

*User Guide*

# Integration of ShapeAccelArray and Campbell Scientific CR800/CR1000 Data Loggers

---



## 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

Table of Contents .....	iii
List of Figures .....	iv
Preface .....	v
1. Introduction .....	1
2. Hardware Setup .....	1
3. Checking SAA Performance Prior to Installation.....	4
4. Using LoggerNet.....	7
5. Creating a Main Program File .....	7
6. Compiling SAA Program Files .....	10
7. Uploading Program Files to the CR1000 .....	10
8. Data File Management.....	13
9. Manual Data Collection.....	14
10. Automated Data Collection.....	16
11. Manual Conversions of SAA Data .....	17
12. Viewing SAA Data .....	18
13. Automated Data Conversions .....	19
14. Saving Data in Overwrite File Mode .....	22
Appendix I: Integrating other Sensors into the Main Program (Advanced) .....	24

## 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 panel. ....	1
Figure 2.2: Wiring diagram for CR1000 with SAA232.....	2
Figure 2.3: Wiring diagram for CR1000 with SAA232-5. ....	3
Figure 2.4: Example CR800 & SAA Integration using one SAAREG, two SAA232, a CR800 logger, and a Raven XT Wireless Modem. ....	4
Figure 3.1: SAA232-USB connector cable. ....	5
Figure 3.2: Opening the Diagnostics Tests window in SAAREcorder. ....	6
Figure 3.3: Graph for Total Acceleration Check in SAAREcorder. ....	6
Figure 4.1: LoggerNet Utilities menu.....	7
Figure 5.1: SAACR_FileGen utility.....	8
Figure 5.2: SAA232 or SAA232-5 Channel selection. ....	9
Figure 5.3: Saving a logger program created using the SAACR_FileGen utility. ....	9
Figure 6.1: Saving and Compiling the SAA logger program. ....	10
Figure 7.1 Selecting File Control in the LoggerNet Connect utility. ....	11
Figure 7.2: Selecting 'Send' option in 'File Control'. ....	12
Figure 7.3: Selecting 'Run Options' when sending files to CR1000 logger. ....	12
Figure 7.4: Sending a program to the CR1000. ....	13
Figure 8.1: Specifying the destination of data files using the LoggerNet Setup utility ....	14
Figure 9.1: Disabling scheduled data collection.....	15
Figure 9.2: Collecting data using the LoggerNet 'Connect' utility. ....	16
Figure 10.1: Enabling scheduled data collection.....	17
Figure 12.1: Selecting an SAA to View in SAAView. ....	18
Figure 12.2: Quickview of absolute shape plots for SAA in SAAView. ....	19
Figure 12.3: Viewing absolute shape for SAA in Main View of SAAView ....	19
Figure 13.1: Creating a task using the LoggerNet 'Task Master' Utility. ....	20
Figure 13.2: Configuring task options in Task Master. ....	21
Figure 13.3: Viewing the status of scheduled tasks in Task Master. ....	22
Figure 14.1: Switching the "Data File Output" option in the LoggerNet 'Setup' utility to 'Overwrite Existing File'. ....	23

## Preface

In many cases it is not convenient or desirable to collect data directly from an SAA using SAASRecorder. 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.

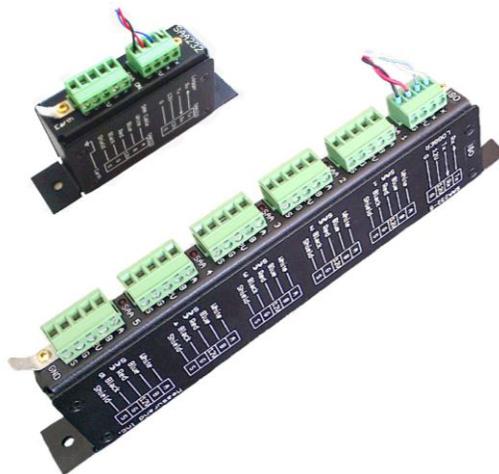
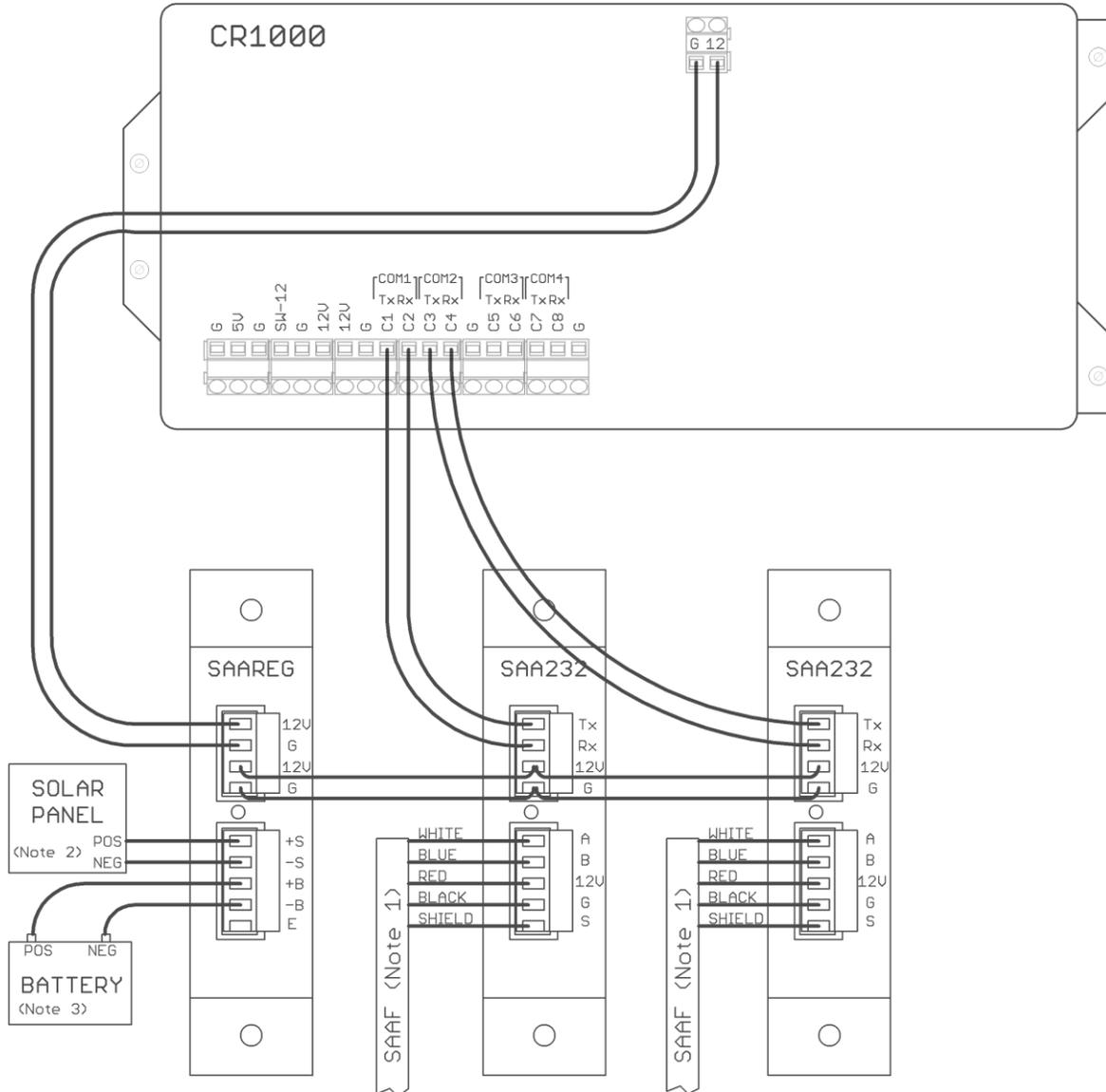


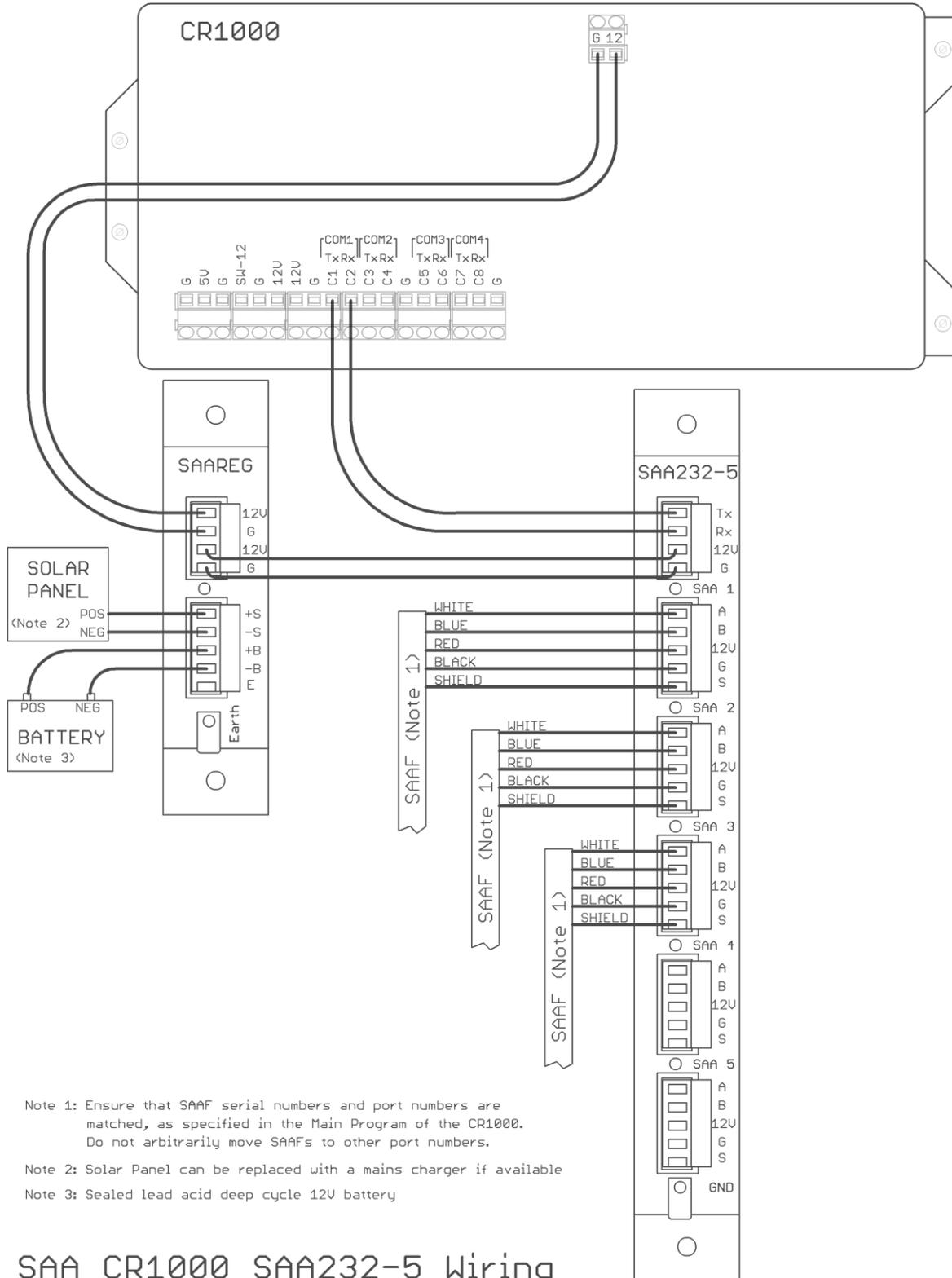
Figure 2.1: SAA232 and SAA232-5 for connecting SAA to COM ports on CR800/CR1000 loggers. Note SAA wiring diagram on the side panel.



- 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 12U 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.



## SAA CR1000 SAA232-5 Wiring

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

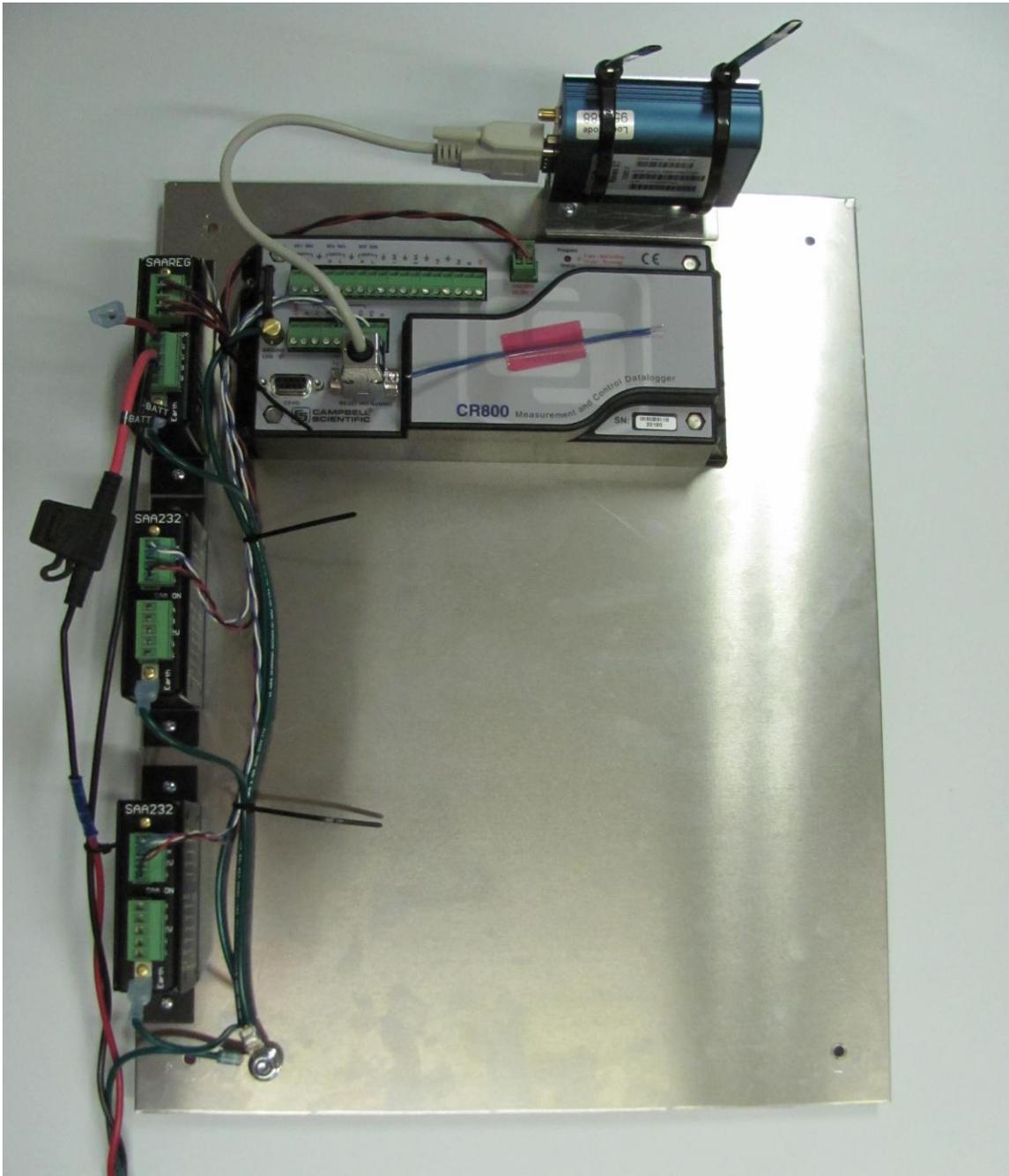


Figure 2.4: Example CR800 & SAA Integration using one SAAREG, two SAA232, a CR800 logger, and a Raven XT Wireless Modem.

### 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 <http://www.MeasurandGeotechnical.com/software.html>. 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 ([http://MeasurandGeotechnical.com/Installation\\_Guide\\_2011.pdf](http://MeasurandGeotechnical.com/Installation_Guide_2011.pdf)). If you have any issues with the calibration, please contact Measurand.

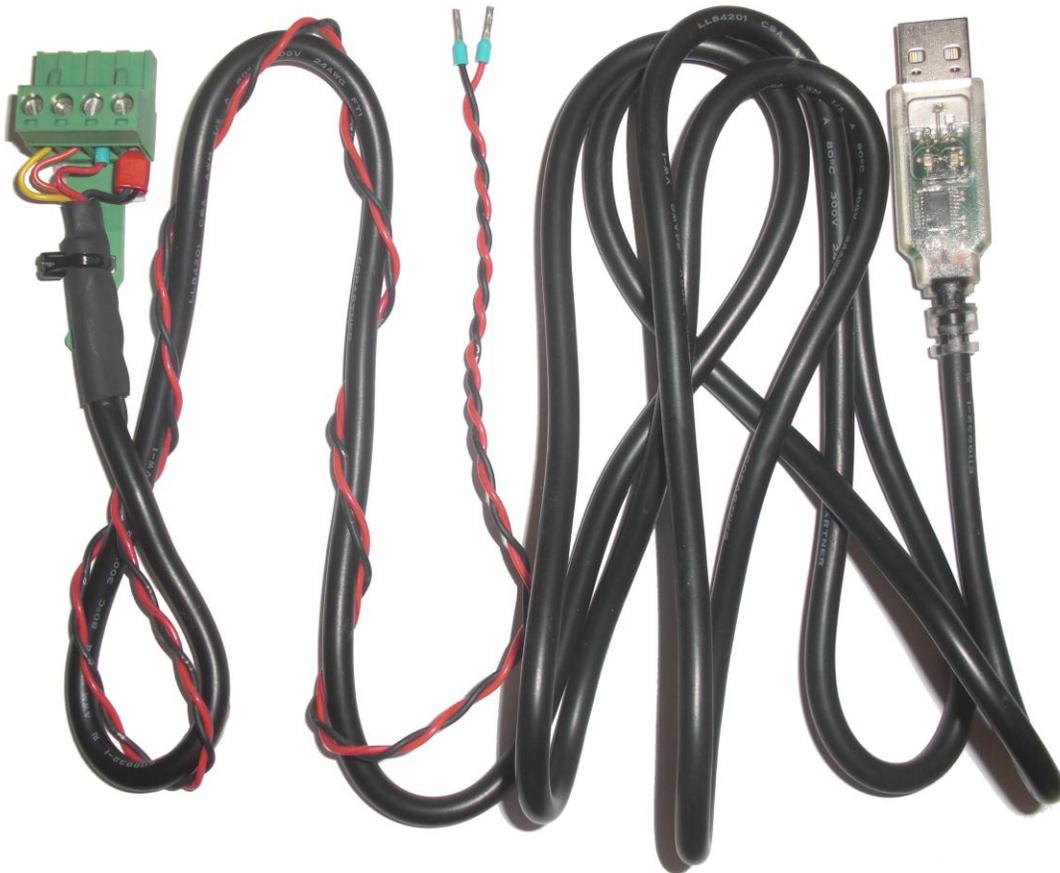


Figure 3.1: SAA232-USB connector cable.

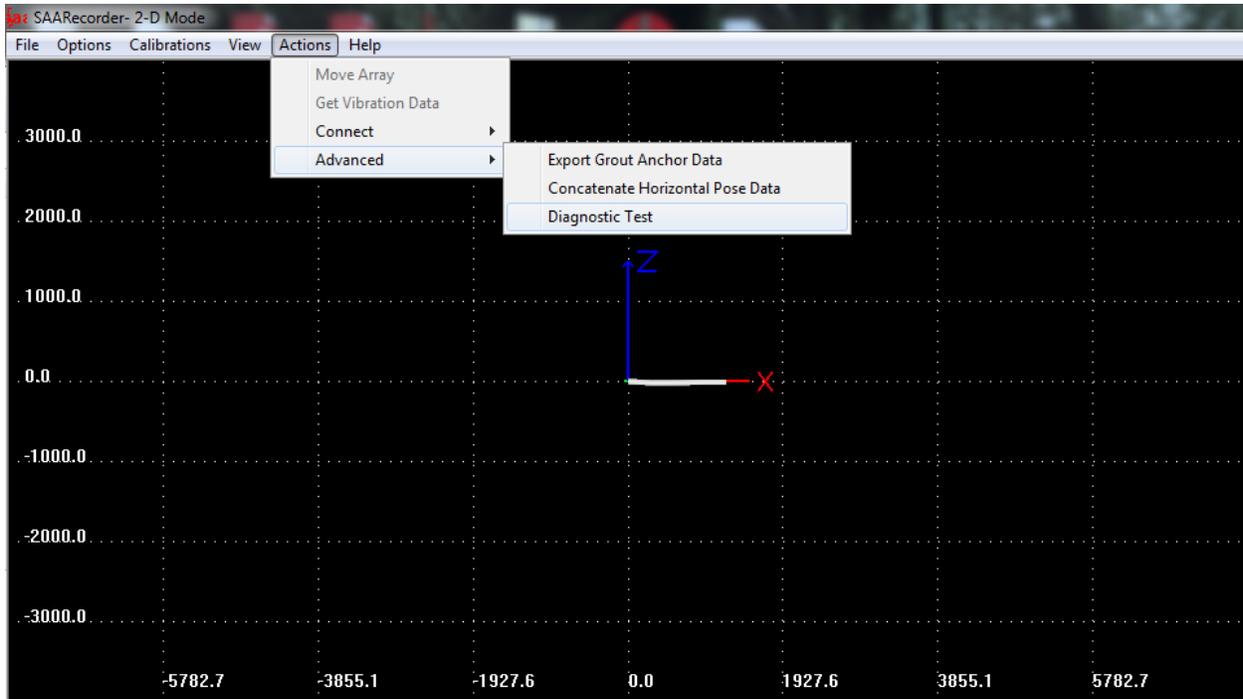


Figure 3.2: Opening the Diagnostics Tests window in SAAREcorder.

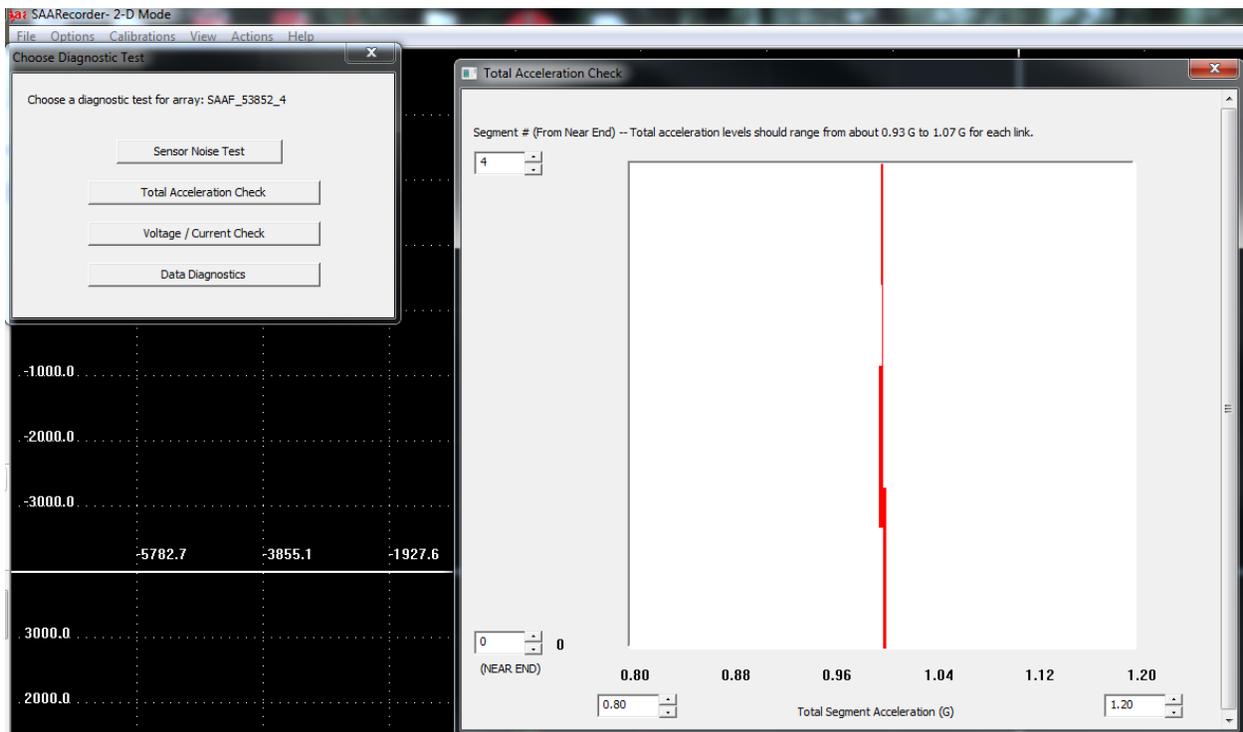


Figure 3.3: Graph for Total Acceleration Check in SAAREcorder.

## 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 Scientific's LoggerNet software, visit the Campbell Scientific website to obtain the user manual (<http://www.campbellsci.ca/>). 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;

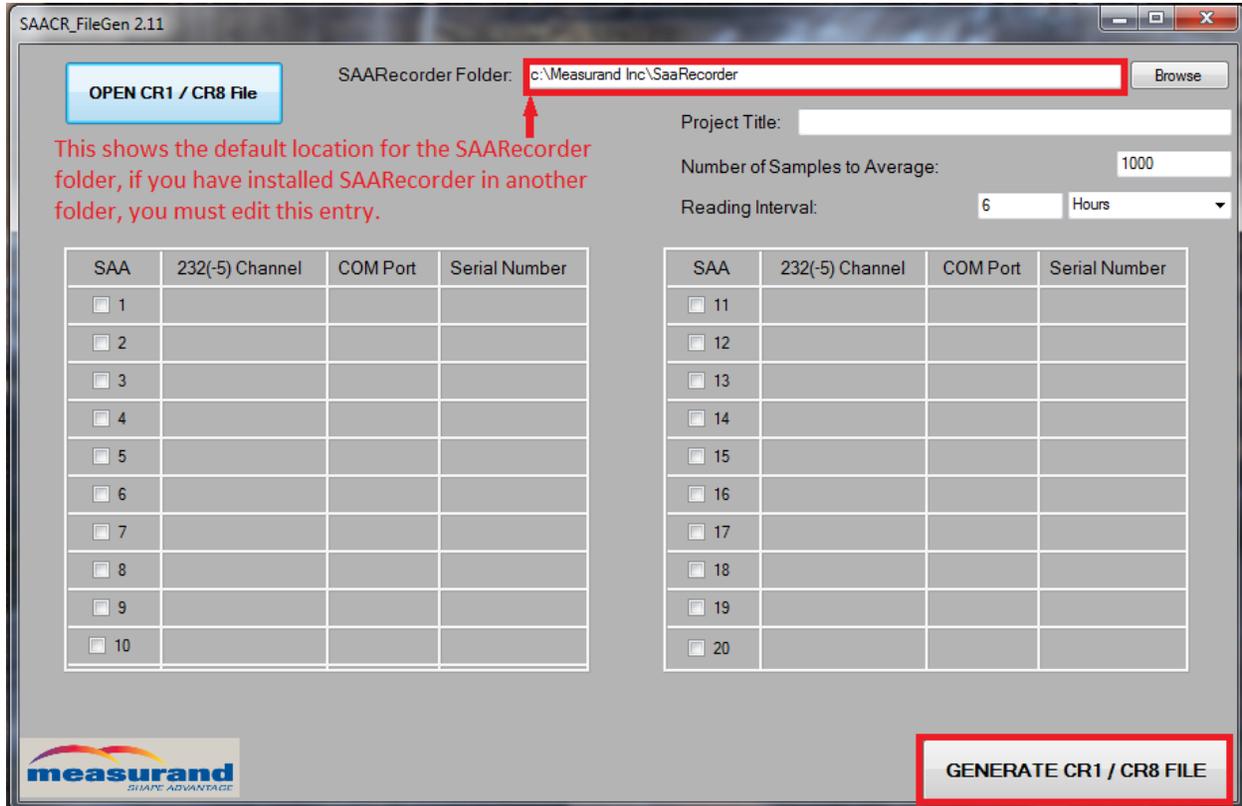


Figure 5.1: SAACR\_FileGen utility.

- 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.

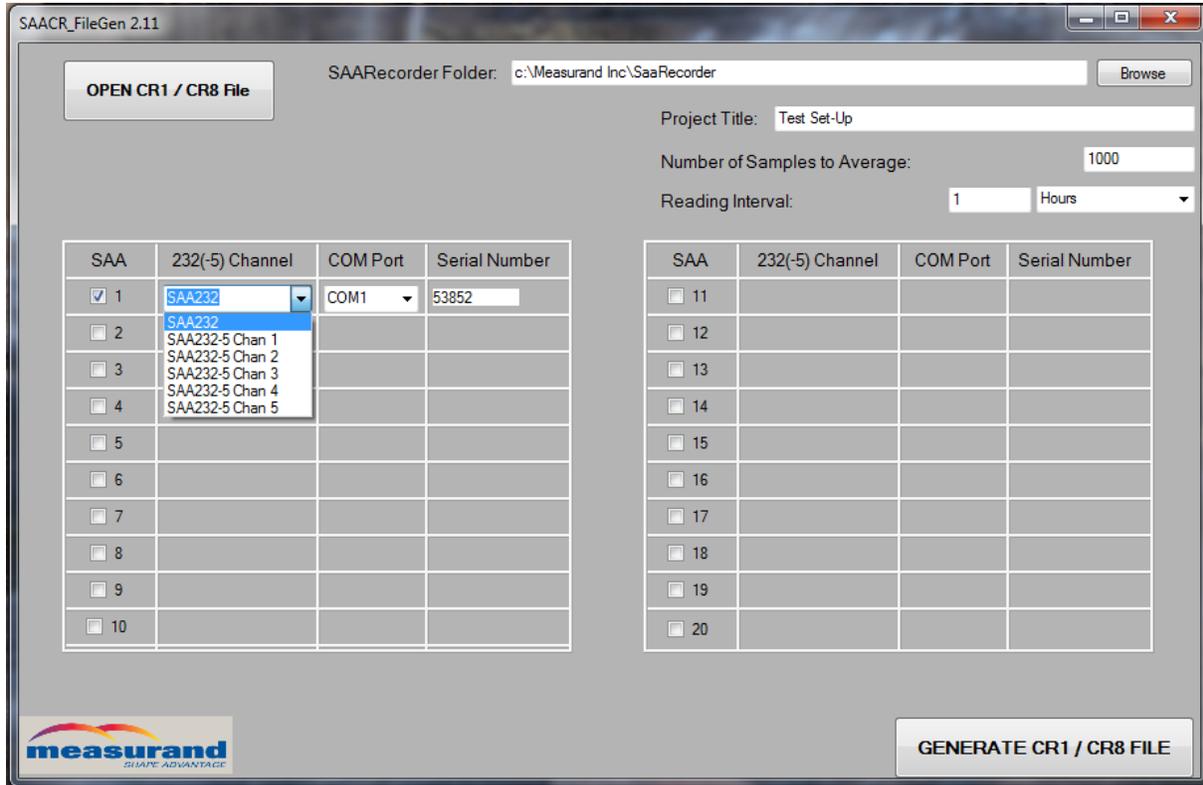


Figure 5.2: SAA232 or SAA232-5 Channel selection.

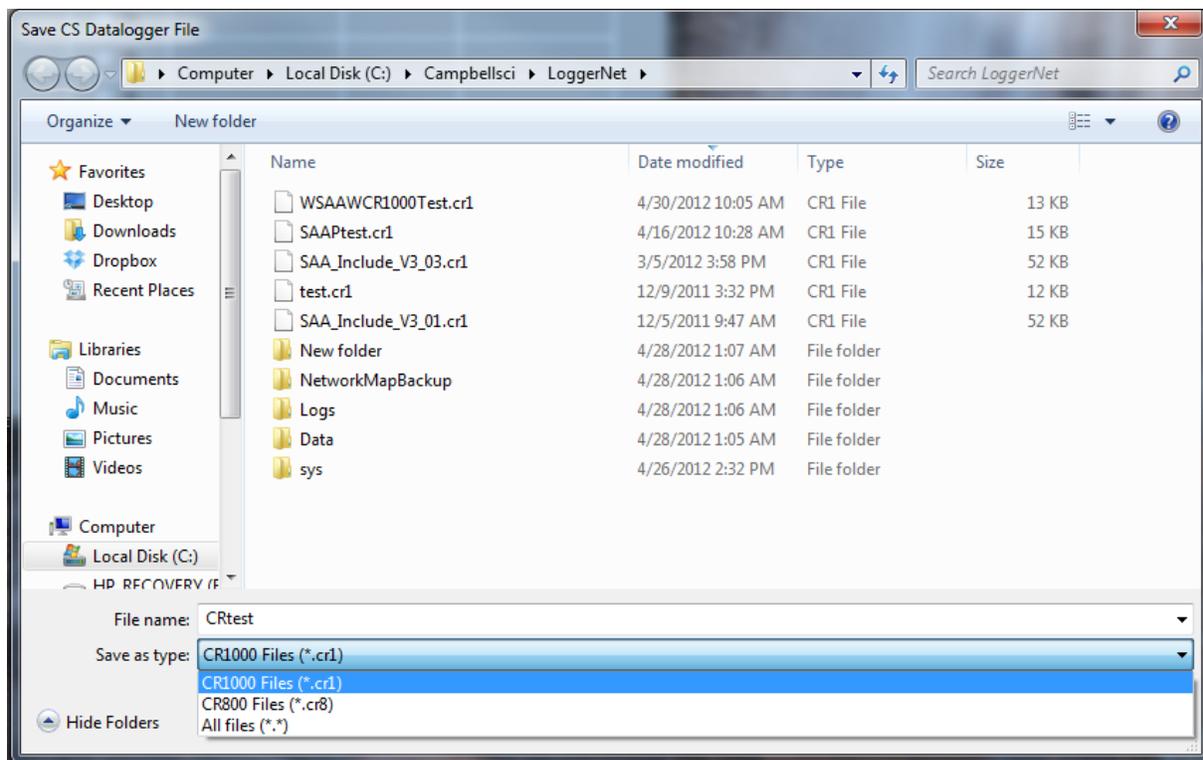


Figure 5.3: Saving a logger program created using the SAACR\_FileGen utility.

## 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.*

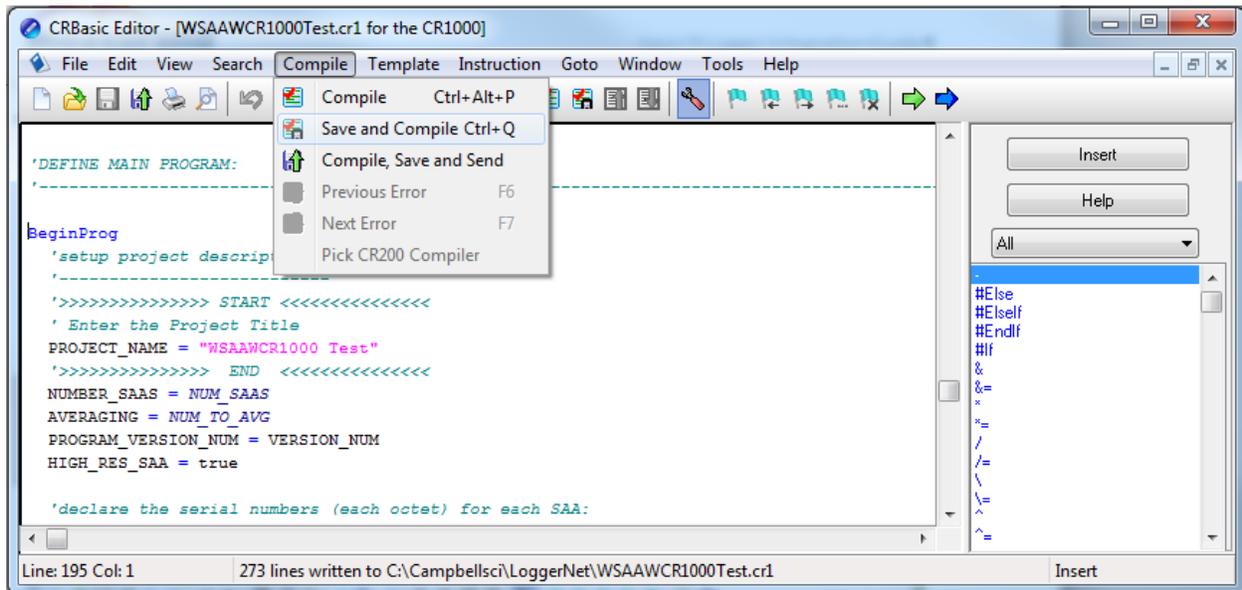


Figure 6.1: Saving and Compiling the SAA logger program.



*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).

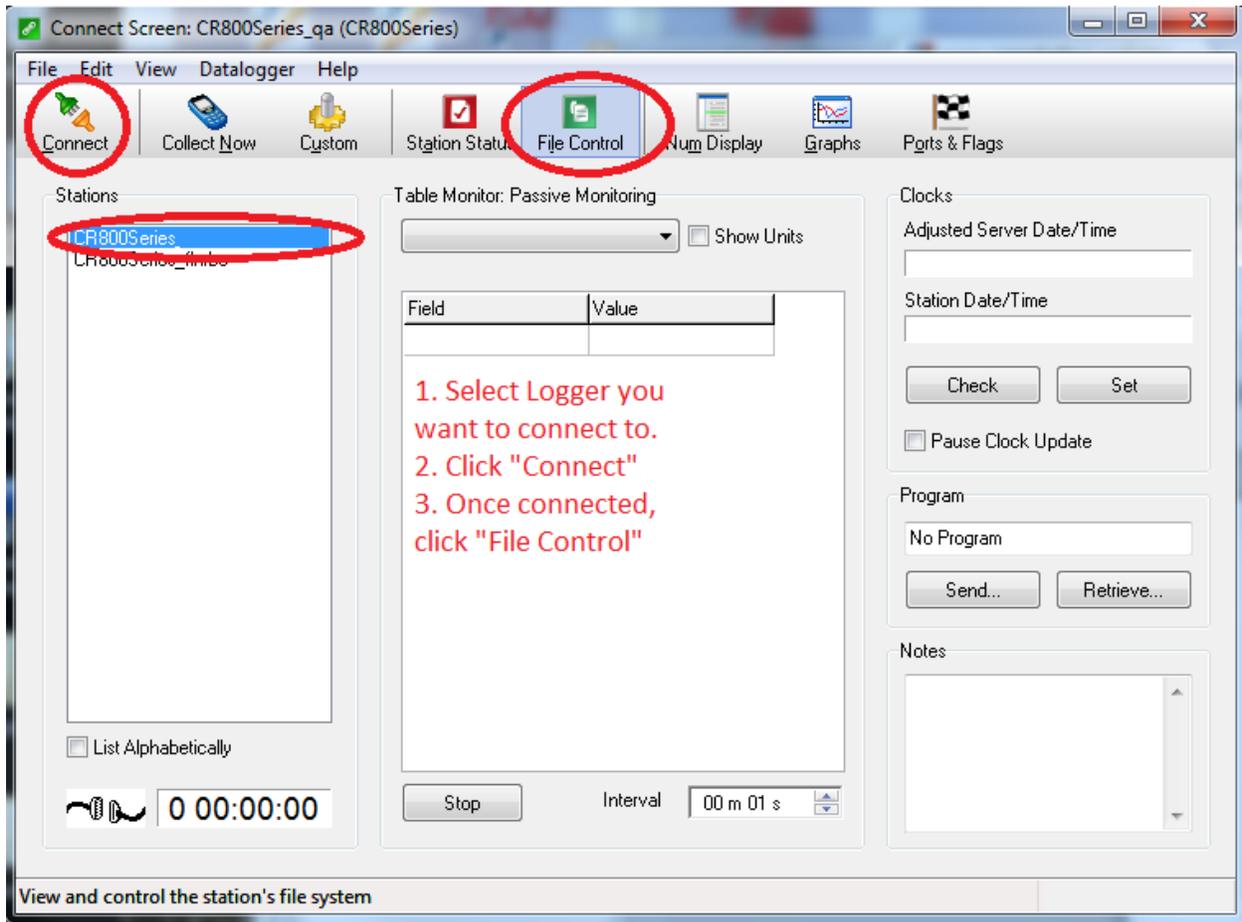


Figure 7.1 Selecting File Control in the LoggerNet Connect utility.

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’.*

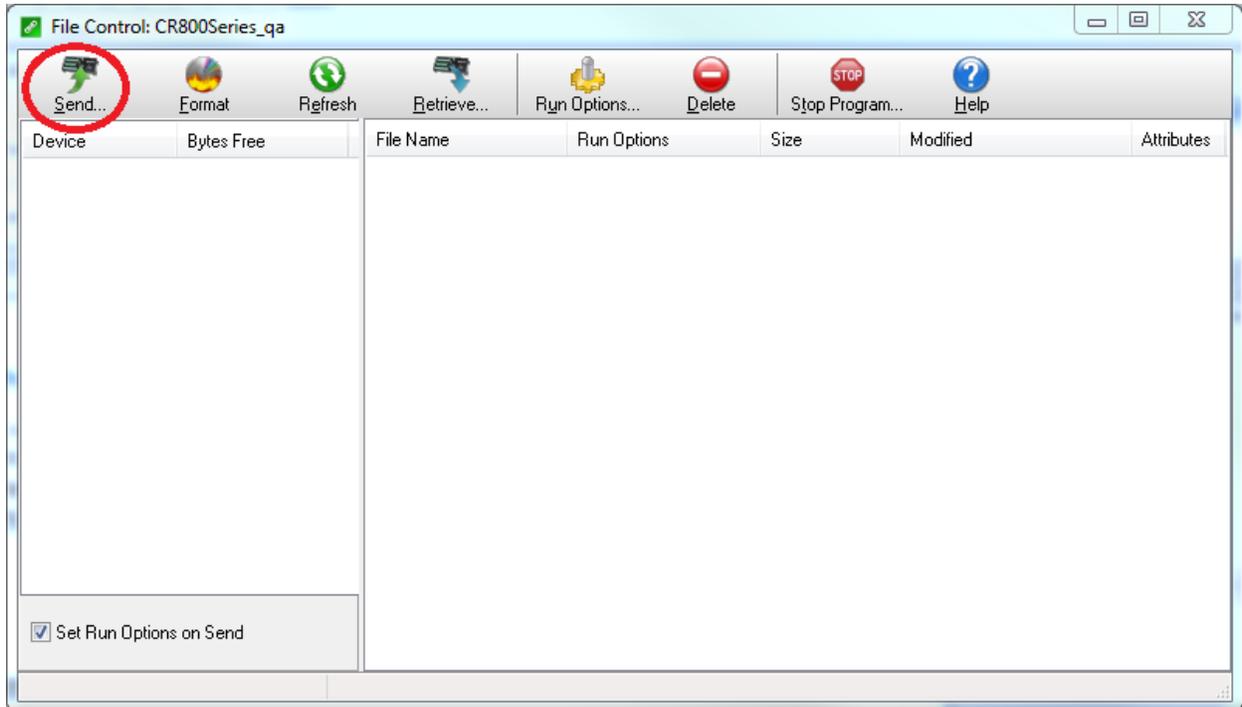


Figure 7.2: Selecting 'Send' option in 'File Control'.

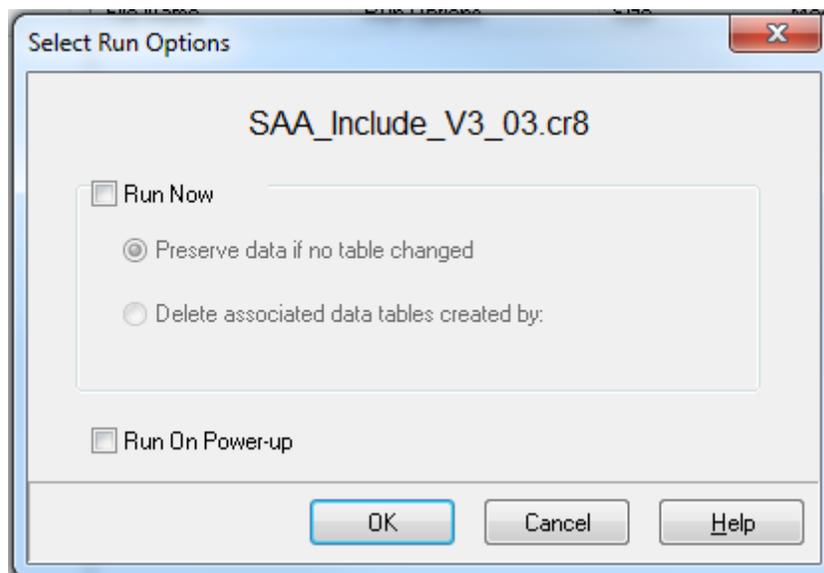


Figure 7.3: Selecting 'Run Options' when sending files to CR1000 logger.

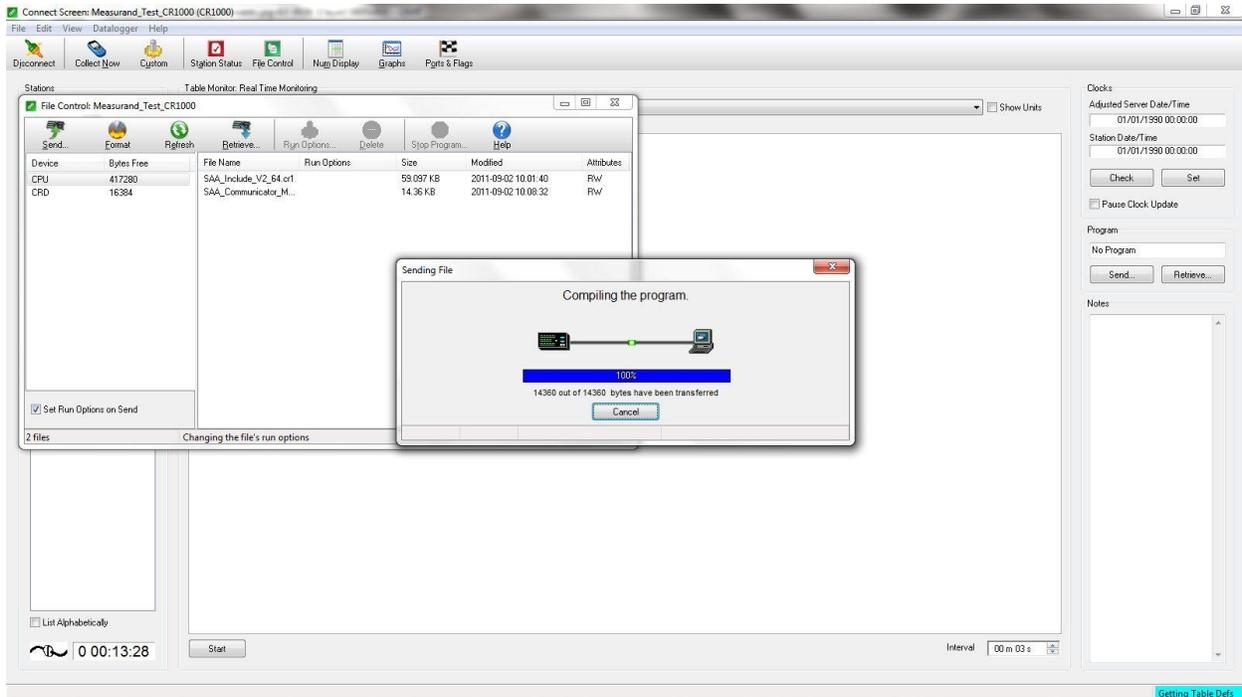


Figure 7.4: Sending a program to the CR1000.

## 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).

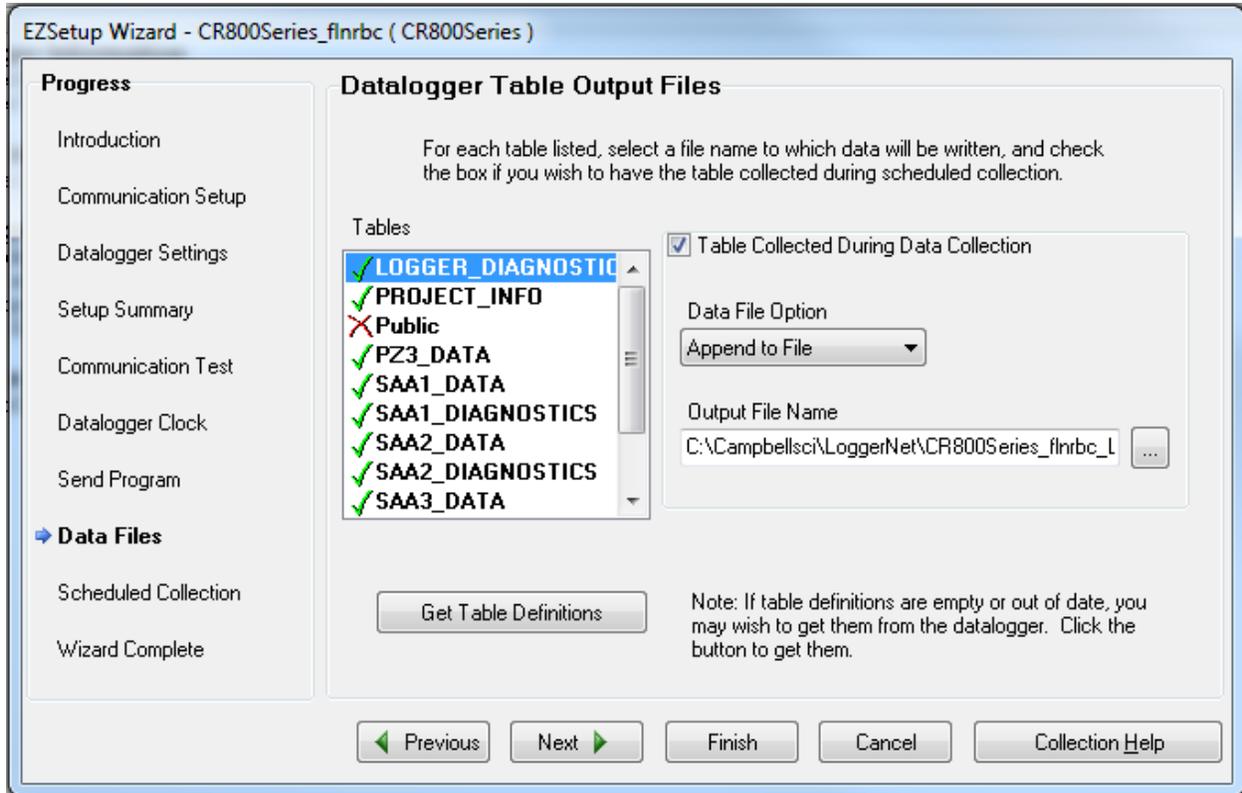


Figure 8.1: Specifying the destination of data files using the LoggerNet Setup utility



*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.*

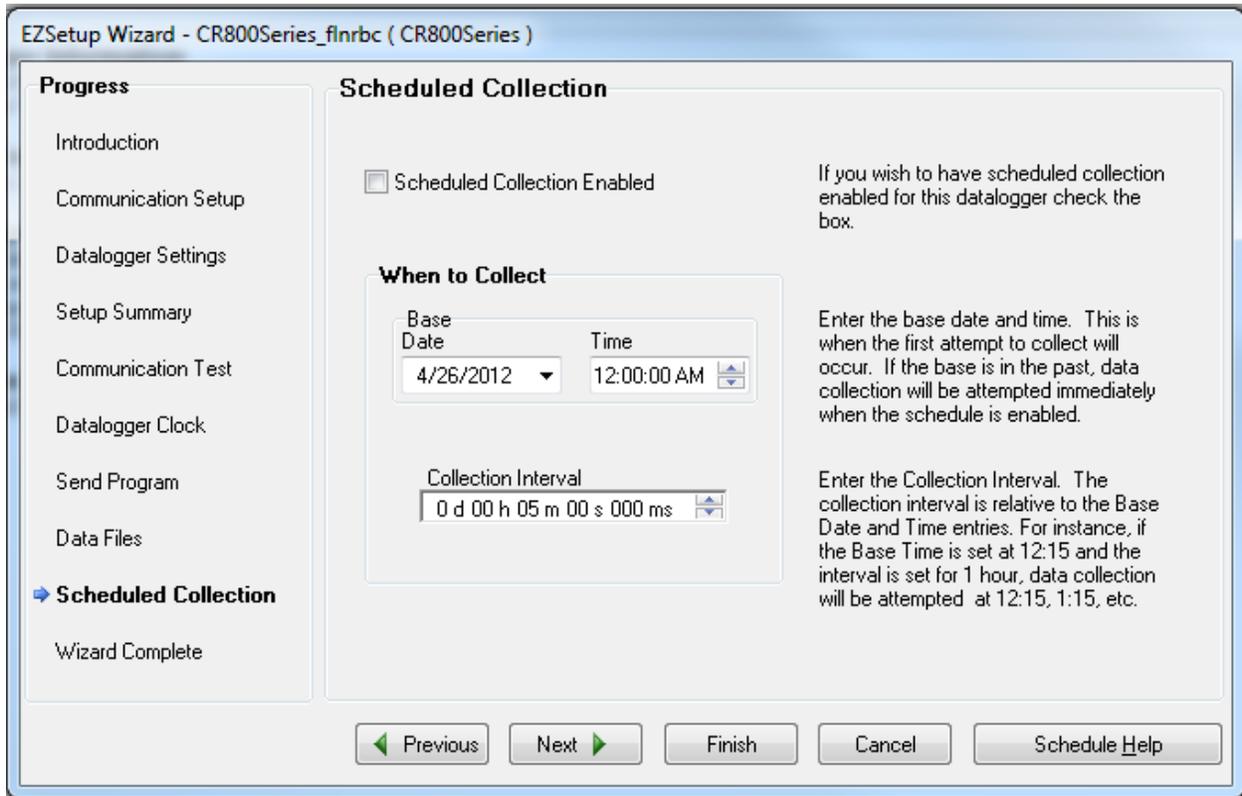


Figure 9.1: Disabling scheduled data collection.

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.*

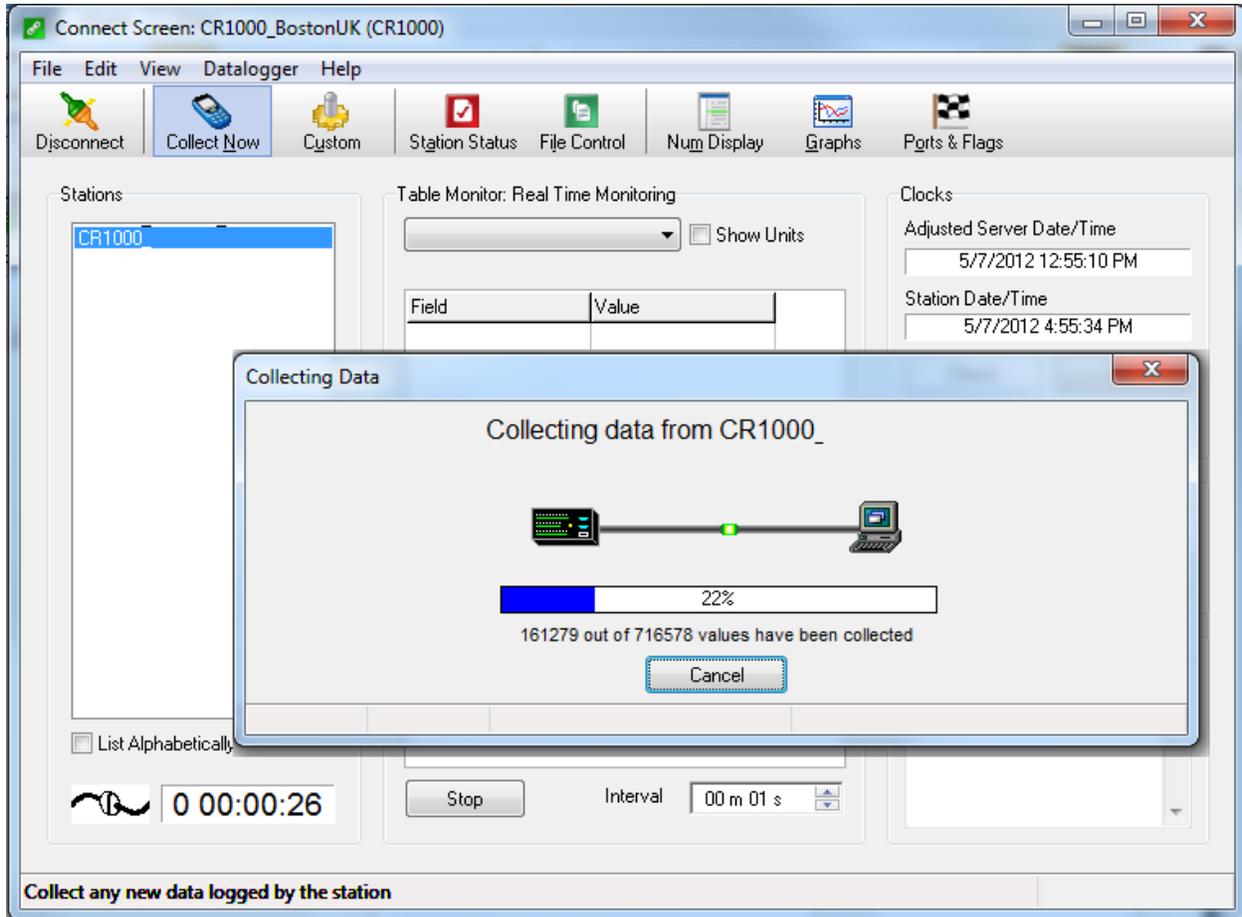


Figure 9.2: Collecting data using the LoggerNet ‘Connect’ utility.

## 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.



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](http://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\non\_saa\_coeffs). For more information on these aspects and how to use the converter, please refer to the SAACR\_raw2data Manual.

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.

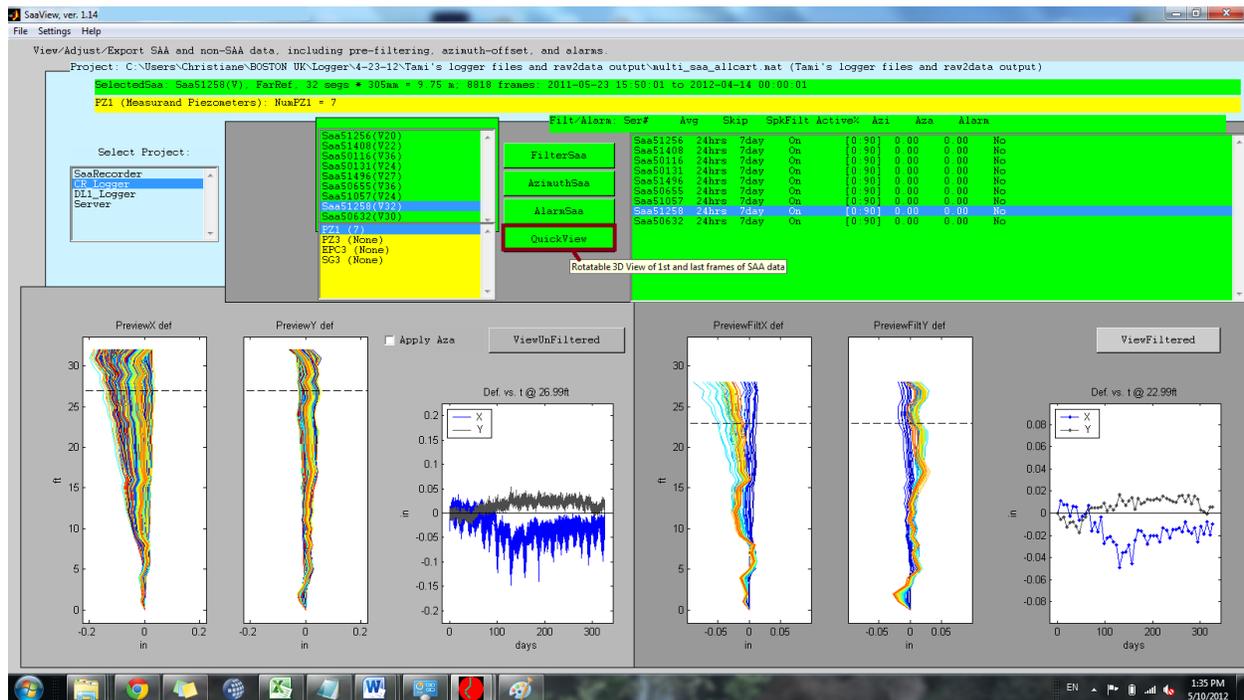


Figure 12.1: Selecting an SAA to View in SAAView.

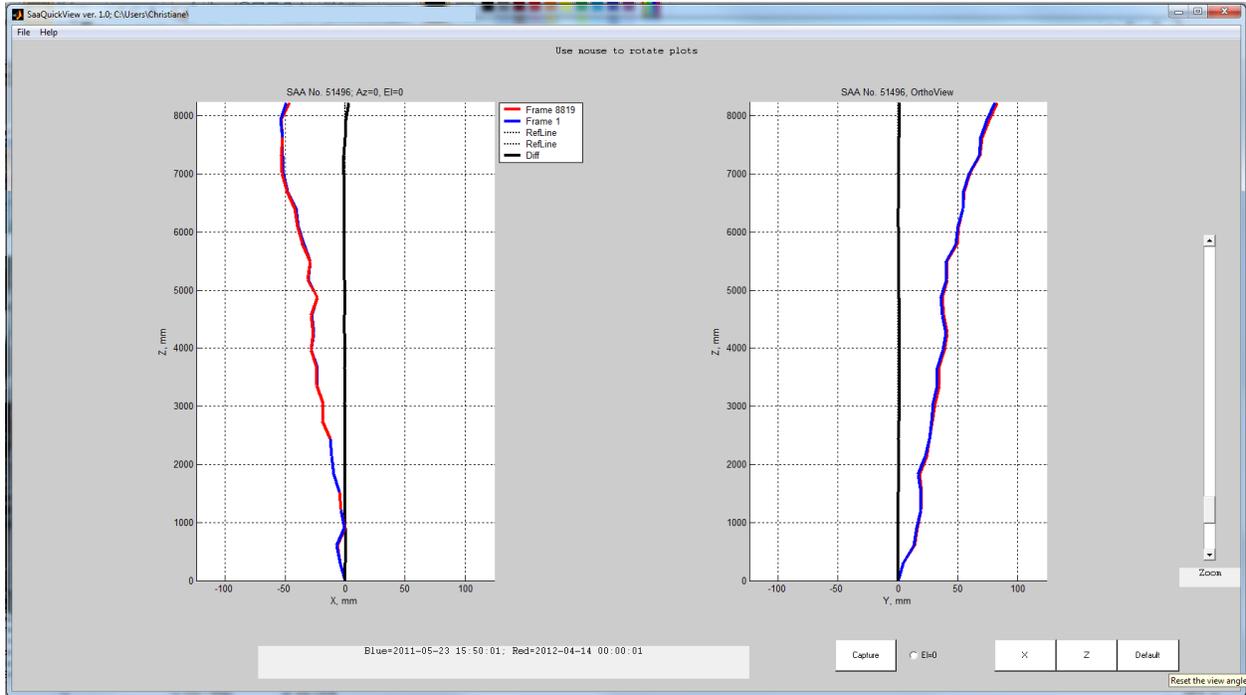


Figure 12.2: Quickview of absolute shape plots for SAA in SAAView.

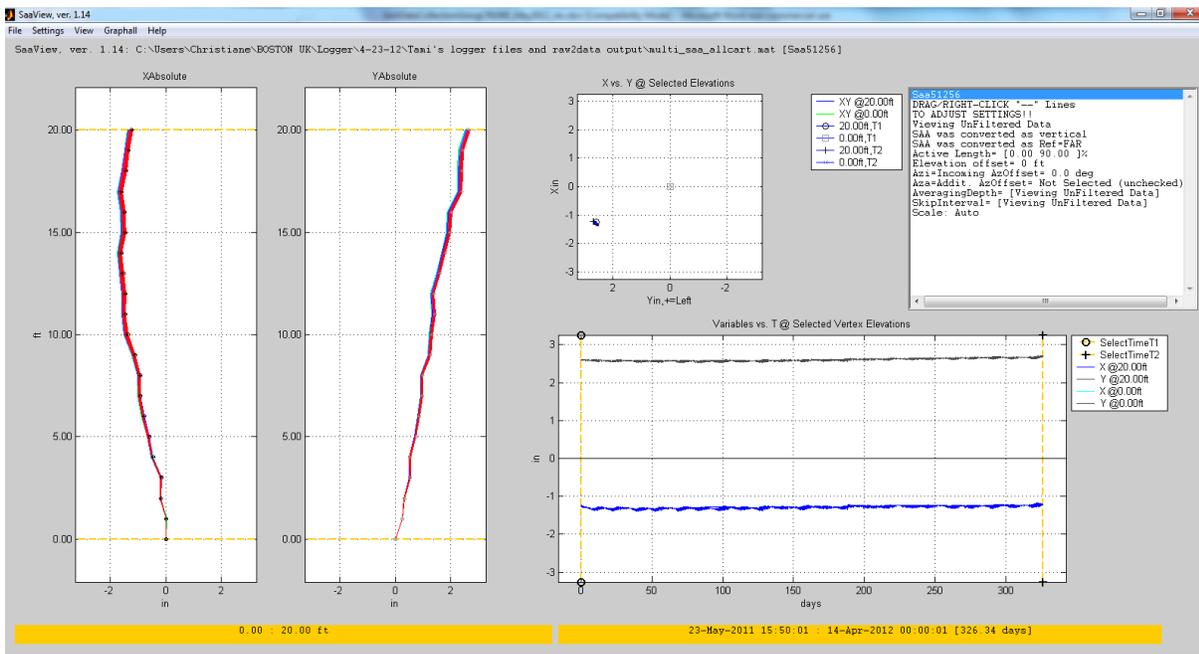


Figure 12.3: Viewing absolute shape for SAA in Main View of SAAView

### 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).

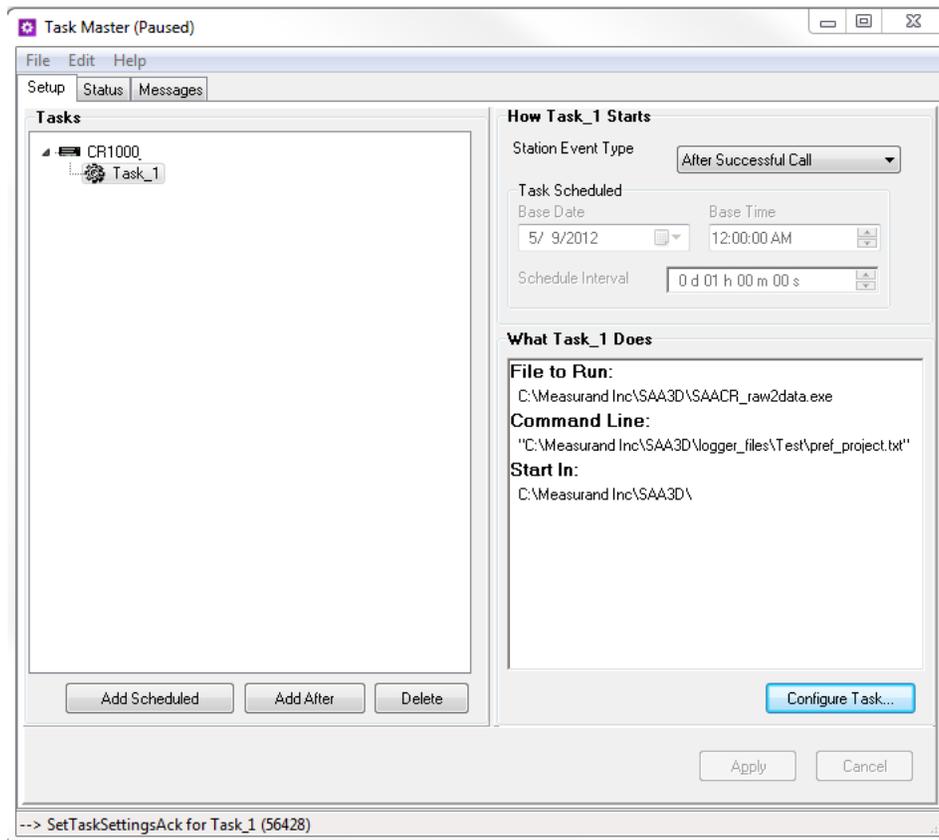


Figure 13.1: Creating a task using the LoggerNet 'Task Master' Utility.

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.

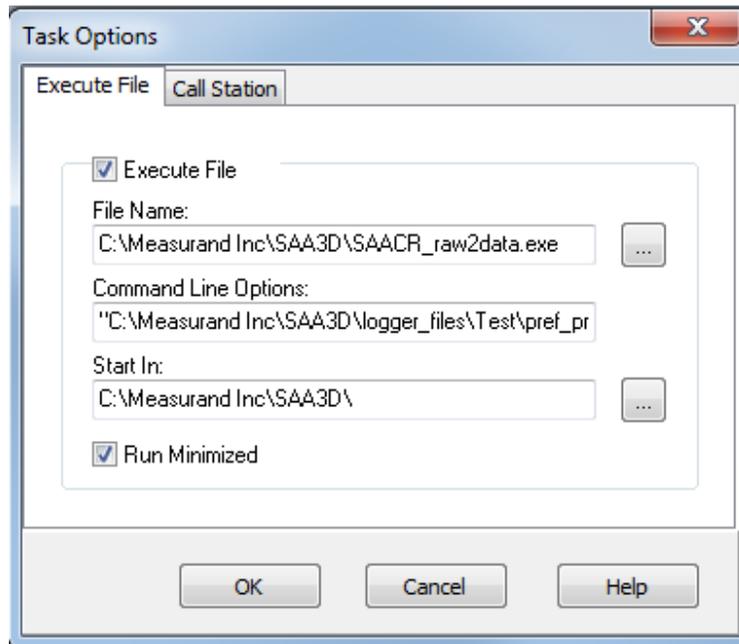


Figure 13.2: Configuring task options in Task Master.



*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.*

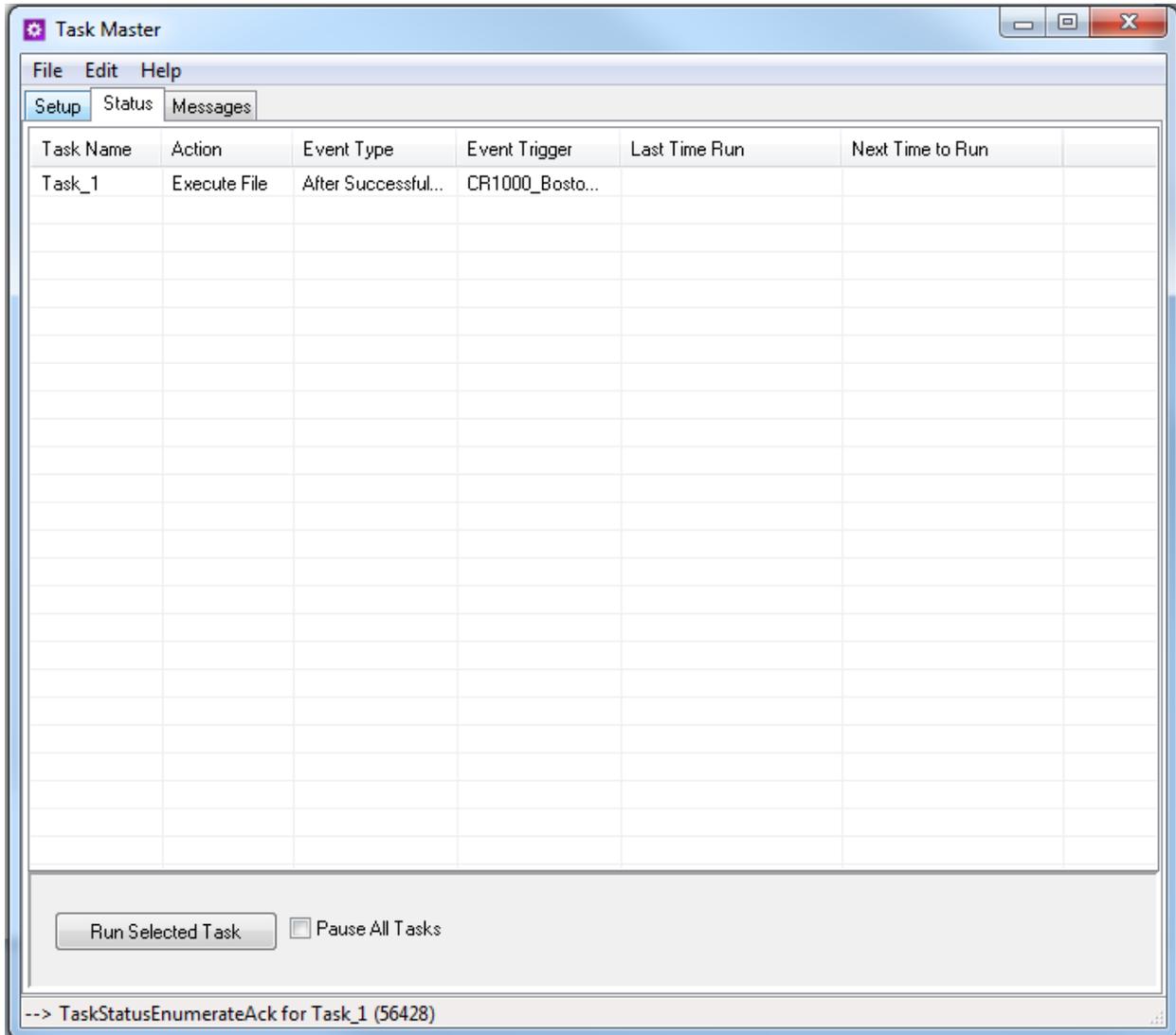


Figure 13.3: Viewing the status of scheduled tasks in Task Master.

## 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).

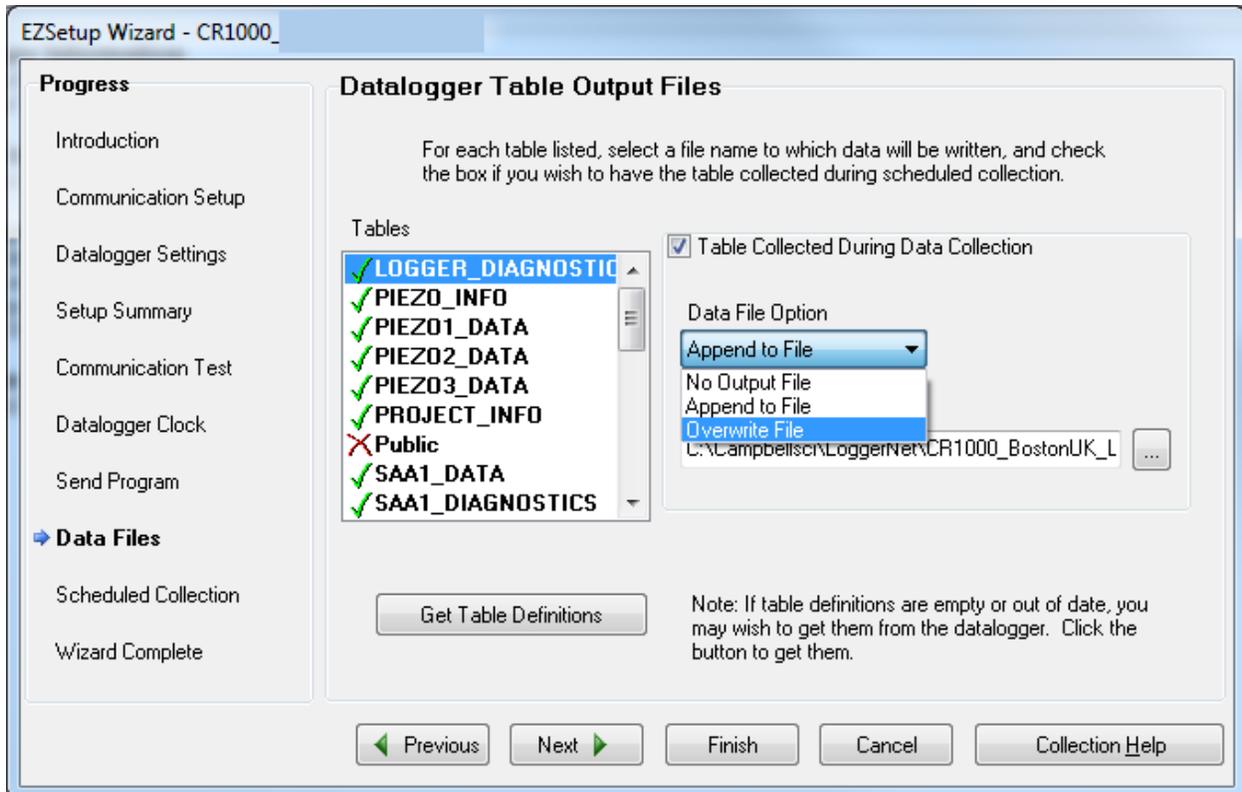


Figure 14.1: Switching the "Data File Output" option in the LoggerNet 'Setup' utility to 'Overwrite Existing File'.

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
```







```
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

' EPC Data Table

DataTable(EPC_DATA, True, -1)
  Sample(NUM_EPCS*6, EPC(), IEEE4)
EndTable

'SUBROUTINES
'-----
!*****
'Sub: GetArrayData
'Purpose: gets all of the array data
'Inputs: None
!*****

Sub GetArrayData()

Scan(1,sec,0,1)'need this dummy scan to auto-allocate data tables
  'fill in project info table:

  CallTable PROJECT_INFO

  Call GetLoggerDiagnostics()

  CallTable LOGGER_DIAGNOSTICS

```

'add calls to each SAA and store data in respective tables:

'-----

'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
```

```
CallTable SAA2_DATA
```

```
If SAA2_SAATOP>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 GetVWDData()
```

```
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)
```

```
AVW200 (Result(),COM3,0,200,SG(1,1),1,1,16,450,3000,2,_60Hz,1,0)
```

```
AVW200 (Result(),COM3,0,200,SG(17,1),2,1,4,450,3000,2,_60Hz,1,0)
```

```
CallTable SG_DATA
```

```
AVW200 (Result(),COM3,0,200,EPC(1,1),2,5,4,1400,3500,1,_60Hz,1,0)
```



'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

'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(2,7) = 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(2,13) = 51946

SERIAL\_NUMS(2,14) = 51970

SERIAL\_NUMS(2,15) = 51094

'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:

'-----

'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
```



