

# Downlink command interface

## General remarks

- Parameter changes without the save option are lost after a reset or power cycle.
- Command with the save option permanently store the current parameter settings in flash.
- Use with caution: Certain commands / parameters may render a device unresponsive / unreachable in the field until it is manually reset.
- Send downlink command using LoRaWAN port 1.
- Downlink command format: | Code | Parameter | CRC |.
- Code: 16-bit unsigned integer, see table below.
- Parameter: 16-bit unsigned integer. Set to 0x0000 if not used.
- CRC: CRC-16 (Modbus) of | Code | Parameter |. See Section CRC-16 computation function. Online calculator: e.g. <https://www.lammertbies.nl/comm/info/crc-calculation.html>
- If CRC does not match, the command is ignored.

## Command list

Command	Code	Description
set period	0x0001	Set sampling period in seconds (1...65535).
set period + save	0x0002	... + save settings.
set dr	0x0003	Set default Tx data rate. Used for next Tx. Actual data rate for following Tx may vary, if ADR is enabled.
set dr + save	0x0004	... + save settings.
set adr on	0x0005	Enable ADR.
set adr on + save	0x0006	... + save settings.
set adr off	0x0007	Disable ADR.
set adr off + save	0x0008	... + save settings.
set dr_min	0x0009	Set minimum data rate (overrides ADR settings).
set dr_min + save	0x000A	... + save settings.
set dr_max	0x000B	Set maximum data rate (overrides ADR settings).
set dr_max + save	0x000C	... + save settings.
set pwridx_min	0x000D	Set minimum Tx power index (overrides ADR settings).
set pwridx_min + save	0x000E	... + save settings.
set pwridx_max	0x000F	Set maximum Tx power index (overrides ADR settings).
set pwridx_max + save	0x0010	... + save settings.

Command	Code	Description
set send_period	0x0011	Set send period. Examples: 0 or 1: send after every sampling; 4: send after every fourth sampling.
set send_period + save	0x0012	... + save settings.
set join_period	0x0013	Set re-join period in hours (0...1000). Examples: 24: re-join network every 24 hours. 0: never re-join.
set join_period + save	0x0014	... + save settings.
set param 0	0x0020	Set parameter 0 (0...65534; 65535: invalid).
set param 1	0x0021	Set parameter 1 (0...65534; 65535: invalid).
...	...	...
set param 15	0x002F	Set parameter 15 (0...65534; 65535: invalid).
set param 0 + save	0x0030	... + save settings.
set param 1 + save	0x0031	... + save settings.
...	...	...
set param 15 + save	0x003F	... + save settings.
reset	0xFEFE	Reset device; un-saved parameter changes are lost.
factory reset	0xFEFE0	Erase settings in flash and reset.

## Examples (EU868 band)

set period 600	000102587E51
set period 3600	00010E104854
set period 600 seconds + save	000202587EA1
set period 60 seconds + save	0002003CF5A1
set dr 3	0003000325B0
set dr 3 + save	00040003E401
set adr on	000500002510
set adr on + save	0006000025E0
set adr off	00070000E5B1
set adr off + save	00080000E681
set send_period 10	0011000A26D0
set send_period 10 + save	0012000A2620
set param 0 1000	002003E85001
set param 0 1000 + save	003003E89500
set param 1 2000	002107D08253
set param 1 2000 + save	003107D04752
reset	FEFE00003C50

factory reset

FEF00000FF31

## Data rate and power index (EU868 band)

Data rate	Configuration	Bit rate	Power idx	Tx Power
0	SF12 / 125 kHz	250 bit/s	1	14 dBm
1	SF11 / 125 kHz	440 bit/s	2	11 dBm
2	SF10 / 125 kHz	980 bit/s	3	8 dBm
3	SF9 / 125 kHz	1760 bit/s	4	5 dBm
4	SF8 / 125 kHz	3125 bit/s	5	2 dBm
5	SF7 / 125 kHz	5470 bit/s		

## Data rate and power index (US915 band)

Data rate	Configuration	Bit rate	Power idx	Tx Power
0	SF10 / 125 kHz	980 bit/s	5	20 dBm
1	SF9 / 125 kHz	1760 bit/s	7	16 dBm
2	SF8 / 125 kHz	3125 bit/s	8	14 dBm
3	SF7 / 125 kHz	5470 bit/s	9	12 dBm
4	SF8 / 500 kHz	12500 bit/s	10	10 dBm

## CRC-16 computation function

```

////////////////////////////////////
// CRC-16-IBM (used by Modbus, USB, others. polynomial: 0x8005 / 0xA001)
////////////////////////////////////

```

```

uint16_t crc16(uint8_t* buf, uint16_t size) {
    uint16_t crc;
    uint8_t n, m, x;
    crc = 0xFFFF;
    m = size;
    x = 0;
    // loop over all bits
    while (m > 0) {
        crc = crc ^ buf[x];
        for (n=0; n<8; n++) {
            if (crc & 1) {
                crc = crc >> 1;
                crc = crc ^ 0xA001;
            }
            else {
                crc = crc >> 1;
            }
        }
        m--;
        x++;
    }
}

```

```
    return crc;  
}
```