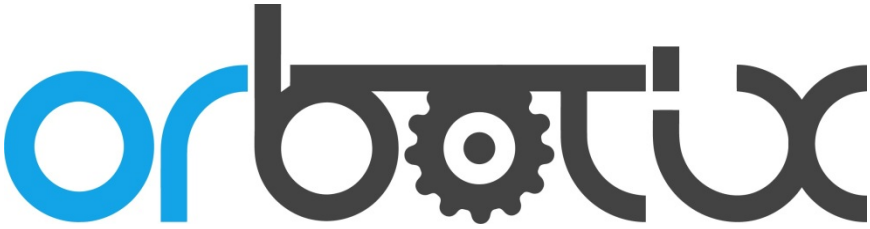
Sphero Locator

Revision 1.1



Introduction 3
Setting Data Streaming 3
Interpreting Locator Data 3
Reporting ..... **Error! Bookmark not defined.**
Interpreting the Reported values ..... **Error! Bookmark not defined.**
Appendix A 6
Configure Collision Detection – 12h ..... **Error! Bookmark not defined.**
Revision History 7

Introduction

Sphero Locator is a firmware feature that provides real-time position and velocity information about the robot. To access it you must use the Set Data Streaming command (See Appendix B). When the Locator X, Locator Y, Velocity X, and Velocity Y bits are set Sphero will stream back position in centimeters and velocity in millimeters per second (at the SDK level, the units are centimeters and centimeters/second). Internally, the calculations are done in floating point at higher precision.

In this document we expand on and clarify the documentation for the Configure Locator command (Appendix A), which we take as a prerequisite. We discuss how to configure the locator, how to interpret the data, and how to write applications that use the locator's coordinate system

Setting Data Streaming

In the Set Data Streaming command, we recommend 20 <= N <= 50 and M=1 for most purposes (Again, see Appendix B). Since the maximum sensor sampling rate is ~420 Hz, if we take N=20 and M=1 we get approx. 420/20 = ~21 packets/second each containing one set of position and velocity data. For iOS devices N=20 works well. For many Android devices N = 10 or less is possible (42+ samples/second).

For real time applications setting M > 1 is usually pointless since you are only interested in the most recent data. However it is possible to obtain all the samples by setting, for instance, N=1 and M=21 (~20 packets/second each containing 21 sets of position/velocity data).

Interpreting Locator Data

The locator treats the ground as a 2d plane and provides Sphero's position in X,Y coordinates. By default, roll heading 0 points down the positive Y-axis with positive X to the right. So, If you shake Sphero awake and send a roll command with heading 0, you will see the Sphero's Y coordinate counting up. If you then send a roll command with heading 90, Sphero's X coordinate will count up.

There are two API commands that affect the locator coordinate system. Most obviously, the Configure Locator command (see Appendix A) allows you to set the position of Sphero and rotate the locator coordinate system with respect to roll headings. For instance, if you would prefer that heading 0 corresponds to the positive X-axis (instead of Y) you could achieve this with the Configure Locator command by setting the yaw tare to 90.

The Set Heading command effects locator coordinates in a different way. Assume no Configure Locator commands are sent. Then heading 0 corresponds to the positive Y-axis. The Set Heading command changes the meaning of "heading 0". There are two options:

1. The locator Y-axis continues to correspond to heading 0 but points in a different real-world direction. (yaw tare is unchanged, see the discussion of yaw tare in Appendix A)
2. The locator Y-axis continues to point in the same real-world direction but corresponds to a different heading angle. (yaw tare is modified).

Depending on your particular needs, one of these behaviors may be more useful/intuitive. We think of the first option as "no correction" and the second as "auto-correction". You can turn this feature on and

off with the configure Locator command. If the first flag bit is set to true auto-correction is turned on. This is the default configuration.

Making the Most of the Locator

Depending on your needs there are a few ways to set up and use the Locator.

1. Do you need to know position (go to 2) or just distance traveled (go to 3)?
2. Do you want to use roll commands to move Sphero to specific locator positions (go to 4) or does it suffice to simply know where Sphero is (go to 5)?
3. Do you want to know an arc length/odometer distance (go to 6) or do you want to the distance "as the crow flies" (go to 5)?
4. See The Full Setup.
5. See The Default Setup.
6. See Distance Traveled.

The Default Setup

Sign up for locator position streaming. Leave the locator in its default configuration with flag 0 set to true. The application may use the standard joystick and calibration UI elements freely.

Distance Traveled

Sign up for locator velocity streaming. Leave the locator in its default configuration with flag 0 set to true. The application may use the standard joystick and calibration UI elements freely. If you want to keep track of distance in the odometer sense, when a data streaming callback comes in accumulate a total using the following formula:

$$Distance\ Traveled\ +=\ \sqrt{V_x^2 + V_y^2} * dt$$

Where V_x and V_y are the x and y velocity components and dt is the time between data samples. Rather than actually measuring dt (and fighting with Bluetooth latency to get a good timestamp) it is better to synthesize it using the formula:

$$dt = \frac{N}{420}$$

Where N is the divisor used in the Set Data Streaming command (see Appendix B).

The Full Setup

The easiest way to use locator coordinates and roll command headings together in your app is to turn auto-correction **off** (set the first flag bit to false) and avoid the Set Heading command altogether. You may freely use the Configure Locator command to set the location of Sphero as long as you always set yaw tare to zero. Again, do **not** use the Set Heading command!

Your app may freely use the standard joystick UI element. Unfortunately, the calibration UI element uses the Set Heading command and so should be avoided. To achieve the same effect one can use Roll

commands with a speed of zero to rotate the ball. Of course, this does not change the meaning of yaw angles the same way Set Heading does. In order to simulate this effect, you must maintain a private yaw offset, which you set when reorienting the ball. Your joystick (or other control mechanism) must be modified to add this yaw offset to every outgoing roll command.

In this way, the user gets all the functionality of the set heading command without altering the locator coordinate system or the IMU headings. If you want to drive from one location toward another the following formula gives the heading you should use

$$\emptyset_{heading} = \left(\frac{\pi}{2}\right) - \text{atan2}\left(y_{goto} - y_{sphero}, x_{goto} - x_{sphero}\right).$$

This angle is in radians. In order to use it in a roll command you must convert it to integer degrees in the range [0,359].

Appendix A

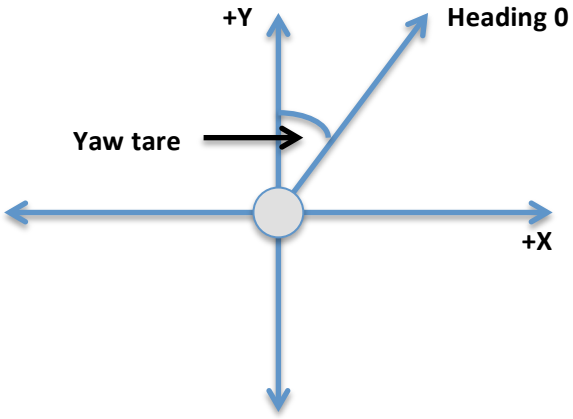
Configure Locator – 13h

Command:

DID	CID	SEQ	DLEN	Flags	X	Y	Yaw Tare
02h	13h	<any>	02h	<8 bit val>	<16 bit signed val>	<16 bit signed val>	<16 bit signed val>

Response: **Simple Response**

Through the streaming interface, Sphero provides real-time location data in the form of (X,Y) coordinates on the ground plane. When Sphero wakes up it has coordinates (0,0) and heading 0, which corresponds to facing down the positive Y-axis with the positive X-axis to your right. This command allows you to move Sphero to a new location and change the alignment of locator coordinates with IMU headings.



When Sphero receives a Set Heading command it changes which direction corresponds to heading 0. By default, the locator compensates for this by modifying its value for yaw tare so that the Y-axis is still pointing in the same real-world direction. For instance, if you wake up Sphero and drive straight, you will be driving down the Y-axis. If you use the Set Heading feature in the drive app to turn 90 degrees, you will still have heading 0, but the locator knows you have turned 90 degrees and are now facing down the X-axis. This feature can be turned off, in which case the locator knows nothing about the Set Heading command. This can lead to some strange results. For instance, if you drive using only roll commands with heading 0 and set heading commands to change direction the locator will perceive your entire path as lying on the Y-axis.

Parameters	Description
Flags	Bit 0 – Determines whether calibrate commands automatically correct the yaw tare value. When false, the positive Y axis coincides with heading 0 (assuming you do not change the yaw tare manually using this API command). Other Bits - Reserved
X, Y	The current (X,Y) coordinates of Sphero on the ground plane in centimeters.
Yaw Tare	Controls how the X,Y-plane is aligned with Sphero's heading coordinate system. When this parameter is set to zero, it means that having yaw = 0 corresponds to facing down the Y-axis in the positive direction. The value will be interpreted in the range 0-359 inclusive.

Appendix B

Set Data Streaming – 11h

Command:

DID	CID	SEQ	DLEN	N	M	MASK	PCNT	MASK2
02h	11h	<any>	0ah or 0eh	16-bit val	16-bit val	32-bit val	8-bit val	32-bit val

Response: **Simple Response**

Sphero supports asynchronous data streaming of certain control system and sensor parameters. This command selects the internal sampling frequency, packet size, parameter mask and optionally, the total number of packets.

param	description
N	Divisor of the maximum sensor sampling rate
M	Number of sample frames emitted per packet
MASK	Bitwise selector of data sources to stream
PCNT	Packet count 1-255 (or 0 for unlimited streaming)
MASK2	Bitwise selector of more data sources to stream (optional)

MASK and PCNT are pretty obvious but the N, M terms bear a little more explanation. Currently the control system runs at 400Hz and because it's pretty unlikely you will want to see data at that rate, N allows you to divide that down. N = 2 yields data samples at 200Hz, N = 10, 40Hz, etc. Every data sample consists of a "frame" made up of the individual sensor values as defined by the MASK. The M value defines how many frames to collect in memory before the packet is emitted. In this sense, it controls the latency of the data you receive. Increasing N and the number of bits set in MASK drive the required throughput. You should experiment with different values of N, M and MASK to see what works best for you.

The MASK2 bitfield was added to extend MASK when we developed more than 32 data sources. The API processor is implemented so that this value is optional; if it isn't included then all of its bits are set to zero. (Added in FW 1.15)

Each parameter is returned as a 16-bit signed integer. The table below defines the bits in MASK to those parameters with the indicated ranges and units. If the command is issued with a MASK of zero, then data streaming is disabled.

MASK			
bit	**sensor**	**range**	**units/LSB**
8000 0000h	accelerometer axis X, raw	-2048 to 2047	4mG
4000 0000h	accelerometer axis Y, raw	-2048 to 2047	4mG
2000 0000h	accelerometer axis Z, raw	-2048 to 2047	4mG
1000 0000h	gyro axis X, raw	-32768 to 32767	0.068 degrees
0800 0000h	gyro axis Y, raw	-32768 to 32767	0.068 degrees
0400 0000h	gyro axis Z, raw	-32768 to 32767	0.068 degrees
0200 0000h	Reserved		
0100 0000h	Reserved		
0080 0000h	Reserved		
0040 0000h	right motor back EMF, raw	-32768 to 32767	22.5 cm
0020 0000h	left motor back EMF, raw	-32768 to 32767	22.5 cm
0010 0000h	left motor, PWM, raw	-2048 to 2047	duty cycle
0008 0000h	right motor, PWM raw	-2048 to 2047	duty cycle
0004 0000h	IMU pitch angle, filtered	-179 to 180	degrees
0002 0000h	IMU roll angle, filtered	-179 to 180	degrees
0001 0000h	IMU yaw angle, filtered	-179 to 180	degrees
0000 8000h	accelerometer axis X, filtered	-32768 to 32767	1/4096 G
0000 4000h	accelerometer axis Y, filtered	-32768 to 32767	1/4096 G
0000 2000h	accelerometer axis Z, filtered	-32768 to 32767	1/4096 G
0000 1000h	gyro axis X, filtered	-20000 to 20000	0.1 dps
0000 0800h	gyro axis Y, filtered	-20000 to 20000	0.1 dps
0000 0400h	gyro axis Z, filtered	-20000 to 20000	0.1 dps
0000 0200h	Reserved		
0000 0100h	Reserved		
0000 0080h	Reserved		
0000 0040h	right motor back EMF, filtered	-32768 to 32767	22.5 cm
0000 0020h	left motor back EMF, filtered	-32768 to 32767	22.5 cm
0000 0010h	Reserved 1		
0000 0008h	Reserved 2		
0000 0004h	Reserved 3		
0000 0002h	Reserved 4		
0000 0001h	Reserved 5		

MASK2			
bit	**sensor**	**range**	**units**
8000 0000h	Quaternion Q0	-10000 to 10000	1/10000 Q
4000 0000h	Quaternion Q1	-10000 to 10000	1/10000 Q
2000 0000h	Quaternion Q2	-10000 to 10000	1/10000 Q
1000 0000h	Quaternion Q3	-10000 to 10000	1/10000 Q
0800 0000h	Odometer X	-32768 to 32767	cm
0400 0000h	Odometer Y	-32768 to 32767	cm
0200 0000h	AccelOne	0 to 8000	1 mG
0100 0000h	Velocity X	-32768 to 32767	mm/s
0080 0000h	Velocity Y	-32768 to 32767	mm/s

Revision History

Revision	Date	Who	Description
1.1	8/8/2012	FP	Improved equations. Corrected sampling divisor suggestion.
1.0	7/30/2012	Fabrizio Polo	Initial release.