

# LSI-3104

## Encoder/ Linear Scale Interface Card

### Software Manual (V1.2)

健昇科技股份有限公司

**JS AUTOMATION CORP.**

台北縣汐止市中興路 100 號 6 樓

6F, No. 100, Chungshin Rd.

Shitsu, Taipei, Taiwan, R.O.C.

TEL : 886-2-2647-6936

FAX : 886-2-2647-6940

<http://www.automation.com.tw>

E-mail : [control.cards@automation.com.tw](mailto:control.cards@automation.com.tw)

# Correction record

Version	Record

# Contents

1.	How to install the software of LSI3104.....	4
1.1	Install the PCI driver.....	4
1.2	How to install the utilities.....	4
2.	About the LSI-3104 software .....	5
2.1	What you need to get started.....	5
2.2	Software programming choices .....	5
3.	LSI3104 Language support.....	6
3.1	Building applications with the LSI3104 software library.....	6
3.2	LSI3104 Windows libraries .....	6
4.	Software overview .....	7
4.1	Initialization function.....	7
4.2	Homing function.....	7
4.3	Counter function .....	7
4.4	Input/Output function .....	8
4.5	Security function.....	8
4.6	Miscellaneous function .....	8
4.7	Compare function(software version V1.42 and after) .....	9
5.	Flow chart of implement an application .....	10
5.1	LSI3104 Flow chart of implementation.....	10
6.	Function reference .....	12
6.1	Error codes and address .....	12
6.2	Variable data types .....	13
6.3	Programming language considerations .....	14
6.4	LSI-3104 Function.....	16
	Initialization function.....	16
	lsi3104_initial .....	16
	lsi3104_close .....	16
	lsi3104_info .....	17
	Homing function.....	18
	lsi3104_set_hard_homing.....	19
	lsi3104_read_hard_homing_flag .....	19
	lsi3104_soft_homing_command.....	19
	Counter function .....	20
	lsi3104_set_counter_mode .....	21
	lsi3104_set_quadrature_times .....	21
	lsi3104_read_counter.....	22
	lsi3104_load_counter.....	22
	lsi3104_preset_counter.....	23
	lsi3104_latch_control.....	23

lsi3104_latch_mode.....	24
lsi3104_read_latch_flag.....	24
lsi3104_read_latched_value.....	25
Input/Output function .....	26
lsi3104_read_input_status .....	27
lsi3104_set_input_polarity .....	28
lsi3104_read_input_polarity .....	29
lsi3104_set_clr_out_mode .....	29
lsi3104_write_output_command .....	30
lsi3104_set_output_polarity.....	30
lsi3104_read_output_status .....	31
lsi3104_read_output_polarity .....	31
Security function.....	32
lsi3104_set_password .....	32
lsi3104_change_password .....	32
lsi3104_clear_password.....	32
lsi3104_unlock_security .....	33
lsi3104_read_security_status .....	33
Miscellaneous function .....	34
lsi3104_com_trigger_control.....	34
lsi3104_read_parameters .....	35
Compare function (software version V1.42 and after) .....	36
lsi3104_set_compare_out_mode(only for x and z axis).....	36
lsi3104_load_compare_value (only for x and z axis).....	36
lsi3104_read_compare_value (only for x and z axis).....	37
lsi3104_load_increase_value (only for x and z axis).....	37
lsi3104_read_increase_value (only for x and z axis).....	38
6.5 Dll list .....	39
7. LSI-3104 Error codes summary.....	40
7.1 LSI3104 Error codes table .....	40

# 1. **How to install the software of LSI3104**

## 1.1 Install the PCI driver

The PCI card is a plug and play card, once you add a new card on the window system will detect while it is booting. Please follow the following steps to install your new card.

In Windows 98/2000 system you should: (take win98 as example)

1. Make sure the power is off
2. Plug in the interface card
3. Power on
4. A hardware install wizard will appear and tell you it finds a new PCI card
5. Tell the wizard the directory of the driver files (\LSI3104\win98\driver), then it will automatically setup the driver
6. After installation, power off
7. Power on , it's ready to use

## 1.2 How to install the utilities

Execute the file \install\setup3104.exe will automatically install all the utilities and setup a new For a more detail descriptions, please visit our user club

<http://automation.com.tw> and register as a member then download the file "Installation" which will take you go through step by step.

## 2. **About the LSI-3104 software**

LSI3104 software includes a set of dynamic link library (DLL) and system driver that you can utilize to control the interface card's ports and points separately.

Your LSI3104 software package includes setup driver, tutorial example and test program that help you how to setup and run appropriately, as well as an executable file which you can use to test each of the LSI3104 functions within Windows' operation system environment.

### 2.1 What you need to get started

To set up and use your LSI3104 software, you need the following:

- LSI3104 software
- LSI3104 hardware
  - Main board
  - Wiring board (Option)

### 2.2 Software programming choices

You have several options to choose from when you are programming LSI3104 software. You can use Borland C/C++, Microsoft Visual C/C++, Microsoft Visual Basic, or any other Windows-based compiler that can call into Windows dynamic link libraries (DLLs) for use with the LSI3104 software.

### 3. **LSI3104 Language support**

The LSI3104 software library is a DLL used with Windows 98/2000/XP. You can use these DLL with any Windows integrating development environment that can call Windows DLLs.

#### 3.1 Building applications with the LSI3104 software library

The LSI3104 function reference topic contains general information about building LSI3104 applications, describes the nature of the LSI3104 files used in building LSI3104 applications, and explains the basics of making applications using the following tools:

##### **Applications tools**

- ◆ **Borland C/C++**
- ◆ **Microsoft Visual C/C++**
- ◆ **Microsoft Visual Basic**

If you are not using one of the tools listed, consult your development tool reference manual for details on creating applications that call DLLs.

#### 3.2 LSI3104 Windows libraries

The LSI3104 for Windows function library is a DLL called **LSI3104.dll**. Since a DLL is used, LSI3104 functions are not linked into the executable files of applications. Only the information about the LSI3104 functions in the LSI3104 import libraries is stored in the executable files.

Import libraries contain information about their DLL-exported functions. They indicate the presence and location of the DLL routines. Depending on the development tools you are using, you can make your compiler and linker aware of the DLL functions through import libraries or through function declarations.

Refer to **Table 1** to determine to which files you need to link and which to include in your development to use the LSI3104 functions in LSI3104.dll.

Header Files and Import Libraries for Different Development Environments		
<b>Development Environment</b>	<b>Header File</b>	<b>Import Library</b>
<b>Microsoft C/C++</b>	LSI3104.h	LSI3104VC.lib
<b>Borland C/C++</b>	LSI3104.h	LSI3104BC.lib
<b>Microsoft Visual Basic</b>	LSI3104.bas	

**Table 1**

## 4. Software overview

These topics describe the features and functionality of the LSI3104 boards and briefly describes the LSI3104 functions.

### 4.1 Initialization function

<a href="#">lsi3104_initial</a> (void)	Software initializaton
<a href="#">lsi3104_close</a> (void)	Software close
<a href="#">lsi3104_info</a> (u8 CardID, u16 *address)	Get card information

### 4.2 Homing function

<a href="#">lsi3104_set_hard_homing</a> (u8 CardID, u8 axis, u8 mode)	Set homing mode
<a href="#">lsi3104_read_hard_homing_flag</a> (u8 CardID, u8 axis, u8 *flag)	Read homing flag
<a href="#">lsi3104_soft_homing_command</a> (u8 CardID, u8 axis)	Software homing command

### 4.3 Counter function

<a href="#">lsi3104_set_counter_mode</a> (u8 CardID, u8 axis, u8 mode)	Set counter mode
<a href="#">lsi3104_set_quadrature_times</a> (u8 CardID, u8 axis, u8 times)	Set decoding rate
<a href="#">lsi3104_read_counter</a> (u8 CardID, u8 axis, i32 *value)	Read encoder counter
<a href="#">lsi3104_load_counter</a> (u8 CardID, u8 axis, i32 load_value)	Load value to counter
<a href="#">lsi3104_preset_counter</a> (u8 CardID, u8 axis, i32 preset_value)	Preset counter value
<a href="#">lsi3104_latch_control</a> (u8 CardID, u8 axis, u8 control)	Enable/disable latch function
<a href="#">lsi3104_latch_mode</a> (u8 CardID, u8 axis, u8 mode)	Set latch/load mode
<a href="#">lsi3104_read_latch_flag</a> (u8 CardID, u8 axis, u8 *flag)	Read latch/load flag
<a href="#">lsi3104_read_latched_value</a> (u8 CardID, u8 axis, i32 *value)	Read latch value

#### 4.4 Input/Output function

<a href="#">lsi3104_read_input_status</a> (u8 CardID, u8 axis, u8 point, u8 *state)	Read input point status
<a href="#">lsi3104_set_input_polarity</a> (u8 CardID, u8 axis, u8 point, u8 polarity)	Set input point polarity
<a href="#">lsi3104_read_input_polarity</a> (u8 CardID, u8 axis, u8 point, u8 *polarity)	Input point polarity readback
<a href="#">lsi3104_set_clr_out_mode</a> (u8 CardID, u8 axis, u8 mode)	Set clear output mode
<a href="#">lsi3104_write_output_command</a> (u8 CardID, u8 axis, u8 point, u8 on_off)	Write output point command
<a href="#">lsi3104_set_output_polarity</a> (u8 CardID, u8 axis, u8 point, u8 polarity)	Set output point polarity
<a href="#">lsi3104_read_output_status</a> (u8 CardID, u8 axis, u8 point, u8 *state)	Output point status readback
<a href="#">lsi3104_read_output_polarity</a> (u8 CardID, u8 axis, u8 point, u8 *polarity)	Output point polarity readback

#### 4.5 Security function

<a href="#">lsi3104_set_password</a> (u8 CardID, u16 password[5])	Set password
<a href="#">lsi3104_change_password</a> (u8 CardID, u16 Oldpassword[5], u16 Password[5])	Change password
<a href="#">lsi3104_clear_password</a> (u8 CardID, u16 Password[5])	Clear password.
<a href="#">lsi3104_unlock_security</a> (u8 CardID, u16 password[5])	Unlock security function
<a href="#">lsi3104_read_security_status</a> (u8 CardID, u8 *lock_status, u8 *security_enable)	Read security status

#### 4.6 Miscellaneous function

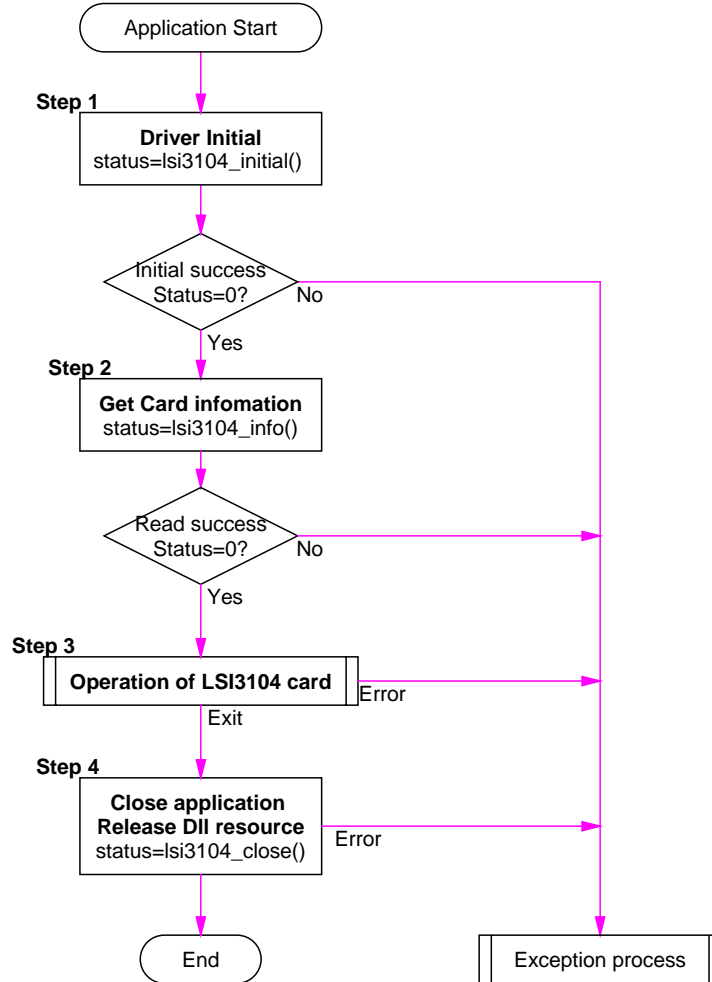
<a href="#">lsi3104_com_trigger_control</a> (u8 CardID, u8 control)	Common trigger control
<a href="#">lsi3104_read_parameters</a> (u8 CardID, u8 axis, u8 parameter_no, u8 *value)	Miscellaneous parameters readback

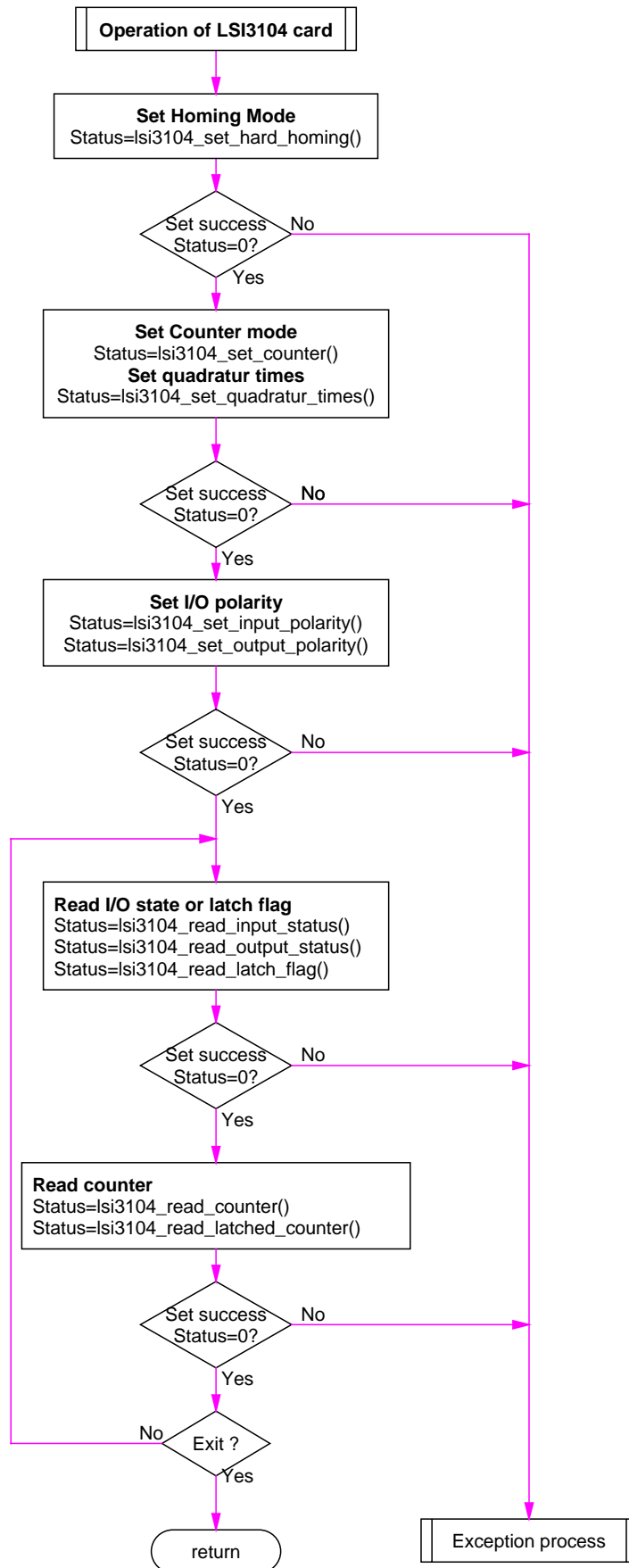
#### 4.7 Compare function(software version V1.42 and after)

<a href="#"><u>lsi3104_set_compare_out_mode</u></a> (u8 CardID,u8 Axis,u8 mode)	Set compare out mode
<a href="#"><u>lsi3104_load_compare_value</u></a> (u8 CardID,u8 Axis,i32 value)	Set compare value
<a href="#"><u>lsi3104_read_compare_value</u></a> (u8 CardID,u8 Axis,i32 *value)	Compare value readback from card
<a href="#"><u>lsi3104_load_increase_value</u></a> (u8 CardID,u8 Axis,i32 value)	Set auto increase value after compare finished.
<a href="#"><u>lsi3104_read_increase_value</u></a> (u8 CardID,u8 Axis,i32 *value)	Increase value readback from card.

## 5. Flow chart of implement an application

### 5.1 LSI3104 Flow chart of implementation





## 6. **Function reference**

These topics contain detailed descriptions of each LSI3104 function. The functions are arranged alphabetically by function name. Refer to LSI3104 Function Reference for additional information.

### 6.1 Error codes and address

Every LSI3104 function is consist of the following format:

**Status = function\_name (parameter 1, parameter 2, ... parameter n)**

Each function returns a value in the **Status** global variable that indicates the success or failure of the function. A returned **Status** equal to zero that indicates the function executed successfully. A non-zero status indicates failure that the function did not execute successfully because of an error, or executed with an error.

**Note** : **Status** is a 32-bit unsigned integer.

The first parameter to almost every LSI3104 function is the parameter **CardID** which is located the driver of LSI3104 board you want to use those given operation. The **CardID** is assigned by DIP SW. You can utilize multiple devices with different card ID within one application; to do so, simply pass the appropriate **CardID** to each function.

**Note**: **CardID** is set by dip switch (**0x0-0xF**)

## 6.2 Variable data types

Every function description has a parameter table that lists the data types for each parameter. The following sections describe the notation used in those parameter tables and throughout the manual for variable data types.

Primary Type Names					
Name	Description	Range	C/C++	Visual BASIC	Pascal (Borland Delphi)
<b>u8</b>	8-bit ASCII character	0 to 255	char	Not supported by BASIC. For functions that require character arrays, use string types instead.	Byte
<b>i16</b>	16-bit signed integer	-32,768 to 32,767	short	Integer (for example: deviceNum%)	SmallInt
<b>u16</b>	16-bit unsigned integer	0 to 65,535	unsigned short for 32-bit compilers	Not supported by BASIC. For functions that require unsigned integers, use the signed integer type instead. See the i16 description.	Word
<b>i32</b>	32-bit signed integer	-2,147,483,648 to 2,147,483,647	long	Long (for example: count&)	LongInt
<b>u32</b>	32-bit unsigned integer	0 to 4,294,967,295	unsigned long	Not supported by BASIC. For functions that require unsigned long integers, use the signed long integer type instead. See the i32 description.	Cardinal (in 32-bit operating systems). Refer to the i32 description.
<b>f32</b>	32-bit single-precision floating-point value	-3.402823E+38 to 3.402823E+38	float	Single (for example: num!)	Single
<b>f64</b>	64-bit double-precision floating-point value	-1.797683134862315E+308 to 1.797683134862315E+308	double	Double (for example: voltage Number)	Double

**Table 2**

### 6.3 Programming language considerations

Apart from the data type differences, there are a few language-dependent considerations you need to be aware of when you use the LSI3104 API. Read the following sections that apply to your programming language.

**Note:** Be sure to include the declaration functions of LSI3104 prototypes by including the appropriate LSI3104 header file in your source code. Refer to Building Applications with the LSI3104 Software Library for the header file appropriate to your compiler.

#### 6.3.1 C/C++

For C or C++ programmers, parameters listed as Input/Output parameters or Output parameters are pass-by-reference parameters, which means a pointer points to the destination variable should be passed into the function. For example, the Read Port function has the following format:

```
Status = lsi3104_read_input_status (u8 CardID, u8 axis, u8 point, u8 *state);
```

where **CardID** and **axis** and **point** are input parameters, and **state** is an output parameter.

Consider the following example:

```
u8 CardID, axis , point ;
```

```
u8 state,
```

```
u32 Status;
```

```
Status = lsi3104_read_input_status (u8 CardID, u8 axis, u8 point, u8 &state);
```

#### 6.3.2 Visual basic

The file lsi3104.bas contains definitions for constants required for obtaining LSI Card information and declared functions and variable as global variables. You should use these constants symbols in the lsi3104.bas, do not use the numerical values.

In Visual Basic, you can add the entire lsi3104.bas file into your project. Then you can use any of the constants defined in this file and call these constants in any module of your program. To add the lsi3104.bas file for your project in Visual Basic 4.0, go to the **File** menu and select the **Add File...** option. Select lsi3104.bas, which is browsed in the lsi3104 \ api directory. Then, select **Open** to add the file to the project.

To add the lsi3104.bas file to your project in Visual Basic 5.0 and 6.0, go to the **Project** menu and select **Add Module**. Click on the Existing tab page. **Select** lsi3104.bas, which is in the lsi3104 \ api directory. Then, select **Open** to add the file to the project.

### 6.3.3 Borland C++ builder

To use Borland C++ builder as development tool, you should generate a .lib file from the .dll file by implib.exe.

**implib lsi3104bc.lib lsi3104.dll**

Then add the **LSI3104bc.lib** to your project and add

**#include "LSI3104.h"** to main program.

Now you may use the dll functions in your program. For example, the Read Input function has the following format:

**Status = lsi3104\_read\_input\_status (u8 CardID, u8 axis, u8 point, u8 \*state);**

where **CardID** and **axis, point** are input parameters, and **state** is an output parameter. Consider the following example:

*u8 CardID, axis, point;*

*u8 state;*

*u32 Status;*

*Status = lsi3104\_read\_input\_status (u8 CardID, u8 axis, u8 point, u8 &state);*

### **Initialization function**

#### ● **lsi3104\_initial**

##### **Description**

This function is used to initialize lsi3104 card. Every lsi3104 card windows application has to be initialized by this function before calling other functions.

The lsi3104\_close () function is accompanied with lsi3104\_initial () function to make lsi3104 card windows application program completely ended and memory fully be released.

##### **Syntax**

```
U32 status = lsi3104_initial (void)
```

##### **Input parameters**

Null

##### **Echo**

Status: the function working status. Reference appendix A for status value corresponds to related error code.

#### ● **lsi3104\_close**

##### **Description**

This function is used to release windows application resource that created by lsi3104 card. Every lsi3104 card windows application has to call this function before ending the windows application program.

The lsi3104\_close () function is corresponded with lsi3104\_initial () function to make lsi3104 card windows application program completely ended and memory fully be released.

##### **Syntax**

```
U32 status = lsi3104_close (void)
```

##### **Input parameters**

Null

##### **Echo**

Status: the function working status. Reference appendix A for status value corresponds to related error code.

## ● **lsi3104\_info**

---

### **Description**

This function is used to get lsi3104 card information from device driver. It is not every necessary but may be useful for users to call this function.

### **Syntax**

U32 status = lsi3104\_info (u8 CardID, u16 \*address)

### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

### **Echo**

Address: the card base address assigned by window system.

Status: the function working status. Reference appendix A for status value corresponds to related error code.

## Homing function

### General description

In many of precisely motion control system or three dimensional measuring system, not only need a high speed and stable accurate counter reading, but also need to have a various conditional hardware homing function, to make sure the repeat accuracy reliable.

Lsi3104 card provide with several conditional hardware homing mode, by calling lsi3104\_set\_hard\_homing () function, users can easily select the suitable homing mode to fit current mechanical requirement.

There are 9 homing mode with active conditions list bellowing:

- 0000H-->Normal operation (default)
- 0001H-->Clear counter while A , B , Z , Home signal are "LOW" simultaneously.
- 0002H-->Clear counter while home signal has just inactive and counter up count and the first time A , B , Z are "LOW" simultaneously.
- 0003H--> Clear counter while home signal has just inactive and counter down count and the first time A , B , Z are "LOW" simultaneously.
- 0004H--> Clear counter at the tailing edge of home signal.
- 0005H--> Clear counter at CLR\_CNTR input active low.
- 0006H--> Clear counter while A , B , Z , Home signal are "LOW" simultaneously. Once the counter cleared, this command will also cleared to "0" (Normal mode).
- 0007H--> Clear counter while home signal has just inactive and counter up count and the first time A , B , Z are "LOW" simultaneously. Once the counter cleared, this command will also cleared to "0" (Normal mode).
- 0008H--> Clear counter while home signal has just inactive and counter down count and the first time A , B , Z are "LOW" simultaneously. Once the counter cleared, this command will also cleared to "0" (Normal mode).
- 0009H--> Clear counter at the tailing edge of home signal. Once the counter cleared, this command will also cleared to "0" (Normal mode).
- 000AH--> Clear counter at CLR\_CNTR input active low. Once the counter cleared, this command will also cleared to "0" (Normal mode).

When lsi3104 card has set lsi3104\_set\_hard\_homing () function, once the hardware homing happens, by polling this flag of lsi3104\_read\_hard\_homing\_flag () function, program can easily know hardware homing happened, and then continue to next procedure.

The homing flag only can read once when every time hardware homing happens; because the function will clear itself after been called.

Hardware homing also can lead an immediately interrupt to run the interrupt program. Which need to be set in interrupt section.

Lsi3104 card also provide software homing method, by calling the function of lsi3104\_soft\_homing\_command (), operator can easily define the logical homing point.

## ● **lsi3104 set hard homing**

### **Description**

This function is used to set lsi3104 card homing mode. The definition of hard homing mode is described as above.

### **Syntax**

U32 status = lsi3104\_set\_hard\_homing (u8 CardID, u8 axis, u8 mode)

### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (=0, 1, 2, 3)

Mode: designated hardware homing mode. (=0 ~ 0x0a)

### **Echo**

Status: the function working status. Reference appendix A for status value corresponds to related error code.

## ● **lsi3104 read hard homing flag**

### **Description**

This function is used to read lsi3104 card hardware homing flag.

### **Syntax**

U32 status = lsi3104\_read\_hard\_homing\_flag (u8 CardID, u8 axis, u8 \*flag)

### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (=0, 1, 2, 3)

### **Echo**

Flag: 0 → no operation.

1 → hardware homing happens.

Status: the function working status. Reference appendix A for status value corresponds to related error code.

## ● **lsi3104 soft homing command**

### **Description**

This function is used to clear lsi3104 card counter by software command method.

### **Syntax**

U32 status = lsi3104\_soft\_homing\_command (u8 CardID, u8 axis)

### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be clear. (=0, 1, 2, 3)

### **Echo**

Status: the function working status. Reference appendix A for status value corresponds to related error code.

## **Counter function**

### **General description**

4 counters with several count mode and exclusive latch/load function in lsi3104 card.

Lsi3104 card counters can accept normally quadrature , clock/direction, cw/ccw count mode, which can be set in the function of lsi3104\_set\_counter\_mode ().

Users always choose quadrature mode for encoder or linear scale use. Since quadrature mode have x4, x2, x1 decoding rate, for best revolution reason, take the default value (x4) for granted use, however the decoding rate can changed to lower revolution by calling lsi3104\_set\_quadrature\_times () function.

Counters can be read, load, preset, and latch. Latch function begins at external trigger input signal activate from high to low level moment, cause a hardware trigger and latch the counter value into a latch buffer, the value can read from calling lsi3104\_read\_latched\_value () function, A latch/load flag can automatically result after latch happens. Polling lsi3104\_read\_latch\_flag () function to know latch/load triggered and proceeding next program process.

Preset function need to preset value (by calling lsi3104\_preset\_counter ()) into preset buffer. It is also triggered by external trigger input signal cause an instant preset value load into counter, also a latch/load flag can automatically result after preset load happens.

Use lsi3104\_read\_counter () to read counter value, and use lsi3104\_load\_counter () to load value into counter. Before using latch/load function, calling lsi3104\_latch\_control () to enable latch/load function, since the default value is enabling.

By setting lsi3104\_latch\_mode () to determine external trigger input behavior. Four kinds of latch/load are provided. If choose one time trigger latch/load mode, once trigger happened, the register of function lsi3104\_latch\_control () will be reset to “0”.

## ● **lsi3104 set counter mode**

### **Description**

This function is used to set lsi3104 card counter mode. Three counter's mode with count direction can be set as parameters description following.

### **Syntax**

U32 status = lsi3104\_set\_counter\_mode (u8 CardID, u8 axis, u8 mode)

### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (=0, 1, 2, 3)

Mode: 0000H-->A , B phases input quadrature up count mode(if A lead B).(default)

0001H-->A , B phases input quadrature down count mode(if A lead B).

0002H-->A input is CLOCK , B input is DIRECTION , up count mode.

0003H--> A input is CLOCK , B input is DIRECTION , down count mode. °

0004H-->A input is UP CLOCK , B input is DOWN CLOCK , dual clock mode.

0005H-->A input is DOWN CLOCK , B input is UP CLOCK , dual clock mode

### **Echo**

Status: the function working status. Reference appendix A for status value corresponds to related error code.

## ● **lsi3104 set quadrature times**

### **Description**

This function is used to set lsi3104 card quadrature decoding rate as the counter mode setting in A/B phase quadrature mode.

For the best resolution reason, users always take default value setting as the quadrature decoding rate.

### **Syntax**

U32 status = lsi3104\_set\_quadrature\_times (u8 CardID, u8 axis, u8 times)

### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (=0, 1, 2, 3)

Times: 0000H -->X 4. (default)

0001H -->X 2.

0002H -->X 1.

### **Echo**

Status: the function working status. Reference appendix A for status value corresponds to related error code.

## ● **lsi3104\_read\_counter**

### **Description**

This function is used to read lsi3104 card counter. The 32bit counter values will return.

### **Syntax**

U32 status = lsi3104\_read\_counter (u8 CardID, u8 axis, i32 \*value)

### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (=0, 1, 2, 3)

### **Echo**

Value: 32bit counter value. (-2,147,483,648 ~ 2,147,483,647)

Status: the function working status. Reference appendix A for status value corresponds to related error code.

## ● **lsi3104\_load\_counter**

### **Description**

This function is used to load value into lsi3104 card counter.

By using this function, users can change the counter value, as they want.

### **Syntax**

U32 status = lsi3104\_load\_counter (u8 CardID, u8 axis, i32 load\_value)

### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (=0, 1, 2, 3)

load\_value: assigned value to load counter. (-2,147,483,648 ~ 2,147,483,647)

### **Echo**

Status: the function working status. Reference appendix A for status value corresponds to related error code.

## ● **lsi3104\_preset\_counter**

### **Description**

This function is used to set value into lsi3104 card preset buffer.

For the lsi3104 card specific hardware external trigger load function, users need to preset value into buffer, also need to call lsi3104\_latch\_control () to enable the latch/load function, and call lsi3104\_latch\_mode () to determine the load mode.

After set up these functions, once the hardware external trigger input activate at edge trigger from high to low level, immediately the preset value will transfer to counter by hardware mechanism, simultaneously interrupt request could occurrence to lead an interrupt program, which need to be set in interrupt section.

### **Syntax**

```
U32 status = lsi3104_preset_counter (u8 CardID, u8 axis, i32 preset_value)
```

### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (=0, 1, 2, 3)

preset\_value: assigned value to load counter. (-2,147,483,648 ~ 2,147,483,647)

### **Echo**

Status: the function working status. Reference appendix A for status value corresponds to related error code.

## ● **lsi3104\_latch\_control**

### **Description**

This function is used to enable/disable latch (load) function, which designated to external trigger input.

### **Syntax**

```
U32 status = lsi3104_latch_control (u8 CardID, u8 axis, u8 control)
```

### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (=0, 1, 2, 3)

Control: 0 → disable latch/load function.

1 → enable latch/load function. (Default value)

### **Echo**

Status: the function working status. Reference appendix A for status value corresponds to related error code.

## ● **lsi3104\_latch\_mode**

### **Description**

This function is used to assign the hardware external trigger input function.

By setting this function, users could have one time trigger or continues trigger latch/load function.

### **Syntax**

U32 status = lsi3104\_latch\_mode (u8 CardID, u8 axis, u8 mode)

### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (=0, 1, 2, 3)

Mode: 0 → continuous external trigger latch (counter) mode. (default)

1 → one-shot external trigger latch mode. Once triggered the register of function lsi3104\_latch\_control () will be reset to “0”.

2 → continuous external trigger load (counter) mode.

3 → one-shot external trigger load mode. Once triggered the register of function lsi3104\_latch\_control () will be reset to “0”.

### **Echo**

Status: the function working status. Reference appendix A for status value corresponds to related error code.

## ● **lsi3104\_read\_latch\_flag**

### **Description**

This function is used read latch flag, which accused by hardware external trigger input.

Reading this flag get “1” indicate occurrence latch/load trigger event. After calling this function, the function clear itself flag as reset to prepare next event beginning.

### **Syntax**

U32 status = lsi3104\_read\_latch\_flag (u8 CardID, u8 axis, u8 \*flag)

### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (=0, 1, 2, 3)

### **Echo**

Flag: 0 → no operation.

1 → latch/load event happens.

Status: the function working status. Reference appendix A for status value corresponds to related error code.

## ● **lsi3104\_read\_latched\_value**

### **Description**

This function is used to read lsi3104 card latch buffer value, which has been latched from counter, as the external latch trigger happens. The 32bit counter values will return.

### **Syntax**

U32 status = lsi3104\_read\_latched\_value (u8 CardID, u8 axis, i32 \*value)

### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (=0, 1, 2, 3)

### **Echo**

Value: 32bit latched value. (-2,147,483,648 ~ 2,147,483,647)

Status: the function working status. Reference appendix A for status value corresponds to related error code.

## **Input/Output function**

### **General description**

Lsi3104 card provide 6 exclusive input (A\_PHASE,B\_PHASE,ZERO,HOME,LATCH,CLEAR\_IN) and 1 general input (GENERAL\_IN) for each axis. Each input point also combines with a polarity bit. However each exclusive input point also can use for general purpose.

By calling `lsi3104_read_input_status ()` to get each input status, the status can be switched, since the related polarity has changed, which is setting by `lsi3104_set_input_polarity ()` function. Also each polarity bit can read back from calling `lsi3104_read_input_polarity ()` function.

Besides, lsi3104 card provide each zero input mapping a ZERO TOGGLE FLAG, which is a toggle bit caused by each zero input edge trigger from high to low level moment.

Each axis has one general output (GENERAL\_OUT) and one exclusive output (CLEAR\_OUT). CLEAR\_OUT output point's behavior is defined by `lsi3104_set_clr_out_mode ()` function, which determine CLEAR\_OUT to be the second general output point or as a clear counter exclusive output point for each axis. When using as an exclusive clear output point, it should be connected to driver's specific input to clear remainder pulse, as the hardware homing happens.

When using CLEAR\_OUT as the second general output point, two output's action is controlled by `lsi3104_write_output_command ()` function. The output status also can read back by calling `lsi3104_read_output_status ()` function. Besides, output's polarity setting and read back is determined by `lsi3104_set_output_polarity ()` and `lsi3104_read_output_polarity ()` function

## ● **lsi3104 read input status**

### **Description**

This function is used to read lsi3104 card input status.

Calling lsi3104\_set\_input\_polarity () can change the related polarity.

### **Syntax**

```
U32 status = lsi3104_read_input_status (u8 CardID, u8 axis, u8 point, u8 *state)
```

### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (=0, 1, 2, 3)

Point: designated which point to be read.

0 → A\_PHASE input point

1 → B\_PHASE input point

2 → ZERO input point

3 → HOME input point

4 → LATCH input point

5 → CLEAR\_IN input point

6 → GENERAL\_IN input point

7 → ZERO\_TOGGLE\_FLAG point

### **Echo**

State: 0 → inactive

1 → active

Status: the function working status. Reference appendix A for status value corresponds to related error code.

## ● **lsi3104 set input polarity**

### **Description**

This function is used to set lsi3104 card input point polarity.

### **Syntax**

U32 status = lsi3104\_set\_input\_polarity (u8 CardID, u8 axis, u8 point, u8 polarity)

### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (=0, 1, 2, 3)

Point: designated which point to set.

0 → A\_PHASE input point

1 → B\_PHASE input point

2 → ZERO input point

3 → HOME input point

4 → LATCH input point

5 → CLEAR\_IN input point

6 → GENERAL\_IN input point

Polarity: 0 → negative polarity, active low (default)

1 → positive polarity, active high

### **Echo**

Status: the function working status. Reference appendix A for status value corresponds to related error code.

## ● **lsi3104\_read\_input\_polarity**

### **Description**

This function is used to read polarity back from lsi3104 card input point.

### **Syntax**

U32 status = lsi3104\_read\_input\_polarity (u8 CardID, u8 axis, u8 point, u8 \*polarity)

### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (=0, 1, 2, 3)

Point: designated which point to read back.

0 → A\_PHASE input point

1 → B\_PHASE input point

2 → ZERO input point

3 → HOME input point

4 → LATCH input point

5 → CLEAR\_IN input point

6 → GENERAL\_IN input point

### **Echo**

Polarity: 0 → negative polarity, active low (default)

1 → positive polarity, active high

Status: the function working status. Reference appendix A for status value corresponds to related error code.

## ● **lsi3104\_set\_clr\_out\_mode**

### **Description**

This function is used to set lsi3104 card clear output mode.

### **Syntax**

U32 status = lsi3104\_set\_clr\_out\_mode (u8 CardID, u8 axis, u8 mode)

### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (=0, 1, 2, 3)

Mode: 0 → automatic generate an active low output for about 0.1s, as hardware homing happens. (Default)

1 → Take CLR\_CNTR as second general output.

### **Echo**

Status: the function working status. Reference appendix A for status value corresponds to related error code.

## ● **lsi3104 write output command**

### **Description**

This function is used to write lsi3104 card output point command.

Calling lsi3104\_set\_output\_polarity () can change the related polarity.

### **Syntax**

U32 status = lsi3104\_write\_output\_command (u8 CardID, u8 axis, u8 point, u8 on\_off)

### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (=0, 1, 2, 3)

Point: 0 → GENERAL\_OUT point

1 → CLEAR\_OUT point, where use this point before, need to call  
lsi3104\_set\_clr\_out\_mode (), and set mode = 1.

On\_off: 0 → inactive

1 → active

### **Echo**

Status: the function working status. Reference appendix A for status value corresponds to related error code.

## ● **lsi3104 set ooutput polarity**

### **Description**

This function is used to set lsi3104 card general output polarity.

### **Syntax**

U32 status = lsi3104\_set\_output\_polarity (u8 CardID, u8 axis, u8 point, u8 polarity)

### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (=0, 1, 2, 3)

Point: 0 → GENERAL\_OUT point

1 → CLEAR\_OUT point, where use this point before, need to call  
lsi3104\_set\_clr\_out\_mode (), and set mode = 1.

Polarity: 0 → negative polarity, NO. Connection (default)

1 → positive polarity, NC. Connection

### **Echo**

Status: the function working status. Reference appendix A for status value corresponds to related error code.

## ● **lsi3104\_read\_output\_status**

### **Description**

This function is used to read status back from lsi3104 card output point.

### **Syntax**

U32 status = lsi3104\_read\_output\_status (u8 CardID, u8 axis, u8 point, u8 \*state)

### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (=0, 1, 2, 3)

Point: 0 → GENERAL\_OUT point

1 → CLEAR\_OUT point

### **Echo**

State: 0 → inactive

1 → active

Status: the function working status. Reference appendix A for status value corresponds to related error code.

## ● **lsi3104\_read\_output\_polarity**

### **Description**

This function is used to read polarity back from lsi3104 card output point.

### **Syntax**

U32 status = lsi3104\_read\_output\_status (u8 CardID, u8 axis, u8 point, u8 \*polarity)

### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (=0, 1, 2, 3)

Point: 0 → GENERAL\_OUT point

1 → CLEAR\_OUT point

### **Echo**

Polarity: 0 → negative polarity, NO. Connection (default)

1 → positive polarity, NC. Connection

Status: the function working status. Reference appendix A for status value corresponds to related error code.

## Security function

### ● lsi3104\_set\_password

**Format :** u32 status = lsi3104\_set\_password(u8 CardID,u16 password[5])

**Purpose:** To set password and if the password is not all “0”, security function will be enabled.

**Parameters:**

**Input:**

Name	Type	Description
CardID	u8	assigned by DIP/ROTARY SW
password[5]	u16	Password, 5 words

**Note:** If the password is all “0”, the security function is disabled.

### ● lsi3104\_change\_password

**Format :** u32 status = lsi3104\_change\_password(u8 CardID,u16 Oldpassword[5],  
u16 password[5])

**Purpose:** To replace old password with new password.

**Parameters:**

**Input:**

Name	Type	Description
CardID	u8	assigned by DIP/ROTARY SW
Oldpassword [5]	u16	The previous password
password[5]	u16	The new password to be set

### ● lsi3104\_clear\_password

**Format :** u32 status = lsi3104\_clear\_password(u8 CardID,u16 password[5])

**Purpose:** To clear password, to set password to all “0”, i.e. disable security function.

**Parameters:**

**Input:**

Name	Type	Description
CardID	u8	assigned by DIP/ROTARY SW
password[5]	u16	The password previous set

● **lsi3104\_unlock\_security**

**Format :** u32 status = lsi3104\_unlock\_security(u8 CardID,u16 password[5])

**Purpose:** To unlock security function and enable the further operation of this card

**Parameters:**

**Input:**

Name	Type	Description
CardID	u8	assigned by DIP/ROTARY SW
password[5]	u16	The password previous set

● **lsi3104\_read\_security\_status**

**Format :** u32 status = lsi3104\_read\_security\_status(u8 CardID,u8 \*lock\_status,  
u8 \*security\_enable )

**Purpose:** To read security status for checking if the card security function is unlocked.

**Parameters:**

**Input:**

Name	Type	Description
CardID	u8	assigned by DIP/ROTARY SW

**Output:**

Name	Type	Description
lock_status	u8	0: security unlocked 1: locked 2: dead lock (must return to original maker to unlock)
security_enable	u8	0: security function disabled 1: security function enabled

## Miscellaneous function

### General description

This section contains utility function, which is different from other sections.

### ● lsi3104\_com\_trigger\_control

#### Description

This function is used to set lsi3104 card 4 axes, whose are common triggered (latch/load) by X trigger (latch/load) input or triggered from respectively axis.

#### Syntax

U32 status = lsi3104\_com\_trigger\_control (u8 CardID, u8 control)

#### Input parameters

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Control: 0 → individual mode, triggered respectively (Default value)

1 → 4 axes common triggered by X axis trigger input.

#### Echo

Status: the function working status. Reference appendix A for status value corresponds to related error code.

## ● **lsi3104\_read\_parameters**

### **Description**

This function is used to read parameters of each axis.

### **Syntax**

U32 status = lsi3104\_read\_parameters (u8 CardID, u8 axis, u8 parameter\_no, u8 \*value)

### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (0:X axis, 1:Y axis, 2:Z axis, 3:A axis)

parameter\_no:

0: homing mode (refer lsi3104\_set\_hard\_homing())

1: counter mode (refer lsi3104\_set\_counter\_mode())

2: quadrature times (refer lsi3104\_set\_counter\_mode())

3: latch mode (refer lsi3104\_set\_latch\_mode())

4: latch control (refer lsi3104\_set\_latch\_control())

5: clear out mode (refer lsi3104\_clr\_out\_mode())

6: common trigger control (refer lsi3104\_com\_trigger\_control())

(Axis parameter is no need)

7: compare out mode (refer lsi3104\_set\_compare\_out\_mode())

(only for x and z axis)

### **Echo**

Status: the function working status. Reference appendix A for status value corresponds to related error code.

value: the mode value

## Compare function (software version V1.42 and after)

### ● lsi3104 set compare out mode(only for x and z axis)

#### **Description**

This function is used to set the compare out mode

#### **Syntax**

U32 status = lsi3104\_set\_compare\_out\_mode (u8 CardID, u8 axis, u8 mode)

#### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (0:X axis, 2:Z axis)

mode:

0: output point (corresponding to each axis) is general purpose output and its polarity can be programmed.

1: during the counter meet the preset compare value a pulse of duration 0.1s is active low at out point.

2: during the counter meet the preset compare value a pulse of duration 0.1s is active low at out point. The next compare value auto loaded (refer lsi3104\_load\_increment\_value)

#### **Echo**

Status: the function working status. Reference appendix A for status value corresponds to related error code.

### ● lsi3104 load compare value (only for x and z axis)

#### **Description**

This function is used to set the compare value

#### **Syntax**

U32 status = lsi3104\_load\_compare\_value (u8 CardID, u8 axis, i32 value)

#### **Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (0:X axis, 2:Z axis)

value: 32 bit value (-2,147,483,648 ~ 2,147,483,647) to be compared with position counter.

#### **Echo**

Status: the function working status. Reference appendix A for status value corresponds to related error code.

● **lsi3104\_read\_compare\_value (only for x and z axis)**

**Description**

This function is used to read the compare value

**Syntax**

U32 status = lsi3104\_read\_compare\_value (u8 CardID, u8 axis, i32 \*value)

**Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (0:X axis, 2:Z axis)

**Echo**

Status: the function working status. Reference appendix A for status value corresponds to related error code.

value: 32 bit preset compare value (-2,147,483,648 ~ 2,147,483,647)

● **lsi3104\_load\_increase\_value (only for x and z axis)**

**Description**

This function is used to load the increase value.

Next compare value = previous compare value + increase value

At compare out mode 2, next compare value will be loaded after compare out trigger.

**Syntax**

U32 status = lsi3104\_load\_increase\_value (u8 CardID, u8 axis, i32 value)

**Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (0:X axis, 2:Z axis)

value: 32 bit value (-2,147,483,648 ~ 2,147,483,647) to be compared with position counter.

**Echo**

Status: the function working status. Reference appendix A for status value corresponds to related error code.

- **lsi3104 read increase value (only for x and z axis)**

**Description**

This function is used to read the increase value.

**Syntax**

U32 status = lsi3104\_load\_increase\_value (u8 CardID, u8 axis, i32 \*value)

**Input parameters**

CardID: assigned card number by dipswitch setting on board. (= 0 ~ 15)

Axis: designated which axis to be set. (0:X axis, 2:Z axis)

**Echo**

Status: the function working status. Reference appendix A for status value corresponds to related error code.

value: 32 bit preset increase value (-2,147,483,648 ~ 2,147,483,647)

## 6.5 Dll list

	<b>Function Name</b>	<b>Description</b>
1	lsi3104_initial ( )	Software initializaton
2	lsi3104_close ( )	Software close
3	lsi3104_info ( )	Get card information
4	lsi3104_set_hard_homing ( )	Set homing mode
5	lsi3104_read_hard_homing_flag ( )	Read homing flag
6	lsi3104_soft_homing_command ( )	Software homing command
7	lsi3104_set_counter_mode ( )	Set counter mode
8	lsi3104_set_quadrature_times ( )	Set decoding rate
9	lsi3104_read_counter ( )	Read encoder counter
10	lsi3104_load_counter ( )	Load value to counter
11	lsi3104_preset_counter ( )	Preset counter value
12	lsi3104_latch_control ( )	Enable/disable latch function
13	lsi3104_latch_mode ( )	Set latch/load mode
14	lsi3104_read_latch_flag ( )	Read latch/load flag
15	lsi3104_read_latched_value ( )	Read latch value
16	lsi3104_read_input_status ( )	Read input point status
17	lsi3104_set_input_polarity ( )	Set input point polarity
18	lsi3104_read_input_polarity ( )	Input point polarity readback
19	lsi3104_set_clr_out_mode ( )	Set clear output mode
20	lsi3104_write_output_command ( )	Write output point command
21	lsi3104_set_output_polarity ( )	Set output point polarity
22	lsi3104_read_output_status ( )	Output point status readback
23	lsi3104_read_output_polarity ( )	Output point polarity readback
24	lsi3104_set_password( )	Set password
25	lsi3104_change_password( )	Change password
26	lsi3104_clear_password( )	Clear password.
27	lsi3104_unlock_security( )	Unlock security function
28	lsi3104_read_security_status( )	Read security status
29	lsi3104_com_trigger_control ( )	Common trigger control
30	lsi3104_read_parameters ( )	Miscellaneous parameters readback
31	lsi3104_set_compare_out_mode( )	Set compare out mode
32	lsi3104_load_compare_value( )	Set compare value
33	lsi3104_read_compare_value( )	Compare value readback from card
34	lsi3104_load_increase_value( )	Set auto increase value after compare finished.
35	lsi3104_read_increase_value( )	Increase value readback from card.

## 7. LSI-3104 Error codes summary

### 7.1 LSI3104 Error codes table

<b>Error Code</b>	<b>Symbolic Name</b>	<b>Description</b>
<b>0</b>	JSDRV_NO_ERROR	No error.
<b>2</b>	JSDRV_INIT_ERROR	Initial error
<b>100</b>	DEVICE_IO_ERROR	Device drive error
<b>101</b>	JSDRV_NO_CARD	No card find error
<b>102</b>	JSDRV_DUPLICATE_ID	Card duplicate error
<b>300</b>	JLSI_ID_ERROR	CardID setting error, CardID doesn't match the DIP SW setting
<b>301</b>	JLSI_COUNTER_MODE_ERROR	lsi3104_set_counter_mode(),"mode" parameter out of range.
<b>302</b>	JLSI_QUADRATURE_TIMES_ERROR	lsi3104_set_quadrature_times(),"time" parameter out of range.
<b>303</b>	JLSI_LATCH_CTRL_ERROR	lsi3104_latch_control(),"control" parameter out of range.
<b>304</b>	JLSI_LATCH_MODE_ERROR	lsi3104_latch_mode(),"mode" parameter out of range.
<b>305</b>	JLSI_POINT_ERROR	"point" parameter out of range.
<b>306</b>	JLSI_AXIS_ERROR	"axis" parameter out of range.
<b>307</b>	JLSI_CLR_OUT_MODE_ERROR	lsi3104_set_clr_out_mode(),"mode" parameter out of range.
<b>308</b>	JLSI_HOME_MODE_ERROR	lsi3104_set_hard_homing(),"mode" parameter out of range.
<b>309</b>	JLSI_POLARITY_ERROR	"polarity" parameter out of range.
<b>310</b>	JLSI_ON_OFF_ERROR	"on_off" parameter out of range.
<b>311</b>	JLSI_TRIGGER_CTRL_ERROR	lsi3104_com_trigger_control(),"control" parameter out of range.
<b>400</b>	JLSI_DRIVER_NOT_SUPPORT	driver not support interrupt function.