



BPS204 Programming Manual

BPS204-Client 1.0

Notice

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Langer EMV-Technik GmbH. The management of Langer EMV-Technik GmbH assumes no liability for damages which may result from the use of this printed information.

1 Introduction	1
2 Flowchart	1
3 File Documentation	2
4 Function Documentation	6
4.1 bps_close()	6
4.2 bps_connect_single()	6
4.3 bps_control()	7
4.4 bps_destroy()	7
4.5 bps_get_connections()	8
4.6 bps_get_delta_trigger_delay_maximum()	8
4.7 bps_get_delta_trigger_delay_minimum()	9
4.8 bps_get_device_id()	9
4.9 bps_get_firmware_id()	10
4.10 bps_get_hardware_id()	10
4.11 bps_get_hv_state()	11
4.12 bps_get_info()	11
4.13 bps_get_main_trigger_delay_maximum()	12
4.14 bps_get_main_trigger_delay_minimum()	12
4.15 bps_get_probe_device_id()	13
4.16 bps_get_probe_firmware_id()	13
4.17 bps_get_probe_serial()	14
4.18 bps_get_probe_state()	14
4.19 bps_get_pulse_frequency()	15
4.20 bps_get_pulse_frequency_maximum()	15
4.21 bps_get_pulse_frequency_minimum()	16
4.22 bps_get_pulse_level()	16
4.23 bps_get_pulse_level_difference_minimum()	17
4.24 bps_get_pulse_level_maximum()	17
4.25 bps_get_pulse_level_minimum()	18
4.26 bps_get_pulse_mode()	18
4.27 bps_get_pulse_polarity()	19
4.28 bps_get_state()	19
4.29 bps_get_states()	21

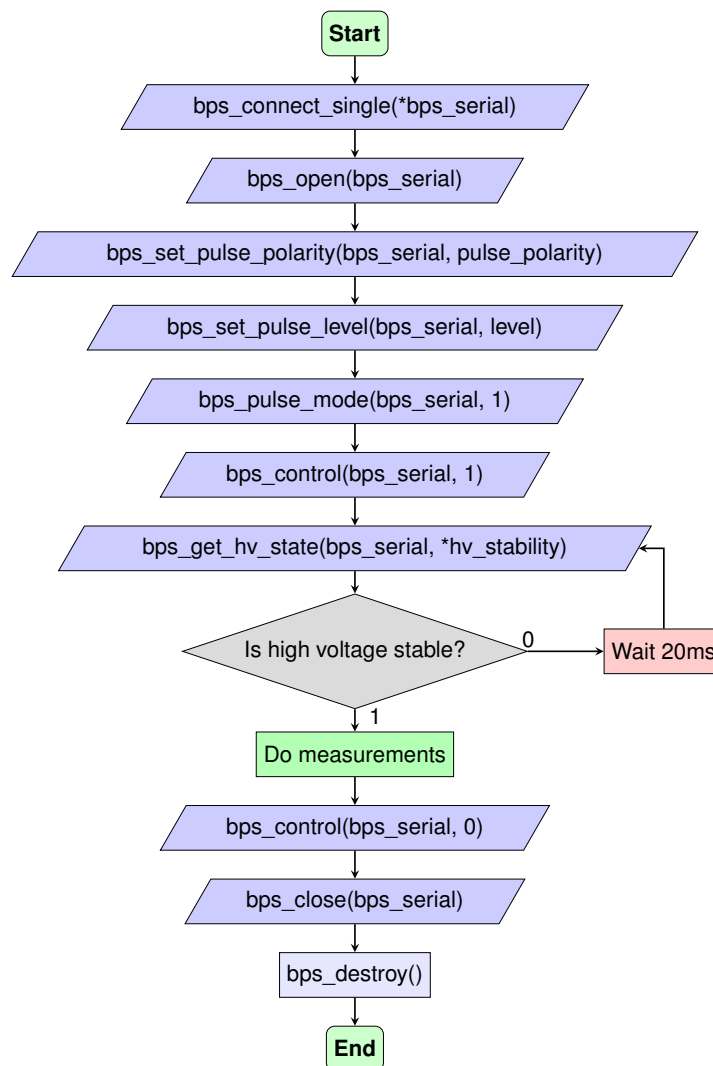
4.30 bps_get_trigger_delay()	21
4.31 bps_get_trigger_mode()	22
4.32 bps_init()	23
4.33 bps_lost_connection()	23
4.34 bps_open()	24
4.35 bps_search()	24
4.36 bps_set_pulse_frequency()	25
4.37 bps_set_pulse_level()	25
4.38 bps_set_pulse_mode()	26
4.39 bps_set_pulse_polarity()	26
4.40 bps_set_reset()	27
4.41 bps_set_trigger_delay()	28
4.42 bps_set_trigger_mode()	28
Index	30

1 Introduction

This document describes the BPS204 remote control commands that are implemented in the provided DLL file.

The BPS204 can either be controlled through the provided GUI application called "BPS204-Client" or by your own application via its USB interface. A variety of remote control commands are available which can be intergrated into your software from the library "bps204.dll". You can thus write your own application to control the BPS204 or integrate the BPS204 into existing applications.

2 Flowchart



3 File Documentation

bps204.dll File Reference

Functions

- DLL uint8_t [bps_connect_single](#) (uint32_t *bps_serial)
Searches for a connected BPS204 and fetch its serial number.
- DLL uint8_t [bps_init](#) ()
Preperes for an initial connection to a BPS204.
- DLL uint8_t [bps_destroy](#) ()
Ends the DLL communication.
- DLL uint8_t [bps_search](#) (uint8_t *new_bps)
Searches for newly connected BPS204 devices.
- DLL uint8_t [bps_get_connections](#) (uint8_t *num_bps)
Gets the number of connected BPS204 devices.
- DLL uint8_t [bps_get_info](#) (uint8_t bps_index, uint32_t *bps_serial)
Gets the serial number from a BPS204.
- DLL uint8_t [bps_open](#) (uint32_t bps_serial)
Opens a connection to a BPS204.
- DLL uint8_t [bps_close](#) (uint32_t bps_serial)
Closes a connection to a BPS204.
- DLL uint8_t [bps_lost_connection](#) (uint32_t bps_serial, uint8_t *lost_connection)
Checks if BPS204 has lost connection.
- DLL uint8_t [bps_get_device_id](#) (uint32_t bps_serial, uint32_t *id)
Gets the device ID of BPS204.
- DLL uint8_t [bps_get_hardware_id](#) (uint32_t bps_serial, uint32_t *hardware_id)
Gets the hardware version of a connected probe.

- DLL uint8_t [bps_get_firmware_id](#) (uint32_t bps_serial, uint32_t *firmware_id)

Gets the firmware ID installed on a connected probe.

- DLL uint8_t [bps_get_probe_serial](#) (uint32_t bps_serial, uint32_t *probe_serial)

***** Gets the serial number from a connected probe.*****

- DLL uint8_t [bps_get_probe_device_id](#) (uint32_t bps_serial, uint32_t *probe_device_id)

Gets the device ID of BPS204.

- DLL uint8_t [bps_get_probe_firmware_id](#) (uint32_t bps_serial, uint32_t *probe_firmware_id)

Gets the firmware ID installed on BPS204.

- DLL uint8_t [bps_set_reset](#) (uint32_t bps_serial)

Resets the BPS204.

- DLL uint8_t [bps_get_state](#) (uint32_t bps_serial, uint8_t *bps_state)

Get state of BPS204.

- DLL uint8_t [bps_get_hv_state](#) (uint32_t bps_serial, uint8_t *hv_stability)

Gets high voltage state of BPS204.

- DLL uint8_t [bps_get_probe_state](#) (uint32_t bps_serial, uint8_t *probe_state)

Gets state of probe connected to BPS204.

- DLL uint8_t [bps_get_states](#) (uint32_t bps_serial, uint8_t *bps_state, uint8_t *hv_stability, uint8_t *probe_state)

Get all 3 states at once.

- DLL uint8_t [bps_get_pulse_level](#) (uint32_t bps_serial, float *level)

Gets the amplitude [V] of the pulse that the BPS204 is set to generate.

- DLL uint8_t [bps_set_pulse_level](#) (uint32_t bps_serial, float level)

Sets the amplitude [V] of the pulse that the BPS204 is set to generate.

- DLL uint8_t [bps_get_pulse_level_minimum](#) (uint32_t bps_serial, float *min_level)

Gets the minimum voltage level [V] that the BPS204 can generate.

- DLL uint8_t [bps_get_pulse_level_maximum](#) (uint32_t bps_serial, float *max_level)

Gets the maximum voltage level [V] that the BPS204 can generate.

- DLL uint8_t [bps_get_pulse_level_difference_minimum](#) (uint32_t bps_serial, float *delta_level)

Gets the minimum voltage level [V] resolution that the BPS204 can provide.

- DLL uint8_t [bps_get_pulse_frequency](#) (uint32_t bps_serial, float *frequency)

Gets the frequency [Hz] at which the BPS204 will generate pulses.

- DLL uint8_t [bps_set_pulse_frequency](#) (uint32_t bps_serial, float frequency)

Sets the frequency [Hz] at which the BPS204 will generate pulses.

- DLL uint8_t [bps_get_pulse_frequency_minimum](#) (uint32_t bps_serial, float *min_frequency)

Gets the lowest frequency [Hz] at which the BPS204 can generate pulses.

- DLL uint8_t [bps_get_pulse_frequency_maximum](#) (uint32_t bps_serial, float *max_frequency)

Gets the highest frequency [Hz] at which the BPS204 can generate pulse.

- DLL uint8_t [bps_set_pulse_mode](#) (uint32_t bps_serial, uint8_t pulse_mode)

Sets the type of pulse that the BPS204 should generate.

- DLL uint8_t [bps_get_pulse_mode](#) (uint32_t bps_serial, uint16_t *pulse_mode)

Gets the type of pulse that the BPS204 will generate.

- DLL uint8_t [bps_get_pulse_polarity](#) (uint32_t bps_serial, uint8_t *pulse_polarity)

Gets the polarity of the pulses that the BPS204 will generate.

- DLL uint8_t [bps_set_pulse_polarity](#) (uint32_t bps_serial, uint8_t pulse_polarity)

Sets the polarity of the pulses that the BPS204 should generate.

- DLL uint8_t [bps_control](#) (uint32_t bps_serial, uint8_t state)

Controls the pulse generation state of the BPS204.

- DLL uint8_t [bps_set_trigger_mode](#) (uint32_t bps_serial, uint8_t trigger_mode)

Sets the type of trigger mode of the BPS204.

- DLL uint8_t [bps_get_trigger_mode](#) (uint32_t bps_serial, uint16_t *trigger_mode)

Gets the trigger mode that the BPS204 is currently set to.

- DLL uint8_t [bps_set_trigger_delay](#) (uint32_t bps_serial, uint8_t trigger_channel, uint16_t trigger_delay)

Sets a trigger delay of BPS204.

- [DLL](#) uint8_t [bps_get_trigger_delay](#) (uint32_t bps_serial, uint8_t trigger_channel, uint16_t *trigger_delay)

Gets a trigger delay that the BPS204 is set to.

- [DLL](#) uint8_t [bps_get_main_trigger_delay_minimum](#) (uint32_t bps_serial, uint16_t *trigger↵_delay_minimum)

Gets the smallest main trigger delay that the BPS204 can provide.

- [DLL](#) uint8_t [bps_get_main_trigger_delay_maximum](#) (uint32_t bps_serial, uint16_t *trigger↵_delay_maximum)

Gets the biggest main trigger delay that the BPS204 can provide.

- [DLL](#) uint8_t [bps_get_delta_trigger_delay_minimum](#) (uint32_t bps_serial, uint16_t *trigger↵_delay_minimum)

Gets the smallest delta trigger delay that the BPS204 can provide.

- [DLL](#) uint8_t [bps_get_delta_trigger_delay_maximum](#) (uint32_t bps_serial, uint16_t *trigger↵_delay_maximum)

Gets the biggest delta trigger delay that the BPS204 can provide.

4 Function Documentation

4.1 *bps_close()*

```
DLL uint8_t bps_close (
    uint32_t bps_serial )
```

Closes a connection to a BPS204.

Terminates the exclusive communication with a BPS204 and allows you to establish another communication connection.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
----	-------------------	-------------------------

Returns

Error Code

Definition at line [403](#) of file [bps204.cpp](#).

4.2 *bps_connect_single()*

```
DLL uint8_t bps_connect_single (
    uint32_t * bps_serial )
```

Searches for a connected BPS204 and fetch its serial number.

Easier way to find the serial of first connected BPS204. This function combines the following functions: *bps_init*, *bps_search*, *bps_get_info*.

Parameters

out	* <i>bps_serial</i>	Serial number of BPS204
-----	---------------------	-------------------------

4.4 *bps_destroy()*

Returns

Error Code

Definition at line 133 of file [bps204.cpp](#).

4.3 **bps_control()**

```
DLL uint8_t bps_control (
    uint32_t bps_serial,
    uint8_t state )
```

Controls the pulse generation state of the BPS204.

Starts or stops the high voltage generation of the BPS204. Depending on the trigger mode and stability of high voltage, the BPS204 will either start/stop generating pulses or wait for the external trigger.

Remarks

If no probe or high voltage cable is connected, the status will remain in "stop pulsing"

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
in	<i>state</i>	Definition of state parameter: 0 : BPS204 stops pulsing 1 : BPS204 starts pulsing (depending on trigger)

Returns

Error Code

Definition at line 1129 of file [bps204.cpp](#).

4.4 **bps_destroy()**

```
DLL uint8_t bps_destroy ( )
```

Ends the DLL communication.

Closes all active connections and stops all background threads created by the DLL.

Remarks

Make sure to use this function for a clean exit after usage.

Returns

Error Code

Definition at line 181 of file [bps204.cpp](#).

4.5 `bps_get_connections()`

```
DLL uint8_t bps_get_connections (
    uint8_t * num_bps )
```

Gets the number of connected BPS204 devices.

Returns the amount of BPS204 devices that have been previously mapped internally.

Remarks

This function does not actively search for new devices.

Parameters

out	* <i>num_bps</i>	Number of connected BPS204 devices
-----	------------------	------------------------------------

Returns

Error Code

Definition at line 293 of file [bps204.cpp](#).

4.6 `bps_get_delta_trigger_delay_maximum()`

```
DLL uint8_t bps_get_delta_trigger_delay_maximum (
    uint32_t bps_serial,
    uint16_t * trigger_delay_maximum )
```

Gets the biggest delta trigger delay that the BPS204 can provide.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>*trigger_delay_maximum</i>	Highest possible delta trigger delay

Returns

Error Code

Definition at line [1306](#) of file [bps204.cpp](#).

4.7 `bps_get_delta_trigger_delay_minimum()`

```
DLL uint8_t bps_get_delta_trigger_delay_minimum (
    uint32_t bps_serial,
    uint16_t * trigger_delay_minimum )
```

Gets the smallest delta trigger delay that the BPS204 can provide.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>*trigger_delay_minimum</i>	Lowest possible delta trigger delay

Returns

Error Code

Definition at line [1293](#) of file [bps204.cpp](#).

4.8 `bps_get_device_id()`

```
DLL uint8_t bps_get_device_id (
    uint32_t bps_serial,
    uint32_t * id )
```

Gets the device ID of BPS204.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>*id</i>	Device ID of connected BPS204

Returns

Error Code

Definition at line [445](#) of file [bps204.cpp](#).

4.9 `bps_get_firmware_id()`

```
DLL uint8_t bps_get_firmware_id (
    uint32_t bps_serial,
    uint32_t * firmware_id )
```

Gets the firmware ID installed on a connected probe.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>*firmware_id</i>	Specific firmware version installed on a probe

Returns

Error Code

Definition at line [502](#) of file [bps204.cpp](#).

4.10 `bps_get_hardware_id()`

```
DLL uint8_t bps_get_hardware_id (
    uint32_t bps_serial,
    uint32_t * hardware_id )
```

Gets the hardware version of a connected probe.

4.12 bps_get_info()

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>*hardware_id</i>	Specific hardware version of a connected probe

Returns

Error Code

Definition at line [483](#) of file [bps204.cpp](#).

4.11 bps_get_hv_state()

```
DLL uint8_t bps_get_hv_state (
    uint32_t bps_serial,
    uint8_t * hv_stability )
```

Gets high voltage state of BPS204.

The high voltage state reflects the stability of the BPS204 to deliver the exact voltage set by user.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>*hv_stability</i>	Definition of hv_stability parameter: 0 : high voltage value and stability is not guaranteed 1 : high voltage is set and stable

Returns

Error Code

Definition at line [528](#) of file [bps204.cpp](#).

4.12 bps_get_info()

```
DLL uint8_t bps_get_info (
    uint8_t bps_index,
    uint32_t * bps_serial )
```

Gets the serial number from a BPS204.

Goes through the internal map of connected BPS204 devices and returns the serial number at a given index.

Parameters

in	<i>bps_index</i>	Index of connected BPS204 device
out	<i>*bps_serial</i>	Serial number of BPS204

Returns

Error Code

Definition at line [304](#) of file [bps204.cpp](#).

4.13 `bps_get_main_trigger_delay_maximum()`

```
DLL uint8_t bps_get_main_trigger_delay_maximum (
    uint32_t bps_serial,
    uint16_t * trigger_delay_maximum )
```

Gets the biggest main trigger delay that the BPS204 can provide.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>*trigger_delay_maximum</i>	Highest possible main trigger delay

Returns

Error Code

Definition at line [1267](#) of file [bps204.cpp](#).

4.14 `bps_get_main_trigger_delay_minimum()`

```
DLL uint8_t bps_get_main_trigger_delay_minimum (
    uint32_t bps_serial,
    uint16_t * trigger_delay_minimum )
```

Gets the smallest main trigger delay that the BPS204 can provide.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>*trigger_delay_minimum</i>	Lowest possible main trigger delay

Returns

Error Code

Definition at line [1254](#) of file [bps204.cpp](#).

4.15 `bps_get_probe_device_id()`

```
DLL uint8_t bps_get_probe_device_id (
    uint32_t bps_serial,
    uint32_t * probe_device_id )
```

Gets the device ID of BPS204.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>*id</i>	Device ID of connected BPS204

Returns

Error Code

Definition at line [658](#) of file [bps204.cpp](#).

4.16 `bps_get_probe_firmware_id()`

```
DLL uint8_t bps_get_probe_firmware_id (
    uint32_t bps_serial,
    uint32_t * probe_firmware_id )
```

Gets the firmware ID installed on BPS204.

4.18 `bps_get_probe_state()`

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>*firmware_id</i>	Specific bps firmware version of connected BPS204

Returns

Error Code

Definition at line [683](#) of file [bps204.cpp](#).

4.17 `bps_get_probe_serial()`

```
DLL uint8_t bps_get_probe_serial (
    uint32_t bps_serial,
    uint32_t * probe_serial )
```

**** Gets the serial number from a connected probe.****

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>*probe_serial</i>	Serial number of a connected probe

Returns

Error Code

Definition at line [633](#) of file [bps204.cpp](#).

4.18 `bps_get_probe_state()`

```
DLL uint8_t bps_get_probe_state (
    uint32_t bps_serial,
    uint8_t * probe_state )
```

Gets state of probe connected to BPS204.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>*probe_state</i>	Definition of probe_state parameter: 0 : probe is not connected 1 : probe is connected but no communication can be established 2 : no high voltage cable is connected to the probe 3 : probe ready to pulse

Returns

Error Code

Definition at line [535](#) of file [bps204.cpp](#).

4.19 `bps_get_pulse_frequency()`

```
DLL uint8_t bps_get_pulse_frequency (
    uint32_t bps_serial,
    float * frequency )
```

Gets the frequency [Hz] at which the BPS204 will generate pulses.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>*frequency</i>	Frequency in hertz [Hz]

Returns

Error Code

Definition at line [945](#) of file [bps204.cpp](#).

4.20 `bps_get_pulse_frequency_maximum()`

```
DLL uint8_t bps_get_pulse_frequency_maximum (
    uint32_t bps_serial,
    float * max_frequency )
```

Gets the highest frequency [Hz] at which the BPS204 can generate pulse.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>*max_frequency</i>	Highest possible frequency in hertz [Hz]

Returns

Error Code

`*uint8_t error_code = BPS_OK;`

Definition at line [1018](#) of file [bps204.cpp](#).

4.21 `bps_get_pulse_frequency_minimum()`

```
DLL uint8_t bps_get_pulse_frequency_minimum (
    uint32_t bps_serial,
    float * min_frequency )
```

Gets the lowest frequency [Hz] at which the BPS204 can generate pulses.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>*min_frequency</i>	Lowest possible frequency in hertz [Hz]

Returns

Error Code

Definition at line [991](#) of file [bps204.cpp](#).

4.22 `bps_get_pulse_level()`

```
DLL uint8_t bps_get_pulse_level (
    uint32_t bps_serial,
    float * level )
```

Gets the amplitude [V] of the pulse that the BPS204 is set to generate.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>*level</i>	Voltage level in volt [V]

Returns

Error Code

Definition at line [722](#) of file [bps204.cpp](#).

4.23 `bps_get_pulse_level_difference_minimum()`

```
DLL uint8_t bps_get_pulse_level_difference_minimum (
    uint32_t bps_serial,
    float * delta_level )
```

Gets the minimum voltage level [V] resolution that the BPS204 can provide.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>delta_level</i>	Voltage level resolution in volt [V]

Returns

Error Code

Definition at line [764](#) of file [bps204.cpp](#).

4.24 `bps_get_pulse_level_maximum()`

```
DLL uint8_t bps_get_pulse_level_maximum (
    uint32_t bps_serial,
    float * max_level )
```

Gets the maximum voltage level [V] that the BPS204 can generate.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>max_level</i>	Maximum possible voltage in volt [V]

Returns

Error Code

Definition at line [750](#) of file [bps204.cpp](#).

4.25 *bps_get_pulse_level_minimum()*

```
DLL uint8_t bps_get_pulse_level_minimum (
    uint32_t bps_serial,
    float * min_level )
```

Gets the minimum voltage level [V] that the BPS204 can generate.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>min_level</i>	Minimum possible voltage in volt [V]

Returns

Error Code

Definition at line [736](#) of file [bps204.cpp](#).

4.26 *bps_get_pulse_mode()*

```
DLL uint8_t bps_get_pulse_mode (
    uint32_t bps_serial,
    uint16_t * pulse_mode )
```

Gets the type of pulse that the BPS204 will generate.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>*pulse_mode</i>	Definition of pulse_mode parameter 0 : BPS204 generates single pulses 1 : BPS204 generates double pulses

Returns

Error Code

Definition at line [1069](#) of file [bps204.cpp](#).

4.27 bps_get_pulse_polarity()

```
DLL uint8_t bps_get_pulse_polarity (
    uint32_t bps_serial,
    uint8_t * pulse_polarity )
```

Gets the polarity of the pulses that the BPS204 will generate.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>*pulse_polarity</i>	Definition of pulse_polarity parameter: 0 : BPS204 generates positiv pulses 1 : BPS204 generates negative pulses

Returns

Error Code

Definition at line [708](#) of file [bps204.cpp](#).

4.28 bps_get_state()

```
DLL uint8_t bps_get_state (
    uint32_t bps_serial,
    uint8_t * bps_state )
```

Get state of BPS204.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>*bps_state</i>	Definition of bps_state parameter: 0 : BPS204 is idling 1 : BPS204 is busy 2 : BPS204 is pulsing continuously 3 : BPS204 is ready and waiting for sync trigger 4 : BPS204 is ready and waiting for probe trigger

Returns

Error Code

Definition at line [521](#) of file [bps204.cpp](#).

4.29 **bps_get_states()**

```
DLL uint8_t bps_get_states (
    uint32_t bps_serial,
    uint8_t * bps_state,
    uint8_t * hv_stability,
    uint8_t * probe_state )
```

Get all 3 states at once.

Remarks

All states are requested in the background automaticly every 20ms.

Parameters

in	<i>bps_serial</i>	serial number of BPS204
out	<i>*bps_state</i>	Definition of bps_state parameter: 0 : BPS204 is idle 1 : BPS204 is busy 2 : BPS204 is pulsing continuously 3 : BPS204 is ready and waiting for BPS trigger 4 : BPS204 is ready and waiting for probe trigger
out	<i>*hv_stability</i>	Definition of hv_stability parameter: 0 : high voltage value and stability is not guaranteed 1 : high voltage is set and stable
out	<i>*probe_state</i>	Definition of probe_state parameter: 0 : probe is not connected 1 : probe is connected but no communication can be established 2 : no high voltage cable is connected to the probe 3 : probe ready to pulse

Returns

Error Code

Definition at line [542](#) of file [bps204.cpp](#).

4.30 **bps_get_trigger_delay()**

```
DLL uint8_t bps_get_trigger_delay (
    uint32_t bps_serial,
```

4.31 `bps_get_trigger_mode()`

```
uint8_t trigger_channel,  
uint16_t * trigger_delay )
```

Gets a trigger delay that the BPS204 is set to.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
in	<i>trigger_channel</i>	Definition of trigger_channel parameter: 0 : Main trigger delay 1 : Delta delay between first and second pulse
out	<i>*trigger_delay</i>	Definition of trigger_delay parameter: <= 510 : Value in steps

Returns

Error Code

Definition at line [1192](#) of file [bps204.cpp](#).

4.31 `bps_get_trigger_mode()`

```
DLL uint8_t bps_get_trigger_mode (  
    uint32_t bps_serial,  
    uint16_t * trigger_mode )
```

Gets the trigger mode that the BPS204 is currently set to.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>*trigger_mode</i>	Definition of trigger_mode parameter 0 : Free running mode / ext. trigger off 1 : BPS single trigger 2 : BPS doublet trigger 3 : Probe trigger single cable 4 : Probe trigger dual cable

\return Error Code

Definition at line [1160](#) of file [bps204.cpp](#).

4.32 bps_init()

```
DLL uint8_t bps_init ( )
```

Preperes for an initial connection to a BPS204.

Closes previous and possibly still active connections and clears internal list of connected BPS204 devices.

Remarks

Make sure to call this function before using any of the other function. Otherwise, they will return invalid information.

Returns

Error Code

Definition at line 158 of file [bps204.cpp](#).

4.33 bps_lost_connection()

```
DLL uint8_t bps_lost_connection (
    uint32_t bps_serial,
    uint8_t * lost_connection )
```

Checks if BPS204 has lost connection.

After an exclusive connection for communication is established, an internal thread is constantly checking different states of the BPS204. If this communication is interrupted, a flag is set.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
out	<i>*lost_connection</i>	Definition of lost_connection parameter: 0 : no connection was lost 1 : connection was lost

Returns

Error Code

Definition at line 428 of file [bps204.cpp](#).

4.34 bps_open()

```
DLL uint8_t bps_open (
    uint32_t bps_serial )
```

Opens a connection to a BPS204.

Creates an internal object, establishes an exclusive USB communication with a BPS204 and resets the connectionlost flag to 0.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
----	-------------------	-------------------------

Returns

Error Code

Definition at line [322](#) of file [bps204.cpp](#).

4.35 bps_search()

```
DLL uint8_t bps_search (
    uint8_t * new_bps )
```

Searches for newly connected BPS204 devices.

Looks for BPS204 devices connected by USB through analyzing their preprogrammed descriptions and adds them to an internal map.

Remarks

Every subsequent call of this function returns a zero if setup does not change.

Parameters

out	<i>*new_bps</i>	Number of newly connected BPS204 devices
-----	-----------------	--

Returns

Error Code

Definition at line [213](#) of file [bps204.cpp](#).

4.36 *bps_set_pulse_frequency()*

```
DLL uint8_t bps_set_pulse_frequency (
    uint32_t bps_serial,
    float frequency )
```

Sets the frequency [Hz] at which the BPS204 will generate pulses.

Remarks

Only relevant in free running mode (no external trigger).

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
in	<i>frequency</i>	Frequency in hertz [Hz]

Returns

Error Code

Definition at line [958](#) of file [bps204.cpp](#).

4.37 *bps_set_pulse_level()*

```
DLL uint8_t bps_set_pulse_level (
    uint32_t bps_serial,
    float level )
```

Sets the amplitude [V] of the pulse that the BPS204 is set to generate.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
in	<i>level</i>	Voltage level in volt [V]

Returns

Error Code

Definition at line [1108](#) of file [bps204.cpp](#).

4.38 *bps_set_pulse_mode()*

```
DLL uint8_t bps_set_pulse_mode (
    uint32_t bps_serial,
    uint8_t pulse_mode )
```

Sets the type of pulse that the BPS204 should generate.

Remarks

Note that some pulse mode and trigger combinations are not practical i.e. single pulse mode and double trigger input.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
in	<i>pulse_mode</i>	Definition of pulse_mode parameter: 0 : BPS204 generates single pulse 1 : BPS204 generates double pulse

Returns

Error Code

Definition at line [1044](#) of file [bps204.cpp](#).

4.39 *bps_set_pulse_polarity()*

```
DLL uint8_t bps_set_pulse_polarity (
    uint32_t bps_serial,
    uint8_t pulse_polarity )
```

Sets the polarity of the pulses that the BPS204 should generate.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
in	<i>pulse_polarity</i>	Definition of pulse_polarity parameter: 0 : BPS204 generates positiv pulses 1 : BPS204 generateS negativ pulses

Returns

Error Code

Definition at line [1088](#) of file [bps204.cpp](#).

4.40 `bps_set_reset()`

```
DLL uint8_t bps_set_reset (
    uint32_t bps_serial )
```

Resets the BPS204.

Sends a reset command to the BPS204, which turns the BPS204 in its initial state.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
----	-------------------	-------------------------

Returns

Error Code

Definition at line [1342](#) of file [bps204.cpp](#).

4.41 **bps_set_trigger_delay()**

```
DLL uint8_t bps_set_trigger_delay (
    uint32_t bps_serial,
    uint8_t trigger_channel,
    uint16_t trigger_delay )
```

Sets a trigger delay of BPS204.

Remarks

Depending on pulse and trigger mode this option may be ignored.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
in	<i>trigger_channel</i>	Definition of trigger_channel parameter: 0 : Main trigger delay 1 : Delta delay between first and second pulse
in	<i>trigger_delay</i>	Definition of trigger_delay parameter: <= 510 : Value in steps

Returns

Error Code

Definition at line [1173](#) of file [bps204.cpp](#).

4.42 **bps_set_trigger_mode()**

```
DLL uint8_t bps_set_trigger_mode (
    uint32_t bps_serial,
    uint8_t trigger_mode )
```

Sets the type of trigger mode of the BPS204.

BPS204 is able to generate pulses periodically without an external trigger while in "free running mode". Otherwise, a pulse is only generated after an external trigger event.

Parameters

in	<i>bps_serial</i>	Serial number of BPS204
in	<i>trigger_mode</i>	Definition of trigger_mode parameter 0 : Free running mode / ext. trigger off 1 : BPS single trigger 2 : BPS doublet trigger 3 : Probe trigger single cable 4 : Probe trigger dual cable

Returns

Error Code

Definition at line [1140](#) of file [bps204.cpp](#).

Index

bps204.dll, [2](#)

bps204.h

- [bps_close, 6](#)
- [bps_connect_single, 6](#)
- [bps_control, 7](#)
- [bps_destroy, 7](#)
- [bps_get_connections, 8](#)
- [bps_get_delta_trigger_delay_maximum, 8](#)
- [bps_get_delta_trigger_delay_minimum, 9](#)
- [bps_get_device_id, 9](#)
- [bps_get_firmware_id, 10](#)
- [bps_get_hardware_id, 10](#)
- [bps_get_hv_state, 11](#)
- [bps_get_info, 11](#)
- [bps_get_main_trigger_delay_maximum, 12](#)
- [bps_get_main_trigger_delay_minimum, 12](#)
- [bps_get_probe_device_id, 13](#)
- [bps_get_probe_firmware_id, 13](#)
- [bps_get_probe_serial, 14](#)
- [bps_get_probe_state, 14](#)
- [bps_get_pulse_frequency, 15](#)
- [bps_get_pulse_frequency_maximum, 15](#)
- [bps_get_pulse_frequency_minimum, 16](#)
- [bps_get_pulse_level, 16](#)
- [bps_get_pulse_level_difference_minimum, 17](#)
- [bps_get_pulse_level_maximum, 17](#)
- [bps_get_pulse_level_minimum, 18](#)
- [bps_get_pulse_mode, 18](#)
- [bps_get_pulse_polarity, 19](#)
- [bps_get_state, 19](#)
- [bps_get_states, 21](#)
- [bps_get_trigger_delay, 21](#)
- [bps_get_trigger_mode, 22](#)
- [bps_init, 22](#)
- [bps_lost_connection, 23](#)
- [bps_open, 23](#)
- [bps_search, 24](#)
- [bps_set_pulse_frequency, 24](#)
- [bps_set_pulse_level, 25](#)
- [bps_set_pulse_mode, 25](#)
- [bps_set_pulse_polarity, 26](#)
- [bps_set_reset, 27](#)

[bps_set_trigger_delay, 28](#)

[bps_set_trigger_mode, 28](#)

- [bps_close](#)
 - [bps204.h, 6](#)
- [bps_connect_single](#)
 - [bps204.h, 6](#)
- [bps_control](#)
 - [bps204.h, 7](#)
- [bps_destroy](#)
 - [bps204.h, 7](#)
- [bps_get_connections](#)
 - [bps204.h, 8](#)
- [bps_get_delta_trigger_delay_maximum](#)
 - [bps204.h, 8](#)
- [bps_get_delta_trigger_delay_minimum](#)
 - [bps204.h, 9](#)
- [bps_get_device_id](#)
 - [bps204.h, 9](#)
- [bps_get_firmware_id](#)
 - [bps204.h, 10](#)
- [bps_get_hardware_id](#)
 - [bps204.h, 10](#)
- [bps_get_hv_state](#)
 - [bps204.h, 11](#)
- [bps_get_info](#)
 - [bps204.h, 11](#)
- [bps_get_main_trigger_delay_maximum](#)
 - [bps204.h, 12](#)
- [bps_get_main_trigger_delay_minimum](#)
 - [bps204.h, 12](#)
- [bps_get_probe_device_id](#)
 - [bps204.h, 13](#)
- [bps_get_probe_firmware_id](#)
 - [bps204.h, 13](#)
- [bps_get_probe_serial](#)
 - [bps204.h, 14](#)
- [bps_get_probe_state](#)
 - [bps204.h, 14](#)
- [bps_get_pulse_frequency](#)
 - [bps204.h, 15](#)
- [bps_get_pulse_frequency_maximum](#)
 - [bps204.h, 15](#)
- [bps_get_pulse_frequency_minimum](#)
 - [bps204.h, 16](#)
- [bps_get_pulse_level](#)

bps204.h, [16](#)
bps_get_pulse_level_difference_minimum
bps204.h, [17](#)
bps_get_pulse_level_maximum
bps204.h, [17](#)
bps_get_pulse_level_minimum
bps204.h, [18](#)
bps_get_pulse_mode
bps204.h, [18](#)
bps_get_pulse_polarity
bps204.h, [19](#)
bps_get_state
bps204.h, [19](#)
bps_get_states
bps204.h, [21](#)
bps_get_trigger_delay
bps204.h, [21](#)
bps_get_trigger_mode
bps204.h, [22](#)
bps_init
bps204.h, [22](#)
bps_lost_connection
bps204.h, [23](#)
bps_open
bps204.h, [23](#)
bps_search
bps204.h, [24](#)
bps_set_pulse_frequency
bps204.h, [24](#)
bps_set_pulse_level
bps204.h, [25](#)
bps_set_pulse_mode
bps204.h, [25](#)
bps_set_pulse_polarity
bps204.h, [26](#)
bps_set_reset
bps204.h, [27](#)
bps_set_trigger_delay
bps204.h, [28](#)
bps_set_trigger_mode
bps204.h, [28](#)