

TNGames C/C++ SDK Version 1.5

1. API Functions

i) SetUpJacket

Set up the connection with the gaming jacket.

Int SetUpJacket(void);

Parameters

None.

Return Values

If the function succeeds, the return value is zero (GLIB_OK) .

If the function fails, the return value is nonzero.

To get extended error information, call **GetErrorText();**

Remarks

If successful will return zero (GLIB_OK) all other return values mean failure.

Requirements

Header: Declared in tngaming.lib

Import Library: Use tngaming.lib

ii) TearDownJacket

Void TearDownJacket(void);

Parameters

None.

Return Values

If the function succeeds, the return value is zero (GLIB_OK) .

If the function fails, the return value is nonzero.

To get extended error information, call **GetErrorText();**

Remarks

Disconnect from the device and de-allocate any resources.

Requirements

Header: Declared in tngaming.h

Import Library: Use tngaming.lib

iii) SetEffect

Int SetEffect (int nEffect);

Parameters

nEffect

Specifies a tactile effect. These codes define how the jacket will actuate. **See section 2.**

Return Values

If a failure occurs the return value will specify the error.

To get extended error information, call **GetErrorText();**

Remarks

Send a predefined effect to the gaming jacket. Returns zero (GLIB_OK) on success. If a failure occurs the return value will specify the error.

Requirements

Header: Declared in tngaming.h

Import Library: Use tngaming.lib

iv) SetEffect2

Int SetEffect2(int speed, int actuator);

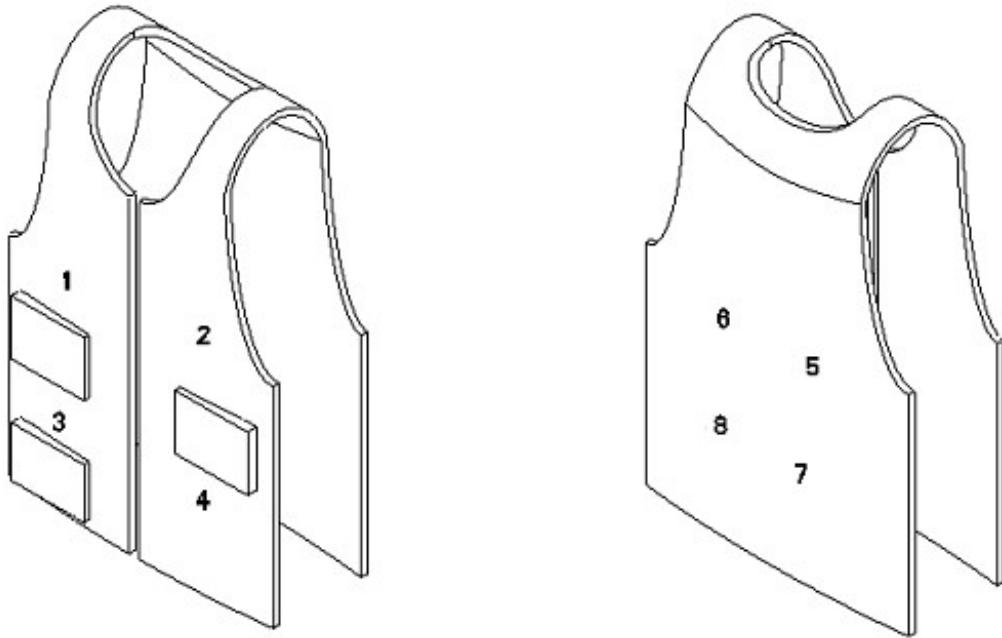
Parameters

speed

Specifies the length of a tactile effect. The value must be in the range of 1 – 255. All other values will be rejected.

actuator

Specifies the area on the jacket to actuate. The value must be in the range of 1 – 8. All other values will be rejected. The area id's are:



Return Values

If a failure occurs the return value will specify the error.

To get extended error information, call **GetErrorText()**;

Remarks

Send a custom effect to the gaming jacket. Returns zero (GLIB_OK) on success. If a failure occurs the return value will specify the error.

Requirements

Header: Declared in tngaming.h

Import Library: Use tngaming.lib

v) GetErrorCode

```
Int GetErrorCode(void);
```

Parameters

None.

Return Values

Returns the last error code. Zero indicates no error.

To get extended error information, call **GetErrorText()**;

Remarks

Returns the last error code. Zero indicates no error.

Requirements

Header: Declared in tngaming.h

Import Library: Use tngaming.lib

vi) GetErrorText

Char* GetErrorText(void);

Parameters

None.

Return Values

Returns a pointer containing the location of the error code text.

Remarks

Returns a pointer containing the location of the error code text.

Requirements

Header: Declared in tngaming.h

Import Library: Use tngaming.lib

vii) FlushBuffer

void FlushBuffer(int actuator);

Parameters

actuator

Specifies the actuators buffer that will be cleared.

Return Values

None.

Requirements

Header: Declared in tngaming.h

Import Library: Use tngaming.lib

2. API Predefined Effects

Version 1.5 of TNGames tactile library contains the following effect constants:

Tactile Effect	Description
----------------	-------------

E_MACHINEGUN_FRONT	Simulates machine gun fire to the front.
E_MACHINEGUN_BACK	Simulates machine gun fire to the back.
E_BIG_BLAST_FRONT	Simulates large explosion to the front.
E_BIG_BLAST_BACK	Simulates large explosion to the back.
E_SMALL_BLAST_FRONT	Simulates small explosion to the front.
E_SMALL_BLAST_BACK	Simulates small explosion to the front.
E_PISTOL_FRONT	Simulates handgun fire to the front.
E_PISTOL_BACK	Simulates handgun fire to the back.
E_PUNCH_FRONT	Simulates punch to the front.
E_PUNCH_BACK	Simulates punch to the back.
E_STAB_FRONT	Simulates stab to the front.
E_STAB_BACK	Simulates stab to the back.
E_SHOTGUN_FRONT	Simulates shotgun fire to the front.
E_SHOTGUN_BACK	Simulates shotgun fire to the back.
E_RIFLE_FRONT	Simulates rifle fire to the front.
E_RIFLE_BACK	Simulates rifle fire to the back.
E_LEFT_SIDE_HIT	Simulates explosion to the left side.
E_RIGHT_SIDE_HIT	Simulates explosion to the right side.
E_ACCELERATION	Simulates acceleration.
E_DECELERATION	Simulates deceleration.
E_LEFTTURN	Simulates turning left
E_RIGHTTURN	Simulates turning right
E_ACCELERATION_STOP	Stops the acceleration command.
E_DECELERATION_STOP	Stops the deceleration command.
E_LEFTTURN_STOP	Stops the turning left command.
E_RIGHTTURN_STOP	Stops the turning right command.

3. Integration And Examples

```

/* --- include file */
#include "tngaming.h"

/* --- link library */
#pragma comment(lib,"tngaming.lib")

int main(int argc, char* argv[])
{
    /* Initailized gaming library */
    if (SetUpJacket() != GLIB_OK)
    {
        printf("Error: %s", GetErrorText());
    }

    /* Send a predefined effect */
    if (SetEffect(E_MACHINEGUN_FRONT) != GLIB_OK)
    {
        printf("Error: %s", GetErrorText());
    }

    /* Send a custom effect

```

```
    length = 10 // fast
    cell = 1 // Top left front

    */
    if (SetEffect2(10,1) != GLIB_OK)
    {
        printf("Error: %s", GetErrorText());
    }

/* clean up and exit from gaming library */
TearDownJacket();

return 0;
}
```