Touch Service Reference

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Document serial number: 000004892117

Table of Contents

Constants

Touch Panel Port Number Definition

Definition of port numbers of touch panels

Definition

```
#include <touch.h>
#define SCE_TOUCH_PORT_FRONT              0
#define SCE_TOUCH_PORT_BACK               1
#define SCE_TOUCH_PORT_MAX_NUM            2
```

Description

These are the port numbers specified to access touch data with `sceTouchRead()` and `sceTouchPeek()`.

To obtain the front touch data, obtain `SCE_TOUCH_PORT_FRONT`, and to obtain the rear touch data, obtain `SCE_TOUCH_PORT_BACK`.

See Also

`sceTouchRead()`,`sceTouchPeek()`

Definition for Sampling Port Setting

Definition for sampling port setting of the touch panel

Definition

```
#include <touch.h>
#define SCE_TOUCH_SAMPLING_STATE_STOP   0
#define SCE_TOUCH_SAMPLING_STATE_START  1
```

Description

This is the definition value for setting the port for obtaining touch data with `sceTouchSetSamplingState()`.

For details, refer to the description of `sceTouchSetSamplingState()`.

See Also

`sceTouchSetSamplingState()`,`sceTouchGetSamplingState()`

Definition for Touch Status

Definition for *status* field of SceTouchData structure

Definition

```
#include <touch.h>
#define SCE_TOUCH_STATUS_INTERCEPTED      0x00010000
```

Description

This is the definition used in the *status* field of SceTouchData.
For details, refer to the "SceTouchData" section.

See Also

SceTouchReport

Definition for Touch Information Field

Definition for *info* field of SceTouchReport structure

Definition

```
#include <touch.h>
#define SCE_TOUCH_REPORT_INFO_HIDE_UPPER_LAYER  0x0001
```

Description

This is the definition used in the *info* field of SceTouchReport.
For details, refer to the "SceTouchReport" section.

See Also

SceTouchReport

Error Codes

List of error codes

Definition

Macro	Value	Description
SCE_TOUCH_ERROR_INVALID_ARG	0x80350001	Invalid argument specified
SCE_TOUCH_ERROR_PRIV_REQUIRED	0x80350002	Access with invalid privilege
SCE_TOUCH_ERROR_FATAL	0x803500FF	Fatal error occurred

Datatypes

SceTouchPanelInfo

Structure for touch panel information obtainment

Definition

```
#include <touch.h>
typedef struct SceTouchPanelInfo {
        SceInt16 minAaX;
        SceInt16 minAaY;
        SceInt16 maxAaX;
        SceInt16 maxAaY;
        SceInt16 minDispX;
        SceInt16 minDispY;
        SceInt16 maxDispX;
        SceInt16 maxDispY;
        SceUInt8 minForce;
        SceUInt8 maxForce;
        SceUInt8 rsv[30];
} SceTouchPanelInfo;
```

Members

minAaX	Minimum value of X coordinate of active area of touch panel
minAaY	Minimum value of Y coordinate of active area of touch panel
maxAaX	Maximum value of X coordinate of active area of touch panel
maxAaY	Maximum value of Y coordinate of active area of touch panel
minDispX	Touch panel X coordinate of display coordinate origin (top left corner)
minDispY	Touch panel Y coordinate of display coordinate origin (top left corner)
maxDispX	Touch panel X coordinate of display coordinate maximum value (bottom right corner)
maxDispY	Touch panel Y coordinate of display coordinate maximum value (bottom right corner)
minForce	Minimum value of touch force
maxForce	Maximum value of touch force
rsv[30]	Reserved area

Description

This structure is for obtaining the touch panel information.

The touch panel outputs the data of the X and Y coordinates in the range of coordinates obtained in the active area (*minAaX*, *minAaY*, *maxAaX*, *maxAaY*).

(*minDispX*, *minDispY*, *maxDispX*, *maxDispY*) indicates the display position in the touch panel coordinate system.

Minimum value and maximum value of touch force can be obtained with *minForce*/*maxForce*.

Notes

The relation between the active area and display area is not the same for the front touch panel and the rear touch panel.

Before using the touch panel, it is recommended to obtain the information for each touch panel to be used with `sceTouchGetPanelInfo()`.

See Also

`sceTouchGetPanelInfo()`

SceTouchData, SceTouchReport

Data definition for touch data obtainment

Definition

```
#include <touch.h>
#define SCE_TOUCH_MAX_REPORT 8
typedef struct SceTouchReport {
        SceUInt8 id;
        SceUInt8 force;
        SceInt16 x;
        SceInt16 y;
        SceUInt8 rsv[8];
        SceUInt16 info;
} SceTouchReport;

typedef struct SceTouchData {
        SceUInt64 timeStamp;
        SceUInt32 status;
        SceUInt32 reportNum;
        SceTouchReport report[SCE_TOUCH_MAX_REPORT];
} SceTouchData;
```

Members

`SceTouchReport` members

id	Touch report ID
force	Force of touch at ground point
x	X coordinate of touch point
y	Y coordinate of touch point
rsv	Reserved area
info	Touch information field

`SceTouchData` members

timeStamp	Time stamp during touch data obtainment (process time: μ sec)
status	Stores touch data status
reportNum	Returns the number of currently active touch points.
report	Touch report data

Description

This structure is for obtaining the touch data state information.

The time stamp indicating the time (process time) at which the data were obtained is placed in the *timeStamp* argument of `SceTouchData`.

The *status* argument of `SceTouchData` stores the touch data status. Currently, only the `SCE_TOUCH_STATUS_INTERCEPTED` state is defined. For details, refer to the "Touch Data Status" section in the "Touch Data" chapter of the "Touch Service Overview" document.

The number of valid touch reports is placed in the *reportNum* argument of `SceTouchData`. The maximum number of valid touch reports is currently 6 for the front touch panel and 4 for the rear touch panel.

The touch report ID is placed in the *id* argument of `SceTouchReport`. One such ID is allocated each time the touch panel is touched, and the ID is guaranteed to remain unchanged until that touch point is released. The output range of *id* is the range from 0 to 127.

For the *force* argument of SceTouchReport, when enabling touch force information with sceTouchEnableTouchForce(), 0 will be set when there is no touch and 128 will be set when there is a touch regardless of its force. When disabling touch force information with sceTouchDisableTouchForce(), 0 will always be set regardless of touch/no touch. Support for touch force information ended with SDK 2.500. For SDK 2.500 and later, Do not use the *force* value in an application.

The position of the X coordinate of the touch point is placed in the *x* argument of SceTouchReport.

The output range is the range from SceTouchPanelInfo::*minAaX* to SceTouchPanelInfo::*maxAaX*.

The position of the Y coordinate of the touch point is placed in the *y* argument of SceTouchReport.

The output range is the range from SceTouchPanelInfo::*minAaY* to SceTouchPanelInfo::*maxAaY*.

The *info* argument of SceTouchReport is the touch information field.

Currently, only the SCE_TOUCH_REPORT_INFO_HIDE_UPPER_LAYER state is defined.

For details, refer to the "Touch information field" section in the "Touch Data" chapter of the "Touch Service Overview" document.

See Also

sceTouchRead(),sceTouchPeek()

Mode Setting Functions

sceTouchGetPanelInfo

Get touch panel information

Definition

```
#include <touch.h>
int sceTouchGetPanelInfo(
        SceUInt32 port,
        SceTouchPanelInfo* pPanelInfo
);
```

Arguments

port	Front touch panel: SCE_TOUCH_PORT_FRONT Rear touch panel : SCE_TOUCH_PORT_BACK
pPanelInfo	Area for obtaining the touch panel information

Return Values

If an error occurs, a negative value is returned. For details on the error, refer to "Error Codes."

Description

This function obtains the touch panel information of the panel specified in the *port* argument.
For details on the touch panel information, refer to the description of SceTouchPanelInfo.

See Also

SceTouchPanelInfo

sceTouchSetSamplingState

Set sampling state

Definition

```
#include <touch.h>
int sceTouchSetSamplingState(
        SceUInt32 port,
        SceUInt32 state
);
```

Arguments

port	Front touch panel: SCE_TOUCH_PORT_FRONT Rear touch panel: SCE_TOUCH_PORT_BACK
state	Sampling stop: SCE_TOUCH_SAMPLING_STATE_STOP Sampling start: SCE_TOUCH_SAMPLING_STATE_START

Return Values

The setting value of the data sampling state set until now is returned.

If an error occurs, a negative value is returned. For details on the error, refer to "Error Codes."

Description

This function sets the sampling state of the touch panel.

To enable the front touch panel, specify SCE_TOUCH_PORT_FRONT in the *port* argument and specify SCE_TOUCH_SAMPLING_STATE_START in the *state* argument.

To enable the rear touch panel, specify SCE_TOUCH_PORT_BACK in the *port* argument and specify SCE_TOUCH_SAMPLING_STATE_START in the *state* argument.

By default, both the front and the rear touch panels are in the sampling stop state.

See Also

sceTouchGetSamplingState()

sceTouchGetSamplingState

Get value of data obtainment port mask setting of touch panel

Definition

```
#include <touch.h>
int sceTouchGetSamplingState(
        SceUInt32 port,
        SceUInt32* pState
);
```

Arguments

port	Front touch panel: SCE_TOUCH_PORT_FRONT Rear touch panel : SCE_TOUCH_PORT_BACK
pState	Area for obtaining the current sampling state Obtainment is possible for the following states. Sampling stop: SCE_TOUCH_SAMPLING_STATE_STOP Sampling start: SCE_TOUCH_SAMPLING_STATE_START

Return Values

If an error occurs, a negative value is returned. For details on the error, refer to "Error Codes."

Description

This function obtains the sampling state of the touch panel that is currently set.

The setting can be changed with the sceTouchSetSamplingState() function.

See Also

sceTouchSetSamplingState()

sceTouchEnableTouchForce

Enable the force output of touch panel (not recommended)

Definition

```
#include <touch.h>
int sceTouchEnableTouchForce(
        SceUInt32 port
);
```

Arguments

port	Front touch panel: SCE_TOUCH_PORT_FRONT Rear touch panel: SCE_TOUCH_PORT_BACK

Return Values

If an error occurs, a negative value is returned. For details on the error, refer to "Error Codes."

Description

Support for touch force information ended with SDK 2.500. Do not call this API.

See Also

sceTouchDisableTouchForce()

sceTouchDisableTouchForce

Disable the force output of touch panel (not recommended)

Definition

```
#include <touch.h>
int sceTouchDisableTouchForce(
        SceUInt32 port
);
```

Arguments

port	Front touch panel: SCE_TOUCH_PORT_FRONT Rear touch panel: SCE_TOUCH_PORT_BACK

Return Values

If an error occurs, a negative value is returned. For details on the error, refer to "Error Codes."

Description

Support for touch force information ended with SDK 2.500. Do not call this API.

See Also

sceTouchEnableTouchForce()

Data Obtainment Functions

sceTouchRead

Get touch data by blocking

Definition

```
#include <touch.h>
int sceTouchRead(
        SceUInt32 port,
        SceTouchData* pData,
        SceUInt32 nBufs
);
```

Arguments

port	Front touch panel: SCE_TOUCH_PORT_FRONT Rear touch panel: SCE_TOUCH_PORT_BACK
pData	Buffer to receive touch data
nBufs	Number of buffers to receive touch data (1 to 64)

Return Values

The number of sets of touch data that were returned to *pData* is returned. The value is in the range of 1 to *nBufs*.

If an error occurs, a negative value is returned. For details on the error, refer to "Error Codes."

Description

This function obtains the touch data by blocking.

Up to 64 of the most recent sets of touch data are held in the buffers in the touch service, and the sceTouchRead() function obtains touch data sets of the number specified by the *nBufs* argument, starting from the current position of the read pointer, and then updates the read pointer.

If sampling of the touch data was not performed even once during the period from the last time the sceTouchRead() function was called until the current call, then the thread is blocked in the sceTouchRead() function until the touch data is sampled again. Consequently, the return value is always 1 or greater, and the most recent data is obtained.

When the sceTouchRead() function cannot be called at each set sampling period due to a processing delay, etc., by setting the value of *nBufs* to 2 or greater, it is possible to ascertain whether or not there was a processing delay based on whether or not the return value is greater than 1.

The touch service performs touch data sampling at the VSYNC period. Since SceTouchData::*reportNum* is 0 and no valid touch data is included in SceTouchData::*report* when no touch point is detected on the touch panel, be careful not to assume that touch data is always included in the data obtained using the sceTouchRead() function.

Notes

Differences with Similar Functions

With the `sceTouchPeek()` function, touch data is obtained by snooping, so if the function is called earlier than the sampling period, the latest sampling results are repeatedly obtained without blocking the thread.

Behavior when this function is called while touch data sampling is not enabled

When this function is called while touch data sampling is not enabled by `sceTouchSetSamplingState()`, data is obtained with `SceTouchData::`*`reportNum`* as '0'.

See Also

`SceTouchData`, `sceTouchPeek()`

sceTouchPeek

Get touch data by polling

Definition

```
#include <touch.h>
int sceTouchPeek(
        SceUInt32 port,
        SceTouchData* pData,
        SceUInt32 nBufs
);
```

Arguments

port	Front touch panel: SCE_TOUCH_PORT_FRONT Rear touch panel: SCE_TOUCH_PORT_BACK
pData	Buffer to receive touch data
nBufs	Number of buffers to receive touch data (1 to 64)

Return Values

The number of sets of touch data that were returned to *pData* is returned. The value is in the range of 1 to *nBufs*.

If an error occurs, a negative value is returned. For details on the error, refer to "Error Codes."

Description

This function obtains the touch data by polling.

Up to 64 of the most recent sets of touch data are held in the buffers in the touch service, and the sceTouchPeek() function obtains touch data sets of the number specified by the *nBufs* argument, starting from the current position of the read pointer, and without updating the read pointer.

The contents of the buffers are updated by interrupt every time touch data is sampled. For this reason, note that, depending on the timing at which the sceTouchPeek() function is called, the data may be obtained either before or after the buffers are routinely updated at the VSYNC period.

Notes

Differences with Similar Functions

With the sceTouchRead() function, touch data is obtained by blocking, so if the function is called earlier than the sampling period, the latest information is obtained by blocking the thread.

Behavior when this function is called while touch data sampling is not enabled

When this function is called while touch data sampling is not enabled by sceTouchSetSamplingState(), data is obtained with SceTouchData::*reportNum* as '0'.

See Also

SceTouchData, sceTouchRead()