Measurement Board Integration Guide

Using the Big Fin Scientific Data-Entry Boards with Custom Applications

▸ Version Table

2.0	07/27/2020	2nd Release with new DCS5 commands.

For help, write to: info@bigfinscientific.com or call 512-808-0FIN (0346)

Table of Contents

Introduction 5

Conventions 5

Host Commands 6

Presence Check (Ping) 6

Board Stats 6

Reply variation for Model: 10MF1 7

Reply variation for Model: DCS1 7

Reply variation for Model: DCS5 7

Command: Beep 7

Initialize Board 8

Reboot 8

Set Interface 9

OTA Firmware Update 9

BT Communication Relay 10

LED-Ring Light-Up Sequences 11

LED-Ring Illuminate w/ Meter 11

Set Sensor Mode 12

Check Battery Level (all models) 13

Battery Reset Options (DCS5 Micro Only) 14

Battery Presence Check 14

Power Save Modes 15

Control-Box Temperature and Humidity 16

Stylus offset 17

Stylus Status Messages Enable/Disable 18
Sensor Settling Delay 18
Stylus Reading Max Deviation Allowed 19
Sensor Number of Readings 20
Check Cal State 21
Specify mm values for Calibration Points 21
Take Reading for a Calibration Point 22
Command: Clear Calibration Data 23
Command: Restore Calibration Data 23
Backlighting 24
Set Backlighting Level 24
Set Backlighting Sensitivity (for Auto Mode) 24
Set Backlighting Auto Mode 25
Command: Column List 25
Command: Dump Data 26
Unsolicited Messages from Board to Host 26
Stylus Down/Up Indicator 27
Sensor Position (Length) Readout in mm 27
Swipes 28
Faceplate Key Pushes 28
Length Measurements from 10MF1 to Host 29
Calibration Example 31
Specify Calibration Point 1 31
Specify Calibration Point 2 32

Obtain Raw Value for Calibration Point 1 32

Obtain Raw Value for Calibration Point 2 33

Clearing Calibration Data 34

Restoring Calibration using Predetermined Raw Values 34

Inquiries **36**

Introduction

This document describes the serial commands used to communicate with a Big Fin Scientific data-entry board (Electronic Fish Board), such that you can use Big Fin Scientific boards with your own applications. The boards are capable of functions beyond length-measurements, including "swipe" detection, opening up more possibilities for interacting with your users. Should you find that the existing command-set is insufficient for your application, please call to discuss your requirements.

See the "Getting Started Guide" for information regarding establishing basic communication between the Big Fin Scientific data-entry board and the computer or device on which you run your application.

Conventions

In the examples below, blue text denotes commands which are sent from the host to the measurement-board and green text are messages sent from the measurement-board to the host.

"Host Commands" are commands initiated by the host. "Device Messages" are sent from the measurement board to the host without a preceding request from the host. The host must be ready to receive at all times in order to take advantage of these commands, as the boards do not use any handshaking or RTS/CTS signals.

A DTR signal toggle from high to low will cause the measurement-board to reboot. A DTR toggle often happens during connection of any USB device on a USB bus, so the application should expect that the measurement board will lose connection at (any) USB connect/disconnect events.

Not all commands are available on all measurement-boards. Check for the Applicable Models entry for each command or message. If a model is not listed for that command, there may be a reply indicating an error or there may be no reply.

Host Commands

Presence Check (Ping)

Use this command to verify the board is communicating correctly with your application. A <CR> (carriage return character) is not needed for the command, but is included at the end of the return string.

Applicable Models: All

Send to Board: a#
Receive from Board: %a:e#

%a, → Preamble
e → Value returned from board
: → Separator between preamble and value returned
→ Response termination character

Example:
a#
%a:e#<CR>

Board Stats

Use this command to determine board characteristics, including the board type, the firmware version, the number of records present in the memory of the board*, the total number of records available for storage* and a maximum sensor reading value**

(*) if applicable to model and firmware version.
(**) applicable to DCS5 models only.

Applicable Models: All, see variations described below

Send to Board: b#
Receive from Board: %b:T,V,rU,rT,mThresh#

Received communication description:
%b → preamble
: → separator between preamble and value returned
T → board type: 10MF1 (T=0), DCS1 (T=1), 10MF2 (T=2),DCS5(T=3)
V → firmware version numeric, with right-most 2 digits being minor (i.e. 102 = ver1.02)
rU → fish records used (In the case that you're using a 10MF1 in standalone mode, records are stored on the 10MF1. rU represents the total # of fish-record slots on the measurement-board which have been filled with

information (used slots).) As fish-records are not stored on DCS1 and DCS5 boards, this value isn't meaningful for those models and is typically zero.
rT → fish records total (In the case that you're using a 10MF1 in standalone mode, records are stored on the 10MF1. rT represents the total # of fish-records slots which are available on the measurement-board, both empty and filled.) DCS1 and DCS5 boards reply with the mThresh value instead.
mThresh → max possible length-sensor reply (ie, a maximum number which your sensor will report in terms of length units). mThresh is useful for troubleshooting and sanity-checking a sensor response during application start-up or diagnostics. This value is reported by DCS5 and a small number of DCS1 boards.

, → separator between responses

→ response termination character.

Reply variation for Model: 10MF1

Receive from Board: %b:0,V,rU,rT#

Example:
b#<CR>
%b:0,216,1910,15655#<CR><LF>

Example Explanation::
Board responds to our inquiry that it's a 10MF1 board, using firmware version 2.16 with 1910 records used of a possible 15655 record slots available. Depending on the particular firmware version of the 10MF1 measurement-board you have (which may have been specially made for you), the available record number will vary from 5K to 20K.

Reply variation for Model: DCS1

Note that DCS1 and DCS5 boards do not store fish-records on-board.

A DCS1 board, depending on firmware version, may reply with: %b:1,V,0,0# OR with %b:T,V,0,0,mThresh#

Reply variation for Model: DCS5

Note that DCS1 and DCS5 boards do not store fish-records on-board.

A DCS5 board will reply with: %b:3,V,0,0,mThresh#

Command: Beep

Applicable Models: 10MF1, DCS5-XT*

Use this command to identify a board with an audible beep and lights flashing on the faceplate. Useful if you're connected to multiple measurement-boards.

Send to Board: g#
Receive from Board: no response
---

* In the case of the DCS5-XT, the Beep command requires the speaker/resonator upgrade. Speakers are standard on 10MF1 measurement-boards.

Initialize Board

The &init# command will reset the measurement-board Persistent Storage (memory) and the measurement-board Bluetooth module to factory settings. This command is typically only required to solve a memory corruption issue or in the case that you have written some parameter values to the BT module over Command Mode which you wish to restore to factory defaults.

Initializing the measurement-board will:

1) Erase calibration and sensor data
2) Restore / reprogram the BT module to "standard" settings - any customization you've done to connection or module name settings will be lost
3) Re-format the on-board EEPROM (persistent storage)

The init command will not initialize the battery fuel-gauge. Fuel gauge initialization is done separately.

Applicable Models: DCS5

Send to Board: %init#
Receive from Board: "Rebooting in 2 seconds..."

Reboot

Use this command to reboot the board.

Applicable Models: DCS5 Micro

Send to Board: &rr#
Receive from Board: %rebooting#

A DTR signal toggle from high to low will also cause the measurement-board to reboot.

Set Interface

Applicable Models: 10MF1, DCS1, DCS5 with specific firmware

DCS1 and DCS5 measurement-boards with specific firmware offer two Interfaces. Interfaces currently supported are:

1) DCSLinkstream
2) FEED

The primary difference between the two is that the FEED Interface offers measurement-line key code modes while the DCSLinkstream Interface always sends measurements in millimeters.

Send to Board: &fm,x#
Receive from Board: "DCSLinkstream Interface Active" or "FEED Interface Active"

OTA Firmware Update

The purpose of this command is to load firmware to the measurement-board Microcontroller / CPU via Bluetooth (ie, Over-the-Air). This command shuts down all activity except for the routine which receives and loads the new firmware to the persistent memory module which is used by the Microcontroller. A successful firmware loading sequence will cause the board to reset after programming.

Applicable Models: DCS1, DCS5

Send to Board: %h,VER,BR#
Receive from Board: %h:0#

%h, → Preamble
VER → Must be 1 unless supplied with special firmware.
, → Separator between arguments to command
BR → Baud Rate (always 115200)
: → Separator between preamble and value returned
0 → (zero) Command successful indicator. A non-zero value indicates an error (please contact us).
→ Send and response termination character

Example:
%h,1,115200#
%h:0#<CR>

This command and response are followed by transmission by the host of a firmware file per the specification found in the Parallax Propeller 1 programming guide.

Note: Implementing OTA firmware updating for your application is not recommended. Updating firmware via USB is much easier to implement. See the Parallax documentation regarding the Propellent commands, which provide command-line/batch file Propeller Microcontroller programming support over USB. If USB programming is not desirable for your application, please contact us to arrange for assistance.

BT Communication Relay

Use this command to directly communicate with the on-board Bluetooth module. The BT module has its own microcontroller and command-set, separate from the measurement-board microcontroller. This command creates a "relay" communication channel through the measurement-board micro to the BT module micro and can be used to change settings on the BT module.

When this command is issued, all measurement-board activity stops, other than the relay communication and a communication channel from your terminal (or program) to the BT module has been established. Refer to the following BT module technical documentation for an explanation of applicable commands:

- 10MF1, 10MF2,DFS/2, DCS1 and DCS5-XT boards → MicroChip RN42
- DCS5 Micro → MicroChip RN4678

Note: This command does not send a "command mode" message to the BT module; you must do that yourself. This command merely opens a relay channel. A measurement-board reboot is required to restore normal operation.

Note2: The BT modules of all models are programmed such that the Command Mode Timer never expires. Thus, you can reach Command Mode over the USB and BT channels at any time. You can change this behavior for security purposes, should that be a concern.

The BT module communicates with the measurement-board CPU at a baud rate of 115200. Changing the BT module baud rate via BT Module Command Mode commands will break the relay communication functions.

A list of default BT settings and values are sent from the board after an &Init# command. You must be connected to the measurement board via USB to see these messages, as the BT module is reset and connections are list during the Initialize process.

Applicable Models: DCS1, DCS5

Send to Board: %r#
Receive from Board: nothing

Communication description:

%r → preamble
→ command termination character.

Example:
%r#<CR>

Any further characters sent from the host to the measurement-board will be relayed to the BT module. Any characters sent from the BT module will be relayed to the host. (measurement-board commands are ignored.)

Reset the board to exit this mode. In order to save BT module settings which you have changed, you may need to reboot the BT module prior to rebooting the measurement-board.

LED-Ring Light-Up Sequences

Applicable Models: DCS5 Micro

Message to Board: &ra# OR &ra,x# (wherein x = 0,1 or higher number)
Message from Board: (no reply)

&ra, → Send preamble with separator - alternative command is &ra# with no separator and no digit (x)
0,1,2 or higher number → Must be 0 to 254. Different firmware versions will offer different RA Flash sequences. Sequences will be added over time.
, → Separator between arguments to command
→ Command and response termination character

Example1:
&ra,2#
(no reply via comm channels) - board LED ring flashes with sequence #2

Example2:
&ra#
(no reply via comm channels) - board LED ring flashes with default sequence

Note: If you have a particular sequence you'd like to see, please get in touch. We understand that some sequences are triggering for some people, so we'll work with you to establish a custom Record-Accept flash sequence to your specifications and provide a new command number for you to use.

LED-Ring Illuminate w/ Meter

Applicable Models: DCS5 Micro

Message to Board: %li, color, intensity, meter-level#
Message from Board: %li#

&li, → Send preamble with separator
color → See color codes below.
intensity → Integer from 1 to 100 with 100 representing maximum intensity.
meter-level → Integer from 0 to 8. Zero turns off the LED ring. The meter-level integer will turn on the integer quantity of LEDs starting at the LED1 position at bottom left and moving clockwise around the circle.
, → Separator between arguments to command
%li → Return message preamble
→ Command and response termination character.

Color Table	
01	Green
02	Yellow
03	Orange
04	Red
05	Purple
06	Blue
07	White

Example:
%li,04,100,5#
%li#<CR>

In this example, the LED ring illuminates the first 5 LEDs as max-intensity red.

Set Sensor Mode

Message to Board: &m,x# (wherein x is 0 or 1)
Message from Board: %m:x#

&m, → Send preamble with separator
x → 0 or 1

, → Separator between arguments to command
%m → Return message preamble
: → Separator between return preamble and value returned
→ Command and response termination character

Example:
%m,0#
%m:0#<CR>

The measurement-board is then in "Length Measure Mode", in which error checking according to the error-checking parameters set elsewhere is done prior to the measurement-board sending a length. (stylus-down and stylus-up indication is sent without error checking.)

Sending this command with x=1 puts the measurement-board into "Keyboard Mode" which will reply as quickly as possible with minimal error-checking. This is useful for hot-key and alphanumeric data-entry.

The measurement-board messages (format and content) are the same, no matter which mode is selected. This command merely affects whether error-correction is active during stylus-down events.

Check Battery Level (all models)

(Note: Some measurement-boards do not have batteries. Boards without batteries will return invalid values.)

Use this command to ascertain the State of Charge (SOC) of the board. The returned SOC value is an integer from 0 to 100 (%). Note that it's best not to deplete the battery below 15% to maintain longest battery life, so it's best practice to remind the user to charge the board as the battery moves below 25% SOC.

Applicable Models: All boards containing batteries (any model).

Send to Board: &q#
Receive from Board: %q:xxx

&q → Preamble for command sent from host
%q → Reply Preamble
, → Separator between preamble and value returned
xxx → The charge state as an integer between 0 and 100. This is the battery level percentage wherein 100 is completely charged and zero is completely drained.
→ Query and response termination character

Example:
&q#
%q,15#<CR>

In this example, the board reports a battery SOC of 15%.

Battery Reset Options (DCS5 Micro Only)

The DCS5 Micro uses field-swappable battery packs that feature LiFePO4 cells (Lithium Ferrous Phosphate). These cells offer a longer life than standard Lithium Polymer batteries, but are more difficult to gauge. From time to time, it may be required to "re-initialize" the battery fuel-gauge circuitry in order to receive accurate SOC values.

Applicable Models: DCS5 Micro

Send to Board: &qi,reset-option#
Receive from Board: %qi,reset-option#

%qi, → Preamble
, → Separator between preamble and value returned
reset-option→ The battery reset type, wherein:
0 and 1 are reserved (do not use)
2 is a full reset and start to battery pack initialization.

Use this command to re-initialize and start the "learning" sequence of the on-board fuel-gauge. After issuing this command, approximately 4 sequential discharge/charge cycles of the same battery pack will allow the fuel-gauge to have learned the characteristics of the pack and provide accurate SOC values. If the pack is replaced with another, the fuel-gauge will detect this event and restart learning for the new pack. The SOC values will already be fairly close for the new pack. The fuel-gauge will recognize up to 6 packs, allowing field-swapping of those packs without losing SOC reporting accuracy and without the need for the 4-cycle learning process.

Note: This command affects other commands: &qv# , &qa# , &qe# , &qf#
in that it will take at least 4 charge/discharge cycles in order for the above commands to reply with accurate information.

Battery Presence Check

Applicable Models: DCS5 Micro

Send to Board: &qq#
Receive from Board: %qq#

&qq → Command preamble

%qq → Reply preamble
→ Query and response termination character

Note: &qq# command is only supported on the DCS5 Micro. All other models will ignore this command.

Power Save Modes

The DCS5 boards offer several power-save modes to increase battery life. Do not use modes which are marked as "Reserved".

Applicable Models: DCS5 Micro, DCS5 XT

Message to Board: &ps,x,y#
Message from Board: %ps:x,y#

&ps, → Send preamble
%ps → Reply Preamble
x → Integer from 0-255 representing the sum of the power save modes desired.
, → Separator between arguments to command
: → separator between preamble from board and value returned from board
→ Query and response termination character

Power Save Mode			
X	Name	Description	Y
0	None	No power savings - turn off all settings (reset to full power)	n/a
1	Sensor Off	The sensor is turned off. Sensor must be turned back on by teh host (ie, this is not a sleep mode with wake function.) Wake up time for the sensor is approximately 300ms after receipt of sensor on command.	0=off 1=on
2	Dim LEDs	LED indicators (Light Ring or Light Bar) are dimmed to half their settings.	0=off 1=on
3	Ambient Sleep	Temp/Humidity sensor, Ambient Light and Battery Fuel Gauge sensors enter sleep mode. Set the wake/update interval with the Y parameter (in seconds).	0=off 1=on
4	BT Sleep	The Bluetooth module enters sleep mode. The board will wake	0=off

		with a faceplate keypress or measurement-line keypress if the sensor has not been turned off. The board will wake with a single character but the character will be lost (not buffered).	1=on
5	BT Deep Sleep	The BT module enters Deep Sleep and will consume almost no energy. This will cause large latency in reconnections. Wake with a single character (which will be lost). Expect longer wake-up time than with Sleep.	0=off 1=on
6	BT TX PWR	The transmit power of the BT module can be turned down to save energy. This results in decreased transmission range. Default is 7 (highest power).	0-7
7	Reserved		
8	Reserved		

Example1:
&ps,0#
%ps:0#
(return to full power mode and reset all other X settings to defaults).

Example2:
&ps,4,1#
%ps:4,1#
&ps,1,1#
%ps:1,1#
(turn Length Sensor off and turn on BT Sleep Mode)

Control-Box Temperature and Humidity

Applicable Models: All

Send to Board: &t#
Receive from Board: %t,xx,yy#

&t → Command Preamble
%t → Reply Preamble
, → Separator between values.
xx → Temperature inside board
yy → Humidity inside board.
→ Query and response termination character

This command will return the temperature and humidity levels inside the board or control box (whichever is sealed to the environment). This command is useful for alerting the user to replace the desiccant pack (humidity > 40) or telling the user to move the board out of the sun if temp > 60C.

example:

&t#<CR>
%t,32,19#<CR>

The measurement-board returns 32C internal temperature and 19% relative humidity.

Stylus offset

Applicable Models: DCS1,DCS5 with specific firmware

This command is useful if your host application is providing data-entry functions with the stylus (on the measurement line).

It is intuitive for most users to line up the center of the magnet with a "key" on the measurement graphic. Since the board has been calibrated to send positions relative to the stylus tip, and since the tip is some distance away from magnet center, an offset can be used to send more relevant position data.

This is only useful if you switch modes and if you're using firmware which supports measurement-line "key" sending. You can accomplish the same thing by interpreting position data as appropriate in your host application.

Send to Board: &so,v#
Receive from Board: %so:v#

&so → Command Preamble
%so → Reply Preamble
, → Separator between values.
v → Integer number representing offset from stylus measurement tip to center of the magnet
: → separator between preamble from board and value returned from board
→ Query and response termination character

example:

&t#<CR>
%t,32,19#<CR>

The measurement-board returns 32C internal temperature and 19% relative humidity.

Stylus Status Messages Enable/Disable

Applicable Models: DCS1,DCS5

This command enables or disables the stylus up/down status messaging.

Send to Board: &sn,x#
Receive from Board: %sn:x#

%sn, → Preamble for command sent from host
x → Value of 0 or 1 with 0 being messages are turned off and 1 being messages are turned on
: → separator between preamble from board and value returned from board
→ Query and response termination character

Example1:
&sn,0#
%sn:0#<CR>

In this example, stylus up/down messaging is turned off and no messages are sent during stylus detection events. Subsequently a length message might look as the following:

%l,265 (with stylus up/down messaging disabled)

Example2:
&sn,1#
%sn:1#<CR>

In this example, stylus up/down messaging is turned on and messages are sent during stylus detection events, in the following format:

%t,0#%l,265#%t,1# (with a length of 265 message in between a stylus-down and stylus-up message)

Sensor Settling Delay

Applicable Models: DCS1,DCS5

This command controls the initial "settling" delay which is used when a stylus is first placed on the sensing line. It's part of the error-correction / measurement verification scheme used by the measurement-board.

When a stylus is detected by the measurement-board, it will delay by a time constant multiplied by the value you use with this command before evaluating any values returned from the sensor. This eliminates erroneous values when a stylus approaches the sensor while moving erratically, such as can be the case on a moving boat.

Send to Board: &di,x#
Receive from Board: %di:x#

&di → Command Preamble
%di → Reply Preamble
, → Command separator
: → Reply separator
x → Integer number representing multiplier to use for "settling" delay, from zero to 20. Default is 1.
→ Query and response termination character

example:

&di,3#<CR>
%di:3#<CR>

Stylus Reading Max Deviation Allowed

Applicable Models: DCS1,DCS5

This command controls the amount that the stylus is allowed to move between each sensor reading, before the reading-set is thrown out and restarted. It's part of the error-correction / measurement verification scheme used by the measurement-board.

The measurement-board will sample the stylus position many times per second, attempting to collect a set number of "good" readings. If a reading of the sensor does not deviate more than the value set with this command from the previous reading, it's considered "good". After the set number of "good" readings have been collected, the measurement-board reports the average of those readings.

Send to Board: &dm,x#
Receive from Board: %dm:x#

&dm → Command Preamble
%dm → Reply Preamble
, → Command separator
: → Reply separator
x → Integer number representing max raw value deviation, from 1 to 100 (default is 6)
→ Query and response termination character

example:

&dm,15#<CR>
%dm:15#<CR>

Sensor Number of Readings

Applicable Models: DCS1,DCS5

This command sets the number of "good" sensor readings required during measurements in order for the measurement-board to return a value to the host (as an average). It's part of the error-correction / measurement verification scheme used by the measurement-board.

The measurement-board will sample the stylus position many times per second, attempting to collect a set number of "good" readings. If a reading of the sensor does not deviate more than the value set with this command from the previous reading, it's considered "good". After the set number of "good" readings have been collected, the measurement-board reports the average of those readings.

Send to Board: &dn,x#
Receive from Board: %dn:x#

&dn → Command Preamble
%dn → Reply Preamble
, → Command separator
: → Reply separator
x → Integer number representing number of sequential required in order to return a value to the host (default is 5)
→ Query and response termination character

example:

&dn,10#<CR>
%dn:10#<CR>

Check Cal State

Applicable Models: DCS1,DCS5

Send to Board: &u#
Receive from Board:&u:0# for not calibrated , &u:1# for calibrated
---

Specify mm values for Calibration Points

Applicable Models: All boards

Calibration involves establishing two calibration points of known distance apart by defining the locations and the "raw values" that are returned from the sensor for those respective locations. This command will establish the intended location of your calibration points in mm. If you are using your first calibration point as the zero line of your board, then Cal Pt 1 will use a value of 0 for v.

Send to Board: &1mm,v# or &2mm,v#
Receive from Board: Recognized &1mm(or &2mm),v#<*CR*> Android specified cal_pt_1 (or cal_pt_2) as v <*CR*>
---
cal_pt_1 → Calibration Point 1

cal_pt_2 → Calibration Point 2
v → Location of calibration point in mm.
, → Separator between values.
<CR> → Carriage Return separates the returned lines.

example1:

&1mm,0#<CR>

Recognized &1mm,0#<CR>

example2:

&2mm,375#<CR>

Recognized &2mm,375#<CR>

Take Reading for a Calibration Point

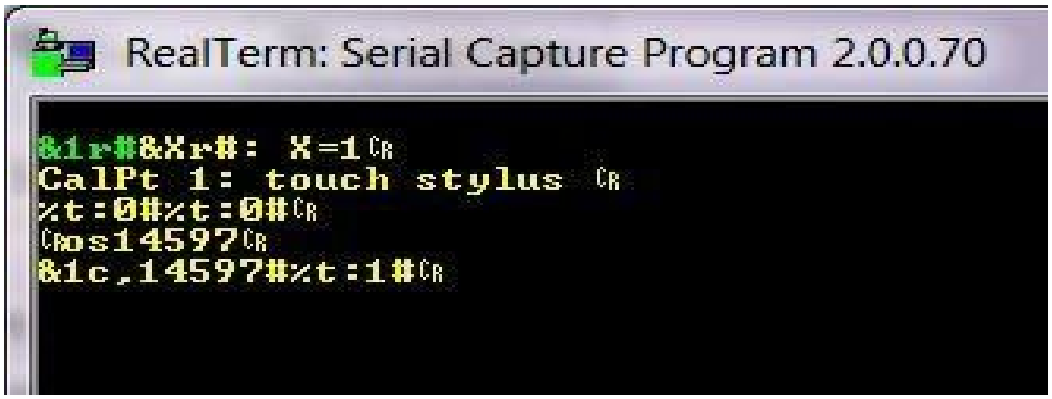
Applicable Models: All boards

This command will request the "raw value" sensor reading for a calibration point. After sending this command, the board will wait for a touch of the stylus on the board. The host application should prompt the user to place the stylus down at the location of Calibration point N (either 1 or 2) established from the **Specify Calibration Point N Location in mm** command, described above. The user holds the stylus down at the corresponding mm location on the board until the sensor-reading value is accepted. Since a lot of error correction and double-checking is occurring during this process, the board will be slow to respond. Your application should instruct the user to continue holding the stylus steady for several seconds and tell the user they should remove the stylus after the application has received a message from the board.

Send to Board: &1r# OR &2r#
Receive from Board: *&Xr#: X=1<CR> CalPt 1: touch stylus <CR> OR X=2<CR> CalPt 2: touch stylus <CR>*

&Xr# → Preamble
: → Separator between preamble and value returned
X → Calibration Point
X = 1 OR 2 → Calibration Point 1 or 2
<CR> → Carriage Return character

*An example of this command being used is shown in the terminal picture below. **%t:0#** is the board telling us that the stylus has touched down on the sensor (ignore the double %t message). In the example below, **14597** corresponds to the value that was received from the sensor at the location that the stylus was placed down. The final line **&1c, 14597#%t:1#<CR>** is stating that the first calibration point has been assigned a value of 14597.*



The response %t is the board receiving the signal that the stylus has touched down on the sensor. The final line is stating that the second calibration point has been assigned a value. Now that both cal points have values assigned, the calibration is finalized. See "Calibration Example", toward the end of this document for more information.

Command: Clear Calibration Data

Applicable Models: All boards

Send to Board: **&ca#**
Receive from Board: **CalMode <*CR*> Cleared working set calibration information <*CR*>**

CalMode → Calibration Mode
<CR> → Carriage Return separates returned lines
Explanation: This command will clear the calibration data currently on the board. This command is useful for when a user wants to recalibrate the board. Note that calibration data is stored on the measurement-board, but could be stored by the host and re-established on the board using the "Restore Calibration Data" command, below.

Command: Restore Calibration Data

Applicable Models: All boards

Use this command to restore calibration based on known calibration points and their respective sensor values. If the host application has stored the calibration values and it is assured that the board does not need to re-acquire the values through stylus placements, the host can recalibrate the board without going through the entire calibration routine again.

The measurement-board will store calibration data on-board, such that restoring calibration data to the board via this method isn't required for normal use-cases. This command serves two primary purposes: shifting the calibration some known distance for special functions (and then back to original), and as an alternate method for achieving calibration wherein the calibration data is stored by the host and then transmitted to the board in a single command.

Send to Board: **&cr,m1,m2,raw1,raw2#**
Receive from Board:
Cal restored: calPT1 =m1 mm, calPt2 = m2 mm, raw1=raw1, raw2=raw2 <CR>
Calibrated! Alpha = ###, beta =-###, invAlpha = ### <CR>
raw1 raw1<CR>
raw2 raw2<CR>
cal_point_1_mm m1<CR>
cal_point2_mm m2<CR>
<CR>NotOK 0<CR>
-----
calPT1 → Calibration Point 1
calPT2 → Calibration Point 2
, → Separator between values
raw1 → Raw value from sensor at Calibration Point 1
raw2 → Raw value from sensor at Calibration Point 2
Alpha → This is a value which is calculated internally from the set calibration points and raw values.

beta → This is a value which is calculated internally from the set calibration points and raw values
invAlpha → This is the inverse of Alpha.
notOK → An inverse Acknowledge flag wherein zero indicates a successful restore.

Backlighting

Applicable Models: DCS5 XT

The DCS5 XT offers back-lit keys for night work. Use this command to set the backlighting level. In addition, an Auto setting, along with an Auto-response sensitivity setting, will cause the XT control box to adjust the backlighting level automatically according to ambient light levels.

Set Backlighting Level

Send to Board: &o,x#
Receive from Board: %o:x#

&o → Command Preamble
%o → Reply Preamble
, → Command separator
: → Reply separator
x → Integer number from 0-95 wherein 0 is backlighting off and 95 is backlighting full on. (default is zero)
→ Query and response termination character

example:

&o,75#<CR>
%o:75#<CR>

Set Backlighting Sensitivity (for Auto Mode)

The backlighting sensitivity controls how quickly the backlighting will turn with an increase to ambient light. A higher setting will turn on the backlighting earlier and with more intensity at higher ambient light levels.

Send to Board: &os,x#
Receive from Board: %os:x#

&os → Command Preamble
%os → Reply Preamble
, → Command separator

: → Reply separator
x → Integer number from 0-7 (default is zero)
→ Query and response termination character

example:

&os,4#<CR>
%os:4#<CR>

Set Backlighting Auto Mode

The backlighting auto mode will automatically adjust the backlighting level according to the ambient light level and the Backlighting Sensitivity setting. The update frequency will vary from 3 to 10 seconds.

Send to Board: &oa,x#
Receive from Board: %oa:x#

&os → Command Preamble
%os → Reply Preamble
, → Command separator
: → Reply separator
x → Integer number of 0 or 1, wherein 0 is Auto=off and 1 is Auto=on (default is zero/off)
→ Query and response termination character

example:

&oa,1#<CR>
%oa:1#<CR>

Command: Column List

Applicable Models: 10MF1

Send to Board: c#
Receive from Board: %c,col1...coln,#

%c → Preamble
col1 → Column name in first position
coln → Column name in nth position
, → Separator between column names
#→ Response termination character
Explanation: This command will return a list of all of the pre-programmed field names on the board. Note the final ,

(comma) before the termination character. Every field name, including the last, is followed by a "," (comma). When data is off-loaded from the board in CSV format, the field names will constitute the first row of data.

Example:
c#<CR>
%c,length,weight,species-code,location-code,pan,vessel,trip,time,date,#<CR><LF>

Command: Dump Data

Applicable Models: 10MF1

Send to Board: **d#**
Receive from Board: **%d:1,r1f1,r1f2,r1f3...<CR>2,r2f1,r2f2,...,<CR>...N,rNf1,...,<CR>#<CR>**

%d → Preamble
: → Separator between preamble and value returned
1 → Record 1
, → Separator between values
r1f1 → Record 1 Field 1
r1f2 → Record 1 Field 2
... etc, for as many record fields as exist
<CR> → Carriage Return separating Record numbers
rNfN → Record N Field N
<CR>#<CR> → Response termination character
Explanation: This command will dump all of the fish record data saved onto the board. The first field of each record is the record number, starting at 1 and incrementing for each successive record. Note that fields within each record are comma separated.

Example:

d#<CR>
%d:1,length,weight,species-code,location-code,pan,vessel,trip,time,date,2,length,weight,species-code,location-code,pan,vessel,trip,time,date,3,length,weight,species-code,location-code,pan,vessel,trip,time,date,<CR>#<CR><LF>
(3 records)

Unsolicited Messages from Board to Host

These messages are sent from the board to the host periodically without a request from the host.

Stylus Down/Up Indicator

Applicable Models: DCS1,DCS5

This message is sent from the board upon stylus status changes (status = up or down).

Receive from Board: %t,v#
Response from Host: None/ignored
%t → Preamble
, → Separator between values
v → If v is 0, then the board is saying that the stylus is "down" on the sensor. If v is 1, then the board is saying that the stylus is "up" off of the sensor
→ Message termination character

example:
When a user lifts the stylus off of the board, the value for v will be set to 1 and a message will be sent, as follows:

%t,1#

Sensor Position (Length) Readout in mm

Applicable Models: All boards

This message contains the length readout in mm where the stylus is placed along the sensor line. Calibration, Delays, and multiple samples are applied before the reading is taken and the message is sent. This message is enclosed between **%t** messages, if that function is enabled.

Receive From Board: %l,v#
Response from Host: None/ignored
%l → Preamble
, → Separator between preamble and value
v → Integer representing the distance from zero that the stylus was touched to the board, in millimeters.
→ Message termination character.

examples:

%t,0#%l,265#%t,1# (with stylus up/down messaging enabled)
%l,265# (with stylus up/down messaging disabled)

Swipes

Applicable Models: DCS1,DCS5, 10MF1 with specific firmware

Stylus swipes can be used to enter different modes, depending on start location and length of swipe, as examples. As with many other messages, this one may be sandwiched between %t status messages.

This message will relay the direction and length of a swipe of the stylus . If v is positive, a rightward swipe (from left to right) has occurred. If v is negative, a leftward swipe (from right to left) has occurred.

In the case of Right-Swipes (left to right), the %s message will be followed by a %l message, to indicate where the swipe started. The start, end and length of right-swipes can be used to direct custom actions. In DCSLinkstream, the start locations of right-swipes are used to change between various data-entry modes, according to graphics printed on the measurement-board surface.

Length of left-swipes could also be interpreted into various actions.

Receive From Board: %s,v# (followed by %l,w# in the case of right-swipes)
Response from Host: None/ignored
%s → Preamble
, → Separator between preamble and value
v → Length of the swipe in millimeters.
→ Message termination character.

example1:

%s,-100#

would indicate a 100mm swipe to the left while stylus up/down notifications are turned off.

example2:

%t,0#%s,150#%l,50%t,1#

would indicate that the user swiped to the right with a swipe-length of 150mm, starting at the 50mm measurement line.

Faceplate Key Pushes

Applicable Models: DCS5 XT, DCS5 Micro

The DCS5 XT and DCS5 Micro have some keys available on the faceplates. These can be mapped to functions within your application and custom graphics to match can be placed on the faceplates. In the case of the XT, the keys are back-lit for night work.

XT → 32 keys
Micro → 4 keys

Receive From Board: %d,xx#
Response from Host: None/ignored
%d → Preamble
, → Separator between preamble and value
xx → 2 character integer from 00 to 31 (XT) or from 00 to 03 (Micro)
 # → Message termination character.

examples:

%t,0#%d,01#%t,1# (with stylus up/down messaging enabled)
%d,31# (with stylus up/down messaging disabled)

Length Measurements from 10MF1 to Host

In most 10MF1 firmware versions, immediately after each record is completed on the board, the record is transmitted via Bluetooth and USB to the host.

The record format is:

recordNum,field1,field2,field3,...,checksum,@<CR>
recordNum → Integer that represents this record's place in the storage on the fishboard starting at 1
fieldN → Data field for that particular record.
, → Separator between field values.
checksum → Helps detect transmission errors. Can be ignored

The data fields will be in the same order as in the Dump Data command. The checksum is calculated over the entire string, and is there to help detect transmission errors, but can also be ignored by the host. No response is expected.

Checksum details

The checksum algorithm consists of these steps:
1) Initialize the checksum to 0

2) For each character in the string (not including the newline at the end, but including the commas between fields), logical wrap the current checksum left one bit (wraps the MSB to the LSB, so bits are not lost)

3) Add the value of the current character to the checksum.

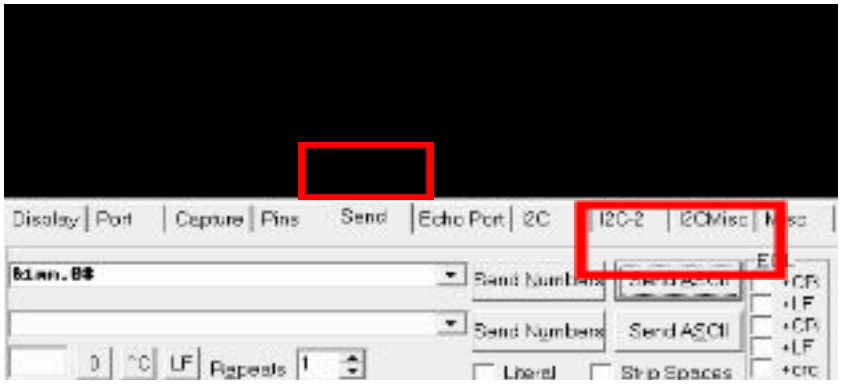
For the echo of the records on Bluetooth as they are entered, the checksum is an additional field at the end of the record, and is terminated by an "**@**" character. For example, an echoed record with checksum at the end looks like:

00047,000000,5,AEG,56,,10/30/77 10:36,00,Not set,,AAC7340B**@**

Calibration Example

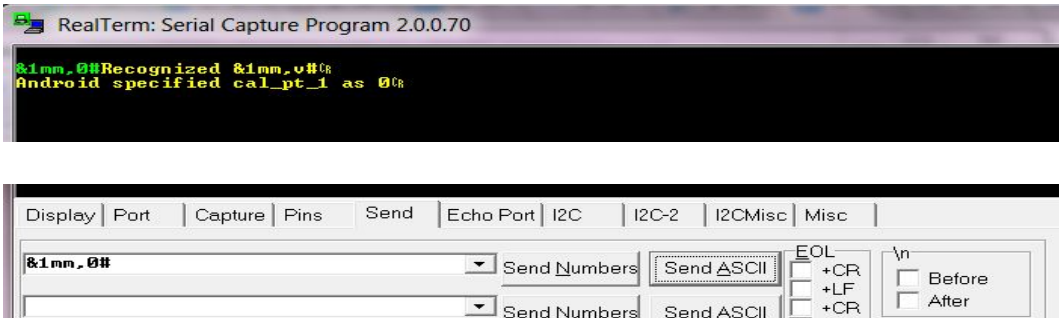
This section will walk you through using a set of commands to calibrate a Big Fin data-entry board, using the calibration-related commands found above. These commands will specify the two calibration points in millimeters that you will use to calibrate the board. After specifying the real lengths of the calibration points in mm, you will send a command to specify the raw calibration values which are then inputted by placing the stylus down on the board at the specified lengths. First, we will separate the commands so that it is easier to follow along. Then, we will string commands together to show that multiple commands can be sent at once. We're using a terminal emulator program called RealTerm, which is available for free on the internet. For each command line that we send through the terminal, we will use the "Send ASCII" button outlined with red in the picture below, accessed via the 'Send' tab.

Note: *For some (typically older) firmware builds, you will not receive all of the command responses seen in this example if you are connecting a terminal over Bluetooth. In this example, the board is connected over USB so that all command responses are received and seen from the board. The (more recent) DCS5 firmware builds tend to reply the same over BT and USB while older firmware builds may only reply over USB. Contact us if this creates an issue for you.*



Specify Calibration Point 1

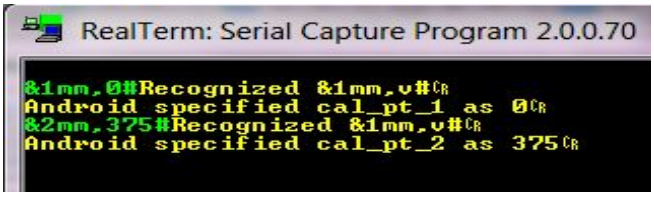
We must set the first calibration point in order to begin the calibration procedure. This is accomplished using the "**Specify Calibration Point 1 Location in mm**" command in the picture below:



The green text (shown above) "**&1mm,0#**" is the command sent by the user and the yellow text shows the board's response to the command. In this example, we are specifying that the first calibration point is at the 0 mm measurement line (at the nose-stop). If desired, the first calibration point can be set as a different value than zero. The board receives this command and relays that calibration point 1 is now set as 0 mm.

Specify Calibration Point 2

Setting the second calibration point is accomplished in much the same manner as the first.



In this example, we specify 375mm for the second calibration point since that's the length of the "Calibration Stick" which is supplied with the measurement-board kits. Another value can be used if another object of known length is used in-place of the Calibration Stick. The board recognizes this command and relays that calibration point 2 has been specified as 375.

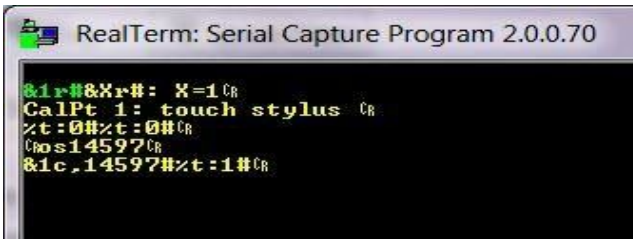
Note that the calibration numbers represent absolute measurement numbers from the measurement line. In other words, using our Calibration Stick of 375mm, we could accomplish the same calibration starting at 50mm by using:

%1mm,50#

%2mm,425#

Obtain Raw Value for Calibration Point 1

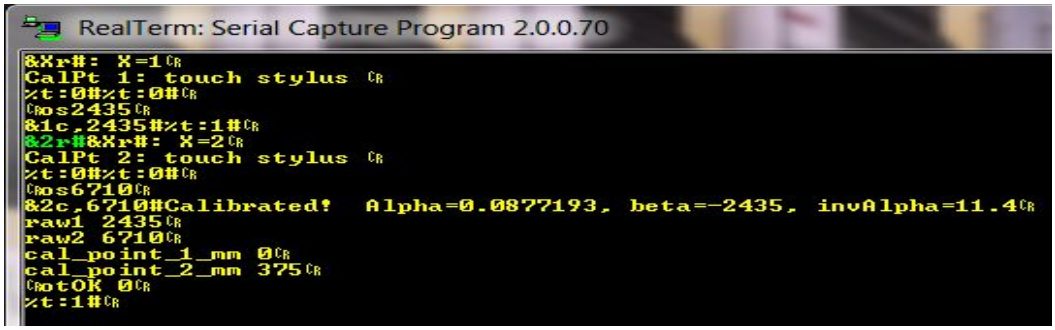
Now that both calibration points have been specified, we can now set the raw values for each point to complete the calibration sequence. First, we set the raw value for calibration point 1 (0 mm in this example). This is shown in the picture below.



We send the green **&1r#** command to tell the board to ask for the first raw value. The board will relay that you need to touch the stylus down on the calibration point 1 that we specified earlier. This is shown on the second line of the picture. At this point, we place the stylus down on the board and hold it at the 0 mm mark. Use the same reference point on the stylus for each calibration point. Typically, the tangential edge of the stylus is used as a reference point because that is the point that will be touching the nose-stop at 0 mm and this point will also end of fish when making measurements. The next 3 lines in yellow will then appear in the terminal which show the value of the raw point given from the sensor at that location.

Obtain Raw Value for Calibration Point 2

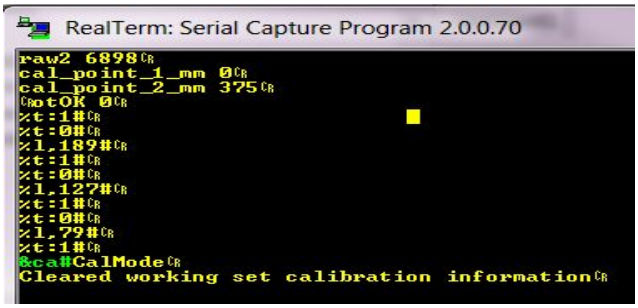
We must now complete the final step in the calibration procedure by setting the second raw value for calibration point 2. From earlier, we specified that calibration point 2 was at 375 mm.



The raw value can be set for calibration point 2 by sending the the **&2r#** command to the board shown in green. After sending this command, the board will prompt you to touch the stylus down at point 2. Now, place the stylus down at the 375 mm mark on the measurement-board and hold it using the same reference point on the stylus as used in the last step. The terminal will update and the board will convey that it has been calibrated. After this, the board will send both raw values along with the set calibration points back through the terminal along with some calculated values that you can ignore (Alpha, beta, invAlpha). You have now successfully calibrated your board.

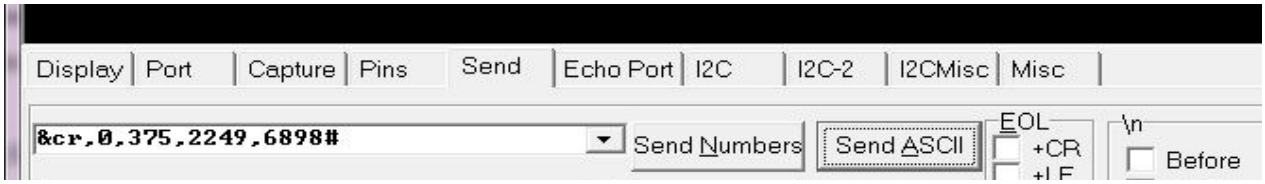
Clearing Calibration Data

If at any point you want to clear your calibration data, you can simply send the command &ca# and this will clear any calibration data that is currently on the board. This is shown in the picture below. Keep in mind that if you clear calibration data, the board will always send a value of 0 mm for length until it is calibrated once again.

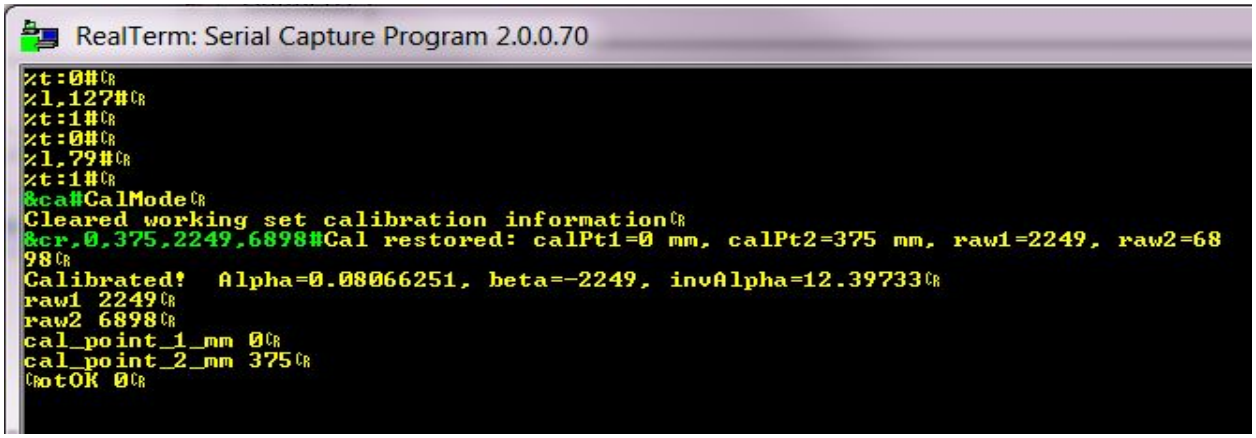


Restoring Calibration using Predetermined Raw Values

After we have calibrated the board once, we have a set of raw values that we can send to the board instead of re-establishing raw values with the stylus. This is useful during development and testing, and in other circumstances wherein a known-good calibration state has been pre-determined. For the following example, 2435 will be used for the raw value for calibration point 1 at 0mm. 6710 will be used for the raw value for calibration point 2 at 375mm. The calibration data is first cleared with the **&ca#** command.



After clearing the calibration data, you can send **&cr,0,375,2249,6898#** and the board will restore calibration data from the predetermined raw values associated with 0 mm and 375 mm lengths.


RealTerm: Serial Capture Program 2.0.0.70
%t:0#CR
%l,127#CR
%t:1#CR
%t:0#CR
%l,79#CR
%t:1#CR
&ca#CalModeCR
Cleared working set calibration informationCR
&cr,0,375,2249,6898#Cal restored: calPt1=0 mm, calPt2=375 mm, raw1=2249, raw2=68
98CR
Calibrated! Alpha=0.08066251, beta=-2249, invAlpha=12.39733CR
raw1 2249CR
raw2 6898CR
cal_point_1_mm 0CR
cal_point_2_mm 375CR
CRotOK 0CR

Inquiries

Questions or comments? Please contact us:

info@bigfinscientific.com or 512-808-0346 (8080-FIN).