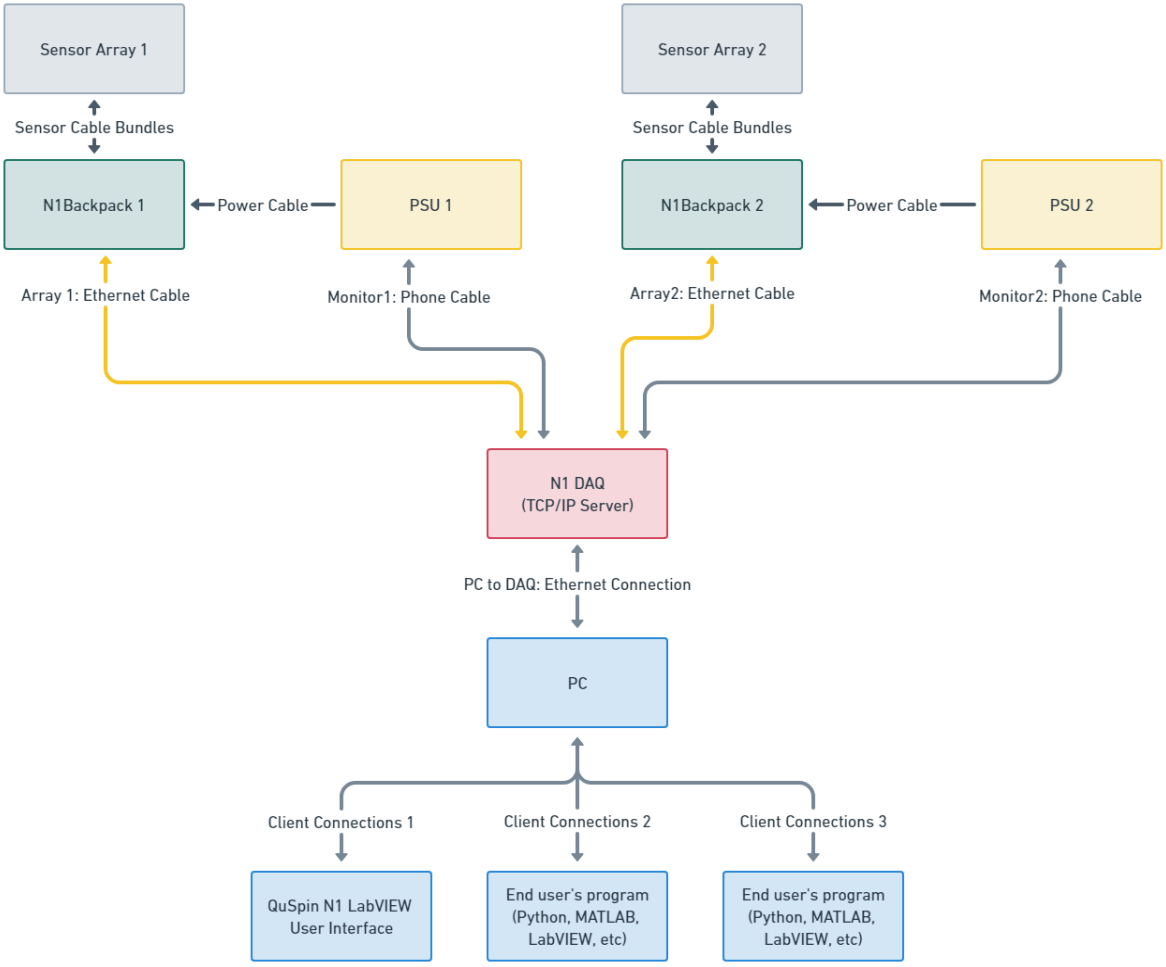
Neuro-1 API Guide V1.3

Introduction

Welcome to the Neuro-1 (N-1) API Guide; a resource for developers creating custom software solutions for controlling a Neuro-1 system via TCP/IP connections. This document details the functionalities, protocols, and methods available in our API. Whether you're developing a simple monitoring tool or a complex control system, this guide will provide you with the necessary information to integrate your software seamlessly with our instruments.

Our API is built with flexibility and ease of use in mind, enabling you to send and receive data over TCP/IP connections. This guide will walk you through the setup process, demonstrate how to establish and manage connections, and explain how to send commands and receive responses from the instrument.



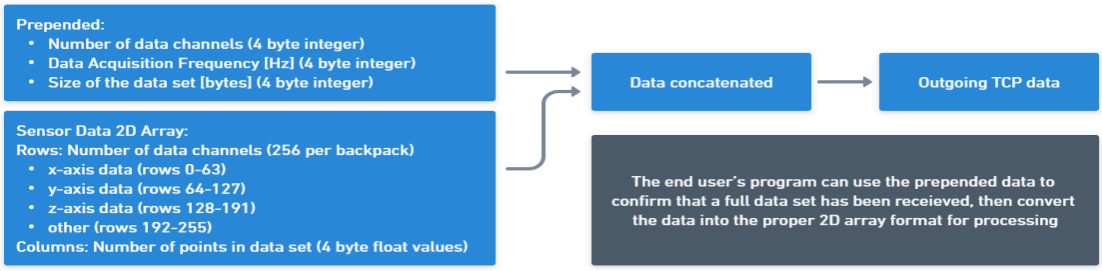
Overview of Data Flow

A basic data flow chart for the N-1 System is shown in Figure 1. Data from the sensor array is transmitted from the N-1 backpack to the data acquisition (DAQ) module for processing. The DAQ opens 4 TCP/IP server ports and allows for up to client connections to send and receive data. The first 3 ports are for transmitting sensor data, sensor status data, and system status data. The last port is for receiving command data, which are instructions for operating the N-1 System (i.e Auto Start, Field Zero, Calibrate, etc). In the following sections, we review the data structure for each of the TCP/IP ports.

Monitoring N-1 Sensor, Sensor Status, and System Status Data

All data transmitted over the N-1 TCP/IP server ports are sent using a general format consisting of three 4-byte integers prepended to a larger data packet. The three prepended values provide information about the incoming data packet, which the end user can use to reconstruct the data into a 2D array for further processing. Data flow charts showing the prepended data and the data packet are shown in Figures 2,3 and 4.

Sensor Data streamed from N-1 Server Port



X-Axis Rows 1-64

Ch1 data1	Ch1 data2	Ch1 data3	Ch1 data[n]
Ch2 data1	Ch2 data2	Ch2 data3	Ch2 data[n]
Ch3 data1	Ch3 data2	Ch3 data3	Ch3 data[n]
Ch[n] data1	Ch[n] data2	Ch[n] data3	Ch[n] data[n]

Y-Axis Rows 65-128

Ch1 data1	Ch1 data2	Ch1 data3	Ch1 data[n]
Ch2 data1	Ch2 data2	Ch2 data3	Ch2 data[n]
Ch3 data1	Ch3 data2	Ch3 data3	Ch3 data[n]
Ch[n] data1	Ch[n] data2	Ch[n] data3	Ch[n] data[n]

Z-Axis Rows 129-192

Ch1 data1	Ch1 data2	Ch1 data3	Ch1 data[n]
Ch2 data1	Ch2 data2	Ch2 data3	Ch2 data[n]
Ch3 data1	Ch3 data2	Ch3 data3	Ch3 data[n]
Ch[n] data1	Ch[n] data2	Ch[n] data3	Ch[n] data[n]

AUX Rows 193-224

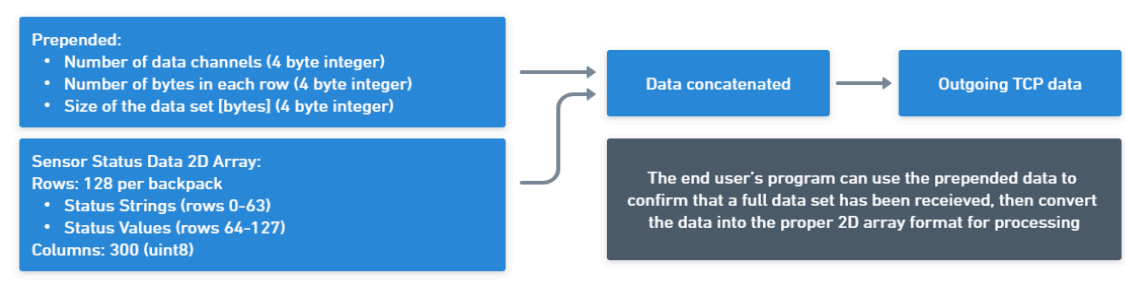
1	DI_1 data1	DI_1 data2	Ch1 DI_1 data3	DI_1 data[n]
…	…	…	…	…
11	DI_11 data1	DI_11 data2	Ch1 DI_11 data3	DI_11 data[n]
12	AI_1 data1	AI_1 data2	AI_1 data3	AI_1 data[n]
…	…	…	…	…
27	AI_16 data1	AI_16 data2	AI_16 data3	AI_16 data[n]
28	*[1]MUX_Counter1 data1	MUX_Counter1 data2	MUX_Counter1 data3	MUX_Counter1 data[n]
29	*[1]MUX_Counter2 data1	MUX_Counter2 data2	MUX_Counter2 data3	MUX_Counter2 data[n]
30	*[2]DAQ_Counter data1	DAQ_Counter data2	DAQ_Counter data3	DAQ_Counter data[n]
31	*[3]MUX_Data_Drop1 data1	MUX_Data_Drop1 data2	MUX_Data_Drop1 data3	MUX_Data_Drop1 data[n]
32	*[3]MUX_Data_Drop2 data1	MUX_Data_Drop2 data2	MUX_Data_Drop2 data3	MUX_Data_Drop2 data[n]
33-64	unused	unused	unused	unused

*[1]MUX_Counter1/2 is used to confirm no-losses in data transmitted from MUX1/2 to the N1 DAQ. This can be checked manually by subtracting adjacent data points, however the MUX_Data_Drop1/2 value also tracks if data is dropped between the MUX and DAQ by printing a value of 0 for no data lost or 1 if data loss is detected.

*[2]DAQ_Counter1 is used to confirm no-loss in data transmitted from the N1 DAQ to the PC. Each data set is stamped with a DAQ_Counter value, the first data point in each data set should be incremented by 1 from the previous data set. If a value other than 1 is detected in the subtracted data, then a data set has been lost.

*[3]MUX_Data_Drop1/2 is used to confirm that the data point is valid. A value of 0 indicates no error in that data point, a value of 1 indicates that the data point is not valid.

Sensor Status Data streamed from N-1 Server Port

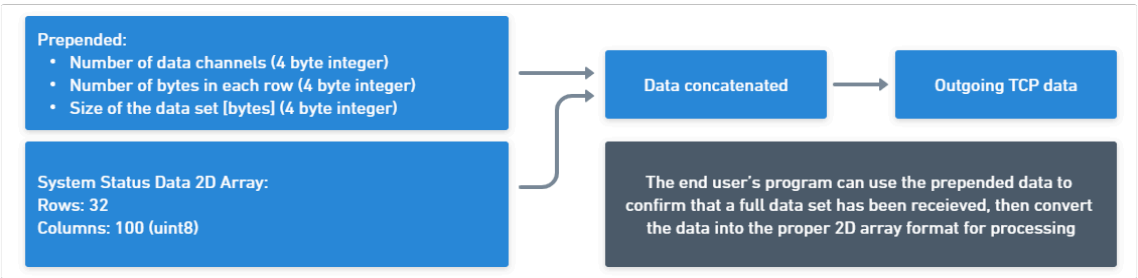


Status Strings:
Status messages from the cardmags, up to 300 characters
End of message character is carriage return (byte value 13)

Status Values Array Table:

Delimiter	Data Type	Description
ACT	Uint8	1: active to commands, 0: inactive to commands
LLS	Uint8	1: laser locked, 0: laser not locked
SLS	Uint8	1: slow lock active, 0: slow lock inactive
FLS	Uint8	1: fast lock active, 0: fast lock inactive
TLS	Uint8	1: temperature locked, 0: temperature not locked
CNT	Uint8	1: channel receiving data, 0: channel not receiving data
CBS	Uint8	1: calibration applied, 0: calibration not applied
DIS	Uint8	1: channel disabled, 0: channel enabled
BFX	Float32	Bx Field Applied [nT]
BFY	Float32	By Field Applied [nT]
BFZ	Float32	Bz Field Applied [nT]
CLX	Float32	Bx fast lock closed loop value [DAC units] Range: +- 0.6
CLY	Float32	By fast lock closed loop value [DAC units] Range: +- 0.6
CLZ	Float32	Bz fast lock closed loop value [DAC units] Range: +- 0.6
LTC	Float32	Laser temperature control value [DAC units]
CTC	Float32	Cell temperature control value [DAC units]
CTE	Float32	Cell temperature error [DAC units]
PDX	Float32	PDX DC value
PDZ	Float32	PDZ DC value
CBX	Float32	X-axis calibration factor
CBY	Float32	Y-axis calibration factor
CBZ	Float32	Z-axis calibration factor
TST	Float32	Test read

System Status Data streamed from N-1 Server Port



Descriptions of System Status Array:

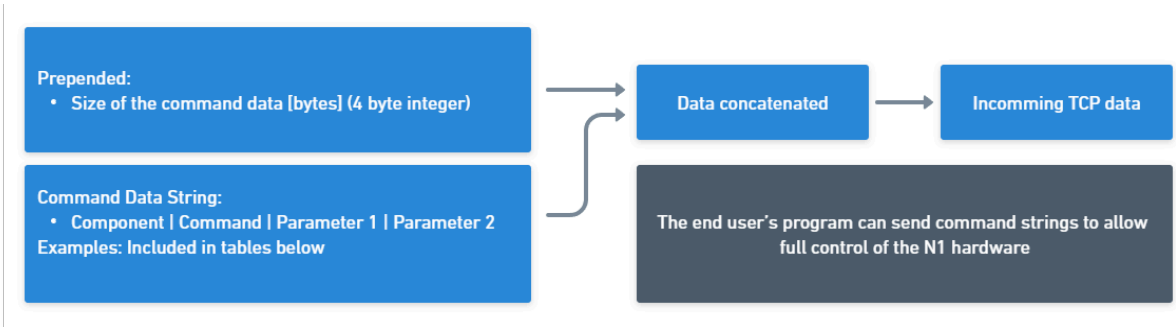
Row	Description	Values	Example Data
1	Data Acquisition Rate*[1]	1500 Hz, 750 Hz, 375 Hz	Data Acquisition Rate:1500.000 Hz
2	Acquisition Status	Running, Stopped	Acquisition Running
3	PSU1 info string	4 voltage readings, 4 current readings, 1 status byte(internal use)	PSU1:6.958 (V), 0.050 (A)| -6.166 (V), 0.031 (A)| 21.649 (V), 0.011 (A)| 6.351 (V), 0.189 (A)|126
4	PSU2 info string	4 voltage readings, 4 current readings, 1 status byte(internal use)	PSU2:6.759 (V), 0.049 (A)| -6.134 (V), 0.030 (A)| 21.165 (V), 0.010 (A)| 6.397 (V), 0.190 (A)|126
5	DAQ Status	Arrays Disconnected, Array1 Nominal, Array2 Nominal, Arrays Synced, Arrays Not Synced, Firmware Update Progress	Array1 Nominal
6	MUX1 Firmware Version	Firmware Version	MUX1 Firmware Version:Z64_M_V3_22
7	MUX2 Firmware Version	Firmware Version	MUX2 Firmware Version:Z64_M_V3_22
8	N1 DAQ CPU Usage	Core1 usage, Core2 usage	DAQ CPU Usage:40.8%,31.0%
9	Array1 Status	Array1 Connected, Array1 No Clocks, Array1 BCLK Error, Array1 LRCLK Error, Array1 LRCLK/BCLK Error	Array1 Connected
10	Array2 Status	Array2 Connected, Array2 No Clocks, Array2 BCLK Error, Array2 LRCLK Error, Array2 LRCLK/BCLK Error	Array2 Connected
12	Connection Status	Sys, Cmd, Status, Sensor	Sys Data Connected, Cmd Data Connected, Status Data Connected, Sensor Data Connected
13	N1 DAQ Firmware Version	Firmware Version	N1 DAQ Firmware Version: N1_DAQ_V3_03_(01-11-2024)

*[1] Data Acquisition Rate: Sensor data can be downsampled from the DAQ to the PC. The downsampled data is averaged to avoid aliasing and loss of resolution.

Controlling the N-1 System from the TCP/IP Command Server Port

All control functionality provided by the N-1 user interface is also available to external users through client connections to the command port.
Commands are constructed using the general format: Component|Command|Parameter 1|Parameter 2, where “component” refers to either Sensor, DAQ, or PSU. The “command” input is the name of the command (i.e Auto Start). The “parameter 1/2” inputs are used for additional inputs for certain commands. The command data must be prepended with a 4-byte value of the size of the command data that is being sent (in bytes).
A complete list of commands, and their example usages are shown in the Tables below.

Client Command Port Command Structure



Command Table (with examples)

Sensor Operations: Basic commands

Command	Description	Required Parameter	Optional Parameter	Example
Auto Start	Initiate sensor auto-start routine	N/A	N/A	Sensor|Auto Start
Reboot	Reboot the entire sensor array			Sensor|Reboot
Read Sensor Keys	Retrieve sensor key numbers			Sensor|Read Sensor Keys
Reconfigure Array	Activates and updates all channel numbers			Sensor|Reconfigure Array
Coils Reset	Erase field coil settings			Sensor|Coils Reset
Field Zero ON	Enable field zeroing routine			Sensor|Field Zero ON
Field Zero OFF	Disable field zeroing routine			Sensor|Field Zero OFF
Reset Ortho & Calibration Values	Erase all sensor orthogonalization and calibration values			Sensor|Reset Ortho & Calibration Values
Ortho & Calibrate	Engage orthogonalization and calibration routine			Sensor|Ortho & Calibrate
Closed Loop ON	Enable closed loop routine for all sensors			Sensor|Closed Loop ON
Closed Loop OFF	Disable closed loop routine for all sensors			Sensor|Closed Loop OFF
All ON (Change to Activate All)	Activate all sensors			Sensor|Activate All
All OFF (Change to Deactivate All)	Deactivate all sensors			Sensor|Deactivate All
Module A ON/OFF	Turn on/off all sensors in module A	Array number: 1 or 2	0 = off, 1 = on, toggle if no parameter listed	Sensor|Module A ON/OFF|1|0
Module B ON/OFF	Turn on/off all sensors in module B			Sensor|Module B ON/OFF|2|1
Module C ON/OFF	Turn on/off all sensors in module C			Sensor|Module C ON/OFF|2|
Module D ON/OFF	Turn on/off all sensors in module D			Sensor|Module D ON/OFF|1|1
Module E ON/OFF	Turn on/off all sensors in module E			Sensor|Module E ON/OFF|2|0
Module F ON/OFF	Turn on/off all sensors in module F			Sensor|Module F ON/OFF|1|0
Module G ON/OFF	Turn on/off all sensors in module G			Sensor|Module G ON/OFF|1|0
Module H ON/OFF	Turn on/off all sensors in module H			Sensor|Module H ON/OFF|1|0
Active Sensors	Activate sensors specified by the range	Range of sensor numbers, 1-128 (use commas or dashes to separate)		Sensor|Active Sensors|2,6,7-55|1
Disable Sensor	Disables a sensor specified by the user. Note: disabled sensors no longer receive commands including firmware updates. Use “Activate Sensor” command to re-enable a previously disabled sensor, or “Reconfigure Array” to enable and activate all sensors.	Number of sensor to disable		Sensor|Disable Sensor|1
Activate Sensor	Enables a sensor specified by the user.	Number of sensor to activate		Sensor|Activate Sensor|1

Sensor Operations: Advanced commands (no required parameters)

Command	Description	Example
Cycle inc value	Change the increment size for Bx/By/Bz ++/-- commands	Sensor|Cycle inc value
Bx++	Increment field on Bx coil ++	Sensor|Bx++
Bx--	Increment field on Bx coil --	Sensor|Bx--
By++	Increment field on By coil ++	Sensor|By++
By--	Increment field on By coil --	Sensor|By--
Bz++	Increment field on Bz coil ++	Sensor|Bz++
Bz--	Increment field on Bz coil --	Sensor|Bz--
Bx MOD OFF	Turn off Bx coil modulation	Sensor|Bx MOD OFF
By MOD OFF	Turn off By coil modulation	Sensor|By MOD OFF
Bz MOD OFF	Turn off Bz coil modulation	Sensor|Bz MOD OFF
All B MOD OFF	Turn off Bx,By,Bz coil modulations	Sensor|All B MOD OFF
Bx MOD ON	Turn on Bx coil modulation	Sensor|Bx MOD ON
By MOD ON	Turn on By coil modulation	Sensor|By MOD ON
Bz MOD ON	Turn on Bz coil modulation	Sensor|Bz MOD ON
All B MOD ON	Turn on Bx,By,Bz coil modulations	Sensor|All B MOD ON
Only Ortho	Run orthogonalization procedure	Sensor|Only Ortho
Only Calibrate	Run calibration procedure	Sensor|Only Calibrate
Reset Ortho Values	Reset only the orthogonalization parameters	Sensor|Reset Ortho Values
Reset Calibration Values	Reset only calibration parameters	Sensor|Reset Calibration Values

DAQ and PSU Operations (no required parameters)

Command	Description	Example
Set Frequency: 1500 Hz	Sets the data sampling rate to 1500 Hz	DAQ|Set Frequency: 1500 Hz
Set Frequency: 750 Hz	Sets the data sampling rate to 750 Hz	DAQ|Set Frequency: 750 Hz
Set Frequency: 375Hz	Sets the data sampling rate to 375 Hz	DAQ|Set Frequency: 375 Hz
PSU Enable Outputs	Enables all power supplies	PSU|PSU Enable Outputs
PSU Disable Outputs	Disables all power supplies	PSU|PSU Disable Outputs