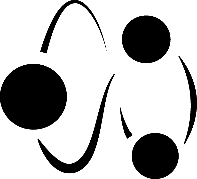
**SMARTBALL v1.0**

Datagram specifications v0.2

last update – 16/09/2019

# INTRODUCTION

Smartballs send and receive specific UDP Datagrams, this choice has been made over OSC, Artnet or other kind of existing protocol to save bandwidth as much as possible. This protocol is still in active development and will likely to evolve during the alpha and beta test phase.

This document is compatible with the following release:

* hardware v1.x : <https://github.com/siteswapjuggler/smartball-hardware>
* firmware v0.2 beta 4 : <https://github.com/siteswapjuggler/smartball-firmware/tree/v0.2-beta.4>
* externals v0.2 beta 2 : <https://github.com/siteswapjuggler/smartball-externals/tree/v0.2-beta.2>

# VERSION NOTE

Command updates:

* Color become Foreground color

New commands:

* Background color
* Strobe command
* Master command

SUMMARY

[INTRODUCTION 1](#_Toc19532087)

[VERSION NOTE 1](#_Toc19532088)

[DATAGRAM STRUCTURE 4](#_Toc19532089)

[UPSTREAM PACKETS 4](#_Toc19532090)

[DOWNSTREAM PACKETS 4](#_Toc19532096)

[UPSTREAM COMMANDS 5](#_Toc19532103)

[PING 5](#_Toc19532104)

[REBOOT 5](#_Toc19532105)

[SLEEP 5](#_Toc19532106)

[FACTORY SETTINGS 5](#_Toc19532107)

[SAVE FACTORY SETTINGS 5](#_Toc19532108)

[COLOUR (FOREGROUND) 6](#_Toc19532109)

[COLOUR (BACKGROUND) 6](#_Toc19532110)

[STREAM 6](#_Toc19532111)

[IMU SETTINGS 6](#_Toc19532112)

[ACCELEROMETER RANGE 7](#_Toc19532113)

[GYROSCOPE RANGE 7](#_Toc19532114)

[SAVE IMU SETTINGS 7](#_Toc19532115)

[INFRARED 8](#_Toc19532116)

[MOTOR 8](#_Toc19532117)

[STROBE 8](#_Toc19532118)

[MASTER 8](#_Toc19532119)

[DOWNSTREAM COMMANDS 9](#_Toc19532120)

[PONG 9](#_Toc19532121)

[BATTERY VALUE 9](#_Toc19532122)

[IMU VALUES 9](#_Toc19532123)

# DATAGRAM STRUCTURE

## UPSTREAM PACKETS

* Start of packet 1 byte
* Command 1 byte
* Data Length 2 bytes
* Data (optional)
* Checksum 1 byte

### SOP – Start of Packet

Start of packet is always 0xE7.

### CMD – Command

Commands byte is unique to every message.

### LEN – Data length

Length bytes are the MSB and LSB of the uint16 number of data.

### DATA (optional)

Data depends on each command and is optional. See command description below for further information.

### CKS – Checksum

Checksum is the sum modulo 256 of every byte from CMD byte to the last DATA byte.

## DOWNSTREAM PACKETS

* Start of packet 1 byte
* Serial number 2 bytes
* Command 1 byte
* Length 2 bytes
* Data depending on length
* Checksum 1 byte

### SOP – Start of Packet

Start of packet is always 0xE7.

### SN – Serial Number

Two bytes uint16 serial number.

### CMD – Command

Commands byte is unique to every message.

### LEN – Data length

Length bytes are the MSB and LSB of the uint16 number of data.

### DATA (optional)

Data depends on each command and is optional. See command description below for further information.

### CKS – Checksum

Checksum is the sum module 256 of every byte from CMD byte to the last DATA byte.

# UPSTREAM COMMANDS

## PING

Ping command trigger a pong answer.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SOP | CMD | LEN | | CKS |
| 0xE7 | 0x01 | 0x00 | 0x00 | 0x01 |

## REBOOT

Reboot command trigger the reboot of the ball (*does not work on first boot after firmware update*).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SOP | CMD | LEN | | CKS |
| 0xE7 | 0x02 | 0x00 | 0x00 | 0x02 |

## SLEEP

Sleep command trigger the deep sleep mode. Balls need to be power off and on again to restart.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SOP | CMD | LEN | | CKS |
| 0xE7 | 0x03 | 0x00 | 0x00 | 0x03 |

## FACTORY SETTINGS

Factory settings command sets new factory settings value. Factory settings includes a serial number, a device flag and an adc scaling factor.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SOP | CMD | LEN | | DATA | CKS |
| 0xE7 | 0x10 | 0x00 | 0x05 | 5 bytes | checksum |

Data structure:

* serial number: uint16 value from 0 to 65535
* device flag: 16 bits flag describing available devices
* adc scaling factor: unsigned Q15.1 value from 0.0 to 6553.5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SN\_MSB | SN\_LSB | FLAG\_MSB | FLAG\_LSB | SCL\_MSB | SCL\_LSB |

DEVICE FLAG structure:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | - | - | - | - | - | - | - | - | - | - | BUZ | MOT | IRL | IMU | RGB |

## SAVE FACTORY SETTINGS

Save factory command trigger EEPROM writing of the factory settings.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SOP | CMD | LEN | | CKS |
| 0xE7 | 0x11 | 0x00 | 0x00 | 0x11 |

## COLOUR (FOREGROUND)

Colour command change current colours according to the number of RGB values sent. To be clearer data length will implicitly tell the ball which colour mode to apply, see options below:

* 3 values: 1 colour mode
* 6 values: 2 colours mode (1 per hemisphere)
* 9 values: 3 colours mode (3 pairs of opposite leds)
* 18 values: 6 colours mode (6 individual leds)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SOP | CMD | LEN | | DATA | CKS |
| 0xE7 | 0x20 | 0x00 | variable | variable | checksum |

Data structure:

* Colour 1 Red value : uint8 value from 0 to 255
* Colour 1 Green value : uint8 value from 0 to 255
* Colour 1 Blue value : uint8 value from 0 to 255
* (optional) colour 2/3/4/5/6 RGB values

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| R1 | G1 | B1 | (R2) | (G2) | (B2) | and so on… |

## COLOUR (BACKGROUND)

Background colour command change the current background colours according to the number of RGB values sent. Background colour is the second colour of the stroboscopic effect. Data structures and options are the as the foreground colour command.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SOP | CMD | LEN | | DATA | CKS |
| 0xE7 | 0x22 | 0x00 | variable | variable | checksum |

## STREAM

Stream command act almost the same way as colour command apart that you can address multiple balls at one time. Doing this you need to add one extra data byte to precise the number of colours sent for each ball. Balls will change colours according to its serial number modulo (LEN-1)/(N\_COLOURS \* 3).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SOP | CMD | LEN | | DATA | CKS |
| 0xE7 | 0x21 | variable | variable | variable | checksum |

Data structure EXAMPLE:

Even serial number balls goes red & blue while odd serial number balls goes black & white:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N\_COL | RGB1.1 | RGB1.2 | RGB2.1 | RGB 2.2 |
| 0x02 | 0xFF 0x00 0x00 | 0x00 0x00 0xFF | 0xFF 0xFF 0xFF | 0x00 0x00 0x00 |

This command has been design for the first 30 balls during the alpha tests. Future firmware will use customisable id values instead of serial numbers.

## IMU SETTINGS

IMU settings command will change current IMU feedback settings.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SOP | CMD | LEN | | FLAG | CKS |
| 0xE7 | 0x30 | 0x00 | 0x01 | variable | Checksum |

Data structure:

* IMU flag: 8 bits flag describing available feedbacks.

DEVICE FLAG structure:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | - | - | - | TMP | MAG | GYR | ACC |

## ACCELEROMETER RANGE

Accelerometer range command will change current accelerometer full-scale range settings.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SOP | CMD | LEN | | PAR | CKS |
| 0xE7 | 0x32 | 0x00 | 0x01 | variable | checksum |

Data structure:

* Full-scale parameter: 8 bits parameter.

PARAMETER VALUEs:

* 0x00: -2g / +2g.
* 0x01: -4g / +4g.
* 0x02: -8g / +8g.
* 0x03: -16g / +16g (default).

## GYROSCOPE RANGE

Gyroscope range command will change current gyroscope full-scale range settings.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SOP | CMD | LEN | | PAR | CKS |
| 0xE7 | 0x33 | 0x00 | 0x01 | variable | checksum |

Data structure:

* Full-scale parameter: 8 bits parameter.

PARAMETER VALUEs:

* 0x00: -250dps / +250dps.
* 0x01: -500dps / +500dps.
* 0x02: -1000dps / +1000dps.
* 0x03: -2000dps / +2000dps (default).

## SAVE IMU SETTINGS

Save IMU command trigger EEPROM writing of the IMU settings.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SOP | CMD | LEN | | CKS |
| 0xE7 | 0x031 | 0x00 | 0x00 | 0x31 |

## INFRARED

Infrared command change the current value of the infrared channel PWM.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SOP | CMD | LEN | | DATA | CKS |
| 0xE7 | 0x40 | 0x00 | 0x02 | 2 bytes | checksum |

Data structure:

* value : uint16 value from 0 to 1023

|  |  |
| --- | --- |
| VAL\_MSB | VAL\_LSB |

## MOTOR

Motor command change the current value of the motor channel PWM.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SOP | CMD | LEN | | DATA | CKS |
| 0xE7 | 0x50 | 0x00 | 0x02 | 2 bytes | checksum |

Data structure:

* value : uint16 value from 0 to 1023

|  |  |
| --- | --- |
| MSB | LSB |

## STROBE

Strobe command change the current interval of the inner stroboscopic effect. Using this effect switch between foreground and background colour every *n* milliseconds.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SOP | CMD | LEN | | DATA | CKS |
| 0xE7 | 0x60 | 0x00 | 0x01 | 1 bytes | checksum |

Data structure:

* interval (ms) : uint8 value from 10 to 255

## MASTER

Master command is a global dimmer for every colour channel. The dimmer value does not affect the colours memory it only change the amount of colours forwarded to the RGB Leds.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SOP | CMD | LEN | | DATA | CKS |
| 0xE7 | 0x70 | 0x00 | 0x01 | 1 bytes | checksum |

Data structure:

* dimmer : uint8 value from 0 to 255

# DOWNSTREAM COMMANDS

## PONG

Pong is answering the ping command.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SOP | SN | | CMD | LEN | | CKS |
| 0xE7 | MSB | LSB | 0x01 | 0x00 | 0x00 | 0x01 |

## BATTERY VALUE

Battery feedback transmit Vshare voltage value stored as 1/100th Volts per LSB.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SOP | SN | | CMD | LEN | | DATA | CKS |
| 0xE7 | MSB | LSB | 0x00 | 0x00 | 0x02 | 2 bytes | 0x01 |

Data structure:

* value: Q14.2 value from 0V to 5.2V, this value is adjusted by the adc scaling factory setting.

|  |  |
| --- | --- |
| VAL\_MSB | VAL\_LSB |

## IMU VALUES

IMU feedback transmit IMU values according to the current IMU flag.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SOP | SN | | CMD | LEN | | DATA | CKS |
| 0xE7 | MSB | LSB | 0x30 | 0x00 | variable | variable | Checksum |

Data structure:

* IMU flag: 8 bits flag describing available feedbacks (see IMU settings for description)
* (optional) accelerometer XYZ : 6 bytes containing the XYZ Q14.2 values of the accelerometer.
* (optional) gyroscope XYZ : 6 bytes containing the XYZ Q14.2 values of the gyroscope.
* (optional) magnetometer XYZ : 6 bytes containing the XYZ Q14.2 values of the magnetometer.
* (optional) temperature : Q14.2 temperature value.

Data structure EXAMPLE:

Optional data stacks after directly after the flag depending on enable bits order.

|  |  |  |
| --- | --- | --- |
| FLAG | GYR\_XYZ | TEMP |
| 0x0A | 6 bytes | 2 bytes |

|  |  |
| --- | --- |
| document version | |
| V1.0 | Initial release |