libmotion Reference

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Document serial number: 000004892117

Document serial number: 000004892117

Table of Contents

Datatypes

SceMotionState

State of the PlayStation®Vita and motion sensor values

Definition

```
#include <motion.h>
struct SceMotionState {
        SceUInt32 timestamp;
        SceFVector3 acceleration;
        SceFVector3 angularVelocity;
        SceUInt8 reserve1[12];

        SceFQuaternion deviceQuat;
        SceUMatrix4 rotationMatrix;
        SceUMatrix4 nedMatrix;

        SceUInt8 reserve2[4];

        SceFVector3 basicOrientation;
        SceUInt64 hostTimestamp;
        SceUInt8 reserve3[36];

        SceUInt8 magnFieldStability;

        SceUInt8 dataInfo;

        SceUInt8 reserve4[3];
};
```

Members

timestamp	The device timestamp of the sensor device (microseconds)
acceleration	The accelerometer readings <x, y, z> (G)
angularVelocity	The gyro sensor readings <x, y, z> (rad/s)
reserve1	A reserved value
deviceQuat	The orientation of the PlayStation®Vita in a quaternion <x, y, z, w>
rotationMatrix	A 4x4 rotation matrix for orientation
nedMatrix	A 4x4 rotation matrix for orientation based on NED (North East Down)
reserve2	A reserved value
basicOrientation	The basic orientation of the PlayStation®Vita in reference to gravity
hostTimestamp	The timestamp of when a data packet is received by the host (microseconds)
reserve3	A reserved value
magnFieldStability	A value representing the magnetic field stability
dataInfo	A bit field containing special information about the current data
reserve4	A reserved value

Description

This structure contains the state of the PlayStation®Vita as calculated from the motion sensors. For details on the libmotion coordinate system, refer to the "libmotion Overview" document.

magnFieldStability takes one of the following values.

Value	Description
SCE_MOTION_MAGNETIC_FIELD_UNSTABLE	Magnetometer sensor output is unstable
SCE_MOTION_MAGNETIC_FIELD_STABLE	Magnetometer sensor output is stable
SCE_MOTION_MAGNETIC_FIELD_VERYSTABLE	Magnetometer sensor output is very stable

See Also

`sceMotionGetState(),sceMotionReset()`

SceMotionSensorState

Current state of the motion sensors

Definition

```
#include <motion.h>
struct SceMotionSensorState {
        SceFVector3 accelerometer;
        SceFVector3 gyro;
        SceUInt8 reserve1[12];

        SceUInt32 timestamp;
        SceUInt32 counter;

        SceUInt8 reserve2[4];

        SceUInt64 hostTimestamp;

        SceUInt8 dataInfo;

        SceUInt8 reserve3[7];

};
```

Members

accelerometer	The accelerometer readings <x, y, z> (G)
gyro	The gyro sensor readings <x, y, z> (rad/s)
reserve1	A reserved value
timestamp	The timestamp of the gyro sensor device (microseconds)
counter	The ID number of the state The count is incremented by one every time a state is calculated
reserve2	A reserved value
hostTimestamp	The timestamp of when a data packet is received by the host (microseconds)
dataInfo	A bit field containing special information about the current data
reserve3	A reserved value

Description

This structure contains the sensor value output after factory calibrations are applied. This data can be used to define an algorithm for calculating the orientation of the PlayStation®Vita. The timestamp included in the data structure is the gyro data timestamp. This means calculation of orientation is possible with integration of gyro sensor readings.

See Also

`sceMotionGetSensorState()`

SceMotionDeviceLocation

Physical location of the motion sensors in relation to the center of the PlayStation®Vita

Definition

```
#include <motion.h>
struct SceMotionDeviceLocation {
        SceFVector3 accelerometer;
        SceFVector3 gyro;
        SceUInt8 reserve1[24];
};
```

Members

accelerometer	The accelerometer sensor position <x, y, z> (mm)
gyro	The gyro sensor position <x, y, z> (mm)
reserve1	A reserved value

Description

This structure contains the physical location of the motion sensors in relation to the center of the PlayStation®Vita. The units are in millimeters with a value range of +/- 300 mm and a resolution of +/- 1 mm.

Note that the physical location of the sensors in relation to the center of the PlayStation®Vita will change along with Development Kit/Testing Kit/retail unit specification changes.

Get Functions

sceMotionGetState

Return the current state and orientation of the PlayStation®Vita

Definition

```
#include <motion.h>
int sceMotionGetState(
        SceMotionState *motionState
);
```

Arguments

motionState A pointer to hold the return data

Return Values

Value	Description
SCE_OK	Normal termination
SCE_MOTION_ERROR_NULL_PARAMETER	The *motionState* pointer is NULL
SCE_MOTION_ERROR_NOT_SAMPLING	sceMotionStartSampling() has not yet been called

Description

This function returns the current state of the PlayStation®Vita as calculated from the motion sensors.

Notes

sceMotionGetState() and sceMotionGetSensorState() cannot be used simultaneously.

sceMotionGetSensorState

Get the state of the motion sensors

Definition

```
#include <motion.h>
int sceMotionGetSensorState(
        SceMotionSensorState *sensorState,
        int numRecords
);
```

Arguments

`*sensorState`	A pointer to hold the return data
`numRecords`	The number of records returned The number of records that can be obtained is 1 to 64

Return Values

Value	Description
`SCE_OK`	Normal termination
`SCE_MOTION_ERROR_NULL_PARAMETER`	The *`*sensorState`* pointer is NULL
`SCE_MOTION_ERROR_NOT_SAMPLING`	`sceMotionStartSampling()` has not yet been called
`SCE_MOTION_ERROR_OUT_OF_BOUNDS`	The *`numRecords`* value is too large

Description

This function returns the unfiltered state of the motion sensors.

The data record is stored in the user's array from low to high, where lower is the newer one and higher is the older one. If the function is called too quickly, the data received may double up from the last time the function was called. For example, if the user asks for 32 data records, but the data has only been sampled 16 times since the last call to the function, then only 16 data records will be new. By using *`sensorState.counter`*, the user can determine whether the data received is new or old data.

Notes

`sceMotionGetState()` and `sceMotionGetSensorState()` cannot be used simultaneously.

sceMotionGetBasicOrientation

Return the current orientation of the PlayStation®Vita in reference to gravity

Definition

```
#include <motion.h>
int sceMotionGetBasicOrientation(
        SceFVector3 *basicOrientation
);
```

Arguments

`*basicOrientation` A pointer to hold the return data

Return Values

Value	Description
`SCE_OK`	Normal termination
`SCE_MOTION_ERROR_NULL_PARAMETER`	The *`*basicOrientation`* pointer is NULL
`SCE_MOTION_ERROR_NOT_SAMPLING`	`sceMotionStartSampling()` has not yet been called

Description

This function returns the orientation of the PlayStation®Vita in reference to gravity. The orientation should be used to understand which axis the screen is facing. From this, the application can decide whether to use portrait or landscape mode for the screen based on gravity.

Notes

If `sceMotionGetState()` or `sceMotionGetSensorState()` has not been called, the data will not be updated.

sceMotionGetDeviceLocation

Get the physical location of the motion sensors in relation to the center of the PlayStation®Vita

Definition

```
#include <motion.h>
int sceMotionGetDeviceLocation(
        SceMotionDeviceLocation *devLocation
);
```

Arguments

**devLocation* A pointer to hold the return data

Return Values

Value	Description
SCE_OK	Normal termination

Description

This function returns the physical location of the motion sensors in relation to the center of the PlayStation®Vita.

Notes

The units are in millimeters, and the max/min value returned will be +/- 300 mm, with a resolution of +/- 1 mm. Note that the physical location of the sensors in relation to the center of the PlayStation®Vita will change along with Development Kit/Testing Kit/retail unit specification changes.

sceMotionGetMagnetometerState

Get the state of the magnetometer sensor sampling

Definition

```
#include <motion.h>
SceBool sceMotionGetMagnetometerState(void);
```

Arguments

None

Return Values

Value	Description
SCE_TRUE	Magnetometer sensor sampling is on.
SCE_FALSE	Magnetometer sensor sampling is off.

Description

This function returns the state of the magnetometer sensor sampling.

Utility Functions

sceMotionStartSampling

Start the sampling of the motion sensors

Definition

```
#include <motion.h>
int sceMotionStartSampling(void);
```

Arguments

None

Return Values

Value	Description
SCE_OK	Normal termination
SCE_MOTION_ERROR_ALREADY_SAMPLING	Motion sensors are already sampling

Description

This function starts the device sampling. It must be called before `sceMotionGetState()` or `sceMotionGetSensorState()`.

Document serial number: 000004892117

sceMotionStopSampling

Stop the sampling of the motion sensors

Definition

```
#include <motion.h>
int sceMotionStopSampling(void);
```

Arguments

None

Return Values

Value	Description
SCE_OK	Normal termination
SCE_MOTION_ERROR_NOT_SAMPLING	Motions sensors have not yet started sampling

Description

This function stops the motion sensor sampling.

sceMotionRotateYaw

Rotate the yaw orientation of the PlayStation®Vita

Definition

```
#include <motion.h>
int sceMotionRotateYaw(
        const float radians
);
```

Arguments

radians The angle to rotate the yaw by

Return Values

Value	Description
SCE_OK	Normal termination

Description

This function allows the user to rotate around the yaw axis and thereby correct the yaw orientation of the PlayStation®Vita.

sceMotionGetTiltCorrection

Get accelerometer tilt correction filter state

Definition

```
#include <motion.h>
SceBool sceMotionGetTiltCorrection(void);
```

Arguments

None

Return Values

Value	Description
SCE_TRUE	Accelerometer tilt correction filter is enabled
SCE_FALSE	Accelerometer tilt correction filter is disabled

Description

This function returns the value for whether the accelerometer tilt correction filter is enabled or disabled. The accelerometer tilt correction filter is enabled (SCE_TRUE) by default.

For details on the accelerometer tilt correction filter, refer to the "Usage Procedure" chapter of the "libmotion Overview" document.

sceMotionSetTiltCorrection

Set accelerometer tilt correction filter

Definition

```
#include <motion.h>
int sceMotionSetTiltCorrection(
        SceBool setValue
);
```

Arguments

setValue	SCE_TRUE: Enables accelerometer tilt correction filter SCE_FALSE: Disables accelerometer tilt correction filter

Return Values

Value	Description
SCE_OK	Normal termination

Description

This function sets the accelerometer tilt correction filter. The accelerometer tilt correction filter is enabled (SCE_TRUE) by default.

For details on the accelerometer tilt correction filter, refer to the "Usage Procedure" chapter of the "libmotion Overview" document.

sceMotionGetDeadband

Get the gyro deadband filter setting value

Definition

```
#include <motion.h>
SceBool sceMotionGetDeadband(void);
```

Arguments

None

Return Values

Value	Description
SCE_TRUE	Gyro deadband filter is enabled
SCE_FALSE	Gyro deadband filter is disabled

Description

This function returns the setting value for the gyro deadband filter. The gyro deadband filter is enabled (SCE_TRUE) by default.

For details on the gyro deadband filter, refer to the "Usage Procedure" chapter of the "libmotion Overview" document.

sceMotionSetDeadband

Set the gyro deadband filter

Definition

```
#include <motion.h>
int sceMotionSetDeadband(
        SceBool setValue
);
```

Arguments

setValue	SCE_TRUE: Enables gyro deadband filter SCE_FALSE: Disables gyro deadband filter

Return Values

Value	Description
SCE_OK	Normal termination

Description

This function sets the gyro deadband filter. The gyro deadband filter is enabled (SCE_TRUE) by default.

sceMotionSetAngleThreshold

Set the angle threshold used in determining the basic orientation output

Definition

```
#include <motion.h>
int sceMotionSetAngleThreshold(
        const float angle
);
```

Arguments

angle The angle (in degrees) to set the threshold to. Can be set with a range of 0 to 45

Return Values

Value	Description
SCE_OK	Normal termination
SCE_MOTION_ERROR_ANGLE_OUT_OF_RANGE	The set angle was out of range

Description

This function sets the angle threshold used in determining the basic orientation output.
For details on the angle threshold, refer to the "libmotion Overview" document.

sceMotionGetAngleThreshold

Gets the angle threshold used in determining the basic orientation output

Definition

```
#include <motion.h>
float sceMotionGetAngleThreshold(void);
```

Arguments

None

Return Values

Value	Description
float	Current angle threshold value (in degrees)

Description

This function returns the current angle threshold value (in degrees).

sceMotionMagnetometerOn

Turn on sampling of the magnetometer sensor

Definition

```
#include <motion.h>
int sceMotionMagnetometerOn(void);
```

Arguments

None

Return Values

Value	Description
SCE_OK	Normal termination

Description

This function is used to turn on magnetometer sensor sampling. Magnetometer sensor sampling is turned off by default. While the sampling is turned off, *nedMatrix* of SceMotionState will output a unit matrix. The SceMotionSensorState magnetometer sensor value will output 0 in all cases.

sceMotionMagnetometerOff

Turn off sampling of the magnetometer sensor

Definition

```
#include <motion.h>
int sceMotionMagnetometerOff(void);
```

Arguments

None

Return Values

Value	Description
SCE_OK	Normal termination

Description

This function is used to turn off magnetometer sampling. Magnetometer sampling is turned off by default. While the sampling is turned off, *nedMatrix* of SceMotionState will output a unit matrix. The SceMotionSensorState magnetometer sensor value will output 0 in all cases.

sceMotionSetGyroBiasCorrection

Enable/disable the gyro bias correction filter

Definition

```
#include <motion.h>
int sceMotionSetGyroBiasCorrection(
        SceBool setValue
);
```

Arguments

setValue	SCE_TRUE: Enables gyro bias correction filter SCE_FALSE: Disables gyro bias correction filter

Return Values

Value	Description
SCE_OK	Normal termination

Description

This function enables/disables the gyro bias correction filter.

This function can be used even if sampling has not been started.

Note that turning off the gyro bias correction filter will result in drifting of the libmotion output. Because of this, libmotion enables the gyro bias correction filter by default. It is highly recommend that this is not changed.

For details on the gyro bias correction filter, refer to the "Usage Procedure" chapter of the "libmotion Overview" document.

sceMotionGetGyroBiasCorrection

Get the gyro bias correction filter setting value

Definition

```
#include <motion.h>
SceBool sceMotionGetGyroBiasCorrection(void);
```

Arguments

None

Return Values

Value	Description
SCE_TRUE	Gyro bias correction filter is enabled
SCE_FALSE	Gyro bias correction filter is disabled

Description

This function gets the current gyro bias correction state.

The gyro bias correction filter is enabled by default. It is highly recommended this setting is used at all times.

For details on the gyro bias correction filter, refer to the "Usage Procedure" chapter of the "libmotion Overview" document.

Reset Functions

sceMotionReset

Reset the PlayStation®Vita orientation

Definition

```
#include <motion.h>
int sceMotionReset(void);
```

Arguments

None

Return Values

Value	Description
SCE_OK	Normal termination

Description

This function resets the PlayStation®Vita orientation. The filter settings/sampling modes are not changed when reset is called.

For the behavior during a reset, refer to the "Usage Procedure" chapter of the "libmotion Overview" document.

Return Codes

Return Codes

List of return codes

Definition

The following are the common libmotion return codes.

As a general rule, the return codes that can occur within a function are listed in its description.

Value	Hexadecimal	Description
SCE_OK	0	No error occurred
SCE_MOTION_ERROR_DATA_INVALID	0x80360200	The argument was outside the range or was invalid
SCE_MOTION_ERROR_READING	0x80360201	An I/O read error occurred
SCE_MOTION_ERROR_NON_INIT_ERR	0x80360202	The library has not been initialized
SCE_MOTION_ERROR_STATE_INVALID	0x80360203	The requested state was invalid
SCE_MOTION_ERROR_CALIB_READ_FAIL	0x80360204	The calibration data cannot be read
SCE_MOTION_ERROR_OUT_OF_BOUNDS	0x80360205	The requested data index was out of bounds
SCE_MOTION_ERROR_NOT_SAMPLING	0x80360206	`sceMotionStartSampling()` has not been called
SCE_MOTION_ERROR_ALREADY_SAMPLING	0x80360207	`sceMotionStartSampling()` has already been called
SCE_MOTION_ERROR_MEM_IN_USE	0x80360208	The library has already been allocated
SCE_MOTION_ERROR_NULL_PARAMETER	0x80360208	A NULL parameter was passed to the function
SCE_MOTION_ERROR_ANGLE_OUT_OF_RANGE	0x8036020A	The set angle is above 45 degrees or below 0 degrees