# Command structure for ineedMD Bluetooth dongle.

Version: 0.8

Date: 20140715

### Changes

0.5

Added clarification on nomenclature

Corrected ACK and NAK packet length

Reordered the datagrams

Added examples for each datagram

0.6

Revised datagram types

0.7

Added clarification on command datagram’s length

0.8

Added examples for Stored Data Information Block packet

Added command for “Send Test Signal”, and added it as a datagram type

0.9

Added example for response to “Erase Data Set” command

Defined end of file action for transferring data sets

Connection to the ineedMD dongle will be using SPP profile.

Commands will be inband

Additional communication – such as audio – will be using a compound profile.

No custom profiles will be developed.

# Control Commands

The application passes commands to the dongle which performs the commanded task and returns records.

The dongle can return records to alert the application of status changes without a request.

Commands are variable length

Start and end markers are included to improve machine and human readability.

\*\*Please note that this is not XML protocol, rather the <EXAMPLE>: tags are used to enhance readability.

### Status Request

Command - 0x11

Data – Null

|  |  |
| --- | --- |
| Program | Dongle |
| Status command |  |
|  | Status record |
|  |  |

### Status Set

Program -> dongle - “Status record”

Dongle-> program – ACK/NAK

Some cannot be changed if a capture is underway

Command - 0x12

Data – Status Datagram

|  |  |
| --- | --- |
| Program | Dongle |
| Status record command with status datagram |  |
|  | ACK/NAK |

### Stored Data Sets Information

Program -> dongle - “Transfer”

Dongle-> program – stored data information block

Command - 0x13

Data – Null

|  |  |
| --- | --- |
| Program | Dongle |
| Transfer info of stored data command |  |
|  | Stored data information block |

### Data Set Transfer

Program -> dongle - “Transfer data set #”

Dongle-> program – ACK

Dongle -> program – Data Set

Command - 0x14

Data – Data Set ID:2B

|  |  |
| --- | --- |
| Program | Dongle |
| Transfer data set command (w/ data set ID) |  |
|  | NAK or Respective data set |

### Erase Data Set

Program -> dongle - “Erase Data Set #”

Dongle-> program – ACK

Command - 0x15

Data - data set ID:2B

|  |  |
| --- | --- |
| Program | Dongle |
| Erase data set command  (w/ data set ID) |  |
|  | ACK/NAK |
| After erase complete | |
|  | Sequence of data information block |

After the erase is complete

Dongle-> program – sequence of data information block

#### Example:

After data set 2 is erased, this sequence of data information block is sent back to the program.

Before erase – data sets 1, 2, 3 are in the stored data information packet:

|  |  |
| --- | --- |
| DESCRIPTION | BYTE SEQUENCE |
| <START> | 9C |
| <STORED DATA INFO PACKET> | 05 |
| <LENGTH> | 20 |
| <# OF STORED DATA SETS> | 03 |
| <SEQUENCE TAG> | 00 |
| <SEQUENCE TAG> | 01 |
| <START TIME> | D7 |
| <START TIME> | 68 |
| <START TIME> | 56 |
| <START TIME> | 00 |
| <LENGTH> | 00 |
| <LENGTH> | 00 |
| <LENGTH> | 14 |
| <SEQUENCE TAG> | 00 |
| <SEQUENCE TAG> | 02 |
| <START TIME> | D7 |
| <START TIME> | 68 |
| <START TIME> | AA |
| <START TIME> | 00 |
| <LENGTH> | 00 |
| <LENGTH> | 00 |
| <LENGTH> | 60 |
| <SEQUENCE TAG> | 00 |
| <SEQUENCE TAG> | 03 |
| <START TIME> | D7 |
| <START TIME> | 68 |
| <START TIME> | 55 |
| <START TIME> | 11 |
| <LENGTH> | 00 |
| <LENGTH> | 12 |
| <LENGTH> | 00 |
| <STOP> | C9 |

After erase (data set 2 erased, data set 3 becomes data set 2 with sequence tag 02):

|  |  |
| --- | --- |
| DESCRIPTION | BYTE SEQUENCE |
| <START> | 9C |
| <STORED DATA INFO PACKET> | 05 |
| <LENGTH> | 17 |
| <# OF STORED DATA SETS> | 03 |
| <SEQUENCE TAG> | 00 |
| <SEQUENCE TAG> | 01 |
| <START TIME> | D7 |
| <START TIME> | 68 |
| <START TIME> | 56 |
| <START TIME> | 00 |
| <LENGTH> | 00 |
| <LENGTH> | 00 |
| <LENGTH> | 14 |
| <SEQUENCE TAG> | 00 |
| <SEQUENCE TAG> | 02 |
| <START TIME> | D7 |
| <START TIME> | 68 |
| <START TIME> | 55 |
| <START TIME> | 11 |
| <LENGTH> | 00 |
| <LENGTH> | 12 |
| <LENGTH> | 00 |
| <STOP> | C9 |

### Capture Data Set

Program -> dongle - “Capture Data Set for Duration #”

Dongle-> program – ACK

Command - 0x16

Data – capture duration in seconds:3B

Dongle captures Data to local storage for duration #

If a subsequent capture command is received before the end of duration # a new capture file is started.

If a “capture duration 0” is received the present capture is terminated and no new capture is started.

If the dongle is sending current measurements it will store the current measurements and continue broadcasting them.

|  |  |
| --- | --- |
| Program | Dongle |
| Capture data set for duration command (w/ data set ID) |  |
|  | ACK/NAK |

### Display Real Time Data Measurements

Program -> dongle - “Current Measurements TRUE/FALSE”

Dongle-> program – ACK

Command - 0x17

Data – TRUE (0xhFF)/FALSE(0xh00)

If Current Measurements TRUE

\*forces display of real time data on app

Dongle-> program – measurement block

If Current Measurements FALSE

Dongle ceases sending the measurement blocks or ignores the command.

|  |  |
| --- | --- |
| Program | Dongle |
| Current measurements TRUE/FALSE |  |
|  | NAK or Measurement block (TRUE) |

### Send Test Signal

Program ->dongle - “request to send test signal”

Dongle -> program – sends sequence of real time data

Command – 0x18

Data—1B: corresponding test pattern

0x00 - stop sending test data/ 0x01 - send test data “pattern 1”

|  |  |
| --- | --- |
| Program | Dongle |
| Request to send test signal |  |
|  | Test data “pattern 1” |

# Datagrams

## General form

General form of the datagrams is

<Start Marker> - 0xh9C

<datagram\_type> :1B

<length>:1B length of datagram

<Data>: 0B to 251B

<End Marker> - 0xhC9

### Datagram Types

|  |  |
| --- | --- |
| Type | Hex Value Corresponding to Type |
| Command/sets | 0xh01 |
| Status/information (sent from dongle to app) | 0xh02 |
| Data | 0xh03 |
| Real time data | 0xh04 |
| Stored data information | 0xh05 |

### ACK

0xh9C

0xh00

0xh04

0xhC9

\* Additionally sent to program as an indication of reaching the end of a file after a data set is transferred

### Status

Status is # bytes long

0xh9C

0xh02

0xh – Length

<firmware version>:1B; R

<battery voltage>:1B (mV above 2.5 V); R

<operating mode>:1B RW

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | Power state | Power state | Capture state | Stream state | Error | USB connected | Reserved |
| Writing a “1” causes the device to do a warm reset | 00: hibernation  01: sleep  10: active  11: high power | | Capturing EKG to internal flash memory | Streaming EKG data through Bluetooth | 1 indicates an error | 1 indicates connection to USB |  |

<capture setting>:2B; RW (per ADS1198 capture setting)

<time>:4B:RW - seconds since 1900 ( most significant four of NTP )

< Alarm Block>:4B

Most significant:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Pacer Suspected | RA Lead off | LA Lead off | RL Lead off | LL Lead off | V1 Lead off | V2 Lead off | V3 Lead off |
| 1 indicates evidence of a pacemaker spike | 1 indicates lead off/loose | 1 indicates lead off/loose | 1 indicates lead off/loose | 1 indicates lead off/loose | 1 indicates lead off/loose | 1 indicates lead off/loose | 1 indicates lead off/loose |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| V4 Lead off | V5 Lead off | V6 Lead off | Reserved | Reserved | Reserved | Reserved | Reserved |
| 1 indicates lead off/loose | 1 indicates lead off/loose | 1 indicates lead off/loose |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved |
|  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved |
|  |  |  |  |  |  |  |  |

0xhC9

#### Example

Status data packet for retrieving the Dongle’s current status:

Firmware version 1

Battery voltage = 3.5 V

Operating mode = 0xh50 (active, capturing to flash)

Capture setting = 0xh0BB8

Time = 0xhD7685600 (seconds since 1900)

Alarm block = 0xhC0000000 (pacer suspected, RA lead off)

The byte stream will be - 9C:02:11:01:64:50:0B:B8:D7:68:56:00:C0:00:00:00:C9

|  |  |
| --- | --- |
| DESCRIPTION | BYTE SEQUENCE |
| <START> | 9C |
| <STATUS PACKET> | 02 |
| <LENGTH> | 11 |
| <FIRMWARE VERSION> | 01 |
| <BATTERY VOLTAGE> | 64 |
| <OPERATING MODE> | 50 |
| <CAPTURE SETTING> | 0B |
| <CAPTURE SETTING > | B8 |
| <TIME> | D7 |
| <TIME> | 68 |
| <TIME> | 56 |
| <TIME> | 00 |
| <ALARM BLOCK> | C0 |
| <ALARM BLOCK> | 00 |
| <ALARM BLOCK> | 00 |
| <ALARM BLOCK> | 00 |
| <STOP> | C9 |

### Command

0xh9C

0xh01

0xh <LENGTH> - Length

<COMMAND>:1B

<Command DATA>: #B

0xhC9

\*\*Note: the number of bytes corresponding with the command data is variable depending on the command being sent; thus, the total length of a command packet varies.

#### Example

Command data packet for capturing 40137s (0x9CC9 seconds) of EKG data:

The byte stream will be - 9C:01:07:16:9C:C9:C9

|  |  |
| --- | --- |
| DESCRIPTION | BYTE SEQUENCE |
| <START> | 9C |
| <COMMAND PACKET> | 01 |
| <LENGTH> | 07 |
| <CAPTURE COMMAND> | 16 |
| <BYTE 1 OF DATA> | 9C |
| <BYTE 2 OF DATA> | C9 |
| <STOP> | C9 |

### Measurement Data

0xh9C

0xh03 – EKG Measurement

0xh0E - length

<RA Measurement>:2B

<LA Measurement>:2B

<RL Measurement>:2B

<LL Measurement>:2B

<V1 Measurement>:2B

<V2 Measurement>:2B

<V3 Measurement>:2B

<V4 Measurement>:2B

<V5 Measurement>:2B

<V6 Measurement>:2B

0xhC9

#### Example

Measurement data packet for EKG Measurement Data:

The byte stream will be - 9C:03:0E:16:03:10:0A:9C:C9:C9:0E:2C:9C:C9

|  |  |
| --- | --- |
| DESCRIPTION | BYTE SEQUENCE |
| <START> | 9C |
| <EKG MEASUREMENT PACKET> | 03 |
| <LENGTH> | 0E |
| < RA MEASUREMENT > | 16 |
| < LA MEASUREMENT > | 03 |
| < RL MEASUREMENT > | 10 |
| < LL MEASUREMENT > | 0A |
| < V1 MEASUREMENT > | 9C |
| < V2 MEASUREMENT > | C9 |
| <V3 MEASUREMENT > | C9 |
| <V4 MEASUREMENT > | 0E |
| <V5 MEASUREMENT > | 2C |
| <V6 MEASUREMENT > | 9C |
| <STOP> | C9 |

### Stored Data Information Block

0xh9C

0xh05 – Stored data information

0xh<length> - length

<number of stored data sets>:2B 0000 to FFFF records

{

<Sequence Tag> : 2B- The sequence is continuous 1 to last record. 0 is not valid here.

<start time of recording>:4B – system time

<length>:3B – recording time (seconds)

}

0xhC9

If the number of stored data sets exceeds the 255 byte packet limit additional stored data sets will be sent.

On the dongle data sets are tagged with the start recording time and the MAC address of the tablet that initiated the recording. Linking of data set to patient information is performed on the tablet that initiated the capture.

#### Example 1

Stored data information packet containing TWO data sets: one recorded at time 0xD7685600 seconds and of length 20 seconds; one recorded at time 0xD768AA00 seconds and of length 96 seconds.

The byte stream will be: 9C:05:18:02:00:01:D7:68:56:00:00:00:14:00:02:D7:68:AA:00:00:00:60:C9

|  |  |
| --- | --- |
| DESCRIPTION | BYTE SEQUENCE |
| <START> | 9C |
| <STORED DATA INFO PACKET> | 05 |
| <LENGTH> | 17 |
| <# OF STORED DATA SETS> | 02 |
| <SEQUENCE TAG> | 00 |
| <SEQUENCE TAG> | 01 |
| <START TIME> | D7 |
| <START TIME> | 68 |
| <START TIME> | 56 |
| <START TIME> | 00 |
| <LENGTH> | 00 |
| <LENGTH> | 00 |
| <LENGTH> | 14 |
| <SEQUENCE TAG> | 00 |
| <SEQUENCE TAG> | 02 |
| <START TIME> | D7 |
| <START TIME> | 68 |
| <START TIME> | AA |
| <START TIME> | 00 |
| <LENGTH> | 00 |
| <LENGTH> | 00 |
| <LENGTH> | 60 |
| <STOP> | C9 |

#### Example 2

Stored data information packet for ZERO data sets stored in memory.

The byte stream will be 9C:05:05:00:C9

|  |  |
| --- | --- |
| DESCRIPTION | BYTE SEQUENCE |
| <START> | 9C |
| <STORED DATA INFO PACKET> | 05 |
| <LENGTH> | 05 |
| <# OF STORED DATA SETS> | 00 |
| <STOP> | C9 |

### NAK

0xh9C

0xhFF

0xh04

0xhC9

## Error Recovery

It should be assumed that the link is not classically lossy but is a temporary connection. The code should be designed to accept missing blocks of data and dropped connections.