PRODUCT MANUAL

Measurement and Control Peripheral

AVW200 Series

2-Channel Vibrating-Wire Analyzer Modules

Revision: 02/2025

RELIABLE MONITORING SINCE 1974

Campbell SCIENTIFIC®

Please read first

About this manual

Please note that this manual was produced by Campbell Scientific Inc. primarily for the North American market. Some spellings, weights and measures may reflect this. In addition, while most of the information in the manual is correct for all countries, certain information is specific to the North American market and so may not be applicable to European users. Differences include the U.S. standard external power supply details where some information (for example the AC transformer input voltage) will not be applicable for British/European use. Please note, however, *that when a power supply adapter is ordered from Campbell Scientific it will be suitable for use in your country*.

Reference to some radio transmitters, digital cell phones and aerials (antennas) may also not be applicable according to your locality. Some brackets, shields and enclosure options, including wiring, are not sold as standard items in the European market; in some cases alternatives are offered.

Recycling information for countries subject to WEEE regulations 2012/19/EU

At the end of this product's life it should not be put in commercial or domestic refuse but sent for recycling. Any batteries contained within the product or used during the products life should be removed from the product and also be sent to an appropriate recycling facility, per The Waste Electrical and Electronic Equipment (WEEE) Regulations 2012/19/EU. Campbell Scientific can advise on the recycling of the equipment and in some cases arrange collection and the correct disposal of it, although charges may apply for some items or territories. For further support, please contact Campbell Scientific, or your local agent.

Table of contents

1. Introduction

The AVW200-series consist of an AVW200 base model and one of three wireless models: AVW206, AVW211, or AVW216 The wireless models connect the AVW200 with a spread spectrum radio. The different model numbers of the wireless versions are for different spread spectrum frequency ranges.

	Compatible radios
• AVW206—910 to 918 MHz (US/Canada)	RF401A
• AVW211—920 to 928 MHz (Australia/Israel)	RF411A
• AVW216—2.450 to 2.482 GHz (worldwide)	RF416

CAUTION:
Products using the 24XStream radio, including the AVW216, have not been available for sale in Europe since 1/1/2015 due to changes in EU legislation. Consequently, purchase of the AVW216 is not recommended for new installations in Europe.

Throughout this manual, AVW200 will refer to all models unless specified otherwise. Likewise, AVW206 typically refers to all wireless models, and RF401A refers to the corresponding spread spectrum radio.

2. Precautions

- READ AND UNDERSTAND the Safety section at the back of this manual.
- The wireless models (AVW206, AVW211, or AVW216) generate, use, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. See Part 15 FCC compliance warning (p. 81), for more information.
- Ensure maximum protection against surges. Use coaxial (antenna) surge protection. Keep RS-232, CS I/O, and USB connections short or use protective isolation and surge protection when appropriate.

- Where an AC adapter is used, Campbell Scientific recommends pn 15966. Any other AC adapter used must have a DC output not exceeding 16.5 volts measured without a load to avoid damage to the radio. Over-voltage damage is not covered by factory warranty.
- Campbell Scientific does not recommend using RF401A-series, RF401-series, or RF430-series radios in networks containing RF450 radios. The RF450 radios will interfere with the transmission of the RF401A-series, RF401-series, and RF430-series radios.

3. Initial inspection

The AVW200 package includes the following:

- One serial data cable (9-pin socket [female] to 9-pin plug [male])
- Four screws (#6-32 x 0.375" Pan Phillips)
- Four grommets for #6 or #8 screws
- One power cable (AVW200 to data logger)

Upon receiving the AVW200, inspect the packaging and contents for any damage. If damage is found, file a claim with the shipping company and contact Campbell Scientific to arrange for repair or replacement. For assistance, refer to the back page of the manual for a list of regional offices or visit www.campbellsci.com/contact to determine which Campbell Scientific office serves your country.

Immediately verify the package contents against the shipping documentation. Carefully inspect all packaging materials to ensure no product is trapped inside. If there are any discrepancies, contact Campbell Scientific promptly. Model numbers are located on each product, and for cables, they are often found near the connection end.

4. Overview

The AVW200 module supports measurements from vibrating-wire strain gauges, pressure transducers, piezometers, tiltmeters, crackmeters, and load cells. These sensors are widely used in structural, hydrological, and geotechnical applications due to their stability, accuracy, and durability. The AVW200 can accommodate up to two vibrating-wire transducers. More sensors can be measured by using multiplexers (see Multiplexer use [p. 7]).

4.1 Design features

Historically, vibrating-wire sensors faced a significant challenge: external noise. The AVW200 effectively reduces and, in most cases, eliminates incorrect readings caused by noise. This breakthrough was achieved through advancements in technology and mathematical processing, enabling frequency-based measurements—a complete departure from the earlier time-domain measurement approach. Refer to Measurements (p. 3) for more details on vibrating-wire measurements.

Updated firmware for the AVW200 has been developed to simplify programming requirements. Parameters previously required, such as the number of steps, cycles, and the duration of the swept frequency, are now managed internally by the AVW200 operating system, eliminating the need for user input. Users now only need to specify the start and end frequencies and the sensor's excitation voltage. Refer to Programming with the AVW200() instruction (p. 44) for more detailed programming information.

The AVW200 returns five or six values per measurement. The first value is the vibrating-wire frequency in Hz. The sixth value is the optional thermistor measurement in ohms (Ω). Values two through five are diagnostic information giving an indication or validation of the measurement.

(1) = Frequency (Hz)

(2) = SignalStrength (mV_RMS)

(3) = Signal-to-noise ratio (unitless)

(4) = Noise frequency (Hz)

(5) = DecayRatio (unitless)

(6) = Thermistor output (Ω of resistance; see Temperature [p. 7])

On board diagnostics can be monitored to identify issues such as faulty wiring, faulty sensors, incorrect frequency range, or sensor degradation over time. See Time series and Spectrum graph information (p. 92) for information on how to use the on-board diagnostics.

4.2 Measurements

4.2.1 Vibrating-wire interface

The spectral approach implemented by the AVW200 offers significantly improved noise immunity when compared to older period-averaging techniques implemented by other vibrating-wire interfaces, such as the AVW1, AVW4, and AVW100. Testing revealed more than two to three orders of magnitude better noise immunity with the AVW200. In addition, the spectral analysis gives improved frequency resolution (0.001 Hz RMS) during quiet conditions.

The AVW200 measures the resonant frequency of the taut wire in a vibrating-wire sensor (see Figure 4-1 [p. 4]) with the following procedure: First, the AVW200 excites the wire with a swept-frequency excitation signal. Next, the AVW200 records the response from the vibrating wire. Finally, the AVW200 Fourier transforms the recorded response and analyzes the resulting spectrum to determine wire resonant frequency. This analysis also provides diagnostic information indicating the quality of the resonant-frequency measurement.

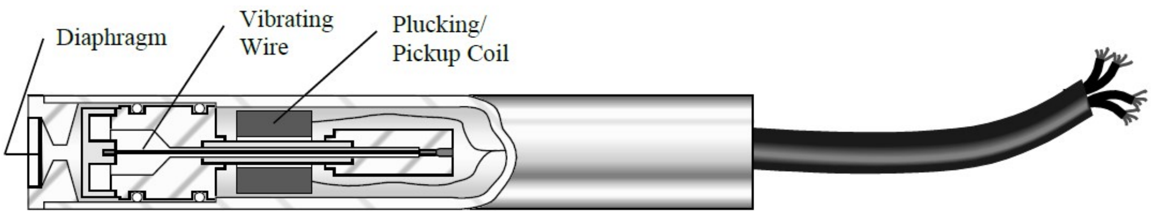


Figure 4-1. Cutaway of vibrating-wire sensor

The AVW200 measurement process involves three user-determined inputs and five outputs. The input parameters control the excitation frequency range (`BeginFreq` and `EndFreq`) and the excitation amplitude (`ExVolt`); see Table 4-1 (p. 4). The supported frequency range spans from 100 to 6500 Hz.

Table 4-1: AVW200 measurement inputs

Input	Unit	Description
BeginFreq	Hz	Minimum excitation and analysis frequency
EndFreq	Hz	Maximum excitation and analysis frequency
ExVolt	unitless	Excitation voltage 1: 5 VDC peak to peak 2: 12 VDC peak to peak
Therm50_60Hz (see Temperature [p. 7])	unitless	Thermistor measurement parameter 0: No thermistor measurement _60Hz Use 60 Hz noise rejection _50Hz: Use 50 Hz noise rejection

The measurement outputs are resonant frequency, response amplitude, signal-to-noise ratio, noise frequency, decay ratio, and thermistor resistance; see Table 4-2 (p. 5). The raw frequency measurement output of the AVW200 is in Hz, which differs from previous Campbell Scientific interface outputs in kHz^2 or $1/T^2$, where *T* is the period in milliseconds. The Hz output is converted to the appropriate units of measurement (such as pressure, strain, or displacement) by using information provided on the sensor calibration report. Refer to Displacement example (p. 69) and Wireless/one sensor/resistance converted to temperature (p. 55) for examples of converting Hz to displacement.

Table 4-2: AVW200 measurement outputs

Output	Unit	Description
Resonant frequency	Hz	Frequency of peak response
Response amplitude	mV RMS	Amplitude of peak response[1]
Signal-to-noise ratio	unitless	Response amplitude divided by amplitude of largest noise candidate[1]
Noise frequency	Hz	Frequency of largest noise candidate[1]
Decay ratio	unitless	Ending time-series amplitude divided by beginning time-series amplitude[1]
Thermistor resistance (see Temperature [p. 7])	ohms	On-gauge thermistor resistance[2]

[1] Use for measurement diagnostics.

[2] Optional output; not measured if Therm50_60Hz is set to 0.

When using firmware version Std.04 or higher and with the response amplitude diagnostic measuring as <0.01 mV RMS (10 microvolts), the resonant frequency reading will be modified to warn the user about the occurrence of low signal strength amplitudes. If SDI-12 is used to communicate with the AVW200-series device, the frequency will be given as –9,999,999 under those conditions. For all other communications methods, the frequency will be given as **NAN** (invalid) during this low signal strength condition.

If you desire the frequency to be returned as **NAN** for a higher or more pessimistic threshold than 0.01 mV, then use an optional parameter in the AVW200 CRBasic instruction (see Programming with the AVW200() instruction [p. 44] for details).

The resonant frequency reading is also used to warn when an invalid voltage supply is in the hardware of the device (firmware std.04 and higher). If an internal calibration factor is outside the expected range, then a –555,555 value is returned for the resonant frequency measurement. This indicates a hardware issue on the device that requires a factory examination and/or repair.

Figure 4-2 (p. 6) is a representative output from ***Device Configuration Utility*** troubleshooter illustrating the AVW200 measurements. The upper graph shows the spectrum after the AVW200 has applied the fast Fourier transform (FFT) algorithm. In addition to resonant frequency, the spectrum shows response amplitude, noise amplitude, and noise frequency. The lower graph shows the raw time series data recorded from a vibrating-wire sensor after the sensor has been excited with the swept-frequency voltage signal. The AVW200 computes the signal-to-noise ratio diagnostic by dividing the response amplitude by the noise amplitude. The AVW200 computes the decay ratio diagnostic from the time series ending amplitude divided by the beginning amplitude, as shown in the bottom graph in Figure 4-2 (p. 6).

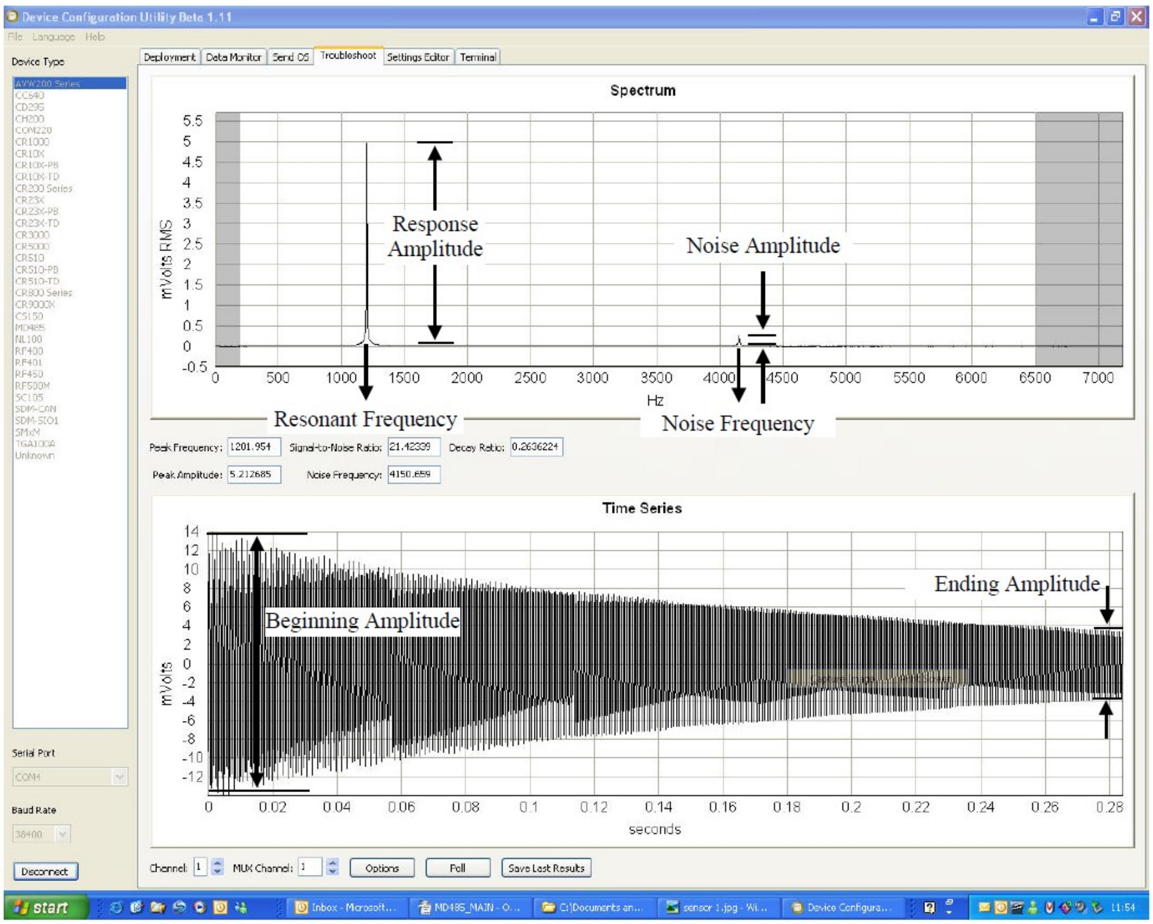


Figure 4-2. Device Configuration Utility plots showing the AVW200 measurement approach

Using the special FFT algorithm to achieve better noise immunity does require time for computation, which limits the maximum vibrating-wire measurement rate to 2 seconds per sensor. Running a program at rates faster than this will result in compile/download errors. (See Troubleshoot [p. 39], Programming with the AVW200() instruction [p. 44], and Time series and Spectrum graph information [p. 92].)

4.2.2 Temperature

The AVW200 contains a completion resistor for measuring the internal thermistor contained in many vibrating-wire sensors. Thermistor resistance changes with the internal temperature of the sensor. This temperature can be used to correct errors in the vibrating-wire measurement due to thermal expansion/contraction of the sensor body. The temperature correction is often used when the temperature of the measured medium is changing, such as water temperature in a river or shallow lake. Temperature is calculated by applying the resistance to a known equation such as the Steinhart-Hart equation. Sensor-specific Steinhart-Hart coefficients are found in the sensor manual. Refer to Wireless/one sensor/resistance converted to temperature (p. 55) for an example program and to Thermistor information (p. 71) for more details.

4.2.3 Multiplexer use

For AVW200 interfaces, the AM16/32B multiplexer is recommended over its predecessors (for example, the AM16/32 or AM16/32A). The AM16/32B has a clocking mode that can use relay addressing to go directly to a specific channel, thereby reducing power consumption and wear on relay switches.

Up to 32 vibrating-wire sensors without thermistors or 16 vibrating-wire sensors with thermistors can be connected to one multiplexer. Two multiplexers can be connected to one AVW200. Using a direct RS-232 connection, up to four AVW200 interfaces can be connected to one CR1000X Series, CR6, or Granite 6 data logger. This allows up to 256 vibrating-wire sensors (128 with thermistors) to be measured by one of these data loggers (see Figure 4-3 [p. 8]).

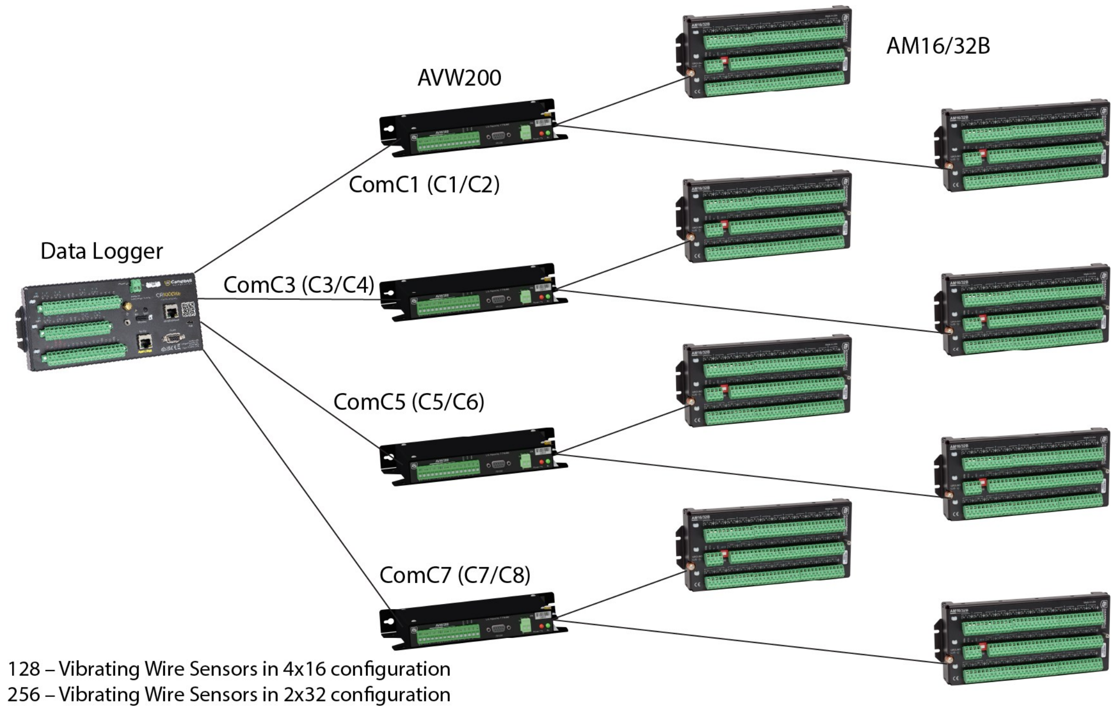


Figure 4-3. Network of AVW200s and AM16/32Bs (using a direct RS-232 connection)

Figure 4-4 (p. 9). and Figure 4-5 (p. 10) illustrate using multiple multiplexers with wireless AVW206s or the SDI-12 protocol. Refer to Multiplexers controlled by AVW200 (p. 19), Multiplexers controlled by data logger (p. 22), Multiplexer wiring (p. 29), Programming with the AVW200() instruction (p. 44), and Program examples (p. 52).

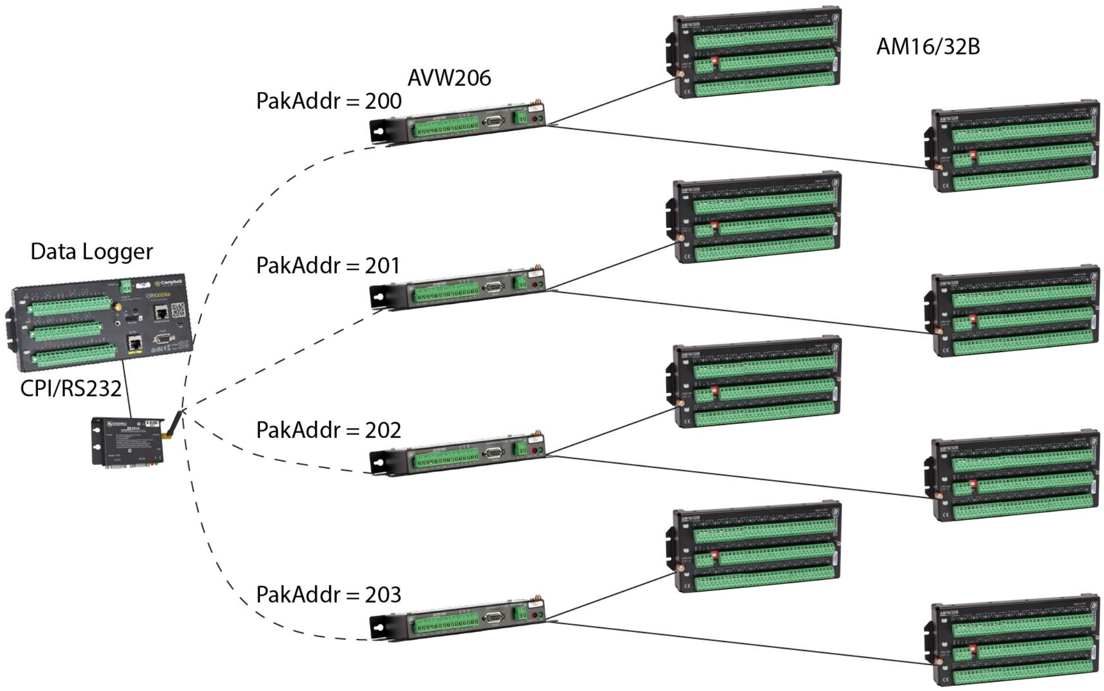


Figure 4-4. Network of AVW206s and AM16/32Bs (wireless)

> **NOTE:**
> Figure 4-4 (p. 9) shows four AVW206s, but a wireless network can support more. The total number of supported devices depends on network topology, bandwidth and data rate, as well as power and range limitations.

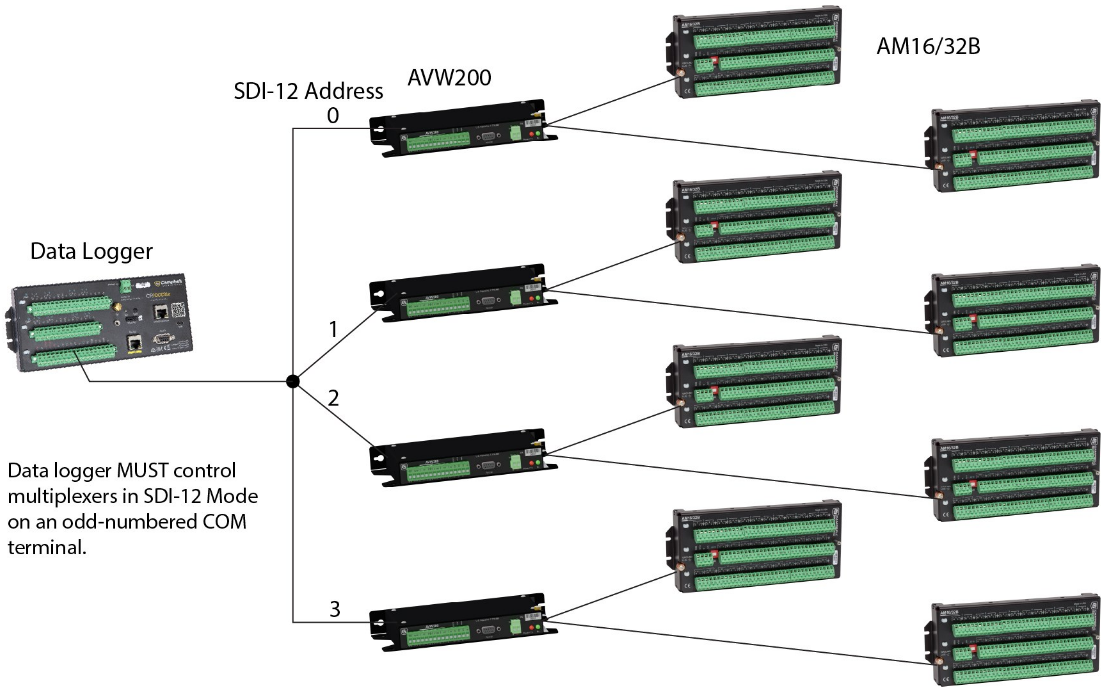


Figure 4-5. Network of AVW200 interfaces (SDI-12)

> **NOTE:**
> Figure 4-5 (p. 10) shows four AVW200s, but a SDI-12 network can support more. If more than one SDI-12 sensor is measured on the same port, each sensor requires a unique SDI-12 address.

4.3 Device configuration and programming software

The AVW200 is configured with the***Device Configuration Utility***, which supports configuration, operating system download, and vibrating-wire spectrum analysis and troubleshooting (see Device Configuration Utility [p. 32]). ***Device Configuration Utility*** is included in ***PC400*** and ***LoggerNet***, so no additional download of ***Device Configuration Utility*** is required if you are using either of these programs. Alternatively, the ***Device Configuration Utility*** is available as a free download from the Campbell Scientific website: www.campbellsci.com/downloads.

PC400 is Campbell Scientific's free entry-level data logger support software. This software supports a variety of telecommunication options, manual data collection, and data display. ***PC400*** includes an easy-to-use program generator, ***Short Cut***, as well as a full-featured program editor, ***CRBasic Editor***. ***PC400*** is available as a free download from the Campbell Scientific website: www.campbellsci.com/downloads.

LoggerNet support software is used for data logger programming and monitoring public data tables and device status information. It supports programming, communication, and data retrieval between data loggers and a PC. For more information, refer to Programming with the AVW200() instruction (p. 44) and the ***LoggerNet*** manual s.campbellsci.com/documents/us/manuals/loggernet.pdf.

4.4 Data logger to AVW200 module communications

The AVW200 module is designed to work seamlessly with Campbell Scientific data loggers using the `AVW200()` instruction in CRBasic. The AVW200 module can communicate with data acquisition products from other manufacturers using the SDI-12 communications protocol.

> **NOTE:**
> The AVW200 does not support PakBus Encryption Keys. If a data logger has a Pakbus Encryption key, you must either remove the encryption key from the data logger settings or include the `EncryptExempt()` instruction in the data logger ***CRBasic*** program.

The following types of communication networks are supported between a data logger and an AVW200 module:

- **Direct connection over RS-232** – For this simple configuration, the AVW200 can be used straight from the box without changing settings in ***Device Configuration Utility*** – Sensor(s) are attached directly to the AVW200, which is connected to the data logger with a DB9 pin-to-pigtail serial cable or null modem cable. Refer to Direct RS-232 connection (p. 15) for more information.
- **Wireless connection** – Sensor(s) are attached directly to the AVW206. The AVW206 interface transmits data to an RF401A spread spectrum radio that is connected to the data logger. See Wireless connection (p. 1) for details.
- **SDI-12** – Campbell Scientific CRBasic data loggers use the `SDI12Recorder()` instruction for SDI-12 measurements and communications. For non-Campbell Scientific data loggers, SDI-12 is the only available option. Refer to SDI-12 measurements (p. 1) and SDI-12 example (p. 63) for more information.
- **MD485 multidrop modems** – MD485 multidrop modems can extend the distance between the AVW200 interfaces and are ideal for situations where wireless communications are impractical. See MD485 multidrop modems used with AVW200 interfaces (p. 110) for more information.

Multiplexers can be used in any of the above network types to increase the number of sensors measured by an AVW200 module. Depending of the configuration, the multiplexer is controlled either by the AVW200 (see Multiplexers controlled by AVW200 [p. 19]) or by the data logger (See Multiplexers controlled by data logger [p. 22]).

5. Specifications

Electrical specifications are valid over a –25 to 50°C range unless otherwise specified; non-condensing environment required.

5.1 Analog inputs/outputs

Description: 2 differential (DF) Vibrating Wire measurements (V+ and V–) and 2 single-ended (SE) ratiometric resistive half-bridge measurements (T+ and T–)

Vibrating wire (V+ and V–)

Range, resolution, and accuracy: 24-bit basic resolution

Input range	Measurement resolution (–50 to 85°C)	Accuracy basic (–50 to 85°C)
±250 (mV) DF	0.001 (Hz RMS)	±0.013% of reading

Input resistance: 4.75 kOhms for the Vibrating Wire Measurement inputs (V+ and V–)

Vibrating wire measurement: Differential Coil+ (V+) and Coil– (V–) outputs/inputs for direct connection excite and resonant frequency measure of vibrating wire transducers. ±2.5 V (5 V peak-to-peak) or ±6 V (12 V peak-to-peak), logarithmic sine wave frequency excitation programmable from 100 Hz to 6.5 kHz, followed by frequency domain measurements via digital signal processing for excellent noise rejection.

Resistive thermistor (T+ and T–)

Range, resolution, and accuracy: 24-bit basic resolution

Input range	Measurement resolution (–50 to 85°C)	Accuracy basic (–50 to 85°C)
±2500 (mV) SE	0.001 (Ohms RMS)	±0.25% of reading)[1]

[1] Thermistor interchangeability, resistance of the wire and thermistor linearization errors should also be considered

Input resistance: 5 kOhms for the thermistor input T– (5 kOhm 0.1% completion resistor)

Thermistor measurement: A half-bridge ratiometric measurement. The value returned is in Ohms. This can be used for temperature correction of the vibrating wire measurement.

Common mode range: ±25 V

Sustained input voltage w/o damage: ±16 Vdc max.

Measurement speed: The AVW200 Vibrating Wire measurement (DF measurement) and the Half Bridge thermistor measurement (SE measurement) combined take less than 2 seconds per measurement. The DF measurement time depends on the beginning and ending frequency range selected and will take between 1.4 to 1.6 seconds. The Half Bridge thermistor measurement (SE) takes 60 milliseconds or 70 milliseconds depending on the integration time selected. The thermistor measurement integrates for 20 milliseconds (50 Hz) or 16.66 milliseconds (60 Hz) with a positive excite and then 20 milliseconds or 16.66 milliseconds with a negative excite.

5.2 Digital control ports

Description: 3 digital control ports (C1 - C3). C1 functions as an SDI-12 I/O communication port. C2 functions as a Clk output for Mux control C3 functions as a Reset output for Mux Control.

Input state: high 2.5 to 5.3 V; iow -0.3 to 1.0 V

Input hysteresis: 1.32 V

Input resistance: 100kOhms

Output voltages (no load): high 5.0 V ±0.1 V; low <0.1

Output resistance: 330 Ohms

5.3 Communication

RS-232:	Non Isolated
Baud rates:	Selectable from 1200 to 38.4 kbps. ASCII protocol is one start bit, one stop bit, eight data bits, and no parity.
SDI-12:	Control Ports 1 is configured forSDI-12 Sensor asynchronous communication. Meets SDI-12 Standard version 1.3

5.4 System

Program execution interval:	1 second
Processor:	Hitachi H8S 2324 (16-bit CPU with 32 bit internal core)
Memory:	Either 128 or 512 kbytes of SRAM; 2 Mbyte of OS Flash
Clock accuracy:	±10 minute per month. The clock is not compensated over temperature.

> NOTE:
> The AVW200-series module synchronizes with the data logger clock with every execution of the `AVW200()` CRBasic instruction.

5.5 CE compliance

Standard(s) to which conformity is declared:	IEC61326:2002

5.6 Compliance documentation

Links to EU and UK compliance documents are found at the bottom of the product web page:	www.campbellsci.com/avw200

5.7 Power requirements

Voltage:	9.6 to 16 Vdc
Typical current drain @ 12 Vdc	
Quiescent (no radio or radio off):	~0.3 mA Radio duty cycling Vz second: ~5 mA (includes quiescent current) Radio duty cycling 1 second: ~3 mA (includes quiescent current) Radio duty cycling 8 second: ~0.75 mA (includes quiescent current) Radio always on: ~26 mA (radio transmit current 100 mA)
Active RS-232 communication:	~6 mA (3 seconds after communication stops the current will drop to the quiescent current) Measurement: ~25 mA (averaged over the 2 seconds measurement)

5.8 Physical specifications

Size:	21.6 x 11.18 x 3.18 cm (8.5 x 4.4 x 1.25 in)
Weight:	0.43 kg (0.95 lb)

6. QuickStart guides

The AVW200 can be used in many types of systems, from simple to complex. The following quick start guides provide steps used to set up a system for some example configurations.

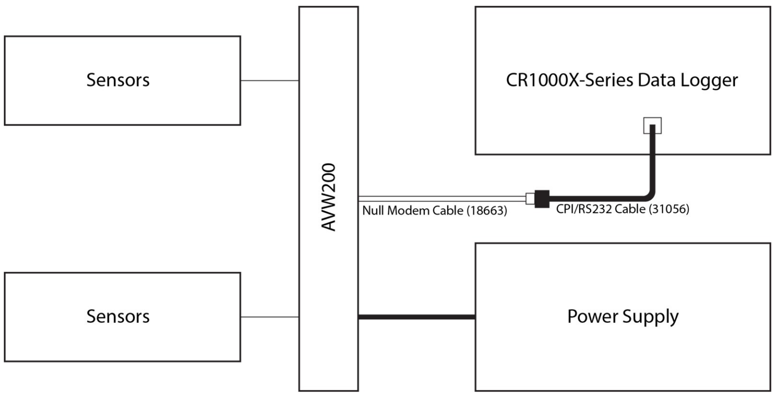
> NOTE:
> In all systems, check specific manufacturer specifications for the sensor frequency and excitation range before picking the begin and end frequencies and excitation voltage.

6.1 One or two sensors (no multiplexers)

6.1.1 Direct RS-232 connection

For this simple configuration, the AVW200 can be used straight from the box without changing settings in ***Device Configuration Utility***. Sensor(s) are attached directly to the AVW200, which is connected directly to the data logger via a DB9 pin-to-pigtail serial cable or null modem cable. The serial cable pigtail end connects to data logger control port pairs (**C1**/**C2** through **C7**/**C8**). The null modem cable has a DB9 connector to attach to the data logger **RS-232** port.

> **NOTE:**
> The CR1000X-Series features an RS-232/CPI port that requires a 31056 cable (with a female DB9 socket) to connect the data logger's RS-232/CPI port to the 18633 null modem cable.



The following steps are used to measure the sensor(s):

1. Attach the vibrating-wire sensor(s) to the AVW200 as shown in Figure 7-1 (p. 25).
2. Use the serial cable to attach the AVW200 to a control port pair (such as **C1**/**C2**, **C3**/**C4**, **C5**/**C6**, or **C7**/**C8**) on the data logger, or use the null modem cable to attach the AVW200 to the **RS-232** port on the data logger.
3. Connect one end of the power cable to the **12V** and **G** terminals on the AVW200; connect the other end to the **12V** and **G** terminals on the data logger or external power supply. See Power and ground (p. 25) and Data logger wiring for a direct connection (p. 26).
4. Create a CRBasic program that includes an `AVW200()` instruction for each sensor.

   For example, the following `AVW200()` instructions can be used to measure two sensors:

```
AVW200(Result,ComC1,200,200,Dst(1,1),1,1,1,1000,3500,2,_60HZ,1,0)
AVW200(Result,ComC1,200,200,Dst(2,1),2,1,1,1000,3500,2,_60HZ,1,0)
```

where:

AVW200 connects to data logger control ports **C1**/**C2** via a serial cable (option **ComC1**)

Begin frequency = 1000

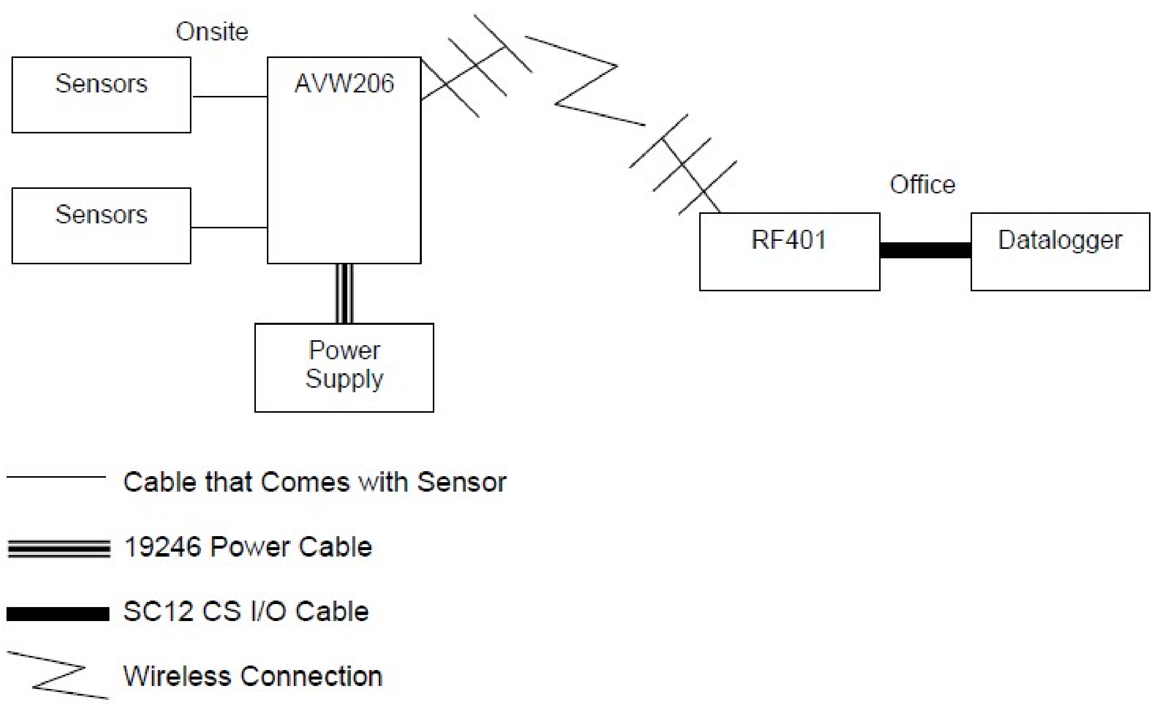
End frequency = 3500

Excitation voltage = 12 VDC peak to peak (option 2)

Refer to Programming with the AVW200() instruction (p. 44) for a thorough description of the `AVW200()` instruction and its parameters. Refer to Direct RS-232 connection with two sensors (p. 53) for a complete example program that measures two vibrating-wire sensors with no multiplexer.

6.1.2 Wireless connection

For this configuration, sensor(s) are attached directly to the AVW206. The AVW206 interface transmits data to an RF401A spread spectrum radio that is connected to the data logger.



At the AVW206 site, do the following:

1. Configure the AVW206 for RF communications as described in Connecting to Device Configuration Utility (p. 33) and Communications (p. 34).
2. Attach the vibrating-wire sensor(s) to the AVW206 as shown in Figure 7-1 (p. 25).
3. Connect a Yagi or omnidirectional antenna to the antenna connector on the side of the AVW206. (Refer to Antennas, antenna cables, and surge protectors for the AVW206, AVW211, and AVW216 [p. 76] for more detail.)
4. Use the power cable to connect the **12V** and **G** terminals on the AVW206 to the **12V** and **G** terminals on the PS200 or another power supply.

At the data logger/RF401A site, do the following:

1. Configure the RF401A radio so its parameters match the AVW206.

   > NOTE:
   > The protocol setting for the RF401A must be **PB Aware** or **PB Node**.

2. Attach a Yagi or omnidirectional antenna to the antenna connector on the RF401A.
3. Use an SC12 serial cable to attach the data logger **CS I/O** port to the RF401A **CS I/O** port. The data logger **CS I/O** port powers the RF401A. Refer to the spread spectrum data radio/modem manual for more information: https://s.campbellsci.com/documents/us/manuals/rf401a-series.pdf.
4. Create a CRBasic program that includes an `AVW200()` instruction for each sensor.

   For example, the following `AVW200()` instructions can be used to measure two sensors:

```
AVW200(Result,ComSDC7,200,200,Dst(1,1),1,1,1,1000,3500,2,_60HZ,1,0)
AVW200(Result,ComSDC7,200,200,Dst(2,1),2,1,1,1000,3500,2,_60HZ,1,0)
```

   where:

   RF401A = configured for SDC7

   Begin frequency = 1000

   End frequency = 3500

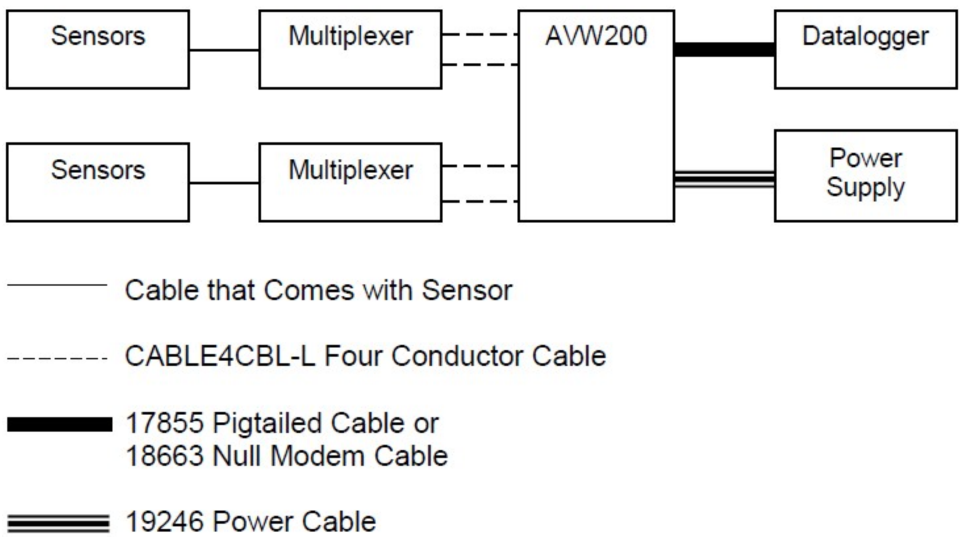
   Excitation voltage = 12 VDC peak to peak (option 2)

Refer to Programming with the AVW200() instruction (p. 44) for details about the `AVW200()` instruction and its parameters and to Wireless/one sensor/resistance converted to temperature (p. 55) for a complete example program that measures one vibrating-wire sensor with no multiplexer.

6.2 Multiplexers controlled by AVW200

6.2.1 Direct RS-232 connection

For this configuration, vibrating-wire sensors are attached to multiplexers, which are controlled by the AVW200. The AVW200 is connected directly to the data logger.



Do the following:

1. If a multiplexer other than the default is being used, go to the **Deployment/Measurement** tab in ***Device Configuration Utility*** and select the correct multiplexer (Connecting to Device Configuration Utility [p. 33] and Communications [p. 34]). The default multiplexer for the AVW200 is the AM16/32A.
2. Attach the multiplexers to the AVW200 as shown in Figure 7-4 (p. 30).
3. Connect the sensors to the multiplexers. Refer to the AM16/32B relay multiplexer manual: https://s.campbellsci.com/documents/us/manuals/am16-32b.pdf.
4. Use the serial cable to attach the AVW200 to control port pairs on the data logger, or use the null modem cable to attach the AVW200 to the data logger **RS-232** port.
5. Connect one end of the power cable to the **12V** and **G** terminals on the AVW200 and the other end to the **12V** and **G** terminals on the data logger or external power supply. Refer to Power and ground (p. 25) and Data logger wiring for a direct connection (p. 26) for more information.

6. Create a CRBasic program that includes the AVW200() instruction for each multiplexer to be controlled by theAVW200.

   For example, the following AVW200() instructions can be used to control two multiplexers:

```
AVW200(Data1(),ComC1,200,200,mux1(1,1),1,1,16,450,3000,2,_60HZ,1,0)
AVW200(Data2(),ComC1,200,200,mux2(1,1),2,1,16,450,3000,2,_60HZ,1,0)
```

   where:

   AVW200 connects to data logger control ports **C1/C2** via serial cable (option **Com1**)

   Each multiplexer has 16 sensors connected to it

   Begin frequency = 450

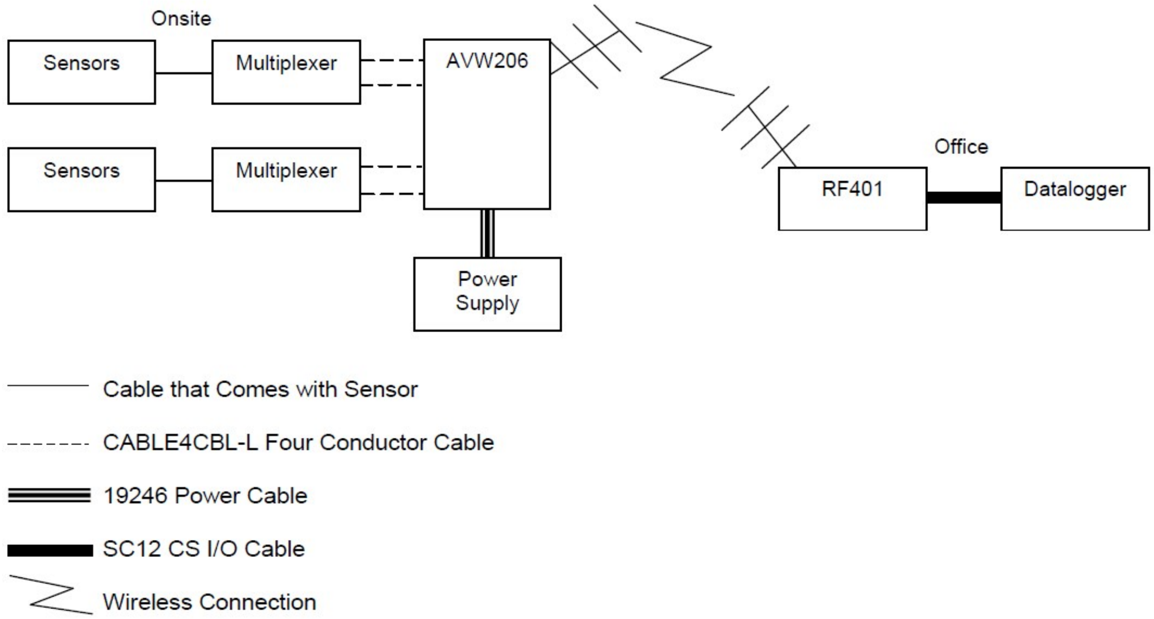
   End frequency = 3000

   Excitation voltage = 12 VDC peak to peak (option 2)

Refer to Programming with the AVW200() instruction (p. 44) for details about the AVW200() instruction and its parameters and to AVW200() instruction controlling two multiplexers (p. 56), AVW200() instruction running in pipeline mode (p. 57), and AVW200 controlling two multiplexers in sequential mode (p. 61) for complete example programs that control two multiplexers.

6.2.2 Wireless connection

For this configuration, vibrating-wire sensors are attached to multiplexers, which are controlled by an AVW206. The AVW206 interface transmits the data to an RF401A spread spectrum radio that is connected to the data logger.



At the AVW206 site, do the following:

1. Use ***Device Configuration Utility*** to configure the AVW206 for RF communications (Connecting to Device Configuration Utility [p. 33] and Communications [p. 34]).
2. If a multiplexer other than the default is used, go to the **Deployment/Measurement** tab in ***Device Configuration Utility*** and select the applicable multiplexer (Connecting to Device Configuration Utility [p. 33] and Communications [p. 34]). The default multiplexer for the AVW206 is the AM16/32A.
3. Attach the multiplexers to the AVW206 as shown in Figure 7-4 (p. 30).
4. Connect the sensors to the multiplexers. Refer to the AM16/32B relay multiplexer manual: https://s.campbellsci.com/documents/us/manuals/am16-32b.pdf.
5. Attach a Yagi or omnidirectional antenna to the antenna connector on the side of the AVW206. Refer to Antennas, antenna cables, and surge protectors for the AVW206, AVW211, and AVW216 (p. 76) for a description of antenna options.
6. Use the power cable to connect the **12V** and **G** terminals on the AVW206 to the **12V** and **G** terminals on the PS100 or another power supply.

At the data logger/RF401A site, do the following:

1. Configure the RF401A radio so its parameters match the AVW206.

   > **NOTE:**
   > The protocol setting for the RF401A must be PB Aware or PB Node.

2. Attach a Yagi or omnidirectional antenna to the antenna connector on the RF401A.
3. Use an SC12 serial cable to attach the data logger **CS I/O** port to the RF401A **CS I/O** port. The data logger **CS I/O** port powers the RF401A. Refer to the spread spectrum data radio/modem manual for more information: https://s.campbellsci.com/documents/us/manuals/rf401a-series.pdf.
4. Create a CRBasic program that includes the `AVW200()` instruction for each multiplexer controlled by the AVW200.

   For example, the following `AVW200()` instructions can be used to control the multiplexers:

   ```
   AVW200(Data1(),ComSDC7,200,200,mux1(1,1),1,1,16,450,3000,2,_60HZ,1,0)
   AVW200(Data2(),ComSDC7,200,200,mux2(1,1),2,1,16,450,3000,2,_60HZ,1,0)
   ```

   where:

   RF401A = configured for SDC7

   Each multiplexer has 16 sensors connected to it

   Begin frequency = 450

   End frequency = 3000

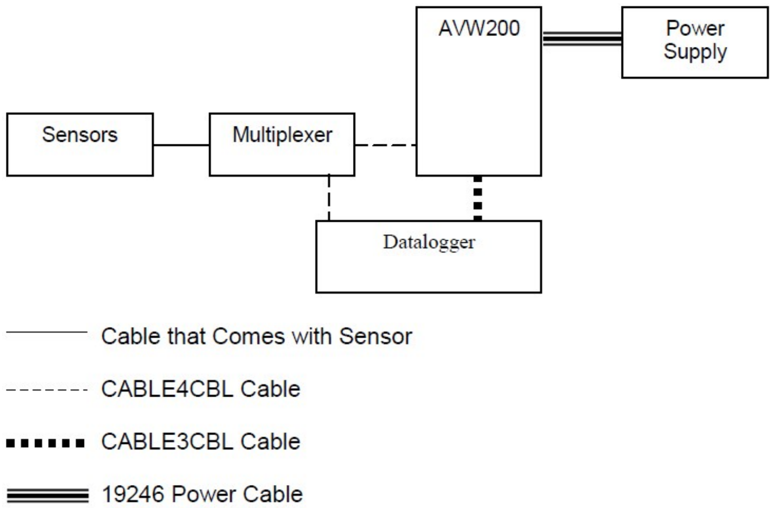
   Excitation voltage = 12 VDC peak to peak (option 2)

Refer to Programming with the AVW200() instruction (p. 44) for details about the `AVW200()` instruction and its parameters and to Wireless/sensors with different frequencies (p. 101) for a complete example program that controls two multiplexers.

6.3 Multiplexers controlled by data logger

For this configuration, SDI-12 is used to measure the vibrating-wire sensors. The vibrating-wire sensors are attached to multiplexers, which are controlled by the data logger.

6.3.1 SDI-12 communications



> **NOTE:**
> 1. When using SDI-12, multiplexers must be controlled by the data logger.
> 2. SDI-12 is the only option available for non-Campbell Scientific data loggers.

If using SDI-12 communications, do the following:

1. Access ***Device Configuration Utility*** to configure the AVW200 for SDI-12 communications. Go to the **Deployment/Measurement** tab in ***Device Configuration Utility*** and enter the SDI-12 address, multiplexer type, begin frequency, end frequency, and excitation (see Connecting to Device Configuration Utility [p. 33] and Measurement [p. 35]).
2. Use a CABLE4CBL-L cable to connect the AVW200 to the multiplexers (see Figure 7-5 [p. 31]).

   > **NOTE:**
   > SDI-12 uses **CLK** and **RESET** on the multiplexer instead of the **CLK** and **RESET** address on the AVW200.

3. Connect the sensors to the multiplexers. Refer to the spread spectrum data radio/modem manual for more information: https://s.campbellsci.com/documents/us/manuals/rf401a-series.pdf.
4. Use a CABLE4CBL-L cable to connect the multiplexers to the data logger (see Figure 7-6 [p. 32]).
5. Use a CABLE3CBL-L cable to connect the **C1** terminal on the AVW200 to a control port and ground on the data logger.

6. Connect one end of the power cable to the **12V** and **G** terminals on the AVW200 and the other end to the **12V** and **G** terminals on the data logger or external power supply. Refer to Power and ground (p. 25) and Data logger wiring for a direct connection (p. 26) for more information.
7. Program the data logger. Campbell Scientific CRBasic data loggers use the `SDI12Recorder()` instruction. The `SDI12Recorder()` instruction should only be run in the sequential mode for this configuration. See Programming for SDI-12 measurements (p. 50), SDI-12 example (p. 63).

7. Connections

7.1 Sensor wiring (no multiplexers)

Up to two vibrating-wire sensors can be directly connected to the AVW200 (see Figure 7-1 [p. 25]) using wires of specific colors. Refer to the sensor manual for details on wire colors and connections. Cable options for connecting the AVW200 to the data logger are listed in Figure 7-1 (p. 25).

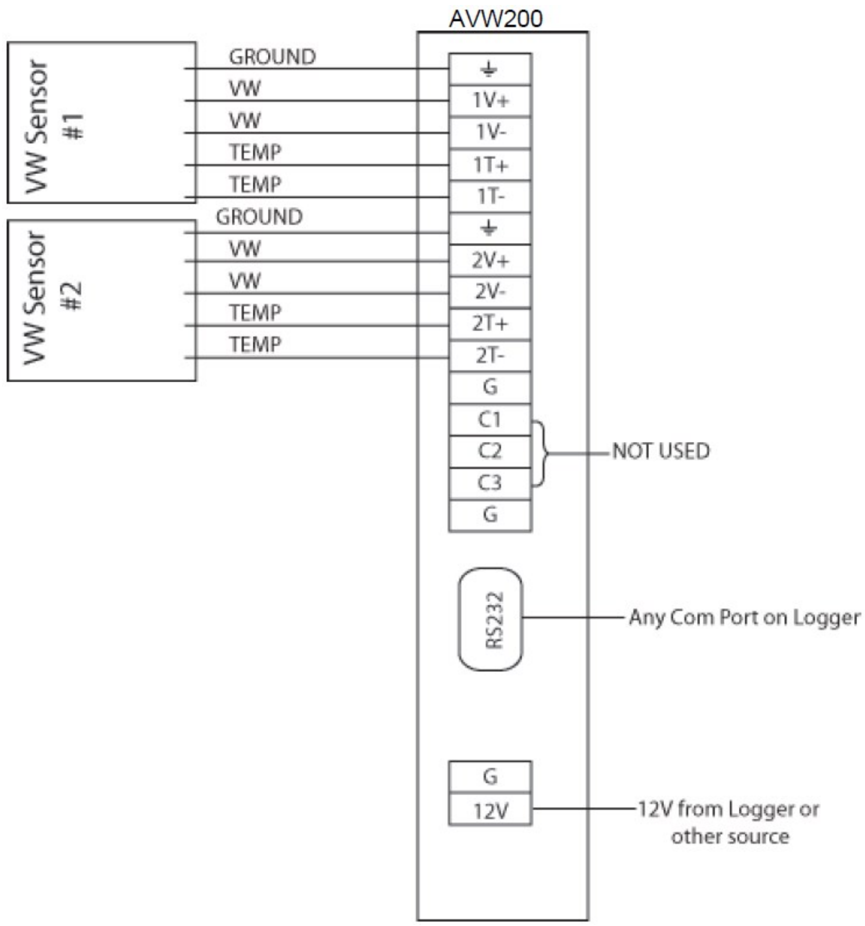


Figure 7-1. Wiring for sensor connections

7.2 Power and ground

Each AVW200 has a ground lug for connection to earth (chassis) ground and a green connector for attachment to a power source (see Figure 7-2 [p. 26]).

> **NOTE:**
> Only connect the AVW200 ground lug to earth ground when the AVW200 is not directly connected to the data logger. When a data logger is in the same enclosure, only connect the data logger ground lug to earth ground.

The AVW200 ground lug is connected to earth ground using an 8 AWG wire. This connection should be as short as possible.

The power cable terminates in pigtails that attach to the **12V** and **G** terminals on the AVW200 and the power source. The AVW200 is often powered by the data logger, but another 12 VDC power source may be used.

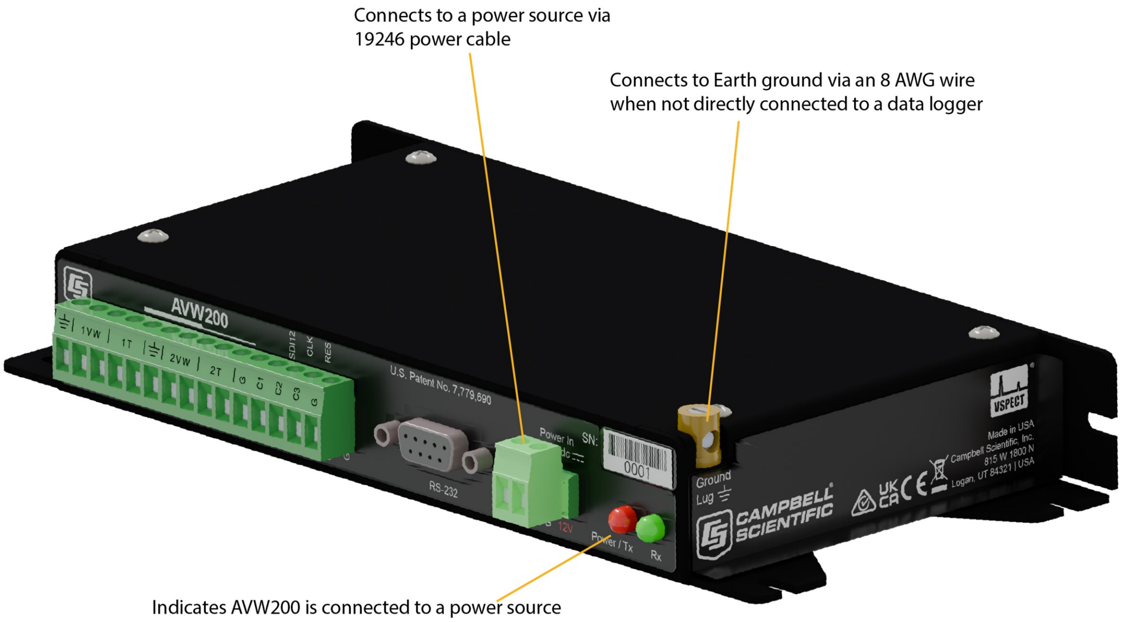


Figure 7-2. Ground lug and power connectors on the AVW200

7.3 Data logger wiring for a direct connection

Three options for connecting the AVW200 directly to the data logger are available (see Table 7-1 [p. 27]); the cable is ordered as a common accessory.

NOTE:
The CR6 is compatible with the AVW200. However, because the CR6 has built-in vibrating wire measurement capabilities, the AVW200 is unnecessary. See the VibratingWire() instruction in the CRBasic Help for details.

Table 7-1: Data logger to AVW200 cable options						
Communications protocol	Cable	Data logger port for cable attachment				AVW200 port for cable attachment
		CR6	CR1000X-Series	CR310	CR350	
PakBus (direct RS-232 connection)	Null modem cable	**RS-232**	**RS-232**	**RS-232**	**RS-232**	**RS-232**
PakBus (direct RS-232 connection)	DB9 pin-to-pigtail or SC110 DTE serial cable (see Table 7-2 [p. 27])	Control port pair (**C1**/**C2** or **C3**/**C4**) or **U** terminal pair and **G**	Control port pair (**C1**/**C2**, **C3**/**C4**, **C5**/**C6**, or **C7**/**C8**) and **G**	Control port pair (**C1**/**C2** and **G**	Control port pair (**C1**/**C2** and **G**	**RS-232**
SDI-12	CABLE3CBL-L cable	**C1** or **C3** and **G** or odd **U** terminal (**U1** to **U11**) and **G**	**C1**, **C3**, **C5**, or **C7** and **G**	**C1** or **C2** and **G**	**C1** or **C2** and **G**	**C1** or **C2** and **G**

Table 7-2: DB9 pin-to-pigtail or SC110 DTE serial cable wiring				
Wire color	CR6 terminal	CR1000X-Series terminal	CR310	CR350
Brown	Odd **C** or **U**	Odd **C**	**C1**	**C1**
White	Even **C** or **U**	Even **C**	**C2**	**C2**
Yellow	**G**	**G**	**G**	**G**

7.4 Wireless connections (AVW206, AVW211, AVW216)

The connector on the side of an AVW206, AVW211, and AVW216 is for attaching a whip antenna or antenna cable (see Figure 7-3 [p. 28]).

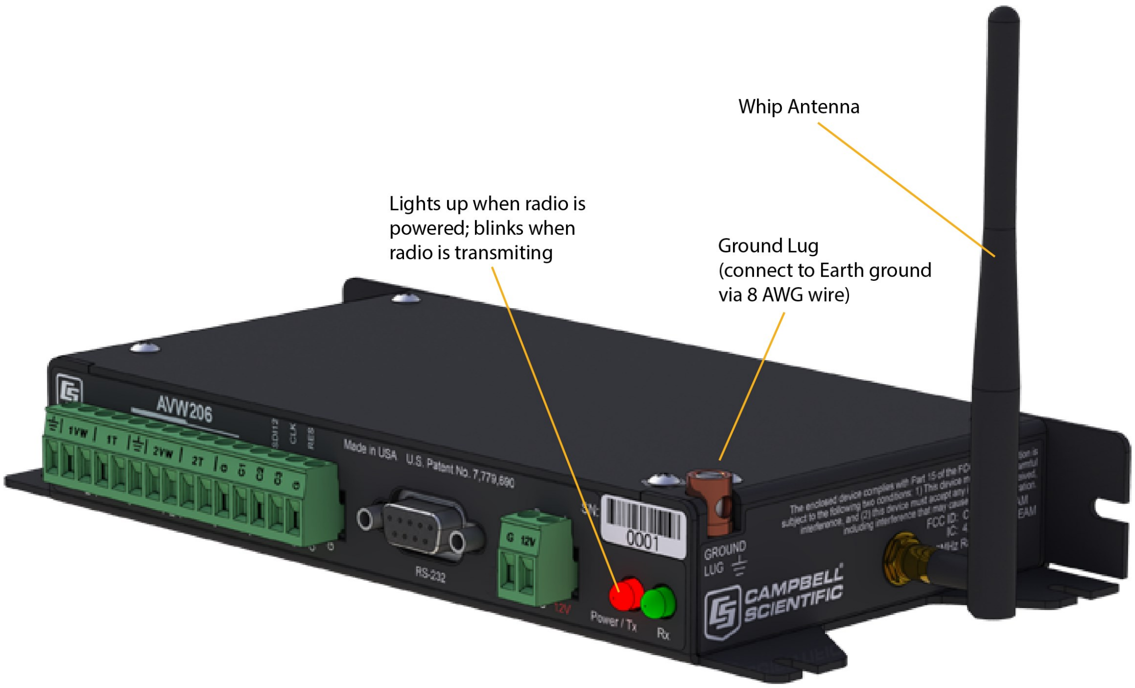


Figure 7-3. AVW206 with whip antenna

Wireless communications require the appropriate spread spectrum radio to be connected to the data logger (see Table 7-3 [p. 28]). ***Device Configuration Utility*** is used to configure the AVW206, AVW211, or AVW216 for RF communications (Connecting to Device Configuration Utility [p. 33] and Communications [p. 34]).

Table 7-3: Data logger to spread spectrum radio connections

Vibrating-wire interface model	Required spread spectrum radio model	Cable used for data logger to radio connection	Radio port for cable attachment	Data logger port for cable attachment
AVW206	RF401A	SC12	CS I/O	CS I/O
AVW211	RF411A	SC12	CS I/O	CS I/O
AVW216	RF416	SC12	CS I/O	CS I/O

NOTE:

1. (1) AVW206, AVW211, and AVW216 are incompatible with RF450, RF400, RF410, and RF415 spread spectrum radios.
2. (2) For communications between the vibrating-wire interface and spread spectrum radio to be successful, the radio **Protocol** must be set to **PakBus Aware** or **PakBus Node**.

Refer to Antennas, antenna cables, and surge protectors for the AVW206, AVW211, and AVW216 (p. 76) and to the spread spectrum data radio/modem manual: https://s.campbellsci.com/documents/us/manuals/RF401A-rf430.pdf.

7.5 Multiplexer wiring

Wire the sensors to the multiplexer according to the multiplexer manual. Other multiplexer wiring depends on whether the AVW200 or the data logger will control the multiplexer.

7.5.1 AVW200 control of the multiplexer

For most applications, having the AVW200 control the multiplexer is preferable. A CABLE4CBL or MUXSIGNAL cable connects to the multiplexer **COM** terminals, and another CABLE4CBL cable or the MUXPOWER cable connects to the multiplexer **CLK** and **RESET** terminals (see Figure 7-4 [p. 30]).

> NOTE:
> When two multiplexers are connected to the AVW200, they share the **CLK** and **RESET** lines. Because of this, while one multiplexer is making measurements, the other multiplexer will advance its channels but without making any measurements.

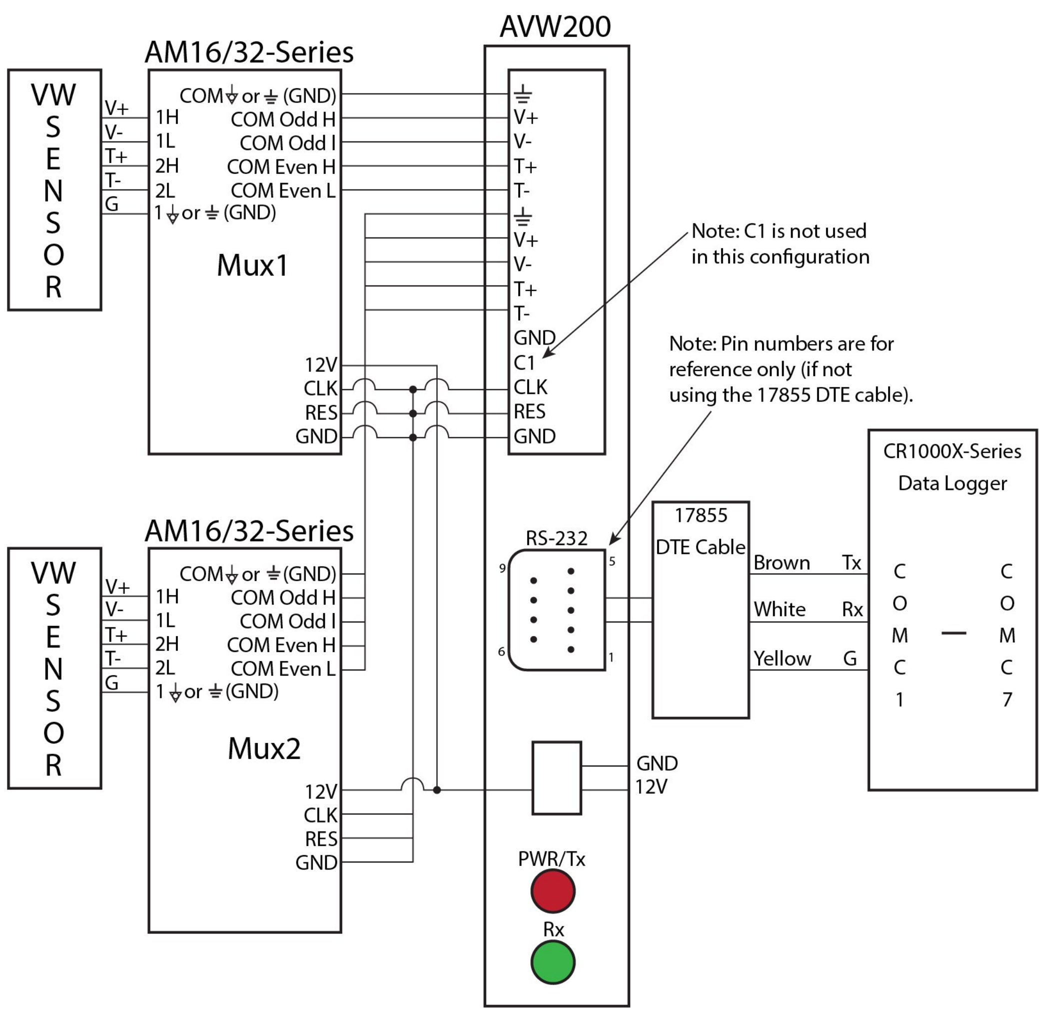


Figure 7-4. Example AM16/32-series to AVW200 hookup (multiplexers controlled by AVW200)

7.5.2 Data logger control of the multiplexer

When using SDI-12, the data logger must control the multiplexer. Use a CABLE4CBL-L cable to connect the AVW200 to the multiplexer if the vibrating-wire sensors contain a thermistor (see Figure 7-5 [p. 31]). Use the CABLE3CBL-L or equivalent cable if the vibrating-wire sensors do not contain a thermistor. The **CLK** and **RESET** lines on the multiplexer are used instead of the **CLK** and **RESET** lines on the AVW200.

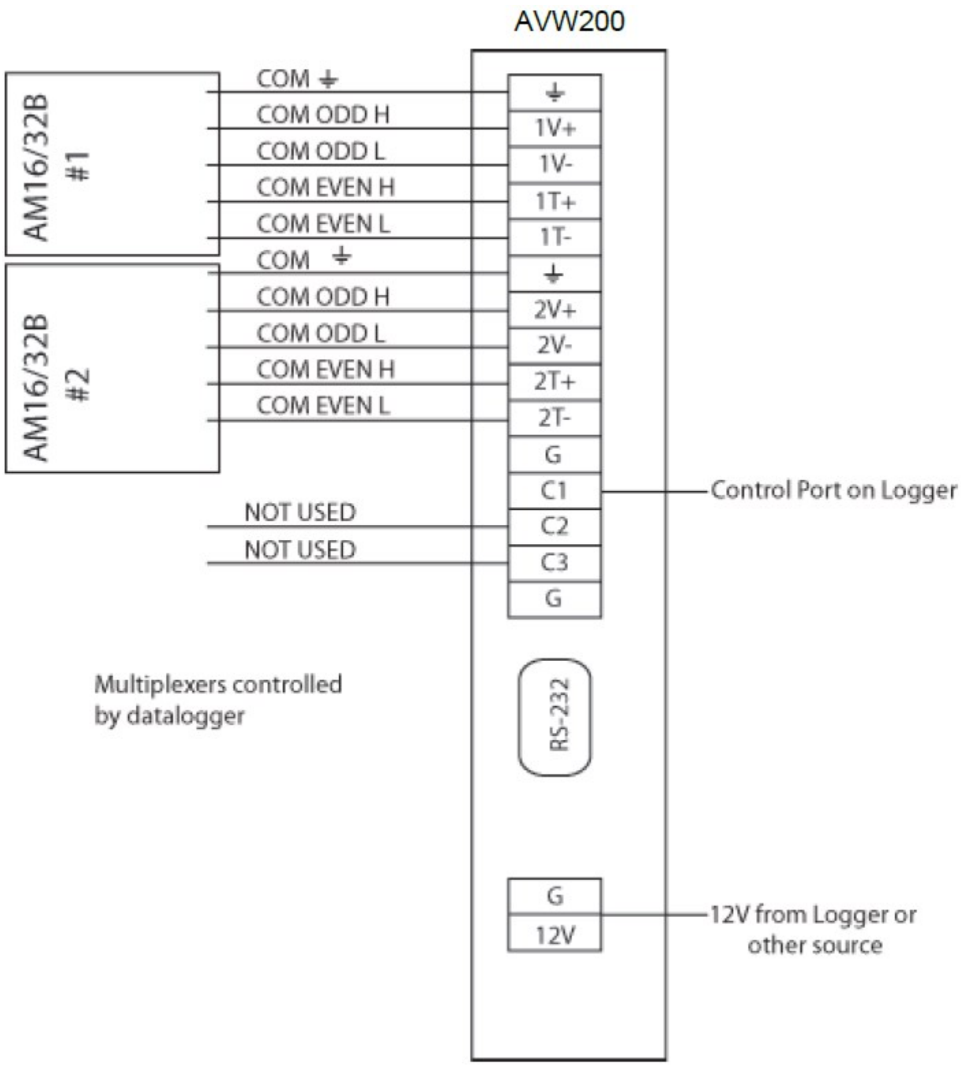


Figure 7-5. AM16/32B to AVW200 hookup (AM16/32Bs controlled by data logger and using SDI-12)

A CABLE4CBL-L cable is used to connect the multiplexer to the data logger (see Figure 7-6 [p. 32]).

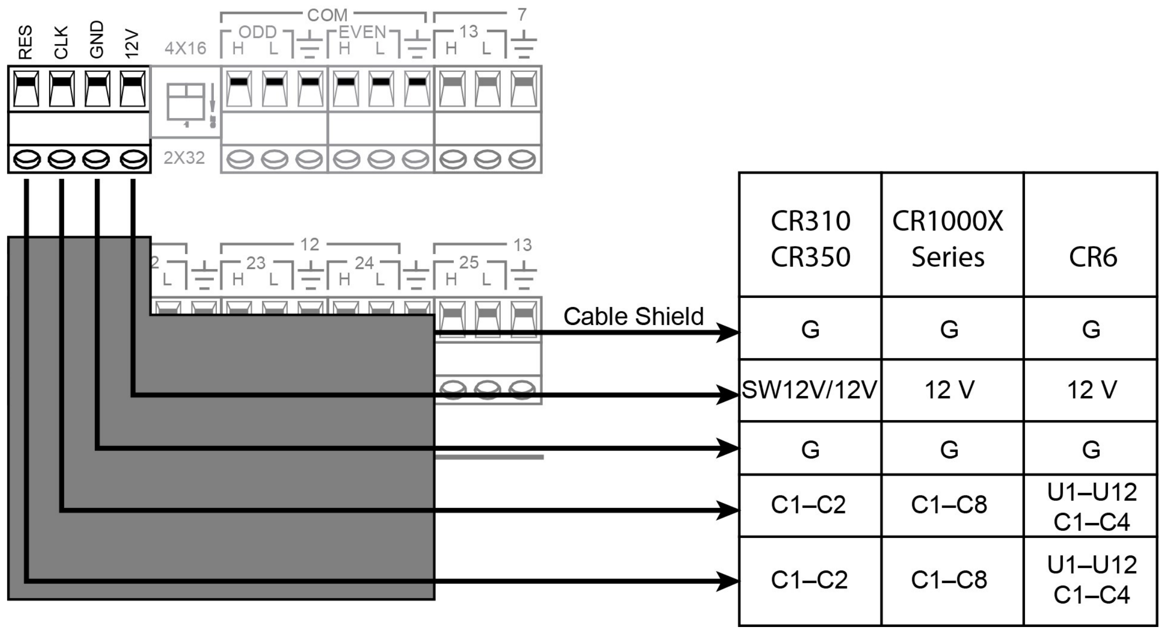


Figure 7-6. Multiplexer to data logger power/control hookup (multiplexer controlled by data logger)

8. *Device Configuration Utility*

Device Configuration Utility supports AVW200 configuration, real-time data display, operating system download, and vibrating-wire spectrum analysis troubleshooting. You will need to configure the AVW200 if you are using RF (PakBus), SDI-12, and RS-232 (terminal commands). To use ***Device Configuration Utility***, the AVW200 must be connected to a computer and a power source.

Device Configuration Utility support software is available as a free download from the Campbell Scientific website: www.campbellsci.com/downloads, www.campbellsci.com/19_1_9999_83. Check the version of any previously installed ***Device Configuration Utility*** software to ensure it is version 2.10 or later; install an upgrade if it is an earlier version.

> NOTE:
> (1) Default settings for the AVW200 may be used for many applications.
>
> (2) ***Device Configuration Utility*** has built-in help for each setting, which is displayed at the bottom of the window.

8.1 Connecting to *Device Configuration Utility*

To connect to ***Device Configuration Utility***, do the following:

1. Use the power cable to connect the AVW200 to the data logger 12 VDC supply or a regulated external power source. When connecting the power wires, the ground wire should be connected first, followed by the 12 VDC wire.
2. Connect the AVW200 to a **COM** port on your computer using the RS-232 cable.
3. Open ***Device Configuration Utility***.
4. Under **Device Type**, click **AVW 200 Series** (see Figure 8-1 [p. 33]).

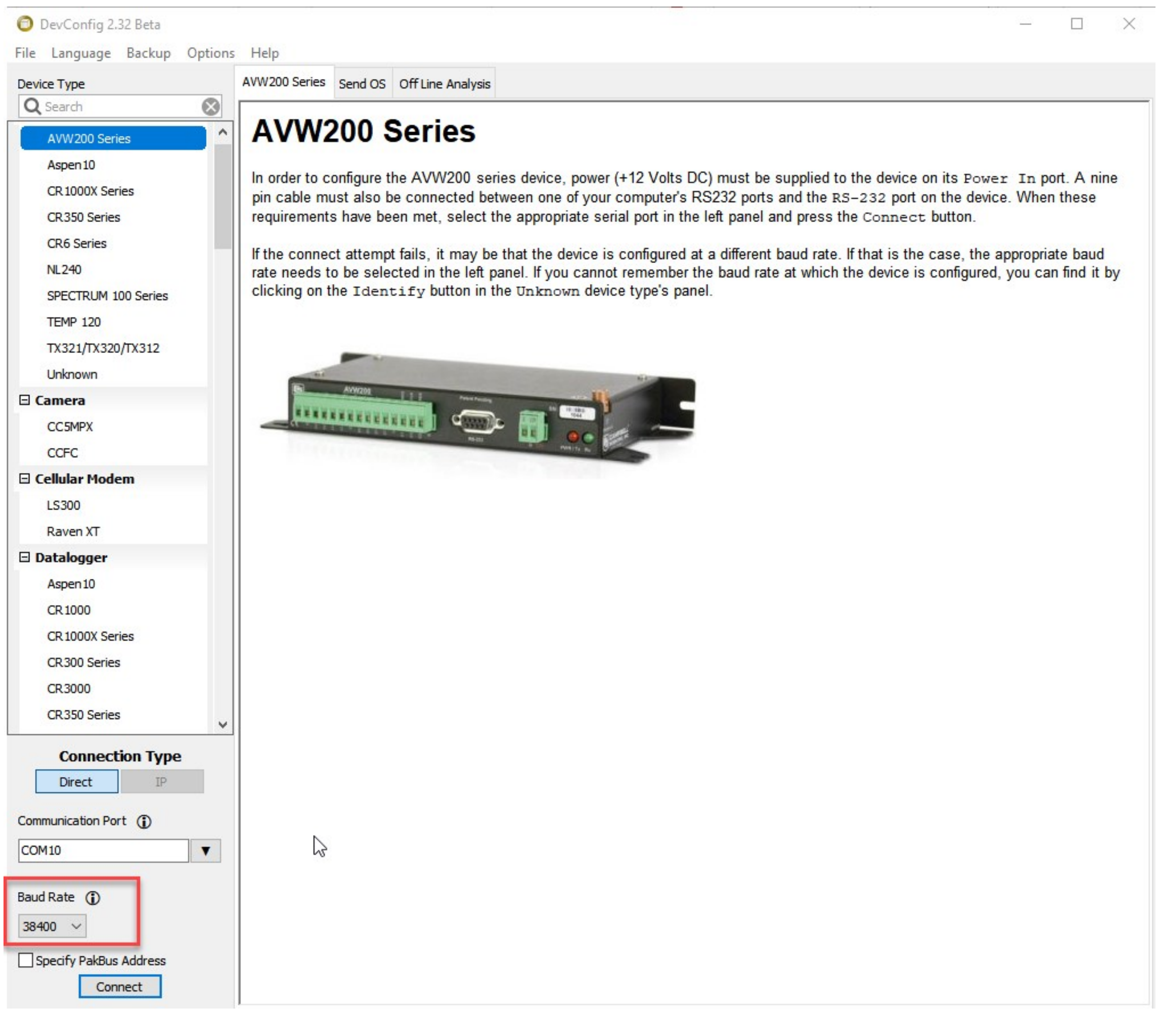


*Figure 8-1. Opening page in **Device Configuration Utility***

5. Select the **Serial** port matching the **COM** port on the computer to which the AVW200 is connected.
6. Use the drop-down menu to set the **Baud Rate**. The default baud rate is 38400.
7. Press **Connect**. The device may take up to 60 seconds to respond to ***Device Configuration Utility*** and for the current settings to be loaded into **Settings Editor**.

8.2 Deployment tab

8.2.1 Communications

The active **Deployment/Communications** tab appears at first connection (see Figure 8-2 [p. 34]). These settings are used for RF communications.

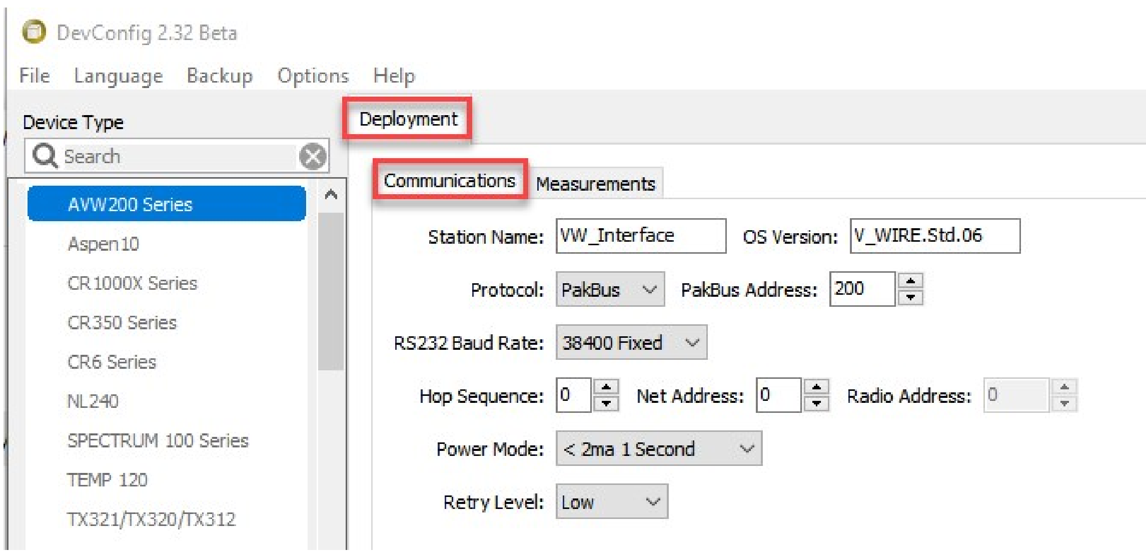


Figure 8-2. ***Deployment/Communications*** *tab in* ***Device Configuration Utility***

> **NOTE:**
> Certain AVW206 settings must match the RF401A settings for communications between the interface and radio to be successful.

The following list includes a description of **Communications Settings**:

Protocol: Choose **PakBus** for the **Protocol** setting. For communications to be successful, **Protocol** for the RF401A-series radio must be set to either **PakBus Aware** or **PakBus Node**.

PakBus address: Enter a **PakBus address** that matches the PakBus address specified in the CRBasic program that will control the AVW200. Each device in a PakBus network must have a unique address. Valid settings are 1–4094.

RS-232 baud rate: Enter the baud rate in which you want to communicate.

Hop sequence: Enter the radio **Hop sequence** that matches all RF401A radios and other AVW206 interfaces in the network. Valid entries are 0–6.

Net address: Enter the radio network address that matches all RF401A radios and other AVW206s in the network. Valid entries are 0–3.

Power mode: If not using a radio, select **Radio off** for the **Power mode**. Otherwise, select a power mode that works with the RF401A power mode (see Table 8-1 [p. 35]).

Table 8-1: AVW206 power modes and the recommended corresponding RF401A power modes

AVW206 power mode	Recommended RF401A power mode
<24 ma always on	<24 ma always on
<4 ma 1/2 second	<4 ma 1/2 second
<2 ma 1 second	<2 ma 1 second
<0.4 ma 8 second	<0.4 ma 8 second

Retry level: Select the desired **Retry level** (None, Low, Medium, or High) according to the level of RF collisions you expect. This depends on how many neighboring spread spectrum radios are in and out of your network and the frequency of transmissions. For most applications, select **Low**.

Once the settings have been defined, press **Apply** to save the changes to the AVW200.

8.2.2 Measurement

The **Deployment/Measurement** tab is used to configure the SDI-12 address, multiplexer type, begin frequency, end frequency, and excitation (see Figure 8-3 [p. 36]).

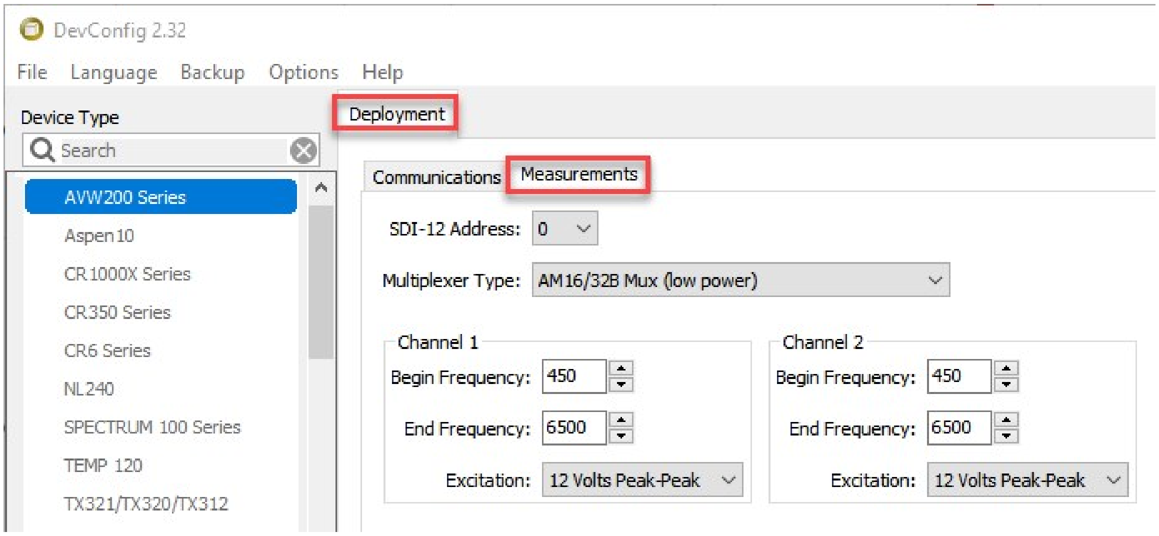


Figure 8-3. ***Deployment/Measurement*** *tab in* ***Device Configuration Utility***

> NOTE:
> The `begin frequency`, `end frequency`, and `excitation` parameters in ***Device Configuration Utility*** are only used for the RS-232 (terminal commands) and SDI-12 communications modes. When using RS-232 (PakBus) or RF (PakBus) communications modes, those parameter settings are entered in the `AVW200()` CRBasic instruction (see Programming with the AVW200() instruction [p. 44]).

The following list includes a description of **Measurement Settings**:

- **SDI-12 address:** Select an address if using SDI-12; otherwise, use the default value of 0.
- **Multiplexer type:** Choose the appropriate multiplexer type; the default is the AM16/32B. The AM16/32 and AM16/32A models support higher current requirements. For lower power operation, select the AM16/32B. Choose 'Other Mux' for multiplexers with slower clocking requirements and longer delays between Reset and Clock signals, including a 20 ms delay from Reset to Clocking and 20 ms Clock delays. See Multiplexer use (p. 7) for more information.
- **Begin frequency:** If using RS-232 (terminal commands) or SDI-12, enter the sensor manufacturer recommendation for this value.
- **End frequency:** If using RS-232 (terminal commands) or SDI-12, enter the sensor manufacturer recommendation for this value.

- **Excitation:** If using RS-232 (terminal commands) or SDI-12, choose either **5 Volts Peak–Peak** or **12 Volts Peak–Peak** excitation; use the sensor manufacturer recommendation for this value.

Once settings have been defined, press **Apply** to save the changes to the AVW200.

8.3 Data monitor

The **Data Monitor** tab in ***Device Configuration Utility*** can display either the **Public** table or **Status** table. The **Public** table displays the current sensor measurement values as well as the current settings (see Figure 8-4 [p. 37]). Refer to Public table (p. 83) for fields in the **Public** table and a brief description of each.

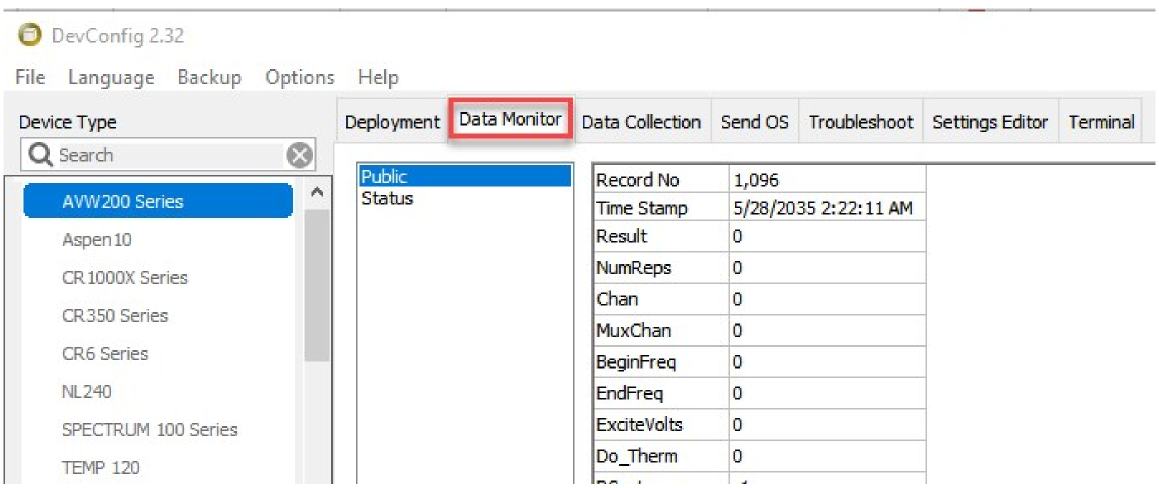


*Figure 8-4. Data display/**Public** table in **Device Configuration Utility***

The **Status** table contains accessible system operating status information (see Figure 8-5 [p. 38]).

> NOTE:
> ***Device Configuration Utility*** polls the **Status** table at regular intervals and then updates the status information. Refer to Status table (p. 88) for a comprehensive list of **Status** table variables and a brief description of each.

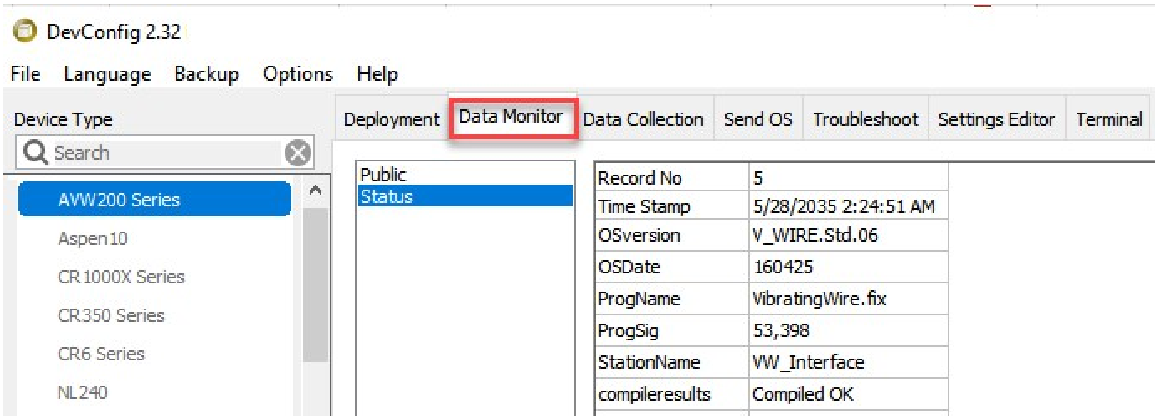


*Figure 8-5. Data display/**Status** table in **Device Configuration Utility***

8.4 Send OS

For most applications, Campbell Scientific does not anticipate the downloading of a new operating system to the AVW200 to be necessary. However, if a new operating system (OS) is required, ***Device Configuration Utility*** version 2.10 is needed.

Use a DB9 serial cable to connect the **RS-232** port of the AVW200 to a serial port on your computer, then do the following:

1. Remove power from the AVW200.
2. Open ***Device Configuration Utility***.
3. Highlight the AVW200 in the list of devices in the left portion of the window.
4. Select the **COM** port to which the AVW200 is connected from the drop-down list box located on the lower left.
5. Select the correct **Baud Rate**.

6. Click the **Start** and follow the directions on the screen (Figure 8-6 [p. 39]).

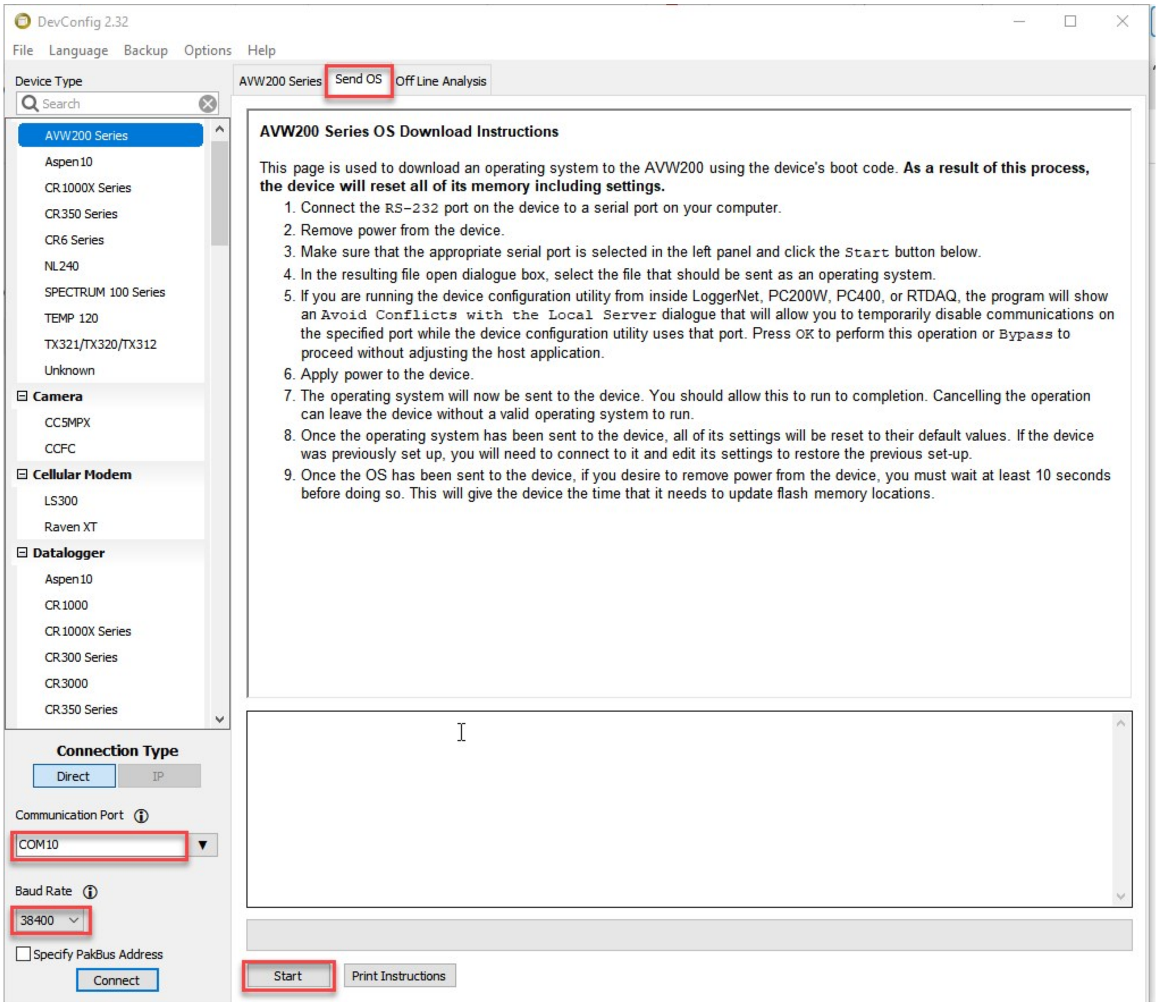


*Figure 8-6. Sending an operating system using **Device Configuration Utility***

8.5 Troubleshoot

The **Troubleshoot** tool in ***Device Configuration Utility*** can be used to evaluate the frequency spectrum of a sensor and to determine the most appropriate begin and end frequencies for a sensor.

To access **Troubleshoot**, do the following:

1. Follow the connection procedure provided in Connecting to Device Configuration Utility (p. 33).

2. Click the **Troubleshoot** tab on the ***Device Configuration Utility*** opening window (Figure 8-7 [p. 40]).

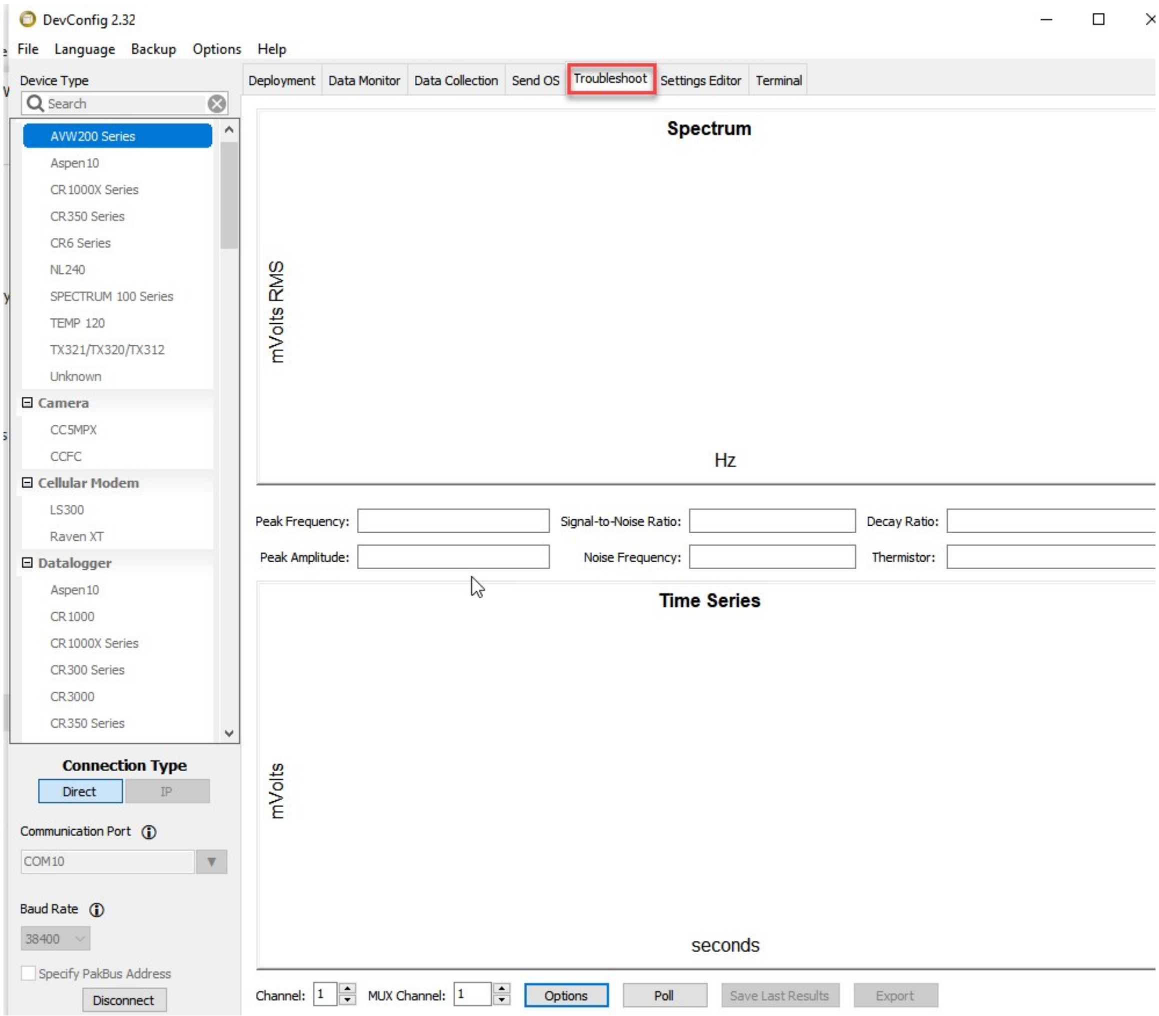


Figure 8-7. Opening page of the Troubleshoot tool

3. Click the lower-left **Options** tab to set the begin and end frequencies and the excitation voltage to test for a given sensor (Figure 8-8 [p. 41]). You may also choose to poll (default) or not poll the time series data from this **Options** window by checking or unchecking the **Poll** box.

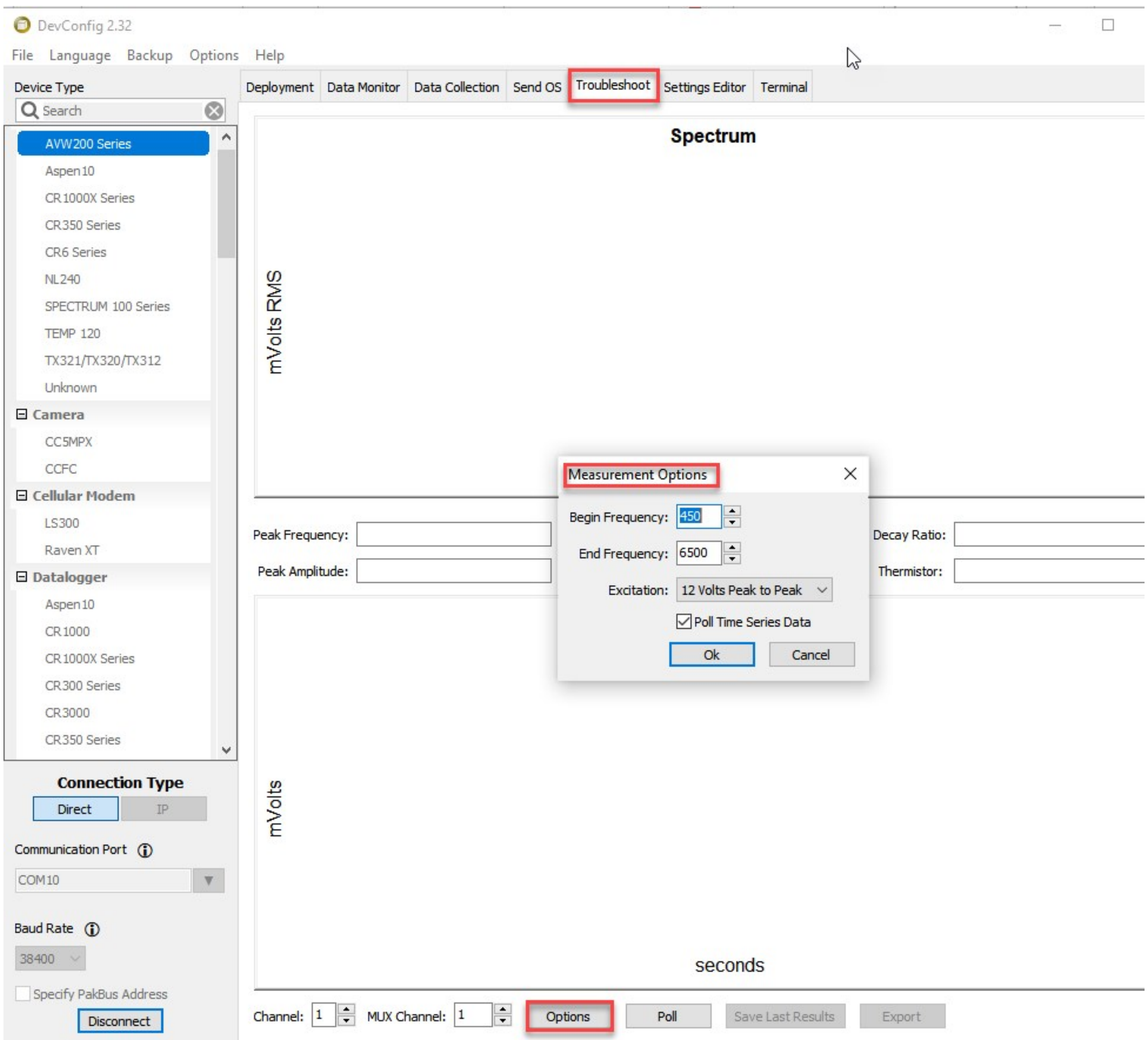


Figure 8-8. Options tab of the Troubleshoot tool

4. Select AVW200 channel either **1** or **2** and the multiplexer channel to which the sensor is attached. If not using a multiplexer, set the multiplexer channel to 1.

5. Once the appropriate settings have been specified in the **Options** window, click **OK**, then click the **Poll** tab on the lower-left portion of the **Troubleshoot** window. The results of the poll will be displayed on a Spectrum graph and a Time Series graph (see Figure 8-9 [p. 42]).

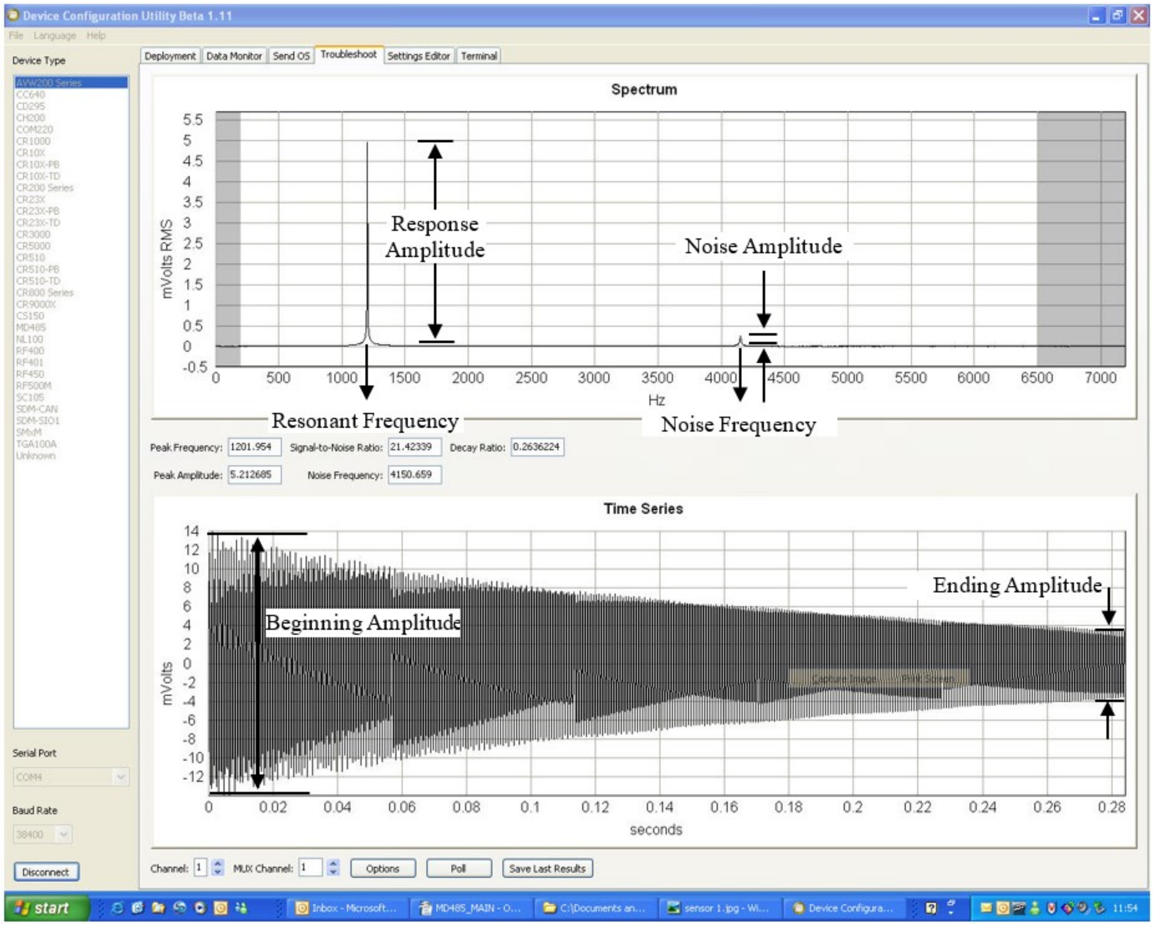


Figure 8-9. Graphs for evaluating spectral analysis of a sensor

> **NOTE:**
> Check manufacturer specifications for the sensor frequency and excitation range before picking the begin and end frequencies and excitation voltage.

In Figure 8-9 (p. 42), the lower graph shows the raw time-series data recorded from a vibrating-wire sensor after the sensor has been excited with the swept-frequency voltage signal. The upper graph shows the spectrum after the AVW200 has applied the FFT. In addition to the wire resonant frequency, the spectrum shows the response amplitude, noise amplitude, and noise frequency. The AVW200 computes the signal-to-noise ratio diagnostic by dividing the response amplitude by the noise amplitude. The AVW200 computes the decay ratio diagnostic from the time series end amplitude divided by the begin amplitude. (See Time series and Spectrum graph information (p. 92) for more information.)

6. The results of the poll may be saved by clicking the **Save Last Results** tab in the lower window.

8.6 Settings editor

Settings Editor in ***Device Configuration Utility*** can also be used to enter the **Deployment** parameters (see Figure 8-2 [p. 34]). Refer to Deployment tab (p. 34) and Data monitor (p. 37) for a description of the setting parameters.

8.7 Terminal

You can monitor the AVW200 with terminal commands via the terminal emulator in ***Device Configuration Utility*** or ***LoggerNet***. You can also use a terminal emulator, such as ***HyperTerminal*** or ***ProComm***.

To enter terminal commands, first connect the **RS-232** port of the AVW200 to a serial port on your computer using an RS-232 socket-to-pin serial cable. After specifying the appropriate **COM** port (port to which the AVW200 is attached) and communications baud rate (AVW200 baud rate = 38400), press **Carriage Return** (CR) four times or until `AVW200>` is returned. A description of the available terminal commands and the values returned for each command is listed in 8.7 (p. 43).

Table 8-2: Terminal mode commands

Command	Description	Return values
1	Make VW measurement on channel 1	Six values: resonant frequency, response amplitude, signal-to-noise ratio, noise frequency, decay ratio, thermistor resistance
2	Make VW measurement on channel 2	Six values: resonant frequency, response amplitude, signal-to-noise ratio, noise frequency, decay ratio, thermistor resistance
3	Company and model info	Company, mode, and PakCode information
4	Status table	Status table
5	Public table	Public table
S or s	Set date/time	Set the date and time
G or g	Get date/time	Get the date and time

Table 8-2: Terminal mode commands		
Command	Description	Return values
Mcmm	Make VW measurement on channel (c) and mux channel (mm)	Six values: resonant frequency, response amplitude, signal-to-noise ratio, noise frequency, decay ratio, thermistor resistance
H or h	Help menu	List of commands

NOTE:
The `Mcmm` terminal command was added to OS version STD.02. Customers with OS version STD.01 can get this terminal command by downloading a new OS (refer to Send OS [p. 38]).

CAUTION:
When using the `Mcmm` terminal command, no other method of measurement should be used, else multiplexing gets out of sequence and measurement errors result.

Refer to Public table (p. 83) and Status table (p. 88) for more information.

9. Programming with the `AVW200()` instruction

When using the RS-232 (PakBus) or RF (PakBus) protocol, the data logger is programmed using the `AVW200()` instruction in CRBasic.

NOTE:
If using SDI-12 to communicate with the AVW200, then use the `SDI12Recorder()` instruction to trigger and retrieve measurements from the AVW200 (see Programming for SDI-12 measurements [p. 50]).

The `AVW200()` instruction is used to read measurements from one or more vibrating-wire sensors when the sensors are connected to the AVW200 channels directly or connected to a multiplexer attached to the AVW200. In addition to frequency, diagnostic information is returned. (Refer to the Dest parameter in the following section.)

The data logger program can run the `AVW200()` instruction in either the pipeline mode (Pipeline mode [p. 48]) or sequential mode (Sequential mode [p. 49]). In the pipeline mode, the first execution of the instruction sets up the AVW200; subsequent execution intervals retrieve data values. If different begin and end frequencies are required to measure different types of sensors, use multiple AVW200() instructions with the different begin and end frequencies specified in each instruction. The sequential mode performs each instruction in sequence, waits for the completion of each instruction, then repeats this process for each execution interval. The minimum scan rate for an AVW200 program is 2 seconds per sensor.

Syntax

```
AVW200 (Result, ComPort, NeighborAddr, PakBusAddr, Dest, AVWChan, MuxChan, Reps,
BeginFreq, EndFreq, ExVolt, Therm50_60Hz, Multiplier, Offset, AmpThreshold
[optional])
```

`Result` Result Variable that indicates the success or failure of the communications attempt of the data logger with the AVW200. If more than one AVW200 is used and the instructions are run in sequential mode, a different result variable should be specified for each AVW200 (see Sequential mode [p. 49]). The result codes are as follows:

Code	Description
0	Communications successful. Values have been written to the destination array.
>1	Number of communications failures. **NAN** values will be stored in the destination array. Resets to 0 upon successful communications.
–3	First communication. Values will be available on the next scan.

`ComPort` ComPort Used to specify the communications port that will be used to communicate with the AVW200. Enter a numeric or alphanumeric code:

Alphanumeric	Description
ComRS232	AVW200 connects to data logger **RS-232** port via null modem cable
ComME	RF401A connects to data logger **CS I/O** port; RF401A configured as modem enabled
ComSDC7	RF401A connects to data logger **CS I/O** port; RF401A configured as SDC7
ComSDC8	RF401A connects to data logger **CS I/O** port; RF401A configured as SDC8
ComSDC10	RF401A connects to data logger **CS I/O** port; RF401A configured as SDC10

ComSDC11	RF401A connects to data logger **CS I/O** port; RF401A configured as SDC11
ComC1	AVW200 connects to data logger control ports **C1**/**C2** via DB9 serial cable
ComC3	AVW200 connects to data logger control ports **C3**/**C4** via DB9 serial cable
ComC5	AVW200 connects to data logger control ports **C5**/**C6** via DB9 serial cable
ComC7	AVW200 connects to data logger control ports **C7**/**C8** via DB9 serial cable

`NeighborAddr` Used to specify a static route to the AVW200 (for example, the PakBus address of a neighbor device through which the host can communicate with the AVW200). If **0** is entered, the AVW200 is assumed to be a neighbor, meaning that the host data logger can communicate with the AVW200 directly.

`PakBusAddr` Parameter to identify the PakBus address of the AVW200 with which the host data logger is trying to communicate. Valid entries are **1** through **4094**. Each PakBus device in the network must have a unique address.

`Dest` Variable array in which to store the results of the instruction. If only one sensor is being measured, `Dest` is a single-dimensioned array of five if no thermistor is being measured or six if a thermistor is. `Dest` is a multi-dimensioned array of five (no thermistor) or six (with thermistor) if multiple sensors are being measured using a multiplexer. The first dimension is set equal to the number of sensors being measured, and the second dimension is set equal to the number of values returned for each sensor (5 or 6). For example, to measure four sensors with thermistor measurements attached to a multiplexer, `Dest` would be declared as Array(4,6). Values for sensor 1 would be stored in Array(1,1) through Array(1,6), values for sensor 2 stored in Array(2,1) through Array(2,6), and so on.

The values returned for each sensor are:

(1) Resonant frequency in Hz

(2) Response amplitude in mV RMS

(3) Signal-to-noise ratio

(4) Noise frequency

(5) Decay ratio

(6) Thermistor in ohms (if measured)

The units and description of these values are provided in Table 4-2 (p. 5).

If communications are unsuccessful, **NAN**s are stored.

> **NOTE:**
> When using the CRBasic `AVW200()` instruction, the signal strength amplitude in mV RMS (Dest[2]) has a minimum resolution of 0.0625 mV (62.5 microvolts). This means that values between 0.01 and 0.06249 mV (10 and 62.49 microvolts) will be given as **0**. Values between 0.0625 and 0.1249 will be given as **0.0625**. Values between 0.125 and 0.18749 will be given as **0.125**, and so forth. With firmware Std.04 and higher, values less than 0.01 will be shown as **NAN**; older firmware will show them as **0**.

To see amplitudes with higher resolution, use SDI-12 (), Terminal mode (Terminal [p. 43]), or the Troubleshoot tab in Device Configuration Utility (Troubleshoot [p. 39]) to examine signal strength amplitude values.

`AVWChan` Channel on the AVW200 where the sensor or multiplexer is wired. Valid options are **1** (+/–1VDC) or **2** (+/–2VDC).

`MuxChan` Channel on the multiplexer where measurements should start. Valid options are **1** through **32**; anything outside this range returns an error. Enter **1** if a multiplexer is not used.

`Reps` Number of measurements to be made on the multiplexer. This parameter does not affect `AVW200Chan`.

`BeginFreq` Beginning frequency to use for the vibrating-wire measurement. Minimum possible value is **100**. Refer to the specifications of the vibrating-wire sensor for recommended `BeginFreq` values.

`EndFreq` Ending frequency to use for the vibrating-wire measurement. Maximum possible value is **6500**; typical sweep range is **450** to **6000**. Refer to the specifications of the vibrating-wire sensor for recommended `EndFreq` values.

`ExVolt` Excitation voltage to be used to excite the vibrating wire. Valid options are **1** (5 VDC peak-to-peak) or 2 (12 VDC peak-to-peak).

`Therm50_60Hz` Used to set the integration time for the thermistor in the vibrating-wire sensor.

Code	Description
0	No thermistor measurement (five values returned in Dest)
_60Hz	Use 60 Hz noise rejection (six values returned in Dest)
_50Hz	Use 50 Hz noise rejection (six values returned in Dest)

Multiplier
Multiplier to be used for frequency measurements.

Offset
Offset to be used for frequency measurements.

AmpThreshold
(for data logger OS versions 23 and higher) Optional parameter that is used to define a minimum value in mV for the amplitude of the signal. If an amplitude less than the threshold is measured, **NAN** (invalid) will be stored for the frequency measurement. If AmpThreshold is omitted, a default value of 0.01 mV is used. If a value <0.01 mV is entered for this parameter, the precompiler will return an error.

> NOTE:
> The signal strength amplitude ((2) (p. 46)) returned by the AVW200() CRBasic instruction is limited in its resolution.To avoid unexpected results, use a value that is an integer multiple of 0.0625 for the AmpThreshold parameter.

9.1 Pipeline mode

When the CRBasic program starts running, the information specified in the AVW200() instruction is sent to the attached AVW200 interface module via the communication port and PakBus address specified in the instruction. Along with the instruction parameter information, the data logger also sends its clock information. For every execution interval, the data logger clock and measured data are exchanged with the attached AVW200 interface module. The AVW200 interface module then determines how early before the data logger interval the vibrating-wire measurement should start so that the measurement series finishes before the data logger execution interval.

As the first parameter in the AVW200() instruction, a returned Result value of 0 indicates successful communications and confirms that the data values have been stored in the destination variable. If a failure in communications occurs, the Result value increments for each failure, and destination values are filled with **NAN**s. The data logger will retry communications three times before returning a failed communication or incrementing the result. Retries occur every 3 seconds or greater, depending on the radio-power-cycle configuration. A negative value returned for the Result variable indicates status information. For example, **–3** indicates the

AVW200 has not made the first measurement; **–4** indicates that no parameter information is available. Multiple `AVW200()` instructions can use the same `Result` variable; in fact, having different `Result` variables for a given communications port in the pipeline mode is discouraged. Refer to AVW200() instruction running in pipeline mode (p. 57) for an example program.

> NOTE:
> In the pipeline mode of operation, the `AVW200()` instruction cannot be in a conditional statement or in a subroutine.

9.2 Sequential mode

In sequential mode, during each execution interval, the CRBasic program executes each instruction in sequence, meaning that the first `AVW200()` instruction is executed through to completion, followed by the second instruction, and so on. The first `AVW200()` instruction will communicate with the attached AVW200 interface module and tell the AVW200 to make the measurements. The data logger will then wait at the `AVW200()` instruction until all measurements are returned for that first instruction, or until the instruction times out. Timeout occurs after two times the number of reps. If an instruction times out, it will make three additional attempts before advancing to the next instruction. The next `AVW200()` instruction is then executed.

`Result` codes returned via sequential mode function the same as in the pipeline mode; however, having different `Result` code variables for each `AVW200()` instruction is recommended in sequential mode. This is because if communications were disconnected between two `AVW200()` instructions, then data collected via the first instruction would correctly be stored into destination variables for that instruction, but destination variables for the second instruction would be filled with NANs, and the result code would increment, which indicates a failed communication. With different `Result` variables for each `AVW200()` instruction, this situation can be detected because the `Result` variable for the first instruction would be `0`, indicating successful communications, and the `Result` variable for the second instruction would increment, indicating failed communications. In the pipeline mode, this situation does not exist, so the `Result` code variables can be the same for multiple `AVW200()` instructions on a given communications port. Refer to AVW200() instruction running in sequential mode (p. 59) for an example program.

NOTE:
(1) When running in the sequential mode, programs that contain multiple AVW200 instructions using the same **COM** port should have different `Result` variables for each `AVW200()` instruction, such as `Result1` and `Result2`, to detect and isolate any communications errors for a given AVW200.

(2) If the `AVW200()` instruction is in a slowsequence, the mode of operation is always forced to sequential mode.

10. Programming for SDI-12 measurements

An alternative to the `AVW200()` instruction is to trigger and retrieve measurements from the AVW200 using the SDI-12 protocol. Campbell Scientific CRBasic data loggers use the `SDI12Recorder()` instruction to make measurements over SDI-12, which should only be run in the Sequential mode (p. 49). When using SDI-12, the data logger must control any multiplexers. Additionally, SDI-12 is the only method for non-Campbell Scientific data loggers to communicate with an AVW200.

10.1 SDI12Recorder() instruction

The values returned from the `SDI12Recorder()` instruction are different depending on the SDI-12 measurement command issued. The `SDI12Recorder()` instruction sends the command specified by the `SDI12Command` parameter as `(address)SDI12Command!`. The AVW200 always returns six values.

Sytnax

```
SDI12Recorder ( Dest, SDIPort, SDIAddress, "SDICommand", Multiplier, Offset )
```

The `SDI12Recorder()` instruction has the following parameters:

`Dest` Variable in which to store the results of a measurement. `Dest` must have enough elements to store all data returned by the SDI-12 sensor, or a **variable out of range** error will result during the execution of the instruction.

`SDIPort` SDIPort Port to which the SDI-12 sensor is connected. A numeric value is entered:

Code	Description
1	Control port 1
3	Control port 3
5	Control port 5
7	Control port 7

`SDIAddress` Address of the SDI-12 sensor that will be affected by this instruction. Valid addresses are **0** through **9**, **A** through **Z**, and **a** through **z**. Alphabetical characters should be enclosed in quotation marks (for example, **"0"**).

`SDICommand` Used to specify the command strings that will be sent to the sensor. The command should be enclosed in quotation marks. Table 10-1 (p. 51) shows the specific SDI-12 command codes and their returned values.

Table 10-1: SDI-12 command codes

SDI-12 measurement command	Returned values
`aM!` (measures both AVW200 channel 1 and channel 2)	1) Chan1 frequency 2) Chan1 therm resistance 3) Chan1 signal amplitude mV RMS 4) Chan2 frequency 5) Chan2 therm resistance 6) Chan2 signal amplitude mV RMS
`aM1!` (measures AVW200 channel 1; may be used for non-Campbell Scientific data loggers)	1) Chan1 frequency 2) Chan1 signal amplitude mV RMS 3) Chan1 signal-to-noise ratio 4) Chan1 noise frequency 5) Chan1 decay ratio 6) Chan1 therm resistance

Table 10-1: SDI-12 command codes	
SDI-12 measurement command	**Returned values**
`aM2!` (measures AVW200 channel 2; may be used for non-Campbell Scientific data loggers)	1) Chan2 frequency 2) Chan2 signal amplitude mV RMS 3) Chan2 signal-to-noise ratio 4) Chan2 noise frequency 5) Chan2 decay ratio 6) Chan2 therm resistance
`aV!`	1) Battery voltage 2) Trapcodes 3) Watchdog counts
`aXVWbbbb,eeee,v!` where: bbbb = Begin freq (100 < bbbb < 6.5K) eeee = End freq (100 < eeee < 6.5K) v = Excite voltage (1 = 5VDC and 2 = 12 VDC)	Returns **OK** if accepted Returns **ERROR** if not accepted

If a check summed command fails, **NAN** will be returned, and the command will be retried.

`Mult, Offset` Each a constant, variable, array, or expression by which to scale the results of the measurement.

10.2 Use with multiplexers

The AVW200 interface module cannot control multiplexers in the SDI-12 communications mode. Hence, when communicating with the AVW200 via SDI-12, multiplexers attached to the AVW200 must be controlled by the data logger. This is achieved by using `PortSet()` instructions in the data logger program and by connecting the clock and reset lines of the multiplexers to control ports on the data logger. When using SDI-12 with the AVW200, the clock and reset lines on the AVW200 are **not** used.

Refer to SDI-12 example (p. 63) to see example programs.

11. Program examples

This section includes several program examples for the Campbell Scientific CR1000X-Series data loggers. Although the examples are for the CR1000X, programming for other data loggers is similar. More complex programming examples are provided in Additional programming examples (p. 97). Downloadable example programs are available at www.campbellsci.com/downloads/avw200-example-programs.

11.1 AVW200() instruction with no multiplexers

11.1.1 Direct RS-232 connection with two sensors

Table 11-1 (p. 53) shows wiring used for this example.

Table 11-1: Wiring for a direct RS-232 connection with two sensors

Data logger port for cable attachment	Cable needed to connect to AVW200	AVW200 port or model
ComC1 (control port pairs **C1**/**C2**)	DB9 pin-to-pigtail cable	RS-232

CRBasice Example: Direct RS-232 connection

```
'CR1000X Series Data Logger
'This program measures 2 sensors on the AVW200, 1 on channel 1 and 1 on channel 2

Public PTemp, batt_volt
Public Dst(2,6)
Public result

DataTable (AVW200,1,-1) 'stores data from both sensors into a table named AVW200
  DataInterval (0,10,Sec,10)
  Sample (6,Dst(1,1),IEEE4)
  Sample (6,Dst(2,1),IEEE4)
EndTable

'The CardOut instruction is used to create a new DataTable that will be saved on
'a compact flash card.
DataTable (AVWcard,1,-1)
  CardOut (0 ,-1)
  DataInterval (0,10,Sec,10)
  Sample (6,Dst(1,1),IEEE4)
  Sample (6,Dst(2,1),IEEE4)
EndTable

BeginProg
  SerialOpen (ComC1,38400,0,0,0)
  Scan (10,Sec,0,0)
    PanelTemp (PTemp,60)
    Battery (Batt_volt)

    'Result,comport,neighbor,PBA,Dst,chan,muxchan,reps,begFreq,endFreq,Vx,
    'IntegrationTime,Mult,Offset
    'sensor 1, channel 1
    AVW200(Result,ComC1,200,200,Dst(1,1),1,1,1,1000,3500,2,_60HZ,1,0)
    'sensor 2, channel 2
    AVW200(Result,ComC1,200,200,Dst(2,1),2,1,1,1000,3500,2,_60HZ,1,0)

    CallTable AVW200
    CallTable AVWcard

  NextScan
EndProg
```

11.1.2 Wireless/one sensor/resistance converted to temperature

CRBasic Example: Wireless-One sensor

```
'This is an example of a program used by a CR1000X and AVW206 to one Geokon 4450
'VW displacement sensor. The sensor provides a frequency, which is converted to
'displacement, and resistance, which is converted to temperature. Polynomial Gage
'Factors used in this example were taken from the calibration sheets of the 4450
'sensor. The coefficients used to convert resistance to temperature are from the
'Steinhart-Hart equation.

'The CR1000X communicates with the remote AVW206 through an RF401A radio attached
'to the logger's CS/IO port in SDC7 mode.
'The Pakbus address of the AVW206 used in this example is 15.

Public batt_volt,Ptemp
Public VWvalues(6)
Public VWResults
Public Psi,Temp

Alias VWvalues(1) = Freq
Alias VWvalues(2) = Amp
Alias VWvalues(3) = Sig2Noise
Alias VWvalues(4) = NoiseFreq
Alias VWvalues(5) = DecayRatio
Alias VWvalues(6) = Therm

Dim Digits

'Below are coefficients for Steinhart-Hart equation used to convert resistance
'to Temp
Const A=.0014051
Const B=.0002369
Const C=.0000001019

BeginProg

  Scan (10,Sec,0,0)
    PanelTemp (Ptemp,60)
    Battery (batt_volt)

    AVW200(VWResults,ComSDC7,0,15,VWvalues(1),1,1,1,1000,2500,2,_60Hz,1,0)
    Digits = (Freq/1000)^2 * 1000 'Convert frequency to Digits

    'Convert resistance to temp F.
    Temp = (1/(A + B*LN(Therm) + C*(LN(Therm))^3)-273.15)*1.8+32
```

CRBasic Example: Wireless-One sensor

```
    'Calculate displacement (inches) from Digits and calibration polynomial
    Psi=2.49866e-10*Digits^2 + 8.716e-5*Digits + -.2

  NextScan
EndProg
```

11.2 AVW200() instruction controlling two multiplexers

Table 11-2 (p. 56) shows wiring used for this example. This program measures 16 sensors on each multiplexer.

Table 11-2: Wiring for an **AVW200()** instruction controlling two multiplexers

Data logger port for cable attachment	Cable needed to connect to AVW200	AVW200 port or model
ComC1 (control port pairs **C1**/**C2**)	DB9 pin-to-pigtail cable	RS-232

CRBasic Example 1: AVW200 controlling two multiplexers

```
'CR1000X Series Datalogger

'Declare Variables and Units
Public BattV
Public PTemp_C

'Public Temp_C
Public Data1, Data2
Public Mux1(16,6), Mux2(16,6)
Units BattV=Volts
Units PTemp_C=Deg C

'Define Data Tables
DataTable(VWTable1,True,-1)
  DataInterval(0,90,Sec,10)
  Sample (96,Mux1(),IEEE4)
  Sample (96,Mux2(),IEEE4)
  Minimum(1,BattV,FP2,False,False)
EndTable

'Main Program
BeginProg
  SerialOpen (ComC1,38400,0,0,0)
  Scan(90,Sec,1,0)
    AVW200(Data1(),ComC1, 200, 200, Mux1(1,1),1,1,16,450,3000,2,_60HZ,1,0)
    AVW200(Data2(),ComC1, 200, 200, Mux2(1,1),2,1,16,450,3000,2,_60HZ,1,0)
    Battery(BattV)
    PanelTemp(PTemp_C,60)
    CallTable(VWTable1)
  NextScan
EndProg
```

11.3 AVW200() instruction running in pipeline mode

The following program is an example of how to run the AVW200 with a CR1000Xe using multiple AVW200() instructions in the pipeline mode of operation. When this CRBasic program starts running, the information specified in the AVW200() instruction is sent to the attached AVW200 interface module via the **ComC1** communication port and PakBus address **200**. Along with the parameter information of the instruction, the CR1000Xe also sends its clock information. Every 64 seconds, the CR1000Xe clock and measured data are exchanged with the attached AVW200 interface module. The AVW200 interface module then determines how much time before the

CR1000Xe interval it should start making the vibrating-wire measurement so that the measurement series is completed before the CR1000Xe execution interval.

In the following example program, a multiplexer is attached to each AVW200 channel. Using the multiplexers, 16 sensors are measured on channel 1 and 16 sensors on channel 2. Both multiplexers will share the same clock (**CLK**) and reset (**RST**) lines on the AVW200.

In this example program, a returned `Result` value, which is the first parameter in the `AVW200()` instruction, of **0** indicates successful communications and storage of the data values in the destination variable, which is `AVWDst()` in this case. If communications fail, the `Result` value increments for each failure, and the `AVWDst()` values fill with **NAN**s. The data logger will retry communications three times before returning a failed communication or incrementing the result; retries are every 3 seconds or greater, depending on the radio-power-cycle configuration. A negative `Result` value indicates status information; for example, a **–3** indicates the AVW200 has not made the first measurement, or a **–4** indicates unavailable parameter information. Note that the `Result` variable in both `AVW200()` instructions are the same; different `Result` variables for a given communications port in the pipeline mode are unnecessary.

> NOTE:
> In the pipeline mode of operation, the `AVW200()` instruction cannot be in a conditional statement or in a subroutine. Placing the instruction at the top of the program is recommended.

Table 11-3 (p. 58) shows wiring used for this example.

Table 11-3: Wiring for `AVW200()` instruction running in pipeline mode

Data logger port for cable attachment	Cable needed to connect to AVW200	AVW200 port or model
ComC1 (control port pairs **C1**/**C2**)	DB9 pin-to-pigtail serial cable	RS-232

CRBasic Example 2: AVW200 running in pipeline mode

```
'Example Program running in the PipeLine mode
'The clock and reset lines of both muxes are connected to the clk and rst lines
'of the AVW200.

PipeLineMode

Public PTemp, batt_volt
Public Result, AVWDst(32,6)

Const Chan1 = 1 'AVW200 channel 1
Const Chan2 = 2 'AVW200 channel 2
Const MuxChan = 1 'Starting Mux Channel
Const Reps = 16 'Number of Reps
Const BFreq = 450 'Begin Frequency
Const EFreq = 6000 'End Frequency
Const Xvolt = 2 '12p-p Volt Excite

BeginProg
  SerialOpen (ComC1,38400,0,0,0)

  Scan (64,Sec,0,0) '(2 * 32 measurement) = 64 seconds
    PanelTemp (PTemp,60)
    Battery (batt_volt)
    AVW200(Result,ComC1,200,200,AVWDst
(1,1),Chan1,MuxChan,Reps,BFreq,EFreq,Xvolt,_60Hz,1,0)
    AVW200(Result,ComC1,200,200,AVWDst
(17,1),Chan2,MuxChan,Reps,BFreq,EFreq,Xvolt,_60Hz,1,0)
  NextScan
EndProg
```

11.4 AVW200() instruction running in sequential mode

The examples in AVW200 controlling two multiplexers in sequential mode (p. 61) and Data logger controlling two multiplexers in sequential mode (p. 61) run the AVW200 with a CR1000Xe using multiple AVW200() instructions in a sequential mode of operation. In these example programs, the first AVW200() instruction communicates with the attached AVW200 interface module using **ComC1** and PakBus address **200** and tells the AVW200 to make 16 measurements. The CR1000Xe will wait at the AVW200() instruction until the 16 measurements are returned from first instruction or until the instruction times out, which occurs after two times the number of reps. If an instruction times out, it will try up to three more times before advancing to the next instruction. The next AVW200() instruction on channel 2 is then executed and repeats the process.

Each `AVW200()` instruction in the examples uses a different `Result` code variable. In sequential mode, it is a good idea to have different `Result` code variables for each `AVW200()` instruction.

> **NOTE:**
> 1. When running in sequential mode, programs that contain multiple `AVW200()` instructions using the same **COM** port should have different `Result` variables for each `AVW200()` instruction, such as `Result1` and `Result2`, in order to detect and isolate any communications errors for a given AVW200.
> 2. If the `AVW200()` instruction is in a slowsequence, the mode of operation is always forced to sequential mode.

Table 11-4 (p. 60) shows the wiring used for both sequential mode examples detailed in AVW200 controlling two multiplexers in sequential mode (p. 61) and Section Data logger controlling two multiplexers in sequential mode (p. 61).

Table 11-4: Wiring for sequential mode examples

Data logger port for cable attachment	Cable needed to connect to AVW200	AVW200 port or model
ComC1 (control port pairs **C1**/**C2**)	DB9 pigtail-to-pin cable	RS-232

11.4.1 AVW200 controlling two multiplexers in sequential mode

CRBasic Example 3: AVW200 running in sequential mode and controlling 2 multiplexers

```
'Example Program running in the Sequential mode with AVW200 controlling
'2 muxes. The clock and reset lines of both muxes are connected to the clk
'and rst lines of the AVW200.

SequentialMode

Public PTemp, batt_volt
Public Result1, Result2, AVWDst(32,6)

Const Chan1 = 1 'AVW200 channel 1
Const Chan2 = 2 'AVW200 channel 2
Const MuxChan = 1 'Starting Mux Channel
Const Reps = 16 'Number of Reps
Const BFreq = 450 'Begin Frequency
Const EFreq = 6000 'End Frequency
Const Xvolt = 2 '12p-p Volt Excite

BeginProg
  SerialOpen (ComC1,38400,0,0,10000)
  Scan (64,Sec,0,0) '(2 * 32 measurement) = 64 seconds
    PanelTemp(PTemp,60)
    Battery (Batt_volt)
    AVW200(Result1,ComC1,200,200,AVWDst
(1,1),Chan1,MuxChan,Reps,Bfreq,Efreq,Xvolt,_60Hz,1,0)
    AVW200(Result2,ComC1,200,200,AVWDst
(17,1),Chan2,MuxChan,Reps,Bfreq,Efreq,Xvolt,_60Hz,1,0)
  NextScan
EndProg
```

11.4.2 Data logger controlling two multiplexers in sequential mode

The following example program has the data logger controlling the multiplexers. Note that the `AVW200()` parameter for reps is 1, and *x* is an index variable in `AVWDst(x,1)`.

CRBasic Example: Data logger controlling 2 multiplexers

```
'Example Program running in the Sequential mode with the data logger
'controlling the muxes. For this program, the reset line of both muxes is
'connected to data logger C3. Mux1 clock line is connected to DL C4 and Mux2
'clock line is connected to DL C5.

SequentialMode
Public PTemp, batt_volt, x
Public Result1, Result2, AVWDst(32,6)

Const Chan1 = 1 'AVW200 channel 1
Const Chan2 = 2 'AVW200 channel 2
Const MuxChan = 1 'Starting Mux Channel
Const Reps = 1 'Number of Reps
Const BFreq = 450 'Begin Frequency
Const EFreq = 6000 'End Frequency
Const Xvolt = 2 '12p-p Volt Excite

Dim tmpavw200(6)

BeginProg
  SerialOpen (ComC1,38400,0,0,0)
  Scan (64,Sec,0,0) '(2 * 32 measurement) = 64 seconds
    PanelTemp (PTemp,60)
    Battery (Batt_volt)
    PortSet(C3, 1) 'Reset High Mux On, both muxes share the RST
    Delay(1, 100, mSec) 'before clocking

    For x = 1 To 16
      PulsePort(C4, 2000) 'Advance Mux #1 on C4 port (clock high for 2mSec)
      PulsePort(C5, 2000) 'Advance Mux #2 on C5 (clock high for 2mSec)
      Delay(1, 10, mSec) 'Mux Settling Time
      AVW200(Result1,ComC1,200,200,tmpavw200
(1),Chan1,MuxChan,Reps,Bfreq,Efreq,Xvolt,_60Hz,1,0)
      Move(AVWDst(x,1),6,tmpavw200(1),6) 'now copy tmp value to the Dst
      AVW200(Result2,ComC1,200,200,tmpavw200
(1),Chan2,MuxChan,Reps,Bfreq,Efreq,Xvolt,_60Hz,1,0)
      Move(AVWDst(x+16,1),6,tmpavw200(1),6) 'now copy tmp value to the Dst
    Next
    PortSet(C3, 0) 'Reset Low Mux off
  NextScan
EndProg
```

11.5 SDI-12 example

The following program is an example of how to run the AVW200 with a CR1000Xe using the `SDI12Recorder()` instruction. Note that the values returned from the `SDI12Recorder()` instruction are different depending on the SDI-12 measurement command issued.

For this example, two multiplexers are measured by the data logger. The AVW200 interface module cannot control multiplexers in the SDI-12 communications mode. Hence, when communicating to the AVW200 via SDI-12, any multiplexers attached to the AVW200 must be controlled by the data logger. This is achieved by using `PortSet()` instructions in the data logger program (see the following example) and by connecting the clock and reset lines of the multiplexers to control ports on the data logger. When using SDI-12 with the AVW200, the clock and reset lines of the AVW200 are unused.

Extended SDI-12 commands can be used to change the begin frequency, end frequency, and excitation voltage of the vibrating-wire sensors attached to the AVW200. However, these extended SDI-12 commands only work for the next measurement command. By default, standard SDI-12 measurement commands use the begin/end/excite voltage settings specified in the AVW200 settings using ***Device Configuration Utility***. However, after issuing an extended SDI-12 command, the very next measurement will use the *bbbb*, *eeee*, and *v* values specified in the extended command. The second and remaining measurements will revert to the settings specified via ***Device Configuration Utility***.

An example of an extended command is `0XVW450,5000,1!`. This command will configure the next measurement with begin freq = 450, end freq = 5000, and 5 VDC excitation. Following the extended command, `aM1!` will use the parameters set by the extended command.

Table 11-5 (p. 64) shows the specific SDI-12 command codes and their returned values.

Table 11-5: SDI-12 command codes	
SDI-12 measurement command	**Returned values**
`aM!` (measures both AVW200 channel 1 and channel 2)	1) Chan1 frequency 2) Chan1 therm resistance 3) Chan1 signal amplitude mV RMS 4) Chan2 frequency 5) Chan2 therm resistance 6) Chan2 signal amplitude mV RMS
`aM1!` (measures AVW200 channel 1)	1) Chan1 Frequency 2) Chan1 signal amplitude mV RMS 3) Chan1 signal-to-noise ratio 4) Chan1 noise frequency 5) Chan1 decay ratio 6) Chan1 therm resistance
`aM2!` (measures AVW200 channel 2)	1) Chan2 frequency 2) Chan2 signal amplitude mV RMS 3) Chan2 signal-to-noise ratio 4) Chan2 noise frequency 5) Chan2 decay ratio 6) Chan2 therm resistance
`aV!`	1) Battery voltage 2) Trapcodes 3) Watchdog counts
`aXVWbbbb,eeee,v!` where: bbbb = Begin freq (100 < bbbb < 6.5K) eeee = End freq (100 < eeee < 6.5K) v = Excite voltage (1= 5v VDC and 2 = 12 VDC)	Returns OK if accepted Returns ERROR if not accepted

CRBasic Example 4: SDI-12 - data logger controlling 2 multiplexers

```
'Example Program running SDI12 commands with the data logger controlling
'2 mux's. For this program, the AVW SDI-12 port is connected to DL C1.
'The reset line of both muxes is connected to data logger C3. Mux1 clock line
'is connected to DL C4 and Mux2clock line is connected to DL C5. The SDI-12
'address of the AVW200 is set to 1.

SequentialMode

Public PTemp, batt_volt
Public Chan1_Val(16,6),Chan2_Val(16,6)

Dim I

BeginProg
  Scan (150,Sec,0,0)
    PanelTemp(PTemp,60)
    Battery(batt_volt)
    PortSet(C3, 1) 'Reset High, Mux On, both muxes share this reset port
    Delay(1, 100, mSec) 'delay before clocking

    '************ Measure 16 vibrating wire sensor on AVW200 channel 1
    For I=1 To 16

      'Advance Mux #1 (clock line connected to C4; clock high for 2mSec)
      PulsePort(C4, 2000)
      Delay(1, 10, mSec) 'Mux Settling Time

      'measures sensor on channel 1
      SDI12Recorder (Chan1_Val(I,1),C1,1,"M1!",1.0,0)
    Next I

    '************ Measure 16 vibrating wire sensor on AVW200 channel 2
    For I=1 To 16

      'Advance Mux #2 (clock line connected to C5; clock high for 2mSec)
      PulsePort(C5, 2000)
      Delay(1, 10, mSec) 'Mux Settling Time

      'measures sensor on channel 2
      SDI12Recorder (Chan2_Val(I,1),C1,1,"M2!",1.0,0)
    Next I

    PortSet(C3, 0) 'Reset Low; turn both Muxes off
  NextScan
EndProg
```

12. Troubleshooting communications problems

12.1 Unable to communicate with *Device Configuration Utility* or terminal emulator

If you are unable to communicate with ***Device Configuration Utility*** or the terminal emulator, verify that:

1. The AVW200 is powered and, when the device is turned on, its front red LED remains lit for 15 seconds then blinks intermittently.
2. The correct **COM** port has been selected. The **COM** port entry is provided on the lower-left corner of the ***Device Configuration Utility*** screen.
3. The correct baud rate of the AVW200 has been selected; default is 38400.

12.2 Data logger to AVW200 communications

If the data logger fails to communicate with the AVW200, verify that:

1. The AVW200 is powered and, when the device is turned on, its front red LED remains lit for 15 seconds then blinks intermittently.
2. The AVW200 PakBus address is different than the PakBus address of the data logger.
3. The AVW200 PakBus address is entered correctly in the `AVW200()` instruction of the data logger program.

12.3 Wireless communications

If you are unable to connect, consider the following possible causes:

1. Turned off or not powered

   Verify that the AVW200 is powered. The red LED at the front of the AVW200 will remain lit for 15 seconds after it is turned on then blink intermittently.

2. Active interface set wrong

   The active interface on the radio attached to the data logger running the `AVW200()` instruction must match the `ComPort` specified in the `AVW200()` instruction. For example, if you are using an RF401A (configured for SDC7) attached to a data logger to communicate with a remote AVW206, then the `ComPort` specified in the `AVW200()` instruction must be SDC7 (or whatever active interface the RF401A is set for).

3. Low or weak battery voltage or 12 VDC supply voltage

   The power supply battery may not be charging properly due to solar panel orientation, poor connection, or a charging transformer problem. The battery itself may have discharged too low too many times, ruining the battery. Lead acid batteries perform best when they are topped off.

   Power supply must be able to sustain a data logger minimum of at least 9.6 VDC, even during 75 mA transmitter bursts lasting only a few milliseconds.

4. Lightning damage to RF401A or AVW206

   Swap in a known good RF401A or AVW206 with the same settings and see if this cures the problem. Lightning damage can occur with no visible indications. Both direct hits and near misses can cause damage, with or without evidence of smoke .

5. Lightning damage to antenna and/or cable

   Swap in a known good antenna and/or cable. Hidden damage may exist.

6. Moisture in coaxial antenna cable

   It is possible that moisture has penetrated inside the plastic sheath of the coaxial cable. Water inside the cable can absorb RF energy and attenuate the transmitted signal, making the received signal attenuated also. Drying out the interior of a coaxial cable is difficult, so substitution of a dry cable is recommended.

   Placing a wet cable in a conventional oven at 160 °F for a couple hours should dry it out. Shield the antenna cable against damage from radiated heat from the oven element by placing the coiled cable on a large cookie sheet or a sheet of aluminum foil.

7. AVW206 receiver overwhelmed or de-sensed by nearby transmitter

   This problem can be observed from LED behavior when operating a hand-held radio near an AVW206 that is receiving collected data from a remote station. If you key a hand-held 150 MHz or 450 MHz transmitter, even though its frequency of operation is far removed from the 900 MHz band, its proximity to the AVW206 can overwhelm or de-sense the AVW206 receiver, resulting in failed packets and LoggerNet retries. This problem could also occur if you located an AVW206 at a site containing commercial transmitters or repeaters. In general, avoiding such sites is best, especially the high-power FM or AM transmitter antenna sites that can change at any time with added equipment.

It is possible to avoid de-sensing in some cases if the RF link is solid enough due to the proximity of your remote AVW206(s), high antenna gains and directionality, high elevation, and sufficient distance separation between AVW206 and commercial transmitter antenna. Try horizontal polarization of antennas. A field test in such situations is essential.

8. Insufficient signal strength

   You can try some adjustments to get a few extra dBs of signal strength, which are sometimes necessary for a dependable RF link. The drop in signal going from winter season, with no deciduous tree leaves, to spring season, with more leaves, sometimes requires a little more signal.

   a. Raise the antenna height using a mast, tower or higher terrain. Often, a little extra height makes the difference.
   b. Change to a higher-gain antenna.
   c. Change the polarization or element orientation of all Yagi and omnidirectional antennas in your network from vertical to horizontal or vice versa.

9. Interference from 900 MHz transmitter

   Some measures you can take to reduce interference from neighboring 900 MHz transmitters are as follows:

   a. Move base station as far as possible from offending transmitter antenna.
   b. Install a 9 dBd Yagi and position station so that the offending transmitter is located behind or to the side of the Yagi to take advantage of the Yagi front-to-back or front-to-side ratio, also referred to as back and side signal rejection.
   c. Change the polarization or element orientation of all Yagi and omnidirectional antennas in your network to see if that reduces the effects of the offending transmitter.

10. AVW206 or other radio in the network has the wrong network address, radio address, hopping sequence, or standby mode

    It is improbable that an RF401A or AVW206 network that has been working would ever change address, hopping sequence, or other settings. However, check the settings for the unlikely event that this may have happened. Try **Restore Defaults** and set up RF401A or AVW206 again from that point.

Appendix A. Conversion from Hz

The calibration report provided with each vibrating-wire sensor contains the information required to convert Hz, the frequency value output by the AVW200, to the appropriate units, such as displacement or pressure.

These steps convert Hz to the appropriate unit, such as displacement or pressure:

1. If the values in the calibration report are in digits, use the following equation to convert the AVW200 frequency values from Hz to digits:

   $Digits = Hz^2/1000$

2. Use the gauge factors and polynomial provided in the calibration report to calculate displacement.

A.1 Displacement example

Figure A-1 (p. 70) provides a calibration report for a displacement transducer. The following steps convert the value output by the AVW200 in Hz to displacement.

1. Convert Hz to digits:

   $Digits = Freq^2/1000$

2. Displacement = (3.598e–9) × Digits2 + (1.202e–3) × Digits + (–3.1682)

   Therefore,

   if Freq = 2400, then:

   $digits = 2400^2/1000 = 5760$

   displacement = (3.598e–9) × (5760)2 + (1.202e–3) × 5760 + (–3.1682) = 3.875 inches

GEOKON 48 Spencer St. Lebanon, N.H. 03766 USA

Vibrating Wire Displacement Transducer Calibration Report

Range: 150 mm

Calibration Date: January 7, 2010

Serial Number: 0939696

Temperature: 23.4 °C

Calibration Instruction: CI-4400

Technician:

GK-401 Reading Position B

Actual Displacement (mm)	Gage Reading 1st Cycle	Gage Reading 2nd Cycle	Average Gage Reading	Calculated Displacemen (Linear)	Error Linear (%FS)	Calculated Displacement (Polynomial)	Error Polynomial (%FS)
0.0	2614	2613	2614	-0.30	-0.20	-0.03	-0.02
30.0	3580	3579	3580	30.08	0.05	30.03	0.02
60.0	4539	4539	4539	60.26	0.18	60.04	0.03
90.0	5492	5489	5491	90.19	0.13	89.98	-0.02
120.0	6437	6439	6438	120.00	0.00	119.95	-0.04
150.0	7385	7383	7384	149.76	-0.16	150.03	0.02

(mm) Linear Gage Factor (G): 0.03146 (mm/ digit) **Regression Zero: 2623**

Polynomial Gage Factors: A: 9.139E-08 B: 0.03054 C: -80.471

(inches) Linear Gage Factor (G): 0.001238 (inches/ digit)

Polynomial Gage Factors: A: 3.598E-09 B: 0.001202 C: -3.1682

Calculated Displacement: **Linear, $D = G(R_1 - R_0)$**

Polynomial, $D = AR_1^2 + BR_1 + C$

Refer to manual for temperature correction information.

Function Test at Shipment:

GK-401 Pos. B : 4239 Temp(T_0): 24.1 °C Date: January 28, 2010

The above instrument was found to be in tolerance in all operating ranges.

The above instrument was found to be in tolerance in all operating ranges.

The above named instrument has been calibrated by comparison with standards traceable to the NIST, in compliance with ANSI Z540-1.

Figure A-1. Geokon calibration report of a sensor without a thermistor

Appendix B. Thermistor information

B.1 Converting resistance to temperature

The AVW200 outputs a resistance value for sensors that contain a thermistor. Temperature is calculated by applying the resistance to a known equation, such as the Steinhart-Hart equation, which converts resistance to temperature.

The Steinhart-Hart equation for converting resistance to degrees Celsius is as follows:

Temperature = 1/(A + B × LN(resistance) + C × (LN(resistance))^3) – 273.15

where A, B, and C are coefficients for the Steinhart-Hart equation.

The coefficients for the Steinhart-Hart equation are specific to the thermistor contained in your sensor and are obtained from the sensor manufacturer.

> **NOTE:**
> Please check with the manufacturer to get the coefficients for your specific thermistor.

B.1.1 Resistance conversion example – Geokon sensor

If the coefficients for the Steinhart-Hart equation are as follows:

A = 0.0014051
B = 0.0002369
C = 0.0000001019

then the equation for converting the resistance measurement to degrees Celsius is:

Temperature = 1/(0.0014051 + 0.0002369 × LN(resistance) + 0.0000001019 × (LN(resistance))^3) – 273.15

If the measured resistance is 2221 ohms, the calculated temperature in degrees Celsius is:

Temperature = 1/(0.0014051 + 0.0002369 × LN(2221) + 0.0000001019 × (LN(2221))^3) – 273.15

Temperature = 31.98 °C

B.2 Accuracy and resolution

The accuracy of the temperature measurement is a function of the following factors:

1. Thermistor interchangeability
2. Resistance of the wire
3. Steinhart-Hart equation error
4. Precision of the bridge resistors
5. Accuracy of the data logger voltage measurement
6. Temperature coefficient of the bridge resistors

Errors three through six can probably be ignored. The wire resistance is primarily an offset error, and its effect can be removed by the initial calibration. Errors caused by the change in wire resistance due to temperature and thermistor interchangeability are not removed by the initial calibration. Figure B-1 (p. 73) through Figure B-4 (p. 75) show how wire resistance affects the temperature measurement for a Geokon 4500 vibrating-wire piezometer.

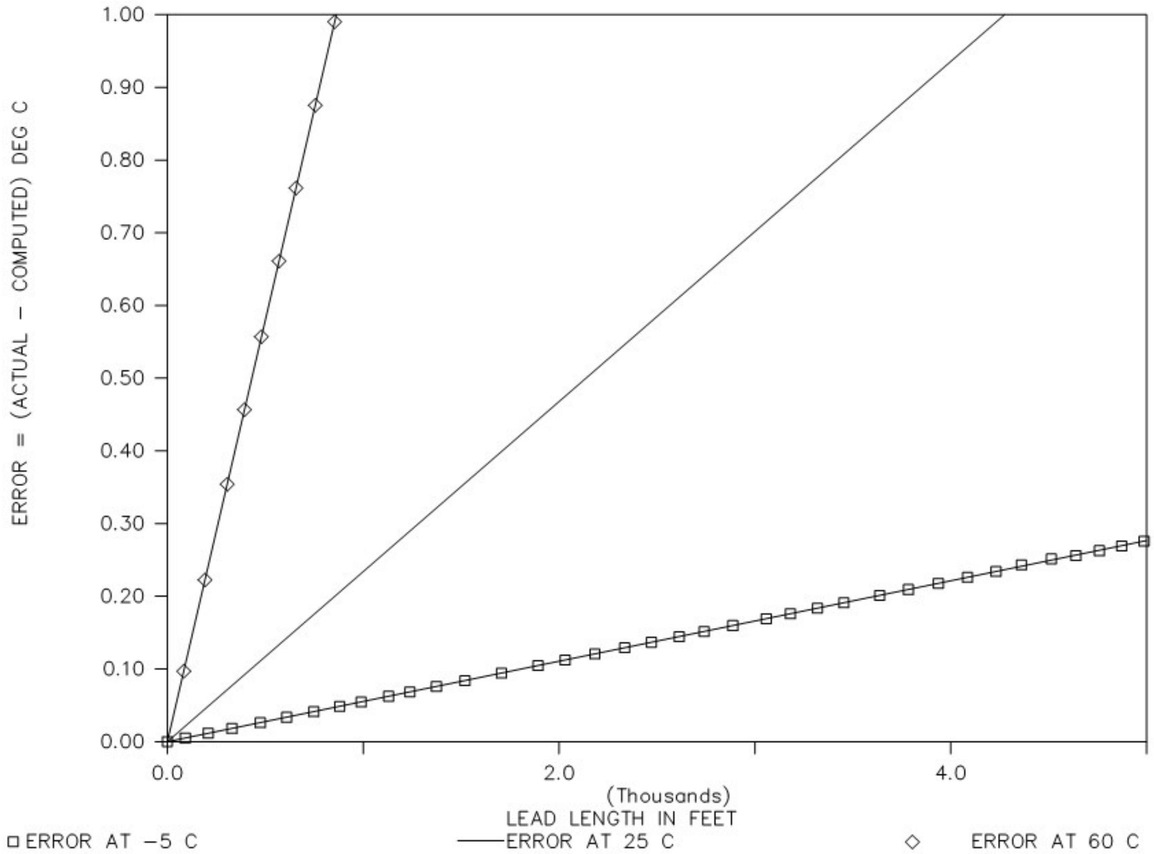


Figure B-1. Temperature measurement error at three temperatures as a function of wire length; wire is 22 AWG with 16 ohms per 1000 ft

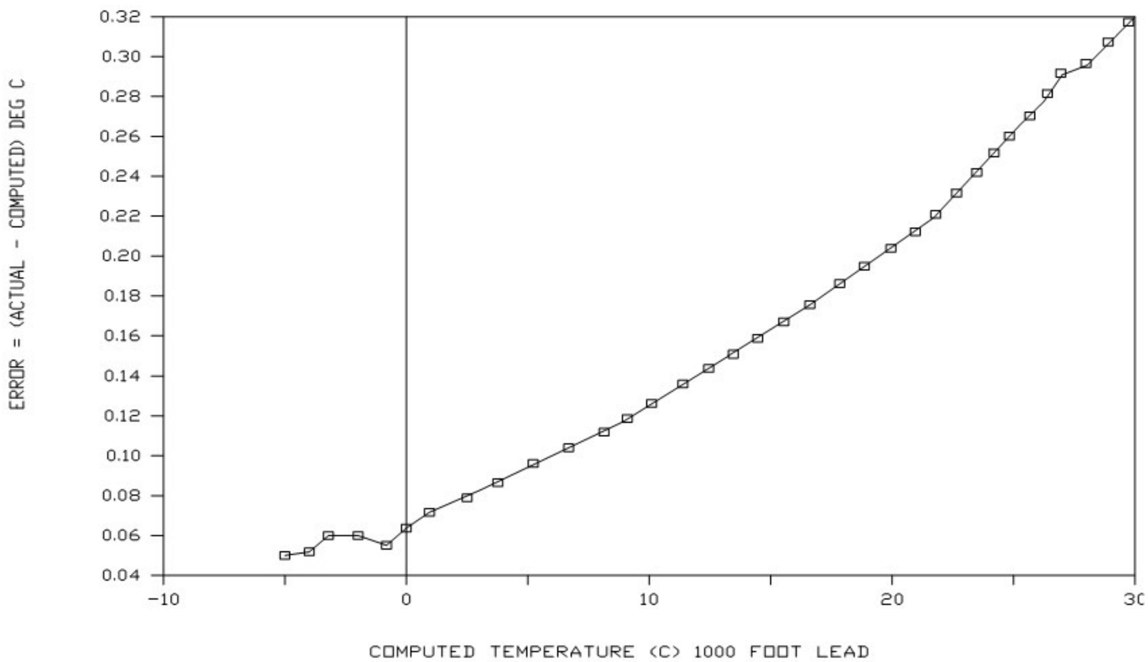


Figure B-2. Temperature measurement error on a 1000-foot wire; wire is 22 AWG with 16 ohms per 1000 ft

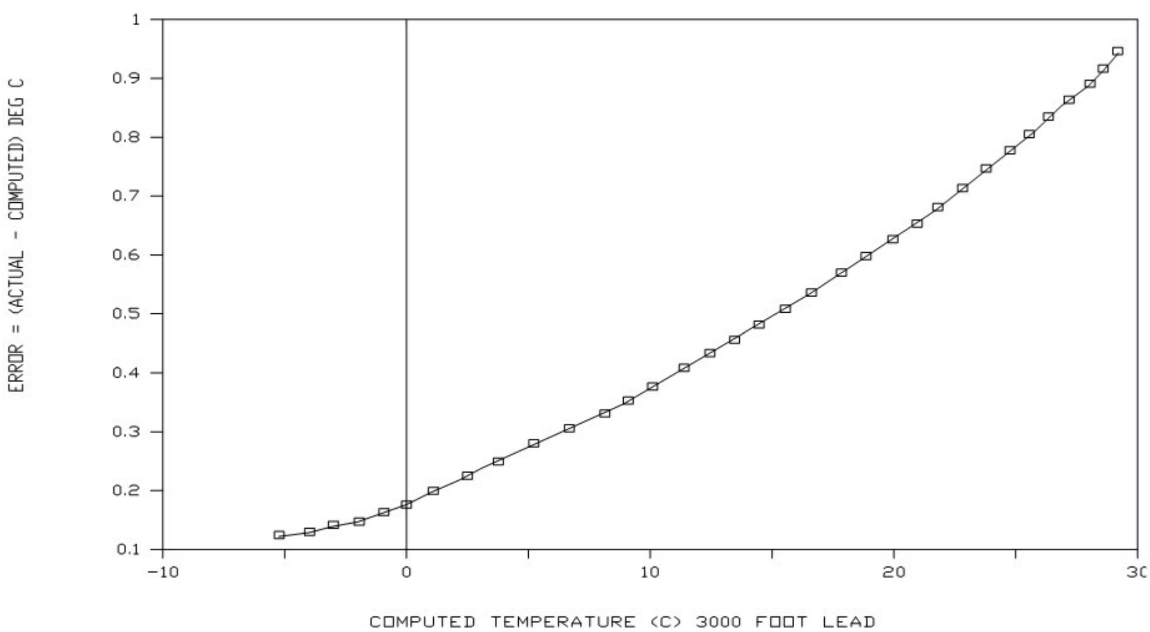


Figure B-3. Temperature measurement error on a 3000-foot wire; wire is 22 AWG with 16 ohms per 1000 ft

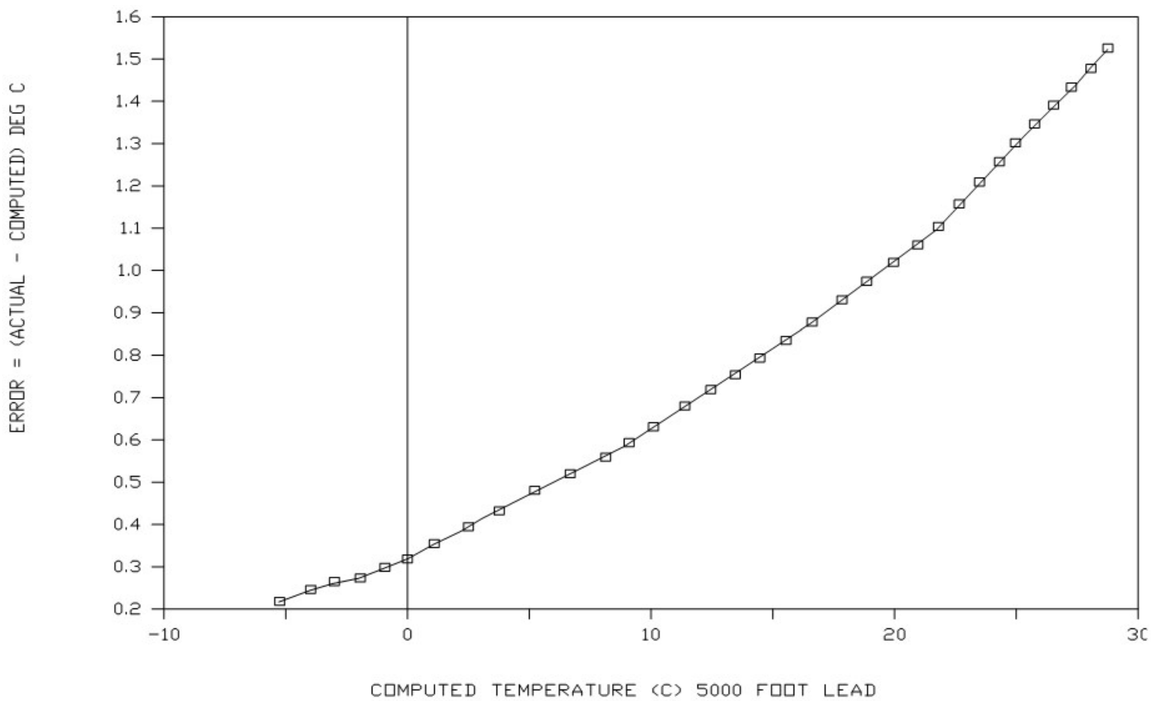


Figure B-4. Temperature measurement error on a 5000-foot wire; wire is 22 AWG with 16 ohms per 1000 ft

Appendix C. Antennas, antenna cables, and surge protectors for the AVW206, AVW211, and AVW216

C.1 Antenna cables

The 9 dBd Yagi, 3 dBd omnidirectional, 6 dBd Yagi, and 13 dBd Yagi antennas each require an antenna cable, either the COAXRPSMA or the COAXNTN with surge protector. Indoor omnidirectional antennas are either supplied with an appropriate cable or connect directly to the AVW206.

C.2 Surge protectors

C.2.1 Electrostatic issues

Many installations are outside and thus susceptible to lightning damage, especially via the antenna system. Also, depending on climate and location, electrostatically charged wind can damage sensitive electronics if sufficient electric charge is allowed to accumulate on the antenna and cable. To protect against this, Campbell Scientific offers a surge suppressor kit.

The COAXNTN-L cable is a low-loss RG8 coaxial cable that requires the surge suppressor kit to connect to the AVW206. This cable-and-surge-suppressor combination is recommended over the COAXRPSMA in the following applications:

- When the antenna cable length exceeds 10 feet
- When use of COAXRPSMA would result in too much signal loss
- When the interface will be used in an environment susceptible to lightning or electrostatic buildup

C.2.1.1 Antennas

Several antennas are offered to satisfy various base station and remote station requirements. These antennas have been tested at an authorized FCC open-field test site and are certified to comply with FCC emissions limits. All antennas or antenna cables have an SMA socket connector for connection to the AVW206. The use of an unauthorized antenna could cause field transmission strengths greater than what FCC allows, interfere with licensed services, and result in FCC sanctions against the user.

> **NOTE:**
> An FCC authorized antenna is needed for wireless communications. The 900 MHz antennas are compatible with the AVW206 and AVW211. The 2.4 GHz antennas are compatible with the AVW216. Pick one of the antennas listed below.

CSI item number	Description
14310	0 dBd antenna, 900 MHz, omni ¼ wave whip, RPSMA straight, Linx, 3.2 inches long.
14204	0 dBd antenna, 900 MHz, omni ½ wave whip, RPSMA rt angle, Astron, 6.75 inches long.
14221	3 dBd antenna, 900 MHz, omni collinear, Antenex FG9023, 24 inches tall, W/FM2 mounts, fits 1 in. to 2 in. O.D. mast (requires COAX RPSMA-L or COAX NTN-L)
15970	1 dBd antenna, 900 MHz, indoor omni ½ wave dipole, 10 ft. cable with SMA connector to fit RF401A Series, window or wall mounted by sticky back, 4 inches wide.
14205	6 dBd antenna, 900 MHz, Yagi, Larsen YA6900 TYPE N-F, boom length 17.25 inches, longest element 7.25 inches, w/mounts, fits 1 in. to 2 in. O.D. mast (requires COAX RPSMA-L or COAX NTN-L)
14201	9 dBd antenna, 900 MHz, Yagi, MAXRAD BMOY8905 TYPE N-F, boom length 21.4 inches, longest element 6.4 inches, w/mounts, fits 1 in. to 2 in. O.D. mast (requires COAX RPSMA-L or COAX NTN-L)
16005	0 dBd antenna, 2.4 GHz, omni ½ wave whip, RPSMA rt angle, LINX ANT-2.4-CW-RCT-RP, 4.5 inches long.

CSI item number	Description
16755	13 dBd antenna, 2.4 GHz, enclosed Yagi, allows vertical or horizontal polarization, MAXRAD WISP24015PTNF, boom length 17 inches, diameter 3 inches, w/end mount to fit 1 to 2 in. O.D. mast (requires either (1) COAX RPSMA-L for short runs or (2) COAX NTN-L with Antenna Surge Protector Kit)
COAX RPSMA-L	LMR 195 antenna cable, reverse polarity SMA to type N pin
COAX NTN-L	RG8 antenna cable, type N pin to type N pin connectors, requires 14462
14462	Antenna Surge Protector Kit

FCC OET Bulletin No. 63 (October 1993)

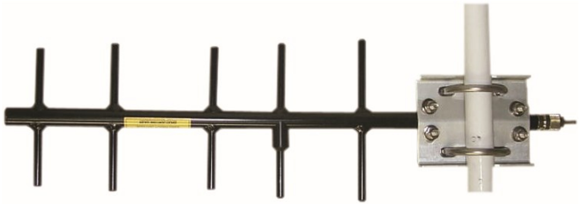
Changing the antenna on a transmitter can significantly increase, or decrease, the strength of the signal that is ultimately transmitted. Except for cable locating equipment, the standards in Part 15 are not based solely on output power but also take into account the antenna characteristics. Thus, a low power transmitter that complies with the technical standards in Part 15 with a particular antenna attached can exceed the Part 15 standards if a different antenna is attached. Should this happen it could pose a serious interference problem to authorized radio communications such as emergency, broadcast, and air-traffic control communications.

> **CAUTION:**
> In order to comply with FCC RF exposure requirements, the AVW206 series may be used only with approved antennas that have been tested with this radio, and a minimum separation distance of 20 cm must be maintained from the antenna to any nearby persons.

See Part 15 FCC compliance warning (p. 81) for important FCC information.

900 MHZ OMNI ¼ WAVE WHIP 0 dBd

900 MHZ OMNI ½ WAVE WHIP 0 dBd



900 MHZ YAGI 9 dBd w/MOUNTS

900 MHz YAGI 6 dBd w/MOUNTS

900 MHZ OMNI COLLINEAR 3 dBd w/MOUNTS

900 MHZ Indoor OMNI 1 dBd Window/Wall Mounted

2.4 GHz OMNI HALF WAVE WHIP 0 dBd

2.4 GHz ENCLOSED YAGI, 13 dBd w/MOUNTS

Figure C-1. Some FCC-approved antennas

Figure C-2. Example COAX RPSMA-L cable for Yagi or omnidirectional

Figure C-3. Antenna surge protector

C.2.2 Surge suppressor kit

The surge suppressor kit for the RF401A series radios includes the following:

- Polyphaser protector
- 18 inches of COAXRPSMA to connect the surge protector tail end to the RF401A
- Ground wire
- Screw and grommet to secure ground wire and polyphaser to backplate of enclosure

The surge suppressor has type N socket connectors on both ends, one for COAXNTN-L cable connection, and the other for connection to the 18-in. length of COAXRPSMA cable. The COAXRPSMA cable is an LMR195 type that terminates in a type N pin connector on the antenna end and a reverse polarity SMA (RPSMA) connector on the RF401A end.

C.3 Part 15 FCC compliance warning

Changes or modifications to the AVW206, AVW211, or AVW216 not expressly approved by Campbell Scientific, Inc. could void the authority of the user to operate this product.

Note: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:

1. This device may not cause harmful interference, and
2. This device must accept any interference received, including interference that may cause undesired operation.

Appendix D. Public table

The **Public** table of the AVW200 displays the current sensor measurement values as well as the current settings (see Table D-1 [p. 83]).

When ***Device Configuration Utility*** runs the troubleshooter, it forces a measurement by writing to the `Timeseries()` array in the **Public** table. When the `Timeseries(1)` through `Timeseries(4)` elements are written with the proper values, a measurement is performed, and the files **Timeseries.bin** and **Spectrum.bin** are created or overwritten if previous measurements have been forced. These files are then used by ***Device Configuration Utility*** to create the time domain and frequency domain graphs in the troubleshooter.

A data logger program can also force a measurement by using the `GetVariable()` and/or `SetVariable()` instructions (see Forced measurement program [p. 86]). To do this, use the CRBasic `SetVariable()` instruction to set the `TimeSeries(2)`, `TimeSeries(3)`, and `TimeSeries(4)` variables. Once these variables have the correct values for begin frequency, end frequency, and excite voltage, the measurement can be forced by writing `TimeSeries(1)` with the AVW200 channel and mux channel (for example, 101 = AVW200 Chan1 and MuxChan1, 205 = AVW200 Chan2 and MuxChan5, or 208 = AVW200 Chan2 and MuxChan8). The measurement is completed when the `TimeSeries(1)` value is zeroed by the AVW200. Once zeroed, the `TimeSeries(5)` through `TimeSeries(11)` values can be read using the CRBasic `GetVariable()` instruction.

> NOTE:
> `TimeSeries(5)` through `TimeSeries(11)` were added to OS version STD.02. Customers with OS version STD.01 can get these `Public` variables by downloading a new OS (refer to Send OS [p. 38]).

Table D-1: Description of the Public table	
Record no	Current record number for this set of data
Time stamp	Time the record was recorded
Control parameters for AVW200 instruction communications	
Result	Result of the last measurement
NumReps	Number of replications specified for this channel
Chan	Channel from which this record was recorded

Table D-1: Description of the Public table	
MuxChan	Multiplexer channel from which this data was recorded
BeginFreq	Begin frequency used for this measurement
EndFreq	End frequency used for this measurement
ExciteVolts	Excitation voltage used for this measurement
Do_Therm	Do the thermistor measurement or not
RScnt	Countdown to start of measurement for the **RS-232** communications port; **–1** if no interval or not used
RFcnt	Countdown to start of measurement for the **RF** communications port; **–1** if no interval or not used
Values and control parameters for SDI-12 communications	
SDI12val(1)	Frequency value obtained by SDI recorder instruction, if used; unchanged if not using SDI-12
SDI12val(2)	Amplitude value obtained by SDI recorder instruction, if used; unchanged if not using SDI-12
SDI12val(3)	Signal-to-noise ratio obtained by SDI recorder instruction, if used; unchanged if not using SDI-12
SDI12val(4)	Signal-to-noise frequency obtained by SDI recorder instruction, if used; unchanged if not using SDI-12
SDI12val(5)	Decay ratio obtained by SDI recorder instruction, if used; unchanged if not using SDI-12
SDI12val(6)	Thermistor reading obtained by SDI recorder instruction, if used; unchanged if not using SDI-12
SDIChan	Channel used for SDI-12 measurement
SDIBFreq	Begin frequency used for SDI-12 measurement
SDIEFreq	End frequency used for SDI-12 measurement
SDIExVolt	Excitation voltage used for SDI-12 measurement
Value of last instruction running	
Ch1Freq	Frequency value measured on channel 1
Ch1Amp	Amplitude value measured on channel 1

Table D-1: Description of the Public table	
Ch1SNR	Signal-to-noise ratio measured on channel 1
Ch1SNRFreq	Signal-to-noise frequency measured on channel 1
Ch1Decay	Decay ratio measured on channel 1
Ch1Therm	Thermistor reading measured on channel 1
Ch2Freq	Frequency value measured on channel 2
Ch2Amp	Amplitude value measured on channel 2
Ch2SNR	Signal-to-noise ratio measured on channel 2
Ch2SNRFreq	Signal-to-noise frequency measured on channel 2
Ch2Decay	Decay ratio measured on channel 2
Ch2Therm	Thermistor reading measured on channel 2
Control parameters when troubleshooter is running	
TimeSeries(1)	Writing this variable will force a vibrating-wire measurement and create the TimeSeries.bin and Spectrum.bin files. Example: 101 = measures AVW200 chan1 and Mux chan1 102 = measures AVW200 chan1 and Mux chan2 ..... 201 = measures AVW200 chan2 and Mux chan1 202 = measures AVW200 chan2 and Mux chan2 .... 232 = measures AVW200 chan2 and Mux chan32 Note: TimeSeries(2)..(4) require valid values before making the measurement
TimeSeries(2)	Begin frequency for force measurement (100 – 6500)
TimeSeries(3)	End frequency for forcing measurement (100 – 6500)
TimeSeries(4)	Excite voltage for forcing measurement (1 = 5 VDC or 2 = 12 VDC)
TimeSeries(5)	Copy of the Chan/MuxChan executed in TimeSeries(1)
TimeSeries(6)	Measured frequency (Hz)

Table D-1: Description of the Public table	
TimeSeries(7)	Measured amplitude (MVolts_RMS)
TimeSeries(8)	Measured signal-to-noise ratio
TimeSeries(9)	Measured noise frequency (Hz)
TimeSeries(10)	Measured decay ratio
TimeSeries(11)	Measured thermistor (ohms)

D.1 Forced measurement program

CRBasic Example 5: CRBasic Forced Measurement Program

```
SequentialMode
Public UsrForcedMsmnt
Public SVResult(2), GVResult(2), TimeSeries(11)
Dim TS_done

BeginProg
  TimeSeries(1) = 101  'Measure command with XYY as described below.
  'X is the AVW channel, 1 or 2, and YY is the multiplexer channel, 00-32
  TimeSeries(2) = 450  'Sweep start frequency, 450 Hz minimum.
  TimeSeries(3) =6500  'Sweep stop frequency, 6500 Hz maximum.
  TimeSeries(4) = 1  'Excitation level code, 0=5Volt, 1=12Volt.
  'TimeSeries(5)  'Echo of what was used For TimeSeries(1) measure command, XYY.
  'TimeSeries(6)  'Frequency of peak, Hz.
  'TimeSeries(7)  'Amplitude of peak frequency, mVrms.
  'TimeSeries(8)  'SNR, Signal To Noise Ratio.
  'TimeSeries(9)  'Peak noise frequency, Hz.
  'TimeSeries(10)  'Decay ratio.
  'TimeSeries(11)  'Thermistor measurement, Ohms.
  SerialOpen (Com1,38400,0,0,0)
  Scan(5,Sec,0,0)
    If UsrForcedMsmnt=True Then
      'Set remote measurement parameters.
      SendVariables(SVResult(1),Com1,200,200,0000,100, "Public", _
      "TimeSeries(2)",TimeSeries(2),3)
      'Next Force measurement on indicated channel.
      SendVariables(SVResult(2),Com1,200,200,0000,100, "Public", _
      "TimeSeries(1)",TimeSeries(1),1)
      Delay (1,2,Sec)  'wait for 2 second measurement
      Do  'Check that measure is done
        GetVariables(GVResult(1),Com1,200,200,0000,100,"Public", _
        "TimeSeries(1)",TS_done,1)
        If GVResult(1)
          Exit Do  'failed communications
```

CRBasic Example 5: CRBasic Forced Measurement Program

```
        EndIf
      Loop Until TS_done  'when TS_done equals zero.. the measurement is done
      'Get the data from AVW206.
      GetVariables(GVResult(2),Com1,200,200,0000,100,"Public", _
      "TimeSeries(5)",TimeSeries(5),7)
      UsrForcedMsmnt=False
      If SVResult(1) OR SVResult(2) OR GVResult(1) OR GVResult(2) Then
        Move (TimeSeries(5),7,NAN,1)  'failed communication..so fill win NANs
      EndIf
    EndIf
  NextScan
EndProg
```

Appendix E. Status table

The AVW200 **Status** table contains system operating status information accessible via ***Device Configuration Utility***, terminal emulator, or another PakBus device such as a data logger. Status table information is easily viewed by going to ***Device Configuration Utility*** > **AVW200** > **Connect** > **Data monitor** > **Status**. The **Status** table can be viewed via a terminal emulator and command 4.

The status information can be retrieved by the data logger by using the CRBasic `GetVariables()` instruction. Following is an example of retrieving the `BattVoltage` status of the AVW200 using the CRBasic `GetVariables()` instruction:

```
Public RC,AVW_BV
GetVariables(RC,ComSDC7,200,200,0000,0,"Status","BattVoltage",AVW_BV,1)
```

> **NOTE:**
> ***Device Configuration Utility*** polls the **Status** table at regular intervals, updating status information.

Table E-1 (p. 88) is a comprehensive list of **Status** table variables with brief descriptions.

Table E-1: Status fields and descriptions

Status fieldname	Description	Changeable
Record No	Record number for this set of data	No
Time Stamp	Time the record was generated	No
OSversion	Version of the operating system	No
OSdate	Date OS was released	No
ProgName	Name of the running program	No
ProgSig	Signature of the running program	No
StationName	User-defined station name	Yes
Compileresults	Compiled results of the running program	Yes
PakBusAddress	AV200 PakBus address	Yes

Table E-1: Status fields and descriptions		
Status fieldname	Description	Changeable
RfInstalled	Specifies the model number of the MaxStream radio if it is recognized by the data logger; it will have a value of 0 if no radio is recognized by the AVW200	No
RfNetAddr	Specifies the radio network address of the built-in radio; should be set to match the network address for the RF401A base used to communicate with the data logger	Yes
RfNetHopSeq	Specifies the hopping sequence that will be used for the built-in radio; should be set to match the value of the same setting for the RF401A base station used to communicate with the data logger	Yes
Rf_ForceOn	When set to 1, radio is always on, ignoring the duty cycle setting	Yes
Rf_Protocol	Identification of radio protocol that will be used; AVW200 is always fixed at 2 (PakBus aware mode)	Yes, although changing this parameter to **1** will upset the RF communication); all other values will revert to a value of **2**
RfSignalLevel	Signal level of every fifth PakBus packet received over RF	Yes (clear to zero)
RfRxPakBusCnt	Number of PakBus packets that have been received over RF communication	Yes (clear to zero)
RfPwrMode	Radio power modes: NO_RF (No radio) RF_ON (<24ma always on) RF_1/2_Sec (<4ma ½ second) RF_1_Sec (<2ma 1 second) RF_8_Sec (<0.4ma 8 seconds) RF_OFF (Radio off)	Yes; to change from ½ second duty cycle to 1 second duty cycle mode, edit the parameter with RF_1_Sec

Table E-1: Status fields and descriptions		
Status fieldname	Description	Changeable
PortStatus(1)	Indicates control port 1 level 0 = off (low, 0 VDC) –1 = on (high, 5 VDC)	No
PortStatus(2)	Indicates control port 2 level 0 = off (low, 0 VDC) –1 = on (high, 5 VDC)	No
PortStatus(3)	Indicates control port 3 level 0 = off (low, 0 VDC) –1 = on (high, 5 VDC)	No
PortConfig(1)	Indicates control port 1 configuration; function disabled reserved for future use	Yes
PortConfig(2)	Indicates control port 2 configuration; function disabled reserved for future use	Yes
PortConfig(3)	Indicates control port 3 configuration; function disabled reserved for future use	Yes
MSPversion(1)	MSP430 CPU #1 OS version	No
MSPversion(2)	MSP430 CPU #2 OS version	No
MSPversion(3)	MSP430 CPU #3 OS version	No
MSPversion(4)	MSP430 CPU #4 OS version	No
MSPversion(5)	MSP430 CPU #5 OS version	No
MSPClkFreq(1)	MSP430 CPU #1 RC oscillator frequency in Hz	No
MSPClkFreq(2)	MSP430 CPU #2 RC oscillator frequency in Hz	No
MSPClkFreq(3)	MSP430 CPU #3 RC oscillator frequency in Hz	No
MSPClkFreq(4)	MSP430 CPU #4 RC oscillator frequency in Hz	No
MSPClkFreq(5)	MSP430 CPU #5 RC oscillator frequency in Hz	No
CalOffset	Calibration offset voltage	No

Table E-1: Status fields and descriptions		
Status fieldname	Description	Changeable
VarOutOfBounds	Number of times an array was accessed out of bounds	Yes (clear to zero)
SkipScan	Number of skipped scans that have occurred while running the current scan; when making the vibrating-wire measurement, it is normal for skipscan values to increment	Yes (clear to zero)
TrapCode	A code number that describes the last watchdog event that happened; updated when turned on	Yes
WatchDogCnt	Number of watchdog errors that occurred while running this program	Yes (clear to zero)
ResetTables	Not used	Yes (function disabled)
BattVoltage	Current value of the AVW200 battery voltage; value is updated every 8 sec	Yes
SRAMMemSize	Size of the SRAM memory	No

NOTE:
The `SRAMMemSize` parameter was added to OS version STD.02. Older modules have 128 KB of SRAM, and newer modules have 512 KB.

If `SRAMMemSize` = 512 KB, then the AVW200 will create and overwrite a file for every measurement on each channel. The files are called TS_chan1.bin and TS_chan2.bin. These files have 4,096 samples or **TimeSeries** data for the last measurement. These files can be retrieved using ***LoggerNet*** **FileControl** or the data logger instruction `GetFile()`. A post-processing program in ***Device Configuration Utility*** under device type AVW200 series is called “Off Line Analysis” and can be used to analyze the files.

Appendix F. Time series and Spectrum graph information

The AVW200 uses an audio A/D for capturing the sensor signal. The number of samples acquired in this period is 4,096 points. A fast Fourier transform (FFT) algorithm is used to create a frequency spectrum. The frequency spectrum is displayed in the graph labeled **Spectrum** (see FIGURE F 1), which shows each of the frequencies and the voltage amplitude in mV RMS.

The **Time Series** graph is the acquired or sampled data in the time domain. The graph shows the combination of all the frequencies coming from the vibrating-wire sensor shortly after sensor excitation. The dominant frequency is the natural resonating frequency of the vibrating wire. The other frequencies can include noise pickup (from motors close to the sensor or pickup due to long wire lengths), harmonics of the natural frequency or harmonics of the noise (50/60 Hz harmonics), and/or mechanical obstruction (such as wire loosening or package movement that causes physical changes to wire vibration). The AVW200 computes a signal-to-noise diagnostic by dividing the response amplitude by the noise amplitude.

The **Time Series** graph shows the decay from the start of the sampling to the end of the sampling. The decay is the dampening of the wire over time. The AVW200 computes a decay ratio diagnostic from the time series ending amplitude divided by the beginning amplitude. Some sensors will decay very rapidly, others not. Characterizing the sensor decay and amplitude when the sensor is new is a good idea, so that over time, the health of the sensor can be monitored.

By changing the begin and end frequencies in the **Options** tab, the effects of narrowing can be of value for troubleshooting and solving problems with errant sensors or for improving the measurement. Ensure that when the begin and end frequencies are changes, the frequency range still captures the sensor signal.

F.1 Good sensor examples

Figure F-1 (p. 93) and Figure F-2 (p. 94) are measurement results from the same sensor. The first measurement was taken with a swept frequency between 200 and 2200 Hz, while the second measurement was taken with a swept frequency between 200 and 6500 Hz. Using the tighter frequency range (Figure F-1 [p. 93]), the measurement recorded the greatest sensor noise at a frequency of 935 Hz with a signal-to-noise ratio of 318. Sweeping the same sensor over the far wider range of 200 to 6500 Hz (Figure F-2 [p. 94]) uncovers noise at 4150 Hz with a signal-to-noise ratio of 21.4, which is 15 times less than the signal-to-noise ratio of the first measurement.

This illustrates that better readings are produced when the sensor is swept over more narrow frequency ranges. Also, with the narrowed range (Figure F-1 [p. 93]), the noise frequency that exists at 4150 Hz is completely ignored and is not relevant because it lies outside the sampling frequency range. Excitation is limited to within the swept-frequency range as well.

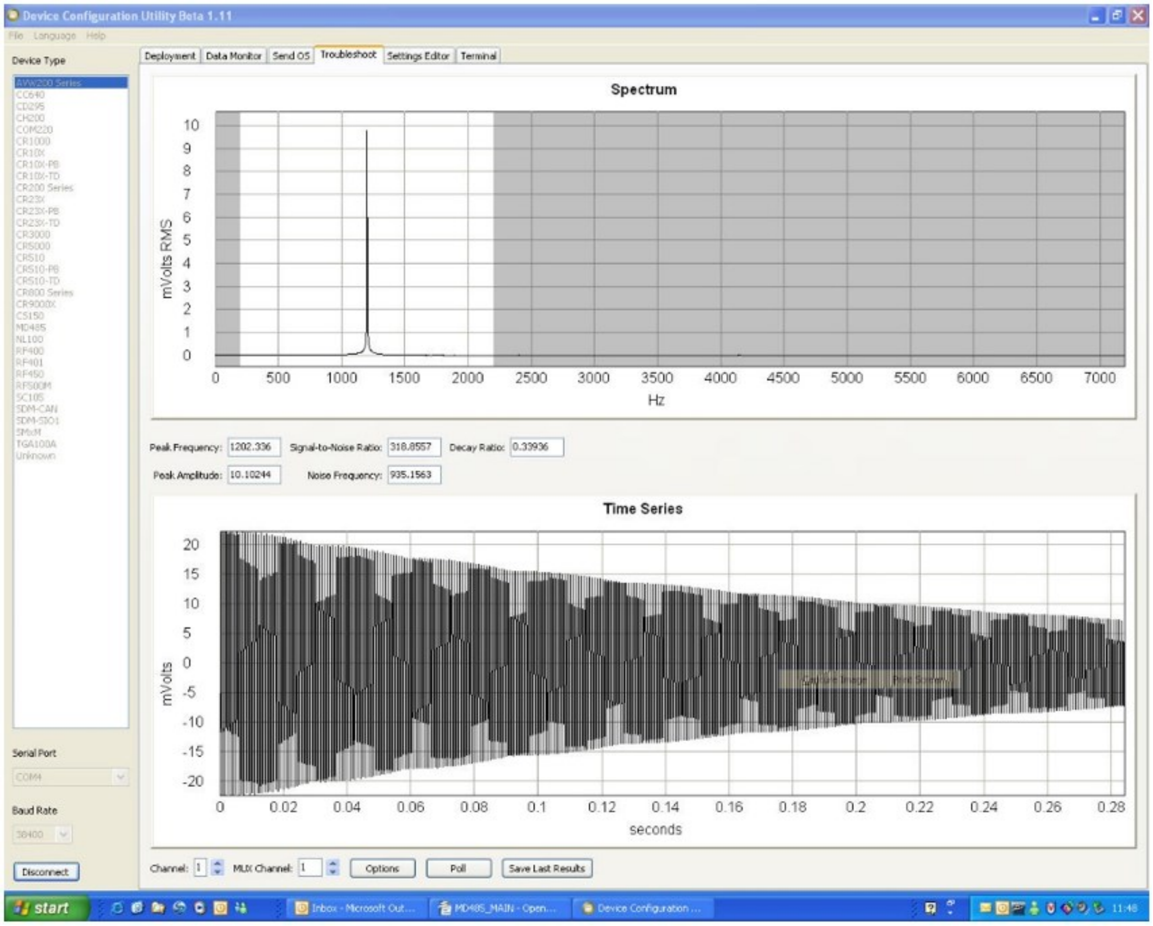


Figure F-1. Good sensor with a narrower range (200 to 2200 Hz)

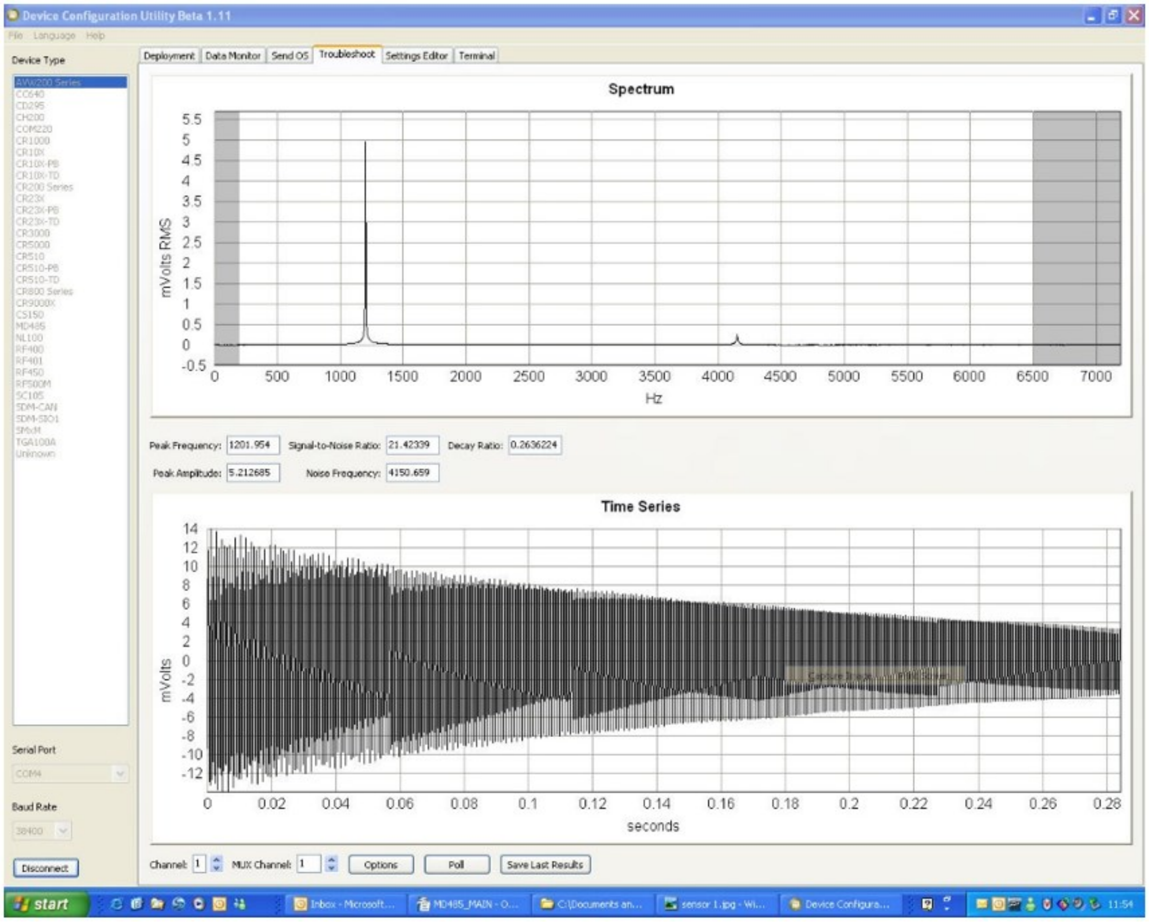


Figure F-2. Good sensor with a wider range (200 to 6500 Hz)

F.2 Good sensors with noise

The measurements graphed in Figure F-3 (p. 95) and Figure F-4 (p. 96) are made by the same sensor used for the two previous figures. However, for Figure F-3 (p. 95) and Figure F-4 (p. 96), a drill is running about 1/2 inch away from the sensor. This shows the effects of narrowing the begin and end frequencies to deal with noise generated by an electric motor. The narrow frequency range in Figure F-3 (p. 95) has reduced the effects of the noise source and has yielded a signal-to-noise ratio of four times better than in Figure F-4 (p. 96). Notice that if the begin frequency in Figure F-4 (p. 96) was much less than 450 Hz, the 60 Hz harmonic would have been the dominant frequency.

A powered drill 1/2 inch away from the sensor is an invasive noise source. When the sensor is measured with the drill a few inches away, the harmonics of the 60 Hz decrease and do not dominate the wire natural frequency. Sensors with a frequency range of less than 450 Hz should work fine even in the presence of a 50 or 60 Hz noise source; however, they should be characterized.

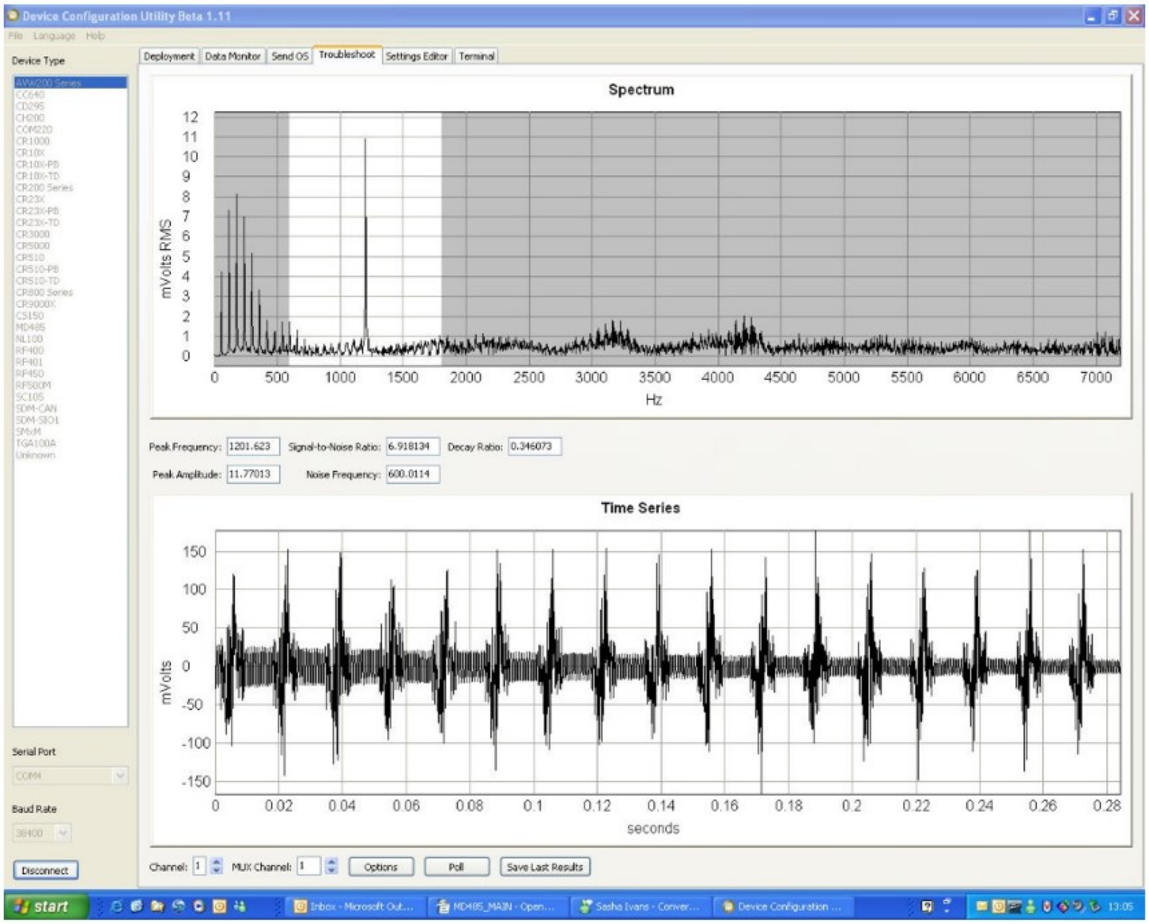


Figure F-3. Good sensor with a narrower range and noise (600 to 1800 Hz)

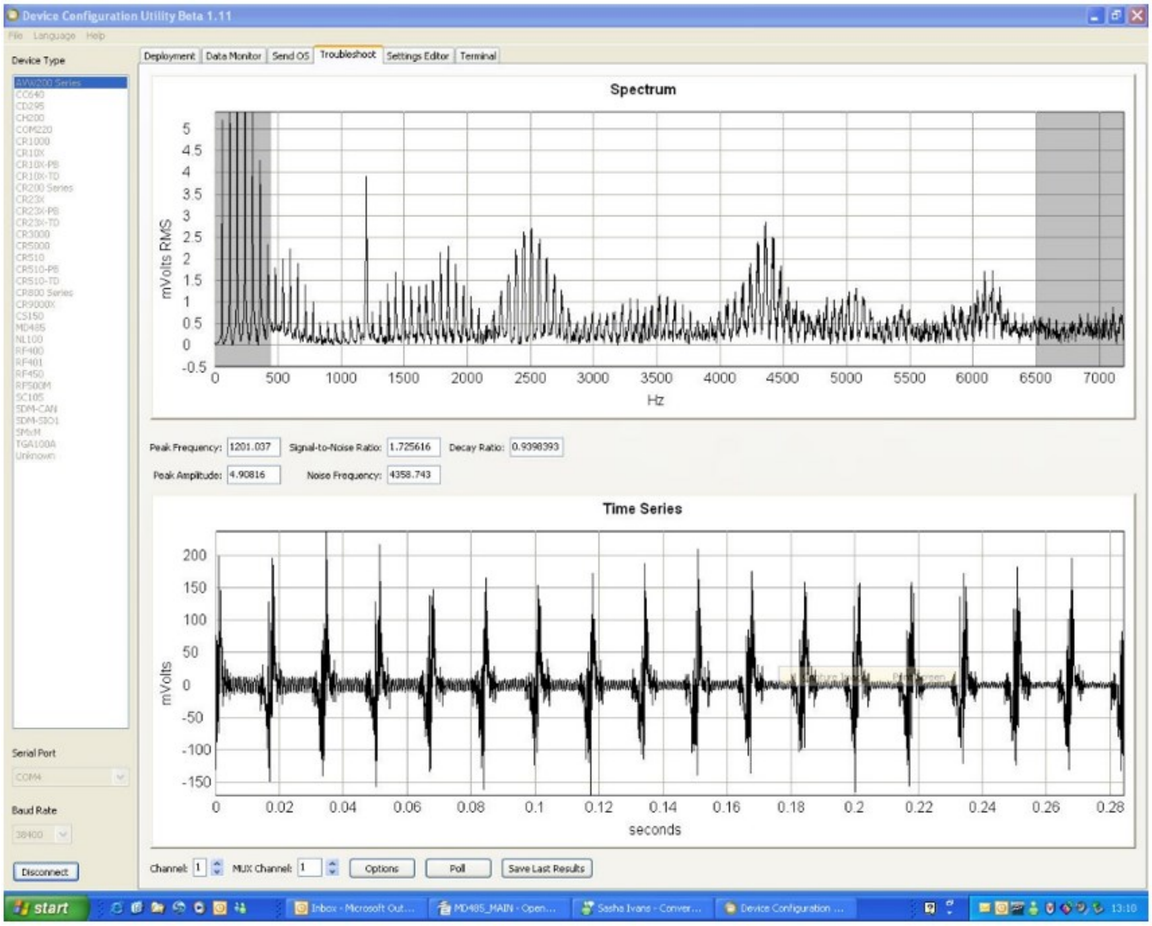


Figure F-4. Good sensor with a wider range and noise (450 to 6500 Hz)

> **NOTE:**
> Check manufacture specifications for the sensor frequency and excitation range before picking the begin and end frequencies and excitation voltage.

Appendix G. Additional programming examples

Downloadable example programs are available at www.campbellsci.com/downloads/avw200-example-programs.

G.1 AVW200-controlled multiplexer

G.1.1 Direct RS-232 connection

CRBasic Example: Two multiplexers with 16 sensors each

```
'This is an example of a program used by a CR1000 and AVW200 to control two
'AM16/32B multiplexers. Sixteen Geokon 4450 VW displacement sensors are attached
'to each multiplexer and each sensor provides a frequency, which is converted to
'displacement, and resistance, which is converted to temperature. Polynomial Gage
'Factors used in this example were taken from the calibration sheets of the
'individual 4450 sensors. The coefficients used to convert resistance to
'temperature are from the Steinhart-Hart equation.

Public batt_volt,Ptemp
Public Mux1(16,6)
Public Mux2(16,6)
Public VWResults(2)

Public Amp1(16),Amp2(16)
Public Temp1(16),Temp2(16)
Public Therm1(16),Therm2(16)
Public VWfreq1(16),VWfreq2(16)
Public Sig2Noise1(16),Sig2Noise2(16)
```

CRBasic Example: Two multiplexers with 16 sensors each

```
Public DecayRatio1(16),DecayRatio2(16)
Public FreqOfNoise1(16),FreqOfNoise2(16)
Public Displacement1(16),Displacement2(16)

Dim i
Dim j
Dim Digits
Dim ZeroRding(32)
Dim GageFactor(32)
Dim PolyCoef1(48) As Float
Dim PolyCoef2(48) As Float
Dim CoefString1(16) As String *30
Dim CoefString2(16) As String *30

Const A=.001403040 'Coefficients for Steinhart-Hart equation
Const B=.000237318 'used to convert resistance to Temp
Const C=.00000009

DataTable (MuxExample,1,-1)
  DataInterval (0,10,Min,10)
  Minimum (1,batt_volt,FP2,0,False)
  Sample (16,Displacement1(),FP2)
  Sample (16,VWfreq1(),FP2)
  Sample (16,Temp1(),FP2)
  Sample (16,Amp1(),FP2)
  Sample (16,Sig2Noise1(),FP2)
  Sample (16,FreqOfNoise1(),FP2)
  Sample (16,DecayRatio1(),FP2)
  Sample (16,Displacement2(),FP2)
  Sample (16,VWfreq2(),FP2)
  Sample (16,Temp2(),FP2)
  Sample (16,Amp2(),FP2)
  Sample (16,Sig2Noise2(),FP2)
  Sample (16,FreqOfNoise2(),FP2)
  Sample (16,DecayRatio2(),FP2)
EndTable

BeginProg

  SerialOpen (COMRS232,38400,0,0,10000)

  'Enter the 3 Polynomial Gage Factors for each sensor as listed on each
  'Calibration Report
  CoefString1(1) = "2.49866e-10, 8.716e-5, -0.20003"
  CoefString1(2) = "2.56640e-10, 8.762e-5, -0.20437"
  CoefString1(3) = "2.93650e-10, 8.715e-5, -0.19679"
  CoefString1(4) = "1.99647e-10, 8.868e-5, -0.19430"
```

CRBasic Example: Two multiplexers with 16 sensors each

```
CoefString1(5) = "3.41276e-10, 8.777e-5, -0.19042"
CoefString1(6) = "2.30397e-10, 8.720e-5, -0.19085"
Coefstring1(7) = "2.54131e-10, 8.743e-5, -0.19218"
CoefString1(8) = "2.21677e-10, 8.832e-5, -0.20539"
CoefString1(9) = "2.85034e-10, 8.734e-5, -0.19341"
CoefString1(10) = "2.42310e-10, 8.808e-5, -0.19576"
CoefString1(11) = "2.52871e-10, 8.804e-5, -0.19232"
CoefString1(12) = "2.27416e-10, 8.797e-5, -0.19552"
CoefString1(13) = "2.27264e-10, 8.798e-5, -0.19522"
CoefString1(14) = "2.87777e-10, 8.682e-5, -0.20353"
CoefString1(15) = "2.81051e-10, 8.767e-5, -0.19691"
CoefString1(16) = "2.41462e-10, 8.747e-5, -0.19481"

CoefString2(1) = "2.73949e-10, 8.726e-5, -0.20799"
CoefString2(2) = "2.91941e-10, 8.722e-5, -0.20443"
CoefString2(3) = "2.32081e-10, 8.789e-5, -0.20064"
CoefString2(4) = "3.17163e-10, 8.741e-5, -0.19108"
CoefString2(5) = "2.66284e-10, 8.810e-5, -0.19411"
CoefString2(6) = "2.7768e-10, 8.756e-5, -0.19630"
CoefString2(7) = "2.06552e-10, 8.866e-5, -0.20578"
CoefString2(8) = "2.22761e-10, 8.848e-5, -0.20013"
CoefString2(9) = "2.31843e-10, 8.880e-5, -0.19643"
CoefString2(10) = "2.42310e-10, 8.808e-5, -0.19576"
CoefString2(11) = "1.99647e-10, 8.868e-5, -0.19430"
CoefString2(12) = "2.56640e-10, 8.762e-5, -0.20437"
CoefString2(13) = "2.81051e-10, 8.767e-5, -0.19691"
CoefString2(14) = "2.81051e-10, 8.767e-5, -0.19691"
CoefString2(15) = "2.49866e-10, 8.716e-5, -0.20003"
CoefString2(16) = "2.27264e-10, 8.798e-5, -0.19522"

For i = 1 To 16
  SplitStr (PolyCoef1(3*i-2),CoefString1(i),",",3,5)
  'Assign coefficients listed in CoefString1 to individual variables
Next i

'Assign coefficients listed in CoefString2 to individual variables
For i = 1 To 16
  SplitStr (PolyCoef2(3*i-2),CoefString2(i),",",3,5)
Next i

Scan (2,Min,0,0)
  PanelTemp (PTemp,60)
  Battery (Batt_volt)

  AVW200(VWResults(1),ComRS232,0,15,Mux1(1,1),1,1,16,1000,2500,2,_60Hz,1,0)

  For i = 1 To 16
```

CRBasic Example: Two multiplexers with 16 sensors each

```
      Amp1(i) = Mux1(i,2)
      Therm1(i) = Mux1(i,6)
      VWFreq1(i) = Mux1(i,1)
      Sig2Noise1(i) = Mux1(i,3)
      DecayRatio1(i) = Mux1(i,5)
      FreqOfNoise1(i) = Mux1(i,4)
      Digits = (VWFreq1(i)/1000)^2 * 1000 'Convert frequency to Digits

      'Convert resistance to temp F.
      Temp1(i) = (1/(A + B*LN(Therm1(i)) +C*(LN(Therm1(i)))^3)-273.15)*1.8+32

      'Calculate displacement (inches) from Digits and calibration polynomial
      Displacement1(i)=PolyCoef1(3*i-2)*Digits^2 + PolyCoef1(3*i-1)*Digits+ _
      PolyCoef2(3*i)
    Next i

    AVW200(VWResults(2),ComRS232,0,15,Mux2(1,1),2,1,8,1000,2500,2,_60Hz,1, 0)
    AVW200(VWResults(2),ComRS232,0,15,Mux2(9,1),2,9,8,450,6500,2,_60Hz,1,0)

    For i = 1 To 16
      Amp2(i) = Mux1(i,2)
      Therm2(i) = Mux1(i,6)
      VWFreq2(i) = Mux1(i,1)
      Sig2Noise2(i) = Mux1(i,3)
      DecayRatio2(i) = Mux1(i,5)
      FreqOfNoise2(i) = Mux1(i,4)
      Digits = (VWFreq2(i))^2/1000 'Convert frequency to Digits

      'Calculate displacement (inches) from Digits and calibration polynomial

      Displacement2(i)=PolyCoef2(3*i-2)*Digits^2 + PolyCoef2(3*i-1)*Digits + _
      PolyCoef2(3*i)

      'Convert resistance to temp F.
      Temp2(i) = (1/(A + B*LN(Therm2(i)) + C*(LN(Therm2(i)))^3)-273.15)*1.8+32
    Next i

    CallTable MuxExample
  NextScan
EndProg
```

G.1.2 Wireless/sensors with different frequencies

CRBasic Example: AVW206-2-multiplexers-sensors with different frequencies

```
'This is an example of a program used by a CR1000X-Series data logger and AVW206
'to control two AM16/32B multiplexers. Sixteen Geokon 4450 VW displacement
'sensors are attached to each multiplexer and each sensor provides a frequency,
'which is converted to displacement, and resistance, which is converted to
'temperature. Polynomial Gage Factors used in this example were taken
'from the calibration sheets of the individual 4450 sensors. The
'coefficients used to convert resistance to temperature are from the
'Steinhart-Hart equation.

'The CR1000X communicates with the remote AVW206 through a RF401A radio
'attached to the data logger's CS/IO port
'The Pakbus address of the AVW206 used in this example is 20

Public batt_volt,Ptemp
Public Mux1(16,6)
Public Mux2(16,6)
Public VWResults(2)

Public Amp1(16),Amp2(16)
Public Temp1(16),Temp2(16)
Public Therm1(16),Therm2(16)
Public VWfreq1(16),VWfreq2(16)
Public Sig2Noise1(16),Sig2Noise2(16)
Public DecayRatio1(16),DecayRatio2(16)
Public FreqOfNoise1(16),FreqOfNoise2(16)
Public Displacement1(16),Displacement2(16)

Dim i
Dim Digits
Dim PolyCoef1(48) As Float
Dim PolyCoef2(48) As Float
Dim CoefString1(16) As String *30
Dim CoefString2(16) As String *30

Const A=.0014051 'Coefficients for Steinhart-Hart equation
Const B=.0002369 'used to convert resistance to Temp
Const C=.0000001019

'Store Freq, amplitude, signal to noise, freq of noise, decay ratio and
'resistance from both mux's.
DataTable (MuxExample,1,-1)
  DataInterval (0,10,Min,10)
  Minimum (1,batt_volt,FP2,0,False)
  Sample (16,Displacement1(),FP2)
  Sample (16,VWfreq1(),FP2)
```

CRBasic Example: AVW206-2-multiplexers-sensors with different frequencies

```
  Sample (16,Temp1(),FP2)
  Sample (16,Amp1(),FP2)
  Sample (16,Sig2Noise1(),FP2)
  Sample (16,FreqOfNoise1(),FP2)
  Sample (16,DecayRatio1(),FP2)
  Sample (16,Displacement2(),FP2)
  Sample (16,VWfreq2(),FP2)
  Sample (16,Temp2(),FP2)
  Sample (16,Amp2(),FP2)
  Sample (16,Sig2Noise2(),FP2)
  Sample (16,FreqOfNoise2(),FP2)
  Sample (16,DecayRatio2(),FP2)
EndTable

BeginProg

  'Enter the 3 Polynomial Gage Factors for each sensor
  'as listed on each Calibration Report
  CoefString1(1) = "2.49866e-10, 8.716e-5, -0.20003"
  CoefString1(2) = "2.56640e-10, 8.762e-5, -0.20437"
  CoefString1(3) = "2.93650e-10, 8.715e-5, -0.19679"
  CoefString1(4) = "1.99647e-10, 8.868e-5, -0.19430"
  CoefString1(5) = "3.41276e-10, 8.777e-5, -0.19042"
  CoefString1(6) = "2.30397e-10, 8.720e-5, -0.19085"
  CoefString1(7) = "2.54131e-10, 8.743e-5, -0.19218"
  CoefString1(8) = "2.21677e-10, 8.832e-5, -0.20539"
  CoefString1(9) = "2.85034e-10, 8.734e-5, -0.19341"
  CoefString1(10) = "2.42310e-10, 8.808e-5, -0.19576"
  CoefString1(11) = "2.52871e-10, 8.804e-5, -0.19232"
  CoefString1(12) = "2.27416e-10, 8.797e-5, -0.19552"
  CoefString1(13) = "2.27264e-10, 8.798e-5, -0.19522"
  CoefString1(14) = "2.87777e-10, 8.682e-5, -0.20353"
  CoefString1(15) = "2.81051e-10, 8.767e-5, -0.19691"
  CoefString1(16) = "2.41462e-10, 8.747e-5, -0.19481"

  CoefString2(1) = "2.73949e-10, 8.726e-5, -0.20799"
  CoefString2(2) = "2.91941e-10, 8.722e-5, -0.20443"
  CoefString2(3) = "2.32081e-10, 8.789e-5, -0.20064"
  CoefString2(4) = "3.17163e-10, 8.741e-5, -0.19108"
  CoefString2(5) = "2.66284e-10, 8.810e-5, -0.19411"
  CoefString2(6) = "2.7768e-10, 8.756e-5, -0.19630"
  CoefString2(7) = "2.06552e-10, 8.866e-5, -0.20578"
  CoefString2(8) = "2.22761e-10, 8.848e-5, -0.20013"
  CoefString2(9) = "2.31843e-10, 8.880e-5, -0.19643"
  CoefString2(10) = "2.42310e-10, 8.808e-5, -0.19576"
  CoefString2(11) = "1.99647e-10, 8.868e-5, -0.19430"
  CoefString2(12) = "2.56640e-10, 8.762e-5, -0.20437"
```

CRBasic Example: AVW206-2-multiplexers-sensors with different frequencies

```
CoefString2(13) = "2.81051e-10, 8.767e-5, -0.19691"
CoefString2(14) = "2.81051e-10, 8.767e-5, -0.19691"
CoefString2(15) = "2.49866e-10, 8.716e-5, -0.20003"
CoefString2(16) = "2.27264e-10, 8.798e-5, -0.19522"

'Assign coefficients listed in CoefString1 to individual variables
For i = 1 To 16
  SplitStr (PolyCoef1(3*i-2),CoefString1(i),",",3,5)
Next i

'Assign coefficients listed in CoefString2 to individual variables
For i = 1 To 16
  SplitStr (PolyCoef2(3*i-2),CoefString2(i),",",3,5)
Next i

Scan (2,Min,0,0)
  PanelTemp (Ptemp,60)
  Battery (batt_volt)

  AVW200(VWResults(1),ComSDC7,0,20,Mux1(1,1),1,1,16,1000,2500,2,_60Hz,1,0)
  For i = 1 To 16
    Amp1(i) = Mux1(i,2)
    Therm1(i) = Mux1(i,6)
    VWfreq1(i) = Mux1(i,1)
    Sig2Noise1(i) = Mux1(i,3)
    DecayRatio1(i) = Mux1(i,5)
    FreqOfNoise1(i) = Mux1(i,4)
    Digits = (VWfreq1(i)^2)/1000 'Convert frequency to Digits

    'Convert resistance to temp F.
    Temp1(i) = (1/(A + B*LN(Therm1(i)) + C*(LN(Therm1(i)))^3)-273.15)*1.8+32

    'Calculate displacement (inches) from Digits and calibration polynomial
    Displacement1(i)=PolyCoef1(3*i-2)*Digits^2 + PolyCoef1(3*i-1)*Digits + _
    PolyCoef2(3*i)

  Next i

  'Sensors 1-8 are excited over the freq range of 1000 - 2500
  AVW200(VWResults(2),ComSDC7,0,20,Mux2(1,1),2,1,8,1000,2500,2,_60Hz,1,0)

  'Sensors 9-16 are excited over the freq range of 450 - 6500
  AVW200(VWResults(2),ComSDC7,0,20,Mux2(9,1),2,9,8,450,6500,2,_60Hz,1,0)
  For i = 1 To 16
    Amp2(i) = Mux1(i,2)
    Therm2(i) = Mux1(i,6)
    VWfreq2(i) = Mux1(i,1)
```

CRBasic Example: AVW206-2-multiplexers-sensors with different frequencies

```
      Sig2Noise2(i) = Mux1(i,3)
      DecayRatio2(i) = Mux1(i,5)
      FreqOfNoise2(i) = Mux1(i,4)
      Digits = VWfreq2(i)2)/1000 'Convert frequency to Digits

      'Calculate displacement (inches) from Digits and calibration polynomial
      Displacement2(i)=PolyCoef2(3*i-2)*Digits^2 + PolyCoef2(3*i-1)*Digits + _
      PolyCoef2(3*i)

      'Convert resistance to temp F.
      Temp2(i) = (1/(A + B*LN(Therm2(i)) + C*(LN(Therm2(i)))^3)-273.15)*1.8+32
    Next i

    CallTable MuxExample
  NextScan
EndProg
```

G.2 Data logger-controlled multiplexer

CRBasic Example: 4 multiplexers controlled by the data logger

```
'This example demonstrates how to program a CR1000X-Series data logger to collect
'measurements from sensors attached to four AM16/32 multiplexers. The four
'multiplexers are controlled directly by the data logger, not through the AVW200
'as in other examples contained in this manual. Displacement is calculated from
'the measured frequencies by applying the Polynomial Gage Factors contained
'in each sensors' calibration report. Only the resistance from each thermistor
'is measured. Temp is calculated by applying the resistance to a known
'equation which converts resistance To temp. Temperature correction is not
'shown in this example. In this example 16 VW displacement sensors with
'thermistors are attached to both the first and second multiplexers, while
'only 9 similar sensors are attached to the third multiplexer. 32 VW
'displacement sensors with no thermistors are attached to the fourth
'multiplexer.

SequentialMode

Public batt_volt
Public Mux(6)
Public VWResults

Public Temp1(16),Temp2(16),Temp3(12)
Public Amp1(16),Amp2(16),Amp3(12),Amp4(32)
Public VWfreq1(16),VWfreq2(16), VWFreq3(9),VWFreq4(32)
Public Sig2Noise1(16),Sig2Noise2(16),Sig2Noise3(12),Sig2Noise4(32)
```

CRBasic Example: 4 multiplexers controlled by the data logger

```
Public DecayRatio1(16),DecayRatio2(16),DecayRatio3(9),DecayRatio4(32)
Public FreqOfNoise1(16),FreqOfNoise2(16),FreqOfNoise3(12),FreqOfNoise4(32)
Public Displacement1(16),Displacement2(16), Displacement3(9),Displacement4(32)

Dim i
Dim j
Dim Digits
Dim Coef1(48)
Dim Coef2(48)
Dim Coef3(27)
Dim Coef4(96)

Const A=.0014051 'Coefficients for Steinhart-Hart equation
Const B=.0002369 'used to convert resistance to Temp
Const C=.0000001019

DataTable (MuxExample,1,-1)
  DataInterval (0,15,Min,10)
  Minimum (1,batt_volt,FP2,0,False)
  Sample (16,Displacement1(),FP2)
  Sample (16,VWfreq1(),FP2)
  Sample (16,Temp1(),FP2)
  Sample (16,Amp1(),FP2)
  Sample (16,Sig2Noise1(),FP2)
  Sample (16,FreqOfNoise1(),FP2)
  Sample (16,DecayRatio1(),FP2)
  Sample (16,Displacement2(),FP2)
  Sample (16,VWfreq2(),FP2)
  Sample (16,Temp2(),FP2)
  Sample (16,Amp2(),FP2)
  Sample (16,Sig2Noise2(),FP2)
  Sample (16,FreqOfNoise2(),FP2)
  Sample (16,DecayRatio2(),FP2)
  Sample (9,Displacement3(),FP2)
  Sample (9,VWFreq3(),FP2)
  Sample (9,Temp3(),FP2)
  Sample (9,Amp3(),FP2)
  Sample (9,Sig2Noise3(),FP2)
  Sample (9,FreqOfNoise3(),FP2)
  Sample (9,DecayRatio3(),FP2)
  Sample (32,Displacement4(),FP2)
  Sample (32,VWFreq4(),FP2)
  Sample (32,Amp4(),FP2)
  Sample (32,Sig2Noise4(),FP2)
  Sample (32,FreqOfNoise4(),FP2)
  Sample (32,DecayRatio4(),FP2)
EndTable
```

CRBasic Example: 4 multiplexers controlled by the data logger

```
BeginProg

  'Enter the 3 Polynomial Gage Factors for each sensor as listed on each
  'Calibration Report

  'Gage Factors for sensors attached to AM16/32 #1
  Coef1(1) = 2.49866e-10: Coef1(2) = 8.716e-5: Coef1(3) = -0.20003
  Coef1(4) = 2.56640e-10: Coef1(5) = 8.762e-5: Coef1(6) = -0.20437
  Coef1(7) = 2.93650e-10: Coef1(8) = 8.715e-5: Coef1(9) = -0.19679
  Coef1(10) = 1.99647e-10: Coef1(11) = 8.868e-5: Coef1(12) = -0.19430
  Coef1(13) = 3.41276e-10: Coef1(14) = 8.777e-5: Coef1(15) = -0.19042
  Coef1(16) = 2.30397e-10: Coef1(17) = 8.720e-5: Coef1(18) = -0.19085
  Coef1(19) = 2.54131e-10: Coef1(20) = 8.743e-5: Coef1(21) = -0.19218
  Coef1(22) = 2.21677e-10: Coef1(23) = 8.832e-5: Coef1(24) = -0.20539
  Coef1(25) = 2.85034e-10: Coef1(26) = 8.734e-5: Coef1(27) = -0.19341
  Coef1(28) = 2.42310e-10: Coef1(29) = 8.808e-5: Coef1(30) = -0.19576
  Coef1(31) = 2.52871e-10: Coef1(32) = 8.804e-5: Coef1(33) = -0.19232
  Coef1(34) = 2.27416e-10: Coef1(35) = 8.797e-5: Coef1(36) = -0.19552
  Coef1(37) = 2.27264e-10: Coef1(38) = 8.798e-5: Coef1(39) = -0.19522
  Coef1(40) = 2.87777e-10: Coef1(41) = 8.682e-5: Coef1(42) = -0.20353
  Coef1(43) = 2.81051e-10: Coef1(44) = 8.767e-5: Coef1(45) = -0.19691
  Coef1(46) = 2.41462e-10: Coef1(47) = 8.747e-5: Coef1(48) = -0.19481

  'Gage Factors for sensors attached to AM16/32 #2
  Coef2(1) = 2.73949e-10: Coef2(2) = 8.726e-5: Coef2(3) = -0.20799
  Coef2(4) = 2.91941e-10: Coef2(5) = 8.722e-5: Coef2(6) = -0.20443
  Coef2(7) = 2.32081e-10: Coef2(8) = 8.789e-5: Coef2(9) = -0.20064
  Coef2(10) = 3.17163e-10: Coef2(11) = 8.741e-5: Coef2(12) = -0.19108
  Coef2(13) = 2.66284e-10: Coef2(14) = 8.810e-5: Coef2(15) = -0.19411
  Coef2(16) = 2.77680e-10: Coef2(17) = 8.756e-5: Coef2(18) = -0.19630
  Coef2(19) = 2.06552e-10: Coef2(20) = 8.866e-5: Coef2(21) = -0.20578
  Coef2(22) = 2.22761e-10: Coef2(23) = 8.848e-5: Coef2(24) = -0.20013
  Coef2(25) = 2.31843e-10: Coef2(26) = 8.880e-5: Coef2(27) = -0.19643
  Coef2(28) = 2.42310e-10: Coef2(29) = 8.808e-5: Coef2(30) = -0.19576
  Coef2(31) = 1.99647e-10: Coef2(32) = 8.868e-5: Coef2(33) = -0.19430
  Coef2(34) = 2.56640e-10: Coef2(35) = 8.762e-5: Coef2(36) = -0.20437
  Coef2(37) = 2.81051e-10: Coef2(38) = 8.767e-5: Coef2(39) = -0.19691
  Coef2(40) = 2.81051e-10: Coef2(41) = 8.767e-5: Coef2(42) = -0.19691
  Coef2(43) = 2.49866e-10: Coef2(44) = 8.716e-5: Coef2(45) = -0.20003
  Coef2(46) = 2.27264e-10: Coef2(47) = 8.798e-5: Coef2(48) = -0.19522

  'Gage Factors for sensors attached to AM16/32 #3
  Coef3(1) = 2.73949e-10: Coef3(2) = 8.726e-5: Coef3(3) = -0.20799
  Coef3(4) = 3.17163e-10: Coef3(5) = 8.741e-5: Coef3(6) = -0.19108
  Coef3(7) = 2.49866e-10: Coef3(8) = 8.716e-5: Coef3(9) = -0.20003
  Coef3(10) = 2.31843e-10: Coef3(11) = 8.880e-5: Coef3(12) = -0.19643
```

CRBasic Example: 4 multiplexers controlled by the data logger

```
Coef3(13) = 2.22761e-10: Coef3(14) = 8.848e-5: Coef3(15) = -0.20013
Coef3(16) = 2.32081e-10: Coef3(17) = 8.789e-5: Coef3(18) = -0.20064
Coef3(19) = 3.41276e-10: Coef3(20) = 8.777e-5: Coef3(21) = -0.19042
Coef3(22) = 2.85034e-10: Coef3(23) = 8.734e-5: Coef3(24) = -0.19341
Coef3(25) = 2.66284e-10: Coef3(26) = 8.810e-5: Coef3(27) = -0.19411

'Gage Factors for sensors attached to AM16/32 #4
Coef4(1) = 2.49866e-10: Coef4(2) = 8.716e-5: Coef4(3) = -0.20003
Coef4(4) = 2.56640e-10: Coef4(5) = 8.762e-5: Coef4(6) = -0.20437
Coef4(7) = 2.93650e-10: Coef4(8) = 8.715e-5: Coef4(9) = -0.19679
Coef4(10) = 1.99647e-10: Coef4(11) = 8.868e-5: Coef4(12) = -0.19430
Coef4(13) = 3.41276e-10: Coef4(14) = 8.777e-5: Coef4(15) = -0.19042
Coef4(16) = 2.30397e-10: Coef4(17) = 8.720e-5: Coef4(18) = -0.19085
Coef4(19) = 2.54131e-10: Coef4(20) = 8.743e-5: Coef4(21) = -0.19218
Coef4(22) = 2.21677e-10: Coef4(23) = 8.832e-5: Coef4(24) = -0.20539
Coef4(25) = 2.85034e-10: Coef4(26) = 8.734e-5: Coef4(27) = -0.19341
Coef4(28) = 2.42310e-10: Coef4(29) = 8.808e-5: Coef4(30) = -0.19576
Coef4(31) = 2.52871e-10: Coef4(32) = 8.804e-5: Coef4(33) = -0.19232
Coef4(34) = 2.27416e-10: Coef4(35) = 8.797e-5: Coef4(36) = -0.19552
Coef4(37) = 2.27264e-10: Coef4(38) = 8.798e-5: Coef4(39) = -0.19522
Coef4(40) = 2.87777e-10: Coef4(41) = 8.682e-5: Coef4(42) = -0.20353
Coef4(43) = 2.81051e-10: Coef4(44) = 8.767e-5: Coef4(45) = -0.19691
Coef4(46) = 2.41462e-10: Coef4(47) = 8.747e-5: Coef4(48) = -0.19481
Coef4(49) = 2.73949e-10: Coef4(51) = 8.726e-5: Coef4(52) = -0.20799
Coef4(53) = 2.81051e-10: Coef4(54) = 8.767e-5: Coef4(55) = -0.19691
Coef4(56) = 2.81051e-10: Coef4(57) = 8.767e-5: Coef4(58) = -0.19691
Coef4(59) = 2.49866e-10: Coef4(60) = 8.716e-5: Coef4(61) = -0.20003
Coef4(62) = 2.27264e-10: Coef4(63) = 8.798e-5: Coef4(64) = -0.19522
Coef4(65) = 2.91941e-10: Coef4(66) = 8.722e-5: Coef4(67) = -0.20443
Coef4(68) = 2.32081e-10: Coef4(69) = 8.789e-5: Coef4(70) = -0.20064
Coef4(71) = 3.17163e-10: Coef4(72) = 8.741e-5: Coef4(73) = -0.19108
Coef4(74) = 2.66284e-10: Coef4(75) = 8.810e-5: Coef4(76) = -0.19411
Coef4(77) = 2.77680e-10: Coef4(77) = 8.756e-5: Coef4(78) = -0.19630
Coef4(79) = 2.06552e-10: Coef4(80) = 8.866e-5: Coef4(81) = -0.20578
Coef4(82) = 2.22761e-10: Coef4(83) = 8.848e-5: Coef4(84) = -0.20013
Coef4(85) = 2.31843e-10: Coef4(86) = 8.880e-5: Coef4(87) = -0.19643
Coef4(88) = 2.42310e-10: Coef4(89) = 8.808e-5: Coef4(90) = -0.19576
Coef4(91) = 1.99647e-10: Coef4(92) = 8.868e-5: Coef4(93) = -0.19430
Coef4(94) = 2.56640e-10: Coef4(95) = 8.762e-5: Coef4(96) = -0.20437

SerialOpen (COMRS232,38400,0,0,10000)
'Prepare the logger to receive serial data

Scan (15,Min,0,0)
  Battery (batt_volt)

  PortSet(C1,1) 'Enable Mux1
```

CRBasic Example: 4 multiplexers controlled by the data logger

```
For i = 1 To 16 'Do the following for each of 16 sensors:
  PulsePort(C2,1000) 'Provide pulse to advance to next channel on Mux1
  Delay (0,100,mSec)
  AVW200(VWResults,ComRS232,0,15,Mux(1),1,1,1,2500,3500,2,_60Hz,1,0)
  'Make VW measurement
  VWfreq1(i) = Mux(1) 'Assign vw frequency to the VWFreq1 variable
  Amp1(i) = Mux(2) 'Assign signal amplitude to Amp1 variable
  Sig2Noise1(i) = Mux(3) 'Assign signal to noise ratio to Sig2Noise1 variable
  FreqOfNoise1(i) = Mux(4) 'Assign frequency of competing noise to
                            'FreqOfNoise1 variable
  DecayRatio1(i) = Mux(5) 'Assign signal decay ratio to DecayRatio1 variable
  Digits = (VWFreq3(i)^2)/1000 'Convert frequency to Digits
  Temp1(i) = (1/(A + B*LN(Mux(6)) + C*(LN(Mux(6)))^3)-273.15)*1.8+32
  'Convert resistance to temp F.
  'Calculate displacement (inches) from Digits and calibration polynomial
  Displacement1(i)=Coef1(3*i-2)*Digits^2 + Coef1(3*i-1)*Digits + Coef1(3*i)
Next i
PortSet(C1,0) 'Reset and Disable Mux1

PortSet(C3,1) 'Enable Mux2
For i = 1 To 16  'Do the following for each of 16 sensors:
  PulsePort(C4,1000) 'Provide pulse to advance to next channel on Mux2
  AVW200(VWResults,ComRS232,0,15,Mux(1),1,1,1,2500,3500,2,_60Hz,1,0)
  'Make VW measurement
  VWfreq2(i) = Mux(1) 'Assign vw frequency to the VWFreq2 variable
  Amp2(i) = Mux(2) 'Assign signal amplitude to Amp2 variable
  Sig2Noise2(i) = Mux(3) 'Assign signal to noise ratio to Sig2Noise2 variable
  FreqOfNoise2(i) = Mux(4) 'Assign frequency of competing noise to
                            'FreqOfNoise2 variable
  DecayRatio2(i) = Mux(5) 'Assign signal decay ratio to DecayRatio2 variable
  Digits = (VWfreq2(i)^2)/1000 'Convert frequency to Digits
  Temp2(i) = (1/(A + B*LN(Mux(6)) + C*(LN(Mux(6)))^3)-273.15)*1.8+3
  'Convert resistance to temp F.
  'Calculate displacement (inches) from Digits and calibration ploynomial
  Displacement2(i) = Coef2(3*i-2)*Digits^2 + Coef2(3*i-1)*Digits + Coef2(3*i)
Next i
PortSet(C3,0) 'Reset and Disable Mux2

PortSet(C5,1) 'Enable Mux3
For i = 1 To 9 'Do the following for each of 9 sensors:
  PulsePort(C6,1000) 'Provide pulse to advance to next channel on Mux3
  AVW200(VWResults,ComRS232,0,15,Mux(1),1,1,1,2500,3500,2,_60Hz,1,0)
  VWFreq3(i) = Mux(1) 'Assign vw frequency to the VWFreq2 variable
  Amp3(i) = Mux(2) 'Assign signal amplitude to Amp2 variable
  Sig2Noise3(i) = Mux(3) 'Assign signal to noise ratio to Sig2Noise2 variable
  FreqOfNoise3(i) = Mux(4) 'Assign frequency of competing noise to
                            'FreqOfNoise2 variable
```

CRBasic Example: 4 multiplexers controlled by the data logger

```
      DecayRatio3(i) = Mux(5) 'Assign signal decay ratio to DecayRatio2 variable
      Digits = (VWFreq3(i)^2)/1000 'Convert frequency to Digits
      Temp3(i) = (1/(A + B*LN(Mux(6)) + C*(LN(Mux(6)))^3)-273.15)*1.8+3
      'Convert resistance to temp F.
      'Calculate displacement (inches) from Digits and calibration ploynomial
      Displacement3(i) = Coef3(3*i-2)*Digits^2 + Coef3(3*i-1)*Digits + Coef3(3*i)
    Next i
    PortSet(C5,0) 'Reset and disable Mux3

    PortSet(C7,1) 'Enable Mux4
    For i = 1 To 32 'Do the following for each of 32 sensors:
      PulsePort(C8,1000) 'Provide pulse to advance to next channel on Mux4
      AVW200(VWResults,ComRS232,0,15,Mux(1),1,1,1,2500,3500,2,_60Hz,1,0)
      'Make VW measurement
      VWFreq4(i) = Mux(1) 'Assign vw frequency to the VWFreq2 variable
      Amp4(i) = Mux(2) 'Assign signal amplitude to Amp2 variable
      Sig2Noise4(i) = Mux(3) 'Assign signal to noise ratio to Sig2Noise2 variable
      FreqOfNoise4(i) = Mux(4) 'Assign frequency of competing noise to
                               'FreqOfNoise2 variable
      DecayRatio4(i) = Mux(5) 'Assign signal decay ratio to DecayRatio2 variable
      Digits = (VWFreq4(i)^2)/1000 'Convert frequency to Digits
      'Calculate displacement (inches) from Digits and calibration ploynomial
      Displacement4(i) = Coef4(3*i-2)*Digits^2 + Coef4(3*i-1)*Digits + Coef4(3*i)
    Next i
    PortSet(C7,0) 'Reset and disable Mux4

    CallTable MuxExample
  NextScan
EndProg
```

Appendix H. MD485 multidrop modems used with AVW200 interfaces

For situations where wireless communication s are impractical, MD485 multidrop modems may be used to extend the distance between the AVW200 interfaces. This application is not compatible with SDI-12 communications.

H.1 Required settings

Device Configuration Utility is used to configure the settings for the AVW200 interfaces and the MD485 multidrop modems. A unique address must be assigned to each AVW200 in the network. This address may be entered in Deployment, Communications, or Settings Editor tabs (refer to Device Configuration Utility (p. 32) for more information).

Each MD485 in the network must be configured with the following settings (see also Figure H-1 [p. 111]):

Active ports: RS-232 and RS-485
Protocol configuration: PakBus networking
RS-232 baud rate: 38.4k
RS-485 baud rate: 38.4k

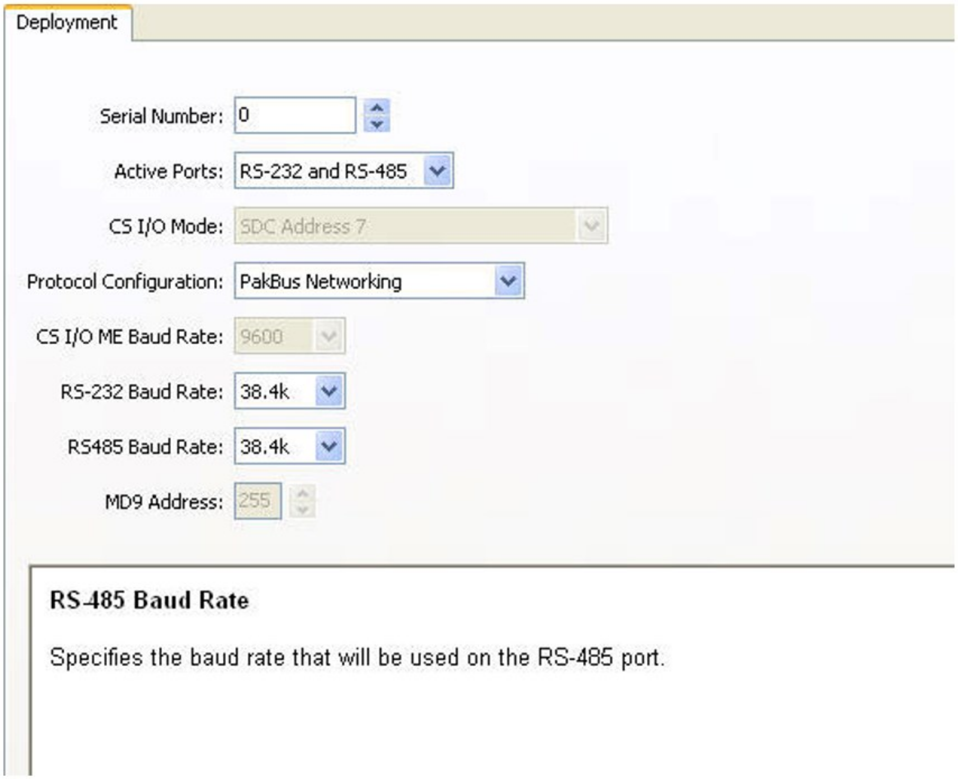


*Figure H-1. Deployment tab in **Device Configuration Utility** with proper MD-485 configuration*

H.2 Connections

The point-to-point configuration is the simplest MD485-to-AVW200 network. In this configuration, two MD485s are required (see Figure H-2 [p. 111]).

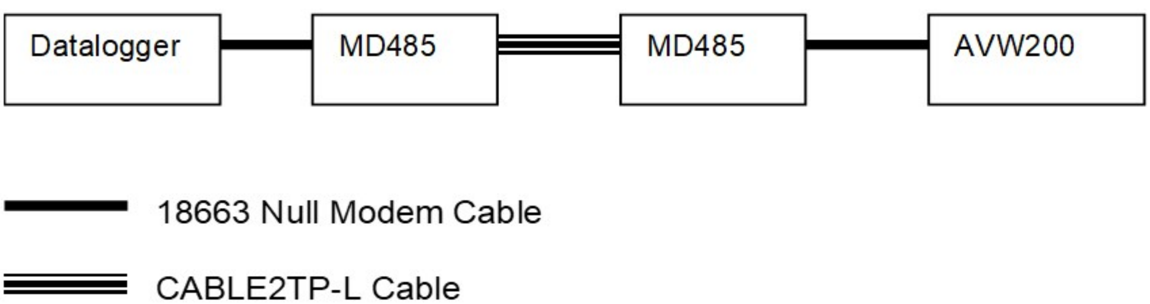


Figure H-2. Point-to-point MD485-to-AVW200 network

> **NOTE:**
> The CR1000Xe and CR6 feature an RS-232/CPI port that requires a 31056 cable (with a female DB9 socket) to connect the data logger's RS-232/CPI port to the 18633 null modem cable.

The point-to-multipoint configuration uses several AVW200s. This configuration requires one MD485 to connect with the data logger and another MD485 for each AVW200 in the network (see Figure H-3 [p. 112]).

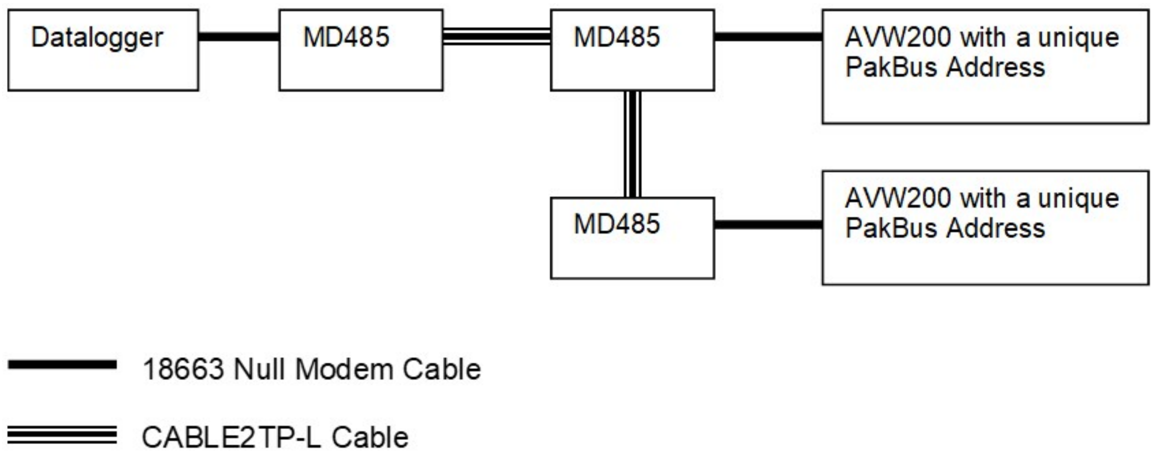


Figure H-3. Point-to-multipoint MD485-to-AVW200 network

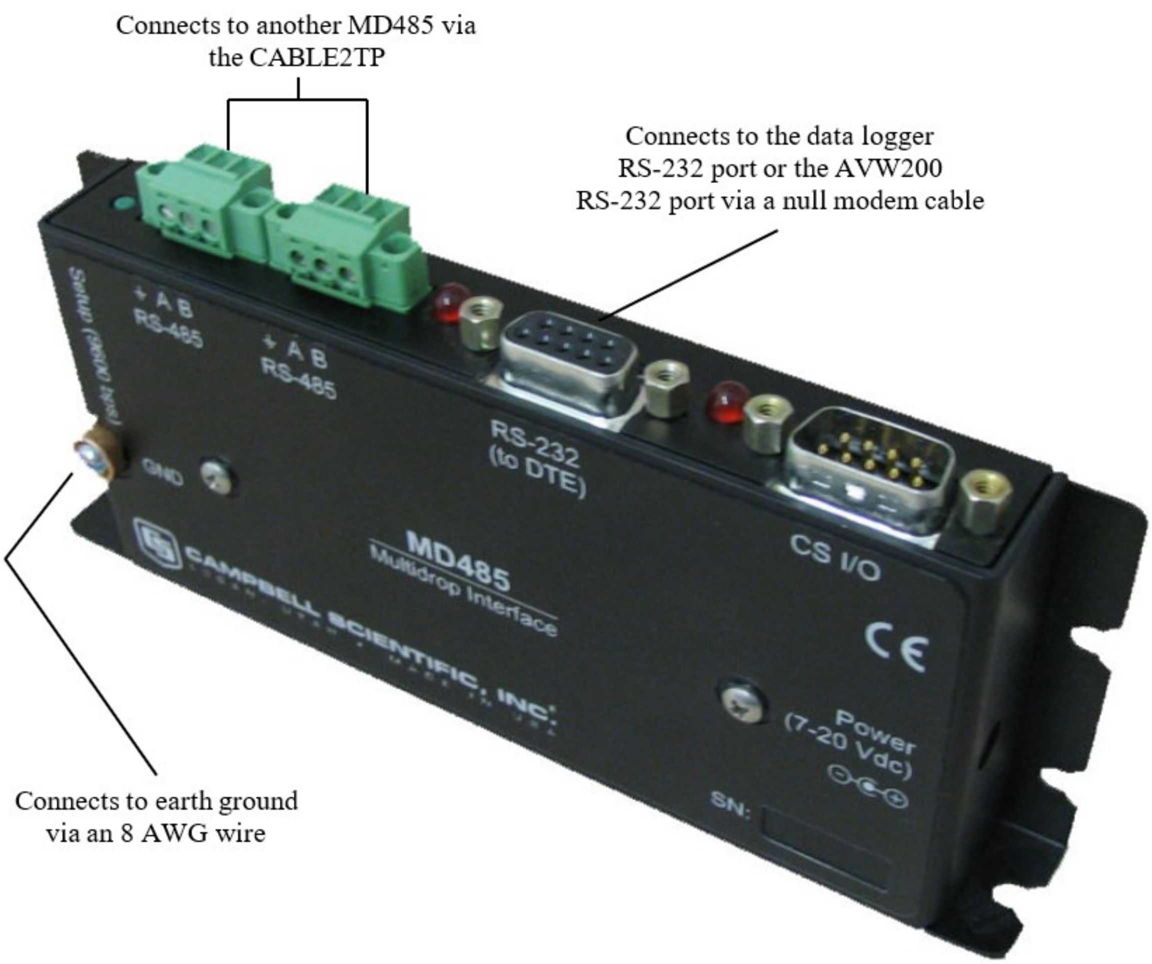


Figure H-4. MD485 and its connectors

H.2.1 Data logger to MD485

The null modem cable is used to connect an MD485 with the CR800, CR850, CR1000, or CR3000 data logger. One end of the null modem cable attaches to the RS-232 port on the MD485; the other end attaches to the **RS-232** port on the data logger (see Figure H-4 [p. 113]).

H.2.2 MD485 to MD485

The connection between MD485s is made with a CABLE2TP two-twisted pair cable with shield. Insulation colors are red/back and green/white. One pair is used for the differential data ("A" connects to "A", "B" connects to "B"), and one line of the other twisted pair is used for the signal ground (third connection on the MD485 terminal block). This is shown in Figure H-4 (p. 113) and Figure H-5 (p. 114). The cable shield should be connected to a chassis or earth ground (NOT the signal ground) at one end as shown in Figure H-5 (p. 114).

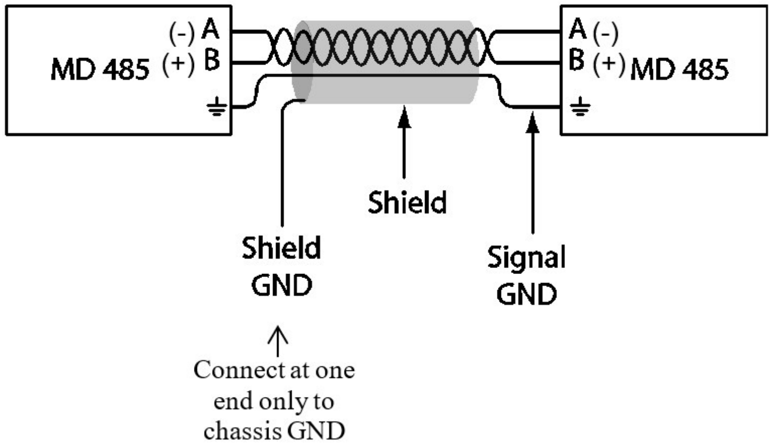


Figure H-5. MD485-to-MD485 connections and grounding

H.2.3 MD485 to AVW200

The null modem cable is used to connect an MD485 with an AVW200. One end of the null modem cable attaches to the **RS-232** port on the MD485; the other end attaches to the **RS-232** port on the AVW200 (see Figure H-4 [p. 113]).

H.2.4 Multiplexer connections

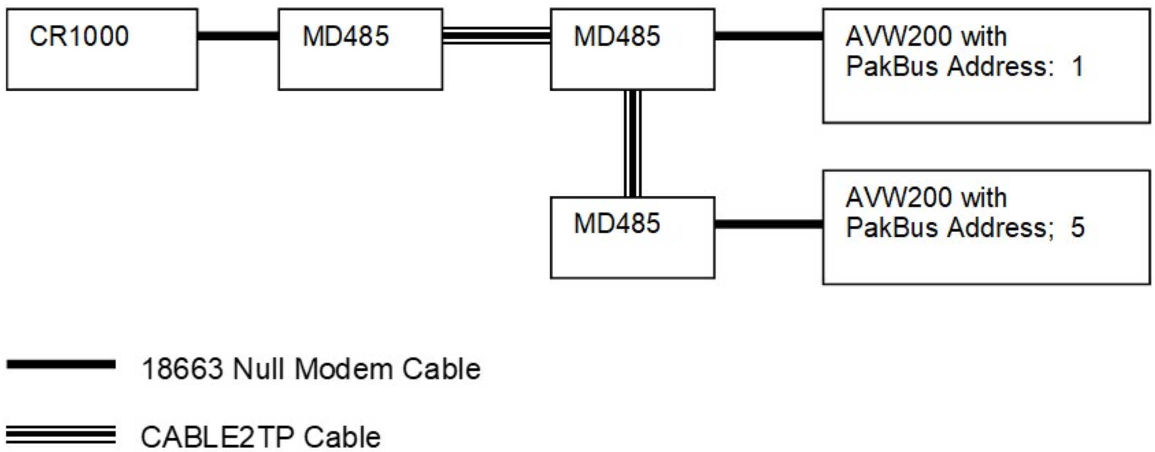
Multiplexers can be used in this configuration, but the AVW200 must control the multiplexers. Refer to AVW200 control of the multiplexer (p. 29) for information on connecting the multiplexers.

H.3 Programming

An `AVW200()` instruction is entered for each AVW200. The `ComPort` parameter needs to be **ComRS232**.

H.3.1 MD485 to AVW200 example program

The example is for a network that looks like the following:



CRBasic Example: MD485 to AVW200

```
'CR1000X Series Data Logger
'This program measures 2 sensors on 2 AVW200s PBA1 and PBA5.
'Each AVW200 is connected to a MD485 via a RS-232 null modem. The RS-232 baud
'rate of the MD485s is 38.4 k as is the RS-485 baud rate. The MD485s are set to
'Pakbus Network

'the Thermistor reading is converted from Ohms to Deg C

Public PTemp, batt_volt
Public Dst(2,6)
Public TempC(2), TempK(2)
Public Rf(2)
Public result(2)
'ABC=temp coefficients for the Steinhart-Hart equation to convert Ohms to TempC
Const A=.001403040
Const B=.000237318
Const C=.00000009

DataTable (AVW200,1,-1)
  DataInterval (0,10,Sec,10)
  Sample (6,Dst(1,1),IEEE4)
  Sample (6,Dst(2,1),IEEE4)
EndTable

DataTable (AVWcard,1,-1)
  CardOut (0 ,-1)
  DataInterval (0,10,Sec,10)
  Sample (6,Dst(1,1),IEEE4)
  Sample (6,Dst(2,1),IEEE4)
```

CRBasic Example: MD485 to AVW200

```
EndTable

BeginProg
  SerialOpen (ComRS232,38400,0,0,10000)
  Scan (10,Sec,0,0)
    PanelTemp (PTemp,60)
    Battery (batt_volt)
    'Result, comport,neighbor,PBA,Dst,chan,muxchan,reps,begFreq,endFreq,Vx,
    'IntegrationTime,Mult,Offset)
    'sensor 1, channel 1
    AVW200(result(1),ComRS232,1,1,Dst(1,1),1,1,1,1000,3500,2,_60HZ,1,0)
    'sensor 2, channel 2
    AVW200(result(2),ComRS232,5,5,Dst(2,1),1,1,1,1000,3500,2,_60HZ,1,0)
    Rf(1)=Dst(1,6)
    Rf(2)=Dst(2,6)
    TempK(1) = 1/(A + B*LN(Rf(1)) + C*(LN(Rf(1)))^3)
    TempK(2) = 1/(A + B*LN(Rf(2)) + C*(LN(Rf(2)))^3)
    TempC(1) = TempK(1)-273.15
    TempC(2) = TempK(2)-273.15

    CallTable AVW200
    CallTable AVWcard

  NextScan
EndProg
```

Limited warranty

Covered equipment is warranted/guaranteed against defects in materials and workmanship under normal use and service for the period listed on your sales invoice or the product order information web page. The covered period begins on the date of shipment unless otherwise specified. For a repair to be covered under warranty, the following criteria must be met:

1. There must be a defect in materials or workmanship that affects form, fit, or function of the device.

2. The defect cannot be the result of misuse.

3. The defect must have occurred within a specified period of time; and

4. The determination must be made by a qualified technician at a Campbell Scientific Service Center/ repair facility.

The following is not covered:

1. Equipment which has been modified or altered in any way without the written permission of Campbell Scientific.

2. Batteries; and

3. Any equipment which has been subjected to misuse, neglect, acts of God or damage in transit.

Campbell Scientific regional offices handle repairs for customers within their territories. Please see the back page of the manual for a list of regional offices or visit www.campbellsci.com/contact to determine which Campbell Scientific office serves your country. For directions on how to return equipment, see Assistance.

Other manufacturer's products, that are resold by Campbell Scientific, are warranted only to the limits extended by the original manufacturer.

CAMPBELL SCIENTIFIC EXPRESSLY DISCLAIMS AND EXCLUDES ANY IMPLIED WARRANTIES OF

MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Campbell Scientific 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.

Campbell Scientific will, as a default, return warranted equipment by surface carrier prepaid. However, the method of return shipment is at Campbell Scientific's sole discretion. Campbell Scientific will not reimburse the claimant for costs incurred in removing and/or reinstalling equipment. This warranty and the Company’s obligation thereunder is in lieu of all other warranties, expressed or implied, including those of suitability and fitness for a particular purpose. Campbell Scientific is not liable for consequential damage.

In the event of any conflict or inconsistency between the provisions of this Warranty and the provisions of Campbell Scientific's Terms, the provisions of Campbell Scientific's Terms shall prevail. Furthermore, Campbell Scientific's Terms are hereby incorporated by reference into this Warranty. To view Terms and conditions that apply to Campbell Scientific, Logan, UT, USA, see Terms and Conditions. To view terms and conditions that apply to Campbell Scientific offices outside of the United States, contact the regional office that serves your country.

Assistance

Products may not be returned without prior authorization. Please inform us before returning equipment and obtain a **return material authorization (RMA) number** whether the repair is under warranty/guarantee or not. See Limited warranty for information on covered equipment.

Campbell Scientific regional offices handle repairs for customers within their territories. Please see the back page of the manual for a list of regional offices or visit www.campbellsci.com/contact to determine which Campbell Scientific office serves your country.

When returning equipment, a RMA number must be clearly marked on the outside of the package. Please state the faults as clearly as possible. Quotations for repairs can be given on request.

It is the policy of Campbell Scientific to protect the health of its employees and provide a safe working environment. In support of this policy, when equipment is returned to Campbell Scientific, Logan, UT, USA, it is mandatory that a "Declaration of Hazardous Material and Decontamination" form be received before the return can be processed. If the form is not received within 5 working days of product receipt or is incomplete, the product will be returned to the customer at the customer's expense. For details on decontamination standards specific to your country, please reach out to your regional Campbell Scientific office.

> **NOTE:**
> All goods that cross trade boundaries may be subject to some form of fee (customs clearance, duties or import tax). Also, some regional offices require a purchase order upfront if a product is out of the warranty period. Please contact your regional Campbell Scientific office for details.

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 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

- Protect from over-voltage.
- Protect electrical equipment from water.
- Protect from electrostatic discharge (ESD).
- Protect from lightning.
- 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, 6 meters (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.
- Only use power sources approved for use in the country of installation to power Campbell Scientific devices.

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.

Internal Battery

- Be aware of fire, explosion, and severe-burn hazards.
- Misuse or improper installation of the internal lithium battery can cause severe injury.

- Do not recharge, disassemble, heat above 100 °C (212 °F), solder directly to the cell, incinerate, or expose contents to water. Dispose of spent batteries properly.

Use and disposal of batteries

- Where batteries need to be transported to the installation site, ensure they are packed to prevent the battery terminals shorting which could cause a fire or explosion. Especially in the case of lithium batteries, ensure they are packed and transported in a way that complies with local shipping regulations and the safety requirements of the carriers involved.
- When installing the batteries follow the installation instructions very carefully. This is to avoid risk of damage to the equipment caused by installing the wrong type of battery or reverse connections.
- When disposing of used batteries, it is still important to avoid the risk of shorting. Do not dispose of the batteries in a fire as there is risk of explosion and leakage of harmful chemicals into the environment. Batteries should be disposed of at registered recycling facilities.

Avoiding unnecessary exposure to radio transmitter radiation

- Where the equipment includes a radio transmitter, precautions should be taken to avoid unnecessary exposure to radiation from the antenna. The degree of caution required varies with the power of the transmitter, but as a rule it is best to avoid getting closer to the antenna than 20 cm (8 inches) when the antenna is active. In particular keep your head away from the antenna. For higher power radios (in excess of 1 W ERP) turn the radio off when servicing the system, unless the antenna is installed away from the station, e.g. it is mounted above the system on an arm or pole.

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.

Global Sales and Support Network

A worldwide network to help meet your needs

Campbell Scientific Regional Offices

Australia

Location: Garbutt, QLD Australia
Phone: 61.7.4401.7700
Email: info@campbellsci.com.au
Website: www.campbellsci.com.au

Brazil

Location: São Paulo, SP Brazil
Phone: 11.3732.3399
Email: vendas@campbellsci.com.br
Website: www.campbellsci.com.br

Canada

Location: Edmonton, AB Canada
Phone: 780.454.2505
Email: dataloggers@campbellsci.ca
Website: www.campbellsci.ca

China

Location: Beijing, P. R. China
Phone: 86.10.6561.0080
Email: info@campbellsci.com.cn
Website: www.campbellsci.com.cn

Costa Rica

Location: San Pedro, Costa Rica
Phone: 506.2280.1564
Email: info@campbellsci.cc
Website: www.campbellsci.cc

France

Location: Montrouge, France
Phone: 0033.0.1.56.45.15.20
Email: info@campbellsci.fr
Website: www.campbellsci.fr

Germany

Location: Bremen, Germany
Phone: 49.0.421.460974.0
Email: info@campbellsci.de
Website: www.campbellsci.de

India

Location: New Delhi, DL India
Phone: 91.11.46500481.482
Email: info@campbellsci.in
Website: www.campbellsci.in

Japan

Location: Kawagishi, Toda City, Japan
Phone: 048.400.5001
Email: jp-info@campbellsci.com
Website: www.campbellsci.co.jp

South Africa

Location: Stellenbosch, South Africa
Phone: 27.21.8809960
Email: sales@campbellsci.co.za
Website: www.campbellsci.co.za

Spain

Location: Barcelona, Spain
Phone: 34.93.2323938
Email: info@campbellsci.es
Website: www.campbellsci.es

Thailand

Location: Bangkok, Thailand
Phone: 66.2.719.3399
Email: info@campbellsci.asia
Website: www.campbellsci.asia

UK

Location: Shepshed, Loughborough, UK
Phone: 44.0.1509.601141
Email: sales@campbellsci.co.uk
Website: www.campbellsci.co.uk

USA

Location: Logan, UT USA
Phone: 435.227.9120
Email: info@campbellsci.com
Website: www.campbellsci.com