

LSI3123A

Quadrature Encoder Counter Card

Software Manual (V1.2)

健昇科技股份有限公司

JS AUTOMATION CORP.

新北市汐止區中興路 100 號 6 樓
6F., No.100, Zhongxing Rd.,
Xizhi Dist., New Taipei City, Taiwan

TEL : +886-2-2647-6936

FAX : +886-2-2647-6940

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

<http://www.automation-js.com/>

E-mail : control.cards@automation.com.tw

Correction record

Version	Record
1.0	for driver V5.0 up
	1. extend function
	mode=0Bh,0Ch for <i>LSI3123_set_hard_homing</i>
	flag=2 for <i>LSI3123_read_hard_homing_flag</i>
	2. add new function
	<i>LSI3123_start_homing_calibration</i>
	<i>LSI3123_calibrate_ABS_position</i>
1.0->1.1	1. add new function
	<i>LSI3123_check_homed_calibration_flag</i>
1.1->1.2	Modify all function name (lowercase → uppercase)

Contents

1.	How to install the software of LSI3123A	4
1.1	Install the PCI driver.....	4
2.	Where to find the file you need.....	5
3.	About the LSI3123A software	6
3.1	What you need to get started	6
3.2	Software programming choices	6
4.	LSI3123A Language support	7
4.1	Building applications with the LSI3123A software library	7
4.2	LSI3123A Windows libraries.....	7
5.	Software overview	8
5.1	Initialization	8
5.2	I/O Port R/W.....	8
5.3	Homing functions	8
5.4	Counter functions.....	9
5.5	Security functions	9
5.6	Miscellaneous functions	10
5.7	Error conditions	10
6.	Flow chart of application implementation	11
6.1	LSI3123A Flow chart of application implementation	11
7.	Function reference.....	15
7.1	Error codes and CardID	15
7.2	Variable data types	16
7.3	Programming language considerations	17
7.4	LSI3123A Functions	19
Initialization function group	19	
LSI3123_initial	19	
LSI3123_close	19	
LSI3123_info.....	19	
Input/output function group.....	20	
LSI3123_set_input_polarity	20	
LSI3123_set_led_polarity.....	20	
LSI3123_read_input_polarity.....	21	
LSI3123_read_input_status	21	
Homing function group.....	22	
LSI3123_set_hard_homing.....	22	
LSI3123_read_hard_homing_flag.....	22	
LSI3123_soft_homing_command	23	
Counter function group.....	24	
LSI3123_set_counter_mode	24	

LSI3123_set_quadrature_times	24
LSI3123_read_counter.....	24
LSI3123_load_counter.....	25
LSI3123_latch_control	25
LSI3123_latch_mode.....	25
LSI3123_preset_counter	26
LSI3123_read_latch_flag.....	26
LSI3123_read_latch_value	26
Security function group	27
LSI3123_set_password.....	27
LSI3123_change_password.....	27
LSI3123_clear_password.....	28
LSI3123_unlock_security	28
LSI3123_read_security_status.....	29
Miscellaneous function group.....	30
LSI3123_read_parameters	30
LSI3123_start_homing_calibration	30
LSI3123_calibrate_ABS_position.....	31
LSI3123_check_homed_calibration_flag.....	31
7.5 Dll list	32
8. LSI3123A Error codes summary.....	33
8.1 LSI3123A Error codes table	33

1. **How to install the software of LSI3123A**

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 Win98/2K/XP and up system you should: (take Win XP 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
(..\LSI3123A\Software\Win98_2K_XP\Driver\ or if you download from website please execute the file LSI3123A_Driver.exe to get the file), then it will automatically setup the driver
6. After installation, power off
7. Power on, it's ready to use

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. **Where to find the file you need**

Win98/2K/XP and up

In Win98/2K/XP and up system, the demo program can be setup by

\LSI3123A\Software\Win98_2K_XP\Install\LSI3123A_Install.exe

The directory will be located at

../**JS Automation /LSI3123A/API** (header files and lib files for VB,VC)

../**JS Automation /LSI3123A/Driver** (backup copy of LSI3123A drivers)

../**JS Automation /LSI3123A/exe** (demo program and source code)

The system driver is located at ..\system32\Drivers and the DLL is located at ..\system.

For your easy startup, the demo program with source code demonstrates the card functions and help file.

3. About the LSI3123A software

LSI3123A software includes a set of dynamic link library (DLL) and system driver that you can utilize to control the I/O card's ports and points separately.

Your LSI3123A 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 LSI3123A functions within Windows' operation system environment.

3.1 What you need to get started

To set up and use your LSI3123A software, you need the following:

- LSI3123A software
- LSI3123A hardware
 - Main board
 - Wiring board (Option)

3.2 Software programming choices

You have several options to choose from when you are programming LSI3123A 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 LSI3123A software.

4. **LSI3123A Language support**

The LSI3123A software library is a DLL used with Win98/2K/XP and up. You can use these DLL with any Windows integrating development environment that can call Windows DLLs.

4.1 Building applications with the LSI3123A software library

The LSI3123A function reference topic contains general information about building LSI3123A applications, describes the nature of the LSI3123A files used in building LSI3123A applications, and explains the basics of making applications using the following tools:

Applications tools

- Microsoft Visual C/C++
- Borland 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.

4.2 LSI3123A Windows libraries

The LSI3123A for Windows function library is a DLL called **LSI3123.dll**. Since a DLL is used, LSI3123A functions are not linked into the executable files of applications. Only the information about the LSI3123A functions in the LSI3123A 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 LSI3123A functions in LSI3123.dll.

Header Files and Import Libraries for Different Development Environments		
Language	Header File	Import Library
Microsoft Visual C/C++	LSI3123.h	LSI3123VC.lib
Borland C/C++	LSI3123.h	LSI3123BC.lib
Microsoft Visual Basic	LSI3123.bas	

Table 1

5. Software overview

These topics describe the features and functionality of the LSI3123A boards and briefly describes the LSI3123A functions.

5.1 Initialization

You need to initialize each time you run your application and close your driver before you will close applications.

[*LSI3123 initial\(\)*](#)

[*LSI3123 close\(\)*](#)

If you need to get the card address assigned by the system,

[*LSI3123 info\(\)*](#) will do.

5.2 I/O Port R/W

Use the following functions for I/O port output value reading and control:

Set I/O Port Polarity use:

[*LSI3123 set input polarity\(\)*](#)

[*LSI3123 set led polarity\(\)*](#)

To read back input polarity use:

[*LSI3123 read input polarity\(\)*](#) and to read input status with

[*LSI3123 read input status\(\)*](#).

5.3 Homing functions

If you want to clear counter with external trigger , use

[*LSI3123 set hard homing \(\)*](#) first to setup external trigger function. After configuration, use

[*LSI3123 read hard homing flag \(\)*](#) to check the occurrence of external trigger.

Sometimes you want to clear counter with software, use

[*LSI3123 soft homing command\(\)*](#) to implement.

5.4 Counter functions

Counting is the major function of this card. To use counter function setup the function mode is the first step. Use

[*LSI3123 set counter mode\(\)*](#) to configure the working mode of the counter.

If you are working with a quadrature encoder use

[*LSI3123 set quadrature times\(\)*](#) to setup the multiple rate.

To read the counter on the fly or to load it

[*LSI3123 read counter\(\)*](#) and [*LSI3123 load counter\(\)*](#) will do.

If you are using a external device trigger to load the counter, first you must set the latch control mode.

[*LSI3123 latch control\(\)*](#) is required at initial stage to configure the trigger source and

[*LSI3123 latch mode\(\)*](#) to setup the trigger mode and finally the value to be load should set by

[*LSI3123 preset counter\(\)*](#).

If you are using a external device trigger to latch the counter on the fly, first you must set the latch control mode.

[*LSI3123 latch control\(\)*](#) is required at initial stage to configure the trigger source and

[*LSI3123 latch mode\(\)*](#) to setup the trigger mode and verify if any trigger occurs by

[*LSI3123 read latch flag\(\)*](#). If it does, read the latched value by using

[*LSI3123 read latched value\(\)*](#).

5.5 Security functions

Since LSI3123A is a general purpose card, anyone who can buy from JS automation corp. or her distributors, your program is the fruit of your intelligence, un-authorized copy maybe prevent by the security function enabled.

You can use

[*LSI3123 set password\(\)*](#) to set password and [*LSI3123 change password\(\)*](#) to change it.

If you don't want to use security function after the password being setup,

[*LSI3123 clear password\(\)*](#) will reset to the virgin state.

Once the password is set, any function call of the dll's (except for the security functions) will be blocked until the

[*LSI3123 unlock security\(\)*](#) unlock the security.

You can also use [*LSI3123 read security status\(\)*](#) to check the current status of security.

5.6 Miscellaneous functions

Sometimes you need to check the working modes, use [LSI3123 read parameters\(\)](#) to check the homing mode, counter mode, latch mode, multiple rate and latch control.

For some linear scale with ABS coordinate reference marker, the card can change to continuous external trigger mode to latch the reference position then use the 4 latched position to rebuild the coordinate. This means you do not need to absolute reference point for homing while power on.

Use [LSI3123 start homing calibration\(\)](#) to rebuild the coordinate after you have latched the 4 reference maker position (using [LSI3123 set hard homing\(\)](#) in mode 0B or 0C to latch).

While first setup, use

[LSI3123 calibrate ABS position\(\)](#) to setup the virgin scale to the system after you have latched the 4 reference maker position.

If at certain condition you must hot reset computer after the coordinate system is build up, the coordinate system is still kept in LSI3123 counters, you may use

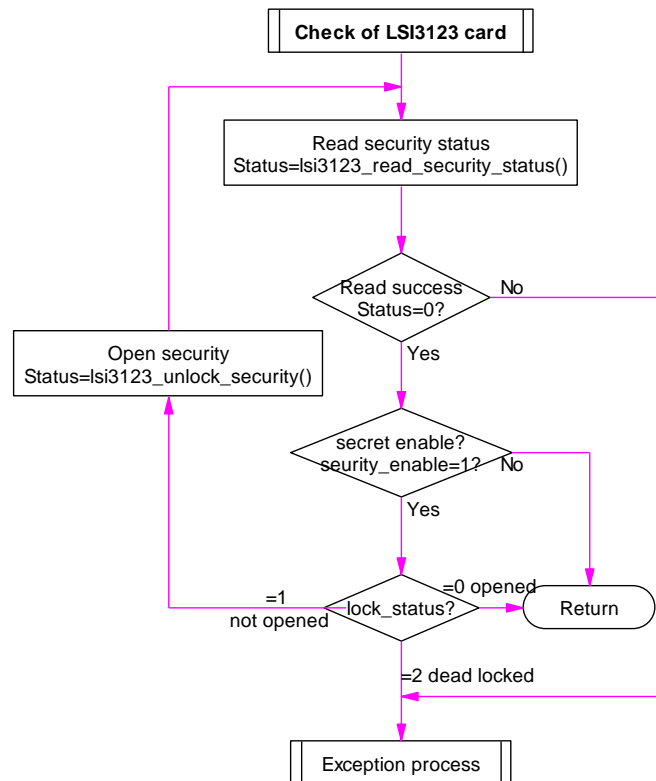
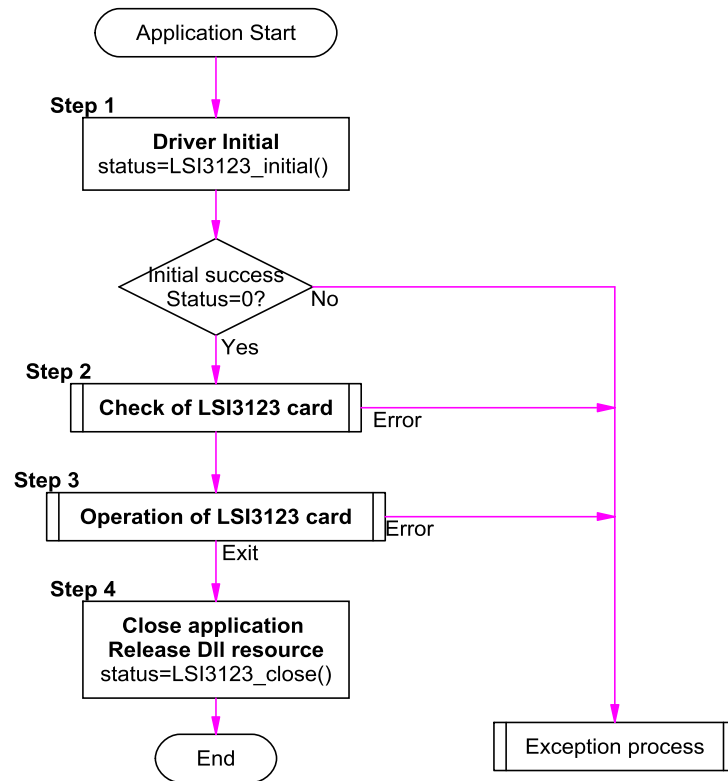
[LSI3123 check homed calibration flag\(\)](#) to check the status of coordinate build-up before system reboot, if homed, no need to re-build the coordinate system.

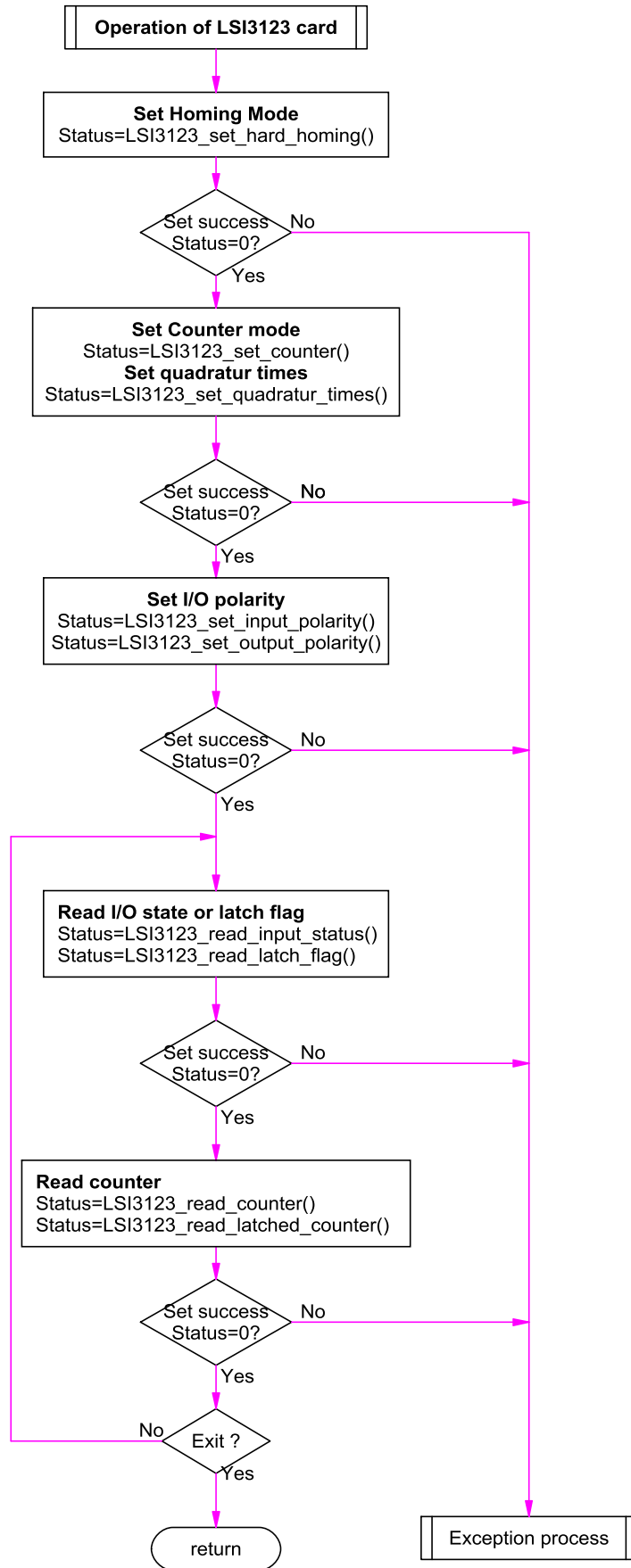
5.7 Error conditions

LSI3123A dll's will report several errors according the conditions it occurs. Error Codes contains a detailed listing of the error status returned by LSI3123A functions.

6. Flow chart of application implementation

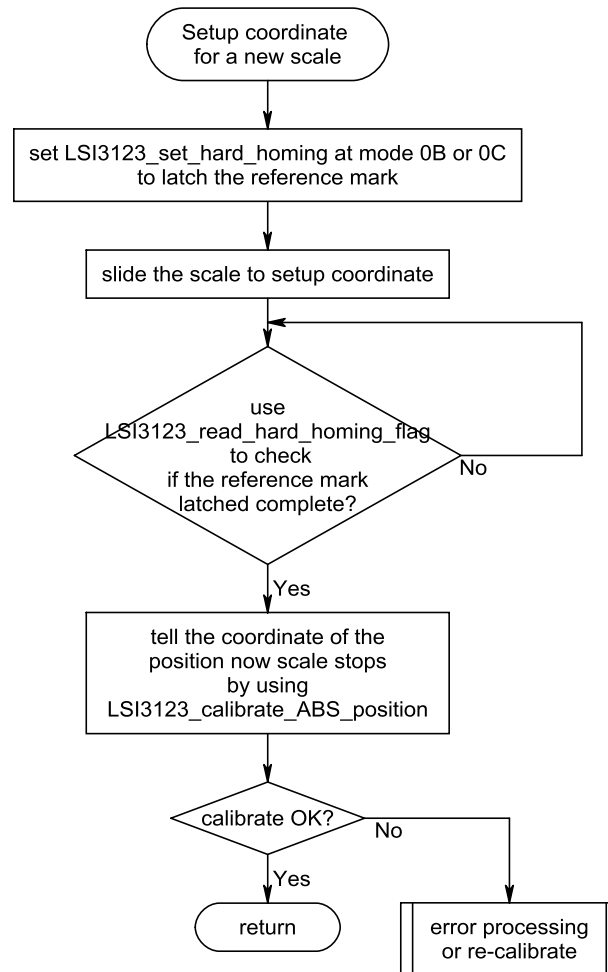
6.1 LSI3123A Flow chart of application implementation

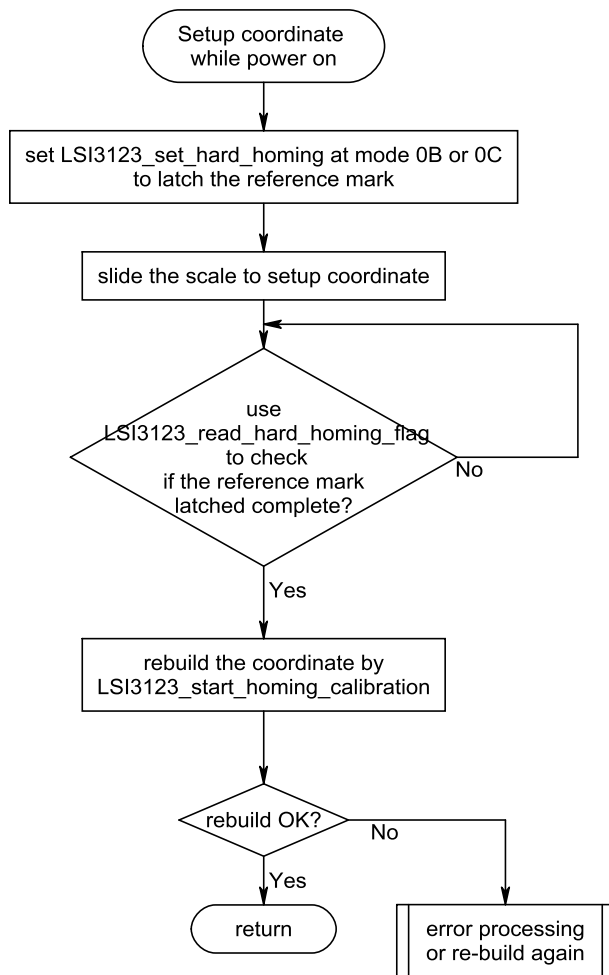




Special notes on ABS coordinate application for Acu-rite linear scale

1. If use *LSI3123_set_hard_homing* function in mode=0Bh , Acu-rite linear scale must program A and B phase in invert mode. There is no A,B,Z phase low condition at normal mode.
2. Use *LSI3123_set_hard_homing* function in mode=0Ch, only need the Zero phase active low to trigger the latch circuit.
3. Check if the trigger data is complete by *LSI3123_read_hard_homing_flag*.
4. If the scale is a virgin to the machine, use *LSI3123_calibrate_ABS_position* to set up the coordinate you need at the position the scale stops.
5. After a new power on, if the scale has been setup before, use *LSI3123_start_homing_calibration* to rebuild the coordinate.





7. **Function reference**

These topics contain detailed descriptions of each LSI3123A function. The functions are arranged by function group. Refer to LSI3123A Function Reference for additional information.

7.1 Error codes and CardID

Every LSI3123A 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 LSI3123A function is the parameter **CardID** which is located the driver of LSI3123A board you want to use those given operation. The **CardID** is assigned by ROTARY(or DIP) SW. You can utilize multiple devices with different card CardID within one application; to do so, simply pass the appropriate **CardID** to each function.

Note: **CardID** is set by ROTARY SW (**0x0-0xF**)

7.2 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)
u8	8-bit ASCII character	0 to 255	char	Not supported by BASIC. For functions that require character arrays, use string types instead.	Byte
i16	16-bit signed integer	-32,768 to 32,767	short	Integer (for example: deviceNum%)	SmallInt
u16	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
i32	32-bit signed integer	-2,147,483,648 to 2,147,483,647	long	Long (for example: count&)	LongInt
u32	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.
f32	32-bit single-precision floating-point value	-3.402823E+38 to 3.402823E+38	float	Single (for example: num!)	Single
f64	64-bit double-precision floating-point value	-1.797683134862315E+308 to 1.797683134862315E+308	double	Double (for example: voltage Number)	Double

Table 2

7.3 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 LSI3123A API. Read the following sections that apply to your programming language.

Note: Be sure to include the declaration functions of LSI3123A prototypes by including the appropriate LSI3123A header file in your source code. Refer to Building Applications with the LSI3123A Software Library for the header file appropriate to your compiler.

7.3.1 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 = LSI3123_read_port(CardID, port, data);
```

where **CardID** and **port** are input parameters, and **data** is an output parameter. Consider the following example:

```
u8 CardID, port;  
u8 data,  
u32 Status;  
Status = read_port (CardID, port, &data);
```

7.3.2 Visual basic

The file LSI3123.bas contains definitions for constants required for obtaining DIO Card information and declared functions and variable as global variables. You should use these constants symbols in the LSI3123.bas, do not use the numerical values.

In Visual Basic, you can add the entire LSI3123.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 LSI3123.bas file for your project in Visual Basic 4.0, go to the **File** menu and select the **Add File... option**. Select LSI3123.bas, which is browsed in the LSI3123A \ API directory. Then, select **Open** to add the file to the project.

To add the LSI3123.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** LSI3123.bas, which is in the LSI3123A \ API directory. Then, select **Open** to add the file to the project.

7.3.3 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 LSI3123BC.lib LSI3123.dll

Then add the **LSI3123BC.lib** to your project and add

#include "LSI3123.h" to main program.

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

```
Status = LSI3123_read_port(CardID, port, data);
```

where **CardID** and **port** are input parameters, and **data** is an output parameter. Consider the following example:

```
u16 CardID, port;
```

```
u8 data;
```

```
u32 Status;
```

```
Status = read_port (CardID, port, &data);
```

Initialization function group

● **LSI3123 initial**

Format : u32 Status =LSI3123_initial (void)

Purpose: Initialize the LSI3123 resource when start the Windows applications.

● **LSI3123 close**

Format : u32 Status =LSI3123_close (void);

Purpose: Release the LSI3123 resource when close the Windows applications.

● **LSI3123 info**

Format : u32 Status =LSI3123_info (u8 CardID,u16 *address);

Purpose: To access the LSI3123a physical address assigned by OS

Input:

Name	Type	Description
CardID	u8	assigned by DIP(rotary) SW

Output:

Name	Type	Description
address	u16	physical address assigned by OS

Input/output function group

● LSI3123 set input polarity

Format : u32 Status = LSI3123_set_input_polarity(u8 CardID,u8 Axis,u8 point,
u8 polarity);

Purpose: Set input polarity

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis
point	u8	0: A phase input 1: B phase input 2: C(Zero) phase input 4: latch input while Axis=0, latch input is Probe_IN while Axis=1, latch input is push button input (Foot_SW_IN)
Polarity	u8	0: normal 1: invert

● LSI3123 set led polarity

Format : u32 Status = LSI3123_set_led_polarity(u8 CardID,u8 polarity);

Purpose: Set the LED polarity. The LED is activated by Probe_IN(LatchIN0).

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
polarity	u8	0: normal 1: inverse

● **LSI3123 read input polarity**

Format : u32 Status = LSI3123_read_input_polarity(u8 CardID,u8 Axis,u8 point,
u8 *polarity);

Purpose: Read input polarity

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis
point	u8	0: A phase input 1: B phase input 2: C(Zero) phase input 4: latch input while Axis=0, latch input is Probe_IN while Axis=1, latch input is push botton input (Foot_SW_IN)

Output:

Name	Type	Description
Polarity	u8	0: normal 1: invert

● **LSI3123 read input status**

Format : u32 Status = LSI3123_read_input_status(u8 CardID,u8 Axis,u8 point,u8 *state);

Purpose: Read the input status

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis
point	u8	0: A phase input 1: B phase input 2: C(Zero) phase input 4: latch input while Axis=0, latch input is Probe_IN while Axis=1, latch input is push botton input (Foot_SW_IN)

Output:

Name	Type	Description
state	u8	input status

Homing function group

● **LSI3123 set hard homing**

Format : u32 Status = LSI3123_set_hard_homing(u8 CardID,u8 Axis,u8 mode);

Purpose: To set the working mode of clearing counter.

Input:

Name	Type	Description
CardID	u8	assigned by DIP(rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis
mode	u8	00h: no hardware homing (default) 01h: clear counter at A,B,C(Zero) phases goes low simultaneously 05h: clear counter at C(Zero) phases goes low 06h: clear counter at A,B, C (Zero) phases goes low simultaneously and reset to “no hardware homing” after the occurrence 0Ah: clear counter at C(Zero) phases goes low and reset to “no hardware homing” after the occurrence 0Bh: latch for 4 continuous A,B,Z phase low points and reset to “no hardware homing” after completion . 0Ch: latch for 4 continuous Z phase low points and reset to “no hardware homing” after completion .

● **LSI3123 read hard homing flag**

Format : u32 Status = LSI3123_read_hard_homing_flag(u8 CardID,u8 Axis,u8 *flag);

Purpose: Read hard homing flag for checking the occurrence of counter clear.

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis

Output:

Name	Type	Description
flag	u8	0: not ready 1: cleared already and after reading, this flag will reset to 0 automatically. 2: latched positions in mode 0Bh, 0Ch of LSI3123A_set_hard_homing has completed, this flag will reset to 0 automatically.

● **LSI3123 soft homing command**

Format : u32 Status = LSI3123_soft_homing_command (u8 CardID,u8 Axis);

Purpose: To do software homing (software clear counter)

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis

Counter function group

● LSI3123 set counter mode

Format : u32 Status = LSI3123_set_counter_mode(u8 CardID,u8 Axis,u8 mode);

Purpose: Set counter working mode

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis
mode	u8	00h: quadrature mode, A lead B upcount (default) 01h: quadrature mode, B lead A upcount 02h: single clock mode. A->Clock B->Dir and up count while Dir=1 03h: single clock mode. A->Clock B->Dir and up count while Dir=0 04h: dual clock mode. A->CW B->CCW 05h: dual clock mode. A->CCW B->CW

● LSI3123 set quadrature times

Format : u32 Status = LSI3123_set_quadrature_times(u8 CardID,u8 Axis,u8 mode)

Purpose: Set the multiple rate of quadrature input (this function is valid while counter mode is set in quadrature mode)

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis
mode	u8	00h: input x4 (default) 01h: input x2 02h: input x1

● LSI3123 read counter

Format : u32 Status = LSI3123_read_counter(u8 CardID,u8 Axis,i32 *value);

Purpose: Read the counter value (on the fly)

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis

Output:

Name	Type	Description
value	i32	current counter value

● **LSI3123 load counter**

Format : u32 Sstatus = LSI3123_load_counter(u8 CardID,u8 Axis,i32 value);

Purpose: Load (set) the counter value.

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis
value	i32	counter value to be set

● **LSI3123 latch control**

Format : u32 Status = LSI3123_latch_control(u8 CardID,u8 Channel,u8 On_Off);

Purpose: To set the external latch input source.

(reference: LSI3123_latch_mode for working mode setting)

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Channel	u8	0: probe_in (latch IN0) 1: foot_SW_IN (latch IN1)
On_Off	u8	0: off (disable) 1: on (enable)

● **LSI3123 latch mode**

Format : u32 Status = LSI3123_latch_mode(u8 CardID,u8 mode);

Purpose: To set the working mode of latch function.

(reference: LSI3123_latch_control for external latch input source setting

LSI3123_read_latched_value for external trigger latched data reading

LSI3123_preset_counter for the data of external trigger to load counter)

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
mode	u8	0: continuous external trigger to latch mode 1: single shot external trigger to latch mode After the latched trigger latch_control will automatically reset to OFF (disable) 2: continuous external trigger to load mode 3: single shot external trigger to load mode After the latched trigger latch_control will automatically reset to OFF (disable)

● **LSI3123 preset counter**

Format : u32 Status = LSI3123_preset_counter(u8 CardID,u8 Axis,i32 value);

Purpose: Load (set) the preset-counter. Preset-counter is used to load counter while the counter latch mode is set at external trigger to load.

(reference: LSI3123_latch_mode)

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis
value	i32	counter value to be set

● **LSI3123 read latch flag**

Format : u32 Status = LSI3123_read_latch_flag(u8 CardID,u8 Channel,u8 *Flag);

Purpose: To read the latch flag

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Channel	u8	0: Probe_IN(Latch IN0) 1: Foot_SW_IN(Latch IN1)

Output:

Name	Type	Description
Flag	u8	0: no latch/load trigger occurs 1: latch/load trigger occurred

● **LSI3123 read latch value**

Format : u32 Status = LSI3123_read_latch_value(u8 CardID,u8 Axis,i32 *value);

Purpose: To read the latched value

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis

Output:

Name	Type	Description
value	i32	latched counter value

Security function group

● **LSI3123 set password**

Format : u32 Status = LSI3123_set_password(u8 CardID,u16 password[5]);

Purpose: To set password and enable the security function (security_enable will set to 1 , reference: LSI3123A_set_password) .

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
password	u16	the password in array format, each element in the range of 0~65535.

● **LSI3123 change password**

Format : u32 Status = LSI3123_change_password (u8 CardID, u16 Oldpassword[5] , u16 Newpassword[5]);

Purpose: To change password. When security_enable=1 (reference: LSI3123A_read_security_status) use this function will change old password to new one. If the old password is not correct, the security error counter will increase by 1, after 10 error trial, the card will be dead locked. If the change password is success, the security_enable will be 1 (the card security is enabled) and lock_status will be 0 (security lock is opened).

If Newpassword[5]=[0,0,0,0,0], the card will reset to security disabled state.

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Oldpassword	u16	the old (previous) password in array format, each element in the range of 0~65535.
Newpassword	u16	the new password to be set in array format, each element in the range of 0~65535.

● **LSI3123 clear password**

Format : u32 Status = LSI3123_clear_password(u8 CardID,u16 Password[5]);

Purpose: To clear password and diable the security function. If you try with error passwords more than 10 times, the card will be dead locked. Any further function of unlock will not response. You should send back to the maker to release the card. If clear password succeed, the security_enable will be 0 (the card security is disabled) and lock_status will be 0 (security lock is opened).

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
password	u16	the old password in array format, each element in the range of 0~65535.

● **LSI3123 unlock security**

Format : u32 Status = LSI3123_unlock_security(u8 CardID,u16 password[5]);

Purpose: To unlock the card. Use this function to open the card function only the card's lock_status(reference: LSI3123_read_security_status) is 1(locked).
If you try with error passwords more than 10 times, the card will be dead locked. Any further function of unlock will not response. You should send back to the maker to release the card.

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
password	u16	the password in array format, each element in the range of 0~65535.

● **LSI3123 read security status**

Format : u32 Status = LSI3123_read_security_status(u8 CardID,u8 *lock_status,
u8 *security_enable);

Purpose: Read the security status for checking if the security is opened, locked or dead locked.

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW

Output:

Name	Type	Description
lock_status	u8	0: security lock is opened, the card functions will work. 1: card is locked, the card functions will not work. 2: dead locked, the card has been open the security foe more than 10 times. Any further unlock process will not response. You should send back to the maker to release the locked state.
security_enable	u8	0: the card security is disabled 1: the card security is enabled

Miscellaneous function group

● LSI3123 read parameters

Format : u32 Status = LSI3123_read_parameters(u8 CardID,u8 Axis,u8 Parameter_NO,
u8 *Value);

Purpose: Read the parameters of any axis.

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis
Parameter_NO	u8	0: HOMING_MODE 1: COUNTER_MODE 2: QUADRATURE_TIMES 3: LATCH_MODE 4: LATCH_CONTROL

Output:

Name	Type	Description
value	u8	the value set by corresponding functions

● LSI3123 start homing calibration

Format : u32 Status = LSI3123_start_homing_calibration(u8 CardID,u8 Axis,
u8 scale_type,u8 *flag);

Purpose: use latched data (latched by function *LSI3123_set_hard_homing* mode 0B or 0C) as absolute reference mark to calculate the coordinate.

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis
scale_type	u8	0: Acu-rite others now not available, but on request

Output:

Name	Type	Description
flag	u8	0: fail maybe the reference mark position latched not accurate. 1: success, the coordinate re-build.

● **LSI3123 calibrate ABS position**

Format : u32 Status = LSI3123_calibrate_ABS_position(u8 CardID,u8 Axis,i32 position,
u8 scale_type,u8 *flag);

Purpose: use to setup the coordinate of a new scale.

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis
position	i32	the position coordinat that the scale stops at
scale_type	u8	0: Acu-rite others now not available, but on request

Output:

Name	Type	Description
flag	u8	0: fail maybe the reference mark position latched not accurate. 1: success, the coordinate build.

● **LSI3123 check homed calibration flag**

Format : u32 Status = LSI3123_check_homed_calibration_flag(u8 CardID,u8 Axis,u8
*flag);

Purpose: use to check if the system coordinate system has been built-up before system reboot.

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis

Output:

Name	Type	Description
flag	u8	0: fail, coordinate not build-up before system reboot. 1: success, the coordinate build-up before system boot.

7.5 Dll list

LSI3123A DLL(V5.0) Function Table		
Function Name		Description
1	LSI3123_initial ()	Initialize LSI3123 resource
2	LSI3123_close ()	Release LSI3123 resource
3	LSI3123_info ()	Access LSI3123 physical address
4	LSI3123_set_input_polarity ()	Set input polarity
5	LSI3123_set_led_polarity ()	Set the LED polarity
6	LSI3123_read_input_polarity ()	Read input polarity
7	LSI3123_read_input_status ()	Read the input status
8	LSI3123_set_hard_homing ()	Set the working mode of clearing counter
9	LSI3123_read_hard_homing_flag ()	Read hard homing flag
10	LSI3123_soft_homing_command ()	Software homing (software clear counter)
11	LSI3123_set_counter_mode ()	Set counter working mode
12	LSI3123_set_quadrature_times ()	Set the multiple rate of quadrature input
13	LSI3123_read_counter ()	Read the counter value
14	LSI3123_load_counter ()	Set the counter value
15	LSI3123_latch_control ()	Set the external latch input source
14	LSI3123_latch_mode ()	Set the working mode of latch function
17	LSI3123_preset_counter ()	Set the preset-counter
18	LSI3123_read_latch_flag ()	Read the latch flag
19	LSI3123_read_latch_value ()	Read the latched value
20	LSI3123_set_password ()	Set password and enable the security function
21	LSI3123_change_password ()	Change password
22	LSI3123_clear_password ()	Clear password and enable the security function
23	LSI3123_unlock_security ()	To unlock the card
24	LSI3123_read_security_status ()	Read the security status
25	LSI3123_read_parameters ()	Read the parameters of any axis
26	LSI3123_start_homing_calibration ()	Rebuild coordinate after power on
27	LSI3123_calibrate_ABS_position ()	Setup coordinate of a new scale
28	LSI3123_check_homed_calibration_flag ()	Check if system coordinate has been built-up before system reboot

8. LSI3123A Error codes summary

8.1 LSI3123A Error codes table

Error Code	Symbolic Name	Description
0	JSDRV_NO_ERROR	No error.
2	JSDRV_INIT_ERROR	Driver initial error
3	JSDRV_UNLOCK_ERROR	Security unlock failure
4	JSDRV_LOCK_COUNTER_ERROR	Dead lock, unlock failure more than 10 times
5	SDRV_SET_SECURITY_ERROR	Password overwrite error
100	DEVICE_RW_ERROR	Device Read/Write error
101	JSDRV_NO_CARD	No LSI3123A card on the system.
102	JSDRV_DUPLICATE_ID	LSI3123A CardID duplicate error.
300	JLSI_ID_ERROR	Function input parameter error. CardID setting error, CardID doesn't match the DIP (rotary) SW setting
305	JLSI_POINT_ERROR	Input function parameter error.
306	JLSI_AXIS_ERROR	Axis parameter error.
308	JLSI_HOME_MODE_ERROR	Mode parameter error.
309	JLSI_POLARITY_ERROR	Polarity parameter error
310	JLSI_ON_OFF_ERROR	On_off parameter error
400	JLSI_DRIVER_NOT_SUPPORT	Driver function not support error