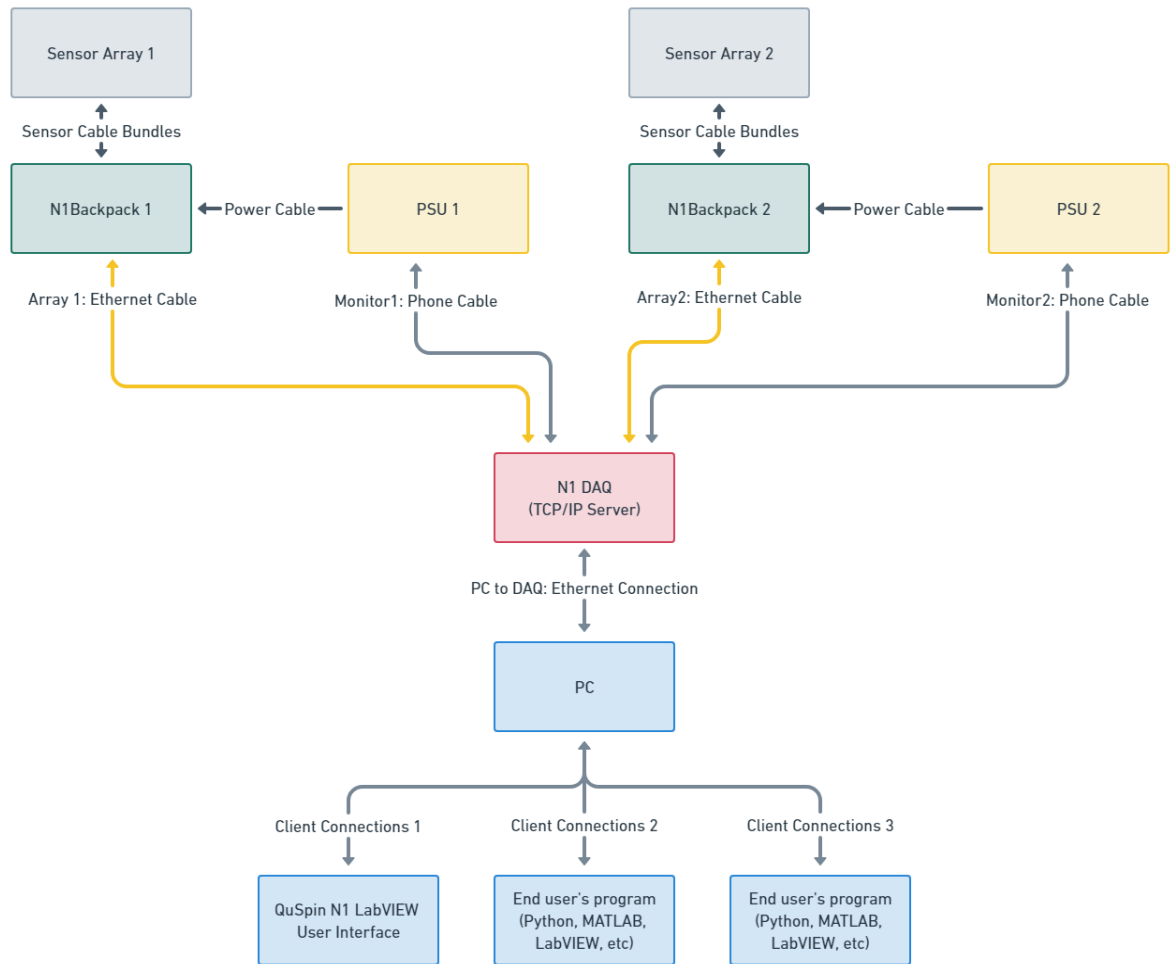


Neuro-1 API Guide V1.6

Introduction

Welcome to the Neuro-1 (N1) API Guide; a resource for developers creating custom software solutions for controlling an N1 system via TCP/IP connections. This document details the functionalities, protocols, and methods available in our API. Whether developing simple monitoring tools or complex control systems, this guide will provide the necessary information to integrate custom software seamlessly with the N1 system. This guide will cover the setup process, establishing and managing connections, and sending and receiving data.



System Diagram

Overview of Data Flow

Data from the sensor array is transmitted from the N1 backpack to the data acquisition (DAQ) module for processing. The DAQ opens 4 TCP/IP server ports and allows for up to 10 client connections to each port to send and receive data. The first 3 ports (8089,8090,8091) are for transmitting sensor data, sensor status data, and system status data. The last port (8092) is for receiving command data, which are instructions for operating the N1 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.

DAQ Transmitted Data Format

The DAQ sends out the following data to be read by any connected TCP clients:

- **Sensor Data (port:8089):** This is the X,Y, Z magnetic field data from each sensor, and AUX data (digital triggers, analog inputs, data counter, etc (discussed further in a later section). The data type is float32 (IEEE 754 binary32 format).
- **Sensor Status Data (port:8090):** This includes information about the status of each sensor, and includes information such as if auto start is complete, field zero has been run, the sensor has been successfully calibrated, etc (discussed further in a later section). The data type is uint8_t.
- **System Status Data(port:8091):** This includes information about the system such as the sensor data acquisition rate, firmware versions of specific components, power supply information, etc (discussed further in a later section). The data type is an array of uint8_t.

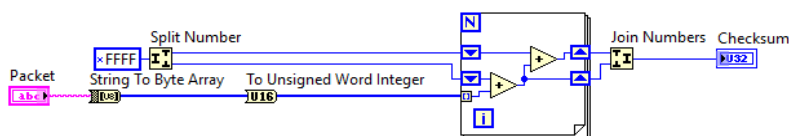
Transmission intervals:

- **Sensor Data:** Is sent from the DAQ approximately every 42 milliseconds (Number of sensor data points / Acquisition Rate), the number of sensor data points transmitted is set by the data acquisition rate (1500 Hz sends sensor 64 data points, 750 Hz sends 32 sensor data points, 375 Hz sends 16 sensor data points). Note: Data is downsampled (averaged) when lower acquisition rates are selected to preserve data quality.
- **Sensor Status Data:** Is sent from the DAQ approximately every 333 milliseconds.
- **System Status Data:** Is sent from the DAQ approximately every 333 milliseconds.

General Data Frame Structure:

`uint32_t Frame_Start | uint32_t Frame_Number | uint32_t Payload_Size | uint32_t Param1 | uint32_t Param2 | Payload | uint32_t Payload_End | uint32_t Checksum | uint32_t Frame_End`

- **Frame Start:** 4 bytes corresponding to 'BLCK'
- **Frame_Number:** 4 bytes representing the frame number
- **Payload_Size:** 4 bytes representing the size of the payload
- **Param1:** Explained in following sections
- **Param2:** Explained in following sections
- **Payload:** Sensor data / Sensor Status Data / System Status data
- **Payload_End:** 4 bytes corresponding to 'BEND'
- **Checksum:** 4 byte Fletcher Checksum value



- **Frame_End:** 4 bytes corresponding to 'BLCK'

Monitoring N1 Sensor, Sensor Status, and System Status Data

All data transmitted from the N1 DAQ TCP/IP server ports are sent using the frame format described in the previous section.

Sensor Data Frame Structure: Port 8089

uint32_t Frame_Start | uint32_t Frame_Number | uint32_t Payload_Size | uint32_t Number_of_Sensors | uint32_t Acquisition_Rate | Payload | uint32_t Payload_End | uint32_t Checksum | uint32_t Frame_End

- Frame Start: 4 bytes corresponding to 'BLCK'
- Frame_Number: 4 bytes representing the frame number
- Payload_Size: 4 bytes representing the size of the payload
- Number_of_Sensors: Number of sensor channels (default 64)
- Acquisition_Rate: Sampling rate of sensor data from the DAQ [Hz]
- Payload: Sensor data
- Payload_End: 4 bytes corresponding to 'BEND'
- Checksum: 4 byte Fletcher Checksum value
- Frame_End: 4 bytes corresponding to 'BLCK'

Sensor Data Payload Format:

Sensor Data can be read as a 3D array with the following dimensions:

- Pages: 4 (X-Data, Y-Data, Z-Data, AUX-DATA)
- Rows: number_of_sensors (default is 64)
- Columns: acquisition_rate [Hz] * 0.042667 [seconds]
 - Acquisition_rate 1500 Hz = 64 columns
 - Acquisition_rate 750 Hz = 32 columns
 - Acquisition_rate 375 Hz = 16 columnsNote: columns are the number of data points for each channel

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	*1MUX_Counter1 data1	MUX_Counter1 data2	MUX_Counter1 data3	MUX_Counter1 data[n]
29	*1MUX_Counter2 data1	MUX_Counter2 data2	MUX_Counter2 data3	MUX_Counter2 data[n]
30	*2DAQ_Counter data1	DAQ_Counter data2	DAQ_Counter data3	DAQ_Counter data[n]
31	*3MUX_Data_Drop1 data1	MUX_Data_Drop1 data2	MUX_Data_Drop1 data3	MUX_Data_Drop1 data[n]
32	*3MUX_Data_Drop2 data1	MUX_Data_Drop2 data2	MUX_Data_Drop2 data3	MUX_Data_Drop2 data[n]
33-64	unused	unused	unused	unused

*1MUX_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*3 values 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.

*2DAQ_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.

*3MUX_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 Frame Structure: Port 8090

uint32_t Frame_Start | *uint32_t* Frame_Number | *uint32_t* Payload_Size | *uint32_t* Number_of_Sensors | *uint32_t* Total_Bytes | Payload | *uint32_t* Payload_End | *uint32_t* Checksum | *uint32_t* Frame_End

- Frame Start: 4 bytes corresponding to 'BLCK'
- Frame_Number: 4 bytes representing the frame number
- Payload_Size: 4 bytes representing the size of the payload
- Number_of_Sensors: Number of sensor channels (default 64)
- Total_Bytes: 600 bytes total. 300 for Status Strings, 300 for Status Values
- Payload: Sensor data
- Payload_End: 4 bytes corresponding to 'BEND'
- Checksum: 4 byte Fletcher Checksum value
- Frame_End: 4 bytes corresponding to 'BLCK'

Sensor Status Payload Format:

Sensor Status Data can be read as a 3D array with the following dimensions:

- Pages: 2 (Status Strings, Status Values)
- Rows: Number_of_Sensors (default is 64)
- Columns: Total_Bytes / 2

Status Strings: End of message characters are carriage return, new line (byte values 13, 10)

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
STS	Uint32_t	Additional status info* ¹

*¹Additional status info:

Bit [31-0]	Value	Description
31 [MSB]	0,1	1 triaxial mode 0 dual-axis mode
30	0,1	1 light check passed 0 light check not passed
29	0,1	1 auto start procedure is active 0 auto start procedure is not active
28	0,1	1 auto start has completed 0 auto start has not completed
27	0,1	1 field zero is active 0 field zero is not active
26	0,1	1 fields have been applied from field zero process 0 fields have not been applied from field zero process
25	0,1	1 calibration procedure is active 0 calibration procedure is not active
24	0,1	1 calibration has been applied and is within normal range 0 calibration is not been applied or is outside normal range
23	0,1	1 Bx Mod ON 0 Bx Mod OFF
22	0,1	1 By Mod ON 0 By Mod OFF
21	0,1	1 Bz Mod ON 0 Bz Mod OFF
20	0,1	1 Slow Closed Loop ON 0 Slow Closed Loop OFF
19	0,1	1 Bx Slow Lock ON 0 Bx Slow Lock OFF
18	0,1	1 By Slow Lock ON 0 By Slow Lock OFF
17	0,1	1 Bz Slow Lock ON 0 Bz Slow Lock OFF
16	0,1	1 Bx Fast Lock ON 0 Bx Fast Lock OFF
15	0,1	1 By Fast Lock ON 0 By Fast Lock OFF
14	0,1	1 Bz Fast Lock ON 0 Bz Fast Lock OFF
13-0	unused	Reserved for future status info

System Status Data Frame Structure: Port 8091

`uint32_t Frame_Start` | `uint32_t Frame_Number` | `uint32_t Payload_Size` | `uint32_t Rows` | `uint32_t Columns` | `Payload` | `uint32_t Payload_End` | `uint32_t Checksum` | `uint32_t Frame_End`

- Frame Start: 4 bytes corresponding to 'BLCK'
- Frame_Number: 4 bytes representing the frame number
- Payload_Size: 4 bytes representing the size of the payload
- Rows: number of rows (32)
- Columns: number of columns (100)
- Payload: Sensor data
- Payload_End: 4 bytes corresponding to 'BEND'
- Checksum: 4 byte Fletcher Checksum value
- Frame_End: 4 bytes corresponding to 'BLCK'

System Status Payload Format:

System Status Data can be read as a 2D array with the following dimensions:

- Rows: number of rows (32)
- Columns: number of columns (100)

Descriptions of System Status Array:

Row	Description	Values	Example Data
1	Data Acquisition Rate*	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), firmware version	PSU1:6.4 (V), 0.1 (A) -5.7 (V), 0.1 (A) 27.4 (V), -0.1 (A) 5.7 (V), 0.3 (A) 126, V2_2
4	PSU2 info string	4 voltage readings, 4 current readings, 1 status byte(internal use), firmware version	PSU2:6.3 (V), 0.1 (A) -5.6 (V), 0.1 (A) 27.2 (V), -0.1 (A) 5.6 (V), 0.3 (A) 126, V2_2
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_24
7	MUX2 Firmware Version	Firmware Version	MUX2 Firmware Version:Z64_M_V3_24
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
11	Connection Status	Data Connections: x, Status Connections: x, System Connections: x, Command Connections: x	Data Connections: 1, Status Connections: 1, System Connections: 1, Command Connections: 1
12	N1 DAQ Firmware Version	Firmware Version	N1 DAQ Firmware Version: N1_DAQ_V3_03_(01-11-2024)
13	Command Validation	Valid Command Received [x], ERROR: Invalid Command Received [x]	Valid Command Received [25856]

*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.

Command Data: Port 8092

All control functionality provided by the N1 user interface is also available to external users through client connections to the command port.

Client Command Port Command Structure

Frame Structure:

`uint32_t Payload_Size | Payload`

- Payload size: 4 bytes representing the size of the payload
- Payload: Command data

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.

Command Tables (with examples)

Sensor Operations: Basic commands

Command	Description	Required Parameter	Optional Parameter	Command Data Examples
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	Reset applied fields with internal coils			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 1
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 the "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.	Channel number of sensor to activate		Sensor Activate Sensor 1
Deactivate Sensor	Disables a sensor specified by the user.	Channel number of sensor to deactivate		Sensor Deactivate 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

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