Application Programming Interface (API) for LEAP Wireless Sensor Electronics

SCPI interface module (API)

The SCPI (Standard Commands for Programmable Instruments) standard will be used for the communication in between the LEAP wireless sensor electronics (herein, called "device") and the GUI. More about this standard can be found in the following link:

http://en.wikipedia.org/wiki/Standard_Commands_for_Programmable_Instruments

Commands summary

Many of the commands are similar for the eight available channels on the device. Whenever a command will be available for all channels, CHn has to be exchanged with either CH1, CH2 …, CH8. Note that special wiring will be required to connect more than 4 sensors to the device, please contact LEAP in this circumstance.

Calibration commands

CALibration:CHn:CURRent:SELm value

CALibration:CHn:CURRent:SELm?

CALibration:CHn:CAPacitance value

CALibration:CHn:CAPacitance?

Configuration commands

CONFiguration:BANKn:EXCitation:FREQuency value

CONFiguration:BANKn:EXCitation:FREQuency?

CONFiguration:BANKn:UPDate:FREQuency value

CONFiguration:BANKn:UPDate:FREQuency?

CONFiguration:BANKn:PACKetsize value

CONFiguration:BANKn:PACKetsize?

CONFiguration:CHn:CURRent:SELect value

CONFiguration:CHn:CURRent:SELect?

CONFiguration:CHn:AVGbuf value

CONFiguration:CHn:AVGbuf?

CONFiguration:CHn:MEAsure:CAPacitance state

CONFiguration:CHn:MEAsure:CAPacitance?

CONFiguration:CHn:MEAsure:ESR state

CONFiguration:CHn:MEAsure:ESR?

CONFiguration:BLUEtooth:ID value

CONFiguration:BLUEtooth:ID?

CONFiguration:STREAMing:METHod state

CONFiguration:STREAMing:METHod?

CONFiguration:TRIGger state

CONFiguration:TRIGger?

VERBOSE

RADIO:CONFIG

Streaming commands

STREAMing state

STREAMing?

STREAMing:BANKn state

STREAMing:BANKn?

STREAMing:CHn state

STREAMing:CHn?

STREAMing:IMU state

STREAMing:IMU?

Measurement and read commands

MEASurement:CHn:CAPacitance?

MEASurement:CHn:RESistance?

MEASurement:BATTery?

READ:HardWare:REVision?

READ:SoftWare:REVision?

Communication headers from device responses to PC:

<ACK> 0x06 Acknowledge – Positive response on command

<NAK> 0x15 Negative acknowledge – Negative response on command

<ESC> 0x1B Escape – Spontaneous messages from print

<DC1> 0x11 Device control1 – Streaming data from bank 1

<DC2> 0x12 Device control2 – Streaming data from bank 2

Command details

All communication back and forth between the PC and the device shall be ended with a linefeed and newline character, <LF><NL>. All responses from the device will have a header which tells what kind of response it is. See the paragraph "Communication headers from device response to PC" above.

Set calibration value of excitation current

This command is used to set the calibration value of the excitation current. Each of the eight channels has four different excitation currents to choose from. Each of these selections needs its own calibration value.

These values are only settable from factory

Command:

CALibration:CHn:CURRent:SELm value short form CAL:CHn:CURR:SELm value

Response:

:CAL:CHn:CURR:SELm value

- n is the channel number which can be 1-8

- m is the selection of excitation current which can be 1-4, 1 being the lowest current and 4 the highest current
- value is an integer representation of the current in Pico ampere (pA – 10^{-12})

Example:

Command CAL:CH4:CURR:SEL2 3000000<LF><NL>

Response <ACK>:CAL:CH4:CURR:SEL2 3000000<LF><NL>

This will set the calibration value for the second lowest excitation current for channel 4 to 3000000pA which is the same as 3uA or $3*10^{-6}$A.

Get calibration value of excitation current

This command is used to get the calibration value of the excitation current from the device. Each of the eight channels has four different excitation currents to choose from. Each of these selections needs its own calibration value.

These values are only settable from factory

Command:

CALibration:CHn:CURRent:SELm? short form CAL:CHn:CURR:SELm?

Response:

:CAL:CHn:CURR:SELm? value

- n is the channel number which can be 1-8
- m is the selection of excitation current which can be 1-4, 1 being the lowest current and 4 the highest current
- value is an integer representation of the current in Pico ampere (pA – 10^{-12})

Example:

Command CAL:CH4:CURR:SEL2?<LF><NL>

Response <ACK>:CAL:CH4:CURR:SEL2? 1000000<LF><NL>

This will get the calibration value for the second highest excitation current for channel 4. In this example the value is 1000000pA which is the same as 1uA or $1*10^{-6}$A.

Set calibration value of the device offset capacitance

This command is used to set the calibration value of the device offset capacitance. Each of the eight channels has its own value.

These values are only settable from factory

Command:

CALibration:CHn:CAPacitance value short form CAL:CHn:CAP value

Response:

:CAL:CHn:CAP value

- n is the channel number which can be 1-8
- value is an integer representation of the device offset capacitance in Femto farad (fF – 10^{-15}F)

Example:

Command CAL:CH1:CAP 25000<LF><NL>

Response <ACK>:CAL:CH1:CAP 25000<LF><NL>

This will set the calibration value of the device offset capacitance for channel 1 to 25000fF which is the same $25*10^{-12}$F.

Get calibration value of the device offset capacitance

This command is used to get the calibration value of the device offset capacitance. Each of the eight channels has its own value.

These values are only settable from factory

Command:

CALibration:CHn:CAPacitance? short form CAL:CHn:CAP?

Response:

:CAL:CHn:CAP? value

- n is the channel number which can be 1-8
- value is an integer representation of the device offset capacitance in Femto farad (fF – 10^{-15}F)

Example:

Command CAL:CH2:CAP?<LF><NL>

Response <ACK>:CAL:CH2:CAP? 15000<LF><NL>

This will get the calibration value of the device offset capacitance for channel 2. In this example the value is 15000fF which is the same $15*10^{-12}$F.

Set excitation frequency of the device

This command is used to set the excitation frequency of the device. The eight sensor channels of the device are placed in two banks with four channels in each. Each bank coincides with the two ADCs measuring the voltage signals on the sensors. Bank 1 contains sensor 1,2,5 and 6, and Bank 2 contains sensor 3,4,7 and 8.

Command:

CONFiguration:BANKn:EXCitation:FREQuency value short form CONF:BANKn:EXC:FREQ value

Response:

:CONF:BANKn:EXC:FREQ value

- n is the bank number which can be either 1 or 2
- value is an integer representation of the excitation frequency in milli Hertz (mHz – 10^{-3}Hz)

Example:

Command CONF:BANK1:EXC:FREQ 100000<LF><NL>

Response <ACK>:CONF:BANK1:EXC:FREQ 100000<LF><NL>

This will set the excitation frequency for bank 1 to 100000mHz which is the same as 100Hz.

Get excitation frequency of the device

This command is used to get the excitation frequency of the device. The eight sensor channels of the device are placed in two banks with four channels in each. Each bank coincides with the two ADCs measuring the voltage signals on the sensors. Bank 1 contains sensor 1,2,5 and 6, and Bank 2 contains sensor 3,4,7 and 8.

Command:

CONFiguration:BANKn:EXCitation:FREQuency? short form CONF:BANKn:EXC:FREQ?

Response:

:CONF:BANKn:EXC:FREQ? value

- n is the bank number which can be either 1 or 2
- value is an integer representation of the excitation frequency in milli Hertz (mHz – 10^{-3}Hz)

Example:

Command CONF:BANK1:EXC:FREQ?<LF><NL>

Response <ACK>:CONF:BANK1:EXC:FREQ? 500000<LF><NL>

This will get the excitation frequency for bank 1. In this example the value is 500000mHz which is the same as 500Hz.

Set update frequency of the device

This command is used to set the update frequency of the device. The eight sensor channels of the device are placed in two banks with four channels in each. Each bank coincides with the two ADCs measuring the voltage signals on the sensors. Bank 1 contains sensor 1,2,5 and 6, and Bank 2 contains sensor 3,4,7 and 8.

Command:

CONFiguration:BANKn:UPDate:FREQuency value short form CONF:BANKn:UPD:FREQ value

Response:

:CONF:BANKn:UPD:FREQ value

- n is the bank number which can be either 1 or 2
- value is an integer which provides the update frequency by taking the excitation frequency and divide with this integer. The value can be set from 1 to 999

Example:

Command CONF:BANK2:UPD:FREQ 10<LF><NL>

Response <ACK>:CONF:BANK2:UPD:FREQ 10<LF><NL>

This will set the update frequency for bank 2 to 50Hz assuming that the excitation frequency is 500Hz.

Get update frequency of the device

This command is used to get the update frequency of the device. The eight sensor channels of the device are placed in two banks with four channels in each. Each bank coincides with the two ADCs measuring the voltage signals on the sensors. Bank 1 contains sensor 1,2,5 and 6, and Bank 2 contains sensor 3,4,7 and 8.

Command:

CONFiguration:BANKn:UPDate:FREQuency? short form CONF:BANKn:UPD:FREQ?

Response:

:CONF:BANKn:UPD:FREQ? value

- n is the bank number which can be either 1 or 2
- value is an integer which provides the update frequency by taking the excitation frequency and divide with this integer. The value can be set from 1 to 999

Example:

Command CONF:BANK1:UPD:FREQ?<LF><NL>

Response <ACK>:CONF:BANK1:UPD:FREQ? 25<LF><NL>

This will get the update frequency for bank 1. Assuming that the excitation frequency is 500Hz, this results in an update rate of 20Hz.

Set packet size of the device

This command is used to set packet size of the device. The packet size tells how many measurements per channel to send in each streaming response. This value is set for each bank.

Command:

CONFiguration:BANKn:PACKetsize value short form CONF:BANKn:PACK value

Response:

:CONF:BANKn:PACK value

- n is the bank number which can be either 1 or 2
- value is an integer representation of the number of measurements per channel to send in each streaming response. The packet size can be set from 1 to 19

Example:

Command CONF:BANK1:PACK 10<LF><NL>

Response <ACK>:CONF:BANK1:PACK 10<LF><NL>

This will set the packet size for bank 1 to 10 measurements per channel per streaming response.

Get packet size of the device

This command is used to get packet size of the device. The packet size tells how many measurements per channel to send in each streaming response. This value is set for each bank.

Command:

CONFiguration:BANKn:PACKetsize? short form CONF:BANKn:PACK?

Response:

:CONF:BANKn:PACK? value

- n is the bank number which can be either 1 or 2
- value is an integer representation of the number of measurements per channel to send in each streaming response. The packet size can be set from 1 to 19

Example:

Command CONF:BANK1:PACK?<LF><NL>

Response <ACK>:CONF:BANK1:PACK? 5<LF><NL>

This will get the packet size for bank 1. In this example the value is 5 measurements per channel per streaming response.

Set selection of excitation current

This command is used to set which excitation current to use. Each of the eight channels has four different excitation currents to choose from.

Command:

CONFiguration:CHn:CURRent:SELect value short form CONF:CHn:CURR:SEL value

Response:

:CONF:CHn:CURR:SEL value

- n is the channel number which can be 1-8
- value is the selection of excitation current which can be 1-4, 1 being the lowest current and 4 the highest current

Example:

Command CONF:CH2:CURR:SEL 2<LF><NL>

Response <ACK>:CONF:CH2:CURR:SEL 2<LF><NL>

This will select the second lowest excitation current for channel 2.

Get selection of excitation current

This command is used to get which excitation current currently selected. Each of the eight channels has four different excitation currents to choose from.

Command:

CONFiguration:CHn:CURRent:SELect? short form CONF:CHn:CURR:SEL?

Response:

:CONF:CHn:CURR:SEL? value

- n is the channel number which can be 1-8
- value is the selection of excitation current which can be 1-4, 1 being the lowest current and 4 the highest current

Example:

Command CONF:CH3:CURR:SEL?<LF><NL>

Response <ACK>:CONF:CH3:CURR:SEL 4<LF><NL>

This will get the excitation current selection for channel 3. In this example the value is 4 which is the highest excitation current.

Set length of averaging buffer

This command is used to set the length of the averaging buffer. The averaging buffer is used to make a running average of the measured values.

Command:

CONFiguration:CHn:AVGbuf value short form CONF:CHn:AVG value

Response:

:CONF:CHn:AVG value

- n is the channel number which can be 1-8
- value is an integer representation of the length of the averaging buffer which can have a value from 1 to 128

Example:

Command CONF:CH5:AVG 64<LF><NL>

Response <ACK>:CONF:CH5:AVG 64<LF><NL>

This will set the length of the averaging buffer for channel 5 to 64.

Get length of averaging buffer

This command is used to get the length of the averaging buffer. The averaging buffer is used to make a running average of the measured values.

Command:

CONFiguration:CHn:AVGbuf? short form CONF:CHn:AVG?

Response:

:CONF:CHn:AVG? value

- n is the channel number which can be 1-8
- value is an integer representation of the length of the averaging buffer which can have a value from 1 to 128

Example:

Command CONF:CH7:AVG?<LF><NL>

Response <ACK>:CONF:CH7:AVG? 20<LF><NL>

This will get the length of the averaging buffer for channel 7. In this example the length is 20.

Set capacitance measurement state

This command is used to set the capacitance measurement state to ON or OFF.

Command:

CONFiguration:CHn:MEAsure:CAPacitance state short form CONF:CHn:MEA:CAP state

Response:

:CONF:CHn:MEA:CAP state

- n is the channel number which can be 1-8
- state is either 0 or 1 equal to OFF or ON

Example:

Command CONF:CH1:MEA:CAP 1<LF><NL>

Response <ACK>:CONF:CH1:MEA:CAP 1<LF><NL>

This will set the capacitance measurement state off channel 1 to ON.

Get capacitance measurement state

This command is used to get the capacitance measurement state.

Command:

CONFiguration:CHn:MEAsure:CAPacitance? short form CONF:CHn:MEA:CAP?

Response:

:CONF:CHn:MEA:CAP? state

- n is the channel number which can be 1-8
- state is either 0 or 1 equal to OFF or ON

Example:

Command CONF:CH8:MEA:CAP?<LF><NL>

Response <ACK>:CONF:CH8:MEA:CAP? 0<LF><NL>

This will get the capacitance measurement state off channel 8. In this example the state is OFF.

Set serial resistance (ESR) measurement state

This command is used to set the serial resistance (ESR) measurement state to ON or OFF.

Command:

CONFiguration:CHn:MEAsure:ESR state short form CONF:CHn:MEA:ESR state

Response:

:CONF:CHn:MEA:ESR state

- n is the channel number which can be 1-8
- state is either 0 or 1 equal to OFF or ON

Example:

Command CONF:CH3:MEA:ESR 0<LF><NL>

Response <ACK>:CONF:CH3:MEA:ESR 0<LF><NL>

This will set the serial resistance (ESR) measurement state off channel 3 to OFF.

Get serial resistance (ESR) measurement state

This command is used to get the serial resistance (ESR) measurement state.

Command:

CONFiguration:CHn:MEAsure:ESR? short form CONF:CHn:MEA:ESR?

Response:

:CONF:CHn:MEA:ESR? state

- n is the channel number which can be 1-8
- state is either 0 or 1 equal to OFF or ON

Example:

Command CONF:CH8:MEA:ESR?<LF><NL>

Response <ACK>:CONF:CH8:MEA:ESR? 1<LF><NL>

This will get the serial resistance (ESR) measurement state off channel 8. In this example the state is ON.

Set Bluetooth ID

This command is used to set the Bluetooth ID which is the name that the Bluetooth module will broadcast.

Command:

CONFiguration:BLUEtooth:ID value short form CONF:BLUE:ID value

Response:

:CONF:BLUE:ID value

- value is the alphanumeric name of the device which can maximum be 12 characters long

Example:

Command CONF:BLUE:ID LEAPTECH0001<LF><NL>

Response <ACK>:CONF:BLUE:ID LEAPTECH0001<LF><NL>

This will set the Bluetooth ID to LEAPTECH0001.

Get Bluetooth ID

This command is used to get the Bluetooth ID which is the name that the Bluetooth module will broadcast.

Command:

CONFiguration:BLUEtooth:ID? short form CONF:BLUE:ID?

Response:

:CONF:BLUE:ID? value

- value is the alphanumeric name of the device which can maximum be 12 characters long

Example:

Command CONF:BLUE:ID?<LF><NL>

Response <ACK>:CONF:BLUE:ID LEAPTECH0002<LF><NL>

This will get the Bluetooth ID. In this example the Bluetooth ID is LEAPTECH0002.

Set streaming method

This command is used to set the streaming method to either ASCII or binary streaming. The binary streaming option is made in order to optimize streaming speed.

Command:

CONFiguration:STREAMing:METHod state short form CONF:STREAM:METH state

Response:

:CONF:STREAM:METH state

- state is either 0 or 1 equal to either ASCII or binary streaming

Example:

Command CONF:STREAM:METH 1<LF><NL>

Response <ACK>:CONF:STREAM:METH 1<LF><NL>

This will turn on binary streaming.

Get streaming method

This command is used to get the streaming method which is either ASCII or binary streaming. The binary streaming option is made in order to optimize streaming speed.

Command:

CONFiguration:STREAMing:METHod? short form CONF:STREAM:METH?

Response:

:CONF:STREAM:METH? state

- state is either 0 or 1 equal to either ASCII or binary streaming

Example:

Command CONF:STREAM:METH?<LF><NL>

Response <ACK>:CONF:STREAM:METH? 0<LF><NL>

This will get the streaming method which in this example is ASCII streaming.

Set trigger

This command is used to decide which trigger clock to use for the interrupts that control the excitation and measurement module. It can be set to use either the internal trigger or an external trigger. This option is provided for the purpose of synchronizing multiple devices. The internal trigger clock runs at 9998Hz. The external trigger clock should be set to approximately the same value.

Command:

CONFiguration:TRIGger state short form CONF:TRIG state

Response:

:CONF:TRIG state

- state is either 0 or 1 equal to either internal or external trigger

Example:

Command CONF:TRIG 1<LF><NL>

Response <ACK>:CONF:TRIG 1<LF><NL>

This will select the external trigger.

Get trigger

This command is used to get which trigger clock is used for the interrupts that control the excitation and measurement module. It can be set to use either the internal trigger or an external trigger. This option is provided for the purpose of synchronizing multiple devices. The internal trigger clock runs at 9998Hz. The external trigger clock should be set to approximately the same value.

Command:

CONFiguration:TRIGger? short form CONF:TRIGger?

Response:

:CONF:TRIGger? state

- state is either 0 or 1 equal to either internal or external trigger

Example:

Command CONF:TRIG?<LF><NL>

Response <ACK>:CONF:TRIG? 0<LF><NL>

This will get the selected trigger. In this example the internal trigger is selected.

Set streaming state

This command is used to turn streaming ON or OFF. It is only channels which are turned ON that are streamed. The streaming responses are divided into two different responses which links to the two banks.

To see the streaming response, see the paragraph “Streaming response” below

Command:

STREAMing state short form STREAM state

Response:

:STREAM state

- state is either 0 or 1 equal to OFF or ON

Example:

Command STREAM 1<LF><NL>

Response <ACK>:STREAM 1<LF><NL>

This will turn ON streaming.

Get streaming state

This command is used to get the streaming state.

Command:

STREAMing? short form STREAM?

Response:

:STREAM? state

- state is either 0 or 1 equal to OFF or ON

Example:

Command STREAM?<LF><NL>

Response <ACK>:STREAM? 0<LF><NL>

This will get the streaming state. In this example the streaming is turned OFF.

Set streaming state for a bank

This command is used to turn streaming ON or OFF for one of the banks. It is only channels which are turned ON in the selected bank that are streamed.

To see the streaming response, see the paragraph “Streaming response” below

Command:

STREAMing:BANKn state short form STREAM:BANKn state

Response:

:STREAM:BANKn state

- n is the bank number which can be either 1 or 2
- state is either 0 or 1 equal to OFF or ON

Example:

Command STREAM:BANK1 1<LF><NL>

Response <ACK>:STREAM:BANK1 1<LF><NL>

This will turn ON streaming for bank 1.

Get streaming state for a bank

This command is used to get the streaming state for one of the banks.

Command:

STREAMing:BANKn? short form STREAM:BANKn?

Response:

:STREAM:BANKn? state

- n is the bank number which can be either 1 or 2
- state is either 0 or 1 equal to OFF or ON

Example:

Command STREAM:BANK2?<LF><NL>

Response <ACK>:STREAM:BANK2? 0<LF><NL>

This will get the streaming state for bank 2. In this example the streaming is turned OFF for bank 2.

Set streaming state for a channel

This command is used to turn streaming ON or OFF for one of the channels. This command is primarily made for troubleshooting, as it is also streaming the actual measured ADC values for the sensor. It is not all the data that is streamed.

To see the streaming response, see the paragraph “Streaming response for a channel” below

Command:

STREAMing:CHn state short form STREAM:CHn state

Response:

:STREAM:CHn state

- n is the channel number which can be 1-8
- state is either 0 or 1 equal to OFF or ON

Example:

Command STREAM:CH1 1<LF><NL>

Response <ACK>:STREAM:CH1 1<LF><NL>

This will turn ON streaming for channel 1.

Get streaming state for a channel

This command is used to get the streaming state for one of the channels. This command is primarily made for troubleshooting, as it is also streaming the actual measured ADC values for the sensor. It is not all the data that is streamed.

Command:

STREAMing:CHn? short form STREAM:CHn?

Response:

:STREAM:CHn? state

- n is the channel number which can be 1-8
- state is either 0 or 1 equal to OFF or ON

Example:

Command STREAM:CH2?<LF><NL>

Response <ACK>:STREAM:CH2? 0<LF><NL>

This will get the streaming state for channel 2. In this example the streaming is turned OFF for channel 2.

Streaming response

When streaming is turned on, the measurements for the channels which are turned on is returned in two streaming responses, one for each bank of channels. The update rate decides how many measurements per channel per second that will be streamed. The packet size decides how many measurements per channel to put in each response.

Depending on Stream method either being ASCII or binary, the response will look different.

ASCII streaming response

Response for bank1:

CAP1 CAP2 CAP5 CAP6 ESR1 ESR2 ESR5 ESR6 : CAP1 CAP2 CAP5 CAP6 ESR1 ESR2 ESR5 ESR6

Response for bank2:

CAP3 CAP4 CAP7 CAP8 ESR3 ESR4 ESR7 ESR8 : CAP3 CAP4 CAP7 CAP8 ESR3 ESR4 ESR7 ESR8

- CAP1 to CAP8 is the measured capacitance values for the eight channels. The values are integer representations of the capacitances in Femto farad (fF – 10^{-15}F). If the measurement is turned OFF the value is returned as NA.
- ESR1 to ESR8 is the measured serial resistances for the eight channels. The values are integer representations of the serial resistances in Ohm. If the measurement is turned OFF the value is returned as NA.
- Each measurement set in a package is separated by a colon (:).

Example for bank1:

<DC1>:123456 234567 NA NA 100000 50000 NA NA : 123456 234567 NA NA 100000 50000 NA NA : 123456 234567 NA NA 100000 50000 NA NA<LF><NL>

In the above example, channel 1 and 2 for both capacitance and serial resistance is turned on. The packet size is 3 which therefore gives three measurements in each response. The measurements are the same all three times and are 123.456pF for channel 1, 234.567pF for channel 2, 100kOhm for channel 1 and 50kOhm for channel 2.

Example for bank2:

<DC2>:123456 234567 NA NA 100000 50000 NA NA : 123456 234567 NA NA 100000 50000 NA NA : 123456 234567 NA NA 100000 50000 NA NA<LF><NL>

In the above example, channel 3 and 4 for both capacitance and serial resistance is turned on. The packet size is 3 which therefore gives three measurements in each response. The measurements are the same all three times and are 123.456pF for channel 3, 234.567pF for channel 4, 100kOhm for channel 3 and 50kOhm for channel 4.

Binary streaming response

In the binary streaming response, only the capacitance values will be streamed. If the ESR measurement are needed, the ASCII streaming response has to be used.

Response for bank1:

#ofbytesCAP1CAP2CAP5CAP6CAP1CAP2CAP5CAP6

Response for bank2:

#ofbytes CAP3CAP4CAP7CAP8CAP3CAP4CAP7CAP8

CAP1 to CAP8 is the measured capacitance values for the eight channels. The values are binary signed integer four byte long representations of the capacitances in Femto farad (fF – 10^{-15}F). If

the measurement is turned OFF the value is returned as 0. For each bank one measurement set is 16 bytes long. Each measurement set just comes after each other, so if the packet size is five the full length of measurements will be 80 bytes long. #ofbytes is a three character ASCII representation of the number of bytes in the packet, e.g. 80 if the packet size is five.

Example for bank1:

The example is displayed in a two-byte hex format, though the #ofbytes is three ASCII characters put in the "" quotations.

<DC1> "032" 000F 4240 001E 8480 002D C6C0 003D 0900 000F 6950 001E D2A0 002E 3BF0 003D A540 <LF><NL>

In the above example the packet size is 2 which can be by the number of bytes in the measurement is 32 bytes, remember each measurement set is 16 bytes long. The measurements in the example are:

1000pF 2000pF 3000pF 4000pF 1010pF 2020pF 3030pF 4040pF

Example for bank2:

The example is displayed in a two-byte hex format, though the #ofbytes is three ASCII characters put in the "" quotations.

<DC2> "032" 0000 0064 0000 0000 0000 0000 0000 0000 0000 0064 0000 0000 0000 0000 0000 0000 <LF><NL>

In the above example the packet size is 2 which can be by the number of bytes in the measurement is 32 bytes, remember each measurement set is 16 bytes long. The measurements in the example are:

0.1pF 0pF 0pF 0pF 0.1pF 0pF 0pF 0pF

Set streaming IMU state

This command is not implemented yet.

Command:

STREAMing:IMU state short form STREAM:IMU state

Response:

:STREAM:IMU state

- state is either 0 or 1 equal to OFF or ON

Example:

Command STREAM:IMU 1<LF><NL>

Response <ACK>:STREAM:IMU 1<LF><NL>

This will turn ON streaming for the IMU.

Get streaming IMU state

This command is not implemented yet.

Command:

STREAMing:IMU? short form STREAM:IMU?

Response:

:STREAM:IMU? state

- state is either 0 or 1 equal to OFF or ON

Example:

Command STREAM:IMU?<LF><NL>

Response <ACK>:STREAM:IMU? 0<LF><NL>

This will get the streaming state for the IMU. In this example the streaming is turned OFF for the IMU.

Get capacitance measurement

This command is used to get the current capacitance measurement.

Command:

MEASurement:CHn:CAPacitance? short form MEAS:CHn:CAP?

Response:

:MEAS:CHn:CAP? value

- n is the channel number which can be 1-8
- value is an integer representation of the capacitance in Femto farad (fF – 10^{-15}F). If the measurement is turned OFF the value is returned as NA.

Example:

Command MEAS:CH2:CAP?<LF><NL>

Response <ACK>:MEAS:CH2:CAP? 1250000<LF><NL>

This will get the capacitance of channel 2. In this example the capacitance is 1250000fF which is similar to 1.25nF.

Get serial resistance (ESR) measurement

This command is used to get the current serial resistance (ESR) measurement.

Command:

MEASurement:CHn:ESR? short form MEAS:CHn:ESR?

Response:

:MEAS:CHn:ESR? value

- n is the channel number which can be 1-8
- value is an integer representation of the serial resistance (ESR) in Ohm. If the measurement is turned OFF the value is returned as NA.

Example:

Command MEAS:CH5:ESR?<LF><NL>

Response <ACK>:MEAS:CH5:ESR? 25000<LF><NL>

This will get the serial resistance (ESR) of channel 5. In this example the serial resistance (ESR) is 25000Ohm which is similar to 25kOhm.

Get battery status

This command is used to get the current status of the battery.

Command:

MEASurement:BATTery? short form MEAS:BATT?

Response:

:MEAS:BATT? value

- value is an integer representation of the battery voltage represented in a 12bit digitized 5V value. This means that the battery voltage is equal to "5V * value / 4095". If the battery is fully charged the voltage will be 4.2V equal to a value of 3440 and if the battery is completely drained the voltage will be 3.0V equal to a value of 2457.

Example:

Command MEAS:BATT?<LF><NL>

Response <ACK>:MEAS:BATT? 3276<LF><NL>

This will get the battery status. In this example the value is 3276 which equals a battery voltage of 4V.

Read hardware revision

This command is used to get the hardware revision of the device

Command:

READ:HardWare:REVision? short form READ:HW:REV?

Response:

:READ:HW:REV? value

- value is the hardware revision of the device in the format major.minor e.g. 1.02

Example:

Command READ:HW:REV?<LF><NL>

Response <ACK>:READ:HW:REV? 2.01<LF><NL>

This will get hardware revision of the device. In this example the revision is 2.01.

Read software revision

This command is used to get the software revision of the device

Command:

READ:SoftWare:REVision? short form READ:SW:REV?

Response:

:READ:SW:REV? value

- value is the software revision of the device in the format major.minor e.g. 1.02

Example:

Command READ:SW:REV?<LF><NL>

Response <ACK>:READ:SW:REV? 3.05<LF><NL>

This will get software revision of the device. In this example the revision is 3.05.

Verbose

This command is used to set the verbose level of the print?

Command:

VERBOSE value

Response:

:VERBOSE value

- value is the verbose level which can be set from 1-3

Example:

Command VERBOSE 3<LF><NL>

Response <ACK>:VERBOSE 3<LF><NL>

This will set the verbose level to 3.

Configure Bluetooth radio

This command is used to configure the Bluetooth radio. For example, when the Bluetooth ID has been changed, this command has to be called in order for the broadcasted Bluetooth ID to change (configuration of the Bluetooth radio also happens at power up of the device).

Command:

RADIO:CONFIG

Response:

:RADIO:CONFIG

Example:

Command RADIO:CONFIG<LF><NL>

Response <ACK>:RADIO:CONFIG<LF><NL>

This will configure the Bluetooth radio.

Event handling (Spontaneous messages)

Syntax error

In case of a command not being understood by the device, it will send a syntax error message.

Response:

:Syntax error

Parameter error

In case of a parameter in a command is out of the defined limits for the parameter, the device will send a parameter error message.

Response:

:Parameter error

Support

Please contact LEAP Technology in case support is required on this document.

Contact@leaptechnology.com