User Guide

Integration of ShapeAccelArray and Campbell Scientific CR800/CR1000 Data Loggers





Copyright © 2012 by Measurand Inc.



Notices

Measurand shall have no liability for incidental or consequential damages of any kind arising out of the sale, installation, or use of its products. Please read this document and any notes and instructions carefully before proceeding with installation and operation. The information herein is subject to change without prior notification.

ShapeAccelArray is covered by patents including: 6127672, 6563107, 7296363, WO 02/055958, WO 02/055958, WO 98/41815 and others pending.

Measurand Software is copyrighted. Any unauthorized use is strictly prohibited.

Revisions No: 1



Table of Contents

Tab	le of Contentsiii						
List	ist of Figuresiv						
Pref	⁻ acev						
1.	Introduction1						
2.	Hardware Setup1						
3.	Checking SAA Performance Prior to Installation4						
4.	Using LoggerNet7						
5.	Creating a Main Program File7						
6.	Compiling SAA Program Files						
7.	Uploading Program Files to the CR100010						
8.	Data File Management13						
9.	Manual Data Collection14						
10.	Automated Data Collection16						
11.	Manual Conversions of SAA Data17						
12.	Viewing SAA Data18						
13.	Automated Data Conversions19						
14.	Saving Data in Overwrite File Mode22						
Арр	endix I: Integrating other Sensors into the Main Program (Advanced)						





List of Figures

Figure 2.1: SAA232 and SAA232-5 for connecting SAA to COM ports on CR800/CR1000 loggers. Note SAA wiring
diagram on the side panel1
Figure 2.2: Wiring diagram for CR1000 with SAA2322
Figure 2.3: Wiring diagram for CR1000 with SAA232-5
Figure 2.4: Example CR800 & SAA Integration using one SAAREG, two SAA232, a CR800 logger, and a Raven XT
Wireless Modem
Figure 3.1: SAA232-USB connector cable
Figure 3.2: Opening the Diagnostics Tests window in SAARecorder
Figure 3.3: Graph for Total Acceleration Check in SAARecorder
Figure 4.1: LoggerNet Utilities menu7
Figure 5.1: SAACR_FileGen utility
Figure 5.2: SAA232 or SAA232-5 Channel selection9
Figure 5.3: Saving a logger program created using the SAACR_FileGen utility9
Figure 6.1: Saving and Compiling the SAA logger program10
Figure 7.1 Selecting File Control in the LoggerNet Connect utility
Figure 7.2: Selecting 'Send' option in 'File Control'
Figure 7.3: Selecting 'Run Options' when sending files to CR1000 logger12
Figure 7.4: Sending a program to the CR100013
Figure 8.1: Specifying the destination of data files using the LoggerNet Setup utility14
Figure 9.1: Disabling scheduled data collection15
Figure 9.2: Collecting data using the LoggerNet 'Connect' utility
Figure 10.1: Enabling scheduled data collection17
Figure 12.1: Selecting an SAA to View in SAAView
Figure 12.2: Quickview of absolute shape plots for SAA in SAAView
Figure 12.3: Viewing absolute shape for SAA in Main View of SAAView19
Figure 13.1: Creating a task using the LoggerNet 'Task Master' Utility
Figure 13.2: Configuring task options in Task Master
Figure 13.3: Viewing the status of scheduled tasks in Task Master
Figure 14.1: Switching the "Data File Output" option in the LoggerNet 'Setup' utility to 'Overwrite Existing File'.





Preface

In many cases it is not convenient or desirable to collect data directly from an SAA using SAARecorder. In these cases, SAA data can be collected using Campbell Scientific's CR800 or CR1000 data loggers. This user guide presents information needed to correctly set up a Campbell Scientific logger and SAAs for data collection. In this guide, for simplicity's sake, only CR1000 is mentioned in most cases. Unless otherwise noted, it can be assumed that both the CR800 and CR1000 can be used in the same fashion with the same settings.



1. Introduction

ShapeAccelArrays (SAA) can be configured for use with a CR800 or CR1000. This is accomplished using a Measurand "SAA232" (or "SAA232-5") device and "SAACR1000 Integration Suite" software. The SAACR1000 Integration Suite includes:

- SAACR File Gen: a utility to create the program files that will allow the user to collect data from the SAAs sensors; and
- SAACR_raw2data: a conversion utility that converts the raw data to a format recognized by Measurand's SAA3D_Viewer software.

The CR1000 is advantageous for applications logging data from large amounts of various sensors. Additionally, the CR1000 allows up to 20 SAAs to be connected at a single time (the CR800 allows for up to 10 SAAFs connected via 2 SAA232-5s). The "Averaging in Array" (AIA) functionality in all SAAs manufactured after March 2010 makes it possible to collect a high number of averaged samples in only a few seconds. Arrays manufactured before March 2010 take up to 30 minutes to collect and average 1000 samples.

2. Hardware Setup

As previously mentioned up to 20 SAAs can be connected to a CR1000 logger via the 4 available COM ports. A CR800 has 2 COM ports, thus up to 10 SAAs can be connected in the same manner. Each COM port on the CR1000 requires a Measurand SAA232 or SAA232-5 device (Figure 2.1) to connect an SAA. The SAA232 interface is necessary to convert the SAA's RS485 communication protocol to the RS232 format recognized by the CR1000; in addition to providing surge protection and controlling power to the array. The power control aids in saving power and providing SAA microprocessor resets for trouble-free operation spanning years. A wiring diagram for connecting the SAA232 to the CR1000 and to the SAA is located on the side of the SAA232 or SAA232-5, as can be seen in Figure 2.1. Wiring diagrams for back panels containing CR1000, an SAAREG, and SAA232 or SAA232-5 are shown in Figure 2.2 and Figure 2.3, respectively.



Figure 2.1: SAA232 and SAA232-5 for connecting SAA to COM ports on CR800/CR1000 loggers. Note SAA wiring diagram on the side panel.





 Note 1: Ensure that SAAF serial numbers and COM port numbers are matched, as specified in the Main Program of the CR1000.
 Do not arbitrarily move SAAFs to other serial port COM numbers.
 Note 2: Solar Panel can be replaced with a mains charger if available

Note 3: Sealed lead acid deep cycle 12V battery

Figure 2.2: Wiring diagram for CR1000 with SAA232.

An example setup using a CR1000, 2 SAA232s, and an Airlink Raven XT modem is illustrated in Figure 2.4. If the back panel assembly and wiring has been done by Measurand, you will only need to connect the SAA wires to the SAA232 or SAA232-5 channel. This should be done prior to connecting the power to the logger (i.e. the 4-pin green connector on SAAREG should not be plugged in while you are wiring the SAAs into the SAA232 or SAA232-5). The green connectors can be removed from the SAA232 for easier wiring. When wiring the SAAs to the SAA232 and logger, make sure to note which SAA (serial number) is connected to which SAA232 or SAA232-5 channel (if applicable), and the COM port to which the SAA232 or SAA232-5 is connected. These must match the logger program.





Figure 2.3: Wiring diagram for CR1000 with SAA232-5.





Figure 2.4: Example CR800 & SAA Integration using one SAAREG, two SAA232, a CR800 logger, and a Raven XT Wireless Modem.

3. Checking SAA Performance Prior to Installation

Prior to installing the SAA, it is desirable to check the performance of the SAA to verify that no damage was incurred during shipping. This is done by using the Advanced Diagnostics applications in SAARecorder software. SAARecorder is freely available from the Measurand website at the following address <u>http://www.MeasurandGeotechnical.com/software.html</u>. In order to do this, you must connect



the SAA232 device to your laptop by means of a "SAA232-USB" Connector (Figure 3.1). The SAA232-USB 4-pin green connector plugs into the top receptacle on the SAA232. Power is supplied to the SAA232 by connecting the red (12V) and black (Ground) wires to available ports on the CR1000 (Figure 2.2). The USB connector is then plugged into an available USB port on a computer running SAARecorder. For more details about SAARecorder, consult the SAARecorder Manual included in the installation.

Once connected to the SAA using the SAA232-USB, open SAARecorder and connect to the SAA. In SAARecorder, click on Actions | Advanced | Diagnostic Test and then click on 'Total Acceleration Check' (Figure 3.2). A plot showing the total acceleration levels for each segment appears (Figure 3.3). Verify that all values lie between 0.93 G and 1.07 G. If this is not the case, please contact Measurand for further diagnostic support.

SAA roll calibration can also be performed by connecting to an SAA in this manner. SAA is shipped with a factory roll calibration, ready for installation. It is wise to check the roll calibration on subsequent installations. The method for doing so is described in the SAA Installations Guide, available from the Measurand website (<u>http://MeasurandGeotechnical.com/Installation Guide 2011.pdf</u>). If you have any issues with the calibration, please contact Measurand.



Figure 3.1: SAA232-USB connector cable.



at SAARecorder- 2-D Mode and a second s							
File Options Calil	brations View Actio	ons Help					
. 3000.0		Move Array Get Vibration Data Connect	•				· · · ·
2000.0		Advanced	Export Grou Concatenat Diagnostic	ıt Anchor Data te Horizontal Pose Data Test	a		•
1000 0				Z			
			· · · · · · · · · · · · · · · · · · ·				· · · · · · · · · · · · · · · · · · ·
-1000 0				X			,
-2000.0							· · · · · · · · · · · · · · · · · · ·
-3000.0			· · · · · · · · · · · · · · · · · · ·				· · · · · · · · · · · · · · · · · · ·
	-5782.7	-3855.1	-1927.6	0.0	1927.6	3855.1	5782.7

Figure 3.2: Opening the Diagnostics Tests window in SAARecorder.



Figure 3.3: Graph for Total Acceleration Check in SAARecorder.



4. Using LoggerNet

Setup of the CR1000 to collect SAA data requires a basic understanding of Campbell Scientific's LoggerNet software. For more information on Campbell Scientifics's LoggerNet software, visit the Campbell Scientific website to obtain the user manual (<u>http://www.campbellsci.ca/</u>). The LoggerNet toolbar is shown in Figure 4.1. For the purpose of this manual, the following utilities will be used:

- Setup: Utility for connecting to or changing connection settings of a Campbell Scientific device (e.g, the CR1000 or CR800). This utility is found in the Main Menu.
- Connect: Utility for managing device connection. This utility is found in the Main Menu.
- CRBasic Editor: Editor and compiler for Campbell Scientific programs. This utility is found in the Program Menu.
- Task Master: Utility will be discussed in Section 13. This utility is found in the Main Menu.



Figure 4.1: LoggerNet Utilities menu.

5. Creating a Main Program File

Program files for collecting SAA data using the CR1000 can be created using the Measurand SAACR_FileGen utility (Figure 5.1). The following fields must be completed before generating the file:

- Project Title: A descriptor of the monitoring setup;
- Number of Samples to Average: The number of SAA samples used to compute an average reading. Values can range from 100 to 25500. The larger the number of samples the better the precision, but computational time also increases, albeit, very minimally. With AIA technology, data samples can be collected much faster than previously available. For example, taking the average of 1000 readings (standard averaging level) will take approximately 10 s to output a value from the SAA;



SAACR	AACR_FileGen 2.11								
	OPEN CE	R1 / CR8 File	SAARecord	er Folder: c:\Measura	and Inc\Sa	aRecorder			Browse
				1 t		Project Tit	tle:		
I fo	his show older, if v	's the default loc You have installe	ation for th d SAAReco	e SAARecorder rder in another		Number o	f Samples to Averag	e:	1000
fo	older, yo	u must edit this	entry.			Reading I	nterval:	6	Hours -
	SAA	232(-5) Channel	COM Port	Serial Number		SAA	232(-5) Channel	COM Port	Serial Number
	1					🗖 11			
	2					12			
	3					🔲 13			
	4					🔲 14			
	5					🔲 15			
	6					🔲 16			
	7					17			
	8					🔲 18			
	9					19			
	10					20			
m	GENERATE CR1 / CR8 FILE								

Figure 5.1: SAACR_FileGen utility.

- Reading Interval: The frequency at which SAA observations are made within the program (also known as 'scan interval'). An SAAF can perform measurements at 400Hz (see above), thus taking an average of 1000 samples (typical value) in less than 10 seconds. However, sufficient time must be included to account for other commands and functions within the CR1000. The number of SAAFs connected to a CR1000 also affects the collection time. Based on past experience, the logger also needs time to process all of its commands and has been known to crash when the reading interval was less than 10 minutes. For this reason, the minimum reading interval is 10 minutes.
- SAA Table: For each SAA, the following information must be included, (as seen in Figure 5.1):
 - 1. SAA232 or SAA232-5 Channel that the SAA is connected to (See Figure 5.2);
 - 2. COM port to which the SAA232 or SAA232-5 is connected on the CR1000; and
 - 3. Serial Number (physically marked on the 'cable-end' of the SAA)
- Generate CR1/CR8 File: The main difference between the CR800 and CR1000 is the number of COM ports available, and thus the number of SAAs that be connected to each. For a CR1000, up to 20 SAAs can be connected at one time, but for a CR800 only 10 SAAs. (Please contact Measurand or Campbell Scientific for more information on the differences between the two). To choose the data logger, click the "Generate CR1/CR8 File" button, and select the desired file type in the save screen (Figure 5.3). An SAA_Include file will also be generated at this time.

The SAACR_FileGen utility can be used to create programs for SAAs and SAAPs (Measurand vibrating wire piezometers). Once the program is completed, it is possible to add coding for other instruments into the program.



SAACE	FileGen 2.11			and the second s			100	-		x
	OPEN CB1 / CB8 File		SAARecord	AARecorder Folder: c:\Measurand Inc\SaaRecorder					Brow	se
						Project Ti	tle: Test Set-Up			
						Number o	f Samples to Averag	e:	1000	
						Reading	nterval:	1	Hours	•
	SAA	232(-5) Channel	COM Port	Serial Number		SAA	232(-5) Channel	COM Port	Serial Number	
	☑ 1	SAA232	COM1 👻	53852		🔲 11				
	2	SAA232 SAA232-5 Chan 1				12				
	3	SAA232-5 Chan 2 SAA232-5 Chan 3				🔲 13				
	4	SAA232-5 Chan 4 SAA232-5 Chan 5				14				
	5					15				
	6					16				
	7					17				
	8					18				
	9					19				
	10					20				
m	easu	and						GENERA	TE CR1 / CR8 FI	LE



Save CS Datalogger File								
○○ - ↓ Co	mputer	▶ Local Disk (C:) ▶ Campbellsci	LoggerNet 🕨	• 4	Search LoggerNet	٩		
Organize ▼ New folder								
☆ Favorites	^	Name	Date modified	Туре	Size			
🧮 Desktop		WSAAWCR1000Test.cr1	4/30/2012 10:05 AM	CR1 File	13 KB			
🗼 Downloads		SAAPtest.cr1	4/16/2012 10:28 AM	CR1 File	15 KB			
💠 Dropbox		SAA_Include_V3_03.cr1	3/5/2012 3:58 PM	CR1 File	52 KB			
📃 Recent Places	=	test.cr1	12/9/2011 3:32 PM	CR1 File	12 KB			
		SAA_Include_V3_01.cr1	12/5/2011 9:47 AM	CR1 File	52 KB			
詞 Libraries		鷆 New folder	4/28/2012 1:07 AM	File folder				
Documents		퉬 NetworkMapBackup	4/28/2012 1:06 AM	File folder				
J Music		퉬 Logs	4/28/2012 1:06 AM	File folder				
Pictures		퉬 Data	4/28/2012 1:05 AM	File folder				
Videos		퉬 sys	4/26/2012 2:32 PM	File folder				
Computer								
File name:	File name: CRtest							
Save as type:	Save as type: CR1000 Files (*.cr1)							
Alide Folders	CR800 All files	Files (*.cr8) (*.*)				.:		

Figure 5.3: Saving a logger program created using the SAACR_FileGen utility.



6. Compiling SAA Program Files

Once the main program has been successfully generated, it can be viewed using the LoggerNet 'CRBasic Editor'. At this time, it is also possible to add other code for instruments or powering the SW12V port on the logger. Once the code has been added, it should be compiled to make sure that there are no errors. To successfully compile the new program, the SAA_Include file generated with the program is also required. The SAA_Include file should be placed in the same directory as the main program to avoid compilation errors. The SAA_Include file will not successfully compile alone, it gets compiled as part of the main program. Once the two files are located within the same directory, it can be attempted to save and compile the program as illustrated in Figure 6.1.



Note: The SAA_Include file is not meant to compile on its own. The SAA_Include file should be placed in the same directory as the main program file. When the SAA_Communicator program is compiled, the SAA_Include is referenced and is compiled as well. Compile errors will appear if it is attempted to compile the SAA_Include file.



Figure 6.1: Saving and Compiling the SAA logger program.



Note: When using other sensors with the CR1000, it is advised to add the necessary code to the CR1000 program generated for the SAA rather than trying to integrate the SAA specific program with the other sensors. Appendix I provides examples of integrating other sensors.

7. Uploading Program Files to the CR1000

Once the program has successfully compiled, both the program file and the SAA_Include file must be uploaded to the CR1000 before the main program is run. This is accomplished in LoggerNet using the 'Connect' utility (Figure 7.1). Right click on the Station of interest and click on 'File Control' (the station must be setup using the LoggerNet 'Setup' utility).



Connect Screen: CR800Series_qa (CR8 File_Edit View Datalogger Help	300Series)	
Connect Collect Now	Station Statul File Control Num Display Graphs	Ports & Flags
Stations	Table Monitor: Passive Monitoring	Clocks Adjusted Server Date/Time
	Field Value	Station Date/Time
	1. Select Logger you want to connect to. 2. Click "Connect"	Check Set
	3. Once connected, click "File Control"	Program No Program
		Send Retrieve
List Alphabetically		A
~0 00:00:00	Stop Interval 00 m 01 s 🚽	
/iew and control the station's file system		

Figure 7.1 Selecting File Control in the LoggerNet Connect utility.

Once the File Control window opens, the 'Send' option is used to place the SAA_Include file on the CR1000, as illustrated in Figure 7.2. The 'Run Now' and 'Run On Power-up' options should be deselected when prompted (Figure 7.3) (since the SAA_Include file will not run as a standalone application).



Note: The SAA_Include file must be placed on the CR1000 before the main program will run. This is accomplished using 'File Control' from the Connect utility.

The main program file can be placed in the same way by using the 'File Control' and 'Send' options. When uploading the main program file, the 'Run Now' and 'Run On Power-up' options should be selected. Note since the run program options are selected, the main program will compile at this point (Figure 7.4). Alternatively the 'Save, Compile and Send' option in the CRBasic Editor can be used. When prompted for Run Options, 'Run Now' and 'Run On Power-up' should be selected.



Note: If you run the program to test the logger set-up, make sure to remove garbage data from the logger by clicking the 'Stop Program' button. When prompted, choose the 'Clear the Program' option. When finished setting up the site, click 'Run Options' and select 'Run Now' and 'Run on Power-up' to start collecting data. To make sure no garbage data is left behind, when prompted select 'Delete associated data tables created by'.





File Control:	CR800Series_qa	I							Ω Σ	8
? <u>S</u> end	🥌 Eormat	(3) R <u>e</u> fresh	<u>R</u> etrieve	R <u>u</u> n	ф Options	 <u>D</u> elete	Stop Program	. <u>H</u> elp		
Device	Bytes Free		File Name		Run Options		Size	Modified	Attribut	es
V Set Run Optio	ons on Send									

Figure 7.2: Selecting 'Send' option in 'File Control'.

Select Run Options		×					
SAA_Include_V3_03.cr8							
🔲 Run Now							
Preserve	Preserve data if no table changed						
🔿 Delete as	Delete associated data tables created by:						
Run On Power-up							
	OK Cance						

Figure 7.3: Selecting 'Run Options' when sending files to CR1000 logger.



le Control: Measurand_Test_CR10			Liocks
ቻ 🥶 🔇			Show Units Adjusted Server Date/Time O1/01/1590 00:00 Station Date/Time
ice Bytes Free 417280	File Name Run Options SAA Include V2 64.cr1	Size Modified Attributes 59,097 KB 2011-09-02 10:01:40 Rv/	01/01/1990 00:00
16384	SAA_Communicator_M	14.36 KB 2011-09-02 10.08.32 RW	Pause Clock Update
			Program No Program
		Sending File	Send Ref
	_		Notes
t Run Options on Send		Cancel	
	hanging the file's run options		

Figure 7.4: Sending a program to the CR1000.

8. Data File Management

It is critical that the data files generated by the CR1000 are effectively managed. The output folders for CR1000 data files are specified under the 'Data Files' tab in the LoggerNet 'Setup' utility (Figure 8.1). By default, the data is stored in the C:\CampbellSci\LoggerNet directory and given a name of "DataloggerName_TableName_#.dat". It is suggested that the output folder location be changed from this default location to another folder prior to collecting data. It is generally easiest to create a data folder in the C:\Measurand Inc\SAA3D\logger_files\ directory to facilitate automated data conversions later on. An 'archive_raw' folder should be created in the data folder for conversion purposes. To change the output file name you will need to select each file in the 'Tables' box shown in Figure 8.1 and point it towards the desired output folder individually using the 'Output File Name'. It is important not to change the table name, but only the folder. For example the Project Information file is saved by default to C:\Campbellsci\LoggerNet\CR1000_PROJECT_INFO.DAT. This can be changed to C:\Measurand Inc\SAA3D\logger_files\Test\CR1000_PROJECT_INFO.DAT.



Note: If you opt to change the output file name, DO NOT change the output table name, only change the folder location.

The 'File Output' option should be set to "Append to End of File" (Figure 8.1). The CR1000 keeps track of the last record that was successfully sent to the destination folder. If the output file already exists, the new data will be appended to the end of the file. If the file does not exist or has been moved elsewhere, a new file will be created with all records that have not yet been recorded.

The 'Output Format' should be left at the default ASCII Table Data, Long Header (TOA5).



EZSetup Wizard - CR800Series_flnrbc (CR800Series)					
Progress	Datalogger Table Output Files				
Introduction	For each table listed, select a file name to which data will be written, and check the bay if you wish to have the table collected during scheduled collection.				
Communication Setup	the box if you wish to have the table collected during scheduled collection.				
Datalogger Settings	Tables Table Collected During Data Collection Table Collected During Data Collection				
Setup Summary	VPROJECT_INFO Data File Option Data File Option				
Communication Test	✓PZ3_DATA				
Datalogger Clock	\[SAA1_DIAGNOSTICS \[J \[
Send Program	√SAA2_DIAGNOSTICS √SAA3_DATA ▼				
🕈 Data Files					
Scheduled Collection	Get Table Definitions Note: If table definitions are empty or out of date, you may wish to get them from the datalogger. Click the				
Wizard Complete	button to get them.				
	Previous Next Finish Cancel Collection <u>H</u> elp				

Figure 8.1: Specifying the destination of data files using the LoggerNet Setup utility



Note: Before a new project begins, any CR1000 data files residing in the output file directory should be removed. Otherwise, data from the new project may get concatenated to the existing data file, causing abnormalities in the results.

9. Manual Data Collection

For each setup, the PROJECT_INFO, SAAx_DATA and SERIAL_ERRORS data files must be recorded. Data files scheduled to be collected are listed in the 'Data Files' tab of the Setup utility and have a green 'check mark' next to them. Data Tables "Public" and" Status" are not required for data collections as they only store one record of data at a time. They are used only for real time viewing. Ensure that these tables are not selected for Scheduled Collection by de-selecting the 'Included for Scheduled collection' option for that table.

Until data collection is properly configured, scheduled data collection should be disabled. This is accomplished by unchecking the 'Scheduled Collection Enabled' checkbox in the Schedule tab of the LoggerNet 'Setup' utility (Figure 9.1). Data can only be collected now by forcing a data download in the LoggerNet 'Connect' utility by clicking on 'Collect Now' (Figure 9.2).

Whenever the CR1000 is connected using the 'Connect' utility in LoggerNet, the station's clock time should be checked to ensure that valid time stamps are being applied to the data. Correct time stamps cannot be easily recovered from erroneous values. The CR1000's clock is set by clicking on 'Set Station Clock' in the Connect utility, which will assign the time currently set on your PC to the CR1000.





Note: If your PC is using Daylight Savings Time, take care NOT to disrupt the time setting currently employed by the CR1000. If the station clock is adjusted to match Daylight Savings Time on your PC, gaps or overwrites in the data file will occur. If you need to adjust the station clock, make sure you set your PC clock to Standard Time.

EZSetup Wizard - CR800Series_flnrbc (CR800Series)							
Progress	Scheduled Collection						
Introduction							
Communication Setup	Scheduled Collection Enabled	If you wish to have scheduled collection enabled for this datalogger check the box.					
Datalogger Settings	When to Collect						
Setup Summary	Base Date Time	Enter the base date and time. This is when the first attempt to collect will					
Communication Test	4/26/2012 12:00:00 AM 🚔	occur. If the base is in the past, data collection will be attempted immediately					
Datalogger Clock		when the schedule is enabled.					
Send Program	Collection Interval	Enter the Collection Interval. The collection interval is relative to the Base					
Data Files		Date and Time entries. For instance, if the Base Time is set at 12:15 and the					
Scheduled Collection		interval is set for 1 hour, data collection will be attempted at 12:15, 1:15, etc.					
Wizard Complete							
◀ Previous Next ▶ Finish Cancel Schedule <u>H</u> elp							

Figure 9.1: Disabling scheduled data collection.

Once sufficient time has elapsed to ensure that data has been collected from the SAAs (based upon the reading frequency), data can be manually collected by clicking on 'Collect Now' in the Connect utility. A summary of the data collected is summarized as illustrated in Figure 9.2.



Note: Data is collected based upon scheduled interval, not upon elapsed time since the program was started. If the program is written so that data is collected every 30 minutes and it is 29 minutes after the hour, data will be collected in 1 minute. If it is 31 minutes after the hour, data will be collected in 29 minutes.



Connect Screen: CR1000_BostonUK (C	R1000)	
File Edit View Datalogger Help		
Disconnect Collect Now	Station Status File Control Num Display Graphs	Ports & Flags
Stations	Table Monitor: Real Time Monitoring	Clocks Adjusted Server Date/Time 5/7/2012 12:55:10 PM Station Date/Time 5/7/2012 4:55:34 PM
Collecting Data	Collecting data from CR1000_	cted
List Alphabetically	Stop Interval 00 m 01 s 💌	
Collect any new data logged by the station	1	

Figure 9.2: Collecting data using the LoggerNet 'Connect' utility.

10. Automated Data Collection

Once the manual data collection process has been tested, the data collection process can be automated by specifying the desired data collection interval in the 'Schedule' tab of the LoggerNet 'Setup' utility (Figure 10.1). Check the 'Scheduled Collection Enabled' checkbox.

As long as the destination CR1000 is connected (wired or wirelessly), data will be downloaded to the Output File Directory. The Campbell Scientific Raven X manual provides details of configuring a wireless connection with the CR1000. The Campbell Scientific LoggerNet manual provides details of configuring a connection with the CR1000 using a land line. A trial test should always be conducted before beginning a project to ensure that the system is performing as expected.



EZSetup Wizard - CR1000_Bost	tonUK (CR1000)	
Progress	Scheduled Collection	
Introduction		
Communication Setup	Scheduled Collection Enabled	If you wish to have scheduled collection enabled for this datalogger check the box.
Datalogger Settings	When to Collect	
Setup Summary	Base Date Time	Enter the base date and time. This is when the first attempt to collect will
Communication Test	4/26/2012 12:00:00 AM 🚔	occur. If the base is in the past, data collection will be attempted immediately
Datalogger Clock		when the schedule is enabled.
Send Program	Collection Interval	Enter the Collection Interval. The collection interval is relative to the Base
Data Files		Date and Time entries. For instance, if the Base Time is set at 12:15 and the
Scheduled Collection		interval is set for 1 hour, data collection will be attempted at 12:15, 1:15, etc.
Wizard Complete		
I		
	✓ Previous Next ► Finish	Cancel Schedule <u>H</u> elp

Figure 10.1: Enabling scheduled data collection.

11. Manual Conversions of SAA Data

SAA data collected using the CR1000 can be manually converted to a format viewable in the Measurand SAA3D or SAAView viewer software using the Measurand SAACR_raw2data utility. Detailed instructions are found in the SAACR_raw2data instruction manual (located in the installation directory). Before data can be converted, calibration files for your SAAs must reside on your PC. Calibration files are most easily attained by downloading SAARecorder software from www.MeasurandGeotechnical.com website. SAARecorder will also be used to perform a pre-installation diagnostic check. Additionally, SAA3D software must also be downloaded from the website.

It is possible to use the data converter to convert SAA data, and data from the following instruments: vibrating wire piezometers, strain gauges, and vibrating wire earth pressure cells. It is also possible to convert multiple SAAs into one long "superset" array. In order to use these functionalities you will need to create a site file for the project and place it in the appropriate SAARecorder folder (c:\Measurand Inc\SAARecorder\Calibrations\site\[site filename]). N.B. if the site folder does not exist in the c:\Measurand Inc.\SAARecorder\Calibrations folder, you will need to create it to save your site files there. You will also need to place the calibration files for the other instruments into the appropriate SAARecorder calibrations folder (c:\Measurand Inc\SAARecorder\Calibrations\source folder (c:\Measurand Inc\SAARecorder calibrations folder (c:\Measurand Inc\SAARecorder\Calibrations\source folder (c:\Measurand Inc\SAARecorder calibrations folder (c:\Measurand Inc\SAARecorder\Calibrations\source folder calibrations folder (c:\Measurand Inc\SAARecorder\Calibrations\source folder (c:\Measurand Inc\SAARecorder\Calibrations\source folder calibrations folder (c:\Measurand Inc\SAARecorder\Calibrations\source folder calibrations folder (c:\Measurand Inc\SAARecorder\Calibrations\source folder calibrations folder (c:\Measurand Inc\SAARecorder\Calibrations\source folder calibrations\source folder calibrations folder (c:\Measurand Inc\SAARecorder\Calibrations\source folder calibrations\source folder calibrations folder (c:\Measurand Inc\SAARecorder\Calibrations\source folder calibrations\source folder calibrations\source folder calibrations\source folder calibrations folder (c:\Measurand Inc\SAARecorder\Calibrations\source folder calibrations\source folder calibrations\source folder calibrations fo



Once the conversion is completed, you will be prompted on whether or not you want to view the data using either SAA3D or SAAView.

12. Viewing SAA Data

When proceeding to view SAA data directly after a SAACR_raw2data conversion, you will be prompted to select which SAA viewer you would like to use. In most cases, you can use either SAAView or SAA3D. However, if you are converting convergence data, you can only view the data in SAAView. Complete instructions on how to use SAAView and SAA3D are given in their respective manuals, which can be found in the SAA3D folder.

After setting up a monitoring station, it is sometimes helpful to look at the absolute shape of the array in the ground. To do this, open SAAView and select the SAA you wish to view (Figure 12.1). To look at the absolute shape of the array, click on the 'QuickView' button. This will open the window shown in Figure 12.2. The 3D shape of the array is plotted on two graphs, the second plot showing the orthogonal view of the first one. It is possible to view the X-Z position data plot and the Y-Z position data plot by clicking the X button at the bottom right of the screen (Figure 12.2). It is also possible to view the X-Y position data plots by clicking on the Z button at the bottom right of the screen. The plots shown in Figure 12.2 contain three solid lines. These are: a blue line representing the initial shape reading of the array, a red line representing the latest shape reading, and a black line representing the difference between the initial and latest readings. In order to view intermediate readings, you will need to use the 'ViewUnfiltered' or 'ViewFiltered' option in the main menu. Note that it is also possible to view the absolute shape of the SAA in the main view. This is done by selecting 'Absolute Shape' from the 'View' menu (Figure 12.3). Complete instructions for the use of SAAView and SAA3D are provided in their respective manuals.



Figure 12.1: Selecting an SAA to View in SAAView.





Figure 12.2: Quickview of absolute shape plots for SAA in SAAView.



Figure 12.3: Viewing absolute shape for SAA in Main View of SAAView

13. Automated Data Conversions

Once at least one successful manual conversion has been completed, SAA data can be automatically converted to a format viewable in SAA3D or SAAView software. This is accomplished using the



LoggerNet 'Task Master' utility. The first manual conversion is necessary to generate the pref_project.txt file which contains conversion settings for subsequent operations.

The procedure for automating the data conversion process is:

- a) Disable scheduled data collection in the LoggerNet 'Setup' utility
- b) Send program files (both main and SAA_Include) to the CR1000
- c) Set the path for data files in the LoggerNet 'Setup' utility

d) Manually download the data from the logger and perform a manual conversion. This will generate the required pref_project.txt file.

- e) Setup a new task using the Task Master utility (subsequently described)
- f) Manually download the data again.
- g) Enable scheduled data collection at the desired data collection interval.

If you do not follow this procedure you run the risk of having data gaps.

To add a task in the Task Master, click on the CR1000 of interest, and click the 'Add After' button. Under 'How Task_n Starts', in the Station Event Type, select the 'After Successful Call' Option (Figure 13.1).

Task Master (Paused)	
File Edit Help	
Setup Status Messages	
Tasks	How Task_1 Starts
Tasks Tasks Task_1	How Task_1 Starts Station Event Type After Successful Call Task Scheduled Base Date Base Time 5/ 9/2012 2 12:00:00 AM Schedule Interval 0 d 01 h 00 m 00 s What Task_1 Does File to Run: C:\Measurand Inc\SAA3D\SAACR_raw2data.exe Command Line: "C:\Measurand Inc\SAA3D\logger_files\Test\pref_project.txt" Start In: C:\Measurand Inc\SAA3D\
Add Scheduled Add After Delete	Configure Task Apply Cancel
> SetTaskSettingsAck for Task_1 (56428)	

Figure 13.1: Creating a task using the LoggerNet 'Task Master' Utility.



To configure the task, click on the 'Configure Task' button. Once the configure screen is open (Figure 13.2), check 'Execute File' and browse to C:\Measurand Inc\SAA3D\SAACR_raw2data.exe (or the appropriate folder containing this file). Under 'Command Line Options' you must specify the pref_project.txt file that was generated when the first manual conversion was completed. It should be located in the same directory as the raw data. This entry must be surrounded by double quotation marks ("") since there is a space in the C:\Measurand Inc\ path. This file contains settings necessary for SAACR_raw2data to complete subsequent conversions. Under 'Start In', enter C:\Measurand Inc\SAA3D\. Select 'Run Minimized' if you don't want to see the application running when it gets called. Hit 'Apply'. The Task Master will now automatically convert raw data that is scheduled to be collected using the LoggerNet 'Setup' utility.

Task Options	X
Execute File Call Stat	ion
Execute File	
File Name:	
C:\Measurand In	ic\SAA3D\SAACR_raw2data.exe
Command Line O	ptions:
"C:\Measurand I	nc\SAA3D\logger_files\Test\pref_pr
Start In:	
C:\Measurand In	ic\SAA3D\
🔽 Run Minimized	Ŀ
0	K Cancel Help

Figure 13.2: Configuring task options in Task Master.



Note: The path name to the pref_project.txt file must be surrounded by double quotation marks since there is a space in C:\Measurand Inc\

Verify that the 'Pause All Tasks' check box under the Status tab of the Task Master utility is unchecked, Figure 13.3. The SAACR_raw2data utility will now run upon successful completion of a scheduled data download. A progress bar will now appear after a scheduled data download indicating that the SAACR_raw2data utility is running. If any of the files required by SAACR_raw2data are open while the program is being executed an error message will be generated. Data can be viewed at any point by running the SAA3D and viewing the most recent data which has already been automatically converted.





Note: Any of the files required by SAACR_raw2data (E.g., calibration files, multi_saa_allcart.txt) cannot be open while the process is running. Otherwise an error will be generated.

Task Maste	r				
etup Status	Messages				
ask Name	Action	Event Type	Event Trigger	Last Time Run	Next Time to Run
ask_1	Execute File	After Successful	CR1000_Bosto		
		Ramo All Taska			
Run Sel	ected Lask	E Fause All Lasks			
TaskStatus	EnumerateAck f	for Task_1 (56428)			

Figure 13.3: Viewing the status of scheduled tasks in Task Master.

14.Saving Data in Overwrite File Mode

If data is to be collected over long time periods, eventually the SAA data files will become very large and require an increasing amount of time to process. Once the automated data conversion process has been setup, it is possible to switch the File Output Option to 'Overwrite Existing File' (Figure 14.1) for the SAA data files (the PROJECT_INFO and SERIAL_ERRORS files should NOT be changed). The SAACR_raw2data program will recognize that only the most current data is being collected. This data will be appended to the multi_saa_allcart.mat file so that a complete data history is maintained. The multi_saa_allcart.txt



file (text file of all Cartesian coordinate records) will no longer be updated in this mode, just the multi_saa_curcart.txt (text file of most recent Cartesian coordinate records).

EZSetup Wizard - CR1000_	
Progress	Datalogger Table Output Files
Introduction	For each table listed, select a file name to which data will be written, and check the box if you wish to have the table collected during scheduled collection.
Communication Setup	
Datalogger Settings	Tables Image: Construction Image: Construction Image: Construction
Setup Summary	VPIEZO_INFO PIEZO1_DATA Data File Option
Communication Test	VPIEZ02_DATA Append to File VPIEZ03_DATA No Output File
Datalogger Clock	PROJECT_INFO Append to File Overwrite File UNCampbellscrNLoggerNetvCR1000 BostonUK L
Send Program	√SAA1_DATA √SAA1_DIAGNOSTICS ▼
🗢 Data Files	
Scheduled Collection	Get Table Definitions Note: If table definitions are empty or out of date, you may wish to get them from the datalogger. Click the
Wizard Complete	button to get them.
	Previous Next Finish Cancel Collection <u>H</u> elp

Figure 14.1: Switching the "Data File Output" option in the LoggerNet 'Setup' utility to 'Overwrite Existing File'.

If the SAA Data file is viewed when 'Overwrite File' is implemented, only the most recent data records will be contained in the file.



Appendix I: Integrating other Sensors into the Main Program (Advanced)

** Please contact Measurand if you need assistance in this procedure. The below example is not exactly how the code looks today, but it is similar. **

When adding additional sensor programming to the main SAA program, conflicts can arise with existing programming. The SAA_Include references COM Ports 1-4. If any of these ports are used for connecting other sensors, then all references to these ports must be commented out of the serial port subroutines in the SAA_Include file. Provided is an example program that reads data from piezometers, crackmeters, tiltmeters, a barometer, a temperature and relative humidity sensor, a wind speed sensor, and SAAs.



Note: The SAAF-related code in the following example is not the most recent code. This example is meant to give an idea as to how to add other sensors into the 'Main Program'. Please consult Measurand for assistance when trying to combine measurements from various sensors into a single program.

The SAA data is collected over Scan Interval 1 (15 minutes) and is averaged using Data Tables over 16 x Scan Interval 1 (4 hours). 50 samples are taken per scan, so 800 samples are averaged every 4 hours. The piezometer data is collected at a less frequent interval than the other sensors (every 8 hours) using a 'SlowSequence'. Data from all other sensors is collected every 15 minutes.

'DECLARE USER SPECIFIED CONSTANTS (must be modified by user to match application specifications):

'Enter the number of preliminary samples to capture

Const NUM_PRELIM_SAMPLES = 5

'Enter the number of SAAs to be monitored:

Const NUM_SAAS = 2

'Enter the number of PZ strings to be monitored (has to be at least 1 even if there are no PZ strings !!!)

Const NUM_PIEZO_STRINGS = 0

'Enter the number of total devices (number of SAAs + PZ strings):

Const NUM_DEVICES = 2

'Enter the number of Earth Pressure Cells (EPC) to be monitored:

Const NUM_EPCS = 4



```
'Enter the number of Strain Gages (SG) to be monitored:
Const NUM SGS = 20
'Enter the number of raw data samples to average for each measurement:
' Samples can range from 100-25500; samples are collected at 400 Hz, thus ~64 seconds are required for
NUM_TO_AVG = 25500
Const NUM_TO_AVG = 1000
'>>>>>> END <<<<<<<
'Enter the maximum number of octets (8 segment units) of any single SAA being used:
Const MAX_NUM_OCTETS = 15
'Enter the maximum number of PZs of any single PZ string being used:
Const MAX NUM PIEZOS = 1
'>>>>>> END <<<<<<<<
'Enter number of octets for each SAA (each SAA must have an entry):
Const SAA1_NUM_OCTETS = 15
Const SAA2 NUM OCTETS = 15
'>>>>>> END <<<<<<<
' Enter COM port index for each SAA (each SAA must have an entry) COM1=1, COM2=2, COM3=3, COM4=4,
RS232=5
Const SAA1_COM_PORT = 1
Const SAA2 COM PORT = 2
'>>>>>> END <<<<<<<<
'Enter SAA232-5 port number for each SAA (each SAA must have an entry) 1-5 if SAA232-5 port is being used,
otherwise enter 0
Const SAA1_232_5_PORT = 0
Const SAA2 232 5 PORT = 0
'>>>>>> END <<<<<<<<
```



'>>>>>>> START <<<<<<
'Input number of masked segments:
Const NUM_MASKED_SEGMENTS = 0
'>>>>>>> END <<<<<<<
'>>>>>>> START <<<<<<<
'Input tilt alarm in degrees
Const TILT_ALARM_COUNTS = 20
'>>>>>>> END <<<<<<<
'>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
'Input reference measurement: (1 = initial readings, 2 = previous readings)
Const REF_MEAS = 2
'>>>>>> END <<<<<<<
'>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
'Input SAA orientation (1 = horizontal, 2 = vertical
Const SAA1_ORIENTATION = 1
Const SAA2_ORIENTATION = 1
'>>>>>>>> END <<<<<<<
'>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
'Input whether or not SAATop is present ($0 = not present$, $1 = SAATop$ is present)
Const SAA1_SAATOP = 1
Const SAA2_SAATOP = 1
'>>>>>>> END <<<<<<<
'>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
'Input channel that SAATop is using for communications, 0 to 9, usually 0 in most situations, unless 2 or more arrays happen to be sharing the exact same serial connection
Const SAA1_SAATOP_CHANNEL = 0
Const SAA2_SAATOP_CHANNEL = 0
'>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>



Const ServerAddr="mail.1xbell.ca"

Const ToAddr="youremail@yourcompany.com" Const FromAddr="youremail@yourcompany.com" Const Subject="CR1000 Alarm Notification" Const Attach="" Const UserName="" Const Password="" Const CRLF = CHR(13)+CHR(10) '>>>>>> END <<<<<<< Include "CPU:SAA_Include_V2_65.CR1" 'DECLARE PUBLIC VARIABLES (must be modified by user to match application specifications): 'Earth Pressure Cell Public Variables Public EPC(NUM EPCS,6) Public EPC_Digits(NUM_EPCS) Public EPC_Temp(NUM_EPCS) 'Strain Gages Public Variables Public SG(NUM_SGS,6) Public SG_Digits(NUM_SGS) Public SG_Temp(NUM_SGS) Public Result Public Power_Save_Flag As Boolean Dim x 'add Public ACC and TEMP variables for each SAA: 'SAA1: Public SAA1_ACC_VALUES(MAX_NUM_OCTETS,NUM_SENSORS_PER_OCTET) As Float Public SAA1_TEMP_VALUES(MAX_NUM_OCTETS) As Float

'Input email parameter strings (as constants), Message String & Result Variable



Public SAA1_REF_ACC_VALUES(MAX_NUM_OCTETS,NUM_SENSORS_PER_OCTET) As Float Public SAA1_SAATOP_VOLTAGE As Float Public SAA1_SAATOP_CURRENT As Float Public SAA1_SAATOP_TEMPERATURE As Float

'SAA2:

Public SAA2_ACC_VALUES(MAX_NUM_OCTETS,NUM_SENSORS_PER_OCTET) As Float

Public SAA2_TEMP_VALUES(MAX_NUM_OCTETS) As Float

Public SAA2_REF_ACC_VALUES(MAX_NUM_OCTETS,NUM_SENSORS_PER_OCTET) As Float

Public SAA2_SAATOP_VOLTAGE As Float

Public SAA2_SAATOP_CURRENT As Float

Public SAA2_SAATOP_TEMPERATURE As Float

'preliminary loop count

Public PRELIM_COUNT As Float

'Project Descriptor Info:

DataTable(PROJECT_INFO,FIRST_SCAN,-1) Sample(1,PROGRAM_VERSION_NUM,Float) Sample(1,AVERAGING,Long) Sample(1,PROJECT_NAME,String) Sample(1,NUMBER_SAAS,Long) Sample(NUM_SAAS,NUM_OCTETS_INDEX(),Long) Sample(NUM_SAAS*MAX_NUM_OCTETS,SERIAL_NUMS(),Long) EndTable

'Serial Port Errors:

DataTable(SERIAL_ERRORS,true,-1) Sample(1,SERIAL_NUMS(1,1),Long) Sample(1,SERIAL_NUMS(2,1),Long) Sample(NUM_SAAS,NUM_CRC_ERRORS,Long) Sample(NUM_SAAS,NUM_COM_ERRORS,Long) EndTable



'Logger Diagnostics:

DataTable(LOGGER_DIAGNOSTICS,true,-1) Sample(1,LOGGER_VOLTAGE,Float) Sample(1,LOGGER_TEMPERATURE,Float) EndTable

'SAA Data Table Values (each SAA requires a data table): 'SAA1:

DataTable(SAA1_DATA,True,-1)

Sample(1,SERIAL_NUMS(1,1),Long)

Sample(SAA1_NUM_OCTETS*NUM_SENSORS_PER_OCTET,SAA1_ACC_VALUES(),Float)

Sample(SAA1_NUM_OCTETS,SAA1_TEMP_VALUES(),Float)

EndTable

DataTable(SAA1_DIAGNOSTICS,True,-1)

Sample(1,SERIAL_NUMS(1,1),Long)

Sample(1,SAA1_SAATOP_VOLTAGE,Float)

Sample(1,SAA1_SAATOP_CURRENT,Float)

Sample(1,SAA1_SAATOP_TEMPERATURE,Float)

EndTable

'SAA2:

DataTable(SAA2_DATA,True,-1)

Sample(1,SERIAL_NUMS(2,1),Long)

Sample(SAA2_NUM_OCTETS*NUM_SENSORS_PER_OCTET,SAA2_ACC_VALUES(),Float)

Sample(SAA2_NUM_OCTETS,SAA2_TEMP_VALUES(),Float)

EndTable

DataTable(SAA2_DIAGNOSTICS,True,-1)

Sample(1,SERIAL_NUMS(2,1),Long)

Sample(1,SAA2_SAATOP_VOLTAGE,Float)



Sample(1,SAA2_SAATOP_CURRENT,Float)
Sample(1,SAA2_SAATOP_TEMPERATURE,Float)
EndTable
' SG Data Table
DataTable(SG_DATA, True, -1)
Sample(NUM_SGS*6, SG(), IEEE4)
EndTable
' FPC Data Table
DataTable(EPC_DATA_True1)
$Sample(NUIM_EDCS*6_EDC()) [EEE4]$
Sample(NOIVI_EPCS_0, EPC(), IEEE4)
SUBROUTINES
1**************************************
'*************************************
'*************************************
'*************************************
<pre>'************************************</pre>
'*************************************
'************************************
'*************************************
'************************************
'************************************
'************************************
'************************************
'*************************************
'*************************************
<pre>'************************************</pre>
<pre>'************************************</pre>



la del sella da serada CAA estada da esta da sera está serada da la ser
'
'SAA1:
If HIGH_RES_SAA
Call GetHighResSAAData(SAA1_ACC_VALUES,SAA1_TEMP_VALUES,1)
Else
Call GetSAAData(SAA1_ACC_VALUES,SAA1_TEMP_VALUES,1)
EndIf
CallTable SAA1_DATA
If SAA1_SAATOP>0
Call GetSAATopData(SAA1_SAATOP_VOLTAGE,SAA1_SAATOP_CURRENT,SAA1_SAATOP_TEMPERATURE,1)
CallTable SAA1_DIAGNOSTICS
EndIf
'SAA2:
If HIGH_RES_SAA
Call GetHighResSAAData(SAA2_ACC_VALUES,SAA2_TEMP_VALUES,2)
Else
Call GetSAAData(SAA2_ACC_VALUES,SAA2_TEMP_VALUES,2)



EndIf
II SAAZ_SAATUP>0
Call GetSAATopData(SAA2_SAATOP_VOLTAGE,SAA2_SAATOP_CURRENT,SAA2_SAATOP_TEMPERATURE,2)
CallTable SAA2_DIAGNOSTICS
EndIf
'record serial errors table:
CallTable SERIAL_ERRORS
'turn off outputting of project descriptor info after first run:
FIRST_SCAN = false
NextScan
EndSub
Sub GetVWData()
Scan(1.sec.0.1)'need this dummy scan to auto-allocate data tables
'Get Piezometer Data
SerialOpen (COM3 38400 4 0 0)
SerialOpen (COM4 38400 4.0.0)
$\frac{1}{2} = \frac{1}{2} = \frac{1}$
A $v v z v v (result(), CON(2, 0, 200, 50(1, 1), 1, 1, 1, 0, 450, 5000, 2, _0002, 1, 0)$
AV WZUU (Kesult(),CUMI3,U,ZUU,SG(17,1),Z,1,4,45U,3UUU,Z,_6UHZ,1,U)
CallTable SG_DATA
AVW200 (Result(),COM3,0,200,EPC(1,1),2,5,4,1400,3500,1,_60Hz,1,0)



CallTable EPC_DATA
SerialClose(COM3)
'SerialClose(COM4)
NextScan
EndSub
'DEFINE MAIN PROGRAM:
·
BeginProg
'setup project descriptors:
'
'>>>>>>> START <<<<<<<
' Enter the Project Title
PROJECT_NAME = "TEST"
'>>>>>>> END <<<<<<
NUMBER_SAAS = NUM_SAAS
NUMBER_PIEZO_STRINGS = NUM_PIEZO_STRINGS
AVERAGING = NUM_TO_AVG
PROGRAM_VERSION_NUM = VERSION_NUM
ALARMS_SET = false
HIGH_RES_SAA = true
TILT_ALARM_DEG = TILT_ALARM_COUNTS *1 'FIGURE OUT LATER
'declare the serial numbers (each octet) for each SAA:
'



'SAA1 - S/N: 50115
'nodes 50115 50207 50241 50242 50253 50255 50274 50287 50292 50316 50350 50511 50512 50527 51498
SERIAL_NUMS(1,1) = 50115
SERIAL_NUMS(1,2) = 50207
SERIAL_NUMS(1,3) = 50241
SERIAL_NUMS(1,4) = 50242
SERIAL_NUMS(1,5) = 50253
SERIAL_NUMS(1,6) = 50255
SERIAL_NUMS(1,7) = 50274
SERIAL_NUMS(1,8) = 50287
SERIAL_NUMS(1,9) = 50292
SERIAL_NUMS(1,10) = 50316
SERIAL_NUMS(1,11) = 50350
SERIAL_NUMS(1,12) = 50511
SERIAL_NUMS(1,13) = 50512
SERIAL_NUMS(1,14) = 50527
SERIAL_NUMS(1,15) = 51498
'SAA2 - S/N: 51143
'SAA2 - S/N: 51143 'nodes 51143 51200 51226 51369 51576 51578 51780 51921 51927 51933 51936 51945 51946 51970 51094
'SAA2 - S/N: 51143 'nodes 51143 51200 51226 51369 51576 51578 51780 51921 51927 51933 51936 51945 51946 51970 51094 SERIAL_NUMS(2,1) = 51143
'SAA2 - S/N: 51143 'nodes 51143 51200 51226 51369 51576 51578 51780 51921 51927 51933 51936 51945 51946 51970 51094 SERIAL_NUMS(2,1) = 51143 SERIAL_NUMS(2,2) = 51200
'SAA2 - S/N: 51143 'nodes 51143 51200 51226 51369 51576 51578 51780 51921 51927 51933 51936 51945 51946 51970 51094 SERIAL_NUMS(2,1) = 51143 SERIAL_NUMS(2,2) = 51200 SERIAL_NUMS(2,3) = 51226
'SAA2 - S/N: 51143 'nodes 51143 51200 51226 51369 51576 51578 51780 51921 51927 51933 51936 51945 51946 51970 51094 SERIAL_NUMS(2,1) = 51143 SERIAL_NUMS(2,2) = 51200 SERIAL_NUMS(2,3) = 51226 SERIAL_NUMS(2,4) = 51369
'SAA2 - S/N: 51143 'nodes 51143 51200 51226 51369 51576 51578 51780 51921 51927 51933 51936 51945 51946 51970 51094 SERIAL_NUMS(2,1) = 51143 SERIAL_NUMS(2,2) = 51200 SERIAL_NUMS(2,3) = 51226 SERIAL_NUMS(2,4) = 51369 SERIAL_NUMS(2,5) = 51576
'SAA2 - S/N: 51143 'nodes 51143 51200 51226 51369 51576 51578 51780 51921 51927 51933 51936 51945 51946 51970 51094 SERIAL_NUMS(2,1) = 51143 SERIAL_NUMS(2,2) = 51200 SERIAL_NUMS(2,3) = 51226 SERIAL_NUMS(2,4) = 51369 SERIAL_NUMS(2,5) = 51576 SERIAL_NUMS(2,6) = 51578
'SAA2 - S/N: 51143 'nodes 51143 51200 51226 51369 51576 51578 51780 51921 51927 51933 51936 51945 51946 51970 51094 SERIAL_NUMS(2,1) = 51143 SERIAL_NUMS(2,2) = 51200 SERIAL_NUMS(2,3) = 51226 SERIAL_NUMS(2,4) = 51369 SERIAL_NUMS(2,5) = 51576 SERIAL_NUMS(2,6) = 51578 SERIAL_NUMS(2,6) = 51578
'SAA2 - S/N: 51143 'nodes 51143 51200 51226 51369 51576 51578 51780 51921 51933 51936 51945 51946 51970 51094 SERIAL_NUMS(2,1) = 51143 SERIAL_NUMS(2,2) = 51200 SERIAL_NUMS(2,3) = 51226 SERIAL_NUMS(2,4) = 51369 SERIAL_NUMS(2,5) = 51576 SERIAL_NUMS(2,6) = 51578 SERIAL_NUMS(2,7) = 51780 SERIAL_NUMS(2,8) = 51921
'SAA2 - S/N: 51143 'nodes 51143 51200 51226 51369 51576 51578 51780 51921 51927 51933 51936 51945 51946 51970 51094 SERIAL_NUMS(2,1) = 51143 SERIAL_NUMS(2,2) = 51200 SERIAL_NUMS(2,2) = 51226 SERIAL_NUMS(2,3) = 51226 SERIAL_NUMS(2,4) = 51369 SERIAL_NUMS(2,5) = 51576 SERIAL_NUMS(2,5) = 51578 SERIAL_NUMS(2,7) = 51780 SERIAL_NUMS(2,8) = 51921
'SAA2 - S/N: 51143 'nodes 51143 51200 51226 51369 51576 51578 51780 51921 51927 51933 51936 51945 51946 51970 51094 SERIAL_NUMS(2,1) = 51143 SERIAL_NUMS(2,2) = 51200 SERIAL_NUMS(2,3) = 51226 SERIAL_NUMS(2,4) = 51369 SERIAL_NUMS(2,5) = 51576 SERIAL_NUMS(2,5) = 51576 SERIAL_NUMS(2,6) = 51578 SERIAL_NUMS(2,7) = 51780 SERIAL_NUMS(2,8) = 51921 SERIAL_NUMS(2,9) = 51927
'SAA2 - S/N: 51143 'nodes 51143 51200 51226 51369 51576 51578 51780 51921 51927 51933 51936 51945 51946 51970 51094 SERIAL_NUMS(2,1) = 51143 SERIAL_NUMS(2,2) = 51200 SERIAL_NUMS(2,3) = 51226 SERIAL_NUMS(2,3) = 51226 SERIAL_NUMS(2,4) = 51369 SERIAL_NUMS(2,5) = 51576 SERIAL_NUMS(2,5) = 51578 SERIAL_NUMS(2,6) = 51578 SERIAL_NUMS(2,7) = 51780 SERIAL_NUMS(2,7) = 51921 SERIAL_NUMS(2,9) = 51927 SERIAL_NUMS(2,10) = 51933 SERIAL_NUMS(2,11) = 51936
'SAA2 - S/N: 51143'nodes 51143 51200 51226 51369 51576 51578 51780 51921 51927 51933 51936 51945 51946 51970 51094SERIAL_NUMS(2,1) = 51143SERIAL_NUMS(2,2) = 51200SERIAL_NUMS(2,3) = 51226SERIAL_NUMS(2,3) = 51226SERIAL_NUMS(2,4) = 51369SERIAL_NUMS(2,5) = 51576SERIAL_NUMS(2,6) = 51578SERIAL_NUMS(2,6) = 51578SERIAL_NUMS(2,7) = 51780SERIAL_NUMS(2,8) = 51921SERIAL_NUMS(2,9) = 51927SERIAL_NUMS(2,10) = 51933SERIAL_NUMS(2,11) = 51936SERIAL_NUMS(2,12) = 51945
'SAA2 - S/N: 51143'nodes 51143 51200 51226 51369 51576 51578 51780 51921 51927 51933 51936 51945 51946 51970 51094SERIAL_NUMS(2,1) = 51143SERIAL_NUMS(2,2) = 51200SERIAL_NUMS(2,3) = 51226SERIAL_NUMS(2,4) = 51369SERIAL_NUMS(2,5) = 51576SERIAL_NUMS(2,6) = 51578SERIAL_NUMS(2,6) = 51578SERIAL_NUMS(2,7) = 51780SERIAL_NUMS(2,7) = 51921SERIAL_NUMS(2,9) = 51927SERIAL_NUMS(2,10) = 51933SERIAL_NUMS(2,11) = 51936SERIAL_NUMS(2,12) = 51945SERIAL_NUMS(2,13) = 51946
'SAA2 - S/N: 51143 'nodes 51143 51200 51226 51369 51576 51578 51780 51927 51933 51936 51945 51946 51970 51094 SERIAL_NUMS(2,1) = 51143 SERIAL_NUMS(2,2) = 51200 SERIAL_NUMS(2,3) = 51226 SERIAL_NUMS(2,3) = 51226 SERIAL_NUMS(2,4) = 51369 SERIAL_NUMS(2,5) = 51576 SERIAL_NUMS(2,5) = 51576 SERIAL_NUMS(2,6) = 51578 SERIAL_NUMS(2,7) = 51780 SERIAL_NUMS(2,8) = 51921 SERIAL_NUMS(2,9) = 51927 SERIAL_NUMS(2,10) = 51933 SERIAL_NUMS(2,11) = 51936 SERIAL_NUMS(2,12) = 51945 SERIAL_NUMS(2,12) = 51945 SERIAL_NUMS(2,12) = 51945



'declare the number of octets for each SAA:

'_____

'SAA 1:

NUM_OCTETS_INDEX(1) = SAA1_NUM_OCTETS

'SAA2:

NUM_OCTETS_INDEX(2) = SAA2_NUM_OCTETS

'declare the com port assignment for each SAA:

·_____

'SAA1: COM_PORT_INDEX(1) = SAA1_COM_PORT SAA232_5_PORT_INDEX(1) = SAA1_232_5_PORT

'SAA2; COM_PORT_INDEX(2) = SAA2_COM_PORT SAA232_5_PORT_INDEX(2) = SAA2_232_5_PORT

'declare the orientation of each SAA:

۱_____

'SAA1:

ORIENTATION_INDEX(1) = SAA1_ORIENTATION

'SAA2:

ORIENTATION_INDEX(2) = SAA2_ORIENTATION

'initialize COM, CRC and RANGE errors:

```
NUM_COM_ERRORS = 0
```



```
NUM_CRC_ERRORS = 0
NUM_RANGE_ERRORS = 0
'initialize SAATOP parameters
'SAA1:
SAA1_SAATOP_VOLTAGE = 0.0
SAA1_SAATOP_CURRENT = 0.0
SAA1_SAATOP_TEMPERATURE = 0.0
'SAA2:
SAA2_SAATOP_VOLTAGE = 0.0
SAA2_SAATOP_CURRENT = 0.0
SAA2_SAATOP_TEMPERATURE = 0.0
'declare the SAATop channel of each SAA:
!_____
'SAA1:
SAATOP_CHANNEL_INDEX(1) = SAA1_SAATOP_CHANNEL
'SAA2:
SAATOP_CHANNEL_INDEX(2) = SAA2_SAATOP_CHANNEL
'initialize output descriptor Boolean (only need 1 record of project descriptor):
FIRST_SCAN = True
'initialize power saving flag
Power_Save_Flag = True
SW12(1)'make sure modem starts off being on
'do preliminary scan first
```

For PRELIM_COUNT = 1 To NUM_PRELIM_SAMPLES



```
Call GetVWData()
 Call GetArrayData()
 Delay(0,10,2)
Next PRELIM_COUNT
'now do actual scan indefinitely
Scan(30, Min, 0, 0)
 'power switching
 If Power_Save_Flag = True Then
   If TimeIntoInterval(510,1440,Min) Then
    SW12(1)
   EndIf
  If TimeIntoInterval(1170,1440,Min) Then
    SW12(0)
   EndIf
 Else
  SW12(1)
 EndIf
 ' get VW sensor data
 Call GetVWData()
 ' get SAA data
```



Call GetArrayData()

NextScan

'>>>>>>> END <<<<<<<<

EndProg