Description	Technical specification for interface between the BCInline and other devices through a TCP socket connection		
Author	Mike Geren		
Distribution	Individuals implementing the socket communications.		
Doc. History	**Revision**	**Date**	**Change**
	1	2025-06-22	First release – based on Archer 8.16
	2	2026-01-05	Minor updates and clarifications.

1. Introduction

This specification describes a command/response communication channel for integrators or end customers to use for the purpose of automating the operation of a BCInline in a production or laboratory setting. It is expected that the BCInline local touchscreen is still utilized for configuration settings and many maintenance functions.

2. Structure of the protocol and document

There are two general types of communication commands, those which are used to control a normal production sequence, and those commands which are only used for troubleshooting or debugging purposes. In this document the two types will be separated but the commands are not limited to the functions described (so the troubleshooting type commands can be used in production if they prove to be useful).

ABBREVIATIONS:

- BCI - BCInline
- RD - Remote device. This is the device which is connected through the Ethernet connection to the BCI

KEY:

- Green items are the normal operation of the commands.
- Orange items indicate a possible path for a retry or automatic correction taking place.
- Red items indicate an exception has occurred which disrupts the normal operational flow.
- Blue items are cancellation commands which only need to be used if a long running command needs to be cancelled. Because the cancellation commands fall outside of the normal command sequence it is possible for the communications to overlap and extra responses be received by the RD. Care should be taken to provide a robust communications handling routine which can gracefully handle these types of situations.

3. Ethernet connection details

The BCI connects through a USB to Ethernet adaptor which gets the IP address from a DHCP server.

- The BCI accepts TCP commands on port 2222. This port can be changed if needed by using the MC command “tcpc:x”, contact Brighton Science for assistance.
- The data in each packet sent from the RD contains a single string using ASCII (UTF8) encoding. All commands sent from the RD are terminated with the '>' character followed by a CrLf pair (except for binary image data packets).

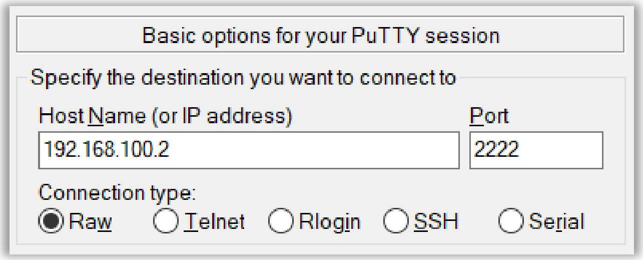
- The BCI will send reply TCP packets that are ASCII (UTF8) to the opened connection. Images are binary and packaged in PNG format. Some commands will echo the command immediately then return a second result packet when the request has been completed.
- Commands should only be issues when the BCI is in measurement mode, or in the main menu system. Commands should not be issued and may not be responded to when any popup message is present, the menu system is in any sub-menus (profiles, about, etc.) or any user interface sequence (cartridge change, manual performance check, etc.)
- This protocol is designed so the RD is the master which will initiate all communications. After a command initiation there is a sequence of commands between the RD and the BCI which should be completed before starting a new command sequence. Because the BCI does not initiate a communications sequence it is important that the RD regularly request a status message to determine if there are pending issues which need to be addressed such as needing service due to a cartridge being empty or recommended performance checks.
- The BCI will use the IP address assigned from a DHCP server on the network used, for a fixed IP address the DHCP server should be configured to always allocate a fixed IP address to the MAC address of the BCI. The BCI can initially be hooked up to a router or switch to initially determine the MAC address and most USB to Ethernet adaptors have the MAC address printed on the device. Options for DHCP servers include using a router with DHCP server capability, or PC options like http://www.dhcpserver.de/ which can be configured to give a consistent IP address to the BCI.
    - A sample configuration file for the DHCP server for windows application (V2.5.1 – “dhcpsrv.ini”) is shown below for reference. This file shows the RD Ethernet board configured to 192.168.100.1 and sets the BCI to always have an IP address of 192.168.100.2 (Change these addresses to match your specific configuration)

```
[General]
SUBNETMASK=255.255.255.0
LEASETIME=32000000 ; approx. 1 year
ROUTER_1=192.168.100.1 ; Network card to bind to the DHCP server
DNS_1=192.168.100.1 ; Network card to bind to the DHCP server

[Settings]
IPBIND_1=192.168.100.1 ; Network card to bind to the DHCP server
IgnoreUnknownClients=1 ; Ignore all other clients except those below

[00-50-B6-07-19-C9] ; BCInline USB to Ethernet adaptor MAC Address
IPADDR=192.168.100.2 ; IP address to use when opening the port on the BCI
```

- A terminal program can be used on a laptop for testing the connection. As an example, the application “Putty” can be used (https://www.putty.org/) with settings similar to the following (change address to match your specific configuration):



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.100.2

Port: 2222

Connection type:

● Raw ○ Telnet ○ Rlogin ○ SSH ○ Serial

4. Commands for normal production use

These commands alter the settings in the BCI or command it to take an action.
NOTE: All BCI responses will be terminated with CrLf except for binary image data (unless changed via GUI with admin access)

Perform an alignment by taking an image and returning target location

RD command	BCI responses	Description
		Normal sequence
Align> *Or* **AlignNP>**	**Align(a,b,c,d,e,f,g,h)>**	*NOTE: AlignNP> initiates a standard alignment without returning the image packet. All other responses are the same.* Successful alignment: Packet 1: a = X coordinate of the center of the circle. Result is a decimal (2 digits after decimal) location in pixels (0-511). b = Y coordinate of the center of the circle. Result is a decimal (2 digits after decimal) location in pixels (0-511). c = Area of circle. Result is an integer size in pixels. d = Size of the image in bytes (which is the size of the next packet if not the NP version). e = Number of outlier points f = Compactness value g = Timestamp of the inspection - “yyyy-mm-ddThh:mm:ss.nnn” h = Detection flags (result code indicating if the drop was automatically found to be good or defective) The possible codes are as follows: BD_OUTLIERS BD_COMPACTNESS BD_OUT_OF_FOCUS BD_DROP_TOO_SMALL GD *Example response (out of focus error):* Align(127.58,144.02,22951,284519,0,0.99,2018-05-09T15:03:52.879,BD_OUT_OF_FOCUS)> *Example response (acceptable result)* Align(256.37,280.99,23712,285723,0,1,2018-05-09T15:03:52.879,GD)>
	a	Packet 2 (only returned for an “Align>” command): a = Image from inspection. This will be a binary packet encoded in PNG image format. The image is 480x480 pixels ***NOTE: This packet will not be sent if the “AlignNP>” version of the command initiated the process.***

RD command	BCI responses	Description
Possible Failure responses		
Align> *Or* **AlignNP>**	**TM_ERROR_NOT_IN_PREVIEW>**	Failed alignment: The BCI is not in the measurement mode.
Align> *Or* **AlignNP>**	**ERROR_ALIGN>**	Failed alignment: Unable to locate and measure the alignment target

Prepare and initiate a cartridge change

RD command	BCI responses	Description
Normal sequence		
ChangeCartridge(a)>	**CC_SERIAL_OK>**	Starts the cartridge change process by validating the new cartridge serial number. The serial number is made up of 16 HEX digits. Dashes will be removed from the received string so embedded dashes are acceptable. Either upper or lower-case letters are acceptable. a = Cartridge serial number (16 HEX digits including 0-9 and A-F) printed on cartridge.
	CC_COMPLETE>	After the user has completed the change process the BCI will respond with a complete message. This message will not be sent if the serial number is invalid or the cartridge is empty or the user cancels the process (see errors below).
Invalid serial number entered		
ChangeCartridge(a)>	**CC_INVALID_SERIAL>**	The serial number entered is not valid. Verify serial number was entered correctly.
Cartridge has already been used and is empty		
ChangeCartridge(a)>	**CC_ERROR_EMPTY>**	The cartridge serial number is valid but has already been used and is empty. Use a fresh cartridge.
Cartridge change process was cancelled by the user on the screen		
ChangeCartridge(a)>	**CC_USER_CANCEL>**	During the cartridge change process, the operator has the opportunity to cancel on screen. This error indicates that has happened and so the change process has been terminated.

Start a continuous purge (Approximately 140 drops – takes a medium amount of time)

RD command	BCI responses	Description
Normal sequence		
ContinuousPurge>	**ContinuousPurge>**	Response is sent once purge is complete (medium amount of time)

Get current user drop count

RD command	BCI responses	Description
Normal sequence		
DropCount>	**DropCount(a,b)>**	Volume used for measurements out of the cartridge and the total volume the cartridge holds (microliters). a = Volume Used (float) b = Volume available (float) *Example response:* DropCount(12177.898,90000.0)> NOTE: A “standard” measurement drop is approximately 1.5 microliters

Get status of an input bit (0-3)

RD command	BCI responses	Description
Normal sequence		
GetInputPin(a)>	**GetInputPin(b,c)>**	Return the state of a physical input on the system. a = Pin number 0-3 of the input to be read The BCI response contains the actual read values: b = Pin number 0-3 of the input which was read c = Status of the input. Valid status responses are: HIGH = input was read and is in a high state LOW = input was read and is in a low state ERROR_PIN = input pin number is not valid (must be 0-3) ERROR_IO = I/O board not found

Get time and date of last performance check

RD command	BCI responses	Description
Normal sequence		
GetLastPCHK>	**GetLastPCHK(a)>**	This returns the date and time of the last passed performance check a = Date and time of last performance check in the format “yyyy-mm-ddThh:mm:ss.nnn” *Example response:* GetLastPCHK(04-02-2018T14:41:57.492)>

Get status of an output bit (0-3)

RD command	BCI responses	Description
		Normal sequence
GetOutputPin(a)>	**GetOutputPin (b,c)>**	Return the state of a physical output on the system. a = Pin number 0-3 of the output to be read The BCI response contains the actual read values: b = Pin number 0-3 of the output which was read c = Status of the input. Valid status responses are: HIGH = output was read and is in a high state LOW = output was read and is in a low state ERROR_PIN = input pin number is not valid (must be 0-3) ERROR_IO = I/O board not found

Get all the process monitors

RD command	SA responses	Description
GetProcessMonList>	**GetProcessMonList>** **(a1,a2,etc.)>**	RESPONSE: GetProcessMonList(Name1 :: ID1, Name2 :: ID2, Name3 :: ID3)> • The response is always enclosed in GetProcessMonList( )>. • Inside the parentheses, each Workflow is represented by Name :: ID, separated by commas. Example: GetProcessMonList(20241008.2 test 3 :: 631c20c0-1e61-4568-84bc-eea6eb53ce04, Make this name really long to test the overlapping bug in the software ticket for Alan :: e3ff8ee4-1c52-41de-9cb7-a8d8991886ae, SOF-3479 Process Test :: 683d77e3-b5d0-4e9f-af25-178ddeb613da)> PARSING TIPS: 1. Remove the leading GetProcessMonList(and trailing)>. 2. Split on commas to isolate each Name :: ID pair. 3. For each element, split on :: to separate the name and the workflow ID.

Download update from the web

RD command	SA responses	Description
GetProcessMonData(a)>	**GetProcessMonData(a,b,c,d,e,f,g,h)>**	Where: • a = Program Name • b = Number of Measurements if no parts, else -1 • c = Facilities array [{facilityId=...,name=...,facilityType=...,productionLines={id|name,...}}, ...] • d = Control Points array [{id=...,name=...,condition=...,customCondition=...,targetContactAngleType=...,targetContactAngleMin=...,targetContactAngleMax=...}, ...] • e = Profile used id • f = Parts array [{id=...,name=...,partNumber=...,numberOfMeasurements=...,hasImage=true|false}, ...] • g = metadataLabel (string, may be empty) • h = regex (string, may be empty) REQUEST: • GetProcessMonData(some-workflow-id from GetProcessMonList)> PARSING TIPS: • Remove GetProcessMonData(and the trailing)>. • Identify the six top-level segments separated by commas. • Inside the bracketed arrays ([...]), each facility, control point, or part is enclosed in braces {...} with its fields, and some fields (like productionLines) may include nested braces {...}. For example: • Facilities: [{facilityId=..., name=..., productionLines={...}}, ...] • Control points: [{id=..., name=..., condition=...}, ...] • Parts: [{id=..., name=..., numberOfMeasurements=..., hasImage=...}, ...] • Avoid a naive "split by comma" on these bracketed sections; commas appear inside {} and []. You should

		parse the brackets and braces carefully (like lightweight JSON parsing) or use a method to isolate the six top-level segments first. ADDITIONAL NOTES • All commands terminate with a > at the end of the line. • Be mindful that commas, curly braces, and square brackets occur in nested data, so parsing must handle that structure correctly. • Fields can be empty (e.g. customCondition=) or partial (e.g. targetContactAngleNumber=). Handle them gracefully in your parser. • If the workflow or its parts/facilities are missing or invalid, you may receive an error response like `GetProcessMonDataError>` instead of the full structure.

Get a list of surface profiles available on the BCI

RD command	BCI responses	Description
		Normal sequence
GetProfiles>	**GetProfiles(a,b,…)>**	Returns the names of the profiles available on the BCI. Each profile will be separated by a comma. The profile names are not guaranteed to be returned in any specific order. a = Name of the first profile. b = Name of the second profile. … = Rest of the profile names separated by commas

Get the current image

RD command	BCI responses	Description
		Normal sequence
GetScreen>	**GetScreen(a)>**	This can be used to implement a live video stream when the BCI is ready for a measurement by repeatedly sending this command. Commands cannot overlap with the responses (wait for one command to finish before starting another) so do not implement this call on a simple timer unless there is a check for the last call to be completed. Packet 1: a = Size of the image in bytes (which is the size of the next packet).
	b	Packet 2: b = Current image. This will be a binary packet encoded in PNG image format. The image is 480x480 pixels.

Request the status of the BCI

RD command	BCI responses	Description
		Normal sequence
GetStatus>	**GetStatus(a,b,c,d)>**	Provides a mechanism to gather information about faults which have occurred and general status. This message can also be used to prove connectivity has not been lost by polling on a regular basis. It is recommended that this be used regfularly as both a test for fault conditions and as a keep alive signal to check for proper connectivity. This command should not be sent during another command sequence. a = Free space available on the BCI. This will be a value between 0 and 100 and represents the percentage of available free space. This can be used to tell if the "DeleteResults>" command should be issued to free space. b = Cartridge status. Possible codes are as follows: CART_OK CART_EMPTY CART_PURGE_NEEDED c = Performance check status. Possible codes are as follows: PCHECK_OK PCHECK_DUE d = Pump status PUMP_OK PUMP_TIMEOUT *Example response:* GetStatus(91,CART_OK,PCHECK_OK,PUMP_OK)>

Change the BCI into measurement mode where it is ready to accept measurement commands

RD command	BCI responses	Description
Normal sequence when BCI is already in measurement mode or at the main menu		
GoToMeasurement>	**GoToMeasurement>**	The BCI will exit any menus which the user has placed the BCI into using the touchscreen. The command will return once the system is in measurement mode, but may not be fully up to pressure. If a measurement is taken before full pressure is reached, the measure command will respond properly with a pressure error. NOTE: This is not meant to be used as a cancel for other modes, but will act as one if the BCI is in barcode scanning, performance check, or factory purge. When used as a cancel there will be a delay during the cancel operation before the GoToMeasurement> response is sent. This command will not exit from modal popup windows or menus. NOTE: If the login screen is showing when this command is issued, the default user will be logged in and the system we move to the measurement screen (functionally the same as setting the auto login feature)

Change active surface profile on the BCI

RD command	BCI responses	Description
Normal sequence		
LoadProfile(a)>	**LoadProfile>**	Changes the active profile to the profile specified by “a” in the RD command. a = Name of the profile to switch to. NOTE: The profile names are case sensitive and must match exactly.
Failure response		
LoadProfile(a)>	**LoadProfileNotFound>**	If the active profile cannot be changed to “a” then this response is given instead of the response above. The profile is not changed.
LoadProfile(a)>	**LoadProfileDynamicDetectionLocked>**	If the device does not have Dynamic Detection enabled, but the profile “a” has Dynamic Detection enabled, then the profile will not be loaded. The profile is not changed.

Take a measurement and return the results

RD command	BCI responses	Description
Normal sequence		
Measure> *or* **MeasureNP>**	**Measure(a,b,c,d,e,f,g,h,i)>**	*NOTE: MeasureNP> initiates a standard measurement without returning the image packet. All other responses are the same.* Successful measurement Packet 1: The results of the measurement which consists of the following: a = Measured contact angle (This will be set to 999 for measurement failures) b = Outlier points (number of outliers if turned on, if turned off this will be 0) c = Compactness value of the drop (decimal number <=1) d = Drop center distance (integer distance from cross hair location to drop found location) e = Timestamp (internal time of measurement) - “yyyy-mm-ddThh:mm:ss.nnn” f = Drop count (number of measurement drops used out of the cartridge) g = Detection flags (result code indicating if the drop was automatically found to be good or defective) The possible codes are as follows: BD_COMPACTNESS BD_OUT_OF_FOCUS BD_DROP_TOO_SMALL BD_BAD_DISPENSE BD_TOO_MANY_SATELLITES BD_SATELLITES_ML GD h = Pass Fail Flag (status of the inspection using the internal pass/fail limits. The code is a single character.) The possible codes are as follows: P (pass) F (fail) S (fail, surfactant detected) N (none, limits set to 0-180 which disables limits) i = Size of the image in bytes (which is the size of the next packet if not the NP version). *Example response (Failed detection due to outliers and failed measurement):* Measure(999,40,0.93,62,2018-05-03T15:32:05.327,251,BD_OUTLIERS,F,153815)> *Example response (Good detection but failed measurement):* Measure(58,0,0.94,9,2018-05-03T15:31:49.972,250,GD,F,160560)> *Example response (Good detection passing measurement):* Measure(52,6,0.96,9,2018-05-03T15:40:31.011,256,GD,P,161005)>

RD command	BCI responses	Description
	a	Packet 2 (only returned for a “Measure>” command): a = Image from inspection. This will be a binary packet encoded in PNG image format. The image is 480x480 pixels. ***NOTE: This packet will not be sent if the “MeasureNP>” version of the command initiated the process.***
Possible Failure responses		
Measure> *or* **MeasureNP>**	**TM_ERROR_PUMP_RAMPING>**	Failed measurement: System has not reached pressure yet, measurement was taken too soon after entering measurement mode or system is taking a long time to reach pressure. This response happens if a measure is requested while the BCI is displaying the black pressurizing screen.
Measure> *or* **MeasureNP>**	**TM_ERROR_PRESSURE:XXXX>**	Failed measurement: The system is not at the proper pressure. The “XXXX” is replaced with the current pressure value. *Example response:* TM_ERROR_PRESSURE: +0768>
Measure> *or* **MeasureNP>**	**TM_ERROR_NOT_IN_PREVIEW>**	Failed measurement: The BCI is not in the measurement mode.
Measure> *or* **MeasureNP>**	**TM_ERROR_OVER_DROP_COUNT>**	Failed measurement: The BCI is unable to take a measurement because the cartridge is out of liquid. The cartridge must be replaced before measurements can be taken.
Measure> *or* **MeasureNP>**	**TM_ERROR_CART_PURGE_NEEDED>**	Failed measurement: The BCI is unable to take a measurement because the system requires purging (typically after a cartridge change). Use ContinuousPurge> before attempting again.
Measure> *or* **MeasureNP>**	**TM_ERROR_DB_TRANSFER>**	Failed measurement: The BCI is unable to take a measurement because the system is currently transferring the result database to the RD.

Take a measurement (broken into discrete steps) and return the results

RD command	BCI responses	Description
Normal sequence		
MeasureDiscreteStart>	**SubstrateCaptured>**	Starts the sequence for a measurement which is broken up into individual steps. If the timing is held the same as a standard measurement then this command will achieve the same results as a standard "Measure>" or "MeasureNP>" command while allowing an automatic system to take extra actions in-between steps. Response indicates that the substrate image has been taken and system is ready for movement if needed once response is received. ***NOTE: The normal imaging and dispensing cycle includes specific delays and pauses between steps setup in the software. When using this feature the timing and delays are no longer controlled by the BCInline and must be properly controlled by the Remote Device.***
MeasureDropDispense>	**DropDispensed>**	Initiate a drop dispense using the appropriate settings for the current profile. Response once drop has been dispensed and the system is ready for movement if needed.
MeasureInspect> *or* **MeasureInspectNP>**	**DropCaptured>**	Trigger the inspection of the drop which was just dispensed. The response indicates the drop image has been taken and the system is ready for movement if needed. *MeasureInspect> results in 3 response packets while MeasureInspectNP> results in only 2 packets.* Packet 1: drop image has been captured and system is ready for movement if needed. ***NOTE: While using Surfactant detection mode this command will initiate two images being taken with a delay between them. The delay can be set in the Surfactant detection settings of the BCI.*** ***NOTE: MeasureInspectNP> initiates a measurement without returning the image packet. All other responses are the same.***

RD command	BCI responses	Description
	Measure(a,b,c,d,e,f,g,h,i)>	Successful measurement Packet 2: The results of the measurement which consists of the following: a = Measured contact angle (This will be set to 999 for measurement failures) b = Outlier points (number of outliers if turned on, if turned off this will be 0) c = Compactness value of the drop (decimal number <=1) d = Drop center distance (integer distance from cross hair location to drop found location) e = Timestamp (internal time of measurement) - "yyyy-mm-ddThh:mm:ss.nnn" f = Drop count (number of measurement drops used out of the cartridge) g = Detection flags (result code indicating if the drop was automatically found to be good or defective) The possible codes are as follows: BD_COMPACTNESS BD_OUT_OF_FOCUS BD_DROP_TOO_SMALL BD_BAD_DISPENSE BD_TOO_MANY_SATELLITES BD_SATELLITES_ML GD h = Pass Fail Flag (status of the inspection using the internal pass/fail limits. The code is a single character.) The possible codes are as follows: P (pass) F (fail) S (fail, surfactant detected) N (none, limits set to 0-180 which disables limits) i = Size of the image in bytes (which is the size of the next packet if not the NP version). *Example response (Failed detection due to outliers and failed measurement):* Measure(999,40,0.93,62,2018-05-03T15:32:05.123,251,BD_OUTLIERS,F,153815)> *Example response (Good detection but failed measurement):* Measure(58,0,0.94,9,2018-05-03T15:31:49.937,250,GD,F,160560)> *Example response (Good detection passing measurement):* Measure(52,6,0.96,9,2018-05-03T15:40:31.041,256,GD,P,161005)>

RD command	BCI responses	Description
	a	Packet 3 (only returned for a "MeasureInspect>" command): a = Image from inspection. This will be a binary packet encoded in PNG image format. The image is 480x480 pixels. ***NOTE: This packet will not be sent if the "MeasureInspectNP>" version of the command initiated the process.***
Possible Failure responses		
MeasureInspect > *or* **MeasureInspectNP >**	**TM_ERROR_PUMP_RAMPING>**	Failed measurement: System has not reached pressure yet, measurement was taken too soon after entering measurement mode or system is taking a long time to reach pressure. This response happens if a measure is requested while the BCI is displaying the black pressurizing screen.
MeasureInspect > *or* **MeasureInspectNP >**	**TM_ERROR_PRESSURE:XXXX>**	Failed measurement: The system is not at the proper pressure. The "XXXX" is replaced with the current pressure value.
MeasureInspect > *or* **MeasureInspectNP >**	**TM_ERROR_NOT_IN_PREVIEW>**	Failed measurement: The BCI is not in the measurement mode.
MeasureInspect > *or* **MeasureInspectNP >**	**TM_ERROR_OVER_DROP_COUNT>**	Failed measurement: The BCI is unable to take a measurement because the cartridge is out of liquid. The cartridge must be replaced before measurements can be taken.
MeasureInspect > *or* **MeasureInspectNP >**	**TM_ERROR_CART_PURGE_NEEDED>**	Failed measurement: The BCI is unable to take a measurement because the system requires purging (typically after a cartridge change). Use ContinuousPurge> before attempting again.

Measurement with position and other information from the robot

RD command	SA responses	Description
MeasureProcess (a,b,c,d,e,f,g,h,i,j)> *or* **MeasureProcessNP (a,b,c,d,e,f,g,h,l,j)>**	**MeasureProcess (r,s,t,u,v,w,x,y,z)>**	*NOTE: MeasureProcessNP> initiates a standard measurement without returning the image packet. All other responses are the same.* The measurement data from the robot system consists of the following: a = Program name b = Part ID (barcode or manually entered value) c = Measurement position/location d = X Coordinate (in mm) e = Y Coordinate (in mm) f = Z Coordinate (in mm) g = X Roll (in deg; 0 if 3 axis) h = Y Roll (in deg; 0 if 3 axis) i = Z Roll (in deg; 0 if 3 axis) J = Metadata Successful measurement Packet 1: The results of the measurement which consists of the following: r = Measured contact angle (This will be set to 999 for measurement failures) s = Outlier points (number of outliers if turned on, if turned off this will be 0) t = Compactness value of the drop (decimal number <=1) u = Drop center distance (integer distance from cross hair location to drop found location) v = Timestamp (internal time of measurement) - “yyyy-mm-ddThh:mm:ss.nnn” w = Drop count (number of measurement drops used out of the cartridge) x = Detection flags (result code indicating if the drop was automatically found to be good or defective) The possible codes are as follows: BD_COMPACTNESS BD_OUT_OF_FOCUS BD_DROP_TOO_SMALL GD y = Pass Fail Flag (status of the inspection using the internal pass/fail limits. The code is a single character.) The possible codes are as follows: P (pass)

		F (fail) S (fail, surfactant detected) N (none, limits set to 0-180 which disables limits) z = Size of the image in bytes (which is the size of the next packet if not the NP version). *Example response (Failed detection due to outliers and failed measurement):* MeasurePos(999,40,0.93,62,2018-05-03T15:32:05.327,251,BD_OUTLIERS,F,153815)> *Example response (Good detection but failed measurement):* MeasurePos(58,0,0.94,9,2018-05-03T15:31:49.972,250,GD,F,160560)>
	a	Packet 2 (only returned for a "Measure>" command): a = Image from inspection. This will be a binary packet encoded in PNG image format. The image is 480x480 pixels. ***NOTE: This packet will not be sent if the "MeasureNP>" version of the command initiated the process.***
MeasureProcess (a,b,c,d,e,f,g,h,i,j)> *or* **MeasureProcessNP (a,b,c,d,e,f,g,h,i,j)>**	**TM_ERROR _PUMP_RAMPING>**	Failed measurement: System has not reached pressure yet, measurement was taken too soon after entering measurement mode or system is taking a long time to reach pressure. This response happens if a measure is requested while the SA is displaying the black pressurizing screen.
MeasureProcess (a,b,c,d,e,f,g,h,i,j)> *or* **MeasureProcessNP (a,b,c,d,e,f,g,h,i,j)>**	**TM_ERROR _PRESSURE:XXXX>**	Failed measurement: The system is not at the proper pressure. The "XXXX" is replaced with the current pressure value.
MeasureProcess (a,b,c,d,e,f,g,h,i,j)> *or* **MeasureProcessNP (a,b,c,d,e,f,g,h,i,j)>**	**TM_ERROR _NOT_IN_PREVIEW>**	Failed measurement: The SA is not in the measurement mode.
MeasureProcess (a,b,c,d,e,f,g,h,i,j)> *or* **MeasureProcessNP (a,b,c,d,e,f,g,h,i,j)>**	**TM_ERROR _OVER_DROP _COUNT>**	Failed measurement: The SA is unable to take a measurement because the cartridge is out of liquid. The cartridge must be replaced before measurements can be taken.
MeasureProcess (a,b,c,d,e,f,g,h,i,j)> *or* **MeasureProcessNP (a,b,c,d,e,f,g,h,i,j)>**	**TM_ERROR _CART_PURGE _NEEDED>**	Failed measurement: The SA is unable to take a measurement because the system requires purging (typically after a cartridge change). Use ContinuousPurge> before attempting again.
MeasureProcess (a,b,c,d,e,f,g,h,i,j)>	**TM_ERROR _DB_TRANSFER>**	Failed measurement: The SA is unable to take a measurement because the system is currently transferring the result database to the RD.

or **MeasureProcessNP (a,b,c,d,e,f,g,h,i,j)>**		
MeasureProcess (a,b,c,d,e,f,g,h,i,j)> *or* **MeasureProcessNP (a,b,c,d,e,f,g,h,i,j)>**	**WrongProfileLoaded>**	returned when the incoming `surfaceProfileId` does not match the currently loaded profile

Discrete Measurement with position and other information from the robot

RD command	SA responses	Description
MeasureDiscreteStart>	**SubstrateCaptured>**	Starts the sequence for a measurement which is broken up into individual steps. If the timing is held the same as a standard measurement then this command will achieve the same results as a standard “Measure>” or “MeasureNP>” command while allowing an automatic system to take extra actions in-between steps. Response indicates that the substrate image has been taken and system is ready for movement if needed once response is received. ***NOTE: The normal imaging and dispensing cycle includes specific delays and pauses between steps setup in the software. When using this feature the timing and delays are no longer controlled by the BCInline and must be properly controlled by the Remote Device.***
MeasureDropDispense>	**DropDispensed>**	Initiate a drop dispense using the appropriate settings for the current profile. Response once drop has been dispensed and the system is ready for movement if needed.

MeasureInspectProcess(a,b,c,d,e,f,g,h,i,j,k,l,m,n)>
or
MeasureInspectProcessNP(a,b,c,d,e,f,g,h,i,j,k,l,m,n)>
	DropCaptured>	Trigger the inspection of the drop which was just dispensed. The response indicates the drop image has been taken and the system is ready for movement if needed. *MeasureInspectProcess> results in 3 response packets while MeasureInspectProcessNP> results in only 2 packets.* Packet 1: drop image has been captured and system is ready for movement if needed.
		NOTE: MeasureInspectProcessNP > initiates a measurement without returning the image packet. All other responses are the same.
		
		The measurement data from the robot system consists of the following:
		a = Process Monitor ID (UUID)
		b = Unique part identifier (UUID)
		c = Measurement position (integer)
		d = Facility (UUID)
		e = Production line identifier (UUID)
		f = Control point (UUID)
		g = Surface profile (UUID)
		h = X coordinate in millimeters
		i = Y coordinate in millimeters
		j = Z coordinate in millimeters
		k = Rotation around the X-axis in degrees (0 if 3-axis only)
		l = Rotation around the Y-axis in degrees (0 if 3-axis only)
		m = Rotation around the Z-axis in degrees (0 if 3-axis only)
		n = Metadata string passed through for tracking/logging

	MeasureInspectProcess (r,s,t,u,v,w,x,y,z)>	Successful measurement Packet 2: *NOTE: MeasureInspectProcessNP> initiates a standard measurement without returning the image packet. All other responses are the same.* The results of the measurement which consists of the following: r = Measured contact angle (This will be set to 999 for measurement failures) s = Outlier points (number of outliers if turned on, if turned off this will be 0) t = Compactness value of the drop (decimal number <=1) u = Drop center distance (integer distance from cross hair location to drop found location) v = Timestamp (internal time of measurement) - “yyyy-mm-ddThh:mm:ss.nnn” w = Drop count (number of measurement drops used out of the cartridge) x = Detection flags (result code indicating if the drop was automatically found to be good or defective) The possible codes are as follows: BD_COMPACTNESS BD_OUT_OF_FOCUS BD_DROP_TOO_SMALL BD_BAD_DISPENSE BD_TOO_MANY_SATELLITES BD_SATELLITES_ML GD y = Pass Fail Flag (status of the inspection using the internal pass/fail limits. The code is a single character.) The possible codes are as follows: P (pass) F (fail) S (fail, surfactant detected) N (none, limits set to 0-180 which disables limits) z = Size of the image in bytes (which is the size of the next packet if not the NP version). *Example response (Failed detection due to outliers and failed measurement):* MeasureInspectProcess(999,40,0.93,62,2018-05-03T15:32:05.327,251,BD_OUTLIERS,F,153815)> *Example response (Good detection but failed measurement):* MeasureInspectProcess(58,0,0.94,9,2018-05-03T15:31:49.972,250,GD,F,160560)>

	a	Packet 3 (only returned for a "MeasureInspectProcess>" command): a = Image from inspection. This will be a binary packet encoded in PNG image format. The image is 480x480 pixels. ***NOTE: This packet will not be sent if the "MeasureInspectProcessNP>" version of the command initiated the process.***
MeasureInspectProcess(a,b,c,d,e,f,g,h,i,j,k,l,m,n)> *or* **MeasureInspectProcessNP(a,b,c,d,e,f,g,h,i,j,k,l,m,n)>**	**TM_ERROR_PUMP_RAMPING>**	Failed measurement: System has not reached pressure yet, measurement was taken too soon after entering measurement mode or system is taking a long time to reach pressure. This response happens if a measure is requested while the SA is displaying the black pressurizing screen.
MeasureInspectProcess(a,b,c,d,e,f,g,h,i,j,k,l,m,n)> *or* **MeasureInspectProcessNP(a,b,c,d,e,f,g,h,i,j,k,l,m,n)>**	**TM_ERROR_PRESSURE:XXXX>**	Failed measurement: The system is not at the proper pressure. The "XXXX" is replaced with the current pressure value.
MeasureInspectProcess(a,b,c,d,e,f,g,h,i,j,k,l,m,n)> *or* **MeasureInspectProcessNP(a,b,c,d,e,f,g,h,i,j,k,l,m,n)>**	**TM_ERROR_NOT_IN_PREVIEW>**	Failed measurement: The SA is not in the measurement mode.
MeasureInspectProcess(a,b,c,d,e,f,g,h,i,j,k,l,m,n)> *or* **MeasureInspectProcessNP(a,b,c,d,e,f,g,h,i,j,k,l,m,n)>**	**TM_ERROR_OVER_DROP_COUNT>**	Failed measurement: The SA is unable to take a measurement because the cartridge is out of liquid. The cartridge must be replaced before measurements can be taken.
MeasureInspectProcess(a,b,c,d,e,f,g,h,i,j,k,l,m,n)> *or* **MeasureInspectProcessNP(a,b,c,d,e,f,g,h,i,j,k,l,m,n)>**	**TM_ERROR_CART_PURGE_NEEDED>**	Failed measurement: The SA is unable to take a measurement because the system requires purging (typically after a cartridge change). Use ContinuousPurge> before attempting again.

MeasureInspectProcess(a,b,c,d,e,f,g,h,i,j,k,l,m,n)> *or* **MeasureInspectProcessNP(a,b,c,d,e,f,g,h,i,j,k,l,m,n)>**	**TM_ERROR_DB_TRANSFER>**	Failed measurement: The SA is unable to take a measurement because the system is currently transferring the result database to the RD.

Initiate a performance check sequence using printed performance check surface card. Because of the complexity of the performance check sequence a flow chart of possible operations and responses is included in Appendix A.

RD command	BCI responses	Description
		Normal sequence
PCHK(a)>	**PCHK>**	Performance check has been started. a = Timeout in seconds. Enter 0 to disable timeout. The timeout starts timing once the barcode scanner has started reading the image. The total time for this command to fully timeout is longer to account for the period of time when the camera is switching modes to prepare for barcode scanning. Minimum time recommended is 2 seconds to ensure a good read. Packet 1: Camera needs to be over the barcode for proper reading when this command is issued.
	ScanOK(a)>	Packet 2: Barcode has been read. *Example response:* ScanOK(31176,241017,2.90,94,02,02.5,2503,2609,https://brightsci.io/A9MzZCH?lot_id=241017)>
	PCHK_CAM_READY_1>	Packet 3: The BCI is ready for the system to move to the first measurement location on the performance check surface card.
Measure> *Or* **MeasureNP>**	**1 or 2 standard measure response packets. See measure response above for details**	Response is identical to a normal measurement response. See measurement command above for details on response format. If the measure is unsuccessful this command will need to be repeated.
	PCHK_CAM_READY_2>	Indicates that the above measurement command was successful and the BCI is ready for the next measurement.
Measure> *Or* **MeasureNP>**	**1 or 2 standard measure response packets. See measure response above for details**	Response is identical to a normal measurement response. See measurement command above for details on response format. If the measure is unsuccessful this command will need to be repeated.

RD command	BCI responses	Description
	PCHK_CAM_READY_3>	Indicates that the above measurement command was successful and the BCI is ready for the next measurement. If the first two measurements were sufficient **PCHK_CAM_READY_3>** won't be sent, instead the PCHK will return a PASSED, FAILED, ERROR, or OVER/UNDER_LIMITS message instead.
Measure> *Or* **MeasureNP>**	**1 or 2 standard measure response packets. See measure response above for details**	Response is identical to a normal measurement response. See measurement command above for details on response format. If the measure is unsuccessful this command will need to be repeated.
	PCHK_PASSED_STOP>	Passing result: The performance check sequence has completed successfully, any adjustments made were minor and do not require a re-test.
Individual measurement bad flag response sample sequence – repeat of measurement required.		
. . . . . .	**PCHK_CAM_READY_1>**	Example for measurement 1, this example can be substituted for any measurement 1-3. Standard sequence contains the BCI response to take measurement #1 which causes the RD to initiate a measurement.
Measure> *Or* **MeasureNP>**	**Measure(?,?,?,?,?,?,BD_COMPACTNESS,?,?)>** *Or* **Measure(?,?,?,?,?,?,BD_OUT_OF_FOCUS,?,?)>** *Or* **Measure(?,?,?,?,?,?,BD_DROP_TOO_SMALL,?,?)>** *Or* **Measure(?,?,?,?,?,?, BD_BAD_DISPENSE,?,?)>** *Or* **Measure(?,?,?,?,?,?, BD_TOO_MANY_SATELLITES,?,?)>** *Or* **Measure(?,?,?,?,?,?, BD_SATELLITES_ML ,?,?)>**	The BCI responded with the measure response but the flags indicate that the drop did not meet the acceptance criteria and was rejected. See measurement command above for full explanation of the fields and their meanings for the measurement return message. ***NOTE: Any flag response other than "GD" will result in the sequence repeating.***
	PCHK_CAM_READY_1>	The same measurement needs to be repeated, so the BCI will send the same response as above in this example causing the sequence to repeat until a good measurement is taken.
Individual measurement failure response sample sequence – repeat of measurement required		
. . . . . .	**PCHK_CAM_READY_1>**	Example for measurement 1, this example can be substituted for any measurement 1-3. Standard sequence contains the BCI response to take measurement #1 which causes the RD to initiate a measurement.
Measure> *Or* **MeasureNP>**	**TM_ERROR_*?????*>**	The BCI responded with a failure message instead of the normal measurement response packets. See measurement command above for possible failure responses.

RD command	BCI responses	Description
	PCHK_CAM_READY_1>	The same measurement needs to be repeated, so the BCI will send the same response as above in this example causing the sequence to repeat until a good measurement is taken.
		Failure response due to lack of fluid. The sequence is terminated.
PCHK(a)>	**PCHK_ERROR_CART_EMPTY>**	The fluid cartridge is empty and needs to be replaced. The performance check sequence is terminated.
		Failure response due to scanning timeout. The sequence is terminated.
PCHK(a)>	**PCHK>**	Immediate response to the performance check with a timeout on the barcode scan set to a value that is not long enough for the read.
	SCAN_TIMEOUT>	After the timeout has elapsed without reading a barcode this response will be sent.
		Failure sequence when the barcode is not from a performance check surface
PCHK(a)>	**PCHK>**	Example request for performance check with a timeout of “a” second(s) with the immediate response indicating acceptance of the command.
	PCHK_ERROR_INVALID_QR_CODE>	The barcode was able to be read but is not of the proper format for a performance check surface.
		Failure sequence when the barcode indicates that the card has expired or is otherwise invalid
PCHK(a)>	**PCHK>**	Example request for performance check with a timeout of “a” second(s) with the immediate response indicating acceptance of the command.
	PCHK_ERROR_CARD_EXPIRED>	The barcode was able to be read and is of the proper format for a performance check surface but the card has expired.
	PCHK_ERROR_OLD_CARD>	The barcode was able to be read and is of the proper format for a performance check surface but the card is an old format.
	PCHK_ERROR_CARD_MISMATCH>	Card does not match the first one used in this check, please use a new card from the same batch of cards.
		Failure sequence – on last measurement request the response is changed to one of the below. The sequence is terminated.
Measure> *Or* **MeasureNP>**	**PCHK_FAILED_STD_DEV_STOP>**	Failed result: This will occur after the 3 inspections have completed and there was too much variation in the drops. A purge is recommended before restarting the performance check.
Measure> *Or* **MeasureNP>**	**PCHK_OVER_LIMITS_STOP>**	Failed result: This will occur after the 3 inspections have completed and the adjustments required push the settings over the upper limits. A purge is recommended before restarting the performance check.

RD command	BCI responses	Description
Measure> *Or* **MeasureNP>**	**PCHK_UNDER_LIMITS_STOP>**	Failed result: This will occur after the 3 inspections have completed and the adjustments required push the settings below the lower limits. A purge is recommended before restarting the performance check.
Measure> *Or* **MeasureNP>**	**PCHK_ERROR_BD>**	Failed result: A drop failed to detect properly, try again with a new card. If this error repeats check alignment, height and deposition visually.
		Cancel request from the RD during the above sequences. This terminates the sequence
CancelPCHK>	**CancelPCHK>**	Sequence is terminated and no further adjustments are made.

Start a prime shot (1 drop) used before a dispense sequence to fill the nozzle with liquid

RD command	BCI responses	Description
		Normal sequence
PrimeShot>	**PrimeShot>**	This purge is extremely short so will return almost immediately – response is sent once purge is complete.

Start command for measuring a part

RD command	SA responses	Description
StartPart(a,b,c)>	**StartPart>**	*NOTE: Not currently used for anything.* The StartPart data from the robotic system consists of the following: a = Program name b = Number of Measurement positions c = Part ID (barcode or manually entered value)

Stop command for measuring a part

RD command	SA responses	Description
StopPart(a,b,c)>	**StopPart>**	*NOTE: Not currently used for anything.* The StopPart data from the robotic system consists of the following: a = Program name b = Number of measured positions c = Part ID (barcode or manually entered value)

Set status of an output bit (0-3)

RD command	BCI responses	Description
		Normal sequence
SetOutputPin(a,b)>	**SetOutputPin(c,d)>**	Set the state of a physical output on the system. a = Pin number 0-3 of the output to be set. ***NOTE: When DIO result mode has been enabled the first three pins (0-2) are used for the ready, pass, and fail outputs and cannot be used for general outputs (0=Ready, 1=Pass, 2=Fail).*** b = Desired state for the output. Valid states are HIGH and LOW The BCI response contains the results of the operation. c = Pin number 0-3 of the output which was set d = Status of the output. Valid status responses are: HIGH = output was written and is in a high state LOW = output was written and is in a low state ERROR_PIN = output pin number is not valid (must be 0-3) ERROR_IO = I/O board not found

Shutdown the BCI

RD command	BCI responses	Description
		Normal sequence
ShutDown>	**ShutDown>**	This powers the BCI down.

Start a 10 shot purge (Approximately 10 drops – short amount of time)

RD command	BCI responses	Description
		Normal sequence
TenShotPurge>	**TenShotPurge>**	Response is sent once purge is complete. There will be a short time delay between sending the command and receiving the response which indicates completion. The timeout for this command should be appropriately long to not time out during the purge.

5. Commands for troubleshooting and debugging use

These commands are more unusual and typically not needed to implement a standard interface.
NOTE: All BCI responses will be terminated with CrLf except for binary image data.

Dispense a custom drop

RD command	BCI responses	Description
		Normal sequence
DSP(a,b,c)>	**DSP>**	Dispense a drop with custom parameters (Echo response is immediate): a = Number of droplets - small droplets that make up the larger drop. b = Valve open time (microseconds) – how long the valve will be open for each droplet. Normally 519 c = Valve Period (microseconds) – time between each valve opening (this must be longer than the valve open time). Normal value is 6863
	DSP_Complete>	Response once the dispense has completed.

Get configuration information contained in the BCI About Screen

RD command	BCI responses	Description
		Normal sequence
GetInfo>	**GetInfo(a)>**	This returns the current configuration values in the BCI as displayed in the about screen. The information provided through this call includes information about the currently loaded profile, keys describing the profile can be different between different profiles so it should not be assumed that all keys will always be present. Note: Special characters are stripped out of the data stream. a = returned information from the about screen formatted with descriptors and separators. *Example response:* GetInfo(Serial Number: BCBB8,Software version: 20250924.3,Firmware version: BC2.00BETA,Head Firmware version: BC0.41BETA,Head Fan On Time: 827,Head Fan Setpoint (C): 40,Head Fan Warning Tol. (C): 5,Head Fan Temperature (C): 33,Transducer setting: 0004,Stability setting: 0025,Valve use: 1963340,OS build: development.20250821.081554,Kernel build: 0.1buildVersion,Device IP: 10.0.0.154,Cloud Record: 64221/64221,Cloud Image: 11540/11540,Available Memory: 7762 MB,Total Memory: 8508 MB,Available Heap: 168 MB,Total Heap: 384 MB,Battery: 8.37 V,User: user1,Surface Profile: Default Profile,User Drop remaining: 77822.1,Purge Drop used: 70931,Autologin: Enabled,Drop Note: CT39|20/20|,Min Pass Angle: 0,Max Pass Angle: 180,Near Fail Limit: 0,Detection Accept/Reject: Auto,SmartDrop Limit: 0.7,Number of Outliers Pass Limit: 19,Continuous Outliers Pass Limit: 16,Time: Thu Aug 21 03:19:03 CDT 2025,Time zone: Central Daylight Time,Cartridge Serial #: 5BA76E2E7D03C1E5,Calibration Due: April 29 2022,Days Since Performance Check: 823,PCHK Offset: 0,*Drop Dispense Parameters*,Drop settle time: 0.35 seconds,Pressure: 3.5,Pressure Tolerance: 50 %,Droplets per Drop: 100,Valve spike time: 519,Valve open time: 519,Valve period: 6863,Drop Mass: 1.5,Extended Purge #: 0,Continuous Purge Time: 350.4,Quick Purge Shots: 0,*Analysis Parameters*,Dynamic Detection: Enabled,Drop center: Auto,Crosshair position: [50%|50%],Outlier Rejection: Enabled,Outlier % Diff. Threshold: 15,Ellipse Mode: Disabled,Image Alignment: Disabled,Image Alignment Eps: 1.0E-5,Image Alignment Max Count: 25,Invert Finding: Disabled,Edge Width Filter: 15,Edge Length Filter: 15,Edge Separation: 5,Max Passes: 10,Pass 1 Min Size: 75,Pass 1 No Center Size: 150,Center Blur Size: 21,Center Min Diameter: 15,Center Max Diameter: 370,Center Merge All: Disabled,Center Merge Distance: 7,Multiplier: Pass 1 Near: 0.5,Multiplier: Pass 1 Far: 1.5,Multiplier: Pass 2 Near: 0.55,Multiplier: Pass 2 Far: 1.4,Multiplier: Pass 3 Near: 0.7,Multiplier: Pass 3 Far: 1.3,Dyne Mode: Disabled,Dyne Parameters: Not Assigned,Wetting Analytics: Disabled,Wetting Overall Time: 0.7 s,Wetting Image Interval: 0,Wetting Delta: 2,*Optical Parameters*,Illumination: 70,Exposure: 0,Optical Cal: 1831.2101,DiamCorrectB: 1.0,*Purchasable Options*,Unlock All: Enabled)>

Read pump pressure setpoint and actual value (values are in PSI)

RD command	BCI responses	Description
		Normal sequence
GetPRS>	**GetPRS(a,b)>**	Returns both the setpoint for the pressure and the current pressure as read by the BCI. a = The set pressure in PSI as a decimal number with up to 2 significant digits. b = The actual pressure in PSI as a decimal number with up to 2 significant digits. *Example response (pressure is set to 3 PSI):* GetPRS(3,2.94)>

Verify BCI is alive and responsive

RD command	BCI responses	Description
		Normal sequence
Ping>	**Ping>**	Provides a method to prove connectivity has not been lost without commanding an action. This should normally be replaced by the GetStatus> command which also returns the status of the device giving information on important faults and status which should be handled by the RD. Use Ping> for troubleshooting connectivity.

Turn the pump off

RD command	BCI responses	Description
		Normal sequence
PumpOff>	**PumpOff>**	Turns the pump off which will depressurize the cartridge.

Turn the pump on

RD command	BCI responses	Description
		Normal sequence
PumpOn>	**PumpOn>**	Turns the pump on which will pressurize the cartridge.

Set pump pressure - Pressures are in PSI

RD command	BCI responses	Description
		Normal sequence
SetPRS(a)>	**SetPRS>**	Sets the target pressure (PSI) to the value contained in the command. a = The new target pressure in PSI as a decimal number with up to 2 significant digits *Example command to set the pump pressure to 3.5 PSI:* SetPRS(3.5)> ***NOTE: This command does not wait until the target pressure is reached before returning.*** ***NOTE: This command does not change the pressure stored in the surface profile, switching surface profiles will return the pressure to the stored value for the surface profile.*** ***NOTE: Prime and Purge commands can change the pressure if the set pressure is too low to perform the function. If the pressure was changed, then the pressure will return (after the purge) to the value stored in the active surface profile.***

Ethernet message logging - Disable

RD command	BCI responses	Description
		Normal sequence
TCPLoggingOff>	**TCPLoggingOff>**	Disables logging of the Ethernet messages to the “log_tcp.db” database file.

Ethernet message logging - Enable

RD command	BCI responses	Description
		Normal sequence
TCPLoggingOn>	**TCPLoggingOn>**	Enables logging of the Ethernet messages to the “log_tcp.db” database file. ***NOTE: Logging will be automatically disabled on a power cycle of the BCInline to reduce the risk of accidentally leaving this feature on and filling the storage space with the log database.***

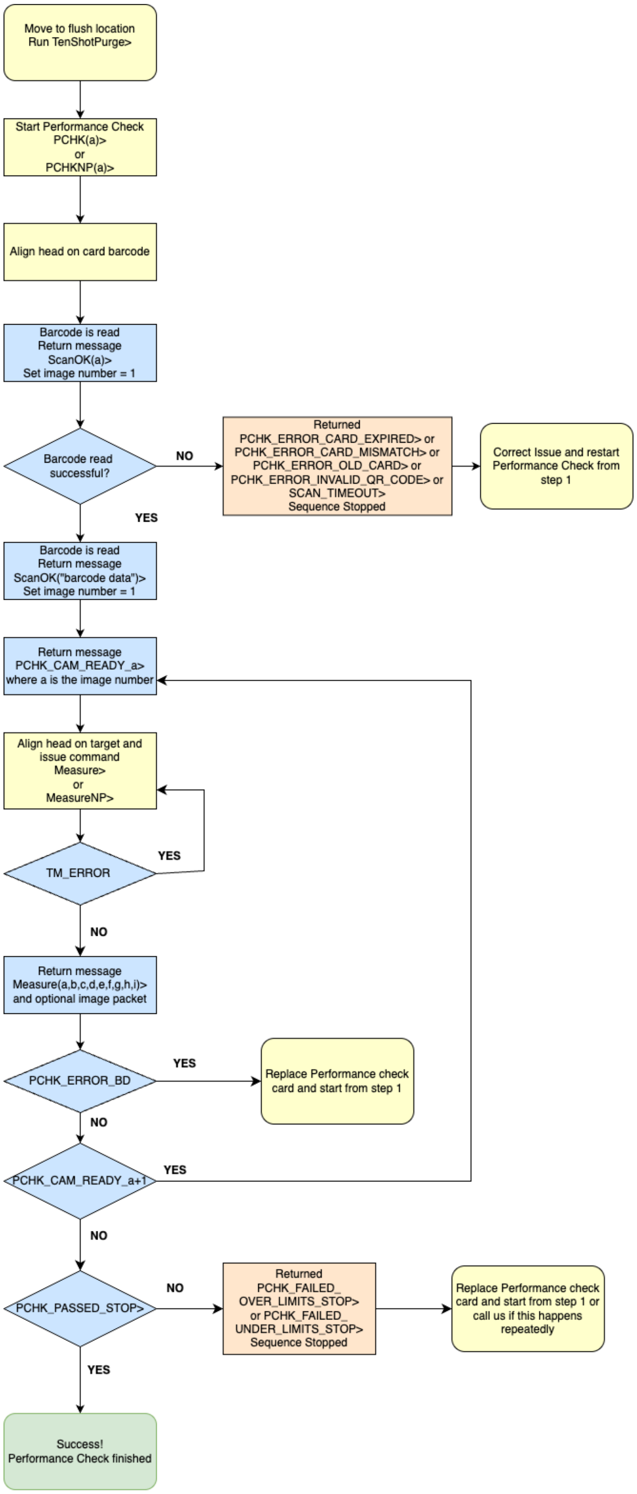
Additional Commands

RD command	SA responses	Description
MeasureAreaNP(openTimeMs,periodMs,numDroplets)>	**MeasureAreaNP(area,r,s,t,u,v,w,x,y,z)>**	`openTimeMs`, `periodMs`, `numDroplets` are integers. `area` (int) is prepended to the standard measurement fields `r`…`z` (contact angle or 999, outliers, compactness, center distance, timestamp `yyyy-MM-dd'T'HH:mm:ss.SSS`, remaining drops, flags, pass/fail, image-size placeholder). single packet; no image

LoadProfileById(profileId)>	**LoadProfile>**	Succcess: profile becomes active; illumination set Not found: `LoadProfileNotFound>`
MeasMetaUp(programName,partId,condition,coordinates,sampleNumber,measurementNumber,trayNumber)>	**MeasMeta>**	*NOTE: Not currently used for anything.* Caches these seven values on the device for later retrieval.
MeasMetaDown>	**MeasMetaDown(programName,partId,condition,coordinates,sampleNumber,measurementNumber,trayNumber)>**	*NOTE: Not currently used for anything.*
GetCartridges>	**GetCartridges(code1:used1:total1,code2:used2:total2,...)>**	Comma-separated list; each entry is `cartridge.code:usedVolume:totalVolume`
PurgeClear	**PurgeCleared>**	If no purge was needed: `PurgeClearNotNeeded>`
GetPartImageWithMarkers(partId)>	**GetPartImageWithMarkers(<byteCount>)>**	Two-packet response: a length header followed by raw PNG bytes including marker overlays. **On error:** `GetPartImageWithMarkersError>`

6. Appendix A

Flow chart for the possible paths and outcomes of a performance check sequence when a BCInline is in robot mode and commanded to execute a performance check. See PCHK> for commands associated.


Move to flush location
Run TenShotPurge>
Start Performance Check
PCHK(a)>
or
PCHKNP(a)>
Align head on card barcode
Barcode is read
Return message
ScanOK(a)>
Set image number = 1
Barcode read successful?
NO
Returned
PCHK_ERROR_CARD_EXPIRED> or
PCHK_ERROR_CARD_MISMATCH> or
PCHK_ERROR_OLD_CARD> or
PCHK_ERROR_INVALID_QR_CODE> or
SCAN_TIMEOUT>
Sequence Stopped
Correct Issue and restart Performance Check from step 1
YES
Barcode is read
Return message
ScanOK("barcode data")>
Set image number = 1
Return message
PCHK_CAM_READY_a>
where a is the image number
Align head on target and issue command
Measure>
or
MeasureNP>
TM_ERROR
YES
NO
Return message
Measure(a,b,c,d,e,f,g,h,i)>
and optional image packet
PCHK_ERROR_BD
YES
Replace Performance check card and start from step 1
NO
PCHK_CAM_READY_a+1
YES
NO
PCHK_PASSED_STOP>
NO
Returned
PCHK_FAILED_
OVER_LIMITS_STOP>
or PCHK_FAILED_
UNDER_LIMITS_STOP>
Sequence Stopped
Replace Performance check card and start from step 1 or call us if this happens repeatedly
YES
Success!
Performance Check finished