

# INSTRUCTION MANUAL



## Wireless Sensor Network

Revision: 11/17



Copyright © 2010-2017  
Campbell Scientific, Inc.



# **Limited Warranty**

---

“Products manufactured by CSI are warranted by CSI to be free from defects in materials and workmanship under normal use and service for twelve months from the date of shipment unless otherwise specified in the corresponding product manual. (Product manuals are available for review online at [www.campbellsci.com](http://www.campbellsci.com).) Products not manufactured by CSI, but that are resold by CSI, are warranted only to the limits extended by the original manufacturer. Batteries, fine-wire thermocouples, desiccant, and other consumables have no warranty. CSI’s obligation under this warranty is limited to repairing or replacing (at CSI’s option) defective Products, which shall be the sole and exclusive remedy under this warranty. The Customer assumes all costs of removing, reinstalling, and shipping defective Products to CSI. CSI will return such Products by surface carrier prepaid within the continental United States of America. To all other locations, CSI will return such Products best way CIP (port of entry) per Incoterms ® 2010. This warranty shall not apply to any Products which have been subjected to modification, misuse, neglect, improper service, accidents of nature, or shipping damage. This warranty is in lieu of all other warranties, expressed or implied. The warranty for installation services performed by CSI such as programming to customer specifications, electrical connections to Products manufactured by CSI, and Product specific training, is part of CSI’s product warranty. **CSI EXPRESSLY DISCLAIMS AND EXCLUDES ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CSI hereby disclaims, to the fullest extent allowed by applicable law, any and all warranties and conditions with respect to the Products, whether express, implied or statutory, other than those expressly provided herein.”**

# Assistance

---

Products may not be returned without prior authorization. The following contact information is for US and international customers residing in countries served by Campbell Scientific, Inc. directly. Affiliate companies handle repairs for customers within their territories. Please visit [www.campbellsci.com](http://www.campbellsci.com) to determine which Campbell Scientific company serves your country.

To obtain a Returned Materials Authorization (RMA) number, contact CAMPBELL SCIENTIFIC, INC., phone (435) 227-9000. Please write the issued RMA number clearly on the outside of the shipping container. Campbell Scientific's shipping address is:

**CAMPBELL SCIENTIFIC, INC.**

RMA# \_\_\_\_\_  
815 West 1800 North  
Logan, Utah 84321-1784

For all returns, the customer must fill out a "Statement of Product Cleanliness and Decontamination" form and comply with the requirements specified in it. The form is available from our website at [www.campbellsci.com/repair](http://www.campbellsci.com/repair). A completed form must be either emailed to [repair@campbellsci.com](mailto:repair@campbellsci.com) or faxed to (435) 227-9106. Campbell Scientific is unable to process any returns until we receive this form. If the form is not received within three days of product receipt or is incomplete, the product will be returned to the customer at the customer's expense. Campbell Scientific reserves the right to refuse service on products that were exposed to contaminants that may cause health or safety concerns for our employees.

# 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.com](http://www.campbellsci.com) or by telephoning (435) 227-9000 (USA). 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, 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

---

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

<b>1. Introduction .....</b>	<b>1</b>
<b>2. Precautions .....</b>	<b>1</b>
<b>3. Initial Inspection.....</b>	<b>1</b>
<b>4. Overview .....</b>	<b>2</b>
4.1 CWB100 Wireless Base Station.....	2
4.2 Wireless Sensors .....	2
4.2.1 CWS220 Wireless Infrared Radiometer.....	3
4.2.2 CWS655 Wireless Soil-Water Probe .....	3
4.2.3 CWS900 Configurable Wireless Sensor Interface .....	4
<b>5. Specifications.....</b>	<b>5</b>
5.1 CWB100 .....	5
5.2 CWS220 Specifications .....	6
5.3 CWS655 Specifications .....	6
5.4 CWS900 Specifications .....	7
5.5 Internal Radio.....	8
<b>6. Installation.....</b>	<b>9</b>
6.1 Configuring the Wireless Sensor Network.....	9
6.1.1 Adding Devices to the Wireless Sensor Network .....	10
6.1.2 Adding Communication Links to Devices.....	10
6.1.3 Configure Devices .....	11
6.1.3.1 Configuring the CWB100 .....	12
6.1.3.2 Configuring a Wireless Sensor.....	13
6.1.4 Send the Sensor Configuration File to the Datalogger.....	15
6.2 Programming the Datalogger .....	16
6.2.1 Datalogger Scan Rate.....	16
6.2.2 CWB100() Instruction .....	16
6.3 Wiring the CWB100 .....	18
6.4 Installing the Wireless Sensors .....	18
6.4.1 Connecting Sensors to the CWS900 .....	19
6.5 Verify that Data is being Transmitted .....	20
<b>7. Operation .....</b>	<b>20</b>
7.1 Auto-Discovery.....	20
7.2 RF Connection Type Setting.....	21
7.3 Advanced Programming .....	22
7.3.1 Configuration Strings.....	22
7.3.2 ArrayIndex() Function .....	23
7.3.3 CWB100Routes() Instruction .....	24
7.3.4 CWB100RSSI() Instruction .....	25

7.3.5	CWB100Diagnostics() Instruction .....	26
7.3.6	Field Names .....	26
<b>8.</b>	<b>Troubleshooting.....</b>	<b>27</b>
8.1	Checking Sensor Battery Voltage .....	27
8.2	Checking Sensor Links to the Base Station.....	27
8.3	Checking Sensor Signal Strength .....	28
<b>9.</b>	<b>Maintenance .....</b>	<b>28</b>
9.1	Removing a Wireless Sensor from the Network .....	28
9.2	Replacing a Wireless Sensor in the Network .....	29
9.3	Stopping or Loading a New Datalogger Program .....	29
<b>10.</b>	<b>Troubleshooting.....</b>	<b>29</b>
<b>Appendices</b>		
<b>A.</b>	<b>CRBasic Example Programs .....</b>	<b>A-1</b>
A.1	Basic Program .....	A-1
A.2	Program with Diagnostics .....	A-2
<b>B.</b>	<b>Using <i>Device Configuration Utility</i> for Sensor Setup.....</b>	<b>B-1</b>
<b>C.</b>	<b>Wireless Sensor Measurement Names and Meanings .....</b>	<b>C-1</b>
C.1	CCWS220 Measurements .....	C-1
C.2	CWS655 Measurements .....	C-1
C.3	CWS900 Measurements .....	C-2
C.4	Wind Vector Calculations .....	C-4
<b>D.</b>	<b>Battery Life .....</b>	<b>D-1</b>
D.1	Battery Characteristics .....	D-1
D.2	Effect of Polling Interval and Repeaters on Battery Life .....	D-1
<b>E.</b>	<b>Using SDI-12 with a Wireless Sensor Network ....</b>	<b>E-1</b>
E.1	Configure the CWB100.....	E-1
E.2	Configure Wireless Sensors .....	E-1
E.3	Available SDI-12 Commands .....	E-2
E.3.1	Address Query Command (!) .....	E-3
E.3.2	Acknowledge Active Command (a!).....	E-3
E.3.3	Start Measurement Command (aM!).....	E-3
E.3.4	Start Measurement Commands with Cyclic Redundancy Check (aMC!) .....	E-4
E.3.5	Concurrent Measurement Command (aC!) .....	E-4
E.3.6	Send Data Command (aD0! ... aD9!) .....	E-4
E.3.7	Send Identification Command (aI!).....	E-5
E.3.8	Start Verification Command (aV!).....	E-5

E.3.9	Get Base Radio Address (aX0!)	E-5
E.3.10	Get Sensor Field Names (aX1!)	E-6
E.3.11	Get Route by Name (aX2!)	E-6
E.3.12	Get Route by Number (aX3!)	E-6
E.4	Program the Datalogger	E-6
E.4.1	Basic SDI-12 CRBasic Example Program	E-7
E.4.2	SDI-12 CRBasic Example Program with Diagnostics	E-7

## Figures

4-1.	CWB100 Wireless Base Station	2
4-2.	CWS220 Wireless Infrared Radiometer	3
4-3.	CWS655 Wireless Water Content Reflectometer	4
4-4.	CWS900 Wireless Sensor Interface	5
6-1.	Wireless Sensor Planner Interface	9
6-2.	Connecting a computer to a sensor	14
6-3.	CWS900 Connector Pin-Out	20
C-1.	Input Sample Vectors	C-5
C-2.	Mean Wind Vector	C-6
C-3.	Standard Deviation of Direction	C-7

## Tables

5-1.	Temperature Accuracy, Uniformity, and Repeatability	6
6-1.	Values Transmitted by CWS Sensors	17
6-2.	CWB100() Error Codes	17
6-3.	CWB100 Connection to Datalogger	18
8-1.	Blue LED and Battery Voltage	27
8-2.	Red LED and Routes	28
A-1.	Wiring for CRBasic Examples	A-1
C-1.	CWS900 Measurement Configurations and Outputs	C-2
C-2.	CWS900 Measurement Names	C-3
D-1.	Approximate Battery Life Expectancy	D-2
E-1.	SDI-12 Commands	E-3

## CRBasic Examples

A-1.	Basic Wireless Sensor Network Program	A-1
A-2.	Program with Diagnostics	A-2
E-1.	SDI-12 Basic Program	E-7
E-2.	SDI-12 Program with Diagnostics	E-8



# Wireless Sensor Network

---

## 1. Introduction

A Campbell Scientific Wireless Sensor Network consists of a CWB100 Wireless Base Station which uses RF to communicate with one or more wireless sensors. The base station serves as the gateway to the network, communicating with a Campbell Scientific CR800-series, CR1000, or CR3000 datalogger via a control port as specified in the CRBasic **CWB100()** instruction. The base station communicates with all wireless sensors in the network using its own sensor network protocol. Any sensor can serve as an RF repeater to communicate with other wireless sensors. A sensor can route its transmissions through up to three other sensors on the way back to the base station.

---

**NOTE**

Wireless Sensor Networks are not meant to move a lot of data quickly. When planning a network measurement rate, allow for one minute per sensor.

---

## 2. Precautions

- READ AND UNDERSTAND the *Safety* section at the front of this manual.
- For sensors being used with the CWS900 Configurable Wireless Sensor Interface, refer to the sensor manual for operating details.

## 3. Initial Inspection

The CWB100 ships with

- Screw, 2 each pn 505
- Grommet, 2 each pn 6044
- Communications Cable, pn 26862
- USB Cable, Type A Male to Type B Male, pn 17648

CWS220, CWS655, and CWS900 sensors ship with the user-chosen battery pack, either alkaline or rechargeable.

Check the packaging and contents of the shipment. If damage occurred during transport, immediately file a claim with the carrier. Contact Campbell Scientific to facilitate repair or replacement.

Check model information against the shipping documents to ensure the expected products and the correct lengths of cable are received. Model numbers are found on each product. On cables and cabled items, the model number is usually found at the connection end of the cable. Report any shortages immediately to Campbell Scientific.

## 4. Overview

The CWB100 Wireless Base Station and one or more wireless sensors are first configured using Campbell Scientific's *Wireless Sensor Planner*, *Network Planner*, or *Device Configuration Utility* software. Wireless sensors interface with the PC for configuration via the A205 CWS-to-PC Interface. One A205 is required per wireless system.

A CRBasic datalogger must be programmed to interface with the CWB100 and determine a polling interval. Sensors are synchronized to measure at the top of the datalogger scan interval. The base station polls all sensors and stores the collected data until the datalogger requests it through the **CWB100()** instruction.

### 4.1 CWB100 Wireless Base Station

In the wireless sensor network, the CWB100 serves as the master radio and the interface between the datalogger and the wireless sensors. It holds information for the routes to all the sensors in the network and is responsible for polling the sensors. It synchronizes wireless sensors to make their measurements at the beginning of the datalogger scan.



FIGURE 4-1. CWB100 Wireless Base Station

### 4.2 Wireless Sensors

The CWS220 and CWS655 wireless sensors, and the CWS900 wireless sensor interface make measurements at the same time, at the top of the datalogger scan. Each sends its data to the CWB100 when polled. Only the last measurement is stored in preparation to send to the CWB100.

Throughout this manual, the CWS220, CWS655, and CWS900 are collectively called wireless sensors unless referred to specifically.

#### 4.2.1 CWS220 Wireless Infrared Radiometer

The CWS220 is a Wireless Infrared Radiometer that incorporates Apogee Instrument's SI-111 infrared sensor for non-contact surface temperature measurements. It senses the infrared radiation being emitted by the target. The CWS220 is used to measure leaf, canopy, and average surface temperature.

It returns five measurements. Default measurement names and meanings are shown in Appendix C.1, *CCWS220 Measurements (p. C-1)*.

The CSW220 includes a thermopile for measuring a millivolt output dependent on the target to sensor body temperature difference. A thermistor measures the temperature of the sensor body. The sensor body temperature is used to reference the target temperature.

The calibration coefficients used for determining the target temperature by means of the Stefan-Boltzman equation are stored in the CWS220 at the time of calibration.



FIGURE 4-2. CWS220 Wireless Infrared Radiometer

#### 4.2.2 CWS655 Wireless Soil-Water Probe

The CWS655 is based on Campbell Scientific's CS655 Soil Water Content Reflectometer. It returns nine measurements. Default measurement names and meanings are shown in Appendix C.2, *CWS655 Measurements (p. C-1)*.

The water content information is derived from the probe's sensitivity to the dielectric permittivity of the medium surrounding its stainless-steel rods. Dielectric permittivity increases with volumetric water content in a predictable fashion. During water content measurements, the CWS655 is configured as a water content reflectometer with the two parallel rods forming an open-ended

transmission line. A differential oscillator circuit is connected to the rods. An oscillator state change is triggered by the return of a reflected signal from one of the rods. The two-way travel time of the oscillator-induced electromagnetic waves on the rod increases with increasing volumetric water content of the surrounding media, hence the name water content reflectometer.

The electrical conductivity (EC) of the surrounding medium is derived from signal attenuation information. This is accomplished by exciting the rods with a known non-polarizing waveform and determining the signal attenuation due to EC. The EC measurement is used to correct the oscillator period and is also available to the user.

Temperature is measured with a thermistor that is in contact with one of the rods. The thermistor makes a point measurement near the top of the sensor rods. In a typical application, this will correspond to a soil temperature reading near the soil surface so care should be taken in interpreting soil temperature data.



FIGURE 4-3. CWS655 Wireless Water Content Reflectometer

#### 4.2.3 CWS900 Configurable Wireless Sensor Interface

The CWS900 measures analog voltages, low-level AC, and pulses, and can supply an excitation voltage for powering some sensors or making bridge measurements. It has a sealed connector for use with a variety of sensors, such as temperature/humidity probes, rain gages, wind monitors, or other low-power sensors. The type of measurement that the sensor performs is configured using *Wireless Sensor Planner*, *Network Planner*, or *Device Configuration Utility* software. The number of measurements returned ranges from 4 to 12 and depends on the chosen configuration. Refer to the software for the measurement names and meanings for that configuration. See Appendix C, *Wireless Sensor Measurement Names and Meanings (p. C-1)*, for a list of all measurement names and meanings.



FIGURE 4-4. CWS900 Wireless Sensor Interface

## 5. Specifications

### 5.1 CWB100

<b>Operating Temperature:</b>	-25 to 50 °C (-13 to 122 °F)
<b>Operating Relative Humidity:</b>	Store and use in a noncondensing (desiccated) environment.
<b>Power Requirements</b>	
<b>Voltage:</b>	4.5 to 22 Vdc
<b>Typical Current Drain</b>	
<b>Standby:</b>	<1 mA
<b>Receiving:</b>	10 mA
<b>Transmitting:</b>	20 mA
<b>Memory:</b>	Can store routing table for approximately 44 to 64 wireless sensors, depending on field names
<b>Communication:</b>	Serial Protocol or USB
<b>Terminal Block Connector:</b>	Bidirectional serial datalogger connection
<b>USB Port:</b>	Computer connection for configuration
<b>Antenna Connector:</b>	Reverse Polarity SMA (RPSMA)
<b>Dimensions:</b>	10.8 x 4.4 x 4.4 cm (4.25 x 1.75 x 1.75 in)
<b>Weight:</b>	140 g (5 oz)

## 5.2 CWS220 Specifications

Temperature Range	Absolute Accuracy	Uniformity	Repeatability
-10 to 65 °C	±0.2 °C	±0.1 °C	±0.05 °C
-40 to 70 °C	±0.5 °C	±0.3 °C	±0.1 °C

<b>Response Time:</b>	<1 s to changes in target temperature
<b>Wavelength Range:</b>	8 to 14 μm (corresponds to atmospheric window)
<b>Field of View (FOV):</b>	22° half angle
<b>Operating Temperature:</b>	-25 to 50 °C (-13 to 122 °F)
<b>Operating Relative Humidity:</b>	0 to 100%
<b>Weather Resistance:</b>	IP67 rating for sensor and battery pack (battery pack must be properly installed; each sensor is leak tested)
<b>Power:</b>	2 AA batteries with a battery life of 1 year, assuming sensor samples taken every 10 minutes
<b>Average Current Drain:</b>	300 μA with 15-minute polling
<b>Dimensions:</b>	15 x 6 x 4.5 cm (5.9 x 2.4 x 1.8 in)
<b>Weight:</b>	270 g (9.6 oz)

## 5.3 CWS655 Specifications

<b>Accuracy</b>	
<b>Volumetric Water Content:</b>	±3% VWC typical in mineral soils that have solution electrical conductivity ≤ 10 dS/m. Uses Topps Equation (m <sup>3</sup> /m <sup>3</sup> ).
<b>Relative Dielectric Permittivity:</b>	±(3% of reading + 0.8) for solution EC ≤ 8 dS/m (1 to 40 dielectric permittivity range) ±2 for solution EC ≤ 2.8 dS/m (40 to 81 dielectric permittivity range)
<b>Bulk Electrical Conductivity:</b>	±(5% of reading + 0.05 dS/m)
<b>Soil Temperature:</b>	±0.5 °C
<b>Operating Temperature:</b>	-25 to 50 °C (-13 to 122 °F)

<b>Operating Relative Humidity:</b>	0 to 100%
<b>Weather Resistance:</b>	IP67 rating for sensor and battery pack (battery pack must be properly installed; each sensor is leak tested)
<b>Power:</b>	2 AA batteries with a battery life of 1 year, assuming sensor samples taken every 10 minutes
<b>Average Current Drain:</b>	300 $\mu$ A with 15-minute polling
<b>Body Dimensions:</b>	14.5 x 6 x 4.5 cm (5.7 x 2.4 x 1.8 in)
<b>Rod Diameter:</b>	3.2 mm (0.13 in)
<b>Rod Length:</b>	12 cm (4.7 in)
<b>Weight:</b>	216 g (7.6 oz)

## 5.4 CWS900 Specifications

<b>Analog Input Range:</b>	-1 to 2.5 Vdc
<b>Analog Measurement Accuracy</b>	
<b>0 to 50 °C:</b>	$\pm(0.02\%$ of reading + 2 $\mu$ V)
<b>-35 to 70 °C:</b>	$\pm(0.05\%$ of reading + 2 $\mu$ V)
<b>Analog Measurement Resolution:</b>	0.3 $\mu$ V
<b>Excitation Voltage:</b>	2.5 V, 3.3 V, 5.0 V (20 mA maximum)
<b>Excitation Voltage Accuracy:</b>	$\pm 2\%$ from -35 to 70 °C
<b>Bridge Measurement Accuracy (+2.5 V excitation only)<sup>1</sup></b>	
<b>0 to 50 °C:</b>	$\pm(0.03\%$ of reading + 3 $\mu$ V)
<b>-35 to 70 °C:</b>	$\pm(0.07\%$ of reading + 3 $\mu$ V)
<b>Low Level AC Input:</b>	20 mV minimum, 10 kHz maximum frequency
<b>Switch Closure</b>	
<b>Maximum Count Rate:</b>	100 Hz
<b>Minimum Open Time:</b>	5 ms
<b>Minimum Closed Time:</b>	5 ms
<b>Maximum Bounce Time:</b>	4 ms
<b>Body Temperature Accuracy:</b>	$\pm 0.2$ °C
<b>Operating Temperature:</b>	-25 to 50 °C (-13 to 122 °F)

<b>Operating Relative Humidity:</b>	0 to 100%
<b>Weather Resistance:</b>	IP67 rating for sensor and battery pack (battery pack must be properly installed; each sensor is leak tested)
<b>Power:</b>	2 AA batteries with a battery life of 1 year, assuming sensor samples taken every 10 minutes
<b>Average Current Drain:</b>	300 $\mu$ A with 15-minute polling
<b>Dimensions:</b>	15 x 6 x 4.5 cm (5.9 x 2.4 x 1.8 in)
<b>Weight:</b>	184 g (6.5 oz)

<sup>1</sup>Does not include sensor and measurement noise, or external bridge resistor errors.

## 5.5 Internal Radio

<b>Radio Type:</b>	25 mW Frequency Hopping Spread Spectrum (FHSS)
<b>Frequency</b>	
<b>902 – 928 MHz:</b>	CWB100, CWS220, CWS655, CWS900 CWB100A, CWS220A, CWS655A, CWS900A
<b>868 MHz:</b>	CWB100E, CWS220E, CWS655E, CWS900E
<b>FHSS Channels</b>	
<b>902 – 928 MHz:</b>	50
<b>868 MHz:</b>	16
<b>Transmitter Power Output:</b>	25 mW (+14 dBm)
<b>Receiver Sensitivity:</b>	-110 dBm (0.1% frame error rate)
<b>Current Drain</b>	
<b>Standby:</b>	3 $\mu$ A
<b>Receiving:</b>	18 mA typical (full run)
<b>Transmitting:</b>	45 mA
<b>Average Operating Current:</b>	15 $\mu$ A (with 1-second access time)
<b>Additional Features:</b>	GFSK modulation, data interleaving, forward error correction, data scrambling, RSSI reporting

## 6. Installation

The following hardware and software are required:

- A205 CWS-to-PC Interface (pn 25933)
- USB Cable, Type A Male to Type B Male (pn 17648)
- *Wireless Sensor Planner*, available for download at [www.campbellsci.com/cwb100](http://www.campbellsci.com/cwb100)
- or-
- Network Planner*, included in *LoggerNet*
- CWB100 USB Driver, available at [www.campbellsci.com/cwb100](http://www.campbellsci.com/cwb100)
- A205 CWS Interface USB Driver, available at [www.campbellsci.com/a205](http://www.campbellsci.com/a205)
- Compatible datalogger with a computer connection

### 6.1 Configuring the Wireless Sensor Network

Use *Wireless Sensor Planner* or *Network Planner* to design and configure the network. The following steps show *Wireless Sensor Planner*, but the same steps apply to *Network Planner*.

Open *Wireless Sensor Planner*. The window includes a **Device Palette**, **Drawing Canvas**, **Configure Devices** list, and **Device Summary** as shown in FIGURE 6-1.

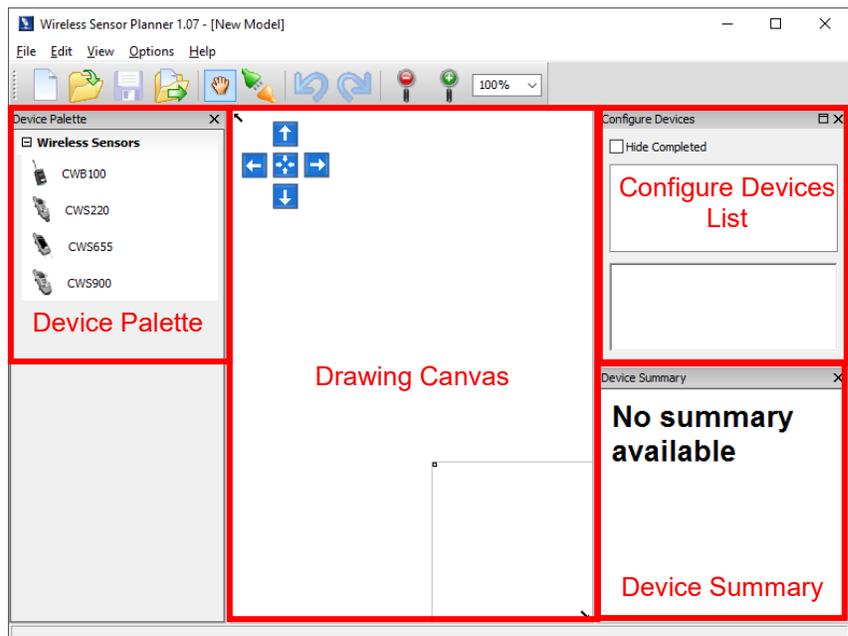


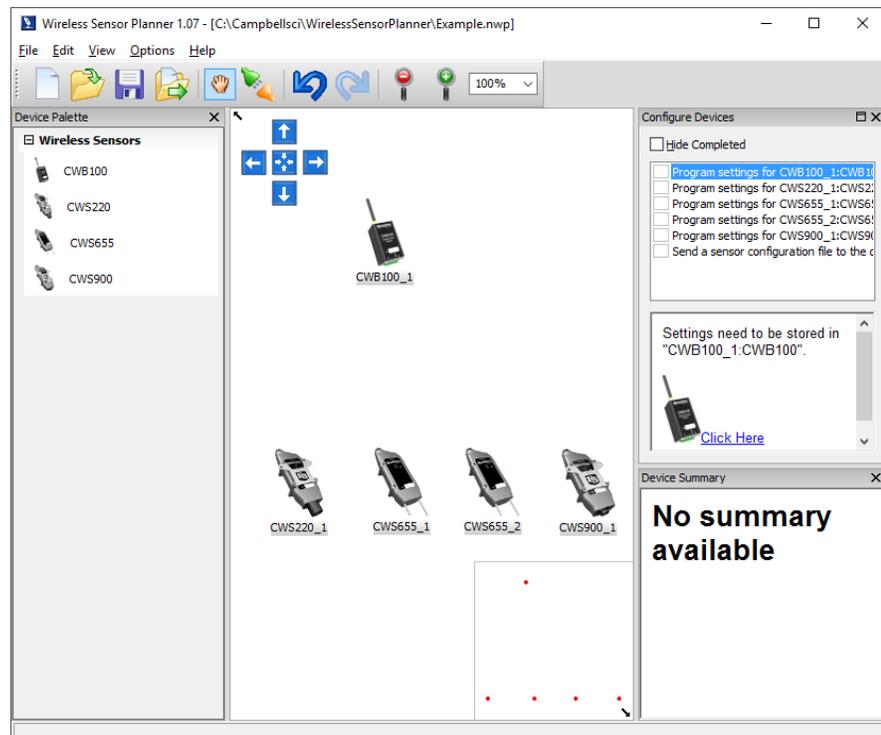
FIGURE 6-1. *Wireless Sensor Planner* Interface

### 6.1.1 Adding Devices to the Wireless Sensor Network

Click on the device in the **Device Palette**, then click anywhere on the **Drawing Canvas** to place it there. Double-click on the placed device, and enter its **Base Station Address** or **Radio Address** in the **Properties for Device** window. The address is the serial number, which is found on the device label.

When placing a CWS900, a **New Station Properties** window immediately opens. Enter the **Radio Address**, and then select the **Measurement Configuration** to be used. If excitation voltage will be used, select an **Excitation Voltage** option and enter an **Excitation Delay**.

Repeat with each device to be included in your wireless sensor network. The icons on the **Drawing Canvas** may be repositioned by selecting the icon and dragging it to a different part of the canvas.

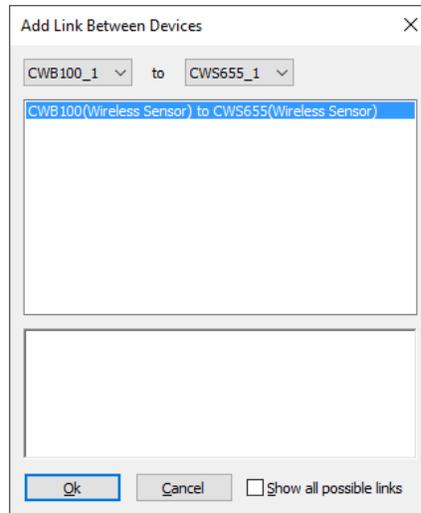


### 6.1.2 Adding Communication Links to Devices

The next step is to add links from the base station to each sensor. This represents the radio communication link between the devices. To create links, click the **Link Tool** icon on the toolbar (shown below).

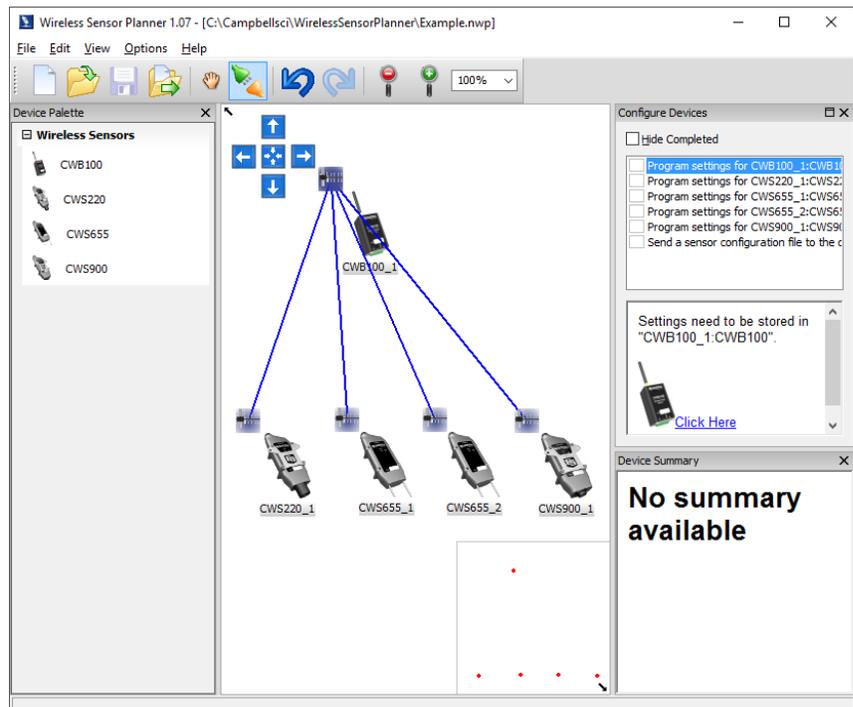


Select the CWB100 and then click on a sensor to complete a link. An **Add Link Between Devices** window will open showing the link. Click **OK** to accept the link.



The link is now represented on the **Drawing Canvas** by a line between the base station and the sensor. Repeat for each sensor.

When linking to a CWS900, an additional window opens showing the addresses of the base station and the settings of the CWS900. Click **Apply** to accept the settings.



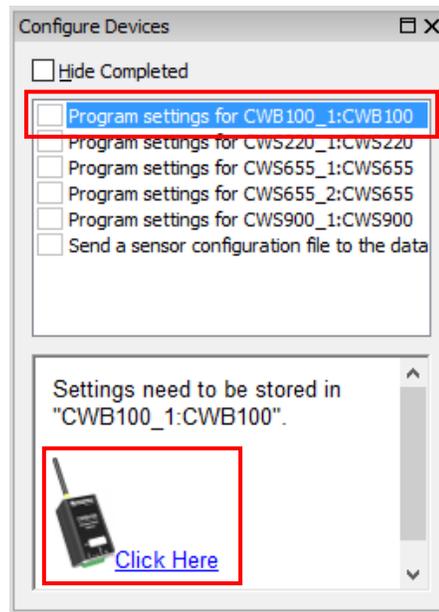
### 6.1.3 Configure Devices

After the links are made in *Wireless Sensor Planner*, each device must be configured. These are listed in the **Configure Devices** panel. If the **Configure Devices** list is not visible to the right of the **Drawing Canvas**, enable it by navigating to **View | Show Device Configuration List**. The **Configure**

**Devices** list shows each task that needs to be completed to set up the network successfully.

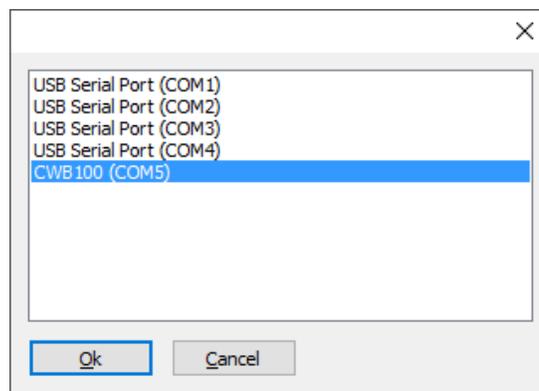
### 6.1.3.1 Configuring the CWB100

Select **Program settings for CWB100\_1:CWB100** in the list, then click **Click Here** in the panel below.

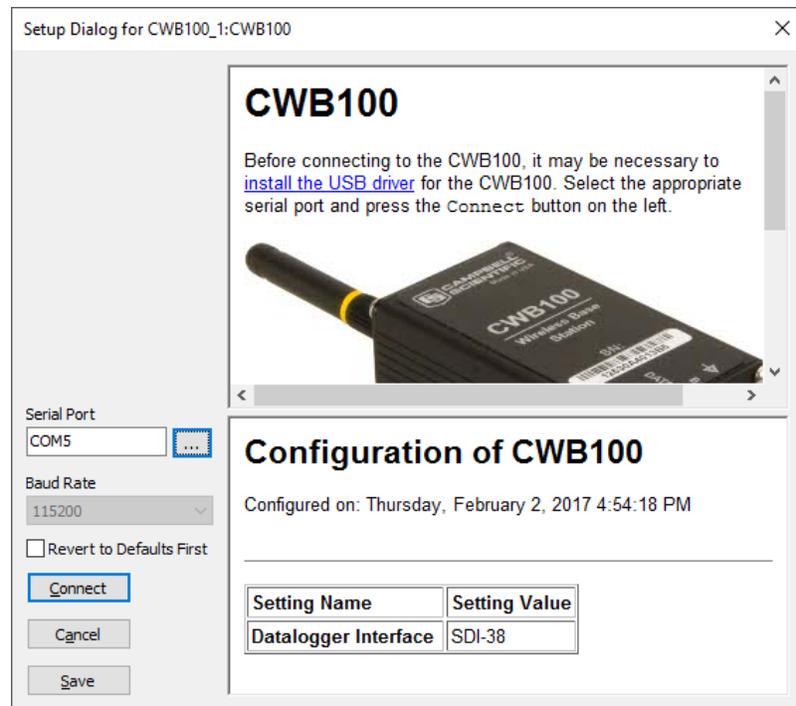


When the **Setup Dialog** window opens, connect the CWB100 to a USB port on your computer using the included cable (pn 17648).

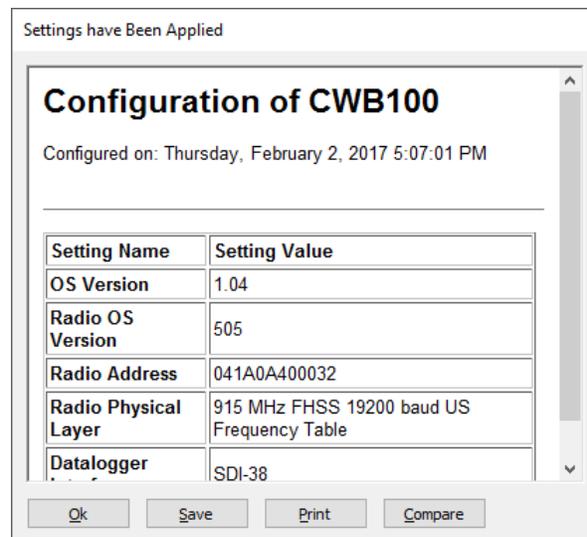
Click **Serial Port**. Select the port named CWB100 (COMxx). Click **OK**.



A table showing configuration settings is shown. To save these settings in a XML file for future reference, click **Save**. To send these settings to the CWB100, click **Connect**.

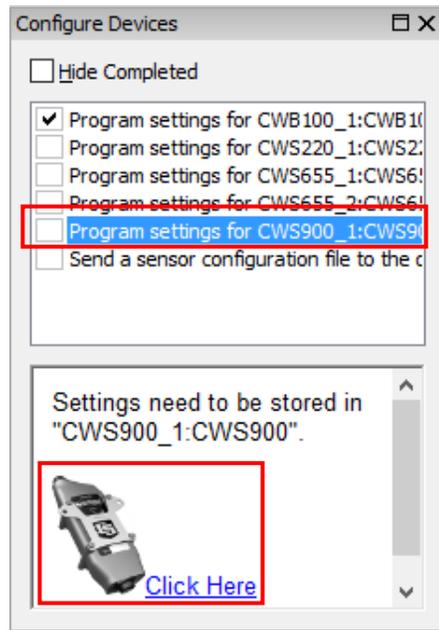


After the new settings have been sent to the device, the **Settings have Been Applied** window opens. Options are available for saving the settings to a XML file, printing the settings, and comparing the settings to another XML file. When finished with these actions, click **OK**.



### 6.1.3.2 Configuring a Wireless Sensor

Select the option to program settings for the desired sensor in the **Configure Devices** list, then click on **Click Here** in the panel below.



Remove the battery pack from the sensor. Wait 15 seconds. Connect the A205 to the four-pin connector near the top of the sensor. The A205 label should be facing toward the sensor antenna. Connect the A205 to the computer with the USB cable as shown in FIGURE 6-2.



FIGURE 6-2. Connecting a computer to a sensor

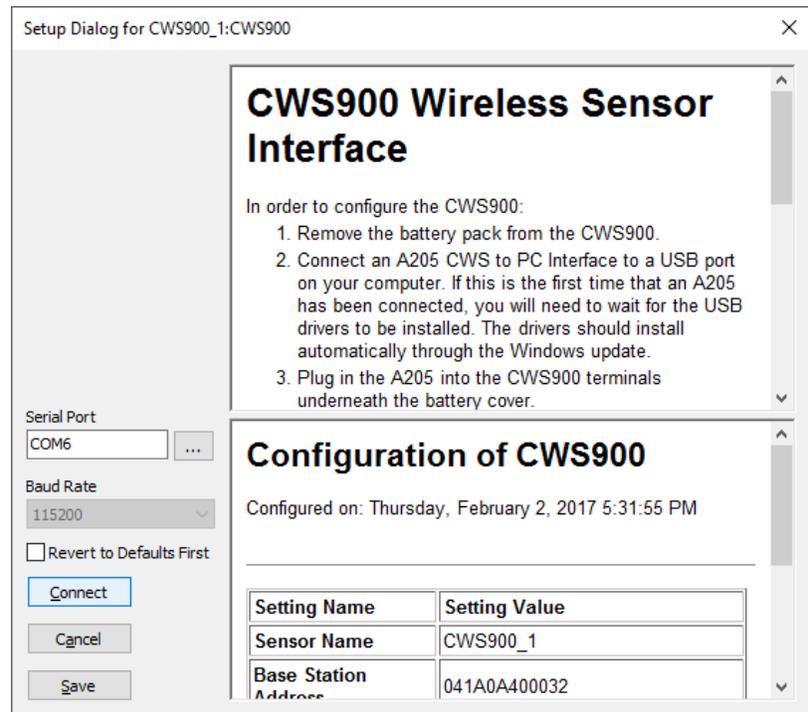
In the **Setup Dialog** window, click **Serial Port**. Select the port where the A205 is connected to the computer and click **OK**.

Click **Connect**. Press the **Setup** button on the back of the sensor. Settings will be transferred and applied. In the **Settings have Been Applied** window,

options are available for saving the settings to a XML file, printing the settings, and comparing the settings to another XML file. When finished with these actions, click **OK**.

**NOTE** The **Setup** button must be pressed to enable communications between the wireless sensor and computer.

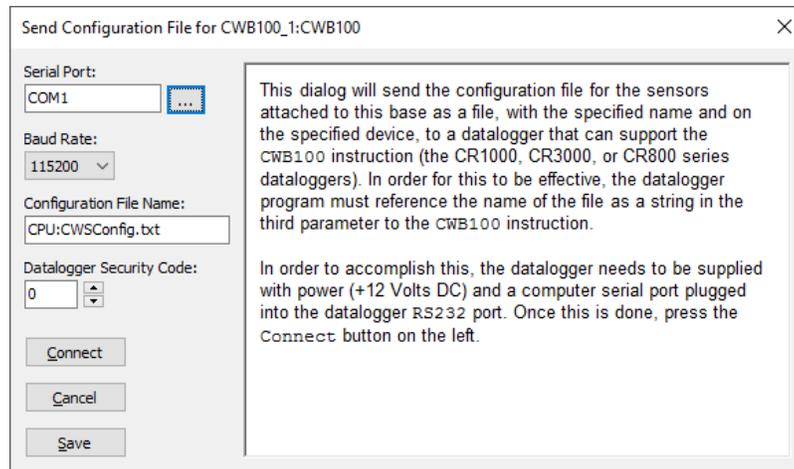
Repeat these steps for each sensor in the network.



#### 6.1.4 Send the Sensor Configuration File to the Datalogger

This step is optional, but is recommended for any network with more than one sensor. Select **Send a sensor configuration file to the datalogger attached in the Configure Devices list**, then click on **Click Here** in the panel below.

Apply power to the datalogger. Physically connect the computer to the datalogger through one of its ports. Select that port in the **Serial Port** field. If the datalogger has a security code, enter it in the **Datalogger Security Code** field. Click **Connect** to send the configuration file to the datalogger.



## 6.2 Programming the Datalogger

Programming basics are provided in the following section. More programming options are discussed in Section 7, *Operation* (p. 20). Complete program examples can be found in Appendix A, *CRBasic Example Programs* (p. A-1).

### 6.2.1 Datalogger Scan Rate

The CWB100 synchronizes the network measurement rate to match the main datalogger scan rate, so the datalogger scan rate must give enough time for all sensors to transmit their data. When creating a program at first, set the scan rate to allow for 1 minute per sensor. For example, if the wireless sensor network includes two sensors, the scan rate should be at least 2 minutes, or 120 seconds. If necessary, the scan rate can later be optimized using results from the **CWB100Diagnostics()** instruction, discussed in Section 7.3.5, *CWB100Diagnostics() Instruction* (p. 26).

A more typical scan rate would be 5 to 15 minutes to increase the life of the batteries:

```
Scan(5,Min,0,0)
.
.
.
NextScan
```

---

**NOTE** See Appendix D, *Battery Life* (p. D-1), for details on battery life.

---

### 6.2.2 CWB100() Instruction

The **CWB100()** instruction is used to initiate wireless sensor network measurements and to retrieve data from the CWB100 base station.

**CWB100()** instruction and parameters are as follows:

```
CWB100(CWBPort, CWSDest, CWSConfig)
```

- **CWBPort** specifies which terminal the CWB100 **Data/A** terminal is wired to.

- **CWSDest** must be a variable array large enough to store all values returned by the wireless sensors. If the array is not large enough, it will be filled with **NAN**.
- **CWSConfig** is an optional parameter. It can be either a string or a text file. When the configuration file is sent to the datalogger in *Wireless Sensor Planner* or *Network Planner*, the file name is shown in the **Configuration File Name** field of the window. The default name and location is CPU:CWSConfig.txt. This string or file name must be in quotation marks.

The size of the **CWSDest** array must be large enough to store all values returned by the wireless sensors. The number of values transmitted is shown in TABLE 6-1. For example, if the wireless sensor network includes a CWS655 (9 values) and a CWS900 configured to measure a 109 Temperature Probe (4 values), the array must be dimensioned to at least 13:

`Public WSN(13)`

Note that since the array is sized to fit the number of incoming values exactly, no additional wireless sensors may be added to this network without first modifying the datalogger program.

**TABLE 6-1. Values Transmitted by CWS Sensors**

Sensor	Values Transmitted
CWS220 Wireless Infrared Radiometer	5
CWS655 Wireless Water Content Reflectometer	9
CWS900 Wireless Sensor Interface*	4 – 12

\*See Appendix C, *Wireless Sensor Measurement Names and Meanings* (p. C-1), for more information on number of values transmitted by the CWS900.

One of the following error codes may be written to the first element of the destination array if there is a problem with communication between the datalogger and the CWB100 (see Section 10, *Troubleshooting* (p. 29), for more information):

**TABLE 6-2. CWB100() Error Codes**

Error Code	Meaning of Error Code
-1	The datalogger has failed after three consecutive attempts to collect data from the base station.
-2	The signature verifying communication is wrong.
-3	The signature of the names held by the datalogger is different from the signature of names held by the CWB100.

## 6.3 Wiring the CWB100

To prevent damage to the datalogger and CWB100, power down the datalogger before wiring, and connect the **G** terminal first.

**TABLE 6-3. CWB100 Connection to Datalogger**

CWB100 Terminal	Function	Wire Color	Datalogger Connection Terminal
+12V	12 V Power	Red	12V
G	Power Ground	Black	G
DATA/A	Signal	Green	Odd-numbered Control Port or U <sup>1</sup> terminal configured for CWB100 communication
B	Not Used	No Connection	No Connection
RG	Not Used	No Connection	No Connection

<sup>1</sup>U channels are automatically configured by the CWB100() measurement instruction.

After the datalogger is powered, the LED activity indicator on the CWB100 will flash red and green rapidly as it goes through the power-up sequence. Then, the red LED will flash every 5 seconds.

## 6.4 Installing the Wireless Sensors

The installation can be tested in the office, but remove power from the CWB100 and all sensors before transporting to the field. This will erase the routing table and halt network traffic while in transit. The routing table must be erased because the routes found in the lab will likely be different from the routes used in the field.

For best results, install the network as follows:

- Install the CWB100 and datalogger first, and power them up together.
- After the CWB100 and datalogger are powered, set up the sensors closest to the base first and work your way outward, especially if you suspect there will be repeaters in your network.
- Mount the base station antenna and the wireless sensors as high as possible and away from metal objects for the best radio signal.

Follow the process below to set up a sensor:

1. For a CWS900, connect the attaching sensor. Press the connector firmly into the sockets and rotate the locking nut to hold it in position.
2. Attach the sensor battery pack to the 4-pin connector. The red LED will turn on solid for about 4 seconds and then turn off. Tighten the four battery-pack screws.
3. Press and hold the **Setup** button on the back of the sensor. The blue LED will flash up to four times to indicate the battery voltage (see Section 8.1,

*Checking Sensor Battery Voltage (p. 27)*. Keep holding the **Setup** button until the blue LED starts flashing again about once per second. This indicates that the sensor is searching for a route to the base station. This process is known as auto-discovery.

4. Release the **Setup** button and wait for the sensor to establish communication with the CWB100 radio.

---

**NOTE**

The time required to establish the radio link will depend on the number of sensors in the network and the RF path to the base. It can range from a few minutes to a couple of days.

---

Once the radio link is established, the red sensor LED will flash, and the sensor will be established on the CWB100 routing table. During this process the red LED on the CWB100 stays on longer than usual, indicating that it is receiving a transmission. Afterward, the blue sensor LED will flash when the sensor makes a measurement, and the red sensor LED will flash when the data is transmitted to the CWB100.

If the **Setup** button is not pressed to initiate auto-discovery, the wireless sensor will automatically begin auto-discovery 12 hours after being powered.

For additional information about CWS sensor diagnostics, see Section 8, *Troubleshooting (p. 27)*.

### 6.4.1 Connecting Sensors to the CWS900

For sensors that are compatible with the CWS900 but do not have the CWS connector option available, or for compatible sensors not sold by Campbell Scientific, a connector will need to be added to the sensor in order to interface it with the CWS900. Two options are available:

- A CWS900 Mating Connector may be purchased from Campbell Scientific and connected to the sensor pigtail. This will require soldering the sensor wires to pins on the connector and ensuring the connections are waterproof.
- An A150 Single Sensor Terminal Case with a CWS cable may be purchased from Campbell Scientific and the sensor connected to the CWS cable through a terminal block. This is required to interface with the CS450 Pressure Transducer.

The CWS900 connector pin-out is shown in FIGURE 6-3.

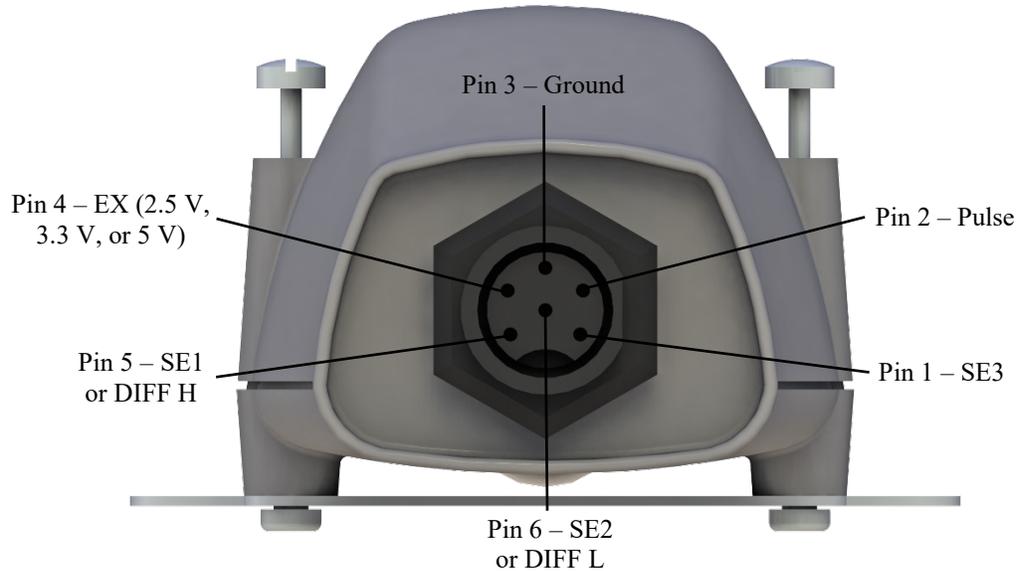


FIGURE 6-3. CWS900 Connector Pin-Out

After connecting the non-standard sensor to the CWS900, test the connections and sensor performance with *Device Configuration Utility* software. See Appendix B, *Using Device Configuration Utility for Sensor Setup (p. B-1)*, for details.

## 6.5 Verify that Data is being Transmitted

After the sensors have established links to the CWB100 radio, verify that data is being received by looking at the *CWSDest* array in datalogger support software.

## 7. Operation

The CWB100() instruction sets the sensor measurement and polling interval. Each measurement is synchronized with the top of the datalogger main scan interval. To conserve battery power, the sensors do not initiate any communication except during the network discovery step of installation. Instead they transmit only when polled by the base station. If no acknowledgment is received from the base, then it will do one retry. If the base does not receive data from a sensor it polls, it will do one retry. Between the base and the sensor, two retries will be made before storing NANs in the array indices for that sensor.

Each hop takes a minimum of 5 seconds, so a sensor hopping through three repeater sensors would require at least 20 seconds to be polled.

### 7.1 Auto-Discovery

During installation, sensors find their own RF route to the base. A sensor can hop through up to three repeater sensors for up to four hops to the base. It can act as a repeater for up to eight wireless sensors.

**NOTE**

---

When a sensor is used as a repeater for another sensor, it is effectively polled three additional times, which will shorten its battery life.

---

If the **Setup** button is held, the blue LED will begin flashing about once per second, and the radio will attempt to find a link to the base station. This process is called auto-discovery. It will do this even if it already has a link to the base.

The sequence for the search is:

1. Attempt to link to base. The search attempt takes from 30 seconds to 5 minutes. During this time the blue LED blinks about once per second.
2. If the initial attempt fails to link to the base station, the wireless sensor will delay 5 minutes before attempting to link again. The blue LED will be off during this delay. The sensor will try to link to the base at a five-minute interval a total of three times.
3. If the 5-minute interval attempt fails to link to the base station, the wireless sensor will delay one hour before attempting to link again. The blue LED will be off during this delay. The sensor will try to link to the base at a one-hour interval a total of four times.
4. If the hourly attempts fail, the wireless sensor will try to connect to the base station every 12 hours. At the beginning of each 12-hour interval, it will make three attempts spaced 5 minutes apart. This sequence will be repeated indefinitely.

If the **Setup** button is not pressed to initiate auto-discovery, the wireless sensor will automatically begin a search for the base station 12 hours after being powered.

If discovery has been successful and the base station later loses power, the wireless sensor keeps making measurements at the same polling interval. It does not transmit the measured values because it only transmits data when polled by the CWB100 base station. After 4 to 12 hours of no communication with the base station, the sensor will start the auto-discovery mode and follow the schedule shown above.

If a sensor has not responded to six polls by the CWB100 and it has been more than two hours since the last communication, the CWB100 will stop polling that sensor.

## 7.2 RF Connection Type Setting

The **RF Connection Type** setting is found in the **Settings Editor** tab of *Device Configuration Utility* when connected to a wireless sensor.

This setting is used to control how the sensor radio discovers its path to the base station. The default setting is **Auto-Connect**, which is what should be used for most networks. With this setting, a sensor first searches for a direct path (one hop) to the base. It does this three times over 15 minutes. It then searches for a path to the base using a repeater. It also does this three times.

It can be useful to change this setting from its default in certain instances. For example, if the wireless sensor network is composed of many sensors that are all within range of the base, changing it to **Connect Direct to Base** will force all the connections to the base. There will be no attempt to use repeaters. If every sensor is within range of the base, this saves battery power.

In another situation, there may be a sensor in a location that permits a direct connection to the base, but misses a significant number of data polls. Changing the **RF Connection Type** setting to **Connect Through Repeater** will force it to go through a repeater, and it will not attempt any direct connection to the base, which may result in fewer missed polls.

When changing the setting to **Connect Through Repeater**, battery life of the repeater should be taken into consideration (see Appendix D.2, *Effect of Polling Interval and Repeaters on Battery Life* (p. D-1)).

## 7.3 Advanced Programming

Section 6.2, *Programming the Datalogger* (p. 16), discusses the basic datalogger program to set up a network of two wireless sensors. More advanced programming concepts are described below.

### 7.3.1 Configuration Strings

The configuration string predefines field names and locations for measurements in the destination array. If radio addresses are entered in *Wireless Sensor Planner* and *Network Planner*, a configuration string can be automatically generated using the field names entered in the sensor properties.

With a configuration string, field names and locations are set so they can be used within the program for processing. Without a configuration string, sensor measurements are shown in the destination array in the order in which the sensors discover the base, which is unpredictable with more than one sensor in a network. If not using a configuration string, see Section 7.3.2, *ArrayIndex() Function* (p. 23), for another way of locating sensor measurements.

The configuration string is a list of comma-separated sensor descriptions, where each sensor description is a space-separated list of the following items:

Configuration String Format		
Sensor ID	Radio ID of the sensor	Required
Sensor Name	Name of the sensor	Required. Does not need to match the setting of the sensor.
Number of Fields	Number of values to be transmitted by the sensor	Optional. Defaults to 1 if not specified.
1 <sup>st</sup> Field Name	Name of first value transmitted	Optional. Treated as unknown if not specified.
2 <sup>nd</sup> Field Name	Name of second value transmitted	Optional. Treated as unknown if not specified.
N <sup>th</sup> Field Name	Name of last value transmitted	Optional. Treated as unknown if not specified.

It is good practice to keep the custom field names short to minimize radio transmission time from a wireless sensor. If the sensor name is different from the sensor setting, the sensor name in the configuration string will override the sensor setting.

Unknown field names are serialized with an uncommon root name, “\_F”, with the serial number starting at one greater than the number of field names specified for that sensor.

#### Configuration String Example:

```
07530A4002D1 CWS900 4 Ts Ti BV SS, 18534B402008 CWS655 9
VWC EC Ts Ka PA VR Ti BV SS
```

Using this configuration string example, index 6 of the destination array would be the EC field of the CWS655 with the radio address of 18534B402008.

The configuration string may be written directly in the **CWB100()** instruction or in a separate text file that is stored on the datalogger. The configuration strings generated by *Wireless Sensor Planner* and *Network Planner* are written to text files. See the two CRBasic examples below.

```
CWB100 (1,CWSArray()),"07530A4002D1 CWS900 4 Ts Ti BV SS,
18534B402008 CWS655 9 VWC EC Ts Ka PA VR Ti BV SS")
```

```
CWB100 (1,CWSArray()),"CPU:CWSConfig.txt")
```

Configuration string files can be sent to the datalogger through *Wireless Sensor Planner*, *Network Planner*, and through File Control.

### 7.3.2 ArrayIndex() Function

When a configuration string is not used, the **ArrayIndex()** function is used to return the index of a named element in an array. If the named element is not found, the function returns **0**, which will result in a **Variable Out of Bounds** error.

Without a configuration string, sensor measurements are shown in the destination array in the order in which the sensors are discovered by the base. It is not known until the sensors are discovered where in the array each sensor's measurement values will be stored. The names of the values returned by the sensors are known, however. Thus, the **ArrayIndex()** function can be used to return the correct index value so that further processing such as output processing or unit conversion can be performed.

The instruction has the following format:

```
ArrayIndex (Name)
```

The *Name* parameter is a string that contains the name of the value for which an index is desired. All wireless sensors have a default sensor name and a field name for each returned value. For instance, a CWS900 with a sensor name of CWS900\_3AF has a 109 Temperature Probe attached, which returns four default values: **Ts** (109 temperature), **Ti** (internal temperature), **BV** (battery voltage), and **SS** (signal strength). Thus, the default name for the 109

temperature measurement would be CWS900\_3AF\_Ts. This would be entered as a string in the **ArrayIndex()** function:

```
Public CWSArray(13)
Public Tair_F

DataTable (WSNData,True,-1)
  DataInterval (0,5,Min,0)
  Sample (1,CWSArray(ArrayIndex("CWS900_TS")),FP2)
  FieldNames ("CWS900_Temp")
  Sample (1,CWSArray(ArrayIndex("CWS655_VWC")),FP2)
  FieldNames ("CWS655_VWC")
EndTable

BeginProg
  Scan (5,Min,0,0)
  CWB100 (1,CWSArray())
  Tair_F = CWSArray(ArrayIndex("CWS900_TS")) * 1.8 + 32
  CallTable WSNData
NextScan
EndProg
```

In this example, wireless sensor data is written to a destination array called **CWSArray**. Because the default measurement name is known, it is entered in quote marks in the *Name* parameter of the **ArrayIndex()** instruction. The datalogger can then search for that name in the destination array and reference its position in the array for program use regardless of the order in which the sensor was discovered.

CRBasic instruction **FieldNames()** is used after each output instruction to provide a unique output name to the values. If more than one output instruction is in the data table and **FieldNames()** is omitted, a compiler error will be returned indicating duplicate output names. If the compiler error is ignored and the program sent to the datalogger, the datalogger may generate additional compiler errors, or it may accept the program but size the data table with fewer fields than specified in the program. It is therefore good practice to put a **FieldNames()** instruction after each output instruction that uses **ArrayIndex()**.

### 7.3.3 CWB100Routes() Instruction

Use the CRBasic instruction **CWB100Routes()** to view the routes used by the wireless sensors to transmit their data to the CWB100 base station. For each wireless sensor in the network, the instruction returns the sensor radio address, the sensor name, any routing hops to the base, and the time required (in seconds) for the last poll. This instruction does not trigger any additional communication to the sensors in the wireless sensor network, so it does not affect power consumption of the sensors.

The instruction has the following format:

```
CWB100Routes (Port,Routes)
```

The *Port* parameter is the datalogger port communicating with the CWB100. The *Routes* parameter is the string variable where the routing information will be stored. It is important to size the variable large enough to hold all of the routing information. If the *Routes* variable is an array of strings, the sensor routes will be shown in separate elements of the array, so it must be sized to the number of sensors in the network.

The format of the returned string is:

```
RadioID SensorName: Hop1; Hop2; Hop3 Time
```

The path specified by **Hop1; Hop2; Hop3** is the path from the base to the sensor (the sensor would be after **Hop3**). A hop is a sensor's link through a routing sensor back to the wireless base. The hop is specified by that sensor name. If a sensor transmits directly to the base with no hops, only its **RadioID** and **SensorName** will be returned. The route for each sensor is terminated by a carriage return and line feed.

**Example:** In a network with sensors named T5C3, I2D6, S38D, and S089, the **CWB100Routes()** instruction returns the following **Routes** array named **RoutesArray**:

```
RoutesArray(1) = 07530A4002D6 I2D6 5.1
```

```
RoutesArray(2) = 07530A400246 T5C3: I2D6 10.2
```

```
RoutesArray(3) = 041A0A400089 S089: I2D6; T5C3 17.9
```

```
RoutesArray(4) = 12530A4013BD S38D 5.1
```

This shows that sensor I2D6 and S38D hop directly to the base station. Data from T5C3 is routed (or hops) through I2D6 and from there hops to the base station. Data from S089 hops through T5C3, then I2D6, and then to the base station.

### 7.3.4 CWB100RSSI() Instruction

The CRBasic instruction **CWB100RSSI()** is used to trigger a round of RSSI (Received Signal Strength Indicator) polling of each sensor in the network. The signal strength values are returned along with other data in the next measurement cycle. If RSSI polling is never triggered, the signal strength field associated with each sensor will always return **NAN**. The instruction has the following format:

```
CWB100RSSI (Port)
```

The **Port** parameter is the datalogger port communicating with the CWB100.

Because it requires an additional poll, only use the **CWB100RSSI()** instruction occasionally to increase battery life (see Section 6.2, *Programming the Datalogger* (p. 16)). To achieve this, it can be placed in a conditional statement triggered on time and/or a flag:

```
Public RSSITest As Boolean
.
.
If RSSITest OR TimeIntoInterval(0,12,hr) Then
  CWB100RSSI (1)
  RSSITest = false
EndIf
```

The returned value (the signal strength field in the measurement array) is the signal strength from the sensor to its closest hop. It can range from **0** to **32**,

where values under **14** indicate a weak RF signal and values over **20** indicate a strong RF signal.

### 7.3.5 CWB100Diagnostics() Instruction

The **CWB100Diagnostics()** instruction returns diagnostic information from the last poll of the wireless network. It does not initiate a new poll and so does not affect battery life.

**CWB diagnostics** (CWBPort, CWSDiag)

The **CWBPort** parameter is the datalogger port communicating with the CWB100. The **CWSDiag** parameter is the array of type Long that will hold the five values returned. These five values are shown below:

CWSDiag(1) = Number of sensors discovered by the base

CWSDiag(2) = Number of sensors polled by the base

CWSDiag(3) = Number of retries performed by the base

CWSDiag(4) = Time (in seconds) to poll all sensors

CWSDiag(5) = Network health value

The network health value is a ratio of the actual poll time to the best-case poll time for the network, scaled to the difference between the worst- and best-case times. Best-case poll time is 5 seconds per hop. Worst-case poll time, which includes time for retries, is 20 seconds per hop.

#### Example

A network has 40 wireless sensors. 26 sensors use 1 hop to get the base; 14 use 2 hops. The best-case poll time is 270 seconds:

$$26 \cdot 5 \text{ seconds} + 14 \cdot 10 \text{ seconds} = 270 \text{ seconds}$$

The worst-case poll time is 1080 seconds:

$$26 \cdot 20 \text{ seconds} + 14 \cdot 40 \text{ seconds} = 1080 \text{ seconds}$$

If the actual time to poll the sensors was 288 seconds, the network health value would be 98:

$$\frac{1080 - 288}{1080 - 270} = 98$$

### 7.3.6 Field Names

Field names are generated or learned by the base and are provided to the datalogger for the purpose of generating table definitions. The configuration string is used as a starting point for field name generation. If the configuration string does not fully specify field names, default names with a serialize root name “\_Fn” are generated by the base. Later, as new sensors are discovered, the generated names are modified by field names learned from the sensor with

the “\_Fn” portion of the name replaced by the field name learned from the sensor.

For example, before discovery, a generated field name might be: “CWS650\_\_F3”, then after discovery, this may change to “CWS650\_BatV” where “BatV” was the field name provided for the third field by the sensor.

If a sensor name is not provided by a configuration string, then the sensor name learned from sensor discovery will be used to name the sensor. These auto-discovered sensor field names are added to the end of the list of field names until all available space in the Destination Array is consumed.

The default sensor field names are shown in *Wireless Sensor Planner*, *Network Planner*, and *Device Configuration Utility* software. Descriptions of the default sensor names are found in Appendix C, *Wireless Sensor Measurement Names and Meanings* (p. C-1).

## 8. Troubleshooting

### 8.1 Checking Sensor Battery Voltage

Battery voltage is returned with the data from each sensor. Recharge or replace batteries when the voltage falls below 2.25 V. The recharging battery pack provides a trickle charge and cannot recharge depleted batteries.

Battery voltage can also be the field. Press (do not hold) the **Setup** button on the back of the wireless sensor. Red and blue LEDs flash simultaneously. The blue LED indicates the battery voltage:

<b>Blue LED Response</b>	<b>Battery Voltage</b>
No flashes	No power
1 flash	< 2.25 V
2 flashes	2.25 to 2.5 V
3 flashes	2.5 to 2.75 V
4 flashes	>2.75 V

### 8.2 Checking Sensor Links to the Base Station

When the sensor has established a route to the base, that route is held in the CWB100 routing table. To access that table, use the **CWB100Routes()** instruction. See Section 7.3.3, *CWB100Routes() Instruction* (p. 24), for details.

A sensor can also be checked in the field to see if it has established a route. Press (do not hold) the **Setup** button on the back of the wireless sensor. Red and blue LEDs flash simultaneously. The red LED indicates the number of hops to the base radio as follows:

**TABLE 8-2. Red LED and Routes**

Red LED Response	Meaning
No flashes	No route to base
1 flash	Direct route to base (1 hop)
2 flashes	Route to base uses 1 repeater (2 hops)
3 flashes	Route to base uses 2 repeaters (3 hops)
4 flashes	Route to base uses 3 repeaters (4 hops)

### 8.3 Checking Sensor Signal Strength

Signal strength is measured when the datalogger executes a **CWB100RSSI()** instruction (see Section 7.3.4, *CWB100RSSI() Instruction* (p. 25)). Values under **14** indicate a weak RF signal and values over **20** indicate a strong RF signal. If the measured signal strength is strong but the poll time shown in the **CWB100Diagnostics()** response is long, check for RF interference.

## 9. Maintenance

- If batteries are solar charged, periodically clean the solar cell battery lid with water to remove dust and debris.
- Replace batteries as indicated by the battery voltage (see Section 8.1, *Checking Sensor Battery Voltage* (p. 27)).
- Keep the lens and view of the CWS200 clean. Otherwise the sensor will be measuring the temperature of the obstruction instead of the surface of interest. Clean the lens gently with a moistened cotton swab. Distilled water or alcohol works well for most dust/dirt. Salt deposits dissolve better in a weak acid solution (~0.1 molar).
- Sensors connected to the CWS900 may require regular maintenance and recalibration. See the documentation for the sensor of interest for recommended maintenance and recalibration details.

### 9.1 Removing a Wireless Sensor from the Network

If a sensor has not responded to six polls by the CWB100 and it has been more than two hours since the last communication, the CWB100 will stop polling that sensor. The wireless sensor will still appear in the final storage data file with NAN's instead of values because the sensor is no longer linked.

The only way to fully remove a wireless sensor from the network is to force the datalogger to reread the routing table from the CWB100 after the sensor is no longer being polled. Changing the size of the array or adjusting the scan interval in the datalogger program, for example, will force that reading when the program is compiled.

## 9.2 Replacing a Wireless Sensor in the Network

If a wireless sensor in the network needs to be replaced, the CWB100 treats the replacement sensor as a new sensor because it has a different radio address than the original sensor. Data from the replacement sensor will be saved to a different position in the final storage data file with field names that include the tilde (~) character.

To configure the replacement sensor, use *Network Planner* or *Device Configuration Utility*, and update the sensor configuration string. If the same field names are used as in the original sensor, datalogger support software will append a tilde (~) to each name to make it unique until the original sensor is fully removed from the routing table.

After the configuration file is sent to the datalogger, the datalogger program must be restarted for the new file to be used. This is done through File Control:

1. Select the program from the file list and click **Stop Program**.
2. Click **Run Options**.
3. In the Select Run Options window, select **Run Now** and **Preserve data if no table changed**. Also select **Run On Power-up**.
4. Click **OK**.

The datalogger will recompile the program and use the new configuration file.

## 9.3 Stopping or Loading a New Datalogger Program

If the datalogger program is stopped or a new program loaded, the CWB100 base station will still retain its routing table. The sensors will not have to initiate auto-discovery.

If the sensors are not polled by the base at least once in 4 to 12 hours, the sensors will initiate the auto-discovery process just as if the **Setup** button had been pressed to start auto-discovery.

# 10. Troubleshooting

**Symptom:** Cannot connect to the CWB100 in *Device Configuration Utility*

- Check that the correct communication port was chosen.
- Ensure that the device driver was installed.

**Symptom:** Cannot connect to a wireless sensor in *Device Configuration Utility*

- Check that the correct communication port was chosen.
- Ensure that the device driver for the A205 was installed.

**Symptom:** Datalogger program compile results indicate that the configuration file does not exist.

- Check that the configuration file was sent to the datalogger. It can be sent through *Wireless Sensor Planner*, *Network Planner*, or *File Control*.
- Check the file pathway used in the **CWSConfig** parameter of the **CWB100()** instruction.

**Symptom:** Wireless sensor LED does not respond when the **Setup** button is pressed.

- It could be due to depleted batteries. Replace or recharge.
- Connect to the wireless sensor in *Device Configuration Utility* to verify performance. *Device Configuration Utility* triggers a measurement of the wireless sensor every 5 seconds. This is a quick method for determining whether an issue with a sensor is a measurement problem or a communication problem.

**Symptom:** Pressing **Setup** on the wireless sensor for 8 to 10 seconds does not initiate auto-discovery (blue LED does not flash every second indicating search for the base station).

- Reconfigure the sensor, making sure to include the radio base station address.

**Symptom:** The first element of the destination array is **-1**, and all other elements are **0**.

- Verify that the CWB100 is connected to the control port specified in the **CWB100()** instruction.
- Verify that **+12V** and **G** terminals on CWB100 are well connected to **12V** and **G** terminals on the datalogger.
- Check that the datalogger is powered.

**Symptom:** The first element of the destination array is **-2**, and all other elements are **0**.

A device other than the CWB100 is connected to the control port specified in the **CWB100()** instruction **Port** parameter. Change the program and/or rewire to move the **DATA/A** terminal connection to the port specified.

**Symptom:** The first element of the destination array is **-3**, and all other elements are **0**.

The measurement names in the datalogger do not match the names stored in the CWB100 base station. This is a rare error message that should disappear after the datalogger has made its second attempt to communicate with the CWB100 base station. On the second attempt,

the datalogger will send the CWB100 base its version of the names and then the name signatures from both devices will match.

**Symptom:** Values in the destination array are NAN.

This indicates that the CWB100 is unable to get data from the wireless sensor.

- Verify that the scan rate in the datalogger program is longer than the minimum poll rate of 15 seconds per sensor per hop.
- Check the sensor battery voltage and replace or recharge the batteries if needed.
- Move the sensor closer to the CWB100 or add a wireless sensor to serve as a router.
- Limit RF noise from other radios.

**Symptom:** Incorrect radio address or extra characters are returned by the **CWB100Routes()** instruction.

Increase the string size of the **CWB100Routes()** destination variable or array.

**Symptom:** A “Variable Out of Bounds” error is reported in the datalogger status table.

- Check that the sensor name matches the name used in the **ArrayIndex()** instruction.
- This can also happen when the sensor has not discovered the base yet. Look at the routing table returned by the **CWB100Routes()** instruction to confirm.



# Appendix A. CRBasic Example Programs

---

Though written for CR800-series dataloggers, the following example programs can be used directly with CR1000 and CR3000 dataloggers. By changing the port used in the **CWB100()** instruction, they can be used with CR6-series dataloggers as well. The wiring table below is used in each example.

TABLE A-1. Wiring for CRBasic Examples		
CWB100 Terminal	Wire Color	CR800 Terminal
+12V	Red	12V
G	Black	G
DATA/A	Green	C1

## Configuration String File for CRBasic Examples

The following configuration string file, saved on the datalogger CPU as CWSConfig.txt, is used in each program example:

```
07530A4002D1 CWS900 4 Ts Ti BV SS, 18534B402008 CWS655 9  
VWC EC Ts Ka PA VR Ti BV SS
```

## A.1 Basic Program

A simple CRBasic program is shown below for a network consisting of a CWS655 and a CWS900 connected to a 109 Temperature Probe. Because the CWS655 returns 9 values and the CWS900 returns 4, the destination array is sized for 13 values. The configuration string file is called CWSConfig.txt and is stored in the CPU drive of the datalogger. The configuration string file is shown above.

CRBasic Example A-1. Basic Wireless Sensor Network Program
<pre>Public CWSARRAY(13)  DataTable (CWSData,True,-1)   DataInterval (0,5,Min,0)   Sample (13,CWSARRAY(),FP2) EndTable  BeginProg   Scan (5,Min,0,0)   CWB100 (1,CWSARRAY(),"CPU:CWSConfig.txt")   CallTable CWSData   NextScan EndProg</pre>

## A.2 Program with Diagnostics

The following program reports all available diagnostics for a wireless sensor network: battery voltages, routes, signal strength, and network diagnostics.

It measures a CWS655 and a CWS900 connected to a Campbell Scientific 109 Temperature Probe. Sensor measurements are made every 10 minutes, while signal strength is measured every 60 minutes. A Boolean variable called **RSSITest** is available to manually force the signal strength measurement. Sensor routes and network diagnostics are reported every scan.

CWS900\_SS and CWS655\_SS will display NAN except during the scan immediately following the **CWB100RSSI()** instruction. NANs are prevented from influencing average and minimum calculations through an expression entered in the **DisableVar** (disable variable) parameter.

### CRBasic Example A-2. Program with Diagnostics

```
Public CWSArray(13)
Public RoutesArray(2) As String *50
Public RSSITest As Boolean
Public DiagArray(5) As Long

DataTable (CWSData,True,-1)
  DataInterval (0,10,Min,10)
  Average (1,CWSArray(1),IEEE4,CWSArray(1)= NAN) '109 Temp
  Average (1,CWSArray(5),IEEE4,CWSArray(5)= NAN) 'CWS655 VWC
EndTable

DataTable (Diagnostics,True,-1)
  DataInterval (0,12,Hr,10)
  Minimum (1,CWSArray(3),FP2,CWSArray(3)= NAN,False) 'CWS900 Battery
  Minimum (1,CWSArray(4),FP2,CWSArray(4)= NAN,False) 'CWS900 Signal Strength
  Minimum (1,CWSArray(12),FP2,CWSArray(12)= NAN,False) 'CWS655 Battery
  Minimum (1,CWSArray(13),FP2,CWSArray(13)= NAN,False) 'CWS655 Signal Strength
  Sample (5,DiagArray(),Long)
EndTable

DataTable (CWSRoutes,True,-1)
  DataInterval (0,60,Min,10)
  Sample (2,RoutesArray(),String)
EndTable

'Main Program
BeginProg
  Scan (1,Min,0,0)
    'Poll sensors for measurements
    CWB100(1,CWSArray(),"CPU:CWSConfig.txt")
    'Retrieve routing table
    CWB100Routes(1,RoutesArray)
    'Check signal strength
    If RSSITest OR TimeIntoInterval(11,12,hr) Then
      CWB100RSSI (1) 'triggers separate poll, so do not do often
      RSSITest = false
    EndIf
    'Check network diagnostics
    CWB100Diagnostics (1,DiagArray)

    CallTable CWSData
    CallTable Diagnostics
    CallTable CWSRoutes
  NextScan
EndProg
```

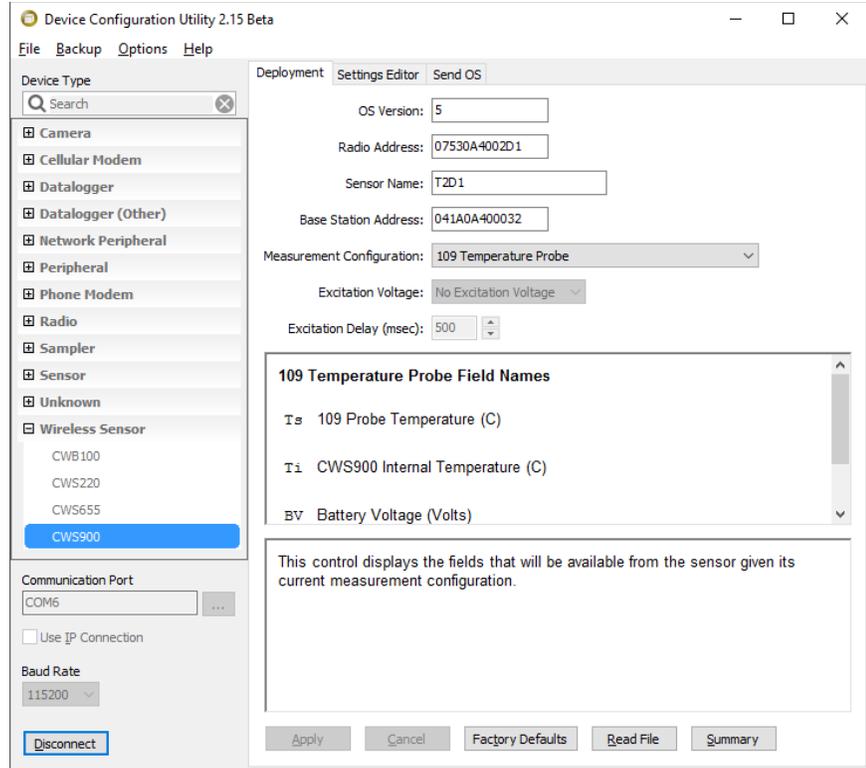
# Appendix B. Using Device Configuration Utility for Sensor Setup

Wireless sensors can be configured using *Device Configuration Utility*, which can be downloaded at [www.campbellsci.com/downloads](http://www.campbellsci.com/downloads). *Device Configuration Utility* does not allow custom field names and will not generate a configuration string, but it can be used to specify the base station radio address, the sensor name, and the measurement configuration for the wireless sensor. It can also configure a wireless sensor to connect directly to the base station or use repeaters (see Section 7.2, *RF Connection Type Setting (p. 21)*).

To connect to a sensor through *Device Configuration Utility*, first select the wireless sensor from the **Device Type** list. Then, follow the instructions shown in the window on the right.

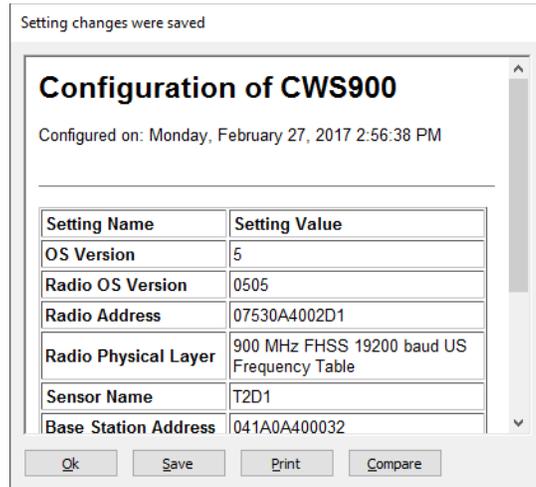


Once *Device Configuration Utility* connects to the sensor, enter the address of the CWB100 in the **Base Station Address** field. If configuring a CWS900, select the type of measurement the sensor will make from the **Measurement Configuration** list. If desired, change the **Sensor Name** from its default value.

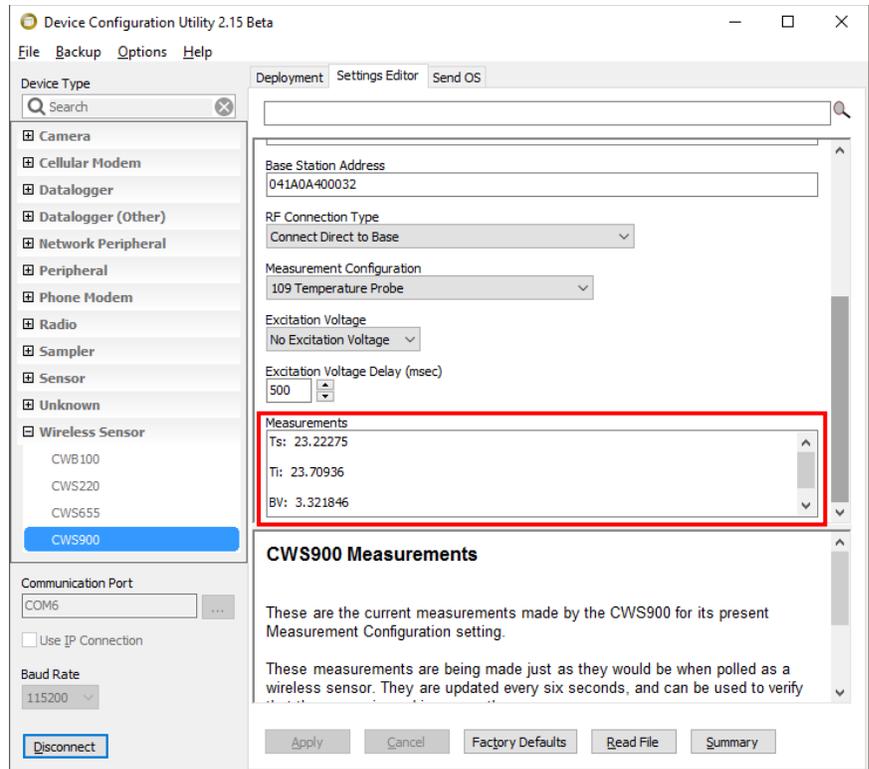


Click **Apply** to apply the changes.

A configuration summary will be displayed. Save the configuration information if desired, and then click **OK** to close the window.



*Device Configuration Utility* can also be used to check sensor measurements. While connected to the sensor, *Device Configuration Utility* triggers a measurement of the wireless sensor every 5 seconds. These measurements are shown in the **Measurements** field of the **Settings Editor** tab.





# Appendix C. Wireless Sensor Measurement Names and Meanings

---

Default measurement names are shown in *Wireless Sensor Planner*, *Network Planner*, and *Device Configuration Utility* software. Custom names may be assigned to the measurements with *Wireless Sensor Planner* or *Network Planner*. It is good practice to keep the measurement names short. The default names, meanings, and measurement units for each sensor and configuration are listed in the following sections.

## C.1 CCWS220 Measurements

Default Name	Meaning	Units
Tt	Target temperature	°C
Tb	Sensor body temperature	°C
Ti	Internal temperature of wireless sensor	°C
BV	Battery voltage of wireless sensor	Volts
SS	Signal strength of last received radio transmission	Unitless, ranging from 0 (no signal) to 32 (full signal)

## C.2 CWS655 Measurements

Default Name	Meaning	Units
VWC	Volumetric water content	m <sup>3</sup> water/m <sup>3</sup> soil
EC	Bulk electrical conductivity	dS/m
Ts	Soil temperature	°C
Ka	Apparent permittivity	unitless
PA	Period average	μsec
VR	Voltage ratio	unitless
Ti	Internal temperature of wireless sensor	°C
BV	Battery voltage of wireless sensor	Volts
SS	Signal strength of last received radio transmission	Unitless, ranging from 0 (no signal) to 32 (full signal)

### C.3 CWS900 Measurements

Depending on the configuration, the CWS900 returns 4 to 12 values. TABLE C-1 shows all configuration options and which outputs each gives. TABLE C-2 shows the default names of the outputs and provides a description for each.

TABLE C-1. CWS900 Measurement Configurations and Outputs																		
Configuration	SE1	SE2	SE3	DF	TC	IC	MF	AF	Ts	RH	Pr	LWS	DO	pH	Ti	BV	SS	Total
Single-Ended Channel 1 Measurement	✓														✓	✓	✓	4
Single-Ended Channel 1 and 2 Measurement	✓	✓													✓	✓	✓	5
Single-Ended Channel 1, 2, and 3 Measurement	✓	✓	✓												✓	✓	✓	6
Differential Channel Measurement				✓											✓	✓	✓	4
Switch Closure					✓	✓									✓	✓	✓	5
Pulse Count							✓	✓							✓	✓	✓	5
Low Level AC							✓	✓							✓	✓	✓	5
Rain Gage					✓	✓									✓	✓	✓	5
Pyranometer (or Quantum Sensor)				✓											✓	✓	✓	4
109 Temperature Probe									✓						✓	✓	✓	4
HMP60 Temp/Humidity Probe									✓	✓					✓	✓	✓	5
LWS Leaf Wetness Sensor												✓			✓	✓	✓	4
CS511 Dissolved Oxygen Sensor													✓		✓	✓	✓	4
CS525 ISFET pH Probe														✓	✓	✓	✓	4
CS450 Pressure Transducer									✓		✓				✓	✓	✓	5
All compatible wind sensors return the following 12 measurements:																		
<ul style="list-style-type: none"> <li>• WSA                      • WDA                      • RWDS</li> <li>• WSMX                    • WDSD                    • Ti</li> <li>• WSMN                    • RWS                      • BV</li> <li>• WSSD                    • RWDA                    • SS</li> </ul>																		

TABLE C-2. CWS900 Measurement Names		
Default Name	Meaning	Units
Ti	Internal temperature of wireless sensor	°C
BV	Battery voltage of wireless sensor	Volts
SS	Signal strength of last received radio transmission	Unitless, ranging from 0 (no signal) to 32 (full signal)
SE1	Single-ended DC voltage measured on pin 5 with respect to pin 3	Volts, may need additional processing to convert to engineering units
SE2	Single-ended DC voltage measured on pin 6 with respect to pin 3	Volts, may need additional processing to convert to engineering units
SE3	Single-ended DC voltage measured on pin 1 with respect to pin 3	Volts, may need additional processing to convert to engineering units
DF	Differential DC voltage measured between pin 5 (high) and pin 6 (low)	Volts, may need additional processing to convert to engineering units
TC	Total count of switch closures measured since sensor configuration	Integer, requires additional processing to convert to engineering units
IC	Number of switch closures detected during polling interval	Integer, requires additional processing to convert to engineering units
MF	Maximum frequency of pulses detected during polling interval	Hz, may need additional processing to convert to engineering units
AF	Average frequency of pulses detected during polling interval	Hz, may need additional processing to convert to engineering units
Ts	Temperature of sensor connected to CWS900	°C
RH	Relative humidity	%
LWS	Leaf wetness sensor voltage	Volts
DO	Voltage returned from CS511 Dissolved Oxygen Sensor	Volts, requires additional processing to convert to DO
pH	Voltage returned from CS525 ISFET pH Probe	Volts, requires additional processing to convert to pH
Pr	Pressure	kPa
WSA	Wind speed average during last polling interval	m/s

**TABLE C-2. CWS900 Measurement Names**

<b>Default Name</b>	<b>Meaning</b>	<b>Units</b>
WSMX	Wind speed maximum during last polling interval. Calculated from a running average of three successive, one-second wind speed measurements.	m/s
WSMN	Wind speed minimum during last polling interval. Calculated from a running average of three successive, one-second wind speed measurements.	m/s
WSSD	Wind speed standard deviation during last polling interval.	m/s
WDA	Wind direction average. Calculated over the polling interval using the average of one-second wind direction measurements, excluding zero wind speed measurements.	Degrees from north, ranging from 0° to 355°
WDSD	Wind direction standard deviation. Calculated over the polling interval using the average of one-second wind direction measurements and the Yamartino algorithm (See B3.1 for details).	Degrees
RWS	Resultant wind speed. Vector mean horizontal wind speed calculated over the polling interval using one-second measurements including measurements of zero wind speed.	m/s
RWDA	Resultant wind direction average. Calculated over the polling interval using one second wind direction measurements, excluding zero wind speed measurements.	Degrees from north, ranging from 0° to 355°
RWSD	Resultant wind direction standard deviation. Vector mean wind direction standard deviation calculated over the polling interval with the CSI wind speed weighted algorithm (See B3.1 for details).	Degrees

## C.4 Wind Vector Calculations

When a wind speed sample is 0, the CWS900 uses 0 to process scalar wind speed and wind speed standard deviation, but the sample is not used in the computation of wind direction.

### Measured raw data:

- $S_i$  = horizontal wind speed
- $\Theta_i$  = horizontal wind direction
- $U_{e_i}$  = east-west component of wind
- $U_{n_i}$  = north-south component of wind
- $N$  = number of samples

## Calculations:

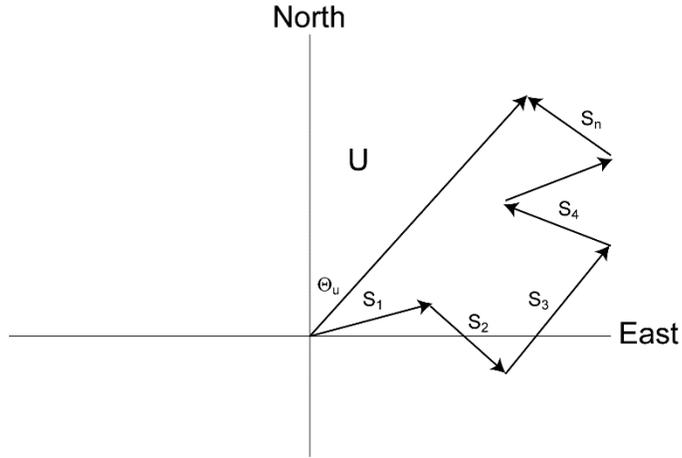


FIGURE C-1. Input Sample Vectors

In FIGURE C-1, the short, head-to-tail vectors are the input sample vectors described by  $s_i$  and  $\Theta_i$ , the sample speed and direction, or by  $U_{e_i}$  and  $U_{n_i}$ , the east and north components of the sample vector. At the end of output interval  $T$ , the sum of the sample vectors is described by a vector of magnitude  $U$  and direction  $\Theta_u$ . If the input sample interval is  $t$ , the number of samples in output interval  $T$  is  $N = T/t$ . The mean vector magnitude is  $\bar{U} = U/N$ .

### Scalar mean horizontal wind speed, $S$ :

$$S = (\sum s_i)/N$$

where in the case of orthogonal sensors:

$$S_i = (U_{e_i}^2 + U_{n_i}^2)^{1/2}$$

### Unit vector mean wind direction, $\Theta_1$ :

$$\Theta_1 = \text{Arctan}(U_x/U_y)$$

where:

$$U_x = (\sum \sin \Theta_i)/N$$

$$U_y = (\sum \cos \Theta_i)/N$$

or, in the case of orthogonal sensors:

$$U_x = (\sum (U_{e_i}/U_i))/N$$

$$U_y = (\sum (U_{n_i}/U_i))/N$$

$$\text{where } U_i = (U_{e_i}^2 + U_{n_i}^2)^{1/2}$$

**Standard deviation of wind direction,  $\sigma(\Theta_1)$** , using Yamartino algorithm:

$$\sigma(\Theta_1) = \arcsin(\varepsilon)[1+0.1547 \varepsilon^3]$$

where,

$$\varepsilon = [1-((U_x)^2+(U_y)^2)]^{1/2}$$

and  $U_x$  and  $U_y$  are as defined above.

**Resultant mean horizontal wind speed,  $\bar{U}$** :

$$\bar{U} = (U_e^2+U_n^2)^{1/2}$$

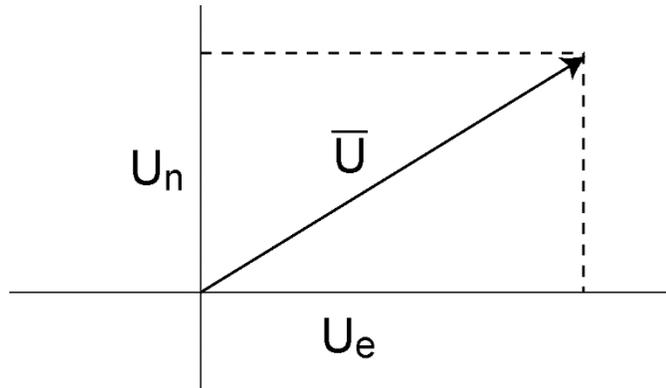


FIGURE C-2. Mean Wind Vector

where for polar sensors:

$$U_e = (\sum S_i \sin \Theta_i)/N$$

$$U_n = (\sum S_i \cos \Theta_i)/N$$

or, in the case of orthogonal sensors:

$$U_e = (\sum U_{e_i})/N$$

$$U_n = (\sum U_{n_i})/N$$

**Resultant mean wind direction,  $\Theta_u$** :

$$\Theta_u = \text{Arctan}(U_e/U_n)$$

**Standard deviation of wind direction,  $\sigma(\Theta_u)$** , using Campbell Scientific algorithm:

$$\sigma(\Theta_u) = 81(1-\bar{U}/S)^{1/2}$$

The algorithm for  $\sigma(\theta_u)$  is developed by noting (FIGURE C-3) that

$$\cos(\Theta_i') = U_i/s_i; \text{ where } \Theta_i' = \Theta_i - \Theta_u$$

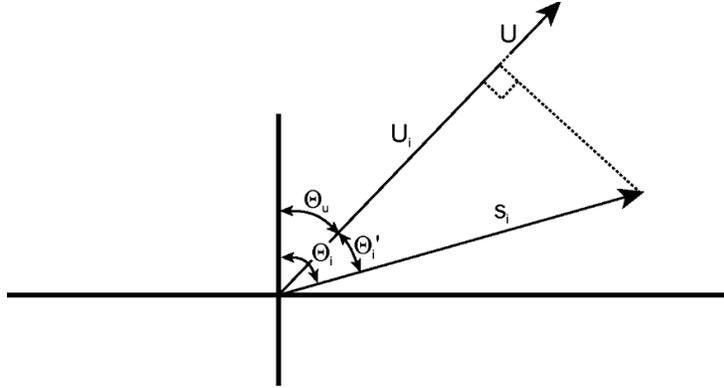


FIGURE C-3. Standard Deviation of Direction

The Taylor Series for the Cosine function, truncated after 2 terms is:

$$\text{Cos}(\Theta_i') \cong 1 - (\Theta_i')^2 / 2$$

For deviations less than 40 degrees, the error in this approximation is less than 1%. At deviations of 60 degrees, the error is 10%.

The speed sample may be expressed as the deviation about the mean speed,

$$s_i = s_i' + S$$

Equating the two expressions for Cos( $\theta'$ ) and using the previous equation for  $s_i$ ;

$$1 - (\Theta_i')^2 / 2 = U_i / (s_i' + S)$$

Solving for  $(\Theta_i')^2$ , one obtains;

$$(\Theta_i')^2 = 2 - 2U_i / S - (\Theta_i')^2 s_i' / S + 2s_i' / S$$

Summing  $(\Theta_i')^2$  over N samples and dividing by N yields the variance of  $\Theta_u$ . Note that the sum of the last term equals 0.

$$(\sigma(\Theta_u))^2 = \sum_{i=1}^N (\Theta_i')^2 / N = 2(1 - \bar{U} / S) - \sum_{i=1}^N ((\Theta_i')^2 s_i') / NS$$

The term,  $\sum((\Theta_i')^2 s_i') / NS$ , is 0 if the deviations in speed are not correlated with the deviation in direction. This assumption has been verified in tests on wind data by CSI; the Air Resources Laboratory, NOAA, Idaho Falls, ID; and MERDI, Butte, MT. In these tests, the maximum differences in

$$\sigma(\Theta_u) = (\sum(\Theta_i')^2 / N)^{1/2} \text{ and } \sigma(\Theta_u) = (2(1 - \bar{U} / S))^{1/2}$$

have never been greater than a few degrees.

The final form is arrived at by converting from radians to degrees (57.296 degrees/radian).

$$\sigma(\Theta u) = (2(1 - \bar{U}/S))^{1/2} = 81(1 - \bar{U}/S)^{1/2}$$

# Appendix D. Battery Life

---

## D.1 Battery Characteristics

The battery life of the CWS wireless sensors is a function of the polling interval, repeater usage, type of batteries, and temperature. The types of batteries that can be used in the sensors are AA sized alkaline, Lithium, NiMH, or NiCd. Some are characterized in the following table.

Battery Type	Manufacturer	Capacity (mWH)	Temperature Characteristics
Alkaline	Duracell MX1500	3800	-20 to 54 °C
Alkaline	Energizer EN91	3800	-18 to 55 °C
Lithium	Energizer L91	4000	-40 to 60 °C
NiMH	Duracell DX1500	2500	-20 to 50 °C
NiMH	Rayovac LD715	2500	-10 to 45 °C

The capacities that are listed are for 21 °C operation; for all the battery types (except Lithium), the capacity decreases as the temperature decreases. For example, the capacity of the Alkaline batteries decreases by 30% when the temperature is at 0 °C. All these batteries can be used below the stated temperature range (the Duracell Alkaline and NiMH have both been tested to -30 °C), but their capacity is reduced.

For the rechargeable batteries, while all types can be used, by far the best type to use is the “low self-discharge” NiMH batteries. Both of the NiMH batteries listed above are of this type. The normal NiMH and NiCd batteries have very high self-discharge, and would probably only last for a couple of months.

## D.2 Effect of Polling Interval and Repeaters on Battery Life

The largest factor affecting battery life is polling interval, followed by repeater usage. This is because the radio RX/TX current is the dominant drain on the batteries. Any sensor that is used as a repeater by another sensor will also have its battery life decreased, as it has to send its own data as well as data from sensors repeating through it.

---

**NOTE**

When a sensor is used as a repeater for another sensor, it is effectively polled three additional times, which will shorten its battery life.

---

Using the Duracell MX1500 Alkaline batteries, derated by ~20%, gives the following expected sensor battery life:

<b>TABLE D-1. Approximate Battery Life Expectancy</b>			
<b>Sensor and Connection</b>	<b>5 Min Polling</b>	<b>10 Min Polling</b>	<b>15 Min Polling</b>
CWS900 with 109 Temperature Probe or CWS655	15 months	21 months	24 months
Same as above, used as a repeater for 1 sensor	8 months	12 months	15 months
Same as above, used as a repeater for 2 sensors	6 months	8 months	12 months

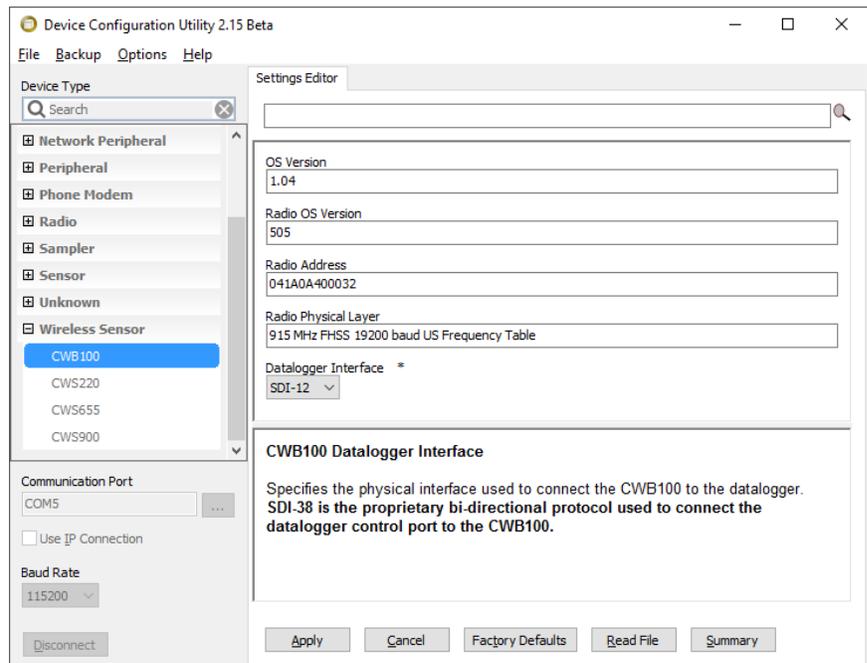
# Appendix E. Using SDI-12 with a Wireless Sensor Network

This appendix describes how to set up a network to communicate with the SDI-12 protocol. The SDI-12 protocol allows communication with dataloggers that would otherwise be incompatible, such as the CR300. For more information about the SDI-12 protocol, visit [www.sdi-12.org](http://www.sdi-12.org).

## E.1 Configure the CWB100

Connect to the CWB100 in *Device Configuration Utility*. First select the wireless sensor from the **Device Type** list. Then, follow the instructions shown in the window on the right.

In the **Datalogger Interface** setting list, select **SDI-12**.



Click **Apply**.

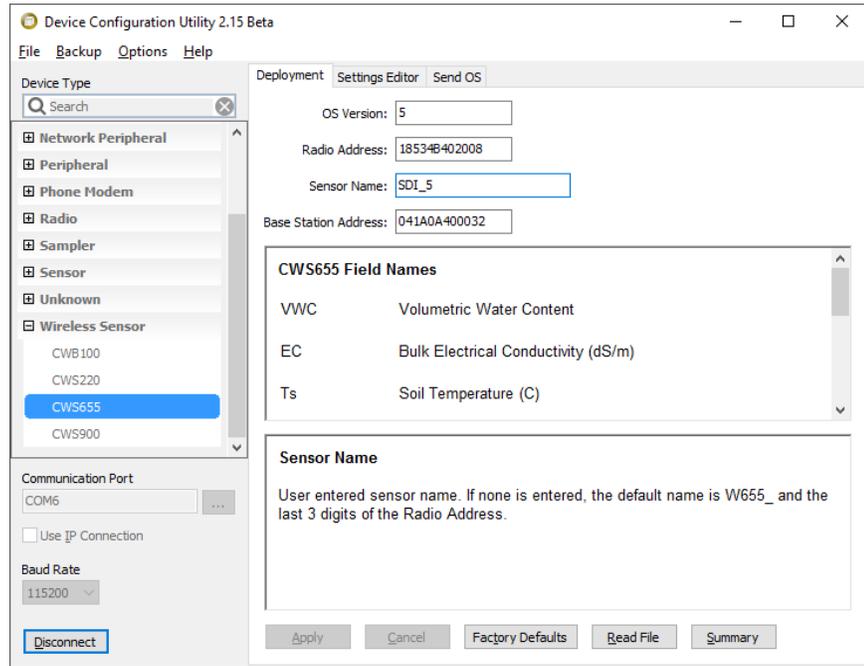
## E.2 Configure Wireless Sensors

Connect to the sensor in *Device Configuration Utility*.

Enter the **Sensor Name**. To be used in a wireless sensor network using the SDI-12 protocol, the sensor name must be entered as **SDI\_#number**, where **number** is SDI-12 address. Each SDI-12 address must be unique.

Also, enter the **Base Station Address**. This is the serial number of the CWB100, found on the CWB100 label.

Note the number of values the sensor returns, not counting RSSI.



Click **Apply**.

Repeat for each wireless sensor.

## E.3 Available SDI-12 Commands

TABLE E-1 shows all available SDI-12 commands and their responses. Information on each of these commands is given in following sections.

SDI-12 commands have three components:

*Sensor address (a)* – a single character and the first character of the command.

*Command body* – an upper-case letter (the “command”), optionally followed by one or more alphanumeric qualifiers.

*Command termination (!)* – an exclamation mark.

TABLE E-1. SDI-12 Commands

Name	Command	Response
Address Query (p. E-3)	?!	a
Acknowledge Active (p. E-3)	a!	a
Start Concurrent Measurement (p. E-4)	aC!	atttn
Send Data (p. E-4)	aD!	a<values>
Send Identification (p. E-5)	aI!	allccccccmmmmmmvvvxxxxxxxxxxxx
Start Measurement (p. E-3)	aM!	atttn
Start Measurement and Request CRC (p. E-4)	aMC!	atttn
Start Verification (p. E-5)	aV!	atttn
Get Base Radio Address (p. E-5)	aX0!	a:<address>
Get Sensor Field Names (p. E-6)	aX1!	a:<sensor name>,<field name 1>,<field name 2>,...
Get Route by Name (p. E-6)	aX2!	a:<sensor name 1>,<sensor name 2>,...
Get Route by Number (p. E-6)	aX3!	a:<address 1>,<address 2>,...

### E.3.1 Address Query Command (?!)

The response to the Address Query command (?! ) is the address of the first sensor in the CWB100 routing table. If no sensors have been discovered, it will return an empty line.

	Command	Response
<b>Format:</b>	?!	a
<b>Example:</b>	?!	2

### E.3.2 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.

	Command	Response
<b>Format:</b>	a!	a
<b>Example:</b>	2!	2

### E.3.3 Start Measurement Command (aM!)

The Start Measurement command (aM! ) triggers an exchange with the sensor to pick up measured values. The sensor will return the usual values except for the signal strength value, which requires a separate command. Up to 9 values may be returned in response to the aM! command. If the sensor returns more than 9 values, subsequent commands (aM1! through aM9! ) can be used to pick up the remaining values, with each command picking up a maximum of 9

values each. Subsequent commands only serve to pick up additional values and do not trigger an exchange with the sensor.

When the **aM!** is issued, the sensor responds with its address, the time until the data will be available, and how many values the data will include.

	<b>Command</b>	<b>Response</b>
<b>Format:</b>	aM!	atttn
<b>Example:</b>	3M!	30304

<b>Format</b>	<b>Example</b>	<b>Description</b>
a	3	Sensor SDI-12 address
ttt	030	Time until data is ready, in seconds
n	4	Number of values in data

The example response indicates that it came from sensor with the SDI-12 address **3**, the data will be ready in **030** seconds, and **4** values will be available.

After the data is available, the sensor sends a service request. To retrieve the data, use the **aD0! ... aD9!** commands.

Note that the values being returned will not include the RSSI value of the sensor. The **aV!** command is used to get RSSI.

### E.3.4 Start Measurement Commands with Cyclic Redundancy Check (**aMC!**)

The **aMC!** command follows the same pattern as the **aM!** command, but it adds a cyclic redundancy check (CRC) to the **aD0! ... aD9!** responses.

### E.3.5 Concurrent Measurement Command (**aC!**)

The Concurrent Measurement command (**aC!**) works as described in SDI-12 documentation, but because it includes the allotted time for wireless sensor retries, it is much longer than would typically be needed. Measurements are best done with the **aM!** command instead.

### E.3.6 Send Data Command (**aD0! ... aD9!**)

After receiving a service request from the **aM!** command or after waiting for a sequence of **aC!** commands, **aD0! ... aD9!** commands are used in succession to retrieve values.

	<b>Command</b>	<b>Response</b>
<b>Format:</b>	aD0!...aD9!	a<values>
<b>Example:</b>	3M! 3D0! 3D1!	30304 3+21.57494+21.63065+21.62282 3+2.770481

Format	Example	Description
a	3	Sensor SDI-12 address
<values>	+21.57494	Value 1
	+21.63065	Value 2
	+21.62282	Value 3
	+2.770481	Value 4

### E.3.7 Send Identification Command (aI!)

The response to the Send Identification command (**aI!**) is the model and version of the CWB100 base station and the radio address of the sensor.

	Command	Response
<b>Format:</b>	aI!	a11ccccccmmmmmvvxxxxxxxxxxxx
<b>Example:</b>	2I!	213Campbe11CWB10010412530A401388

Format	Example	Description
a	2	Sensor SDI-12 address
11	13	SDI-12 version number
ccccccc	Campbe11	Company name
mmmmm	CWB100	Base station model name
vvv	104	Base station operating system version
xxxxxxxxxxx	12530A401388	Sensor radio address (serial number)

### E.3.8 Start Verification Command (aV!)

The Start Verification command (**aV!**) is used to retrieve the signal strength from the sensor. It triggers an exchange with the sensor and returns one value when the **aD0!** command is issued. This follows the same pattern as the **aM!** command.

	Command	Response
<b>Format:</b>	aV!	attn
<b>Example:</b>	4V!	40301
	4D0!	4+27.00000

The above example shows a sensor addressed **4** with a signal strength of **27**.

### E.3.9 Get Base Radio Address (aX0!)

The Get Base Radio Address (**aX0!**) command is used to retrieve the radio address of the CWB100 base station.

	Command	Response
<b>Format:</b>	aX0!	a:<address>
<b>Example:</b>	2X0!	2:12530A401396

The above example shows a sensor addressed 2 responding with the address of the base station it is linked to.

### E.3.10 Get Sensor Field Names (aX1!)

The Get Sensor Field Names (aX1!) command is used to retrieve the field names for a sensor. The string cannot exceed 120 characters, so the field name list may be cut short.

	Command	Response
<b>Format:</b>	aX1!	a:<field name 1>,<field name 2>,...
<b>Example:</b>	2X1!	2:VWC,EC,Ts,Ka,PA,VR,Ti,BV

### E.3.11 Get Route by Name (aX2!)

The response to the Get Route by Name (aX2!) command is a list of sensor names through which the base hops to reach the sensor.

	Command	Response
<b>Format:</b>	aX2!	a:<sensor name 1>,<sensor name 2>,...
<b>Example:</b>	3X2!	3:SDI_2,SDI_4

In this example, the base station reaches sensor SDI\_3 by hopping through SDI\_2, then SDI-4.

### E.3.12 Get Route by Number (aX3!)

The response to the Get Route by Number (aX3!) command is a list of radio addresses through which the base hopes to reach the sensor.

	Command	Response
<b>Format:</b>	aX3!	a:<address 1>,<address 2>,...
<b>Example:</b>	4X3!	4:12530A401388

## E.4 Program the Datalogger

Using the information from Appendix E.3, *Available SDI-12 Commands* (p. E-2), you can write a program for a datalogger using the **SDI12Recorder()** CRBasic instruction. After issuing a command that requires follow-up **aD0!** ... **aD9!** commands, such as the **aM!** or **aV!** command, the datalogger waits the time specified by the sensor and then automatically issues the appropriate number of **aD!** commands to retrieve the values.

Because neither a configuration string nor the **ArrayIndex()** function can be used with the SDI-12 setup, it is best to use the **Alias** declaration to provide meaningful names for each value in the sensor arrays. The following CRBasic example programs show the **Alias** declaration in use.

**NOTE** When declaring destination arrays for measurements, remember that the measurement commands do not return signal strength. Therefore, the number of values returned is one less than the totals shown in Appendix C, *Wireless Sensor Measurement Names and Meanings* (p. C-1).

### E.4.1 Basic SDI-12 CRBasic Example Program

The following program measures a CWS655 and a CWS900 configured for a Campbell Scientific 109 Temperature Probe. Using the tables in Appendix C, *Wireless Sensor Measurement Names and Meanings* (p. C-1), and subtracting the signal strength value, the CWS900 returns 3 values, and the CWS655 returns 8. In this example, the CWS900 SDI-12 address is 8, and the CWS655 SDI-12 address is 5.

#### CRBasic Example E-1. SDI-12 Basic Program

```
Public CWS900(3)
Alias CWS900(1) = CWS900_Ts
Alias CWS900(2) = CWS900_Ti
Alias CWS900(3) = CWS900_BV

Public CWS655(8)
Alias CWS655(1) = CWS655_VWC
Alias CWS655(2) = CWS655_EC
Alias CWS655(3) = CWS655_Ts
Alias CWS655(4) = CWS655_Ka
Alias CWS655(5) = CWS655_PA
Alias CWS655(6) = CWS655_VR
Alias CWS655(7) = CWS655_Ti
Alias CWS655(8) = CWS655_BV

DataTable (CWSData,True,-1)
  DataInterval (0,5,Min,0)
  Sample (3,CWS900(),FP2)
  Sample (8,CWS655(),FP2)
EndTable

BeginProg
  Scan (5,Min,0,0)
  SDI12Recorder (CWS900(),1,8,"M!",1.0,0)
  SDI12Recorder (CWS655(),1,5,"M!",1.0,0)
  CallTable CWSData
NextScan
EndProg
```

### E.4.2 SDI-12 CRBasic Example Program with Diagnostics

The following program reports all available diagnostics for a wireless sensor network communicating with the SDI-12 protocol: battery voltages, routes, signal strength, and network diagnostics.

It measures a CWS655 and a CWS900 connected to a Campbell Scientific 109 Temperature Probe. The CWS655 SDI-12 address is 5, and the CWS900

address is 8. Sensor measurements are made every 10 minutes, while signal strength is measured every 60 minutes. A Boolean variable called **DiagTest** is available to manually force the signal strength measurement. Sensor routes and network diagnostics are reported every scan.

**CRBasic Example E-2. SDI-12 Program with Diagnostics**

```

Public CWS900(3)
Alias CWS900(1) = CWS900_Ts
Alias CWS900(2) = CWS900_Ti
Alias CWS900(3) = CWS900_BV

Public CWS655(8)
Alias CWS655(1) = CWS655_VWC
Alias CWS655(2) = CWS655_EC
Alias CWS655(3) = CWS655_Ts
Alias CWS655(4) = CWS655_Ka
Alias CWS655(5) = CWS655_PA
Alias CWS655(6) = CWS655_VR
Alias CWS655(7) = CWS655_Ti
Alias CWS655(8) = CWS655_BV

Public DiagTest As Boolean
Public CWS900_SS
Public CWS655_SS
Public CWS900_Route As String *50
Public CWS655_Route As String *50

DataTable (CWSDData,True,-1)
  DataInterval (0,5,Min,0)
  Sample (3,CWS900(),FP2)
  Sample (8,CWS655(),FP2)
EndTable

DataTable (Diagnostics, True, -1)
  DataInterval (0,12,Hr,10)
  Minimum (1,CWS900_SS,FP2,CWS900_SS=NAN,False)
  Minimum (1,CWS655_SS,FP2,CWS655_SS=NAN,False)
  Sample (1,CWS900_Route,String)
  Sample (1,CWS655_Route,String)
EndTable

BeginProg
  Scan (5,Min,0,0)
  'Poll sensors for measurements
  SDI12Recorder (CWS900(),1,8,"M!",1.0,0)
  SDI12Recorder (CWS655(),1,5,"M!",1.0,0)

  'Check RSSI and routes
  If DiagTest OR TimeIntoInterval (11,12,Hr) Then
    SDI12Recorder (CWS900_SS,1,8,"V!",1.0,0)
    SDI12Recorder (CWS655_SS,1,5,"V!",1.0,0)
    SDI12Recorder (CWS900_Route,1,8,"X2!",1.0,0)
    SDI12Recorder (CWS655_Route,1,5,"X2!",1.0,0)
    DiagTest = False
  EndIf

  CallTable CWSDData
  CallTable Diagnostics
NextScan
EndProg

```



## Campbell Scientific Companies

---

**Campbell Scientific, Inc.**

815 West 1800 North  
Logan, Utah 84321  
UNITED STATES

[www.campbellsci.com](http://www.campbellsci.com) • [info@campbellsci.com](mailto:info@campbellsci.com)

**Campbell Scientific Canada Corp.**

14532 – 131 Avenue NW  
Edmonton AB T5L 4X4  
CANADA

[www.campbellsci.ca](http://www.campbellsci.ca) • [dataloggers@campbellsci.ca](mailto:dataloggers@campbellsci.ca)

**Campbell Scientific Africa Pty. Ltd.**

PO Box 2450  
Somerset West 7129  
SOUTH AFRICA

[www.campbellsci.co.za](http://www.campbellsci.co.za) • [cleroux@csafrica.co.za](mailto:cleroux@csafrica.co.za)

**Campbell Scientific Centro Caribe S.A.**

300 N Cementerio, Edificio Breller  
Santo Domingo, Heredia 40305  
COSTA RICA

[www.campbellsci.cc](http://www.campbellsci.cc) • [info@campbellsci.cc](mailto:info@campbellsci.cc)

**Campbell Scientific Southeast Asia Co., Ltd.**

877/22 Nirvana@Work, Rama 9 Road  
Suan Luang Subdistrict, Suan Luang District  
Bangkok 10250  
THAILAND

[www.campbellsci.asia](http://www.campbellsci.asia) • [info@campbellsci.asia](mailto:info@campbellsci.asia)

**Campbell Scientific Ltd.**

Campbell Park  
80 Hathern Road  
Shepshed, Loughborough LE12 9GX  
UNITED KINGDOM

[www.campbellsci.co.uk](http://www.campbellsci.co.uk) • [sales@campbellsci.co.uk](mailto:sales@campbellsci.co.uk)

**Campbell Scientific Australia Pty. Ltd.**

PO Box 8108  
Garbutt Post Shop QLD 4814  
AUSTRALIA

[www.campbellsci.com.au](http://www.campbellsci.com.au) • [info@campbellsci.com.au](mailto:info@campbellsci.com.au)

**Campbell Scientific Ltd.**

3 Avenue de la Division Leclerc  
92160 ANTONY  
FRANCE

[www.campbellsci.fr](http://www.campbellsci.fr) • [info@campbellsci.fr](mailto:info@campbellsci.fr)

**Campbell Scientific (Beijing) Co., Ltd.**

8B16, Floor 8 Tower B, Hanwei Plaza  
7 Guanghua Road  
Chaoyang, Beijing 100004  
P.R. CHINA

[www.campbellsci.com](http://www.campbellsci.com) • [info@campbellsci.com.cn](mailto:info@campbellsci.com.cn)

**Campbell Scientific Ltd.**

Fahrenheitstraße 13  
28359 Bremen  
GERMANY

[www.campbellsci.de](http://www.campbellsci.de) • [info@campbellsci.de](mailto:info@campbellsci.de)

**Campbell Scientific do Brasil Ltda.**

Rua Apinagés, nbr. 2018 – Perdizes  
CEP: 01258-00 – São Paulo – SP  
BRASIL

[www.campbellsci.com.br](http://www.campbellsci.com.br) • [vendas@campbellsci.com.br](mailto:vendas@campbellsci.com.br)

**Campbell Scientific Spain, S. L.**

Avda. Pompeu Fabra 7-9, local 1  
08024 Barcelona  
SPAIN

[www.campbellsci.es](http://www.campbellsci.es) • [info@campbellsci.es](mailto:info@campbellsci.es)

Please visit [www.campbellsci.com](http://www.campbellsci.com) to obtain contact information for your local US or international representative.