**Introduction**

This report describes the data logging for the CU MOSAiC project as it pertains to the Campbell Scientific loggers. There are four CR1000X data loggers active for the campaign, one at each of the three ASFS stations and another at the tower. There is also a reserve logger in the spare ASFS interface box and a CR1000 model logger in the MARC system. Each station has a unique logger program, including the three ASFS stations, which perform identical functions, but have some different parameter settings such as IP addresses, instrument calibration coefficients and Iridium modem dial schedules. The main scan sequence for the ASFS stations is 5 sec and for the tower station it is 1 sec.

The loggers are controlled using Campbell LoggerNet software version 4.5.0.1. This software is installed on the ship DAQ computers, the tower Cincoze computer, and the tablets. A series of “how to” style videos describing LoggerNet, logger programming, and the codes we are using are available with the project documentation.

**1. Data Collection and Archiving**

A schematic of the data flow for the ASFS is shown in Figure 1. The native functionality for accumulation of storage and data on the logger are data tables within the system’s internal drives, SRAM (~4 mb) and USR flash memory (~72 mb). Data tables are functionally similar to workspaces within analytical programming software, such as Matlab. Typically, data is stored in an ASCII file only on the remote DAQ and only after the data tables are collected by LoggerNet on that computer (however, in the case of the ASFS, a similar process is also used to store data locally on SD cards). A unique data table is maintained by the program for each data set that is to be accumulated. The ASFS systems have four data tables, and therefore four data sets, which are referred to as follows:

1) “fast”: This is the 20 Hz output from the Licor and the Metek.

2) “slow”: These are 1 min averages, standard deviations, etc of all ASFS data streams, including Licor and Metek.

3) “met”: These are samples of p, t, q, and positioning variables reported by the Vaisala and GPS at 5 sec intervals. The data are not averages, but are instead polled samples traceable to approximately 1 s clock accuracy. The same data appears also in “slow” as time averages. These were recorded at higher temporal resolution at the request of O. Persson (p,t,q) and J. Hutchings (GPS).

4) “sumfile”: These are 10 min averages of a critical subset of “slow” data that are sent by Iridium in the event that the ASFS loses radio contact with the ship.

*1.1. ASFS: Primary transfer and SD card backup*

When the logger makes a measurement, the data is first stored on data tables on the logger’s internal memory. The tables are ring buffers so once filled, the oldest data will be overwritten by the newest data. This internal memory is the most reliable available to us, but is also limited in size. The fast data accumulates very quickly and makes up about ~98% of the total data volume generated on the ASFS, but this volume is disproportionate to its value. To prevent the fast data from using up all the available memory (only storing a few hours of data), the size of the data tables were pre-allocated to prioritize preservation of the slow, met and sumfile data streams. Only 30 min of data (36,001 records) from the fast data table are stored and 6 hours (36 records of 10 min values) are stored for the sumfile. The remaining available data (10s of mb) are allocated evenly between slow and met data tables such that those tables will fill at the same time; this results in about 67 days of data storage for these tables.

Each ASFS logger has a 16 Gb single-level industrial-grade Delkin Devices microSD card. All data tables (except sumfile) are also archived on this card in ascii format (.dat) every 30 min. The naming convention of the files is the table name followed by an incrementing integer. An earlier version of the logger code renamed the files with a time stamp, but in testing it was determined that the action of renaming the files caused temporary lags in the program that resulted in a small number of skipped scans at 30 min intervals. Data quality was prioritized over convenience and so the renaming task was removed from the program. To see time stamps you would have to open the files.

If the radio is functioning properly, all data (except sumfile) will be collected in quasi-real time from the internal memory. Therefore, by default, the microSD card is meant to serve as a backup, extending the logger memory to provide a buffer for the time window during which data must be collected. Once filled, the oldest data on the microSD card will be erased to accommodate the newest data. Assuming a nominal data storage of 5 mb per hour, the SD card has enough storage space for about 133 days of measurements. How this system is managed (for example, removing and replacing cards, data collection methods, etc) will necessarily be a field decision and the attempt was made to provide some flexibility. Spare SD cards are stored at the Polarstern.

The SD cards are not infallible. It is preferable to minimize extraction and injection of the cards into the card slot on the logger. Data corruption is a risk and the electrical contacts are not designed for frequent removal and replacement.

**!**

If a card is removed, it is critical that it first be ejected by pressing the black eject button located next to the card on the logger surface. Only when the LED is **green** is it safe to remove the card. The light will remain green for 15 sec. If the light is **red**, DO NOT remove the card; data is actively being written and removal is likely to result in corruption. If the light is **orange**, there is an error.

Data from fast, slow, met and sumfile are accumulated on the logger’s internal memory and retrieved from this memory for final archiving at the ship DAQ every 5 minutes. The primary mode of data collection and archiving is adapted from the system used operationally by NOAA-GMD. The DAQ LoggerNet software collects the data and appends it to an ASCII file on the DAQ archive at the ship here in D:/asfs#/. There is one file accumulating for each of the three tables, “fast”, “slow” and “met”. At 4 min past 0 UTC each day, a Python script (asfs\_rename.py) on the DAQ is executed by the Windows Task Manager. This renames and moves the files the the following location:

D:/data/asfs#/daily\_files/ASFS#\_fast\_YYYYMMDDhhmm.dat

D:/data/asfs#/ daily\_files/ASFS#\_slow\_YYYYMMDDhhmm.dat

D:/data/asfs#/ daily\_files/ASFS#\_met\_YYYYMMDDhhmm.dat

Therefore, three data sets are maintained for each ASFS and daily files are automatically created and archived. After being moved and renamed, accumulating files in D:/asfs#/ are regenerated automatically at the next collection (about 1 min after the move).

*1.2. ASFS: Iridium*

In the situation where radio communications between ship and ASFS are down, status updates are sent to the DAQ at Polarstern via Iridium at 6 hour intervals. Once every scan the ASFS pings the ship through the radio. If the ping fails, at 0, 6, 12 and 18 UTC, the Iridium modem will first be turned on, then initialized and then will initiate a “Callback” with the ship via Iridium. The callback instructs LoggerNet to collect the previous 6 hours of 10 min averages (36 lines) from the sumfile data table. This data is appended to D:/asfs#/ASFS#\_sumfile. The sumfile perpetually accumulates and is not transferred to the daily archive at the end of the day. The messages are staggered between the stations by 11 minutes (so 0, 11 min and 22 min after the 6 hour mark) to avoid concurrent transmissions, which may result in a busy signal at the Polarstern and thus a failed transfer. In this mode, data is still being archived on the microSD card for retrieval later, either by a direct connect during a visit to the station or via remote download from the card when radio comms are re-established. In tests, the reliability of this system was intermittent though we are hopeful that these problems were resolved late in testing. If the transfers do not arrive as expected, it is possible to manually initiate a collection from LoggerNet at the ship (the Collect Now button). The catch is that the modem on the ASFS must be on and ready to receive a call, which only occurs for a few minutes during the procedure. The schedule (below) begins at the times so time = 0 (minutes) in the schedule refers to the times below. The instructions can be found proximately between lines 606 and 695 in the logger program.

ASFS30: 00:00, 06:00, 12:00, and 18:00 UTC

ASFS40: 00:11, 06:11, 12:11, and 18:11 UTC

ASFS50: 00:22, 06:22, 12:22, and 18:22 UTC

Time = 0 min: Iridium modem is powered on

Time = 1 min: Iridium modem is configured

Time = 3 min: Iridium is registered and the call to the ship is made

Time = 4 min: The ASFS sends a callback to the ship that initiates a LoggerNet Task, which immediately Collects the data from the sumfile table. If this is successful, tests in Boulder indicate that it happens quickly (seconds).

Time = 10 min: Iridium modem is shut off

Between about minute 5 and minute 10 there is either an active phone call with no scheduled transmissions (an open line) or the call failed but the ASFS modem is powered, set up and registered. During this 5 min period, it is therefore possible to attempt to manually connect to the logger from the LoggerNet Connect Screen. Make sure to highlight the ASFS\_#\_Iridium station on the Connect screen: the information needed to make the call is stored in the LoggerNet Setup and the connection should happen automatically when you click Connect regardless of the status of the call, though it might take a while to establish connection.

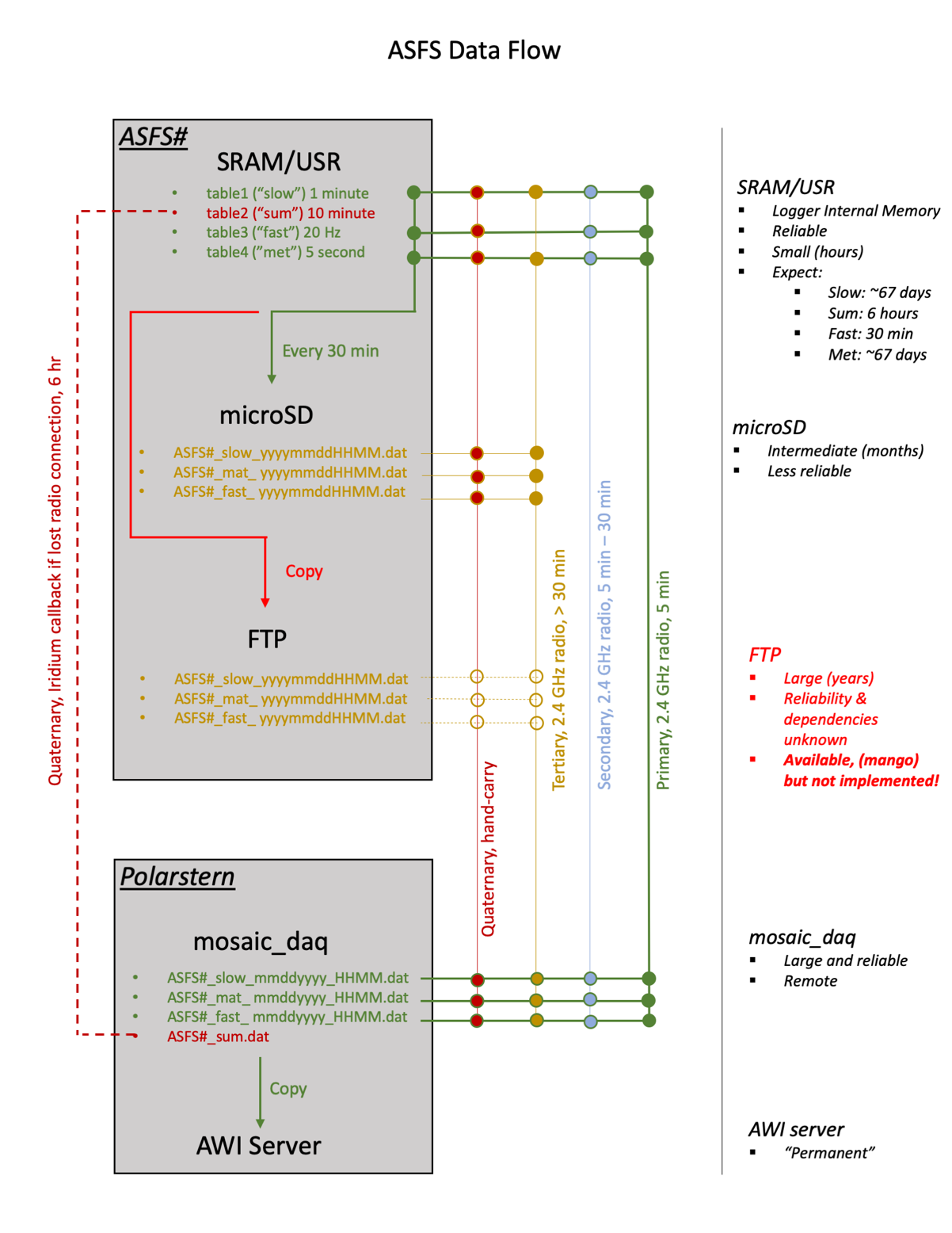


Figure 1. ASFS Data Flow.

**2. Fault Tolerance**

*2.1. Logger scan time and instrument response time*

The scan schedule is 5 sec for the ASFS. All instructions must be completed within the scan time. If the time taken to complete the scan exceeds 5 sec the scan is skipped if the delay is sufficiently long to overrun the buffer. The actual scan time for the ASFS varies, but is generally between 2.8 and 3.6 sec, providing some cushion.

If an instrument for some reason loses connection with the logger, failed attempts at communication with the instrument by the logger within the scan cycle can slow the progress of the scan, potentially leading to skipped scans. Additionally, in a previous campaign we noticed that response times increased at cold temperatures for some instruments; no temperature dependencies were found during testing of the ASFS at -25 C in the freezer in INSTAAR.

We configured the response wait time and number of timeouts for the RS485 instruments to prioritize speed: Only 1-2 attempts are allowed with a timeout of 15 ms. This slightly increases the risk that an individual polling of an instrument fails, but decreases the risk that a malfunctioning instrument cause problems that affect the rest of the system; i.e., the program is more likely to skip and instrument instruction within the scan than to skip the scan because of a delay in a single instruction. These parameters are set in the ModbusMaster function that is called 11 times between lines 858 and 920 in the logger program.

We also tested the resiliency of the program against instrument failures was tested for all instruments by disconnecting each in turn and checking for a change in the timing of the scan cycle followed by re-connecting the instrument and checking that data collection resumed. This is not a potential issue for the IR20s and flux plates, which are passive analog sensors. Loss of the SDI-12 data streams (IRT and SR50A) results in either no change in the scan cycle or a decrease in the scan duration, and both sensors come back online when re-connected. Loss of the GPS results in no delays and data returns to normal when the GPS is re-connected. Loss of the Licor and Metek does not affect scan time, though note that unexpected messages from either of these instruments can affect scan time, probably because of the potential for conflicting data types. A single MODBUS call is made for Vaisala. We have not observed any missed communications, suggesting this is adequate for the sensor. In practice, if Vaisala is disconnected, the call actually increases the scan by 100 ms, possibly due to processing of empty variables. However, this is minimal disruption and is acceptable. The Vaisala comes back online when re-connected. For the SR30s, due to the way that the registers are organized, MODBUS calls are made 3 separate times for each instrument in each scan in order to collect all of the necessary variables. Two attempts are granted per scan for the SR30 science data (first MODBUS call) and 1 attempt for the calls that acquire housekeeping data. If disconnected, the SR30 delay could be up to +400 ms. Attempts were made to manage this by negating the timeout parameter, allowing for the wait time to be performed in the background. While the wiring of the RS-485 sensors permits this, conflicts with changes in baud rate required to contact the EFOY were observed to corrupt data and the wait time was set to 15 ms. The SR30s come back online when re-connected. Four MOBUS calls are required for the EFOY, and it was observed to cause about +400 ms increase in the scan cycle when disconnected. The EFOY is not RS-485, but is rather MODBUS RS-232, a non-standard protocol. Consequently, there is little that can be done to mitigate the delay. A single attempt for call is made with a 15 ms timeout. EFOY comes back online when re-connected. However, note that some variables (especially temperatures) requested from EFOY appear to sometimes fail. A full accounting of this problem has not been carried out because sufficient amounts of data are collected and EFOY data is housekeeping rather than scientific.

The ASFS is therefore expected to be robust to failures of one or two instruments. Skipped scans could still occur rarely for various reasons. Iridium calls will completely stop data collection. If the scan time increases and skipped scans occur regularly, it will be necessary to modify the program. For example, if an SR30 fails and skipped scan occur, comment out the MODBUS calls for the problem instrument and upload the new program. If instruments continue to work but slow down too much (for example in very cold weather), it may be necessary to disable some of the MODBUS calls that only collect housekeeping data (again, by commenting out selected ModbusMaster calls and uploading the new program). Remember, if you do this that uploading a new program will erase data on the logger’s internal memory!

*2.2. Rebooting CR1000X*

The loss of the network, or loss of network connection to other devices, is the only reason to reboot the logger that we saw in testing. We have successfully implemented a network socket reset for the Metek, which will engage automatically is needed. The equivalent was unsuccessful for the Licor. The logger itself must be setup in the Device Configuration to have the ethernet port set to Always On. If this does not happen, the logger may not reconnect to the instruments if that connection is temporarily lost without rebooting the OS. In DevConfig, this option can be set under the Ethernet tab above the setting for the IP address. Setting the ethernet to Always On should mitigate that problem. However, if an unforeseen circumstance requires the need to reboot the logger for any reason, there are some things to know. There are three ways to reboot the logger OS:

*(1) Remote restarts* ***PREFERRED OPTION, try this first***

This option is safe (no data is deleted) and can be performed remotely. Therefore, it is the first option you should try. However, it is a sort of cheat and its effectiveness has not been extensively tested. In the program a **ConstTable** called **restarts** is defined. A dummy variable called “restartstable” is stated within. The variable is meaningless and is only used to activate the table, which enables an option for restating. If you Connect to the logger and go to the restarts table from the dropdown menu, you will see a variable called ApplyAndRestart that is set to false. This is an “editable constant”: Right-click on the word “false”, then click “View/Modify Value”, then change the false to true, then click ok. The logger will reset without recompiling the program. Check an see if this solved your problem.

*(2) Power cycle it by unplugging its power source*

This is a safe option because it resets the logger without recompiling the program, which would result in the loss of all data on the internal memory. However, it is impractical because it is impossible to do this remotely.

*(3) Upload a new program through the Connect screen*

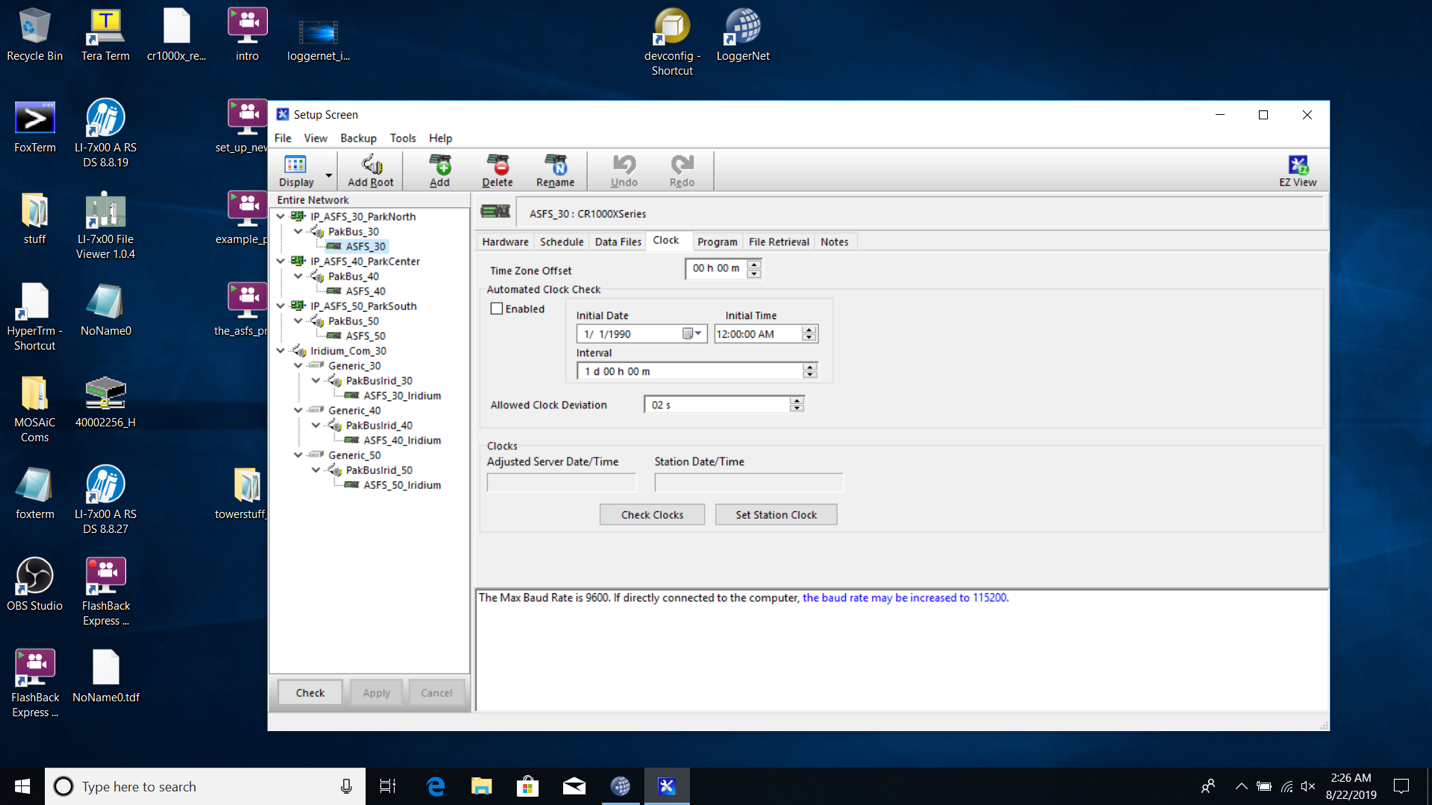
If you do this, **make sure** that you have manually collected the data and confirm that all data that could reside on the logger has been successfully retrieved. You are about to delete the data on the logger! Uploading a new program will delete all data saved on the logger internal memory (data on the SD card is preserved). This option can be performed remotely if the logger can be contacted by the radio. Since the upload will be a relatively large file, consider temporarily disabling the scheduled collection from the other loggers to free up the radio.

**3. Accuracy**

*3.1. Timekeeping*

All ASFS measurements are time stamped using the CR1000X logger clock. Clocks are also integrated into some of the ASFS instruments (e.g., Licor and Metek), but the accuracy of those clocks are not maintained. The accuracy of the clock on the CR1000X is specified to ±3 min per year, and therefore the expected drift is up to 0.5 sec per day. There are three synching methods available to the ASFS. The primary is implemented. We considered implementing a secondary method using a less strict tolerance so that it would take over if the primary failed. However, we opted not to do this to avoid potential conflicts between the two methods. Therefore, backup time synching will not automatically engage on the ASFS if a failiure in the primary (GPS) occurs. If problems with the GPS are suspected, a new method will need to be chosen an implemented.

* Primary: Time on the ASFS is synched using the on-board Vector V102 GPS. This function is performed using the GPS service within the logger’s OS. At each scan (every 5 sec), the most recent 1 Hz NMEA messages are ingested for reporting of lat/lon/alt and satellite quality. The reported GPS time (NMEA $GPGZDA) is automatically compared to the logger’s clock by the CRBasic GPS function. If the difference exceeds a threshold set by a constant in the program, the clock is reset. This threshold is set to 499 msec. We therefore expect the clock to be reset by 0.499 sec about once per day. This is too small of a shift to affect the time stamps of the slow, met or sumfile tables, but discontinuities will likely appear occasionally in the fast data. No blackout on the fast data is imposed so these periods will need to be identified and managed in post-processing. The CRBasic function that does the synching is the GPS function and it is called at line 705 of the ASFS logger program.
* Alternative Method 1 LoggerNet: Under the third level of the LoggerNet Setup Screen’s communications tree, within the Clock tab (see image), you can enable LoggerNet to manage the logger’s clock using the time from the computer you are running LoggerNet from. There are settings for how often to check the clock and how much deviation to allow in this tab. The must be done individually for each station that requires it. It will only work when there is radio communications with the ASFS station.



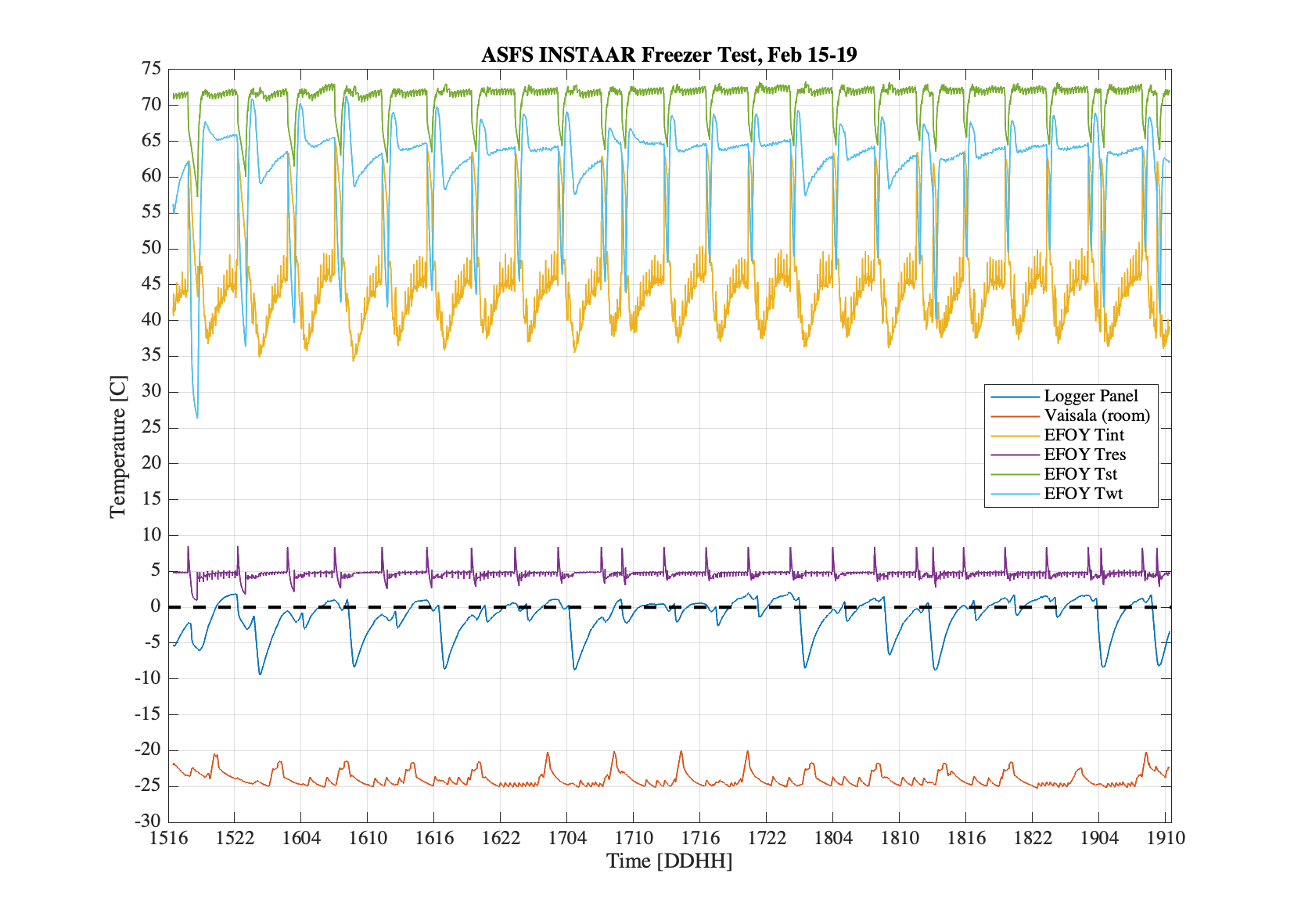
*Screenshot showing the set up screen for managing the clock on ASFS30 from the DAQ using LoggerNet. This service is currently OFF and would need to be enabled (check box) to turn it on. Note the other settings and adjust as necessary.*

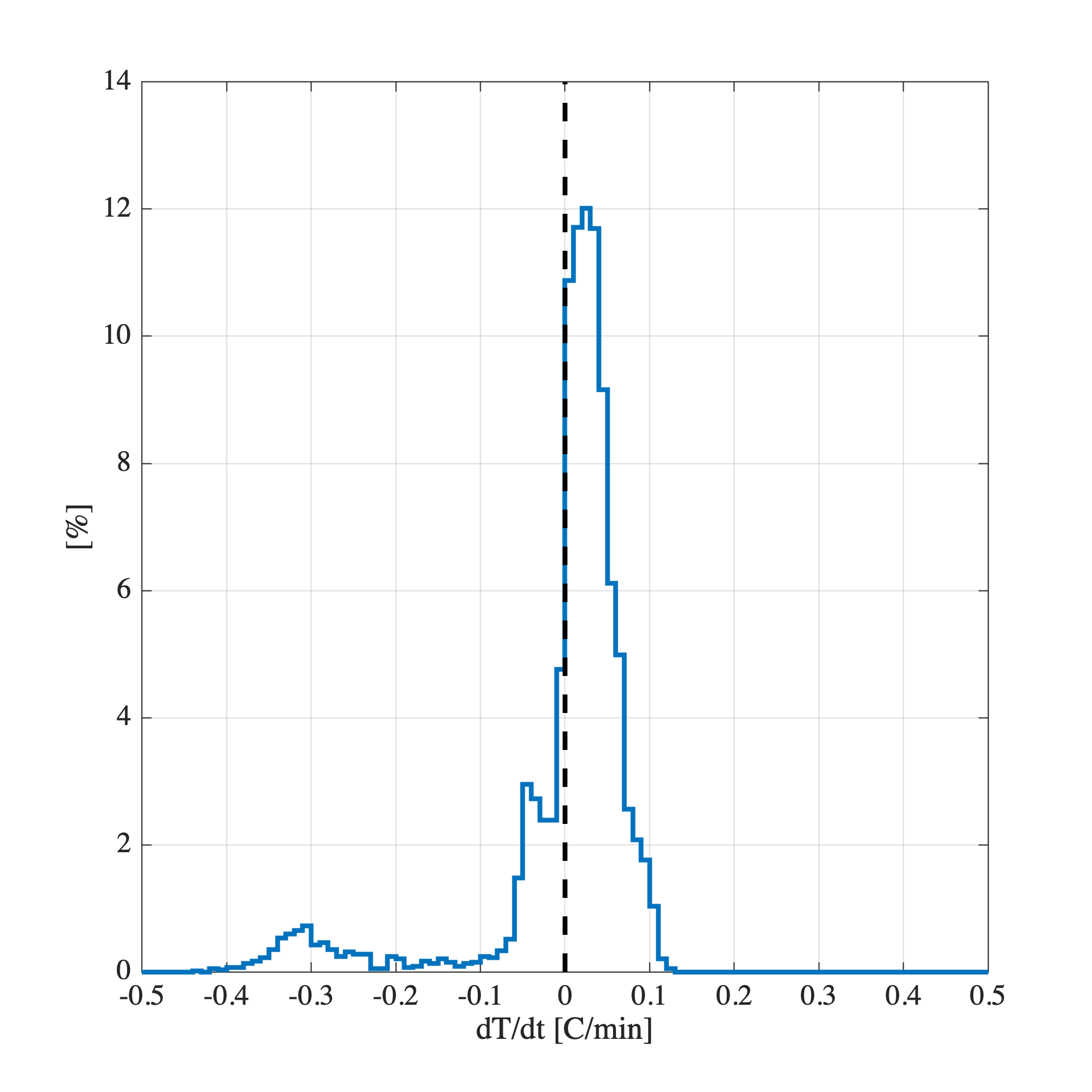
* Alternative Method 2 NTP: The logger has the capability to interact with an NTP server using the NetowrkTimeProtocol CRBasic function. The function is called at line 597 of the logger program. In the current configuration, NetworkTimeProtocol is used with open quotes in place of an NTP IP so that the logger itself acts as an NTP server. This was done to facilitate time keeping for the ASFS camera. If all other methods fail, it is possible to edit the code to provide the ship NTP server IP to NetworkTimeProtocol by changing NTPServIP at line 54 to whatever the ship’s NTP IP is and changing the ““ at argument one of NetworkTimeProtocol at line 597 to say NTPServIP (or alternatively replace “” in NetworkTimeProtocol with a string containing the ship’s NTP IP). We have not tested this and you should review the NetworkTimeProtocol help if you try to implement this.

*3.2. Analog channel calibration*

Tests of the ASFS power system and instrumentation/logging were carried out in a freezer at INSTAAR over several days in February 2019 (Figure). The freezer air temperature was between -20 and -25 C (red line) and the temperature of the logger electronics was recorded (blue line). The power system cools itself using a fan that exhausts internal air near from the batteries allowing outside air to enter through an inlet near the interface box causing outside air to be dumped directly onto the logger. This happened at least 9 times over the 83-hour test and is observable in the figure, which shows a rapid cooling of the logger from near 0 C to near -8 C followed by a more gradual warming back to 0 C. The rate maximum rate of change of the panel temperature was between -0.3 and -0.4 C per min (Figure).

These rapid changes in temperature are concerning primarily because the accuracy of the analog measurements made by the logger are temperature dependent. The logger automatically recalibrates using an internal reference voltage approximately every 36 to 364 seconds to adjust for changes in temperature and aging. This feature is disabled when the logger is unable to perform the function within the scan interval (i.e., it prioritizes the code instructions). In the range of -55 to 85 C the specified accuracy is ±0.08%, increasing to ±0.8% if calibration is disabled. The accuracy is reported to degrade by about 0.01% per degree temperature change. It is possible to override the automated calibration procedure by forcing a calibration at a desired interval, but if the freezer test is considered representative, the automated interval is sufficient. This error propagates linearly into the radiometer thermopile and flux plate fluxes and thus is roughly an order of magnitude smaller than the instrument calibration uncertainty over a 10 C change in temperature. It also has negligible error on temperature derived from resistance measurements. The largest error comes from the case temperature of the pyrgeometer, which is represented as flux in the calibration equation (4th power) and results in about 0.1 W m-2.





**4. Logger and LoggerNet settings**

*4.1. CR1000X Operating System*

The OS for CR1000X that we are using is CR1000X.Std.03.02. The CR1000X loggers (ASFS & Tower) already have this OS, but instructions for the update are below. Note that the logger in the MARC system is a CR1000 and has a different OS.

1) In our documentation archive go to software/Campbell/OS\_for\_CR1000X/ and take the CR1000X\_OS.03.02.exe to the computer you will be using to connect to the logger. Also take note of the IP address of the logger you are updating if it exists.

2) Run the CR1000X\_OS.03.02.exe executable on the computer. It will say that a .obj file has been sent to a local Campbell directory on the computer. Take note of this directory.

3) Connect to the logger using a USB to micro-usb and open the DeviceConfig utility from LoggerNet. In the bottom left Communication Port connection screen of DevConfig uncheck "Use IP Connection" and connect choosing the USB com port (click the "..." button). You may have to install the driver: If so follow instructions on big blue button to right under the picture of the logger ("CR1000X Series" tab on the main DevConfig screen). You can either click

connect and find the "Send OS" tab or click the "Send OS" tab on the main screen.

4) Click start and navigate to the obj file. Install it. Should take about a minute.

5) The logger is back to factory settings. Connect to the logger using the usb cable in DevConfig. Under the Ethernet tab enter the logger IP address. Save. Disconnect. Reconnect using the "Use IP Connection" option to test. It should work. If not and there are no lights on the IP port of the logger or switch you might need to go back to usb and under the Ethernet tab change Ethernet Power to Always On.

**5. Troubleshooting**

*5.1. LoggeNet Terminal Emulator*

Loggernet has its own terminal emulator that can be used similarly to Tera Term, FoxTerm, etc. It is not expected that you will need this, but it is useful if an instrument needs to be configured, for example to send a new variable, change the sonic heating mode or to reconfigure a device if it loses its configuration for some reason.

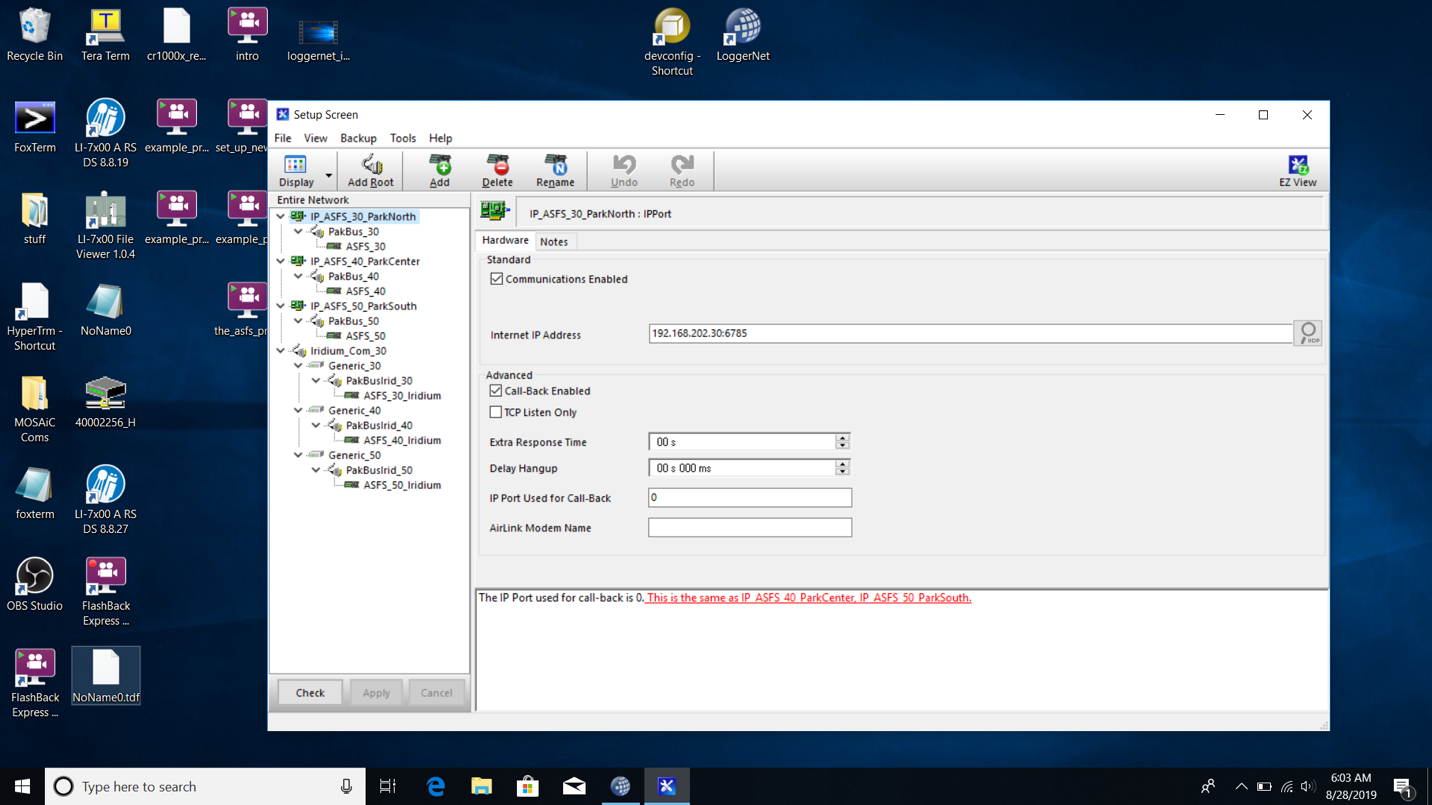
Using Loggernet’s terminal emulator allows you to communicate with all of the digital instruments on the ASFS with the exception of the Licor, which must be configured using its own software (even remotely). To use the terminal, from Connect Screen, Datalogger dropdown -> Terminal Emulator. Select a baud rate: this is the baud rate LoggerNet will use to connect with the logger and therefore need not match the baud rate of the device you are contacting, but selecting a slower baud rate may be more successful over the radio. The click Open Terminal and hit enter a couple times until you see CR1000X>, indicating you are connected. There are three useful modes. If you want to watch the serial activity on a channel, use the “watch” mode by typing “W” (no quotes) and hitting enter. You will see a list of channels. Select your channel by entering the number it is listed under and clicking enter. You should see output appear immediately if there are serial streams being sent through that channel. You only see received data on the channel and cannot send data. To send data, use the talk-thru mode (“P”). In talk-thru mode, you can sometimes receive data too depending on the protocol. The RS-485 (EFOY, Vaisala, SR30, sonic) and RS-232 (GPS) can be configured through talk-thru. There is also an SDI-12 mode (“SDI12”). Use this to communicate with the IRT or SR50a ranger. After you are done with any mode, Close Terminal. Consult the individual instrument manuals for commands and syntax.

*5.2. LoggerNet Connect Screen*

You can make a connection to the logger from the LoggerNet Connect screen. This allows you to access all the information being collected by the logger, including all defined variables from the program (Public screen) as well as internal housekeeping (Status screen) and information associated with individual or all Data Tables. This information may be very useful in troubleshooting. However, while we used this feature frequently in testing at ESRL you should only use it during MOSAiC when you must. Older models of loggers and older versions of LoggerNet actually stopped all data collection when a connection was made through the Connect screen. The loggers in use at MOSAiC will continue to collect data when you are connected to them, but the connection puts unnecessary stress on the system and takes a large amount of bandwidth on the radio. For normal monitoring purposes please review data that is already collected and stored locally. The RTMC, View Pro, and Status Monitor utilities in LoggerNet are better suited for monitoring.

*5.3. Loggernet Setup Backups*

The configuration for the communications between LoggerNet at the DAQ and the ASFS loggers is found in the LoggerNet setup screen:



There is a separate communications tree for each ASFS station, and each station also has a separate communications tree for IP (radio) comms and Iridium. LoggerNet at the tower Cincoze has a communications tree for working with the logger there, but it is much simpler than the one for the ASFS. It is likely that during the campaign adjustments will be made to these settings, perhaps even often. It is possible to manually backup all of the settings from the Backup dropdown menu in the top left corner of the Setup Screen. From this same menu it is also possible to restore the settings from a previously backed-up Setup. We have backups of the LoggerNet Setup screen configurations for the LoggerNet installations on both the ship DAQ and the tower Cincoze based on the settings on 24 July 2019. The backup files are found with the documentation in the directory

MOSAiC\_Documentation/Software/Campbell/loggernet\_config\_backups/

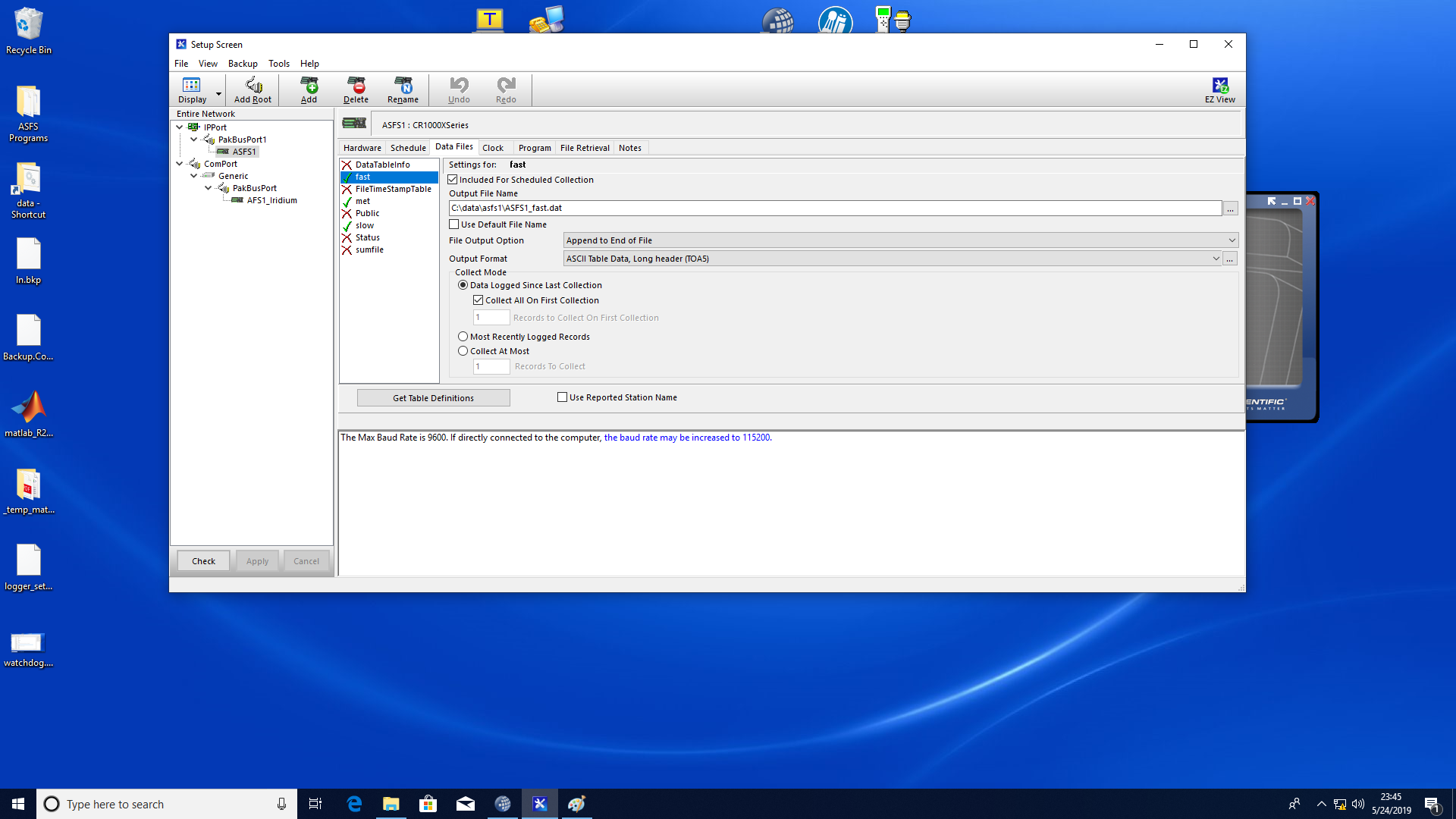
The Backup menu also has a way to create a task that automatically performs scheduled backups. This was not implemented in Boulder, but is maybe worth considering.

*5.4. If the radio can’t keep up with the ASFS data transfers…*

If the radio is functioning but is not cannot transfer the data fast enough to keep up, it may be necessary to stop collecting the fast data files, which are approximately 98% of the total data volume. This would implement the tertiary (brown) mode of data collection in Figure 1: only data from the slow and met data tables would be collected every 5 min. The fast data would continue to be stored automatically by the station on the SD card to be downloaded manually during the next visit to the ASFS station (they could be retrieved remotely manually using the LoggerNet DevConfig utility or the web browser).

This can be implemented independently for different stations. For example, radio trouble might only be a problem with one station and so normal fast/slow/met data collections are continued for two stations while only slow/met are collected from the third. You could also stop scheduled collections for the met data, if necessary.

To implement this, open the LoggerNet Setup screen and highlight the lowest level of the communications tree of the logger you need to change (in the image below “ASFS1” for example). Then navigate to the Data Files tab (highlighted in the image). This lists all of the data tables associated with that logger. Data tables with a green checkmark are part of the scheduled collection (the parameters of the schedule are found under the Schedule tab) and tables with a red X are excluded from the collection. The image shows the normal setup with fast, met and slow being part of the scheduled collection (sumfile is checked under the logger’s Iridium comms tree). You should highlight the fast table and turn off collection by unchecking the check box “Included for Scheduled Collection”. When this has been done the DAQ will stop collecting the fast data.



*5.4. Logger Operating System*

The CR1000X logger OS/firmware is CR1000X.Std.03.02 (ver. 22 Apr 2019). lt There is probably no reason that this would need to be reinstalled on the loggers, but just the instructions are below. Campbell’s recommendation is that you do not attempt this using a remote connection.

1) In our documentation archive go to MOSAiC\_Documentation/software/Campbell/ and take the CR1000X\_OS.03.02.exe to the computer you will use to connect to the logger. Also take note of the IP address of the logger you are working.

2) Run the executable on the computer. It will say that a .obj has been sent to a local Campbell directory on the computer. Take note of this directory.

3) Connect to the logger using a USB to micro-usb and open the DeviceConfig utility from LoggerNet. In the bottom left Communication Port connection screen of DevConfig uncheck "Use IP Connection" and connect choosing the USB com port (click the "..." button). You may have to install the driver: If so follow instructions on big blue button to right under the picture of the logger ("CR1000X Series" tab on the main DevConfig screen).You can either click

connect and find the "Send OS" tab or click the "Send OS" tab on the main screen.

4) Click start and navigate to the obj file. Install it. Should take about a minute.

5) The logger is back to factory settings. Connect to the logger using the usb cable in DevConfig. Under the Ethernet tab enter the logger IP address. Save. Disconnect. Reconnect using the "Use IP Connection" option to test. It should work. If not and there are no lights on the IP port of the logger or switch you might need to go back to usb and under the Ethernet tab change Ethernet Power to Always On.

*5.5. Uploading a new program*

If you need to upload a new program, the data **must be collected** **first.** When a program is uploaded to the logger (even if it is the same program), it is compiled and the data tables are reinitialized for the newly compiled program. Consequently, all data on the logger’s internal memory (USR and SRAM) is permanently deleted! The logger will warn you about this when you try to upload a program. The data on the SD card will be safe because of the way that we are using the SD card (using the SD card with “option 64” is mostly, but not 100% safe, but as of August 2019 we are not planning to do this at MOSAiC). To upload the program, open the LoggerNet Connect screen and click Send New under Current Program on the right-hand side of the screen.

When you create a new program, save the file with a unique version number and also add that number to the PrgVersion string that is on Line 1 of the code. This is our (rudimentary) method for version tracking.