Medtronic	NEUROMODULATION CONFIDENTIAL	DOCUMENT/RECORD

Identifier	Version	Author
NRP1088-35790	8.0	Bourget, Duane; Ahrens, Terry

Title: Interface Specification, Nexus-D System	**Pages:** (including this page) 37

APPROVALS

Signed By	Responsibility	Date/Time (GMT)
Bourget, Duane	Project/System/Mfg Lead	09/15/2014 07:16:53 PM
Isaacson, Benjamin	SW Design Lead	09/16/2014 12:34:55 PM
Johnsen, Tom	Regulatory Affairs Approver	09/16/2014 03:20:57 PM
Carter, Steve	Quality System Approver	09/17/2014 02:17:52 PM

Nexus-D System

Interface Specification

CAUTION – Investigational device. Limited by Federal (or United States) law to investigational use.

Table of Contents

Table of Figures

Table of Tables

1 INTRODUCTION

The Nexus-D System acts as the communication conduit between a Host and an Activa family neurostimulator (Activa PC+S, Activa PC, Activa RC, and Activa SC). Throughout this document, the term “Activa PC+S” is typically used to describe the Nexus-D System interface. Refer to Table 11 – Nexus-D System Commands to determine the instances where this term can also apply to an Activa PC, RC or SC. The Nexus-D System physically interfaces to the Host via a USB connection.

This specification applies to both the Nexus-D System and the Nexus-D2 System. The Nexus-D2 System supports all the Nexus-D System functionality. New functionality that is only supported by the Nexus-D2 system will be noted. The Nexus-D family is defined by the application version (see Table 1 - Nexus-D System Family Versions). The version is displayed when batteries are replaced in the Streaming Telemetry Module (STM). A “Splash Screen” will display for several seconds after battery replacement with the following information:

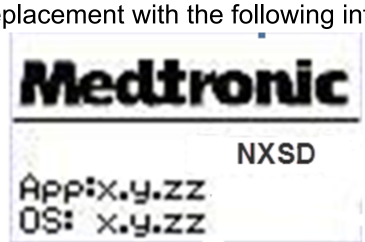

Table 1 - Nexus-D System Family Versions

System Designation	App Version
Nexus-D	App: 2.6
Nexus-D2	App: 2.7

1.1 PURPOSE

This document specifies the Nexus-D System interface and defines how a Host establishes a communication link with a Nexus-D System and uses the Nexus-D System to communicate with the Activa family of neurostimulators.

1.2 SCOPE

This document covers the communication link between a Host and the Nexus-D System. This specification does not cover the interface (e.g. – telemetry protocols) between the Nexus-D System and the INS or any interface contained within the Nexus-D System itself.

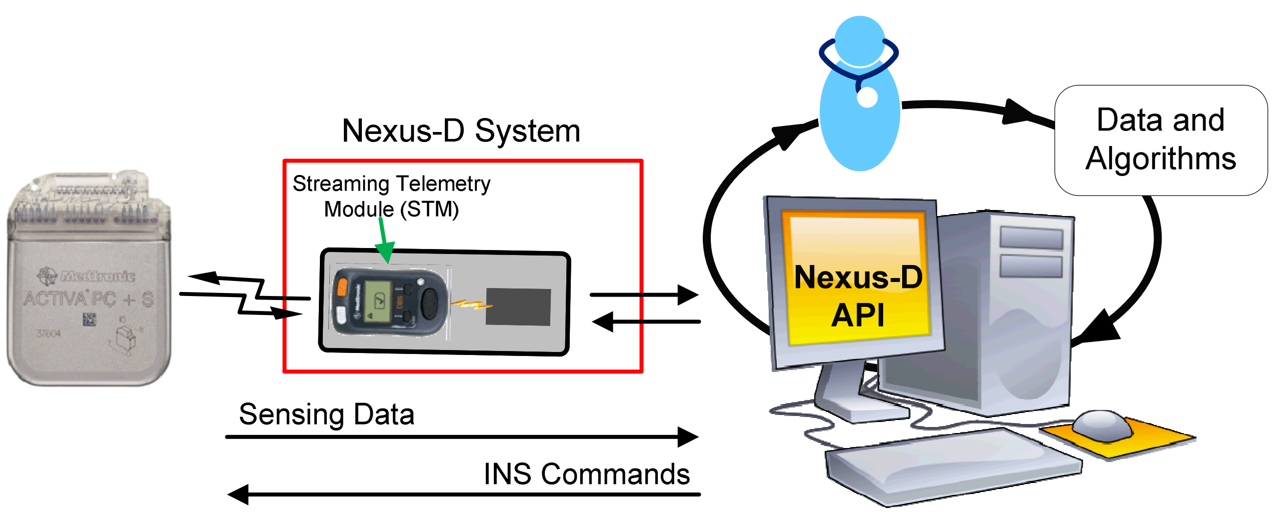

Figure 1-1 – *Nexus-D System Components*

1.3 ACRONYMS AND DEFINITIONS

Acronyms and Terms used in this document are listed in the following table.

Term/Acronyms	Definition
Activa PC	Primary Cell Implantable Neurostimulator without sensing capabilities
Activa PC+S	Primary Cell Implantable Neurostimulator with sensing capabilities
Activa RC	Rechargeable Implantable Neurostimulator without sensing capabilities
Activa SC	Single Channel Implantable Neurostimulator without sensing capabilities
Connected	A state in which two instruments are able to send data back and forth.
Host	Another name for the computer running the algorithm in the context of this document.
IMD	Implanted Medical Device
INS	Implantable Neurostimulator
Master	An instrument that instigates both the connection and communication
Slave	An instrument that awaits a connection/communication from a master.
STM	Streaming Telemetry Module
STS	Streaming Telemetry Software
LD	Linear Discriminant (LD) algorithm. This algorithm evaluates sensing data and determines which side of a linear boundary it falls.
Linked	The Nexus-D System has successfully communicated with the INS and saved its ID for this session.

1.4 APPLICABLE DOCUMENTS

The following is a list of all documents and other sources of information referenced in this document.

Number	Version	Title
NRP1088-35854	2.0	Nexus-D System USB Driver Installation Guide

2 Interface Configuration

To enable communication between the Host and the Nexus-D System, a USB driver must be installed to enable the Nexus-D System to enumerate when physically connected to the Host. Refer to the Nexus-D System USB Driver Installation Guide, NRP1088-35854, for details on how to install and configure the Nexus-D System USB driver on the Host.

3 Interface Definition

The following sections will describe the message sequencing requirements for connection establishment, message formation, and status reporting.

3.1 Interface Overview

When the Nexus-D System is physically connected to the Host, the Nexus-D System enumerates as a virtual com port on the Host; therefore, the Host must use serial communications for transmitting and receiving data to and from the Nexus-D System. When the host opens the serial virtual com port, the parameters used are as follows for both send and receive:

Baud Rate = 38,400
8 Data bits
No parity
1 stop bit
No CTS/RTS flow control

When the Nexus-D System and the Host are communicating, the Host acts as the master at all times. The communication protocol takes on the form of command/response.

3.2 Frames

The same basic frame is used for all communication between the Host and the Nexus-D System.

3.2.1 Frame format

The basic frame format used for all communication frames is shown in Table 2 - Frame Format. It is composed of a header, payload, and payload CRC. Big Endian byte ordering (most significant byte first) will be used for multi-byte numeric fields in the header and payload data (e.g. - Frame ID, Payload Length, CRC).

Table 2 - Frame Format

Field	Frame Header	Payload	Payload CRC
# of Bytes	10	Variable	2

3.2.1.1 FRAME HEADER FORMAT

The frame header format is shown in Table 3 - Frame Header Format. The frame header contains information about the payload as well as frame identification and CRC checks for the header data. The header has its own CRC value, so that in the event of a large payload, the header can be verified and the payload length can be determined before the entire payload is received.

Table 3 - Frame Header Format

Byte	0	1	2	3	4-5	6-7	8-9
Field	Version	Source	Frame Type	ACK/NAK Code	Frame ID	Payload Length	Header CRC
# of Bits	8	8	8	8	16	16	16

Version – The version of the frame. The version is currently restricted to a single value:

Table 4: Version

Code	Description
0x01	Initial Frame Format

The use of a single allowable value assists in re-synching after a missed character or other link error.

Source – The source indicates who sent the original message and can have the following values:

Table 5: Source

Code	Description
0x00	Message is from a Host
0x01	Message is from a Nexus-D System

The frames have different payload types based on source of the frame.

Frame Type –For the Nexus-D System, the frame type will always be 0x01.

Table 6: Frame type

Code	Description
0x01	Application Frame

ACK/NAK Code – This field indicates success or problems encountered when processing the associated request. Codes other than Success (00) are NAK codes and are used only when the message is a response (see 3.2.2.2 Response Payload). For all command payloads (see 3.2.2.1 Command Payload), only Success (00) is used. The possible ACK/NAK code values are the following:

Table 7: ACK/NAK Codes

Code	Description
0x00	Success (ACK)
0x01	Payload CRC Error (NAK)
0x02	Invalid Frame Type (NAK)
0x03	Message Incomplete or Error (NAK)
0x04	Duplicate (repeated) Frame ID (NAK)
0x05	Invalid Payload Length (NAK)
0x06	Header CRC Error (NAK)
0x07	Previous Command Busy(NAK)
0x08	INS POR Indicated (NAK)
0x09	Battery Depleted (NAK)

Frame ID – Frame IDs are 16 bit counters seeded at 0. For real-time data responses, the two bytes of the Frame ID will contain the actual Activa PC+S sequence numbers for the two real time data blocks. For non-real time data command/response format, the response frame will send back the same Frame ID as in the original command.

NAKs contain the same frame ID as that of the frame being NAK'd.

The Nexus-D System does not check Frame IDs of successive messages for out-of-order or missing Frame IDs. The Nexus-D System does check for duplicate frame IDs. If a duplicate frame ID is received, the NAK from the Nexus-D System will carry a NAK code of 0x04 for "Duplicate Frame ID".

Payload Length – Identifies the length of the payload, including its 2 byte CRC value.

Header CRC – The CRC value of the header data. The CRC consists of the calculation on the data from the Version through the Payload Length. The CRC used will be a CRC-16-CCITT using an $x^{16} + x^{12} + x^5 + 1$ polynomial reversed (i.e. Truncated polynomial = 0x8408). The CRC uses a seed value of FFFF (hex) and is complemented after the calculation.

Example:
Header without checksum
01 00 01 00 00 0B 00 04 CRC (8 bytes) = 969A (seed = FFFF hex)

Header with checksum
01 00 01 00 00 0B 00 04 69 65 (69x = ~96x; 65x = ~9Ax)

The calculation can be implemented in code several ways, here is an example:

```
int ComputeCRC16(int crc, int byte)
{
   int w_val = byte & 0xff;
   int tmp = crc ^ w_val;
   int newcrc;

   newcrc =  ((((tmp<<12) & 0xf000) | ((tmp>>4) & 0x000f)) ^
              ((((tmp<< 8) & 0xff00) | ((tmp>>8) & 0x00ff)) ^
               ((( tmp<< 7) & 0x0780) ^
                ((( tmp    ) & 0x000f) ^
                 ((tmp<<3) & 0x07f8)))));
   return newcrc & 0x0ffff; // Mask so it is only 16 bits
}

int CalculateBufferCRC(unsigned char *buffer, unsigned int offset, unsigned int cnt)
{
   int CRCVal = 0xFFFF;   // Initialize CRC value
   unsigned int j;
   char value;

   for(j = 0; j < cnt; j++) {  // update the CRC value with each byte in the data
      value = buffer[offset++];
      CRCVal = ComputeCRC16(CRCVal, (int)value);
   }

   CRCVal = 0x0ffff-CRCVal;      // compliment the result
   return CRCVal;
}
```

3.2.2 PAYLOAD TYPES

3.2.2.1 COMMAND PAYLOAD

A command payload is used by the Host to send a command to the Nexus-D System. The Command Payload format is shown in Table 8 – Command Payload Format.

Table 8 – Command Payload Format

Field	Command	Command Parameters	Payload CRC
# of Bytes	2	Variable	2

The commands and command parameters are defined in Table 11 – Nexus-D System Commands. The first bit of the first command byte indicates the direction of the command (i.e. – 0 when from Host to the Nexus-D System and 1 when from the Nexus-D System to Host). This first bit is explained more in Table 11 – Nexus-D System Commands.

The Payload CRC is calculated from the payload. The CRC used will be a CRC-16-CCITT using an $x^{16} + x^{12} + x^5 + 1$ polynomial seeded with FFFF (hex). The CRC is stored with bytes complemented (see example with header CRC in section 3.2.1.1 Frame Header Format).

3.2.2.2 RESPONSE PAYLOAD

A Response payload is used by the Nexus-D System to send a response to the Host. The Response Payload format is shown in Table 9 – Response Payload Format.

Table 9 – Response Payload Format

Field	Response Command	Response Code	Response	Payload CRC
# of Bytes	2	1	Variable	2

The Response Command is the original command sent by the Host with the highest bit set.

The individual (non-zero) NAK codes in the header of the response frame are intended for debug purposes.

The Response Codes are defined in section Table 10 - Response Codes.

The Response is unique to the command.

The Payload CRC is calculated from the payload. The CRC used will be a CRC-16-CCITT using an $x^{16} + x^{12} + x^5 + 1$ polynomial seeded with FFFF (hex). The CRC is stored with bytes complemented (see example with header CRC in section 3.2.1.1 Frame Header Format).

3.3 TIMEOUTS

The Nexus-D System does not implement timeouts. Instead, variable timeouts for each command are specified in this specification to allow the Host to implement timeouts to add robustness to the physical interface. The Host can implement timeouts to retry commands which do not get timely responses.

Certain physical interfaces may benefit from checking for interruptions of data streams (gaps of time detected in the middle of received frames). Since the Nexus-D System sends and receives binary data (i.e. - no reserved characters for the protocol), an out-of-sync condition caused by a dropped or garbled character may be difficult to recover from because any character could be interpreted as the start of a new message. The Host should detect time gaps with no receipt of characters as a break between frames; thereby, allowing for easier re-synchronization.

4 Implementation Specifications

This section describes the details of the Nexus-D System implementation.

4.1 Startup Command Flow

An example startup command flow is shown in Figure 4-1 - Example Startup Flow. The first command from the Host should be Get Status to determine that the Nexus-D System is finished with its startup and ready for normal operation.

After the first Get Status command, the Nexus-D System will attempt to bond with an Activa PC+S device. In addition to reading the INS ID which is used to communicate to a specific INS, the Nexus-D System will read the Activa PC+S status (e.g. – sensing state, stimulation state) and keep this data available for the Host to request as needed.

The Nexus-D System will not start a Maintenance Session (i.e. – automatic retrieval of real-time data, see section 4.3.2 Maintenance Session for more details) until the Start Real-Time command is received from the Host.

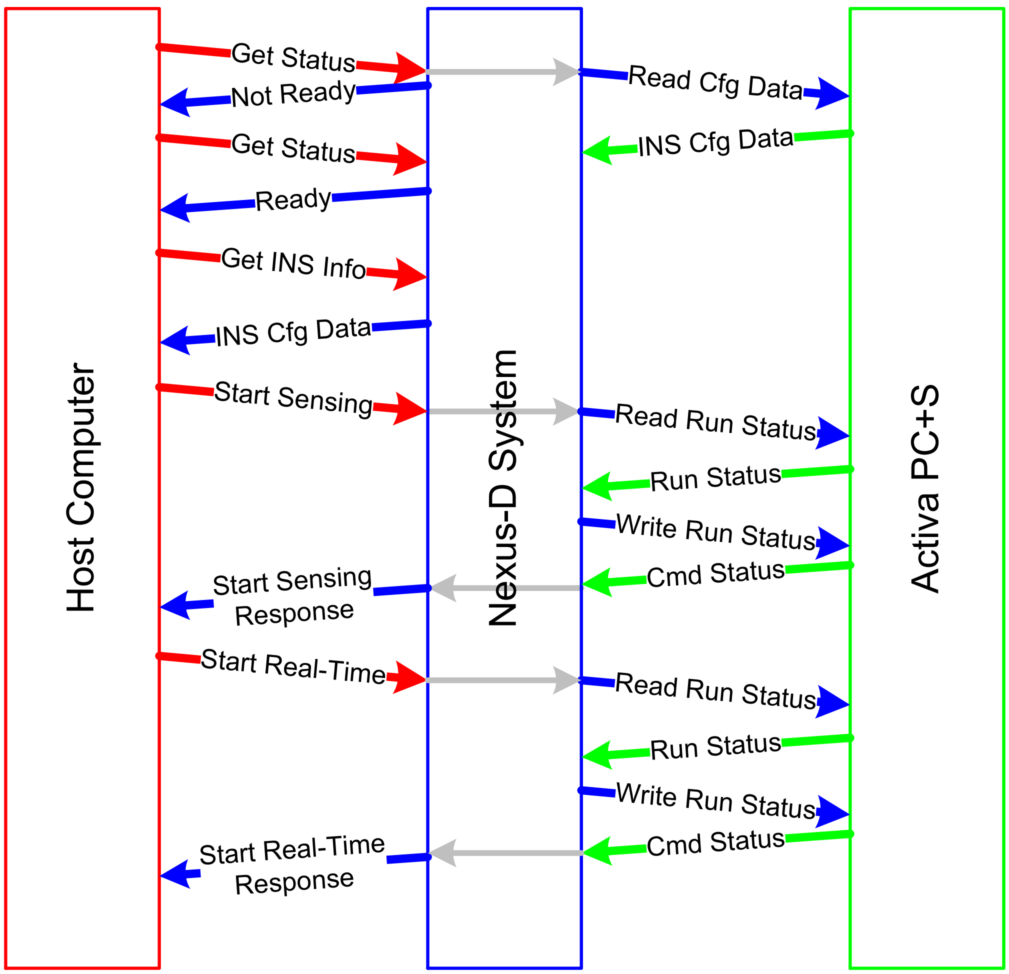

Figure 4-1 - Example Startup Flow

4.2 Real-Time Data Command Flow

An example startup command flow is shown in Figure 4-2 - Example Real-Time Command Flow.

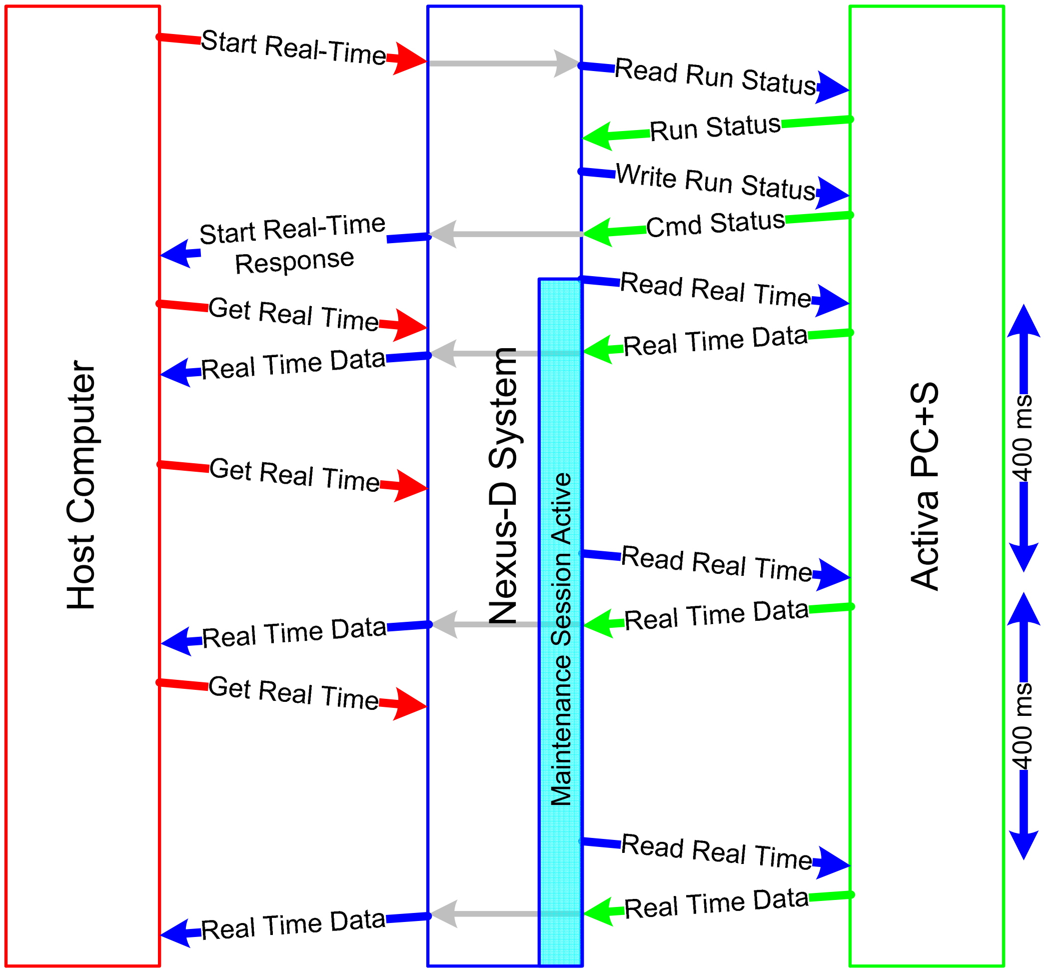

Figure 4-2 - Example Real-Time Command Flow

4.3 Sessions

For purpose of discussion, the Nexus-D System defines two session types: Supervisory and Maintenance. A Supervisory Session covers the Host-initiated commands. The Maintenance Session covers the Nexus-D System initiated commands used to automatically request the real-time data packets from the Activa PC+S INS. The following sections describe the sessions in more detail.

NOTE: It is not possible to initiate a maintenance session with an Activa PC, Activa RC, or Activa SC INS.

4.3.1 Supervisory Session

A Supervisory Session is defined by the establishment and continuation of a communication link between the Host and the Nexus-D System. In a Supervisory Session, all commands are initiated by the Host.

The Host should not send a new command until a response message from the previous command is received or the previous command specific timeout period has expired. Violating this rule may result in a collision of the new command and the previous command response, causing both to be lost.

4.3.2 MAINTENANCE SESSION

A maintenance session is activated when the Host sends a “Start Real-Time” command and the Activa PC+S INS is configured properly for real-time data output (i.e.- INS Filtering is not on).

During a Maintenance Session, the Nexus-D System will automatically request a real-time data packet from the Activa PC+S INS every 400ms. The Nexus-D System will buffer the received real-time data until the Host requests the data. If another real-time data packet is received from the Activa PC+S INS before the Host requests the data, the old data will be over-written.

The Host may request real-time data at any time when a Maintenance Session is active; if the data is ready in the Nexus-D System buffer, it will be sent to the Host immediately. If the Host requests the real-time data before it is ready, it will be sent to the Host as soon as it is ready in the Nexus-D System buffer. If the real-time data in the Nexus-D System buffer has already been sent to the Host when a real-time data request is received by the Nexus-D System, the Nexus-D System will wait for the next real-time data packet to be received from the Activa PC+S INS and then send that real-time data to the Host. If the real-time data packet sent from the Nexus-D System is corrupted in some way, that data is lost.

When a maintenance session is active, it will end if one of the following events occurs:

- If a “Stop Sensing” request is received from the Host, the Nexus-D System terminates the maintenance session.
- If the Maintenance Session timer expires, the Nexus-D System terminates the real-time uplink mode in the INS, restores the INS sensing state to the state it was before the maintenance session was started (i.e. enabled or disabled), and terminates the maintenance session..
- If an “Stop Real-Time” request is received from the Host, the Nexus-D System terminates the real-time uplink mode in the INS, restores the INS sensing state to the state it was before the maintenance session was started (i.e. enabled or disabled), and terminates the maintenance session.
- If a Power Off key event is received, the Nexus-D System will terminate the real-time uplink mode in the INS, restore the INS sensing state to the state it was before the maintenance session was started (i.e. enabled or disabled), and terminate the maintenance session.
- If the Nexus-D System battery becomes depleted, the Nexus-D System will attempt to terminate the real-time uplink mode in the INS, attempt to restore the INS sensing state to the state it was before the maintenance session was started (i.e. enabled or disabled), and terminate the maintenance session.
- If the Activa PC+S battery level becomes too low (i.e. below 2.75v), the maintenance session will end.

4.4 COMMAND RESPONSE CODES

The possible responses codes in a Response Frame are shown in Table 10 - Response Codes.

Table 10 - Response Codes

Code Dec(hex)	Description
00(0x00)	Success
10(0x0a)	Program number out of range
11(0x0b)	Program set number out of range

Code Dec(hex)	Description
12(0x0c)	Invalid program in set
13(0x0d)	Specified program invalid
14(0x0e)	Specified program set invalid
15(0x0f)	Stimulation therapy not on
18(0x12)	No EEPROM backup; application not saved
20(0x14)	Specified program not in active set
21(0x15)	No active set defined (the null set should never be used in an Activa family INS)
22(0x16)	Amplitude at upper limit
23(0x17)	Amplitude at lower limit
24(0x18)	Pulse width at upper limit
25(0x19)	Pulse width at lower limit
26(0x1a)	Rate at upper limit
27(0x1b)	Rate at lower limit
29(0x1d)	The specified program number (INC Amplitude, Pulse Width) or a program in the Program Set (INC Set Rate) is at maximum stack and out of regulation (OOR). Use the “Cycle Stim” command to reset the OOR condition.
30(0x1e)	Previous program change not yet delivered
32(0x20)	Reject – Device reset
50(0x32)	Set transition in progress
51(0x33)	Invalid amplitude parameter
54(0x36)	Specified program set not the active set
59(0x3b)	Null Set active
101(0x65)	Not Ready to Execute Command
102(0x66)	Invalid Command ID (i.e. unsupported command)
103(0x67)	Invalid Command Data
105(0x69)	Real Time not active
106(0x6a)	Sense Configuration Error
107(0x6b)	No Telemetry Response (i.e. for a telemetry type command)
108(0x6c)	Telemetry Error
109(0x6d)	INS Filtering On (i.e. real-time will not start if filtering is enabled in the Activa PC+S)
110(0x6e)	INS POR Indicated
111(0x6f)	INS has responded unexpectedly while up linking real time data (zero length data, etc.)
112(0x70)	INS has responded with null packets for too long while up linking real-time data
114(0x72)	The INS did not execute all the batch commands
115(0x73)	Command not valid for connected device (i.e. attempt to use an Activa PC+S command when connected to an Activa PC, RC or SC INS)
232(0xe8)	Reject - I2C/PIC: Interface busy
233(0xe9)	Reject - I2C/PIC: Invalid length
234(0xea)	Reject - I2C/PIC: No response
235(0xeb)	Reject - I2C/PIC: Read back failure
241(0xf1)	Reject - PIC: Busy
242(0xf2)	Reject - PIC: Invalid command type
243(0xf3)	Reject - PIC: Invalid command parameter

Code Dec(hex)	Description
244(0xf4)	Reject - PIC: Unsupported command
245(0xf5)	Reject - PIC: Invalid address (Flash/EEPROM/SRAM)
246(0xf6)	Reject - PIC: Test in progress

4.5 COMMANDS

The commands supported by the Nexus-D System are shown in Table 11 – Nexus-D System Commands. The commands added to the Nexus-D2 System are shown in Table 12 – Nexus-D2 System Commands.

Table 11 – Nexus-D System Commands

Note: All commands are used with Activa PC+S. Refer to right-hand column to determine if a command is used with the Activa PC, RC, or SC.

Command Name	Payload data type	Data content	Response Timeout in ms	Activa PC, SC, RC
Get INS Info	0x00 01	None	2000	PC, SC, RC
Get INS Info Response	0x80 01	22 bytes: (See section 4.8.5 for an explanation of the data) (1) Response Code (2) Therapy State (0=off, 1=on) (3) Sensing State (0=disabled, 1=enabled, 2=not available i.e. Activa PC, RC or SC device) (4) Active Group Number (5) Active Group Rate high byte (6) Active Group Rate low byte (7) 1st Active Program (L1) Number (8) 1st Active Program (L1) Amplitude (9) 1st Active Program (L1) Amplitude Resolution (10)1st Active Program (L1) Pulse Width (11)2nd Active Program (R2) Number (12)2nd Active Program (R2) Amplitude (13)2nd Active Program (R2) Amplitude Resolution (14)2nd Active Program (R2) Pulse Width (15)3rd Active Program (L2) Number (16)3rd Active Program (L2) Amplitude (17)3rd Active Program (L2) Amplitude Resolution (18)3rd Active Program (L2) Pulse Width (19)4th Active Program (R1) Number (20)4th Active Program (R1) Amplitude (21)4th Active Program (R1) Amplitude Resolution (22)4th Active Program (R1) Pulse Width		
Set Nexus Configuration	0x00 02	2 bytes: (1) Maintenance Session Timeout (1 – 30 sec) (2) Host Session Timeout (1 – 15 min)	250	PC, SC, RC
Set Nexus Configuration Response	0x80 02	1 byte: Response Code		

Command Name	Payload data type	Data content	Response Timeout in ms	Activa PC, SC, RC
Start Sensing	0x00 03	None	250	No
Start Sensing Response	0x80 04	1 byte: Response Code		
Stop Sensing	0x00 04	None	250	No
Stop Sensing Response	0x80 03	1 byte: Response Code		
Start Real-Time	0x00 05	1 byte: TD Channel (1 or 3) when run speed is 422 Hz (See section 4.8.3)	800	No
Start Real-Time Response	0x80 05	1 byte: Response Code		
Stop Real-Time	0x00 06	None	250	No
Stop Real-Time Response	0x80 06	1 byte: Response Code		
Get Status	0x00 08	None	250	PC, SC, RC
Get Status Response	0x80 08	8 bytes: (See section 4.8.4) (1) Response Code (e.g.: Not Ready) (2) Nexus State 0=Idle (INS not connected) 1=Linking to INS 2=Link Failed No Response (antenna location) 3=Link Failed Device Error (check INS with an 8840) 4=Host Supervisory Mode (INS is connected) 5=Maintenance Mode (real-time uplink active) (3) STS App Version Major (4) STS App Version Minor (5) Battery % (6) Battery Depleted (true/false) (7) Host Timeout Value (in minutes) (8) Maintenance Timeout Value (in seconds)		
Get Real Time data	0x00 0c	None	500	No
Real Time data	0x80 0c	Variable number of bytes: Response Code, followed by data as described in Figure 4-4 – Real-Time Data Format. (See section 4.8.7)		
Cycle Stim	0x00 0d	None	500	PC, SC, RC
Cycle Stim Response	0x80 0d	1 byte: Response Code		
Reset Cycle	0x00 0e	None	500	PC, SC
Reset Cycle Response	0x80 0e	1 byte: Response Code		
Therapy On	0x01 01	None	300	PC, SC, RC
Therapy On Response	0x81 01	1 byte: Response Code		
Therapy Off	0x01 02	None	300	PC, SC, RC
Therapy Off Response	0x81 02	1 byte: Response Code		

Command Name	Payload data type	Data content	Response Timeout in ms	Activa PC, SC, RC
Set Active Program Group	0x01 B6	1 byte: Group Number (1-4)	300	PC, SC, RC
Set Active Response	0x81 B6	1 byte: Response Code		
INC Amplitude	0x01 BD	3 byte: (See section 4.8.1) (1) Program Number (1-16) (2) Number of Steps (1-40) (3) Number of Repeats (0-20)	300 to 1400 depending on number of repeats	PC, SC, RC
INC Amplitude Response	0x81 BD	3 bytes: (See section 4.8.1) (1) Response Code (2) New Value of Amplitude (3) Number of INC's executed		
DEC Amplitude	0x01 BE	3 byte: (See section 4.8.1) (1) Program Number (1-16) (2) Number of Steps (1-40) (3) Number of Repeats (0-20)	300 to 1400 depending on number of repeats	PC, SC, RC
DEC Amplitude Response	0x81 BE	3 bytes: (See section 4.8.1) (1) Response Code (2) New Value of Amplitude (3) Number of DEC's Executed		
INC Pulse Width by 10uS	0x01 BF	1 byte: Program Number (1-16) (See section 4.8.2)	300	PC, SC, RC
INC Pulse Width Response	0x81 BF	2 bytes: (See section 4.8.2) (1) Response Code (2) New Pulse Width		
DEC Pulse Width by 10uS	0x01 C0	1 byte: Program Number (1-16) (See section 4.8.2)		PC, SC, RC
DEC Pulse Width Response	0x81 C0	2 bytes: (See section 4.8.2) (1) Response Code (2) New Pulse Width		
INC Set Rate - Increase the rate by 1hz if current rate is < 10Hz, otherwise by 5hz.	0x01 C3	1 byte: Group # (1-4)	300	PC, SC, RC
INC Set Rate Response	0x81 C3	3 bytes: (1) Response Code (2) New Rate high byte (See section 4.8.5 for rate value description) (3) New Rate low byte		

Command Name	Payload data type	Data content	Response Timeout in ms	Activa PC, SC, RC
DEC Set Rate – Decrease rate by 1hz if current rate is < 10Hz, otherwise by 5hz.	0x01 C4	1 byte: Group # (1-4)	300	PC, SC, RC
DEC Set Rate Response	0x81 C4	3 bytes: (1) Response Code (2) New Rate high byte (See section 4.8.5 for rate value description) (3) New Rate low byte		
Restore Clinician Settings	0x01 C6	1 byte: Program Group # (0-4, 0 = all) Note: "Stim Off" and “Restore Clinician Settings” commands are sent to the INS in one transaction. (See section 4.8.8)	400	PC, SC, RC
Restore Clinician Settings Response	0x81 C6	1 byte: Response Code		
Set Trigger	0x01 D8	None	400	No
Set Trigger Response	0x81 D8	1 byte: Response Code		

Table 12 – Nexus-D2 System Commands

Note: When using the Nexus-D System, the response to these commands added in the Nexus-D2 System will be 102 (unsupported command).

Command Name	Payload data type	Data content	Response Timeout in ms	Activa PC, RC, SC
Get Sense Status	0x00 07	None	500	No
Get Sense Status Response	0x80 07	6 bytes: (See section 4.8.11) (1) Response Code (e.g.: Not Ready) (2) Sense State 0 = Program Mode 1 = Standard Run Mode (Sensing enabled) 2 = Montage Run Mode (3) Detection State 0 = Detection Disabled 1 = Detection Enabled (4) LD State (true/false) (5) In Detect (true/false) (6) In Cluster Detect (true/false)		
Stim Window	0x00 0f	3 byte: (See section 4.8.10) (1) Program Number (1-16) (2) Number of Steps (1-40) (3) Number of Delays (0-20)	700	PC, SC, RC
Stim Window Response	0x80 0f	3 bytes: (See section 4.8.10) (1) Response Code (2) New Value of Stimulation Amplitude (3) Stimulation Resolution (1 is Voltage any other value is Current)		

4.6 COMMAND PARAMETERS

All commands contain the command ID in the 1st two bytes of the payload (i.e. the “Payload data type”). If any parameters are required, they follow per the description in Table 11 – Nexus-D System Commands.

Commands with parameters indicated as “Group # (1-4)” or “Program Number (1-16)” are mapped to the 8840 configuration as indicated in Table 13 - Group and Program Number Mapping. For example, the 8840 Group A maps to 1 and Group D maps to 4. For programs, the 8840 uses program number 1 for Group A Left Tab L1 (See Figure 4-3 - 8840 Group and Program Examples).

Table 13 - Group and Program Number Mapping

	8840 Program Layout			
Group	L1	L2	R2	R1
A(1)	1	2	3	4
B(2)	5	6	7	8
C(3)	9	10	11	12
D(4)	13	14	15	16

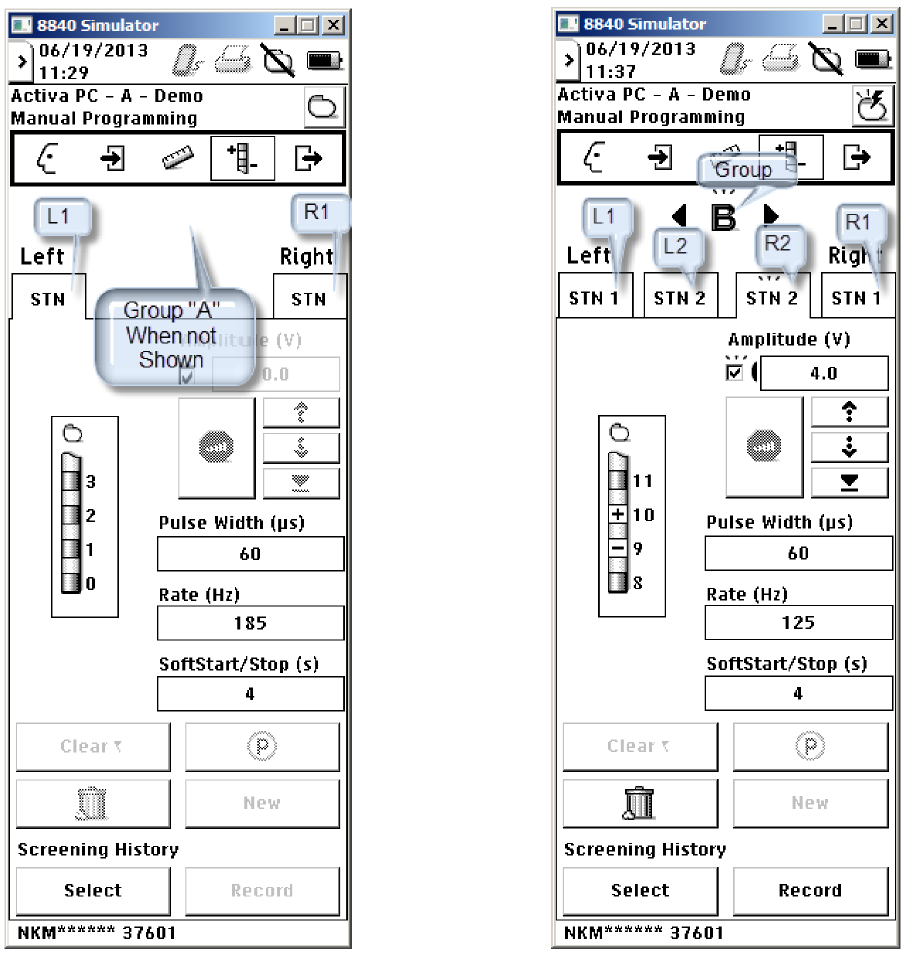

Figure 4-3 - 8840 Group and Program Examples

4.7 COMMAND RESPONSES

All command responses contain the response command code in the 1st two bytes of the payload (i.e.- payload data type) and a response code in the 3rd byte of the payload. Any remaining payload data is unique to the command.

4.8 COMMAND EXPLANATIONS

Some commands are simple and require no further explanation. Commands that may need more detail are listed here.

4.8.1 INC AND DEC AMPLITUDE COMMANDS

The INC Amplitude and DEC Amplitude commands use three parameters:

- One byte for the Program Number as described in Table 13 - Group and Program Number Mapping
- A one byte number of steps where a step for constant current is 100uA and a step for constant voltage is either 0.05V or 0.1V as configured for the specified program using the 8840
- A one byte number of repeats, i.e. the number of times the INC or DEC will repeat. If the repeat count is 0, the command will execute only one time; if the repeat count is 1, the command will execute 2 times; etc. The repeat is used to moderate the speed at which the stimulation is increased or decreased, but there are some implementation limitations. There is an interlock in the Activa PC, SC, RC and PC+S that will ignore the next INC command if the previous INC has not been delivered to the patient. When the group rate period is too low, not all of the repeated INC commands will be executed because they are attempted before the previous INC is delivered to the patient. At group rates less than 100Hz, the number of repeats that actually will execute is not predictable, but the count of commands that do execute is returned in the command response. Note: the Host must calculate its timeout value based on the number of times the increment or decrement will execute (e.g. the Host timeout should be 100ms + (70ms * repeat count)).

The response for both the INC and DEC Amplitude commands contain:

- Payload Data Type – two bytes containing 0x81BD for INC or 0x81BE for DEC
- Response Code – one byte value defined in Table 10 - Response Codes
  - If the INC command returns a response code 29(0x1d), use the Cycle Stim command to clear the condition.
- New value of amplitude – one byte encoded in 50mV steps for Voltage or 100uA steps for Current
- Number of INC or DEC commands executed – the actual number executed may not match the number requested as explained above

4.8.2 INC AND DEC PULSE WIDTH COMMANDS

The “INC Pulse Width” and “DEC Pulse Width” commands use one parameter, the Program Number, as described in Table 13 - Group and Program Number Mapping. These commands will change the pulse width by one step of 10 micro seconds.

4.8.3 START REAL-TIME COMMAND

The "Start Real-Time" command has one parameter, but it is only relevant when the sensing time domain sample rate is set to 422Hz and both TD channels are enabled. In this configuration, only one of the TD channels, 1 or 3, can be used for real-time uplink data. Therefore, when the sample rate is 422Hz and both TD channels are active, the parameter is used to select which channel (1 or 3) to use for the streaming real-time data.

4.8.4 GET STATUS COMMAND

The “Get Status” command will return the current status of the Nexus-D System. This command should be the first command used in a Supervisory session. There will be no response if the Nexus-D System is initializing or displaying the startup “Splash Screen”. If the Nexus-D System is not linked to an INS, when the command is received, it will initiate the linking process. The Host can continue to send the “Get Status” command until the Nexus-D System is linked to the INS. The response contains:

- Payload Data Type – two bytes containing 0x8008
- Response Code – one byte always 0 (Success)
- Nexus State – one byte as follows:
  - 0=Idle - this is only returned for the first command in a Supervisory session, the linking process has been initiated by this “Get Status” command.
  - 1=Linking to INS – the processes of linking to an INS is in progress, this may take up to 2 seconds.
  - 2=Link Failed No Response – no INS responded to the link request. Adjust the antenna and send a new “Get Status” command to try the process again.
  - 3=Link Failed Device Error (check INS with an 8840) - a device responded but is in a state not compatible with the Nexus-D System.
  - 4=Supervisory Session - The INS is connected and the Nexus-D System is waiting for commands.
  - 5=Maintenance Session – The Activa PC+S is streaming real-time data to the Nexus-D System.
- STS App version major – one byte indicating the version of the active STS application.
- STS App version minor – one byte indicating the version of the active STS application.
- Battery % - one byte indicating the percent of battery life remaining (25, 50, 75, 100 or 255(0xff)). Note: a value of 255 indicates the battery level is unknown; this will occur when there has not been a 30 second interval between commands to do a battery reading in the previous 10 minutes.
- Battery depleted – one byte where 0 indicates not depleted and any other value indicates depleted
- Supervisory timeout value – one byte indicating the number of minutes that a Supervisory Session will remain active after the last Host command is received by the Nexus-D System.
- Maintenance timeout value – one byte indicating the number of seconds that a Maintenance Session will remain active after the last “Get Real Time data” command is received by the Nexus-D System.

Data Exchange Example:

***Get Status command sent by host*:**

```
01 00 01 00 00 01 00 04 1A 1F    00 08 83 0F
^  ^  ^  ^  ^     ^     ^        ^     ^
|  |  |  |  |     |     |        |     |-> Payload CRC
|  |  |  |  |     |     |        |->Payload type 0008 = get Status
|  |  |  |  |     |     |-> Header CRC
|  |  |  |  |     |->Payload Length in bytes
|  |  |  |  |->Message Sequence Number
|  |  |  |->ACK/NAK code (always 0 when sending a command)
|  |  |->Frame Type (always 1 "Application Frame")
|  |->Source (always 0 "Clinician machine" when sending a command)
|->Message format version (always 1)
```

***Get Status response from the Nexus-D System*:**

```
01 01 01 00 00 01 00 0C 09 82  80 08 00 00 02 01 64 00 02 0A CA B4
^  ^  ^  ^  ^     ^     ^      ^     ^  ^  ^  ^  ^  ^  ^  ^  ^
|  |  |  |  |     |     |      |     |  |  |  |  |  |  |  |  |->Payload CRC
|  |  |  |  |     |     |      |     |  |  |  |  |  |  |  |->Maint. Timeout
|  |  |  |  |     |     |      |     |  |  |  |  |  |  |->Super. Timeout
|  |  |  |  |     |     |      |     |  |  |  |  |  |->Battery Depleted
|  |  |  |  |     |     |      |     |  |  |  |  |->Battery Percent
|  |  |  |  |     |     |      |     |  |  |  |->App Minor Version
|  |  |  |  |     |     |      |     |  |  |->App Major Version
|  |  |  |  |     |     |      |     |  |->Nexus State
|  |  |  |  |     |     |      |     |->Response code 0=Ok
|  |  |  |  |     |     |      |->Payload type 8008 = get Status response
```

```
|   |   |   |   |      |      |-> Header CRC
|   |   |   |   |      |->Payload Length in bytes
|   |   |   |   |->Message Sequence Number
|   |   |   |->ACK/NAK code (0=Ok otherwise look at Table 7: ACK/NAK Codes)
|   |   |->Frame Type (always 1 "Application Frame")
|   |->Source (always 1 "Nexus-D System" when it is response data)
|->Message format version (always 1)
```

4.8.5 GET INS INFO COMMAND

The “Get INS Info” command will return the current values for the active set of parameters in use in the INS. The response contains:

- Payload Data Type – two bytes containing 0x8001
- Response Code – if the response code is not 0 (Success) then the remaining data may not be current and should not be used.
- Therapy state (0=off, 1=on)
- Sensing state (0=disabled, any non-zero value = enabled)
- Active group number (1 – 4)
- Active group rate high byte and low byte – a 16bit value representing the current rate at which therapy is delivered. This value is in 10us units. Therefore, a value of 50000 means therapy is delivered every 0.5 seconds and is expressed as 2Hz. Here is a table with the values used to calculate the steps for an increment or decrement rate command:
    - 50000 = 2Hz
    - 33333 = 3Hz
    - 25000 = 4Hz
    - 20000 = 5Hz
    - 16667 = 6Hz
    - 14286 = 7Hz
    - 12500 = 8Hz
    - 11111 = 9Hz
    - 10000 = 10Hz
    - 6667 = 15Hz
    - 5000 = 20Hz
    - 4000 = 25Hz
    - 3333 = 30Hz
    - 2857 = 35Hz
    - 2500 = 40Hz
    - 2222 = 45Hz
    - 2000 = 50Hz
    - 1818 = 55Hz
    - 1667 = 60Hz
    - 1538 = 65Hz
    - 1429 = 70Hz
    - 1333 = 75Hz
    - 1250 = 80Hz
    - 1176 = 85Hz
    - 1111 = 90Hz
    - 1053 = 95Hz
    - 1000 = 100Hz
    - 952 = 105Hz
    - 909 = 110Hz
    - 870 = 115Hz
    - 833 = 120Hz
    - 800 = 125Hz
    - 769 = 130Hz

        - 741 = 135Hz
        - 714 = 140Hz
        - 690 = 145Hz
        - 667 = 150Hz
        - 645 = 155Hz
        - 625 = 160Hz
        - 606 = 165Hz
        - 588 = 170Hz
        - 571 = 175Hz
        - 556 = 180Hz
        - 541 = 185Hz
        - 526 = 190Hz
        - 513 = 195Hz
        - 500 = 200Hz
        - 488 = 205Hz
        - 476 = 210Hz
        - 465 = 215Hz
        - 455 = 220Hz
        - 444 = 225Hz
        - 435 = 230Hz
        - 426 = 235Hz
        - 417 = 240Hz
        - 408 = 245Hz
        - 400 = 250Hz
    - 1st active program (L1) number – 0 indicates not active; any other number is the index of the program, refer to Table 13 - Group and Program Number Mapping for possible values
    - 1st active program (L1) amplitude – encoded in 50mV steps for Voltage or 100uA steps for Current
    - 1st active program (L1) amplitude resolution – 1 indicates Voltage, any other number indicates Current
    - 1st active program (L1) pulse width - encoded in 10uSec steps
    - 2nd active program (R2) number – encoding is the same as 1st active program
    - 2nd active program (R2) amplitude – encoding is the same as 1st active program
    - 2nd active program (R2) amplitude resolution – encoding is the same as 1st active program
    - 2nd active program (R2) pulse width – encoding is the same as 1st active program
    - 3rd active program (L2) number – encoding is the same as 1st active program
    - 3rd active program (L2) amplitude – encoding is the same as 1st active program
    - 3rd active program (L2) amplitude resolution – encoding is the same as 1st active program
    - 3rd active program (L2) pulse width – encoding is the same as 1st active program
    - 4th active program (R1) number – encoding is the same as 1st active program
    - 4th active program (R1) amplitude – encoding is the same as 1st active program
    - 4th active program (R1) amplitude resolution – encoding is the same as 1st active program
    - 4th active program (R1) pulse width – encoding is the same as 1st active program

Note: the active programs are returned in the order that the stimulation pulses are delivered.

4.8.6 Set Trigger Command

The “Set Trigger” command may respond with the rather generic response code 235 (0xeb) which generally indicates that Sensing is not active (i.e. this command is only available when sensing is on). When recording is enabled, a second “Set Trigger” command should not be sent for at least 5 seconds after the first one; any “Set Trigger” commands within 5 seconds of the first will not be acted upon by the Activa PC+S. The exception to this rule is when the Activa PC+S is in "Log Only" mode and not in the process of sending real-time uplink. In this case, the entire Activa PC+S memory is designated for Log events so back-to-back External Triggers are allowed.

4.8.7 Get Real-Time Data Command

The “Get Real-Time Data” command may be sent at any time during a maintenance session. The Nexus-D System will wait for the next real-time data to be retrieved from the Activa PC+S if real-time data is not ready to be sent to the host when the command is received from the host. The layout of the response for the "Get Real-Time Data" command is shown in Figure 4-4 – Real-Time Data Format.

Channel data is only present in the message payload if the channel is enabled. The sampling rate and channel type (i.e. – power or time domain) determine the number of bytes in the payload data. All channels shall be in the order specified in Figure 4-4 – Real-Time Data Format regardless of settings. There are two Activa PC+S data patterns decoded into each message payload (see Sample Patterns in Figure 4-4 – Real-Time Data Format); the actual Activa PC+S one byte sequence numbers for these patterns are in the message header "Frame ID" bytes (i.e. the first byte is pattern 1 sequence number and the second byte is pattern 2 sequence number). Note: pattern sequence numbers are sequentially incremented from 1 through 255. When 255 is used, the pattern sequence number will cycle back to 1. The pattern sequence numbers are used to detect missed packets due to timing.

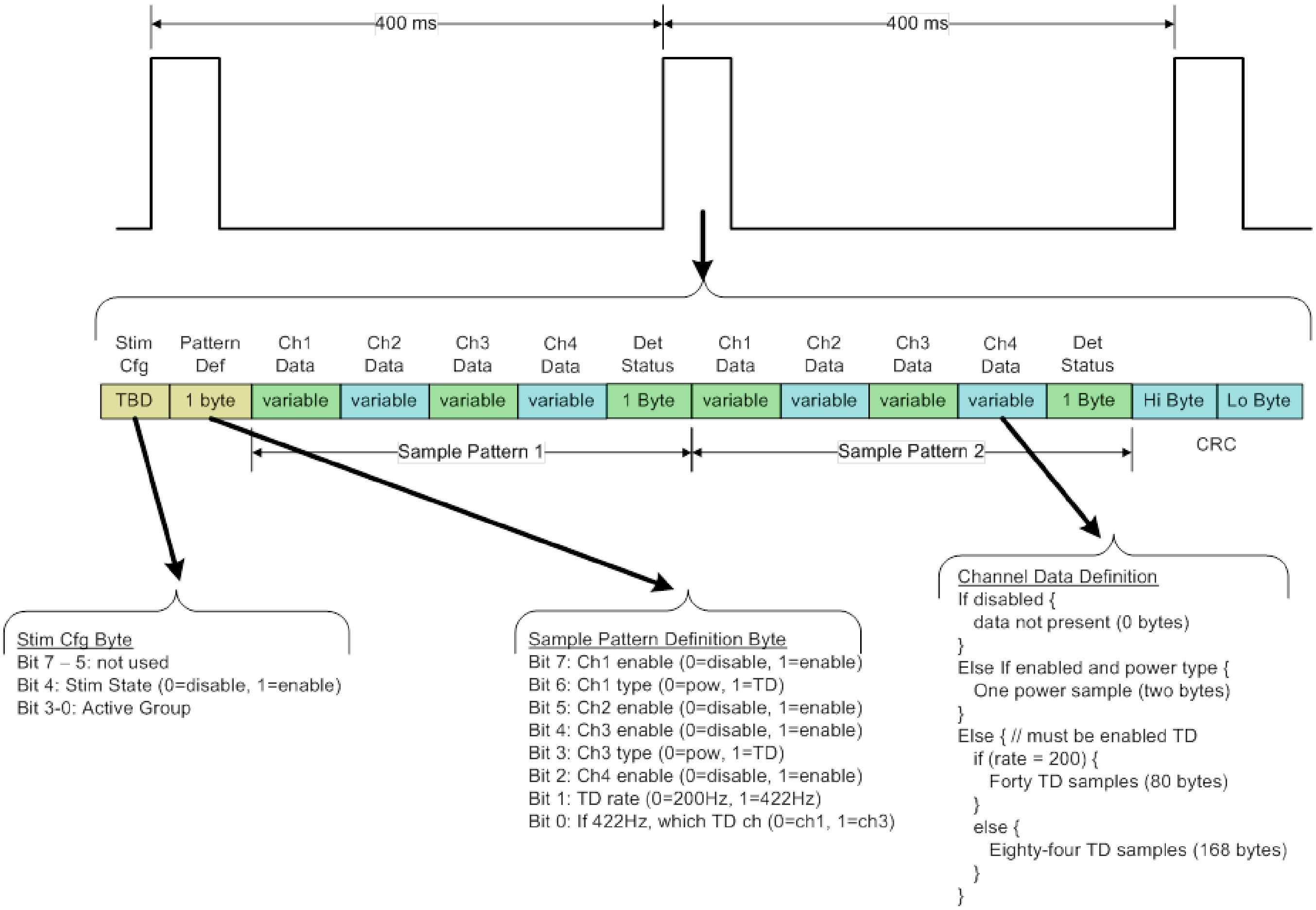

Figure 4-4 – Real-Time Data Format

The response payload contains:

- Two bytes containing the Payload Data Type – 0x800C
- One byte Response Code – if this code is not 0 (Success), the real-time data will not be present.
- One byte Stim Configuration – the low 4 bits contain the Active Group and bit 4 indicates the therapy state (1 = on, 0 = off)
- One byte data pattern definition byte (i.e. the key for decoding the remaining bytes). See Figure 4-4 – Real-Time Data Format for the Pattern Definition Byte layout.
- Two sample patterns follow – Each of these patterns contain one decoded Activa PC+S 200ms packet of data and each packet contains up to 4 channels of data. The number of data bytes for each channel depends on the channel data type. There are three general types of real-time data:
    1) Power – a two byte field containing the 10-bit power channel reading
    2) 200Hz TD data – 40 16 bit signed TD values. There may be 1 or 2 TD channels represented in a packet.
    3) 422Hz TD data – 84 16 bit signed TD values. There is only one TD channel represented in a packet. Bit 0 of the Sample Pattern Definition Byte indicates which TD channel this is.

4.8.8 RESTORE CLINICIAN SETTINGS COMMAND

The “Restore Clinician Settings” command is used to restore the INS therapy settings that were in effect when the last 8840 session ended normally. Therapy is turned off by this command before the settings are restored, therefore the Host should send a “Therapy On” command after this command executes if therapy is needed. The parameter specifies which single group should be restored when it is 1 – 4; a value of zero indicates “all” groups should be restored.

4.8.9 RESET CYCLE COMMAND

The “Reset Cycle” command can be used to initiate a short period of stimulation. The 8840 Clinician Programmer is used to setup a group with Cycling On for a short time and off for a longer time (e.g. 0.1 second on and 58 seconds off). When this group is active and therapy is on, the “Reset Cycle” command will cause the cycle to re-start and stimulation will be delivered for the cycle on time specified. By adjusting the Cycle-On Time, Pulse Width, and Rate, the stimulation delivered when the “Reset Cycle” command is sent can be a single pulse or several pulses.

NOTE: if the cycle-off time is less than 59 seconds, then the stimulation is delivered as soon as the command is received by the Activa device; if the cycle-off time is more than 59 seconds then the stimulation circuits are turned off after stimulation is delivered and when the reset cycle command is sent, a delay of up to 1 second is encountered while the stimulation circuits are turned on.

4.8.10 STIM WINDOW COMMAND

The “Stim Window” command is new in the Nexus-D2 System.

The “Stim Window” command can be used to initiate a defined period of stimulation. The 8840 Clinician Programmer is used to setup a group with an appropriate pulse rate and amplitude program(s) that allow the patient programmer to change the stim value from 0 to the maximum desired value. When this group is active, therapy is on, and the stim amplitude is set to “0”, the “Stim Window” command will increment the stim value the specified number of steps, delay the specified number of intervals (i.e. allowing the therapy to be delivered) and then decrement the stim value the specified number of steps (i.e. back to the initial value (usually 0)). Depending on the group rate, the stimulation delivered when the “Stim Window” command is sent may be zero pulses, a single pulse or several pulses. For example, if the pulse rate is low (e.g. less than 60Hz), the increment, delay, and decrement may all execute between pulses and therefore no therapy is delivered. If the pulse rate is high (e.g. 300Hz), then 9 or 10 pulses may be delivered in the time it takes to execute the decrement command with 0 delays. With 20 delays added to the command, up to 42 pulses may be delivered. See Table 14 – Stim Window Sample for some examples of the number of pulses delivered at several pulse rates when different delay values are used.

NOTE: In a host session, the battery icon displays the last reading as a percentage (100%, 75%, 50%, or 25%); however a battery measurement will not be taken until INS communication has not occurred for at least 30 seconds. When the last battery measurement is 10 minutes old, the battery icon is removed from the display and this display change takes about 100ms. To avoid this 100 ms delay at the 10 minute mark, the battery icon will be removed from the screen when the first “Stim Window” command is received. The removal of the battery icon will cause the second “Stim Window” command to be delayed by 100 ms. An initial command with zero steps can be used to “prime” the system and prevent the delay from occurring for the rest of the session. If no commands are issued for 30 seconds, a battery measurement will be taken and the battery icon will be re-displayed on the screen.

Table 14 – Stim Window Sample

This table shows the number of pulses in each burst for the specified rate (PW is 90us)											
Stimulation Rate	0 Delay	1 Delay	3 Delay	5 Delay	8 Delay	10 Delay	12 Delay	14 Delay	16 Delay	18 Delay	20 Delay
60Hz	1 or 2	2	2 or 3	3 or 4	3 or 4	5 or 6	5 or 6	6 or 7	7	7 or 8	8 or 9
80Hz	2 or 3	2 or 3	3 or 4	4 or 5	6	6 or 7	7 or 8	8 or 9	9 or 10	10 or 11	11 or 12
100Hz	3 or 4	3 or 4	4 or 5	5 or 6	7 or 8	8 or 9	9 or 10	10 or 11	11 or 12	12 or 13	13 or 14
120Hz	3 or 4	4 or 5	5 or 6	7	9	10	11 or 12	13	14 or 15	15 or 16	16 or 17

150Hz	4 or 5	5 or 6	7	8 or 9	11 or 12	12 or 13	14 or 15	16	17 or 18	19	21
180Hz	5 or 6	6 or 7	8 or 9	11	13 or 14	15 or 16	17	19 or 20	21	23 or 24	25 or 26
210Hz	6 or 7	7 or 8	10	12 or 13	15 or 16	18 or 19	20	22 or 23	24 or 25	27	29 or 30
240Hz	7 or 8	9	11 or 12	14	18	20	23	26	27-29	31	33
270Hz	8 or 9	9 or 10	13	15 or 16	20	23	26	29	31 or 32	34 or 35	38
300Hz	9 or 10	11	14 or 15	17 or 18	22 or 23	26	29 or 30	33	36	38 or 39	42

NOTE: The delay interval is not specific and the increase in stim time per delay is dependent on the stim rate. Each delay will increase the transaction time due to the added telemetry required; however 10ms per delay can be used for rough calculations. See Table 15 – Stim Window Time Sample for examples of the time it takes to deliver a command from the host to the INS and see a pulse on the lead, the time until the stim stops and the total command transaction time.

Table 15 – Stim Window Time Samples

This table shows the approximate transaction time in milliseconds from Host Send to: Stim Pulse Start / Stim Pulse End / Ack to Host											
Stimulation Rate	0 Delay	1 Delay	3 Delay	5 Delay	8 Delay	10 Delay	12 Delay	14 Delay	16 Delay	18 Delay	20 Delay
60Hz	90/ 107/ 138	90/ 117/ 147	110/ 132/ 169	110/ 149/ 189	115/ 178/ 215	120/ 202/ 239	133/ 215/ 255	136/ 231/ 276	149/ 249/ 300	152/ 269/ 318	167/ 289/ 337
80Hz	90/ 115/ 141	91/ 117/ 152	104/ 128/ 167	107/ 154/ 186	117/ 178/ 218	124/ 197/ 236	126/ 212/ 258	136/ 236/ 276	144/ 257/ 297	154/ 280/ 316	157/ 295/ 334
100Hz	90/ 110/ 138	92/ 117/ 152	101/ 141/ 167	108/ 155/ 186	115/ 183/ 220	122/ 202/ 236	123/ 223/ 260	140/ 237/ 268	149/ 257/ 297	149/ 268/ 313	157/ 289/ 342
120Hz	85/ 112/ 138	91/ 120/ 152	99/ 141/ 169	109/ 157/ 186	117/ 183/ 220	122/ 204/ 239	133/ 217/ 260	138/ 236/ 280	146/ 260/ 296	154/ 273/ 316	154/ 292/ 335
150Hz	88/ 112/ 144	89/ 120/ 152	93/ 140/ 168	103/ 156/ 191	116/ 181/ 217	123/ 205/ 237	130/ 220/ 258	142/ 241/ 277	145/ 255/ 295	154/ 274/ 316	160/ 294/ 334
180Hz	82/ 111/ 138	91/ 119/ 152	95/ 140/ 169	103/ 158/ 189	115/ 184/ 219	123/ 207/ 240	130/ 224/ 258	138/ 240/ 276	146/ 258/ 296	154/ 279/ 318	159/ 294/ 333
210Hz	85/ 110/ 141	90/ 120/ 151	97/ 139/ 171	101/ 158/ 188	113/ 184/ 221	123/ 204/ 241	131/ 222/ 259	138/ 238/ 280	146/ 261/ 299	151/ 278/ 319	158/ 293/ 337
240Hz	84/ 114/ 140	88/ 119/ 150	95/ 139/ 167	102/ 155/ 189	115/ 185/ 220	121/ 204/ 241	129/ 225/ 259	136/ 241/ 278	143/ 258/ 296	153/ 279/ 316	158/ 297/ 337
270Hz	84/ 112/ 140	87/ 121/ 151	98/ 139/ 172	102/ 156/ 190	115/ 185/ 222	121/ 203/ 239	131/ 224/ 258	135/ 239/ 278	144/ 258/ 300	152/ 278/ 318	157/ 295/ 337
300Hz	82/ 100/ 141	87/ 212/ 150	97/ 142/ 171	102/ 160/ 191	114/ 184/ 220	120/ 205/ 240	130/ 224/ 259	136/ 240/ 279	146/ 259/ 299	150/ 275/ 316	157/ 296/ 336

4.8.11 Get Sense Status Command

The "Get Sense Status" command is new in the Nexus-D2 System.

The “Get Sense Status” command will return the current sensing status as follows:

- Payload Data Type – two bytes containing 0x8007
- Response Code – one byte value defined in Table 10 - Response Codes
    - If the command response code is non-zero, the data should not be considered useful.
- Sense State – one byte where 0 indicates not sensing (i.e. Program Mode), 1 indicates sensing is enabled, and 2 indicates Montage Run Mode.
    - NOTE: Detection data is not available in Montage Run Mode or Program Mode.
- Detection State – 0 indicates detection is disabled and 1 indicates detection is enabled.
- LD State – this is the state of the input signal for the Duration Constraint block; If the Linear Discriminant (LD) calculated Boundary Distance is greater than 0, the LD State is 1, otherwise the LD State is 0.
- In Detect – this is the output of the Duration Constraint block; 0 is not in detect and 1 is in detect.

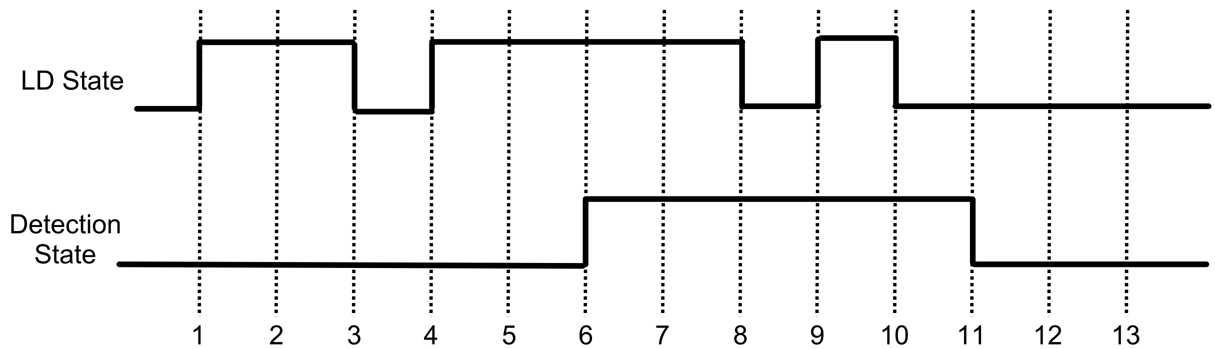

Figure 4-5 - Example of Duration Constraint logic. D_{onset} = 3, D_{term} = 2.

Figure 4-5 shows an example of Duration Constraint logic. In this case, the Onset Duration is 3 and the Termination Duration is 2. The labels across the bottom are Detection Algorithm updates. Here is an update-by-update explanation of the logic:

1) The LD State goes high. The Onset Count increments to 1. This is less than 3, so the Detection State remains low.
2) The LD State is still high, so the Onset Count increments to 2. This is less than 3, so the Detection State remains low.
3) The LD State goes low, so the Onset Count resets to 0. The Detection State remains low.
4) The LD State goes high. The Onset Count increments to 1. This is less than 3, so the Detection State remains low.
5) The LD State is still high, so the Onset Count increments to 2. This is less than 3, so the Detection State remains low.

6) The LD State is still high, so the Onset Count increments to 3. This meets the Onset Duration constraint of 3, so the Detection State goes high. The Termination Count is reset to 0.
7) The LD state is still high, so the Termination Count stays at 0 and the Detection State remains high.
8) The LD State goes low, so the Termination Count is incremented to 1. This is less than 2, so the Detection State remains high.
9) The LD State goes high, so the Termination Count is reset to 0. The Detection State remains high.
10) The LD State goes low, so the Termination Count is incremented to 1. This is less than 2, so the Detection State remains high.
11) The LD State stays low, so the Termination Count is incremented to 2. This meets the Termination Duration constraint of 2, so the Detection State goes low.

- In Cluster Detect – The Activa PC+S has some clustering logic that takes the Detection State as an input and generates an output signal known as the Cluster State. When the system is in Run Mode with the Detection Algorithm enabled and the Detection State is high, Cluster Detect will contain a 1 and when the cluster timer expires, Cluster Detect will contain a 0.

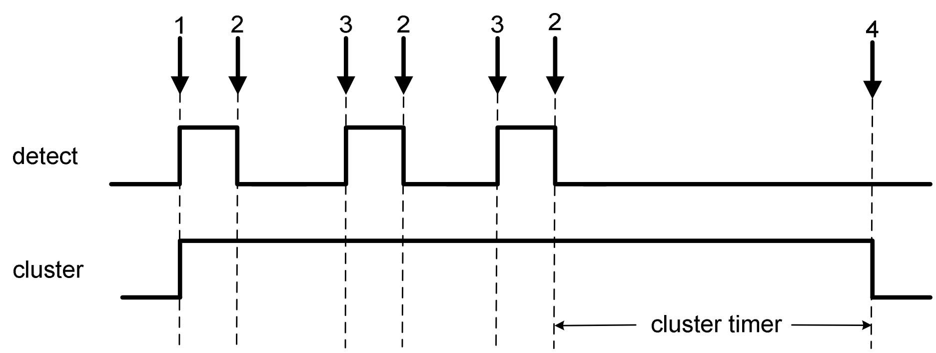

Figure 4-6 - Example of Cluster Logic

Figure 4-6 shows an example of cluster logic. The numbers at the top of the figure indicate the following:

1. Detection onset, start of cluster: reset cluster timer
2. End of detection: start cluster timer
3. New detection before cluster timer expired: stop timer, reset timer
4. Cluster timer expires, end of cluster

4.9 Communication Protocol Use Scenarios

Because bandwidth is limited, when a maintenance session is active, the Host application developer should adopt a strategy when modifying therapy while retrieving real-time data. This section describes the data flow to help define a strategy.

4.9.1 Real-Time Data Priority

If receiving all real time sensing data is more important than modifying the therapy, then the Host must interleave one therapy command and one Get Real Time Data command. See Figure 4-7 - Multiple Commands during Real Time (Data Priority) for an example of this scenario. With this method, the Host won't miss data packets but will have more delays and limited windows to send therapy modification commands.

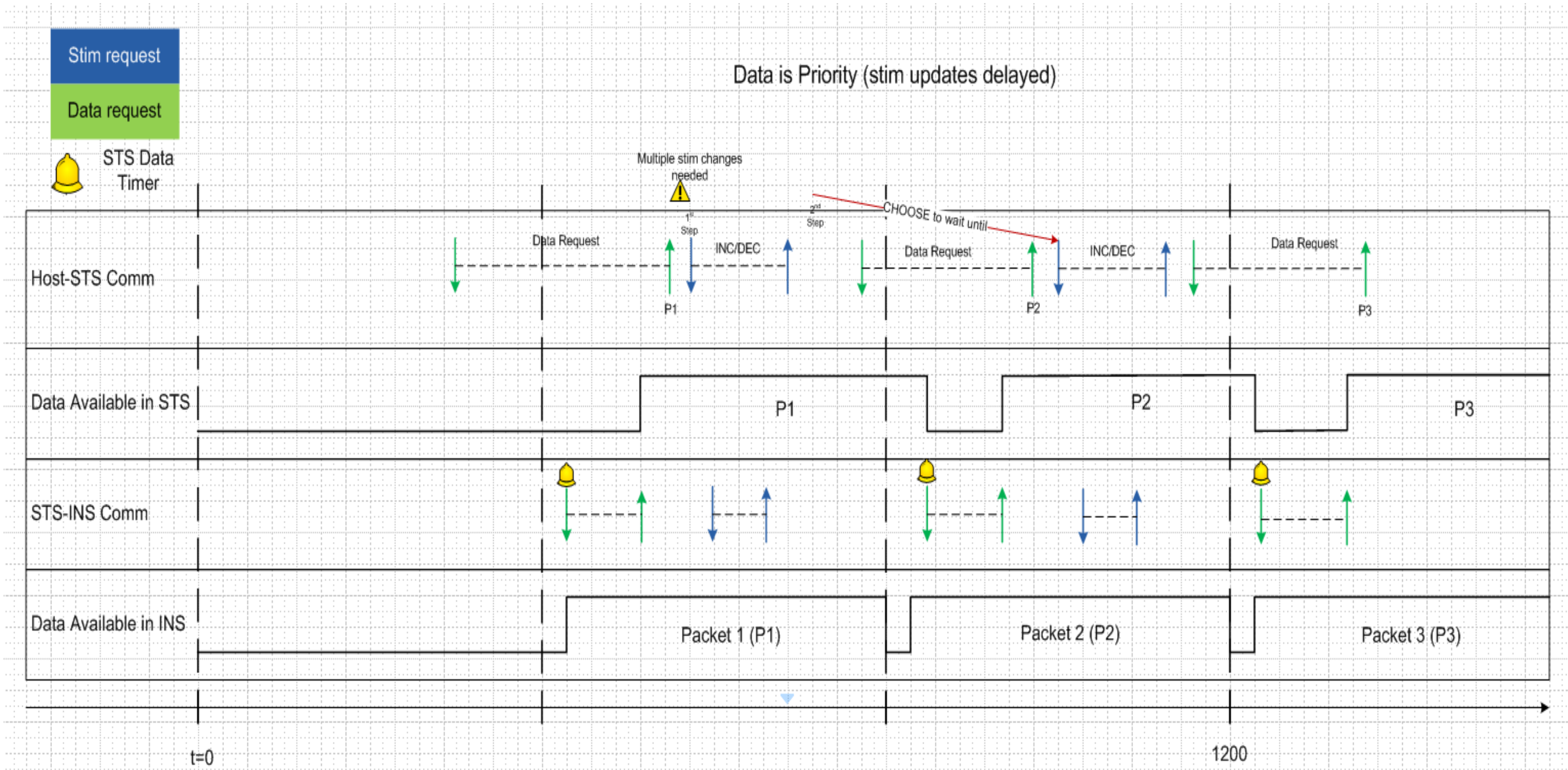

Figure 4-7 - Multiple Commands during Real Time (Data Priority)

4.9.2 THERAPY MODIFICATION PRIORITY

If modifying therapy is more important than receiving all of the real time data, then the Host may send a sequence of commands before requesting the next real time data packet; in this case some data packets will be lost. See Figure 4-8 - Multiple Commands during Real Time (Therapy Priority) for an example of this scenario.

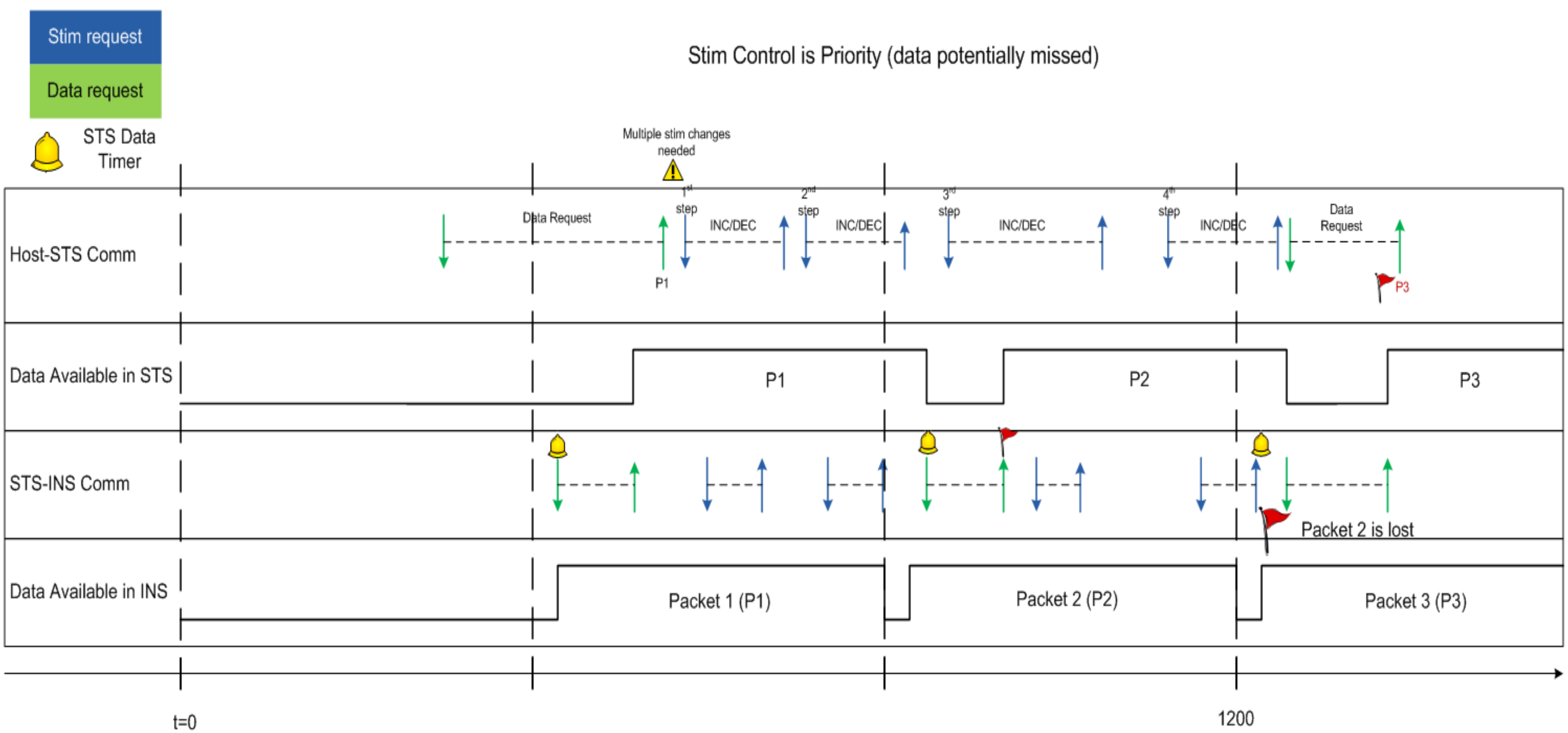

Figure 4-8 - Multiple Commands during Real Time (Therapy Priority)

4.9.3 DATA FLOW TIMING

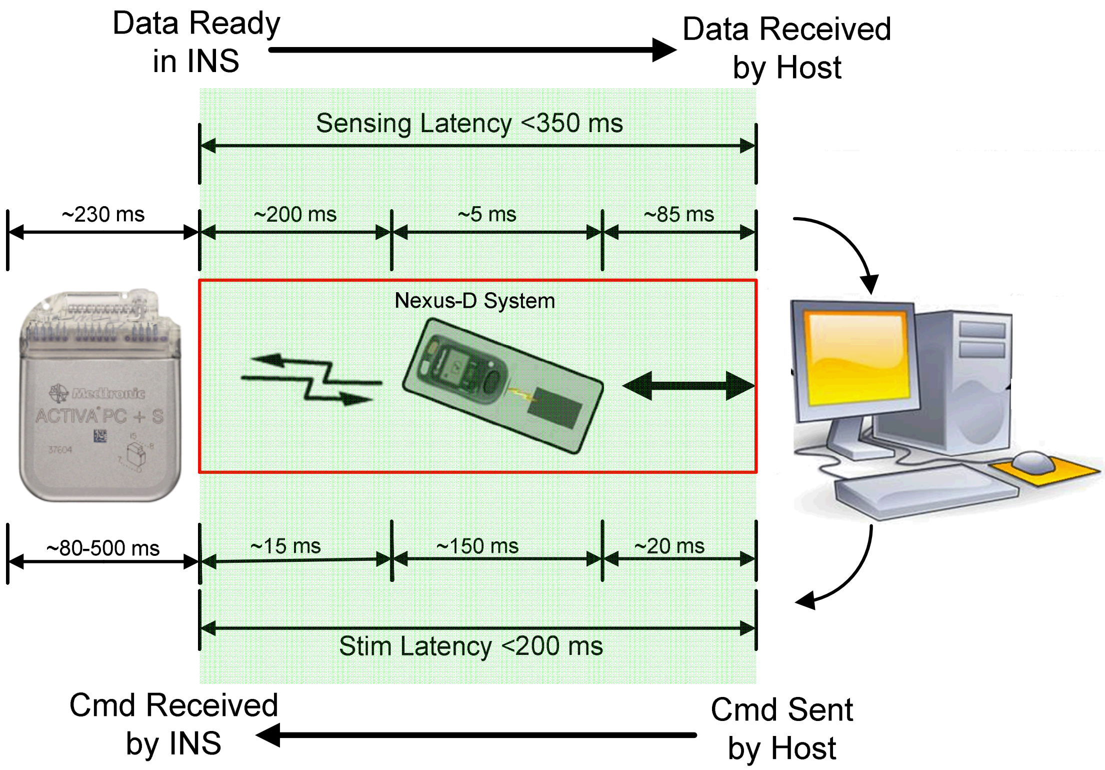

Figure 4-9 - Latency introduced by the Nexus-D System

The latency introduced in the uplink of real time data is broken down as follows: up to 200ms for the telemetry to transfer the data from the INS to the Nexus-D System, about 5ms to decompress the data, and about 85ms to uplink the data to the Host via IrDA (See Figure 4-11 - Typical Real Time Data Latency Introduced by Nexus-D System).

The latency introduced in the downlink of a therapy command depends on the relative timing of when the command is sent by the Host. If the command is received by the Nexus-D System while a real time telemetry command is in progress, then the Nexus-D System must wait for that command to finish which could take up to 150ms. If the Host sends the command within the wait period for the next real time telemetry command (i.e. less than 100ms after the Host received the last uplink), then the latency is minimal (See Figure 4-12 – Multiple INC/DEC during real-time).

Note: The Host may receive a therapy command response before the therapy is actually delivered to the patient. The INS replies that the command has been accepted and started but may take from 80ms (for INC/DEC commands) to 500ms (for group switch command) to deliver the change to the patient. Figure 4-10 – "Therapy On" Example shows one example of this situation.

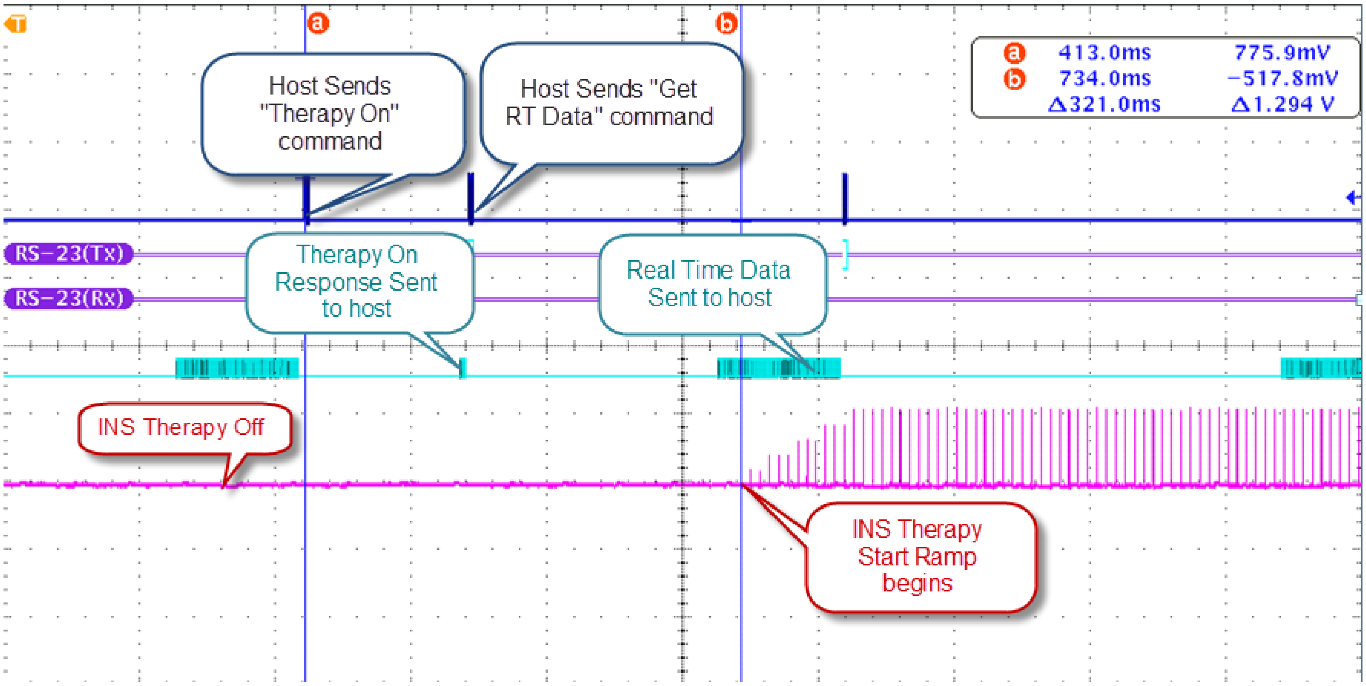

Figure 4-10 – “Therapy On” Example

In the example shown in Figure 4-10 – “Therapy On” Example, therapy is off and the Host sends the “Therapy On” command; the command is acknowledged by the Nexus-D System and therefore the Host sends the next get real time data command. Notice that it takes 321ms for therapy to begin but the Host received the “therapy on” success response well before that. The full therapy point is not reached until the soft start ramp is finished. The soft start ramp time is dependent on how the 8840 configured the INS. Therefore, the next real time data block received by the Host does not contain data that was collected while therapy was on.

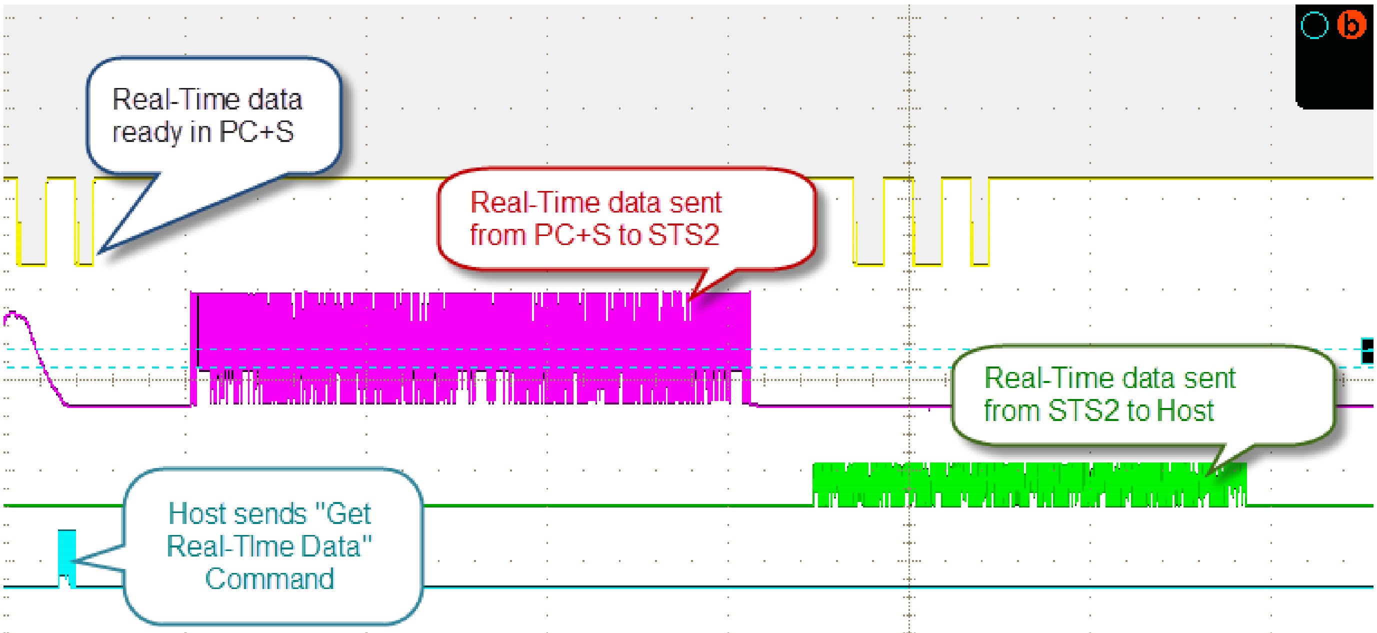

Figure 4-11 - Typical Real Time Data Latency Introduced by Nexus-D System

The example shown in Figure 4-11 - Typical Real Time Data Latency Introduced by Nexus-D System shows the upload of real time data from an Activa PC+S that is configured with sensing set to 422hz Channel 1 TD, Channel 2, 3 and 4 Power. This example represents the largest data packet size. The latency introduced by the Nexus-D System in this example is 256ms, however, since the Nexus-D System does not know where the Activa PC+S is in the data collection cycle there may be times when the Nexus-D System does not request the data immediately when it is ready in the Activa PC+S. An algorithm is used to try to hone in on the optimum time to request the real time data but there may be times after a Host command that the algorithm is out of sync, and latency introduced by the Nexus-D System will be increased.

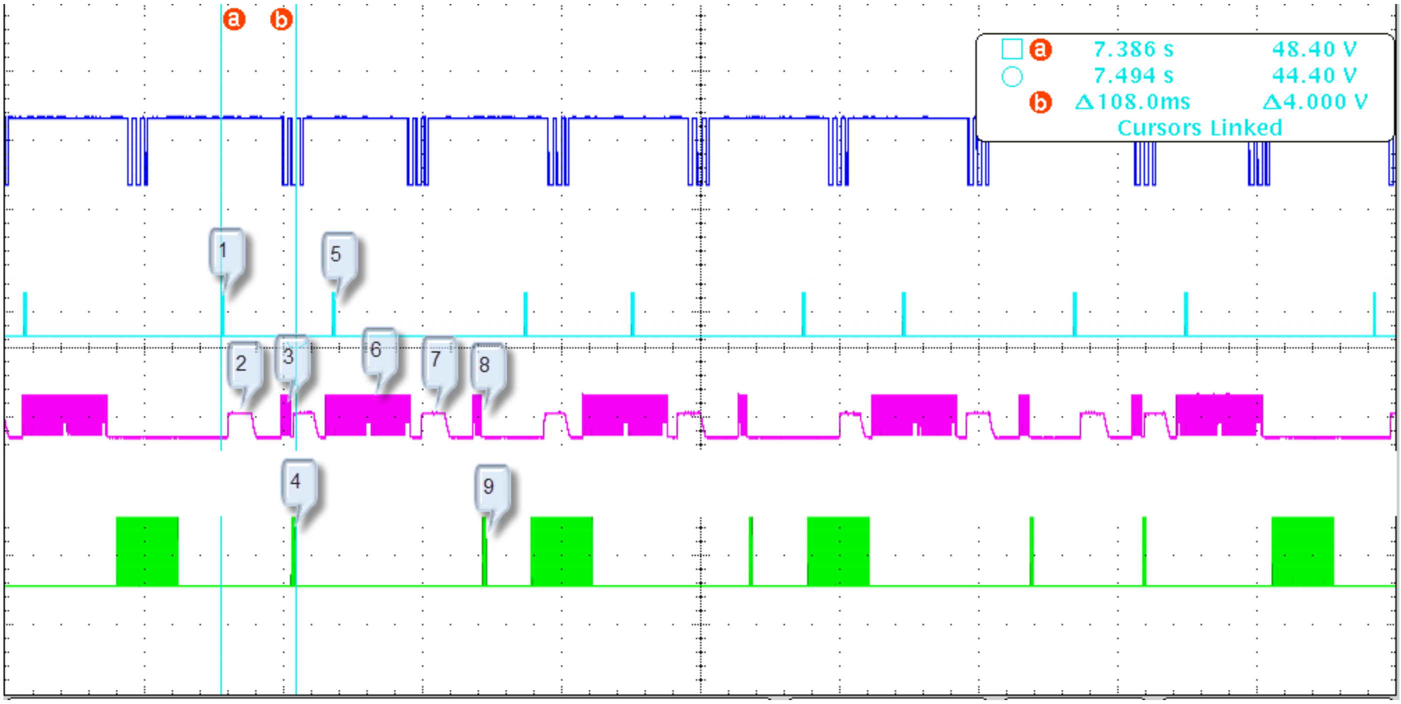

Figure 4-12 – Multiple INC/DEC during real-time

In the example shown in Figure 4-12 – Multiple INC/DEC during real-time, (1) an INC Amplitude command is sent from the Host to the Nexus-D System with a single step and no repeats. (2) The command is sent from the Nexus-D System to the Activa PC+S. (3) the response is sent from the Activa PC+S to the Nexus-D System. (4) The response is sent from the Nexus-D System to the Host. Notice that the time from the Host sending the command (1) to the Host receiving the response (4) is 108ms. Then the Host sends a second INC amplitude command (5), but the Nexus-D System is busy receiving the next real-time packets from the Activa PC+S (6). When the real-time packet has been received by the Nexus-D System, it sends the INC amplitude command to the Activa PC+S (7). The INC amplitude response is received by the Nexus-D System (8). The INC amplitude response is sent to the Host (9). For the second INC, the time from the Host sending the command (5) to the Host receiving the response (9) is more than 200ms.

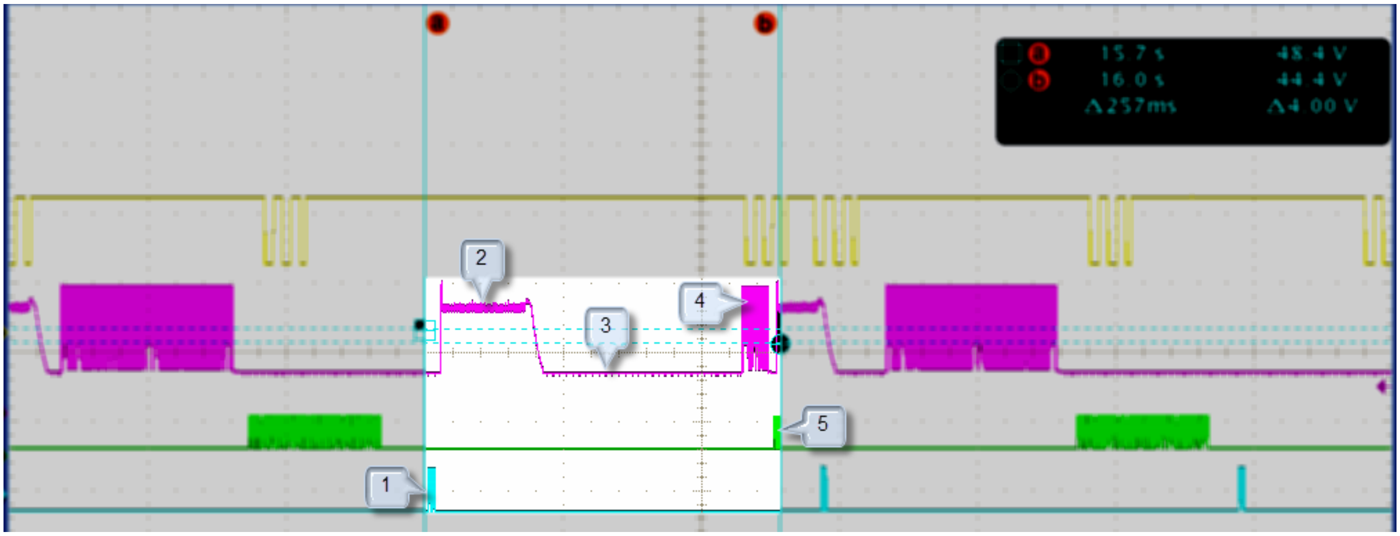

Figure 4-13 - Batched INC Amplitude during Real Time

In the example shown in Figure 4-13 - Batched INC Amplitude during Real Time, (1) a command was sent from the Host to increment the amplitude 5 steps 4 times. (2) A batch of 4 commands to increment amplitude 5 steps is sent to the Activa PC+S. (3) The Activa PC+S executes each of the 4 commands. (4) The new Amplitude value is sent from the Activa PC+S to the Nexus-D System. (4) The new value is sent from the Nexus-D System to the Host. Notice that the downlink time and wait for telemetry response is greater than 250ms; while busy executing the command the INS delayed getting the real-time data and the Nexus-D System was delayed in requesting the real time data because it was waiting for the INC command response.

5 REVISION HISTORY

Change History	
Version	Change Description
8.0	PNITS00060936 – clarified that Nexus-D2 incorporates all Nexus-D System functionality in section 1. Removed PC+S column from table 11. Removed comment about uncompressed data from section 4.8.3 and 4.8.7. PNITS00061044 – note that the Activa RC does not support the Reset Stim Cycle command
7.1 - 7.4	PNITS00060896 – add note to section 5.8.10 PNITS00060736 - Add a new command to open a "Stim Window". Add a new command to "Get Sense Status". Correct response payload type for "Start Sensing" and "Stop Sensing" commands.
7.0 (6.1)	PNITS00060157 – Updated timeout values for GetStatus and Get INS Info commands. Changed “Nexus-2” to “Nexus-D” throughout.
6.0(5.4)	PNITS00059857 – Added wording throughout to distinguish where use of the PC device is allowed.
5.3	PNITS00059857– Added new response code 0x73 to table 9. Added new command “Reset Cycle” to table 10. Updated Table 10 to include Activa PC/PC+S supported columns. Added “Reset Cycle” command explanation section 5.8.9
5.2	PNITS00059693 – Added response code 109 to Table 10 - Response Codes and updated Section 5.8.3
5.1	PNITS00059668 – Added response code 21 to Table 10 - Response Codes
5.0(4.2)	Route for approval
4.1	PNITS00059633 – add battery level “unknown” description to section 5.8.4
4.0(3.5)	Route for Approval
3.2 - 3.5	PNITS00059528 – changes listed in review document NRP1088-36275
3.1	PNITS00059528 - Nexus-D System Interface Definition Document Review
3.0(2.3)	Routed for approval
2.2	Corrected the referenced sections in the 2.1 Change History in this table. Incorrect referenced 5.81 and 5.85.
2.1	PNITS00059467 – add constant current amplitude description to sections 5.8.1 and 5.8.5.
2.0	Initial version.

6 END OF DOCUMENT

This section concludes this document.