IBAC
Interface Design Description (IDD)

ICx Biodefense
208 Business Center Drive
Reisterstown, MD 21136

REVISION HISTORY

REVISION	DATE	CHANGE DESCRIPTION	AFFECTED SECTIONS/PAGES
0	19 Mar 09	Draft Version	All

Table of Contents

Tables ii

1. Introduction 1

2. Hardware Set-Up 1

3. Data Communication Parameters 1

4. Data Interface 1

5. Remote Programmability 6

Tables

Table 1. 6-Pin Power and Communication Connector 1

Table 2. Particle Count Data Description 2

Table 3. Diagnostics Data Description 2

Table 4. Baseline Data Description 3

Table 5. IBAC Messages 3

Table 6. IBAC Commands 5

1. Introduction

The Interface Design Description (IDD) document for the IBAC instrument describes the data interface for the IBAC biological particle detector. It includes connector description, messages transmitted by the IBAC and commands that can be sent to the unit.

2. Hardware Set-Up

1. Power and RS232 serial data connections to the IBAC circuit card are via a 6-pin dual-row connector header, Molex P/N 43045-0619. The user mates to this using a 6-pin connector receptacle, Molex P/N 43025-0600 with 20-24 AWG female crimps, Molex P/N 43030-0008.

   The 6-pin connector on the IBAC, labeled P2, is wired as described in Table 1:

   Table 1. 6-Pin Power and Communication Connector

Pin Designator	Function
1	**RS232 Ground (DB9 pin 5)**
2	**Power Ground**
3	**20-35V DC Input**
4	**No connection**
5	**IBAC Rx (DB9 pin 3)**
6	**IBAC TX (DB9 pin 2)**

3. Data Communication Parameters

The IBAC is equipped with an RS-232 communication port for interface with a data acquisition system or computer. The RS-232 port is set up as follows:

- 57600 Baud Rate
- 8 data bits
- 1 stop bit
- No parity
- No handshake

4. Data Interface

The IBAC operates in continuous sampling mode and can provide data in real time or at a specified interval.

When first powered on, the unit will perform its initialization routine and output a status message indicating it is ready:

$info, revision 1.04, ICx Biodefense IBAC, unit number = IBAC-WACS-1A-163\r\n
$info, system ready\r\n

If previously enabled via the **$trace rate,Y** command, threat information will output once every Y seconds in this format (quotes not output):

"**$trace,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P**" where the particle count data is described in Table 2:

Table 2. Particle Count Data Description

Data Code	Description	Format	Data Range
A	Small instantaneous particle counts (C-S-I)	%d	0-50000
B	Large instantaneous particle counts (C-L-I)	%d	0-50000
C	Small instantaneous fluorescent particle counts (BC-S-I)	%d	0-50000
D	Large instantaneous fluorescent particle counts (BC-L-I)	%d	0-50000
E	Small MA particle counts (C-S-A)	%.1f	0-50000
F	Large MA particle counts (C-L-A)	%.1f	0-50000
G	Small MA fluorescent particle counts (BC-S-A)	%.1f	0-50000
H	Large MA fluorescent particle counts (BC-L-A)	%.1f	0-50000
I	Small MA biological percent (B%-S-A)	%.1f	0-100
J	Large MA biological percent (B%-L-A)	%.1f	0-100
K	Instantaneous size fraction percent (SF-I)	%.1f	0-100
L	MA size fraction percent (SF-A)	%.1f	0-100
M	Alarm state counter	%d	0-32767
N	Valid baseline data?	%d	0-1
O	Alarm status	%d	0-1
P	Alarm latch state	%d	0-1

The string above and all of the strings transmitted by the IBAC are followed by Carriage Return (CR (\r)) and Line Feed (LF (\n)) ASCII characters.

Received characters are echoed byte for byte as they are received. When a CR is received, it signals the termination of a command and is echoed followed with an LF character. Null bytes (0) are not discarded by the data interface, so please be careful not to inadvertently transmit a null character at the end of a command. Output messages from the IBAC are not suspended as input bytes are received.

The IBAC will periodically output diagnostic monitor values and diagnostic alarm states. The default output rate is once every 7 seconds, but can be set by the **$diag rate, Y** command to output diagnostics every Y seconds. The diagnostics and diagnostic alarms are output in this format:

"**$diagnostics,A,B,C,D,E,F,G,H,I,J,K,L,M,N**" where the diagnostic data is described in Table 3:

Table 3. Diagnostics Data Description

Data Code	Description	Format	Data Range
A	Outlet pressure (psi)	%.1f	0-5
B	Pressure alarm	%d	0-1
C	Temperature (deg C)	%.1f	-20 to 90
D	Temperature alarm	%d	0-1
E	Laser power monitor	%d	0-800
F	Laser power alarm	%d	0-1
G	Laser current monitor (mA)	%.1f	0-80
H	Laser current alarm	%d	0-1
I	Background monitor (V)	%.2f	0-5
J	Background alarm	%d	0-1
K	Input voltage (V)	%.1f	0-35
L	Input voltage alarm	%d	0-1
M	Input current (mA)	%d	0-2000
N	Input current alarm	%d	0-1

A sample transmission over a small time interval will appear as follows:

$trace,540,108,180,18,720.6,97.6,453.5,30.8,62.9,31.6,16.7,11.9,0,0,0,0
$trace,600,120,200,20,719.8,99.0,446.3,30.6,62.0,30.9,16.7,12.1,0,0,0,0
$trace,660,132,220,22,719.4,100.6,438.7,30.4,61.0,30.2,16.7,12.3,0,0,0,0
$diagnostics,1.7,0,31.0,0,280,0,51.3,0,0.21,0,24.1,0,416,0
$trace,720,144,240,24,719.4,102.4,430.7,30.2,59.9,29.5,16.7,12.5,0,0,0,0
$baseline,30.8,38.1,33.4
$trace,780,156,260,26,719.8,104.4,422.3,30.0,58.7,28.7,16.7,12.7,0,0,0,0

Every 60 seconds, the unit will output baseline values (long term-moving averages) in this format:

"**$baseline,A,B,C**" where the baseline data is described in Table 4:

Table 4. Baseline Data Description

Data Code	Description	Format	Data Range
A	Large MA fluorescent particle counts baseline	%.1f	0-50000
B	Large MA biological percentage baseline	%.1f	0-100
C	Size fraction baseline	%.1f	0-100

If the unit encounters a fault condition, it will transmit a fault string in the following format:

"**$fault,fault_code,fault_string**" where **fault_code** is an integer number and "**fault_string**" is an information string that is terminated by the \r and \n characters. Fault messages are transmitted periodically so long as they are present, but no message is transmitted when the fault clears. The repeat interval for active fault messages to be retransmitted is configurable via the **$fault repeat** command.

Other strings transmitted by the unit are unit status information. The format is **"$info, info_string"** terminated by '\r' and '\n' characters. An example of this transmission is: **"$info, collecting sample"** or "**$info, the unit has alarmed**". Table 5 shows a complete list of the IBAC messages:

Table 5. IBAC Messages

String	Description	Parameters
$info, Revision 1.04, ICx Biodefense IBAC, unit number = IBAC-WACS-1A-003\r\n	Indicates version number and unit number	***1.04*** – current firmware version number ***IBAC-1000J-1A-003*** – Serial number
$fault, 10, pressure = x.x psi is outside range.\r\n	The intake may be clogged or the unit pump damaged.	***x.x*** – the pressure value read by the instrument in psi. Values above 3 psi may indicate a clogged intake
$fault, 20, laser power above range\r\n	Laser output level is outside range	
$fault, 20, laser power below range\r\n		
$fault, 30, laser current out of range, init = x, curr = y\r\n	Laser current is outside of the allowed bounds. A problem exists.	***x*** – Current reading at startup ***y*** – Current reading at the present
$fault, 40, background light monitor below range\r\n	Background light monitor below allowable level.	
$info, collecting sample\r\n	Unit is collecting a sample onto the sampler disk	This command is transmitted in response to $collect,1 command or in the event the collector is automatically triggered from an alarm

<table>
<tr><th>String</th><th>Description</th><th>Parameters</th></tr>
<tr><td></td><td></td><td>condition</td></tr>
<tr><td>$info, the unit has alarmed\r\n</td><td>The unit has detected an alarm condition</td><td></td></tr>
<tr><td>$info, system ready\r\n</td><td>The unit has finished initialization process and it is ready to receive any command</td><td></td></tr>
<tr><td>$s,version_number, serial_number, collection_disk_status, fault_status, fault_code</td><td>This message is sent in response to the $status command</td><td>version_number – The firmware version number. Example, 1.04
serial_number – The unit’s serial number. Example, IBAC-1000T-BETA-1B-003
collection_disk_status – 1 if disk is spinning, 0 otherwise.
fault_status – 0 if no faults, 1 if unit shows a fault,
fault_code – decimal number whose bits represent which fault codes are currently active. These bits are defined as follows:
<table>
<tr><th>Bit</th><th>Fault code (refer to $fault)</th></tr>
<tr><td>0</td><td>10</td></tr>
<tr><td>1</td><td>20</td></tr>
<tr><td>2</td><td>30</td></tr>
<tr><td>3</td><td>40</td></tr>
<tr><td>4</td><td>50</td></tr>
<tr><td>5</td><td>60</td></tr>
<tr><td>6</td><td>70</td></tr>
<tr><td>7</td><td>80</td></tr>
</table>
If bit is set, that fault is active</td></tr>
<tr><td>$trace,A,B,C,D, E,F,G, H,I,J, K,L,M, N,O,P</td><td>This message is sent in response to the $air sample command or when the trace is enabled via the $trace rate, X command</td><td>The codes A-P are defined in Table 2 for the $trace command</td></tr>
<tr><td>$diagnostics,A,B,C,D, E,F,G, H,I,J, K,L,M,N</td><td>This message when diagnostics trace is enabled via the $diag rate, X command</td><td>The codes A-N are defined in Table 3</td></tr>
<tr><td>$baseline, A, B, C</td><td>This message is sent every 60 seconds</td><td>The codes A-C are defined in Table 4</td></tr>
</table>

The unit is also designed to accept commands. All commands transmitted to the IBAC (terminated with CR) will be echoed back to the host interface (terminated by CR+LF). These commands are described in Table 6 below:

Table 6. IBAC Commands

Command	Description	Parameters
$alarm,w	Turn alarm function on or off	**w** – 0 Turn alarm capability off 1 Turn alarm capability on (default)
$clear alarm	Clears the alarm latch state	The alarm latch state is asserted when a biological particle alarm is triggered and continues to be asserted until cleared by the user. This allows for an alarm to be detected in the event communication is lost during an alarm event.
$status	The unit sends back it's status	Unit answers with $s, version_number, serial_number, collection_disk_status, fault_status
$sleep	Puts the unit in sleep mode	Put the unit in low power mode (sleep), unit will wake up if it receives any command over the serial port (except another $sleep command). When it wakes up, the unit will reset itself. The unit will run through the initialization process and output the $info,system ready message.
$trace rate, p	Set the data output rate	**p** – output rate in seconds. Default is 1 second. If p is set to zero, the output will be disabled (but alarm events will still be reported)
$diag rate, p	Set the diagnostic output rate	**p** – output rate in seconds. Default is 7 seconds. If p is set to zero, the output will be disabled (but alarm events will still be reported)
$air_sample	Queries current particulate level and alarm state	Unit answers with $trace,A,B, C,D,E, F,G,H, I,J,K, L,M,N,O,P where the codes A-P are defined in Table 2 for the $trace command
$collect, n	Turn on or off the sampler disk. This command can be issued at any time	**n** – 0 off (default) 1 on
$auto_collect,n,p	Turn off automatic triggering of the collector via the internal alarm status	**n** – 0 off 1 on (default) **p** – minimum runtime for collector in event of alarm (default is 60 seconds)
$prog	Put IBAC into remote reprogramming mode	This will allow the unit to update its firmware using the **dpgm** program as described in section 5. Once in programming mode, make sure you only use the dpgm.exe program to communicate with the unit. The unit returns to normal operation when programming is complete.

All of these strings must be terminated by CR (ASCII 13, \r). If the unit receives an invalid command, it will transmit "**$invalid**" back to the host. Otherwise, it will perform the command and only sends a string back if data is requested.

5. Remote Programmability

The firmware on the IBAC may be updated via the serial interface to a host computer. To use this feature, follow these steps:

1) Connect a computer to the IBAC as described in Section 2.
2) Create a directory on the computer that will communicate with the IBAC, for example, C:/wacs. Unzip the files in the prog_taci.zip file into the newly created directory.
3) Take the new firmware file, such as wacs_v102.hex, and copy it to the directory created in step 2.
4) Use a serial terminal program to connect to the IBAC.
    a. Setup a new session.
    b. Setup the COM port that you are using to communicate with the IBAC.
    c. RS-232 parameters are:
        i. Baud: 57600
        ii. Data bits: 8
        iii. Stop bits: 1
        iv. Parity: None
        v. Flow control: None
5) Using HyperTerminal, send the command "**$prog**" followed by CR to the IBAC. You now have 250 seconds (a little over 4 minutes) to complete the following steps or the IBAC will automatically reboot and you have to start over at step 1.
6) Open a DOS command window.
7) Go to the directory where the files in step 2 were unzipped.
8) Type the following (use the appropriate com port if not using com1): **dpgm –i COM1 –b 57600 wacs_v102.hex** (where "wacs_v102.hex" is the filename).

    You will receive an output similar to the following.

```
C:\ibac>dpgm -i COM1 -b 57600 wacs_v102.hex

Reading Target Device ID...   Found microcontroller

Reading HexFile.
Reading Target

Programming Device
.............................................................
.............................................. Done.
```

9) The unit will reset and will run the new code.

For Customer Support call:
410.517.0800
For Technical Assistance call:
410.517.0800