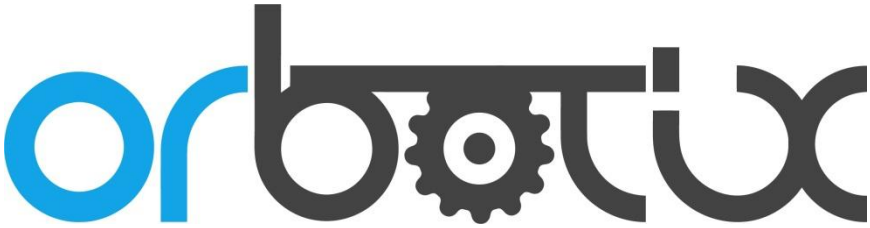
Sphero Shell Commands

document revision 1.0



©Orbotix Inc.

Introduction 3
[System] 4
[Control System] 4
[LEDs] 4
[Raw Motor] 5
[Configuration Block] 5
[Standby/Sleep] 5
[Voltage Commands] 6
[Debugging] 6
[Gyro Sensor] 7
[Accelerometer Sensor] 8
[Bluetooth] 8
[Collision Detection] 8
[Test Results] 9
[Macros] 9
[orbBasic] 9
Revision History 10

Introduction

In addition to binary API commands (that begin with FFh FFh …) the communications processor will also accept ASCII characters. This out-of-band data is buffered separately until a CR (0Dh) character is received and then it is submitted to the Shell Processer for decoding and execution. This document lists and describes the various commands along with which version of the firmware they were introduced (if added in 2012).

Note that you will need to place Sphero into User Hack mode to enable the Shell. The simple byte string

```
FFh FFh 02h 42h 33h 02h 01h 85h
```

will accomplish this transition, and the mode persists across power cycles. So you only need to send it once.

You can connect to the Shell through a standard terminal program (like TeraTerm) and you will want to configure it to send a CR when Enter is pressed, and interpret LF (0Ah) as a linefeed. Turn local echo off.

Parameters are italicized. Numbers expected to be decimal. Multiple parameters are separated by a space (for example lc 255 128 64).

[System]

Command	Description	Ver
ver	Returns module versions	
sv	Displays various state variables	
vxr *x.y*	Temporary override of the Main App version to x.y	0.87
br	Get board revision ID and raw ADC value	
sof *x*	Set option flag x [0..31] or 32 to display	1.15
cof *x*	Clear option flag x [0..31] or 32 to display	1.15
x11	Hard reset of Sphero	
r *x*	Clear the core RCC flag (x=0) or display the RCC_CSR register (x > 0)s	
st	Display detailed results of processor self test	
smd	Display device mode	
smn	Set device mode to Normal	
smu	Set device mode to User Hack	

[Control System]

Command	Description	Ver
l *x*	Enable the control system (x>0) or disable it (l=0)	
pd	Display PID constants for pitch, roll, yaw	
ppp *x*	Set proportional gain on pitch axis to x	
ppi *x*	Set integral gain on pitch axis to x	
ppd *x*	Set differential gain on pitch axis to x	
prp *x*	Set proportional gain on roll axis to x	
pri *x*	Set integral gain on roll axis to x	
prd *x*	Set differential gain on roll axis to x	
pyp *x*	Set proportional gain on yaw axis to x	
pyi *x*	Set integral gain on pitch yaw to x	
pyd *x*	Set differential gain on pitch yaw to x	
hss *x*	Set fast heading slew rate to x [40..1200 degrees/sec]	1.16
hsg	Display fast heading slew rate	1.16
sr *x y*	Synthesize a roll command at heading x [0..359] and speed y [0..255] (removed in 1.21)	1.20-
roll *x y*	Synthesize a roll command at heading x [0..359] and speed y [0..255]	1.21
hs *x*	Set heading to x [0..359]	
vdd	Display Vector Drive settings	1.20
vdc *x*	Set Vector Drive coefficient to x [0.0 .. 0.005]	1.20
vdm *x*	Set Dynamic Vector Drive minimum to x [0..200]	1.20
vdx *x*	Set Dynamic Vector Drive maximum to x [0..200]	1.20
sl	Start self leveling routine with parameters assigned by slp	1.20
slp *x y z*	Display self leveling parameters if x=y=z=0 or assign x to the angle limit ([1..90] degrees), y to the timeout ([1..255] seconds) and z as the level criteria met threshold in 1/100ths of a second [1..255].	1.20

[LEDs]

Command	Description	Ver

lc *r g b*	Set the RGB LED to r,g,b [0..255]	
bl *x*	Set the back LED brightness to x [0..255]	

[Raw Motor]

Note that the control system must first be disabled for these to have an effect.

Command	Description	Ver
mf *x*	Command both motors forward with raw power x [0..255]	
mb *x*	Command both motors backward with raw power x [0..255]	
ml *x*	Set left motor power to x [0..255], right is set to 0	
mr *x*	Set right motor power to x [0..255], left is set to 0	
mh 0	Halt motors	
ms *x*	Spin clockwise; left motor set to forward, right to backward and both powers to x [0..255]	
mc *x*	Spin counterclockwise; left motor set to backward, right to forward and both powers to x [0..255]	
bme	Enable motor driver chip	
bm	Disable motor driver chip	
bke	Enable BEMF sensing system	
bkd	Disable BEMF sensing system	
sf *x*	Assign internal speed correction factor from 50% to 150% [50..150]	0.96

[Configuration Block]

Command	Description	Ver
cw	Write config block from RAM into flash	
cr	Read config block from flash into RAM (overwriting current copy)	
cd2	Display config block info	
cf!	Reload factory configuration, lose all user config and statistics	0.82

[Standby/Sleep]

Command	Description	Ver
zd	Display awake and sleep durations	
za *x*	Set awake timeout to x seconds [10..2^{32}-1]	
zs *x*	Auto-wakeup x seconds in the future (0 = never, otherwise [1..2^{32}-1]. Paid attention to by the zz command.	
zz *x*	Go to sleep and wakeup in x seconds if x>0, else wakeup in zs seconds [0..2^{32}-1]	

[Voltage Commands]

Command	Description	Ver
vd	Display voltage setpoints and coefficients	
vp1	Display raw power system info	
vp2	Display charging statistics	
vl *x*	Set low battery threshold to x in hundredths of a volt [675..725]	
vc *x*	Set critical battery threshold to x in hundredths of a volt [625..675]	

[Debugging]

Each of these commands takes one optional parameter x where x=0 disables the debugging info and x>0 enables it. Entering the command without a parameter is also a way to enable it.

Command	Subsystem/sensor	Ver
da *x*	Accelerometer	
do *x*	IMU orientation	
dv *x*	Power system	
dz	All debugging disabled	

[Gyro Sensor]

Command	Description	Ver
gm	Display minimum and maximum gyro readings	
gmz	Clear minimum and maximum gyro readings	

[Accelerometer Sensor]

Command	Description	Ver
am	Display accumulated min/max accelerometer data	
amz	Clear accumulated min/max accelerometer data	
agr	Display current accelerometer range	
asr *x*	Temporarily reset the accelerometer range: 0 for 2g, 1 for 4g, 2 for 8g and 3 for 16g	1.16

[Bluetooth]

Command	Description	Ver
be	Enable module power	
bd	Disable module power	
bz	Assign new random color name	0.82
ba	Display MAC address	

[Collision Detection]

Command	Description	Ver
id	Display collision thresholds	1.10
ie *x*	Enable collision detection if x>0, else disable	1.10
ih *x*	Set post-collision inhibit time (milliseconds)	1.10
ixt	Set X axis collision base threshold	1.10
ixs	Set X axis collision speed coefficient	1.10
iyt	Set Y axis collision base threshold	1.10
iys	Set Y axis collision speed coefficient	1.10

[Macros]

Command	Description	Ver
mac *x*	Run macro ID x	
kil	Abort any active macro	
maci	Initialize the executive and erase any persistent macros	

[orbBasic]

Command	Description	Ver
basmru	Display the line number MRU table	1.15
basstats	Display various stats about the current program	1.15
rrun *x*	Run a program from RAM starting at line x	1.15
frun *x*	Run a program from flash starting at line x	1.15
break	Abort any running program	1.15
hash *x*	Generate a 32-bit hash of the provided string	1.15
rload	Begin loading a program from the shell into RAM (experimental)	1.21
rlist	List the RAM program	1.21

Revision History

Revision	Date	Who	Description
1.0	9 Aug 2012	Dan Danknick	Initial revision (converted from old format)