FORCEN

BonneChere
Live Data Viewer

MAN-00027-01

ForceN Inc. BonneChere Software User Manual

Revision History

Rev.	Effective Date	Author(s)	Change #	Change Summary
A	2025/JUN/11	Roberto S. Enkerlin, Jana El-Zoghbi	N/A	Initial Release

Table of Contents

1. Introduction

This document is a user manual for Forcen's RTD Visualization software, BonneChere. This manual provides essential information on how to install and use BonneChere as well as all the features available. Please retain this manual for future reference and troubleshooting.

2. Acronyms

Acronym	Description
ASCII	American Standard Code for Information Interchange
DOF	Degrees Of Freedom
EMI	Electromagnetic Interference
EOF	End Of Frame
IFS	Internal Field Separator
RFI	Radio Frequency Interference
RTD	Real-Time Data
SOF	Start Of Frame
SPC	Specifications
TCP	Transmission Control Protocol
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
UDP	User Datagram Protocol

3. Reference Documents

Document Number	Description
COM PROT TDS-00007-001 - REV B	ETHERCAT PROTOCOL SPECS, FOR POP-00006-XXX DEVICES
ASCII TDS-00005-001 - REV D	ASCII PROTOCOL SPECS, FOR POP-00006-XXX DEVICES

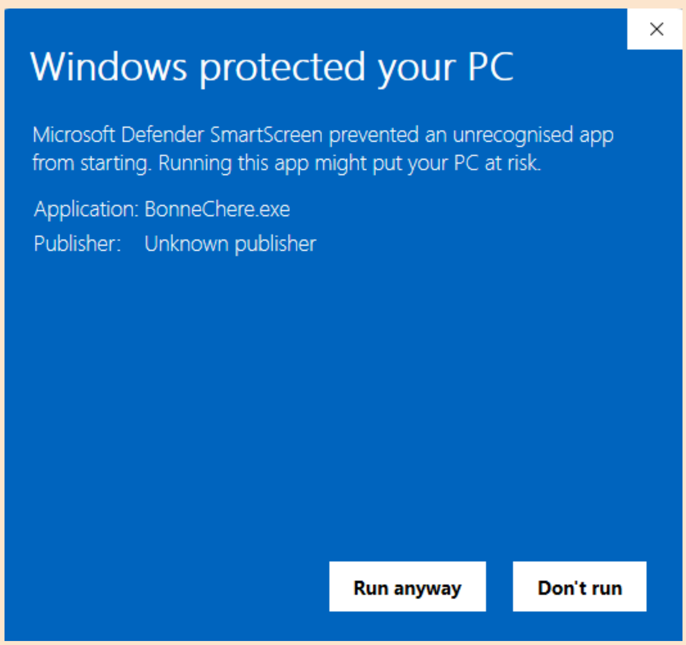
4. Installation

This section outlines the procedure for installing BonneChere.

1. Download and unzip the software package, and save the extracted files to an appropriate folder.
2. Launch the BonneChere executable application.
    a. Windows: double-click on the “BonneChere.exe” file.

NOTE:

If you encounter the following pop-up when trying to run the app, click “more details” and then “run anyway.”



    b. Linux: open a terminal in the extracted folder and run ”sudo ./Bonnechere”.

5. Adding Device

Connect the sensor to your PC, once the Bonnechere program loads, the following window will appear.

Figure 2: BonneChere Main Program Window.

Click the " + New Device " button, a pop-up to select the session type will show up. Once the parameters have been configured for the type, click "OK" to begin a session.

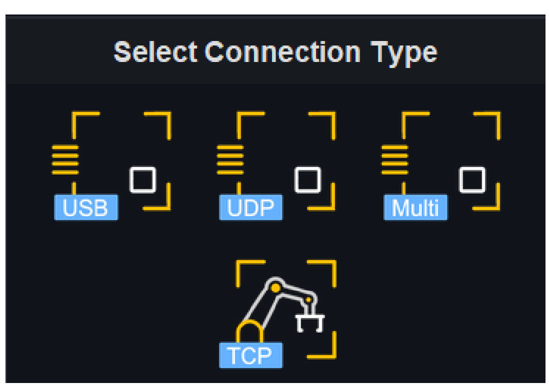
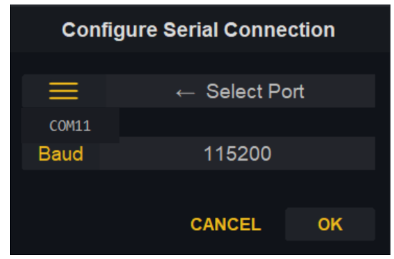
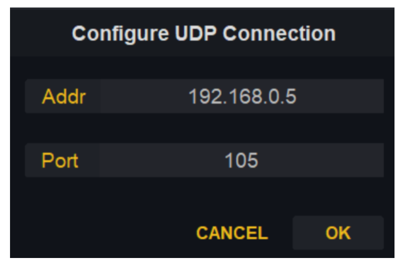
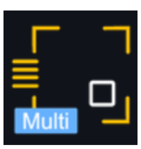
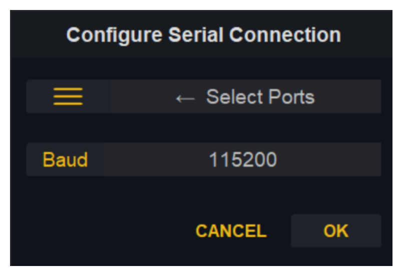
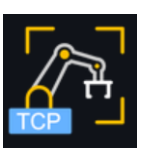
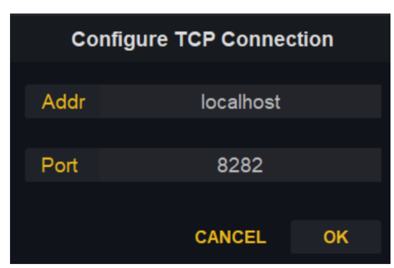


Figure 3: BonneChere Session Types

Button	Details	Connection Popup
	**Connect Over Single COM Port** This option is for starting a session for a single sensor that is connected to the computer over USB. The device is connected by selecting the desired COM port from a drop-down. Note: Baud rate is ignored for purely USB connections. If a USB-to-UART adapter is used, the baud rate must be set to match the sensor's configured baud rate (default: 115200).	
	**Connect Over Single UDP Port** This option is for starting a session for a single sensor that is connected to the computer over Ethernet. This form of connection utilizes a UDP connection with the Forcen-ASCII protocol. The device is connected by specifying the IP address and port number of the UDP connection (defaults shown, port number for UDP connection should not change).	
	**Connect Over Multiple COM Ports** Similar to the USB device option, except multiple COM ports can be selected for a single connection. This allows the program to treat all the devices selected for a given session as a single device.	
	**Connect Over Single TCP Port** This option allows connection to a ForceN-configured Jig server over TCP. This is a generic mechanism that allows linking to any jig with inputs and outputs into BonneChere. The connection is created by specifying the IP address and port number of the jig server to connect to.	

6. Main Window

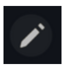
Once a device is added, you will be taken to the main window in Bonnechere.

Figure 4: BonneChere Main Window

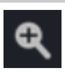
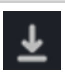
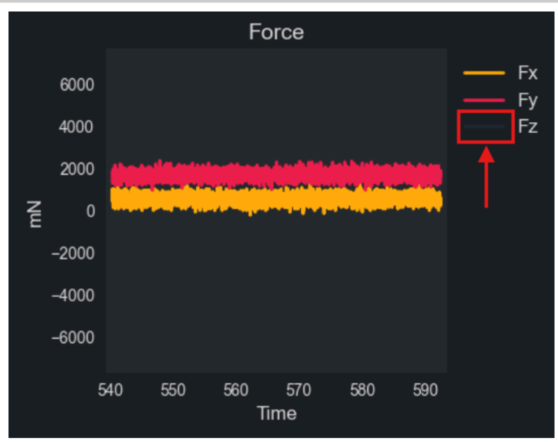
6.1. Tool Overview

Tab	Details
6-DOF Wrist	**Info Tool** Title may differ per sensing system, this is the main tab where live sensor data is plotted.
Log	**Logging Tool** The log tool allows easy real-time data logging from a connected device.
Config	**Config Tool** The config tool can be used to create / run custom python scripts, & develop test routines.
Terminal	**Terminal Tool** The terminal tool is used to communicate directly to the connected sensor for configuration.
Program	**Program Tool** The program tool is used to update the firmware version flashed onto the connected device.

6.2. Adding / Removing Device

Button	Details
	**Add Device** Add another device & start a new session.
	**Update Device Info** Rename the device or disconnect.
	**Remove Device** Remove the device & end the session.

6.3. Live Plotting Tab

Button	Details	Notes
Start Plotting	Start/stop plotting live data.	-
^ 5200 v	Change the time step size with the arrows.	-
Clear Plot	Clear the live plot.	Live plotting will continue.
	Zoom into a section of a plot.	These plot tools (located at the bottom of the page) are only enabled when live plotting is stopped.
	Pan over & move the plotted data.	
	Download the plots as images.	
Fz	Hide or show channels by clicking on the legend lines.	

7. Log Window

You can easily configure real-time data logging from the device using the log tool.

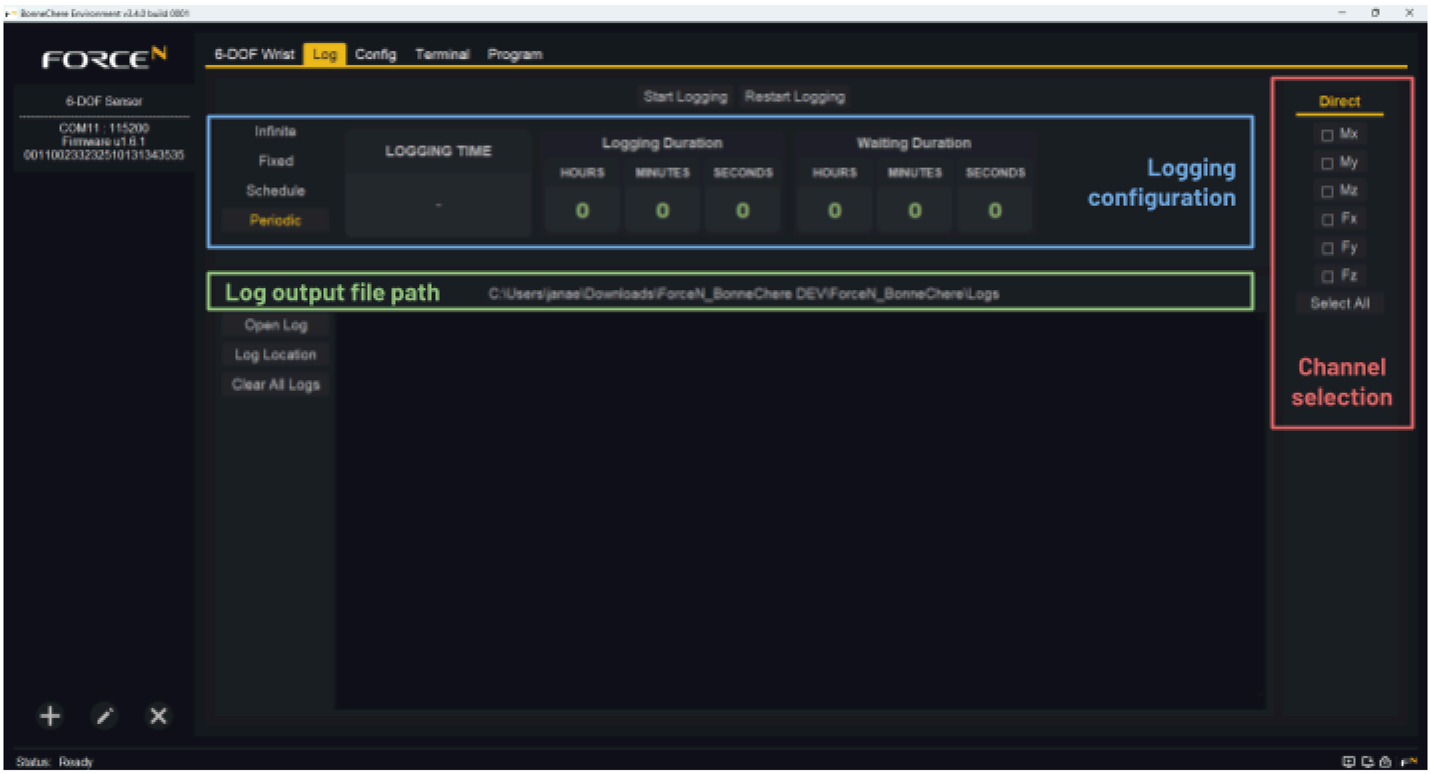
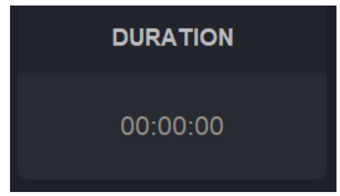
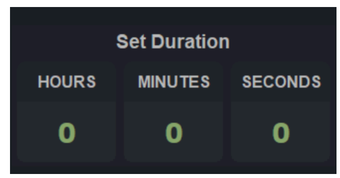
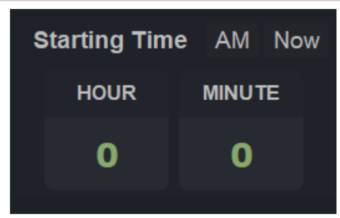
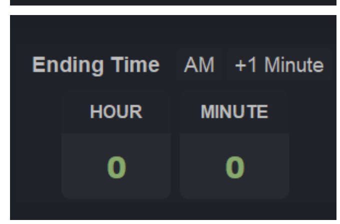
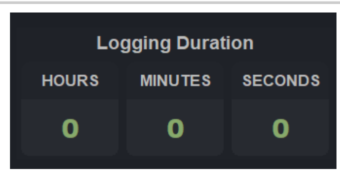
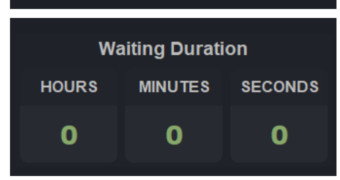


Figure 5: BonneChere Logging Window

Choose one of the four logging configurations, define the logging duration & parameters. Select the channels of interest for logging, then click "Start Logging" to begin. Once logging is stopped, either manually or when the set logging duration has been met, the log will be saved as a TXT document in the file path specified.

7.1. Logging Configurations

Configuration	Details	Configurable Table
Infinite	**Infinite Logging** Log data for an unspecified amount of time. Simple start and stop logging using the buttons located at the top of the page.	
Fixed	**Fixed Logging** Log data for a fixed amount of time. Enter the desired logging time into the duration table to define the fixed logging duration.	
Schedule	**Scheduled Logging** Log data during a scheduled period. Enter the start and end times into the tables to define the schedule.	 
Periodic	**Periodic Logging** Log data periodically over a specified duration. The waiting duration is the time spacing for when data should be logged. Ex: waiting duration of 5 seconds means there will be a 5-second break between each data point logged over the set duration.	 

8. Config Window

The config tool is an integrated coding environment to create and test scripts that convert raw sensor data into useful outputs.

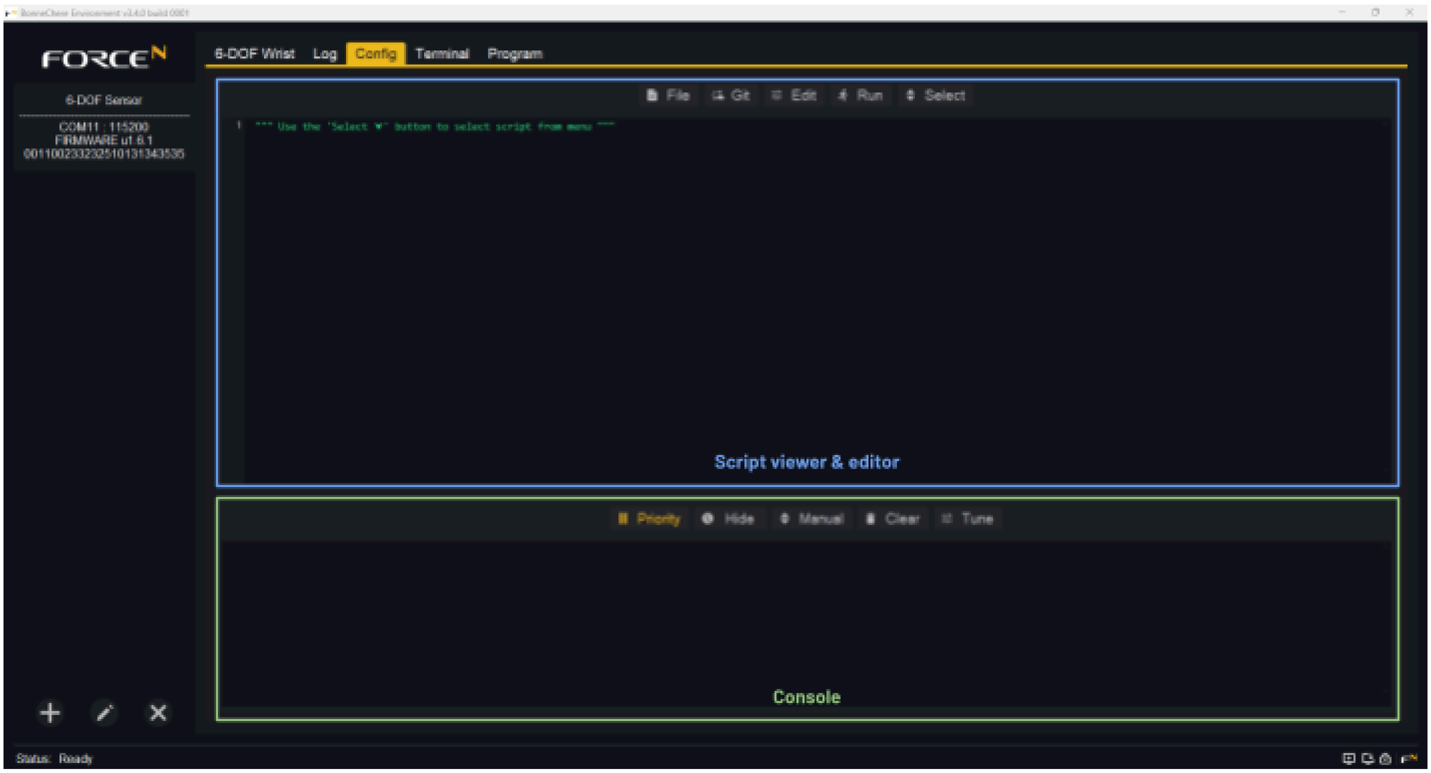


Figure 6: BonneChere Config Window

Console Button	Details
Priority	**Toggle Console Data** Priority" shows only priority messages in the console; "Live" shows all messages.
Hide	**Toggle Timestamps** "Hide" removes timestamps in the console; "Show" displays them.
Manual	**Toggle Scroll** "Manual" disables the auto-scroll feature; "Auto" enables it.
Clear	**Clear Console** Clears all currently displayed console text.
Tune	**Tune Script** Edit the selected working script's description & constants.

Script Viewer & Editor Buttons

Button	Sub-buttons	Details
File	Save Save As Delete Format	**Save** current script to git managed files. **Save As** un-editable copy of current script into internal files. **Delete** currently selected script from its location (managed or internal). **Format** spacing and tabbing in the current script to fix any space mismatch issues.
Git	Commit Push Pull	**Commit** all local changes to local git repo. **Push** all locally committed changes to remote repo. **Pull** all remote changes into local git repo.
Edit	Undo Redo Cut Copy Paste Configure... How To Use	**Undo / Redo** changes. **Cut / Copy / Paste** code sections. **Configure** script parameters through a popup. **How To Use** details printed in terminal, provides info on how to use the script editor on the console window.
Run	Compile Run Link	**Compile** script, checks for syntax errors using Python's syntax checker. **Run** the currently selected script. **Link** the current script to device so it automatically loads and runs when that specific device is connected in Bonnechere.
Select	New... Internal Scripts Managed Scripts	**New** script creation. Opens a new script configurator popup. **Internal Scripts** are saved in Bonnechere's configuration files, move with shared zipped folders, and cannot be edited. **Managed Scripts** are pulled from a Git repository and can be edited, committed, and pushed for shared access by users with Git privileges.

9. Terminal Window

The terminal tool is used to communicate directly to the connected sensor for configuration.

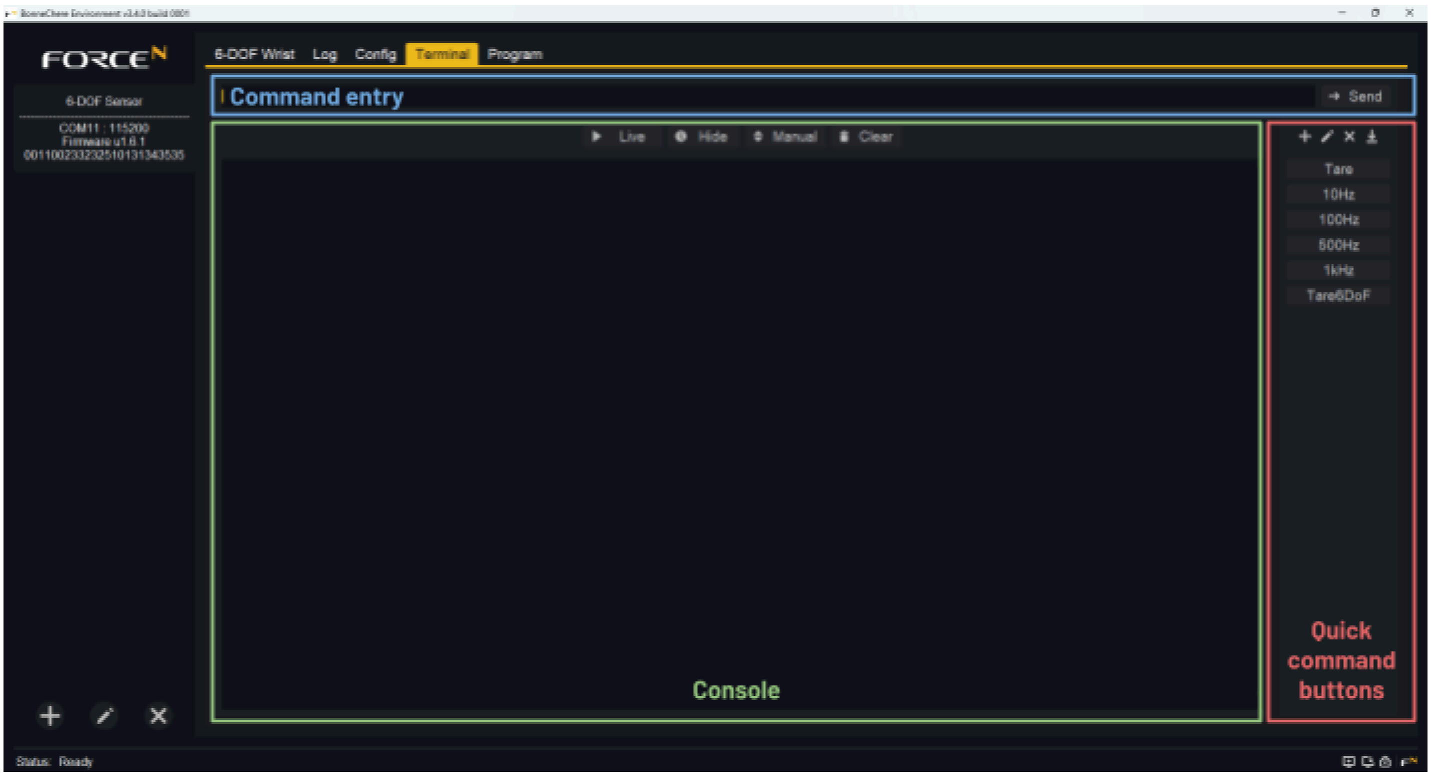


Figure 7: BonneChere Terminal Window

ASCII commands can be sent to the sensor using the command entry bar. The return codes will populate in the console. For quick commands, simply click the buttons to the right of the console.

Quick Command Button Info

Button	Details
Tare	Tare the sensor readings using the **<SDM3>** ASCII command. Plotted data will be zeroed.
10Hz	Set the sensor data rate to 10 Hz.
100Hz	Set the sensor data rate to 100 Hz.
500Hz	Set the sensor data rate to 500 Hz.
1kHz	Set the sensor data rate to 1 kHz.
Tare6DoF	Tare the sensor readings using the **<SDM5>** ASCII command. Plotted data will be zeroed.

To create a new quick command, click on the + button. Define the command name and the ASCII input - see section 9 for ASCII command information.

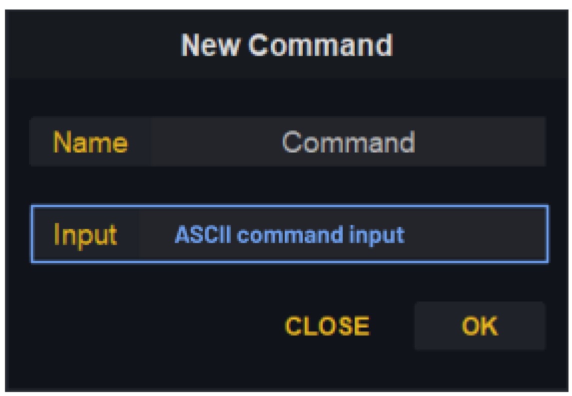


Figure 8: Creating A New Command

10. ASCII Commands

10.1. ASCII Protocols & Commands - Overall Structure

The protocol utilizes an ASCII-based packet structure where each character is used to specify an action and/or information. The overall packet structure of ForceN's ASCII protocol is shown below.

ForceN ASCII packet structure

ForceN ASCII Protocol Data				
SOF	**Command Type**	**Data Packet**	**EOF**	**IFS**
<	1 ASCII Character	Up to 50 ASCII Characters	>	\n

The first character is a Start of Frame (SOF) indicator; this acts as a marker to the beginning of the packet. The second character specifies the command type; the device supports different command types mapped to different actions. After the second character, there is a multi-character data packet which includes the information that is required for a given command type. Each of these command types and their corresponding data packets are explained further in the Commands section. The last two characters of the packet are the End of Frame (EOF) and Internal Field Separator (IFS), indicating where the frame ends.

NOTE:
The device must receive the SOF and EOF characters in order to process any incoming messages. The IFS character is desired to ensure separation between packets but is not required.

10.2. ForceN ASCII Commands

10.2.1. Overview

The ForceN ASCII protocol supports multiple types of commands that specify different actions. These actions are indicated by a single character within the ASCII packet. Each action and its corresponding character are summarized in the table below.

Available ForceN ASCII commands

Character	Command Type	*Direction	Description
C	CNF	Receive	Configuration (state control) command
S	SET	Receive	Set (write) device register command
G	GET	Receive	Get (read) device register command
r	RSP	Transmit	Register information response command
e	ERR	Transmit	Error response command
a	ACK	Transmit	Acknowledgement response command

10.2.2. Configuration (CNF) Command

The configuration command is used to update or change the module state. The data packet of the command is a single ASCII character that specifies the requested module state to achieve the desired action. More details regarding the CNF command packet, and the available module states, are summarized in the table below.

Command Type	Data Packet
C	1 ASCII Character

The command type character is set to the ASCII character 'C' referring to the configuration command. This command is used to save the configuration settings. The data packet is a single character that specifies the requested state to achieve the desired action as summarized below.

Character	Frame	State	Description
S	<CS>	Save Configurations	Saves any updated configurations to EEPROM

10.2.3. Device Register Command (SET)

The set device register command is used to write data to the device's built-in registers. All the available registers are shown in the Register Map section of this document. The structure of the command is shown below.

Command Type	Data Packet	
S	2 ASCII Characters Register Index	Multi-Character Data

The register index is a unique two-character index as defined in the register map of the hub which refers to the specific register of interest. The data is multi-character information either defined as a base 10 number (default) or a hexadecimal number (if it starts with 0x) which is written to the selected register.

10.2.4. Get Device Register Command (GET)

The get device register command is used to read data from the device's built-in registers. All the available registers are shown in the Register Map section of this document. The structure of this command is summarized below.

Command Type	Data Packet
G	2 ASCII Characters - Register Index

The register index is a unique two-character index (defined in the register map of the module) which refers to the specific register of interest. After the GET command is sent, the module will respond with a register information response if the read was successful (see Register Information Response (RSP) Command) or an ERR message otherwise (see Error Response (ERR) Command).

10.2.5. Register Information Response (RSP)

The register information response is used by the hub to respond with data in the specified register of a GET command. The structure of the register information response is shown below.

Command Type	Data Packet
r	Multiple ASCII Characters - Data

The data packet of a RSP response represents the value of the register. If the data starts with characters "0x" then the following characters should be read as hex numbers. Otherwise, these characters should be read as a base 10 number.

10.2.6. Error Response (ERR)

When the hub is unable to process or perform a user request, the device responds with this command type indicating the error that occurred back to the user. The structure of this response is shown below.

Command Type	Data Packet	
e	0x	1-2 ASCII Characters - Error Code

The data packet of an ERR response specifies the error code which maps to a specific message. The table below summarizes all these error codes and their interpretations.

Error Code	Definition	Description
0x0	Unknown Error	An unknown error occurred
0x1	Action unsupported	Unsupported command type received
0x2	Action invalid	Invalid command type received
0x3	Register Address Invalid	Invalid register address requested
0x4	Register Data Invalid	Invalid register data received
0x5	Packet length Invalid	Invalid packet length received, packet may be missing information
0x6	Configuration State Invalid	Invalid state requested
0x7	Save Failed	Failed to execute save request
0x8	Read Access Denied	Attempting to read write-only register
0x9	Write Access Denied	Attempting to write to read only register
0xA	Write Error	Value being written to register is out of range
0xB	Wrong Mode	Device in wrong mode for requested action (eg: requesting data using RQT while in SLEEP mode)
0xC	Metadata Error	Internal error with device's protocol/peripheral metadata
0xD	Read Error	Read error occurred while reading specified register

10.2.7. Acknowledgement Response (ACK)

When the hub successfully processes a user request, the device responds with this command type indicating the acknowledgment that occurred back to the user. The structure of this response is shown below.

Command Type	Data Packet	
a	0x	1-2 ASCII Characters - Ack Code

The data packet of an ACK response specifies the acknowledgment code which maps to a specific message. The table below summarizes all these acknowledgment codes and their interpretations.

Code	Definition	Description
0x0	Boot Message	Boot up message signifying the device is ready
0x1	Write Success	Register write command is executed successfully
0x2	Save Success	Save command is executed successfully

10.2.7.1. Saving (ACK)

When the hub successfully processes a user request, the device responds with this command type indicating the acknowledgment that occurred back to the user. The structure of this response is shown below.

Command Type	Data Packet	
a	0x	1-2 ASCII Characters - Ack Code

The data packet of an ACK response specifies the acknowledgment code which maps to a specific message. The table below summarizes all these acknowledgment codes and their interpretations.

Code	Definition	Description
0x0	Boot Message	Boot up message signifying the device is ready
0x1	Write Success	Register write command is executed successfully
0x2	Save Success	Save command is executed successfully

10.3. Device Registers

The access coding for each register is shown in the table below.

Coding Legend	Read/Write	Read only	Write only
Command	<GX> / <SX>	<GX>	<SX>

**X represents the address input from the register map.*

High-level register map for devices

Addr	Name	Min	Default	Max	Description
Device Configuration & Model Taring Registers					
DR	DATA RATE* (R)	1	100	1000	Select the device operating data rate (Hz)
TT	TARE TIME	1	5	250	Time in seconds for data collection during taring
Device Information Registers					
I1	UNIQUE ID 1	32-Bit unsigned Integer			Unique ID part 1
I2	UNIQUE ID 2	32-Bit unsigned Integer			Unique ID part 2
I3	UNIQUE ID 3	32-Bit unsigned Integer			Unique ID part 3
FV	FIRMWARE VERSION	32-Bit unsigned Integer			Firmware version currently running
E0	ERROR 0	0	0	-	Error Register 0
E1	ERROR 1	0	0	-	Error Register 1
E2	ERROR 2	0	0	-	Error Register 2
Ethernet Setting Registers *For Ethernet Sensors Only					
IP	ETH IP ADDRESS	32-Bit unsigned Integer			Configured IPV4 device address. *Default: 0xC0A80005 = 192.168.0.5*
NM	ETH NETMASK	32-Bit unsigned Integer			Configured Ethernet netmask. *Default: 0xFFFFFF00 = 255.255.255.0*
GW	ETH GATEWAY	32-Bit unsigned Integer			Configured gateway. *Default: 0xC0A80001 = 192.168.0.1*
MC	ETH MAC ADDRESS	48-Bit unsigned Integer			Configured MAC address. *Default: 0x808080808080*

10.4. Sending ASCII Commands

This section outlines how to send ASCII commands in the terminal tab of BonneChere to your sensor. The tables below show the list of commands available for the device.

> **NOTE:**
> *The **save command <CS> must be sent** to save any modified configurations. If the save command is not sent, the modified registers will reset back to the default value. To check the current set value for a register, use the <G**X**> command, where **X** represents the register address of interest.*

10.4.1. Changing Data Rate

The structure to set the data rate and the expected success response is shown in the table below.

Type	Value	Description
Command	<SDR**X**>	Set data rate to **X** Hz
Response	<a0x1>	Write acknowledgment

* Where **X** represents the value of the data rate set in Hz.

The data rate can be set from 1 Hz - 1000 Hz; the table below lists recommended data rates.

Data Rate (Hz)	Command
100	<SDR**100**>
500	<SDR**500**>
1000	<SDR**1000**>

Remember that if the data rate has been modified using a SET command, the save command <CS> must be sent to save the modified data rate register value.

10.4.2. Taring

Vertical Taring Procedure

Step	Command	Command Description	Response (ordered)	Response Description
1	<GU6>	Read the current set taring register mode	r1.0	Taring routine is set to horizontal
			r0.0	Taring routine is set to vertical
1.1	<SU60>	If the response from step 1 is *not* **r0.0**, this command sets the taring routine to vertical.	a0x1	Register write command is executed successfully.
2	<GTT>	Check the current taring time value	rXX	Taring time (seconds) is equal to XX
2.1	<STTXX>	Set the desired taring time to XX seconds. 5 seconds is the minimum time recommendation.	a0x1	Register write command is executed successfully.
3	<GDR>	Read the current data rate.	rXXx	Current data rate (Hz) is equal to XXX
3.1	<SDRXX>	Set the data rate (XX) to your ideal operating data rate.	a0x1	Register write command is executed successfully.
4	<CS>	Save any changes made	a0x1	Save successful
Once step 4 is complete, orient the sensor horizontally before sending the step 5 command.				
5	<SDM5>	Initiate vertical taring.	1 0x10	Transition codes, you should see two.
			2 0x21	Device is waiting for rolling to start.
			3 0x20	Waiting code, arm is not considered fully horizontal by the device.
			4 0x11	Taring has begun, system is collecting data.
			5 0x10	Transition code, data collection finished
			6 0x0	Success code
If taring was successful, outputs in the plotting tab should be centered around zero.				

Horizontal Taring Procedure

<table>
<tr><th>Step</th><th>Command</th><th>Command Description</th><th colspan="2">Response (ordered)</th><th>Response Description</th></tr>
<tr><td rowspan="2">1</td><td rowspan="2"><GU6></td><td rowspan="2">Read the current set taring register mode</td><td colspan="2">r1.0</td><td>Taring routine is set to horizontal</td></tr>
<tr><td colspan="2">r0.0</td><td>Taring routine is set to vertical</td></tr>
<tr><td>1.1</td><td><SU61></td><td>If the response from step 1 is not r1.0, this command sets the taring routine to horizontal.</td><td colspan="2">a0x1</td><td>Register write command is executed successfully.</td></tr>
<tr><td>2</td><td><GTT></td><td>Check the current taring time value</td><td colspan="2">rXX</td><td>Taring time (seconds) is equal to XX</td></tr>
<tr><td>2.1</td><td><STTXX></td><td>Set the desired taring time to XX seconds. 5 seconds is the minimum time recommendation.</td><td colspan="2">a0x1</td><td>Register write command is executed successfully.</td></tr>
<tr><td>3</td><td><GDR></td><td>Read the current data rate.</td><td colspan="2">rXXx</td><td>Current data rate (Hz) is equal to XXX</td></tr>
<tr><td>3.1</td><td><SDRXX></td><td>Set the data rate (XX) to your ideal operating data rate.</td><td colspan="2">a0x1</td><td>Register write command is executed successfully.</td></tr>
<tr><td>4</td><td><CS></td><td>Save any changes made</td><td colspan="2">a0x1</td><td>Save successful</td></tr>
<tr><td colspan="6">Once step 4 is complete, orient the sensor horizontally before sending the step 5 command.</td></tr>
<tr><td rowspan="6">5</td><td rowspan="6"><SDM5></td><td rowspan="6">Initiate horizontal taring.</td><td>1</td><td>0x10</td><td>Transition codes, you should see two.</td></tr>
<tr><td>2</td><td>0x21</td><td>Device is waiting for rolling to start.</td></tr>
<tr><td>3</td><td>0x20</td><td>Waiting code, arm is not considered fully horizontal by the device.</td></tr>
<tr><td>4</td><td>0x11</td><td>Taring begun, system is collecting data.</td></tr>
<tr><td>5</td><td>0x10</td><td>Data collection finished.</td></tr>
<tr><td>6</td><td>0x13</td><td>Checkpoint code, move the arm to vertical position.</td></tr>
<tr><td>6</td><td><STC1></td><td>Confirming vertical orientation is true, routine can resume.</td><td colspan="2">0x10</td><td>Device is computing the weight of the end-of-arm tool</td></tr>
<tr><td colspan="6">If taring was successful, outputs in the plotting tab should be centered around zero.</td></tr>
</table>

Available taring commands for sensing system types

Sensor	Command	Description
1-DOF	<SDM3>	This mode initiates the model taring process for **1-DOF model sensors**. In this mode, the module tares the process model to determine the true baselines of the sensor.
3-DOF	<SDM5>	This mode initiates the model taring for **6-DOF & 3-DOF model sensors**. In this mode, the hub will tare the mathematical model to determine the true baselines of the sensor.
6-DOF		

* These modes will automatically revert back to the previously selected modes once completed

11. Troubleshooting & FAQs

This section outlines common issues that may arise while using BonneChere and the recommended solutions. For any abnormal sensor or software behaviour not listed below, please contact customer support.

11.1. ASCII Command Errors

Symptom	Explanation	Action
Error response after sending commands.	An error response *(e0x0_)* from a SET command may be due to an invalid command or improper command format. See Error Response (ERR) for more on error register codes.	Make sure the command sent is valid: ● Commands are case sensitive - see ASCII Commands ● Ensure correct data packet structure - see ForceN ASCII Commands ● Ensure register map address is valid - see Device Register Table
Modified registers not saved.	You must save the configured settings from a SET command, otherwise, it will reset to the default.	After sending a SET command, the **<CS>** command must be sent to save the newly written register value. ● Resend the SET command in the terminal, then send <CS>. ○ SET: <SXX>, where XX is the register map address ● Check if the value has been saved using the GET command. ○ GET: <GXX>, where XX is the register map address

11.2. Force/Torque Data Errors

Symptom	Explanation	Action
Noise	Peaks in sensor readings above the specifications can be due to many different factors. The most common sources of electrical noise include: • Mechanical vibrations • Electromagnetic interference (EMI) • Radio frequency interference (RFI) • Thermal fluctuations • Grounding loops	Ensure your sensor environment is clear of any external factors to reduce noise: • Mount the sensor to stable, vibration-damped surfaces and avoid placing it near heavy machinery to minimize mechanical vibrations - see the installation section. • Use shielded cables and keep cables away from power lines, radio transmitters, and wireless devices to reduce EMI & RFI effects. • Keep your sensing system in a thermally stable environment. • Ensure cables are only shielded on one end to prevent ground loops. *Try reducing the data rates. Larger data rates will naturally experience more noise than lower ones (ie: 1 kHz data rate will be noisier than 100 Hz). See section 9 for how to change the data rate.*
Drift	Sensor drift refers to the shift of readings over time despite the actual input remaining unchanged. Sensor drift above the specifications is abnormal.	• Allow the sensor to warm up for 5+ minutes after plugging in before taking measurements to get accurate data. • Keep the sensor away from external heat/cooling sources like end effectors, pressure lines or robots to reduce thermal drift. • Keep your sensing system in a thermally stable environment. • Ensure end-effector components are rigidly fixed to the sensor and not deforming under load.
Sensor not streaming data	The sensor may be improperly connected or powered off.	• Check all cable connections and ensure the sensor is properly plugged in. • Restart the software application and reconnect the sensor to the PC by unplugging and plugging it back in. • Verify the sensor power supply is on and functioning. • Verify the communication box is connected correctly. • Inspect communication settings and protocols for correct configuration. • Test with a known good cable or port to rule out hardware faults.

Unexpected outputs / sensor inaccuracies	Data readings are not matching the expected loads. This can be due to improperly calculated expected loads, poor sensor mounting, as well as not taring the sensor.	• Ensure that you are accounting for all additional weights:
		  ○ Mounting plates & screws.
		  ○ Attached arms/tools.
		• Ensure that the correct arm length is used when calculating expected moments.
		• Ensure the sensor is properly mounted per the guidelines.
		  ○ Torque bolts to correct specification.
		  ○ Use the correct alignment pins.
		• Tare the sensor before collecting data using the ASCII commands (see section 9).

Thank you for choosing ForceN!

For additional assistance, issues or faults please contact **support@forcen.tech**

We always seek feedback to make our products great, feel free to send us comments or suggestions to **product@forcen.tech**