



Firmware

Modified: August 23, 2019
Version: v0.80

Protocol Commands

The listed commands can be used to build measurement protocols. Some commands require a specific firmware version or Instrument. Make sure to check, if the commands that being used, are compatible.

Available Commands

`_protocol_set_`

Treating protocols as integrated sets.

Protocols are separate units which produce separate data sets. Using `_protocol_set_`, multiple protocols can be chained together into one protocol.

Note: This is different from chaining protocols together in a project. There, the protocols are still separate units within a measurement.

Input: [array](#)

Example:

```
1  "_protocol_set_": [  
2    { Protocol 1 },  
3    ...  
4  ],
```

Instruments:

- MultispeQ 1: `2.0038` `2.0036` `2.0035`
- MultispeQ 2: `2.0038` `2.0036` `2.0035`

Last Updated: July 1, 2019

autogain

The `autogain` function is added to the main section of the protocol. Multiple `autogain` functions can be run in a single protocol, and the indexed values should be useable throughout the experiment.

`<index>` which index to store gain settings (0-9)
`<pulsed_LED>` which pulsed LED to test (0-9)
`<detector>` which detector to use (0-3)

`<pulse_duration>` what duration to use (in microseconds) (1-200)
`<target_value>` what target value to use (0-65535, recommended: 40000-50000)

Input: nested array

Example:

```
1 "autogain": [  
2   [<index>, <pulsed_LED>, <detector>, <pulse_duration>, <target_value>],  
3   ...  
4 ]
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

averages

`averages` sets the number of times to average the protocol. The protocol will be repeated equal to the number of averages, and the resulting data will be averaged and outputted as a single data point. Averaging is often used to reduce noise and improve the quality of measurements.

Input: [number](#)

Values:

- 0 - 10000

Example:

```
1 "averages": <number>
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

averages_delay

`averages_delay` defines the delay between protocol averages in in milliseconds (ms).

Input: auto

Values:

- 0 - 9999999999

Example:

```
1 "averages_delay": <number>,
```

Dependencies:

- averages

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

dac_lights

When `dac_lights` is set to 1, `pulsed_lights_brightness` and `nonpulsed_lights_brightness` settings will be interpreted by the device as 12-bit values (0 - 4095) directly controlling the LED voltage (via the DAC), instead of microEinsteins ($\mu\text{E} \cdot \text{s}^{-1} \cdot \text{m}^2$). This is used during factory calibration.

Note: Setting brightness to high using raw DAC values could permanently disable LEDs, so be careful!

Input: [number](#)

Values:

- 0, 1

Example:

```
1 "dac_lights": <number>,
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

detectors

`detectors` defines which detectors are being measured, and in what order. The chosen `meas_light` supplies the light source to be measured. The detector measurement is recorded in "data_raw" in the data received from the MultispeQ, which is then graphed for the user. When `detectors` is set to 0, no detector is read and responses will be recorded as zero.

Input: nested array

Values:

- 0 - 4

Example:

```
1 "detectors": [  
2   [ <detector> ],  
3   ...  
4 ]
```

Dependencies:

- pulses
- pulse_length
- pulse_distance
- pulsed_lights
- pulsed_lights_brightness

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

energy_save_timeout

Adjusting Energy Save Timeout Time. The value of time is in milliseconds, and must be between zero and 10^6 milliseconds. This setting changes the `energy_save_timeout` time to prevent interference from the requirement for a 7 second wake up time from power save mode.

Important: Note that the change in `energy_save_timeout` will remain in effect until the instrument is reset, when it will return to the default value of 120 s.

Input: [number](#)

Values:

- 0 - 10⁶

Example:

```
1 "energy_save_timeout": <number>,
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

environmental

`environmental` defines which additional sensor(s) to measure. The environmental calls occur at the very beginning of the measurement, prior to any pulses. Some sensors require additional information, like specifying which pin to measure, brightness, etc. In addition to sensor measurements, there are also calls to flip digital pins on/off or set a pwm which could be used to control external lights or even motors. Below is the full list of parameters available in `environmental`.

. Adjusting protocols in real time based on these values can be very handy for certain types of protocols.

Input: nested array

Example:

```
1 "environmental": [  
2   [ <sensor> ],  
3   ...  
4 ],
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: August 9, 2019

environmental_array

`environmental_array` defines additional sensors or functionality, just like `environmental`. However, `environmental_array` returns sensor data once per pulse, as defined in `pulses`. This allows a string of sensor measurements to be used in a single macro. The advantage is you can calculate and return in real time things like rates of change, minima and maxima, etc. See examples for details. **Note:** `pulsed_lights_brightness` must NOT be set to zero for any given pulse set for the sensors set in `environmental_array` to measure.

Input: nested array

Example:

```
1 "environmental_array": [  
2   [ <number> ],  
3   ...  
4 ],
```

Dependencies:

- pulses
- pulse_length
- pulse_distance
- pulsed_lights
- pulsed_lights_brightness

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

ir_baseline

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

label

Each protocol or sub-protocol can now output a `label` that can be used to indicate something about the protocol, conditions etc.

Input: [string](#)

Example:

```
1 "label": <string>,
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

max_hold_time

This parameter sets the time (in ms) at which the hold commands timeout. The default value is 15000, or 15 seconds. This value applies to the following control points:

`start_on_open`, `start_on_close`, `start_on_open_close`.

Input: [number](#)

Example:

```
1 "max_hold_time": <number>,
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

measurements

number of times to repeat a measurement, which is a set of protocols

Example:

```
1 "measurements": <number>
```

Last Updated: July 1, 2019

measurements_delay

delay between measurements in milliseconds

Example:

```
1 "measurements_delay": <number>
```

Last Updated: July 1, 2019

message

`message` sends text to the user between pulse sets, and waits for a response before proceeding. Set `<message type>` to "0" to not send a message. `message` should have the same length as `pulses`.

Input: nested array

Values:

- `alert` - message with an OK button + `prompt` - message with an OK and Cancel button + `confirm` - message with an OK and Cancel button and text input box

Example:

```
1 "message": [  
2   [ <message type>, <message> ],  
3   ...  
4 ],
```

Dependencies:

- `pulses`
- `pulse_length`
- `pulse_distance`

Instruments:

- MultispeQ 1: `2.0038` `2.0036` `2.0035` `1.17` `1.16` `1.14` `1.11` `1.10` `1.08` `1.07` `1.06`
- MultispeQ 2: `2.0038` `2.0036` `2.0035`

Last Updated: July 1, 2019

nonpulsed_lights

`nonpulsed_lights` defines which lights are not pulsed (always on) during a pulse set, and in what order.

Input: nested array

Values:

- 0 - 10

Example:

```
1 "nonpulsed_lights": [  
2   [ <number> ],  
3   ...  
4 ],
```

Dependencies:

- pulses
- pulse_distance
- pulse_length
- nonpulsed_lights_brightness

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

nonpulsed_lights_brightness

`nonpulsed_lights_brightness` sets the brightness in microEinsteins ($\mu\text{E} \cdot \text{s}^{-1} \cdot \text{m}^2$) of the nonpulsed lights.

Input: nested array

Values:

- 0 - 15000

Example:

```
1 "nonpulsed_lights_brightness": [  
2   [ <number> ],  
3   ...  
4 ],
```

Dependencies:

- pulses
- pulse_distance

- pulse_length
- nonpulsed_lights

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

number_samples

`number_samples` sets the number of samples taken by the ADC (analog to digital converter) on the sample and hold circuit. The median value of these samples is then saved as a single detector value in "data_raw". The ADC is used to measure the detectors (1 - 4). This value is already set to the optimum value, and most signal noise comes from other source, but in certain cases increasing the value may yield improved signal.

Input: [number](#)

Values:

- 1 - 500

Example:

```
1 "number_samples": <number>,
```

Dependencies:

- pulses
- pulse_length
- pulse_distance
- detectors

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

open_close_start

When `open_close_start` is set to 1, the protocol waits until the user fully opens the clamp and closes the clamp before proceeding. Also works if the clamp starts fully open and then closes. Open and close is detected using the Hall effect sensor on the main body and magnet on the clamp (same sensor used to determine sample thickness). If the device is not calibrated, or calibration is off, or the clamping mechanism does not fully open or close, this function will not work. Calibration details can be found by typing `print_memory` into the console - see `thickness_a`, `thickness_b`, `thickness_c`, `thickness_min`, and `thickness_max` for relevant calibration values.

Input: [number](#)

Values:

- 0, 1

Example:

```
1 "open_close_start": <number>,
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

par_led_start_on_close

The Ambient light is recreated inside the instrument on closing the clamp. During the hold time, the ambient light is resampled in ~100 ms intervals and the LED output is adjusted to match. When the hold event occurs (e.g. the clamp is closed), the final ambient intensity and LED setting are held and used in the protocol. The number for the LED is defining the LED used for matching the ambient light intensity.

Input: [number](#)

Example:

```
1 "par_led_start_on_close": <number for LED>,
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

par_led_start_on_open

The Ambient light is recreated inside the instrument on opening the clamp. During the hold time, the ambient light is resampled in ~100 ms intervals and the LED output is adjusted to match. When the hold event occurs (e.g. the clamp is closed), the final ambient intensity and LED setting are held and used in the protocol. The number for the LED is defining the LED used for matching the ambient light intensity.

Input: [number](#)

Example:

```
1 "par_led_start_on_open": <number for LED>,
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

par_led_start_on_open_close

The Ambient light is recreated inside the instrument on opening the clamp. During the hold time, the ambient light is resampled in ~100 ms intervals and the LED output is adjusted to match. When the hold event occurs (e.g. the clamp is closed), the final ambient intensity and LED setting are held and used in the protocol. The number for the LED is defining the LED used for matching the ambient light intensity.

Input: [number](#)

Example:

```
1 "par_led_start_on_open_close": <number for LED>,
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

protocols

`protocols` sets the number of times to repeat the protocol. Unlike `averages`, this outputs data every time the protocol is repeated (instead of averaging the repeats and outputting data only once).

Input: [number](#)

Values:

- 0 - 9999999999

Example:

```
1 "protocols": <number>,
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

protocols_delay

`protocols_delay` defines the delay between protocol repeats in in milliseconds (ms).

Input: auto

Values:

- 0 - 99999999999

Example:

```
1 "protocols_delay": <number>,
```

Dependencies:

- protocols

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

pulse_distance

`pulse_distance` defines the distance, in microseconds (μs) between pulses.

Input: [array](#)

Values:

- 750 - 999999999999

Example:

```
1 "pulse_distance": [  
2   <number>,  
3   ...  
4 ],
```

Dependencies:

- pulses
- pulse_length
- pulsed_lights
- pulsed_lights_brightness

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

pulse_length

`pulse_length` defines the length, in microseconds (μs), of a pulse.

Input: nested array

Values:

- 1 - 150

Example:

```
1 "pulse_length": [  
2   [ <number> ],  
3   ...  
4 ],
```

Dependencies:

- pulses
- pulse_distance
- pulsed_lights
- pulsed_lights_brightness

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

pulsed_lights

`pulsed_lights` defines which lights are pulsed, and in what order. When set to 0, no lights are pulsed and no readings are recorded (see Example section here for details).

Input: nested array

Values:

- 0 - 10

Example:

```

1 "pulsed_lights": [
2   [ <number> ],
3   ...
4 ]
```

Dependencies:

- pulses
- pulse_distance
- pulse_length
- pulsed_lights_brightness

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

pulsed_lights_brightness

`pulsed_lights_brightness` sets the brightness in microEinsteins ($\mu\text{E} \cdot \text{s}^{-1} \cdot \text{m}^2$) of the pulsed lights.

Input: nested array

Values:

- 0 - 15000

Example:

```
1 "pulsed_lights_brightness": [  
2   [ <number> ],  
3   ...  
4 ]
```

Dependencies:

- pulses
- pulse_distance
- pulse_length
- pulsed_lights

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

pulses

`pulses` defines the number of pulse sets, and quantity of pulses per set.

Input: [array](#)

Values:

- 1 - 8000

Example:

```
1 "pulses": [  
2   <number>,  
3   ...  
4 ]
```

Dependencies:

- pulse_distance

- pulse_length
- pulsed_lights
- pulsed_lights_brightness

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

recall

This command returns values from the device memory (EEPROM). This includes values saved by the user (see Examples below for details) as well as values saved during factory calibration.

Input: [array](#)

Example:

```

1 "recall": [
2   "userdef[<location>]",
3   ...
4 ],
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: August 9, 2019

reference

The MultispeQ is designed to be able to nearly simultaneously measure two detectors (within $\sim 4\mu\text{s}$), normalize and subtract the two signals from each other. This can be useful if there is an optical, electronic (interference), or LED (heating) artifact that needs to be removed. `reference` specifies another detector to be measured and subtracted from `detectors`.

Note: It is possible to add two additional detector circuits to the device, on the main body and clamp circuit boards. The devices come unpopulated by default. If added, these two additional detectors are available by settings `detectors` or `reference` to 2 (main body) or 4 (clamp). These can then also be used as reference or main detectors.

Input: nested array

Values:

- 1 - 4

Example:

```
1 "reference": [  
2   [ <detector> ],  
3   ...  
4 ],
```

Dependencies:

- pulses
- pulse_length
- pulse_distance

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

save

This command saves a value to a location in the device memory (EEPROM).

Input: nested array

Example:

```
1 "save": [  
2   [ <location, <value> ],  
3   ...  
4 ],
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: August 9, 2019

save_trace_time_scale

Saving Time Scale Values. Setting the value to 0 will inactivate the function (preset), setting it to 1 will activate the function. If activated `data_raw_time` will be added to the measurement, containing the timing information.

Input: [number](#)

Values:

- 0 + 1

Example:

```
1 "save_trace_time_scale": <number>,
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_led_delay

Pre-illuminating a sample. It is often useful to pre-illuminate a sample at a given light intensity, for a given amount of time.

Input: nested array

Example:

```
1 "set_led_delay": [  
2   [ <number LED>, <number duration in ms>, <number PAR> ],  
3   ...  
4 ],  
5  
6 // The following example gives two,  
7 // 20 second pre-illuminations with the red LED (#2),  
8 // the first at 0  $\mu\text{E} * \text{s}^{-1} * \text{m}^{-2}$  and the second at 100  $\mu\text{E} * \text{s}^{-1} * \text{m}^{-2}$ :  
9  
10 "set_led_delay": [  
11   [2, 20000,0],  
12   [2, 20000,100]  
13 ],
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_light_intensity

Instead of measuring the ambient light intensity, use `set_light_intensity` to set a fixed light intensity. Use the parameters `light_intensity` or `previous_light_intensity` to use the defined light intensity.

Note: Instead of defining the light intensity for each pulse set use `set_light_intensity` to only have one place to change the light intensity.

Input: [number](#)

Example:

```
1 "set_light_intensity": <number>,
```

Instruments:

- MultispeQ 1: `2.0038` `2.0036` `2.0035`
- MultispeQ 2: `2.0038` `2.0036` `2.0035`

Last Updated: July 1, 2019

start_on_close

If set to 0, the command will be ignored. If set to 1, will wait until the clamp is closed, then proceed with the rest of the experiment.

Input: [number](#)

Values:

- 0 + 1

Example:

```
1 "start_on_close": <number>,
```

Instruments:

- MultispeQ 1: `2.0038` `2.0036` `2.0035`
- MultispeQ 2: `2.0038` `2.0036` `2.0035`

Last Updated: July 1, 2019

start_on_open

If set to 0, the command will be ignored. If set to 1 or higher, it will wait until the clamp is opened, then proceed with the rest of the experiment.

Input: [number](#)

Values:

- 0 + 1

Example:

```
1 "start_on_open": <number>,
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

start_on_open_close

If set to 0, the command will be ignored. If set to 1, will wait until the clamp is opened then closed, then proceed with the rest of the experiment.

Alias: `open_close_start`

Example:

```
1 "start_on_open_close": <number>,
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

adc_show deprecated

When "adc_show" is 1, the readings taken by the ADC (analog to digital converter) on the sample and hold circuit are outputted in "data_raw" instead of the normal output. No other output is recorded, and only the last set of adc readings is shown (all other readings are not

outputted). By default there are 19 samples, but this can be changed using "number_samples".

Input: [number](#)

Values:

- 0, 1

Example:

```
1 "adc_show": <number>,
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

energy_min_wake_time deprecated

The value of time is in milliseconds, and must be between zero and 10^6 milliseconds. This setting changes the delay between shutting off the 5V (following `energy_save_timeout`) and the wake up time. The `energy_min_wake_time` is needed to prevent brown out following the initiation of energy save.

Important: Note that the change in `energy_min_wake_time` will remain in effect until the instrument is reset, when it will return to the default value of 10 s.

Input: [number](#)

Values:

- 0 - 10^6

Example:

```
1 "energy_min_wake_time": <number>,
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

Console Commands

Using the Console in the Desktop application, you can use these commands to directly communicate with the Instrument. It allows to change settings and display informations, which are mostly only for advanced users or developers. When working with the console and these commands, be careful!

Note

Some of the commands will introduce changes without warning, which could cause the Instrument to malfunction.

Command Switches

Silent: `s+` Sets command console in silent mode. Commands will not send back queries for user input. `s+` will remain in effect until reset or user enters `v+`.

Verbose: `v+` Sets command console in verbose mode. When available, commands will send back queries for user input. `v+` will remain in effect until reset or user enters `s+`.

Available Commands

1053

Continuous feed of roll, pitch, compass, direction, tilt and tilt direction. Cancel the command with `-1+`

Example:

```
1 1053
```

Instruments:

- MultispeQ 1: `2.0038` `2.0036` `2.0035` `1.17` `1.16` `1.14` `1.11` `1.10` `1.08` `1.07` `1.06`
- MultispeQ 2: `2.0038` `2.0036` `2.0035`

Last Updated: July 1, 2019

any_light

Example:

1 any_light

Instruments:

- MultispeQ 1: 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: not available

Last Updated: July 1, 2019

battery

Test the instruments battery with load returning the charge state in percent (%)

Example:

1 battery

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

compiled

Returns date and time for when the Firmware was compiled.

Example:

1 compiled

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

configure_bluetooth

Set the bluetooth name and baud rate

Example:

```
1 configure_bluetooth
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

device_info

Receive information from the Instrument, including name, version, id, battery level, firmware and configuration.

Alias: 1007

Example:

```
1 device_info
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

digital_write

Asks for address location and output value (0 or 1) and sets the digital address to this value. This is a dangerous if you do not know what you are doing.

Input: [number](#)

Values:

- 0 + 1

Example:

```
1 digital_write
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

expr

Example:

```
1 expr
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

flow_calibration_set_point

Alias: fcsp

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

flow_calibration_setting

Alias: fcv

Example:

```
1 flow_calibration_setting
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

flow_calibration_value

Alias: fcv

Example:

```
1 flow_calibration_value
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

flow_off

Set air flow to zero

Example:

```
1 flow_off
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

flow_v

Example:

```
1 flow_v
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

get_flow

Example:

```
1 get_flow
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

hall

Continues output of the hall sensor readings. Stop output using the command `-1+`

Example:

```
1 hall
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

hello

Check if your instrument is connected. On success the response is `Instrument Ready`.

Alias: `1000`

Example:

```
1 hello
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

indicate

Sets the indicator LED (RGB LED). Values in the range from 0-255 need to be set for the Red, Green and Blue channel.

Input: [number](#)

Example:

```
1 indicate+<Red>+<Green>+<Blue>
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

indicate_off

Turn off indicator LED

Example:

```
1 indicate_off
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

memory

Returns the memory usage of the instrument.

Example:

1 memory

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

p2p

Read multiple pulses with increasing intensity or pulse width for linearity test. With constant DAC value and pulse width, it is good for a pulse-to-pulse stdev test

Example:

1 p2p

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

par_led

Input: par_led

Example:

1 par_led

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

print_date

Print date based on instruments RTC (if available).

e.g. 2004-02-12T15:19:21.000Z

Example:

```
1 print_date
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

print_magnetometer

Example:

```
1 print_magnetometer
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

print_magnetometer_bias

Example:

```
1 print_magnetometer_bias
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

print_memory

Print all calibration values as they are saved to the instruments storage (EEPROM). Output is formatted as a JSON with a checksum.

Alias: `print_calibrations`

Example:

```
1 print_memory
```

Instruments:

- MultispeQ 1: `2.0038` `2.0036` `2.0035` `1.17` `1.16` `1.14` `1.11` `1.10` `1.08` `1.07` `1.06`
- MultispeQ 2: `2.0038` `2.0036` `2.0035`

Last Updated: July 1, 2019

readonce

Access write once flash

Example:

```
1 readonce
```

Instruments:

- MultispeQ 1: `2.0038` `2.0036` `2.0035` `1.17` `1.16` `1.14` `1.11` `1.10` `1.08` `1.07` `1.06`
- MultispeQ 2: `2.0038` `2.0036` `2.0035`

Last Updated: July 1, 2019

reboot

Example:

```
1 reboot
```

Instruments:

- MultispeQ 1: `2.0038` `2.0036` `2.0035` `1.17` `1.16` `1.14` `1.11` `1.10` `1.08` `1.07` `1.06`
- MultispeQ 2: `2.0038` `2.0036` `2.0035`

Last Updated: July 1, 2019

reset

Reboot the instrument.

Alias: 1027

Example:

```
1 reset
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

reset_flow_calibration

Set air flow calibration to factory settings

Example:

```
1 reset_flow_calibration
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

reset_flow_zero_point

Example:

```
1 reset_flow_zero_point
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

scan_i2c

Scan for available i2c devices.

Example:

```
1 scan_i2c
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_accelerometer

Example:

```
1 set_accelerometer
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_accelerometer_bias

Example:

```
1 set_accelerometer_bias
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_colorcal1

Example:

```
1 set_colorcal1
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_colorcal2

Example:

```
1 set_colorcal2
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_colorcal3

Example:

```
1 set_colorcal3
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_colorcal_blanks

Example:

```
1 set_colorcal_blanks
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_cp

Set the **close** position of the leave clamp.

Example:

```
1 set_cp+
```

Instruments:

- MultispeQ 1: 2.0038
- MultispeQ 2: 2.0038

Last Updated: July 1, 2019

set_dac

Set DAC addresses to 1,2,3 assuming addresses are unset and all are factory (0,0,0).

Example:

```
1 set_dac
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_date

Set the date of the RTC of the instrument (if available)

Example:

```
1 set_date+
2 hours+min+sec+day+month+year
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_default_flow_rate

Input: [number](#)

Example:

```
1 set_default_flow_rate+<number>
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_detector1_offset

Example:

```
1 set_detector1_offset
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_detector2_offset

Example:

```
1 set_detector2_offset
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_detector3_offset

Example:

```
1 set_detector3_offset
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_detector4_offset

Example:

```
1 set_detector4_offset
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_device_info

Set the device name

Alias: 1008

Example:

```
1 set_device_info
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_energy_save_time

Input: [number](#)

Example:

```
1 set_energy_save_time+<number>
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_flow

Input: [number](#)

Example:

```
1 set_flow
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_led_par

Example:

```
1 set_led_par
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_magnetometer

Example:

```
1 set_magnetometer
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_magnetometer_bias

Example:

```
1 set_magnetometer_bias
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_op

Set the **open** position of the leave clamp.

Example:

```
1 set_op+
```

Instruments:

- MultispeQ 1: 2.0038
- MultispeQ 2: 2.0038

Last Updated: July 1, 2019

set_open_closed_positions

Input: set_open_closed_positions

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_par

Example:

```
1 set_par
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_serial

Example:

```
1 set_serial
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_shutdown_time

Sets the time (in seconds) before the device automatically shuts down to save energy.

Input: [number](#)

Example:

```
1 set_shutdown_time+<seconds>
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_thickness

Example:

```
1 set_thickness
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_thickness_quick

Example:

```
1 set_thickness_quick
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

set_user_defined

Save an user defined value to the EEPROM. You can abort the input using `-1+`.

Example:

```
1 set_user_defined+<EEPROM Location>+<value>
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

single_pulse

Read and analyze noise on ADC from a single LED pulse (only in testmode).

Example:

```
1 single_pulse
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

sleep

Puts instrument into sleep mode. Hold the button for at least 5s to wake it up.

Example:

```
1 sleep
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

start_watchdog

Example:

```
1 start_watchdog
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

stop_watchdog

Example:

```
1 stop_watchdog
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

tcs_length

Example:

```
1 tcs_length
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

temp

Return the readings for both BME280 sensors including temperature and relative humidity.

Example:

```
1 temp
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

testmode

Example:

```
1 testmode
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

upgrade

Start over the air firmware update. The instrument is waiting for the new firmware to be transferred to the instrument.

Alias: 1078

Example:

1 upgrade

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

usb

Check, if the Instrument is connected via USB

Example:

1 usb

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

1054 deprecated

Example:

1 1054

Instruments:

- MultispeQ 1: 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: not available

Last Updated: July 1, 2019

4048 deprecated

Read multiple pulses with increasing intensity or pulse width for linearity test. With constant DAC value and pulse width, it is good for a pulse-to-pulse stdev test.

Example:

1 4048

Instruments:

- MultispeQ 1: 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: not available

Last Updated: July 1, 2019

adc1 **deprecated**

Example:

1 adc1

Instruments:

- MultispeQ 1: 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: not available

Last Updated: July 1, 2019

adc_check **deprecated**

Output all ADC values

Example:

1 adc_check

Instruments:

- MultispeQ 1: 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: not available

Last Updated: July 1, 2019

all_sensors **deprecated**

Continuously all sensor output including par, temp, rH and pressure, until user enter **-1+**

Example:

```
1 all_sensors
```

Instruments:

- MultispeQ 1: 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: not available

Last Updated: July 1, 2019

calibrate_compass **deprecated**

Example:

```
1 calibrate_compass
```

Instruments:

- MultispeQ 1: 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: not available

Last Updated: July 1, 2019

calibrate_leds **deprecated**

Example:

```
1 calibrate_leds
```

Instruments:

- MultispeQ 1: 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: not available

Last Updated: July 1, 2019

calibrate_leds_manual **deprecated**

Example:

```
1 calibrate_leds_manual
```

Instruments:

- MultispeQ 1: 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: not available

Last Updated: July 1, 2019

constant_light **deprecated**

Starting constant light source. Stop output using the command `-1+`.

Example:

```
1 constant_light
```

Instruments:

- MultispeQ 1: 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: not available

Last Updated: July 1, 2019

cut_through **deprecated**

Example:

```
1 cut_through
```

Instruments:

- MultispeQ 1: 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: not available

Last Updated: July 1, 2019

cycle5v **deprecated**

Example:

```
1 cycle5v
```

Instruments:

- MultispeQ 1: 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: not available

Last Updated: July 1, 2019

dac50 deprecated

Set all DAC outputs to 50%

Example:

```
1 dac50
```

Instruments:

- MultispeQ 1: 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: not available

Last Updated: July 1, 2019

feed_watchdog deprecated

Example:

```
1 feed_watchdog
```

Instruments:

- MultispeQ 1: 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: not available

Last Updated: July 1, 2019

get_co2 deprecated

Sensair S8 CO2 requests. Only works if you have connected the sensair on Serial Port 3

Example:

```
1 get_co2
```

Instruments:

- MultispeQ 1: 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: not available

Last Updated: July 1, 2019

light **deprecated**

Turn on 5V to turn on a light. Replace the <light number> with the number of a light, e.g. `light1`

Example:

```
1 light<light number>
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

on_5v **deprecated**

Turn on 5V for 30 seconds

Example:

```
1 on_5v
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

print_all **deprecated**

print everything in the eeprom (all values defined in eeprom.h)

Example:

```
1 print_all
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019

pulse deprecated

Example:

```
1 pulse
```

Instruments:

- MultispeQ 1: 2.0038 2.0036 2.0035 1.17 1.16 1.14 1.11 1.10 1.08 1.07 1.06
- MultispeQ 2: 2.0038 2.0036 2.0035

Last Updated: July 1, 2019