4.2.2 Transfer protocol of measurement data stream

Refer to *Section 4.3.2: Command description*. Several data format are available, among them ASCII and binary.

4.2.3 Command management

The shell always replies to a command sent.

Feedback can be either:

- Acknowledge: string “ack” is sent back with the command string.
  The acknowledge is followed by the command effectively applied or command returned data.
- Error: string “err” is sent back with the command string.
  On following lines, a description of the error can be sent.

The commands can take some arguments, either string or numerical values.

Numerical values of arguments can be formatted with:

- Numerical characters only (possible when numbers are ≥ 1)
- Numerical characters with unit characters 'u', 'm', 'k', 'm'
- Numerical characters with powers of ten '-xx' or '+xx' (xx: number on two digits maximum)

Example: Value '2 milliseconds' can be entered with: '2 m' or '2-3'

4.3 Interface commands

Interface commands are available when PowerShield is in controlled by host mode.

4.3.1 Command list summary

When sending command “help” to PowerShield firmware, it returns back the list of available commands and a short description for each command.

Table 1. List summary as displayed by firmware when entering command “help”

PowerShield commands	
Command	**Description**
Common operation	
help	Displays list of commands
echo <arg1>	Loopback to check functionality of communication Rx and Tx. <arg1>: string of characters
powershield	Check PowerShield device availability, can be used to scan on which serial port is connected the PowerShield. Response: “PowerShield present” with board unique ID
version	Get PowerShield FW revision. Response: “<main>.<sub1>.<sub2>”

Table 1. List summary as displayed by firmware when entering command “help” (continued)

PowerShield commands	
Command	**Description**
htc	Host take control (goes from standalone mode to controlled by host mode)
htc	Host release control (goes from controlled by host mode to standalone mode)
lcd	Displays a custom string on LCD display when PowerShield is controlled by host. <arg1>: LCD line. Numerical value among list: {1, 2} <arg2>: string to be displayed, surrounded by double quotes and with 16 characters maximum. Example: lcd 1 “custom display” Note: Available only on standalone board X-NUCLEO-LPM01A.
psrst	Reset PowerShield (hardware reset, host communication have to be restored)
Measurement acquisition configuration	
volt <arg1>	Set or gets power supply voltage level, unit: V. <arg1>: set voltage: numerical value in range [1800 m; 3300 m] Default value: 3300 m Get voltage: string “get” (*Section 4.2.3: Command management*)
freq <arg1>	Set sampling frequency, unit: Hz. <arg1>: Numerical value among list: {100 k, 50 k, 20 k, 10 k, 5 k, 2 k, 1 k, 500, 200 100, 50, 20, 10, 5, 2, 1} Default value: 100 Hz (*Section 4.2.3: Command management*)
acqtime <arg1>	Set acquisition time, unit: s. <arg1>: For limited acquisition duration: numerical value in range: [100 µs; 10] For infinite acquisition duration: numerical value “0” or string “inf” Caution: maximum acquisition time depends on other parameters. Refer to *Table 2*. Default value: 10 s (*Section 4.2.3: Command management*)
acqmode <arg1>	Set acquisition mode: dynamic or static. Dynamic: current can vary, range [100 nA; 10 mA] Static: current must be constant, range [2 nA; 200 mA] <arg1>: string among list: {dyn, stat} available only on standalone board X-NUCLEO-LPM01A. Default value: “dyn”

Table 1. List summary as displayed by firmware when entering command “help” (continued)

PowerShield commands	
Command	**Description**
funcmode <arg1>	Set optimization of acquisition mode dynamic (applicable only with command “output” set to parameter “current”): optim: priority on current resolution (100 nA-10 mA), max sampling frequency at 100 kHz. high: high current (30 µA–10 mA), high sampling frequency (50-100 kHz), high resolution. <arg1>: String among list: {optim, high} Default value: “optim”
output <arg1>	Set output type. Current: instantaneous current. Energy: integrated energy, reset after each sample sent (integration time set by parameter “freq”, limited at 100 Hz max (⇔ 10 ms min)). <arg1>: string among list: {current, energy} Default value: “current”'
format <arg1>	Set measurement data format. Data format 1: ASCII, decimal basis. Format readable directly, but sampling frequency limited to 10 kHz. Decoding: 6409-07 ⇔ 6409 x 10^-7 = 640.9 µA Data format 2: Binary, hexadecimal basis. Format optimized data stream size. Decoding: 52A0 ⇔ (2A0)16 x 16^-5 = 640.9 µA <arg1>: string among list: {ascii_dec, bin_hexa} Caution: Data format depends on other parameters. Refer to *Table 2*. Default value: 'ascii_dec'
trigsrc <arg1>	Set trigger source to start measurement acquisition. Software trigger source (immediate trig after software start), trigger from signal rising edge on Arduino connector D7 (via solder bridge). <arg1>: string among list: {sw, d7} Default value: “sw”
trigdelay <arg1>	Set trigger delay between target power-up and starts measurement acquisition, unit: s, resolution: 1 ms. <arg1>: numerical value in range [0; 30] Default value: 1 m
currthre <arg1>	Set current threshold to trig an event, unit: A. Event triggered when threshold exceeded: signal generated on Arduino connector D2 or D3 (via solder bridge) and LED4 (green) turned on. <arg1>: Numerical value in range [0; 10 m] Default value: 1 m

Table 1. List summary as displayed by firmware when entering command “help” (continued)

PowerShield commands	
Command	**Description**
pwr <arg1> <arg2>	Set target power supply connection. – Automatic: On first run, power-on when acquisition start.Then, power state depends on command 'pwrend'. – Manual: Force power state. Note: Can be used during acquisition. To perform successive power off and on, it is preferable to use command 'targrst'. Optionally, connection status can be sent at the beginning and end of acquisition data stream. <arg1>: Set pwr: String among list: {'auto', 'on', 'off'} Default value: 'auto' Get pwr: String 'get' (response: state 'on' or 'off') <arg2>: Optional, string among list: {'nostatus', 'status'} Default value: 'nostatus'
pwrend <arg1>	Set target power supply to be applied after current measurement acquisition: power-on or power-off. <arg1>: String among list: {on, off} Default value: “on”
Measurement acquisition operation	
start	Starts current measurement
stop	Stops current measurement
targrst <arg1>	Reset target by disconnecting power supply during a configurable duration, unit: s. *Note: can be performed during acquisition to monitor target transient current consumption during its power-up.* <arg1>: numerical value in range [1 m; 1] or value “0” to let target powered-down
temp <arg1>	Gets temperature from temperature sensor on PowerShield board, on unit: Celsius or Fahrenheit degrees <arg1>: String among list: {degc, degf} Default value: “degc”
Note: Numerical values of arguments can be formatted in: – Numerical characters only, when numbers are ≥ 1) – Numerical characters with unit characters “u”, “m”, “k”, “m” – Numerical characters with power of ten “-xx” or “+xx” (xx: number on two digits maximum) Example: Value “2 milliseconds” can be entered with: “2 m” or “2-3”	

Table 2. Maximum acquisition time possible for a baudrate of 3686400 bauds

Format	Frequency	Acqtime max	Correspondent number of samples
ascii_dec	≤ 5 kHz	unlimited	(unlimited)
ascii_dec	10 kHz	1 s	(10000)
ascii_dec	20 kHz	500 ms	(5000)
bin hexa	≤ 100 kHz	unlimited	(unlimited)

4.3.2 Command description

1. Commands: common operation
   These commands can be sent to PowerShield in both the functional standalone mode or controlled by host mode.

Table 3. “help” command (1)

Argument	(none)
Description	Displays list of commands

1. This command can be used when PowerShield is controlled in standalone mode.

Table 4. “echo” command

Argument	1. String of characters
Description	Loopback to check functionality of communication Rx and Tx

Table 5. “powershield” command(1)

Argument	(none)
Description	Checks PowerShield device availability, can be used to scan on which serial port is connected the PowerShield. Response: “PowerShield present” with board unique ID. Example of shell feedback: *PowerShield > ack powershield 540619864-1110659081-4784204*

1. This command can be used when PowerShield is controlled in standalone mode

Table 6. “version” command(1)

Argument	(none)
Description	Gets PowerShield firmware revision. Response: “<main>.<sub1>.<sub2>” Example of shell feedback: `PowerShield > ack version: 1.0.0`

1. This command can be used when PowerShield is controlled in standalone mode.

Table 7. “status” command(1)

Argument	(none)
Description	Gets PowerShield status. Response: “ok” or “error: <error description>” In case of error, running this command releases error status: turn-off LED red, clear PowerShield state machine error state.

1. This command can be used when PowerShield is controlled in standalone mode.

Table 8. “htc” command

Argument	(none)
Description	Host takes control (goes from standalone mode to controlled by host mode)

Table 9. “hrc” command

Argument	(none)
Description	Host releases control (goes from controlled by host mode to standalone mode)

Table 10. “lcd” command

Argument	1. LCD line. Numerical value among list: {1, 2} 2. String to be displayed, surrounded by double quotes with 16 characters maximum. Example of command sent to shell: `lcd 1 "  custom display"`
Description	Display a custom string on LCD display when PowerShield is controlled by host. Note: Available only on standalone board X-NUCLEO-LPM01A.

Table 11. “psrst” command(1)

Argument	(none)
Description	Reset PowerShield (hardware reset, host communication have to be restored).

1. This command can be used when PowerShield is controlled in standalone mode.

2. Commands: measurement acquisition configuration

Table 12. “volt” command

Argument	1. Set voltage: Numerical value in range [1800 m; 3300 m] Default value: 3000 m Get voltage: String “get”
Description	Set or gets power supply voltage level, unit: V.

Table 13. “freq” command

Argument	1. Numerical value among list: {100 k, 50 k, 20 k, 10 k, 5 k, 2 k, 1 k, 500, 200, 100, 50, 20, 10, 5, 2, 1} Default value: 100 Hz
Description	Set sampling frequency, unit: Hz

Table 14. “acqtime” command

Argument	1. For limited acquisition duration: – Numerical value in range: [100 µ; 10] For infinite acquisition duration: – Numerical value “0” or string “inf” Caution: Maximum acquisition time depends on other parameters. Refer to *Section 5.2.1: Acquisition frequency limitations*. Default value: 10 s
Description	Set acquisition time, unit: s

Table 15. “acqmode” command

Argument	1. String among list: {dyn, stat} Default value: “dyn”
Description	Set acquisition mode: dynamic or static. dynamic: current can vary, range [100 nA; 10 mA] static: current must be constant, range [2 nA; 200 mA], available only on standalone board X-NUCLEO-LPM01A.

Table 16. “funcmode” command

Argument	1. String among list: {optim, high} Default value: “optim”
Description	Set optimization of acquisition mode dynamic (applicable only with command “output” set to parameter “current”): optim: priority on current resolution (100 nA-10 mA), max sampling frequency at 100 kHz. high: high current (30 µA–10 mA), high sampling frequency (50-100 kHz), high resolution. *Note: In mode optimized for amplitude, the benefit is to have complete dynamic of signal [100 nA; 10 mA] but with drawback of timing artifacts (few ms max)*

Table 17. “output” command

Argument	1. String among list: {current, energy} Default value: “current”
Description	Set output type. current: instantaneous current energy: integrated energy, reset after each sample sent (integration time set by parameter “freq”, limited at 100 Hz max (⇔ 10 ms min)).

Table 18. “format” command

Argument	1. String among list: {ascii_dec, bin_hexa} Caution: Data format depends on other parameters. Refer to *Section 5.2.1: Acquisition frequency limitations*. Default value: “ascii_dec”
Description	Set measurement data format. Data format 1: ASCII, decimal basis. – Format readable directly, but sampling frequency limited to 10 kHz. Decoding: 6409-07 ⇔ 6409 x 10^-7 = 640.9 µA Data format 2: Binary, hexadecimal basis. – Format optimized data stream size. Decoding: 52A0 ⇔ (2A0)16 x 16^-5 = 640.9 µA

Table 19. “trigsrc” command

Argument	1. String among list: {sw, d7} Default value: “sw”
Description	Set trigger source to start measurement acquisition: trigger source software (immediate trig after software start), trigger from external signal rising or falling edge on Arduino connector D7 (via solder bridge). *Note: Trigger from external signal also requires command “start” (similar software start) to arm the trigger, then following triggers are effective without any command.* Command “stop” disarms the trigger (acquisition stop after acquisition time elapsed does not disarm the trigger). *Note: When trigger source from Arduino connector D7 is used, the alternate communication interface with UART connector cannot be used after trigger is armed (due to Arduino connector D7 and UART Rx sharing the same input).*

Table 20. “trigdelay” command

Argument	1. Numerical value in range [0; 600] Default value: 1 m
Description	Set trigger delay between target power-up and starts measurement acquisition, unit: s, resolution: 1 ms. This command allows the voltage and current to stabilize before start of current acquisition.

Table 21. “currthre” command

Argument	1. Numerical value in range [0; 10 m] Default value: 1 m
Description	Set current threshold to trig an event, unit: A. Event triggered when threshold exceeded: signal generated on Arduino connector D2 or D3 (via solder bridge) and LED4 (green) turned on.

Table 22. “pwr” command

Argument	1. Set pwr: String among list: {'auto', 'on', 'off'} Default value: 'auto' Get pwr: String 'get' (response: state 'on' or 'off') 2. Optional, string among list: {'nostatus', 'status'} Default value: 'nostatus'
Description	Set target power supply connection. – Automatic: • On first run, power-on when acquisition start. • Then, power state depends on command 'pwrend'. – Manual: • Force power state. *Note: Can be used during acquisition. To perform successive power off and on, it is preferable to use command 'targrst'.* Optionally, connection status can be sent at the beginning and end of acquisition data stream. *Note: This command is available since PowerShield FW revision 1.0.2.*

Table 23. “pwrend” command

Argument	1. String among list: {on, off} Default value: “on”
Description	Set target power supply to be applied after current measurement acquisition: power-on or power-off.

3. Commands: Measurement acquisition operation

Table 24. “start” command

Argument	(none)
Description	Starts acquisition (measurement of current or energy depending on configuration).

Table 25. “stop” command

Argument	(none)
Description	Stops acquisition. If acquisition is set to a finite duration, it stops when reaching the target duration.

Table 26. “targrst” command

Argument	1. Numerical value in range [1 m; 1] or value “0” to let target power-down.
Description	Reset target by disconnecting power supply during a configurable duration, unit: s. *Note: can be performed during acquisition to monitor target transient current consumption during its power-up.*

Table 27. “temp” command

Argument	1. String among list: {degc, degf} Default value: “degc”
Description	Gets temperature from temperature sensor on PowerShield board, on unit: Celsius or Fahrenheit degrees. *Note: reported temperature is an approximation of ambient temperature; measured temperature corresponds to temperature on board surface, which is higher than ambient temperature of approximatively 3 Celsius degrees (due to board self-heating with surrounding components), therefore this value is subtracted to measured temperature on board surface.* *In case of specific conditions (for example, forced convection in a laboratory oven), user must apply a temperature offset to reported temperature.*

4. Commands: Board state operation

Table 28. “autotest” command

Argument	1. Optional: string among list: {start, status} or no argument equivalent to value: “start”
Description	Performs board auto-test and display auto-test results. *Note: auto-test is done at PowerShield power-up.* *Argument “status” can be used to check results of auto-test done at PowerShield power-up.*

Table 29. “calib” command

Argument	(none)
Description	Performs board self-calibration. *Note: new calibration should be performed when temperature shifts of more than 5 °C since the previous calibration.*

4.4 Data stream format

Measurement data contain the main information: current or energy (depending on configuration sent to PowerShield shell).

Note: Information of voltage is not sent. Effective voltage is assumed to be close to targeted voltage (tolerance approximatively ±1%).

Note: Information of timing is not sent and must be deduced from data count. For example, if the acquisition frequency is set at 10 kHz, the first data corresponds to 10 µs, the second data to 20 µs, the third data to 30 µs, and so on.

4.4.1 Data format 1: ASCII, decimal basis

1. Measurement data of current or energy
   Format intended when PowerShield is used with a COM port terminal: data are formatted in ASCII characters, values are in decimal basis.

Note: Due to higher data size in ASCII format and to data bandwidth constraints, this data format can be used with a sampling up to 10 ksamples/sec.

Each measurement data is formatted on eight ASCII characters:

Table 30. ASCII characters description

Byte on serial port	Byte number	Description
ASCII [0; 9]	1	Current measurement digit 4
ASCII [0; 9]	2	Current measurement digit 3
ASCII [0; 9]	3	Current measurement digit 2
ASCII [0; 9]	4	Current measurement digit 1
ASCII {'-';'+'}	5	Current measurement power of 10 sign
ASCII [0; 9]	6	Current measurement power of 10 value
ASCII [0; 9]	7	Current measurement power of 10 value
ASCII '\r'	8	Carriage return
ASCII '\n'	9	Line feedback

Example of data stream and corresponding conversion to decimal values:

$6409*10^{-7}$

$$(6409)_{10} \times 10^{-7} = 640.9 \cdot 10^{-6} = 640.9 \quad \mu A$$

2. Metadata inserted into data stream
   Metadata are inserted into data stream to provide other information.
   Data must be filtered in data stream to isolate measurement data (current or energy values) versus metadata.

Differentiator of measurement data versus metadata:

– Measurement data are beginning by a number in ASCII format (first byte corresponding to decimal values from 48 to 57).
– Metadata are beginning with a letter in ASCII format (first byte corresponding to decimal value other than a number, described above).

Metadata: Timestamp and buffer Tx load

Each 1000 samples, a timestamp is sent.

It can be used to check data count matches with each timestamp occurrence, and to resynchronize timing if some data have been lost.

Timestamp also includes information of time elapsed in milliseconds.

Additionally, the information of PowerShield buffer Tx load is added after time elapsed.

It is useful in case host encounters a delay on USB bus (can occur when host is busy by CPU load or file access delay), to take appropriate action before buffer overload and acquisition stop.

Timestamp format: ASCII, decimal format

Table 31. Timestamp, format 1: ASCII characters description

Byte on serial port	Byte number	Description
ASCII ‘\r’	1	Carriage return
ASCII ‘\n’	2	Line feedback
ASCII ‘T’	3	Timestamp tag characters
ASCII ‘i’	4	Timestamp tag characters
ASCII ‘m’	5	Timestamp tag characters
ASCII ‘e’	6	Timestamp tag characters
ASCII ‘s’	7	Timestamp tag characters
ASCII ‘t’	8	Timestamp tag characters
ASCII ‘a’	9	Timestamp tag characters
ASCII ‘m’	10	Timestamp tag characters
ASCII ‘p’	11	Timestamp tag characters
ASCII ‘:’	12	Timestamp tag characters
ASCII ‘ ’	13	Timestamp tag characters
ASCII [0; 9]	14	Timestamp value in s, digit 2
ASCII [0; 9]	15	Timestamp value in s, digit 1
ASCII [0; 9]	16	Timestamp value in s, digit 0
ASCII ‘s’	17	Timestamp value character
ASCII ‘ ‘	18	Timestamp value character
ASCII [0; 9]	19	Timestamp value in ms, digit 2
ASCII [0; 9]	20	Timestamp value in ms, digit 1
ASCII [0; 9]	21	Timestamp value in ms, digit 0

Table 31. Timestamp, format 1: ASCII characters description (continued)

Byte on serial port	Byte number	Description
ASCII ‘m’	22	Timestamp value character
ASCII ‘s’	23	Timestamp value character
ASCII ‘,‘	24	Timestamp tag characters
ASCII ‘ ‘	25	Timestamp tag characters
ASCII ‘b‘	26	Timestamp tag characters
ASCII ‘u‘	27	Timestamp tag characters
ASCII ‘f‘	28	Timestamp tag characters
ASCII ‘f‘	29	Timestamp tag characters
ASCII ‘ ‘	30	Timestamp tag characters
ASCII [0; 9]	31	Buffer Tx load value in percent, digit 1
ASCII [0; 9]	32	Buffer Tx load value in percent, digit 0
ASCII ‘%‘	33	Timestamp tag characters
ASCII ‘\r‘	34	Carriage return
ASCII ‘\n ‘	35	Line feedback

Metadata: Error

An error message (voltage drop) can be sent as a stream of ASCII characters.

Table 32. Error, format 1: ASCII characters description

Byte on serial port	Byte number	Description
ASCII ‘\r‘	1	Carriage return
ASCII ‘\n ‘	2	Line feedback
ASCII ‘e‘	3	Error tag characters
ASCII ‘r‘	4	Error tag characters
ASCII ‘r‘	5	Error tag characters
ASCII ‘o‘	6	Error tag characters
ASCII ‘r‘	7	Error tag characters
ASCII char	8	Message content: ASCII character
ASCII char	...	Message content: ASCII character
ASCII char	x	Message content: ASCII character
ASCII char	x + 1	Message content: ASCII character
ASCII ‘\r’	x + 2	Carriage return
ASCII ‘\n’	x + 3	Line feedback

Metadata: End of acquisition

Metadata send when acquisition is completed: integration time reached or command “stop” sent by user, and all data in Tx buffer sent to host.

Table 33. End of acquisition, format 1: ASCII characters description

Byte on serial port	Byte number	Description
ASCII ‘\r‘	1	Carriage return
ASCII ‘\n ‘	2	Line feedback
ASCII ‘e‘	3	End of acquisition tag characters
ASCII ‘n‘	4	End of acquisition tag characters
ASCII ‘d‘	5	End of acquisition tag characters
ASCII ‘\r‘	6	Carriage return
ASCII ‘\n ‘	7	Line feedback

Metadata: Power to target connection

Metadata sent as acknowledge and data of command “pwr get”.

Metadata also sent at the beginning (after metadata of acquisition start) and end (before metadata of acquisition end) of each acquisition if second parameter of command “pwr” is set to argument “status” (optional).

Power to target connection status is coded on two or three characters:

“off” <=> power off (power supply disconnected from target)

“on” <=> power on (power supply connected to target)

Table 34. Power to target, format 1: ASCII characters description

Byte on serial port	Byte number	Description
ASCII ‘\r‘	1	Carriage return
ASCII ‘\n ‘	2	Line feedback
ASCII ‘p‘	3	Power to target tag characters
ASCII ‘w‘	4	Power to target tag characters
ASCII ‘r‘	5	Power to target tag characters
ASCII ‘ ‘	6	Power to target tag characters
ASCII char	7	Power to target connection status characters
ASCII char	8	Power to target connection status characters
ASCII char	...	Power to target connection status characters
ASCII ‘\r‘	x	Carriage return
ASCII ‘\n ‘	x	Line feedback

Metadata: Summary

After end of acquisition, a summary is displayed between tags “summary begin” and “summary end”.

Description and data sent in ASCII: acquisition mode, sampling frequency

Table 35. Summary, format 1: ASCII characters description

Byte on serial port	Byte number	Description
ASCII ‘\r‘	1	Carriage return
ASCII ‘\n ‘	2	Line feedback
ASCII ‘s‘	3	Summary tag characters
ASCII ‘u‘	4	Summary tag characters
ASCII ‘m‘	5	Summary tag characters
ASCII ‘m‘	6	Summary tag characters
ASCII ‘a‘	7	Summary tag characters
ASCII ‘r‘	8	Summary tag characters
ASCII ‘y‘	9	Summary tag characters
ASCII ‘ ‘	10	Summary tag characters
ASCII ‘b‘	11	Summary tag characters
ASCII ‘e‘	12	Summary tag characters
ASCII ‘g‘	13	Summary tag characters
ASCII ‘\r‘	14	Carriage return
ASCII ‘\n ‘	15	Line feedback
ASCII [0; 9]	16	current measurement min digit 4
ASCII [0; 9]	17	current measurement min digit 3
ASCII [0; 9]	18	current measurement min digit 2
ASCII [0; 9]	19	current measurement min digit 1
ASCII {‘-’; ‘+’}	20	current measurement min power of 10 sign
ASCII [0; 9]	21	current measurement min power of 10 value
ASCII [0; 9]	22	current measurement min power of 10 value
ASCII ‘\r‘	23	Carriage return
ASCII ‘\n ‘	24	Line feedback
ASCII [0; 9]	25	current measurement max digit 4
ASCII [0; 9]	26	current measurement max digit 3
ASCII [0; 9]	27	current measurement max digit 2
ASCII [0; 9]	28	current measurement max digit 1
ASCII {‘-’; ‘+’}	29	current measurement max power of 10 sign
ASCII [0; 9]	30	current measurement max power of 10 value
ASCII [0; 9]	31	current measurement max power of 10 value

Table 35. Summary, format 1: ASCII characters description (continued)

Byte on serial port	Byte number	Description
ASCII '\r'	32	Carriage return
ASCII '\n '	33	Line feedback
ASCII 's'	34	Summary tag characters
ASCII 'u'	35	Summary tag characters
ASCII 'm'	36	Summary tag characters
ASCII 'm'	37	Summary tag characters
ASCII 'a'	38	Summary tag characters
ASCII 'r'	39	Summary tag characters
ASCII 'y'	40	Summary tag characters
ASCII ' '	41	Summary tag characters
ASCII 'e'	42	Summary tag characters
ASCII 'n'	43	Summary tag characters
ASCII 'd'	44	Summary tag characters
ASCII '\r'	45	Carriage return
ASCII '\n '	46	Line feedback

4.4.2 Data format 2: Binary, hexadecimal basis

1. Measurement data of current or energy

   Format intended when PowerShield is used with host software "Power Monitor": software must decode data from hexadecimal to decimal.

   This data format is optimized to have to lowest data width per measurement data.

   Each measurement data is formatted on two bytes (binary value, not ASCII):

   – Data size of two bytes characters compresses data size as much as possible (it allows the user to transmit a data stream of 50 ksamples/sec at bit rate 921.600 kbps).

   – Each data is coded in hexadecimal: 12 bits of data and four bits of negative power of 16.

Data accuracy: Decimation error of base 16 is data ±0.20% worst case.

Table 36. Serial byte 1

Serial data 1	Serial data 2	Serial data 3	Serial data 4	Serial data 5	Serial data 6	Serial data 7	Serial data 8	Serial stop bit
Current neg pow16 bit 3	Current neg pow16 bit 2	Current neg pow16 bit 1	Current neg pow16 bit 0	Current value bit 11	Current value bit 10	Current value bit 9	Current value bit 8	

Table 37. Serial byte 2

Serial data 1	Serial data 2	Serial data 3	Serial data 4	Serial data 5	Serial data 6	Serial data 7	Serial data 8	Serial stop bit
Current value bit 7	Current value bit 6	Current value bit 5	Current value bit 4	Current value bit 3	Current value bit 2	Current value bit 1	Current value bit 0	

Example of measurement data sent on serial port and corresponding conversion to decimal values:

52A0:

$$(2A0)_{16} \times 16^{-5} = (672)_{10} / 16^{5} = 640.9 \cdot 10^{-6} = 640.9 \quad \mu A$$

3145:

$$(145)_{16} \times 16^{-3} = (325)_{10} / 16^{3} = 793.5 \cdot 10^{-4} = 79.35 \quad mA$$

Note: Negative power of 16 is typically in the range of [-10; -3], allowing a current range of [0.2 nA; 999 mA].

Note: Negative power of 16 is limited to range {0 (0x0; 14 (0xE)}. Value 15 (0xF) is reserved as an information tag (refer to time stamp description).

2. Metadata inserted into data stream
   Metadata is inserted into data stream to provide other information.
   Data must be filtered in data stream to isolate measurement data (current or energy values) versus metadata.
   Differentiator of measurement data versus metadata:
   - Metadata
     - Metadata start: two consecutive bytes starting by 0xF (measurement data can have only one of the bytes having this value): 0xF0 and 0xFx (value depending on metadata type, see *Table 38*)
     - Metadata stop: two consecutive bytes at value 0xFF.
   - Measurement data: All other data

Metadata: Error

An error message (voltage drop) can be sent as a stream of ASCII characters.

Table 38. Metadata error

Byte on serial port	Byte number	Description
0xF0	1	Metadata beginning tag
0xF1	2	Metadata ASCII error message tag
ASCII char	3	Message content: ASCII character
ASCII char	...	Message content: ASCII character

Table 38. Metadata error (continued)

Byte on serial port	Byte number	Description
ASCII char	x	Message content: ASCII character
ASCII char	x + 1	Message content: ASCII character
‘\r’	x + 2	Message content: ASCII value of carriage return, for indication in case of data stream watched in terminal
‘\n’	x + 3	Message content: ASCII value of line feedback, for indication in case of data stream watched in terminal
0xFF	x + 4	Metadata end tag (1/2)
0xFF	x + 5	Metadata end tag (2/2)

Metadata: Information

Similar to error message, with a different metadata tag.

Table 39. Metadata information

Byte on serial port	Byte number	Description
0xF0	1	Metadata beginning tag
0xF2	2	Metadata ASCII information message tag
ASCII char	3	Message content: ASCII character
ASCII char	...	Message content: ASCII character
ASCII char	x	Message content: ASCII character
ASCII char	x + 1	Message content: ASCII character
‘\r’	x + 2	Message content: ASCII value of carriage return, for indication in case of data stream watched in terminal
‘\n’	x + 3	Message content: ASCII value of line feedback, for indication in case of data stream watched in terminal
0xFF	x + 4	Metadata end tag (1/2)
0xFF	x + 5	Metadata end tag (2/2)

Metadata: Timestamp

Each 1000 samples, a timestamp is sent.

It can be used to check data count matches with each timestamp occurrence and resynchronize timing in case of some data have been lost.

Timestamp also includes information of time elapsed in milliseconds.

Additionally, the information of PowerShield buffer Tx load is added after time elapsed.

It is useful in case of host encounters a delay on USB bus (can occur when host is busy by CPU load or file access delay), to take appropriate action before buffer overload and acquisition stop.

Timestamp format: Binary, nine bytes

Table 40. Metadata timestamp

Byte on serial port	Byte number	Description
0xF0	1	Metadata beginning tag
0xF3	2	Metadata timestamp message tag
x	3	Timestamp value in ms, byte 3
x	4	Timestamp value in ms, byte 2
x	5	Timestamp value in ms, byte 1
x	6	Timestamp value in ms, byte 0
x	7	Buffer Tx load value in percent, byte 0
0xFF	8	Metadata end tag (1/2)
0xFF	9	Metadata end tag (2/2)

Note: Timestamp value is coded on four bytes:
- Timing value is coded on 31 bits (value from 0 to 2147483647: allowing unique timestamps up to two million of seconds, equivalent to 23 days)
- Bit 32 is used to indicate counter overflow. In case of overflow, bit 32 is set and timing value is restarting from zero.

Metadata: End of acquisition

Metadata send when acquisition is completed: integration time reached or command “stop” sent by user, and all data in Tx buffer sent to host.

Metadata format: Binary, four bytes

Table 41. Metadata end of acquisition

Byte on serial port	Byte number	Description
0xF0	1	Metadata beginning tag
0xF4	2	Metadata end of acquisition tag
0xFF	3	Metadata end tag (1/2)
0xFF	4	Metadata end tag (2/2)

Metadata: Overcurrent

Metadata send when current sinked by target device exceeds board maximum suply capacity

Metadata format: Binary, four bytes

Table 42. Metadata overcurrent

Byte on serial port	Byte number	Description
0xF0	1	Metadata beginning tag
0xF4	2	Metadata overcurrent tag
0xFF	3	Metadata end tag (1/2)
0xFF	4	Metadata end tag (2/2)

Metadata: Acknowledge and data of command “target reset (target power down)”

Metadata send (in format binary) after command from host is received (in format ASCII), only under conditions:

– Acquisition is ongoing

Metadata format: Binary, four bytes

Table 43. Metadata target power down

Byte on serial port	Byte number	Description
0xF0	1	Metadata beginning tag
0xF6	2	Metadata target power down tag
0xFF	3	Metadata end tag (1/2)
0xFF	4	Metadata end tag (2/2)

Metadata: Acknowledge and data of command “Voltage get”

Metadata send (in binary format) after command from host is received (in ASCII format), only under conditions:

– Acquisition is ongoing

Metadata format: Binary, six bytes

Voltage is coded on two bytes on format unsigned:

0x0CE4 ⇔ 3300 mV

0x0708 ⇔ 1800 mV

Table 44. Metadada Acknowledge and data command “voltage get”

Byte on serial port	Byte number	Description
0xF0	1	Metadata beginning tag
0xF7	2	Metadata voltage tag
x	3	Voltage value in mV, byte 1
x	4	Voltage value in mV, byte 0
0xFF	5	Metadata end tag (1/2)
0xFF	6	Metadata end tag (2/2)

Metadata: Acknowledge and data of command “Temperature”

Metadata send (in binary format) after command from host is received (in ASCII format),

only under conditions:

– Acquisition is ongoing

Metadata format: Binary, six bytes

Temperature is coded on two bytes on format signed:

0x000A ⇔ +10 degC

0xFFFD ⇔ -3 degC

Table 45. Metadata temperature

Byte on serial port	Byte number	Description
0xF0	1	Metadata beginning tag
0xF8	2	Metadata temperature tag
x	3	Temperature value in degC, byte 1
x	4	Temperature value in degC, byte 0
0xFF	5	Metadata end tag (1/2)
0xFF	6	Metadata end tag (2/2)

Metadata: Acknowledge and data of command “pwr get”

Metadata send (in format binary) after command from host is received (in format ASCII),

only under conditions:

- Acquisition is on going
- Acquisition is armed and not started (case of trigger D7 selected, command “start” sent and signal event of connector D7 not yet occurred).

Metadata also sent at the beginning (after metadata of acquisition start) and end (before metadata of acquisition end) of each acquisition if second parameter of command “pwr” is set to argument “status” (optional).

Metadata format: Binary, 5 bytes

Power to target connection status is coded on 1 byte:

0x0 <=> power off (power supply disconnected from target)

0x1 <=> power on (power supply connected to target)

Table 46. Metadata power to target connection

Byte on serial port	Byte number	Description
0xF0	1	Metadata beginning tag
0xF9	2	Metadata power to target connection tag
x	3	Power to target connection, byte 1

Table 46. Metadata power to target connection (continued)

Byte on serial port	Byte number	Description
0xFF	4	Metadata end tag (1/2)
0xFF	5	Metadata end tag (2/2)

Metadata reserved

Metadata reserved for potential future usage:

{0xF0; 0xF5}

{0xF0; 0xFE}

Example of data stream

Measurement with 150 data.

Table 47. Example of data stream in format 2: Binary, hexadecimal basis

Byte on serial port	Byte number	Description
0xF0	1	Metadata: Timestamp 0 ms
0xF3	2	
0x00	3	
0x00	4	
0x00	5	
0x00	6	
0x00	7	
0xFF	8	
0xFF	9	
0x3A	10	Measurement data 1 (1/2)
0xB8	11	Measurement data 1 (2/2)
0x3C	12	Measurement data 2 (1/2)
0x86	13	Measurement data 2 (2/2)
...	...	...
...	...	...
0x47	208	Measurement data 99 (1/2)
0x16	209	Measurement data 99 (2/2)

Table 47. Example of data stream in format 2: Binary, hexadecimal basis (continued)

Byte on serial port	Byte number	Description
0xF0	210	Metadata: Timestamp 0 ms
0xF1	211	
0x00	212	
0x00	213	
0x00	214	
0x01	215	
0xFF	216	
0xFF	217	
0x5E	218	Measurement data 100 (1/2)
0xB1	219	Measurement data 100 (2/2)
0x5C	220	Measurement data 101 (1/2)
0x27	221	Measurement data 101(2/2)
...	...	...
...	...	...
0x47	-	Measurement data 150 (1/2)
0x16	-	Measurement data 150 (2/2)
0xF0	-	Metadata: end of acquisition
0xF4	-	
0xFF	-	
0xFF	-	
0xF0	-	Data post-acquisition: summary
...	-	
...	-	
...	-	
...	-	
...	-	
...	-	
...	-	

4.5 Examples of typical use cases

4.5.1 Minimal mandatory commands

The not used commands are implicitly used with default settings.

Table 48. Minimal mandatory commands

Data sent by host	Data sent by PowerShield
(Not displayed)	COM port terminal
htc	
start	*PowerShield > ack htc* *PowerShield > ack start*
(delay)	*1958-09* *2041-09* *...* *1853-09* *1742-09* *end*
hrc	*PowerShield > ack hrc*

4.5.2 Continuous measurement (infinite samples) with target reset during acquisition

Table 49. Continuous measurement

Data sent by host	Data sent by PowerShield
(Not displayed)	COM port terminal
htc	
volt 3300 m (or “volt 3300-3”) *freq 1 k (or “freq 1000", or “freq 1+3”)* *acqtime inf (or “acqtime 0")*	*PowerShield > ack htc* *PowerShield > ack volt 3300 m* *PowerShield > ack freq 1 k* *PowerShield > ack acqtime inf*
start	*PowerShield > ack start*
(delay)	*11958-09* *2041-09* *...* *1853-09*
targrst 100 m	*PowerShield > ack targrst 100 m*
(delay)	*0023-10* *0008-10* *...* *1742-09* *2013-09* *end*
hrc	*PowerShield > ack hrc*

4.5.3 Two single measurements of 100 samples with power-down of board under test at the end of acquisition

Table 50. Two single measurements of 100 samples

Data sent by host	Data sent by PowerShield
(Not displayed)	COM port terminal
htc	
volt 3300 m	*PowerShield > ack htc* *PowerShield > ack volt 3300 m*
freq 1 k	*PowerShield > ack freq 1 k*
acqtime 100 m	*PowerShield > ack acqtime 100 m*
pwrend off	*PowerShield > pwrend off*
start	*PowerShield > ack start*
(delay)	*1958-09* *2041-09* *...* *1853-09* *end*
start	*PowerShield > ack start*
(delay)	*1958-09* *2041-09* *...* *1853-09* *end*
hrc	*PowerShield > ack hrc*