Naval Research Laboratory

Washington, DC 20375-5320

NRL/MR/6930--20-10,065

Reflectance-Based Sensing: Data and Device Outputs

Brandy J. White

Jeffrey S. Erickson

Laboratory for the Study of Molecular Interfacial Interactions
Center for Bio/Molecular Science & Engineering

Anthony P. Malanoski

Laboratory for Biosensors and Biomaterials
Center for Bio/Molecular Science & Engineering

July 1, 2020

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
01-07-2020	NRL Memorandum Report	08/10/2019 – 11/01/2019

4. TITLE AND SUBTITLE

Reflectance-Based Sensing: Data and Device Outputs

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Brandy J. White, Jeffrey S. Erickson, and Anthony P. Malanoski

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
6A26

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/MR/6930--20-10,065

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375-5320

10. SPONSOR / MONITOR'S ACRONYM(S)
NRL 6.2

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report is work focused on a component of an effort intended to develop wireless sensor networks for real-time monitoring of airborne targets across a broad area. Prior reporting on this effort has captured design of the six element array, prototype hardware, and algorithms as well as extensive testing of that system. The current document provides a brief description of a new 15 element prototype device with details on the files and data provided following use of the devices. Guidance for use of the prototypes via the in-house designed user interfaces is also provided.

15. SUBJECT TERMS

Environmental sensor Chemical sensor
Colorimetric Porphyrin

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Brandy J. White
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	Unclassified Unlimited	34	**19b. TELEPHONE NUMBER** *(include area code)*
Unclassified Unlimited	Unclassified Unlimited	Unclassified Unlimited			(202) 404-6100

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

This page intentionally left blank.

CONTENTS

FIGURES

TABLES

EXECUTIVE SUMMARY

In October 2012, the Center for Bio/Molecular Science and Engineering at the Naval Research Laboratory (NRL) began an effort intended to develop wireless sensor networks for real-time monitoring of airborne targets across a broad area. The goal was to apply the spectrophotometric characteristics of porphyrins and metalloporphyrins in a colorimetric array for detection and discrimination of changes in the chemical composition of environmental air samples. Prior reporting on this effort has captured design of the six element array, prototype hardware, and algorithms as well as extensive testing of that system. The current document provides a brief description of a new 15 element prototype device with details on the files and data provided following use of the devices.

This page intentionally left blank.

REFLECTANCE-BASED SENSING: DATA AND DEVICE OUTPUTS

INTRODUCTION

In October 2012, the Center for Bio/Molecular Science and Engineering at the Naval Research Laboratory (NRL) began an effort (69-6594) intended to develop wireless sensor networks for real-time monitoring of airborne targets. The goal was to apply the spectrophotometric characteristics of porphyrins and metalloporphyrins in a colorimetric array for detection and discrimination of changes in the chemical composition of environmental air samples. The effort encompasses hardware, software, and firmware development as well as development of algorithms for identification of event occurrence and discrimination of targets.[1-5] Prior reporting on the devices, versions 1 through 2.08, has addressed the development of a six element array, six element prototype hardware (Figure 1), and relevant detection algorithms as well as extensive testing of that system focused on exposure to either vapors or aerosols. Here, we briefly describe a new prototype iteration for the Array Based Environmental Air Monitor (ABEAM v3) and provide details on the output and use variations for the devices.

Fig. 1 — The six element prototype device includes six color sensing breakout boards, a custom control board, fans, and indicator supports with custom housing. The device requires external power and is controlled by a laptop computer.

The ABEAM v3.2 device iteration is specifically designed to provide isolation of the electronics from the environment as well as from targets. This type of isolation is designed to prevent the failures experienced by the six element prototype during Cl_2 exposures and would be expected to extend overall device durability.[5] Beyond protection of the electronic components, the v3.2 device iteration provides an array of 15 indicators, occupying a footprint of 7.5” x 3.5” x 4.63” at a weight of 293 g (2,000 g with battery pack and housing; Figure 2). For comparison, the six element prototype was 10.8” x 3” x 3” and weighed 1,585 g as well as requiring an external power source. Incorporation of 15 indicators provides the potential for greater target discrimination based on the relative response across the array elements. For example, indicators with greater sensitivity to VX could be incorporated to improve the performance against those targets noted in the previous study.[5] Indicators with lesser sensitivity to Cl_2 and HD mustard could be used to provide improved discrimination between those targets.

METHODS

Meso-tetra(4-sulfonatophenyl) porphyrin (S_4TPP), meso-tetra(4-aminophenyl) porphyrin (N_4TPP), and Deuteroporphyrin IX 2,4 bis ethylene glycol (DIX) were obtained from Frontier Scientific (Logan, UT). Metalloporphyrin variants of S_4TPP, N_4TPP, and DIX were prepared by reflux.[1-3] The porphyrin (20 mg) was dissolved in water (4 mL) or dimethyl sulfoxide (DIX only). The metal salt was added to this solution in a 3:1 molar ratio with the porphyrin. The total volume was brought to 100 mL with deionized water and refluxed overnight. The volume of the resulting solution was reduced to 10 mL through rotary

Manuscript approved April 15, 2020.

evaporation. Prepared porphyrin solutions were stored in the dark at room temperature. The metal salts used here were: copper (II) chloride, gold (III) chloride, silver chloride, and thallium (III) chloride. Paper supported porphyrin indicators were prepared on Whatman Filter Paper by spotting 5 μL of the porphyrin solution (2 mg/mL) onto the appropriate area of the coupon (Figure 3). [2, 3] 'Blank' indicator spots used ink from a red Sharpie® to provide a non-responsive, colored indicator. Samples were dried at 100°C before storing in the dark in foil wrappers.

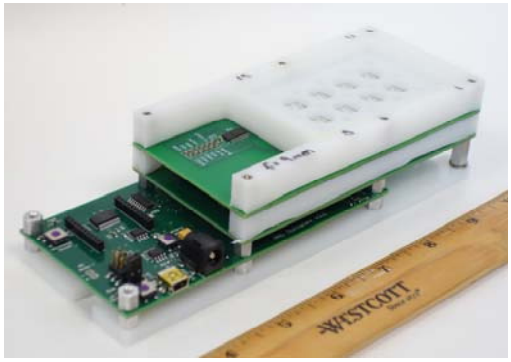


Fig. 2 — The v3.2 prototype iteration includes fifteen surface mount color sensors with custom control board, wireless communications, and can be powered using a battery pack.

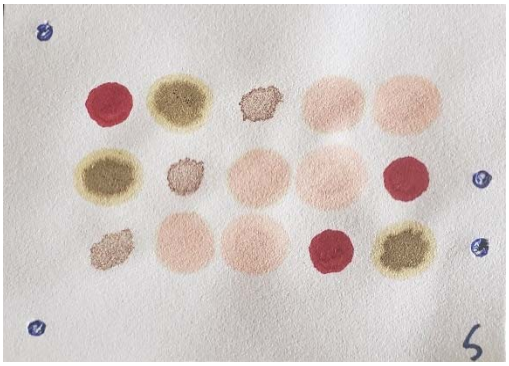


Fig. 3 — The coupon layout used for the trials described here unless otherwise indicated. This coupon includes three copies of each of four indicators as well as three negative control spots.

5 Red Sharpie®	4 AgN_4TPP	3 TlDIX	2 CuS_4TPP	1 AuS_4TPP
10 AgN_4TPP	9 TlDIX	8 CuS_4TPP	7 AuS_4TPP	6 Red Sharpie®
15 TlDIX	14 CuS_4TPP	13 AuS_4TPP	12 Red Sharpie®	11 AgN_4TPP

The original six element prototype reflectance instrument developed by NRL utilizes low cost, commercially available color sensing breakout boards from Parallax, Inc. (model TCS3200-DB, Rocklin, CA), providing a color light-to-frequency integrated circuit from AMS (model TCS3200, Plano, TX), a pair of white LEDs, and an adjustable lens.[1] The device output consists of a stream of digital pulses proportional to the intensity of the color being measured. A custom printed circuit board (PCB) interfaces with and controls six of the commercial color sensors. Communications between the instrument and the computer are via USB; power is supplied through a DC barrel jack. A LabWindows developed software-based graphical user interface (GUI) communicates with the PCB firmware through simple ASCII commands. The prototype sensor device tested through independent evaluation is a slightly modified version of that original NRL device (v2.08) [1-6] (Figure 1). Airflow through the sample tube at 2.7 CFM is provided by two small 5 VDC fans (Orion Fans, model #OD2510-05HB), one mounted at each end.

The new v3.2 fifteen element prototype device was completely redesigned based on user feedback and experimental data. Rather than being composed of a single circuit board with an exposed "wind tunnel" sensor and illumination design, the new device is a stack of three boards. This strategy allowed for a vertical light path directly into the device, completely encapsulating both the electronic and the optical elements in a single compact package and physically protecting them from environmental exposure. The bottom layer board is the heart of the instrument; it consists of power management, data storage, communications, and

processing. The middle layer contains the RGB sensors. In between the middle and top layers, a machined plastic mount holds a set of molded aspheric lenses which are used to focus the reflected light onto the sensors. The top layer board contains eight LEDs, each with an emission profile that has a maximum intensity at 45 degrees, an ideal incidence angle for reflectance measurements. It also contains holes for the reflected light to access the RGB sensors located below. A plastic mount above the top circuit board spaces the sample at the proper distance from both the LEDs and the lenses. A glass plate seals the electronics from the environment while allowing light transmission to the sample.

The older TCS3200-DB breakout boards were replaced with TCS34725 surface mount RGB sensors. Rather than a pulse train proportional to intensity, these sensors output a voltage proportional to intensity of light being measured. This allows a much higher data throughput, resulting in faster cycles than those of the v2.08 board even with double the number of sensing elements. As an example, the v2.08 design could only perform a 5 s sampling cycle at the fastest integration time, 100 ms. The new design can perform this cycle at integration times up to 600 ms. In addition, the new ABEAM v3.2 design includes expanded flash memory, electronics for battery management, and wireless communications. Finally, the all of the software has been completely re-written in Java. The user interface has been completely re-written using Java FX, which removed many of the limitations of the old LabWindows based GUI. The software now contains a suite of tools including calibration and offline data analysis utilities, and experimental modes have been expanded to include drip-feed analysis and a distributed microsensor network in a star-point topology.

The detection algorithm used to identify the occurrence of events has been described previously.[3, 4] A detailed description with implementation approaches is provided in a recent NRL report.[6] The algorithm first populates background windows with the time duration required dependent on sampling increment (total number of points, rather than a time interval). With data collected at the 30 s increment used here, it is necessary to have 120 points for a stable initial condition (Background); 20 additional points fill the detection windows (Active and Snap). The 120 point Background window is intended to provide a smooth, slowly changing slope. This should capture any device drift over time as well as any changes resulting from diurnal and environmental changes. The Active window (20 points) provides a faster changing slope that will respond to chemical presence, while the shorter Snap window (10 points) is used to capture large, rapid changes. Comparing the Active and Snap windows to the slowly changing Background window provides the discrimination needed for identification of an event. Here, the conditions for ending positive event identification were changed based on recent evaluations.[5] The global cool down was changed from 60 min (120 points) to 10 min (20 points), and the buffering period for the global event was changed from 5 min (10 points) to 1 min (2 points).

DATA OUTPUT

The 15-element ABEAM device can be used singly or as a network containing up to six devices controlled by a laptop computer. When used singly, these devices can be monitored in real-time with drip-feed analysis, with live data transfer to the control computer without analysis, or autonomously with the data downloaded for analysis after collection. The following sections provide information on data formats and output files. Here, we focus on the data produced rather than on how to use the prototype devices. Operating guides are provided as Appendices to this document (refer to Appendices A, B, C).

Single device.

In an autonomous run, the device parameters are fixed using the custom software interface (Figure 4) and the device is started. It is then disconnected from the control computer. This device will continue to collect data until it is reconnected to the control computer and ordered to stop, the onboard memory is completely filled, or there is a power interruption. For either of the latter two conditions, the data in onboard memory is retained and can be recovered by connecting to the control computer. When the device is ordered

to stop, the data can then be downloaded. This approach allows an area to be monitored over durations of up to two weeks with the data analyzed offline at a later time. Live and Live/Dripfeed data collection include real-time transfer of the data to the control computer with real-time analysis included in the Live/Dripfeed version.

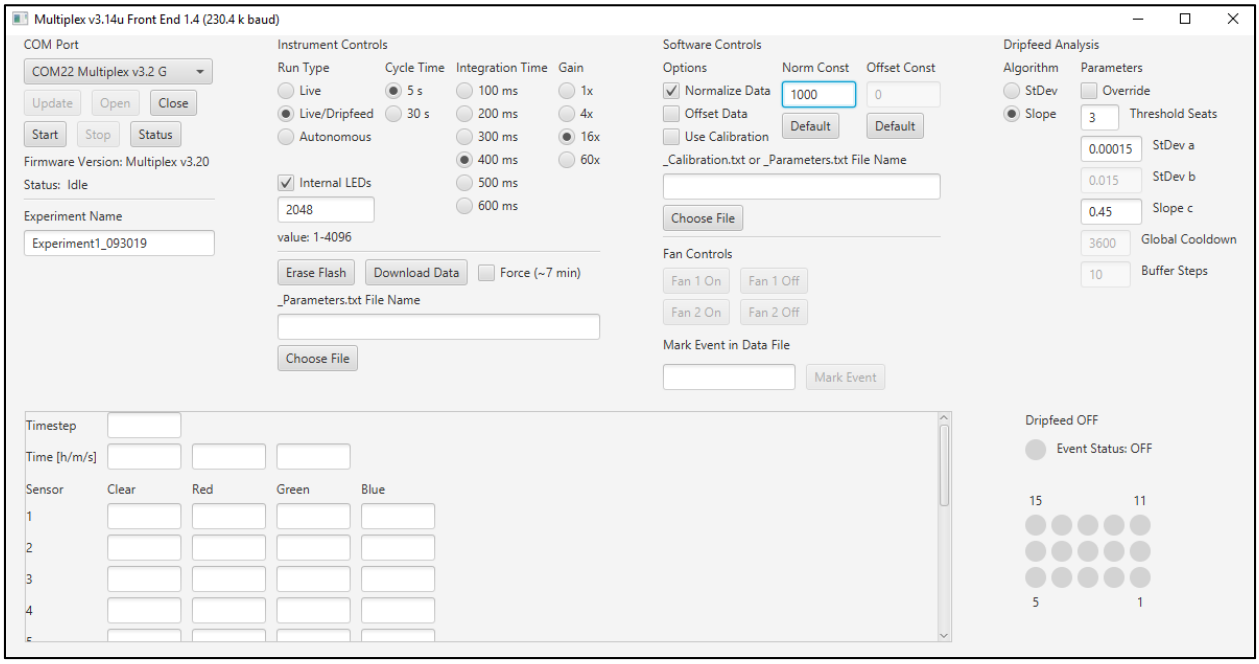


Fig. 4 — Screenshot for the control interface used for single device, autonomous, live, and live/dripfeed data collection.

For each set of data, several text files are generated: _Parameters, _Metadata, _Tagged_Actions, _Raw, and _Data. Tables 1 through 4 provide examples of the contents of the files with descriptions of the values. An example of the _Data file is not included here; when no normalization is used, this file is identical to the _Raw file. In Figure 5, a graphical representation for collected data is provided. Here, the data from the _Raw file has been normalized to the average of the first 120 points. Each element of the array (15 total) provides a stream of red, green, and blue reflectance values on a 30 s sampling increment.

Table 1. _Parameters file for autonomous, single device.

Description	**Output by Line**
Type of Illumination	1
LED Intensity	3072
Integration Time	3
Gain	0
Sampling Increment	1
Normalization Constant (-1 indicates none)	-1
Calibration Constants (nine total)	1.00
	1.00
	1.00
	1.00
	1.00
	1.00
	1.00
	1.00
	1.00
Offset Constant (-1 indicates none)	-1

Table 2. _Metadata file for autonomous, single device.

Description	**Output by Line**
	Multiplex v3.10u Data
	Specific device ID: Multiplex v3.2 G
	Firmware version 3.20
	Software version 1.4
Unit definition	Time data is: timesteps, seconds. These times are equivalent.
	Timesteps and actual times are measured in firmware.
Data organization	Data order is: clear, red, green, blue
Time device was started	Autonomous run started on 03 Oct 2019 at 06:44:08 PM [America/New_York]
Defined integration time	Integration Time: 400 ms
Defined gain	Gain: 1x
Defined sampling increment	Cycle Time: 30 s
Type of illumination	Lighting: internal LEDs
Intensity setting	Intensity: 3072
Use of normalization	Data is not normalized
Use of offset	Data is not offset
Use of calibration	No calibration file used

Table 3. _Tagged_Actions file for autonomous, single device.

Description	**Output by Line**
	Tagged Actions
	Format is event: seconds.
	All times measured in software only.
First manually stamped event	Event xxx: 153, 4558
Second manually stamped event	Event xxx: 265, 7916

Table 4. _Raw file for autonomous, single device.

	Timestep, Time, clear1, red1, green1, blue1, clear2, …, …, blue14, clear15, red15, green15, blue15
Line 1	0, 0, 981, 394, 320, 287, 1489, 608, 506, 427, 1246, 471, 437, 347, 804, 315, 279, 188, 203, 120, 47, 48, 384, 227, 94, 92, 3725, 1419, 1208, 1030, 3363, 1374, 1038, 892, 2546, 943, 941, 749, 857, 332, 324, 219, 466, 185, 167, 117, 490, 292, 107, 112, 2112, 828, 669, 592, 1790, 722, 583, 488, 850, 307, 304, 236
Line 2	1, 0, 981, 394, 320, 287, 1489, 608, 505, 427, 1246, 471, 437, 347, 804, 315, 279, 188, 203, 120, 47, 48, 384, 227, 94, 92, 3725, 1419, 1208, 1030, 3362, 1374, 1038, 892, 2546, 943, 941, 749, 857, 332, 324, 219, 466, 185, 167, 117, 490, 292, 107, 112, 2111, 827, 669, 592, 1790, 722, 583, 488, 850, 307, 303, 236
Line 3	2, 0, 982, 394, 320, 287, 1489, 608, 505, 427, 1248, 471, 437, 348, 806, 315, 279, 188, 203, 120, 47, 48, 384, 227, 94, 92, 3724, 1418, 1207, 1030, 3357, 1372, 1036, 891, 2545, 943, 940, 749, 858, 332, 324, 219, 465, 185, 167, 117, 491, 292, 107, 112, 2111, 827, 669, 592, 1793, 723, 584, 489, 850, 307, 303, 236
…	
Line x	The total number of lines depends on the length of the run and the sampling increment

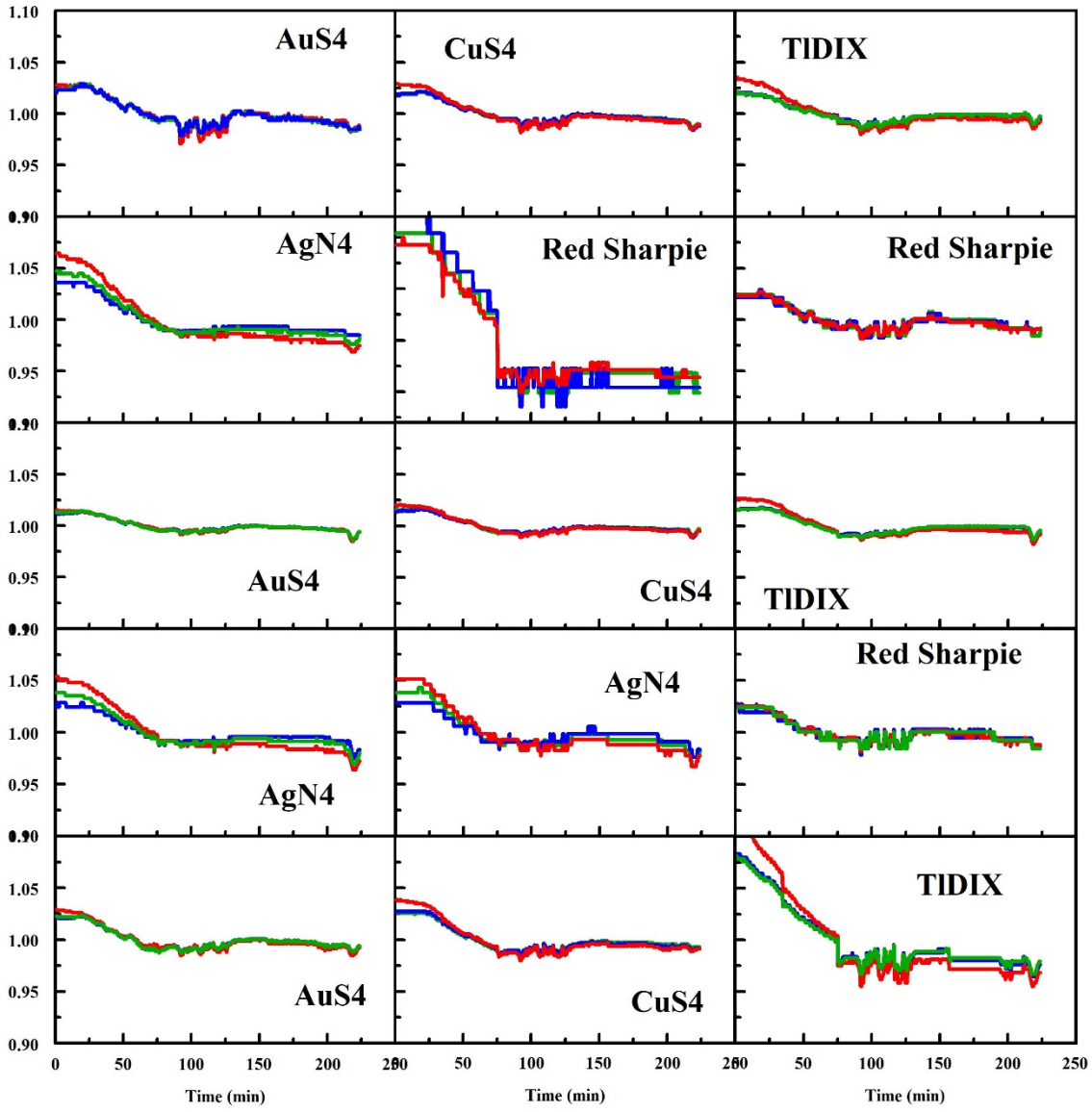


Fig. 5 —Normalized reflectance for the 15 element array. All data normalized to first 120 points collected.

In addition to this data stream, data analysis is provided in one of two ways. In the Live/Dripfeed usage, analysis is completed in real-time. In this case, indicators are provided on the control computer interface (Figure 6). Red indicates an active event; yellow indicates the cool down window is in progress; gray indicates negative detection status. Two files are also generated: _slp and _slp_detail. The _slp file provides a running list of the detected events (Table 5) while _slp_detail provides this list and includes the details of which indicator elements are involved in the events at what timesteps (Table 6). If offline data analysis is utilized (Appendix Z) for Live or Autonomous data, the _slp and _slp_detail files are generated at that time.

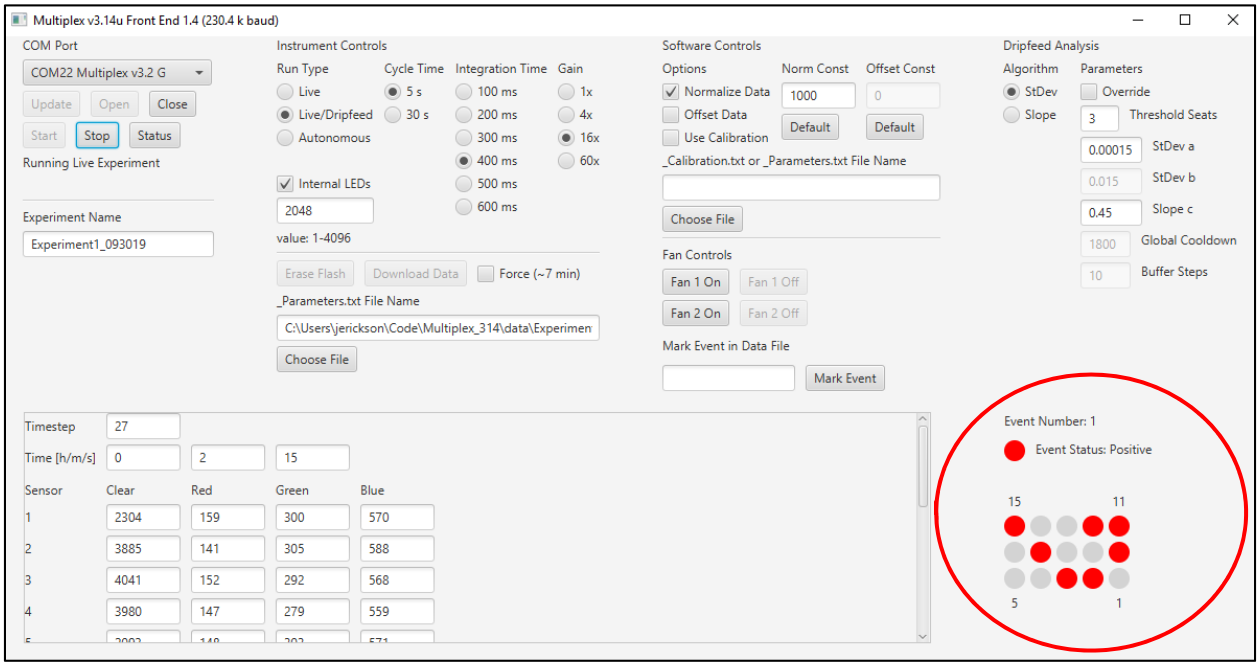


Fig. 6 — Control interface showing ongoing detection event.

Table 5. Event report provided by _slp file.

Description	**Output by Line**
Type of analysis used	Slope Data Analysis
Device specification	15 instrument seats numbered 1 - 15
User selected threshold minimum	Minimum Threshold Seats: 2
Sampling increment	Timestep: 30
Default detection parameter	Slope Threshold parameter: 0.45 degrees
Active event cool down window	Global Cooldown [s]: 600
	Buffer Zone [timesteps]: 10
Beginning of event 1 and involved elements	New Event: 1 at 4708 s. Initial Positive Seat(s): 9 10
End of event 1 and all elements involved	Event 1 ended at 6087 s. All Positive Seat(s): 1 3 5 6 9 10 12 13 15
Beginning of event 2 and involved elements	New Event: 2 at 8066 s. Initial Positive Seat(s): 1 2 12 14 15
End of event 2 and all elements involved	Event 2 ended at 10855 s. All Positive Seat(s): 1 2 3 4 5 6 7 8 10 11 12 13 14 15
	…

Table 6. Event report provided by _slp_detail file.

Description	**Output by Line**
Type of analysis used	Slope Data Analysis
Device specification	15 instrument seats numbered 1 - 15
User selected threshold minimum	Minimum Threshold Seats: 2
Sampling increment	Timestep: 30
Default detection parameter	Slope Threshold parameter: 0.45 degrees
Active event cool down window	Global Cooldown [s]: 600
	Buffer Zone [timesteps]: 10
Beginning of event 1	New Event: 1 at 4708 s.
Elements leading to triggering of event	Positive Seat(s): 9 10
Involved elements at next change of state	Added new Positive Seat (s) at 4768 s: 5 9 10
Involved elements at next change of state	Added new Positive Seat (s) at 4798 s: 5 6 9 10 12 15
Involved elements at next change of state	Added new Positive Seat (s) at 4828 s: 1 5 6 9 12 15
All elements reporting non-event	Event 1 ended at 4978 s
New detection within the cool down window	Event 1 extended at 5427 s
Elements leading to triggering of event	Added new Positive Seat(s): 3 13
Involved elements at next change of state	Added new Positive Seat (s) at 5787 s: 3 13
Involved elements at next change of state	Added new Positive Seat (s) at 5847 s: 3 13
Involved elements at next change of state	Added new Positive Seat (s) at 5997 s: 10 13
End of event	Event 1 ended at 6087 s.
Beginning of event 2	New Event: 2 at 8066 s.
Elements leading to triggering of event	Positive Seat(s): 1 2 12 14 15
	…

Device Network.

In networked use scenario, several devices (for examples provided here, we use six) are controlled by a single computer. Communication is established between the controlling computer and each device via onboard wireless one at a time, but parameters are fixed simultaneously with all devices using the same parameters. The custom software interface is used to begin data collection by all devices with the start time for each device associated with the sequence used to establish communication with the controlling software (Figure 7). The devices collect data until ordered to stop. As in the single device use case, interruptions, for example communications or power, will result in cessation of the real-time reporting, but the data in onboard memory is retained and can be recovered by reconnecting to the control computer.

For each set of data, several text files are generated: _Network_Data (Table 7), _Parameters (as in Table 1), and _Tagged_Actions (as in Table 3). In addition, a folder is created for each of the devices in the network. These folders contain _Metadata (as in Table 2), _Raw (as in Table 4), and _Data (again omitted here) files as well as a subfolder, ‘Dripfeed_Data’. The subfolders contain _slp and _slp_detail files as described above (Tables 5 and 6).

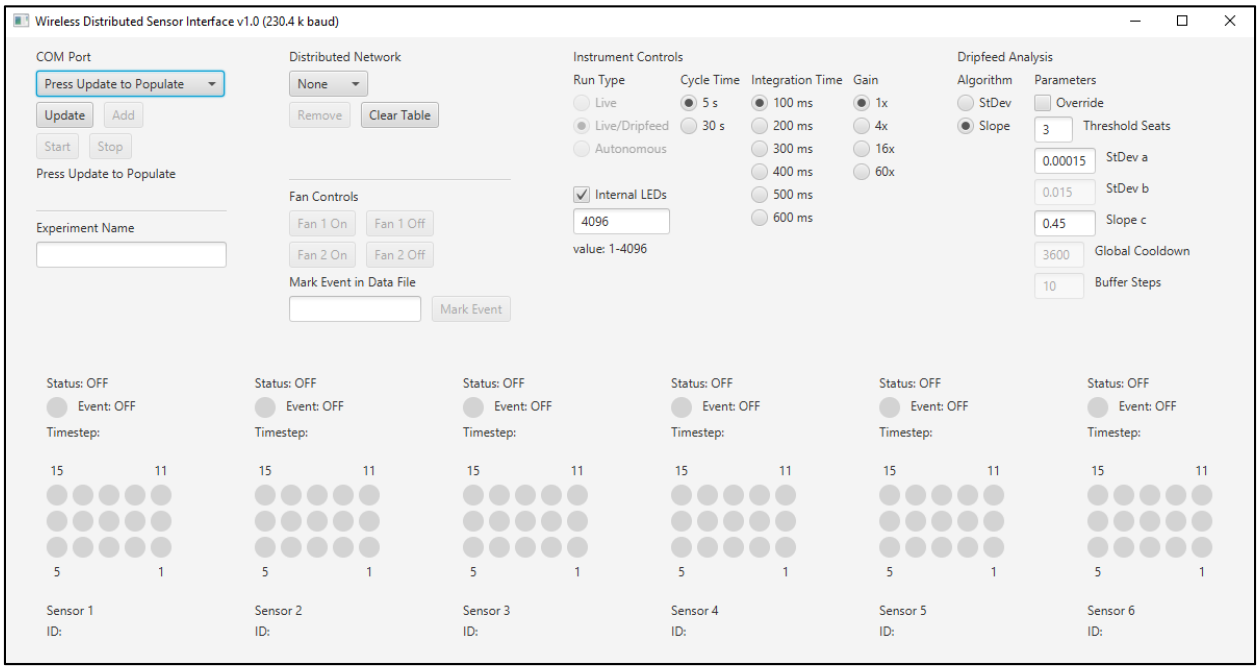


Fig. 7 — Screenshot for the control interface used for networked devices with live/dripfeed data collection.

Table 7. _Network_Data file for multiple device use scenario.

Description	**Output by Line**
	Distributed Microsensor Experiment Data
	Global Level Information
Total number of devices used	Number of sensors in network: 6
Individual device identification for each element in the network	Specific device IDs:
	Sensor 1: Multiplex v3.1 AW
	Sensor 2: Multiplex v3.1 BW
	Sensor 3: Multiplex v3.1 CW
	Sensor 4: Multiplex v3.2 DW
	Sensor 5: Multiplex v3.1 EW
	Sensor 6: Multiplex v3.1 FW
Software iteration	Software version 1.4
Start time stamp for data collection	Network run started on 04 Oct 2019 at 10:24:24 AM [America/New_York]
Parameters fixed within the user interface	All instrument parameters are identical. They are:
	Integration Time: 400 ms
	Gain: 1x
	Cycle Time: 30 s
	Lighting: internal LEDs
	Intensity: 3072
	Data is not normalized
	Data is not offset
	No calibration file used
Stop time stamp for data collection	Network run stopped at 04 Oct 2019 at 02:08:22 PM [America/New_York]

Data for each of the devices in the networked scenario is processed independently while the output is displayed within the single interface. Report of an event by Sensor 1 has no impact on Sensor 2 in the current implementation. Data analysis, forced data downloads, etc can be completed offline using

approaches as described above for single devices. As in the case of the single device, Live/Dripfeed analysis is completed in real-time. In this case, indicators are provided on the control computer interface (Figure 8). Red indicates an active event; yellow indicates the cool down window is in progress; gray indicates negative detection status. The data reported for each device will reflect that presented in Figure 5.

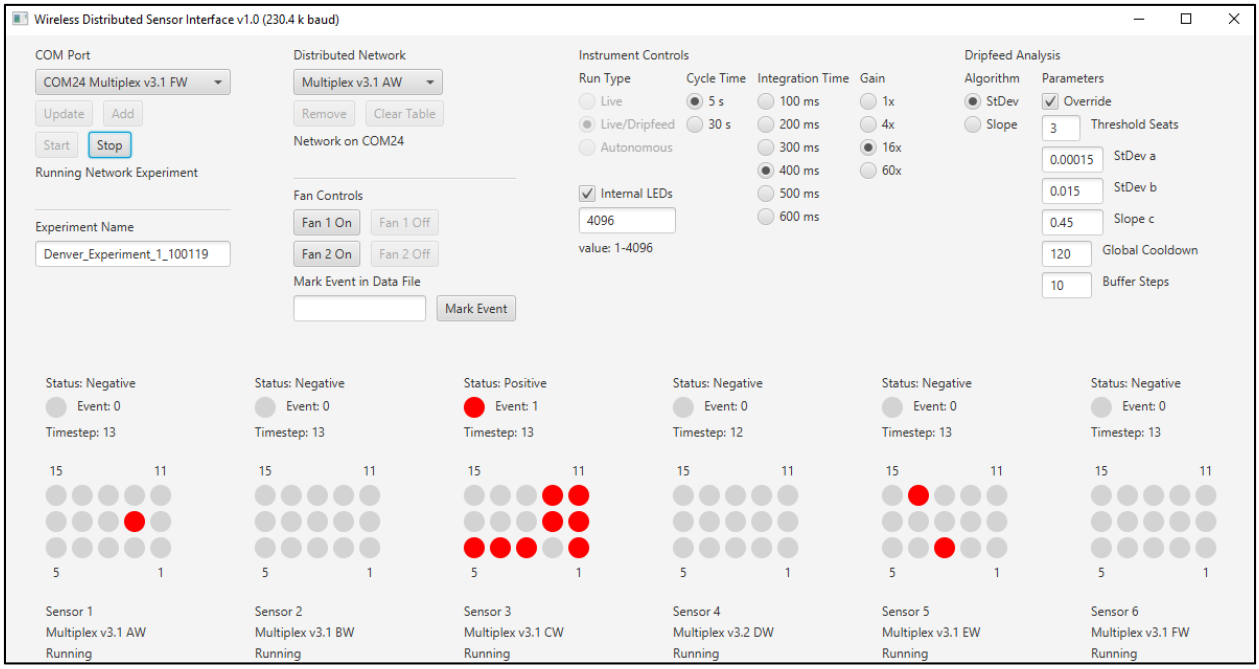


Fig. 8 — Control interface showing ongoing detection event. Here, a three element minimum response threshold is used. Sensors 1 and 5, therefore, are in an overall negative state by failing to meet this requirement.

Offline Data Analysis.

Analysis of data files subsequent to cessation of data collection provides the opportunity to process data following a communications failure. This would be the data downloaded from the device flash memory upon reconnection or that collected during Autonomous or Live device use scenarios. Alternatively, it provides an opportunity to process the data using different requirements for detection, for example, with higher or lower sensitivities, different numbers of element minimums, etc (Figure 10). The output from this process is a subfolder 'Offline_Analysis' that contains _slp and _slp_detail files (Tables 5 and 6).

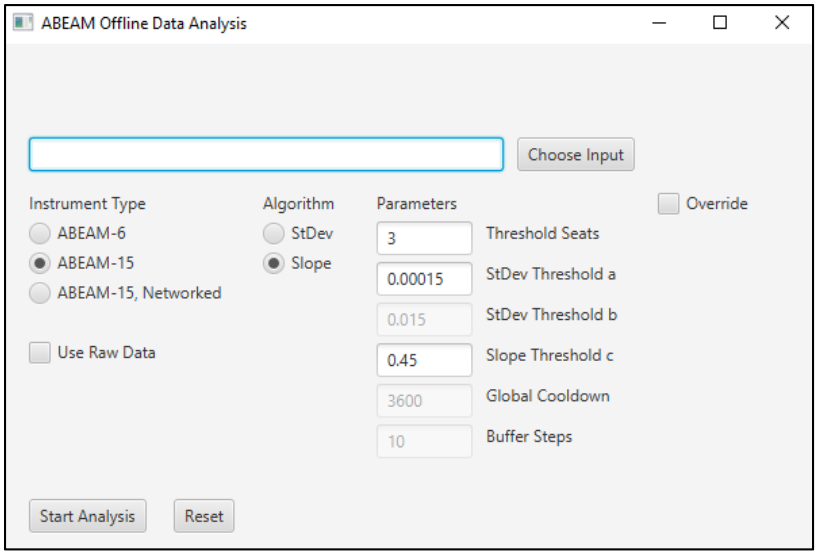


Fig. 9 — Interface for post data collection, offline analysis.

CONCLUSIONS

The discussion presented here is intended to offer a quick reference to the data output and supporting files provided under the various use scenarios for the v3.2 prototype reflectance devices. The files resulting from use of these prototypes vary significantly from those of the original reports on the six element prototypes.[1, 5, 6] The ongoing effort continues to screen additional indicators and targets as well as to address development of the target identification algorithms.

REFERENCES

1. Johnson, B.J., Erickson, J.S., Kim, J., Malanoski, A.P., Leska, I.A., Monk, S.M., Edwards, D.J., Young, T.N., Verbarg, J., Bovais, C., Russell, R.D., and Stenger, D.A., "Miniaturized reflectance devices for chemical sensing," *Measurement Science and Technology*, **2014**, 25, 095101.
2. Johnson, B.J., Liu, R., Neblett, R.C., Malanoski, A.P., Xu, M., Erickson, J.S., Zang, L., Stenger, D.A., and Moore, M.H., "Reflectance-based detection of oxidizers in ambient air," *Sensors and Actuators B-Chemical*, **2016**, 227, 399-402.
3. Malanoski, A.P., Johnson, B.J., Erickson, J.S., and Stenger, D.A., "Development of a Detection Algorithm for Use with Reflectance-Based, Real-Time Chemical Sensing," *Sensors*, **2016**, 16.
4. Johnson, B.J., Malanoski, A.P., Erickson, J.S., Liu, R., Remenapp, A.R., Stenger, D.A., and Moore, M.H., "Reflectance-based detection for long term environmental monitoring," *Heliyon*, **2017**, 3, e00312.
5. White, B.J., Erickson, J.S., Malanoski, A.P., and Moore, M.H., "Reflectance-Based Sensing: Post-Evaluation Analysis of Sensor Responses," US Naval Research Laboratory, Washington, DC, **2019**; NRL/MR/6930-19,9852.
6. Erickson, J.S., Malanoski, A.P., White, B.J., Stenger, D.A., and Tankard, E.R., "Practical Implementation of Detection Algorithm for Reflectance-Based, Real-Time Sensing," US Naval Research Laboratory, Washington, DC, **2018**; NRL/MR/6930-18,9812.

Appendix A

ABEAM OPERATING INSTRUCTIONS: SINGLE DEVICE, AUTONOMOUS, LIVE, AND LIVE/DRIPFEED

DEVICE CONSIDERATIONS

Housing options: The ABEAM-15 has two separate housing options: an open housing, and an outdoor housing. The open housing is designed for indoor use and for the detection of non-corrosive compounds. It provides access to the power and USB connectors, as well as the wireless module. Reset pins are easily available. The outdoor housing is designed to completely enclose the instrument electronics. As such, it is specifically designed for use with batteries and wireless communications only. The sealed nature of the housing provides protection from the elements and against corrosive compounds that might otherwise damage the instrument. It is not designed for submersion. In addition, the outdoor housing has a sun shield to prevent lighting variations from impacting measurements.

Power options: The ABEAM-15 can use either a 7.5 V DC power supply, or a 6S-NiMH rechargeable battery pack. The DC supply should have a 2.5 mm barrel jack with positive center polarity. The battery pack should be terminated with a standard male Tamiya connector. Configurations other than 6S are not recommended. ABEAM-15 hardware version 3.1 must be manually switched between the power supplies. This is accomplished by moving the position of a jumper.

ABEAM-15 hardware version 3.2 will automatically switch between the supplies. If a 7.5 V DC supply is plugged into an instrument with an attached battery, the instrument will automatically switch to the DC supply to conserve battery life. If the DC power supply is subsequently removed, the instrument will switch back to battery power.

Future hardware versions will incorporate built-in charging. At the moment, batteries must be externally charged.

Known Bug: Version 3.2 hardware will not cold start from a battery if a USB cable is connected and powered. You must first disconnect the USB cable, then wait a few seconds for the instrument capacitors to discharge.

Communication options: The ABEAM-15 can use either wireless or tethered (USB) communications. The USB requires a mini-B terminated cable. Wireless uses XBee modules, from Digi International. Series 3 modules are preferred, although series 2 modules are compatible. Due to speed, reliability, and power issues, all instrument hardware versions will automatically switch from wireless to tethered communications if a USB cable is plugged in. Note that in order to use XBee wireless, you will also need a coordinator, typically in the form of a dongle with a USB plug (not shown here). The coordinator is plugged into the computer.

USE MODES

The ABEAM-15 is a multiplexed, reflectance-based sensor for detection of chemical target vapors and aerosols. The full instrument package contains sensor hardware, a housing, and a software suite to control the instrument and for offline data analysis. The ABEAM-15 operates in three different modes:

1. Live Mode
2. Autonomous Mode
3. Network Mode

These three different modes are designed to provide utility in a wide variety of operational scenarios.

Live mode: In live mode, a single sensor is connected directly to a computer through either a USB cable or by wireless communications. Data generated by the sensor is transmitted directly from the instrument to the computer in real time. For each experiment, the software produces a folder containing files with raw data, run parameters, and metadata such as the hardware and firmware versions. Results are

available in real time through dripfeed data analysis, if desired. In addition, an offline data analysis can be performed using the analysis utility tool, which allows the user to optimize analysis parameters.

Live mode is useful for short term experiments, primarily laboratory experiments. Live mode provides the opportunity to obtain calibration data, to select indicators, or to optimize the instrument for use in a specific scenario. Live mode is also useful when the instrument will be used in conjunction with an unmanned vehicle, especially if the instrument can be directly controlled from the platform.

Autonomous mode: A sensor can be set up to run in autonomous mode. In this case, the instrument is initially connected to a computer in order to select run parameters and issue a 'start' command. Once the instrument has started running, the USB cable or wireless connection is removed, and the sensor can be transported to any desired location. The sensor will continue to run and collect data until either there is a power interruption, the internal (flash) memory is filled, or the user re-connects the control computer and issues a 'stop' command. In autonomous mode, a live stream of data is not available. In order to recover data, the user must re-connect the control computer and manually download the files. Once this has been completed, offline data analysis can be performed anytime using the analysis utility tool.

Autonomous mode is useful for long-term experiments, especially those completed outdoors. A large number of instruments can be set up, one at a time, in an outdoor location for perimeter monitoring or other applications. Experiments can be started and stopped at the point of collection using the control computer.

Network mode: Network mode combines features of both live and autonomous modes. Similar to autonomous mode, multiple sensors can be run simultaneously, and data is stored on the instrument in flash memory for later recovery if desired. Similar to live mode, data is also transferred to a computer in real time. Dripfeed data analysis is performed on each connected sensor as data comes in, making results available in real time. Offline data analysis can be performed at any time on the raw data files. If a network connection is broken, data can still be recovered through individual sensor downloads.

At present, the software suite allows network runs of up to six sensors. All sensors must be connected through wireless; USB is not an option for a network run. We anticipate that future software versions will relax this requirement and increase the number of sensors that can be networked in a single experiment.

Network mode is useful for setting up distributed microsensor networks with real-time feedback.

Live, Live/Dripfeed, Autonomous

The software is a file named "Multiplex_314u.jar". You can run the program by double-clicking on it. After doing this, you should see the startup screen, shown in Figure A1.

In order to run an experiment, the following steps should be performed:

1. Connect the ABEAM-15 USB cable to the computer, or connect the XBee wireless dongle
2. Start the software
3. Populate the COM port list and open the selected instrument
4. Choose a filename
5. Set the instrument parameters
6. Set the software parameters
7. Set the dripfeed analysis parameters (if using)
8. Start the experiment

Press Update to scan the COM ports for attached devices. In the example shown (Figure A2), two different instruments are attached to the computer; they are identified as G and H. Select the desired instrument and press Open (the button is hidden under the drop box).

Fig. A1. Startup screen.

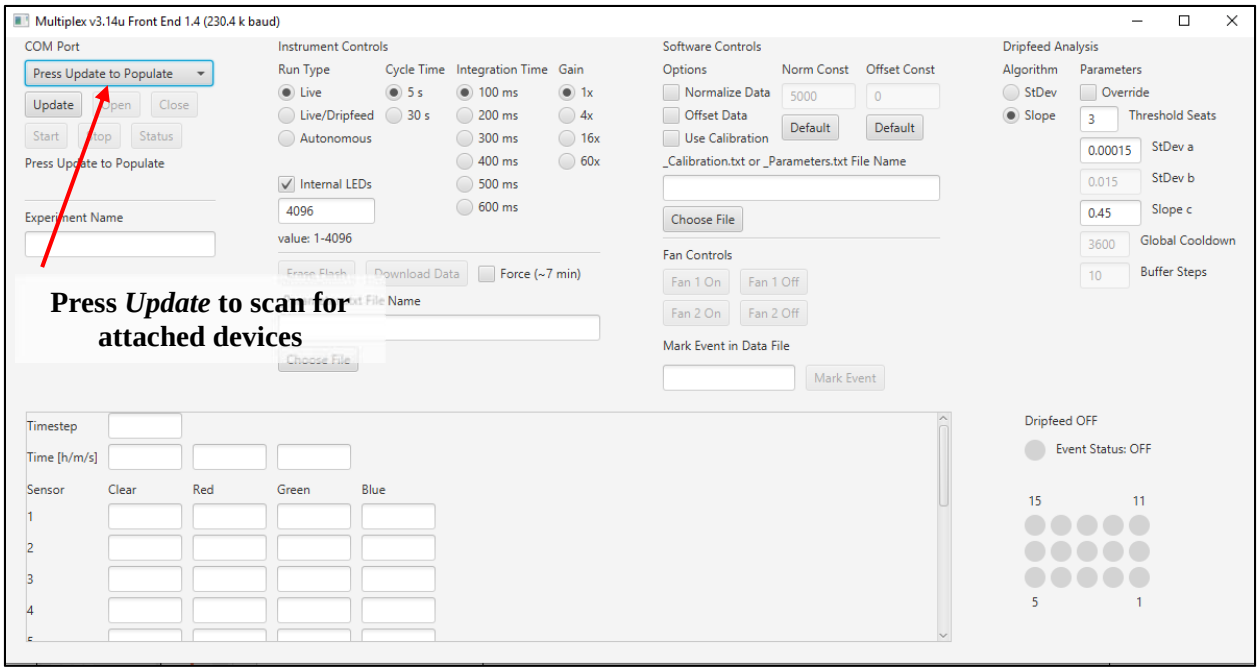
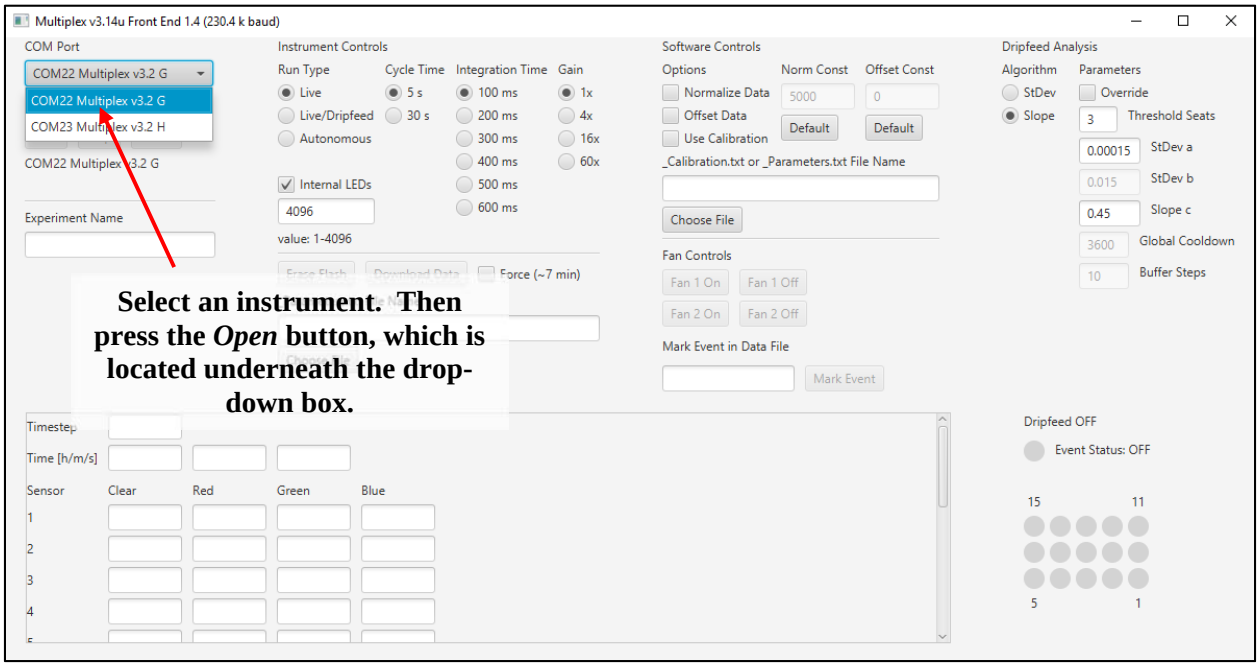
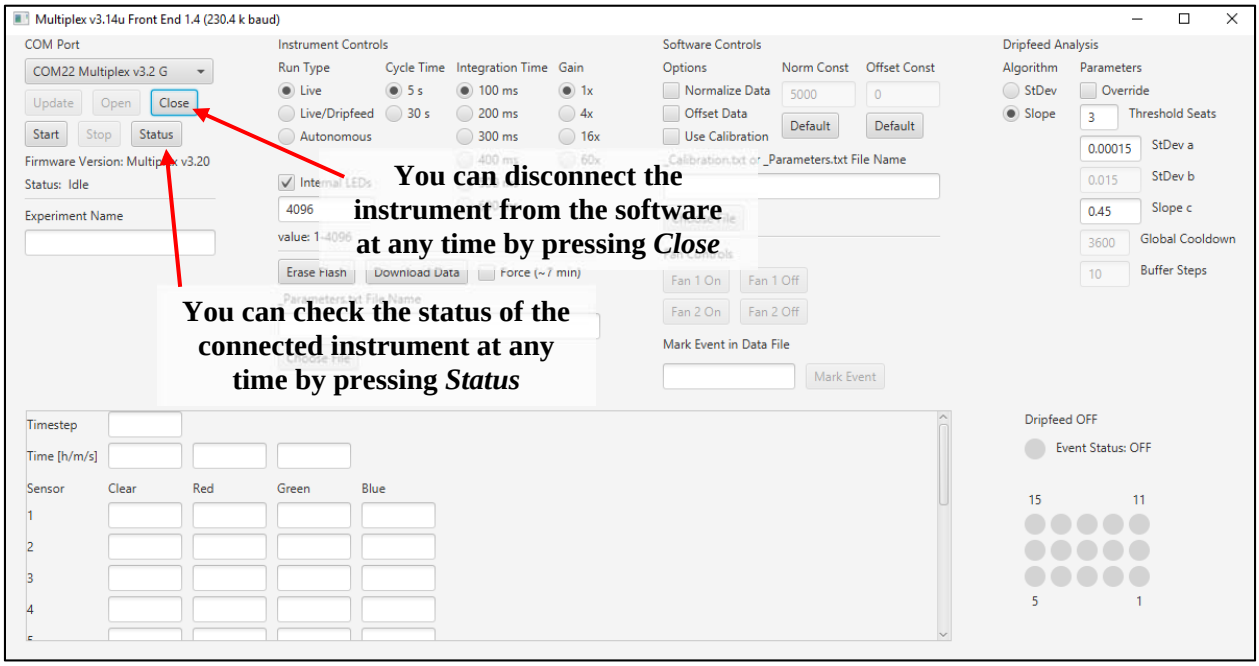


Fig. A2. Connect the software to the instrument.



After opening the instrument, the firmware version and instrument status will appear in the status box (Figure A3). You can check the status of the connected instrument at any time by pressing Status.

Fig. A3. Instrument status



Before starting an experiment, it is necessary to give it a unique name. Type it into the Output File box, under the heading “Experiment Name” (Figure 4). Do not add file extensions to the file name. This will be done automatically as needed.

A unique Experiment Name must be chosen before starting an experiment. When running an experiment, the software generates a folder, named Experiment Name, containing five different files. The folder and all of these files will have their names constructed from the Experiment Name. These five files are:

[Experiment Name]_Raw.txt Raw experimental data collected from the instrument. It is not modified in any way.

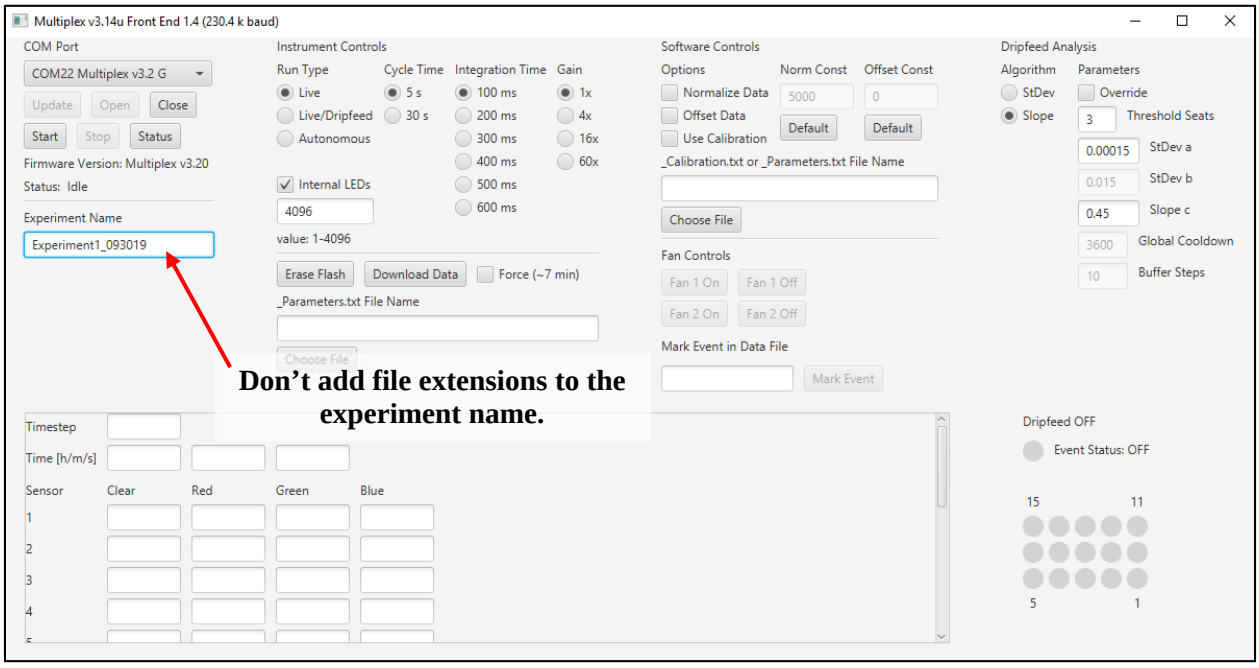
[Experiment Name]_Data.txt Similar to the _Raw file, the data file is experimental data collected by the instrument, but modified by any software parameters or calibration as selected by the user. This will be described later.

[Experiment Name]_Metadata.txt Contains information about the experiment including firmware and software version, timestamp, and values for the user selected instrument parameters. If a calibration file was used, the filename and its values are listed.

[Experiment Name]_Parameters.txt Similar to the metadata file, it contains the user selected instrument parameters. The difference is that the _Metadata file is in a text format intended to be read by the user. The _Parameters file is in a numerical format meant to be used as an input to the software.

[Experiment Name]_Tagged_Actions.txt Any user tagged actions and the time at which they were tagged are listed here. Tagging the experiment in real time will be described later.

Fig. A4. Choose experiment name



Instrument parameters (Figure A5) control the way data is collected at the hardware level, such as integration time. The effect of any instrument parameters will be reflected in both the _Raw and _Data files. Before starting an experiment, it is necessary to set the instrument parameters. They are: run type, cycle time, integration time, gain, LED, and LED intensity. Once an experiment has started, these parameters may not be changed.

Run Type: Live or Autonomous. In a live run, the instrument must remain tethered to the computer and connected to the software. Results are updated at the end of each measurement cycle. In contrast, after starting an autonomous run, the software can be disconnected and the instrument detached from the computer. Results can only be downloaded after the experiment is complete.

Cycle Time: This is the amount of time between each successive measurement. One measurement is a datapoint from all 15 sensors.

Integration Time: Choices are 100 – 600 ms. Each sensor will collect light over a finite period of time, the integration time, before reporting the result. The longer the collection period, the higher the signal.

Gain: Choices are 1x, 4x, 16x, and 64x. This is a digital gain; it simply multiples the result by the selected constant after the integration has finished.

LED: Select this box to use the instrument's eight internal LEDs. Un-check the box if you wish to use your own light source, or none at all.

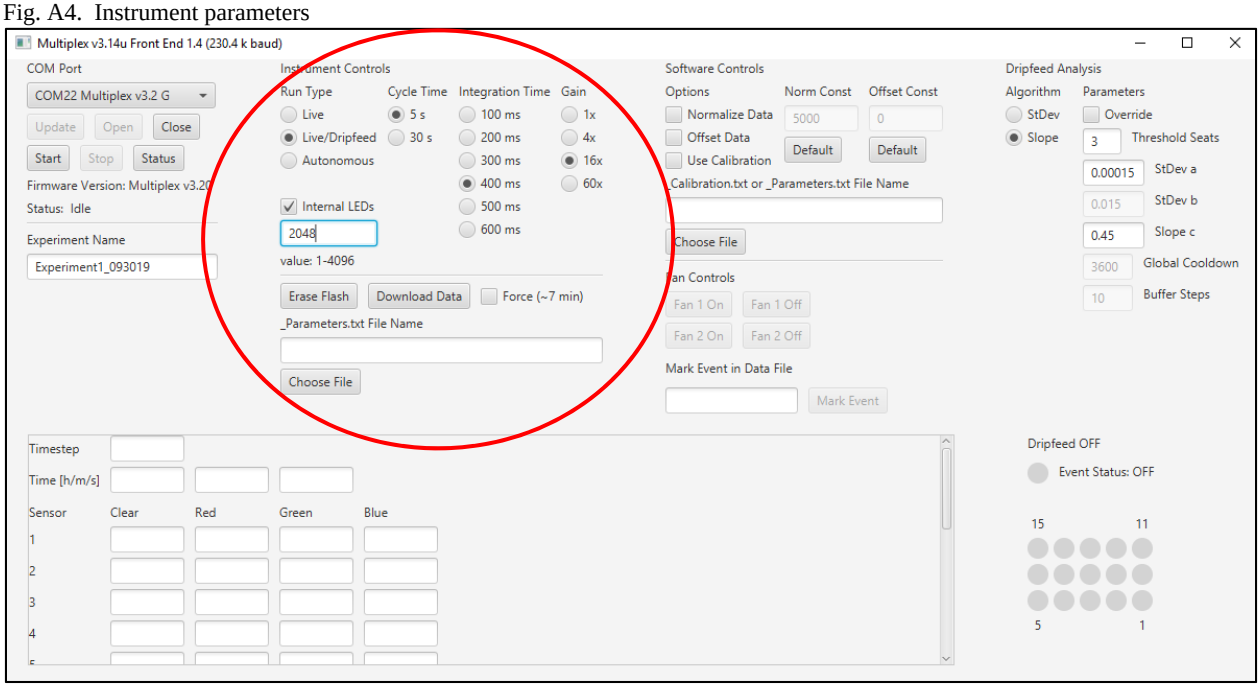
LED Intensity: If the internal LEDs are used, this value specifies their intensity. 1 is the minimum value, and 4096 is full. Response is roughly linear.

Advice: sensor data is reported as a 16-bit number (0 – 65,535). Making full use of this range may be advantageous. Then darkness would be close to 0, while complete saturation would be close to 65,535. For fast sensor response, it is good to minimize integration time. Therefore, increase the gain first to expand the instrument range. Only when gain is maximum should integration time be increased. Note that for

technical reasons, integration times of less than 154 ms will provide less than the full 16-bit range. High LED intensity (especially coupled with long integration times) can cause photobleaching, but, in general, LED intensity should be set as large as possible.

During a live experiment, the ABEAM-15 sends its data to software as soon as it is collected. During an autonomous experiment, this data is saved to internal flash memory and retrieved at a later time. Before an experiment can be run, the firmware must first erase the flash. This process can take 30 seconds or more. In many cases, this is not a problem; however, if the operator requires an "instant start" or needs to know the exact time at which the ABEAM-15 starts collecting data, this erase cycle can be completed in advance by selecting the Erase Flash button, before starting an experiment.

Fig. A4. Instrument parameters



Software parameters control modifications to the data after it is collected and downloaded (Figure A5). The effect of any software parameters will be reflected in only the _Data file. The _Raw file captures the same data, but without any of the software parameters applied. It is not necessary to select any of the software parameters before starting an experiment; however, these parameters may not be changed during an experiment. Live data will be displayed with modifications specified by the software parameters, and software modified data is recorded in the _data.txt file. Raw data (before modification by any software parameters) is always available to the user. It is stored in a separate location, in the _raw.txt file.

Software data modification is a three-step process. All of these processes are optional and any of them can be skipped. Those processes that are selected will always be performed in order: normalization, calibration, offset.

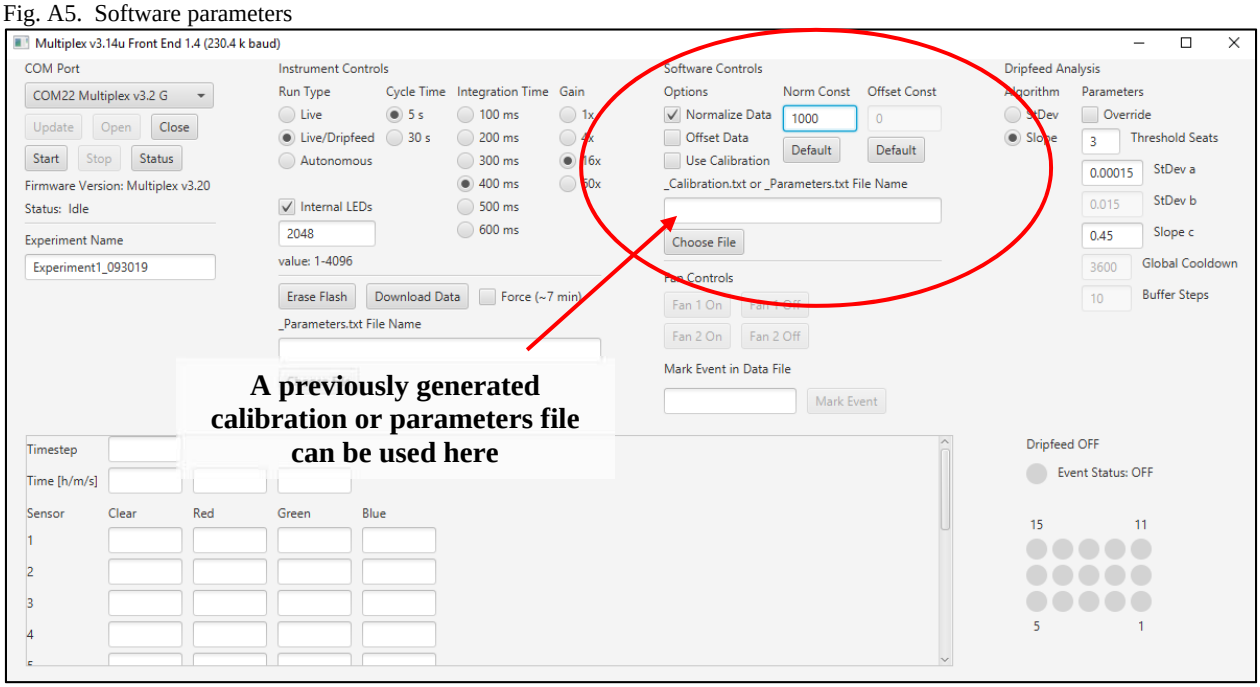
Normalize Data: This attempts to address incident light non-uniformity and drift. If Normalize Data is checked, a value for Normalize Constant must be specified (a value of 1 is acceptable). For each sensor, the R, G, and B values will be divided by the C value. This addresses the fact that different sensors may be illuminated at different intensities. Then, the data will be multiplied by the value of Normalize Constant, which sets the magnitude of the baseline. After normalization, it is possible to get fractional (but not negative) data.

Calibration: Calibration allows the user to experimentally measure color standards with the sensor and set their default values in a calibration matrix. If Use Calibration is checked, a file must be provided with

these calibration constants. After calibration, it is possible to get negative data. It is also possible to expand (stretch) the range of data beyond 16 bits. To create a calibration, a different software program is required. The calibration file ALSO stores hardware parameters from the instrument control menu. When Use Calibration is checked, these parameters will also be implemented. The calibration matrix is currently an average over ALL sensors.

Offset: This attempts to deal with negative data that may result from using a calibration. If Offset Data is checked, it transforms the entire dataset by adding a constant value (the Offset Constant). The value of Offset Constant must be between 0 and 65,535.

Fig. A5. Software parameters



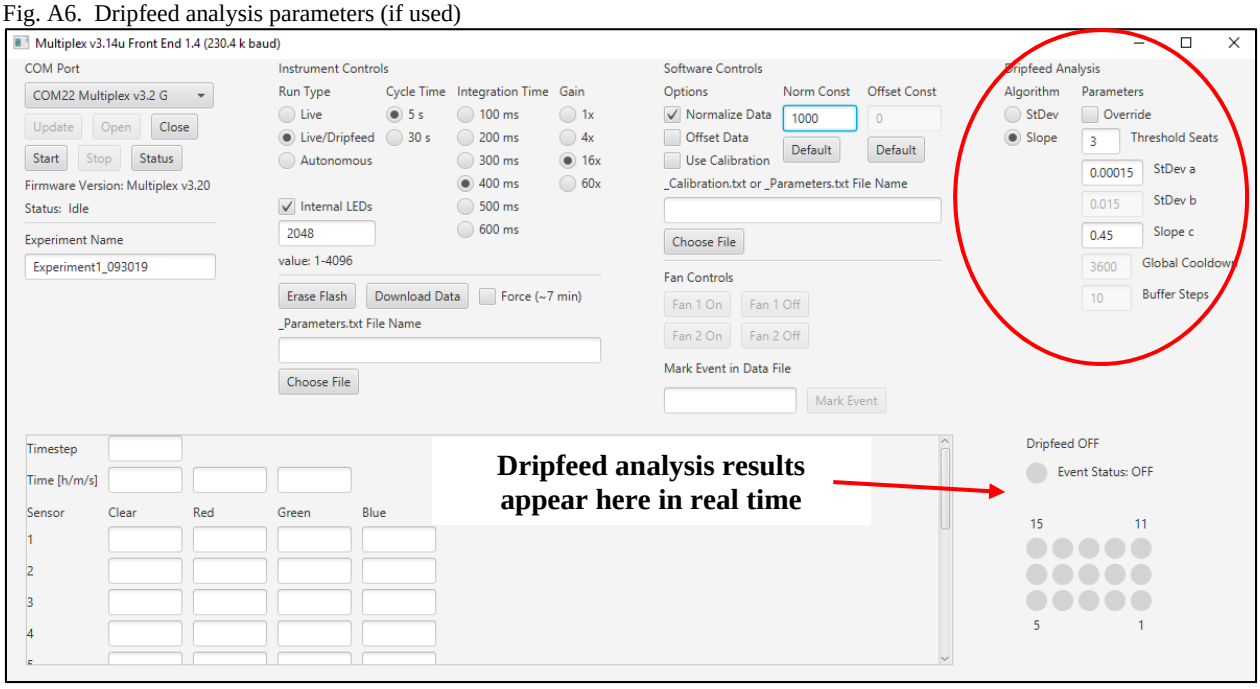
A file can be used to set the instrument and software parameters. As noted above, when Use Calibration is checked, it implements both the calibration matrix and (hardware) instrument parameters stored in the file.

There are three things to note:

(1) It is possible to duplicate all of the software and hardware parameters from a previous run by checking Use Calibration and simply using the _parameters.txt file as input. Either a calibration file ("Calibration.txt") or a _parameters.txt file can be used here. Both can be read by the software.
(2) A calibration file with values of 1 in every entry in its matrix will have no effect on the data. When the _parameters.txt file is generated by the software, if no calibration file was used, it does exactly this: the matrix values are all populated with 1's.
(3) This allows the user to regenerate all of the conditions used in a previous experiment by simply linking to the old parameter file that was used to collect that particular data set.

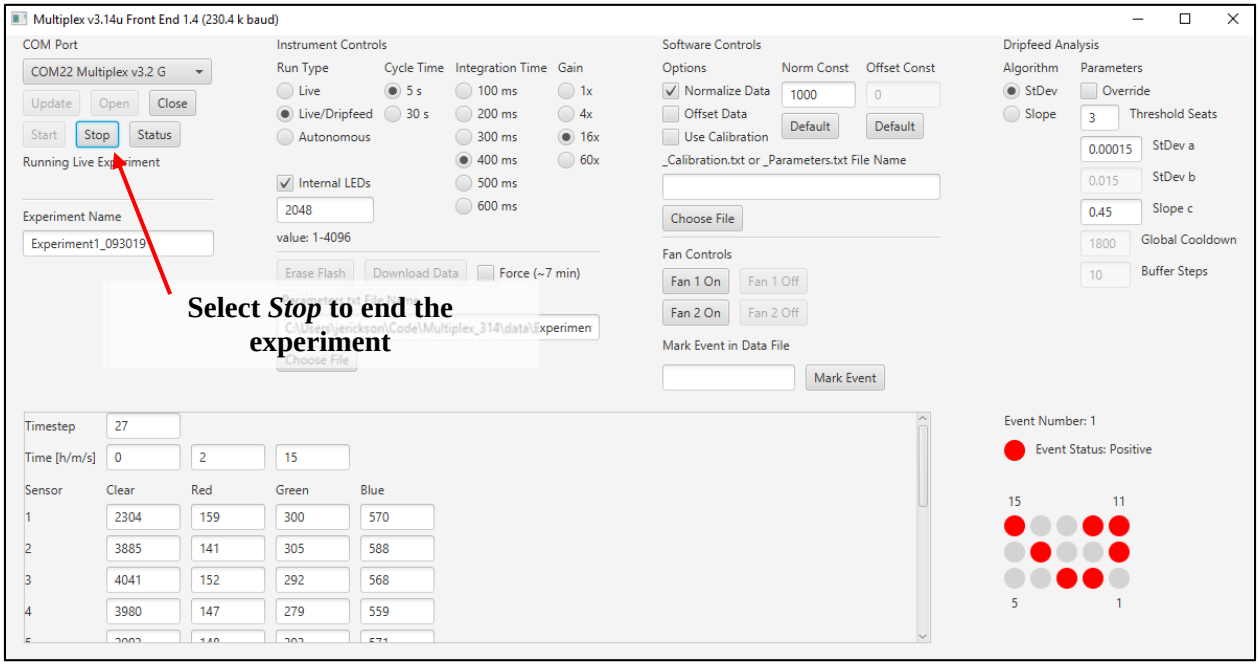
In order to use real-time, dripfeed analysis, you must select Live/Dripfeed from the instrument controls (Figure A6). These parameters affect the data analysis and are fully described in an NRL Report (NRL/MR/6930-18-9812; this report is available at Distribution A) Selecting "Override" will allow you to set the values that are normally greyed out.

Fig. A6. Dripfeed analysis parameters (if used)



To stop a device, press the Stop button (Figure A7). For autonomous use, you must first tether the instrument to the computer and connect the software.

Fig. A7. Stopping an experiment



Specifically for autonomous experiments: When a live device run is started, five different files are created (as described in a previous slide). In the case of an autonomous experiment, only three are created immediately: _Metadata.txt, _Parameters.txt, and _Tagged_Actions.txt. The other two files are created during the download process itself: _Raw.txt and _Data.txt. In order to properly process the incoming data, the software parameters must be known. Unlike the instrument parameters, they are not stored in the flash memory at start time. In theory, software parameters could be entered at the time of the download. However, there are two problems with this:

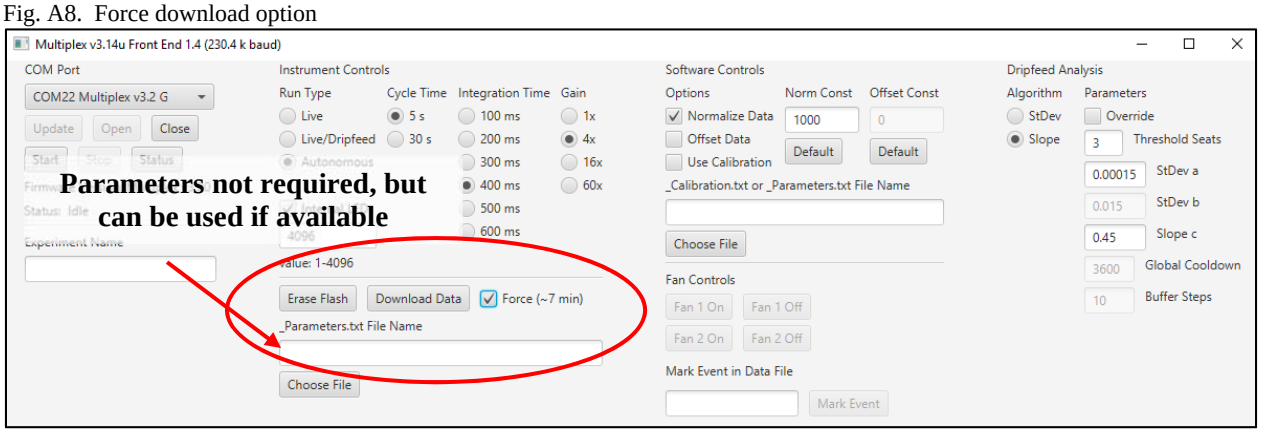
(1) It is important to record the exact conditions used to create the _Data.txt file.

(2) An autonomous experiment could have been started some time ago (days, weeks, or even months) by one operator and later retrieved by a second one. It is possible that the second operator does not remember or have access to the desired software parameters for that particular experiment.

Issue #1 could be addressed by generating the _Parameters.txt file at the time of download, rather than at the time the experiment was started. However, this does not address issue #2. For now, the software is set up to require a _Parameters.txt file to complete the download. This may be changed in the future by adding two radio buttons: "Use _Parameters file" and "Generate _Parameters file". (In case of lost files, there is an alternative described later)

If an instrument is not shut down in an orderly way during an experiment, there may be problems downloading the data. When this happens, the operator should check the Force button before pressing Download Data (Figure A8). This does two things: (1) It downloads the entire contents of the flash memory, whether there is data there or not. Blank data points will have a raw value of 65,535. (2) It allows the operator to download data from the instrument without the use of a _Parameters.txt file. In this case, only _Raw.txt will be created, and it will assume that no software modifications to the data are desired. If modifications are desired, they will need to be performed at a later time.

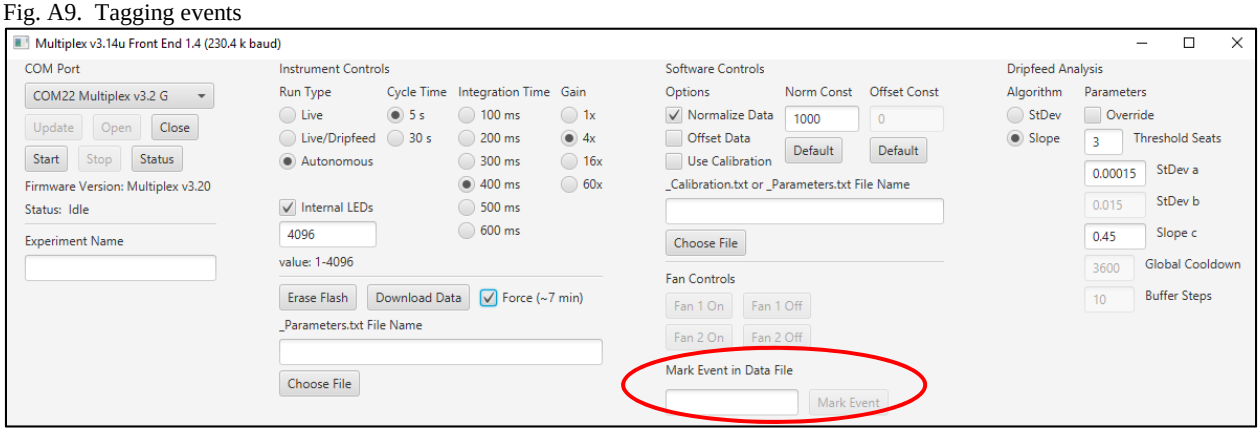
Fig. A8. Force download option



Time is measured in two different ways during an experiment. The hardware inside the ABEAM-15 times the measurement cycles. The experiment start and stop times are recorded using the software computer's clock. Both of these will appear in the output files. However, it is possible that neither of these agree with the operator's clock (a watch, cell phone, or similar). Tagging events can be used to synchronize these two clocks to the third (the operator's clock).

To tag an event, simply name the event and press the Mark Event button. The event will appear in the _Tagged_Events.txt file. For a live run, the event name is accompanied by the current datapoint (hardware clock) and timestamp (software clock). For an autonomous run, there is no datapoint information. This function can be used to effectively identify calibration peaks, to mark a synchronization point, or to measure instrument response time to an exposure.

Fig. A9. Tagging events

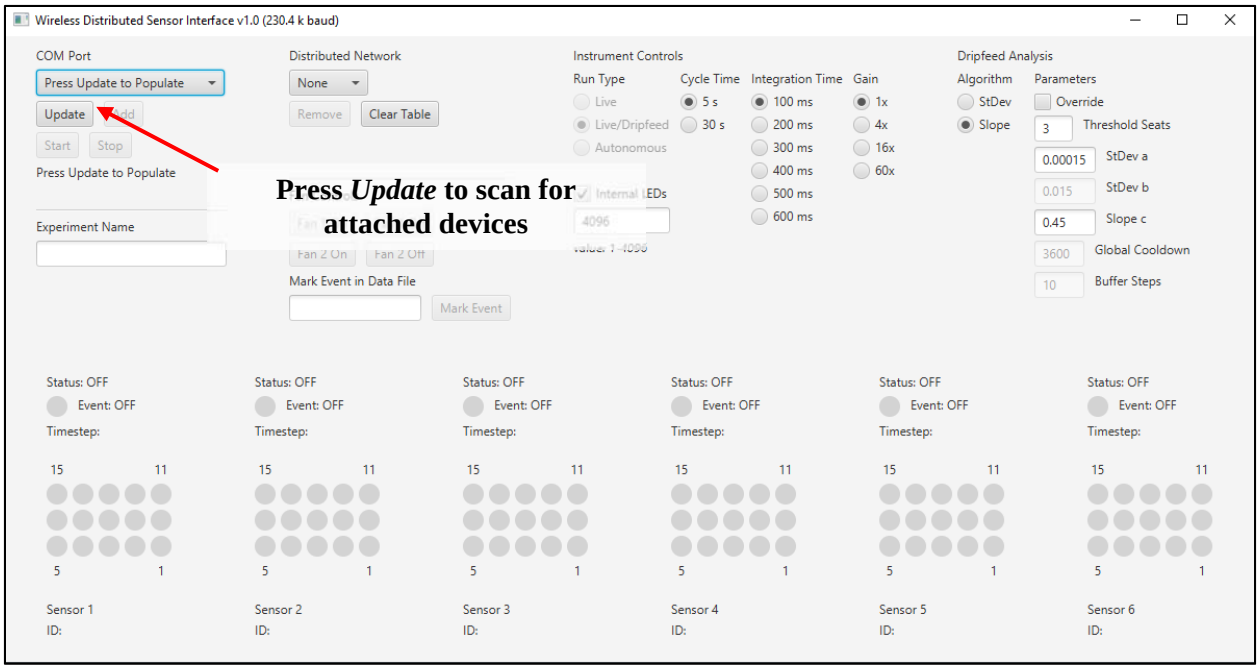


Appendix B

ABEAM OPERATING INSTRUCTIONS: MULTIPLE NETWORKED DEVICES

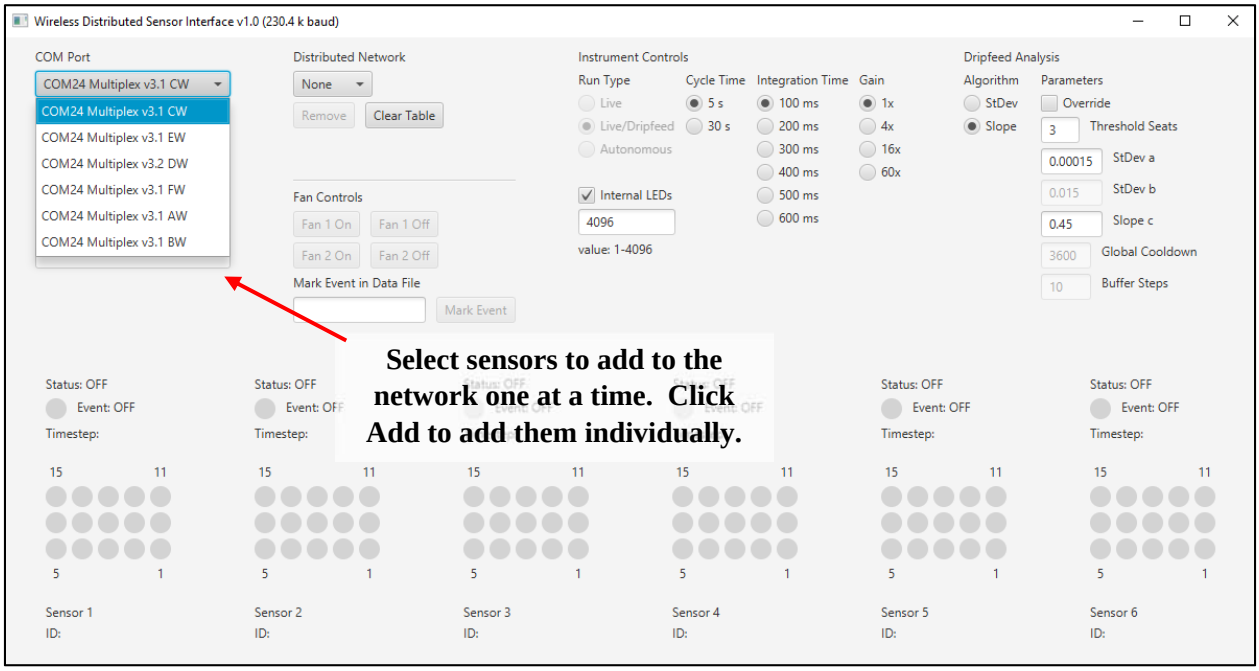
The interface and use guidance for ABEAM-15 devices in a network scenario is similar to that described for the individual units of Appendix A. In this case, the software is a file named "starPointSix.jar".

Fig. B1. The startup screen for the network software



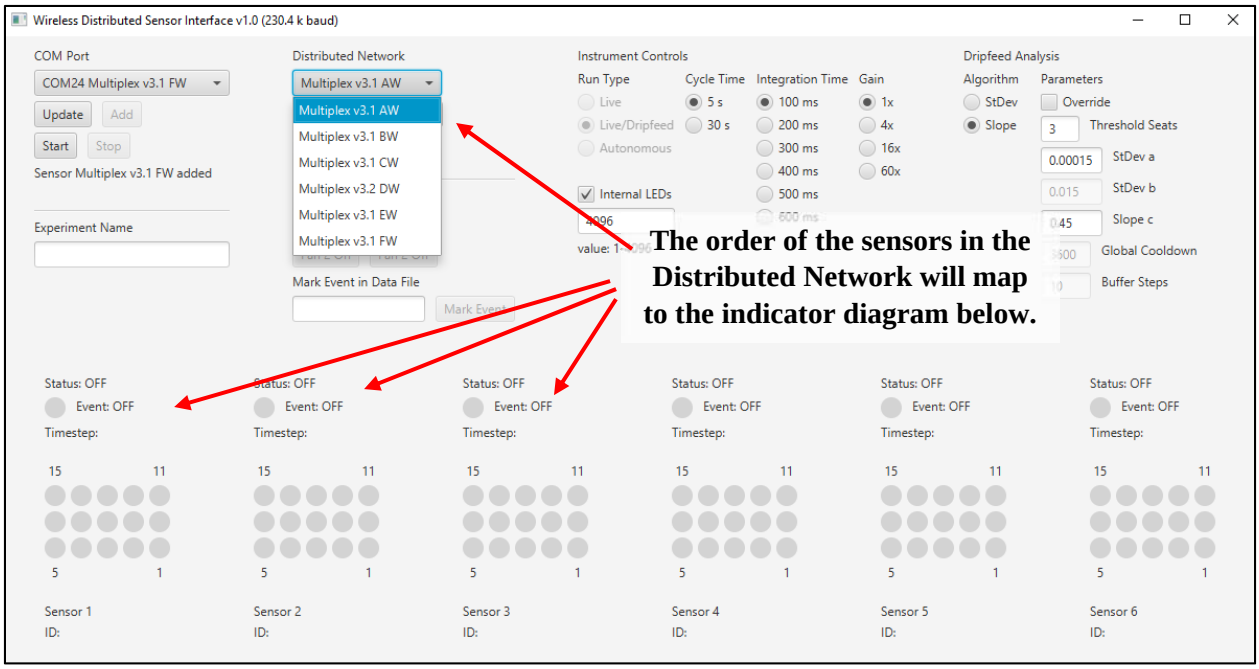
All sensors in the network must be wireless and connected to the same dongle (single COM port). The order of the instruments in the COM port list is not important. Updating the COM port list will not change the Distributed Network list.

Fig. B2. Populating the network



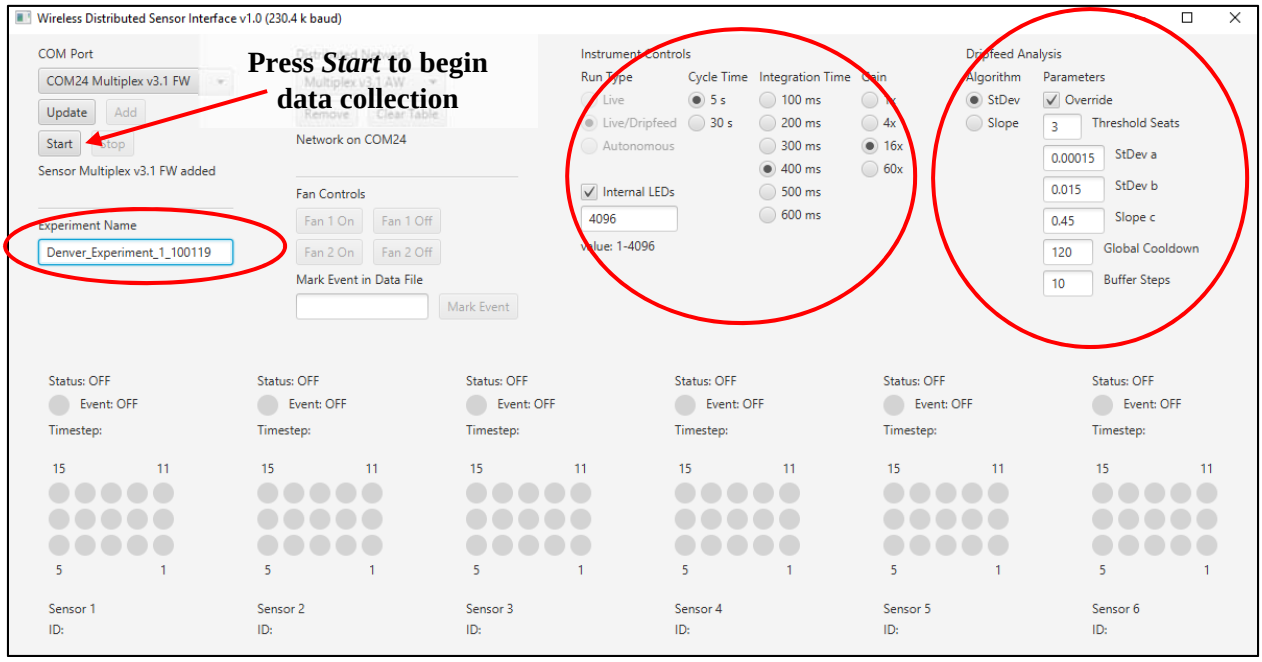
The order of the sensors in the Distributed Network is important. You can remove a sensor at any time by clicking *Remove*. Or, clear the entire list by clicking *Clear Table*. The Network can hold no more than six instruments. Less than six is OK.

Fig. B3. Device selection order



Network experiments will always use dripfeed analysis. Set instrument controls, dripfeed analysis parameters, and choose an experiment name. Details are provided in Appendix A. Press Start to begin data collection.

Fig. B4. Set instrument and dripfeed parameters



As in Appendix A, selecting *Stop* will end data collection. This will take 1 to 3 minutes during which the software will be frozen; do not force quit.

The StarPointSix interface only contains options that affect all six sensors. However, individual instruments do save their data to flash, and it is possible to analyze an individual instrument's data offline.

(1) To download data from an individual sensor, connect to it using Multiplex_314u (Appendix A).
(2) To analyze data from an individual sensor file in a network run, use the Analysis.jar tool (Appendix C).

Tagging events in StarPointSix.jar works exactly like it does in Mulitplex_314u.jar, except that it affects all of the attached sensors.

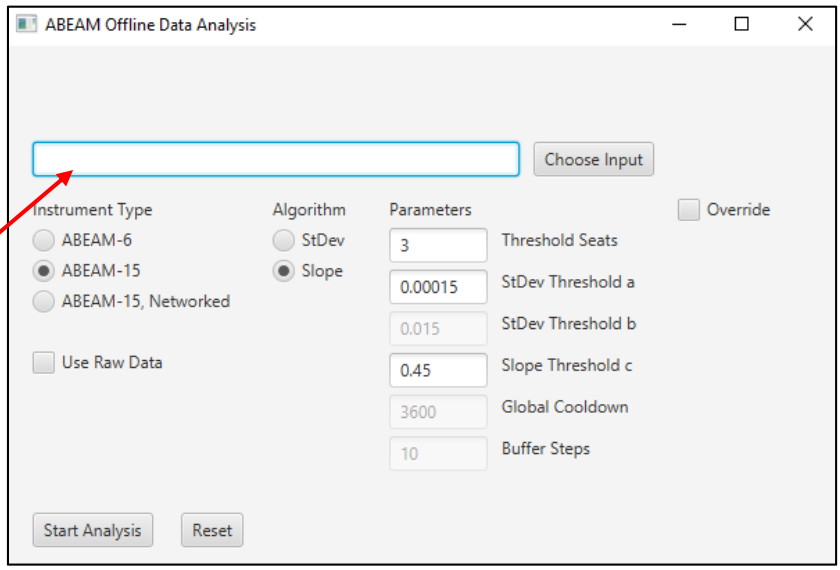
There are both global and local metadata files in a network run. Tagged events are saved to the global file. Also, time is kept from a single master clock. Individual timesteps are recorded for each sensor, but the real time measurement is in reference to the central clock. The times from different sensor files can be compared directly against each other.

Appendix C

ABEAM OPERATING INSTRUCTIONS: OFFLINE DATA ANALYSIS

Offline, post-collection analysis of the data can be completed using the software is a file “Analysis.jar”. Many of the options look identical to those in Multiplex_314u (Appendix B), and they work the same way. When selecting the instrument data to analyze, the exact file chosen is not important. It just needs to be in the correct directory. For example, you could choose the _data, _parameters, or _raw file for the particular set of data.

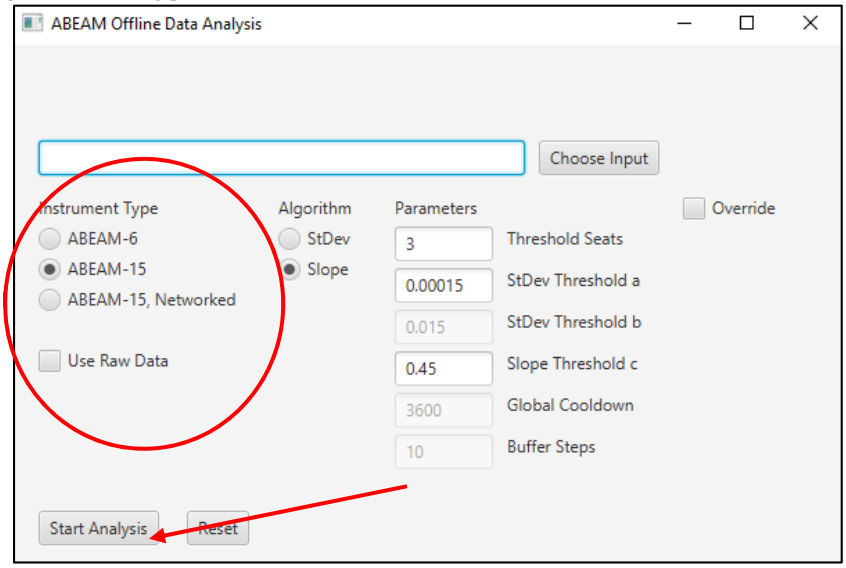
Fig. C1. Starting the analysis software



Choose the file from the instrument run that you would like to analyze

Be sure to specify the instrument type that was used to obtain the data. Instruments using network mode have different file structures than instruments in live and autonomous mode. It is important to get this distinction correct or the software will return errors.

Fig. C2. Selecting parameters



Select *Start Analysis* to analyze your data set.

A data file may be analyzed more than once with different sets of parameters. In order to do this you must re-name or remove the original offline analysis. It is located in a folder named “Offline_Analysis”. The entire folder must be moved, renamed, or deleted.