogView and add a log file for viewing by choosing File | Open Log. If the log file is a low-level log containing PakBus packet information, a filter can be applied to show the packet information in a format that is easier to understand. Select the open low-level log file and choose View | Add PakBus Filter.

After the filter has been applied, a split screen will show the low-level log and the translation of that information. Scroll through the PakBus packets using the Next and Previous View buttons. The lower screen will show translated information describing PakBus packets that are highlighted in the upper screen.

The PakBus filter is useful to see device addressing, record numbers requested for collection, and other communication attempts.



Appendix A. Entering CR10X-PB PakBus Settings with a Keyboard Display

If using a keyboard display to enter PakBus settings, they can be configured using *D settings anytime after downloading a PakBus operating system to the datalogger and either before or after sending the program to the datalogger. Pressing *0 recompiles the datalogger and saves the *D settings to FLASH memory. Current PakBus settings will persist through a new program send as long as the new program does not contain PakBus settings that were entered in Edlog.

*D15 The PakBus Address and Routing Table Creation

CAUTION

Collect any wanted data before typing *0 to save *D15 edits to FLASH because final storage data is cleared!

*D15 settings allocate memory similar to *A. It is a good idea when configuring them to leave "room to grow." Changing *D15 settings later on could result in loss of data.

Set the PakBus address and routing table information of the datalogger by entering the following settings:

*D15	:nnnn	The PakBus address from 1-4094. Remember that each node on the network must have a unique PakBus address. Avoid PakBus addresses used by support software and be careful with PakBus addresses larger than 3999 if using allowed neighbour lists.
	01 :nnnn	The maximum number of nodes in the network. Enter 0 if the datalogger is a leaf node and make sure to enter a non-zero number if the datalogger is a router.
	02 :nnnn	The maximum neighbours for this node. The maximum must be a non-zero number.
	03 :nnnn	The maximum routers in the network and must be non-zero.
	04 :nnnn	The default router's PakBus address. Used so that packets can be passed efficiently through the network.

If a device is unable to find a route to a destination in its routing table, and it has a default router address configured, it will send the packet out via the default router.

If in doubt about how large to make max nodes, neighbours, or routers in the datalogger configuration, guess a little high. Setting these values too low can be catastrophic if it means required routes are missing from lack of space. The following table shows how datalogger memory usage rises per the square of the number of routers.

*D15 Routing Table Datalogger Memory Usage:

Per node in network	16 bytes
Per neighbour	8 bytes
Per router in network	2 x routers x (routers –1) bytes

For example:

Number of nodes = 4 CR10X-PB memory usage = 64 bytes
Number of neighbours = 3 CR10X-PB memory usage = 24 bytes
Number of routers = 2 CR10X-PB memory usage = 4 bytes

*D17 View Routing Table

*D17	01 :xxxx	PakBus Address of destination node
	01 :yyyy	Packet sent via router with PakBus Address yyyy
	01 :zzzz	The worst case response time between nodes in seconds (Hop Metric)
	02 :xxxx	PakBus Address of destination node
	02 :yyyy	Packet sent via router with PakBus Address yyyy
	02 :zzzz	The worst case response time between nodes in seconds (Hop Metric)
	0n :xxxx	PakBus Address of destination node
	0n :yyyy	Packet sent via router with PakBus Address yyyy
	0n :zzzz	The worst case response time between nodes in seconds (Hop Metric)

If there is no router, only the neighbour is indicated in the *D17 routing table. For example, if the device with PakBus address 3 is a neighbour to a datalogger, the datalogger's routing table will look like this:

01 :0003 PakBus Address of destination node
01 :1.0000 Worst case response time in seconds

The *D17 listing will change once the datalogger discovers a change in the neighbour list that results in a change in the best routes to other nodes in the network. *D17 shows the current best routes. It does not show all routes even though the datalogger may have access to all possible routes to any given node. If a router's neighbour list changes, all routers are notified and will recalculate their best route routing table based on the new information.

Although *D17 shows neighbours, in the case of multiple peripherals it does not show the ports used to access the respective neighbours.

*D18 Set Beacon Interval and Port Protocol

CAUTION Collect any wanted data before typing *0 to save *D18 edits to FLASH because final storage data will be cleared!

*D18	01 :xxxx	Enter the interval in seconds for CSDC 7
	02 :xxxx	Enter the interval in seconds for CSDC 8
	03 :xxxx	Enter the interval in seconds for ME
	04 :xxxx	Enter the interval in seconds for RS-232

A value of "0" turns off beacons.

*D19 Set Allowed Neighbour List

CAUTION Collect any wanted data before typing *0 to save *D19 edits to FLASH because final storage data will be cleared!

NOTE Configuring the *D19 allowed neighbour list requires that the *D15 Routing Table also be configured with max nodes, max neighbours, and max routers so the datalogger has a routing table in which to put neighbours discovered from *D19 allowed neighbours.

*D19	:xx	Port: 17 – CSDC 7 18 – CSDC 8 02 – ME 02 – RS-232 on the CR23X-PB
	:xxxx	Communication Verification Interval in seconds
	01 :xxxx	Starting PakBus Address of allowed neighbours
	01 :ss	Swath of allowed neighbours addresses
	02 :yyyy	Starting PakBus Address of allowed neighbours
	02 :ss	Swath of allowed neighbours addresses
	0n :zzzz	Starting PakBus Address of allowed neighbours
	0n :ss	Swath of allowed neighbours addresses

After compiling the allowed neighbour list settings, the datalogger starts sending directed hello messages. Once the hello exchange completes, the neighbour routes appear in the *D17 Routing Table, enabling the datalogger to communicate with the discovered neighbour.

If there are two neighbour devices using allowed neighbour lists, they must be in each other's list in order for them to discover each other.

Changes to the *D19 window take place immediately after editing and pressing A. Pressing *0 saves the edits to FLASH memory.

To delete a setting, for example, port "17", type 0 over the 17 and press A to enter the new value. Then type the desired port, if any, over the "0" and finish the configuration. Entering "0" or changing the *D19's COM port code resets the dataloggers' neighbour list and routing table and restarts the discovery process.

NOTE If a "17", for example, is typed over an existing "02" without first typing "0" and A to enter, the result will be two ports 02 and 17.

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
sales@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 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)
Miniparc du Verger - Bat. H
1, rue de Terre Neuve - Les Ulis
91967 COURTABOEUF CEDEX
FRANCE
www.campbellsci.fr
info@campbellsci.fr

Campbell Scientific Spain, S. L.
Psg. Font 14, local 8
08013 Barcelona
SPAIN
www.campbellsci.es
info@campbellsci.es

Campbell Scientific Ltd. (Germany)
Fahrenheitstrasse13, D-28359 Bremen
GERMANY
www.campbellsci.de
info@campbellsci.de

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